

R7F0C014B2D、 R7F0C014L2D

用户手册 硬件篇

16位单芯片微控制器

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

Notice

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

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

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

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

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

注意事项

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

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

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

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

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

（注 1） 瑞萨电子：在本文档中指瑞萨电子株式会社及其控股子公司。

（注 2） 瑞萨电子产品：指瑞萨电子开发或生产的任何产品。

关于 CMOS 器件的注意事项

① 输入引脚处的施加电压波形

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

② 未使用的输入引脚的处理

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

③ ESD 防护措施

如果MOS器件周围有强电场，将会击穿氧化栅极，降低器件的工作性能。因此必须采取措施，尽可能防止静电产生。一旦有静电，必须立即释放。环境必须控制适当。如果空气干燥，应当使用加湿器。建议避免使用容易产生静电的绝缘体。半导体器件的存放和运输必须使用抗静电容器、静电屏蔽袋或导电材料包装。所有包括工作台和工作面的测试和测量工具必须良好接地。操作员应当佩戴手腕带以保证良好接地。不能用手直接接触半导体器件。对装配有半导体器件的PW板也应采取类似的静电防范措施。

④ 初始化之前的状态

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

⑤ 电源上电 / 断电序列

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

⑥ 断电状态期间的信号输入

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

本手册的使用方法

对 象 本手册是以理解 R7F0C014B2D、R7F0C014L2D 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：R7F0C014B2D
- 64 引脚：R7F0C014L2D

目 的 本手册以帮助用户理解以下构成所示的功能为目的。

构 成 R7F0C014B2D、R7F0C014L2D 的手册分为本手册和指令篇（RL78 族共通）共 2 册。

R7F0C014B2D、R7F0C014L2D 用户手册	RL78 族用户手册 软件篇
• 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 本手册的读者必须具备电气、逻辑电路和微控制器的基础知识。

- 要理解全部功能时
→ 按照目录的顺序阅读本手册。
- 寄存器格式的阅读方法
→ 关于方框（□）内的位号，其位的名称在汇编程序中被定义为保留字，而在编译程序中被 #pragma sfr 指令定义为 sfr 变量。
- 要详细了解 R7F0C014B2D、R7F0C014L2D 微控制器的指令功能时
→ 请参照另一册《RL78 族用户手册软件篇》（R01US0015E）。

- 凡例
- 数据的表示

有效低电平的表示

注

注意

备注

数字的表示
- : 左侧为高位，右侧为低位。

: xxx （在引脚或者信号名称上有上划线）

: 正文中加注的说明

: 阅读时需要特别注意的内容

: 征文的补充说明

: 二进制数 xxxx 或者 xxxxB

十进制数 xxxx

十六进制数 xxxxH

相关资料 相关资料可能是暂定版，但是没有“暂定”的表示。

芯片的相关资料

资料名称	资料号
R7F0C014B2D、R7F0C014L2D 用户手册硬件篇	本手册
RL78 族用户手册指令篇	R01US0015E

闪存编程器的相关资料 (用户手册)

资料名称	资料号
PG-FP5 闪存编程器用户手册	R20UT0008E

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

其他资料

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

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

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

目 录

第 1 章	概述	1
1.1	特点	1
1.2	产品型号一览表	2
1.3	引脚连接图 (Top View)	3
1.3.1	32 引脚产品	3
1.3.2	64 引脚产品	4
1.4	引脚名	5
1.5	框图	6
1.5.1	32 引脚产品	6
1.5.2	64 引脚产品	7
1.6	功能概要	8
第 2 章	引脚功能	10
2.1	端口功能	10
2.1.1	32 引脚产品	11
2.1.2	64 引脚产品	13
2.2	端口以外的功能	15
2.2.1	各产品配置的功能	15
2.2.2	功能说明	18
2.3	未使用引脚的处理	20
第 3 章	CPU 体系结构	21
3.1	存储空间	21
3.1.1	内部程序存储空间	25
3.1.2	镜像区	28
3.1.3	内部数据存储空间	30
3.1.4	特殊功能寄存器 (SFR: Special Function Register) 的区域	30
3.1.5	扩展特殊功能寄存器 (2nd SFR: 2nd Special Function Register) 的区域	30
3.1.6	数据存储器的寻址	31
3.2	处理器的寄存器	32
3.2.1	控制寄存器	32
3.2.2	通用寄存器	34
3.2.3	ES 寄存器和 CS 寄存器	35
3.2.4	特殊功能寄存器 (SFR: Special Function Register)	36
3.2.5	扩展特殊功能寄存器 (2nd SFR: 2nd Special Function Register)	42
3.3	指令地址的寻址	50
3.3.1	相对寻址	50
3.3.2	立即寻址	50
3.3.3	表间接寻址	51
3.3.4	寄存器直接寻址	51
3.4	处理数据地址的寻址	52
3.4.1	隐含寻址	52
3.4.2	寄存器寻址	52
3.4.3	直接寻址	53
3.4.4	短直接寻址	54
3.4.5	SFR 寻址	55
3.4.6	寄存器间接寻址	56
3.4.7	基址寻址	57
3.4.8	基址变址寻址	61

3.4.9	堆栈寻址	62
第 4 章	端口功能	66
4.1	端口功能	66
4.2	端口结构	66
4.2.1	端口 0	67
4.2.2	端口 1	76
4.2.3	端口 2	86
4.2.4	端口 3	88
4.2.5	端口 4	91
4.2.6	端口 5	96
4.2.7	端口 6	104
4.2.8	端口 7	106
4.2.9	端口 12	112
4.2.10	端口 13	116
4.2.11	端口 14	118
4.3	控制端口功能的寄存器	122
4.3.1	端口模式寄存器 (PMxx)	125
4.3.2	端口寄存器 (Pxx)	126
4.3.3	上拉电阻选择寄存器 (PUxx)	127
4.3.4	端口输入模式寄存器 (PIM0、PIM1、PIM3、PIM5)	128
4.3.5	端口输出模式寄存器 (POM0、POM1、POM3、POM5、POM7)	129
4.3.6	端口模式控制寄存器 0、12、14 (PMC0、PMC12、PMC14)	130
4.3.7	A/D 端口配置寄存器 (ADPC)	131
4.3.8	外围 I/O 重定向寄存器 0 (PIOR0)	132
4.3.9	外围 I/O 重定向寄存器 1 (PIOR1)	133
4.3.10	全局数字输入禁止寄存器 (GDIDIS)	134
4.4	端口功能的运行	135
4.4.1	输入 / 输出端口的写操作	135
4.4.2	输入 / 输出端口的读操作	135
4.4.3	输入 / 输出端口的运算	135
4.4.4	通过 $EV_{DD} \leq V_{DD}$ 进行的不同电位 (1.8V、2.5V、3V) 的对应	136
4.4.5	通过输入 / 输出缓冲器进行的不同电位 (1.8V、2.5V、3V) 的对应	136
4.5	使用复用功能时的端口相关寄存器的设定	138
4.6	使用端口功能时的注意事项	143
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	143
4.6.2	有关 64 引脚以外的产品的引脚设定的注意事项	144
第 5 章	时钟发生电路	145
5.1	时钟发生电路的功能	145
5.2	时钟发生电路的结构	147
5.3	控制时钟发生电路的寄存器	149
5.3.1	时钟运行模式控制寄存器 (CMC)	149
5.3.2	系统时钟控制寄存器 (CKC)	152
5.3.3	时钟运行状态控制寄存器 (CSC)	153
5.3.4	振荡稳定时间计数器的状态寄存器 (OSTC)	154
5.3.5	振荡稳定时间选择寄存器 (OSTS)	156
5.3.6	外围允许寄存器 0、1 (PER0、PER1)	157
5.3.7	副系统时钟提供模式控制寄存器 (OSMC)	162
5.3.8	高速内部振荡器的频率选择寄存器 (HOCODIV)	163
5.3.9	高速内部振荡器的微调寄存器 (HIOTRM)	164
5.4	系统时钟振荡电路	165

5.4.1	X1 振荡电路	165
5.4.2	XT1 振荡电路	165
5.4.3	高速内部振荡器	169
5.4.4	低速内部振荡器	169
5.5	时钟发生电路的运行	170
5.6	时钟控制	172
5.6.1	高速内部振荡器的设定例子	172
5.6.2	X1 振荡电路的设定例子	174
5.6.3	XT1 振荡电路的设定例子	175
5.6.4	CPU 时钟的状态转移图	176
5.6.5	CPU 时钟转移前的条件和转移后的处理	182
5.6.6	切换 CPU 时钟和系统时钟所需要的时间	184
5.6.7	时钟振荡停止前的条件	185
5.7	谐振器和振荡电路常数	185
第 6 章	定时器阵列单元	188
6.1	定时器阵列单元的功能	189
6.1.1	独立通道运行功能	189
6.1.2	多通道联动运行功能	190
6.1.3	8 位定时器运行功能（只限于通道 1 和通道 3）	191
6.1.4	LIN-bus 支持功能（只限于单元 0 的通道 3）	191
6.2	定时器阵列单元的结构	192
6.2.1	定时器计数寄存器 mn (TCRmn)	196
6.2.2	定时器数据寄存器 mn (TDRmn)	197
6.3	控制定时器阵列单元的寄存器	198
6.3.1	外围允许寄存器 0 (PER0)	199
6.3.2	定时器时钟选择寄存器 m (TPSm)	200
6.3.3	定时器模式寄存器 mn (TMRmn)	203
6.3.4	定时器状态寄存器 mn (TSRmn)	208
6.3.5	定时器通道允许状态寄存器 m (TEm)	209
6.3.6	定时器通道开始寄存器 m (TSM)	210
6.3.7	定时器通道停止寄存器 m (TTm)	211
6.3.8	定时器输入选择寄存器 0 (TIS0)	212
6.3.9	定时器输出允许寄存器 m (TOEm)	213
6.3.10	定时器输出寄存器 m (TOM)	214
6.3.11	定时器输出电平寄存器 m (TOLm)	215
6.3.12	定时器输出模式寄存器 m (TOMm)	216
6.3.13	输入切换控制寄存器 (ISC)	217
6.3.14	噪声滤波器允许寄存器 1 (NFEN1)	218
6.3.15	控制定时器输入 / 输出引脚端口功能的寄存器	219
6.4	定时器阵列单元的基本规则	220
6.4.1	多通道联动运行功能的基本规则	220
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）	222
6.5	计数器的运行	223
6.5.1	计数时钟 (f_{TCLK})	223
6.5.2	计数器的开始时序	225
6.5.3	计数器的运行	226
6.6	通道输出 (TOMn 引脚) 的控制	231
6.6.1	TOMn 引脚输出电路的结构	231
6.6.2	TOMn 引脚的输出设定	232
6.6.3	通道输出运行的注意事项	233
6.6.4	TOMn 位的一次性操作	237

6.6.5	有关开始计数时的定时器中断和 T0mn 引脚输出	238
6.7	定时器输入 (TImn) 的控制	239
6.7.1	TImn 引脚输入电路的结构	239
6.7.2	噪声滤波器	239
6.7.3	操作通道输入时的注意事项	240
6.8	定时器阵列单元的独立通道运行功能	241
6.8.1	作为间隔定时器 / 方波输出的运行	241
6.8.2	作为外部事件计数器的运行	246
6.8.3	作为分频器的运行 (只限于单元 0 的通道 0)	250
6.8.4	作为输入脉冲间隔测量的运行	254
6.8.5	作为输入信号高低电平宽度测量的运行	258
6.8.6	作为延迟计数器的运行	262
6.9	定时器阵列单元的多通道联动运行功能	266
6.9.1	作为单触发脉冲输出功能的运行	266
6.9.2	作为 PWM 功能的运行	273
6.9.3	作为多重 PWM 输出功能的运行	280
6.10	使用定时器阵列单元时的注意事项	288
6.10.1	使用定时器输出时的注意事项	288
第 7 章	定时器 RJ	289
7.1	定时器 RJ 的功能	289
7.2	定时器 RJ 的结构	290
7.3	控制定时器 RJ 的寄存器	291
7.3.1	外围允许寄存器 1 (PER1)	292
7.3.2	副系统时钟提供模式控制寄存器 (OSMC)	293
7.3.3	定时器 RJ 计数寄存器 0 (TRJ0)	294
7.3.4	定时器 RJ 控制寄存器 0 (TRJCR0)	295
7.3.5	定时器 RJ I/O 控制寄存器 0 (TRJIOC0)	296
7.3.6	定时器 RJ 控制寄存器 0 (TRJMR0)	298
7.3.7	定时器 RJ 事件引脚选择寄存器 0 (TRJISR0)	299
7.3.8	端口模式寄存器 0、3、4、5 (PM0、PM3、PM4、PM5)	300
7.4	定时器 RJ 的运行	301
7.4.1	重加载寄存器和计数器的改写	301
7.4.2	定时器模式	302
7.4.3	脉冲输出模式	303
7.4.4	事件计数器模式	304
7.4.5	脉宽测量模式	305
7.4.6	脉冲周期测量模式	306
7.4.7	与事件链接控制器 (ELC) 的协作	307
7.4.8	各模式的输出设定	307
7.5	使用定时器 RJ 时的注意事项	308
7.5.1	计数的开始和停止控制	308
7.5.2	标志的存取 (TRJCR0 寄存器的 TEDGF 位和 TUNDF 位)	308
7.5.3	计数寄存器的存取	308
7.5.4	模式的变更	308
7.5.5	TRJ00 引脚和 TRJIO0 引脚的设定步骤	309
7.5.6	不使用定时器 RJ 的情况	309
7.5.7	定时器 RJ 运行时钟的停止	309
7.5.8	STOP 模式 (事件计数器模式) 的设定步骤	309
7.5.9	STOP 模式中 (只限于事件计数器模式) 的功能限制	310
7.5.10	通过 TSTOP 位进行强制的计数停止	310
7.5.11	数字滤波器	310

7.5.12	选择 f_{IL} 作为计数源的情况	310
第 8 章	定时器 RD	311
8.1	定时器 RD 的功能	311
8.2	定时器 RD 的结构	312
8.3	控制定时器 RD 的寄存器	313
8.3.1	外围允许寄存器 1 (PER1)	314
8.3.2	定时器 RD ELC 寄存器 (TRDELIC)	315
8.3.3	定时器 RD 启动寄存器 (TRDSTR)	316
8.3.4	定时器 RD 模式寄存器 (TRDMR)	317
8.3.5	定时器 RD PWM 功能选择寄存器 (TRDPMR)	318
8.3.6	定时器 RD 功能控制寄存器 (TRDFCR)	319
8.3.7	定时器 RD 输出主允许寄存器 1 (TRDOER1)	320
8.3.8	定时器 RD 输出主允许寄存器 2 (TRDOER2)	321
8.3.9	定时器 RD 输出控制寄存器 (TRDOCR)	322
8.3.10	定时器 RD 数字滤波器功能选择寄存器 i (TRDDFi) (i=0、1)	325
8.3.11	定时器 RD 控制寄存器 i (TRDCRi) (i=0、1)	327
8.3.12	定时器 RD I/O 控制寄存器 Ai (TRDIORAi) (i=0、1)	332
8.3.13	定时器 RD I/O 控制寄存器 Ci (TRDIORCi) (i=0、1)	334
8.3.14	定时器 RD 状态寄存器 0 (TRDSR0)	336
8.3.15	定时器 RD 状态寄存器 1 (TRDSR1)	340
8.3.16	定时器 RD 中断允许寄存器 i (TRDIERi) (i=0、1)	344
8.3.17	定时器 RD PWM 功能输出电平控制寄存器 i (TRDPOCRi) (i=0、1)	345
8.3.18	定时器 RD 计数器 i (TRDi) (i=0、1)	346
8.3.19	定时器 RD 通用寄存器 Ai、Bi、Ci、Di (TRDGRAi、TRDGRBi、TRDGRCi、TRDGRDi) (i=0、1)	348
8.3.20	端口模式寄存器 1 (PM1)	357
8.4	有关多个模式的共同事项	358
8.4.1	计数源	358
8.4.2	缓冲器运行	359
8.4.3	同步运行	362
8.4.4	脉冲输出的强制截止	363
8.4.5	从事件链接控制器 (ELC) 输入的事件	365
8.4.6	向事件链接控制器 (ELC) / 数据传送控制器 (DTC) 输出的事件	365
8.5	定时器 RD 的运行	366
8.5.1	输入捕捉功能	366
8.5.2	输出比较功能	370
8.5.3	PWM 功能	375
8.5.4	复位同步 PWM 模式	379
8.5.5	互补 PWM 模式	382
8.5.6	PWM3 模式	386
8.6	定时器 RD 中断	389
8.7	使用定时器 RD 时的注意事项	391
8.7.1	SFR 的读写存取	391
8.7.2	模式的切换	391
8.7.3	计数源	392
8.7.4	输入捕捉功能	392
8.7.5	TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚的设定步骤 (i=0、1)	392
8.7.6	外部时钟 TRDCLK	392
8.7.7	复位同步 PWM 模式	393
8.7.8	互补 PWM 模式	393

第 9 章 实时时钟	397
9.1 实时时钟的功能	397
9.2 实时时钟的结构	397
9.3 控制实时时钟的寄存器	399
9.3.1 外围允许寄存器 0 (PER0)	400
9.3.2 副系统时钟提供模式控制寄存器 (OSMC)	401
9.3.3 实时时钟控制寄存器 0 (RTCC0)	402
9.3.4 实时时钟控制寄存器 1 (RTCC1)	403
9.3.5 秒计数寄存器 (SEC)	405
9.3.6 分钟计数寄存器 (MIN)	405
9.3.7 小时计数寄存器 (HOUR)	406
9.3.8 日计数寄存器 (DAY)	408
9.3.9 星期计数寄存器 (WEEK)	409
9.3.10 月计数寄存器 (MONTH)	410
9.3.11 年计数寄存器 (YEAR)	410
9.3.12 时钟误差校正寄存器 (SUBCUD)	411
9.3.13 闹钟分钟寄存器 (ALARMWM)	412
9.3.14 闹钟小时寄存器 (ALARMWH)	412
9.3.15 闹钟星期寄存器 (ALARMWW)	412
9.3.16 端口模式寄存器 3 (PM3)	413
9.3.17 端口寄存器 3 (P3)	413
9.4 实时时钟的运行	414
9.4.1 实时时钟的运行开始	414
9.4.2 开始运行后 HALT/STOP 模式的转移	415
9.4.3 实时时钟计数器的读写	416
9.4.4 实时时钟的闹钟设定	418
9.4.5 实时时钟的 1Hz 输出	419
9.4.6 实时时钟的时钟误差校正例子	420
第 10 章 12 位间隔定时器	425
10.1 12 位间隔定时器的功能	425
10.2 12 位间隔定时器的结构	425
10.3 控制 12 位间隔定时器的寄存器	426
10.3.1 外围允许寄存器 0 (PER0)	426
10.3.2 副系统时钟提供模式控制寄存器 (OSMC)	427
10.3.3 12 位间隔定时器的控制寄存器 (ITMC)	428
10.4 12 位间隔定时器的运行	429
10.4.1 12 位间隔定时器的运行时序	429
10.4.2 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	430
第 11 章 时钟输出 / 蜂鸣器输出控制电路	431
11.1 时钟输出 / 蜂鸣器输出控制电路的功能	431
11.2 时钟输出 / 蜂鸣器输出控制电路的结构	433
11.3 控制时钟输出 / 蜂鸣器输出控制电路的寄存器	433
11.3.1 时钟输出选择寄存器 n (CKSn)	433
11.3.2 控制时钟输出 / 蜂鸣器输出引脚端口功能的寄存器	435
11.4 时钟输出 / 蜂鸣器输出控制电路的运行	436
11.4.1 输出引脚的运行	436
11.5 时钟输出 / 蜂鸣器输出控制电路的注意事项	436
第 12 章 看门狗定时器	437
12.1 看门狗定时器的功能	437

12.2	看门狗定时器的结构	438
12.3	控制看门狗定时器的寄存器	439
12.3.1	看门狗定时器的允许寄存器 (WDTE)	439
12.4	看门狗定时器的运行	440
12.4.1	看门狗定时器的运行控制	440
12.4.2	看门狗定时器上溢时间的设定	441
12.4.3	看门狗定时器窗口打开期间的设定	442
12.4.4	看门狗定时器间隔中断的设定	443
第 13 章	A/D 转换器	444
13.1	A/D 转换器的功能	444
13.2	A/D 转换器的结构	446
13.3	控制 A/D 转换器的寄存器	448
13.3.1	外围允许寄存器 0 (PER0)	449
13.3.2	A/D 转换器的模式寄存器 0 (ADM0)	450
13.3.3	A/D 转换器的模式寄存器 1 (ADM1)	458
13.3.4	A/D 转换器的模式寄存器 2 (ADM2)	459
13.3.5	10 位 A/D 转换结果寄存器 (ADCR)	461
13.3.6	8 位 A/D 转换结果寄存器 (ADCRH)	461
13.3.7	模拟输入通道指定寄存器 (ADS)	462
13.3.8	转换结果比较上限值设定寄存器 (ADUL)	464
13.3.9	转换结果比较下限值设定寄存器 (ADLL)	464
13.3.10	A/D 测试寄存器 (ADTES)	465
13.3.11	控制模拟输入引脚端口功能的寄存器	466
13.4	A/D 转换器的转换运行	467
13.5	输入电压和转换结果	469
13.6	A/D 转换器的运行模式	470
13.6.1	软件触发模式 (选择模式、连续转换模式)	470
13.6.2	软件触发模式 (选择模式、单次转换模式)	471
13.6.3	软件触发模式 (扫描模式、连续转换模式)	472
13.6.4	软件触发模式 (扫描模式、单次转换模式)	473
13.6.5	硬件触发无等待模式 (选择模式、连续转换模式)	474
13.6.6	硬件触发无等待模式 (选择模式、单次转换模式)	475
13.6.7	硬件触发无等待模式 (扫描模式、连续转换模式)	476
13.6.8	硬件触发无等待模式 (扫描模式、单次转换模式)	477
13.6.9	硬件触发等待模式 (选择模式、连续转换模式)	478
13.6.10	硬件触发等待模式 (选择模式、单次转换模式)	479
13.6.11	硬件触发等待模式 (扫描模式、连续转换模式)	480
13.6.12	硬件触发等待模式 (扫描模式、单次转换模式)	481
13.7	A/D 转换器的设定流程图	482
13.7.1	软件触发模式的设定	482
13.7.2	硬件触发无等待模式的设定	483
13.7.3	硬件触发等待模式的设定	484
13.7.4	选择温度传感器的输出电压 / 内部基准电压时的设定 (以软件触发模式、单次转换模式为例)	485
13.7.5	测试模式的设定	486
13.8	SNOOZE 模式功能	487
13.9	A/D 转换器特性表的阅读方法	491
13.10	A/D 转换器的注意事项	494
第 14 章	串行阵列单元	498
14.1	串行阵列单元的功能	499

14.1.1	3 线串行 I/O (CSI00、CSI01、CSI10、CSI11、CSI20、CSI21)	499
14.1.2	UART (UART0 ~ UART2)	500
14.1.3	简易 I ² C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)	501
14.2	串行阵列单元的结构	502
14.2.1	移位寄存器	505
14.2.2	串行数据寄存器 mn (SDRmn) 的低 8 位或者低 9 位	505
14.3	控制串行阵列单元的寄存器	507
14.3.1	外围允许寄存器 0 (PER0)	508
14.3.2	串行时钟选择寄存器 m (SPSm)	509
14.3.3	串行模式寄存器 mn (SMRmn)	510
14.3.4	串行通信运行设定寄存器 mn (SCRmn)	511
14.3.5	串行数据寄存器 mn (SDRmn)	514
14.3.6	串行标志清除触发寄存器 mn (SIRmn)	515
14.3.7	串行状态寄存器 mn (SSRmn)	516
14.3.8	串行通道开始寄存器 m (SSm)	518
14.3.9	串行通道停止寄存器 m (STm)	519
14.3.10	串行通道允许状态寄存器 m (SEm)	520
14.3.11	串行输出允许寄存器 m (SOEm)	521
14.3.12	串行输出寄存器 m (SOM)	522
14.3.13	串行输出电平寄存器 m (SOLm)	523
14.3.14	串行待机控制寄存器 m (SSCm)	525
14.3.15	输入切换控制寄存器 (ISC)	526
14.3.16	噪声滤波器允许寄存器 0 (NFEN0)	527
14.3.17	控制串行输入 / 输出引脚端口功能的寄存器	528
14.4	运行停止模式	529
14.4.1	以单元为单位停止运行的情况	529
14.4.2	按通道停止运行的情况	530
14.5	3 线串行 I/O (CSI00、CSI01、CSI10、CSI11、CSI20、CSI21) 通信的运行	531
14.5.1	主控发送	533
14.5.2	主控接收	541
14.5.3	主控的发送和接收	549
14.5.4	从属发送	557
14.5.5	从属接收	565
14.5.6	从属发送和接收	571
14.5.7	SNOOZE 模式功能	579
14.5.8	传送时钟频率的计算	583
14.5.9	在 3 线串行 I/O (CSI00、CSI01、CSI10、CSI11、CSI20、CSI21) 通信过程中 发生错误时的处理步骤	585
14.6	从属选择输入功能的时钟同步串行通信的运行	586
14.6.1	从属发送	589
14.6.2	从属接收	599
14.6.3	从属的发送和接收	606
14.6.4	传送时钟频率的计算	616
14.6.5	在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤	617
14.7	UART (UART0 ~ UART2) 通信的运行	618
14.7.1	UART 发送	620
14.7.2	UART 接收	629
14.7.3	SNOOZE 模式功能	636
14.7.4	波特率的计算	643
14.7.5	在 UART (UART0 ~ UART2) 通信过程中发生错误时的处理步骤	647
14.8	LIN 通信的运行	648
14.8.1	LIN 发送	648

14.8.2	LIN 接收	651
14.9	简易 I ² C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21) 通信的运行	656
14.9.1	地址段的发送	658
14.9.2	数据发送	663
14.9.3	数据接收	666
14.9.4	停止条件的产生	670
14.9.5	传送速率的计算	671
14.9.6	在简易 I ² C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21) 通信过程中 发生错误时的处理步骤	673
第 15 章	串行接口 IICA	674
15.1	串行接口 IICA 的功能	674
15.2	串行接口 IICA 的结构	677
15.3	控制串行接口 IICA 的寄存器	679
15.3.1	外围允许寄存器 0 (PER0)	680
15.3.2	IICA 控制寄存器 n0 (IICCTLn0)	680
15.3.3	IICA 状态寄存器 n (IICSn)	685
15.3.4	IICA 标志寄存器 n (IICFn)	687
15.3.5	IICA 控制寄存器 n1 (IICCTLn1)	689
15.3.6	IICA 低电平宽度设定寄存器 n (IICWLn)	691
15.3.7	IICA 高电平宽度设定寄存器 n (IICWHn)	691
15.3.8	端口模式寄存器 6 (PM6)	692
15.4	I ² C 总线模式的功能	693
15.4.1	引脚结构	693
15.4.2	通过 IICWLn 寄存器和 IICWHn 寄存器设定传送时钟的方法	694
15.5	I ² C 总线的定义和控制方法	695
15.5.1	开始条件	695
15.5.2	地址	696
15.5.3	传送方向的指定	696
15.5.4	应答 (ACK)	697
15.5.5	停止条件	698
15.5.6	等待	699
15.5.7	等待的解除方法	701
15.5.8	中断请求 (INTIICAn) 的产生时序和等待控制	702
15.5.9	地址匹配的检测方法	703
15.5.10	错误的检测	703
15.5.11	扩展码	704
15.5.12	仲裁	705
15.5.13	唤醒功能	707
15.5.14	通信预约	710
15.5.15	其他注意事项	714
15.5.16	通信运行	715
15.5.17	I ² C 中断请求 (INTIICAn) 的产生时序	723
15.6	时序图	744
第 16 章	数据传送控制器 (DTC)	759
16.1	DTC 的功能	759
16.2	DTC 的结构	760
16.3	控制 DTC 的寄存器	761
16.3.1	DTC 控制数据区和 DTC 向量表区的分配	762
16.3.2	控制数据的分配	763
16.3.3	向量表	763

16.3.4	外围允许寄存器 1 (PER1)	765
16.3.5	DTC 控制寄存器 j (DTCCRj) (j=0 ~ 23)	766
16.3.6	DTC 块大小寄存器 j (DTBLSj) (j=0 ~ 23)	767
16.3.7	DTC 传送次数寄存器 j (DTCCTj) (j=0 ~ 23)	767
16.3.8	DTC 传送次数重加载寄存器 j (DTRL Dj) (j=0 ~ 23)	768
16.3.9	DTC 源地址寄存器 j (DTSARj) (j=0 ~ 23)	768
16.3.10	DTC 目标地址寄存器 j (DTDARj) (j=0 ~ 23)	768
16.3.11	DTC 启动允许寄存器 i (DTCENi) (i=0 ~ 4)	769
16.3.12	DTC 基址寄存器 (DTCBAR)	771
16.4	DTC 的运行	772
16.4.1	启动源	772
16.4.2	正常模式	773
16.4.3	重复模式	776
16.4.4	链传送	779
16.5	使用 DTC 时的注意事项	781
16.5.1	DTC 控制数据和向量表的设定	781
16.5.2	DTC 控制数据区和 DTC 向量表区的分配	781
16.5.3	DTC 保留指令	781
16.5.4	存取数据闪存空间时的运行	781
16.5.5	DTC 的执行时钟数	782
16.5.6	DTC 的响应时间	783
16.5.7	DTC 的启动源	783
16.5.8	待机模式中的运行	784
第 17 章	事件链接控制器 (ELC)	785
17.1	ELC 的功能	785
17.2	ELC 的结构	785
17.3	控制 ELC 的寄存器	786
17.3.1	事件输出目标选择寄存器 n (ELSELRn) (n=00 ~ 17)	787
17.4	ELC 的运行	790
第 18 章	中断功能	792
18.1	中断功能的种类	792
18.2	中断源和结构	792
18.3	控制中断功能的寄存器	799
18.3.1	中断请求标志寄存器 (IF0L、IF0H、IF1L、IF1H、IF2L、IF2H)	803
18.3.2	中断屏蔽标志寄存器 (MK0L、MK0H、MK1L、MK1H、MK2L、MK2H)	805
18.3.3	优先级指定标志寄存器 (PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、 PR10L、PR10H、PR11L、PR11H、PR12L、PR12H)	807
18.3.4	外部中断上升沿允许寄存器 (EGP0、EGP1) 和 外部中断下降沿允许寄存器 (EGN0、EGN1)	810
18.3.5	程序状态字 (PSW)	812
18.4	中断处理的操作	813
18.4.1	可屏蔽中断请求的接受	813
18.4.2	软件中断请求的接受	816
18.4.3	多重中断处理	816
18.4.4	除法运算指令执行过程中的中断处理	820
18.4.5	中断请求的保留	821
第 19 章	键中断功能	822
19.1	键中断的功能	822
19.2	键中断的结构	822

19.3 控制键中断的寄存器	824
19.3.1 键返回模式寄存器 (KRM)	824
19.3.2 端口模式寄存器 7 (PM7)	825
第 20 章 待机功能	826
20.1 待机功能	826
20.2 控制待机功能的寄存器	827
20.3 待机功能的运行	827
20.3.1 HALT 模式	827
20.3.2 STOP 模式	832
20.3.3 SNOOZE 模式	837
第 21 章 复位功能	840
21.1 复位时序	842
21.2 确认复位源的寄存器	846
21.2.1 复位控制标志寄存器 (RESF)	846
第 22 章 上电复位电路	849
22.1 上电复位电路的功能	849
22.2 上电复位电路的结构	850
22.3 上电复位电路的运行	850
第 23 章 电压检测电路	854
23.1 电压检测电路的功能	854
23.2 电压检测电路的结构	855
23.3 控制电压检测电路的寄存器	856
23.3.1 电压检测寄存器 (LVIM)	856
23.3.2 电压检测电平选择寄存器 (LVIS)	857
23.4 电压检测电路的运行	860
23.4.1 用作复位模式时的设定	860
23.4.2 用作中断模式时的设定	862
23.4.3 用作中断 & 复位模式时的设定	864
23.5 电压检测电路的注意事项	869
第 24 章 安全功能	871
24.1 安全功能的概要	871
24.2 安全功能使用的寄存器	872
24.3 安全功能的运行	872
24.3.1 闪存 CRC 运算功能 (高速 CRC)	872
24.3.2 CRC 运算功能 (通用 CRC)	876
24.3.3 RAM 奇偶校验错误检测功能	878
24.3.4 RAM 保护功能	880
24.3.5 SFR 保护功能	881
24.3.6 非法存储器存取检测功能	882
24.3.7 频率检测功能	884
24.3.8 A/D 测试功能	886
24.3.9 输入 / 输出引脚的数字输出信号电平检测功能	891
第 25 章 稳压器	892
25.1 稳压器的概要	892
第 26 章 选项字节	893
26.1 选项字节的功能	893

26.1.1	用户选项字节 (000C0H ~ 000C2H/010C0H ~ 010C2H)	893
26.1.2	片上调试选项字节 (000C3H/010C3H)	894
26.2	用户选项字节的格式	895
26.3	片上调试选项字节的格式	901
26.4	选项字节的设定	902
第 27 章	闪存	903
27.1	使用闪存编程器的串行编程	904
27.1.1	编程环境	906
27.1.2	通信方式	906
27.2	使用外部器件 (内置 UART) 的串行编程	907
27.2.1	编程环境	907
27.2.2	通信方式	908
27.3	电路板上的引脚处理	909
27.3.1	P40/TOOL0 引脚	909
27.3.2	$\overline{\text{RESET}}$ 引脚	909
27.3.3	端口引脚	910
27.3.4	REGC 引脚	910
27.3.5	X1 引脚和 X2 引脚	910
27.3.6	电源	910
27.4	编程方法	911
27.4.1	串行编程的步骤	911
27.4.2	闪存编程模式	912
27.4.3	通信方式	914
27.4.4	通信命令	915
27.5	使用 PG-FP5 时的各命令处理时间 (参考值)	917
27.6	自编程	918
27.6.1	自编程的步骤	919
27.6.2	引导交换功能	920
27.6.3	闪存屏蔽窗口功能	922
27.7	安全设定	923
27.8	数据闪存	925
27.8.1	数据闪存的概要	925
27.8.2	控制数据闪存的寄存器	925
27.8.3	数据闪存的存取步骤	926
第 28 章	片上调试功能	927
28.1	和 E1 片上调试仿真器的连接	927
28.2	片上调试安全 ID	928
28.3	用户资源的确保	928
第 29 章	十进制校正 (BCD) 电路	930
29.1	十进制校正电路的功能	930
29.2	十进制校正电路使用的寄存器	930
29.2.1	BCD 校正结果寄存器 (BCDADJ)	930
29.3	十进制校正电路的运行	931
第 30 章	指令集的概要	933
30.1	凡例	933
30.1.1	操作数的表现形式和记述方法	933
30.1.2	操作栏的说明	934
30.1.3	标志栏的说明	935

30.1.4	PREFIX 指令	935
30.2	操作一览表	936
第 31 章	电特性	954
31.1	绝对最大额定值	955
31.2	振荡电路特性	957
31.2.1	X1、XT1 振荡电路特性	957
31.2.2	内部振荡器特性	957
31.3	DC 特性	958
31.3.1	引脚特性	958
31.3.2	电源电流特性	963
31.4	AC 特性	968
31.5	外围功能特性	974
31.5.1	串行阵列单元	974
31.5.2	串行接口 IICA	1003
31.6	模拟特性	1007
31.6.1	A/D 转换器特性	1007
31.6.2	温度传感器 / 内部基准电压的特性	1011
31.6.3	POR 电路特性	1011
31.6.4	LVD 电路特性	1012
31.6.5	电源电压的上升斜率特性	1012
31.7	数据存储器的保持特性	1013
31.8	闪存编程特性	1013
31.9	专用闪存编程器通信 (UART)	1013
31.10	闪存编程模式的进入时序	1014
第 32 章	封装尺寸图	1015
32.1	32 引脚产品	1015
32.2	64 引脚产品	1016
附录 A	修订记录	1018

第 1 章 概述

1.1 特点

- 能更改高速（0.03125 μ s：高速内部振荡器时钟 32MHz 运行）到低速（30.5 μ s：副系统时钟 32.768kHz 运行）的最短指令执行时间。
- 通用寄存器：(8 位寄存器 $\times$ 8 个) $\times$ 4 组
- ROM：128KB，RAM：8KB，数据闪存：8KB
- 内置高速内部振荡器时钟。
 - 可选择 64MHz(TYP.)、48MHz(TYP.)、32MHz(TYP.)、24MHz(TYP.)、16MHz(TYP.)、12MHz(TYP.)、8MHz(TYP.)、6MHz(TYP.)、4MHz(TYP.)、3MHz(TYP.)、2MHz(TYP.) 或者 1MHz(TYP.)。
- 内置单电源闪存（有块擦除/编程禁止功能）。
- 支持自编程功能（有引导交换功能/闪存屏蔽窗口功能）。
- 内置片上调试功能。
- 内置上电复位（POR）电路和电压检测电路（LVD）。
- 内置看门狗定时器（能以专用的低速内部振荡器时钟运行）。
- 支持乘除和乘加运算指令。
- 内置键中断功能。
- 内置时钟输出/蜂鸣器输出的控制电路。
- 内置十进制校正（BCD）电路。
- I/O 端口：28 $\sim$ 58 个（N 沟道漏极开路：3 $\sim$ 4 个）
- 定时器
 - 16 位定时器：7 个通道（TAU：4 个通道，定时器 RJ：1 个通道，定时器 RD：2 个通道）
 - 看门狗定时器：1 个通道
 - 实时时钟：0 $\sim$ 1 个通道（校正时钟的输出）
 - 12 位间隔定时器：1 个通道
- 串行接口
 - CSI
 - UART/UART（支持 LIN-bus）
 - I²C/简易 I²C
- 不同电位接口：在 4.0 $\sim$ 5.5V 工作时，和 2.5V、3V 工作的器件连接。
- 8/10 位分辨率 A/D 转换器（V_{DD}=EV_{DD}=1.6 $\sim$ 5.5V）：8 $\sim$ 12 个通道
- 待机功能：HALT、STOP、SNOOZE 模式
- 内置数据传送控制器（DTC）。
- 内置事件链接控制器（ELC）。
- 电源电压：V_{DD}=1.6 $\sim$ 5.5V
- 工作环境温度：T_A=−40 $\sim$ +85°C

备注 内置的功能因产品而不同。请参照“1.6 功能概要”。

○ ROM、RAM 容量

闪存ROM	数据闪存	RAM	32 引脚	64 引脚
128KB	8KB	8KB 注	R7F0C014B2D	R7F0C014L2D

注 在使用自编程功能和数据闪存功能时，大约为 7KB。

1.2 产品型号一览表

图 1-1 产品型号、存储容量和封装

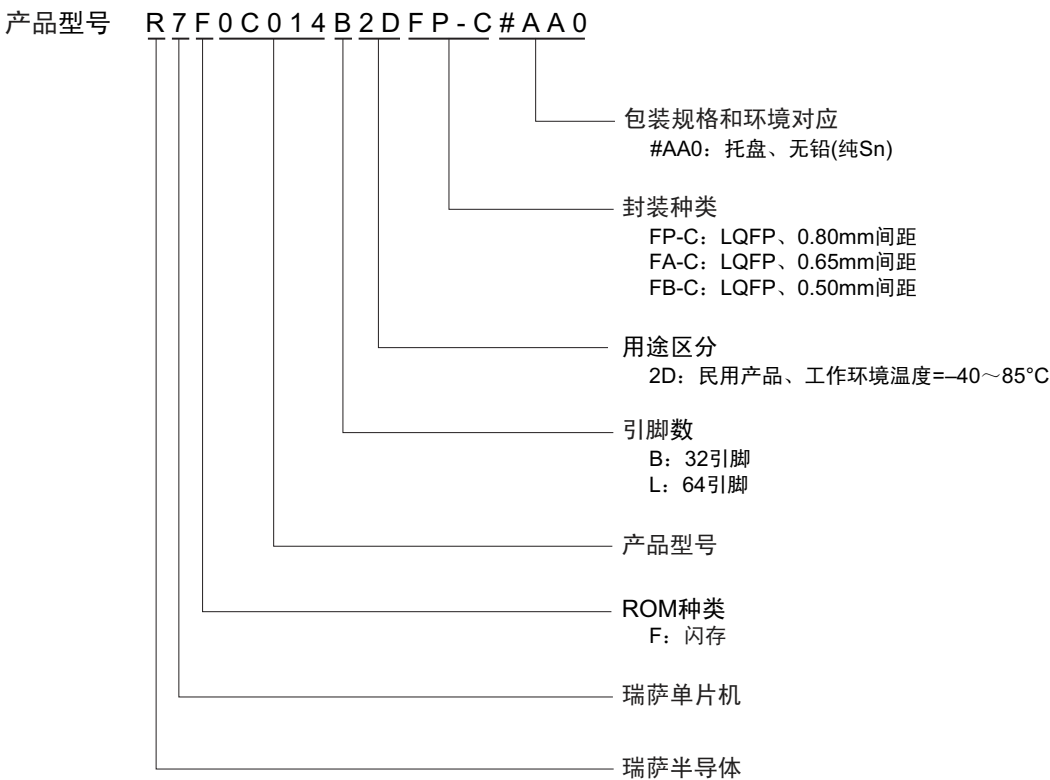


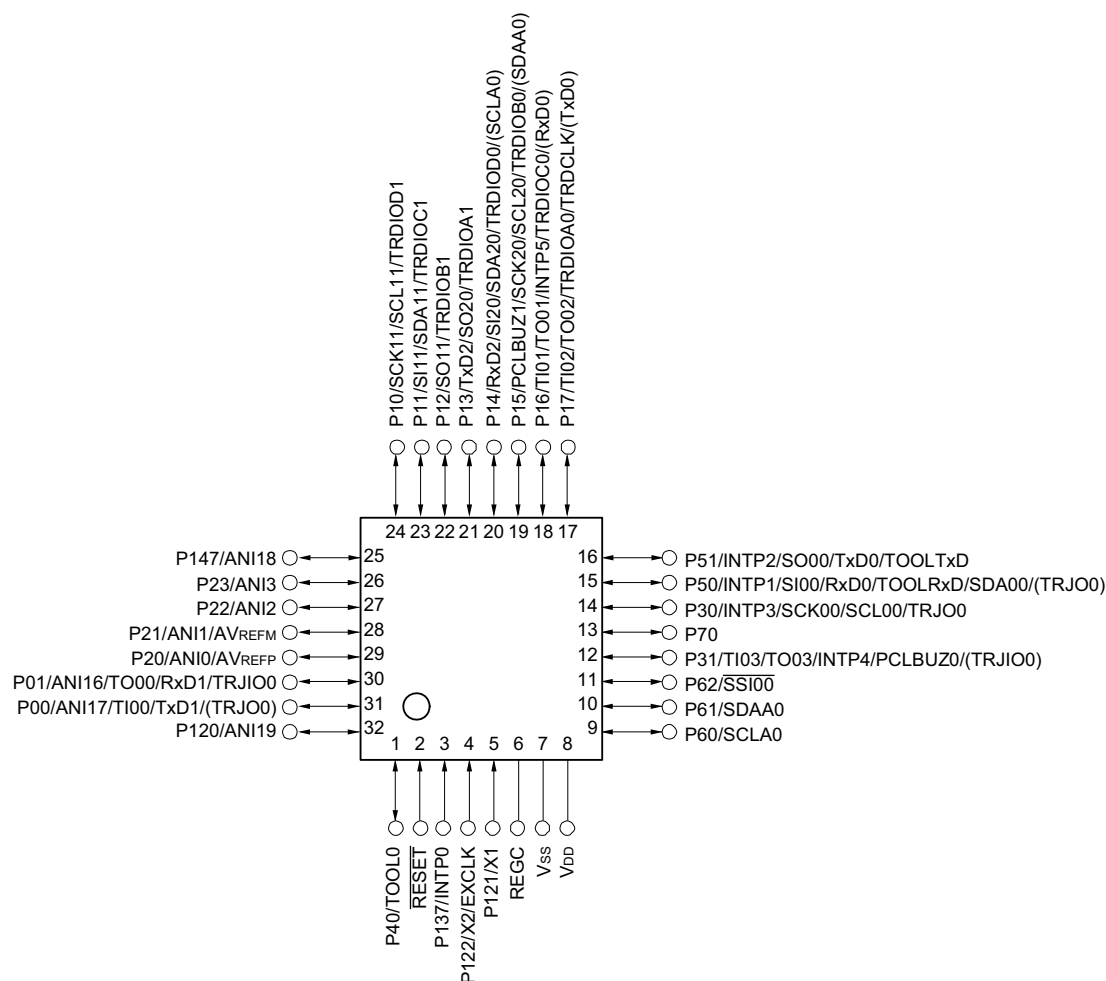
表 1-1 订购产品型号一览表

引脚数	封装	数据闪存	包装规格和环境对应	订购产品型号
32 引脚	32 引脚塑封 LQFP (7×7)	内置	托盘、无铅 (纯 Sn)	R7F0C014B2DFP-C#AA0
64 引脚	64 引脚塑封 LQFP (12×12)			R7F0C014L2DFA-C#AA0
	64 引脚塑封 LQFP (细间距) (10×10)			R7F0C014L2DFB-C#AA0

1.3 引脚连接图 (Top View)

1.3.1 32 引脚产品

- 32 引脚塑封 LQFP (7×7)



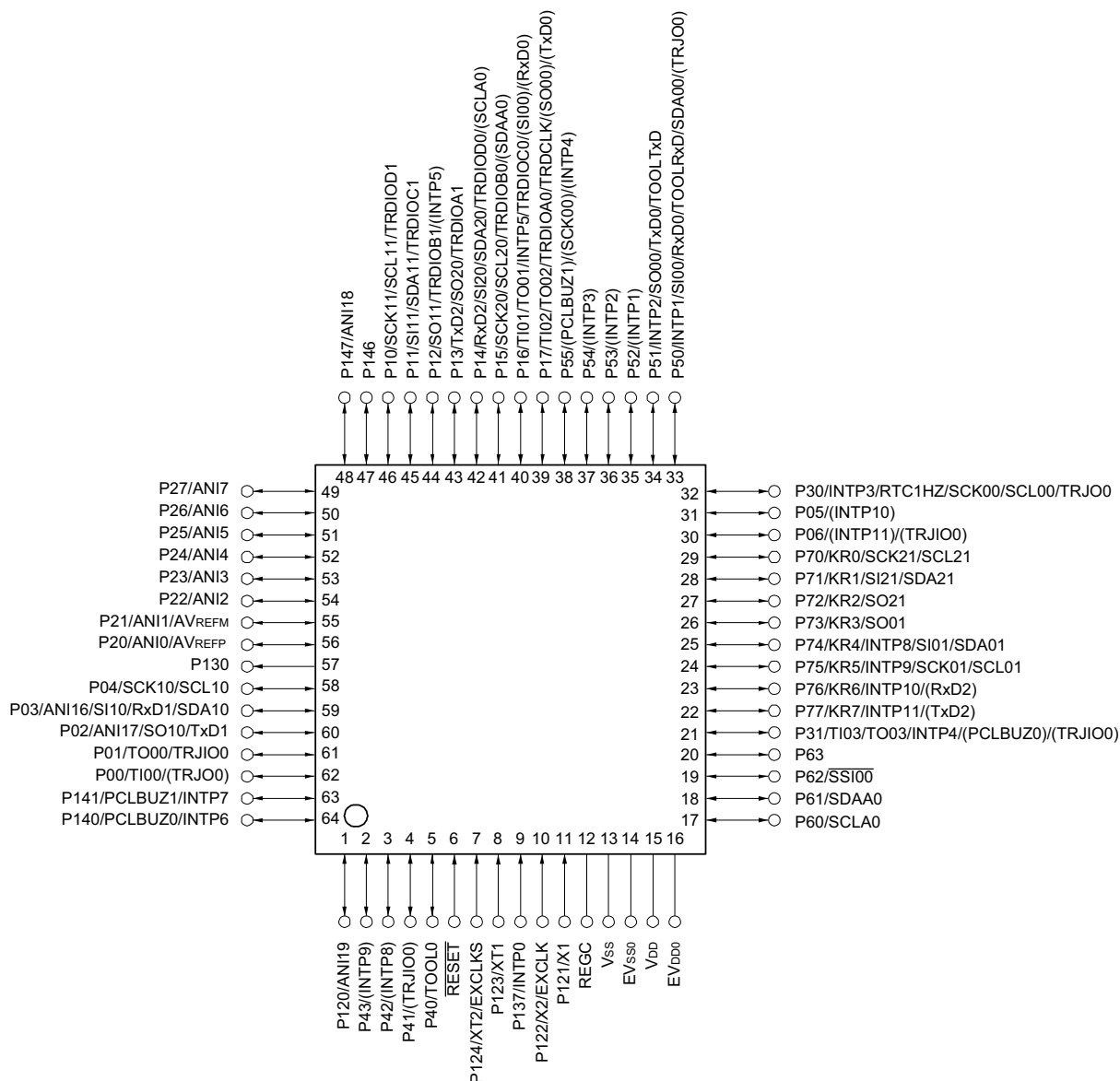
注意 必须通过电容器 (0.47 ~ 1μF) 将 REGC 引脚连接 V_{SS} 引脚。

备注 1. 有关引脚名, 请参照“1.4 引脚名”。

- 能通过设定外围 I/O 重定向寄存器 0、1 (PIOR0、PIOR1), 分配上图 () 内的功能。

1.3.2 64 引脚产品

- 64 引脚塑封 LQFP (12×12)
- 64 引脚塑封 LQFP (细间距) (10×10)



注意 1. EV_{SS0} 引脚和 V_{SS} 引脚必须同电位。

2. V_{DD} 引脚的电压必须大于等于 EV_{DD0} 引脚的电压。
3. 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”。

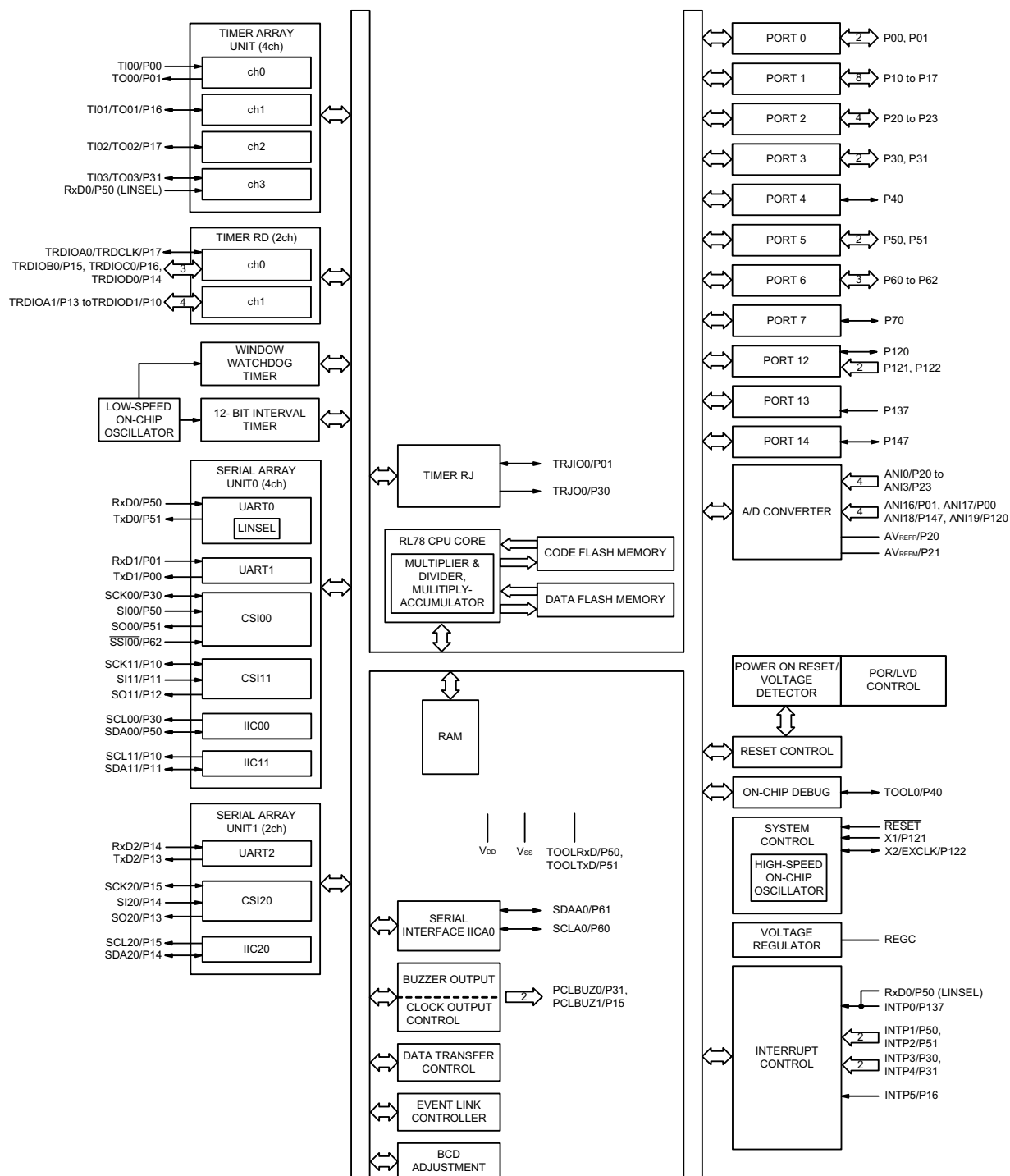
2. 在应用领域中需要降低从单片机内部产生的噪声的情况下, 建议采取单独给 V_{DD} 和 EV_{DD0} 提供电源并且将 V_{SS} 和 EV_{SS0} 单独接地等噪声对策。
3. 能通过设定外围 I/O 重定向寄存器 0、1 (PIOR0、PIOR1), 分配上图 () 内的功能。

1.4 引脚名

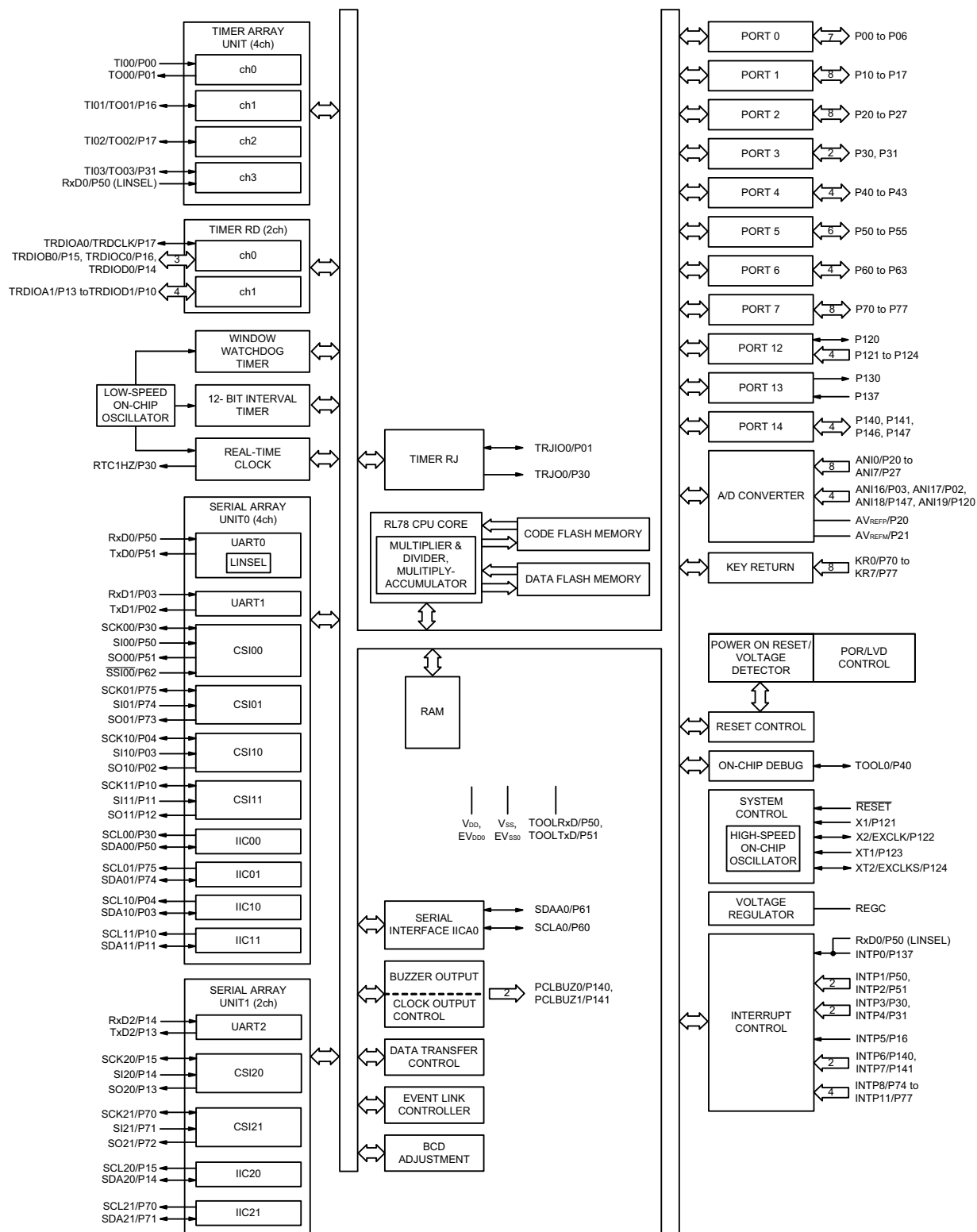
ANI0 ~ ANI7、	: Analog Input	RxD0 ~ RxD2	: Receive Data
ANI16 ~ ANI19		SCK00、SCK01、SCK10、	: Serial Clock Input/Output
AVREFM	: Analog Reference Voltage Minus	SCK11、SCK20、SCK21	
AVREFP	: Analog Reference Voltage Plus	SCLA0	: Serial Clock Input/Output
EVDD0	: Power Supply for Port	SCL00、SCL01、SCL10、	: Serial Clock Output
EVSS0	: Ground for Port	SCL11、SCL20、SCL21	
EXCLK	: External Clock Input (Main System Clock)	SDAA0、SDA00、SDA01、	: Serial Data Input/Output
EXCLKS	: External Clock Input (Subsystem Clock)	SDA10、SDA11、SDA20、	
INTP0 ~ INTP11	: External Interrupt Input	SDA21	
KR0 ~ KR7	: Key Return	SI00、SI01、SI10、SI11、	: Serial Data Input
P00 ~ P06	: Port0	SI20、SI21	
P10 ~ P17	: Port1	SO00、SO01、SO10、	: SerialDataOutput
P20 ~ P27	: Port2	SO11、SO20、SO21	
P30、P31	: Port3	<u>SSI00</u>	: Serial Interface Chip Select Input
P40 ~ P43	: Port4	TI00 ~ TI03	: Timer Input
P50 ~ P55	: Port5	TO00 ~ TO03、TRJ00	: Timer Output
P60 ~ P63	: Port6	TOOL0	: Data Input/Output for Tool
P70 ~ P77	: Port7	TOOLRxD、TOOLTxD	: Data Input/Output for External Device
P120 ~ P124	: Port12	TRDCLK	: Timer External Input Clock
P130、P137	: Port13	TRDIOA0、TRDIOB0、	: Timer Input/Output
P140、P141、P146、P147	: Port14	TRDIOC0、TRDIOD0、	
PCLBUZ0、PCLBUZ1	: Programmable Clock Output/ Buzzer Output	TRDIOA1、TRDIOB1、	
REGC	: Regulator Capacitance	TRDIOC1、TRDIOD1、	
<u>RESET</u>	: Reset	TRJIO0	
RTC1HZ	: Real ~ time Clock Correction Clock (1Hz) Output	TxD0 ~ TxD2	: Transmit Data
		VDD	: Power Supply
		VSS	: Ground
		X1、X2	: Crystal Oscillator (Main System Clock)
		XT1、XT2	: Crystal Oscillator (Subsystem Clock)

1.5 框图

1.5.1 32 引脚产品



1.5.2 64 引脚产品



1.6 功能概要

【32 引脚和 64 引脚的产品】

注意 这是将外围 I/O 重定向寄存器 0、1（PIOR0、PIOR1）置“00H”时的功能概要。

(1/2)

项目		32 引脚	64 引脚
		R7F0C014B2D	R7F0C014L2D
代码闪存		128KB	
数据闪存		8KB	
RAM		8KB 注	
地址空间		1M 字节	
主系统时钟	高速系统时钟	X1（晶体 / 陶瓷）振荡、外部主系统时钟输入（EXCLK） 1 ~ 20MHz: $V_{DD}=2.7 \sim 5.5V$, 1 ~ 16MHz: $V_{DD}=2.4 \sim 2.7V$ 1 ~ 8MHz: $V_{DD}=1.8 \sim 2.7V$, 1 ~ 4MHz: $V_{DD}=1.6 \sim 1.8V$	
	高速内部振荡器时钟 (f_{IH})	HS（高速主）模式: 1 ~ 32MHz ($V_{DD}=2.7 \sim 5.5V$) HS（高速主）模式: 1 ~ 16MHz ($V_{DD}=2.4 \sim 5.5V$) LS（低速主）模式: 1 ~ 8MHz ($V_{DD}=1.8 \sim 5.5V$) LV（低电压主）模式: 1 ~ 4MHz ($V_{DD}=1.6 \sim 5.5V$) 精度 $\pm 2\%$	
副系统时钟		—	XT1（晶体）振荡、外部副系统时钟的输入（EXCLKS）32.768kHz: $V_{DD}=1.6 \sim 5.5V$
低速内部振荡器时钟		15kHz(TYP.): $V_{DD}=1.6 \sim 5.5V$	
通用寄存器		8 位 $\times$ 32 个寄存器（8 位 $\times$ 8 个寄存器 $\times$ 4 组）	
最短指令执行时间		0.03125 μ s（高速内部振荡器时钟: $f_{IH}=32MHz$ 运行时）	
		0.05 μ s（高速系统时钟: $f_{MX}=20MHz$ 运行时）	
		—	30.5 μ s（副系统时钟: $f_{SUB}=32.768kHz$ 运行时）
指令集		• 数据传送（8/16 位） • 加减 / 逻辑运算（8/16 位） • 乘法运算（8 位 $\times$ 8 位、16 位 $\times$ 16 位），除法运算（16 位 $\div$ 16 位、32 位 $\div$ 32 位） • 乘加运算（16 位 $\times$ 16 位 + 32 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	58
	CMOS 输入 / 输出	22	48
	CMOS 输入	3	5
	CMOS 输出	—	1
	N 沟道漏极开路输入 / 输出（6V 耐压）	3	4
定时器	16 位定时器	7 个通道（TAU: 4 个通道，定时器 RJ: 1 个通道，定时器 RD: 2 个通道）	
	看门狗定时器	1 个通道	
	实时时钟（RTC）	—	1 个通道
	12 位间隔定时器	1 个通道	
	定时器输出	定时器输出: 14 个 PWM 输出: 9 个	
	RTC 输出	—	1 个 • 1Hz（副系统时钟: $f_{SUB}=32.768kHz$ ）

注 在使用自编程功能和数据闪存功能时，大约为 7KB。

(2/2)

项目		32 引脚	64 引脚
		R7F0C014B2D	R7F0C014L2D
时钟输出 / 蜂鸣器输出		2 个	2 个
		【32 引脚产品】 • 2.44kHz、4.88kHz、9.76kHz、1.25MHz、2.5MHz、5MHz、10MHz （主系统时钟：f _{MAIN} =20MHz 运行时） 【64 引脚产品】 • 2.44kHz、4.88kHz、9.76kHz、1.25MHz、2.5MHz、5MHz、10MHz （主系统时钟：f _{MAIN} =20MHz 运行时） • 256Hz、512Hz、1.024kHz、2.048kHz、4.096kHz、8.192kHz、16.384kHz、32.768kHz （副系统时钟：f _{SUB} =32.768kHz 运行时）	
8/10 位分辨率 A/D 转换器		8 个通道	12 个通道
串行接口		【32 引脚产品】 • CSI: 1 个通道 / UART（支持 LIN-bus）: 1 个通道 / 简易 I ² C: 1 个通道 • CSI: 1 个通道 / UART: 1 个通道 / 简易 I ² C: 1 个通道 • CSI: 1 个通道 / UART: 1 个通道 / 简易 I ² C: 1 个通道 【64 引脚产品】 • CSI: 2 个通道 / UART（支持 LIN-bus）: 1 个通道 / 简易 I ² C: 2 个通道 • CSI: 2 个通道 / UART: 1 个通道 / 简易 I ² C: 2 个通道 • CSI: 2 个通道 / UART: 1 个通道 / 简易 I ² C: 2 个通道	
	I ² C 总线	1 个通道	1 个通道
数据传送控制器（DTC）		26 个源	29 个源
事件链接控制器（ELC）		事件输入：16 个，事件触发输出：6 个	事件输入：18 个，事件触发输出：6 个
向量 中断源	内部	22	23
	外部	6	13
键中断		—	8
复位		• 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		• 上电复位：1.51±0.03V • 断电复位：1.50±0.03V	
电压检测电路		2.75V ~ 4.06V（6 种）	
片上调试功能		有	
电源电压		V _{DD} =1.6 ~ 5.5V	
工作环境温度		T _A =−40 ~ +85°C	

注 当执行指令码 FFH 时，发生非法指令错误。

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

第2章 引脚功能

2.1 端口功能

引脚的输入 / 输出缓冲器电源因产品而不同。各电源和引脚的关系如下所示。

表 2-1 各引脚的输入 / 输出缓冲器电源

(1) 32 引脚产品

电源	对应的引脚
V_{DD}	全部引脚

(2) 64 引脚产品

电源	对应的引脚
EV_{DD0}	• P20 ~ P27、P121 ~ P124、P137 以外的端口引脚
V_{DD}	• P20 ~ P27、P121 ~ P124、P137 • $\overline{RESET}$ 、REGC

对于复用功能，各端口设定的输入 / 输出、缓冲器和上拉电阻也有效。

2.1.1 32 引脚产品

(1/2)

功能名称	输入 / 输出	复位后	复用功能	功能
P00	输入 / 输出	模拟输入	ANI17/TI00/TxD1/ (TRJO0)	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			ANI16/TO00/RxD1/ TRJIO0	
P10	输入 / 输出	输入端口	SCK11/SCL11/ TRDIOD1	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 和 P14 ~ P17 的输入能设定为 TTL 输入缓冲。 P10、P11、P13 ~ P15 和 P17 的输出能设定为 N 沟道漏极开路输出 (V_{DD} 耐压)。
P11			SI11/SDA11/TRDIOC1	
P12			SO11/TRDIOB1	
P13			TxD2/SO20/TRDIOA1	
P14			RxD2/SI20/SDA20/ TRDIOD0/(SCLA0)	
P15			PCLBUZ1/SCK20/ SCL20/TRDIOB0/ (SDAA0)	
P16			TI01/TO01/INTP5/ TRDIOC0/(RxD0)	
P17			TI02/TO02/TRDIOA0/ TRDCLK/(TxD0)	
P20	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	输入 / 输出	输入端口	INTP3/SCK00/SCL00/ TRJO0	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。 P30 和 P31 的输入能设定为 TTL 输入缓冲。 P30 的输出能设定为 N 沟道漏极开路输出 (V_{DD} 耐压)。
P31			TI03/TO03/INTP4/ PCLBUZ0/(TRJIO0)	
P40	输入 / 输出	输入端口	TOOL0	端口 4 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。

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

2. 必须通过 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 0、1 (PIOR0、PIOR1)，分配上表 () 内的功能。

(2/2)

功能名称	输入 / 输出	复位后	复用功能	功能
P50	输入 / 输出	输入端口	INTP1/SI00/RxD0/ TOOLRxD/SDA00/ (TRJ00)	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 的输入能设定为 TTL 输入缓冲。 P50 和 P51 的输出能设定为 N 沟道漏极开路输出 (V_{DD} 耐压)。
P51			INTP2/SO00/TxD0/ TOOLTxD	
P60	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 (V_{DD} 耐压)。
P61			SDAA0	
P62			SS100	
P70	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	输入 / 输出	模拟输入	ANI19	端口 12 1 位输入 / 输出端口和 2 位输入端口 P120 能设定为模拟输入。 只有 P120 才能指定输入或者输出。 只有 P120 才能通过软件的设定，使用内部上拉电阻。
P121		输入端口	X1	
P122			X2/EXCLK	
P137	输入	输入端口	INTP0	端口 13 1 位输入专用端口
P147	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	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 必须通过端口模式控制寄存器 x (PMCx) 将各引脚设定为数字或者模拟 (能以位为单位进行设定)。

备注 能通过设定外围 I/O 重定向寄存器 0、1 (PIOR0、PIOR1)，分配上表 () 内的功能。

2.1.2 64 引脚产品

(1/2)

功能名称	输入 / 输出	复位后	复用功能	功能
P00	输入 / 输出	输入端口	TI00/(TRJO0)	端口 0 7 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、P03 和 P04 的输入能设定为 TTL 输入缓冲。 P00 和 P02 ~ P04 的输出能设定为 N 沟道漏极开路输出（EV _{DD} 耐压）。 P02 和 P03 能设定为模拟输入注 1。
P01			TO00/TRJIO0	
P02		模拟输入	ANI17/SO10/TxD1	
P03			ANI16/SI10/RxD1/SDA10	
P04		输入端口	SCK10/SCL10	
P05			(INTP10)	
P06			(INTP11)/(TRJIO0)	
P10	输入 / 输出	输入端口	SCK11/SCL11/TRDIOD1	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 和 P14 ~ P17 的输入能设定为 TTL 输入缓冲。 P10、P11、P13 ~ P15 和 P17 的输出能设定为 N 沟道漏极开路输出（EV _{DD} 耐压）。
P11			SI11/SDA11/TRDIOC1	
P12			SO11/TRDIOB1/(INTP5)	
P13			TxD2/SO20/TRDIOA1	
P14			RxD2/SI20/SDA20/TRDIOD0/(SCLA0)	
P15			SCK20/SCL20/TRDIOB0/(SDAA0)	
P16			TI01/TO01/INTP5/TRDIOC0/(SI00)/(RxD0)	
P17			TI02/TO02/TRDIOA0/TRDCLK/(SO00)/(TxD0)	
P20	输入 / 输出	模拟输入	ANI0/AV _{REFP}	端口 2 8 位输入 / 输出端口 能以位为单位指定输入或者输出。 能设定为模拟输入注 2。
P21			ANI1/AV _{REFM}	
P22			ANI2	
P23			ANI3	
P24			ANI4	
P25			ANI5	
P26			ANI6	
P27			ANI7	
P30	输入 / 输出	输入端口	INTP3/RTC1HZ/SCK00/SCL00/TRJO0	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。 P30 和 P31 的输入能设定为 TTL 输入缓冲。 P30 的输出能设定为 N 沟道漏极开路输出（EV _{DD} 耐压）。
P31			TI03/TO03/INTP4/(PCLBUZ0)/(TRJIO0)	

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

2. 必须通过 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 0、1（PIOR0、PIOR1），分配上表（）内的功能。

(2/2)

功能名称	输入 / 输出	复位后	复用功能	功能
P40	输入 / 输出	输入端口	TOOL0	端口 4
P41			(TRJ00)	4 位输入 / 输出端口
P42			(INTP8)	能以位为单位指定输入或者输出。
P43			(INTP9)	输入端口能通过软件的设定, 使用内部上拉电阻。
P50	输入 / 输出	输入端口	INTP1/SI00/RxD0/ TOOLRxD/SDA00/ (TRJ00)	端口 5
P51			INTP2/SO00/TxD0/ TOOLTxD	6 位输入 / 输出端口
P52			(INTP1)	能以位为单位指定输入或者输出。
P53			(INTP2)	输入端口能通过软件的设定, 使用内部上拉电阻。
P54			(INTP3)	P50 和 P55 的输入能设定为 TTL 输入缓冲。
P55			(PCLBUZ1)/(SCK00)/ (INTP4)	P50、P51 和 P55 的输出能设定为 N 沟道漏极开路输出 (EV _{DD} 耐压)。
P60	输入 / 输出	输入端口	SCLA0	端口 6
P61			SDAA0	4 位输入 / 输出端口
P62			SSI00	能以位为单位指定输入或者输出。
P63			—	P60 ~ P63 的输出为 N 沟道漏极开路输出 (6V 耐压)。
P70	输入 / 输出	输入端口	KR0/SCK21/SCL21	端口 7
P71			KR1/SI21/SDA21	8 位输入 / 输出端口
P72			KR2/SO21	能以位为单位指定输入或者输出。
P73			KR3/SO01	输入端口能通过软件的设定, 使用内部上拉电阻。
P74			KR4/INTP8/SI01/ SDA01	P71 和 P74 的输出能设定为 N 沟道漏极开路输出 (EV _{DD} 耐压)。
P75			KR5/INTP9/SCK01/ SCL01	
P76			KR6/INTP10/(RxD2)	
P77			KR7/INTP11/(TxD2)	
P120	输入 / 输出	模拟输入	ANI19	端口 12
P121	输入	输入端口	X1	1 位输入 / 输出端口和 4 位输入专用端口
P122			X2/EXCLK	P120 能设定为模拟输入。
P123			XT1	只有 P120 才能指定输入或者输出。
P124			XT2/EXCLKS	只有 P120 才能通过软件的设定, 使用内部上拉电阻。
P130	输出	输出端口	—	端口 13
P137	输入	输入端口	INTP0	1 位输出专用端口和 1 位输入专用端口
P140	输入 / 输出	输入端口	PCLBUZ0/INTP6	端口 14
P141			PCLBUZ1/INTP7	4 位输入 / 输出端口
P146			—	能以位为单位指定输入或者输出。
P147		模拟输入	ANI18	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	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 必须通过端口模式控制寄存器 x (PMCx) 将各引脚设定为数字或者模拟 (能以位为单位进行设定)。

备注 能通过设定外围 I/O 重定向寄存器 0、1 (PIOR0、PIOR1), 分配上表 () 内的功能。

2.2 端口以外的功能

2.2.1 各产品配置的功能

(1/3)

功能名称	64 引脚	32 引脚
ANI0	○	○
ANI1	○	○
ANI2	○	○
ANI3	○	○
ANI4	○	—
ANI5	○	—
ANI6	○	—
ANI7	○	—
ANI16	○	○
ANI17	○	○
ANI18	○	○
ANI19	○	○
INTP0	○	○
INTP1	○	○
INTP2	○	○
INTP3	○	○
INTP4	○	○
INTP5	○	○
INTP6	○	—
INTP7	○	—
INTP8	○	—
INTP9	○	—
INTP10	○	—
INTP11	○	—
KR0	○	—
KR1	○	—
KR2	○	—
KR3	○	—
KR4	○	—
KR5	○	—
KR6	○	—
KR7	○	—
PCLBUZ0	○	○
PCLBUZ1	○	○
REGC	○	○
RTC1HZ	○	—
RESET	○	○
RxD0	○	○
RxD1	○	○
RxD2	○	○

(2/3)

功能名称	64 引脚	32 引脚
SCK00	○	○
SCK01	○	—
SCK10	○	—
SCK11	○	○
SCK20	○	○
SCK21	○	—
SCLA0	○	○
SCL00	○	○
SCL01	○	—
SCL10	○	—
SCL11	○	○
SCL20	○	○
SCL21	○	—
SDAA0	○	○
SDA00	○	○
SDA01	○	—
SDA10	○	—
SDA11	○	○
SDA20	○	○
SDA21	○	—
SI00	○	○
SI01	○	—
SI10	○	—
SI11	○	○
SI20	○	○
SI21	○	—
SO00	○	○
SO01	○	—
SO10	○	—
SO11	○	○
SO20	○	○
SO21	○	—
SSI00	○	○

(3/3)

功能名称	64 引脚	32 引脚
TI00	○	○
TI01	○	○
TI02	○	○
TI03	○	○
TO00	○	○
TO01	○	○
TO02	○	○
TO03	○	○
TRJIO0	○	○
TRJO0	○	○
TRDCLK	○	○
TRDIOA0	○	○
TRDIOB0	○	○
TRDI0C0	○	○
TRDI0D0	○	○
TRDIOA1	○	○
TRDIOB1	○	○
TRDI0C1	○	○
TRDI0D1	○	○
TxD0	○	○
TxD1	○	○
TxD2	○	○
X1	○	○
X2	○	○
EXCLK	○	○
EXCLKS	○	—
XT1	○	—
XT2	○	—
V _{DD}	○	○
EV _{DD0}	○	—
AV _{REFP}	○	○
AV _{REFM}	○	○
V _{SS}	○	○
EV _{SS0}	○	—
TOOLRxD	○	○
TOOLTxD	○	○
TOOL0	○	○

2.2.2 功能说明

(1/2)

功能名称	输入 / 输出	功能
ANI0 ~ ANI7、ANI16 ~ ANI19	输入	A/D 转换器的模拟输入（参照“图 13-44 模拟输入引脚的处理”）
INTP0 ~ INTP11	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
KR0 ~ KR7	输入	键中断输入
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 ~ 1μF）连接 V _{SS} 。 为了稳定内部电压，必须使用特性好的电容器。
RTC1HZ	输出	实时时钟的校正时钟（1Hz）输出
RESET	输入	低电平有效的系统复位输入 当不使用外部复位时，必须直接或者通过电阻连接 V _{DD} 。
RxD0 ~ RxD2	输入	串行接口 UART0、UART1 和 UART2 的串行数据输入
TxD0 ~ TxD2	输出	串行接口 UART0、UART1 和 UART2 的串行数据输出
SCK00、SCK01、SCK10、SCK11、SCK20、SCK21	输入 / 输出	串行接口 CSI00、CSI01、CSI10、CSI11、CSI20、CSI21 的串行时钟输入 / 输出
SCL00、SCL01、SCL10、SCL11、SCL20、SCL21	输出	串行接口 IIC00、IIC01、IIC10、IIC11、IIC20、IIC21 的串行时钟输出
SDA00、SDA01、SDA10、SDA11、SDA20、SDA21	输入 / 输出	串行接口 IIC00、IIC01、IIC10、IIC11、IIC20、IIC21 的串行数据输入 / 输出
SI00、SI01、SI10、SI11、SI20、SI21	输入	串行接口 CSI00、CSI01、CSI10、CSI11、CSI20、CSI21 的串行数据输入
SSI00	输入	串行接口 CSI00 的芯片选择输入
SO00、SO01、SO10、SO11、SO20、SO21	输出	CSI00、CSI01、CSI10、CSI11、CSI20、CSI21 的串行数据输出
SCLA0	输入 / 输出	串行接口 IICA0 的时钟输入 / 输出
SDAA0	输入 / 输出	串行接口 IICA0 的串行数据输入 / 输出
TI00 ~ TI03	输入	16 位定时器 00 ~ 03 的外部计数时钟 / 捕捉触发输入
TO00 ~ TO03	输出	16 位定时器 00 ~ 03 的定时器输出
TRJIO0	输入 / 输出	定时器 RJ 的输入 / 输出
TRJO0	输出	定时器 RJ 的输出
TRDCLK	输入	定时器 RD 的外部时钟输入
TRDIOA0、TRDIOB0、TRDIOC0、TRDIOD0、TRDIOA1、TRDIOB1、TRDIOC1、TRDIOD1	输入 / 输出	定时器 RD 的输入 / 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

(2/2)

功能名称	输入 / 输出	功能
XT1、XT2	—	连接用于副系统时钟的谐振器。
EXCLKS	输入	副系统时钟的外部时钟输入
V _{DD}	—	< 32 引脚产品的情况 > 全部引脚的正电源 < 64 引脚产品的情况 > P20 ~ P27、P121 ~ P124、P137 和端口以外引脚的正电源
EV _{DD0}	—	端口引脚 (P20 ~ P27、P121 ~ P124 和 P137) 的正电源
AV _{REFP}	输入	A/D 转换器的基准电压 (+) 输入
AV _{REFM}	输入	A/D 转换器的基准电压 (-) 输入
V _{SS}	—	< 32 引脚产品的情况 > 全部引脚的接地电位 < 64 引脚产品的情况 > P20 ~ P27、P121 ~ P124、P137 和端口以外引脚的接地电位
EV _{SS0}	—	端口引脚 (P20 ~ P27、P121 ~ P124 和 P137 除外) 的接地电位
TOOLRxD	输入	用于在闪存编程时连接外部设备的 UART 串行数据接收
TOOLTxD	输出	用于在闪存编程时连接外部设备的 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所示：

表 2-2 解除复位时的 P40/TOOL0 和运行模式的关系

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

详细内容请参照“27.4 编程方法”。

备注 作为噪声和锁定的对策，必须在 V_{DD}-V_{SS}、EV_{DD0}-EV_{SS0} 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 2-3 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（Top View）”和“2.1 端口功能”。

表 2-3 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 ~ P06	输入 / 输出	输入：必须单独通过电阻连接 EV_{DD0} 或者 EV_{SS0} 。
P10 ~ P17		输出：必须置为开路。
P20 ~ P27		输入：必须单独通过电阻连接 V_{DD} 或者 V_{SS} 。
P30、P31		输出：必须置为开路。
P40/TOOL0		输入：必须单独通过电阻连接 EV_{DD0} 或者置为开路。
P41 ~ P43		输出：必须置为开路。
P50 ~ P55		输入：必须单独通过电阻连接 EV_{DD0} 或者 EV_{SS0} 。
P60 ~ P63		输出：必须置为开路。
P70 ~ P77		输入：必须单独通过电阻连接 EV_{DD0} 或者 EV_{SS0} 。
P120		输出：必须置为开路。
P121 ~ P124	输入	必须单独通过电阻连接 V_{DD} 或者 V_{SS} 。
P130	输出	必须置为开路。
P137	输入	必须单独通过电阻连接 V_{DD} 或者 V_{SS} 。
P140、P141、 P146、P147	输入 / 输出	输入：必须单独通过电阻连接 EV_{DD0} 或者 EV_{SS0} 。
		输出：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} 。

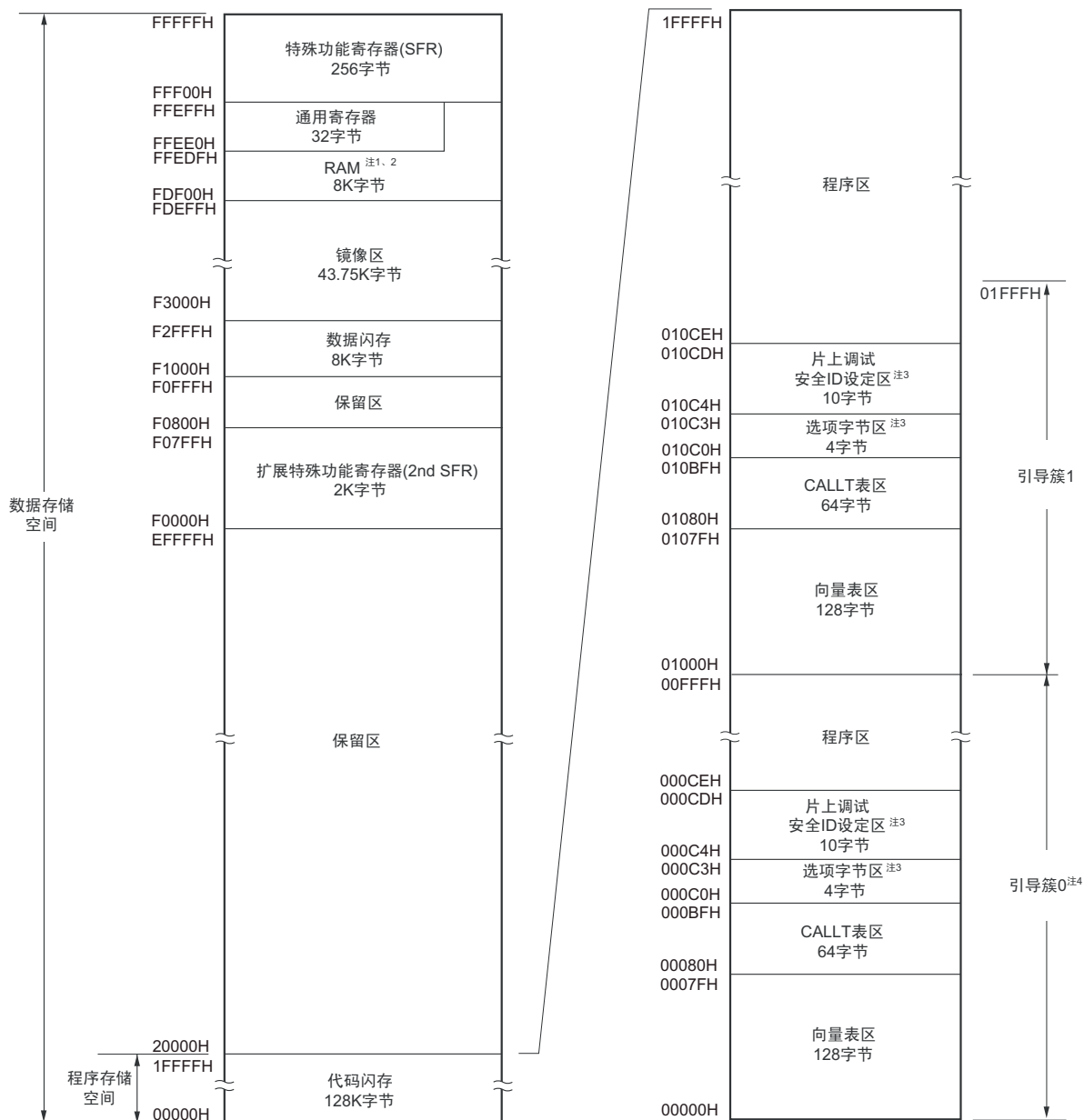
备注 对于没有 EV_{DD0} 引脚和 EV_{SS0} 引脚的产品，必须将 EV_{DD0} 替换为 V_{DD} 并且将 EV_{SS0} 替换为 V_{SS} 。

第 3 章 CPU 体系结构

3.1 存储空间

R7F0C014B2D、R7F0C014L2D 能存取 1M 字节的地址空间。存储器映像如图 3-1 所示。

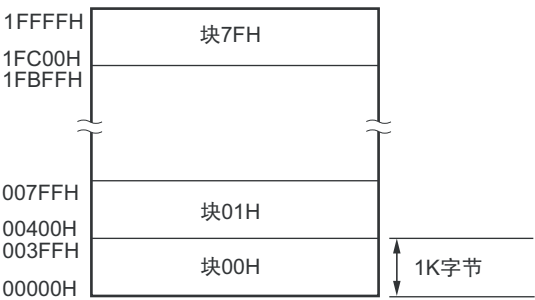
图 3-1 存储器映像



- 注 1. 当进行自编程以及改写数据闪存时，不能将堆栈数据缓冲器、向量中断处理的转移目标和 DTC 传送目标 / 传送源使用的 RAM 地址分配到 FFE20H ~ FFEDEFH 的区域。另外，FDF00H ~ FE309H 的区域用于各库，因此禁止使用此区域。
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“27.7 安全设定”）。

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“24.3.3 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”。



闪存地址值和块号的对应如下所示。

表 3-1 闪存地址值和块号的对应

地址值	块号	地址值	块号	地址值	块号	地址值	块号
00000H ~ 003FFH	00H	08000H ~ 083FFH	20H	10000H ~ 103FFH	40H	18000H ~ 183FFH	60H
00400H ~ 007FFH	01H	08400H ~ 087FFH	21H	10400H ~ 107FFH	41H	18400H ~ 187FFH	61H
00800H ~ 00BFFH	02H	08800H ~ 08BFFH	22H	10800H ~ 10BFFH	42H	18800H ~ 18BFFH	62H
00C00H ~ 00FFFH	03H	08C00H ~ 08FFFH	23H	10C00H ~ 10FFFH	43H	18C00H ~ 18FFFH	63H
01000H ~ 013FFH	04H	09000H ~ 093FFH	24H	11000H ~ 113FFH	44H	19000H ~ 193FFH	64H
01400H ~ 017FFH	05H	09400H ~ 097FFH	25H	11400H ~ 117FFH	45H	19400H ~ 197FFH	65H
01800H ~ 01BFFH	06H	09800H ~ 09BFFH	26H	11800H ~ 11BFFH	46H	19800H ~ 19BFFH	66H
01C00H ~ 01FFFH	07H	09C00H ~ 09FFFH	27H	11C00H ~ 11FFFH	47H	19C00H ~ 19FFFH	67H
02000H ~ 023FFH	08H	0A000H ~ 0A3FFH	28H	12000H ~ 123FFH	48H	1A000H ~ 1A3FFH	68H
02400H ~ 027FFH	09H	0A400H ~ 0A7FFH	29H	12400H ~ 127FFH	49H	1A400H ~ 1A7FFH	69H
02800H ~ 02BFFH	0AH	0A800H ~ 0ABFFH	2AH	12800H ~ 12BFFH	4AH	1A800H ~ 1ABFFH	6AH
02C00H ~ 02FFFH	0BH	0AC00H ~ 0AFFFH	2BH	12C00H ~ 12FFFH	4BH	1AC00H ~ 1AFFFH	6BH
03000H ~ 033FFH	0CH	0B000H ~ 0B3FFH	2CH	13000H ~ 133FFH	4CH	1B000H ~ 1B3FFH	6CH
03400H ~ 037FFH	0DH	0B400H ~ 0B7FFH	2DH	13400H ~ 137FFH	4DH	1B400H ~ 1B7FFH	6DH
03800H ~ 03BFFH	0EH	0B800H ~ 0BBFFH	2EH	13800H ~ 13BFFH	4EH	1B800H ~ 1BBFFH	6EH
03C00H ~ 03FFFH	0FH	0BC00H ~ 0BFFFH	2FH	13C00H ~ 13FFFH	4FH	1BC00H ~ 1BFFFH	6FH
04000H ~ 043FFH	10H	0C000H ~ 0C3FFH	30H	14000H ~ 143FFH	50H	1C000H ~ 1C3FFH	70H
04400H ~ 047FFH	11H	0C400H ~ 0C7FFH	31H	14400H ~ 147FFH	51H	1C400H ~ 1C7FFH	71H
04800H ~ 04BFFH	12H	0C800H ~ 0CBFFH	32H	14800H ~ 14BFFH	52H	1C800H ~ 1CBFFH	72H
04C00H ~ 04FFFH	13H	0CC00H ~ 0CFFFH	33H	14C00H ~ 14FFFH	53H	1CC00H ~ 1CFFFH	73H
05000H ~ 053FFH	14H	0D000H ~ 0D3FFH	34H	15000H ~ 153FFH	54H	1D000H ~ 1D3FFH	74H
05400H ~ 057FFH	15H	0D400H ~ 0D7FFH	35H	15400H ~ 157FFH	55H	1D400H ~ 1D7FFH	75H
05800H ~ 05BFFH	16H	0D800H ~ 0DBFFH	36H	15800H ~ 15BFFH	56H	1D800H ~ 1DBFFH	76H
05C00H ~ 05FFFH	17H	0DC00H ~ 0DFFFH	37H	15C00H ~ 15FFFH	57H	1DC00H ~ 1DFFFH	77H
06000H ~ 063FFH	18H	0E000H ~ 0E3FFH	38H	16000H ~ 163FFH	58H	1E000H ~ 1E3FFH	78H
06400H ~ 067FFH	19H	0E400H ~ 0E7FFH	39H	16400H ~ 167FFH	59H	1E400H ~ 1E7FFH	79H
06800H ~ 06BFFH	1AH	0E800H ~ 0EBFFH	3AH	16800H ~ 16BFFH	5AH	1E800H ~ 1EBFFH	7AH
06C00H ~ 06FFFH	1BH	0EC00H ~ 0EFFFH	3BH	16C00H ~ 16FFFH	5BH	1EC00H ~ 1EFFFH	7BH
07000H ~ 073FFH	1CH	0F000H ~ 0F3FFH	3CH	17000H ~ 173FFH	5CH	1F000H ~ 1F3FFH	7CH
07400H ~ 077FFH	1DH	0F400H ~ 0F7FFH	3DH	17400H ~ 177FFH	5DH	1F400H ~ 1F7FFH	7DH
07800H ~ 07BFFH	1EH	0F800H ~ 0FBFFH	3EH	17800H ~ 17BFFH	5EH	1F800H ~ 1FBFFH	7EH
07C00H ~ 07FFFH	1FH	0FC00H ~ 0FFFFH	3FH	17C00H ~ 17FFFH	5FH	1FC00H ~ 1FFFFH	7FH

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，R7F0C014B2D、R7F0C014L2D 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
R7F0C014B2D、R7F0C014L2D	闪存	131072×8 位（00000H ~ 1FFFFH）

内部程序的存储空间分为以下区域。

(1) 向量表区

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

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

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

向量表如表 3-3 所示。“○”表示支持的中断源，“—”表示不支持的中断源。

表 3-3 向量表 (1/2)

向量表地址	中断源	64 引脚	32 引脚
0000H	RESET、POR、LVD、WDT、TRAP、IAW、RPE	○	○
0004H	INTWDTI	○	○
0006H	INTLVI	○	○
0008H	INTP0	○	○
000AH	INTP1	○	○
000CH	INTP2	○	○
000EH	INTP3	○	○
0010H	INTP4	○	○
0012H	INTP5	○	○
0014H	INTST2/INTCSI20/INTIIC20	○	○
0016H	INTSR2/INTCSI21/INTIIC21	○	注 1
0018H	INTSRE2	○	○
001EH	INTST0/INTCSI00/INTIIC00	○	○
0020H	INTSR0/INTCSI01/INTIIC01	○	注 2
0022H	INTSRE0	○	○
	INTTM01H	○	○
0024H	INTST1/INTCSI10/INTIIC10	○	注 3
0026H	INTSR1/INTCSI11/INTIIC11	○	○
0028H	INTSRE1	○	○
	INTTM03H	○	○
002AH	INTIICA0	○	○
002CH	INTTM00	○	○
002EH	INTTM01	○	○
0030H	INTTM02	○	○
0032H	INTTM03	○	○
0034H	INTAD	○	○
0036H	INTRTC	○	○
0038H	INTIT	○	○
003AH	INTKR	○	—
0040H	INTTRJ0	○	○
004AH	INTP6	○	—
004CH	INTP7	○	—
004EH	INTP8	○	—
0050H	INTP9	○	—
0052H	INTP10	○	—
0054H	INTP11	○	—
0056H	INTTRD0	○	○

注 1. 只支持 INTSR2。

2. 只支持 INTSR0。

3. 只支持 INTST1。

表 3-3 向量表 (2/2)

向量表地址	中断源	64 引脚	32 引脚
0058H	INTTRD1	○	○
0062H	INTFL	○	○
007EH	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。必须给予程序入口地址设定 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 26 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 28 章 片上调试功能”。

3.1.2 镜像区

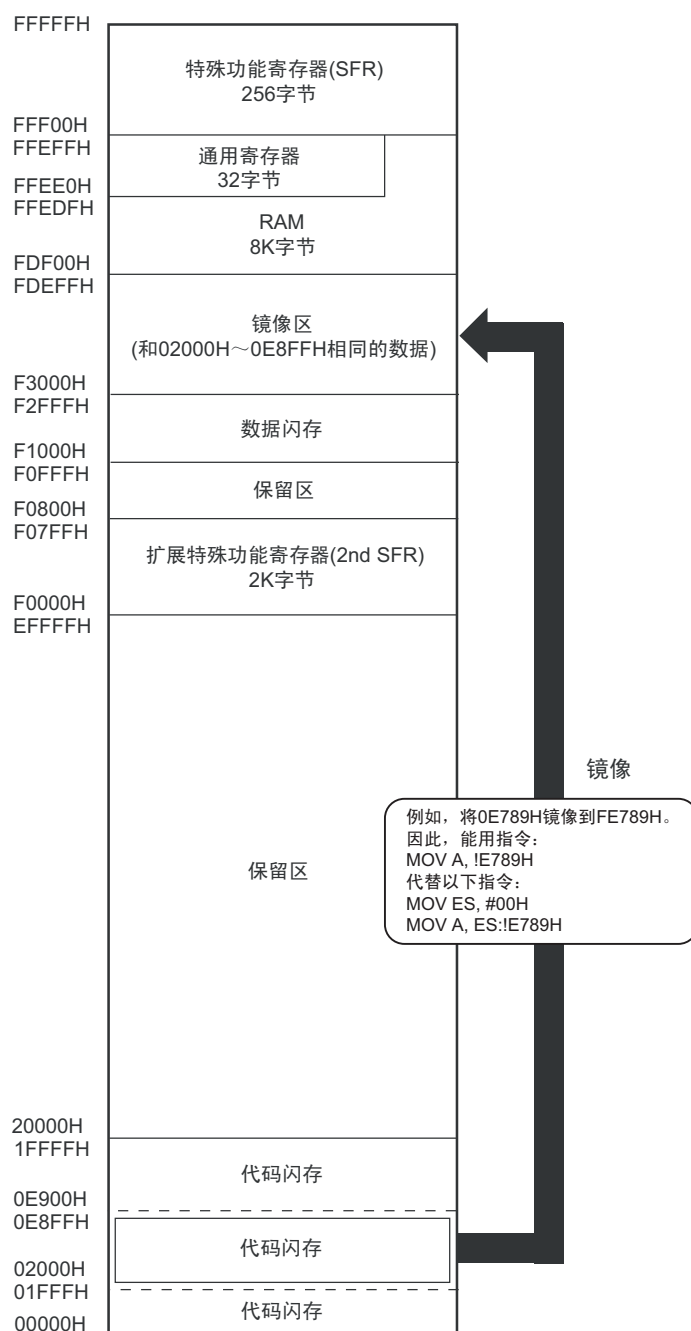
R7F0C014B2D、R7F0C014L2D 将 00000H ~ 0FFFFH 或者 10000H ~ 1FFFFH 的代码闪存区镜像到 F0000H ~ FFFFFH（通过处理器模式控制寄存器（PMC）进行设定）。

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特殊功能寄存器（SFR）、扩展特殊功能寄存器（2nd SFR）、RAM 区、数据闪存区以及保留区。

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。



PMC 寄存器的说明如下。

- 处理器模式控制寄存器（PMC）
这是设定要镜像到F0000H～FFFFFH的闪存空间的寄存器。
通过1位或者8位存储器操作指令设定PMC寄存器。
在产生复位信号后，此寄存器的值变为“00H”。

图 3-2 处理器模式控制寄存器（PMC）的格式

地址: FFFFEH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PMC	0	0	0	0	0	0	0	MAA

MAA	要镜像到 F0000H ~ FFFFFH 的闪存空间的设定
0	将 00000H ~ 0FFFFH 镜像到 F0000H ~ FFFFFH。
1	将 10000H ~ 1FFFFH 镜像到 F0000H ~ FFFFFH。

注意 在设定 PMC 寄存器后，必须至少在等待 1 条指令后存取镜像区。

3.1.3 内部数据存储空间

R7F0C014B2D、R7F0C014L2D 内置以下 RAM。

表 3-4 内部 RAM 容量

产品	内部 RAM
R7F0C014B2D、R7F0C014L2D	8192×8 位（FDF00H ~ 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. 当进行自编程以及改写数据闪存时，不能将堆栈数据缓冲器、向量中断处理的转移目标和 DTC 传送目标 / 传送源使用的 RAM 地址分配到 FFE20H ~ FFEDFH 的区域。

3. 当进行自编程以及改写数据闪存时，以下产品的内部 RAM 区用于各库，因此禁止使用：

R7F0C014B2D、R7F0C014L2D : FDF00H ~ FE309H

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）。

注意 1. 不能存取未分配扩展 SFR 的地址。

2. 当存取被分配在扩展 SFR（2nd SFR）的部分区域 F0500H 中的定时器 RJ 的计数寄存器 0（TRJ0）时，CPU 不进入下一条指令的处理而处于 CPU 处理的等待状态。因此，当发生此等待时，指令执行的时钟数增加等待的时钟数。存取定时器 RJ 的计数寄存器 0（TRJ0）时的读写等待时钟数都为 1 个时钟。

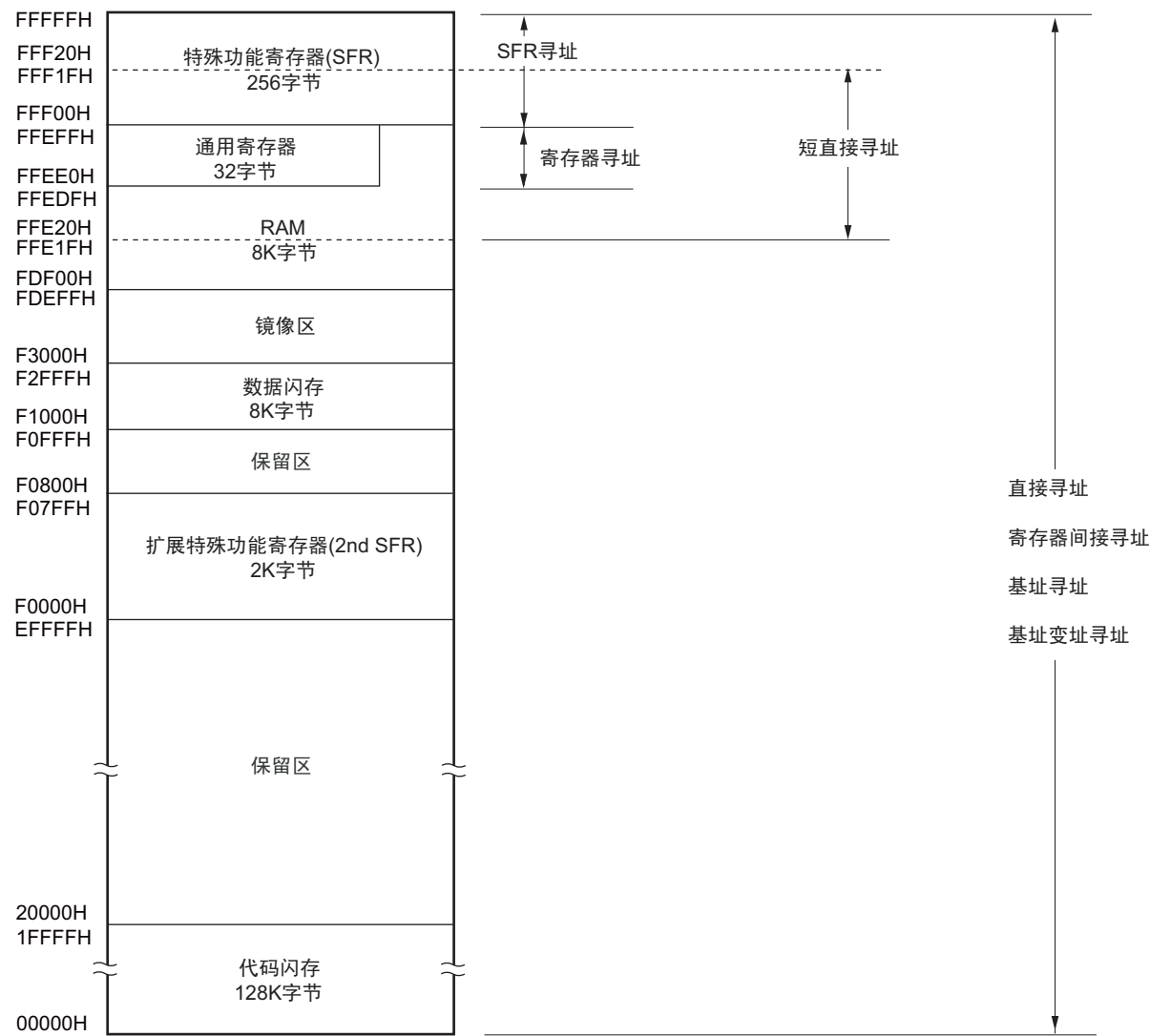
3.1.6 数据存储器的寻址

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，R7F0C014B2D、R7F0C014L2D 提供了丰富的寻址方式。尤其是对于内置数据存储器的区域，能根据特殊功能寄存器（SFR）和通用寄存器等各种功能进行特殊的寻址。数据存储器和寻址的对应如图 3-3 所示。

有关各寻址的详细内容，请参照“3.4 处理数据地址的寻址”。

图 3-3 数据存储器和寻址的对应



3.2 处理器的寄存器

R7F0C014B2D、R7F0C014L2D 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）。

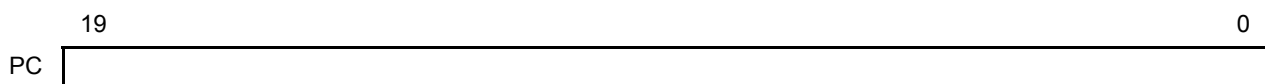
(1) 程序计数器（PC）

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，给程序计数器设定地址 0000H 和 0001H 的复位向量表的值。

图 3-4 程序计数器的结构



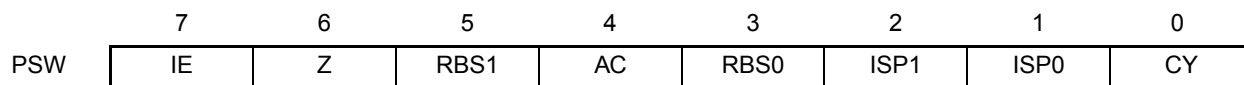
(2) 程序状态字（PSW）

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

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

在产生复位信号后，PSW 的值变为“06H”。

图 3-5 程序状态字的结构



(a) 中断允许标志（IE）

这是控制 CPU 的中断请求接受运行的标志。

当 IE 位是“0”时，为中断禁止（DI）状态，禁止全部可屏蔽中断。

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

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）

当运算结果为零或者相等时，将此标志置“1”。否则，将此标志清“0”。

(c) 寄存器组选择标志 (RBS0、RBS1)

这是从 4 组寄存器中选择 1 组的 2 位标志。

此标志保存通过执行 SEL RBn 指令所选寄存器组的 2 位信息。

(d) 辅助进位标志 (AC)

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

(e) 服务优先级标志 (ISP1、ISP0)

这是管理能接受的可屏蔽向量中断优先级的标志。禁止接受优先级指定标志寄存器 (PRn0L、PRn0H、PRn1L、PRn1H、PRn2L、PRn2H) (参照 18.3.3) 指定的低于 ISP0 标志值和 ISP1 标志值的向量中断请求。另外，实际上根据中断允许标志 (IE) 的状态控制是否接受向量中断请求。

备注 n=0、1

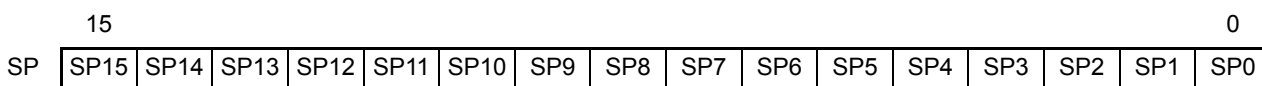
(f) 进位标志 (CY)

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

(3) 堆栈指针 (SP)

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

图 3-6 堆栈指针的结构



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

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

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

3. 当进行自编程以及改写数据闪存时，不能将堆栈数据缓冲器、向量中断处理的转移目标和 DTC 传送目标 / 传送源使用的 RAM 地址分配到 FFE20H ~ FFEDFH 的区域。

4. 当进行自编程以及改写数据闪存时，以下产品的内部 RAM 区用于各库，因此禁止使用：

R7F0C014B2D、R7F0C014L2D : FDF00H ~ FE309H

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。

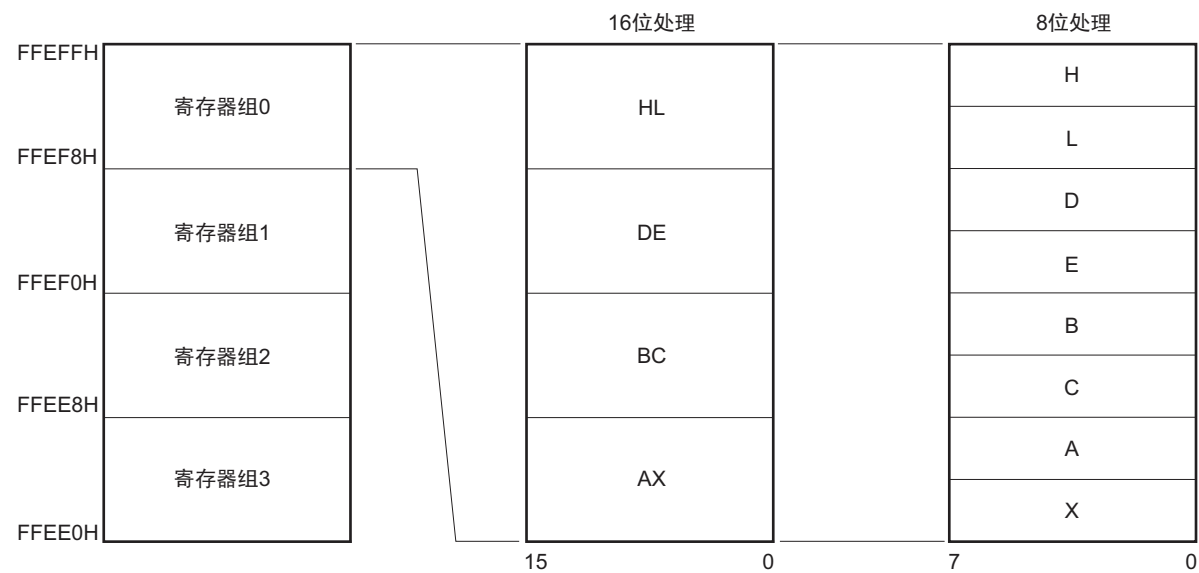
各寄存器除了能分别用作 8 位寄存器以外，还能将 2 个 8 位寄存器成对用作 1 个 16 位寄存器（AX、BC、DE、HL）。

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 3-7 通用寄存器的结构

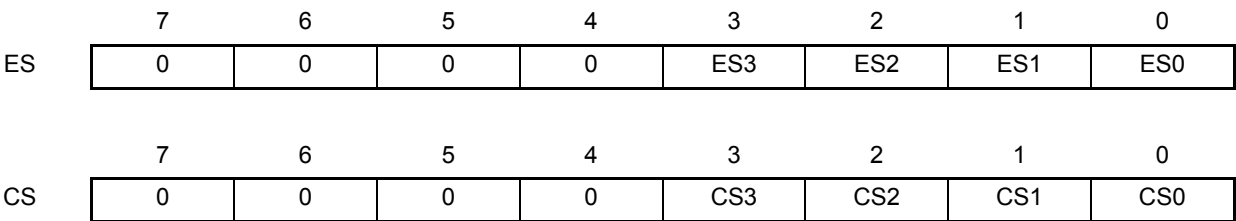
(a) 功能名称



3.2.3 ES 寄存器和 CS 寄存器

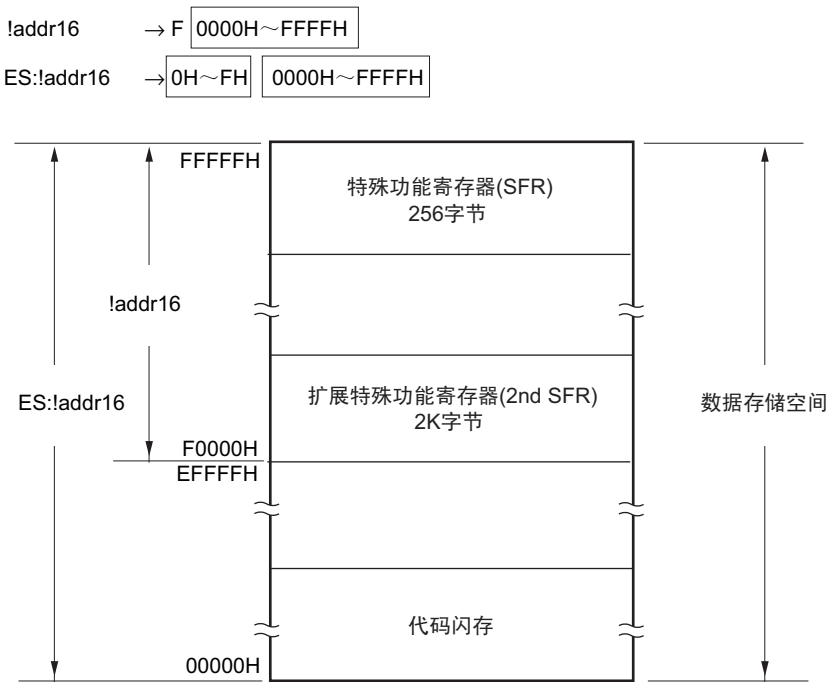
能通过 ES 寄存器和 CS 寄存器（寄存器直接寻址）分别指定存取数据和执行转移指令时的高位地址。
ES 寄存器的复位后的初始值为“0FH”，CS 寄存器的复位后的初始值为“00H”。

图 3-8 ES/CS 寄存器的结构



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

图 3-9 数据存取区的扩展



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

SFR 是和通用寄存器不同的并且分别具有特殊功能的寄存器。

SFR 空间分配在 FFF00H ~ FFFFFH 的区域。

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

各操作位单位的指定方法如下所示。

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

SFR 一览表如表 3-5 所示。表中的项目的含义如下所示。

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

注意 不能存取未分配 SFR 的地址。

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

表 3-5 特殊功能寄存器（SFR）一览表 (1/5)

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
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
FFF37H	键返回模式寄存器	KRM		R/W	○	○	—	00H
FFF38H	外部中断上升沿允许寄存器 0	EGP0		R/W	○	○	—	00H
FFF39H	外部中断下降沿允许寄存器 0	EGN0		R/W	○	○	—	00H

表 3-5 特殊功能寄存器（SFR）一览表 (2/5)

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	
FFF3AH	外部中断上升沿允许寄存器 1	EGP1		R/W	○	○	—	00H
FFF3BH	外部中断下降沿允许寄存器 1	EGN1		R/W	○	○	—	00H
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
FFF58H	定时器 RD 通用寄存器 C0	TRDGRC0		R/W	—	—	○	FFFFH
FFF59H								
FFF5AH	定时器 RD 通用寄存器 D0	TRDGRD0		R/W	—	—	○	FFFFH
FFF5BH								
FFF5CH	定时器 RD 通用寄存器 C1	TRDGRC1		R/W	—	—	○	FFFFH
FFF5DH								
FFF5EH	定时器 RD 通用寄存器 D1	TRDGRD1		R/W	—	—	○	FFFFH
FFF5FH								
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
FFF90H	12 位间隔定时器的控制寄存器	ITMC		R/W	—	—	○	0FFFH
FFF91H								

表 3-5 特殊功能寄存器（SFR）一览表 (3/5)

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	
FFF92H	秒计数寄存器	SEC	R/W	—	○	—	00H
FFF93H	分钟计数寄存器	MIN	R/W	—	○	—	00H
FFF94H	小时计数寄存器	HOUR	R/W	—	○	—	12H 注
FFF95H	星期计数寄存器	WEEK	R/W	—	○	—	00H
FFF96H	日计数寄存器	DAY	R/W	—	○	—	01H
FFF97H	月计数寄存器	MONTH	R/W	—	○	—	01H
FFF98H	年计数寄存器	YEAR	R/W	—	○	—	00H
FFF99H	时钟误差校正寄存器	SUBCUD	R/W	—	○	—	00H
FFF9AH	闹钟分钟寄存器	ALARMWM	R/W	—	○	—	00H
FFF9BH	闹钟小时寄存器	ALARMWH	R/W	—	○	—	12H
FFF9CH	闹钟星期寄存器	ALARMWW	R/W	—	○	—	00H

注 如果在复位后将 AMPM 位（实时时钟控制寄存器 0（RTCC0）的 bit3）置“1”，就变为“00H”。

表 3-5 特殊功能寄存器 (SFR) 一览表 (4/5)

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	
FFF9DH	实时时钟控制寄存器 0	RTCC0	R/W	○	○	—	00H
FFF9EH	实时时钟控制寄存器 1	RTCC1	R/W	○	○	—	00H
FFFA0H	时钟运行模式控制寄存器	CMC	R/W	—	○	—	00H
FFFA1H	时钟运行状态控制寄存器	CSC	R/W	○	○	—	C0H
FFFA2H	振荡稳定时间计数器的状态寄存器	OSTC	R	○	○	—	00H
FFFA3H	振荡稳定时间选择寄存器	OSTS	R/W	—	○	—	07H
FFFA4H	系统时钟控制寄存器	CKC	R/W	○	○	—	00H
FFFA5H	时钟输出选择寄存器 0	CKS0	R/W	○	○	—	00H
FFFA6H	时钟输出选择寄存器 1	CKS1	R/W	○	○	—	00H
FFFA8H	复位控制标志寄存器	RESF	R	—	○	—	不定值 ^{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	—	○	—	9AH/1AH ^{注2}
FFFACH	CRC 输入寄存器	CRCIN	R/W	—	○	—	00H
FFFD0H	中断请求标志寄存器 2L	IF2L	IF2	R/W	○	○	00H
FFFD1H	中断请求标志寄存器 2H	IF2H		R/W	○	○	00H
FFFD4H	中断屏蔽标志寄存器 2L	MK2L	MK2	R/W	○	○	FFH
FFFD5H	中断屏蔽标志寄存器 2H	MK2H		R/W	○	○	FFH
FFFD8H	优先级指定标志寄存器 02L	PR02L	PR02	R/W	○	○	FFH
FFFD9H	优先级指定标志寄存器 02H	PR02H		R/W	○	○	FFH
FFFDCH	优先级指定标志寄存器 12L	PR12L	PR12	R/W	○	○	FFH
FFDDH	优先级指定标志寄存器 12H	PR12H		R/W	○	○	FFH

注 1. 以下寄存器的内容因复位源而不同。

复位源 寄存器		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	
	IAWRF			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 特殊功能寄存器（SFR）一览表 (5/5)

地址	特殊功能寄存器（SFR）名称	符号		R/W	可操作位的范围			复位后
					1 位	8 位	16 位	
FFFE0H	中断请求标志寄存器 0L	IF0L	IF0	R/W	○	○	○	00H
FFFE1H	中断请求标志寄存器 0H	IF0H		R/W	○	○		00H
FFFE2H	中断请求标志寄存器 1L	IF1L	IF1	R/W	○	○	○	00H
FFFE3H	中断请求标志寄存器 1H	IF1H		R/W	○	○		00H
FFFE4H	中断屏蔽标志寄存器 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	乘和运算累加寄存器（L）	MACRL		R/W	—	—	○	0000H
FFFF1H								
FFFF2H	乘和运算累加寄存器（H）	MACRH		R/W	—	—	○	0000H
FFFF3H								
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。

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

各操作位单位的指定方法如下所示。

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

扩展 SFR 一览表如表 3-6 所示。表中的项目的含义如下所示。

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

注意 不能存取未分配 2nd SFR 的地址。

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

表 3-6 扩展特殊功能寄存器 (2nd SFR) 一览表 (1/7)

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
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
F0043H	端口输入模式寄存器 3	PIM3	R/W	○	○	—	00H
F0045H	端口输入模式寄存器 5	PIM5	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
F0053H	端口输出模式寄存器 3	POM3	R/W	○	○	—	00H
F0055H	端口输出模式寄存器 5	POM5	R/W	○	○	—	00H
F0057H	端口输出模式寄存器 7	POM7	R/W	○	○	—	00H
F0060H	端口模式控制寄存器 0	PMC0	R/W	○	○	—	FFH
F006CH	端口模式控制寄存器 12	PMC12	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
F0073H	输入切换控制寄存器	ISC	R/W	○	○	—	00H
F0074H	定时器输入选择寄存器 0	TIS0	R/W	—	○	—	00H
F0076H	A/D 端口配置寄存器	ADPC	R/W	—	○	—	00H
F0077H	外围 I/O 重定向寄存器 0	PIOR0	R/W	—	○	—	00H
F0078H	非法存储器存取检测控制寄存器	IAWCTL	R/W	—	○	—	00H
F0079H	外围 I/O 重定向寄存器 1	PIOR1	R/W	—	○	—	00H
F007AH	外围允许寄存器 1	PER1	R/W	○	○	—	00H
F007BH	端口模式选择寄存器	PMS	R/W	○	○	—	00H
F007DH	全局数字输入禁止寄存器	GDIDIS	R/W	○	○	—	00H

表 3-6 扩展特殊功能寄存器 (2nd SFR) 一览表 (2/7)

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	
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}
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	SSR00	R	—	○	0000H
F0101H		—			—	—	
F0102H	串行状态寄存器 01	SSR01L	SSR01	R	—	○	0000H
F0103H		—			—	—	
F0104H	串行状态寄存器 02	SSR02L	SSR02	R	—	○	0000H
F0105H		—			—	—	
F0106H	串行状态寄存器 03	SSR03L	SSR03	R	—	○	0000H
F0107H		—			—	—	
F0108H	串行标志清除触发寄存器 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							

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 扩展特殊功能寄存器 (2nd SFR) 一览表 (3/7)

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	
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		—			—	—		
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
		—			—	—		
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								

表 3-6 扩展特殊功能寄存器 (2nd SFR) 一览表 (4/7)

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	
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	—	—	○	0F0FH
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		—			—			
F0178H	串行待机控制寄存 1	SSC1L	SSC1	R/W	—	○	○	0000H
F0179H		—			—			
F0180H	定时器计数寄存器 00	TCR00		R	—	—	○	FFFFH
F0181H								
F0182H	定时器计数寄存器 01	TCR01		R	—	—	○	FFFFH
F0183H								
F0184H	定时器计数寄存器 02	TCR02		R	—	—	○	FFFFH
F0185H								
F0186H	定时器计数寄存器 03	TCR03		R	—	—	○	FFFFH
F0187H								
F0190H	定时器模式寄存器 00	TMR00		R/W	—	—	○	0000H
F0191H								
F0192H	定时器模式寄存器 01	TMR01		R/W	—	—	○	0000H
F0193H								
F0194H	定时器模式寄存器 02	TMR02		R/W	—	—	○	0000H
F0195H								
F0196H	定时器模式寄存器 03	TMR03		R/W	—	—	○	0000H
F0197H								
F01A0H	定时器状态寄存器 00	TSR00L	TSR00	R	—	○	○	0000H
F01A1H		—			—			
F01A2H	定时器状态寄存器 01	TSR01L	TSR01	R	—	○	○	0000H
F01A3H		—			—			
F01A4H	定时器状态寄存器 02	TSR02L	TSR02	R	—	○	○	0000H
F01A5H		—			—			
F01A6H	定时器状态寄存器 03	TSR03L	TSR03	R	—	○	○	0000H
F01A7H		—			—			
F01B0H	定时器通道允许状态寄存器 0	TE0L	TE0	R	○	○	○	0000H
F01B1H		—			—			

表 3-6 扩展特殊功能寄存器 (2nd SFR) 一览表 (5/7)

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	
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
F0240H	定时器 RJ 控制寄存器 0	TRJCR0		R/W	—	○	—	00H
F0241H	定时器 RJ I/O 控制寄存器 0	TRJIOC0		R/W	○	○	—	00H
F0242H	定时器 RJ 模式寄存器 0	TRJMR0		R/W	○	○	—	00H
F0243H	定时器 RJ 事件引脚选择寄存器 0	TRJISR0		R/W	○	○	—	00H
F0260H	定时器 RD ELC 寄存器	TRDEL0		R/W	○	○	—	00H
F0263H	定时器 RD 启动寄存器	TRDSTR		R/W	—	○	—	0CH 注
F0264H	定时器 RD 模式寄存器	TRDMR		R/W	○	○	—	00H
F0265H	定时器 RD PWM 功能选择寄存器	TRDPMR		R/W	○	○	—	00H
F0266H	定时器 RD 功能控制寄存器	TRDFCR		R/W	○	○	—	80H 注
F0267H	定时器 RD 输出主允许寄存器 1	TRDOER1		R/W	○	○	—	FFH 注
F0268H	定时器 RD 输出主允许寄存器 2	TRDOER2		R/W	○	○	—	00H
F0269H	定时器 RD 输出控制寄存器	TRDOCR		R/W	○	○	—	00H
F026AH	定时器 RD 数字滤波器功能选择寄存器 0	TRDDF0		R/W	○	○	—	00H
F026BH	定时器 RD 数字滤波器功能选择寄存器 1	TRDDF1		R/W	○	○	—	00H

注 当用户选项字节 (000C2H) 的 FRQSEL4 位为“1”并且 PER1 寄存器的 TRD0EN 位为“0”时, 定时器 RD 的 SFR 为“00H”。当需要读初始值时, 必须在将 f_{CLK} 设定为 f_{IH} 并且将 TRD0EN 位置“1”后读初始值。

表 3-6 扩展特殊功能寄存器 (2nd SFR) 一览表 (6/7)

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	
F0270H	定时器 RD 控制寄存器 0	TRDCR0	R/W	○	○	—	00H
F0271H	定时器 RD I/O 控制寄存器 A0	TRDIORA0	R/W	○	○	—	00H
F0272H	定时器 RD I/O 控制寄存器 C0	TRDIORC0	R/W	○	○	—	88H 注
F0273H	定时器 RD 状态寄存器 0	TRDSR0	R/W	○	○	—	00H
F0274H	定时器 RD 中断允许寄存器 0	TRDIER0	R/W	○	○	—	00H
F0275H	定时器 RD PWM 功能输出电平 控制寄存器 0	TRDPOCR0	R/W	○	○	—	00H
F0276H	定时器 RD 计数器 0	TRD0	R/W	—	—	○	0000H
F0277H							
F0278H	定时器 RD 通用寄存器 A0	TRDGRA0	R/W	—	—	○	FFFFH
F0279H							
F027AH	定时器 RD 通用寄存器 B0	TRDGRB0	R/W	—	—	○	FFFFH
F027BH							
F0280H	定时器 RD 控制寄存器 1	TRDCR1	R/W	○	○	—	00H
F0281H	定时器 RD I/O 控制寄存器 A1	TRDIORA1	R/W	○	○	—	00H
F0282H	定时器 RD I/O 控制寄存器 C1	TRDIORC1	R/W	○	○	—	88H 注
F0283H	定时器 RD 状态寄存器 1	TRDSR1	R/W	○	○	—	00H
F0284H	定时器 RD 中断允许寄存器 1	TRDIER1	R/W	○	○	—	00H
F0285H	定时器 RD PWM 功能输出电平 控制寄存器 1	TRDPOCR1	R/W	○	○	—	00H
F0286H	定时器 RD 计数器 1	TRD1	R/W	—	—	○	0000H
F0287H							
F0288H	定时器 RD 通用寄存器 A1	TRDGRA1	R/W	—	—	○	FFFFH
F0289H							
F028AH	定时器 RD 通用寄存器 B1	TRDGRB1	R/W	—	—	○	FFFFH
F028BH							
F02E0H	DTC 基址寄存器	DTCBAR	R/W	○	○	—	FDH
F02E8H	DTC 启动允许寄存器 0	DTCEN0	R/W	○	○	—	00H
F02E9H	DTC 启动允许寄存器 1	DTCEN1	R/W	○	○	—	00H
F02EAH	DTC 启动允许寄存器 2	DTCEN2	R/W	○	○	—	00H
F02EBH	DTC 启动允许寄存器 3	DTCEN3	R/W	○	○	—	00H
F02ECH	DTC 启动允许寄存器 4	DTCEN4	R/W	○	○	—	00H
F02F0H	闪存 CRC 控制寄存器	CRC0CTL	R/W	○	○	—	00H
F02F2H	闪存 CRC 运算结果寄存器	PGCRCL	R/W	—	—	○	0000H
F02FAH	CRC 数据寄存器	CRCD	R/W	—	—	○	0000H

注 当用户选项字节 (000C2H) 的 FRQSEL4 位为“1”并且 PER1 寄存器的 TRD0EN 位为“0”时, 定时器 RD 的 SFR 为“00H”。当需要读初始值时, 必须在将 f_{CLK} 设定为 f_{IH} 并且将 TRD0EN 位置“1”后读初始值。

表 3-6 扩展特殊功能寄存器（2nd SFR）一览表 (7/7)

地址	扩展特殊功能寄存器 (2nd SFR) 名称	符号	R/W	可操作位的范围			复位后
				1 位	8 位	16 位	
F0300H	事件输出目标选择寄存器 00	ELSELR00	R/W	—	○	—	00H
F0301H	事件输出目标选择寄存器 01	ELSELR01	R/W	—	○	—	00H
F0302H	事件输出目标选择寄存器 02	ELSELR02	R/W	—	○	—	00H
F0303H	事件输出目标选择寄存器 03	ELSELR03	R/W	—	○	—	00H
F0304H	事件输出目标选择寄存器 04	ELSELR04	R/W	—	○	—	00H
F0305H	事件输出目标选择寄存器 05	ELSELR05	R/W	—	○	—	00H
F0306H	事件输出目标选择寄存器 06	ELSELR06	R/W	—	○	—	00H
F0307H	事件输出目标选择寄存器 07	ELSELR07	R/W	—	○	—	00H
F0308H	事件输出目标选择寄存器 08	ELSELR08	R/W	—	○	—	00H
F0309H	事件输出目标选择寄存器 09	ELSELR09	R/W	—	○	—	00H
F030AH	事件输出目标选择寄存器 10	ELSELR10	R/W	—	○	—	00H
F030BH	事件输出目标选择寄存器 11	ELSELR11	R/W	—	○	—	00H
F030CH	事件输出目标选择寄存器 12	ELSELR12	R/W	—	○	—	00H
F030DH	事件输出目标选择寄存器 13	ELSELR13	R/W	—	○	—	00H
F030EH	事件输出目标选择寄存器 14	ELSELR14	R/W	—	○	—	00H
F030FH	事件输出目标选择寄存器 15	ELSELR15	R/W	—	○	—	00H
F0310H	事件输出目标选择寄存器 16	ELSELR16	R/W	—	○	—	00H
F0311H	事件输出目标选择寄存器 17	ELSELR17	R/W	—	○	—	00H
F0500H	定时器 RJ 计数寄存器 0	TRJ0	R/W	—	—	○	FFFFH
F0501H							

备注 有关 SFR 区的 SFR，请参照“表 3-5 SFR 一览表”。

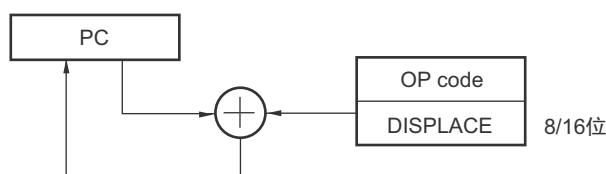
3.3 指令地址的寻址

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 3-10 相对寻址的概要



3.3.2 立即寻址

【功能】

立即寻址将指令码中的立即数保存到程序计数器，指定转移目标的程序地址。

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

图 3-11 CALL !!addr20/BR !!addr20 的例子

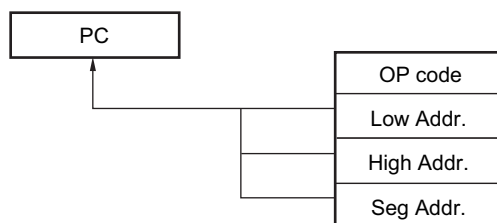
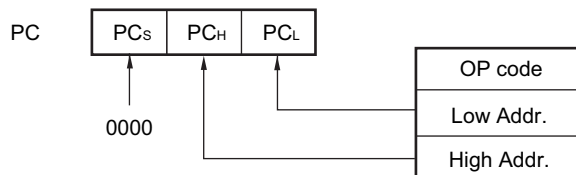


图 3-12 CALL !addr16/BR !addr16 的例子



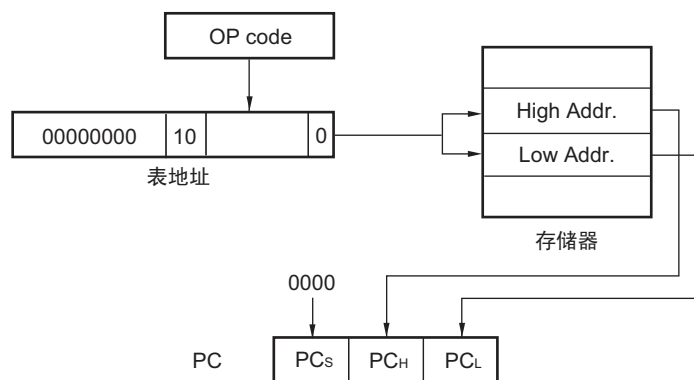
3.3.3 表间接寻址

【功能】

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

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

图 3-13 表间接寻址的概要

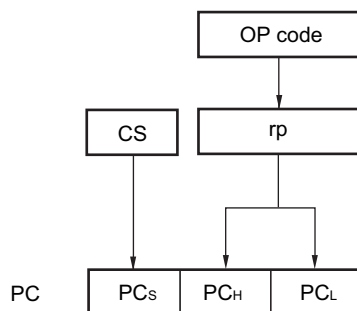


3.3.4 寄存器直接寻址

【功能】

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

图 3-14 寄存器直接寻址的概要



3.4 处理数据地址的寻址

3.4.1 隐含寻址

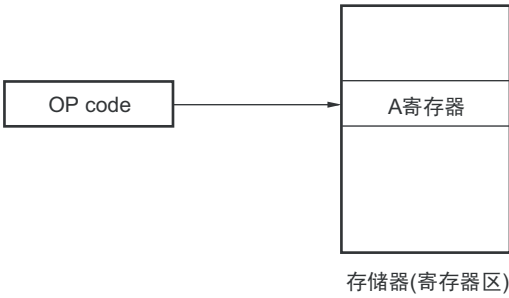
【功能】

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

【操作数形式】

隐含寻址只适用于 MULU X 指令。

图 3-15 隐含寻址的概要



3.4.2 寄存器寻址

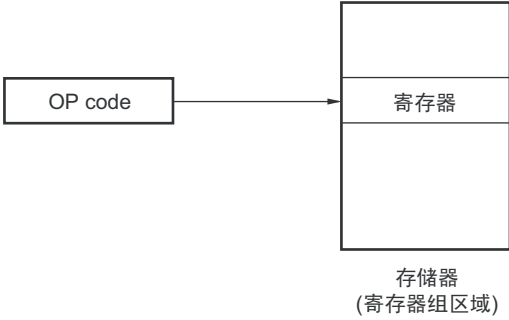
【功能】

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

【操作数形式】

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

图 3-16 寄存器寻址的概要



3.4.3 直接寻址

【功能】

直接寻址是以指令码中的立即数为操作数地址来直接指定对象地址的寻址方式。

【操作数形式】

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

图 3-17 !addr16 的例子

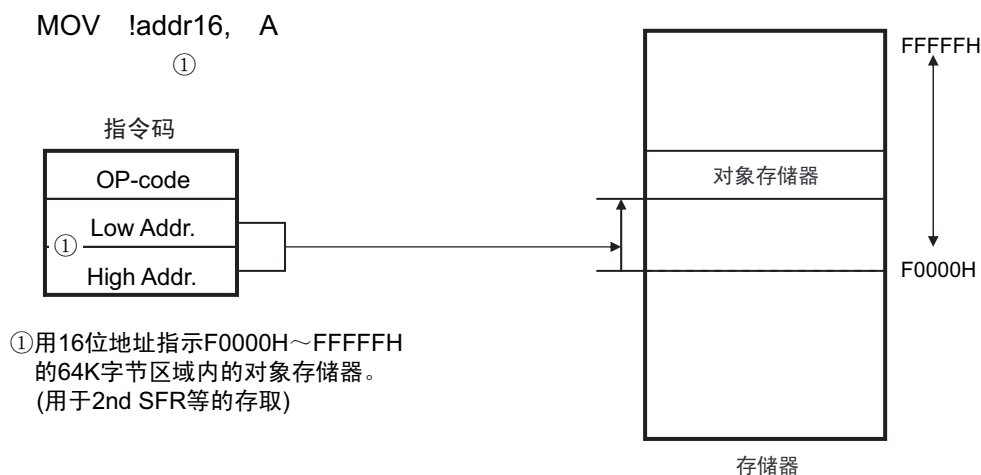
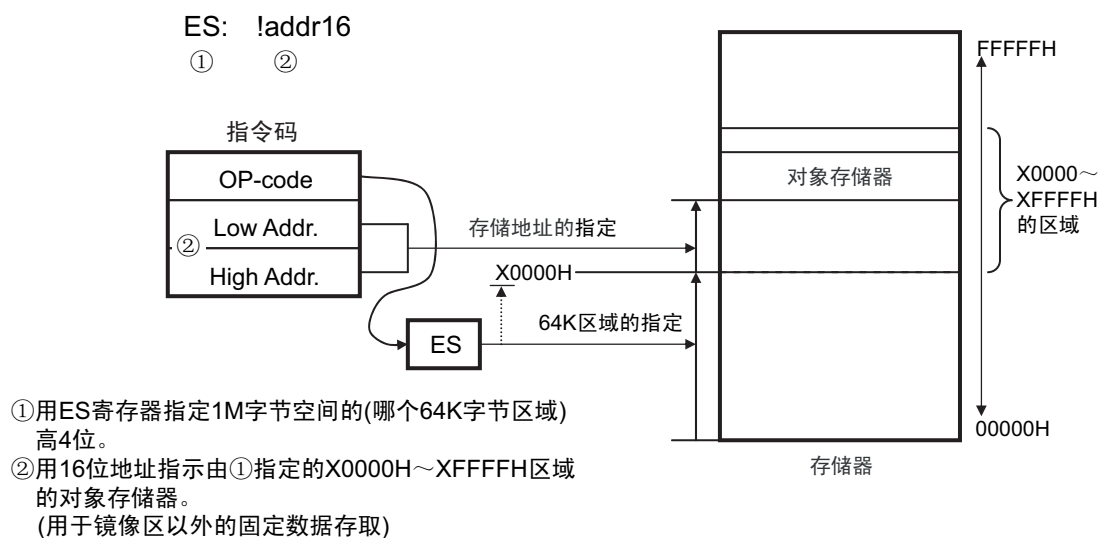


图 3-18 ES:!addr16 的例子



3.4.4 短直接寻址

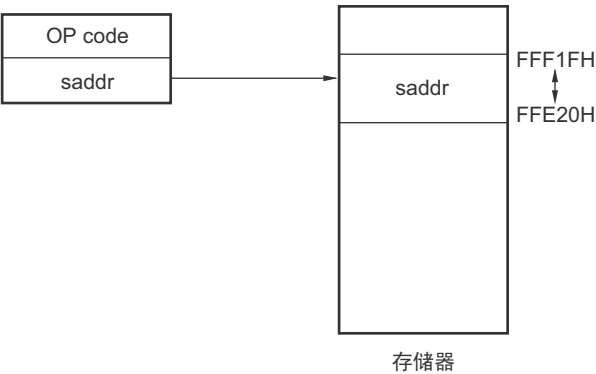
【功能】

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

【操作数形式】

表现形式	记述方法
SADDR	标号、 FFE20H ~ FFF1FH 的立即数或者 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 3-19 短直接寻址的概要



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

3.4.5 SFR 寻址

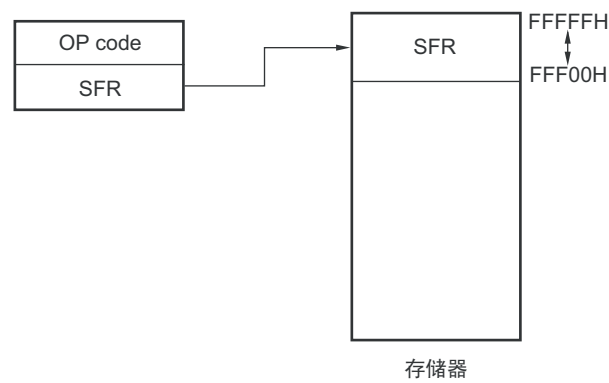
【功能】

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

【操作数形式】

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

图 3-20 SFR 寻址的概要



3.4.6 寄存器间接寻址

【功能】

寄存器间接寻址以指令码指定的寄存器对的内容为操作数地址，指定对象地址。

【操作数形式】

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

图 3-21 [DE]、[HL] 的例子

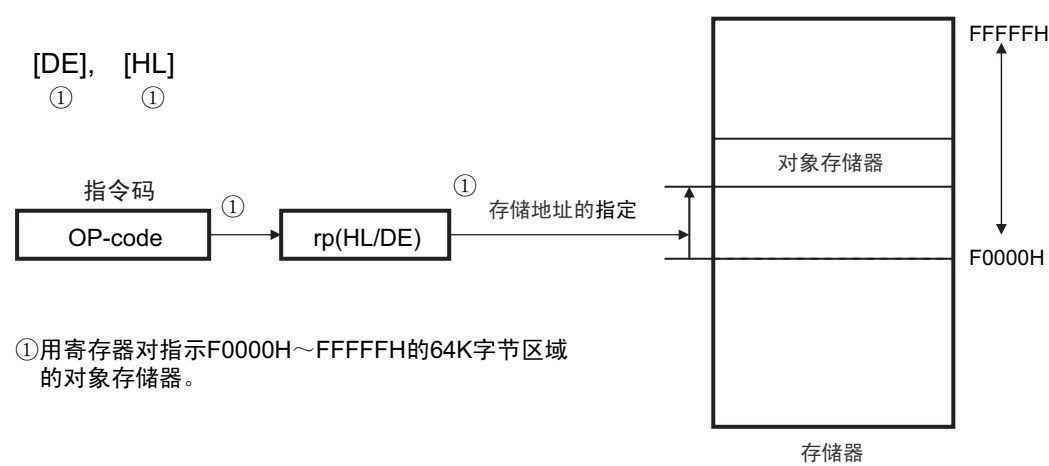
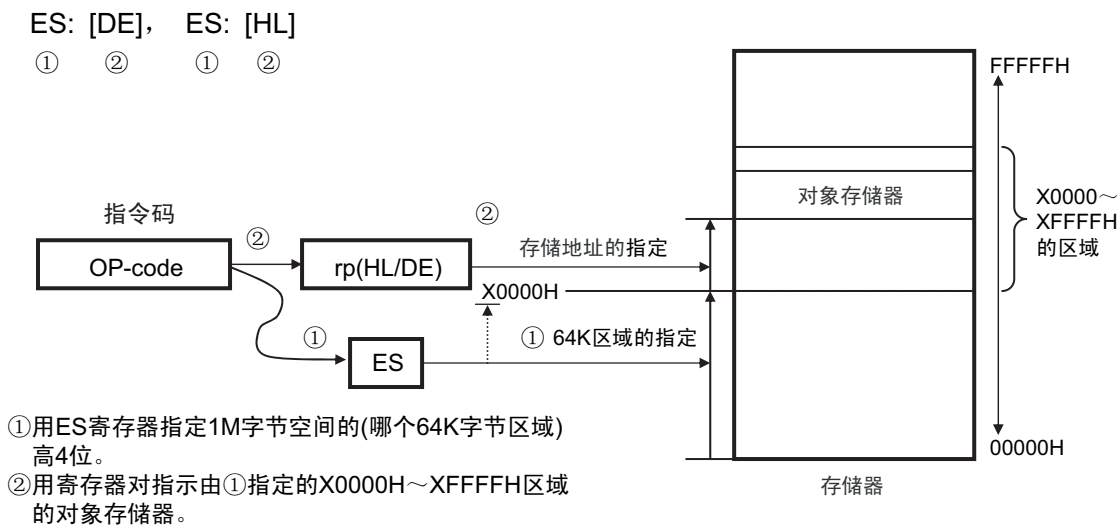


图 3-22 ES:[DE]、ES:[HL] 的例子



3.4.7 基址寻址

【功能】

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

【操作数形式】

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

图 3-23 [SP+byte] 的例子

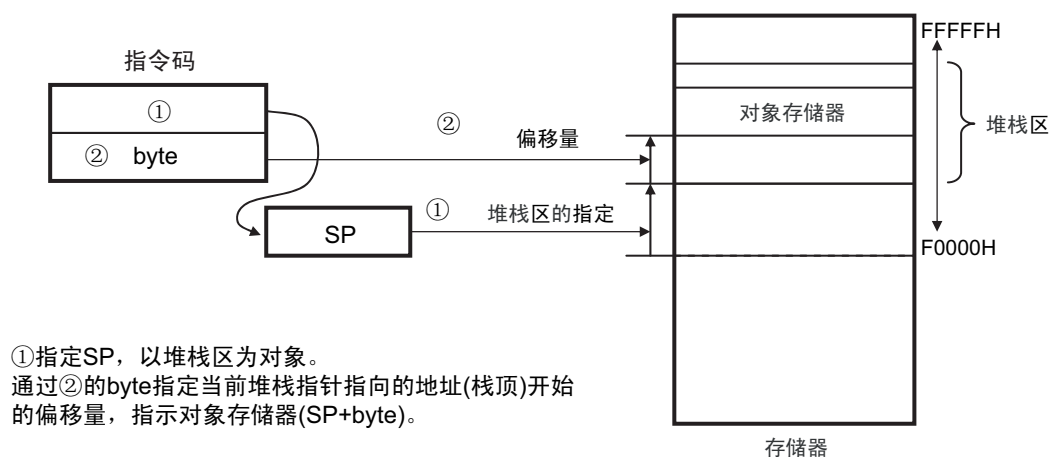


图 3-24 [HL+byte]、[DE+byte] 的例子

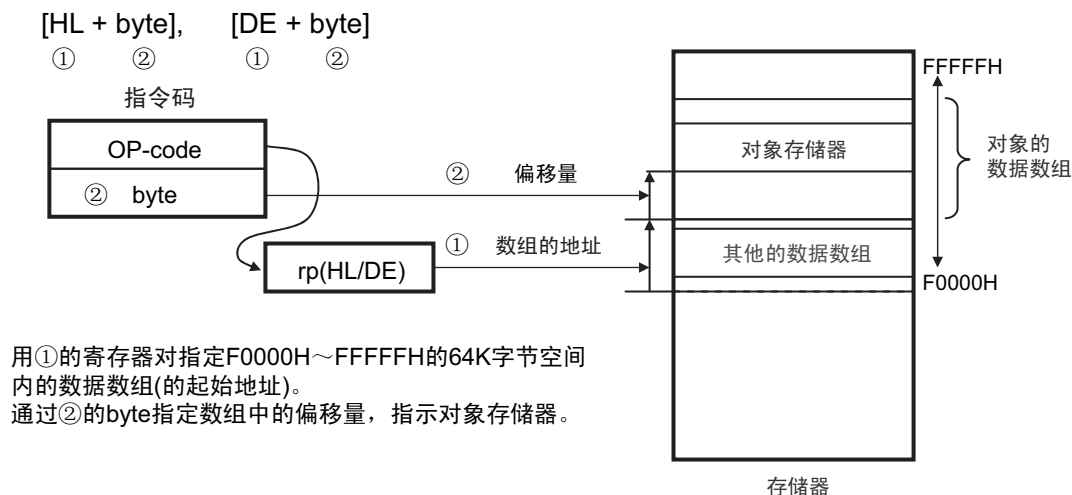


图 3-25 word[B]、word[C] 的例子

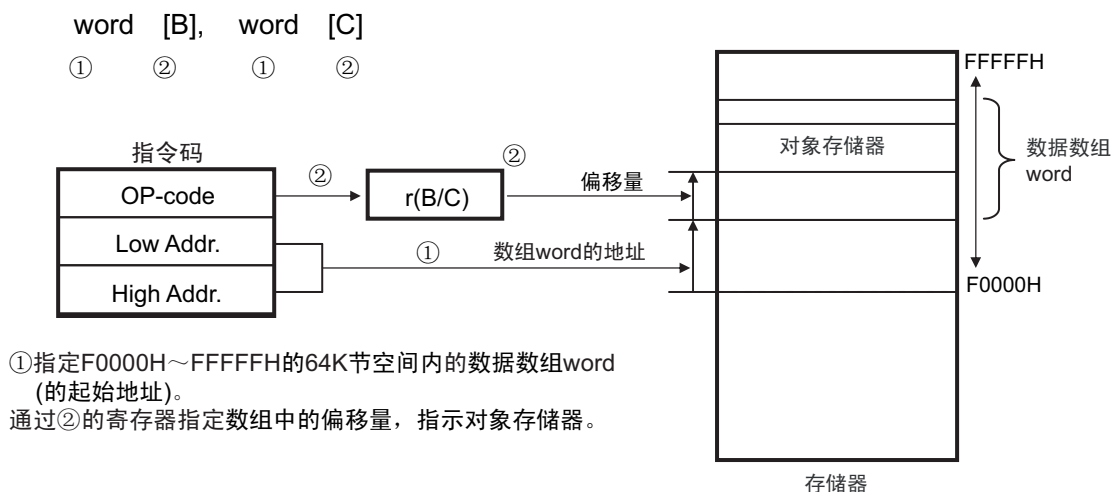


图 3-26 word[BC] 的例子

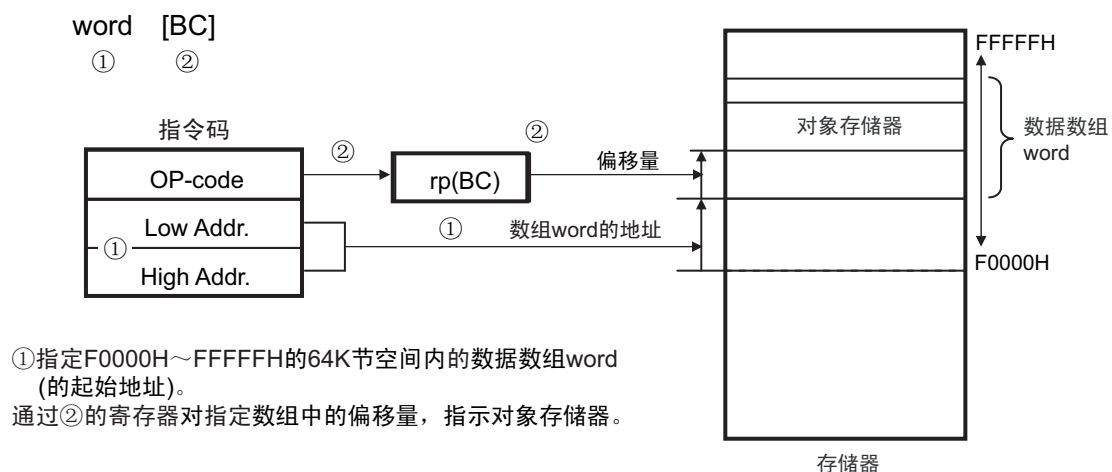


图 3-27 ES:[HL+byte]、ES:[DE+byte] 的例子

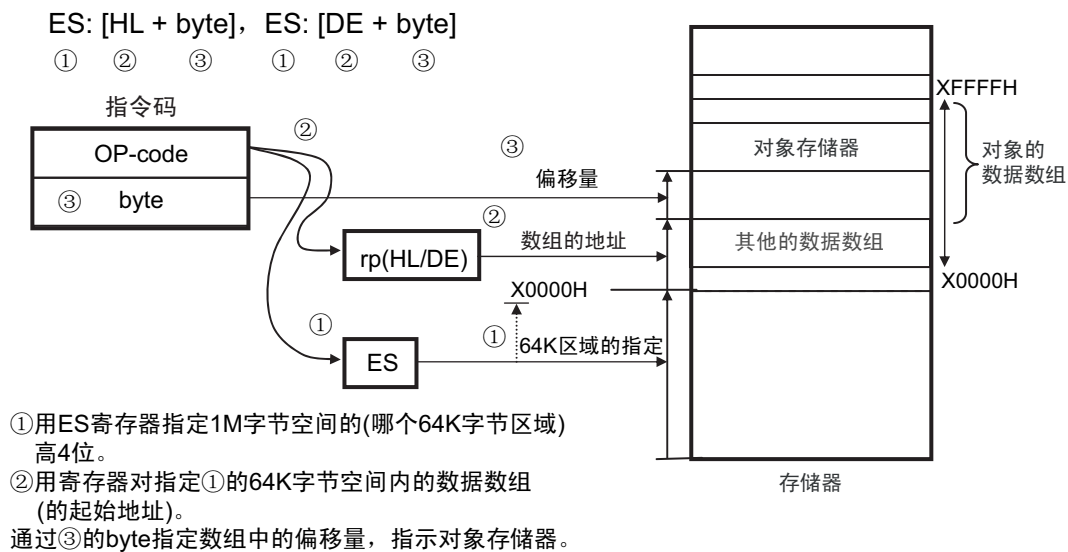


图 3-28 ES:word[B]、ES:word[C] 的例子

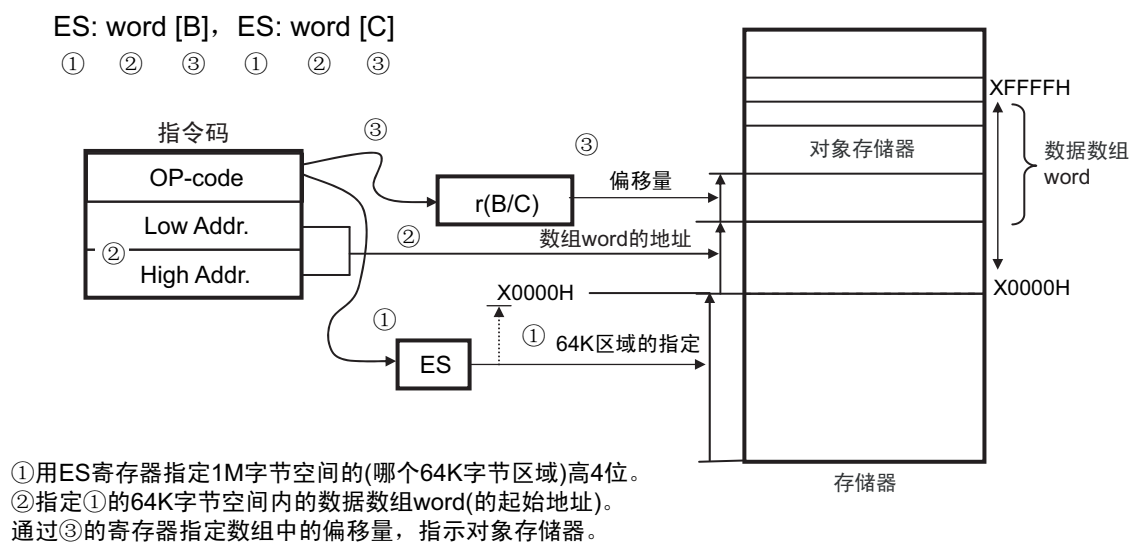
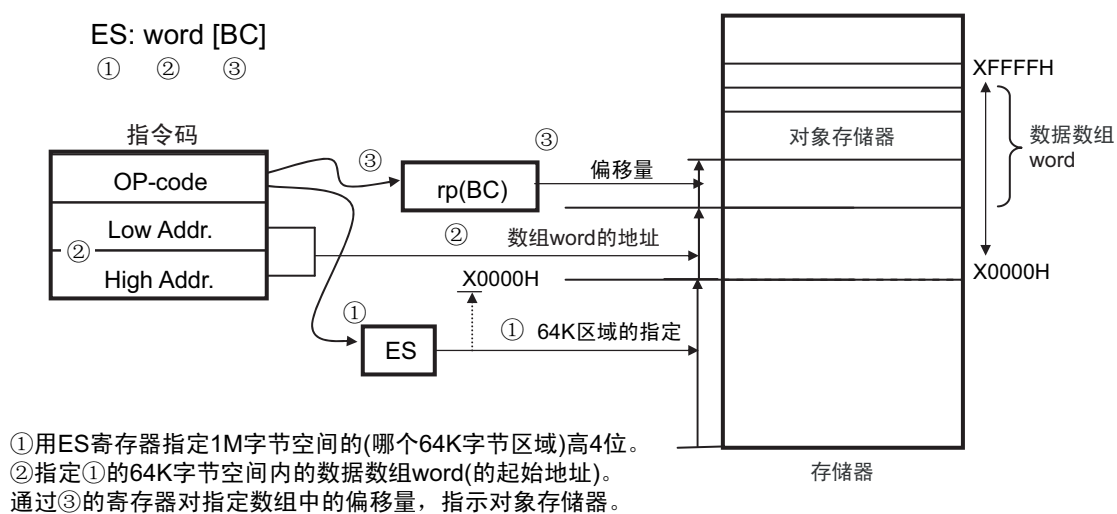


图 3-29 ES:word[BC] 的例子



3.4.8 基址变址寻址

【功能】

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

【操作数形式】

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

图 3-30 [HL+B]、[HL+C] 的例子

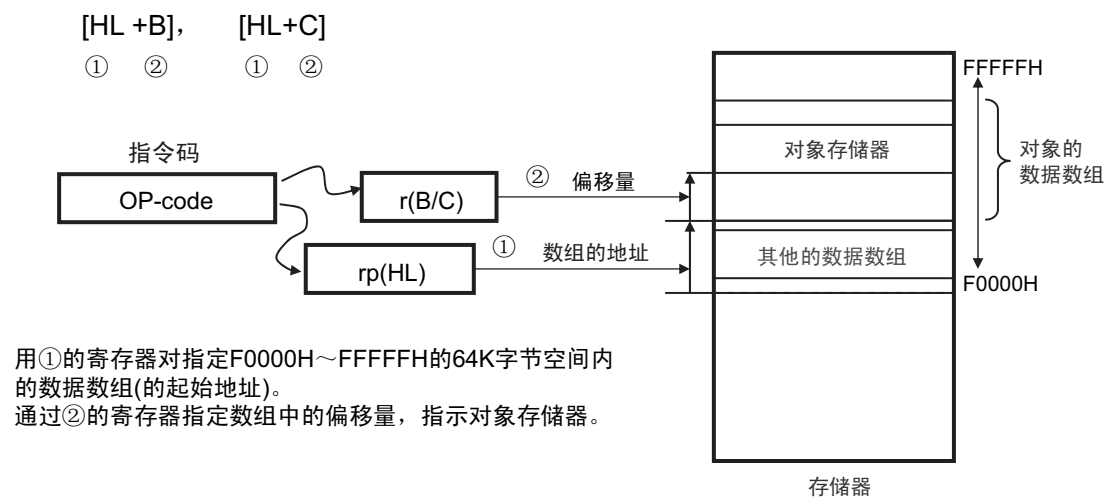
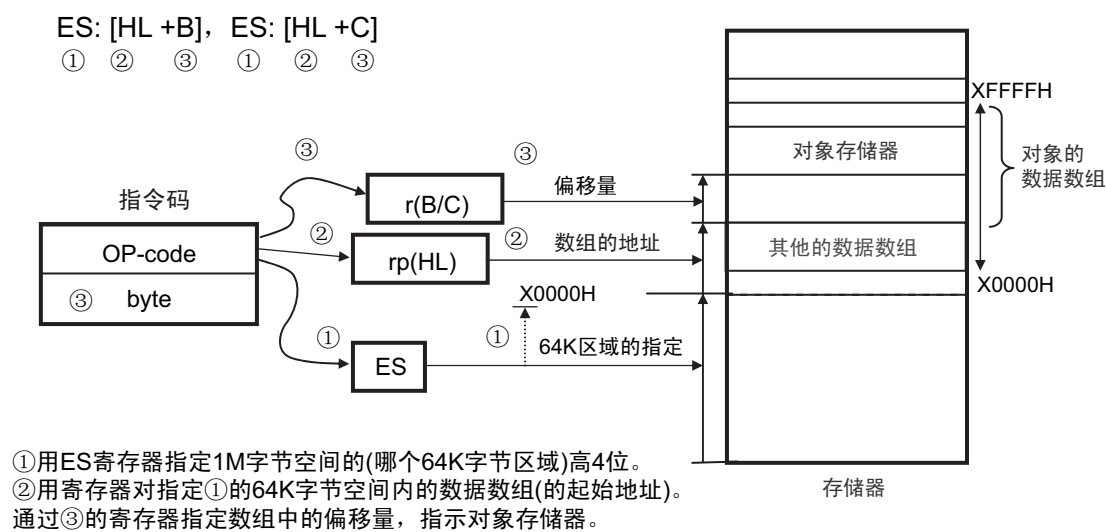


图 3-31 ES:[HL+B]、ES:[HL+C] 的例子



3.4.9 堆栈寻址

【功能】

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

只能将堆栈区设定在内部 RAM 区。

【记述形式】

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

根据各堆栈的运行，压栈和退栈的数据如图 3-32 ~ 图 3-37 所示。

图 3-32 PUSH rp 的例子

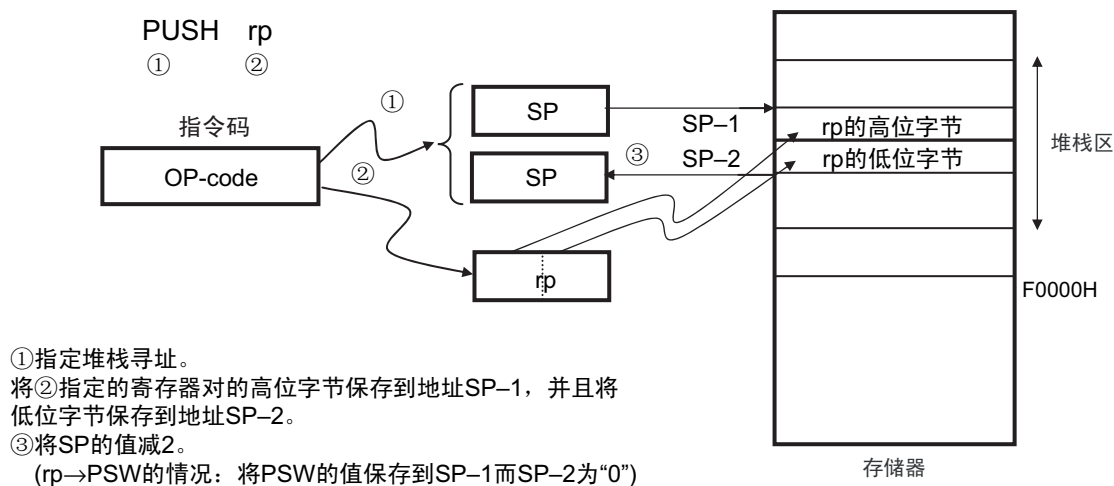


图 3-33 POP 的例子

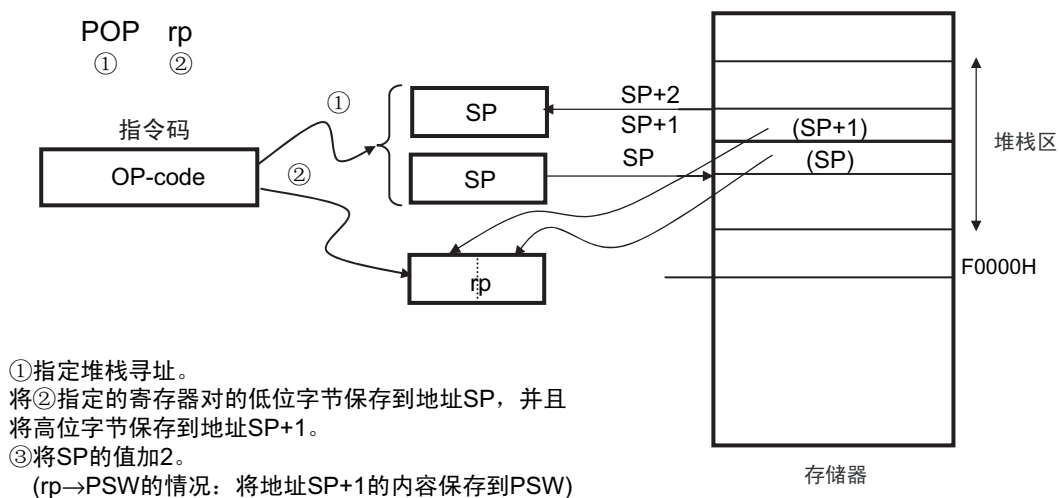


图 3-34 CALL、CALLT 的例子

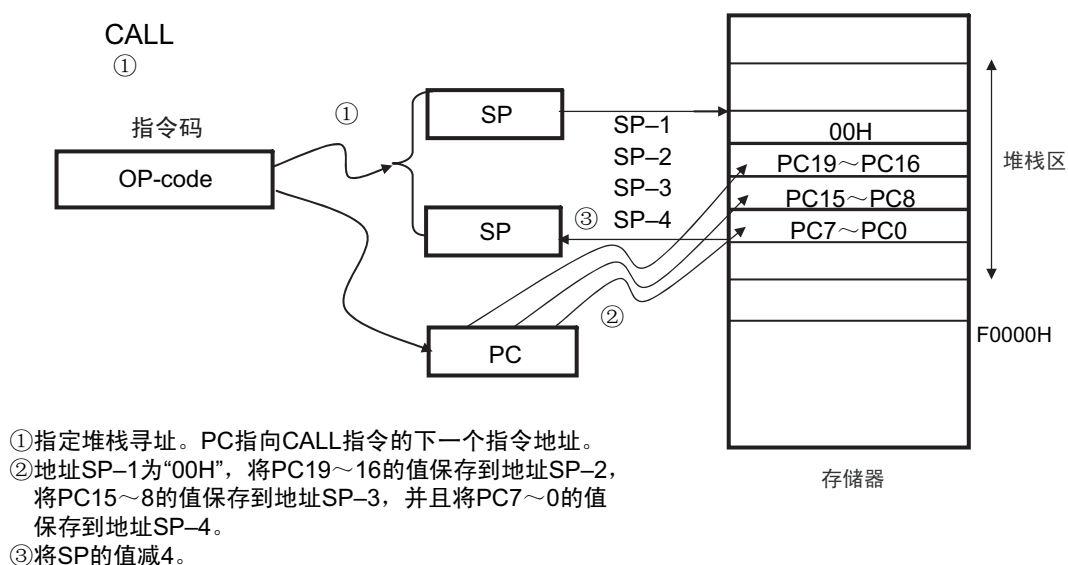


图 3-35 RET 的例子

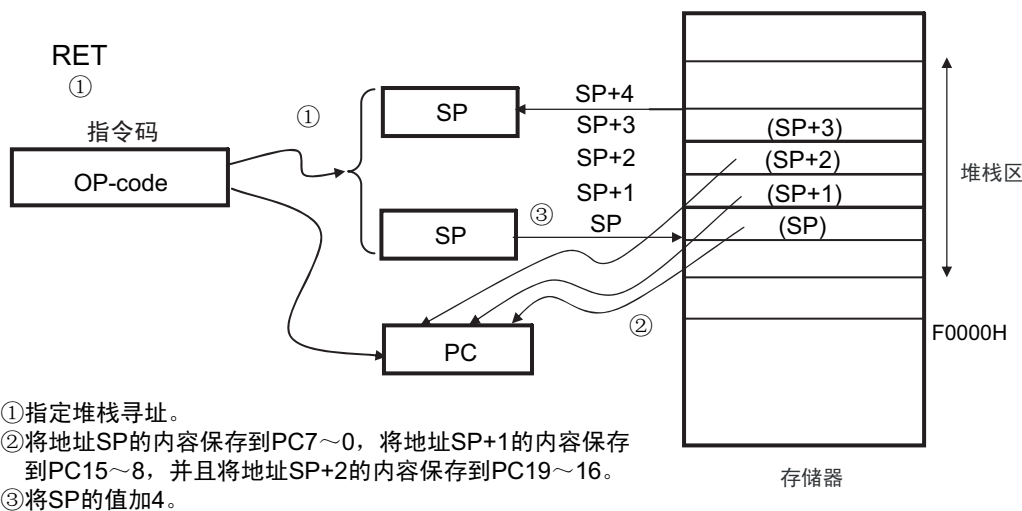


图 3-36 中断、BRK 的例子

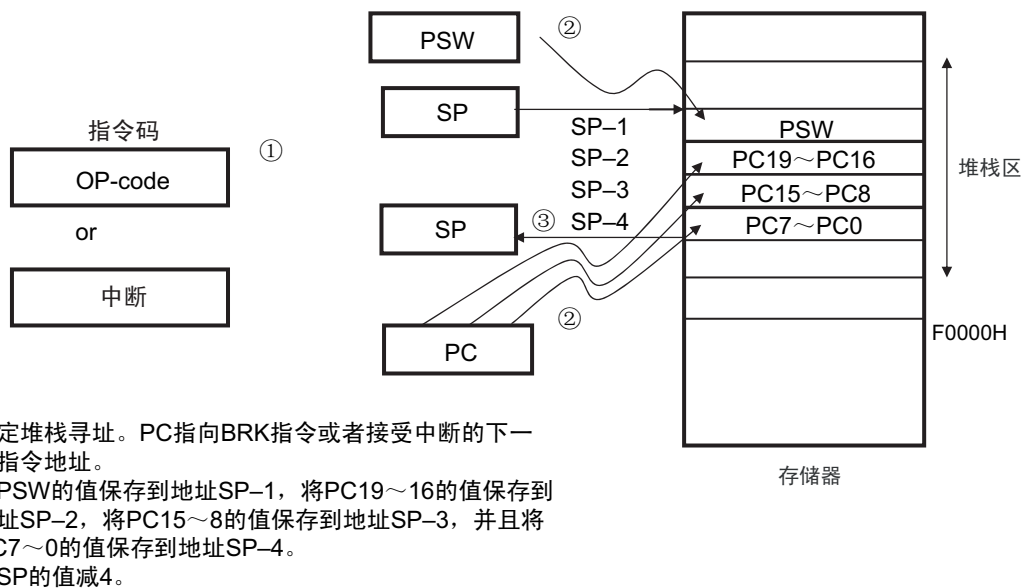
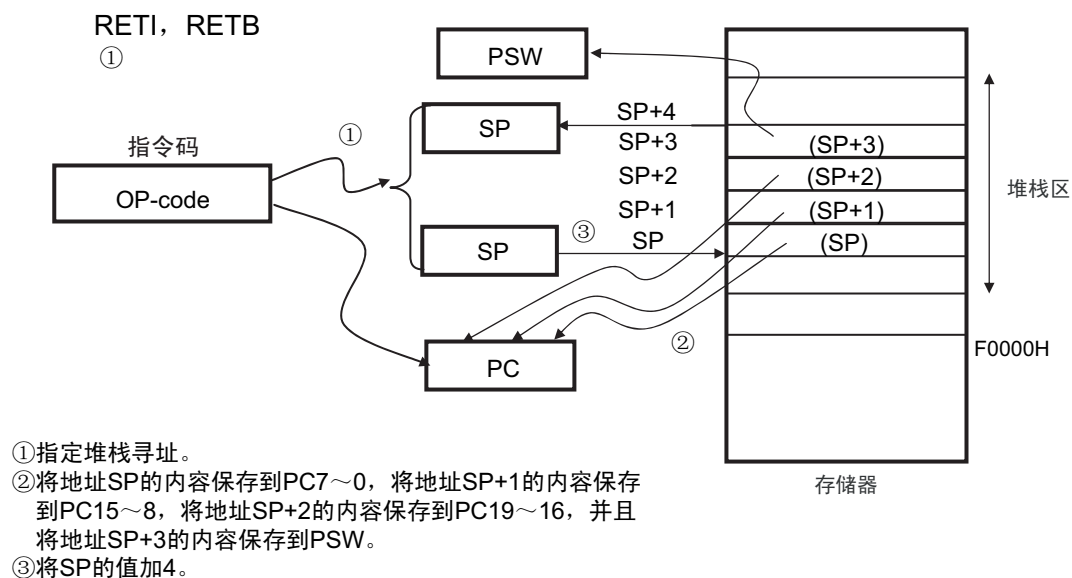


图 3-37 RETI、RETB 的例子



第4章 端口功能

4.1 端口功能

R7F0C014B2D、R7F0C014L2D 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、PIM3、PIM5） 端口输出模式寄存器（POM0、POM1、POM3、POM5、POM7） 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（PIOR0、PIOR1） 全局数字输入禁止寄存器（GDIDIS）
端口	32 引脚产品： 合计：28 个（CMOS 输入 / 输出：22 个，CMOS 输入：3 个，N 沟道漏极开路输入 / 输出：3 个） 64 引脚产品： 合计：58 个（CMOS 输入 / 输出：48 个，CMOS 输入：5 个，CMOS 输出：1 个，N 沟道漏极开路输入 / 输出：4 个）
上拉电阻	32 引脚产品：合计：18 个 64 引脚产品：合计：40 个

注意 本章的下述内容主要是针对 64 引脚产品在外围 I/O 重定向寄存器 0、1（PIOR0、PIOR1）为“00H”时的说明。

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 ~ P06 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（EV_{DD} 耐压）。

当将 P00 ~ P03 引脚用作输入时，必须通过端口模式控制寄存器 0（PMC0）设定数字或者模拟（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，变为以下状态：

- 32 引脚产品的 P00 引脚和 P01 引脚为模拟输入。
- 其他产品的 P00、P01、P04 ~ P06 引脚为输入模式，P02 引脚和 P03 引脚为模拟输入。

表 4-2 使用端口 0 时的寄存器设定

引脚名		PM0x	PIM0x	POM0x	PMC0x	复用功能的设定	备注
名称	输入 / 输出						
P00	输入	1	—	×	0 注 1	×	
	输出	0		0	0 注 1	TxD1 输出 =1 注 3	CMOS 输出
		0		1	0 注 1		N 沟道漏极开路输出
P01	输入	1	0	—	0 注 1	×	CMOS 输入
		1	1		0 注 1	×	TTL 输入
	输出	0	×		0 注 1	TO00 输出 =0 注 4 TRJIO0 输出 =0 注 5	
P02	输入	1	—	×	0 注 2	×	
	输出	0		0	0 注 2	SO10/TxD1 输出 =1 注 6	CMOS 输出
		0		1	0 注 2		N 沟道漏极开路输出
P03	输入	1	0	×	0 注 2	×	CMOS 输入
		1	1	×	0 注 2	×	TTL 输入
	输出	0	×	0	0 注 2	SDA10 输出 =1 注 6	CMOS 输出
		0	×	1	0 注 2		N 沟道漏极开路输出
P04	输入	1	0	×	—	×	CMOS 输入
		1	1	×		×	TTL 输入
	输出	0	×	0		SCK10/SCL10 输出 =1 注 6	CMOS 输出
		0	×	1			N 沟道漏极开路输出
P05、P06	输入	1	—	—	—	—	
	输出	0					

注 1. 只限于 32 引脚产品。

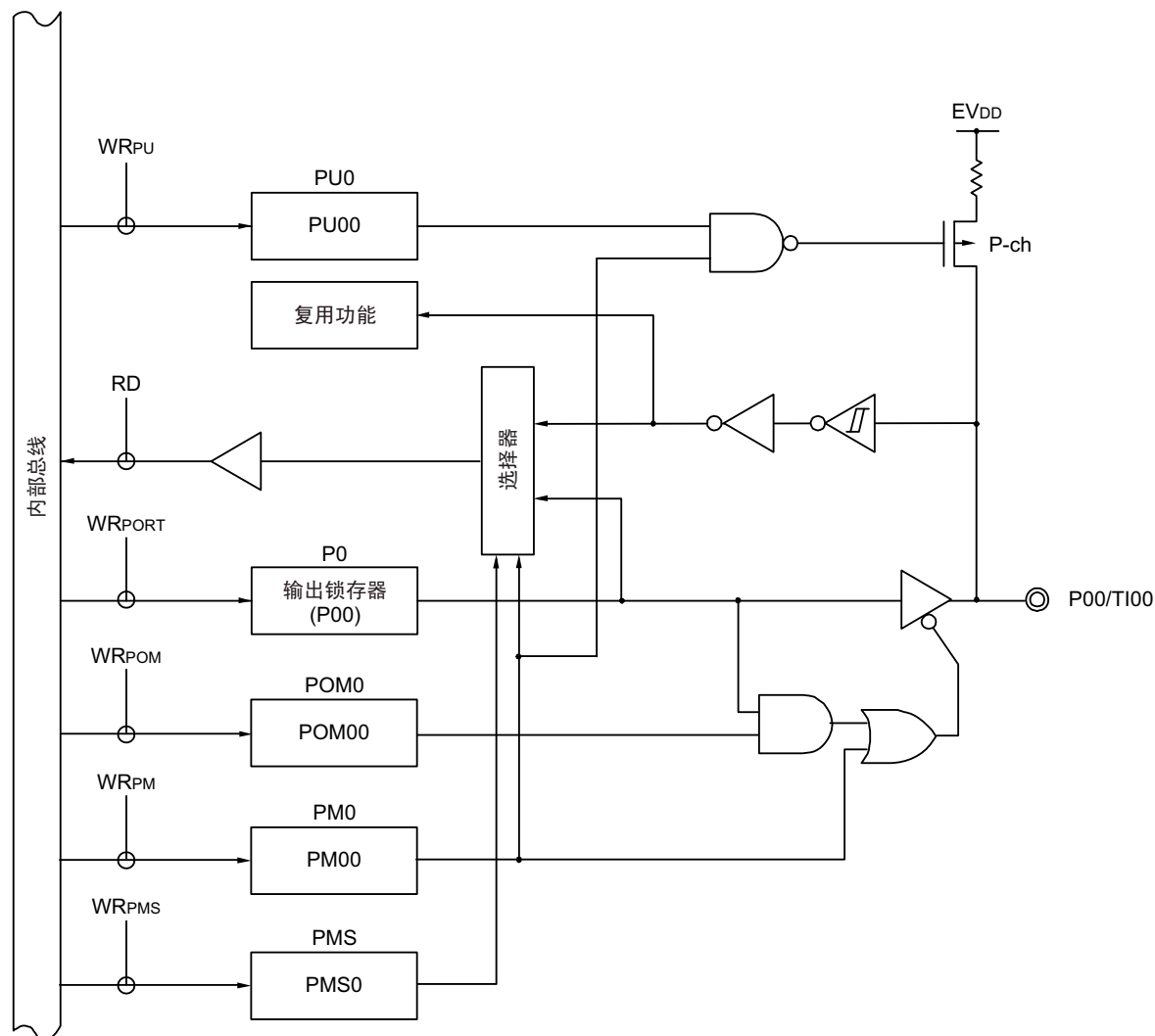
2. 只限于 64 引脚产品。

- 注 3. 在 32 引脚产品的情况下，当将串行阵列单元功能的复用引脚用作通用端口时，必须将对象单元通道对应的串行输出寄存器 m (SOM) 的 SOMn 位、串行输出允许寄存器 m (SOEm) 的 SOEmn 位和串行通道允许状态寄存器 m (SEm) 的 SEmn 位置初始值 (mn=02)。
4. 当将定时器阵列单元的定时器输出功能的复用引脚用作通用端口时，必须将对象单元通道对应的定时器输出寄存器 m (TOM) 的 TOMn 位和定时器输出允许寄存器 m (TOEm) 的 TOEmn 位置初始值 (m=0, n=0)。
5. 当将定时器 RJ 的定时器输入 / 输出功能的复用引脚用作通用端口时，必须将定时器 RJ 模式寄存器 0 (TRJMR0) 的 TMOD2 ~ 0 位置初始值或者“001B”以外的数据。
6. 在 64 引脚产品的情况下，当将串行阵列单元功能的复用引脚用作通用端口时，必须将对象单元通道对应的串行输出寄存器 m (SOM) 的 SOMn 位、串行输出允许寄存器 m (SOEm) 的 SOEmn 位和串行通道允许状态寄存器 m (SEm) 的 SEmn 位置初始值 (mn=02)。

备注 × : 忽略
 PM0x : 端口模式寄存器 0
 PIM0x : 端口输入模式寄存器 0
 POM0x : 端口输出模式寄存器 0
 PMC0x : 端口模式控制寄存器 0

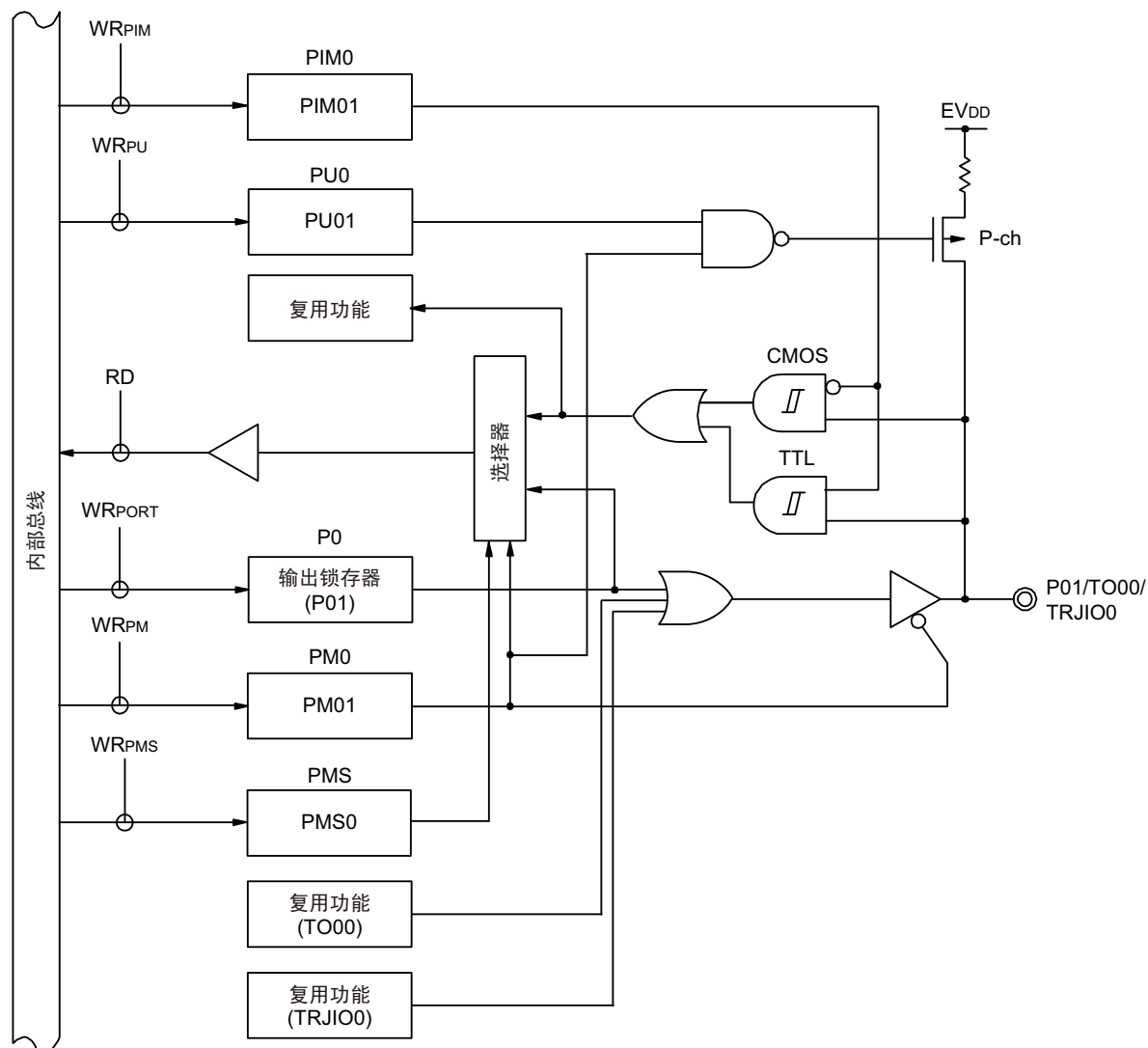
例如，64 引脚产品在 PIOR0 为“00H”并且 PIOR1 为“00H”时的端口 0 框图如图 4-1～图 4-7 所示。

图 4-1 P00 的框图



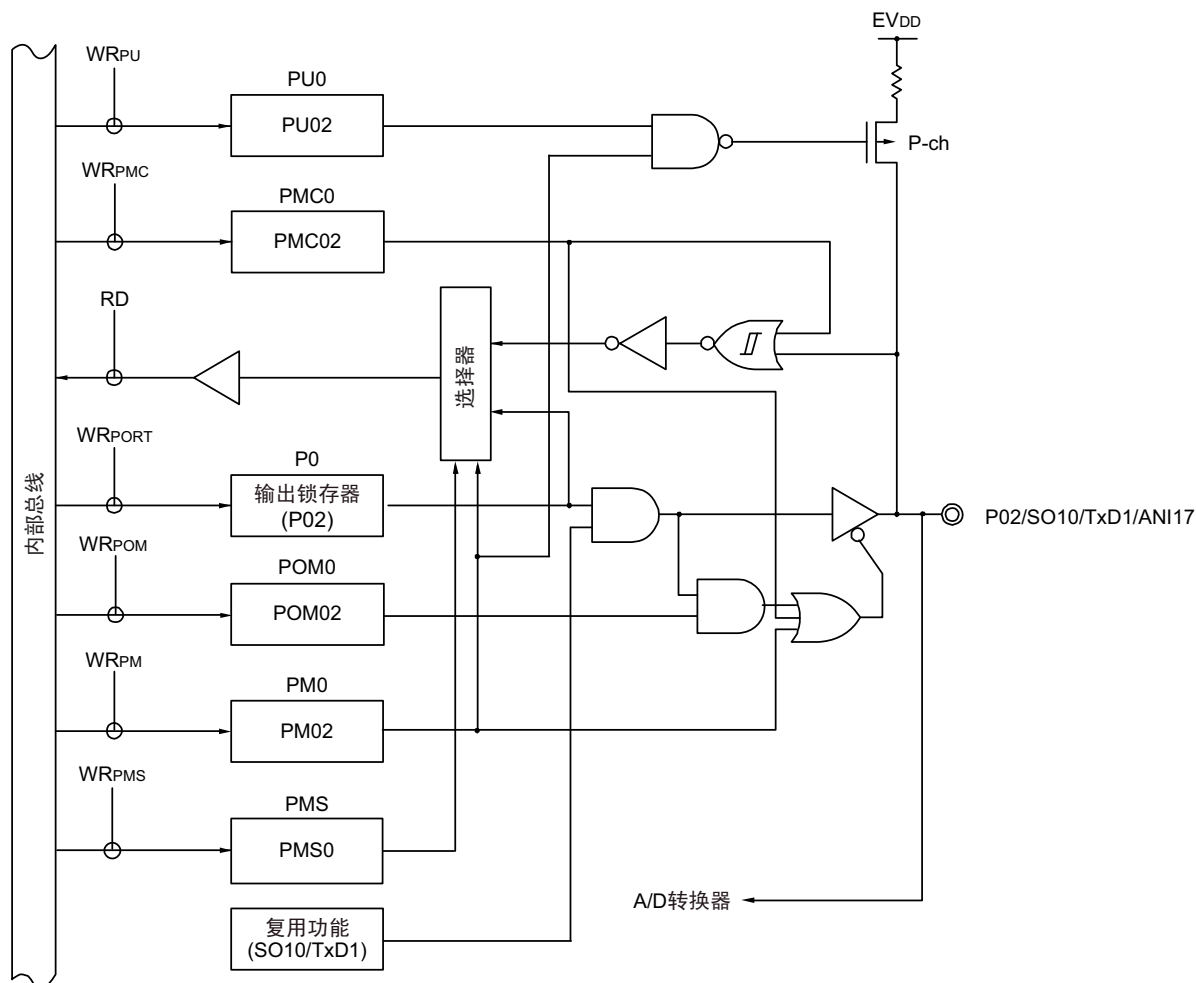
P0 : 端口寄存器 0
 PU0 : 上拉电阻选择寄存器 0
 PM0 : 端口模式寄存器 0
 POM0 : 端口输出模式寄存器 0
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-2 P01 的框图



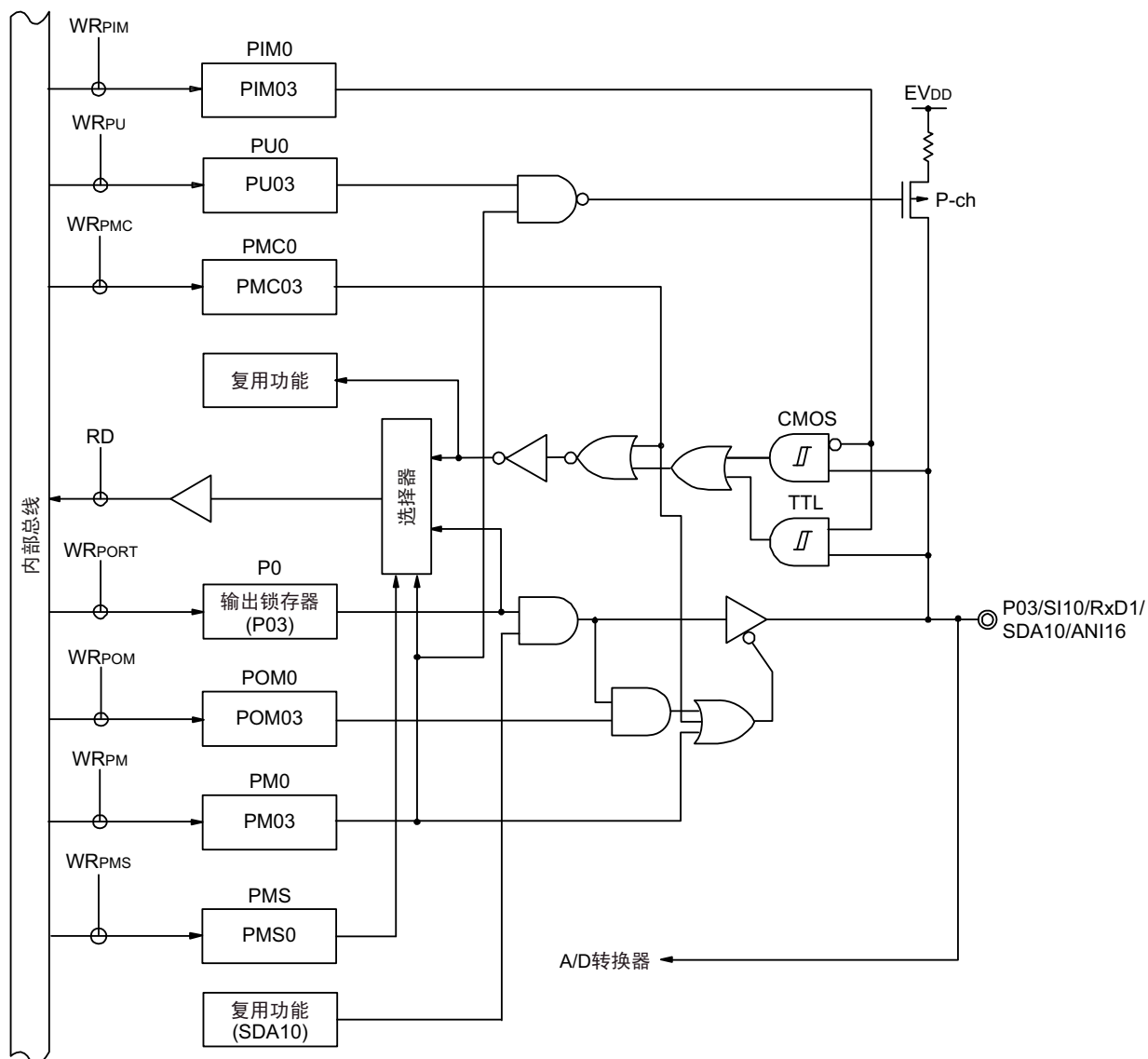
- P0 : 端口寄存器 0
 PU0 : 上拉电阻选择寄存器 0
 PM0 : 端口模式寄存器 0
 PIM0 : 端口输入模式寄存器 0
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-3 P02 的框图



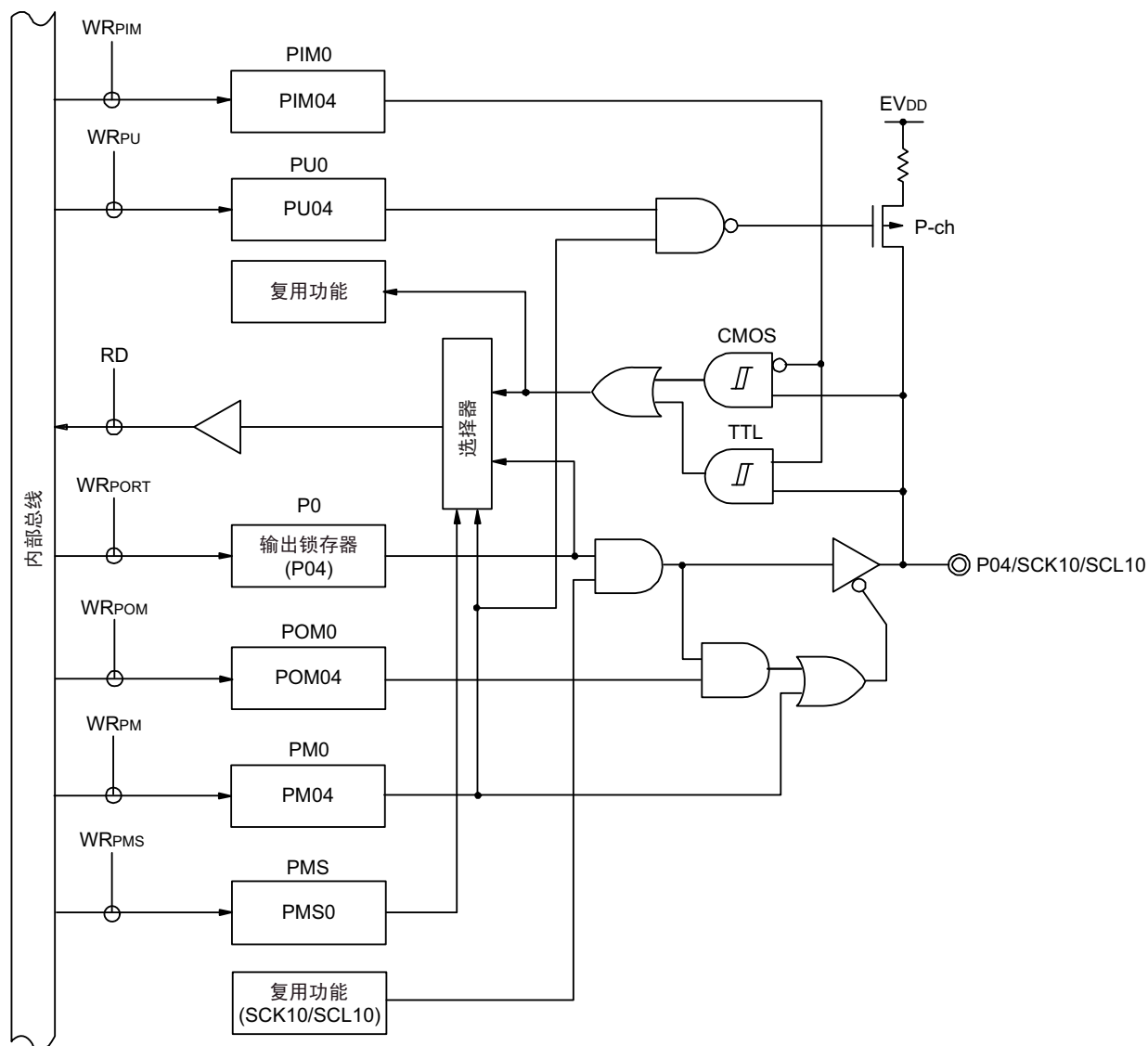
- P0 : 端口寄存器 0
 PU0 : 上拉电阻选择寄存器 0
 PM0 : 端口模式寄存器 0
 POM0 : 端口输出模式寄存器 0
 PMC0 : 端口模式控制寄存器 0
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-4 P03 的框图



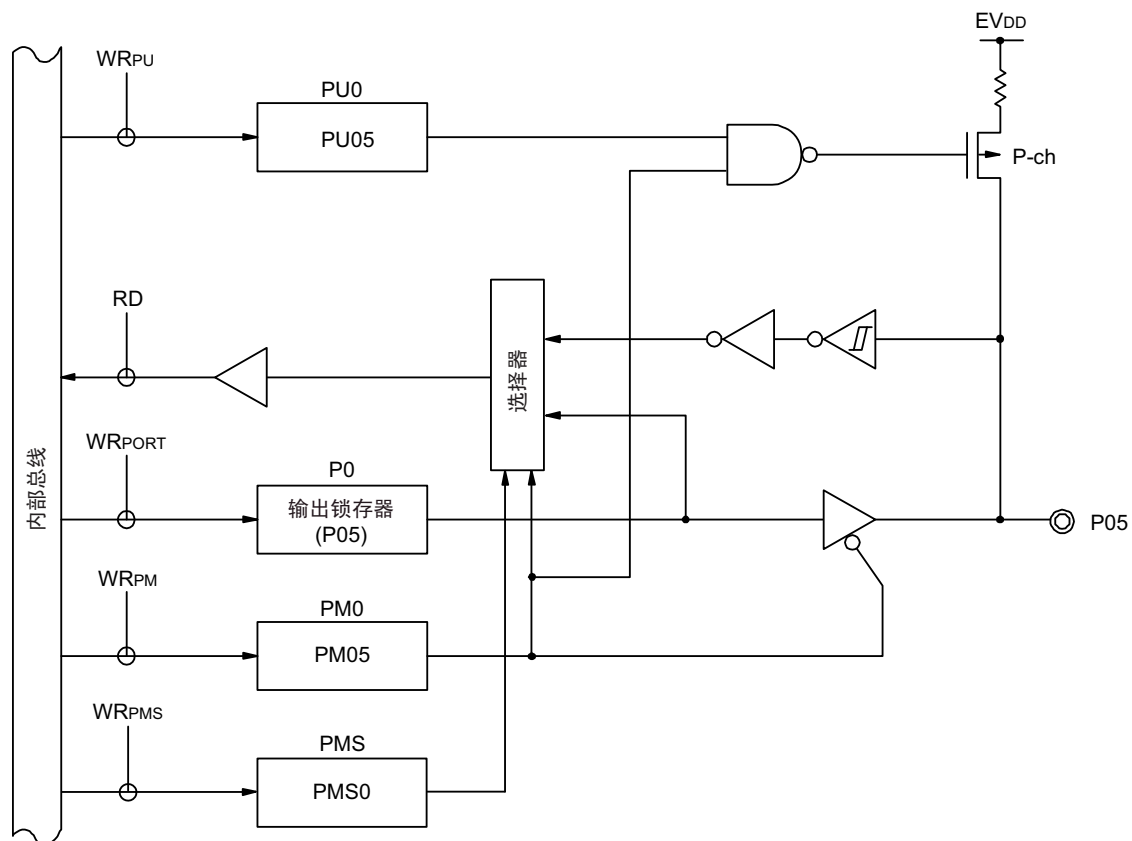
- P0 : 端口寄存器 0
 PU0 : 上拉电阻选择寄存器 0
 PM0 : 端口模式寄存器 0
 PIM0 : 端口输入模式寄存器 0
 POM0 : 端口输出模式寄存器 0
 PMC0 : 端口模式控制寄存器 0
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-5 P04 的框图



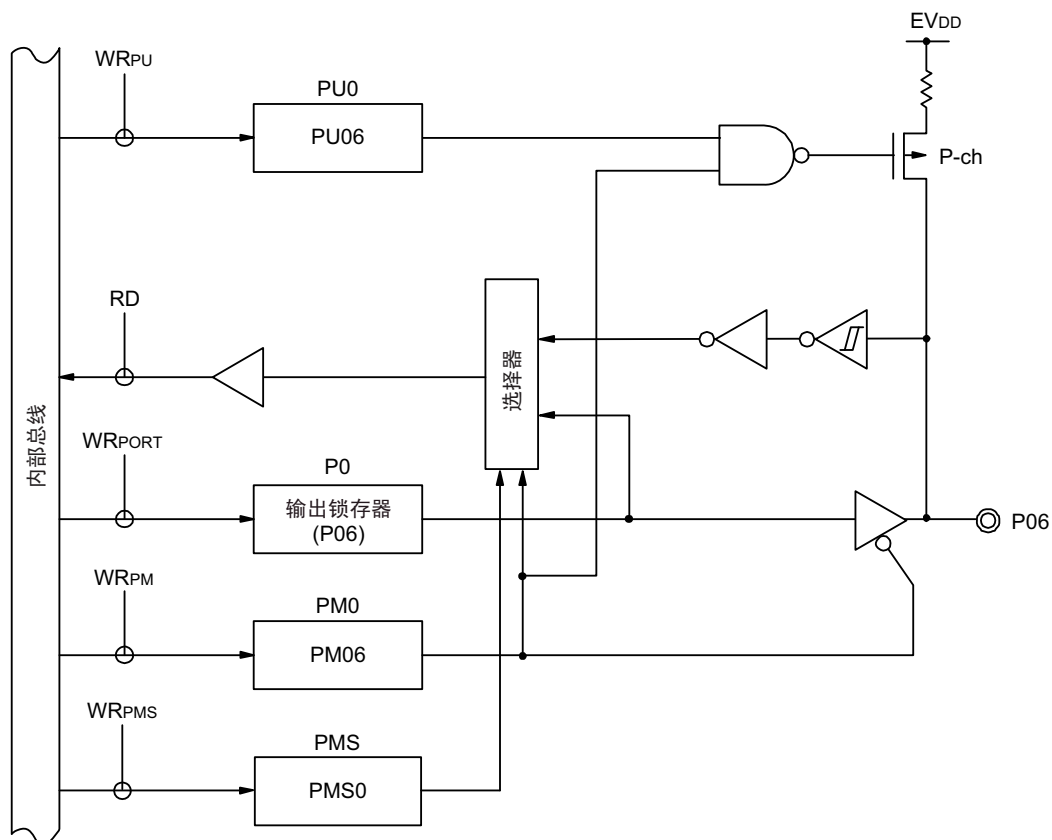
- P0 : 端口寄存器 0
 PU0 : 上拉电阻选择寄存器 0
 PM0 : 端口模式寄存器 0
 PIM0 : 端口输入模式寄存器 0
 POM0 : 端口输出模式寄存器 0
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-6 P05 的框图



- P0 : 端口寄存器 0
 PU0 : 上拉电阻选择寄存器 0
 PM0 : 端口模式寄存器 0
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-7 P06 的框图



- P0 : 端口寄存器 0
 PU0 : 上拉电阻选择寄存器 0
 PM0 : 端口模式寄存器 0
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

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 引脚和 P14 ~ 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、P11、P13 ~ P15、P17 引脚的输出设定为 N 沟道漏极开路输出（EV_{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、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。

表 4-3 使用端口 1 时的寄存器设定 (1/2)

引脚名		PM1x	PIM1x	POM1x	复用功能的设定	备注
名称	输入 / 输出					
P10	输入	1	0	×	×	CMOS 输入
		1	1	×	×	TTL 输入
	输出	0	×	0	SCK11/SCL11 输出 =1 注 1 TRDIOD1 输出 =0 注 2	CMOS 输出
		0	×	1		N 沟道漏极开路输出
P11	输入	1	—	×	×	
	输出	0		0	SDA11 输出 =1 注 3 TRDIOC1 输出 =0 注 2	CMOS 输出
		0		1		N 沟道漏极开路输出
P12	输入	1	—	—	×	
	输出	0			SO11 输出 =1 注 3 TRDIOB1 输出 =0 注 2	
P13	输入	1	—	×	×	
	输出	0		0	TxD2/SO20 输出 =1 注 3 TRDIOA1 输出 =0 注 2	CMOS 输出
		0		1		N 沟道漏极开路输出
P14	输入	1	0	×	×	CMOS 输入
		1	1	×	×	TTL 输入
	输出	0	×	0	SDA20 输出 =1 注 3 TRDIOD0 输出 =0 注 2 (SCLA0 输出 =0 注 4)	CMOS 输出
		0	×	1		N 沟道漏极开路输出
P15	输入	1	0	×	×	CMOS 输入
		1	1	×	×	TTL 输入
	输出	0	×	0	PCLBUZ1 输出 =0 注 5 SCK20/SCL20 输出 =1 注 1 TRDIOB0 输出 =0 注 2 (SDAA0 输出 =0 注 4)	CMOS 输出
		0	×	1		N 沟道漏极开路输出

表 4-3 使用端口 1 时的寄存器设定 (2/2)

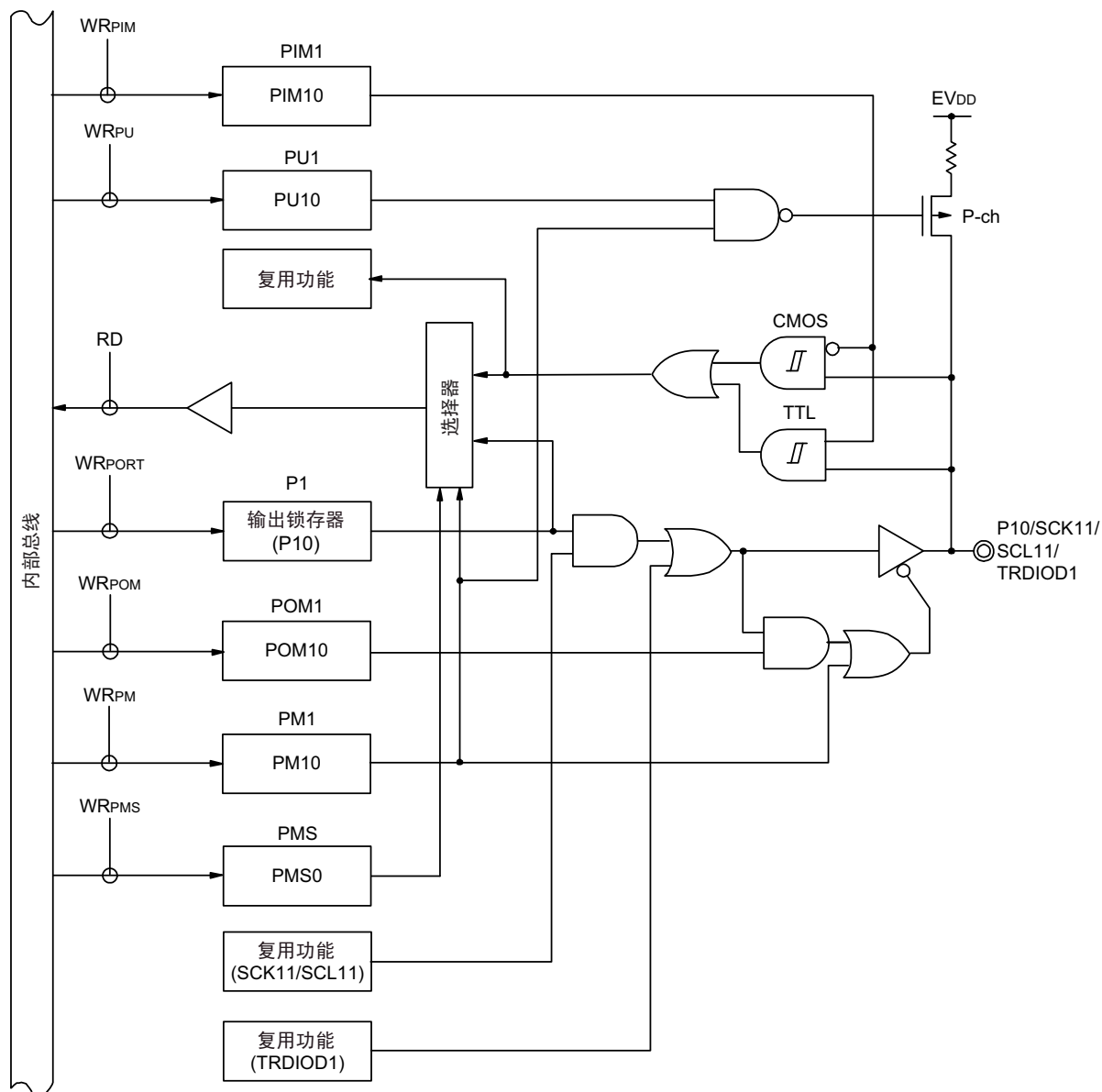
引脚名		PM1x	PIM1x	POM1x	复用功能的设定	备注
名称	输入 / 输出					
P16	输入	1	0	—	×	CMOS 输入
	输出	1	1		×	TTL 输入
		0	×		TO01 输出 =0 注 6 TRDIOC0 输出 =0 注 2	
P17	输入	1	0	×	×	CMOS 输入
		1	1	×	×	TTL 输入
	输出	0	×	0	TO02 输出 =0 注 6 TRDIOA0 输出 =0 注 2	CMOS 输出
		0	×	1	(SO00/TxD0 输出 =1 注 7)	N 沟道漏极开路输出

- 注 1. 当将串行阵列单元功能的复用引脚用作通用端口时, 必须将对象单元通道对应的串行输出寄存器 m (SOM) 的 CKOmn 位、串行输出允许寄存器 m (SOEm) 的 SOEmn 位和串行通道允许状态寄存器 m (SEm) 的 SEMn 位置初始值 (mn=03、10)。
2. 当将定时器 RD 功能的复用引脚用作通用端口时, 必须通过定时器 RD 的输出主控允许寄存器 1 (TRDOER1) 将对象 TRDIOij 引脚的输出控制位置初始值 (i=A、B、C、D, j=0、1)。
3. 当将串行阵列单元功能的复用引脚用作通用端口时, 必须将对象单元通道对应的串行输出寄存器 m (SOM) 的 SOMn 位、串行输出允许寄存器 m (SOEm) 的 SOEmn 位和串行通道允许状态寄存器 m (SEm) 的 SEMn 位置初始值 (mn=03、10)。
4. 当外围 I/O 重定向寄存器 0 (PIOR0) 的 PIOR02 位为“1”并且将串行接口 IICA 功能的复用引脚用作通用端口时, 必须停止对应的串行接口 IICA 的运行。
5. 在 32 引脚产品的情况下, 当将时钟 / 蜂鸣器输出功能的复用引脚用作通用端口时, 必须将时钟输出选择寄存器 i (CKSi) 的 PCLOEi 位置初始值 (i=1)。
6. 当将定时器阵列单元的定时器输出功能的复用引脚用作通用端口时, 必须将对象单元通道对应的定时器输出寄存器 m (TOM) 的 TOMn 位和定时器输出允许寄存器 m (TOEm) 的 TOEmn 位置初始值 (m=0, n=1、2)。
7. 当外围 I/O 重定向寄存器 0 (PIOR0) 的 PIOR01 位为“1”并且将串行阵列单元功能的复用引脚用作通用端口时, 必须将对象单元通道对应的串行输出寄存器 m (SOM) 的 SOMn 位、串行输出允许寄存器 m (SOEm) 的 SOEmn 位和串行通道允许状态寄存器 m (SEm) 的 SEMn 位置初始值 (mn=00)。
8. 能通过设定外围 I/O 重定向寄存器 0、1 (PIOR0、PIOR1), 分配 () 内的功能。

备注 × : 忽略
 PM1x : 端口模式寄存器 1
 PIM1x : 端口输入模式寄存器 1
 POM1x : 端口输出模式寄存器 1

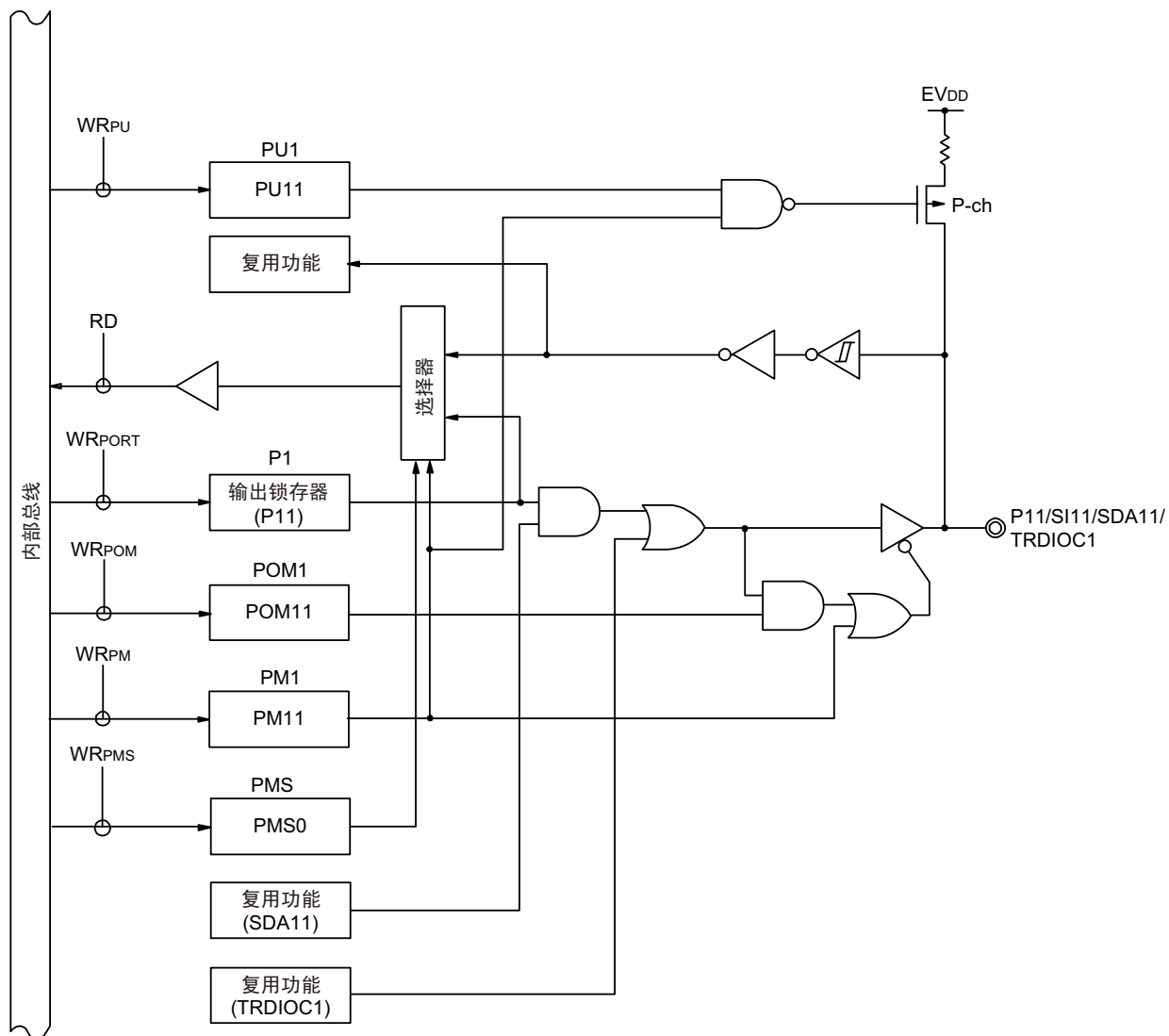
例如，64 引脚产品在 P1OR0 为“00H”并且 P1OR1 为“00H”时的端口 1 框图如图 4-8 ~图 4-15 所示。

图 4-8 P10 的框图



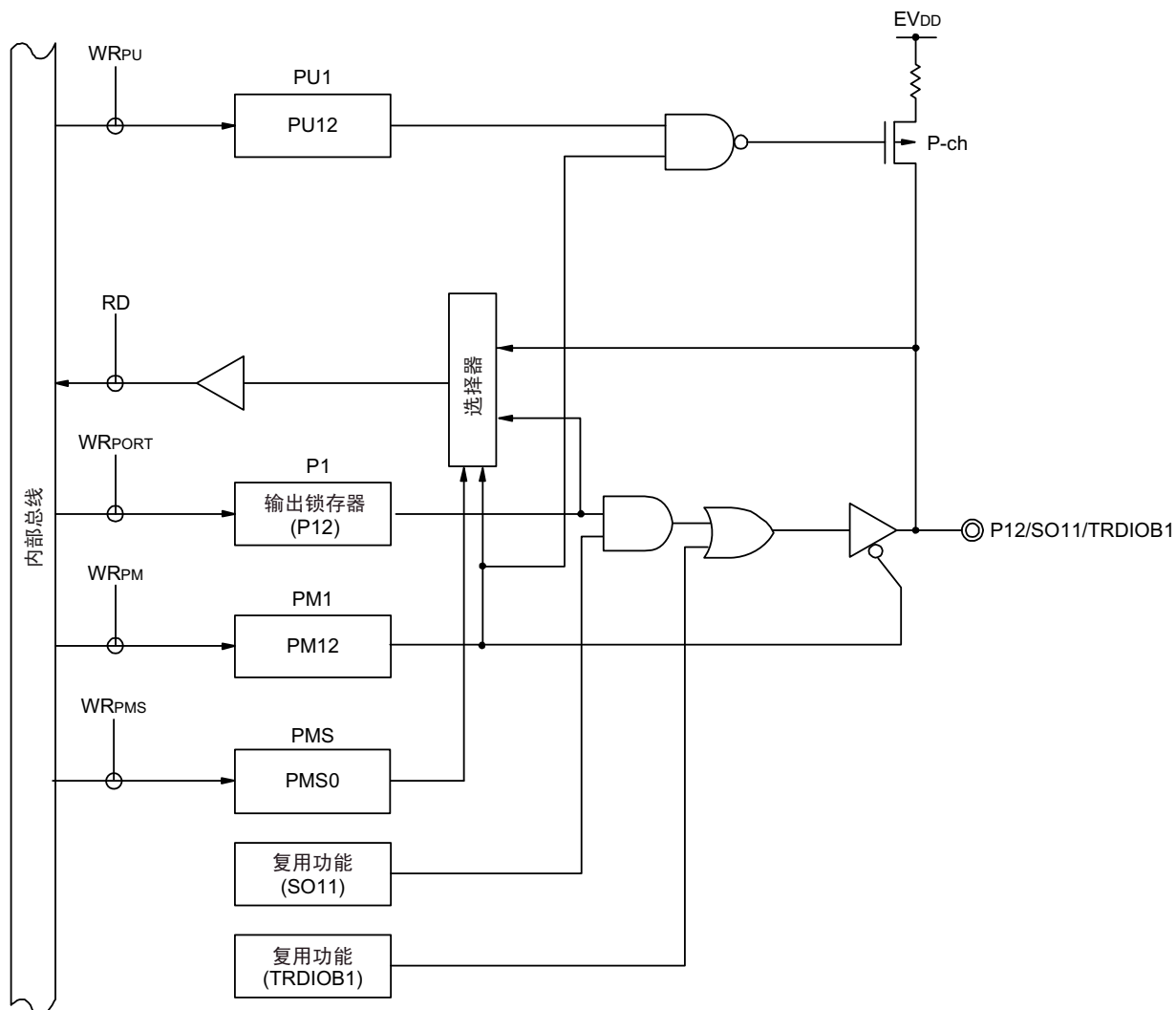
- P1 : 端口寄存器 1
- PU1 : 上拉电阻选择寄存器 1
- PM1 : 端口模式寄存器 1
- PIM1 : 端口输入模式寄存器 1
- POM1 : 端口输出模式寄存器 1
- PMS : 端口模式选择寄存器
- RD : 读信号
- WRxx : 写信号

图 4-9 P11 的框图



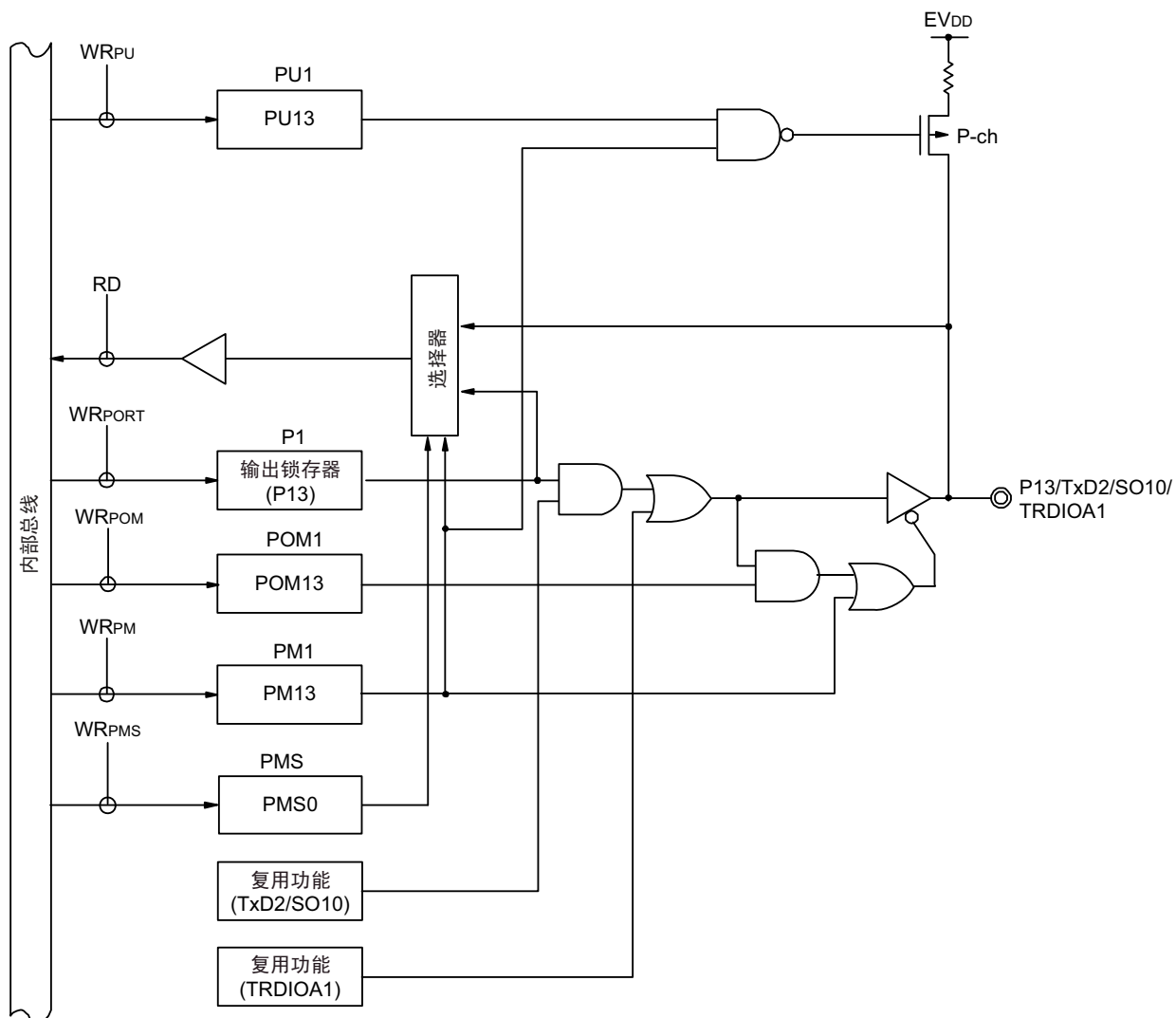
- P1 : 端口寄存器 1
 PU1 : 上拉电阻选择寄存器 1
 PM1 : 端口模式寄存器 1
 POM1 : 端口输出模式寄存器 1
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-10 P12 的框图



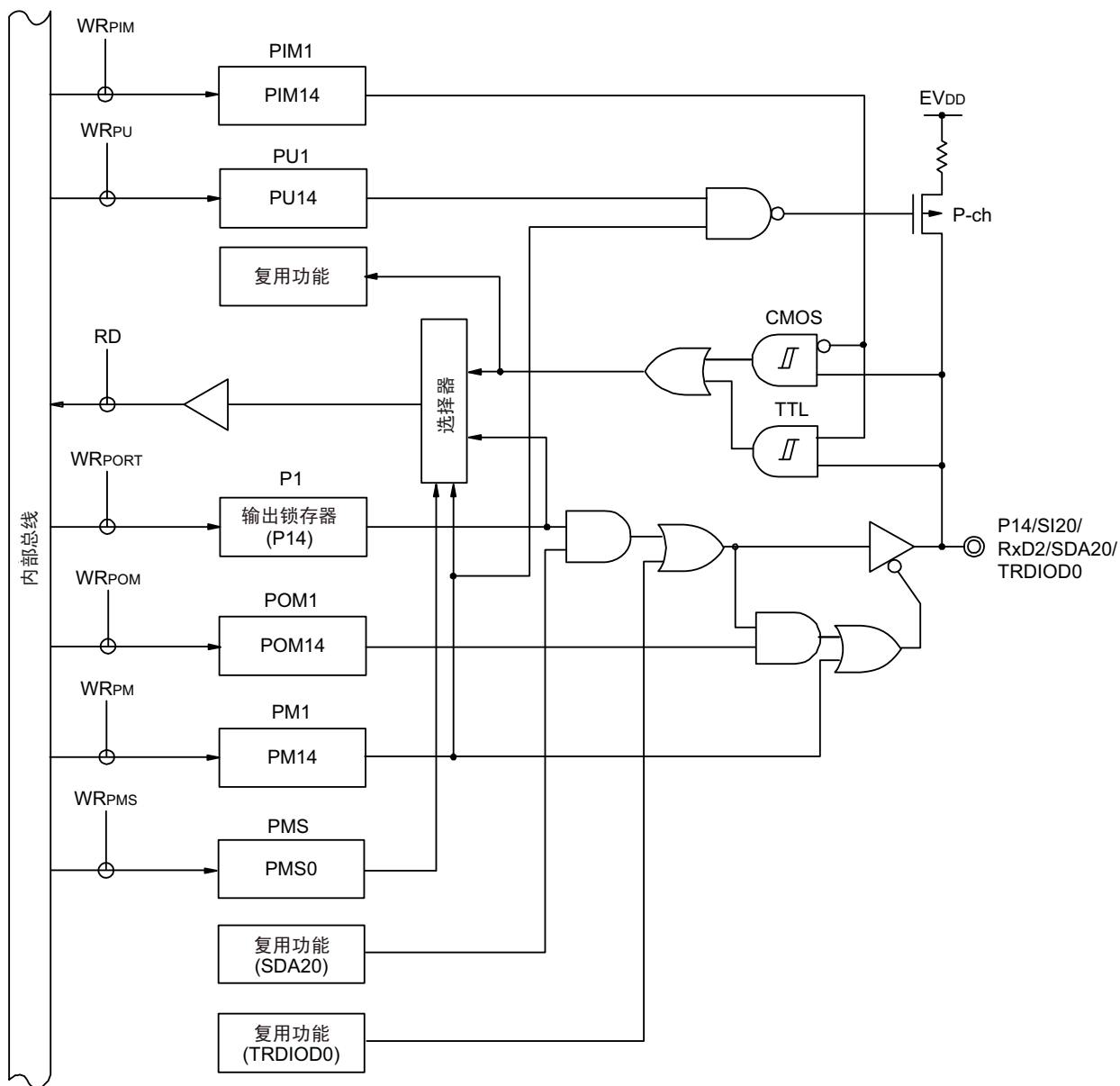
- P1 : 端口寄存器 1
 PU1 : 上拉电阻选择寄存器 1
 PM1 : 端口模式寄存器 1
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-11 P13 的框图



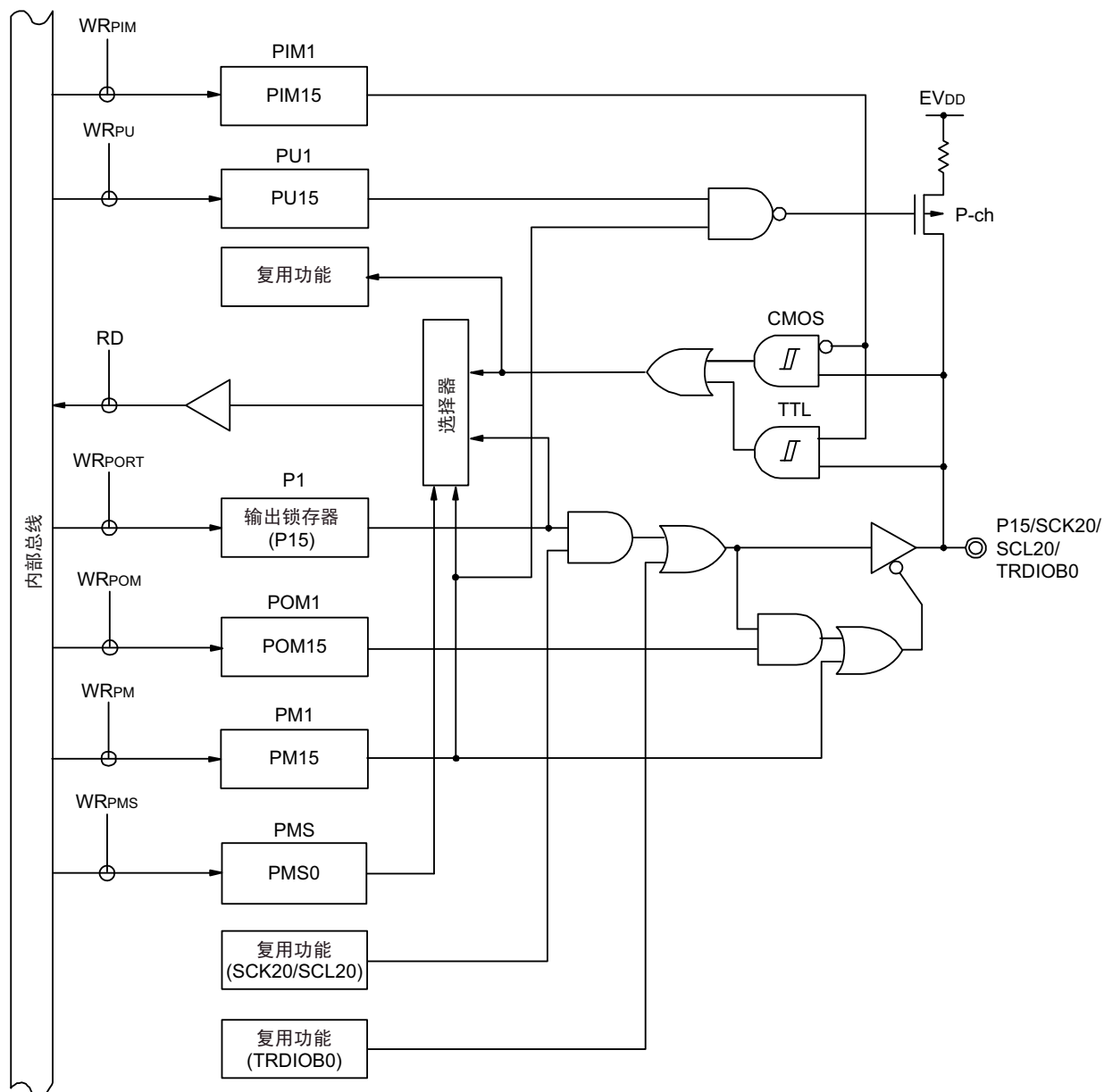
- P1 : 端口寄存器 1
 PU1 : 上拉电阻选择寄存器 1
 PM1 : 端口模式寄存器 1
 POM1 : 端口输出模式寄存器 1
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-12 P14 的框图



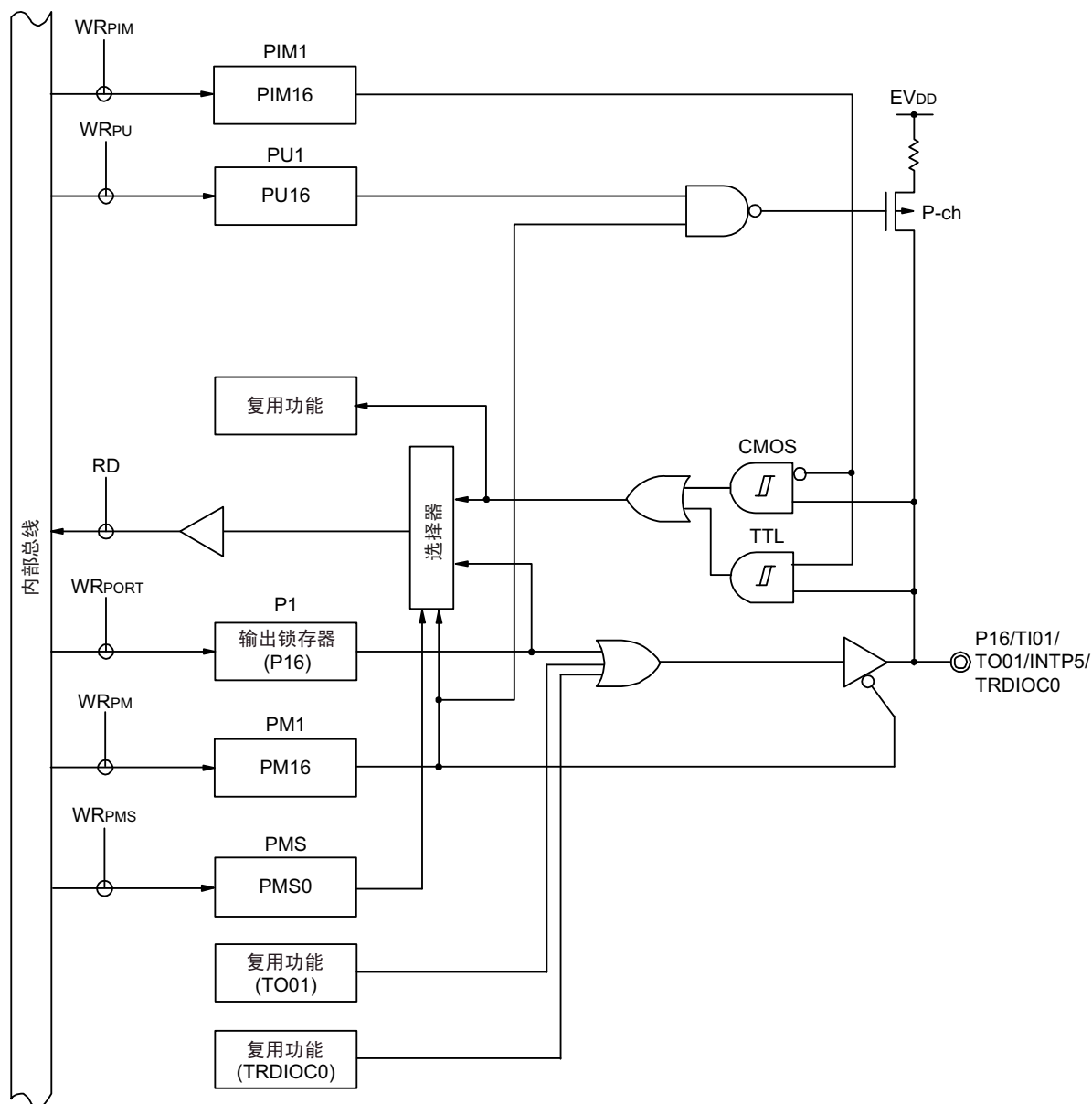
- P1 : 端口寄存器 1
 PU1 : 上拉电阻选择寄存器 1
 PM1 : 端口模式寄存器 1
 PIM1 : 端口输入模式寄存器 1
 POM1 : 端口输出模式寄存器 1
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-13 P15 的框图



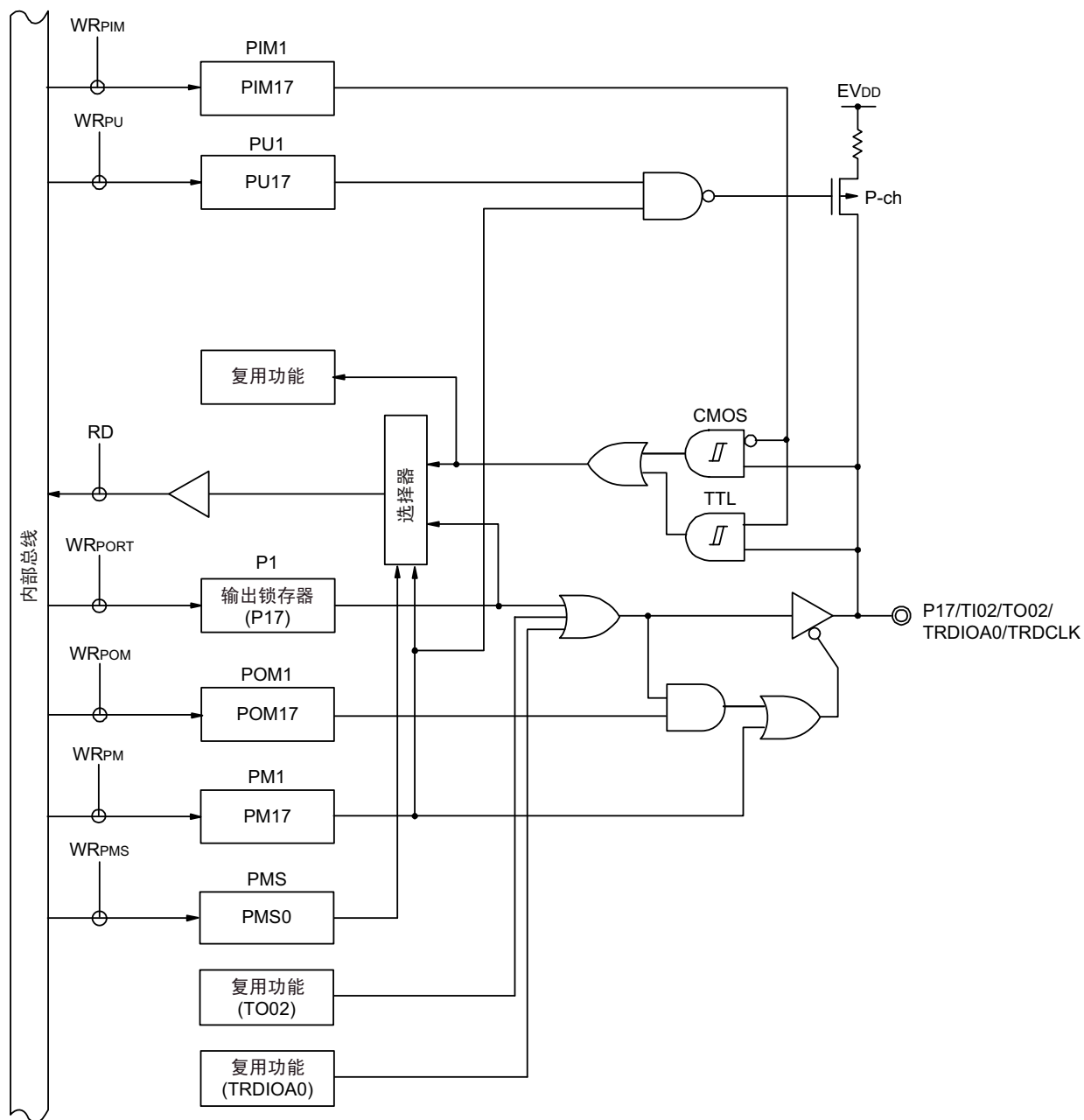
- P1 : 端口寄存器 1
 PU1 : 上拉电阻选择寄存器 1
 PM1 : 端口模式寄存器 1
 PIM1 : 端口输入模式寄存器 1
 POM1 : 端口输出模式寄存器 1
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-14 P16 的框图



- P1 : 端口寄存器 1
 PU1 : 上拉电阻选择寄存器 1
 PM1 : 端口模式寄存器 1
 PIM1 : 端口输入模式寄存器 1
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-15 P17 的框图



- P1 : 端口寄存器 1
 PU1 : 上拉电阻选择寄存器 1
 PM1 : 端口模式寄存器 1
 PIM1 : 端口输入模式寄存器 1
 POM1 : 端口输出模式寄存器 1
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

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 转换器的基准电位（+）输入和 A/D 转换器的基准电位（-）输入。

当将 P20/ANI0 ~ P27/ANI7 用作数字输入时，必须在通过 A/D 端口配置寄存器（ADPC）设定为数字输入 / 输出并且通过 PM2 寄存器设定为输入模式后从高位开始使用。

当将 P20/ANI0 ~ P27/ANI7 用作数字输出时，必须通过 ADPC 寄存器设定为数字输入 / 输出并且通过 PM2 寄存器设定为输出模式。

当将 P20/ANI0 ~ P27/ANI7 用作模拟输入 / 输出时，必须在通过 A/D 端口配置寄存器（ADPC）设定为模拟输入 / 输出并且通过 PM2 寄存器设定为输入模式后从低位开始使用。

表 4-4 使用端口 2 时的寄存器设定

引脚名		PM2x	ADPC	复用功能的设定	备注
名称	输入 / 输出				
P2n	输入	1	01 ~ n+1H	—	当用作端口时，从高位开始使用。
	输出	0	01 ~ n+1H		

备注 1. PM2x : 端口模式寄存器 2
 ADPC : A/D 端口配置寄存器
 2. n=0 ~ 7

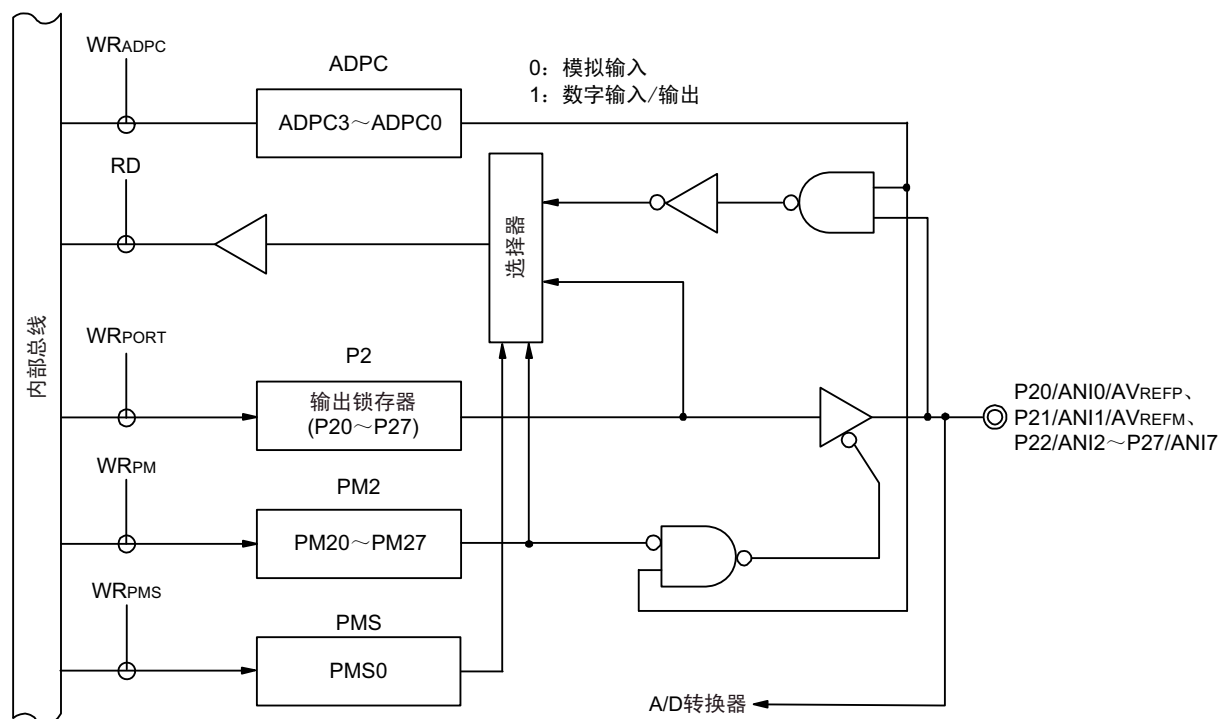
表 4-5 P20/ANI0 ~ P27/ANI7 引脚功能的设定

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

在产生复位信号后，P20/ANI0 ~ P27/ANI7 都变为模拟输入。

例如，64 引脚产品的端口 2 框图如图 4-16 所示。

图 4-16 P20 ~ P27 的框图



- P2 : 端口寄存器 2
 PM2 : 端口模式寄存器 2
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

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。

能通过设定端口输入模式寄存器 3（PIM3），以位为单位给 P30 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 3（POM3），以位为单位将 P30 引脚的输出设定为 N 沟道漏极开路输出（ E_{VDD} 耐压）。

端口 3 作为复用功能，有外部中断的请求输入、实时时钟的校正时钟输出、串行接口的时钟输入 / 输出和定时器的输入 / 输出。

在产生复位信号后，端口 3 变为输入模式。

表 4-6 使用端口 3 时的寄存器设定

引脚名		PM3x	PIM3x	POM3x	复用功能的设定注 8	备注
名称	输入 / 输出					
P30	输入	1	0	×	×	CMOS 输入
		1	1	×	×	TTL 输入
	输出	0	×	0	RTC1HZ 输出 =0 注 1 SCK00/SCL00 输出 =1 注 2 TRJ00 输出 =0 注 3	CMOS 输出
		0	×	1		N 沟道漏极开路输出
P31	输入	1	—	—	×	CMOS 输入
	输出	0			TO03 输出 =0 注 4 PCLBUZ0 输出 =0 注 5 (PCLBUZ0 输出 =0 注 6) (TRJIO0 输出 =0 注 7)	

- 注 1. 在 64 引脚产品的情况下，当将 RTC1HZ 引脚的输出（1Hz）功能的复用引脚用作通用端口时，必须将实时时钟控制寄存器 0（RTCC0）的 RCLOE1 位置初始值。
2. 当将串行阵列单元功能的复用引脚用作通用端口时，必须将对象单元通道对应的串行输出寄存器 m（SOM）的 CKOm 位、串行输出允许寄存器 m（SOEm）的 SOEmn 位和串行通道允许状态寄存器 m（SEm）的 SEmn 位置初始值（mn=00）。
3. 当将定时器 RJ 的定时器输出功能的复用引脚用作通用端口时，必须通过定时器 RJ I/O 控制寄存器 0（TRJIOC0）的 bit2（TOENA）置初始值。
4. 当将定时器阵列单元的定时器输出功能的复用引脚用作通用端口时，必须将对象单元通道对应的定时器输出寄存器 m（TOM）的 TOMn 位和定时器输出允许寄存器 m（TOEm）的 TOEmn 位置初始值（m=0，n=4）。
5. 在 32 引脚产品的情况下，当将时钟 / 蜂鸣器输出功能的复用引脚用作通用端口时，必须将时钟输出选择寄存器 i（CKSi）的 PCLOEi 位置初始值（i=0）。
6. 在 64 引脚产品的情况下，当外围 I/O 重定向寄存器 0（PIOR0）的 PIOR03 位为“1”并且将时钟 / 蜂鸣器输出功能的复用引脚用作通用端口时，必须将时钟输出选择寄存器 i（CKSi）的 PCLOEi 位置初始值（i=0）。
7. 当外围 I/O 重定向寄存器 1（PIOR1）的 PIOR11 位和 PIOR10 位为“01B”并且将定时器 RJ 的定时器输入 / 输出功能的复用引脚用作通用端口时，必须将定时器 RJ 模式寄存器 0（TRJMR0）的 TMOD2 ~ 0 位置初始值或者“001B”以外的数据。
8. 能通过设定外围 I/O 重定向寄存器 0、1（PIOR0、PIOR1），分配（）内的功能。

备注 × : 忽略

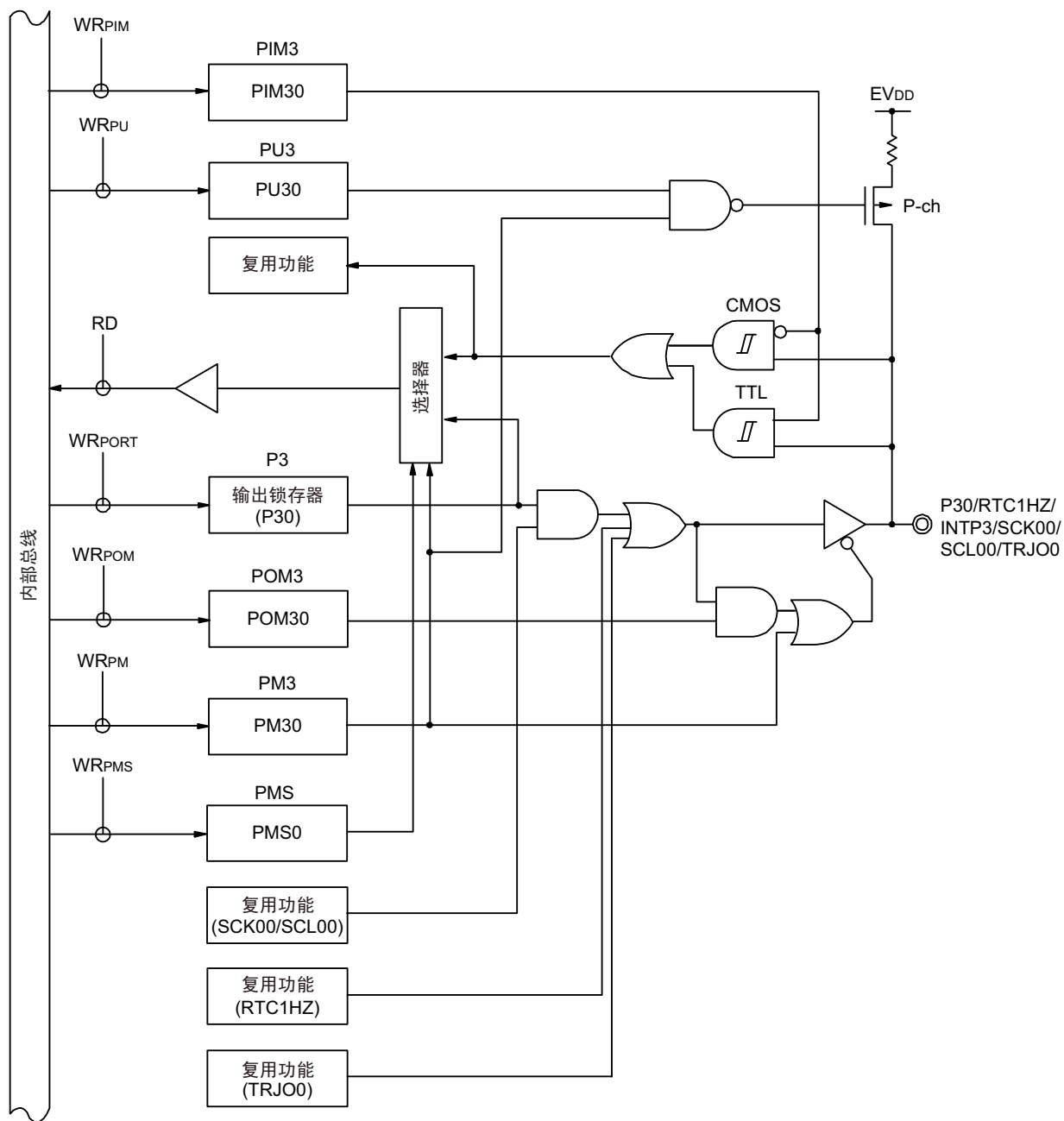
PM3x : 端口模式寄存器 3

PIM3x : 端口输入模式寄存器 3

POM3x : 端口输出模式寄存器 3

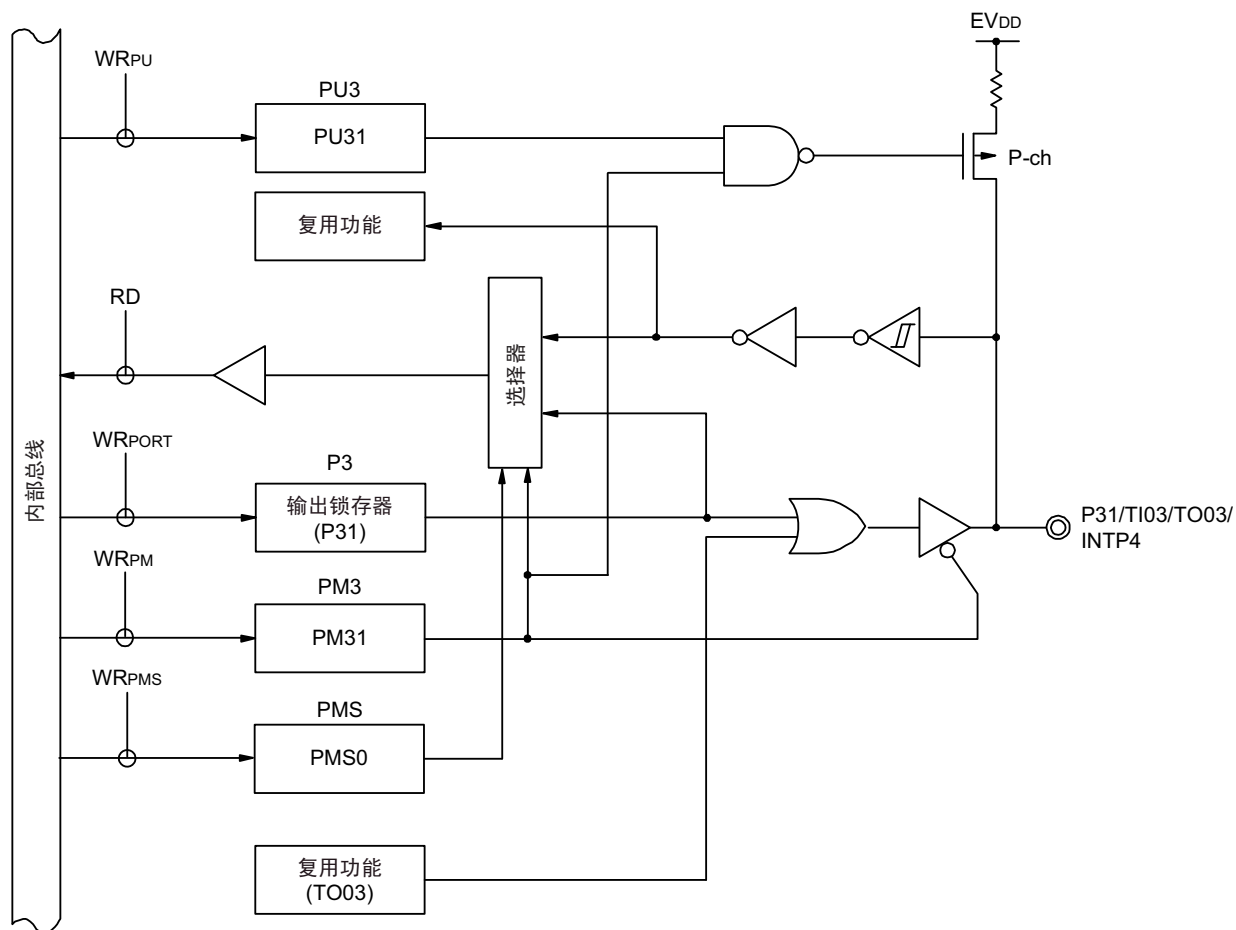
例如，64 引脚产品在 PIOR1 为“00H”时的端口 3 框图如图 4-17 和图 4-18 所示。

图 4-17 P30 的框图



- P3 : 端口寄存器 3
- PU3 : 上拉电阻选择寄存器 3
- PM3 : 端口模式寄存器 3
- PIM3 : 端口输入模式寄存器 3
- POM3 : 端口输出模式寄存器 3
- PMS : 端口模式选择寄存器
- RD : 读信号
- WRxx : 写信号

图 4-18 P31 的框图



- P3 : 端口寄存器 3
 PU3 : 上拉电阻选择寄存器 3
 PM3 : 端口模式寄存器 3
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

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 ~ P43 引脚用作输入端口时，能通过上拉电阻选择寄存器 4（PU4）以位为单位使用内部上拉电阻。

端口 4 作为复用功能，有用于闪存编程器 / 调试器的数据输入 / 输出。

在产生复位信号后，端口 4 变为输入模式。

表 4-7 使用端口 4 时的寄存器设定

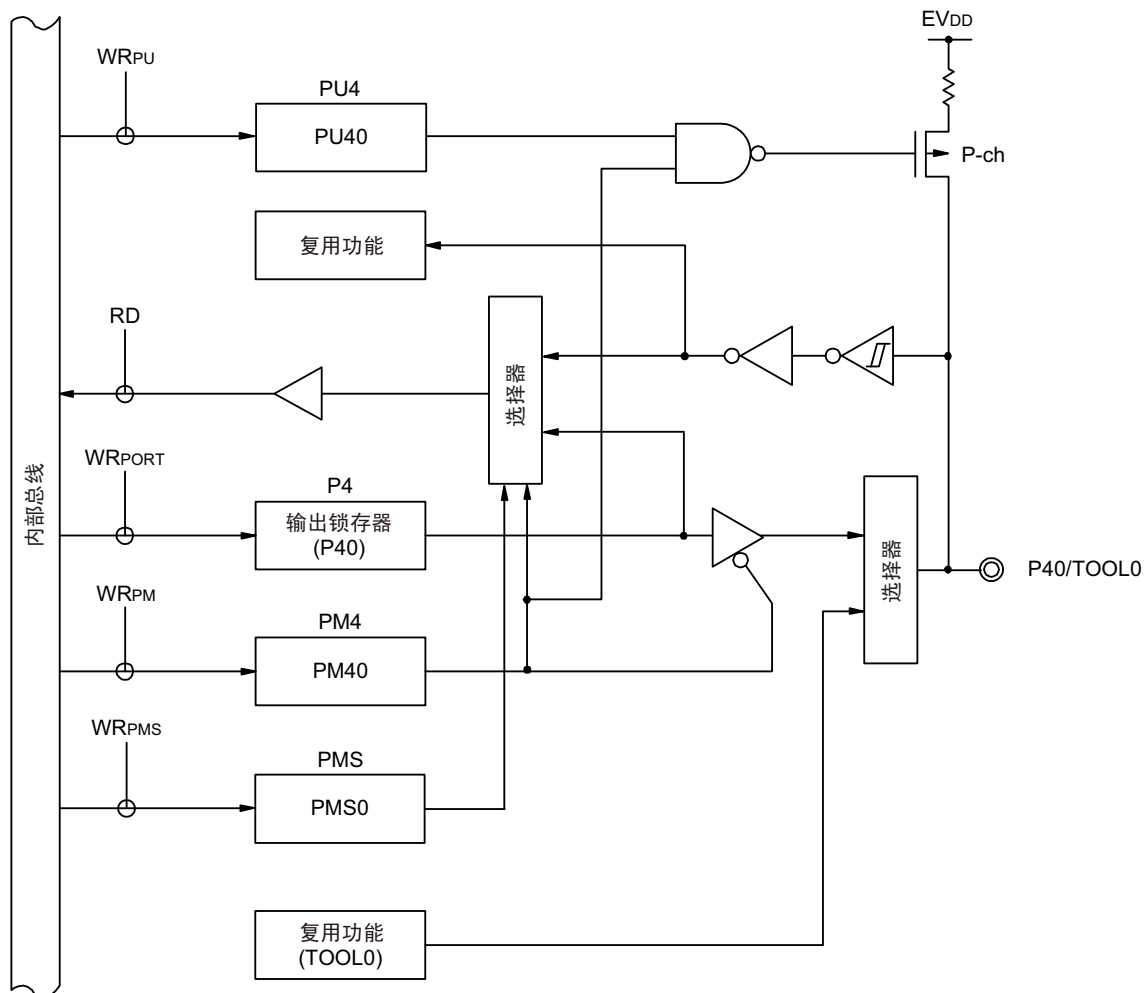
引脚名		PM4x	复用功能的设定	备注
名称	输入 / 输出			
P40 ~ P43	输入	1	×	
	输出	0	×	

注意 在连接工具时，不能将 P40 引脚用作端口引脚。

备注 × : 忽略
PM4x : 端口模式寄存器 4

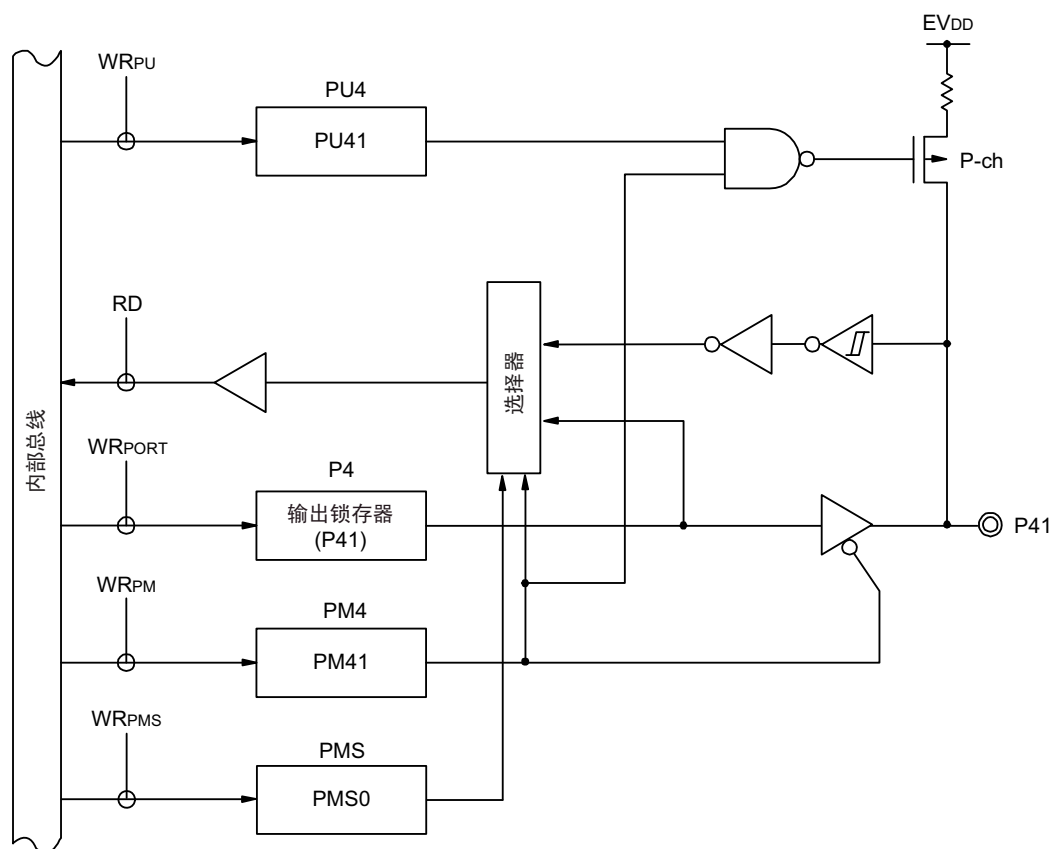
例如，64 引脚产品的端口 4 框图如图 4-19 ~图 4-22 所示。

图 4-19 P40 的框图



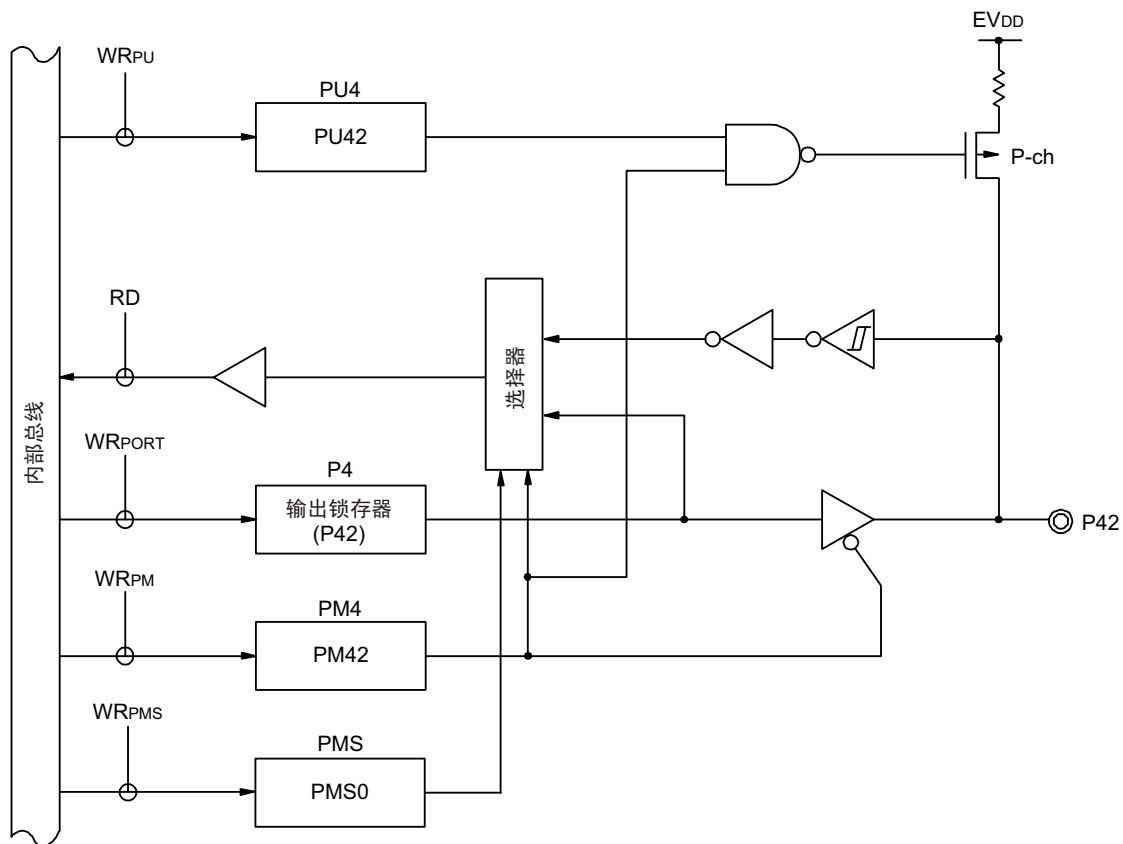
- P4 : 端口寄存器 4
 PU4 : 上拉电阻选择寄存器 4
 PM4 : 端口模式寄存器 4
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-20 P41 的框图



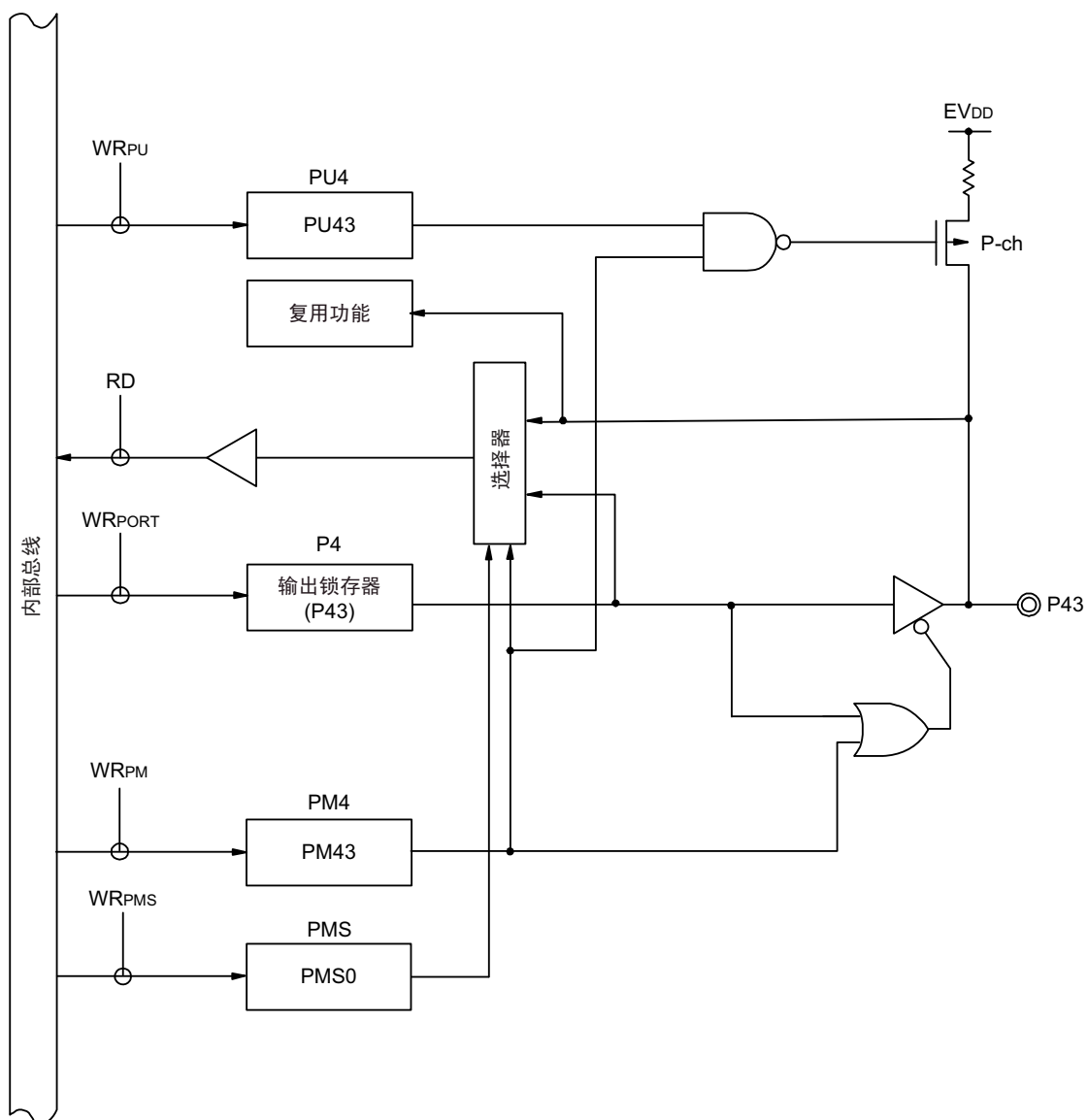
- P4 : 端口寄存器 4
 PU4 : 上拉电阻选择寄存器 4
 PM4 : 端口模式寄存器 4
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-21 P42 的框图



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

图 4-22 P43 的框图



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

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 ~ P55 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。

能通过设定端口输入模式寄存器 5（PIM5），以位为单位给 P50 引脚和 P51 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 5（POM5），以位为单位将 P50、P51、P55 引脚的输出设定为 N 沟道漏极开路输出（EV_{DD} 耐压）。

端口 5 作为复用功能，有外部中断的请求输入、串行接口的数据输入 / 输出、时钟的输入 / 输出以及编程 UART 的发送和接收。

在产生复位信号后，端口 5 变为输入模式。

表 4-8 使用端口 5 时的寄存器设定

引脚名		PM5x	PIM5x	POM5x	复用功能的设定 ^{注 5}	备注
名称	输入 / 输出					
P50	输入	1	0	×	×	CMOS 输入
		1	1	×	×	TTL 输入
	输出	0	×	0	SDA00 输出 =1 ^{注 1}	CMOS 输出
		0	×	1		N 沟道漏极开路输出
P51	输入	1	—	×	×	
	输出	0		0	SO00/TxD0 输出 =1 ^{注 2}	CMOS 输出
		0		1		N 沟道漏极开路输出
P52	输入	1	—	—	×	
	输出	0			×	
P53	输入	1	—	—	×	
	输出	0			×	
P54	输入	1	—	—	×	
	输出	0			×	
P55	输入	1	0	×	×	CMOS 输入
		1	1	×	×	TTL 输入
	输出	0	×	0	(SCK00 输出 =1 ^{注 4})	CMOS 输出
		0	×	1	(PCLBUZ1 输出 =0 ^{注 3})	N 沟道漏极开路输出

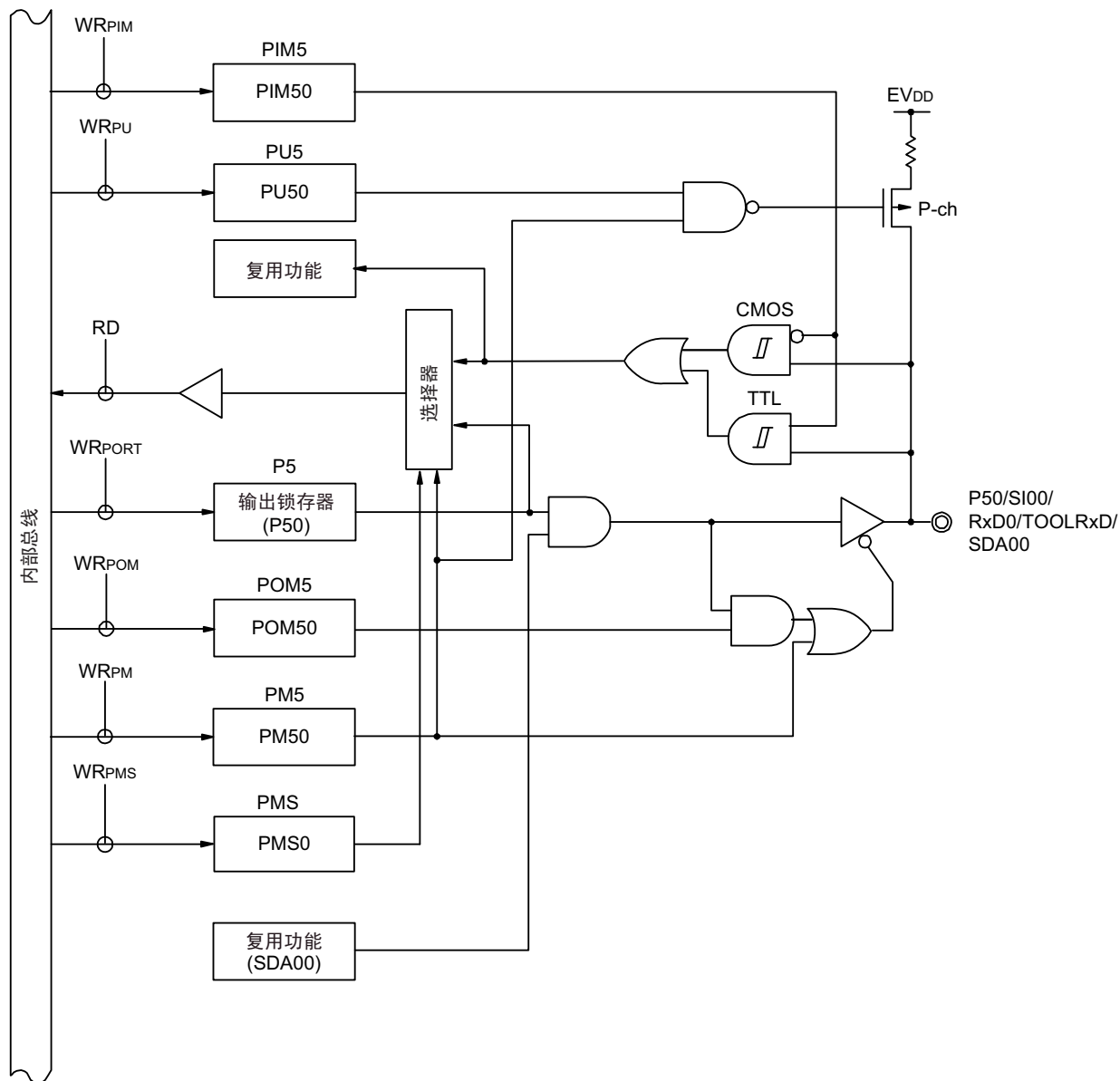
- 注
1. 当将串行阵列单元功能的复用引脚用作通用端口时, 必须将对象单元通道对应的串行输出寄存器 m (SOM) 的 CKOmn 位、串行输出允许寄存器 m (SOEm) 的 SOEmn 位和串行通道允许状态寄存器 m (SEm) 的 SEmn 位置初始值 (mn=03、10)。
 2. 当将串行阵列单元功能的复用引脚用作通用端口时, 必须将对象单元通道对应的串行输出寄存器 m (SOM) 的 SOMn 位、串行输出允许寄存器 m (SOEm) 的 SOEmn 位和串行通道允许状态寄存器 m (SEm) 的 SEmn 位置初始值 (mn=03、10)。
 3. 当外围 I/O 重定向寄存器 0 (PIOR0) 的 PIOR04 位为“1”并且将时钟 / 蜂鸣器输出功能的复用引脚用作通用端口时, 必须将时钟输出选择寄存器 i 的 PCLOEi 位置初始值 (i=1)。
 4. 当外围 I/O 重定向寄存器 0 (PIOR0) 的 PIOR01 位为“1”并且将串行阵列单元功能的复用引脚用作通用端口时, 必须将对象单元通道对应的串行输出寄存器 m (SOM) 的 SOMn 位、串行输出允许寄存器 m (SOEm) 的 SOEmn 位和串行通道允许状态寄存器 m (SEm) 的 SEmn 位置初始值 (mn=00)。
 5. 能通过设定外围 I/O 重定向寄存器 0、1 (PIOR0、PIOR1), 分配 () 内的功能。

备注

×	: 忽略
PM5x	: 端口模式寄存器 5
PIM5x	: 端口输入模式寄存器 5
POM5x	: 端口输出模式寄存器 5
PMC5x	: 端口模式控制寄存器 5

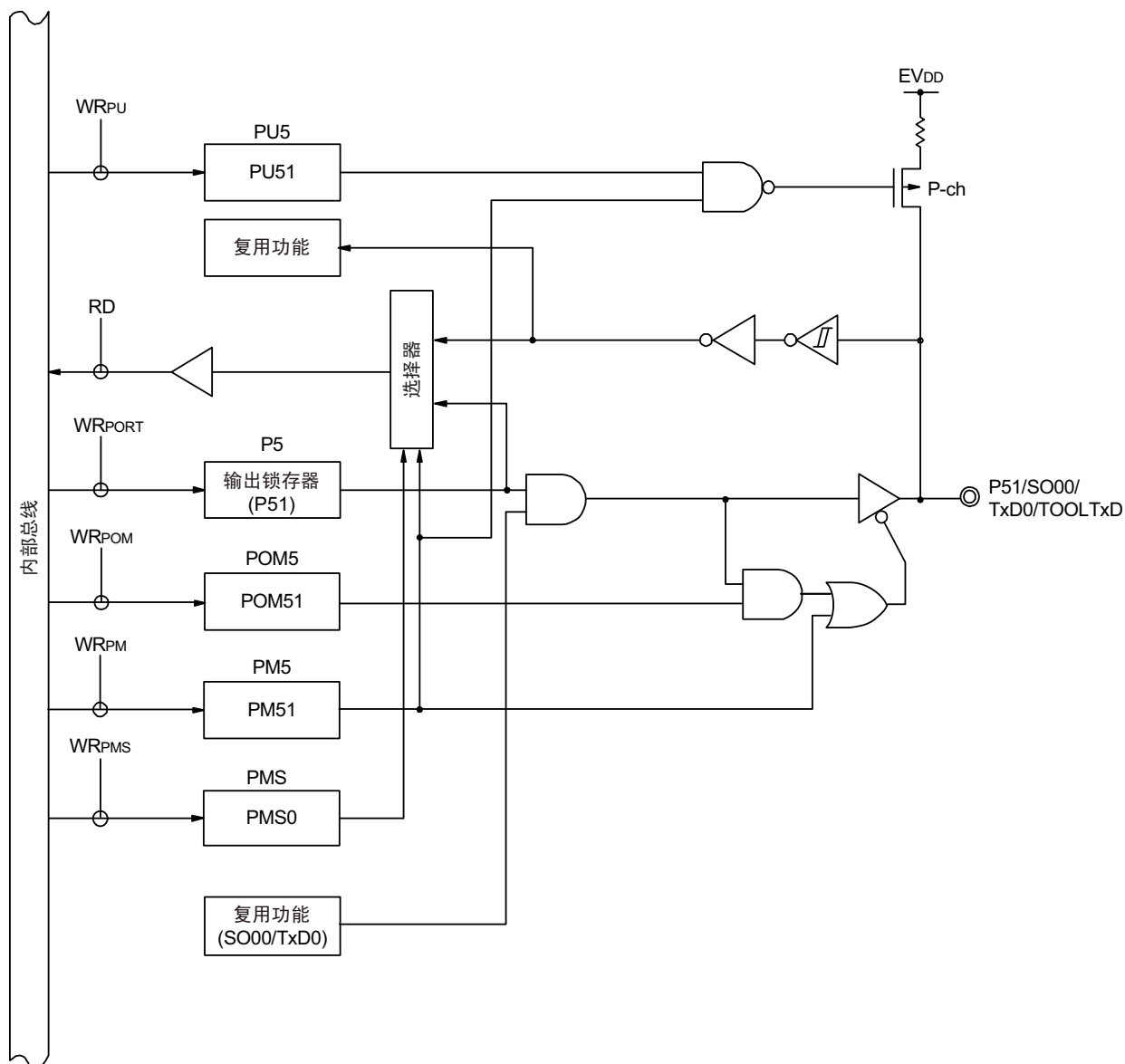
例如，64 引脚产品在 PIOR0 为“00H”时的端口 5 框图如图 4-23 ~ 图 4-28 所示。

图 4-23 P50 的框图



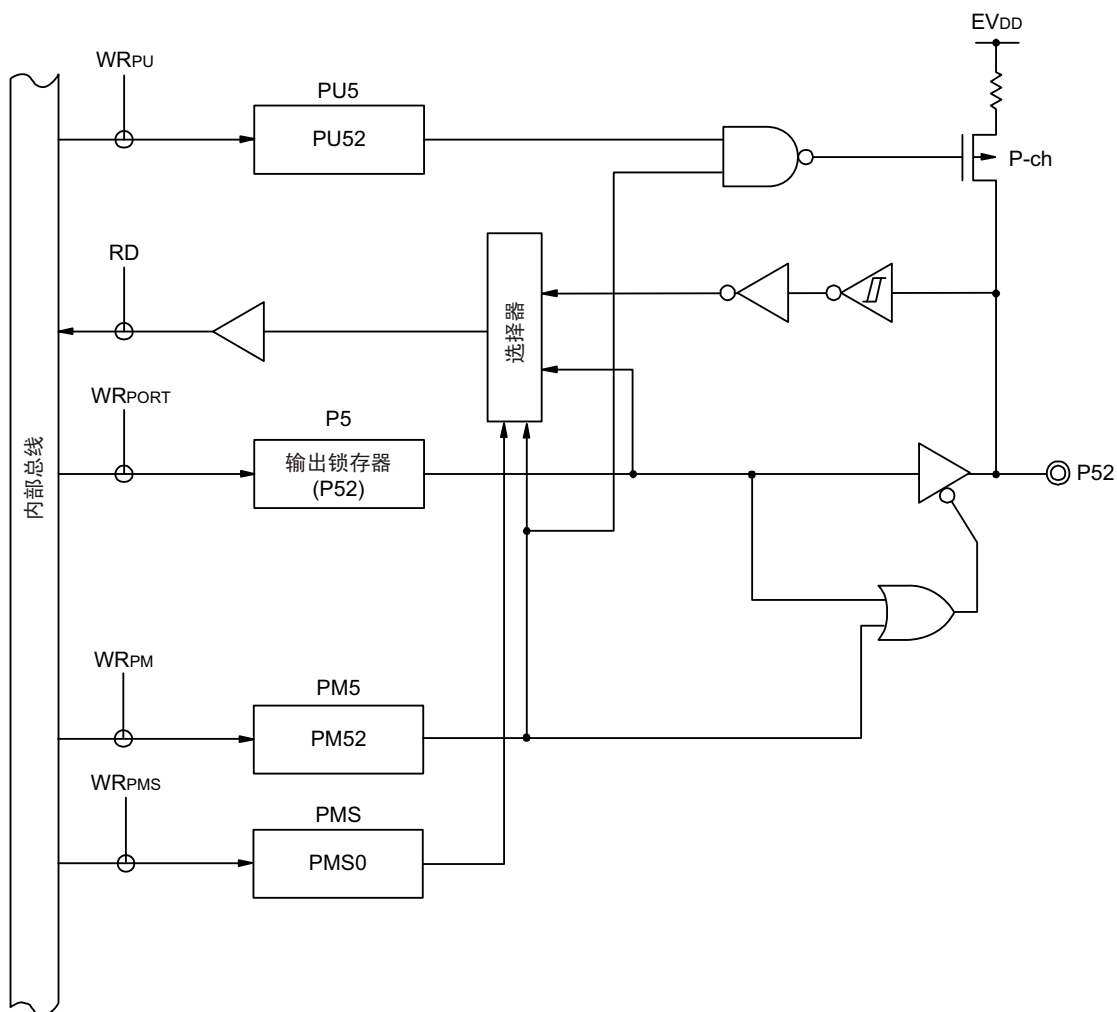
- P5 : 端口寄存器 5
- PU5 : 上拉电阻选择寄存器 5
- PM5 : 端口模式寄存器 5
- PIM5 : 端口输入模式寄存器 5
- POM5 : 端口输出模式寄存器 5
- PMS : 端口模式选择寄存器
- RD : 读信号
- WRxx : 写信号

图 4-24 P51 的框图



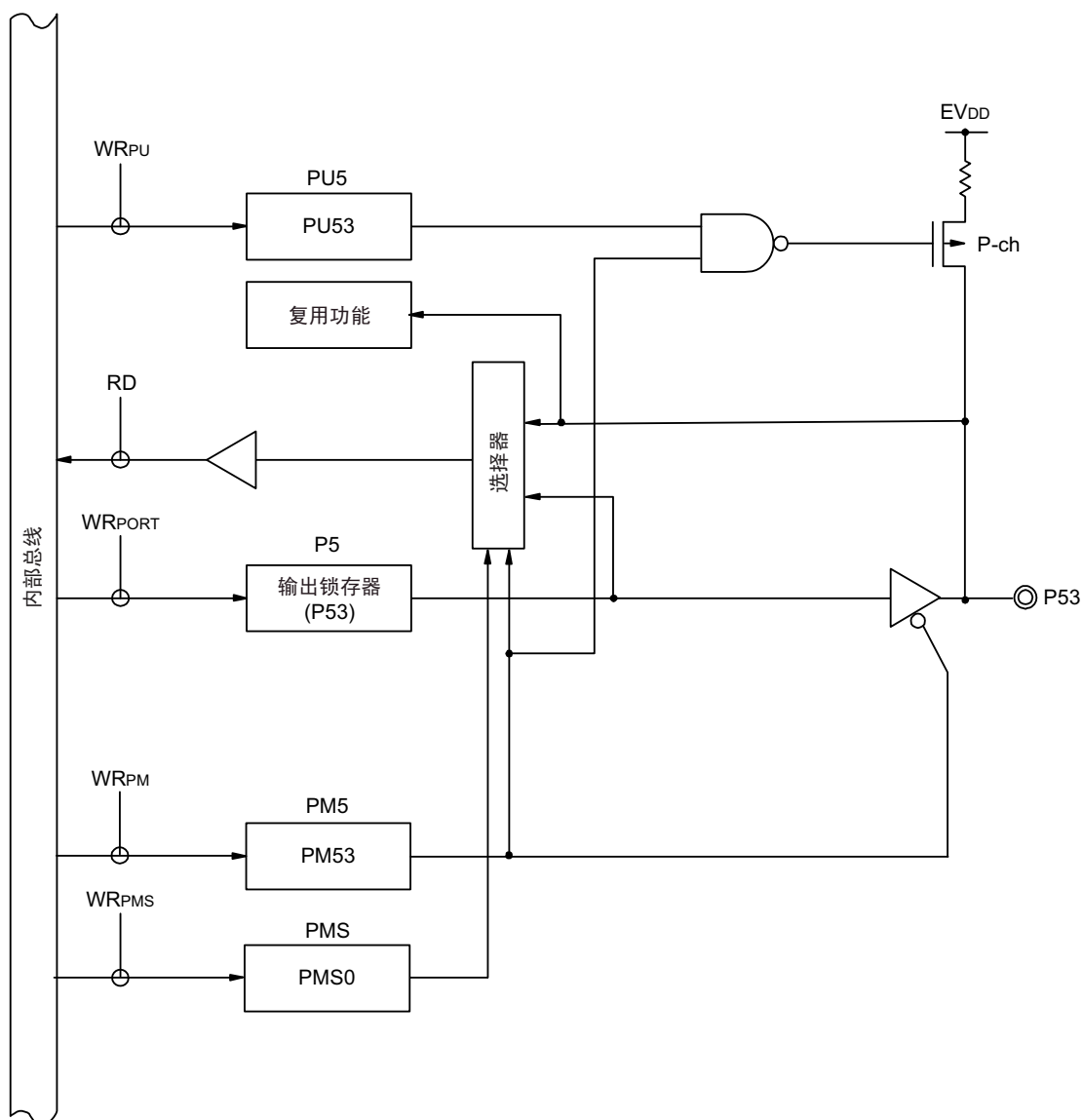
- P5 : 端口寄存器 5
 PU5 : 上拉电阻选择寄存器 5
 PM5 : 端口模式寄存器 5
 POM5 : 端口输出模式寄存器 5
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-25 P52 的框图



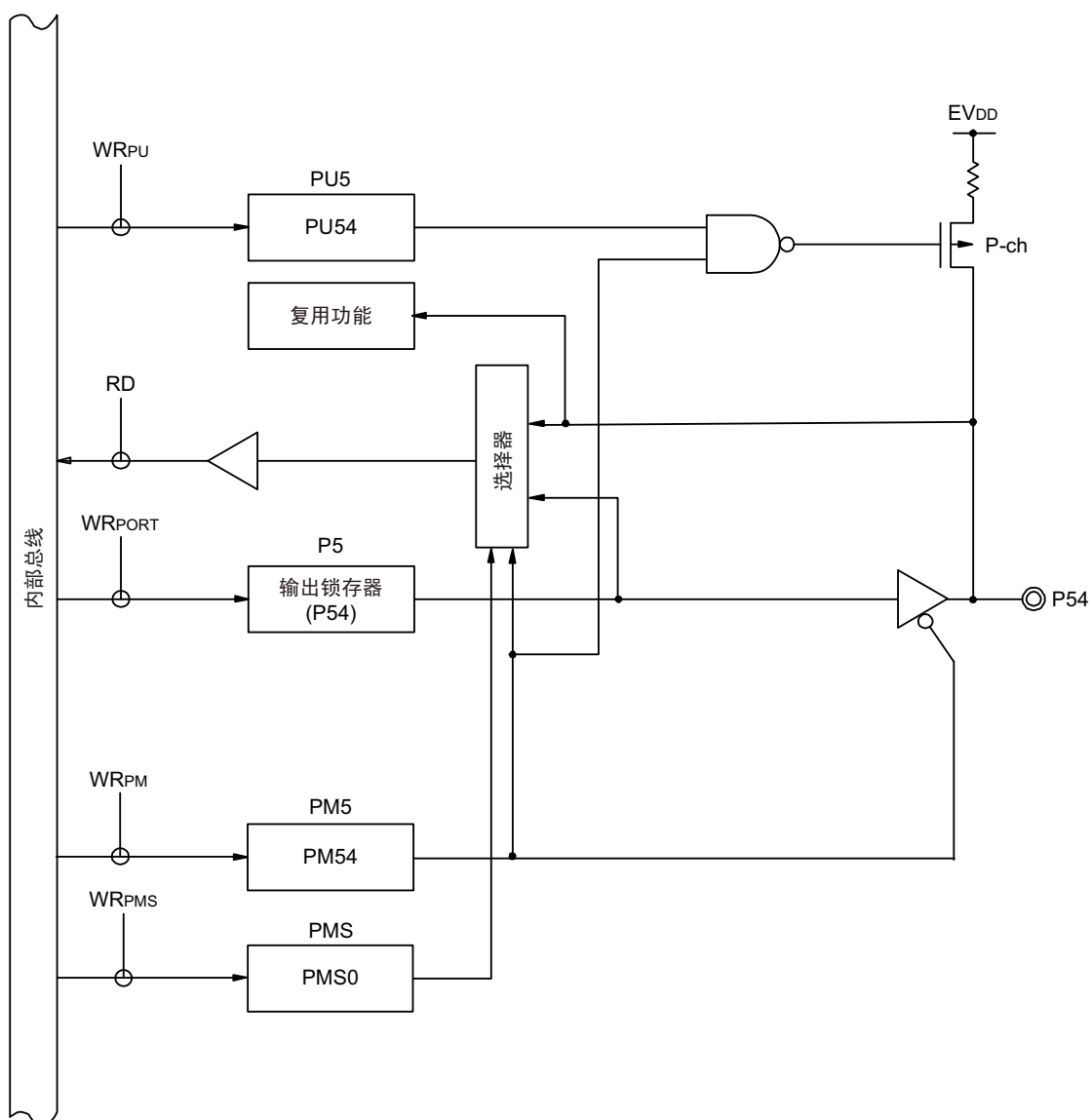
- P5 : 端口寄存器 5
 PU5 : 上拉电阻选择寄存器 5
 PM5 : 端口模式寄存器 5
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-26 P53 的框图



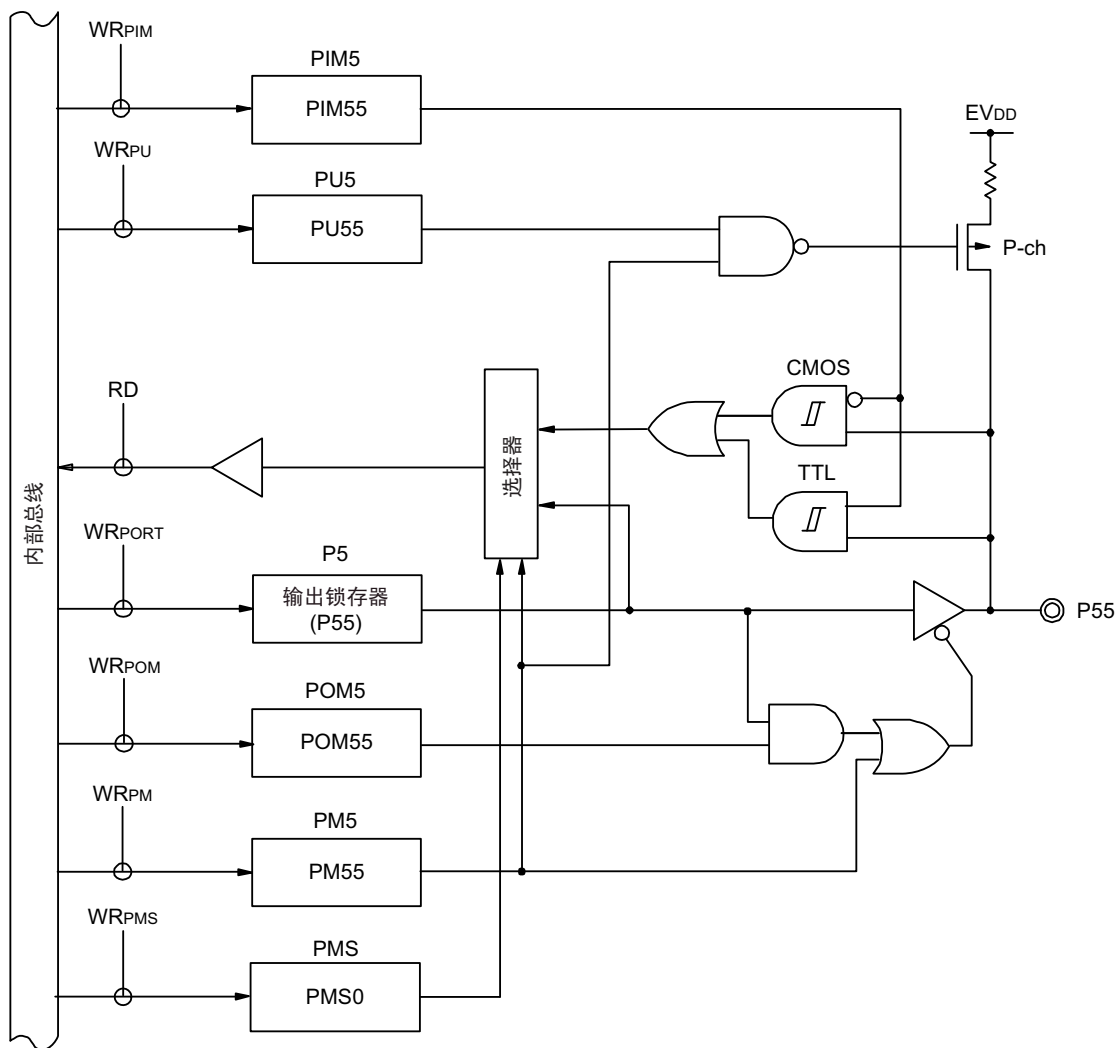
- P5 : 端口寄存器 5
 PU5 : 上拉电阻选择寄存器 5
 PM5 : 端口模式寄存器 5
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-27 P54 的框图



- P5 : 端口寄存器 5
 PU5 : 上拉电阻选择寄存器 5
 PM5 : 端口模式寄存器 5
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-28 P55 的框图



- P5 : 端口寄存器 5
 PU5 : 上拉电阻选择寄存器 5
 PM5 : 端口模式寄存器 5
 PIM5 : 端口输入模式寄存器 5
 POM5 : 端口输出模式寄存器 5
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

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 ~ P63 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、时钟的输入 / 输出和芯片选择输入。

在产生复位信号后，端口 6 变为输入模式。

表 4-9 使用端口 6 时的寄存器设定

引脚名		PM6x	复用功能的设定	备注
名称	输入 / 输出			
P60	输入	1	×	
	输出	0	SCLA0 输出 =0 注	
P61	输入	1	×	
	输出	0	SDAA0 输出 =0 注	
P62	输入	1	×	
	输出	0	×	
P63	输入	1	×	
	输出	0	×	

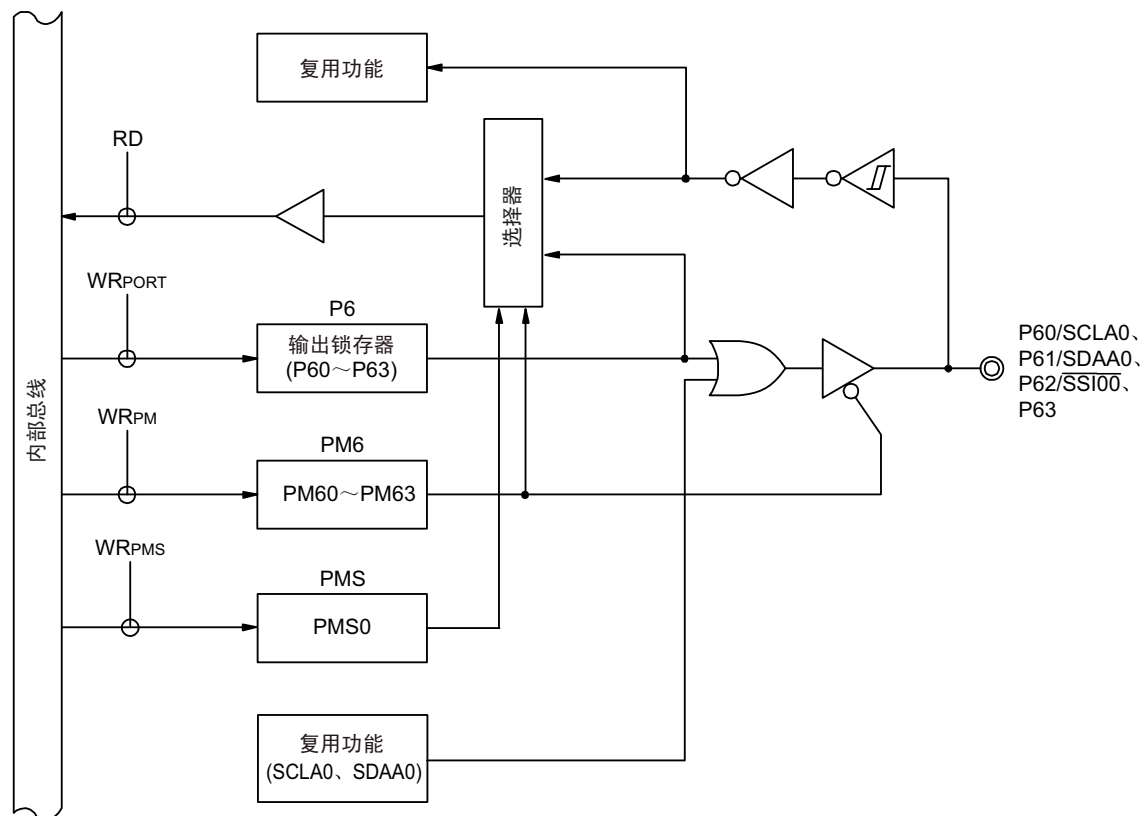
注 当将串行接口 IICA 功能的复用引脚用作通用端口时，必须停止对应的串行接口 IICA 的运行。

备注 × : 忽略

PM6x : 端口模式寄存器 6

例如，64 引脚产品的端口 6 框图如图 4-29 所示。

图 4-29 P60 ~ P63 的框图



- P6 : 端口寄存器 6
 PM6 : 端口模式寄存器 6
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

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。

能通过端口输出模式寄存器 7（POM7），以位为单位将 P71 引脚和 P74 引脚的输出设定为 N 沟道漏极开路输出（EV_{DD} 耐压）。

端口 7 作为复用功能，有键中断的输入、串行接口的数据输入 / 输出、时钟的输入 / 输出和外部中断的请求输入。

在产生复位信号后，端口 7 变为输入模式。

表 4-10 使用端口 7 时的寄存器设定

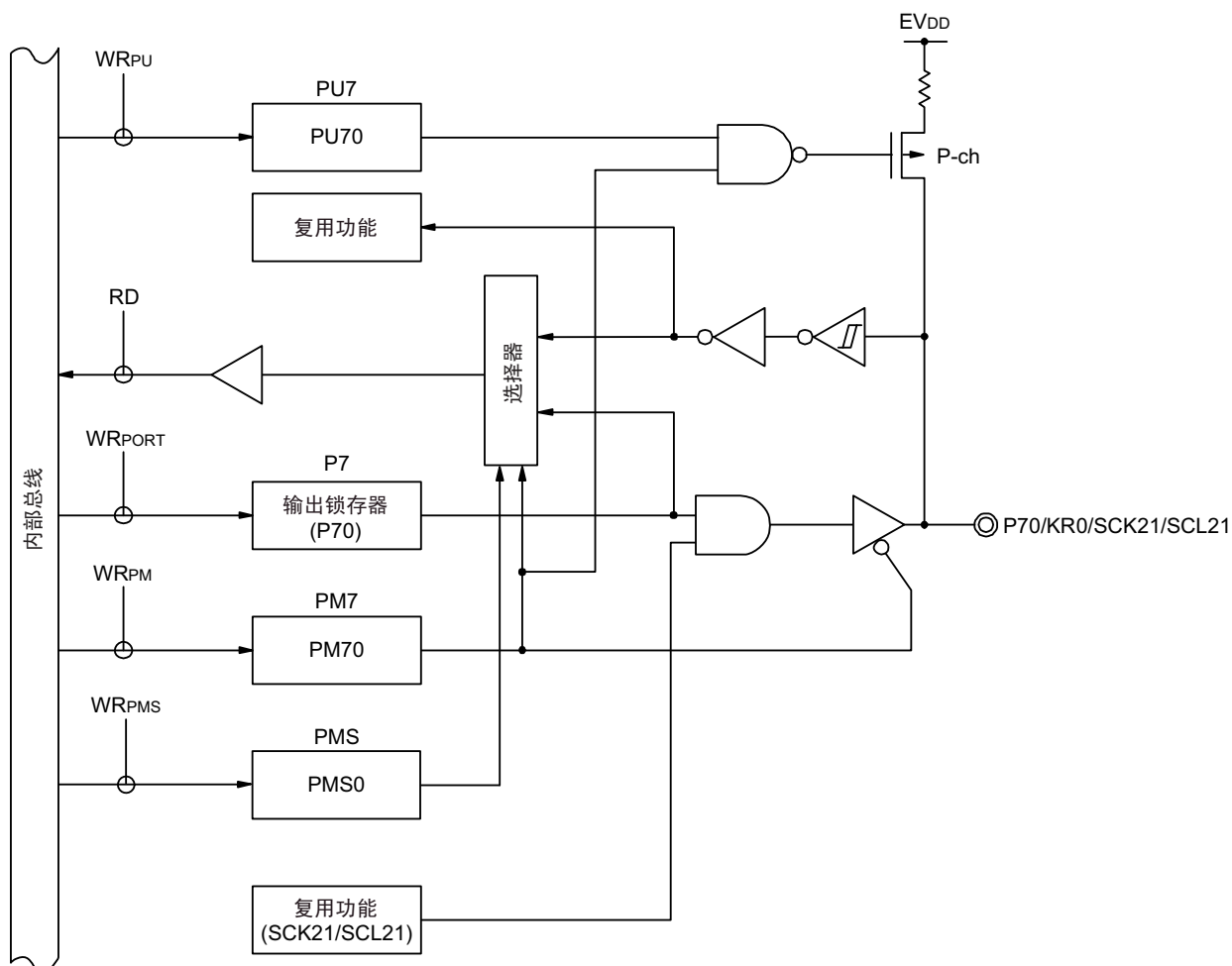
引脚名		PM7x	POM7x	复用功能的设定注 4	备注
名称	输入 / 输出				
P70	输入	1	×	×	
	输出	0	0	SCK21/SCL21 输出 =1 注 1	
P71	输入	1	×	×	
	输出	0	0	SDA21 输出 =1 注 2	CMOS 输出
		0	1		N 沟道漏极开路输出
P72	输入	1	—	×	
	输出	0		SO21 输出 =1 注 2	
P73	输入	1	×	×	
	输出	0	0	SO01 输出 =1 注 2	
P74	输入	1	×	×	
	输出	0	0	SDA01 输出 =1 注 2	CMOS 输出
		0	1		N 沟道漏极开路输出
P75	输入	1	×	×	
	输出	0	0	SCK01/SCL01 输出 =1 注 1	
P76	输入	1	—	×	
	输出	0		×	
P77	输入	1	×	×	
	输出	0	0	(TxD2 输出 =0 注 3)	

- 注 1. 当将串行阵列单元功能的复用引脚用作通用端口时，必须将对象单元通道对应的串行输出寄存器 m（SOm）的 CKOm_n 位、串行输出允许寄存器 m（SOEm）的 SOEm_n 位和串行通道允许状态寄存器 m（SEm）的 SEEm_n 位置初始值（mn=01、11）。
2. 当将串行阵列单元功能的复用引脚用作通用端口时，必须将对象单元通道对应的串行输出寄存器 m（SOm）的 SOm_n 位、串行输出允许寄存器 m（SOEm）的 SOEm_n 位和串行通道允许状态寄存器 m（SEm）的 SEEm_n 位置初始值（mn=01、11）。
3. 当外围 I/O 重定向寄存器 0（PIOR0）的 PIOR01 位为“1”并且将串行阵列单元功能的复用引脚用作通用端口时，必须将对象单元通道对应的串行输出寄存器 m（SOm）的 SOm_n 位、串行输出允许寄存器 m（SOEm）的 SOEm_n 位和串行通道允许状态寄存器 m（SEm）的 SEEm_n 位置初始值（mn=00）。
4. 能通过设定外围 I/O 重定向寄存器 0（PIOR0），分配（）内的功能。

备注 × : 忽略
 PM7x : 端口模式寄存器 7
 POM7x : 端口输出模式寄存器 7

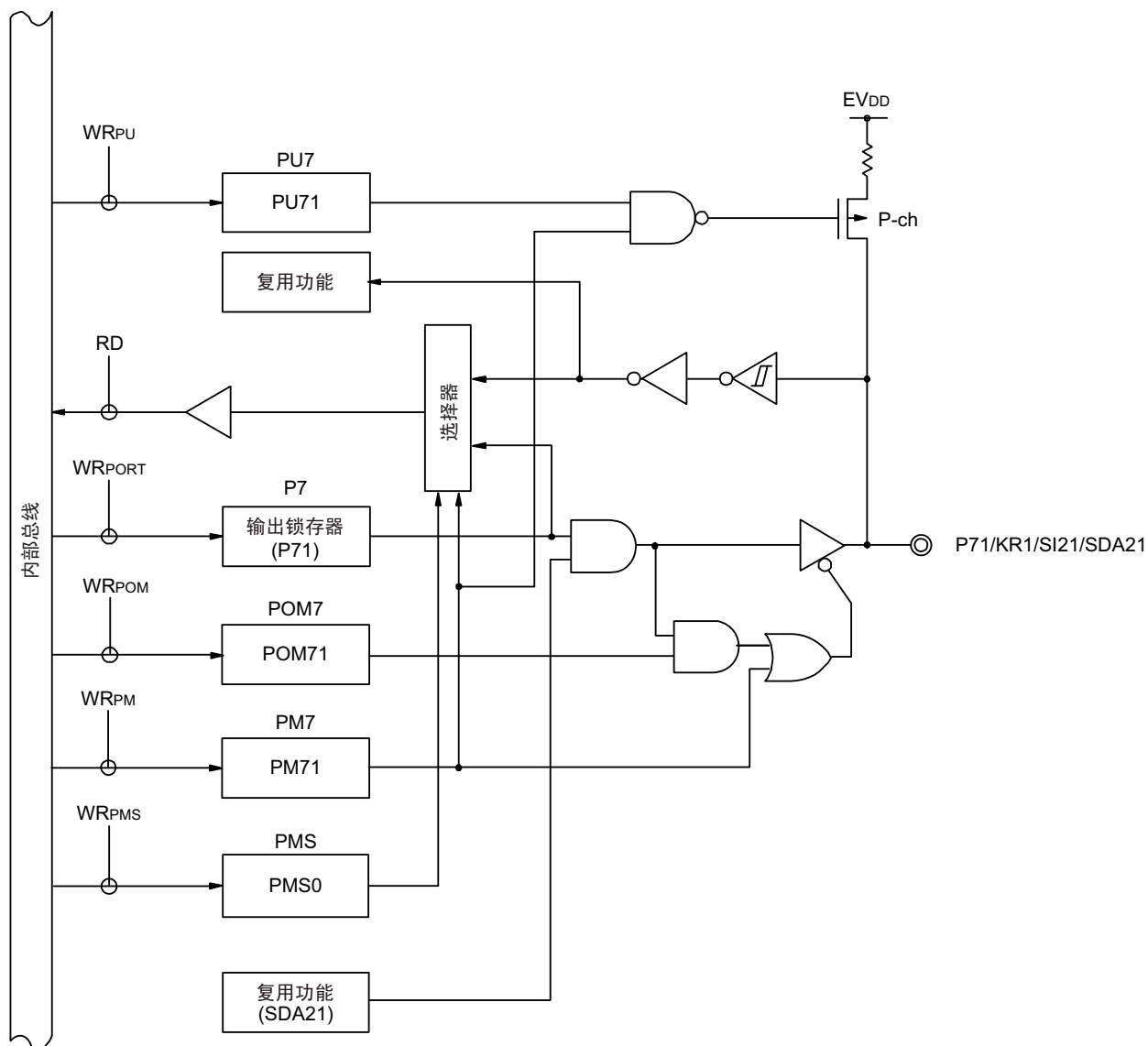
例如，64 引脚产品在 PIOR0 为“00H”时的端口 7 框图如图 4-30 ~ 图 4-34 所示。

图 4-30 P70 的框图



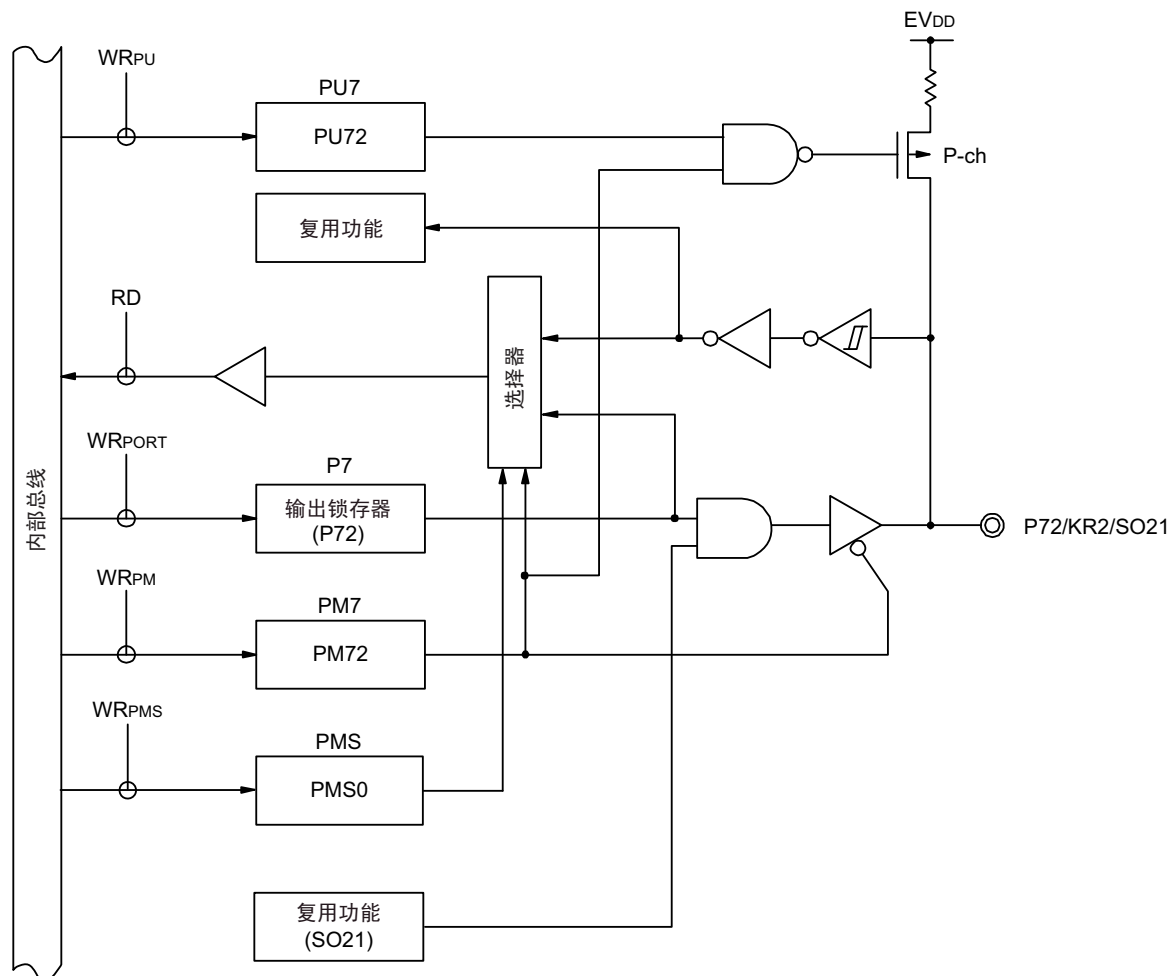
- P7 : 端口寄存器 7
 PU7 : 上拉电阻选择寄存器 7
 PM7 : 端口模式寄存器 7
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-31 P71 的框图



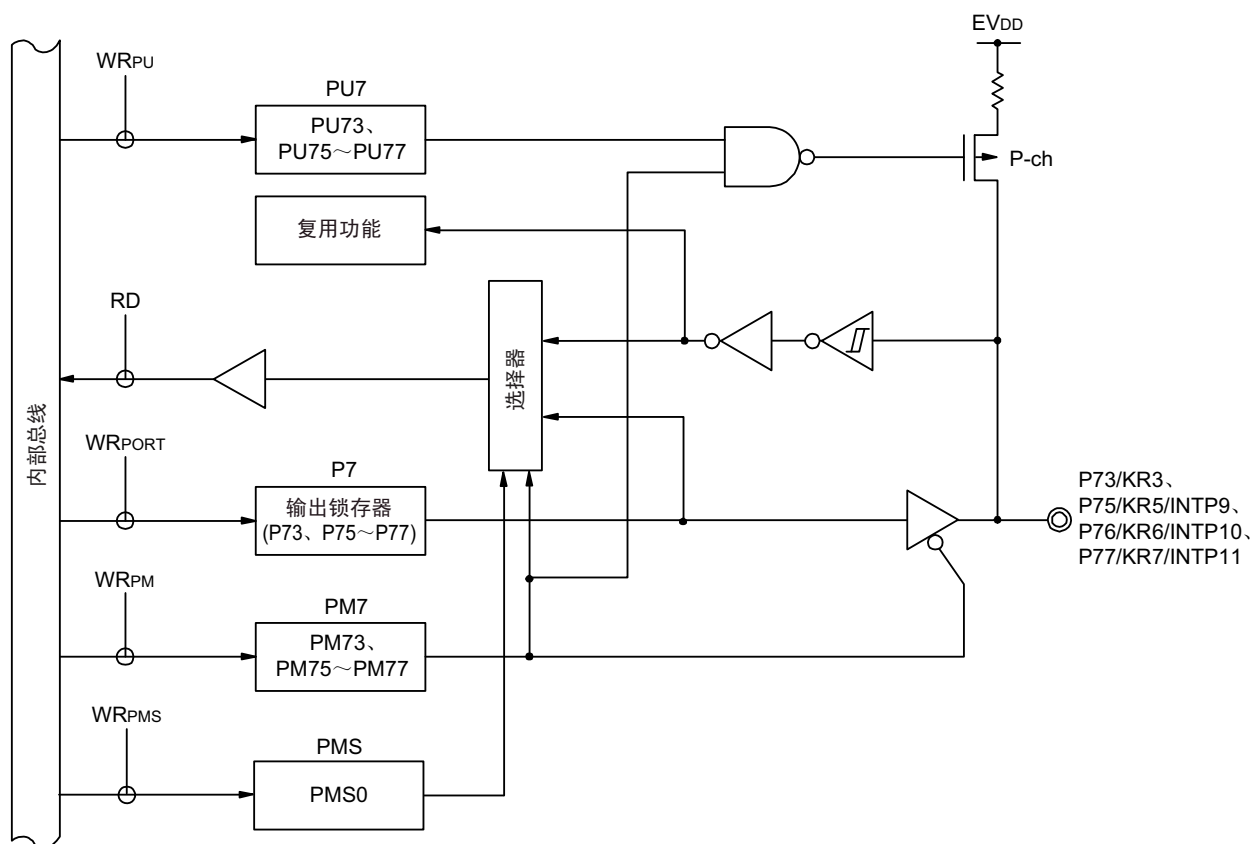
- P7 : 端口寄存器 7
 PU7 : 上拉电阻选择寄存器 7
 PM7 : 端口模式寄存器 7
 POM7 : 端口输出模式寄存器 7
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-32 P72 的框图



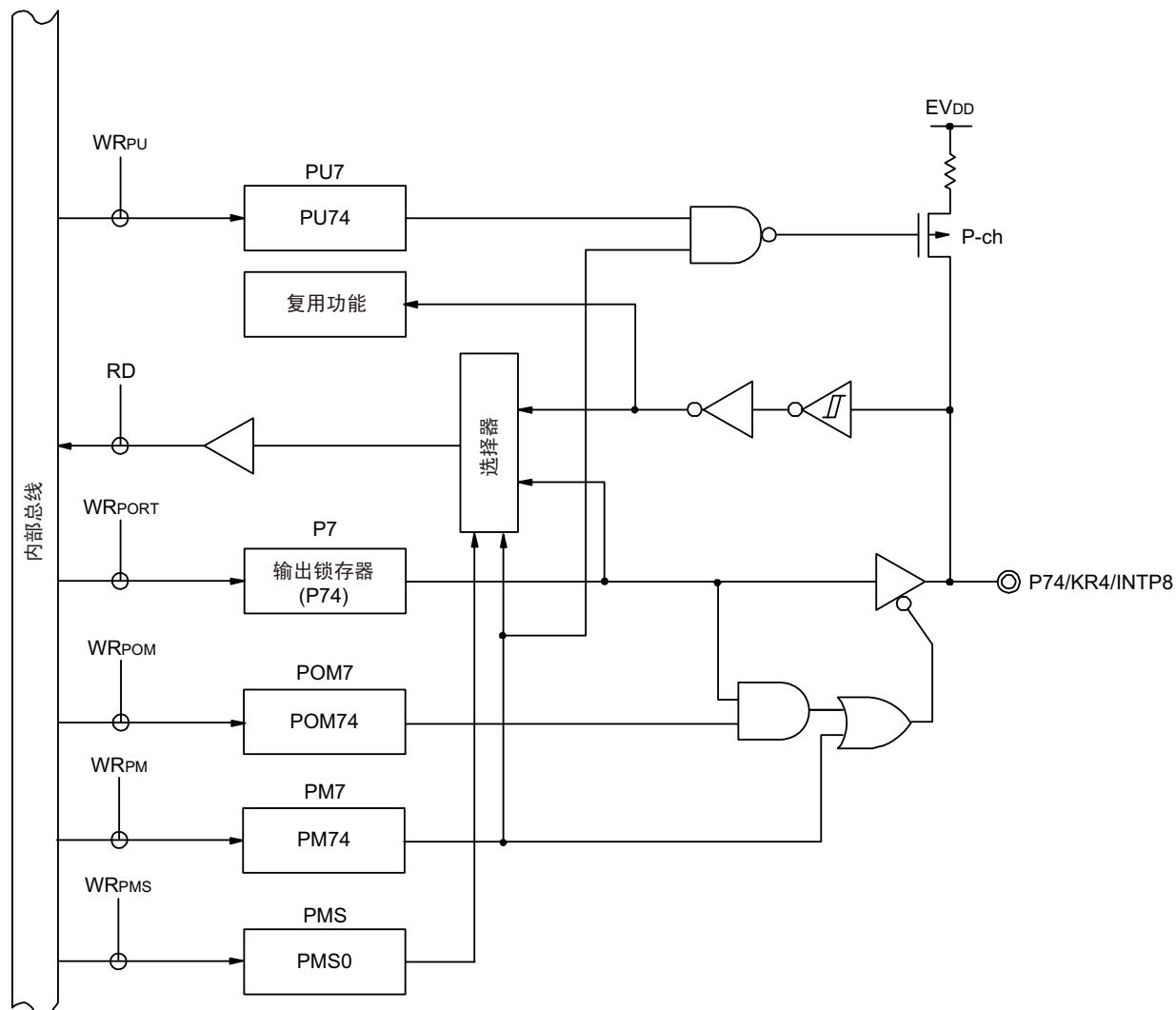
- P7 : 端口寄存器 7
 PU7 : 上拉电阻选择寄存器 7
 PM7 : 端口模式寄存器 7
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-33 P73、P75 ~ P77 的框图



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

图 4-34 P74 的框图



- P7 : 端口寄存器 7
 PU7 : 上拉电阻选择寄存器 7
 PM7 : 端口模式寄存器 7
 POM7 : 端口输出模式寄存器 7
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

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 ~ P124 是 4 位输入端口。

当将 P120 引脚用作输入时，必须通过端口模式控制寄存器 12（PMC12）设定数字或者模拟（能以位为单位进行设定）。

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、副系统时钟的谐振器连接、主系统时钟的外部时钟输入和副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 ~ P124 变为输入模式。

表 4-11 使用端口 12 时的寄存器设定

引脚名		PM12x	PMC12x	复用功能的设定	备注
名称	输入 / 输出				
P120	输入	1	0	×	
	输出	0	0	×	
P121	输入	—	—	CMC 寄存器的 OSCSEL 位 =0 或者 EXCLK 位 =1	
P122	输入	—	—	CMC 寄存器的 OSCSEL 位 =0	
P123	输入	—	—	CMC 寄存器的 OSCSELS 位 =0 或者 EXCLKS 位 =1	
P124	输入	—	—	CMC 寄存器的 OSCSELS 位 =0	

注意 只能在解除复位后设定 1 次 P121 ~ P124 的功能。设定为连接谐振器或者振荡器的端口在不复位的情况下不能用作输入端口。

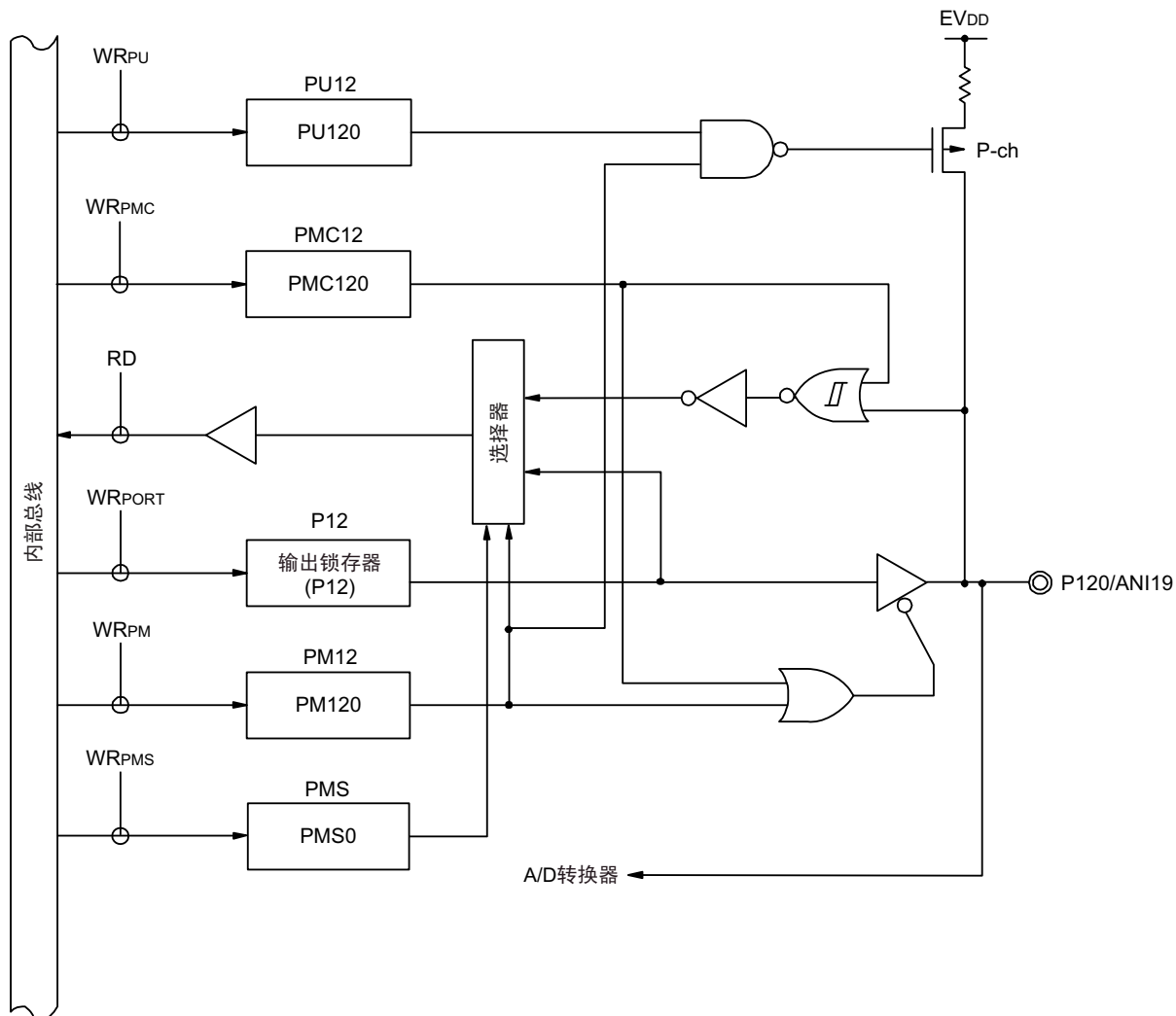
备注 1. × : 忽略

PM12x : 端口模式寄存器 12

PMC12x : 端口模式控制寄存器 12

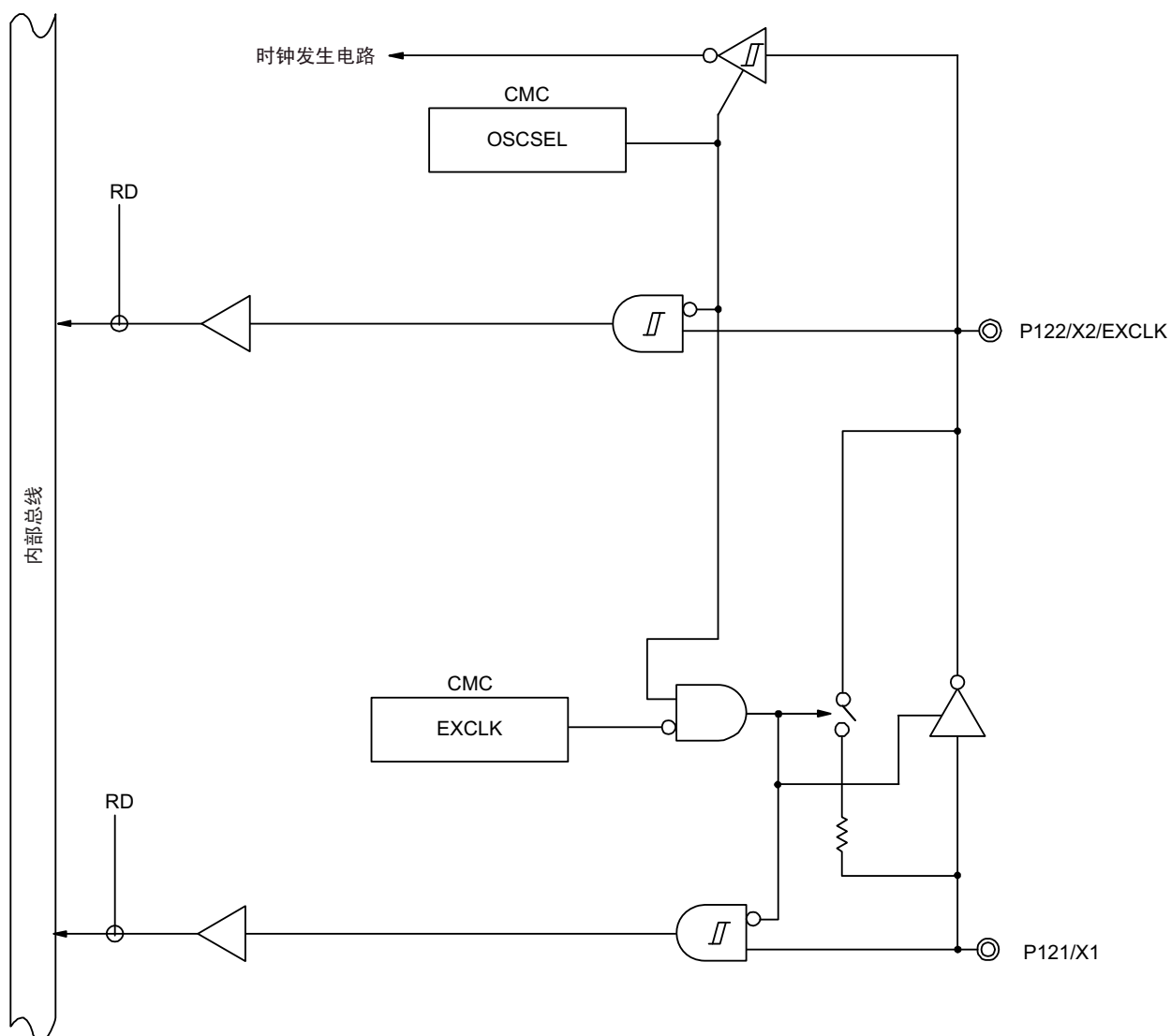
例如，64 引脚产品的端口 12 框图如图 4-35 ~ 图 4-37 所示。

图 4-35 P120 的框图



- P12 : 端口寄存器 12
 PU12 : 上拉电阻选择寄存器 12
 PM12 : 端口模式寄存器 12
 PMC12 : 端口模式控制寄存器 12
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

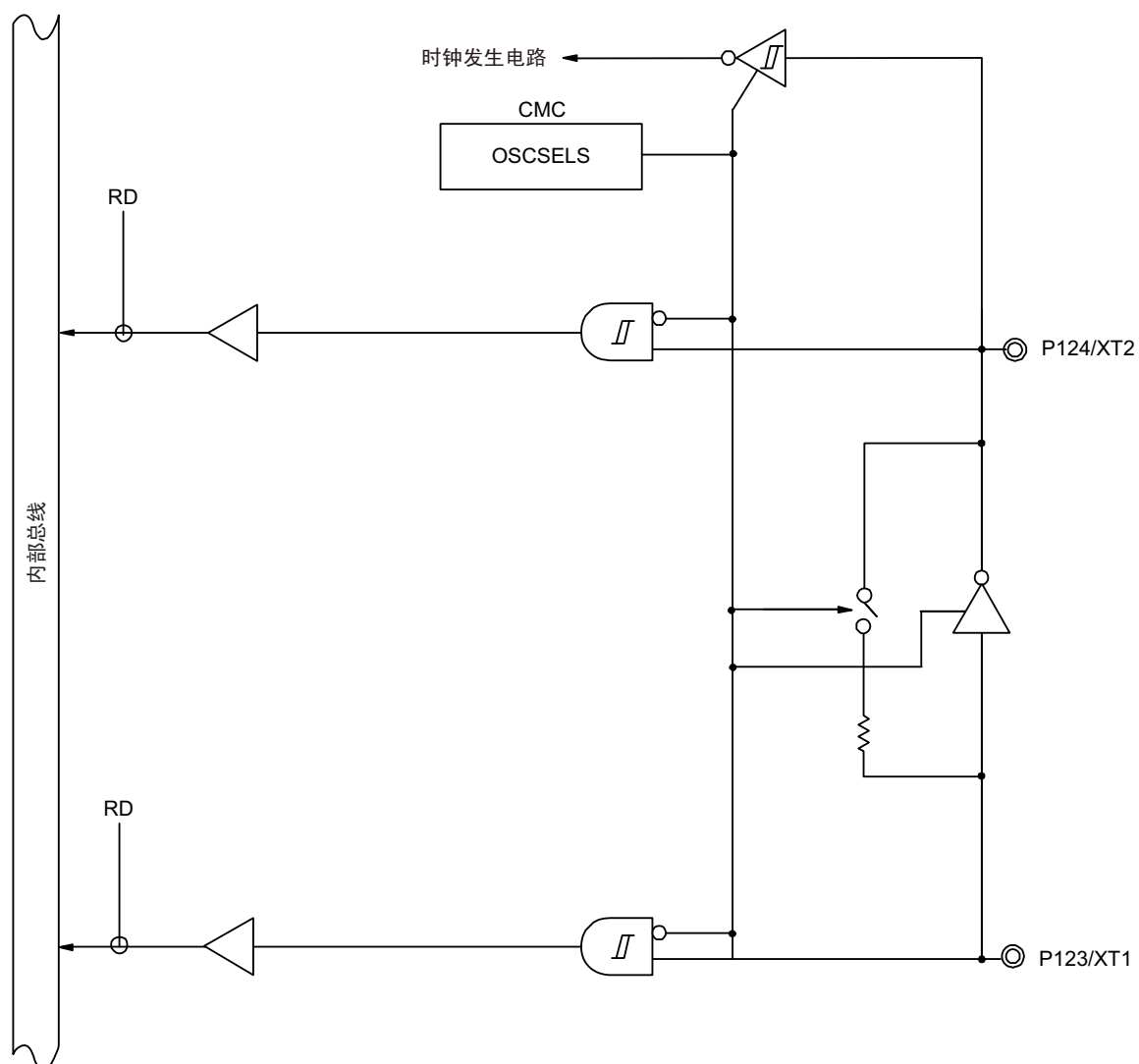
图 4-36 P121 和 P122 的框图



CMC : 时钟运行模式控制寄存器

RD : 读信号

图 4-37 P123 和 P124 的框图



CMC : 时钟运行模式控制寄存器

RD : 读信号

4.2.10 端口 13

P130 是带输出锁存器的 1 位输出专用端口，P137 是 1 位输入专用端口。
P130 固定为输出模式，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。

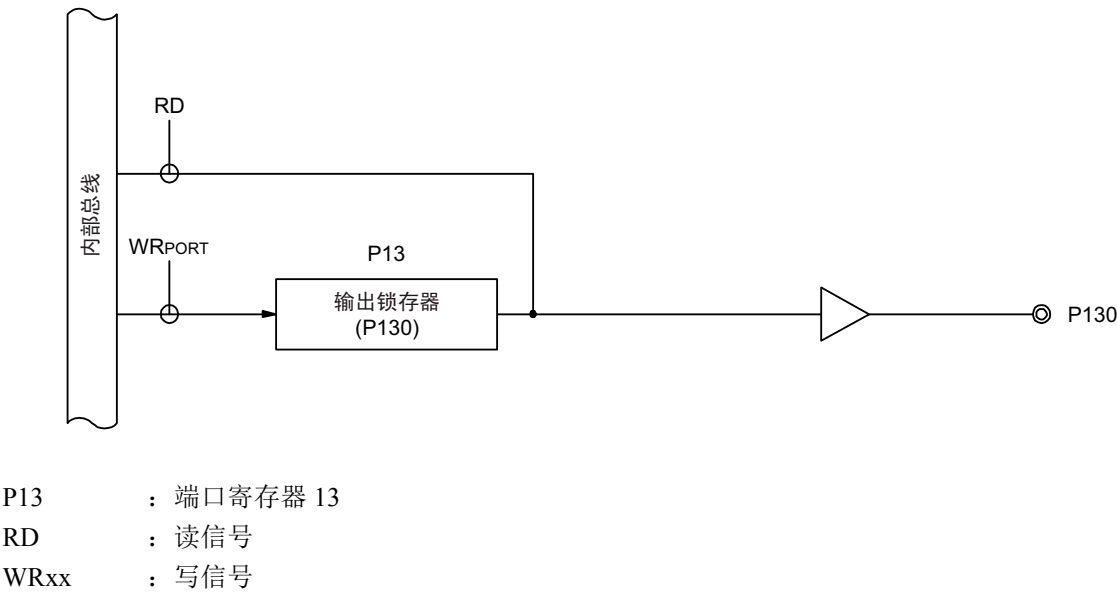
表 4-12 使用端口 13 时的寄存器设定

引脚名		复用功能的设定	备注
名称	输入 / 输出		
P130	输出	—	
P137	输入	×	

备注 ×：忽略

例如，64 引脚产品的端口 13 框图如图 4-38 和图 4-39 所示。

图 4-38 P130 的框图



备注 如果发生复位，P130 就输出低电平。因此，如果在发生复位前将 P130 设定为高电平输出，就能将 P130 的输出作为 CPU 的复位信号进行虚拟输出。

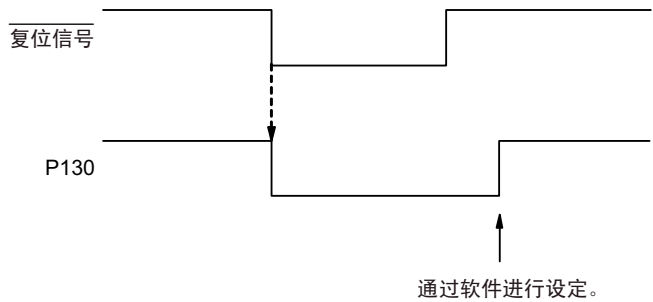
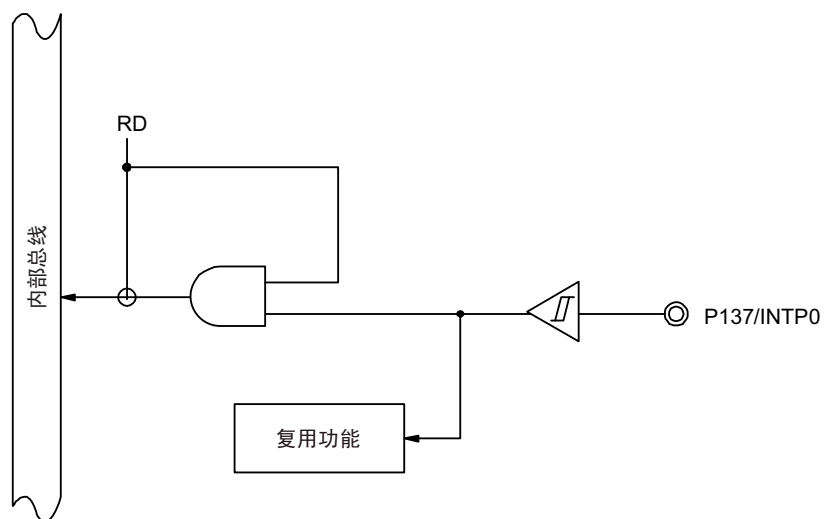


图 4-39 P137 的框图



RD : 读信号

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。当将 P140、P141、P146、P147 引脚用作输入端口时，能通过上拉电阻选择寄存器 14（PU14）以位为单位使用内部上拉电阻。

当将 P147 引脚用作输入端口时，必须通过端口模式控制寄存器 14（PMC14）设定数字或者模拟（能以位为单位进行设定）。

端口 14 作为复用功能，有时钟 / 蜂鸣器的输出、外部中断的请求输入和 A/D 转换器的模拟输入。

在产生复位信号后，P140、P141、P146 变为输入模式，P147 变为模拟输入。

表 4-13 使用端口 14 时的寄存器设定

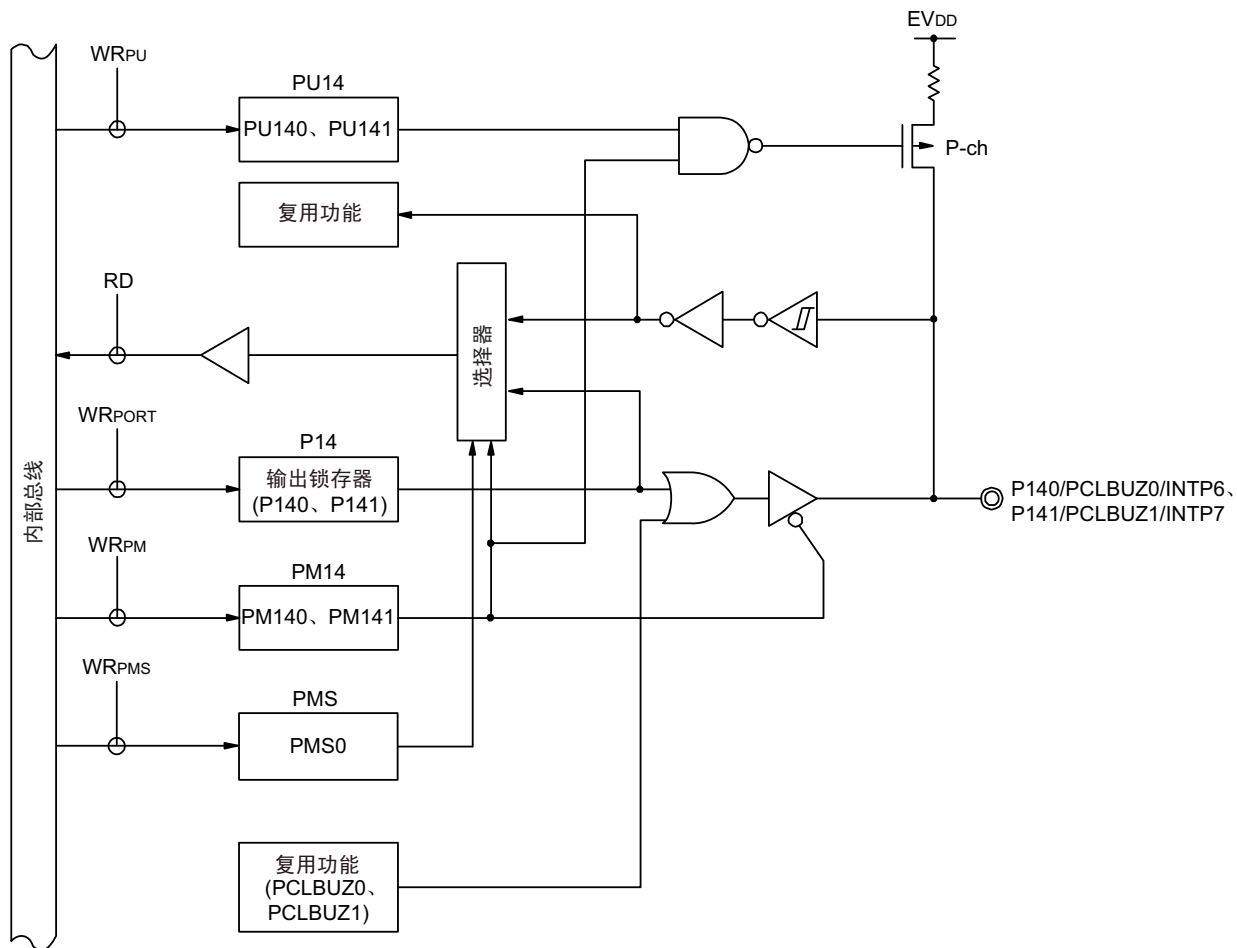
引脚名		PM14x	PIM14x	POM14x	PMC14x	复用功能的设定	备注
名称	输入 / 输出						
P140	输入	1	—	—	—	×	
	输出	0				PCLBUZ0 输出 =0 注	
P141	输入	1	—	—	—	×	
	输出	0				PCLBUZ1 输出 =0 注	
P146	输入	1	—	—	—	×	
	输出	0				×	
P147	输入	1	—	—	0	×	
	输出	0			0	×	

注 当将时钟 / 蜂鸣器输出功能的复用引脚用作通用端口时，必须将时钟输出选择寄存器 i（CKSi）的 PCLOEi 位置初始值（i=0、1）。

备注 × : 忽略
 PM14x : 端口模式寄存器 14
 PIM14x : 端口输入模式寄存器 14
 POM14x : 端口输出模式寄存器 14
 PMC14x : 端口模式控制寄存器 14

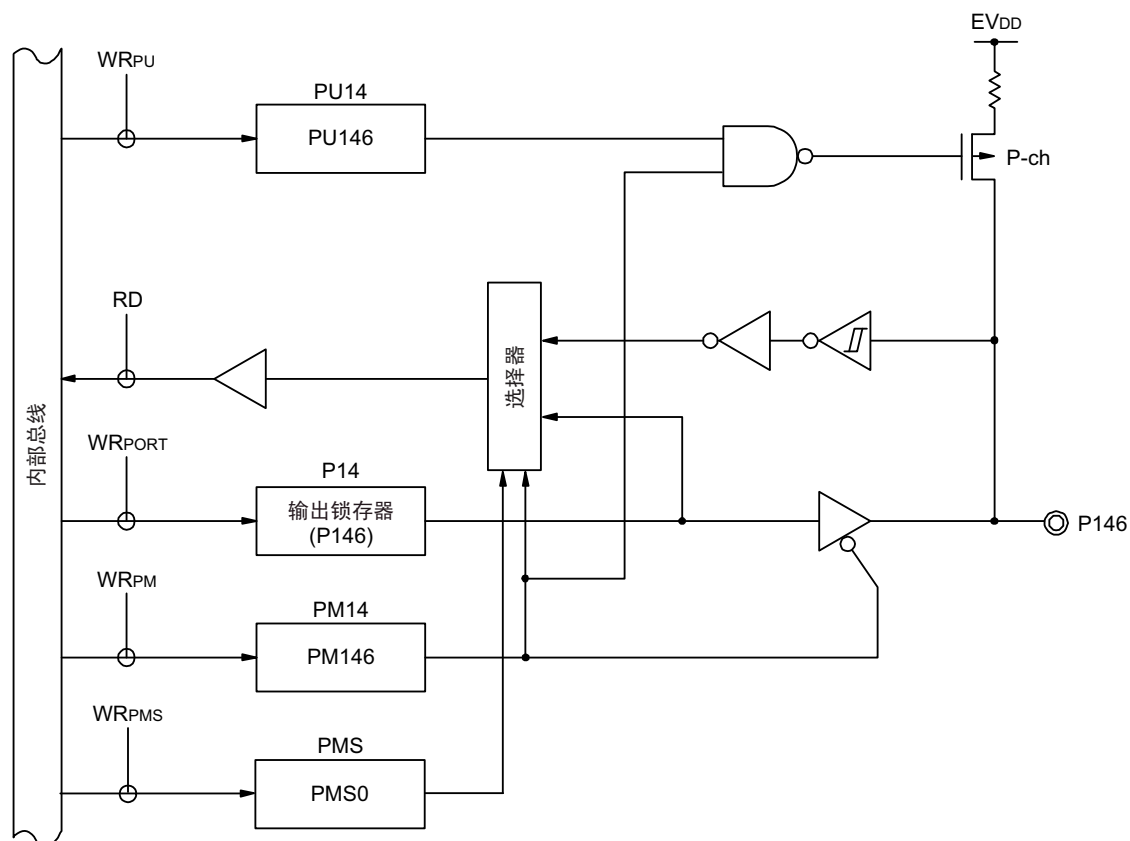
例如，64 引脚产品的端口 14 框图如图 4-40 ~ 图 4-42 所示。

图 4-40 P140 和 P141 的框图



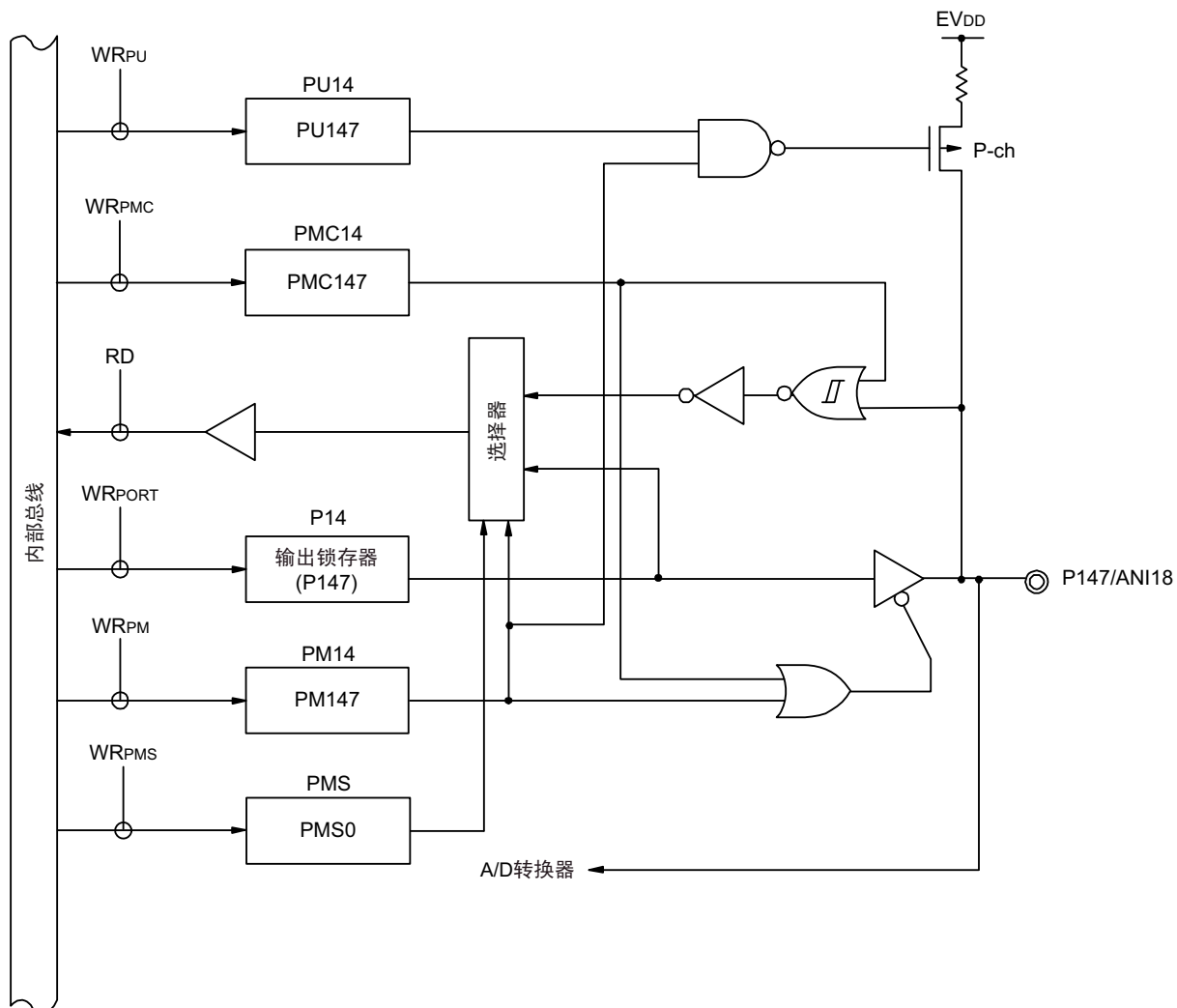
- P14 : 端口寄存器 14
- PU14 : 上拉电阻选择寄存器 14
- PM14 : 端口模式寄存器 14
- PMS : 端口模式选择寄存器
- RD : 读信号
- WRxx : 写信号

图 4-41 P146 的框图



- P14 : 端口寄存器 14
 PU14 : 上拉电阻选择寄存器 14
 PM14 : 端口模式寄存器 14
 PMS : 端口模式选择寄存器
 RD : 读信号
 WRxx : 写信号

图 4-42 P147 的框图



P14	: 端口寄存器 14
PU14	: 上拉电阻选择寄存器 14
PM14	: 端口模式寄存器 14
PMC14	: 端口模式控制寄存器 14
PMS	: 端口模式选择寄存器
RD	: 读信号
WRxx	: 写信号

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 (PM_{xx})
- 端口寄存器 (P_{xx})
- 上拉电阻选择寄存器 (PU_{xx})
- 端口输入模式寄存器 (PIM_x)
- 端口输出模式寄存器 (POM_x)
- 端口模式控制寄存器 (PMC_{xx})
- A/D 端口配置寄存器 (ADPC)
- 外围 I/O 重定向寄存器 0、1 (PIOR0、PIOR1)
- 全局数字输入禁止寄存器 (GDIDIS)

注意 分配的寄存器和位因产品而不同。有关各产品分配的寄存器和位，请参照表 4-14。必须给未分配的位设定初始值。

表 4-14 各产品分配的 PM_{xx}、P_{xx}、PU_{xx}、PIM_{xx}、POM_{xx}、PMC_{xx} 寄存器及其位
(32 引脚产品、64 引脚产品) (1/3)

端口		位名						64-pin	32-pin
		PM _{xx} 寄存器	P _{xx} 寄存器	PU _{xx} 寄存器	PIM _{xx} 寄存器	POM _{xx} 寄存器	PMC _{xx} 寄存器		
端口 0	0	PM00	P00	PU00	—	POM00	PMC00 注	○	○
	1	PM01	P01	PU01	PIM01	—	PMC01 注	○	○
	2	PM02	P02	PU02	—	POM02	PMC02	○	—
	3	PM03	P03	PU03	PIM03	POM03	PMC03	○	—
	4	PM04	P04	PU04	PIM04	POM04	—	○	—
	5	PM05	P05	PU05	—	—	—	○	—
	6	PM06	P06	PU06	—	—	—	○	—
	7	—	—	—	—	—	—	—	—
端口 1	0	PM10	P10	PU10	PIM10	POM10	—	○	○
	1	PM11	P11	PU11	—	POM11	—	○	○
	2	PM12	P12	PU12	—	—	—	○	○
	3	PM13	P13	PU13	—	POM13	—	○	○
	4	PM14	P14	PU14	PIM14	POM14	—	○	○
	5	PM15	P15	PU15	PIM15	POM15	—	○	○
	6	PM16	P16	PU16	PIM16	—	—	○	○
	7	PM17	P17	PU17	PIM17	POM17	—	○	○
端口 2	0	PM20	P20	—	—	—	—	○	○
	1	PM21	P21	—	—	—	—	○	○
	2	PM22	P22	—	—	—	—	○	○
	3	PM23	P23	—	—	—	—	○	○
	4	PM24	P24	—	—	—	—	○	—
	5	PM25	P25	—	—	—	—	○	—
	6	PM26	P26	—	—	—	—	○	—
	7	PM27	P27	—	—	—	—	○	—

注 只限于 32 引脚产品。

表 4-14 各产品分配的 PMxx、Pxx、PUxx、PIMxx、POMxx、PMCxx 寄存器及其位
(32 引脚产品、64 引脚产品) (2/3)

端口		位名						64-pin	32-pin
		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 3	0	PM30	P30	PU30	PIM30	POM30	—	○	○
	1	PM31	P31	PU31	PIM31	—	—	○	○
	2	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—
端口 4	0	PM40	P40	PU40	—	—	—	○	○
	1	PM41	P41	PU41	—	—	—	○	—
	2	PM42	P42	PU42	—	—	—	○	—
	3	PM43	P43	PU43	—	—	—	○	—
	4	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—
端口 5	0	PM50	P50	PU50	PIM50	POM50	—	○	○
	1	PM51	P51	PU51	—	POM51	—	○	○
	2	PM52	P52	PU52	—	—	—	○	—
	3	PM53	P53	PU53	—	—	—	○	—
	4	PM54	P54	PU54	—	—	—	○	—
	5	PM55	P55	PU55	PIM55	POM55	—	○	—
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—
端口 6	0	PM60	P60	—	—	—	—	○	○
	1	PM61	P61	—	—	—	—	○	○
	2	PM62	P62	—	—	—	—	○	○
	3	PM63	P63	—	—	—	—	○	—
	4	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—
端口 7	0	PM70	P70	PU70	—	—	—	○	○
	1	PM71	P71	PU71	—	POM71	—	○	—
	2	PM72	P72	PU72	—	—	—	○	—
	3	PM73	P73	PU73	—	—	—	○	—
	4	PM74	P74	PU74	—	POM74	—	○	—
	5	PM75	P75	PU75	—	—	—	○	—
	6	PM76	P76	PU76	—	—	—	○	—
	7	PM77	P77	PU77	—	—	—	○	—

表 4-14 各产品分配的 PMxx、Pxx、PUxx、PIMxx、POMxx、PMCxx 寄存器及其位
(32 引脚产品、64 引脚产品) (3/3)

端口		位名						64-pin	32-pin
		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 12	0	PM120	P120	PU120	—	—	PMC120	○	○
	1	—	P121	—	—	—	—	○	○
	2	—	P122	—	—	—	—	○	○
	3	—	P123	—	—	—	—	○	—
	4	—	P124	—	—	—	—	○	—
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—
端口 13	0	—	P130	—	—	—	—	○	—
	1	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	P137	—	—	—	—	○	○
端口 14	0	PM140	P140	PU140	—	—	—	○	—
	1	PM141	P141	PU141	—	—	—	○	—
	2	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—
	6	PM146	P146	PU146	—	—	—	○	—
	7	PM147	P147	PU147	—	—	PMC147	○	○

以下以 64 引脚产品为例说明各寄存器的格式。

有关 64 引脚以外的产品的寄存器分配，请参照表 4-14。

4.3.1 端口模式寄存器 (PMxx)

这是以位为单位设定端口输入 / 输出的寄存器。

通过 1 位或者 8 位存储器操作指令设定端口模式寄存器。

在产生复位信号后，这些寄存器的值变为“FFH”。

当将端口引脚用作复用功能的引脚时，必须参照“4.5 使用复用功能时的端口相关寄存器的设定”进行设定。

图 4-43 端口模式寄存器的格式 (64 引脚产品)

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	PM147	PM146	1	1	1	1	PM141	PM140	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 P02、P03、P20 ~ P27、P120、P147 设定为 A/D 转换器的模拟输入功能的情况下，如果在输入模式中读端口，读取值就不是引脚电平而总是“0”。

图 4-44 端口寄存器的格式 (64 引脚产品)

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
P0	0	P06	P05	P04	P03	P02	P01	P00	FFF00H	00H (输出锁存器)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (输出锁存器)	R/W
P2	P27	P26	P25	P24	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	P43	P42	P41	P40	FFF04H	00H (输出锁存器)	R/W
P5	0	0	P55	P54	P53	P52	P51	P50	FFF05H	00H (输出锁存器)	R/W
P6	0	0	0	0	P63	P62	P61	P60	FFF06H	00H (输出锁存器)	R/W
P7	P77	P76	P75	P74	P73	P72	P71	P70	FFF07H	00H (输出锁存器)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	不定值	R/W注1
P13	P137	0	0	0	0	0	0	P130	FFF0DH	注 2	R/W注1
P14	P147	P146	0	0	0	0	P141	P140	FFF0EH	00H (输出锁存器)	R/W

Pmn	m=0 ~ 7、12 ~ 14, n=0 ~ 7	
	输出数据的控制 (输出模式)	输入数据的读取 (输入模式)
0	输出“0”。	输入低电平。
1	输出“1”。	输入高电平。

注 1. P121 ~ P124 和 P137 是只读位。

2. P137: 不定值

P130: 0 (输出锁存器)

注意 必须给未分配的位设定初始值。

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通过 POMmn=0 设定为输入模式（PMmn=1）的位，以位为单位使用内部上拉电阻。对于设定为输出模式的位，与上拉电阻选择寄存器的设定无关，不连接内部上拉电阻。当用作复用功能的输出引脚时也相同。

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

图 4-45 上拉电阻选择寄存器的格式（64 引脚产品）

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	PU43	PU42	PU41	PU40	F0034H	01H	R/W
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	PU147	PU146	0	0	0	0	PU141	PU140	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（PIM0、PIM1、PIM3、PIM5）

这是以位为单位设定 P01、P03、P04、P10、P14 ~ P17、P30、P50、P55 的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 PIM0、PIM1、PIM3、PIM5 寄存器。
在产生复位信号后，这些寄存器的值变为“00H”。

图 4-46 端口输入模式寄存器的格式（64 引脚产品）

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PIM0	0	0	0	PIM04	PIM03	0	PIM01	0	F0040H	00H	R/W
PIM1	PIM17	PIM16	PIM15	PIM14	0	0	0	PIM10	F0041H	00H	R/W
PIM3	0	0	0	0	0	0	PIM31	PIM30	F0043H	00H	R/W
PIM5	0	0	PIM55	0	0	0	0	PIM50	F0045H	00H	R/W

PIMmn	Pmn 引脚的输入缓冲器的选择（m=0、1、3、5，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（POM0、POM1、POM3、POM5、POM7）

这些寄存器以位为单位设定 P00、P02 ~ P04、P10、P11、P13 ~ P15、P17、P30、P50、P51、P55、P71、P74 的输出模式。

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、SDA01、SDA10、SDA11、SDA20、SDA21 引脚选择 N 沟道漏极开路输出（EV_{DD} 耐压）模式。

通过 1 位或者 8 位存储器操作指令设定 POM0、POM1、POM3、POM5、POM7 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（EV_{DD} 耐压）模式（POMmn=1）的位，不连接内部上拉电阻。

图 4-47 端口输出模式寄存器的格式（64 引脚产品）

地址: F0050H	复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0
POM0	0	0	0	POM04	POM03	POM02	0	POM00

地址: F0051H	复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0
POM1	POM17	0	POM15	POM14	POM13	0	POM11	POM10

地址: F0053H	复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0
POM3	0	0	0	0	0	0	0	POM30

地址: F0055H	复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0
POM5	0	0	POM55	0	0	0	POM51	POM50

地址: F0057H	复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0
POM7	0	0	0	POM74	0	0	POM71	0

POMmn	Pmn 引脚的输出缓冲器的选择（m=0、1、3、5、7，n=0 ~ 5、7）
0	通常的输出模式
1	N 沟道漏极开路输出（EV _{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 0、12、14（PMC0、PMC12、PMC14）

这些寄存器以位为单位设定 P02、P03、P120、P147 的数字输入 / 输出或者模拟输入。
通过 1 位或者 8 位存储器操作指令设定 PMC0、PMC12、PMC14 寄存器。
在产生复位信号后，这些寄存器的值变为“FFH”。

图 4-48 端口模式控制寄存器的格式（64 引脚产品）

地址: F0060H	复位后: FFH	R/W								
符号	7	6	5	4	3	2	1	0		
PMC0	1	1	1	1	PMC03 注 2	PMC02 注 2	PMC01 注 1	PMC00 注 1		

地址: F006CH	复位后: FFH	R/W								
符号	7	6	5	4	3	2	1	0		
PMC12	1	1	1	1	1	1	1	PMC120		

地址: F006EH	复位后: FFH	R/W								
符号	7	6	5	4	3	2	1	0		
PMC14	PMC147	1	1	1	1	1	1	1		

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、2、3、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 1. 只限于 32 引脚产品。
2. 只限于 64 引脚产品。

注意 必须给未分配的位设定初始值。

4.3.7 A/D 端口配置寄存器（ADPC）

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

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 4-49 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）和数字输入 / 输出（D）的切换							
				ANI7/P27	ANI6/P26	ANI5/P25	ANI4/P24	ANI3/P23	ANI2/P22	ANI1/P21	ANI0/P20
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	D	D	D	D	D	D	D	D
0	0	1	0	D	D	D	D	D	D	D	A
0	0	1	1	D	D	D	D	D	D	A	A
0	1	0	0	D	D	D	D	D	A	A	A
0	1	0	1	D	D	D	D	A	A	A	A
0	1	1	0	D	D	D	A	A	A	A	A
0	1	1	1	D	D	A	A	A	A	A	A
1	0	0	0	D	A	A	A	A	A	A	A
上述以外				禁止设定。							

注意 1. 必须通过端口模式寄存器 2（PM2）将 A/D 转换使用的通道设定为输入模式。

2. 对于由 ADPC 寄存器设定为数字输入 / 输出的引脚，不能通过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 0 (PIOR0)

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。

通过 8 位存储器操作指令设定 PIOR0 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 4-50 外围 I/O 重定向寄存器 0 (PIOR0) 的格式

地址: F0077H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PIOR0	0	0	0	PIOR04	PIOR03	PIOR02	PIOR01	PIOR00

位	功能	64 引脚		32 引脚	
		设定值		设定值	
		0	1	0	1
PIOR04	PCLBUZ1	P141	P55	不能使用，必须置“0”（初始值）。	
	INTP5	P16	P12		
PIOR03	PCLBUZ0	P140	P31		
PIOR02	SCLA0	P60	P14	P60	P14
	SDAA0	P61	P15	P61	P15
PIOR01	INTP10	P76	P05	—	—
	INTP11	P77	P06	—	—
	TxD2	P13	P77	P13	—
	RxD2	P14	P76	P14	—
	TxD0	P51	P17	P51	P17
	RxD0	P50	P16	P50	P16
	SI00	P50	P16	P50	—
	SO00	P51	P17	P51	—
	SCK00	P30	P55	P30	—
PIOR00	INTP1	P50	P52	不能使用，必须置“0”（初始值）。	
	INTP2	P51	P53		
	INTP3	P30	P54		
	INTP4	P31	P55		
	INTP8	P74	P42		
	INTP9	P75	P43		

注意 1. 当将 PIOR0 寄存器的 bit1 (PIOR01) 置“1”时，因为 TxD2 和 RxD2 重新定向而 SCL20、SDA20、SI20、SO20 和 SCK20 没有重新定向，所以不能使用 IIC20 和 CS120。但是，即使在置“1”的状态下，只要不使用 UART2 就能使用 CSI21 或者 IIC21。

2. 当将 PIOR0 寄存器的 bit1 (PIOR01) 置“1”时，即使是小于等于 32 引脚的产品，SO00 和 SI00 也被重新定向。但是，SCK00 不被重新定向，所以 CSI00 不能使用。

4.3.9 外围 I/O 重定向寄存器 1（PIOR1）

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。
通过 8 位存储器操作指令设定 PIOR1 寄存器。
在产生复位信号后，此寄存器的值变为“00H”。

图 4-51 外围 I/O 重定向寄存器 1（PIOR1）的格式

地址: F0079H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PIOR1	0	0	0	0	PIOR13	PIOR12	PIOR11	PIOR10

PIOR13	PIOR12	定时器 RJ TRJO0 引脚的选择
0	0	和 P30/INTP3/RTC1HZ/SCK00/SCL00 复用。
0	1	和 P50/SI00/RxD0/TOOLRxD/SDA00 复用。
1	0	和 P00/TI00 复用。
1	1	禁止设定。

PIOR11	PIOR10	定时器 RJ TRJIO0 引脚的选择
0	0	和 P01/TO00 复用。
0	1	和 P31/TI03/TO03/INTP4 复用。
1	0	和 P41 复用（只限于 64 引脚产品）。
1	1	和 P06 复用（只限于 64 引脚产品）。

4.3.10 全局数字输入禁止寄存器（GDIDIS）

在切断 EV_{DD} 的电源时，对于以 EV_{DD} 为电源的输入端口，此寄存器防止输入缓冲器的贯通电流。

当不使用以 EV_{DD} 为电源的全部输入 / 输出端口时，能通过将 GDIDIS 寄存器置“1”来切断 EV_{DD} 的电源，降低功耗。

通过将 GDIDIS0 位置“1”，禁止以 EV_{DD} 为电源的全部输入缓冲器的输入，防止切断 EV_{DD} 的电源时的贯通电流。

通过 1 位或者 8 位存储器操作指令设定 GDIDIS 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

备注 64 引脚产品有 GDIDIS 寄存器。

图 4-52 全局数字输入禁止寄存器（GDIDIS）

地址: F007DH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
GDIDIS	0	0	0	0	0	0	0	GDIDIS0

GDIDIS0	EV _{DD} 电源的输入缓冲器的设定
0	允许输入缓冲器的输入（默认值）。
1	禁止输入缓冲器的输入。防止流向输入缓冲器的贯通电流。

要切断 EV_{DD} 的电源时，必须按照以下步骤进行设定：

- ① 设定为禁止输入缓冲器的输入（GDIDIS0=1）。
- ② 切断 EV_{DD} 的电源。

要重新接通 EV_{DD} 的电源时，必须按照以下步骤进行设定：

- ① 接通 EV_{DD} 的电源。
- ② 设定为允许输入缓冲器的输入（GDIDIS0=0）。

注意 1. 对于以 EV_{DD} 为电源的输入端口，不能输入超过 EV_{DD} 的电压。

2. 如果设定为禁止输入缓冲器的输入（GDIDIS0=1），以 EV_{DD} 为电源的端口的端口寄存器（Pxx）的读取值就为“1”。当端口输出模式寄存器（POMxx）为“1”（N 沟道漏极开路输出（ EV_{DD} 耐压））时，端口寄存器（Pxx）的读取值为“0”。

备注 1. 64 引脚产品有 GDIDIS 寄存器。

2. 即使设定为禁止输入缓冲器的输入（GDIDIS0=1），也能使用以 EV_{DD} 为电源的未使用端口功能的外围功能。

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，并且从引脚输出输出锁存器的内容。

保持被写在输出锁存器中的数据，直到下次写数据为止。

在产生复位信号时清除输出锁存器的数据。

(2) 输入模式的情况

通过传送指令，给输出锁存器写值，但是因输出缓冲器处于 OFF 状态而引脚的状态不发生变化。

保持被写在输出锁存器中的数据，直到下次写数据为止。

在产生复位信号时清除输出锁存器的数据。

4.4.2 输入 / 输出端口的读操作

(1) 输出模式的情况

能通过传送指令读输出锁存器的内容，但是输出锁存器的内容不发生变化。

(2) 输入模式的情况

能通过传送指令读引脚状态，但是输出锁存器的内容不发生变化。

4.4.3 输入 / 输出端口的运算

(1) 输出模式的情况

对输出锁存器的内容进行运算，将结果写到输出锁存器，并且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 $EV_{DD} \leq V_{DD}$ 进行的不同电位（1.8V、2.5V、3V）的对应

在连接不同电位（1.8V、2.5V、3V）的外部设备时，能通过使 EV_{DD} 适合连接对象的电源，连接通用端口的输入 / 输出。

4.4.5 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（PIMxx）和端口输出模式寄存器（POMxx）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、3、5（PIM0、PIM1、PIM3、PIM5），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、3、5、7（POM0、POM1、POM3、POM5、POM7），进行通常的输出（CMOS）和 N 沟道漏极开路（ EV_{DD} 耐压）的切换。

以下说明有关串行接口的连接。

(1) 通过TTL输入缓冲器使用UART0~UART2、CSI00、CSI01、CSI10、CSI20功能的输入端口时的设定步骤

UART0的情况 : P50

UART1的情况 : P03

UART2的情况 : P14

CSI00的情况 : P30、P50

CSI01的情况 : P74、P75

CSI10的情况 : P03、P04

CSI20的情况 : P14、P15

- 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、PIM1、PIM3、PIM5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) 在N沟道漏极开路输出模式中使用UART0～UART2、CSI00、CSI01、CSI10、CSI20功能的输出端口时的设定步骤

UART0的情况：P51

UART1的情况：P02

UART2的情况：P13

CSI00的情况：P30、P51

CSI01的情况：P73、P75

CSI10的情况：P02、P04

CSI20的情况：P13、P15

- ① 通过外部电阻将使用的输出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、POM1、POM3、POM5、POM7寄存器的对应位置“1”，设定为N沟道漏极开路输出（ EV_{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 通过PM0、PM1、PM3、PM5、PM7寄存器设定为输出模式。此时，因为输出数据是高电平，所以引脚处于Hi-Z状态。

(3) 以不同电位（1.8V、2.5V、3V）使用IIC00、IIC01、IIC10、IIC20功能的输入/输出端口时的设定步骤

简易IIC00的情况：P30、P50

简易IIC01的情况：P74、P75

简易IIC10的情况：P03、P04

简易IIC11的情况：P10、P11

简易IIC20的情况：P14、P15

- 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、POM1、POM3、POM5、POM7寄存器的对应位置“1”，设定为N沟道漏极开路输出（ EV_{DD} 耐压）模式。
- ⑤ 将PIM0、PIM1、PIM3、PIM5、PIM7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、PM1、PM3、PM5、PM7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 使用复用功能时的端口相关寄存器的设定

当将端口引脚用作复用功能的引脚时，必须对端口相关寄存器进行如表 4-15 所示的设定。

表 4-15 使用复用功能时的端口相关寄存器的设定 (1/4)

引脚名	复用功能		PIORxx	POMxx	PMCxx	PMxx	Pxx
	名称	输入 / 输出					
P00	TI00	输入	×	×	—	1	×
	(TRJO0)	输出	PIOR13、PIOR12=10B	0	—	0	0
P01	TO00	输出	×	—	—	0	0
	TRJIO0	输入	PIOR13、PIOR12=00B	—	—	1	×
		输出	PIOR13、PIOR12=00B	—	—	0	0
P02	ANI17	输入	×	×	1	1	×
	SO10	输出	PIOR05=0	0/1	0	0	1
	TxD1	输出	PIOR05=0	0/1	0	0	1
P03	ANI16	输入	×	×	1	1	×
	SI10	输入	PIOR05=0	×	0	1	×
	RxD1	输入	PIOR05=0	×	0	1	×
	SDA10	输入 / 输出	PIOR05=0	1	0	0	1
P04	SCK10	输入	PIOR05=0	×	—	1	×
		输出	PIOR05=0	0/1	—	0	1
	SCL10	输出	PIOR05=0	0/1	—	0	1
P05	(INTP10)	输入	PIOR01=1	—	—	1	×
P06	(TRJIO0)	输入	PIOR13、PIOR12=11B	—	—	1	×
		输出	PIOR13、PIOR12=11B	—	—	0	0
	(INTP11)	输入	PIOR01=1	—	—	1	×
P10	SCK11	输入	×	×	—	1	×
		输出	×	0/1	—	0	1
	SCL11	输出	×	1	—	0	1
	TRDIOD1	输入	×	×	—	1	×
		输出	×	0	—	0	0
P11	SI11	输入	×	×	—	1	×
	SDA11	输入 / 输出	×	1	—	0	1
	TRDIOC1	输入	×	×	—	1	×
		输出	×	0	—	0	0
P12	SO11	输出	×	—	0	0	1
	TRDIOB1	输入	×	—	0	1	×
		输出	×	—	0	0	0
	(INTP5)	输入	PIOR04=1	—	0	1	×

备注 1. × : 忽略

PIORxx : 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 : 端口输出锁存器

2. 各引脚和复用功能的关系是 64 引脚产品的情况。当其他产品使用复用功能时，可能分配给不同的引脚，但是 PIORxx、POMxx、PMCxx、PMxx 和 Pxx 的设定相同。

3. 能通过设定外围 I/O 重定向寄存器 0、1 (PIOR0、PIOR1)，分配上表 () 内的功能。

(注在表的末尾)

表 4-15 使用复用功能时的端口模式寄存器和输出锁存器的设定 (2/4)

引脚名	复用功能		PIORxx	POMxx	PMCxx	PMxx	Pxx
	名称	输入 / 输出					
P13	TxD2	输出	PIOR01=0	0/1	0	0	1
	SO20	输出	PIOR01=0	0/1	0	0	1
	TRDIOA1	输入	×	×	0	1	×
		输出	×	0	0	0	0
P14	RxD2	输入	PIOR01=0	×	—	1	×
	SI20	输入	PIOR01=0	×	—	1	×
	SDA20	输入 / 输出	PIOR01=0	1	—	0	1
	TRDIOD0	输入	×	×	—	1	×
		输出	×	0	—	0	0
	(SCLA0)	输入 / 输出	PIOR02=1	1	—	0	0
P15	SCK20	输入	PIOR01=0	×	—	1	×
		输出	PIOR01=0	0/1	—	0	1
	SCL20	输出	PIOR01=0	0/1	—	0	1
	TRDIOB0	输入	×	×	—	1	×
		输出	×	0	—	0	0
	(SDAA0)	输入 / 输出	PIOR02=1	1	—	0	1
P16	TI01	输入	×	—	0	1	×
	TO01	输出	×	—	0	0	0
	INTP5	输入	PIOR04=0	—	0	1	×
	TRDIOC0	输入	×	—	0	1	×
		输出	×	—	0	0	0
	(SI00)	输入	PIOR01=1	—	0	1	×
	(RxD0)	输入	PIOR01=1	—	0	1	×
P17	TI02	输入	×	×	0	1	×
	TO02	输出	×	0	0	0	0
	TRDIOA0	输入	×	×	0	1	×
		输出	×	0	0	0	0
	TRDCLK	输入	×	×	0	1	×
	(SO00)	输出	PIOR01=1	0/1	0	0	1
	(TxD0)	输出	PIOR01=1	0/1	0	0	1
P20	ANI0	输入	×	—	—	1	×
	AV _{REFP}	输入	×	—	—	1	×

备注 1. × : 忽略

PIORxx : 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 : 端口输出锁存器

2. 各引脚和复用功能的关系是 64 引脚产品的情况。当其他产品使用复用功能时, 可能分配给不同的引脚, 但是 PIORxx、POMxx、PMCxx、PMxx 和 Pxx 的设定相同。

3. 能通过设定外围 I/O 重定向寄存器 0、1 (PIOR0、PIOR1), 分配上表 () 内的功能。

(注在表的末尾)

表 4-15 使用复用功能时的端口模式寄存器和输出锁存器的设定 (3/4)

引脚名	复用功能		PIORxx	POMxx	PMCxx	PMxx	Pxx
	名称	输入 / 输出					
P21	ANI1	输入	×	—	—	1	×
	AV _{REFM}	输入	×	—	—	1	×
P22	ANI2	输入	×	—	—	1	×
P23	ANI3	输入	×	—	—	1	×
P24 ~ P27	ANI4 ~ ANI7	输入	×	—	—	1	×
P30	INTP3	输入	PIOR05=0	×	—	1	×
	RTC1HZ	输出	×	0	—	0	0
	SCK00	输入	PIOR01=0	×	—	1	×
		输出	PIOR01=0	0/1	—	0	1
	SCL00	输出	PIOR01=0	0/1	—	0	1
	TRJ00	输出	PIOR13、PIOR12=00B	0	—	0	0
P31	TI03	输入	×	—	—	1	×
	TO03	输出	×	—	—	0	0
	INTP4	输入	PIOR05=0	—	—	1	×
	(PCLBUZ0)	输出	PIOR03=1	—	—	0	0
	(TRJIO0)	输入	PIOR13、PIOR12=01B	—	—	1	×
		输出	PIOR13、PIOR12=01B	—	—	0	0
P40	TOOL0	输入 / 输出	×	—	—	×	×
P41	(TRJIO0)	输入	PIOR11、PIOR10=10B	—	—	1	×
		输出	PIOR11、PIOR10=10B	—	—	0	0
P50	SI00	输入	PIOR01=0	×	—	1	×
	RxD0	输入	PIOR01=0	×	—	1	×
	TOOLRxD	输入	×	×	—	1	×
	SDA00	输入 / 输出	PIOR01=0	1	—	0	1
	(TRJ00)	输出	PIOR13、PIOR12=01B	0	—	0	0
P51	SO00	输出	PIOR01=0	0/1	—	0	1
	TxD0	输出	PIOR01=0	0/1	—	0	1
	TOOLTxD	输出	×	0/1	—	0	1
P52	(INTP1)	输入	PIOR00=1	—	—	1	×
P53	(INTP2)	输入	PIOR00=1	—	—	1	×
P54	(INTP3)	输入	PIOR00=1	—	—	1	×
P55	(PCLBUZ1)	输出	PIOR04=1	0	—	0	0
	(SCK00)	输入	PIOR01=1	×	—	1	×
		输出	PIOR01=1	0/1	—	0	1
	(INTP4)	输入	PIOR00=1	—	—	1	×
P60	SCLA0	输入 / 输出	PIOR02=0	—	—	0	0
P61	SDAA0	输入 / 输出	PIOR02=0	—	—	0	0
P62	SSI00	输入	×	—	—	1	×

备注 1. × : 忽略
 PIORxx : 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 : 端口输出锁存器

- 各引脚和复用功能的关系是 64 引脚产品的情况。当其他产品使用复用功能时，可能分配给不同的引脚，但是 PIORxx、POMxx、PMCxx、PMxx 和 Pxx 的设定相同。
- 能通过设定外围 I/O 重定向寄存器 0、1 (PIOR0、PIOR1)，分配上表 () 内的功能。
(注在表的末尾)

表 4-15 使用复用功能时的端口模式寄存器和输出锁存器的设定 (4/4)

引脚名	复用功能		PIORxx	POMxx	PMCxx	PMxx	Pxx
	名称	输入 / 输出					
P70	KR0	输入	×	—	—	1	×
	SCK21	输入	×	—	—	1	×
		输出	×	—	—	0	1
	SCL21	输出	×	—	—	0	1
P71	KR1	输入	×	×	—	1	×
	SI21	输入	×	×	—	1	×
	SDA21	输入 / 输出	×	1	—	0	1
P72	KR2	输入	×	—	—	1	×
	SO21	输出	×	—	—	0	1
P73	KR3	输入	×	—	—	1	×
P74	KR4	输入	×	×	—	1	×
	INTP8	输入	PIOR05=0	×	—	1	×
	SI01	输入	×	—	—	1	×
	SDA01	输入 / 输出	×	—	—	0	1
P75	KR5	输入	×	—	—	1	×
	INTP9	输入	PIOR05=0	—	—	1	×
	SCK01	输入	×	—	—	1	×
		输出	×	—	—	0	1
	SCL01	输出	×	—	—	0	1
P76	KR6	输入	×	—	—	1	×
	INTP10	输入	PIOR01=0	—	—	1	×
	(RxD2)	输入	PIOR01=1	—	—	1	×
P77	KR7	输入	×	—	—	1	×
	INTP11	输入	PIOR01=0	—	—	1	×
	(Tx/D2)	输出	PIOR01=1	—	—	0	1
P120	ANI19	输入	×	—	1	1	×
P137	INTP0	输入	×	—	—	—	×
P140	PCLBUZ0	输出	PIOR03=0	—	—	0	0
	INTP6	输入	PIOR05=0	—	—	1	×
P141	PCLBUZ1	输出	PIOR04=0	—	—	0	0
	INTP7	输入	PIOR05=0	—	—	1	×
P147	ANI18	输入	×	—	1	1	×

备注 1. × : 忽略

PIORxx : 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 : 端口输出锁存器

2. 各引脚和复用功能的关系是 64 引脚产品的情况。当其他产品使用复用功能时, 可能分配给不同的引脚, 但是 PIORxx、POMxx、PMCxx、PMxx 和 Pxx 的设定相同。

3. 能通过设定外围 I/O 重定向寄存器 0、1 (PIOR0、PIOR1), 分配上表 () 内的功能。

(注在表的末尾)

注 1. ANI16/P03、ANI17/P02、ANI18/P147、ANI19/P120 的引脚功能取决于端口模式控制寄存器 0、12、14（PMC0、PMC12、PMC14）、模拟输入通道指定寄存器（ADS）以及 PM0、PM12、PM14 寄存器的设定。

表 4-16 ANI16/P03、ANI17/P02、ANI18/P147、ANI19/P120 引脚功能的设定

PMC0、PMC12、 PMC14 寄存器	PM0、PM12、 PM14 寄存器	ADS 寄存器	ANI16/P03、ANI17/P02、 ANI18/P147、ANI19/P120 引脚
数字输入 / 输出的选择	输入模式	—	数字输入
	输出模式	—	数字输出
模拟输入的选择	输入模式	选择 ANI。	模拟输入（转换对象）
		不选择 ANI。	模拟输入（非转换对象）
	输出模式	选择 ANI。	禁止设定。
		不选择 ANI。	

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

表 4-17 P20/ANI0 ~ P27/ANI7 引脚功能的设定

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

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 R7F0C014B2D、R7F0C014L2D 内部，位存储器操作指令按照以下顺序执行：

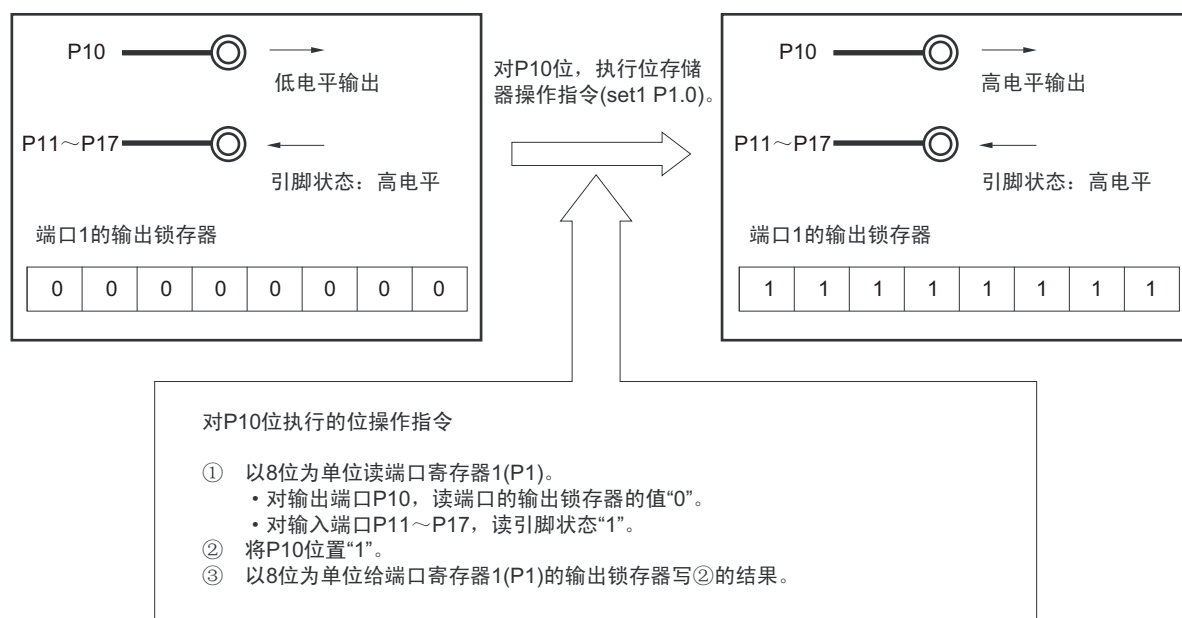
- <1> 以 8 位为单位读 Pn 寄存器。
- <2> 操作 1 位对象位。
- <3> 以 8 位为单位写 Pn 寄存器。

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 4-53 位存储器操作指令（P10 的情况）



4.6.2 有关 64 引脚以外的产品的引脚设定的注意事项

对于 64 引脚以外的产品，可能给 P15 引脚分配了多个复用输出功能。
此时，除了表 4-15 所示的设定以外，还需要将不使用的复用功能的输出置初始值。
具体对象和处理方法如下所示。

(1) 32 引脚产品的 P15/PCLBUZ1/SCK20/SCL20/TRDI0B0/(SDAA0)

- 当用作 PCLBUZ1 时，SCK20/SCL20 输出“1”，TRDI0B0 输出“0”。
必须将串行输出寄存器 1 (SO1) 的 SO10 位、串行输出允许寄存器 1 (SOE1) 的 SOE10 位和串行通道允许状态寄存器 1 (SE1) 的 SE10 位置初始值。
必须通过定时器 RD 的输出主控允许寄存器 1 (TRDOER1) 将 TRDI0B0 引脚的输出控制位置初始值。
- 当用作 SCK20/SCL20 时，PCLBUZ1 输出“0”，TRDI0B0 输出“0”。
必须将时钟输出选择寄存器 1 (CKS1) 的 PCLOE1 位置初始值。
必须通过定时器 RD 的输出主控允许寄存器 1 (TRDOER1) 将 TRDI0B0 引脚的输出控制位置初始值。
- 当用作 TRDI0B0 输出时，SCK20/SCL20 输出“1”，PCLBUZ1 输出“0”。
必须将时钟输出选择寄存器 1 (CKS1) 的 PCLOE1 位置初始值。
必须将串行输出寄存器 1 (SO1) 的 SO10 位、串行输出允许寄存器 1 (SOE1) 的 SOE10 位和串行通道允许状态寄存器 1 (SE1) 的 SE10 位置初始值。

第 5 章 时钟发生电路

用于主系统时钟的谐振器连接引脚 / 外部时钟输入引脚、用于副系统时钟的谐振器连接引脚 / 外部时钟输入引脚的有无因产品而不同。

	32 引脚产品	64 引脚产品
X1 引脚和 X2 引脚	○	○
EXCLK 引脚	○	○
XT1 引脚和 XT2 引脚	—	○
EXCLKS 引脚	—	○

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（高速 OCO）

能通过选项字节（000C2H）从 $f_{\text{HOCO}}=64\text{MHz}$ 、48MHz、32MHz、24MHz、16MHz、12MHz、8MHz、6MHz、4MHz、3MHz、2MHz 和 1MHz(TYP.) 中选择频率进行振荡。当选择 64MHz 或者 48MHz 作为 f_{HOCO} 时， f_{IH} 分别为 32MHz 或者 24MHz；当选择 32MHz 或者更低的频率作为 f_{HOCO} 时， f_{IH} 不被分频而和 f_{HOCO} 的频率相同。在解除复位后，CPU 一定以此高速内部振荡器时钟开始运行。能通过执行 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 5-10 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	32	48	64
$2.7\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$	○	○	○	○	○	○	○	○	○	○	○	○
$2.4\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$	○	○	○	○	○	○	○	○	—	—	—	—
$1.8\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$	○	○	○	○	○	○	—	—	—	—	—	—
$1.6\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$	○	○	○	○	—	—	—	—	—	—	—	—

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

(2) 副系统时钟

- XT1 振荡电路

能通过给 XT1 引脚和 XT2 引脚连接 32.768kHz 的谐振器使 $f_{XT}=32.768\text{kHz}$ 的时钟振荡，并且能通过设定 XTSTOP 位（时钟运行状态控制寄存器（CSC）的 bit6）使振荡停止。

另外，能由 EXCLKS/XT2/P124 引脚提供外部副系统时钟（ $f_{EXS}=32.768\text{kHz}$ ），并且能通过设定 XTSTOP 位将外部副系统时钟的输入置为无效。

(3) 低速内部振荡器时钟（低速 OCO）

能使 $f_{LL}=15\text{kHz}$ (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
- 实时时钟
- 12 位间隔定时器
- 定时器 RJ

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。

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

备注	f_X	: X1 时钟振荡频率
	f_{HOCO}	: 高速内部振荡器的时钟频率（最大 64MHz）
	f_{IH}	: 高速内部振荡器的时钟频率（最大 32MHz）注
	f_{EX}	: 外部主系统时钟频率
	f_{XT}	: XT1 时钟振荡频率
	f_{EXS}	: 外部副系统时钟频率
	f_{LL}	: 低速内部振荡器的时钟频率

注 由硬件进行控制，当将 f_{HOCO} 设定为 64MHz 或者 48MHz 时，使时钟频率和 f_{HOCO} 的 2 分频相同；当设定为 32MHz 或者更低的频率时，使时钟频率和 f_{HOCO} 相同。要给定定时器 RD 提供 64MHz 或者 48MHz 时，必须将 f_{CLK} 设定为 f_{IH} 。

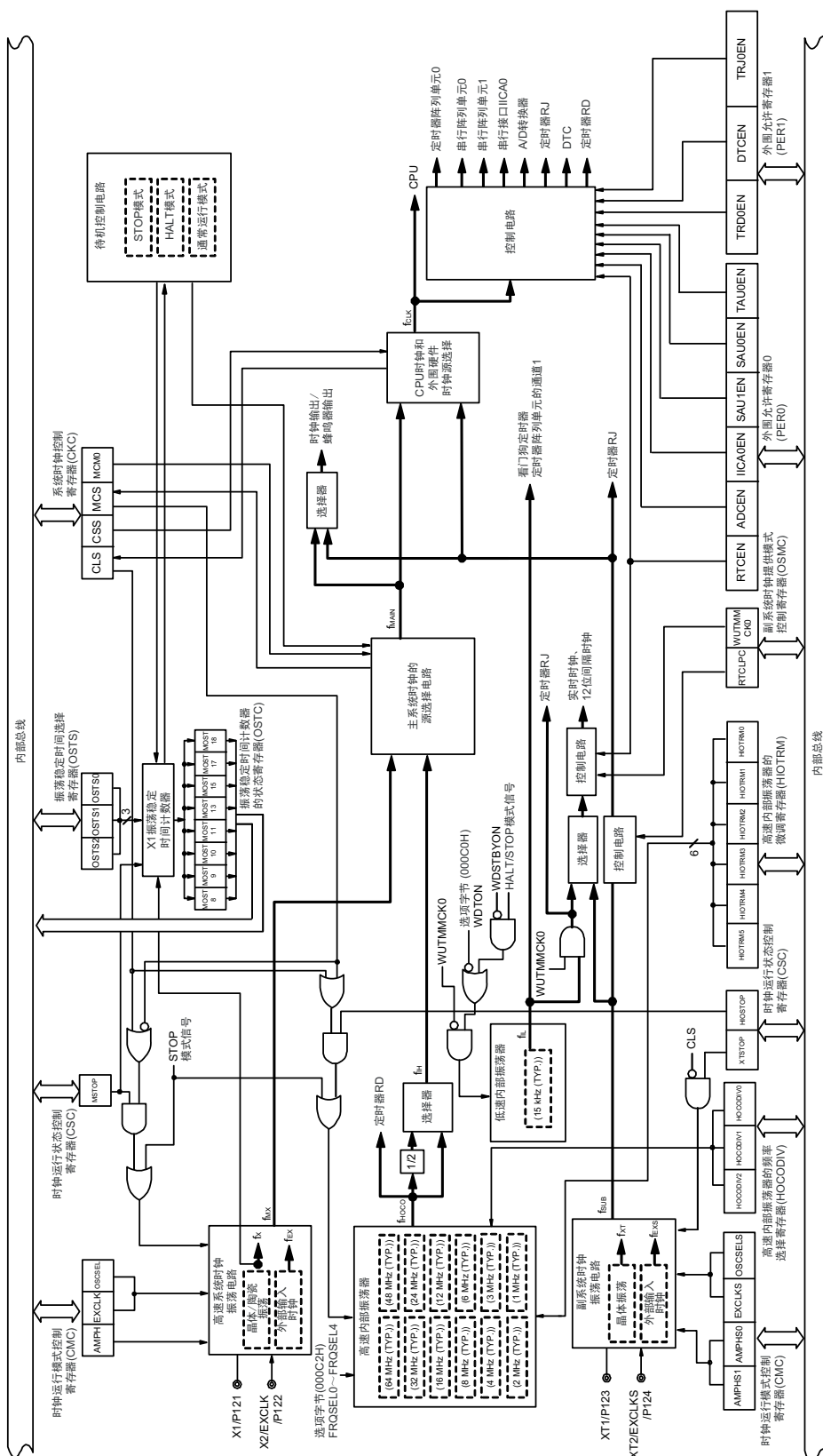
5.2 时钟发生电路的结构

时钟发生电路由以下硬件构成。

表 5-1 时钟发生电路的结构

项目	结构
控制寄存器	时钟运行模式控制寄存器 (CMC) 系统时钟控制寄存器 (CKC) 时钟运行状态控制寄存器 (CSC) 振荡稳定时间计数器的状态寄存器 (OSTC) 振荡稳定时间选择寄存器 (OSTS) 外围允许寄存器 0、1 (PER0、PER1) 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 XT1 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_{HOCO}	: 高速内部振荡器的时钟频率 (最大 64MHz)
	f_{IH}	: 高速内部振荡器的时钟频率 (最大 32MHz) 注
	f_{EX}	: 外部主系统时钟频率
	f_{MX}	: 高速系统时钟频率
	f_{MAIN}	: 主系统时钟频率
	f_{XT}	: XT1 时钟振荡频率
	f_{EXS}	: 外部副系统时钟频率
	f_{SUB}	: 副系统时钟频率
	f_{CLK}	: CPU/外围硬件的时钟频率
	f_{IL}	: 低速内部振荡器的时钟频率

注 由硬件进行控制, 当将 f_{HOCO} 设定为 64MHz 或者 48MHz 时, 使时钟频率和 f_{HOCO} 的 2 分频相同; 当设定为 32MHz 或者更低的频率时, 使时钟频率和 f_{HOCO} 相同。要给定定时器 RD 提供 64MHz 或者 48MHz 时, 必须将 f_{CLK} 设定为 f_{IH} 。

5.3 控制时钟发生电路的寄存器

通过以下寄存器控制时钟发生电路。

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

注意 分配的寄存器和位因产品而不同。必须给未分配的位设定初始值。

5.3.1 时钟运行模式控制寄存器 (CMC)

这是设定 X1/P121、X2/EXCLK/P122、XT1/P123、XT2/EXCLKS/P124 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	EXCLKS	OSCSELS	0	AMPHS1	AMPHS0	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

EXCLKS	OSCSELS	副系统时钟 引脚的运行模式	XT1/P123 引脚	XT2/EXCLKS/P124 引脚
0	0	输入端口模式	输入端口	
0	1	XT1 振荡模式	连接晶体谐振器。	
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

AMPHS1	AMPHS0	XT1 振荡电路的振荡模式选择
0	0	低功耗振荡（默认）
0	1	通常的振荡
1	0	超低功耗振荡
1	1	禁止设定。

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 或者 XT1 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} 或者 f_{SUB} 前的状态）下设定 AMPH 位、AMPHS1 位和 AMPHS0 位。
5. 必须通过软件对 f_{XT} 的振荡稳定时间进行计数。
6. 系统时钟的频率上限为 32MHz，但是 X1 振荡电路的频率上限为 20MHz。

注意 7. 为了实现低功耗，XT1 振荡电路是低增幅电路。必须在设计时注意以下几点：

- 引脚和电路板含有寄生电容。因此，必须通过实际使用的电路板进行振荡评估，确认是否有问题。
- 在将 XT1 振荡电路的模式用于超低功耗振荡（AMPHS1、AMPHS0=1、0）时，必须在对“5.7 谐振器和振荡电路常数”记载的谐振器进行充分评估后使用。
- 必须尽量缩短 XT1 引脚、XT2 引脚和谐振器之间的布线，减小寄生电容和布线电阻。尤其在选择超低功耗振荡（AMPHS1、AMPHS0=1、0）时要注意。
- 必须使用寄生电容和布线电阻小的电路板材料构成电路。
- 必须尽量在 XT1 振荡电路的附近配置和 V_{SS} 同电位的接地图形。
- XT1 引脚、XT2 引脚和谐振器的信号线不能和其他信号线交叉，并且不能接近有变化的大电流流过的布线。
- 在高湿度环境中，因电路板的吸湿或者结露可能导致 XT1 引脚和 XT2 引脚之间的阻抗下降而影响振荡。在这样的环境中使用时，必须对电路板进行涂层等防潮措施。
- 在对电路板进行涂层处理时，必须在 XT1 引脚和 XT2 引脚之间使用不产生电容或者漏电流的材料。

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

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

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

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

注 1. bit7 和 bit5 是只读位。

2. 禁止在将 CSS 位置“1”的状态下更改 MCM0 位的值。

备注 f_{HOCO} : 高速内部振荡器的时钟频率 (最大 64MHz)
 f_{IH} : 高速内部振荡器的时钟频率 (最大 32MHz) 注
 f_{MX} : 高速系统时钟频率
 f_{MAIN} : 主系统时钟频率
 f_{SUB} : 副系统时钟频率

注 由硬件进行控制, 当将 f_{HOCO} 设定为 64MHz 或者 48MHz 时, 使时钟频率和 f_{HOCO} 的 2 分频相同; 当设定为 32MHz 或者更低的频率时, 使时钟频率和 f_{HOCO} 相同。要给定定时器 RD 提供 64MHz 或者 48MHz 时, 必须将 f_{CLK} 设定为 f_{IH} 。

注意 1. 必须将 bit0 ~ 3 置“0”。

- 给 CPU 和外围硬件提供 CSS 位设定的时钟。如果更改 CPU 时钟, 就同时更改外围硬件的时钟 (实时时钟、12 位间隔定时器、时钟输出 / 蜂鸣器输出和看门狗定时器除外)。因此, 如果要更改 CPU/ 外围硬件的时钟, 就必须停止各外围功能。
- 如果将副系统时钟用作外围硬件时钟, 就无法保证 A/D 转换器和 IICA 的运行。有关外围硬件的运行特性, 请参照各外围硬件的章节和“第 31 章 电特性”。
- 要选择 f_{HOCO} 作为定时器 RD 的计数源时, 必须在将外围允许寄存器 1 (PER1) 的 bit4 (TRD0EN) 置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟, 就必须在清除外围允许寄存器 1 (PER1) 的 bit4 (TRD0EN) 后进行更改。

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、高速内部振荡器时钟和副系统时钟（低速内部振荡器时钟除外）运行的寄存器。

通过 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	XTSTOP	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	

XTSTOP	副系统时钟的运行控制		
	XT1 振荡模式	外部时钟输入模式	输入端口模式
0	XT1 振荡电路运行	EXCLKS 引脚的外部时钟有效	输入端口
1	XT1 振荡电路停止	EXCLKS 引脚的外部时钟无效	

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

- 注意 1. 在解除复位后，必须在设定时钟运行模式控制寄存器（CMC）后设定 CSC 寄存器。
2. 在解除复位后并且在将 MSTOP 位置“0”前，必须设定振荡稳定时间选择寄存器（OSTS）。但是，当以初始值使用 OSTS 寄存器时，不需要设定 OSTS 寄存器。
3. 要通过设定 MSTOP 位开始 X1 振荡时，必须通过振荡稳定时间计数器的状态寄存器（OSTC）确认 X1 时钟的振荡稳定时间。
4. 要通过设定 XTSTOP 位开始 XT1 振荡时，必须通过软件等待副系统时钟所需的振荡稳定时间。
5. 不能通过 CSC 寄存器停止被选择为 CPU/ 外围硬件时钟（ f_{CLK} ）的时钟。
6. 有关用于停止时钟振荡（外部时钟输入无效）的寄存器标志设定和停止前的条件，请参照表 5-2。

表 5-2 时钟停止方法

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

5.3.4 振荡稳定时间计数器的状态寄存器（OSTC）

这是表示 X1 时钟的振荡稳定时间计数器计数状态的寄存器。

能在以下情况下确认 X1 时钟的振荡稳定时间：

- 当 CPU 时钟为高速内部振荡器时钟或者副系统时钟并且开始 X1 时钟的振荡时
- 当 CPU 时钟为高速内部振荡器时钟并且在 X1 时钟振荡的状态下转移到 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

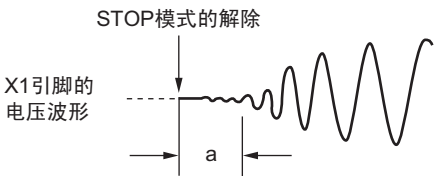
图 5-5 振荡稳定时间计数器的状态寄存器（OSTC）的格式

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	MOST	MOST	MOST	MOST	MOST	MOST	MOST	振荡稳定时间状态		
8	9	10	11	13	15	17	18		$f_X=10\text{MHz}$	$f_X=20\text{MHz}$
0	0	0	0	0	0	0	0	小于 $2^8/f_X$	小于 $25.6\mu\text{s}$	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或者副系统时钟并且要开始 X1 时钟的振荡时
 - 当 CPU 时钟为高速内部振荡器时钟并且在 X1 时钟振荡的状态下转移到 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或者副系统时钟切换到 X1 时钟，或者如果 CPU 时钟为高速内部振荡器时钟并且在 X1 时钟振荡的状态下转移到 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

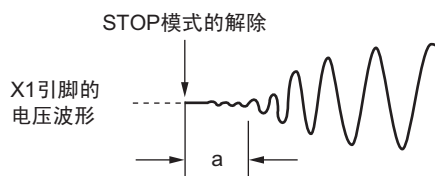
注意 1. 要更改 OSTS 寄存器的设定时，必须在将时钟运行状态控制寄存器（CSC）的 MSTOP 位置“0”前进行更改。

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或者副系统时钟并且要开始 X1 时钟的振荡时
- 当 CPU 时钟为高速内部振荡器时钟并且在 X1 时钟振荡的状态下转移到 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、1（PER0、PER1）

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

当使用由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的位置“1”。

- 实时时钟、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
- 定时器RD
- DTC
- 定时器RJ

通过 1 位或者 8 位存储器操作指令设定 PER0 寄存器和 PER1 寄存器。
在产生复位信号后，这些寄存器的值变为“00H”。

图 5-7 外围允许寄存器 0（PER0）的格式 (1/3)

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

RTCEN	提供实时时钟（RTC）和 12 位间隔定时器的输入时钟的控制
0	停止提供输入时钟。 • 不能写实时时钟（RTC）和 12 位间隔定时器使用的 SFR。 • 实时时钟（RTC）和 12 位间隔定时器处于复位状态。
1	提供输入时钟。 • 能读写实时时钟（RTC）和 12 位间隔定时器使用的 SFR。

注意 必须将以下的位置“0”。
bit1和bit6

图 5-7 外围允许寄存器 0（PER0）的格式 (2/3)

地址: F00F0H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PER0	RTCEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

ADCEN	提供 A/D 转换器的输入时钟的控制
0	停止提供输入时钟。 - 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	提供串行接口 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。

SAU1EN	提供串行阵列单元 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	提供串行阵列单元 0 的输入时钟的控制
0	停止提供输入时钟。 - 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。

注意 必须将以下的位置“0”。

bit1 和 bit6

图 5-7 外围允许寄存器 0（PER0）的格式 (3/3)

地址: F00F0H

复位后: 00H

R/W

符号

7

6

5

4

3

2

1

0

PER0

RTCEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

TAU0EN	提供定时器阵列单元 0 的输入时钟的控制
0	停止提供输入时钟。 - 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。

注意

必须将以下的位置“0”。

bit1 和 bit6

图 5-8 外围允许寄存器 1（PER1）的格式 (1/2)

地址: F007AH

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
PER1	0	0	0	TRD0EN 注	DTCEN	0	0	TRJ0EN

TRD0EN 注	提供定时器 RD 的输入时钟的控制
0	停止提供输入时钟。 - 不能写定时器 RD 使用的 SFR。 - 定时器 RD 处于复位状态。
1	提供输入时钟。 能读写定时器 RD 使用的 SFR。

注 当用户选项字节（000C2H）的 FRQSEL4 位为“1”时，必须在将外围允许寄存器 1（PER1）的 bit4（TRD0EN）置位前将 f_{CLK} 设定为 f_{IH} 。
如果要 将 f_{CLK} 改为 f_{IH} 以外的时钟，就必须在清除外围允许寄存器 1（PER1）的 bit4（TRD0EN）后进行更改。

注意 必须将以下的位置“0”。
bit1、2、5～7

图 5-8 外围允许寄存器 1（PER1）的格式 (2/2)

地址: F007AH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PER1	0	0	0	TRD0EN 注	DTCEN	0	0	TRJ0EN

DTCEN	提供 DTC 的输入时钟的控制
0	停止提供输入时钟。 • DTC 不能运行。
1	提供输入时钟。 • DTC 能运行。

TRJ0EN	提供定时器 RJ0 的输入时钟的控制
0	停止提供输入时钟。 • 不能写定时器 RJ0 使用的 SFR。 • 定时器 RJ0 处于复位状态。
1	提供输入时钟。 • 能读写定时器 RJ0 使用的 SFR。

注意 必须将以下的位置“0”。

bit1、2、5~7

5.3.7 副系统时钟提供模式控制寄存器（OSMC）

OSMC 寄存器是通过停止不需要的时钟功能来降低功耗的寄存器。

如果将 RTCLPC 位置“1”，就在 STOP 模式或者 CPU 以副系统时钟运行的 HALT 模式中停止给实时时钟和 12 位间隔定时器以外的外围功能提供时钟，因此能降低功耗。

另外，能通过 OSMC 寄存器选择实时时钟和 12 位间隔定时器的运行时钟。

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 5-9 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	RTCLPC	0	0	WUTMMCK0	0	0	0	0

RTCLPC	STOP 模式和 CPU 以副系统时钟运行的 HALT 模式中的设定
0	允许给外围功能提供副系统时钟 (有关允许运行的外围功能, 请参照表 20-1 ~ 表 20-3)。
1	停止给实时时钟和 12 位间隔定时器以外的外围功能提供副系统时钟。

WUTMMCK0	实时时钟、12 位间隔定时器和定时器 RJ 的运行时钟的选择
0	- 副系统时钟为实时时钟和 12 位间隔定时器的运行时钟。 - 不能选择低速内部振荡器作为定时器 RJ 的计数源。
1	- 低速内部振荡器时钟为实时时钟和 12 位间隔定时器的运行时钟。 - 能选择低速内部振荡器或者副系统时钟作为定时器 RJ 的计数源。

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 FRQSEL4 位和 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 5-10 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			
			FRQSEL4=0		FRQSEL4=1	
			FRQSEL3=0	FRQSEL3=1	FRQSEL3=0	FRQSEL3=1
0	0	0	$f_{IH}=24\text{MHz}$	$f_{IH}=32\text{MHz}$	$f_{IH}=24\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=32\text{MHz}$ $f_{HOCO}=64\text{MHz}$
0	0	1	$f_{IH}=12\text{MHz}$	$f_{IH}=16\text{MHz}$	$f_{IH}=12\text{MHz}$ $f_{HOCO}=24\text{MHz}$	$f_{IH}=16\text{MHz}$ $f_{HOCO}=32\text{MHz}$
0	1	0	$f_{IH}=6\text{MHz}$	$f_{IH}=8\text{MHz}$	$f_{IH}=6\text{MHz}$ $f_{HOCO}=12\text{MHz}$	$f_{IH}=8\text{MHz}$ $f_{HOCO}=16\text{MHz}$
0	1	1	$f_{IH}=3\text{MHz}$	$f_{IH}=4\text{MHz}$	$f_{IH}=3\text{MHz}$ $f_{HOCO}=6\text{MHz}$	$f_{IH}=4\text{MHz}$ $f_{HOCO}=8\text{MHz}$
1	0	0	禁止设定。	$f_{IH}=2\text{MHz}$	禁止设定。	$f_{IH}=2\text{MHz}$ $f_{HOCO}=4\text{MHz}$
1	0	1	禁止设定。	$f_{IH}=1\text{MHz}$	禁止设定。	$f_{IH}=1\text{MHz}$ $f_{HOCO}=2\text{MHz}$
上述以外			禁止设定。			

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

选项字节（000C2H）的值		闪存运行模式	工作频率范围	工作电压范围
CMODE1	CMODE0			
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～32MHz	2.7V～5.5V
上述以外		禁止设定。		

2. 必须在选择高速内部振荡器时钟（ f_{IH} ）作为 CPU/ 外围硬件时钟（ f_{CLK} ）的状态下设定 HOCODIV 寄存器。

3. 在通过 HOCODIV 寄存器更改频率后，经过以下转移时间之后进行频率切换：

- 以更改前的频率，最多进行 3 个时钟的运行。
- 以更改后的频率，最多等待 3 个 CPU/ 外围硬件的时钟。

5.3.9 高速内部振荡器的微调寄存器（HIOTRM）

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

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

图 5-11 高速内部振荡器的微调寄存器（HIOTRM）的格式

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

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

注 复位值是产品发货时的调整值。

备注 1. HIOTRM 寄存器的每 1 位能对高速内部振荡器的时钟精度进行 0.05% 左右的校正。
2. 有关 HIOTRM 寄存器的使用例子，请参照 RL78 MCU 系列高速内部振荡器时钟频率校正的应用说明（R01AN0464）。

5.4 系统时钟振荡电路

5.4.1 X1 振荡电路

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

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

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

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

而且，当也不用作输入端口时，请参照“表 2-3 各未使用引脚的处理”。

X1 振荡电路的外接电路例子如图 5-12 所示。

图 5-12 X1 振荡电路的外接电路例子



注意事项如下页所示。

5.4.2 XT1 振荡电路

XT1 振荡电路通过连接 XT1 引脚和 XT2 引脚的晶体谐振器（32.768kHz(TYP.)) 进行振荡。

当使用 XT1 振荡电路时，必须将时钟运行模式控制寄存器（CMC）的 bit4（OSCSELS）置“1”。

也能输入外部时钟，此时必须给 EXCLKS 引脚输入时钟信号。

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

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

当不使用 XT1 振荡电路时，必须设定为输入端口模式（EXCLKS、OSCSELS=0、0）。

而且，当也不用作输入端口时，请参照“表 2-3 各未使用引脚的处理”。

XT1 振荡电路的外接电路例子如图 5-13 所示。

图 5-13 XT1 振荡电路的外接电路例子



注意 当使用 X1 振荡电路和 XT1 振荡电路时，为了避免布线电容等的影响，必须通过以下方法对图 5-12 和图 5-13 中的虚线部分进行布线：

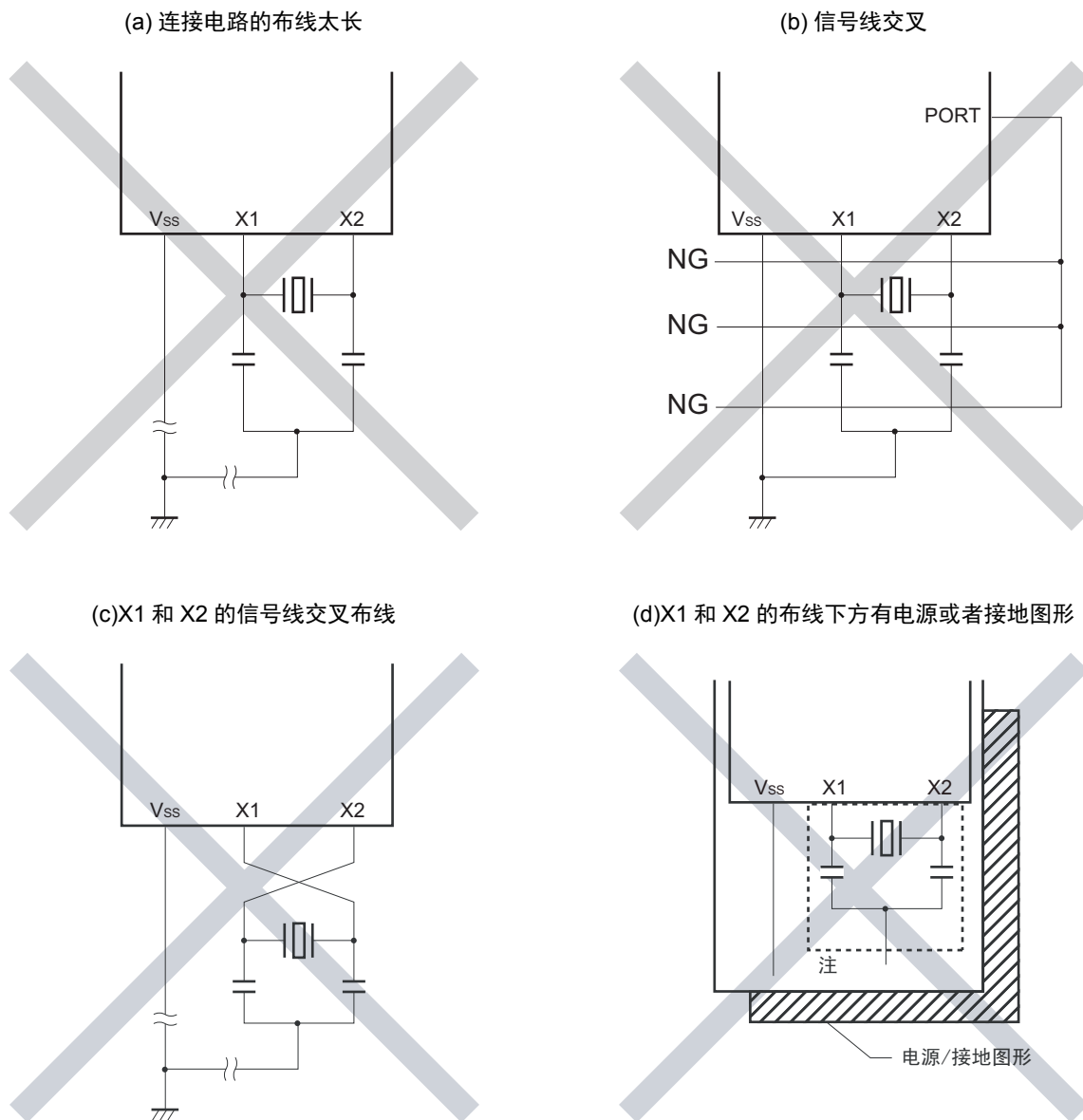
- 必须尽量缩短布线。
- 不能和其他的信号线交叉，并且不能接近有变化的大电流流过的布线。
- 必须始终保持振荡电路的电容器接地点和 V_{SS} 同电位，而且不能给大电流流过的接地图形接地。
- 不能从振荡电路取出信号。

尤其是为了实现低功耗，XT1 振荡电路是低增幅电路。必须在设计时注意以下几点：

- 引脚和电路板含有寄生电容。因此，必须通过实际使用的电路板进行振荡评估，确认是否有问题。
- 在将 XT1 振荡电路的模式用于超低功耗振荡（AMPHS1、AMPHS0=1、0）时，必须在对“5.7 谐振器和振荡电路常数”记载的谐振器进行充分评估后使用。
- 必须尽量缩短 XT1 引脚、XT2 引脚和谐振器之间的布线，减小寄生电容和布线电阻。尤其在超低功耗振荡（AMPHS1、AMPHS0=1、0）时要注意。
- 必须使用寄生电容和布线电阻小的电路板材料构成电路。
- 必须尽量在 XT1 振荡电路的附近配置和 V_{SS} 同电位的接地图形。
- XT1 引脚、XT2 引脚和谐振器的信号线不能和其他信号线交叉，并且不能接近有变化的大电流流过的布线。
- 在高湿度环境中，因电路板的吸湿或者结露可能导致 XT1 引脚和 XT2 引脚之间的阻抗下降而影响振荡。在这样的环境中使用时，必须对电路板进行涂层等防潮措施。
- 在对电路板进行涂层处理时，必须在 XT1 引脚和 XT2 引脚之间使用不产生电容或者漏电流的材料。

不正确的谐振器连接例子如图 5-14 所示。

图 5-14 不正确的谐振器连接例子 (1/2)

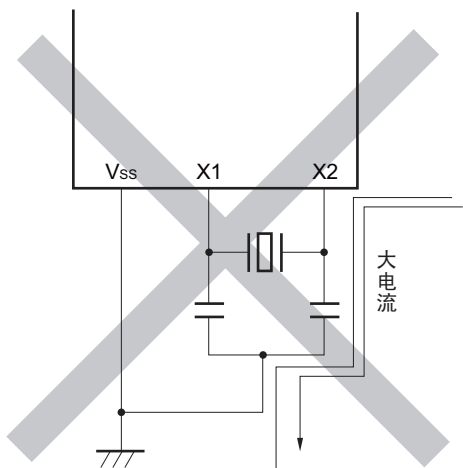
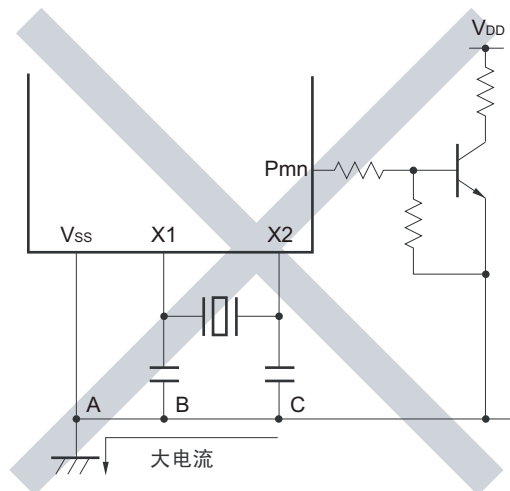


注 在多层板或者双面板中，不能在 X1 引脚、X2 引脚和谐振器的布线区（图中虚线部分）下方配置电源或者接地图形。布线不能产生电容成分而影响振荡特性。

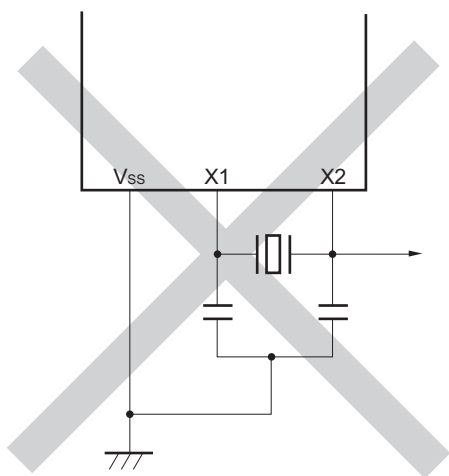
备注 在使用副系统时钟的情况下，请阅读时分别用 XT1 和 XT2 代替 X1 和 X2，并且在 XT2 侧插入串联电阻。

图 5-14 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



注意 当 X2 和 XT1 并行布线时，X2 的串扰噪声会叠加到 XT1 而导致误动作。

备注 在使用副系统时钟的情况下，请阅读时分别用 XT1 和 XT2 代替 X1 和 X2，并且在 XT2 侧插入串联电阻。

5.4.3 高速内部振荡器

R7F0C014B2D、R7F0C014L2D 内置高速内部振荡器。能通过选项字节（000C2H）从 64MHz、48MHz、32MHz、24MHz、16MHz、12MHz、8MHz、6MHz、4MHz、3MHz、2MHz 和 1MHz 中选择频率。当选择 64MHz 或者 48MHz 时，CPU 时钟为 2 分频时钟。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。

5.4.4 低速内部振荡器

R7F0C014B2D、R7F0C014L2D 内置低速内部振荡器。

低速内部振荡器时钟用作看门狗定时器、实时时钟、12 位间隔定时器的时钟和定时器 RJ 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，当看门狗定时器运行并且只有在 WUTMMCK0 位为“0”时，WDSTBYON 位为“0”并且在 HALT 模式、STOP 模式和 SNOOZE 模式中低速内部振荡器停止振荡。当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。

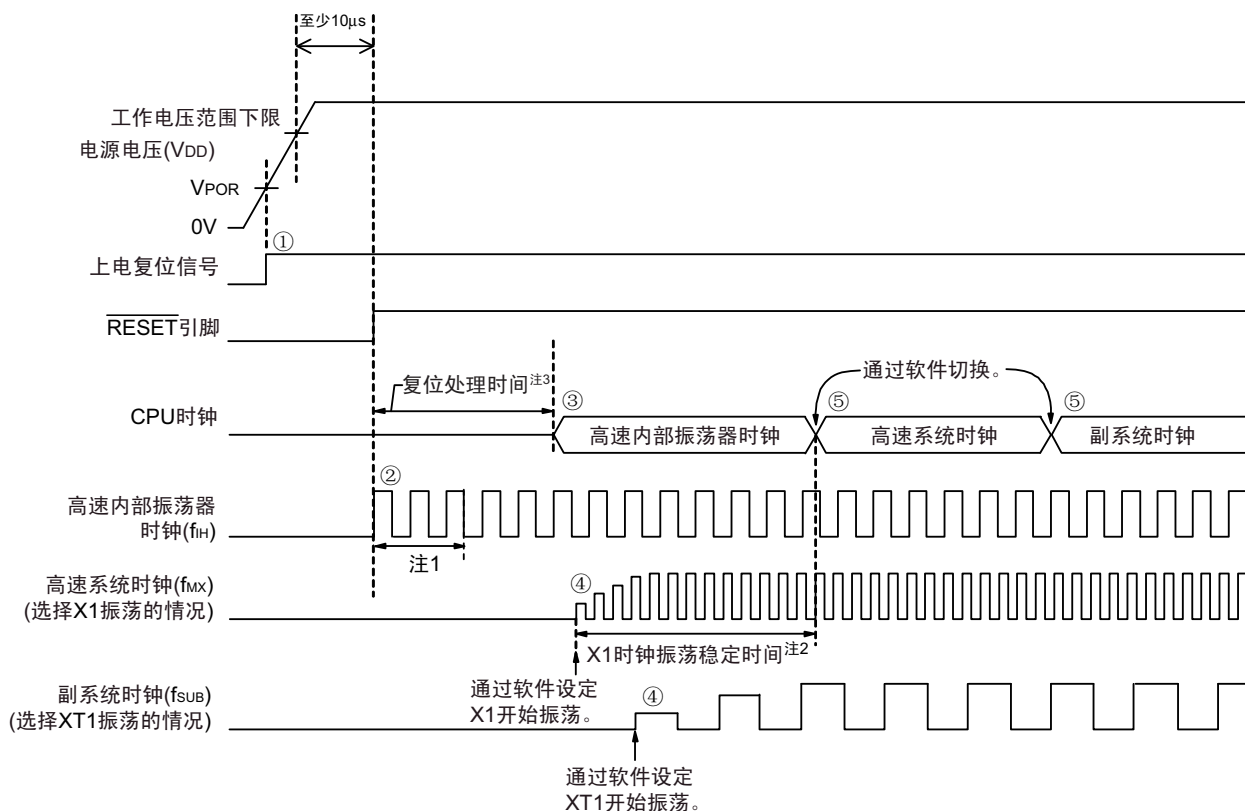
5.5 时钟发生电路的运行

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}
- 副系统时钟 f_{SUB}
 - XT1 时钟 f_{XT}
 - 外部副系统时钟 f_{EXS}
- 低速内部振荡器时钟 f_{IL}
- CPU/外围硬件时钟 f_{CLK}

R7F0C014B2D、R7F0C014L2D 在解除复位后，CPU 通过高速内部振荡器的输出开始运行。
接通电源时的时钟发生电路的运行如图 5-15 所示。

图 5-15 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“31.4 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时钟或者XT1时钟的开始振荡（参照“5.6.2 X1振荡电路的设定例子”和“5.6.3 XT1振荡电路的设定例子”）。
- ⑤ 如果要将CPU时钟切换到X1时钟或者XT1时钟，就必须在等待时钟振荡稳定后通过软件设定切换（参照“5.6.2 X1振荡电路的设定例子”和“5.6.3 XT1振荡电路的设定例子”）。

- 注
1. 复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。
 2. 当解除复位时，必须通过振荡稳定时间计数器的状态寄存器（OSTC）确认X1时钟的振荡稳定时间。
 3. 有关复位处理时间，请参照“第22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 ~ FRQSEL4 位，从 64MHz、48MHz、32MHz、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，能通过高速内部振荡器的频率选择寄存器 (HOCODIV) 更改频率。

【选项字节的设定】

地址：000C2H

选项 字节 (000C2H)	7	6	5	4	3	2	1	0
	CMODE1 0/1	CMODE0 0/1	1	FRQSEL4 0/1	FRQSEL3 0/1	FRQSEL2 0/1	FRQSEL1 0/1	FRQSEL0 0/1

CMODE1	CMODE0	闪存的运行模式设定	
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	$V_{DD}=2.4V \sim 5.5V@1MHz \sim 16MHz$ $V_{DD}=2.7V \sim 5.5V@1MHz \sim 32MHz$
上述以外		禁止设定。	

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	高速内部振荡器的频率	
					f_{HOCO}	f_{IH}
1	1	0	0	0	64MHz	32MHz
1	0	0	0	0	48MHz	24MHz
0	1	0	0	0	32MHz	32MHz
0	0	0	0	0	24MHz	24MHz
0	1	0	0	1	16MHz	16MHz
0	0	0	0	1	12MHz	12MHz
0	1	0	1	0	8MHz	8MHz
0	0	0	1	0	6MHz	6MHz
0	1	0	1	1	4MHz	4MHz
0	0	0	1	1	3MHz	3MHz
0	1	1	0	0	2MHz	2MHz
0	1	1	0	1	1MHz	1MHz
上述以外					禁止设定。	

【高速内部振荡器的频率选择寄存器（HOCODIV）的设定】

地址：F00A8H

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

HOCODIV2	HOCODIV1	HOCODIV0	高速内部振荡器时钟频率的选择			
			FRQSEL4=0		FRQSEL4=1	
			FRQSEL3=0	FRQSEL3=1	FRQSEL3=0	FRQSEL3=1
0	0	0	$f_{IH}=24\text{MHz}$	$f_{IH}=32\text{MHz}$	$f_{IH}=24\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=32\text{MHz}$ $f_{HOCO}=64\text{MHz}$
0	0	1	$f_{IH}=12\text{MHz}$	$f_{IH}=16\text{MHz}$	$f_{IH}=12\text{MHz}$ $f_{HOCO}=24\text{MHz}$	$f_{IH}=16\text{MHz}$ $f_{HOCO}=32\text{MHz}$
0	1	0	$f_{IH}=6\text{MHz}$	$f_{IH}=8\text{MHz}$	$f_{IH}=6\text{MHz}$ $f_{HOCO}=12\text{MHz}$	$f_{IH}=8\text{MHz}$ $f_{HOCO}=16\text{MHz}$
0	1	1	$f_{IH}=3\text{MHz}$	$f_{IH}=4\text{MHz}$	$f_{IH}=3\text{MHz}$ $f_{HOCO}=6\text{MHz}$	$f_{IH}=4\text{MHz}$ $f_{HOCO}=8\text{MHz}$
1	0	0	禁止设定。	$f_{IH}=2\text{MHz}$	禁止设定。	$f_{IH}=2\text{MHz}$ $f_{HOCO}=4\text{MHz}$
1	0	1	禁止设定。	$f_{IH}=1\text{MHz}$	禁止设定。	$f_{IH}=1\text{MHz}$ $f_{HOCO}=2\text{MHz}$
上述以外			禁止设定。			

5.6.2 X1 振荡电路的设定例子

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

【寄存器的设定】 必须按照①～⑤的顺序设定寄存器。

- ① 将CMC寄存器的OSCSSEL位置“1”，当 f_X 大于等于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	OSCSSEL	EXCLKS	OSCSSELS		AMPHS1	AMPHS0	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	XTSTOP						HIOSTOP
	0	1	0	0	0	0	0	0

- ④ 通过OSTC寄存器等待X1振荡电路的振荡稳定。

例）要通过10MHz谐振器至少等待102 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	CLS	CSS	MCS	MCM0				
	0	0	0	1	0	0	0	0

5.6.3 XT1 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 XT1 振荡时钟，就通过副系统时钟提供模式控制寄存器 (OSMC)、时钟运行模式控制寄存器 (CMC) 和时钟运行状态控制寄存器 (CSC) 进行振荡电路的设定和振荡开始的控制，并且通过系统时钟控制寄存器 (CKC) 将 XT1 振荡时钟设定为 f_{CLK} 。

【寄存器的设定】必须按照①～⑤的顺序设定寄存器。

- ① 在 STOP 模式或者 CPU 以副系统时钟运行的 HALT 模式中，当只要使实时时钟和 12 位间隔定时器以副系统时钟运行（超低消费电流）时，必须将 RTCLPC 位置“1”。

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

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

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 0	EXCLKS 0	OSCSELS 1	0	AMPHS1 0/1	AMPHS0 0/1	AMPH 0

AMPHS0 位和 AMPHS1 位：设定 XT1 振荡电路的振荡模式。

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

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

- ④ 必须通过软件和定时器功能等，等待副系统时钟所需的振荡稳定时间。

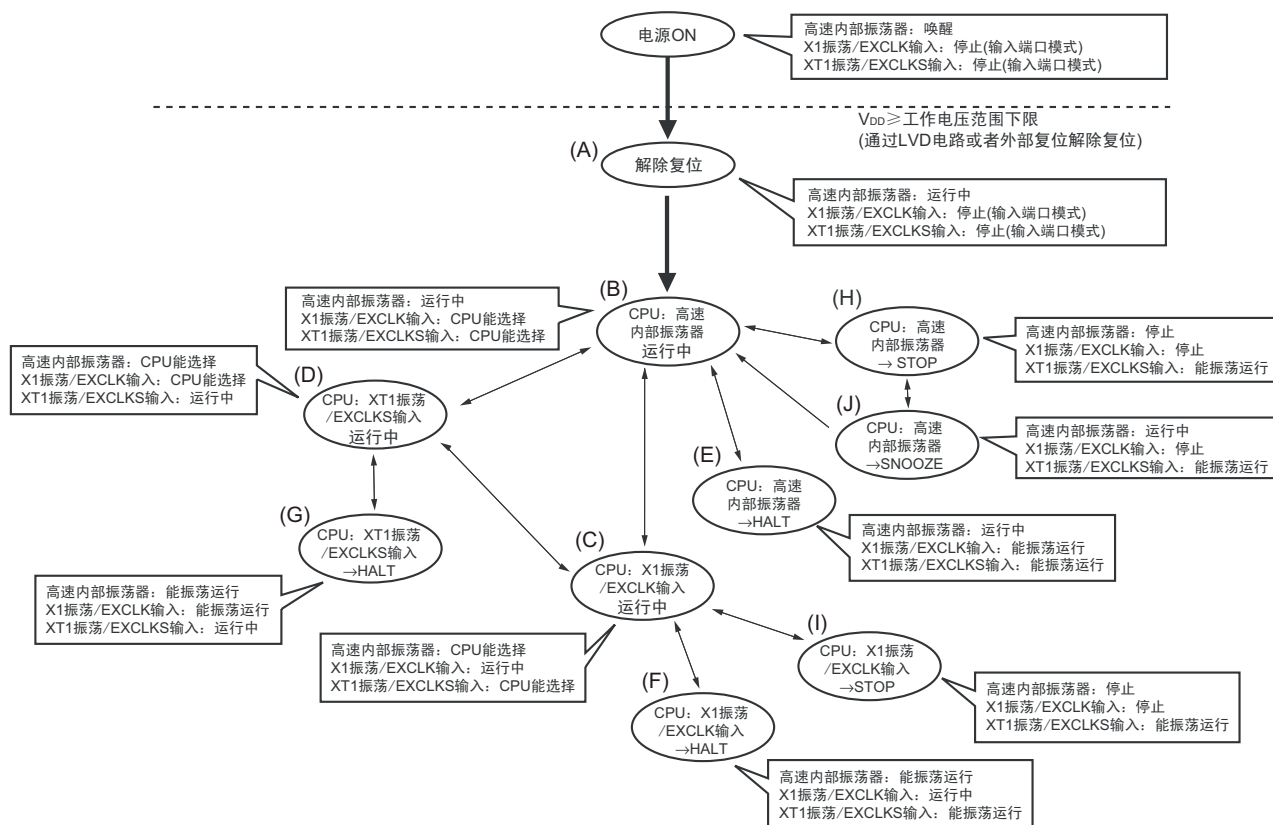
- ⑤ 通过 CKC 寄存器的 CSS 位将 XT1 振荡时钟设定为 CPU/ 外围硬件时钟。

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

5.6.4 CPU 时钟的状态转移图

本产品的CPU时钟状态转移图如图5-16所示。

图 5-16 CPU 时钟的状态转移图



CPU 时钟的转移和 SFR 寄存器的设定例子等如表 5-3 所示。

表 5-3 CPU 时钟的转移和 SFR 寄存器的设定例子 (1/5)

(1) 在解除复位(A)后, CPU 转移到高速内部振荡器时钟运行(B)。

状态转移	SFR 寄存器的设定
(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		
(A)→(B)→(C) (X1 时钟: $1\text{MHz} \leq f_X \leq 10\text{MHz}$)	0	1	0	注 2	0	需要确认	1
(A)→(B)→(C) (X1 时钟: $10\text{MHz} < f_X \leq 20\text{MHz}$)	0	1	1	注 2	0	需要确认	1
(A)→(B)→(C) (外部主时钟)	1	1	×	注 2	0	不需要 确认	1

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

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

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

注意 必须在电源电压达到设定的时钟可运行电压 (参照“第 31 章 电特性”)后设定时钟。

(3) 在解除复位(A)后, CPU 转移到副系统时钟运行(D)。

(CPU 在解除复位后立即以高速内部振荡器时钟运行(B))

(SFR 寄存器的设定顺序) →

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

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

备注 1. ×: 忽略

2. 表 5-3 的 (A) ~ (J) 对应图 5-16 的 (A) ~ (J)。

表 5-3 CPU 时钟的转移和 SFR 寄存器的设定例子 (2/5)

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

(SFR 寄存器的设定顺序) →

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	OSTC 寄存器	CKC 寄存器
	EXCLK	OSCSSEL	AMPH		MSTOP		
(B)→(C) (X1 时钟: $1\text{MHz} \leq f_X \leq 10\text{MHz}$)	0	1	0	注 2	0	需要确认	1
(B)→(C) (X1 时钟: $10\text{MHz} < f_X \leq 20\text{MHz}$)	0	1	1	注 2	0	需要确认	1
(B)→(C) (外部主时钟)	1	1	×	注 2	0	不需要确认	1

如果已设定就不需要。

在高速系统时钟运行中不需要。

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

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

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

注意 必须在电源电压达到设定的时钟可运行电压 (参照“第 31 章 电特性”)后设定时钟。

(5) CPU 从高速内部振荡器时钟运行(B)转移到副系统时钟运行(D)。

(SFR 寄存器的设定顺序) →

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

如果已设定就不需要。

在副系统时钟运行中不需要。

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

备注 1. ×: 忽略

2. 表 5-3 的 (A) ~ (J) 对应图 5-16 的 (A) ~ (J)。

表 5-3 CPU 时钟的转移和 SFR 寄存器的设定例子 (3/5)

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

(SFR 寄存器的设定顺序) →

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

在高速内部振荡器时钟运行中不需要。

注 FRQSEL4=0 时: 18μs ~ 65μs
 FRQSEL4=1 时: 18μs ~ 135μs

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。

(7) CPU 从高速系统时钟运行 (C) 转移到副系统时钟运行 (D)。

(SFR 寄存器的设定顺序) →

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

在副系统时钟运行中不需要。

(8) CPU 从副系统时钟运行 (D) 转移到高速内部振荡器时钟运行 (B)。

(SFR 寄存器的设定顺序) →

SFR 寄存器的设定标志 状态转移	CSC 寄存器	振荡稳定的等待	CKC 寄存器
	HIOSTOP		CSS
(D)→(B)	0	注	0

在高速内部振荡器时钟运行中不需要。

注 FRQSEL4=0 时: 18μs ~ 65μs
 FRQSEL4=1 时: 18μs ~ 135μs

备注 1. 表 5-3 的 (A) ~ (J) 对应图 5-16 的 (A) ~ (J)。
 2. 高速内部振荡器时钟的振荡精度稳定等待因温度条件和 STOP 模式期间而变。

表 5-3 CPU 时钟的转移和 SFR 寄存器的设定例子 (4/5)

(9) CPU 从副系统时钟运行(D)转移到高速系统时钟运行(C)。

(SFR 寄存器的设定顺序) →

状态转移 / SFR 寄存器的设定标志	OSTS 寄存器	CSC 寄存器	OSTC 寄存器	CKC 寄存器
		MSTOP		CSS
(D)→(C) (X1 时钟: $1\text{MHz} \leq f_X \leq 10\text{MHz}$)	注	0	需要确认	0
(D)→(C) (X1 时钟: $10\text{MHz} < f_X \leq 20\text{MHz}$)	注	0	需要确认	0
(D)→(C) (外部主时钟)	注	0	不需要确认	0

在高速系统时钟运行中不需要。

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

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

注意 必须在电源电压达到设定的时钟可运行电压 (参照“第 31 章 电特性”)后设定时钟。

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

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

备注 表 5-3 的 (A) ~ (J) 对应图 5-16 的 (A) ~ (J)。

表 5-3 CPU 时钟的转移和 SFR 寄存器的设定例子 (5/5)

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

(设定顺序) —————→

状态转移		设定内容		
(B)→(H)		停止不能在 STOP 模式中运行的外围功能。	—	执行 STOP 指令。
(C)→(I)	X1 振荡		设定 OSTS 寄存器。	
	外部时钟		—	

- (12) CPU从STOP模式(H)转移到SNOOZE模式(J)。
- 有关从STOP模式转移到SNOOZE模式的设定，请参照“13.8 SNOOZE 模式功能”、“14.5.7 SNOOZE 模式功能”和“14.7.3 SNOOZE 模式功能”。

备注 表 5-3 的 (A) ~ (J) 对应图 5-16 的 (A) ~ (J)。

5.6.5 CPU 时钟转移前的条件和转移后的处理

CPU 时钟转移前的条件和转移后的处理如下所示。

表 5-4 有关 CPU 时钟的转移 (1/2)

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

表 5-4 有关 CPU 时钟的转移 (2/2)

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

5.6.6 切换 CPU 时钟和系统时钟所需要的时间

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

在改写 CKC 寄存器后不立即进行实际的切换，而是在更改 CKC 寄存器后仍然以切换前的时钟继续运行数个时钟（参照表 5-5 ～表 5-7）。

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

如果切换 CPU 时钟，就同时切换外围硬件时钟。

表 5-5 切换主系统时钟所需要的最长时间

时钟 A	切换方向	时钟 B	备注
f_{IH}	↔	f_{MX}	参照表 5-6。
f_{MAIN}	↔	f_{SUB}	参照表 5-7。

表 5-6 $f_{IH} \leftrightarrow f_{MX}$ 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	

表 5-7 $f_{MAIN} \leftrightarrow f_{SUB}$ 所需要的最大时钟数

切换前的设定值		切换后的设定值	
CSS		CSS	
		0 ($f_{CLK}=f_{MAIN}$)	1 ($f_{CLK}=f_{SUB}$)
0 ($f_{CLK}=f_{MAIN}$)			1+2 个 f_{MAIN}/f_{SUB} 时钟
1 ($f_{CLK}=f_{SUB}$)		3 个时钟	

备注 1. 表 5-6 和表 5-7 中的时钟数是切换前的 CPU 时钟数。
2. 表 5-6 和表 5-7 中的时钟数是舍入小数部分的时钟数。
例 主系统时钟从高速系统时钟切换到高速内部振荡器时钟的情况（选择 $f_{IH}=8\text{MHz}$ 、 $f_{MX}=10\text{MHz}$ 振荡的情况）
 $2f_{MX}/f_{IH}=2(10/8)=2.5 \rightarrow 3$ 个时钟

5.6.7 时钟振荡停止前的条件

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

表 5-8 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 或者 CLS=1 （CPU 以高速内部振荡器时钟以外的时钟运行）	HIOSTOP=1
X1 时钟	MCS=0 或者 CLS=1 （CPU 以高速系统时钟以外的时钟运行）	MSTOP=1
外部主系统时钟		
XT1 时钟	CLS=0 （CPU 以副系统时钟以外的时钟运行）	XTSTOP=1
外部副系统时钟		

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 5-17 外接振荡电路的例子



(1) X1 振荡

截至 2014 年 3 月

厂商	谐振器	产品名称	SMD/ 引线	频率 (MHz)	闪存运行 模式注 1	振荡电路常数 (供参考) 注 2			电压范围 (V)	
						C1(pF)	C2(pF)	Rd(kΩ)	MIN.	MAX.
村田制作 所公司 注 3	陶瓷 谐振器	CSTCR4M00G55-R0	SMD	4.0	LV	(39)	(39)	0	1.6	5.5
		CSTLS4M00G53-B0	引线			(15)	(15)	0		
		CSTCR4M00G55-R0	SMD	4.0	LS	(39)	(39)	0	1.8	5.5
		CSTLS4M00G53-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		
		CSTCR5M00G53-R0	SMD	5.0	HS	(15)	(15)	0	2.4	5.5
		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		
		CSTCE10M0G52-R0	SMD	10.0		(10)	(10)	0		
		CSTLS10M0G53-B0	引线			(15)	(15)	0		
		CSTCE16M0V53-R0	SMD	16.0		(15)	(15)	0		
		CSTLS16M0X51-B0	引线			(5)	(5)	0		
		CSTCE20M0V51-R0	SMD	20.0	HS	(5)	(5)	0	2.7	5.5
		CSTLS20M0X51-B0	引线			(5)	(5)	0		
日本电波 工业公司 注 4	晶体 振荡器	NX8045GB	SMD	8	LS	注 4			1.8	5.5
		NX8045GB	SMD	8	HS				2.4	5.5
		NX5032GA	SMD	16	HS				2.7	5.5
		NX3225HA	SMD	20	HS					
RIVER ELETEC 公司	晶体 振荡器	FCX-03-8.000MHZ-J21140 注 5	SMD	8.0	HS	3	3	0	2.4	5.5
		FCX-04C-10.000MHZ-J21139注5	SMD	10.0		4	4	0		
		FCX-05-12.000MHZ-J21138 注 5	SMD	12.0		6	6	0		
		FCX-06-16.000MHZ-J21137 注 5	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. C1 和 C2 栏的 () 内表示内部电容值。

3. 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.co.jp>) 询问。

4. 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>) 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>) 询问。

备注 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 : $2.7V \leq V_{DD} \leq 5.5V @ 1MHz \sim 32MHz$

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

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) X1 振荡（晶体谐振器）

截至 2014 年 3 月

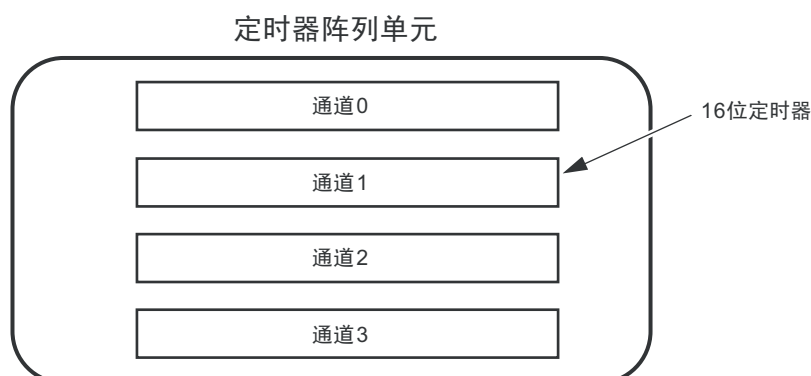
厂商	产品名称	SMD/ 引线	频率 (kHz)	负载 电容 C _L (pF)	XT1 振荡 模式注 1	振荡电路常数 (供参考)			电压范围 (V)	
						C3(pF)	C4(pF)	Rd(kΩ)	MIN.	MAX.
精工电子 公司注 2	SSP-T7-FL	SMD	32.768	6.0	通常振荡	10	9	0	1.6	5.5
				4.4	低功耗振荡	7	5	0		
				3.7	超低功耗振荡	6	3	0		
	VT-200-FL	引线		6.0	通常振荡	10	9	0		
				4.4	低功耗振荡	7	5	0		
				3.7	超低功耗振荡	6	3	0		
日本电波 工业公司注 3	NX3215SA	SMD	32.768	6.0	通常振荡	注 3			1.6	5.5
					低功耗振荡					
					超低功耗振荡					
RIVER ELETEC 公司	TFX-02-32.768KHZ- J20986 注 4	SMD	32.768	9	通常振荡	12	10	0	1.6	5.5
					低功耗振荡					
	TFX-03-32.768KHZ- J13375 注 4	SMD	32.768	9	通常振荡	12	10	0		

- 注 1. 通过时钟运行模式控制寄存器（CMC）的 AMPHS0 位和 AMPHS1 位设定 XT1 振荡模式。
2. 在使用此振荡器时，有关匹配的详细内容，请向精工电子公司（<http://www.sii-crystal.com>）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>）询问。
4. 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>）询问。

第6章 定时器阵列单元

定时器阵列单元有4个16位定时器。

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



有关各功能的详细内容，请参照下表。

独立通道运行功能	多通道联动运行功能
• 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） • 分频器注（→ 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）	• 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）

注 只限于单元0的通道0。

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

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

单元0的通道3能和串行阵列单元的UART0一起使用，实现LIN-bus通信。

6.1 定时器阵列单元的功能

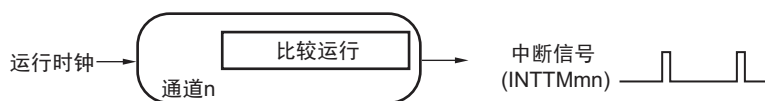
定时器阵列单元有以下功能：

6.1.1 独立通道运行功能

独立通道运行功能是能不受其他通道运行模式的影响而独立使用任意通道的功能。

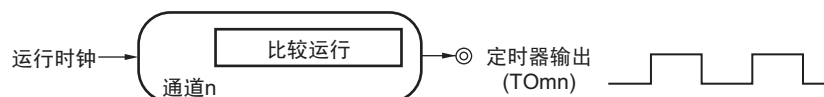
(1) 间隔定时器

能用作以固定间隔产生中断（INTTMmn）的基准定时器。



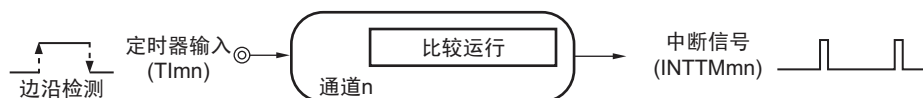
(2) 方波输出

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



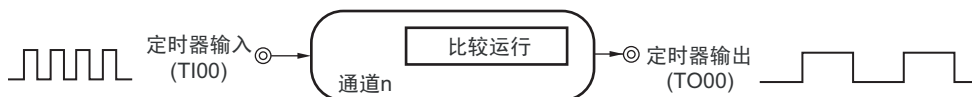
(3) 外部事件计数器

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



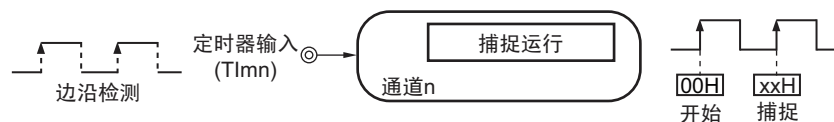
(4) 分频器功能（只限于单元 0 的通道 0）

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



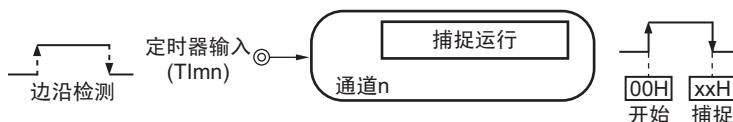
(5) 输入脉冲间隔的测量

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



(6) 输入信号的高低电平宽度的测量

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



(7) 延迟计数器

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



备注 1. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

2. 通道 0 ~ 3 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 各产品具有的定时器输入 / 输出引脚”。

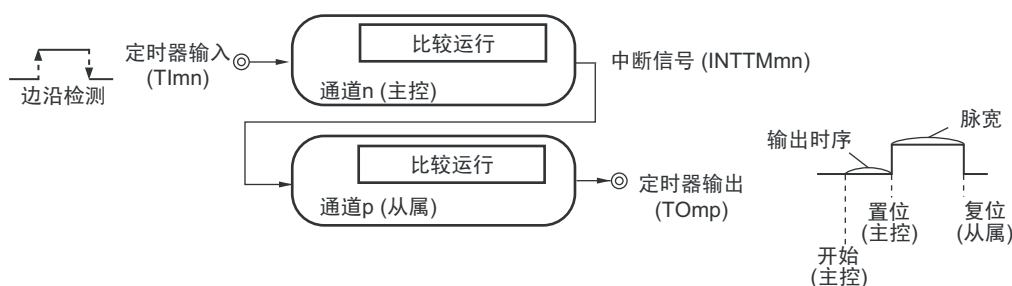
6.1.2 多通道联动运行功能

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

多通道联动运行功能能用作以下模式。

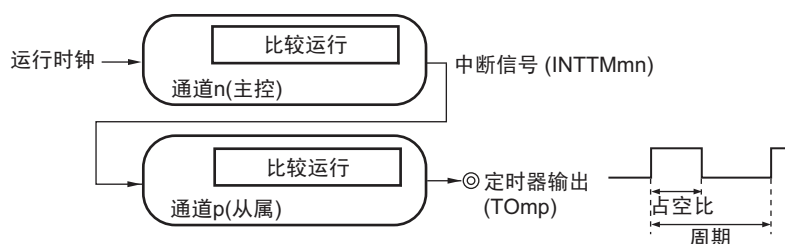
(1) 单触发脉冲输出

将 2 个通道成对使用，生成能任意设定输出时序和脉宽的单触发脉冲。



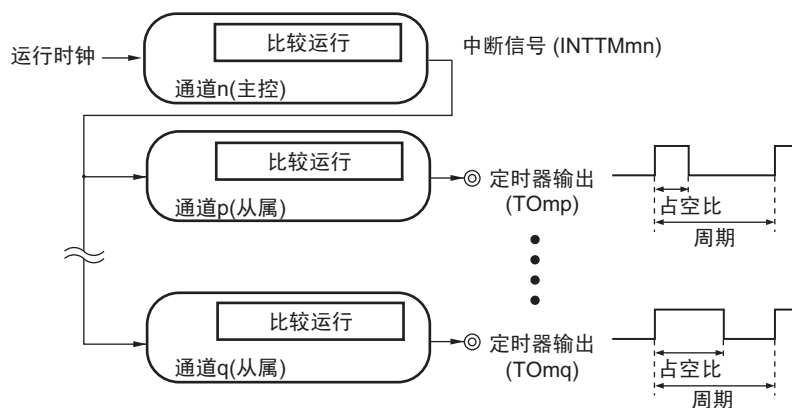
(2) PWM (Pulse Width Modulation) 输出

将 2 个通道成对使用，生成能任意设定周期和占空比的脉冲。



(3) 多重 PWM (Pulse Width Modulation) 输出

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



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

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

6.1.3 8 位定时器运行功能 (只限于通道 1 和通道 3)

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

注意 在使用 8 位定时器运行功能时，有几个规则。

详细内容请参照“6.4.2 8 位定时器运行功能的基本规则 (只限于通道 1 和通道 3)”。

6.1.4 LIN-bus 支持功能 (只限于单元 0 的通道 3)

通过定时器阵列单元检查 LIN-bus 通信中的接收信号是否适合 LIN-bus 通信格式。

(1) 唤醒信号的检测

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

(2) 间隔段的检测

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

(3) 同步段脉宽的测量

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

备注 有关 LIN-bus 支持功能的运行设定，请参照“6.3.13 输入切换控制寄存器 (ISC)”和“6.8.5 作为输入信号高低电平宽度测量的运行”。

6.2 定时器阵列单元的结构

定时器阵列单元由以下硬件构成。

表 6-1 定时器阵列单元的结构

项目	结构
计数器	定时器计数寄存器 mn (TCRmn)
寄存器	定时器数据寄存器 mn (TDRmn)
定时器的输入	TI00 ~ TI03、RxD0 引脚 (用于 LIN-bus)
定时器的输出	TO00 ~ TO03、输出控制电路
控制寄存器	<单元设定部的寄存器> • 外围允许寄存器 0 (PER0) • 定时器时钟选择寄存器 m (TPSm) • 定时器通道允许状态寄存器 m (TEm) • 定时器通道开始寄存器 m (TSM) • 定时器通道停止寄存器 m (TTm) • 定时器输入选择寄存器 0 (TIS0) • 定时器输出允许寄存器 m (TOEm) • 定时器输出寄存器 m (TOM) • 定时器输出电平寄存器 m (TOLm) • 定时器输出模式寄存器 m (TOMm)
	<每个通道的寄存器> • 定时器模式寄存器 mn (TMRmn) • 定时器状态寄存器 mn (TSRmn) • 输入切换控制寄存器 (ISC) • 噪声滤波器允许寄存器 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 使用复用功能时的端口相关寄存器的设定”。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

定时器阵列单元的各通道的定时器输入 / 输出引脚的有无因产品而不同。

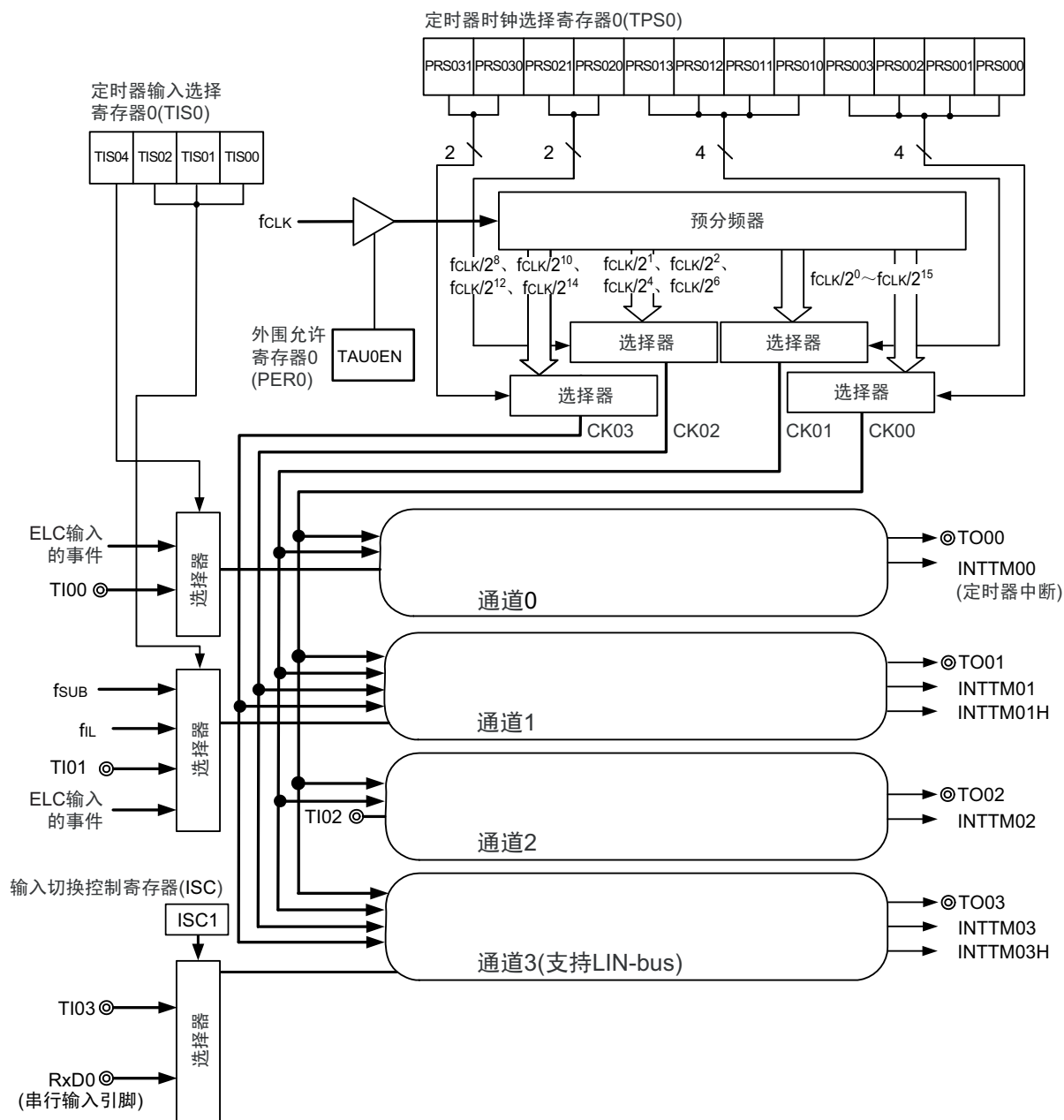
表 6-2 各产品具有的定时器输入 / 输出引脚

定时器阵列单元的通道		各产品的输入 / 输出引脚的有无
		32 引脚和 64 引脚
单元 0	通道 0	TI00、TO00
	通道 1	TI01/TO01
	通道 2	TI02/TO02
	通道 3	TI03/TO03

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

定时器阵列单元的框图如图 6-1 ~ 图 6-5 所示。

图 6-1 定时器阵列单元 0 的整体框图



备注 f_{SUB} : 副系统时钟频率
 f_{IL} : 低速内部振荡器时钟频率

图 6-2 定时器阵列单元 0 的通道 0 的内部框图

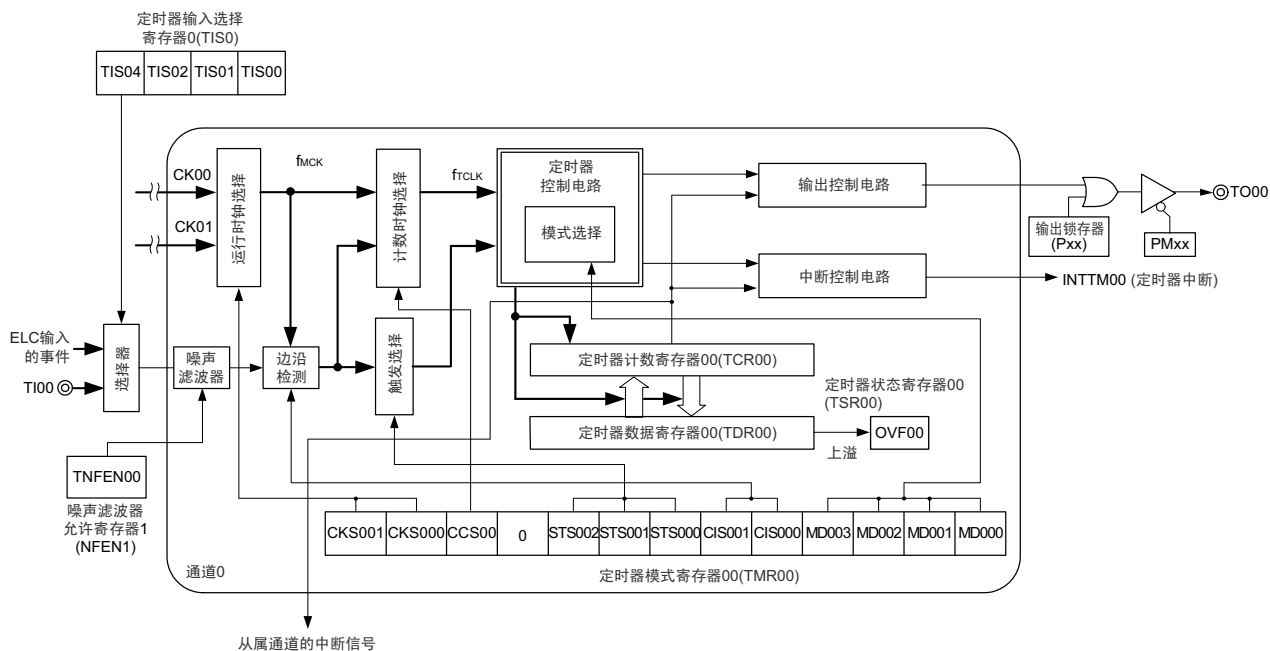


图 6-3 定时器阵列单元 0 的通道 1 的内部框图

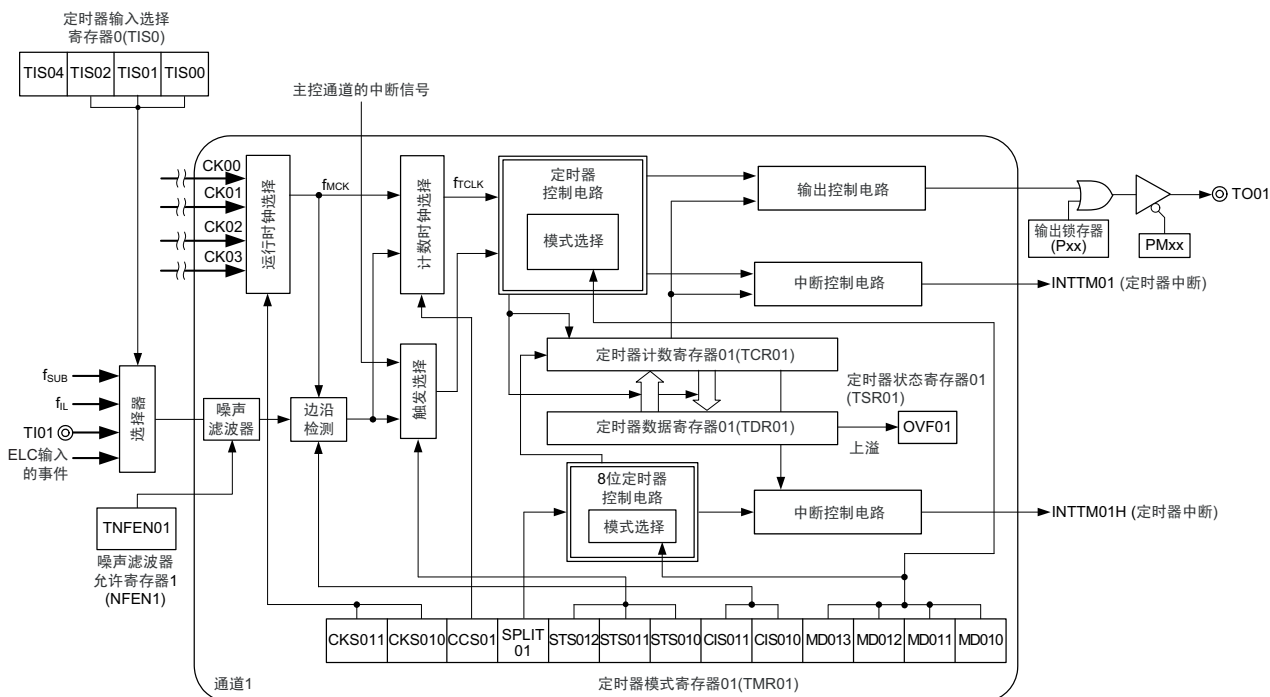


图 6-4 定时器阵列单元 0 的通道 2 的内部框图

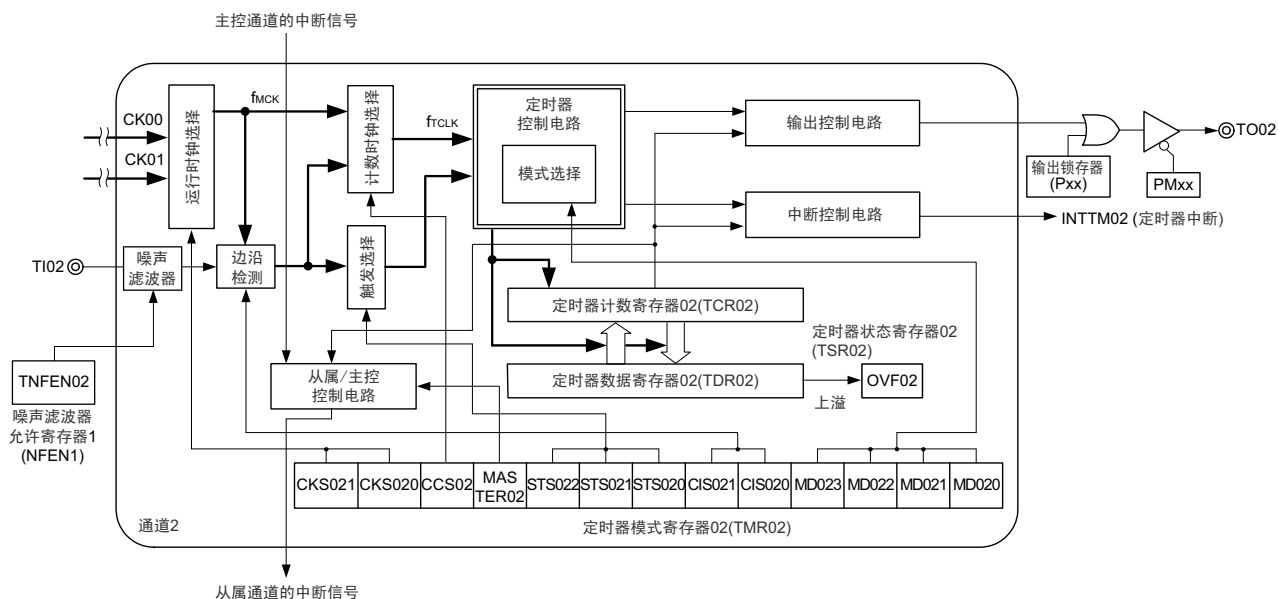
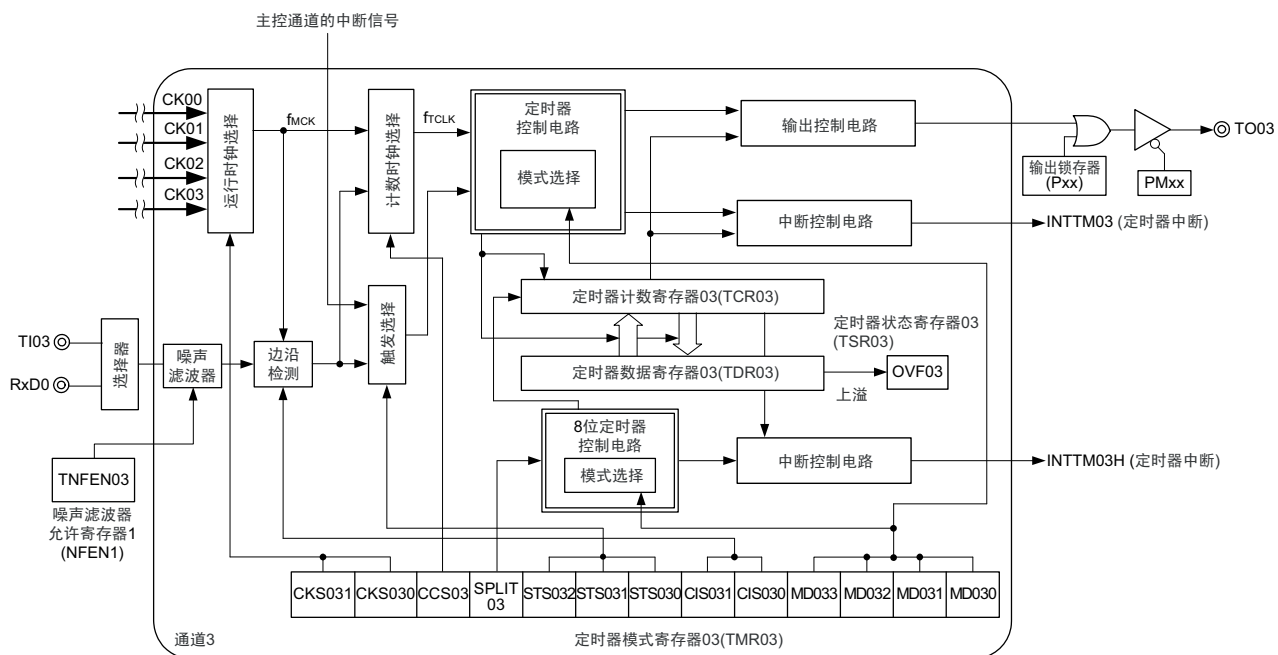


图 6-5 定时器阵列单元 0 的通道 3 的内部框图



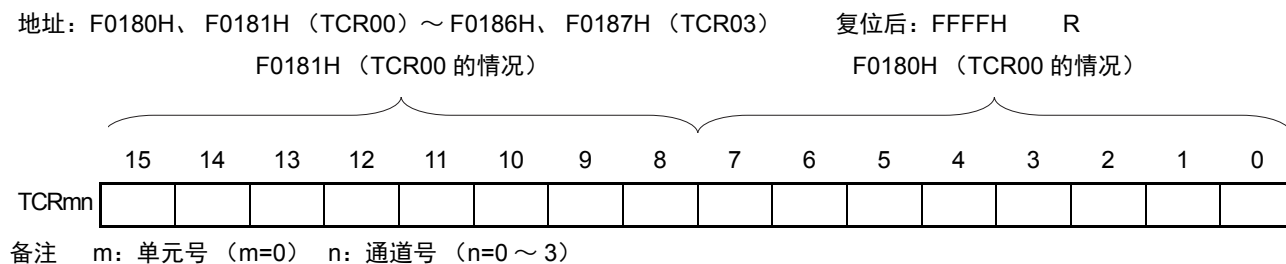
6.2.1 定时器计数寄存器 mn (TCRmn)

TCRmn 寄存器是对计数时钟进行计数的 16 位只读寄存器。

与计数时钟的上升沿同步进行递增或者递减计数。

通过定时器模式寄存器 mn (TMRmn) 的 MDmn3 ~ MDmn0 位来选择运行模式，进行递增和递减计数的切换（参照“6.3.3 定时器模式寄存器 mn (TMRmn)”）。

图 6-6 定时器计数寄存器 mn (TCRmn) 的格式



能通过读定时器计数寄存器 mn (TCRmn) 来读计数值。

在以下情况下，计数值变为“FFFFH”。

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

在以下情况下，计数值变为“0000H”。

- 在捕捉模式中输入开始触发时
- 在捕捉模式中捕捉结束时

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

如下所示，TCRmn 寄存器的读取值因运行模式和运行状态而不同。

表 6-3 各运行模式中的定时器计数寄存器 mn (TCRmn) 的读取值

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

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.2.2 定时器数据寄存器 mn (TDRmn)

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

能随时改写 TDRmn 寄存器的值。

能以 16 位为单位读写此寄存器。

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

在产生复位信号后，TDRmn 寄存器的值变为“0000H”。

图 6-7 定时器数据寄存器 mn (TDRmn) 的格式 (n=0、2)

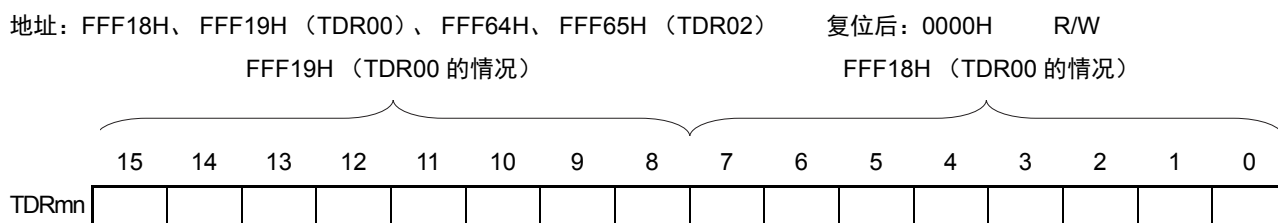
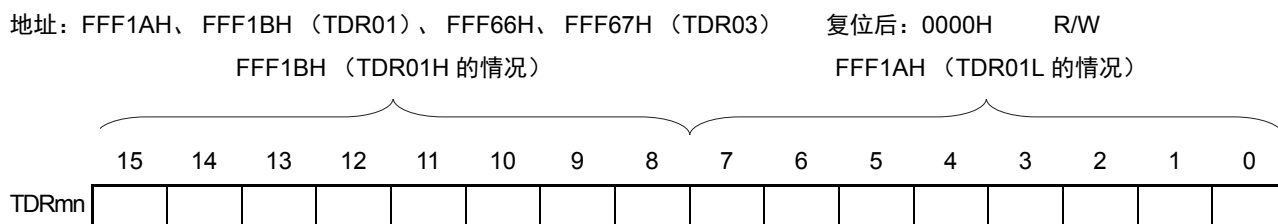


图 6-8 定时器数据寄存器 mn (TDRmn) 的格式 (n=1、3)



(i) 定时器数据寄存器 mn (TDRmn) 用作比较寄存器的情况

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

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

(ii) 定时器数据寄存器 mn (TDRmn) 用作捕捉寄存器的情况

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

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.3 控制定时器阵列单元的寄存器

控制定时器阵列单元的寄存器如下所示：

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

注意 分配的寄存器和位因产品而不同。必须给未分配的位设定初始值。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.3.1 外围允许寄存器 0 (PER0)

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

要使用定时器阵列单元 0 时，必须将 bit0 (TAU0EN) 置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER0 寄存器。

在产生复位信号后，PER0 寄存器的值变为“00H”。

图 6-9 外围允许寄存器 0 (PER0) 的格式

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

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。

注意 1. 要设定定时器阵列单元时，必须先在 TAUmEN 位为“1”的状态下设定以下的寄存器。当 TAUmEN 位为“0”时，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 (TIS0)、输入切换控制寄存器 (ISC)、噪声滤波器允许寄存器 1 (NFEN1)、端口模式控制寄存器 0、1、10、12、14 (PMC0、PMC1、PMC10、PMC12、PMC14)、端口模式寄存器 0、1、3、6 (PM0、PM1、PM3、PM6) 和端口寄存器 0、1、3、6 (P0、P1、P3、P6) 除外）。

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

2. 必须将以下的位置“0”。

bit1 和 bit6

6.3.2 定时器时钟选择寄存器 m (TPSm)

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

只有在以下情况下才能改写定时器运行中的 TPSm 寄存器。

能改写 PRSm00 ~ PRSm03 位的情况 (n=0 ~ 3):

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

能改写 PRSm10 ~ PRSm13 位的情况 (n=0 ~ 3):

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

能改写 PRSm20 位和 PRSm21 位的情况 (n=1、3):

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

能改写 PRSm30 位和 PRSm31 位的情况 (n=1、3):

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

通过 16 位存储器操作指令设定 TPSm 寄存器。

在产生复位信号后, TPSm 寄存器的值变为“0000H”。

图 6-10 定时器时钟选择寄存器 m (TPSm) 的格式 (1/2)

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

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

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

注 在更改选择为 f_{CLK} 的时钟 (更改系统时钟控制寄存器 (CKC) 的值) 的情况下, 必须停止定时器阵列单元 (TTm=000FH)。即使在选择运行时钟 (f_{MCK}) 或者 TImn 引脚输入信号的有效边沿时, 也需要停止定时器阵列单元。

注意 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, m=0~3), 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. TPSm 寄存器选择的时钟波形从上升沿开始只有 1 个 f_{CLK} 周期为高电平 (m=1~15)。详细内容请参照“6.5.1 计数时钟 (f_{TCLK})”。

图 6-10 定时器时钟选择寄存器 m (TPSm) 的格式 (2/2)

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

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

PRS m21	PRS m20	运行时钟 (CKm2) 的选择注					
			f _{CLK} =2MHz	f _{CLK} =4MHz	f _{CLK} =8MHz	f _{CLK} =20MHz	f _{CLK} =32MHz
0	0	f _{CLK} /2	1MHz	2MHz	4MHz	10MHz	16MHz
0	1	f _{CLK} /2 ²	500kHz	1MHz	2MHz	5MHz	8MHz
1	0	f _{CLK} /2 ⁴	125kHz	250kHz	500kHz	1.25MHz	2MHz
1	1	f _{CLK} /2 ⁶	31.3kHz	62.5kHz	125kHz	313kHz	500kHz

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

注 在更改选择为 f_{CLK} 的时钟 (更改系统时钟控制寄存器 (CKC) 的值) 的情况下, 必须停止定时器阵列单元 (TTm=000FH)。即使在选择运行时钟 (f_{MCK}) 或者 TImn 引脚输入信号的有效边沿时, 也需要停止定时器阵列单元。

注意 必须将 bit15、14、11、10 置“0”。

如果在 8 位定时器模式中使用通道 1 和通道 3 并且将 CKm2 和 CKm3 作为运行时钟, 就能通过间隔定时器功能实现表 6-4 所示的间隔时间。

表 6-4 运行时钟 CKSm2 和 CKSm3 能设定的间隔时间

时钟		间隔时间注 (f _{CLK} =32MHz)			
		10μs	100μs	1ms	10ms
CKm2	f _{CLK} /2	○	—	—	—
	f _{CLK} /2 ²	○	—	—	—
	f _{CLK} /2 ⁴	○	○	—	—
	f _{CLK} /2 ⁶	○	○	—	—
CKm3	f _{CLK} /2 ⁸	—	○	○	—
	f _{CLK} /2 ¹⁰	—	○	○	—
	f _{CLK} /2 ¹²	—	—	○	○
	f _{CLK} /2 ¹⁴	—	—	○	○

注 ○包含 5% 以内的误差。

备注 1. f_{CLK}: CPU/ 外围硬件的时钟频率

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

6.3.3 定时器模式寄存器 mn (TMRmn)

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

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

通过 16 位存储器操作指令设定 TMRmn 寄存器。

在产生复位信号后，TMRmn 寄存器的值变为“0000H”。

注意 TMRmn 寄存器的 bit11 因通道而不同。

TMRm2	: MASTERmn 位 (n=2)
TMRm1、TMRm3	: SPLITmn 位 (n=1、3)
TMRm0	: 固定为“0”。

图 6-11 定时器模式寄存器 mn (TMRmn) 的格式 (1/4)

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

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

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

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

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

CCSmn	通道 n 计数时钟 (f_{TCLK}) 的选择
0	CKSmn0 位和 CKSmn1 位指定的运行时钟 (f_{MCK})
1	TImn 引脚输入信号的有效边沿
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 1. bit11 是只读位, 固定为“0”, 忽视写操作。

注意 1. 必须将 bit13、5、4 置“0”。

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-11 定时器模式寄存器 mn (TMRmn) 的格式 (2/4)

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

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

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

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

(TMRmn (n=2) 的 bit11)

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

(TMRmn (n=1、3) 的 bit11)

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

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

注 1. bit11 是只读位, 固定为“0”, 忽视写操作。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-11 定时器模式寄存器 mn (TMRmn) 的格式 (3/4)

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

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

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

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

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

注 1. bit11 是只读位, 固定为“0”, 忽视写操作。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-11 定时器模式寄存器 mn (TMRmn) 的格式 (4/4)

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

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

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

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

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

运行模式 (MDmn3 ~ MDmn1 位的设定 (参照上表))	MD mn0	开始计数和中断的设定
• 间隔定时器模式 (0、0、0) • 捕捉模式 (0、1、0)	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。
	1	在开始计数时产生定时器中断 (定时器的输出也发生变化)。
• 事件计数器模式 (0、1、1)	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。
• 单次计数模式 ^{注2} (1、0、0)	0	计数运行中的开始触发无效。此时不产生中断。
	1	计数运行中的开始触发有效 ^{注3} 。此时不产生中断。
• 捕捉 & 单次计数模式 (1、1、0)	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。计数运行中的开始触发无效。此时不产生中断。

注 1. bit11 是只读位, 固定为“0”, 忽视写操作。

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

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.3.4 定时器状态寄存器 mn (TSRmn)

TSRmn 寄存器是表示通道 n 计数器的上溢状态的寄存器。

TSRmn 寄存器只在捕捉模式 (MDmn3~MDmn1=010B) 和捕捉 & 单次计数模式 (MDmn3~MDmn1=110B) 中有效。有关各运行模式中的 OVF 位的变化和置位 / 清除条件, 请参照表 6-5。

通过 16 位存储器操作指令读 TSRmn 寄存器。

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

在产生复位信号后, TSRmn 寄存器的值变为“0000H”。

图 6-12 定时器状态寄存器 mn (TSRmn) 的格式

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

复位后: 0000H

R

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

OVF	通道 n 的计数器上溢状态
0	没有发生上溢。
1	发生上溢。

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

表 6-5 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
• 捕捉模式	清除	在捕捉时没有发生上溢的情况
• 捕捉 & 单次计数模式	置位	在捕捉时发生上溢的情况
• 间隔定时器模式	清除	— (不能使用)
• 事件计数器模式	置位	
• 单次计数模式		

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

6.3.5 定时器通道允许状态寄存器 m（TEm）

TEm 寄存器是表示各通道定时器运行的允许或者停止状态的寄存器。

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

通过 16 位存储器操作指令读 TEm 寄存器。

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

在产生复位信号后，TEm 寄存器的值变为“0000H”。

图 6-13 定时器通道允许状态寄存器 m（TEm）的格式

地址：F01B0H、F01B1H（TE0）

复位后：0000H

R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEm	0	0	0	0	TEH m3	0	TEH m1	0	0	0	0	0	TEm 3	TEm 2	TEm 1	TEm 0

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

TEH m1	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														

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

在通道 1 和通道 3 为 8 位定时器模式时，TEm1 和 TEm3 表示低 8 位定时器的运行允许或者停止状态。

备注 m: 单元号（m=0） n: 通道号（n=0～3）

6.3.6 定时器通道开始寄存器 m (TSm)

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

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

通过 16 位存储器操作指令设定 TSm 寄存器。

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

在产生复位信号后，TSm 寄存器的值变为“0000H”。

图 6-14 定时器通道开始寄存器 m (TSm) 的格式

地址: F01B2H、F01B3H (TS0)

复位后: 0000H

R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSm	0	0	0	0	TSH m3	0	TSH m1	0	0	0	0	0	TSm 3	TSm 2	TSm 1	TSm 0

TSH m3	通道 3 为 8 位定时器模式时的高 8 位定时器的运行允许（开始）触发
0	没有触发。
1	将 TEHm3 位置“1”，进入计数允许状态。 如果在计数允许状态下开始 TCRm3 寄存器的计数，就进入间隔定时器模式（参照“6.5.2 计数器的开始时序”的表 6-6）。

TSH m1	通道 1 为 8 位定时器模式时的高 8 位定时器的运行允许（开始）触发
0	没有触发。
1	将 TEHm1 位置“1”，进入计数允许状态。 如果在计数允许状态下开始 TCRm1 寄存器的计数，就进入间隔定时器模式（参照“6.5.2 计数器的开始时序”的表 6-6）。

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

- 注意 1. 必须将 bit15 ~ 12、10、8 ~ 4 置“0”。
2. 在从不使用 TImn 引脚输入的功能切换到使用 TImn 引脚输入的功能时，从设定定时器模式寄存器 mn (TMRmn) 到将 TSmn (TSHm1、TSHm3) 位置“1”为止，需要以下期间的等待：
TImn 引脚噪声滤波器有效时 (TNFENmn=1)：4 个运行时钟 (f_{MCK})
TImn 引脚噪声滤波器无效时 (TNFENmn=0)：2 个运行时钟 (f_{MCK})

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

6.3.7 定时器通道停止寄存器 m (TTm)

TTm 寄存器是设定各通道计数停止的触发寄存器。

如果将各位置“1”，定时器通道允许状态寄存器 m (TEm) 的对应位就被清“0”。因为 TTmn 位、TTHm1 位和 TTHm3 位是触发位，所以如果变为运行停止状态 (TEmn、TEHm1、TEHm3=0)，就立即清除 TTmn 位、TTHm1 位和 TTHm3 位。

通过 16 位存储器操作指令设定 TTm 寄存器。

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

在产生复位信号后，TTm 寄存器的值变为“0000H”。

图 6-15 定时器通道停止寄存器 m (TTm) 的格式

地址: F01B4H、F01B5H (TT0)

复位后: 0000H

R/W

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

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

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

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

注意 必须将 bit15 ~ 12、10、8 ~ 4 置“0”。

备注 1. TTm 寄存器的读取值总是“0”。

2. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.3.8 定时器输入选择寄存器 0 (TIS0)

TIS0 寄存器是选择单元 0 的通道 0 和通道 1 定时器输入的寄存器。

通过 8 位存储器操作指令设定 TIS0 寄存器。

在产生复位信号后，TIS0 寄存器的值变为“00H”。

图 6-16 定时器输入选择寄存器 0 (TIS0) 的格式

地址: F0074H	复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0
TIS0	0	0	0	TIS04	0	TIS02	TIS01	TIS00

TIS04	通道 0 使用的定时器输入的选择
0	定时器输入引脚 (TI00) 的输入信号
1	ELC 的事件输入信号

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

注意 1. 选择的定时器输入的高低电平宽度必须大于等于 $1/f_{MCK}+10ns$ 。

因此，在选择 f_{SUB} 作为 f_{CLK} 时 (CKS 寄存器的 CSS=1)，不能将 TIS02 位置“1”。

- 在通过定时器输入选择寄存器 0 (TIS0) 选择 ELC 的事件输入信号时，必须通过定时器时钟选择寄存器 0 (TPS0) 选择 f_{CLK} 。

6.3.9 定时器输出允许寄存器 m（TOEm）

TOEm 寄存器是设定允许或者禁止各通道定时器输出的寄存器。

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

通过 16 位存储器操作指令设定 TOEm 寄存器。

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

在产生复位信号后，TOEm 寄存器的值变为“0000H”。

图 6-17 定时器输出允许寄存器 m（TOEm）的格式

地址: F01BAH、F01BBH (TOE0)

复位后: 0000H

R/W

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

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

注意 必须将 bit15 ~ 4 置“0”。

备注 m: 单元号（m=0） n: 通道号（n=0 ~ 3）

6.3.10 定时器输出寄存器 m (TOM)

TOM 寄存器是各通道定时器输出的缓冲寄存器。

此寄存器各位的值从各通道定时器的输出引脚 (TOMn) 输出。

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

要将 TI00、TO00、TI01/TO01、TI02/TO02、TI03/TO03 引脚用作端口功能时，必须将相应的 TOMn 位置“0”。

通过 16 位存储器操作指令设定 TOM 寄存器。

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

在产生复位信号后，TOM 寄存器的值变为“0000H”。

图 6-18 定时器输出寄存器 m (TOM) 的格式

地址: F01B8H、F01B9H (TO0)

复位后: 0000H

R/W

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

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

注意 必须将 bit15 ~ 4 置“0”。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.3.11 定时器输出电平寄存器 m (TOLm)

TOLm 寄存器是控制各通道定时器输出电平的寄存器。

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

通过 16 位存储器操作指令设定 TOLm 寄存器。

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

在产生复位信号后, TOLm 寄存器的值变为“0000H”。

图 6-19 定时器输出电平寄存器 m (TOLm) 的格式

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

注意 必须将 bit15 ~ 4 和 bit0 置“0”。

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

2. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.3.12 定时器输出模式寄存器 m（TOMm）

TOMm 寄存器是控制各通道定时器输出模式的寄存器。

当用作独立通道运行功能时，将所用通道的对应位置“0”。

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

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

通过 16 位存储器操作指令设定 TOMm 寄存器。

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

在产生复位信号后，TOMm 寄存器的值变为“0000H”。

图 6-20 定时器输出模式寄存器 m（TOMm）的格式

地址：F01BEH、F01BFH（TOM0）

复位后：0000H

R/W

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

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

注意 必须将 bit15～4 和 bit0 置“0”。

备注 m: 单元号（m=0）
 n: 通道号
 n=0～3（主控通道时：n=0、2）
 p: 从属通道号
 n=0: p=1、2、3
 n=2: p=3
 （有关主控通道和从属通道关系的详细内容，请参照“6.4.1 多通道联动运行功能的基本规则”）

6.3.13 输入切换控制寄存器（ISC）

ISC 寄存器的 ISC1 位和 ISC0 位用于将通道 3 和串行阵列单元联动来实现 LIN-bus 通信运行。如果将 ISC1 位置“1”，就选择串行数据输入引脚（RxD0）的输入信号作为定时器的输入。

有关 SSIE00 位的设定，请参照“14.3.15 输入切换控制寄存器（ISC）”。

通过 1 位或者 8 位存储器操作指令设定 ISC 寄存器。

在产生复位信号后，ISC 寄存器的值变为“00H”。

图 6-21 输入切换控制寄存器（ISC）的格式

地址: F0073H

复位后: 00H

R/W

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

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

ISC1	定时器阵列单元 0 的通道 3 的输入切换
0	将 TI03 引脚的输入信号作为定时器的输入（通常运行）。
1	将 RxD0 引脚的输入信号作为定时器的输入（唤醒信号检测、间隔段低电平宽度测量和同步段脉宽测量）。

ISC0	外部中断（INTP0）的输入切换
0	将 INTP0 引脚的输入信号作为外部中断的输入（通常运行）。
1	将 RxD0 引脚的输入信号作为外部中断的输入（唤醒信号检测）。

注意 必须将 bit6 ~ 2 置“0”。

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

6.3.14 噪声滤波器允许寄存器 1（NFEN1）

NFEN1 寄存器设定噪声滤波器是否用于各通道定时器输入引脚的输入信号。

对于需要消除噪声的引脚，必须将对应的位置“1”，使噪声滤波器有效。

当噪声滤波器有效时，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 TImn 引脚输入信号的有效边沿的情况（CCSmn=1）”、“6.5.2 计数器的开始时序”和“6.7 定时器输入（TImn）的控制”。

图 6-22 噪声滤波器允许寄存器 1（NFEN1）的格式

地址: F0071H

复位后: 00H

R/W

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

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

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

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

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

注 能通过设定输入切换控制寄存器（ISC）的 ISC1 位来切换适用的引脚。

ISC1=0: 可选择是否使用 TI03 引脚的噪声滤波器。

ISC1=1: 可选择是否使用 RxD0 引脚的噪声滤波器。

备注 通道 0 ~ 3 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 各产品具有的定时器输入 / 输出引脚”。

6.3.15 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 0、12、14（PMC0、PMC12、PMC14）”。

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 使用复用功能时的端口相关寄存器的设定”。

在将定时器输出引脚的复用端口（P00/TI00、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”并且将端口模式控制寄存器（PMCxx）的位置“0”。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。

（例）将 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。

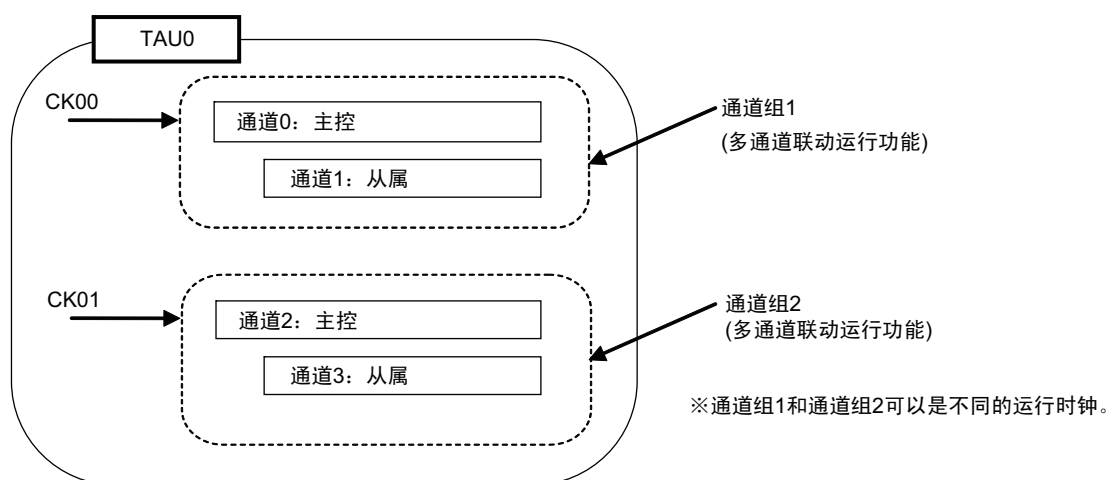
- (1) 只能将偶数通道（通道0、通道2）设定为主控通道。
- (2) 通道0以外的任何通道都能设定为从属通道。
- (3) 只能将主控通道的低位通道设定为从属通道。
例 在将通道0设定为主控通道时，能将通道1开始的通道（通道1、通道2、通道3）设定为从属通道。
- (4) 能对1个主控通道设定多个从属通道。
- (5) 当使用多个主控通道时，不能设定跨越主控通道的从属通道。
例 在将通道0和通道2设定为主控通道时，能将通道1设定为主控通道0的从属通道，而不能将通道3设定为主控通道0的从属通道。
- (6) 和主控通道联动的从属通道需要设定相同的运行时钟。和主控通道联动的从属通道的CKSmn0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

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

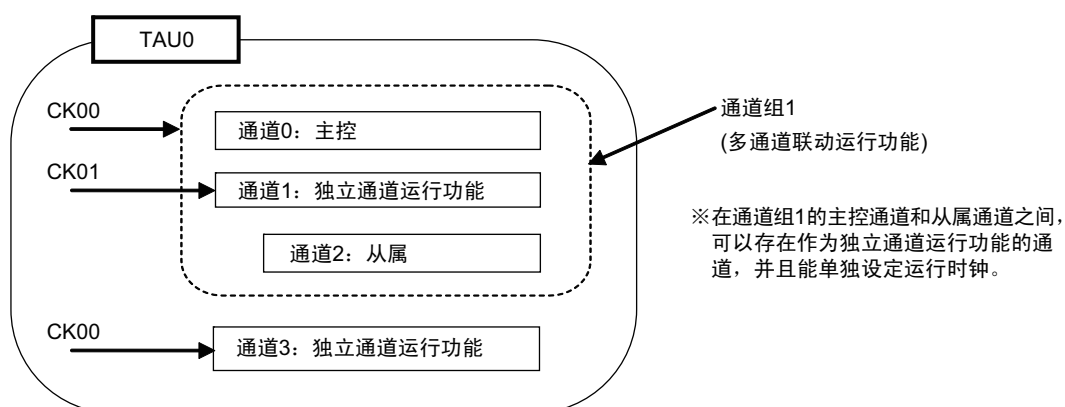
如果设定2个或者更多的相互不联动的通道群，通道群之间就不适用上述的基本规则。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

例 1



例 2



6.4.2 8位定时器运行功能的基本规则（只限于通道1和通道3）

8位定时器运行功能是将16位定时器的通道用作2个8位定时器的通道的功能。

只有通道1和通道3才能使用8位定时器运行功能，使用时需要遵守几个规则。

8位定时器运行功能的基本规则如下所示。

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

备注 m: 单元号 (m=0) n: 通道号 (n=1、3)

6.5 计数器的运行

6.5.1 计数时钟 (f_{TCLK})

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

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

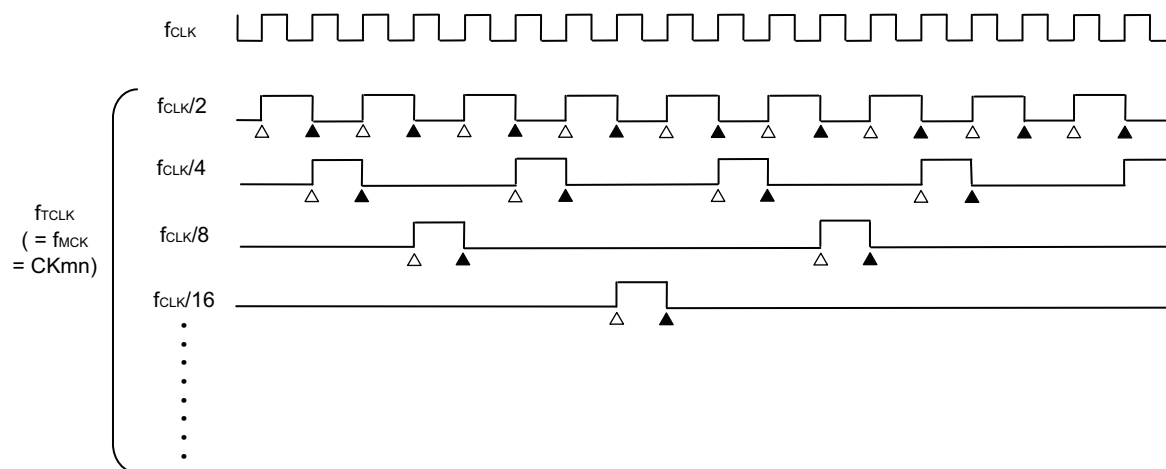
定时器阵列单元被设计为与 f_{CLK} 同步运行, 因此计数时钟 (f_{TCLK}) 的时序如下。

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

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

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

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

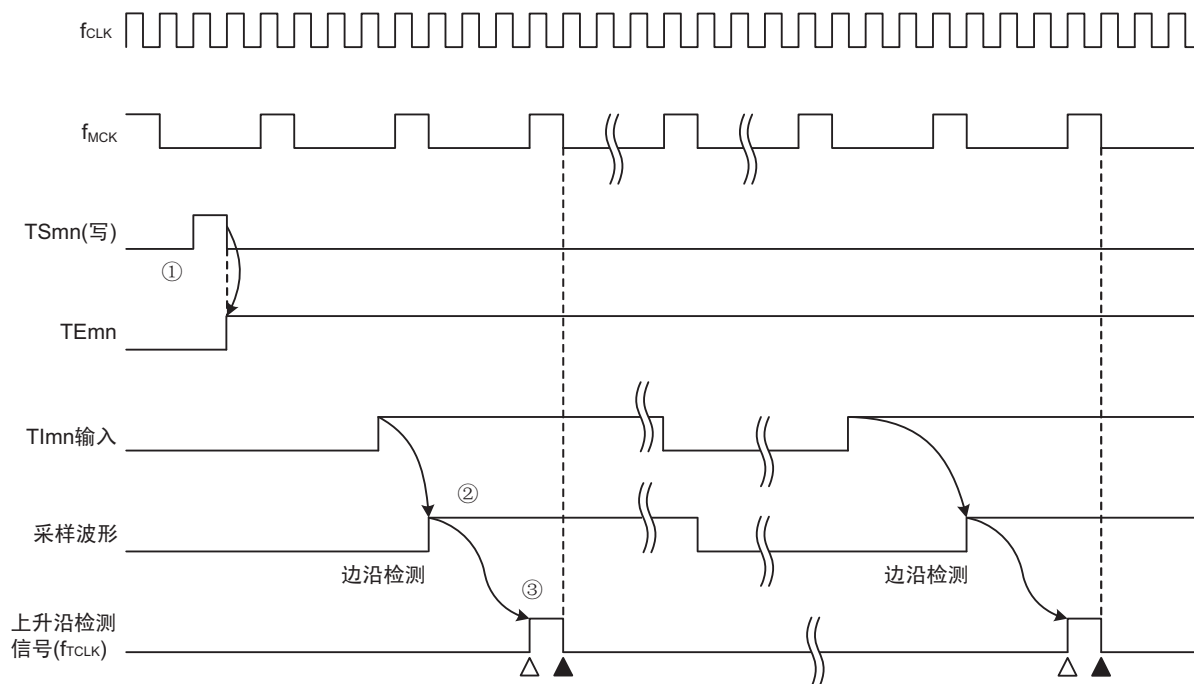


- 备注 1. Δ : 计数时钟的上升沿
 $\blacktriangle$: 同步、计数器的递增 / 递减
2. f_{CLK} : CPU/ 外围硬件的时钟

(2) 选择 TImn 引脚输入信号的有效边沿的情况 (CCSmn=1)

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

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

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

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

备注 1. 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 TImn 输入也是同样的波形。

6.5.2 计数器的开始时序

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

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

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

定时器的运行模式	将 TSmn 位置“1”后的运行
• 间隔定时器模式	从检测到开始触发 (TSmn=1) 到产生计数时钟为止, 不执行任何操作。 通过第 1 个计数时钟将 TDRmn 寄存器的值装入 TCRmn 寄存器, 并且通过后续的计数时钟进行递减计数 (参照“6.5.3 (1) 间隔定时器模式的运行”)。
• 事件计数器模式	通过给 TSmn 位写“1”, 将 TDRmn 寄存器的值装入 TCRmn 寄存器。 如果检测到 TImn 的输入边沿, 就通过后续的计数时钟进行递减计数。(参照“6.5.3 (2) 事件计数器模式的运行”)。
• 捕捉模式	从检测到开始触发到产生计数时钟为止, 不执行任何操作。 通过第 1 个计数时钟将“0000H”装入 TCRmn 寄存器, 并且通过后续的计数时钟进行递增计数 (参照“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 (TEmn=0) 的状态下给 TSmn 位写“1”, 进入开始触发的等待状态。 从检测到开始触发到产生计数时钟为止, 不执行任何操作。 通过第 1 个计数时钟将 TDRmn 寄存器的值装入 TCRmn 寄存器, 并且通过后续的计数时钟进行递减计数 (参照“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 (TEmn=0) 的状态下给 TSmn 位写“1”, 进入开始触发的等待状态。 从检测到开始触发到产生计数时钟为止, 不执行任何操作。 通过第 1 个计数时钟将“0000H”装入 TCRmn 寄存器, 并且通过后续的计数时钟进行递增计数 (参照“6.5.3 (5) 捕捉 & 单次计数模式的运行 (高电平的宽度测量)”)。

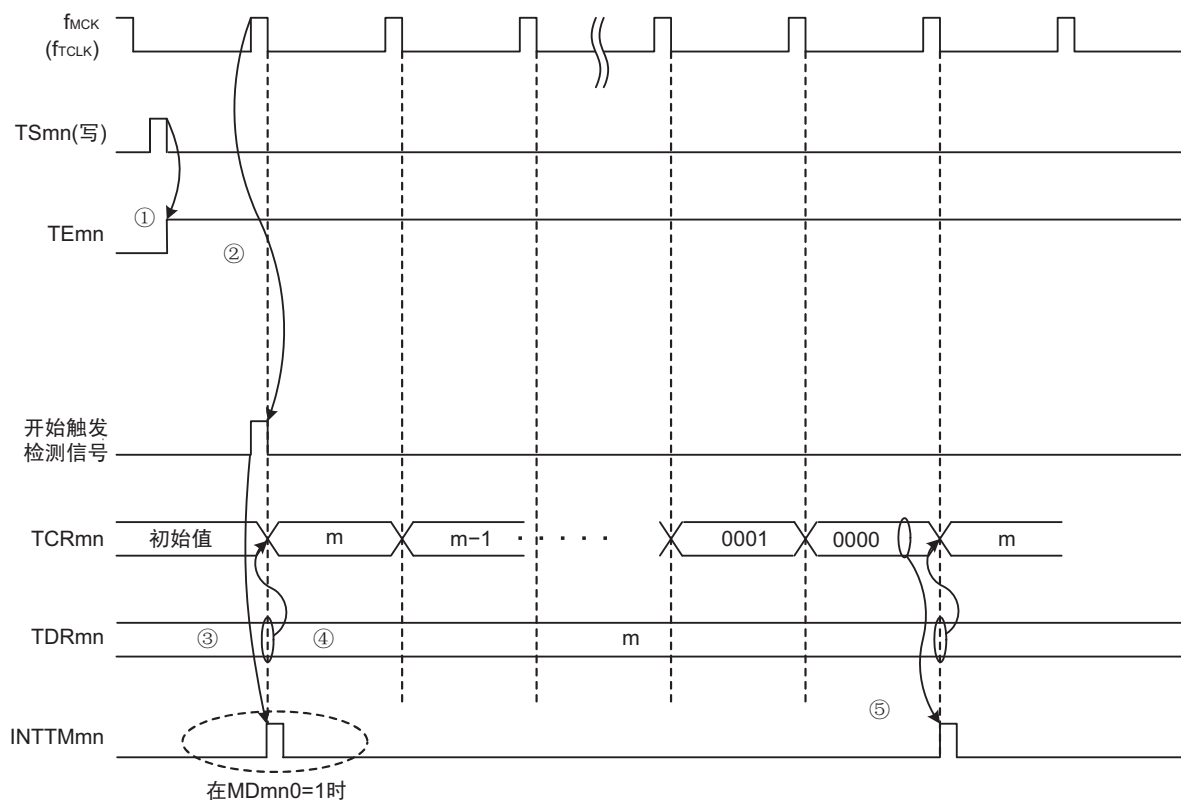
6.5.3 计数器的运行

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

图 6-25 运行时序（间隔定时器模式）



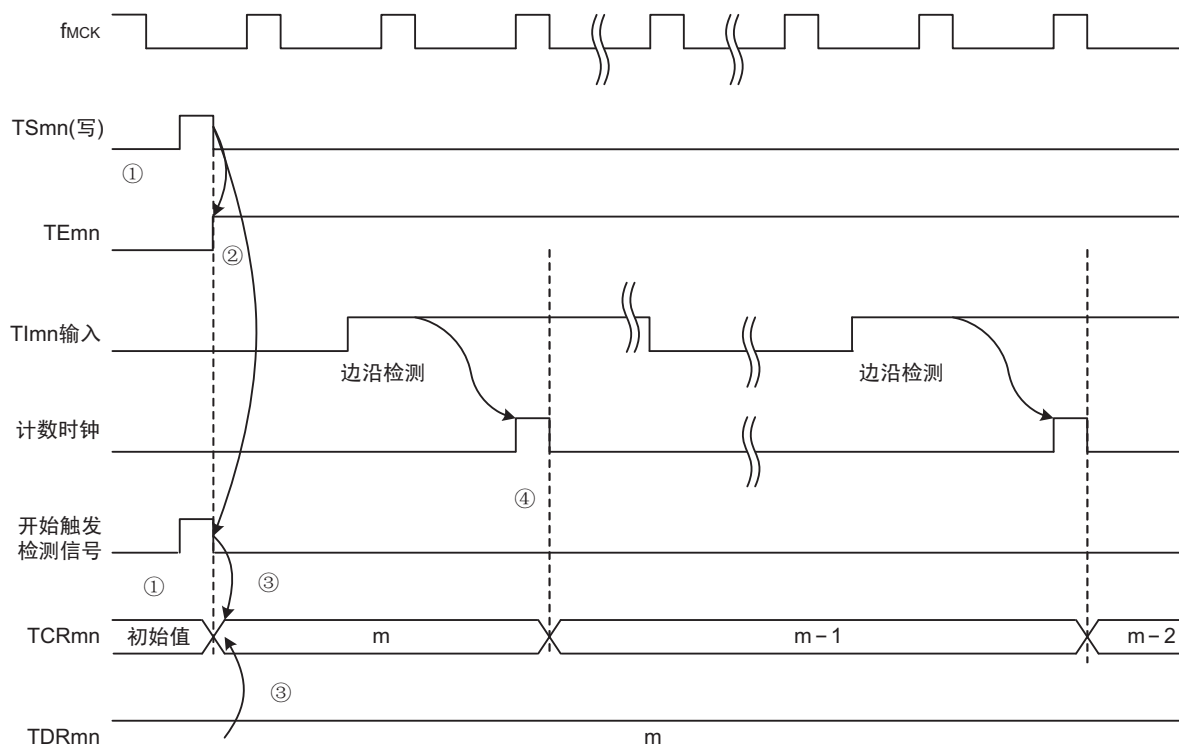
注意 因为第 1 个计数时钟周期的运行在写 TSmn 位后并且在产生计数时钟前延迟计数的开始，所以产生最大为 1 个时钟周期的误差。另外，如果需要开始计数时序的信息，就将 MDmn0 位置“1”，以便能在开始计数时产生中断。

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

(2) 事件计数器模式的运行

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

图 6-26 运行时序 (事件计数器模式)

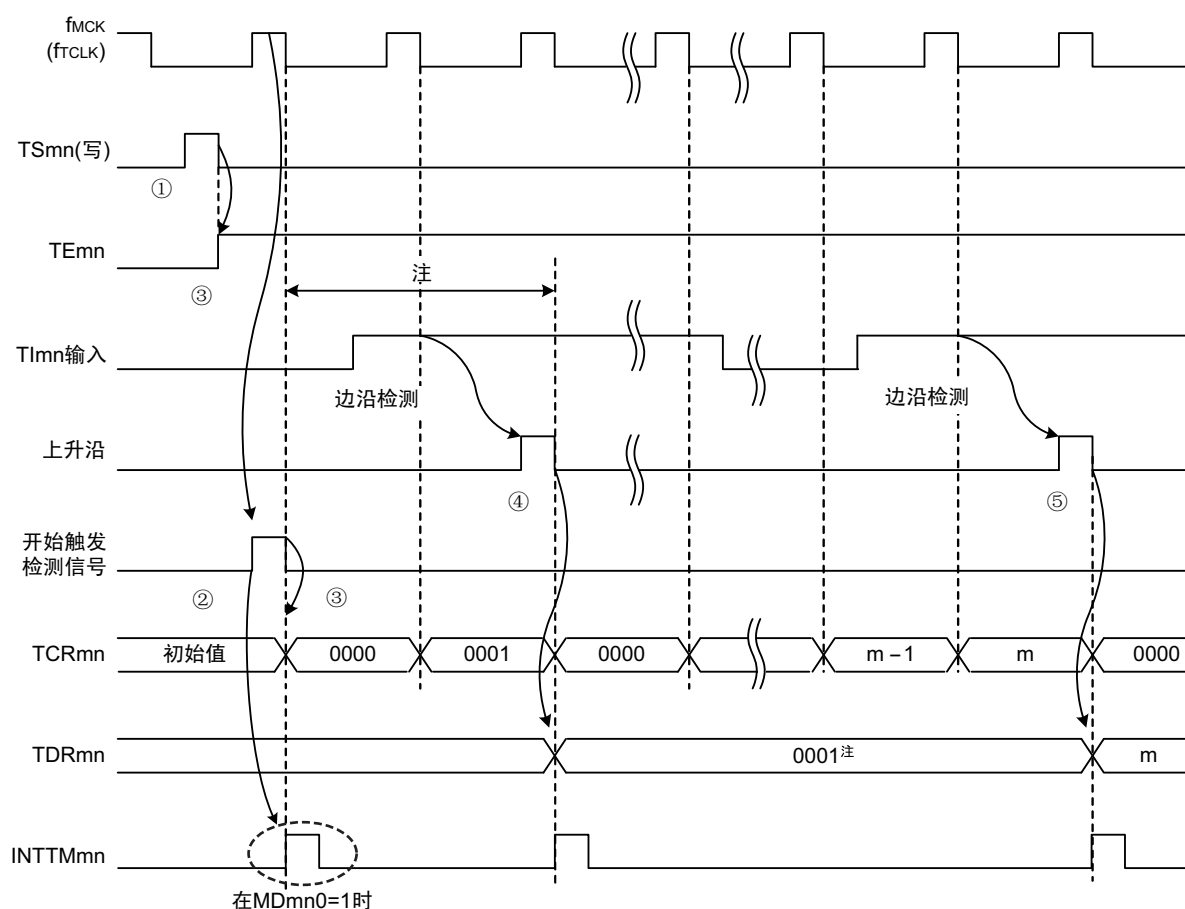


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器, 边沿检测就从 TI_{mn} 输入开始再延迟 2 个 f_{MCK} 周期 (合计 3 ~ 4 个周期)。1 个周期的误差是因为 TI_{mn} 输入与计数时钟 (f_{MCK}) 不同步。

(3) 捕捉模式的运行（输入脉冲的间隔测量）

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

图 6-27 运行时序（捕捉模式：输入脉冲的间隔测量）



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

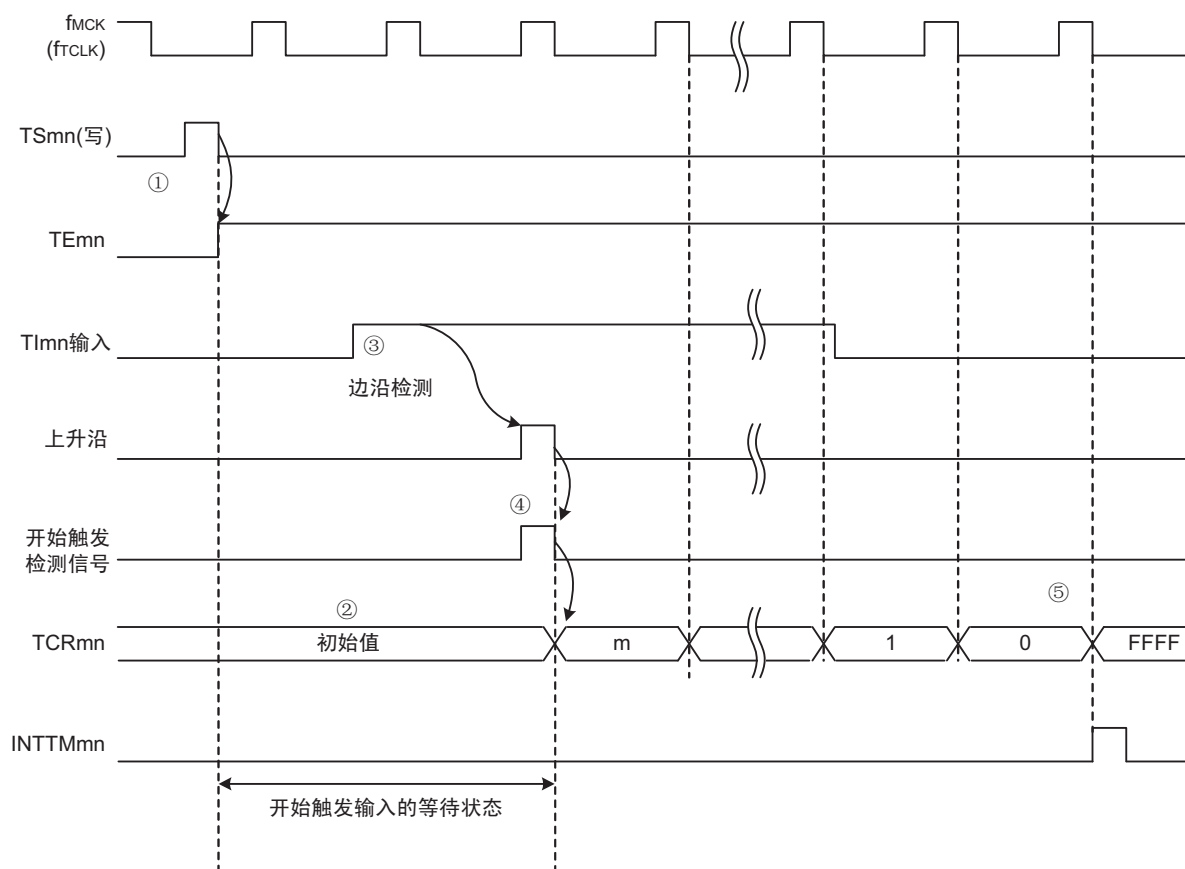
注意 因为第 1 个计数时钟周期的运行在写 TSmn 位后并且在产生计数时钟前延迟计数的开始，所以产生最大为 1 个时钟周期的误差。另外，如果需要开始计数时序的信息，就将 MDmn0 位置“1”，以便能在开始计数时产生中断。

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

(4) 单次计数模式的运行

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

图 6-28 运行时序 (单次计数模式)

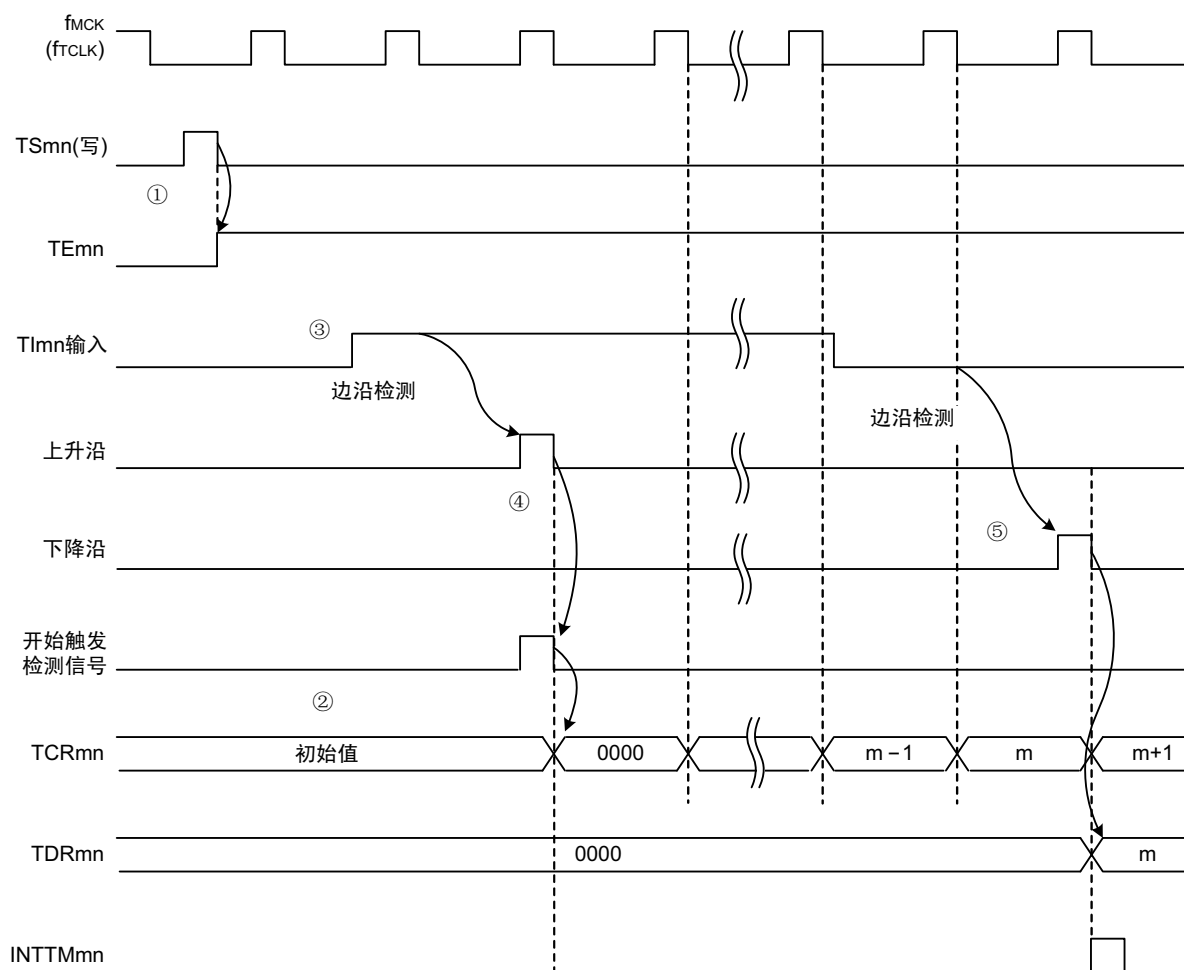


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从 TImn 输入开始再延迟 2 个 f_{MCK} 周期（合计 3 ~ 4 个周期）。1 个周期的误差是因为 TImn 输入与计数时钟 (f_{MCK}) 不同步。

(5) 捕捉 & 单次计数模式的运行（高电平的宽度测量）

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

图 6-29 运行时序（捕捉 & 单次计数模式：高电平的宽度测量）

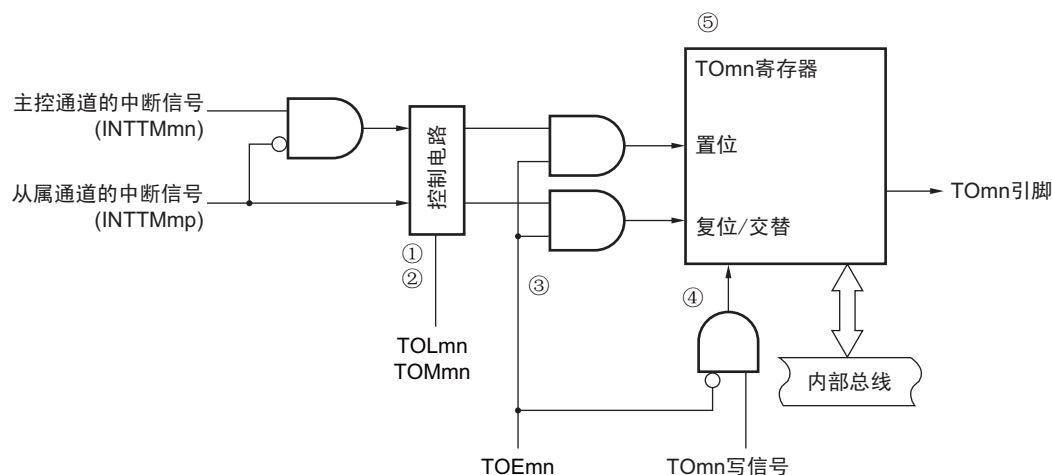


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从 TI_{mn} 输入开始再延迟 2 个 f_{MCK} 周期（合计 3 ~ 4 个周期）。1 个周期的误差是因为 TI_{mn} 输入和计数时钟（ f_{MCK} ）不同步。

6.6 通道输出（TOmn 引脚）的控制

6.6.1 TOmn 引脚输出电路的结构

图 6-30 输出电路的结构



以下说明 TOmn 引脚的输出电路。

- ① 当 TOMmn 位为“0”（主控通道输出模式）时，忽视定时器输出电平寄存器 m（TOLm）的设定值，只将 INTTMmp（从属通道定时器中断）传给定时器输出寄存器 m（TOM）。
- ② 当 TOMmn 位为“1”（从属通道输出模式）时，将 INTTMmn（主控通道定时器中断）和 INTTMmp（从属通道定时器中断）传给 TOM 寄存器。

此时，TOLm 寄存器有效并且进行以下信号的控制：

TOLmn=0 时：正相运行（INTTMmn→置位、INTTMmp→复位）

TOLmn=1 时：反相运行（INTTMmn→复位、INTTMmp→置位）

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

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

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

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

- ④ 在禁止定时器输出（TOEmn=0）的状态下，对象通道的 TOmn 位的写操作（TOmn 写信号）有效。当定时器输出为禁止状态（TOEmn=0）时，不将 INTTMmn（主控通道定时器中断）和 INTTMmp（从属通道定时器中断）传给 TOM 寄存器。
- ⑤ 能随时读 TOM 寄存器，并且能确认 TOmn 引脚的输出电平。

备注 m：单元号（m=0）

n：通道号

n=0~3（主控通道：n=0、2）

p：从属通道号

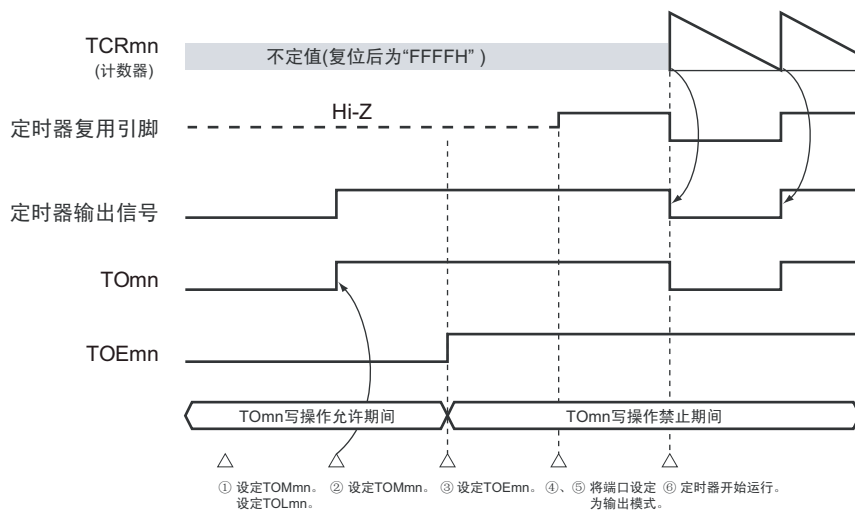
n=0：p=1、2、3

n=2：p=3

6.6.2 TOMn 引脚的输出设定

从 TOMn 输出引脚的初始设定到定时器开始运行的步骤和状态变化如下所示。

图 6-31 从设定定时器的输出到开始运行的状态变化



- ① 设定定时器输出的运行模式。
 - TOMmn位 (0: 主控通道输出模式、1: 从属通道输出模式)
 - TOLmn位 (0: 正逻辑输出、1: 负逻辑输出)
- ② 通过设定定时器输出寄存器m (TOM)，将定时器输出信号设定为初始状态。
- ③ 给TOEmn位写“1”，允许定时器输出 (禁止写TOM寄存器)。
- ④ 通过端口模式控制寄存器 (PMCxx) 将端口设定为数字输入/输出 (参照“6.3.15 控制定时器输入/输出引脚端口功能的寄存器”)。
- ⑤ 将端口的输入/输出设定为输出 (参照“6.3.15 控制定时器输入/输出引脚端口功能的寄存器”)。
- ⑥ 允许定时器运行 (TSmn=1)。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.6.3 通道输出运行的注意事项

(1) 有关定时器运行中的 T_{Om}、T_{OEm}、T_{OLm}、T_{OMm} 寄存器的设定值变更

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

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

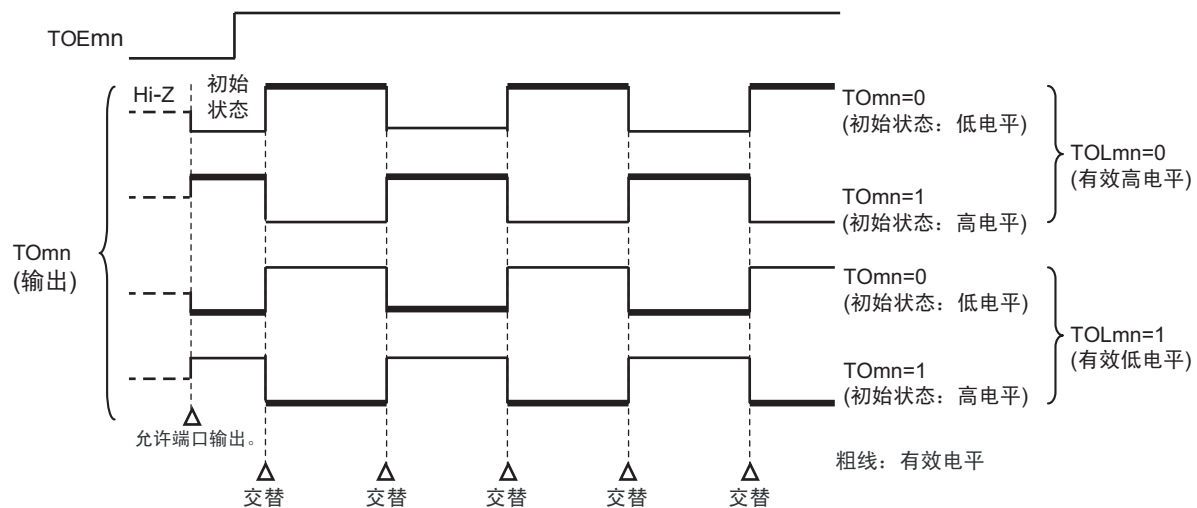
(2) 有关 T_{Om n} 引脚的初始电平和定时器开始运行后的输出电平

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

(a) 在主导通道输出模式（T_{OMmn}=0）中开始运行的情况

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

图 6-32 交替输出时（T_{OMmn}=0）的 T_{Om n} 引脚输出状态

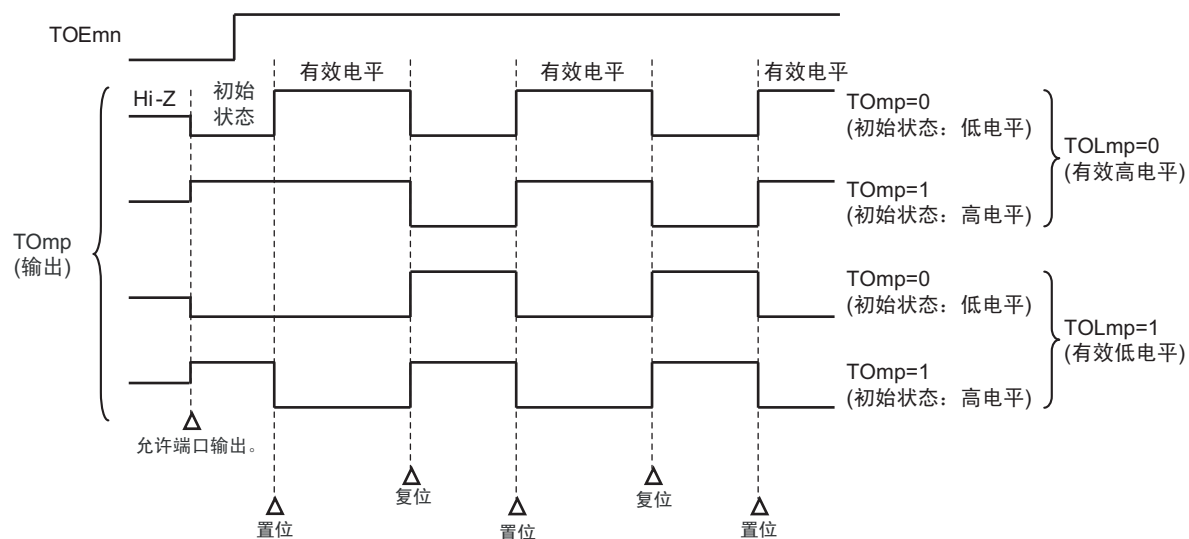


备注 1. 交替: 反相 T_{Om n} 引脚的输出状态。
2. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

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

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

图 6-33 PWM 输出时 (TOMmn=1) 的 TOmn 引脚输出状态



备注 1. 置位: TOmp 引脚的输出信号从无效电平变为有效电平。

复位: TOmp 引脚的输出信号从有效电平变为无效电平。

2. m: 单元号 (m=0) n: 通道号 (p=1 ~ 3)

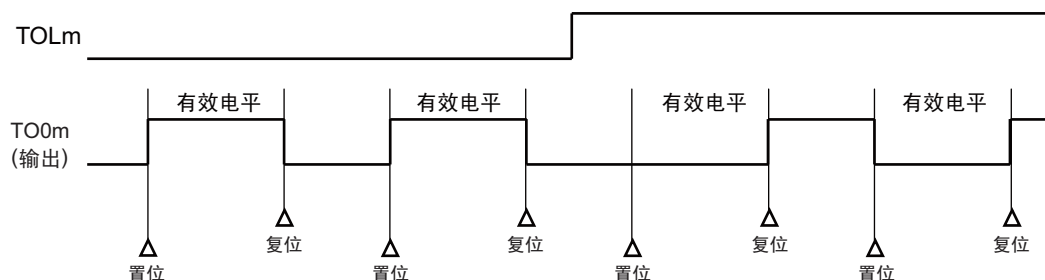
(3) 有关从属通道输出模式 (TOMmn=1) 的 TOMn 引脚变化

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

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

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

图 6-34 在定时器运行中更改 TOLm 寄存器的内容时的运行



备注 1. 置位：TOMn 引脚的输出信号从无效电平变为有效电平。

复位：TOMn 引脚的输出信号从有效电平变为无效电平。

2. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

(b) 置位 / 复位时序

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

当置位条件和复位条件同时产生时，优先复位条件。

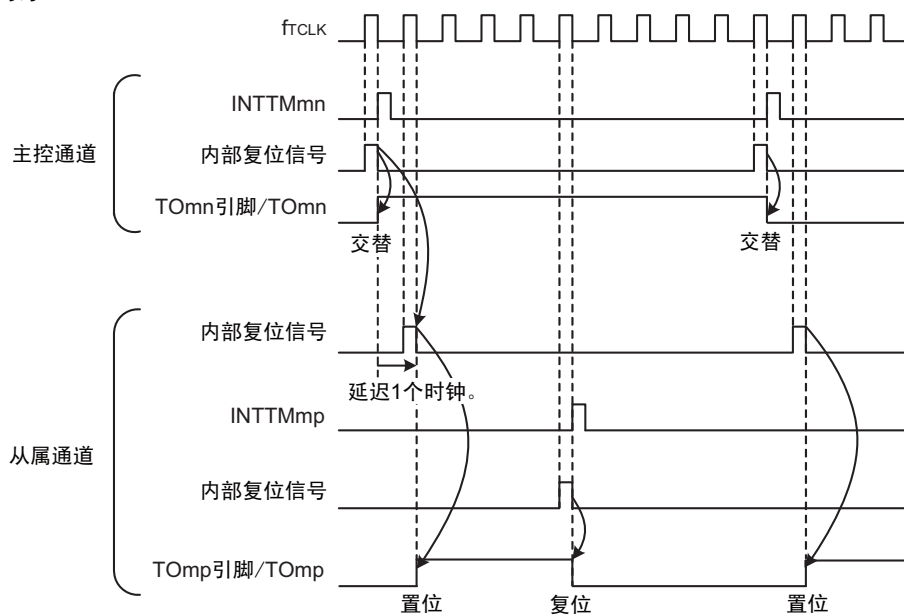
按照以下方法设定主控 / 从属通道时的置位 / 复位运行状态如图 6-35 所示。

主控通道: TOEmn=1、 TOMmn=0、 TOLmn=0

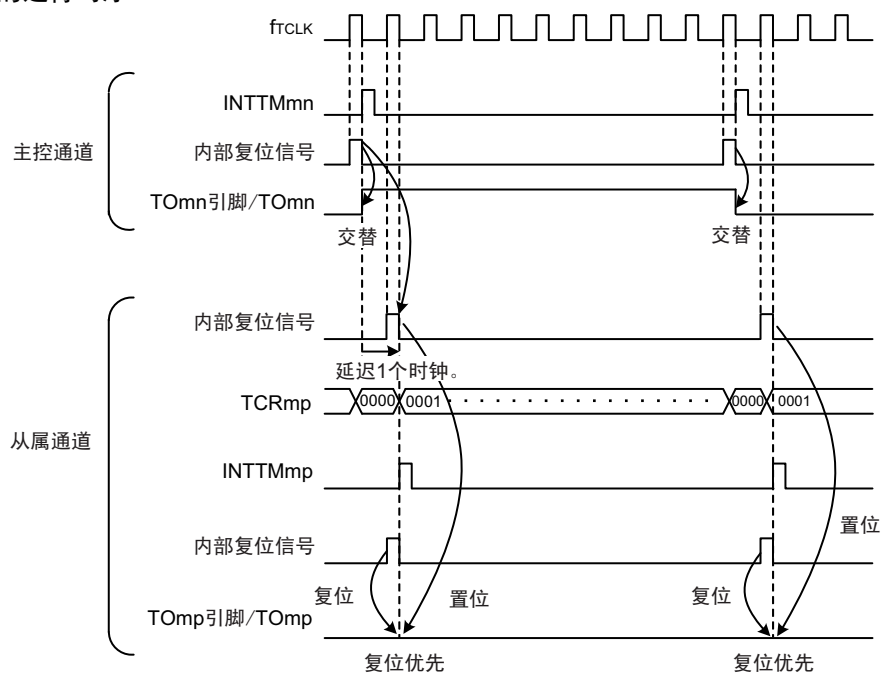
从属通道: TOEmp=1、 TOMmp=1、 TOLmp=0

图 6-35 置位 / 复位时序运行状态

(1) 基本运行时序



(2) 0% 占空比的运行时序



备注 1. 内部复位信号: TOnn 引脚的复位 / 交替信号

内部置位信号: TOnn 引脚的置位信号

2. m: 单元号 (m=0)

n: 通道号

n=0 ~ 3 (主控通道: n=0、2)

p: 从属通道号

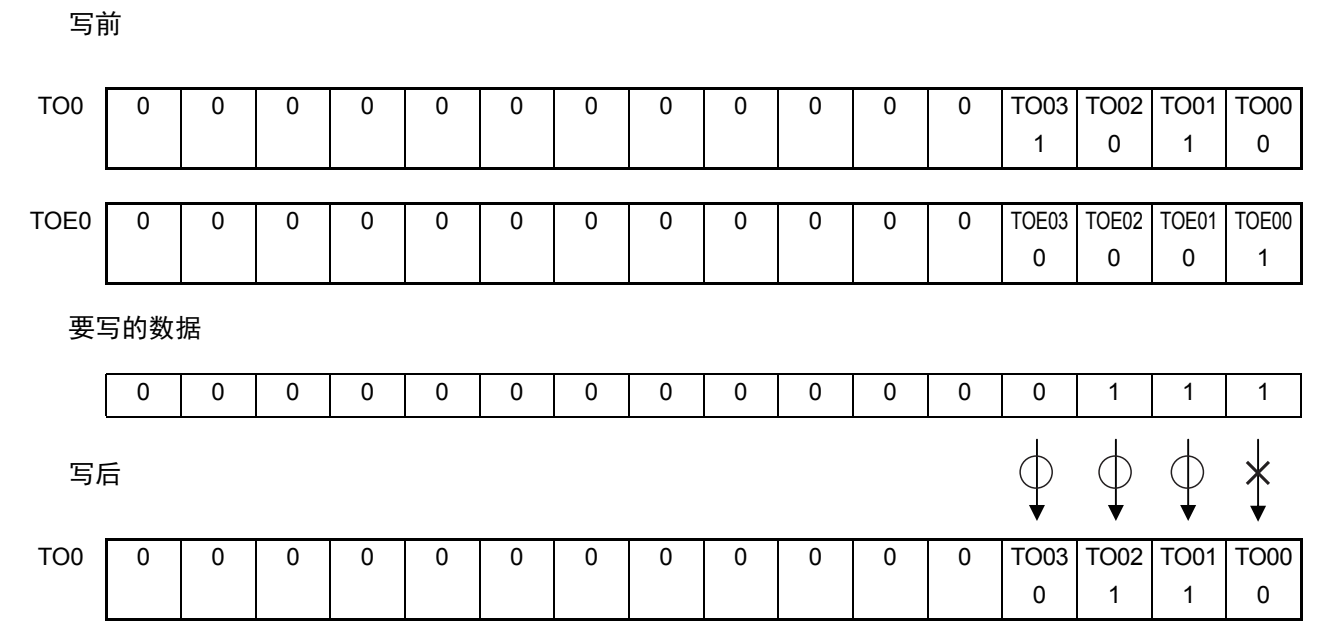
n=0: p=1、2、3

n=2: p=3

6.6.4 TOmn 位的一次性操作

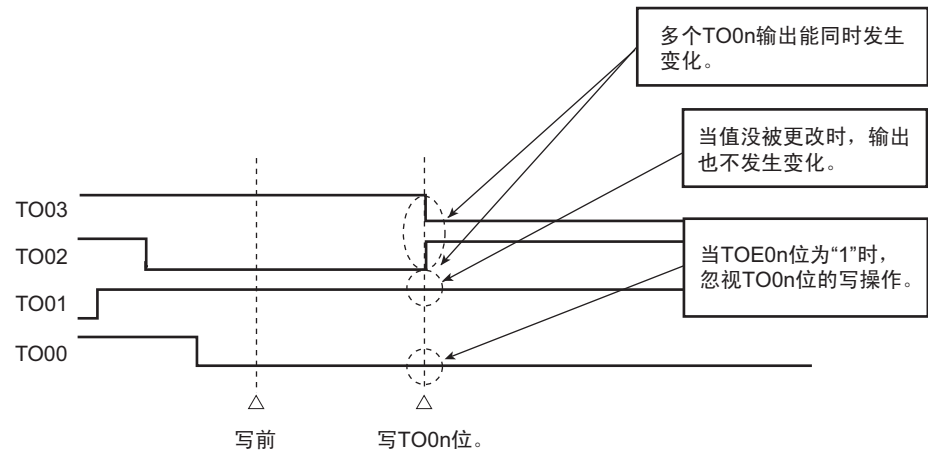
和定时器通道开始寄存器 m (TSm) 相同，定时器输出寄存器 m (TOm) 有全部通道的设定位 (TOmn)，因此能一次性地操作全部通道的 TOmn 位。另外，能给只想作为操作对象的通道输出 (TOmn) 的 TOmn 位写数据 (TOEmn=0)，因此也能操作任意的位。

图 6-36 TO0n 位的一次性操作例子



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

图 6-37 一次性操作 TO0n 位时的 TO0n 引脚状态



备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.6.5 有关开始计数时的定时器中断和 T0mn 引脚输出

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

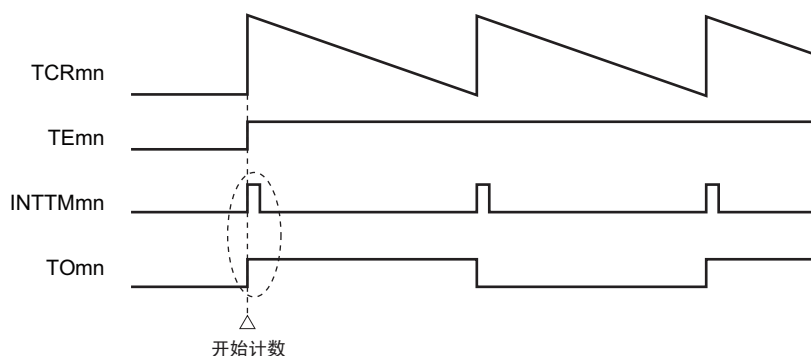
当 MDmn0 位为“1”时，能通过产生定时器中断（INTTMmn）得知计数的开始时序。

在其他模式中，不控制开始计数时的定时器中断和 T0mn 输出。

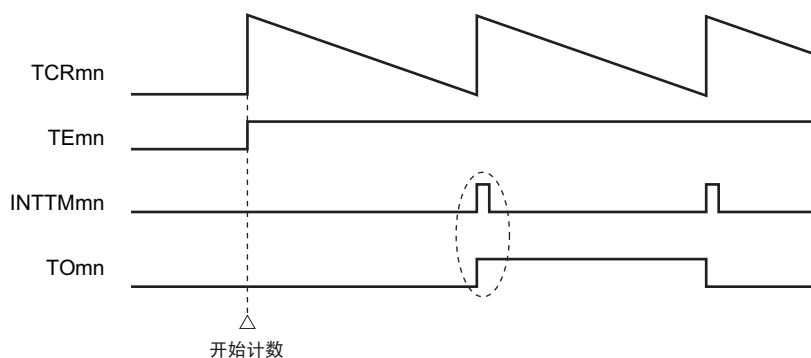
设定为间隔定时器模式（TOEmn=1、TOMmn=0）时的运行例子如下所示。

图 6-38 开始计数时的定时器中断和 T0mn 输出的运行例子

(a) MDmn0 位为“1”的情况



(b) MDmn0 位为“0”的情况



当 MDmn0 位为“1”时，在开始计数时输出定时器中断（INTTMmn）并且 T0mn 进行交替输出。

当 MDmn0 位为“0”时，在开始计数时不输出定时器中断（INTTMmn）并且 T0mn 也不发生变化，而在对 1 个周期进行计数后输出 INTTMmn 并且 T0mn 进行交替输出。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

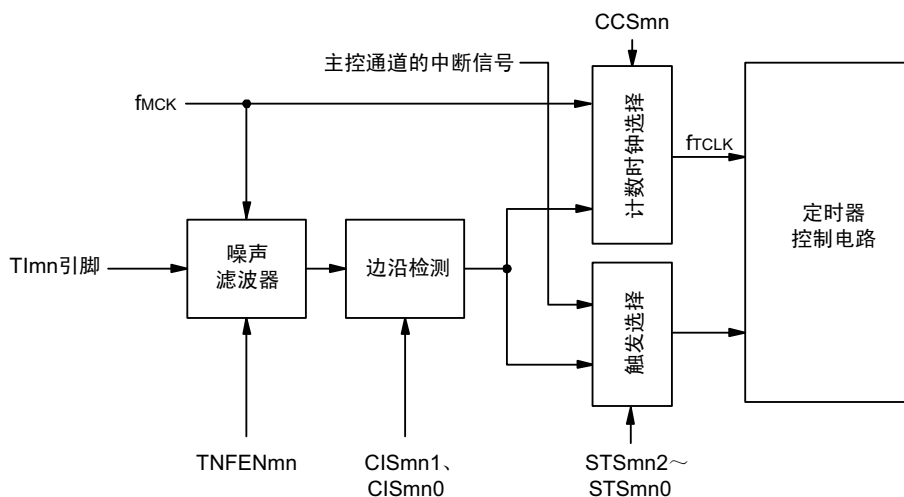
6.7 定时器输入（Tlmn）的控制

6.7.1 Tlmn 引脚输入电路的结构

定时器输入引脚的信号通过噪声滤波器和边沿检测电路输入到定时器控制电路。

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

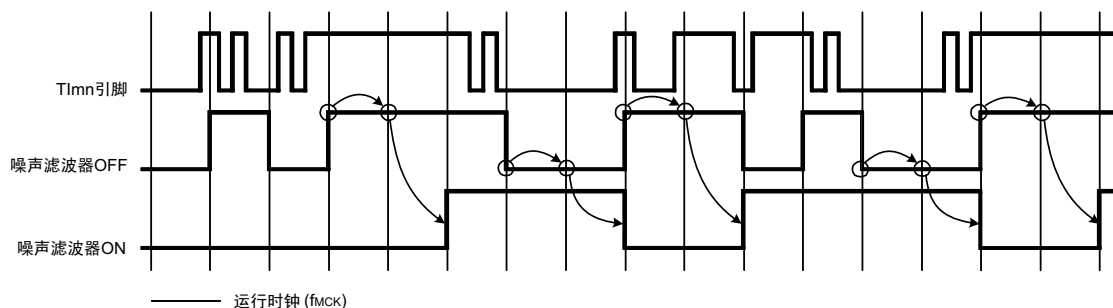
图 6-39 输入电路的结构



6.7.2 噪声滤波器

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

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



注意 Tlmn 引脚的输入波形用于说明噪声滤波器 ON 或者 OFF 的运行。实际使用时，必须按照“31.4 AC 特性”所示的 Tlmn 输入高低电平宽度进行输入。

6.7.3 操作通道输入时的注意事项

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

(1) 噪声滤波器为 OFF 的情况

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

(2) 噪声滤波器为 ON 的情况

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

6.8 定时器阵列单元的独立通道运行功能

6.8.1 作为间隔定时器 / 方波输出的运行

(1) 间隔定时器

能用作以固定间隔产生 INTTMmn（定时器中断）的基准定时器。

中断产生周期能用以下计算式进行计算：

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

(2) 作为方波输出的运行

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

TOmn 输出方波的周期和频率能用以下计算式进行计算：

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

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

在间隔定时器模式中，定时器计数寄存器 mn（TCRmn）用作递减计数器。

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

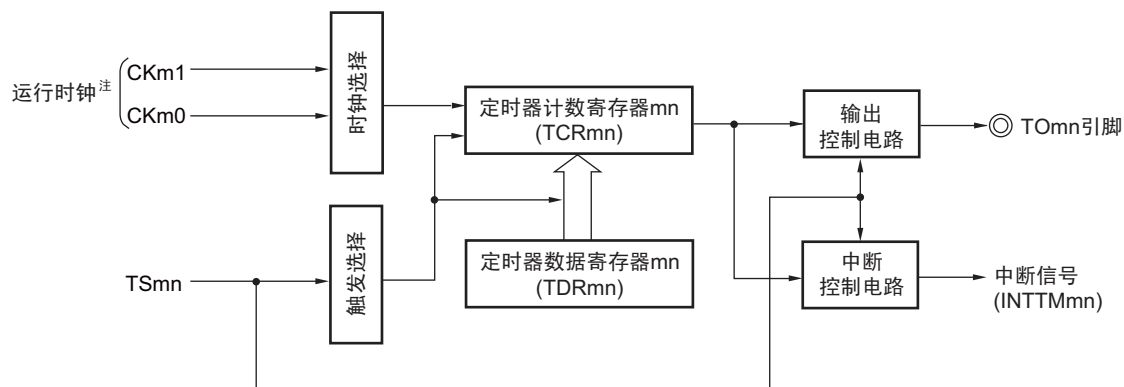
然后，TCRmn 寄存器通过计数时钟进行递减计数。

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

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

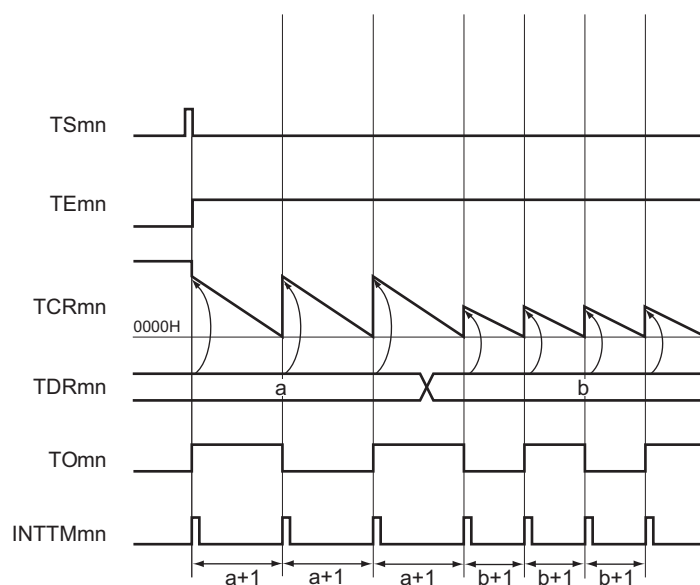
备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-41 作为间隔定时器 / 方波输出运行的框图



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

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

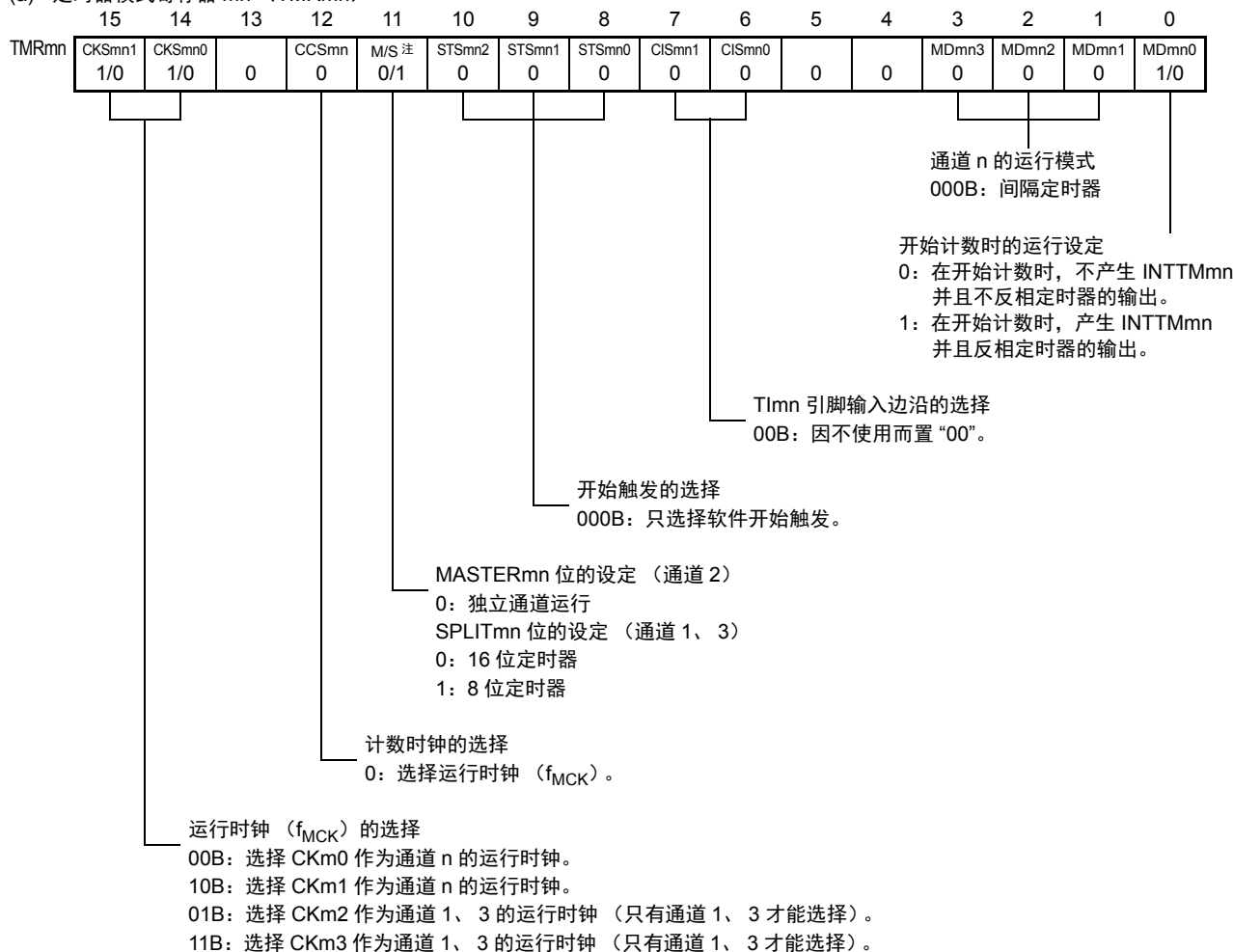


备注 1. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

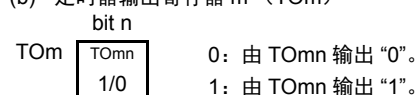
2. TSmn : 定时器通道开始寄存器 m (TSM) 的 bit n
- TEmn : 定时器通道允许状态寄存器 m (TEM) 的 bit n
- TCRmn : 定时器计数寄存器 mn (TCRmn)
- TDRmn : 定时器数据寄存器 mn (TDRmn)
- T0mn : T0mn 引脚输出信号

图 6-43 间隔定时器 / 方波输出时的寄存器设定内容例子

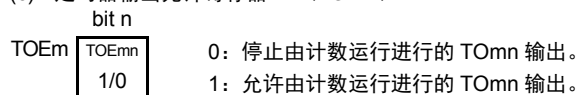
(a) 定时器模式寄存器 mn (TMRmn)



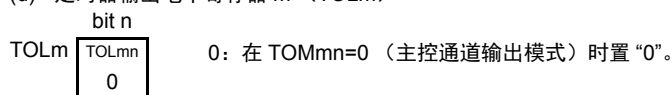
(b) 定时器输出寄存器 m (TOM)



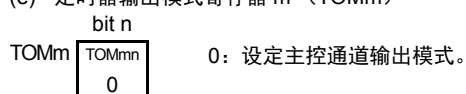
(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



(e) 定时器输出模式寄存器 m (TOMm)



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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-44 间隔定时器 / 方波输出功能时的操作步骤 (1/2)

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

图 6-44 间隔定时器 / 方波输出功能时的操作步骤 (2/2)

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.8.2 作为外部事件计数器的运行

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

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

在事件计数器模式中，定时器计数寄存器 mn（TCRmn）用作递减计数器。

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

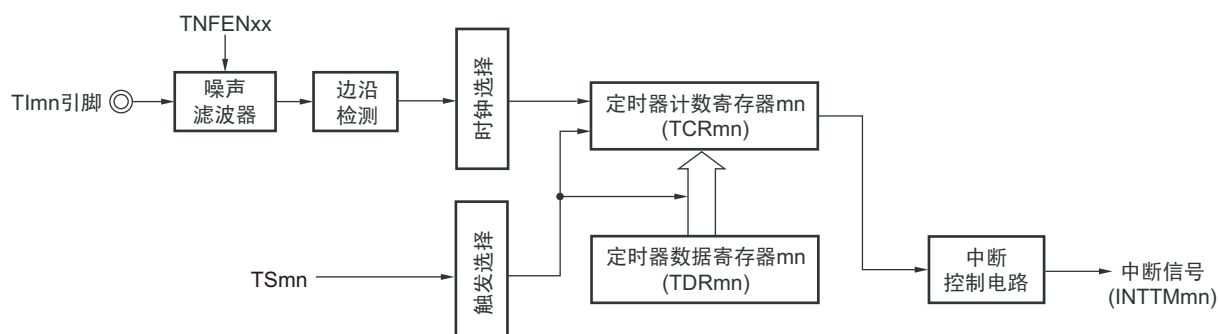
TCRmn 寄存器在检测到 TIMn 引脚输入的有效边沿的同时进行递减计数。如果 TCRmn 变为“0000H”，就再次装入 TDRmn 寄存器的值并且输出 INTTMmn。

此后，继续同样的运行。

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

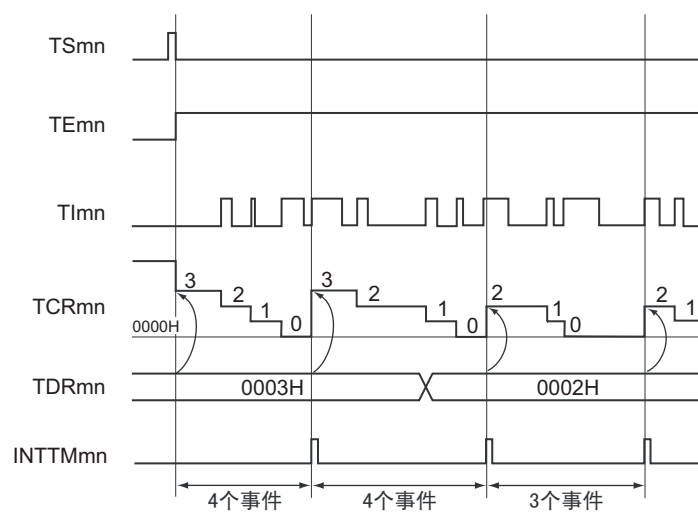
能随时改写 TDRmn 寄存器，改写的 TDRmn 寄存器的值在下一个计数期间有效。

图 6-45 作为外部事件计数器运行的框图



备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-46 作为外部事件计数器运行的基本时序例子

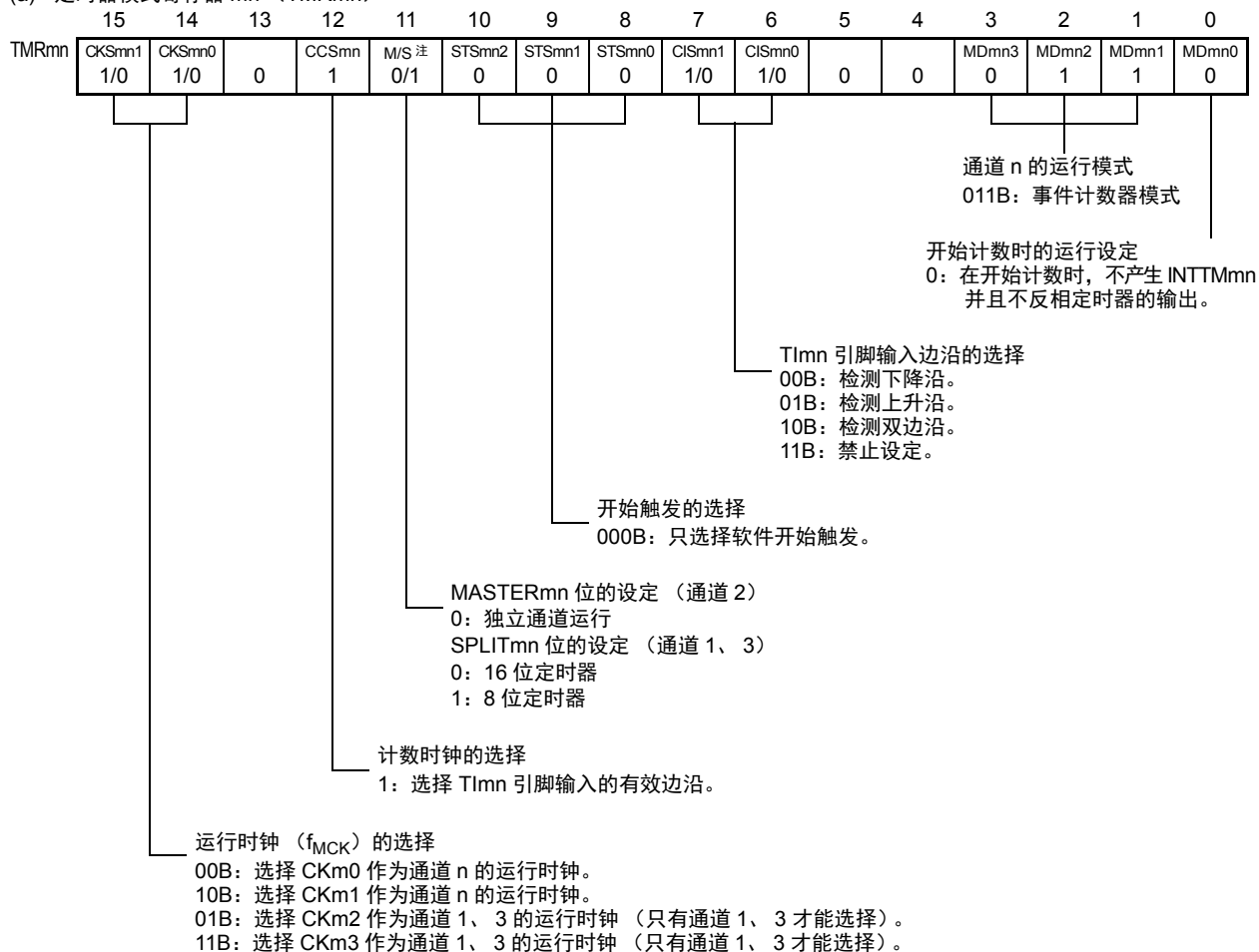


备注 1. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

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

图 6-47 外部事件计数器模式时的寄存器设定内容例子

(a) 定时器模式寄存器 mn (TMRmn)



(b) 定时器输出寄存器 m (TOM)

bit n
 TOM TOMn
0 0: 由 TOMn 输出 "0"。

(c) 定时器输出允许寄存器 m (TOEm)

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

(d) 定时器输出电平寄存器 m (TOLm)

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

(e) 定时器输出模式寄存器 m (TOMm)

bit n
 TOMm TOMmn
0 0: 设定主控通道输出模式。

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-48 外部事件计数器功能时的操作步骤

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.8.3 作为分频器的运行（只限于单元0的通道0）

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，继续进行计数。

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 6-49 作为分频器运行的框图

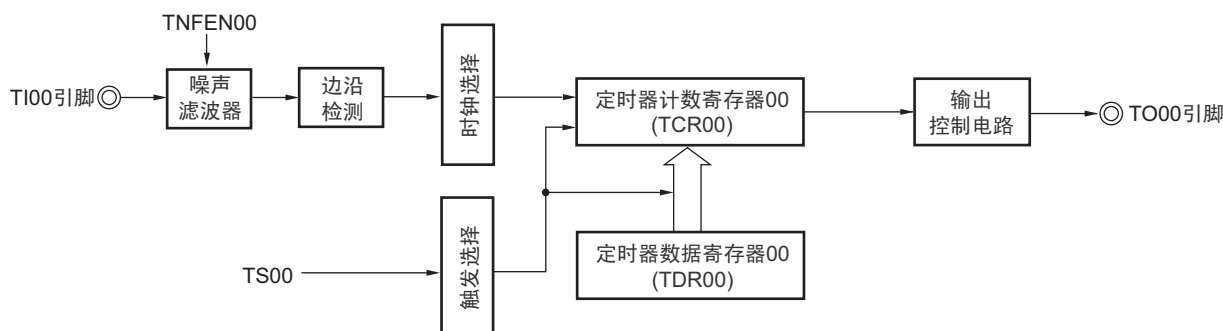
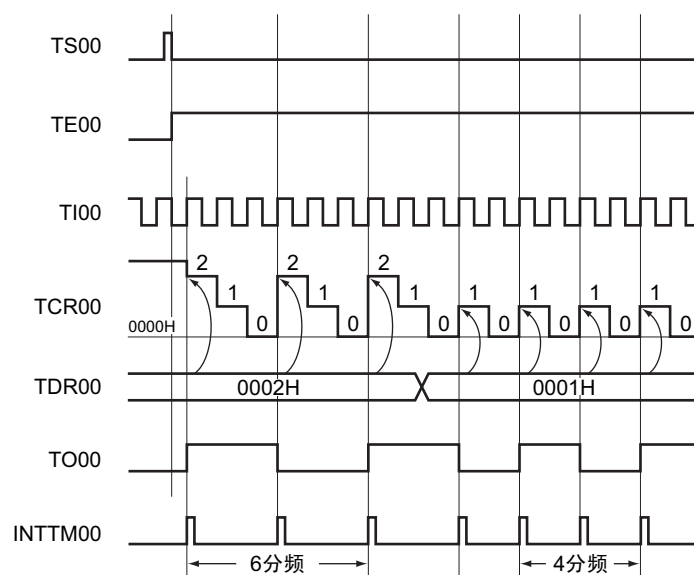


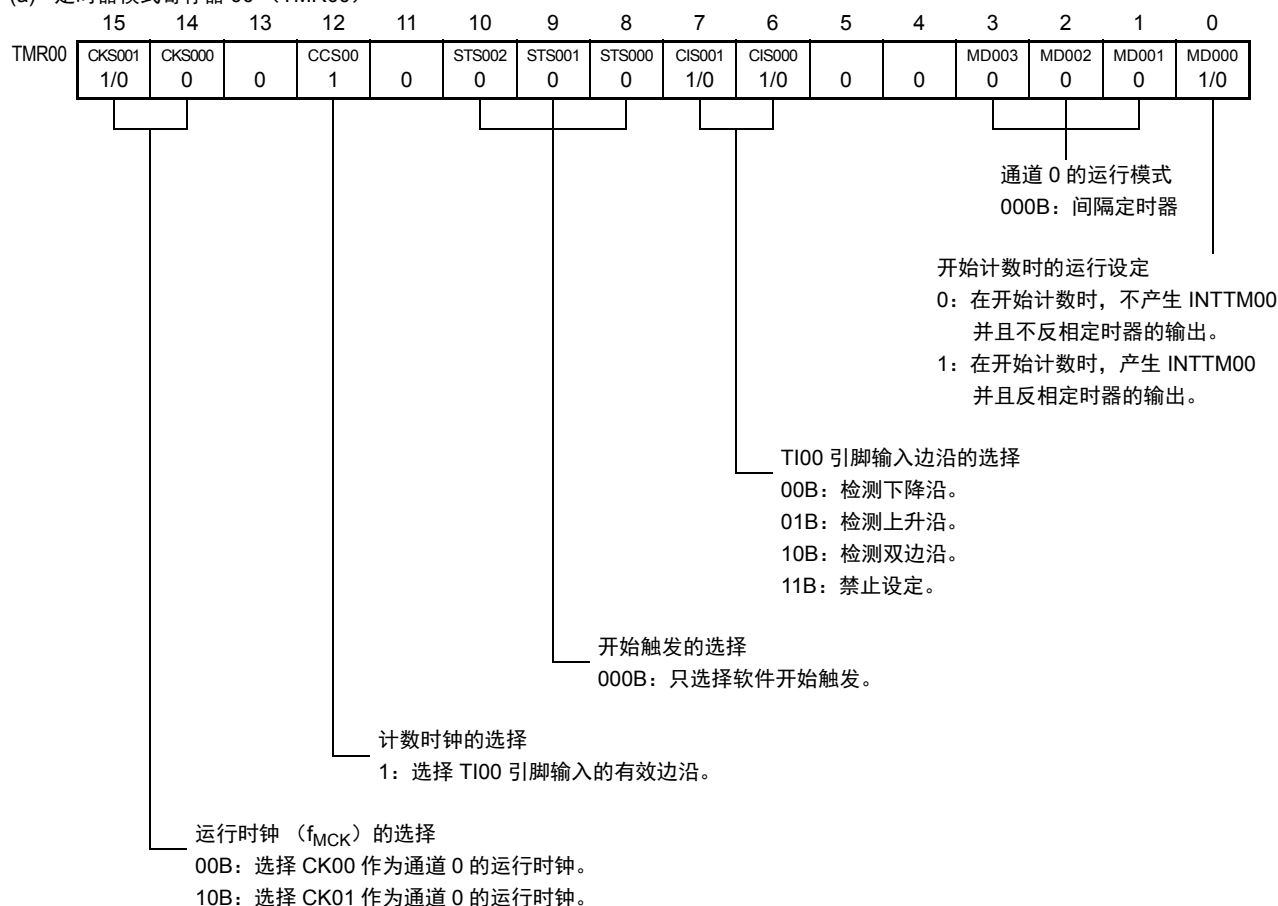
图 6-50 作为分频器运行的基本时序例子 (MD000=1)



- 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 6-51 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	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>TOE00</td></tr><tr><td>1/0</td></tr></table>	TOE00	1/0
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 (TOM0=0) 中置“0”。

(e) 定时器输出模式寄存器 0 (TOM0)

bit0	
TOM0	TOM00 0
0: 设定主控通道输出模式。	

图 6-52 分频器功能时的操作步骤

	软件操作	硬件状态
TAU 初始 设定		定时器单元 0 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置“1”。	定时器单元 0 的输入时钟处于提供状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定 CK00 ~ CK03 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器 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”。	TO00 引脚处于 Hi-Z 输出状态。
	设定 TO00 位并且确定 TO00 输出的初始电平。→	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 的初始设定电平。
	将 TOE00 位置“1”, 允许 TO00 运行。→	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 的设定电平。
重新 开始 运行	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。 计数器 (TCR00) 进行递减计数。如果计数到“0000H”, 就再次将 TDR00 寄存器的值装入 TCR00 寄存器, 继续进行计数。当检测到 TCR00 为“0000H”时, 产生 INTTM00 并且 TO00 进行交替输出。此后, 重复此运行。
	停止运行	将 TT00 位置“1”。→ 因为 TT00 位是触发位, 所以自动返回到“0”。 TE00 位为“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 的设定电平。
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在的值后将 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”。→	定时器单元 0 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。 (TO00 位变为“0”并且 TO00 引脚变为端口功能)

6.8.4 作为输入脉冲间隔测量的运行

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

脉冲间隔能用以下计算式进行计算：

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

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

在捕捉模式中，定时器计数寄存器 mn（TCRmn）用作递增计数器。

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

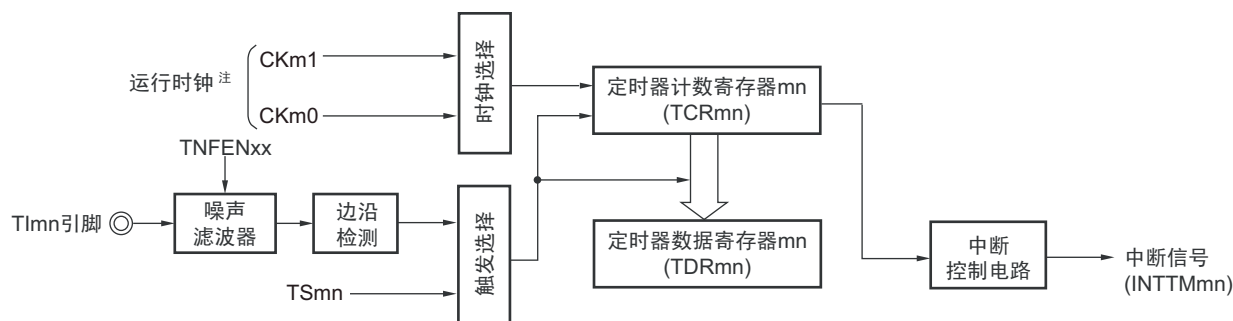
如果检测到 TImn 引脚输入的有效边沿，就将 TCRmn 寄存器的计数值传送（捕捉）到定时器数据寄存器 mn（TDRmn），同时将 TCRmn 寄存器清“0000H”，然后输出 INTTMmn。此时，如果计数器发生上溢，就将定时器状态寄存器 mn（TSRmn）的 OVF 位置“1”。如果计数器没有发生上溢，就清除 OVF 位。此后，继续同样的运行。

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

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

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

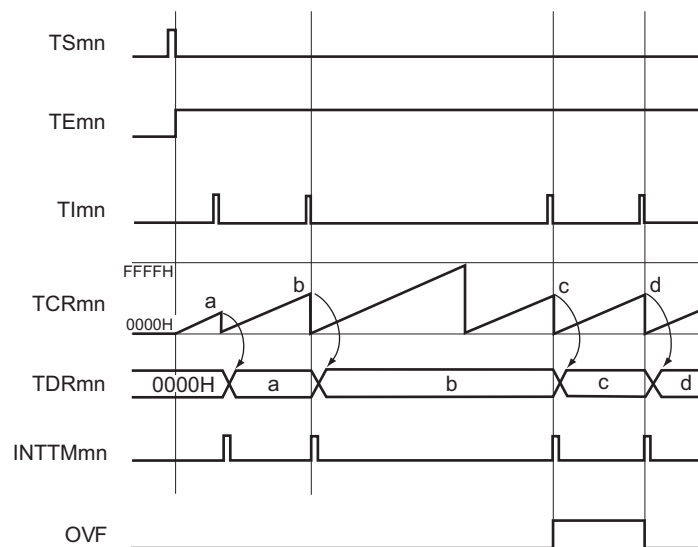
图 6-53 作为输入脉冲间隔测量运行的框图



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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

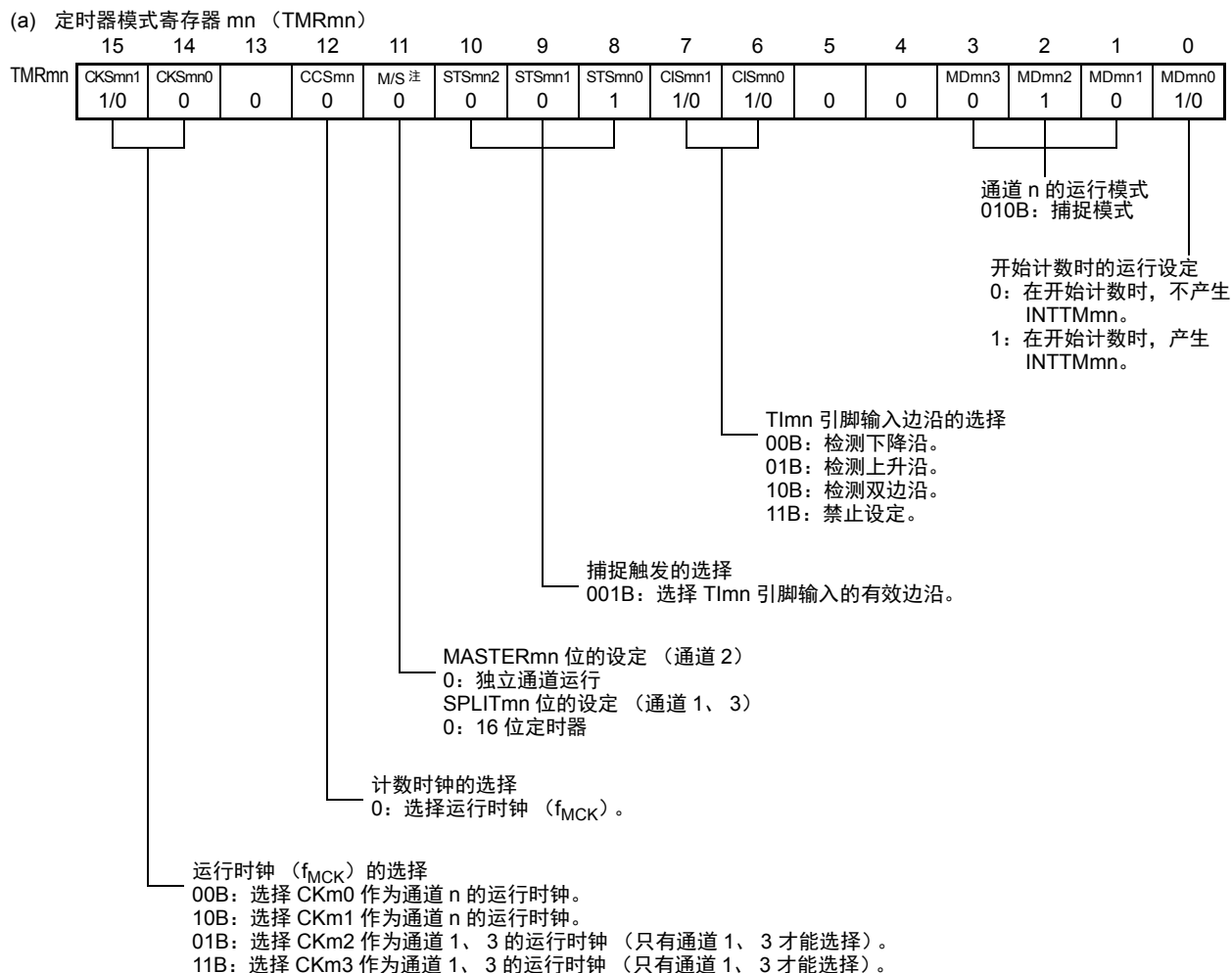
图 6-54 作为输入脉冲间隔测量的运行基本时序例子 (MDmn0=0)



备注 1. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

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

图 6-55 测量输入脉冲间隔时的寄存器设定内容例子



(b) 定时器输出寄存器 m (TOM)

bit n

TOM TOMn
0 0: 由 TOMn 输出 "0"。

(c) 定时器输出允许寄存器 m (TOEm)

bit n

TOEm TOEmn
0 0: 停止由计数运行进行的 TOMn 输出。

(d) 定时器输出电平寄存器 m (TOLm)

bit n

TOLm TOLmn
0 0: 在主导通道输出模式 (TOMmn=0) 中置 "0"。

(e) 定时器输出模式寄存器 m (TOMm)

bit n

TOMm TOMmn
0 0: 设定主导通道输出模式。

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-56 输入脉冲间隔测量功能时的操作步骤

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.8.5 作为输入信号高低电平宽度测量的运行

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

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

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

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

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

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

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

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

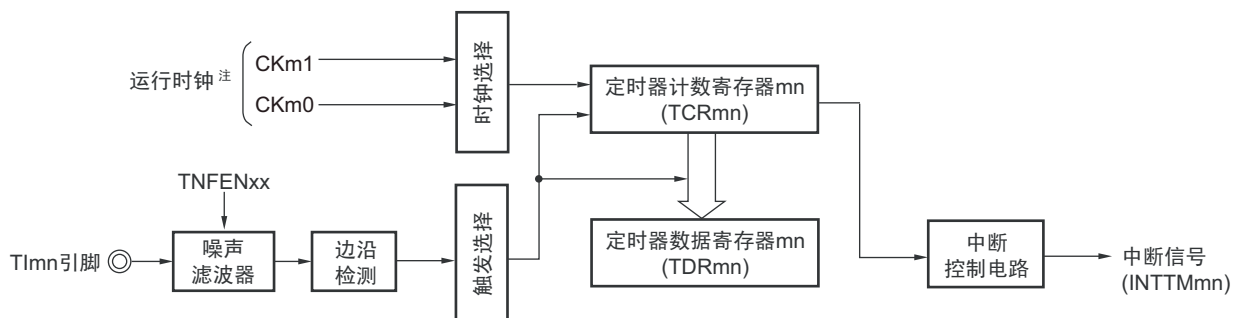
即使计数器进行了 2 个周期或者 2 个周期以上的完整计数，也认为发生上溢而将 TSRmn 寄存器的 OVF 位置“1”。但是，在发生 2 次或者 2 次以上的上溢时，无法通过 OVF 位正常测量间隔值。

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

TMRmn 寄存器的 CISmn1、CISmn0=10B：测量低电平宽度。

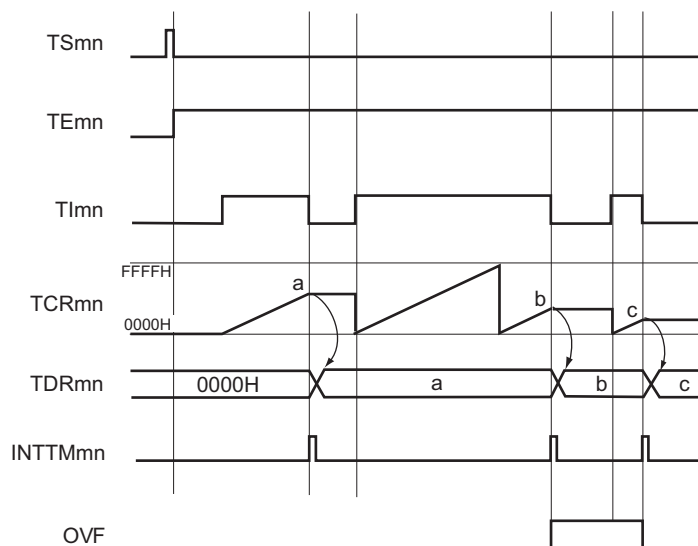
TMRmn 寄存器的 CISmn1、CISmn0=11B：测量高电平宽度。

图 6-57 作为输入信号高低电平宽度测量运行的框图



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

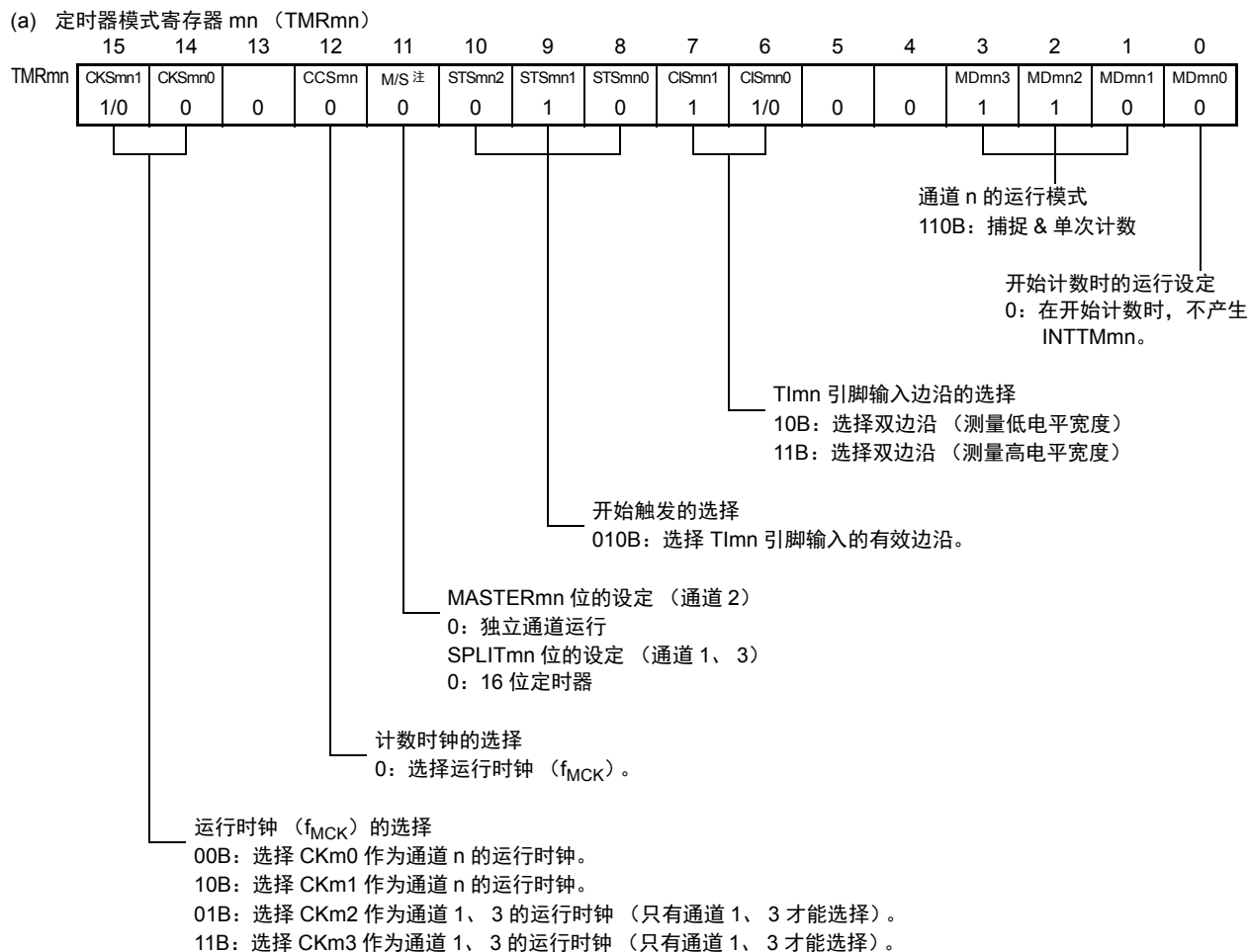
图 6-58 作为输入信号高低电平宽度测量的运行基本时序例子



备注 1. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

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

图 6-59 测量输入信号的高低电平宽度时的寄存器设定内容例子



(b) 定时器输出寄存器 m (TOM)

bit n
 TOMn
 0: 由 TOMn 输出 "0"。

(c) 定时器输出允许寄存器 m (TOEm)

bit n
 TOEmn
 0: 停止由计数运行进行的 TOMn 输出。

(d) 定时器输出电平寄存器 m (TOLm)

bit n
 TOLmn
 0: 在主导通道输出模式 (TOMmn=0) 中置 "0"。

(e) 定时器输出模式寄存器 m (TOMm)

bit n
 TOMmn
 0: 设定主导通道输出模式。

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-60 输入信号高低电平宽度测量功能时的操作步骤

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.8.6 作为延迟计数器的运行

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

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

中断产生周期能用以下计算式进行计算：

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

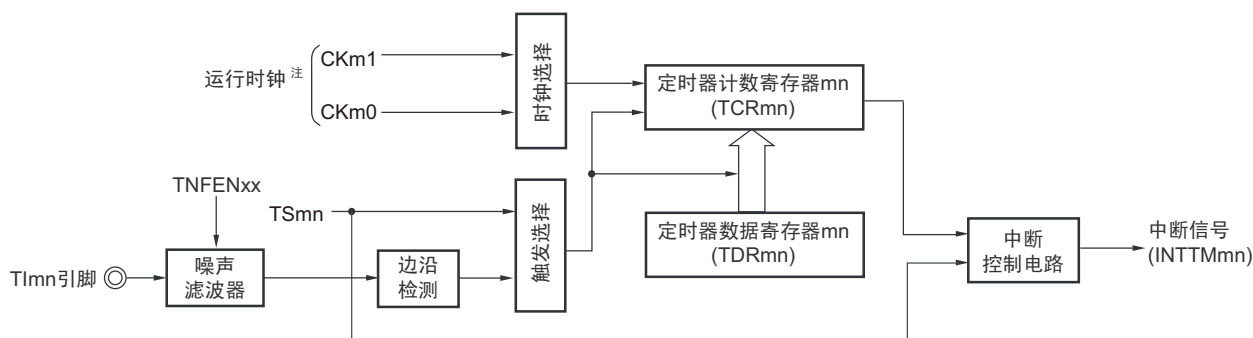
在单次计数模式中，定时器计数寄存器 mn（TCRmn）用作递减计数器。

如果将定时器通道开始寄存器 m（TSM）的通道开始触发位（TSmn、TSHm1、TSHm3）置“1”，TEMn 位、TEHm1 位和 TEHm3 位就变为“1”，并且进入 TIMn 引脚的有效边沿检测等待状态。

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

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

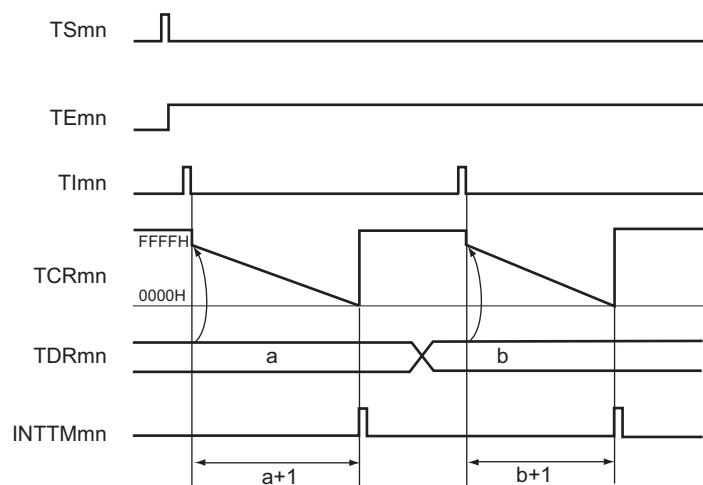
图 6-61 作为延迟计数器运行的框图



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

备注 m：单元号（m=0） n：通道号（n=0～3）

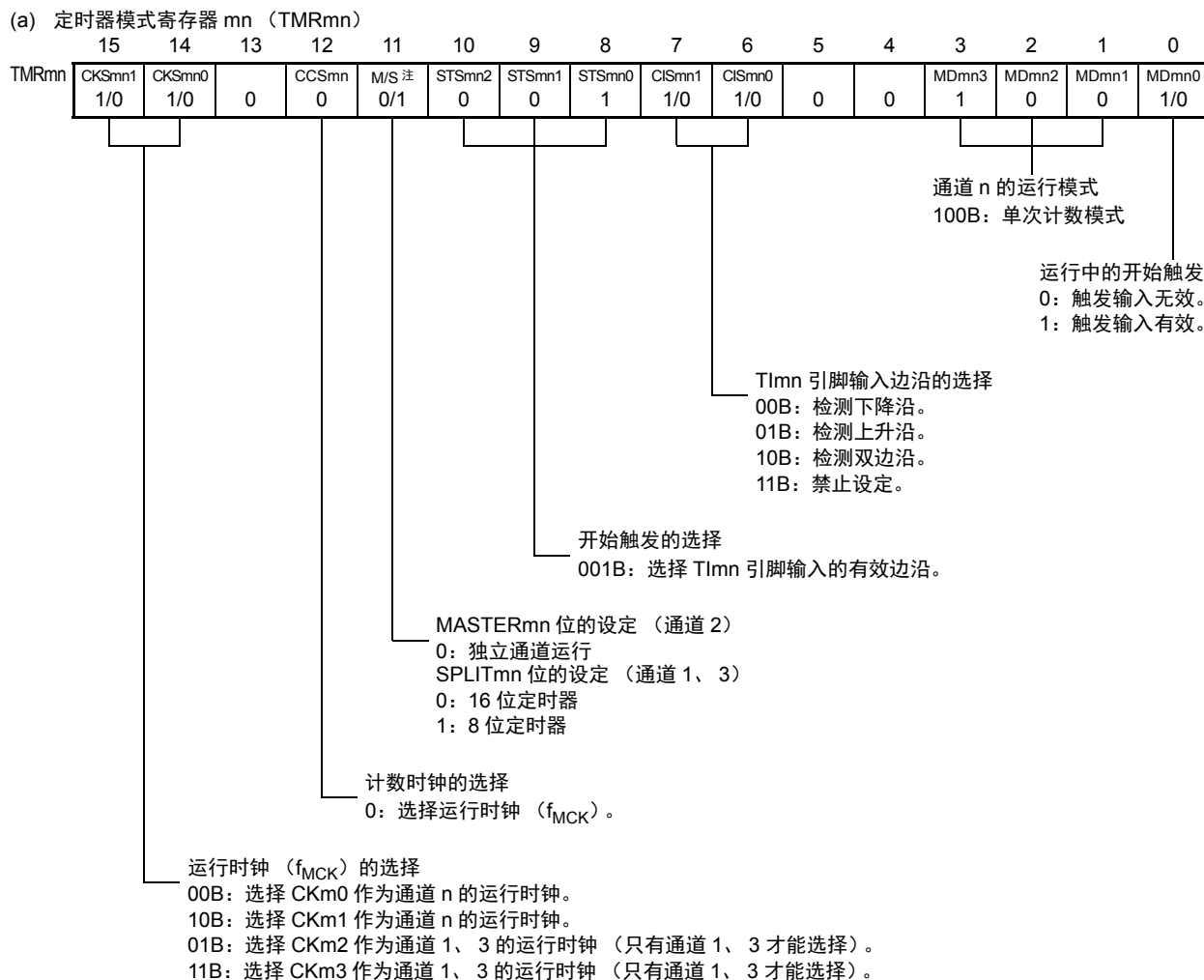
图 6-62 作为延迟计数器的运行基本时序例子



备注 1. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

2. TSmn : 定时器通道开始寄存器 m (TSm) 的 bit n
- TE_{mn} : 定时器通道允许状态寄存器 m (TE_m) 的 bit n
- TImn : TImn 引脚输入信号
- TCR_{mn} : 定时器计数寄存器 mn (TCR_{mn})
- TDR_{mn} : 定时器数据寄存器 mn (TDR_{mn})

图 6-63 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 m (TOM)

bit n

TOM

TOMn
0

 0: 由 TOMn 输出 "0"。

(c) 定时器输出允许寄存器 m (TOEm)

bit n

TOEm

TOEmn
0

 0: 停止由计数运行进行的 TOMn 输出。

(d) 定时器输出电平寄存器 m (TOLm)

bit n

TOLm

TOLmn
0

 0: 在主导通道输出模式 (TOMmn=0) 中置 "0"。

(e) 定时器输出模式寄存器 m (TOMm)

bit n

TOMm

TOMmn
0

 0: 设定主导通道输出模式。

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-64 延迟计数器功能时的操作步骤

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

重新
开始
运行

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.9 定时器阵列单元的多通道联动运行功能

6.9.1 作为单触发脉冲输出功能的运行

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

$$\begin{aligned}\text{延迟} &= \{\text{TDRmn (主控) 的设定值} + 2\} \times \text{计数时钟周期} \\ \text{脉宽} &= \{\text{TDRmp (从属) 的设定值}\} \times \text{计数时钟周期}\end{aligned}$$

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

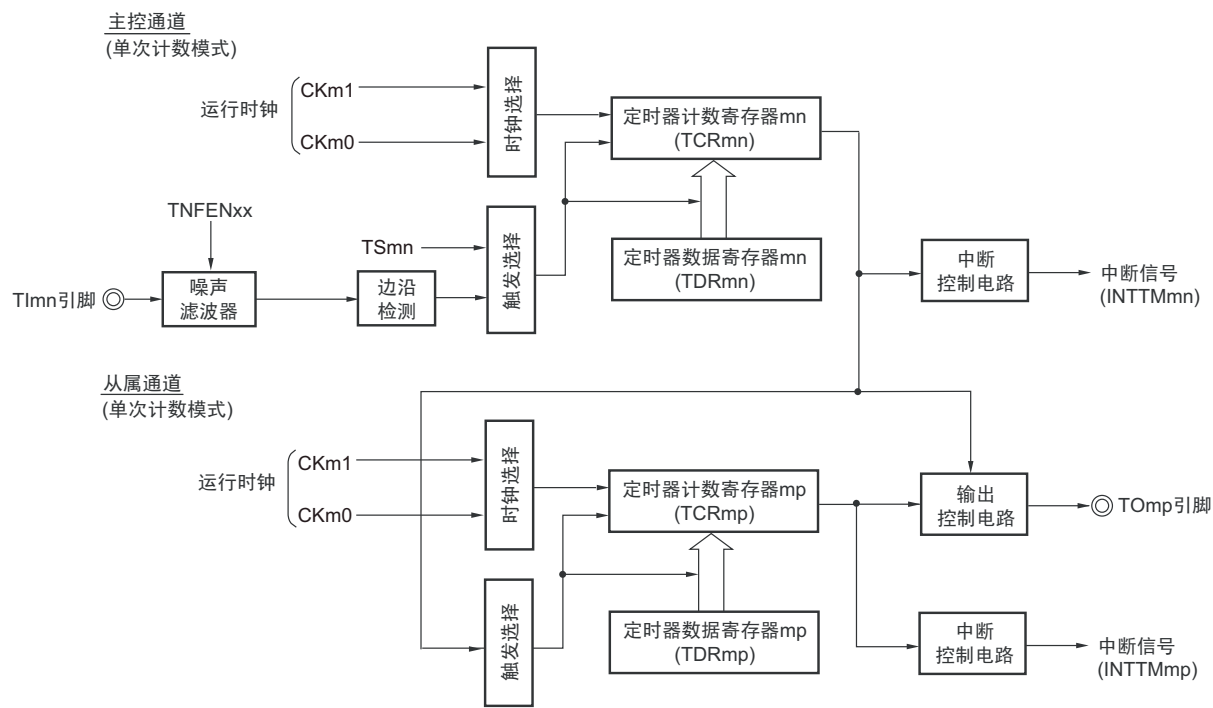
在单次计数模式中，从属通道运行并且对脉宽进行计数。将主控通道的 INTTMmn 作为开始触发，从属通道的 TCRmp 寄存器开始运行并且装入 TDRmp 寄存器的值。TCRmp 寄存器通过计数时钟，从装入的 TDRmp 寄存器值开始递减计数。如果计数值变为“0000H”，就输出 INTTMmp，并且在检测到下一个开始触发（主控通道的 INTTMmn）前停止计数。在从主控通道产生 INTTMmn 并且经过1个计数时钟后，TOmp 的输出电平变为有效电平，如果 TCRmp 变为“0000H”，就变为无效电平。

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

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

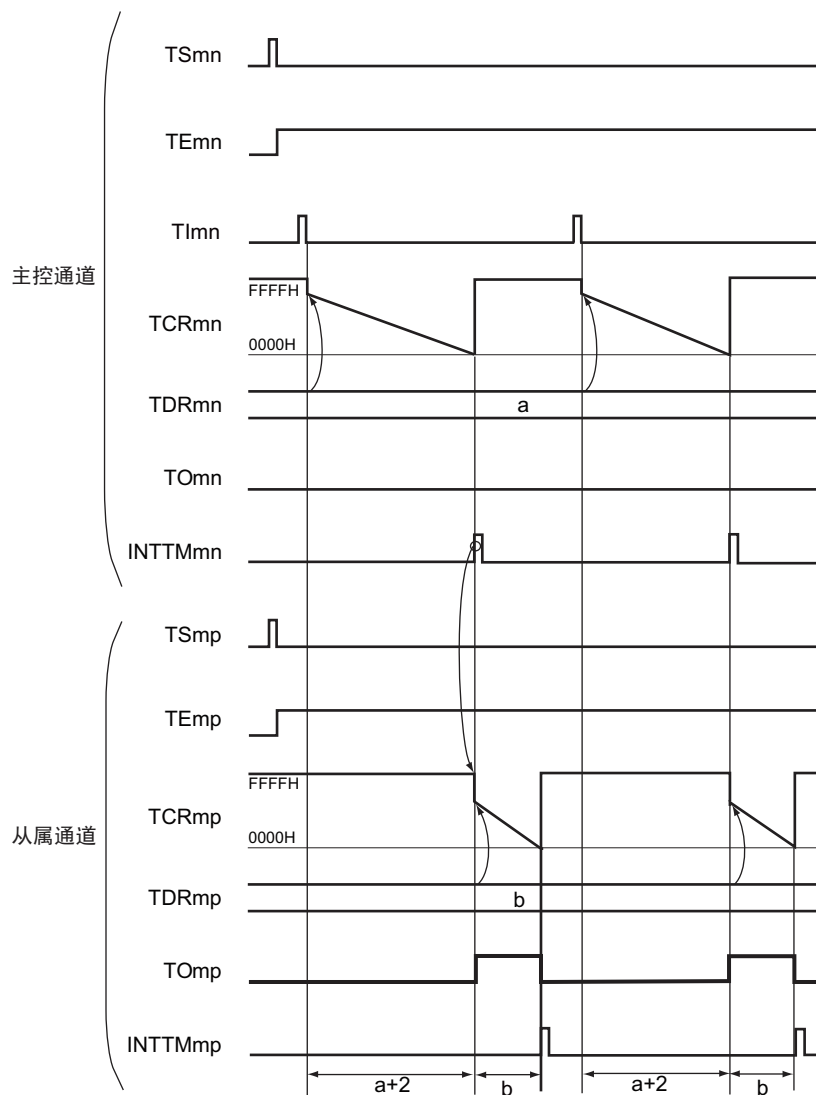
备注 m: 单元号 (m=0) n: 主控通道号 (n=0、2) p: 从属通道号 (n=0: p=1、2、3, n=2: p=3)

图 6-65 作为单触发脉冲输出功能运行的框图



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

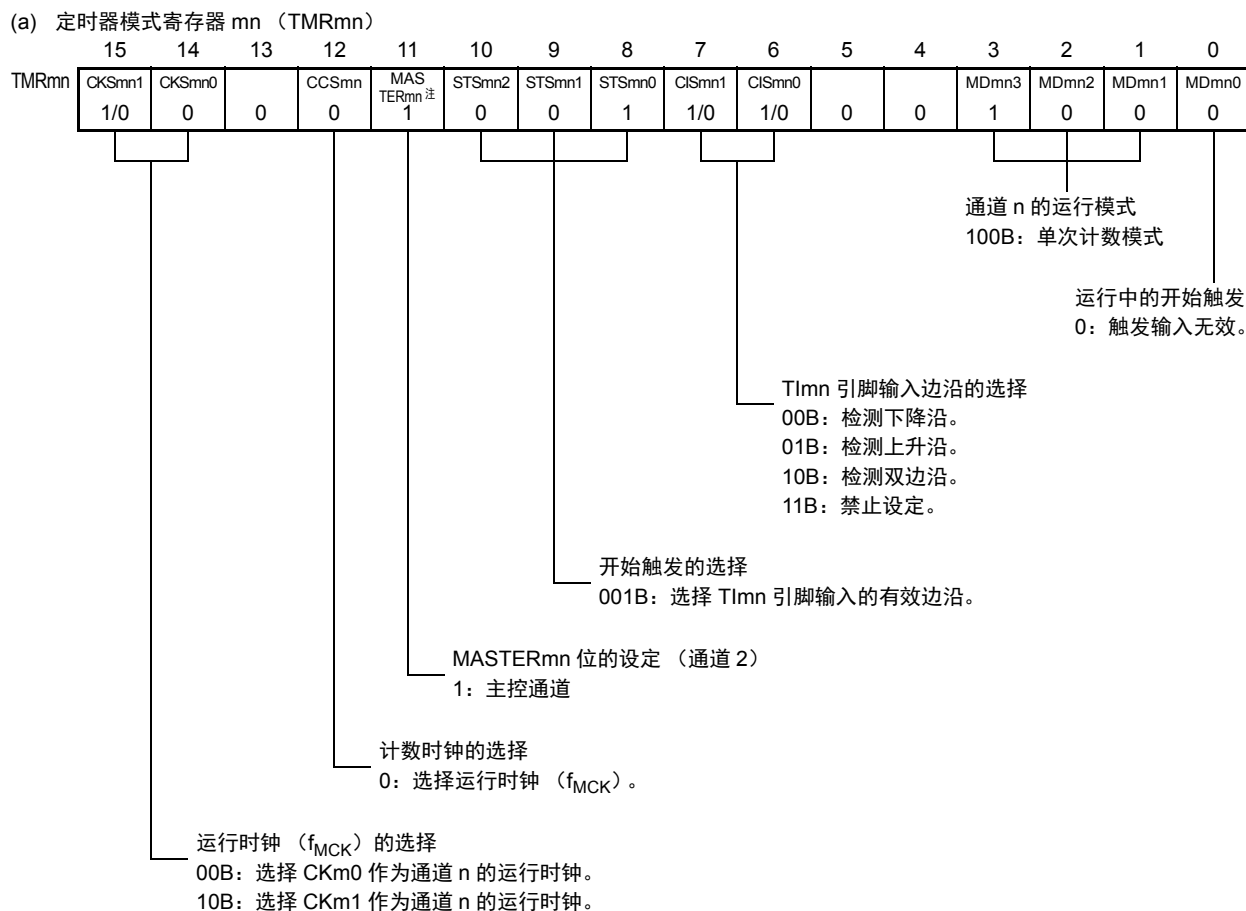
图 6-66 作为单触发脉冲输出功能的运行基本时序例子



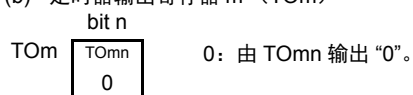
备注 1. m: 单元号 (m=0) n: 主控通道号 (n=0、2) p: 从属通道号 (n=0: p=1、2、3, n=2: p=3)

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

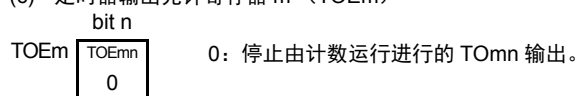
图 6-67 单触发脉冲输出功能时（主控通道）的寄存器设定内容例子



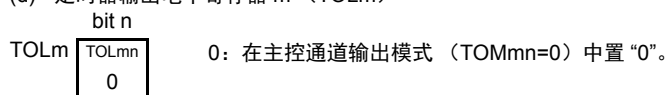
(b) 定时器输出寄存器 m (TOM)



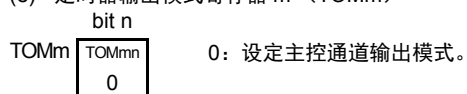
(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



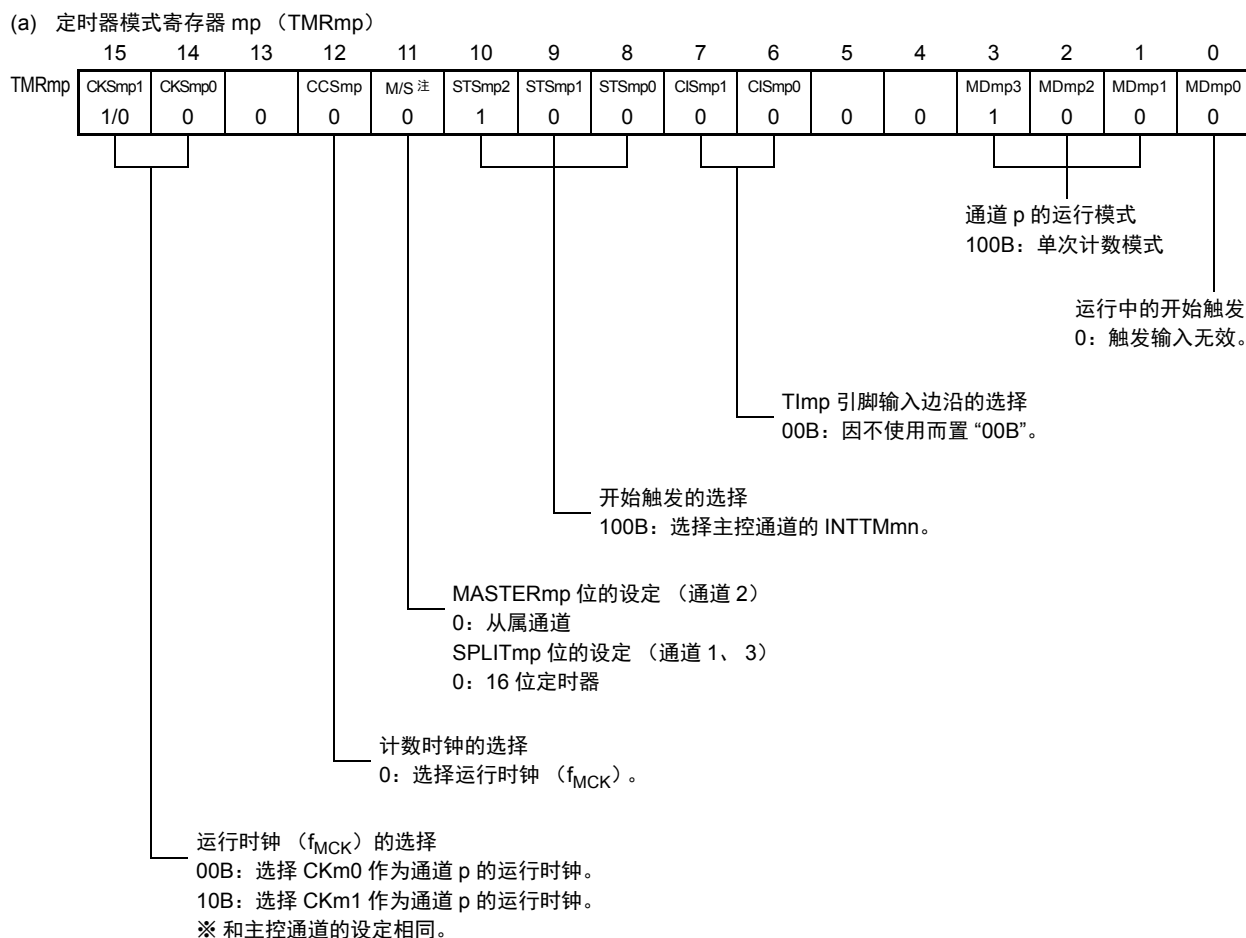
(e) 定时器输出模式寄存器 m (TOMm)



注 TMRm2 : MASTERmn=1
TMRm0 : 固定为 "0"。

备注 m: 单元号 (m=0) n: 主控通道号 (n=0、2)

图 6-68 单触发脉冲输出功能时（从属通道）的寄存器设定内容例子



(b) 定时器输出寄存器 m (TOM)

bit p	
TOM	TOMp
	1/0
	0: 由 TOMp 输出“0”。
	1: 由 TOMp 输出“1”。

(c) 定时器输出允许寄存器 m (TOEm)

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

(d) 定时器输出电平寄存器 m (TOLm)

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

(e) 定时器输出模式寄存器 m (TOMm)

bit p	
TOMm	TOMmp
	1
	1: 设定从属通道输出模式。

注 TMRm2 : MASTERmp 位
TMRm1、TMRm3 : SPLITmp 位

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

图 6-69 单触发脉冲输出功能时的操作步骤 (1/2)

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

图 6-69 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 TOEmp 位 (从属) 置 “1” (只限于重新开始运行)。 将定时器通道开始寄存器 m (TSm) 的 TSmn (主控) 和 TSmp (从属) 位同时置 “1”。 → 因为 TSmn 位和 TSmp 位是触发位, 所以自动返回到 “0”。 通过检测到主控通道的开始触发, 开始主控通道的计数。 - 检测 TImn 引脚输入的有效边沿。 - 通过软件将主控通道的 TSmn 位置 “1” 注。
	运行中	只能更改 TMRmn 寄存器的 CISmn1 位和 CISmn0 位的设定值。 禁止更改 TMRmp、TDRmn、TDRmp 寄存器以及 TOMmn 位、TOMmp 位、TOLmn 位和 TOLmp 位的设定值。 能随时读 TCRmn 寄存器和 TCRmp 寄存器。 不使用 TSRmn 寄存器和 TSRmp 寄存器。 能更改从属通道的 TOM 寄存器和 TOEm 寄存器的设定值。
	停止运行	将 TTmn 位 (主控) 和 TTmp 位 (从属) 同时置 “1”。 → 因为 TTmn 位和 TTmp 位是触发位, 所以自动返回到 “0”。 将从属通道的 TOEmp 位置 “0” 并且给 TOmp 位设定值。 →
	TAU 停止	要保持 TOmp 引脚输出电平的情况: 在给端口寄存器设定要保留的值后将 TOmp 位置 “0”。 → 不需要保持 TOmp 引脚输出电平的情况: 不需要设定。 将 PER0 寄存器的 TAUmEN 位置 “0”。 →
		TEmn 位和 TEmn 位都为 “1”, 主控通道进入开始触发 (检测 TImn 引脚输入的有效边沿或者将主控通道的 TSmn 位置 “1”) 的检测等待状态。计数器还处于停止状态。 主控通道开始计数。 如果主控通道通过检测开始触发 (检测 TImn 引脚输入的有效边沿或者将主控通道的 TSmn 位置 “1”), 就将 TDRmn 寄存器的值装入定时器计数寄存器 mn (TCRmn), 并且开始递减计数。如果 TCRmn 计数到 “0000H”, 就输出 INTTMmn, 并且在下一次 TImn 引脚输入前停止计数。 从属通道以主控通道的 INTTMmn 作为触发, 将 TDRmp 寄存器的值装入 TCRmp 寄存器并且开始递减计数。在从主控通道输出 INTTMmn 并且经过 1 个计数时钟后, 将 TOmp 的输出电平置为有效电平。然后, 如果 TCRmp 计数到 “0000H”, 就在将 TOmp 的输出电平置为无效电平后停止计数。 此后, 重复此运行。 TEmn 位和 TEmn 位都变为 “0” 并且停止计数。TCRmn 寄存器和 TCRmp 寄存器保持计数值而停止计数。TOmp 输出不被初始化而保持状态。 TOmp 引脚输出 TOmp 设定的电平。 通过端口功能保持 TOmp 引脚的输出电平。 定时器单元 m 的输入时钟处于停止提供状态。对全部电路和各通道的 SFR 进行初始化。(TOmp 位变为 “0” 并且 TOmp 引脚变为端口功能)

注 不能将从属通道的 TSmn 位置 “1”。

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

6.9.2 作为 PWM 功能的运行

将 2 个通道成对使用，能生成任意周期和占空比的脉冲。

输出脉冲的周期和占空比能用以下计算式进行计算：

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

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

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

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

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

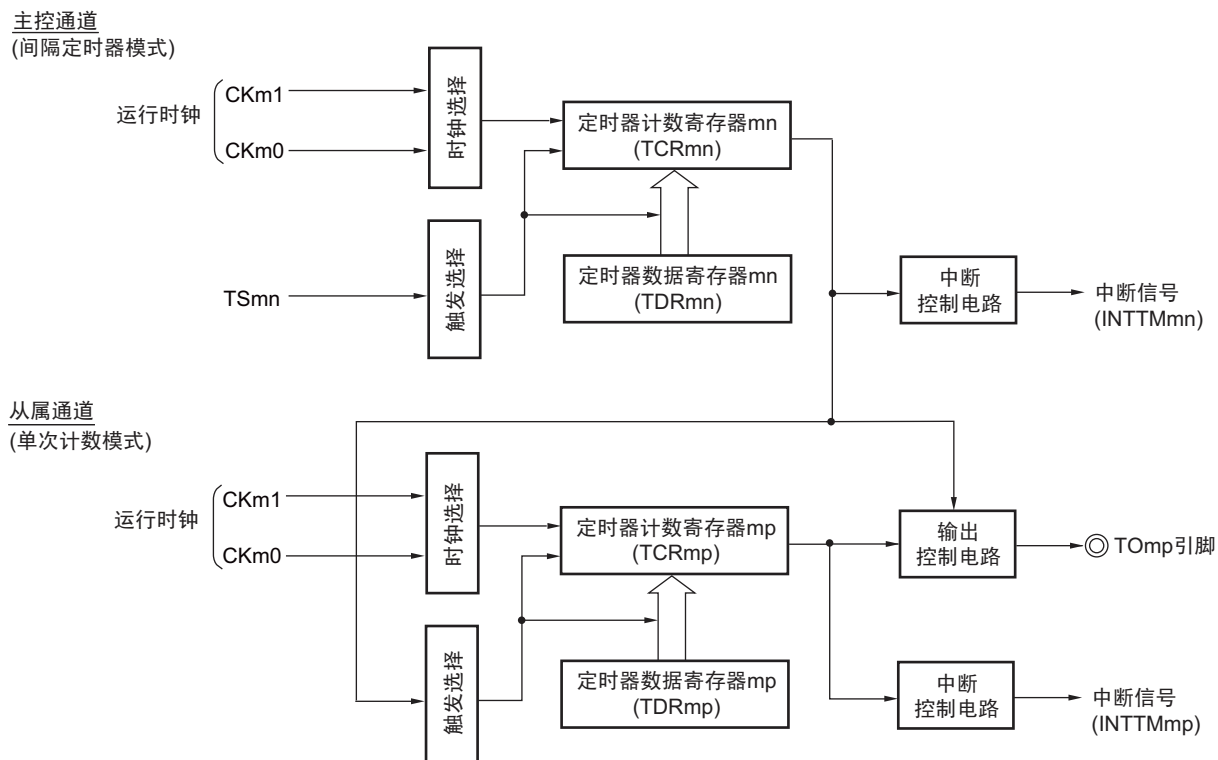
当用作 PWM 功能时，从属通道进行递减计数，在计数到“0000H”为止的期间为 PWM 输出 (TOmp) 的占空比。

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

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

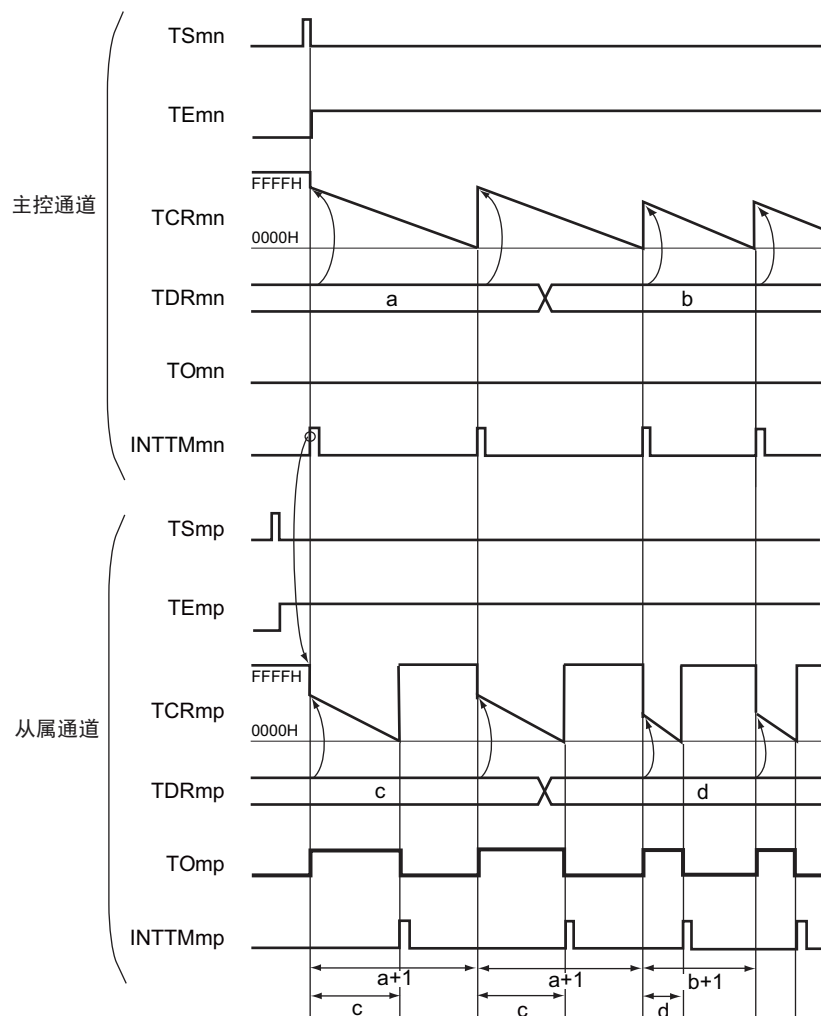
备注 m: 单元号 (m=0) n: 主控通道号 (n=0、2) p: 从属通道号 (n=0: p=1、2、3, n=2: p=3)

图 6-70 作为 PWM 功能运行的框图



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

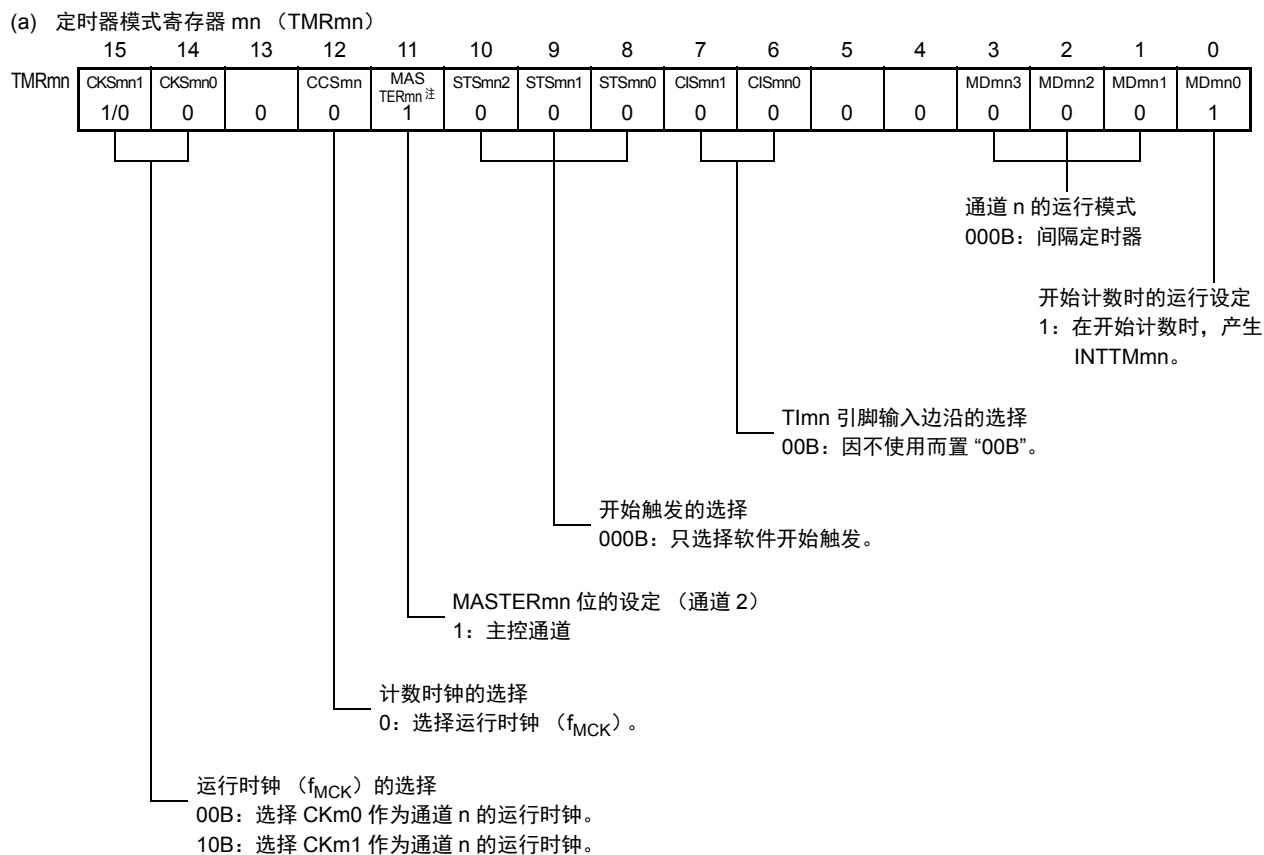
图 6-71 作为 PWM 功能的运行基本时序例子



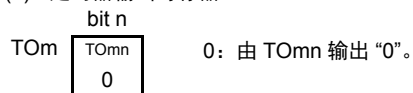
备注 1. m: 单元号 (m=0) n: 主控通道号 (n=0、2) p: 从属通道号 (n=0: p=1、2、3, n=2: p=3)

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

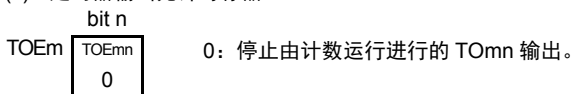
图 6-72 PWM 功能时（主控通道）的寄存器设定内容例子



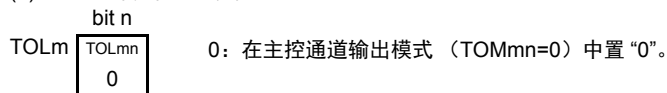
(b) 定时器输出寄存器 m (TOM)



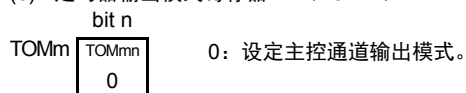
(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



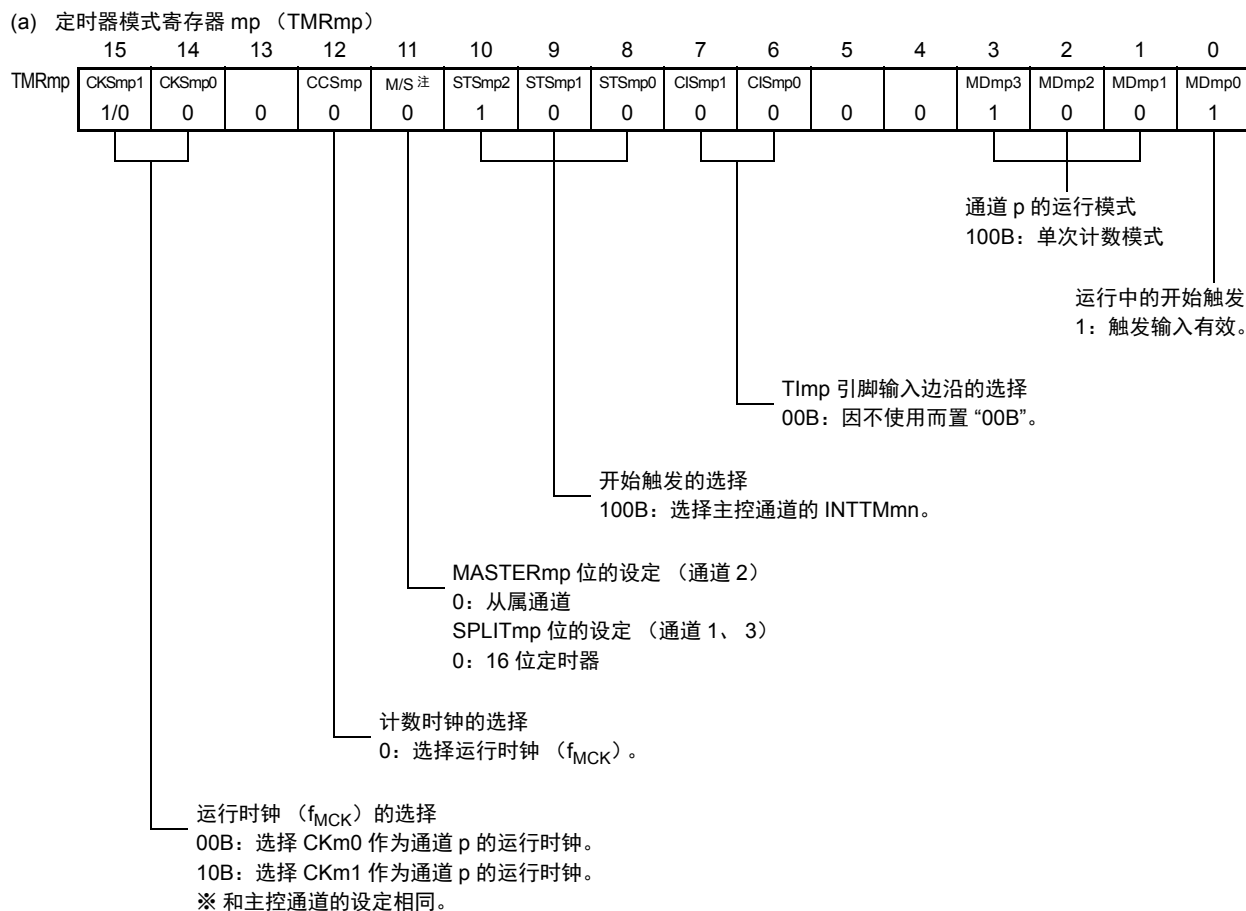
(e) 定时器输出模式寄存器 m (TOMm)



注 TMRm2 : MASTERmn=1
TMRm0 : 固定为“0”。

备注 m: 单元号 (m=0) n: 主控通道号 (n=0、2)

图 6-73 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 m (TOM)

bit p	
TOM	TOMP
	1/0
0: 由 TOMP 输出“0”。	
1: 由 TOMP 输出“1”。	

(c) 定时器输出允许寄存器 m (TOEm)

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

(d) 定时器输出电平寄存器 m (TOLm)

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

(e) 定时器输出模式寄存器 m (TOMm)

bit p	
TOMm	TOMmp
	1
1: 设定从属通道输出模式。	

注 TMRm2 : MASTERmp 位
TMRm1、TMRm3 : SPLITmp 位

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

图 6-74 PWM 功能时的操作步骤 (1/2)

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

图 6-74 PWM 功能时的操作步骤 (2/2)

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

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

6.9.3 作为多重 PWM 输出功能的运行

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

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

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

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

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

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

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

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

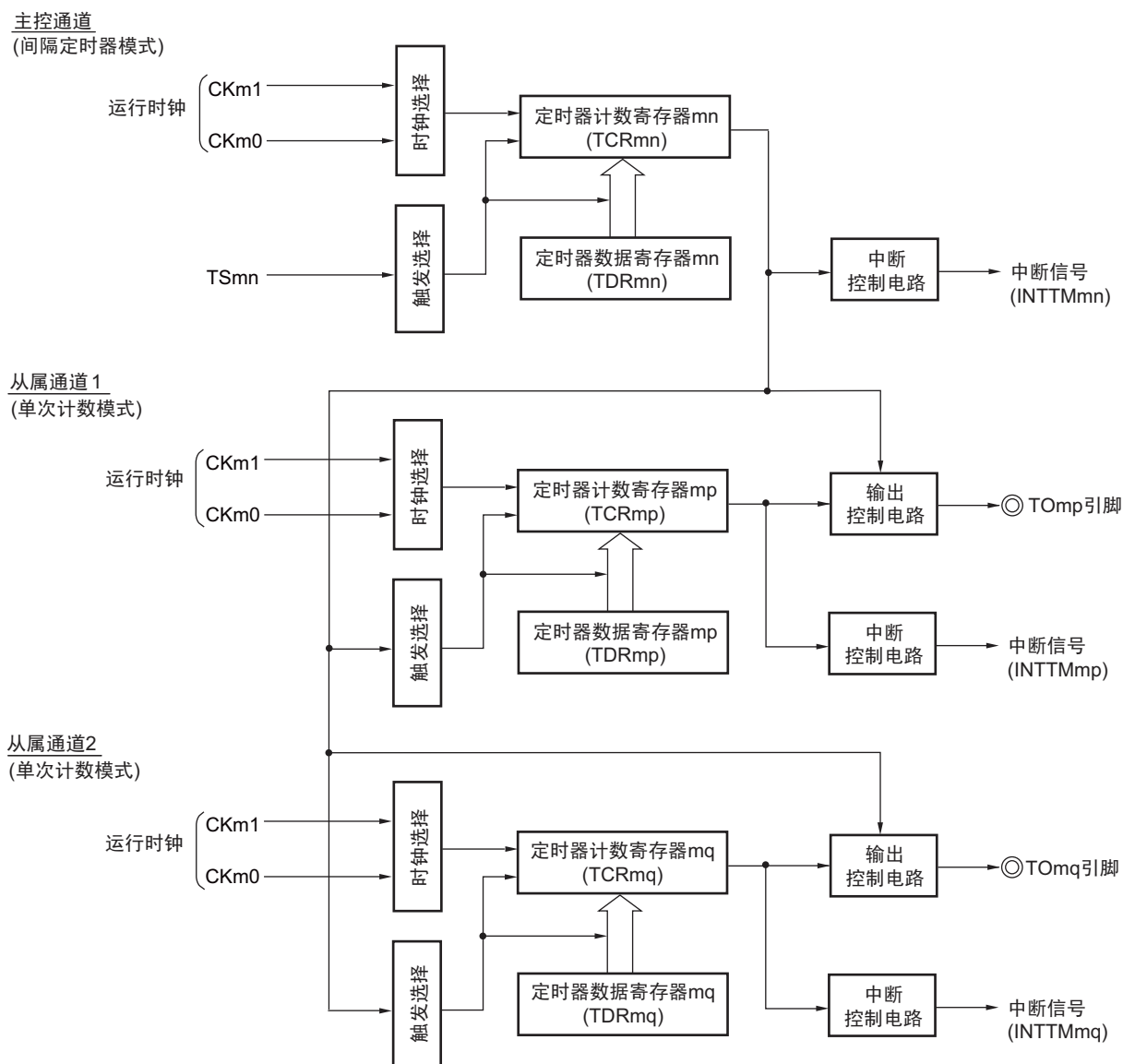
注意 要同时改写主控通道的定时器数据寄存器 mn (TDRmn) 和从属通道 1 的 TDRmp 寄存器时，至少需要 2 次写存取。因为在主控通道产生 INTTMmn 时将 TDRmn 寄存器和 TDRmp 寄存器的值装入 TCRmn 寄存器和 TCRmp 寄存器，所以如果分别在主控通道产生 INTTMmn 前和产生后进行改写，TOmp 引脚就不能输出期待的波形。因此，要同时改写主控的 TDRmn 寄存器和从属的 TDRmp 寄存器时，必须在主控通道产生 INTTMmn 后立即改写这 2 个寄存器（同样也适用于从属通道 2 的 TDRmq 寄存器）。

备注 m: 单元号 (m=0) n: 主控通道号 (n=0)

p: 从属通道号 q: 从属通道号

$n < p < q \leq 3$ (p 和 q 是大于 n 的整数)

图 6-75 作为多重 PWM 输出功能运行的框图（输出 2 种 PWM 的情况）

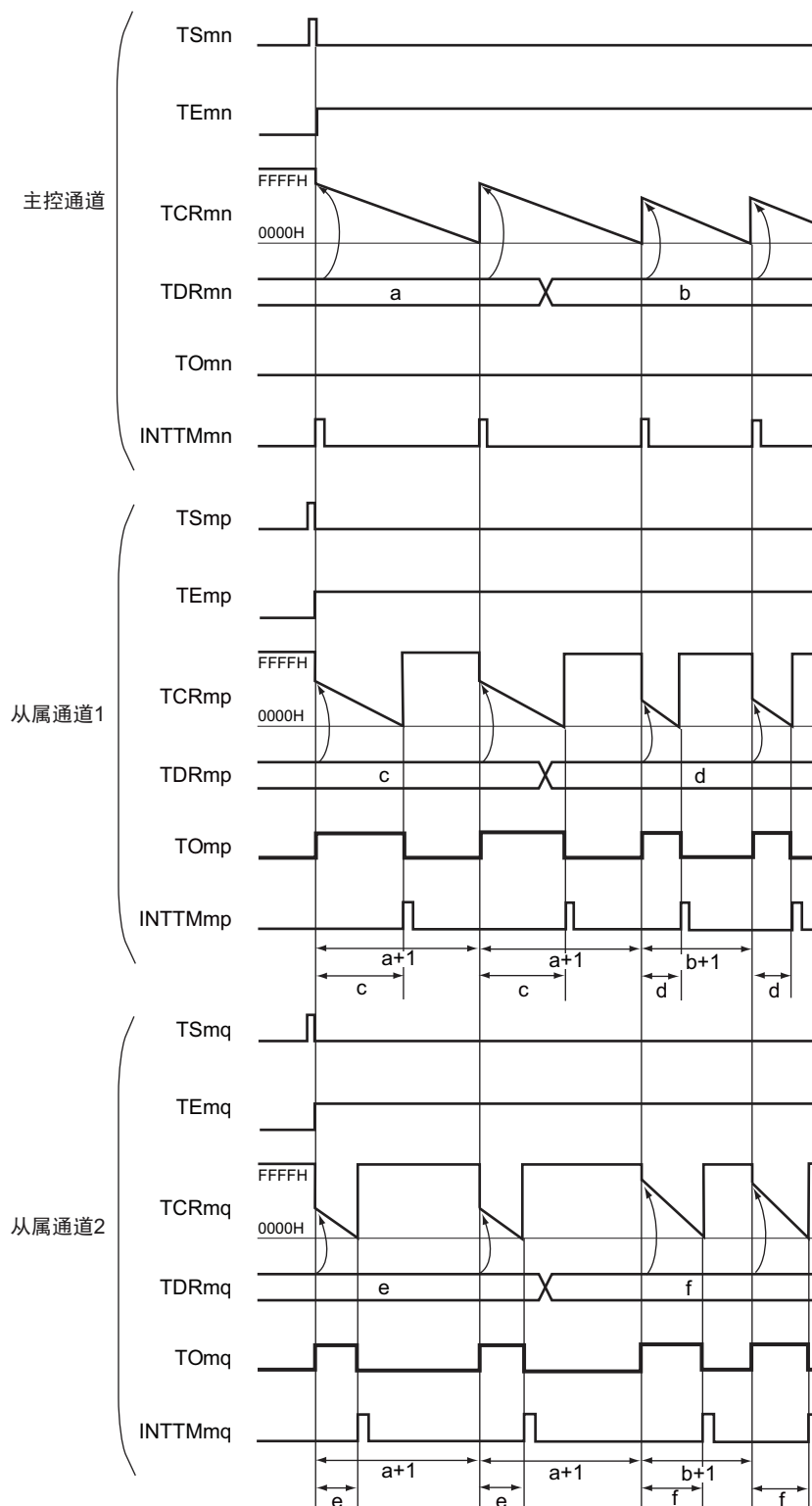


备注 m: 单元号 (m=0) n: 主控通道号 (n=0)

p: 从属通道号 q: 从属通道号

$n < p < q \leq 3$ (p和q是大于n的整数)

图 6-76 作为多重 PWM 输出功能的运行基本时序例子（输出 2 种 PWM 的情况）



备注 1. m: 单元号 (m=0) n: 主控通道号 (n=0)

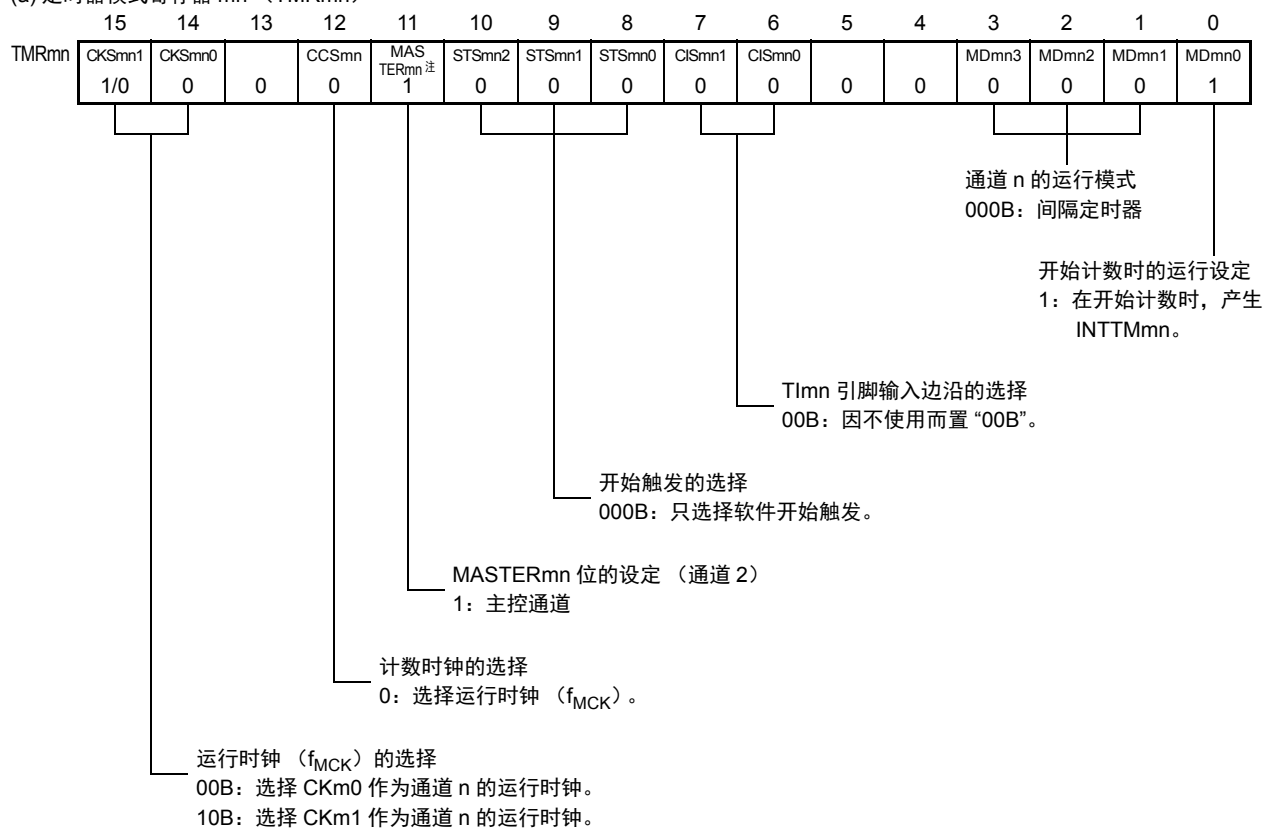
p: 从属通道号 q: 从属通道号

$n < p < q \leq 3$ (p 和 q 是大于 n 的整数)

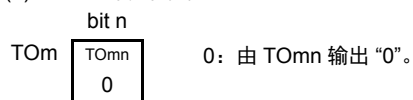
- 2. TSmn、TSmp、TSmq : 定时器通道开始寄存器 m (TSm) 的 bit n、p、q
- TEmn、TEmp、TEmq : 定时器通道允许状态寄存器 m (TEm) 的 bit n、p、q
- TCRmn、TCRmp、TCRmq : 定时器计数寄存器 mn、mp、mq (TCRmn、TCRmp、TCRmq)
- TDRmn、TDRmp、TDRmq : 定时器数据寄存器 mn、mp、mq (TDRmn、TDRmp、TDRmq)
- TOmn、TOmp、TOmq : TOmn、TOmp、TOmq 引脚的输出信号

图 6-77 多重 PWM 输出功能时（主控通道）的寄存器设定内容例子

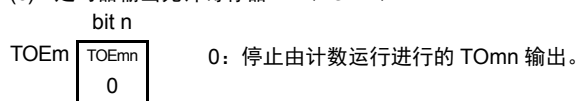
(a) 定时器模式寄存器 mn (TMRmn)



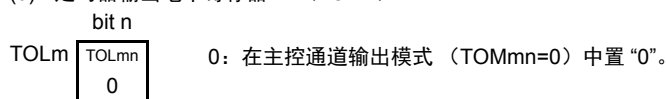
(b) 定时器输出寄存器 m (TOM)



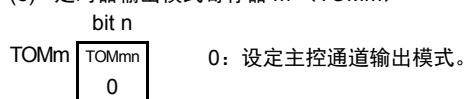
(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



(e) 定时器输出模式寄存器 m (TOMm)

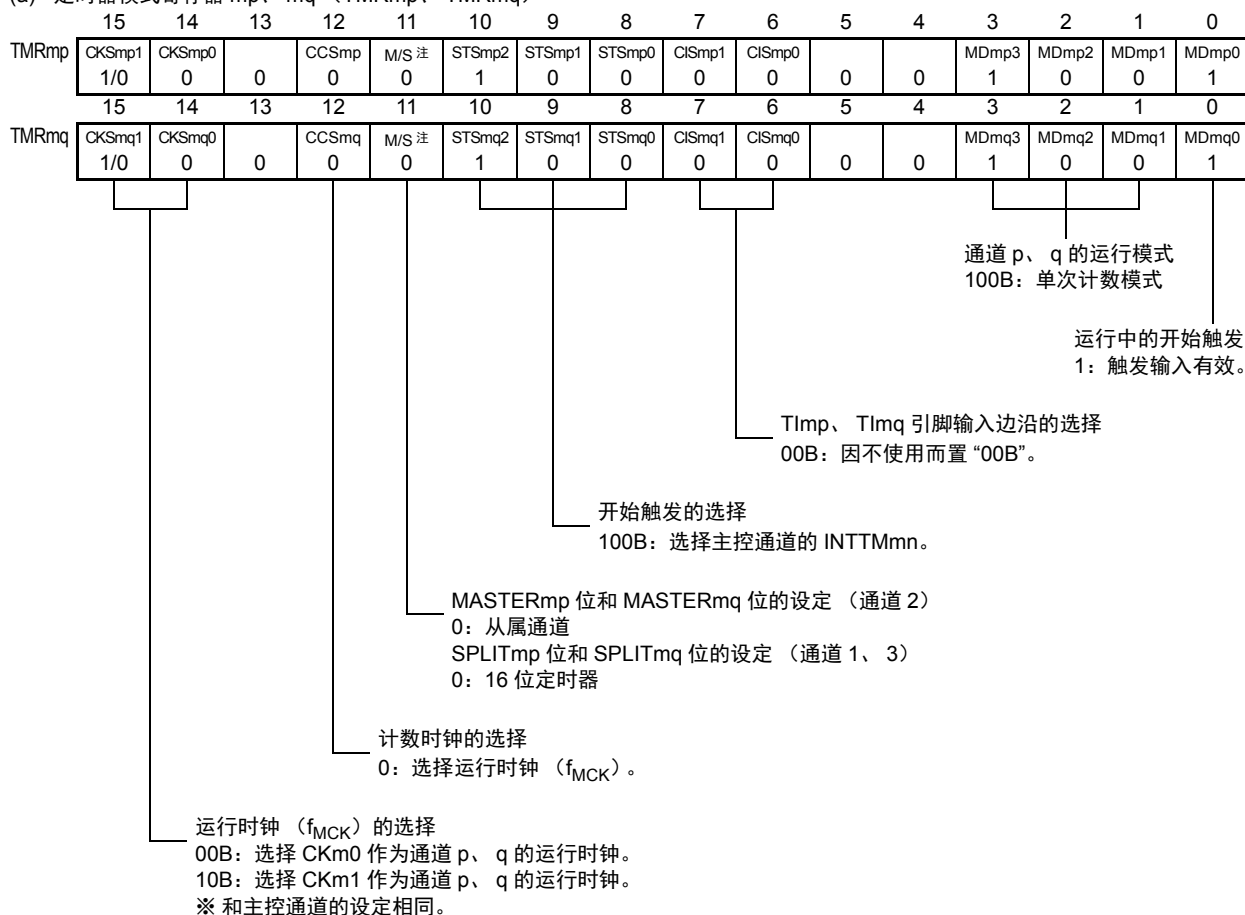


注 TMRm2 : MASTERmn=1
TMRm0 : 固定为“0”。

备注 m: 单元号 (m=0) n: 主控通道号 (n=0)

图 6-78 多重 PWM 输出功能时（从属通道）的寄存器设定内容例子（输出 2 种 PWM 的情况）

(a) 定时器模式寄存器 mp、mq（TMRmp、TMRmq）



(b) 定时器输出寄存器 m（TOM）

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

(c) 定时器输出允许寄存器 m（TOEm）

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

(d) 定时器输出电平寄存器 m（TOLm）

	bit q	bit p	
TOLm	TOLmq	TOLmp	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 m（TOMm）

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

注 TMRm2 : MASTERmp 位、MASTERmq 位
TMRm1、TMRm3 : SPLITmp 位、SPLITmq 位

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

图 6-79 多重 PWM 输出功能时的操作步骤（输出 2 种 PWM 的情况）(1/2)

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

图 6-79 多重 PWM 输出功能时的操作步骤 (输出 2 种 PWM 的情况) (2/2)

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

备注 m: 单元号 (m=0) n: 主控通道号 (n=0)

p: 从属通道号 q: 从属通道号

n < p < q ≤ 3 (p 和 q 是大于 n 的整数)

6.10 使用定时器阵列单元时的注意事项

6.10.1 使用定时器输出时的注意事项

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

详细内容请参照“4.5 使用复用功能时的端口相关寄存器的设定”。

第 7 章 定时器 RJ

7.1 定时器 RJ 的功能

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

16 位定时器由重加载寄存器和递减计数器构成，重加载寄存器和递减计数器分配在相同的地址。如果存取 TRJ0 寄存器，就能存取重加载寄存器和计数器。

定时器 RJ 的规格和框图分别如表 7-1 和图 7-1 所示。

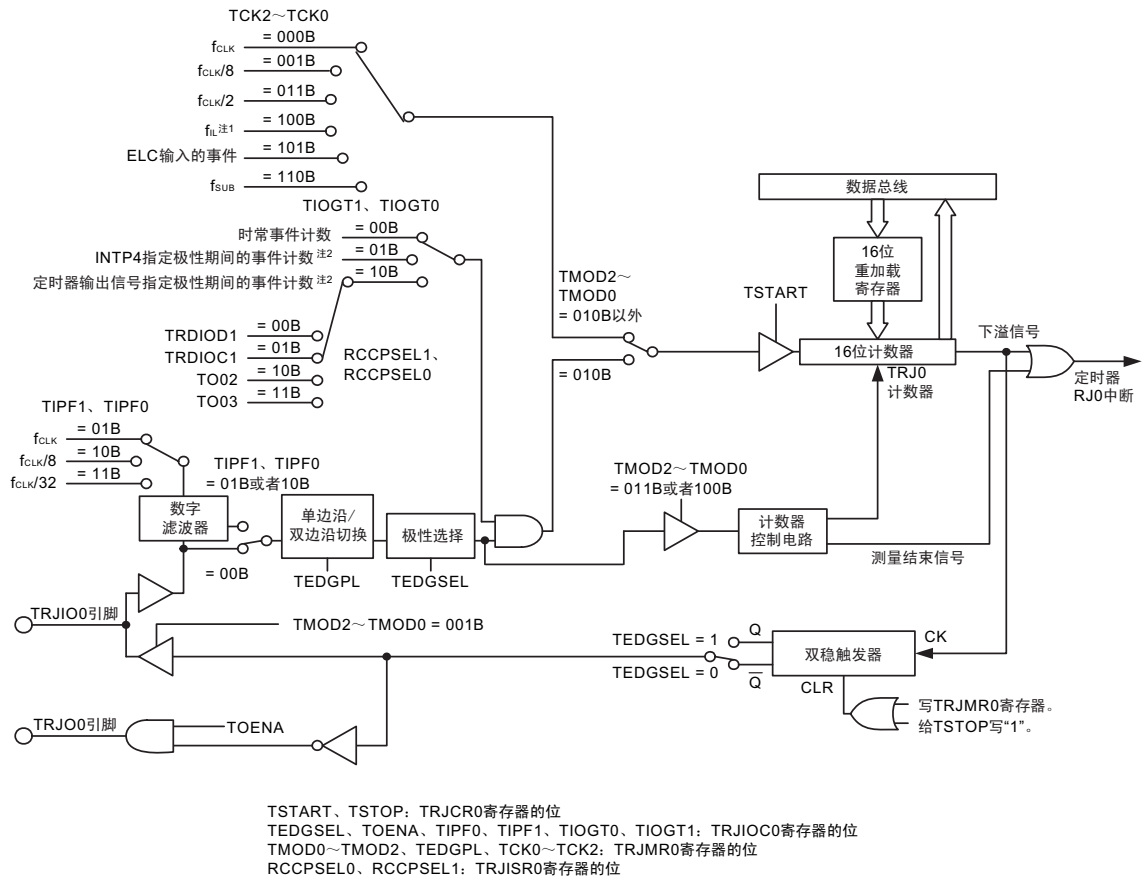
表 7-1 定时器 RJ 的规格

项目		内容
运行模式	定时器模式	对计数源进行计数。
	脉冲输出模式	对计数源进行计数，在定时器发生下溢时输出极性相反的脉冲。
	事件计数器模式	对外部事件进行计数。 也能在 STOP 模式中运行。
	脉宽测量模式	测量外部输入的脉冲宽度。
	脉冲周期测量模式	测量外部输入的脉冲周期。
计数源（运行时钟）		可选择 f_{CLK} 、 $f_{CLK}/2$ 、 $f_{CLK}/8$ 、 f_{IL} 、 f_{SUB} 或者事件链接控制器（ELC）输入的事件。
中断		- 当计数器发生下溢时 - 在脉宽测量模式中外部输入（TRJIO0）的有效宽度测量结束时 - 在脉冲周期测量模式中外部输入（TRJIO0）的设定边沿时
选择功能		与事件链接控制器（ELC）的协作 可选择 ELC 输入的事件作为计数源。

7.2 定时器 RJ 的结构

定时器 RJ 的框图和引脚结构分别如图 7-1 和表 7-2 所示。

图 7-1 定时器 RJ 的框图



- 注 1. 要选择 f_{IL} 作为计数源时，必须将副系统时钟提供模式控制寄存器（OSMC）的 WUTMMCK0 位置“1”。但是，当选择 f_{SUB} 作为实时时钟或者 12 位间隔定时器的计数源时，不能选择 f_{IL} 作为定时器 RJ 的计数源。
2. 能通过 TRJISR0 寄存器的 RCCPSEL2 位选择极性。

表 7-2 定时器 RJ 的引脚结构

引脚名	输入 / 输出	功能
INTP4	输入	定时器 RJ 的事件计数器模式控制
TRJIO0 注	输入 / 输出	定时器 RJ 的外部事件输入和脉冲输出
TRJO0 注	输出	定时器 RJ 的脉冲输出

- 注 能通过 PIOR1 寄存器的 PIOR12 位和 PIOR13 位选择 TRJO0 引脚的配置，并且能通过 PIOR1 寄存器的 PIOR10 位和 PIOR11 位选择 TRJIO0 引脚的配置。详细内容请参照“第 4 章 端口功能”。

7.3 控制定时器 RJ 的寄存器

控制定时器 RJ 的寄存器如表 7-3 所示。

表 7-3 控制定时器 RJ 的寄存器

寄存器名	符号
外围 I/O 重定向寄存器 1	PIOR1
外围允许寄存器 1	PER1
副系统时钟提供模式控制寄存器	OSMC
定时器 RJ 计数寄存器 0 注	TRJ0
定时器 RJ 控制寄存器 0	TRJCR0
定时器 RJ I/O 控制寄存器 0	TRJIOC0
定时器 RJ 模式寄存器 0	TRJMR0
定时器 RJ 事件引脚选择寄存器 0	TRJISR0
端口寄存器 0	P0
端口寄存器 3	P3
端口寄存器 4	P4
端口寄存器 5	P5
端口模式寄存器 0	PM0
端口模式寄存器 3	PM3
端口模式寄存器 4	PM4
端口模式寄存器 5	PM5

注 当存取 TRJ0 寄存器时，CPU 不进入下一条指令的处理而处于 CPU 处理的等待状态。因此，当发生此等待时，指令执行的时钟数增加等待的时钟数。存取 TRJ0 寄存器时的读写等待时钟数都为 1 个时钟。

7.3.1 外围允许寄存器 1（PER1）

PER1 寄存器是设定允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用定时器 RJ 时，必须将 bit0（TRJ0EN）置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER1 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 7-2 外围允许寄存器 1（PER1）的格式

地址: F007AH

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
PER1	0	0	0	TRD0EN	DTCEN	0	0	TRJ0EN

TRJ0EN	提供定时器 RJ0 的输入时钟的控制
0	停止提供输入时钟。 - 不能写定时器 RJ0 使用的 SFR。 - 定时器 RJ0 处于复位状态。
1	提供输入时钟。 能读写定时器 RJ0 使用的 SFR。

- 注意 1. 要设定定时器 RJ 时，必须先将 TRJ0EN 位置“1”。当 TRJ0EN 位为“0”时，忽视定时器 RJ 的控制寄存器的写操作，而且读取值都为初始值（端口模式寄存器 0、3、4、5（PM0、PM3、PM4、PM5）和端口寄存器 0、3、4、5（P0、P3、P4、P5）除外）。
2. 必须将以下的位置“0”。
- bit1、2、5～7

7.3.2 副系统时钟提供模式控制寄存器（OSMC）

能通过 WUTMMCK0 位选择定时器 RJ 的运行时钟。

RTCLPC 位是通过停止不需要的时钟功能来降低功耗的位。有关 RTCLPC 位的设定，请参照“第 5 章 时钟发生电路”。

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	RTCLPC	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	实时时钟、12 位间隔定时器的运行时钟和定时器 RJ 的运行时钟的选择
0	副系统时钟（f _{SUB} ） - 副系统时钟为实时时钟和 12 位间隔定时器的运行时钟。 - 不能选择低速内部振荡器作为定时器 RJ 的计数源。
1	低速内部振荡器时钟（f _{IL} ） - 低速内部振荡器时钟为实时时钟和 12 位间隔定时器的运行时钟。 - 能选择低速内部振荡器或者副系统时钟作为定时器 RJ 的计数源。

7.3.3 定时器 RJ 计数寄存器 0 (TRJ0)

这是 16 位寄存器。如果写此寄存器，就将数据写到重加载寄存器。如果读此寄存器，就读计数器的值。

重加载寄存器和计数器的状态因 TRJCR0 寄存器的 TSTART 位的值而变。详细内容请参照“7.4.1 重加载寄存器和计数器的改写”。

通过 16 位存储器操作指令设定 TRJ0 寄存器。

在产生复位信号后，TRJ0 寄存器的值变为“FFFFH”。

图 7-4 定时器 RJ 计数寄存器 0 (TRJ0) 的格式

地址: F0500H	复位后: FFFFH	R/W													
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1 0
TRJ0															

—	功能	设定范围
bit15 ~ 0	16 位计数器注 1、2	0000H ~ FFFFH

- 注 1. 如果给 TRJCR0 寄存器的 TSTOP 位写“1”，就强制停止 16 位计数器的计数，并且计数器的值变为“FFFFH”。
2. 如果 TRJMR0 寄存器的 TCK2 ~ TCK0 位的设定值不为“001B” ($f_{CLK}/8$) 和“011B” ($f_{CLK}/2$) 而 TRJ0 寄存器的值为“0000H”，就只在开始计数后立即向 DTC 和 ELC 产生 1 次请求信号。但是，TRJO0 和 TRJIO0 进行交替输出。

在事件计数器模式中，与 TCK2 ~ TCK0 位的值无关，如果 TRJ0 寄存器的值为“0000H”，就只在开始计数后立即向 DTC 和 ELC 产生 1 次请求信号，并且即使不在计数指定周期，TRJO0 也进行交替输出。

如果 TRJ0 寄存器的值大于等于“0001H”，就在 TRJ 每次发生下溢时产生请求信号。

注意 当存取 TRJ0 寄存器时，CPU 不进入下一条指令的处理而处于 CPU 处理的等待状态。因此，当发生此等待时，指令执行的时钟数增加等待的时钟数。存取 TRJ0 寄存器时的读写等待时钟数都为 1 个时钟。

7.3.4 定时器 RJ 控制寄存器 0 (TRJCR0)

TRJCR0 寄存器是控制寄存器 RJ 的计数和停止以及表示定时器 RJ 状态的寄存器。

通过 8 位存储器操作指令设定 TRJCR0 寄存器。

在产生复位信号后，TRJCR0 寄存器的值变为“00H”。

图 7-5 定时器 RJ 控制寄存器 0 (TRJCR0) 的格式

地址: F0240H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TRJCR0	0	0	TUNDF	TEDGF	0	TSTOP	TCSTF	TSTART

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

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

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

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

TSTART	定时器 RJ 的计数开始注 2
0	停止计数。
1	开始计数。
通过给 TSTART 位写“1”开始计数；通过给 TSTART 位写“0”停止计数。如果将 TSTART 位置“1” (开始计数)，TCSTF 位就与计数源同步变为“1” (正在计数)。另外，在给 TSTART 位写“0”后，TCSTF 位与计数源同步变为“0” (停止计数)。详细内容请参照“7.5.1 计数的开始和停止控制”。	

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

2. 有关使用 TSTART 位和 TCSTF 位时的注意事项，请参照“7.5.1 计数的开始和停止控制”。

7.3.5 定时器 RJ I/O 控制寄存器 0（TRJIOC0）

TRJIOC0 寄存器是设定寄存器 RJ 的输入 / 输出的寄存器。
通过 1 位或者 8 位存储器操作指令设定 TRJIOC0 寄存器。
在产生复位信号后，TRJIOC0 寄存器的值变为“00H”。

图 7-6 定时器 RJ I/O 控制寄存器 0（TRJIOC0）的格式

地址: F0241H

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
TRJIOC0	TIOGT1	TIOGT0	TIPF1	TIPF0	0	TOENA	0	TEDGSEL

TIOGT1	TIOGT0	TRJIO 的计数控制 ^{注 1、2}
0	0	总是对事件进行计数。
0	1	在 INTP4 指定的极性期间对事件进行计数。
1	0	在定时器输出信号指定的极性期间对事件进行计数。
上述以外		禁止设定。

TIPF1	TIPF0	TRJIO 输入滤波器的选择
0	0	没有滤波器。
0	1	有滤波器，通过 f_{CLK} 进行采样。
1	0	有滤波器，通过 $f_{\text{CLK}}/8$ 进行采样。
1	1	有滤波器，通过 $f_{\text{CLK}}/32$ 进行采样。

这些位指定 TRJIO 输入滤波器的采样频率。对 TRJIO0 引脚的输入进行采样，如果采样值连续 3 次相同，就确定此值为输入值。

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

TEDGSEL	输入 / 输出的极性切换

功能因运行模式而不同（参照表 7-4 和表 7-5）。

注 1. 当使用 INTP4 或者定时器输出信号时，能通过 TRJISR0 寄存器的 RCCPSEL2 位选择事件的计数极性。
2. TIOGT0 位和 TIOGT1 位只在事件计数器模式中有效。

表 7-4 TRJIO 输入 / 输出的边沿和极性切换

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

表 7-5 TRJO 输出的极性切换

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

7.3.6 定时器 RJ 控制寄存器 0 (TRJMR0)

TRJMR0 寄存器是设定寄存器 RJ 的运行模式的寄存器。

通过 1 位或者 8 位存储器操作指令设定 TRJMR0 寄存器。

在产生复位信号后，TRJMR0 寄存器的值变为“00H”。

图 7-7 定时器 RJ 控制寄存器 0 (TRJMR0) 的格式

地址: F0242H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TRJMR0	0	TCK2	TCK1	TCK0	TEDGPL	TMOD2	TMOD1	TMOD0

TCK2	TCK1	TCK0	定时器 RJ 的计数源选择注 1、2
0	0	0	f_{CLK}
0	0	1	$f_{CLK}/8$
0	1	1	$f_{CLK}/2$
1	0	0	f_{IL} 注 4
1	0	1	ELC 输入的事件
1	1	0	f_{SUB}
上述以外			禁止设定。

TEDGPL	TRJIO 边沿极性的选择注 5
0	单边沿
1	双边沿

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

- 注 1. 如果选择事件计数器模式，就选择外部输入 (TRJIO) 作为计数源，而与 TCK0 ~ TCK2 位的设定无关。
2. 不能在计数过程中切换计数源。如果要切换计数源，就必须在 TRJCR0 寄存器的 TSTART 位和 TCSTF 位都为“0” (停止计数) 时进行切换。
3. 只有在停止计数 (TRJCR0 寄存器的 TSTART 位和 TCSTF 位都为“0” (停止计数)) 时才能更改运行模式，不能在计数过程中进行更改。
4. 要选择 f_{IL} 作为计数源时，必须将副系统时钟提供模式控制寄存器 (OSMC) 的 WUTMMCK0 位置“1”。但是，当选择 f_{SUB} 作为实时时钟或者 12 位间隔定时器的计数源时，不能选择 f_{IL} 作为定时器 RJ 的计数源。
5. TEDGPL 位只在事件计数器模式中有效。
6. 通过写 TRJMR0 寄存器，对定时器 RJ 的 TRJIO0 引脚和 TRJIO1 引脚的输出进行初始化。有关初始化时的输出电平，请参照“图 7-6 定时器 RJ I/O 控制寄存器 0 (TRJIOC0) 的格式”的说明。

7.3.7 定时器 RJ 事件引脚选择寄存器 0（TRJISR0）

TRJISR0 寄存器是选择在事件计数器模式中控制事件计数期间的定时器以及设定极性的寄存器。
通过 1 位或者 8 位存储器操作指令设定 TRJISR0 寄存器。
在产生复位信号后，TRJISR0 寄存器的值变为“00H”。

图 7-8 定时器 RJ 事件引脚选择寄存器 0（TRJISR0）的格式

地址: F0243H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TRJISR0	0	0	0	0	0	RCCPSEL2 ^注	RCCPSEL1 ^注	RCCPSEL0 ^注

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

RCCPSEL1 ^注	RCCPSEL0 ^注	定时器输出信号的选择
0	0	TRDIOD1
0	1	TRDIOC1
1	0	TO02
1	1	TO03

注 RCCPSEL0 ~ RCCPSEL2 位只在事件计数器模式中有效。

7.3.8 端口模式寄存器 0、3、4、5（PM0、PM3、PM4、PM5）

这是以 1 位为单位设定端口 0、3、4、5 输入 / 输出的寄存器。

要将定时器输出引脚的复用端口（P01/TRJIO0、P30/TRJO0 等）用作定时器输出时，必须将各端口对应的端口模式寄存器（PM_{xx}）的位和端口寄存器（P_{xx}）的位置“0”。

（例）将 P01/TRJIO0 用作定时器输出的情况

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”。

要将定时器输入引脚的复用端口（P01/TRJIO0 等）用作定时器输入时，必须将各端口对应的端口模式寄存器（PM_{xx}）的位置“1”。此时，端口寄存器（P_{xx}）位可以是“0”或者“1”。

（例）将 P01/TRJIO0 用作定时器输入的情况

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”。

通过 1 位或者 8 位存储器操作指令设定 PM0、PM3、PM4、PM5 寄存器。

在产生复位信号后，这些寄存器的值变为“FFH”。

图 7-9 端口模式寄存器 0、3、4、5（PM0、PM3、PM4、PM5）的格式（64 引脚产品）

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

地址: FFF23H	复位后: FFH		R/W					
PM3	1	1	1	1	1	1	PM31	PM30

地址: FFF24H	复位后: FFH		R/W					
PM4	1	1	1	1	PM43	PM42	PM41	PM40

地址: FFF25H	复位后: FFH		R/W					
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50

PM _{mn}	P _{mn} 引脚的输入 / 输出模式的选择 (m=0、3、4、5, n=0 ~ 6)
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

备注 上述格式是 64 引脚产品的端口模式寄存器 0、3、4、5 的格式。有关其他产品的端口模式寄存器的格式，请参照“表 4-14 各产品分配的 PM_{xx}、P_{xx}、PU_{xx}、PIM_{xx}、POM_{xx}、PMC_{xx} 寄存器及其位”。

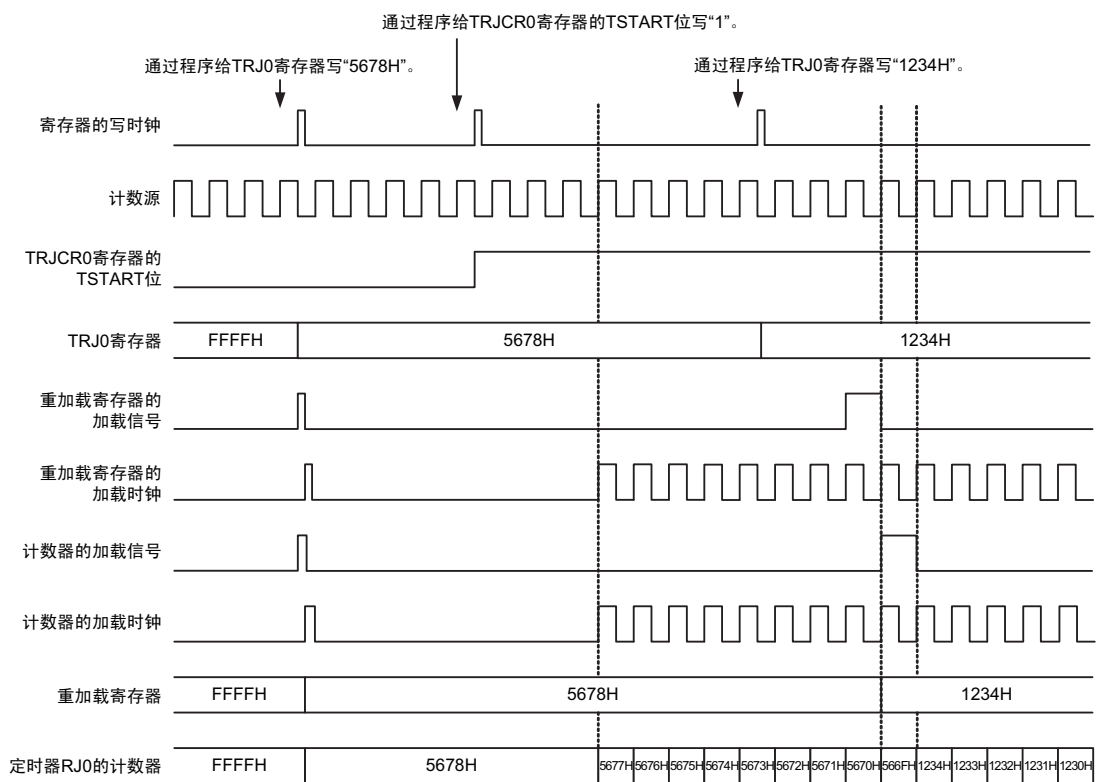
7.4 定时器 RJ 的运行

7.4.1 重加载寄存器和计数器的改写

与运行模式无关，重加载寄存器和计数器的改写时序因 TRJCR0 寄存器的 TSTART 位的值而变。当 TSTART 位为“0”（停止计数）时，直接写重加载寄存器和计数器；当 TSTART 位为“1”（开始计数）时，在与计数源同步写重加载寄存器后，与下一个计数源同步写计数器。

由 TSTART 位的值决定的改写时序图如图 7-10 所示。

图 7-10 由 TSTART 位的值决定的改写时序图



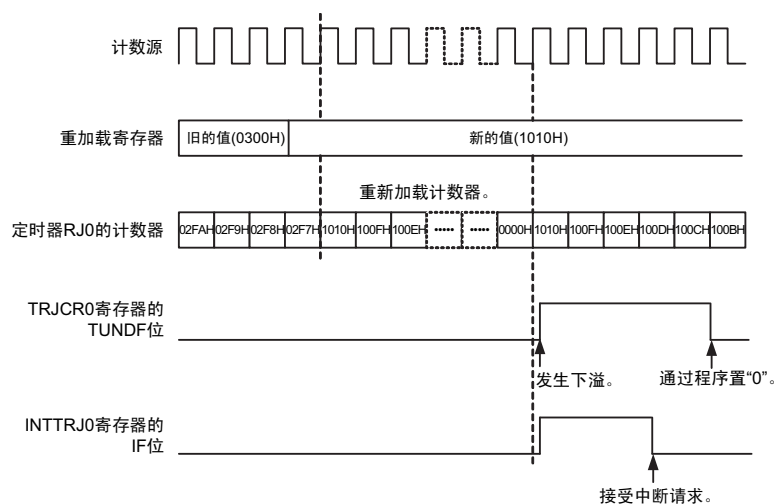
7.4.2 定时器模式

这是通过 TRJMR0 寄存器的 TCK0 ~ TCK2 位选择的计数源进行递减计数的模式。

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

定时器模式的运行例子如图 7-11 所示。

图 7-11 定时器模式的运行例子



7.4.3 脉冲输出模式

在此模式中，通过 TRJMR0 寄存器的 TCK0 ~ TCK2 位选择的计数源进行递减计数，每当发生下溢时，就将 TRJIO 引脚和 TRJO 引脚的输出电平反相输出。

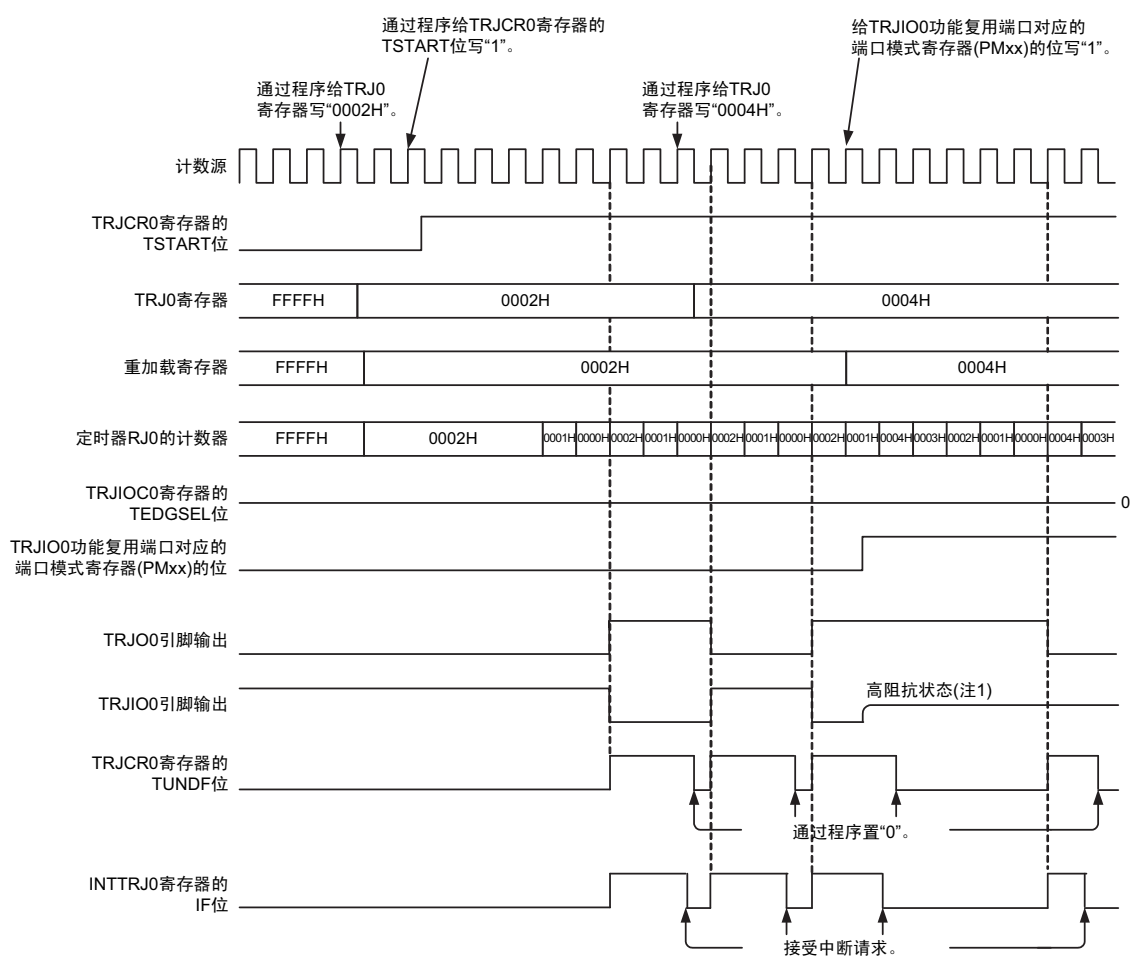
在脉冲输出模式中，每当输入计数源时计数值就减 1，如果计数值变为“0000H”并且输入下一个计数源，就发生下溢并且产生中断请求。

能从 TRJIO0 引脚和 TRJO0 引脚输出脉冲，并且每当发生下溢时就将输出电平进行反相。能通过 TRJIOC0 寄存器的 TOENA 位停止 TRJO0 引脚的脉冲输出。

另外，能通过 TRJIOC0 寄存器的 TEDGSEL 位选择输出电平。

脉冲输出模式的运行例子如图 7-12 所示。

图 7-12 脉冲输出模式的运行例子



注1. 通过被选择为TRJIO0功能的端口输出允许控制，变为高阻抗状态。

7.4.4 事件计数器模式

这是通过 TRJIO0 引脚输入的外部事件信号（计数源）进行递减计数的模式。

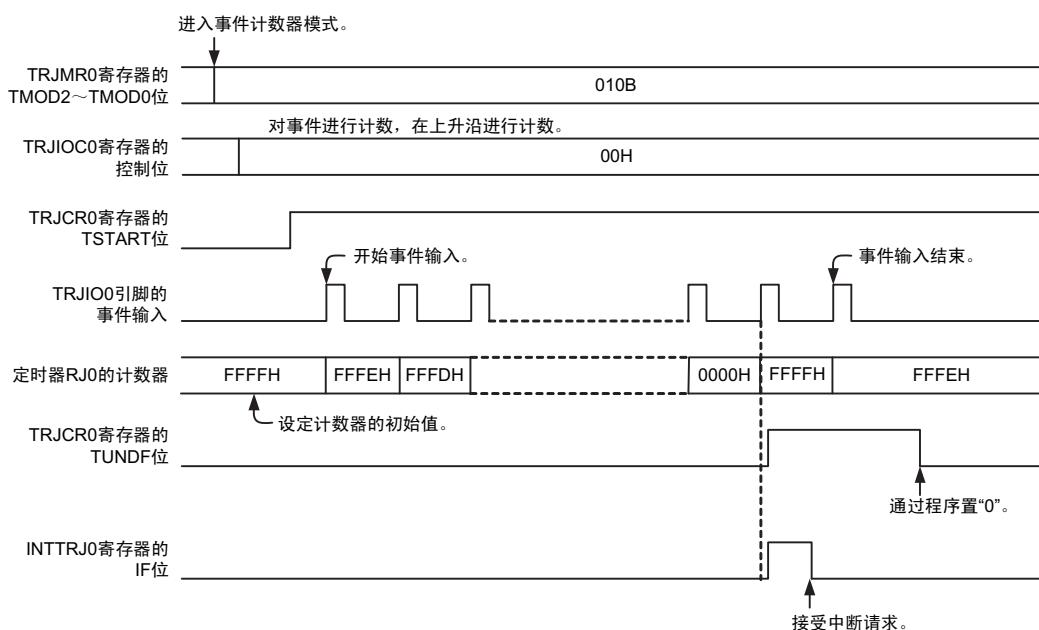
能通过 TRJIOC0 寄存器的 TIOGT0 ~ TIOGT1 位和 TRJISR0 寄存器进行事件计数期间的各种设定，并且能通过 TRJIOC0 寄存器的 TIFP0 ~ TIFP1 位指定 TRJIO0 输入的滤波器功能。

即使在事件计数器模式中 TRJO0 引脚也能进行交替输出。

要使用事件计数器模式时，请参照“7.5.5 TRJO0 引脚和 TRJIO0 引脚的设定步骤”。

事件计数器模式的运行例子 1 如图 7-13 所示。

图 7-13 事件计数器模式的运行例子 1

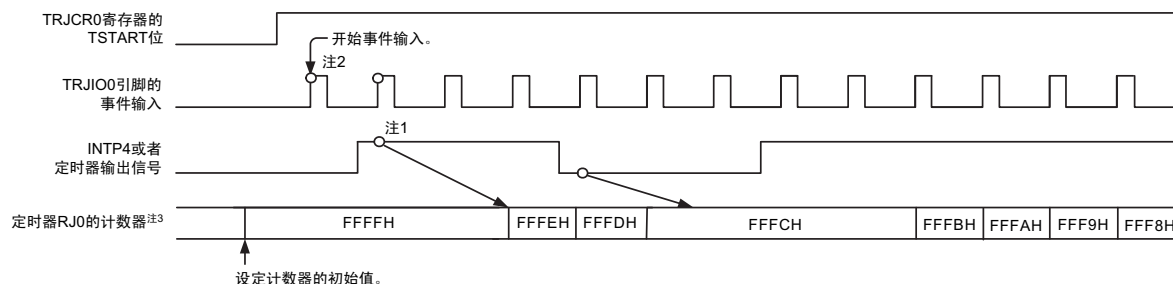


事件计数器模式中的指定时间计数（TRJIOC0 寄存器的 TIOGT1 位和 TIOGT0 位为“01B”或者“10B”）的运行例子如图 7-14 所示。

图 7-14 事件计数器模式的运行例子 2

■将运行模式设定为以下情况的时序例子

TRJMR0寄存器: TMOD2、1、0 = 010B(事件计数器模式)
 TRJIOC0寄存器: TIOGT1、0 = 01B(外部中断引脚指定期间的事件计数)
 TIFP1、0 = 00B(没有滤波器)
 TEDGSEL = 0(在上升沿进行计数)
 TRJISR0寄存器: RCCPSEL2 = 1(对“H”期间进行计数)



以下注意事项限于事件计数器模式的运行模式设定是TRJIOC0寄存器的TIOGT1位和TIOGT0位为“01B”或者“10B”的情况。

- 注1. 为了进行同步控制，可能在反映到计数运行前发生2个计数源时钟的延迟。
- 注2. 开始计数后的2个计数源时钟可能根据前一次计数停止前的状态进行计数。
 要使开始计数后的2个计数源时钟变为无效时，必须给TRJCR0寄存器的TSTOP位写“1”，对内部电路进行初始化并且在运行设定后开始计数。
- 注3. 对于由TRJISR0寄存器的RCCPSEL1位和RCCPSEL0位选择的定时器输出信号，不能将被分配了该定时器输出功能的引脚用作定时器以外的复用功能的输出。

7.4.5 脉宽测量模式

这是测量 TRJIO0 引脚输入的外部信号脉宽的模式。

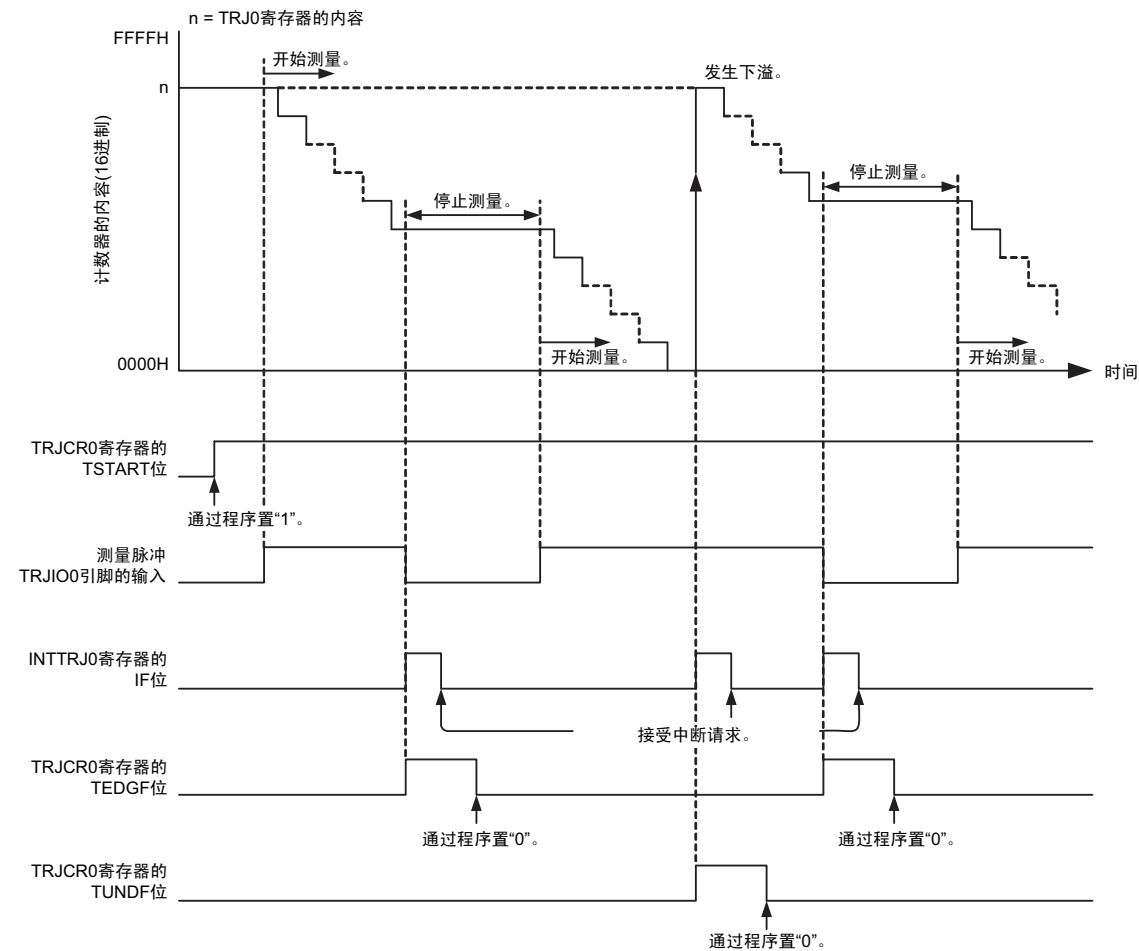
在脉宽测量模式中，如果给 TRJIO0 引脚输入 TRJIOC0 寄存器的 TEDGSEL 位指定的电平，就通过选择的计数源开始递减计数。如果 TRJIO0 引脚输入的指定电平结束，计数器就停止计数，TRJCR0 寄存器的 TEDGF 位变为“1”（有有效边沿）并且产生中断请求。通过在计数器停止计数时读计数值进行脉宽数据的测量。如果在测量过程中计数器发生下溢，TRJCR0 寄存器的 TUNDF 位就变为“1”（发生下溢）并且产生中断请求。

脉宽测量模式的运行例子如图 7-15 所示。

要存取 TRJCR0 寄存器的 TEDGF 位和 TUNDF 位时，请参照“7.5.2 标志的存取（TRJCR0 寄存器的 TEDGF 位和 TUNDF 位）”。

图 7-15 脉宽测量的运行例子

这是对测量脉冲的“H”电平进行测量的情况(TRJIOC0寄存器的TEDGSEL = 1)。



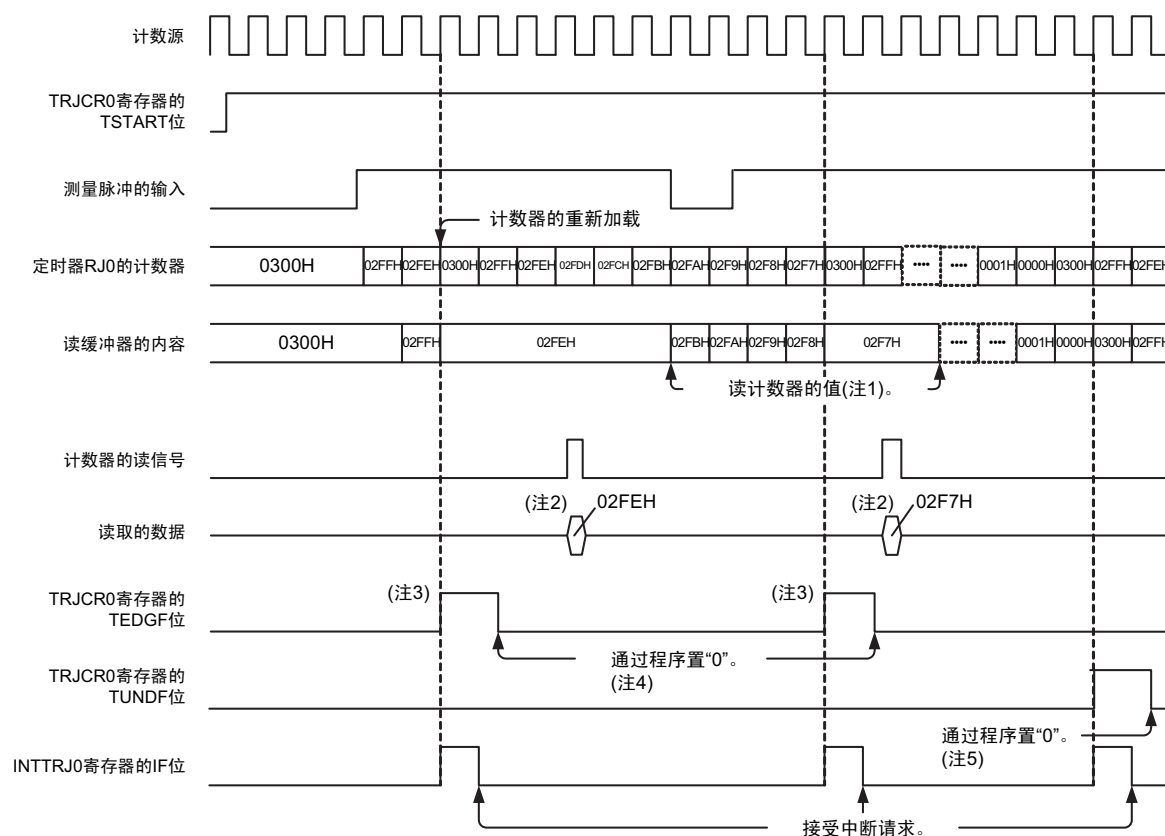
7.4.6 脉冲周期测量模式

这是测量 TRJIO0 引脚输入的外部信号脉冲周期的模式。

通过 TRJMR0 寄存器的 TCK0 ~ TCK2 位选择的计数源，计数器进行递减计数。如果给 TRJIO0 引脚输入 TRJIOC0 寄存器的 TEDGSEL 位指定期间的脉冲，就在计数源的上升沿将计数值传送到读缓冲器，并且在下一个上升沿将重加载寄存器的值加载到计数器，同时 TRJCR0 寄存器的 TEDGF 位变为“1”（有有效边沿）并且产生中断请求。此时，读 TRJ0 寄存器（读缓冲器），读取值和重加载值的差为输入脉冲的周期数据。周期数据被保持到读缓冲器为止。如果计数器发生下溢，TRJCR0 寄存器的 TUNDF 位就变为“1”（发生下溢）并且产生中断请求。脉冲周期测量模式的运行例子如图 7-16 所示。

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

图 7-16 脉冲周期测量模式的运行例子



这是 TRJ0 寄存器的初始值为“0300H”并且将 TRJIOC0 寄存器的 TEDGSEL 位置“0”以及在测量脉冲上升前进行测量的情况。

- 注1. 必须是在从 TEDGF 位变为“1”（有有效边沿）到输入下一个有效边沿为止的期间读 TRJ0 寄存器。读缓冲器的内容被保持到读 TRJ0 寄存器为止，因此，如果在输入有效边沿前不读 TRJ0 寄存器，就保持以前周期的测量结果。
2. 如果在脉冲周期测量模式中读 TRJ0 寄存器，读取值就为读缓冲器的内容。
3. 如果在输入测量脉冲的有效边沿后输入外部脉冲的设定边沿，TRJCR0 寄存器的 TEDGF 位就变为“1”（有有效边沿）。
4. 要通过程序置“0”时，必须通过 8 位存储器操作指令给 TRJCR0 寄存器的 TEDGF 位写“0”。
5. 要通过程序置“0”时，必须通过 8 位存储器操作指令给 TRJCR0 寄存器的 TUNDF 位写“0”。

7.4.7 与事件链接控制器（ELC）的协作

能通过与 ELC 的协作，将 ELC 输入的事件设定为计数源。

通过 TRJMR0 寄存器的 TCK0 ~ TCK2 位，在 ELC 输入的事件上升沿进行计数。但是，在事件计数器模式中 ELC 输入不起作用。

ELC 设定步骤如下所示。

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

7.4.8 各模式的输出设定

各模式中的 TRJO0 引脚和 TRJIO0 引脚的状态如表 7-6 和表 7-7 所示。

表 7-6 TRJO0 引脚的设定

运行模式	TRJIOC0 寄存器		TRJO0 引脚的输出
	TOENA 位	TEDGSEL 位	
全部模式	1	1	反相输出
		0	正相输出
	0	0 或者 1	禁止输出

表 7-7 TRJIO0 引脚的设定

运行模式	TRJIOC0 寄存器		TRJIO0 引脚的输入 / 输出
	PMXX 位注	TEDGSEL 位	
定时器模式	0 或者 1	0 或者 1	输入（不使用）
脉冲输出模式	1	0 或者 1	禁止输出（Hi-z 输出）
	0	1	正相输出
		0	反相输出
事件计数器模式	1	0 或者 1	输入
脉宽测量模式			
脉冲周期测量模式			

注 这是和 TRJIO0 功能复用端口对应的端口模式寄存器（PMxx）的位。

7.5 使用定时器 RJ 时的注意事项

7.5.1 计数的开始和停止控制

- 事件计数模式或者将计数源设定为非ELC的情况
如果在计数停止过程中给 TRJCR0 寄存器的 TSTART 位写“1”（开始计数），就在3个计数源周期内 TRJCR0 寄存器的 TCSTF 位为“0”（停止计数）。除了 TCSTF 位以外，不能在 TCSTF 位变为“1”（正在计数）前存取定时器 RJ 的相关寄存器注。
如果在计数过程中给 TSTART 位写“0”（停止计数），就在3个计数源周期内 TCSTF 位为“1”。在 TCSTF 位变为“0”时停止计数。除了 TCSTF 位以外，不能在 TCSTF 位变为“0”前存取定时器 RJ 的相关寄存器注。
必须在将 TSTART 位从“0”改为“1”前清除中断寄存器。详细内容请参照“第18章 中断功能”。

注 定时器 RJ 的相关寄存器：TRJ0、TRJCR0、TRJIOC0、TRJMR0、TRJISR0

- 事件计数模式或者将计数源设定为 ELC 的情况
如果在计数停止过程中给 TRJCR0 寄存器的 TSTART 位写“1”（开始计数），就在2个 CPU 时钟周期内 TRJCR0 寄存器的 TCSTF 位为“0”（停止计数）。除了 TCSTF 位以外，不能在 TCSTF 位变为“1”（正在计数）前存取定时器 RJ 的相关寄存器注。
如果在计数过程中给 TSTART 位写“0”（停止计数），就在2个 CPU 时钟周期内 TCSTF 位为“1”。在 TCSTF 位变为“0”时停止计数。除了 TCSTF 位以外，不能在 TCSTF 位变为“0”前存取定时器 RJ 的相关寄存器注。
必须在将 TATART 位从“0”改为“1”前清除中断寄存器。详细内容请参照“第18章 中断功能”。

注 定时器 RJ 的相关寄存器：TRJ0、TRJCR0、TRJIOC0、TRJMR0、TRJISR0

7.5.2 标志的存取（TRJCR0 寄存器的 TEDGF 位和 TUNDF 位）

如果通过程序给 TRJCR0 寄存器的 TEDGF 位和 TUNDF 位写“0”，这些位就变为“0”。但是，即使写“1”值也不变。如果对 TRJCR0 寄存器使用读 - 修改 - 写指令，就在指令执行过程中即使 TEDGF 位变为“1”（有有效边沿）并且 TUNDF 位变为“1”（发生下溢），也可能因时序而误将 TEDGF 位和 TUNDF 位置“0”。必须通过 8 位存储器操作指令存取 TRJCR0 寄存器。

7.5.3 计数寄存器的存取

在 TRJCR0 寄存器的 TSTART 位和 TCSTF 位都为“1”（正在计数）的情况下连续写 TRJ0 寄存器时，必须在各自的写操作之间至少间隔 3 个计数源时钟周期。

7.5.4 模式的变更

只有在停止计数（TRJCR0 寄存器的 TSTART 位和 TCSTF 位都为“0”（停止计数））时才能更改定时器 RJ 的运行模式相关寄存器（TRJIOC0、TRJMR0、TRJISR0），不能在计数过程中进行更改。

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

7.5.5 TRJO0 引脚和 TRJIO0 引脚的设定步骤

在复位后，TRJO0 引脚和 TRJIO0 引脚的复用 I/O 端口为输入端口。

要从 TRJO0 引脚和 TRJIO0 引脚输出时，必须按照以下步骤进行设定。

更改步骤

- (1) 设定模式。
- (2) 设定初始值，允许输出。
- (3) 将 TRJO0 引脚和 TRJIO0 引脚对应的端口寄存器的位置“0”。
- (4) 将 TRJO0 引脚和 TRJIO0 引脚对应的端口模式寄存器的位设定为输出模式。
(从 TRJO0 引脚和 TRJIO0 引脚开始输出)
- (5) 开始计数 (TRJCR0 寄存器的 TSTART=1)。

要从 TRJIO0 引脚输入时，必须按照以下步骤进行设定。

- (1) 设定模式。
- (2) 设定初始值，选择边沿。
- (3) 将 TRJIO0 引脚对应的端口模式寄存器的位设定为输入模式。
(从 TRJIO0 引脚开始输入)
- (4) 开始计数 (TRJMR0 寄存器的 TSTART=1)。
- (5) 等到 TRJCR0 寄存器的 TCSTF 位变为“1” (正在计数)。
(只限于事件计数器模式)
- (6) 从 TRJIO0 引脚输入外部事件。
- (7) 必须在第一次测量结束时进行测量值的无效处理 (第二次及以后的测量值有效)。
(只限于脉宽测量模式和脉冲周期测量模式)

7.5.6 不使用定时器 RJ 的情况

当不使用定时器 RJ 时，必须将 TRJMR0 寄存器的 TMOD2 ~ TMOD0 位置“000B” (定时器模式) 并且将 TRJIOC0 寄存器的 TOENA 位置“0” (禁止 TRJO 输出)。

7.5.7 定时器 RJ 运行时钟的停止

能通过 PER1 寄存器的 TRJ0EN 位控制定时器 RJ 时钟的提供或者停止。但是，不能在定时器 RJ 的时钟停止时存取以下的 SFR，而必须在提供定时器 RJ 时钟的状态下进行存取。

TRJO 寄存器、TRJCR0 寄存器、TRJMR0 寄存器、TRJIOC0 寄存器和 TRJISR0 寄存器

7.5.8 STOP 模式 (事件计数器模式) 的设定步骤

要在 STOP 模式中使事件计数器模式运行时，必须在提供定时器 RJ 的时钟后按照以下步骤转移到 STOP 模式。

设定步骤

- (1) 设定运行模式。
- (2) 开始计数 (TSTART=1、TCSTF=1)。
- (3) 停止提供定时器 RJ 的时钟。

要在 STOP 模式中停止事件计数器模式时，必须按照以下步骤进行运行停止处理。

- (1) 提供定时器 RJ 的时钟。
- (2) 停止计数 (TSTART=0、TCSTF=0)

7.5.9 STOP 模式中（只限于事件计数器模式）的功能限制

要在 STOP 模式中运行事件计数器模式时，不能使用数字滤波器功能。

7.5.10 通过 TSTOP 位进行强制的计数停止

不能在通过 TRJCR0 寄存器的 TSTOP 位强制停止计数器的计数后的 1 个计数源周期内存取以下的 SFR。
TRJ0 寄存器、TRJCR0 寄存器和 TRJMR0 寄存器

7.5.11 数字滤波器

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

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

7.5.12 选择 f_{IL} 作为计数源的情况

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

第 8 章 定时器 RD

8.1 定时器 RD 的功能

定时器 RD 有以下 4 种模式：

- 定时器模式
 - 输入捕捉功能 以外部信号为触发，将计数器的值取到寄存器。
 - 输出比较功能 检测计数器的值和寄存器的值是否相同（能在检测时更改引脚的输出）。
 - PWM 功能 连续输出任意的脉宽。

以下 3 种模式使用 PWM 功能：

- 复位同步 PWM 模式 这是输出锯齿波调制、无死区时间的三相波形（6 个）的模式。
- 互补 PWM 模式 这是输出三角波调制、有死区时间的三相波形（6 个）的模式。
- PWM3 模式 这是输出同周期 PWM 波形（2 个）的模式。

在定时器模式中，定时器 RD0 和定时器 RD1 具有同等的输入捕捉功能、输出比较功能和 PWM 功能，每个引脚能选择这些功能，并且能在定时器 RD0 和定时器 RD1 中组合使用这些功能。

复位同步 PWM 模式、互补 PWM 模式和 PWM3 模式通过定时器 RD0 和定时器 RD1 的计数器和寄存器的组合来输出波形，引脚功能取决于运行模式。

定时器 RD 有 4 个输入 / 输出引脚。

定时器 RD 的运行时钟为 f_{CLK} 或者 f_{HOCO} 。

8.2 定时器 RD 的结构

定时器 RD 的框图和引脚结构分别如图 8-1 和表 8-1 所示。

图 8-1 定时器 RD 的框图

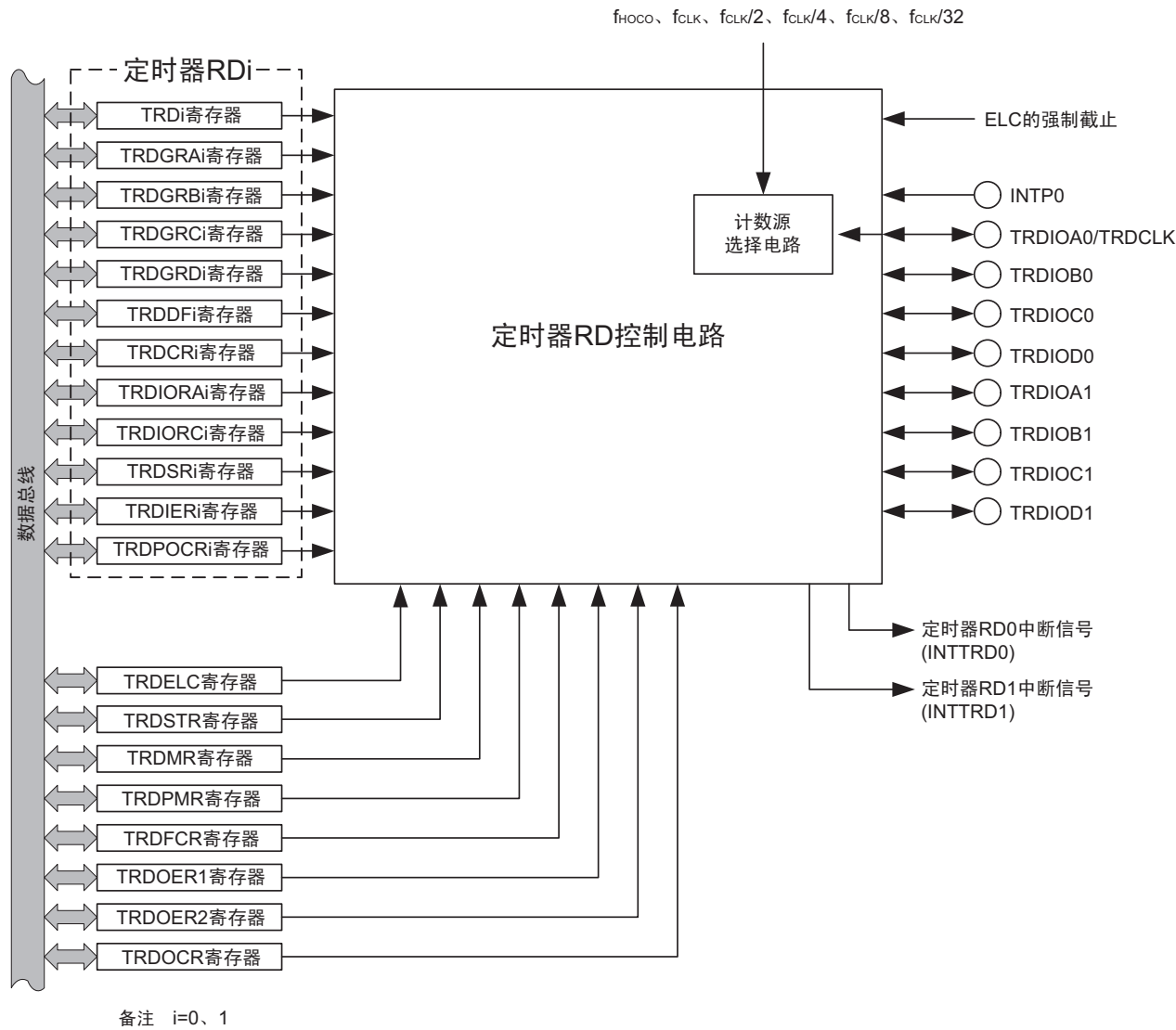


表 8-1 定时器 RD 的引脚结构

引脚名	复用的端口名	输入 / 输出	功能
TRDIOA0/TRDCLK	P17	输入 / 输出	功能因运行模式而不同，详细内容请参照各模式。
TRDIOB0	P15	输入 / 输出	
TRDIOC0	P16	输入 / 输出	
TRDIOD0	P14	输入 / 输出	
TRDIOA1	P13	输入 / 输出	
TRDIOB1	P12	输入 / 输出	
TRDIOC1	P11	输入 / 输出	
TRDIOD1	P10	输入 / 输出	

8.3 控制定时器 RD 的寄存器

控制定时器 RD 的寄存器如表 8-2 所示。

表 8-2 控制定时器 RD 的寄存器

寄存器名	符号
外围允许寄存器 1	PER1
定时器 RD ELC 寄存器	TRDEL
定时器 RD 启动寄存器	TRDSTR
定时器 RD 模式寄存器	TRDMR
定时器 RD PWM 功能选择寄存器	TRDPMR
定时器 RD 功能控制寄存器	TRDFCR
定时器 RD 输出主允许寄存器 1	TRDOER1
定时器 RD 输出主允许寄存器 2	TRDOER2
定时器 RD 输出控制寄存器	TRDOCR
定时器 RD 数字滤波器功能选择寄存器 0	TRDDF0
定时器 RD 数字滤波器功能选择寄存器 1	TRDDF1
定时器 RD 控制寄存器 0	TRDCR0
定时器 RD I/O 控制寄存器 A0	TRDIORA0
定时器 RD I/O 控制寄存器 C0	TRDIORC0
定时器 RD 状态寄存器 0	TRDSR0
定时器 RD 中断允许寄存器 0	TRDIER0
定时器 RD PWM 功能输出电平控制寄存器 0	TRDPOCR0
定时器 RD 计数器 0	TRD0
定时器 RD 通用寄存器 A0	TRDGRA0
定时器 RD 通用寄存器 B0	TRDGRB0
定时器 RD 通用寄存器 C0	TRDGRC0
定时器 RD 通用寄存器 D0	TRDGRD0
定时器 RD 控制寄存器 1	TRDCR1
定时器 RD I/O 控制寄存器 A1	TRDIORA1
定时器 RD I/O 控制寄存器 C1	TRDIORC1
定时器 RD 状态寄存器 1	TRDSR1
定时器 RD 中断允许寄存器 1	TRDIER1
定时器 RD PWM 功能输出电平控制寄存器 1	TRDPOCR1
定时器 RD 计数器 1	TRD1
定时器 RD 通用寄存器 A1	TRDGRA1
定时器 RD 通用寄存器 B1	TRDGRB1
定时器 RD 通用寄存器 C1	TRDGRC1
定时器 RD 通用寄存器 D1	TRDGRD1
端口寄存器 1	P1
端口模式寄存器 1	PM1

8.3.1 外围允许寄存器 1（PER1）

PER1 寄存器是设定允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用定时器 RD 时，必须将 bit4（TRD0EN）置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER1 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 8-2 外围允许寄存器 1（PER1）的格式

地址：F007AH	复位后：00H		R/W					
符号	7	6	5	4	3	2	1	0
PER1	0	0	0	TRD0EN	DTCEN	0	0	TRJ0EN

TRD0EN	提供定时器 RD 的输入时钟的控制
0	停止提供输入时钟。 - 不能写定时器 RD 使用的 SFR。 - 定时器 RD 处于复位状态。
1	提供输入时钟。 能读写定时器 RD 使用的 SFR。

- 注意 1. 要设定定时器 RD 时，必须先将 TRD0EN 位置“1”。当 TRD0EN 位为“0”时，忽视定时器 RD 的控制寄存器的写操作，而且读取值都为初始值（端口模式寄存器 1（PM1）和端口寄存器 1（P1）除外）。
2. 必须将以下的位置“0”。
- bit1、2、5～7
3. 要选择 f_{HOCO} 作为定时器 RD 的计数源时，必须在将外围允许寄存器 1（PER1）的 bit4（TRD0EN）置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟，就必须在清除外围允许寄存器 1（PER1）的 bit4（TRD0EN）后进行更改。

8.3.2 定时器 RD ELC 寄存器（TRDEL）

图 8-3 定时器 RD ELC 寄存器（TRDEL）的格式

地址: F0260H	复位后: 00H	R/W							
符号	7	6	5	4	3	2	1	0	
TRDEL	0	0	ELCOBE1	ELCICE1	0	0	ELCOBE0	ELCICE0	

ELCOBE1	ELC 事件输入 1（用于强制截止定时器 RD 的脉冲输出）的允许
0	禁止强制截止。
1	允许强制截止。

ELCICE1	ELC 事件输入 1（用于定时器 RD 的输入捕捉 D1）的选择
0	选择输入捕捉 TRDIOD1。
1	选择来自事件链接控制器（ELC）的事件输入 1。

ELCOBE0	ELC 事件输入 0（用于强制截止定时器 RD 的脉冲输出）的允许
0	禁止强制截止。
1	允许强制截止。

ELCICE0	ELC 事件输入 0（用于定时器 RD 的输入捕捉 D0）的选择
0	选择输入捕捉 TRDIOD0。
1	选择来自事件链接控制器（ELC）的事件输入 0。

8.3.3 定时器 RD 启动寄存器（TRDSTR）

能通过8位存储器操作指令设定TRDSTR寄存器。请参照使用定时器RD时的注意事项的“8.7.1(1) TRDSTR 寄存器”。

图 8-4 定时器 RD 启动寄存器（TRDSTR）的格式

地址：F0263H

复位后：0CH 注 1

R/W

符号	7	6	5	4	3	2	1	0
TRDSTR	0	0	0	0	CSEL1	CSEL0	TSTART1	TSTART0

CSEL1	TRD1 计数运行的选择注 2
0	在和 TRDGRA1 寄存器比较匹配时停止计数。
1	在和 TRDGRA1 寄存器比较匹配后还继续计数注 3。

CSEL0	TRD0 计数运行的选择
0	在和 TRDGRA0 寄存器比较匹配时停止计数。
1	在和 TRDGRA0 寄存器比较匹配后还继续计数注 3。

TSTART1	TRD1 计数的开始标志注 4、5
0	停止计数。
1	开始计数。

TSTART0	TRD0 计数的开始标志注 6、7
0	停止计数。
1	开始计数。

- 注
1.

当用户选项字节（000C2H）的FRQSEL4位为“1”并且PER1寄存器的TRD0EN位为“0”时，复位后的值为“00H”。
当需要读初始值时，必须在将f_{CLK}设定为f_{IH}并且将TRD0EN位置“1”后读初始值。
2.

不能用于PWM3模式。
3.

在使用输入捕捉功能时，必须将此位置“1”。
4.

当CSEL1位为“1”时，必须给TSTART1位写“0”。
5.

当CSEL1位为“0”并且产生比较匹配信号（TRDIOA1）时，此标志为“0”（停止计数）。
6.

当CSEL0位为“1”时，必须给TSTART0位写“0”。
7.

当CSEL0位为“0”并且产生比较匹配信号（TRDIOA0）时，此标志为“0”（停止计数）。

8.3.4 定时器 RD 模式寄存器（TRDMR）

图 8-5 定时器 RD 模式寄存器（TRDMR）的格式

地址: F0264H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TRDMR	TRDBFD1	TRDBFC1	TRDBFD0	TRDBFC0	0	0	0	TRDSYNC

TRDBFD1	TRDGRD1 寄存器功能的选择注 1
0	通用寄存器
1	TRDGRB1 寄存器的缓冲寄存器

TRDBFC1	TRDGRC1 寄存器功能的选择注 1
0	通用寄存器
1	TRDGRA1 寄存器的缓冲寄存器

TRDBFD0	TRDGRD0 寄存器功能的选择注 1
0	通用寄存器
1	TRDGRB0 寄存器的缓冲寄存器

TRDBFC0	TRDGRC0 寄存器功能的选择注 1、2
0	通用寄存器
1	TRDGRA0 寄存器的缓冲寄存器

TRDSYNC	定时器 RD 的同步注 3
0	TRD0 和 TRD1 独立运行。
1	TRD0 和 TRD1 同步运行。

注 1. 在使用输出比较功能时，如果通过 TRDIORCi 寄存器（i=0、1）的 IOj3（j=C 或者 D）位选择“0”（更改 TRDGRji 寄存器的输出引脚），就必须将 TRDMR 寄存器的 TRDBFji 位置“0”。

2. 在互补 PWM 模式中，必须将此位置“0”（通用寄存器）。

3. 在复位同步 PWM 模式、互补 PWM 模式或者 PWM3 模式中，必须将此位置“0”（TRD0 和 TRD1 独立运行）。

8.3.5 定时器 RD PWM 功能选择寄存器 (TRDPMR)

图 8-6 定时器 RD PWM 功能选择寄存器 (TRDPMR) 的格式 [定时器模式]

地址: F0265H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TRDPMR	0	TRDPWMD1	TRDPWMC1	TRDPWMB1	0	TRDPWMD0	TRDPWMC0	TRDPWMB0

TRDPWMD1	TRDIOD1 的 PWM 功能选择
0	输入捕捉功能或者输出比较功能
1	PWM 功能

TRDPWMC1	TRDIOC1 的 PWM 功能选择
0	输入捕捉功能或者输出比较功能
1	PWM 功能

TRDPWMB1	TRDIOB1 的 PWM 功能选择
0	输入捕捉功能或者输出比较功能
1	PWM 功能

TRDPWMD0	TRDIOD0 的 PWM 功能选择
0	输入捕捉功能或者输出比较功能
1	PWM 功能

TRDPWMC0	TRDIOC0 的 PWM 功能选择
0	输入捕捉功能或者输出比较功能
1	PWM 功能

TRDPWMB0	TRDIOB0 的 PWM 功能选择
0	输入捕捉功能或者输出比较功能
1	PWM 功能

8.3.6 定时器 RD 功能控制寄存器 (TRDFCR)

图 8-7 定时器 RD 功能控制寄存器 (TRDFCR) 的格式

地址: F0266H 复位后: 80H 注 1 R/W

符号	7	6	5	4	3	2	1	0
TRDFCR	PWM3	STCLK	0	0	OLS1	OLS0	CMD1	CMD0

PWM3	PWM3 模式的选择注 2
- 在定时器模式中, 必须置“1”(非 PWM3 模式)。 - 在 PWM3 模式中, 必须置“0”(PWM3 模式)。 - 在复位同步 PWM 模式和互补 PWM 模式中无效。	

STCLK	外部时钟输入的选择
- 定时器模式、复位同步 PWM 模式、互补 PWM 模式 0: 外部时钟输入无效 1: 外部时钟输入有效 - 在 PWM3 模式中, 必须置“0”(外部时钟输入无效)。	

OLS1	反相输出电平的选择 (复位同步 PWM 模式或者互补 PWM 模式)
- 复位同步 PWM 模式、互补 PWM 模式 0: 初始输出“H”电平, “L”电平有效。 1: 初始输出“L”电平, “H”电平有效。 - 在定时器模式和 PWM3 模式中无效。	

OLS0	正相输出电平的选择 (复位同步 PWM 模式或者互补 PWM 模式)
- 复位同步 PWM 模式、互补 PWM 模式 0: 初始输出“H”电平, “L”电平有效。 1: 初始输出“L”电平, “H”电平有效。 - 在定时器模式和 PWM3 模式中无效。	

CMD1	CMD0	组合模式的选择注 3、4
- 在定时器模式和 PWM3 模式中, 必须置“00B”(定时器模式或者 PWM3 模式)。 - 在复位同步 PWM 模式中, 必须置“01B”(复位同步 PWM 模式)。 - 互补 PWM 模式		
CMD1 CMD0 1 0: 互补 PWM 模式 (在 TRD1 发生下溢时, 将数据从缓冲寄存器传送到通用寄存器) 1 1: 互补 PWM 模式 (在 TRD0 和 TRDGRA0 寄存器比较匹配时, 将数据从缓冲寄存器传送到通用寄存器) 上述以外: 禁止设定。		

- 注 1. 当用户选项字节(000C2H)的 FRQSEL4 位为“1”并且 PER1 寄存器的 TRD0EN 位为“0”时, 复位后的值为“00H”。当需要读初始值时, 必须在将 f_{CLK} 设定为 f_{IH} 并且将 TRD0EN 位置“1”后读初始值。
2. 在 CMD1 位和 CMD0 位为“00B”(定时器模式或者 PWM3 模式)时, PWM3 位的设定有效。
3. 必须在 TRDSTR 寄存器的 TSTART0 位和 TSTART1 位都为“0”(停止计数)时写 CMD0 位和 CMD1 位。
4. 当 CMD1 位和 CMD0 位为“01B”、“10B”或者“11B”时, 与 TRDPMR 寄存器的设定无关, 为复位同步 PWM 模式或者互补 PWM 模式。

8.3.7 定时器 RD 输出主允许寄存器 1 (TRDOER1)

图 8-8 定时器 RD 输出主允许寄存器 1 (TRDOER1) 的格式
 [输出比较功能、PWM 功能、复位同步 PWM 模式、互补 PWM 模式和 PWM3 模式]

地址: F0267H 复位后: FFH 注 1 R/W

符号	7	6	5	4	3	2	1	0
TRDOER1	ED1	EC1	EB1	EA1	ED0	EC0	EB0	EA0
ED1	TRDIOD1 输出的禁止注 2							
0	允许输出。							
1	禁止输出 (TRDIOD1 引脚为 I/O 端口)。							
EC1	TRDIOC1 输出的禁止注 2							
0	允许输出。							
1	禁止输出 (TRDIOC1 引脚为 I/O 端口)。							
EB1	TRDIOB1 输出的禁止注 2							
0	允许输出。							
1	禁止输出 (TRDIOB1 引脚为 I/O 端口)。							
EA1	TRDIOA1 输出的禁止注 2、3							
0	允许输出。							
1	禁止输出 (TRDIOA1 引脚为 I/O 端口)。							
ED0	TRDIOD0 输出的禁止注 2							
0	允许输出。							
1	禁止输出 (TRDIOD0 引脚为 I/O 端口)。							
EC0	TRDIOC0 输出的禁止注 2							
0	允许输出。							
1	禁止输出 (TRDIOC0 引脚为 I/O 端口)。							
EB0	TRDIOB0 输出的禁止							
0	允许输出。							
1	禁止输出 (TRDIOB0 引脚为 I/O 端口)。							
EA0	TRDIOA0 输出的禁止注 3、4							
0	允许输出。							
1	禁止输出 (TRDIOA0 引脚为 I/O 端口)。							

- 注 1. 当用户选项字节 (000C2H) 的 FRQSEL4 位为“1”并且 PER1 寄存器的 TRD0EN 位为“0”时, 复位后的值为“00H”。当需要读初始值时, 必须在将 f_{CLK} 设定为 f_{IH} 并且将 TRD0EN 位置“1”后读初始值。
2. 在 PWM3 模式中, 必须将此位置“1”。
3. 当使用 PWM 功能时, 必须将此位置“1”。
4. 在复位同步 PWM 模式和互补 PWM 模式中, 必须将此位置“1”。

8.3.8 定时器 RD 输出主允许寄存器 2（TRDOER2）

图 8-9 定时器 RD 输出主允许寄存器 2（TRDOER2）的格式
[PWM 功能、复位同步 PWM 模式、互补 PWM 模式和 PWM3 模式]

地址：F0268H

复位后：00H

R/W

符号

7

6

5

4

3

2

1

0

TRDOER2

TRDPTO	0	0	0	0	0	0	TRDSHUTS

TRDPTO	脉冲输出强制截止信号 INTP0 引脚输入的有效注
0	脉冲输出强制截止输入无效。
1	脉冲输出强制截止输入有效（如果给 INTP0 引脚输入“L”电平，TRDSHUTS 位就为“1”）。

TRDSHUTS	强制截止标志
0	不强制截止。
1	处于强制截止。

当通过 INTP0 引脚或者 ELC 输入事件来强制截止脉冲时，此位变为“1”，而且不自动清除。因此，要停止强制截止脉冲时，必须在停止计数（TSTARTi=0）的过程中给此位写“0”。即使在有效模式中给 TRDSHUTS 位写“1”也强制截止脉冲。

注 请参照“8.4.4 脉冲输出的强制截止”。

8.3.9 定时器 RD 输出控制寄存器 (TRDOCR)

必须在 TRDSTR 寄存器的 TSTART0 位和 TSTART1 位都为“0”（停止计数）时写 TRDOCR 寄存器。

图 8-10 定时器 RD 输出控制寄存器 (TRDOCR) 的格式 [输出比较功能]

地址: F0269H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TRDOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0

TOD1	TRDIOD1 初始输出电平的选择注
0	初始输出“L”电平。
1	初始输出“H”电平。

TOC1	TRDIOC1 初始输出电平的选择注
0	初始输出“L”电平。
1	初始输出“H”电平。

TOB1	TRDIOB1 初始输出电平的选择注
0	初始输出“L”电平。
1	初始输出“H”电平。

TOA1	TRDIOA1 初始输出电平的选择
0	初始输出“L”电平。
1	初始输出“H”电平。

TOD0	TRDIOD0 初始输出电平的选择注
0	初始输出“L”电平。
1	初始输出“H”电平。

TOC0	TRDIOC0 初始输出电平的选择注
0	初始输出“L”电平。
1	初始输出“H”电平。

TOB0	TRDIOB0 初始输出电平的选择注
0	初始输出“L”电平。
1	初始输出“H”电平。

TOA0	TRDIOA0 初始输出电平的选择
0	初始输出“L”电平。
1	初始输出“H”电平。

注 在 TRDOCR 寄存器的引脚功能为波形输出的情况下设定 TRDOCR 寄存器时，输出初始的输出电平。

图 8-11 定时器 RD 输出控制寄存器 (TRDOCR) 的格式 [PWM 功能]

地址: F0269H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TRDOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0

TOD1	TRDIOD1 初始输出电平的选择 ^注
0	初始输出为无效电平。
1	初始输出为有效电平。

TOC1	TRDIOC1 初始输出电平的选择 ^注
0	初始输出为无效电平。
1	初始输出为有效电平。

TOB1	TRDIOB1 初始输出电平的选择 ^注
0	初始输出为无效电平。
1	初始输出为有效电平。

TOA1	TRDIOA1 初始输出电平的选择
必须置“0”。	

TOD0	TRDIOD0 初始输出电平的选择 ^注
0	初始输出为无效电平。
1	初始输出为有效电平。

TOC0	TRDIOC0 初始输出电平的选择 ^注
0	初始输出为无效电平。
1	初始输出为有效电平。
在复位同步 PWM 模式和互补 PWM 模式中有效。	

TOB0	TRDIOB0 初始输出电平的选择 ^注
0	初始输出为无效电平。
1	初始输出为有效电平。

TOA0	TRDIOA0 初始输出电平的选择
必须置“0”。	

注 在 TRDOCR 寄存器的引脚功能为波形输出的情况下设定 TRDOCR 寄存器时，输出初始的输出电平。

图 8-12 定时器 RD 输出控制寄存器 (TRDOCR) 的格式 [PWM3 模式]

地址：F0269H	复位后：00H		R/W						
符号	7	6	5	4	3	2	1	0	
TRDOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0	
TOD1		TRDIOD1 初始输出电平的选择							
在 PWM3 模式中无效。									
TOC1		TRDIOC1 初始输出电平的选择							
在 PWM3 模式中无效。									
TOB1		TRDIOB1 初始输出电平的选择							
在 PWM3 模式中无效。									
TOA1		TRDIOA1 初始输出电平的选择							
在 PWM3 模式中无效。									
TOD0		TRDIOD0 初始输出电平的选择							
在 PWM3 模式中无效。									
TOC0		TRDIOC0 初始输出电平的选择							
在 PWM3 模式中无效。									
TOB0		TRDIOB0 初始输出电平的选择注							
0		初始输出“L”电平，“H”电平有效。 在 TRDGRB1 比较匹配时输出“H”电平，在 TRDGRB0 比较匹配时输出“L”电平。							
1		初始输出“H”电平，“L”电平有效。 在 TRDGRB1 比较匹配时输出“L”电平，在 TRDGRB0 比较匹配时输出“H”电平。							
TOA0		TRDIOA0 初始输出电平的选择							
0		初始输出“L”电平，“H”电平有效。 在 TRDGRA1 比较匹配时输出“H”电平，在 TRDGRA0 比较匹配时输出“L”电平。							
1		初始输出“H”电平，“L”电平有效。 在 TRDGRA1 比较匹配时输出“L”电平，在 TRDGRA0 比较匹配时输出“H”电平。							

注 在 TRDOCR 寄存器的引脚功能为波形输出的情况下设定 TRDOCR 寄存器时，输出初始的输出电平。

8.3.10 定时器 RD 数字滤波器功能选择寄存器 i (TRDDFi) (i=0、1)

图 8-13 定时器 RD 数字滤波器功能选择寄存器 i (TRDDFi) (i=0、1) 的格式 [输入捕捉功能]

地址: F026AH (TRDDF0)、F026BH (TRDDF1)

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
TRDDFi	DFCK1	DFCK0	PENB1	PENB0	DFD	DFC	DFB	DFA

DFCK1	DFCK0	数字滤波器功能的时钟选择注 1
0	0	$f_{CLK}/32$ 注 2
0	1	$f_{CLK}/8$ 注 2
1	0	f_{CLK} 注 2
1	1	计数源 (TRDCRi 寄存器的 TCK0 ~ TCK2 位选择的时钟)

PENB1	PENB0	TRDIOB 引脚的脉冲强制截止的控制
0	0	必须置“00B”。

DFD	TRDIODi 引脚的数字滤波器功能的选择
0	没有数字滤波器功能。
1	有数字滤波器功能。
当有数字滤波器功能时, 最多需要 5 个数字滤波器的采样时钟周期进行边沿检测。	

DFC	TRDIOCi 引脚的数字滤波器功能的选择
0	没有数字滤波器功能。
1	有数字滤波器功能。
当有数字滤波器功能时, 最多需要 5 个数字滤波器的采样时钟周期进行边沿检测。	

DFB	TRDIOBi 引脚的数字滤波器功能的选择
0	没有数字滤波器功能。
1	有数字滤波器功能。
当有数字滤波器功能时, 最多需要 5 个数字滤波器的采样时钟周期进行边沿检测。	

DFA	TRDIOAi 引脚的数字滤波器功能的选择
0	没有数字滤波器功能。
1	有数字滤波器功能。
当有数字滤波器功能时, 最多需要 5 个数字滤波器的采样时钟周期进行边沿检测。	

注 1. 必须在设定 DFCK0 位和 DFCK1 位后开始计数。

2. 当用户选项字节 (000C2H) 的 FRQSEL4 位为“1”时, $f_{CLK}/32$ 、 $f_{CLK}/8$ 和 f_{CLK} 分别为 $f_{HOCO}/32$ 、 $f_{HOCO}/8$ 和 f_{HOCO} 。

图 8-14 定时器 RD 数字滤波器功能选择寄存器 i (TRDDFi) (i=0、1) 的格式
[PWM 功能、复位同步 PWM 模式、互补 PWM 模式和 PWM3 模式]

地址: F026AH (TRDDF0)、F026BH (TRDDF1) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TRDDFi	DFCK1	DFCK0	PENB1	PENB0	DFD	DFC	DFB	DFA

DFCK1	DFCK0	TRDIOA 引脚的脉冲强制截止的控制
0	0	禁止强制截止。
0	1	高阻抗输出
1	0	“L” 电平输出
1	1	“H” 电平输出

如果在这些模式中不将对应的引脚用作定时器 RD 的输出端口，就必须置“00B”（禁止强制截止）。而且，必须在停止计数的过程中设定这些位。

PENB1	PENB0	TRDIOB 引脚的脉冲强制截止的控制
0	0	禁止强制截止。
0	1	高阻抗输出
1	0	“L” 电平输出
1	1	“H” 电平输出

如果在这些模式中不将对应的引脚用作定时器 RD 的输出端口，就必须置“00B”（禁止强制截止）。而且，必须在停止计数的过程中设定这些位。

DFD	DFC	TRDIOC 引脚的脉冲强制截止的控制
0	0	禁止强制截止。
0	1	高阻抗输出
1	0	“L” 电平输出
1	1	“H” 电平输出

如果在这些模式中不将对应的引脚用作定时器 RD 的输出端口，就必须置“00B”（禁止强制截止）。而且，必须在停止计数的过程中设定这些位。

DFB	DFA	TRDIOD 引脚的脉冲强制截止的控制
0	0	禁止强制截止。
0	1	高阻抗输出
1	0	“L” 电平输出
1	1	“H” 电平输出

如果在这些模式中不将对应的引脚用作定时器 RD 的输出端口，就必须置“00B”（禁止强制截止）。而且，必须在停止计数的过程中设定这些位。

8.3.11 定时器 RD 控制寄存器 i (TRDCRi) (i=0、1)

在复位同步 PWM 模式和 PWM3 模式中不使用 TRDCR1 寄存器。

图 8-15 定时器 RD 控制寄存器 i (TRDCRi) (i=0、1) 的格式 [输入捕捉功能和输出比较功能]

地址: F0270H (TRDCR0)、F0280H (TRDCR1)

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
TRDCRi	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TRDi 计数器的清除选择
0	0	0	禁止清除 (自由运行)。
0	0	1	在 TRDGRAi 的输入捕捉 / 比较匹配时进行清除。
0	1	0	在 TRDGRBi 的输入捕捉 / 比较匹配时进行清除。
0	1	1	同步清除 (和其他定时器 RDi 的计数器同时清除) 注 1
1	0	1	在 TRDGRci 的输入捕捉 / 比较匹配时进行清除。
1	1	0	在 TRDGRDi 的输入捕捉 / 比较匹配时进行清除。
上述以外			禁止设定。

CKEG1	CKEG0	外部时钟沿的选择注 2
0	0	在上升沿进行计数。
0	1	在下降沿进行计数。
1	0	在双边沿进行计数。
上述以外		禁止设定。

TCK2	TCK1	TCK0	计数源的选择
0	0	0	f_{CLK} 、 f_{HOCO} 注 3
0	0	1	$f_{CLK}/2$ 注 4
0	1	0	$f_{CLK}/4$ 注 4
0	1	1	$f_{CLK}/8$ 注 4
1	0	0	$f_{CLK}/32$ 注 4
1	0	1	TRDCLK 的输入注 5
上述以外			禁止设定。

- 注 1. 在 TRDMR 寄存器的 TRDSYNC 位为“1” (TRD0 和 TRD1 同步运行) 时有效。
2. 在 TCK2 ~ TCK0 位为“101B” (TRDCLK 的输入) 并且 STCLK 位为“1” (外部时钟输入有效) 时有效。
3. 当用户选项字节 (000C2H) 的 FRQSEL4 位为“0”时, 选择 f_{CLK} ; 当 FRQSEL4 位为“1”时, 选择 f_{HOCO} 。要选择 f_{HOCO} 作为定时器 RD 的计数源时, 必须在将外围允许寄存器 1 (PER1) 的 bit4 (TRD0EN) 置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟, 就必须在清除外围允许寄存器 1 (PER1) 的 bit4 (TRD0EN) 后进行更改。
4. 不能在用户选项字节 (000C2H) 的 FRQSEL4 位为“1”时进行设定。
5. 在 TRDFCR 寄存器的 STCLK 位为“1” (外部时钟输入有效) 时有效。

图 8-16 定时器 RD 控制寄存器 i (TRDCRi) (i=0、1) 的格式 [PWM 功能]

地址: F0270H (TRDCR0)、F0280H (TRDCR1)

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
TRDCRi	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TRDi 计数器的清除选择
必须置“001B”（在和 TRDGRAi 寄存器比较匹配时清除 TRDi 寄存器）。			

CKEG1	CKEG0	外部时钟沿的选择注 1
0	0	在上升沿进行计数。
0	1	在下降沿进行计数。
1	0	在双边沿进行计数。
上述以外		禁止设定。

TCK2	TCK1	TCK0	计数源的选择
0	0	0	f_{CLK} 、 f_{HOCO} 注 2
0	0	1	$f_{CLK}/2$ 注 3
0	1	0	$f_{CLK}/4$ 注 3
0	1	1	$f_{CLK}/8$ 注 3
1	0	0	$f_{CLK}/32$ 注 3
1	0	1	TRDCLK 的输入注 4
上述以外			禁止设定。

- 注 1. 在 TCK2 ~ TCK0 位为“101B”（TRDCLK 的输入）并且 STCLK 位为“1”（外部时钟输入有效）时有效。
2. 当用户选项字节（000C2H）的 FRQSEL4 位为“0”时，选择 f_{CLK} ；当 FRQSEL4 位为“1”时，选择 f_{HOCO} 。要选择 f_{HOCO} 作为定时器 RD 的计数源时，必须在将外围允许寄存器 1（PER1）的 bit4（TRD0EN）置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟，就必须在清除外围允许寄存器 1（PER1）的 bit4（TRD0EN）后进行更改。
3. 不能在用户选项字节（000C2H）的 FRQSEL4 位为“1”时进行设定。
4. 在 TRDFCR 寄存器的 STCLK 位为“1”（外部时钟输入有效）时有效。

图 8-17 定时器 RD 控制寄存器 0 (TRDCR0) 的格式 [复位同步 PWM 模式]

地址: F0270H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TRDCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TRD0 计数器的清除选择
必须置“001B”（在和 TRDGRA0 寄存器比较匹配时清除 TRD0 寄存器）。			

CKEG1	CKEG0	外部时钟沿的选择 ^{注 1}
0	0	在上升沿进行计数。
0	1	在下降沿进行计数。
1	0	在双边沿进行计数。
上述以外		禁止设定。

TCK2	TCK1	TCK0	计数源的选择
0	0	0	f_{CLK} 、 f_{HOCO} ^{注 2}
0	0	1	$f_{CLK}/2$ ^{注 3}
0	1	0	$f_{CLK}/4$ ^{注 3}
0	1	1	$f_{CLK}/8$ ^{注 3}
1	0	0	$f_{CLK}/32$ ^{注 3}
1	0	1	TRDCLK 的输入 ^{注 4}
上述以外			禁止设定。

- 注 1. 在 TCK2 ~ TCK0 位为“101B”（TRDCLK 的输入）并且 STCLK 位为“1”（外部时钟输入有效）时有效。
2. 当用户选项字节（000C2H）的 FRQSEL4 位为“0”时，选择 f_{CLK} ；当 FRQSEL4 位为“1”时，选择 f_{HOCO} 。要选择 f_{HOCO} 作为定时器 RD 的计数源时，必须在将外围允许寄存器 1（PER1）的 bit4（TRD0EN）置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟，就必须在清除外围允许寄存器 1（PER1）的 bit4（TRD0EN）后进行更改。
3. 不能在用户选项字节（000C2H）的 FRQSEL4 位为“1”时进行设定。
4. 在 TRDFCR 寄存器的 STCLK 位为“1”（外部时钟输入有效）时有效。

图 8-18 定时器 RD 控制寄存器 0 (TRDCR0) 的格式 [互补 PWM 模式]

地址: F0270H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TRDCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TRD0 计数器的清除选择
必须置“000B”（禁止清除（自由运行））。			

CKEG1	CKEG0	外部时钟沿的选择注 1、2
0	0	在上升沿进行计数。
0	1	在下降沿进行计数。
1	0	在双边沿进行计数。
上述以外		禁止设定。

TCK2	TCK1	TCK0	计数源的选择
0	0	0	f_{CLK} 、 f_{HOCO} 注 3
0	0	1	$f_{CLK}/2$ 注 4
0	1	0	$f_{CLK}/4$ 注 4
0	1	1	$f_{CLK}/8$ 注 4
1	0	0	$f_{CLK}/32$ 注 4
1	0	1	TRDCLK 的输入注 5
上述以外			禁止设定。

- 注 1. 在 TCK2 ~ TCK0 位为“101B”（TRDCLK 的输入）并且 STCLK 位为“1”（外部时钟输入有效）时有效。
2. 必须给 TRDCR0 寄存器和 TRDCR1 寄存器的 TCK0 ~ TCK2 位、CKEG0 位和 CKEG1 位设定相同的值。
3. 当用户选项字节（000C2H）的 FRQSEL4 位为“0”时，选择 f_{CLK} ；当 FRQSEL4 位为“1”时，选择 f_{HOCO} 。要选择 f_{HOCO} 作为定时器 RD 的计数源时，必须在将外围允许寄存器 1（PER1）的 bit4（TRD0EN）置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟，就必须在清除外围允许寄存器 1（PER1）的 bit4（TRD0EN）后进行更改。
4. 不能在用户选项字节（000C2H）的 FRQSEL4 位为“1”时进行设定。
5. 在 TRDFCR 寄存器的 STCLK 位为“1”（外部时钟输入有效）时有效。

图 8-19 定时器 RD 控制寄存器 0 (TRDCR0) 的格式 [PWM3 模式]

地址: F0270H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TRDCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TRD0 计数器的清除选择
必须置“001B”（在和 TRDGRA0 寄存器比较匹配时清除 TRD0 寄存器）。			

CKEG1	CKEG0	外部时钟沿的选择
在 PWM3 模式中无效。		

TCK2	TCK1	TCK0	计数源的选择
0	0	0	f_{CLK} 、 f_{HOCO} 注 1
0	0	1	$f_{CLK}/2$ 注 2
0	1	0	$f_{CLK}/4$ 注 2
0	1	1	$f_{CLK}/8$ 注 2
1	0	0	$f_{CLK}/32$ 注 2
上述以外			禁止设定。

- 注 1. 当用户选项字节（000C2H）的 FRQSEL4 位为“0”时，选择 f_{CLK} ；当 FRQSEL4 位为“1”时，选择 f_{HOCO} 。要选择 f_{HOCO} 作为定时器 RD 的计数源时，必须在定时器开始计数前将 f_{CLK} 设定为 f_{IH} 。
2. 不能在用户选项字节（000C2H）的 FRQSEL4 位为“1”时进行设定。

8.3.12 定时器 RD I/O 控制寄存器 Ai (TRDIORAi) (i=0、1)

图 8-20 定时器 RD I/O 控制寄存器 Ai (TRDIORAi) (i=0、1) 的格式 [输入捕捉功能]

地址: F0271H (TRDIORA0)、F0281H (TRDIORA1) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TRDIORAi	0	IOB2	IOB1	IOB0	0	IOA2	IOA1	IOA0

IOB2	TRDGRB 模式的选择注 1
在使用输入捕捉功能时, 必须置“1”(输入捕捉)。	

IOB1	IOB0	TRDGRB 控制
0	0	在上升沿输入捕捉到 TRDGRBi。
0	1	在下降沿输入捕捉到 TRDGRBi。
1	0	在双边沿输入捕捉到 TRDGRBi。
上述以外		禁止设定。

IOA2	TRDGRA 模式的选择注 2
在使用输入捕捉功能时, 必须置“1”(输入捕捉)。	

IOA1	IOA0	TRDGRA 控制
0	0	在上升沿输入捕捉到 TRDGRAi。
0	1	在下降沿输入捕捉到 TRDGRAi。
1	0	在双边沿输入捕捉到 TRDGRAi。
上述以外		禁止设定。

- 注 1. 如果通过 TRDMR 寄存器的 TRDBFDi 位选择“1”(TRDGRBi 寄存器的缓冲寄存器), 就必须给 TRDIORAi 寄存器的 IOB2 位和 TRDIORCi 寄存器的 IOD2 位设定相同的值。
2. 如果通过 TRDMR 寄存器的 TRDBFCi 位选择“1”(TRDGRAi 寄存器的缓冲寄存器), 就必须给 TRDIORAi 寄存器的 IOA2 位和 TRDIORCi 寄存器的 IOC2 位设定相同的值。

图 8-21 定时器 RD I/O 控制寄存器 Ai (TRDIORAi) (i=0、1) 的格式 [输出比较功能]

地址: F0271H (TRDIORA0)、F0281H (TRDIORA1) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TRDIORAi	0	IOB2	IOB1	IOB0	0	IOA2	IOA1	IOA0

IOB2	TRDGRB 模式的选择注 1
在使用输出比较功能时, 必须置“0”(输出比较)。	

IOB1	IOB0	TRDGRB 控制
0	0	禁止比较匹配的引脚输出 (TRDIOBi 引脚为 I/O 端口)。
0	1	在 TRDGRBi 比较匹配时输出“L”电平。
1	0	在 TRDGRBi 比较匹配时输出“H”电平。
1	1	在 TRDGRBi 比较匹配时进行交替输出。

IOA2	TRDGRA 模式的选择注 2
在使用输出比较功能时, 必须置“0”(输出比较)。	

IOA1	IOA0	TRDGRA 控制
0	0	禁止比较匹配的引脚输出 (TRDIOAi 引脚为 I/O 端口)。
0	1	在 TRDGRAi 比较匹配时输出“L”电平。
1	0	在 TRDGRAi 比较匹配时输出“H”电平。
1	1	在 TRDGRAi 比较匹配时进行交替输出。

- 注 1. 如果通过 TRDMR 寄存器的 TRDBFDi 位选择“1”(TRDGRBi 寄存器的缓冲寄存器), 就必须给 TRDIORAi 寄存器的 IOB2 位和 TRDIORCi 寄存器的 IOD2 位设定相同的值。
2. 如果通过 TRDMR 寄存器的 TRDBFCi 位选择“1”(TRDGRAi 寄存器的缓冲寄存器), 就必须给 TRDIORAi 寄存器的 IOA2 位和 TRDIORCi 寄存器的 IOC2 位设定相同的值。

8.3.13 定时器 RD I/O 控制寄存器 Ci (TRDIORCi) (i=0、1)

图 8-22 定时器 RD I/O 控制寄存器 Ci (TRDIORCi) (i=0、1) 的格式 [输入捕捉功能]

地址: F0272H (TRDIORC0)、F0282H (TRDIORC1) 复位后: 88H 注 1 R/W

符号	7	6	5	4	3	2	1	0
TRDIORCi	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0

IOD3	TRDGRD 寄存器功能的选择
在使用输入捕捉功能时, 必须置“1”(通用寄存器或者缓冲寄存器)。	

IOD2	TRDGRD 模式的选择注 2
在使用输入捕捉功能时, 必须置“1”(输入捕捉)。	

IOD1	IOD0	TRDGRD 控制
0	0	在上升沿输入捕捉到 TRDGRDi。
0	1	在下降沿输入捕捉到 TRDGRDi。
1	0	在双边沿输入捕捉到 TRDGRDi。
上述以外		禁止设定。

IOC3	TRDGRC 寄存器功能的选择
在使用输入捕捉功能时, 必须置“1”(通用寄存器或者缓冲寄存器)。	

IOC2	TRDGRC 模式的选择注 3
在使用输入捕捉功能时, 必须置“1”(输入捕捉)。	

IOC1	IOC0	TRDGRC 控制
0	0	在上升沿输入捕捉到 TRDGRCi。
0	1	在下降沿输入捕捉到 TRDGRCi。
1	0	在双边沿输入捕捉到 TRDGRCi。
上述以外		禁止设定。

- 注 1. 当用户选项字节(000C2H)的 FRQSEL4 位为“1”并且 PER1 寄存器的 TRD0EN 位为“0”时, 复位后的值为“00H”。当需要读初始值时, 必须在将 f_{CLK} 设定为 f_{IH} 并且将 TRD0EN 位置“1”后读初始值。
2. 如果通过 TRDMR 寄存器的 TRDBFDi 位选择“1”(TRDGRBi 寄存器的缓冲寄存器), 就必须给 TRDIORAi 寄存器的 IOB2 位和 TRDIORCi 寄存器的 IOD2 位设定相同的值。
3. 如果通过 TRDMR 寄存器的 TRDBFCi 位选择“1”(TRDGRAi 寄存器的缓冲寄存器), 就必须给 TRDIORAi 寄存器的 IOA2 位和 TRDIORCi 寄存器的 IOC2 位设定相同的值。

图 8-23 定时器 RD I/O 控制寄存器 Ci (TRDIORCi) (i=0、1) 的格式 [输出比较功能]

地址: F0272H (TRDIORC0)、F0282H (TRDIORC1) 复位后: 88H 注 1 R/W

符号	7	6	5	4	3	2	1	0
TRDIORCi	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0

IOD3	TRDGRD 寄存器功能的选择
0	TRDIOB 输出寄存器 (参照“8.5.2(2) TRDGRCi 寄存器和 TRDGRDi 寄存器 (i=0、1) 的输出引脚的变更”)
1	通用寄存器或者缓冲寄存器

IOD2	TRDGRD 模式的选择注 2
在使用输出比较功能时, 必须置“0”(输出比较)。	

IOD1	IOD0	TRDGRD 控制
0	0	禁止比较匹配的引脚输出。
0	1	在 TRDGRDi 比较匹配时输出“L”电平。
1	0	在 TRDGRDi 比较匹配时输出“H”电平。
1	1	在 TRDGRDi 比较匹配时进行交替输出。

IOC3	TRDGRC 寄存器功能的选择
0	TRDIOA 输出寄存器 (参照“8.5.2(2) TRDGRCi 寄存器和 TRDGRDi 寄存器 (i=0、1) 的输出引脚的变更”)
1	通用寄存器或者缓冲寄存器

IOC2	TRDGRC 模式的选择注 3
在使用输出比较功能时, 必须置“0”(输出比较)。	

IOC1	IOC0	TRDGRC 控制
0	0	禁止比较匹配的引脚输出。
0	1	在 TRDGRCi 比较匹配时输出“L”电平。
1	0	在 TRDGRCi 比较匹配时输出“H”电平。
1	1	在 TRDGRCi 比较匹配时进行交替输出。

- 注 1. 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TRD0EN位为“0”时, 复位后的值为“00H”。当需要读初始值时, 必须在将 f_{CLK} 设定为 f_{IH} 并且将TRD0EN位置“1”后读初始值。
2. 如果通过TRDMR寄存器的TRDBFDi位选择“1”(TRDGRBi寄存器的缓冲寄存器), 就必须给TRDIORAi寄存器的IOB2位和TRDIORCi寄存器的IOD2位设定相同的值。
3. 如果通过TRDMR寄存器的TRDBFCi位选择“1”(TRDGRAi寄存器的缓冲寄存器), 就必须给TRDIORAi寄存器的IOA2位和TRDIORCi寄存器的IOC2位设定相同的值。

8.3.14 定时器 RD 状态寄存器 0 (TRDSR0)

图 8-24 定时器 RD 状态寄存器 0 (TRDSR0) 的格式 [输入捕捉功能]

地址: F0273H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TRDSR0	0	0	0	OVF	IMFD	IMFC	IMFB	IMFA
OVF		上溢标志注 1						
[为“0”的条件]		读后写“0”注 2。						
[为“1”的条件]		当 TRD0 发生上溢时						
IMFD		输入捕捉 / 比较匹配标志 D 注 5						
[为“0”的条件]		读后写“0”注 2。						
[为“1”的条件]		TRDIOD0 引脚的输入边沿注 3						
IMFC		输入捕捉 / 比较匹配标志 C 注 5						
[为“0”的条件]		读后写“0”注 2。						
[为“1”的条件]		TRDIOC0 引脚的输入边沿注 3						
IMFB		输入捕捉 / 比较匹配标志 B 注 5						
[为“0”的条件]		读后写“0”注 2。						
[为“1”的条件]		TRDIOB0 引脚的输入边沿注 4						
IMFA		输入捕捉 / 比较匹配标志 A 注 5						
[为“0”的条件]		读后写“0”注 2。						
[为“1”的条件]		TRDIOA0 引脚的输入边沿注 4						

注 1. 在定时器 RD0 的计数器的值从“FFFFH”变为“0000H”时，上溢标志变为“1”。另外，根据 TRDCR0 寄存器的 CCLR0 ~ CCLR2 位的设定，如果在运行过程中因发生输入捕捉或者比较匹配而使定时器 RD0 的计数器的值从“FFFFH”变为“0000H”，上溢标志就变为“1”。

注 2. 写的结果如下：

- 当写“1”时，此位不变。
- 在读取值为“0”的情况下，即使给相同的位写“0”也不变（在读后从“0”变为“1”的情况下，即使写“0”也保持“1”的状态）。
- 在读取值为“1”的情况下，如果给相同的位写“0”，此位就变为“0”。

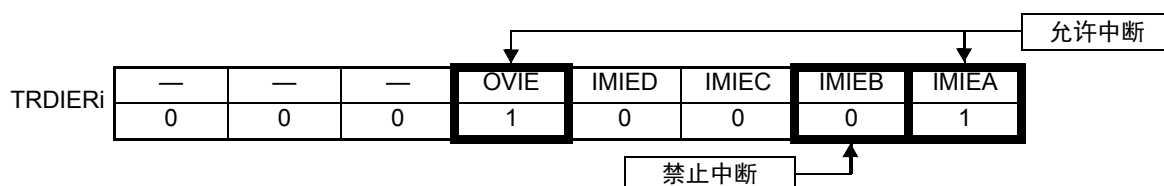
但是，要将定时器 RD 的某个中断源的状态标志（以下称为“对象状态标志”）置“0”时，如果该中断被定时器 RD 中断允许寄存器 i（TRDIERi）设定为禁止中断，就必须用以下 (a) ~ (c) 的任意一种方法置“0”。

(a) 必须在将定时器 RD 中断允许寄存器 i（TRDIERi）置“00H”（禁止全部中断）后给对象状态标志写“0”。

(b) 当定时器 RD 中断允许寄存器 i（TRDIERi）中有被置“1”（允许）的位并且该位允许的中断源状态标志为“0”时，必须给对象状态标志写“0”。

（例）在 IMIEA 和 OVIE 为允许中断而 IMIEB 为禁止中断的状态下清除 IMFB 的情况

- 定时器 RD 中断允许寄存器 i（TRDIERi）的状态



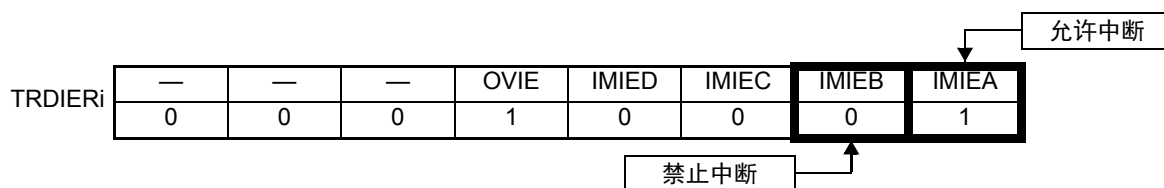
- 定时器 RD 状态寄存器 i（TRDSRi）的状态



(c) 当定时器 RD 中断允许寄存器 i（TRDIERi）中有被置“1”（允许）的位并且该位允许的中断源状态标志为“1”时，必须同时给此状态标志和对象状态标志写“0”。

（例）在 IMIEA 为允许中断而 IMIEB 为禁止中断的状态下清除 IMFB 的情况

- 定时器 RD 中断允许寄存器 i（TRDIERi）的状态



- 定时器 RD 状态寄存器 i（TRDSRi）的状态



3. 这是 TRDIORC0 寄存器的 IOk1 位和 IOk0 位（k=C 或者 D）选择的边沿。
包括 TRDMR 寄存器的 TRDBFk0 位为“1”（TRDGRk0 为缓冲寄存器）的情况。
4. 这是 TRDIORA0 寄存器的 IOj1 位和 IOj0 位（j=A 或者 B）选择的边沿。
5. 当使用 DTC 时，IMFA 位、IMFB 位、IMFC 位和 IMFD 位在 DTC 传送结束后变为“1”。

图 8-25 定时器 RD 状态寄存器 0 (TRDSR0) 的格式 [输入捕捉以外的功能]

地址: F0273H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TRDSR0	0	0	0	OVF	IMFD	IMFC	IMFB	IMFA

OVF	上溢标志注 2
[为“0”的条件] 读后写“0”注 1。 [为“1”的条件] 当 TRD0 发生上溢时	

IMFD	输入捕捉 / 比较匹配标志 D 注 4
[为“0”的条件] 读后写“0”注 1。 [为“1”的条件] 当 TRD0 和 TRDGRD0 的值相同时注 3	

IMFC	输入捕捉 / 比较匹配标志 C 注 4
[为“0”的条件] 读后写“0”注 1。 [为“1”的条件] 当 TRD0 和 TRDGRD0 的值相同时注 3	

IMFB	输入捕捉 / 比较匹配标志 B 注 4
[为“0”的条件] 读后写“0”注 1。 [为“1”的条件] 当 TRD0 和 TRDGRB0 的值相同时	

IMFA	输入捕捉 / 比较匹配标志 A 注 4
[为“0”的条件] 读后写“0”注 1。 [为“1”的条件] 当 TRD0 和 TRDGRA0 的值相同时	

注 1. 写的结果如下：

- 当写“1”时，此位不变。
- 在读取值为“0”的情况下，即使给相同的位写“0”也不变（在读后从“0”变为“1”的情况下，即使写“0”也保持“1”的状态）。
- 在读取值为“1”的情况下，如果给相同的位写“0”，此位就变为“0”。

但是，要将定时器 RD 的某个中断源的状态标志（以下称为“对象状态标志”）置“0”时，如果该中断被定时器 RD 中断允许寄存器 i（TRDIERi）设定为禁止中断，就必须用以下 (a) ~ (c) 的任意一种方法置“0”。

(a) 必须在将定时器 RD 中断允许寄存器 i（TRDIERi）置“00H”（禁止全部中断）后给对象状态标志写“0”。

(b) 当定时器 RD 中断允许寄存器 i（TRDIERi）中有被置“1”（允许）的位并且该位允许的中断源状态标志为“0”时，必须给对象状态标志写“0”。

（例）在 IMIEA 和 OVIE 为允许中断而 IMIEB 为禁止中断的状态下清除 IMFB 的情况

- 定时器 RD 中断允许寄存器 i（TRDIERi）的状态

TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
	0	0	0	1	0	0	0	1

禁止中断

允许中断

- 定时器 RD 状态寄存器 i（TRDSRi）的状态

TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
	0	0	0	0	1	0	1	0

因为允许中断的位所对应的状态标志（OVF、IMFA）为“0”，所以必须给 IMFB 写“0”。

要清除请求的位

(c) 当定时器 RD 中断允许寄存器 i（TRDIERi）中有被置“1”（允许）的位并且该位允许的中断源状态标志为“1”时，必须同时给此状态标志和对象状态标志写“0”。

（例）在 IMIEA 为允许中断而 IMIEB 为禁止中断的状态下清除 IMFB 的情况

- 定时器 RD 中断允许寄存器 i（TRDIERi）的状态

TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
	0	0	0	1	0	0	0	1

禁止中断

允许中断

- 定时器 RD 状态寄存器 i（TRDSRi）的状态

TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
	0	0	0	0	1	0	1	1

因为允许中断的位所对应的状态标志（IMFA）为“1”，所以必须同时给 IMFA 和 IMFB 写“0”。

要清除请求的位

- 在定时器 RD0 的计数器的值从“FFFFH”变为“0000H”时，上溢标志变为“1”。另外，根据 TRDCR0 寄存器的 CCLR0 ~ CCLR2 位的设定，如果在运行过程中因发生输入捕捉或者比较匹配而使定时器 RD0 的计数器的值从“FFFFH”变为“0000H”，上溢标志就变为“1”。
- 包括 TRDMR 寄存器的 TRDBFk0 位（k=C 或者 D）为“1”（TRDGRk0 为缓冲寄存器）的情况。
- 当使用 DTC 时，IMFA 位、IMFB 位、IMFC 位和 IMFD 位在 DTC 传送结束后变为“1”。

8.3.15 定时器 RD 状态寄存器 1 (TRDSR1)

图 8-26 定时器 RD 状态寄存器 1 (TRDSR1) 的格式 [输入捕捉功能]

地址: F0283H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TRDSR1	0	0	UDF	OVF	IMFD	IMFC	IMFB	IMFA

UDF	下溢标志
在使用输入捕捉功能时无效。	

OVF	上溢标志注 1
[为“0”的条件] 读后写“0”注 2。 [为“1”的条件] 当 TRD1 发生上溢时	

IMFD	输入捕捉 / 比较匹配标志 D 注 5
[为“0”的条件] 读后写“0”注 2。 [为“1”的条件] TRDIOD1 引脚的输入边沿注 3	

IMFC	输入捕捉 / 比较匹配标志 C 注 5
[为“0”的条件] 读后写“0”注 2。 [为“1”的条件] TRDIOC1 引脚的输入边沿注 3	

IMFB	输入捕捉 / 比较匹配标志 B 注 5
[为“0”的条件] 读后写“0”注 2。 [为“1”的条件] TRDIOB1 引脚的输入边沿注 4	

IMFA	输入捕捉 / 比较匹配标志 A 注 5
[为“0”的条件] 读后写“0”注 2。 [为“1”的条件] TRDIOA1 引脚的输入边沿注 4	

注 1. 在定时器 RD1 的计数器的值从“FFFFH”变为“0000H”时，上溢标志变为“1”。另外，根据 TRDCR1 寄存器的 CCLR0 ~ CCLR2 位的设定，如果在运行过程中因发生输入捕捉或者比较匹配而使定时器 RD1 的计数器的值从“FFFFH”变为“0000H”，上溢标志就变为“1”。

注 2. 写的结果如下：

- 当写“1”时，此位不变。
- 在读取值为“0”的情况下，即使给相同的位写“0”也不变（在读后从“0”变为“1”的情况下，即使写“0”也保持“1”的状态）。
- 在读取值为“1”的情况下，如果给相同的位写“0”，此位就变为“0”。

但是，要将定时器 RD 的某个中断源的状态标志（以下称为“对象状态标志”）置“0”时，如果该中断被定时器 RD 中断允许寄存器 i（TRDIERi）设定为禁止中断，就必须用以下 (a) ~ (c) 的任意一种方法置“0”。

(a) 必须在将定时器 RD 中断允许寄存器 i（TRDIERi）置“00H”（禁止全部中断）后给对象状态标志写“0”。

(b) 当定时器 RD 中断允许寄存器 i（TRDIERi）中有被置“1”（允许）的位并且该位允许的中断源状态标志为“0”时，必须给对象状态标志写“0”。

（例）在 IMIEA 和 OVIE 为允许中断而 IMIEB 为禁止中断的状态下清除 IMFB 的情况

- 定时器 RD 中断允许寄存器 i（TRDIERi）的状态

TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
	0	0	0	1	0	0	0	1

允许中断

禁止中断

- 定时器 RD 状态寄存器 i（TRDSRi）的状态

TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
	0	0	0	0	1	0	1	0

要清除请求的位

因为允许中断的位所对应的状态标志（OVF、IMFA）为“0”，所以必须给 IMFB 写“0”。

(c) 当定时器 RD 中断允许寄存器 i（TRDIERi）中有被置“1”（允许）的位并且该位允许的中断源状态标志为“1”时，必须同时给此状态标志和对象状态标志写“0”。

（例）在 IMIEA 为允许中断而 IMIEB 为禁止中断的状态下清除 IMFB 的情况

- 定时器 RD 中断允许寄存器 i（TRDIERi）的状态

TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
	0	0	0	1	0	0	0	1

允许中断

禁止中断

- 定时器 RD 状态寄存器 i（TRDSRi）的状态

TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
	0	0	0	0	1	0	1	1

要清除请求的位

因为允许中断的位所对应的状态标志（IMFA）为“1”，所以必须同时给 IMFA 和 IMFB 写“0”。

3. 这是 TRDIORC1 寄存器的 IOk1 位和 IOk0 位（k=C 或者 D）选择的边沿。
包括 TRDMR 寄存器的 TRDBFk1 位为“1”（TRDGRk1 为缓冲寄存器）的情况。
4. 这是 TRDIORA1 寄存器的 IOj1 位和 IOj0 位（j=A 或者 B）选择的边沿。
5. 当使用 DTC 时，IMFA 位、IMFB 位、IMFC 位和 IMFD 位在 DTC 传送结束后变为“1”。

图 8-27 定时器 RD 状态寄存器 1 (TRDSR1) 的格式 [输入捕捉以外的功能]

地址: F0283H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TRDSR1	0	0	UDF	OVF	IMFD	IMFC	IMFB	IMFA

UDF	下溢标志
互补 PWM 模式的情况 [为“0”的条件] 读后写“0”注 1。 [为“1”的条件] 当 TRD1 发生下溢时 在非互补 PWM 模式中无效。	

OVF	上溢标志注 2
[为“0”的条件] 读后写“0”注 1。 [为“1”的条件] 当 TRD1 发生上溢时	

IMFD	输入捕捉 / 比较匹配标志 D 注 4
[为“0”的条件] 读后写“0”注 1。 [为“1”的条件] 当 TRD1 和 TRDGRD1 的值相同时注 3	

IMFC	输入捕捉 / 比较匹配标志 C 注 4
[为“0”的条件] 读后写“0”注 1。 [为“1”的条件] 当 TRD1 和 TRDGRC1 的值相同时注 3	

IMFB	输入捕捉 / 比较匹配标志 B 注 4
[为“0”的条件] 读后写“0”注 1。 [为“1”的条件] 当 TRD1 和 TRDGRB1 的值相同时	

IMFA	输入捕捉 / 比较匹配标志 A 注 4
[为“0”的条件] 读后写“0”注 1。 [为“1”的条件] 当 TRD1 和 TRDGRA1 的值相同时	

注 1. 写的结果如下：

- 当写“1”时，此位不变。
- 在读取值为“0”的情况下，即使给相同的位写“0”也不变（在读后从“0”变为“1”的情况下，即使写“0”也保持“1”的状态）。
- 在读取值为“1”的情况下，如果给相同的位写“0”，此位就变为“0”。

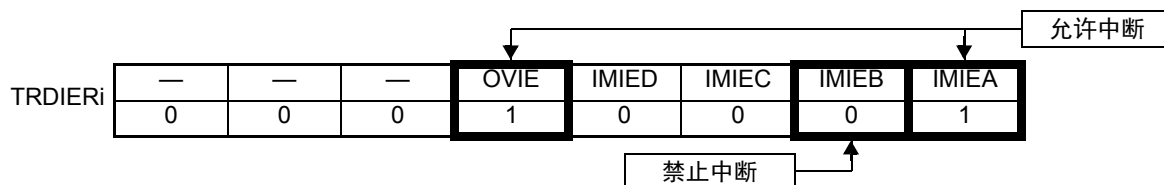
但是，要将定时器 RD 的某个中断源的状态标志（以下称为“对象状态标志”）置“0”时，如果该中断被定时器 RD 中断允许寄存器 i（TRDIERi）设定为禁止中断，就必须用以下 (a) ~ (c) 的任意一种方法置“0”。

(a) 必须在将定时器 RD 中断允许寄存器 i（TRDIERi）置“00H”（禁止全部中断）后给对象状态标志写“0”。

(b) 当定时器 RD 中断允许寄存器 i（TRDIERi）中有被置“1”（允许）的位并且该位允许的中断源状态标志为“0”时，必须给对象状态标志写“0”。

（例）在 IMIEA 和 OVIE 为允许中断而 IMIEB 为禁止中断的状态下清除 IMFB 的情况

- 定时器 RD 中断允许寄存器 i（TRDIERi）的状态



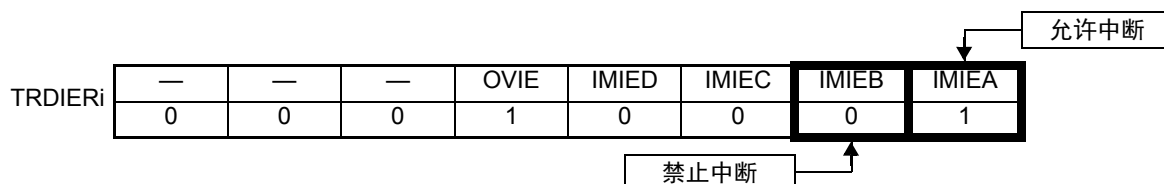
- 定时器 RD 状态寄存器 i（TRDSRi）的状态



(c) 当定时器 RD 中断允许寄存器 i（TRDIERi）中有被置“1”（允许）的位并且该位允许的中断源状态标志为“1”时，必须同时给此状态标志和对象状态标志写“0”。

（例）在 IMIEA 为允许中断而 IMIEB 为禁止中断的状态下清除 IMFB 的情况

- 定时器 RD 中断允许寄存器 i（TRDIERi）的状态



- 定时器 RD 状态寄存器 i（TRDSRi）的状态



2. 在定时器 RD1 的计数器的值从“FFFFH”变为“0000H”时，上溢标志变为“1”。另外，根据 TRDCR1 寄存器的 CCLR0 ~ CCLR2 位的设定，如果在运行过程中因发生输入捕捉或者比较匹配而使定时器 RD1 的计数器的值从“FFFFH”变为“0000H”，上溢标志就变为“1”。
3. 包括 TRDMR 寄存器的 TRDBFk1 位（k=C 或者 D）为“1”（TRDGRk1 为缓冲寄存器）的情况。
4. 当使用 DTC 时，IMFA 位、IMFB 位、IMFC 位和 IMFD 位在 DTC 传送结束后变为“1”。

8.3.16 定时器 RD 中断允许寄存器 i (TRDIERi) (i=0、1)

图 8-28 定时器 RD 中断允许寄存器 i (TRDIERi) (i=0、1) 的格式

地址: F0274H (TRDIER0)、F0284H (TRDIER1)

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
TRDIERi	0	0	0	OVIE	IMIED	IMIEC	IMIEB	IMIEA

OVIE	上溢 / 下溢中断的允许
0	禁止因 OVF 位和 UDF 位产生的中断 (OVI)。
1	允许因 OVF 位和 UDF 位产生的中断 (OVI)。

IMIED	输入捕捉 / 比较匹配的中断允许 D
0	禁止因 IMFD 位产生的中断 (IMID)。
1	允许因 IMFD 位产生的中断 (IMID)。

IMIEC	输入捕捉 / 比较匹配的中断允许 C
0	禁止因 IMFC 位产生的中断 (IMIC)。
1	允许因 IMFC 位产生的中断 (IMIC)。

IMIEB	输入捕捉 / 比较匹配的中断允许 B
0	禁止因 IMFB 位产生的中断 (IMIB)。
1	允许因 IMFB 位产生的中断 (IMIB)。

IMIEA	输入捕捉 / 比较匹配的中断允许 A
0	禁止因 IMFA 位产生的中断 (IMIA)。
1	允许因 IMFA 位产生的中断 (IMIA)。

8.3.17 定时器 RD PWM 功能输出电平控制寄存器 i (TRDPOCRi) (i=0、1)

只在使用 PWM 功能时，TRDPOCRi 寄存器的设定有效，否则 TRDPOCRi 寄存器的设定无效。

图 8-29 定时器 RD PWM 功能输出电平控制寄存器 i (TRDPOCRi) (i=0、1) 的格式 [PWM 功能]

地址: F0275H (TRDPOCR0)、F0285H (TRDPOCR1)					复位后: 00H		R/W	
符号	7	6	5	4	3	2	1	0
TRDPOCRi	0	0	0	0	0	POLD	POLC	POLB

POLD	PWM 功能的输出电平控制 D
0	TRDIODi 输出电平为“L”电平有效。
1	TRDIODi 输出电平为“H”电平有效。

POLC	PWM 功能的输出电平控制 C
0	TRDIOCi 输出电平为“L”电平有效。
1	TRDIOCi 输出电平为“H”电平有效。

POLB	PWM 功能的输出电平控制 B
0	TRDIOBi 输出电平为“L”电平有效。
1	TRDIOBi 输出电平为“H”电平有效。

8.3.18 定时器 RD 计数器 i (TRDi) (i=0、1)

[定时器模式]

必须以 16 位而不能以 8 位为单位存取 TRDi 寄存器。

[复位同步 PWM 模式和 PWM3 模式]

必须以 16 位而不能以 8 位为单位存取 TRD0 寄存器。在复位同步 PWM 模式和 PWM3 模式中，不使用 TRD1 寄存器。

[互补 PWM 模式 (TRD0)]

必须以 16 位而不能以 8 位为单位存取 TRD0 寄存器。

[互补 PWM 模式 (TRD1)]

必须以 16 位而不能以 8 位为单位存取 TRD1 寄存器。

图 8-30 定时器 RD 计数器 i (TRDi) (i=0、1) 的格式 [定时器模式]

地址: F0276H (TRD0)、F0286H (TRD1)						复位后: 0000H		R/W								
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDi																

—	功能	设定范围
bit15 ~ 0	对计数源进行递增计数。 如果发生上溢, TRDSRi 寄存器的 OVF 位就变为“1”。	0000H ~ FFFFH

图 8-31 定时器 RD 计数器 i (TRDi) (i=0、1) 的格式 [复位同步 PWM 模式和 PWM3 模式]

地址: F0276H (TRD0)、F0286H (TRD1)																	复位后: 0000H		R/W	
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
TRDi																				

—	功能	设定范围
bit15 ~ 0	对计数源进行递增计数。 如果发生上溢, TRDSR0 寄存器的 OVF 位就变为“1”。	0000H ~ FFFFH

图 8-32 定时器 RD 计数器 i（TRDi）（i=0、1）的格式 [互补 PWM 模式（TRD0）]

地址: F0276H (TRD0)、F0286H (TRD1)																	复位后: 0000H		R/W	
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
TRDi																				

—	功能	设定范围
bit15 ~ 0	必须设定死区时间。 对计数源进行递增计数或者递减计数。 如果发生上溢， TRDSR0 寄存器的 OVF 位就变为“1”。	0001H ~ FFFFH

图 8-33 定时器 RD 计数器 i（TRDi）（i=0、1）的格式 [互补 PWM 模式（TRD1）]

地址: F0276H (TRD0)、F0286H (TRD1)																复位后: 0000H		R/W	
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
TRDi																			

—	功能	设定范围
bit15 ~ 0	必须置“0000H”。 对计数源进行递增计数或者递减计数。 如果发生上溢， TRDSR1 寄存器的 UDF 位就变为“1”。	0000H ~ FFFFH

8.3.19 定时器 RD 通用寄存器 Ai、Bi、Ci、Di (TRDGRAi、TRDGRBi、TRDGRCi、TRDGRDi) (i=0、1)

[输入捕捉功能]

必须以 16 位而不能以 8 位为单位存取 TRDGRAi ~ TRDGRDi 寄存器。

在使用输入捕捉功能时，以下寄存器无效：

TRDOER1、TRDOER2、TRDOCR、TRDPOCR0、TRDPOCR1

当不使用数字滤波器时 (TRDDFi 寄存器的 DFj 位为“0”)，TRDIOji 引脚输入的捕捉信号的脉宽必须至少为 3 个定时器 RD 的运行时钟 (f_{CLK}) 周期。

[输出比较功能]

必须以 16 位而不能以 8 位为单位存取 TRDGRAi ~ TRDGRDi 寄存器。

在使用输出比较功能时，以下寄存器无效：

TRDDF0、TRDDF1、TRDPOCR0、TRDPOCR1

[PWM 功能]

必须以 16 位而不能以 8 位为单位存取 TRDGRAi ~ TRDGRDi 寄存器。

在使用 PWM 时，以下寄存器无效：

TRDDF0、TRDDF1、TRDIORA0、TRDIORC0、TRDIORA1、TRDIORC1

[复位同步 PWM 模式]

必须以 16 位而不能以 8 位为单位存取 TRDGRAi ~ TRDGRDi 寄存器。

在复位同步 PWM 模式中，以下寄存器无效：

TRDPMR、TRDOCR^注、TRDDF0、TRDDF1、TRDIORA0、TRDIORC0、TRDPOCR0、
TRDIORA1、TRDIORC1、TRDPOCR1

注 作为复位同步 PWM 模式和互补 PWM 模式中的 TRDIOC0 的初始输出设定，只有 TRDOCR 寄存器的 TOC0 位有效。

[互补 PWM 模式]

必须以 16 位而不能以 8 位为单位存取 TRDGRAi ~ TRDGRDi 寄存器。

在互补 PWM 模式中，不使用 TRDGRC0 寄存器。

在互补 PWM 模式中，以下寄存器无效：

TRDPMR、TRDOCR^注、TRDDF0、TRDDF1、TRDIORA0、TRDIORC0、TRDPOCR0、
TRDIORA1、TRDIORC1、TRDPOCR1

注 作为复位同步 PWM 模式和互补 PWM 模式中的 TRDIOC0 的初始输出设定，只有 TRDOCR 寄存器的 TOC0 位有效。

因为不能在开始计数后直接写 TRDGRB0、TRDGRA1、TRDGRB1 寄存器（禁止事项），所以必须将 TRDGRD0、TRDGRC1、TRDGRD1 寄存器用作缓冲寄存器。

但是，要写 TRDGRD0、TRDGRC1、TRDGRD1 寄存器时，在将 TRDBFD0 位、TRDBFC1 位和 TRDBFD1 位置“0”（通用寄存器）后写这些寄存器。此后，能将 TRDBFD0 位、TRDBFC1 位和 TRDBFD1 位置“1”（缓冲寄存器）。

[PWM3 模式]

必须以 16 位而不能以 8 位为单位存取 TRDGRAi ~ TRDGRDi 寄存器。

在 PWM3 模式中，以下寄存器无效：

TRDPMR、TRDDF0、TRDDF1、TRDIORA0、TRDIORC0、TRDPOCR0、TRDIORA1、TRDIORC1、TRDPOCR1

在 PWM3 模式中，不使用 TRDGRC0、TRDGRC1、TRDGRD0、TRDGRD1 寄存器。但是，要将这些寄存器用作缓冲寄存器时，在将 TRDBFC0 位、TRDBFC1 位、TRDBFD0 位和 TRDBFD1 位置“0”（通用寄存器）后给 TRDGRC0、TRDGRC1、TRDGRD0、TRDGRD1 寄存器写值。此后，能将 TRDBFC0 位、TRDBFC1 位、TRDBFD0 位和 TRDBFD1 位置“1”（缓冲寄存器）。

图 8-34 定时器 RD 通用寄存器 Ai、Bi、Ci、Di
(TRDGRAi、TRDGRBi、TRDGRCi、TRDGRDi) (i=0、1) 的格式 [输入捕捉功能]

地址：F0278H (TRDGRA0)、F027AH (TRDGRB0)、复位后：FFFFH R/W

FFF58H (TRDGRC0)、FFF5AH (TRDGRD0)、

F0288H (TRDGRA1)、F028AH (TRDGRB1)、

FFF5CH (TRDGRC1)、FFF5EH (TRDGRD1)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDGRAi																
TRDGRBi																
TRDGRCi																
TRDGRDi																

—	功能
bit15 ~ 0	参照“表 8-3 使用输入捕捉功能时的 TRDGRji 寄存器功能”。

表 8-3 使用输入捕捉功能时的 TRDGRji 寄存器功能

寄存器	设定	寄存器功能	输入捕捉的输入引脚
TRDGRAi	—	通用寄存器，能读输入捕捉时的 TRDi 寄存器的值。	TRDIOAi
TRDGRBi			TRDIOBi
TRDGRCi	TRDBFCi=0	通用寄存器，能读输入捕捉时的 TRDi 寄存器的值。	TRDIOCi
TRDGRDi	TRDBFDi=0		TRDIODi
TRDGRCi	TRDBFCi=1	通用寄存器，能读输入捕捉时的 TRDi 寄存器的值 (参照“8.4.2 缓冲器运行”)。	TRDIOAi
TRDGRDi	TRDBFDi=1		TRDIOBi

备注 i=0、1, j=A、B、C、D

TRDBFCi、TRDBFDi: TRDMR 寄存器的位

图 8-35 定时器 RD 通用寄存器 Ai、Bi、Ci、Di
(TRDGRAi、TRDGRBi、TRDGRCi、TRDGRDi) (i=0、1) 的格式 [输出比较功能]

地址: F0278H (TRDGRA0)、F027AH (TRDGRB0)、 复位后: FFFFH R/W
 FFF58H (TRDGRC0)、FFF5AH (TRDGRD0)、
 F0288H (TRDGRA1)、F028AH (TRDGRB1)、
 FFF5CH (TRDGRC1)、FFF5EH (TRDGRD1)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDGRAi																
TRDGRBi																
TRDGRCi																
TRDGRDi																

—	功能
bit15 ~ 0	参照“表 8-4 使用输出比较功能时的 TRDGRji 寄存器功能”。

表 8-4 使用输出比较功能时的 TRDGRji 寄存器功能

寄存器	设定		寄存器功能	输出比较的输出引脚
	TRDBFki	IOj3		
TRDGRAi	—	—	通用寄存器，必须写比较值。	TRDIOAi
TRDGRBi				TRDIOBi
TRDGRCi	0	1	通用寄存器，必须写比较值。	TRDIOCi
TRDGRDi				TRDIODi
TRDGRCi	1	1	缓冲寄存器，必须写下一个比较值 (参照“8.4.2 缓冲器运行”)。	TRDIOAi
TRDGRDi				TRDIOBi
TRDGRCi	0	0	TRDIOAi 输出控制	TRDIOAi
TRDGRDi			TRDIOBi 输出控制	TRDIOBi

注意 如果将 TRDCRi 寄存器的 TCK2 ~ TCK0 位置“000B” (f_{CLK} 、 f_{HOCO}) 并且将比较值置“0000H”，就只在开始计数后立即向 DTC 和 ELC 产生 1 次请求信号。如果比较值大于等于“0001H”，就在每次比较匹配时产生请求信号。

备注 i=0、1, j=A、B、C、D, k=C、D
 TRDBFji: TRDMR 寄存器的位, IOj3: TRDIORCi 寄存器的位

图 8-36 定时器 RD 通用寄存器 Ai、Bi、Ci、Di
(TRDGRAi、TRDGRBi、TRDGRCi、TRDGRDi) (i=0、1) 的格式 [PWM 功能]

地址: F0278H (TRDGRA0)、F027AH (TRDGRB0)、 复位后: FFFFH R/W
 FFF58H (TRDGRC0)、FFF5AH (TRDGRD0)、
 F0288H (TRDGRA1)、F028AH (TRDGRB1)、
 FFF5CH (TRDGRC1)、FFF5EH (TRDGRD1)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDGRAi																
TRDGRBi																
TRDGRCi																
TRDGRDi																

—	功能
bit15 ~ 0	参照“表 8-5 使用 PWM 功能时的 TRDGRji 寄存器功能”。

表 8-5 使用 PWM 功能时的 TRDGRji 寄存器功能

寄存器	设定	寄存器功能	PWM 输出引脚
TRDGRAi	—	通用寄存器，必须设定 PWM 周期。	—
TRDGRBi	—	通用寄存器，必须设定 PWM 输出的变化点。	TRDIOBi
TRDGRCi	TRDBFCi=0	通用寄存器，必须设定 PWM 输出的变化点。	TRDIOCi
TRDGRDi	TRDBFDi=0		TRDIODi
TRDGRCi	TRDBFCi=1	缓冲寄存器，必须设定下一个 PWM 周期 (参照“8.4.2 缓冲器运行”)。	—
TRDGRDi	TRDBFDi=1	缓冲寄存器，必须设定下一个 PWM 输出的变化点 (参照“8.4.2 缓冲器运行”)。	TRDIOBi

注意 如果将 TRDCRi 寄存器的 TCK2 ~ TCK0 位置“000B” (f_{CLK} 、 f_{HOCO}) 并且将比较值置“0000H”，就只在开始计数后立即向 DTC 和 ELC 产生 1 次请求信号。如果比较值大于等于“0001H”，就在每次比较匹配时产生请求信号。

备注 i=0、1, j=A、B、C、D
 TRDBFCi、TRDBFDi: TRDMR 寄存器的位

图 8-37 定时器 RD 通用寄存器 Ai、Bi、Ci、Di
(TRDGRAi、TRDGRBi、TRDGRCi、TRDGRDi) (i=0、1) 的格式 [复位同步 PWM 模式]

地址: F0278H (TRDGRA0)、F027AH (TRDGRB0)、 复位后: FFFFH R/W
FFF58H (TRDGRC0)、FFF5AH (TRDGRD0)、
F0288H (TRDGRA1)、F028AH (TRDGRB1)、
FFF5CH (TRDGRC1)、FFF5EH (TRDGRD1)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDGRAi																
TRDGRBi																
TRDGRCi																
TRDGRDi																

—	功能
bit15 ~ 0	参照“表 8-6 复位同步 PWM 模式中的 TRDGRji 寄存器功能”。

表 8-6 复位同步 PWM 模式中的 TRDGRji 寄存器功能

寄存器	设定	寄存器功能	PWM 输出引脚
TRDGRA0	—	通用寄存器，必须设定 PWM 周期。	(TRDIOC0 在每个 PWM 周期进行反相输出)
TRDGRB0	—	通用寄存器，必须设定 PWM1 输出的变化点。	TRDIOB0 TRDIOD0
TRDGRC0	TRDBFC0=0	(在复位同步 PWM 模式中不使用)	—
TRDGRD0	TRDBFD0=0		
TRDGRA1	—	通用寄存器，必须设定 PWM2 输出的变化点。	TRDIOA1 TRDIOC1
TRDGRB1	—	通用寄存器，必须设定 PWM3 输出的变化点。	TRDIOB1 TRDIOD1
TRDGRC1	TRDBFC1=0	(在复位同步 PWM 模式中不使用)	—
TRDGRD1	TRDBFD1=0		
TRDGRC0	TRDBFC0=1	缓冲寄存器，必须设定下一个 PWM 周期 (参照“8.4.2 缓冲器运行”)。	(TRDIOC0 在每个 PWM 周期进行反相输出)
TRDGRD0	TRDBFD0=1	缓冲寄存器，必须设定下一个 PWM1 的变化点 (参照“8.4.2 缓冲器运行”)。	TRDIOB0 TRDIOD0
TRDGRC1	TRDBFC1=1	缓冲寄存器，必须设定下一个 PWM2 的变化点 (参照“8.4.2 缓冲器运行”)。	TRDIOA1 TRDIOC1
TRDGRD1	TRDBFD1=1	缓冲寄存器，必须设定下一个 PWM3 的变化点 (参照“8.4.2 缓冲器运行”)。	TRDIOB1 TRDIOD1

注意 如果将 TRDCR0 寄存器的 TCK2 ~ TCK0 位置“000B”(f_{CLK} 、 f_{HOCO})并且将比较值置“0000H”，就只在开始计数后立即向 DTC 和 ELC 产生 1 次请求信号。如果比较值大于等于“0001H”，就在每次比较匹配时产生请求信号。

备注 i=0、1, j=A、B、C、D

TRDBFC0、TRDBFD0、TRDBFC1、TRDBFD1: TRDMR 寄存器的位

图 8-38 定时器 RD 通用寄存器 Ai、Bi、Ci、Di
(TRDGRAi、TRDGRBi、TRDGRCi、TRDGRDi) (i=0、1) 的格式 [互补 PWM 模式]

地址: F0278H (TRDGRA0)、F027AH (TRDGRB0)、 复位后: FFFFH R/W
 FFF58H (TRDGRC0)、FFF5AH (TRDGRD0)、
 F0288H (TRDGRA1)、F028AH (TRDGRB1)、
 FFF5CH (TRDGRC1)、FFF5EH (TRDGRD1)

符号

1514131211109876543210

TRDGRAi

TRDGRBi

TRDGRCi

TRDGRDi

—	功能
bit15 ~ 0	参照“表 8-7 互补 PWM 模式中的 TRDGRji 寄存器功能”。

表 8-7 互补 PWM 模式中的 TRDGR*ji* 寄存器功能

寄存器	设定	寄存器功能	PWM 输出引脚
TRDGRA0	—	通用寄存器，必须在初始设定时设定 PWM 周期。 设定范围：TRD0 寄存器的设定值（计数初始值） $\leq$ TRDGRA0 的设定值 $\leq$ FFFFH-TRD0 寄存器的设定值 当 TRDSTR 寄存器的 TSTART0 位和 TSTART1 位为“1”（开始计数）时，不能写此寄存器。	（TRDIOC0 在每半个周期进行反相输出）
TRDGRB0	—	通用寄存器，必须在初始设定时设定 PWM1 输出的变化点。 设定范围：TRD0 寄存器的设定值（计数初始值） $\leq$ TRDGRB0 的设定值 $\leq$ TRDGRA0 的设定值-TRD0 寄存器的设定值 当 TRDSTR 寄存器的 TSTART0 位和 TSTART1 位为“1”（开始计数）时，不能写此寄存器。	TRDIOB0 TRDIOD0
TRDGRA1	—	通用寄存器，必须在初始设定时设定 PWM2 输出的变化点。 设定范围：TRD0 寄存器的设定值（计数初始值） $\leq$ TRDGRA1 的设定值 $\leq$ TRDGRA0 的设定值-TRD0 寄存器的设定值 当 TRDSTR 寄存器的 TSTART0 位和 TSTART1 位为“1”（开始计数）时，不能写此寄存器。	TRDIOA1 TRDIOC1
TRDGRB1	—	通用寄存器，必须在初始设定时设定 PWM3 输出的变化点。 设定范围：TRD0 寄存器的设定值（计数初始值） $\leq$ TRDGRB1 的设定值 $\leq$ TRDGRA0 的设定值-TRD0 寄存器的设定值 当 TRDSTR 寄存器的 TSTART0 位和 TSTART1 位为“1”（开始计数）时，不能写此寄存器。	TRDIOB1 TRDIOD1
TRDGRC0	—	（在互补 PWM 模式中不使用）	—
TRDGRD0	TRDBFD0=1	缓冲寄存器，必须设定下一个 PWM1 输出的变化点 （参照“8.4.2 缓冲器运行”）。 设定范围：TRD0 寄存器的设定值（计数初始值） $\leq$ TRDGRD0 的设定值 $\leq$ TRDGRA0 的设定值-TRD0 寄存器的设定值 在初始设定时必须设定 TRDGRB0 寄存器的相同值。	TRDIOB0 TRDIOD0
TRDGRC1	TRDBFC1=1	缓冲寄存器，必须设定下一个 PWM2 输出的变化点 （参照“8.4.2 缓冲器运行”）。 设定范围：TRD0 寄存器的设定值（计数初始值） $\leq$ TRDGRC1 的设定值 $\leq$ TRDGRA0 的设定值-TRD0 寄存器的设定值 在初始设定时必须设定 TRDGRA1 寄存器的相同值。	TRDIOA1 TRDIOC1
TRDGRD1	TRDBFD1=1	缓冲寄存器，必须设定下一个 PWM3 输出的变化点 （参照“8.4.2 缓冲器运行”）。 设定范围：TRD0 寄存器的设定值（计数初始值） $\leq$ TRDGRD1 的设定值 $\leq$ TRDGRA0 的设定值-TRD0 寄存器的设定值 在初始设定时必须设定 TRDGRB1 寄存器的相同值。	TRDIOB1 TRDIOD1

注意 如果将 TRDCR*i* 寄存器的 TCK2 ~ TCK0 位置“000B”（ f_{CLK} 、 f_{HOCO} ）并且将比较值置“0000H”，就只在开始计数后立即向 DTC 和 ELC 产生 1 次请求信号。如果比较值大于等于“0001H”，就在每次比较匹配时产生请求信号。

备注 $i=0, 1$, $j=A, B, C, D$
TRDBFD0、TRDBFC1、TRDBFD1: TRDMR 寄存器的位

图 8-39 定时器 RD 通用寄存器 Ai、Bi、Ci、Di
(TRDGRAi、TRDGRBi、TRDGRCi、TRDGRDi) (i=0、1) 的格式 [PWM3 模式]

地址: F0278H (TRDGRA0)、F027AH (TRDGRB0)、 复位后: FFFFH R/W
 FFF58H (TRDGRC0)、FFF5AH (TRDGRD0)、
 F0288H (TRDGRA1)、F028AH (TRDGRB1)、
 FFF5CH (TRDGRC1)、FFF5EH (TRDGRD1)

符号

1514131211109876543210

TRDGRAi

TRDGRBi

TRDGRCi

TRDGRDi

—	功能
bit15 ~ 0	参照“表 8-8 PWM3 模式中的 TRDGRji 寄存器功能”。

表 8-8 PWM3 模式中的 TRDGR*ji* 寄存器功能

寄存器	设定	寄存器功能	PWM 输出引脚
TRDGRA0	—	通用寄存器，必须设定 PWM 周期。 设定范围：TRDGRA1 寄存器的设定值 ≤ TRDGRA0 的设定值	TRDIOA0
TRDGRA1		通用寄存器，必须设定 PWM 输出的变化点（变为有效电平的时序）。 设定范围：TRDGRA1 的设定值 ≤ TRDGRA0 寄存器的设定值	
TRDGRB0		通用寄存器，必须设定 PWM 输出的变化点（返回到初始输出电平的时序）。 设定范围：TRDGRB1 寄存器的设定值 ≤ TRDGRB0 的设定值 ≤ TRDGRA0 寄存器的设定值	TRDIOB0
TRDGRB1		通用寄存器，必须设定 PWM 输出的变化点（变为有效电平的时序）。 设定范围：TRDGRB1 的设定值 ≤ TRDGRB0 寄存器的设定值	
TRDGRC0	TRDBFC0=0	(在 PWM3 模式中不使用)	—
TRDGRC1	TRDBFC1=0		
TRDGRD0	TRDBFD0=0		
TRDGRD1	TRDBFD1=0		
TRDGRC0	TRDBFC0=1	缓冲寄存器，必须设定下一个 PWM 周期 (参照“8.4.2 缓冲器运行”)。 设定范围：TRDGRC1 寄存器的设定值 ≤ TRDGRC0 的设定值	TRDIOA0
TRDGRC1	TRDBFC1=1	缓冲寄存器，必须设定下一个 PWM 输出的变化点 (参照“8.4.2 缓冲器运行”)。 设定范围：TRDGRC1 的设定值 ≤ TRDGRC0 寄存器的设定值	
TRDGRD0	TRDBFD0=1	缓冲寄存器，必须设定下一个 PWM 输出的变化点 (参照“8.4.2 缓冲器运行”)。 设定范围：TRDGRD1 寄存器 ≤ TRDGRD0 的设定值 ≤ TRDGRC0 寄存器的设定值	TRDIOB0
TRDGRD1	TRDBFD1=1	缓冲寄存器，必须设定下一个 PWM 输出的变化点 (参照“8.4.2 缓冲器运行”)。 设定范围：TRDGRD1 的设定值 ≤ TRDGRD0 寄存器的设定值	

注意 如果将 TRDCR0 寄存器的 TCK2 ~ TCK0 位置“000B”(f_{CLK} 、 f_{HOCO})并且将比较值置“0000H”，就只在开始计数后立即向 DTC 和 ELC 产生 1 次请求信号。如果比较值大于等于“0001H”，就在每次比较匹配时产生请求信号。

备注 $i=0, 1$, $j=A, B, C, D$

TRDBFC0、TRDBFD0、TRDBFC1、TRDBFD1: TRDMR 寄存器的位

8.3.20 端口模式寄存器 1 (PM1)

这是以位为单位设定端口 1 的输入 / 输出的寄存器。

在将定时器输出引脚的复用端口 (P10/TRDIOD1、P11/TRDIOC1 等) 用作定时器输出时, 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”。

(例) P10/TRDIOD1 用作定时器输出的情况

将端口模式寄存器 1 的 PM10 位置 “0”。

将端口寄存器 1 的 P10 位置 “0”。

在将定时器输入引脚的复用端口 (P10/TRDIOD1、P11/TRDIOC1 等) 用作定时器输入时, 必须将各端口对应的端口模式寄存器 (PMxx) 的位置 “1”。此时, 端口寄存器 (Pxx) 的位可以是 “0” 或者 “1”。

(例) P10/TRDIOD1 用作定时器输入的情况

将端口模式寄存器 1 的 PM10 位置 “1”。

将端口寄存器 1 的 P10 位置 “0” 或者 “1”。

通过 1 位或者 8 位存储器操作指令设定 PM1 寄存器。

在产生复位信号后, 此寄存器的值变为 “FFH”。

图 8-40 端口模式寄存器 1 (PM1) 的格式 (64 引脚产品)

地址：FFF21H

复位后：FFH

R/W

符号

7

6

5

4

3

2

1

0

PM1

PM17

PM16

PM15

PM14

PM13

PM12

PM11

PM10

PMmn	Pmn 引脚的输入 / 输出模式的选择 （m=1, 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）

备注 上述格式是 64 引脚产品的端口模式寄存器 1 的格式。有关其他产品的端口模式寄存器的格式, 请参照 “表 4-14 各产品分配的 PMxx、Pxx、PUxx、PIMxx、POMxx、PMCxx 寄存器及其位”。

8.4 有关多个模式的共同事项

8.4.1 计数源

所有模式的计数源的选择方法都一样。但是，在 PWM3 模式中，不能选择外部时钟。

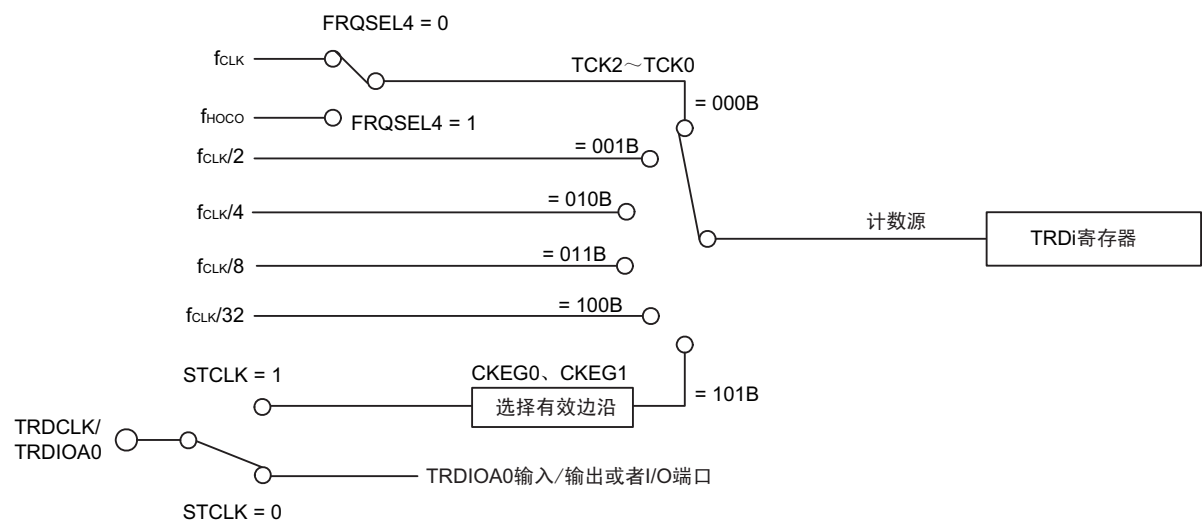
表 8-9 计数源的选择

计数源	选择方法
f_{CLK} 、 f_{HOCO} 注、 $f_{CLK}/2$ 、 $f_{CLK}/4$ 、 $f_{CLK}/8$ 、 $f_{CLK}/32$	通过 TRDCR <i>i</i> 寄存器的 TCK2 ~ TCK0 位选择计数源。
TRDCLK 引脚的外部输入信号	TRDFCR 寄存器的 STCLK 位为“1”（外部时钟输入有效）。 TRDCR <i>i</i> 寄存器的 TCK2 ~ TCK0 位为“101B”（计数源为外部时钟）。 通过 TRDCR <i>i</i> 寄存器的 CKEG1 ~ CKEG0 位选择有效边沿。 TRDCLK 引脚的复用 I/O 端口的端口模式寄存器的位为“1”（输入模式）。

备注 i=0、1

注 当用户选项字节（000C2H）的 FRQSEL4 位为“0”时，选择 f_{CLK} ；当 FRQSEL4 位为“1”时，选择 f_{HOCO} 。要选择 f_{HOCO} 作为定时器 RD 的计数源时，必须在将外围允许寄存器 1（PER1）的 bit4（TRD0EN）置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟，就必须在清除外围允许寄存器 1（PER1）的 bit4（TRD0EN）后进行更改。

图 8-41 计数源的框图



备注 i=0、1

TCK0~TCK2、CKEG0~CKEG1: TRDCR*i*寄存器的位
STCLK: TRDFCR寄存器的位
FRQSEL4: 用户选项字节(000C2H/010C2H)的位

TRDCLK 引脚的外部输入时钟的脉宽必须至少为 3 个定时器 RD 的运行时钟（ f_{CLK} ）周期。

8.4.2 缓冲器运行

能通过 TRDMR 寄存器的 TRDBFCi (i=0、1) 位和 TRDBFDi 位，将 TRDGRCi 寄存器和 TRDGRDi 寄存器分别设定为 TRDGRAi 寄存器和 TRDGRBi 寄存器的缓冲寄存器。

- TRDGRAi 的缓冲寄存器：TRDGRCi 寄存器
- TRDGRBi 的缓冲寄存器：TRDGRDi 寄存器

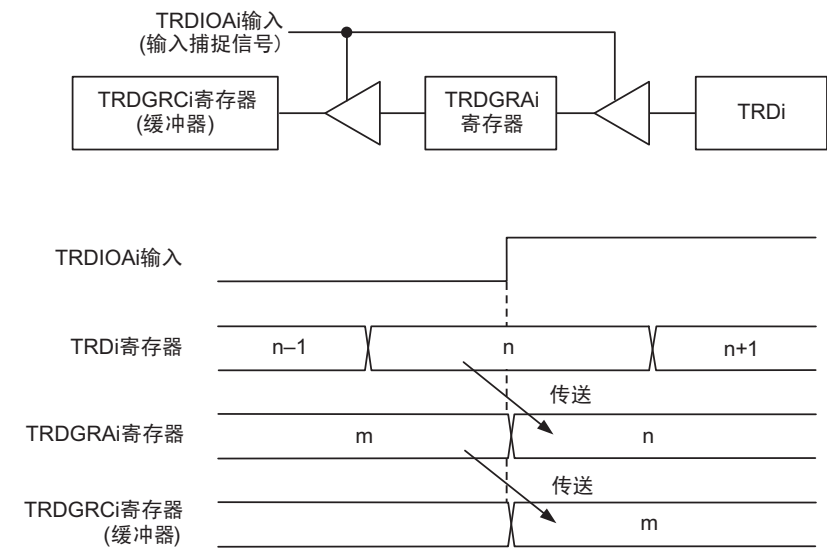
缓冲器运行因定时器模式而不同，各模式的缓冲器运行如表 8-10 所示。

表 8-10 各模式的缓冲器运行

功能和模式		传送时序	传送的寄存器
定时器模式	输入捕捉功能	TRDIOAi 的输入信号 (输入捕捉信号的输入)	将 TRDGRAi 寄存器的内容传送到 TRDGRCi 寄存器（缓冲寄存器）。
		TRDIOBi 的输入信号 (输入捕捉信号的输入)	将 TRDGRBi 寄存器的内容传送到 TRDGRDi 寄存器（缓冲寄存器）。
	输出比较功能	TRDi 寄存器和 TRDGRAi 寄存器的比较匹配	将 TRDGRCi 寄存器（缓冲寄存器）的内容传送到 TRDGRAi 寄存器。
		TRDi 寄存器和 TRDGRBi 寄存器的比较匹配	将 TRDGRDi 寄存器（缓冲寄存器）的内容传送到 TRDGRBi 寄存器。
	PWM 功能	TRDi 寄存器和 TRDGRAi 寄存器的比较匹配	将 TRDGRCi 寄存器（缓冲寄存器）的内容传送到 TRDGRAi 寄存器。
		TRDi 寄存器和 TRDGRBi 寄存器的比较匹配	将 TRDGRDi 寄存器（缓冲寄存器）的内容传送到 TRDGRBi 寄存器。
复位同步 PWM 模式		TRD0 寄存器和 TRDGRA0 寄存器的比较匹配	将 TRDGRCi 寄存器（缓冲寄存器）的内容传送到 TRDGRAi 寄存器。 将 TRDGRDi 寄存器（缓冲寄存器）的内容传送到 TRDGRBi 寄存器。
互补 PWM 模式		• 当 TRDFCR 寄存器的 CMD1 位和 CMD0 位是“11B”时，为 TRD1 寄存器的下溢。 • 当 TRDFCR 寄存器的 CMD1 位和 CMD0 位是“10B”时，为 TRD0 寄存器和 TRDGRA0 寄存器的比较匹配。	将 TRDGRC1 寄存器（缓冲寄存器）的内容传送到 TRDGRA1 寄存器。 将 TRDGRDi 寄存器（缓冲寄存器）的内容传送到 TRDGRBi 寄存器。
PWM3 模式		TRD0 寄存器和 TRDGRA0 寄存器的比较匹配	将 TRDGRCi 寄存器（缓冲寄存器）的内容传送到 TRDGRAi 寄存器。 将 TRDGRDi 寄存器（缓冲寄存器）的内容传送到 TRDGRBi 寄存器。

备注 i=0、1

图 8-42 输入捕捉功能的缓冲器运行

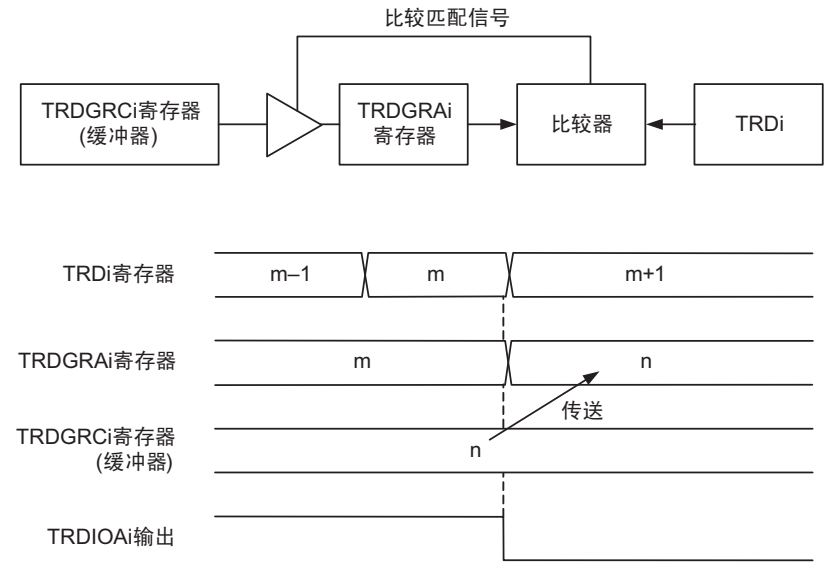


备注 i=0、1

上图的条件如下：

- TRDMR寄存器的TRDBFCi位为“1”(TRDGRci寄存器为TRDGRAi寄存器的缓冲寄存器)。
- TRDIOAi寄存器的IOA2～IOA0位为“100B”(在上升沿输入捕捉)。

图 8-43 输出比较功能的缓冲器运行



备注 i=0、1

上图的条件如下：

- TRDMR寄存器的TRDBFCi位为“1”(TRDGRci寄存器为TRDGRAi寄存器的缓冲寄存器)。
- TRDIOAi寄存器的IOA2～IOA0位为“001B”(在比较匹配时，输出“L”电平)。

在定时器模式（输入捕捉功能和输出比较功能）中，必须进行以下的设定。

将 TRDGRCi (i=0、1) 寄存器用作 TRDGRAi 寄存器的缓冲寄存器情况：

- 必须将 TRDIORCi 寄存器的 IOC3 位置“1”（通用寄存器或者缓冲寄存器）。
- 必须给 TRDIORCi 寄存器的 IOC2 位和 TRDIORAi 寄存器的 IOA2 位设定相同的值。

将 TRDGRDi 寄存器用作 TRDGRBi 寄存器的缓冲寄存器的情况：

- 必须将 TRDIORCi 寄存器的 IOD3 位置“1”（通用寄存器或者缓冲寄存器）。
- 必须给 TRDIORCi 寄存器的 IOD2 位和 TRDIORAi 寄存器的 IOB2 位设定相同的值。

在使用输入捕捉功能时，即使将 TRDGRCi 寄存器和 TRDGRDi 寄存器用作缓冲寄存器，在 TRDIOCi 引脚和 TRDIODi 引脚的输入边沿，TRDSRi 寄存器的 IMFC 位和 IMFD 位也变为“1”。

当使用输出比较功能或者 PWM 功能时或者在复位同步 PWM 模式、互补 PWM 模式和 PWM3 模式中，即使将 TRDGRCi 寄存器和 TRDGRDi 寄存器用作缓冲寄存器，在和 TRDi 寄存器比较匹配时，TRDSRi 寄存器的 IMFC 位和 IMFD 位也变为“1”。

8.4.3 同步运行

使 TRD0 寄存器和 TRD1 寄存器同步。

- 同步预置

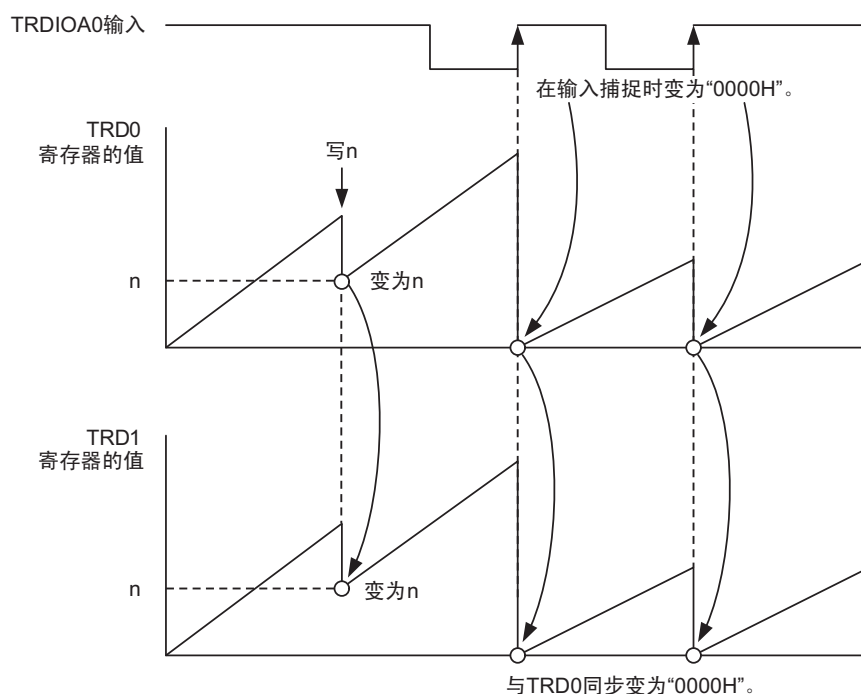
如果在 TRDMR 寄存器的 TRDSYNC 位为“1”（同步运行）时写 TRDi 寄存器，数据就同时被写到 TRD0 寄存器和 TRD1 寄存器。

- 同步清除

当 TRDMR 寄存器的 TRDSYNC 位为“1”并且 TRDCR0 寄存器的 CCLR2～CCLR0 位为“011B”（同步清除）时，TRD0 寄存器和 TRD1 寄存器同时变为“0000H”。

同样，当 TRDMR 寄存器的 TRDSYNC 位为“1”并且 TRDCR1 寄存器的 CCLR2～CCLR0 位为“011B”（同步清除）时，TRD1 寄存器和 TRD0 寄存器同时变为“0000H”。

图 8-44 同步运行



上图的条件如下：

- TRDMR寄存器的TRDSYNC位为“1”(同步运行)。
 - TRDCR0寄存器的CCLR2～CCLR0位为“001B”(在输入捕捉时，将TRD0置“0000H”)。
 - TRDCR1寄存器的CCLR2～CCLR0位为“011B”(与TRD0同步，将TRD1置“0000H”)。
 - TRDIOA0寄存器的IOA2～IOA0位为“100B”。
 - TRDFCR寄存器的CMD1位和CMD0位为“00B”。
- TRDFCR寄存器的PWM3位为“1”。
- （在TRDIOA0输入的上升沿输入捕捉）

8.4.4 脉冲输出的强制截止

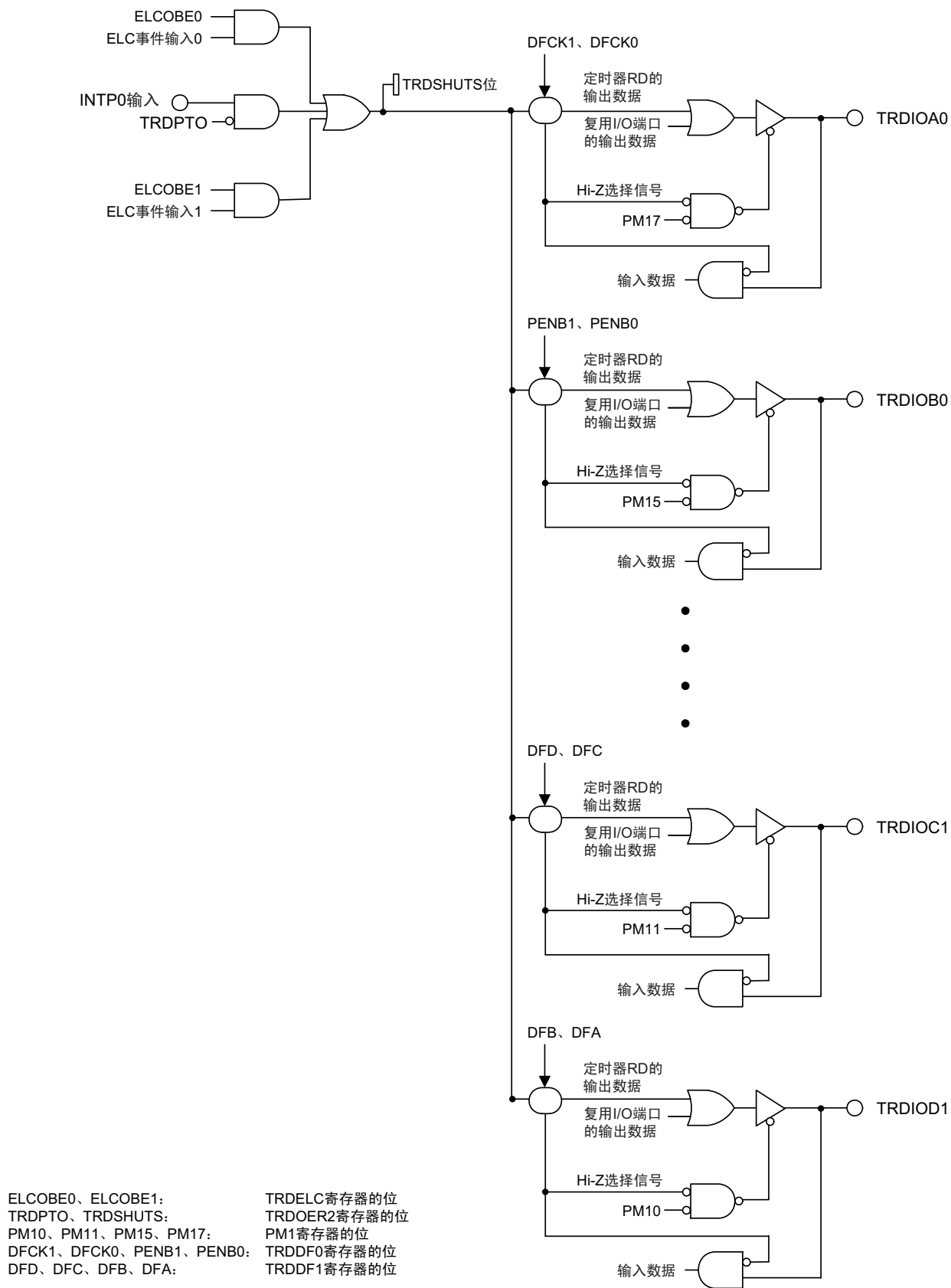
当使用 PWM 功能时或者在复位同步 PWM 模式、互补 PWM 模式和 PWM3 模式中，能通过 INTP0 引脚的输入来截止 TRDIO_{ji} 输出引脚（ $i=0, 1$ ， $j=A, B, C, D$ ）的脉冲输出。

如果将 TRDOER1 寄存器的对应位置“0”（允许定时器 RD 的输出），就将这些功能或者模式中使用的输出引脚用作定时器 RD 的输出引脚。当 TRDOER2 寄存器的 TRDP_{TO} 位为“1”（脉冲输出强制截止信号 INTP0 引脚输入有效）时，TRDDF0 寄存器或者 TRDDF1 寄存器的 DFCK1、DFCK0、PENB1、PENB0、DFD、DFC、DFB、DFA 位设定的输出值从用作定时器 RD 输出端口的输出引脚输出。

在使用此功能时，必须进行以下的设定：

- 通过 TRDDFi 寄存器设定脉冲输出被强制截止时的引脚状态（高阻抗、“L”电平输出或者“H”电平输出）。
- 有关通过 ELC 事件输入进行的脉冲强制截止，请参照“8.4.5 从事件链接控制器（ELC）输入的事件”。
- 在强制截止脉冲的输出时，TRDOER2 寄存器的 TRDSHUTS 位为“1”。要中止强制截止脉冲的输出时，必须在停止计数的过程中（TSTART_i=0）将 TRDSHUTS 位置“0”。
- 将 TRDOER2 寄存器的 TRDP_{TO} 位置“1”（脉冲输出强制截止信号 INTP0 引脚输入有效）。

图 8-45 脉冲输出的强制截止



8.4.5 从事件链接控制器（ELC）输入的事件

通过 ELC 输入的事件，定时器 RD 进行 2 种运行。

(a) TRDIOD0/TRDIOD1 的输入捕捉

通过 ELC 输入的事件，定时器 RD 进行 TRDIOD0/TRDIOD1 的输入捕捉。此时，TRDSRi 寄存器的 IMFD 位为“1”。

要使用此功能时，必须选择定时器模式的输入捕捉功能，并且将 TRDELIC 寄存器的 ELCICE0 位或者 ELCICE1 位置“1”。在其他模式（定时器模式的输出比较功能、PWM 功能、复位同步 PWM 模式、互补 PWM 模式、PWM3 模式）中，此功能无效。

(b) 强制截止脉冲输出的运行注

通过 ELC 输入的事件，强制截止脉冲的输出。要使用此功能时，必须选择脉冲输出模式（PWM 功能、复位同步 PWM 模式、互补 PWM 模式、PWM3 模式）并且将 ELC0BE0 位或者 ELC0BE1 位置“1”。在使用定时器模式的输入捕捉功能时，此功能无效。

注 INTPO 引脚的强制截止功能截止“L”电平输入期间的脉冲输出，但是 ELC 事件的强制截止功能对于 ELC 输入的 1 次事件，截止 1 次脉冲输出。

设定步骤

- (1) 将 ELC 事件链接目标设定为定时器 RD。
- (2) 将 TRDELIC 寄存器的 ELCICEi 位（i=0、1）和 ELC0BEi 位（i=0、1）置“1”。

8.4.6 向事件链接控制器（ELC）/ 数据传送控制器（DTC）输出的事件

定时器 RD 的模式和向 ELC/DTC 输出的事件如表 8-11 所示。

表 8-11 定时器 RD 的模式和向 ELC/DTC 输出的事件

使用模式	输出源	ELC	DTC
输入捕捉功能	通过 TRDIOA0 寄存器的 IOA1 位和 IOA0 位设定的 TRDIOA0 边沿检测	○	○
	通过 TRDIOA0 寄存器的 IOB1 位和 IOB0 位设定的 TRDIOB0 边沿检测	○	○
	通过 TRDIORC0 寄存器的 IOC1 位和 IOC0 位设定的 TRDIOC0 边沿检测	—	○
	通过 TRDIORC0 寄存器的 IOD1 位和 IOD0 位设定的 TRDIOD0 边沿检测	—	○
	通过 TRDIOA1 寄存器的 IOA1 位和 IOA0 位设定的 TRDIOA1 边沿检测	○	○
	通过 TRDIOA1 寄存器的 IOB1 位和 IOB0 位设定的 TRDIOB1 边沿检测	○	○
	通过 TRDIORC1 寄存器的 IOC1 位和 IOC0 位设定的 TRDIOC1 边沿检测	—	○
	通过 TRDIORC1 寄存器的 IOD1 位和 IOD0 位设定的 TRDIOD1 边沿检测	—	○
输出比较功能、 PWM 功能、 复位同步 PWM 模式、 互补 PWM 模式、 PWM3 模式	TRD0 寄存器和 TRDGRA0 寄存器的比较匹配	○	○
	TRD0 寄存器和 TRDGRB0 寄存器的比较匹配	○	○
	TRD0 寄存器和 TRDGRC0 寄存器的比较匹配	—	○
	TRD0 寄存器和 TRDGRD0 寄存器的比较匹配	—	○
	TRD1 寄存器和 TRDGRA1 寄存器的比较匹配	○	○
	TRD1 寄存器和 TRDGRB1 寄存器的比较匹配	○	○
	TRD1 寄存器和 TRDGRC1 寄存器的比较匹配	—	○
	TRD1 寄存器和 TRDGRD1 寄存器的比较匹配	—	○
互补 PWM 模式	TRD1 寄存器的下溢	○	—

8.5 定时器 RD 的运行

8.5.1 输入捕捉功能

这是测量外部信号的宽度和周期的功能。以 TRDIOj_i 引脚 (i=0、1, j=A、B、C、D) 的外部信号为触发, 将 TRDi 寄存器 (计数器) 的内容传送到 TRDGRj_i 寄存器 (输入捕捉)。因为 TRDIOj_i 引脚和 TRDGRj_i 寄存器组合使用, 所以能按引脚选择为输入捕捉功能、或者其他模式和功能。

输入捕捉功能的框图和运行例子分别如图 8-46 和图 8-47 所示, 输入捕捉功能的规格如表 8-12 所示。

图 8-46 输入捕捉功能的框图

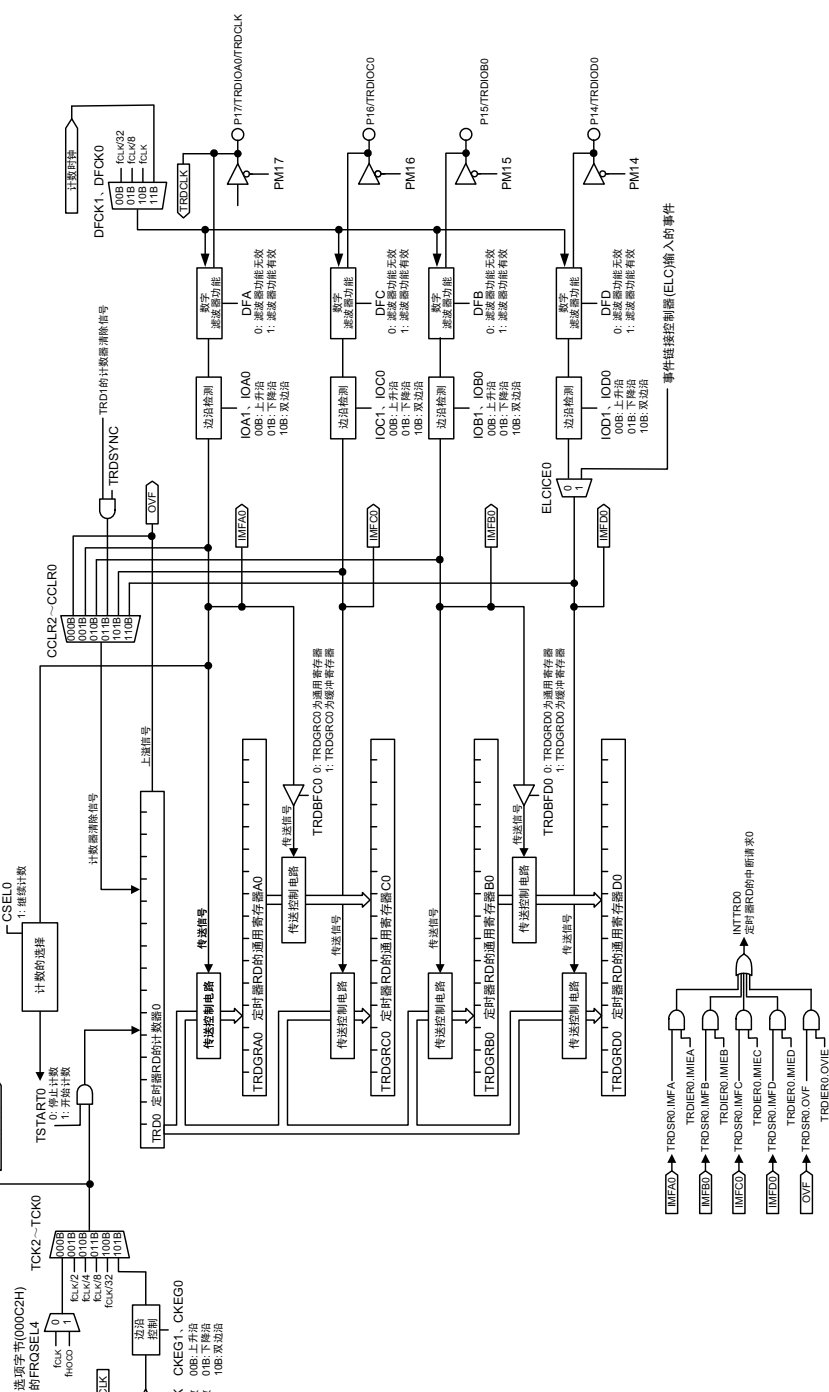


表 8-12 输入捕捉功能的规格

项目	规格
计数源	f_{HOCO} 注、 f_{CLK} 、 $f_{CLK}/2$ 、 $f_{CLK}/4$ 、 $f_{CLK}/8$ 、 $f_{CLK}/32$ TRDCLK 引脚的外部输入信号（通过程序选择有效边沿）
计数	递增计数
计数周期	当 TRDCRi 寄存器的 CCLR2 ~ CCLR0 位为“000B”（自由运行）时 $1/f_k \times 65536$ f_k : 计数源的频率
计数开始条件	给 TRDSTR 寄存器的 TSTARTi 位写“1”（开始计数）。
计数停止条件	当 TRDSTR 寄存器的 CSELi 位为“1”时，给 TSTARTi 位写“0”（停止计数）。
中断请求的发生时序	- 输入捕捉（TRDIOji 输入的有效边沿） - TRDi 的上溢
TRDIOA0 引脚功能	I/O 端口、输入捕捉的输入或者 TRDCLK（外部时钟）输入
TRDIOB0、TRDIOC0、 TRDIOD0、TRDIOA1 ~ TRDIOD1 引脚功能	I/O 端口或者输入捕捉的输入（按引脚进行选择）
INTP0 引脚功能	不使用（输入专用端口或者 INTP0 中断输入）。
读定时器	如果读 TRDi 寄存器，就能读到计数值。
写定时器	- 当 TRDMR 寄存器的 TRDSYNC 位为“0”（定时器 RD0 和定时器 RD1 独立运行）时能写 TRDi 寄存器。 - 当 TRDMR 寄存器的 TRDSYNC 位为“1”（定时器 RD0 和定时器 RD1 同步运行）时如果写 TRDi 寄存器，数据就同时被写到 TRD0 寄存器和 TRD1 寄存器。
选择功能	- 输入捕捉的输入引脚的选择 TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚中的 1 个或者多个引脚 - 输入捕捉的输入有效边沿的选择 上升沿、下降沿或者双边沿 - 将 TRDi 置“0000H”的时序 上溢或者输入捕捉 - 缓冲器运行（参照“8.4.2 缓冲器运行”） - 同步运行（参照“8.4.3 同步运行”） - 数字滤波器 对 TRDIOji 的输入进行采样，如果信号 3 次相同，就视为电平已确定。 - 通过 ELC 输入的事件进行的输入捕捉运行

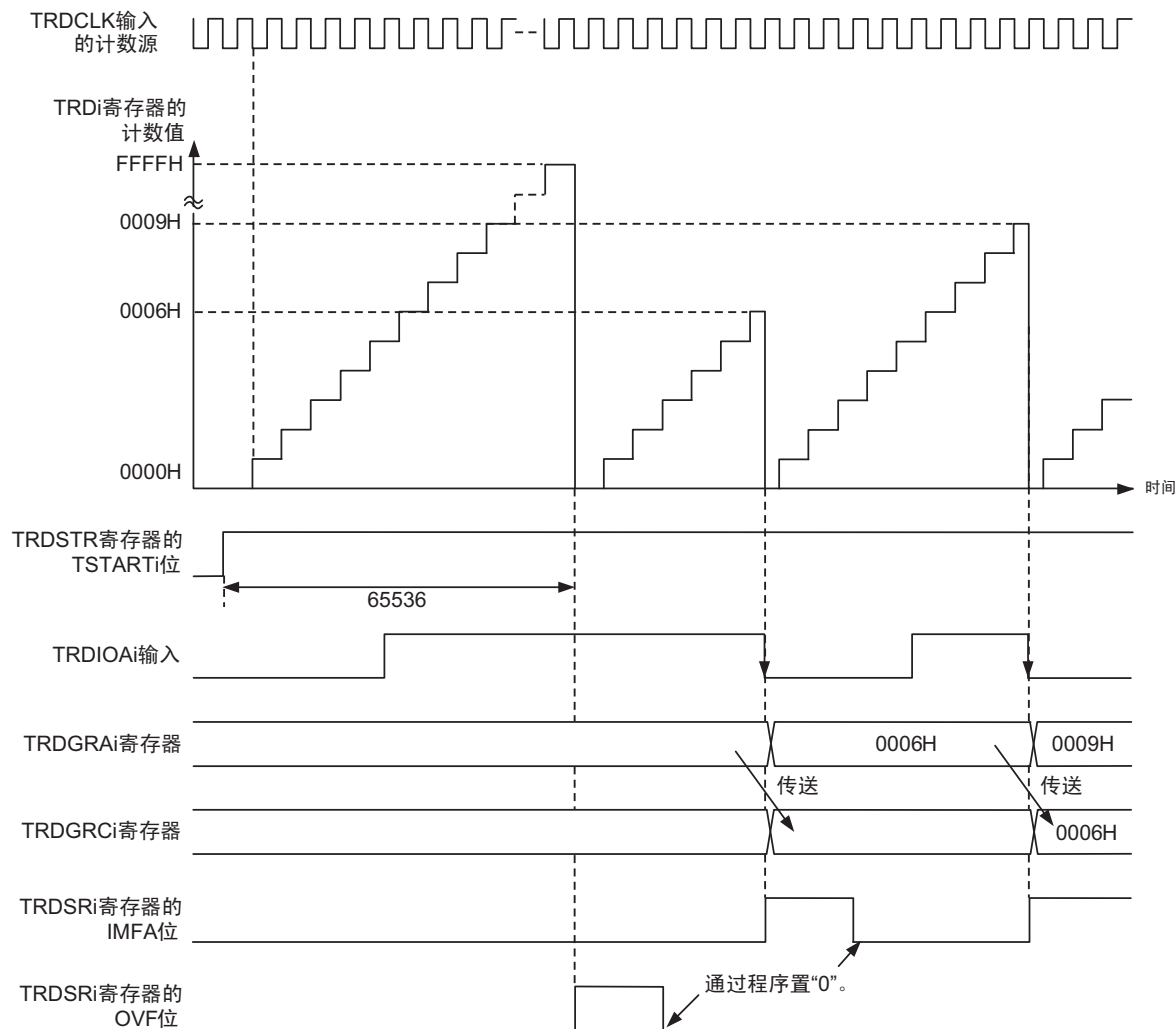
注 只有在用户选项字节（000C2H）的 FRQSEL4 位为“1”时才能选择 f_{HOCO} 。要选择 f_{HOCO} 作为定时器 RD 的计数源时，必须在将外围允许寄存器 1（PER1）的 bit4（TRD0EN）置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟，就必须在清除外围允许寄存器 1（PER1）的 bit4（TRD0EN）后进行更改。

备注 i=0、1, j=A、B、C、D

(1) 运行例子

通过设定 TRDCR_i 寄存器 (i=0、1) 的 CCLR0 ~ CCLR2 位, 在发生输入捕捉或者比较匹配时定时器 RD_i 的计数器的值进行复位。图 8-47 是将 CCLR2 ~ CCLR0 位置“001B”时的运行例子。如果设定为通过运行过程中的输入捕捉进行计数清除并且在定时器的计数值为“FFFFH”时进行输入捕捉, 就根据计数源和输入捕捉的运行时序, TRDSR_i 寄存器的 IMFA ~ IMF_D 位和 OVF 位的中断标志可能同时变为“1”。

图 8-47 输入捕捉功能的运行例子



备注 i=0、1

上图的条件如下:

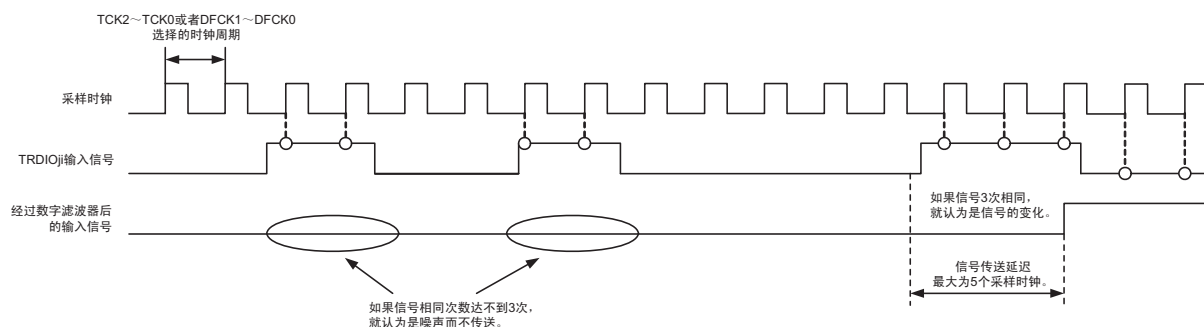
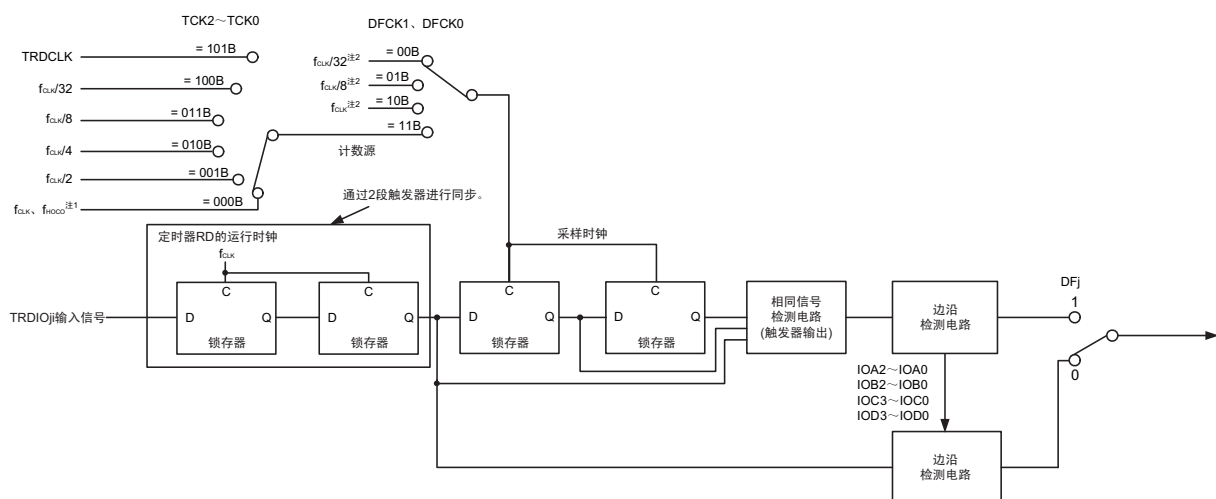
- TRDCR_i寄存器的CCLR2~CCLR0位为“001B”(在TRDGRA_i输入捕捉时, 将TRDi置“0000H”)。
- TRDCR_i寄存器的TCK2~TCK0位为“101B”(计数源为TRDCLK的输入)。
- TRDCR_i寄存器的CKEG1位和CKEG0位为“01B”(在计数源的下降沿进行计数)。
- TRDIORA_i寄存器的IOA2~IOA0位为“101B”(在TRDIOA_i输入的下降沿输入捕捉)。
- TRDMR寄存器的TRDBFC_i位为“1”(TRDGRC_i寄存器为TRDGRA_i寄存器的缓冲寄存器)。

(2) 数字滤波器

对 TRDIO_ji 输入 (i=0, 1, j=A、B、C、D) 进行采样, 如果信号 3 次相同, 就视为电平已确定。必须通过 TRDDFi 寄存器选择数字滤波器的功能和采样时钟。

数字滤波器的框图如图 8-48 所示。

图 8-48 数字滤波器的框图



备注 i=0, 1, j=A、B、C、D

TCK0~TCK2: TRDCRi寄存器的位
DFCK0、DFCK1、DFj: TRDDFi寄存器的位
IOA0~IOA2、IOB0~IOB2: TRDIOAi寄存器的位
IOC0~IOC3、IOD0~IOD3: TRDIORCi寄存器的位

- 注 1. 当用户选项字节 (000C2H) 的 FRQSEL4 位为“0”时, 选择 f_{CLK} ; 当 FRQSEL4 位为“1”时, 选择 f_{HOCO} 。
2. 当用户选项字节 (000C2H) 的 FRQSEL4 位为“1”时, $f_{CLK}/32$ 、 $f_{CLK}/8$ 和 f_{CLK} 分别为 $f_{HOCO}/32$ 、 $f_{HOCO}/8$ 和 f_{HOCO} 。

8.5.2 输出比较功能

这是检测 TRDi 寄存器（计数器）（ $i=0, 1$ ）的内容和 TRDGRji 寄存器（ $j=A, B, C, D$ ）的内容是否相同（比较匹配）的模式。如果内容相同，就从 TRDIOji 引脚输出任意的电平。因为 TRDIOji 引脚和 TRDGRji 寄存器组合使用，所以能按引脚选择为输出比较功能、或者其他模式和功能。

输出比较功能的框图和运行例子分别如图 8-49 和图 8-50 所示，输出比较功能的规格如表 8-13 所示。

图 8-49 输出比较功能的框图

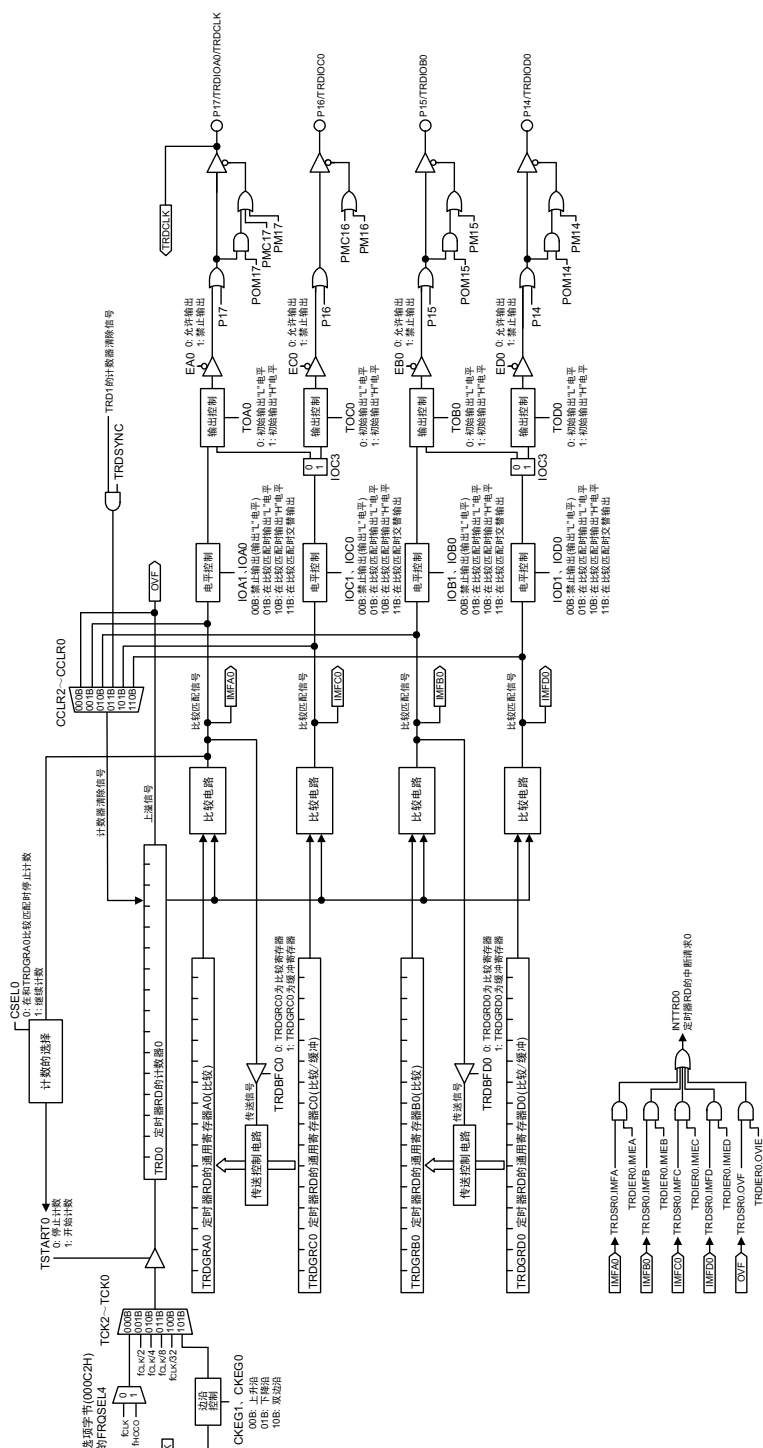


表 8-13 输出比较功能的规格

项目	规格
计数源	f_{HOCO} 注、 f_{CLK} 、 $f_{CLK}/2$ 、 $f_{CLK}/4$ 、 $f_{CLK}/8$ 、 $f_{CLK}/32$ TRDCLK 引脚的外部输入信号（通过程序选择有效边沿）
计数	递增计数
计数周期	- 当 TRDCRi 寄存器的 CCLR2 ~ CCLR0 位为“000B”（自由运行）时 $1/f_k \times 65536$ f_k: 计数源的频率 - 当 TRDCRi 寄存器的 CCLR1 ~ CCLR0 位为“01B”或者“10B”（在 TRDGRji 比较匹配时，将 TRDi 置“0000H”）时 $1/f_k \times (n+1)$ n: TRDGRji 寄存器的设定值
波形输出时序	比较匹配（TRDi 寄存器和 TRDGRji 寄存器的内容相同）
计数开始条件	给 TRDSTR 寄存器的 TSTARTi 位写“1”（开始计数）。
计数停止条件	- 当 TRDSTR 寄存器的 CSELi 位为“1”时，给 TSTARTi 位写“0”（停止计数）。 输出比较的输出引脚保持停止计数前的输出电平。 - 在 TRDSTR 寄存器的 CSELi 位为“0”并且发生 TRDGRAi 的比较匹配时停止计数。 输出比较的输出引脚保持比较匹配引起输出变化后的电平。
中断请求的发生时序	- 比较匹配（TRDi 寄存器和 TRDGRji 寄存器的内容相同） - TRDi 的上溢
TRDIOA0 引脚功能	I/O 端口、输出比较的输出或者 TRDCLK（外部时钟）输入
TRDIOB0、TRDIOC0、 TRDIOD0、TRDIOA1 ~ TRDIOD1 引脚功能	I/O 端口或者输出比较的输出（按引脚进行选择）
INTP0 引脚功能	不使用（输入专用端口或者 INTP0 中断输入）。
读定时器	如果读 TRDi 寄存器，就能读到计数值。
写定时器	- 当 TRDMR 寄存器的 TRDSYNC 位为“0”（定时器 RD0 和定时器 RD1 独立运行）时 能写 TRDi 寄存器。 - 当 TRDMR 寄存器的 TRDSYNC 位为“1”（定时器 RD0 和定时器 RD1 同步运行）时 如果写 TRDi 寄存器，数据就同时被写到 TRD0 寄存器和 TRD1 寄存器。
选择功能	- 输出比较的输出引脚的选择 TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚中的 1 个或者多个引脚 - 比较匹配时的输出电平的选择 “L”电平输出、“H”电平输出或者电平反相输出 - 初始输出电平的选择 设定从开始计数到比较匹配为止的电平。 - 将 TRDi 置“0000H”的时序 上溢或者 TRDGRAi 寄存器的比较匹配 - 缓冲器运行（参照“8.4.2 缓冲器运行”） - 同步运行（参照“8.4.3 同步运行”） - TRDGRCi 和 TRDGRDi 的输出引脚的变更 能将 TRDGRCi 和 TRDGRDi 分别用于 TRDIOAi 引脚和 TRDIOBi 引脚的输出控制。 - 能将定时器 RD 用作内部定时器而不进行输出。

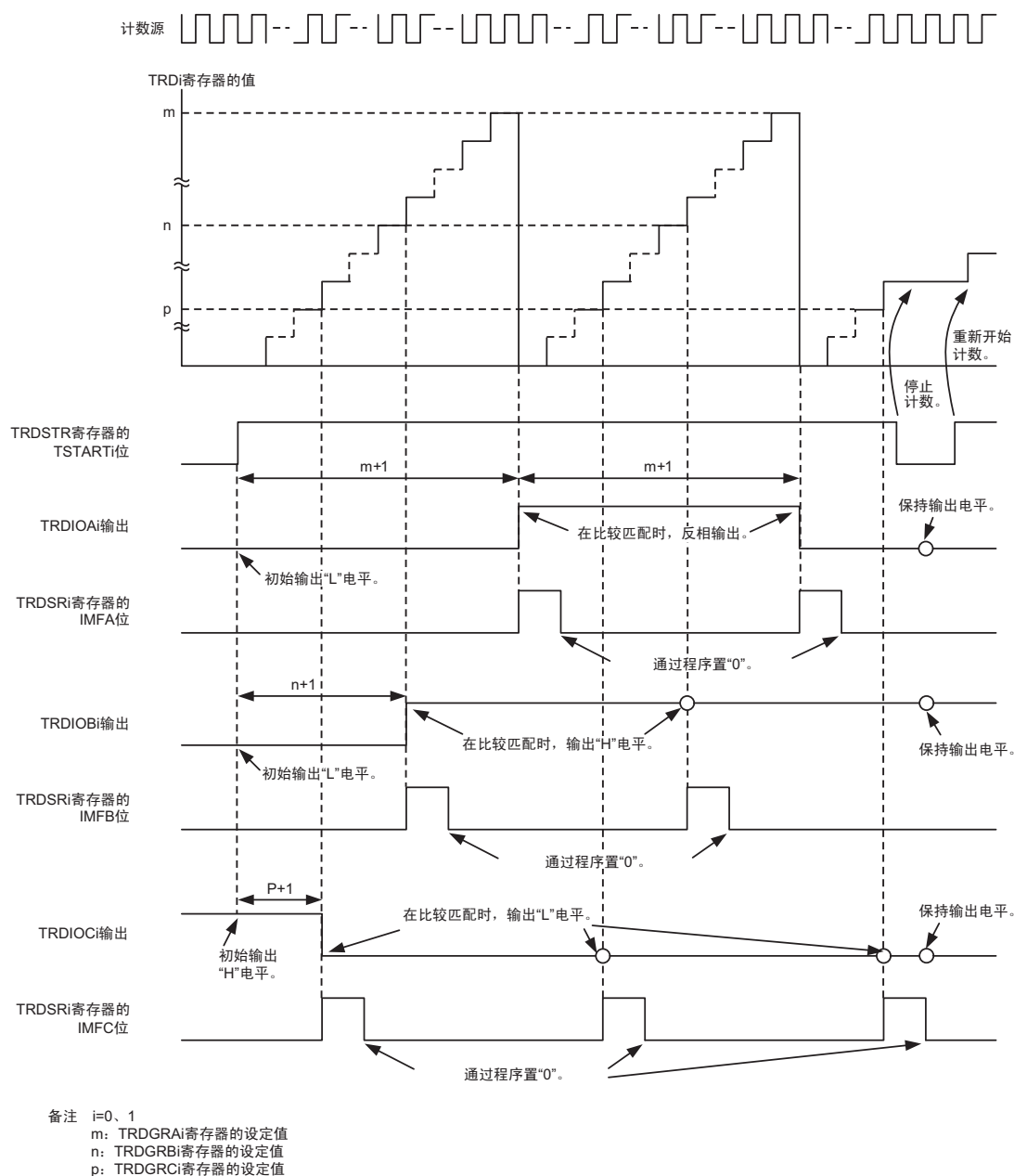
注 只有在用户选项字节（000C2H）的 FRQSEL4 位为“1”时才能选择 f_{HOCO} 。要选择 f_{HOCO} 作为定时器 RD 的计数源时，必须在将外围允许寄存器 1（PER1）的 bit4（TRD0EN）置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟，就必须在清除外围允许寄存器 1（PER1）的 bit4（TRD0EN）后进行更改。

备注 $i=0, 1$, $j=A, B, C, D$

(1) 运行例子

通过设定 TRDCR_i 寄存器 (i=0、1) 的 CCLR0 ~ CCLR2 位, 在发生输入捕捉或者比较匹配时定时器 RD_i 的计数器的值进行复位。如果比较期待值为“FFFFH”, 就和溢出相同从“FFFFH”变为“0000H”, 并且上溢标志变为“1”。

图 8-50 输出比较功能的运行例子



上图的条件如下:

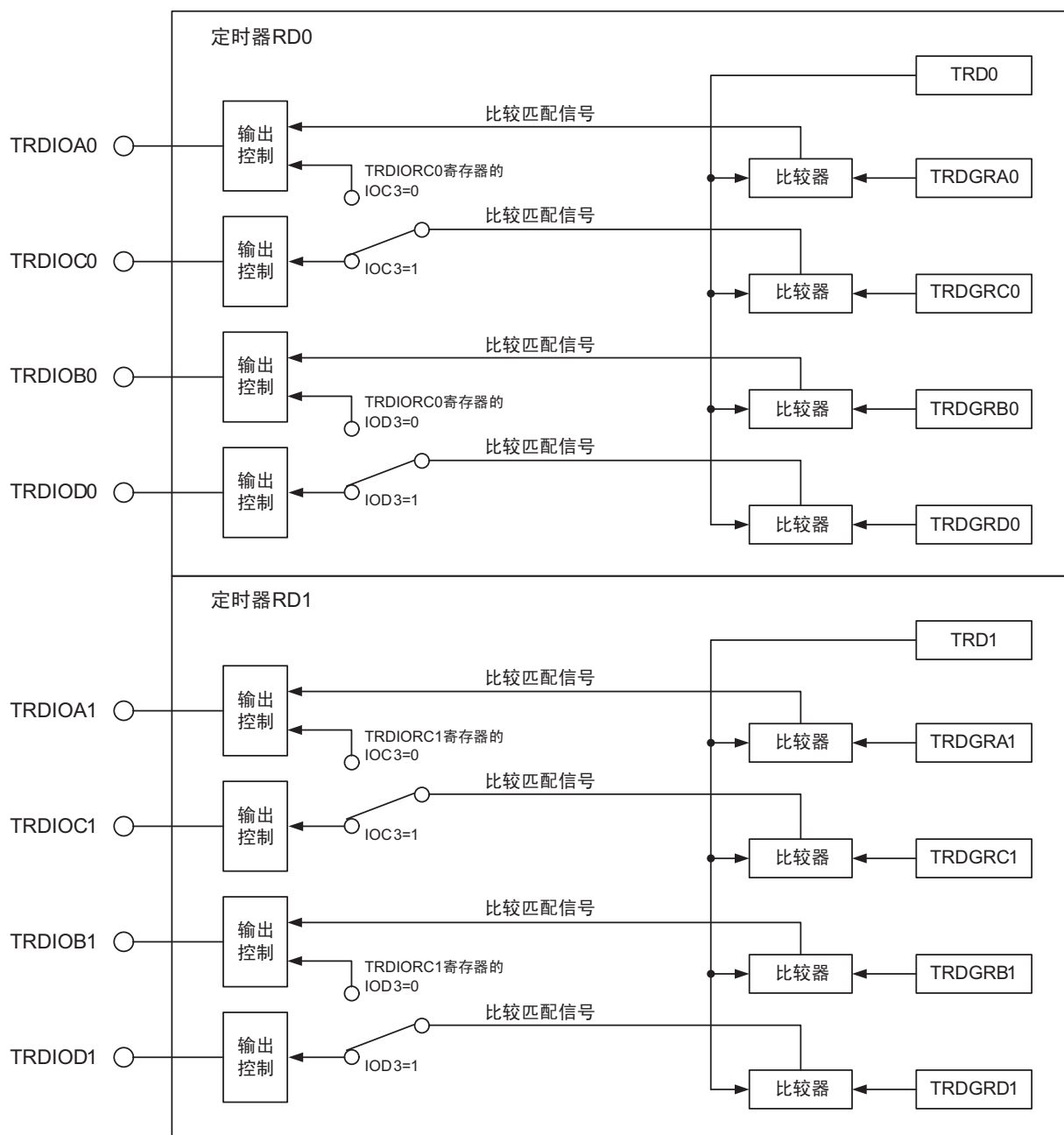
- TRDSTR寄存器的CSELi位为“1”(在比较匹配时, TRDi不停止)。
- TRDMR寄存器的TRDBFCi位和TRDBFDi位为“0”(TRDGRCi和TRDGRDi不作为缓冲器运行)。
- TRDOER1寄存器的EAI位、EBi位和ECi位为“0”(允许TRDIOAi、TRDIOBi和TRDIOCi的输出)。
- TRDCRi寄存器的CCLR2~CCLR0位为“001B”(在TRDGRAi比较匹配时, 将TRDi置“0000H”)。
- TRDOCR寄存器的TOAi位和TOBi位为“0”(在比较匹配前, 初始输出“L”电平), TOCi位为“1”(在比较匹配前, 初始输出“H”电平)。
- TRDIOA寄存器IOA2~IOA0位为“011B”(在TRDGRAi比较匹配时, TRDIOAi反相输出)。
- TRDIOB寄存器IOB2~IOB0位为“101B”(在TRDGRBi比较匹配时, TRDIOBi输出“H”电平)。
- TRDIOCI寄存器的IOC3~IOC0位为“1001B”(在TRDGRCi比较匹配时, TRDIOCi输出“L”电平)。
- TRDIOCI寄存器的IOD3~IOD0位为“1000B”(TRDGRDi寄存器不控制TRDIOBi引脚的输出, 禁止由比较匹配引起的引脚输出)。

(2) TRDGRCi 寄存器和 TRDGRDi 寄存器 (i=0、1) 的输出引脚的变更

能将 TRDGRCi 寄存器和 TRDGRDi 寄存器分别用于 TRDIOAi 引脚和 TRDIOBi 引脚的输出控制。因此，能对各引脚进行如下的输出控制：

- 通过 TRDGRAi 寄存器的值和 TRDGRCi 寄存器的值，控制 TRDIOAi 输出。
- 通过 TRDGRBi 寄存器的值和 TRDGRDi 寄存器的值，控制 TRDIOBi 输出。

图 8-51 TRDGRCi 和 TRDGRDi 的输出引脚的变更

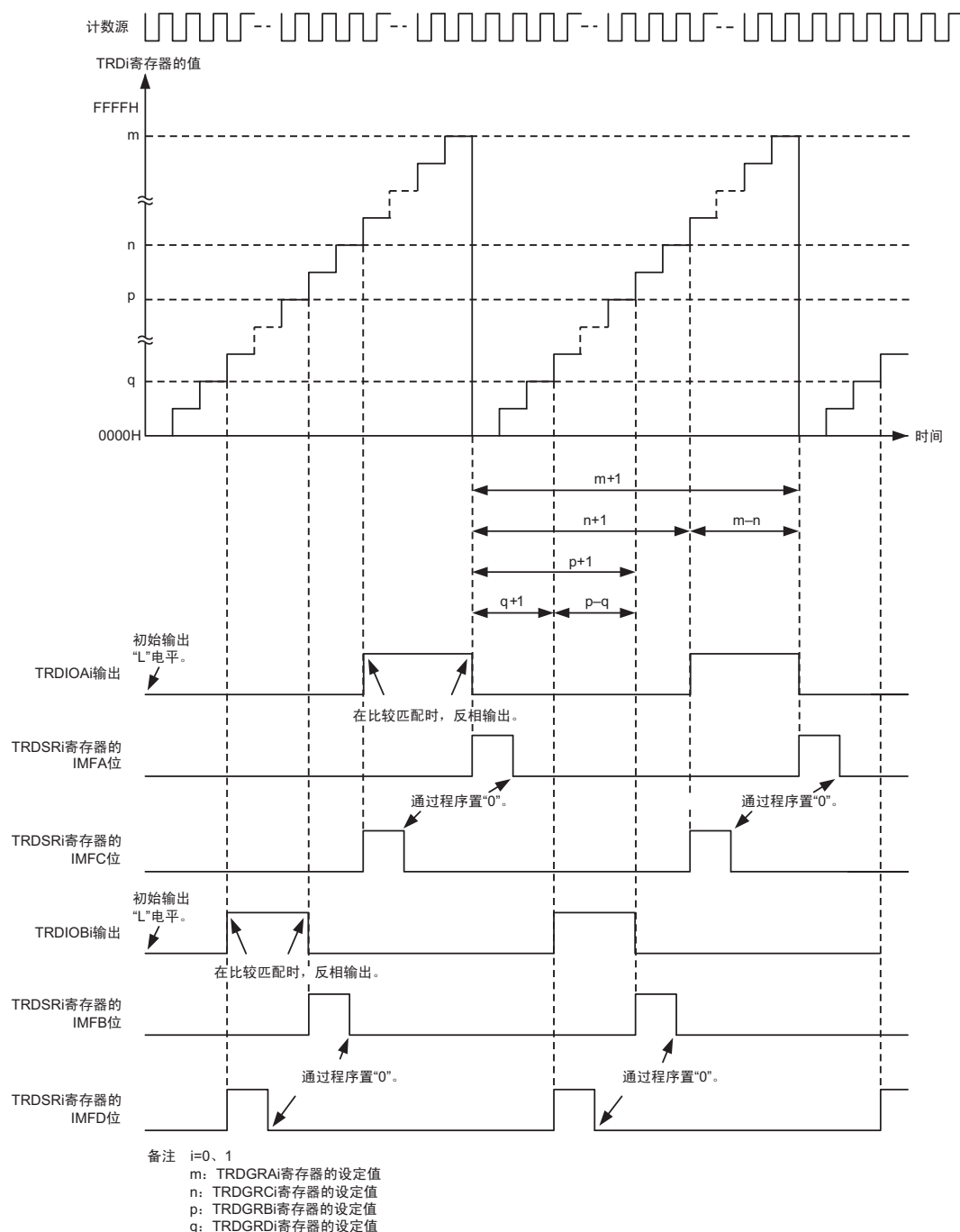


要更改 TRDGRCi 寄存器和 TRDGRDi 寄存器的输出引脚时，必须进行以下的设定：

- 通过 TRDIORCi 寄存器的 IOj3 位 (j=C、D) 选择“0” (TRDGRji 寄存器的输出引脚的变更)。
- 将 TRDMR 寄存器的 TRDBFji 位置“0” (通用寄存器)。
- 给 TRDGRAi 寄存器和 TRDGRCi 寄存器设定不同的值，并且给 TRDGRBi 寄存器和 TRDGRDi 寄存器设定不同的值。

将TRDGRCi和TRDGRDi分别用于TRDIOAi引脚和TRDIOBi引脚的输出控制时的运行例子如图8-52所示。

图 8-52 将 TRDGRCi 和 TRDGRDi 分别用于 TRDIOAi 引脚和 TRDIOBi 引脚的输出控制时的运行例子



上图的条件如下：

- TRDSTR寄存器的CSELi位为“1”(在比较匹配时，TRDi不停止)。
- TRDMR寄存器的TRDBFCi位和TRDBFDi位为“0”(TRDGRCi和TRDGRDi不作为缓冲器运行)。
- TRDOER1寄存器的EAI位和EBi位为“0”(允许TRDIOAi和TRDIOBi的输出)。
- TRDCRi寄存器的CCLR2~CCLR0位为“001B”(在TRDGRAi比较匹配时，将TRDi置“0000H”)。
- TRDOCR寄存器的TOAi位和TOBi位为“0”(在比较匹配前，初始输出“L”电平)。
- TRDIORAi寄存器的IOA2~IOA0位为“011B”(在TRDGRAi比较匹配时，TRDIOAi反相输出)。
- TRDIORBi寄存器的IOB2~IOB0位为“011B”(在TRDGRBi比较匹配时，TRDIOBi反相输出)。
- TRDIORCi寄存器的IOC3~IOC0位为“0011B”(在TRDGRCi比较匹配时，TRDIOAi反相输出)。
- TRDIORCi寄存器的IOD3~IOD0位为“0011B”(在TRDGRDi比较匹配时，TRDIOBi反相输出)。

8.5.3 PWM 功能

这是输出 PWM 波形的功能。通过定时器 RD_i (i=0、1) 最多能输出 3 个同周期的 PWM 波形。通过定时器 RD0 和定时器 RD1 的同步运行，最多能输出 6 个同周期的 PWM 波形。

因为 TRDIO_j 引脚 (j=B、C、D) 和 TRDGR_j 寄存器组合使用，所以能按引脚选择为 PWM 功能、或者其他模式和功能（但是，无论将哪个引脚用作 PWM 功能，都要使用 TRDGRA_i 寄存器，因此 TRDGRA_i 寄存器不能用于其他模式）。

PWM 功能的框图和规格分别如图 8-53 和表 8-14 所示，PWM 功能的运行例子如图 8-54 和图 8-55 所示。

图 8-53 PWM 功能的框图

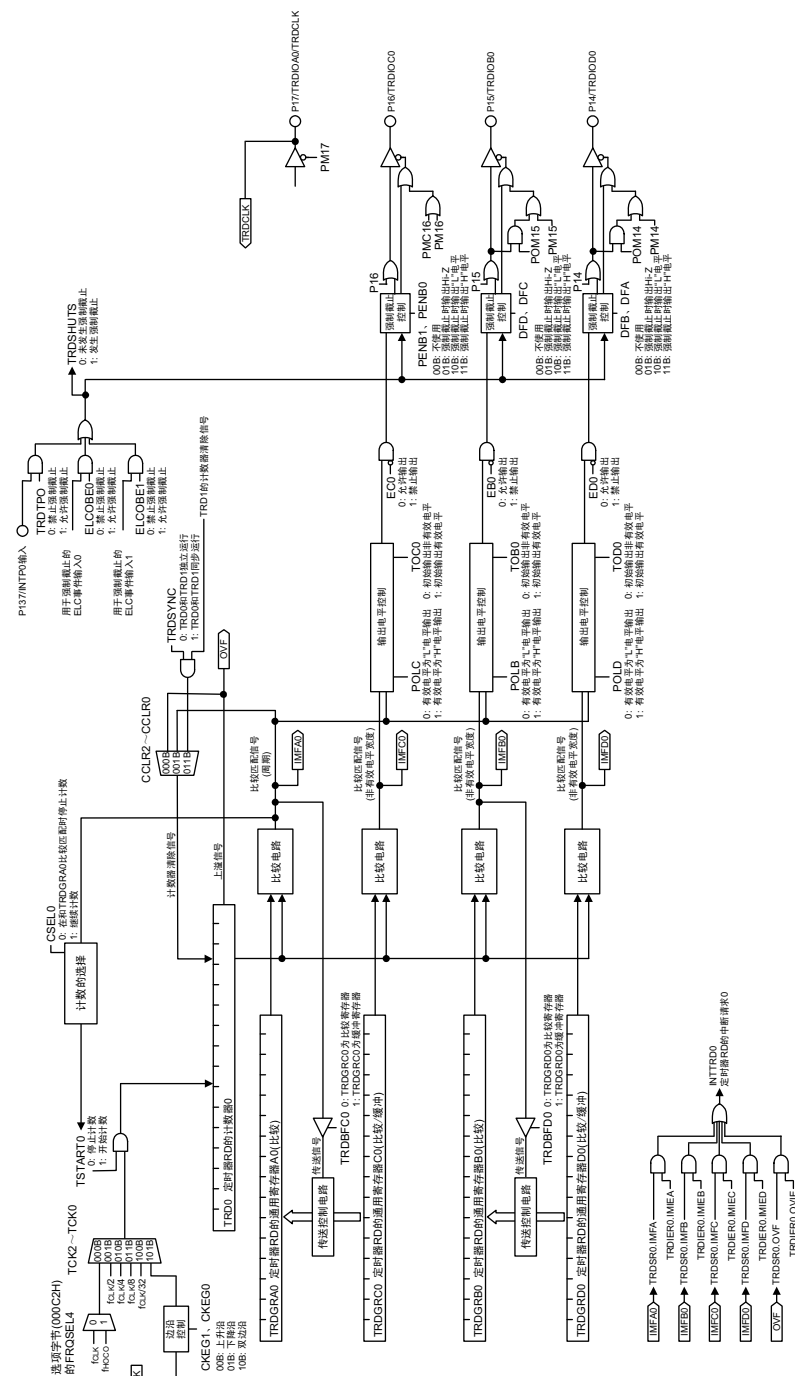


表 8-14 PWM 功能的规格

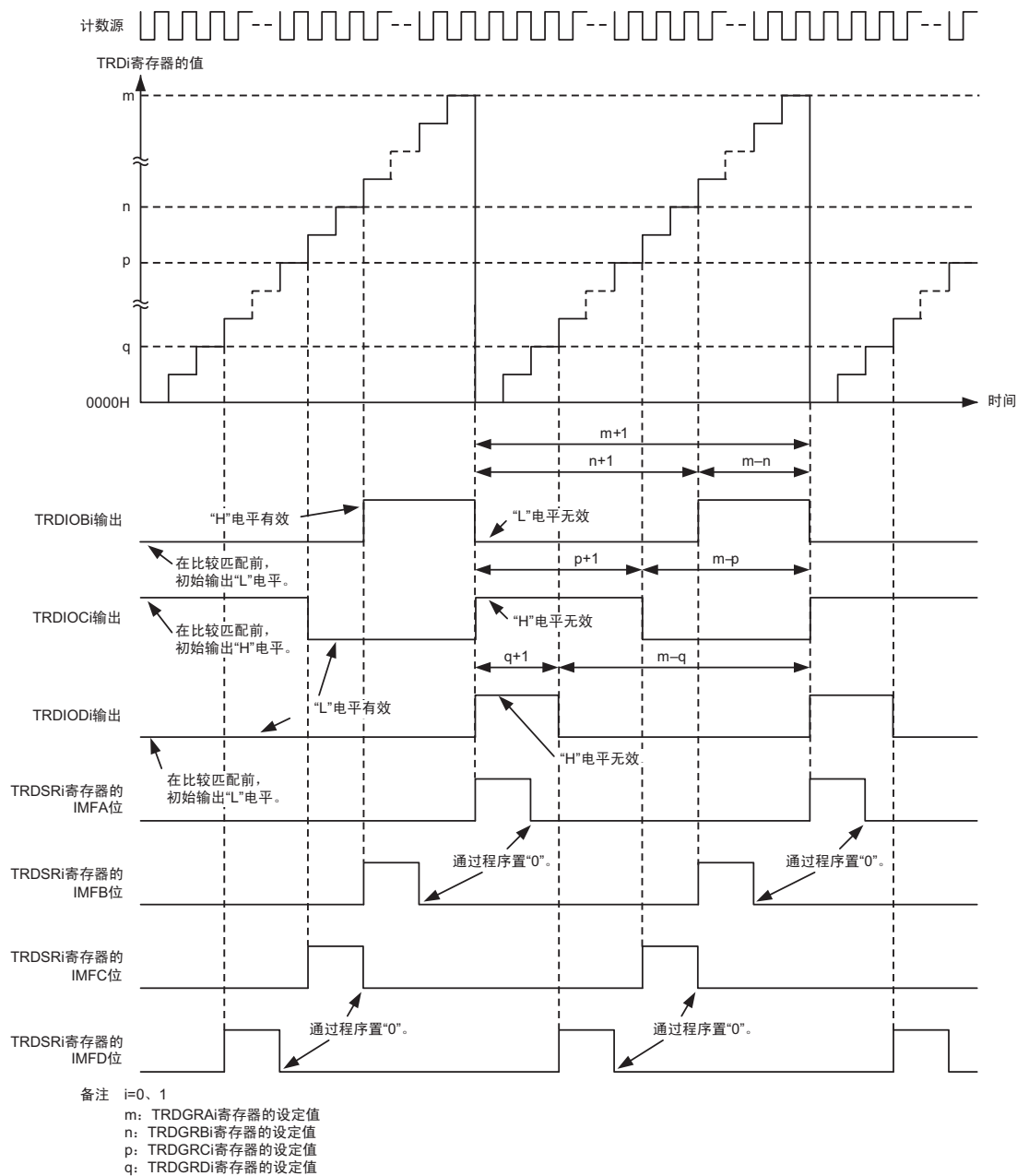
项目	规格
计数源	f_{HOCO} 注、 f_{CLK} 、 $f_{CLK}/2$ 、 $f_{CLK}/4$ 、 $f_{CLK}/8$ 、 $f_{CLK}/32$ TRDCLK 引脚的外部输入信号（能通过程序选择有效边沿）
计数	递增计数
PWM 波形	PWM 周期: $1/f_k \times (m+1)$ 有效电平宽度: $1/f_k \times (m-n)$ 无效电平宽度: $1/f_k \times (n+1)$ f_k : 计数源的频率 m : TRDGRAi 寄存器的设定值 n : TRDGRji 寄存器的设定值
计数开始条件	给 TRDSTR 寄存器的 TSTARTi 位写“1”（开始计数）。
计数停止条件	- 当 TRDSTR 寄存器的 CSELi 位为“1”时，给 TSTARTi 位写“0”（停止计数）。PWM 输出引脚保持停止计数前的输出电平。 - 在 TRDSTR 寄存器的 CSELi 位为“0”并且发生 TRDGRAi 的比较匹配时停止计数。PWM 输出引脚保持比较匹配引起输出变化后的电平。
中断请求的发生时序	- 比较匹配（TRDi 寄存器和 TRDGRhi 寄存器的内容相同） - TRDi 的上溢
TRDIOA0 引脚功能	I/O 端口或者 TRDCLK（外部时钟）输入
TRDIOA1 引脚功能	I/O 端口
TRDIOB0、TRDIOC0、TRDIOD0、TRDIOB1、TRDIOC1、TRDIOD1 引脚功能	I/O 端口或者 PWM 输出（按引脚进行选择）
INTP0 引脚功能	脉冲输出强制截止信号的输入（输入专用端口或者 INTP0 中断输入）
读定时器	如果读 TRDi 寄存器，就能读到计数值。
写定时器	能写 TRDi 寄存器。
选择功能	- 通过定时器 RD_i 进行的 1～3 个 PWM 输出引脚的选择 TRDIOBi、TRDIOC_i、TRDIOD_i 引脚中的 1 个或者多个引脚 - 各引脚有效电平的选择 - 各引脚初始输出电平的选择 - 同步运行（参照“8.4.3 同步运行”） - 缓冲器运行（参照“8.4.2 缓冲器运行”） - 脉冲输出强制截止信号的输入（参照“8.4.4 脉冲输出的强制截止”）

注 只有在用户选项字节（000C2H）的 FRQSEL4 位为“1”时才能选择 f_{HOCO} 。要选择 f_{HOCO} 作为定时器 RD 的计数源时，必须在将外围允许寄存器 1（PER1）的 bit4（TRD0EN）置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟，就必须在清除外围允许寄存器 1（PER1）的 bit4（TRD0EN）后进行更改。

备注 $i=0, 1$, $j=B, C, D$, $h=A, B, C, D$

(1) 运行例子

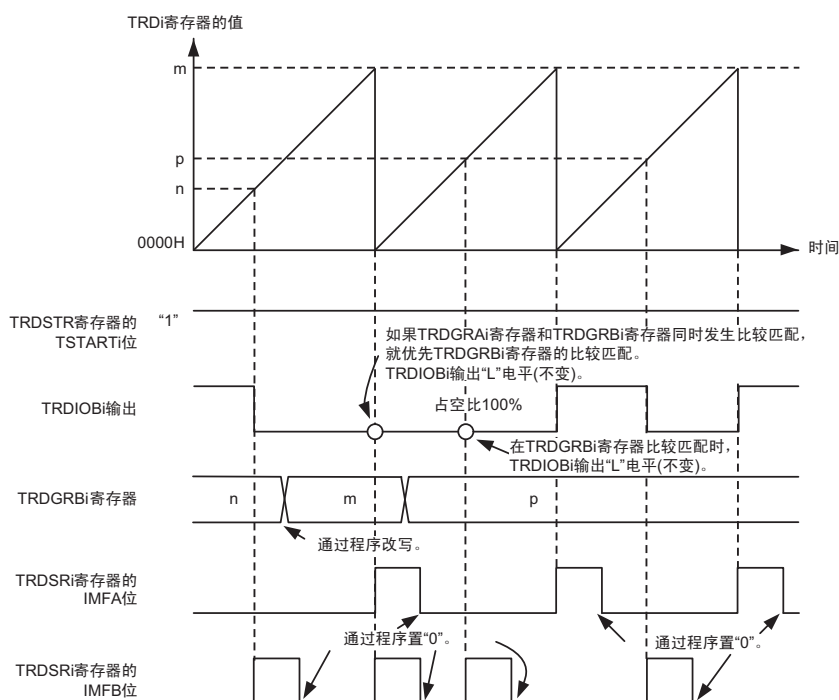
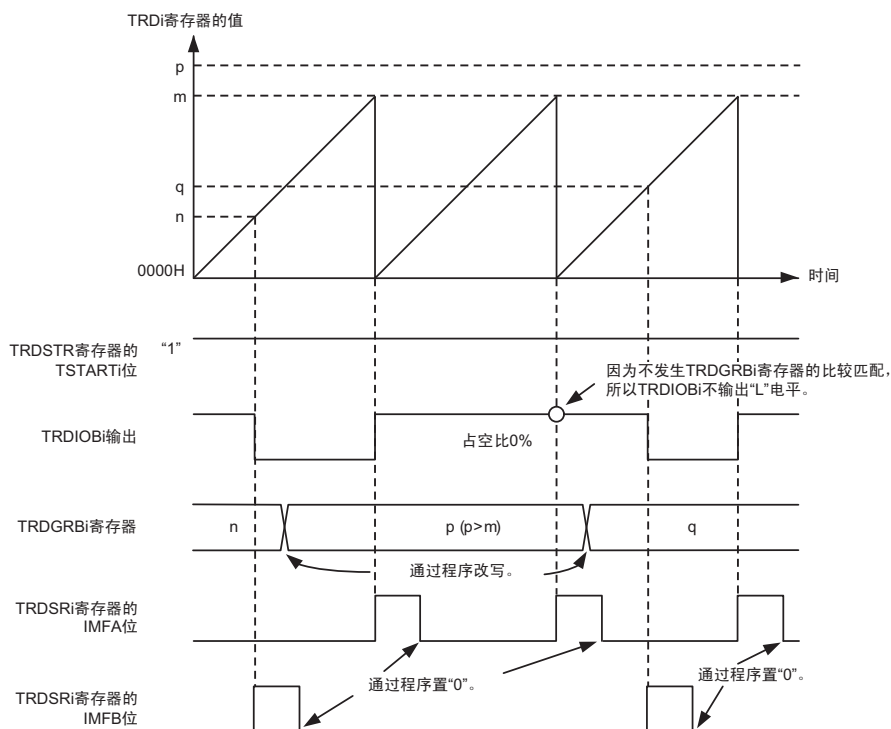
图 8-54 PWM 功能的运行例子



上图的条件如下:

- TRDMR寄存器的TRDBFCi位和TRDBFDi位为“0”(TRDGRCi寄存器和TRDGRDi寄存器不作为缓冲器运行)。
- TRDOER1寄存器的EBi位、ECi位和EDi位为“0”(允许TRDIOBi、TRDIOCi和TRDIODi的输出)。
- TRDOCR寄存器的TOBi位和TOCi位为“0”(无效电平)，TODi位为“1”(有效电平)。
- TRDPCRi寄存器的POLBi位为“1”(“H”电平有效)，POLCi位和POLDi位为“0”(“L”电平有效)。

图 8-55 PWM 功能的运行例子 (占空比为 0% 和 100%)



备注 i=0、1
m: TRDGRAi寄存器的设定值

上图的条件如下：

- TRDOER1寄存器的EBi位为“0”(允许TRDIOBi输出)。
- TRDPOCRi寄存器的POLB位为“0”(“L”电平有效)。

8.5.4 复位同步 PWM 模式

输出 3 个正相和 3 个反相（共 6 个）的同周期的 PWM 波形（三相、锯齿波调制、无死区时间）。

复位同步 PWM 模式的框图和运行例子分别如图 8-56 和图 8-57 所示，复位同步 PWM 模式的规格如表 8-15 所示。

有关占空比为 0% 和 100% 的 PWM 运行例子，请参照“图 8-55 PWM 功能的运行例子（占空比为 0% 和 100%）”。

图 8-56 复位同步 PWM 模式的框图

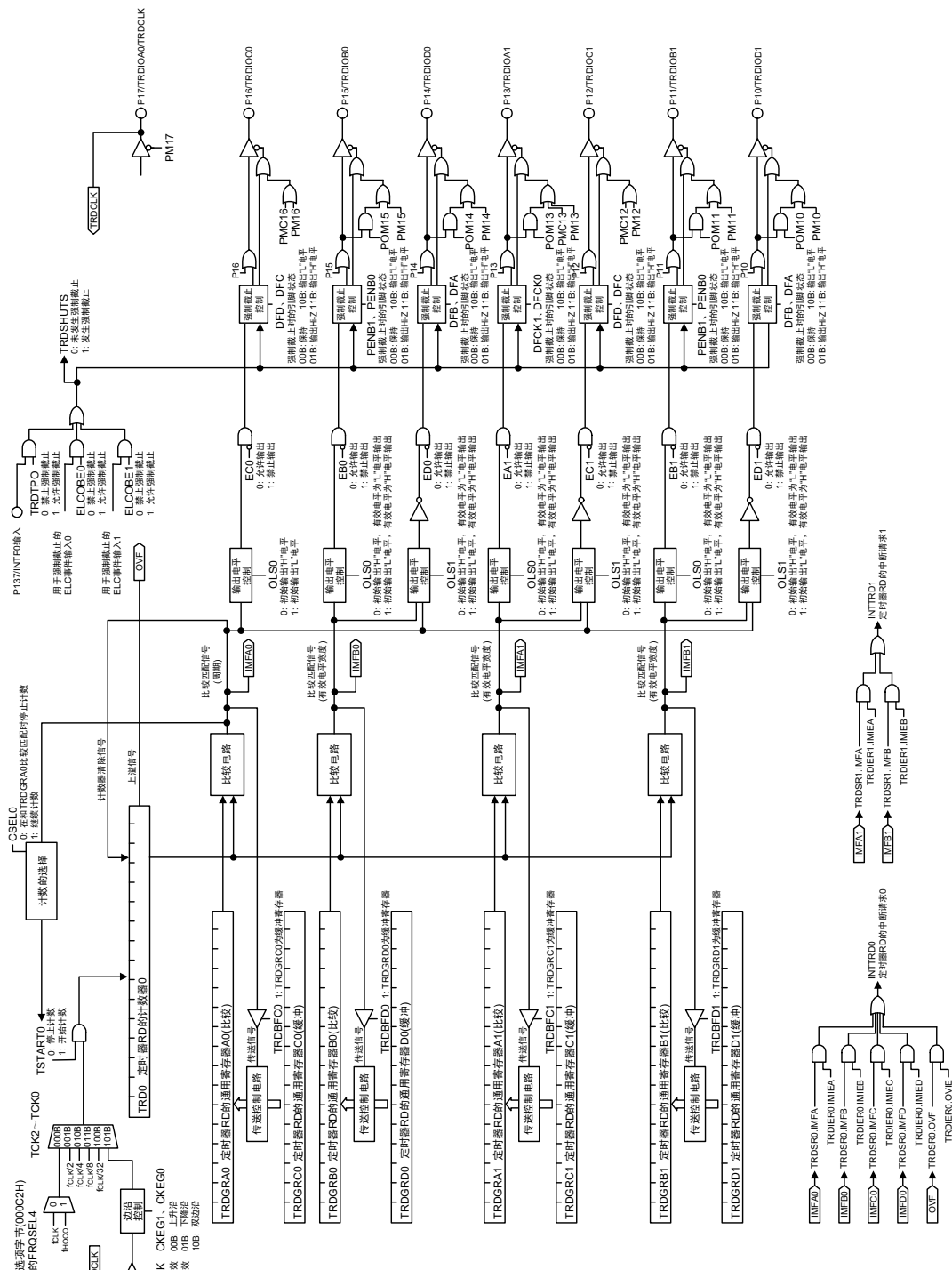
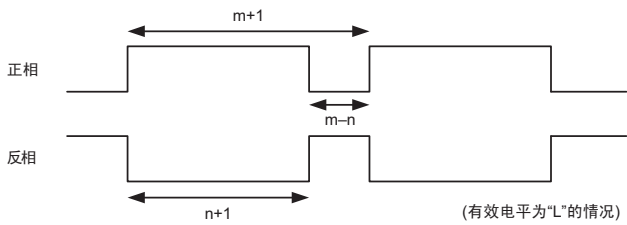


表 8-15 复位同步 PWM 模式的规格

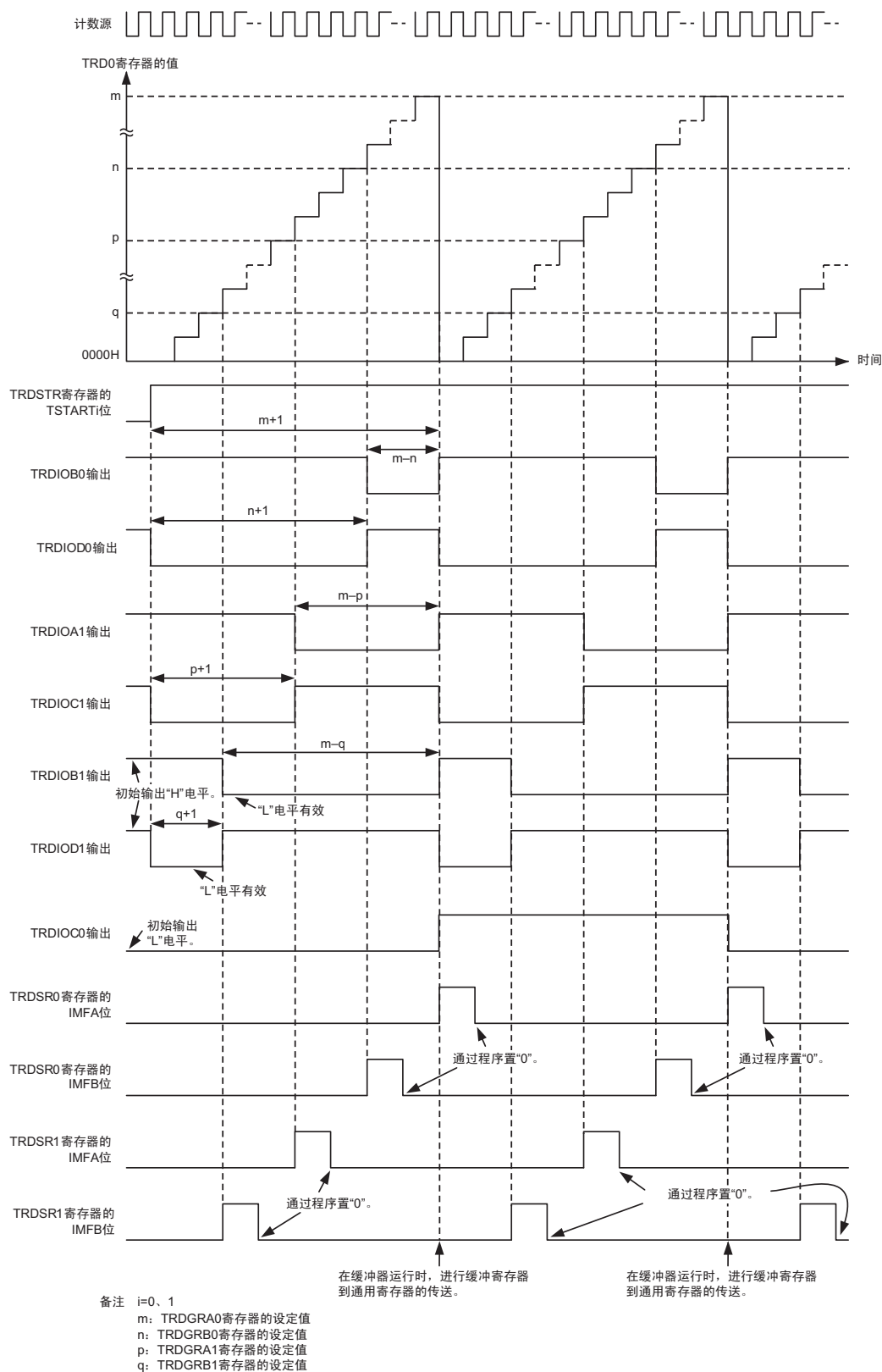
项目	规格
计数源	f_{HOCO} 注、 f_{CLK} 、 $f_{CLK}/2$ 、 $f_{CLK}/4$ 、 $f_{CLK}/8$ 、 $f_{CLK}/32$ TRDCLK 引脚的外部输入信号（能通过程序选择有效边沿）
计数	TRD0 为递增计数（不使用 TRD1）。
PWM 波形	PWM 周期: $1/f_k \times (m+1)$ 正相有效电平宽度: $1/f_k \times (m-n)$ 反相有效电平宽度: $1/f_k \times (n+1)$ f_k: 计数源的频率 m: TRDGRA0 寄存器的设定值 n: TRDGRB0 寄存器的设定值（PWM 输出 1） TRDGRA1 寄存器的设定值（PWM 输出 2） TRDGRB1 寄存器的设定值（PWM 输出 3）  (有效电平为“L”的情况)
计数开始条件	给 TRDSTR 寄存器的 TSTART0 位写“1”（开始计数）。
计数停止条件	- 当 TRDSTR 寄存器的 CSEL0 位为“1”时，给 TSTART0 位写“0”（停止计数）。PWM 输出引脚输出 TRDFCR 寄存器的 OLS0 位和 OLS1 位选择的初始输出电平。 - 在 TRDSTR 寄存器的 CSEL0 位为“0”并且发生 TRDGRA0 的比较匹配时停止计数。PWM 输出引脚输出 TRDFCR 寄存器的 OLS0 位和 OLS1 位选择的初始输出电平。
中断请求的发生时序	- 比较匹配（TRD0 寄存器和 TRDGRj0、TRDGRA1、TRDGRB1 寄存器的内容相同） - TRD0 的上溢
TRDIOA0 引脚功能	I/O 端口或者 TRDCLK（外部时钟）输入
TRDIOB0 引脚功能	PWM 输出 1 的正相输出
TRDIOD0 引脚功能	PWM 输出 1 的反相输出
TRDIOA1 引脚功能	PWM 输出 2 的正相输出
TRDIOC1 引脚功能	PWM 输出 2 的反相输出
TRDIOB1 引脚功能	PWM 输出 3 的正相输出
TRDIOD1 引脚功能	PWM 输出 3 的反相输出
TRDIOC0 引脚功能	在每个 PWM 周期进行反相输出。
INTP0 引脚功能	脉冲输出强制截止信号的输入（输入专用端口或者 INTP0 中断输入）
读定时器	如果读 TRD0 寄存器，就能读到计数值。
写定时器	能写 TRD0 寄存器。
选择功能	- 正反相有效电平和初始输出电平的选择 - 缓冲器运行（参照“8.4.2 缓冲器运行”） - 脉冲输出强制截止信号的输入（参照“8.4.4 脉冲输出的强制截止”）

注 只有在用户选项字节（000C2H）的 FRQSEL4 位为“1”时才能选择 f_{HOCO} 。要选择 f_{HOCO} 作为定时器 RD 的计数源时，必须在将外围允许寄存器 1（PER1）的 bit4（TRD0EN）置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟，就必须在清除外围允许寄存器 1（PER1）的 bit4（TRD0EN）后进行更改。

备注 j=A、B、C、D

(1) 运行例子

图 8-57 复位同步 PWM 模式的运行例子



上图的条件如下：
 TRDFCR寄存器的OLS1位和OLS0位为“0”(初始输出“H”电平，“L”电平有效)。

8.5.5 互补 PWM 模式

输出 3 个正相和 3 个反相（共 6 个）的同周期的 PWM 波形（三相、三角波调制、有死区时间）。

互补 PWM 模式的框图如图 8-58 所示，互补 PWM 模式的规格如表 8-16 所示，互补 PWM 模式的输出模型和运行例子分别如图 8-59 和图 8-60 所示。

图 8-58 互补 PWM 模式的框图

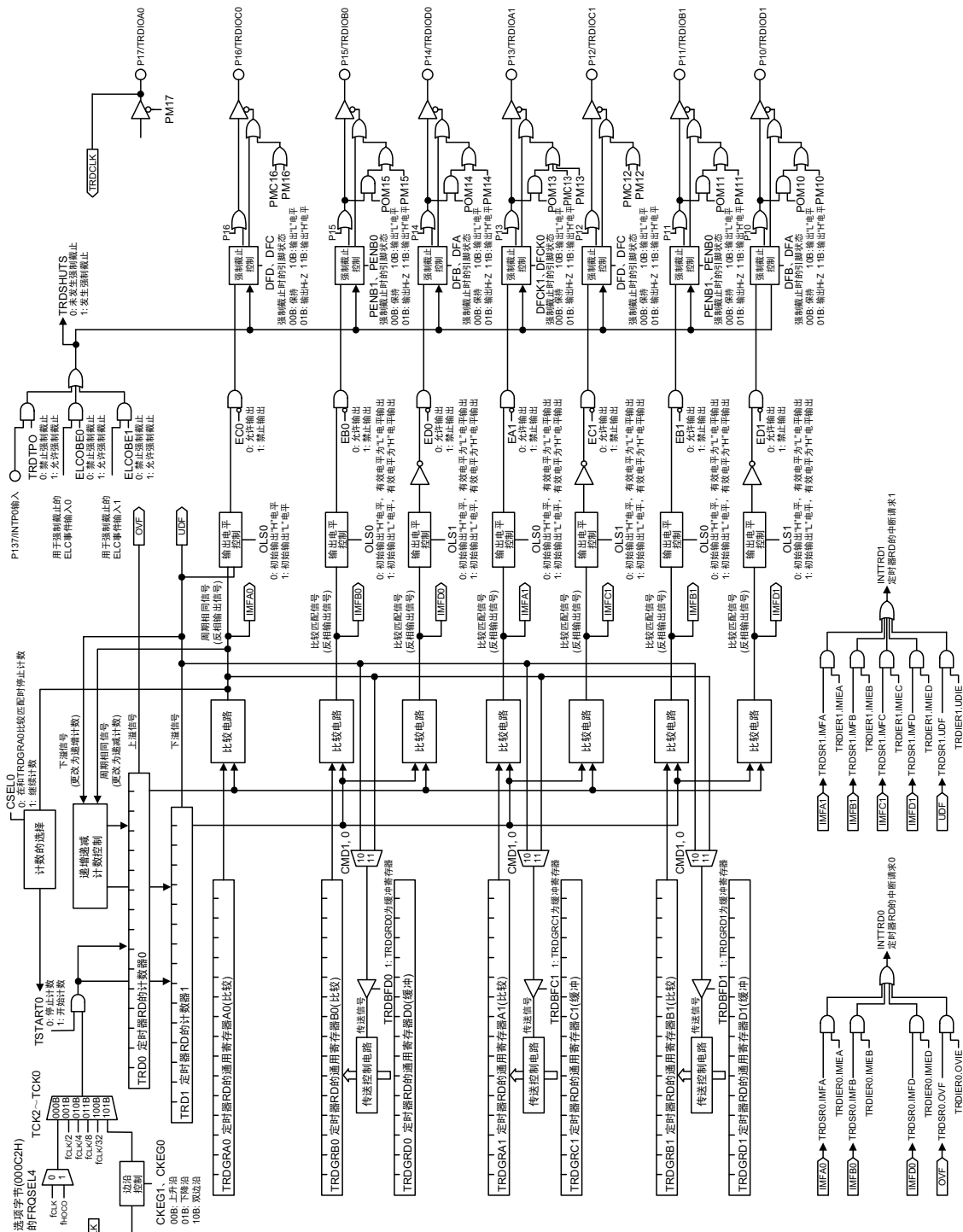
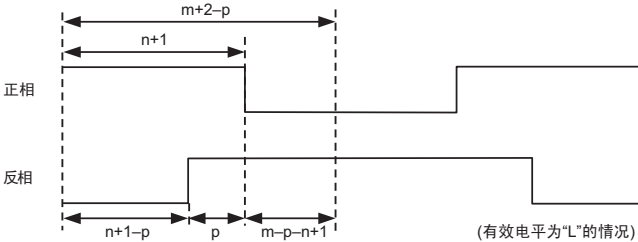


表 8-16 互补 PWM 模式的规格

项目	规格
计数源	f_{HOCO} 注 1、 f_{CLK} 、 $f_{CLK}/2$ 、 $f_{CLK}/4$ 、 $f_{CLK}/8$ 、 $f_{CLK}/32$ TRDCLK 引脚的外部输入信号（能过程序选择有效边沿） 必须给 TRDCR0 寄存器的 TCK0 ~ TCK2 位和 TRDCR1 寄存器的 TCK0 ~ TCK2 设定相同的值（相同的计数源）。
计数	递增计数或者递减计数 如果在递增计数的过程中 TRD0 寄存器和 TRDGRA0 寄存器比较匹配，TRD0 和 TRD1 就都变为递减计数；如果在递减计数的过程中 TRD1 寄存器从“0000H”变为“FFFFH”，TRD0 和 TRD1 就都变为递增计数。
PWM 波形	PWM 周期： $1/f_k \times (m+2-p) \times 2$ 注 2 死区时间： p 正相有效电平宽度： $1/f_k \times (m-n-p+1) \times 2$ 反相有效电平宽度： $1/f_k \times (n+1-p) \times 2$ f_k : 计数源的频率 m : TRDGRA0 寄存器的设定值 n : TRDGRB0 寄存器的设定值（PWM 输出 1） TRDGRA1 寄存器的设定值（PWM 输出 2） TRDGRB1 寄存器的设定值（PWM 输出 3） p : TRD0 寄存器的设定值  (有效电平为“L”的情况)
计数开始条件	给 TRDSTR 寄存器的 TSTART0 位和 TSTART1 位写“1”（开始计数）。
计数停止条件	当 TRDSTR 寄存器的 CSEL0 位为“1”时，给 TSTART0 位和 TSTART1 位写“0”（停止计数）（PWM 输出引脚输出 TRDFCR 寄存器的 OLS0 位和 OLS1 位选择的初始输出电平）。
中断请求的发生时序	- 比较匹配（TRDi 寄存器和 TRDGRji 寄存器的内容相同） - TRD1 的下溢
TRDIOA0 引脚功能	I/O 端口或者 TRDCLK（外部时钟）输入
TRDIOB0 引脚功能	PWM 输出 1 的正相输出引脚
TRDIOD0 引脚功能	PWM 输出 1 的反相输出引脚
TRDIOA1 引脚功能	PWM 输出 2 的正相输出引脚
TRDIOC1 引脚功能	PWM 输出 2 的反相输出引脚
TRDIOB1 引脚功能	PWM 输出 3 的正相输出引脚
TRDIOD1 引脚功能	PWM 输出 3 的反相输出引脚
TRDIOC0 引脚功能	在每 1/2 个 PWM 周期进行反相输出。
INTP0 引脚功能	脉冲输出强制截止信号的输入（输入专用端口或者 INTP0 中断输入）
读定时器	如果读 TRDi 寄存器，就能读到计数值。
写定时器	能写 TRDi 寄存器。
选择功能	- 脉冲输出强制截止信号的输入（参照“8.4.4 脉冲输出的强制截止”） - 正反相有效电平和初始输出电平的选择 - 缓冲寄存器的传送时序的选择

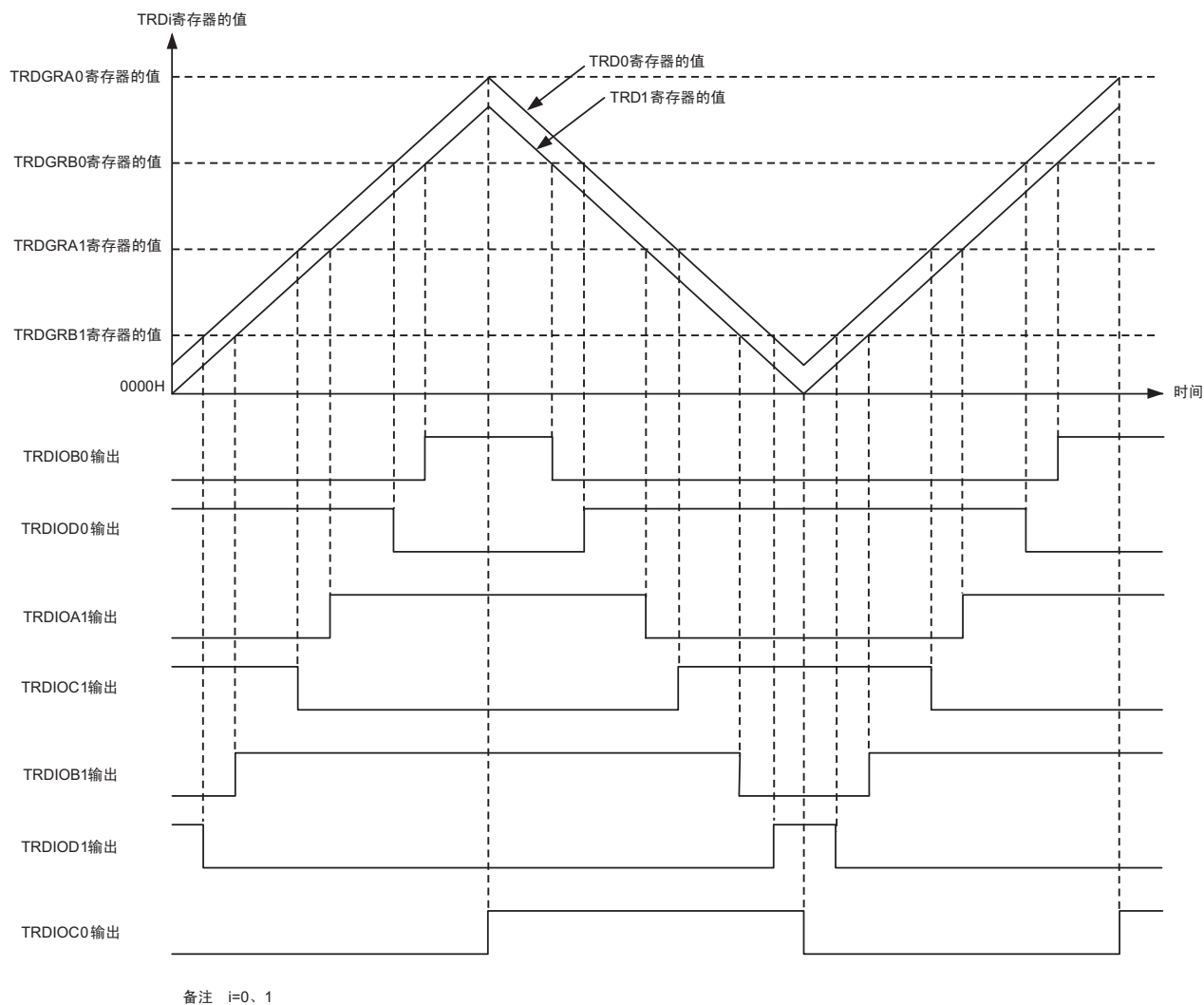
注 1. 只有在用户选项字节（000C2H）的 FRQSEL4 位为“1”时才能选择 f_{HOCO} 。要选择 f_{HOCO} 作为定时器 RD 的计数源时，必须在将外围允许寄存器 1（PER1）的 bit4（TRD0EN）置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟，就必须在清除外围允许寄存器 1（PER1）的 bit4（TRD0EN）后进行更改。

2. 在开始计数后，PWM 周期是固定的。

备注 i=0、1，j=A、B、C、D

(1) 运行例子

图 8-59 互补 PWM 模式的输出模型



(2) 缓冲寄存器的数据传送时序

- TRDGRD0、TRDGRC1、TRDGRD1 寄存器到 TRDGRB0、TRDGRA1、TRDGRB1 寄存器的数据传送给 TRDFCR 寄存器的 CMD1 位和 CMD0 位为“10B”并且 TRD1 发生下溢时进行数据传送。
在 CMD1 位和 CMD0 位为“11B”并且 TRD0 寄存器和 TRDGRA0 寄存器比较匹配时进行数据传送。

8.5.6 PWM3 模式

输出 2 个同周期的 PWM 波形。

PWM3 模式的框图和运行例子分别如图 8-61 和图 8-62 所示，PWM3 模式的规格如表 8-17 所示。

图 8-61 PWM3 模式的框图

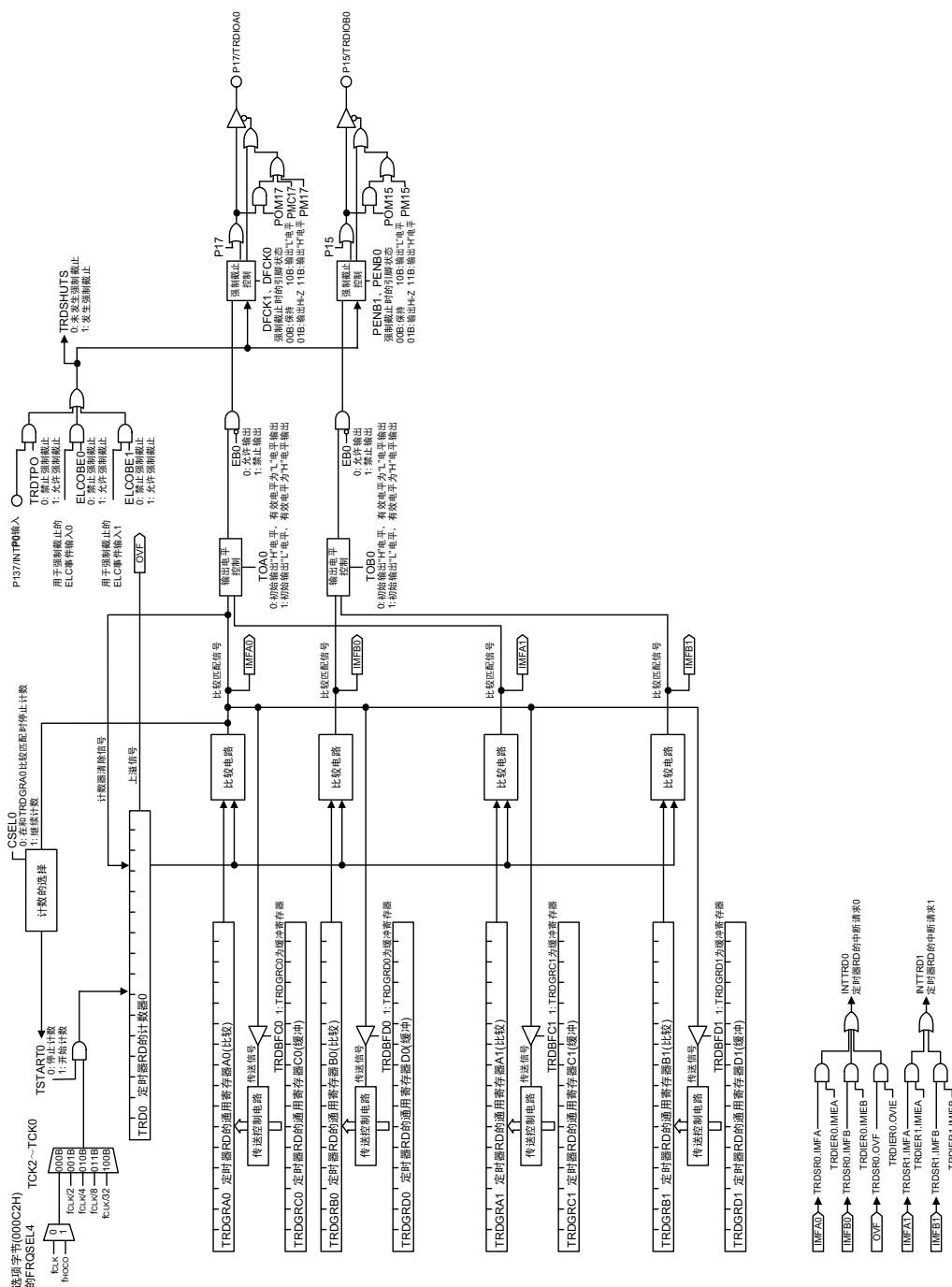
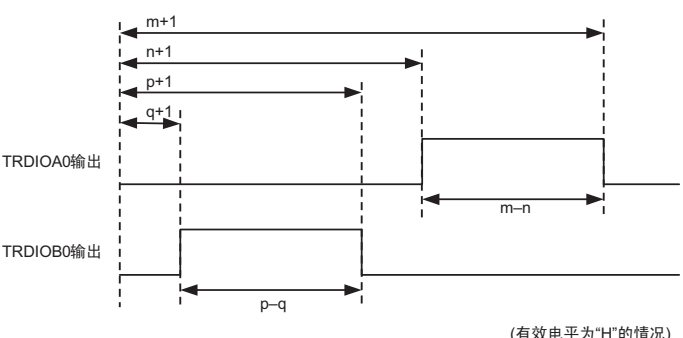


表 8-17 PWM3 模式的规格

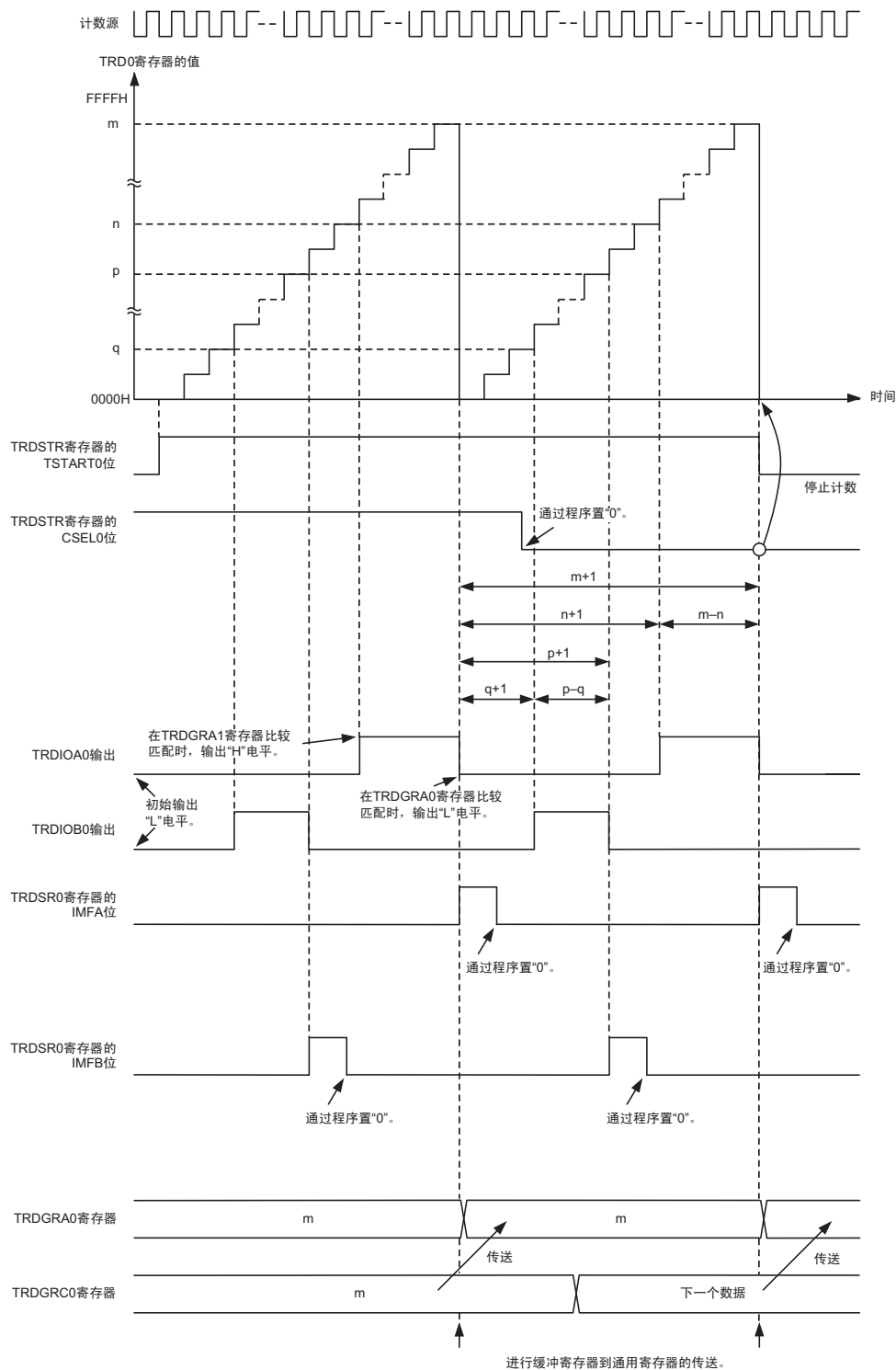
项目	规格
计数源	f_{HOCO} 注、 f_{CLK} 、 $f_{CLK}/2$ 、 $f_{CLK}/4$ 、 $f_{CLK}/8$ 、 $f_{CLK}/32$
计数	TRD0 为递增计数（不使用 TRD1）。
PWM 波形	PWM 周期: $1/f_k \times (m+1)$ TRDIOA0 输出的有效电平宽度: $1/f_k \times (m-n)$ TRDIOB0 输出的有效电平宽度: $1/f_k \times (p-q)$ f_k: 计数源的频率 m: TRDGRA0 寄存器的设定值 n: TRDGRA1 寄存器的设定值 p: TRDGRB0 寄存器的设定值 q: TRDGRB1 寄存器的设定值  (有效电平为“H”的情况)
计数开始条件	给 TRDSTR 寄存器的 TSTART0 位写“1”（开始计数）。
计数停止条件	- 当 TRDSTR 寄存器的 CSEL0 位为“1”时，给 TSTART0 位写“0”（停止计数）。PWM 输出引脚保持停止计数前的输出电平。 - 在 TRDSTR 寄存器的 CSEL0 位为“0”并且发生 TRDGRA0 的比较匹配时停止计数。PWM 输出引脚保持比较匹配引起输出变化后的电平。
中断请求的发生时序	- 比较匹配（TRDi 寄存器和 TRDGRji 寄存器的内容相同） - TRD0 的上溢
TRDIOA0 引脚和 TRDIOB0 引脚的功能	PWM 输出
TRDIOC0、TRDIOD0、TRDIOA1 ~ TRDIOD1 引脚功能	I/O 端口
INTP0 引脚功能	脉冲输出强制截止信号的输入（输入专用端口或者 INTP0 中断输入）
读定时器	如果读 TRD0 寄存器，就能读到计数值。
写定时器	能写 TRD0 寄存器。
选择功能	- 脉冲输出强制截止信号的输入（参照“8.4.4 脉冲输出的强制截止”） - 各引脚有效电平的选择 - 缓冲器运行（参照“8.4.2 缓冲器运行”）

注 只有在用户选项字节（000C2H）的 FRQSEL4 位为“1”时才能选择 f_{HOCO} 。要选择 f_{HOCO} 作为定时器 RD 的计数源时，必须在将外围允许寄存器 1（PER1）的 bit4（TRD0EN）置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟，就必须在清除外围允许寄存器 1（PER1）的 bit4（TRD0EN）后进行更改。

备注 $i=0, 1$, $j=A, B, C, D$

(1) 运行例子

图 8-62 PWM3 模式的运行例子



备注 j=A、B
 m: TRDGRA0寄存器的设定值
 n: TRDGRA1寄存器的设定值
 p: TRDGRB0寄存器的设定值
 q: TRDGRB1寄存器的设定值

上图的条件如下:

- TRDOCR寄存器的TOA0位和TOB0位都为"0"(初始输出"L"电平, 在TRDGRj1寄存器比较匹配时输出"H"电平, 在TRDGRj0寄存器比较匹配时输出"L"电平)。
- TRDMR寄存器的TRDBFC0位为"1"(TRDGR0寄存器为TRDGRA0寄存器的缓冲寄存器)。

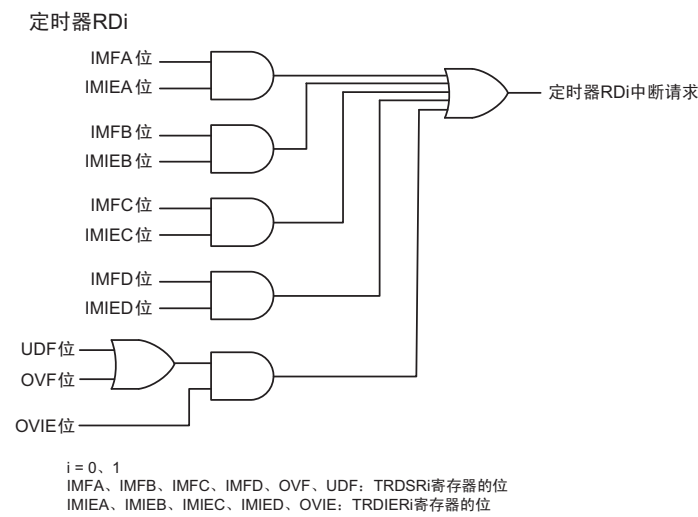
8.6 定时器 RD 中断

定时器 RD 从定时器 RD0 和定时器 RD1 的各 6 个中断源中产生定时器 RD_i (i=0、1) 中断请求。定时器 RD 中断的相关寄存器如表 8-18 所示，定时器 RD 中断的框图如图 8-63 所示。

表 8-18 定时器 RD 中断的相关寄存器

	定时器 RD 的状态寄存器	定时器 RD 的中断允许寄存器	中断请求标志 (寄存器)	中断屏蔽标志 (寄存器)	优先级指定标志 (寄存器)
定时器 RD0	TRDSR0	TRDIER0	TRDIF0 (IF2H)	TRDMK0 (MK2H)	TRDPR00 (PR02H) TRDPR10 (PR12H)
定时器 RD1	TRDSR1	TRDIER1	TRDIF1 (IF2H)	TRDMK1 (MK2H)	TRDPR01 (PR02H) TRDPR11 (PR12H)

图 8-63 定时器 RD 中断的框图



因为定时器 RD 从多个中断请求源中产生 1 个中断请求（定时器 RD 中断），所以和其他可屏蔽中断有以下不同：

- 当 TRDSR_i 寄存器的位为“1”并且对应的 TRDIER_i 寄存器的位为“1”（允许中断）时，IF2H 寄存器的 TRDIF_i 位就变为“1”（有中断请求）。
- 当 TRDIER_i 寄存器的多个位为“1”时，必须通过 TRDSR_i 寄存器判断是哪个请求源产生的中断。
- 因为即使接受中断，TRDSR_i 寄存器的各位也不会自动变为“0”，所以必须在中断程序中将这些位置“0”。

- 要将定时器 RD 的某个中断源的状态标志（以下称为“对象状态标志”）置“0”时，如果该中断被定时器 RD 中断允许寄存器 i（TRDIERi）设定为禁止中断，就必须用以下(a)~(c)的任意一种方法置“0”。

(a) 必须在将定时器 RD 中断允许寄存器 i（TRDIERi）置“00H”（禁止全部中断）后给对象状态标志写“0”。

(b) 当定时器 RD 中断允许寄存器 i（TRDIERi）中有被置“1”（允许）的位并且该位允许的中断源状态标志为“0”时，必须给对象状态标志写“0”。

（例）在 IMIEA 和 OVIE 为允许中断而 IMIEB 为禁止中断的状态下清除 IMFB 的情况

- 定时器 RD 中断允许寄存器 i（TRDIERi）的状态

TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA	允许中断
	0	0	0	1	0	0	0	1	

禁止中断

- 定时器 RD 状态寄存器 i（TRDSRi）的状态

TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA	要清除请求的位
	0	0	0	0	1	0	1	0	

因为允许中断的位所对应的状态标志（OVF、IMFA）为“0”，所以必须给 IMFB 写“0”。

(c) 当定时器 RD 中断允许寄存器 i（TRDIERi）中有被置“1”（允许）的位并且该位允许的中断源状态标志为“1”时，必须同时给此状态标志和对象状态标志写“0”。

（例）在 IMIEA 为允许中断而 IMIEB 为禁止中断的状态下清除 IMFB 的情况

- 定时器 RD 中断允许寄存器 i（TRDIERi）的状态

TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA	允许中断
	0	0	0	1	0	0	0	1	

禁止中断

- 定时器 RD 状态寄存器 i（TRDSRi）的状态

TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA	要清除请求的位
	0	0	0	0	1	0	1	1	

因为允许中断的位所对应的状态标志（IMFA）为“1”，所以必须同时给 IMFA 和 IMFB 写“0”。

8.7 使用定时器 RD 时的注意事项

8.7.1 SFR 的读写存取

要设定定时器 RD 时，必须先将 PER1 寄存器的 TRD0EN 位置“1”。当 TRD0EN 位为“0”时，忽视定时器 RD 的控制寄存器的写操作，而且读取值都为初始值（端口寄存器和端口模式寄存器除外）。

在计数过程中禁止改写以下寄存器：

TRDELC 寄存器、TRDMR 寄存器、TRDPMR 寄存器、TRDFCR 寄存器、TRDOER1 寄存器、TRDOER2 寄存器的 TRDPTo 位、TRDDFi 寄存器、TRDCRi 寄存器、TRDIORAi 寄存器、TRDIORCi 寄存器、TRDPOCRi 寄存器

(1) TRDSTR 寄存器

- 能通过 8 位存储器操作指令设定 TRDSTR 寄存器。
- 当 TRDSTR 寄存器的 CSELi 位（i=0、1）为“0”（在 TRDi 寄存器和 TRDGRAi 寄存器比较匹配时停止计数）时，即使给 TSTARTi 位写“0”（停止计数），也不停止计数并且 TSTARTi 位也不变。
只有在和 TRDGRAi 寄存器比较匹配时，TSTARTi 位才变为“0”（停止计数）。
在改写 TRDSTR 寄存器时，如果要在 CSELi 位为“0”的情况下不影响计数而将 CSELi 位改为“1”，就必须给 TSTARTi 位写“0”。
如果在计数器停止计数的过程中给 TSTARTi 位写“1”，就可能开始计数。
要通过程序停止计数时，必须在将 CSELi 位置“1”后给 TSTARTi 位写“0”。即使同时（使用 1 条指令）给 CSELi 位和 TSTARTi 位写“1”和“0”，也不能停止计数。
- 将 TRDIOji 引脚（j=A、B、C、D）用于定时器 RD 输出时的计数停止期间的输出电平如表 8-19 所示。

表 8-19 停止计数时的 TRDIOji 引脚（j=A、B、C、D）的输出电平

停止计数的方法	停止计数时的 TRDIOji 引脚的输出电平
在 CSELi 位为“1”时，通过给 TSTARTi 位写“0”来停止计数。	保持停止计数前的输出电平（在定时器 RD 的互补 PWM 模式或者复位同步 PWM 模式中，输出 TRDFCR 寄存器的 OLS0 位和 OLS1 位选择的初始输出电平）。
当 CSELi 位为“0”时，在 TRDi 寄存器和 TRDGRAi 寄存器比较匹配时停止计数。	保持比较匹配引起输出变化后的电平（在定时器 RD 的互补 PWM 模式或者复位同步 PWM 模式中，输出 TRDFCR 寄存器的 OLS0 位和 OLS1 位选择的初始输出电平）。

备注 i=0、1，j=A、B、C、D

(2) TRDDFi 寄存器（i=0、1）

必须在设定 TRDDFi 寄存器的 DFCK0 位和 DFCK1 位后开始计数。

(3) TRDi 寄存器（i=0、1）

如果 TRDi 寄存器的值变为“0000H”的时序和写 TRDi 寄存器的时序重叠，就优先写寄存器。

8.7.2 模式的切换

- 要在运行过程中切换模式时，必须在进入计数停止状态（将 TSTART0 位和 TSTART1 位置“0”）后进行切换。
- 在将 TSTART0 位和 TSTART1 位从“0”改为“1”前，必须将 TRDIF0 位和 TRDIF1 位置“0”。详细内容请参考“第 18 章 中断功能”。

8.7.3 计数源

- 要切换计数源时，必须在停止计数后进行切换。
[变更步骤]
(1)将 TRDSTR 寄存器的 TSTARTi 位 (i=0、1) 置“0” (停止计数)。
(2)更改 TRDCRi 寄存器的 TCK0~TCK2 位。
- 要选择 f_{HOCO} (64MHz 或者 48MHz) 作为定时器 RD 的计数源时，必须在将外围允许寄存器 1 (PER1) 的 bit4 (TRD0EN) 置位前将 f_{CLK} 设定为 f_{IH} 。如果要将 f_{CLK} 改为 f_{IH} 以外的时钟，就必须在清除外围允许寄存器 1 (PER1) 的 bit4 (TRD0EN) 后进行更改。

8.7.4 输入捕捉功能

- 输入捕捉信号的脉宽必须至少为 3 个定时器 RD 的运行时钟周期。
- 在输入捕捉信号从 TRDIOji 引脚 (j=A、B、C、D) 输入后，需要等待 2~3 个定时器 RD 的运行时钟 (f_{CLK}) 周期，然后将 TRDi 寄存器的值传送到 TRDGRji 寄存器 (没有数字滤波器的情况)。
- 在输入捕捉模式中，如果在 TRDSTR 寄存器的 TRDTSTARTi 位为“0” (停止计数) 时给 TRDIOji 引脚输入 TRDIORki 寄存器的 IOj0 位和 IOj1 位选择的边沿，就在 TRDIOji 输入的有效边沿产生输入捕捉中断请求 (i=0、1, j=A、B、C、D, k=A、C)。

8.7.5 TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚的设定步骤 (i=0、1)

在复位后，TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚的复用 I/O 端口用作输入端口。

- 要从 TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚输出时，必须按照以下步骤进行设定。
[变更步骤]
(1) 进行模式和初始值的设定。
(2) 将 TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚置为允许输出 (TRDOER1 寄存器)。
(3) 将 TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚对应的端口寄存器的位置“0”。
(4) 将 TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚对应的端口模式寄存器的位设定为输出模式 (从 TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚开始输出)。
(5) 开始计数 (将 TSTART0 位和 TSTART1 位置“1”)。
- 要将 TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚对应的端口模式寄存器的位从输出模式改为输入模式时，必须按照以下步骤进行设定。
[变更步骤]
(1) 将 TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚对应的端口模式寄存器的位设定为输入模式 (从 TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚开始输入)。
(2) 设定为输入捕捉功能。
(3) 开始计数 (将 TSTART0 位和 TSTART1 位置“1”)。
- 在将 TRDIOAi、TRDIOBi、TRDIOCi、TRDIODi 引脚从输出模式切换为输入模式时，可能根据引脚的状态进行输入捕捉的运行。当不使用数字滤波器时，在至少经过 2 个 CPU 时钟周期后检测边沿；当使用数字滤波器时，最多需要 5 个数字滤波器的采样时钟周期进行边沿检测。

8.7.6 外部时钟 TRDCLK

TRDCLK 引脚输入的外部时钟的脉宽必须至少为 3 个定时器 RD 的运行时钟周期。

8.7.7 复位同步 PWM 模式

- 当此模式用于马达控制时，必须在 OLS0=OLS1 的条件下使用。
- 要设定为复位同步 PWM 模式时，必须按照以下步骤进行设定。

[变更步骤]

- (1) 将 TRDSTR 寄存器的 TSTART0 位置“0”（停止计数）。
- (2) 将 TRDFCR 寄存器的 CMD1 位和 CMD0 位置“00B”（定时器模式、PWM 模式和 PWM3 模式）。
- (3) 将 CMD1 位和 CMD0 位置“01B”（复位同步 PWM 模式）。
- (4) 重新设定定时器 RD 的其他相关寄存器。

8.7.8 互补 PWM 模式

- 当此模式用于马达控制时，必须在 OLS0=OLS1 的条件下使用。
- 要更改 TRDFCR 寄存器的 CMD0 位和 CMD1 位时，必须按照以下步骤进行更改。

[变更步骤：设定为互补 PWM 模式（包括重新设定）的情况，或者在互补 PWM 模式中更改缓冲寄存器到通用寄存器的数据传送时序的情况]

- (1) 将 TRDSTR 寄存器的 TSTART0 位和 TSTART1 位都置“0”（停止计数）。
- (2) 将 TRDFCR 寄存器的 CMD1 位和 CMD0 位置“00B”（定时器模式、PWM 模式和 PWM3 模式）。
- (3) 将 CMD1 位和 CMD0 位置“10B”或者“11B”（互补 PWM 模式）。
- (4) 重新设定定时器 RD 的其他相关寄存器。

[变更步骤：中止互补 PWM 模式的情况]

- (1) 将 TRDSTR 寄存器的 TSTART0 位和 TSTART1 位都置“0”（停止计数）。
- (2) 将 CMD1 位和 CMD0 位置“00B”（定时器模式、PWM 模式和 PWM3 模式）

- 不能在运行过程中写 TRDGRA0、TRDGRB0、TRDGRA1、TRDGRB1 寄存器。

要更改 PWM 波形时，必须通过缓冲器运行将 TRDGRD0、TRDGRC1、TRDGRD1 寄存器的写入值传送到 TRDGRB0、TRDGRA1、TRDGRB1 寄存器。

但是，在写 TRDGRD0、TRDGRC1 和 TRDGRD1 时，必须在先将 TRDBFD0 位、TRDBFC1 位和 TRDBFD1 位置“0”（通用寄存器），然后给这些寄存器写数据。此后，能将 TRDBFD0 位、TRDBFC1 位和 TRDBFD1 位置“1”（缓冲寄存器）。

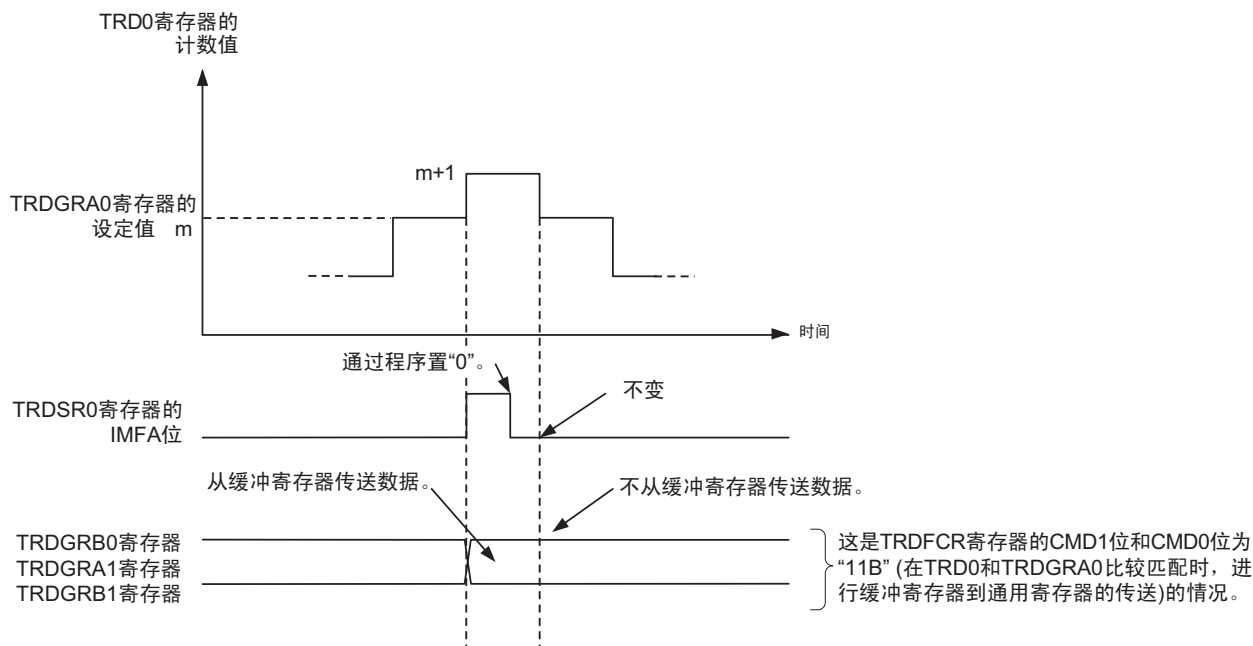
不能更改 PWM 周期。

- 假设 TRDGRA0 寄存器的设定值为 m，则 TRD0 寄存器在从递增计数变为递减计数时进行 $m-1 \rightarrow m \rightarrow m+1 \rightarrow m \rightarrow m-1$ 的计数。

在进行 $m \rightarrow m+1$ 的递增计数时，TRDSR_i 寄存器的 IMFA 位变为“1”。当 TRDFCR 寄存器的 CMD1 位和 CMD0 位为“11B”（互补 PWM 模式，在 TRD0 和 TRDGRA0 寄存器比较匹配时传送缓冲数据）时，将缓冲寄存器（TRDGRD0、TRDGRC1、TRDGRD1）的内容传送到通用寄存器（TRDGRB0、TRDGRA1、TRDGRB1）。

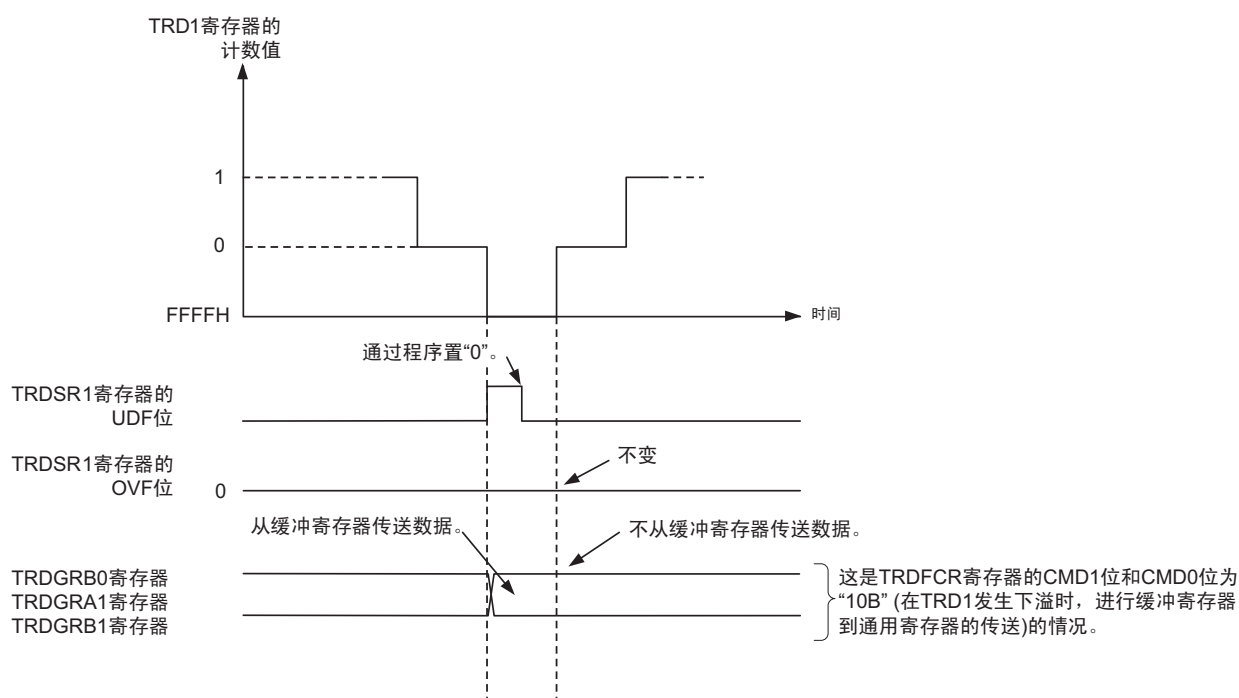
在进行 $m+1 \rightarrow m \rightarrow m-1$ 的递减计数时，IMFA 位不变，并且也不将数据传送到 TRDGRA0 等寄存器。

图 8-64 互补 PWM 模式中的 TRD0 和 TRDGRA0 寄存器比较匹配时的运行例子



- 在从递减计数变为递增计数时, TRD1 进行 $1 \rightarrow 0 \rightarrow \text{FFFFH} \rightarrow 0 \rightarrow 1$ 的计数。
在进行 $1 \rightarrow 0 \rightarrow \text{FFFFH}$ 的递减计数时, TRDSR_i 寄存器的 UDF 位变为“1”。当 TRDFCR 寄存器的 CMD1 位和 CMD0 位为“10B” (互补 PWM 模式, 在 TRD1 发生下溢时传送缓冲数据) 时, 将缓冲寄存器 (TRDGRD0、TRDGRC1、TRDGRD1) 的内容传送到通用寄存器 (TRDGRB0、TRDGRA1、TRDGRB1)。
在进行 $\text{FFFFH} \rightarrow 0 \rightarrow 1$ 的递增计数时, 不将数据传送到 TRDGRB0 等寄存器, 并且 TRDSR_i 寄存器的 OVF 位不变。

图 8-65 互补 PWM 模式中的 TRD1 发生下溢时的运行例子



- 必须通过TRDFCR寄存器的CMD0位和CMD1位选择缓冲寄存器到通用寄存器的数据传送时序。但是，在0%占空比和100%占空比的情况下，与CMD0位和CMD1位的值无关，为以下的传送时序。

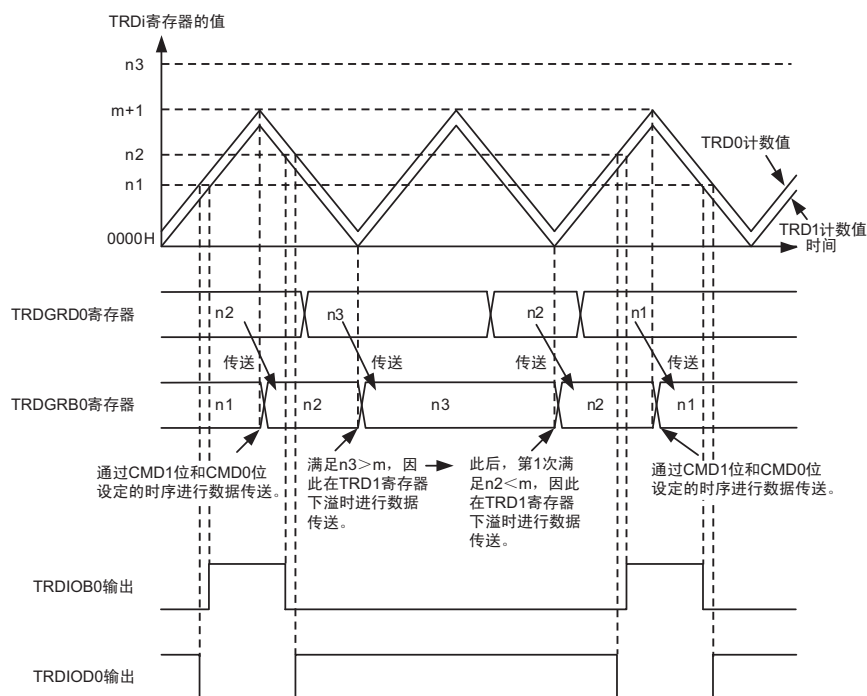
缓冲寄存器的值 $\geq$ TRDGRA0寄存器的值的情况（0%占空比）

在TRD1寄存器发生下溢时进行数据传送。

此后，如果给缓冲寄存器设定值（ $0001H \leq \text{设定值} < \text{TRDGRA0寄存器的值}$ ），就在设定后TRD1寄存器发生第1次下溢时将数据传送到通用寄存器。然后，通过CMD0位和CMD1位选择的时序进行数据传送。

但是，在缓冲寄存器的初始值为“FFFFH”时无法产生0%占空比的波形。要产生0%占空比的波形时，必须通过写缓冲寄存器使缓冲寄存器的值 $\geq$ TRDGRA0寄存器的值。

图 8-66 互补 PWM 模式中的缓冲寄存器的值 $\geq$ TRDGRA0 寄存器的值时的运行例子



备注 m: TRDGRA0寄存器的设定值

上图的条件如下:

- TRDFCR寄存器的CMD1位和CMD0位为“11B”(互补PWM模式，在TRD0寄存器和TRDGRA0寄存器比较匹配时传送缓冲寄存器的数据)。
- TRDFCR寄存器的OLS0位和OLS1位都为“1”(正相和反相都为“H”电平有效)。

如果给缓冲寄存器设定值（设定值 $\geq$ TRDGRA0 寄存器的值），就在 TRD1 计数器发生下溢时将缓冲寄存器的值传送到通用寄存器，并且固定为正相 100% 占空比和反相 0% 占空比的输出电平，而与 CMD0 位的设定无关。

要解除固定的输出电平时，必须给缓冲寄存器设定值（TRD0 寄存器的值 $\leq$ 设定值 $\leq$ (TRDGRA0 的值 - TRD0 寄存器的值)）。在写缓冲器后，与 CMD0 位的设定无关，在 TRD1 计数器发生下溢时将缓冲寄存器的值传送到通用寄存器并且输出 PWM 波形。在输出 PWM 波形后，在 CMD0 位设定的时序将缓冲寄存器的值传送到通用寄存器。

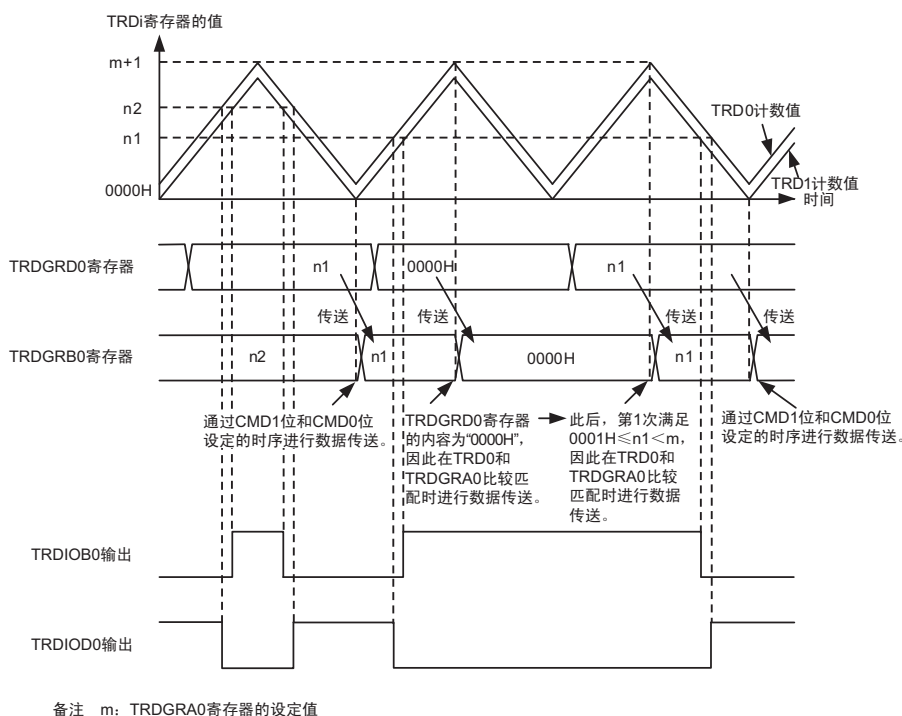
但是，不能用缓冲寄存器的初始值“FFFFH”设定正相 100% 占空比和反相 0% 占空比的输出。也不能直接从正相 100% 占空比和反相 0% 占空比的输出改为正相 0% 占空比和反相 100% 占空比的输出。

缓冲寄存器的值为“0000H”的情况（100%占空比）

在 TRD0 和 TRDGRA0 寄存器比较匹配时进行数据传送。

此后，如果给缓冲寄存器设定值（ $0001H \leq \text{设定值} < \text{TRDGRA0 寄存器的值}$ ），就在设定后 TRD0 和 TRDGRA0 寄存器发生第 1 次比较匹配时，将数据传送到通用寄存器。然后，通过 CMD0 位和 CMD1 位选择的时序进行数据传送。

图 8-67 互补 PWM 模式中的缓冲寄存器的值为“0000H”时的运行例子



上图的条件如下:

- TRDFCR寄存器的CMD1位和CMD0位为“10B”(互补PWM模式，在TRD1寄存器下溢时传送缓冲寄存器的数据)。
- TRDFCR寄存器的OLS0位和OLS1位都为“1”(正相和反相都为“H”电平有效)。

如果给缓冲寄存器写“0000H”，就在 TRD0 寄存器和 TRDGRA0 寄存器比较匹配时将缓冲寄存器的值传送到通用寄存器，并且固定为正相 0% 占空比和反相 100% 占空比的输出电平，而与 CMD0 位的设定无关。

要解除固定的输出电平时，必须给缓冲寄存器设定值（ $\text{TRD0 寄存器的值} \leq \text{设定值} \leq (\text{TRDGRA0 的值} - \text{TRD0 寄存器的值})$ ）。在写缓冲器后，与 CMD0 位的设定无关，在 TRD1 计数器发生下溢时将缓冲寄存器的值传送到通用寄存器并且输出 PWM 波形。在输出 PWM 波形后，在 CMD0 位设定的时序将缓冲寄存器的值传送到通用寄存器。

不能直接从正相 0% 占空比和反相 100% 占空比的输出改为正相 100% 占空比和反相 0% 占空比的输出。

第9章 实时时钟

9.1 实时时钟的功能

实时时钟有以下功能（只限于 64 引脚产品）。

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

能将实时时钟中断信号（INTRTC）用于从 STOP 模式的唤醒以及 A/D 转换器 SNOOZE 模式的触发。

注意 只有在选择副系统时钟（ $f_{\text{SUB}}=32.768\text{kHz}$ ）作为实时时钟的运行时钟的情况下，才能进行年、月、星期、日、小时、分钟和秒的计数。当选择低速内部振荡器时钟（ $f_{\text{IL}}=15\text{kHz}$ ）时，只能使用固定周期中断功能。

选择 f_{IL} 时的固定周期中断间隔用以下计算式进行计算：

$$\text{固定周期(RTCC0寄存器选择的值)} \times f_{\text{SUB}} / f_{\text{IL}}$$

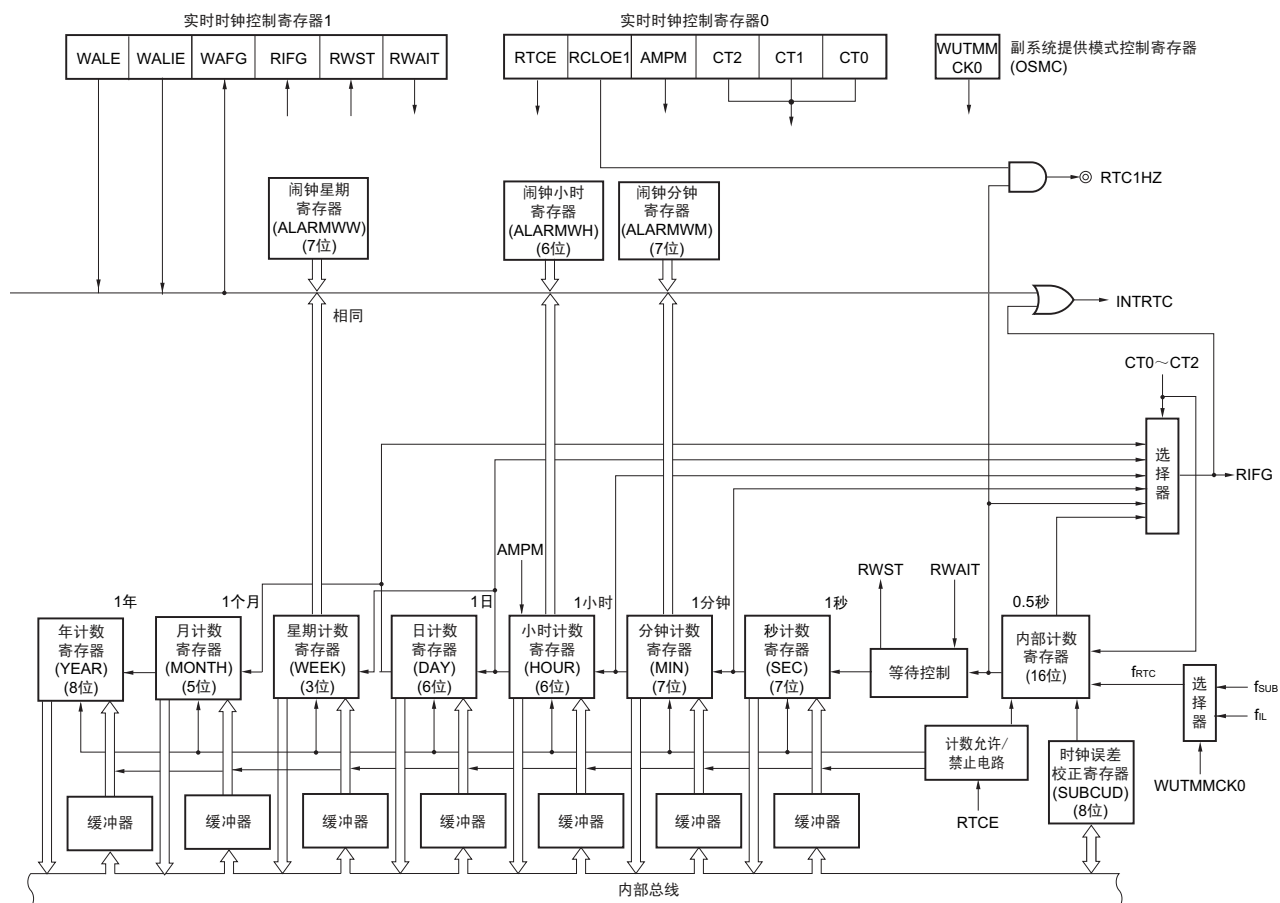
9.2 实时时钟的结构

实时时钟由以下硬件构成。

表 9-1 实时时钟的结构

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

图 9-1 实时时钟的框图



注意 只有在选择副系统时钟（ $f_{\text{SUB}}=32.768\text{kHz}$ ）作为实时时钟的运行时钟的情况下，才能进行年、月、星期、日、小时、分钟和秒的计数。当选择低速内部振荡器时钟（ $f_{\text{IL}}=15\text{kHz}$ ）时，只能使用固定周期中断功能。

选择 f_{IL} 时的固定周期中断间隔用以下计算式进行计算：

$$\text{固定周期(RTCC0寄存器选择的值)} \times f_{\text{SUB}} / f_{\text{IL}}$$

9.3 控制实时时钟的寄存器

通过以下寄存器控制实时时钟。

- 外围允许寄存器0 (PER0)
- 副系统时钟提供模式控制寄存器 (OSMC)
- 实时时钟控制寄存器0 (RTCC0)
- 实时时钟控制寄存器1 (RTCC1)
- 秒计数寄存器 (SEC)
- 分钟计数寄存器 (MIN)
- 小时计数寄存器 (HOUR)
- 日计数寄存器 (DAY)
- 星期计数寄存器 (WEEK)
- 月计数寄存器 (MONTH)
- 年计数寄存器 (YEAR)
- 时钟误差校正寄存器 (SUBCUD)
- 闹钟分钟寄存器 (ALARMWM)
- 闹钟小时寄存器 (ALARMWH)
- 闹钟星期寄存器 (ALARMWW)
- 端口模式寄存器3 (PM3)
- 端口寄存器3 (P3)

9.3.1 外围允许寄存器 0 (PER0)

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

要使用实时时钟时，必须将 bit7 (RTCEN) 置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER0 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 9-2 外围允许寄存器 0 (PER0) 的格式

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

RTCEN	提供实时时钟 (RTC) 和 12 位间隔定时器的输入时钟的控制
0	停止提供输入时钟。 - 不能写实时时钟 (RTC) 和 12 位间隔定时器使用的 SFR。 - 实时时钟 (RTC) 和 12 位间隔定时器处于复位状态。
1	提供输入时钟。 能读写实时时钟 (RTC) 和 12 位间隔定时器使用的 SFR。

注意 1. 如果要使用实时时钟，就必须先在计数时钟 (f_{RTC}) 振荡稳定的状态下将 RTCEN 位置“1”，然后设定以下的寄存器。当 RTCEN 位为“0”时，忽视实时时钟控制寄存器的写操作，而且读取值为初始值（副系统时钟提供模式控制寄存器 (OSMC)、端口模式寄存器 3 (PM3) 和端口寄存器 3 (P3) 除外）。

- 实时时钟控制寄存器 0 (RTCC0)
 - 实时时钟控制寄存器 1 (RTCC1)
 - 秒计数寄存器 (SEC)
 - 分钟计数寄存器 (MIN)
 - 小时计数寄存器 (HOUR)
 - 日计数寄存器 (DAY)
 - 星期计数寄存器 (WEEK)
 - 月计数寄存器 (MONTH)
 - 年计数寄存器 (YEAR)
 - 时钟误差校正寄存器 (SUBCUD)
 - 闹钟分钟寄存器 (ALARMWM)
 - 闹钟小时寄存器 (ALARMWH)
 - 闹钟星期寄存器 (ALARMWW)
2. 能通过将副系统时钟提供模式控制寄存器 (OSMC) 的 RTCLPC 位置“1”，在 STOP 模式或者以副系统时钟运行的 HALT 模式中停止给实时时钟和 12 位间隔定时器以外的外围功能提供副系统时钟。
 3. 必须将以下的位置“0”。
bit1 和 bit6

9.3.2 副系统时钟提供模式控制寄存器（OSMC）

能通过 WUTMMCK0 位选择实时时钟的计数时钟（ f_{RTC} ）。

RTCLPC 位是通过停止不需要的时钟功能来降低功耗的位。有关 RTCLPC 位的设定，请参照“第 5 章 时钟发生电路”。

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 9-3 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	RTCLPC	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	实时时钟、12 位间隔定时器的计数时钟和定时器 RJ 的运行时钟的选择
0	副系统时钟（ f_{SUB} ） - 副系统时钟为实时时钟和 12 位间隔定时器的计数时钟。 - 不能选择低速内部振荡器作为定时器 RJ 的计数源。
1	低速内部振荡器时钟（ f_{IL} ） - 低速内部振荡器时钟为实时时钟和 12 位间隔定时器的计数时钟。 - 能选择低速内部振荡器或者副系统时钟作为定时器 RJ 的计数源。

注意 只有在选择副系统时钟（ $f_{\text{SUB}}=32.768\text{kHz}$ ）作为实时时钟的运行时钟的情况下，才能进行年、月、星期、日、小时、分钟和秒的计数。当选择低速内部振荡器时钟（ $f_{\text{IL}}=15\text{kHz}$ ）时，只能使用固定周期中断功能。

选择 f_{IL} 时的固定周期中断间隔用以下计算式进行计算：

固定周期(RTCC0 寄存器选择的值) $\times f_{\text{SUB}} / f_{\text{IL}}$

9.3.3 实时时钟控制寄存器 0（RTCC0）

这是设定实时时钟的运行开始或者停止、RTC1HZ 引脚的控制、12/24 小时系统和固定周期中断功能的 8 位寄存器。

通过 1 位或者 8 位存储器操作指令设定 RTCC0 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 9-4 实时时钟控制寄存器 0（RTCC0）的格式

地址: FFF9DH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1	0	AMPM	CT2	CT1	CT0

RTCE	实时时钟的运行控制
0	停止计数器的运行。
1	开始计数器的运行。

RCLOE1	RTC1HZ 引脚的输出控制
0	禁止 RTC1HZ 引脚的输出（1Hz）。
1	允许 RTC1HZ 引脚的输出（1Hz）。

AMPM	12 小时系统 /24 小时系统的选择
0	12 小时系统（表示上午或者下午）
1	24 小时系统

- 要更改 AMPM 位的值时，必须在将 RWAIT 位（实时时钟控制寄存器 1（RTCC1）的 bit0）置“1”后进行改写。如果更改 AMPM 位的值，小时计数寄存器（HOUR）的值就变为所设时间系统的对应值。
- 时间位的表示如表 9-2 所示。

CT2	CT1	CT0	固定周期中断（INTRTC）的选择
0	0	0	不使用固定周期中断功能。
0	0	1	0.5 秒一次（与秒累加同步）
0	1	0	1 秒一次（与秒累加同时）
0	1	1	1 分钟一次（每分钟的 00 秒）
1	0	0	1 小时一次（每小时的 00 分 00 秒）
1	0	1	1 日一次（每日的 00 点 00 分 00 秒）
1	1	×	1 个月一次（每月的 1 日上午 00 点 00 分 00 秒）

要在计数器运行中（RTCE=1）更改 CT2～CT0 位的值时，必须在通过中断屏蔽标志寄存器将 INTRTC 设定为禁止中断处理后进行改写，并且必须在改写后清除 RIFG 标志和 RTCIF 标志，然后再设定为允许中断处理。

注意 1. 在 RTCE 位为“1”时，不能更改 RCLOE1 位。

2. 在 RTCE 位为“0”时，即使将 RCLOE1 位置“1”也不输出 1Hz。

备注 ×：忽略

9.3.4 实时时钟控制寄存器 1（RTCC1）

这是控制闹钟中断功能和计数器等待的 8 位寄存器。
通过 1 位或者 8 位存储器操作指令设定 RTCC1 寄存器。
在产生复位信号后，此寄存器的值变为“00H”。

图 9-5 实时时钟控制寄存器 1（RTCC1）的格式 (1/2)

地址：FFF9EH

复位后：00H

R/W

符号

RTCC1

7	6	5	4	3	2	1	0
WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	闹钟的运行控制
0	一致运行无效。
1	一致运行有效。

要在计数器运行中（RTCE=1）并且 WALIE 位为“1”的情况下设定 WALE 位的值时，必须在通过中断屏蔽标志寄存器将 INTRTC 设定为禁止中断处理后进行改写，并且必须在改写后清除 WAFG 标志和 RTCIF 标志。要设定各闹钟寄存器（RTCC1 寄存器的 WALIE 标志、闹钟分钟寄存器（ALARMWM）、闹钟小时寄存器（ALARMWH）和闹钟星期寄存器（ALARMWW））时，必须将 WALE 位置“0”（一致运行无效）。

WALIE	闹钟中断（INTRTC）功能的运行控制
0	不产生闹钟一致中断。
1	产生闹钟一致中断。

WAFG	闹钟检测状态标志
0	闹钟不一致。
1	检测到闹钟一致。

这是表示检测到闹钟一致的状态标志。只在 WALE 位为“1”时有效，在检测到闹钟一致并且经过 1 个 f_{RTC} 时钟后变为“1”。

通过给此标志写“0”来清除此标志。写“1”的操作无效。

图 9-5 实时时钟控制寄存器 1 (RTCC1) 的格式 (2/2)

RIFG	固定周期中断状态标志
0	没有产生固定周期中断。
1	产生固定周期中断。
这是表示产生固定周期中断的状态标志。当产生固定周期中断时，此标志为“1”。 通过给此标志写“0”来清除此标志。写“1”的操作无效。	

RWST	实时时钟的等待状态标志
0	计数器正在运行。
1	正处于计数器的读写模式。
这是表示 RWAIT 位的设定是否有效的状态。 必须在确认此标志为“1”后读写计数器的值。	

RWAIT	实时时钟的等待控制
0	设定为计数器运行。
1	设定为 SEC ~ YEAR 计数器停止运行，进入计数器的读写模式。
此位控制计数器的运行。 要读写计数器的值时，必须给此位写“1”。 因为内部计数器（16 位）继续运行，所以必须在 1 秒内结束读写，然后返回到“0”。 从将 RWAIT 位置“1”到能读写计数器的值（RWST=1）为止，最多需要 1 个 f_{RTC} 时钟的时间。 如果在 RWAIT 位为“1”时发生内部计数器（16 位）上溢，就保持发生上溢的状态，在 RWAIT 位变为“0”后进行递增计数。 但是，当写秒计数寄存器时，不保持发生上溢的状态。	

注意 如果通过位操作指令写 RTCCT1 寄存器，就可能清除 RIFG 标志和 WAFG 标志。因此，必须通过 8 位操作指令写 RTCCT1。为了在写时不清除 RIFG 标志和 WAFG 标志，必须将对应位置“1”（写操作无效）。在不使用 RIFG 标志和 WAFG 标志而且即使值被改写也没有问题的情况下，也可以通过位操作指令写 RTCCT1 寄存器。

备注 1. 固定周期中断和闹钟一致中断使用相同中断源（INTRTC）。在同时使用这 2 个中断的情况下，能在发生 INTRTC 时通过确认固定周期中断状态标志（RIFG）和闹钟检测状态标志（WAFG）来判断发生的是哪个中断。

2. 如果写秒计数寄存器（SEC），就清除内部计数器（16 位）。

9.3.5 秒计数寄存器 (SEC)

这是用 0 ~ 59 (十进制) 表示秒计数值的 8 位寄存器。

通过内部计数器 (16 位) 的上溢进行递增计数。

在写时, 数据先被写到缓冲器, 在经过最多 2 个 f_{RTC} 时钟后被写到计数器。以 BCD 码设定十进制的 00 ~ 59。

通过 8 位存储器操作指令设定 SEC 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 9-6 秒计数寄存器 (SEC) 的格式

地址: FFF92H	复位后: 00H		R/W				
符号	7	6	5	4	3	2	1 0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2 SEC1

注意 要在计数器运行中 (RTCE=1) 读写此寄存器时, 必须按照“9.4.3 实时时钟计数器的读写”记载的步骤进行。

备注 如果写秒计数寄存器 (SEC), 就清除内部计数器 (16 位)。

9.3.6 分钟计数寄存器 (MIN)

这是用 0 ~ 59 (十进制) 表示分钟计数值的 8 位寄存器。

通过秒计数器的上溢进行递增计数。

在写时, 数据先被写到缓冲器, 在经过最多 2 个 f_{RTC} 时钟后被写到计数器。在写操作过程中忽视秒计数寄存器的上溢并且设定为写入值。以 BCD 码设定十进制的 00 ~ 59。

通过 8 位存储器操作指令设定 MIN 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 9-7 分钟计数寄存器 (MIN) 的格式

地址: FFF93H	复位后: 00H		R/W				
符号	7	6	5	4	3	2	1 0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2 MIN1

注意 要在计数器运行中 (RTCE=1) 读写此寄存器时, 必须按照“9.4.3 实时时钟计数器的读写”记载的步骤进行。

9.3.7 小时计数寄存器 (HOUR)

这是用 00 ~ 23 或者 01 ~ 12、21 ~ 32（十进制）表示小时计数值的 8 位寄存器。

通过分钟计数器的上溢进行递增计数。

在写时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。在写操作过程中忽视分钟计数寄存器的上溢并且设定为写入值。

根据实时时钟控制寄存器 0 (RTCC0) 的 bit3 (AMPM) 设定的时间系统，以 BCD 码设定十进制的 00 ~ 23 或者 01 ~ 12、21 ~ 32。

如果更改 AMPM 位的值，HOUR 寄存器的值就变为设定的时间系统所对应的值。

通过 8 位存储器操作指令设定 HOUR 寄存器。

在产生复位信号后，此寄存器的值变为“12H”。

但是，如果在复位后将 AMPM 位置“1”，此寄存器的值就为“00H”。

图 9-8 小时计数寄存器 (HOUR) 的格式

地址: FFF94H		复位后: 12H		R/W				
符号	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

注意 1. 当选择 AMPM 位为“0”（12 小时系统）时，HOUR 寄存器的 bit5 (HOUR20) 表示 AM (0) /PM (1)。

2. 要在计数器运行中 (RTCE=1) 读写此寄存器时，必须按照“9.4.3 实时时钟计数器的读写”记载的步骤进行。

AMPM 位的设定值、小时计数寄存器（HOUR）的值和时间的关系如表 9-2 所示。

表 9-2 时间位的表示

24 小时表示 (AMPM=1)		12 小时表示 (AMPM=0)	
时间	HOUR 寄存器	时间	HOUR 寄存器
0 时	00H	AM 12 时	12H
1 时	01H	AM 1 时	01H
2 时	02H	AM 2 时	02H
3 时	03H	AM 3 时	03H
4 时	04H	AM 4 时	04H
5 时	05H	AM 5 时	05H
6 时	06H	AM 6 时	06H
7 时	07H	AM 7 时	07H
8 时	08H	AM 8 时	08H
9 时	09H	AM 9 时	09H
10 时	10H	AM 10 时	10H
11 时	11H	AM 11 时	11H
12 时	12H	PM 12 时	32H
13 时	13H	PM 1 时	21H
14 时	14H	PM 2 时	22H
15 时	15H	PM 3 时	23H
16 时	16H	PM 4 时	24H
17 时	17H	PM 5 时	25H
18 时	18H	PM 6 时	26H
19 时	19H	PM 7 时	27H
20 时	20H	PM 8 时	28H
21 时	21H	PM 9 时	29H
22 时	22H	PM 10 时	30H
23 时	23H	PM 11 时	31H

当 AMPM 位为“0”时，HOUR 寄存器的值为 12 小时表示；当 AMPM 位为“1”时，HOUR 寄存器的值为 24 小时表示。

在 12 小时表示时，HOUR 寄存器的 bit5 表示上午 / 下午。上午（AM）为“0”，下午（PM）为“1”。

9.3.8 日计数寄存器 (DAY)

这是用 1 ~ 31（十进制）表示日计数值的 8 位寄存器。

通过小时计数器的上溢进行递增计数。

计数器进行以下的计数。

- 01 ~ 31（1、3、5、7、8、10、12月）
- 01 ~ 30（4、6、9、11月）
- 01 ~ 29（2月 闰年）
- 01 ~ 28（2月 平年）

在写时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。在写操作过程中忽视小时计数寄存器的上溢并且设定为写入值。以 BCD 码设定十进制的 01 ~ 31。

通过 8 位存储器操作指令设定 DAY 寄存器。

在产生复位信号后，此寄存器的值变为“01H”。

图 9-9 日计数寄存器 (DAY) 的格式

地址: FFF96H		复位后: 01H		R/W				
符号	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

注意 要在计数器运行中 (RTCE=1) 读写此寄存器时，必须按照“9.4.3 实时时钟计数器的读写”记载的步骤进行。

9.3.9 星期计数寄存器（WEEK）

这是用 0～6（十进制）表示星期计数值的 8 位寄存器。

与日计数器同步进行递增计数。

在写时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。以 BCD 码设定十进制的 00～06。

通过 8 位存储器操作指令设定 WEEK 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 9-10 星期计数寄存器（WEEK）的格式

地址: FFF95H	复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

注意 1. 月计数寄存器（MONTH）和日计数寄存器（DAY）的对应值不自动保存到星期寄存器（WEEK）。

必须在解除复位后进行以下的设定：

星期	WEEK
星期日	00H
星期一	01H
星期二	02H
星期三	03H
星期四	04H
星期五	05H
星期六	06H

2. 要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“9.4.3 实时时钟计数器的读写”记载的步骤进行。

9.3.10 月计数寄存器 (MONTH)

这是用 1 ~ 12 (十进制) 表示月计数值的 8 位寄存器。

通过日计数器的上溢进行递增计数。

在写时, 数据先被写到缓冲器, 在经过最多 2 个 f_{RTC} 时钟后被写到计数器。在写操作过程中忽视日计数寄存器的上溢并且设定为写入值。以 BCD 码设定十进制的 01 ~ 12。

通过 8 位存储器操作指令设定 MONTH 寄存器。

在产生复位信号后, 此寄存器的值变为“01H”。

图 9-11 月计数寄存器 (MONTH) 的格式

地址: FFF97H	复位后: 01H		R/W					
符号	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

注意 要在计数器运行中 (RTCE=1) 读写此寄存器时, 必须按照“9.4.3 实时时钟计数器的读写”记载的步骤进行。

9.3.11 年计数寄存器 (YEAR)

这是用 0 ~ 99 (十进制) 表示年计数值的 8 位寄存器。

通过月计数器 (MONTH) 的上溢进行递增计数。

00、04、08、.....、92、96 是闰年。

在写时, 数据先被写到缓冲器, 在经过最多 2 个 f_{RTC} 时钟后被写到计数器。在写操作过程中忽视 MONTH 寄存器的上溢并且设定为写入值。以 BCD 码设定十进制的 00 ~ 99。

通过 8 位存储器操作指令设定 YEAR 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 9-12 年计数寄存器 (YEAR) 的格式

地址: FFF98H	复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

注意 要在计数器运行中 (RTCE=1) 读写此寄存器时, 必须按照“9.4.3 实时时钟计数器的读写”记载的步骤进行。

9.3.12 时钟误差校正寄存器（SUBCUD）

这是能通过改变从内部计数器（16 位）到秒计数寄存器（SEC）的上溢值（基准值：7FFFH）来高精度地校正时钟快慢的寄存器。

通过 8 位存储器操作指令设定 SUBCUD 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 9-13 时钟误差校正寄存器（SUBCUD）的格式

地址：FFF99H	复位后：00H	R/W						
符号	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	时钟误差的校正时刻的设定
0	在秒位为“00”、“20”、“40”时（每 20 秒）进行时钟误差的校正。
1	只在秒位为“00”时（每 60 秒）进行时钟误差的校正。
在以下所示期间禁止写 SUBCUD 寄存器： • DVE=0：SEC=00H、20H、40H 的期间 • DVE=1：SEC=00H 的期间	

F6	时钟误差校正值的设定
0	$\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2$ 增加
1	$\{(/F5, /F4, /F3, /F2, /F1, /F0) + 1\} \times 2$ 减少
当 (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *) 时，不进行时钟误差的校正。* 是“0”或者“1”。 /F5 ~ /F0 是各位取反后的值（“111100”时，为“000011”）。 校正值的范围：(F6=0) 2、4、6、8、.....、120、122、124 (F6=1) -2、-4、-6、-8、.....、-120、-122、-124	

能通过时钟误差校正寄存器（SUBCUD）进行校正的范围如下所示。

	DEV=0（每 20 秒的校正）	DEV=1（每 60 秒的校正）
能校正的范围	-189.2ppm ~ 189.2ppm	-63.1ppm ~ 63.1ppm
最大量化误差	±1.53ppm	±0.51ppm
最小分辨率	±3.05ppm	±1.02ppm

备注 当校正范围超出 -63.1ppm ~ 63.1ppm 的范围时，必须将 DEV 位置“0”。

9.3.13 闹钟分钟寄存器 (ALARMWM)

这是设定闹钟分钟的寄存器。

通过 8 位存储器操作指令设定 ALARMWM 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

注意 以 BCD 码设定十进制的 00 ~ 59。如果设定范围以外的值，就不检测闹钟。

图 9-14 闹钟分钟寄存器 (ALARMWM) 的格式

地址: FFF9AH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

9.3.14 闹钟小时寄存器 (ALARMWH)

这是设定闹钟小时的寄存器。

通过 8 位存储器操作指令设定 ALARMWH 寄存器。

在产生复位信号后，此寄存器的值变为“12H”。

但是，如果在复位后将 AMPM 位置“1”，此寄存器的值就为“00H”。

注意 以 BCD 码设定十进制的 00 ~ 23 或者 01 ~ 12、21 ~ 32。如果设定范围以外的值，就不检测闹钟。

图 9-15 闹钟小时寄存器 (ALARMWH) 的格式

地址: FFF9BH	复位后: 12H	R/W						
符号	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

注意 当选择 AMPM 位为“0”（12 小时系统）时，ALARMWH 寄存器的 bit5（WH20）表示 AM（0）/PM（1）。

9.3.15 闹钟星期寄存器 (ALARMWW)

这是设定闹钟星期的寄存器。

通过 8 位存储器操作指令设定 ALARMWW 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 9-16 闹钟星期寄存器 (ALARMWW) 的格式

地址: FFF9CH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

闹钟时间的设定例子如下所示。

闹钟设定时间	星期							12 小时表示				24 小时表示			
	星期日	星期一	星期二	星期三	星期四	星期五	星期六								
	W	W	W	W	W	W	W	10 时	1 时	10 分	1 分	10 时	1 时	10 分	1 分
	W	W	W	W	W	W	W								
	0	1	2	3	4	5	6								
每天 上午 0 时 00 分	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
每天 上午 1 时 30 分	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
每天 上午 11 时 59 分	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
星期一～星期五 下午 0 时 00 分	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
星期日 下午 1 时 30 分	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
星期一、星期三、星期五 下午 11 时 59 分	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

9.3.16 端口模式寄存器 3（PM3）

通过 1 位或者 8 位存储器操作指令设定 PM3 寄存器。
在产生复位信号后，此寄存器的值变为“FFH”。
在用作 RTC1Hz 引脚的 1Hz 输出时，必须将 PM30 位置“0”。

图 9-17 端口寄存器 3（PM3）的格式

地址: FFF23H	复位后: FFH	R/W						
符号	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	PM30

9.3.17 端口寄存器 3（P3）

通过 1 位或者 8 位存储器操作指令设定 P3 寄存器。
在产生复位信号后，此寄存器的值变为“00H”。
在用作 RTC1Hz 引脚的 1Hz 输出时，必须将 P30 位置“0”。

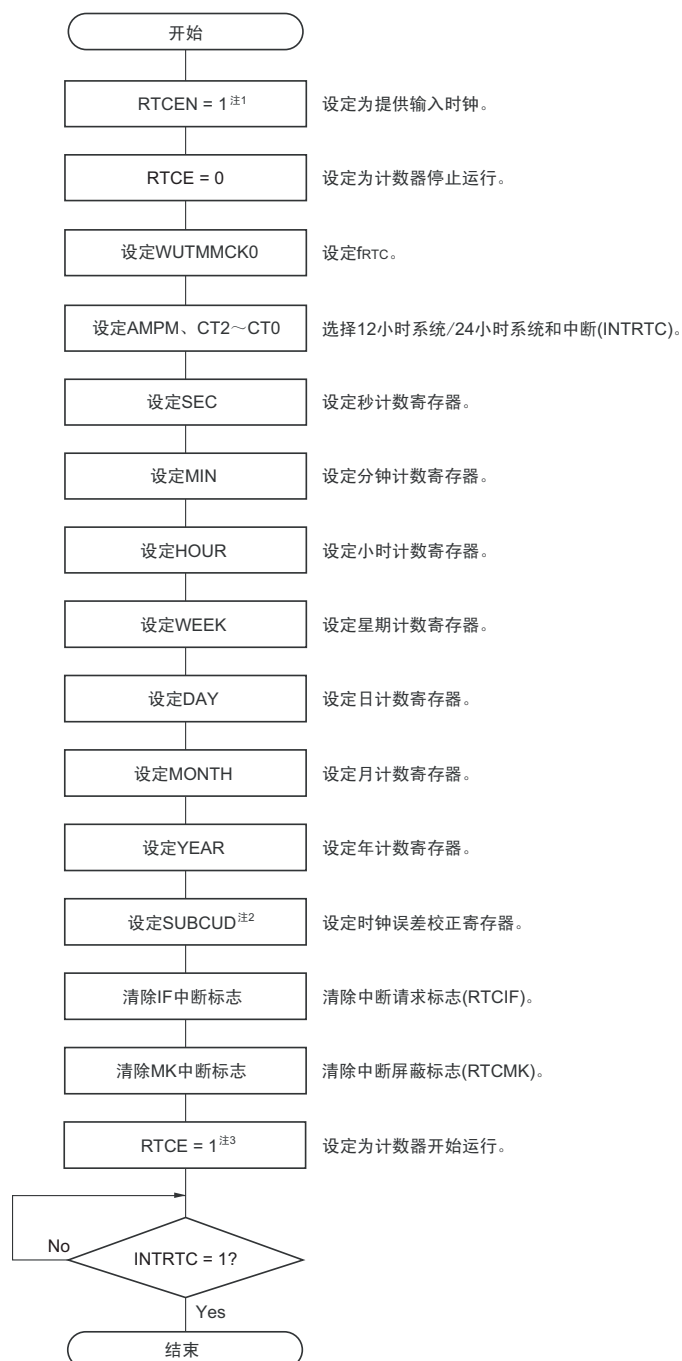
图 9-18 端口寄存器 3（P3）的格式

地址: FFF03H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
P3	0	0	0	0	0	0	P31	P30

9.4 实时时钟的运行

9.4.1 实时时钟的运行开始

图 9-19 实时时钟的运行开始步骤



注 1. 必须先将在计数时钟 (f_{RTC}) 振荡稳定的状态下将 RTCEN 位置“1”。

2. 这只是需要校正时钟误差的情况。有关校正值的计算方法，请参照“9.4.6 实时时钟的时钟误差校正例子”。

3. 在 RTCE 位为“1”后不等待 INTRTC 位变为“1”而转移到 HALT/STOP 模式的情况下，请确认“9.4.2 开始运行后 HALT/STOP 模式的转移”的步骤。

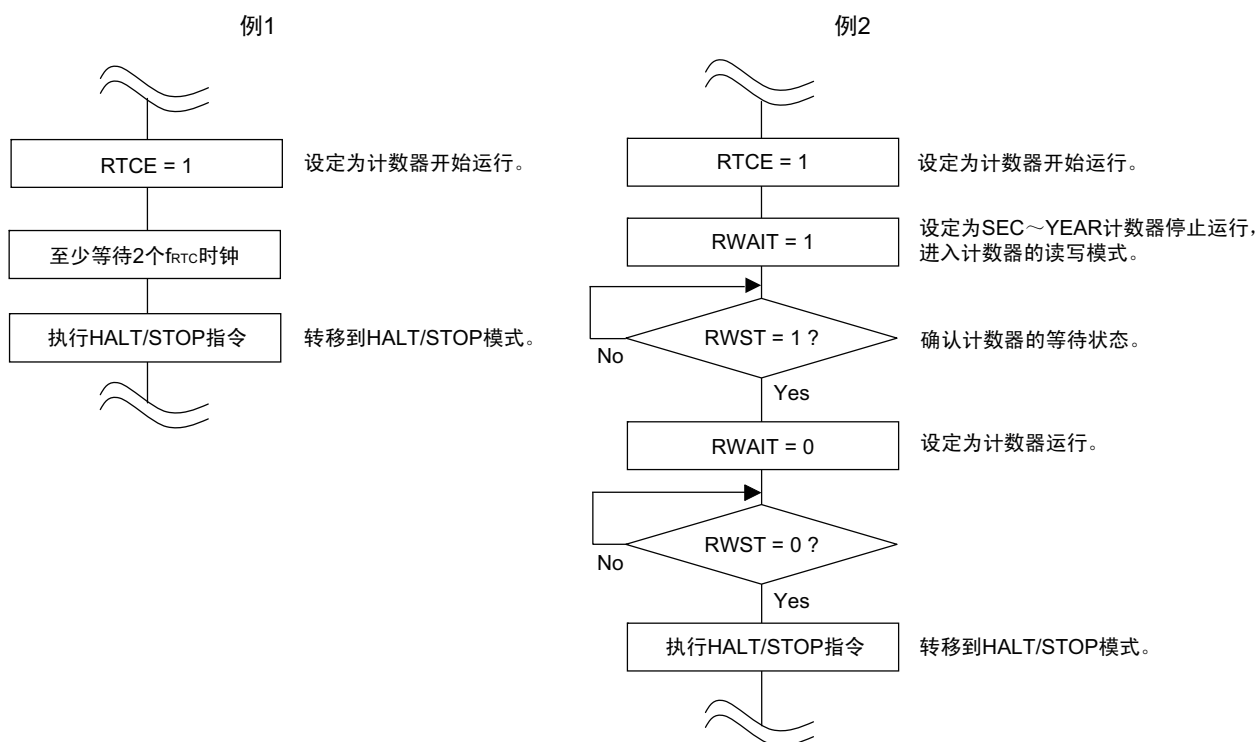
9.4.2 开始运行后 HALT/STOP 模式的转移

要在将 RTCE 位置“1”后立即转移到 HALT/STOP 模式时，必须进行以下某种处理。

但是，在将 RTCE 位置“1”后，如果要在发生 INTRTC 中断后转移到 HALT/STOP 模式，就不需要这些处理。

- 在将 RTCE 位置“1”后至少经过 2 个计数时钟 (f_{RTC}) 之后转移到 HALT/STOP 模式（参照图 9-20 的例 1）。
- 在将 RTCE 位置“1”后将 RWAIT 位置“1”，通过轮询确认 RWST 位变为“1”。然后，将 RWAIT 位置“0”并且再次通过轮询确认 RWST 位变为“0”，然后转移到 HALT/STOP 模式（参照图 9-20 的例 2）。

图 9-20 将 RTCE 位置“1”后的 HALT/STOP 模式的转移步骤

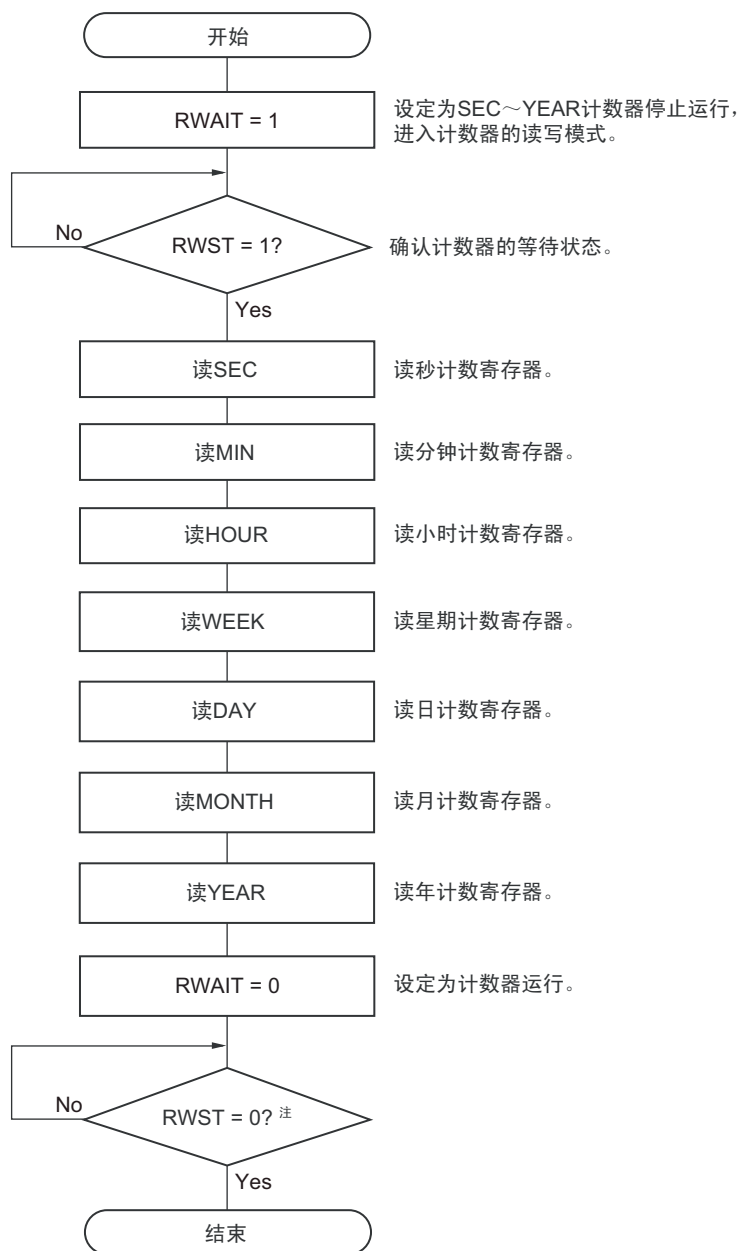


9.4.3 实时时钟计数器的读写

必须先将 RWAIT 位置“1”，然后读写计数器。

必须在读写计数器后将 RWAIT 位置“0”。

图 9-21 实时时钟计数器的读操作步骤

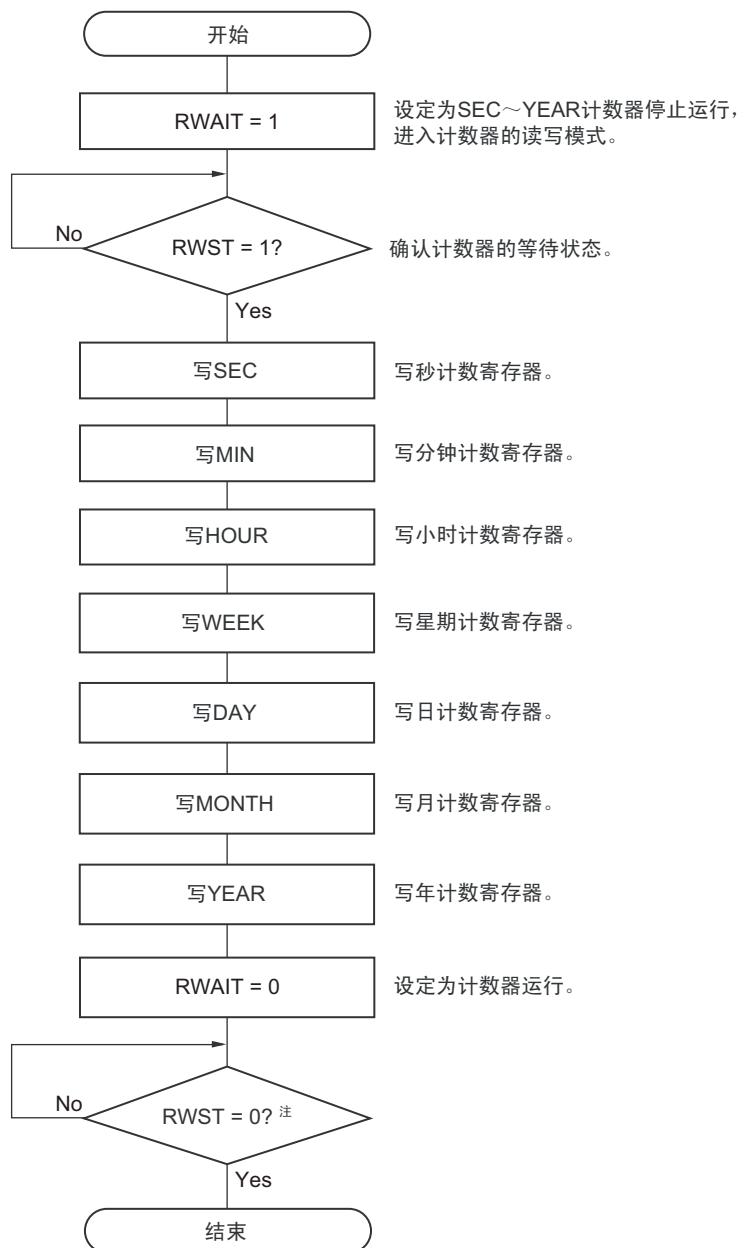


注 必须在转移到 HALT/STOP 模式前，确认 RWST 位为“0”。

注意 必须在 1 秒内进行将 RWAIT 位置“1”到 RWAIT 位置“0”的处理。

备注 不限制秒计数寄存器（SEC）、分钟计数寄存器（MIN）、小时计数寄存器（HOUR）、星期计数寄存器（WEEK）、日计数寄存器（DAY）、月计数寄存器（MONTH）和年计数寄存器（YEAR）的读操作顺序。
可以不读全部寄存器而只读部分寄存器。

图 9-22 实时时钟计数器的写操作步骤



注 必须在转移到 HALT/STOP 模式前，确认 RWST 位为“0”。

注意 1. 必须在 1 秒内进行将 RWAIT 位置“1”到 RWAIT 位置“0”的处理。

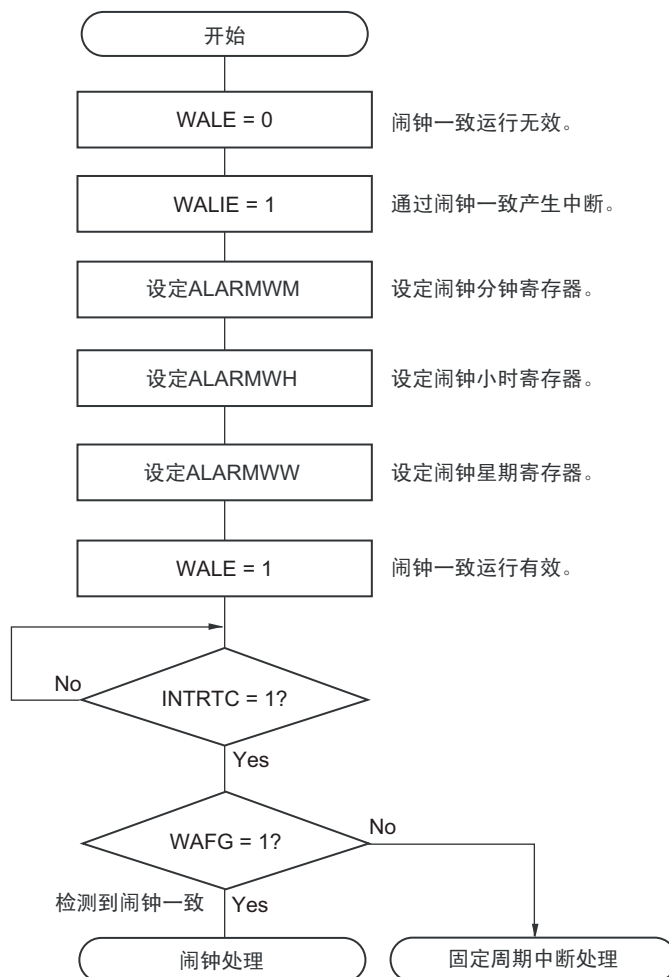
2. 要在计数器运行中 (RTCE=1) 改写 SEC、MIN、HOUR、WEEK、DAY、MONTH、YEAR 寄存器时，必须在通过中断屏蔽标志寄存器将 INTRTC 设定为禁止中断处理后进行改写，并且必须在改写后清除 WAFG 标志、RIFG 标志和 RTCIF 标志。

备注 不限制秒计数寄存器 (SEC)、分钟计数寄存器 (MIN)、小时计数寄存器 (HOUR)、星期计数寄存器 (WEEK)、日计数寄存器 (DAY)、月计数寄存器 (MONTH) 和年计数寄存器 (YEAR) 的写操作顺序。
可以不设定全部寄存器而只改写部分寄存器。

9.4.4 实时时钟的闹钟设定

必须先将 WALE 位置“0”（闹钟运行无效），然后设定闹钟时间。

图 9-23 闹钟设定步骤

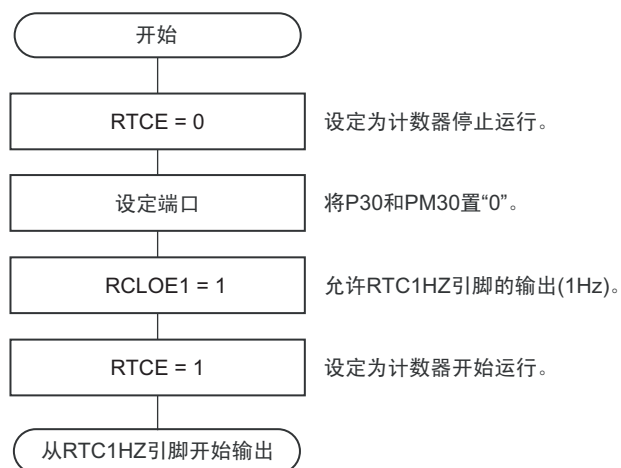


备注 1. 不限制闹钟分钟寄存器（ALARMWM）、闹钟小时寄存器（ALARMWH）和闹钟星期寄存器（ALARMWW）的写操作顺序。

2. 固定周期中断和闹钟一致中断使用相同中断源（INTRTC）。在同时使用这 2 个中断的情况下，能在发生 INTRTC 时通过确认固定周期中断状态标志（RIFG）和闹钟检测状态标志（WAFG）来判断发生的是哪个中断。

9.4.5 实时时钟的 1Hz 输出

图 9-24 1Hz 输出的设定步骤



注意 必须先在计数时钟 (f_{SUB}) 振荡稳定的状态下将 RTCEN 位置 “1”。

9.4.6 实时时钟的时钟误差校正例子

能通过给时钟误差校正寄存器设定值进行高精度的时钟快慢校正。

校正值的计算方法的例子

校正内部计数器（16位）的计数值时的校正值能用以下计算式进行计算。

当校正范围超出 $-63.1\text{ppm} \sim 63.1\text{ppm}$ 的范围时，必须将 DEV 位置“0”。

（DEV=0 的情况）

$$\text{校正值注} = 1 \text{ 分钟的校正计数值} \div 3 = (\text{振荡频率} \div \text{目标频率} - 1) \times 32768 \times 60 \div 3$$

（DEV=1 的情况）

$$\text{校正值注} = 1 \text{ 分钟的校正计数值} = (\text{振荡频率} \div \text{目标频率} - 1) \times 32768 \times 60$$

注 校正值是根据时钟误差校正寄存器（SUBCUD）的 bit6 ~ 0 的值计算的时钟误差校正值。

（F6=0 的情况）校正值 = $\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2$

（F6=1 的情况）校正值 = $-\{(F5, F4, F3, F2, F1, F0) + 1\} \times 2$

当 $(F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *)$ 时，不进行时钟误差的校正。* 是“0”或者“1”。

/F5 ~ /F0 是各位取反后的值（“111100”时，为“000011”）。

备注 1. 校正值为 2、4、6、8、.....、120、122、124 或者 -2、-4、-6、-8、.....、-120、-122、-124。

2. 振荡频率是计数时钟（ f_{RTC} ）的值，能用以下计算式进行计算：

时钟误差校正寄存器为初始值（“00H”）时的 RTC1HZ 引脚的输出频率 $\times 32768$

3. 目标频率是使用时钟误差校正寄存器进行校正后的频率。

校正例子①

从 32772.3Hz 校正到 32768Hz (32772.3Hz-131.2ppm) 的例子

【振荡频率的测量】

通过从 PCLBUZ0 引脚输出大约 32.768kHz 的信号或者在时钟误差校正寄存器 (SUBCUD) 为初始值 (“00H”) 时从 RTC1HZ 引脚输出大约 1Hz 的信号来测量各产品的振荡频率注。

注 有关 RTC1Hz 输出的设定步骤, 请参照“9.4.5 实时时钟的 1Hz 输出”。有关从 PCLBUZ0 引脚输出大约 32kHz 的信号设定步骤, 请参照“11.4 时钟输出 / 蜂鸣器输出控制电路的运行”。

【校正值的计算】

(PCLBUZ0 引脚的输出频率为 32772.3Hz 的情况)

假设目标频率为 32768Hz (32772.3Hz-131.2ppm), 因为 -131.2ppm 为校正范围 $\leq -63.1\text{ppm}$, 所以 DEV=0。适用 DEV 位为“0”时的校正值的计算式。

$$\begin{aligned}\text{校正值} &= 1 \text{ 分钟的校正计数值} \div 3 = (\text{振荡频率} \div \text{目标频率} - 1) \times 32768 \times 60 \div 3 \\ &= (32772.3 \div 32768 - 1) \times 32768 \times 60 \div 3 \\ &= 86\end{aligned}$$

【(F6 ~ F0) 的设定值的计算】

(校正值 = 86 的情况)

因为校正值大于 0 (变慢的情况), 所以 F6=0。

根据校正值计算 (F5, F4, F3, F2, F1, F0)。

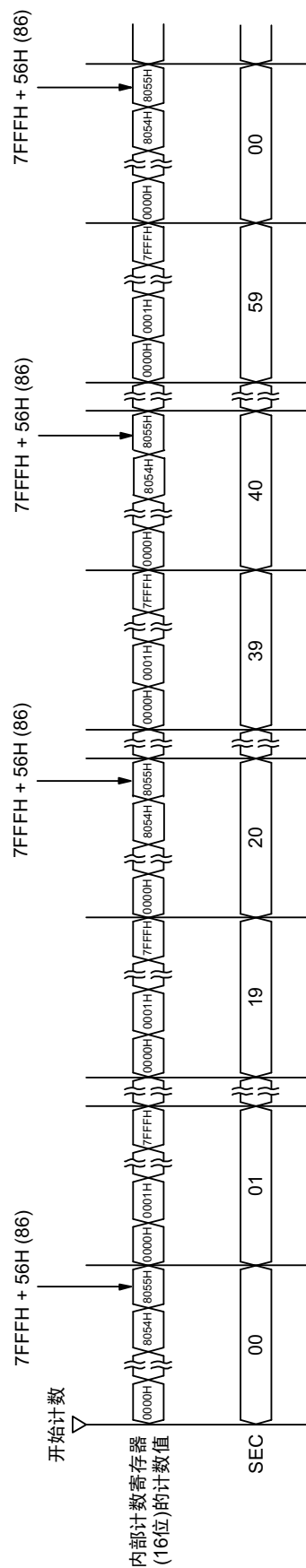
$$\begin{aligned}\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2 &= 86 \\ (F5, F4, F3, F2, F1, F0) &= 44 \\ (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 0, 0)\end{aligned}$$

因此, 从 32772.3Hz 校正到 32768Hz (32772.3Hz-131.2ppm) 的情况如下:

通过 DEV=0 并且校正值 = 86 (SUBCUD 寄存器的 bit6 ~ 0: 0101100) 来设定校正寄存器, 就能校正到 32768Hz (0ppm)。

(DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0) 时的校正如图 9-25 所示。

图 9-25 (DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0) 时的校正



校正例子②

从 32767.4Hz 校正到 32768Hz (32767.4Hz+18.3ppm) 的例子

【振荡频率的测量】

在时钟误差校正寄存器 (SUBCUD) 为初始值 (“00H”) 时, 通过从 RTC1HZ 引脚输出大约 1Hz 的信号来测量各产品的振荡频率注。

注 有关 RTC1Hz 输出的设定步骤, 请参照“9.4.5 实时时钟的 1Hz 输出”。

【校正值的计算】

(RTC1HZ 引脚的输出频率为 0.9999817Hz 的情况)

$$\text{振荡频率} = 32768 \times 0.9999817 \approx 32767.4\text{Hz}$$

假设目标频率为 32768Hz (32767.4Hz+18.3ppm) 并且 DEV=1。

适用 DEV 位为“1”时的校正值的计算式。

$$\begin{aligned}\text{校正} = 1 \text{ 分钟的校正计数值} &= (\text{振荡频率} \div \text{目标频率} - 1) \times 32768 \times 60 \\ &= (32767.4 \div 32768 - 1) \times 32768 \times 60 \\ &= -36\end{aligned}$$

【(F6 ~ F0) 的设定值的计算】

(校正 = -36 的情况)

因为校正 < 0 (变快的情况), 所以 F6=1。

根据校正计算 (F5, F4, F3, F2, F1, F0)。

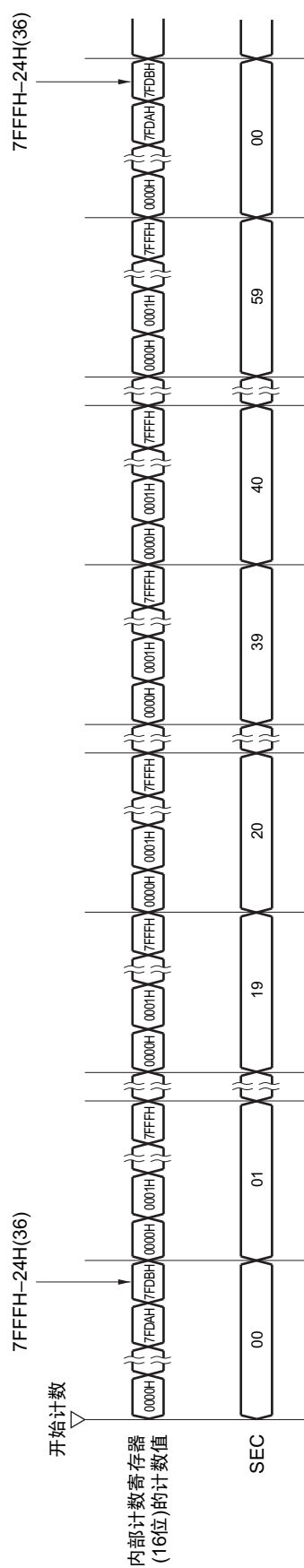
$$\begin{aligned}- \{ (/F5, /F4, /F3, /F2, /F1, /F0) + 1 \} \times 2 &= -36 \\ (/F5, /F4, /F3, /F2, /F1, /F0) &= 17 \\ (/F5, /F4, /F3, /F2, /F1, /F0) &= (0, 1, 0, 0, 0, 1) \\ (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 1, 0)\end{aligned}$$

因此, 从 32767.4Hz 校正到 32768Hz (32767.4Hz+18.3ppm) 的情况如下:

通过 DEV=1 并且校正 = -36 (SUBCUD 寄存器的 bit6 ~ 0: 1101110) 来设定校正寄存器, 就能校正到 32768Hz (0ppm)。

(DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0) 时的校正如图 9-26 所示。

图 9-26 (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0) 时的校正



第 10 章 12 位间隔定时器

10.1 12 位间隔定时器的功能

按事先设定的任意时间间隔产生中断（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。

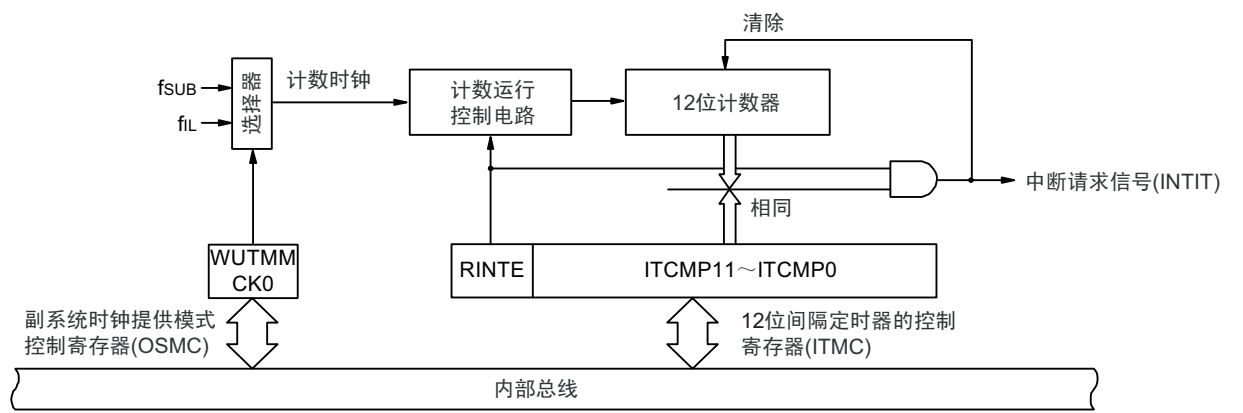
10.2 12 位间隔定时器的结构

12 位间隔定时器由以下硬件构成。

表 10-1 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 10-1 12 位间隔定时器的框图



10.3 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）
- 12 位间隔定时器的控制寄存器（ITMC）

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 12 位间隔定时器时，必须将 bit7（RTCEN）置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER0 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 10-2 外围允许寄存器 0（PER0）的格式

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

RTCEN	提供实时时钟（RTC）和 12 位间隔定时器的输入时钟的控制
0	停止提供时钟。 • 不能写实时时钟（RTC）和 12 位间隔定时器使用的 SFR。 • 实时时钟（RTC）和 12 位间隔定时器处于复位状态。
1	允许提供输入时钟。 • 能读写实时时钟（RTC）和 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，就必须先在计数时钟振荡稳定的状态下将 RTCEN 位置“1”，然后设定以下的寄存器。当 RTCEN 位为“0”时，忽视 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）。
- 12 位间隔定时器的控制寄存器（ITMC）
2. 能通过将副系统时钟提供模式控制寄存器（OSMC）的 RTCLPC 位置“1”，在 STOP 模式或者以副系统时钟运行的 HALT 模式中停止给实时时钟和 12 位间隔定时器以外的外围功能提供时钟。
3. 必须将以下的位置“0”。
- bit1 和 bit6

10.3.2 副系统时钟提供模式控制寄存器（OSMC）

能通过 WUTMMCK0 位选择 12 位间隔定时器、实时时钟和定时器 RJ 的运行时钟。

RTCLPC 位是通过停止不需要的时钟功能来降低功耗的位。有关 RTCLPC 位的设定，请参照“第 5 章 时钟发生电路”。

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 10-3 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	RTCLPC	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	实时时钟、12 位间隔定时器和定时器 RJ 的运行时钟的选择
0	副系统时钟（f _{SUB} ） - 副系统时钟为实时时钟和 12 位间隔定时器的运行时钟。 - 不能选择低速内部振荡器作为定时器 RJ 的计数源。
1	低速内部振荡器时钟（f _{IL} ） - 低速内部振荡器时钟为实时时钟和 12 位间隔定时器的运行时钟。 - 能选择低速内部振荡器或者副系统时钟作为定时器 RJ 的计数源。

10.3.3 12 位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 10-4 12 位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

- ITCMP11 ~ ITCMP0 = 001H, 计数时钟: $f_{SUB}=32.768\text{kHz}$
 $1/32.768[\text{kHz}] \times (1+1) = 0.06103515625[\text{ms}] \approx 61.03[\mu\text{s}]$
- ITCMP11 ~ ITCMP0 = FFFH, 计数时钟: $f_{SUB}=32.768\text{kHz}$
 $1/32.768[\text{kHz}] \times (4095+1) = 125[\text{ms}]$

- 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来禁止 INTIT 中断处理后改写 RINTE 位。要重新开始运行（从“0”改为“1”）时，必须在清除 ITIF 标志后允许中断处理。
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. 要在从待机模式返回后设定 RINTE 位并且再次转移到待机模式时，必须在确认 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。

10.4 12 位间隔定时器的运行

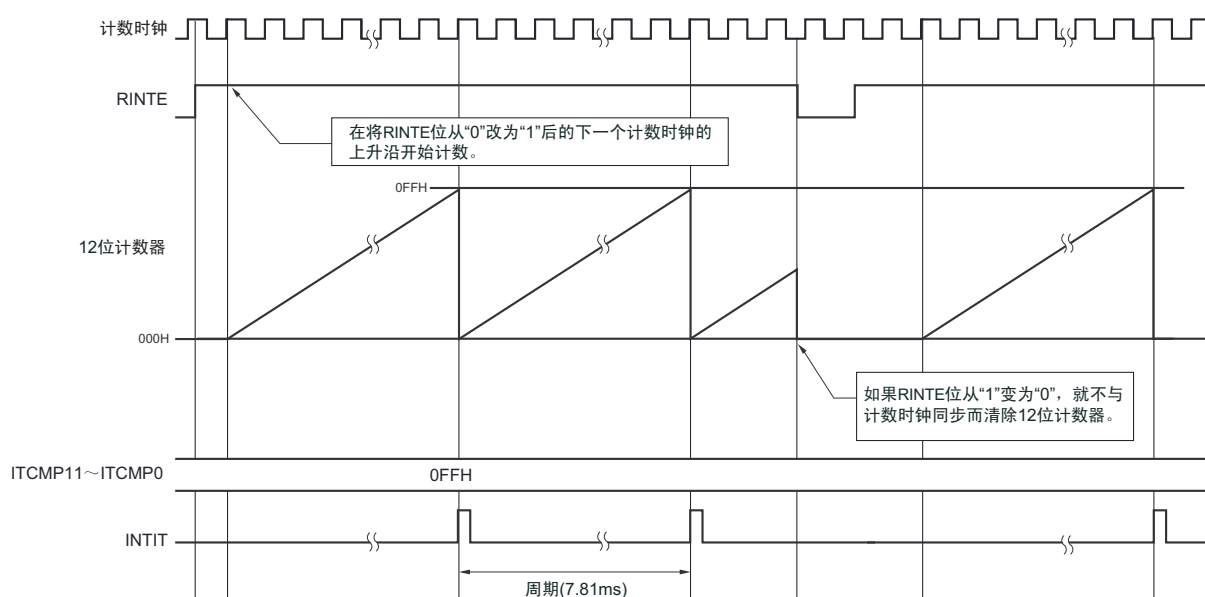
10.4.1 12 位间隔定时器的运行时序

以 ITCMP11 ~ ITCMP0 位设定的计数值为间隔，作为重复产生中断请求 (INTIT) 的 12 位间隔定时器运行。如果将 RINTE 位置“1”，12 位计数器就开始计数。

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

12 位间隔定时器的基本运行如图 10-5 所示。

图 10-5 12 位间隔定时器的运行时序
(ITCMP11 ~ ITCMP0=0FFH, 计数时钟: $f_{SUB}=32.768\text{kHz}$)

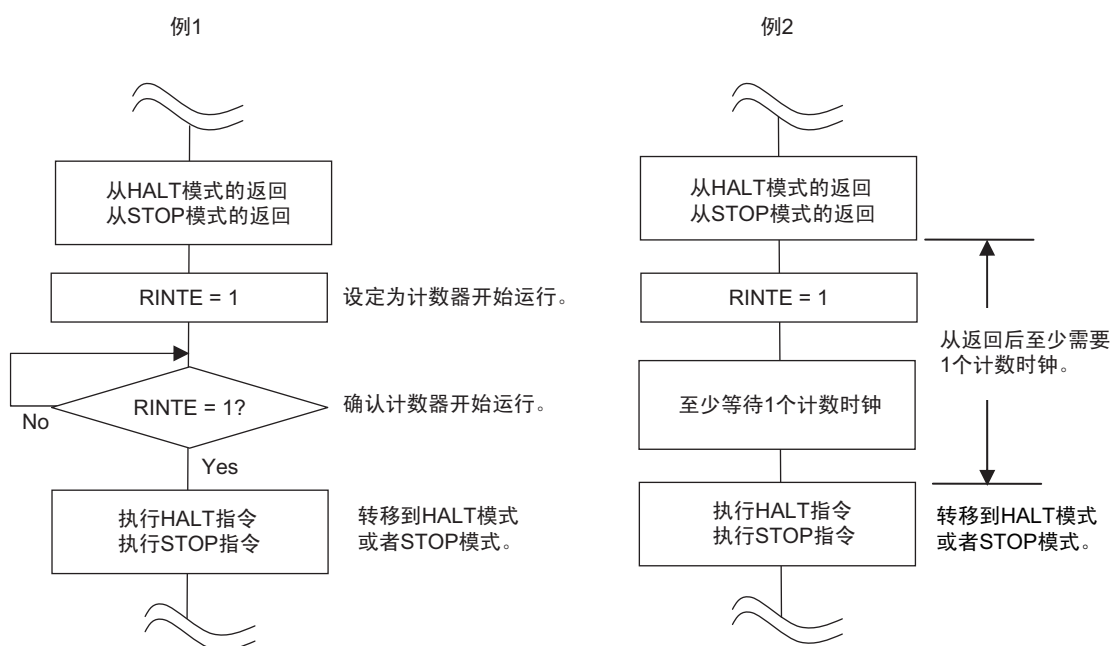


10.4.2 从HALT/STOP模式返回后开始计数器的运行并且再次向HALT/STOP模式的转移

要在从 HALT 模式或者 STOP 模式返回后将 RINTE 位置“1”并且再次转移到 HALT 模式或者 STOP 模式时，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，然后再转移到 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 10-6 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转移到 HALT 模式或者 STOP 模式（参照图 10-6 的例 2）。

图 10-6 将 RINTE 位置“1”后的 HALT 模式或者 STOP 模式的转移步骤



第 11 章 时钟输出 / 蜂鸣器输出控制电路

注意 本章的下述内容主要针对 64 引脚产品进行说明。

11.1 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。

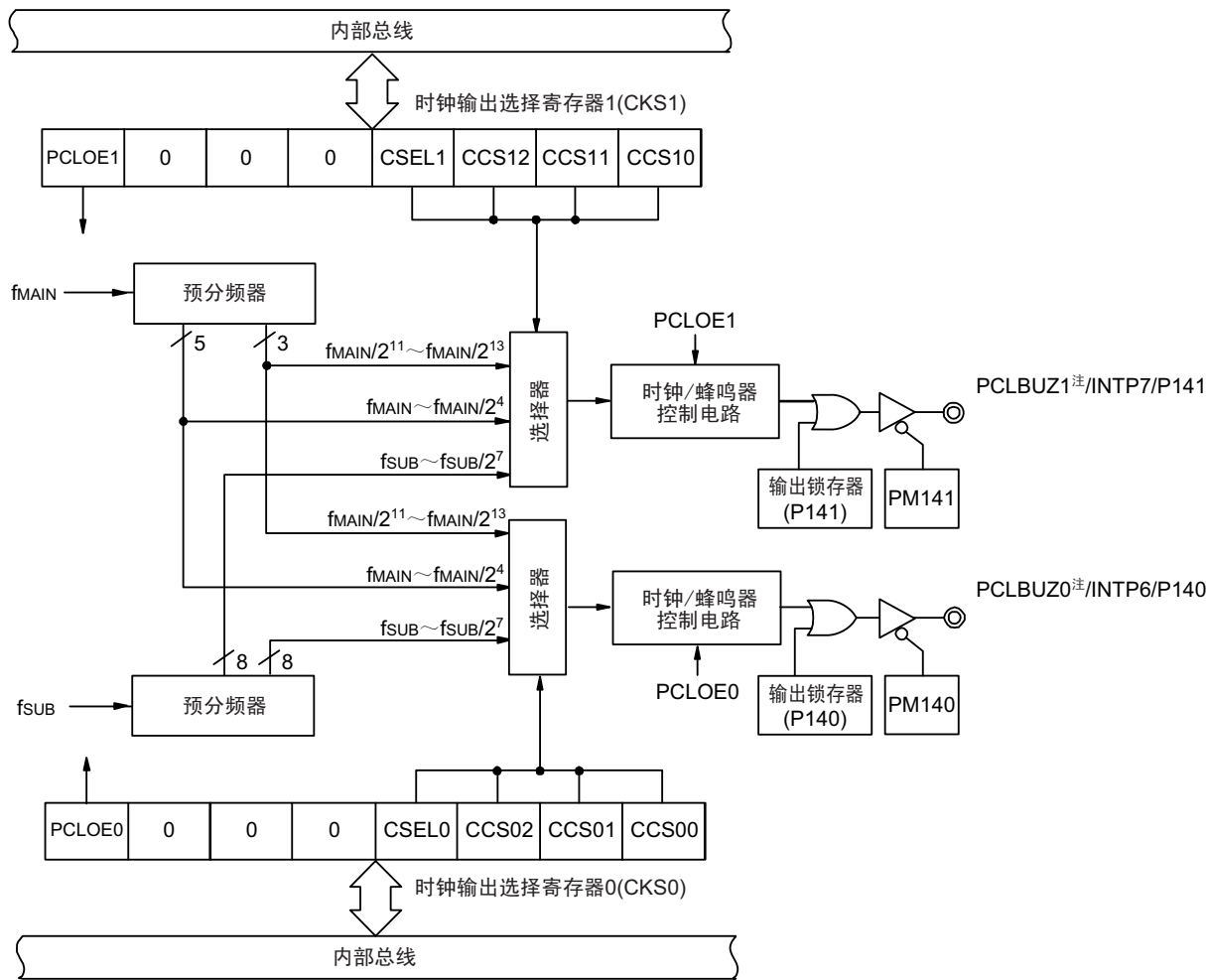
PCLBUZn 引脚输出由时钟输出选择寄存器 n (CKSn) 选择的时钟。

时钟输出 / 蜂鸣器输出控制电路的框图如图 11-1 所示。

注意 在副系统时钟提供模式控制寄存器 (OSMC) 的 RTCLPC 位为“1”时并且在 CPU 以副系统时钟 (f_{SUB}) 运行的 HALT 模式中，不能从 PCLBUZn 引脚输出副系统时钟 (f_{SUB})。

备注 n=0、1

图 11-1 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“31.4 AC 特性”。

备注 此图中的时钟输出 / 蜂鸣器输出引脚是 64 引脚产品并且 PIOR03 位和 PIOR04 位都为“0”的情况。

11.2 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 11-1 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 3、14 (PM3、PM14) 端口寄存器 3、14 (P3、P14)

11.3 控制时钟输出 / 蜂鸣器输出控制电路的寄存器

11.3.1 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 11-2 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				
					$f_{\text{MAIN}}=5\text{MHz}$	$f_{\text{MAIN}}=10\text{MHz}$	$f_{\text{MAIN}}=20\text{MHz}$	$f_{\text{MAIN}}=32\text{MHz}$
0	0	0	0	f_{MAIN}	5MHz	10MHz 注	禁止设定注	禁止设定注
0	0	0	1	$f_{\text{MAIN}}/2$	2.5MHz	5MHz	10MHz 注	16MHz 注
0	0	1	0	$f_{\text{MAIN}}/2^2$	1.25MHz	2.5MHz	5MHz	8MHz
0	0	1	1	$f_{\text{MAIN}}/2^3$	625kHz	1.25MHz	2.5MHz	4MHz
0	1	0	0	$f_{\text{MAIN}}/2^4$	312.5kHz	625kHz	1.25MHz	2MHz
0	1	0	1	$f_{\text{MAIN}}/2^{11}$	2.44kHz	4.88kHz	9.77kHz	15.63kHz
0	1	1	0	$f_{\text{MAIN}}/2^{12}$	1.22kHz	2.44kHz	4.88kHz	7.81kHz
0	1	1	1	$f_{\text{MAIN}}/2^{13}$	610Hz	1.22kHz	2.44kHz	3.91kHz
1	0	0	0	f_{SUB}	32.768kHz			
1	0	0	1	$f_{\text{SUB}}/2$	16.384kHz			
1	0	1	0	$f_{\text{SUB}}/2^2$	8.192kHz			
1	0	1	1	$f_{\text{SUB}}/2^3$	4.096kHz			
1	1	0	0	$f_{\text{SUB}}/2^4$	2.048kHz			
1	1	0	1	$f_{\text{SUB}}/2^5$	1.024kHz			
1	1	1	0	$f_{\text{SUB}}/2^6$	512Hz			
1	1	1	1	$f_{\text{SUB}}/2^7$	256Hz			

注 必须在 16MHz 以内的范围内使用输出时钟。详细内容请参照“31.4 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”; 在选择副系统时钟 (CSELn=1) 时, 因为能在副系统时钟提供模式控制寄存器 (OSMC) 的 RTCLPC 位为“0”时并且在 STOP 模式中输出时钟, 所以能将 PCLOEn 置“1”。
3. 在副系统时钟提供模式控制寄存器 (OSMC) 的 RTCLPC 位为“1”时并且在 CPU 以副系统时钟 (f_{SUB}) 运行的 HALT 模式中, 不能从 PCLBUZn 引脚输出副系统时钟 (f_{SUB})。

备注 1. n=0、1

2. f_{MAIN} : 主系统时钟频率
 f_{SUB} : 副系统时钟频率

11.3.2 控制时钟输出 / 蜂鸣器输出引脚端口功能的寄存器

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

在将时钟输出 / 蜂鸣器输出引脚的复用端口（P140/INTP6/PCLBUZ0、P141/INTP7/PCLBUZ1 等）用作时钟输出 / 蜂鸣器输出时，必须将各端口对应的端口模式寄存器（PM_{xx}）的位和端口寄存器（P_{xx}）的位置“0”。

（例）将 P140/INTP6/PCLBUZ0 用作时钟输出 / 蜂鸣器输出的情况

将端口模式寄存器 14 的 PM140 位置“0”。

将端口寄存器 14 的 P140 位置“0”。

11.4 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。

11.4.1 输出引脚的运行

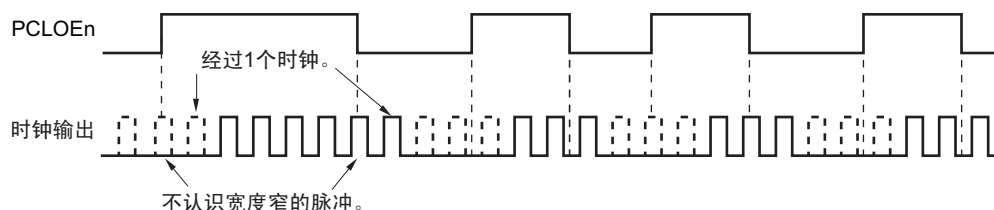
PCLBUZn 引脚按照以下步骤进行输出：

- 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 11-3 所示。

2. n=0、1

图 11-3 PCLBUZ 引脚的时钟输出时序



11.5 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 12 章 看门狗定时器

12.1 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_{LL} ）运行。

看门狗定时器用于检测程序失控。在检测到程序失控时，产生内部复位信号。

下述情况判断为程序失控。

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

当因看门狗定时器而发生复位时，将复位控制标志寄存器（RESF）的 bit4（WDTRF）置“1”。有关 RESF 寄存器的详细内容，请参照“第 21 章 复位功能”。

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

12.2 看门狗定时器的结构

看门狗定时器由以下硬件构成。

表 12-1 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)

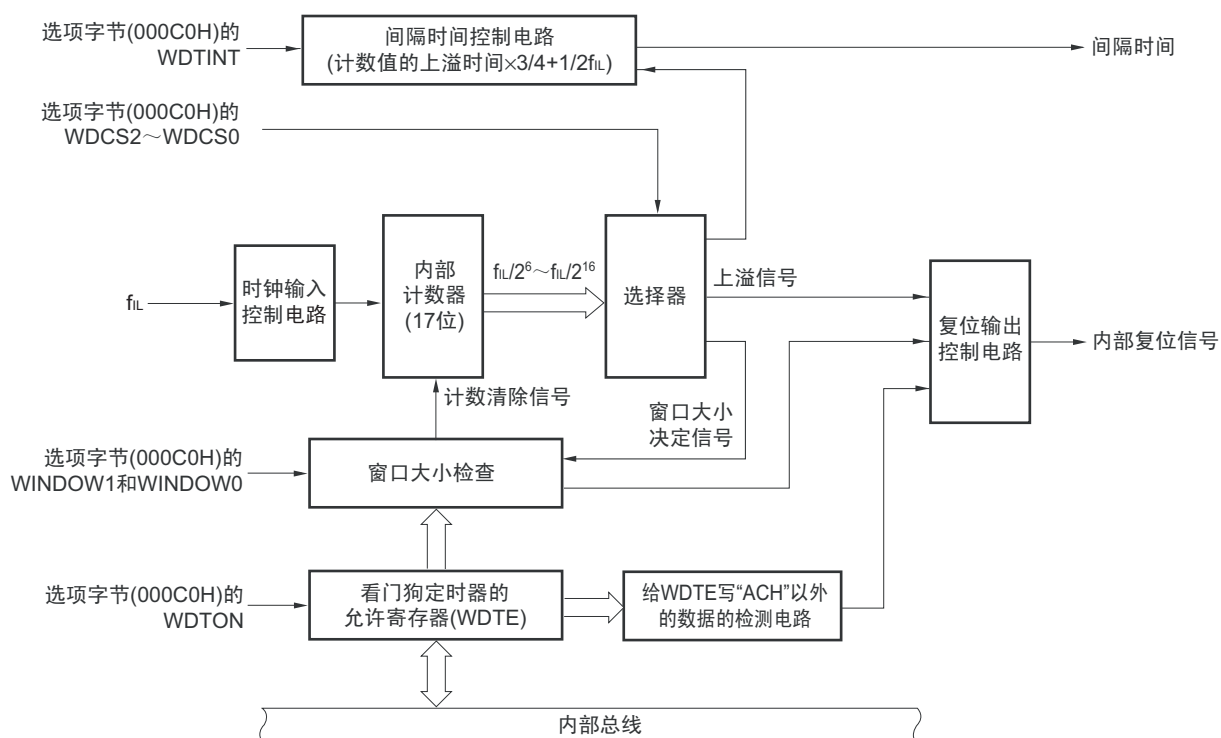
通过选项字节控制计数器的运行以及设定上溢时间、窗口打开期间和间隔中断。

表 12-2 选项字节和看门狗定时器的设定内容

看门狗定时器的设定内容	选项字节 (000C0H)
看门狗定时器的间隔中断的设定	bit7 (WDTINT)
窗口打开期间的设定	bit6 和 bit5 (WINDOW1、WINDOW0)
看门狗定时器的计数器运行控制	bit4 (WDTON)
看门狗定时器的上溢时间的设定	bit3 ~ 1 (WDSC2 ~ WDSC0)
看门狗定时器的计数器运行控制 (HALT/STOP 时)	bit0 (WDSTBYON)

备注 有关选项字节，请参照“第 26 章 选项字节”。

图 12-1 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

12.3 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。

12.3.1 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 12-2 看门狗定时器的允许寄存器（WDTE）的格式

地址: FFFABH		复位后: 9AH/1AH 注			R/W			
符号	7	6	5	4	3	2	1	0
WDTE								

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

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

- 注意 1. 当给 WDTE 寄存器写“ACH”以外的值时，产生内部复位信号。
2. 当对 WDTE 寄存器执行位存储器操作指令时，产生内部复位信号。
3. WDTE 寄存器的读取值为“9AH/1AH”（和写入值（“ACH”）不同）。

12.4 看门狗定时器的运行

12.4.1 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 26 章）。

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

- 必须通过选项字节（000C0H）的 bit3～1（WDSC2～WDSC0）设定上溢时间（详细内容请参照 12.4.2 和第 26 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 12.4.3 和第 26 章）。
- 在解除复位后，看门狗定时器开始计数。
 - 在开始计数后并且在选项字节所设上溢时间前，如果给看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除看门狗定时器并且重新开始计数。
 - 此后，解除复位后第 2 次以后的 WDTE 寄存器的写操作必须在窗口打开期间进行。如果在窗口关闭期间写 WDTE 寄存器，就产生内部复位信号。
 - 如果不给 WDTE 寄存器写“ACH”而超过上溢时间，就产生内部复位信号。
以下情况会产生内部复位信号：
 - 当对 WDTE 寄存器执行位操作指令时
 - 当给 WDTE 寄存器写“ACH”以外的数据时

- 注意 1. 只在解除复位后第 1 次写看门狗定时器的允许寄存器（WDTE）时，与窗口打开期间无关，只要在上溢时间前的任何时刻写 WDTE，就都清除看门狗定时器，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 模式或者 STOP 模式中的运行因选项字节（000C0H）的 bit0（WDSTBYON）的设定值而不同。

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

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

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

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

12.4.2 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。在上溢时间前的窗口打开期间，通过给看门狗定时器的允许寄存器（WDTE）写“ACH”，清除计数并且重新开始计数。

能设定的上溢时间如下所示。

表 12-3 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)
1	1	0	$2^{14}/f_{IL}$ (949.79ms)
1	1	1	$2^{16}/f_{IL}$ (3799.18ms)

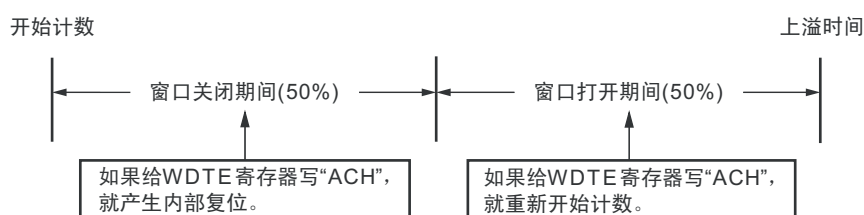
备注 f_{IL} ：低速内部振荡器的时钟频率

12.4.3 看门狗定时器窗口打开期间的设定

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

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

例 窗口打开期间为 50% 的情况



注意 只在解除复位后第 1 次写 WDTE 寄存器时，与窗口打开期间无关，只要在上溢时间前的任何时刻写 WDTE，就都清除看门狗定时器，并且重新开始计数。

能设定的窗口打开期间如下所示。

表 12-4 看门狗定时器窗口打开期间的设定

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

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

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

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

<当窗口打开期间为 50% 时>

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

12.4.4 看门狗定时器间隔中断的设定

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

表 12-5 看门狗定时器间隔中断的设定

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

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

如果从解除 STOP 模式到看门狗定时器发生上溢的时间较短，就会在振荡稳定时间内发生看门狗上溢而产生复位。

因此，在通过间隔中断解除 STOP 模式后，如果要以 X1 振荡时钟运行并且要清除看门狗定时器，因为在经过振荡稳定时间后才清除看门狗定时器，所以必须考虑这种情况进行上溢时间的设定。

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

第 13 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。

	32 引脚	64 引脚
模拟输入通道	8ch (ANI0 ~ ANI3、ANI16 ~ ANI19)	12ch (ANI0 ~ ANI7、ANI16 ~ ANI19)

13.1 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 12 个通道的 A/D 转换器模拟输入（ANI0 ~ ANI7、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。

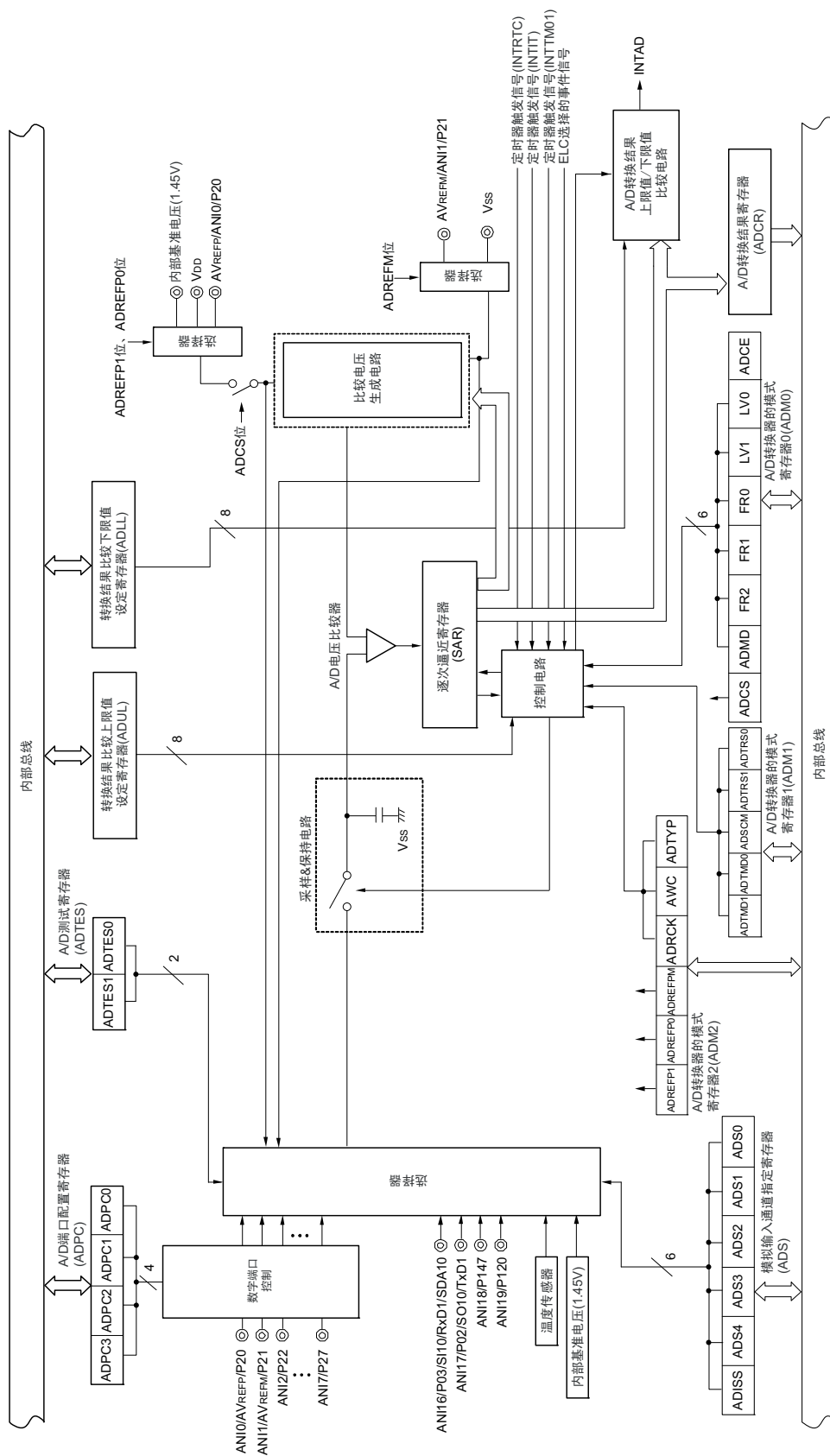
A/D 转换器有以下的功能。

- 10 位或者 8 位分辨率的 A/D 转换
从 ANI0 ~ ANI7、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	对所选 1 个通道的模拟输入进行 A/D 转换。
	扫描模式	按顺序对 4 个通道的模拟输入进行 A/D 转换。能选择 ANI0 ~ ANI14 中连续的 4 个通道作为模拟输入。
转换模式	单次转换模式	对所选通道进行 1 次 A/D 转换。
	连续转换模式	对所选通道进行连续的 A/D 转换，直到被软件停止为止。
工作电压模式	标准 1/ 标准 2 模式	用于在 $2.7V \leq V_{DD} \leq 5.5V$ 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 13-1 A/D 转换器的框图



备注 此图的模拟输入引脚是 64 引脚产品的情况。

13.2 A/D 转换器的结构

A/D 转换器由以下硬件构成。

(1) ANI0 ~ ANI7 引脚和 ANI16 ~ ANI19 引脚

这些是 A/D 转换器的 12 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 ~ ANI7、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。

13.3 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)

13.3.1 外围允许寄存器 0（PER0）

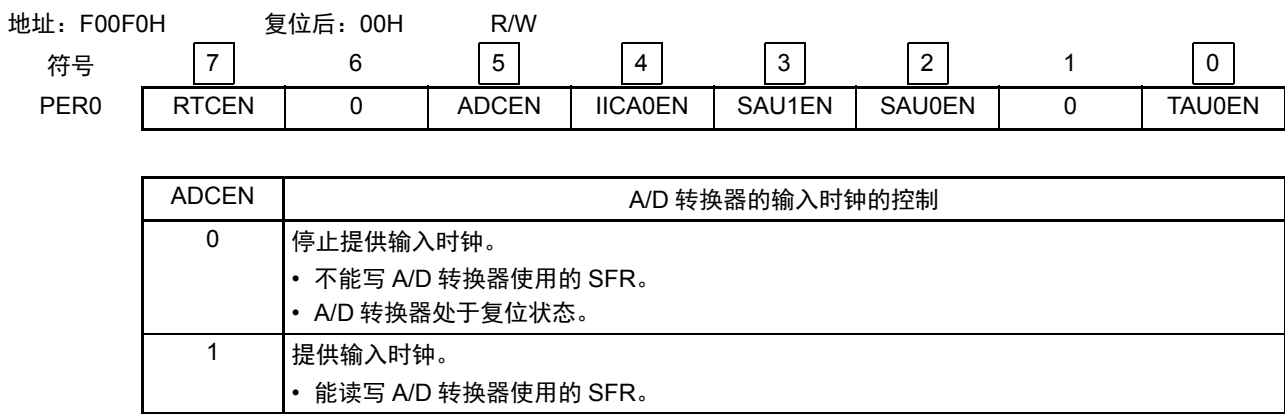
PER0 寄存器是设定允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

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 13-2 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. 必须将以下的位置“0”。
- bit1 和 bit6

13.3.2 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 13-3 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 13-3 A/D 转换时间的选择”。

- 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 1 μ s。因此，在将 ADCE 位置“1”后至少经过 1 μ s，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 1 μ s 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。

- 禁止进行 ADCS=1、ADCE=0 的设定。
- 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“13.7 A/D 转换器的设定流程图”的步骤进行设定。

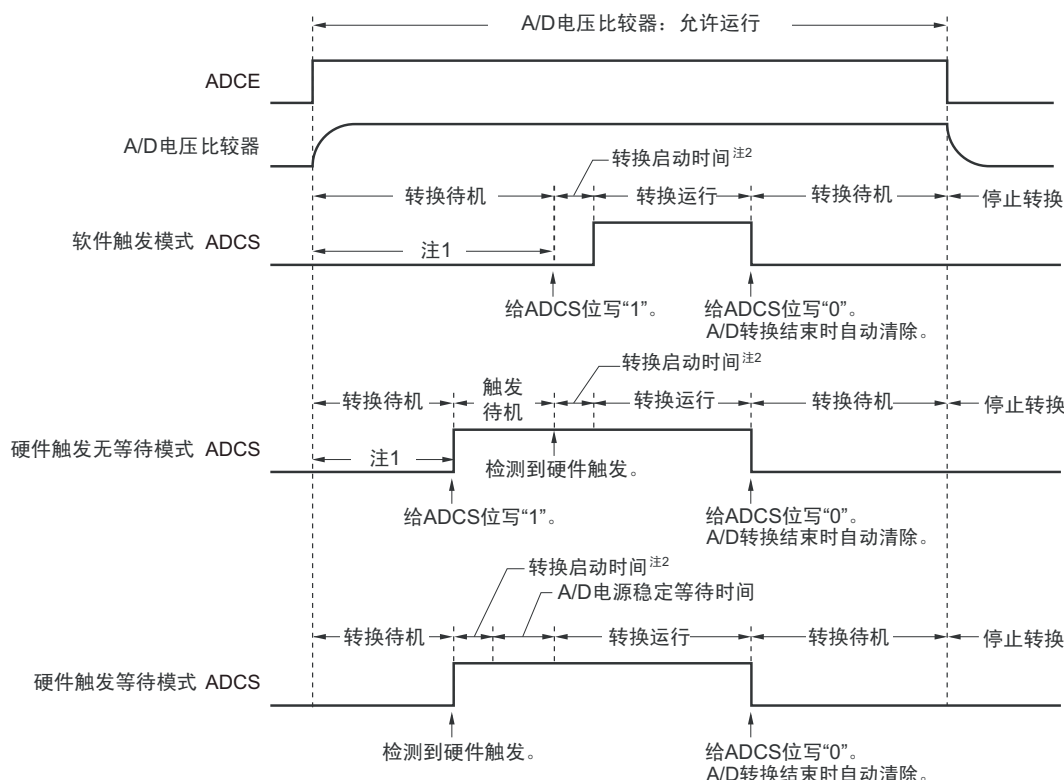
表 13-1 ADCS 位和 ADCE 位的设定

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 13-2 ADCS 位的置位和清除条件

A/D 转换模式			置位条件	清除条件
软件触发	选择模式	连续转换模式	当给 ADCS 位写“1”时	当给 ADCS 位写“0”时
		单次转换模式		- 当给 ADCS 位写“0”时 - 在 A/D 转换结束时自动清“0”。
	扫描模式	连续转换模式		当给 ADCS 位写“0”时
		单次转换模式		- 当给 ADCS 位写“0”时 - 当设定的 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		- 当给 ADCS 位写“0”时 - 在 A/D 转换结束时自动清“0”。
	扫描模式	连续转换模式		当给 ADCS 位写“0”时
		单次转换模式		- 当给 ADCS 位写“0”时 - 当设定的 4 个通道转换结束时，自动清“0”。

图 13-4 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 $1\mu\text{s}$ 。
2. 在启动转换时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	$f_{\text{CLK}}/64$	63	1
0	0	1	$f_{\text{CLK}}/32$	31	
0	1	0	$f_{\text{CLK}}/16$	15	
0	1	1	$f_{\text{CLK}}/8$	7	
1	0	0	$f_{\text{CLK}}/6$	5	
1	0	1	$f_{\text{CLK}}/5$	4	
1	1	0	$f_{\text{CLK}}/4$	3	
1	1	1	$f_{\text{CLK}}/2$	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，在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 13-3 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 ≤ V _{DD} ≤ 5.5V							
									f _{CLK} = 1MHz	f _{CLK} = 4MHz	f _{CLK} = 8MHz	f _{CLK} = 16MHz	f _{CLK} = 32MHz			
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	76μs	38μs			
0	0	1				f _{CLK} /32		608/f _{CLK}				76μs	38μs	19μs		
0	1	0				f _{CLK} /16		304/f _{CLK}				76μs	38μs	19μs	9.5μs	
0	1	1				f _{CLK} /8		152/f _{CLK}				38μs	19μs	9.5μs	4.75μs	
1	0	0				f _{CLK} /6		114/f _{CLK}				28.5μs	14.25μs	7.125μs	3.5625μs	
1	0	1				f _{CLK} /5		95/f _{CLK}				95μs	23.75μs	11.875μs	5.938μs	2.9688μs
1	1	0				f _{CLK} /4		76/f _{CLK}				76μs	19μs	9.5μs	4.75μs	2.375μs
1	1	1				f _{CLK} /2		38/f _{CLK}				38μs	9.5μs	4.75μs	2.375μs	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	68μs	34μs			
0	0	1				f _{CLK} /32		544/f _{CLK}				68μs	34μs	17μs		
0	1	0				f _{CLK} /16		272/f _{CLK}				68μs	34μs	17μs	8.5μs	
0	1	1				f _{CLK} /8		136/f _{CLK}				34μs	17μs	8.5μs	4.25μs	
1	0	0				f _{CLK} /6		102/f _{CLK}				25.5μs	12.75μs	6.375μs	3.1875μs	
1	0	1				f _{CLK} /5		85/f _{CLK}				85μs	21.25μs	10.625μs	5.3125μs	2.6563μs
1	1	0				f _{CLK} /4		68/f _{CLK}				68μs	17μs	8.5μs	4.25μs	2.125μs
1	1	1				f _{CLK} /2		34/f _{CLK}				34μs	8.5μs	4.25μs	2.125μs	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“31.6.1 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 13-3 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} = 32MHz			
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	38μs			
0	0	1				f _{CLK} /32		608/f _{CLK}				76μs	38μs	19μs		
0	1	0				f _{CLK} /16		304/f _{CLK}				76μs	38μs	19μs	9.5μs	
0	1	1				f _{CLK} /8		152/f _{CLK}				38μs	19μs	9.5μs	4.75μs	
1	0	0				f _{CLK} /6		114/f _{CLK}				28.5μs	14.25μs	7.125μs	3.5625μs	
1	0	1				f _{CLK} /5		95/f _{CLK}				95μs	23.75μs	11.875μs	5.938μs	2.9688μs
1	1	0				f _{CLK} /4		76/f _{CLK}				76μs	19μs	9.5μs	4.75μs	2.375μs
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	34μs			
0	0	1				f _{CLK} /32		544/f _{CLK}				68μs	34μs	17μs		
0	1	0				f _{CLK} /16		272/f _{CLK}				68μs	34μs	17μs	8.5μs	
0	1	1				f _{CLK} /8		136/f _{CLK}				34μs	17μs	8.5μs	4.25μs	
1	0	0				f _{CLK} /6		102/f _{CLK}				25.5μs	12.75μs	6.375μs	3.1875μs	
1	0	1				f _{CLK} /5		85/f _{CLK}				85μs	21.25μs	10.625μs	5.3125μs	2.6563μs
1	1	0				f _{CLK} /4		68/f _{CLK}				68μs	17μs	8.5μs	4.25μs	2.125μs
1	1	1				f _{CLK} /2		34/f _{CLK}				34μs	8.5μs	4.25μs	2.125μs	禁止设定

注 1. $1.8V \leq V_{DD} \leq 5.5V$ 2. $2.4V \leq V_{DD} \leq 5.5V$ 3. $2.7V \leq V_{DD} \leq 5.5V$ 4. 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“31.6.1 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 13-3 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})	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 ≤ V _{DD} ≤ 5.5V									
										f _{CLK} =1MHz	f _{CLK} =4MHz	f _{CLK} =8MHz	f _{CLK} =16MHz	f _{CLK} =32MHz					
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	108μs	54μs					
0	0	1				f _{CLK} /32			864/f _{CLK}						108μs	54μs	27μs		
0	1	0				f _{CLK} /16			432/f _{CLK}						108μs	54μs	27μs	13.5μs	
0	1	1				f _{CLK} /8			216/f _{CLK}						54μs	27μs	13.5μs	6.75μs	
1	0	0				f _{CLK} /6			162/f _{CLK}						40.5μs	20.25μs	10.125μs	5.0625μs	
1	0	1				f _{CLK} /5			135/f _{CLK}						135μs	33.75μs	16.875μs	8.4375μs	4.21875μs
1	1	0				f _{CLK} /4			108/f _{CLK}						108μs	27μs	13.5μs	6.75μs	3.375μs
1	1	1				f _{CLK} /2			54/f _{CLK}						54μs	13.5μs	6.75μs	3.375μs	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	100μs	50μs					
0	0	1				f _{CLK} /32			800/f _{CLK}						100μs	50μs	25μs		
0	1	0				f _{CLK} /16			400/f _{CLK}						100μs	50μs	25μs	12.5μs	
0	1	1				f _{CLK} /8			200/f _{CLK}						50μs	25μs	12.5μs	6.25μs	
1	0	0				f _{CLK} /6			150/f _{CLK}						37.5μs	18.75μs	9.375μs	4.6875μs	
1	0	1				f _{CLK} /5			125/f _{CLK}						125μs	31.25μs	15.625μs	7.8125μs	3.90625μs
1	1	0				f _{CLK} /4			100/f _{CLK}						100μs	25μs	12.5μs	6.25μs	3.125μs
1	1	1				f _{CLK} /2			50/f _{CLK}						50μs	12.5μs	6.25μs	3.125μs	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, 在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 13-3)。

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“31.6.1 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

表 13-3 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} = 32MHz		
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	42μs		
0	0	1				f _{CLK} /32			672/f _{CLK}				84μs	42μs	21μs	
0	1	0				f _{CLK} /16			336/f _{CLK}				84μs	42μs	21μs	10.5μs
0	1	1				f _{CLK} /8			168/f _{CLK}				42μs	21μs	10.5μs	5.25μs
1	0	0				f _{CLK} /6			126/f _{CLK}				31.25μs	15.75μs	7.875μs	3.9375μs
1	0	1				f _{CLK} /5			105/f _{CLK}		105μs	26.25μs	13.125μs	6.5625μs	3.238125 μs	
1	1	0				f _{CLK} /4			84/f _{CLK}		84μs	21μs	10.5μs	5.25μs	2.625μs	
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	38μs		
0	0	1				f _{CLK} /32			608/f _{CLK}				76μs	38μs	19μs	
0	1	0				f _{CLK} /16			304/f _{CLK}				76μs	38μs	19μs	9.5μs
0	1	1				f _{CLK} /8			152/f _{CLK}				38μs	19μs	9.5μs	4.75μs
1	0	0				f _{CLK} /6			114/f _{CLK}				28.5μs	14.25μs	7.125μs	3.5625μs
1	0	1				f _{CLK} /5			96/f _{CLK}		96μs	23.75μs	11.875μs	5.938μs	2.9688μs	
1	1	0				f _{CLK} /4			76/f _{CLK}		76μs	19μs	9.5μs	4.75μs	2.375μs	
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, 在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 13-3)。

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“31.6.1 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。

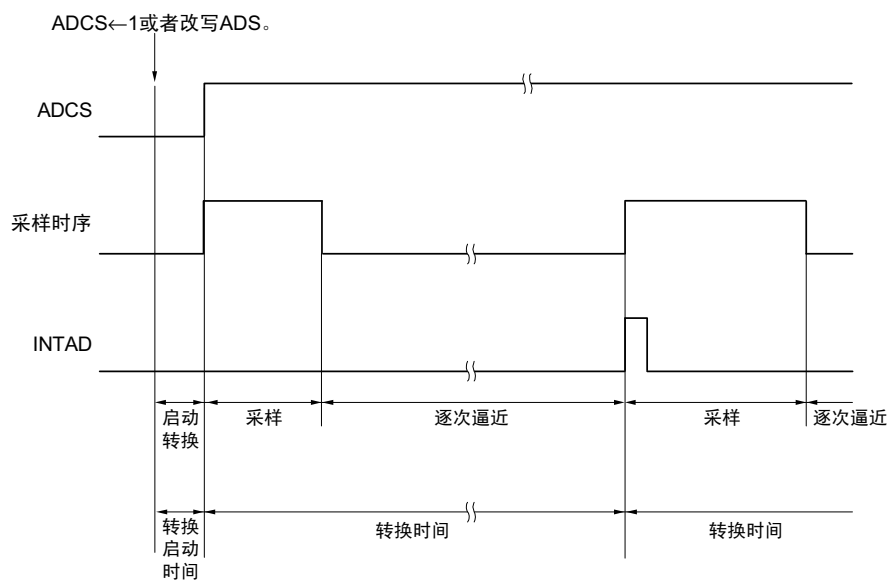
2. 要将 FR2 ~ FR0 位、LV1 位和 LV0 位改写为不同数据时, 必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。

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 13-5 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）



13.3.3 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 13-6 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	0	软件触发模式
0	1	
1	0	硬件触发无等待模式
1	1	硬件触发等待模式

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	ELC 选择的事件信号
1	0	实时时钟中断信号（INTRTC）
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，在输入 INTRTC 和 INTIT 后的最多 4 个 f_{CLK} 时钟周期内，下一个 INTRTC 和 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

13.3.4 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 13-7 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。

- 当将 ADREFP1 位和 ADREFP0 位置“1”和“0”时，不能对温度传感器的输出电压和内部基准电压进行 A/D 转换。必须在 ADISS 位为“0”时进行 A/D 转换。

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

注 只有在 HS (高速主) 模式中才能运行。

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 模式或者在 CPU 以副系统时钟运行中转移到 HALT 模式时，不能将 ADREFP1 位置“1”。在选择内部基准电压 (ADREFP1、ADREFP0=1、0) 时，加上“31.3.2 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 13-7 A/D 转换器的模式寄存器 2（ADM2）的格式 (2/2)

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

ADRCK	转换结果上限值和下限值的检查
0	当 ADLL 寄存器 ≤ ADCR 寄存器 ≤ ADUL 寄存器（AREA1）时，产生中断信号（INTAD）。
1	当 ADCR 寄存器 < ADLL 寄存器（AREA2）或者 ADUL 寄存器 < ADCR 寄存器（AREA3）时，产生中断信号（INTAD）。
AREA1 ~ AREA3 的中断信号（INTAD）的产生范围如图 13-8 所示。	

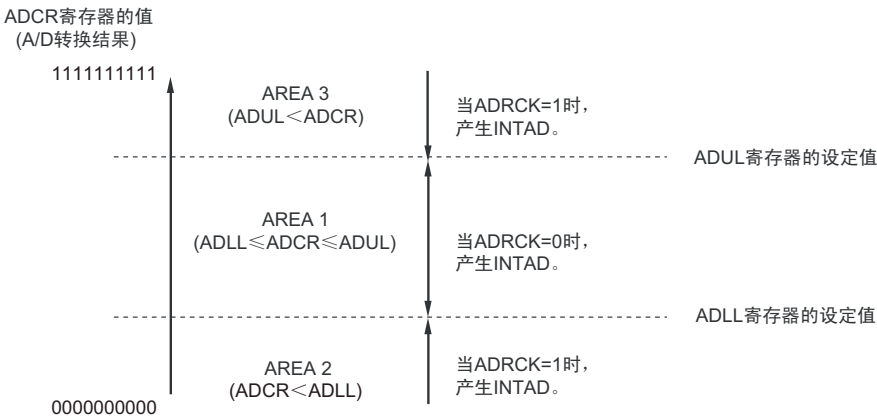
AWC	SNOOZE 模式的设定
0	不使用 SNOOZE 模式功能。
1	使用 SNOOZE 模式功能。
通过 STOP 模式中的硬件触发信号来解除 STOP 模式，并且在 CPU 不运行的情况下进行 A/D 转换（SNOOZE 模式）。 • 只有在选择高速内部振荡器时钟作为 CPU/ 外围硬件的时钟（f_{CLK}）的情况下才能设定 SNOOZE 模式功能。在选择高速内部振荡器以外的时钟的情况下，禁止设定。 • 在软件触发模式或者硬件触发无等待模式中，禁止使用 SNOOZE 模式功能。 • 在连续转换模式中，禁止使用 SNOOZE 模式功能。 • 在使用 SNOOZE 模式功能时，必须至少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 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”。 另外，必须从 STOP 模式返回到通常运行模式后将 AWC 位改为“0”。 如果 AWC 位保持“1”的状态，就无法正常开始 A/D 转换，而与其后的 SNOOZE 模式和通常运行模式无关。	

ADTYP	A/D 转换分辨率的选择
0	10 位分辨率
1	8 位分辨率

注 请参照“20.3.3 SNOOZE 模式”的“STOP 模式 → SNOOZE 模式的转移时间”。

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

图 13-8 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。

13.3.5 10 位 A/D 转换结果寄存器 (ADCR)

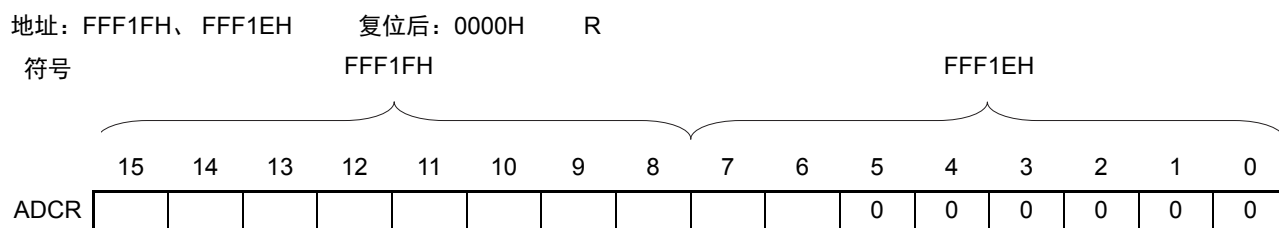
这是保存 A/D 转换结果的 16 位寄存器，低 6 位固定为“0”。每当 A/D 转换结束时，就从逐次逼近寄存器 (SAR) 装入转换结果。转换结果的高 8 位保存到 FFF1FH，低 2 位保存到 FFF1EH 的高 2 位注。

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 13-8））的设定值范围内，就不保存 A/D 转换结果。

图 13-9 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 bit15 依次读转换结果的高 10 位。

13.3.6 8 位 A/D 转换结果寄存器 (ADCRH)

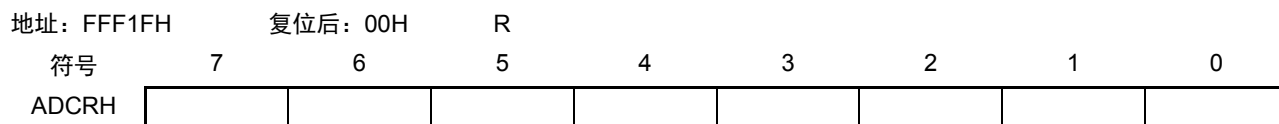
这是保存 A/D 转换结果的 8 位寄存器，保存 10 位分辨率的高 8 位注。

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 13-8））的设定值范围内，就不保存 A/D 转换结果。

图 13-10 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。

13.3.7 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 13-11 模拟输入通道指定寄存器（ADS）的格式 (1/2)

地址: FFF31H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	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	0	0	1	0	0	ANI4	P24/ANI4 引脚
0	0	0	1	0	1	ANI5	P25/ANI5 引脚
0	0	0	1	1	0	ANI6	P26/ANI6 引脚
0	0	0	1	1	1	ANI7	P27/ANI7 引脚
0	1	0	0	0	0	ANI16	P03/ANI16 引脚注 1
0	1	0	0	0	1	ANI17	P02/ANI17 引脚注 2
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	0	—	温度传感器的输出电压注 3
1	0	0	0	0	1	—	内部基准电压（1.45V）注 3
上述以外						禁止设定。	

- 注 1. 在 32 引脚产品的情况下，为 P01/ANI16 引脚。
2. 在 32 引脚产品的情况下，为 P00/ANI17 引脚。
3. 只有在 HS（高速主）模式中才能运行。

图 13-11 模拟输入通道指定寄存器 (ADS) 的格式 (2/2)

地址: FFF31H 复位后: 00H R/W

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

○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	ANI4
0	0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
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 PMCx 寄存器设定为模拟输入的端口，必须通过端口模式寄存器 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“13.7.4 选择温度传感器的输出电压 / 内部基准电压时的设定”。
- 要转移到 STOP 模式或者在 CPU 以副系统时钟运行中转移到 HALT 模式时，不能将 ADISS 位置“1”。如果将 ADISS 位置“1”，就加上“31.3.2 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。

13.3.8 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 13-8）控制中断信号（INTAD）的产生。

通过 8 位存储器操作指令设定 ADUL 寄存器。

在产生复位信号后，此寄存器的值变为“FFH”。

- 注意 1. 当选择 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。

图 13-12 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

13.3.9 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 13-8）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 13-13 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。

13.3.10 A/D 测试寄存器（ADTES）

此寄存器选择 A/D 转换器的正（+）基准电压、负（-）基准电压、模拟输入通道（ANLxx）、温度传感器的输出电压和内部基准电压（1.45V）作为 A/D 转换对象。

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 13-14 A/D 测试寄存器（ADTES）的格式

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

ADTES1	ADTES0	A/D 转换对象
0	0	ANLxx/ 温度传感器的输出电压注 / 内部基准电压（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）模式中才能选择温度传感器的输出电压和内部基准电压（1.45V）。

13.3.11 控制模拟输入引脚端口功能的寄存器

必须设定与 A/D 转换器模拟输入复用的端口功能的控制寄存器（端口模式寄存器（PMxx）、端口模式控制寄存器（PMCxx）和 A/D 端口配置寄存器（ADPC））。详细内容请参照“4.3.1 端口模式寄存器（PMxx）”、“4.3.6 端口模式控制寄存器 0、12、14（PMC0、PMC12、PMC14）”和“4.3.7 A/D 端口配置寄存器（ADPC）”。

在将 ANI0 ~ ANI7 引脚用作 A/D 转换器的模拟输入时，必须将各端口对应的端口模式寄存器（PMxx）的位置“1”，并且通过 A/D 端口配置寄存器（ADPC）设定为模拟输入。

在将 ANI16 ~ 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（PMxx）和端口模式控制寄存器（PMCxx）的位置“1”。

13.4 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）。
- 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”。

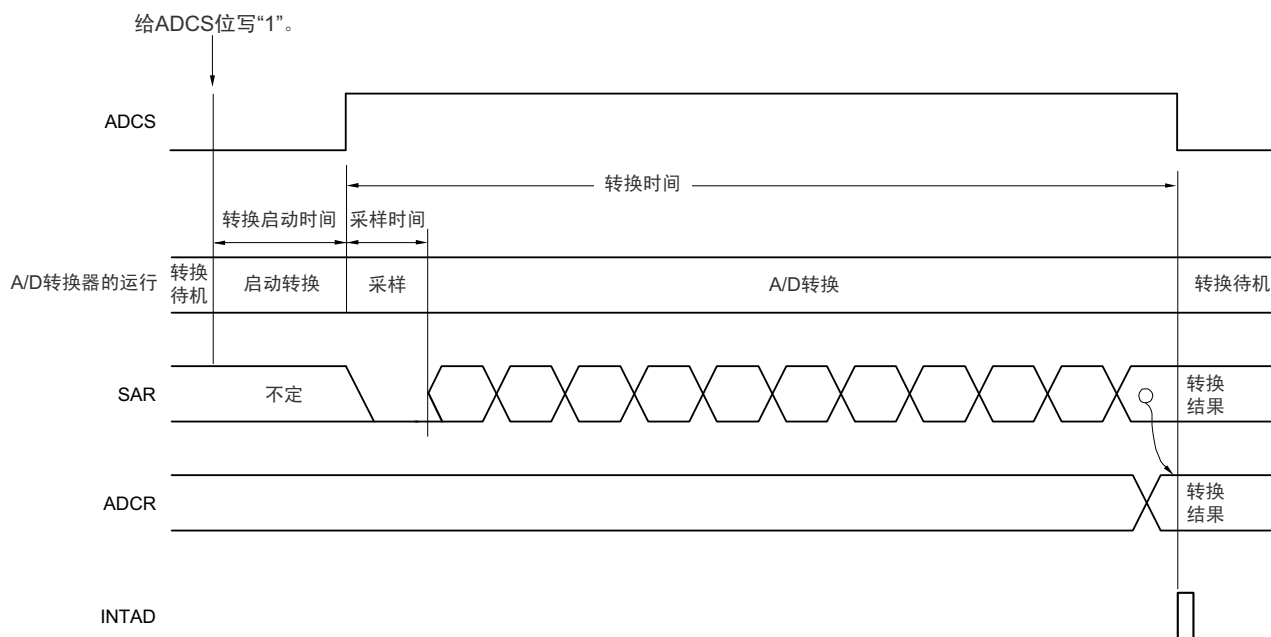
- 注 1. 如果 A/D 转换结果的值不在 A/D 转换结果比较功能（通过 ADRCK 位和 ADUL/ADLL 寄存器进行设定（参照图 13-8））的设定值范围内，就不产生 A/D 转换结束中断请求信号（INTAD），并且 ADCR 寄存器和 ADCRH 寄存器不保存 A/D 转换结果。
2. 在连续转换模式中，ADCS 标志不自动清“0”。即使在硬件触发无等待模式的单次转换模式中，ADCS 标志也不自动清“0”而保持“1”的状态。

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 13-15 A/D 转换器的转换运行（软件触发模式的情况）



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

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

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

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

13.5 输入电压和转换结果

模拟输入引脚（ANI0 ~ ANI7、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。

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$
$$ADCR = SAR \times 64$$

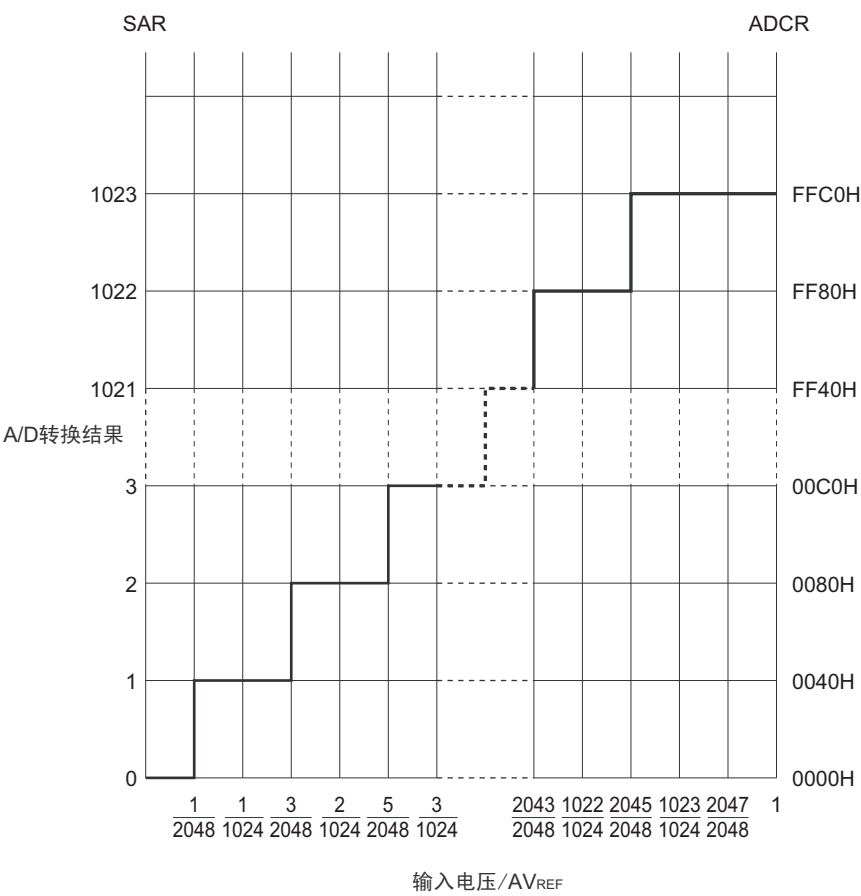
或者

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

- 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
- AV_{REF} : AV_{REF} 引脚电压
- ADCR : A/D 转换结果寄存器（ADCR）的值
- SAR : 逐次逼近寄存器

模拟输入电压和 A/D 转换结果的关系如图 13-16 所示。

图 13-16 模拟输入电压和 A/D 转换结果的关系



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

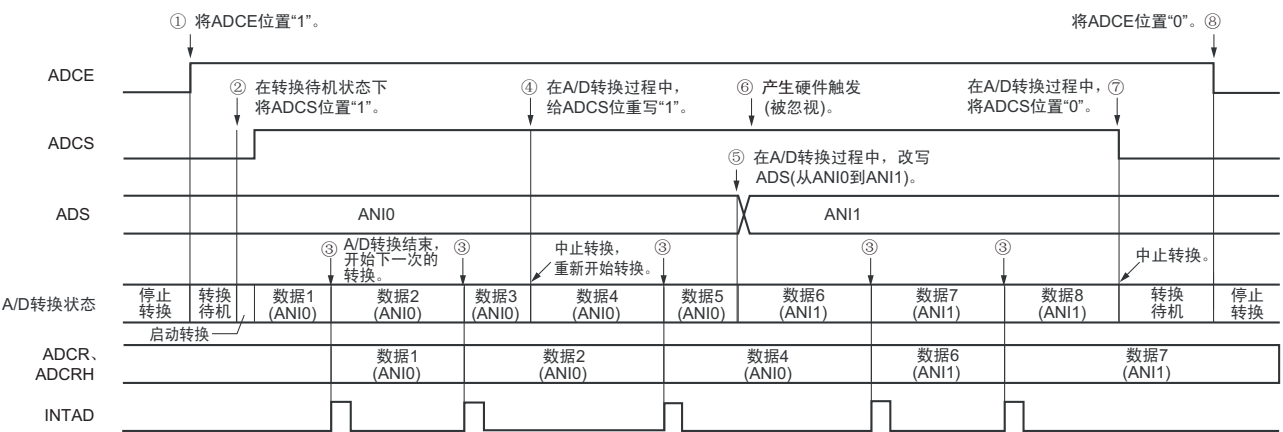
13.6 A/D 转换器的运行模式

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“13.7 A/D 转换器的设定流程图”。

13.6.1 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（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的 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。

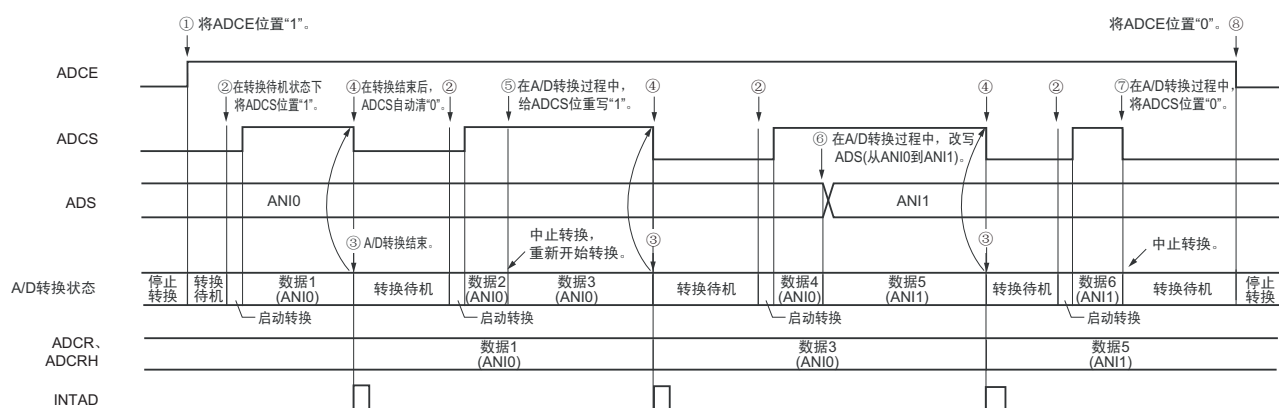
图 13-17 软件触发模式（选择模式、连续转换模式）的运行时序例子



13.6.2 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。

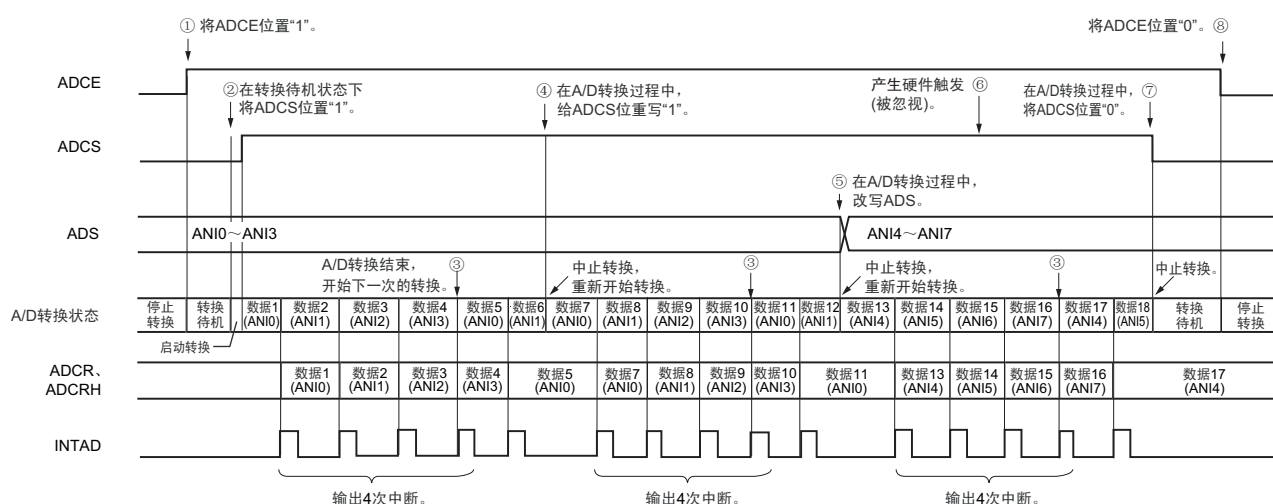
图 13-18 软件触发模式（选择模式、单次转换模式）的运行时序例子



13.6.3 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（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）。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轮的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。

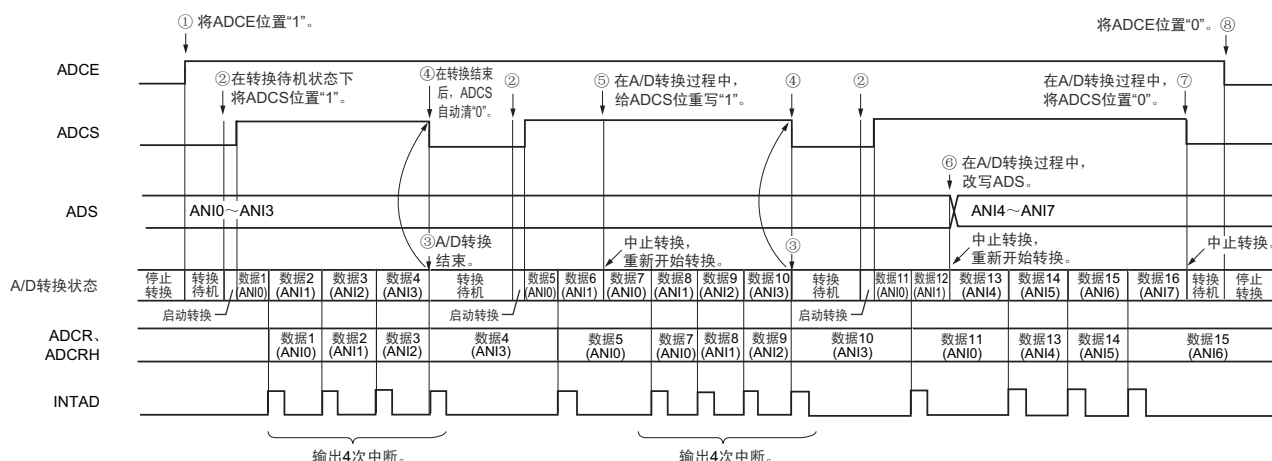
图 13-19 软件触发模式（扫描模式、连续转换模式）的运行时序例子



13.6.4 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。

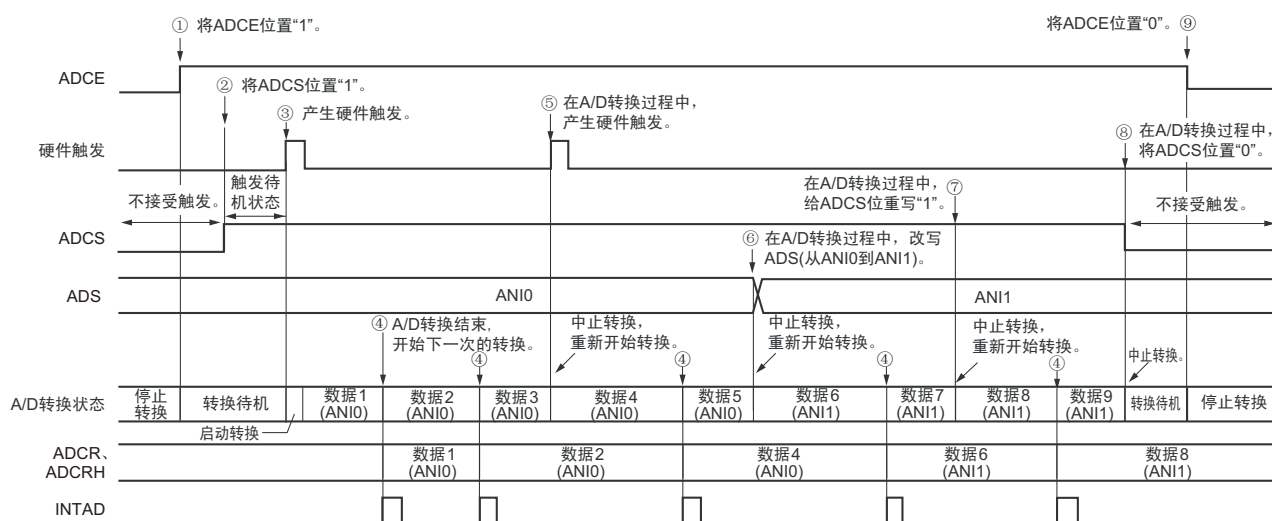
图 13-20 软件触发模式（扫描模式、单次转换模式）的运行时序例子



13.6.5 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轮的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。

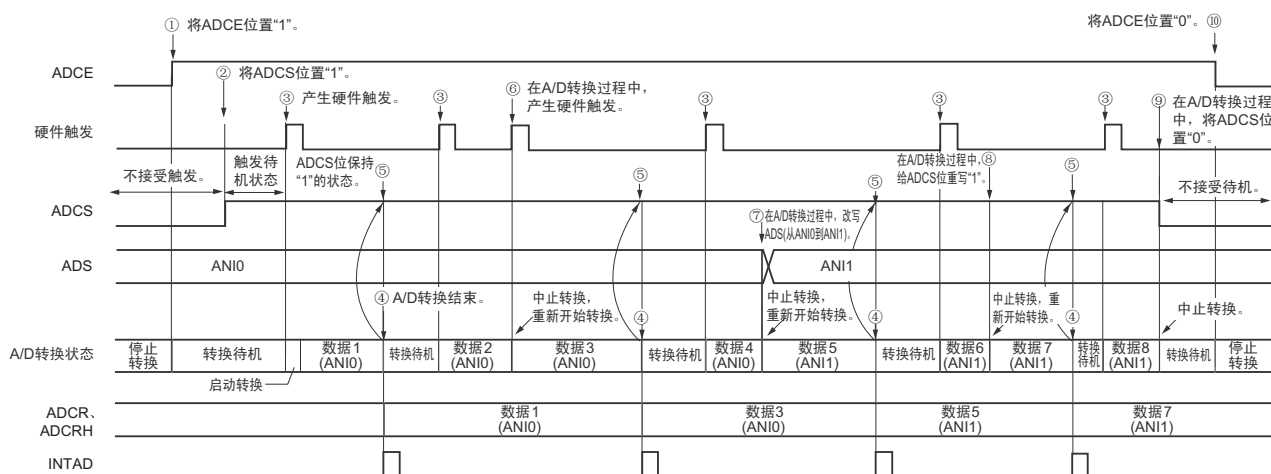
图 13-21 硬件触发无等待模式（选择模式、连续转换模式）的运行时序例子



13.6.6 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（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。

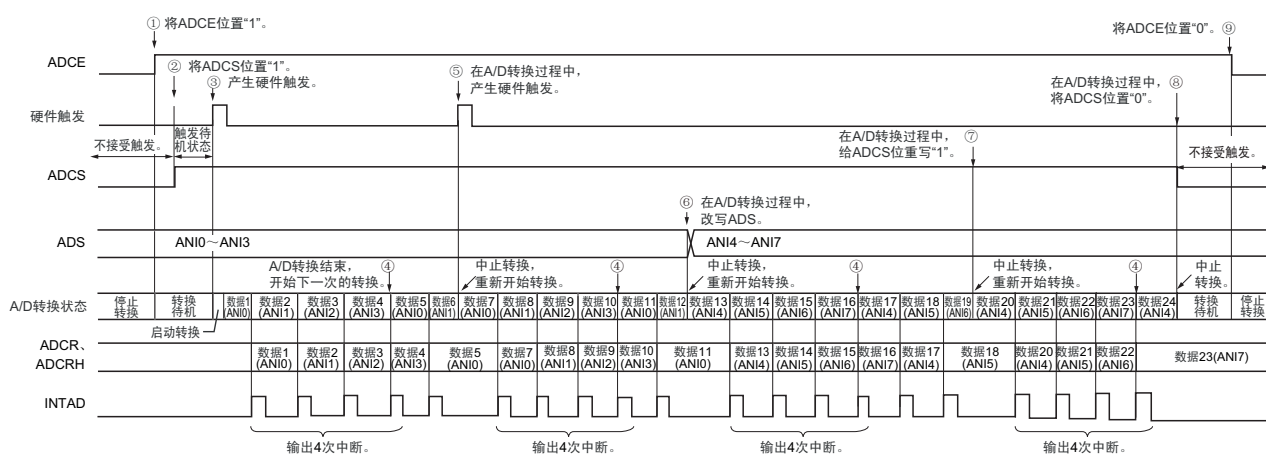
图 13-22 硬件触发无等待模式（选择模式、单次转换模式）的运行时序例子



13.6.7 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轮的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。

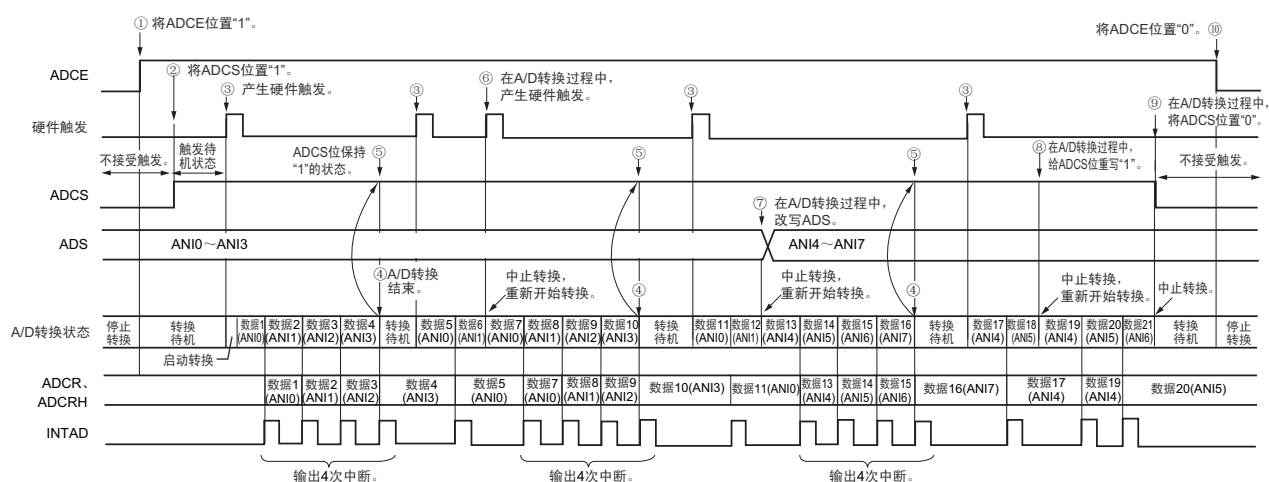
图 13-23 硬件触发无等待模式（扫描模式、连续转换模式）的运行时序例子



13.6.8 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（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。

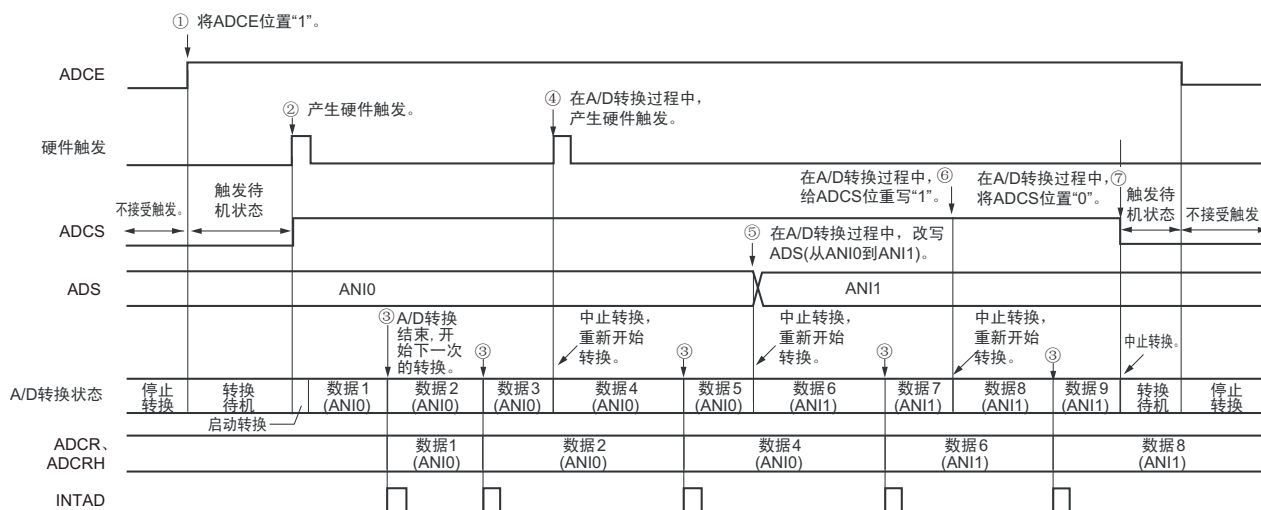
图 13-24 硬件触发无等待模式（扫描模式、单次转换模式）的运行时序例子



13.6.9 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轮的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。

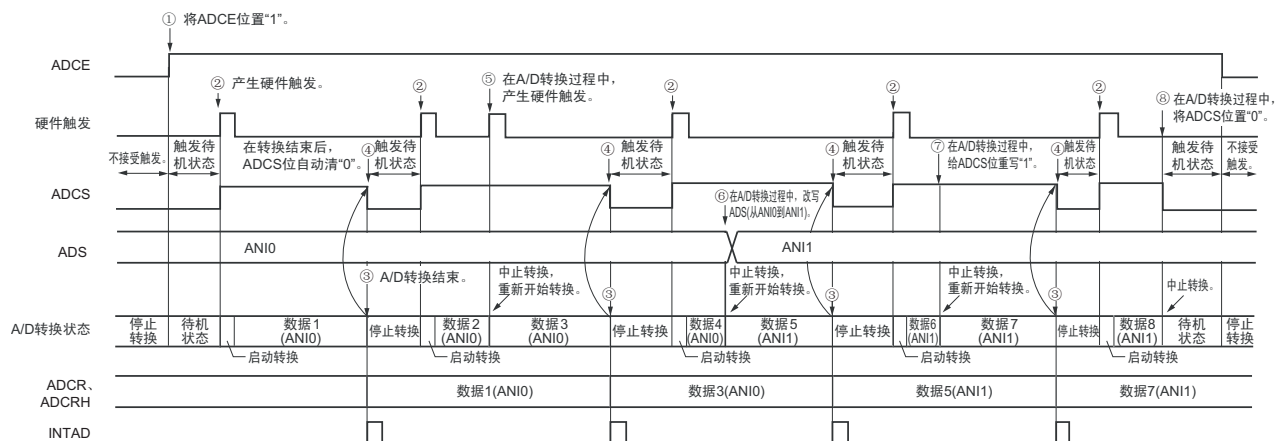
图 13-25 硬件触发等待模式（选择模式、连续转换模式）的运行时序例子



13.6.10 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。

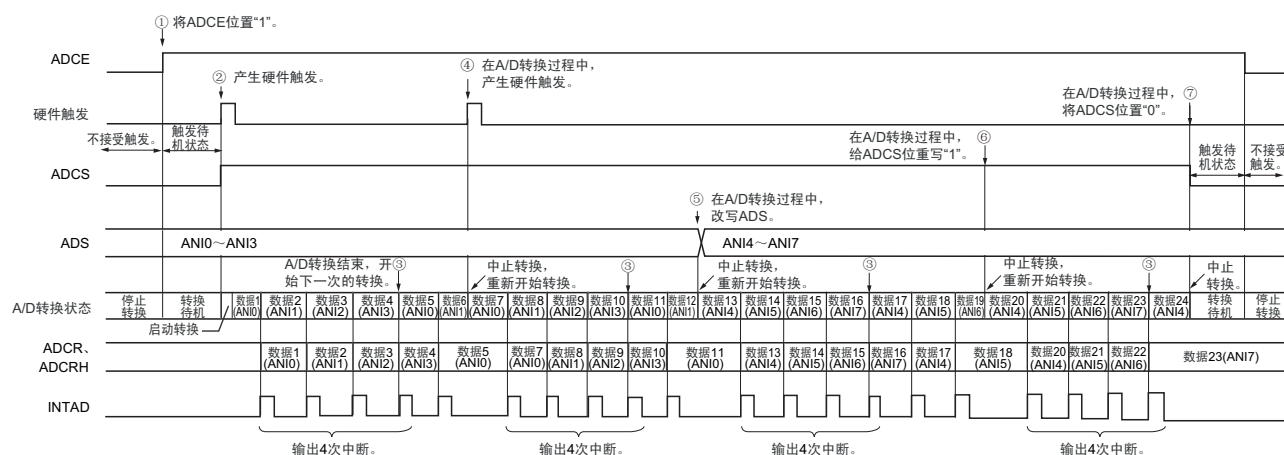
图 13-26 硬件触发等待模式（选择模式、单次转换模式）的运行时序例子



13.6.11 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轮的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。

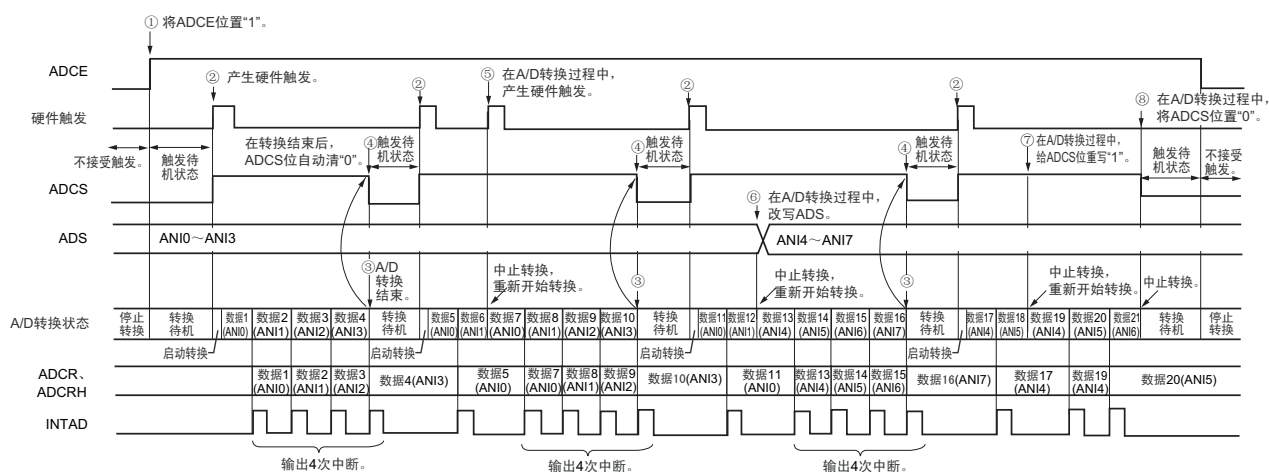
图 13-27 硬件触发等待模式（扫描模式、连续转换模式）的运行时序例子



13.6.12 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 13-28 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

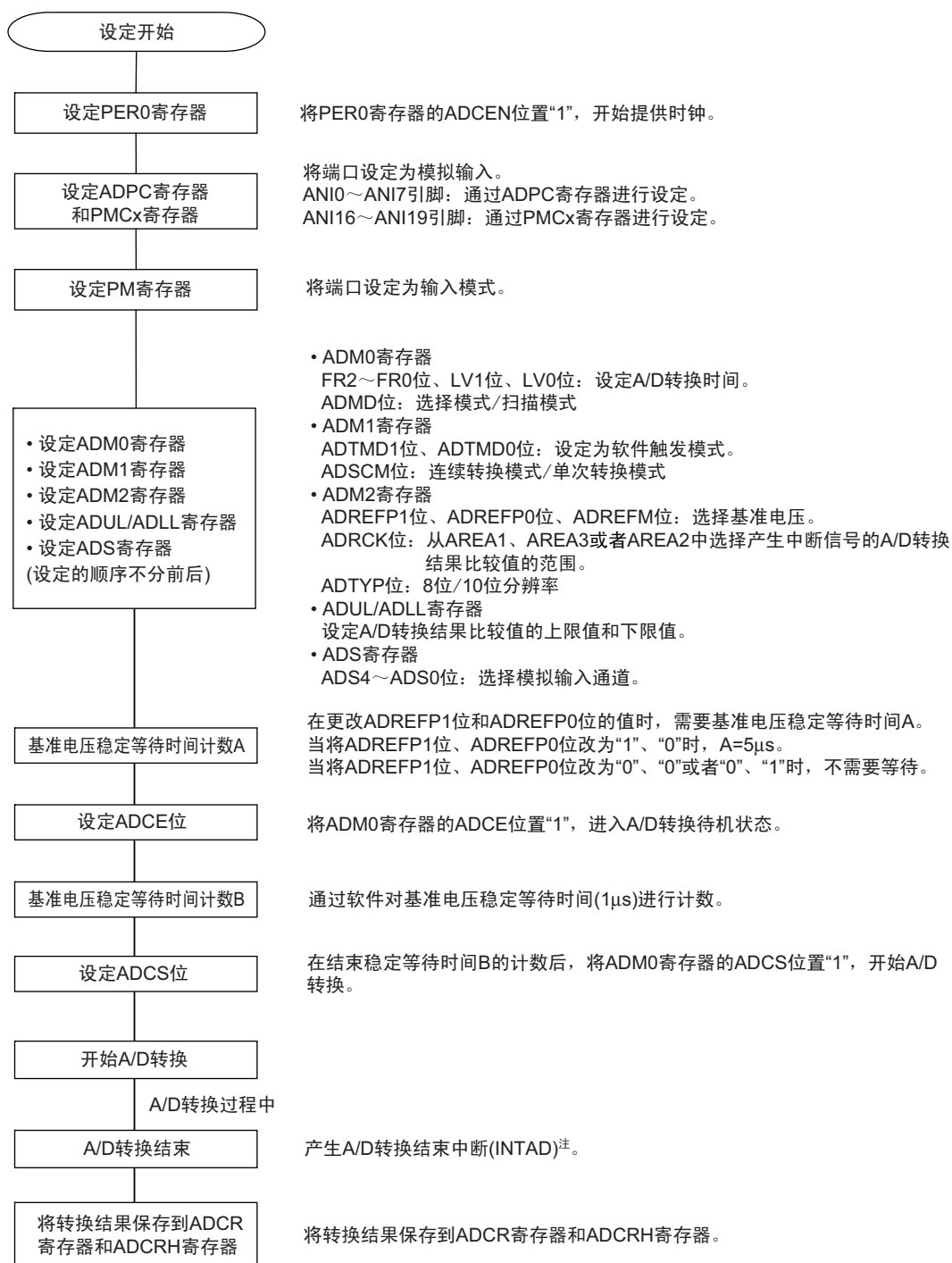


13.7 A/D 转换器的设定流程图

各运行模式的 A/D 转换器的设定流程图如下所示。

13.7.1 软件触发模式的设定

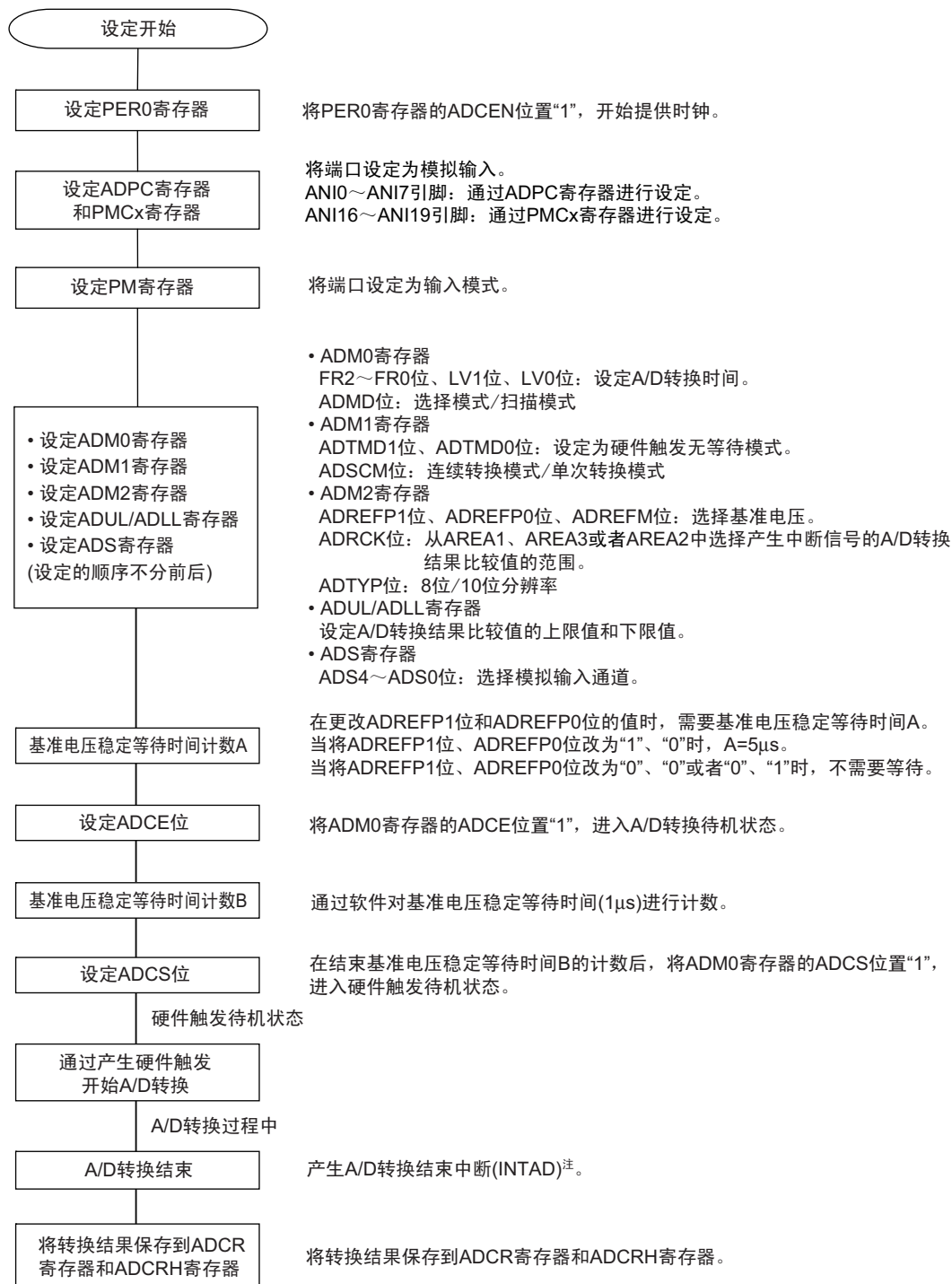
图 13-29 软件触发模式的设定



注 根据 ADRCCK 位和 ADUL/ADLL 寄存器的设定，有可能不产生中断信号。此时，不将结果保存到 ADCR 寄存器和 ADCRH 寄存器。

13.7.2 硬件触发无等待模式的设定

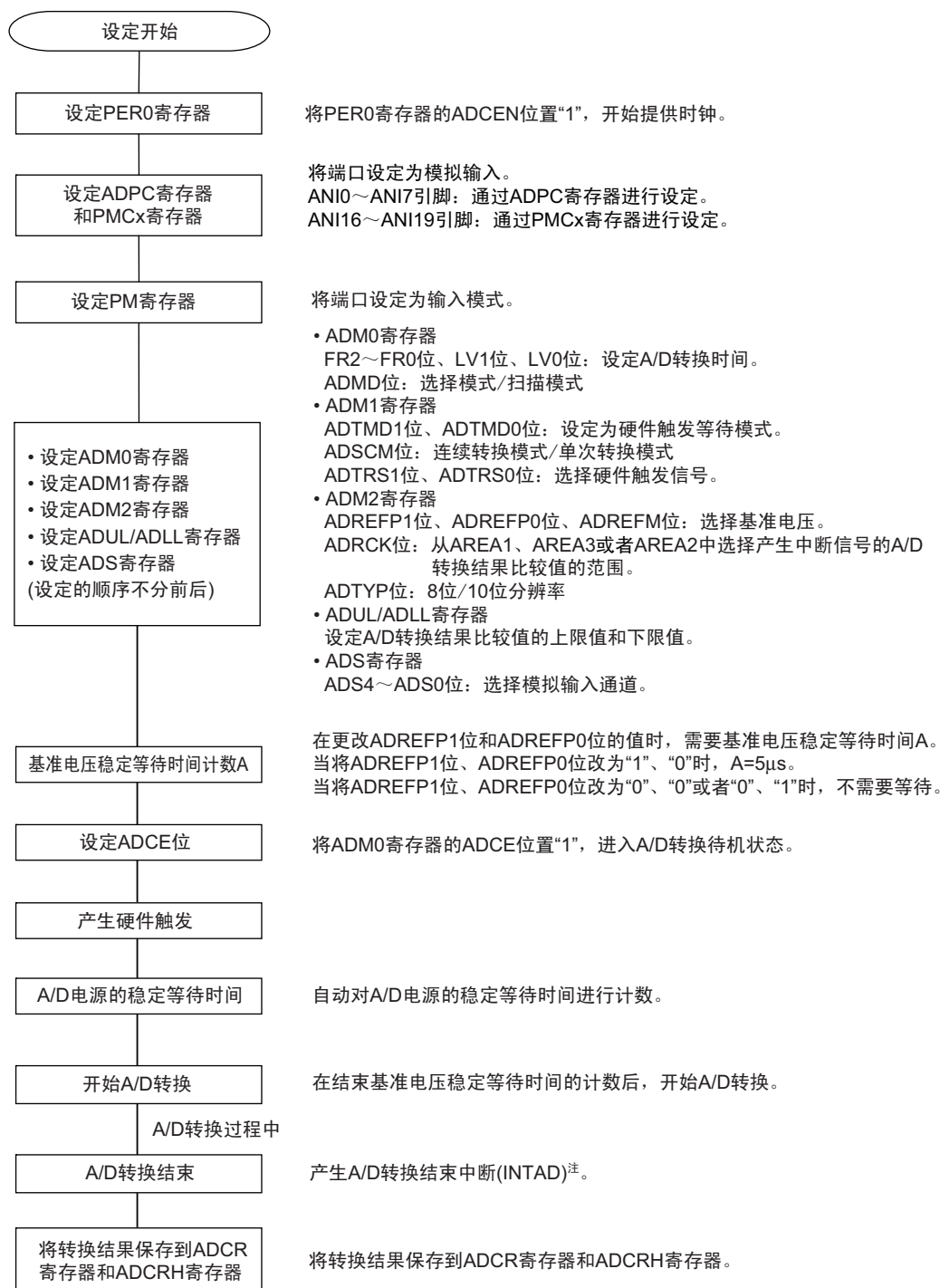
图 13-30 硬件触发无等待模式的设定



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。

13.7.3 硬件触发等待模式的设定

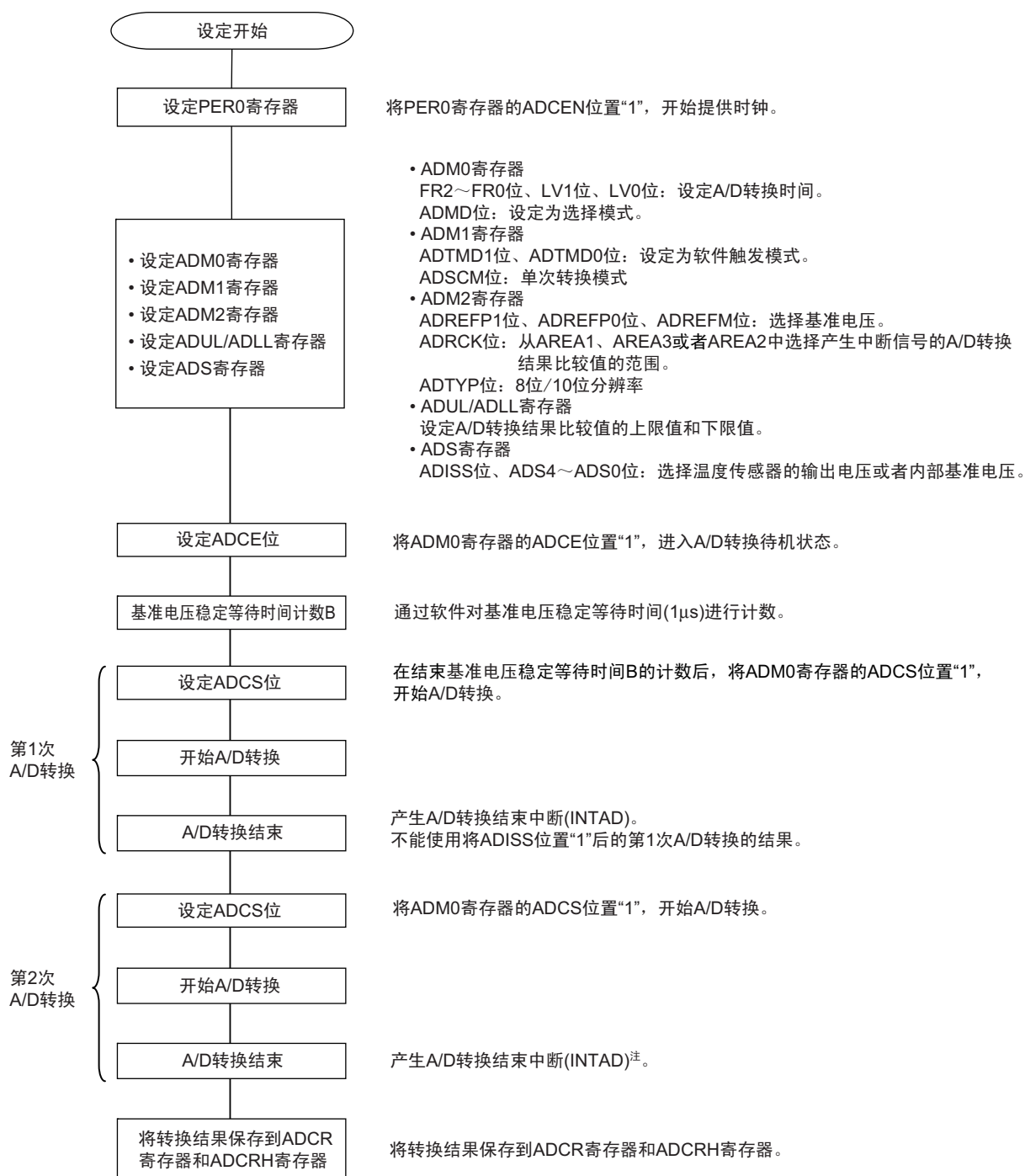
图 13-31 硬件触发等待模式的设定



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。

13.7.4 选择温度传感器的输出电压 / 内部基准电压时的设定 (以软件触发模式、单次转换模式为例)

图 13-32 选择温度传感器的输出电压 / 内部基准电压时的设定

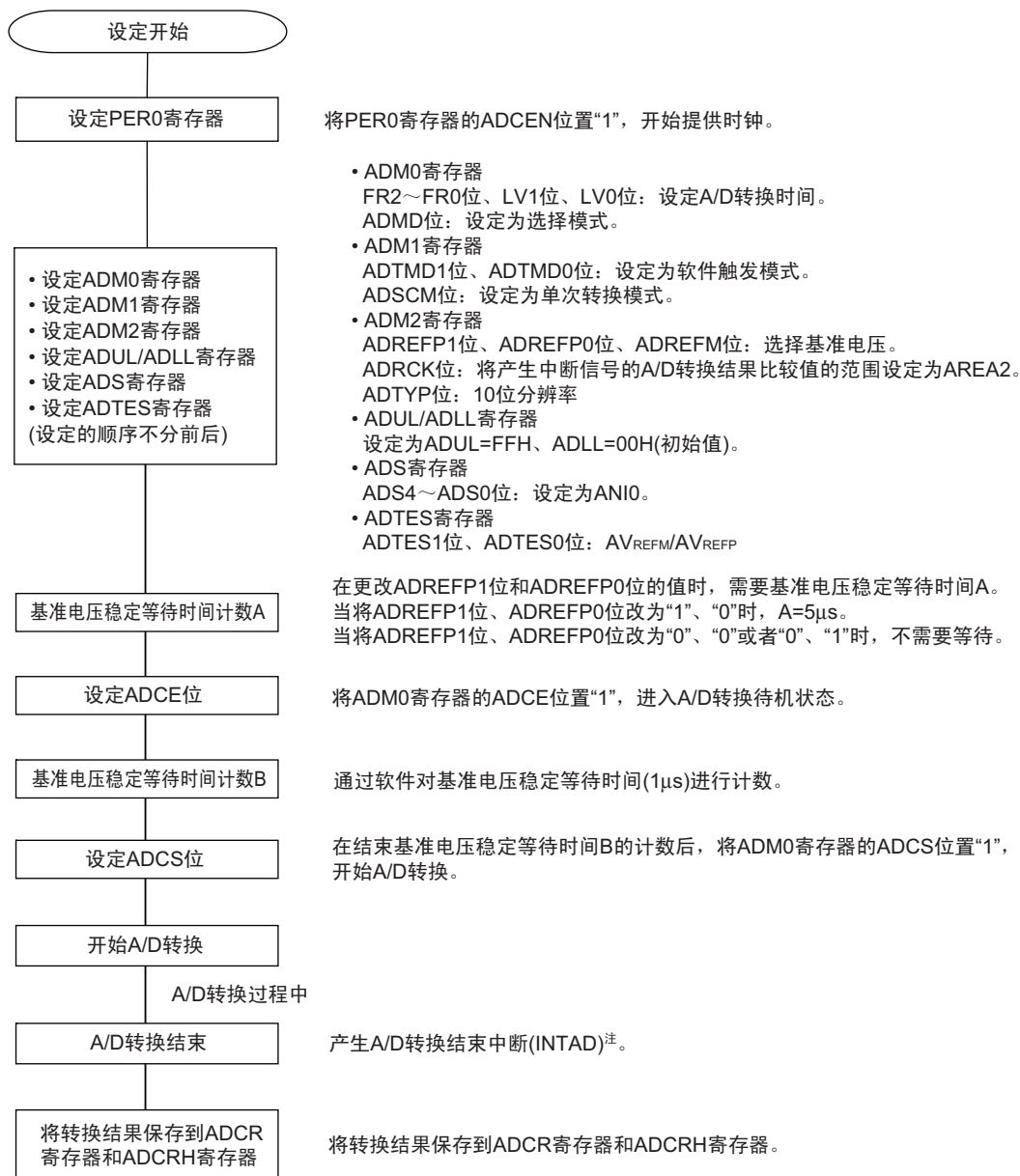


注 根据 ADRCK 位和 ADUL/ADLL 寄存器的设定，有可能不产生中断信号。此时，不将结果保存到 ADCR 寄存器和 ADCRH 寄存器。

注意 只有在 HS（高速主）模式中才能选择。

13.7.5 测试模式的设定

图 13-33 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“24.3.8 A/D 测试功能”。

13.8 SNOOZE 模式功能

这是在 STOP 模式中通过输入硬件触发进行 A/D 转换的模式。在通常的 STOP 模式中停止 A/D 转换，但是如果使用此模式，就能在 CPU 不运行的状态下通过输入硬件触发进行 A/D 转换，减小工作电流。

在 SNOOZE 模式中，如果通过 ADUL 寄存器和 ADLL 寄存器指定转换结果的范围，就能每隔一段时间判断 A/D 转换结果。从而能判断电源电压监视或者通过 A/D 输入进行的键输入。

在 SNOOZE 模式中，只能使用以下 2 种转换模式：

- 硬件触发等待模式（选择模式和单次转换模式）
- 硬件触发等待模式（扫描模式和单次转换模式）

注意 只有在选择高速内部振荡器时钟作为 f_{CLK} 时才能设定 SNOOZE 模式。

图 13-34 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“13.7.3 硬件触发等待模式的设定”注2）。此时，将 A/D 转换器的模式寄存器 2（ADM2）的 bit2（AWC）置“1”。在初始设定结束后，将 A/D 转换器的模式寄存器 0（ADM0）的 bit0（ADCE）置“1”。

如果在转移到 STOP 模式后输入硬件触发，就给 A/D 转换器提供高速内部振荡器时钟。在提供高速内部振荡器时钟后，自动对 A/D 电源稳定等待时间进行计数，开始 A/D 转换。

A/D 转换结束后的 SNOOZE 模式的运行因是否产生中断信号而不同注1。

- 注**
1. 根据 A/D 转换结果比较功能的设定（ADRCK 位、ADUL/ADLL 寄存器），有可能不产生中断请求信号。
 2. 必须将 ADM1 寄存器置“E1H”、“E2H”或者“E3H”。

备注 硬件触发为 ELC 选择的事件、INTRTC 或者 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）。

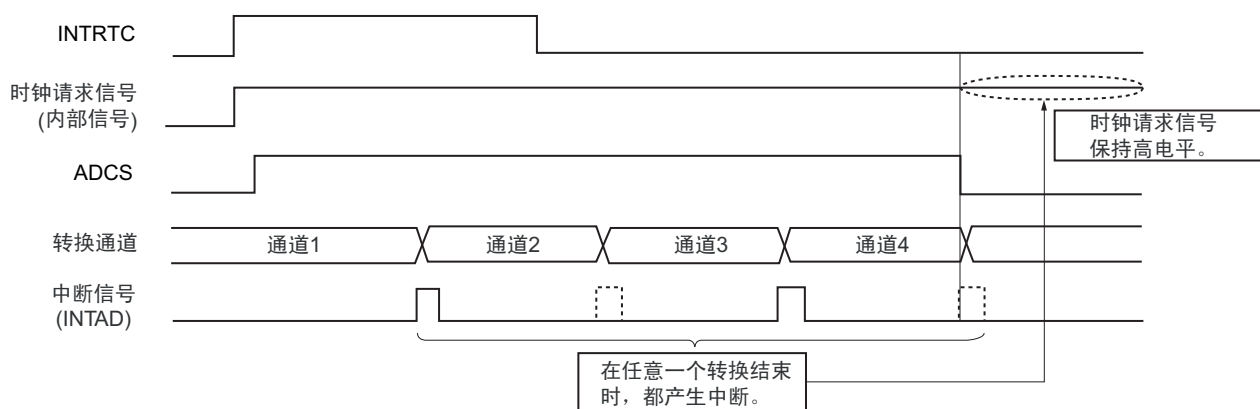
- 选择模式

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 13-35 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 13-36 在 A/D 转换结束后不发生中断时的运行例子（扫描模式）

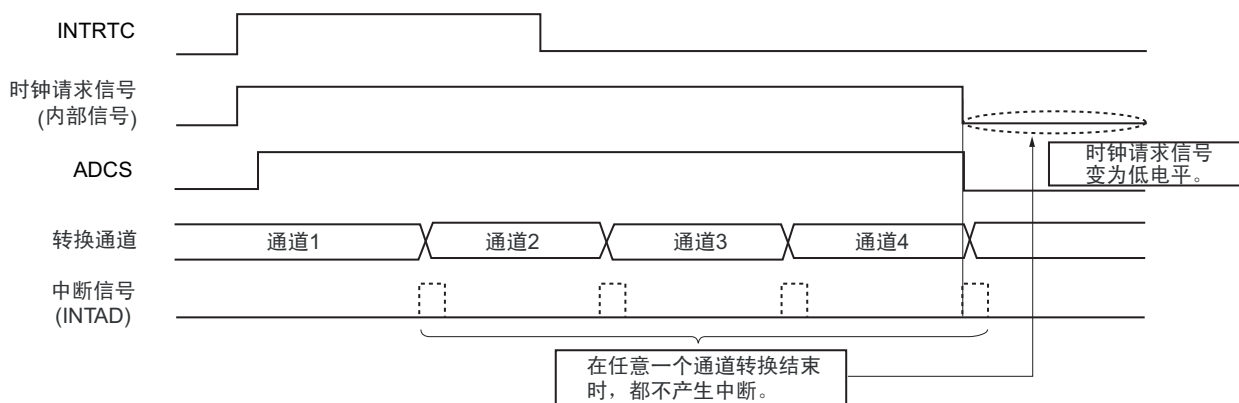
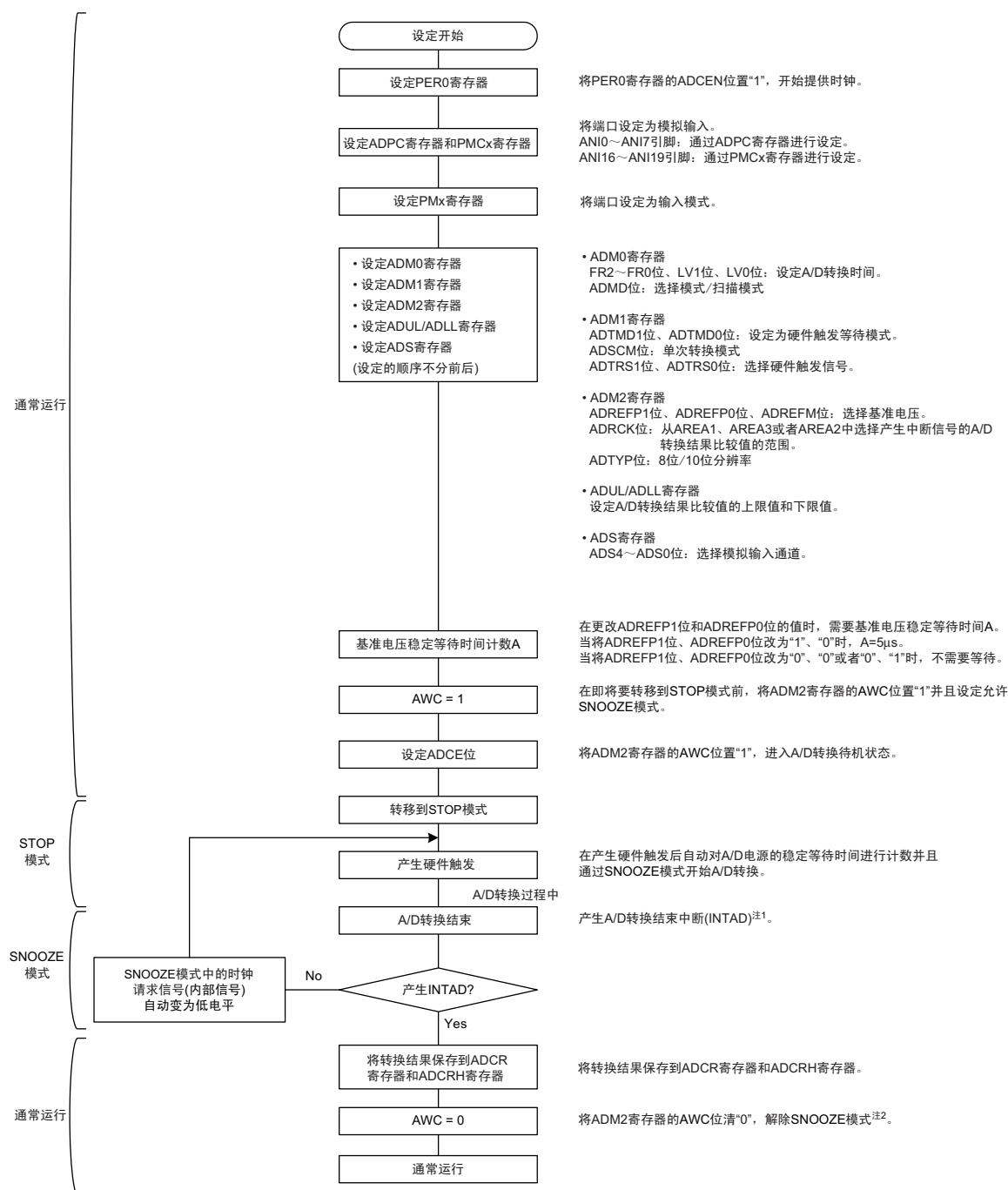


图 13-37 SNOOZE 模式的设定流程



- 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 SNOOZE 模式和通常运行模式无关。必须将 AWC 位置 “0”。

13.9 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 13-38 综合误差

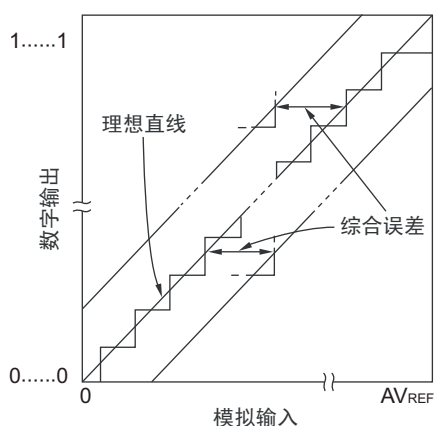
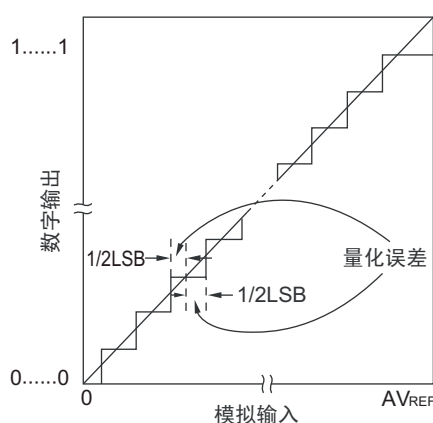


图 13-39 量化误差



(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 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 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 13-40 零刻度误差

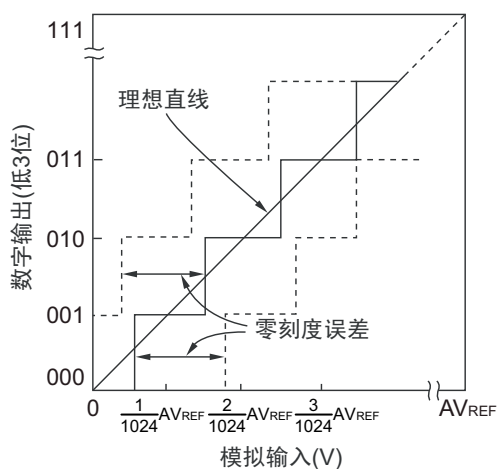


图 13-41 满刻度误差

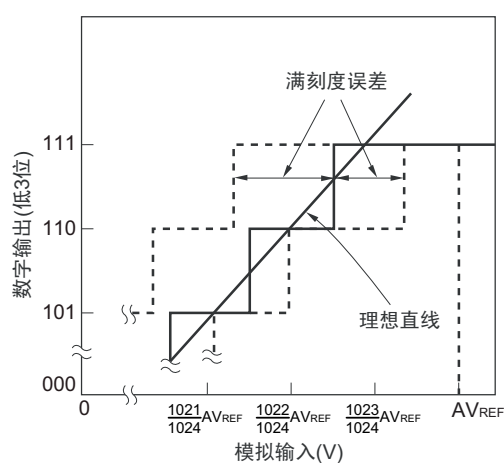


图 13-42 积分线性误差

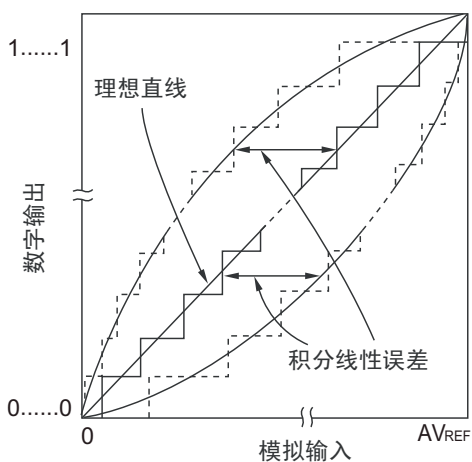
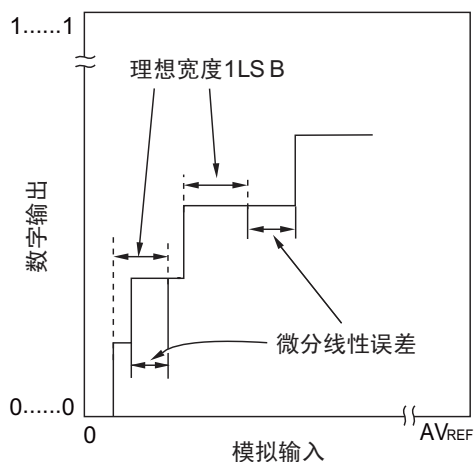


图 13-43 微分线性误差



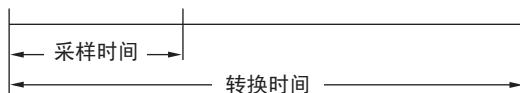
(8) 转换时间

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。



13.10 A/D 转换器的注意事项

(1) 有关 STOP 模式中的工作电流

要转移到 STOP 模式时，必须在停止 A/D 转换器（将 A/D 转换器的模式寄存器 0 (ADM0) 的 bit7 (ADCS) 置“0”) 后进行。此时，还能通过将 ADM0 寄存器的 bit0 (ADCE) 置“0”来减小工作电流。

要从待机状态重新运行时，必须在将中断请求标志寄存器 1H (IF1H) 的 bit0 (ADIF) 清“0”后开始运行。

(2) 有关 ANI0 ~ ANI7 引脚和 ANI16 ~ ANI19 引脚的输入范围

ANI0 ~ ANI7 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的电压。但是，能给 ADS 寄存器没有选择的引脚输入大于内部基准电压的电压。

注意 只有在 HS（高速主）模式中才能选择内部基准电压（1.45V）。

(3) 有关竞争

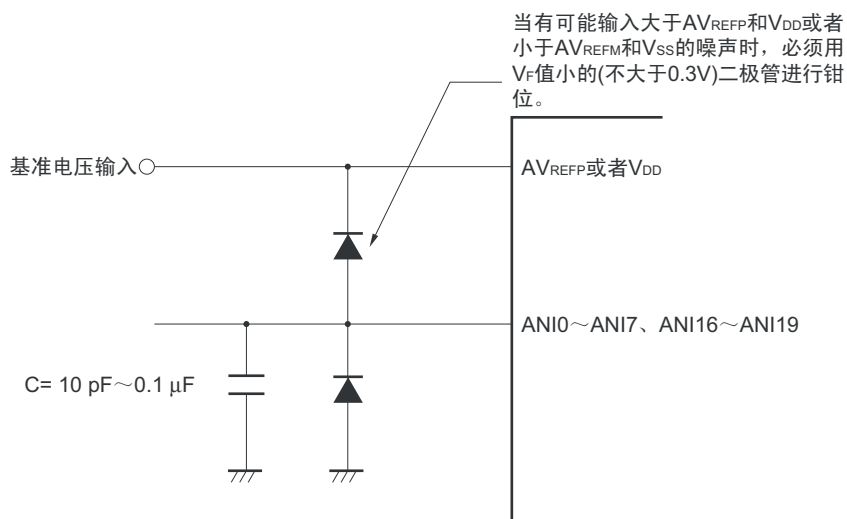
- ① 转换结束时的 A/D 转换结果寄存器 (ADCR、ADCRH) 的写操作与 ADCR 寄存器和 ADCRH 寄存器的指令读操作的竞争
优先读 ADCR 寄存器和 ADCRH 寄存器。在读后，将新的转换结果写到 ADCR 寄存器和 ADCRH 寄存器。
- ② 转换结束时的 ADCR 寄存器和 ADCRH 寄存器的写操作与 A/D 转换器的模式寄存器 0 (ADM0)、模拟输入通道指定寄存器 (ADS) 或者 A/D 端口配置寄存器 (ADPC) 的写操作的竞争
优先写 ADM0、ADS、ADPC 寄存器。不写 ADCR 寄存器和 ADCRH 寄存器，也不产生转换结束中断信号 (INTAD)。

(4) 有关噪声对策

为了保持 10 位分辨率，必须注意输入到 AV_{REFP} 、 V_{DD} 、ANI0 ~ ANI7、ANI16 ~ ANI19 引脚的噪声。

- ① 连接电源的电容器必须使用等效电阻小并且频率响应好的电容器。
- ② 模拟输入源的输出阻抗越高影响就越大，因此为了降低噪声，建议按照图 13-44 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 13-44 模拟输入引脚的处理



(5) 模拟输入 (ANIn) 引脚

- ① 模拟输入 (ANI0~ANI7、ANI16~ANI19) 引脚和输入端口 (P20~P27、P03、P02、P147、P120) 引脚复用。
要选择 ANI0~ANI7 引脚和 ANI16~ANI19 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~P27、P03、P02、P147、P120 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，不能输入或者输出这样的脉冲。

(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~ANI7 引脚和 ANI16~ANI19 引脚连接 $0.1\text{ }\mu\text{F}$ 左右的电容器（参照图 13-44）。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。

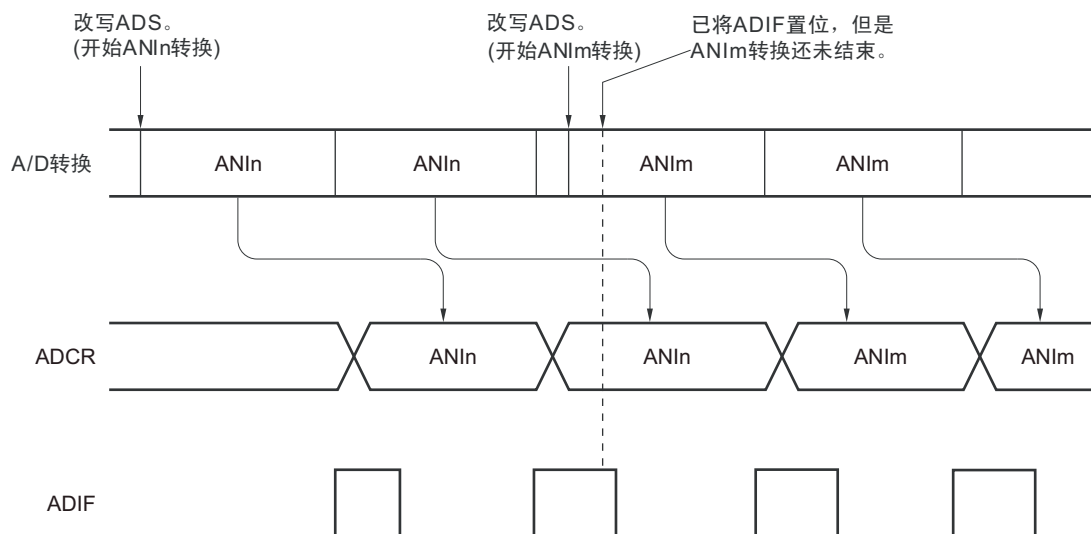
(7) 有关中断请求标志 (ADIF)

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 13-45 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 μ s 内将 ADCS 位置“1”，开始 A/D 转换后的初次 A/D 转换值就可能不满足额定值。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、PMC 寄存器前读转换结果。否则，就可能读不到正确的转换结果。

(10) 有关内部等效电路

模拟输入部的等效电路如下所示。

图 13-46 ANIn 引脚的内部等效电路

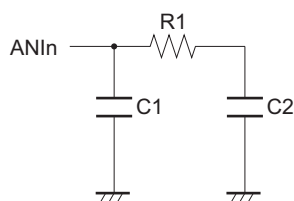


表 13-4 等效电路的各电阻和电容值（参考值）

AV_{REFP} 、 V_{DD}	ANIn 引脚	$R1[k\Omega]$	$C1[pF]$	$C2[pF]$
$3.6V \leq V_{DD} \leq 5.5V$	ANI0 ~ ANI7	14	8	2.5
	ANI16 ~ ANI19	18	8	7.0
$2.7V \leq V_{DD} < 3.6V$	ANI0 ~ ANI7	39	8	2.5
	ANI16 ~ ANI19	53	8	7.0
$1.8V \leq V_{DD} < 2.7V$	ANI0 ~ ANI7	231	8	2.5
	ANI16 ~ ANI19	321	8	7.0
$1.6V \leq V_{DD} < 2.7V$	ANI0 ~ ANI7	632	8	2.5
	ANI16 ~ ANI19	902	8	7.0

备注 表 13-4 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 14 章 串行阵列单元

串行阵列单元 0 有 4 个串行通道，串行阵列单元 1 有 2 个串行通道。各通道能实现 3 线串行（CSI）、UART 和简易 I²C 的通信功能。

R7F0C014B2D、R7F0C014L2D 支持的各通道的功能分配如下：

○ 32 引脚产品

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00 (支持从属选择输入功能)	UART0 (支持 LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

○ 64 引脚产品

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00 (支持从属选择输入功能)	UART0 (支持 LIN-bus)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

在单元 0 的通道 0 和通道 1 使用“UART0”时，不能使用 CSI00 和 CSI01，但是能使用通道 2 和通道 3 的 CSI10、UART1 和 IIC10。

注意 本章使用 64 引脚产品的单元和通道结构进行以下的主要说明。

14.1 串行阵列单元的功能

R7F0C014B2D、R7F0C014L2D 支持的各串行接口的特征如下所示。

14.1.1 3 线串行 I/O (CSI00、CSI01、CSI10、CSI11、CSI20、CSI21)

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“14.5 3 线串行 I/O (CSI00、CSI01、CSI10、CSI11、CSI20、CSI21) 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: $\text{Max. } f_{\text{MCK}}/2$ (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。

- 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 31 章 电特性”。

14.1.2 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, 而且还能通过组合定时器阵列单元和外部中断 (INTP0) 来支持 LIN-bus。

有关具体的设定例子, 请参照“14.7 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。当选项字节 (000C2H) 的 FRQSEL4 位为“0”时, 只有以下的 UART 才能设定 SNOOZE 模式。

- UART0

UART0 (单元 0 的通道 0 和通道 1) 支持 LIN-bus。

[LIN-bus 功能]

• 唤醒信号的检测 • 间隔段 (BF) 的检测 • 同步段的测量、波特率的计算	}	使用外部中断 (INTP0) 和定时器阵列单元。

注 只有以下的 UART 支持 9 位的数据长度。

- UART0

14.1.3 简易 I²C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)

这是通过串行时钟 (SCL) 和串行数据 (SDA) 共 2 条线与多个设备进行时钟同步通信的功能。因为此简易 I²C 是为了与 EEPROM、闪存、A/D 转换器等设备进行单通信而设计的，所以只用作主控设备。

对于开始条件和停止条件，必须遵守 AC 规格，在操作控制寄存器的同时通过软件进行处理。

有关具体的设定例子，请参照“14.9 简易 I²C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21) 通信的运行”。

[数据的发送和接收]

- 主控发送、主控接收 (只限于单主控的主控功能)
- ACK 输出功能注、ACK 检测功能
- 8 位数据长度 (在发送地址时，用高 7 位指定地址，用最低位进行 R/W 控制)
- 开始条件和停止条件的手动产生

[中断功能]

- 传送结束中断

[错误检测标志]

- ACK 错误、溢出错误

※ [简易 I²C 不支持的功能]

- 从属发送、从属接收
- 仲裁失败检测功能
- 等待检测功能

注 在接收最后的数据时，如果给 SOEmn 位 (串行输出允许寄存器 m (SOEm)) 写“0”来停止串行通信数据的输出，就不输出 ACK。详细内容请参照“14.9.3(2) 处理流程”。

备注 在使用全功能的 I²C 总线时，请参照“第 15 章 串行接口 IICA”。

14.2 串行阵列单元的结构

串行阵列单元由以下硬件构成。

表 14-1 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、SCK01、SCK10、SCK11、SCK20、SCK21 引脚 (用于 3 线串行 I/O)、SCL00、SCL01、SCL10、SCL11、SCL20、SCL21 引脚 (用于简易 I ² C)
串行数据输入	SI00、SI01、SI10、SI11、SI20、SI21 引脚 (用于 3 线串行 I/O)、RxD0 (用于支持 LIN-bus 的 UART)、RxD1 引脚和 RxD2 引脚 (用于 UART)
串行数据输出	SO00、SO01、SO10、SO11、SO20、SO21 引脚 (用于 3 线串行 I/O)、TxD0 (用于支持 LIN-bus 的 UART)、TxD1 引脚和 TxD2 引脚 (用于 UART)
串行数据输入 / 输出	SDA00、SDA01、SDA10、SDA11、SDA20、SDA21 引脚 (用于简易 I ² C)
从属选择输入	SSI00 引脚 (用于从属选择输入功能)
控制寄存器	<单元设定部的寄存器> • 外围允许寄存器 0 (PER0) • 串行时钟选择寄存器 m (SPSm) • 串行通道允许状态寄存器 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 m (SSCm) • 输入切换控制寄存器 (ISC) • 噪声滤波器允许寄存器 0 (NFEN0) <各通道部的寄存器> • 串行数据寄存器 mn (SDRmn) • 串行模式寄存器 mn (SMRmn) • 串行通信运行设定寄存器 mn (SCRmn) • 串行状态寄存器 mn (SSRmn) • 串行标志清除触发寄存器 mn (SIRmn) • 端口输入模式寄存器 0、1、3、5 (PIM0、PIM1、PIM3、PIM5) • 端口输出模式寄存器 0、1、3、5、7 (POM0、POM1、POM3、POM5、POM7) • 端口模式寄存器 0、1、3、5 ~ 7 (PM0、PM1、PM3、PM5 ~ PM7) • 端口寄存器 0、1、3、5 ~ 7 (P0、P1、P3、P5 ~ P7)

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。

- 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、01、10、11、20、21)
q: UART 号 (q=0 ~ 2) r: IIC 号 (r=00、01、10、11、20、21)

图 14-1 串行阵列单元 0 的框图

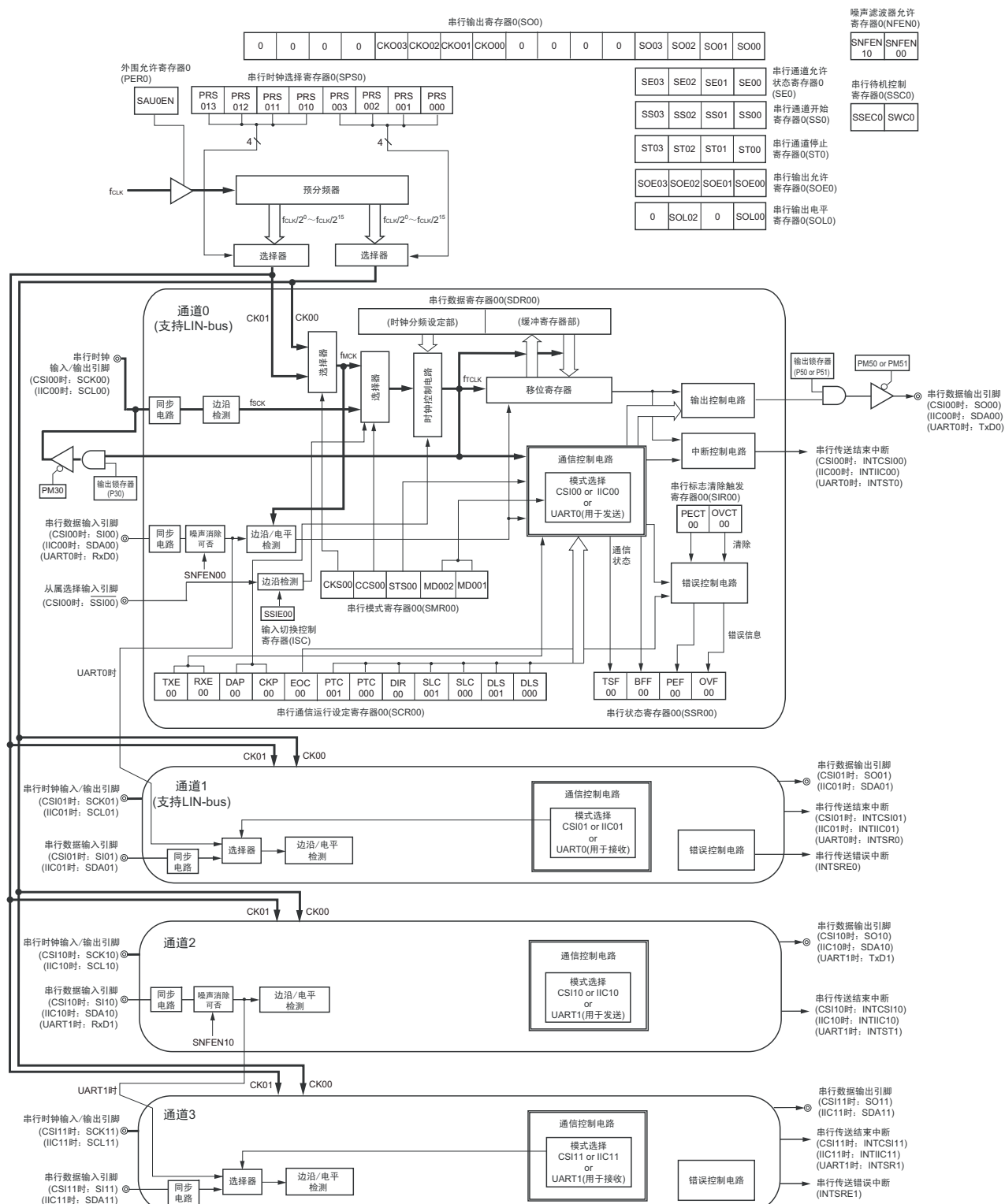
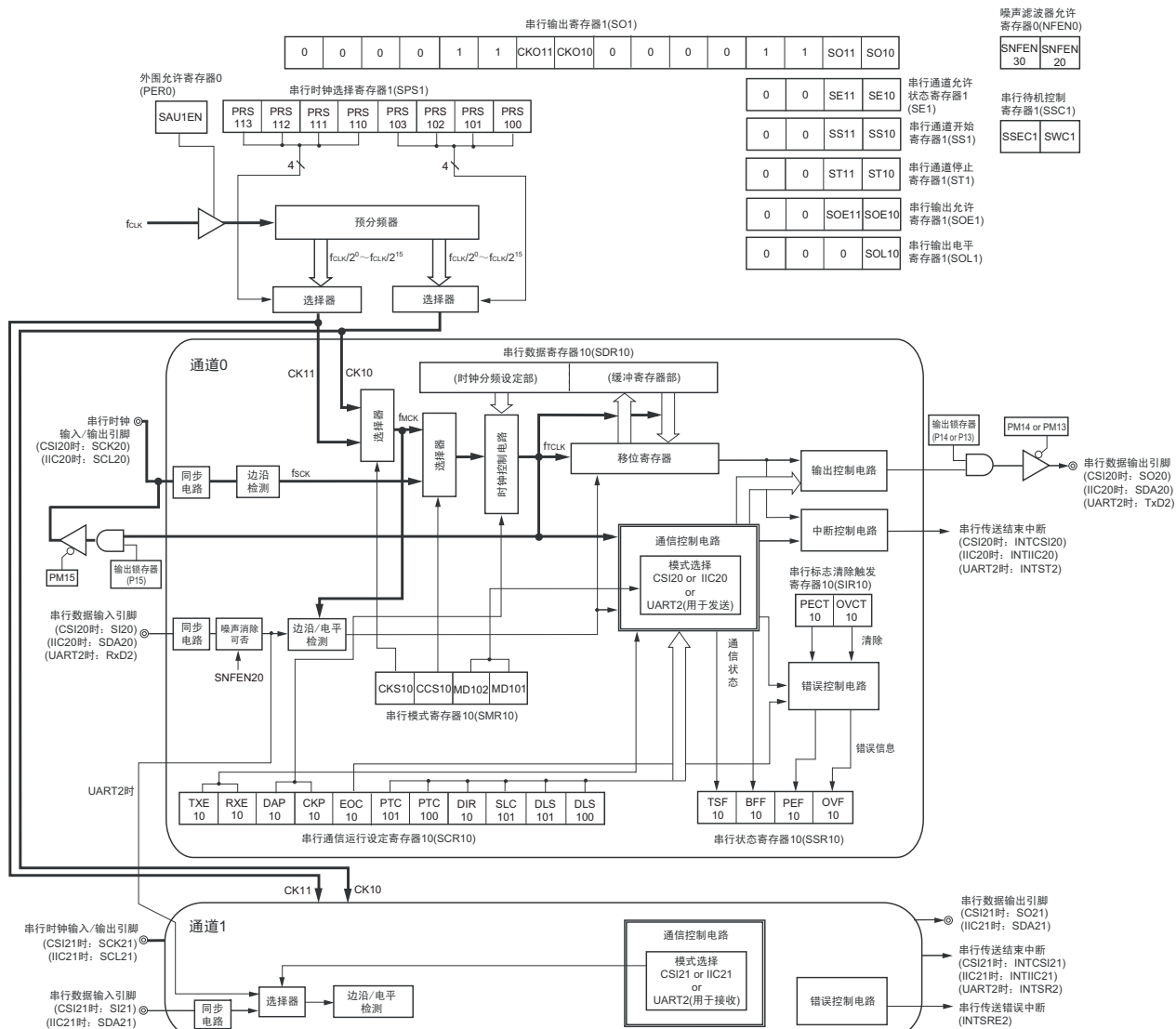


图 14-2 串行阵列单元 1 的框图



14.2.1 移位寄存器

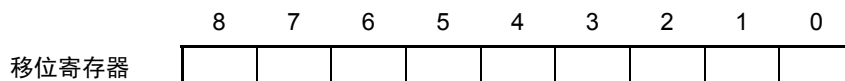
这是进行并行和串行相互转换的 9 位寄存器。

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注 1。

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。



14.2.2 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 位的数据长度。

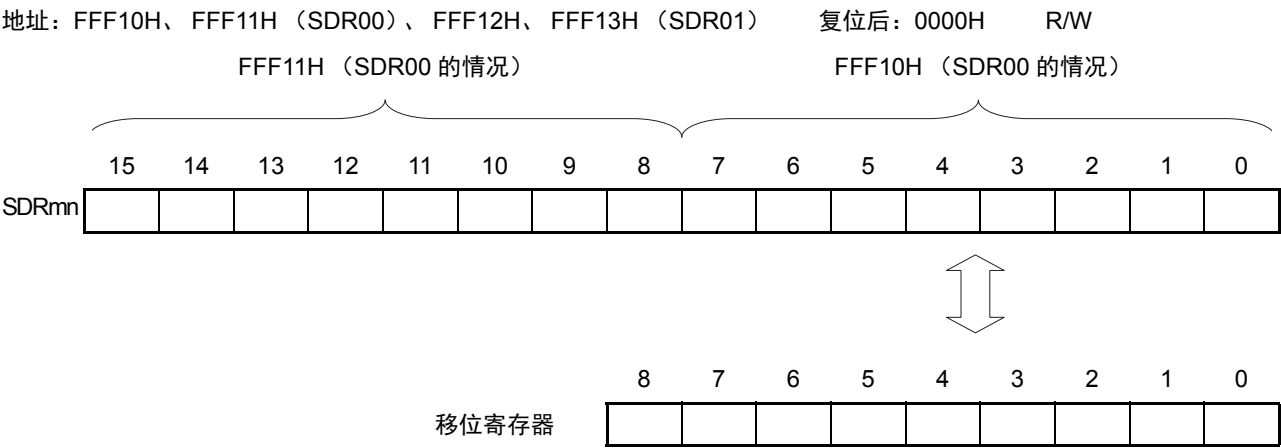
- UART0

2. 在停止运行（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”。

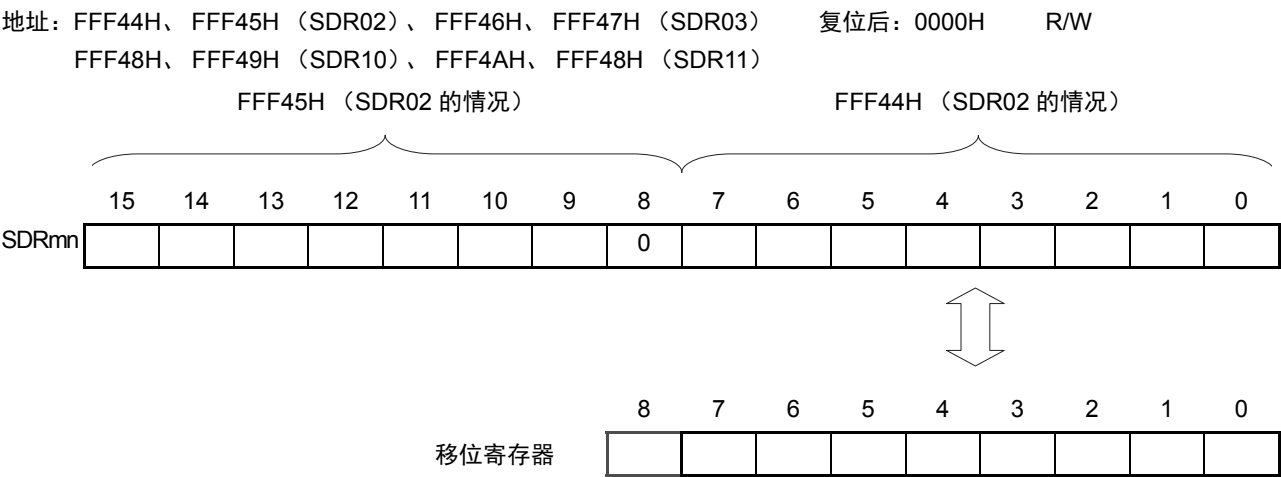
2. m: 单元号（m=0、1） n: 通道号（n=0 ~ 3） p: CSI 号（p=00、01、10、11、20、21）
q: UART 号（q=0 ~ 2） r: IIC 号（r=00、01、10、11、20、21）

图 14-3 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“14.3 控制串行阵列单元的寄存器”。

图 14-4 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“14.3 控制串行阵列单元的寄存器”。

14.3 控制串行阵列单元的寄存器

控制串行阵列单元的寄存器如下所示：

- 外围允许寄存器0 (PER0)
- 串行时钟选择寄存器m (SPSm)
- 串行模式寄存器mn (SMRmn)
- 串行通信运行设定寄存器mn (SCRmn)
- 串行数据寄存器mn (SDRmn)
- 串行标志清除触发寄存器mn (SIRmn)
- 串行状态寄存器mn (SSRmn)
- 串行通道开始寄存器m (SSm)
- 串行通道停止寄存器m (STm)
- 串行通道允许状态寄存器m (SEm)
- 串行输出允许寄存器m (SOEm)
- 串行输出电平寄存器m (SOLm)
- 串行输出寄存器m (SOM)
- 串行待机控制寄存器m (SSCm)
- 输入切换控制寄存器 (ISC)
- 噪声滤波器允许寄存器0 (NFEN0)
- 端口输入模式寄存器0、1、3、5 (PIM0、PIM1、PIM3、PIM5)
- 端口输出模式寄存器0、1、3、5、7 (POM0、POM1、POM3、POM5、POM7)
- 端口模式寄存器0、1、3、5~7 (PM0、PM1、PM3、PM5~PM7)
- 端口寄存器0、1、3、5~7 (P0、P1、P3、P5~P7)

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3)

14.3.1 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 14-5 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	RTCEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

SAUmEN	提供串行阵列单元 m 的输入时钟的控制
0	停止提供输入时钟。 - 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	允许提供输入时钟。 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，忽视串行阵列单元 m 的控制寄存器的写操作，而且读取值都为初始值（输入切换控制寄存器 (ISC)、噪声滤波器允许寄存器 0 (NFEN0)、端口输入模式寄存器 0、1、3、5 (PIM0、PIM1、PIM3、PIM5)、端口输出模式寄存器 0、1、3、5、7 (POM0、POM1、POM3、POM5、POM7)、端口模式寄存器 0、1、3、5 ~ 7 (PM0、PM1、PM3、PM5 ~ PM7)、端口模式控制寄存器 0、1、10、12、14 (PMC0、PMC1、PM10、PMC12、PMC14) 以及端口寄存器 0、1、3、5 ~ 7 (P0、P1、P3、P5 ~ P7) 除外)。

- 串行时钟选择寄存器 m (SPSm)
- 串行模式寄存器 mn (SMRmn)
- 串行通信运行设定寄存器 mn (SCRmn)
- 串行数据寄存器 mn (SDRmn)
- 串行标志清除触发寄存器 mn (SIRmn)
- 串行状态寄存器 mn (SSRmn)
- 串行通道开始寄存器 m (SSm)
- 串行通道停止寄存器 m (STm)
- 串行通道允许状态寄存器 m (SEm)
- 串行输出允许寄存器 m (SOEm)
- 串行输出电平寄存器 m (SOLm)
- 串行输出寄存器 m (SOM)
- 串行待机控制寄存器 m (SSCm)

2. 必须将以下的位置“0”。

bit1 和 bit6

14.3.2 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 14-6 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} =32MHz
0	0	0	0	f _{CLK}	2MHz	5MHz	10MHz	20MHz	32MHz
0	0	0	1	f _{CLK} /2	1MHz	2.5MHz	5MHz	10MHz	16MHz
0	0	1	0	f _{CLK} /2 ²	500kHz	1.25MHz	2.5MHz	5MHz	8MHz
0	0	1	1	f _{CLK} /2 ³	250kHz	625kHz	1.25MHz	2.5MHz	4MHz
0	1	0	0	f _{CLK} /2 ⁴	125kHz	313kHz	625kHz	1.25MHz	2MHz
0	1	0	1	f _{CLK} /2 ⁵	62.5kHz	156kHz	313kHz	625kHz	1MHz
0	1	1	0	f _{CLK} /2 ⁶	31.3kHz	78.1kHz	156kHz	313kHz	500kHz
0	1	1	1	f _{CLK} /2 ⁷	15.6kHz	39.1kHz	78.1kHz	156kHz	250kHz
1	0	0	0	f _{CLK} /2 ⁸	7.81kHz	19.5kHz	39.1kHz	78.1kHz	125kHz
1	0	0	1	f _{CLK} /2 ⁹	3.91kHz	9.77kHz	19.5kHz	39.1kHz	62.5kHz
1	0	1	0	f _{CLK} /2 ¹⁰	1.95kHz	4.88kHz	9.77kHz	19.5kHz	31.3kHz
1	0	1	1	f _{CLK} /2 ¹¹	977Hz	2.44kHz	4.88kHz	9.77kHz	15.6kHz
1	1	0	0	f _{CLK} /2 ¹²	488Hz	1.22kHz	2.44kHz	4.88kHz	7.8kHz
1	1	0	1	f _{CLK} /2 ¹³	244Hz	610Hz	1.22kHz	2.44kHz	3.9kHz
1	1	1	0	f _{CLK} /2 ¹⁴	122Hz	305Hz	610Hz	1.22kHz	1.95kHz
1	1	1	1	f _{CLK} /2 ¹⁵	61Hz	153kHz	305Hz	610Hz	977Hz

注 要在串行阵列单元 (SAU) 运行过程中更改被选择为 f_{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

14.3.3 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 14-7 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	CKS mn	CCS mn	0	0	0	0	0	STS mn 注	0	SIS mn0 注	1	0	0	MD mn2	MD mn1	MD mn0

CKS mn	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})。	

CCS mn	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。	

STS mn 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 (在 SMR00、SMR02、SMR10 寄存器时，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、01、10、11、20、21)
q: UART 号 (q=0 ~ 2) r: IIC 号 (r=00、01、10、11、20、21)

图 14-7 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	SIS mn0 注	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0 注	UART 模式中的通道 n 接收数据的电平反相控制
0	将下降沿检测为起始位。 不将输入的通信数据进行反相。
1	将上升沿检测为起始位。 将输入的通信数据进行反相。

MD mn2	MD mn1	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。

MD mn0	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（在 SMR00、SMR02、SMR10 寄存器时，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、01、10、11、20、21）
q: UART 号（q=0～2） r: IIC 号（r=00、01、10、11、20、21）

14.3.4 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 14-8 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

TXE mn	RXE mn	通道 n 运行模式的设定
0	0	禁止通信。
0	1	只进行接收。
1	0	只进行发送。
1	1	进行发送和接收。

DAP mn	CKP mn	CSI 模式中的数据与时钟的相位选择	类型
0	0		1
0	1		2
1	0		3
1	1		4
在 UART 模式和简易 I ² C 模式中, 必须将 DAPmn 位和 CKPmn 位都置“0”。			

EOC mn	错误中断信号 (INTSREx (x=0 ~ 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 I ² C 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、01、10、11、20、21)

图 14-8 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

PTC mn1	PTC mn0	UART 模式中的奇偶校验位的设定	
		发送	接收
0	0	不输出奇偶校验位。	接收时没有奇偶校验。
0	1	输出奇偶校验注 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”。

DIR mn	CSI 和 UART 模式中的数据传送顺序的选择
0	进行 MSB 优先的输入 / 输出。
1	进行 LSB 优先的输入 / 输出。

在简易 I²C 模式中, 必须将 DIRmn 位置“0”。

SLCm n1 注1	SLC mn0	UART 模式中的停止位的设定
0	0	无停止位
0	1	停止位长度 =1 位
1	0	停止位长度 =2 位 (只限于 mn=00、02、10、12)
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 个停止位 (SLCmn1、SLCmn0=0、1) 或者 2 位 (SLCmn1、SLCmn0=1、0)。

DLSm n1 注2	DLS mn0	CSI 和 UART 模式中的数据长度的设定
0	1	9 位数据长度 (保存在 SDRmn 寄存器的 bit0 ~ 8) (只在 UART 模式中可选择)
1	0	7 位数据长度 (保存在 SDRmn 寄存器的 bit0 ~ 6)
1	1	8 位数据长度 (保存在 SDRmn 寄存器的 bit0 ~ 7)
其他		禁止设定。

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、01、10、11、20、21)

14.3.5 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 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、SDR10、SDR11 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, 只有在运行停止状态 ($SEmn=0$) 时才能读写高 7 位。在运行中 ($SEmn=1$) 只能写 SDRmn 寄存器的低 8 位或者低 9 位, 而且 SDRmn 寄存器高 7 位的读取值总是“0”。

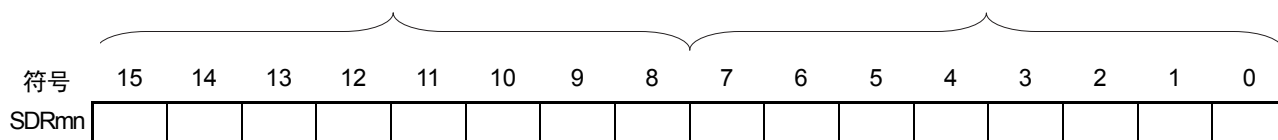
在产生复位信号后, SDRmn 寄存器的值变为“0000H”。

图 14-9 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)

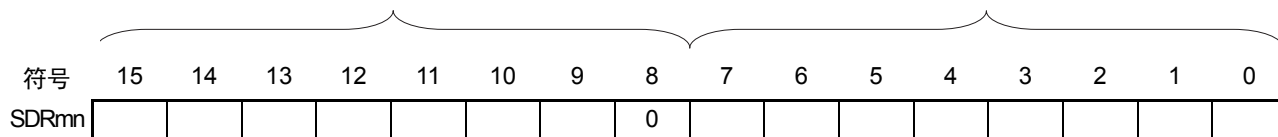


地址: FFF44H、FFF45H (SDR02)、FFF46H、FFF47H (SDR03) 复位后: 0000H R/W

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“00000000B”和“0000001B”。

3. 在使用简易 I²C 时, 禁止将 SDRmn[15:9] 设定为“00000000B”, SDRmn[15:9] 的设定值必须大于等于“00000001B”。

4. 在停止运行 ($SEmn=0$) 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“14.2 串行阵列单元的结构”。

2. m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)

14.3.6 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 SIR_{mn} 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，SIR_{mn}寄存器的值变为“0000H”。

图 14-10 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	FECT mn注	PEC Tmn	OVC Tmn

FEC Tmn 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”。

PEC Tmn	通道 n 奇偶校验错误标志的清除触发
0	不清除。
1	将 SSRmn 寄存器的 PEFmn 位清“0”。

OVC Tmn	通道 n 溢出错误标志的清除触发
0	不清除。
1	将 SSRmn 寄存器的 OVFmn 位清 “0”。

注 只限于 SIR01、SIR03、SIR11 寄存器。

注意 必须将 bit15 ~ 3（在 SIR00、SIR02、SIR10 寄存器时，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”。

14.3.7 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 14-11 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	通信处于停止或者待机状态。
1	通信处于运行状态。
[清除条件]	
- 当将 STmn 寄存器的 STmn 位置“1”（通信停止状态）或者将 SSm 寄存器的 SSmn 位置“1”（通信待机状态）时 - 当通信结束时	
[置位条件]	
当开始通信时	

BFFmn	通道 n 缓冲寄存器的状态表示标志
0	SDRmn 寄存器没有保存有效数据。
1	SDRmn 寄存器保存了有效数据。
[清除条件]	
- 在发送过程中传送完从 SDRmn 寄存器到移位寄存器的发送数据时 - 在接收过程中从 SDRmn 寄存器读完接收数据时 - 当将 STmn 寄存器的 STmn 位置“1”（通信停止状态）或者将 SSm 寄存器的 SSmn 位置“1”（通信允许状态）时	
[置位条件]	
- 在 SCRmn 寄存器的 TXEmn 位为“1”（各通信模式中的发送模式、发送和接收模式）的状态下给 SDRmn 寄存器写发送数据时 - 在 SCRmn 寄存器的 RXEmn 位为“1”（各通信模式中的接收模式、发送和接收模式）的状态下将接收数据保存到 SDRmn 寄存器时 - 当发生接收错误时	

注 只限于 SSR01、SSR03、SSR11 寄存器。

注意 在 SNOOZE 模式（SWCm=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 14-11 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 (SWCm=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)

14.3.8 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 14-12 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 OVFn 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”。

14.3.9 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 14-13 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 运行的开始触发
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”。

14.3.10 串行通道允许状态寄存器 m（SEm）

SEm 寄存器用于确认各通道的串行发送和接收的允许或者停止状态。

如果给串行开始允许寄存器 m（SSm）的各位写“1”，就将其对应的位置“1”。如果给串行通道停止寄存器 m（STm）的各位写“1”，就将其对应的位清“0”。

对于允许运行的通道 n，无法通过软件改写后述的串行输出寄存器 m（SOm）的 CKOmn 位（通道 n 的串行时钟输出）的值，而从串行时钟引脚输出由通信运行反映的值。

对于停止运行的通道 n，能通过软件设定 SOm 寄存器的 CKOmn 位的值，并且从串行时钟引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过 16 位存储器操作指令读取 SEm 寄存器。

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

在产生复位信号后，SEm 寄存器的值变为“0000H”。

图 14-14 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	1514131211109876543210	
SE0	0000000000000SE03SE02SE01SE00	

地址: F0160H、F0161H (SE1)	复位后: 0000H	R
符号	1514131211109876543210	
SE1	0000000000000000SE11SE10	

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)

14.3.11 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 14-15 串行输出允许寄存器 m（SOEm）的格式

地址: F012AH、F012BH	复位后: 0000H	R/W														
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	SOE03	SOE02	SOE01	SOE00

地址: F016AH、F016BH	复位后: 0000H	R/W														
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	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)

14.3.12 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。

只有在停止通道运行时 (SEmn=0) 才能通过软件改写此寄存器的 CKOmn 位。当允许通道运行 (SEmn=1) 时，忽视通过软件的改写而只能通过串行通信更改此寄存器的 CKOmn 位的值。

要将串行接口引脚用作端口功能等非串行接口功能时，必须将相应的 CKOmn 位和 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 14-16 串行输出寄存器 m (SOm) 的格式

地址: F0128H、F0129H	复位后: 0F0FH	R/W														
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	CKO 03	CKO 02	CKO 01	CKO 00	0	0	0	0	SO 03	SO 02	SO 01	SO 00

地址: F0168H、F0169H	复位后: 0F0FH	R/W														
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	CKO 11	CKO 10	0	0	0	0	1	1	SO 11	SO 10

CKO mn	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”。

SO mn	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)

14.3.13 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，必须将对应的位置“0”。

只在允许串行输出时（SOEmn=1），将此寄存器的各通道 n 反相设定反映到引脚输出。在禁止串行输出时（SOEmn=0），将 SOMn 位的值直接输出而不反相输出。

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 14-17 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	SOL02	0	SOL00

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	SOL10

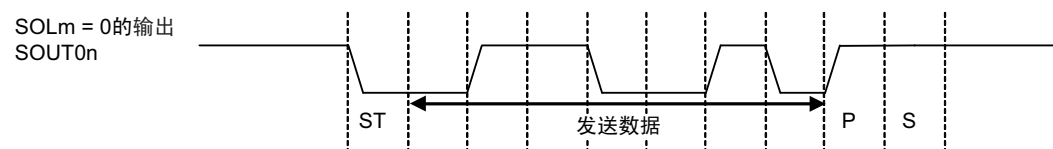
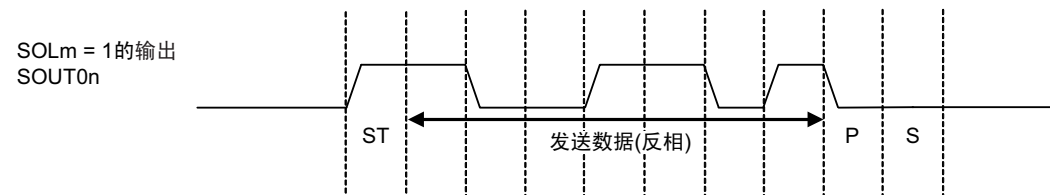
SOLmn	UART 模式中的通道 n 发送数据电平反相的选择														
0	将通信数据直接输出。														
1	将通信数据反相输出。														

注意 必须将 SOL0 寄存器的 bit15 ~ 3 和 bit1 以及 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 14-18 所示。

图 14-18 发送数据的电平反相例子

(a) 非反相输出 ($SOL_{mn} = 0$)(b) 反相输出 ($SOL_{mn} = 1$)

备注 m: 单元号 (m=0、1) n: 通道号 (n=0、2)

14.3.14 串行待机控制寄存器 m（SSCm）

SSC0 寄存器是控制在 CSI00 或者 UART0 的串行数据接收时从 STOP 模式启动接收运行（SNOOZE 模式）的寄存器。

- 通过 16 位存储器操作指令设定 SSCm 寄存器。
- 能用 SSCmL 并且通过 8 位存储器操作指令设定 SSCm 寄存器的低 8 位。
- 在产生复位信号后，SSCm 寄存器的值变为“0000H”。

注意 SNOOZE 模式中的最大传送速率如下所示：

- CSI00：～1Mbps
- UART0：只限于 4800bps（在选项字节（000C2H）的 FRQSEL4 位为“0”时能使用）。

图 14-19 串行待机控制寄存器 m（SSCm）的格式

地址：F0138H（SSC0）	复位后：0000H	R/W														
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSCm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSECm	SWCm

SSECm	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/INTSRE2）。
1	停止产生错误中断（INTSRE0/INTSRE2）。
- 在 SNOOZE 模式中进行 UART 接收的情况下，只有在 SWCm 位为“1”并且 EOCmn 位为“1”时才能将 SSECm 位置“1”或者“0”，否则必须将 SSECm 位置“0”。 - 禁止将 SSECm 位和 SWCm 位分别置“1”和“0”。	

SWCm	SNOOZE 模式的设定
0	不使用 SNOOZE 模式功能。
1	使用 SNOOZE 模式功能。
- 在 STOP 模式中，通过硬件触发信号解除 STOP 模式，并且在 CPU 不运行的状态下进行 CSI/UART 的接收（SNOOZE 模式）。 - 只有在选择高速内部振荡器时钟作为 CPU/ 外围硬件时钟（f_{CLK}）时才能设定 SNOOZE 模式功能，而在选择其他时钟的情况下禁止设定。在选项字节（000C2H）的 FRQSEL4 位为“1”时，即使用作 UART 也禁止设定。 - 即使使用 SNOOZE 模式，也必须在通常运行模式中将 SWCm 位置“0”并且在即将要转移到 STOP 模式前将 SWCm 位改为“1”。 - 另外，必须在从 STOP 模式恢复到通常运行模式后将 SWCm 位改为“0”。	

注意 禁止将 SSECm 位和 SWCm 位分别置“1”和“0”。

图 14-20 在 SNOOZE 模式中进行 UART 接收时的中断

EOCmn 位	SSECm 位	正常接收	接收错误
0	0	产生 INTSRx。	产生 INTSRx。
0	1	产生 INTSRx。	产生 INTSRx。
1	0	产生 INTSRx。	产生 INTSREx。
1	1	产生 INTSRx。	不产生中断。

14.3.15 输入切换控制寄存器（ISC）

在通过 UART0 实现 LIN-bus 通信时，ISC 寄存器的 ISC1 位和 ISC0 位用于协调外部中断和定时器阵列单元的运行。

如果将 bit0 置“1”，就选择串行数据输入（RxD0）引脚的输入信号作为外部中断的输入（INTP0），因此能通过 INTP0 中断检测唤醒信号。

如果将 bit1 置“1”，就选择串行数据输入（RxD0）引脚的输入信号作为定时器的输入，因此能通过定时器检测唤醒信号并且测量间隔段的低电平宽度和同步段的脉宽。

SSI00 位在 CSI00 通信的从属模式中控制通道 0 的 SSI00 引脚输入。在给 SSI00 引脚输入高电平的期间，即使输入串行时钟，也不进行发送和接收。在给 SSI00 引脚输入低电平的期间，如果输入串行时钟，就根据各模式的设定进行发送和接收。

通过 1 位或者 8 位存储器操作指令设定 ISC 寄存器。

在产生复位信号后，ISC 寄存器的值变为“00H”。

图 14-21 输入切换控制寄存器（ISC）的格式

地址: F0073H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ISC	SSI00	0	0	0	0	0	0	0

SSI00	CSI00 通信的从属模式中通道 0 的 SSI00 输入的设定
0	SSI00 引脚输入无效。
1	SSI00 引脚输入有效。

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

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

注意 必须将 bit6 ~ 2 置“0”。

14.3.16 噪声滤波器允许寄存器 0（NFEN0）

NFEN0 寄存器设定噪声滤波器是否用于各通道串行数据输入引脚的输入信号。

对于用于 CSI 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_{CLK}）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 14-22 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”。

14.3.17 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（PMxx）、端口寄存器（Pxx）、端口输入模式寄存器（PIMxx）、端口输出模式寄存器（POMxx）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）”、“4.3.4 端口输入模式寄存器（PIM0、PIM1、PIM3、PIM5）”、“4.3.5 端口输出模式寄存器（POM0、POM1、POM3、POM5、POM7）”和“4.3.6 端口模式控制寄存器 0、12、14（PMC0、PMC12、PMC14）”。

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（P02/ANI17/SO10/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（PMCxx）的位和端口模式寄存器（PMxx）的位置“0”，并且将端口寄存器（Pxx）的位置“1”。

另外，当用于 N 沟道漏极开路输出（ V_{DD} 耐压^{注 1}/ EV_{DD} 耐压^{注 2}）模式时，必须将各端口对应的端口输出模式寄存器（POMxx）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 $EV_{DD} \leq V_{DD}$ 进行的不同电位（1.8V、2.5V、3V）的对应”。

（例）P02/ANI17/SO10/TxD1 用作串行数据输出的情况

将端口模式控制寄存器 0 的 PMC02 位置“0”。

将端口模式寄存器 0 的 PM02 位置“0”。

将端口寄存器 0 的 P02 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（P03/ANI16/SI10/RxD1/SDA10 等）用作串行数据输入或者串行时钟输入时，必须将各端口对应的端口模式寄存器（PMxx）的位置“1”，并且将端口模式控制寄存器（PMCxx）的位置“0”。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。

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（PIMxx）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 $EV_{DD} \leq V_{DD}$ 进行的不同电位（1.8V、2.5V、3V）的对应”。

（例）P03/ANI16/SI10/RxD1/SDA10 用作串行数据输入的情况

将端口模式控制寄存器 0 的 PMC03 位置“0”。

将端口模式寄存器 0 的 PM03 位置“1”。

将端口寄存器 0 的 P03 位置“0”或者“1”。

注 1. 这是 32 引脚产品的情况。

2. 这是 64 引脚产品的情况。

14.4 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。

14.4.1 以单元为单位停止运行的情况

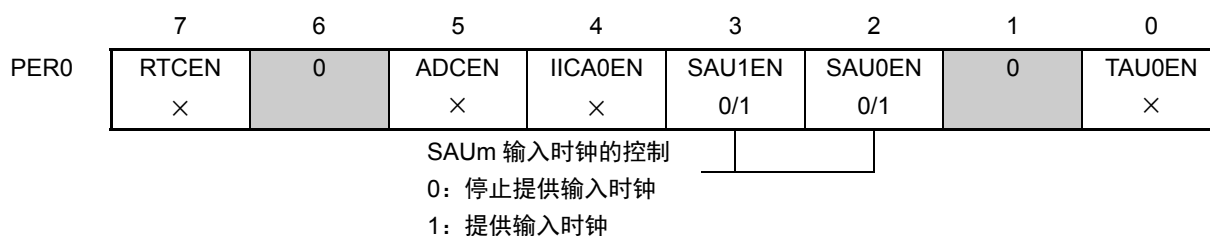
通过外围允许寄存器 0（PER0）设定以单元为单位的停止运行。

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 14-23 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”。



注意 1. 当 SAUmEN 位为“0”时，忽视串行阵列单元 m 的控制寄存器的写操作，而且读取值都为初始值。

但是，以下的寄存器除外：

- 输入切换控制寄存器（ISC）
- 噪声滤波器允许寄存器 0（NFEN0）
- 端口输入模式寄存器 0、1、3、5（PIM0、PIM1、PIM3、PIM5）
- 端口输出模式寄存器 0、1、3、5、7（POM0、POM1、POM3、POM5、POM7）
- 端口模式寄存器 0、1、3、5～7（PM0、PM1、PM3、PM5～PM7）
- 端口寄存器 0、1、3、5～7（P0、P1、P3、P5～P7）

2. 必须将以下的位置“0”。

bit1 和 bit6

备注 : 不能设定（设定初始值）

× : 这是串行阵列单元未使用的位（取决于其他外围功能的设定）。

0/1 : 根据用户的用途置“0”或者“1”。

14.4.2 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 14-24 按通道停止运行时的各寄存器的设定

(a) 串行通道停止寄存器 m (STm) 这是设定允许各通道的通信 / 停止计数的寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	STm3 注	STm2 注	STm1	STm0
													0/1	0/1	0/1	0/1

1: 将 SEmn 位清“0”并且停止通信运行

※ 因为 STmn 位是触发位，所以如果 SEmn 位为“0”就立即清除 STmn 位。

(b) 串行通道允许状态寄存器 m (SEm) 此寄存器表示各通道的数据发送和接收的运行或者停止状态。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	SEm3 注	SEm2 注	SEm1	SEm0
													0/1	0/1	0/1	0/1

0: 停止运行

※SEm 寄存器是只读状态寄存器，通过 STm 寄存器来停止运行。

对于已经停止运行的通道，能通过软件设定 SOM 寄存器的 CKOmn 位的值。

(c) 串行输出允许寄存器 m (SOEm) 这是设定允许或者停止各通道串行通信输出的寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3 注	SOEm2 注	SOEm1	SOEm0
													0/1	0/1	0/1	0/1

0: 通过串行通信运行来停止输出

※ 对于已经停止串行输出的通道，能通过软件设定 SOM 寄存器的 SOMn 位的值。

(d) 串行输出寄存器 m (SOM) 这是各通道串行输出的缓冲寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	CKOm3	CKOm2	CKOm1	CKOm0	0	0	0	0	SOM3 注	SOM2 注	SOM1	SOM0
					0/1	0/1	0/1	0/1					0/1	0/1	0/1	0/1

1: 串行时钟的输出值为“1”

1: 串行数据的输出值为“1”

※ 当将各通道对应的引脚用作端口功能时，必须将相应的 CKOmn 位和 SOMn 位置“1”。

注 在串行阵列单元 1 时，不能设定（设定初始值）。

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”。

14.5 3 线串行 I/O（CSI00、CSI01、CSI10、CSI11、CSI20、CSI21）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: $\text{Max. } f_{\text{MCK}}/2$ （只限于 CSI00）

主控通信: $\text{Max. } f_{\text{CLK}}/4$

从属通信: $\text{Max. } f_{\text{MCK}}/6$

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

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。

- 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 31 章 电特性”。

SAU0 的通道 0 ~ 3 以及 SAU1 的通道 0 和通道 1 是支持 3 线串行 I/O（CSI00、CSI01、CSI10、CSI11、CSI20、CSI21）的通道。

○ 32 引脚产品

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00 (支持从属选择输入功能)	UART0 (支持 LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

○ 64 引脚产品

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00 (支持从属选择输入功能)	UART0 (支持 LIN-bus)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

3 线串行 I/O（CSI00、CSI01、CSI10、CSI11、CSI20、CSI21）有以下 7 种通信运行：

- 主控发送 (参照 14.5.1)
- 主控接收 (参照 14.5.2)
- 主控发送和接收 (参照 14.5.3)
- 从属发送 (参照 14.5.4)
- 从属接收 (参照 14.5.5)
- 从属发送和接收 (参照 14.5.6)
- SNOOZE 模式功能 (参照 14.5.7)

14.5.1 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	CSI01	CSI10	CSI11	CSI20	CSI21
对象通道	SAU0 的通道 0	SAU0 的通道 1	SAU0 的通道 2	SAU0 的通道 3	SAU1 的通道 0	SAU1 的通道 1
使用的引脚	SCK00、SO00	SCK01、SO01	SCK10、SO10	SCK11、SO11	SCK20、SO20	SCK21、SO21
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	无					
传送数据长度	7 位或者 8 位					
传送速率 ^注	Max. $f_{MCK}/2[\text{Hz}]$ （只限于 CSI00）， $f_{CLK}/4[\text{Hz}]$ Min. $f_{CLK}/(2 \times 2^{15} \times 128)[\text{Hz}]$ 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: 不反相 • 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 31 章 电特性”）的范围内使用。

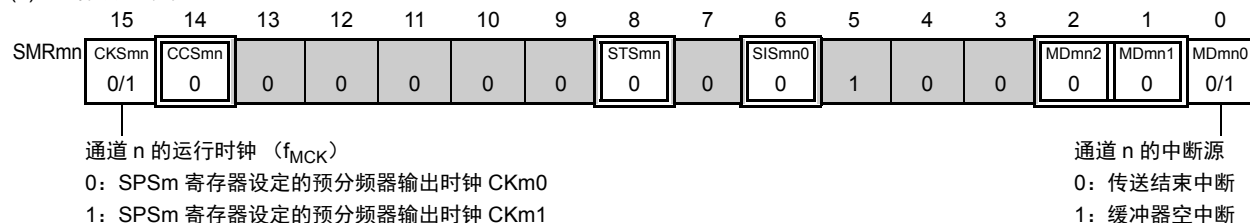
备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) mn=00 ~ 03、10、11

(1) 寄存器的设定

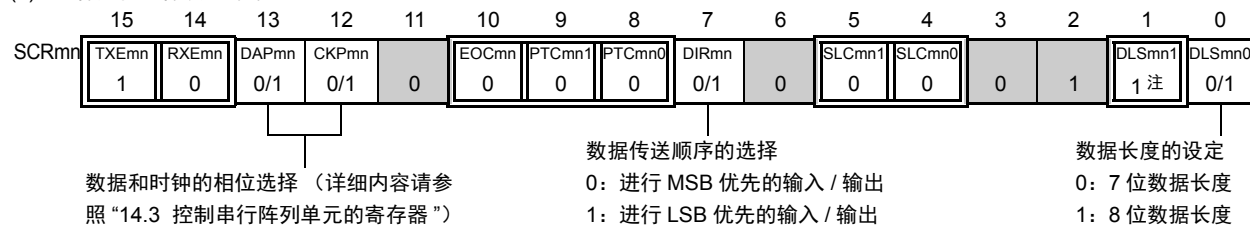
图 14-25 3 线串行 I/O (CSI00、CSI01、CSI10、CSI11、CSI20、CSI21)

主控发送时的寄存器设定内容例子

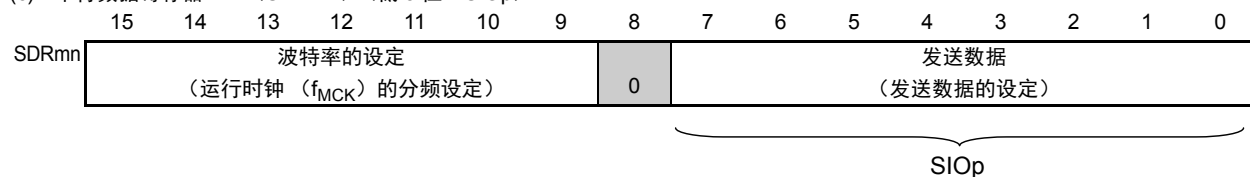
(a) 串行模式寄存器 mn (SMRmn)



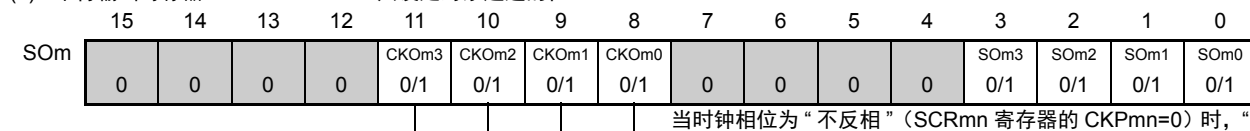
(b) 串行通信运行设定寄存器 mn (SCRmn)



(c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)

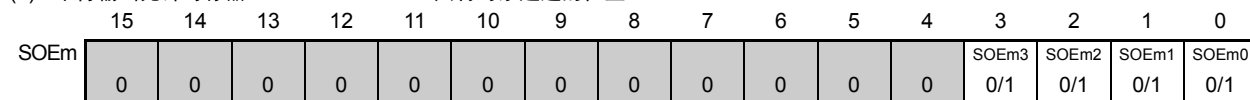


(d) 串行输出寄存器 m (SOM) 只设定对象通道的位。

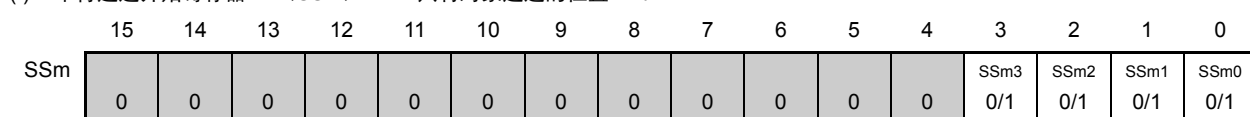


当时钟相位为“不反相”(SCRmn 寄存器的 CKPmn=0) 时, “1” 为开始通信; 当时钟相位为“反相”(CKPmn=1) 时, “0” 为开始通信。

(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、01、10、11、20、21)

mn=00~03、10、11

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 14-26 主控发送的初始设定步骤

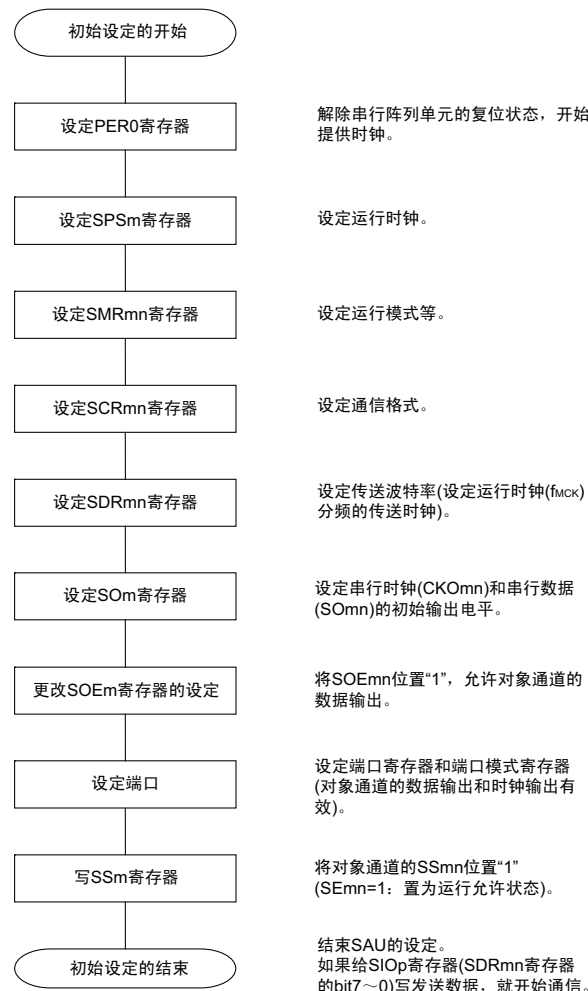


图 14-27 主控发送的中止步骤

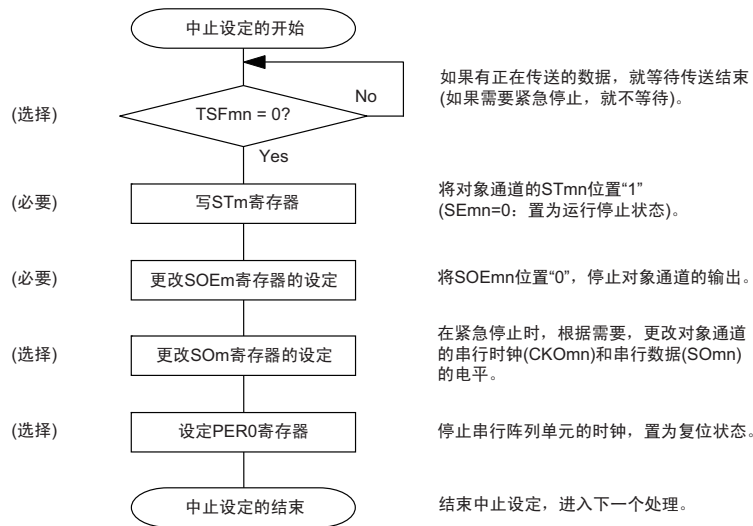


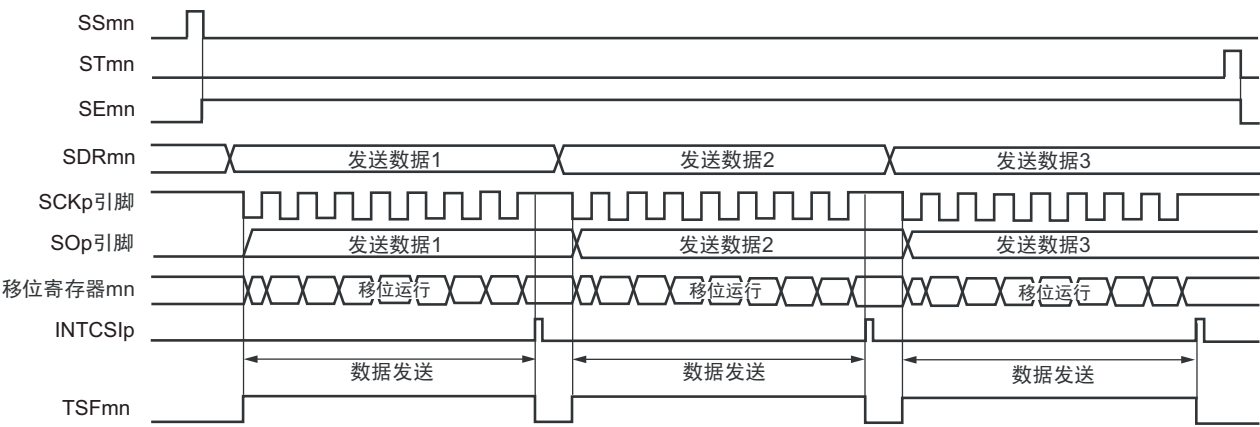
图 14-28 重新开始主控发送的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（从属设备）停止或者通信结束后进行初始设定而不是进行重新开始设定。

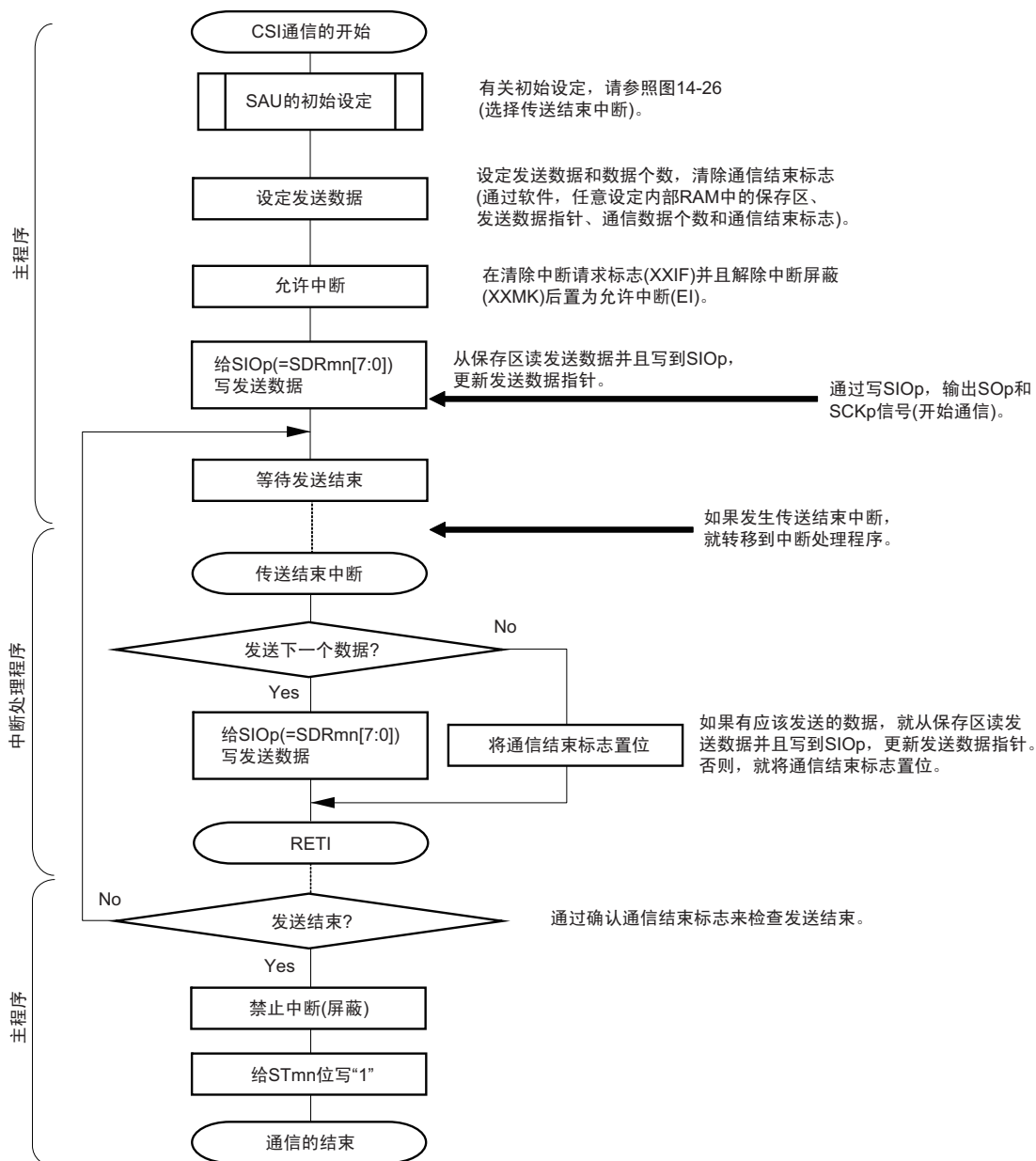
(3) 处理流程（单次发送模式）

图 14-29 主控发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



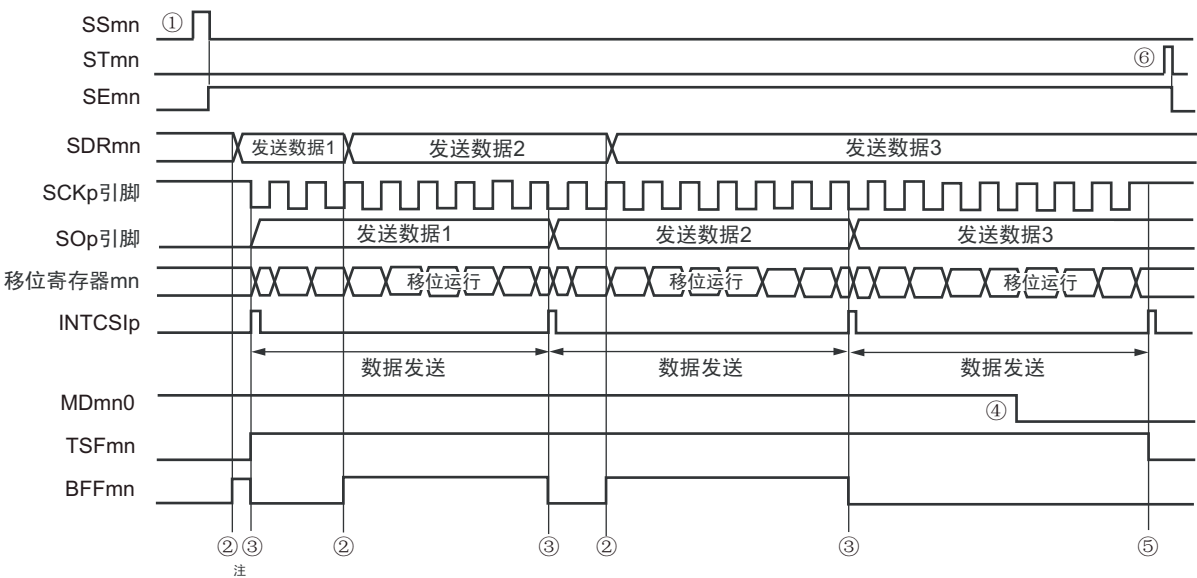
备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) p: CSI 号 (p=00、01、10、11、20、21)
mn=00 ~ 03、10、11

图 14-30 主控发送（单次发送模式）的流程图



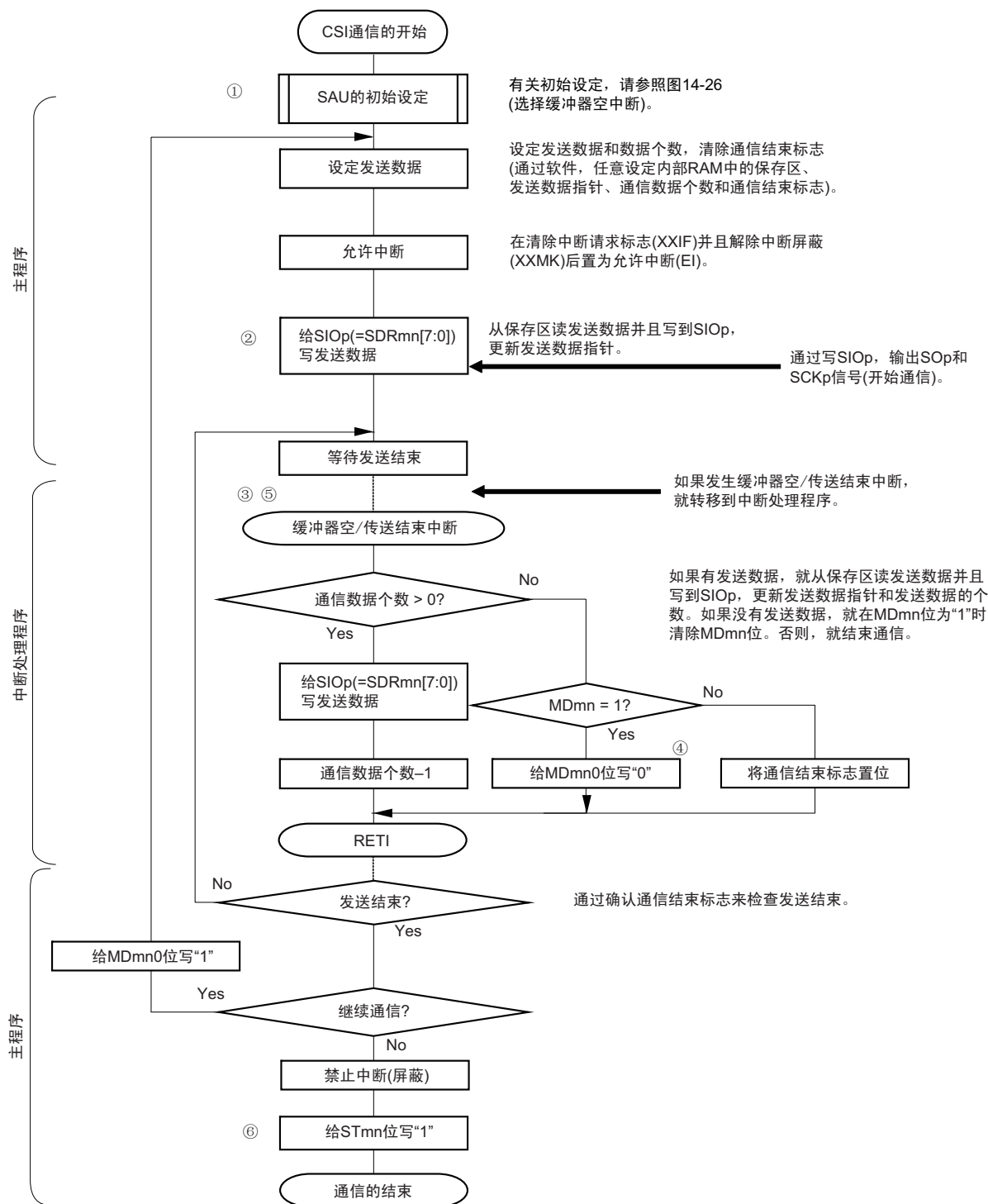
(4) 处理流程（连续发送模式）

图 14-31 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、01、10、11、20、21）
mn=00～03、10、11

图 14-32 主控发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图 14-31 主控发送（连续发送模式）的时序图”中的①~⑥。

14.5.2 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	CSI01	CSI10	CSI11	CSI20	CSI21
对象通道	SAU0 的通道 0	SAU0 的通道 1	SAU0 的通道 2	SAU0 的通道 3	SAU1 的通道 0	SAU1 的通道 1
使用的引脚	SCK00、SI00	SCK01、SI01	SCK10、SI10	SCK11、SI11	SCK20、SI20	SCK21、SI21
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7 位或者 8 位					
传送速率 ^注	Max. $f_{MCK}/2$ [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: 不反相 • 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 31 章 电特性”）的范围内使用。

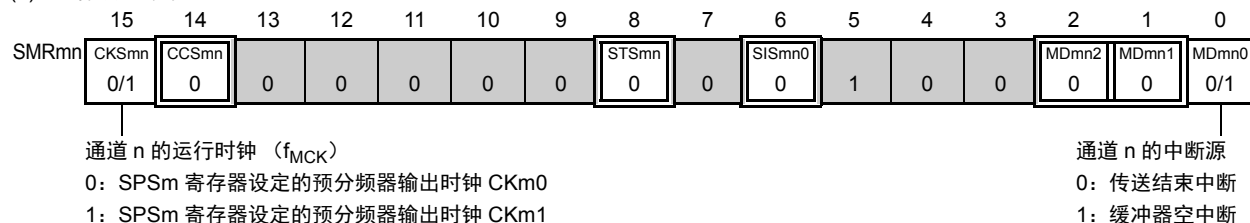
备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) p: CSI 号 (p=00、01、10、11、20、21)
mn=00 ~ 03、10、11

(1) 寄存器的设定

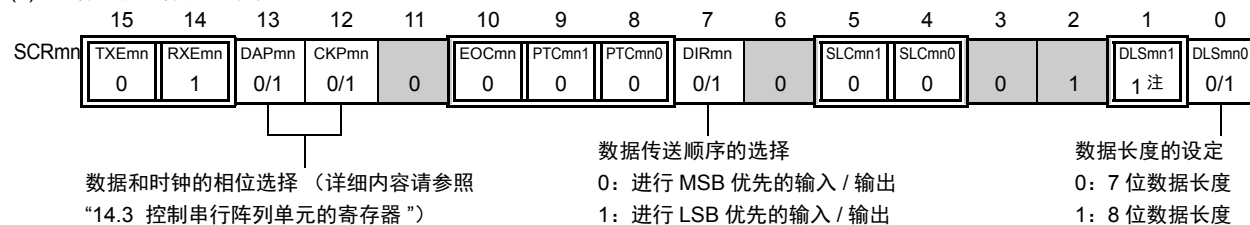
图 14-33 3 线串行 I/O (CSI00、CSI01、CSI10、CSI11、CSI20、CSI21)

主控接收时的寄存器设定内容例子

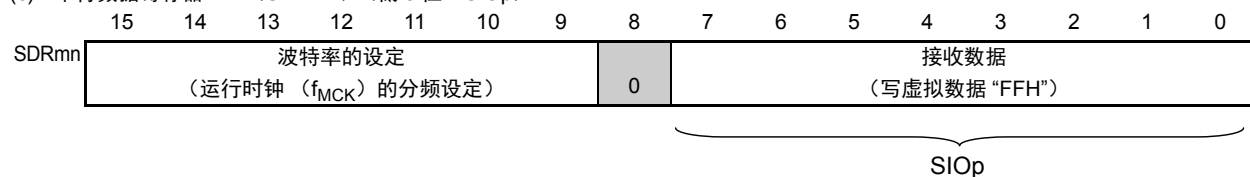
(a) 串行模式寄存器 mn (SMRmn)



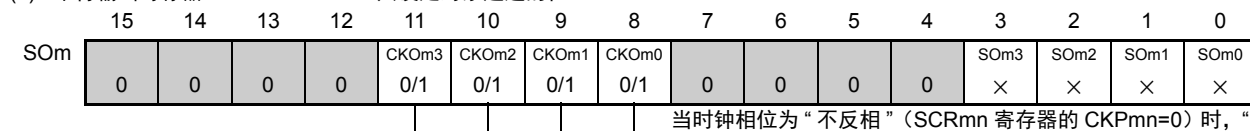
(b) 串行通信运行设定寄存器 mn (SCRmn)



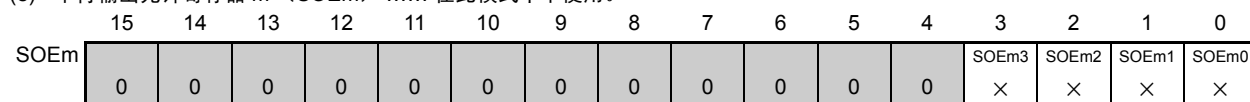
(c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



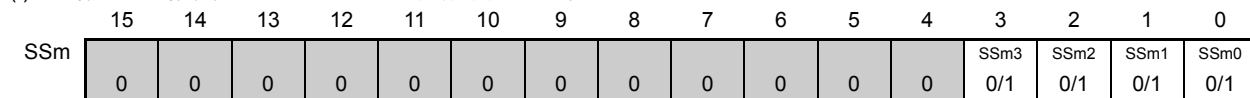
(d) 串行输出寄存器 m (SOm) 只设定对象通道的位。



(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、01、10、11、20、21)
mn=00~03、10、11

2. : 在 CSI 主控接收模式中为固定设定。 : 不能设定 (设定初始值)。
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 14-34 主控接收的初始设定步骤

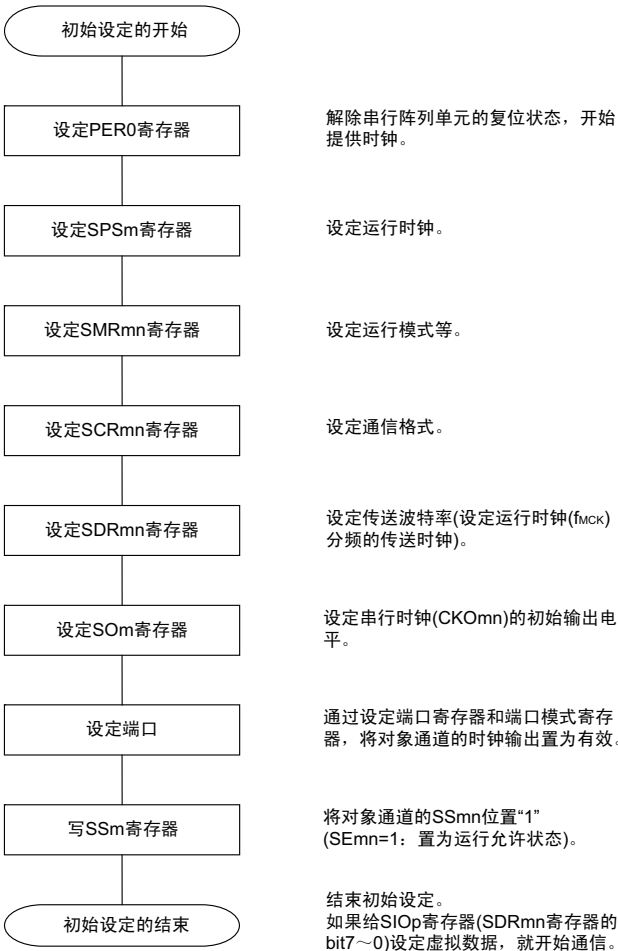


图 14-35 主控接收的中止步骤

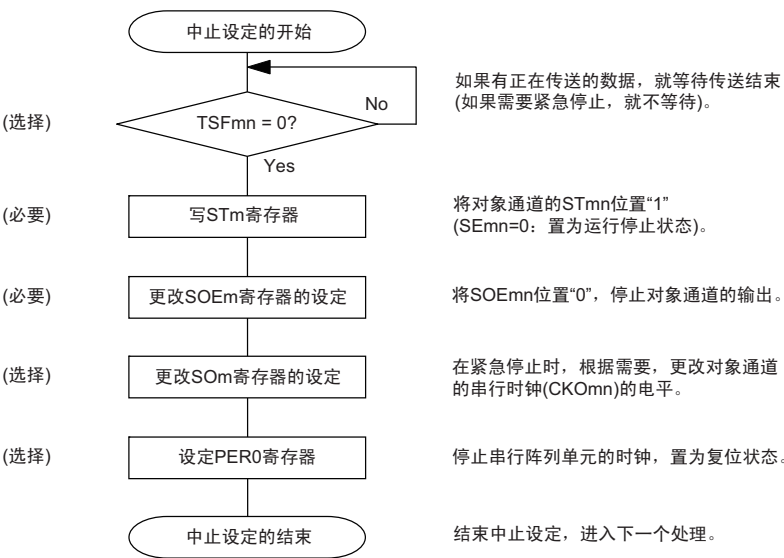
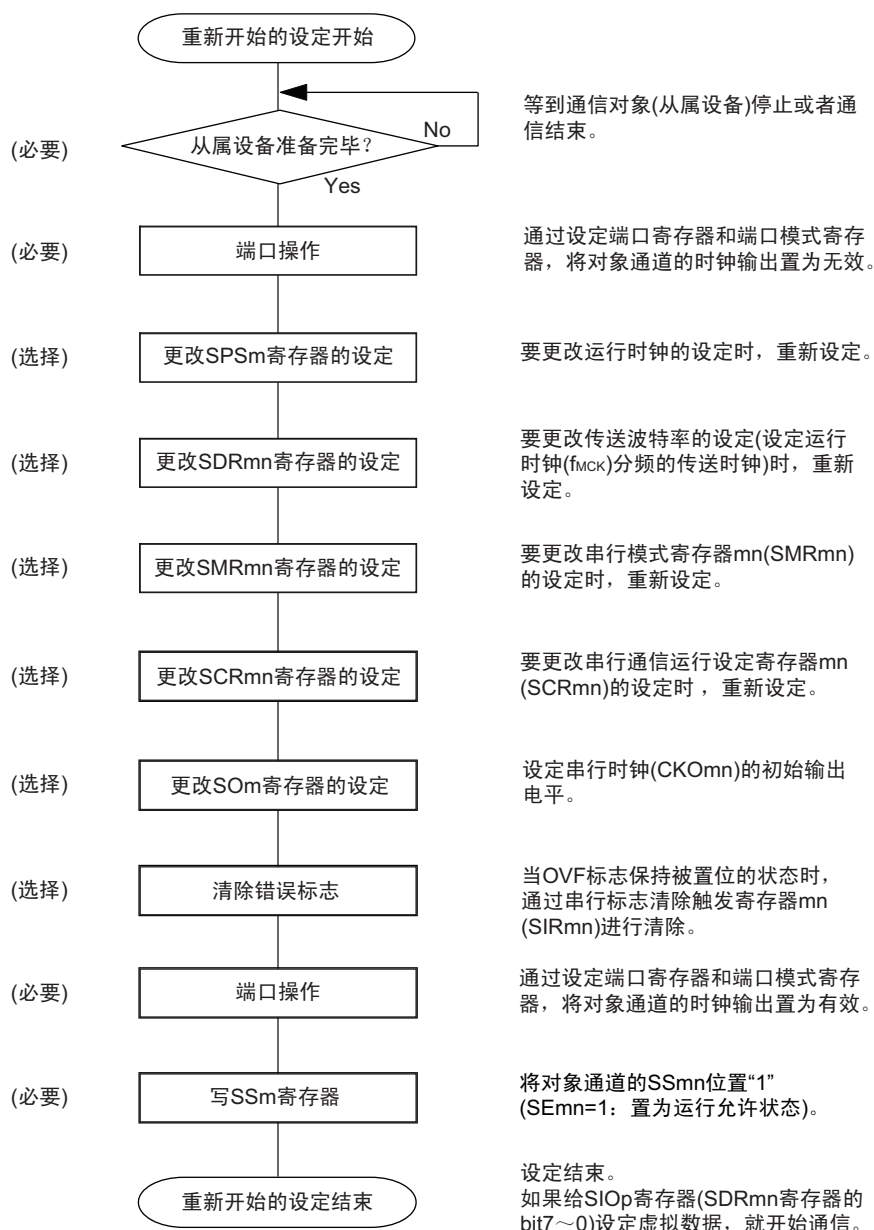


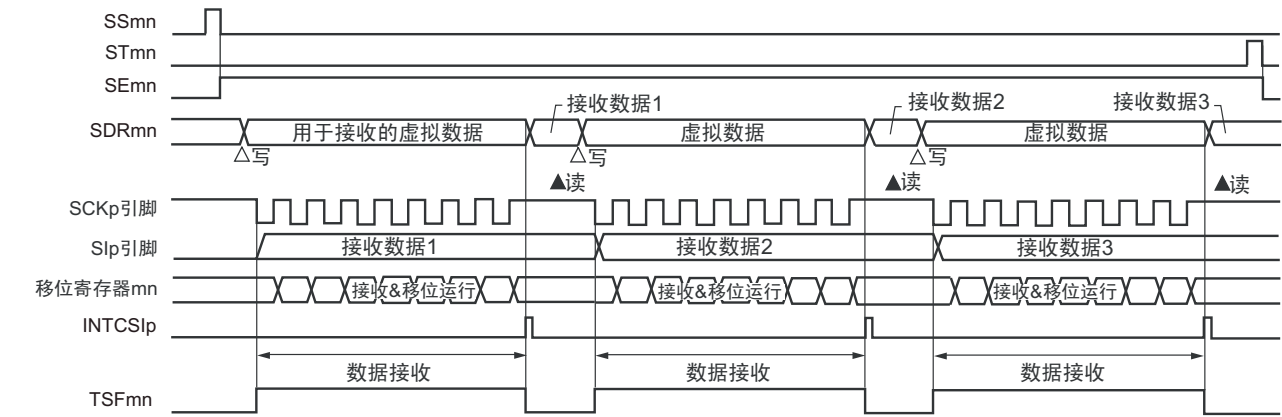
图 14-36 重新开始主控接收的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（从属设备）停止或者通信结束后进行初始设定而不是进行重新开始设定。

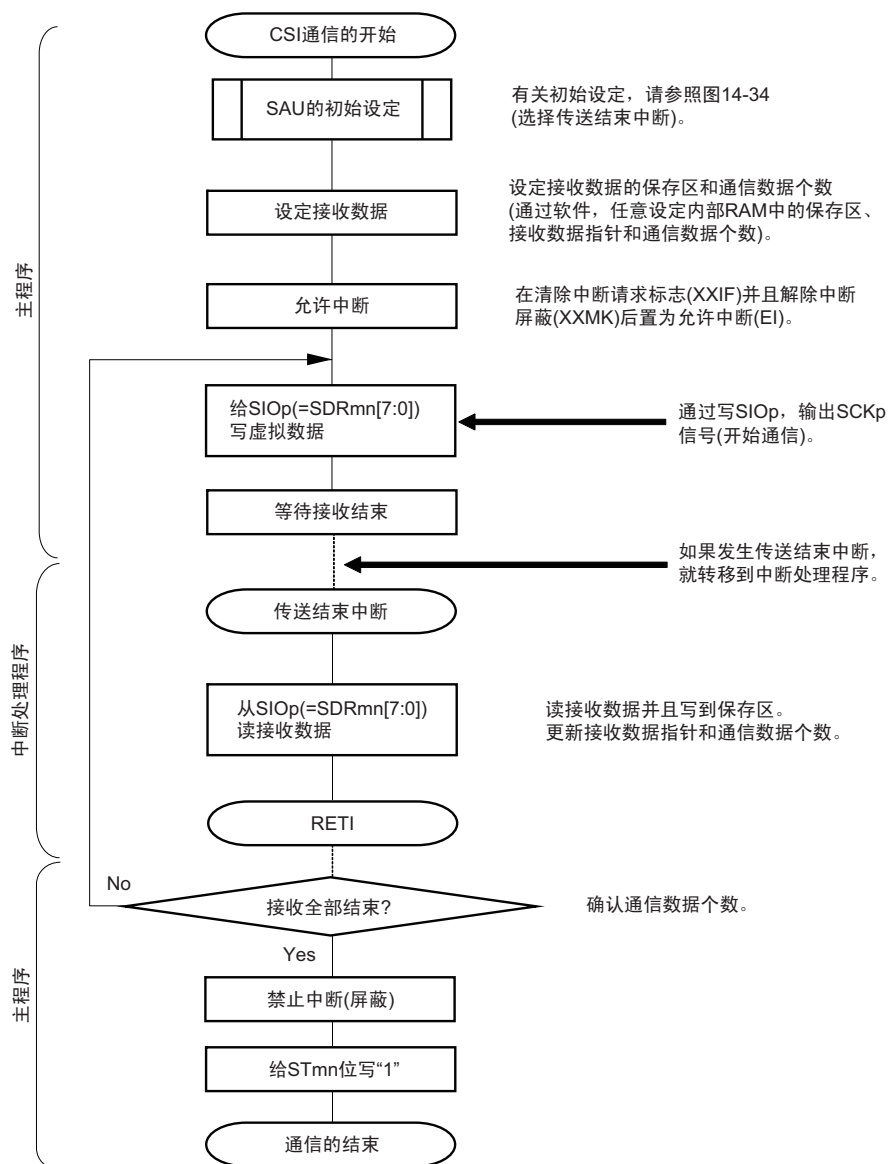
(3) 处理流程（单次接收模式）

图 14-37 主控接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



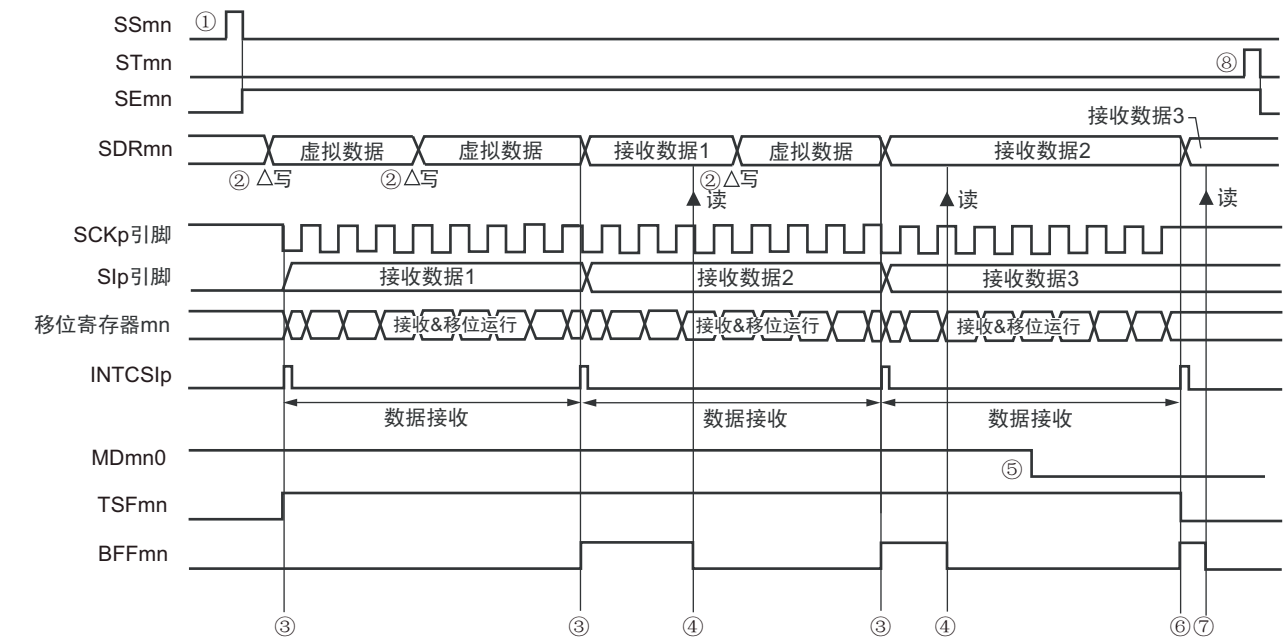
备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) p: CSI 号 (p=00、01、10、11、20、21)
mn=00 ~ 03、10、11

图 14-38 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）

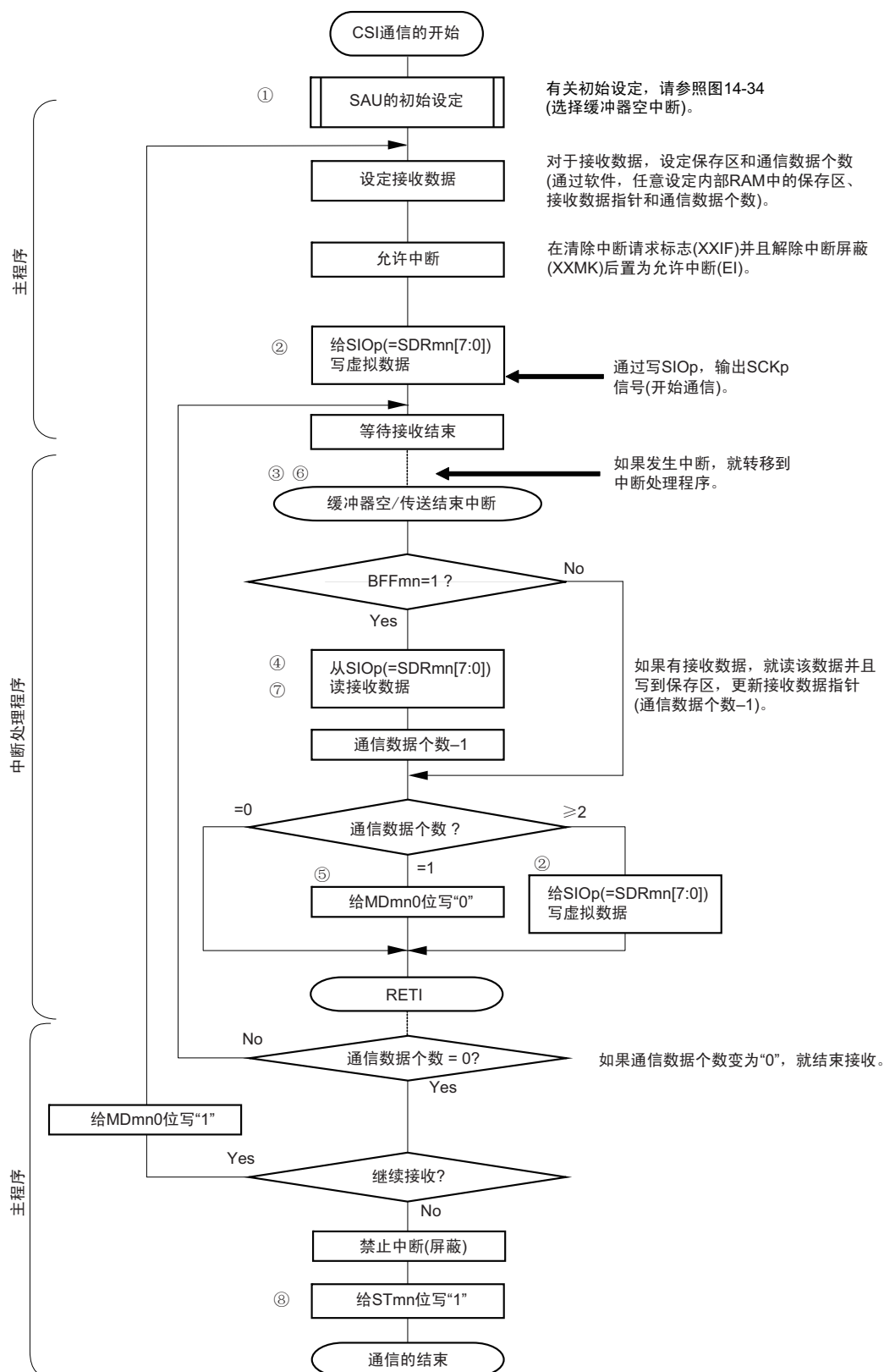
图 14-39 主控接收（连续接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



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 14-40 主控接收（连续接收模式）的流程图”中的①～⑧。
2. m: 单元号 (m=0、1) n: 通道号 (n=0～3) p: CSI 号 (p=00、01、10、11、20、21)
mn=00～03、10、11

图 14-40 主控接收（连续接收模式）的流程图



备注 图中的①~⑧对应“图 14-39 主控接收（连续接收模式）的时序图”中的①~⑧。

14.5.3 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	CSI01	CSI10	CSI11	CSI20	CSI21
对象通道	SAU0 的通道 0	SAU0 的通道 1	SAU0 的通道 2	SAU0 的通道 3	SAU1 的通道 0	SAU1 的通道 1
使用的引脚	SCK00、SI00、SO00	SCK01、SI01、SO01	SCK10、SI10、SO10	SCK11、SI11、SO11	SCK20、SI20、SO20	SCK21、SI21、SO21
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	只有溢出错检测标志（OVFmn）。					
传送数据长度	7 位或者 8 位					
传送速率 ^注	Max. $f_{MCK}/2$ [Hz]（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: 不反相 • CKPmn=1: 反相					
数据方向	MSB 优先或者 LSB 优先					

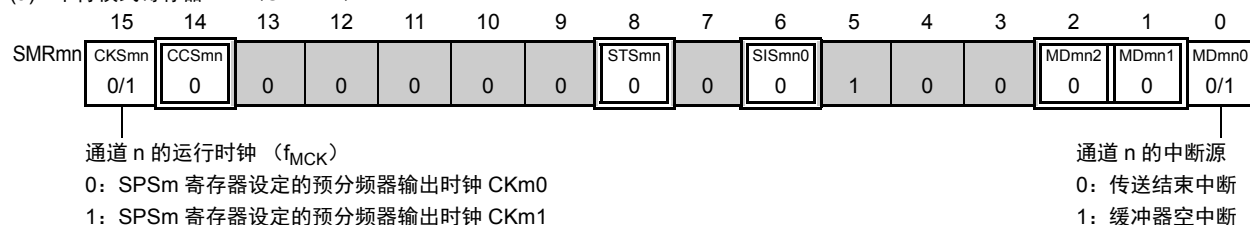
注 必须在满足此条件并且满足电特性的外围功能特性（参照“第 31 章 电特性”）的范围内使用。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) p: CSI 号 (p=00、01、10、11、20、21)
mn=00 ~ 03、10、11

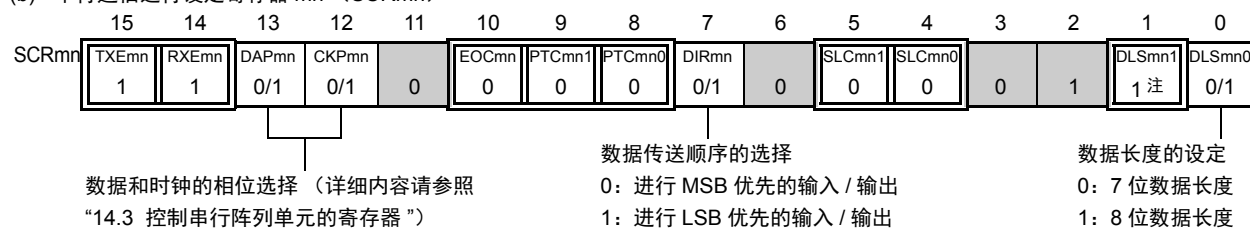
(1) 寄存器的设定

图 14-41 3 线串行 I/O (CSI00、CSI01、CSI10、CSI11、CSI20、CSI21)
主控发送和接收时的寄存器设定内容例子

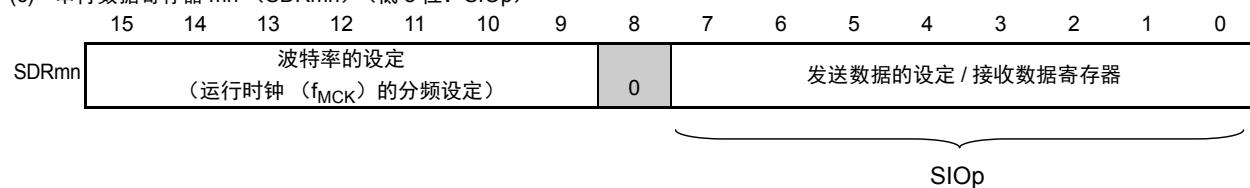
(a) 串行模式寄存器 mn (SMRmn)



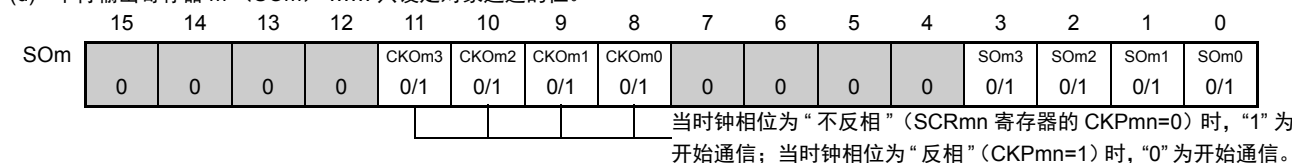
(b) 串行通信运行设定寄存器 mn (SCRmn)



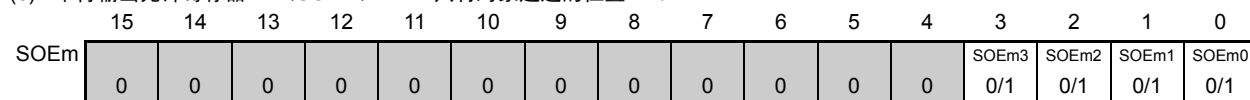
(c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



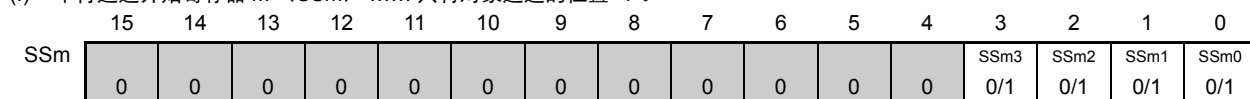
(d) 串行输出寄存器 m (SOm) 只设定对象通道的位。



(e) 串行输出允许寄存器 m (SOEm) 只将对象通道的位置“1”。



(f) 串行通道开始寄存器 m (SSm) 只将对象通道的位置“1”。



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、01、10、11、20、21)
mn=00~03、10、11

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 14-42 主控发送和接收的初始设定步骤

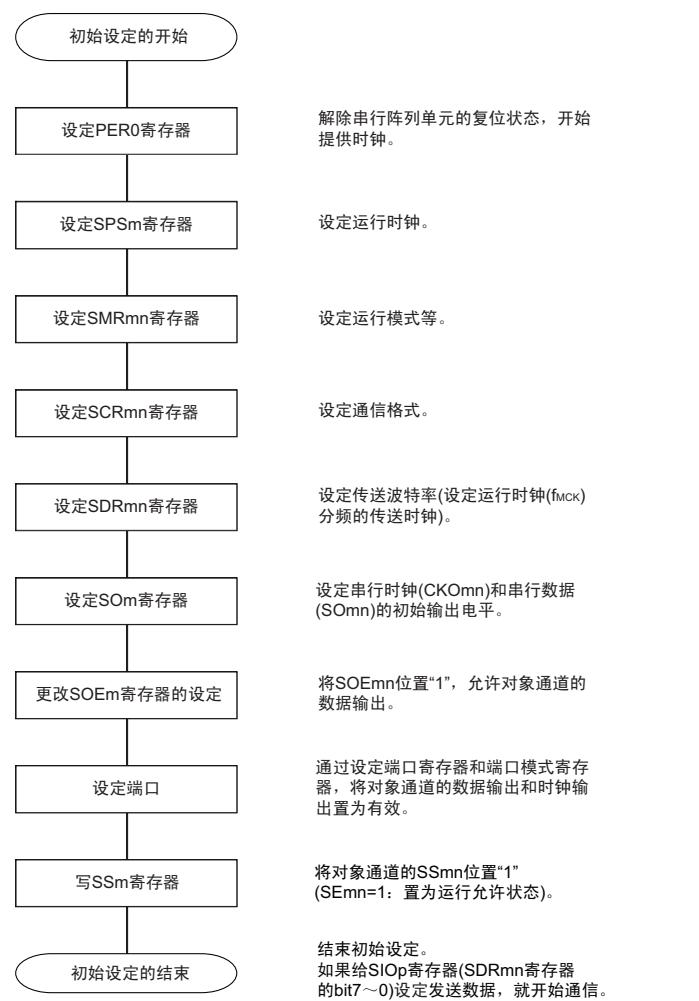


图 14-43 主控发送和接收的中止步骤

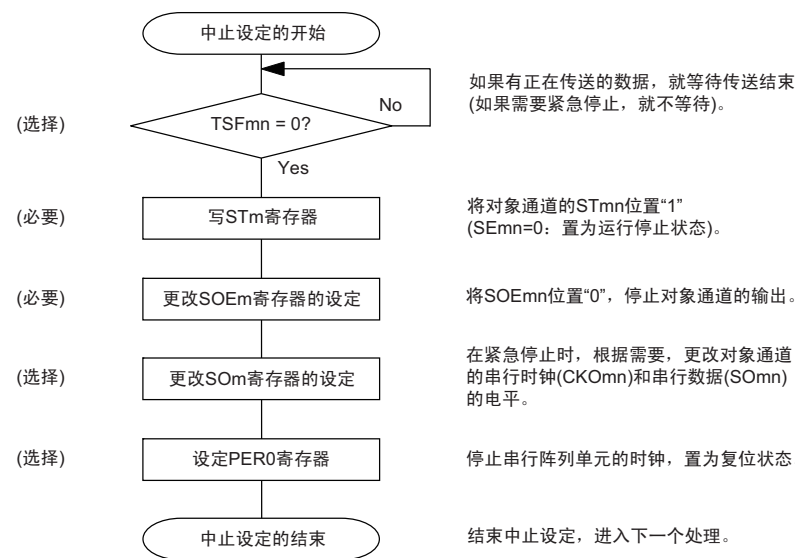
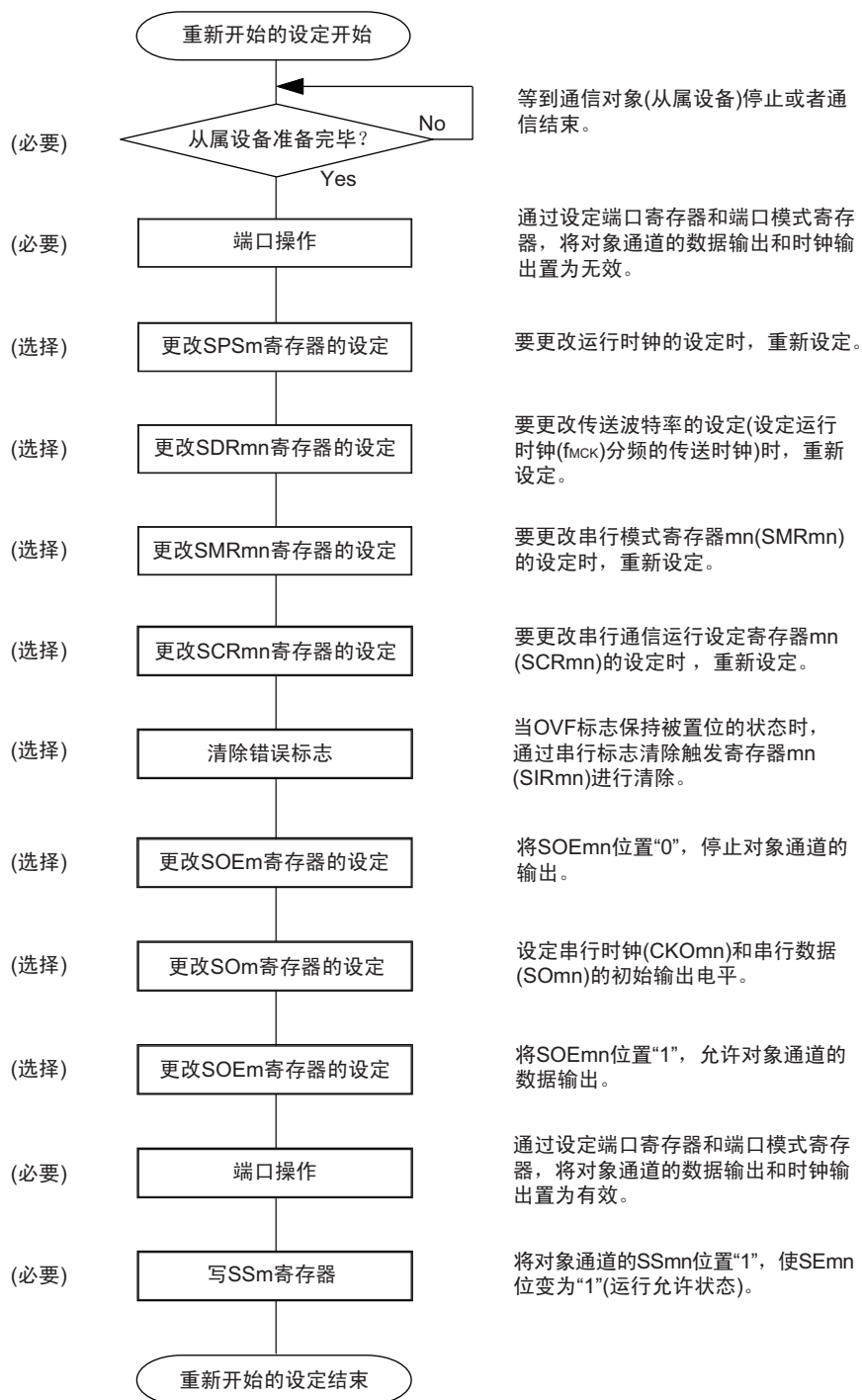
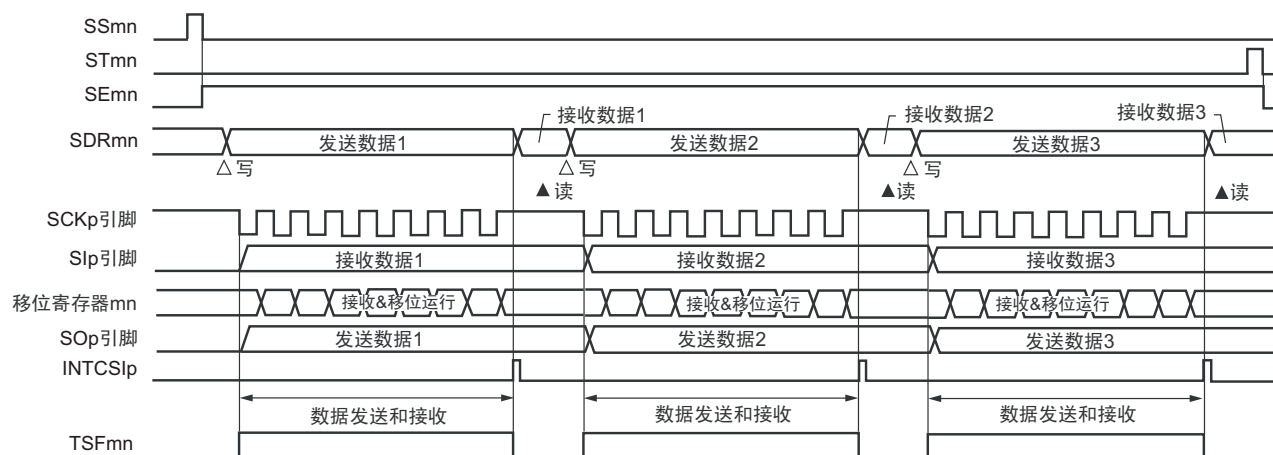


图 14-44 重新开始主控发送和接收的设定步骤



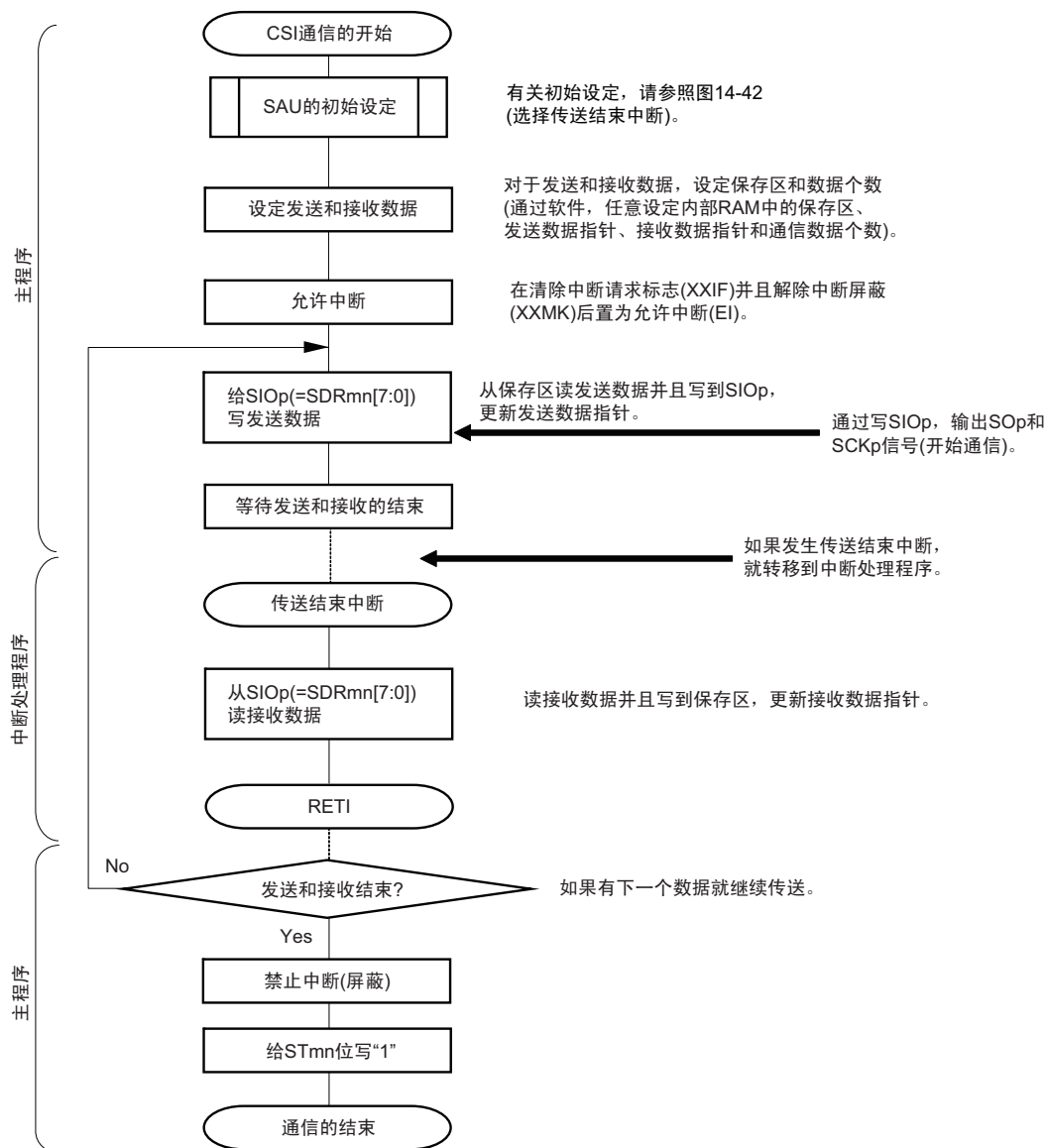
(3) 处理流程（单次发送和接收模式）

图 14-45 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）



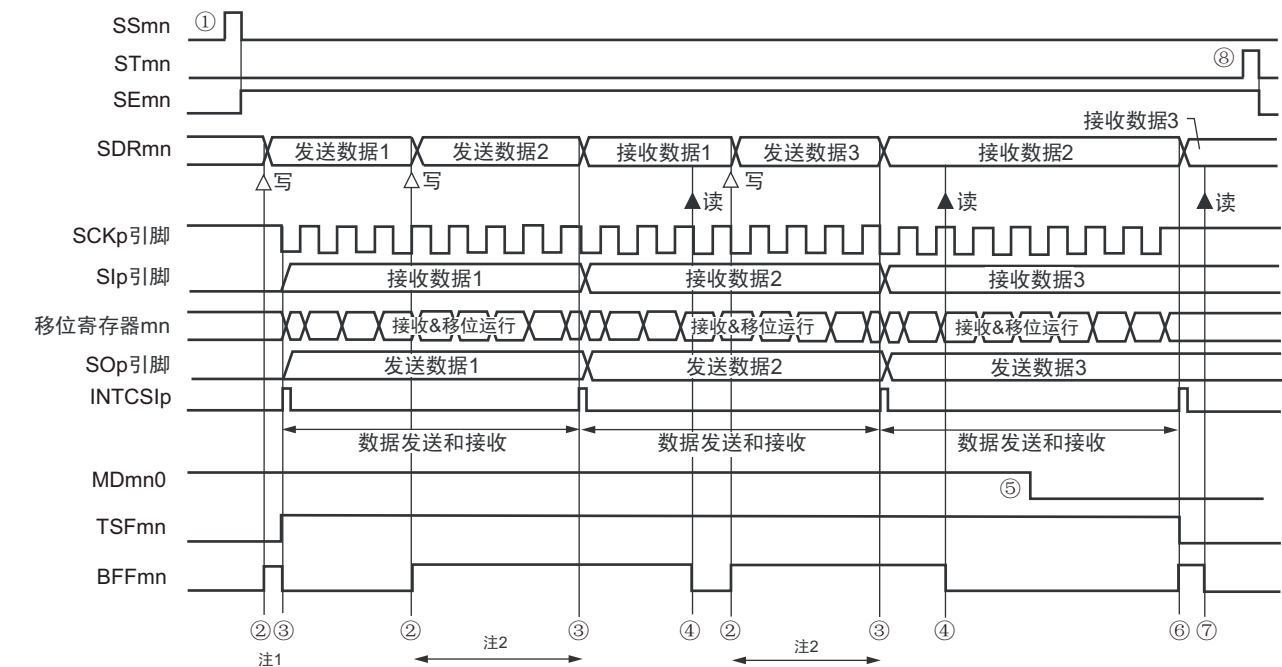
备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) p: CSI 号 (p=00、01、10、11、20、21)
mn=00 ~ 03、10、11

图 14-46 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 14-47 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）

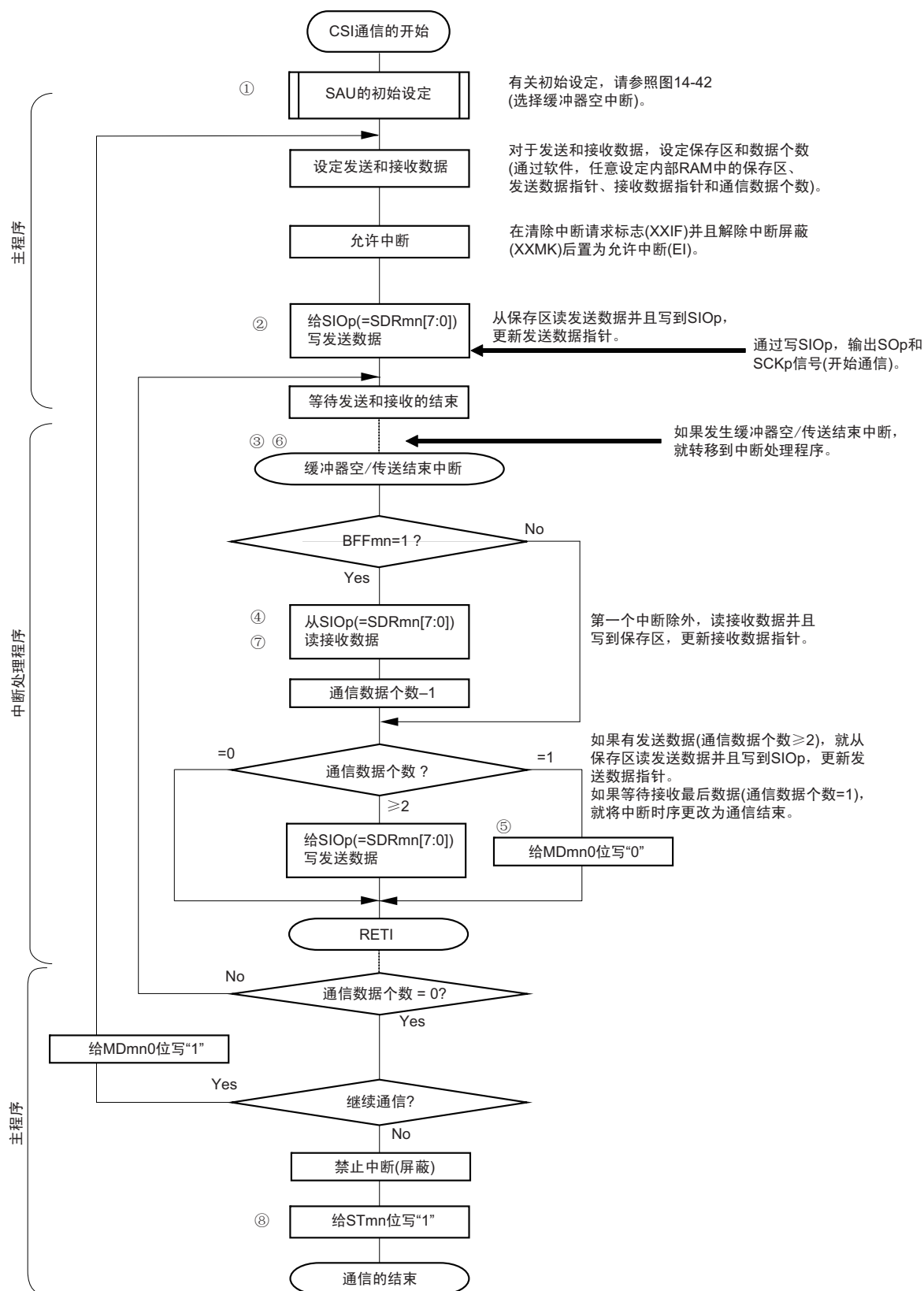


- 注 1. 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。
2. 如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。
- 注意 即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。

备注 1. 图中的①～⑧对应“图 14-48 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、01、10、11、20、21）
mn=00～03、10、11

图 14-48 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 14-47 主控发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

14.5.4 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	CSI01	CSI10	CSI11	CSI20	CSI21
对象通道	SAU0 的通道 0	SAU0 的通道 1	SAU0 的通道 2	SAU0 的通道 3	SAU1 的通道 0	SAU1 的通道 1
使用的引脚	SCK00、SO00	SCK01、SO01	SCK10、SO10	SCK11、SO11	SCK20、SO20	SCK21、SO21
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	只有溢出错检测标志（OVFmn）。					
传送数据长度	7 位或者 8 位					
传送速率	Max. $f_{MCK}/6[\text{Hz}]$ 注 1、2					
数据相位	能通过 SCRMn 寄存器的 DAPmn 位进行选择。 • 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: 不反相 • 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、SCK01、SCK10、SCK11、SCK20、SCK21 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 $f_{MCK}/6[\text{Hz}]$ 。

2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 31 章 电特性”）的范围内使用。

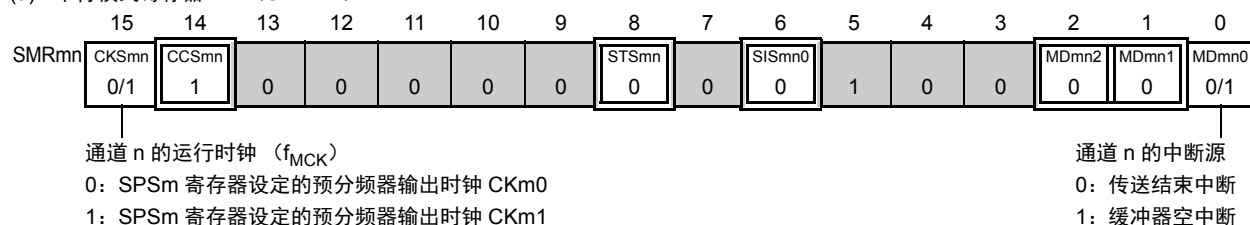
备注 1. f_{MCK} : 对象通道的运行时钟频率

2. m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) mn=00 ~ 03、10、11

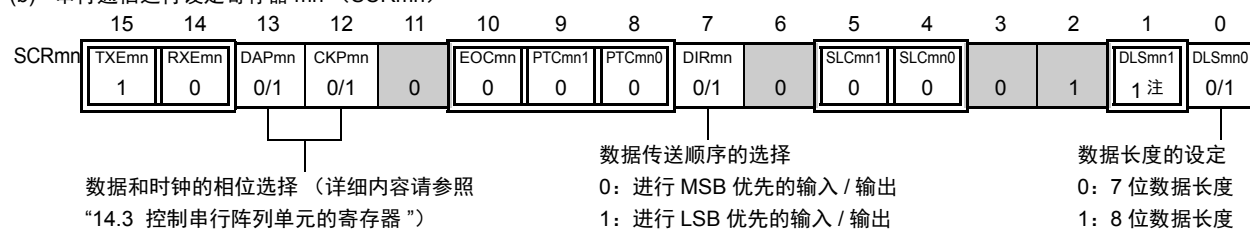
(1) 寄存器的设定

图 14-49 3 线串行 I/O (CSI00、CSI01、CSI10、CSI11、CSI20、CSI21)
从属发送时的寄存器设定内容例子

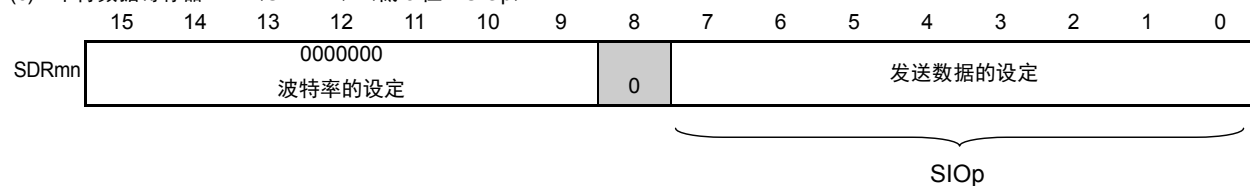
(a) 串行模式寄存器 mn (SMRmn)



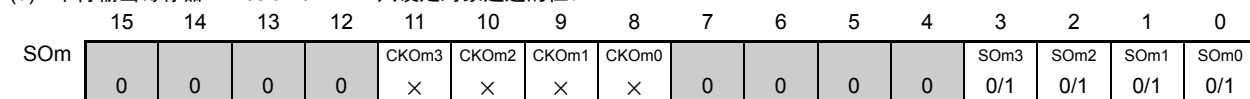
(b) 串行通信运行设定寄存器 mn (SCRmn)



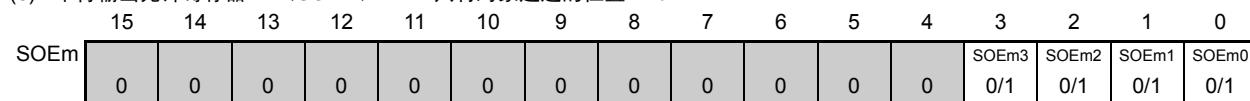
(c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



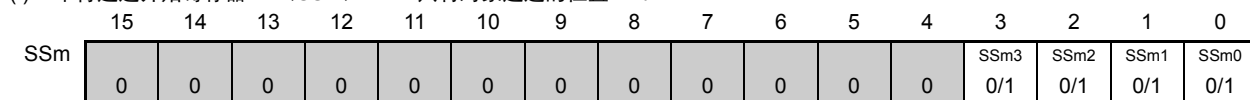
(d) 串行输出寄存器 m (SOMm) 只设定对象通道的位。



(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、01、10、11、20、21)
mn=00~03、10、11

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 14-50 从属发送的初始设定步骤

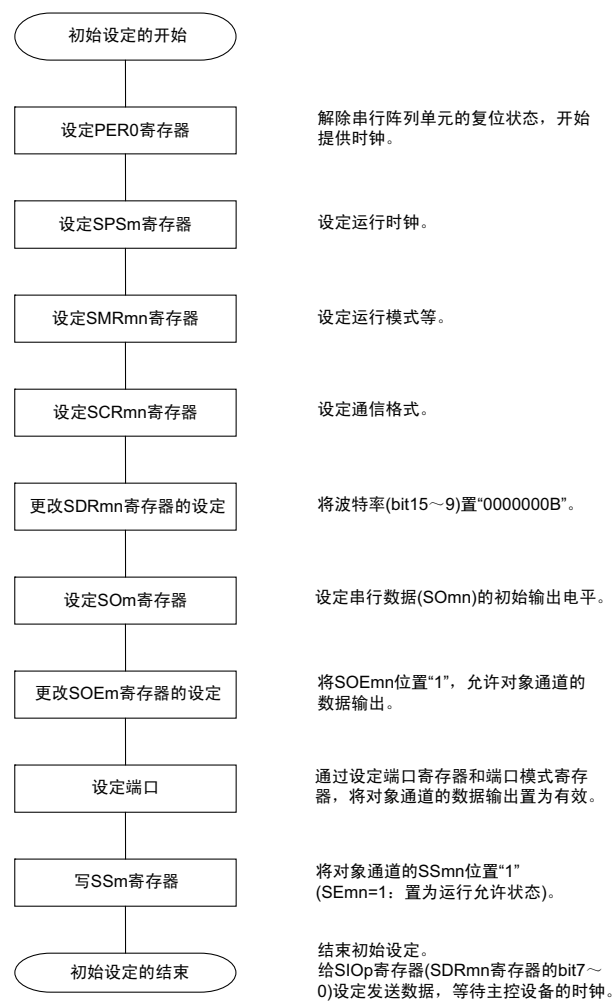


图 14-51 从属发送的中止步骤

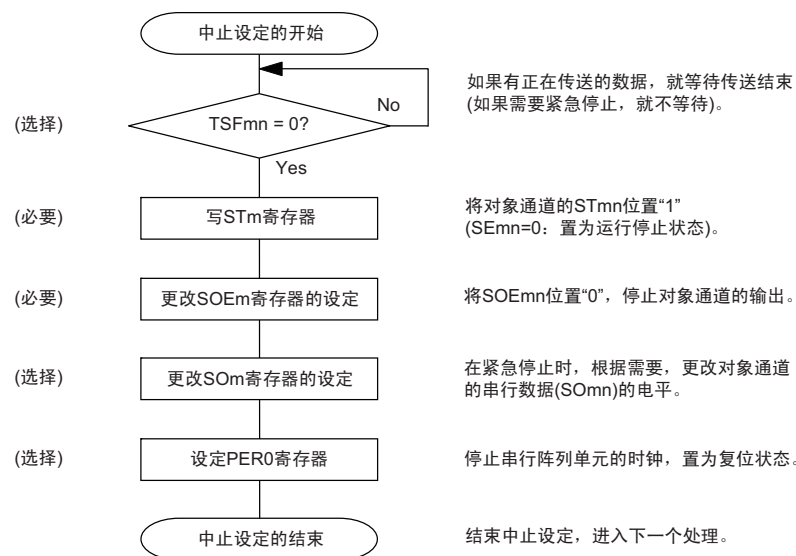
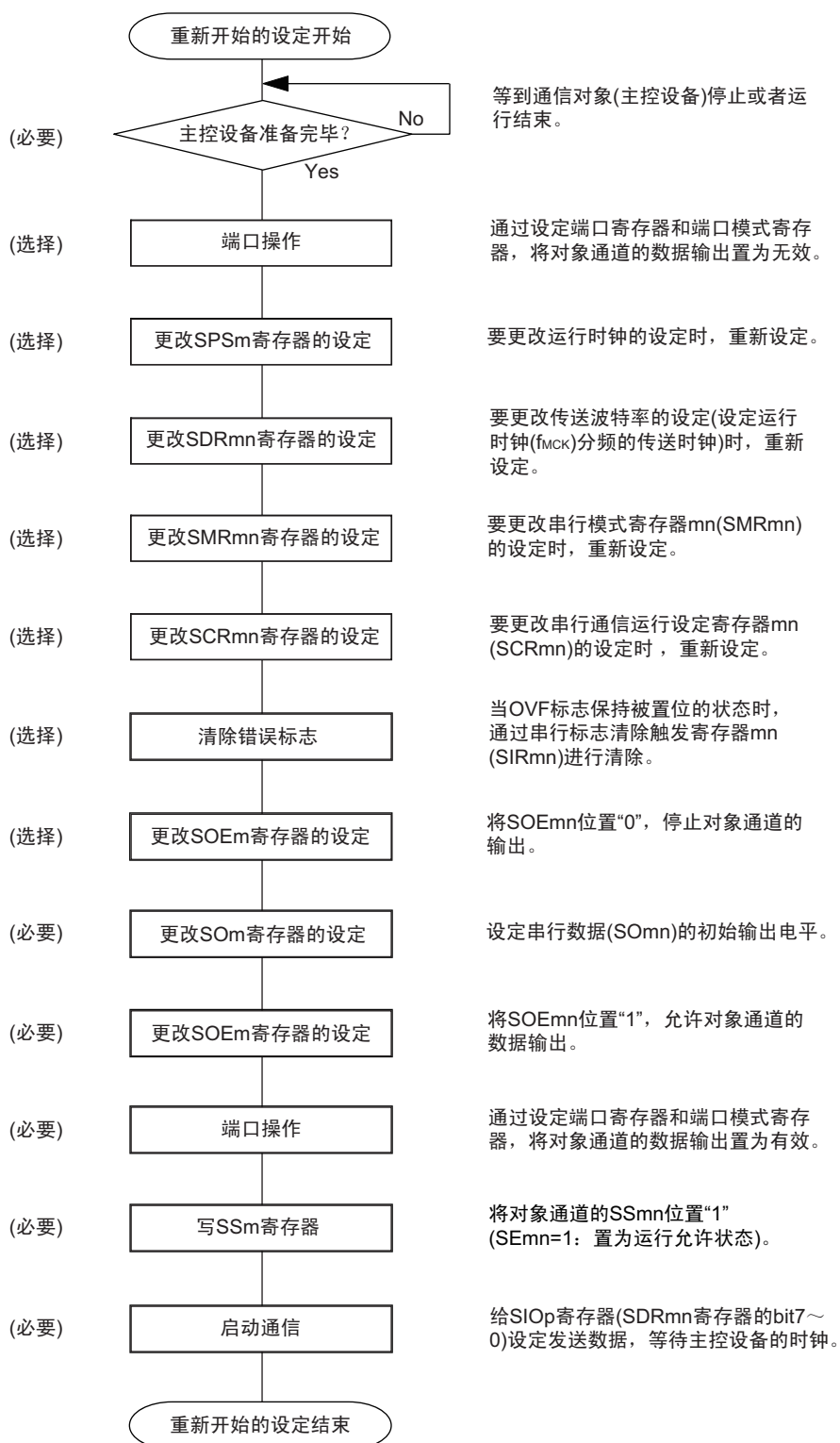


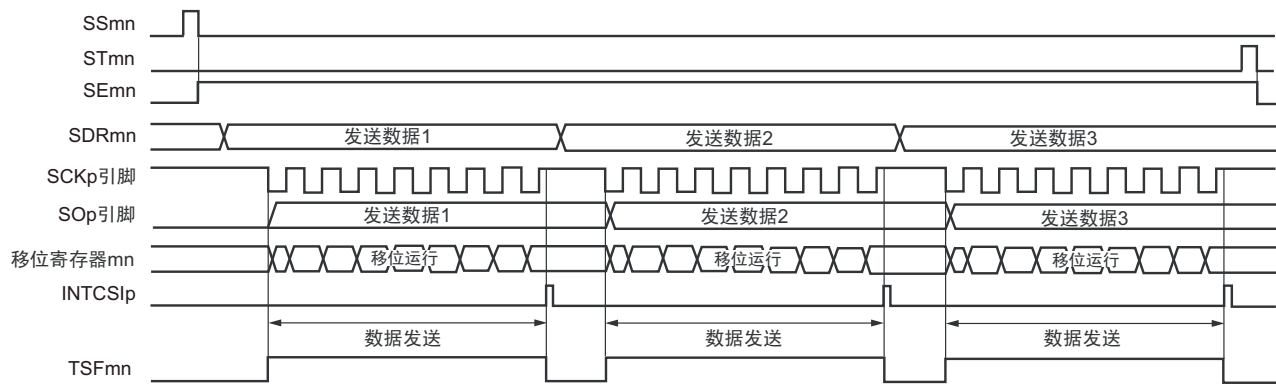
图 14-52 重新开始从属发送的设定步骤



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。

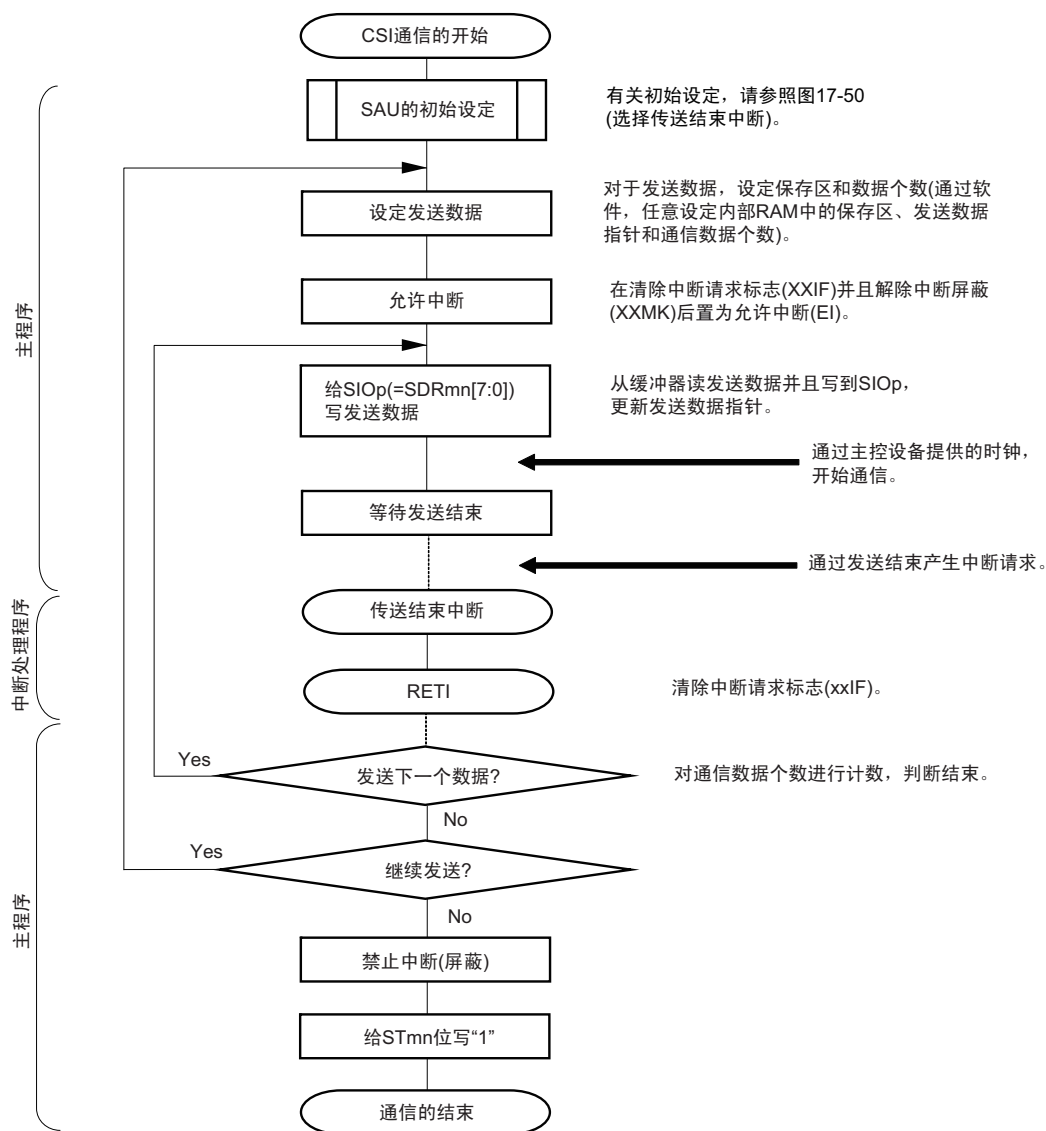
(3) 处理流程（单次发送模式）

图 14-53 从属发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



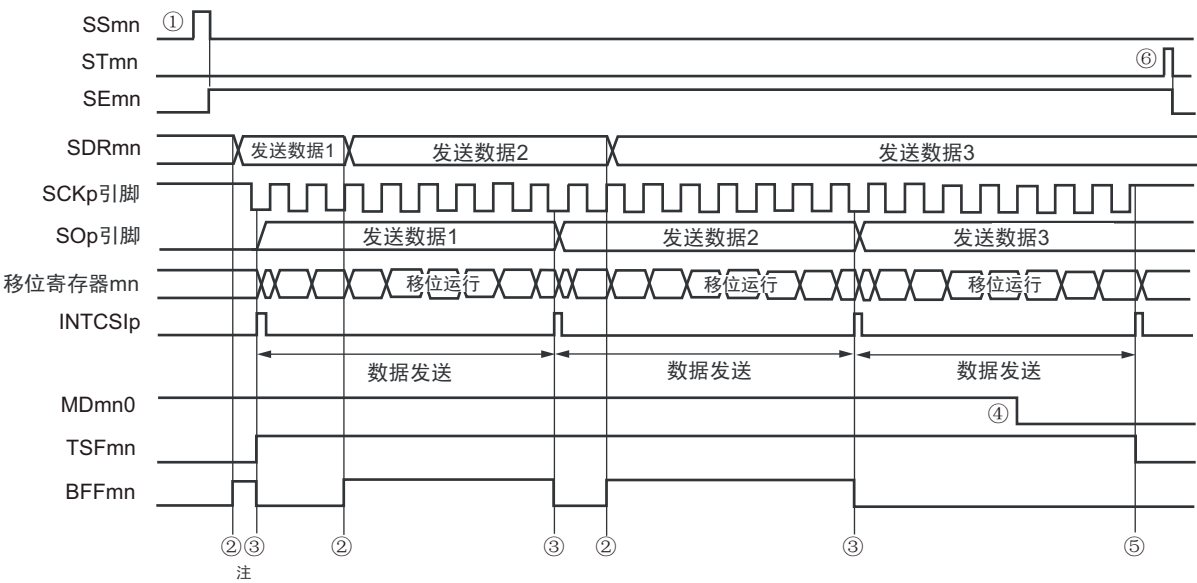
备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) p: CSI 号 (p=00、01、10、11、20、21)
mn=00 ~ 03、10、11

图 14-54 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 14-55 从属发送（连续发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

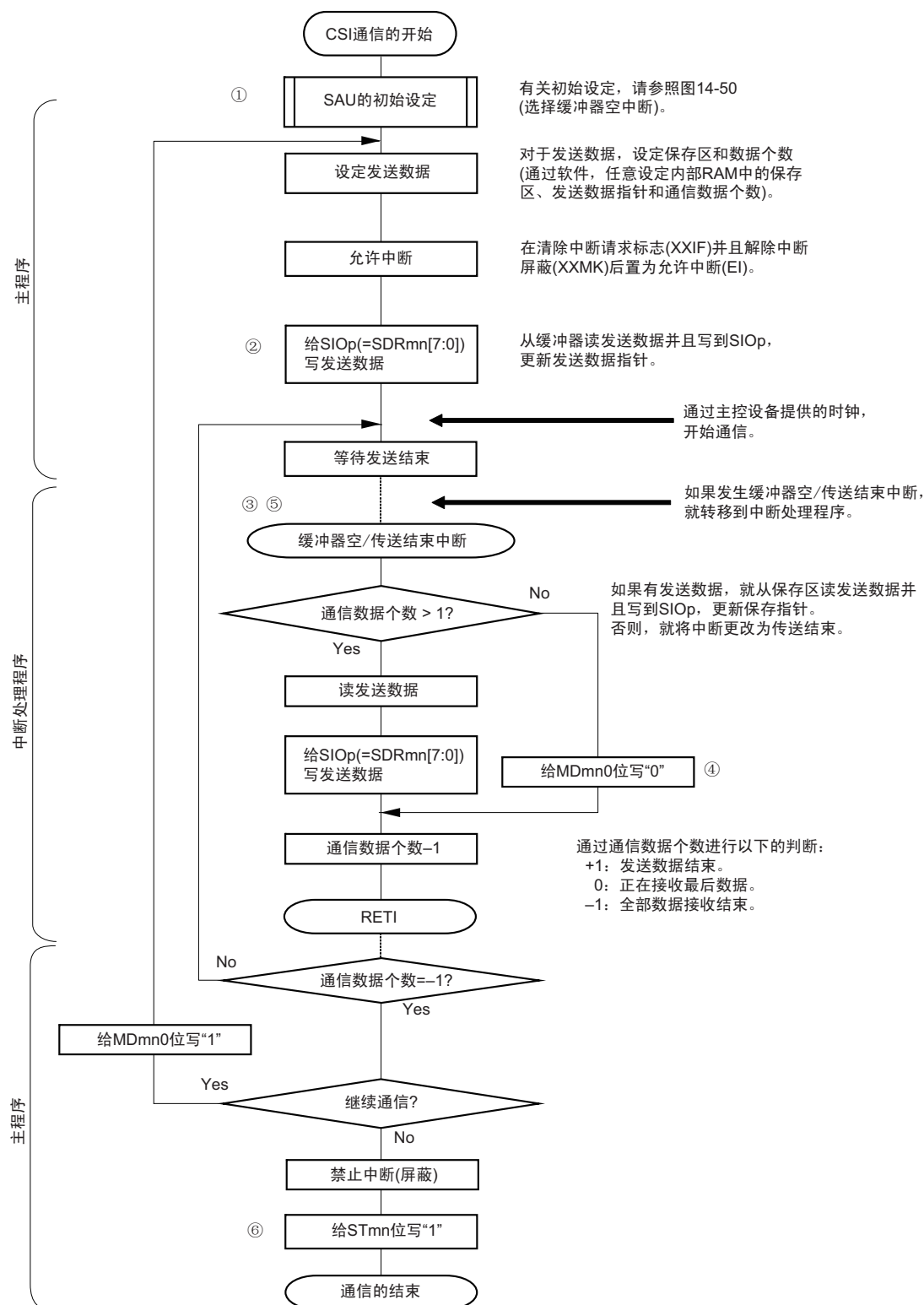


注 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

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、01、10、11、20、21）
mn=00～03、10、11

图 14-56 从属发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图 14-55 从属发送（连续发送模式）的时序图”中的①~⑥。

14.5.5 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	CSI01	CSI10	CSI11	CSI20	CSI21
对象通道	SAU0 的通道 0	SAU0 的通道 1	SAU0 的通道 2	SAU0 的通道 3	SAU1 的通道 0	SAU1 的通道 1
使用的引脚	SCK00、SI00	SCK01、SI01	SCK10、SI10	SCK11、SI11	SCK20、SI20	SCK21、SI21
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	只限于传送结束中断（禁止设定缓冲器空中断）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7 位或者 8 位					
传送速率	Max. $f_{MCK}/6[\text{Hz}]$ 注 1、2					
数据相位	能通过 SCRMn 寄存器的 DAPmn 位进行选择。 • 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: 不反相 • 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、SCK01、SCK10、SCK11、SCK20、SCK21 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 $f_{MCK}/6[\text{Hz}]$ 。

2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 31 章 电特性”）的范围内使用。

备注 1. f_{MCK} : 对象通道的运行时钟频率

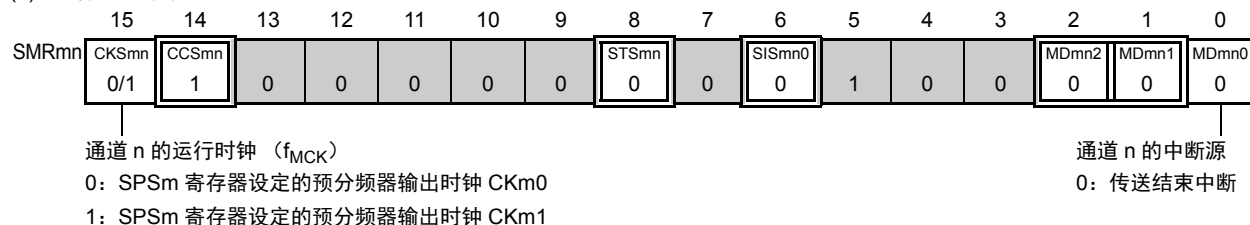
2. m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) mn=00 ~ 03、10、11

(1) 寄存器的设定

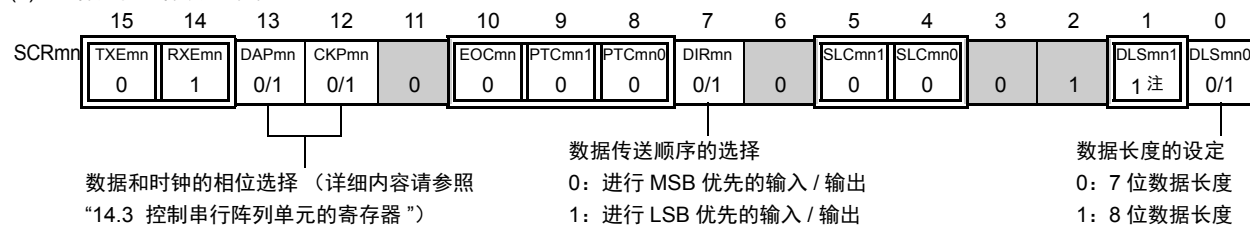
图 14-57 3 线串行 I/O (CSI00、CSI01、CSI10、CSI11、CSI20、CSI21)

从属接收时的寄存器设定内容例子

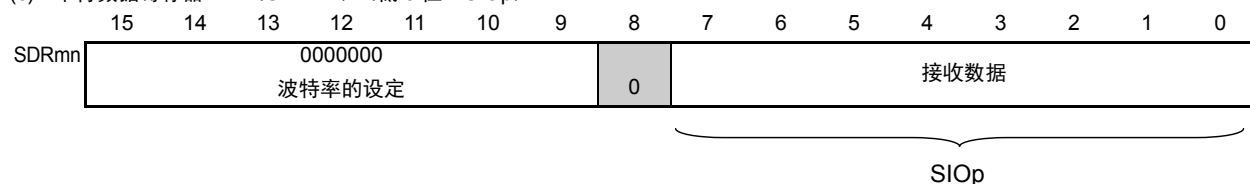
(a) 串行模式寄存器 mn (SMRmn)



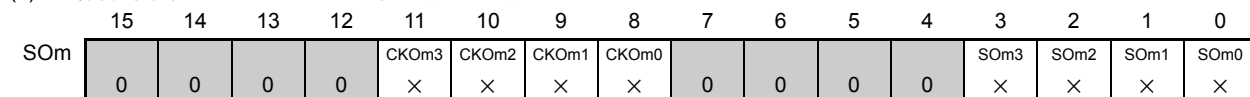
(b) 串行通信运行设定寄存器 mn (SCRmn)



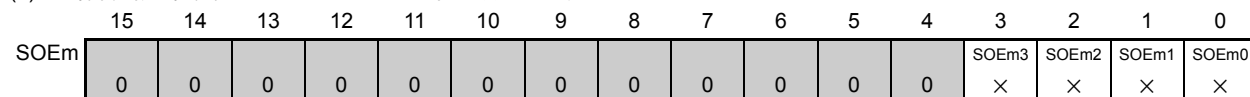
(c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



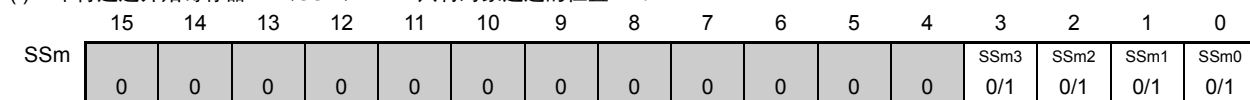
(d) 串行输出寄存器 m (SOM) 在此模式中不使用。



(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、01、10、11、20、21)
 mn=00~03、10、11

2. : 在从属接收模式中为固定设定。 : 不能设定 (设定初始值)。
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 14-58 从属接收的初始设定步骤

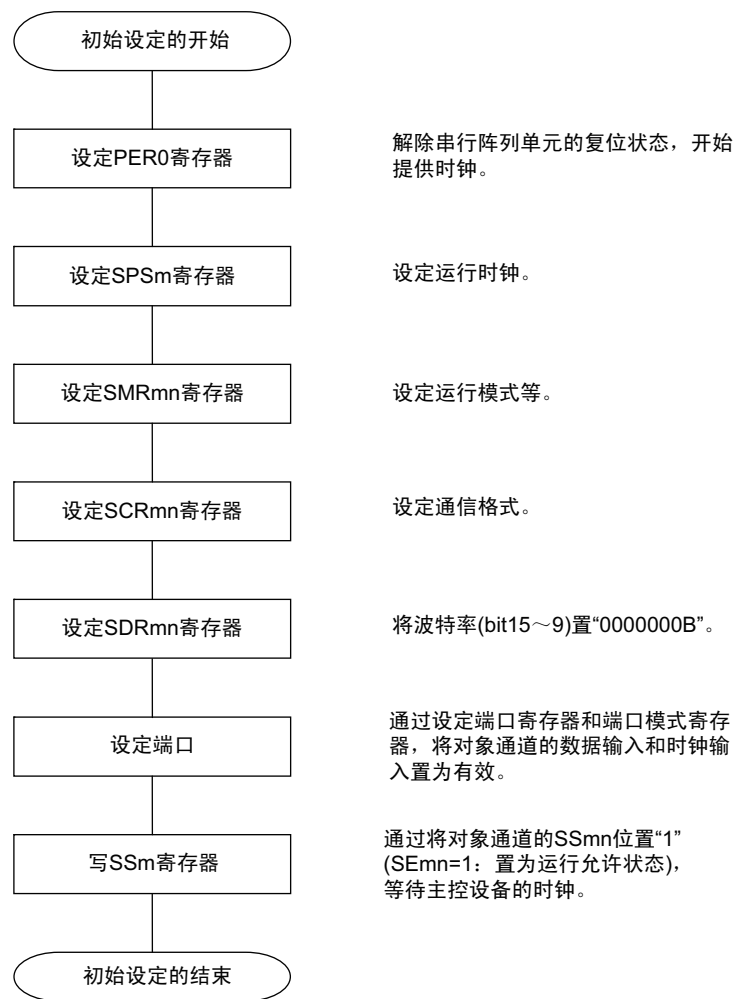


图 14-59 从属接收的中止步骤

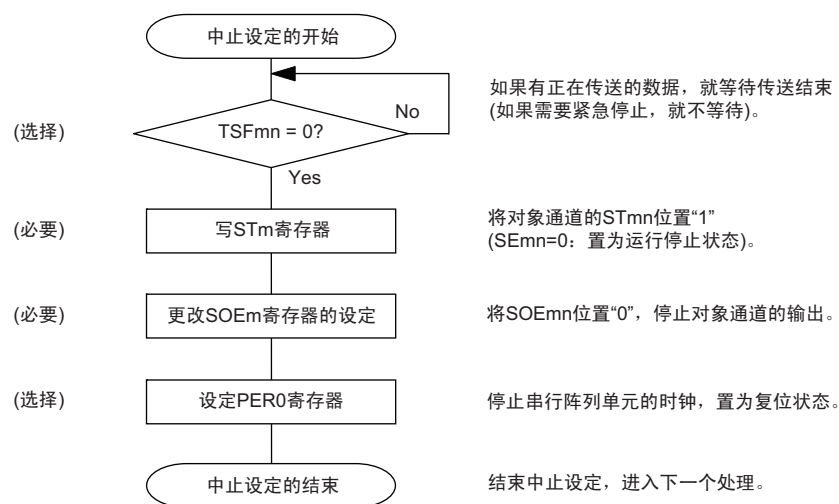
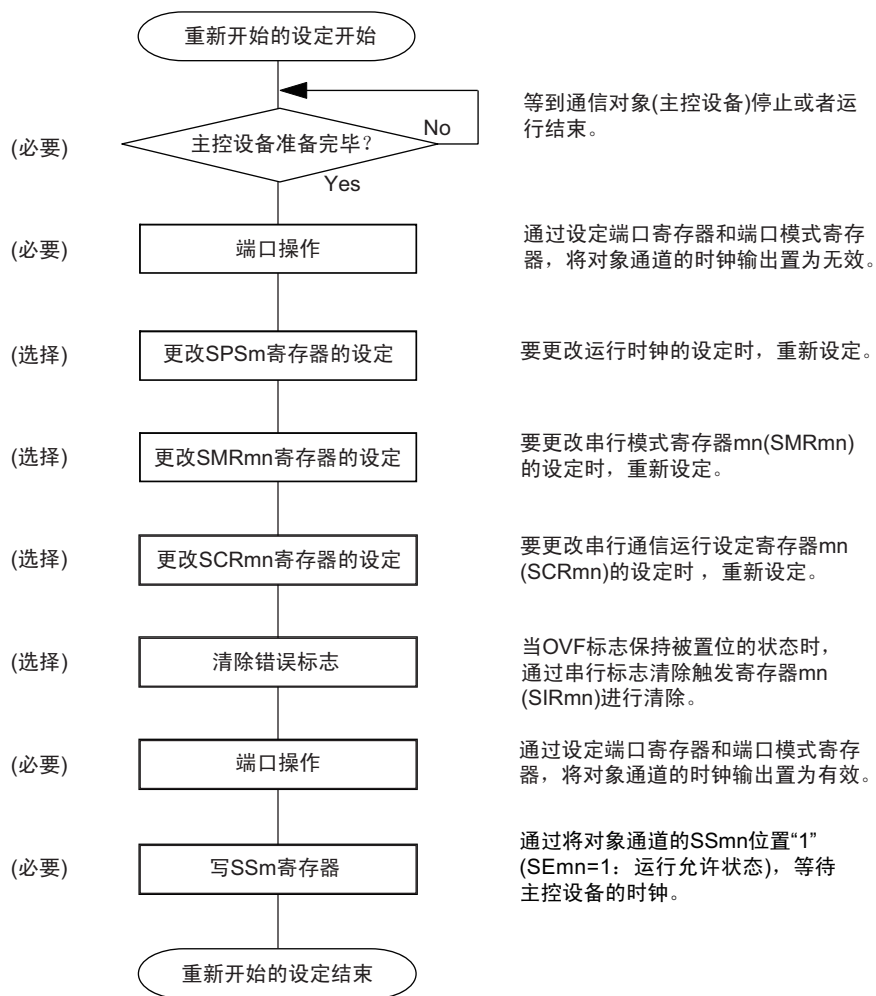


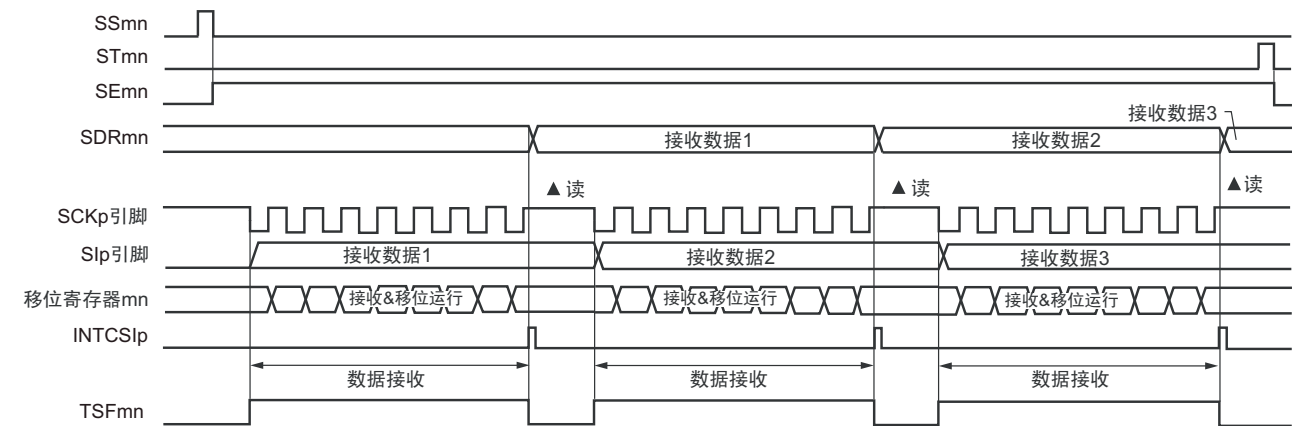
图 14-60 重新开始从属接收的设定步骤



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。

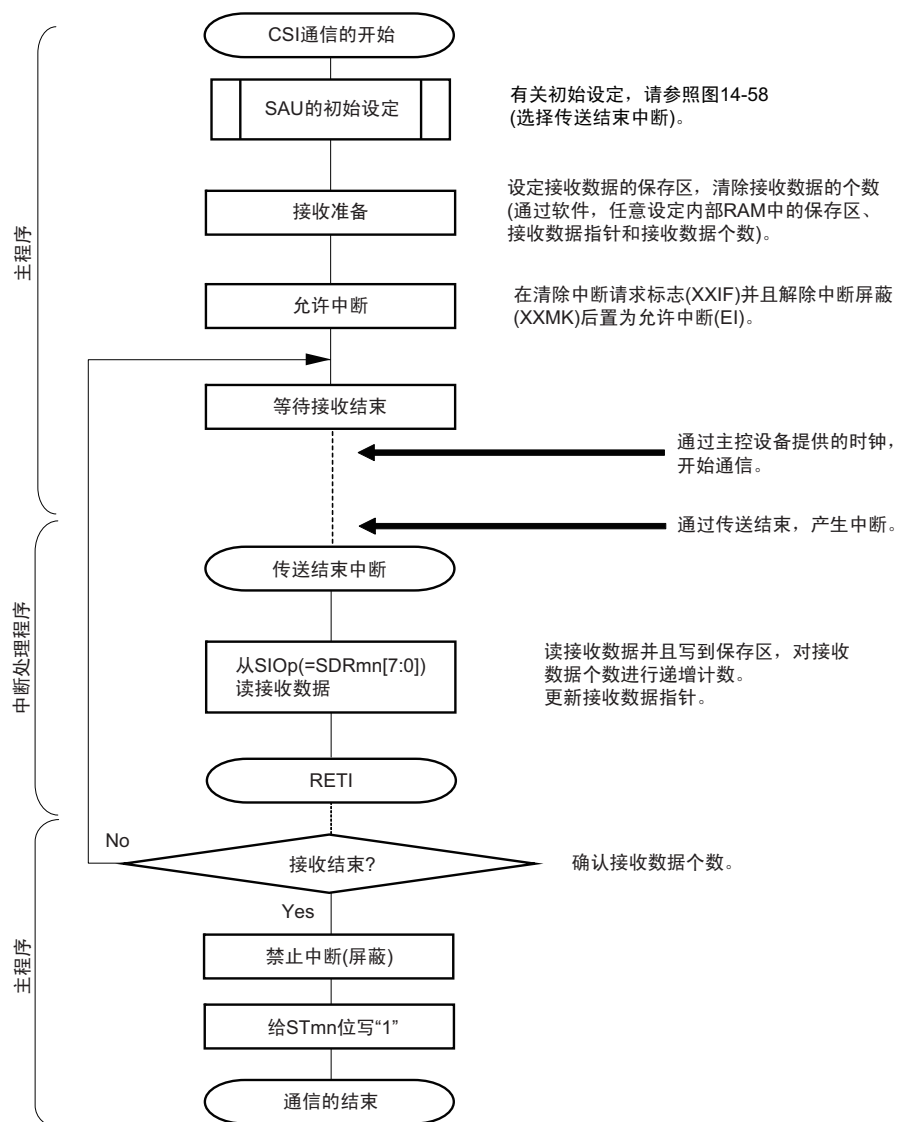
(3) 处理流程（单次接收模式）

图 14-61 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、01、10、11、20、21)
mn=00 ~ 03、10、11

图 14-62 从属接收（单次接收模式）的流程图



14.5.6 从属发送和接收

从属发送和接收是指在从其他设备输入传送时钟的状态下 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	CSI01	CSI10	CSI11	CSI20	CSI21
对象通道	SAU0 的通道 0	SAU0 的通道 1	SAU0 的通道 2	SAU0 的通道 3	SAU1 的通道 0	SAU1 的通道 1
使用的引脚	SCK00、SI00、 SO00	SCK01、SI01、 SO01	SCK10、SI10、 SO10	SCK11、SI11、 SO11	SCK20、SI20、 SO20	SCK21、SI21、 SO21
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7 位或者 8 位					
传送速率	Max. $f_{MCK}/6[\text{Hz}]$ 注 1、2					
数据相位	能通过 SCRMn 寄存器的 DAPmn 位进行选择。 • DAPmn=0: 在串行时钟开始运行时，开始数据输入 / 输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输入 / 输出。					
时钟相位	能通过 SCRMn 寄存器的 CKPmn 位进行选择。 • CKPmn=0: 不反相 • CKPmn=1: 反相					
数据方向	MSB 优先或者 LSB 优先					

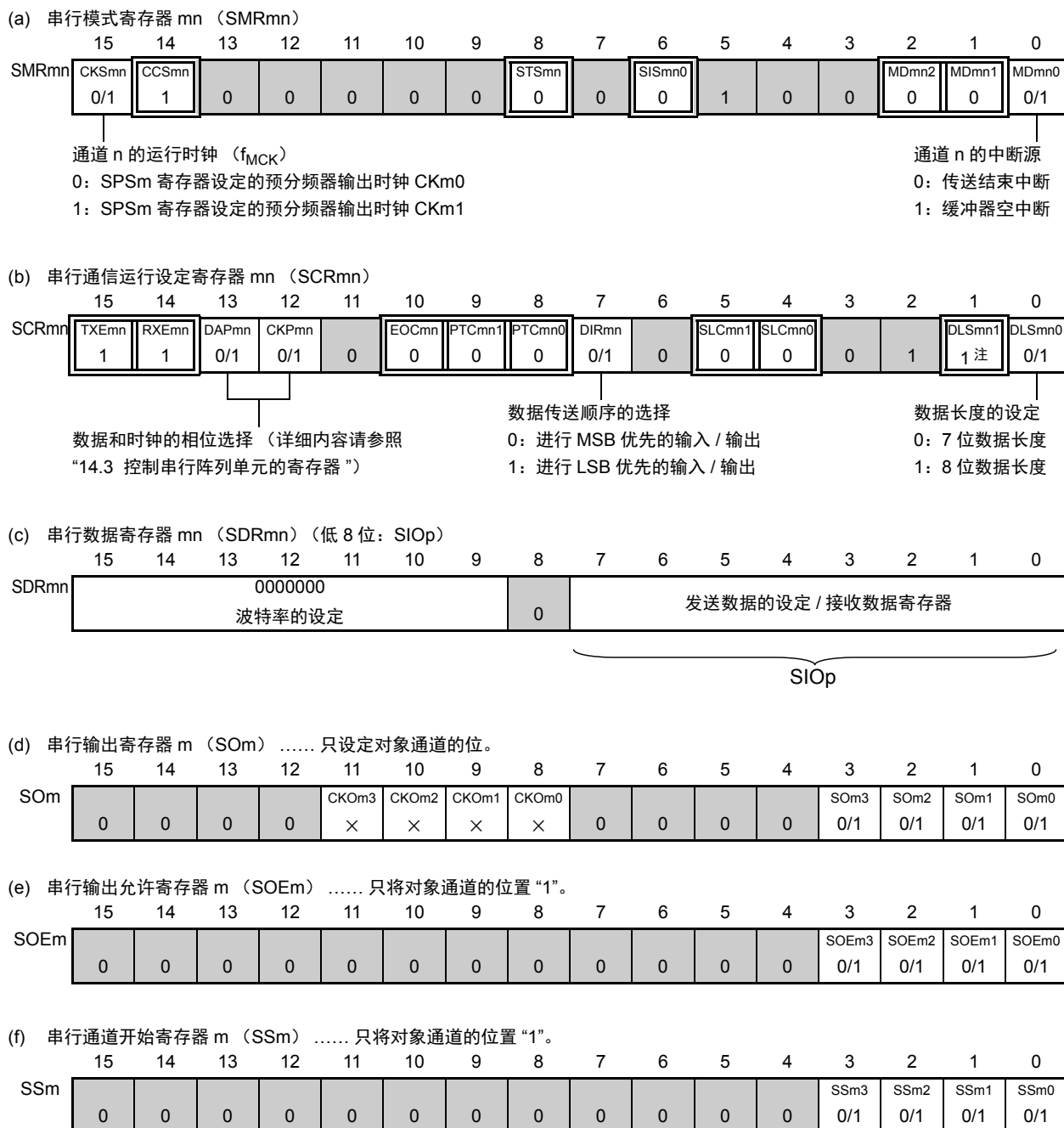
- 注 1. 因为在内部对 SCK00、SCK01、SCK10、SCK11、SCK20、SCK21 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 $f_{MCK}/6[\text{Hz}]$ 。
2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 31 章 电特性”）的范围内使用。

备注 1. f_{MCK} : 对象通道的运行时钟频率

2. m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) mn=00 ~ 03、10、11

(1) 寄存器的设定

图 14-63 3 线串行 I/O (CSI00、CSI01、CSI10、CSI11、CSI20、CSI21)
从属发送和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”。

注意 在主机设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、01、10、11、20、21)

mn=00 ~ 03、 10、 11

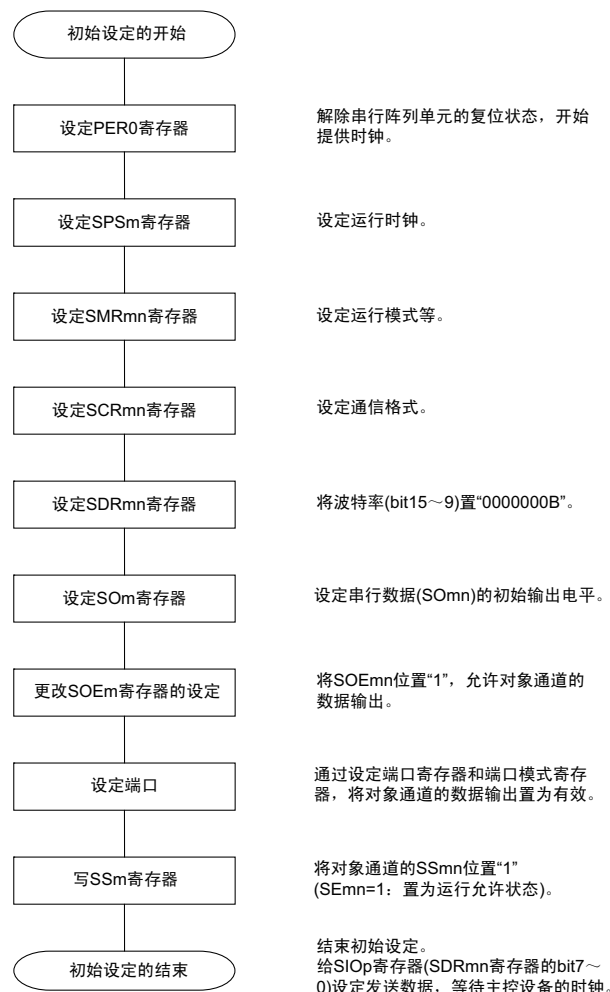
2. ：在 CSI 从属发送和接收模式中为固定设定。：不能设定（设定初始值）。

× : 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 14-64 从属发送和接收的初始设定步骤



注意 在主控设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

图 14-65 从属发送和接收的中止步骤

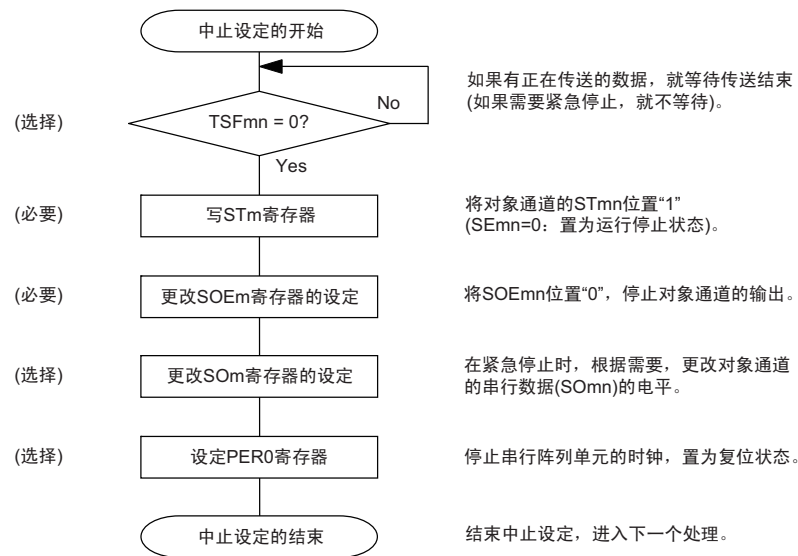
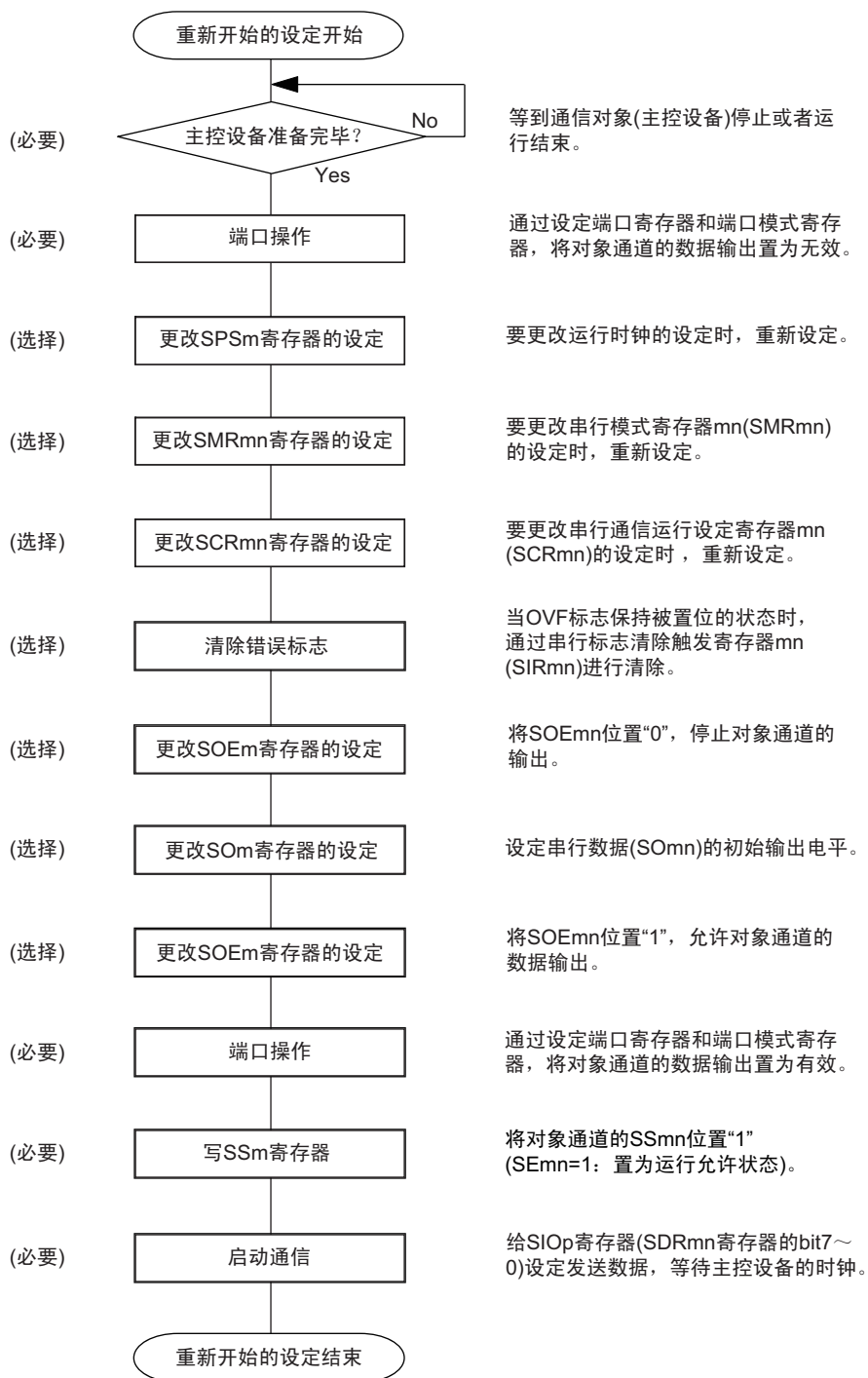


图 14-66 重新开始从属发送和接收的设定步骤

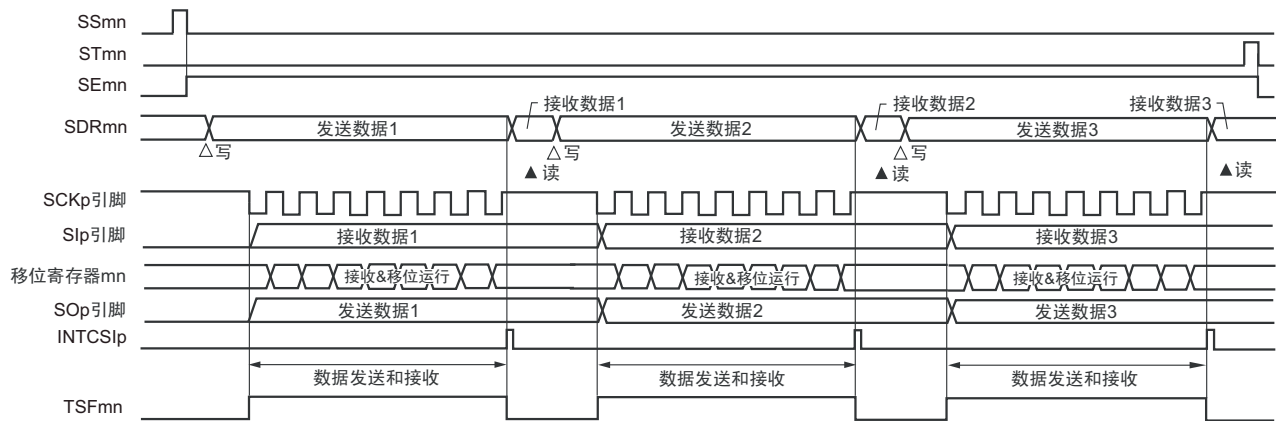


注意 1. 在主控设备开始输出时钟前, 必须给 SIOp 寄存器设定发送数据。

2. 如果在中止设定中改写 PER0 来停止提供时钟, 就必须在等到通信对象 (主控设备) 停止或者通信结束后进行初始设定而不是进行重新开始的设定。

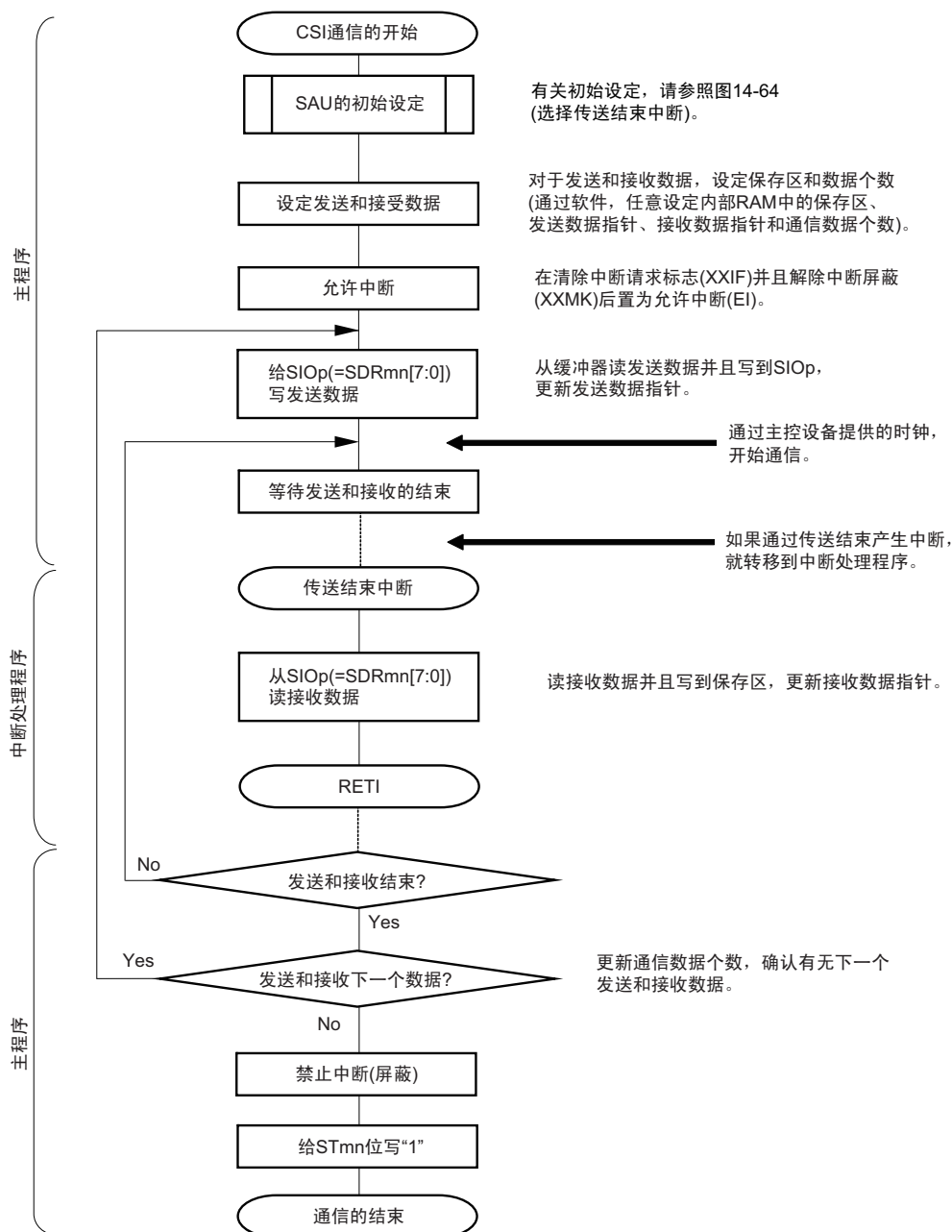
(3) 处理流程（单次发送和接收模式）

图 14-67 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、01、10、11、20、21)
mn=00 ~ 03、10、11

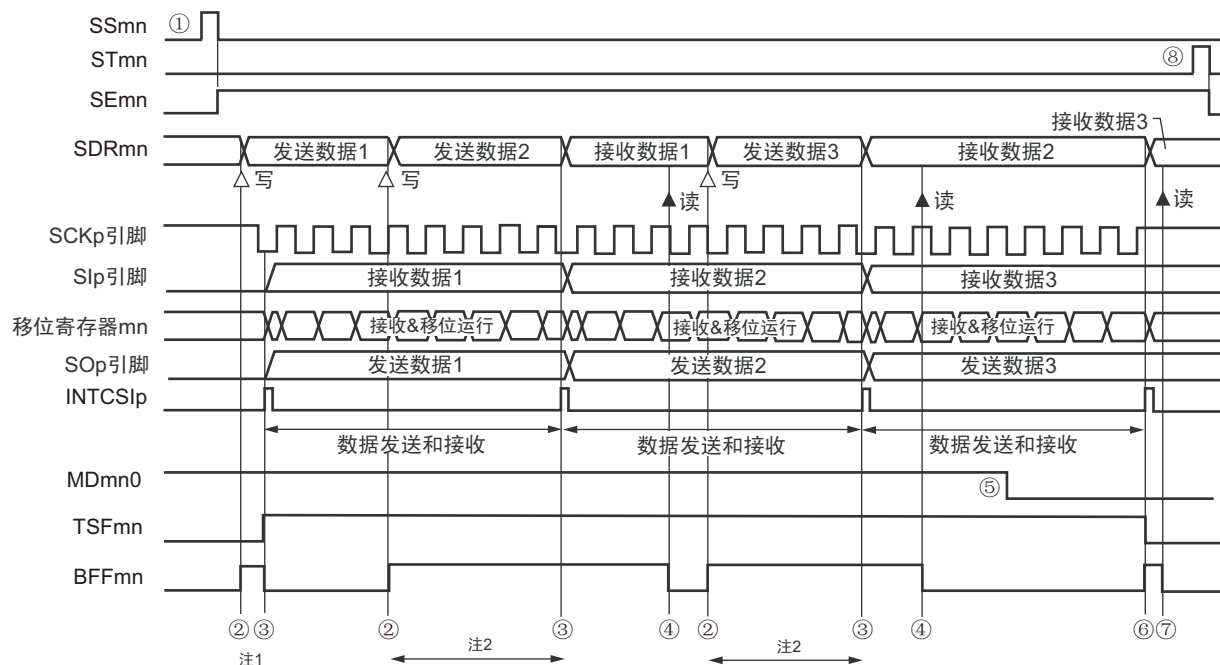
图 14-68 从属发送和接收（单次发送和接收模式）的流程图



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 14-69 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）



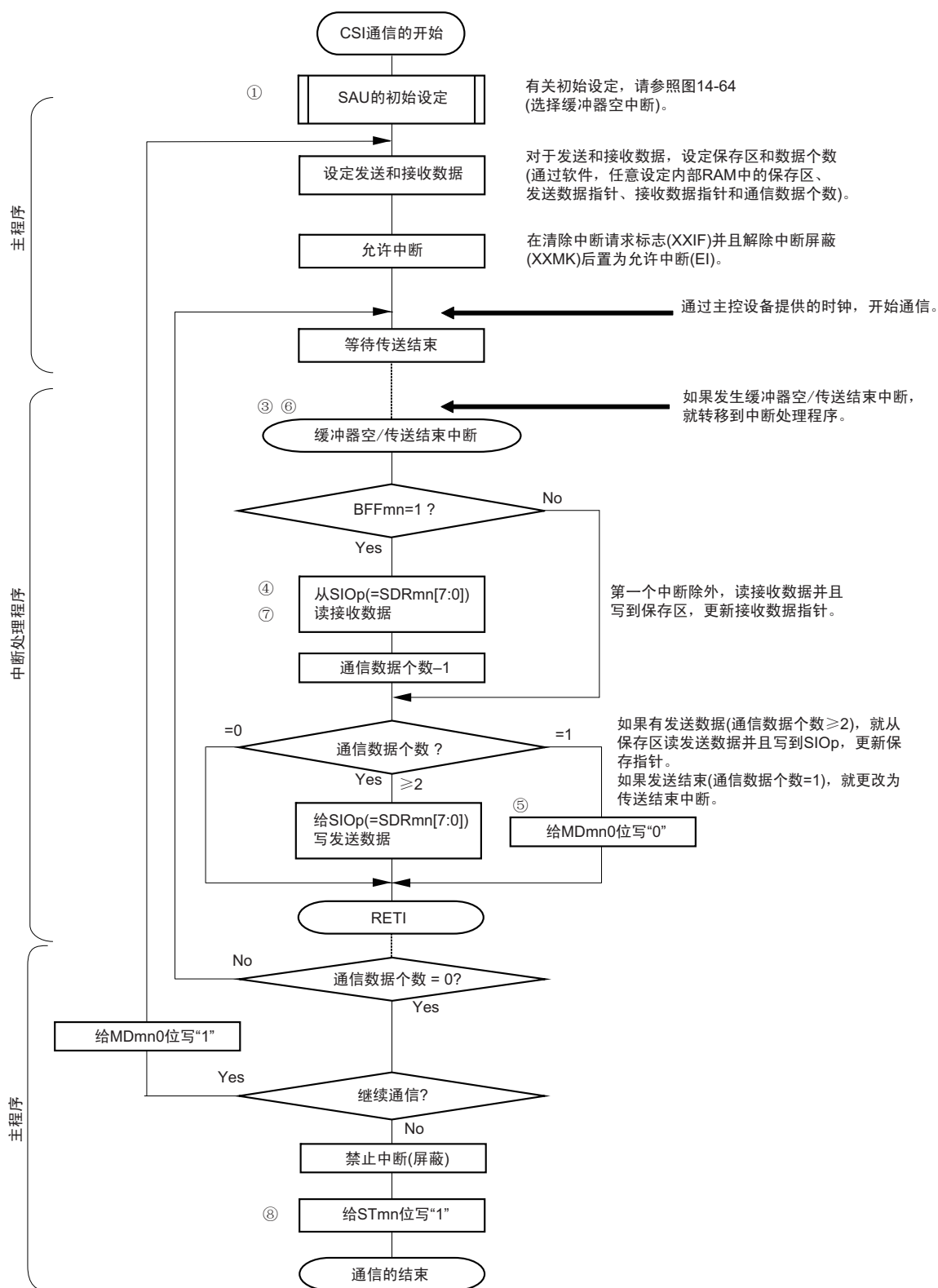
- 注 1. 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。
2. 如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。

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 14-70 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、01、10、11、20、21)
mn=00～03、10、11

图 14-70 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 SI0p 寄存器设定发送数据。

备注 图中的①~⑧对应“图 14-69 从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

14.5.7 SNOOZE 模式功能

这是在 STOP 模式中通过检测 SCKp 引脚的输入使 CSI 进行接收的模式。通常，在 STOP 模式中 CSI 停止通信，但是使用此模式并且通过检测 SCKp 引脚的输入，能在 CPU 不运行的状态下进行 CSI 的接收。只有以下的通道才能设定 SNOOZE 模式。

- 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 14-72 和图 14-74 SNOOZE 模式运行的流程图”）。

- 必须在即将转移到 STOP 模式前将串行待机控制寄存器 m（SSCm）的 SWCm 位置“1”。在初始设定结束后将串行通道开始寄存器 m（SSm）的 SSm1 位置“1”。

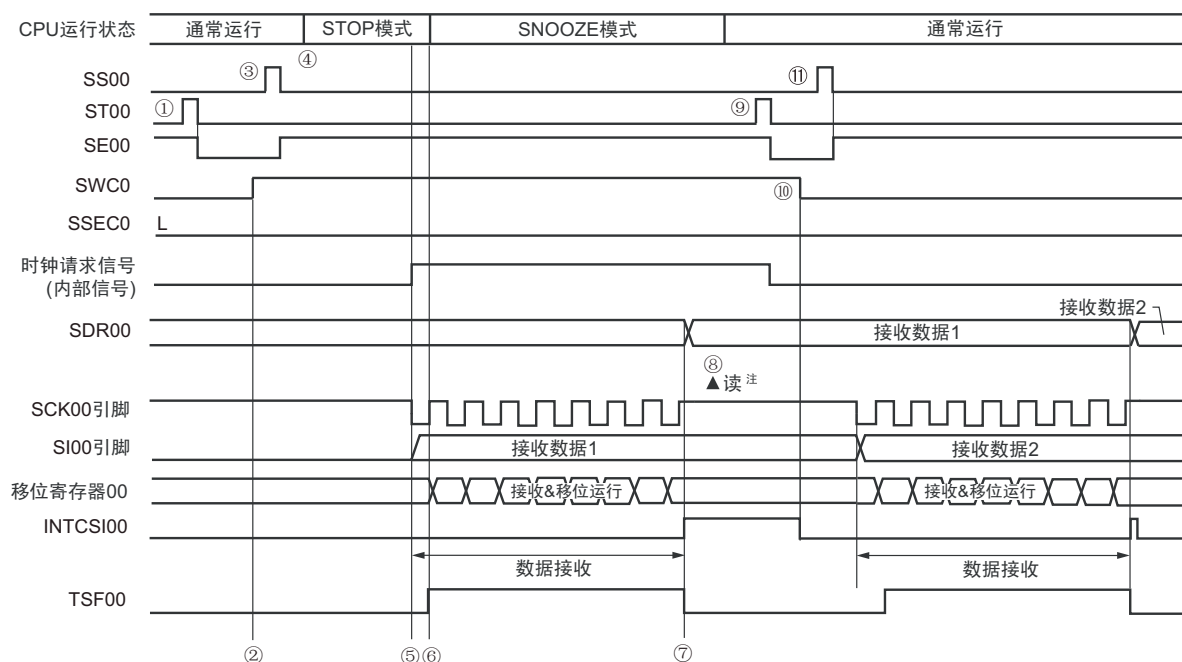
在转移到 STOP 模式后，如果检测到 SCKp 引脚的边沿，就开始 CSI 接收。

注意 1. 只有在选择高速内部振荡器时钟作为 f_{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 14-71 SNOOZE 模式运行（启动一次）的时序图（类型 1: DAPmn=0、CKPmn=0）



注 必须在 SWCm 位为“1”的状态下并且在检测到 SCKp 引脚输入的下一个边沿前读接收数据。

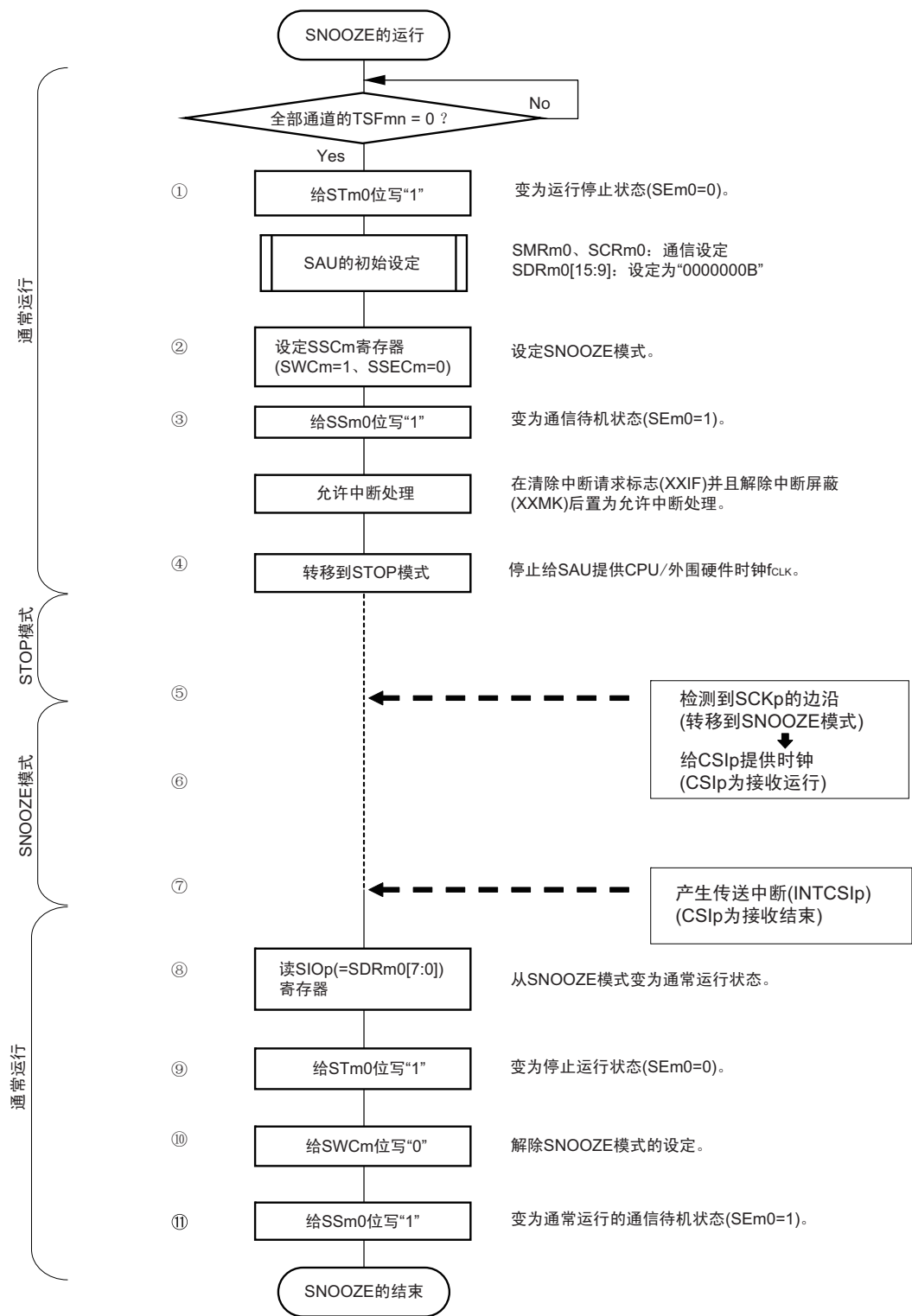
注意 1. 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm0 位置“1”（清除 SEm0 位并且停止运行），而且还必须在接收结束后清除 SWCm 位（解除 SNOOZE 模式）。

2. 当 SWCm 位为“1”时，BFFm1 标志和 OVFm1 标志不变。

备注 1. 图中的①～⑪对应“图 14-72 SNOOZE 模式运行（启动一次）的流程图”中的①～⑪。

2. m=0, p=00

图 14-72 SNOOZE 模式运行（启动一次）的流程图

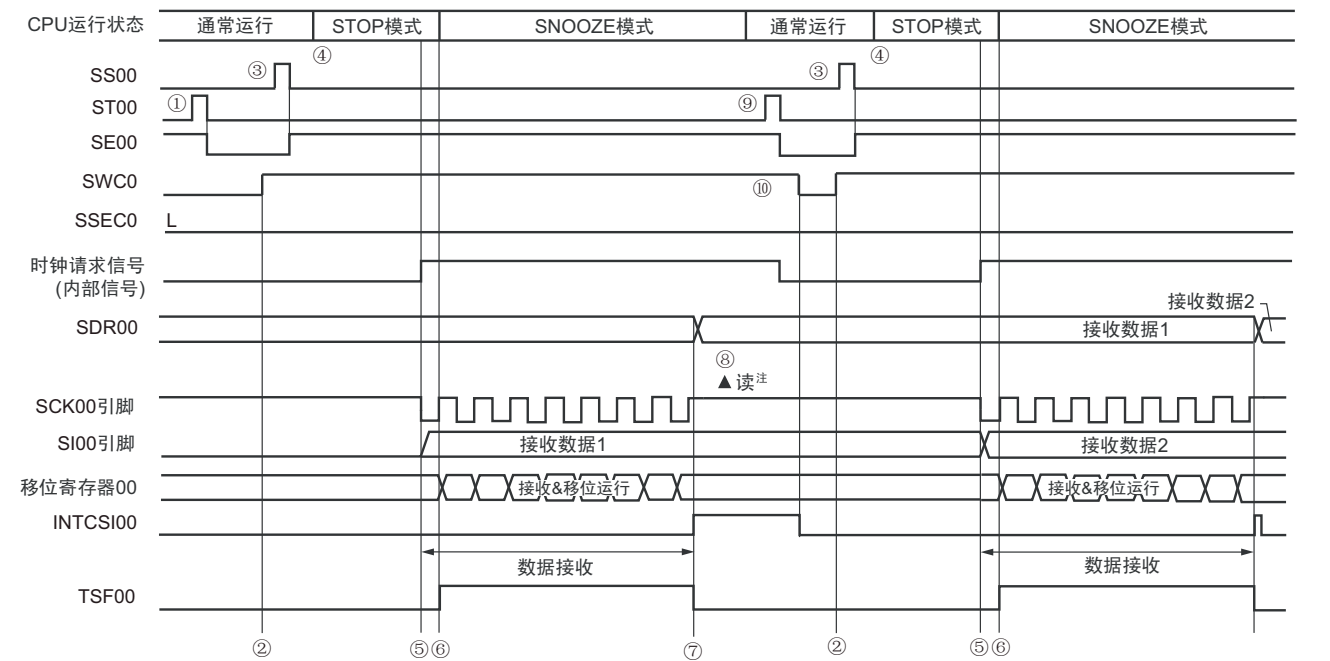


备注 1. 图中的①～⑪对应“图 14-71 SNOOZE 模式运行（启动一次）的时序图”中的①～⑪。

2. m=0, p=00

(2) SNOOZE 模式运行（连续启动）

图 14-73 SNOOZE 模式运行（连续启动）的时序图（类型 1：DAPmn=0、CKPmn=0）



注 必须在 SWCm 位为“1”的状态下并且在检测到 SCKp 引脚输入的下一个边沿前读接收数据。

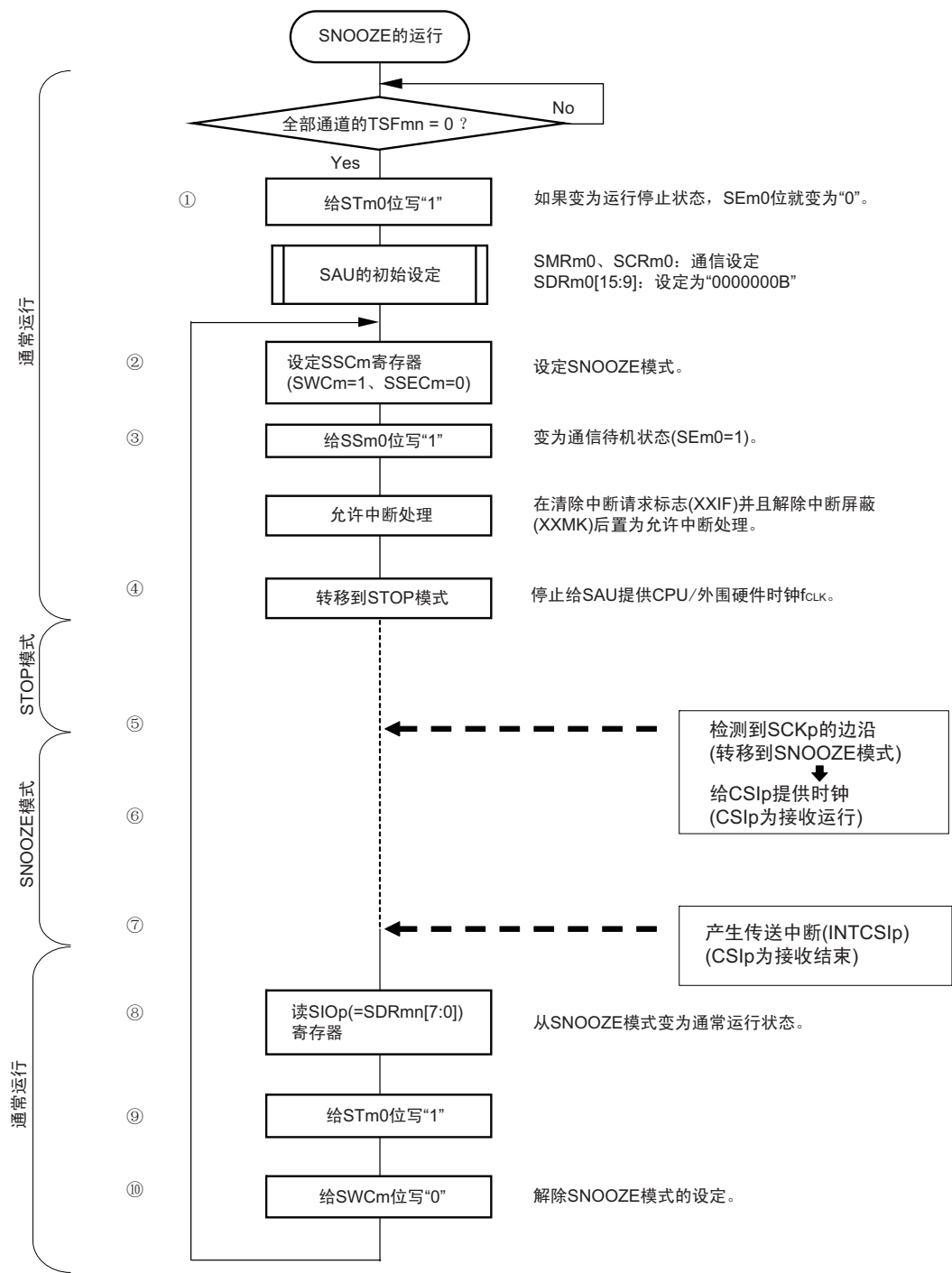
注意 1. 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm0 位置“1”（清除 SEm0 位并且停止运行），而且还必须在接收结束后清除 SWCm 位（解除 SNOOZE 模式）。

2. 当 SWCm 位为“1”时，BFFm1 标志和 OVfm1 标志不变。

备注 1. 图中的①～⑩对应“图 14-74 SNOOZE 模式运行（连续启动）的流程图”中的①～⑩。

2. m=0, p=00

图 14-74 SNOOZE 模式运行（连续启动）的流程图



备注 1. 图中的①~⑩对应“图 14-73 SNOOZE 模式运行（连续启动）的时序图”中的①~⑩。

2. m=0, p=00

14.5.8 传送时钟频率的计算

3线串行I/O（CSI00、CSI01、CSI10、CSI11、CSI20、CSI21）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 14-2 3 线串行 I/O 运行时钟的选择

SMRmn 寄存器	SPSm 寄存器								运行时钟 (f _{MCK}) 注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f _{CLK} =32MHz 运行时
0	X	X	X	X	0	0	0	0	f _{CLK}	32MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	1MHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	500kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	250kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	125kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	62.5kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	31.25kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	15.63kHz
	X	X	X	X	1	1	0	0	f _{CLK} /2 ¹²	7.81kHz
	X	X	X	X	1	1	0	1	f _{CLK} /2 ¹³	3.91kHz
	X	X	X	X	1	1	1	0	f _{CLK} /2 ¹⁴	1.95kHz
	X	X	X	X	1	1	1	1	f _{CLK} /2 ¹⁵	977Hz
1	0	0	0	0	X	X	X	X	f _{CLK}	32MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	16MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	8MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	4MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	2MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	1MHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	500kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	250kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	125kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	62.5kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	31.25kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	15.63kHz
	1	1	0	0	X	X	X	X	f _{CLK} /2 ¹²	7.81kHz
	1	1	0	1	X	X	X	X	f _{CLK} /2 ¹³	3.91kHz
	1	1	1	0	X	X	X	X	f _{CLK} /2 ¹⁴	1.95kHz
	1	1	1	1	X	X	X	X	f _{CLK} /2 ¹⁵	977Hz

注 要更改被选择为 f_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止串行阵列单元（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

14.5.9 在 3 线串行 I/O（CSI00、CSI01、CSI10、CSI11、CSI20、CSI21）通信过程中发生错误时的处理步骤

在 3 线串行 I/O（CSI00、CSI01、CSI10、CSI11、CSI20、CSI21）通信过程中发生错误时的处理步骤如图 14-75 所示。

图 14-75 发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。→	SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn（SSRmn）。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn（SIRmn）写“1”。	清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SIRmn 寄存器，只能清除读操作时的错误。

备注 m: 单元号（m=0、1） n: 通道号（n=0～3） mn=00～03、10、11

14.6 从属选择输入功能的时钟同步串行通信的运行

SAU0 的通道 0 是支持从属选择输入功能的时钟同步串行通信的通道。

[数据的发送和接收]

- 7 位或者 8 位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定

[时钟控制]

- 输入/输出时钟的相位控制
- 设定由预分频器和通道内部计数器产生的传送周期。
- 最大传送速率注

从属通信: $\text{Max. } f_{\text{MCK}}/6$

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

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 31 章 电特性”。

○ 32 引脚产品

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00 (支持从属选择输入功能)	UART0 (支持 LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

○ 64 引脚产品

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00 (支持从属选择输入功能)	UART0 (支持 LIN-bus)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

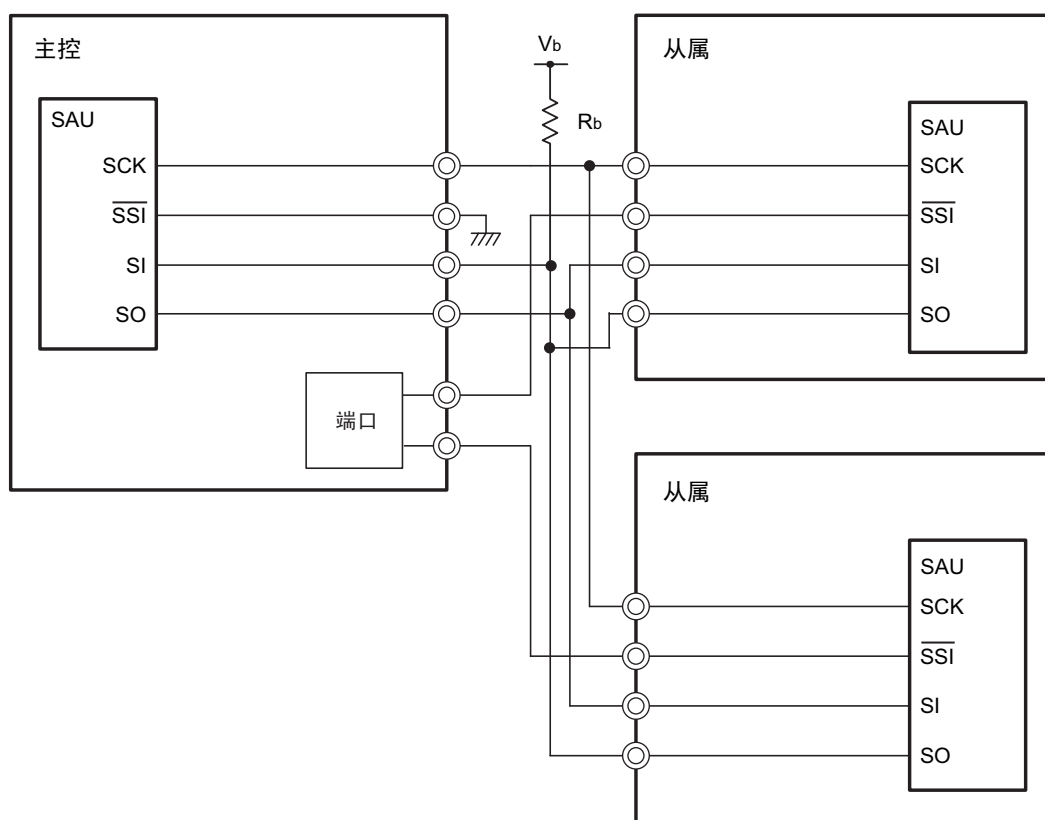
从属选择输入功能有以下 3 种通信运行：

- 从属发送 (参照 14.6.1)
- 从属接收 (参照 14.6.2)
- 从属的发送和接收 (参照 14.6.3)

能通过使用从属选择输入功能，使 1 个主控设备连接多个从属设备进行通信。主控设备对通信对象的从属设备（1 个）进行从属选择信号的输出，各从属设备判断自己是否被选择为通信对象并且控制 SO 引脚的输出。当被选择为通信对象的从属设备时，SO 引脚能对主控设备进行发送数据的通信；当不被选择为通信对象的从属设备时，SO 引脚变为高电平输出，因此在连接多个从属设备的环境下需要将 SO 引脚设定为 Nch-O.D 并且将该节点进行上拉。另外，即使输入主控设备的串行时钟也不进行发送和接收。

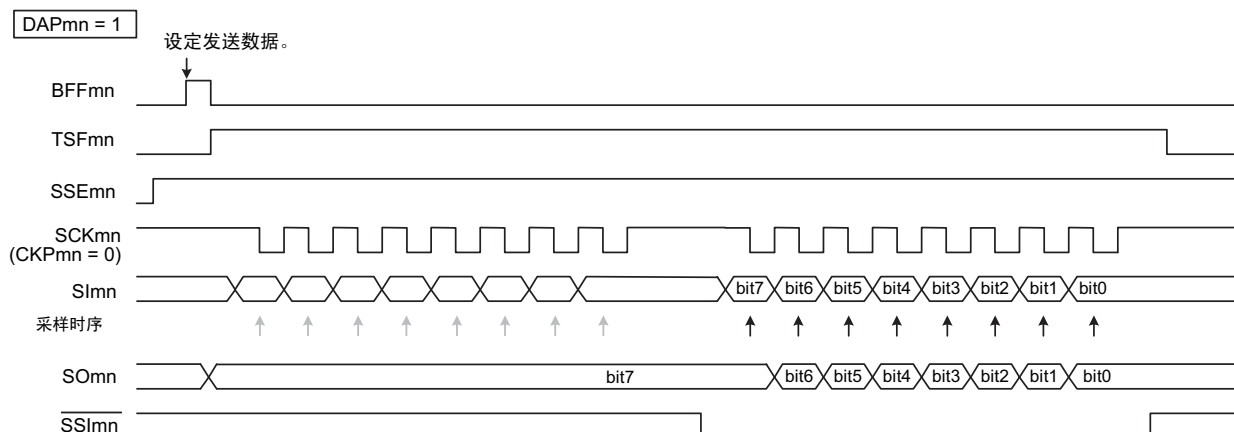
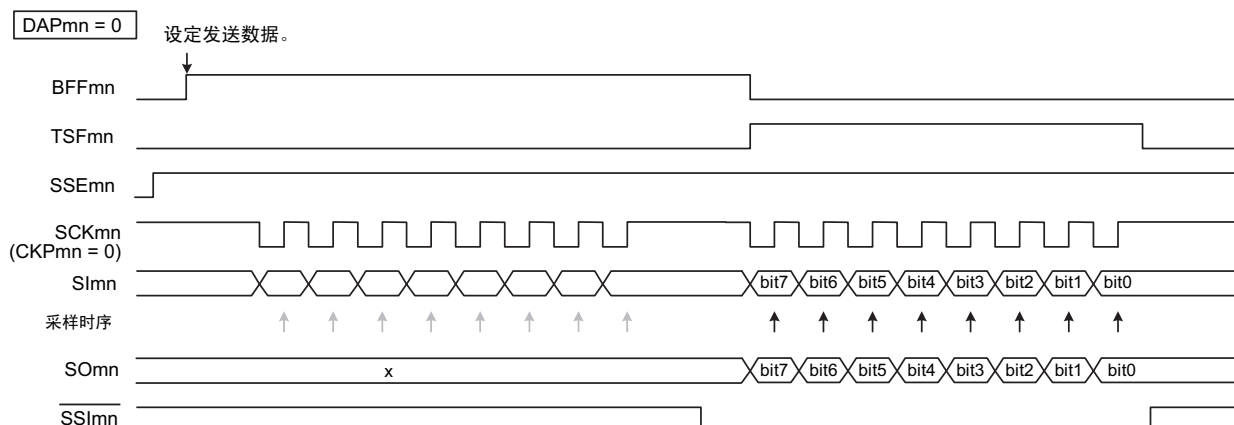
注意 必须通过端口的操作输出从属选择信号。

图 14-76 从属选择输入功能的结构例子



注意 必须在 $EV_{DD0} \geq Vb$ 的情况下使用。
将 SO00 引脚选择为 N 沟道漏极开路输出（ EV_{DD} 耐压）模式。

图 14-77 从属选择输入功能的时序图



备注 m: 单元号 (m=0) n: 通道号 (n=0)

14.6.1 从属发送

从属发送是指在从其他设备输入传送时钟的状态下 RL78 微控制器将数据发送到其他设备的运行。

从属选择输入功能	CSI00
对象通道	SAU0 的通道 0
使用的引脚	SCK00、SO00、SSI00
中断	INTCSI00 可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7 位或者 8 位
传送速率	Max. $f_{MCK}/6[Hz]$ 注 1、2
数据相位	能通过 SCRMn 寄存器的 DAPmn 位进行选择。 - 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: 不反相 - CKPmn=1: 反相
数据方向	MSB 优先或者 LSB 优先
从属选择输入功能	可选择从属选择功能的运行。

注 1. 因为在内部对 SCK00 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 $f_{MCK}/6[Hz]$ 。

2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 31 章 电特性”）的范围内使用。

备注 1. f_{MCK} : 对象通道的运行时钟频率

2. m: 单元号 (m=0) n: 通道号 (n=0)

(1) 寄存器的设定

图 14-78 从属选择输入功能 (CSI00) 从属发送时的寄存器设定内容例子 (1/2)



备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

2. : 在 CSI 从属发送模式中为固定设定。 : 不能设定 (设定初始值)。
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

图 14-78 从属选择输入功能（CSI00）从属发送时的寄存器设定内容例子 (2/2)

(f) 串行通道开始寄存器 m（SSm） 只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

(g) 输入切换控制寄存器（ISC） 这是 CSI00 从属通道（单元 0 的通道 0）的 $\overline{\text{SSI00}}$ 引脚的控制。

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

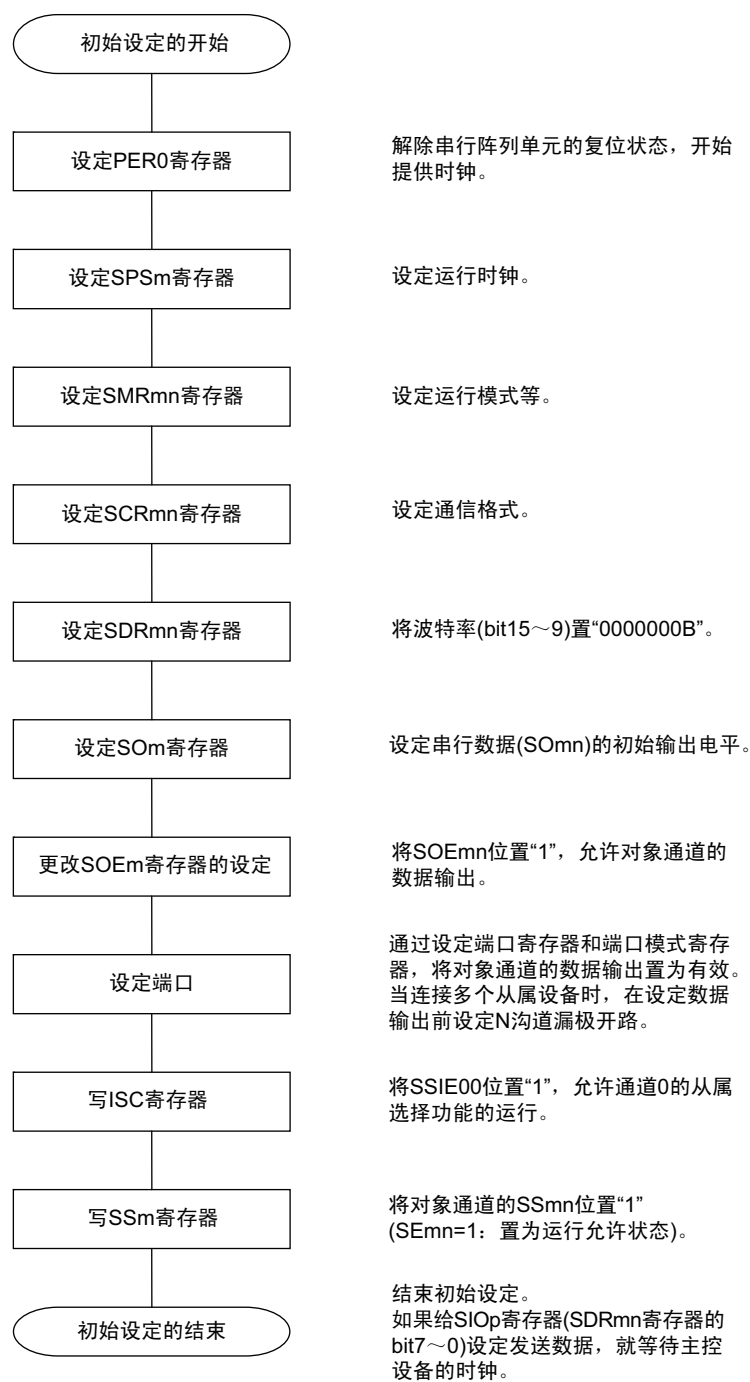
0: $\overline{\text{SSI00}}$ 引脚的输入值无效
1: $\overline{\text{SSI00}}$ 引脚的输入值有效

备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

2. ☐ : 在 CSI 从属发送模式中为固定设定。☐ : 不能设定（设定初始值）。
 × : 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
 0/1 : 根据用户的用途置“0”或者“1”。

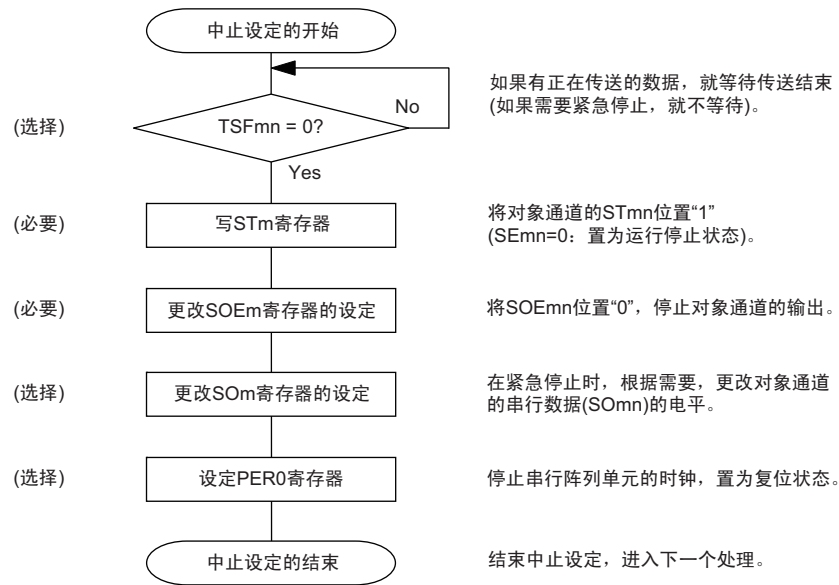
(2) 操作步骤

图 14-79 从属发送的初始设定步骤



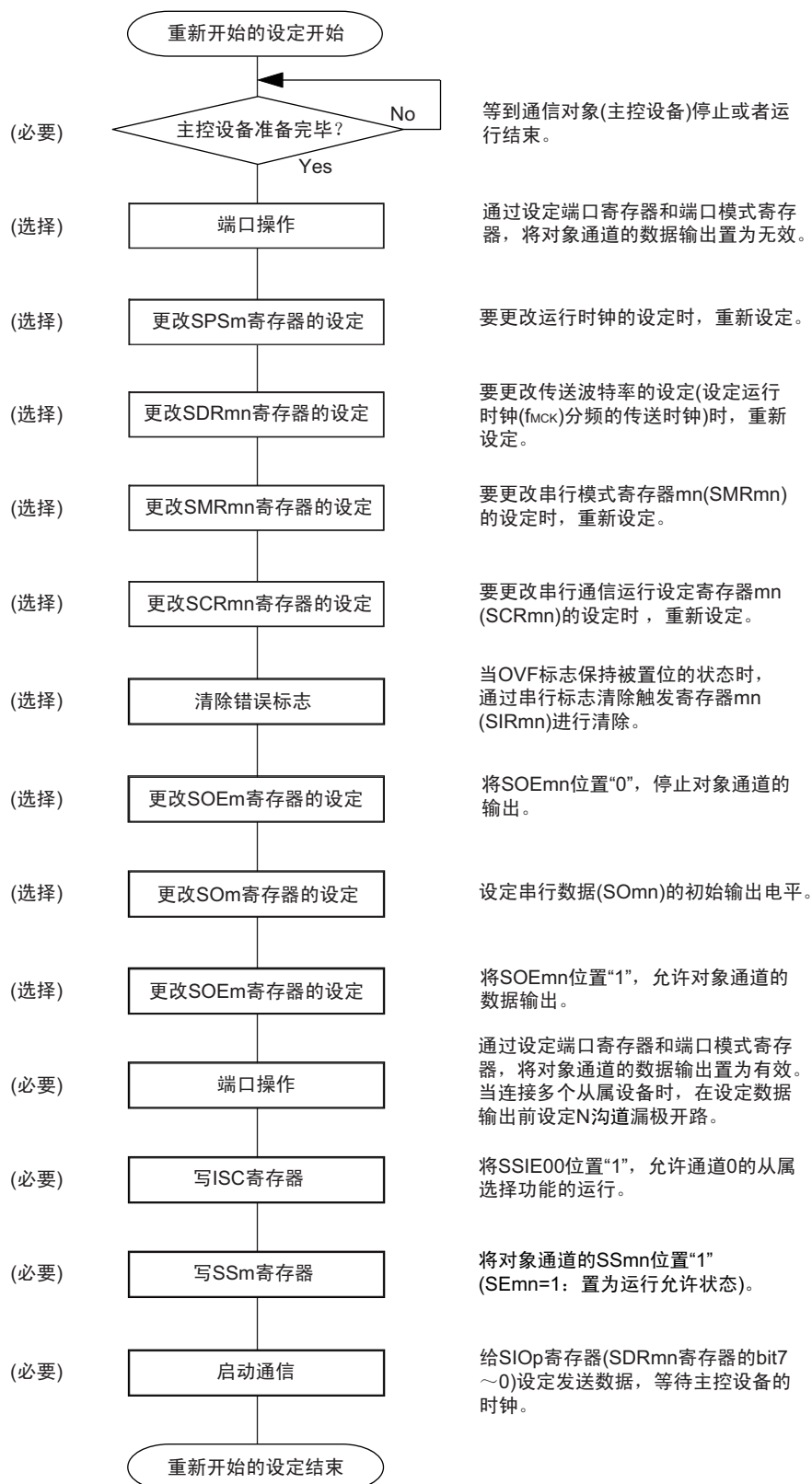
备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

图 14-80 从属发送的中止步骤



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

图 14-81 重新开始从属发送的设定步骤

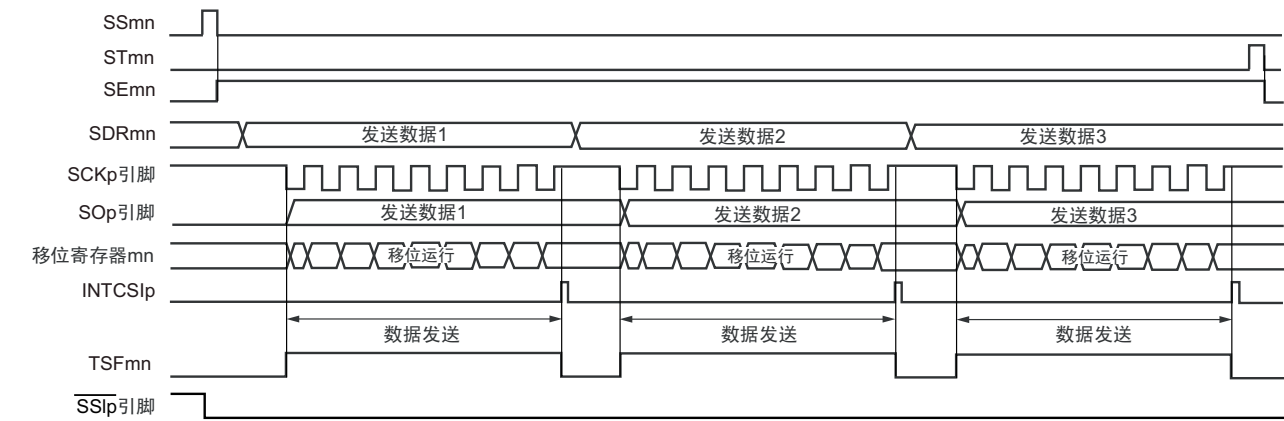


备注 1. 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（主控设备）停止或者通信结束后进行初始设定而不是进行重新开始设定。

2. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

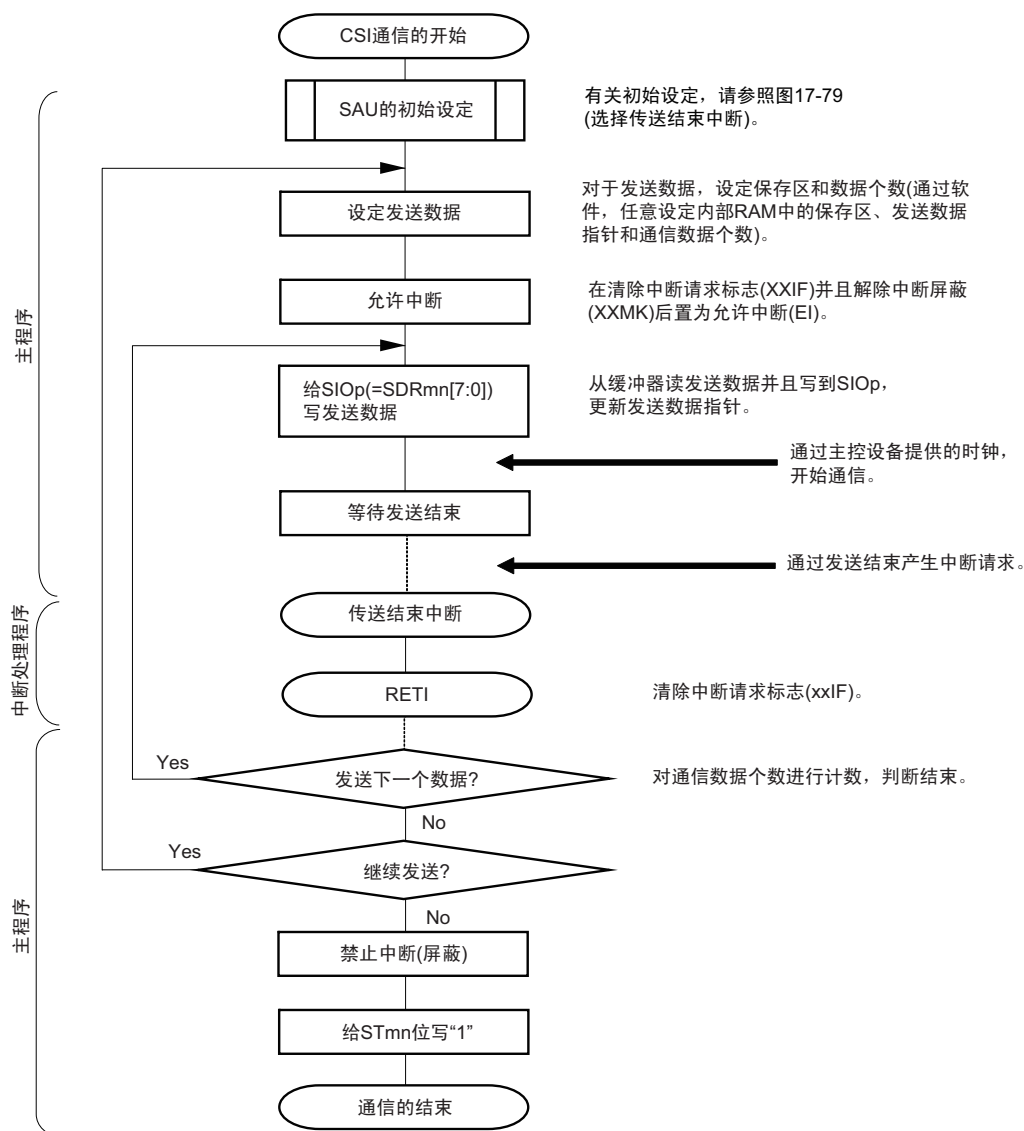
(3) 处理流程（单次发送模式）

图 14-82 从属发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

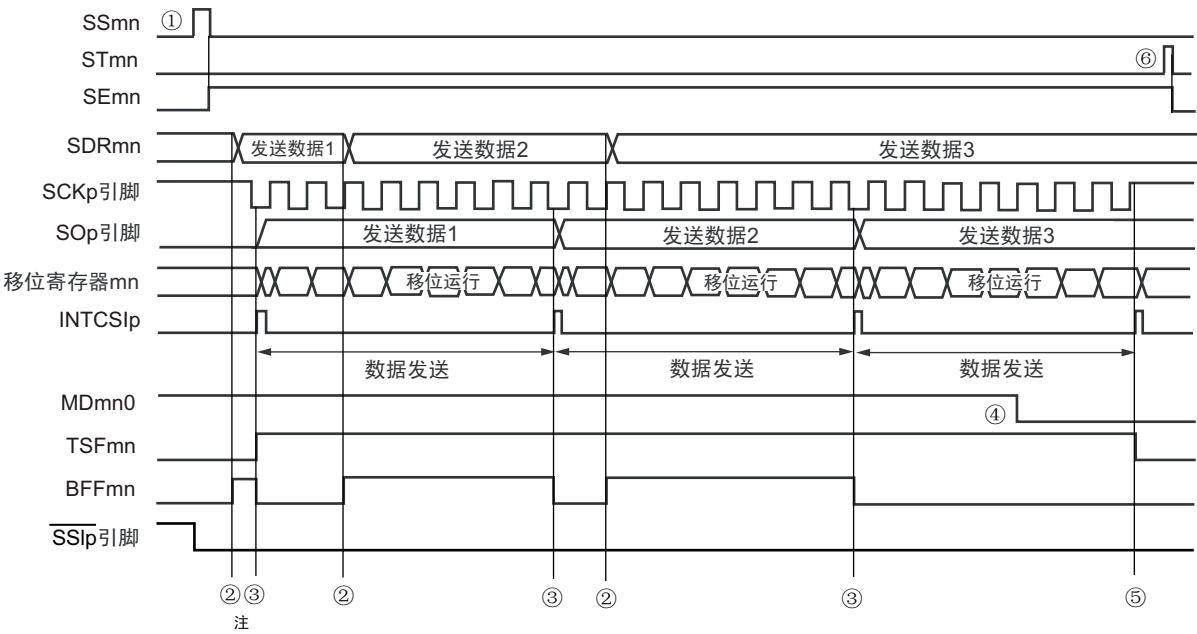
图 14-83 从属发送（单次发送模式）的流程图



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

(4) 处理流程（连续发送模式）

图 14-84 从属发送（连续发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

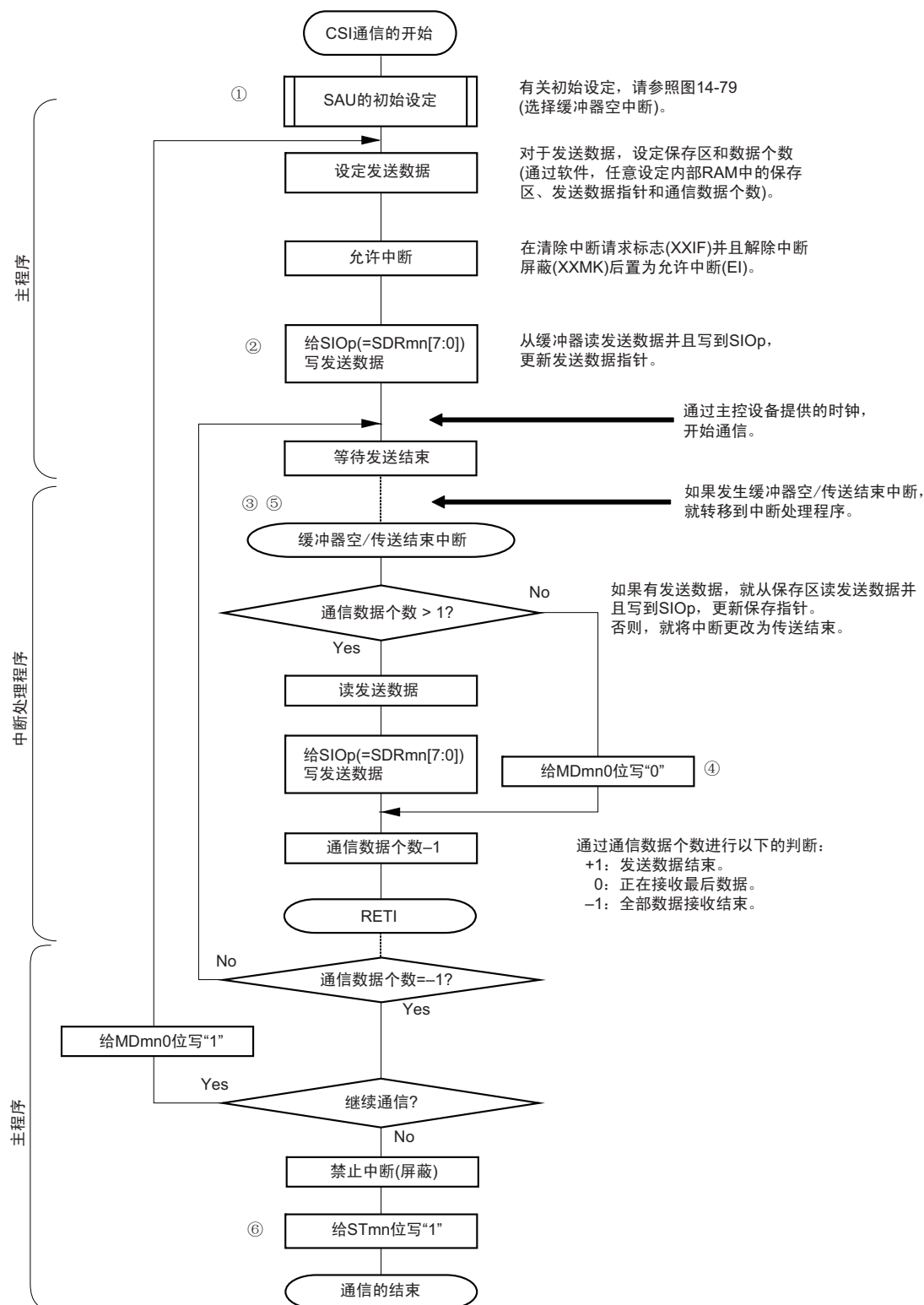


注 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

注意 即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，必须在开始传送最后一位之前进行改写。

备注 m: 单元号（m=0） n: 通道号（n=0） p: CSI 号（p=00）

图 14-85 从属发送（连续发送模式）的流程图



备注 1. 图中的①~⑥对应“图 14-84 从属发送（连续发送模式）的时序图”中的①~⑥。

2. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

14.6.2 从属接收

从属接收是指在从其他设备输入传送时钟的状态下 RL78 微控制器从其他设备接收数据的运行。

从属选择输入功能	CSI00
对象通道	SAU0 的通道 0
使用的引脚	SCK00、SI00、 $\overline{\text{SSI00}}$
中断	INTCSI00 只限于传送结束中断（禁止设定缓冲器空中断）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7 位或者 8 位
传送速率	Max. $f_{\text{MCK}}/6[\text{Hz}]$ 注 1、2
数据相位	能通过 SCRMn 寄存器的 DAPmn 位进行选择。 - 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: 不反相 - CKPmn=1: 反相
数据方向	MSB 优先或者 LSB 优先
从属选择输入功能	可选择从属选择输入功能的运行。

注 1. 因为在内部对 SCK00 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 $f_{\text{MCK}}/6[\text{Hz}]$ 。

2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 31 章 电特性”）的范围内使用。

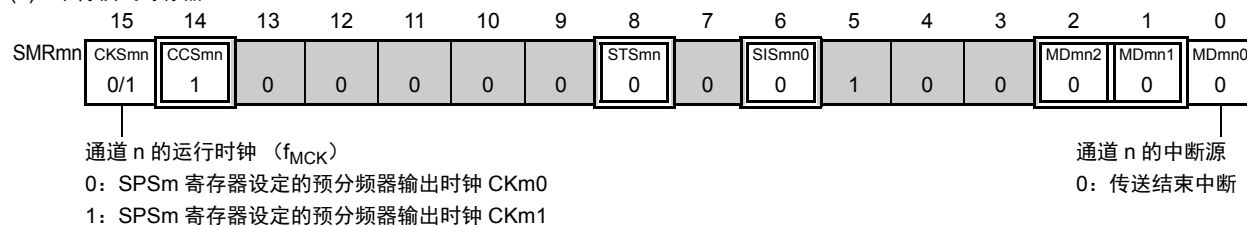
备注 1. f_{MCK} : 对象通道的运行时钟频率

2. m: 单元号 (m=0) n: 通道号 (n=0)

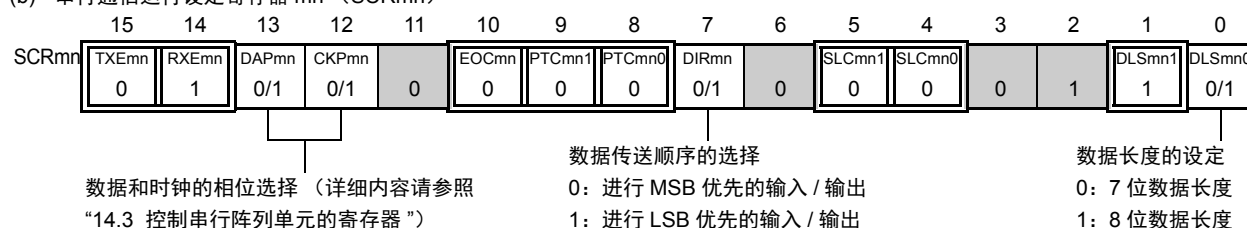
(1) 寄存器的设定

图 14-86 从属选择输入功能 (CSI00) 从属接收时的寄存器设定内容例子 (1/2)

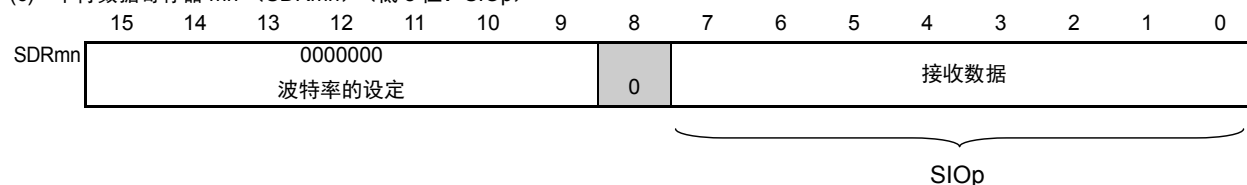
(a) 串行模式寄存器 mn (SMRmn)



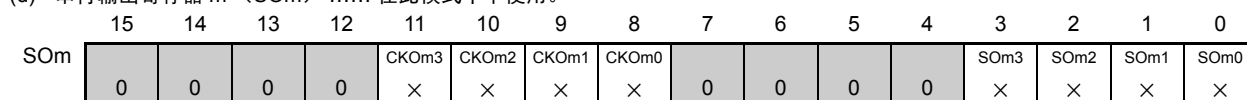
(b) 串行通信运行设定寄存器 mn (SCRmn)



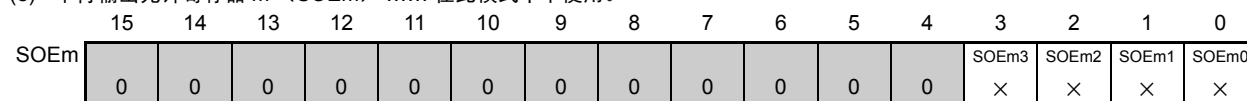
(c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



(d) 串行输出寄存器 m (SOM) 在此模式中不使用。



(e) 串行输出允许寄存器 m (SOEm) 在此模式中不使用。



备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

2. □: 在从属接收模式中为固定设定。■: 不能设定 (设定初始值)。
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

图 14-86 从属选择输入功能（CSI00）从属接收时的寄存器设定内容例子 (2/2)

(f) 串行通道开始寄存器 m（SSm） 只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

(g) 输入切换控制寄存器（ISC） 这是 CSI00 从属通道（单元 0 的通道 0）的 $\overline{\text{SSI00}}$ 引脚的控制。

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

0: $\overline{\text{SSI00}}$ 引脚的输入值无效
1: $\overline{\text{SSI00}}$ 引脚的输入值有效

备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

2. ☐ : 在从属发送模式中为固定设定。☐ : 不能设定（设定初始值）。
 × : 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
 0/1 : 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 14-87 从属接收的初始设定步骤

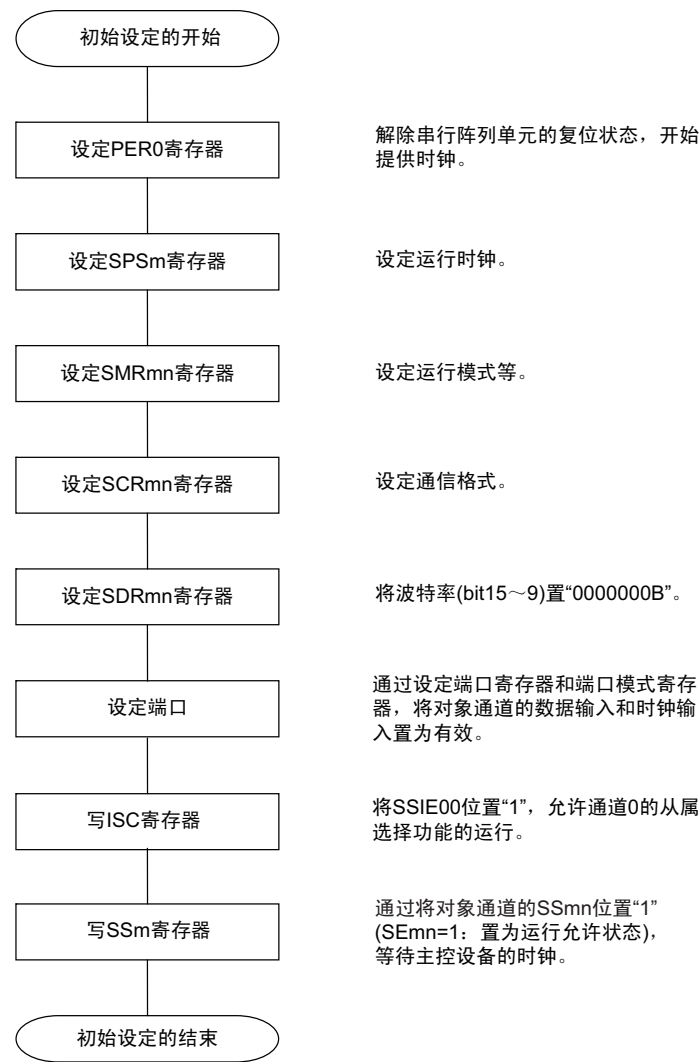
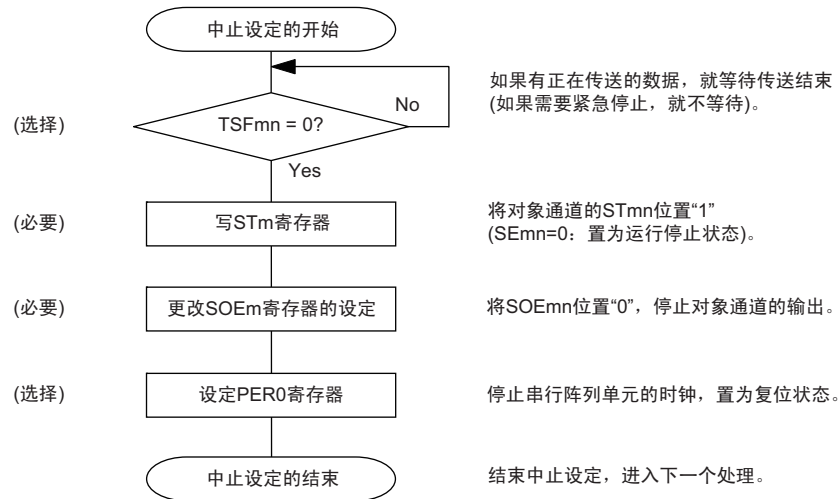
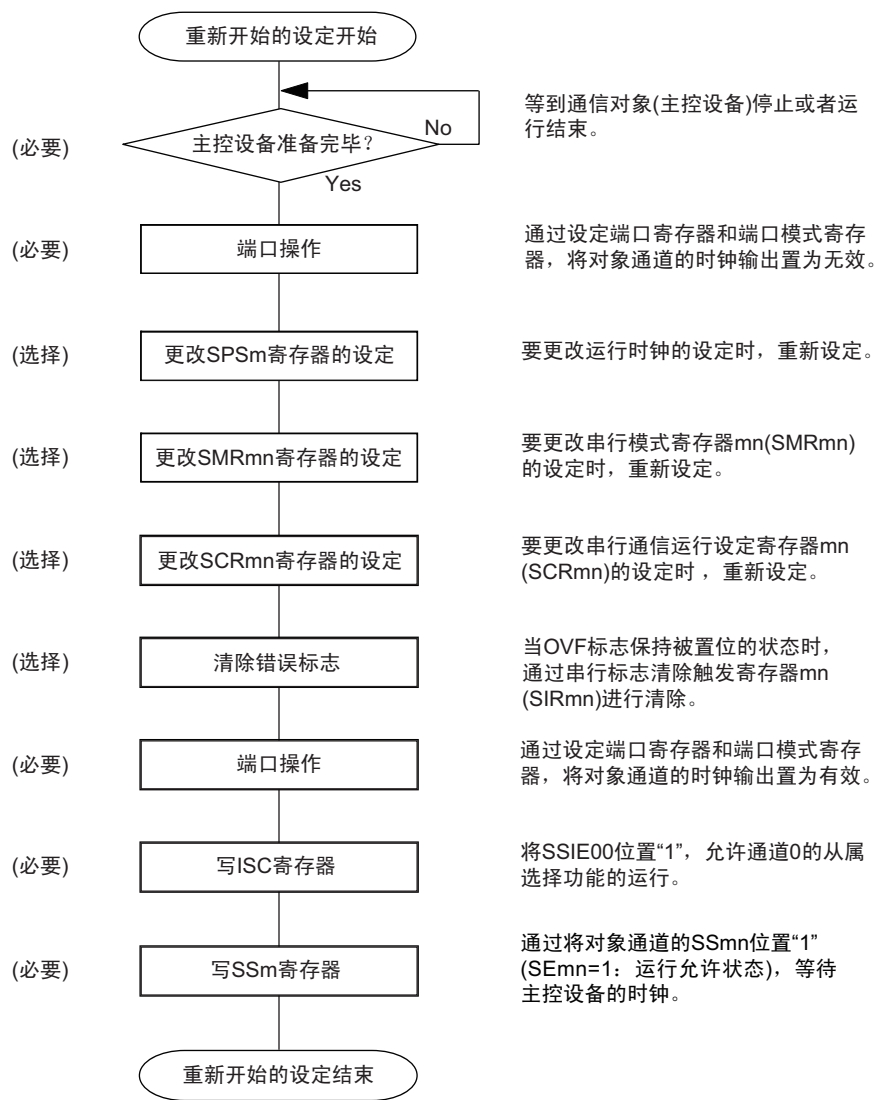


图 14-88 从属接收的中止步骤



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

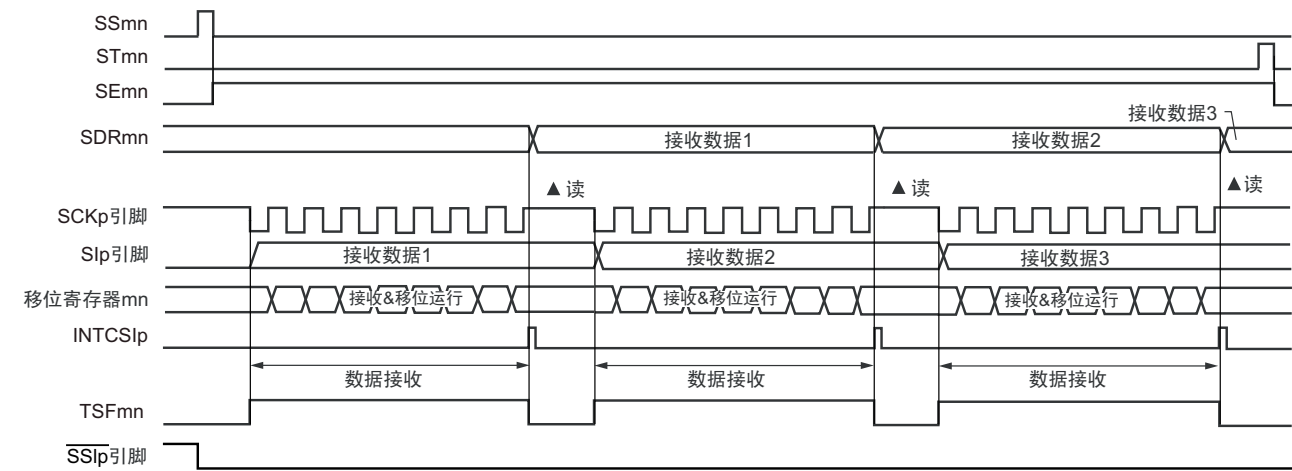
图 14-89 重新开始从属接收的设定步骤



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

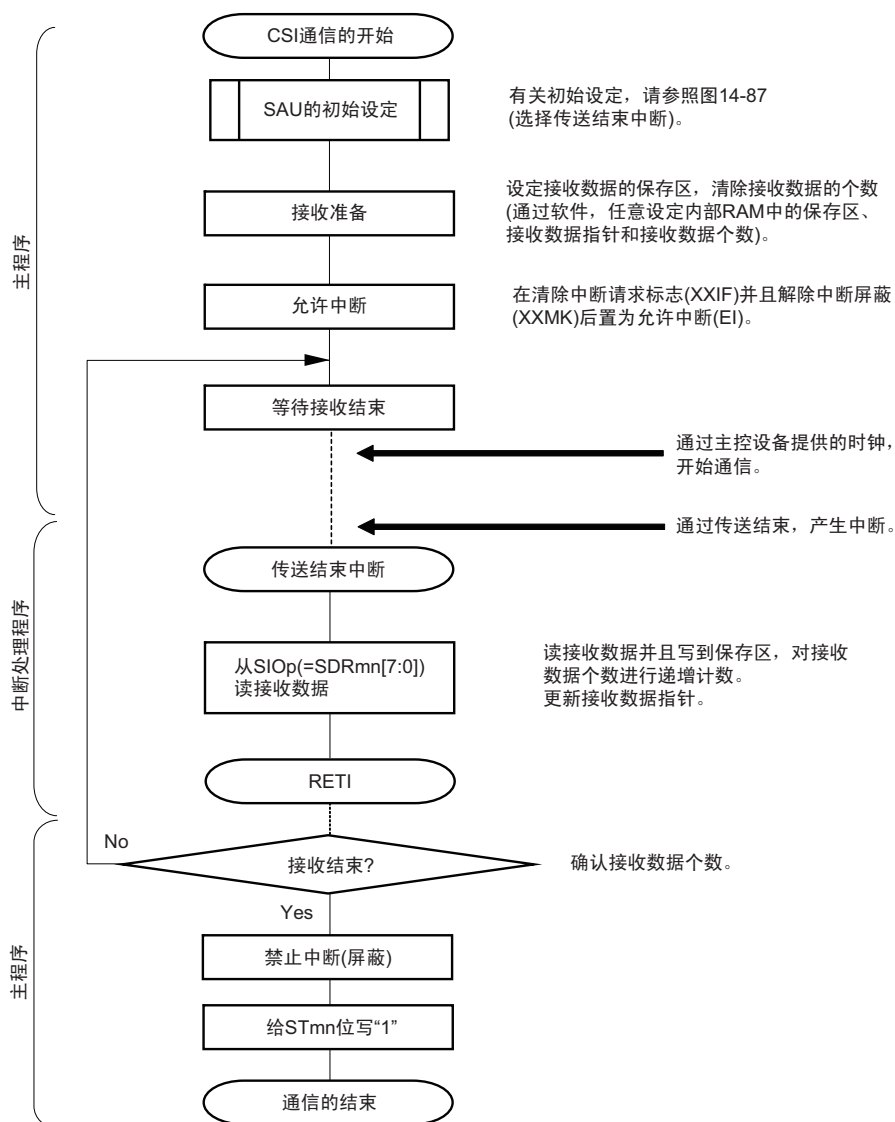
(3) 处理流程（单次接收模式）

图 14-90 从属接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

图 14-91 从属接收（单次接收模式）的流程图



14.6.3 从属的发送和接收

从属发送和接收是指在从其他设备输入传送时钟的状态下 RL78 微控制器和其他设备进行数据发送和接收的运行。

从属选择输入功能	CSI00
对象通道	SAU0 的通道 0
使用的引脚	SCK00、SI00、SO00、 $\overline{\text{SSI00}}$
中断	INTCSI00 可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错检测标志（OVFmn）。
传送数据长度	7 位或者 8 位
传送速率	Max. $f_{\text{MCK}}/6[\text{Hz}]$ 注 1、2
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 - 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: 不反相 - CKPmn=1: 反相
数据方向	MSB 优先或者 LSB 优先
从属选择输入功能	可选择从属选择输入功能的运行。

- 注 1. 因为在内部对 SCK00 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 $f_{\text{MCK}}/6[\text{Hz}]$ 。
 2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 31 章 电特性”）的范围内使用。

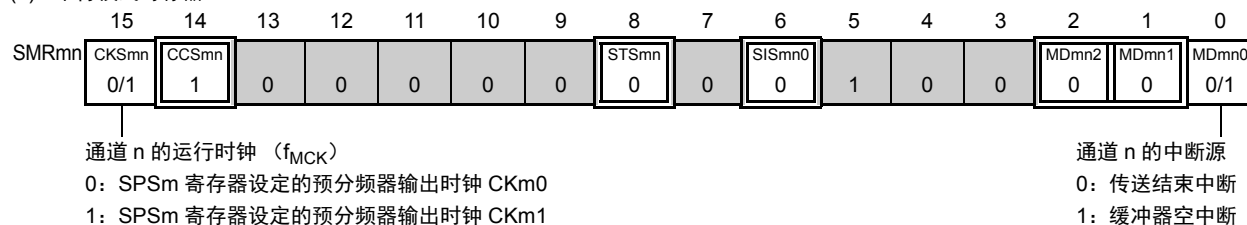
备注 1. f_{MCK} : 对象通道的运行时钟频率

2. m: 单元号 (m=0) n: 通道号 (n=0)

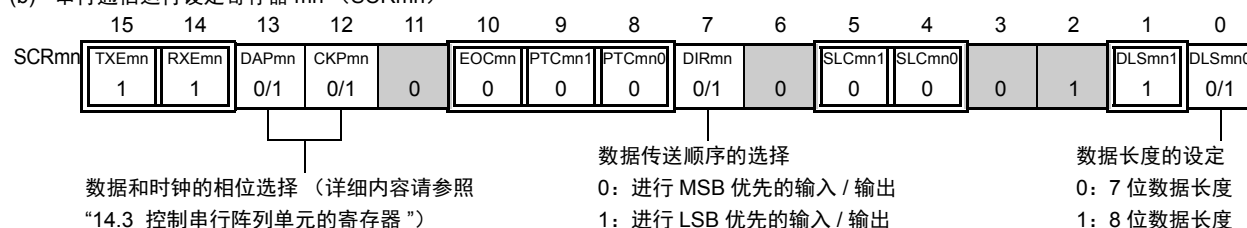
(1) 寄存器的设定

图 14-92 从属选择输入功能 (CSI00) 从属发送和接收时的寄存器设定内容例子 (1/2)

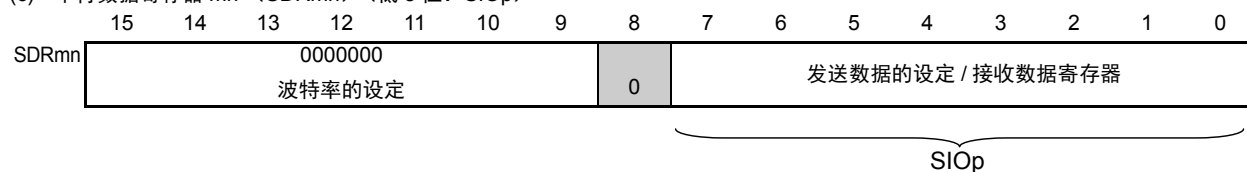
(a) 串行模式寄存器 mn (SMRmn)



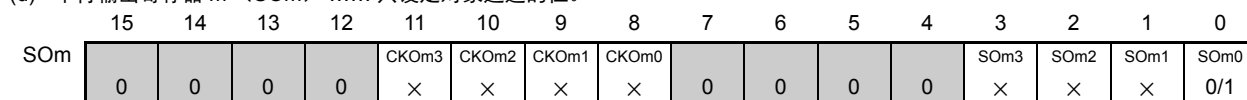
(b) 串行通信运行设定寄存器 mn (SCRmn)



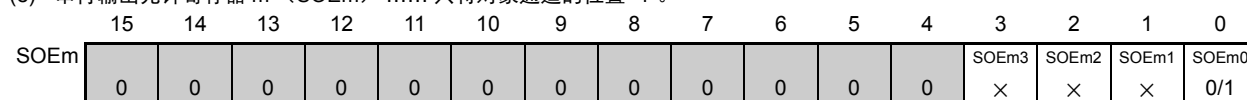
(c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



(d) 串行输出寄存器 m (SOM) 只设定对象通道的位。



(e) 串行输出允许寄存器 m (SOEm) 只将对象通道的位置“1”。



注意 在主控设备开始输出时钟前, 必须给 SIOp 寄存器设定发送数据。

备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

2. : 在 CSI 从属发送和接收模式中为固定设定。 : 不能设定 (设定初始值)。
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

图 14-92 从属选择输入功能（CSI00）从属发送和接收时的寄存器设定内容例子 (2/2)

(f) 串行通道开始寄存器 m（SSm） 只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

(g) 输入切换控制寄存器（ISC） 这是 CSI00 从属通道（单元 0 的通道 0）的 $\overline{\text{SSI00}}$ 引脚的控制。

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

0: $\overline{\text{SSI00}}$ 引脚的输入值无效
1: $\overline{\text{SSI00}}$ 引脚的输入值有效

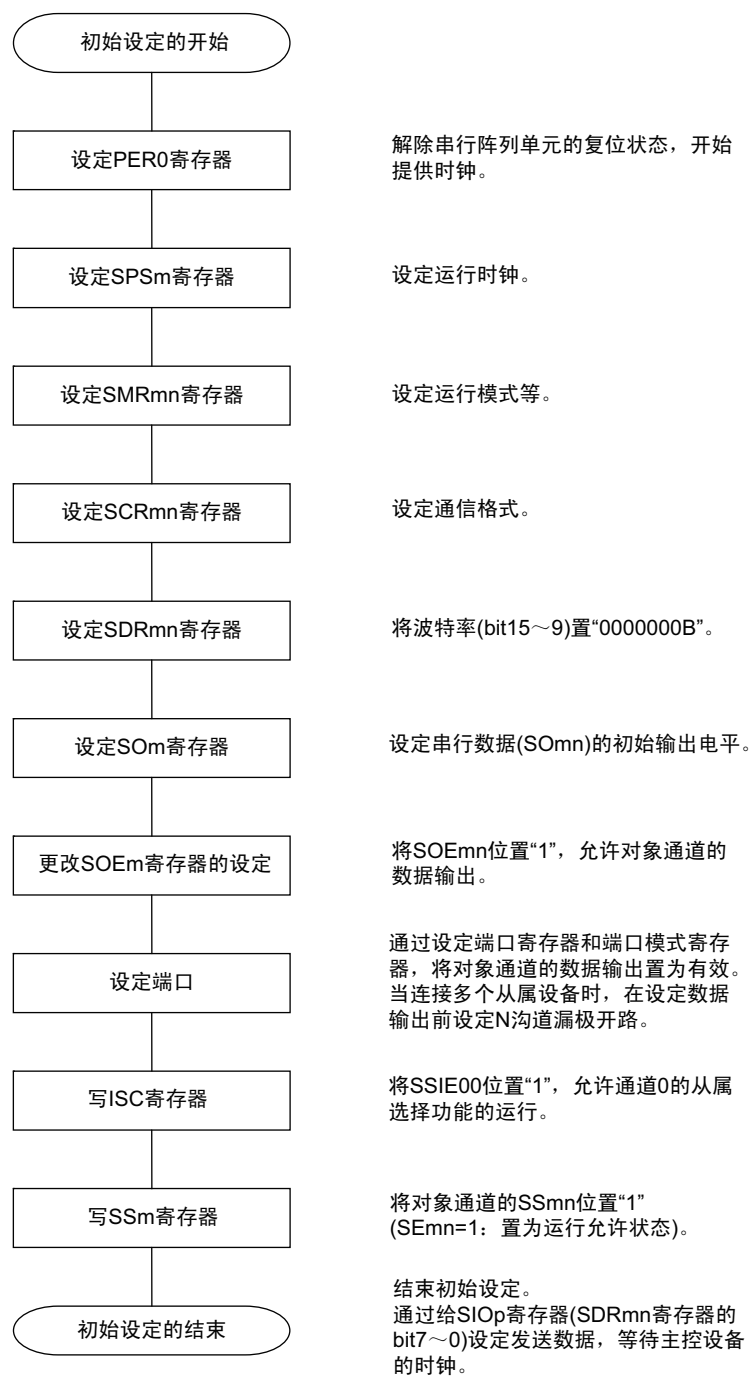
注意 在主机设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

2. ☐ : 在 CSI 从属发送和接收模式中为固定设定。 ☐ : 不能设定（设定初始值）。
 × : 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
 0/1 : 根据用户的用途置“0”或者“1”。

(2) 操作步骤

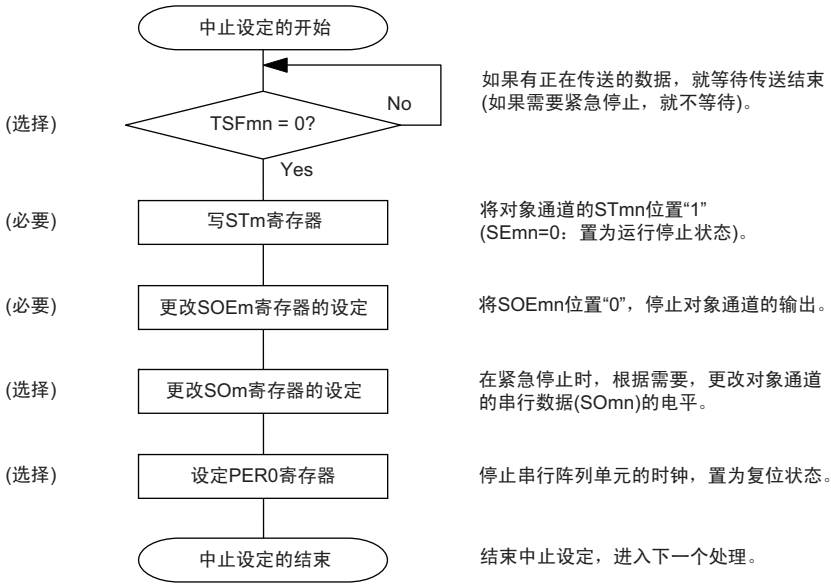
图 14-93 从属发送和接收的初始设定步骤



注意 在主导设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

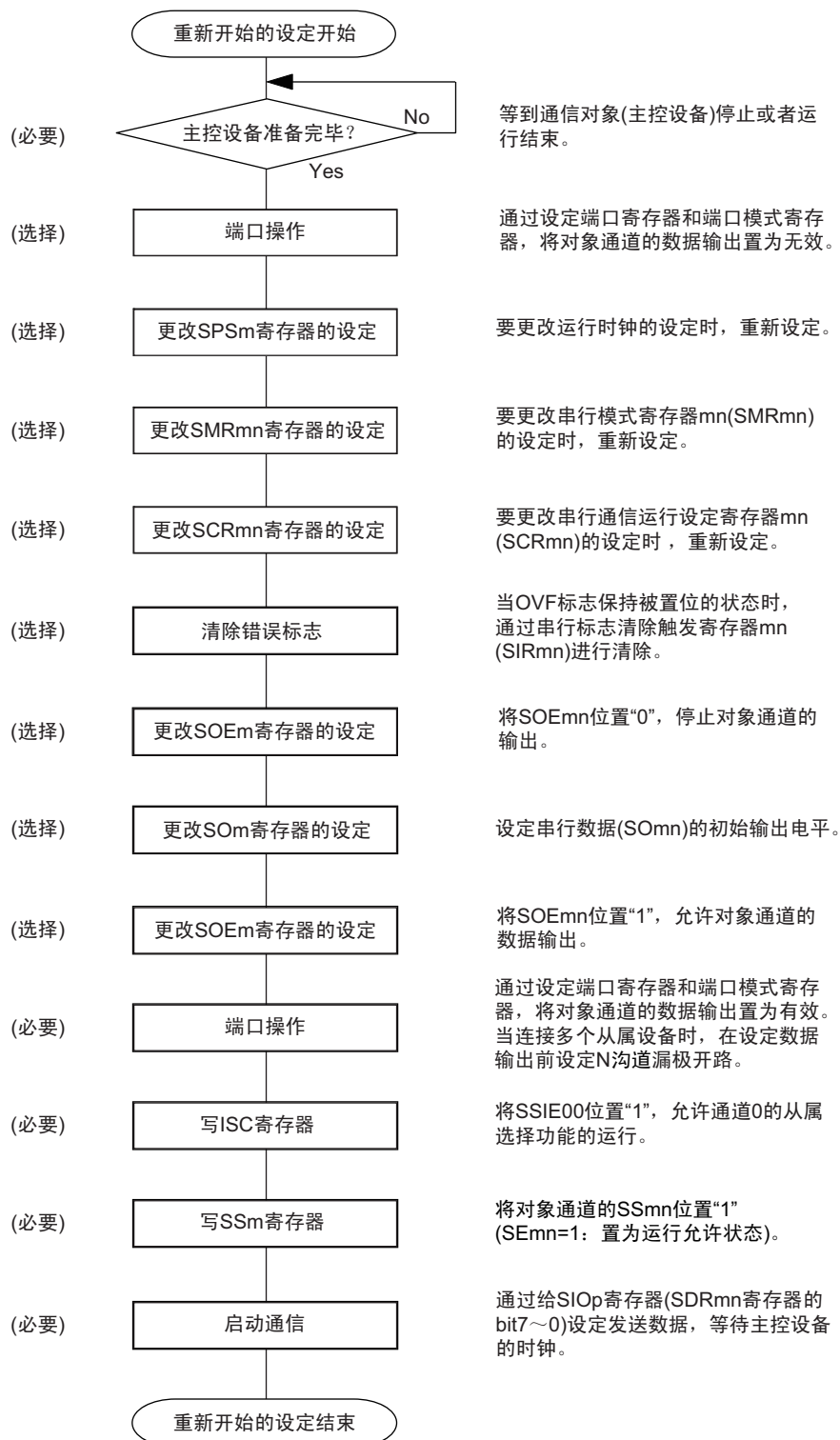
备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

图 14-94 从属发送和接收的中止步骤



备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

图 14-95 重新开始从属发送和接收的设定步骤

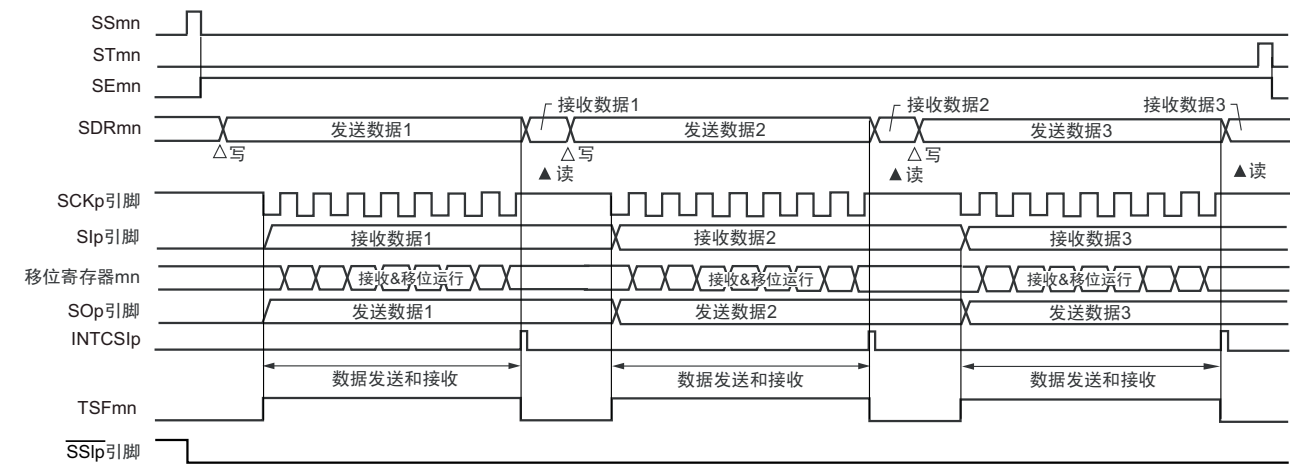


注意 1. 在主控设备开始输出时钟前, 必须给 SIOp 寄存器设定发送数据。

2. 如果在中止设定中改写 PER0 来停止提供时钟, 就必须在等到通信对象 (主控设备) 停止或者通信结束后进行初始设定而不是进行重新开始设定。

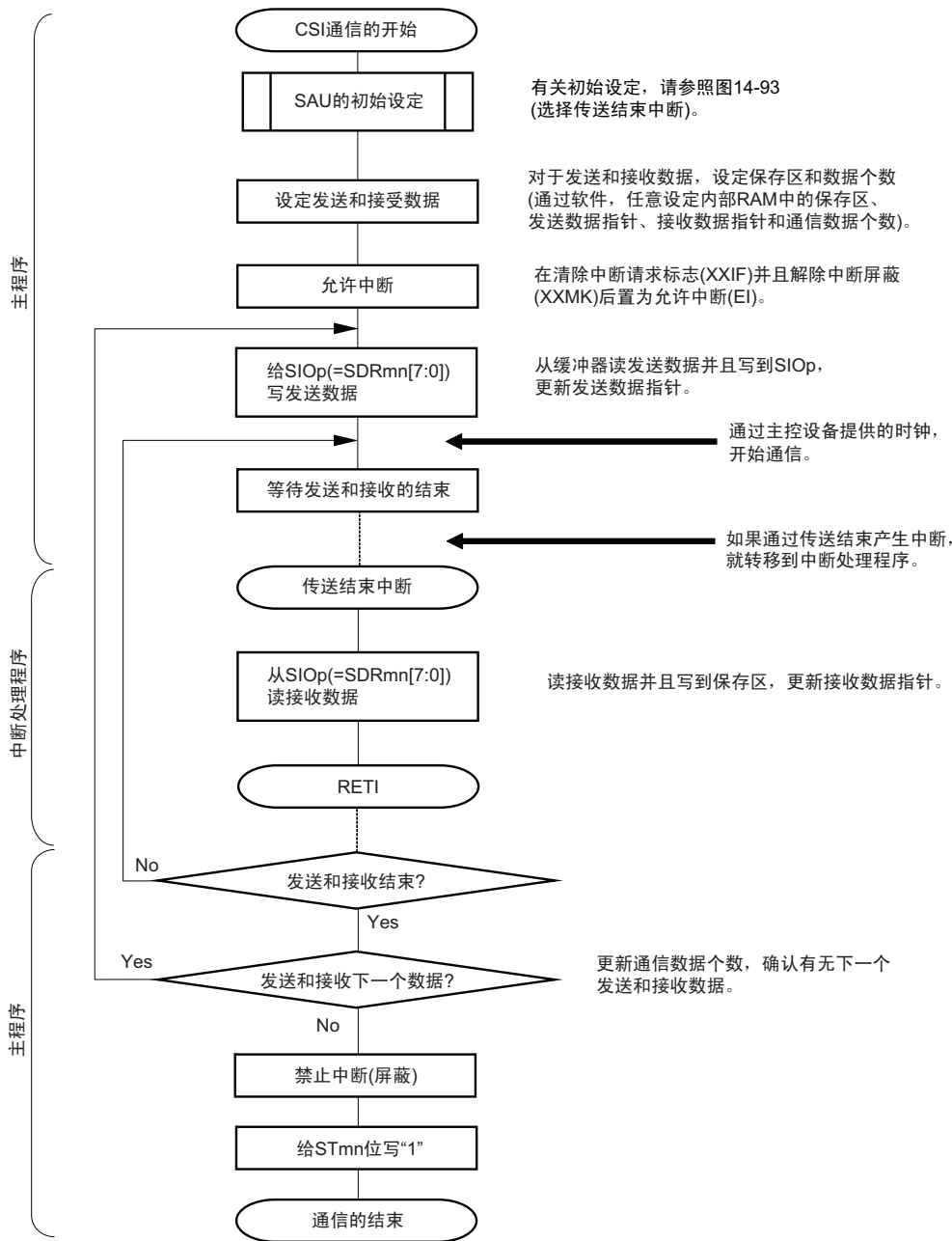
(3) 处理流程（单次发送和接收模式）

图 14-96 从属发送和接收（单次发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

图 14-97 从属发送和接收（单次发送和接收模式）的流程图

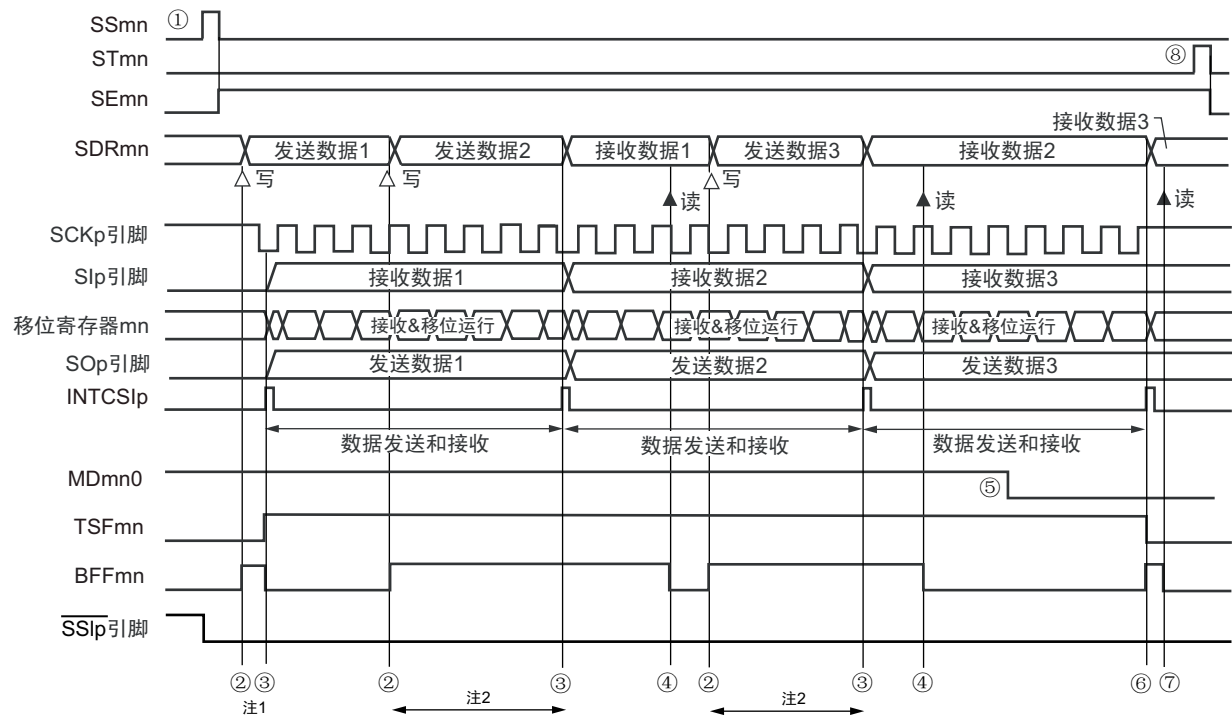


注意 在主控设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

(4) 处理流程（连续发送和接收模式）

图 14-98 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）

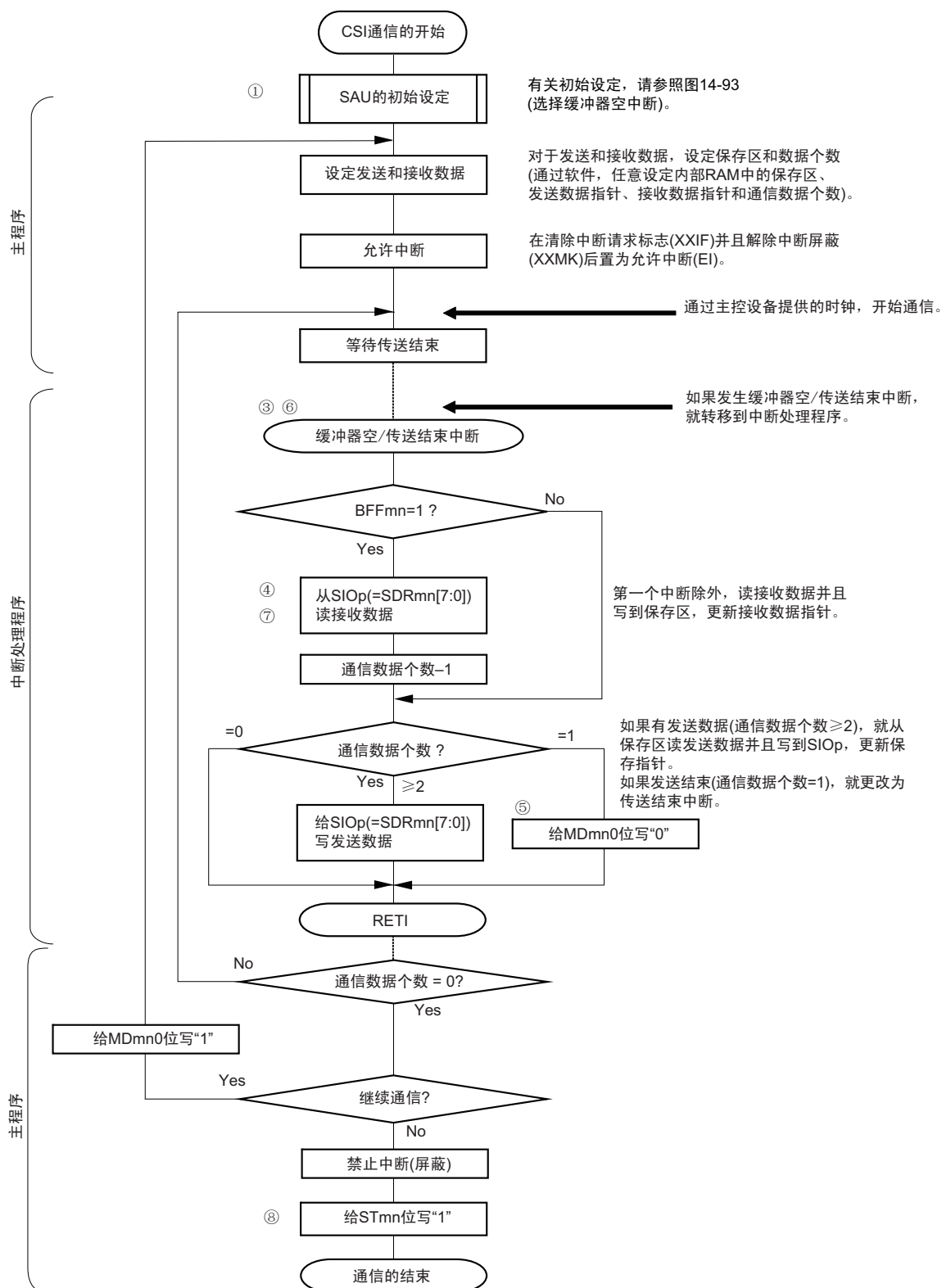


- 注 1. 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。
2. 如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。
- 注意 即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。

备注 1. 图中的①～⑧对应“图 14-99 从属发送和接收（连续发送和接收模式）的流程图”中的①～⑧。

2. m：单元号（m=0） n：通道号（n=0） p：CSI 号（p=00）

图 14-99 从属发送和接收（连续发送和接收模式）的流程图



注意 在主控设备开始输出时钟前，必须给 SIop 寄存器设定发送数据。

备注 1. 图中的①~⑧对应“图 14-98 从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

2. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

14.6.4 传送时钟频率的计算

从属选择输入功能（CSI00）通信的传送时钟频率能用以下计算式进行计算。

(1) 从属设备

(传送时钟频率) = { 主控设备提供的串行时钟 (SCK) 频率 } ^注 [Hz]

注 容许的最大传送时钟频率为 $f_{MCK}/6$ 。

备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

表 14-3 从属选择输入功能运行时钟的选择

SMRmn 寄存器	SPSm 寄存器								运行时钟 (f _{MCK}) 注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f _{CLK} =32MHz 运行时
0	X	X	X	X	0	0	0	0	f _{CLK}	32MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	1MHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	500kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	250kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	125kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	62.5kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	31.25kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	15.63kHz
	X	X	X	X	1	1	0	0	f _{CLK} /2 ¹²	7.81kHz
	X	X	X	X	1	1	0	1	f _{CLK} /2 ¹³	3.91kHz
	X	X	X	X	1	1	1	0	f _{CLK} /2 ¹⁴	1.95kHz
	X	X	X	X	1	1	1	1	f _{CLK} /2 ¹⁵	977Hz

注 要更改被选择为 f_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止串行阵列单元（SAU）的运行（串行通道停止寄存器 m（STm）=000FH）后进行更改。

备注 1. X: 忽略

2. m: 单元号 (m=0) n: 通道号 (n=0)

14.6.5 在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤

在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤如图 14-100 所示。

图 14-100 发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。→	SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn（SSRmn）。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn（SIRmn）写“1”。	清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SIRmn 寄存器，只能清除读操作时的错误。

备注 m: 单元号 (m=0) n: 通道号 (n=0)

14.7 UART (UART0 ~ UART2) 通信的运行

这是通过串行数据发送 (TxD) 和串行数据接收 (RxD) 共 2 条线进行异步通信的功能。使用这 2 条通信线, 按数据帧 (由起始位、数据、奇偶校验位和停止位构成) 与其他通信方进行异步 (使用内部波特率) 的数据发送和接收。能通过使用发送专用 (偶数通道) 和接收专用 (奇数通道) 共 2 个通道来实现全双工异步 UART 通信, 而且还能通过组合 UART0、定时器阵列单元 0 (通道 3) 和外部中断 (INTP0) 来支持 LIN-bus。

[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。当选项字节 (000C2H) 的 FRQSEL4 位为 “0” 时, 只有以下的 UART 才能设定 SNOOZE 模式。

- UART0

UART0 (单元 0 的通道 0 和通道 1) 支持 LIN-bus。

[LIN-bus 功能]

• 唤醒信号的检测 • 间隔段 (BF) 的检测 • 同步段的测量、波特率的计算	}	使用外部中断 (INTP0) 和定时器阵列单元 0 (通道 3)。

注 只有以下的 UART 支持 9 位的数据长度。

- 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。

○ 32 引脚产品

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00 (支持从属选择输入功能)	UART0 (支持 LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

○ 64 引脚产品

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00 (支持从属选择输入功能)	UART0 (支持 LIN-bus)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

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 和 CSI01。但是，在使用 UART0 的同时，不同通道的通道 2 和通道 3 能使用 CSI10、UART1 或者 IIC10。

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 4 种通信运行：

- UART 发送 (参照 14.7.1)
- UART 接收 (参照 14.7.2)
- LIN 发送 (只限于 UART0) (参照 14.8.1)
- LIN 接收 (只限于 UART0) (参照 14.8.2)

14.7.1 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
	可选择发送结束中断（单次发送模式）或者缓冲器空中断（连续发送模式）。		
错误检测标志	无		
传送数据长度	7 位、8 位或者 9 位 ^{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	不反相输出（默认值：高电平）。 反相输出（默认值：低电平）。		
奇偶校验位	可选择以下内容： • 无奇偶校验位。 • 附加零校验。 • 附加偶校验。 • 附加奇校验。		
停止位	可选择以下内容： • 附加 1 位。 • 附加 2 位。		
数据方向	MSB 优先或者 LSB 优先		

注 1. 只有以下的 UART 支持 9 位的数据长度。

- 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 31 章 电特性”）的范围内使用。

备注 1. f_{MCK} : 对象通道的运行时钟频率

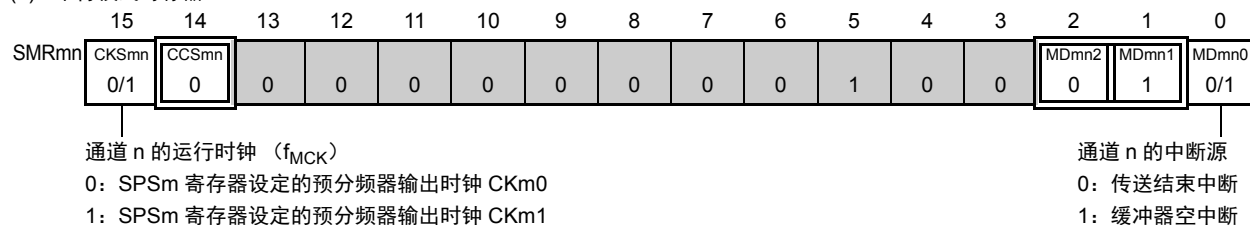
f_{CLK} : 系统时钟频率

2. m: 单元号 (m=0、1) n: 通道号 (n=0、2) mn=00、02、10

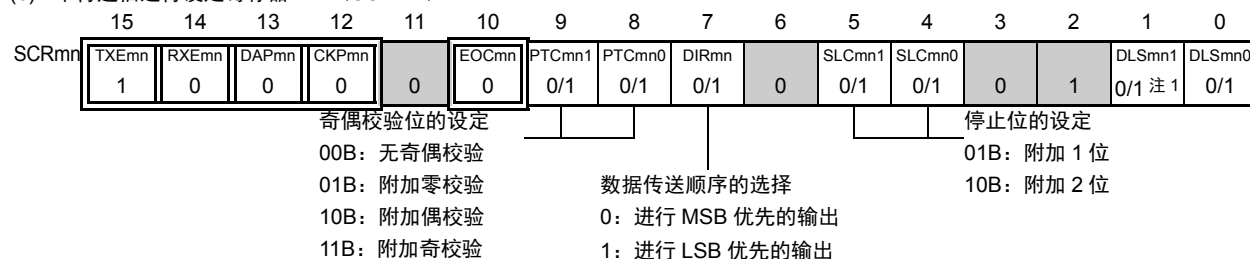
(1) 寄存器的设定

图 14-101 UART (UART0 ~ UART2) 的 UART 发送时的寄存器设定内容例子 (1/2)

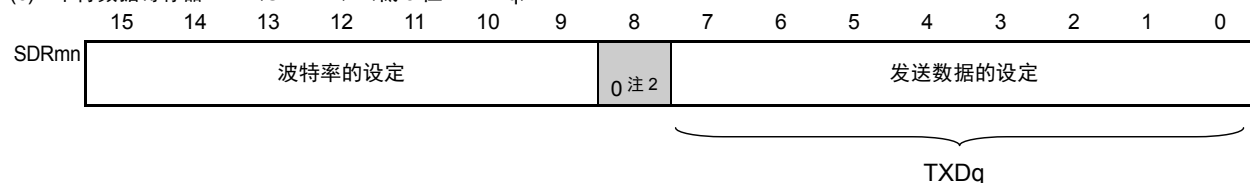
(a) 串行模式寄存器 mn (SMRmn)



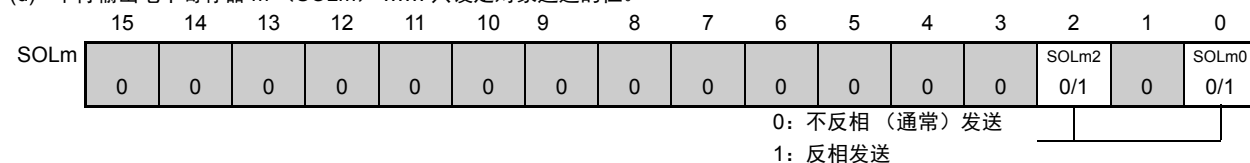
(b) 串行通信运行设定寄存器 mn (SCRmn)



(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。

- 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)。

×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

图 14-101 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) 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)。
 × : 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 14-102 UART 发送的初始设定步骤

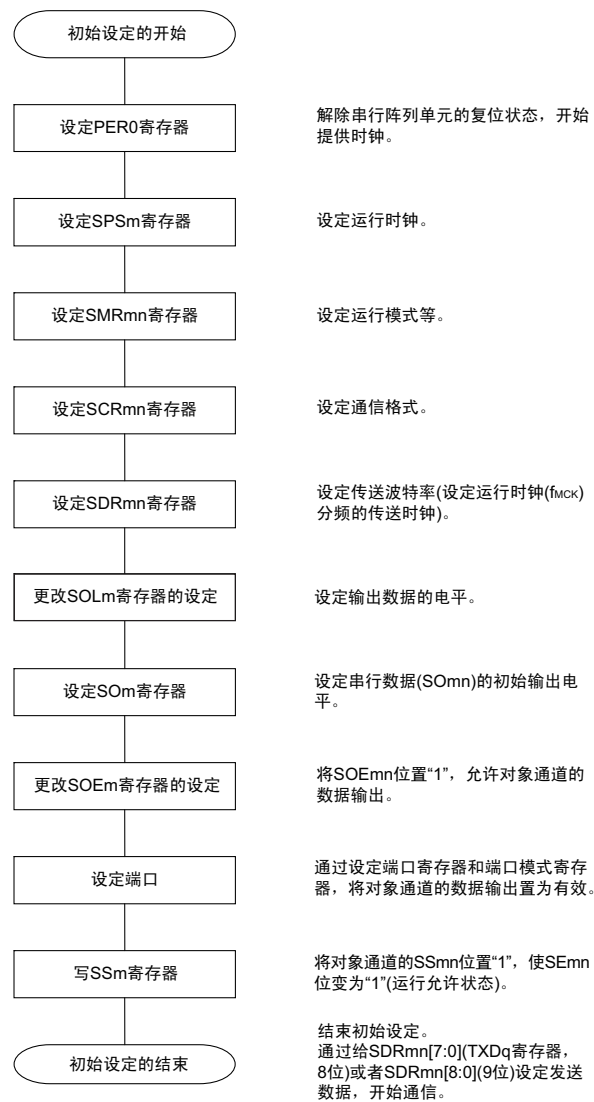


图 14-103 UART 发送的中止步骤

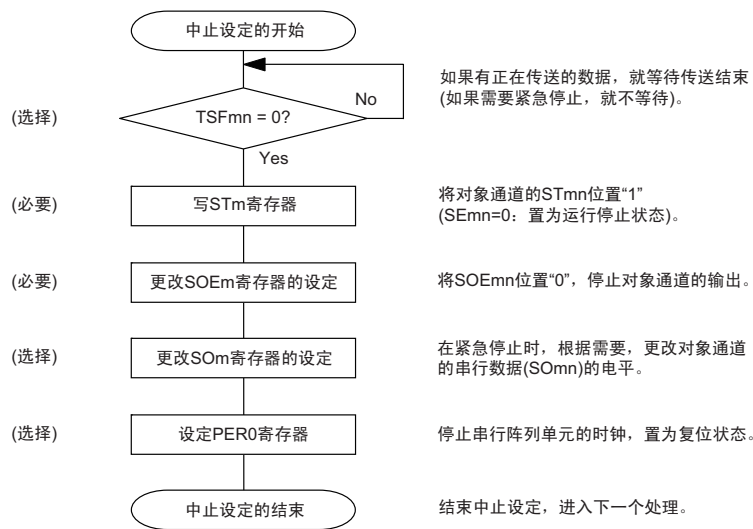
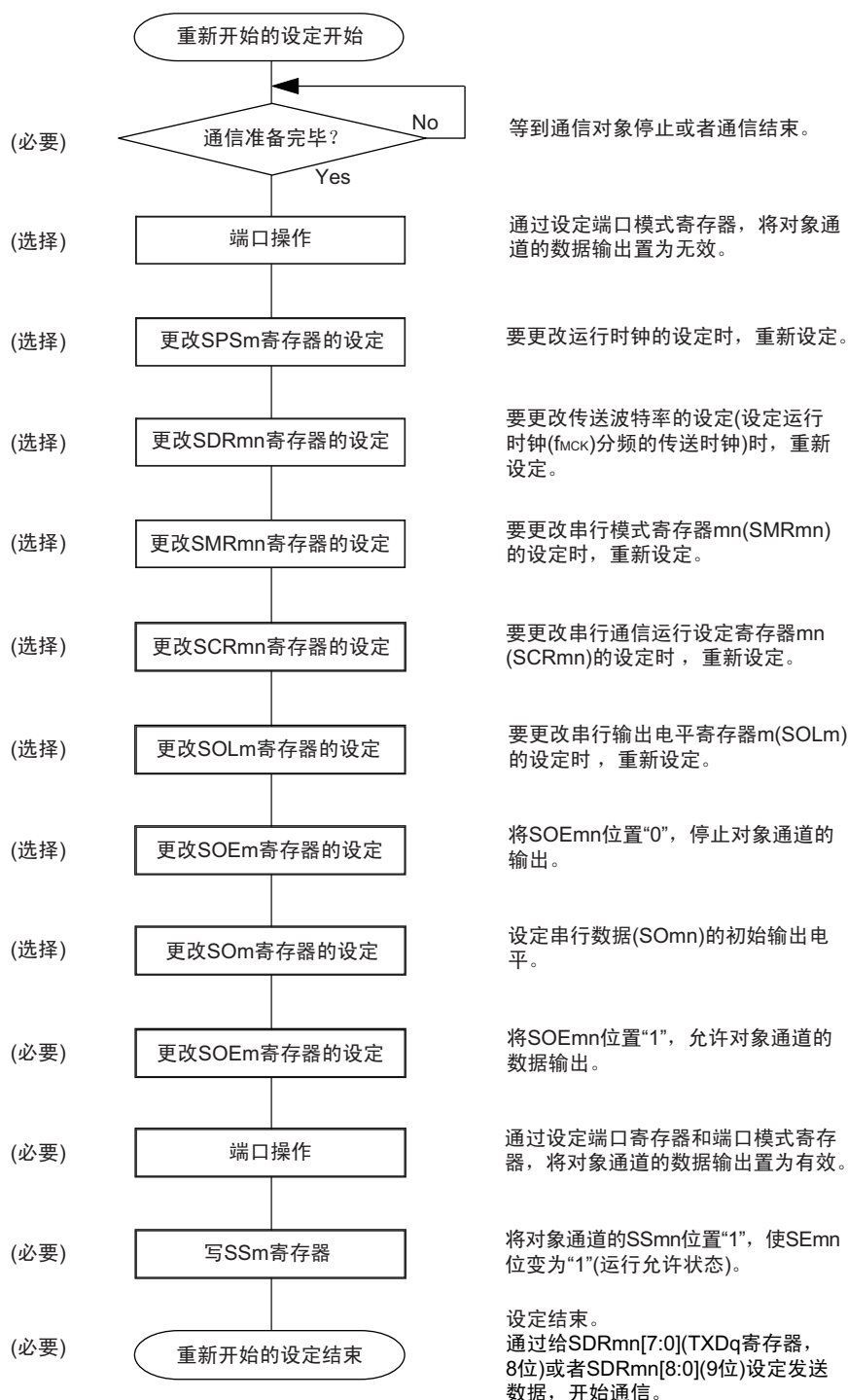


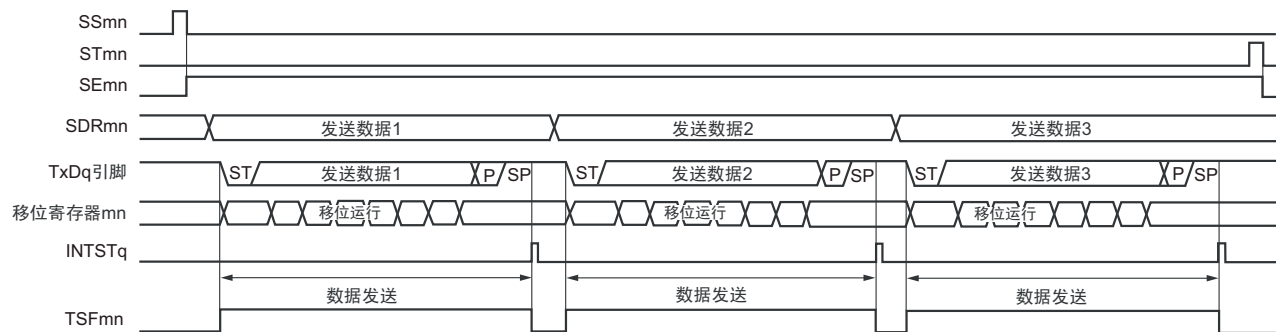
图 14-104 重新开始 UART 发送的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象停止或者通信结束后进行初始设定而不是进行重新开始设定。

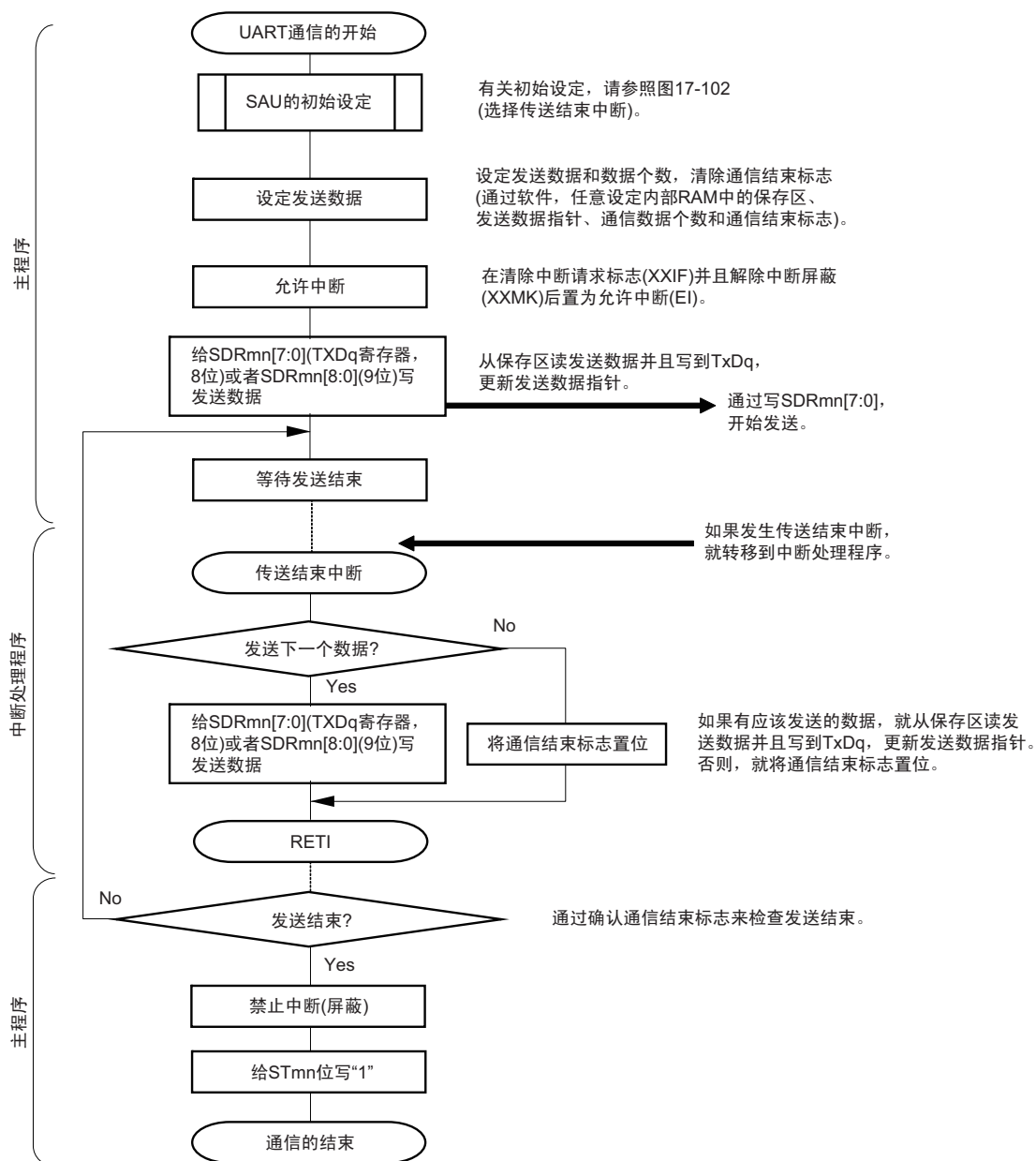
(3) 处理流程（单次发送模式）

图 14-105 UART 发送（单次发送模式）的时序图



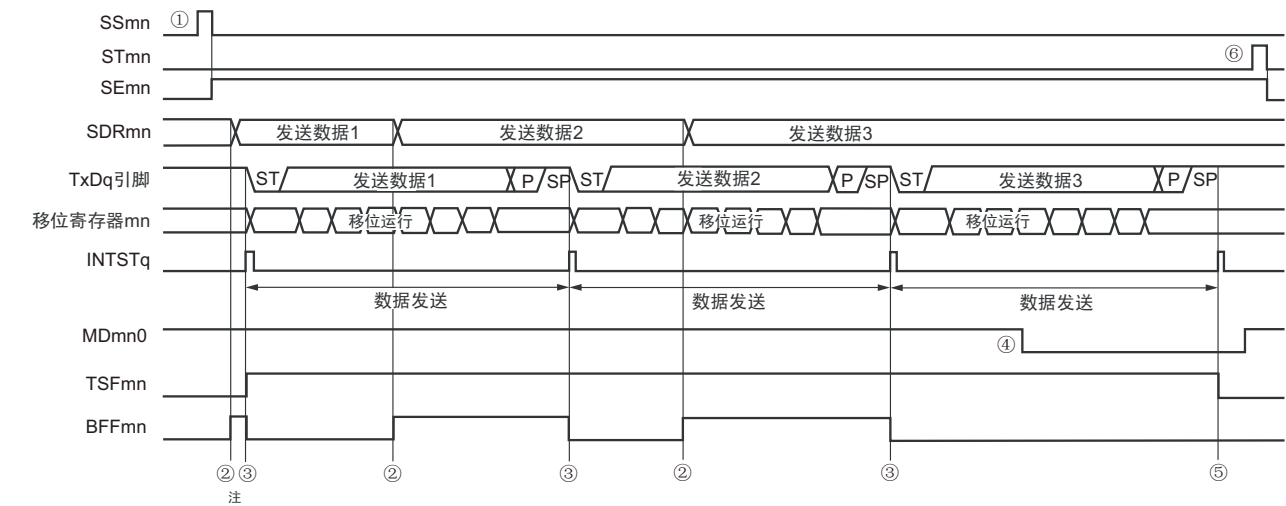
备注 m: 单元号 (m=0、1) n: 通道号 (n=0、2) q: UART 号 (q=0 ~ 2) mn=00、02、10

图 14-106 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 14-107 UART 发送（连续发送模式）的时序图

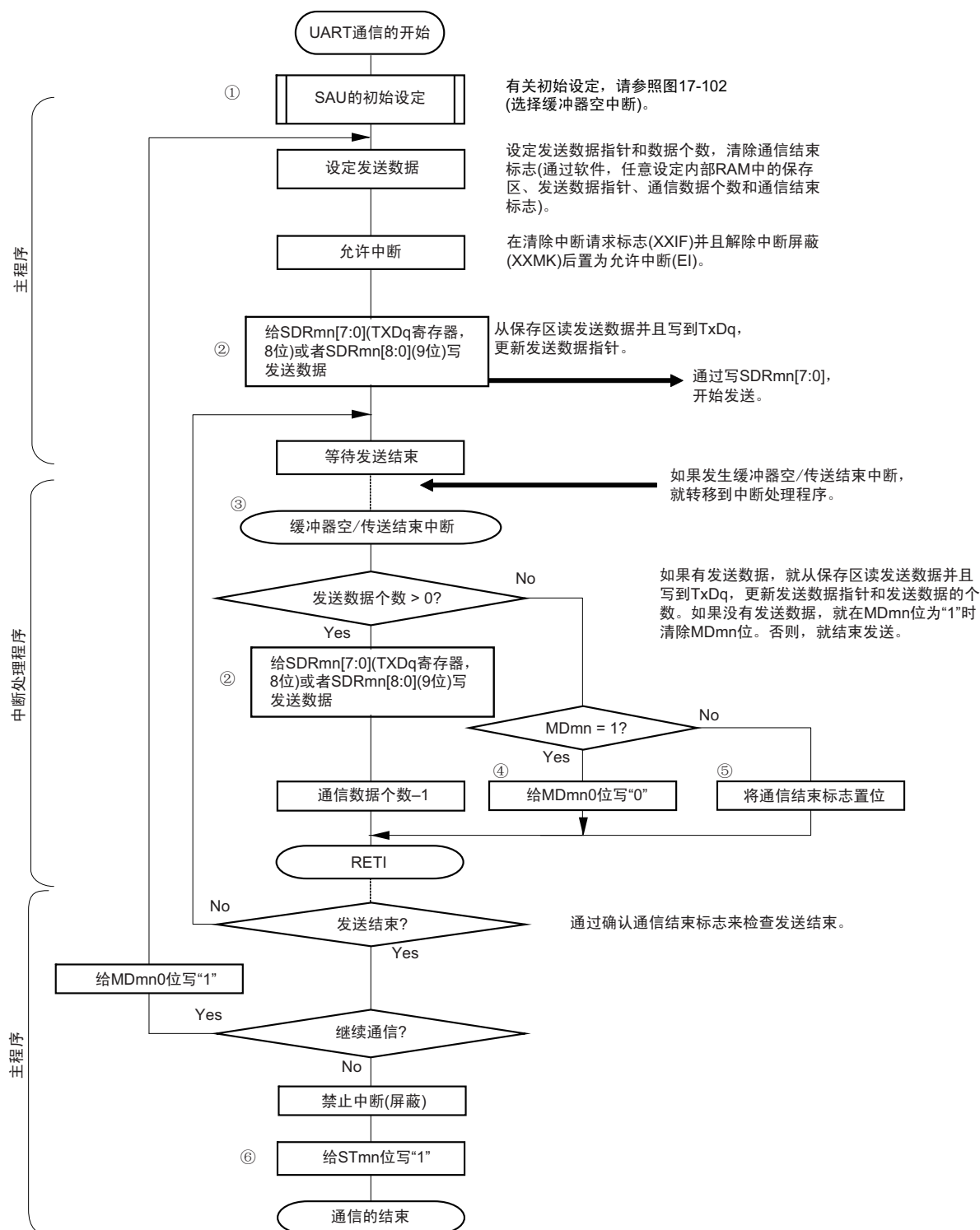


如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

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 14-108 UART 发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图 14-107 UART 发送（连续发送模式）的时序图”中的①~⑥。

14.7.2 UART 接收

UART 接收是 RL78 微控制器其他设备异步接收数据的运行。

UART 使用的 2 个通道中的奇数通道用于 UART 接收。但是，需要设定奇数通道和偶数通道的 SMR 寄存器。

UART	UART0	UART1	UART2
对象通道	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	INTST0	INTST1	INTST2
	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	• 帧错误检测标志（FEFmn） • 奇偶校验错误检测标志（PEFmn） • 溢出错检测标志（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	不反相输出（默认值：高电平）。 反相输出（默认值：低电平）。		
奇偶校验位	可选择以下内容： • 无奇偶校验位（无奇偶校验）。 • 附加零校验（无奇偶校验）。 • 偶校验 • 奇校验		
停止位	附加 1 位。		
数据方向	MSB 优先或者 LSB 优先		

注 1. 只有以下的 UART 支持 9 位的数据长度。

- 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 31 章 电特性”）的范围内使用。

备注 1. f_{MCK} ：对象通道的运行时钟频率

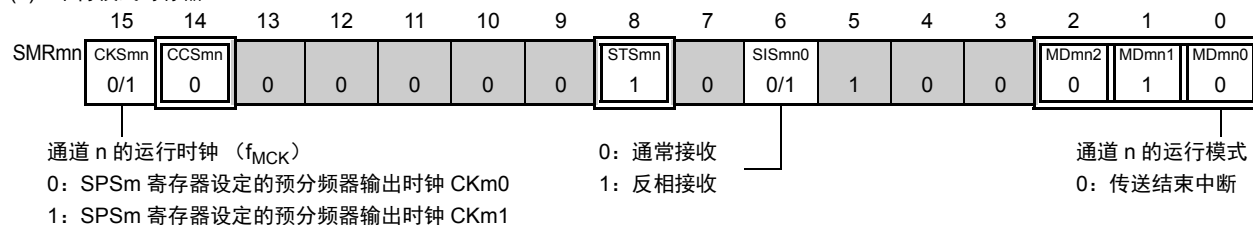
f_{CLK} ：系统时钟频率

2. m：单元号（m=0、1） n：通道号（n=1、3） mn=01、03、11

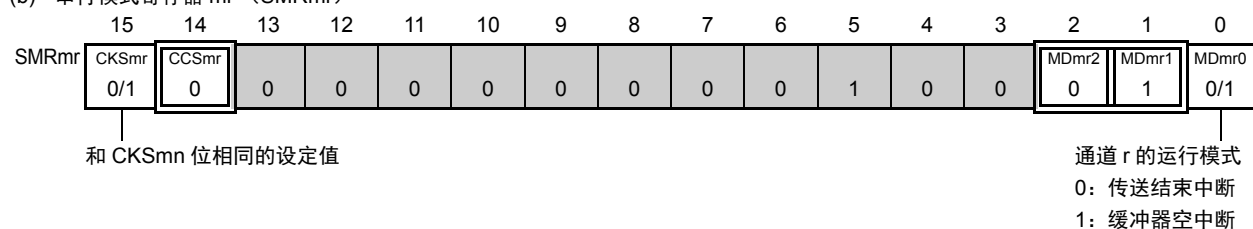
(1) 寄存器的设定

图 14-109 UART (UART0 ~ UART2) 的 UART 接收时的寄存器设定内容例子 (1/2)

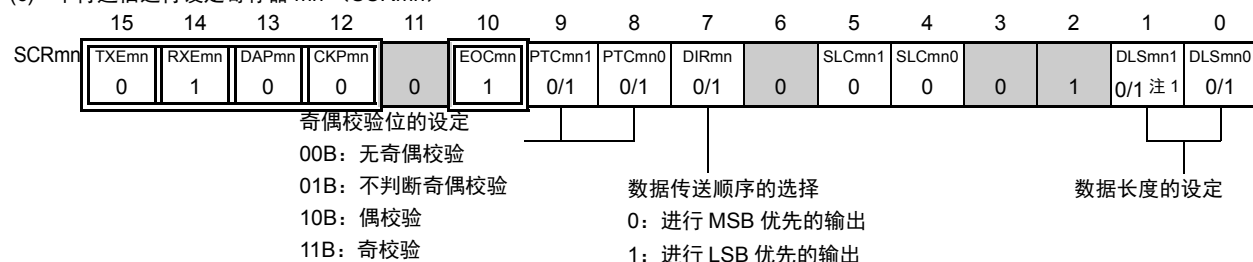
(a) 串行模式寄存器 mn (SMRmn)



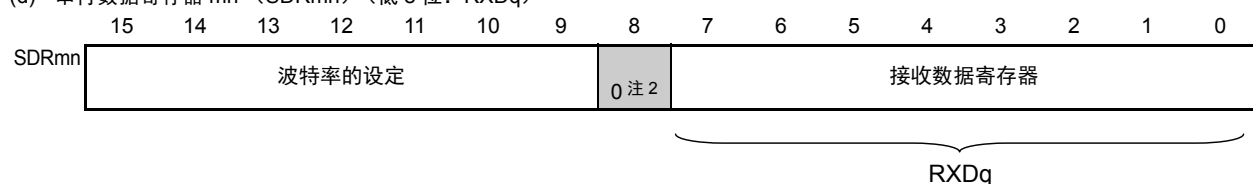
(b) 串行模式寄存器 mr (SMRmr)



(c) 串行通信运行设定寄存器 mn (SCRmn)



(d) 串行数据寄存器 mn (SDRmn) (低 8 位: RXDq)



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。

- UART0

注意 在 UART 接收时, 还必须设定与通道 n 成对的通道 r 的 SMRmr 寄存器。

备注 1. m: 单元号 (m=0、1) 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)。

×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

图 14-109 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”。

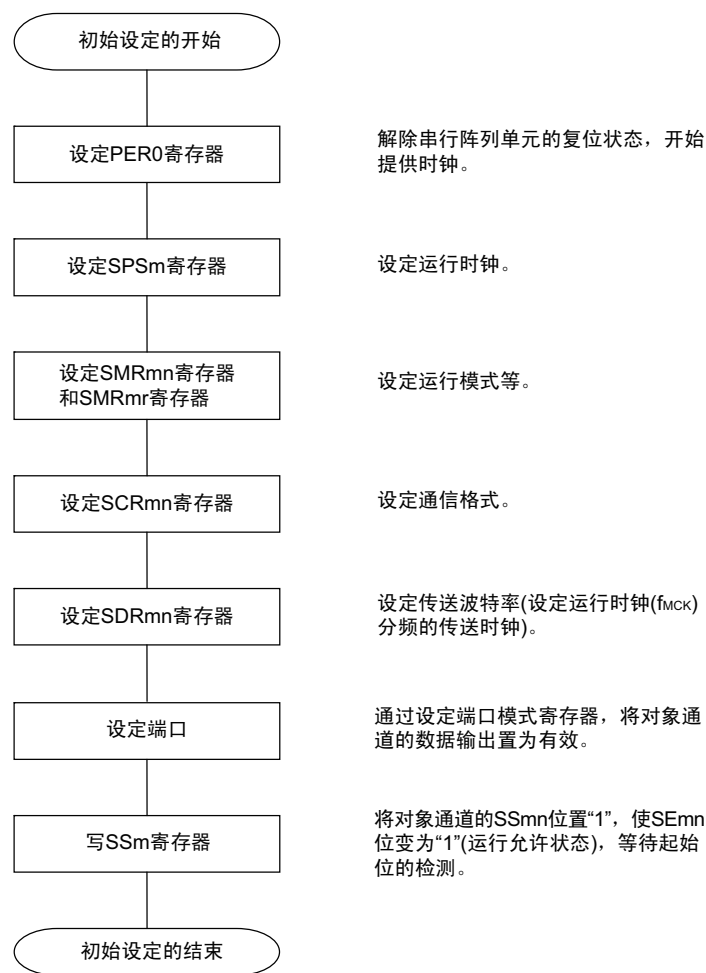
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
													0/1	×	0/1	×

备注 1. m: 单元号 (m=0、1)

2.  : 在 UART 接收模式中为固定设定。 : 不能设定 (设定初始值)。
 × : 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1 : 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 14-110 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 14-111 UART 接收的中止步骤

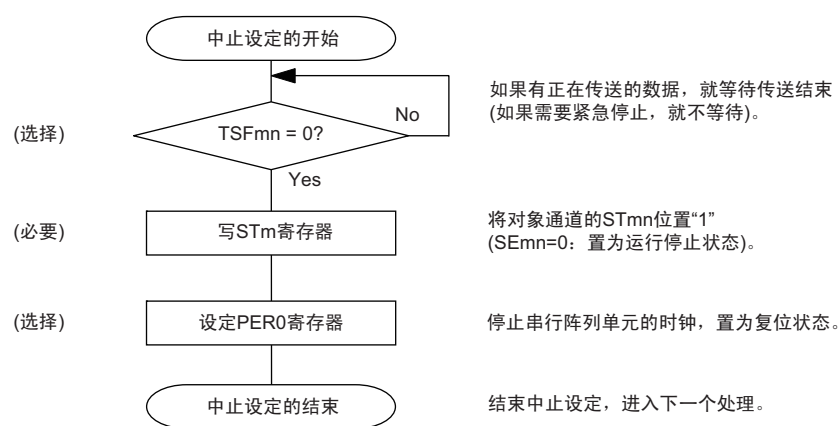
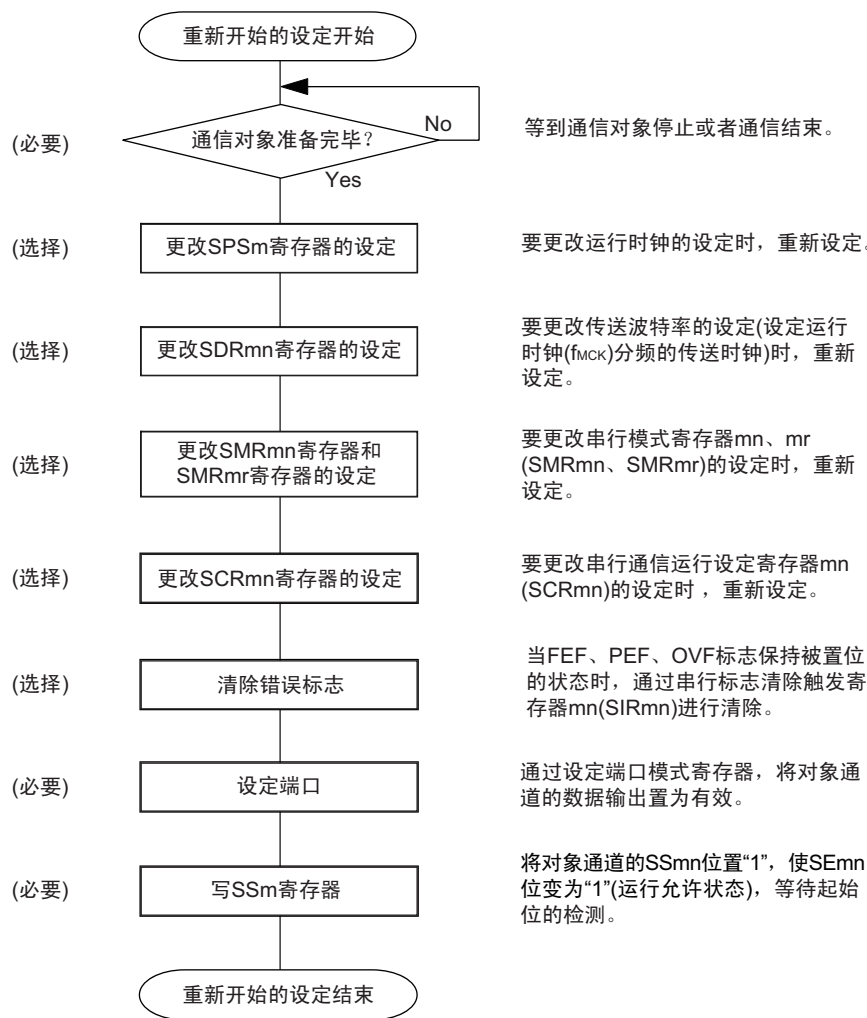


图 14-112 重新开始 UART 接收的设定步骤

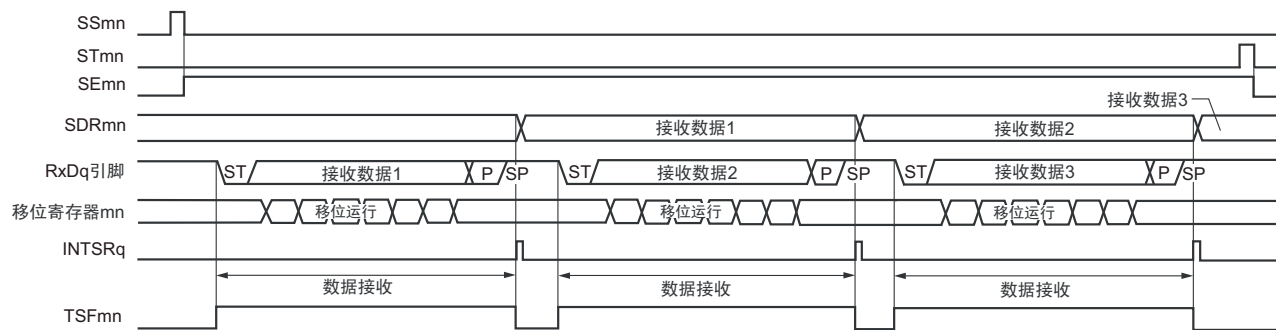


注意 必须在将 SCRmn 寄存器的 RXEmn 位置“1”后至少间隔 4 个 f_{MCK} 时钟，然后将 SSmn 位置“1”。

备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象停止或者通信结束后进行初始设定而不是进行重新开始设定。

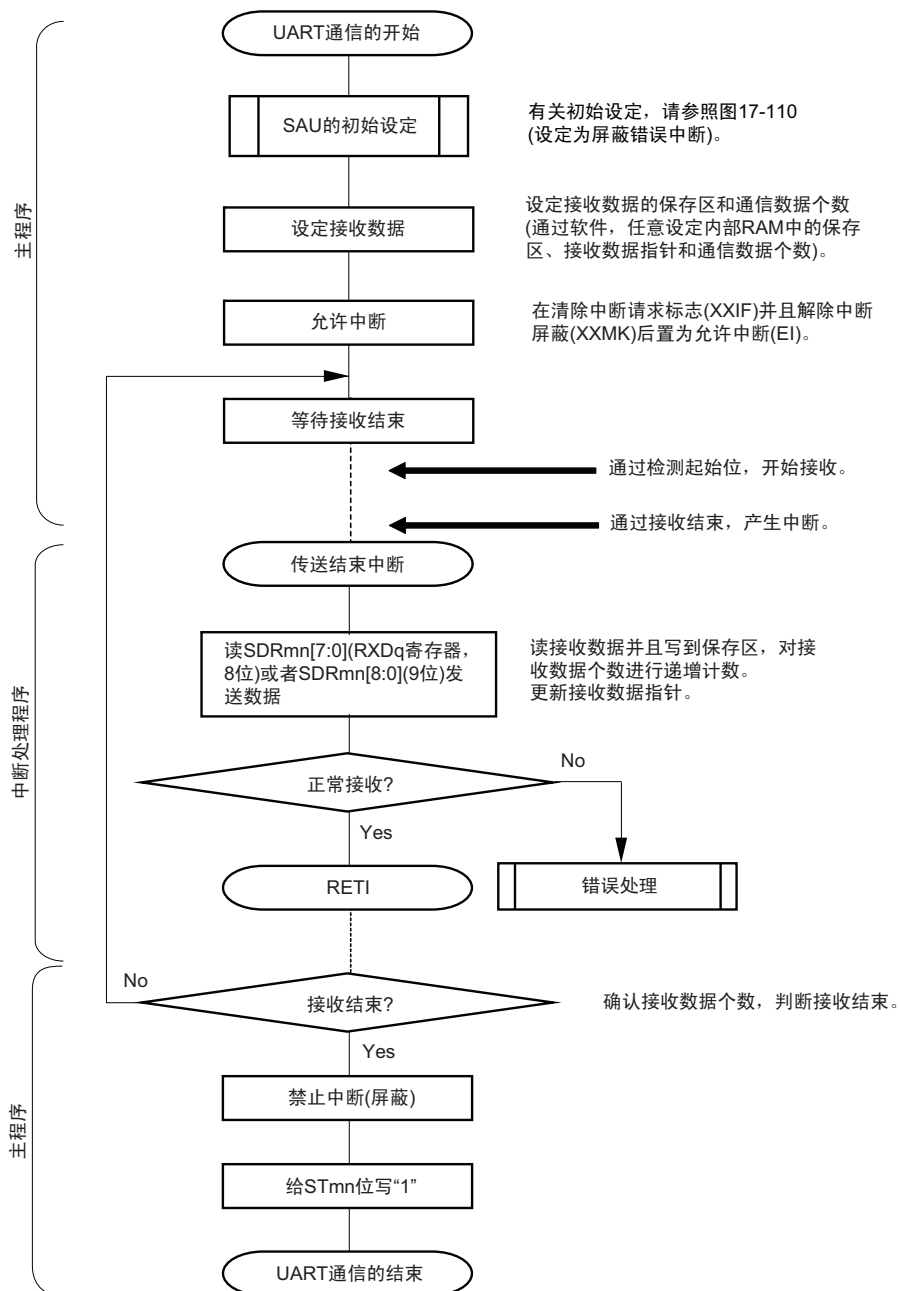
(3) 处理流程

图 14-113 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 14-114 UART 接收的流程图



14.7.3 SNOOZE 模式功能

这是在 STOP 模式中通过检测 RxDq 引脚的输入使 UART 进行接收的模式。通常，在 STOP 模式中 UART 停止通信，但是通过使用 SNOOZE 模式，能在 CPU 不运行的状态下进行 UART 的接收。当选项字节 (000C2H) 的 FRQSEL4 位为“0”时，只有以下的 UART 才能设定 SNOOZE 模式。

- 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 14-117 和图 14-119 SNOOZE 模式运行的流程图”）。

- 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 14-4，设定 SPSm 寄存器和 SDRmn 寄存器[15:9]。
- 设定 EOCmn 位和 SSECmn 位。能设定允许或者停止在发生通信错误时产生错误中断（INTSRE0）。
- 必须在即将要转移到 STOP 模式前将串行待机控制寄存器 m（SSCm）的 SWCm 位置“1”。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 的边沿（输入起始位），就开始 UART 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。

2. SNOOZE 模式中的传送速率只为 4800bps。
3. 如果将 SWCm 位置“1”，就只能在 STOP 模式中开始接收时使用 UARTq。当其他 SNOOZE 功能和中断同时使用并且在以下的非 STOP 模式状态下开始接收时，可能无法正常接收数据而发生帧错误或者奇偶校验错误。
 - 在将 SWCm 位置“1”后并且在转移到 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 SWCm 位置“0”前开始接收的情况
4. 如果将 SSECm 位置“1”，就在发生奇偶校验错误、帧错误或者溢出错误时不将 PEFmn、FEFmn、OVFmn 标志置位，也不产生错误中断（INTSREq）。因此，在 SSECm 位为“1”的情况下使用时，必须在将 SWC0 位置“1”前清除 PEFmn、FEFmn、OVFmn 标志并且读 SDRm1 寄存器的 bit7 ~ 0（RxDq）。

表 14-4 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
32MHz $\pm$ 1.0% 注	$f_{CLK}/2^5$	105	2.27%	-1.53%
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 $\pm 1.5\%$ 时，以下的容许范围会变窄。

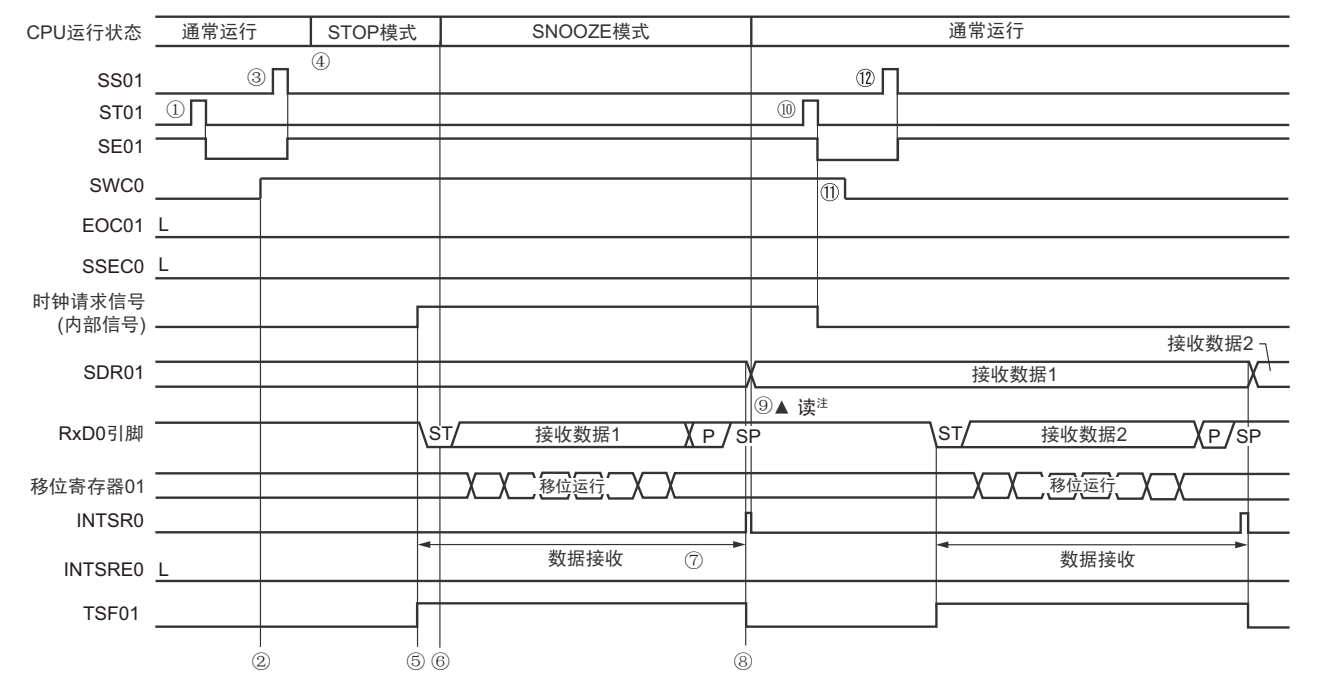
- 在 $f_{IH}\pm 1.5\%$ 的情况下，必须将上表的最大容许值设定为 -0.5%，并且将最小容许值设定为 +0.5%。

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、SSECm=0/1）

因为 EOCm1 位为“0”，所以与 SSECm 位的设定无关，即使发生通信错误也不产生错误中断（INTSREq）。但是，产生传送结束中断（INTSRq）。

图 14-115 SNOOZE 模式运行（EOCm1=0、SSECm=0/1）的时序图



注 必须在 SWCm 位为“1”的状态下读接收数据。

注意 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm1 位置“1”（清除 SEm1 位并且停止运行），而且还必须在接收结束后清除 SWCm 位（解除 SNOOZE 模式）。

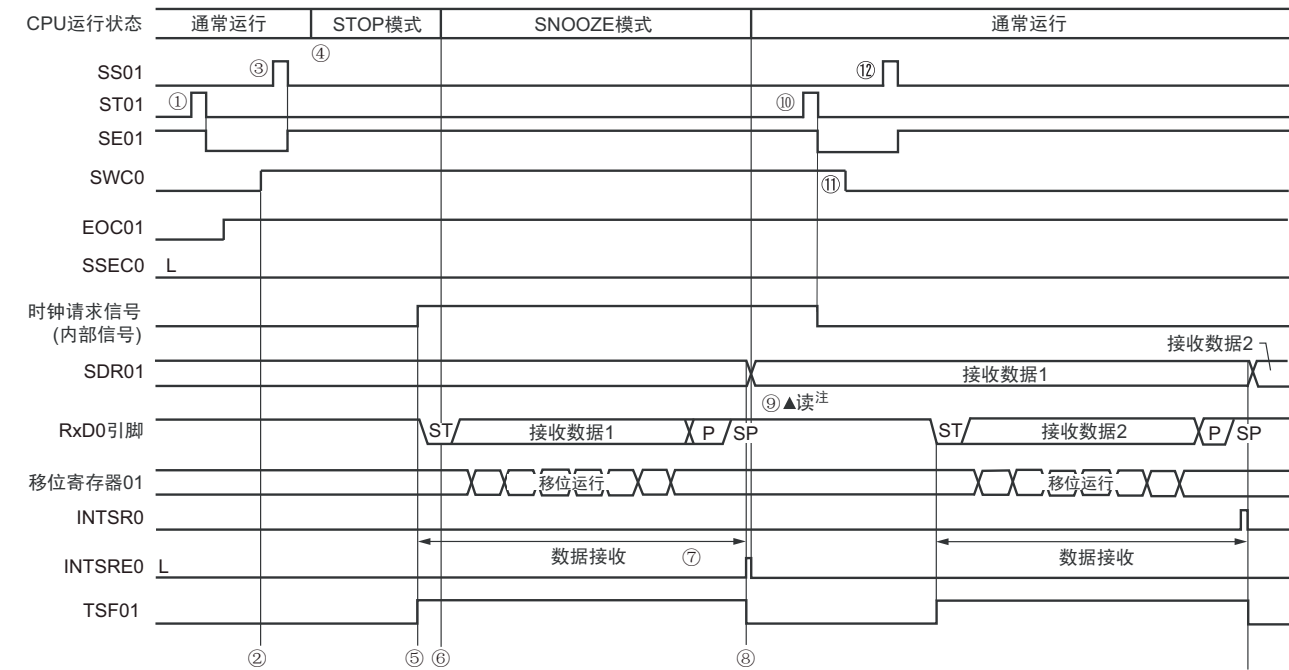
备注 1. 图中的①～⑫对应“图 14-117 SNOOZE 模式运行（EOCm1=0、SSECm=0/1 或者 EOCm1=1、SSECm=0）的流程图”中的①～⑫。

2. m=0，q=0

(2) SNOOZE 模式运行（EOCm1=1、SSECm=0：允许产生错误中断（INTSREq））

因为 EOCm1 位为“1”并且 SSECm 位为“0”，所以在发生通信错误时产生错误中断（INTSREq）。

图 14-116 SNOOZE 模式运行（EOCm1=1、SSECm=0）的时序图



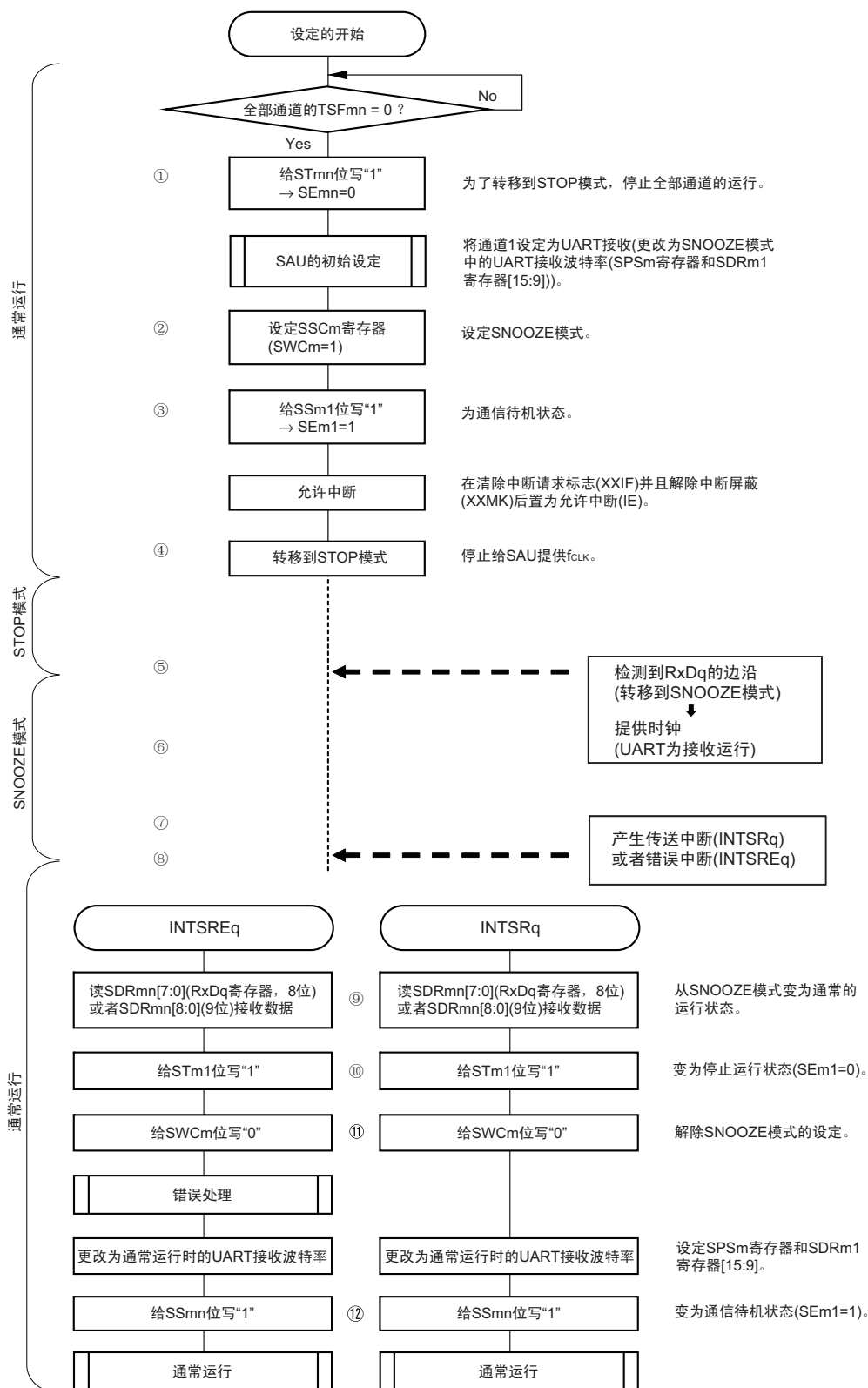
注 必须在 SWCm 位为“1”的状态下读接收数据。

注意 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm1 位置“1”（清除 SEm1 位并且停止运行），而且还必须在接收结束后清除 SWCm 位（解除 SNOOZE 模式）。

备注 1. 图中的①～⑫对应“图 14-117 SNOOZE 模式运行（EOCm1=0、SSECm=0/1 或者 EOCm1=1、SSECm=0）的流程图”中的①～⑫。

2. m=0, q=0

图 14-117 SNOOZE 模式运行 (EOCm1=0、SSECm=0/1 或者 EOCm1=1、SSECm=0) 的流程图



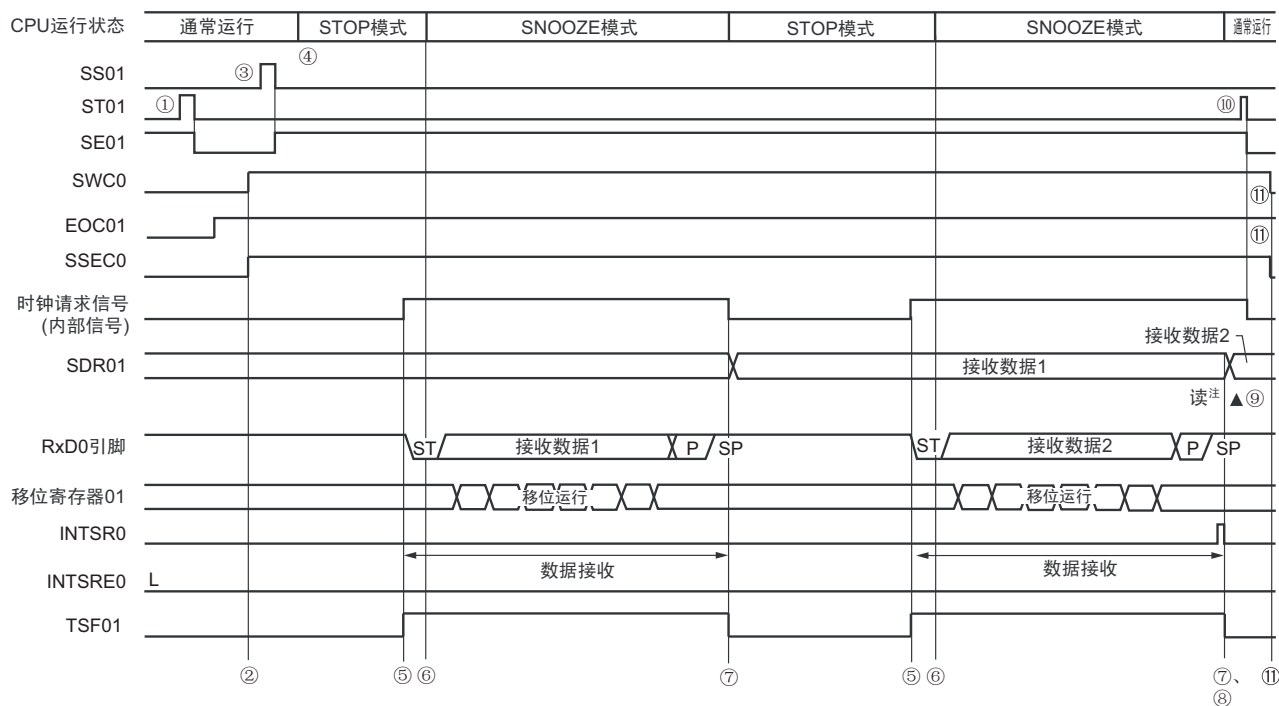
备注 1. 图中的①~⑫对应“图 14-115 SNOOZE 模式运行 (EOCm1=0、SSECm=0/1) 的时序图”和“图 14-116 SNOOZE 模式运行 (EOCm1=1、SSECm=0) 的时序图”中的①~⑫。

2. m=0, q=0

(3) SNOOZE 模式运行 (EOCm1=1、SSECM=1: 停止产生错误中断 (INTSREQ))

因为 EOCm1 位为“1”并且 SSECm 位为“1”，所以在发生通信错误时不产生错误中断（INTSREQ）。

图 14-118 SNOOZE 模式运行 (EOCm1=1、SSECm=1) 的时序图



注 必须在 SWCm 位为“1”的状态下读接收数据。

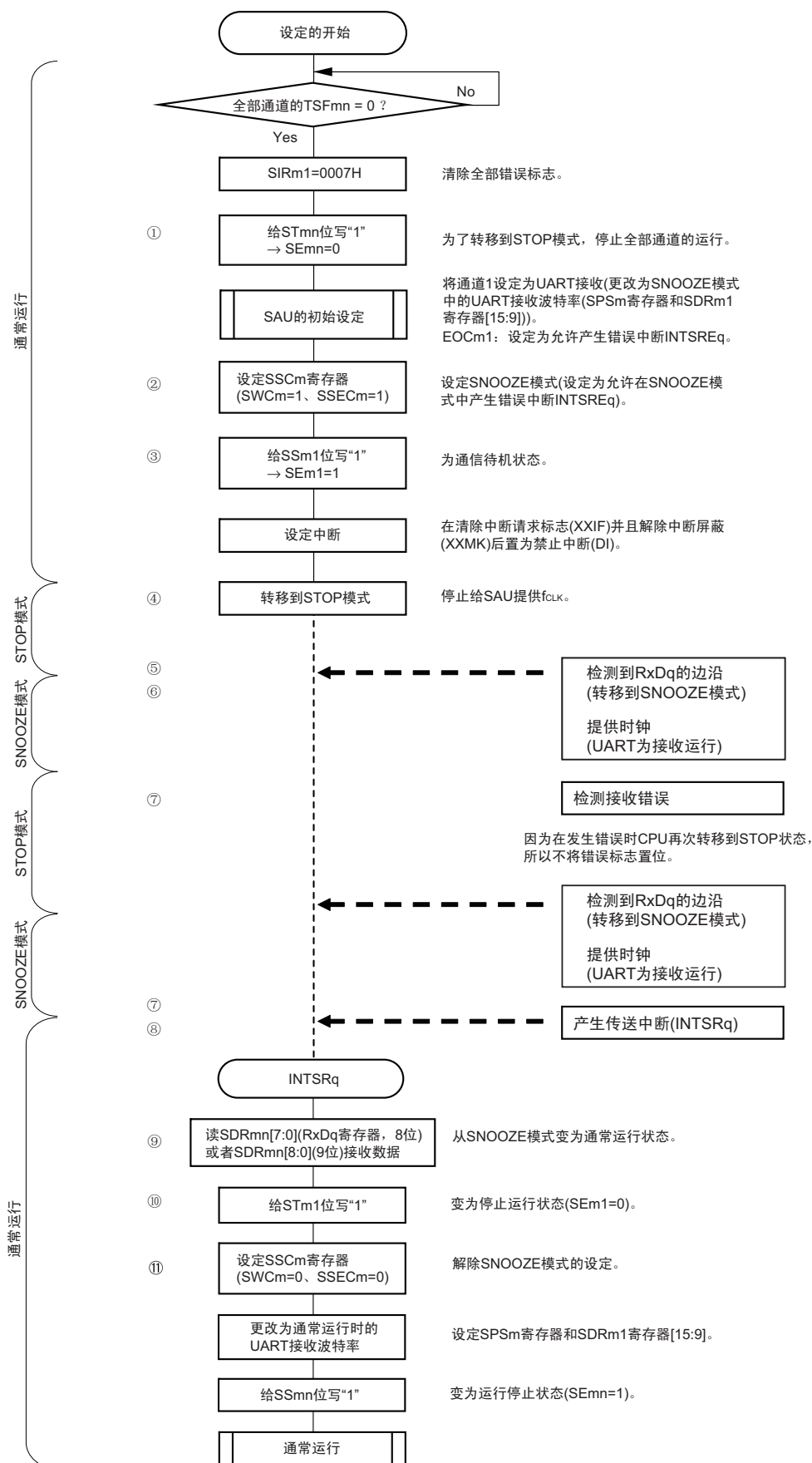
注意 1. 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm1 位置“1”（清除 SEm1 位并且停止运行），而且还必须在接收结束后清除 SWCm 位（解除 SNOOZE 模式）。

2. 如果 SSECm 位为“1”，就在发生奇偶校验错误、帧错误或者溢出错误时不将 PEFm1、FEFm1、OVFm1 标志置位，也不产生错误中断（INTSREq）。因此，在 SSECm 位为“1”的情况下使用时，必须在将 SWCm 位置“1”前清除 PEFm1、FEFm1、OVFm1 标志并且读 SDRm1[7:0]（RXDg 寄存器，8 位）或者 SDRm1[8:0]（9 位）。

备注 1. 图中的①~⑪对应“图 14-119 SNOOZE 模式运行 (EOCm1=1、SSECm=1) 的流程图”中的①~⑪。

2. $m=0, \quad q=0$

图 14-119 SNOOZE 模式运行 (EOCm1=1、SSECm=1) 的流程图



注意 如果 SSECm 位为“1”，就在发生奇偶校验错误、帧错误或者溢出错误时不将 PEFm1、FEFm1、OVFm1 标志置位，也不产生错误中断（INTSReq）。因此，在 SSECm 位为“1”的情况下使用时，必须在将 SWC0 位置“1”前清除 PEFm1、FEFm1、OVFm1 标志并且读 SDRm1[7:0]（RXDq 寄存器，8 位）或者 SDRm1[8:0]（9 位）。

备注 1. 图中的①～⑪对应“图 14-118 SNOOZE 模式运行（EOCm1=1、SSECm=1）的时序图”中的①～⑪。

2. m=0, q=0

14.7.4 波特率的计算

(1) 波特率的计算式

UART（UART0～UART2）通信的波特率能用以下计算式进行计算：

$$(\text{波特率}) = \{ \text{对象通道的运行时钟 (f}_{\text{MCK}}) \text{ 频率} \} \div (\text{SDRmn}[15:9] + 1) \div 2 [\text{bps}]$$

注意 禁止将串行数据寄存器 mn（SDRmn）的 SDRmn[15:9] 设定为“0000000B”和“0000001B”。

备注 1. 因为在使用 UART 时 SDRmn[15:9] 的值为 SDRmn 寄存器的 bit15～9 的值（0000010B～1111111B），所以为 2～127。

2. m：单元号（m=0、1） n：通道号（n=0～3） mn=00～03、10、11

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 14-5 UART 运行时钟的选择

SMRmn 寄存器	SPSm 寄存器								运行时钟 (f _{MCK}) 注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f _{CLK} =32MHz 运行时
0	X	X	X	X	0	0	0	0	f _{CLK}	32MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	1MHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	500kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	250kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	125kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	62.5kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	31.25kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	15.63kHz
	X	X	X	X	1	1	0	0	f _{CLK} /2 ¹²	7.81kHz
	X	X	X	X	1	1	0	1	f _{CLK} /2 ¹³	3.91kHz
	X	X	X	X	1	1	1	0	f _{CLK} /2 ¹⁴	1.95kHz
	X	X	X	X	1	1	1	1	f _{CLK} /2 ¹⁵	977Hz
1	0	0	0	0	X	X	X	X	f _{CLK}	32MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	16MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	8MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	4MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	2MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	1MHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	500kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	250kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	125kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	62.5kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	31.25kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	15.63kHz
	1	1	0	0	X	X	X	X	f _{CLK} /2 ¹²	7.81kHz
	1	1	0	1	X	X	X	X	f _{CLK} /2 ¹³	3.91kHz
	1	1	1	0	X	X	X	X	f _{CLK} /2 ¹⁴	1.95kHz
	1	1	1	1	X	X	X	X	f _{CLK} /2 ¹⁵	977Hz

注 要更改被选择为 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（ST_m）=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

(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。

$$(\text{波特率误差}) = (\text{波特率的计算值}) \div (\text{目标波特率的值}) \times 100 - 100 [\%]$$

$f_{\text{CLK}}=32\text{MHz}$ 时的 UART 波特率的设定例子如下所示。

UART 波特率 (目标波特率)	$f_{\text{CLK}}=32\text{MHz}$			
	运行时钟 (f_{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	$\pm 0.39\%$

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“14.7.4(1) 波特率的计算式”)

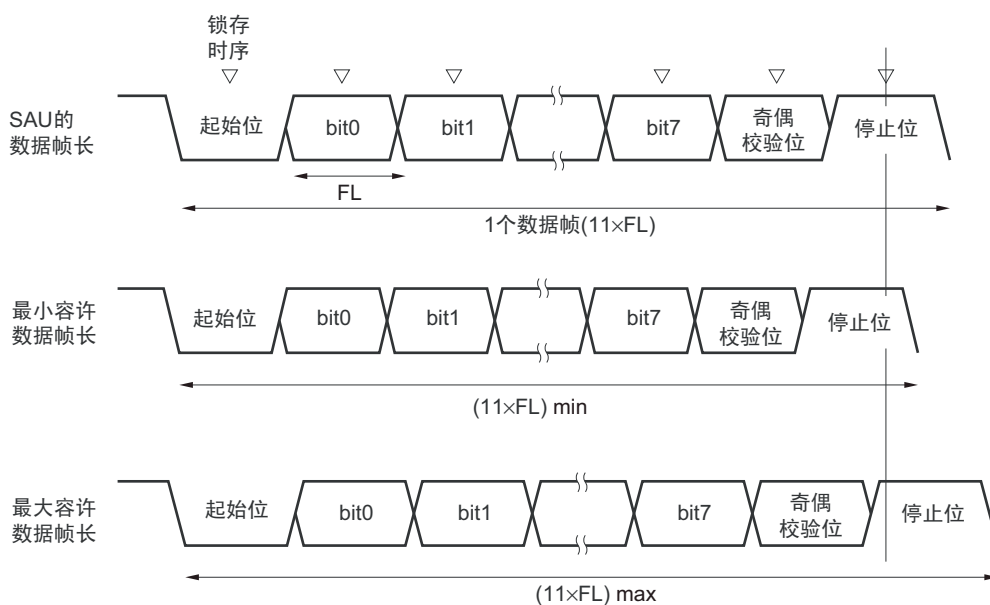
k : SDRmn[15:9]+1

Nfr : 1 个数据帧的帧长 [位]

= (起始位) + (数据长度) + (奇偶校验位) + (停止位)

备注 m: 单元号 (m=0、1) n: 通道号 (n=1、3) mn=01、03、11、13

图 14-120 接收时的波特率容许范围 (1 个数据帧的帧长 =11 位的情况)



如图 14-120 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。

14.7.5 在 UART（UART0～UART2）通信过程中发生错误时的处理步骤

在 UART（UART0～UART2）通信过程中发生错误时的处理步骤如图 14-121 和图 14-122 所示。

图 14-121 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 14-122 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 为停止运行状态。	
与通信方进行同步处理。		因为起始位偏移，所以可认为发生了帧错误。因此，需要与通信方重新取得同步，重新开始通信。
将串行通道开始寄存器 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、12

14.8 LIN 通信的运行

14.8.1 LIN 发送

在 UART 发送中，UART0 支持 LIN 通信。
LIN 发送使用单元 0 的通道 0。

UART	UART0	UART1	UART2
LIN 通信的支持	能	不能	不能
对象通道	SAU0 的通道 0	—	—
使用的引脚	TxD0	—	—
中断	INTST0	—	—
	可选择传送结束中断（单次发送模式）或者缓冲器空中断（连续发送模式）。		
错误检测标志	无		
传送数据长度	8 位		
传送速率注	Max. $f_{MCK}/6[bps]$ （ $SDR00[15:9] \geq 2$ ）、Min. $f_{CLK}/(2 \times 2^{15} \times 128)[bps]$		
数据相位	不反相输出（默认值：高电平）。 反相输出（默认值：低电平）。		
奇偶校验位	无奇偶校验位。		
停止位	附加 1 位。		
数据方向	MSB 优先		

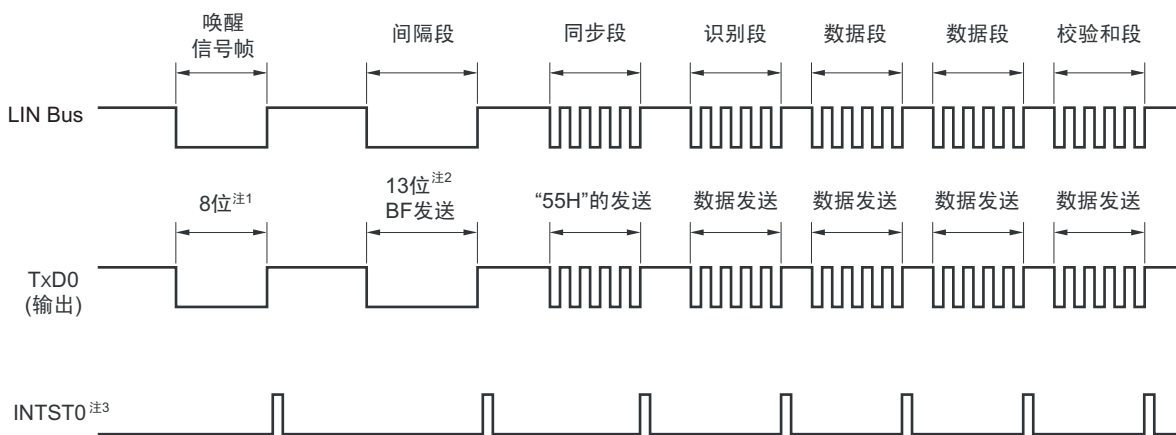
注 必须在满足此条件并且满足电特性的外围功能特性（参照“第 31 章 电特性”）的范围内使用，并且在 LIN 通信中经常使用 2.4/9.6/19.2kbps。

备注 f_{MCK} ：对象通道的运行时钟频率
 f_{CLK} ：系统时钟频率

LIN 是 Local Interconnect Network 的简称，是为降低汽车网络成本的低速（1 ~ 20kbps）串行通信协议。
LIN 通信是单主控通信，一个主控设备最多可连接 15 台从属设备。
LIN 从属设备用于开关、传动装置和传感器等的控制，这些装置通过 LIN 连接到主控设备。
LIN 主控一般连接 CAN（Controller Area Network）等的网络。
LIN 总线是单线方式的总线，通过符合 ISO9141 的收发器连接各节点。
根据 LIN 协议，主控设备发送附加波特率信息的帧，从属设备接收此帧并且校正与主控设备的波特率误差。
因此，如果从属设备的波特率误差不大于 ±15%，就能进行通信。

LIN 的发送操作的概要如图 14-123 所示。

图 14-123 LIN 的发送操作



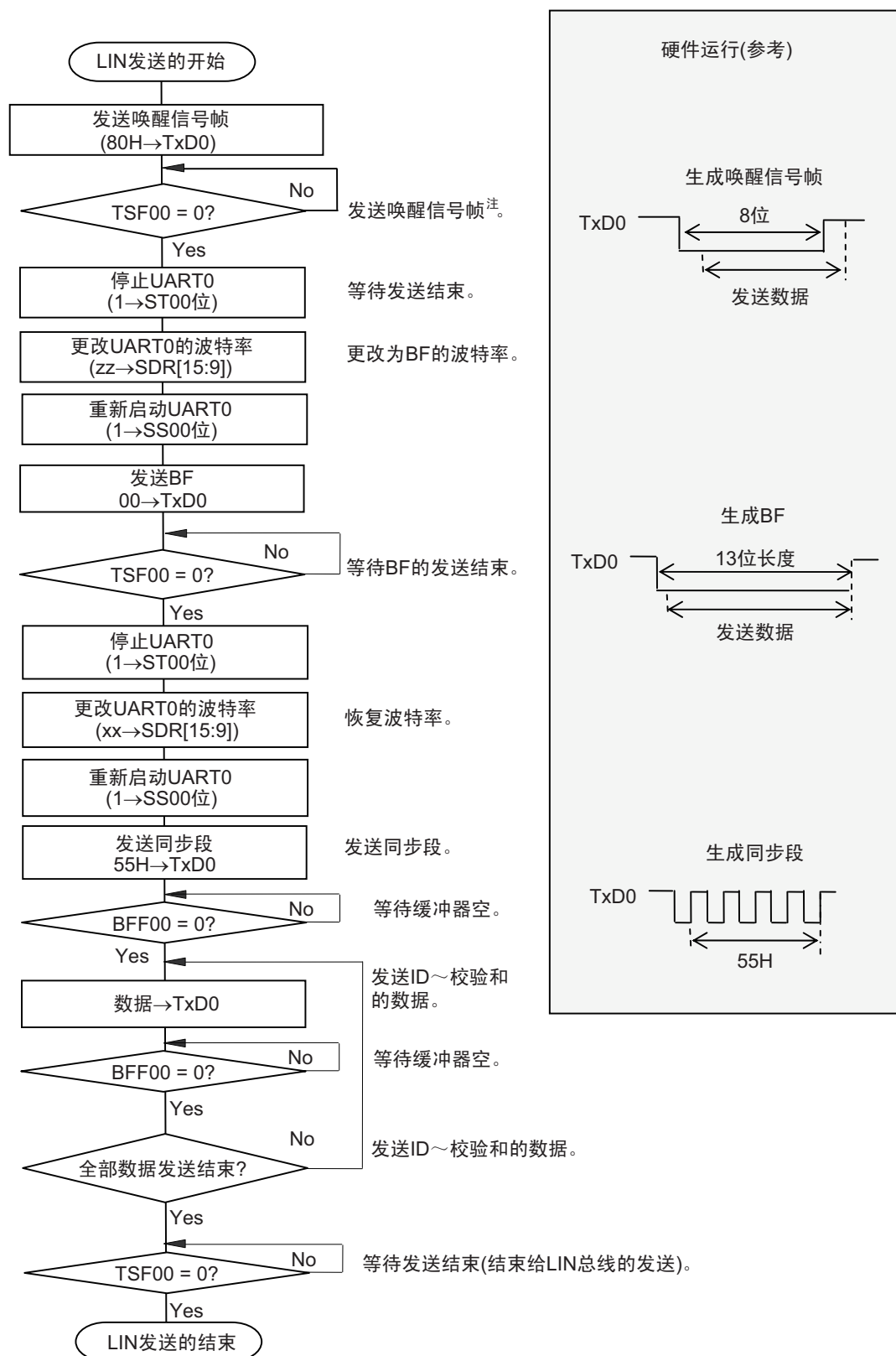
- 注 1. 为了满足唤醒信号的规定，设定波特率并且通过发送“80H”的数据进行对应。
2. 间隔段规定为 13 位宽的低电平输出，因此假设主传送使用的波特率为 N[bps]，间隔段使用的波特率如下：

$$(\text{间隔段的波特率}) = 9/13 \times N$$

- 通过此波特率发送“00H”的数据，生成间隔段。
3. 在各数据发送结束时，输出 INTST0。在 BF 发送时，也输出 INTST0。

备注 由软件控制各段间的间隔。

图 14-124 LIN 发送的流程图



注 只限于从 LIN-bus 睡眠状态启动的情况。

备注 这是从结束 UART 的初始设定并且允许从属发送开始的流程。

14.8.2 LIN 接收

在 UART 接收中，UART0 支持 LIN 通信。

LIN 接收使用单元 0 的通道 1。

UART	UART0	UART1	UART2
LIN 通信的支持	能	不能	不能
对象通道	SAU0 的通道 1	—	—
使用的引脚	RxD0	—	—
中断	INTSR0	—	—
	只限于传送结束中断（禁止设定缓冲器空中断）。		
错误中断	INTSRE0	—	—
错误检测标志	• 帧错误检测标志（FEF01） • 溢出错误检测标志（OVF01）		
传送数据长度	8 位		
传送速率注	Max. $f_{MCK}/6[\text{bps}]$ ($\text{SDR01}[15:9] \geq 2$)、Min. $f_{CLK}/(2 \times 2^{15} \times 128)[\text{bps}]$		
数据相位	不反相输出（默认值：高电平）。 反相输出（默认值：低电平）。		
奇偶校验位	无奇偶校验位（不检查奇偶校验）。		
停止位	附加 1 位。		
数据方向	LSB 优先		

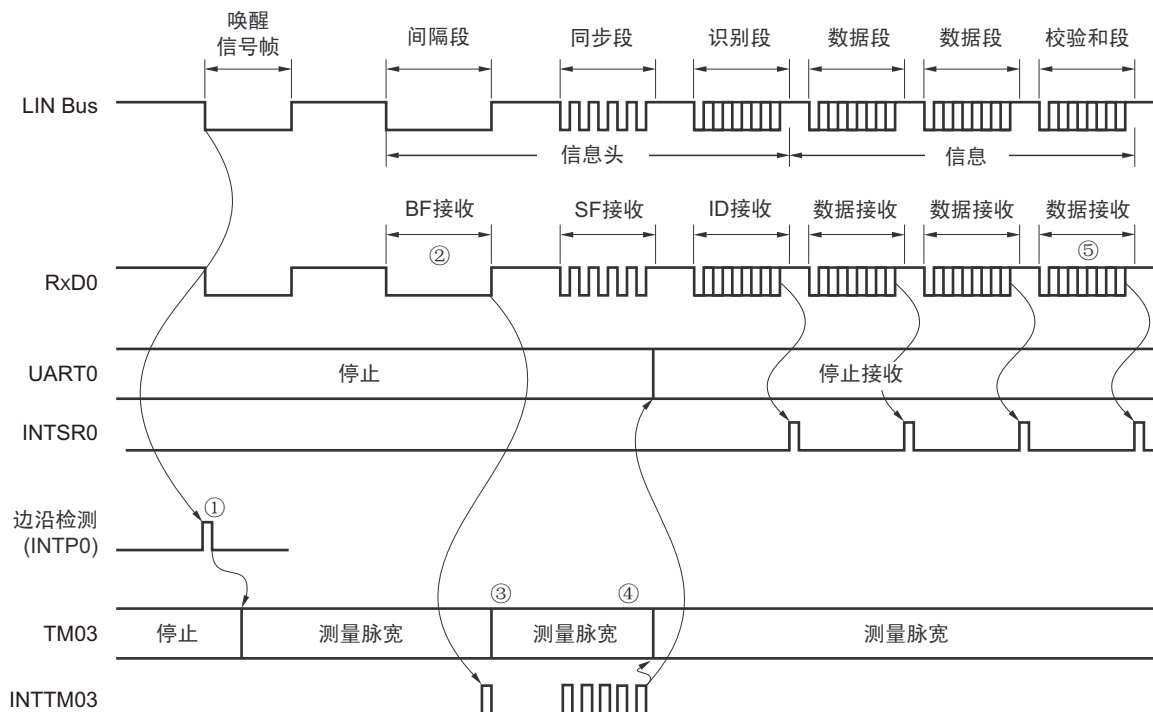
注 必须在满足此条件并且满足电特性的外围功能特性（参照“第 31 章 电特性”）的范围内使用。

备注 f_{MCK} ：对象通道的运行时钟频率

f_{CLK} ：系统时钟频率

LIN 的接收操作的概要如图 14-125 所示。

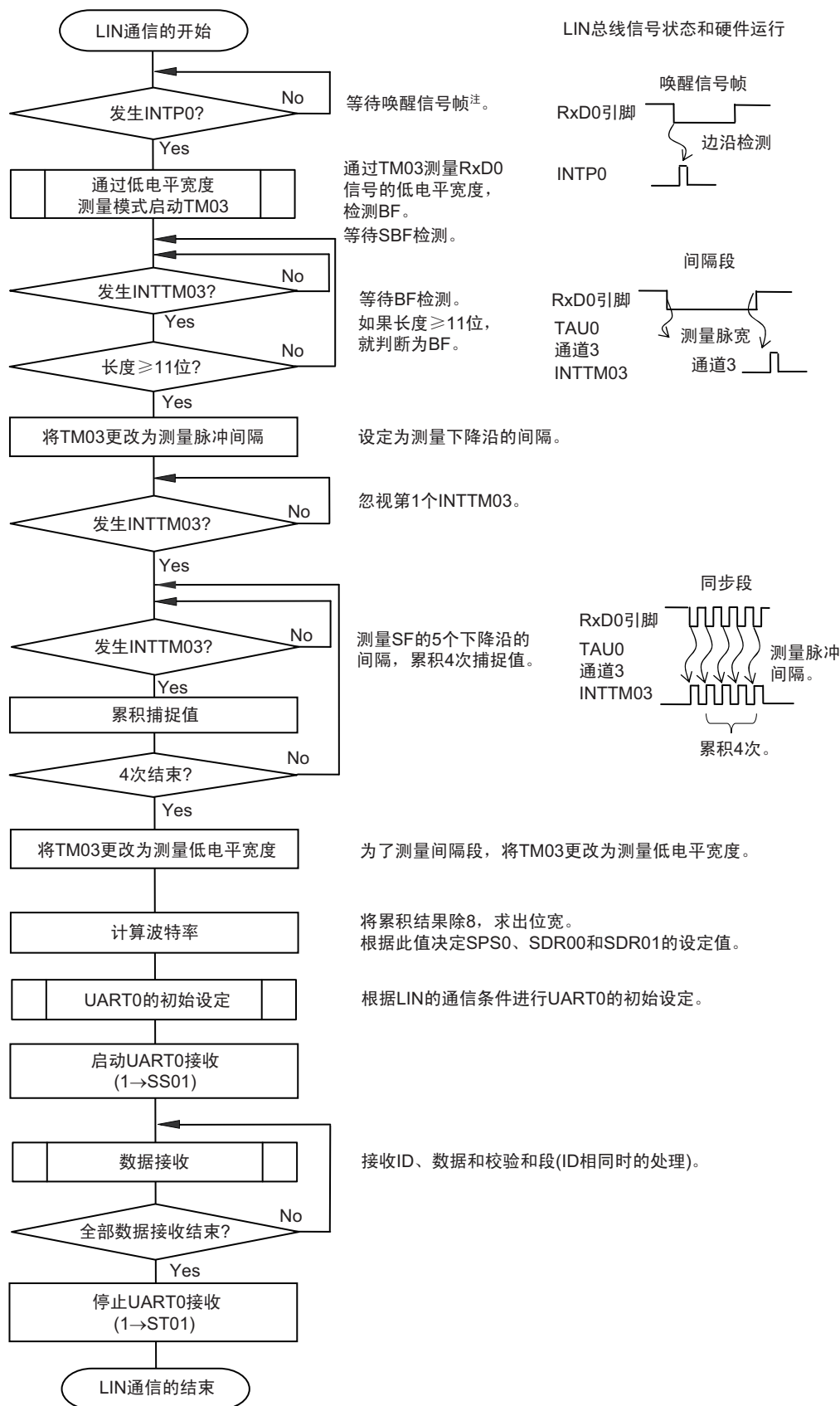
图 14-125 LIN 的接收操作



信号处理的流程如下：

- ① 通过检测引脚的中断边沿 (INTP0) 来检测唤醒信号。当检测到唤醒信号时，为了测量 BF 的低电平宽度，将 TM03 设定为测量脉宽，然后进入 BF 接收等待状态。
- ② 如果检测到 BF 的下降沿，TM03 就开始测量低电平的宽度，并且在 BF 的上升沿进行捕捉。根据捕捉到的值判断是否为 BF 信号。
- ③ 当 BF 接收正常结束时，必须将 TM03 设定为测量脉冲间隔，并且测量 4 次同步段的 RxD0 信号下降沿的间隔（参照“6.8.4 作为输入脉冲间隔测量的运行”）。
- ④ 根据同步段 (SF) 的位间隔计算波特率误差。然后，必须在暂停 UART0 运行后调整（重新设定）波特率。
- ⑤ 必须通过软件区分校验和段。还必须通过软件在接收校验和段后对 UART0 进行初始化并且再次设定为 BF 接收等待状态。

图 14-126 LIN 接收的流程图



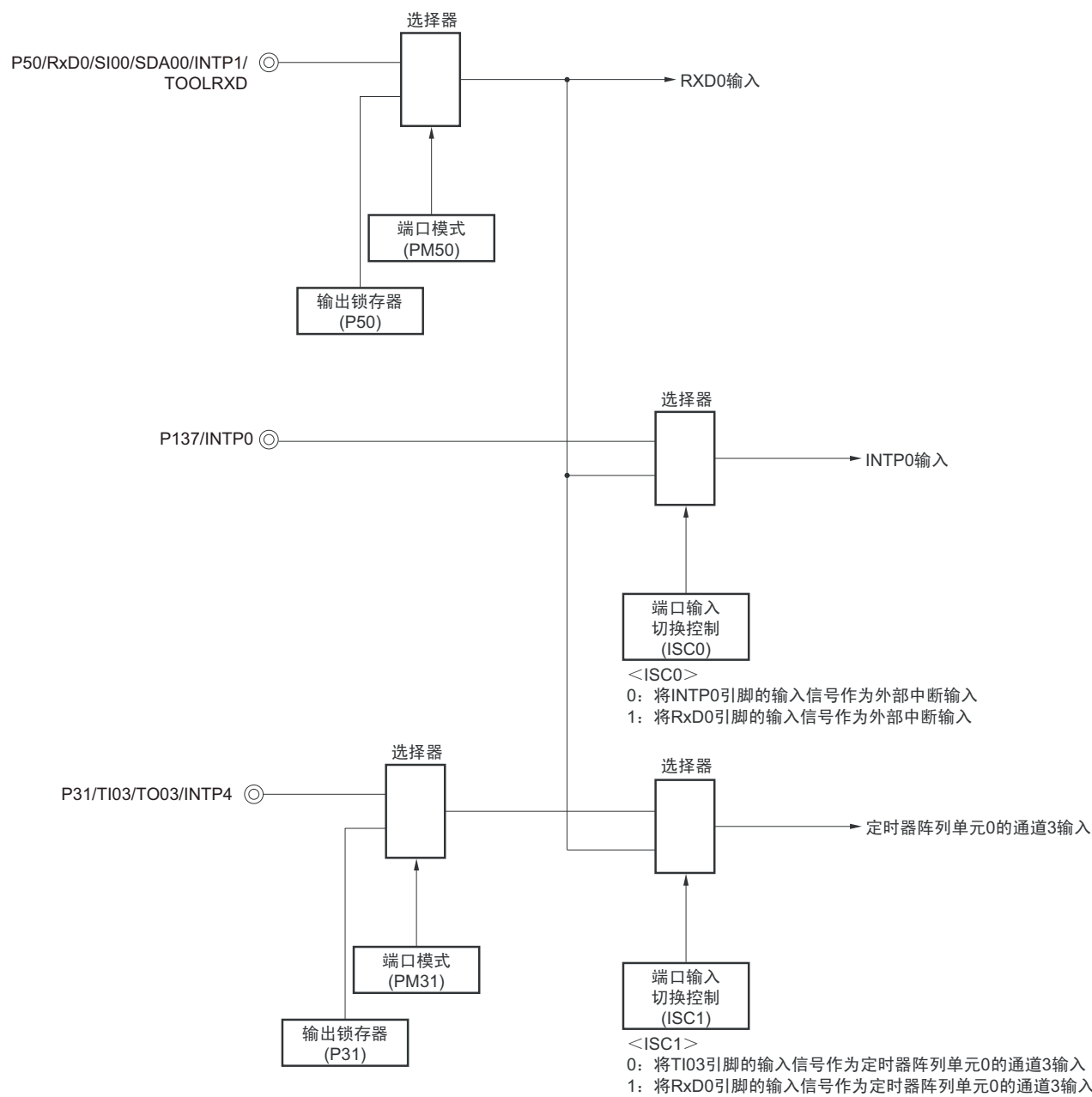
注 只在睡眠状态下才需要。

用于 LIN 接收操作的端口结构图如图 14-127 所示。

通过外部中断（INTP0）的边沿检测，接收 LIN 主控发送的唤醒信号。能通过定时器阵列单元 0 的外部事件捕捉运行，测量 LIN 主控发送的同步段的长度以及计算波特率误差。

通过端口输入切换控制（ISC0/ISC1），能不在外部连线而将用于接收的端口输入（RxD0）的输入源输入到外部中断（INTP0）和定时器阵列单元。

图 14-127 用于 LIN 接收操作的端口结构图



备注 ISC0、ISC1：输入切换控制寄存器（ISC）的 bit0 和 bit1（参照图 14-21）

用于 LIN 通信运行的外围功能总结如下：

<使用的外围功能>

- 外部中断（INTP0）：唤醒信号检测
用途：检测唤醒信号的边沿和通信的开始。
- 定时器阵列单元的通道3：波特率误差检测、间隔段（BF）检测
用途：检测同步段（SF）的长度，并且通过将其长度除以位数来检测波特率误差（通过捕捉模式测量 RxD0 输入边沿的间隔）。
测量低电平宽度，判断是否为间隔段（BF）。
- 串行阵列单元0（SAU0）的通道0和通道1（UART0）

14.9 简易 I²C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21) 通信的运行

这是通过串行时钟 (SCL) 和串行数据 (SDA) 共 2 条线与多个设备进行时钟同步通信的功能。因此简易 I²C 是为了与 EEPROM、闪存、A/D 转换器等设备进行单通信而设计的, 所以只用作主控设备。

对于开始条件和停止条件, 必须遵守 AC 规格, 在操作控制寄存器的同时通过软件进行处理。

[数据的发送和接收]

- 主控发送、主控接收 (只限于单主控的主控功能)
- ACK 输出功能注、ACK 检测功能
- 8 位数据长度 (在发送地址时, 用高 7 位指定地址, 用最低位进行 R/W 控制)
- 通过软件产生开始条件和停止条件。

[中断功能]

- 传送结束中断

[错误检测标志]

- 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 “14.9.3(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、11

SAU0 的通道 0 ~ 3 以及 SAU1 的通道 0 和通道 1 是支持简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）的通道。

○ 32 引脚产品

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00 (支持从属选择输入功能)	UART0 (支持 LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

○ 64 引脚产品

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00 (支持从属选择输入功能)	UART0 (支持 LIN-bus)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）有以下 4 种通信运行：

- 地址段发送 (参照 14.9.1)
- 数据发送 (参照 14.9.2)
- 数据接收 (参照 14.9.3)
- 停止条件的产生 (参照 14.9.4)

14.9.1 地址段的发送

地址段的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	IIC01	IIC10	IIC11	IIC20	IIC21
对象通道	SAU0 的通道 0	SAU0 的通道 1	SAU0 的通道 2	SAU0 的通道 3	SAU1 的通道 0	SAU1 的通道 1
使用的引脚	SCL00、 SDA00 注 1	SCL01、 SDA01 注 1	SCL10、 SDA10 注 1	SCL11、 SDA11 注 1	SCL20、 SDA20 注 1	SCL21、 SDA21 注 1
中断	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	只限于传送结束中断（不能选择缓冲器空中断）。					
错误检测标志	ACK 错误检测标志（PEFmn）					
传送数据长度	8 位（将高 7 位作为地址并且将低 1 位作为 R/W 控制进行发送）					
传送速率注 2	Max. $f_{MCK}/4[Hz]$ ($SDRmn[15:9] \geq 1$) 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： • 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	不反相输出（默认值：高电平）。					
奇偶校验位	无奇偶校验位。					
停止位	附加 1 位（用于 ACK 接收）。					
数据方向	MSB 优先					

注 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（32 引脚产品）/EV_{DD} 耐压（64 引脚产品））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 使用复用功能时的端口相关寄存器的设定”。

要将 IIC00、IIC10、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、SCL10、SCL20）设定为 N 沟道漏极开路输出（V_{DD} 耐压（32 引脚产品）/EV_{DD} 耐压（64 引脚产品））模式（POMxx=1）。详细内容请参照“4.4.4 通过 EV_{DD} ≤ V_{DD} 进行的不同电位（1.8V、2.5V、3V）的对应”。

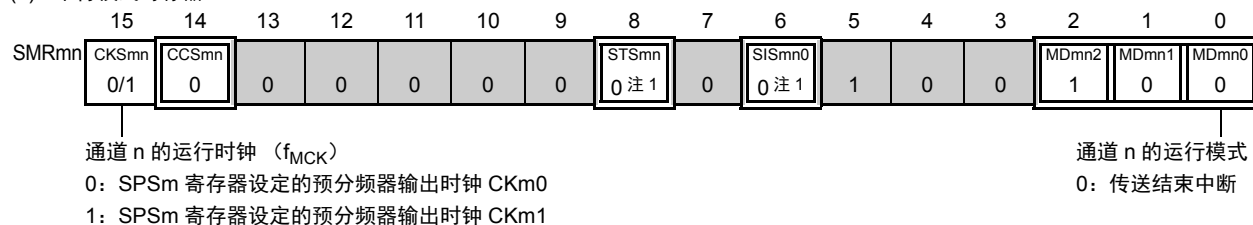
2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 31 章 电特性”）的范围内使用。

备注 m: 单元号（m=0、1） n: 通道号（n=0～3） mn=00～03、10、11

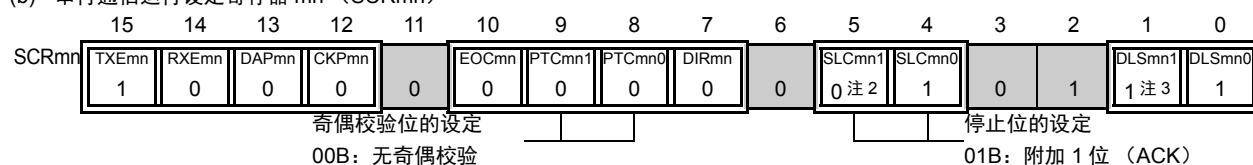
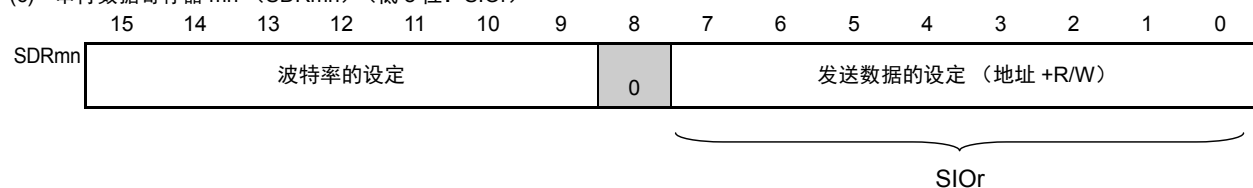
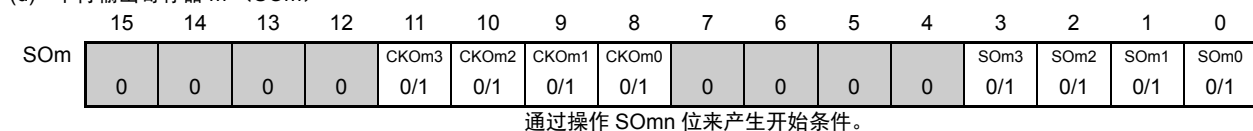
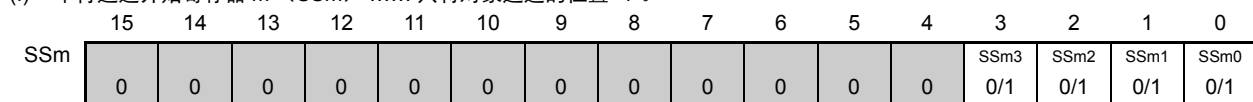
(1) 寄存器的设定

图 14-128 简易 I²C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21) 地址段发送时的寄存器设定内容例子

(a) 串行模式寄存器 mn (SMRmn)



(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: SIO_r)(d) 串行输出寄存器 m (SOM_m)(e) 串行输出允许寄存器 m (SOEm_m)(f) 串行通道开始寄存器 m (SSm_m) 只将对象通道的位置“1”。

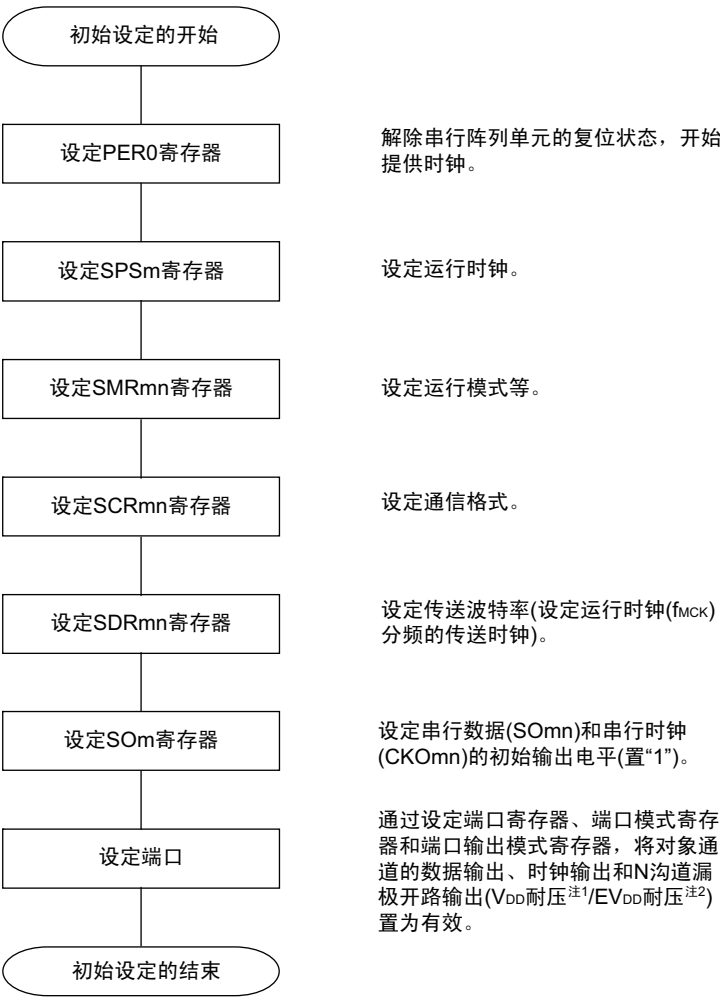
- 注 1. 只限于 SMR00 寄存器和 SMR11 寄存器。
 2. 只限于 SCR00、SCR02、SCR10 寄存器。
 3. 只限于 SCR00 寄存器和 SCR01 寄存器，其他固定为“1”。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0~3) r: IIC 号 (r=00、01、10、11、20、21)
 mn=00~03、10、11

2. : 在 IIC 模式中为固定设定。 : 不能设定 (设定初始值)。
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 14-129 地址段发送的初始设定步骤

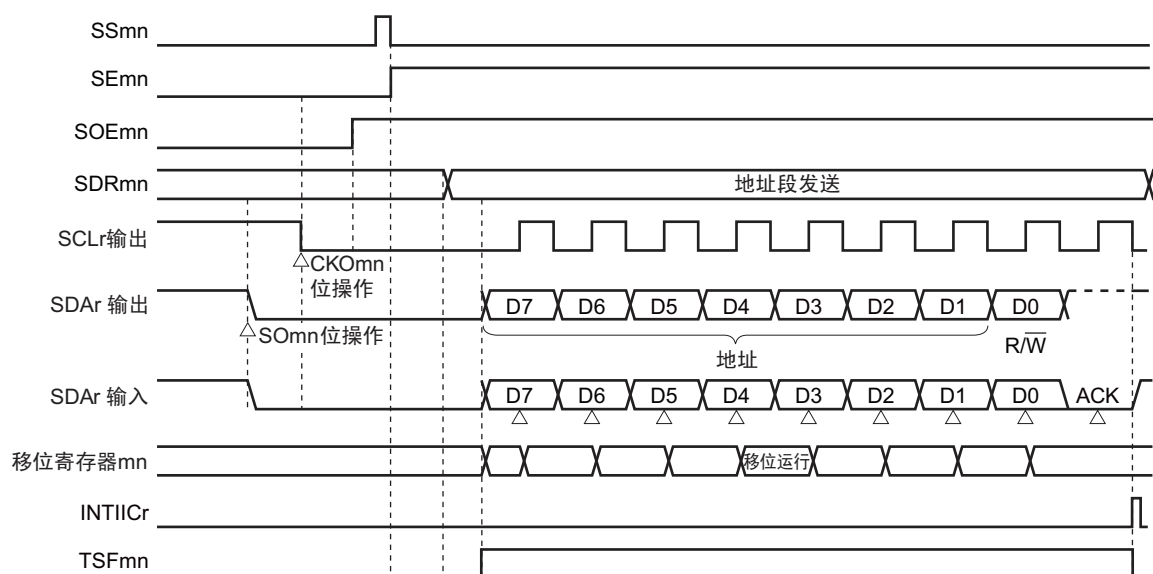


注 1. 这是 32 引脚产品的情况。
2. 这是 64 引脚产品的情况。

备注 在初始设定结束时，简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）为禁止输出并且处于运行停止状态。

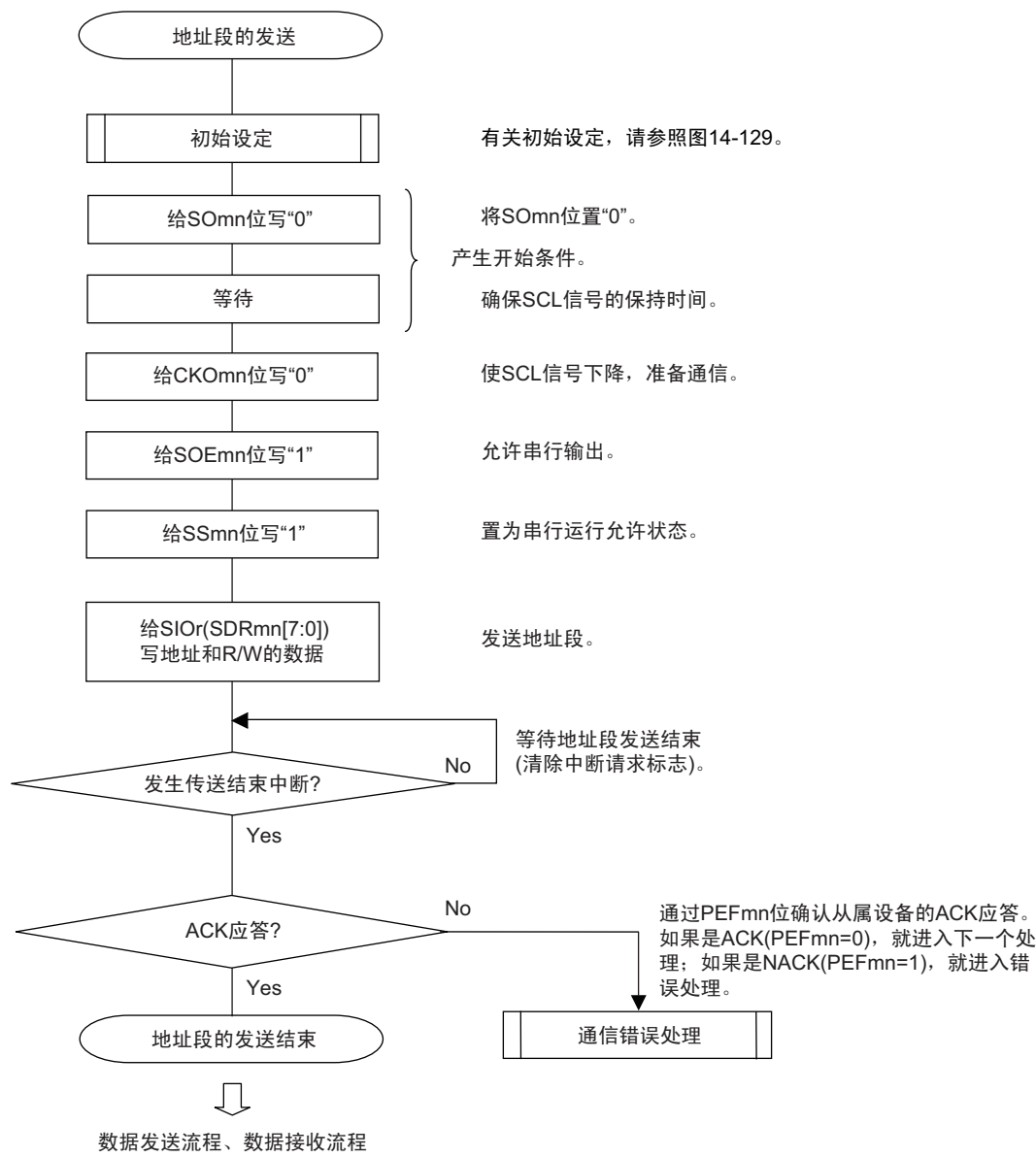
(3) 处理流程

图 14-130 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、01、10、11、20、21)
mn=00 ~ 03、10、11

图 14-131 地址段发送的流程图



14.9.2 数据发送

数据发送是在发送地址段后给该传送对象（从属设备）发送数据的运行。在给对象从属设备发送全部数据后产生停止条件并且释放总线。

简易 I ² C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
对象通道	SAU0 的通道 0	SAU0 的通道 1	SAU0 的通道 2	SAU0 的通道 3	SAU1 的通道 0	SAU1 的通道 1
使用的引脚	SCL00、 SDA00 注 1	SCL01、 SDA01 注 1	SCL10、 SDA10 注 1	SCL11、 SDA11 注 1	SCL20、 SDA20 注 1	SCL21、 SDA21 注 1
中断	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	只限于传送结束中断（不能选择缓冲器空中断）。					
错误检测标志	ACK 错误标志（PEFmn）					
传送数据长度	8 位					
传送速率注 2	Max. $f_{MCK}/4[\text{Hz}]$ ($\text{SDRmn}[15:9] \geq 1$) 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: • 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	不反相输出（默认值: 高电平）。					
奇偶校验位	无奇偶校验位。					
停止位	附加 1 位（用于 ACK 接收）。					
数据方向	MSB 优先					

注 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（32 引脚产品）/EV_{DD} 耐压（64 引脚产品））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 使用复用功能时的端口相关寄存器的设定”。

要将 IIC00、IIC10、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、SCL10、SCL20）设定为 N 沟道漏极开路输出（V_{DD} 耐压（32 引脚产品）/EV_{DD} 耐压（64 引脚产品））模式（POMxx=1）。详细内容请参照“4.4.4 通过 EV_{DD} ≤ V_{DD} 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 31 章 电特性”）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、11

(1) 寄存器的设定

图 14-132 简易 I²C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21) 数据发送时的寄存器设定内容例子

(a) 串行模式寄存器 mn (SMRmn) 在数据发送和接收的过程中不操作此寄存器。

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

(2) 处理流程

图 14-133 数据发送的时序图

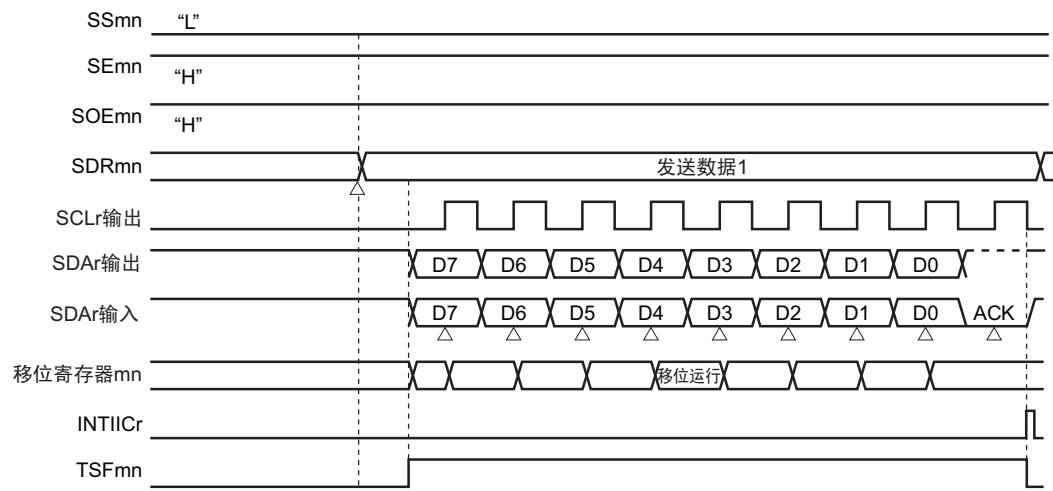
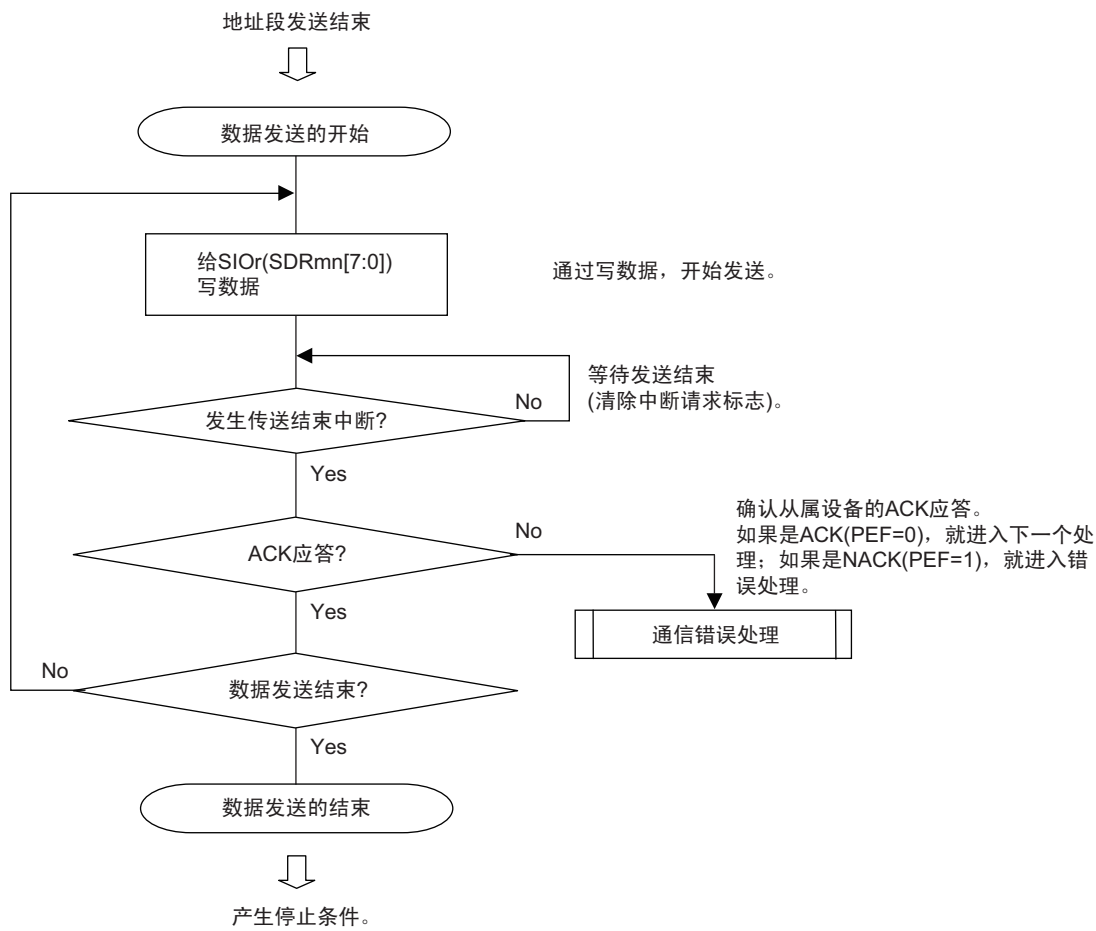


图 14-134 数据发送的流程图



14.9.3 数据接收

数据接收是在发送地址段后从传送对象（从属设备）接收数据的运行。在从对象从属设备接收全部数据后产生停止条件并且释放总线。

简易 I ² C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
对象通道	SAU0 的通道 0	SAU0 的通道 1	SAU0 的通道 2	SAU0 的通道 3	SAU1 的通道 0	SAU1 的通道 1
使用的引脚	SCL00、 SDA00 注 1	SCL01、 SDA01 注 1	SCL10、 SDA10 注 1	SCL11、 SDA11 注 1	SCL20、 SDA20 注 1	SCL21、 SDA21 注 1
中断	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	只限于传送结束中断（不能选择缓冲器空中断）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	8 位					
传送速率注 2	Max. $f_{MCK}/4$ [Hz] ($SDRmn[15:9] \geq 1$) 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： • 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	不反相输出（默认值：高电平）。					
奇偶校验位	无奇偶校验位。					
停止位	附加 1 位（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（32 引脚产品）/EV_{DD} 耐压（64 引脚产品））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 使用复用功能时的端口相关寄存器的设定”。

要将 IIC00、IIC10、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、SCL10、SCL20）设定为 N 沟道漏极开路输出（V_{DD} 耐压（32 引脚产品）/EV_{DD} 耐压（64 引脚产品））模式（POMxx=1）。详细内容请参照“4.4.4 通过 EV_{DD} ≤ V_{DD} 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 31 章 电特性”）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、11

(1) 寄存器的设定

图 14-135 简易 I²C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21) 数据接收时的寄存器设定内容例子

(a) 串行模式寄存器 mn (SMRmn) 在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	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		
	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					
	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
													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																
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
													SOEm3	SOEm2	SOEm1	SOEm0
													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																
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
													SSm3	SSm2	SSm1	SSm0
													0/1	0/1	0/1	0/1

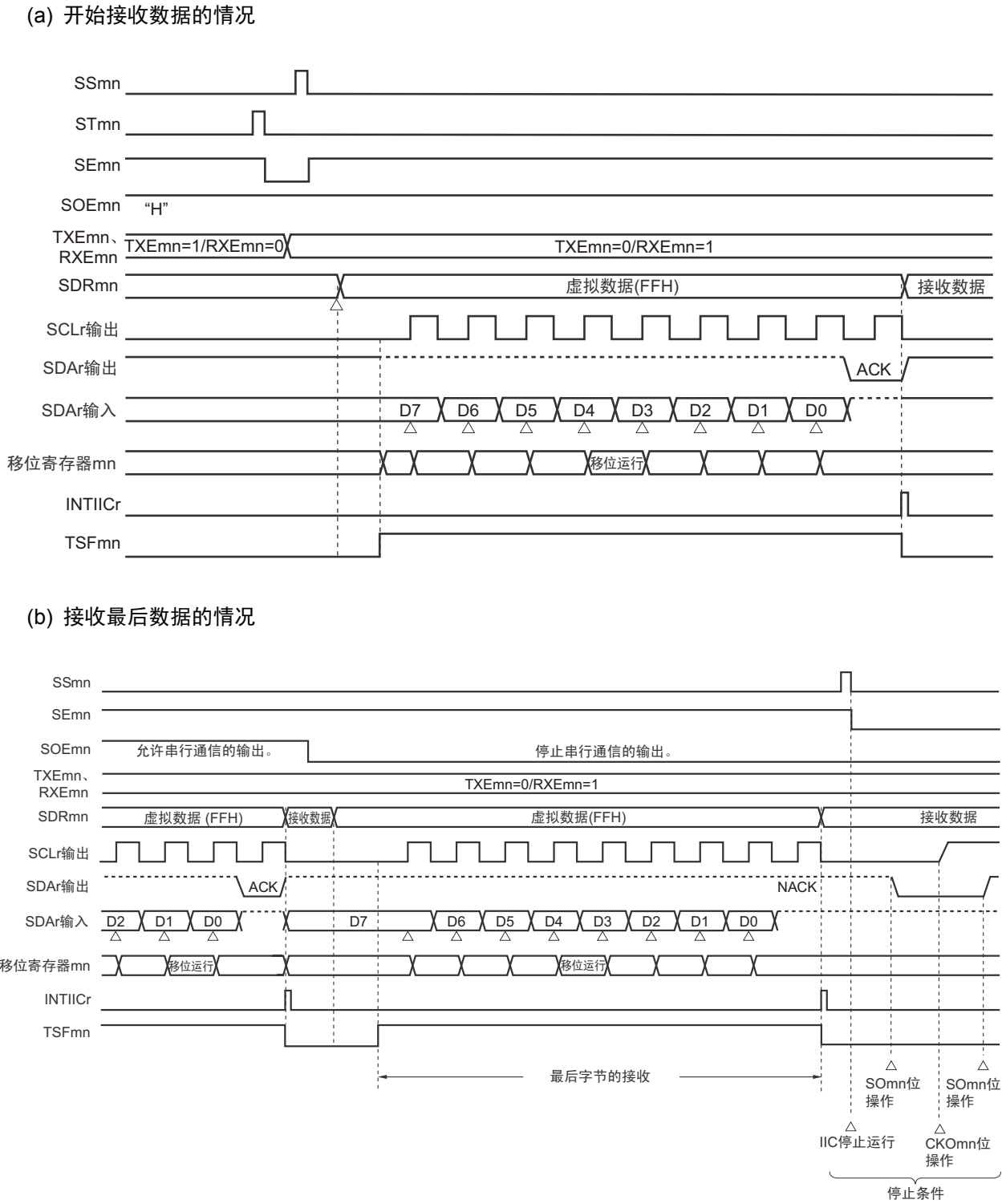
- 注 1. 只限于 SMR01 寄存器和 SMR11 寄存器。
 2. 只限于 SCR00、SCR02、SCR10 寄存器。
 3. 只限于 SCR00 寄存器和 SCR01 寄存器, 其他固定为“1”。
 4. 因为已经在发送地址段时设定, 所以不需要设定。
 5. 在通信运行过程中, 值因通信数据而变。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0~3) r: IIC 号 (r=00、01、10、11、20、21)
 mn=00~03、10、11

2. : 在 IIC 模式中为固定设定。■: 不能设定 (设定初始值)。
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

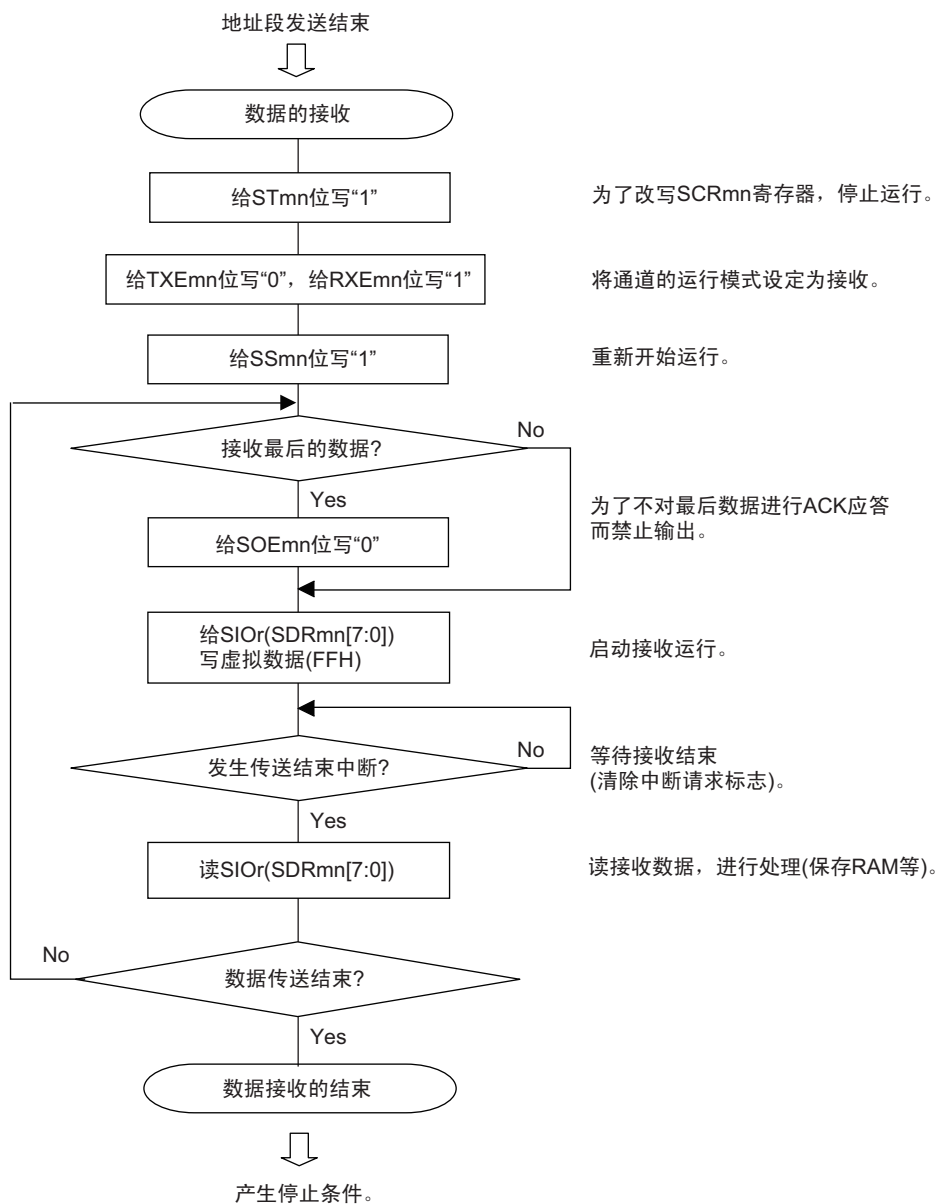
(2) 处理流程

图 14-136 数据接收的时序图



备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) r: IIC 号 (r=00、01、10、11、20、21)
mn=00 ~ 03、10、11

图 14-137 数据接收的流程图



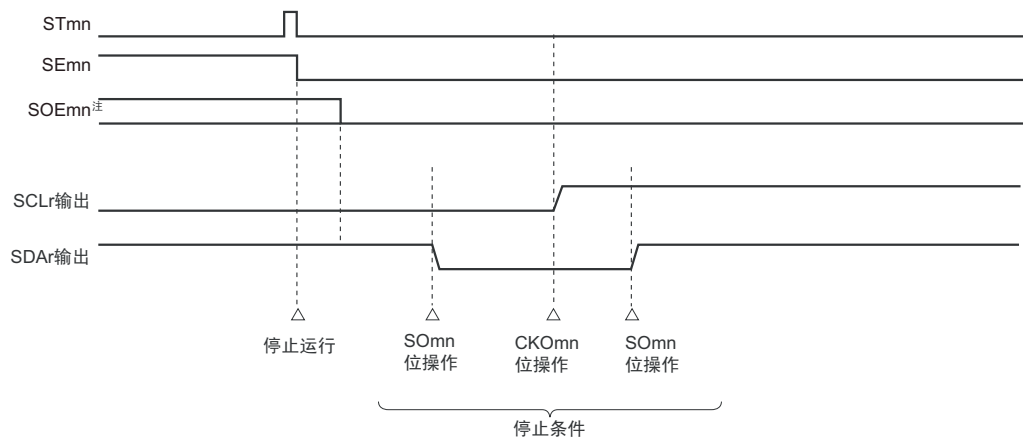
注意 在接收最后数据时不输出 ACK (NACK)。此后，先通过将串行通道停止寄存器 m (STm) 的 STmn 位置“1”来停止运行，然后产生停止条件来结束通信。

14.9.4 停止条件的产生

在与对象从属设备进行了全部数据的发送和接收后，产生停止条件并且释放总线。

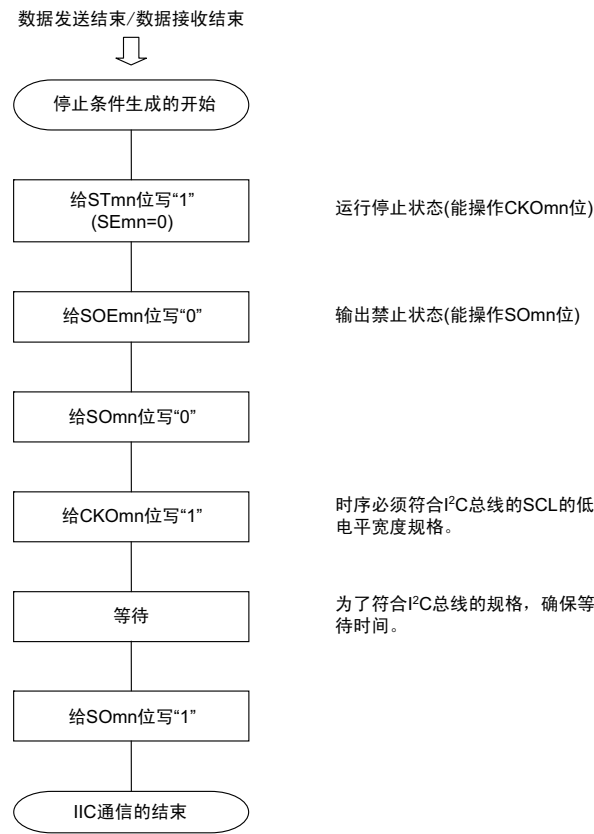
(1) 处理流程

图 14-138 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 14-139 产生停止条件的流程图



14.9.5 传送速率的计算

简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）通信的传送速率能用以下计算式进行计算。

$$(\text{传送速率}) = \{ \text{对象通道的运行时钟 } (f_{\text{MCK}}) \text{ 频率} \} \div (\text{SDRmn}[15:9] + 1) \div 2$$

注意 禁止将 SDRmn[15:9] 设定为“00000000B”，SDRmn[15:9] 的设定值必须大于等于“00000001B”。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、11

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 14-6 简易 I²C 运行时钟的选择

SMRmn 寄存器	SPSm 寄存器								运行时钟 (f _{MCK}) 注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f _{CLK} =32MHz 运行时
0	X	X	X	X	0	0	0	0	f _{CLK}	32MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	1MHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	500kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	250kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	125kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	62.5kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	31.25kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	15.63kHz
1	0	0	0	0	X	X	X	X	f _{CLK}	32MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	16MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	8MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	4MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	2MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	1MHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	500kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	250kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	125kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	62.5kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	31.25kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	15.63kHz
上述以外									禁止设定。	

注 要更改被选择为 f_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止串行阵列单元（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

f_{MCK}=f_{CLK}=32MHz 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} =32MHz			
	运行时钟 (f _{MCK})	SDRmn[15:9]	计算的传送速率	与期待的传送速率的误差
100kHz	f _{CLK} /2	79	100kHz	0.0%
400kHz	f _{CLK}	41	380kHz	5.0% 注
1MHz	f _{CLK}	18	0.84MHz	16.0% 注

注 因为 SCL 信号的占空比为 50%，所以不能将误差设定为“0”% 左右。

14.9.6 在简易 I²C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21) 通信过程中发生错误时的处理步骤

在简易 I²C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21) 通信过程中发生错误时的处理步骤如图 14-140 和图 14-141 所示。

图 14-140 发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn (SDRmn)。→	SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn (SSRmn)。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn (SIRmn) 写“1”。	清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SIRmn 寄存器，只能清除读操作时的错误。

图 14-141 在简易 I²C 模式发生 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、01、10、11、20、21)
mn=00 ~ 03、10、11

第 15 章 串行接口 IICA

15.1 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。

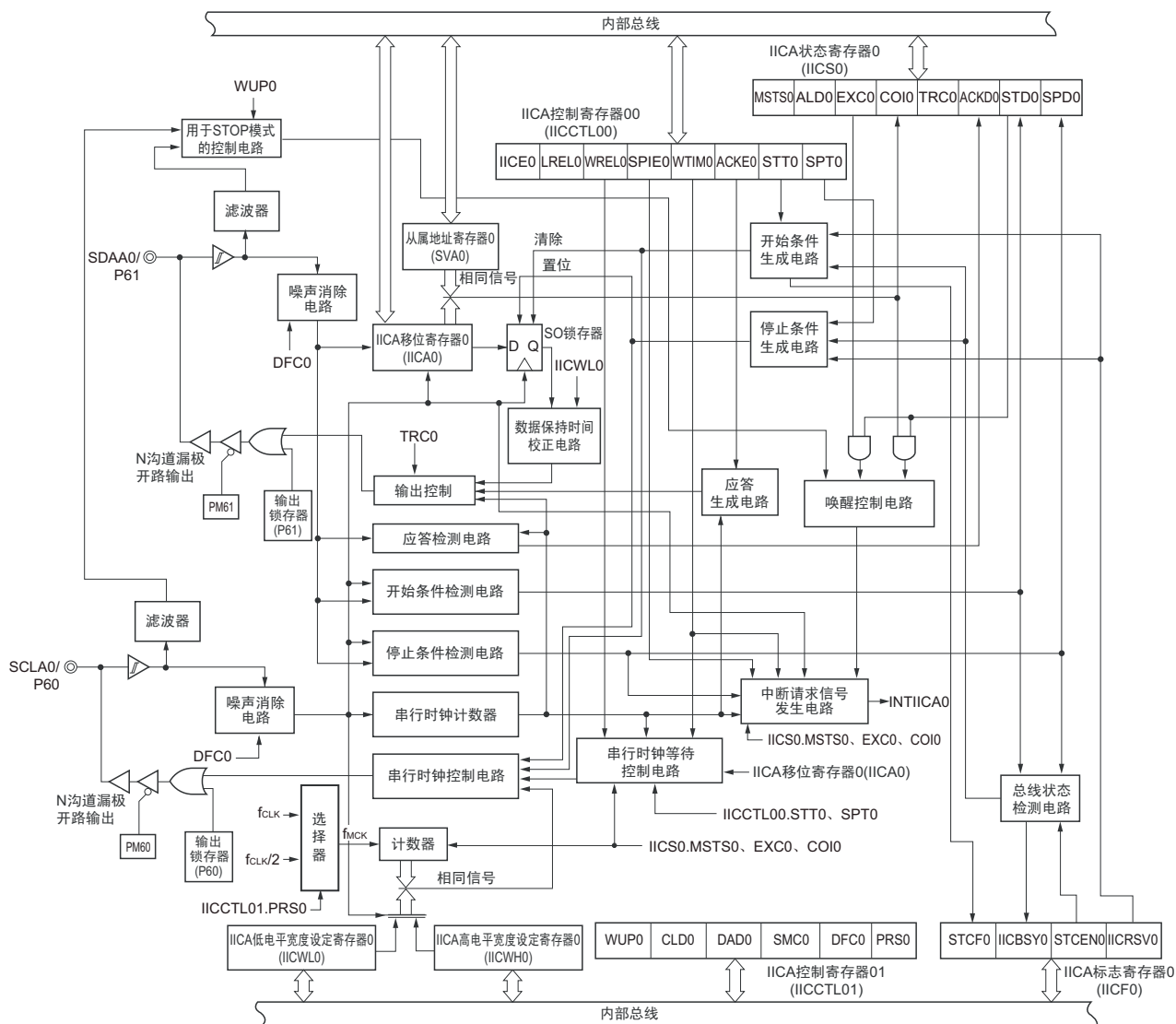
(3) 唤醒模式

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 15-1 所示。

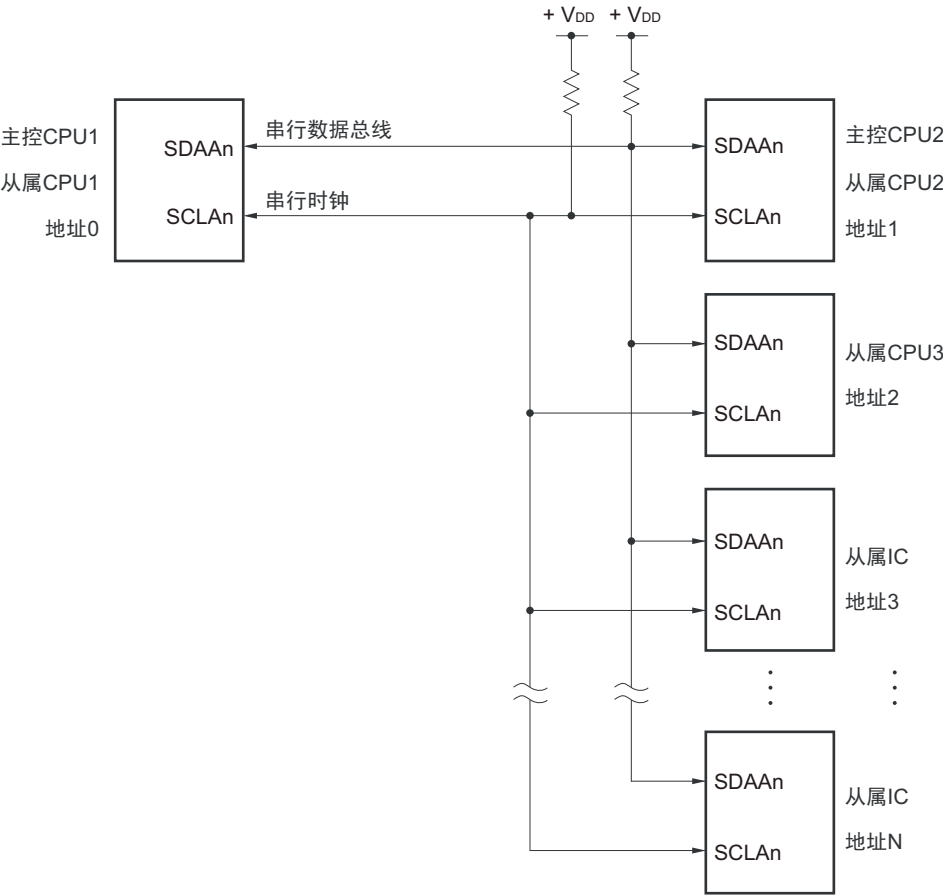
备注 n=0

图 15-1 串行接口 IICA 的框图



串行总线的结构例子如图 15-2 所示。

图 15-2 I²C 总线的串行总线结构例子



备注 n=0

15.2 串行接口 IICA 的结构

串行接口 IICA 由以下硬件构成。

表 15-1 串行接口 IICA 的结构

项目	结构
寄存器	IICA 移位寄存器 n (IICAn) 从属地址寄存器 n (SVAn)
控制寄存器	外围允许寄存器 0 (PER0) IICA 控制寄存器 n0 (IICCTLn0) IICA 状态寄存器 n (IICSn) IICA 标志寄存器 n (IICFn) IICA 控制寄存器 n1 (IICCTLn1) IICA 低电平宽度设定寄存器 n (IICWLn) IICA 高电平宽度设定寄存器 n (IICWHn) 端口模式寄存器 6 (PM6) 端口寄存器 6 (P6)

备注 n=0、1

(1) IICA 移位寄存器 n (IICAn)

IICAn 寄存器是与串行时钟同步进行 8 位串行数据和 8 位并行数据相互转换的寄存器，用于发送和接收。
能通过读写 IICAn 寄存器来控制实际的发送和接收。
在等待期间，通过写 IICAn 寄存器来解除等待，开始传送数据。
通过 8 位存储器操作指令设定 IICAn 寄存器。
在产生复位信号后，此寄存器的值变为“00H”。

图 15-3 IICAn 移位寄存器 n (IICAn) 的格式

地址: FFF50H (IICA0)		复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0	
IICAn									

注意 1. 在数据传送过程中，不能给 IICAn 寄存器写数据。

- 只能在等待期间读写 IICAn 寄存器。除了等待期间以外，禁止在通信状态下存取 IICAn 寄存器。但是，在主控设备的情况下，能在将通信触发位 (STTn) 置“1”后写一次 IICAn 寄存器。
- 当预约通信时，必须在检测到由停止条件产生的中断后给 IICAn 寄存器写数据。

备注 n=0

(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 15-4 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 个或者第 9 个串行时钟的下降 (通过 WTIMn 位进行设定)
- 因检测到停止条件而产生中断请求 (通过 SPIEn 位进行设定)。

备注 WTIMn 位 : IICA 控制寄存器 n0 (IICCTLn0) 的 bit3

SPIEn 位 : IICA 控制寄存器 n0 (IICCTLn0) 的 bit4

(7) 串行时钟控制电路

在主机模式中，此电路从采样时钟生成输出到 SCLAn 引脚的时钟。

(8) 串行时钟等待控制电路

此电路控制等待时序。

备注 n=0

(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

15.3 控制串行接口 IICA 的寄存器

通过以下 8 种寄存器控制串行接口 IICA。

- 外围允许寄存器0 (PER0)
- IICA 控制寄存器 n0 (IICCTLn0)
- IICA 标志寄存器 n (IICFn)
- IICA 状态寄存器 n (IICSn)
- IICA 控制寄存器 n1 (IICCTLn1)
- IICA 低电平宽度设定寄存器 n (IICWLn)
- IICA 高电平宽度设定寄存器 n (IICWHn)
- 端口模式寄存器 6 (PM6)
- 端口寄存器 6 (P6)

备注 n=0、1

15.3.1 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 15-5 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	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

IICAnEN	提供串行接口 IICAn 的输入时钟的控制
0	停止提供输入时钟。 - 不能写串行接口 IICAn 使用的 SFR。 - 串行接口 IICAn 处于复位状态。
1	允许提供输入时钟。 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. 必须将以下的位置“0”。

bit1 和 bit6

备注 n=0

15.3.2 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 15-6 IICA 控制寄存器 n0 (IICCTLn0) 的格式 (1/4)

地址: F0230H (IICCTL00)

复位: 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)
- 通过指令清除。 - 当复位时	通过指令置位。

LRELn ^{注 2、3}	通信的退出
0	通常运行
1	退出当前的通信, 进入待机状态。执行后自动清“0”。 在接收到与本站无关的扩展码等情况时使用。 SCLAn 线和 SDAAn 线变为高阻状态。 IICA 控制寄存器 n0 (IICCTLn0) 和 IICA 状态寄存器 n (IICSn) 中的以下标志被清“0”: •STTn •SPTn •MSTSn •EXCn •COIn •TRCn •ACKDn •STDn
变为退出通信的待机状态, 保持到满足以下的通信参加条件为止。	
- 在检测到停止条件后作为主控设备启动。 - 在检测到开始条件后地址匹配或者接收到扩展码。	
清除条件 (LRELn=0)	置位条件 (LRELn=1)
- 在执行后自动清除。 - 当复位时	通过指令置位。

WRELn ^{注 2、3}	等待的解除
0	不解除等待。
1	解除等待。在解除等待后自动清除。
如果在发送状态下 (TRCn=1) 的第 9 个时钟等待期间将 WRELn 位 (解除等待) 置位, SDAAn 线就变为高阻抗状态 (TRCn=0)。	
清除条件 (WRELn=0)	置位条件 (WRELn=1)
- 在执行后自动清除。 - 当复位时	通过指令置位。

- 注 1. 对 IICA 移位寄存器 n (IICAn)、IICA 标志寄存器 n (IICFn) 的 STCFn 位和 IICBSYn 位以及 IICA 控制寄存器 n1 (IICCTLn1) 的 CLDn 位和 DADn 位进行复位。
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 15-6 IICA 控制寄存器 n0 (IICCTLn0) 的格式 (2/4)

SPIEn 注 1	允许或者禁止停止条件检测产生的中断请求
0	禁止
1	允许
当 IICA 控制寄存器 n1 (IICCTLn1) 的 WUPn 位为“1”时, 即使将 SPIEn 位置“1”也不产生停止条件中断。	
清除条件 (SPIEn=0)	置位条件 (SPIEn=1)
- 通过指令清除。 - 当复位时	通过指令置位。

WTIMn 注 1	等待和中断请求的控制
0	在第 8 个时钟的下降沿产生中断请求信号。 主控设备: 在输出 8 个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入 8 个时钟后, 将时钟置为低电平, 然后等待主控设备。
1	在第 9 个时钟的下降沿产生中断请求信号。 主控设备: 在输出 9 个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入 9 个时钟后, 将时钟置为低电平, 然后等待主控设备。
在地址传送期间, 与此位的设定无关, 在第 9 个时钟的下降沿产生中断; 在地址传送结束后, 此位的设定有效。主控设备在地址传送期间的第 9 个时钟下降沿进入等待。接收到本地站地址的从属设备在产生应答 (ACK) 后的第 9 个时钟下降沿进入等待, 但是接收到扩展码的从属设备在第 8 个时钟下降沿进入等待。	
清除条件 (WTIMn=0)	置位条件 (WTIMn=1)
- 通过指令清除。 - 当复位时	通过指令置位。

ACKEn 注 1、2	应答控制
0	禁止应答。
1	允许应答。在第 9 个时钟期间将 SDAAn 线置为低电平。
清除条件 (ACKEn=0)	置位条件 (ACKEn=1)
- 通过指令清除。 - 当复位时	通过指令置位。

注 1. 在 IICEn 位为“0”的状态下, 此位的信号无效。必须在此期间设定此位。

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 15-6 IICA 控制寄存器 n0 (IICCTLn0) 的格式 (3/4)

STTn 注 1、2	开始条件的触发
0	不生成开始条件。
1	当总线被释放时（待机状态，IICBSYn 位为“0”）： 如果将此位置“1”，就生成开始条件（作为主控设备的启动）。 当第三方正在通信时： - 允许通信预约功能的情况（IICRSVn=0） 用作开始条件预约标志。如果将此位置“1”，就在释放总线后自动生成开始条件。 - 禁止通信预约功能的情况（IICRSVn=1） 即使将此位置“1”，也清除 STTn 位并且将 STTn 清除标志（STCFn）置“1”，不生成开始条件。 等待状态（主控设备）： 在解除等待后生成重新开始条件。
有关置位时序的注意事项： - 主控接收：禁止在传送过程中将此位置“1”。只有在将 ACKEn 位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 - 主控发送：在应答期间，可能无法正常生成开始条件。必须在输出第 9 个时钟后的等待期间将此位置“1”。 - 禁止与停止条件的触发（SPTn）同时置“1”。 - 在将 STTn 位置“1”后，禁止在满足清除条件前再次将此位“1”。	
清除条件（STTn=0）	置位条件（STTn=1）
- 在禁止通信预约的状态下将 STTn 位置“1”。 - 在仲裁失败时 - 主控设备生成开始条件。 - 因 LRELn 位为“1”（退出通信）而进行的清除 - 当 IICEn 位为“0”（停止运行）时 - 当复位时	通过指令置位。

注 1. 在 IICEn 位为“0”的状态下，此位的信号无效。

2. STTn 位的读取值总是“0”。

备注 1. 如果在设定数据后读 bit1（STTn），此位就变为“0”。

2. IICRSVn : IICA 标志寄存器 n (IICFn) 的 bit0

STCFn : IICA 标志寄存器 n (IICFn) 的 bit7

3. n=0

图 15-6 IICA 控制寄存器 n0 (IICCTLn0) 的格式 (4/4)

SPTn 注	停止条件的触发
0	不生成停止条件。
1	生成停止条件 （作为主控设备的传送结束）。
有关置位时序的注意事项：	
• 主控接收：禁止在传送过程中将此位置“1”。只有在将 ACKEn 位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 • 主控发送：在应答期间，可能无法正常生成停止条件。必须在输出第 9 个时钟后的等待期间将此位置“1”。 • 禁止与开始条件的触发（STTn）同时置“1”。 • 只有在主控设备的情况下才能将 SPTn 位置“1”。 • 在 WTIMn 位为“0”时，必须注意：如果在输出 8 个时钟后的等待期间将 SPTn 位置“1”，就在解除等待后的第 9 个时钟的高电平期间生成停止条件。必须在输出 8 个时钟后的等待期间将 WTIMn 位从“0”置为“1”并且在输出第 9 个时钟后的等待期间将 SPTn 位置“1”。 • 在将 SPTn 位置“1”后，禁止在满足清除条件前再次将此位置“1”。	
清除条件（SPTn=0）	置位条件（SPTn=1）
• 当仲裁失败时 • 在检测到停止条件后自动清除。 • 因 LRELn 位为“1”（退出通信）而进行的清除 • 当 IICEn 位为“0”（停止运行）时 • 当复位时	• 通过指令置位。

注 SPTn 位的读取值总是“0”。

注意 在 IICA 状态寄存器 n (IICSn) 的 bit3 (TRCn) 为“1” (发送状态) 时, 如果在第 9 个时钟将 IICCTLn0 寄存器的 bit5 (WRELn) 置“1”来解除等待, 就在清除 TRCn 位 (接收状态) 后将 SDAA_n 线置为高阻抗。必须通过写 IICA 移位寄存器 n 进行 TRCn 位为“1” (发送状态) 时的等待解除。

备注 n=0

15.3.3 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 15-7 IICA 状态寄存器 n (IICSn) 的格式 (1/3)



注 即使对 IICSn 寄存器以外的位执行位存储器操作指令，也清除此位。因此，在使用 ALDn 位时，必须在读其他位前先读 ALDn 位的数据。

备注 1. 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 15-7 IICA 状态寄存器 n (IICSn) 的格式 (2/3)

EXCn	扩展码的接收检测	
0	未接收到扩展码。	
1	接收到扩展码。	
清除条件 (EXCn=0)		置位条件 (EXCn=1)
- 当检测到开始条件时 - 当检测到停止条件时 - 因 LRELn 位为“1”（退出通信）而进行的清除 - 当 IICEn 位从“1”变为“0”（停止运行）时 - 当复位时		当接收的地址数据的高 4 位为“0000”或者“1111”时（在第 8 个时钟的上升沿置位）

COIn	匹配地址的检测	
0	地址不同。	
1	地址相同。	
清除条件 (COIn=0)		置位条件 (COIn=1)
- 当检测到开始条件时 - 当检测到停止条件时 - 因 LRELn 位为“1”（退出通信）而进行的清除 - 当 IICEn 位从“1”变为“0”（停止运行）时 - 当复位时		当接收地址和本地站地址（从属地址寄存器 n (SVAn)）相同时（在第 8 个时钟的上升沿置位）

TRCn	发送 / 接收的状态检测	
0	接收状态（发送状态除外）。将 SDAAn 线置为高阻抗。	
1	发送状态。设定为能将 SOn 锁存器的值输出到 SDAAn 线（在第 1 字节的第 9 个时钟的下降沿以后有效）。	
清除条件 (TRCn=0)		置位条件 (TRCn=1)
<主控设备和从属设备> - 当检测到停止条件时 - 因 LRELn 位为“1”（退出通信）而进行的清除 - 当 IICEn 位从“1”变为“0”（停止运行）时 - 因 WRELn 位为“1”（解除等待）而进行的清除注 - 当 ALDn 位从“0”变为“1”（仲裁失败）时 - 当复位时 - 不参加通信的情况 (MSTSn、EXCn、COIn=0) <主控设备> - 当第 1 字节的 LSB（传送方向指示位）输出“1”时 <从属设备> - 当检测到开始条件时 - 当第 1 字节的 LSB（传送方向指示位）输入“0”时		<主控设备> - 当生成开始条件时 - 当第 1 字节（地址传送）的 LSB（传送方向指示位）输出“0”（主控发送）时 <从属设备> - 当主控设备的第 1 字节（地址传送）的 LSB（传送方向指示位）输入“1”（从属发送）时

注 在 IICA 状态寄存器 n (IICSn) 的 bit3 (TRCn) 为“1”（发送状态）时，如果在第 9 个时钟将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit5 (WRELn) 置“1”来解除等待，就在清除 TRCn 位（接收状态）后将 SDAAn 线置为高阻抗。必须通过写 IICA 移位寄存器 n 进行 TRCn 位为“1”（发送状态）时的等待解除。

备注 1. LRELn : IICA 控制寄存器 n0 (IICCTLn0) 的 bit6

IICEn : IICA 控制寄存器 n0 (IICCTLn0) 的 bit7

2. n=0

图 15-7 IICA 状态寄存器 n (IICSn) 的格式 (3/3)

ACKDn	应答 (ACK) 的检测
0	未检测到应答。
1	检测到应答。
清除条件 (ACKDn=0)	置位条件 (ACKDn=1)
- 当检测到停止条件时 - 当下一字节的第 1 个时钟上升时 - 因 LRELn 位为“1” (退出通信) 而进行的清除 - 当 IICEn 位从“1”变为“0” (停止运行) 时 - 当复位时	在 SCLAn 线的第 9 个时钟上升沿将 SDAAn 线置为低电平时

STDn	开始条件的检测
0	未检测到开始条件。
1	检测到开始条件, 表示处于地址传送期间。
清除条件 (STDn=0)	置位条件 (STDn=1)
- 当检测到停止条件时 - 在地址传送后的下一个字节的第 1 个时钟上升时 - 因 LRELn 位为“1” (退出通信) 而进行的清除 - 当 IICEn 位从“1”变为“0” (停止运行) 时 - 当复位时	当检测到开始条件时

SPDn	停止条件的检测
0	未检测到停止条件。
1	检测到停止条件, 主控设备结束通信并且已释放总线。
清除条件 (SPDn=0)	置位条件 (SPDn=1)
- 在将此位置位后, 在检测到开始条件后的地址传送字节的第 1 个时钟上升时 - 当 WUPn 位从“1”变为“0”时 - 当 IICEn 位从“1”变为“0” (停止运行) 时 - 当复位时	当检测到停止条件时

备注 1. LRELn : IICA 控制寄存器 n0 (IICCTLn0) 的 bit6

IICEn : IICA 控制寄存器 n0 (IICCTLn0) 的 bit7

2. n=0

15.3.4 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 15-8 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)	
- 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	
置位条件 (STCFn=1)	
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)	
- 当检测到停止条件时 - 当 IICEn 位为“0” (停止运行) 时 - 当复位时	
置位条件 (IICBSYn=1)	
- 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)	
- 通过指令清除。 - 当检测到开始条件时 - 当复位时	
置位条件 (STCENn=1)	
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)	
- 通过指令清除。 - 当复位时	
置位条件 (IICRSVn=1)	
通过指令置位。	

注 bit6 和 bit7 是只读位。

注意 1. 只有在停止运行 (IICEn=0) 时才能写 STCENn 位。

2. 如果 STCENn 位为“1”, 就与实际的总线状态无关而认为总线为释放状态 (IICBSYn=0), 因此为了避免在发行第 1 个开始条件 (STTn=1) 时破坏其他通信, 需要确认没有正在通信的第三方。

3. 只有在停止运行 (IICEn=0) 时才能写 IICRSVn。

备注 1. STTn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit1

2. IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7

15.3.5 IICA 控制寄存器 n1 (IICCTLn1)

这是用于设定 I²C 运行模式以及检测 SCLAn 引脚和 SDAAn 引脚状态的寄存器。

通过 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”。

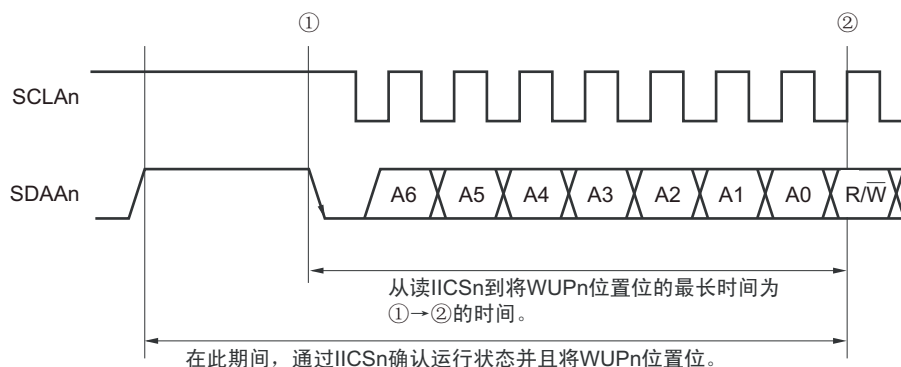
图 15-9 IICA 控制寄存器 n1 (IICCTLn1) 的格式 (1/2)

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 15-22 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)	
通过指令清除（在地址匹配或者接收到扩展码后）。	
置位条件 (WUPn=1)	
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 15-9 IICA 控制寄存器 n1 (IICCTLn1) 的格式 (2/2)

CLDn	SCLAn 引脚的电平检测 (只在 IICEn 位为“1”时有效)
0	检测到 SCLAn 引脚为低电平。
1	检测到 SCLAn 引脚为高电平。
清除条件 (CLDn=0)	
置位条件 (CLDn=1)	
- 当 SCLAn 引脚为低电平时 - 当 IICEn 位为“0” (停止运行) 时 - 当复位时	当 SCLAn 引脚为高电平时

DADn	SDAAn 引脚的电平检测 (只在 IICEn 位为“1”时有效)
0	检测到 SDAAn 引脚为低电平。
1	检测到 SDAAn 引脚为高电平。
清除条件 (DADn=0)	
置位条件 (DADn=1)	
- 当 SDAAn 引脚为低电平时 - 当 IICEn 位为“0” (停止运行) 时 - 当复位时	当 SDAAn 引脚为高电平时

SMCn	运行模式的切换
0	在标准模式中运行 (最大传送速率: 100kbps)。
1	在快速模式 (最大传送速率: 400kbps) 或者增强型快速模式 (最大传送速率: 1Mbps) 中运行。

DFCn	数字滤波器的运行控制
0	数字滤波器 OFF
1	数字滤波器 ON
必须在快速模式或者增强型快速模式中使用数字滤波器。 数字滤波器用于消除噪声。 无论是将 DFCn 位置“1”还是清“0”，传送时钟都不变。	

PRSn	运行时钟 (f_{MCK}) 的控制
0	选择 f_{CLK} ($1MHz \leq f_{CLK} \leq 20MHz$)。
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

15.3.6 IICA 低电平宽度设定寄存器 n (IICWL_n)

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 IICWL_n 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 IICWL_n 寄存器。

在产生复位信号后, 此寄存器的值变为“FFH”。

有关 IICWL_n 寄存器的设定方法, 请参照“15.4.2 通过 IICWL_n 寄存器和 IICWH_n 寄存器设定传送时钟的方法”。

图 15-10 IICA 低电平宽度设定寄存器 n (IICWL_n) 的格式

地址: F0232H (IICWL0)		复位后: FFH		R/W					
符号	7	6	5	4	3	2	1	0	
IICWLn									

15.3.7 IICA 高电平宽度设定寄存器 n (IICWH_n)

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 IICWH_n 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 IICWH_n 寄存器。

在产生复位信号后, 此寄存器的值变为“FFH”。

图 15-11 IICA 高电平宽度设定寄存器 n (IICWH_n) 的格式

地址: F0233H (IICWH0)		复位后: FFH		R/W				
符号	7	6	5	4	3	2	1	0
IICWHn								

备注 1. 有关主控方传送时钟的设定方法, 请参照 15.4.2(1); 有关从属方 IICWL_n 寄存器和 IICWH_n 寄存器的设定方法, 请参照 15.4.2(2)。

2. n=0

15.3.8 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 15-12 端口模式寄存器 6（PM6）的格式

地址：FFF26H	复位后：FFH	R/W						
符号	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PM6n	P6n 引脚输入 / 输出模式的选择（n=0、1）
0	输出模式（输出缓冲器 ON）
1	输入模式（输出缓冲器 OFF）

15.4 I²C 总线模式的功能

15.4.1 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

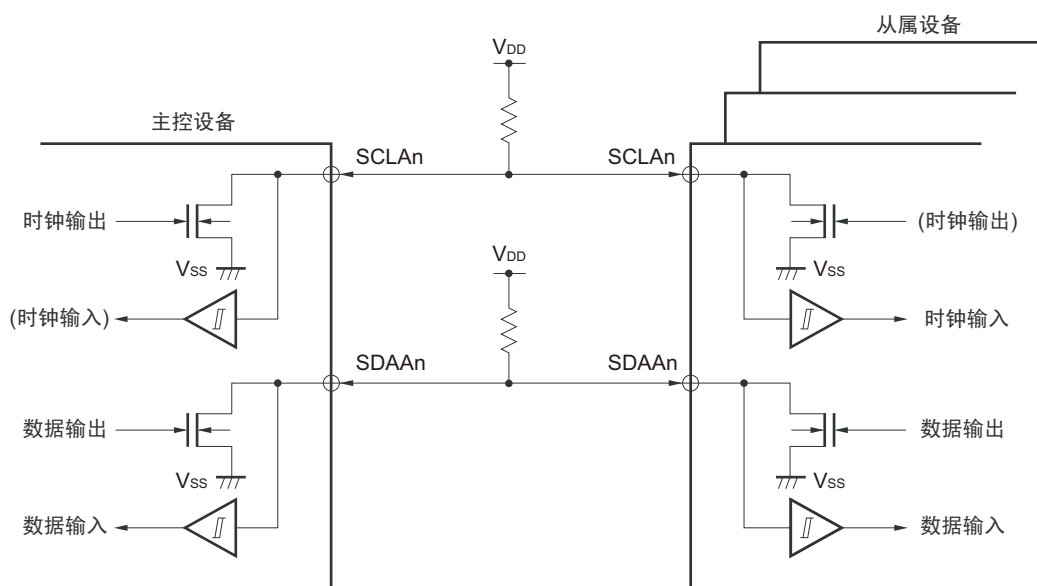
主控设备和从属设备的输出都为N沟道漏极开路输出，输入都为施密特输入。

(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 15-13 引脚结构图



备注 n=0

15.4.2 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 : $f_{\text{CLK}}=3.5\text{MHz}(\text{Min.})$

增强型快速模式 : $f_{\text{CLK}}=10\text{MHz}(\text{Min.})$

标准模式 : $f_{\text{CLK}}=1\text{MHz}(\text{Min.})$

备注 1. 因为 SDAAn 信号和 SCLAn 信号的上升时间 (t_R) 和下降时间 (t_F) 因上拉电阻和布线电容而不同, 所以必须各自计算。

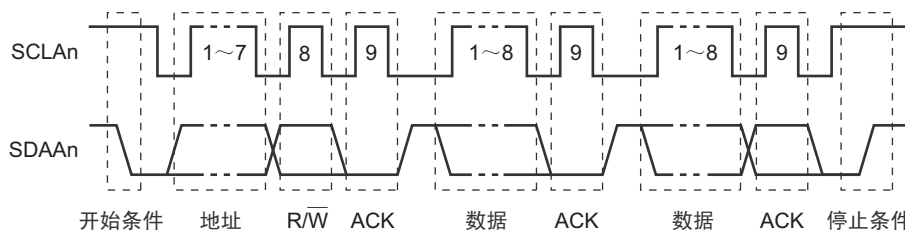
2. IICWLn : IICA 低电平宽度设定寄存器 n
IICWHn : IICA 高电平宽度设定寄存器 n
 t_F : SDAAn 信号和 SCLAn 信号的下降时间
 t_R : SDAAn 信号和 SCLAn 信号的上升时间
 f_{MCK} : IICA 运行时钟频率
3. n=0

15.5 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 15-14 所示。

图 15-14 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。

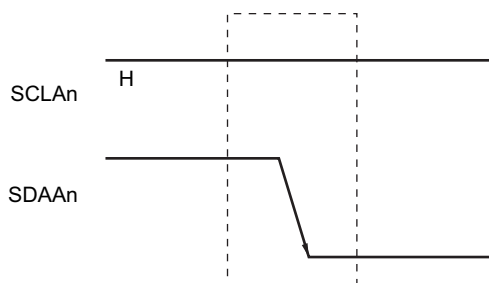
主控设备和从属设备都能生成应答 (ACK) (在一般情况下, 接收方输出 8 位数据)。

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。

15.5.1 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 15-15 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

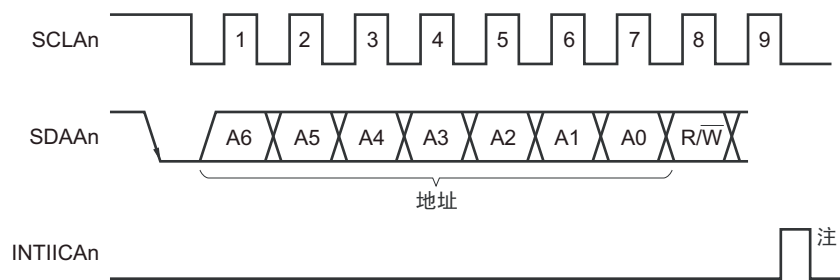
15.5.2 地址

开始条件的后续 7 位数据被定义为地址。

地址是主控设备为了从连接在总线的多个从属设备中选择特定的从属设备而输出的 7 位数据。因此，总线上的从属设备需要设定完全不同的地址。

从属设备通过硬件检测到开始条件，并且检查 7 位数据是否和从属地址寄存器 n（SVAn）的内容相同。此时，如果 7 位数据和 SVAn 寄存器的值相同，该从属设备就被选中，在主控设备生成开始条件或者停止条件前，与主控设备进行通信。

图 15-16 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“15.5.3 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。

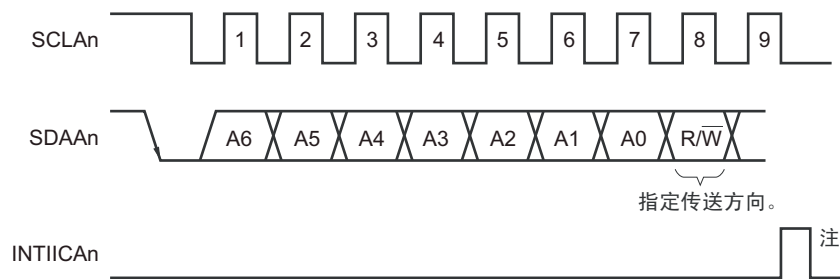
从属地址分配在 IICAn 寄存器的高 7 位。

15.5.3 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 15-17 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

15.5.4 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 (IICS_n) 的 bit2 (ACKD_n) 确认应答的检测。

在主机设备为接收状态下接收到最后的数据时, 不返回应答而生成停止条件。在从属设备接收数据后不返回应答时, 主机设备输出停止条件或者重新开始条件, 中止发送。不返回应答的原因如下:

- ① 没有正常接收。
- ② 已结束最后数据的接收。
- ③ 不存在地址指定的接收方。

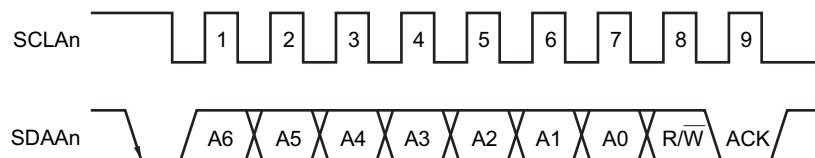
接收方在第 9 个时钟将 SDAAn 线置为低电平, 生成应答 (正常接收)。

通过将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit2 (ACKEn) 置“1”, 变为能自动生成应答的状态。能通过 7 位地址信息后续的第 8 位数据设定 IICS_n 寄存器的 bit3 (TRC_n)。在接收 (TRC_n=0) 的情况下, 通常必须将 ACKEn 位置“1”。

在从属接收运行过程中 (TRC_n=0) 不能接收数据或者不需要下一个数据时, 必须将 ACKEn 位清“0”, 通知主机方不能接收数据。

在主机接收运行过程中 (TRC_n=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 15-18 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 (WTIM_n) = 0):
通过在解除等待前将 ACKEn 位置“1”, 与 SCLAn 引脚的第 8 个时钟下降同步生成应答。
- 当选择 9 个时钟的等待时 (IICCTLn0 寄存器的 bit3 (WTIM_n) = 1):
通过事先将 ACKEn 位置“1”, 生成应答。

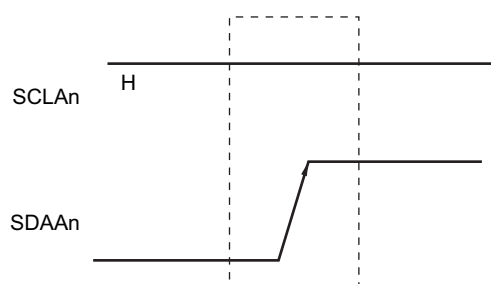
备注 n=0

15.5.5 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 15-19 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

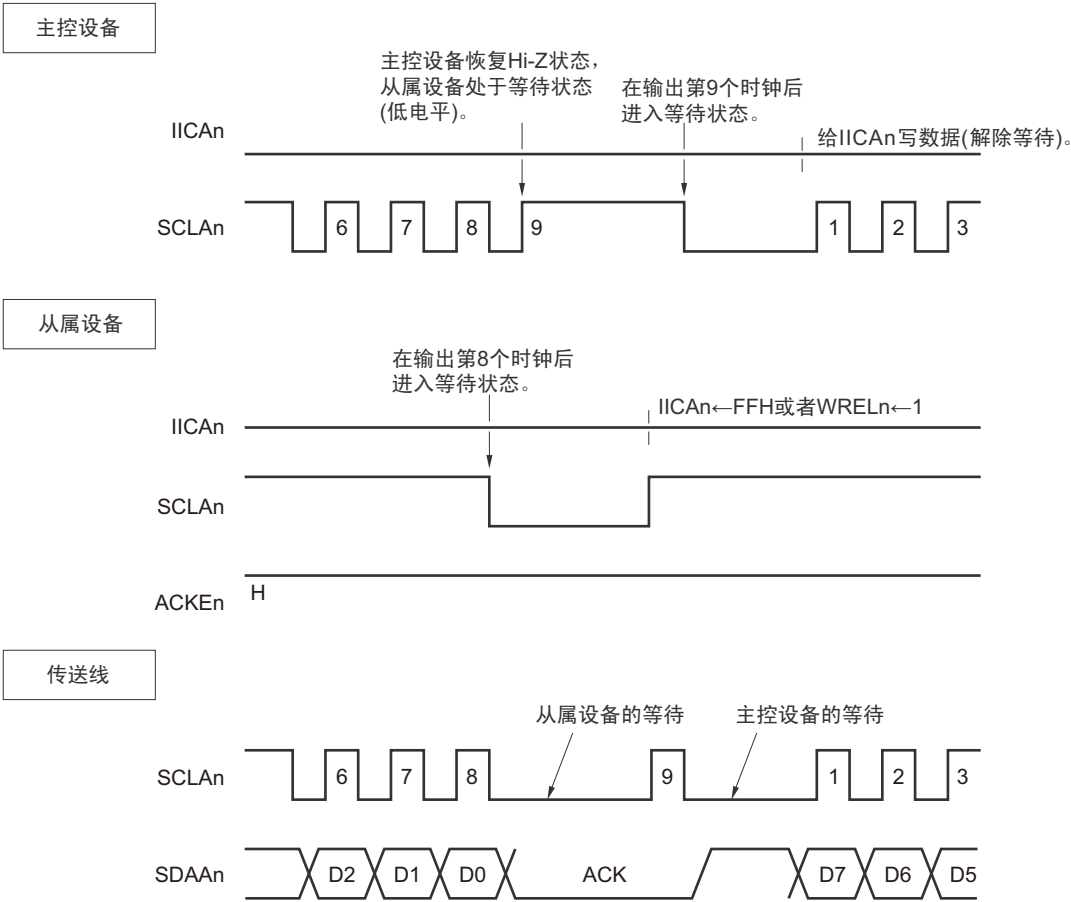
15.5.6 等待

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 15-20 等待 (1/2)

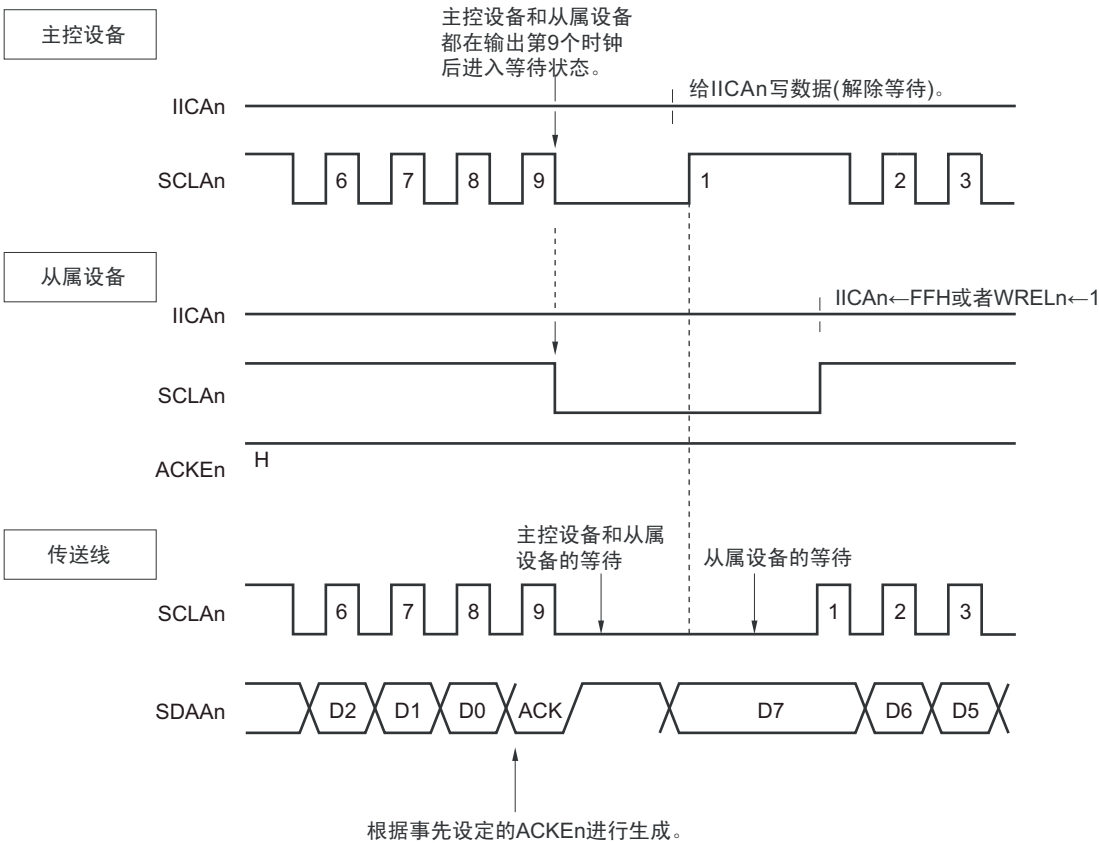
- (1) 主控设备为 9 个时钟等待，从属设备为 8 个时钟等待的情况
（主控设备：发送，从属设备：接收，ACKEn=1）



备注 n=0

图 15-20 等待 (2/2)

(2) 主控设备和从属设备都为 9 个时钟等待的情况
(主控设备：发送，从属设备：接收，ACKEn=1)



备注 ACKEn : IICA 控制寄存器 n0 (IICCTLn0) 的 bit2
WRELn : IICA 控制寄存器 n0 (IICCTLn0) 的 bit5

通过设定 IICA 控制寄存器 n0 (IICCTLn0) 的 bit3 (WTIMn) 自动产生等待。

通常，在接收方，如果 IICCTLn0 寄存器的 bit5 (WRELn) 为“1”或者给 IICA 移位寄存器 n (IICAn) 写“FFH”，就解除等待；在发送方，如果给 IICAn 寄存器写数据，就解除等待。

主控设备还能通过以下方法解除等待：

- 将 IICCTLn0 寄存器的 bit1 (STTn) 置“1”。
- 将 IICCTLn0 寄存器的 bit0 (SPTn) 置“1”。

备注 n=0

15.5.7 等待的解除方法

在一般情况下，I²C 通过以下的处理来解除等待。

- 给 IICA 移位寄存器 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

15.5.8 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 15-2 所示的时序产生 INTIICAn 并且进行等待控制。

表 15-2 INTIICAn 的产生时序和等待控制

WTIMn	从属运行			主控运行		
	地址	数据接收	数据发送	地址	数据接收	数据发送
0	9 注 1、2	8 注 2	8 注 2	9	8	8
1	9 注 1、2	9 注 2	9 注 2	9	9	9

注 1. 只有在接收的地址和从属地址寄存器 n（SVAn）的设定地址相同时，从属设备才在第 9 个时钟的下降沿产生 INTIICAn 信号并且进入等待状态。

此时，与 IICCTLn0 寄存器 bit2（ACKEn）的设定无关，生成应答。接收到扩展码的从属设备在第 8 个时钟的下降沿产生 INTIICAn。如果在重新开始后地址不同，就在第 9 个时钟的下降沿产生 INTIICAn，但是不进入等待状态。

2. 如果接收的地址和从属地址寄存器 n（SVAn）的内容不同并且未接收到扩展码，就不产生 INTIICAn 并且也不进入等待状态。

备注 表中的数字表示串行时钟的时钟数。中断请求和等待控制都与串行时钟的下降沿同步。

(1) 地址的发送和接收

- 从属运行：与 WTIMn 位无关，根据上述注 1 和注 2 的条件决定中断和等待的时序。
- 主控运行：与 WTIMn 位无关，在第 9 个时钟的下降沿产生中断和等待的时序。

(2) 数据的接收

- 主控运行/从属运行：通过 WTIMn 位决定中断和等待的时序。

(3) 数据的发送

- 主控运行/从属运行：通过 WTIMn 位决定中断和等待的时序。

备注 n=0

(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)。

15.5.9 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。当主控设备送来的从属地址和从属地址寄存器 n (SVAn) 的设定地址相同或者接收到扩展码时，产生 INTIICAn 中断请求。

15.5.10 错误的检测

在 I²C 总线模式中，因为发送过程中的串行数据总线 (SDAAn) 的状态被取到发送器件的 IICA 移位寄存器 n (IICAn)，所以能通过将开始发送前和发送结束后的 IICA 数据进行比较来检测发送错误。此时，如果 2 个数据不同，就判断为发生了发送错误。

备注 n=0

15.5.11 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”，进入下次通信的待机状态。

表 15-3 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

15.5.12 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。

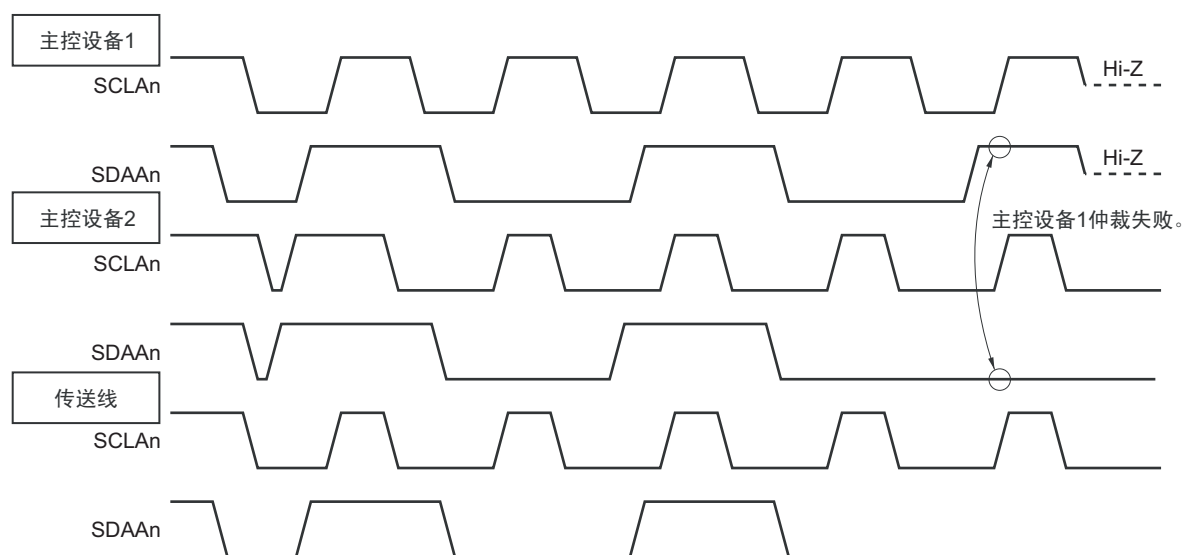
在仲裁失败时，仲裁失败的主控设备将 IICA 状态寄存器 n（ $IICSn$ ）的仲裁失败标志（ $ALDn$ ）置“1”，并且将 $SCLAn$ 线和 $SDAAn$ 线都置为高阻抗状态，释放总线。

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“15.5.8 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 15-21 仲裁时序例子



备注 $n=0$

表 15-4 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

15.5.13 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。

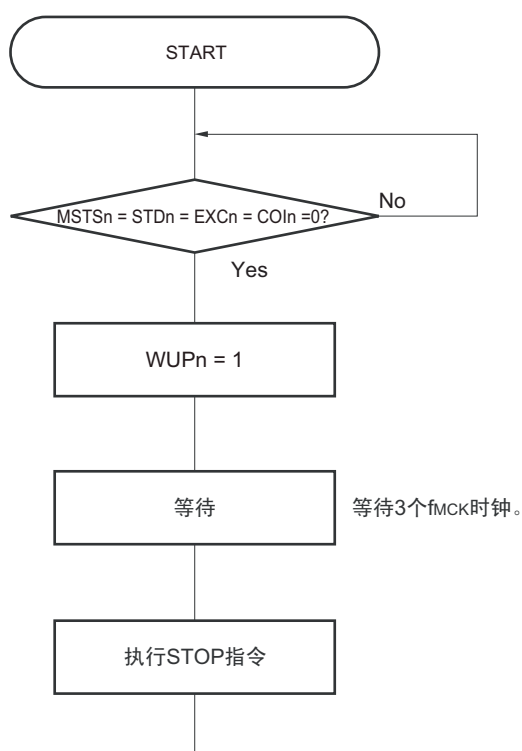
在地址不同的情况下不产生不需要的 INTIICAn 信号，能提高处理效率。

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。

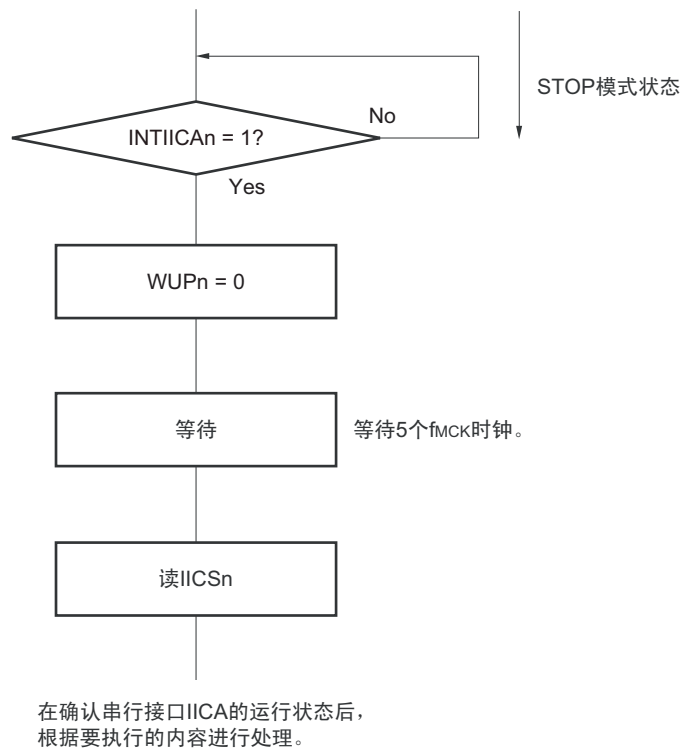
将 WUPn 位置“1”时的流程如图 15-22 所示，通过地址匹配将 WUPn 位置“0”时的流程如图 15-23 所示。

图 15-22 将 WUPn 位置“1”时的流程



备注 n=0

图 15-23 通过地址匹配将 WUPn 位置“0”时的流程（包括接收扩展码）

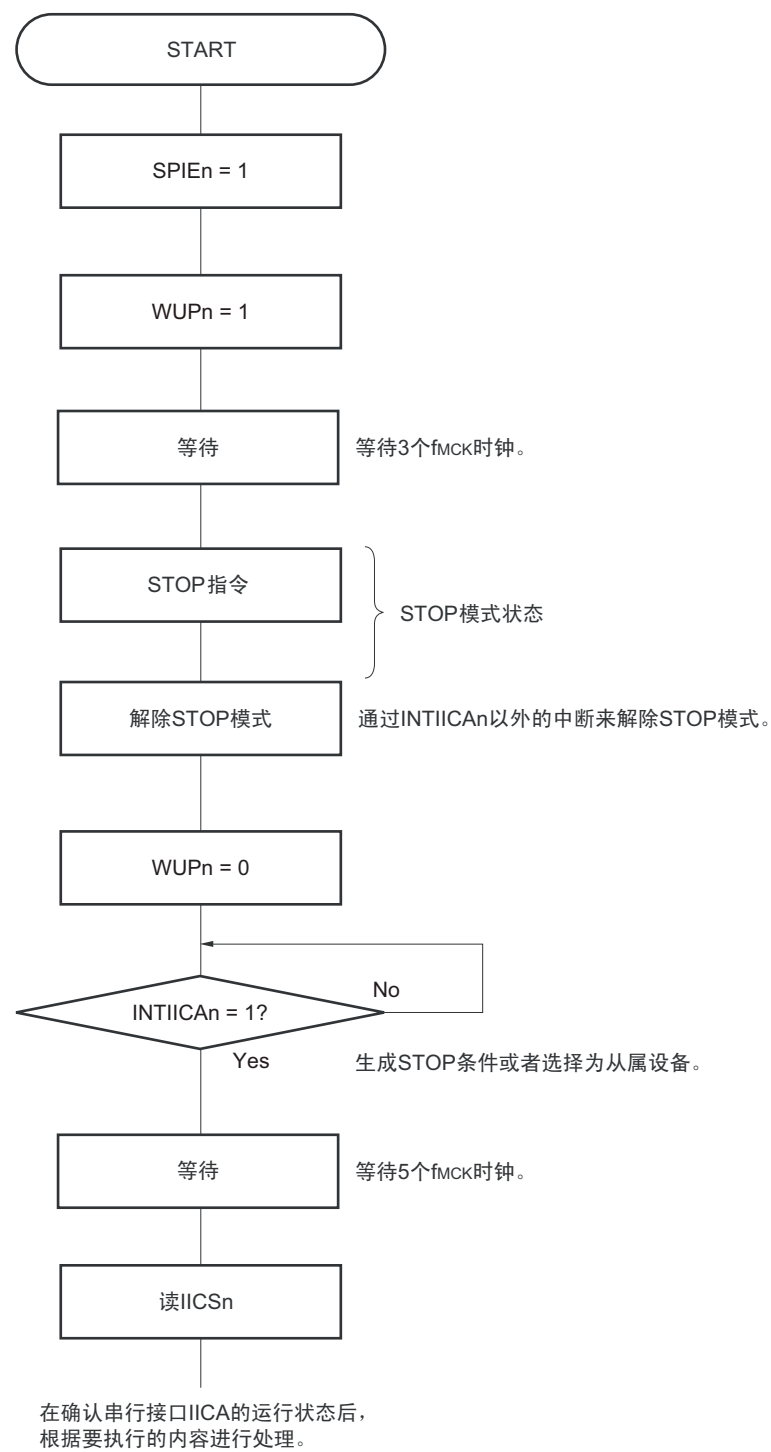


除了串行接口 IICA 产生的中断请求（INTIICAn）以外，必须通过以下的流程解除 STOP 模式。

- 下次IIC通信为主控设备运行的情况：图 15-24 的流程
- 下次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 15-23 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 15-24 在通过 INTIICAn 以外的方法解除 STOP 模式后作为主控设备运行的情况



备注 n=0

15.5.14 通信预约

(1) 允许通信预约功能的情况 (IICA 标志寄存器 n (IICFn) 的 bit0 (IICRSVn) =0)

要在不加入总线的状态下进行下次的主控通信时，能通过通信预约在释放总线时发送开始条件。此时的不加入总线包括以下 2 种状态：

- 在仲裁结果既不是主控设备也不是从属设备时
- 在接收到扩展码后不作为从属设备运行时（不返回应答而将 IICA 控制寄存器 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。

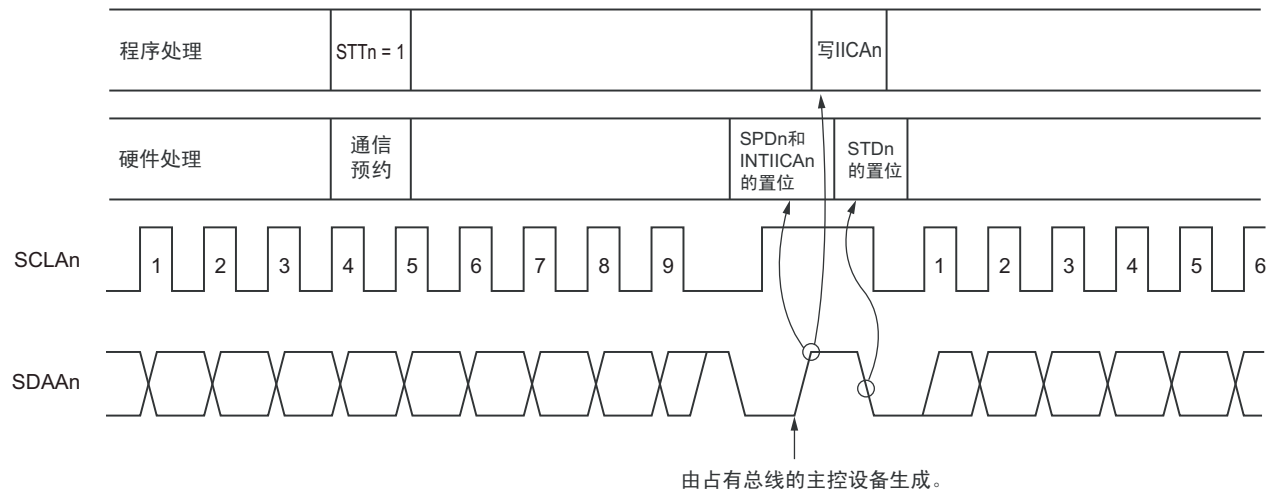
必须通过软件确保以下计算式计算的等待时间：

从将 STTn 位置“1”到确认 MSTSn 标志为止的等待时间 (f_{MCK} 时钟数)：
 $(IICWLn \text{ 的设定值} + IICWHn \text{ 的设定值} + 4) + t_F \times 2 \times f_{MCK} [\text{时钟}]$

- 备注 1. IICWLn : IICA 低电平宽度设定寄存器 n
 IICWHn : IICA 高电平宽度设定寄存器 n
 t_F : SDAAn 信号和 SCLAn 信号的下降时间
 f_{MCK} : IICA 运行时钟频率
 2. n=0

通信预约的时序如图 15-25 所示。

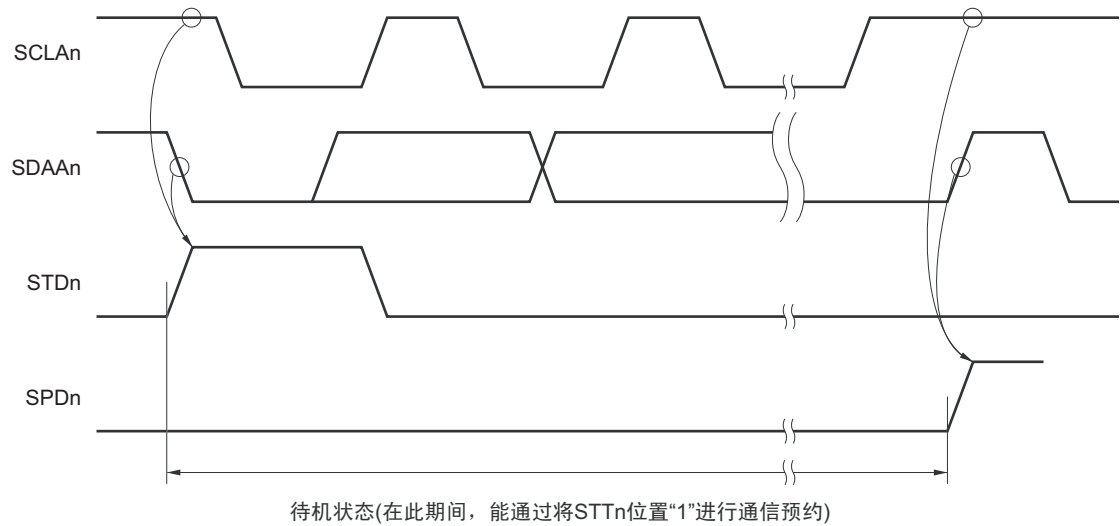
图 15-25 通信预约的时序



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 15-26 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。

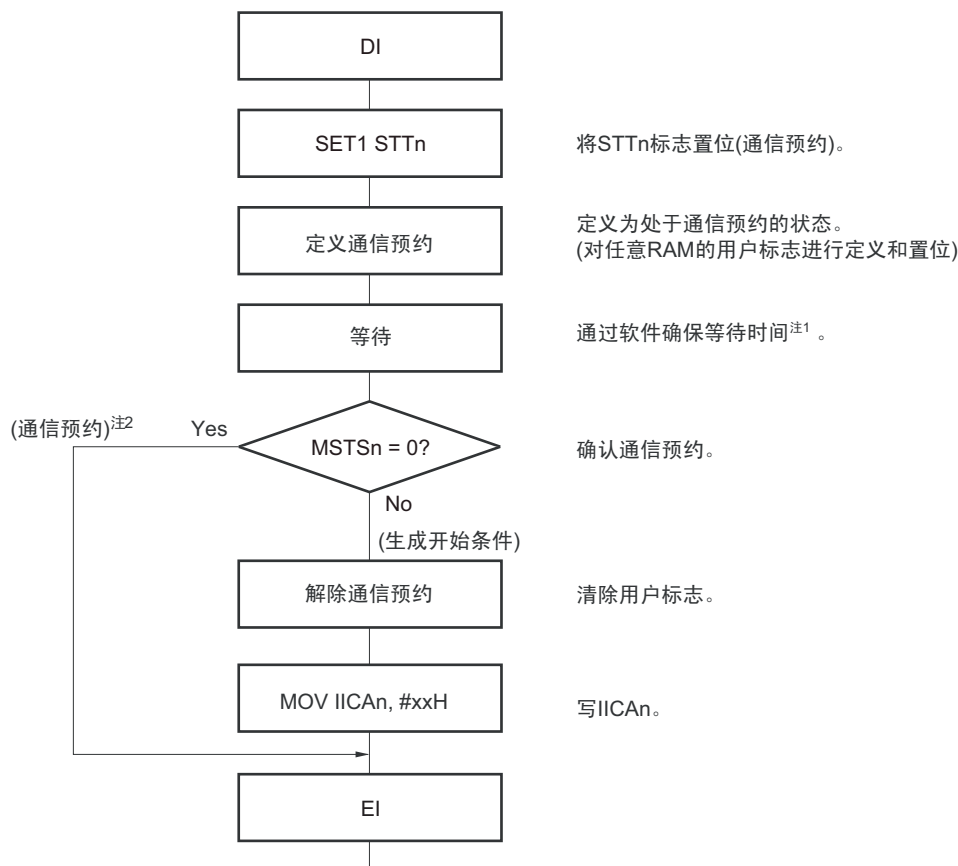
图 15-26 通信预约的接受时序



通信预约的步骤如图 15-27 所示。

备注 n=0

图 15-27 通信预约的步骤



注 1. 等待时间如下：

(IICWLn 的设定值 + IICWHn 的设定值 + 4) + $t_F \times 2 \times f_{MCK}$ [时钟]

2. 在通信预约运行时，通过停止条件中断请求写 IICA 移位寄存器 n (IICAn)。

备注 1. STTn : IICA 控制寄存器 n0 (IICCTLn0) 的 bit1

MSTS n : IICA 状态寄存器 n (IICS n) 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

15.5.15 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

15.5.16 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。

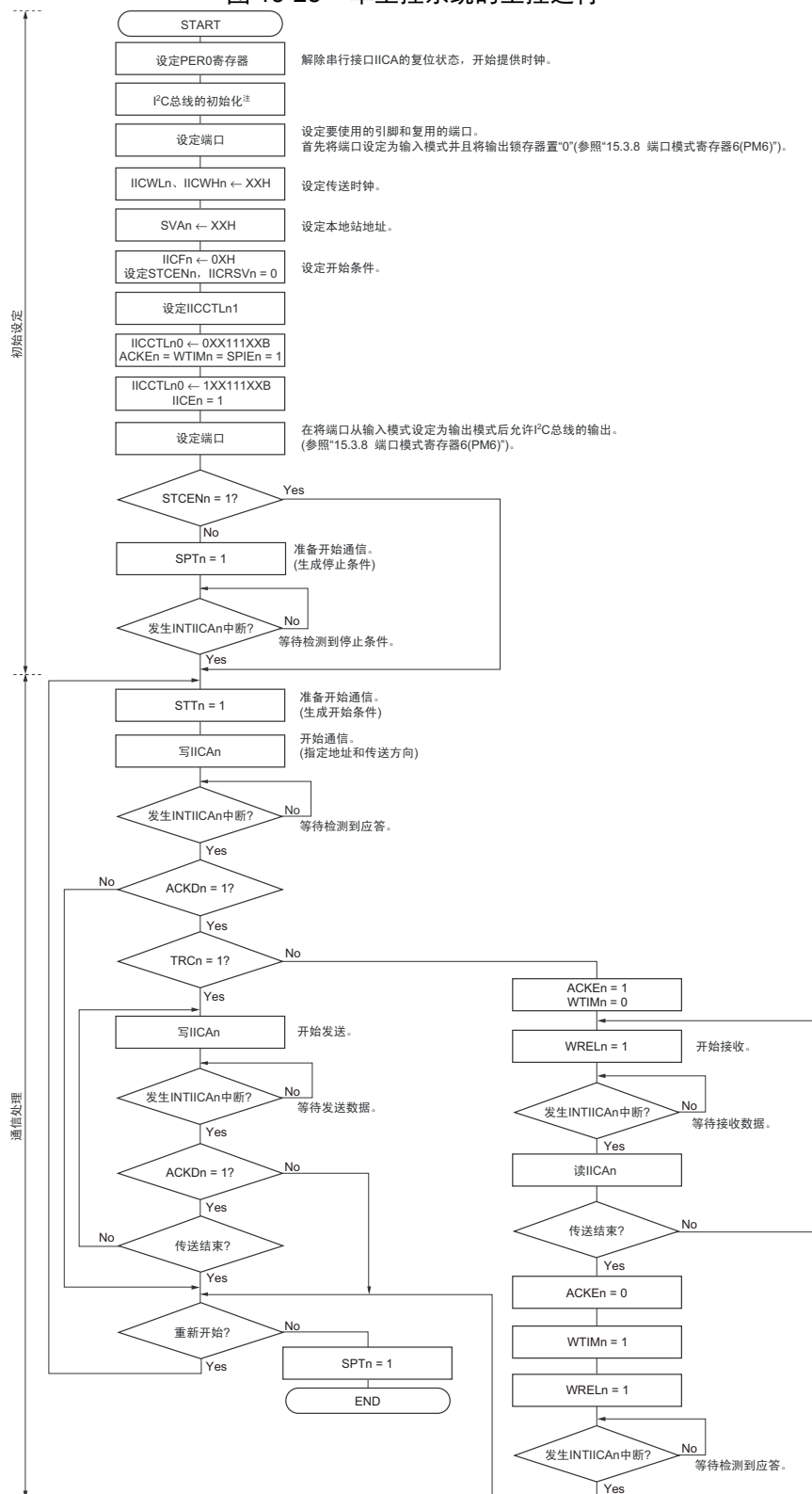
当用作从属设备时，通过中断开始运行。在启动时执行“初始设定”部分，然后通过“通信等待”，等待 INTIICAn 中断的发生。如果发生 INTIICAn 中断，就判断通信状态并且将标志传递给主处理部。

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 15-28 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 15-29 多主控系统的主控运行 (1/3)

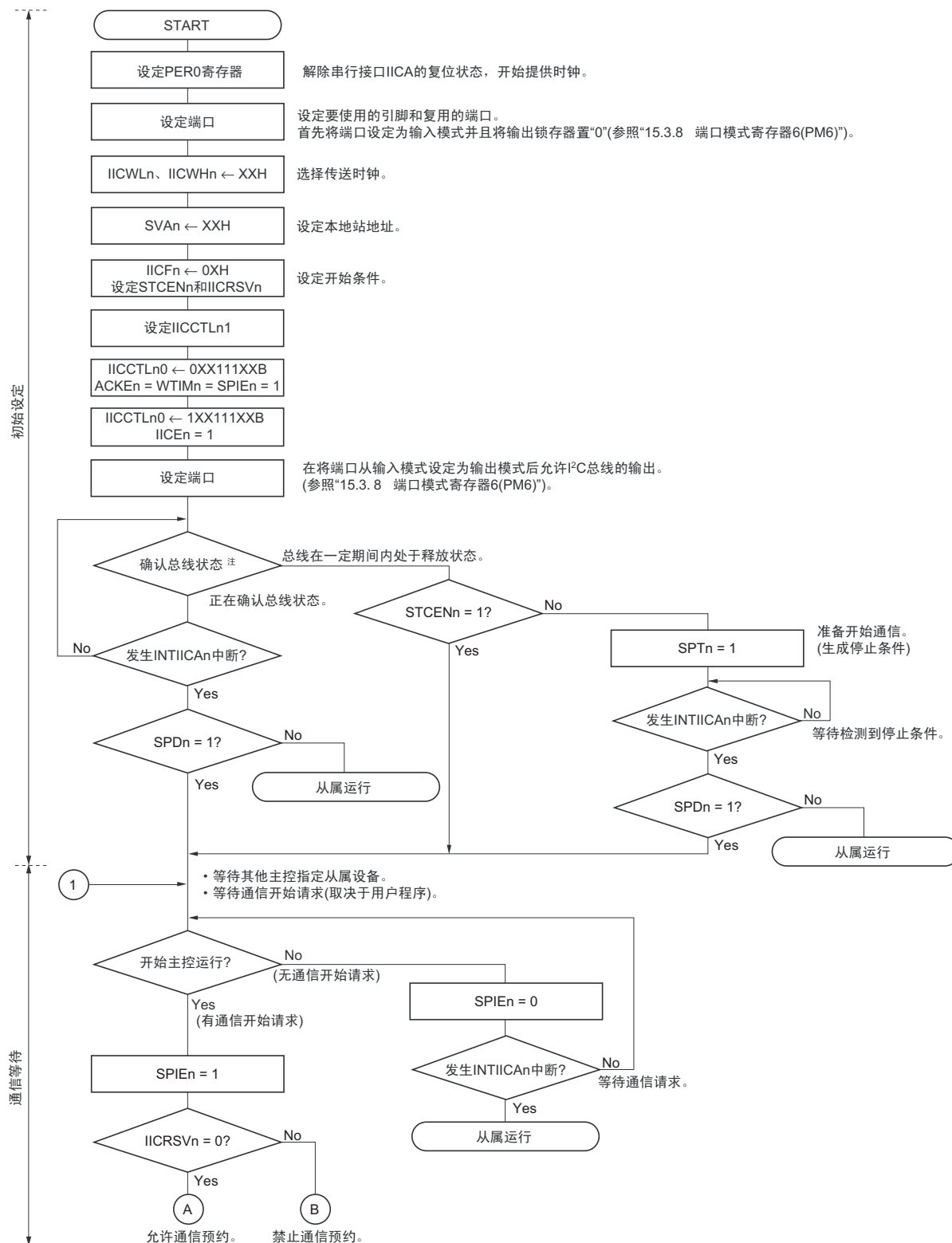
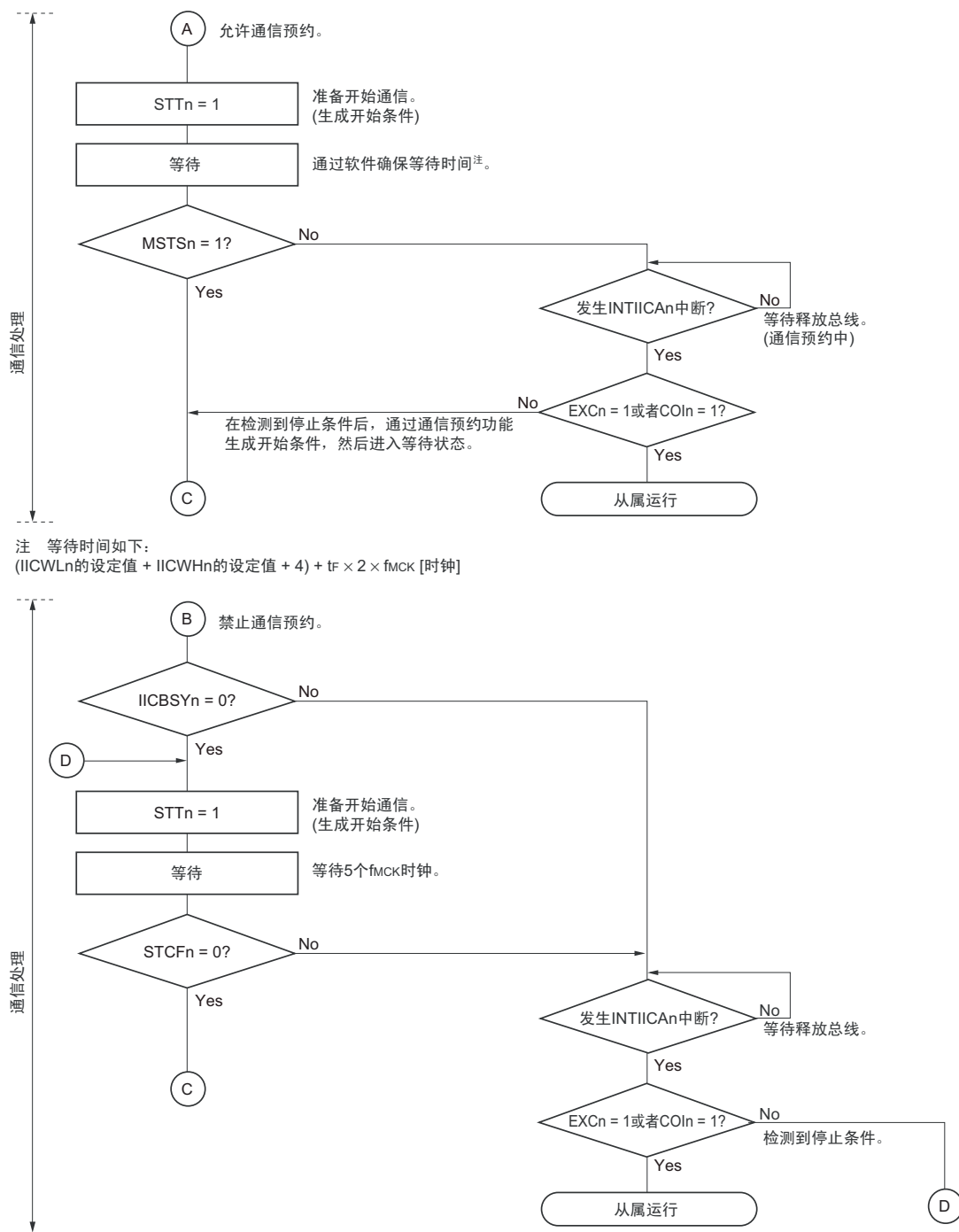
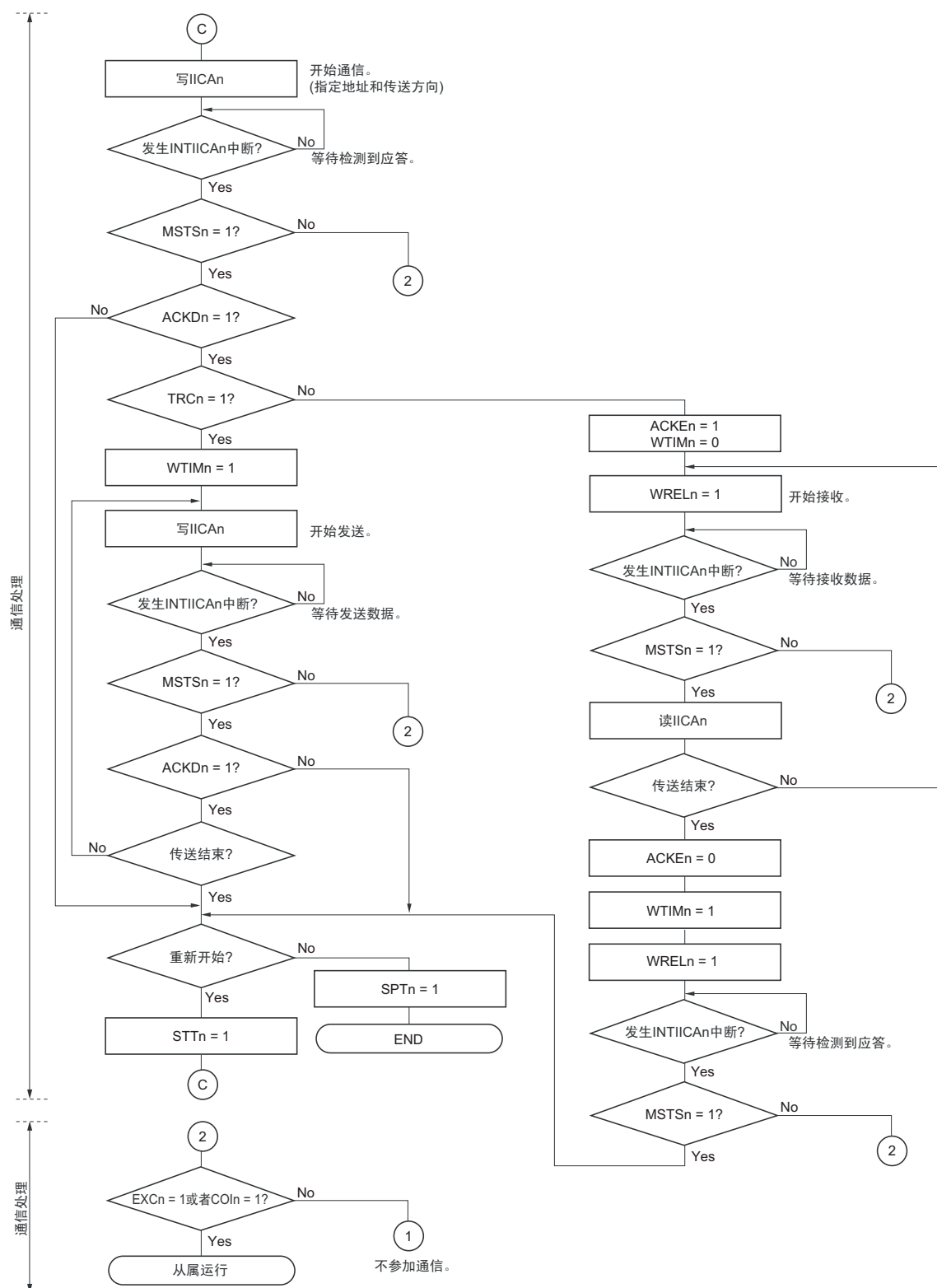


图 15-29 多主控系统的主控运行 (2/3)



- 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 15-29 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。

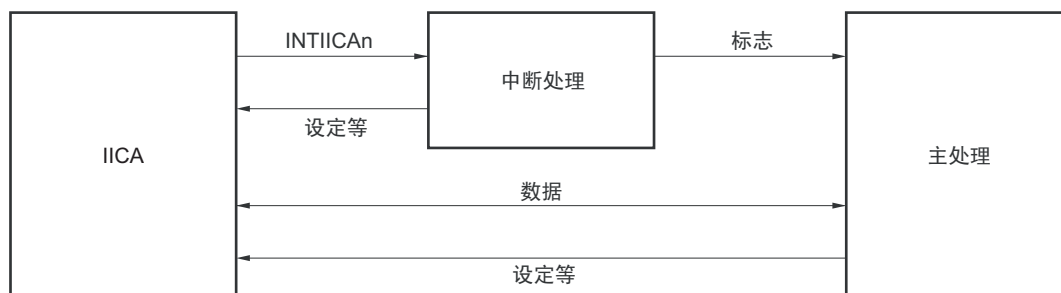
2. 在多主控系统中用作主控设备的情况下，必须在每次发生 INTIICAn 中断时读 MSTSn 位，确认仲裁结果。
3. 在多主控系统中用作从属设备的情况下，必须在每次发生 INTIICAn 中断时通过 IICA 状态寄存器 n (IICSn) 和 IICA 标志寄存器 n (IICFn) 确认状态，决定以后的处理。
4. n=0

(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次的请求）。

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

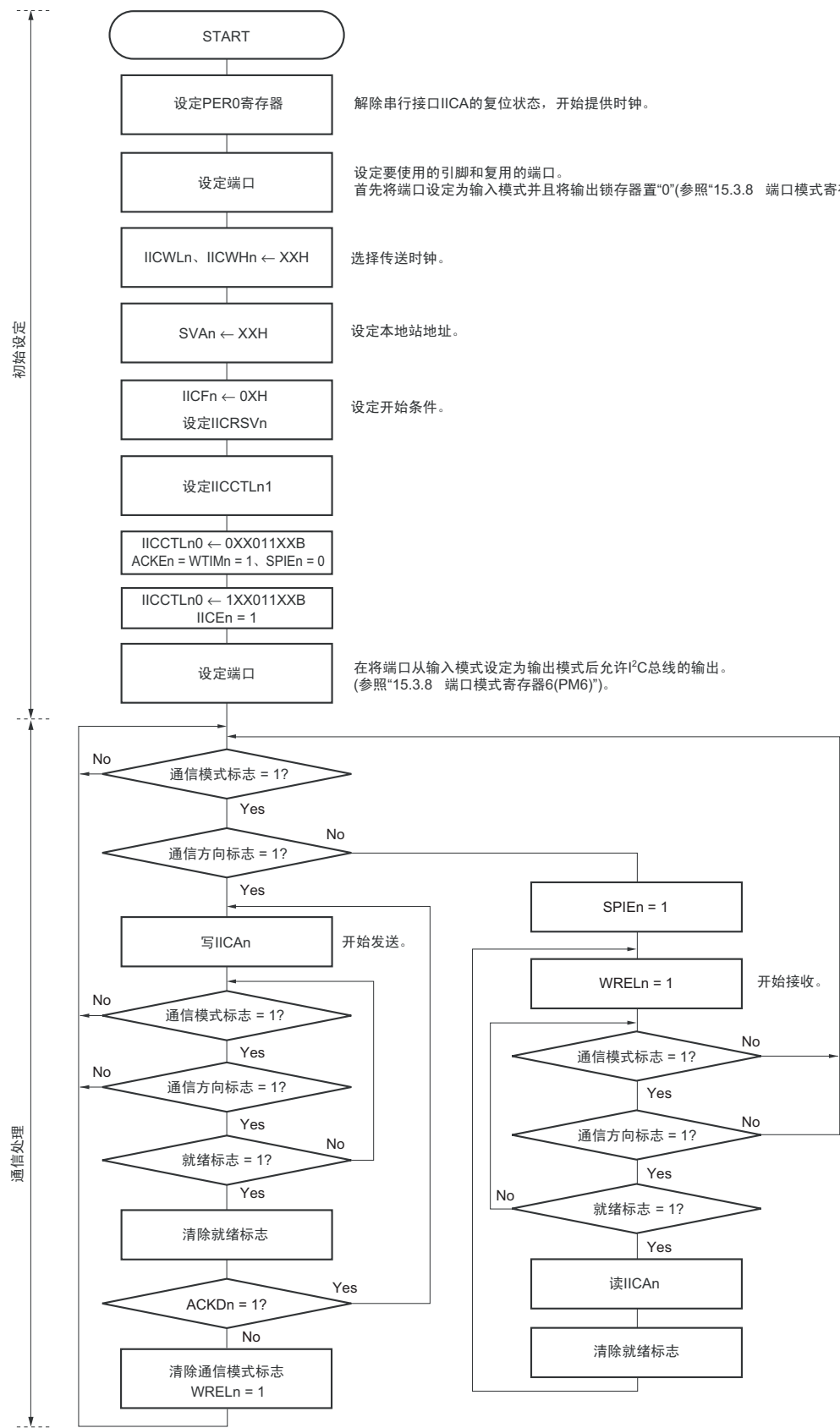
从属运行的主处理部的运行如下所示。

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 15-30 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。

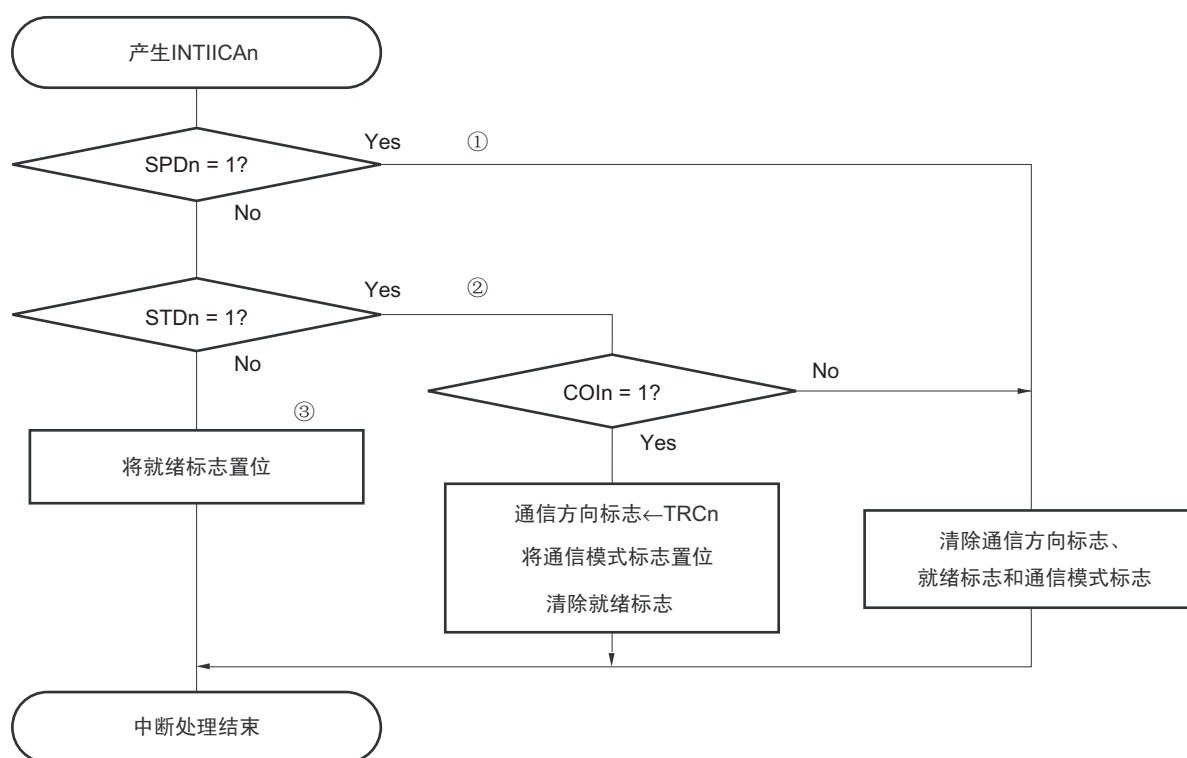
2. n=0

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 15-31 从属运行步骤 (2)”的①～③。

图 15-31 从属运行步骤 (2)



备注 n=0

15.5.17 I²C 中断请求 (INTIICAn) 的产生时序

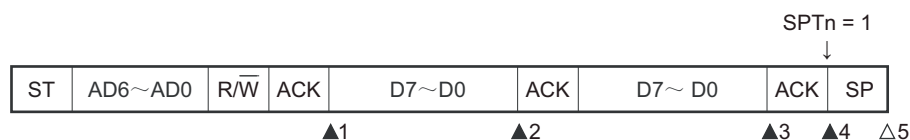
数据的发送和接收时序、INTIICAn 中断请求信号的产生时序以及产生 INTIICAn 信号时的 IICA 状态寄存器 n (IICSn) 的值如下所示。

- 备注 1. ST : 开始条件
AD6 ~ AD0 : 地址
 $\overline{R/W}$: 传送方向的指定
ACK : 应答
D7 ~ D0 : 数据
SP : 停止条件
2. n=0

(1) 主控运行

(a) Start ~ Address ~ Data ~ Data ~ Stop (发送和接收)

(i) WTIMn=0 的情况



▲ 1: IICSn=1000×110B

▲ 2: IICSn=1000×000B

▲ 3: IICSn=1000×000B (将 WTIMn 位置“1”) 注

▲ 4: IICSn=1000××00B (将 SPTn 位置“1”)

△ 5: IICSn=00000001B

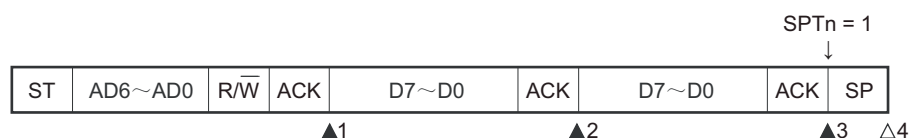
注 为了生成停止条件，必须将 WTIMn 位置“1”并且更改 INTIICAn 中断请求信号的产生时序。

备注 ▲ 一定产生。

 △ 只在 SPIEn 位为“1”时产生。

 × 任意

(ii) WTIMn=1 的情况



▲ 1: IICSn=1000×110B

▲ 2: IICSn=1000×100B

▲ 3: IICSn=1000××00B (将 SPTn 位置“1”)

△ 4: IICSn=00000001B

备注 ▲ 一定产生。

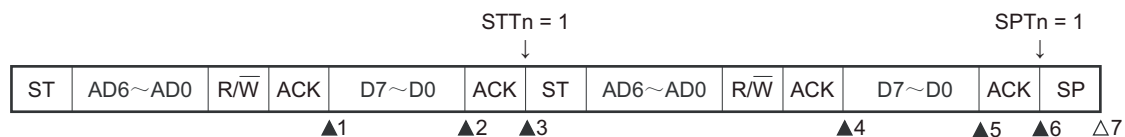
 △ 只在 SPIEn 位为“1”时产生。

 × 任意

备注 n=0

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (重新开始)

(i) WTIMn=0 的情况

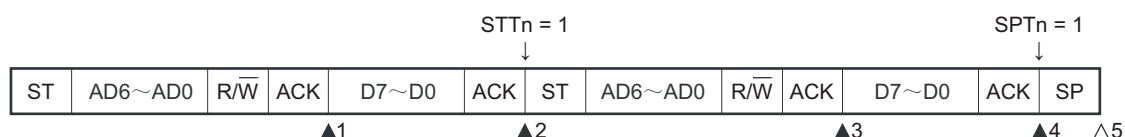


- ▲ 1: IICSn=1000×110B
- ▲ 2: IICSn=1000×000B (将 WTIMn 位置“1”) 注 1
- ▲ 3: IICSn=1000××00B (将 WTIMn 位清“0”注 2 并且将 STTn 位置“1”)
- ▲ 4: IICSn=1000×110B
- ▲ 5: IICSn=1000×000B (将 WTIMn 位置“1”) 注 3
- ▲ 6: IICSn=1000××00B (将 SPTn 位置“1”)
- △ 7: IICSn=00000001B

- 注 1. 为了生成开始条件, 必须将 WTIMn 位置“1”并且更改 INTIICAn 中断请求信号的产生时序。
2. 为了恢复原来的设定, 必须将 WTIMn 位清“0”。
3. 为了生成停止条件, 必须将 WTIMn 位置“1”并且更改 INTIICAn 中断请求信号的产生时序。

备注 ▲ 一定产生
△ 只在 SPIEn 位为“1”时产生。
× 任意

(ii) WTIMn=1 的情况



- ▲ 1: IICSn=1000×110B
- ▲ 2: IICSn=1000××00B (将 STTn 位置“1”)
- ▲ 3: IICSn=1000×110B
- ▲ 4: IICSn=1000××00B (将 SPTn 位置“1”)
- △ 5: IICSn=00000001B

备注 ▲ 一定产生。
△ 只在 SPIEn 位为“1”时产生。
× 任意

备注 n=0

(c) Start ~ Code ~ Data ~ Data ~ Stop (发送扩展码)

(i) WTIMn=0 的情况

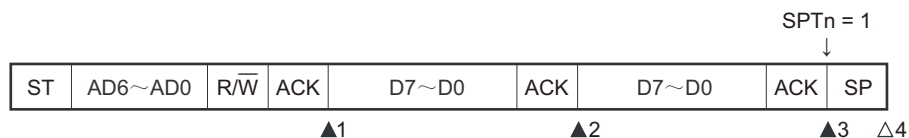


- ▲ 1: IICSn=1010×110B
 ▲ 2: IICSn=1010×000B
 ▲ 3: IICSn=1010×000B (将 WTIMn 位置“1”) 注
 ▲ 4: IICSn=1010××00B (将 SPTn 位置“1”)
 △ 5: IICSn=00000001B

注 为了生成停止条件，必须将 WTIMn 位置“1”并且更改 INTIICAn 中断请求信号的产生时序。

- 备注 ▲ 一定产生。
 △ 只在 SPIEn 位为“1”时产生。
 × 任意

(ii) WTIMn=1 的情况



- ▲ 1: IICSn=1010×110B
 ▲ 2: IICSn=1010×100B
 ▲ 3: IICSn=1010××00B (将 SPTn 位置“1”)
 △ 4: IICSn=00001001B

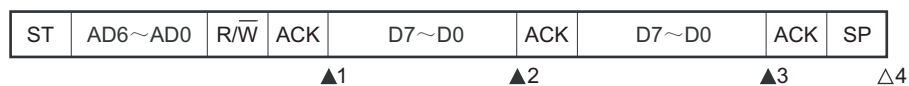
- 备注 ▲ 一定产生。
 △ 只在 SPIEn 位为“1”时产生。
 × 任意

备注 n=0

(2) 从属运行（接收从属地址的情况）

(a) Start ~ Address ~ Data ~ Data ~ Stop

(i) WTIMn=0 的情况



▲ 1: IICSn=0001×110B

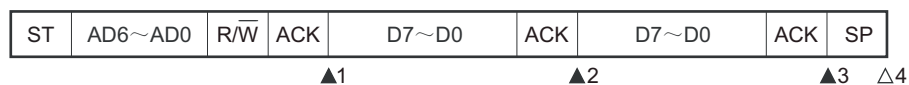
▲ 2: IICSn=0001×000B

▲ 3: IICSn=0001×000B

△ 4: IICSn=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIEn 位为“1”时产生。
 × 任意

(ii) WTIMn=1 的情况



▲ 1: IICSn=0001×110B

▲ 2: IICSn=0001×100B

▲ 3: IICSn=0001××00B

△ 4: IICSn=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIEn 位为“1”时产生。
 × 任意

备注 n=0

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIMn=0 的情况（在重新开始后 SVAn 相同）



▲ 1: IICSn=0001×110B

▲ 2: IICSn=0001×000B

▲ 3: IICSn=0001×110B

▲ 4: IICSn=0001×000B

△ 5: IICSn=00000001B

备注 ▲ 一定产生。

△ 只在 SPIEn 位为“1”时产生。

× 任意

(ii) WTIMn=1 的情况（在重新开始后 SVAn 相同）



▲ 1: IICSn=0001×110B

▲ 2: IICSn=0001××00B

▲ 3: IICSn=0001×110B

▲ 4: IICSn=0001××00B

△ 5: IICSn=00000001B

备注 ▲ 一定产生。

△ 只在 SPIEn 位为“1”时产生。

× 任意

备注 n=0

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) WTIMn=0 的情况（在重新开始后地址不同（扩展码））



▲ 1: IICSn=0001×110B

▲ 2: IICSn=0001×000B

▲ 3: IICSn=0010×010B

▲ 4: IICSn=0010×000B

△ 5: IICSn=00000001B

备注 ▲ 一定产生。
 △ 只在SPIEn位为“1”时产生。
 × 任意

(ii) WTIMn=1 的情况（在重新开始后地址不同（扩展码））



▲ 1: IICSn=0001×110B

▲ 2: IICSn=0001××00B

▲ 3: IICSn=0010×010B

▲ 4: IICSn=0010×110B

▲ 5: IICSn=0010××00B

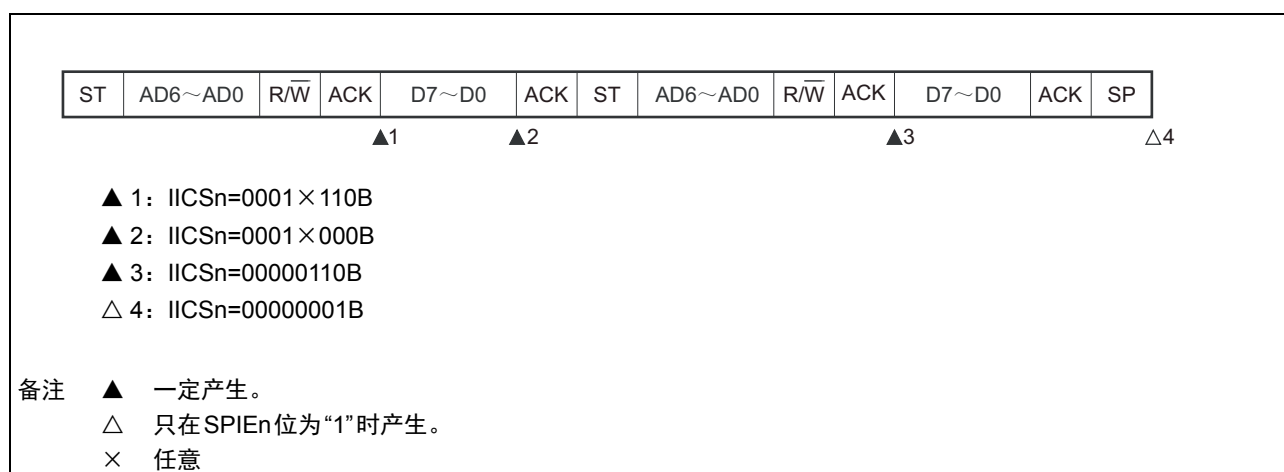
△ 6: IICSn=00000001B

备注 ▲ 一定产生。
 △ 只在SPIEn位为“1”时产生。
 × 任意

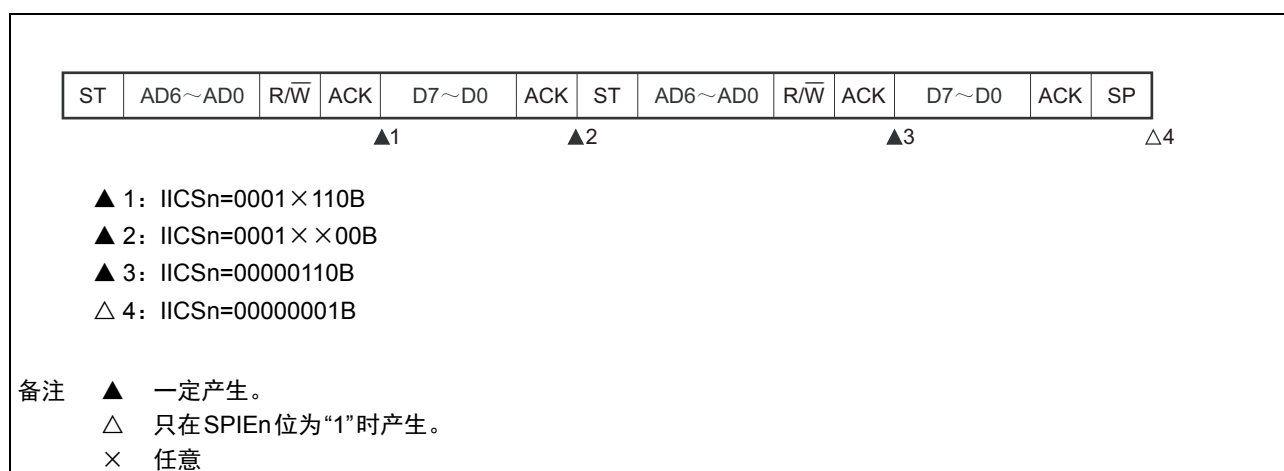
备注 n=0

(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））



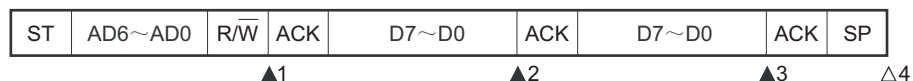
备注 n=0

(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



▲ 1: IICSn=0010×010B

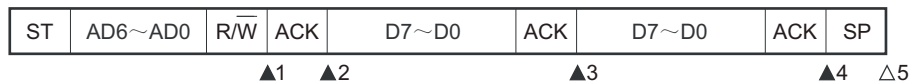
▲ 2: IICSn=0010×000B

▲ 3: IICSn=0010×000B

△ 4: IICSn=00000001B

备注 ▲ 一定产生。
△ 只在 SPIEn 位为“1”时产生。
× 任意

(ii) WTIMn=1 的情况



▲ 1: IICSn=0010×010B

▲ 2: IICSn=0010×110B

▲ 3: IICSn=0010×100B

▲ 4: IICSn=0010××00B

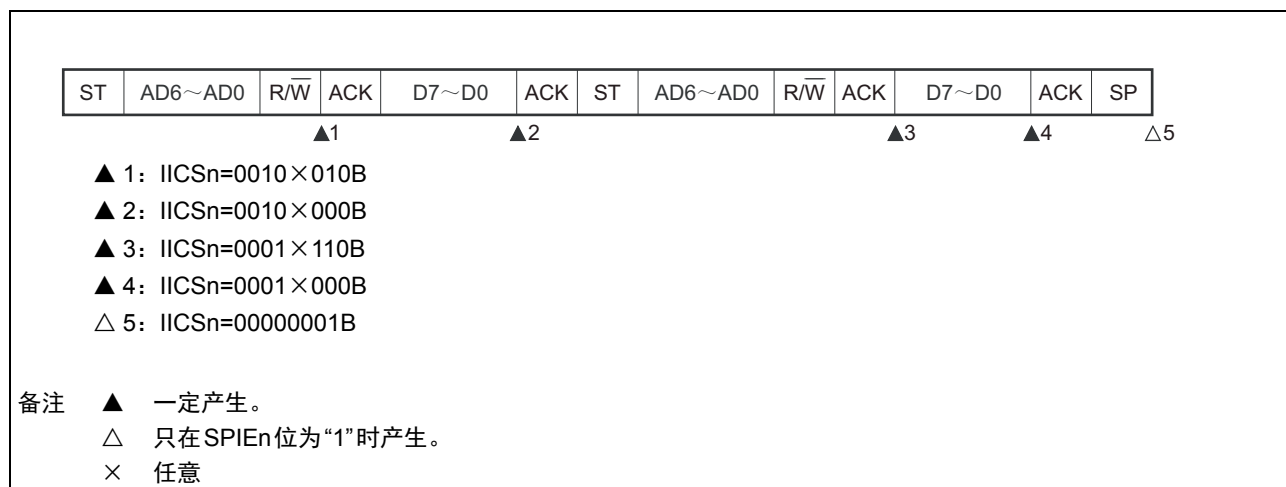
△ 5: IICSn=00000001B

备注 ▲ 一定产生。
△ 只在 SPIEn 位为“1”时产生。
× 任意

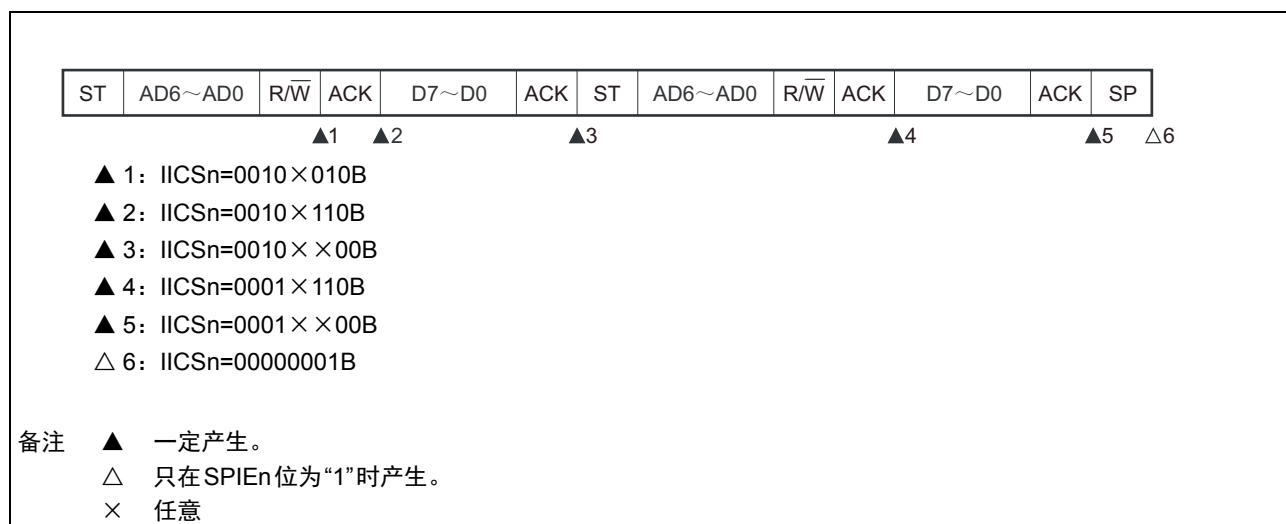
备注 n=0

(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）



▲ 1: IICSn=0010×010B

▲ 2: IICSn=0010×000B

▲ 3: IICSn=0010×010B

▲ 4: IICSn=0010×000B

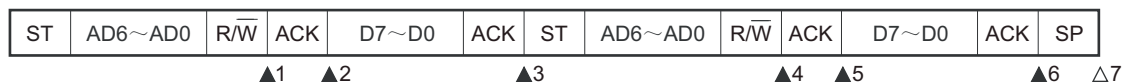
△ 5: IICSn=00000001B

备注 ▲ 一定产生。

△ 只在SPIEn位为“1”时产生。

× 任意

(ii) WTIMn=1 的情况（在重新开始后接收扩展码）



▲ 1: IICSn=0010×010B

▲ 2: IICSn=0010×110B

▲ 3: IICSn=0010××00B

▲ 4: IICSn=0010×010B

▲ 5: IICSn=0010×110B

▲ 6: IICSn=0010××00B

△ 7: IICSn=00000001B

备注 ▲ 一定产生。

△ 只在SPIEn位为“1”时产生。

× 任意

备注 n=0

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIMn=0 的情况（在重新开始后地址不同（非扩展码））



▲ 1: IICSn=0010×010B

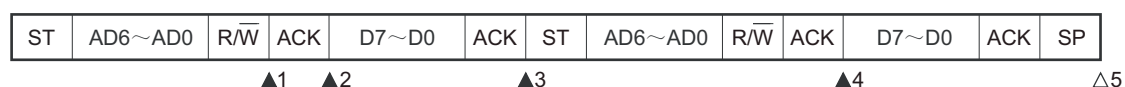
▲ 2: IICSn=0010×000B

▲ 3: IICSn=00000×10B

△ 4: IICSn=00000001B

备注 ▲ 一定产生。
 △ 只在SPIEn位为“1”时产生。
 × 任意

(ii) WTIMn=1 的情况（在重新开始后地址不同（非扩展码））



▲ 1: IICSn=0010×010B

▲ 2: IICSn=0010×110B

▲ 3: IICSn=0010××00B

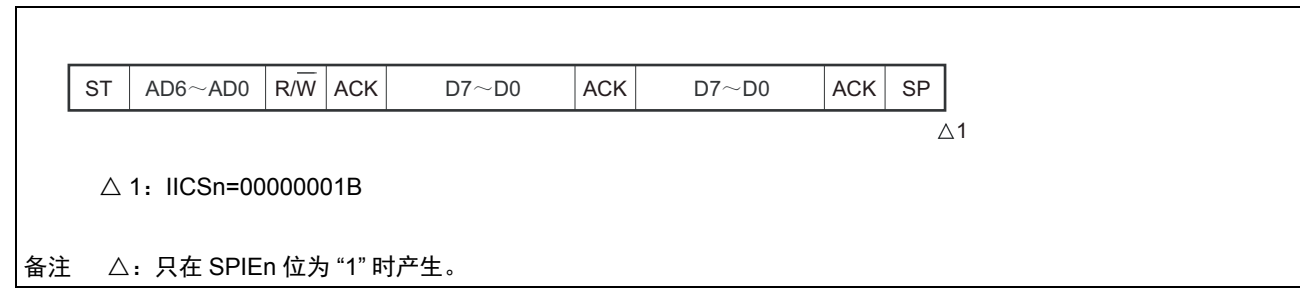
▲ 4: IICSn=00000×10B

△ 5: IICSn=00000001B

备注 ▲ 一定产生。
 △ 只在SPIEn位为“1”时产生。
 × 任意

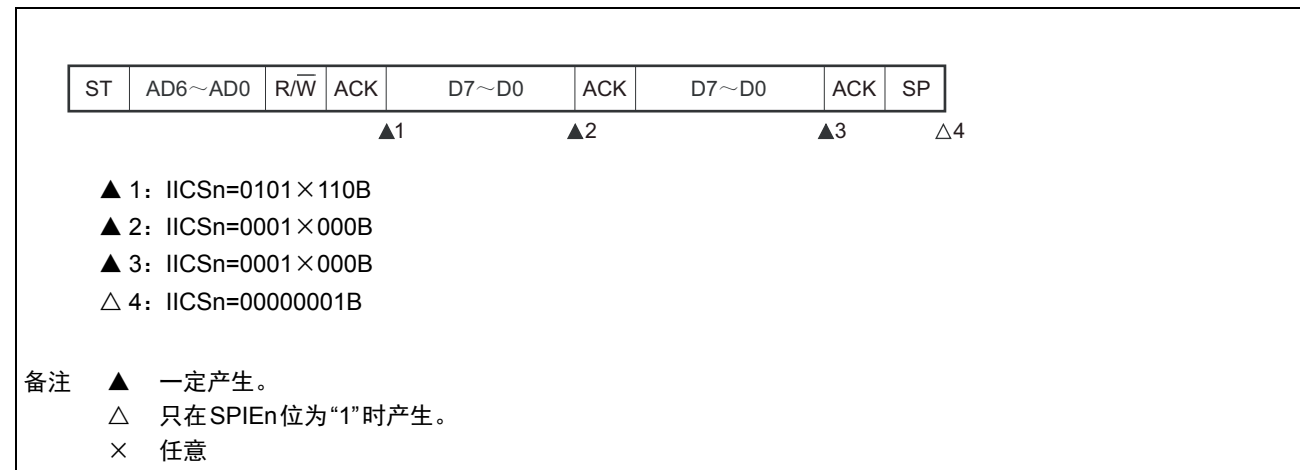
备注 n=0

- (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

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	D7~D0	ACK	SP
			▲1		▲2		▲3	△4

▲ 1: IICSn=0101×110B
▲ 2: IICSn=0001×100B
▲ 3: IICSn=0001××00B
△ 4: IICSn=00000001B

备注 ▲ 一定产生。
 △ 只在SPIEn位为“1”时产生。
 × 任意

(b) 在发送扩展码的过程中仲裁失败的情况

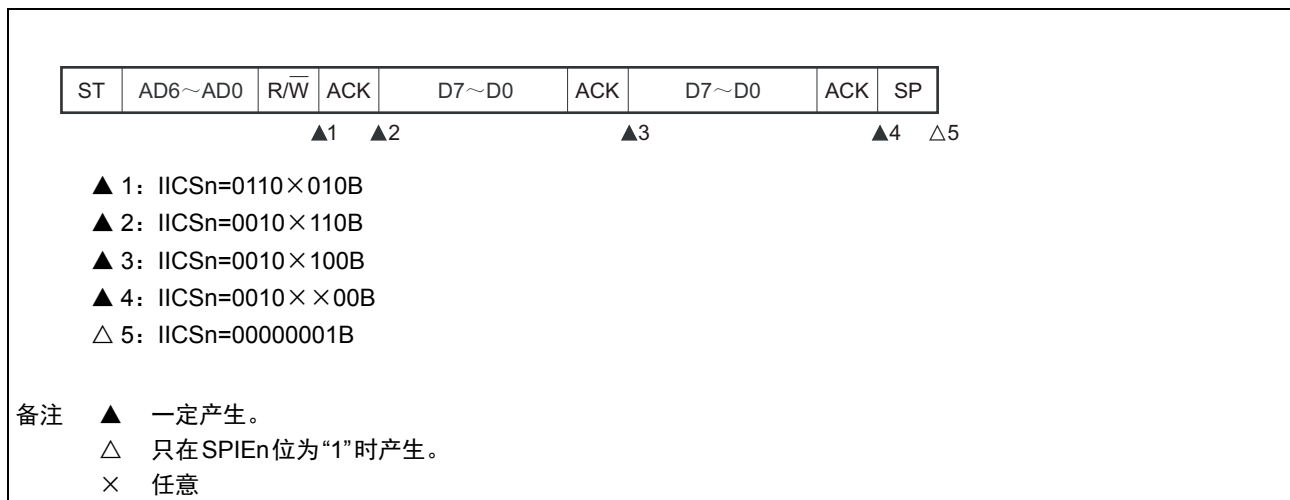
(i) WTIMn=0 的情况

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	D7~D0	ACK	SP
			▲1		▲2		▲3	△4

▲ 1: IICSn=0110×010B
▲ 2: IICSn=0010×000B
▲ 3: IICSn=0010×000B
△ 4: IICSn=00000001B

备注 ▲ 一定产生。
 △ 只在SPIEn位为“1”时产生。
 × 任意

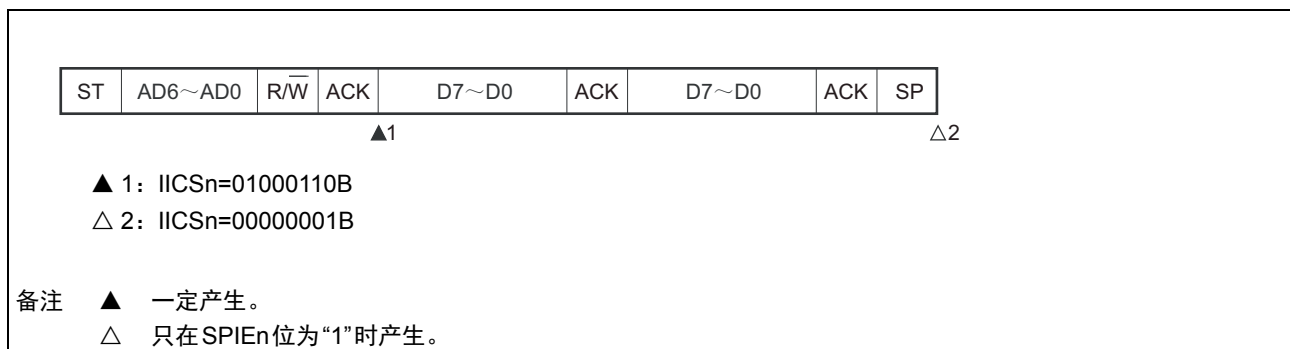
备注 n=0

(ii) WTIM_n=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

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	D7~D0	ACK	SP

▲1

△2

▲ 1: IICSn=0110×010B
通过软件将 LRELn 位置“1”。
△ 2: IICSn=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIEn 位为“1”时产生。
 × 任意

(c) 在传送数据时仲裁失败的情况

(i) WTIMn=0 的情况

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	D7~D0	ACK	SP

▲1

▲2

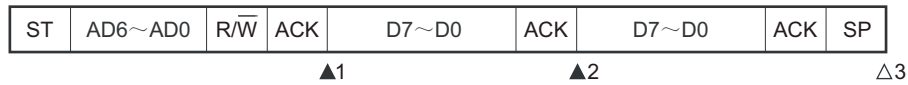
△3

▲ 1: IICSn=10001110B
▲ 2: IICSn=01000000B
△ 3: IICSn=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIEn 位为“1”时产生。

备注 n=0

(ii) WTIM_{n=1}



▲ 1: IICSn=10001110B

▲ 2: IICSn=01000100B

△ 3: IICSn=00000001B

备注 ▲ 一定产生。

△ 只在SPIEn位为“1”时产生。

(d) 在传送数据时因重新开始条件而仲裁失败的情况

(i) 非扩展码 (例如, $SVAn$ 不同)



▲ 1: IICS_n=1000×110B

▲ 2: IICSn=01000110B

△ 3: IICSn=00000001B

备注 ▲ 一定产生。

△ 只在SPIEn位为“1”时产生。

× 任意

 $m=6 \sim 0$

备注 n=0

(ii) 扩展码

ST	AD6~AD0	R/W	ACK	D7~Dm	ST	AD6~AD0	R/W	ACK	D7~D0	ACK	SP
			▲1				▲2			△3	

▲ 1: IICSn=1000×110B
▲ 2: IICSn=01100010B
通过软件将 LRELn 位置“1”。
△ 3: IICSn=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIEn 位为“1”时产生。
 × 任意
 m=6~0

(e) 在传送数据时因停止条件而仲裁失败的情况

ST	AD6~AD0	R/W	ACK	D7~Dm	SP
			▲1		△2

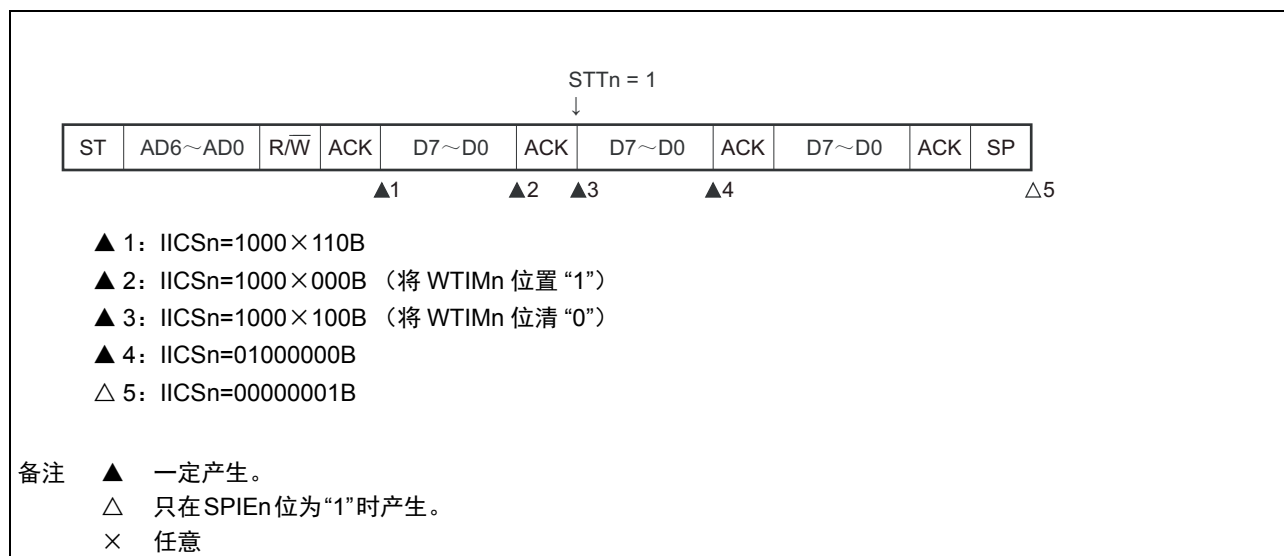
▲ 1: IICSn=10000110B
△ 2: IICSn=01000001B

备注 ▲ 一定产生。
 △ 只在 SPIEn 位为“1”时产生。
 × 任意
 m=6~0

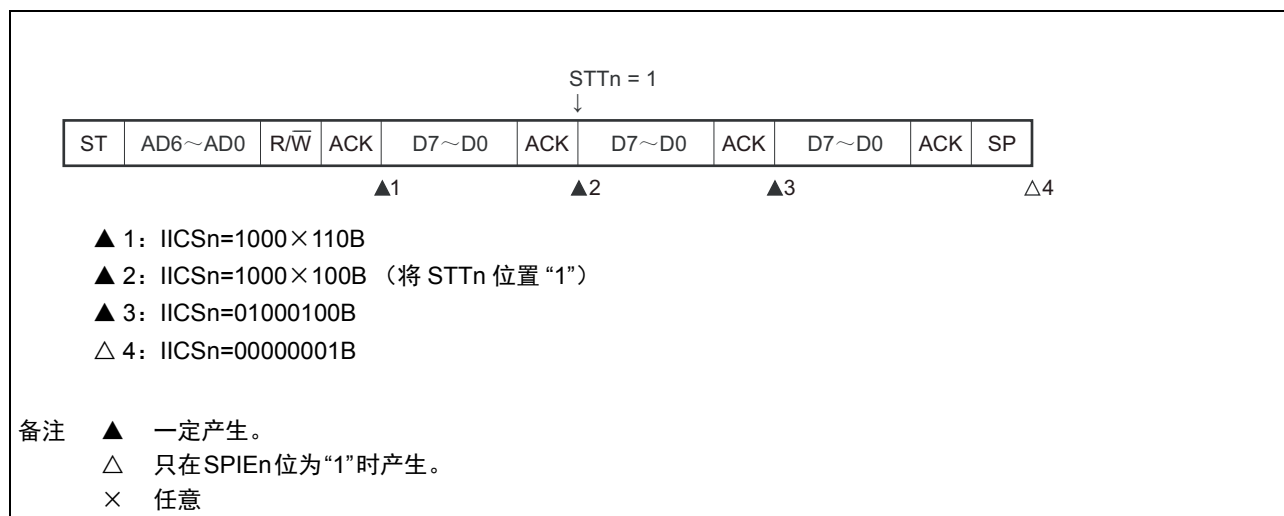
备注 n=0

(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



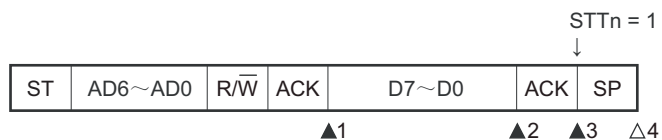
(ii) WTIMn=1 的情况



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

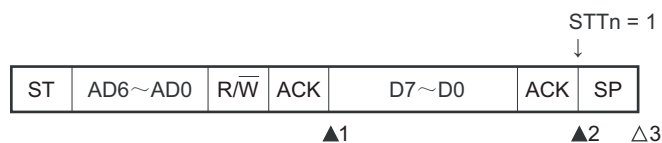
(i) WTIMn=0 的情况



- ▲ 1: IICSn=1000×110B
- ▲ 2: IICSn=1000×000B (将 WTIMn 位置“1”)
- ▲ 3: IICSn=1000××00B (将 STTn 位置“1”)
- △ 4: IICSn=01000001B

备注 ▲ 一定产生。
△ 只在 SPIEn 位为“1”时产生。
× 任意

(ii) WTIMn=1 的情况



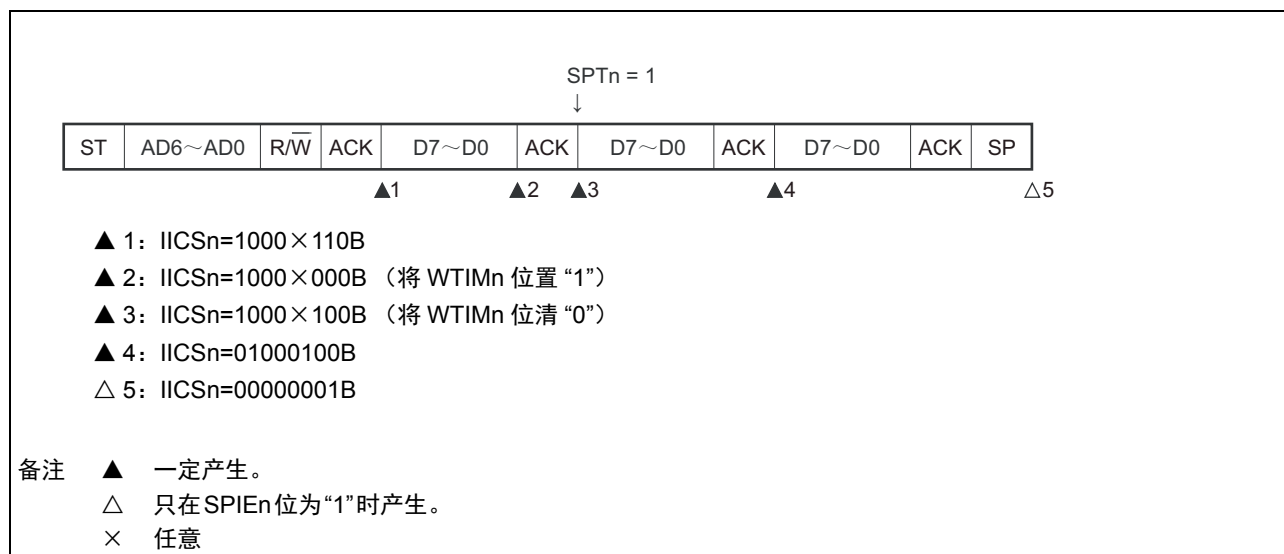
- ▲ 1: IICSn=1000×110B
- ▲ 2: IICSn=1000××00B (将 STTn 位置“1”)
- △ 3: IICSn=01000001B

备注 ▲ 一定产生。
△ 只在 SPIEn 位为“1”时产生。
× 任意

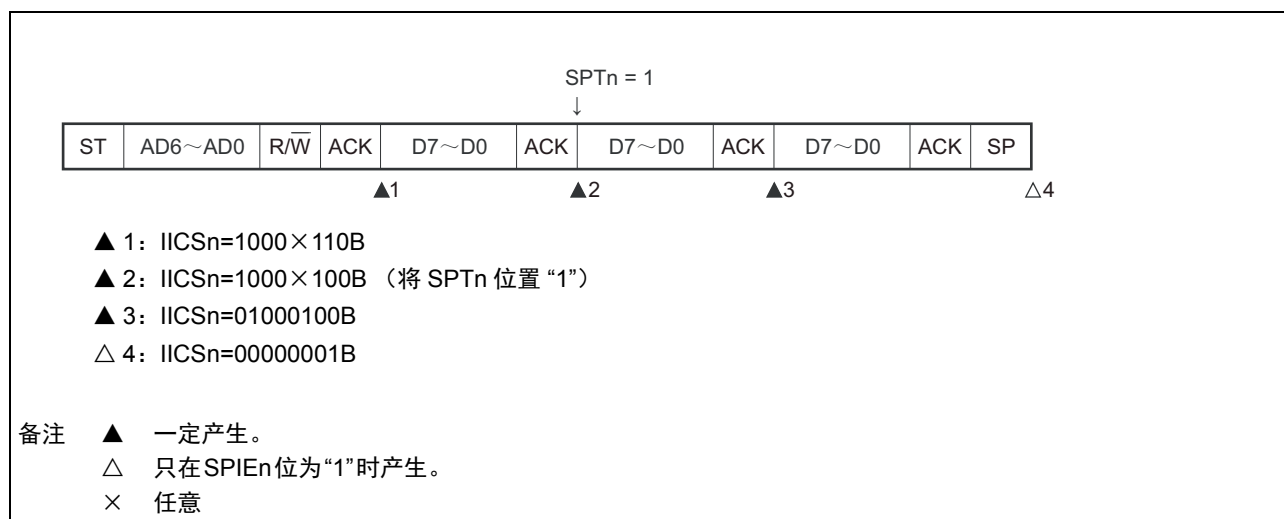
备注 n=0

(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

15.6 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 15-32 和图 15-33 所示。

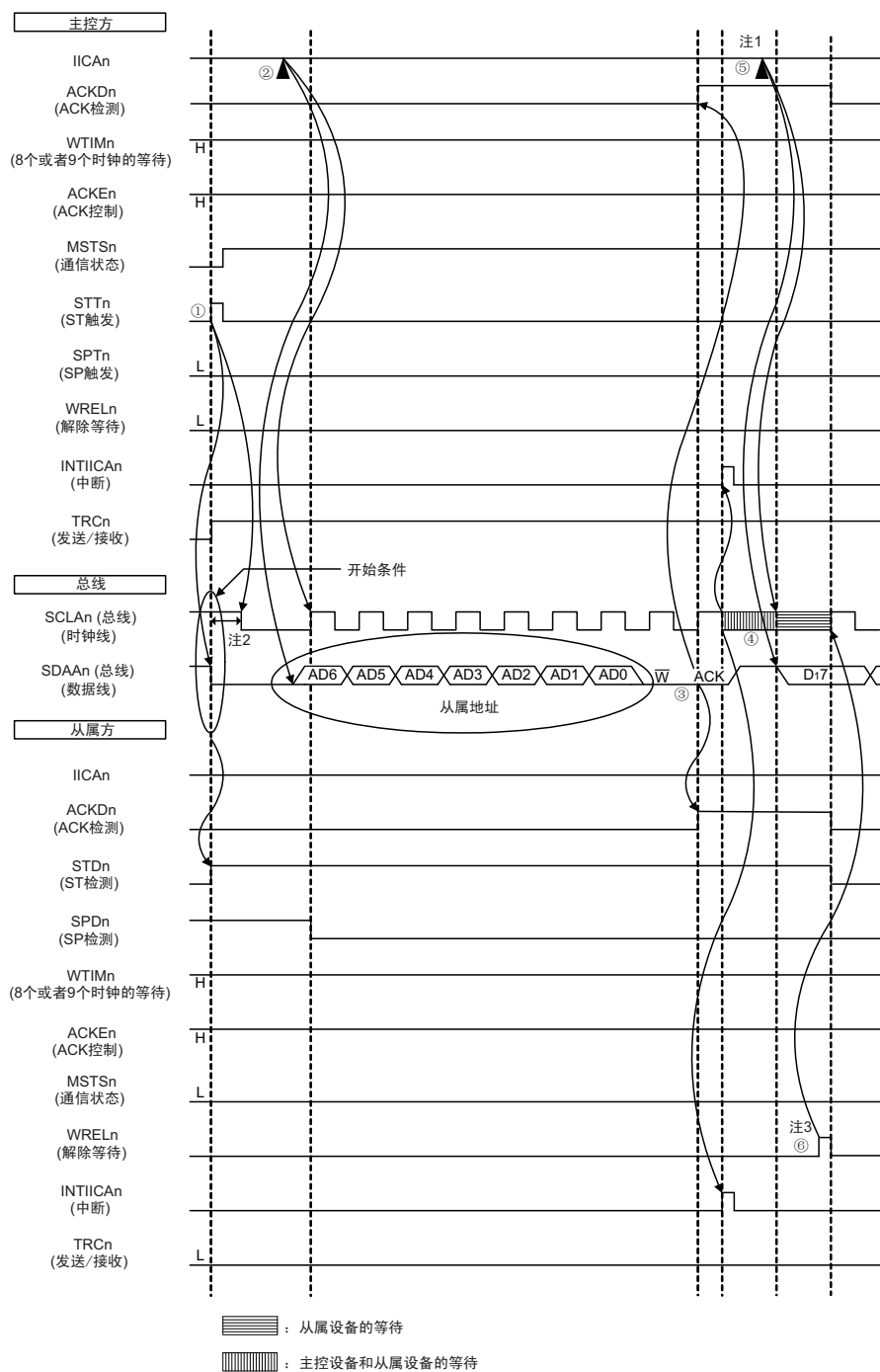
与串行时钟（SCLAn）的下降沿同步进行 IICA 移位寄存器 n（IICAn）的移位，并且将发送数据传送到 SO 锁存器，以 MSB 优先从 SDAAn 引脚输出数据。

在 SCLAn 的上升沿将 SDAAn 引脚输入的数据取到 IICAn。

备注 n=0

图 15-32 主控设备 → 从属设备的通信例子
 (主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待) (1/4)

(1) 开始条件～地址～数据



- 注 1. 要在主控方的发送期间解除等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。
2. 从 SDAAn 引脚信号下降到 SCLAn 引脚信号下降的时间，在设定为标准模式时至少为 4.0μs，在设定为快速模式时至少为 0.6μs。
3. 要在从属方的接收期间解除等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

备注 n=0

图 15-32 的“(1) 开始条件～地址～数据”的①～⑥的说明如下：

- ① 如果在主控方将开始条件触发置位（STTn=1），总线数据线（SDAAn）就下降，生成开始条件（通过 SCLAn=1 使 SDAAn 从“1”变为“0”）。此后，如果检测到开始条件，主控方就进入主控通信状态（MSTS_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+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 15-32 的①～⑮是通过 I²C 总线进行数据通信的一系列运行步骤。

图 15-32 的“(1) 开始条件～地址～数据”说明步骤①～⑥。

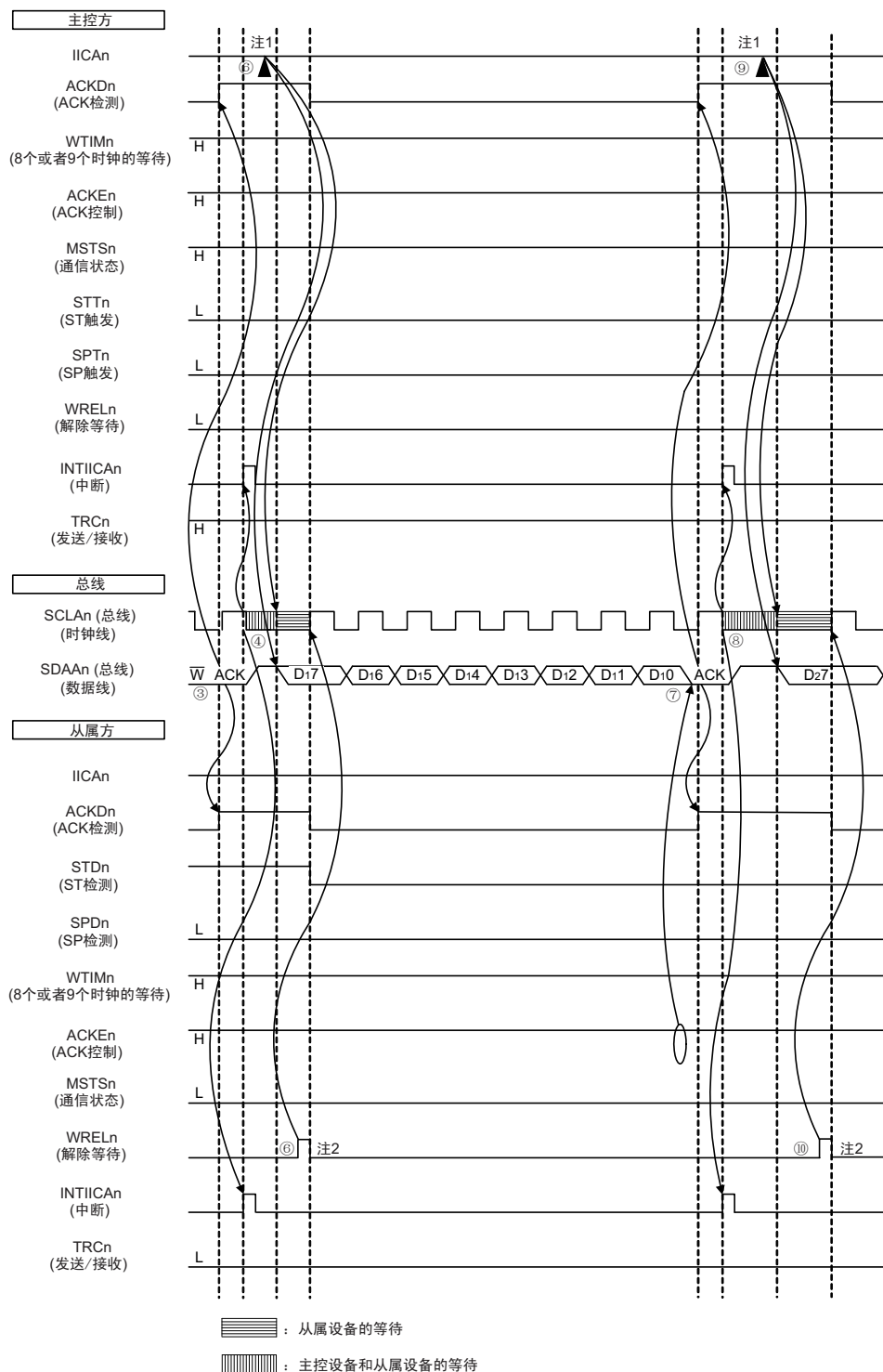
图 15-32 的“(2) 地址～数据～数据”说明步骤③～⑩。

图 15-32 的“(3) 数据～数据～停止条件”说明步骤⑦～⑮。

2. n=0

图 15-32 主控设备 → 从属设备的通信例子
 (主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待) (2/4)

(2) 地址~数据~数据



注 1. 要在主控方的发送期间解除等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。

2. 要在从属方的接收期间解除等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

备注 n=0

图 15-32 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 15-32 的①~⑮是通过 I²C 总线进行数据通信的一系列运行步骤。

图 15-32 的“(1) 开始条件~地址~数据”说明步骤①~⑥。

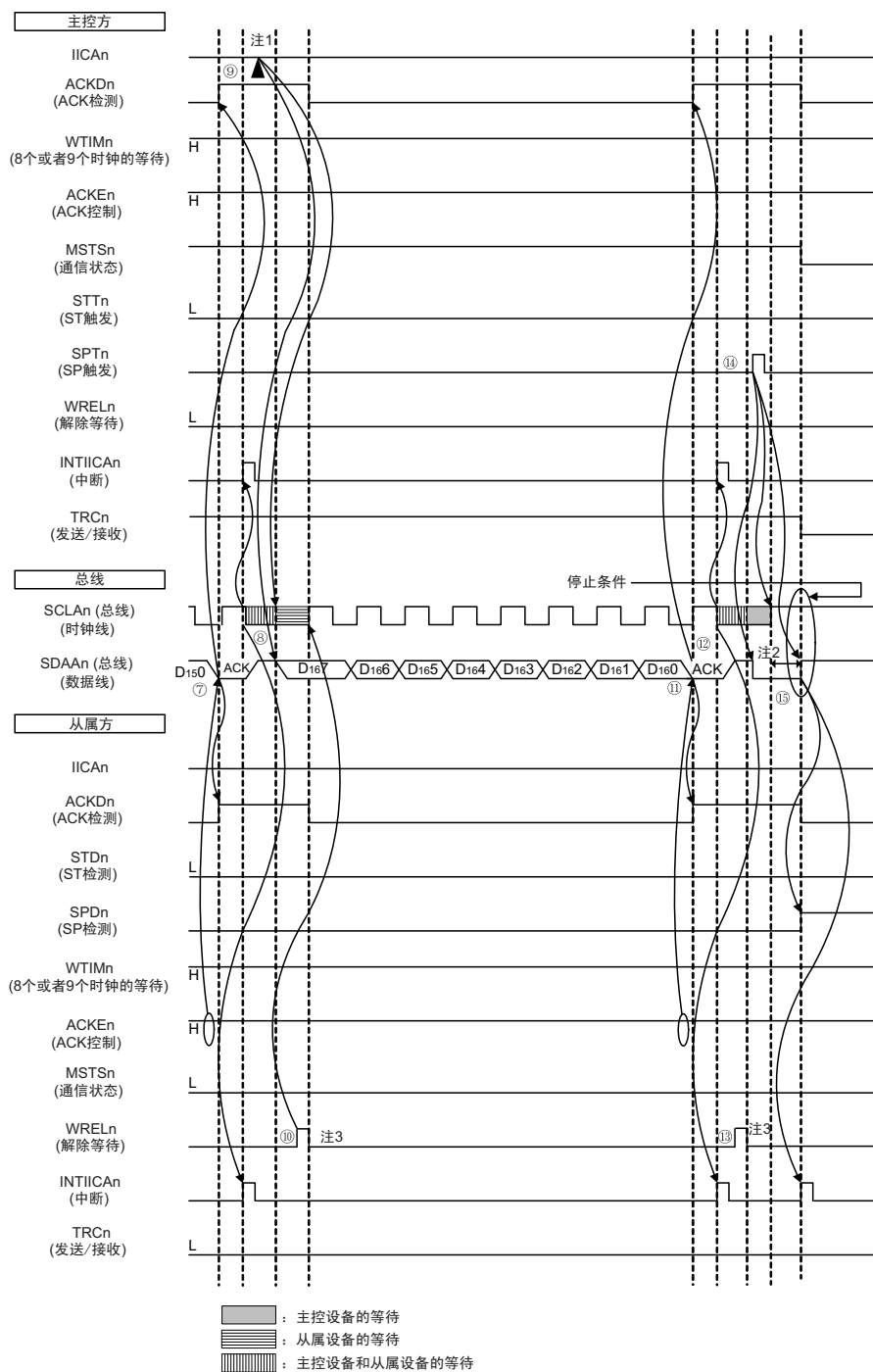
图 15-32 的“(2) 地址~数据~数据”说明步骤③~⑩。

图 15-32 的“(3) 数据~数据~停止条件”说明步骤⑦~⑮。

2. n=0

图 15-32 主控设备 → 从属设备的通信例子
 (主控设备: 选择 9 个时钟的等待, 从属设备: 选择 9 个时钟的等待) (3/4)

(3) 数据~数据~停止条件



- 注 1. 要在主控方的发送期间解除等待时, 必须给 IICAn 写数据而不是将 WRELn 位置位。
2. 在发行停止条件后, 从 SCLAn 引脚信号上升到生成停止条件的的时间, 在设定为标准模式时至少为 4.0μs, 在设定为快速模式时至少为 0.6μs。
3. 要在从属方的接收期间解除等待时, 必须将 IICAn 置 "FFH" 或者将 WRELn 位置位。

备注 n=0

图 15-32 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 15-32 的①～⑮是通过 I²C 总线进行数据通信的一系列运行步骤。

图 15-32 的“(1) 开始条件～地址～数据”说明步骤①～⑥。

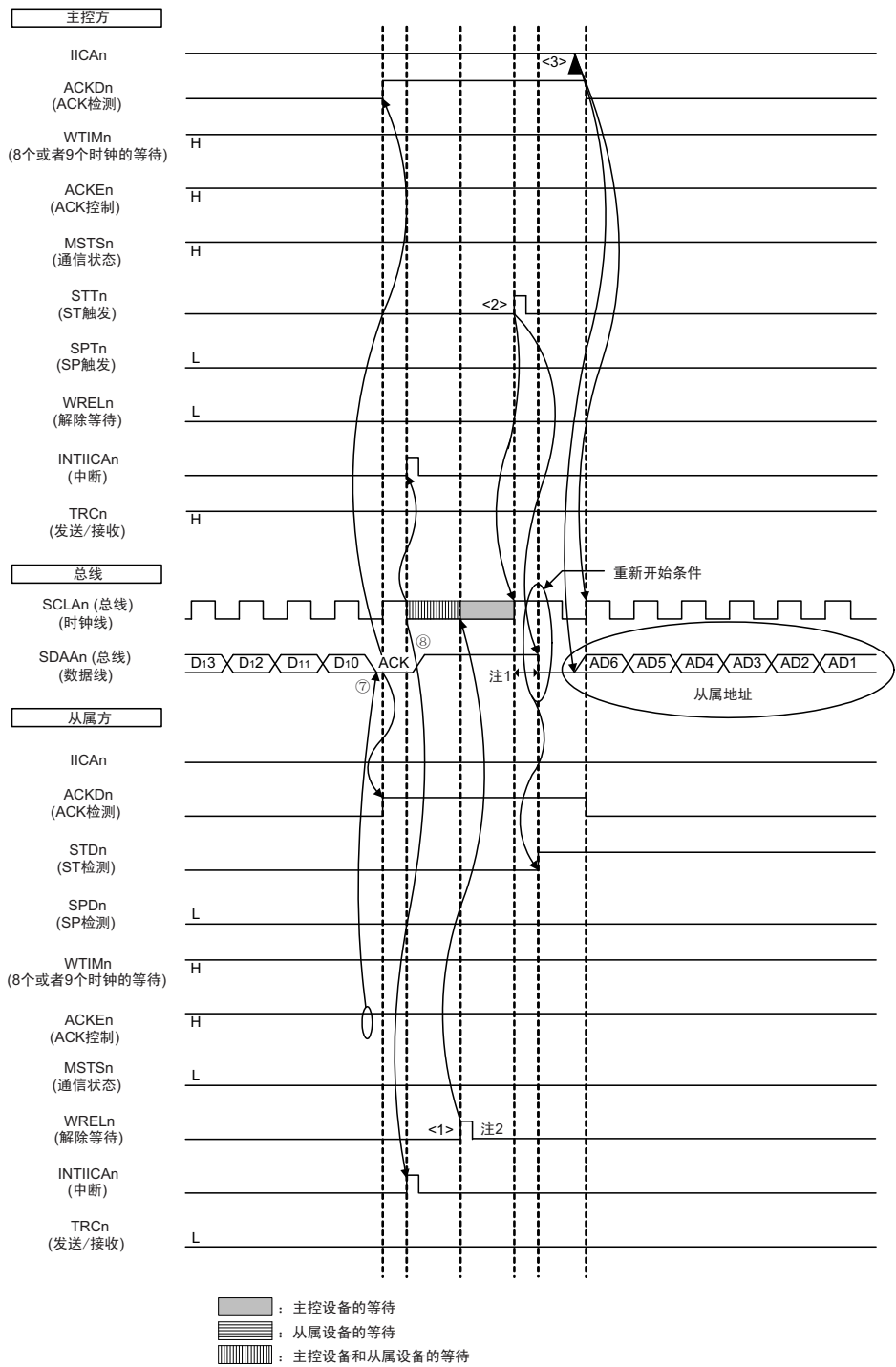
图 15-32 的“(2) 地址～数据～数据”说明步骤③～⑩。

图 15-32 的“(3) 数据～数据～停止条件”说明步骤⑦～⑮。

2. n=0

图 15-32 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时间，在设定为标准模式时至少为 4.7μs，在设定为快速模式时至少为 0.6μs。

2. 要在从属方的接收期间解除等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

备注 n=0

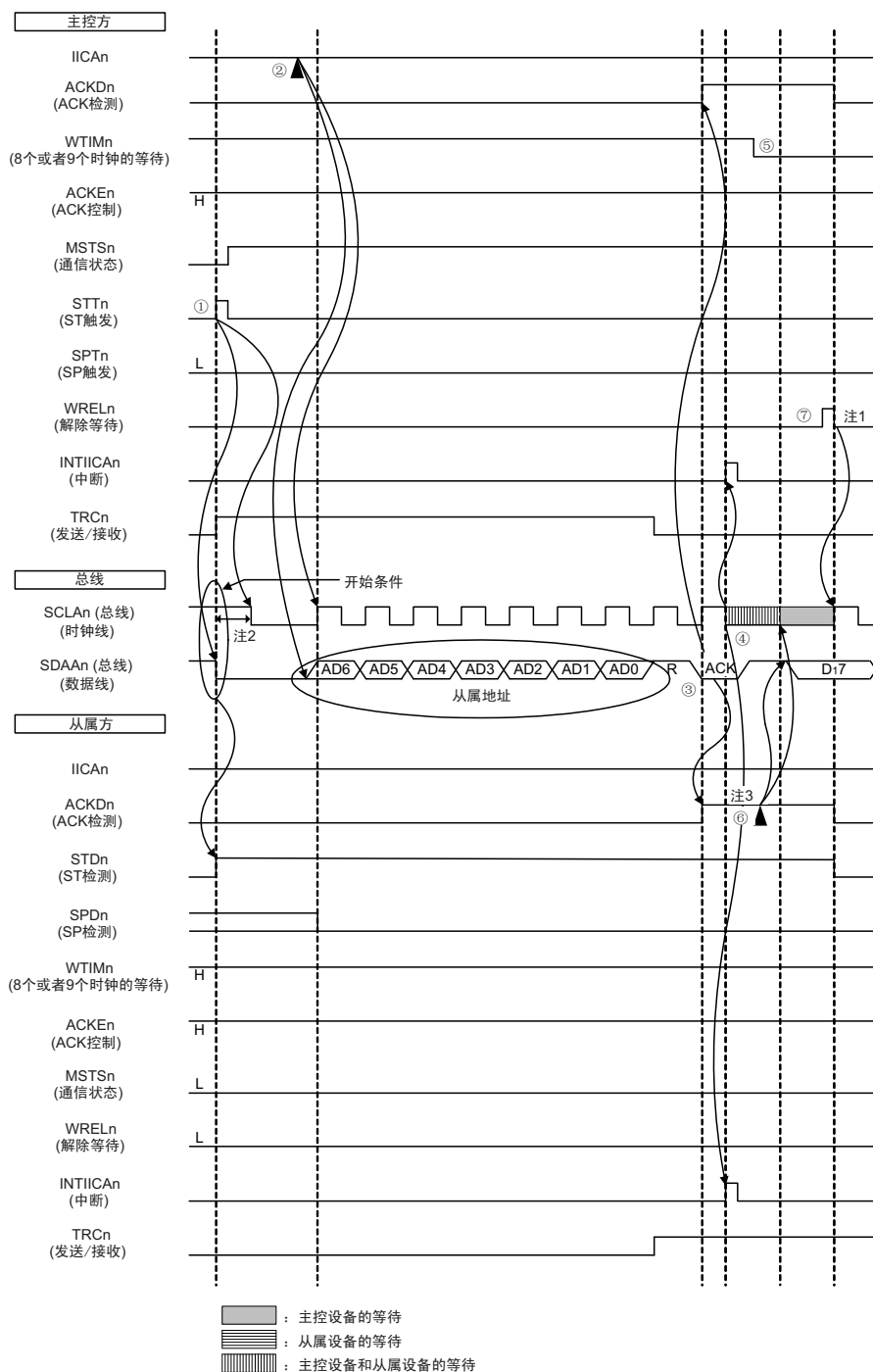
图 15-32 的“(4) 数据～重新开始条件～地址”的运行说明如下。在执行步骤⑦和⑧后执行<1>～<3>，从而返回到步骤③的数据发送步骤。

- 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 15-33 从属设备 → 主控设备的通信例子
(主控设备：选择 8 个时钟的等待，从属设备：选择 9 个时钟的等待) (1/3)

(1) 开始条件～地址～数据



- 注 1. 要在主控方的接收期间解除等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。
2. 从 SDAAn 引脚信号下降到 SCLAn 引脚信号下降的时间，在设定为标准模式时至少为 4.0μs，在设定为快速模式时至少为 0.6μs。
3. 要在从属方的发送期间解除等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。

备注 n=0

图 15-33 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 15-33 的①～⑱是通过 I²C 总线进行数据通信的一系列运行步骤。

图 15-33 的“(1) 开始条件～地址～数据”说明步骤①～⑦。

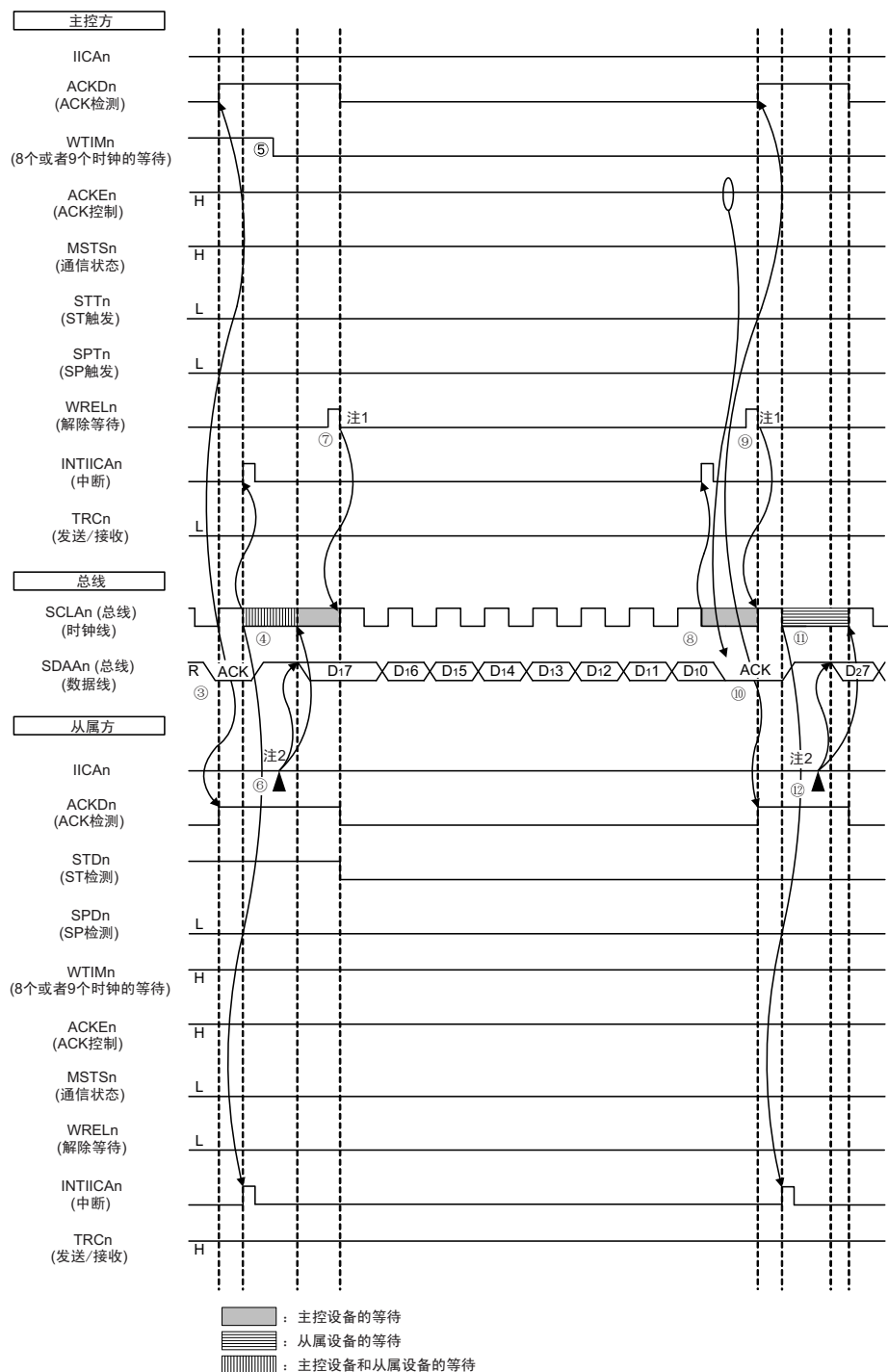
图 15-33 的“(2) 地址～数据～数据”说明步骤③～⑫。

图 15-33 的“(3) 数据～数据～停止条件”说明步骤⑧～⑱。

2. n=0

图 15-33 从属设备 → 主控设备的通信例子
(主控设备：选择 8 个时钟的等待，从属设备：选择 9 个时钟的等待) (2/3)

(2) 地址~数据~数据



- 注 1. 要在主控方的接收期间解除等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。
 2. 要在从属方的发送期间解除等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。

备注 n=0

图 15-33 的“(2) 地址~数据~数据”的③~⑫的说明如下:

- ③ 在从属方, 如果接收地址和本地站地址 (SVAn 的值) 相同注, 就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。
- ④ 主控方在第 9 个时钟的下降沿产生中断 (INTIICAn: 地址发送结束中断)。相同地址的从属设备进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 地址匹配中断) 注。
- ⑤ 主控方将等待时序改为第 8 个时钟 (WTIMn=0)。
- ⑥ 从属方给 IICA 移位寄存器 n (IICAn) 写发送数据, 解除从属方的等待。
- ⑦ 主控方解除等待 (WRELn=1), 开始来自从属设备的数据传送。
- ⑧ 主控方在第 8 个时钟的下降沿进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 传送结束中断)。因为主控方的 ACKEn 位为“1”, 所以通过硬件给从属方发送 ACK。
- ⑨ 主控方读接收数据, 解除等待 (WRELn=1)。
- ⑩ 从属方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。
- ⑪ 从属方在第 9 个时钟的下降沿进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 传送结束中断)。
- ⑫ 如果从属方给 IICAn 寄存器写发送数据, 就解除从属方的等待, 开始从属设备到主控设备的数据传送。

注 如果发送的地址和从属地址不同, 从属方就不给主控方返回 ACK (NACK: SDAAn=1), 并且不产生 INTIICAn 中断 (地址匹配中断), 也不进入等待状态。
但是, 主控方对于 ACK 和 NACK 都产生 INTIICAn 中断 (地址发送结束中断)。

备注 1. 图 15-33 的①~⑭是通过 I²C 总线进行数据通信的一系列运行步骤。

图 15-33 的“(1) 开始条件~地址~数据”说明步骤①~⑦。

图 15-33 的“(2) 地址~数据~数据”说明步骤③~⑫。

图 15-33 的“(3) 数据~数据~停止条件”说明步骤⑧~⑭。

2. n=0

图 15-33 的“(3) 数据～数据～停止条件”的⑧～⑲的说明如下：

- ⑧ 主控方在第 8 个时钟的下降沿进入等待状态（ $SCLAn=0$ ），并且产生中断（INTIICAn：传送结束中断）。因为主控方的 $ACKEn$ 位为“0”，所以通过硬件给从属方发送 ACK。
- ⑨ 主控方读接收数据，解除等待（ $WRELn=1$ ）。
- ⑩ 从属方在第 9 个时钟的上升沿检测到 ACK（ $ACKDn=1$ ）。
- ⑪ 从属方在第 9 个时钟的下降沿进入等待状态（ $SCLAn=0$ ），并且产生中断（INTIICAn：传送结束中断）。
- ⑫ 如果从属方给 IICA 移位寄存器 n（ $IICAn$ ）写发送数据，就解除从属方的等待，开始从属设备到主控设备的数据传送。
- ⑬ 主控方在第 8 个时钟的下降沿产生中断（INTIICAn：传送结束中断），并且进入等待状态（ $SCLAn=0$ ）。因为进行 ACK 控制（ $ACKEn=1$ ），所以此阶段的总线数据线变为低电平（ $SDAAn=0$ ）。
- ⑭ 主控方设定为 NACK 应答（ $ACKEn=0$ ），并且将等待时序改为第 9 个时钟（ $WTIMn=1$ ）。
- ⑮ 如果主控方解除等待（ $WRELn=1$ ），从属方就在第 9 个时钟的上升沿检测到 NACK（ $ACKDn=0$ ）。
- ⑯ 主控方和从属方在第 9 个时钟的下降沿进入等待状态（ $SCLAn=0$ ），并且都产生中断（INTIICAn：传送结束中断）。
- ⑰ 如果主控方发行停止条件（ $SPTn=1$ ），就清除总线数据线（ $SDAAn=0$ ），并且解除主控方的等待。此后，主控设备处于待机状态，直到将总线时钟线置位（ $SCLAn=1$ ）为止。
- ⑱ 从属方在确认 NACK 后停止发送，为了结束通信，解除等待（ $WRELn=1$ ）。如果解除从属方的等待，就将总线时钟线置位（ $SCLAn=1$ ）。
- ⑲ 如果主控方确认到总线时钟线被置位（ $SCLAn=1$ ），就在经过停止条件准备时间后将总线数据线置位（ $SDAAn=1$ ），然后发行停止条件（通过 $SCLAn=1$ 使 $SDAAn$ 从“0”变为“1”）。如果生成停止条件，从属方就检测到停止条件，并且产生中断（INTIICAn：停止条件中断）。

备注 1. 图 15-33 的①～⑲是通过 I²C 总线进行数据通信的一系列运行步骤。

图 15-33 的“(1) 开始条件～地址～数据”说明步骤①～⑦。

图 15-33 的“(2) 地址～数据～数据”说明步骤③～⑫。

图 15-33 的“(3) 数据～数据～停止条件”说明步骤⑧～⑲。

2. $n=0$

第 16 章 数据传送控制器（DTC）

16.1 DTC 的功能

数据传送控制器（DTC）是不使用 CPU 而在存储器之间进行数据传送的功能。通过外围功能中断启动 DTC 进行数据传送。DTC 和 CPU 使用相同的数据总线，其总线使用权高于 CPU。

DTC 的规格如表 16-1 所示。

表 16-1 DTC 的规格

项目		规格
启动源		29 个源
可分配的控制数据		24 组
可传送的地址空间	地址空间	64K 字节空间（F0000H ~ FFFFFH），但是通用寄存器除外。
	源	特殊功能寄存器（SFR）、RAM 区（通用寄存器除外）、镜像区注、数据闪存区注、扩展特殊功能寄存器（2nd SFR）
	目标	特殊功能寄存器（SFR）、RAM 区（通用寄存器除外）、扩展特殊功能寄存器（2nd SFR）
最大传送次数	正常模式	256 次
	重复模式	255 次
最大传送块大小	正常模式（8 位传送）	256 字节
	正常模式（16 位传送）	512 字节
	重复模式	255 字节
传送单位		8 位 /16 位
传送模式	正常模式	在进行 DTCCTj 寄存器从“1”变为“0”的传送后结束。
	重复模式	在 DTCCTj 寄存器从“1”变为“0”的传送结束后，对重复区的地址进行初始化，在将 DTRLDj 寄存器的值重新加载到 DTCCTj 寄存器后继续传送。
地址控制	正常模式	固定或者递增
	重复模式	固定或者递增非重复区的地址。
启动源的优先级		参照“表 16-4 DTC 启动源和向量地址”。
中断请求	正常模式	在进行 DTCCTj 寄存器从“1”变为“0”的数据传送时，向 CPU 请求启动源的中断，并且在数据传送结束后进行中断处理。
	重复模式	在 DTCCRj 寄存器的 RPTINT 位为“1”（允许产生中断）并且进行 DTCCTj 寄存器从“1”变为“0”的数据传送时，向 CPU 请求启动源的中断，并且在数据传送结束后进行中断处理。
传送开始		如果将 DTCENi 寄存器的 DTCENi0 ~ DTCENi7 位置“1”（允许启动），就在每次发生 DTC 启动源时开始数据传送。
传送停止	正常模式	- 将 DTCENi0 ~ DTCENi7 位置“0”（禁止启动）。 - 当 DTCCTj 寄存器从“1”变为“0”的数据传送结束时
	重复模式	- 将 DTCENi0 ~ DTCENi7 位置“0”（禁止启动）。 - 当 RPTINT 位为“1”（允许产生中断）并且 DTCCTj 寄存器从“1”变为“0”的数据传送结束时

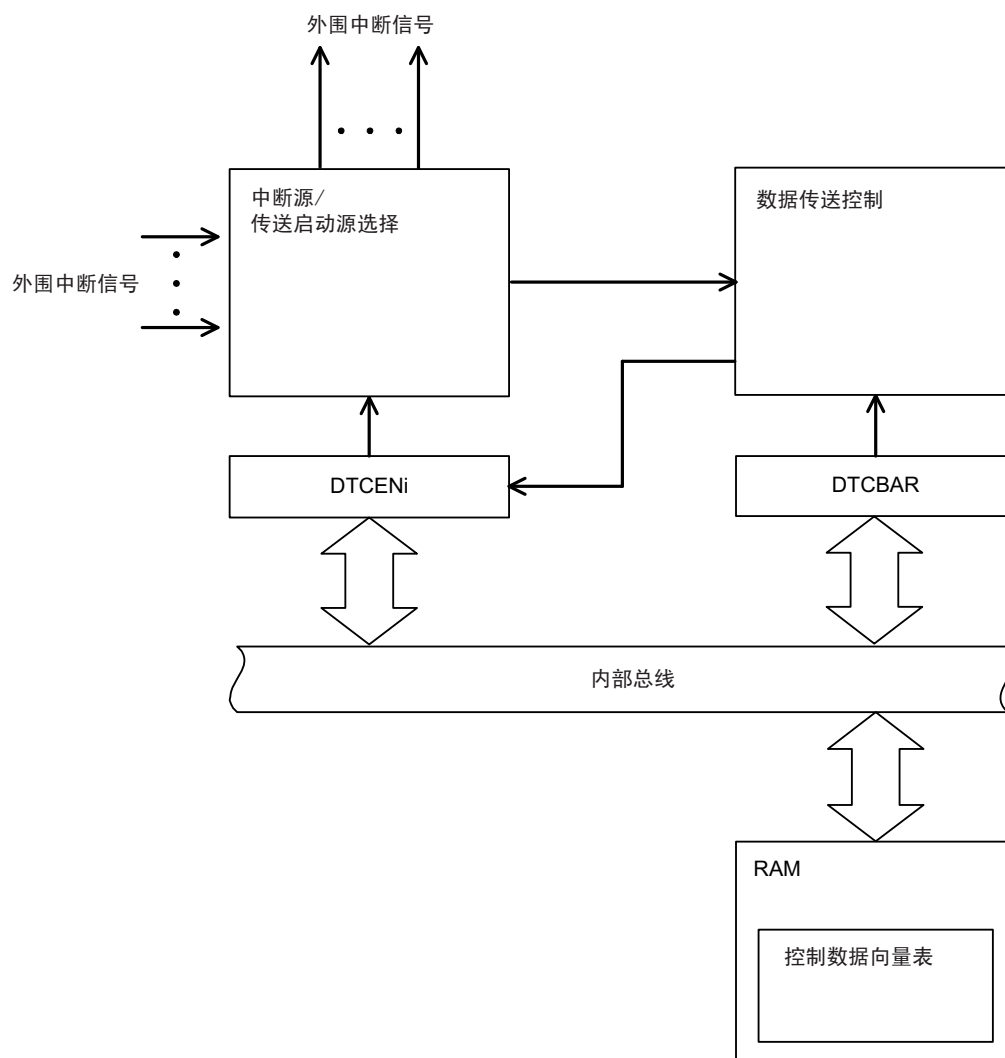
注 在 SNOOZE 模式中，因为闪存停止运行，所以不能作为 DTC 传送源。

备注 i=0 ~ 4、j=0 ~ 23

16.2 DTC 的结构

DTC 的框图如图 16-1 所示。

图 16-1 DTC 的框图



16.3 控制 DTC 的寄存器

控制 DTC 的寄存器如表 16-2 所示。

表 16-2 控制 DTC 的寄存器

寄存器名	符号
外围允许寄存器 1	PER1
DTC 启动允许寄存器 0	DTCEN0
DTC 启动允许寄存器 1	DTCEN1
DTC 启动允许寄存器 2	DTCEN2
DTC 启动允许寄存器 3	DTCEN3
DTC 启动允许寄存器 4	DTCEN4
DTC 基址寄存器	DTCBAR

DTC 的控制数据如表 16-3 所示。

DTC 的控制数据分配在 RAM 的 DTC 控制数据区。

通过 DTCBAR 寄存器设定 DTC 控制数据区和包含 DTC 向量表区（保存控制数据的起始地址）的 256 字节区域。

表 16-3 DTC 的控制数据

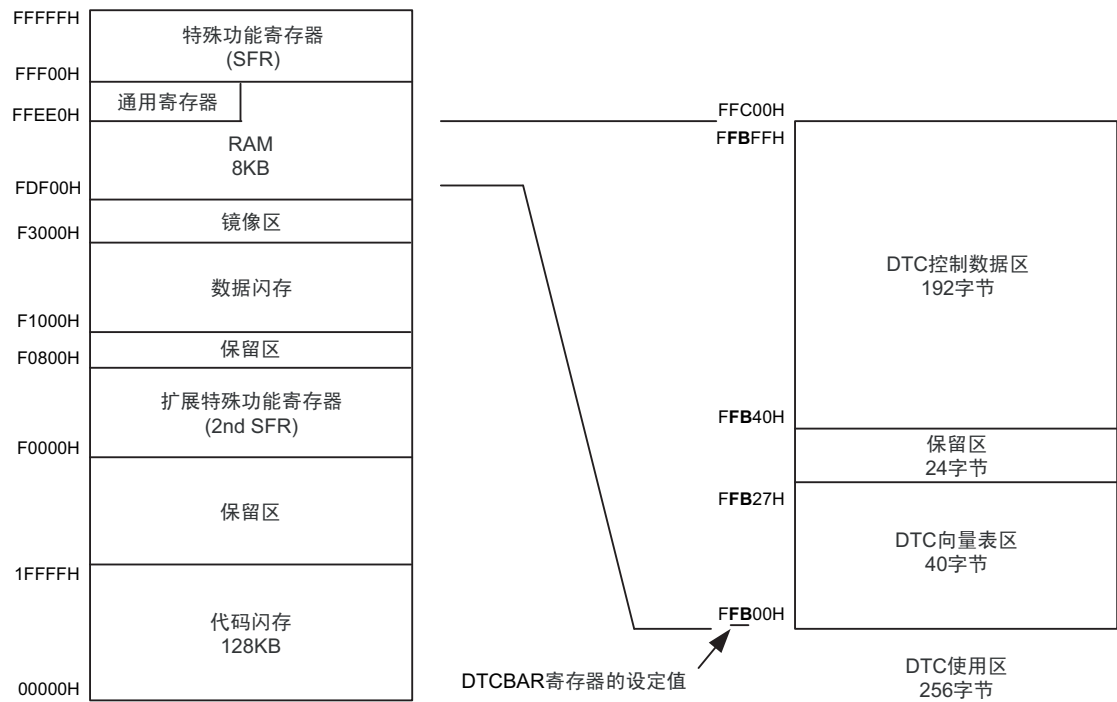
寄存器名	符号
DTC 控制寄存器 j	DTCCRj
DTC 块大小寄存器 j	DTBLSj
DTC 传送次数寄存器 j	DTCCTj
DTC 传送次数重加载寄存器 j	DTRLdj
DTC 源地址寄存器 j	DTSARj
DTC 目标地址寄存器 j	DTDARj

备注 j=0 ~ 23

16.3.1 DTC 控制数据区和 DTC 向量表区的分配

通过 DTCBAR 寄存器将分配 DTC 的控制数据和向量表的 256 字节区域设定到 RAM 区。
DTCBAR 寄存器的设定值为“FBH”时的存储器映像例子如图 16-2 所示。
DTC 控制数据区的 192 字节中 DTC 不使用的空间能用作 RAM。

图 16-2 DTCBAR 寄存器的设定值为“FBH”时的存储器映像例子



能分配 DTC 控制数据和向量表的区域因产品而不同。

- 注意 1. 禁止将通用寄存器（FFEE0H ~ FFEFFH）的空间用作 DTC 控制数据区和 DTC 向量表区。
2. 堆栈区、DTC 控制数据区和 DTC 向量表区不能重叠。
3. 在使用自编程功能和数据闪存功能时，不能将以下产品的内部 RAM 区用作 DTC 控制数据区和 DTC 向量表区。
- R7F0C014B2D、R7F0C014L2D: FDF00H ~ FE309H

16.3.2 控制数据的分配

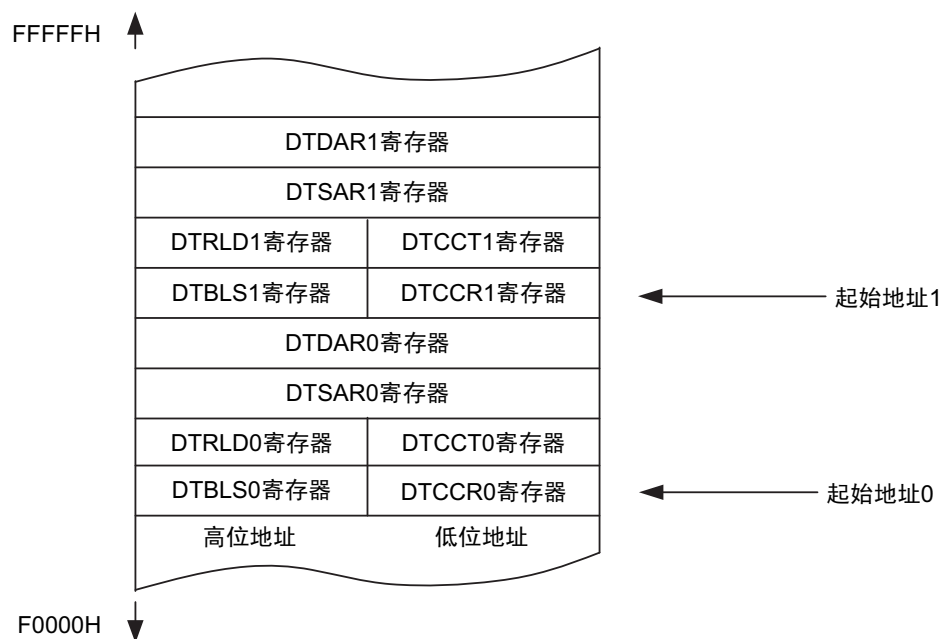
从起始地址开始，按照 DTCCRj、DTBLSj、DTCCTj、DTRL Dj、DTSARj、DTDARj（j=0 ~ 23）寄存器的顺序分配控制数据。

起始地址 0 ~ 23 的高 8 位由 DTCBAR 寄存器设定，低 8 位由各启动源分配的向量表分别设定。

控制数据的分配如图 16-3 所示。

- 注意 1. 必须在对应的 DTCENi（i=0 ~ 4）的 DTCENi0 ~ DTCENi7 位为“0”（禁止启动）时更改 DTCCRj、DTBLSj、DTCCTj、DTRL Dj、DTSARj、DTDARj 寄存器的数据。
2. 不能通过 DTC 传送进行 DTCCRj、DTBLSj、DTCCTj、DTRL Dj、DTSARj 和 DTDARj 的存取。

图 16-3 控制数据的分配



16.3.3 向量表

DTC 一旦启动，就通过从各启动源分配的向量表读取的数据来决定控制数据，读被分配在 DTC 控制数据区的控制数据。

DTC 启动源和向量地址如表 16-4 所示。各启动源的向量表有 1 字节，保存“40H”~“F8H”的数据，从 24 组的控制数据中选择 1 组数据。向量地址的高 8 位由 DTCBAR 寄存器设定，低 8 位被分配了对应启动源的“00H”~“27H”。

注意 必须在对应的 DTCENi（i=0 ~ 4）寄存器的 DTCENi0 ~ DTCENi7 位为“0”（禁止启动）时更改设定在向量表中的 DTC 控制数据区的起始地址。

表 16-4 DTC 启动源和向量地址

DTC 启动源 (中断请求发生源)	源号	向量地址	优先级
保留	0	DTCBAR 寄存器的设定地址 +00H	高 ↑ ↓ 低
INTP0	1	DTCBAR 寄存器的设定地址 +01H	
INTP1	2	DTCBAR 寄存器的设定地址 +02H	
INTP2	3	DTCBAR 寄存器的设定地址 +03H	
INTP3	4	DTCBAR 寄存器的设定地址 +04H	
INTP4	5	DTCBAR 寄存器的设定地址 +05H	
INTP5	6	DTCBAR 寄存器的设定地址 +06H	
INTP6 注	7	DTCBAR 寄存器的设定地址 +07H	
INTP7 注	8	DTCBAR 寄存器的设定地址 +08H	
键输入注	9	DTCBAR 寄存器的设定地址 +09H	
A/D 转换结束	10	DTCBAR 寄存器的设定地址 +0AH	
UART0 接收的传送结束 /CSI01 的传送结束或者 缓冲器空 /IIC01 的传送结束	11	DTCBAR 寄存器的设定地址 +0BH	
UART0 发送的传送结束 /CSI00 的传送结束或者 缓冲器空 /IIC00 的传送结束	12	DTCBAR 寄存器的设定地址 +0CH	
UART1 接收的传送结束 /CSI11 的传送结束或者 缓冲器空 /IIC11 的传送结束	13	DTCBAR 寄存器的设定地址 +0DH	
UART1 发送的传送结束 /CSI10 的传送结束或者 缓冲器空 /IIC10 的传送结束	14	DTCBAR 寄存器的设定地址 +0EH	
UART2 接收的传送结束 /CSI21 的传送结束或者 缓冲器空 /IIC21 的传送结束	15	DTCBAR 寄存器的设定地址 +0FH	
UART2 发送的传送结束 /CSI20 的传送结束或者 缓冲器空 /IIC20 的传送结束	16	DTCBAR 寄存器的设定地址 +10H	
保留	17	DTCBAR 寄存器的设定地址 +11H	
保留	18	DTCBAR 寄存器的设定地址 +12H	
定时器阵列单元 0 的通道 0 的计数或捕捉结束	19	DTCBAR 寄存器的设定地址 +13H	
定时器阵列单元 0 的通道 1 的计数或捕捉结束	20	DTCBAR 寄存器的设定地址 +14H	
定时器阵列单元 0 的通道 2 的计数或捕捉结束	21	DTCBAR 寄存器的设定地址 +15H	
定时器阵列单元 0 的通道 3 的计数或捕捉结束	22	DTCBAR 寄存器的设定地址 +16H	
保留	23	DTCBAR 寄存器的设定地址 +17H	
保留	24	DTCBAR 寄存器的设定地址 +18H	
保留	25	DTCBAR 寄存器的设定地址 +19H	
保留	26	DTCBAR 寄存器的设定地址 +1AH	
定时器 RD 的比较匹配 A0	27	DTCBAR 寄存器的设定地址 +1BH	
定时器 RD 的比较匹配 B0	28	DTCBAR 寄存器的设定地址 +1CH	
定时器 RD 的比较匹配 C0	29	DTCBAR 寄存器的设定地址 +1DH	
定时器 RD 的比较匹配 D0	30	DTCBAR 寄存器的设定地址 +1EH	
定时器 RD 的比较匹配 A1	31	DTCBAR 寄存器的设定地址 +1FH	
定时器 RD 的比较匹配 B1	32	DTCBAR 寄存器的设定地址 +20H	
定时器 RD 的比较匹配 C1	33	DTCBAR 寄存器的设定地址 +21H	
定时器 RD 的比较匹配 D1	34	DTCBAR 寄存器的设定地址 +22H	
保留	35	DTCBAR 寄存器的设定地址 +23H	
保留	36	DTCBAR 寄存器的设定地址 +24H	
定时器 RJ0 的下溢	37	DTCBAR 寄存器的设定地址 +25H	
保留	38	DTCBAR 寄存器的设定地址 +26H	
保留	39	DTCBAR 寄存器的设定地址 +27H	

注 只限于 64 引脚产品。

16.3.4 外围允许寄存器 1（PER1）

PER1 寄存器是设定允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

- 要使用 DTC 时，必须将 bit3（DTCEN）置“1”。
- 通过 1 位或者 8 位存储器操作指令设定 PER1 寄存器。
- 在产生复位信号后，此寄存器的值变为“00H”。

图 16-4 外围允许寄存器 1（PER1）的格式

地址：F007AH	复位后：00H		R/W					
符号	7	6	5	4	3	2	1	0
PER1	0	0	0	TRD0EN	DTCEN	0	0	TRJ0EN

DTCEN	提供 DTC 的输入时钟的控制
0	停止提供输入时钟。 • DTC 不能运行。
1	提供输入时钟。 • DTC 能运行。

注意 必须将以下的位置“0”。

bit1、2、5～7

16.3.5 DTC 控制寄存器 j（DTCCRj）（j=0 ~ 23）

DTCCRj 寄存器控制 DTC 的运行模式。

图 16-5 DTC 控制寄存器 j（DTCCRj）的格式

地址：参照“16.3.2 控制数据的分配”。

复位后：不定值

R/W

符号	7	6	5	4	3	2	1	0
DTCCRj	0	SZ	RPTINT	CHNE	DAMOD	SAMOD	RPTSEL	MODE

SZ	传送数据长度的选择
0	8 位
1	16 位

RPTINT	重复模式中中断的允许 / 禁止
0	禁止发生中断。
1	允许发生中断。
在 MODE 位为“0”（正常模式）时，RPTINT 位的设定无效。	

CHNE	链传送的允许 / 禁止
0	禁止链传送。
1	允许链传送。
必须将 DTCCR23 寄存器的 CHNE 位置“0”（禁止链传送）。	

DAMOD	传送目标地址的控制
0	固定
1	递增
在 MODE 位为“1”（重复模式）并且 RPTSEL 位为“0”（传送目标为重复区）时，DAMOD 位的设定无效。	

SAMOD	传送源地址的控制
0	固定
1	递增
在 MODE 位为“1”（重复模式）并且 RPTSEL 位为“1”（传送源为重复区）时，SAMOD 位的设定无效。	

RPTSEL	重复区的选择
0	传送目标为重复区。
1	传送源为重复区。
在 MODE 位为“0”（正常模式）时，RPTSEL 位的设定无效。	

MODE	传送模式的选择
0	正常模式
1	重复模式

注意 不能通过 DTC 传送进行 DTCCRj 寄存器的存取。

16.3.6 DTC 块大小寄存器 j (DTBLSj) (j=0 ~ 23)

此寄存器设定 1 次启动传送数据的块大小。

图 16-6 DTC 块大小寄存器 j (DTBLSj) 的格式

地址：参照“16.3.2 控制数据的分配”。

复位后：不定值

R/W

符号	7	6	5	4	3	2	1	0
DTBLSj	DTBLSj7	DTBLSj6	DTBLSj5	DTBLSj4	DTBLSj3	DTBLSj2	DTBLSj1	DTBLSj0

DTBLSj	传送块大小	
	8 位传送	16 位传送
00H	256 字节	512 字节
01H	1 字节	2 字节
02H	2 字节	4 字节
03H	3 字节	6 字节
⋮	⋮	⋮
FDH	253 字节	506 字节
FEH	254 字节	508 字节
FFH	255 字节	510 字节

注意 不能通过 DTC 传送进行 DTBLSj 寄存器的存取。

16.3.7 DTC 传送次数寄存器 j (DTCCTj) (j=0 ~ 23)

此寄存器设定 DTC 的数据传送次数。每当启动 1 次 DTC 传送就减 1。

图 16-7 DTC 传送次数寄存器 j (DTCCTj) 的格式

地址：参照“16.3.2 控制数据的分配”。

复位后：不定值

R/W

符号	7	6	5	4	3	2	1	0
DTCCTj	DTCCTj7	DTCCTj6	DTCCTj5	DTCCTj4	DTCCTj3	DTCCTj2	DTCCTj1	DTCCTj0

DTCCTj	传送次数
00H	256 次
01H	1 次
02H	2 次
03H	3 次
⋮	⋮
FDH	253 次
FEH	254 次
FFH	255 次

注意 不能通过 DTC 传送进行 DTCCTj 寄存器的存取。

16.3.8 DTC 传送次数重加载寄存器 j (DTRLDj) (j=0 ~ 23)

此寄存器设定重复模式中的传送次数寄存器的初始值。在重复模式中，因为将此寄存器的值重新加载到 DTCCT 寄存器，所以设定值必须和 DTCCT 寄存器的初始值相同。

图 16-8 DTC 传送次数重加载寄存器 j (DTRLDj) 的格式

地址: 参照“16.3.2 控制数据的分配”。	复位后: 不定值				R/W			
符号	7	6	5	4	3	2	1	0
DTRLDj	DTRLDj7	DTRLDj6	DTRLDj5	DTRLDj4	DTRLDj3	DTRLDj2	DTRLDj1	DTRLDj0

注意 不能通过 DTC 传送进行 DTRLDj 寄存器的存取。

16.3.9 DTC 源地址寄存器 j (DTSARj) (j=0 ~ 23)

此寄存器指定数据传送时的传送源地址。

当 DTCCRj 寄存器的 SZ 位为“1”（16 位传送）时，忽视最低位而作为偶地址进行处理。

图 16-9 DTC 源地址寄存器 j (DTSARj) 的格式

地址：参照“16.3.2 控制数据的分配”。						复位后：不定值				R/W						
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTSARj	DTS	DTS	DTS	DTS	DTS	DTS	DTS	DTS	DTS	DTS	DTS	DTS	DTS	DTS	DTS	DTS
	ARj15	ARj14	ARj13	ARj12	ARj11	ARj10	ARj9	ARj8	ARj7	ARj6	ARj5	ARj4	ARj3	ARj2	ARj1	ARj0

注意 1. 不能将通用寄存器 (FFEE0H ~ FFEFFH) 的空间设定为传送源地址。

2. 不能通过 DTC 传送进行 DTSARj 寄存器的存取。

16.3.10 DTC 目标地址寄存器 j (DTDARj) (j=0 ~ 23)

此寄存器指定数据传送时的传送目标地址。

当 DTCCRj 寄存器的 SZ 位为“1”（16 位传送）时，忽视最低位而作为偶地址进行处理。

图 16-10 DTC 目标地址寄存器 j (DTDARj) 的格式

地址：参照“16.3.2 控制数据的分配”。						复位后：不定值			R/W							
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTDARj	DTD	DTD	DTD	DTD	DTD	DTD	DTD	DTD	DTD	DTD	DTD	DTD	DTD	DTD	DTD	DTD
	ARj15	ARj14	ARj13	ARj12	ARj11	ARj10	ARj9	ARj8	ARj7	ARj6	ARj5	ARj4	ARj3	ARj2	ARj1	ARj0

注意 1. 不能将通用寄存器 (FFEE0H ~ FFEFFH) 的空间设定为传送目标地址。

2. 不能通过 DTC 传送进行 DTDARj 寄存器的存取。

16.3.11 DTC 启动允许寄存器 i（DTCENi）（i=0 ~ 4）

这是控制允许或者禁止通过各中断源启动 DTC 的 8 位寄存器。中断源和 DTCENi0 ~ DTCENi7 位的对应如表 16-5 所示。

能通过 8 位或者 1 位存储器操作指令设定 DTCENi 寄存器。

- 注意 1. 必须在不产生对应该位的启动源的位置更改 DTCENi0 ~ DTCENi7 位。
2. 不能通过 DTC 传送进行 DTCENi 寄存器的存取。
3. 分配的功能因产品而不同，必须将没有分配功能的位置“0”。

图 16-11 DTC 启动允许寄存器 i（DTCENi）（i=0 ~ 4）的格式

地址：F02E8H（DTCEN0）、F02E9H（DTCEN1）、F02EAH（DTCEN2）复位后：00HR/W

F02EBH（DTCEN3）、F02ECH（DTCEN4）

符号	7	6	5	4	3	2	1	0
DTCENi	DTCENi7	DTCENi6	DTCENi5	DTCENi4	DTCENi3	DTCENi2	DTCENi1	DTCENi0

DTCENi7	DTC 启动的允许 i7
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件， DTCENi7 位变为“0”（禁止启动）。	

DTCENi6	DTC 启动的允许 i6
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件， DTCENi6 位变为“0”（禁止启动）。	

DTCENi5	DTC 启动的允许 i5
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件， DTCENi5 位变为“0”（禁止启动）。	

DTCENi4	DTC 启动的允许 i4
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件， DTCENi4 位变为“0”（禁止启动）。	

DTCENi3	DTC 启动的允许 i3
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件， DTCENi3 位变为“0”（禁止启动）。	

DTCENi2	DTC 启动的允许 i2
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DTCENi2 位变为“0”（禁止启动）。	

DTCENi1	DTC 启动的允许 i1
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DTCENi1 位变为“0”（禁止启动）。	

DTCENi0	DTC 启动的允许 i0
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DTCENi0 位变为“0”（禁止启动）。	

表 16-5 中断源和 DTCENi0 ~ DTCENi7 位的对应

寄存器	DTCENi7 位	DTCENi6 位	DTCENi5 位	DTCENi4 位	DTCENi3 位	DTCENi2 位	DTCENi1 位	DTCENi0 位
DTCEN0	保留	INTP0	INTP1	INTP2	INTP3	INTP4	INTP5	INTP6
DTCEN1	INTP7	键输入	A/D 转换结束	UART0 接收的传送结束 / CSI01 的传送结束或者缓冲器空 / IIC01 的传送结束	UART0 发送的传送结束 / CSI00 的传送结束或者缓冲器空 / IIC00 的传送结束	UART1 接收的传送结束 / CSI11 的传送结束或者缓冲器空 / IIC11 的传送结束	UART1 发送的传送结束 / CSI10 的传送结束或者缓冲器空 / IIC10 的传送结束	UART2 接收的传送结束 / CSI21 的传送结束或者缓冲器空 / IIC21 的传送结束
DTCEN2	UART2 发送的传送结束 / CSI20 的传送结束或者缓冲器空 / IIC20 的传送结束	保留	保留	定时器阵列单元 0 的通道 0 的计数结束或者捕捉结束	定时器阵列单元 0 的通道 1 的计数结束或者捕捉结束	定时器阵列单元 0 的通道 2 的计数结束或者捕捉结束	定时器阵列单元 0 的通道 3 的计数结束或者捕捉结束	保留
DTCEN3	保留	保留	保留	定时器 RD 的比较匹配 A0	定时器 RD 的比较匹配 B0	定时器 RD 的比较匹配 C0	定时器 RD 的比较匹配 D0	定时器 RD 的比较匹配 A1
DTCEN4	定时器 RD 的比较匹配 B1	定时器 RD 的比较匹配 C1	定时器 RD 的比较匹配 D1	保留	保留	定时器 RJ0 的下溢	保留	保留

注意 必须将没有分配功能的位置“0”。

备注 i=0 ~ 4

16.3.12 DTC 基址寄存器 (DTCBAR)

这是 8 位寄存器，设定保存 DTC 控制数据区起始地址的向量地址以及 DTC 控制数据区的地址。生成以 DTCBAR 寄存器的值为高 8 位的 16 位地址。

注意 1. 必须在将全部的 DTC 启动源设定为禁止启动的状态下更改 DTCBAR 寄存器。

2. 只能改写 1 次 DTCBAR 寄存器。

3. 不能通过 DTC 传送进行 DTCBAR 寄存器的存取。

4. 有关 DTC 控制数据区和 DTC 向量表区的分配，请参照“16.3.1 DTC 控制数据区和 DTC 向量表区的分配”的注意事项。

图 16-12 DTC 基址寄存器 (DTCBAR) 的格式

地址: F02E0H	复位后: FDH	R/W						
符号	7	6	5	4	3	2	1	0
DTCBAR	DTCBAR7	DTCBAR6	DTCBAR5	DTCBAR4	DTCBAR3	DTCBAR2	DTCBAR1	DTCBAR0

16.4 DTC 的运行

DTC 一旦启动, 就从 DTC 控制数据区读控制数据, 根据此控制数据进行数据传送, 并且将数据传送后的控制数据回写到 DTC 控制数据区。能将 24 组控制数据保存到 DTC 控制数据区, 并且进行 24 组数据的传送。

传送模式有正常模式和重复模式, 传送大小有 8 位传送和 16 位传送。在 DTCCRj (j=0 ~ 23) 寄存器的 CHNE 位为“1” (允许链传送) 时, 通过 1 个启动源读多个控制数据进行连续的数据传送 (链传送)。

通过 16 位 DTSARj 寄存器和 16 位 DTDARj 寄存器分别指定传送源地址和传送目标地址。在数据传送后, 根据控制数据递增或者固定 DTSARj 寄存器和 DTDARj 寄存器的值。

16.4.1 启动源

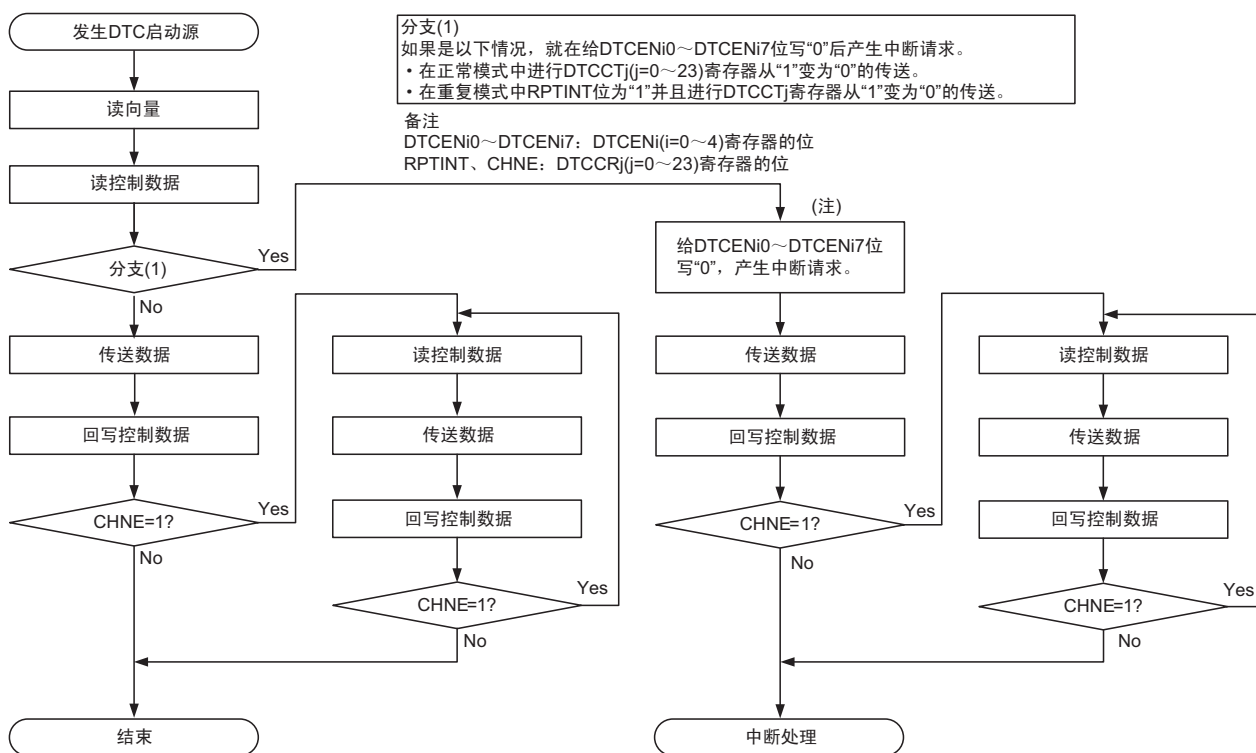
通过外围功能的中断信号启动 DTC, 并且通过 DTCENi 寄存器 (i=0 ~ 4) 选择启动 DTC 的中断信号。

当数据传送 (在进行链传送时, 连续进行最初的传送) 的设定为下述两种情况时, 就在 DTC 运行中将对应的 DTCENi 寄存器的 DTCENi0 ~ DTCENi7 位置“0” (禁止启动)。

- 在正常模式中, 进行 DTCCTj (j=0 ~ 23) 寄存器变为“0”的传送。
- 在重复模式中, DTCCRj 寄存器的 RPTINT 位为“1” (允许发生中断) 并且进行 DTCCTj 寄存器变为“0”的传送。

DTC 的内部运行流程图如图 16-13 所示。

图 16-13 DTC 的内部运行流程图



注 在通过允许链传送(CHNE位为“1”)的设定启动的数据传送中, 不给DTCENi0~DTCENi7位写“0”并且不产生请求中断请求。

16.4.2 正常模式

在 8 位传送时，1 次启动的传送数据为 1 ～ 256 字节；在 16 位传送时，1 次启动的传送数据为 2 ～ 512 字节。传送次数为 1 ～ 256 次。如果进行 DTCCTj（j=0 ～ 23）寄存器变为“0”的数据传送，就在 DTC 运行中向中断控制器产生对应启动源的中断请求，并且将对应的 DTCENi（i=0 ～ 4）寄存器的 DTCENi0 ～ DTCENi7 位置“0”（禁止启动）。

正常模式的寄存器功能和数据传送分别如表 16-6 和图 16-14 所示。

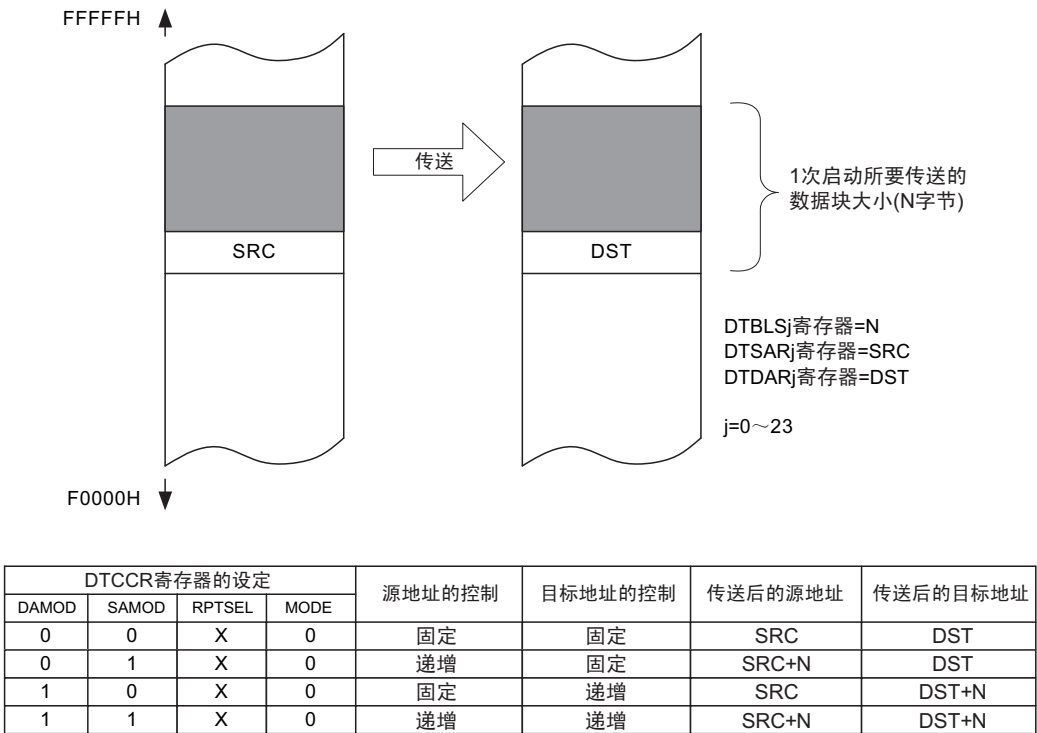
表 16-6 正常模式的寄存器功能

寄存器名	符号	功能
DTC 块大小寄存器 j	DTBLSj	1 次启动所要传送的数据块大小
DTC 传送次数寄存器 j	DTCCTj	数据的传送次数
DTC 传送次数重加载寄存器 j	DTRLDj	不使用注。
DTC 源地址寄存器 j	DTSARj	数据的传送源地址
DTC 目标地址寄存器 j	DTDARj	数据的传送目标地址

注 当通过 RAM 奇偶校验错误检测功能允许产生奇偶校验错误复位（RPERDIS=0）时，必须进行初始化（00H）。

备注 j=0 ～ 23

图 16-14 正常模式的数据传送



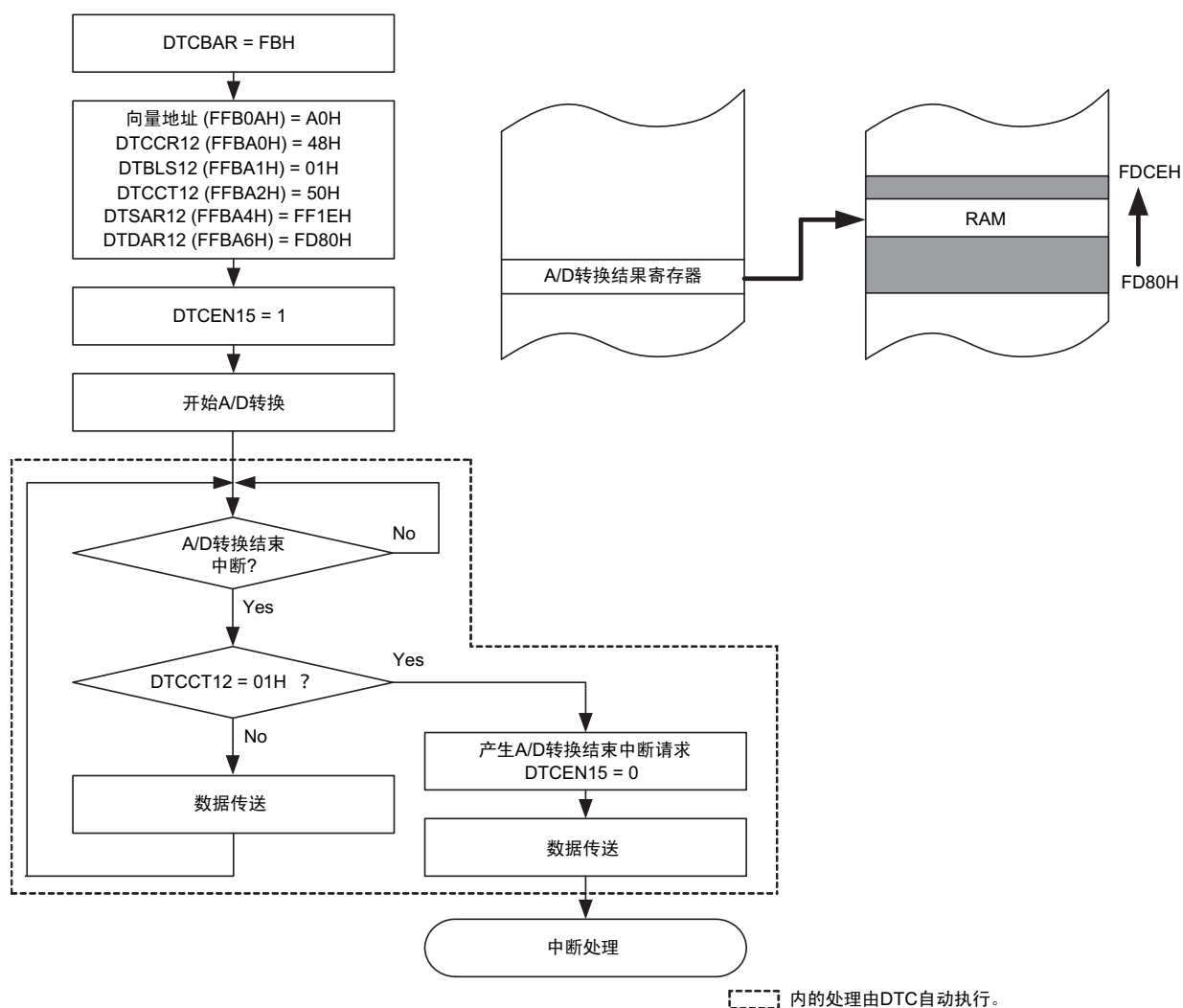
X: “0”或者“1”

(1) 正常模式的使用例子 1: 连续取 A/D 转换结果

通过 A/D 转换结束中断启动 DTC，并且将 A/D 转换结果寄存器的值传送到 RAM。

- 向量地址分配在 FFB0AH，控制数据分配在 FFBA0H ~ FFBA7H。
- 将 A/D 转换结果寄存器 (FFF1EH、FFF1FH) 的 2 字节数据传送到 RAM 的 FFD80H ~ FFDCEH 的 80 字节。

图 16-15 正常模式的使用例子 1: 连续取 A/D 转换结果



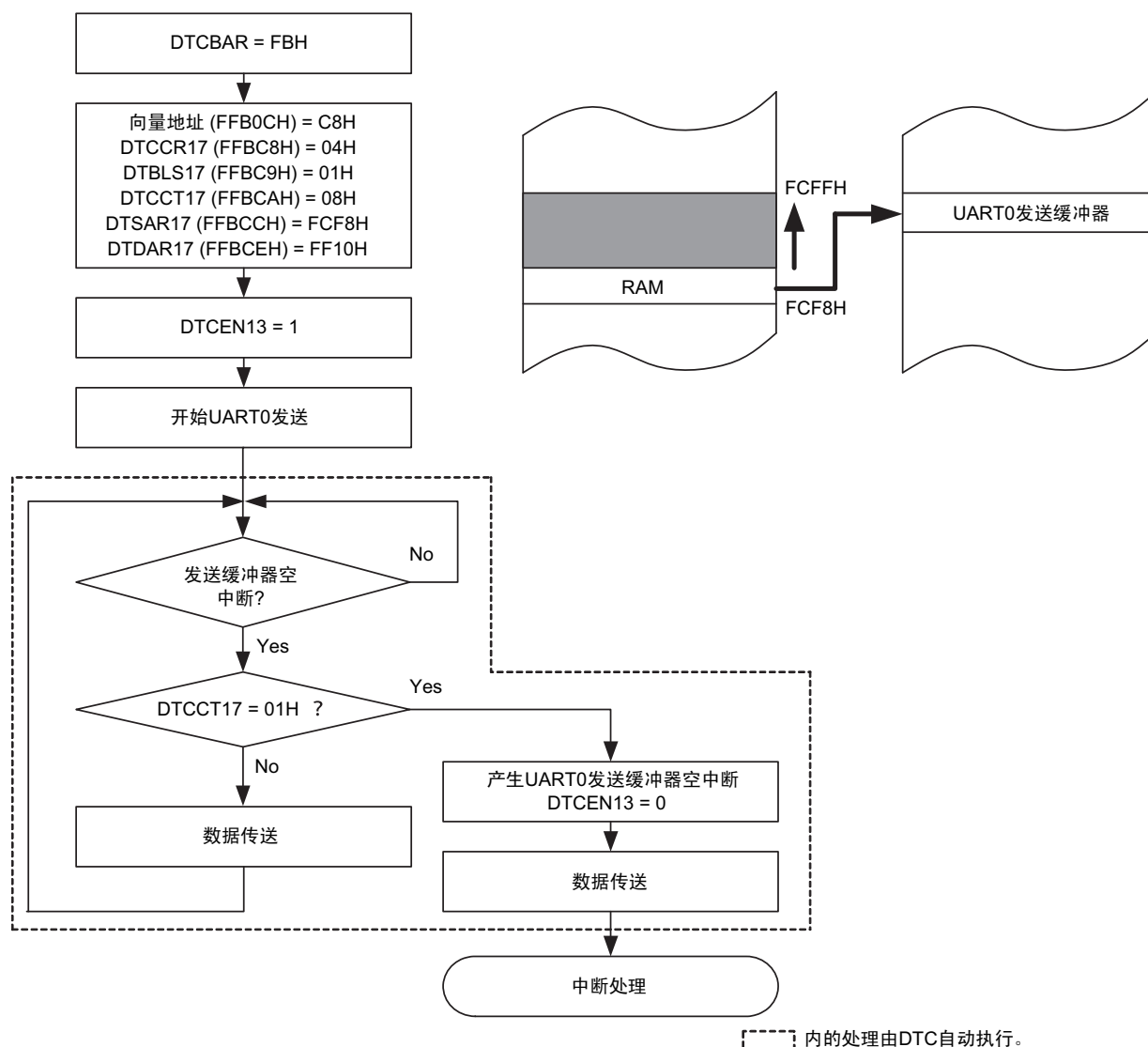
因为是正常模式，所以不使用 DTRL12 寄存器的值。但是，当通过 RAM 奇偶校验错误检测功能允许产生奇偶校验错误复位（RPERDIS=0）时，必须对 DTRL12 寄存器进行初始化（00H）。

(2) 正常模式的使用例子 2: UART0 连续发送

通过 UART0 的发送缓冲器空中断启动 DTC，并且将 RAM 的值传送到 UART0 的发送缓冲器。

- 向量地址分配在 FFB0CH，控制数据分配在 FFBC8H ~ FFBCFH。
- 将 RAM 的 FFCF8H ~ FFCFFH 的 8 字节传送到 UART0 的发送缓冲器 (FFF10H)。

图 16-16 正常模式的使用例子 2: UART0 连续发送



因为是正常模式，所以不使用 DTRL17 寄存器的值。但是，当通过 RAM 奇偶校验错误检测功能允许产生奇偶校验错误复位 (RPERDIS=0) 时，必须对 DTRL17 寄存器进行初始化 (00H)。

必须通过软件开始第 1 次的 UART0 发送。通过发送缓冲器空中断启动 DTC，然后自动进行第 2 次以后的发送。

16.4.3 重复模式

1 次启动的传送数据为 1 ~ 255 字节。将传送源或者传送目标指定为重复区，传送次数为 1 ~ 255 次。一旦指定次数的传送结束，就对 DTCCTj (j=0 ~ 23) 寄存器以及指定为重复区的地址进行初始化，然后重复进行传送。当 DTCCRj 寄存器的 RPTINT 位为“1”（允许发生中断）并且进行 DTCCTj 寄存器变为“0”的数据传送时，就在 DTC 运行中向中断控制器产生对应启动源的中断请求，并且将对应的 DTCENi (i=0 ~ 4) 寄存器的 DTCENi0 ~ DTCENi7 位置“0”（禁止启动）。当 DTCCRj 寄存器的 RPTINT 位为“0”（禁止发生中断）时，即使进行 DTCCTj 寄存器变为“0”的数据传送，也不产生中断请求，而且 DTCENi0 ~ DTCENi7 位不变为“0”。

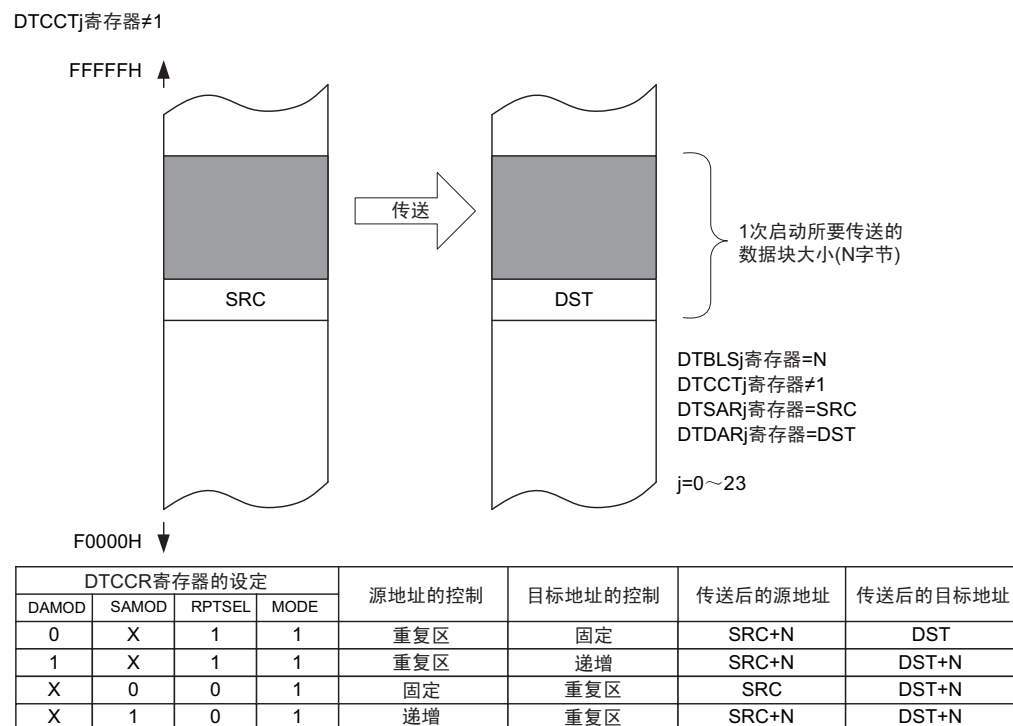
重复模式的寄存器功能和数据传送分别如表 16-7 和图 16-17 所示。

表 16-7 重复模式的寄存器功能

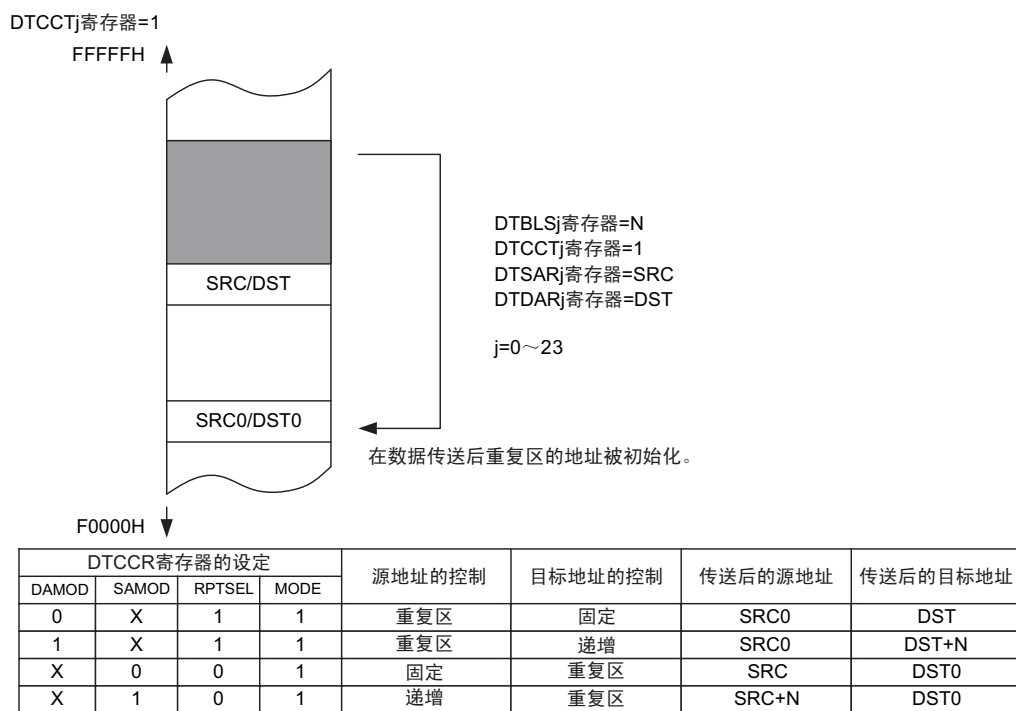
寄存器名	符号	功能
DTC 块大小寄存器 j	DTBLSj	1 次启动所要传送的数据块大小
DTC 传送次数寄存器 j	DTCCTj	数据的传送次数
DTC 传送次数重加载寄存器 j	DTRLDj	将此寄存器的值重新加载到 DTCCT 寄存器。 (对数据的传送次数进行初始化)
DTC 源地址寄存器 j	DTSARj	数据的传送源地址
DTC 目标地址寄存器 j	DTDARj	数据的传送目标地址

备注 j=0 ~ 23

图 16-17 重复模式的数据传送



X: “0”或者“1”



SRC0: 源地址的初始值

DST0: 目标地址的初始值

X: “0”或者“1”

注意 1. 在使用重复模式时, 必须将指定为重复区的地址初始值的低 8 位置 “00H”。

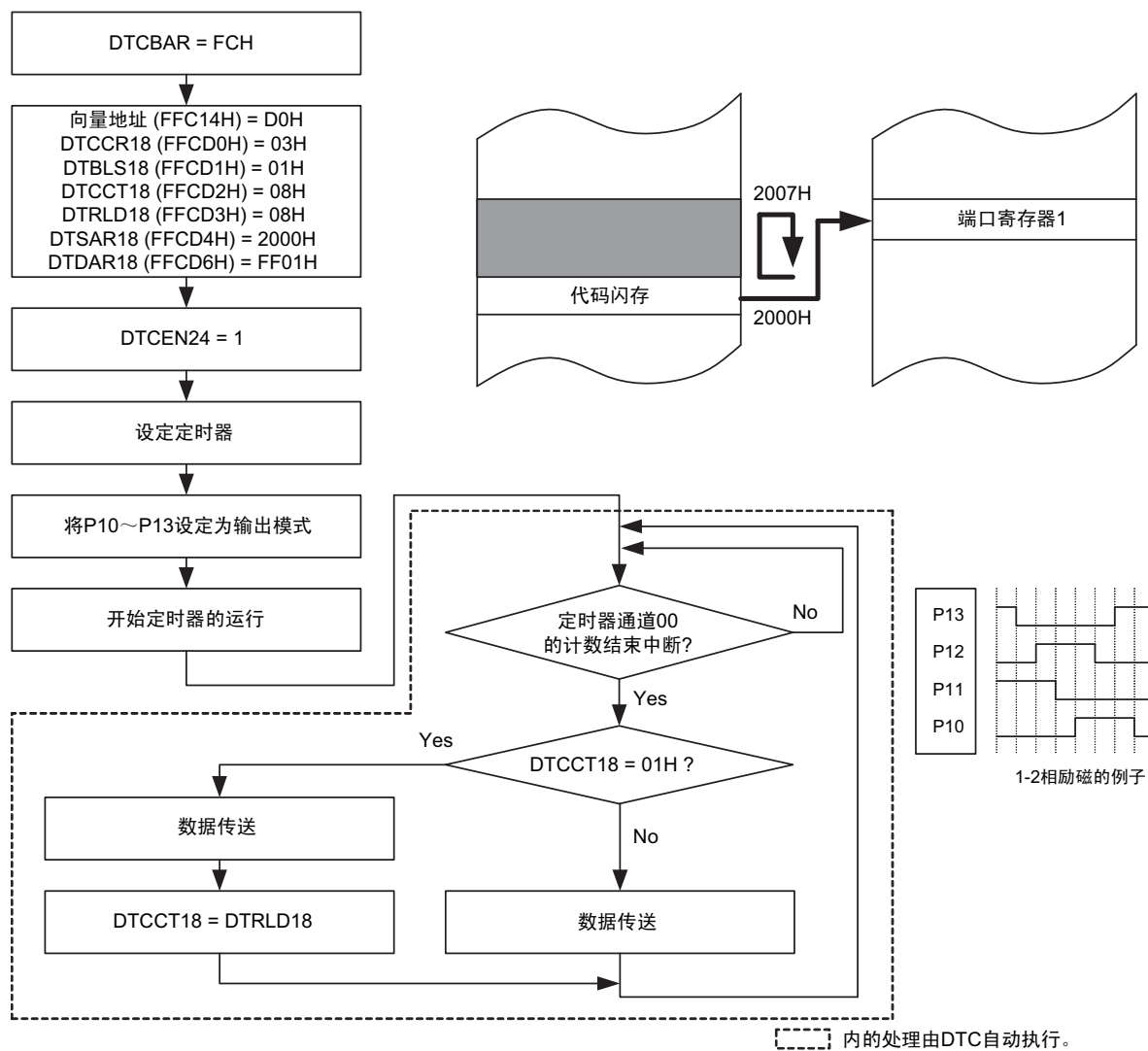
2. 在使用重复模式时, 必须将重复区的数据长度设定在 255 字节以内。

(1) 重复模式的使用例子 1: 使用端口输出步进马达控制脉冲

使用定时器阵列单元 0 的通道 0 间隔定时器功能启动 DTC，并且将保存在代码闪存的马达控制脉冲的模式传送到通用端口。

- 向量地址分配在 FFC14H，控制数据分配在 FFCD0H ~ FFCD7H。
- 将代码闪存的 02000H ~ 02007H 的 8 字节传送到镜像区 F2000H ~ F2007H 开始的端口寄存器 1 (FFF01H)。
- 禁止重复模式中断。

图 16-18 正常模式的使用例子 1: 使用端口输出步进马达控制脉冲



要停止输出时，必须在停止定时器的运行后清除 DTCEN24。

16.4.4 链传送

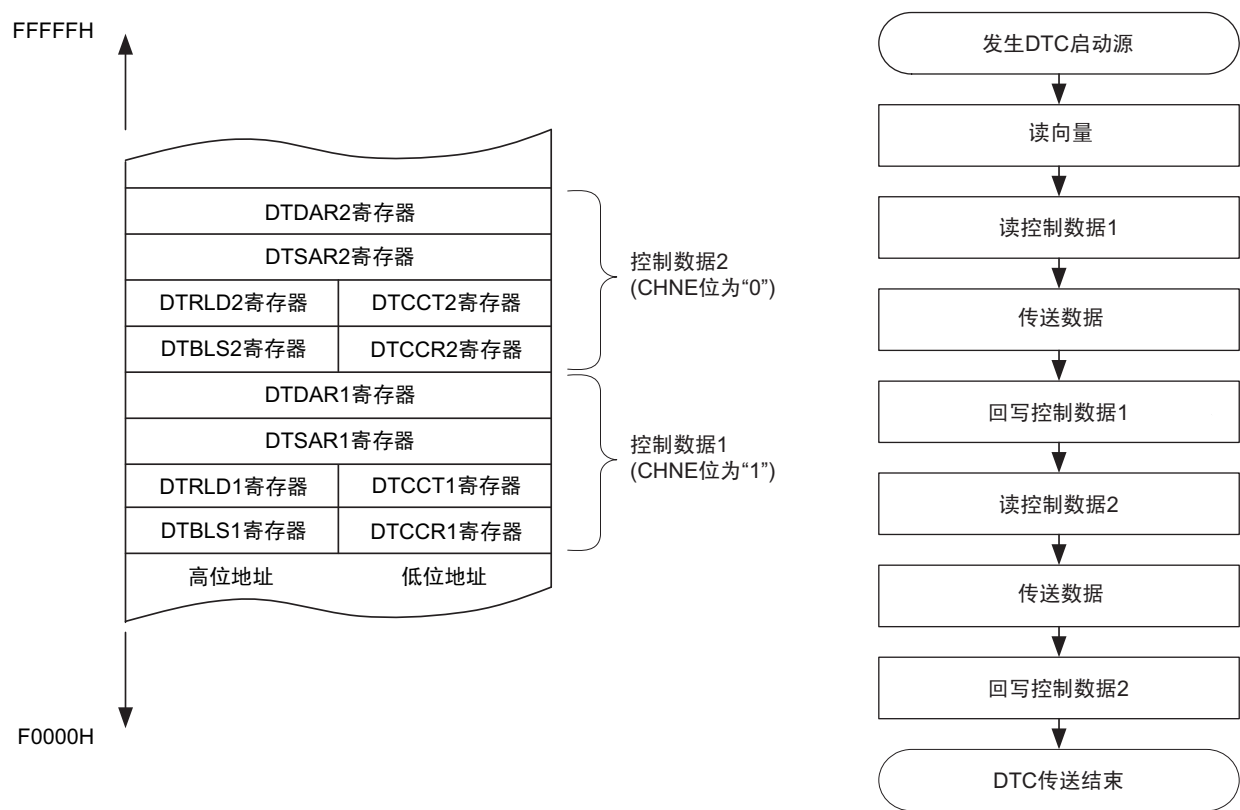
当 DTCCRj (j=0 ~ 22) 寄存器的 CHNE 位为“1（允许链传送）”时，能通过 1 个启动源连续进行多个数据的传送。

DTC 一旦启动，就通过从启动源对应的向量地址读取的数据来选择控制数据，读被分配在 DTC 控制数据区的控制数据。如果读到的控制数据的 CHNE 位为“1”（允许链传送），就在传送结束后读下一个被分配的控制数据，继续进行传送。重复此操作，直到 CHNE 位为“0”（禁止链传送）的控制数据传送结束为止。

在使用多个控制数据进行链传送时，第一个控制数据设定的传送次数有效，而第 2 个以后处理的控制数据的传送次数无效。

链传送的流程图如图 16-19 所示。

图 16-19 链传送的流程图



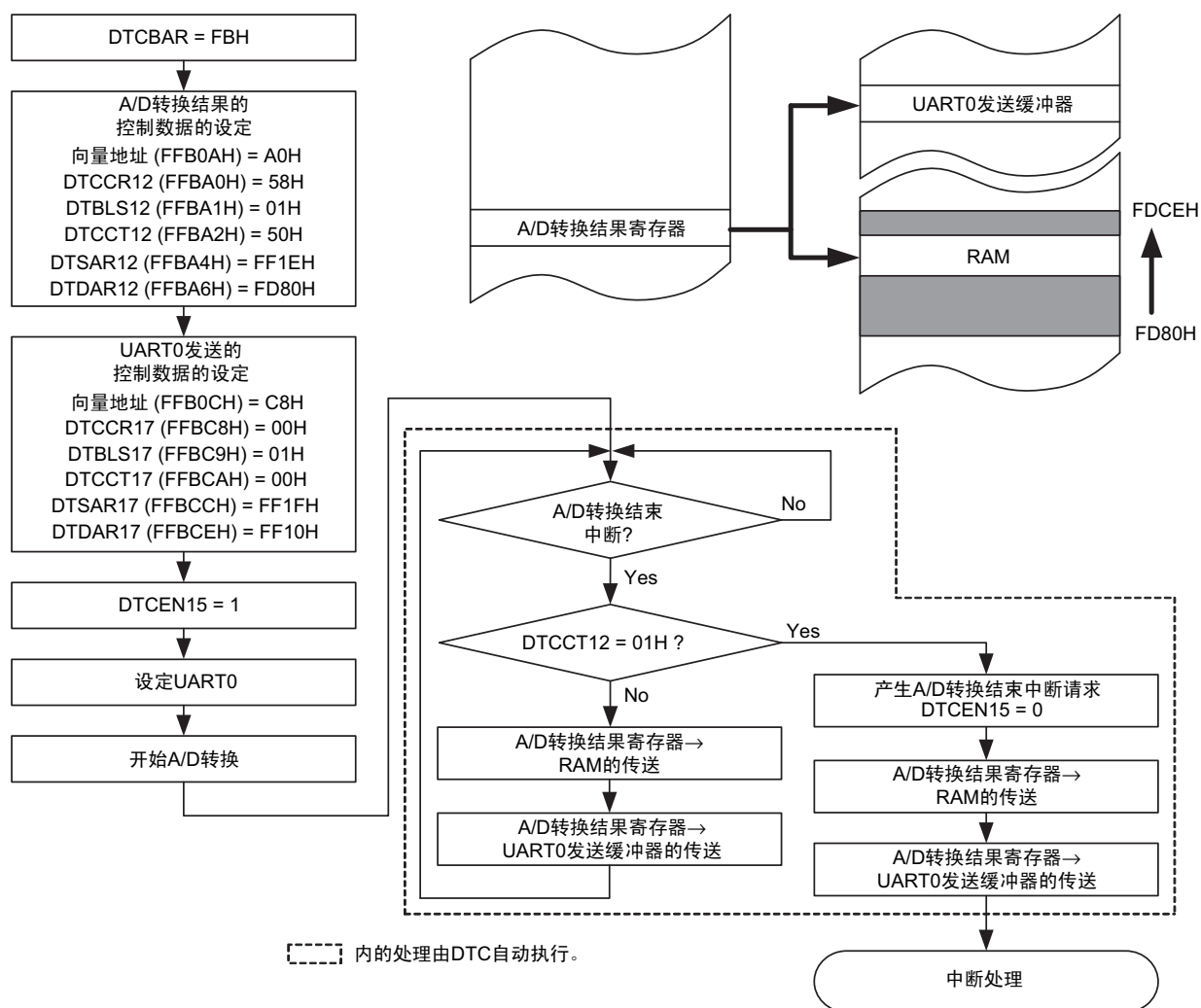
- 注意 1. 必须将 DTCCR23 寄存器的 CHNE 位置“0”（禁止链传送）。
2. 在链传送的第 2 次以后的数据传送时，DTCENi (i=0 ~ 4) 寄存器的 DTCENi0 ~ DTCENi7 位不变为“0”（禁止 DTC 启动），并且不产生中断请求。

(1) 链传送的使用例子：连续取 A/D 转换结果进行 UART0 发送

通过 A/D 转换结束中断启动 DTC，并且将 A/D 转换结果传送到 RAM 进行 UART0 发送。

- 向量地址为 FFB0AH。
- A/D 转换结果的控制数据分配在 FFBA0H ~ FFBA7H。
- UART0 发送的控制数据分配在 FFBA8H ~ FFBAFH。
- 将 A/D 转换结果寄存器 (FFF1FH、FFF1EH) 的 2 字节数据传送到 RAM 的 FFD80H ~ FFD CFH，并且将 A/D 转换结果寄存器的高位 1 字节 (FFF1FH) 传送到 UART 的发送缓冲器 (FFF10H)。

图 16-20 链传送的使用例子：连续取 A/D 转换结果进行 UART0 发送



16.5 使用 DTC 时的注意事项

16.5.1 DTC 控制数据和向量表的设定

- 不能通过 DTC 传送进行 DTC 的扩展特殊功能寄存器 (2nd SFR)、DTC 控制数据区、DTC 向量表区和通用寄存器 (FFEE0H ~ FFEFFH) 空间的存取。
- 必须在将全部的 DTC 启动源设定为禁止启动的状态下更改 DTC 基址寄存器 (DTCBAR)。
- 只能改写 1 次 DTC 基址寄存器 (DTCBAR)。
- 必须在对应的 DTCENi (i=0 ~ 4) 寄存器的 DTCENi0 ~ DTCENi7 位为“0” (禁止 DTC 启动) 时更改 DTCCRj、DTBLSj、DTCCTj、DTRLdj、DTSARj、DTDARj 寄存器的数据。
- 必须在对应的 DTCENi (i=0 ~ 4) 寄存器的 DTCENi0 ~ DTCENi7 位为“0” (禁止 DTC 启动) 时更改设定在向量表中的 DTC 控制数据区的起始地址。
- 在自编程或者改写数据闪存时, 不能将用于 DTC 的传送目标或者传送源的 RAM 地址设定为 FFE20H ~ FFEDFH 的区域。

16.5.2 DTC 控制数据区和 DTC 向量表区的分配

能分配 DTC 控制数据和向量表的区域因产品和使用条件而不同。

- 禁止将通用寄存器 (FFEE0H ~ FFEFFH) 的空间用作 DTC 控制数据区和 DTC 向量表区。
- 堆栈区、DTC 控制数据区和 DTC 向量表区不能重叠。
- 在使用自编程功能和数据闪存功能时, 不能将以下产品的内部 RAM 区用作 DTC 控制数据区和 DTC 向量表区。

R7F0C014B2D、R7F0C014L2D: FDF00H ~ FE309H

- 当通过 RAM 奇偶校验错误检测功能允许产生奇偶校验错误复位 (RPERDIS=0) 时, 即使在使用正常模式时也必须对 DTRLd 寄存器进行初始化 (00H)。

16.5.3 DTC 保留指令

即使发生 DTC 传送请求, 也在以下指令之后保留数据的传送。另外, 在 PREFIX 指令码和紧接之后的指令之间不启动 DTC。

- 调用返回指令
- 无条件转移指令
- 条件转移指令
- 代码闪存的读存取指令
- IFxx、MKxx、PRxx、PSW 的位操作指令和操作数含有 ES 寄存器的 8 位操作指令
- 数据闪存的存取指令

注意 1. 如果接受 DTC 传送请求, 就保留全部中断请求, 直到 DTC 传送结束为止。

2. 在 DTC 保留指令的 DTC 保留期间, 保留全部中断请求。

16.5.4 存取数据闪存空间时的运行

要在 DTC 的 1 条数据传送指令之后存取数据闪存空间时, 必须在指令之间插入 3 个时钟的等待。

指令 1

DTC 的数据传送

指令 2 ← 3 个时钟的等待

MOV A、! 数据闪存空间

16.5.5 DTC 的执行时钟数

DTC 启动时的执行情况和所需的时钟数如表 16-8 所示。

表 16-8 DTC 启动时的执行情况和所需的时钟数

读向量	控制数据		读数据	写数据
	读	回写		
1	4	注 1	注 2	注 2

注 1. 有关回写控制数据所需的时钟数, 请参照“表 16-9 回写控制数据所需的时钟数”。

2. 有关读写数据所需的时钟数, 请参照“表 16-10 读写数据所需的时钟数”。

表 16-9 回写控制数据所需的时钟数

DTCCR 寄存器的设定				地址设定		控制寄存器的回写				时钟数
DAMOD	SAMOD	RPTSEL	MODE	源	目标	DTCCTj 寄存器	DTRLdj 寄存器	DTSARj 寄存器	DTDARj 寄存器	
0	0	X	0	固定	固定	回写	回写	不回写	不回写	1
0	1	X	0	递增	固定	回写	回写	回写	不回写	2
1	0	X	0	固定	递增	回写	回写	不回写	回写	2
1	1	X	0	递增	递增	回写	回写	回写	回写	3
0	X	1	1	重复区	固定	回写	回写	回写	不回写	2
1	X	1	1		递增	回写	回写	回写	回写	3
X	0	0	1	固定	重复区	回写	回写	不回写	回写	2
X	1	0	1	递增		回写	回写	回写	回写	3

备注 j=0 ~ 23, X: “0” 或者 “1”

表 16-10 读写数据所需的时钟数

执行状态	RAM	代码闪存	数据闪存	特殊功能寄存器 (SFR)	扩展特殊功能寄存器 (2nd SFR)	
					无等待	等待
读数据	1	2	4	1	1	1+ 等待数 ^注
写数据	1	—	—	1	1	1+ 等待数 ^注

注 等待数因存取的扩展特殊功能寄存器 (2nd SFR) 的寄存器的规格而不同。

16.5.6 DTC 的响应时间

DTC 的响应时间如表 16-11 所示。DTC 的响应时间是指从检测到 DTC 启动源到开始 DTC 传送的时间，不包括 DTC 的执行时钟数。

表 16-11 DTC 的响应时间

	最短时间	最长时间
响应时间	3 个时钟	19 个时钟

但是，在以下情况下 DTC 的响应可能还会延迟。延迟的时钟数因条件而不同。

- 从内部RAM执行指令的情况
最长响应时间：20 个时钟
- 执行DTC保留指令的情况（参照“16.5.3 DTC保留指令”）
最长响应时间：各条件的最长响应时间+该条件的保留指令的执行时钟。
- 存取发生等待的TRJ0寄存器的情况
最长响应时间：各条件的最长响应时间+1 个时钟

备注 1 个时钟：1/f_{CLK}（f_{CLK}：CPU/ 外围硬件时钟）

16.5.7 DTC 的启动源

- 不能在从输入DTC启动源到结束DTC传送的期间输入相同的启动源。
- 在产生DTC启动源的位置，不能操作该启动源对应的DTC启动允许位。
- 如果DTC启动源发送竞争，就在CPU接受DTC传送时判断优先级，决定启动启动源。有关启动源的优先级，请参照“16.3.3 向量表”。

16.5.8 待机模式中的运行

状态	DTC 运行
HALT 模式	能运行（禁止在低功耗 RTC 模式中运行）。
STOP 模式	能接受 DTC 启动源 ^{注 2} 。
SNOOZE 模式	能运行 ^{注 1、3、4、5} 。

- 注 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 STOP 模式中，能通过 DTC 启动源的检测转移到 SNOOZE 模式进行 DTC 传送，并且在传送结束后返回到 STOP 模式。但是，因为在 SNOOZE 模式中代码闪存和数据闪存停止运行，所以不能将闪存设定为传送源。
3. 如果在使用 CSIp 的 SNOOZE 模式功能时将传送结束中断用作 DTC 启动源，就必须在 DTC 传送结束后通过传送结束中断来解除 SNOOZE 模式并且开始 CPU 处理，或者使用链传送并且通过 DTC 传送重新设定 CSIp 的接收（给 STm0 位写“1”，给 SWCm 位写“0”，设定 SSCm 寄存器并且给 SSm0 位写“1”）。
4. 如果在使用 UARTq 的 SNOOZE 模式功能时将传送结束中断用作 DTC 启动源，就必须在 DTC 传送结束后通过传送结束中断来解除 SNOOZE 模式并且开始 CPU 处理，或者使用链传送并且重新设定 UARTq 的接收（给 STm1 位写“1”，给 SWCm 位写“0”，设定 SSCm 寄存器并且给 SSm1 位写“1”）。
5. 如果在使用 A/D 转换器的 SNOOZE 模式功能时将 A/D 转换结束中断用作 DTC 启动源，就必须在 DTC 传送结束后通过 A/D 转换结束中断来解除 SNOOZE 模式并且开始 CPU 处理，或者使用链传送并且重新设定 A/D 转换器的 SNOOZE 模式功能（在给 AWC 位写“0”后给 AWC 位写“1”）。

备注 p=00, q=0, m=0

第 17 章 事件链接控制器 (ELC)

17.1 ELC 的功能

事件链接控制器 (ELC) 将各外围功能输出的事件进行外围功能之间的相互链接 (链接)。能通过事件链接不经过 CPU 而直接进行外围功能之间的协作运行。

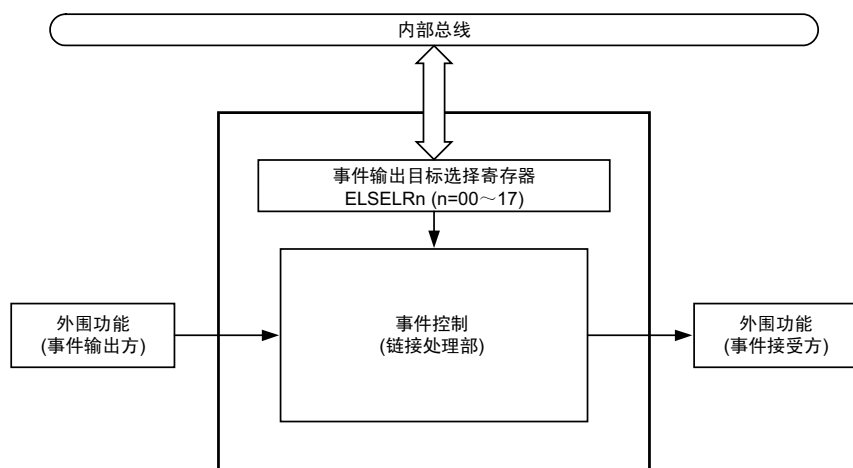
ELC 有以下功能:

- 能将 18 种外围功能的事件信号直接链接到指定的外围功能。
- 能将事件信号用作 6 种外围功能中的 1 种外围功能运行的启动源。

17.2 ELC 的结构

ELC 的框图如图 17-1 所示。

图 17-1 ELC 的框图



17.3 控制 ELC 的寄存器

控制 ELC 的寄存器如表 17-1 所示。

表 17-1 控制 ELC 的寄存器

寄存器名	符号
事件输出目标选择寄存器 00	ELSELR00
事件输出目标选择寄存器 01	ELSELR01
事件输出目标选择寄存器 02	ELSELR02
事件输出目标选择寄存器 03	ELSELR03
事件输出目标选择寄存器 04	ELSELR04
事件输出目标选择寄存器 05	ELSELR05
事件输出目标选择寄存器 06 注	ELSELR06
事件输出目标选择寄存器 07 注	ELSELR07
事件输出目标选择寄存器 08	ELSELR08
事件输出目标选择寄存器 09	ELSELR09
事件输出目标选择寄存器 10	ELSELR10
事件输出目标选择寄存器 11	ELSELR11
事件输出目标选择寄存器 12	ELSELR12
事件输出目标选择寄存器 13	ELSELR13
事件输出目标选择寄存器 14	ELSELR14
事件输出目标选择寄存器 15	ELSELR15
事件输出目标选择寄存器 16	ELSELR16
事件输出目标选择寄存器 17	ELSELR17

注 只限于 64 引脚产品。

17.3.1 事件输出目标选择寄存器 n (ELSELRn) (n=00 ~ 17)

ELSELRn 寄存器将各事件信号链接到事件接受方外围功能（链接目标外围功能）接受事件时的运行。

不能将多个事件输入链接到相同的事件输出目标（事件接受方）。否则，事件接受方外围功能的运行可能不定而无法正常地接受事件信号。另外，不能将事件链接发生源和事件输出目标设定为相同的功能。

必须在全部事件输出方的外围功能不产生事件信号的期间设定 ELSELRn 寄存器。

ELSELRn 寄存器（n=00 ~ 17）和外围功能的对应如表 17-2 所示，ELSELRn 寄存器（n=00 ~ 17）的设定值和链接目标外围功能接受事件时的运行的对应如表 17-3 所示。

图 17-2 事件输出目标选择寄存器 n (ELSELRn) 的格式

地址: F0300H (ELSELR00) ~ F0311H (ELSELR17)	复位后: 00H		R/W				
符号	7	6	5	4	3	2	1 0
ELSELRn	0	0	0	0	0	ELSELn2	ELSELn1 ELSELn0

ELSELn2	ELSELn1	ELSELn0	事件链接的选择
0	0	0	禁止事件链接。
0	0	1	选择所链接的外围功能 1 的运行注。
0	1	0	选择所链接的外围功能 2 的运行注。
0	1	1	选择所链接的外围功能 3 的运行注。
1	0	0	选择所链接的外围功能 4 的运行注。
1	0	1	选择所链接的外围功能 5 的运行注。
1	1	0	选择所链接的外围功能 6 的运行注。
上述以外			禁止设定。

注 请参照“表 17-3 ELSELRn 寄存器（n=00 ~ 17）的设定值和链接目标外围功能接受事件时的运行的对应”。

表 17-2 ELSELRn 寄存器 (n=00 ~ 17) 和外围功能的对应

寄存器名	事件发生源 (事件输入 n 的输出源)	事件内容
ELSELR00	外部中断边沿检测 0	INTP0
ELSELR01	外部中断边沿检测 1	INTP1
ELSELR02	外部中断边沿检测 2	INTP2
ELSELR03	外部中断边沿检测 3	INTP3
ELSELR04	外部中断边沿检测 4	INTP4
ELSELR05	外部中断边沿检测 5	INTP5
ELSELR06	键返回信号检测	INTKR
ELSELR07	RTC 固定周期 / 闹钟一致检测	INTRTC
ELSELR08	定时器 RD0 输入捕捉 A / 比较匹配 A	INTTRD0
ELSELR09	定时器 RD0 输入捕捉 B / 比较匹配 B	INTTRD0
ELSELR10	定时器 RD1 输入捕捉 A / 比较匹配 A	INTTRD1
ELSELR11	定时器 RD1 输入捕捉 B / 比较匹配 B	INTTRD1
ELSELR12	定时器 RD1 下溢	TRD1 下溢信号
ELSELR13	定时器 RJ0 下溢 / 脉宽测量期间结束 / 脉冲周期测量期间结束	INTTRJ0
ELSELR14	TAU 通道 00 的计数结束 / 捕捉结束	INTTM00
ELSELR15	TAU 通道 01 的计数结束 / 捕捉结束	INTTM01
ELSELR16	TAU 通道 02 的计数结束 / 捕捉结束	INTTM02
ELSELR17	TAU 通道 03 的计数结束 / 捕捉结束	INTTM03

表 17-3 ELSELn 寄存器 (n=00 ~ 17) 的设定值和链接目标外围功能接受事件时的运行的对应

ELSELn 寄存器的 ELSELn2 ~ ELSELn0 位	链接目标 No.	链接目标外围功能	接受事件时的运行
001B	1	A/D 转换器	开始 A/D 转换。
010B	2	定时器阵列单元 0 通道 0 的定时器输入 ^{注 1}	延迟计数器、输入脉冲间隔的测量、 外部事件计数器
011B	3	定时器阵列单元 0 通道 1 的定时器输入 ^{注 2}	延迟计数器、输入脉冲间隔的测量、 外部事件计数器
100B	4	定时器 RJ0	计数源
101B	5	定时器 RD0	TRDIOD0 的输入捕捉，脉冲输出的强制截止
110B	6	定时器 RD1	TRDIOD1 的输入捕捉，脉冲输出的强制截止

- 注 1. 要选择定时器阵列单元 0 通道 0 的定时器输入作为链接目标外围功能时，必须先通过定时器时钟选择寄存器 0 (TPS0) 将通道 0 的运行时钟设定为 f_{CLK} ，通过噪声滤波器允许寄存器 1 (NFEN1) 将 TI00 引脚的噪声滤波器设定为 OFF (TNFEN0=0)，并且通过定时器输入选择寄存器 0 (TIS0) 将通道 0 使用的定时器输入设定为 ELC 的事件输入信号。
2. 要选择定时器阵列单元 0 通道 1 的定时器输入作为链接目标外围功能时，必须先通过定时器时钟选择寄存器 0 (TPS0) 将通道 1 的运行时钟设定为 f_{CLK} ，通过噪声滤波器允许寄存器 1 (NFEN1) 将 TI01 引脚的噪声滤波器设定为 OFF (TNFEN01=0)，并且通过定时器输入选择寄存器 0 (TIS0) 将通道 1 使用的定时器输入设定为 ELC 的事件输入信号。

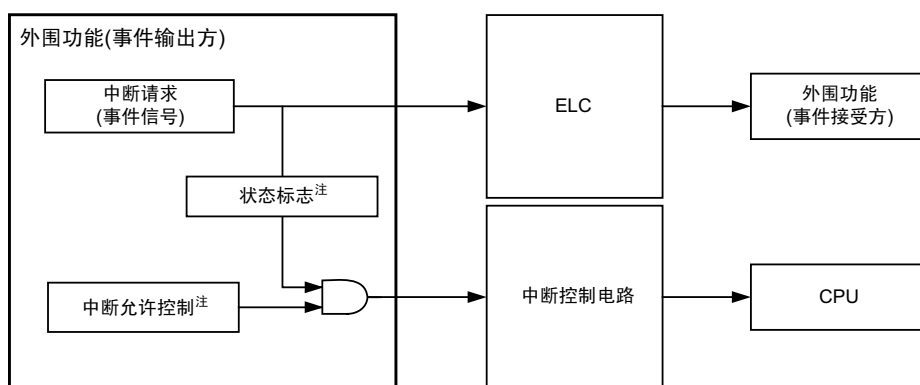
17.4 ELC 的运行

将各外围功能产生的事件信号用作中断控制电路的中断请求所使用的路径和用作 ELC 事件所使用的路径相互独立。因此，各事件信号与中断控制无关，能用作事件接受方外围功能运行的事件信号。

中断处理和 ELC 的关系如图 17-3 所示。此图是以有中断请求状态标志和中断允许位（控制允许或者禁止）的外围功能为例的关系。

通过 ELC 接受事件的外围功能的运行是根据接受方外围功能在接收事件后的运行（参照“表 17-3 ELSELn 寄存器（n=00 ~ 17）的设定值和链接目标外围功能接受事件时的运行的对应”）。

图 17-3 中断处理和 ELC 的关系



注 有些外围功能没有此功能。

接受事件的外围功能的响应如表 17-4 所示。

表 17-4 接受事件的外围功能的响应

事件接受目标 No.	事件链接目标的功能	事件接受后的运行	响应
1	A/D 转换器	A/D 转换	ELC 事件直接变为 A/D 转换的硬件触发。
2	定时器阵列单元 0 通道 0 的定时器输入	延迟计数器 输入脉宽的测量 外部事件计数器	在从发生 ELC 事件经过 3 个或者 4 个 f_{CLK} 周期后进行边沿的检测。
3	定时器阵列单元 0 通道 1 的定时器输入	延迟计数器 输入脉宽的测量 外部事件计数器	在从发生 ELC 事件经过 3 个或者 4 个 f_{CLK} 周期后进行边沿的检测。
4	定时器 RJ	计数源	ELC 事件直接变为定时器 RJ 的计数源。
5	定时器 RD0	TRDIOD0 的输入捕捉	在从发生 ELC 事件经过 2 个或者 3 个定时器 RD 的运行时钟周期后产生计数开始触发。
		脉冲输出的强制截止	在从发生 ELC 事件经过 2 个或者 3 个定时器 RD 的运行时钟周期后变为强制截止状态。
6	定时器 RD1	TRDIOD1 的输入捕捉	在从发生 ELC 事件经过 2 个或者 3 个定时器 RD 的运行时钟周期后产生计数开始触发。
		脉冲输出的强制截止	在从发生 ELC 事件经过 2 个或者 3 个定时器 RD 的运行时钟周期后变为强制截止状态。

第 18 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。

		32 引脚	64 引脚
可屏蔽中断	外部	6	13
	内部	23	24

18.1 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 18-1。

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。

18.2 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 18-1）。当发生复位或者各种中断请求而进行转移时，因为保存程序起始地址的向量代码各为两个字节，所以中断的转移目标地址为 00000H~0FFFFH 的 64K 地址。

表 18-1 中断源一览表 (1/4)

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	64 引脚	32 引脚
		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	0004H	(A)	☐	☐
	1	INTLVI	电压检测注 4		0006H		☐	☐
	2	INTP0	引脚输入边沿的检测	外部	0008H	(B)	☐	☐
	3	INTP1			000AH		☐	☐
	4	INTP2			000CH		☐	☐
	5	INTP3			000EH		☐	☐
	6	INTP4			0010H		☐	☐
	7	INTP5			0012H		☐	☐
	8	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	0014H	(A)	☐	☐
	9	INTSR2/ INTCSI21/ INTIIC21	UART2 接收的传送结束 / CSI21 的传送结束或者缓冲器空中断 / IIC21 的传送结束		0016H		☐	☐ 注 5
	10	INTSRE2	发生 UART2 接收的通信错误		0018H		☐	☐
	11	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		001EH		☐	☐
	12	INTSR0/ INTCSI01/ INTIIC01	UART0 接收的传送结束 / CSI01 的传送结束或者缓冲器空中断 / IIC01 的传送结束		0020H		☐	☐ 注 6
	13	INTSRE0	发生 UART0 接收的通信错误		0022H		☐	☐
		INTTM01H	定时器通道 01 的计数结束或者捕捉结束 (高 8 位定时器工作时)	☐		☐		

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，35 表示最低优先级。

2. 基本构成类型 (A) ~ (D) 分别对应图 18-1 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。

5. 只限于 INTSR2。

6. 只限于 INTSR0。

表 18-1 中断源一览表 (2/4)

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	64 引脚	32 引脚
		名称	触发					
可屏蔽	14	INTST1/ INTCSI10/ INTIIC10	UART1 发送的传送结束或者缓冲器空中断 / CSI10 的传送结束或者缓冲器空中断 / IIC10 的传送结束	内部	0024H	(A)	○	○ 注 3
	15	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		0026H		○	○
	16	INTSRE1	发生 UART1 接收的通信错误		0028H		○	○
		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)				○	○
	17	INTIICA0	IICA0 通信结束		002AH		○	○
	18	INTTM00	定时器通道 00 的计数结束或者捕捉结束		002CH		○	○
	19	INTTM01	定时器通道 01 的计数结束或者捕捉结束		002EH		○	○
	20	INTTM02	定时器通道 02 的计数结束或者捕捉结束		0030H		○	○
	21	INTTM03	定时器通道 03 的计数结束或者捕捉结束		0032H		○	○
	22	INTAD	A/D 转换结束		0034H		○	○
	23	INTRTC	实时时钟的固定周期 / 闹钟一致检测		0036H		○	—
	24	INTIT	间隔信号的检测		0038H		○	○
	25	INTKR	键返回信号的检测	外部	003AH	(C)	○	—
	26	INTTRJ0	定时器 RJ 的下溢	内部	0040H	(B)	○	○

- 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，35 表示最低优先级。
2. 基本构成类型 (A) ~ (D) 分别对应图 18-1 的 (A) ~ (D)。
3. 只限于 INTST1。

表 18-1 中断源一览表 (3/4)

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	64 引脚	32 引脚
		名称	触发					
可屏蔽	27	INTP6	引脚输入边沿的检测	外部	004AH	(B)	○	—
	28	INTP7			004CH		○	—
	29	INTP8			004EH		○	—
	30	INTP9			0050H		○	—
	31	INTP10			0052H		○	—
	32	INTP11			0054H		○	—
	33	INTTRD0	定时器 RD0 的输入捕捉、比较匹配、 上溢、下溢中断	内部	0056H	(A)	○	○
	34	INTTRD1	定时器 RD1 的输入捕捉、比较匹配、 上溢、下溢中断		0058H		○	○
	35	INTFL	保留注 3		0062H		○	○

- 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，35 表示最低优先级。
2. 基本构成类型 (A) ~ (D) 分别对应图 18-1 的 (A) ~ (D)。
3. 用于闪存自编程库或者数据闪存库。

表 18-1 中断源一览表 (4/4)

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	64 引脚	32 引脚
		名称	触发					
软件	—	BRK	BRK 指令的执行	—	007EH	(D)	○	○
复位	—	RESET	RESET 引脚的输入	—	0000H	—	○	○
		POR	上电复位				○	○
		LVD	电压检测注 3				○	○
		WDT	看门狗定时器的上溢				○	○
		TRAP	非法指令的执行注 4				○	○
		IAW	非法存储器的存取				○	○
		RPE	RAM 奇偶校验错误				○	○

注 1. 在发生多个可屏蔽中断时，默认优先级决定优先顺序。0 表示最高优先级，35 表示最低优先级。

2. 基本构成类型 (A) ~ (D) 分别对应图 18-1 的 (A) ~ (D)。

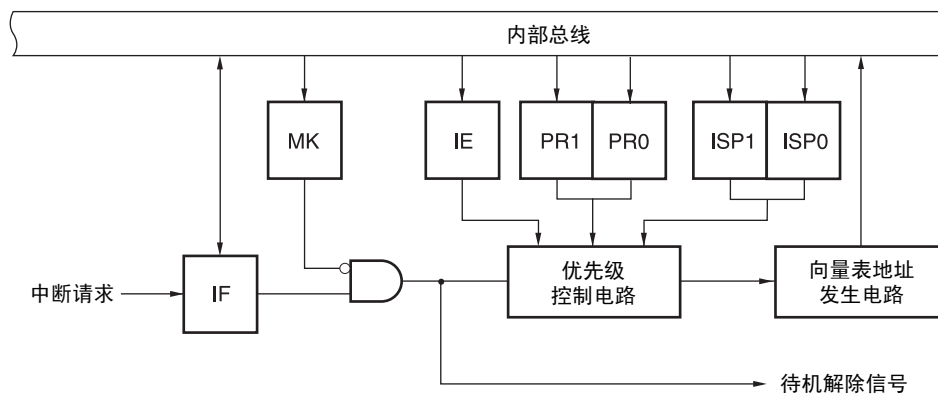
3. 这是将电压检测电平寄存器 (LVIS) 的 bit7 (LVIMD) 置“1”的情况。

4. 在执行指令码 FFH 时发生。

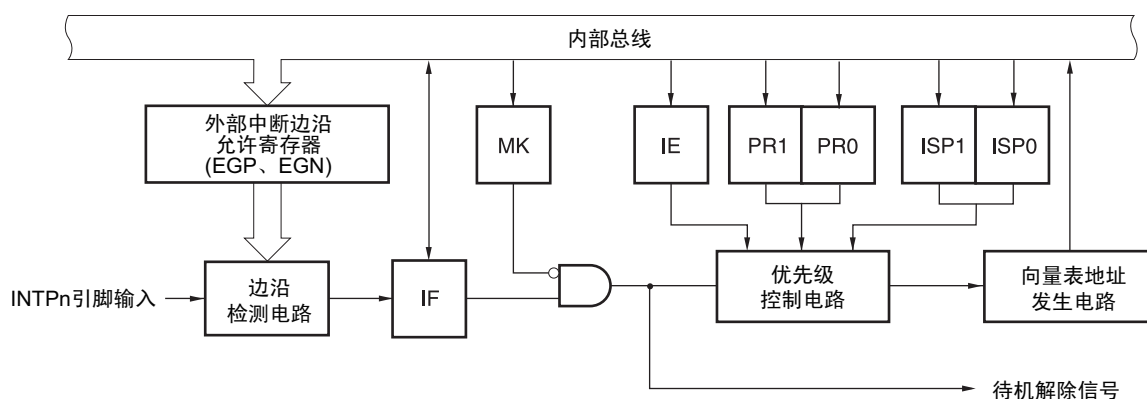
在通过在线仿真器或者片上调试仿真器进行仿真时，不会因执行非法指令而产生复位。

图 18-1 中断功能的基本结构

(A) 内部可屏蔽中断



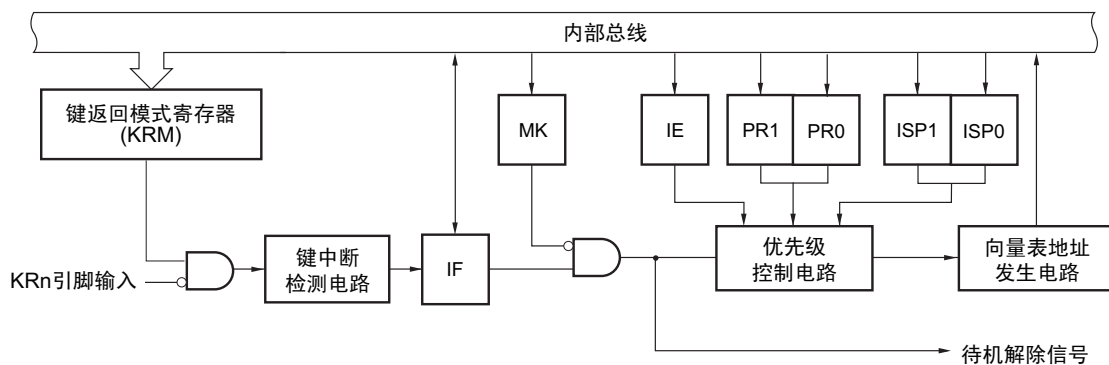
(B) 外部可屏蔽中断 (INTPn)



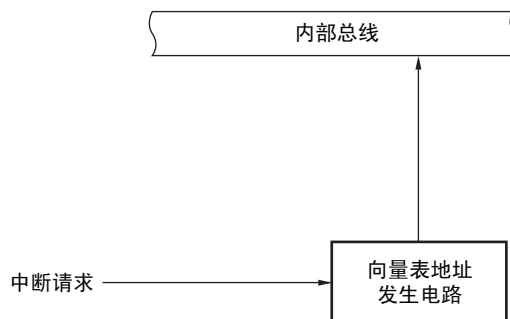
IF : 中断请求标志
 IE : 中断允许标志
 ISP0 : 接受中断的优先级标志 0
 ISP1 : 接受中断的优先级标志 1
 MK : 中断屏蔽标志
 PR0 : 优先级指定标志 0
 PR1 : 优先级指定标志 1

备注 32 引脚: n=0 ~ 5
 64 引脚: n=0 ~ 11

(C) 外部可屏蔽中断 (INTKR)



(D) 软件中断



- IF : 中断请求标志
 IE : 中断允许标志
 ISP0 : 接受中断的优先级标志 0
 ISP1 : 接受中断的优先级标志 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 64 引脚: n=0 ~ 7

18.3 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、EGP1）
- 外部中断下降沿允许寄存器（EGN0、EGN1）
- 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 18-2 所示。

表 18-2 中断请求源对应的各种标志 (1/4)

中断源	中断请求标志		中断屏蔽标志		优先级指定标志		64 位	32 位
		寄存器		寄存器		寄存器		
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0、WDTIPR1	PR00L、PR10L	○	○
INTLVI	LVIIF		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 18-2 中断请求源对应的各种标志 (2/4)

中断源	中断请求标志		中断屏蔽标志		优先级指定标志		PR00H	PR10H
		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	CSIF20 注 1		CSIMK20 注 1		CSIPR020、CSIPR120 注 1		○	○
INTIIC20 注 1	IICIF20 注 1		IICMK20 注 1		IICPR020、IICPR120 注 1		○	○
INTSR2 注 2	SRIF2 注 2		SRMK2 注 2		SRPR02、SRPR12 注 2		○	○
INTCSI21 注 2	CSIF21 注 2		CSIMK21 注 2		CSIPR021、CSIPR121 注 2		○	—
INTIIC21 注 2	IICIF21 注 2		IICMK21 注 2		IICPR021、IICPR121 注 2		○	—
INTSRE2	SREIF2		SREMK2		SREPR02、SREPR12		○	○
INTST0 注 3	STIF0 注 3		STMK0 注 3		STPR00、STPR10 注 3		○	○
INTCSI00 注 3	CSIF00 注 3		CSIMK00 注 3		CSIPR000、CSIPR100 注 3		○	○
INTIIC00 注 3	IICIF00 注 3		IICMK00 注 3		IICPR000、IICPR100 注 3		○	○
INTSR0 注 4	SRIF0 注 4		SRMK0 注 4		SRPR00、SRPR10 注 4		○	○
INTCSI01 注 4	CSIF01 注 4		CSIMK01 注 4		CSIPR001、CSIPR101 注 4		○	—
INTIIC01 注 4	IICIF01 注 4		IICMK01 注 4		IICPR001、IICPR101 注 4		○	—
INTSRE0 注 5	SREIF0 注 5		SREMK0 注 5		SREPR00、SREPR10 注 5		○	○
INTTM01H 注 5	TMIF01H 注 5		TMMK01H 注 5		TMPR001H、TMPR101H 注 5		○	○

- 注 1. 如果发生 INTST2、INTCSI20 或者 INTIIC20 的中断源, 就将 IF0H 寄存器的 bit0 置“1”。MK0H、PR00H、PR10H 寄存器的 bit0 对应这 3 个中断源。
2. 如果发生 INTSR2、INTCSI21 或者 INTIIC21 的中断源, 就将 IF0H 寄存器的 bit1 置“1”。MK0H、PR00H、PR10H 寄存器的 bit1 对应这 3 个中断源。
3. 如果发生 INTST0、INTCSI00 或者 INTIIC00 的中断源, 就将 IF0H 寄存器的 bit5 置“1”。MK0H、PR00H、PR10H 寄存器的 bit5 对应这 3 个中断源。
4. 如果发生 INTSR0、INTCSI01 或者 INTIIC01 的中断源, 就将 IF0H 寄存器的 bit6 置“1”。MK0H、PR00H、PR10H 寄存器的 bit6 对应这 3 个中断源。
5. 因为 UART0 接收的错误中断和 TAU0 的通道 1 (高 8 位定时器工作时) 的中断共用中断请求源对应的各种标志, 所以不能同时使用。当不使用 UART0 接收的错误中断 (EOC01=0) 时, 能同时使用 UART0 和 TAU0 的通道 1 (高 8 位定时器工作时)。如果发生 INTSRE0 或者 INTTM01H 的中断源, 就将 IF0H 寄存器的 bit7 置“1”。MK0H、PR00H、PR10H 寄存器的 bit7 对应这 2 个中断源。

表 18-2 中断请求源对应的各种标志 (3/4)

中断源	中断请求标志		中断屏蔽标志		优先级指定标志		INTST1	INTCS10
		寄存器		寄存器		寄存器		
INTST1 注 1	STIF1 注 1	IF1L	STMK1 注 1	MK1L	STPR01、STPR11 注 1	PR01L、PR11L	○	○
INTCSI10 注 1	CSIF10 注 1		CSIMK10 注 1		CSIPR010、CSIPR110 注 1		○	—
INTIIC10 注 1	IICIF10 注 1		IICMK10 注 1		IICPR010、IICPR110 注 1		○	—
INTSR1 注 2	SRIF1 注 2		SRMK1 注 2		SRPR01、SRPR11 注 2		○	○
INTCSI11 注 2	CSIF11 注 2		CSIMK11 注 2		CSIPR011、CSIPR111 注 2		○	○
INTIIC11 注 2	IICIF11 注 2		IICMK11 注 2		IICPR011、IICPR111 注 2		○	○
INTSRE1 注 3	SREIF1 注 3		SREMK1 注 3		SREPR01、SREPR11 注 3		○	○
INTTM03H 注 3	TMIF03H 注 3		TMMK03H 注 3		TMPR003H、TMPR103H 注 3		○	○
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	○	○
INTRTC	RTCIF		RTCMK		RTCPR0、RTCPR1		○	—
INTIT	ITIF		ITMK		ITPR0、ITPR1		○	○
INTKR	KRIF		KRMK		KRPR0、KRPR1		○	—
INTTRJ0	TRJIF0		TRJMK0		TRJPR00、TRJPR10		○	○

- 注 1. 如果发生 INTST1、INTCSI10 或者 INTIIC10 的中断源, 就将 IF1L 寄存器的 bit0 置“1”。MK1L、PR01L、PR11L 寄存器的 bit0 对应这 3 个中断源。
2. 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。
3. 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。

表 18-2 中断请求源对应的各种标志 (4/4)

中断源	中断请求标志		中断屏蔽标志		优先级指定标志		PR02L PR12L	PR02H PR12H
		寄存器		寄存器		寄存器		
INTP6	PIF6	IF2L	PMK6	MK2L	PPR06、PPR16	PR02L、 PR12L	○	—
INTP7	PIF7		PMK7		PPR07、PPR17		○	—
INTP8	PIF8		PMK8		PPR08、PPR18		○	—
INTP9	PIF9		PMK9		PPR09、PPR19		○	—
INTP10	PIF10		PMK10		PPR010、PPR110		○	—
INTP11	PIF11	IF2H	PMK11	MK2H	PPR011、PPR111	PR02H、 PR12H	○	—
INTTRD0	TRDIF0		TRDMK0		TRDPR00、TRDPR10		○	○
INTTRD1	TRDIF1		TRDMK1		TRDPR01、TRDPR11		○	○
INTFL	FLIF		FLMK		FLPR0、FLPR1		○	○

18.3.1 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 18-2 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	LVIF	WDIIF		

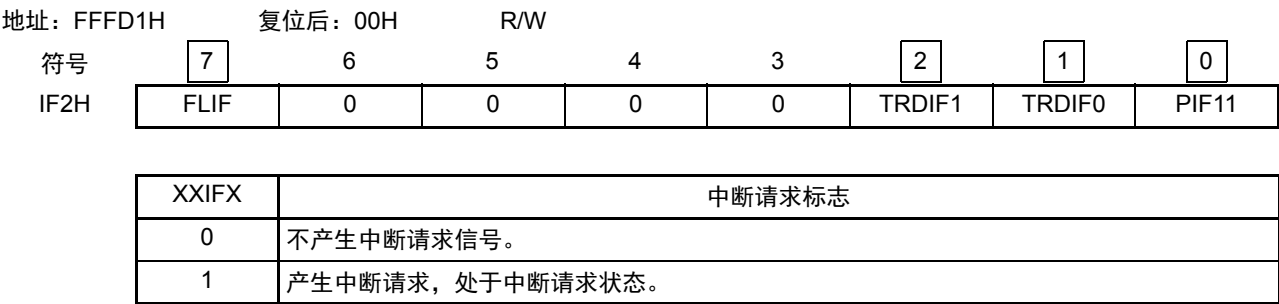
地址: FFFE1H	复位后: 00H	R/W								
符号	7	6	5	4	3	2	1	0		
IF0H	SREIF0 TMIF01H	SRIF0 CSIF01 IICIF01	STIF0 CSIF00 IICIF00	0	0	SREIF2	SRIF2 CSIF21 IICIF21	STIF2 CSIF20 IICIF20		

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 CSIF10 IICIF10		

地址: FFFE3H	复位后: 00H	R/W								
符号	7	6	5	4	3	2	1	0		
IF1H	0	TRJIF0	0	0	KRIF	ITIF	RTCIF	ADIF		

地址: FFFD0H	复位后: 00H	R/W								
符号	7	6	5	4	3	2	1	0		
IF2L	PIF10	PIF9	PIF8	PIF7	PIF6	0	0	0		

图 18-2 中断请求标志寄存器（IF0L、IF0H、IF1L、IF1H、IF2L、IF2H）的格式 (2/2)



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 18-2。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。

18.3.2 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。

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 18-3 中断屏蔽标志寄存器（MK0L、MK0H、MK1L、MK1H、MK2L、MK2H）的格式 (1/2)

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 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	1	1	SREMK2	SRMK2 CSIMK21 IICMK21	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 CSIMK10 IICMK10

地址: FFFE7H	复位后: FFH		R/W					
符号	7	6	5	4	3	2	1	0
MK1H	1	TRJMK0	1	1	KRMK	ITMK	RTCMK	ADMK

地址: FFFD4H	复位后: FFH		R/W					
符号	7	6	5	4	3	2	1	0
MK2L	PMK10	PMK9	PMK8	PMK7	PMK6	1	1	1

图 18-3 中断屏蔽标志寄存器（MK0L、MK0H、MK1L、MK1H、MK2L、MK2H）的格式 (2/2)

地址: FFFD5H	复位后: FFH	R/W						
符号	7	6	5	4	3	2	1	0
MK2H	FLMK	1	1	1	1	TRDMK1	TRDMK0	PMK11

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 18-2。必须给未分配的位设定初始值。

18.3.3 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 18-4 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 (1/3)

地址: FFFE8H	复位后: FFH		R/W						
符号	7	6	5	4	3	2	1	0	
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0	

地址: FFECH	复位后: FFH		R/W					
符号	7	6	5	4	3	2	1	0
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

地址: FFFE9H	复位后: FFH		R/W						
符号	7	6	5	4	3	2	1	0	
PR00H	SREPR00 TMPR001H	SRPR00 CSIPR001 IICPR001	STPR00 CSIPR000 IICPR000	1	1	SREPR02	SRPR02 CSIPR021 IICPR021	STPR02 CSIPR020 IICPR020	

地址: FFFEDH		复位后: FFH		R/W				
符号	7	6	5	4	3	2	1	0
PR10H	SREPR10 TMPR101H	SRPR10 CSIPR101 IICPR101	STPR10 CSIPR100 IICPR100	1	1	SREPR12	SRPR12 CSIPR121 IICPR121	STPR12 CSIPR120 IICPR120

地址: FFFEAH	复位后: FFH		R/W					
符号	7	6	5	4	3	2	1	0
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	IICAPR00	SREPR01 TMPR003H	SRPR01 CSIPR011 IICPR011	STPR01 CSIPR010 IICPR010

图 18-4 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 (2/3)

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 CSIPR110 IICPR110

地址: FFFEBH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR01H	1	TRJPR00	1	1	KRPR0	ITPR0	RTCPR0	ADPR0

地址: FFFEFH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR11H	1	TRJPR10	1	1	KRPR1	ITPR1	RTCPR1	ADPR1

地址: FFFD8H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR02L	PPR010	PPR09	PPR08	PPR07	PPR06	1	1	1

地址: FFFDCH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR12L	PPR110	PPR19	PPR18	PPR17	PPR16	1	1	1

地址: FFFD9H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR02H	FLPR0	1	1	1	1	TRDPR01	TRDPR00	PPR011

地址: FFFDDH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR12H	FLPR1	1	1	1	1	TRDPR11	TRDPR10	PPR111

图 18-4 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 (3/3)

XXPR1X	XXPR0X	优先级的选择
0	0	指定优先级 0（高优先级）。
0	1	指定优先级 1。
1	0	指定优先级 2。
1	1	指定优先级 3（低优先级）。

注意 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 18-2。必须给未分配的位设定初始值。

18.3.4 外部中断上升沿允许寄存器（EGP0、EGP1）和外部中断下降沿允许寄存器（EGN0、EGN1）

这些寄存器设定 INTP0 ~ INTP11 的有效边沿。

通过 1 位或者 8 位存储器操作指令设定 EGP0、EGP1、EGN0、EGN1 寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

图 18-5 外部中断上升沿允许寄存器（EGP0、EGP1）和外部中断下降沿允许寄存器（EGN0、EGN1）的格式

地址: FFF38H	复位后: 00H	R/W								
符号	7	6	5	4	3	2	1	0		
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0		

地址: FFF39H	复位后: 00H	R/W								
符号	7	6	5	4	3	2	1	0		
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0		

地址: FFF3AH	复位后: 00H	R/W								
符号	7	6	5	4	3	2	1	0		
EGP1	0	0	0	0	EGP11	EGP10	EGP9	EGP8		

地址: FFF3BH	复位后: 00H	R/W								
符号	7	6	5	4	3	2	1	0		
EGN1	0	0	0	0	EGN11	EGN10	EGN9	EGN8		

EGPn	EGNn	INTPn 引脚有效边沿的选择 (n=0 ~ 11)
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 18-3 所示。

表 18-3 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	64 引脚	32 引脚
EGP0	EGN0	INTP0	○	○
EGP1	EGN1	INTP1	○	○
EGP2	EGN2	INTP2	○	○
EGP3	EGN3	INTP3	○	○
EGP4	EGN4	INTP4	○	○
EGP5	EGN5	INTP5	○	○
EGP6	EGN6	INTP6	○	—
EGP7	EGN7	INTP7	○	—
EGP8	EGN8	INTP8	○	—
EGP9	EGN9	INTP9	○	—
EGP10	EGN10	INTP10	○	—
EGP11	EGN11	INTP11	○	—

注意 如果将外部中断功能使用的输入端口切换到输出模式，就可能检测有效边沿，产生 INTPn 中断。
当切换到输出模式时，必须在设定为禁止边沿检测后（EGPn、EGNn=0、0）将端口模式寄存器（PMxx）置“0”。

备注 1. 有关边沿检测的端口，请参照“2.1 端口功能”。

2. n=0 ~ 11

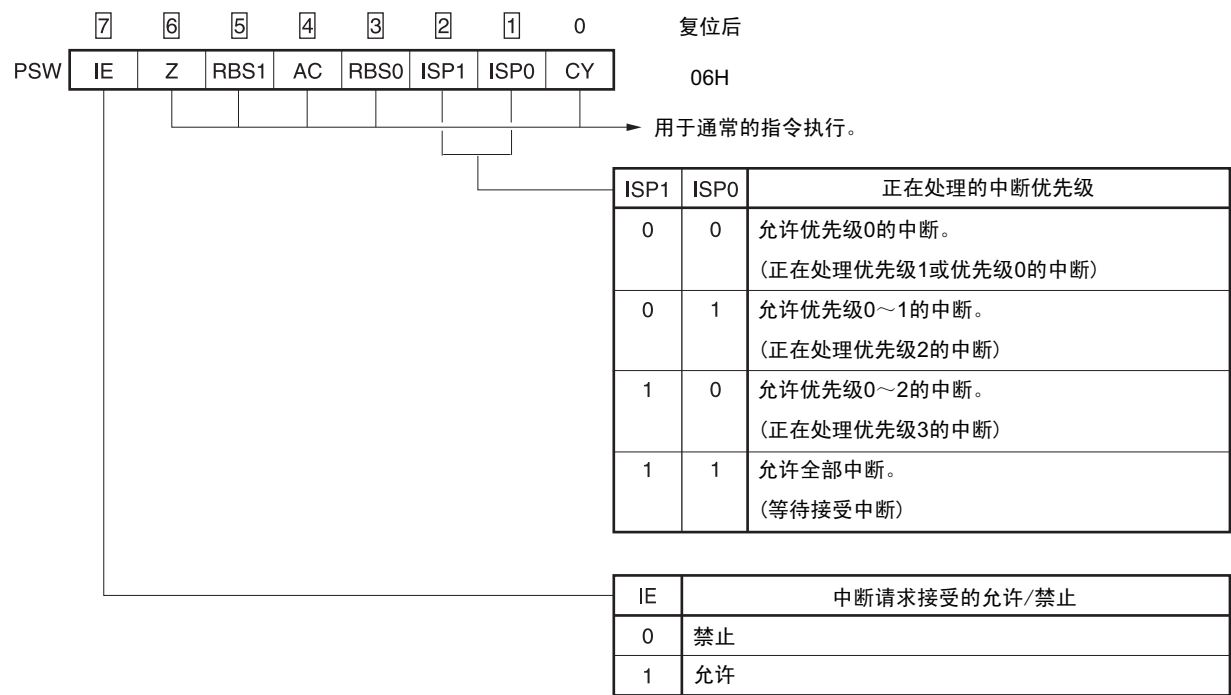
18.3.5 程序状态字（PSW）

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 18-6 程序状态字的结构



18.4 中断处理的操作

18.4.1 可屏蔽中断请求的接受

如果中断请求标志被置“1”并且该中断请求的屏蔽（MK）标志已被清“0”，就进入能接受可屏蔽中断请求的状态，在中断允许状态（IE 标志已被置“1”）下接受向量中断请求。但是，在处理优先级高的中断请求的过程中，不接受被指定的低优先级的中断请求。

从产生可屏蔽中断请求到进行向量中断处理的时间如表 18-4 所示。

有关中断请求的接受时序，请参照图 18-8 和图 18-9。

表 18-4 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 执行内部 RAM 区的指令时除外。

备注 1 个时钟： $1/f_{\text{CLK}}$ （ f_{CLK} ：CPU 时钟）

如果同时发生多个可屏蔽中断请求，就从优先级指定标志所指定的高优先级的请求开始接受。如果优先级指定标志所指定的优先级相同，就从默认优先级高的请求开始接受。

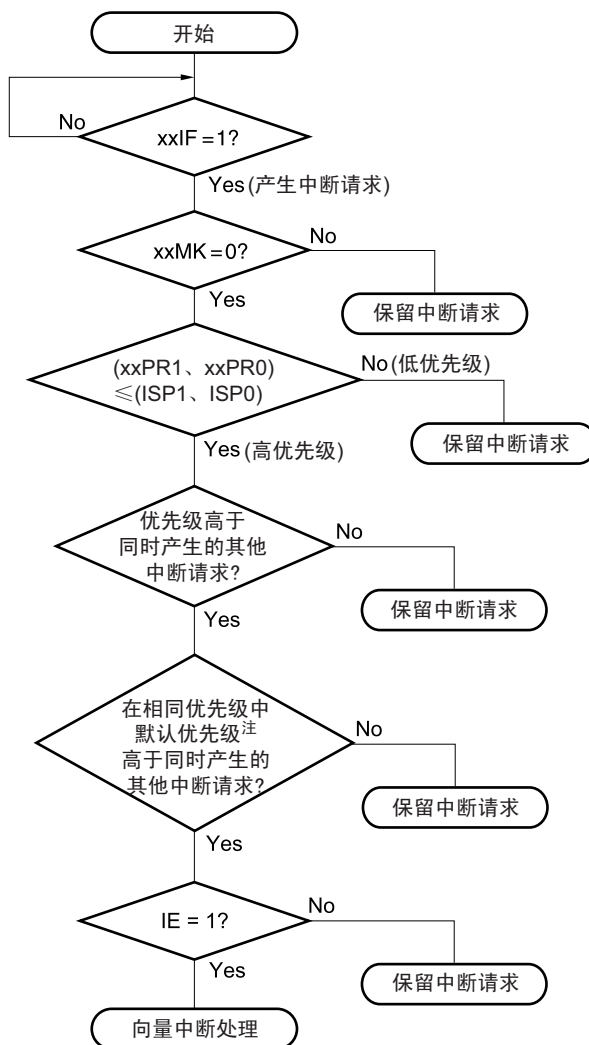
被保留的中断请求在可接受的状态时被接受。

中断请求的接受算法如图 18-7 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 18-7 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 18-6)

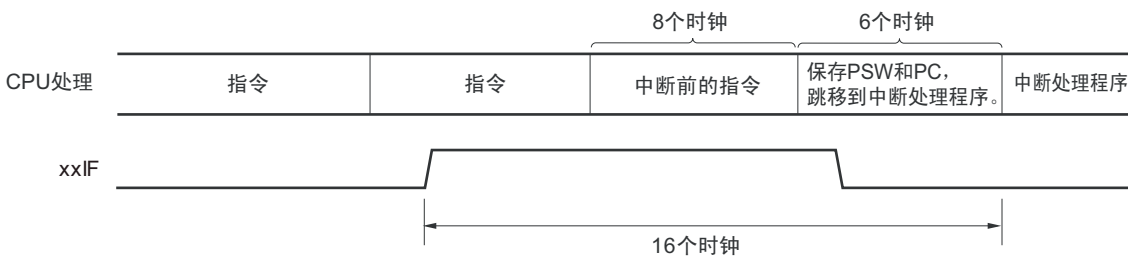
注 有关默认优先级, 请参照“表 18-1 中断源一览表”。

图 18-8 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 18-9 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)

18.4.2 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。

18.4.3 多重中断处理

在中断处理过程中接受了另一个中断请求的情况称为多重中断。

除非中断请求为接受允许状态（IE=1），否则不会发生多重中断。在接受中断请求时，中断请求为接受被禁止状态（IE=0）。因此，如果要允许多重中断，就需要在中断处理过程中用 EI 指令将 IE 标志置“1”，进入中断允许状态。

另外，即使在中断允许状态下也可能不允许多重中断，由中断优先级进行控制。中断优先级有默认优先级和可编程优先级，通过可编程优先级控制多重中断。

在中断允许状态下，如果产生的中断请求的优先级高于正在处理的中断优先级，就作为多重中断进行接受。如果产生的中断请求的优先级等于或者低于正在处理的中断优先级，就不作为多重中断进行接受。但是，如果在处理优先级 0 的中断过程中将 IE 标志置“1”，就允许优先级 0 的其他中断。

保留因中断被禁止或者优先级低而不允许多重中断的中断请求，在当前的中断处理结束后至少执行一条主处理的指令，然后接受被保留的中断请求。

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 18-5 和图 18-10 所示。

表 18-5 在中断处理过程中能发生多重中断的中断请求的关系

多重中断请求 正在处理的中断		可屏蔽中断请求								软件中 断请求
		优先级 0 (PR=00)		优先级 1 (PR=01)		优先级 2 (PR=10)		优先级 3 (PR=11)		
		IE=1	IE=0	IE=1	IE=0	IE=1	IE=0	IE=1	IE=0	
可屏蔽中断	ISP1=0 ISP0=0	○	×	×	×	×	×	×	×	○
	ISP1=0 ISP0=1	○	×	○	×	×	×	×	×	○
	ISP1=1 ISP0=0	○	×	○	×	○	×	×	×	○
	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）。

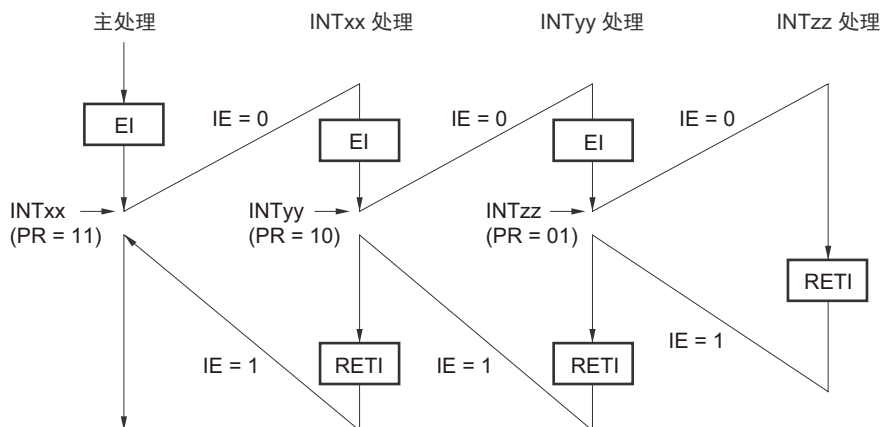
PR=01：通过 xxPR1x=0、xxPR0x=1 指定优先级 1。

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）。

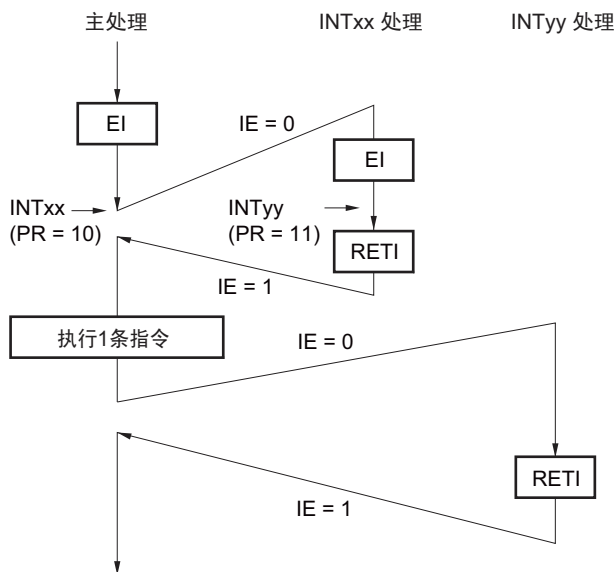
图 18-10 多重中断的例子 (1/2)

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。

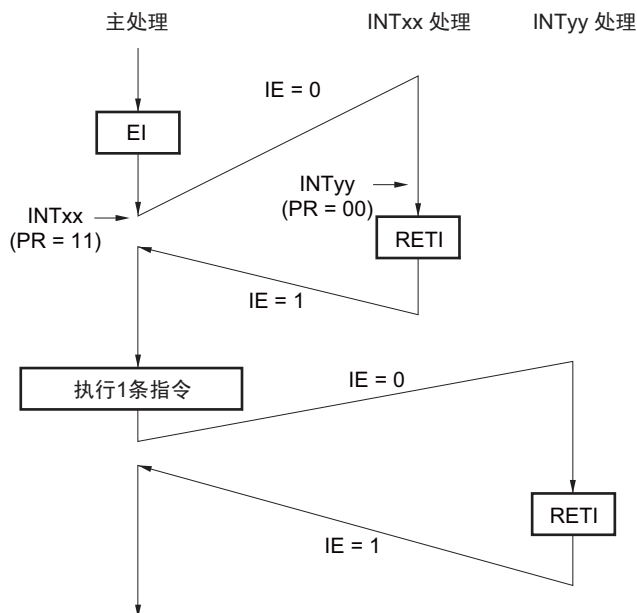
PR=11 : 通过 xxPR1x=1、xxPR0x=1 指定优先级 3（低优先级）。

IE=0 : 禁止接受中断请求。

IE=1 : 允许接受中断请求。

图 18-10 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。

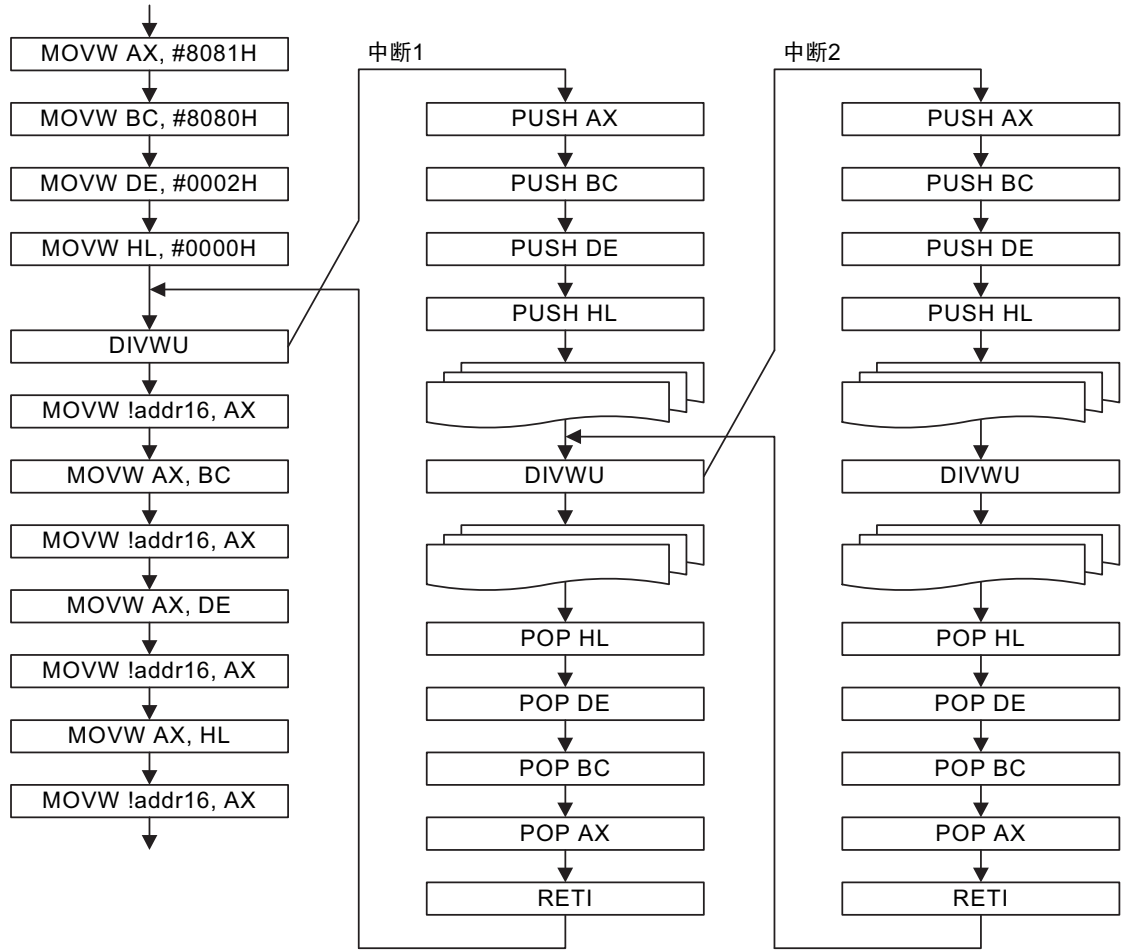
18.4.4 除法运算指令执行过程中的中断处理

在执行除法运算指令时，为了提高中断响应，R7F0C014B2D、R7F0C014L2D 支持 DIVHU/DIVWU 指令执行过程中的中断。

- 如果在执行DIVHU/DIVWU指令的过程中发生中断，就中止DIVHU/DIVWU指令的执行。
- 在中止指令的执行后，PC指示DIVHU/DIVWU的下一条指令。
- 通过下一条指令产生中断。
- 为了重新执行DIVHU/DIVWU指令，将PC-3压栈。

通常中断	DIVHU/DIVWU 指令执行过程中的中断
(SP-1)←PSW	(SP-1)←PSW
(SP-2)←(PC) _S	(SP-2)←(PC-3) _S
(SP-3)←(PC) _H	(SP-3)←(PC-3) _H
(SP-4)←(PC) _L	(SP-4)←(PC-3) _L
PC _S ←0000	PC _S ←0000
PC _H ←(Vector)	PC _H ←(Vector)
PC _L ←(Vector)	PC _L ←(Vector)
SP←SP-4	SP←SP-4
IE←0	IE←0

DIVHU/DIVWU 使用 AX、BC、DE、HL 寄存器。因此在中断处理时，必须在将 AX、BC、DE、HL 寄存器的内容压栈后使用。



18.4.5 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 18-11 所示。

图 18-11 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 19 章 键中断功能

键中断输入的通道数因产品而不同。

	32 引脚	64 引脚
键中断输入通道	—	8ch

19.1 键中断的功能

能通过给键中断输入引脚（KR0 ~ KR7）输入下降沿，产生键中断（INTKR）。

表 19-1 键中断检测引脚的分配

键中断引脚	键返回模式寄存器（KRM）
KR0	KRM0
KR1	KRM1
KR2	KRM2
KR3	KRM3
KR4	KRM4
KR5	KRM5
KR6	KRM6
KR7	KRM7

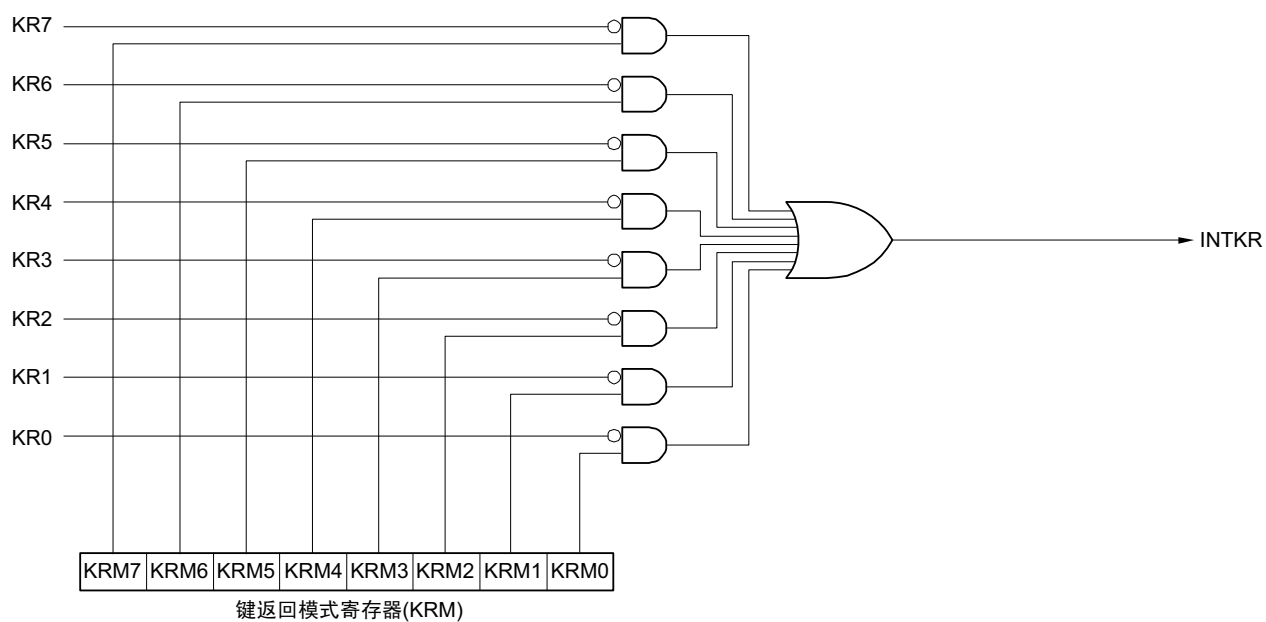
19.2 键中断的结构

键中断由以下硬件构成。

表 19-2 键中断的结构

项目	控制寄存器
控制寄存器	键返回模式寄存器（KRM） 端口模式寄存器 7（PM7）

图 19-1 键中断的框图



19.3 控制键中断的寄存器

通过以下寄存器控制键中断功能。

- 键返回模式寄存器（KRM）
- 端口模式寄存器 7（PM7）

19.3.1 键返回模式寄存器（KRM）

KRM0 ~ KRM7 位控制 KR0 ~ KR7 信号。

通过 1 位或者 8 位存储器操作指令设定 KRM 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 19-2 键返回模式寄存器（KRM）的格式

地址: FFF37H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	键中断模式的控制
0	不检测键中断信号。
1	检测键中断信号。

注意 1. 能通过将键中断输入引脚的上拉电阻寄存器 7（PU7）的对象位置“1”，使用内部上拉电阻。

2. 如果在给键中断输入引脚输入低电平的状态下将 KRM 寄存器的对象位置位，就产生中断。

要想忽视此中断时，必须在通过中断屏蔽标志禁止中断处理后设定 KRM 寄存器。然后，必须在等待键中断输入的低电平宽度（ t_{KR} ）（请参照“31.4 AC 特性”）后清除中断请求标志，允许中断处理。

3. 在键中断模式中未使用的引脚能用作通常的端口。

备注 n=0 ~ 7

19.3.2 端口模式寄存器 7（PM7）

当用作键中断输入引脚（KR0～KR7）时，必须将 PM7n 位分别置“1”。此时，P7n 的输出锁存器可以是“0”或者“1”。通过 1 位或者 8 位存储器操作指令设定 PM7 寄存器。在产生复位信号后，此寄存器的值变为“FFH”。
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。

图 19-3 端口模式寄存器 7（PM7）的格式

地址：FFF27H	复位后：FFH	R/W						
符号	7	6	5	4	3	2	1	0
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70
PM7n	P7n 引脚的输入 / 输出模式的选择（n=0～7）							
0	输出模式（输出缓冲器 ON）							
1	输入模式（输出缓冲器 OFF）							

第 20 章 待机功能

20.1 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、高速内部振荡器或者副系统时钟振荡电路正在振荡，各时钟就继续振荡。虽然此模式无法让工作电流降到 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（INTRTC/INTIT）或者 ELC 事件输入）产生的 A/D 转换请求和 DTC 启动源，解除 STOP 模式，不需要 CPU 运行而进行 CSIp 或者 UARTq 的数据接收、A/D 转换以及 DTC 运行。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。当 CPU 以副系统时钟运行时，不能设定为 STOP 模式。无论 CPU 是以主系统时钟还是以副系统时钟运行，都能使用 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 m（SSCm）和 A/D 转换器模式寄存器 2（ADM2）。详细内容请参照“14.3 控制串行阵列单元的寄存器”和“13.3 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 26 章 选项字节”。

备注 p=00, q=0, m=0

20.2 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13章 A/D转换器”和“第14章 串行阵列单元”。

20.3 待机功能的运行

20.3.1 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或者副系统时钟，都能进行设定。

HALT 模式中的运行状态如下所示。

注意 当中断屏蔽标志为“0”（允许中断处理）并且中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除 HALT 模式。因此，即使在此情况下执行 HALT 指令，也不转移到 HALT 模式。

表 20-1 HALT 模式中的运行状态 (1/2)

HALT 模式的设定 项目		在 CPU 以主系统时钟运行的过程中执行 HALT 指令的情况		
		CPU 以高速内部振荡器时钟 (f_{IH}) 运行	CPU 以 X1 时钟 (f_X) 运行	CPU 以外部主系统时钟 (f_{EX}) 运行
系统时钟		停止给 CPU 提供时钟。		
主系统时钟	f_{IH}	继续运行 (不能停止)。	禁止运行。	
	f_X	禁止运行。	继续运行 (不能停止)。	不能运行。
	f_{EX}		不能运行。	继续运行 (不能停止)。
副系统时钟	f_{XT}	保持设定为 HALT 模式前的状态。		
	f_{EXS}			
低速内部振荡器时钟	f_{IL}	通过选项字节 (000C0H) 的 bit0 (WDSTBYON) 和 bit4 (WDTON) 以及副系统时钟提供模式控制寄存器 (OSMC) 的 WUTMMCK0 位进行设定。 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 DTC 时, 能运行)。		
端口 (锁存器)		保持设定为 HALT 模式前的状态。		
定时器阵列单元		能运行。		
实时时钟 (RTC)				
12 位间隔定时器				
看门狗定时器		参照“第 12 章 看门狗定时器”。		
定时器 RJ		能运行。		
定时器 RD				
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)				
数据传送控制器 (DTC)				
事件链接控制器 (ELC)		能在可运行的功能块之间进行链接。		
上电复位功能		能运行。		
电压检测功能				
外部中断				
键中断功能				
CRC 运算功能	高速 CRC			
	通用 CRC	在 RAM 区的运算中执行 DTC 时, 能运行。		
非法存储器存取检测功能		在执行 DTC 时, 能运行。		
RAM 奇偶校验功能				
RAM 保护功能				
SFR 保护功能				

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

f_{XT} : XT1 时钟

f_{EXS} : 外部副系统时钟

表 20-1 HALT 模式中的运行状态 (2/2)

HALT 模式的设定			在 CPU 以副系统时钟运行的过程中执行 HALT 指令的情况	
项目			CPU 以 XT1 时钟 (f_{XT}) 运行	CPU 以外部副系统时钟 (f_{EXS}) 运行
系统时钟			停止给 CPU 提供时钟。	
主系统时钟	f_{IH}		禁止运行。	
	f_X			
	f_{EX}			
副系统时钟	f_{XT}	继续运行（不能停止）。	不能运行。	
	f_{EXS}	不能运行。	继续运行（不能停止）。	
低速内部振荡器时钟	f_{IL}	通过选项字节（000C0H）的 bit0（WDSTBYON）和 bit4（WDTON）以及副系统时钟提供模式控制寄存器（OSMC）的 WUTMMCK0 位进行设定。 • 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 DTC 时，能运行）。	
端口（锁存器）			保持设定为 HALT 模式前的状态。	
定时器阵列单元			当 RTCLPC=0 时，能运行（否则禁止运行）。	
实时时钟（RTC）			能运行。	
12 位间隔定时器			参照“第 12 章 看门狗定时器”。	
看门狗定时器				
定时器 RJ				
定时器 RD				
时钟输出 / 蜂鸣器输出				
A/D 转换器			禁止运行。	
串行阵列单元（SAU）			当 RTCLPC=0 时，能运行（否则禁止运行）。	
串行接口（IICA）			禁止运行。	
数据传送控制器（DTC）			当 RTCLPC=0 时，能运行（否则禁止运行）。	
事件链接控制器（ELC）			能在可运行的功能块之间进行链接。	
上电复位功能			能运行。	
电压检测功能				
外部中断				
键中断功能				
CRC 运算功能	高速 CRC		禁止运行。	
	通用 CRC		在 RAM 区的运算中执行 DTC 时，能运行。	
非法存储器存取检测功能			在执行 DTC 时，能运行。	
RAM 奇偶校验错误检测功能				
RAM 保护功能				
SFR 保护功能				

备注 停止运行: 在转移到 HALT 模式时自动停止运行。

禁止运行: 在转移到 HALT 模式前停止运行。

f_{IH} : 高速内部振荡器时钟

f_X : X1 时钟

f_{XT} : XT1 时钟

f_{IL} : 低速内部振荡器时钟

f_{EX} : 外部主系统时钟

f_{EXS} : 外部副系统时钟

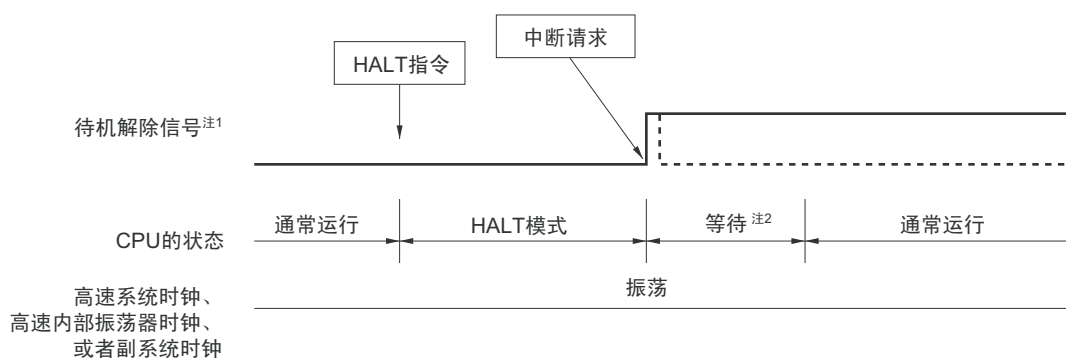
(2) HALT 模式的解除

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 20-1 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 18-1 中断功能的基本结构”。

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
副系统时钟 (RTCLPC=0)	: 10 ~ 11 个时钟
副系统时钟 (RTCLPC=1)	: 11 ~ 12 个时钟

• 不进行向量中断处理的情况

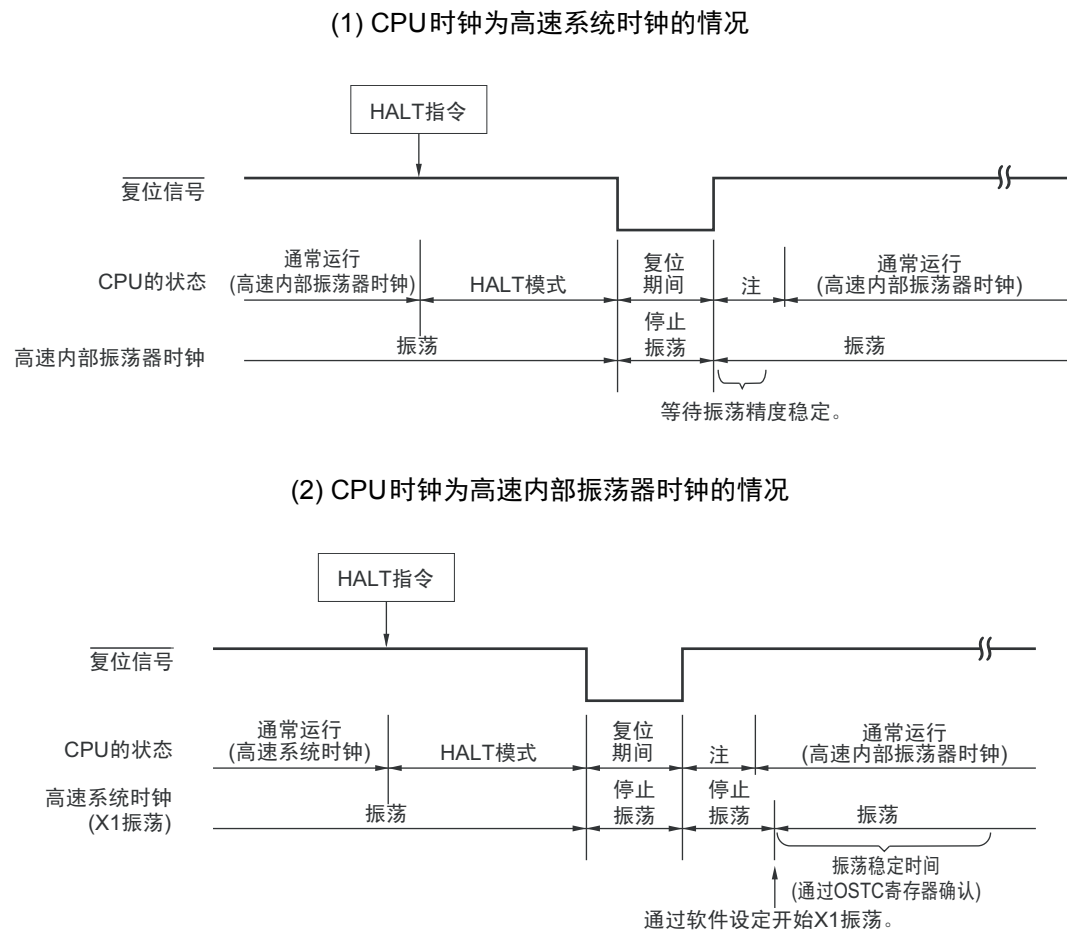
主系统时钟	: 9 ~ 10 个时钟
副系统时钟 (RTCLPC=0)	: 4 ~ 5 个时钟
副系统时钟 (RTCLPC=1)	: 5 ~ 6 个时钟

备注 虚线表示接受解除了待机模式的中断请求的情况。

(b) 通过产生复位信号进行的解除

通过产生复位信号来解除 HALT 模式。然后，和通常的复位一样，在转移到复位向量地址后执行程序。

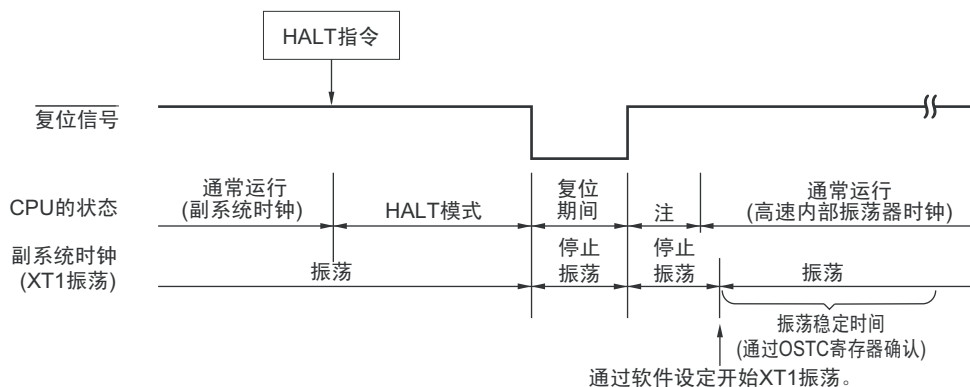
图 20-2 通过复位解除 HALT 模式 (1/2)



注 有关复位处理时间，请参照“第21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22章 上电复位电路”。

图 20-2 通过复位解除 HALT 模式 (2/2)

(3) CPU时钟为副系统时钟的情况



注 有关复位处理时间，请参照“第21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22章 上电复位电路”。

20.3.2 STOP 模式

(1) STOP 模式的设定和运行状态

通过执行 STOP 指令设定为 STOP 模式。只有在设定前的 CPU 时钟为主系统时钟的情况下才能进行设定。

注意 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。

备注 p=00, q=0, m=0

STOP 模式中的运行状态如下所示。

表 20-2 STOP 模式中的运行状态

STOP 模式的设定 项目		在 CPU 以主系统时钟运行的过程中执行 STOP 指令的情况		
		CPU 以高速内部振荡器时钟 (f_{IH}) 运行	CPU 以 X1 时钟 (f_X) 运行	CPU 以外部主系统时钟 (f_{EX}) 运行
系统时钟		停止给 CPU 提供时钟。		
	主系统时钟	f_{IH}	停止	
		f_X		
		f_{EX}		
	副系统时钟	f_{XT}	保持设定为 STOP 模式前的状态。	
		f_{EXS}		
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。 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。		
实时时钟 (RTC)		能运行。		
12 位间隔定时器		参照“第 12 章 看门狗定时器”。		
看门狗定时器				
定时器 RJ		- 在选择没有 TRJIO 输入滤波器时的事件计数模式中, 能运行。 - 在选择副系统时钟作为计数源并且 OSMC 寄存器的 RTCLPC 位为“0”时, 能运行。 - 在选择低速内部振荡器作为计数源时, 能运行。 - 上述以外: 禁止运行。		
定时器 RD		禁止运行。		
时钟输出 / 蜂鸣器输出		在选择副系统时钟作为计数时钟并且 RTCLPC 位为“0”时, 能运行 (否则禁止运行)。		
A/D 转换器		能进行唤醒 (转移到 SNOOZE 模式)。		
串行阵列单元 (SAU)		只有 CSIp 和 UARTq 才能唤醒运行 (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。		
数据传送控制器 (DTC)		能接受 DTC 启动源 (转移到 SNOOZE 模式)。		
事件链接控制器 (ELC)		能在可运行的功能块之间进行链接。		
上电复位功能		能运行。		
电压检测功能				
外部中断				
键中断功能				
CRC 运算 功能	高速 CRC	停止运行。		
	通用 CRC			
非法存储器存取检测功能		停止运行。		
RAM 奇偶校验功能				
RAM 保护功能				
SFR 保护功能				

- 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 |
| f_{XT} : XT1 时钟 | f_{EXS} : 外部副系统时钟 |
2. $p=00, q=0$

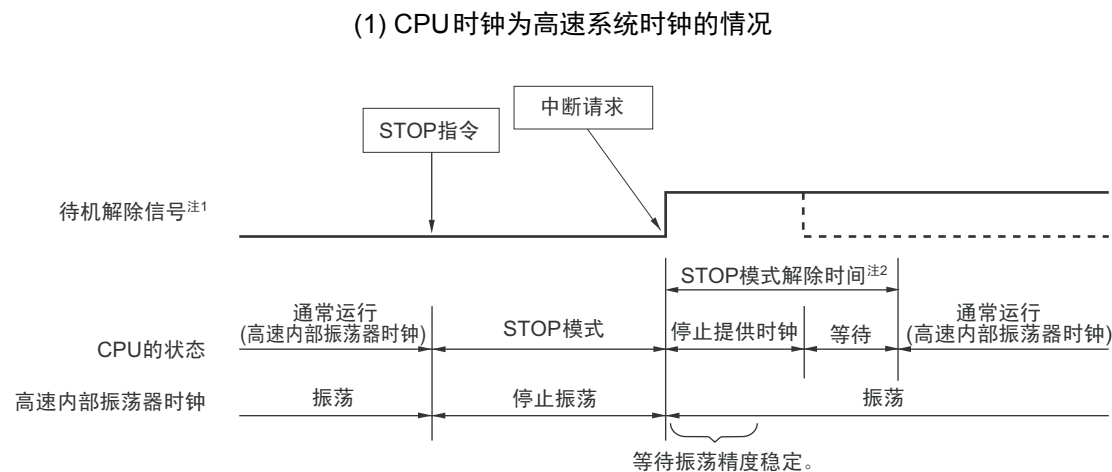
(2) STOP 模式的解除

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 20-3 通过中断请求解除 STOP 模式 (1/2)



注 1. 有关待机解除信号的详细内容，请参照“图 18-1 中断功能的基本结构”。

2. STOP 模式解除时间

停止提供时钟：

FRQSEL4=0: 18 μ s ~ 65 μ s

FRQSEL4=1: 18 μ s ~ 135 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

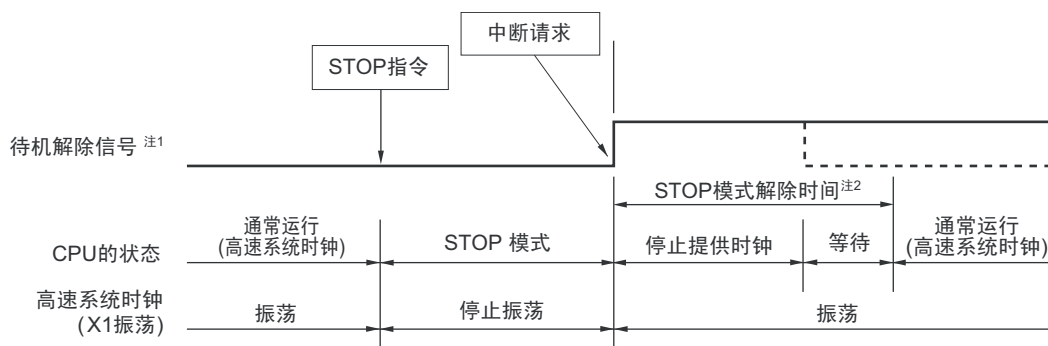
注意 在 CPU 以高速系统时钟（X1 振荡）运行并且要缩短 STOP 模式解除后的振荡稳定时间时，必须在执行 STOP 指令前暂时将 CPU 时钟切换为高速内部振荡器时钟。

备注 1. 时钟停止提供的时间因温度条件和 STOP 模式期间而变。

2. 虚线表示接受解除了待机模式的中断请求的情况。

图 20-3 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 18-1 中断功能的基本结构”。

2. STOP 模式解除时间

停止提供时钟:

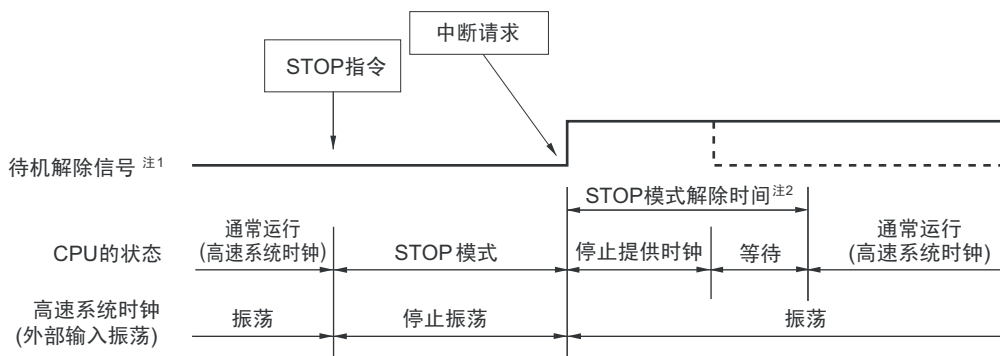
FRQSEL4=0: 18 μ s ~ “65 μ s 和振荡稳定时间 (通过 OSTC 进行设定) 中较长的时间”

FRQSEL4=1: 18 μ s ~ “135 μ s 和振荡稳定时间 (通过 OSTC 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时钟为高速内部振荡器时钟 (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 18-1 中断功能的基本结构”。

2. STOP 模式解除时间

停止提供时钟:

FRQSEL4=0: 18 μ s ~ 65 μ s

FRQSEL4=1: 18 μ s ~ 135 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

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。

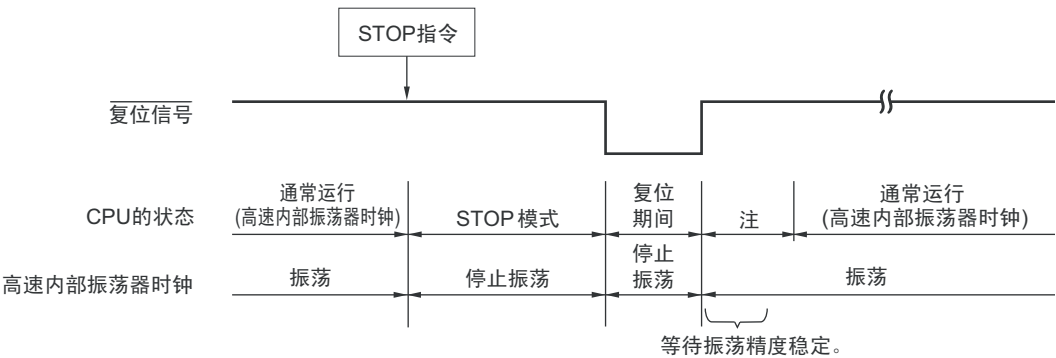
2. 虚线表示接受解除了待机模式的中断请求的情况。

(b) 通过产生复位信号进行的解除

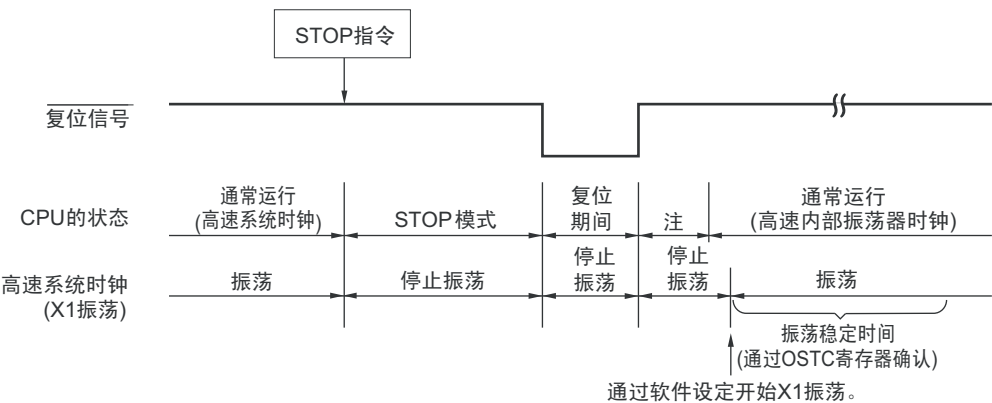
通过产生复位信号来解除 STOP 模式。然后，和通常的复位一样，在转移到复位向量地址后执行程序。

图 20-4 通过复位解除 STOP 模式

(1) CPU时钟为高速系统时钟的情况



(2) CPU时钟为高速内部振荡器时钟的情况



注 有关复位处理时间，请参照“第21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22章 上电复位电路”。

20.3.3 SNOOZE 模式

(1) SNOOZE 模式的设定和运行状态

能给 CSIp、A/D 转换器或者 DTC 设定 SNOOZE 模式，UARTq 只有在选项字节 000C2H 的 FRQSEL4 位为“0”时才能设定 SNOOZE 模式，并且只有在设定前的 CPU 时钟为高速内部振荡器时钟的情况下才能进行设定。

要在 SNOOZE 模式中使用 CSIp 或者 UARTq 时，必须在转移到 STOP 模式前设定串行待机控制寄存器 m (SSCm)。详细内容请参照“14.3 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，必须在转移到 STOP 模式前设定 A/D 转换器模式寄存器 2 (ADM2)。详细内容请参照“13.3 控制 A/D 转换器的寄存器”。

要在 SNOOZE 模式中使用 DTC 传送时，必须在转移到 STOP 模式前允许所用 DTC 的启动源。在 STOP 模式中，如果检测到允许的 DTC 启动源，就自动转移到 SNOOZE 模式。详细内容请参照“16.3 控制 DTC 的寄存器”。

备注 p=00, q=0, m=0

在进行 SNOOZE 模式的转移时，只在以下的时间内为等待状态。

STOP 模式 → SNOOZE 模式的转移时间：

FRQSEL4=0: 18μs ~ 65μs

FRQSEL4=1: 18μs ~ 135μs

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 20-3 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以及发生 DTC 启动源的情况
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}	
副系统时钟	f_{XT}	保持 STOP 模式中的状态。
	f_{EXS}	
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。 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 DTC 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。
实时时钟 (RTC)		能运行。
12 位间隔定时器		
看门狗定时器		参照“第 12 章 看门狗定时器”。
定时器 RJ		- 在选择没有 TRJIO 输入滤波器时的事件计数模式中, 能运行。 - 在选择副系统时钟作为计数源并且 OSMC 寄存器的 RTCLPC 位为“0”时, 能运行。 - 在选择低速内部振荡器作为计数源时, 能运行。 - 上述以外: 禁止运行。
定时器 RD		禁止运行。
时钟输出 / 蜂鸣器输出		在选择副系统时钟作为计数时钟并且 RTCLPC 位为“0”时, 能运行 (否则禁止运行)。
A/D 转换器		能运行。
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。
数据传送控制器 (DTC)		能运行。
事件链接控制器 (ELC)		能在可运行的功能块之间进行链接。
上电复位功能		能运行。
电压检测功能		
外部中断		
键中断功能		
CRC 运算功能	高速 CRC	停止运行。
	通用 CRC	禁止运行。
非法存储器存取检测功能		在执行 DTC 时, 能运行。
RAM 奇偶校验功能		
RAM 保护功能		
SFR 保护功能		

- 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_X : X1 时钟

f_{XT} : XT1 时钟

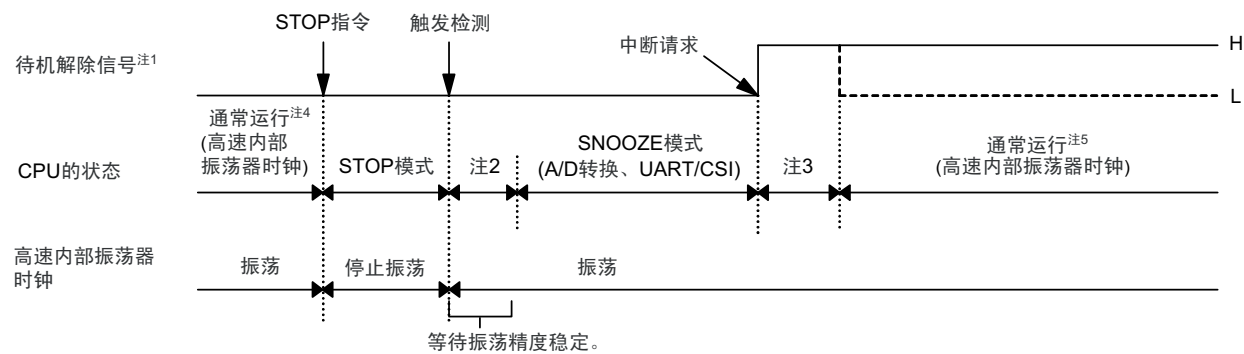
f_{IL} : 低速内部振荡器时钟

f_{EX} : 外部主系统时钟

f_{EXS} : 外部副系统时钟
2. $p=00, q=0$

(2) 在 SNOOZE 模式中产生中断请求信号时的时序图

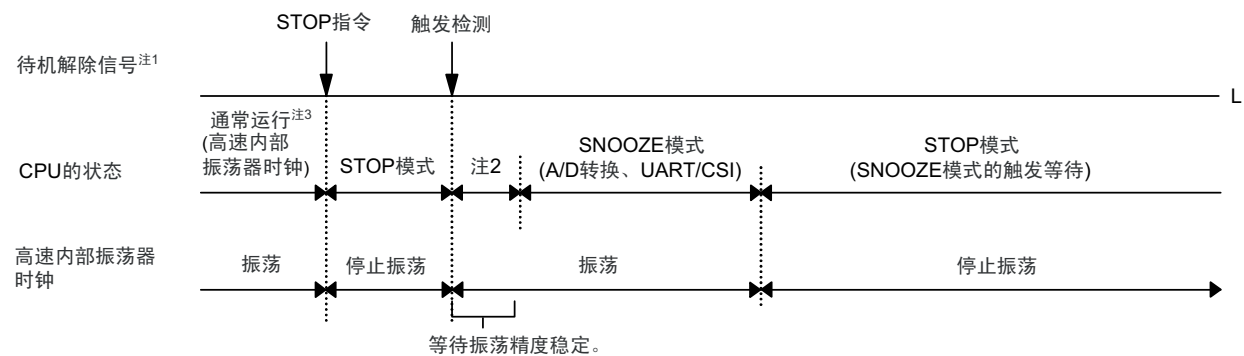
图 20-5 在 SNOOZE 模式中产生中断请求的情况



- 注 1. 有关待机解除信号的详细内容，请参照图 18-1。
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 20-6 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 18-1。
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 13 章 A/D 转换器”和“第 14 章 串行阵列单元”。

第 21 章 复位功能

以下 7 种方法产生复位信号。

- (1) 通过 $\overline{\text{RESET}}$ 引脚输入外部复位。
- (2) 通过看门狗定时器的程序失控检测产生内部复位。
- (3) 通过上电复位 (POR) 电路的电源电压和检测电压的比较产生内部复位。
- (4) 通过电压检测电路 (LVD) 的电源电压和检测电压的比较产生内部复位。
- (5) 因执行非法指令而产生内部复位注。
- (6) 因 RAM 奇偶校验错误而产生内部复位。
- (7) 因存取非法存储器而产生内部复位。

内部复位和外部复位相同，在产生复位信号后，从写在地址 0000H 和 0001H 中的地址开始执行程序。

当给 $\overline{\text{RESET}}$ 引脚输入低电平，或者看门狗定时器检测到程序失控，或者检测到 POR 电路和 LVD 电路的电压，或者执行非法指令注，或者发生 RAM 奇偶校验错误，或者存取非法存储器时，产生复位并且各硬件变为如表 21-1 所示的状态。

注 当执行指令码 FFH 时，发生非法指令错误。

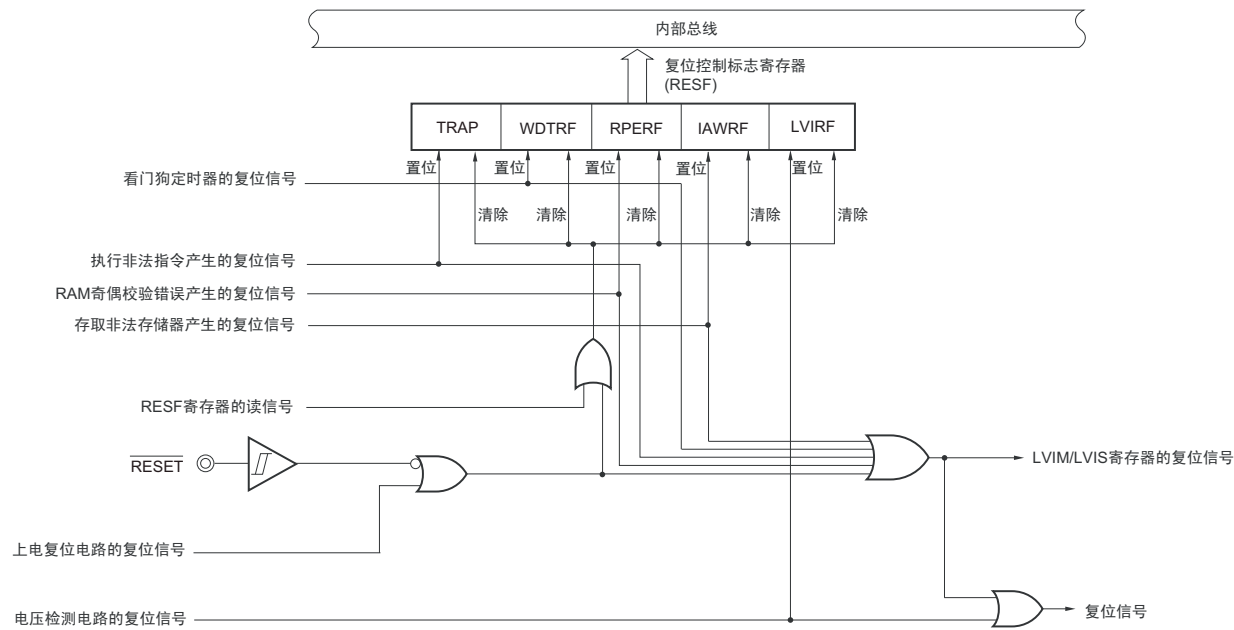
在通过在线仿真器或者片上调试仿真器进行仿真时，不会因执行非法指令而产生复位。

注意 1. 要进行外部复位时，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“31.4 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、XT1 时钟、高速内部振荡器时钟和低速内部振荡器时钟的振荡。外部主系统时钟和外部副系统时钟的输入无效。
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）。
 - P130：在复位期间以及接受复位后输出低电平。
 - P40、P130 以外的端口：在复位期间以及接受复位后为高阻抗。

图 21-1 复位功能的框图



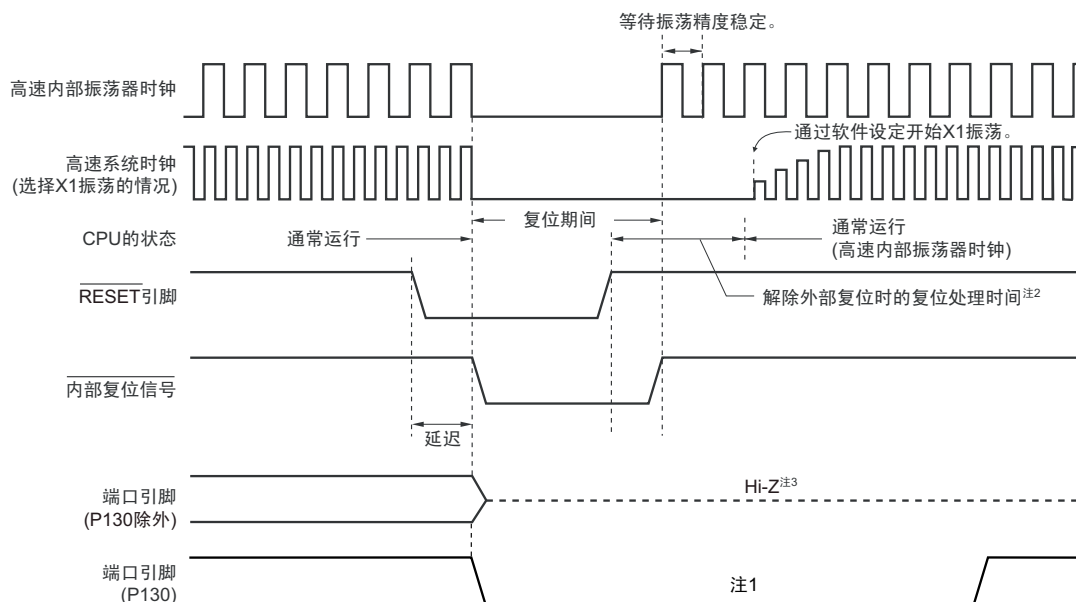
注意 LVD 电路的内部复位不会对 LVD 电路进行复位。

- 备注 1. LVIM：电压检测寄存器
2. LVIS：电压检测电平寄存器

21.1 复位时序

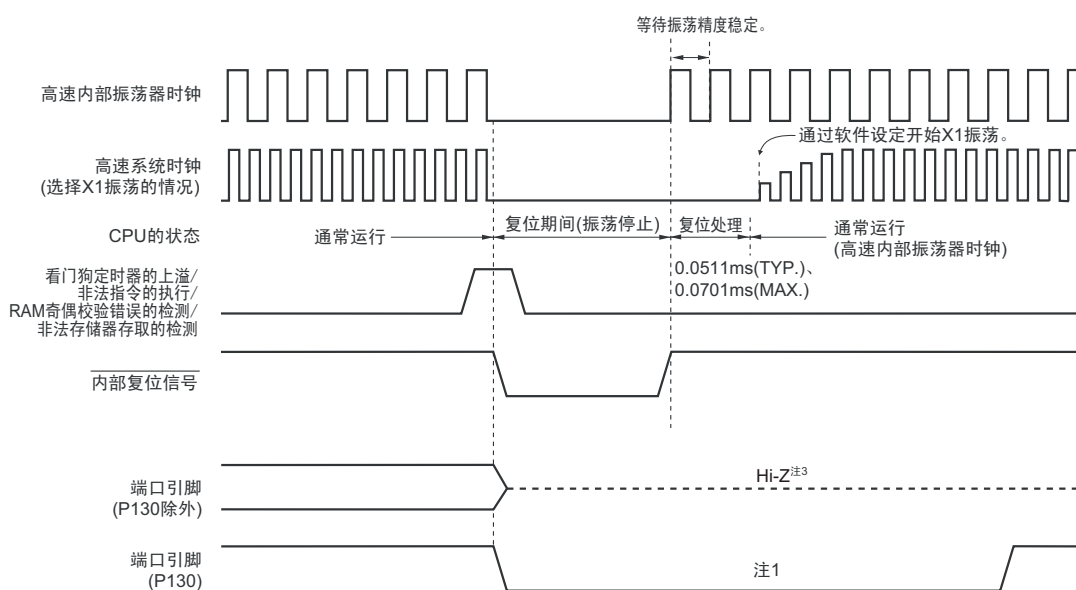
当给 $\overline{\text{RESET}}$ 引脚输入低电平时，产生复位。然后，如果给 $\overline{\text{RESET}}$ 引脚输入高电平就解除复位状态，并且在复位处理结束后以高速内部振荡器时钟开始执行程序。

图 21-2 $\overline{\text{RESET}}$ 输入的复位时序



对于因看门狗定时器的上溢、非法指令的执行、RAM 奇偶校验错误的检测或者非法存储器存取的检测而产生的复位，自动解除复位状态，在复位处理结束后以高速内部振荡器时钟开始执行程序。

图 21-3 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. 如果发生复位，P130 就输出低电平。因此，如果在发生复位前将 P130 设定为高电平输出，就能将 P130 的输出作为外部器件的复位信号进行虚拟输出。要解除外部器件的复位信号时，必须通过软件将 P130 设定为高电平输出。
2. 解除外部复位时的复位时间：
- | | |
|------------------|--|
| 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处理时间之前需要电压稳定等待时间 0.99ms(TYP.)、2.30ms(MAX.)。
3. 端口引脚 P40 变为以下状态：
- 在外部复位或者 POR 复位的期间为高阻抗。
 - 在其他复位期间以及接受复位后为高电平（连接内部上拉电阻）。

注意 在发生内部复位时同样也会对看门狗定时器进行复位。

对于由 POR 电路和 LVD 电路的电压检测产生的复位，如果在复位后满足 $V_{DD} \geq V_{POR}$ 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 22 章 上电复位电路”和“第 23 章 电压检测电路”。

备注 V_{POR} : POR 电源电压上升检测电压
 V_{LVD} : LVD 检测电压

表 21-1 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}	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）。
	副系统时钟	f _{XT}	停止运行（XT1 引脚和 XT2 引脚处于输入端口模式）。
		f _{EXS}	时钟输入无效（引脚处于输入端口模式）。
f _{IL}		停止运行。	
CPU			
代码闪存			停止运行。
数据闪存			停止运行。
RAM			停止运行。
端口（锁存器）			高阻抗注
定时器阵列单元			停止运行。
定时器 RJ			
定时器 RD			
实时时钟（RTC）			
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）			
数据传送控制器（DTC）			
上电复位功能			
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。
键中断功能			
CRC 运算功能	高速 CRC		
	通用 CRC		
非法存储器存取检测功能			
RAM 奇偶校验功能			
RAM 保护功能			
SFR 保护功能			

注 端口引脚 P40、P130 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）。
- P130：在复位期间输出低电平。

备注 f_{IH} : 高速内部振荡器时钟
 f_{EX} : 外部主系统时钟
 f_{EXS} : 外部副系统时钟

f_X : X1 振荡时钟
 f_{XT} : XT1 振荡时钟
 f_{IL} : 低速内部振荡器时钟

表 21-2 接受复位后的各硬件状态

硬件		接受复位后的状态 ^{注 1}
程序计数器 (PC)		设定复位向量表 (0000H、0001H) 的内容。
堆栈指针 (SP)		不定
程序状态字 (PSW)		06H
RAM	数据存储器	不定
	通用寄存器	不定

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”。

21.2 确认复位源的寄存器

21.2.1 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 21-4 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 21-3。

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“24.3.3 RAM 奇偶校验错误检测功能”。

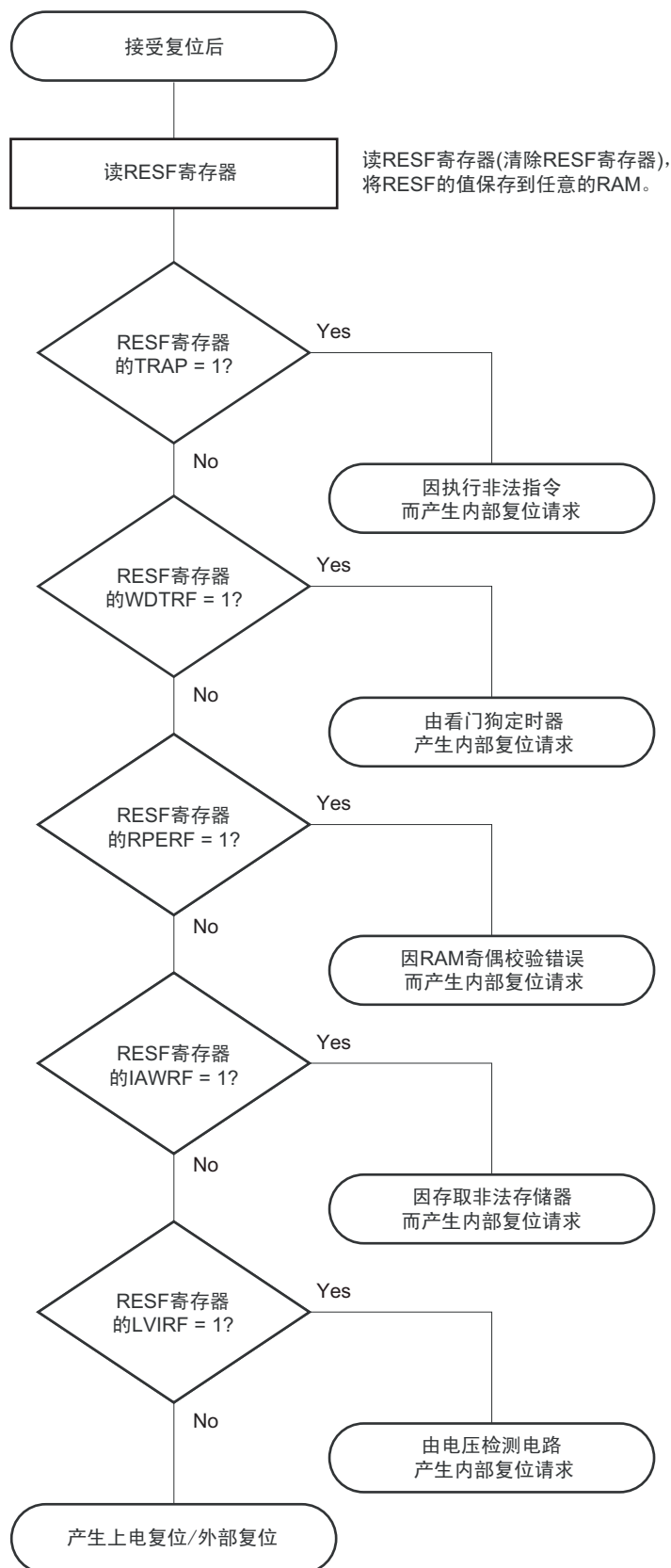
发生复位请求时的 RESF 寄存器状态如表 21-3 所示。

表 21-3 发生复位请求时的 RESF 寄存器状态

复位源 标志	$\overline{\text{RESET}}$ 输入	POR 产生 的复位	执行非法 指令产生 的复位	WDT 产生 的复位	RAM 奇偶 校验错误 产生的复位	存取非法 存储器产生 的复位	LVD 产生 的复位
TRAP	清“0”	清“0”	置“1”	保持	保持	保持	保持
WDTRF			保持	置“1”	置“1”	置“1”	保持
RPERF				保持			
IAWRF				保持	保持	保持	置“1”
LVIRF					保持	保持	

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 21-5 所示。

图 21-5 复位源的确认步骤



第 22 章 上电复位电路

22.1 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“31.4 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“31.4 AC 特性”所示的工作电压范围前，通过 STOP 模式的转移、电压检测电路或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

注意 当上电复位电路产生内部复位信号时，将复位控制标志寄存器（RESF）清“00H”。

备注 1. 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“00H”而将标志置“1”。有关 RESF 寄存器的详细内容，请参照“第 21 章 复位功能”。

2. V_{POR} : POR 电源电压上升检测电压

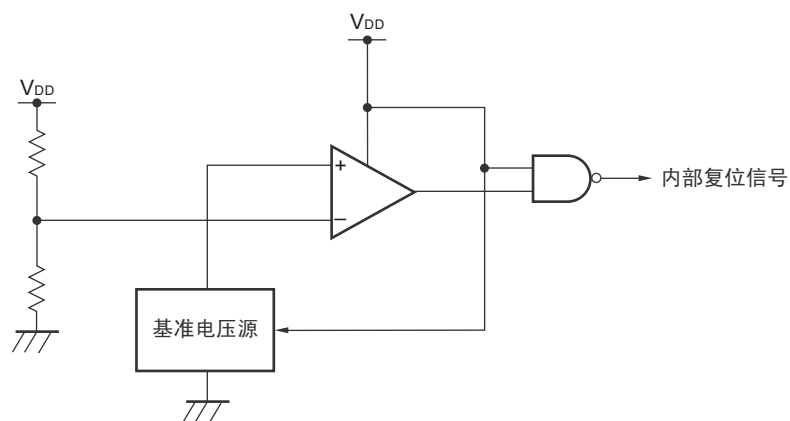
V_{PDR} : POR 电源电压下降检测电压

详细内容请参照“31.6.3 POR 电路特性”。

22.2 上电复位电路的结构

上电复位电路的框图如图 22-1 所示。

图 22-1 上电复位电路的框图

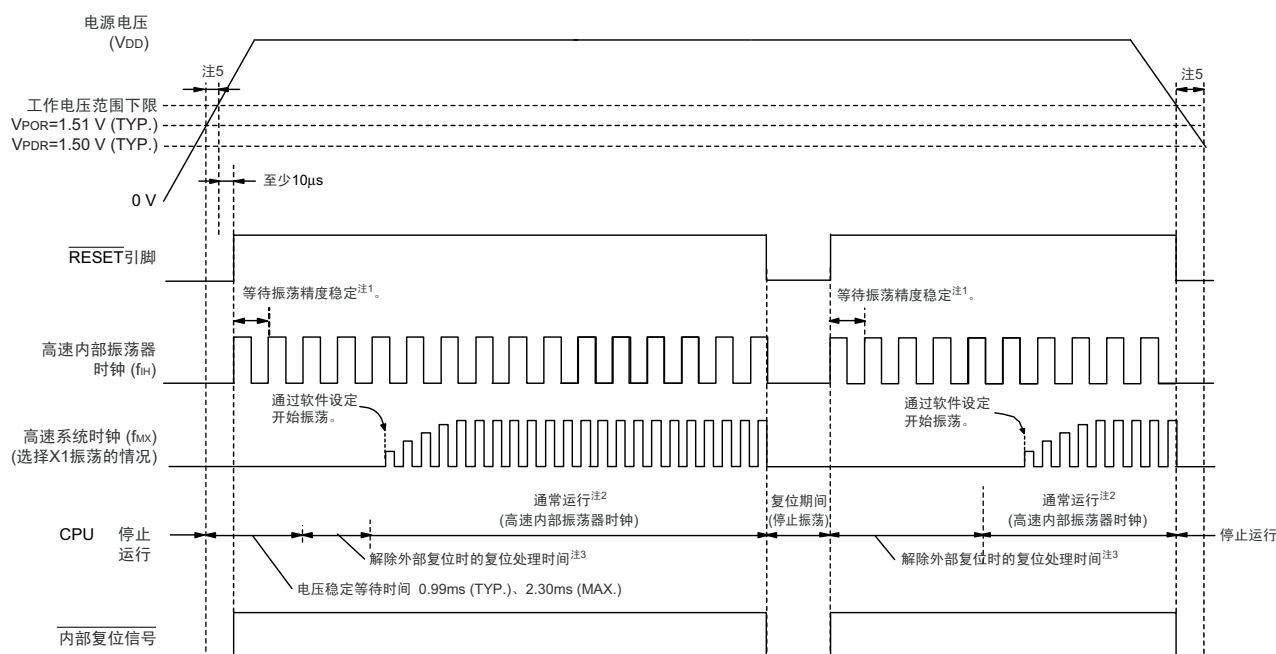


22.3 上电复位电路的运行

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 22-2 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 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. 能将 CPU 时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用 X1 时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换；在使用 XT1 时钟的情况下，必须在利用定时器功能等确认振荡稳定时间后进行切换。
3. 到开始通常运行为止的时间除了达到 V_{POR} （1.51V(TYP.)) 后的“电压稳定等待时间”以外，在将 $\overline{RESET}$ 信号置为高电平（“1”）后还需要以下的“解除外部复位时的复位处理时间（解除 POR 后的第 1 次）”。解除外部复位时的复位处理时间如下所示：
- 解除 POR 后的第 1 次 : 0.672ms(TYP.)、0.832ms(MAX.)（使用 LVD 的情况）
 0.399ms(TYP.)、0.519ms(MAX.)（不使用 LVD 的情况）
4. 解除 POR 后第 2 次以后的解除外部复位时的复位处理时间如下所示：
- 解除 POR 后的第 2 次以后 : 0.531ms(TYP.)、0.675ms(MAX.)（使用 LVD 的情况）
 0.259ms(TYP.)、0.362ms(MAX.)（不使用 LVD 的情况）
5. 当电源电压上升时，必须在电源电压达到“31.4 AC 特性”所示的工作电压范围前，通过外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过 STOP 模式的转移、电压检测电路或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

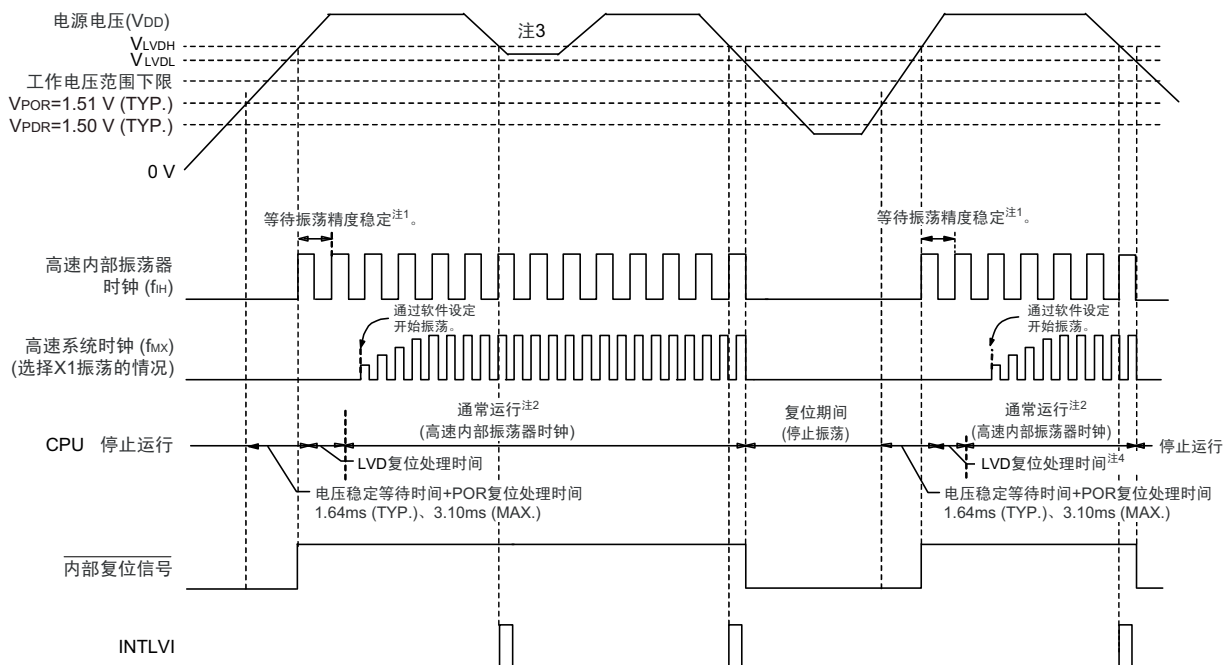
备注 V_{POR} : POR 电源电压上升检测电压

V_{PDR} : POR 电源电压下降检测电压

注意 在 LVD 为 OFF 时，必须使用 $\overline{RESET}$ 引脚的外部复位。详细内容请参照“第 23 章 电压检测电路”。

图 22-2 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)



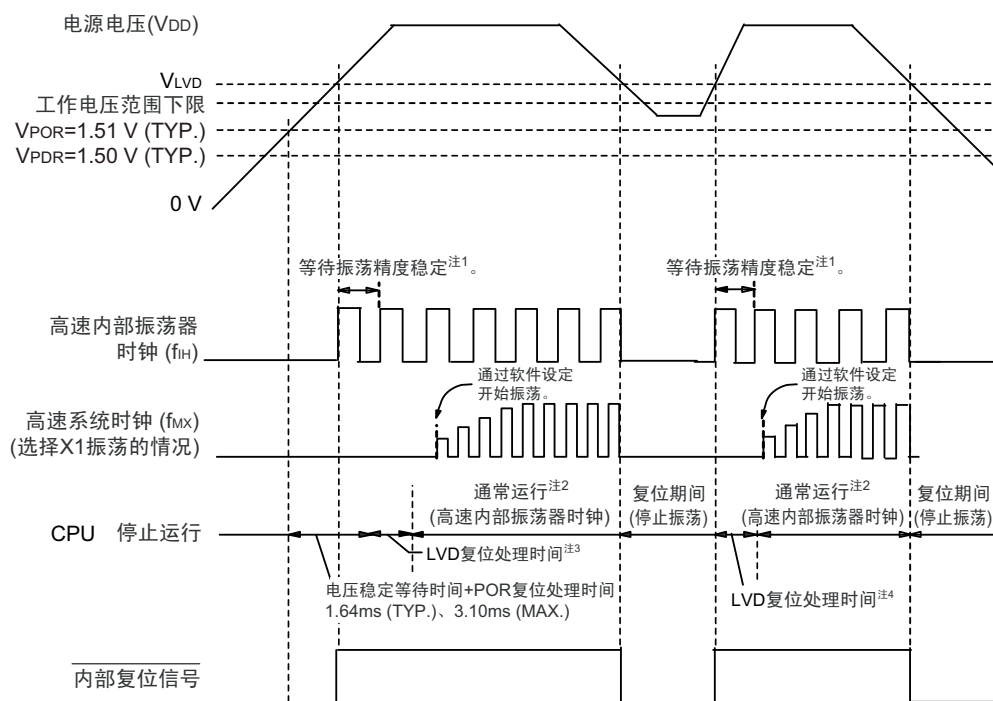
- 注 1. 内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。
2. 能将 CPU 时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用 X1 时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换；在使用 XT1 时钟的情况下，必须在利用定时器功能等确认振荡稳定时间后进行切换。
3. 在产生中断请求信号（INTLVI）后，自动将电压检测电平寄存器（LVIS）的 LVILV 位和 LVIMD 位置“1”。因此，必须考虑可能出现电源电压在不低于电压检测电压（ V_{LVDL} ）的状态下恢复到高电压检测电压（ V_{LVDH} ）或者更高的情况，在产生 INTLVI 后按照“图 23-7 工作电压的确认 / 复位的设定步骤”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 22-2 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. 能将 CPU 时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用 X1 时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换；在使用 XT1 时钟的情况下，必须在利用定时器功能等确认振荡稳定时间后进行切换。
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 22-2 (3) LVD 复位模式的情况”的“注 3”的时间相同。

第 23 章 电压检测电路

23.1 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} ）能通过选项字节选择14种检测电平（参照“第26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“31.4 AC特性”所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过STOP模式的转移、电压检测电路或者外部复位，置为复位状态。工作电压范围取决于用户选项字节（000C2H/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} ）用于产生中断或者解除复位。

在各模式中，产生以下的中断信号和内部复位信号。

中断 & 复位模式 (LVIMDS1、LVIMDS0=1、0)	复位模式 (LVIMDS1、LVIMDS0=1、1)	中断模式 (LVIMDS1、LVIMDS0=0、1)
在工作电压下降时，当检测到 $V_{DD} < V_{LVDH}$ 时，产生中断请求信号；当检测到 $V_{DD} < V_{LVDL}$ 时，产生内部复位；当检测到 $V_{DD} \geq V_{LVDH}$ 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）。

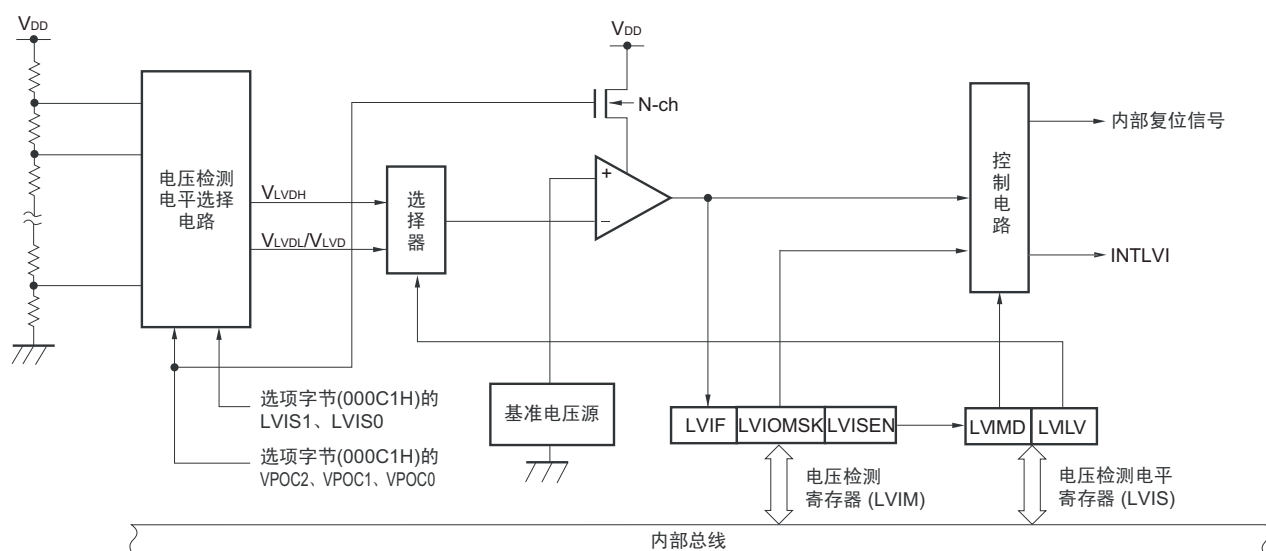
在电压检测电路运行时，能通过读电压检测标志（LVIF：电压检测寄存器（LVIM）的 bit0）来确认电源电压是大于等于检测电压还是小于检测电压。

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 21 章 复位功能”。

23.2 电压检测电路的结构

电压检测电路的框图如图 23-1 所示。

图 23-1 电压检测电路的框图



23.3 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）

23.3.1 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 23-2 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} ）≥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

23.3.2 电压检测电平选择寄存器（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 23-3 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 注 2

0

0

0

0

0

0

0

LVILV 注 2

LVIMD 注 2

电压检测的运行模式

0

中断模式

1

复位模式

LVILV 注 2

LVD 检测电平

0

高电压检测电平 (V_{LVDH})

1

低电压检测电平 (V_{LVDL} 或者 V_{LVD})

- 注 1. 复位值因复位源和选项字节的设定而变。
在发生 LVD 复位时，不将此寄存器清“00H”。
在发生 LVD 以外的复位时，此寄存器的值如下：
- 选项字节的 LVIMDS1、LVIMDS0=1、0 时：00H
 - 选项字节的 LVIMDS1、LVIMDS0=1、1 时：81H
 - 选项字节的 LVIMDS1、LVIMDS0=0、1 时：01H
2. 只有在选择中断 & 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 23-7 的步骤进行。
2. 通过选项字节 000C1H 选择 LVD 的运行模式和各模式的检测电压 (V_{LVDH}、V_{LVDL}、V_{LVD})。用户选项字节 (000C1H/010C1H) 的格式如表 23-1 所示。有关选项字节的详细内容，请参照“第 26 章 选项字节”。

表 23-1 用户选项字节（000C1H/010C1H）的格式 (1/2)

地址：000C1H/010C1H 注

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD 的设定（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
2.92V	2.86V	2.75V	0	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
2.81V	2.75V	0	1	1	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 23 章 电压检测电路”。

2. 检测电压是 TYP. 值。详细内容请参照“31.6.4 LVD 电路特性”。

表 23-1 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
2.81V	2.75V	0	1	1	1	1	0	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	×	×	×	×	×
—		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“31.4 AC 特性”所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过 STOP 模式的转移、电压检测电路或者外部复位，置为复位状态。

工作电压范围取决于用户选项字节（000C2H/010C2H）的设定。

备注 1. ×：忽略

- 有关 LVD 电路的详细内容，请参照“第 23 章 电压检测电路”。
- 检测电压是 TYP. 值。详细内容请参照“31.6.4 LVD 电路特性”。

23.4 电压检测电路的运行

23.4.1 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} ）。

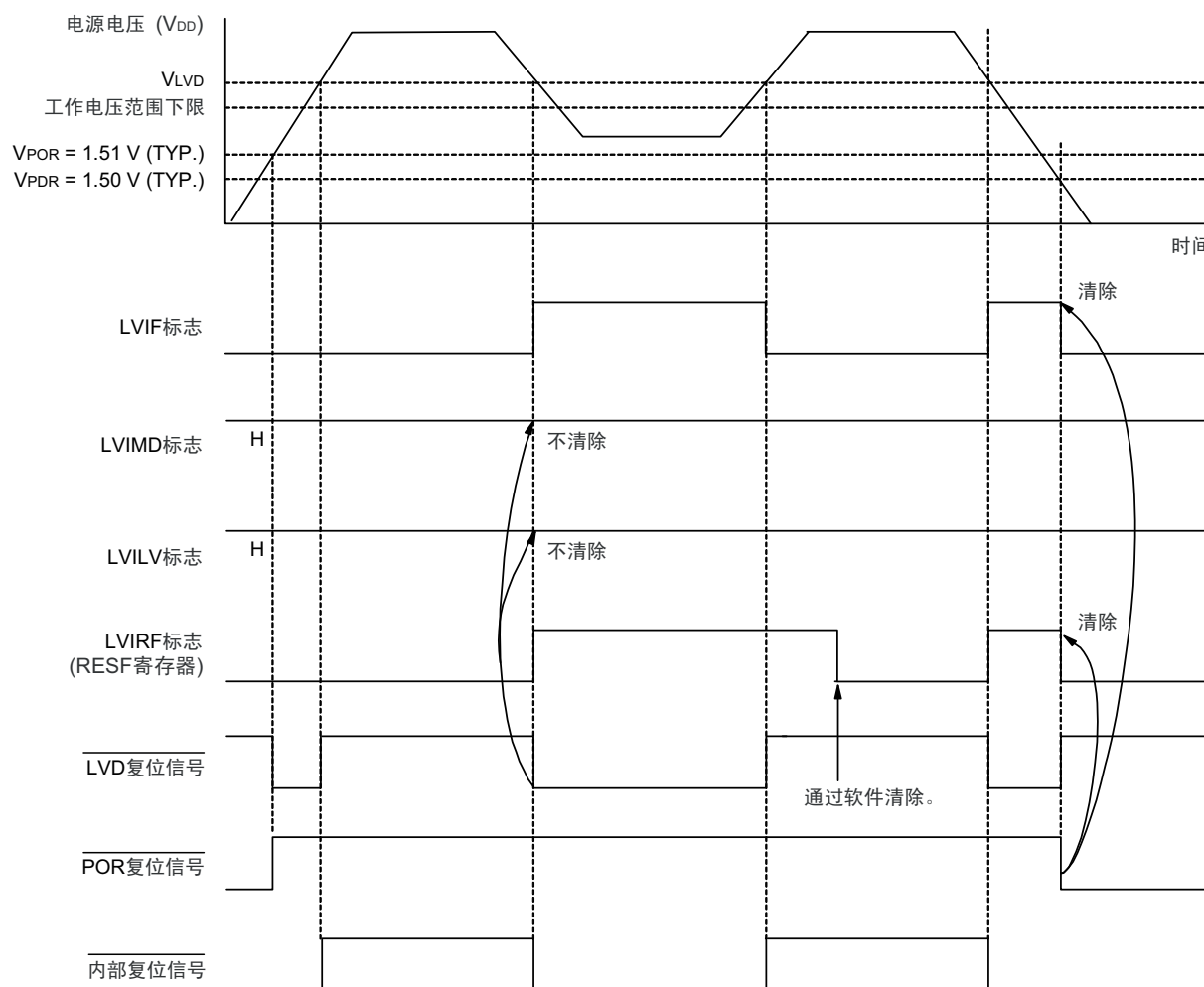
● LVD 复位模式的运行

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。

当工作电压下降时，如果电源电压（ V_{DD} ）低于电压检测电平（ V_{LVD} ），就产生 LVD 的内部复位。

LVD 复位模式的内部复位信号的发生时序如图 23-4 所示。

图 23-4 内部复位信号的发生时序（选项字节的 LVIMDS1、LVIMDS0=1、1）



备注 V_{POR} : POR 电源电压上升检测电压

V_{PDR} : POR 电源电压下降检测电压

23.4.2 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 LVD 的内部复位。

在解除 LVD 的内部复位后，如果电源电压（ V_{DD} ）超过电压检测电平（ V_{LVD} ），就产生 LVD 的中断请求信号（INTLVI）。当工作电压下降时，必须在工作电压低于“31.4 AC 特性”所示的工作电压范围前，通过 STOP 模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

LVD 中断模式的中断请求信号的发生时序如图 23-5 所示。

23.4.3 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

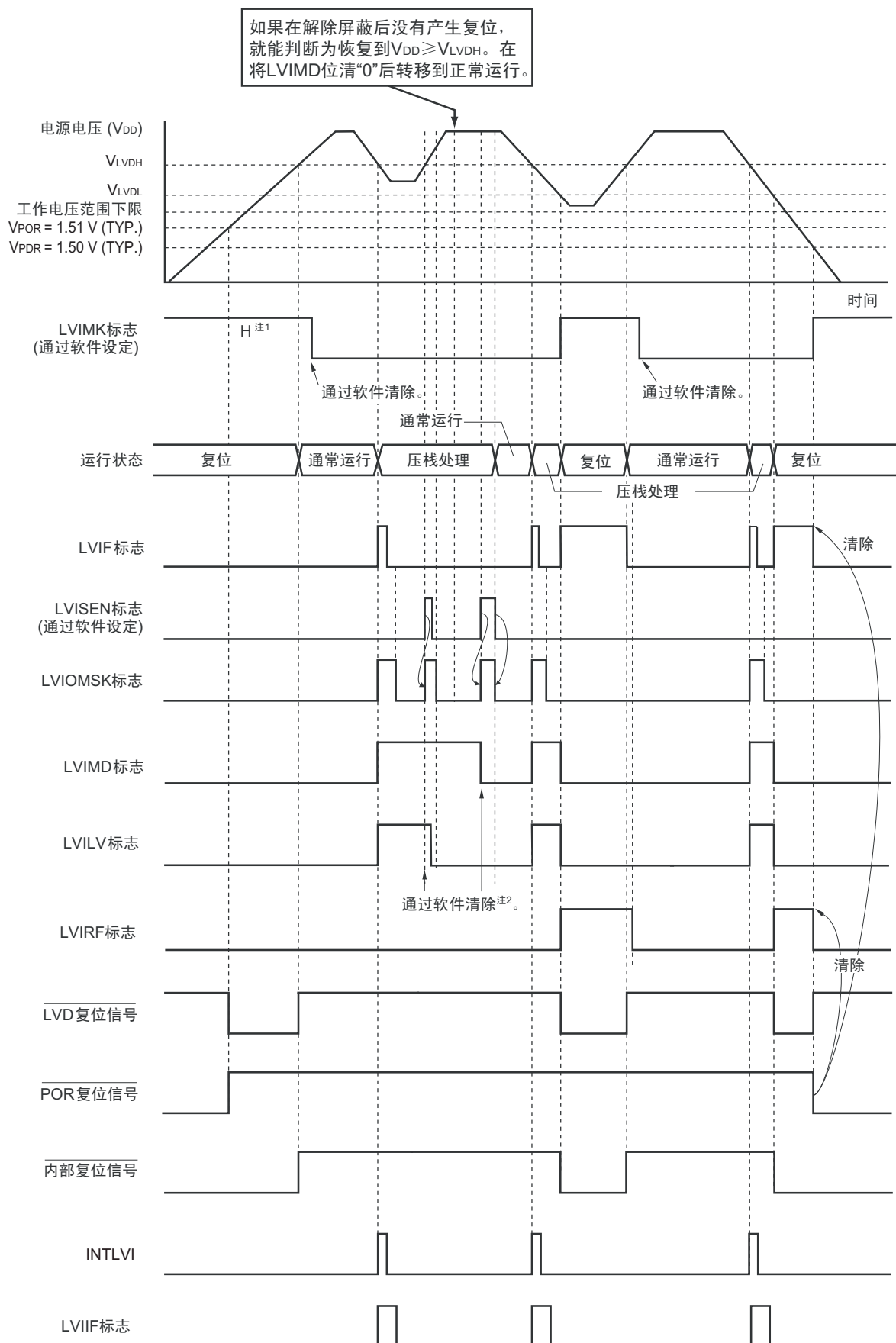
当接通电源时，中断 & 复位模式（选项字节的 LVIMDS1、LVIMDS0=1、0）在电源电压（ V_{DD} ）超过高电压检测电平（ V_{LVDH} ）前保持 LVD 的内部复位状态。如果电源电压（ V_{DD} ）超过高电压检测电平（ V_{LVDH} ），就解除内部复位。

当工作电压下降时，如果电源电压（ V_{DD} ）低于高电压检测电平（ V_{LVDH} ），就产生 LVD 的中断请求信号（INTLVI）并且能进行任意的压栈处理。此后，如果电源电压（ V_{DD} ）低于低电压检测电平（ V_{LVDL} ），就产生 LVD 的内部复位。但是，在发生 INTLVI 后，即使电源电压在不低于低电压检测电压（ V_{LVDL} ）的状态下恢复到高电压检测电压（ V_{LVDH} ）或者更高，也不产生中断请求信号。

当使用 LVD 中断 & 复位模式时，必须按照“图 23-7 工作电压的确认/复位的设定步骤”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发生时序如图 23-6 所示。

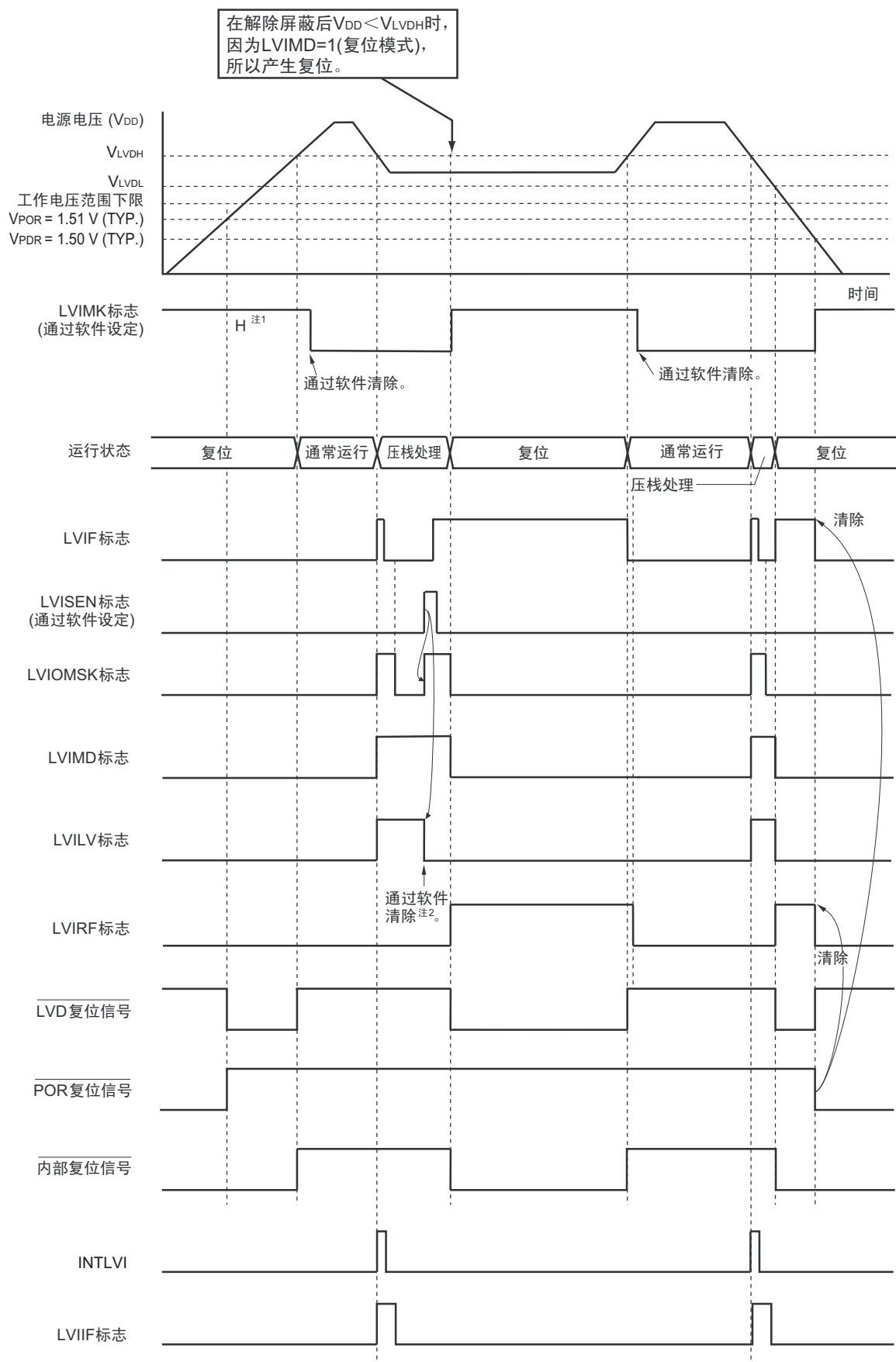
图 23-6 复位 & 中断信号的发生时序 (选项字节的 LVIMDS1、LVIMDS0=1、0) (1/2)



- 注 1. 在产生复位信号后，LVIMK 标志变为“1”。
2. 当使用中断 & 复位模式时，必须在发生中断后按照“图 23-7 工作电压的确认 / 复位的设定步骤”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

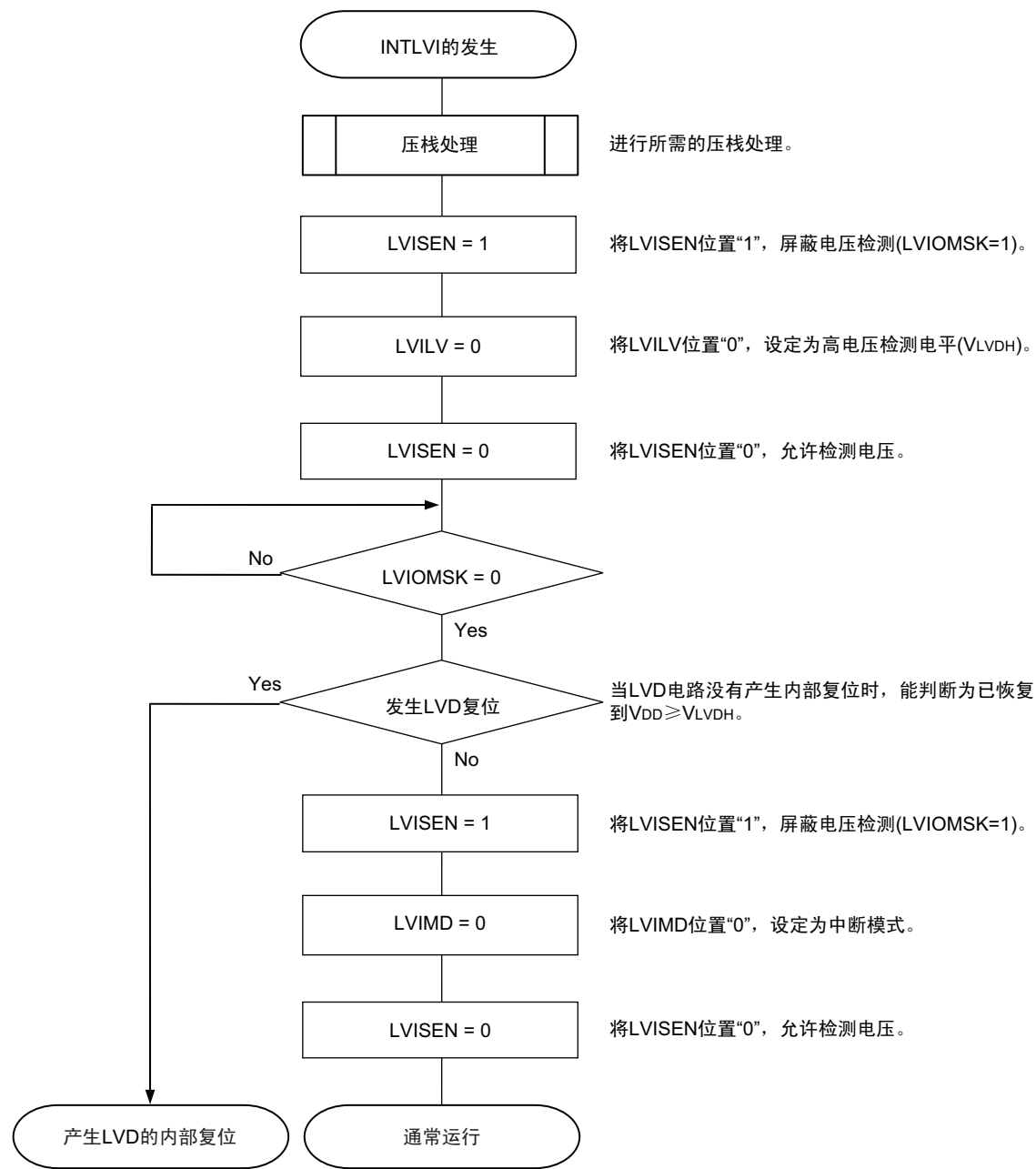
图 23-6 中断 & 复位信号的发生时序 (选项字节的 LVIMDS1、LVIMDS0=1、0) (2/2)



- 注 1. 在产生复位信号后，LVIMK 标志变为“1”。
2. 当使用中断 & 复位模式时，必须在发生中断后按照“图 23-7 工作电压的确认 / 复位的设定步骤”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 23-7 工作电压的确认 / 复位的设定步骤



23.5 电压检测电路的注意事项

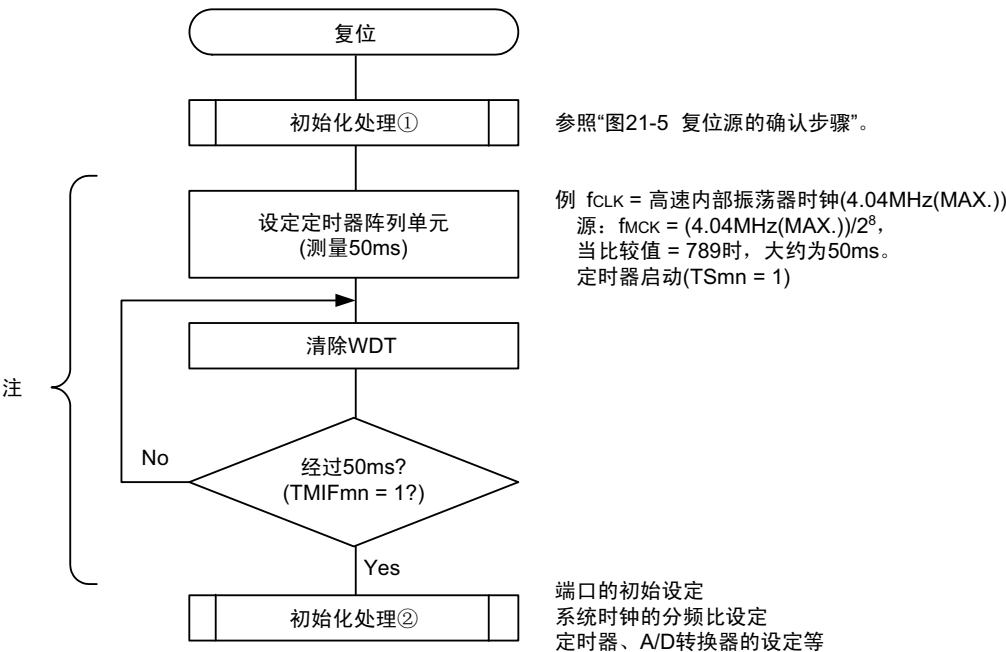
(1) 有关接通电源时的电压波动

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 23-8 LVD 检测电压附近的电源电压波动不超过 50ms 时的软件处理例子



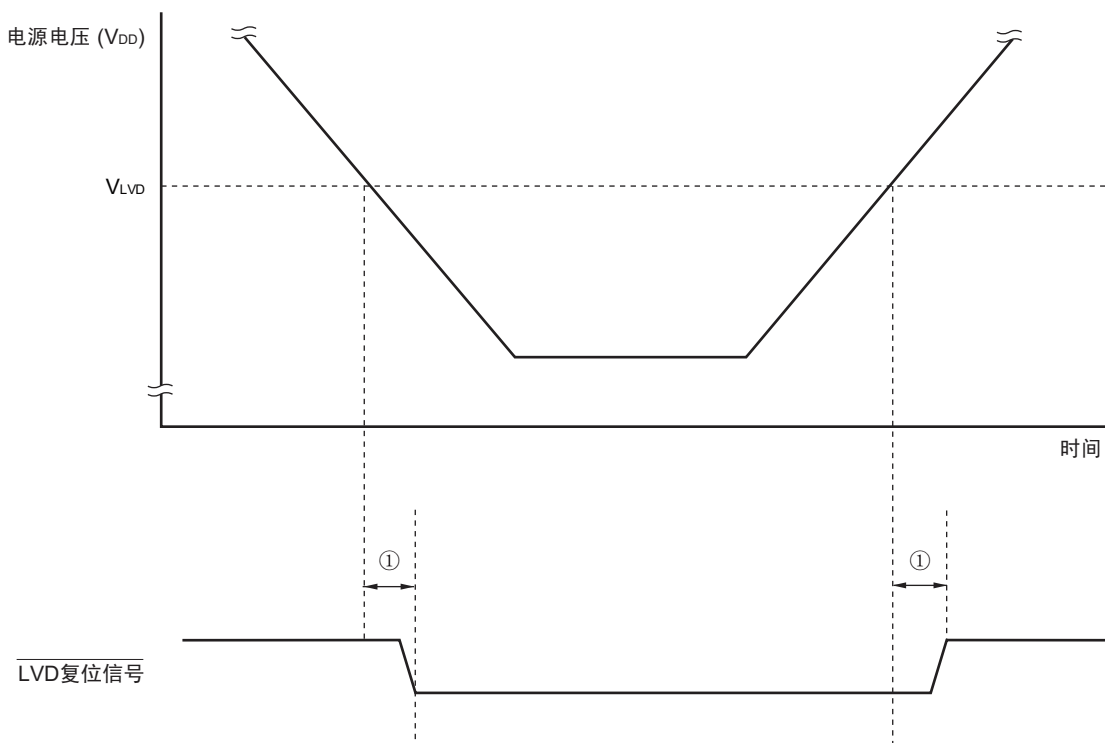
注 如果在此期间再次发生复位，就不转移到初始化处理②。

备注 m=0, n=0 ~ 3

(2) 从产生 LVD 复位源到产生或者解除 LVD 复位的延迟

从满足电源电压 (V_{DD}) < LVD 检测电压 (V_{LVD}) 到产生 LVD 复位为止会发生延迟。同样, 从 LVD 检测电压 (V_{LVD}) ≤ 电源电压 (V_{DD}) 到解除 LVD 复位为止也会发生延迟 (参照图 23-9)。

图 23-9 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 (300μs (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 $\overline{\text{RESET}}$ 引脚的外部复位。

要进行外部复位时, 必须至少给 $\overline{\text{RESET}}$ 引脚输入 10μs 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“31.4 AC 特性”所示的工作电压范围内至少保持 10μs 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 中断模式时工作电压下降的情况

在将 LVD 置为 OFF 并且设定为 LVD 中断模式的情况下, 如果工作电压下降, 就必须在工作电压低于“31.4 AC 特性”所示的工作电压范围前, 通过 STOP 模式的转移或者外部复位, 置为复位状态。在重新开始运行时, 必须确认电源电压是否恢复到工作电压范围。

第 24 章 安全功能

24.1 安全功能的概要

为了对应 IEC60730 和 IEC61508 安全标准，R7F0C014B2D、R7F0C014L2D 内置以下安全功能。此安全功能的目的是通过单片机的自诊断，在检测到故障时安全地停止工作。

(1) 闪存 CRC 运算功能（高速 CRC、通用 CRC）

通过 CRC 运算检测闪存的数据错误。

能根据不同的用途和使用条件，分别使用以下 2 个 CRC。

- “高速 CRC”...在初始化程序中，能停止 CPU 的运行并且高速检查整个代码闪存区。
- “通用 CRC”...在 CPU 运行中，不限于代码闪存区而能用于多用途的检查。

(2) RAM 奇偶校验错误检测功能

在读 RAM 数据时，检测奇偶校验错误。

(3) RAM 保护功能

防止因 CPU 失控而改写 RAM 数据。

(4) SFR 保护功能

防止因 CPU 失控而改写 SFR。

(5) 非法存储器存取检测功能

检测对非法存储区（没有存储器的区域或者限制存取的区域）的非法存取。

(6) 频率检测功能

能使用定时器阵列单元进行 CPU/ 外围硬件时钟频率的自检。

(7) A/D 测试功能

能通过 A/D 转换器的正（+）基准电压、负（-）基准电压、模拟输入通道（ANI）、温度传感器输出和内部基准电压输出的 A/D 转换来进行 A/D 转换器的自检。

(8) 输入 / 输出端口的数字输出信号电平检测功能

在输入 / 输出端口为输出模式时，能读引脚的输出电平。

备注 有关对应 IEC60730 和 IEC61508 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）。

24.2 安全功能使用的寄存器

安全功能的各功能使用以下寄存器。

寄存器名	安全功能的各功能
- 闪存 CRC 控制寄存器 (CRC0CTL) - 闪存 CRC 运算结果寄存器 (PGCRCL)	闪存 CRC 运算功能 (高速 CRC)
- CRC 输入寄存器 (CRCIN) - CRC 数据寄存器 (CRCD)	CRC 运算功能 (通用 CRC)
RAM 奇偶校验错误控制寄存器 (RPECTL)	RAM 奇偶校验错误检测功能
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
定时器输入选择寄存器 0 (TIS0)	频率检测功能
A/D 测试寄存器 (ADTES)	A/D 测试功能
端口模式选择寄存器 (PMS)	输入 / 输出引脚的数字输出信号电平检测功能

有关各寄存器的内容，在“24.3 安全功能的运行”中进行说明。

24.3 安全功能的运行

24.3.1 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（例如，64KB 闪存：512μs@32MHz）。

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。

24.3.1.1 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 24-1 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)

0

0

0

0

1

1

00000H ~ 0FFFBH (64K-4 字节)

0

0

0

1

0

0

00000H ~ 13FFBH (80K-4 字节)

0

0

0

1

0

1

00000H ~ 17FFBH (96K-4 字节)

0

0

0

1

1

0

00000H ~ 1BFFBH (112K-4 字节)

0

0

0

1

1

1

00000H ~ 1FFFBH (128K-4 字节)

上述以外

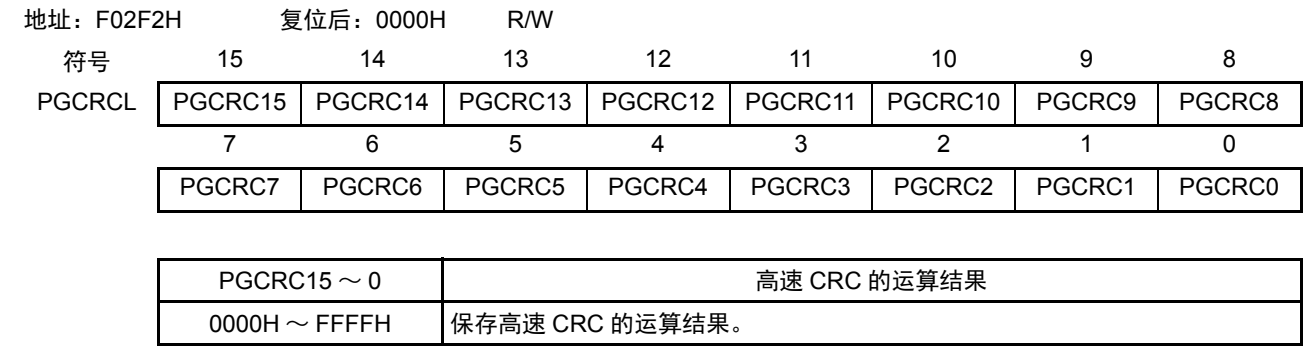
禁止设定。

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。

24.3.1.2 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 24-2 闪存 CRC 运算结果寄存器（PGCRCL）的格式

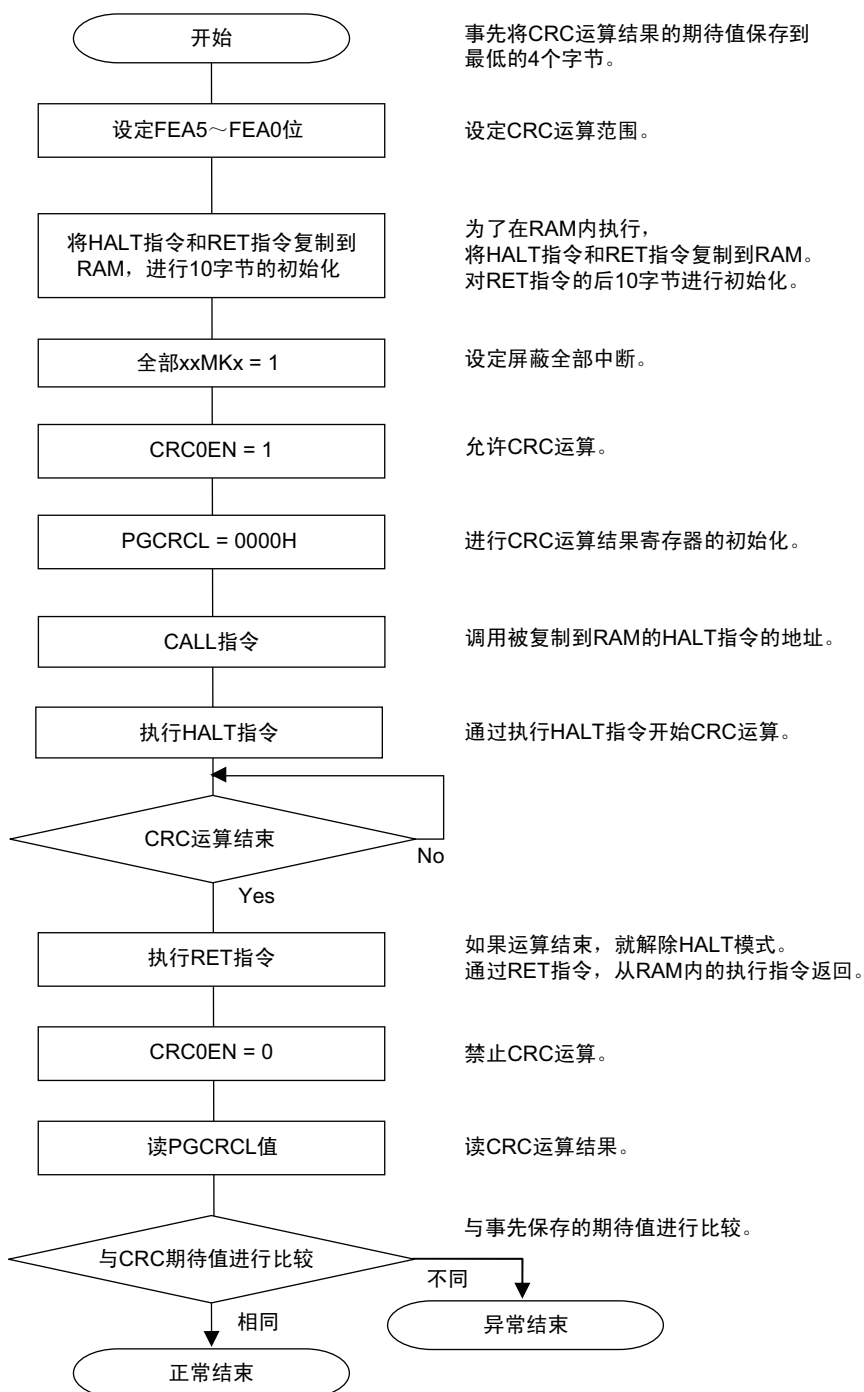


注意 只有在 CRC0EN（CRC0CTL 寄存器的 bit7）位为“1”时才能写 PGCRCL 寄存器。

闪存 CRC 运算功能（高速 CRC）的流程图如图 24-3 所示。

<操作流程>

图 24-3 闪存 CRC 运算功能（高速 CRC）的流程图



注意 1. 只以代码闪存为 CRC 运算的对象。

2. 必须将 CRC 运算的期待值保存在代码闪存中的运算范围后的区域。

3. 通过在 RAM 区执行 HALT 指令，CRC 运算变为有效。

必须在 RAM 区执行 HALT 指令。

CRC 运算的期待值能使用综合开发环境“CubeSuite+”进行计算。有关详细内容，请参照《CubeSuite+ Development Environment User's Manual》。

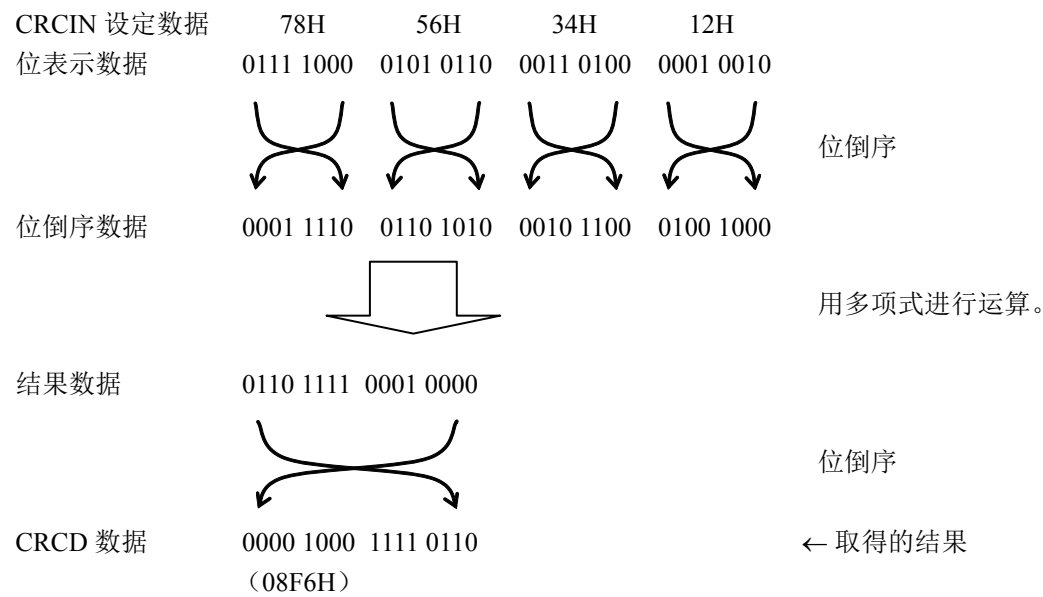
24.3.2 CRC 运算功能（通用 CRC）

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 DTC 传送过程中使用。

在主系统时钟运行模式或者副系统时钟运行模式中，都能使用 CRC 运算功能。

CRC 生成多项式使用 CRC-16-CCITT 的“ $X^{16}+X^{12}+X^5+1$ ”。因为考虑到是以 LSB 优先进行的通信，所以在将输入数据的位序颠倒后进行计算。例如，从 LSB 发送数据“12345678H”的情况，按照“78H”、“56H”、“34H”、“12H”的顺序给 CRCIN 寄存器写值，从 CRCD 寄存器得到“08F6H”的值。这是针对颠倒了数据“12345678H”的位序后的以下位序进行 CRC 运算的结果。



注意 在执行程序的过程中，因为调式程序将软件断点的设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。

24.3.2.1 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”。

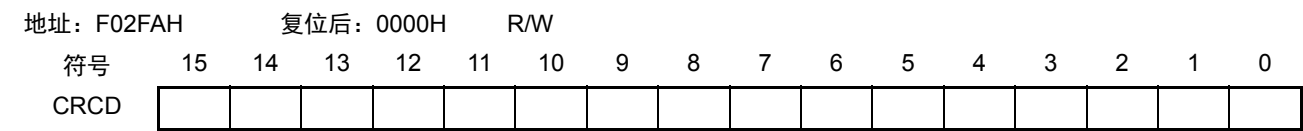
图 24-4 CRC 输入寄存器（CRCIN）的格式

地址：FFFACH	复位后：00H	R/W						
符号	7	6	5	4	3	2	1	0
CRCIN								
bit7 ~ 0		功能						
00H ~ FFH		数据输入						

24.3.2.2 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 寄存器开始经过 1 个 CPU/ 外围硬件时钟 (f_{CLK}) 后，将 CRC 运算结果保存到 CRCD 寄存器。
通过 16 位存储器操作指令设定 CRCD 寄存器。
在产生复位信号后，此寄存器的值变为“0000H”。

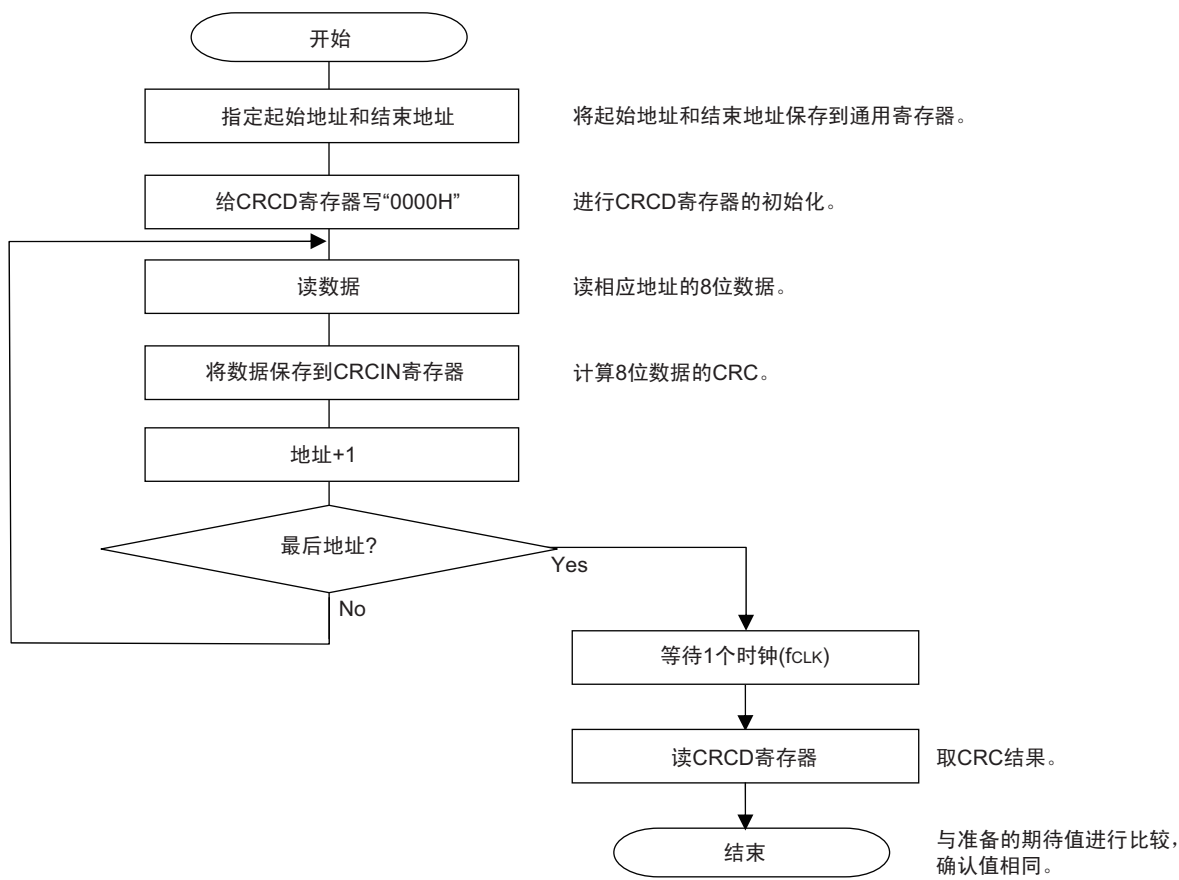
图 24-5 CRC 数据寄存器 (CRCD) 的格式



- 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 24-6 CRC 运算功能 (通用 CRC) 的流程图



24.3.3 RAM 奇偶校验错误检测功能

IEC60730 标准要求确认 RAM 数据。因此，R7F0C014B2D、R7F0C014L2D 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。

24.3.3.1 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”。

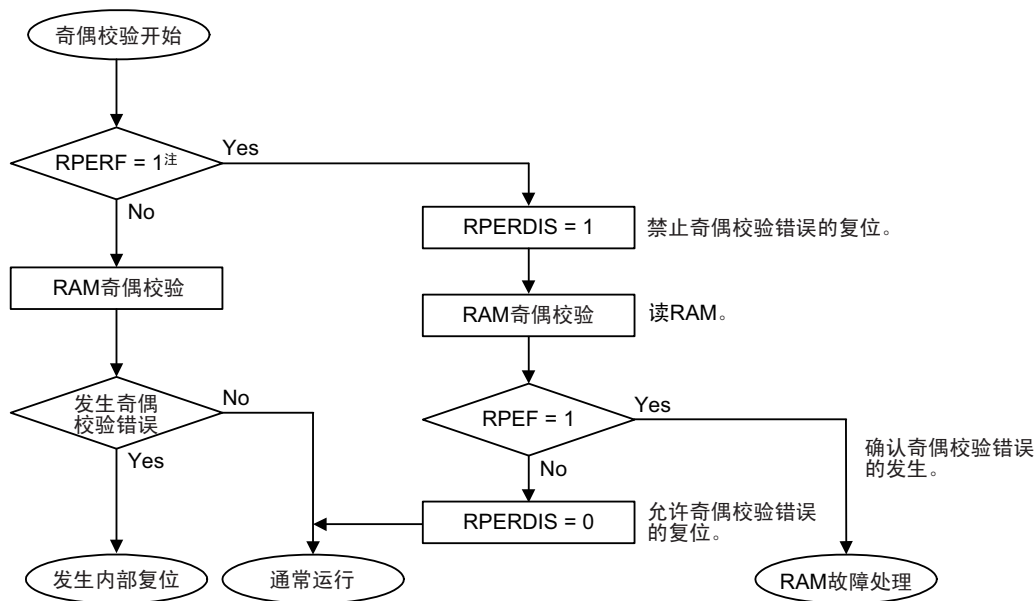
图 24-7 RAM 奇偶校验错误控制寄存器（RPECTL）的格式

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，必须在执行 RAM 区的指令时对“所用 RAM 区+10 字节”的区域进行初始化。

- 备注 1. 初始状态为允许产生奇偶校验错误复位（RPERDIS=0）。
2. 即使设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。
3. 因 RAM 奇偶校验错误而将 RPECTL 寄存器的 RPEF 标志置“1”，通过写“0”或者全部复位源将 RPEF 标志清“0”。当 RPEF 标志为“1”时，即使读未发生奇偶校验错误的 RAM，RPEF 标志也保持“1”的状态。
4. RAM 奇偶校验检测的范围不包括通用寄存器。

图 24-8 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 21 章 复位功能”。

24.3.4 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。

24.3.4.1 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 24-9 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。

24.3.5 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。

24.3.5.1 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 24-10 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、PIORx 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）。

24.3.6 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 24-11 中记载为“NG”的范围。

图 24-11 非法存取检测空间

		能否存取		
		读	写	取指令 (执行)
FFFFFH	特殊功能寄存器 (SFR) 256字节	OK	OK	NG
FFF00H FFEFFH				
FFEE0H FFEDFH	RAM 8K字节		OK	
FDF00H				
镜像区	NG		NG	
数据闪存			OK	OK
F1000H F0FFFH				
保留区			OK	NG
F0800H F07FFH				
扩展特殊功能寄存器 (2nd SFR) 2K字节			OK	OK
F0000H EFFFFH				
EF000H EEFFFH	保留区		NG	NG
1FFFFH	代码闪存 128字节	OK	OK	
00000H				

24.3.6.1 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 24-12 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。

24.3.7 频率检测功能

IEC60730 标准要求确认振荡频率是否正常。

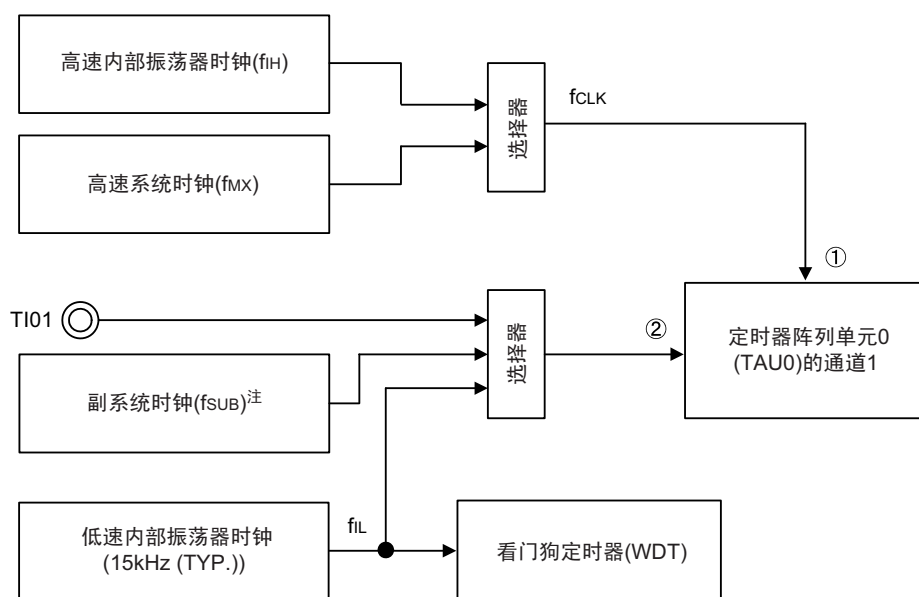
频率检测功能可使用 CPU/ 外围硬件的时钟频率 (f_{CLK})，并且通过测量定时器阵列单元 0 (TAU0) 的通道 1 输入脉冲，判断 2 个时钟的比率关系是否正确。

但是，如果某 1 个时钟或者 2 个时钟停止振荡，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1输入:
 - 通道1的定时器输入 (TI01)
 - 低速内部振荡器时钟 (f_{IL} : 15kHz (TYP.))
 - 副系统时钟 (f_{SUB}) 注

图 24-13 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”。

注 只有内置副系统时钟的产品才能选择。

24.3.7.1 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 0 和通道 1 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 24-14 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	TIS04	0	TIS02	TIS01	TIS00

TIS04	通道 0 使用的定时器输入的选择
0	定时器输入引脚（TI00）的输入信号
1	ELC 的事件输入信号

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

24.3.8 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)、温度传感器的输出电压和内部基准电压进行 A/D 转换, 确认 A/D 转换器是否正常运行。有关详细的确认方法, 请参照《安全功能 (A/D 测试) 的应用说明》(R01AN0955)。

能通过以下步骤确认模拟多路转换器:

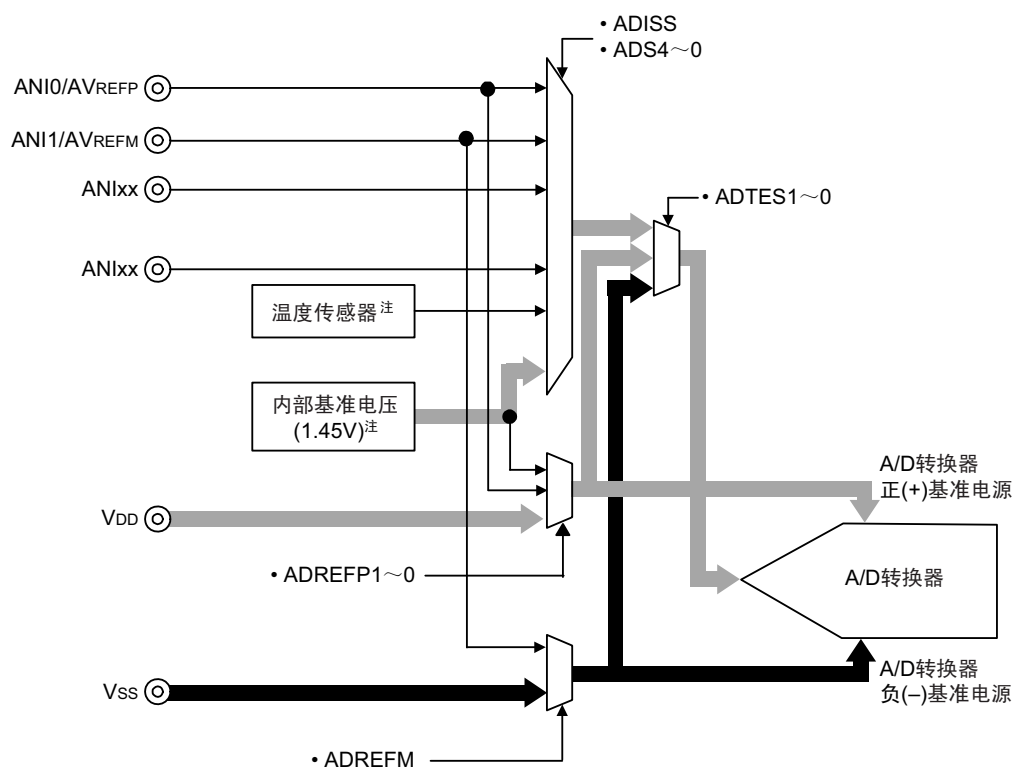
- ① 通过 ADTES 寄存器选择 ANIx 引脚作为 A/D 转换对象 (ADTES1、ADTES0=0、0)。
- ② 对 ANIx 引脚进行 A/D 转换 (转换结果 1-1)。
- ③ 通过 ADTES 寄存器选择 A/D 转换器的负 (-) 基准电压作为 A/D 转换对象 (ADTES1、ADTES0=1、0)。
- ④ 对 A/D 转换器的负 (-) 基准电压进行 A/D 转换 (转换结果 2-1)。
- ⑤ 通过 ADTES 寄存器选择 ANIx 引脚作为 A/D 转换对象 (ADTES1、ADTES0=0、0)。
- ⑥ 对 ANIx 引脚进行 A/D 转换 (转换结果 1-2)。
- 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 24-15 A/D 测试功能的结构



注 只有在 HS（高速主）模式中才能选择。

24.3.8.1 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（ANLxx）、温度传感器的输出电压和内部基准电压（1.45V）作为 A/D 转换对象。

当用作 A/D 测试功能时，进行以下的设定：

- 在测量零刻度时，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 24-16 A/D 测试寄存器（ADTES）的格式

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

ADTES1	ADTES0	A/D 转换对象
0	0	ANLxx/ 温度传感器输出注 / 内部基准电压输出（1.45V）注 （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）模式中才能选择温度传感器的输出电压和内部基准电压（1.45V）。

24.3.8.2 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 ANIxx、温度传感器输出或者内部基准电压 (1.45V) 时，必须将 A/D 测试寄存器 (ADTES) 置“00H”。

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 24-17 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	0	0	1	0	0	ANI4	P24/ANI4 引脚
0	0	0	1	0	1	ANI5	P25/ANI5 引脚
0	0	0	1	1	0	ANI6	P26/ANI6 引脚
0	0	0	1	1	1	ANI7	P27/ANI7 引脚
0	1	0	0	0	0	ANI16	P03/ANI16 引脚注 1
0	1	0	0	0	1	ANI17	P02/ANI17 引脚注 2
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	0	—	温度传感器输出注 3
1	0	0	0	0	1	—	内部基准电压输出 (1.45V) 注 3
上述以外						禁止设定。	

注 1. 在 32 引脚产品的情况下，为 P01/ANI16 引脚。

2. 在 32 引脚产品的情况下，为 P00/ANI17 引脚。

3. 只有在 HS (高速主) 模式中才能选择。

(注意在下一页)

注意 1. 必须将 bit5 和 bit6 位置“0”。

2. 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。
3. 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。
4. 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。
5. 要改写 ADISS 位时，必须在转换停止状态（ADCS=0、ADCE=0）下进行。
6. 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。
7. 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。
8. 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“13.7.4 选择温度传感器输出电压 / 内部基准电压的设定”。
9. 要转移到 STOP 模式时或者要在 CPU 以副系统时钟运行中转移到 HALT 模式时，不能将 ADISS 位置“1”。当 ADISS 位为“1”时，需要加上“31.3.2 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。

24.3.9 输入 / 输出引脚的数字输出信号电平检测功能

IEC60730 标准要求确认 I/O 功能是否正常。

输入 / 输出引脚的数字输出信号电平检测功能在引脚为输出模式时，能读引脚的数字输出电平。

24.3.9.1 端口模式选择寄存器（PMS）

此寄存器选择在引脚为输出模式（端口模式寄存器（PMm）的 PMmn 位为“0”）时是读端口的输出锁存器的值还是读端口的输出电平。

通过 1 位或者 8 位存储器操作指令设定 PMS 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 24-18 端口模式选择寄存器（PMS）的格式

地址: F007BH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0

PMS0	在引脚为输出模式时读数据的选择
0	读 Pmn 寄存器的值。
1	读引脚的数字输出电平。

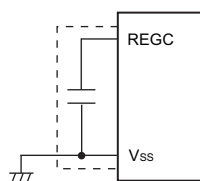
- 注意 1. 如果将 PMS 寄存器的 PMS0 位置“1”，就不能通过位操作指令改写端口寄存器（Pxx）。要改写端口寄存器（Pxx）时，必须使用 8 位存储器操作指令。
2. 对于使用定时器 RD 的脉冲输出强制截止功能而使引脚变为高阻抗状态的引脚，如果读引脚的数字输出电平，读取值就为“0”。

备注 m=0 ~ 7、12、14
 n=0 ~ 7

第 25 章 稳压器

25.1 稳压器的概要

R7F0C014B2D、R7F0C014L2D 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（ $0.47 \sim 1\mu\text{F}$ ）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 25-1 所示。

表 25-1 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
		在 CPU 以副系统时钟（ f_{SUB} ）运行的过程中，高速系统时钟（ f_{MX} ）和高速内部振荡器时钟（ f_{IH} ）都停止振荡。
	2.1V	在设定为 CPU 以副系统时钟（ f_{SUB} ）运行时的 HALT 模式中，高速系统时钟（ f_{MX} ）和高速内部振荡器时钟（ f_{IH} ）都停止振荡。
		上述以外（包含片上调试期间）注

注 在片上调试期间转移到副系统时钟运行模式或者 STOP 模式时，稳压器的输出电压保持 2.1V（不变为 1.8V）。

第 26 章 选项字节

26.1 选项字节的功能

R7F0C014B2D、R7F0C014L2D 的闪存地址 000C0H ~ 000C3H 为选项字节区。

选项字节由用户选项字节（000C0H ~ 000C2H）和片上调试选项字节（000C3H）构成。

在接通电源或者复位启动时，自动参照选项字节进行指定功能的设定。在使用本产品时，必须通过选项字节进行以下功能的设定。对于没有配置功能的位，不能更改初始值。

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。

26.1.1 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“31.4 AC 特性”所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过 STOP 模式的转移、电压检测电路或者外部复位，置为复位状态。

工作电压范围取决于用户选项字节（000C2H/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
 - 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1MHz～32MHz、48MHz、64MHz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。

26.1.2 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。

26.2 用户选项字节的格式

图 26-1 用户选项字节（000C0H/010C0H）的格式

地址：000C0H/010C0H 注 1

符号	7	6	5	4	3	2	1	0
	WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

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

WINDOW1	WINDOW0	看门狗定时器的窗口打开期间注 2
0	0	禁止设定。
0	1	50%
1	0	75%
1	1	100%

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

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间 ($f_{IL}=17.25\text{kHz}(\text{MAX.})$)
0	0	0	$2^6/f_{IL}$ (3.71ms)
0	0	1	$2^7/f_{IL}$ (7.42ms)
0	1	0	$2^8/f_{IL}$ (14.84ms)
0	1	1	$2^9/f_{IL}$ (29.68ms)
1	0	0	$2^{11}/f_{IL}$ (118.72ms)
1	0	1	$2^{13}/f_{IL}$ (474.90ms)
1	1	0	$2^{14}/f_{IL}$ (949.80ms)
1	1	1	$2^{16}/f_{IL}$ (3799.19ms)

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%。

备注 f_{IL} ：低速内部振荡器的时钟频率

图 26-2 用户选项字节（000C1H/010C1H）的格式 (1/4)

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
2.92V	2.86V	2.75V	0	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。						

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 必须给 bit4 写“1”。

- 备注 1. 有关 LVD 电路的详细内容，请参照“第 23 章 电压检测电路”。
2. 检测电压是 TYP. 值。详细内容请参照“31.6.4 LVD 电路特性”。

图 26-2 用户选项字节（000C1H/010C1H）的格式 (2/4)

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（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
2.81V	2.75V	0	1	1	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。

注意 必须给 bit4 写“1”。

- 备注 1. 有关 LVD 电路的详细内容，请参照“第 23 章 电压检测电路”。
2. 检测电压是 TYP. 值。详细内容请参照“31.6.4 LVD 电路特性”。

图 26-2 用户选项字节（000C1H/010C1H）的格式 (3/4)

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
2.81V	2.75V	0	1	1	1	1	0	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。

注意 必须给 bit4 写“1”。

- 备注 1. 有关 LVD 电路的详细内容，请参照“第 23 章 电压检测电路”。
2. 检测电压是 TYP. 值。详细内容请参照“31.6.4 LVD 电路特性”。

图 26-2 用户选项字节（000C1H/010C1H）的格式 (4/4)

地址：000C1H/010C1H 注

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD 为 OFF 时的设定（使用 $\overline{\text{RESET}}$ 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”。
2. 当电源电压上升时，必须在电源电压达到“31.4 AC 特性”所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过 STOP 模式的转移、电压检测电路或者外部复位，置为复位状态。
- 工作电压范围取决于用户选项字节（000C2H/010C2H）的设定

- 备注 1. ×：忽略
2. 有关 LVD 电路的详细内容，请参照“第 23 章 电压检测电路”。
3. 检测电压是 TYP. 值。详细内容请参照“31.6.4 LVD 电路特性”。

图 26-3 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	FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	闪存运行模式的设定		
			工作频率范围	工作电压范围
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 ～ 32MHz	2.7V ～ 5.5V
上述以外		禁止设定。		

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	高速内部振荡器的时钟频率	
					f _{HOCO}	f _{IH}
1	1	0	0	0	64MHz	32MHz
1	0	0	0	0	48MHz	24MHz
0	1	0	0	0	32MHz	32MHz
0	0	0	0	0	24MHz	24MHz
0	1	0	0	1	16MHz	16MHz
0	0	0	0	1	12MHz	12MHz
0	1	0	1	0	8MHz	8MHz
0	0	0	1	0	6MHz	6MHz
0	1	0	1	1	4MHz	4MHz
0	0	0	1	1	3MHz	3MHz
0	1	1	0	0	2MHz	2MHz
0	1	1	0	1	1MHz	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 写“1”。

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“31.4 AC 特性”。

26.3 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 26-4 片上调试选项字节（000C3H/010C3H）的格式

地址：000C3H/010C3H 注

符号	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）。

26.4 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	7AH	; V_{LVDL} 选择 2.75V。 ; V_{LVDH} 选择上升 2.92V、下降 2.86V。 ; 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。

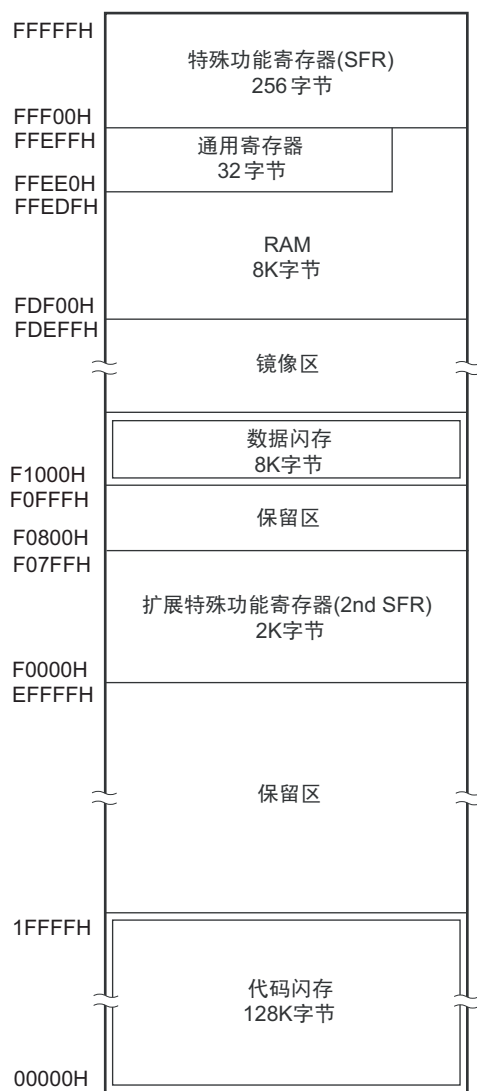
要在自编程过程中使用引导交换功能时，因为 000C0H ~ 000C3H 被 010C0H ~ 010C3H 替换，所以必须按照以下方式给 010C0H ~ 010C3H 设定和 000C0H ~ 000C3H 相同的值。

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		7AH	; V_{LVDL} 选择 2.75V。 ; V_{LVDH} 选择上升 2.92V、下降 2.86V。 ; 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 27 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。闪存有可执行程序的“代码闪存”和数据保存区的“数据闪存”。



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。

- 使用闪存编程器进行的串行编程（参照 27.1）
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）进行的串行编程（参照 27.2）
能通过和外部器件（单片机或者 ASIC）的 UART 通信进行板上或者板外编程。
- 自编程（参照 27.6）
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“27.8 数据闪存”。

27.1 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 Densei Machida Mfg. Co., Ltd 的产品。

表 27-1 R7F0C014B2D、R7F0C014L2D 和专用闪存编程器的连线表

专用闪存编程器的连接引脚				引脚名	引脚号
信号名		输入 / 输出	引脚功能		32 引脚
PG-FP5、FL-PR5	E1 片上调试仿真器				LQFP（7×7）
—	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
EMV _{DD}		—	TOOL0 引脚驱动电源	V _{DD}	8

专用闪存编程器的连接引脚				引脚名	引脚号
					64 引脚
信号名		输入 / 输出	引脚功能		LQFP （12×12）
PG-FP5、FL-PR5	E1 片上调试仿真器				LQFP （10×10）
—	TOOL0	输入 / 输出	发送 / 接收信号	TOOL0/P40	5
SI/RxD	—	输入 / 输出	发送 / 接收信号		
—	$\overline{\text{RESET}}$	输出	复位信号	$\overline{\text{RESET}}$	6
/RESET	—	输出			
V_{DD}		输入 / 输出	V_{DD} 电压生成 / 电源监视	V_{DD}	15
GND		—	接地	V_{SS}	13
				EV_{SS}	14
				REGC 注	12
EMV_{DD}		—	TOOL0 引脚驱动电源	V_{DD}	—
				EV_{DD}	16

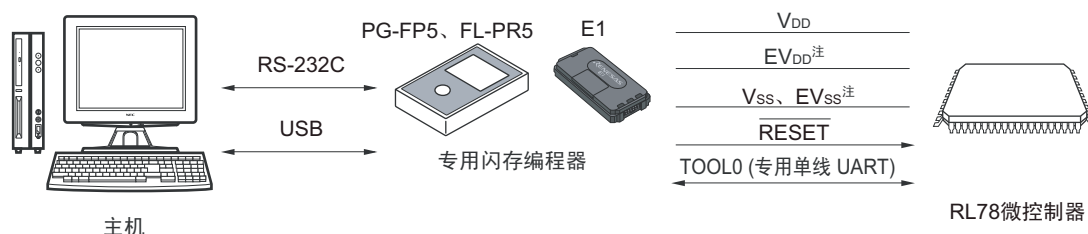
注 必须通过电容器 (0.47 ~ 1μF) 将 REGC 引脚接地。

备注 在使用闪存编程器进行编程时，此表中未记载的引脚可以开路。

27.1.1 编程环境

RL78 微控制器闪存的编程环境如下所示。

图 27-1 闪存的编程环境



注 只限于 64 引脚产品。

需要控制专用闪存编程器的主机。

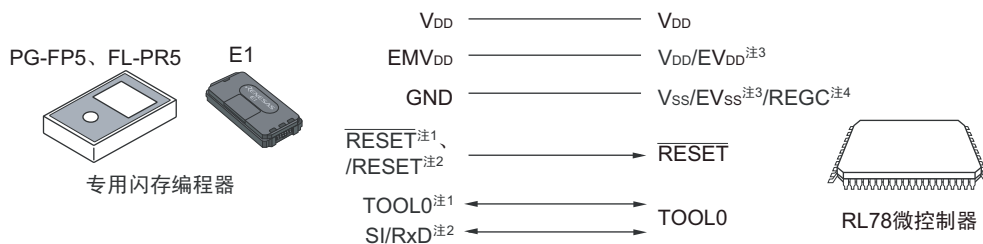
专用闪存编程器和 RL78 微控制器的接口使用 TOOL0 引脚，通过专用的单线 UART 进行编程和擦除。

27.1.2 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 27-2 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. 只限于 64 引脚产品。
 4. 必须通过电容器（0.47 ~ 1μF）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 27-2 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	引脚名 ^{注 2}
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
GND		—	接地
EMV_{DD}		—	TOOL0 引脚驱动电源
/RESET	—	输出	复位信号
—	$\overline{RESET}$	输出	
—	TOOL0	输入 / 输出	发送 / 接收信号
SI/RxD	—	输入 / 输出	

- 注 1. 必须通过电容器（0.47 ~ 1 μ F）将 REGC 引脚接地。
 2. 连接的目标引脚因产品而不同。详细内容请参照表 27-1。

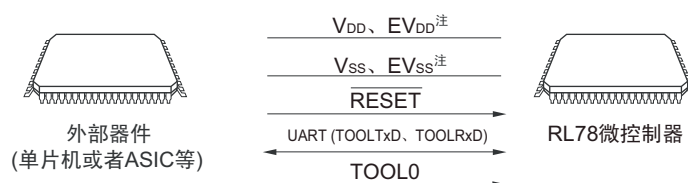
27.2 使用外部器件（内置 UART）的串行编程

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）。

27.2.1 编程环境

RL78 微控制器闪存的编程环境如下所示。

图 27-3 闪存的编程环境



注 只限于 64 引脚产品。

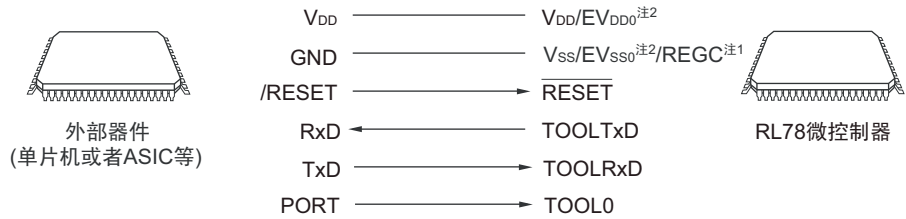
外部器件对 RL78 微控制器进行板上编程和擦除，但是不能进行板外编程。

27.2.2 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 27-4 和外部器件的通信



- 注 1. 必须通过电容器（0.47 ~ 1μF）将 REGC 引脚接地。
2. 只限于 64 引脚产品。

外部器件对 RL78 微控制器生成以下信号。

表 27-3 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} 、EV _{DD0} 注 2
GND	—	接地	V _{SS} 、EV _{SS0} 注 2、REGC 注 1
RESETOUT	输出	复位信号输出	RESET
RxD	输入	接收信号	TOOLTxD
TxD	输出	发送信号	TOOLRxD
PORT	输出	模式信号	TOOL0

- 注 1. 必须通过电容器（0.47 ~ 1μF）将 REGC 引脚接地。
2. 只限于 64 引脚产品。

27.3 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“27.4.2 闪存编程模式”。

27.3.1 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“31.10 闪存编程模式的进入时序”。

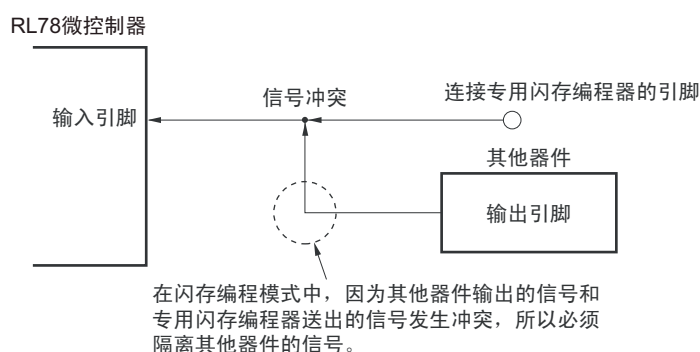
2. 因为 RL78 微控制器和专用闪存编程器的通信使用单线 UART (TOOL0 引脚)，所以不使用 SAU 引脚和 IICA 引脚。

27.3.2 $\overline{\text{RESET}}$ 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 27-5 信号冲突 ($\overline{\text{RESET}}$ 引脚)



27.3.3 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 EV_{DD0} ，或者通过电阻将引脚连接 V_{SS} 或者 EV_{SS0} 进行引脚处理。

27.3.4 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。

27.3.5 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} ）。

27.3.6 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。

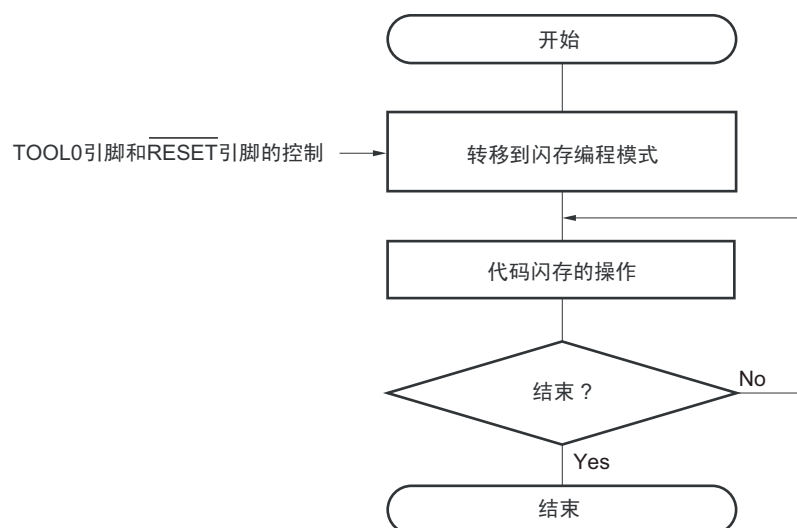
但是，在使用闪存编程器进行编程时，即使使用电路板上的电源，也必须将 V_{DD} 引脚和 V_{SS} 引脚分别连接闪存编程器的 V_{DD} 和 GND，以便通过闪存编程器监视电压。

27.4 编程方法

27.4.1 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 27-6 代码闪存的操作步骤



27.4.2 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。

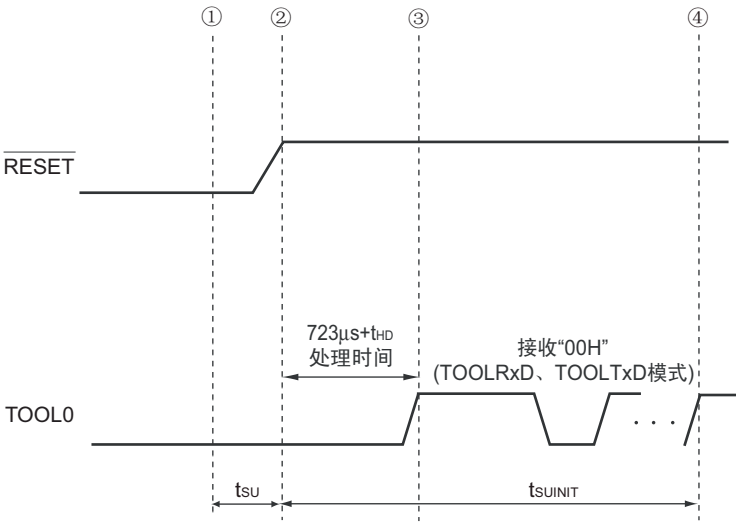
＜使用外部器件进行串行编程的情况＞

在将 TOOL0 引脚置为低电平后解除复位（参照表 27-4），然后按照图 27-7 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 27-4 解除复位时的 TOOL0 引脚和运行模式的关系

TOOL0	运行模式
EV _{DD}	通常运行模式
0V	闪存编程模式

图 27-7 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_{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）。

详细内容请参照“31.10 闪存编程模式的进入时序”。

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 27-5 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 (f_{CLK})	
$2.7V \leq V_{DD} \leq 5.5V$	空白状态		全速模式
	HS (高速主) 模式	1MHz ~ 32MHz	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“27.4.4 通信命令”。

27.4.3 通信方式

RL78 微控制器的通信方式如下所示。

表 27-6 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。

27.4.4 通信命令

RL78 微控制器通过表 27-7 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 27-7 闪存控制命令

分类	命令名称	功能
校验	Verify	将指定闪存区的内容和从编程器送来的数据进行比较。
擦除	Block Erase	擦除指定的闪存区。
空白检查	Block Blank Check	检查指定块的闪存擦除状态。
编程	Programming	将数据写到指定的闪存区注。
信息取得	Silicon Signature	获取 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 27-8 和表 27-9 所示。

表 27-8 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 ~ 0FFFFH （64KB）→FFH、FFH、00H）	3 字节
数据闪存区的结束地址	数据闪存区的结束地址 （从地址的低位开始发送。例：F1000H ~ F1FFFH （4KB）→FFH、1FH、0FH）	3 字节
固件版本	用于编程的固件版本信息 （从版本的高位开始发送。例：Ver.1.23→01H、02H、03H）	3 字节

表 27-9 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	R7F0C014	10 字节	52 = “R” 37 = “7” 46 = “F” 30 = “0” 43 = “C” 30 = “0” 31 = “1” 34 = “4” 20 = “ ” 20 = “ ”
代码闪存区的结束地址	代码闪存区 00000H ~ 0FFFFH（64KB）	3 字节	FF FF 00
数据闪存区的结束地址	数据闪存区 F1000H ~ F1FFFH（4KB）	3 字节	FF 1F 0F
固件版本	Ver.1.23	3 字节	01 02 03

27.5 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 27-10 使用 PG-FP5 时的各命令处理时间（参考值）

PG-FP5 的命令	Port: TOOL0(UART)
	Speed: 1Mbps
	128K 字节
擦除	2s
编程	3.5s
校验	3.5s
擦除后的编程	4.5s

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）

27.6 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闪存，所以能在现场进行程序升级等。

注意 1. 当 CPU 以副系统时钟运行时，不能使用自编程功能。

2. 为了在自编程过程中禁止中断，必须和通常运行模式中一样，在通过 DI 指令将 IE 标志清“0”的状态下执行闪存自编程库。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。
3. 在自编程过程中，需要使高速内部振荡器振荡。当高速内部振荡器处于停止状态时，必须使高速内部振荡器时钟运行（HIOSTOP=0），并且在经过 30μs（用户选项字节（000C2H）的 FRQSEL4 位为“0”）或者 80μs（FRQSEL4 位为“1”）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。

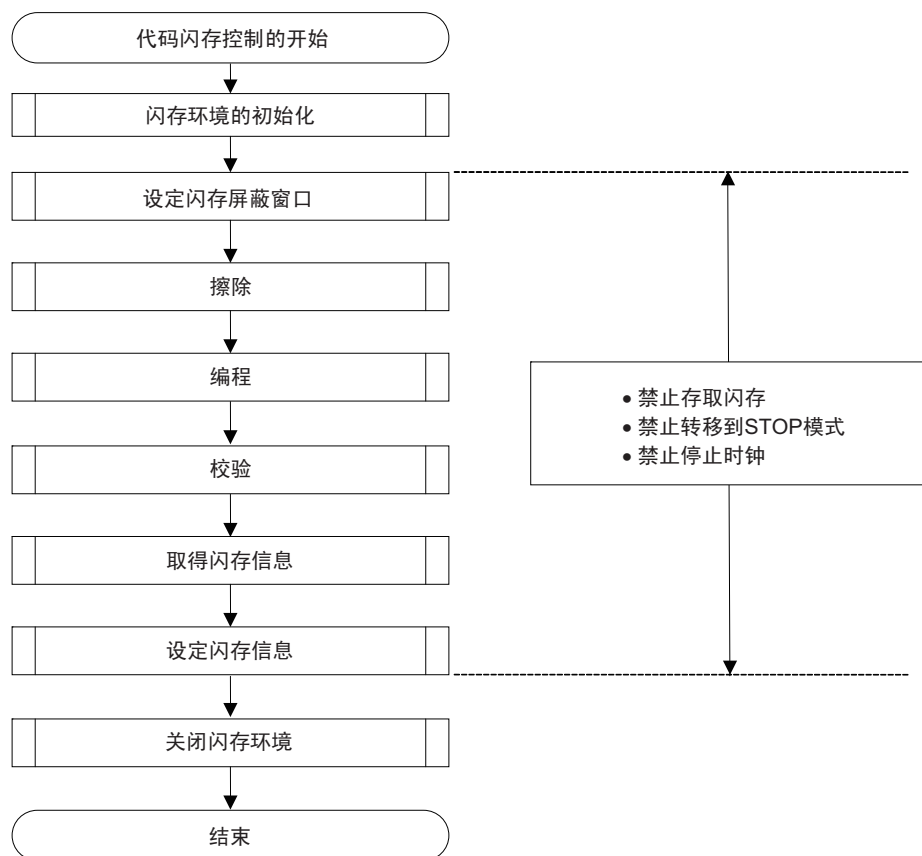
在执行本公司提供的闪存自编程库的“FSL_Init”函数时，如果参数“fsl_flash_voltage_u08”为“00H”，就设定为全速模式。否则，就设定为宽电压模式。

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。

27.6.1 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 27-8 自编程流程（改写闪存）



27.6.2 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。

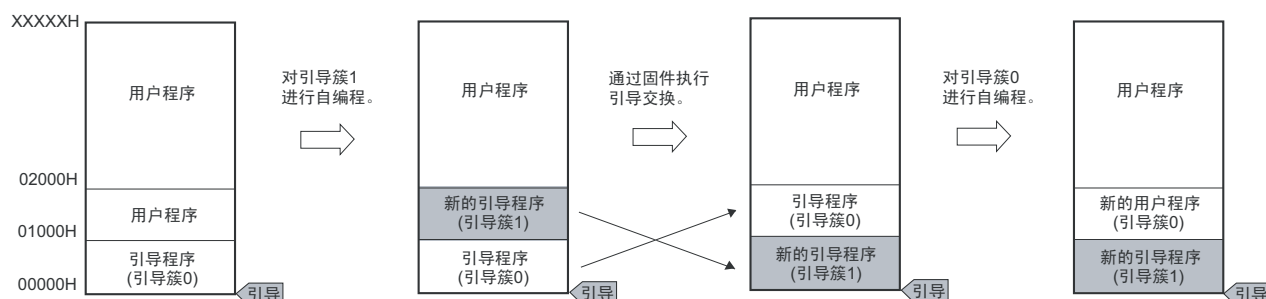
为了避免此问题，提供了引导交换功能。

在擦除引导程序区的引导簇 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，也能在下次的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 27-9 引导交换功能

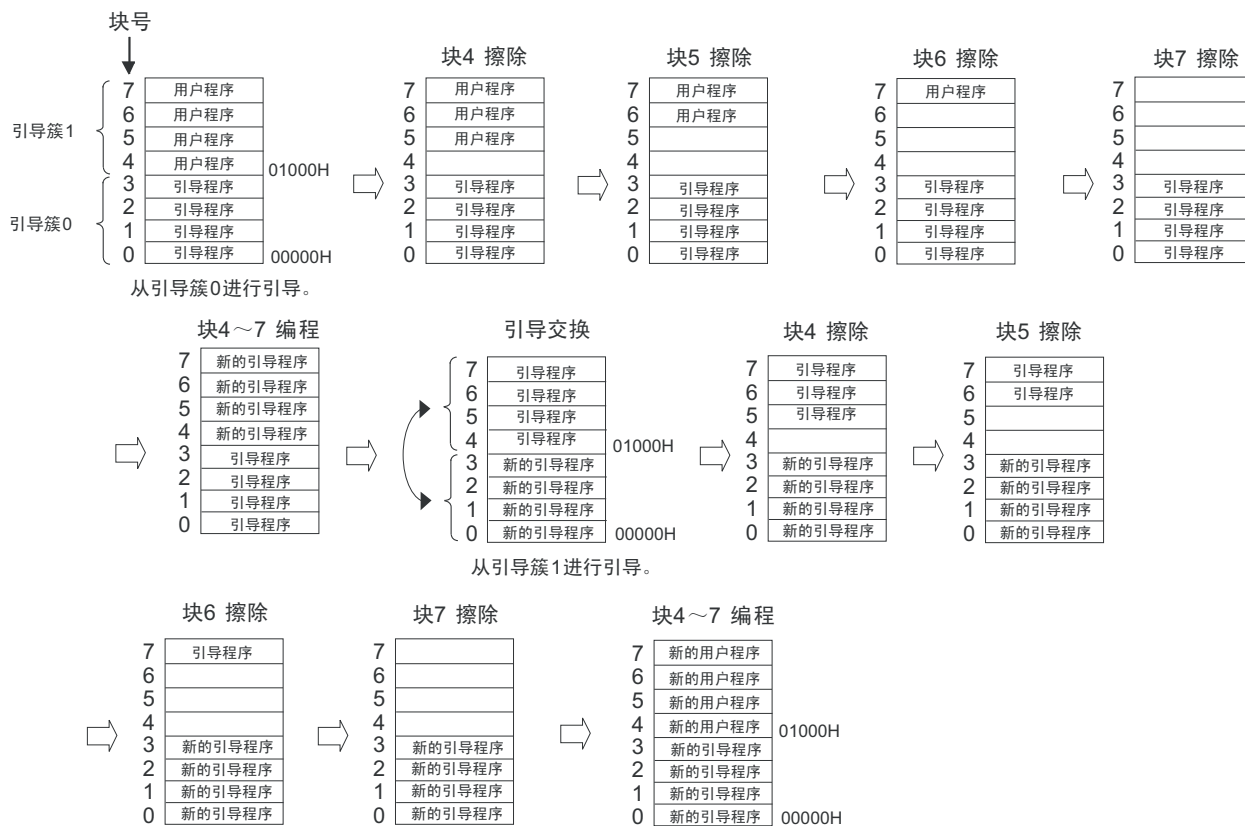


在此图的例子中，

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 27-10 引导交换的执行例子



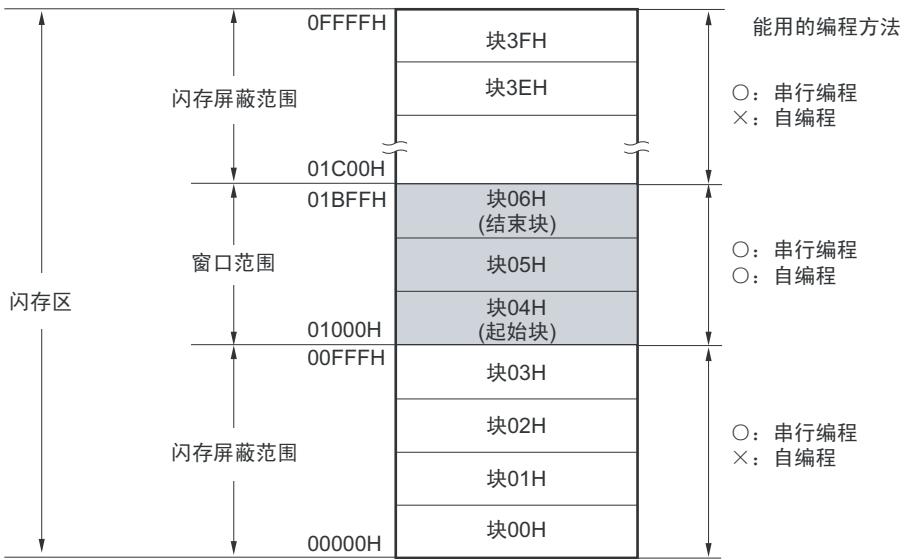
27.6.3 闪存屏蔽窗口功能

闪存屏蔽窗口功能作为自编程的一个安全功能，只在自编程时禁止对指定的窗口范围以外的区域进行编程和擦除。

能通过指定起始块和结束块来设定窗口范围。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 27-11 闪存屏蔽窗口的设定例子
(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 27-11 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 有关要在串行编程时禁止编程或者擦除的情况，请参照“27.7 安全设定”。

27.7 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 27-12 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 27.6.3）。

表 27-12 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 27.6.3）。

表 27-13 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。

27.8 数据闪存

27.8.1 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 Type04》。
- 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 模式状态。

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（用户选项字节（000C2H）的 FRQSEL4 位为“0”）或者 80μs（FRQSEL4 位为“1”）后执行数据闪存库。

备注 有关闪存编程模式，请参照“27.6 自编程”。

27.8.2 控制数据闪存的寄存器

27.8.2.1 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 27-12 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	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。

27.8.3 数据闪存的存取步骤

在解除复位后的初始状态中，数据闪存为停止状态，在此状态下无法存取（读或者编程）数据闪存。要存取时，必须按照以下步骤进行。

- ① 给数据闪存控制寄存器（DFLCTL）的bit0（DFLEN）写“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模式而不同。
<各主时钟模式的准备时间>
 - HS（高速主）模式 : 5μs
 - LS（低速主）模式 : 720ns
 - LV（低电压主）模式 : 10μs
- 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，必须将 DFLEN 位置“0”，然后执行 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 30μs（用户选项字节（000C2H）的 FRQSEL4 位为“0”）或者 80μs（FRQSEL4 位为“1”）后执行数据闪存库。

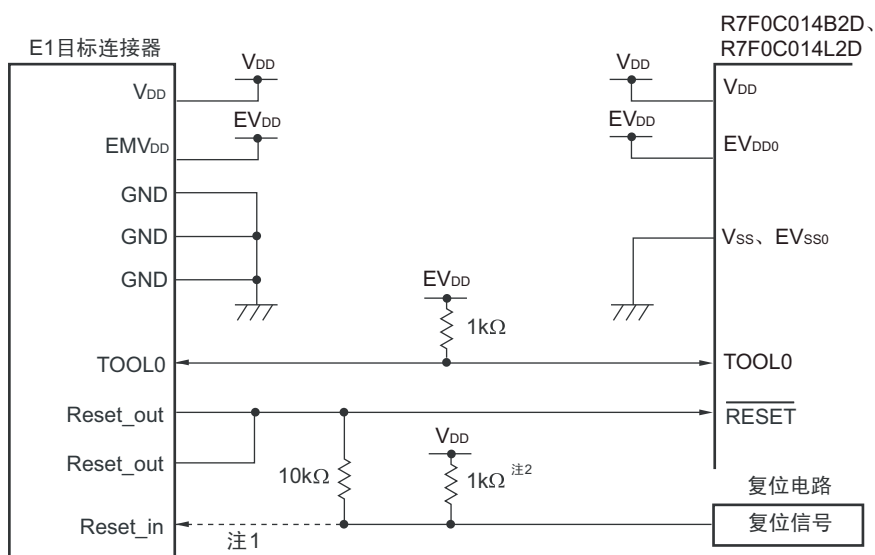
第 28 章 片上调试功能

28.1 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} 、 $\overline{RESET}$ 、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 28-1 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。

备注 对于没有 EV_{DD0} 引脚和 EV_{SS0} 引脚的产品，必须将 EV_{DD0} 替换为 V_{DD} 并且将 EV_{SS0} 替换为 V_{SS} 。

28.2 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 26 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 28-1 片上调试安全 ID

地址	片上调试安全 ID 码
000C4H ~ 000CDH	任意 10 字节的 ID 码
010C4H ~ 010CDH	

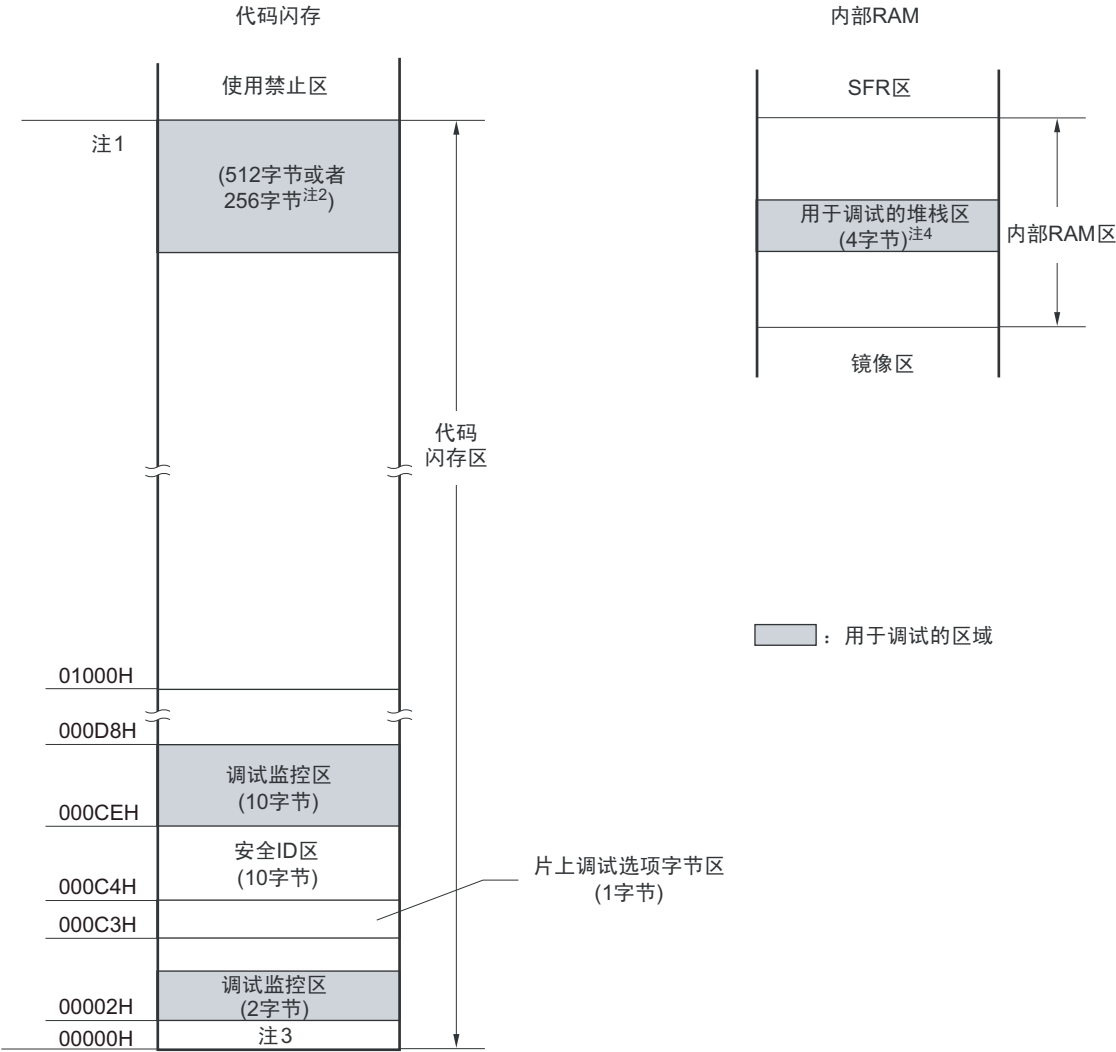
28.3 用户资源的确保

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，需要事先确保存储空间。当使用本公司的汇编程序和编译程序时，也能通过链接选项进行设定。

(1) 存储空间的确保

图 28-2 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 28-2 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
R7F0C014B2D、R7F0C014L2D	1FFFFH

- 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 29 章 十进制校正（BCD）电路

29.1 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。

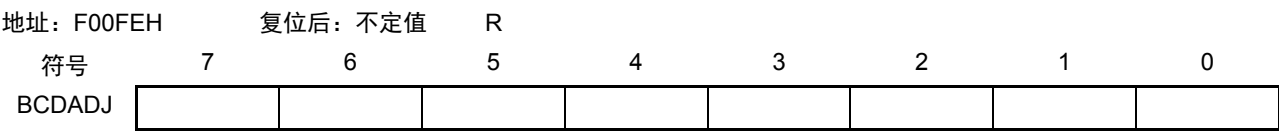
29.2 十进制校正电路使用的寄存器

- 十进制校正电路使用以下寄存器：
- BCD 校正结果寄存器（BCDADJ）

29.2.1 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 29-1 BCD 校正结果寄存器（BCDADJ）的格式



29.3 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 30 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》（R01US0015）。

30.1 凡例

30.1.1 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 30-1 中括号内的名称，如 R0、R1、R2 等）进行记述。

表 30-1 操作数的表现形式和记述方法

表现形式	记述方法
r	X(R0)、A(R1)、C(R2)、B(R3)、E(R4)、D(R5)、L(R6)、H(R7)
rp	AX(RP0)、BC(RP1)、DE(RP2)、HL(RP3)
sfr	特殊功能寄存器的符号（SFR 符号）FFF00H ~ FFFFFH
sfrp	特殊功能寄存器的符号（可进行 16 位操作的 SFR 符号，只限于偶数地址注）FFF00H ~ FFFFFH
saddr	FFE20H ~ FFF1FH 立即数或者标号
saddrp	FFE20H ~ FFF1FH 立即数或者标号（只限于偶数地址注）
addr20	00000H ~ FFFFFH 立即数或者标号
addr16	0000H ~ FFFFH 立即数或者标号（在 16 位数据时，只限于偶数地址注）
addr5	0080H ~ 00BFH 立即数或者标号（只限于偶数地址）
word	16 位立即数或者标号
byte	8 位立即数或者标号
bit	3 位立即数或者标号
RBn	RB0 ~ RB3

注 当指定奇数地址时，bit0 为“0”。

备注 能用符号将特殊功能寄存器记述为操作数 sfr。有关特殊功能寄存器的符号，请参照“表 3-5 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”。

30.1.2 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 30-2 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)

30.1.3 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 30-3 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。

30.1.4 PREFIX 指令

ES: 所示的指令以 PREFIX 指令码为前缀，将能存取的数据区从 F0000H ~ FFFFFH 的 64K 字节空间扩展为附加 ES 寄存器值的 00000H ~ FFFFFH 的 1M 字节空间。将 PREFIX 指令码置于对象指令的前头，只有紧接在 PREFIX 指令码后的 1 条指令作为附加了 ES 寄存器值的地址进行执行。

在 PREFIX 指令码和紧随其后的 1 条指令之间，不接受中断和 DTC 传送。

表 30-4 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。

30.2 操作一览表

表 30-5 操作一览表 (1/18)

指令群	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位数据传送	MOV	r, #byte	2	1	—	r←byte			
		PSW, #byte	3	3	—	PSW←byte	×	×	×
		CS, #byte	3	1	—	CS←byte			
		ES, #byte	2	1	—	ES←byte			
		!addr16, #byte	4	1	—	(addr16)←byte			
		ES:!addr16, #byte	5	2	—	(ES, addr16)←byte			
		saddr, #byte	3	1	—	(saddr)←byte			
		sfr, #byte	3	1	—	sfr←byte			
		[DE+byte], #byte	3	1	—	(DE+byte)←byte			
		ES:[DE+byte], #byte	4	2	—	((ES, DE)+byte)←byte			
		[HL+byte], #byte	3	1	—	(HL+byte)←byte			
		ES:[HL+byte], #byte	4	2	—	((ES, HL)+byte)←byte			
		[SP+byte], #byte	3	1	—	(SP+byte)←byte			
		word[B], #byte	4	1	—	(B+word)←byte			
		ES:word[B], #byte	5	2	—	((ES, B)+word)←byte			
		word[C], #byte	4	1	—	(C+word)←byte			
		ES:word[C], #byte	5	2	—	((ES, C)+word)←byte			
		word[BC], #byte	4	1	—	(BC+word)←byte			
		ES:word[BC], #byte	5	2	—	((ES, BC)+word)←byte			
		A, r 注 3	1	1	—	A←r			
		r, A 注 3	1	1	—	r←A			
		A, PSW	2	1	—	A←PSW			
		PSW, A	2	3	—	PSW←A	×	×	×
		A, CS	2	1	—	A←CS			
		CS, A	2	1	—	CS←A			
		A, ES	2	1	—	A←ES			
		ES, A	2	1	—	ES←A			
		A, !addr16	3	1	4	A←(addr16)			
		A, ES:!addr16	4	2	5	A←(ES, addr16)			
		!addr16, A	3	1	—	(addr16)←A			
		ES:!addr16, A	4	2	—	(ES, addr16)←A			
		A, saddr	2	1	—	A←(saddr)			
		saddr, A	2	1	—	(saddr)←A			

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取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 30-5 操作一览表 (2/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位 数 据 传 送	MOV	A, sfr	2	1	—	$A \leftarrow \text{sfr}$			
		sfr, A	2	1	—	$\text{sfr} \leftarrow A$			
		A, [DE]	1	1	4	$A \leftarrow ([DE])$			
		[DE], A	1	1	—	$([DE]) \leftarrow A$			
		A, ES:[DE]	2	2	5	$A \leftarrow (ES, DE)$			
		ES:[DE], A	2	2	—	$(ES, DE) \leftarrow A$			
		A, [HL]	1	1	4	$A \leftarrow ([HL])$			
		[HL], A	1	1	—	$([HL]) \leftarrow A$			
		A, ES:[HL]	2	2	5	$A \leftarrow (ES, HL)$			
		ES:[HL], A	2	2	—	$(ES, HL) \leftarrow A$			
		A, [DE+byte]	2	1	4	$A \leftarrow ([DE] + \text{byte})$			
		[DE+byte], A	2	1	—	$([DE] + \text{byte}) \leftarrow A$			
		A, ES:[DE+byte]	3	2	5	$A \leftarrow ((ES, DE) + \text{byte})$			
		ES:[DE+byte], A	3	2	—	$((ES, DE) + \text{byte}) \leftarrow A$			
		A, [HL+byte]	2	1	4	$A \leftarrow ([HL] + \text{byte})$			
		[HL+byte], A	2	1	—	$([HL] + \text{byte}) \leftarrow A$			
		A, ES:[HL+byte]	3	2	5	$A \leftarrow ((ES, HL) + \text{byte})$			
		ES:[HL+byte], A	3	2	—	$((ES, HL) + \text{byte}) \leftarrow A$			
		A, [SP+byte]	2	1	—	$A \leftarrow ([SP] + \text{byte})$			
		[SP+byte], A	2	1	—	$([SP] + \text{byte}) \leftarrow A$			
		A, word[B]	3	1	4	$A \leftarrow (B + \text{word})$			
		word[B], A	3	1	—	$(B + \text{word}) \leftarrow A$			
		A, ES:word[B]	4	2	5	$A \leftarrow ((ES, B) + \text{word})$			
		ES:word[B], A	4	2	—	$((ES, B) + \text{word}) \leftarrow A$			
		A, word[C]	3	1	4	$A \leftarrow (C + \text{word})$			
		word[C], A	3	1	—	$(C + \text{word}) \leftarrow A$			
		A, ES:word[C]	4	2	5	$A \leftarrow ((ES, C) + \text{word})$			
		ES:word[C], A	4	2	—	$((ES, C) + \text{word}) \leftarrow A$			
		A, word[BC]	3	1	4	$A \leftarrow (BC + \text{word})$			
		word[BC], A	3	1	—	$(BC + \text{word}) \leftarrow A$			
		A, ES:word[BC]	4	2	5	$A \leftarrow ((ES, BC) + \text{word})$			
		ES:word[BC], A	4	2	—	$((ES, BC) + \text{word}) \leftarrow A$			

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{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 30-5 操作一览表 (3/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位 数 据 传 送	MOV	A, [HL+B]	2	1	4	$A \leftarrow (HL+B)$			
		[HL+B], A	2	1	—	$(HL+B) \leftarrow A$			
		A, ES:[HL+B]	3	2	5	$A \leftarrow ((ES, HL)+B)$			
		ES:[HL+B], A	3	2	—	$((ES, HL)+B) \leftarrow A$			
		A, [HL+C]	2	1	4	$A \leftarrow (HL+C)$			
		[HL+C], A	2	1	—	$(HL+C) \leftarrow A$			
		A, ES:[HL+C]	3	2	5	$A \leftarrow ((ES, HL)+C)$			
		ES:[HL+C], A	3	2	—	$((ES, HL) + C) \leftarrow A$			
		X, !addr16	3	1	4	$X \leftarrow (addr16)$			
		X, ES:!addr16	4	2	5	$X \leftarrow (ES, addr16)$			
		X, saddr	2	1	—	$X \leftarrow (saddr)$			
		B, !addr16	3	1	4	$B \leftarrow (addr16)$			
		B, ES:!addr16	4	2	5	$B \leftarrow (ES, addr16)$			
		B, saddr	2	1	—	$B \leftarrow (saddr)$			
		C, !addr16	3	1	4	$C \leftarrow (addr16)$			
		C, ES:!addr16	4	2	5	$C \leftarrow (ES, addr16)$			
		C, saddr	2	1	—	$C \leftarrow (saddr)$			
		ES, saddr	3	1	—	$ES \leftarrow (saddr)$			
	XCH	A, r 注 3	1 (r=X) 2 (r=X 以外)	1	—	$A \leftrightarrow r$			
		A, !addr16	4	2	—	$A \leftrightarrow (addr16)$			
		A, ES:!addr16	5	3	—	$A \leftrightarrow (ES, addr16)$			
		A, saddr	3	2	—	$A \leftrightarrow (saddr)$			
		A, sfr	3	2	—	$A \leftrightarrow sfr$			
		A, [DE]	2	2	—	$A \leftrightarrow (DE)$			
		A, ES:[DE]	3	3	—	$A \leftrightarrow (ES, DE)$			
		A, [HL]	2	2	—	$A \leftrightarrow (HL)$			
		A, ES:[HL]	3	3	—	$A \leftrightarrow (ES, HL)$			
		A, [DE+byte]	3	2	—	$A \leftrightarrow (DE+byte)$			
		A, ES:[DE+byte]	4	3	—	$A \leftrightarrow ((ES, DE)+byte)$			
		A, [HL+byte]	3	2	—	$A \leftrightarrow (HL+byte)$			
		A, ES:[HL+byte]	4	3	—	$A \leftrightarrow ((ES, HL)+byte)$			

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取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 30-5 操作一览表 (4/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位数据传送	XCH	A, [HL+B]	2	2	—	$A \leftrightarrow (HL+B)$			
		A, ES:[HL+B]	3	3	—	$A \leftrightarrow ((ES, HL)+B)$			
		A, [HL+C]	2	2	—	$A \leftrightarrow (HL+C)$			
		A, ES:[HL+C]	3	3	—	$A \leftrightarrow ((ES, HL)+C)$			
	ONEB	A	1	1	—	$A \leftarrow 01H$			
		X	1	1	—	$X \leftarrow 01H$			
		B	1	1	—	$B \leftarrow 01H$			
		C	1	1	—	$C \leftarrow 01H$			
		!addr16	3	1	—	$(addr16) \leftarrow 01H$			
		ES:!addr16	4	2	—	$(ES, addr16) \leftarrow 01H$			
		saddr	2	1	—	$(saddr) \leftarrow 01H$			
	CLRB	A	1	1	—	$A \leftarrow 00H$			
		X	1	1	—	$X \leftarrow 00H$			
		B	1	1	—	$B \leftarrow 00H$			
		C	1	1	—	$C \leftarrow 00H$			
		!addr16	3	1	—	$(addr16) \leftarrow 00H$			
		ES:!addr16	4	2	—	$(ES, addr16) \leftarrow 00H$			
		saddr	2	1	—	$(saddr) \leftarrow 00H$			
	MOVS	[HL+byte], X	3	1	—	$(HL+byte) \leftarrow X$	×		×
		ES:[HL+byte], X	4	2	—	$(ES, HL+byte) \leftarrow X$	×		×
16 位数据传送	MOVW	rp, #word	3	1	—	$rp \leftarrow word$			
		saddrp, #word	4	1	—	$(saddrp) \leftarrow word$			
		sfrp, #word	4	1	—	$sfrp \leftarrow word$			
		AX, rp 注 3	1	1	—	$AX \leftarrow rp$			
		rp, AX 注 3	1	1	—	$rp \leftarrow AX$			
		AX, !addr16	3	1	4	$AX \leftarrow (addr16)$			
		!addr16, AX	3	1	—	$(addr16) \leftarrow AX$			
		AX, ES:!addr16	4	2	5	$AX \leftarrow (ES, addr16)$			
		ES:!addr16, AX	4	2	—	$(ES, addr16) \leftarrow AX$			
		AX, saddrp	2	1	—	$AX \leftarrow (saddrp)$			
		saddrp, AX	2	1	—	$(saddrp) \leftarrow AX$			
		AX, sfrp	2	1	—	$AX \leftarrow sfrp$			
		sfrp, AX	2	1	—	$sfrp \leftarrow AX$			

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 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 30-5 操作一览表 (5/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位 数 据 传 送	MOVW	AX, [DE]	1	1	4	AX←(DE)			
		[DE], AX	1	1	—	(DE)←AX			
		AX, ES:[DE]	2	2	5	AX←(ES, DE)			
		ES:[DE], AX	2	2	—	(ES, DE)←AX			
		AX, [HL]	1	1	4	AX←(HL)			
		[HL], AX	1	1	—	(HL)←AX			
		AX, ES:[HL]	2	2	5	AX←(ES, HL)			
		ES:[HL], AX	2	2	—	(ES, HL)←AX			
		AX, [DE+byte]	2	1	4	AX←(DE+byte)			
		[DE+byte], AX	2	1	—	(DE+byte)←AX			
		AX, ES:[DE+byte]	3	2	5	AX←((ES, DE)+byte)			
		ES:[DE+byte], AX	3	2	—	((ES, DE)+byte)←AX			
		AX, [HL+byte]	2	1	4	AX←(HL+byte)			
		[HL+byte], AX	2	1	—	(HL+byte)←AX			
		AX, ES:[HL+byte]	3	2	5	AX←((ES, HL)+byte)			
		ES:[HL+byte], AX	3	2	—	((ES, HL)+byte)←AX			
		AX, [SP+byte]	2	1	—	AX←(SP+byte)			
		[SP+byte], AX	2	1	—	(SP+byte)←AX			
		AX, word[B]	3	1	4	AX←(B+word)			
		word[B], AX	3	1	—	(B+word)←AX			
		AX, ES:word[B]	4	2	5	AX←((ES, B)+word)			
		ES:word[B], AX	4	2	—	((ES, B)+word)←AX			
		AX, word[C]	3	1	4	AX←(C+word)			
		word[C], AX	3	1	—	(C+word)←AX			
		AX, ES:word[C]	4	2	5	AX←((ES, C)+word)			
		ES:word[C], AX	4	2	—	((ES, C)+word)←AX			
		AX, word[BC]	3	1	4	AX←(BC+word)			
		word[BC], AX	3	1	—	(BC+word)←AX			
		AX, ES:word[BC]	4	2	5	AX←((ES, BC)+word)			
		ES:word[BC], AX	4	2	—	((ES, BC)+word)←AX			

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取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 30-5 操作一览表 (6/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位 数 据 传 送	MOVW	BC, !addr16	3	1	4	BC←(addr16)			
		BC, ES:!addr16	4	2	5	BC←(ES, addr16)			
		DE, !addr16	3	1	4	DE←(addr16)			
		DE, ES:!addr16	4	2	5	DE←(ES, addr16)			
		HL, !addr16	3	1	4	HL←(addr16)			
		HL, ES:!addr16	4	2	5	HL←(ES, addr16)			
		BC, saddrp	2	1	—	BC←(saddrp)			
		DE, saddrp	2	1	—	DE←(saddrp)			
		HL, saddrp	2	1	—	HL←(saddrp)			
	XCHW	AX, rp 注 3	1	1	—	AX↔rp			
	ONEW	AX	1	1	—	AX←0001H			
		BC	1	1	—	BC←0001H			
	CLRW	AX	1	1	—	AX←0000H			
		BC	1	1	—	BC←0000H			
8 位 运 算	ADD	A, #byte	2	1	—	A, CY←A+byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY←(saddr)+byte	×	×	×
		A, r 注 4	2	1	—	A, CY←A+r	×	×	×
		r, A	2	1	—	r, CY←r+A	×	×	×
		A, !addr16	3	1	4	A, CY←A+(addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY←A+(ES, addr16)	×	×	×
		A, saddr	2	1	—	A, CY←A+(saddr)	×	×	×
		A, [HL]	1	1	4	A, CY←A+(HL)	×	×	×
		A, ES:[HL]	2	2	5	A, CY←A+(ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY←A+(HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A, CY←A+((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY←A+(HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A, CY←A+((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY←A+(HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY←A+((ES, HL)+C)	×	×	×

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取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 30-5 操作一览表 (7/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位运算	ADDC	A, #byte	2	1	—	A, CY←A+byte+CY	×	×	×
		saddr, #byte	3	2	—	(saddr), CY←(saddr)+byte+CY	×	×	×
		A, r 注 3	2	1	—	A, CY←A+r+CY	×	×	×
		r, A	2	1	—	r, CY←r+A+CY	×	×	×
		A, !addr16	3	1	4	A, CY←A+(addr16)+CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY←A+(ES, addr16)+CY	×	×	×
		A, saddr	2	1	—	A, CY←A+(saddr)+CY	×	×	×
		A, [HL]	1	1	4	A, CY←A+(HL)+CY	×	×	×
		A, ES:[HL]	2	2	5	A, CY←A+(ES, HL)+CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY←A+(HL+byte)+CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A, CY←A+((ES, HL)+byte)+CY	×	×	×
		A, [HL+B]	2	1	4	A, CY←A+(HL+B)+CY	×	×	×
		A, ES:[HL+B]	3	2	5	A, CY←A+((ES, HL)+B)+CY	×	×	×
		A, [HL+C]	2	1	4	A, CY←A+(HL+C)+CY	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY←A+((ES, HL)+C)+CY	×	×	×
	SUB	A, #byte	2	1	—	A, CY←A-byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY←(saddr)-byte	×	×	×
		A, r 注 3	2	1	—	A, CY←A-r	×	×	×
		r, A	2	1	—	r, CY←r-A	×	×	×
		A, !addr16	3	1	4	A, CY←A-(addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY←A-(ES, addr16)	×	×	×
		A, saddr	2	1	—	A, CY←A-(saddr)	×	×	×
		A, [HL]	1	1	4	A, CY←A-(HL)	×	×	×
		A, ES:[HL]	2	2	5	A, CY←A-(ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY←A-(HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A, CY←A-((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY←A-(HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A, CY←A-((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY←A-(HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY←A-((ES, HL)+C)	×	×	×

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取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 30-5 操作一览表 (8/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位运算	SUBC	A, #byte	2	1	—	$A, CY \leftarrow A - \text{byte} - CY$	×	×	×
		saddr, #byte	3	2	—	$(saddr), CY \leftarrow (saddr) - \text{byte} - CY$	×	×	×
		A, r 注 3	2	1	—	$A, CY \leftarrow A - r - CY$	×	×	×
		r, A	2	1	—	$r, CY \leftarrow r - A - CY$	×	×	×
		A, !addr16	3	1	4	$A, CY \leftarrow A - (\text{addr16}) - CY$	×	×	×
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A - (\text{ES}, \text{addr16}) - CY$	×	×	×
		A, saddr	2	1	—	$A, CY \leftarrow A - (saddr) - CY$	×	×	×
		A, [HL]	1	1	4	$A, CY \leftarrow A - (\text{HL}) - CY$	×	×	×
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A - (\text{ES}, \text{HL}) - CY$	×	×	×
		A, [HL+byte]	2	1	4	$A, CY \leftarrow A - (\text{HL} + \text{byte}) - CY$	×	×	×
		A, ES:[HL+byte]	3	2	5	$A, CY \leftarrow A - ((\text{ES}, \text{HL}) + \text{byte}) - CY$	×	×	×
		A, [HL+B]	2	1	4	$A, CY \leftarrow A - (\text{HL} + B) - CY$	×	×	×
		A, ES:[HL+B]	3	2	5	$A, CY \leftarrow A - ((\text{ES}, \text{HL}) + B) - CY$	×	×	×
		A, [HL+C]	2	1	4	$A, CY \leftarrow A - (\text{HL} + C) - CY$	×	×	×
		A, ES:[HL+C]	3	2	5	$A, CY \leftarrow A - ((\text{ES}, \text{HL}) + C) - CY$	×	×	×
	AND	A, #byte	2	1	—	$A \leftarrow A \wedge \text{byte}$	×		
		saddr, #byte	3	2	—	$(saddr) \leftarrow (saddr) \wedge \text{byte}$	×		
		A, r 注 3	2	1	—	$A \leftarrow A \wedge r$	×		
		r, A	2	1	—	$r \leftarrow r \wedge A$	×		
		A, !addr16	3	1	4	$A \leftarrow A \wedge (\text{addr16})$	×		
		A, ES:!addr16	4	2	5	$A \leftarrow A \wedge (\text{ES}, \text{addr16})$	×		
		A, saddr	2	1	—	$A \leftarrow A \wedge (saddr)$	×		
		A, [HL]	1	1	4	$A \leftarrow A \wedge (\text{HL})$	×		
		A, ES:[HL]	2	2	5	$A \leftarrow A \wedge (\text{ES}, \text{HL})$	×		
		A, [HL+byte]	2	1	4	$A \leftarrow A \wedge (\text{HL} + \text{byte})$	×		
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \wedge ((\text{ES}, \text{HL}) + \text{byte})$	×		
		A, [HL+B]	2	1	4	$A \leftarrow A \wedge (\text{HL} + B)$	×		
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \wedge ((\text{ES}, \text{HL}) + B)$	×		
		A, [HL+C]	2	1	4	$A \leftarrow A \wedge (\text{HL} + C)$	×		
		A, ES:[HL+C]	3	2	5	$A \leftarrow A \wedge ((\text{ES}, \text{HL}) + C)$	×		

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{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 30-5 操作一览表 (9/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位运算	OR	A, #byte	2	1	—	$A \leftarrow A \vee \text{byte}$	×		
		saddr, #byte	3	2	—	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	×		
		A, r 注 3	2	1	—	$A \leftarrow A \vee r$	×		
		r, A	2	1	—	$r \leftarrow r \vee A$	×		
		A, !addr16	3	1	4	$A \leftarrow A \vee (\text{addr16})$	×		
		A, ES:!addr16	4	2	5	$A \leftarrow A \vee (\text{ES:addr16})$	×		
		A, saddr	2	1	—	$A \leftarrow A \vee (\text{saddr})$	×		
		A, [HL]	1	1	4	$A \leftarrow A \vee (\text{HL})$	×		
		A, ES:[HL]	2	2	5	$A \leftarrow A \vee (\text{ES:HL})$	×		
		A, [HL+byte]	2	1	4	$A \leftarrow A \vee (\text{HL} + \text{byte})$	×		
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + \text{byte})$	×		
		A, [HL+B]	2	1	4	$A \leftarrow A \vee (\text{HL} + B)$	×		
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + B)$	×		
		A, [HL+C]	2	1	4	$A \leftarrow A \vee (\text{HL} + C)$	×		
		A, ES:[HL+C]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + C)$	×		
	XOR	A, #byte	2	1	—	$A \leftarrow A \oplus \text{byte}$	×		
		saddr, #byte	3	2	—	$(\text{saddr}) \leftarrow (\text{saddr}) \oplus \text{byte}$	×		
		A, r 注 3	2	1	—	$A \leftarrow A \oplus r$	×		
		r, A	2	1	—	$r \leftarrow r \oplus A$	×		
		A, !addr16	3	1	4	$A \leftarrow A \oplus (\text{addr16})$	×		
		A, ES:!addr16	4	2	5	$A \leftarrow A \oplus (\text{ES:addr16})$	×		
		A, saddr	2	1	—	$A \leftarrow A \oplus (\text{saddr})$	×		
		A, [HL]	1	1	4	$A \leftarrow A \oplus (\text{HL})$	×		
		A, ES:[HL]	2	2	5	$A \leftarrow A \oplus (\text{ES:HL})$	×		
		A, [HL+byte]	2	1	4	$A \leftarrow A \oplus (\text{HL} + \text{byte})$	×		
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL}) + \text{byte})$	×		
		A, [HL+B]	2	1	4	$A \leftarrow A \oplus (\text{HL} + B)$	×		
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL}) + B)$	×		
		A, [HL+C]	2	1	4	$A \leftarrow A \oplus (\text{HL} + C)$	×		
		A, ES:[HL+C]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL}) + C)$	×		

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{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 30-5 操作一览表 (10/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位运算	CMP	A, #byte	2	1	—	A-byte	×	×	×
		!addr16, #byte	4	1	4	(addr16)-byte	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16)-byte	×	×	×
		saddr, #byte	3	1	—	(saddr)-byte	×	×	×
		A, r 注 3	2	1	—	A-r	×	×	×
		r, A	2	1	—	r-A	×	×	×
		A, !addr16	3	1	4	A-(addr16)	×	×	×
		A, ES:!addr16	4	2	5	A-(ES:addr16)	×	×	×
		A, saddr	2	1	—	A-(saddr)	×	×	×
		A, [HL]	1	1	4	A-(HL)	×	×	×
		A, ES:[HL]	2	2	5	A-(ES:HL)	×	×	×
		A, [HL+byte]	2	1	4	A-(HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A-((ES:HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A-(HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A-((ES:HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A-(HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A-((ES:HL)+C)	×	×	×
	CMP0	A	1	1	—	A-00H	×	0	0
		X	1	1	—	X-00H	×	0	0
		B	1	1	—	B-00H	×	0	0
		C	1	1	—	C-00H	×	0	0
		!addr16	3	1	4	(addr16)-00H	×	0	0
		ES:!addr16	4	2	5	(ES:addr16)-00H	×	0	0
		saddr	2	1	—	(saddr)-00H	×	0	0
	CMPS	X, [HL+byte]	3	1	4	X-(HL+byte)	×	×	×
		X, ES:[HL+byte]	4	2	5	X-((ES:HL)+byte)	×	×	×

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取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 30-5 操作一览表 (11/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位运算	ADDW	AX, #word	3	1	—	AX, CY←AX+word	×	×	×
		AX, AX	1	1	—	AX, CY←AX+AX	×	×	×
		AX, BC	1	1	—	AX, CY←AX+BC	×	×	×
		AX, DE	1	1	—	AX, CY←AX+DE	×	×	×
		AX, HL	1	1	—	AX, CY←AX+HL	×	×	×
		AX, !addr16	3	1	4	AX, CY←AX+(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY←AX+(ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY←AX+(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY←AX+(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY←AX+((ES:HL)+byte)	×	×	×
	SUBW	AX, #word	3	1	—	AX, CY←AX-word	×	×	×
		AX, BC	1	1	—	AX, CY←AX-BC	×	×	×
		AX, DE	1	1	—	AX, CY←AX-DE	×	×	×
		AX, HL	1	1	—	AX, CY←AX-HL	×	×	×
		AX, !addr16	3	1	4	AX, CY←AX-(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY←AX-(ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY←AX-(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY←AX-(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY←AX-((ES:HL)+byte)	×	×	×
	CMPW	AX, #word	3	1	—	AX-word	×	×	×
		AX, BC	1	1	—	AX-BC	×	×	×
		AX, DE	1	1	—	AX-DE	×	×	×
		AX, HL	1	1	—	AX-HL	×	×	×
		AX, !addr16	3	1	4	AX-(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX-(ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX-(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX-(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX-((ES:HL)+byte)	×	×	×

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取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 30-5 操作一览表 (12/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
乘除运算和乘加运算	MULU	X	1	1	—	$AX \leftarrow A \times X$			
	MULHU		3	2	—	$BCAX \leftarrow AX \times BC$ (无符号)			
	MULH		3	2	—	$BCAX \leftarrow AX \times BC$ (带符号)			
	DIVHU		3	9	—	AX (商)、DE (余数) $\leftarrow AX \div DE$ (无符号)			
	DIVWU		3	17	—	BCAX (商)、HLDE (余数) $\leftarrow BCAX \div HLDE$ (无符号)			
	MACHU		3	3	—	$MACR \leftarrow MACR + AX \times BC$ (无符号)	×		×
	MACH		3	3	—	$MACR \leftarrow MACR + AX \times BC$ (带符号)	×		×

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. MACR: 乘加运算累加寄存器 (MACRH、MACRL)

表 30-5 操作一览表 (13/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
递增 / 递减	INC	r	1	1	—	$r \leftarrow r+1$	×	×	
		!addr16	3	2	—	$(addr16) \leftarrow (addr16)+1$	×	×	
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)+1$	×	×	
		saddr	2	2	—	$(saddr) \leftarrow (saddr)+1$	×	×	
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)+1$	×	×	
		ES: [HL+byte]	4	3	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)+1$	×	×	
	DEC	r	1	1	—	$r \leftarrow r-1$	×	×	
		!addr16	3	2	—	$(addr16) \leftarrow (addr16)-1$	×	×	
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)-1$	×	×	
		saddr	2	2	—	$(saddr) \leftarrow (saddr)-1$	×	×	
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)-1$	×	×	
		ES: [HL+byte]	4	3	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)-1$	×	×	
	INCW	rp	1	1	—	$rp \leftarrow rp+1$			
		!addr16	3	2	—	$(addr16) \leftarrow (addr16)+1$			
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)+1$			
		saddrp	2	2	—	$(saddrp) \leftarrow (saddrp)+1$			
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)+1$			
		ES: [HL+byte]	4	3	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)+1$			
	DECW	rp	1	1	—	$rp \leftarrow rp-1$			
		!addr16	3	2	—	$(addr16) \leftarrow (addr16)-1$			
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)-1$			
		saddrp	2	2	—	$(saddrp) \leftarrow (saddrp)-1$			
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)-1$			
		ES: [HL+byte]	4	3	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)-1$			
移位	SHR	A, cnt	2	1	—	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow 0) \times cnt$			×
	SHRW	AX, cnt	2	1	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow 0) \times cnt$			×
	SHL	A, cnt	2	1	—	$(CY \leftarrow A_7, A_m \leftarrow A_{m-1}, A_0 \leftarrow 0) \times cnt$			×
		B, cnt	2	1	—	$(CY \leftarrow B_7, B_m \leftarrow B_{m-1}, B_0 \leftarrow 0) \times cnt$			×
		C, cnt	2	1	—	$(CY \leftarrow C_7, C_m \leftarrow C_{m-1}, C_0 \leftarrow 0) \times cnt$			×
	SHLW	AX, cnt	2	1	—	$(CY \leftarrow AX_{15}, AX_m \leftarrow AX_{m-1}, AX_0 \leftarrow 0) \times cnt$			×
		BC, cnt	2	1	—	$(CY \leftarrow BC_{15}, BC_m \leftarrow BC_{m-1}, BC_0 \leftarrow 0) \times cnt$			×
	SAR	A, cnt	2	1	—	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow A_7) \times cnt$			×
	SARW	AX, cnt	2	1	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow AX_{15}) \times cnt$			×

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取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 30-5 操作一览表 (14/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
循环	ROR	A, 1	2	1	—	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$			×
	ROL	A, 1	2	1	—	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$			×
	RORC	A, 1	2	1	—	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			×
	ROLC	A, 1	2	1	—	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			×
	ROLWC	AX, 1	2	1	—	$(CY \leftarrow AX_{15}, AX_0 \leftarrow CY, AX_{m+1} \leftarrow AX_m) \times 1$			×
		BC, 1	2	1	—	$(CY \leftarrow BC_{15}, BC_0 \leftarrow CY, BC_{m+1} \leftarrow BC_m) \times 1$			×
位操作	MOV1	CY, A.bit	2	1	—	$CY \leftarrow A.bit$			×
		A.bit, CY	2	1	—	$A.bit \leftarrow CY$			
		CY, PSW.bit	3	1	—	$CY \leftarrow PSW.bit$			×
		PSW.bit, CY	3	4	—	$PSW.bit \leftarrow CY$	×	×	
		CY, saddr.bit	3	1	—	$CY \leftarrow (saddr).bit$			×
		saddr.bit, CY	3	2	—	$(saddr).bit \leftarrow CY$			
		CY, sfr.bit	3	1	—	$CY \leftarrow sfr.bit$			×
		sfr.bit, CY	3	2	—	$sfr.bit \leftarrow CY$			
		CY, [HL].bit	2	1	4	$CY \leftarrow (HL).bit$			×
		[HL].bit, CY	2	2	—	$(HL).bit \leftarrow CY$			
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow (ES, HL).bit$			×
		ES:[HL].bit, CY	3	3	—	$(ES, HL).bit \leftarrow CY$			
	AND1	CY, A.bit	2	1	—	$CY \leftarrow CY \wedge A.bit$			×
		CY, PSW.bit	3	1	—	$CY \leftarrow CY \wedge PSW.bit$			×
		CY, saddr.bit	3	1	—	$CY \leftarrow CY \wedge (saddr).bit$			×
		CY, sfr.bit	3	1	—	$CY \leftarrow CY \wedge sfr.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \wedge (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \wedge (ES, HL).bit$			×
	OR1	CY, A.bit	2	1	—	$CY \leftarrow CY \vee A.bit$			×
		CY, PSW.bit	3	1	—	$CY \leftarrow CY \vee PSW.bit$			×
		CY, saddr.bit	3	1	—	$CY \leftarrow CY \vee (saddr).bit$			×
		CY, sfr.bit	3	1	—	$CY \leftarrow CY \vee sfr.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \vee (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \vee (ES, HL).bit$			×

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取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 30-5 操作一览表 (15/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
位操作	XOR1	CY, A.bit	2	1	—	$CY \leftarrow CY \oplus A.bit$			×
		CY, PSW.bit	3	1	—	$CY \leftarrow CY \oplus PSW.bit$			×
		CY, saddr.bit	3	1	—	$CY \leftarrow CY \oplus (saddr).bit$			×
		CY, sfr.bit	3	1	—	$CY \leftarrow CY \oplus sfr.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \oplus (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \oplus (ES, HL).bit$			×
	SET1	A.bit	2	1	—	$A.bit \leftarrow 1$			
		PSW.bit	3	4	—	$PSW.bit \leftarrow 1$	×	×	×
		laddr16.bit	4	2	—	$(addr16).bit \leftarrow 1$			
		ES:laddr16.bit	5	3	—	$(ES, addr16).bit \leftarrow 1$			
		saddr.bit	3	2	—	$(saddr).bit \leftarrow 1$			
		sfr.bit	3	2	—	$sfr.bit \leftarrow 1$			
		[HL].bit	2	2	—	$(HL).bit \leftarrow 1$			
		ES:[HL].bit	3	3	—	$(ES, HL).bit \leftarrow 1$			
	CLR1	A.bit	2	1	—	$A.bit \leftarrow 0$			
		PSW.bit	3	4	—	$PSW.bit \leftarrow 0$	×	×	×
		laddr16.bit	4	2	—	$(addr16).bit \leftarrow 0$			
		ES:laddr16.bit	5	3	—	$(ES, addr16).bit \leftarrow 0$			
		saddr.bit	3	2	—	$(saddr).bit \leftarrow 0$			
		sfr.bit	3	2	—	$sfr.bit \leftarrow 0$			
		[HL].bit	2	2	—	$(HL).bit \leftarrow 0$			
		ES:[HL].bit	3	3	—	$(ES, HL).bit \leftarrow 0$			
	SET1	CY	2	1	—	$CY \leftarrow 1$			1
	CLR1	CY	2	1	—	$CY \leftarrow 0$			0
	NOT1	CY	2	1	—	$CY \leftarrow \overline{CY}$			×

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取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 30-5 操作一览表 (16/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
调用 / 返回	CALL	rp	2	3	—	(SP-2)←(PC+2) _S , (SP-3)←(PC+2) _H , (SP-4)←(PC+2) _L , PC←CS, rp, SP←SP-4			
		\$!addr20	3	3	—	(SP-2)←(PC+3) _S , (SP-3)←(PC+3) _H , (SP-4)←(PC+3) _L , PC←PC+3+jdisp16, SP←SP-4			
		!addr16	3	3	—	(SP-2)←(PC+3) _S , (SP-3)←(PC+3) _H , (SP-4)←(PC+3) _L , PC←0000, addr16, SP←SP-4			
		!!addr20	4	3	—	(SP-2)←(PC+4) _S , (SP-3)←(PC+4) _H , (SP-4)←(PC+4) _L , PC←addr20, SP←SP-4			
	CALLT	[addr5]	2	5	—	(SP-2)←(PC+2) _S , (SP-3)←(PC+2) _H , (SP-4)←(PC+2) _L , PC _S ←0000, PC _H ←(0000, addr5+1), PC _L ←(0000, addr5), SP←SP-4			
	BRK	—	2	5	—	(SP-1)←PSW, (SP-2)←(PC+2) _S , (SP-3)←(PC+2) _H , (SP-4)←(PC+2) _L , PC _S ←0000, PC _H ←(0007FH), PC _L ←(0007EH), SP←SP-4, IE←0			
	RET	—	1	6	—	PC _L ←(SP), PC _H ←(SP+1), PC _S ←(SP+2), SP←SP+4			
	RETI	—	2	6	—	PC _L ←(SP), PC _H ←(SP+1), PC _S ←(SP+2), PSW←(SP+3), SP←SP+4	R	R	R
	RETB	—	2	6	—	PC _L ←(SP), PC _H ←(SP+1), PC _S ←(SP+2), PSW←(SP+3), SP←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 30-5 操作一览表 (17/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
堆栈操作	PUSH	PSW	2	1	—	(SP-1)←PSW, (SP-2)←00H, SP←SP-2			
		rp	1	1	—	(SP-1)←rp _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 30-5 操作一览表 (18/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
条件转移	BF	saddr.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if (saddr).bit=0			
		sfr.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if sfr.bit=0			
		A.bit, \$addr20	3	3/5 注3	—	PC←PC+3+jdisp8 if A.bit=0			
		PSW.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if PSW.bit=0			
		[HL].bit, \$addr20	3	3/5 注3	6/7	PC←PC+3+jdisp8 if (HL).bit=0			
		ES:[HL].bit, \$addr20	4	4/6 注3	7/8	PC←PC+4+jdisp8 if (ES, HL).bit=0			
	BTCLR	saddr.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if (saddr).bit=1 then reset (saddr).bit			
		sfr.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if sfr.bit=1 then reset sfr.bit			
		A.bit, \$addr20	3	3/5 注3	—	PC←PC+3+jdisp8 if A.bit=1 then reset A.bit			
		PSW.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if PSW.bit=1 then reset PSW.bit	×	×	×
		[HL].bit, \$addr20	3	3/5 注3	—	PC←PC+3+jdisp8 if (HL).bit=1 then reset (HL).bit			
		ES:[HL].bit, \$addr20	4	4/6 注3	—	PC←PC+4+jdisp8 if (ES, HL).bit=1 then reset (ES, HL).bit			
条件跳转	SKC	—	2	1	—	Next instruction skip if CY=1			
	SKNC	—	2	1	—	Next instruction skip if CY=0			
	SKZ	—	2	1	—	Next instruction skip if Z=1			
	SKNZ	—	2	1	—	Next instruction skip if Z=0			
	SKH	—	2	1	—	Next instruction skip if (Z∨CY)=0			
	SKNH	—	2	1	—	Next instruction skip if (Z∨CY)=1			
CPU 控制	SEL 注4	RBn	2	1	—	RBS[1:0]←n			
	NOP	—	1	1	—	No Operation			
	EI	—	3	4	—	IE←1 (Enable Interrupt)			
	DI	—	3	4	—	IE←0 (Disable Interrupt)			
	HALT	—	2	3	—	Set HALT Mode			
	STOP	—	2	3	—	Set STOP Mode			

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取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 31 章 电特性

- 注意 1. RL78 微控制器内置用于开发和评估的片上调试功能。如果使用片上调试功能，就可能会超过闪存的保证改写次数而无法保证产品的可靠性，因此批量生产的产品不能使用片上调试功能。对于使用片上调试功能的产品，不作为投诉受理对象。
2. 对于没有 EV_{DD0} 引脚和 EV_{SS0} 引脚的产品，必须将 EV_{DD0} 替换为 V_{DD} 并且将 EV_{SS0} 替换为 V_{SS} 。
3. 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”。

31.1 绝对最大额定值

绝对最大额定值

(1/2)

项目	符号	条件	额定值	单位
电源电压	V_{DD}		$-0.5 \sim +6.5$	V
	EV_{DD0}		$-0.5 \sim +6.5$	V
	EV_{SS0}		$-0.5 \sim +0.3$	V
REGC 引脚的 输入电压	V_{IREGC}	REGC	$-0.3 \sim +2.8$ 并且 $-0.3 \sim V_{DD}+0.3$ 注 1	V
输入电压	V_{I1}	P00 ~ P06、P10 ~ P17、P30、P31、 P40 ~ P43、P50 ~ P55、P70 ~ P77、 P120、P140、P141、P146、P147	$-0.3 \sim EV_{DD0}+0.3$ 并且 $-0.3 \sim V_{DD}+0.3$ 注 2	V
	V_{I2}	P60 ~ P63 (N 沟道漏极开路)	$-0.3 \sim +6.5$	V
	V_{I3}	P20 ~ P27、P121 ~ P124、P137、 EXCLK、EXCLKS、RESET	$-0.3 \sim V_{DD}+0.3$ 注 2	V
输出电压	V_{O1}	P00 ~ P06、P10 ~ P17、P30、P31、 P40 ~ P43、P50 ~ P55、P60 ~ P63、 P70 ~ P77、P120、P130、P140、 P141、P146、P147	$-0.3 \sim EV_{DD0}+0.3$ 并且 $-0.3 \sim V_{DD}+0.3$ 注 2	V
	V_{O2}	P20 ~ P27	$-0.3 \sim V_{DD}+0.3$ 注 2	V
模拟输入电压	V_{AI1}	ANI16 ~ ANI19	$-0.3 \sim EV_{DD0}+0.3$ 并且 $-0.3 \sim AV_{REF(+)}+0.3$ 注 2、3	V
	V_{AI2}	ANI0 ~ ANI7	$-0.3 \sim V_{DD}+0.3$ 并且 $-0.3 \sim AV_{REF(+)}+0.3$ 注 2、3	V

注 1. 必须通过电容器 (0.47 ~ 1 μ F) 将 REGC 引脚连接 V_{SS} 。此值是规定 REGC 引脚的绝对最大额定值, 使用时不能外加电压。

2. 不超过 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

(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 ~ P06、P10 ~ P17、P30、P31、P40 ~ P43、P50 ~ P55、P70 ~ P77、P120、P130、P140、P141、P146、P147	-40	mA
		引脚合计 -170mA	P00 ~ P04、P40 ~ P43、P120、P130、P140、P141	-70	mA
			P05、P06、P10 ~ P17、P30、P31、P50 ~ P55、P70 ~ P77、P146、P147	-100	mA
	I _{OH2}	1 个引脚	P20 ~ P27	-0.5	mA
		引脚合计		-2	mA
低电平输出电流	I _{OL1}	1 个引脚	P00 ~ P06、P10 ~ P17、P30、P31、P40 ~ P43、P50 ~ P55、P60 ~ P63、P70 ~ P77、P120、P130、P140、P141、P146、P147	40	mA
		引脚合计 170mA	P00 ~ P04、P40 ~ P43、P120、P130、P140、P141	70	mA
			P05、P06、P10 ~ P17、P30、P31、P50 ~ P55、P60 ~ P63、P70 ~ P77、P146、P147	100	mA
	I _{OL2}	1 个引脚	P20 ~ P27	1	mA
		引脚合计		5	mA
工作环境温度	T _A	通常运行时		-40 ~ +85	°C
		进行闪存编程时			
保存温度	T _{stg}			-65 ~ +150	°C

注意 即使是各项目中的 1 个项目瞬间超过绝对最大额定值，也可能降低产品的质量。绝对最大额定值是可能给产品带来物理性损伤的额定值，必须在不超过额定值的状态下使用产品。

备注 在没有特别指定的情况下，复用引脚的特性和端口引脚的特性相同。

31.2 振荡电路特性

31.2.1 X1、XT1 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	
XT1 时钟振荡频率 (f_{XT}) 注	晶体谐振器		32	32.768	35	kHz

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、XT1 振荡电路时，请参照“5.4 系统时钟振荡电路”。

31.2.2 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		32	MHz
高速内部振荡器的时钟频率精度		$-20 \sim +85^\circ\text{C}$	$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$	-2.0		+2.0	%
			$1.6\text{V} \leq V_{DD} < 1.8\text{V}$	-5.0		+5.0	%
		$-40 \sim -20^\circ\text{C}$	$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$	-2.0		+2.0	%
			$1.6\text{V} \leq V_{DD} < 1.8\text{V}$	-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 ~ 4 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。

31.3 DC 特性

31.3.1 引脚特性

(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=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 ~ P06、P10 ~ P17、 P30、P31、P40 ~ P43、 P50 ~ P55、P70 ~ P77、 P120、P130、P140、 P141、P146、P147 1 个引脚	1.6V ≤ EV _{DD0} ≤ 5.5V			−10.0 ^{注2} mA
		P00 ~ P04、P40 ~ P43、 P120、P130、P140、 P141 合计 (占空比 ≤ 70% 时注3)	4.0V ≤ EV _{DD0} ≤ 5.5V			−55.0 mA
			2.7V ≤ EV _{DD0} < 4.0V			−10.0 mA
			1.8V ≤ EV _{DD0} < 2.7V			−5.0 mA
			1.6V ≤ EV _{DD0} < 1.8V			−2.5 mA
		P05、P06、P10 ~ P17、 P30、P31、P50 ~ P55、 P70 ~ P77、P146、P147 合计 (占空比 ≤ 70% 时注3)	4.0V ≤ EV _{DD0} ≤ 5.5V			−80.0 mA
			2.7V ≤ EV _{DD0} < 4.0V			−19.0 mA
			1.8V ≤ EV _{DD0} < 2.7V			−10.0 mA
			1.6V ≤ EV _{DD0} < 1.8V			−5.0 mA
		全部引脚合计 (占空比 ≤ 70% 时注3)	1.6V ≤ EV _{DD0} ≤ 5.5V			−135.0 mA
	I _{OH2}	P20 ~ P27 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			−1.5 mA

注 1. 这是即使电流从 EV_{DD0} 引脚和 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、P02 ~ P04、P10、P11、P13 ~ P15、P17、P30、P50、P51、P55、P71、P74 不输出高电平。

备注 在没有特别指定的情况下，复用引脚的特性和端口引脚的特性相同。

(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(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 ~ P06、P10 ~ P17、 P30、P31、P40 ~ P43、 P50 ~ P55、P70 ~ P77、 P120、P130、P140、 P141、P146、P147 1 个引脚			20.0 ^{注2}	mA
		P60 ~ P63 1 个引脚			15.0 ^{注2}	mA
		P00 ~ P04、P40 ~ P43、 P120、P130、P140、 P141 合计 (占空比 ≤ 70% 时注3)	4.0V ≤ EV _{DD0} ≤ 5.5V		70.0	mA
			2.7V ≤ EV _{DD0} < 4.0V		15.0	mA
			1.8V ≤ EV _{DD0} < 2.7V		9.0	mA
			1.6V ≤ EV _{DD0} < 1.8V		4.5	mA
		P05、P06、P10 ~ P17、 P30、P31、P50 ~ P55、 P60 ~ P63、P70 ~ P77、 P146、P147 合计 (占空比 ≤ 70% 时注3)	4.0V ≤ EV _{DD0} ≤ 5.5V		80.0	mA
			2.7V ≤ EV _{DD0} < 4.0V		35.0	mA
			1.8V ≤ EV _{DD0} < 2.7V		20.0	mA
			1.6V ≤ EV _{DD0} < 1.8V		10.0	mA
		全部引脚合计 (占空比 ≤ 70% 时注3)			150.0	mA
	I _{OL2}	P20 ~ P27 1 个引脚			0.4 ^{注2}	mA
		全部引脚合计 (占空比 ≤ 70% 时注3)	1.6V ≤ V _{DD} ≤ 5.5V		5.0	mA

注 1. 这是即使电流从 EV_{SS0} 引脚和 V_{SS} 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）。

$$\bullet \text{ 引脚合计的输出电流} = (I_{OL} \times 0.7) / (n \times 0.01)$$

<计算例子> I_{OL} = 10.0mA、n = 80%

$$\text{引脚合计的输出电流} = (10.0 \times 0.7) / (80 \times 0.01) \approx 8.7\text{mA}$$

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 ~ +85°C、1.6V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(3/5)

项目	符号	条件		MIN.	TYP.	MAX.	单位
高电平输入电压	V _{IH1}	P00 ~ P06、P10 ~ P17、P30、P31、P40 ~ P43、P50 ~ P55、P70 ~ P77、P120、P140、P141、P146、P147	通常的输入缓冲器	0.8EV _{DD0}		EV _{DD0}	V
	V _{IH2}	P01、P03、P04、P10、P14 ~ P17、P30、P31、P50、P55	TTL 输入缓冲器 4.0V≤EV _{DD0} ≤5.5V	2.2		EV _{DD0}	V
			TTL 输入缓冲器 3.3V≤EV _{DD0} <4.0V	2.0		EV _{DD0}	V
			TTL 输入缓冲器 1.6V≤EV _{DD0} <3.3V	1.5		EV _{DD0}	V
	V _{IH3}	P20 ~ P27		0.7V _{DD}		V _{DD}	V
	V _{IH4}	P60 ~ P63		0.7EV _{DD0}		6.0	V
	V _{IH5}	P121 ~ P124、P137、EXCLK、EXCLKS、RESET		0.8V _{DD}		V _{DD}	V
低电平输入电压	V _{IL1}	P00 ~ P06、P10 ~ P17、P30、P31、P40 ~ P43、P50 ~ P55、P70 ~ P77、P120、P140、P141、P146、P147	通常的输入缓冲器	0		0.2EV _{DD0}	V
	V _{IL2}	P01、P03、P04、P10、P14 ~ P17、P30、P31、P50、P55	TTL 输入缓冲器 4.0V≤EV _{DD0} ≤5.5V	0		0.8	V
			TTL 输入缓冲器 3.3V≤EV _{DD0} <4.0V	0		0.5	V
			TTL 输入缓冲器 1.6V≤EV _{DD0} <3.3V	0		0.32	V
	V _{IL3}	P20 ~ P27		0		0.3V _{DD}	V
	V _{IL4}	P60 ~ P63		0		0.3EV _{DD0}	V
	V _{IL5}	P121 ~ P124、P137、EXCLK、EXCLKS、RESET		0		0.2V _{DD}	V

注意 即使在 N 沟道漏极开路模式中，P00、P02 ~ P04、P10、P11、P13 ~ P15、P17、P30、P50、P51、P55、P71、P74 的 V_{IH} 最大值 (MAX.) 也为 EV_{DD0}。

备注 在没有特别指定的情况下，复用引脚的特性和端口引脚的特性相同。

(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(4/5)

项目	符号	条件	MIN.	TYP.	MAX.	单位
高电平输出电压	V _{OH1}	P00 ~ P06、P10 ~ P17、P30、P31、P40 ~ P43、P50 ~ P55、P70 ~ P77、P120、P130、P140、P141、P146、P147	4.0V ≤ EV _{DD0} ≤ 5.5V、I _{OH1} = −10.0mA	EV _{DD0} − 1.5		V
			4.0V ≤ EV _{DD0} ≤ 5.5V、I _{OH1} = −3.0mA	EV _{DD0} − 0.7		V
			1.8V ≤ EV _{DD0} ≤ 5.5V、I _{OH1} = −1.5mA	EV _{DD0} − 0.5		V
			1.6V ≤ EV _{DD0} ≤ 1.8V、I _{OH1} = −1.0mA	EV _{DD0} − 0.5		V
	V _{OH2}	P20 ~ P27	1.6V ≤ V _{DD} ≤ 5.5V、I _{OH2} = −100μA	V _{DD} − 0.5		V
低电平输出电压	V _{OL1}	P00 ~ P06、P10 ~ P17、P30、P31、P40 ~ P43、P50 ~ P55、P60 ~ P63、P70 ~ P77、P120、P130、P140、P141、P146、P147	4.0V ≤ EV _{DD0} ≤ 5.5V、I _{OL1} = 20.0mA		1.3	V
			4.0V ≤ EV _{DD0} ≤ 5.5V、I _{OL1} = 8.5mA		0.7	V
			2.7V ≤ EV _{DD0} ≤ 5.5V、I _{OL1} = 3.0mA		0.6	V
			2.7V ≤ EV _{DD0} ≤ 5.5V、I _{OL1} = 1.5mA		0.4	V
			1.8V ≤ EV _{DD0} ≤ 5.5V、I _{OL1} = 0.6mA		0.4	V
			1.6V ≤ EV _{DD0} ≤ 5.5V、I _{OL1} = 0.3mA		0.4	V
	V _{OL2}	P20 ~ P27	1.6V ≤ V _{DD} ≤ 5.5V、I _{OL2} = 400μA		0.4	V
	V _{OL3}	P60 ~ P63	4.0V ≤ EV _{DD0} ≤ 5.5V、I _{OL3} = 15.0mA		2.0	V
			4.0V ≤ EV _{DD0} ≤ 5.5V、I _{OL3} = 5.0mA		0.4	V
			2.7V ≤ EV _{DD0} ≤ 5.5V、I _{OL3} = 3.0mA		0.4	V
			1.8V ≤ EV _{DD0} ≤ 5.5V、I _{OL3} = 2.0mA		0.4	V
			1.6V ≤ EV _{DD0} ≤ 5.5V、I _{OL3} = 1.0mA		0.4	V

注意 在 N 沟道漏极开路模式中，P00、P02 ~ P04、P10、P11、P13 ~ P15、P17、P30、P50、P51、P55、P71、P74 不输出高电平。

备注 在没有特别指定的情况下，复用引脚的特性和端口引脚的特性相同。

(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(5/5)

项目	符号	条件	MIN.	TYP.	MAX.	单位
高电平输入漏电流	I _{LIH1}	P00 ~ P06、P10 ~ P17、P30、P31、P40 ~ P43、P50 ~ P55、P70 ~ P77、P120、P140、P141、P146、P147	V _I =EV _{DD0}			1 μA
	I _{LIH2}	P20 ~ P27、P137、RESET	V _I =V _{DD}			1 μA
	I _{LIH3}	P121 ~ P124 (X1、X2、EXCLK、XT1、XT2、EXCLKS)	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 ~ P06、P10 ~ P17、P30、P31、P40 ~ P43、P50 ~ P55、P70 ~ P77、P120、P140、P141、P146、P147	V _I =EV _{SS0}			-1 μA
	I _{LIL2}	P20 ~ P27、P137、RESET	V _I =V _{SS}			-1 μA
	I _{LIL3}	P121 ~ P124 (X1、X2、EXCLK、XT1、XT2、EXCLKS)	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 ~ P06、P10 ~ P17、P30、P31、P40 ~ P43、P50 ~ P55、P70 ~ P77、P120、P140、P141、P146、P147	V _I =EV _{SS0} ，输入端口时	10	20	100 kΩ

备注 在没有特别指定的情况下，复用引脚的特性和端口引脚的特性相同。

31.3.2 电源电流特性

(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=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5}	f _{HOCO} =64MHz	基本	V _{DD} =5.0V		2.6	mA
				f _{IH} =32MHz ^{注3}	运行	V _{DD} =3.0V		2.6	
				f _{HOCO} =32MHz	基本	V _{DD} =5.0V		2.3	
				f _{IH} =32MHz ^{注3}	运行	V _{DD} =3.0V		2.3	
			HS (高速主) 模式 ^{注5}	f _{HOCO} =64MHz	基本	V _{DD} =5.0V		5.8	mA
				f _{IH} =32MHz ^{注3}	运行	V _{DD} =3.0V		5.8	
				f _{HOCO} =32MHz	基本	V _{DD} =5.0V		5.4	
				f _{IH} =32MHz ^{注3}	运行	V _{DD} =3.0V		5.4	
				f _{HOCO} =48MHz	通常	V _{DD} =5.0V		4.5	
				f _{IH} =24MHz ^{注3}	运行	V _{DD} =3.0V		4.5	
				f _{HOCO} =24MHz	通常	V _{DD} =5.0V		4.2	
				f _{IH} =24MHz ^{注3}	运行	V _{DD} =3.0V		4.2	
			LS (低速主) 模式 ^{注5}	f _{HOCO} =8MHz	通常	V _{DD} =3.0V		1.4	mA
				f _{IH} =8MHz ^{注3}	运行	V _{DD} =2.0V		1.4	
			LV (低电压主) 模式 ^{注5}	f _{HOCO} =4MHz	通常	V _{DD} =3.0V		1.4	mA
				f _{IH} =4MHz ^{注3}	运行	V _{DD} =2.0V		1.4	
			HS (高速主) 模式 ^{注5}	f _{MX} =20MHz ^{注2}	通常	输入方波		3.7	mA
				V _{DD} =5.0V	运行	连接谐振器		3.9	
				f _{MX} =20MHz ^{注2}	通常	输入方波		3.7	
				V _{DD} =3.0V	运行	连接谐振器		3.9	
				f _{MX} =10MHz ^{注2}	通常	输入方波		2.2	
				V _{DD} =5.0V	运行	连接谐振器		2.3	
				f _{MX} =10MHz ^{注2}	通常	输入方波		2.2	
				V _{DD} =3.0V	运行	连接谐振器		2.3	
			LS (低速主) 模式 ^{注5}	f _{MX} =8MHz ^{注2}	通常	输入方波		1.3	mA
				V _{DD} =3.0V	运行	连接谐振器		1.3	
				f _{MX} =8MHz ^{注2}	通常	输入方波		1.3	
				V _{DD} =2.0V	运行	连接谐振器		1.3	
			副系统时钟运行	f _{SUB} =32.768kHz ^{注4}	通常	输入方波		5.0	μA
				T _A =−40°C	运行	连接谐振器		5.0	
				f _{SUB} =32.768kHz ^{注4}	通常	输入方波		5.0	
				T _A =+25°C	运行	连接谐振器		5.0	
				f _{SUB} =32.768kHz ^{注4}	通常	输入方波		5.1	
				T _A =+50°C	运行	连接谐振器		5.1	
				f _{SUB} =32.768kHz ^{注4}	通常	输入方波		5.5	
				T _A =+70°C	运行	连接谐振器		5.5	
				f _{SUB} =32.768kHz ^{注4}	通常	输入方波		6.5	μA
				T _A =+85°C	运行	连接谐振器		6.5	

- 注 1. 这是流过 V_{DD} 和 EV_{DD0} 的总电流，包含输入引脚固定为 V_{DD} 、 EV_{DD0} 或者 V_{SS} 、 EV_{SS0} 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和副系统时钟停止振荡的情况。
3. 这是高速系统时钟和副系统时钟停止振荡的情况。
4. 这是高速内部振荡器和高速系统时钟停止振荡的情况，或者超低消费振荡（AMPHS1=1）的情况。不包含流到 RTC、12 位间隔定时器和看门狗定时器的电流。
5. 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 | $2.7V \leq V_{DD} \leq 5.5V @ 1MHz \sim 32MHz$ |
| | $2.4V \leq V_{DD} \leq 5.5V @ 1MHz \sim 16MHz$ |
| LS（低速主）模式 | $1.8V \leq V_{DD} \leq 5.5V @ 1MHz \sim 8MHz$ |
| LV（低电压主）模式 | $1.6V \leq V_{DD} \leq 5.5V @ 1MHz \sim 4MHz$ |

- 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_{HOCO} : 高速内部振荡器的时钟频率 (最大 64MHz)
3. f_{IH} : 高速内部振荡器的时钟频率 (最大 32MHz)
4. f_{SUB} : 副系统时钟频率 (XT1 时钟振荡频率)
5. “副系统时钟运行” 以外的 TYP. 值的温度条件是 $T_A=25^{\circ}\text{C}$ 。

(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=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7}	f _{HOCO} =64MHz	V _{DD} =5.0V		0.88	3.32	mA		
				f _{IH} =32MHz ^{注4}	V _{DD} =3.0V		0.88	3.32			
				f _{HOCO} =32MHz	V _{DD} =5.0V		0.62	2.63			
				f _{IH} =32MHz ^{注4}	V _{DD} =3.0V		0.62	2.63			
				f _{HOCO} =48MHz	V _{DD} =5.0V		0.68	2.57			
				f _{IH} =24MHz ^{注4}	V _{DD} =3.0V		0.68	2.57			
				f _{HOCO} =24MHz	V _{DD} =5.0V		0.50	2.00			
				f _{IH} =24MHz ^{注4}	V _{DD} =3.0V		0.50	2.00			
				f _{HOCO} =16MHz	V _{DD} =5.0V		0.44	1.49			
				f _{IH} =16MHz ^{注4}	V _{DD} =3.0V		0.44	1.49			
			LS（低速主） 模式 ^{注7}	f _{HOCO} =8MHz	V _{DD} =3.0V		290	800	μA		
				f _{IH} =8MHz ^{注4}	V _{DD} =2.0V		290	800			
			LV（低电压主） 模式 ^{注7}	f _{HOCO} =4MHz	V _{DD} =3.0V		440	755	μA		
				f _{IH} =4MHz ^{注4}	V _{DD} =2.0V		440	755			
			HS（高速主） 模式 ^{注7}	f _{MX} =20MHz ^{注3} V _{DD} =5.0V	输入方波		0.37	1.63	mA		
					连接谐振器		0.45	1.85			
				f _{MX} =20MHz ^{注3} V _{DD} =3.0V	输入方波		0.37	1.63			
					连接谐振器		0.45	1.85			
				f _{MX} =10MHz ^{注3} V _{DD} =5.0V	输入方波		0.21	0.89			
					连接谐振器		0.26	0.97			
				f _{MX} =10MHz ^{注3} V _{DD} =3.0V	输入方波		0.21	0.89			
					连接谐振器		0.26	0.97			
				LS（低速主） 模式 ^{注7}	f _{MX} =8MHz ^{注3} V _{DD} =3.0V	输入方波		165		580	μA
						连接谐振器		185		630	
			f _{MX} =8MHz ^{注3} V _{DD} =2.0V		输入方波		165	580			
					连接谐振器		185	630			
			副系统时钟运行	f _{SUB} =32.768kHz ^{注5} T _A =−40°C	输入方波		0.28		μA		
					连接谐振器		0.47				
				f _{SUB} =32.768kHz ^{注5} T _A =+25°C	输入方波		0.34	0.66			
					连接谐振器		0.53	0.85			
				f _{SUB} =32.768kHz ^{注5} T _A =+50°C	输入方波		0.37	2.35			
					连接谐振器		0.56	2.54			
				f _{SUB} =32.768kHz ^{注5} T _A =+70°C	输入方波		0.61	4.08			
					连接谐振器		0.80	4.27			
				f _{SUB} =32.768kHz ^{注5} T _A =+85°C	输入方波		1.55	8.09			
					连接谐振器		1.74	8.28			
	I _{DD3} ^{注6}	STOP 模式 ^{注8}	T _A =−40°C					0.19	0.57	μA	
			T _A =+25°C					0.25	0.57		
			T _A =+50°C					0.28	2.26		
			T _A =+70°C					0.52	3.99		
			T _A =+85°C					1.46	8.00		

- 注
1. 这是流过 V_{DD} 和 EV_{DD0} 的总电流，包含输入引脚固定为 V_{DD} 、 EV_{DD0} 或者 V_{SS} 、 EV_{SS0} 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和副系统时钟停止振荡的情况。
 4. 这是高速系统时钟和副系统时钟停止振荡的情况。
 5. 这是高速内部振荡器和高速系统时钟停止振荡的情况，或者 RTCLPC 位为“1”并且超低消费振荡（AMPHS1=1）的情况。包含流到 RTC 的电流，但是不包含流到 12 位间隔定时器和看门狗定时器的电流。
 6. 不包含流到 RTC、12 位间隔定时器和看门狗定时器的电流。
 7. 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： $2.7V \leq V_{DD} \leq 5.5V@1MHz \sim 32MHz$
 $2.4V \leq V_{DD} \leq 5.5V@1MHz \sim 16MHz$
LS（低速主）模式： $1.8V \leq V_{DD} \leq 5.5V@1MHz \sim 8MHz$
LV（低电压主）模式： $1.6V \leq V_{DD} \leq 5.5V@1MHz \sim 4MHz$
 8. 有关 STOP 模式中副系统时钟运行时的电流值，请参照 HALT 模式中副系统时钟运行时的电流值。

- 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_{HOCO} : 高速内部振荡器的时钟频率 (最大 64MHz)
3. f_{IH} : 高速内部振荡器的时钟频率 (最大 32MHz)
4. f_{SUB} : 副系统时钟频率 (XT1 时钟振荡频率)
5. “副系统时钟运行”和“STOP 模式”以外的 TYP. 值的温度条件是 $T_A=25^{\circ}\text{C}$ 。

外围功能（适用于全部产品）

(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=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
RTC 工作电流	I _{RTC} 注 1、2、3				0.02		μA
12 位间隔定时器工作电流	I _{IT} 注 1、2、4				0.02		μA
看门狗定时器工作电流	I _{WDT} 注 1、2、5	f _{IL} =15kHz			0.22		μA
A/D 转换器工作电流	I _{ADC} 注 1、6	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
温度传感器工作电流	I _{TMPS} 注 1				75.0		μA
LVD 工作电流	I _{LVD} 注 1、7				0.08		μA
自编程工作电流	I _{FSP} 注 1、9				2.50	12.20	mA
BGO 电流	I _{BGO} 注 1、8				2.50	12.20	mA
SNOOZE 工作电流	I _{SNOZ} 注 1	ADC 运行	模式转移中注 10		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	
		DTC 运行			3.10		

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. 这是只流到实时时钟 (RTC) 的电流 (不包含低速内部振荡器和 XT1 振荡电路的工作电流)。在运行模式或者 HALT 模式中实时时钟运行的情况下, RL78 微控制器的电流值为 I_{DD1} 或者 I_{DD2} 加上 I_{RTC} 的值。另外, 当选择低速内部振荡器时, 必须加上 I_{FIL}。副系统时钟运行时的 I_{DD2} 包含实时时钟的工作电流。4. 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和 XT1 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}。5. 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。6. 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。7. 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。

8. 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“20.3.3 SNOOZE 模式”。

备注 1. f_{IL} : 低速内部振荡器的时钟频率2. f_{SUB} : 副系统时钟频率 (XT1 时钟振荡频率)3. f_{CLK} : CPU/ 外围硬件的时钟频率4. TYP. 值的温度条件是 T_A=25°C。

31.4 AC 特性

(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(1/2)

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
		副系统时钟 (f _{SUB}) 运行		1.8V ≤ V _{DD} ≤ 5.5V	28.5	30.5	31.3	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
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
	f _{EXS}				32		35	kHz
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
	t _{EXHS} 、 t _{EXLS}				13.7			μs
TI00 ~ TI03 输入的高低 电平宽度	t _{TIH} 、 t _{TIL}				1/f _{MCK} +10 注			ns
定时器 RJ 的输入周期	t _C	TRJIO	2.7V ≤ EV _{DD0} ≤ 5.5V	100				ns
			1.8V ≤ EV _{DD0} < 2.7V	300				ns
			1.6V ≤ EV _{DD0} < 1.8V	500				ns
定时器 RJ 输入的高低 电平宽度	t _{TJH} 、 t _{TJL}	TRJIO	2.7V ≤ EV _{DD0} ≤ 5.5V	40				ns
			1.8V ≤ EV _{DD0} < 2.7V	120				ns
			1.6V ≤ EV _{DD0} < 1.8V	200				ns

注 在 EV_{DD0} < V_{DD} 的低电压接口的情况下，需要满足以下条件：1.8V ≤ EV_{DD0} < 2.7V：MIN.125ns1.6V ≤ EV_{DD0} < 1.8V：MIN.250ns备注 f_{MCK}：定时器阵列单元的运行时钟频率

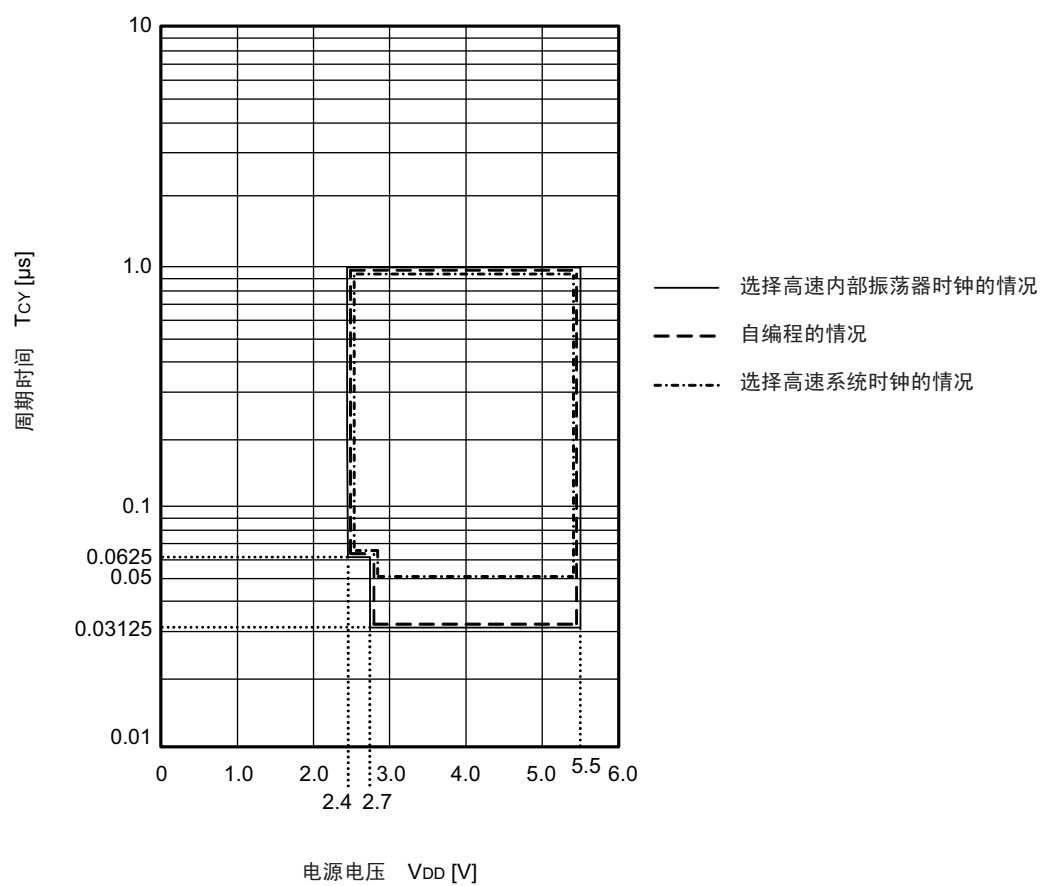
(这是定时器模式寄存器 mn (TMRmn) 的 CKSmn 位设定的运行时钟。m：单元号 (m=0) n：通道号 (n=0~3))

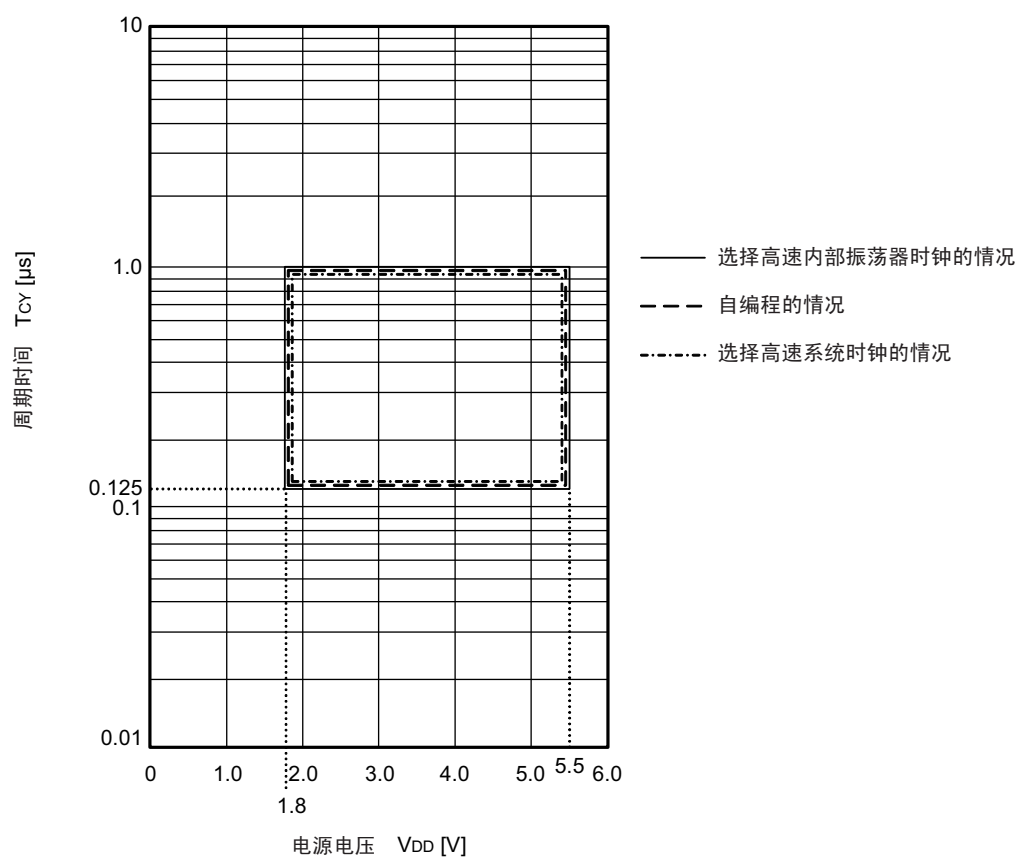
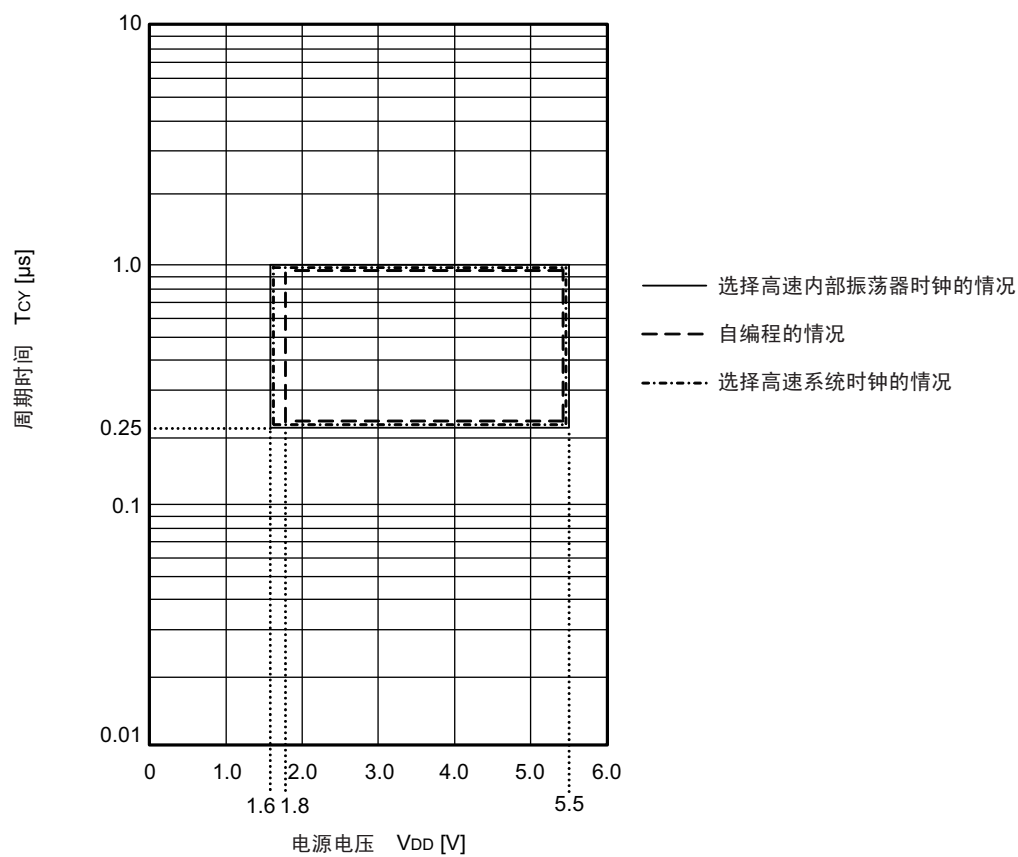
(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(2/2)

项目	符号	条件	MIN.	TYP.	MAX.	单位
定时器 RD 输入的高低电平宽度	t _{TDIH} 、 t _{TDIL}	TRDIOA0、TRDIOA1、TRDIOB0、TRDIOB1、 TRDIOC0、TRDIOC1、TRDIOD0、TRDIOD1	3/f _{CLK}			ns
定时器 RD 强制截止信号输入的低电平宽度	t _{TDSIL}	P130/INTP0	2MHz < f _{CLK} ≤ 32MHz f _{CLK} ≤ 2MHz	1 1/f _{CLK} + 1		μs
TO00 ~ TO03、 TRJIO0、TRJO0、 TRDIOA0、TRDIOA1、 TRDIOB0、TRDIOB1、 TRDIOC0、TRDIOC1、 TRDIOD0、TRDIOD1 的输出频率	f _{TO}	HS (高速主) 模式	4.0V ≤ EV _{DD0} ≤ 5.5V		16	MHz
			2.7V ≤ EV _{DD0} < 4.0V		8	MHz
			1.8V ≤ EV _{DD0} < 2.7V		4	MHz
			1.6V ≤ EV _{DD0} < 1.8V		2	MHz
		LS (低速主) 模式	1.8V ≤ EV _{DD0} ≤ 5.5V		4	MHz
			1.6V ≤ EV _{DD0} < 1.8V		2	MHz
		LV (低电压主) 模式	1.6V ≤ EV _{DD0} ≤ 5.5V		2	MHz
PCLBUZ0、PCLBUZ1 的输出频率	f _{PCL}	HS (高速主) 模式	4.0V ≤ EV _{DD0} ≤ 5.5V		16	MHz
			2.7V ≤ EV _{DD0} < 4.0V		8	MHz
			1.8V ≤ EV _{DD0} < 2.7V		4	MHz
			1.6V ≤ EV _{DD0} < 1.8V		2	MHz
		LS (低速主) 模式	1.8V ≤ EV _{DD0} ≤ 5.5V		4	MHz
			1.6V ≤ EV _{DD0} < 1.8V		2	MHz
		LV (低电压主) 模式	1.8V ≤ EV _{DD0} ≤ 5.5V 1.6V ≤ EV _{DD0} < 1.8V		4 2	MHz 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 ~ INTP11	1.6V ≤ EV _{DD0} ≤ 5.5V	1		μs
键中断输入的高低电平宽度	t _{KR}	KR0 ~ KR7	1.8V ≤ EV _{DD0} ≤ 5.5V	250		ns
			1.6V ≤ EV _{DD0} < 1.8V	1		μs
RESET 的低电平宽度	t _{RSL}		10			μs

主系统时钟运行时的最短指令执行时间

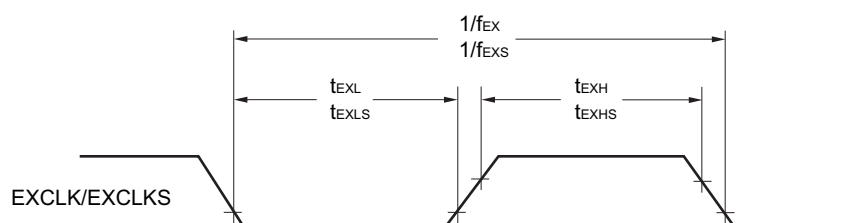
 T_{CY} vs V_{DD} (HS (高速主) 模式)

T_{CY} vs V_{DD} (LS (低速主) 模式) T_{CY} vs V_{DD} (LV (低电压主) 模式)

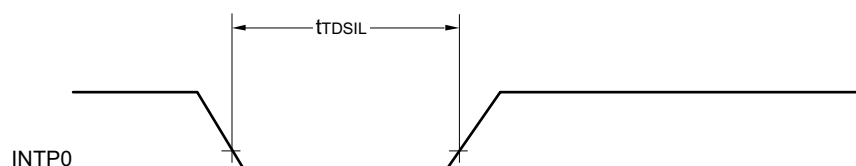
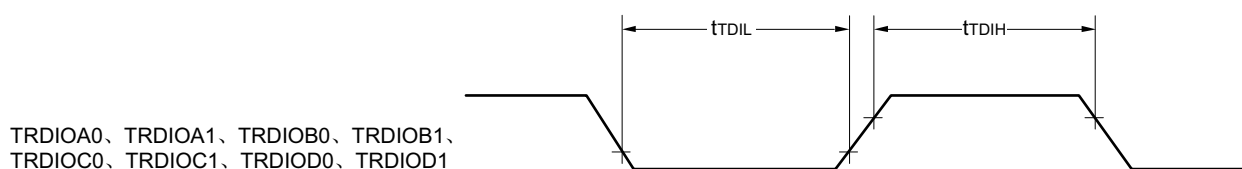
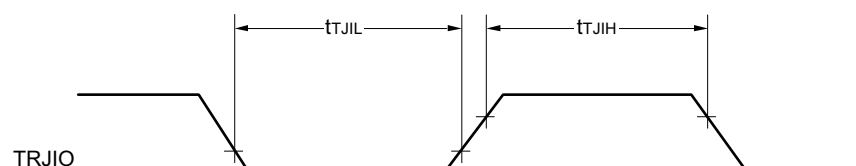
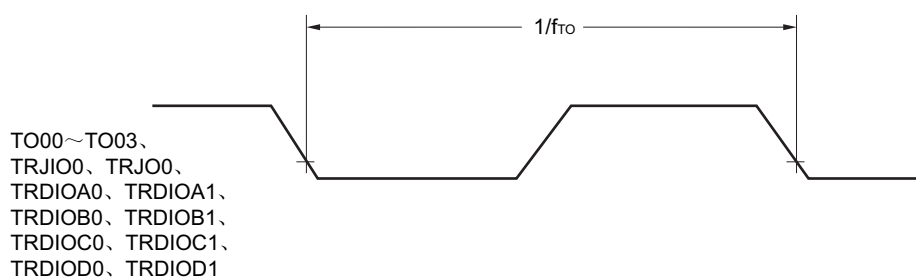
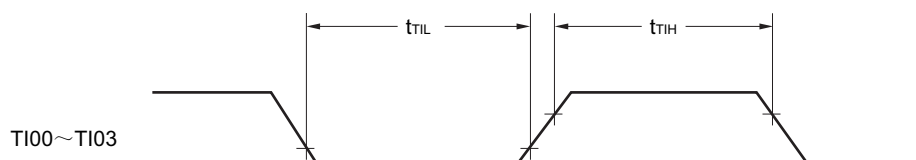
AC 时序测量点



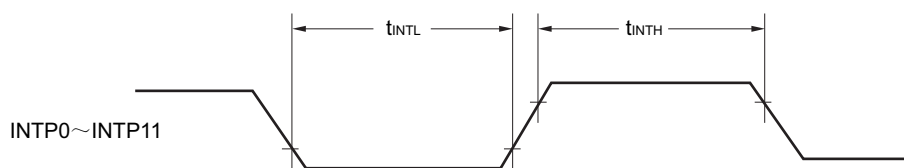
外部系统时钟的时序



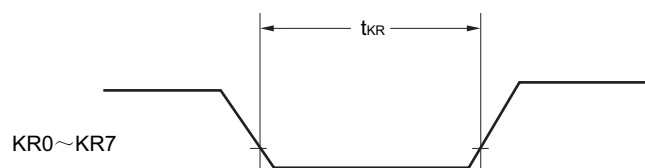
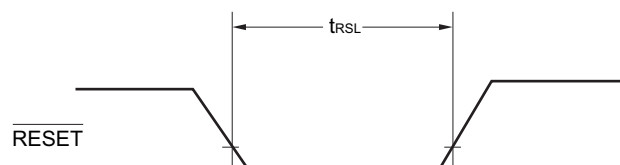
TI/TO 时序



中断请求的输入时序

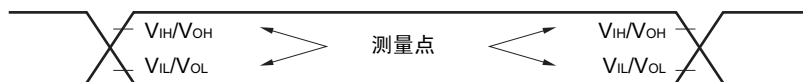


键中断的输入时序

 $\overline{\text{RESET}}$ 的输入时序

31.5 外围功能特性

AC 时序测量点



31.5.1 串行阵列单元

(1) 同电位的通信 (UART 模式)

($T_A = -40 \sim +85^\circ\text{C}$ 、 $1.6\text{V} \leq \text{EV}_{\text{DD0}} \leq V_{\text{DD}} \leq 5.5\text{V}$ 、 $V_{\text{SS}} = \text{EV}_{\text{SS0}} = 0\text{V}$)

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.	
传送速率注 1		$2.4\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$		$f_{\text{MCK}}/6$ 注2		$f_{\text{MCK}}/6$		$f_{\text{MCK}}/6$	bps
		最大传送速率的理论值 $f_{\text{MCK}} = f_{\text{CLK}}$ 注3		5.3		1.3		0.6	Mbps
		$1.8\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$		$f_{\text{MCK}}/6$ 注2		$f_{\text{MCK}}/6$		$f_{\text{MCK}}/6$	bps
		最大传送速率的理论值 $f_{\text{MCK}} = f_{\text{CLK}}$ 注3		5.3		1.3		0.6	Mbps
		$1.7\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$		$f_{\text{MCK}}/6$ 注2		$f_{\text{MCK}}/6$ 注2		$f_{\text{MCK}}/6$	bps
		最大传送速率的理论值 $f_{\text{MCK}} = f_{\text{CLK}}$ 注3		5.3		1.3		0.6	Mbps
		$1.6\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$	—			$f_{\text{MCK}}/6$ 注2		$f_{\text{MCK}}/6$	bps
		最大传送速率的理论值 $f_{\text{MCK}} = f_{\text{CLK}}$ 注3	—			1.3		0.6	Mbps

注 1. SNOOZE 模式中的传送速率只为 4800bps。

但是, 在 FRQSEL4 位为“1”时, 不能使用 SNOOZE 模式。

2. 在 $\text{EV}_{\text{DD0}} < V_{\text{DD}}$ 的低电压接口的情况下, 需要满足以下条件:

$2.4\text{V} \leq \text{EV}_{\text{DD0}} < 2.7\text{V}$: MAX.2.6Mbps

$1.8\text{V} \leq \text{EV}_{\text{DD0}} < 2.4\text{V}$: MAX.1.3Mbps

$1.6\text{V} \leq \text{EV}_{\text{DD0}} < 1.8\text{V}$: MAX.0.6Mbps

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 : 32MHz ($2.7\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$)

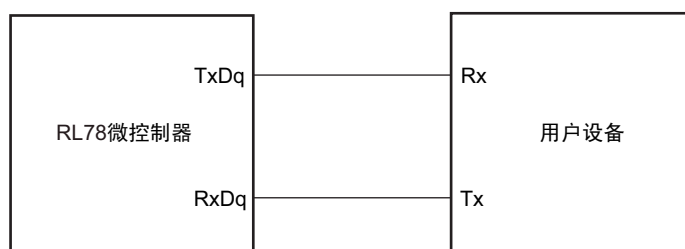
16MHz ($2.4\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$)

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

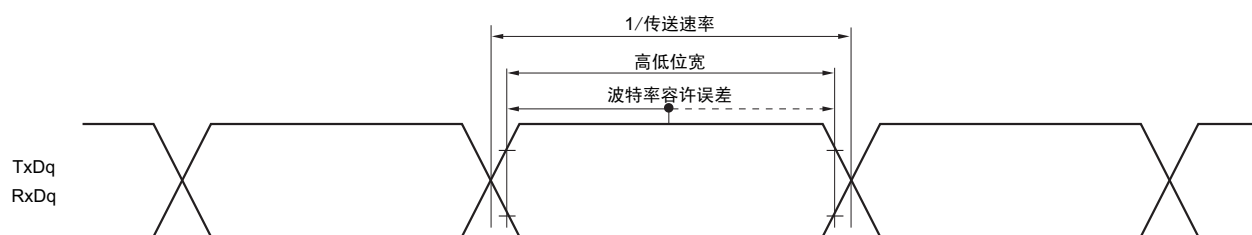
LV (低电压主) 模式 : 4MHz ($1.6\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$)

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、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、11)）

(2) 同电位的通信 (CSI 模式) (主控模式, SCKp..... 内部时钟输出, 只对应 CSI00)

($T_A = -40 \sim +85^\circ\text{C}$ 、 $2.7\text{V} \leq \text{EV}_{\text{DD0}} \leq \text{V}_{\text{DD}} \leq 5.5\text{V}$ 、 $\text{V}_{\text{SS}} = \text{EV}_{\text{SS0}} = 0\text{V}$)

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.	
SCKp 周期 时间	t_{KCY1}	$t_{\text{KCY1}} \geq 2/f_{\text{CLK}}$	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$	62.5		250		500		ns
			$2.7\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$	83.3		250		500		ns
SCKp 高低 电平宽度	t_{KH1} 、 t_{KL1}	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$		$t_{\text{KCY1}}/2$ -7		$t_{\text{KCY1}}/2$ -50		$t_{\text{KCY1}}/2$ -50		ns
		$2.7\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$		$t_{\text{KCY1}}/2$ -10		$t_{\text{KCY1}}/2$ -50		$t_{\text{KCY1}}/2$ -50		ns
Slp 准备时间 (对 SCKp↑) 注 1	t_{SIK1}	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$		23		110		110		ns
		$2.7\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$		33		110		110		ns
Slp 保持时间 (对 SCKp↑) 注 1	t_{KSI1}	$2.7\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$		10		10		10		ns
SCKp↓→SOp 输出延迟时间 注 2	t_{KSO1}	$C=20\text{pF}$ 注 3			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. 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=3、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))

(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 ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=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 ≤ EV _{DD0} ≤ 5.5V	125		500		1000	ns
			2.4V ≤ EV _{DD0} ≤ 5.5V	250		500		1000	ns
			1.8V ≤ EV _{DD0} ≤ 5.5V	500		500		1000	ns
			1.7V ≤ EV _{DD0} ≤ 5.5V	1000		1000		1000	ns
			1.6V ≤ EV _{DD0} ≤ 5.5V	—		1000		1000	ns
SCKp 高低 电平宽度	t _{KH1} 、 t _{KL1}	4.0V ≤ EV _{DD0} ≤ 5.5V	t _{KCY1} /2 −12		t _{KCY1} /2 −50		t _{KCY1} /2 −50		ns
		2.7V ≤ EV _{DD0} ≤ 5.5V	t _{KCY1} /2 −18		t _{KCY1} /2 −50		t _{KCY1} /2 −50		ns
		2.4V ≤ EV _{DD0} ≤ 5.5V	t _{KCY1} /2 −38		t _{KCY1} /2 −50		t _{KCY1} /2 −50		ns
		1.8V ≤ EV _{DD0} ≤ 5.5V	t _{KCY1} /2 −50		t _{KCY1} /2 −50		t _{KCY1} /2 −50		ns
		1.7V ≤ EV _{DD0} ≤ 5.5V	t _{KCY1} /2 −100		t _{KCY1} /2 −100		t _{KCY1} /2 −100		ns
		1.6V ≤ EV _{DD0} ≤ 5.5V	—		t _{KCY1} /2 −100		t _{KCY1} /2 −100		ns
Slp 准备时间 (对 SCKp↑) 注 1	t _{SIK1}	4.0V ≤ EV _{DD0} ≤ 5.5V	44		110		110		ns
		2.7V ≤ EV _{DD0} ≤ 5.5V	44		110		110		ns
		2.4V ≤ EV _{DD0} ≤ 5.5V	75		110		110		ns
		1.8V ≤ EV _{DD0} ≤ 5.5V	110		110		110		ns
		1.7V ≤ EV _{DD0} ≤ 5.5V	220		220		220		ns
		1.6V ≤ EV _{DD0} ≤ 5.5V	—		220		220		ns
Slp 保持时间 (对 SCKp↑) 注 1	t _{KSI1}	1.7V ≤ EV _{DD0} ≤ 5.5V	19		19		19		ns
		1.6V ≤ EV _{DD0} ≤ 5.5V	—		19		19		ns
SCKp↓→SOp 输出延迟时间 注 2	t _{KSO1}	1.7V ≤ EV _{DD0} ≤ 5.5V C=30pF 注 3		25		25		25	ns
		1.6V ≤ EV _{DD0} ≤ 5.5V C=30pF 注 3		—		25		25	ns

注 1. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↓”。

2. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↑”。

3. C 是 SCKp、SOp 输出线的负载电容。

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、01、10、11、20、21) m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)
g: PIM、POM 号 (g=0、1、3、5、7)

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、11))

(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 ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(1/2)

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 4	t _{KCY2}	4.0V ≤ EV _{DD0} ≤ 5.5V	20MHz < f _{MCK}	8/f _{MCK}		—		—		ns
			f _{MCK} ≤ 20MHz	6/f _{MCK}		6/f _{MCK}		6/f _{MCK}		ns
		2.7V ≤ EV _{DD0} ≤ 5.5V	16MHz < f _{MCK}	8/f _{MCK}		—		—		ns
			f _{MCK} ≤ 16MHz	6/f _{MCK}		6/f _{MCK}		6/f _{MCK}		ns
		2.4V ≤ EV _{DD0} ≤ 5.5V		6/f _{MCK} 并且 500		6/f _{MCK} 并且 500		6/f _{MCK} 并且 500		ns
		1.8V ≤ EV _{DD0} ≤ 5.5V		6/f _{MCK} 并且 750		6/f _{MCK} 并且 750		6/f _{MCK} 并且 750		ns
		1.7V ≤ EV _{DD0} ≤ 5.5V		6/f _{MCK} 并且 1500		6/f _{MCK} 并且 1500		6/f _{MCK} 并且 1500		ns
		1.6V ≤ EV _{DD0} ≤ 5.5V		—		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 ≤ EV _{DD0} ≤ 5.5V		t _{KCY2} /2 −7		t _{KCY2} /2 −7		t _{KCY2} /2 −7		ns
		2.7V ≤ EV _{DD0} ≤ 5.5V		t _{KCY2} /2 −8		t _{KCY2} /2 −8		t _{KCY2} /2 −8		ns
		1.8V ≤ EV _{DD0} ≤ 5.5V		t _{KCY2} /2 −18		t _{KCY2} /2 −18		t _{KCY2} /2 −18		ns
		1.7V ≤ EV _{DD0} ≤ 5.5V		t _{KCY2} /2 −66		t _{KCY2} /2 −66		t _{KCY2} /2 −66		ns
		1.6V ≤ EV _{DD0} ≤ 5.5V		—		t _{KCY2} /2 −66		t _{KCY2} /2 −66		ns
Slp 准备时间 (对 SCKp↑) 注 1	t _{SIK2}	2.7V ≤ EV _{DD0} ≤ 5.5V		1/f _{MCK} +20		1/f _{MCK} +30		1/f _{MCK} +30		ns
		1.8V ≤ EV _{DD0} ≤ 5.5V		1/f _{MCK} +30		1/f _{MCK} +30		1/f _{MCK} +30		ns
		1.7V ≤ EV _{DD0} ≤ 5.5V		1/f _{MCK} +40		1/f _{MCK} +40		1/f _{MCK} +40		ns
		1.6V ≤ EV _{DD0} ≤ 5.5V		—		1/f _{MCK} +40		1/f _{MCK} +40		ns
Slp 保持时间 (对 SCKp↑) 注 1	t _{KSI2}	1.8V ≤ EV _{DD0} ≤ 5.5V		1/f _{MCK} +31		1/f _{MCK} +31		1/f _{MCK} +31		ns
		1.7V ≤ EV _{DD0} ≤ 5.5V		1/f _{MCK} +250		1/f _{MCK} +250		1/f _{MCK} +250		ns
		1.6V ≤ EV _{DD0} ≤ 5.5V		—		1/f _{MCK} +250		1/f _{MCK} +250		ns
SCKp↓→SOp 输出延迟时间 注 2	t _{KSO2}	C=30pF 注 3	2.7V ≤ EV _{DD0} ≤ 5.5V		2/f _{MCK} +44		2/f _{MCK} +110		2/f _{MCK} +110	ns
			2.4V ≤ EV _{DD0} ≤ 5.5V		2/f _{MCK} +75		2/f _{MCK} +110		2/f _{MCK} +110	ns
			1.8V ≤ EV _{DD0} ≤ 5.5V		2/f _{MCK} +100		2/f _{MCK} +110		2/f _{MCK} +110	ns
			1.7V ≤ EV _{DD0} ≤ 5.5V		2/f _{MCK} +220		2/f _{MCK} +220		2/f _{MCK} +220	ns
			1.6V ≤ EV _{DD0} ≤ 5.5V		—		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. C 是 SOp 输出线的负载电容。
4. 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、01、10、11、20、21) m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)
g: PIM、POM 号 (g=0、1、3、5、7)
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、11))

(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 ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

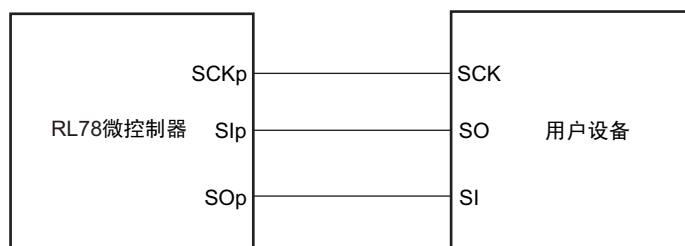
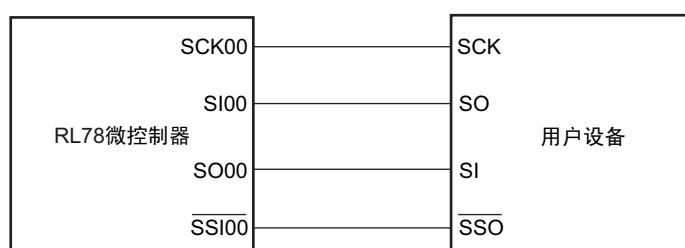
(2/2)

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.	
SSI00 准备时间	t _{SSIK}	DAPmn=0	2.7V ≤ EV _{DD0} ≤ 5.5V	120		120		120		ns
			1.8V ≤ EV _{DD0} ≤ 5.5V	200		200		200		ns
			1.7V ≤ EV _{DD0} ≤ 5.5V	400		400		400		ns
			1.6V ≤ EV _{DD0} ≤ 5.5V	—		400		400		ns
		DAPmn=1	2.7V ≤ EV _{DD0} ≤ 5.5V	1/f _{MCK} +120		1/f _{MCK} +120		1/f _{MCK} +120		ns
			1.8V ≤ EV _{DD0} ≤ 5.5V	1/f _{MCK} +200		1/f _{MCK} +200		1/f _{MCK} +200		ns
			1.7V ≤ EV _{DD0} ≤ 5.5V	1/f _{MCK} +400		1/f _{MCK} +400		1/f _{MCK} +400		ns
			1.6V ≤ EV _{DD0} ≤ 5.5V	—		1/f _{MCK} +400		1/f _{MCK} +400		ns
SSI00 保持时间	t _{kSSI}	DAPmn=0	2.7V ≤ EV _{DD0} ≤ 5.5V	1/f _{MCK} +120		1/f _{MCK} +120		1/f _{MCK} +120		ns
			1.8V ≤ EV _{DD0} ≤ 5.5V	1/f _{MCK} +200		1/f _{MCK} +200		1/f _{MCK} +200		ns
			1.7V ≤ EV _{DD0} ≤ 5.5V	1/f _{MCK} +400		1/f _{MCK} +400		1/f _{MCK} +400		ns
			1.6V ≤ EV _{DD0} ≤ 5.5V	—		1/f _{MCK} +400		1/f _{MCK} +400		ns
		DAPmn=1	2.7V ≤ EV _{DD0} ≤ 5.5V	120		120		120		ns
			1.8V ≤ EV _{DD0} ≤ 5.5V	200		200		200		ns
			1.7V ≤ EV _{DD0} ≤ 5.5V	400		400		400		ns
			1.6V ≤ EV _{DD0} ≤ 5.5V	—		400		400		ns

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 p: CSI 号 (p=00) m: 单元号 (m=0) n: 通道号 (n=0) g: PIM、POM 号 (g=3、5)

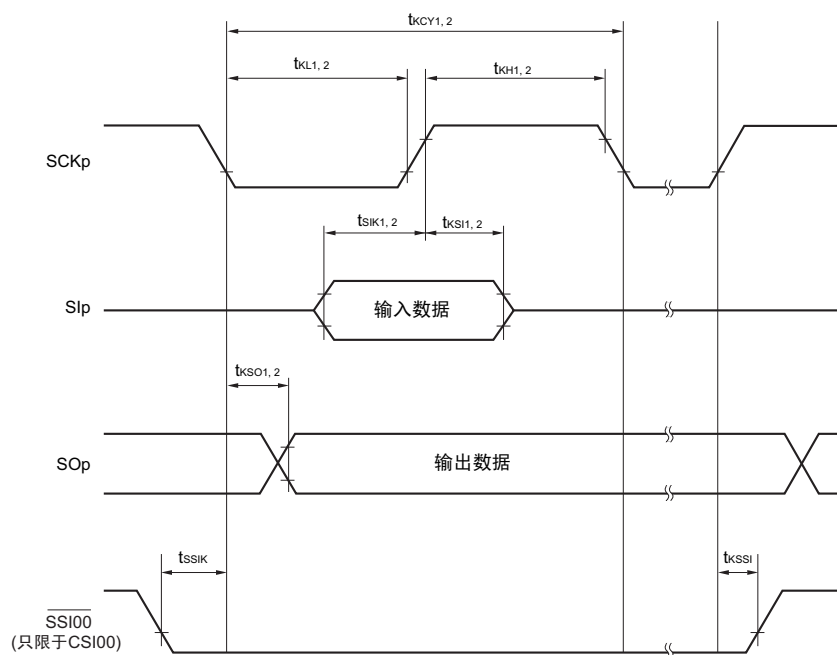
CSI模式的连接图（同电位的通信）


 CSI模式的连接图（同电位的通信）
 （从属选择输入功能（CSI00）的从属发送）


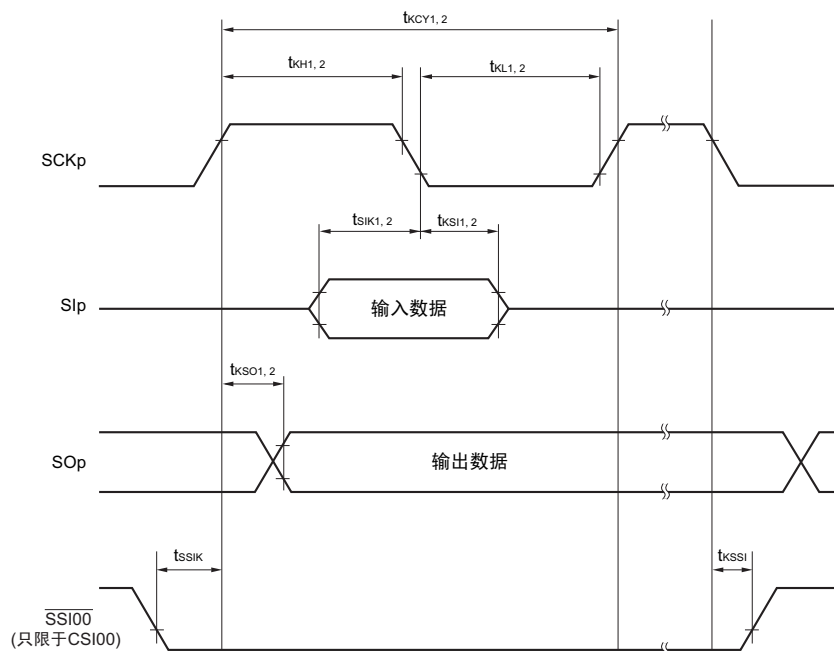
备注 1. p: CSI 号 (p=00、01、10、11、20、21)

2. m: 单元号 n: 通道号 (mn=00 ~ 03、10、11)

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、01、10、11、20、21)

2. m: 单元号 n: 通道号 (mn=00 ~ 03、10、11)

(5) 同电位的通信 (简易 I²C 模式)(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(1/2)

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 ≤ EV _{DD0} ≤ 5.5V、 C _b =50pF、R _b =2.7kΩ		1000 注 1		400 注 1		400 注 1	kHz
		1.8V ≤ EV _{DD0} ≤ 5.5V、 C _b =100pF、R _b =3kΩ		400 注 1		400 注 1		400 注 1	kHz
		1.8V ≤ EV _{DD0} < 2.7V、 C _b =100pF、R _b =5kΩ		300 注 1		300 注 1		300 注 1	kHz
		1.7V ≤ EV _{DD0} < 1.8V、 C _b =100pF、R _b =5kΩ		250 注 1		250 注 1		250 注 1	kHz
		1.6V ≤ EV _{DD0} < 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 ≤ EV _{DD0} ≤ 5.5V、 C _b =50pF、R _b =2.7kΩ	475		1150		1150		ns
		1.8V ≤ EV _{DD0} ≤ 5.5V、 C _b =100pF、R _b =3kΩ	1150		1150		1150		ns
		1.8V ≤ EV _{DD0} < 2.7V、 C _b =100pF、R _b =5kΩ	1550		1550		1550		ns
		1.7V ≤ EV _{DD0} < 1.8V、 C _b =100pF、R _b =5kΩ	1850		1850		1850		ns
		1.6V ≤ EV _{DD0} < 1.8V、 C _b =100pF、R _b =5kΩ	—		1850		1850		ns
SCLr="H" 的 保持时间	t _{HIGH}	2.7V ≤ EV _{DD0} ≤ 5.5V、 C _b =50pF、R _b =2.7kΩ	475		1150		1150		ns
		1.8V ≤ EV _{DD0} ≤ 5.5V、 C _b =100pF、R _b =3kΩ	1150		1150		1150		ns
		1.8V ≤ EV _{DD0} < 2.7V、 C _b =100pF、R _b =5kΩ	1550		1550		1550		ns
		1.7V ≤ EV _{DD0} < 1.8V、 C _b =100pF、R _b =5kΩ	1850		1850		1850		ns
		1.6V ≤ EV _{DD0} < 1.8V、 C _b =100pF、R _b =5kΩ	—		1850		1850		ns

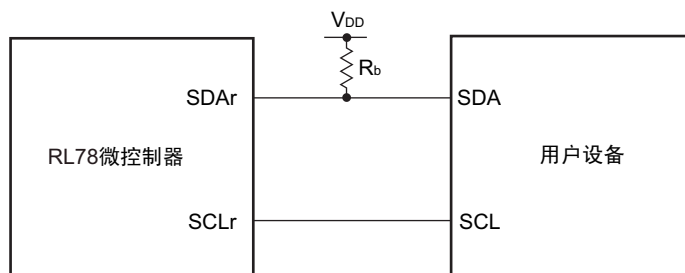
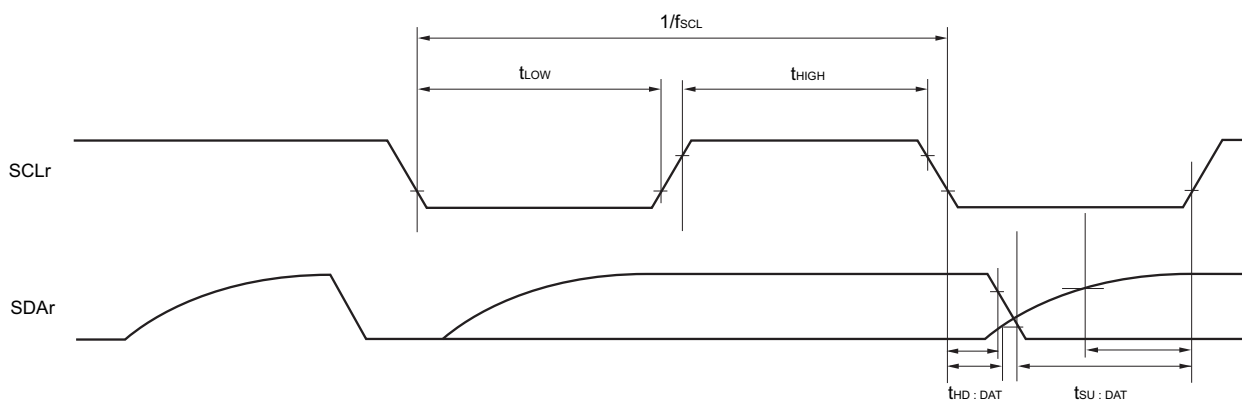
(5) 同电位的通信 (简易 I²C 模式)(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(2/2)

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}	2.7V ≤ EV _{DD0} ≤ 5.5V、 C _b =50pF、R _b =2.7kΩ	1/f _{MCK} +85 注 2		1/f _{MCK} +145 注 2		1/f _{MCK} +145 注 2		ns
		1.8V ≤ EV _{DD0} ≤ 5.5V、 C _b =100pF、R _b =3kΩ	1/f _{MCK} +145 注 2		1/f _{MCK} +145 注 2		1/f _{MCK} +145 注 2		ns
		1.8V ≤ EV _{DD0} < 2.7V、 C _b =100pF、R _b =5kΩ	1/f _{MCK} +230 注 2		1/f _{MCK} +230 注 2		1/f _{MCK} +230 注 2		ns
		1.7V ≤ EV _{DD0} < 1.8V、 C _b =100pF、R _b =5kΩ	1/f _{MCK} +290 注 2		1/f _{MCK} +290 注 2		1/f _{MCK} +290 注 2		ns
		1.6V ≤ EV _{DD0} < 1.8V、 C _b =100pF、R _b =5kΩ	—		1/f _{MCK} +290 注 2		1/f _{MCK} +290 注 2		ns
数据保持时间 (发送时)	t _{HD: DAT}	2.7V ≤ EV _{DD0} ≤ 5.5V、 C _b =50pF、R _b =2.7kΩ	0	305	0	305	0	305	ns
		1.8V ≤ EV _{DD0} ≤ 5.5V、 C _b =100pF、R _b =3kΩ	0	355	0	355	0	355	ns
		1.8V ≤ EV _{DD0} < 2.7V、 C _b =100pF、R _b =5kΩ	0	405	0	405	0	405	ns
		1.7V ≤ EV _{DD0} < 1.8V、 C _b =100pF、R _b =5kΩ	0	405	0	405	0	405	ns
		1.6V ≤ EV _{DD0} < 1.8V、 C _b =100pF、R _b =5kΩ	—		0	405	0	405	ns

注 1. 并且必须至少设定为 f_{MCK}/4。2. f_{MCK} 的设定值不能超过 SCLr="L" 和 SCLr="H" 的保持时间。

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 (32 引脚产品) / EV_{DD} 耐压 (64 引脚产品)) 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、01、10、11、20、21) g: PIM 号 (g=0、1、5、7)

h: POM 号 (h=0、1、3、7)

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、11)

(6) 不同电位 (1.8V、2.5V、3V) 的通信 (UART 模式)

(T_A=−40 ~ +85°C、1.8V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(1/2)

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 ≤ EV _{DD0} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V		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4}	5.3		1.3		0.6	Mbps
			2.7V ≤ EV _{DD0} < 4.0V、 2.3V ≤ V _b ≤ 2.7V		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4}	5.3		1.3		0.6	Mbps
			1.8V ≤ EV _{DD0} < 3.3V、 1.6V ≤ V _b ≤ 2.0V		f _{MCK} /6 ^{注1、2、3}		f _{MCK} /6 ^{注1、2}		bps
			最大传送速率的理论值 f _{MCK} =f _{CLK} ^{注4}	5.3		1.3		0.6	Mbps

注 1. SNOOZE 模式中的传送速率只为 4800bps。

但是, 在 FRQSEL4 位为“1”时, 不能使用 SNOOZE 模式。

2. 必须在 EV_{DD0} ≥ V_b 的情况下使用。
3. 在 EV_{DD0} < V_{DD} 的低电压接口的情况下, 需要满足以下条件:
2.4V ≤ EV_{DD0} < 2.7V: MAX.2.6Mbps
1.8V ≤ EV_{DD0} < 2.4V: MAX.1.3Mbps
4. CPU/ 外围硬件时钟 (f_{CLK}) 的最大工作频率如下所示:
HS (高速主) 模式 : 32MHz (2.7V ≤ 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 (32 引脚产品) /EV_{DD} 耐压 (64 引脚产品)) 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、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 ~ 03、10、11))
4. 当外围 I/O 重定向寄存器 0 (PIOR0) 的 bit1 (PIOR01) 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)

(T_A=−40 ~ +85°C、1.8V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(2/2)

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 ≤ EV _{DD0} ≤ 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 ≤ EV _{DD0} < 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 ≤ EV _{DD0} < 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 ≤ EV_{DD0} ≤ 5.5V、2.7V ≤ V_b ≤ 4.0V 时的传送速率计算式：

$$\text{最大传送速率} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{波特率容许误差 (理论值)} = \frac{\frac{1}{\text{传送速率} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{传送速率}}) \times \text{传送位数}} \times 100 \quad [\%]$$

※ 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 ≤ EV_{DD0} < 4.0V、2.3V ≤ V_b ≤ 2.7V 时的传送速率计算式：

$$\text{最大传送速率} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{波特率容许误差 (理论值)} = \frac{\frac{1}{\text{传送速率} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{传送速率}}) \times \text{传送位数}} \times 100 \quad [\%]$$

※ 此值为发送方和接收方的相对误差的理论值。

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 EV_{DD0} ≥ V_b 的情况下使用。

注 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。

$1.8V \leq EV_{DD0} < 3.3V$ 、 $1.6V \leq V_b \leq 2.0V$ 时的传送速率计算式：

$$\text{最大传送速率} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

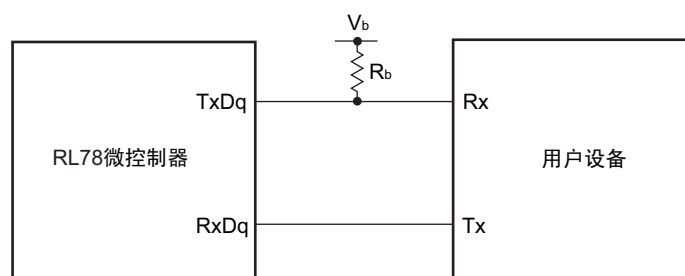
$$\text{波特率容许误差 (理论值)} = \frac{\frac{1}{\text{传送速率} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{传送速率}}) \times \text{传送位数}} \times 100 \quad [\%]$$

※ 此值为发送方和接收方的相对误差的理论值。

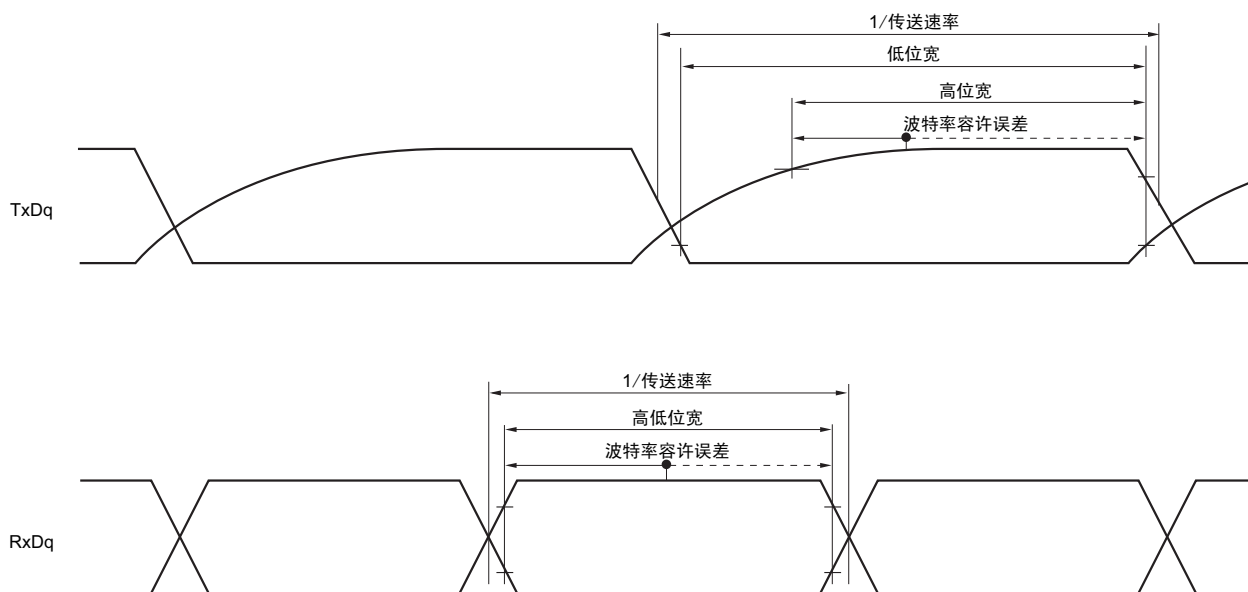
7. 作为例子，此值是满足条件栏中条件的计算值。有关用户条件下的最大传送速率，必须根据注 6 进行计算。

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 (32 引脚产品) / EV_{DD} 耐压 (64 引脚产品)) 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、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 ~ 03、10、11))
4. 当外围 I/O 重定向寄存器 0 (PIOR0) 的 bit1 (PIOR01) 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)

 $(T_A = -40 \sim +85^\circ\text{C}, 2.7\text{V} \leq \text{EV}_{\text{DD0}} \leq V_{\text{DD}} \leq 5.5\text{V}, V_{\text{SS}} = \text{EV}_{\text{SS0}} = 0\text{V})$

(1/2)

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}	$t_{\text{KCY1}} \geq 2/f_{\text{CLK}}$	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$ 、 $2.7\text{V} \leq V_b \leq 4.0\text{V}$ 、 $C_b = 20\text{pF}$ 、 $R_b = 1.4\text{k}\Omega$	200		1150		1150		ns
			$2.7\text{V} \leq \text{EV}_{\text{DD0}} < 4.0\text{V}$ 、 $2.3\text{V} \leq V_b \leq 2.7\text{V}$ 、 $C_b = 20\text{pF}$ 、 $R_b = 2.7\text{k}\Omega$	300		1150		1150		ns
SCKp 高电平 宽度	t_{KH1}	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$ 、 $2.7\text{V} \leq V_b \leq 4.0\text{V}$ 、 $C_b = 20\text{pF}$ 、 $R_b = 1.4\text{k}\Omega$		$t_{\text{KCY1}}/2$ -50		$t_{\text{KCY1}}/2$ -50		$t_{\text{KCY1}}/2$ -50		ns
		$2.7\text{V} \leq \text{EV}_{\text{DD0}} < 4.0\text{V}$ 、 $2.3\text{V} \leq V_b \leq 2.7\text{V}$ 、 $C_b = 20\text{pF}$ 、 $R_b = 2.7\text{k}\Omega$		$t_{\text{KCY1}}/2$ -120		$t_{\text{KCY1}}/2$ -120		$t_{\text{KCY1}}/2$ -120		ns
SCKp 低电平 宽度	t_{KL1}	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$ 、 $2.7\text{V} \leq V_b \leq 4.0\text{V}$ 、 $C_b = 20\text{pF}$ 、 $R_b = 1.4\text{k}\Omega$		$t_{\text{KCY1}}/2$ -7		$t_{\text{KCY1}}/2$ -50		$t_{\text{KCY1}}/2$ -50		ns
		$2.7\text{V} \leq \text{EV}_{\text{DD0}} < 4.0\text{V}$ 、 $2.3\text{V} \leq V_b \leq 2.7\text{V}$ 、 $C_b = 20\text{pF}$ 、 $R_b = 2.7\text{k}\Omega$		$t_{\text{KCY1}}/2$ -10		$t_{\text{KCY1}}/2$ -50		$t_{\text{KCY1}}/2$ -50		ns
Slp 准备时间 (对 SCKp↑) 注 1	t_{SIK1}	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$ 、 $2.7\text{V} \leq V_b \leq 4.0\text{V}$ 、 $C_b = 20\text{pF}$ 、 $R_b = 1.4\text{k}\Omega$		58		479		479		ns
		$2.7\text{V} \leq \text{EV}_{\text{DD0}} < 4.0\text{V}$ 、 $2.3\text{V} \leq V_b \leq 2.7\text{V}$ 、 $C_b = 20\text{pF}$ 、 $R_b = 2.7\text{k}\Omega$		121		479		479		ns
Slp 保持时间 (对 SCKp↑) 注 1	t_{KSI1}	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$ 、 $2.7\text{V} \leq V_b \leq 4.0\text{V}$ 、 $C_b = 20\text{pF}$ 、 $R_b = 1.4\text{k}\Omega$		10		10		10		ns
		$2.7\text{V} \leq \text{EV}_{\text{DD0}} < 4.0\text{V}$ 、 $2.3\text{V} \leq V_b \leq 2.7\text{V}$ 、 $C_b = 20\text{pF}$ 、 $R_b = 2.7\text{k}\Omega$		10		10		10		ns
SCKp↓→SOp 输出延迟时间 注 1	t_{KSO1}	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$ 、 $2.7\text{V} \leq V_b \leq 4.0\text{V}$ 、 $C_b = 20\text{pF}$ 、 $R_b = 1.4\text{k}\Omega$			60		60		60	ns
		$2.7\text{V} \leq \text{EV}_{\text{DD0}} < 4.0\text{V}$ 、 $2.3\text{V} \leq V_b \leq 2.7\text{V}$ 、 $C_b = 20\text{pF}$ 、 $R_b = 2.7\text{k}\Omega$			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)

 $(T_A = -40 \sim +85^{\circ}\text{C}, 2.7\text{V} \leq \text{EV}_{\text{DD0}} \leq \text{V}_{\text{DD}} \leq 5.5\text{V}, \text{V}_{\text{SS}} = \text{EV}_{\text{SS0}} = 0\text{V})$

(2/2)

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}	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$ 、 $2.7\text{V} \leq \text{V}_b \leq 4.0\text{V}$ 、 $\text{C}_b = 20\text{pF}$ 、 $\text{R}_b = 1.4\text{k}\Omega$	23		110		110		ns
		$2.7\text{V} \leq \text{EV}_{\text{DD0}} < 4.0\text{V}$ 、 $2.3\text{V} \leq \text{V}_b \leq 2.7\text{V}$ 、 $\text{C}_b = 20\text{pF}$ 、 $\text{R}_b = 2.7\text{k}\Omega$	33		110		110		ns
Slp 保持时间 (对 SCKp↓) 注 2	t_{KS11}	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$ 、 $2.7\text{V} \leq \text{V}_b \leq 4.0\text{V}$ 、 $\text{C}_b = 20\text{pF}$ 、 $\text{R}_b = 1.4\text{k}\Omega$	10		10		10		ns
		$2.7\text{V} \leq \text{EV}_{\text{DD0}} < 4.0\text{V}$ 、 $2.3\text{V} \leq \text{V}_b \leq 2.7\text{V}$ 、 $\text{C}_b = 20\text{pF}$ 、 $\text{R}_b = 2.7\text{k}\Omega$	10		10		10		ns
SCKp↑→SOp 输出延迟时间 注 2	t_{KS01}	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$ 、 $2.7\text{V} \leq \text{V}_b \leq 4.0\text{V}$ 、 $\text{C}_b = 20\text{pF}$ 、 $\text{R}_b = 1.4\text{k}\Omega$		10		10		10	ns
		$2.7\text{V} \leq \text{EV}_{\text{DD0}} < 4.0\text{V}$ 、 $2.3\text{V} \leq \text{V}_b \leq 2.7\text{V}$ 、 $\text{C}_b = 20\text{pF}$ 、 $\text{R}_b = 2.7\text{k}\Omega$		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 (32 引脚产品) / EV_{DD} 耐压 (64 引脚产品)) 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. $\text{R}_b[\Omega]$: 通信线 (SCKp、SOp) 的上拉电阻值 $\text{C}_b[\text{F}]$: 通信线 (SCKp、SOp) 的负载电容值 $\text{V}_b[\text{V}]$: 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=3、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. 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)

(T_A=−40 ~ +85°C、1.8V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(1/3)

项目	符号	条件		HS (高速主)		LS (低速主)		LV (低电压主)		单位
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp 周期 时间	t _{KCY1}	t _{KCY1} ≥ 4/f _{CLK}	4.0V ≤ EV _{DD0} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、R _b =1.4kΩ	300		1150		1150		ns
			2.7V ≤ EV _{DD0} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、R _b =2.7kΩ	500		1150		1150		ns
			1.8V ≤ EV _{DD0} < 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 ≤ EV _{DD0} ≤ 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 ≤ EV _{DD0} < 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 ≤ EV _{DD0} < 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 ≤ EV _{DD0} ≤ 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 ≤ EV _{DD0} < 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 ≤ EV _{DD0} < 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 EV_{DD0} ≥ 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 (32 引脚产品) / EV_{DD} 耐压 (64 引脚产品)) 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)

 $(T_A = -40 \sim +85^\circ\text{C}, 1.8\text{V} \leq \text{EV}_{\text{DD0}} \leq \text{V}_{\text{DD}} \leq 5.5\text{V}, \text{V}_{\text{SS}} = \text{EV}_{\text{SS0}} = 0\text{V})$

(2/3)

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}	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$ 、 $2.7\text{V} \leq \text{V}_b \leq 4.0\text{V}$ 、 $\text{C}_b = 30\text{pF}$ 、 $\text{R}_b = 1.4\text{k}\Omega$	81		479		479		ns
		$2.7\text{V} \leq \text{EV}_{\text{DD0}} < 4.0\text{V}$ 、 $2.3\text{V} \leq \text{V}_b \leq 2.7\text{V}$ 、 $\text{C}_b = 30\text{pF}$ 、 $\text{R}_b = 2.7\text{k}\Omega$	177		479		479		ns
		$1.8\text{V} \leq \text{EV}_{\text{DD0}} < 3.3\text{V}$ 、 $1.6\text{V} \leq \text{V}_b \leq 2.0\text{V}$ 注 2、 $\text{C}_b = 30\text{pF}$ 、 $\text{R}_b = 5.5\text{k}\Omega$	479		479		479		ns
Slp 保持时间 (对 SCKp↑) 注 1	t_{KSI1}	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$ 、 $2.7\text{V} \leq \text{V}_b \leq 4.0\text{V}$ 、 $\text{C}_b = 30\text{pF}$ 、 $\text{R}_b = 1.4\text{k}\Omega$	19		19		19		ns
		$2.7\text{V} \leq \text{EV}_{\text{DD0}} < 4.0\text{V}$ 、 $2.3\text{V} \leq \text{V}_b \leq 2.7\text{V}$ 、 $\text{C}_b = 30\text{pF}$ 、 $\text{R}_b = 2.7\text{k}\Omega$	19		19		19		ns
		$1.8\text{V} \leq \text{EV}_{\text{DD0}} < 3.3\text{V}$ 、 $1.6\text{V} \leq \text{V}_b \leq 2.0\text{V}$ 注 2、 $\text{C}_b = 30\text{pF}$ 、 $\text{R}_b = 5.5\text{k}\Omega$	19		19		19		ns
SCKp↓→SOp 输出延迟时间 注 1	t_{KSO1}	$4.0\text{V} \leq \text{EV}_{\text{DD0}} \leq 5.5\text{V}$ 、 $2.7\text{V} \leq \text{V}_b \leq 4.0\text{V}$ 、 $\text{C}_b = 30\text{pF}$ 、 $\text{R}_b = 1.4\text{k}\Omega$		100		100		100	ns
		$2.7\text{V} \leq \text{EV}_{\text{DD0}} < 4.0\text{V}$ 、 $2.3\text{V} \leq \text{V}_b \leq 2.7\text{V}$ 、 $\text{C}_b = 30\text{pF}$ 、 $\text{R}_b = 2.7\text{k}\Omega$		195		195		195	ns
		$1.8\text{V} \leq \text{EV}_{\text{DD0}} < 3.3\text{V}$ 、 $1.6\text{V} \leq \text{V}_b \leq 2.0\text{V}$ 注 2、 $\text{C}_b = 30\text{pF}$ 、 $\text{R}_b = 5.5\text{k}\Omega$		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 $\text{EV}_{\text{DD0}} \geq \text{V}_b$ 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 (32 引脚产品) / EV_{DD} 耐压 (64 引脚产品)) 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)

(T_A=−40 ~ +85°C、1.8V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(3/3)

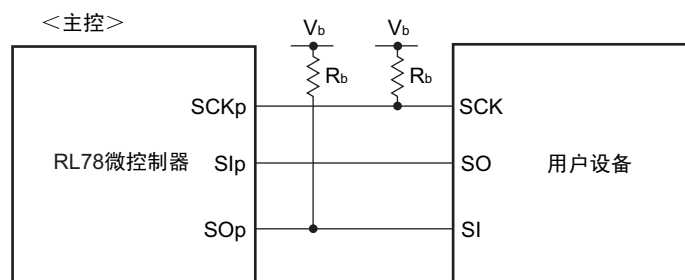
项目	符号	条件	HS (高速主) 模式		LS (低速主) 模式		LV (低电压主) 模式		单位
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Slp 准备时间 (对 SCKp↓) 注 1	t _{SIK1}	4.0V ≤ EV _{DD0} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、R _b =1.4kΩ	44		110		110		ns
		2.7V ≤ EV _{DD0} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、R _b =2.7kΩ	44		110		110		ns
		1.8V ≤ EV _{DD0} < 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 _{KSl1}	4.0V ≤ EV _{DD0} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、R _b =1.4kΩ	19		19		19		ns
		2.7V ≤ EV _{DD0} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、R _b =2.7kΩ	19		19		19		ns
		1.8V ≤ EV _{DD0} < 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 ≤ EV _{DD0} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、R _b =1.4kΩ		25		25		25	ns
		2.7V ≤ EV _{DD0} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、R _b =2.7kΩ		25		25		25	ns
		1.8V ≤ EV _{DD0} < 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 EV_{DD0} ≥ V_b 的情况下使用。

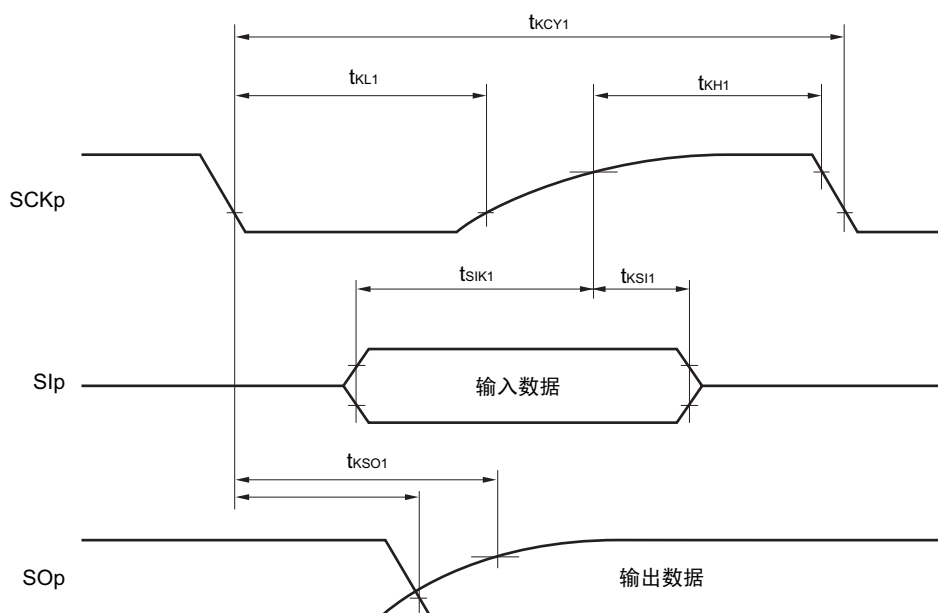
注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg)，将 Slp 引脚选择为 TTL 输入缓冲器并且将 SOp 引脚和 SCKp 引脚选择为 N 沟道漏极开路输出 (V_{DD} 耐压 (32 引脚产品) /EV_{DD} 耐压 (64 引脚产品)) 模式。V_{IH} 和 V_{IL} 请参照选择为 TTL 输入缓冲器时的 DC 特性。

CSI 模式的连接图（不同电位的通信）

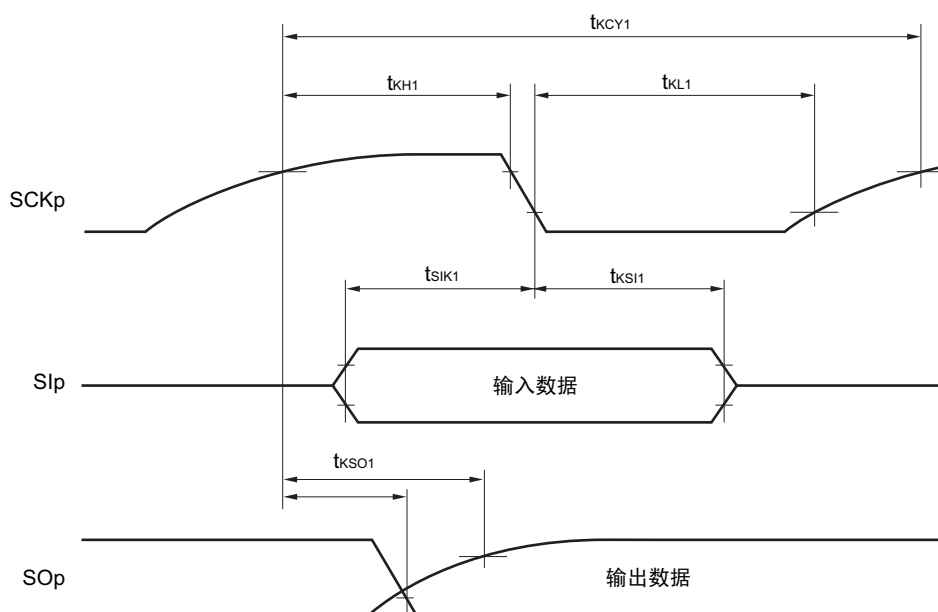


- 备注 1. $R_b[\Omega]$: 通信线 (SCKp、SO_p) 的上拉电阻值 $C_b[F]$: 通信线 (SCKp、SO_p) 的负载电容值 $V_b[V]$: 通信线的电压
2. p: CSI 号 (p=00、01、10、20) m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)
g: PIM、POM 号 (g=0、1、3、5、7)
3. f_{MCK} : 串行阵列单元的运行时钟频率
(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。m: 单元号、n: 通道号 (mn=00))
4. CSI11、CSI21 以及 64 引脚产品的 CSI01 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、01、10、20) m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)
 g: PIM、POM 号 (g=0、1、3、5、7)

2. CSI11、CSI21 以及 64 引脚产品的 CSI01 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)

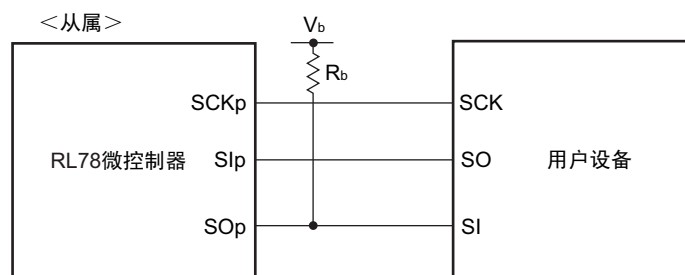
(T_A=−40 ~ +85°C、1.8V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

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≤EV _{DD0} ≤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}		ns
		2.7V≤EV _{DD0} <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
		f _{MCK} ≤ 4MHz	6/f _{MCK}		10/f _{MCK}		10/f _{MCK}		ns	
		1.8V≤EV _{DD0} <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
			f _{MCK} ≤ 4MHz	10/f _{MCK}		10/f _{MCK}		10/f _{MCK}		ns
SCKp 高低 电平宽度	t _{KH2} 、 t _{KL2}	4.0V ≤ EV _{DD0} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V	t _{KCY2} /2 −12		t _{KCY2} /2 −50		t _{KCY2} /2 −50		ns	
		2.7V ≤ EV _{DD0} < 4.0V、 2.3V ≤ V _b ≤ 2.7V	t _{KCY2} /2 −18		t _{KCY2} /2 −50		t _{KCY2} /2 −50		ns	
		1.8V ≤ EV _{DD0} < 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 ≤ EV _{DD0} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V	1/f _{MCK} +20		1/f _{MCK} +30		1/f _{MCK} +30		ns	
		2.7V ≤ EV _{DD0} ≤ 4.0V、 2.3V ≤ V _b ≤ 2.7V	1/f _{MCK} +20		1/f _{MCK} +30		1/f _{MCK} +30		ns	
		1.8V ≤ EV _{DD0} ≤ 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 3	t _{KSI2}		1/f _{MCK} +31		1/f _{MCK} +31		1/f _{MCK} +31		ns	
SCKp↓→SOp 输出延迟时间 注 4	t _{KSO2}	4.0V ≤ EV _{DD0} ≤ 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 ≤ EV _{DD0} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 Cb=30pF、Rb=2.7kΩ		2/f _{MCK} +214		2/f _{MCK} +573		2/f _{MCK} +573	ns	
		1.8V ≤ EV _{DD0} < 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 $EV_{DD0} \geq V_b$ 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↑”。

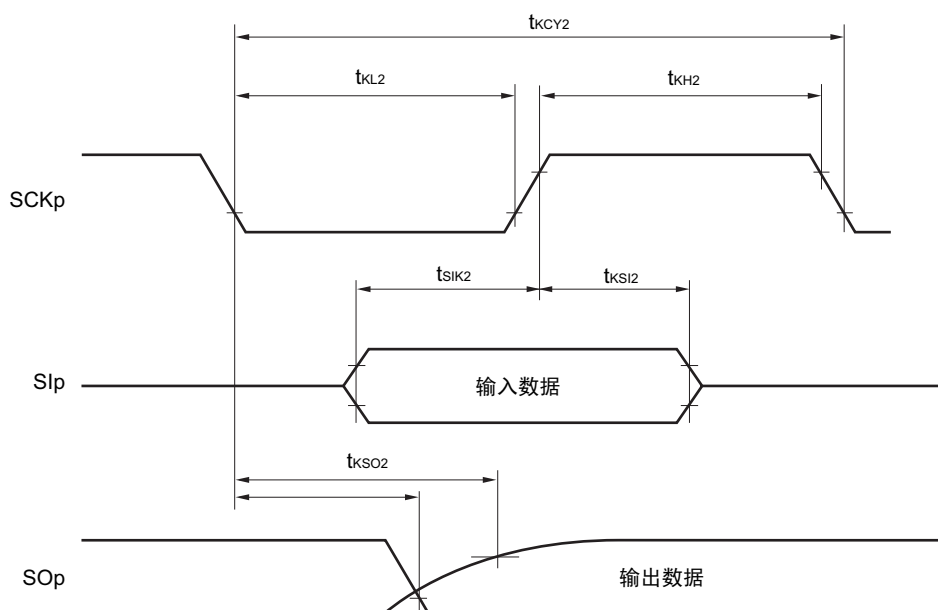
注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg)，将 Slp 引脚和 SCKp 引脚选择为 TTL 输入缓冲器并且将 SOp 引脚选择为 N 沟道漏极开路输出 (V_{DD} 耐压 (32 引脚产品) / EV_{DD} 耐压 (64 引脚产品)) 模式。 V_{IH} 和 V_{IL} 请参照选择为 TTL 输入缓冲器时的 DC 特性。

CSI 模式的连接图 (不同电位的通信)

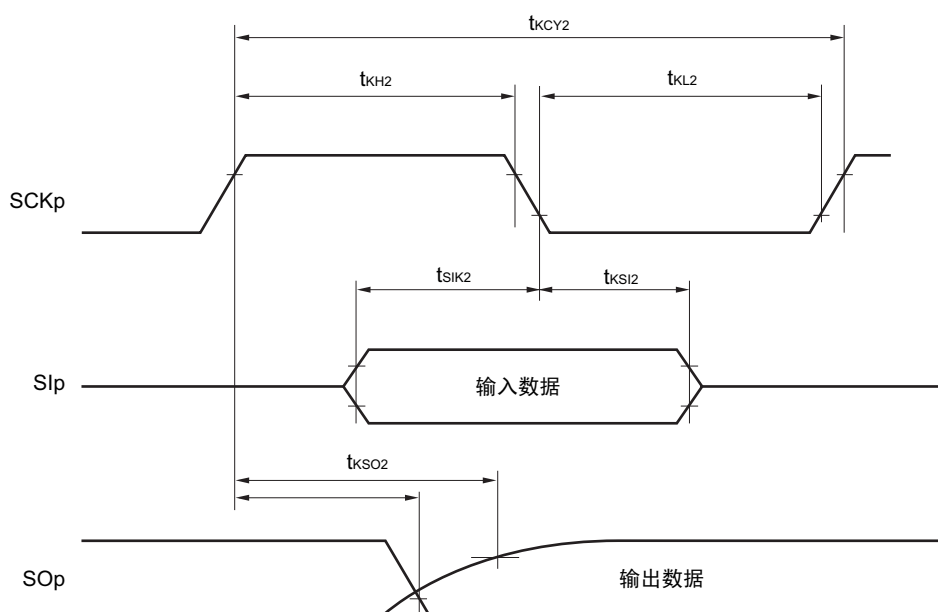


- 备注 1. $R_b[\Omega]$: 通信线 (SOp) 的上拉电阻值 $C_b[F]$: 通信线 (SOp) 的负载电容值 $V_b[V]$: 通信线的电压
2. p: CSI 号 (p=00、01、10、20) m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)
g: PIM、POM 号 (g=0、1、3、5、7)
3. f_{MCK} : 串行阵列单元的运行时钟频率
(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。m: 单元号、
n: 通道号 (mn=00、01、02、10))
4. CSI11、CSI21 以及 64 引脚产品的 CSI01 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。另外，在从属选择功能的时钟同步串行通信时不能进行不同电位的通信。

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、01、10、20) m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)
 g: PIM、POM 号 (g=0、1、3、5、7)

- CSI11、CSI21 以及 64 引脚产品的 CSI01 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。另外，在从属选择功能的时钟同步串行通信时不能进行不同电位的通信。

(10) 不同电位 (1.8V、2.5V、3V) 的通信 (简易 I²C 模式)(T_A=−40 ~ +85°C、1.8V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(1/2)

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 ≤ EV _{DD0} ≤ 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 ≤ EV _{DD0} < 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 ≤ EV _{DD0} ≤ 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 ≤ EV _{DD0} < 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 ≤ EV _{DD0} < 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 ≤ EV _{DD0} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =50pF、R _b =2.7kΩ	475		1550		1550		ns
		2.7V ≤ EV _{DD0} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =50pF、R _b =2.7kΩ	475		1550		1550		ns
		4.0V ≤ EV _{DD0} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =100pF、R _b =2.8kΩ	1150		1550		1550		ns
		2.7V ≤ EV _{DD0} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =100pF、R _b =2.7kΩ	1150		1550		1550		ns
		1.8V ≤ EV _{DD0} < 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 ≤ EV _{DD0} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =50pF、R _b =2.7kΩ	245		610		610		ns
		2.7V ≤ EV _{DD0} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =50pF、R _b =2.7kΩ	200		610		610		ns
		4.0V ≤ EV _{DD0} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =100pF、R _b =2.8kΩ	675		610		610		ns
		2.7V ≤ EV _{DD0} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =100pF、R _b =2.7kΩ	600		610		610		ns
		1.8V ≤ EV _{DD0} < 3.3V、 1.6V ≤ V _b ≤ 2.0V 注 2、 C _b =100pF、R _b =5.5kΩ	610		610		610		ns

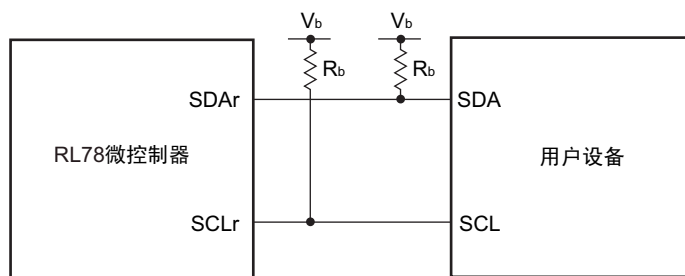
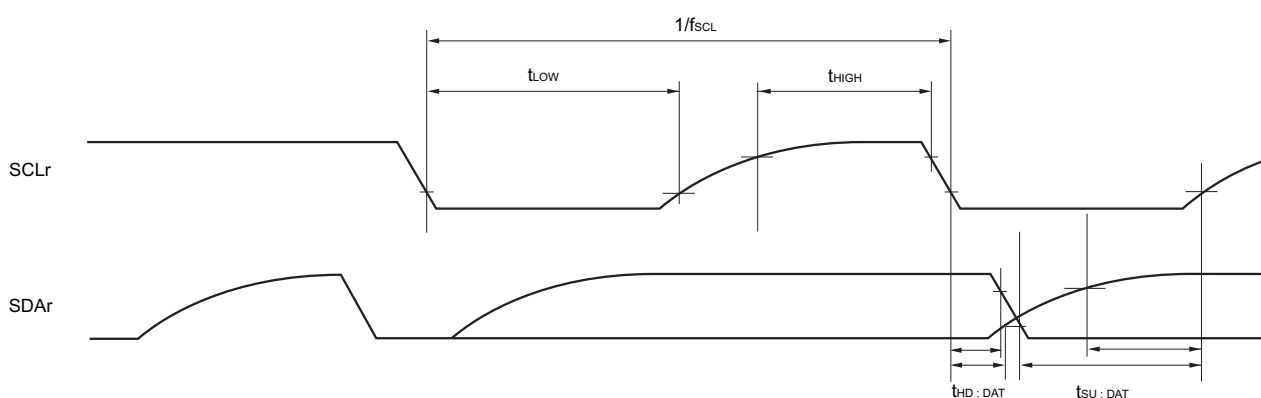
(10) 不同电位 (1.8V、2.5V、3V) 的通信 (简易 I²C 模式)(T_A=−40 ~ +85°C、1.8V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(2/2)

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 ≤ EV _{DD0} ≤ 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 ≤ EV _{DD0} < 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 ≤ EV _{DD0} ≤ 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 ≤ EV _{DD0} < 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 ≤ EV _{DD0} < 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 ≤ EV _{DD0} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =50pF、R _b =2.7kΩ	0	305	0	305	0	305	ns
		2.7V ≤ EV _{DD0} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =50pF、R _b =2.7kΩ	0	305	0	305	0	305	ns
		4.0V ≤ EV _{DD0} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =100pF、R _b =2.8kΩ	0	355	0	355	0	355	ns
		2.7V ≤ EV _{DD0} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =100pF、R _b =2.7kΩ	0	355	0	355	0	355	ns
		1.8V ≤ EV _{DD0} < 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 EV_{DD0} ≥ 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 (32 引脚产品) /EV_{DD} 耐压 (64 引脚产品)) 模式，并且将 SCLr 引脚选择为 N 沟道漏极开路输出 (V_{DD} 耐压 (32 引脚产品) /EV_{DD} 耐压 (64 引脚产品)) 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、01、10、11、20) g: PIM、POM 号 (g=0、1、5、7)
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、2),
mn=00、01、02、10)

31.5.2 串行接口 IICA

(1) I²C 标准模式(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0}=V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(1/2)

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 ≤ EV _{DD0} ≤ 5.5V	0	100	0	100	0	100	kHz
			1.8V ≤ EV _{DD0} ≤ 5.5V	0	100	0	100	0	100	kHz
			1.7V ≤ EV _{DD0} ≤ 5.5V	0	100	0	100	0	100	kHz
			1.6V ≤ EV _{DD0} ≤ 5.5V	—		0	100	0	100	kHz
重新开始条件的准备时间	t _{SU: STA}	2.7V ≤ EV _{DD0} ≤ 5.5V		4.7		4.7		4.7		μs
		1.8V ≤ EV _{DD0} ≤ 5.5V		4.7		4.7		4.7		μs
		1.7V ≤ EV _{DD0} ≤ 5.5V		4.7		4.7		4.7		μs
		1.6V ≤ EV _{DD0} ≤ 5.5V		—		4.7		4.7		μs
保持时间注 1	t _{HD: STA}	2.7V ≤ EV _{DD0} ≤ 5.5V		4.0		4.0		4.0		μs
		1.8V ≤ EV _{DD0} ≤ 5.5V		4.0		4.0		4.0		μs
		1.7V ≤ EV _{DD0} ≤ 5.5V		4.0		4.0		4.0		μs
		1.6V ≤ EV _{DD0} ≤ 5.5V		—		4.0		4.0		μs
SCLA0="L" 的保持时间	t _{LOW}	2.7V ≤ EV _{DD0} ≤ 5.5V		4.7		4.7		4.7		μs
		1.8V ≤ EV _{DD0} ≤ 5.5V		4.7		4.7		4.7		μs
		1.7V ≤ EV _{DD0} ≤ 5.5V		4.7		4.7		4.7		μs
		1.6V ≤ EV _{DD0} ≤ 5.5V		—		4.7		4.7		μs
SCLA0="H" 的保持时间	t _{HIGH}	2.7V ≤ EV _{DD0} ≤ 5.5V		4.0		4.0		4.0		μs
		1.8V ≤ EV _{DD0} ≤ 5.5V		4.0		4.0		4.0		μs
		1.7V ≤ EV _{DD0} ≤ 5.5V		4.0		4.0		4.0		μs
		1.6V ≤ EV _{DD0} ≤ 5.5V		—		4.0		4.0		μs

(1) I²C 标准模式(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0}=V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

(2/2)

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}	2.7V ≤ EV _{DD0} ≤ 5.5V	250		250		250		ns
		1.8V ≤ EV _{DD0} ≤ 5.5V	250		250		250		ns
		1.7V ≤ EV _{DD0} ≤ 5.5V	250		250		250		ns
		1.6V ≤ EV _{DD0} ≤ 5.5V	—		250		250		ns
数据保持时间 (发送时) 注 2	t _{HD: DAT}	2.7V ≤ EV _{DD0} ≤ 5.5V	0	3.45	0	3.45	0	3.45	μs
		1.8V ≤ EV _{DD0} ≤ 5.5V	0	3.45	0	3.45	0	3.45	μs
		1.7V ≤ EV _{DD0} ≤ 5.5V	0	3.45	0	3.45	0	3.45	μs
		1.6V ≤ EV _{DD0} ≤ 5.5V	—		0	3.45	0	3.45	μs
停止条件的 准备时间	t _{SU: STO}	2.7V ≤ EV _{DD0} ≤ 5.5V	4.0		4.0		4.0		μs
		1.8V ≤ EV _{DD0} ≤ 5.5V	4.0		4.0		4.0		μs
		1.7V ≤ EV _{DD0} ≤ 5.5V	4.0		4.0		4.0		μs
		1.6V ≤ EV _{DD0} ≤ 5.5V	—		4.0		4.0		μs
总线空闲时间	t _{BUF}	2.7V ≤ EV _{DD0} ≤ 5.5V	4.7		4.7		4.7		μs
		1.8V ≤ EV _{DD0} ≤ 5.5V	4.7		4.7		4.7		μs
		1.7V ≤ EV _{DD0} ≤ 5.5V	4.7		4.7		4.7		μs
		1.6V ≤ EV _{DD0} ≤ 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 0 (PIOR0) 的 bit2 (PIOR02) 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 ≤ EV_{DD0}=V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=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:	2.7V ≤ EV _{DD0} ≤ 5.5V	0	400	0	400	0	400	kHz
		f _{CLK} ≥ 3.5MHz	1.8V ≤ EV _{DD0} ≤ 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 ≤ EV _{DD0} ≤ 5.5V		0.6		0.6		0.6		μs
		1.8V ≤ EV _{DD0} ≤ 5.5V		0.6		0.6		0.6		μs
保持时间注 1	t _{HD: STA}	2.7V ≤ EV _{DD0} ≤ 5.5V		0.6		0.6		0.6		μs
		1.8V ≤ EV _{DD0} ≤ 5.5V		0.6		0.6		0.6		μs
SCLA0="L" 的保持时间	t _{LOW}	2.7V ≤ EV _{DD0} ≤ 5.5V		1.3		1.3		1.3		μs
		1.8V ≤ EV _{DD0} ≤ 5.5V		1.3		1.3		1.3		μs
SCLA0="H" 的保持时间	t _{HIGH}	2.7V ≤ EV _{DD0} ≤ 5.5V		0.6		0.6		0.6		μs
		1.8V ≤ EV _{DD0} ≤ 5.5V		0.6		0.6		0.6		μs
数据准备时间 (接收时)	t _{SU: DAT}	2.7V ≤ EV _{DD0} ≤ 5.5V		100		100		100		ns
		1.8V ≤ EV _{DD0} ≤ 5.5V		100		100		100		ns
数据保持时间 (发送时) 注 2	t _{HD: DAT}	2.7V ≤ EV _{DD0} ≤ 5.5V		0	0.9	0	0.9	0	0.9	μs
		1.8V ≤ EV _{DD0} ≤ 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 ≤ EV _{DD0} ≤ 5.5V		0.6		0.6		0.6		μs
		1.8V ≤ EV _{DD0} ≤ 5.5V		0.6		0.6		0.6		μs
总线空闲时间	t _{BUF}	2.7V ≤ EV _{DD0} ≤ 5.5V		1.3		1.3		1.3		μs
		1.8V ≤ EV _{DD0} ≤ 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 0 (PIOR0) 的 bit2 (PIOR02) 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

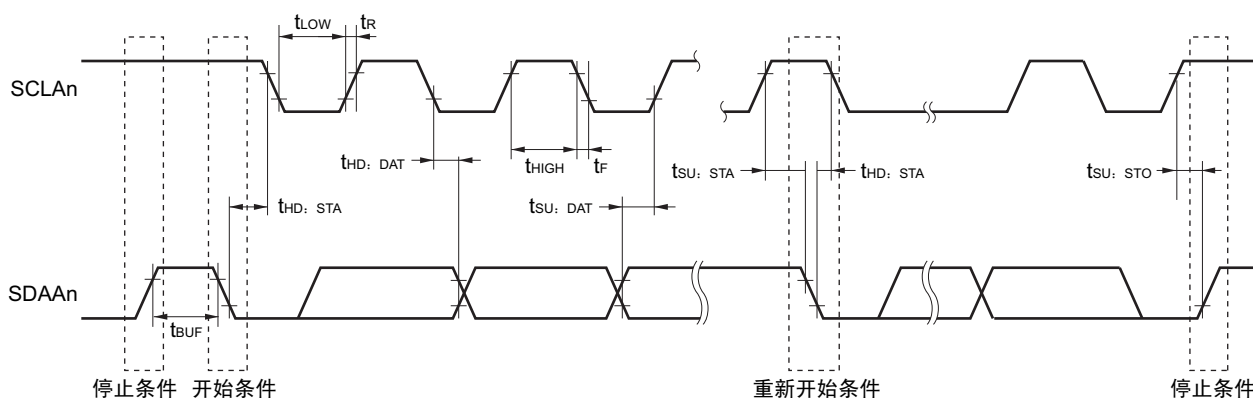
(3) I²C 增强型快速模式(T_A=−40 ~ +85°C、1.6V ≤ EV_{DD0}=V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

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}	增强型快速模式: f _{CLK} ≥ 10MHz	2.7V ≤ EV _{DD0} ≤ 5.5V	0	1000	—	—	—	—	kHz
重新开始条件的准备时间	t _{SU: STA}	2.7V ≤ EV _{DD0} ≤ 5.5V		0.26		—	—	—	—	μs
保持时间注 1	t _{HD: STA}	2.7V ≤ EV _{DD0} ≤ 5.5V		0.26		—	—	—	—	μs
SCLA0="L" 的保持时间	t _{LOW}	2.7V ≤ EV _{DD0} ≤ 5.5V		0.5		—	—	—	—	μs
SCLA0="H" 的保持时间	t _{HIGH}	2.7V ≤ EV _{DD0} ≤ 5.5V		0.26		—	—	—	—	μs
数据准备时间 (接收时)	t _{SU: DAT}	2.7V ≤ EV _{DD0} ≤ 5.5V		50		—	—	—	—	ns
数据保持时间 (发送时) 注 2	t _{HD: DAT}	2.7V ≤ EV _{DD0} ≤ 5.5V		0	0.45	—	—	—	—	μs
停止条件的准备时间	t _{SU: STO}	2.7V ≤ EV _{DD0} ≤ 5.5V		0.26		—	—	—	—	μs
总线空闲时间	t _{BUF}	2.7V ≤ EV _{DD0} ≤ 5.5V		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_{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 0 (PIOR0) 的 bit2 (PIOR02) 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:增强型快速模式: C_b=120pF、R_b=1.1kΩI²C 串行传送时序

备注 n=0

31.6 模拟特性

31.6.1 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 ~ ANI7	参照 31.6.1(1)。	参照 31.6.1(3)。	参照 31.6.1(4)。
ANI16 ~ ANI19	参照 31.6.1(2)。		
内部基准电压 温度传感器的输出电压	参照 31.6.1(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 ~ ANI7、内部基准电压、温度传感器的输出电压

($T_A=-40\sim+85^{\circ}\text{C}$ 、 $1.6\text{V}\leq AV_{REFP}\leq V_{DD}\leq 5.5\text{V}$ 、 $V_{SS}=0\text{V}$ 、基准电压(+)= AV_{REFP} 、基准电压(-)= $AV_{REFM}=0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	RES		8		10	bit
综合误差注 1	AINL	10 位分辨率 $AV_{REFP}=V_{DD}$ 注 3	$1.8\text{V}\leq AV_{REFP}\leq 5.5\text{V}$	1.2	± 3.5	LSB
			$1.6\text{V}\leq AV_{REFP}\leq 5.5\text{V}$ 注 4	1.2	± 7.0	LSB
转换时间	t_{CONV}	10 位分辨率 转换对象: ANI2 ~ ANI7	$3.6\text{V}\leq V_{DD}\leq 5.5\text{V}$	2.125	39	μs
			$2.7\text{V}\leq V_{DD}\leq 5.5\text{V}$	3.1875	39	μs
			$1.8\text{V}\leq V_{DD}\leq 5.5\text{V}$	17	39	μs
			$1.6\text{V}\leq V_{DD}\leq 5.5\text{V}$	57	95	μs
		10 位分辨率 转换对象: 内部基准电压、 温度传感器的输出电压 (HS (高速主) 模式)	$3.6\text{V}\leq V_{DD}\leq 5.5\text{V}$	2.375	39	μs
			$2.7\text{V}\leq V_{DD}\leq 5.5\text{V}$	3.5625	39	μs
			$2.4\text{V}\leq V_{DD}\leq 5.5\text{V}$	17	39	μs
零刻度误差注 1、2	E_{ZS}	10 位分辨率 $AV_{REFP}=V_{DD}$ 注 3	$1.8\text{V}\leq AV_{REFP}\leq 5.5\text{V}$		± 0.25	%FSR
			$1.6\text{V}\leq AV_{REFP}\leq 5.5\text{V}$ 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	$1.8\text{V}\leq AV_{REFP}\leq 5.5\text{V}$		± 0.25	%FSR
			$1.6\text{V}\leq AV_{REFP}\leq 5.5\text{V}$ 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	$1.8\text{V}\leq AV_{REFP}\leq 5.5\text{V}$		± 2.5	LSB
			$1.6\text{V}\leq AV_{REFP}\leq 5.5\text{V}$ 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	$1.8\text{V}\leq AV_{REFP}\leq 5.5\text{V}$		± 1.5	LSB
			$1.6\text{V}\leq AV_{REFP}\leq 5.5\text{V}$ 注 4		± 2.0	LSB
模拟输入电压	V_{AIN}	ANI2 ~ ANI7	0		AV_{REFP}	V
		内部基准电压输出 ($2.4\text{V}\leq V_{DD}\leq 5.5\text{V}$ 、HS (高速主) 模式)		V_{BGR} 注 5		V
		温度传感器的输出电压 ($2.4\text{V}\leq V_{DD}\leq 5.5\text{V}$ 、HS (高速主) 模式)		V_{TMPS25} 注 5		V

注 1. 不包含量化误差 ($\pm 1/2$ LSB)。

2. 用对满刻度值的比率 (%FSR) 表示。

3. 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 ± 1.0 LSB。

零刻度误差 / 满刻度误差: $AV_{REFP}=V_{DD}$ 的 MAX. 值必须加上 $\pm 0.05\%$ FSR。

积分线性误差 / 微分线性误差: $AV_{REFP}=V_{DD}$ 的 MAX. 值必须加上 ± 0.5 LSB。

4. 这是将转换时间设定为 MIN. $57\mu\text{s}$ 、MAX. $95\mu\text{s}$ 时的值。

5. 请参照“31.6.2 温度传感器 / 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

($T_A=-40 \sim +85^{\circ}\text{C}$ 、 $1.6\text{V} \leq EV_{DD0} \leq V_{DD} \leq 5.5\text{V}$ 、 $1.6\text{V} \leq AV_{REFP} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS}=EV_{SS0}=0\text{V}$ 、基准电压(+)= AV_{REFP} 、基准电压(-)= $AV_{REFM}=0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	RES		8		10	bit
综合误差注 1	AINL	10 位分辨率		1.2	± 5.0	LSB
		$EV_{DD0} \leq AV_{REFP} = V_{DD}$ 注3、4		1.2	± 8.5	LSB
转换时间	t_{CONV}	10 位分辨率				
		转换对象: ANI16 ~ ANI19				
		$3.6\text{V} \leq V_{DD} \leq 5.5\text{V}$	2.125		39	μs
		$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	3.1875		39	μs
零刻度误差注 1、2	E_{ZS}	10 位分辨率			± 0.35	%FSR
		$EV_{DD0} \leq AV_{REFP} = V_{DD}$ 注3、4			± 0.60	%FSR
满刻度误差注 1、2	E_{FS}	10 位分辨率			± 0.35	%FSR
		$EV_{DD0} \leq AV_{REFP} = V_{DD}$ 注3、4			± 0.60	%FSR
积分线性误差注 1	ILE	10 位分辨率			± 3.5	LSB
		$EV_{DD0} \leq AV_{REFP} = V_{DD}$ 注3、4			± 6.0	LSB
微分线性误差注 1	DLE	10 位分辨率			± 2.0	LSB
		$EV_{DD0} \leq AV_{REFP} = V_{DD}$ 注3、4			± 2.5	LSB
模拟输入电压	V_{AIN}	ANI16 ~ ANI19	0		AV_{REFP} 并且 EV_{DD0}	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. 当 $EV_{DD0} \leq AV_{REFP} < V_{DD}$ 时, MAX. 值如下:

综合误差: $AV_{REFP}=V_{DD}$ 的 MAX. 值必须加上 $\pm 1.0\text{LSB}$ 。

零刻度误差 / 满刻度误差: $AV_{REFP}=V_{DD}$ 的 MAX. 值必须加上 $\pm 0.05\%\text{FSR}$ 。

积分线性误差 / 微分线性误差: $AV_{REFP}=V_{DD}$ 的 MAX. 值必须加上 $\pm 0.5\text{LSB}$ 。

4. 当 $AV_{REFP} < EV_{DD0} \leq V_{DD}$ 时, MAX. 值如下:

综合误差: $AV_{REFP}=V_{DD}$ 的 MAX. 值必须加上 $\pm 4.0\text{LSB}$ 。

零刻度误差 / 满刻度误差: $AV_{REFP}=V_{DD}$ 的 MAX. 值必须加上 $\pm 0.20\%\text{FSR}$ 。

积分线性误差 / 微分线性误差: $AV_{REFP}=V_{DD}$ 的 MAX. 值必须加上 $\pm 2.0\text{LSB}$ 。

5. 这是将转换时间设定为 MIN.57 μs 、MAX.95 μs 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 ~ ANI7、ANI16 ~ ANI19、内部基准电压、温度传感器的输出电压

($T_A = -40 \sim +85^{\circ}\text{C}$ 、 $1.6\text{V} \leq EV_{DD0} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = EV_{SS0} = 0\text{V}$ 、基准电压 (+)= V_{DD} 、基准电压 (-)= V_{SS})

项目	符号	条件		MIN.	TYP.	MAX.	单位
分辨率	RES			8		10	bit
综合误差注 1	AINL	10 位分辨率	$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$		1.2	± 7.0	LSB
			$1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 注 3		1.2	± 10.5	LSB
转换时间	t_{CONV}	10 位分辨率 转换对象: ANI0 ~ ANI7、 ANI16 ~ ANI19	$3.6\text{V} \leq V_{DD} \leq 5.5\text{V}$	2.125		39	μs
			$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	3.1875		39	μs
			$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$	17		39	μs
			$1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$	57		95	μs
		10 位分辨率 转换对象: 内部基准电压、 温度传感器的输出电压 (HS (高速主) 模式)	$3.6\text{V} \leq V_{DD} \leq 5.5\text{V}$	2.375		39	μs
			$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	3.5625		39	μs
			$2.4\text{V} \leq V_{DD} \leq 5.5\text{V}$	17		39	μs
零刻度误差注 1、2	E_{ZS}	10 位分辨率	$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$			± 0.60	%FSR
			$1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 注 3			± 0.85	%FSR
满刻度误差注 1、2	E_{FS}	10 位分辨率	$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$			± 0.60	%FSR
			$1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 注 3			± 0.85	%FSR
积分线性误差注 1	ILE	10 位分辨率	$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$			± 4.0	LSB
			$1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 注 3			± 6.5	LSB
微分线性误差注 1	DLE	10 位分辨率	$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$			± 2.0	LSB
			$1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 注 3			± 2.5	LSB
模拟输入电压	V_{AIN}	ANI0 ~ ANI7		0		V_{DD}	V
		ANI16 ~ ANI19		0		EV_{DD0}	V
		内部基准电压输出 ($2.4\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$ 、HS (高速主) 模式)		V_{BGR} 注 4			V
		温度传感器的输出电压 ($2.4\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$ 、HS (高速主) 模式)		V_{TMPS25} 注 4			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. 这是将转换时间设定为 MIN.57 μs 、MAX.95 μs 时的值。
 4. 请参照“31.6.2 温度传感器 / 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 ~ ANI7、ANI16 ~ ANI19

($T_A = -40 \sim +85^{\circ}\text{C}$ 、 $2.4\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $1.6\text{V} \leq EV_{DD0} \leq V_{DD}$ 、 $V_{SS} = EV_{SS0} = 0\text{V}$ 、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“31.6.2 温度传感器 / 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。

31.6.2 温度传感器 / 内部基准电压的特性

($T_A = -40 \sim +85^\circ\text{C}$ 、 $2.4\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = EV_{SS0} = 0\text{V}$ 、HS (高速主) 模式)

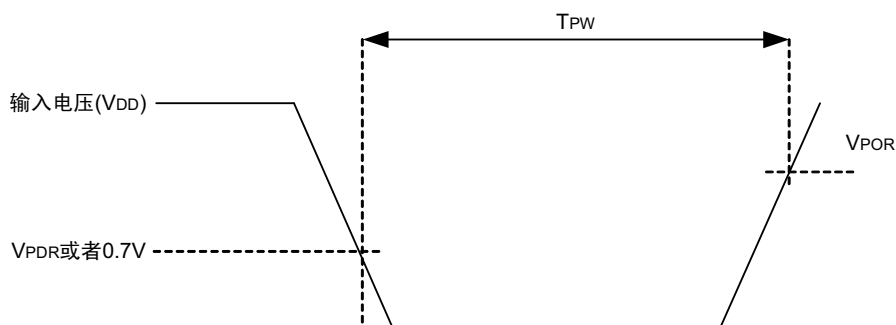
项目	符号	条件	MIN.	TYP.	MAX.	单位
温度传感器的输出电压	V_{TMPS25}	ADS 寄存器 =80H、 $T_A = +25^\circ\text{C}$		1.05		V
内部基准电压	V_{BGR}	ADS 寄存器 =81H	1.38	1.45	1.5	V
温度系数	F_{VTMPS}	取决于温度传感器电压的温度。		-3.6		mV/ $^\circ\text{C}$
运行稳定等待时间	t_{AMP}		5			μs

31.6.3 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}	电源电压下降时注 1	1.46	1.50	1.54	V
最小脉宽注 2	T_{PW}		300			μs

- 注 1. 但是，在 LVD 为 OFF 的条件下工作电压下降时，必须在工作电压低于“31.4 AC 特性”所示的工作电压范围前，通过 STOP 模式的转移、电压检测功能或者外部复位引脚，置为复位状态。
2. 这是在 V_{DD} 低于 V_{PDR} 时 POR 复位所需的时间。另外，在 STOP 模式中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。



31.6.4 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
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 ~ +85°C、V_{PDR} ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

项目	符号	条件		MIN.	TYP.	MAX.	单位
中断 & 复位 模式	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

31.6.5 电源电压的上升斜率特性

(T_A=−40 ~ +85°C、V_{SS}=0V)

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

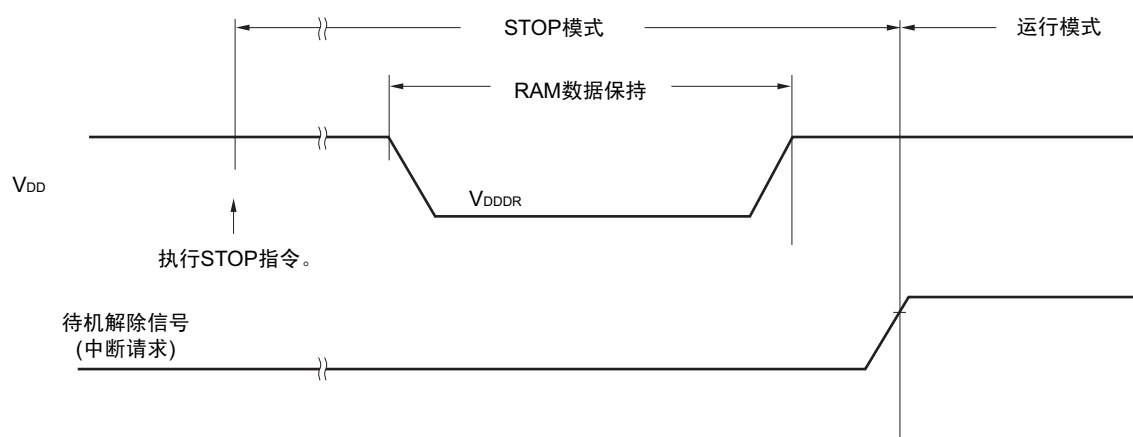
注意 必须在 V_{DD} 达到“31.4 AC 特性”所示的工作电压范围前，通过 LVD 电路或者外部复位保持内部复位状态。

31.7 数据存储器的保持特性

(T_A=-40 ~ +85°C、V_{SS}=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
数据保持电源电压	V _{DDDR}		1.46 注 1、2		5.5	V

- 注 1. 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。
2. 当低于推荐工作电压时，必须在此之前转移到 STOP 模式。



31.8 闪存编程特性

(T_A=-40 ~ +85°C、1.8V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

项目	符号	条件		MIN.	TYP.	MAX.	单位
系统时钟频率	f _{CLK}	1.8V ≤ V _{DD} ≤ 5.5V		1		32	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°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°C		1000000		
		保持年数：5 年	T _A =85°C	100000			
		保持年数：20 年	T _A =85°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 次改写下次改写为止的期间。
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。

31.9 专用闪存编程器通信 (UART)

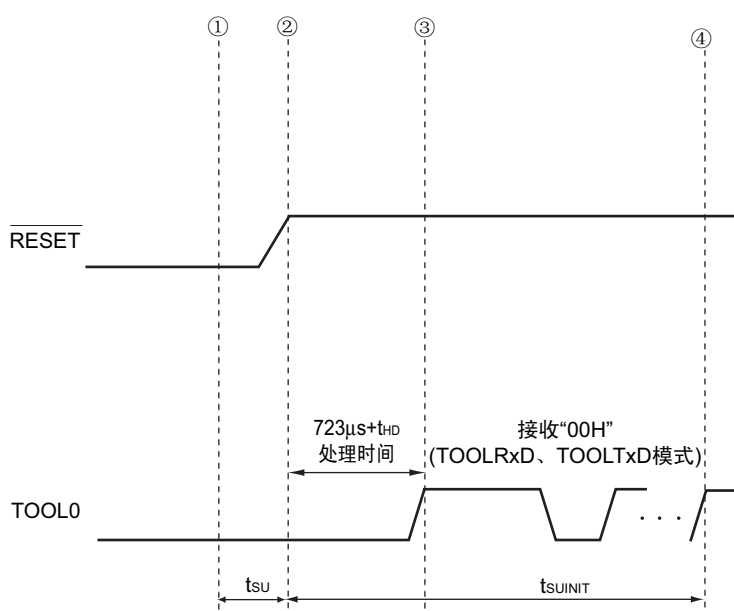
(T_A=-40 ~ +85°C、1.8V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
传送速率		串行编程时	115200		1000000	bps

31.10 闪存编程模式的进入时序

(T_A=−40 ~ +85°C、1.8V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5V、V_{SS}=EV_{SS0}=0V)

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 _{SUINIT}	在解除外部复位前，解除 POR 和 LVD 的复位。			100	ms
从将 TOOL0 引脚置为低电平到解除外部复位为止的时间	t _{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_{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。

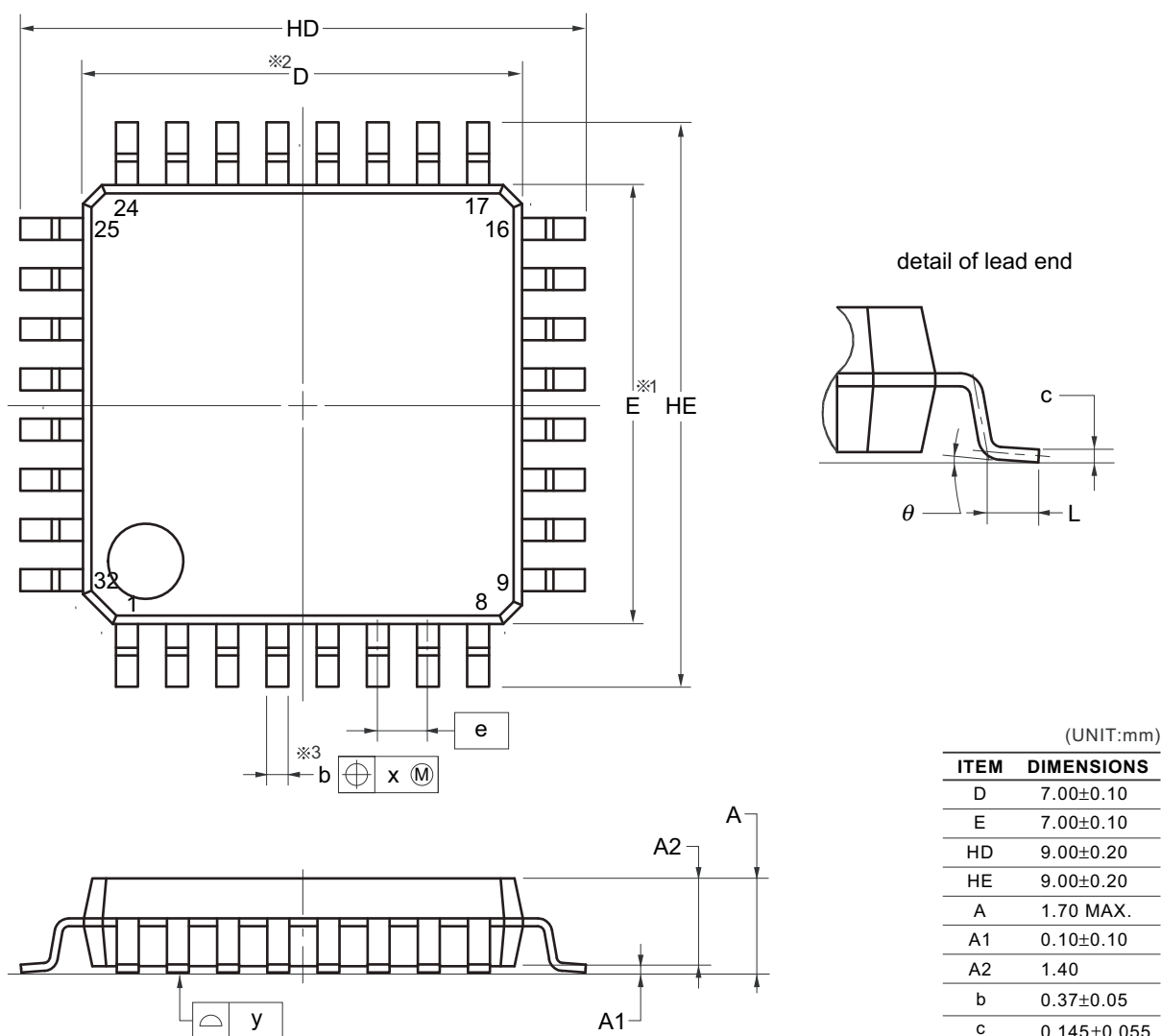
t_{HD}：这是在解除外部复位后保持 TOOL0 引脚低电平的时间（闪存固件处理时间除外）。

第 32 章 封装尺寸图

32.1 32 引脚产品

R7F0C014B2DFP

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP32-7x7-0.80	PLQP0032GB-A	P32GA-80-GBT-1	0.2



NOTE

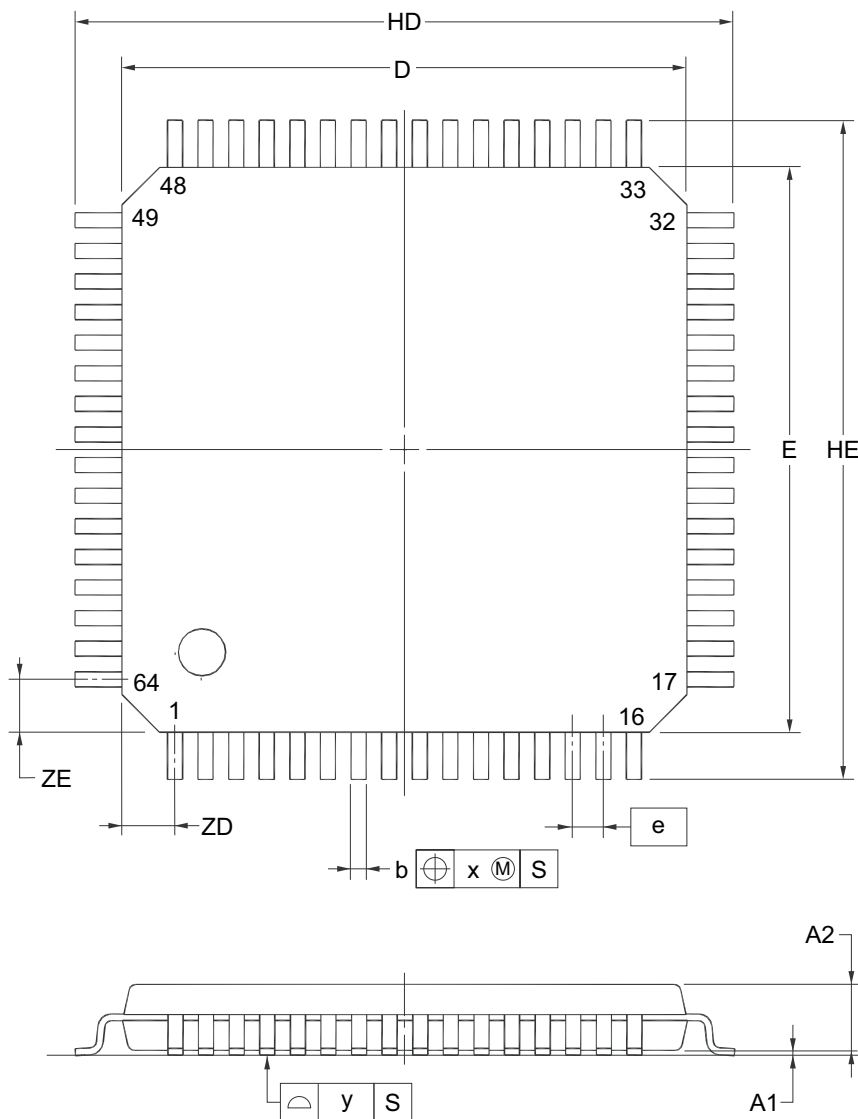
1. Dimensions "※1" and "※2" do not include mold flash.

2. Dimension "※3" does not include trim offset.

32.2 64 引脚产品

R7F0C014L2DFA

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP64-12x12-0.65	PLQP0064JA-A	P64GK-65-UET-2	0.51



detail of lead end

(UNIT:mm)

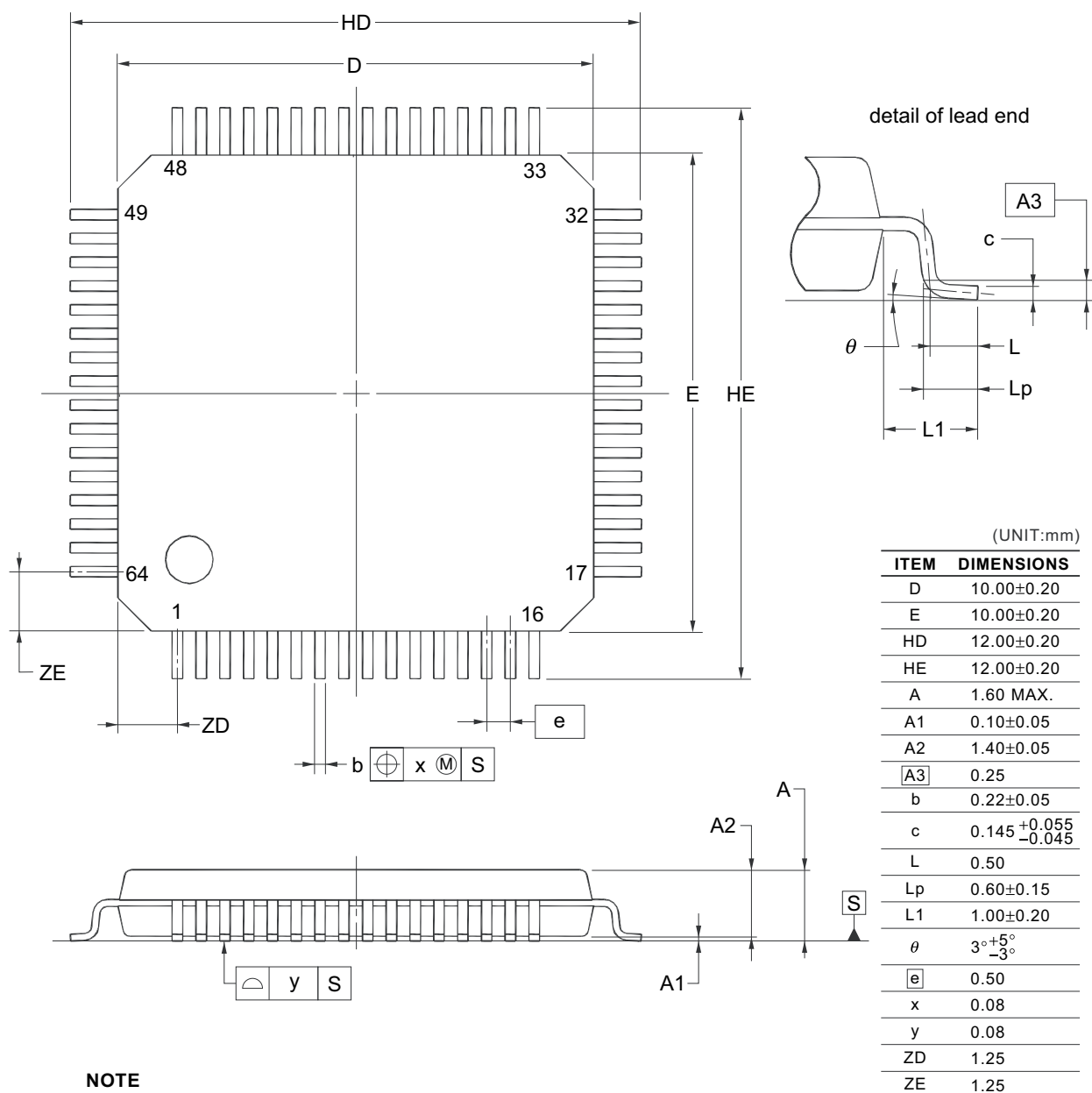
ITEM	DIMENSIONS
D	12.00±0.20
E	12.00±0.20
HD	14.00±0.20
HE	14.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.32 ^{+0.08} _{-0.07}
c	0.145 ^{+0.055} _{-0.045}
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	3° ^{+5°} _{-3°}
e	0.65
x	0.13
y	0.10
ZD	1.125
ZE	1.125

NOTE

Each lead centerline is located within 0.13 mm of its true position at maximum material condition.

R7F0C014L2DFB

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LFQFP64-10x10-0.50	PLQP0064KF-A	P64GB-50-UEU-2	0.35



附录 A 修订记录

Rev.	发行日	修订内容	
		页	修订处
1.00	2013.09.25	—	初版发行
2.20	2014.05.30	总体	将 $\overline{\text{ACK}}$ 改为 ACK。
		30	更改“3.1.5 扩展特殊功能寄存器（2nd SFR: 2nd Special Function Register）的区域”的内容。
		32	更改“(a) 中断允许标志（IE）”和“(b) 零标志（Z）”的内容。
		33	更改“(e) 服务优先级标志（ISP1、ISP0）”的内容。
		126	更改图 4-44。
		129	更改“4.3.5 端口输出模式寄存器（POM0、POM1、POM3、POM5、POM7）”的内容。
		131	更改图 4-49。
		134	更改“4.3.10 全局数字输入禁止寄存器（GDIDIS）”的内容。
		136、137	更改“4.4.4 通过 $\text{EV}_{\text{DD}} \leq \text{V}_{\text{DD}}$ 进行的不同电位（1.8V、2.5V、3V）的对应”和“4.4.5 通过输入 / 输出缓冲器进行的不同电位（1.8V、2.5V、3V）的对应”的内容。
		146	更改“(2) 副系统时钟”的内容。
		148	更改图 5-1。
		153	更改表 5-2 的内容。
		162	更改图 5-9。
		171	更改图 5-15。
		175	更改“5.6.3 XT1 振荡电路的设定例子”的内容。
		179、180	更改表 5-3 的 (3/5) 和 (4/5) 的内容。
		182	更改表 5-4(1/2) 的内容。
		185	更改“5.7 谐振器和振荡电路常数”的内容。
		190	修改“6.1.1 独立通道运行功能”的“(7) 延迟计数器”的图的笔误。
		194、195	更改图 6-2 ~ 图 6-5。
		201、202	更改图 6-10 的 (1/2) 和 (2/2)。
		204	更改图 6-11 的 (1/4)。
		211	更改图 6-15。
		212	更改图 6-16。
		218	更改“6.3.14 噪声滤波器允许寄存器 1（NFEN1）”的内容。
		232	更改图 6-31。
		239	更改图 6-40。
		244	更改图 6-44 的 (1/2)。
		265	更改图 6-64。
		296	更改图 7-6。
		302 ~ 304	更改图 7-11 ~ 图 7-13。
		305	更改图 7-15。
		306	更改图 7-16。
		337	更改图 8-24 的注 2。
		339	更改图 8-25 的注 1。
		341	更改图 8-26 的注 2。
		343	更改图 8-27 的注 1。
		350	更改表 8-4 的内容。
		354	更改表 8-7 的内容。
		358	更改“8.4 有关多个模式的共同事项”和“8.4.1 计数源”。

Rev.	发行日	修订内容	
		页	修订处
2.20	2014.05.30	359	更改表 8-10 的内容。
		365	修改表 8-11 的笔误。
		366	更改图 8-46。
		370	更改图 8-49。
		375	更改图 8-53。
		379	更改图 8-56。
		382	更改图 8-58。
		386	更改图 8-61。
		389、390	更改“8.6 定时器 RD 中断”的内容。
		392	更改“8.7.4 输入捕捉功能”的内容。
		397	更改“9.1 实时时钟的功能”的内容。
		436	更改“11.5 时钟输出 / 蜂鸣器输出控制电路的注意事项”的内容。
		444	更改“13.1 A/D 转换器的功能”的表。
		447	更改“(8) 控制电路”和“(9) AV _{REFP} 引脚”的内容。
		449	更改图 13-2。
		452	更改图 13-4。
		453 ~ 456	更改表 13-3 的 (1/4) ~ (4/4) 内容。
		458	更改图 13-6。
		459、460	更改“13.3.4 A/D 转换器的模式寄存器 2 (ADM2)”的内容、图 13-7 的 (1/2) 和 (2/2)。
		461	更改图 13-9。
		463	更改图 13-11(2/2)。
		470 ~ 481	更改图 13-17 ~ 图 13-28。
		482 ~ 485	更改图 13-29 ~ 图 13-32。
		490	更改图 13-37。
		495	修改图 13-44 的笔误, 更改“(6) 有关模拟输入 (ANIn) 引脚的输入阻抗”的内容。
		496	更改“(9) 有关 A/D 转换结果寄存器 (ADCR、ADCRH) 的读操作”的内容。
		499	更改“14.1.1 3 线串行 I/O (CSI00、CSI01、CSI10、CSI11、CSI20、CSI21)”的内容。
		500	更改“14.1.2 UART (UART0 ~ UART2)”的内容。
		501	更改“14.1.3 简易 I ² C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)”的内容。
		503	更改图 14-1。
		504	更改图 14-2。
		505	更改“14.2.2 串行数据寄存器 mn (SDRmn) 的低 8 位或者低 9 位”的内容。
		508	更改图 14-5。
		511	更改图 14-7(2/2)。
		512、513	更改图 14-8 的 (1/2) 和 (2/2)。
		514	更改“14.3.5 串行数据寄存器 mn (SDRmn)”的内容。
		515	更改图 14-10。
		516、517	更改图 14-11 的 (1/2) 和 (2/2)。
		522	更改“14.3.12 串行输出寄存器 m (SOM)”的内容和图 14-16。
		523	更改图 14-17。
		524	追加图 14-18。

Rev.	发行日	修订内容	
		页	修订处
2.20	2014.05.30	525	更改“14.3.14 串行待机控制寄存器 m (SSCm)”的内容和图 14-19。
		526	追加图 14-20, 更改“14.3.15 输入切换控制寄存器 (ISC)”的内容和图 14-21。
		527	更改“14.3.16 噪声滤波器允许寄存器 0 (NFEN0)”的内容和图 14-22。
		528	更改“14.3.17 控制串行输入 / 输出引脚端口功能的寄存器”的内容。
		—	删除旧版的“14.3.18 端口输出模式寄存器 0、1、3、5、7 (POM0、POM1、POM3、POM5、POM7)”和“14.3.19 端口模式寄存器 0、1、3、5 ~ 7 (PM0、PM1、PM3、PM5 ~ PM7)”的内容。
		530	更改图 14-24。
		531	更改“14.5 3 线串行 I/O (CSI00、CSI01、CSI10、CSI11、CSI20、CSI21) 通信的运行”的内容。
		534、535	更改图 14-25 和图 14-26。
		536 ~ 540	更改图 14-28 ~ 图 14-32。
		542 ~ 548	更改图 14-33 ~ 图 14-40。
		550、551	更改图 14-41 ~ 图 14-43。
		553 ~ 556	更改图 14-45 ~ 图 14-48。
		557	更改“14.5.4 从属发送”的内容。
		558 ~ 564	更改图 14-49 ~ 图 14-56。
		565	更改“14.5.5 从属接收”的内容。
		566、567	更改图 14-57 ~ 图 14-59。
		569、570	更改图 14-61 和图 14-62。
		571	更改“14.5.6 从属发送和接收”的内容。
		572、573	更改图 14-63 ~ 图 14-65。
		575 ~ 578	更改图 14-67 ~ 图 14-70。
		579	更改“14.5.7 SNOOZE 模式功能”的内容和图 14-71。
		580 ~ 582	更改图 14-72 ~ 图 14-74。
		584	更改表 14-2 的内容。
		586	更改“14.6 从属选择输入功能的时钟同步串行通信的运行”的内容。
		589	更改“14.6.1 从属发送”的内容。
		592 ~ 598	更改图 14-79 ~ 图 14-85。
		599	更改“14.6.2 从属接收”的内容。
		602	更改图 14-87 和图 14-88。
		604、605	更改图 14-90 和图 14-91。
		606	更改“14.6.3 从属发送和接收”的内容。
		609 ~ 615	更改图 14-93 ~ 图 14-99。
		616	更改“14.6.4 传送时钟频率的计算”和表 14-3 的内容。
		618	更改“14.7 UART (UART0 ~ UART2) 通信的运行”的内容。
		621	更改图 14-101(1/2)。
		623 ~ 628	更改图 14-102 ~ 图 14-108。
		630	更改图 14-109(1/2)。
		632	更改图 14-111。
		634、635	更改图 14-113 和图 14-114。
		636	更改“14.7.3 SNOOZE 模式功能”的内容。
		637	更改表 14-4 的内容。
		638、639	更改图 14-115 和图 14-116。
		641、642	更改图 14-118 和图 14-119。
		643	更改“14.7.4 波特率的计算”的内容。

Rev.	发行日	修订内容	
		页	修订处
2.20	2014.05.30	644	更改表 14-5 的内容。
		653	更改图 14-126。
		654	更改图 14-127。
		656	更改“14.9 简易 I ² C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21) 通信的运行”的内容。
		658	更改“14.9.1 地址段的发送”的内容。
		659	更改图 14-128。
		663	更改“14.9.2 数据发送”的内容。
		664	更改图 14-132。
		666	更改“14.9.3 数据接收”的内容。
		667	更改图 14-135。
		672	更改表 14-6 的内容。
		675	更改图 15-1。
		680	更改“15.3.1 外围允许寄存器 0 (PER0)”的内容和图 15-5。
		683、684	更改图 15-6 的 (3/4) 和 (4/4)。
		689、690	更改图 15-9 的 (1/2) 和 (2/2)。
		691	更改“15.3.6 IICA 低电平宽度设定寄存器 n (IICWLn)”和“15.3.7 IICA 高电平宽度设定寄存器 n (IICWHn)”的内容及图 15-11。
		694	修改“15.4.2 通过 IICWLn 寄存器和 IICWHn 寄存器设定传送时钟的方法”的笔误。
		707 ~ 709	更改图 15-22 ~ 图 15-24。
		710	更改“15.5.14 通信预约”的内容。
		712	更改图 15-27。
		713	更改“(2) 禁止通信预约功能的情况 (IICA 标志寄存器 n (IICFn) 的 bit0 (IICRSVn) = 1)”的内容。
		714	修改“15.5.15 其他注意事项”的“(3) 正在和其他设备进行 I ² C 通信的情况”的笔误。
		716	更改图 15-28。
		717 ~ 719	更改图 15-29 的 (1/3) ~ (3/3)。
		721	更改图 15-30(1)。
		769	更改“16.3.11 DTC 启动允许寄存器 i (DTCENi) (i=0 ~ 4)”的内容。
		770	更改表 16-5 的内容。
		772	更改图 16-13。
		781	更改“16.5.1 DTC 控制数据和向量表的设定”、“16.5.3 DTC 保留指令”和“16.5.4 存取数据闪存空间时的运行”的内容。
		811	更改表 18-3 的内容。
		817	更改表 18-5 的内容。
		833	更改表 20-2 的内容。
		834、835	更改图 20-3 的 (1/2) 和 (2/2)。
		837	更改“(1) SNOOZE 模式的设定和运行状态”的内容。
		838	更改表 20-3 的内容。
		839	追加“(2) 在 SNOOZE 模式中产生中断请求信号时的时序图”和“(3) 在 SNOOZE 模式中不产生中断请求信号时的时序图”的内容。
		842、843	更改“21.1 复位时序”的内容。
		851、852	更改图 22-2 的 (1/3) 和 (2/3)。
		854	更改“23.1 电压检测电路的功能”的内容。
		855	更改图 23-1。
		856	更改图 23-2。

Rev.	发行日	修订内容	
		页	修订处
2.20	2014.05.30	857	更改图 23-3。
		858、859	更改表 23-1 的 (1/2) 和 (2/2) 的内容。
		861	更改图 23-4。
		862	更改“23.4.2 用作中断模式时的设定”的内容。
		863	更改图 23-5。
		864	更改“23.4.3 用作中断 & 复位模式时的设定”的内容。
		878	更改图 24-7。
		882	更改图 24-11。
		884	更改“24.3.7 频率检测功能”的内容和图 24-13。
		885	更改“24.3.7.1 定时器输入选择寄存器 0 (TIS0)”的内容。
		891	更改图 24-18。
		893	更改“26.1 选项字节的功能”和“26.1.1 用户选项字节 (000C0H ~ 000C2H/010C0H ~ 010C2H)”的内容。
		896 ~ 899	更改图 26-2 的 (1/4) ~ (4/4)。
		903	更改说明的内容。
		909	更改“27.3.1 P40/TOOL0 引脚”的内容。
		913	更改“27.4.2 串行编程模式”和表 27-5 的内容。
		917	更改表 27-10 的内容。
		918	更改“27.6 自编程”的内容。
		921	更改图 27-10。
		924	更改“27.7 安全设定”的表号。
		925	更改“27.8.1 数据闪存的概要”的内容。
		926	更改“27.8.3 数据闪存的存取步骤”的内容。
		927	更改图 28-1。
		933	更改表 30-1 的内容。
		936 ~ 953	更改表 30-5 的 (1/18) ~ (18/18) 的注 2。
		1011	更改“31.6.3 POR 电路特性”的内容。
		1013	更改“31.7 数据存储器的保持特性”的内容。

R7F0C014B2D、R7F0C014L2D
用户手册 硬件篇

Publication Date: Rev.2.20 May 30, 2014

Published by: Renesas Electronics Corporation



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics America Inc.

2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited

1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.

Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.

Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited

Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.

80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.

12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

R7F0C014B2D、 R7F0C014L2D



瑞萨电子株式会社

R01UH0442CJ0220