

R7F0C011B、 R7F0C012B、 R7F0C013B

用户手册 硬件篇

8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、复制或或以其他方式部分或全部地非法使用瑞萨电子的任何产品。对于用户或第三方因上述更改、修改、复制或以其他方式非法使用瑞萨电子产品的行为而遭受的任何损失，瑞萨电子不承担任何责任。
5. 瑞萨电子产品根据其质量等级分为两个等级：“标准等级”和“高质量等级”。每种瑞萨电子产品的推荐用途均取决于产品的质量等级，如下所示：
标准等级： 计算机、办公设备、通讯设备、测试和测量设备、视听设备、家用电器、机械工具、个人电子设备以及工业机器人等。
高质量等级： 运输设备（汽车、火车、轮船等）、交通控制系统、防灾系统、预防犯罪系统以及安全设备等。
瑞萨电子产品无意用于且未被授权用于可能对人类生命造成直接威胁的产品或系统及可能造成人身伤害的产品或系统（人工生命维持装置或系统、植埋于体内的装置等）中，或者可能造成重大财产损失的产品或系统（核反应堆控制系统、军用设备等）中。在将每种瑞萨电子产品用于某种特定应用之前，用户应先确认其质量等级。不得将瑞萨电子产品用于超出其设计用途之外的任何应用。对于用户或第三方因将瑞萨电子产品用于其设计用途之外而遭受的任何损害或损失，瑞萨电子不承担任何责任。
6. 使用本文档中记载的瑞萨电子产品时，应在瑞萨电子指定的范围内，特别是在最大额定值、电源工作电压范围、移动电源电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用瑞萨电子产品而产生的故障或损失，瑞萨电子不承担任何责任。
7. 虽然瑞萨电子一直致力于提高瑞萨电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，瑞萨电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当瑞萨电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。由于难于对微机软件单独进行评估，所以请用户自行对最终产品或系统进行安全评估。
8. 关于环境保护方面的详细内容，例如每种瑞萨电子产品的环境兼容性等，请与瑞萨电子的营业部门联系。使用瑞萨电子产品时，请遵守对管制物质的使用或含量进行管理的所有相应法律法规（包括但不限于《欧盟 RoHS 指令》）。对于因用户未遵守相应法律法规而导致的损害或损失，瑞萨电子不承担任何责任。
9. 不可将瑞萨电子产品和技术用于或者嵌入日本国内或海外相应的法律法规所禁止生产、使用及销售的任何产品或系统中。也不可将本文档中记载的瑞萨电子产品或技术用于与军事应用或者军事用途有关的任何目的（如大规模杀伤性武器的开发等）。在将本文档中记载的瑞萨电子产品或技术进行出口时，应当遵守相应的出口管制法律法规，并按照上述法律法规所规定的程序进行。
10. 向第三方分销或处分产品或以其他方式将产品置于第三方控制之下的瑞萨电子产品买方或分销商，有责任事先向上述第三方通知本文档规定的内容和条件；对于用户或第三方因非法使用瑞萨电子产品而遭受的任何损失，瑞萨电子不承担任何责任。
11. 在事先未得到瑞萨电子书面认可的情况下，不得以任何形式部分或全部转载或复制本文档。
12. 如果对本文档所记载的信息或瑞萨电子产品有任何疑问，或者用户有任何其他疑问，请向瑞萨电子的营业部门咨询。
(注 1) 瑞萨电子：在本文档中指瑞萨电子株式会社及其控股子公司。
(注 2) 瑞萨电子产品：指瑞萨电子开发或生产的任何产品。

关于 CMOS 器件的注意事项

① 输入引脚处的施加电压波形

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

② 未使用的输入引脚的处理

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

③ ESD 防护措施

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

④ 初始化之前的状态

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

⑤ 电源上电 / 断电序列

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

⑥ 断电状态期间的信号输入

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

本手册的使用方法

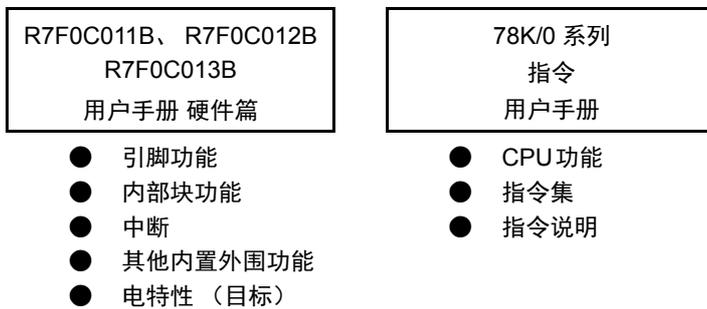
对象 本手册以理解 R7F0C011B、R7F0C012B、R7F0C013B 的功能并且设计和开发其应用系统和程序的用户工程师为对象。

对象产品如下：

- R7F0C011B2DFP
- R7F0C012B2DFP
- R7F0C013B2DFP

目的 本手册以帮助用户理解下述结构中所示的功能为目的。

结构 R7F0C011B、R7F0C012B、R7F0C013B 的用户手册分为 2 本，分别是用户手册硬件篇（本手册）和指令用户手册（78K/0 系列通用）。



阅读方法 阅读本手册的读者应具备电气、逻辑电路以及微控制器的基础知识。

要理解全部功能时

→ 请按照目录的顺序阅读本手册。

寄存器格式的阅读方法

→ 关于使用“<>”的位号，其位名称在 RA78K0 中定义为保留字，在 CC78K0 中通过 #pragma sfr 指令定义为 sfr 变量。

要了解 R7F0C011B、R7F0C012B、R7F0C013B 的指令功能的详细内容时

→ 请参照《78K/0 系列 指令 用户手册》（U12326C）

凡例

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

相关文档

相关文档中可能包括暂定版。但是，在以下文档中并未特别注明“暂定版”，请谅解。

设备的相关文档

文档名称	文档编号
R7F0C011B、R7F0C012B、R7F0C013B 微控制器 用户手册 硬件篇	本手册
78K/0 系列 指令 用户手册	U12326C
78K0/Kx2 Flash 存储器编程器（编程器）应用笔记	U17739C

闪存编程的相关文档（用户手册）

文档名称	文档编号
PG-FP5 Flash Memory Programmer	R20UT0008E
QB-MINI2 On-chip Debug Emulator with Programming User's Manual	R20UT0449E
QB-Programmer Programming GUI Operation	U18527E

开发工具（硬件）的相关文档（用户手册）

文档名称	文档编号
QB-MINI2 On-chip Debug Emulator with Programming User's Manual	R20UT0449E

注意 上述相关文档的内容如有变更，恕不另行通知。设计等时，请使用各文档的最新版本。

开发工具（软件）的相关文档

文档名称		文档编号
RA78K0 Ver.3.80 Assembler Package User's Manual 注 1	Operation	U17199E
	Language	U17198E
	Structured Assembly Language	U17197E
78K0 Assembler Package RA78K0 Ver.4.01 Operation Precautions (notification document) 注 1		ZUD-CD-07-0181-E
CC78K0 Ver.3.70 C Compiler User's Manual 注 2	Operation	U17201E
	Language	U17200E
78K0 C Compiler CC78K0 Ver.4.00 Operation Precautions (notification document) 注 2		ZUD-CD-07-0103-E
SM+ System Simulator User's Manual	Operation	U18601E
	User Open Interface	U18212E
ID78K0-QB Ver.2.94 Integrated Debugger User's Manual	Operation	U18330E
ID78K0-QB Ver.3.00 Integrated Debugger User's Manual	Operation	U18492E
PM+ Ver.5.20 注 3 User's Manual		U16934E
PM+ Ver.6.30 注 4 User's Manual		U18416E

- 注 1. 该文件在安装 RA78K0 Ver.4.01 时，与工具一起安装到 PC 内。关于“78K0 Assembler Package RA78K0 Ver.4.01 Operation Precautions (notification document)”中未记载的内容，请参照“RA78K0 Ver.3.80 User's Manual”。
2. 该文件在安装 CC78K0 Ver.4.00 时，与工具一起安装到 PC 内。关于“78K0 Assembler Package RA78K0 Ver.4.01 Operation Precautions (notification document)”中未记载的内容，请参照“CC78K0 Ver.3.70 User's Manual”。
3. PM+Ver.5.20 是附属在 RA78K0 Ver.3.80 中的集成开发环境。
4. PM+Ver.6.30 是附属在 RA78K0 Ver.4.01 中的集成开发环境。可管理多个不同版本的软件工具（汇编程序、C 编译器、调试程序以及仿真器）产品。

其他文档

文档名称	文档编号
Renesas Microcomputer General Catalog	R01CS0001E
Renesas Semiconductor Package Mount Manual	注
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Derices by Electrostatic Discharge (ESD)	C11892E

注 请参照“Renesas Semiconductor Package Mount Manual”网站。
(<http://www.renesas.com/products/package/manual/index.jsp>)

注意 上述相关文档的内容如有变更，恕不另行通知。设计等时，请使用各文档的最新版本。

所有商标及注册商标分别归属于其所有者。

本用户手册仅为参考译文，对应的日文版和英文版具有正式效力。

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

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

目 录

第 1 章 概述	1
1.1 特点	1
1.2 应用领域	2
1.3 订购信息	2
1.4 引脚连接图（俯视图）	2
1.5 引脚名称	4
1.6 框图	5
1.7 功能概要	7
第 2 章 引脚功能	9
2.1 引脚功能一览	9
2.2 引脚功能的说明	13
2.2.1 P00、P01（Port0）	13
2.2.2 P10 ~ P17（Port1）	13
2.2.3 P20 ~ P23（Port2）	14
2.2.4 P30 ~ P33（Port3）	15
2.2.5 P40、P41（Port4）	15
2.2.6 P60、P61（Port6）	15
2.2.7 P70、P71（Port7）	16
2.2.8 P120 ~ P122（Port12）	16
2.2.9 V_{DD} 、 V_{SS}	17
2.2.10 RESET	17
2.2.11 REGC	17
2.2.12 FLMD0	17
2.3 引脚的输入 / 输出电路和未使用引脚的处理	18
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 内部数据存储空间	27
3.1.3 特殊功能寄存器（SFR: Special Function Register）区域	27
3.1.4 数据存储寻址	28
3.2 处理器寄存器	31
3.2.1 控制寄存器	31
3.2.2 通用寄存器	35
3.2.3 特殊功能寄存器（SFR: Special Function Register）	36
3.3 指令地址寻址	40
3.3.1 相对寻址	40
3.3.2 立即寻址	41
3.3.3 表间接寻址	42
3.3.4 寄存器寻址	42
3.4 操作数地址寻址	43
3.4.1 隐含寻址	43
3.4.2 寄存器寻址	44
3.4.3 直接寻址	45
3.4.4 短直接寻址	46
3.4.5 特殊功能寄存器（SFR）寻址	47
3.4.6 寄存器间接寻址	48
3.4.7 基址寻址	49

3.4.8	基址变址寻址	50
3.4.9	堆栈寻址	51
第 4 章	端口功能	52
4.1	端口功能	52
4.2	端口结构	54
4.2.1	端口 0	54
4.2.2	端口 1	56
4.2.3	端口 2	61
4.2.4	端口 3	62
4.2.5	端口 4	64
4.2.6	端口 6	65
4.2.7	端口 7	66
4.2.8	端口 12	67
4.3	端口功能的控制寄存器	69
4.4	端口功能的运行	73
4.4.1	输入 / 输出端口的写入	73
4.4.2	输入 / 输出端口的读取	73
4.4.3	输入 / 输出端口的运算	73
4.5	使用复用功能时的端口模式寄存器和输出锁存器的设置	74
4.6	端口寄存器 n (Pn) 的 1 位存储器操作指令的注意事项	76
第 5 章	时钟发生电路	77
5.1	时钟发生电路的功能	77
5.2	时钟发生电路的结构	78
5.3	时钟发生电路的控制寄存器	80
5.4	系统时钟振荡电路	87
5.4.1	X1 振荡电路	87
5.4.2	高速内部振荡电路	89
5.4.3	低速内部振荡电路	89
5.4.4	预分频器	89
5.5	时钟发生电路的运行	90
5.6	时钟的控制	93
5.6.1	高速系统时钟的控制示例	93
5.6.2	高速内部振荡时钟的控制示例	96
5.6.3	低速内部振荡时钟的控制示例	98
5.6.4	作为 CPU 时钟和外围硬件时钟提供的时钟	98
5.6.5	CPU 时钟状态转移图	99
5.6.6	CPU 时钟的转移前条件和转移后处理	102
5.6.7	CPU 时钟的切换和主系统时钟的切换所需时间	102
5.6.8	时钟振荡停止前的条件	103
5.6.9	外围硬件和源时钟	104
第 6 章	16 位定时器 / 事件计数器 00	105
6.1	16 位定时器 / 事件计数器 00 的功能	105
6.2	16 位定时器 / 事件计数器 00 的结构	106
6.3	16 位定时器 / 事件计数器 00 的控制寄存器	110
6.4	16 位定时器 / 事件计数器 00 的运行	116
6.4.1	作为间隔定时器的运行	116
6.4.2	方波的输出运行	119
6.4.3	作为外部事件计数器的运行	123
6.4.4	作为输入 TI000 引脚有效边沿时的清除 & 启动模式的运行	126

6.4.5	作为自由运行定时器的运行	138
6.4.6	PPG 的输出运行	146
6.4.7	单次脉冲的输出运行	149
6.4.8	脉冲的测量运行	153
6.5	TM00 的特殊使用方法	160
6.5.1	在 TM00 运行期间改写 CR010	160
6.5.2	LVS00 和 LVR00 的设置	160
6.6	16 位定时器 / 事件计数器 00 的注意事项	162
第 7 章	8 位定时器 / 事件计数器 50、51	166
7.1	8 位定时器 / 事件计数器 50、51 的功能	166
7.2	8 位定时器 / 事件计数器 50、51 的结构	166
7.3	8 位定时器 / 事件计数器 50、51 的控制寄存器	169
7.4	8 位定时器 / 事件计数器 50、51 的运行	174
7.4.1	作为间隔定时器的运行	174
7.4.2	作为外部事件计数器的运行	176
7.4.3	方波的输出运行	177
7.4.4	PWM 的输出运行	179
7.5	8 位定时器 / 事件计数器 50、51 的注意事项	182
第 8 章	8 位定时器 H0、H1	183
8.1	8 位定时器 H0、H1 的功能	183
8.2	8 位定时器 H0、H1 的结构	183
8.3	8 位定时器 H0、H1 的控制寄存器	187
8.4	8 位定时器 H0、H1 的运行	191
8.4.1	作为间隔定时器 / 方波输出的运行	191
8.4.2	PWM 的输出运行	194
8.4.3	作为载波发生器的运行（只限 8 位定时器 H1）	200
第 9 章	时钟定时器	206
9.1	时钟定时器的功能	206
9.2	时钟定时器的结构	207
9.3	时钟定时器的控制寄存器	208
9.4	时钟定时器的运行	209
9.4.1	作为时钟定时器的运行	209
9.4.2	作为间隔定时器的运行	209
9.5	时钟定时器的注意事项	210
第 10 章	看门狗定时器	211
10.1	看门狗定时器的功能	211
10.2	看门狗定时器的结构	212
10.3	看门狗定时器的控制寄存器	213
10.4	看门狗定时器的运行	214
10.4.1	看门狗定时器的运行控制	214
10.4.2	看门狗定时器的上溢时间设置	215
10.4.3	看门狗定时器的窗口打开期间设置	216
第 11 章	A/D 转换器	217
11.1	A/D 转换器的功能	217
11.2	A/D 转换器的结构	218
11.3	A/D 转换器使用的寄存器	220
11.4	A/D 转换器的运行	226

11.4.1	A/D 转换器的基本运行	226
11.4.2	输入电压和转换结果	227
11.4.3	A/D 转换器的运行模式	228
11.5	A/D 转换器特性表的阅读方法	230
11.6	A/D 转换器的注意事项	234
第 12 章	串行接口 UART0	237
12.1	串行接口 UART0 的功能	237
12.2	串行接口 UART0 的结构	238
12.3	串行接口 UART0 的控制寄存器	240
12.4	串行接口 UART0 的运行	245
12.4.1	运行停止模式	245
12.4.2	异步串行接口 (UART) 模式	246
12.4.3	专用波特率发生器	251
12.4.4	波特率的计算	252
第 13 章	串行接口 UART6	256
13.1	串行接口 UART6 的功能	256
13.2	串行接口 UART6 的结构	260
13.3	串行接口 UART6 的控制寄存器	263
13.4	串行接口 UART6 的运行	272
13.4.1	运行停止模式	272
13.4.2	异步串行接口 (UART) 模式	273
13.4.3	专用波特率发生器	284
13.4.4	波特率的计算	285
第 14 章	串行接口 CSII0	289
14.1	串行接口 CSII0 的功能	289
14.2	串行接口 CSII0 的结构	289
14.3	串行接口 CSII0 的控制寄存器	291
14.4	串行接口 CSII0 的运行	294
14.4.1	运行停止模式	294
14.4.2	3 线式串行 I/O 模式	295
第 15 章	串行接口 IIC0	302
15.1	串行接口 IIC0 的功能	302
15.2	串行接口 IIC0 的结构	305
15.3	串行接口 IIC0 的控制寄存器	308
15.4	I ² C 总线模式的功能	321
15.4.1	引脚结构	321
15.5	I ² C 总线的定义和控制方法	322
15.5.1	开始条件	322
15.5.2	地址	323
15.5.3	传送方向的指定	323
15.5.4	应答 (ACK)	324
15.5.5	停止条件	325
15.5.6	等待	326
15.5.7	等待的解除方法	328
15.5.8	中断请求 (INTIIC0) 的产生时序和等待控制	329
15.5.9	地址匹配的检测方法	330
15.5.10	错误的检测	330
15.5.11	扩展码	331

15.5.12	仲裁	332
15.5.13	唤醒功能	334
15.5.14	通信预约	334
15.5.15	其他注意事项	337
15.5.16	通信运行	338
15.5.17	I ² C 中断请求 (INTIIC0) 的产生时序	346
15.6	时序图	365
第 16 章	中断功能	372
16.1	中断功能的种类	372
16.2	中断源和结构	373
16.3	中断功能的控制寄存器	375
16.4	中断处理的运行	381
16.4.1	可屏蔽中断请求的接受运行	381
16.4.2	软件中断请求的接受运行	383
16.4.3	多重中断处理	384
16.4.4	中断请求的保留	387
第 17 章	待机功能	388
17.1	待机功能和结构	388
17.1.1	待机功能	388
17.1.2	待机功能的控制寄存器	389
17.2	待机功能的运行	391
17.2.1	HALT 模式	391
17.2.2	STOP 模式	394
第 18 章	复位功能	399
18.1	复位源的确认寄存器	406
第 19 章	上电清除电路	407
19.1	上电清除电路的功能	407
19.2	上电清除电路的结构	407
19.3	上电清除电路的运行	408
19.4	上电清除电路的注意事项	411
第 20 章	低电压检测电路	413
20.1	低电压检测电路的功能	413
20.2	低电压检测电路的结构	414
20.3	低电压检测电路的控制寄存器	414
20.4	低电压检测电路的运行	417
20.4.1	用作复位时的设置	417
20.4.2	用作中断时的设置	422
20.5	低电压检测电路的注意事项	427
第 21 章	选项字节	430
21.1	选项字节的功能	430
21.2	选项字节的格式	431
第 22 章	闪存	434
22.1	存储容量切换寄存器	434
22.2	闪存编程器的编程方法	435
22.3	编程环境	435
22.4	通信方式	436

22.5	板上引脚处理	438
22.5.1	FLMD0 引脚的连接例子	438
22.5.2	串行接口引脚	438
22.5.3	RESET 引脚	440
22.5.4	端口引脚	440
22.5.5	REGC 引脚	440
22.5.6	其他信号的引脚	440
22.5.7	电源	440
22.6	编程方法	441
22.6.1	闪存控制	441
22.6.2	闪存编程模式	441
22.6.3	通信方式的选择	442
22.6.4	通信命令	442
22.7	安全设置	444
第 23 章	片上调试功能（只限 R7F0C999B）	446
23.1	QB-MINI2 与 R7F0C999B 的连接	446
23.2	片上调试安全 ID	450
23.3	保留用户资源	450
第 24 章	指令集	451
24.1	凡例	451
24.1.1	操作数的标识符和描述方法	451
24.1.2	操作栏的说明	452
24.1.3	标志栏的说明	452
24.2	操作一览	453
24.3	各寻址的指令一览	461
第 25 章	电特性（目标）	464
第 26 章	封装尺寸图	479
第 27 章	等待的注意事项	480
27.1	等待的注意事项	480
27.2	产生等待的外围硬件	480

第 1 章 概述

1.1 特点

- 最短指令执行时间：0.2 μ s（以高速系统时钟 10MHz 运行时）
- 通用寄存器：8 位 \times 32 个寄存器（8 位 \times 8 个寄存器 \times 4 组）
- I/O 端口、ROM 容量、RAM 容量

项目 产品名称	I/O 端口	程序存储器 (闪存)	数据存储器 (内部高速 RAM)
R7F0C011B2001DFP	27 个 (CMOS 输入 / 输出: 25 个、N 沟道漏极开路输入 / 输出: 2 个)	16K 字节	768 字节
R7F0C012B2001DFP		24K 字节	1K 字节
R7F0C013B2001DFP		32K 字节	1K 字节
R7F0C999B2DFP 注			

- 内置单电源闪存
- 内置上电清除 (POC) 电路和低电压检测 (LVI) 电路
- 内置看门狗定时器 (能以低速内部振荡时钟运行)
- 定时器
 - 16 位定时器/事件计数器 ...PPG 输出、捕捉输入、外部事件计数器输入
 - 8 位定时器 H0、H1 ...PWM 输出、能以低速内部振荡时钟运行
 - 8 位定时器/事件计数器 50、51 ... 外部事件计数器输入
 - 时钟定时器
 - 看门狗定时器 ... 能以低速内部振荡时钟运行

项目 产品名称	16 位定时器 / 事件计数器	8 位定时器	时钟定时器	看门狗定时器
R7F0C011B2001DFP	1 个通道	定时器 H: 2 个通道 定时器 5: 2 个通道	1 个通道	1 个通道
R7F0C012B2001DFP				
R7F0C013B2001DFP				
R7F0C999B2DFP 注				

- 串行接口
 - UART ... 支持异步通信的双线式串行接口
 - CSI ... 支持时钟同步通信的 3 线式串行接口
 - IICA ... 支持时钟同步通信的双线式串行接口、支持多主设备、在从属模式中可通过地址匹配解除待机。

项目 产品名称	UART6	CSI10/UART0	IIC
R7F0C011B2001DFP	1 个通道	1 个通道	1 个通道
R7F0C012B2001DFP			
R7F0C013B2001DFP			
R7F0C999B2DFP 注			

注 R7F0C999B2DFP 是仅为使用片上调试功能的产品。

- 10 位分辨率 A/D 转换器 ($AV_{REF}=4.0\sim 5.5V$): 4 个通道
- 电源电压: $V_{DD}=4.0\sim 5.5V$
- 工作环境温度: $T_A=-40\sim +85^{\circ}C$

1.2 应用领域

- 家电产品
 - 空调

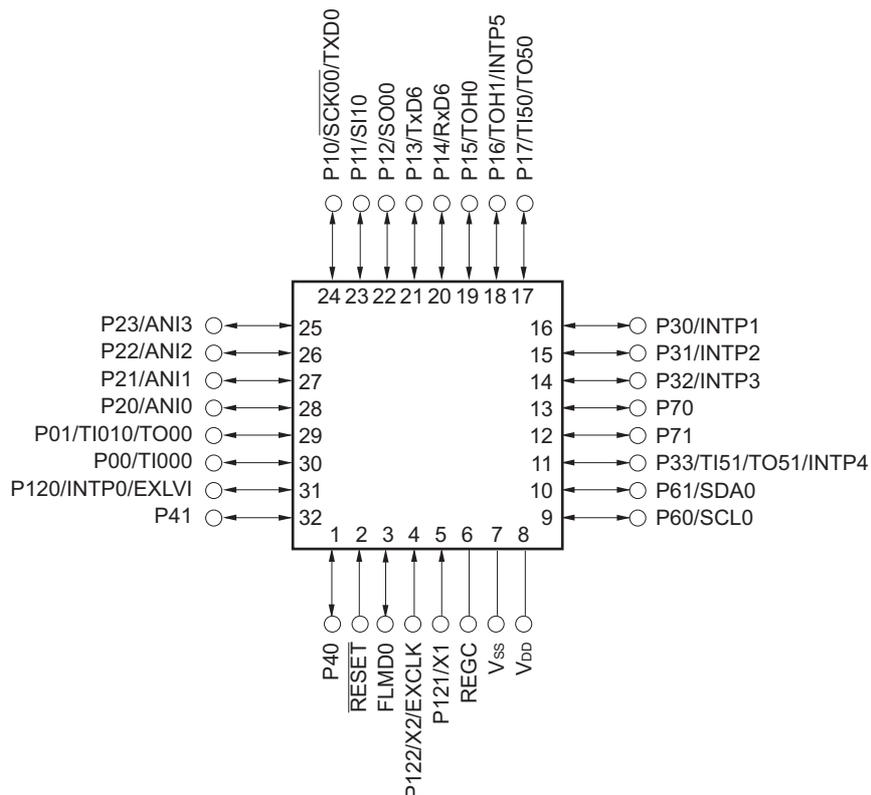
1.3 订购信息

引脚数	封装	ROM	RAM	半导体材料	订购名称
32 引脚	32 引脚塑封 LQFP (7×7)	16KB	768B	无铅产品 (外部引脚为镀 Ni/Pd/Au)	R7F0C011B2001DFP
		24KB	1KB		R7F0C012B2001DFP
		32KB	1KB		R7F0C013B2001DFP
		32KB	1KB		R7F0C999B2DFP (片上调试器专用)

1.4 引脚连接图（俯视图）

(1) R7F0C011B、R7F0C012B、R7F0C013B

- 32 引脚塑封 LQFP（细间距）（7×7）

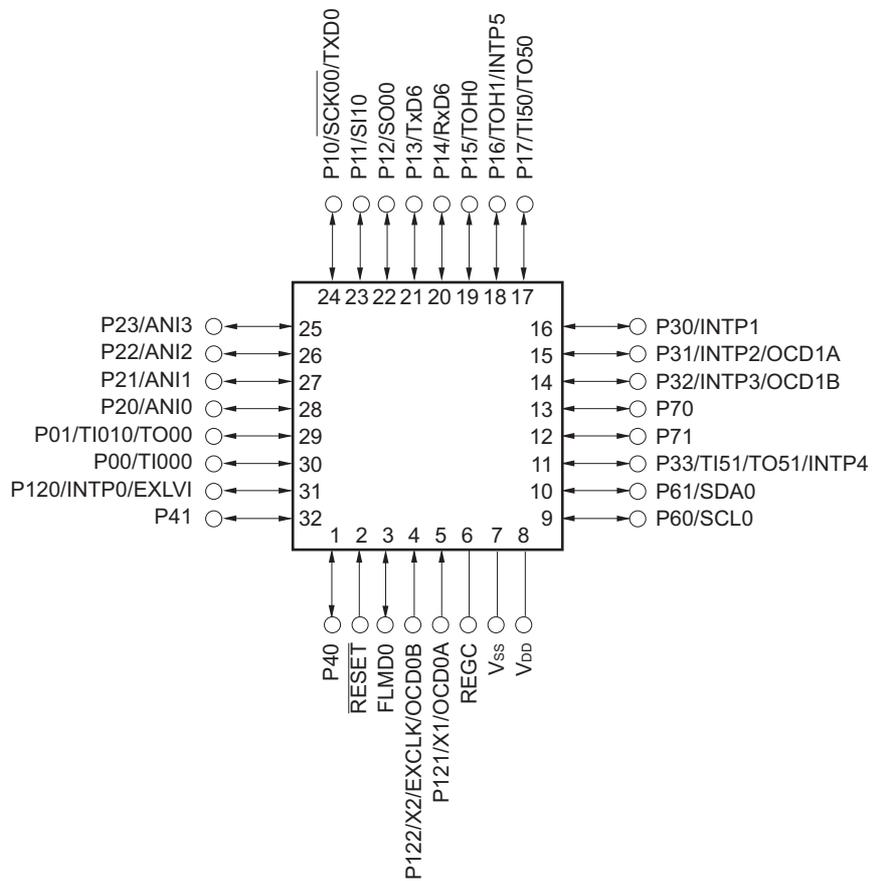


- 注意 1. REGC 必须通过电容器（0.47 ~ 1 μ F）连接到 Vss。
2. ANI0/P20 ~ ANI3/P23 在复位解除后变为模拟输入模式。

备注 引脚名称请参照“1.5 引脚名称”。

(2) R7F0C999B

- 32 引脚塑封 LQFP（细间距）（7×7）



- 注意 1. REGC 必须通过电容器（0.47 ~ 1 μ F）连接到 V_{SS}。
2. ANI0/P20 ~ ANI3/P23 在复位解除后变为模拟输入模式。

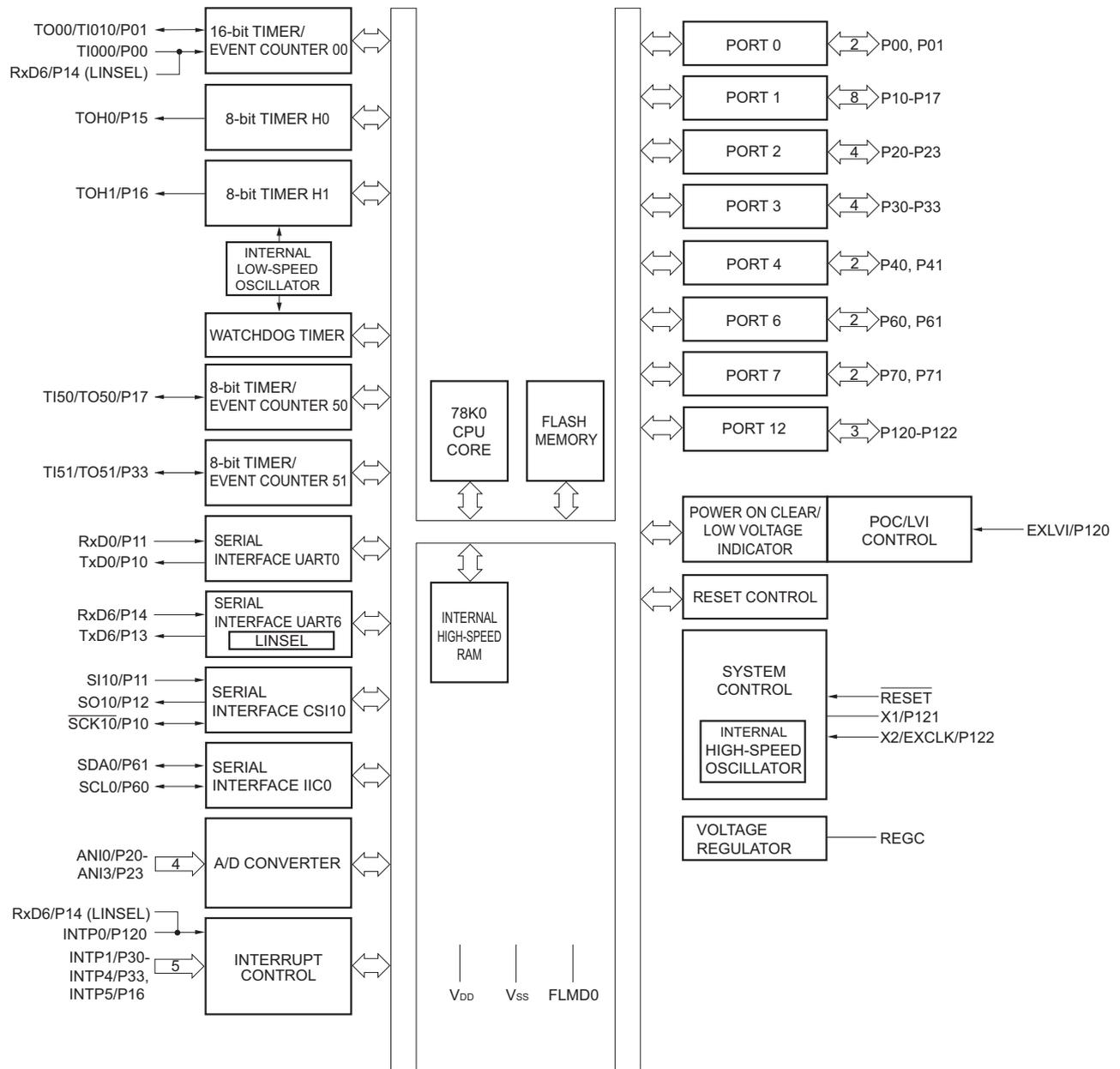
备注 引脚名称请参照“1.5 引脚名称”。

1.5 引脚名称

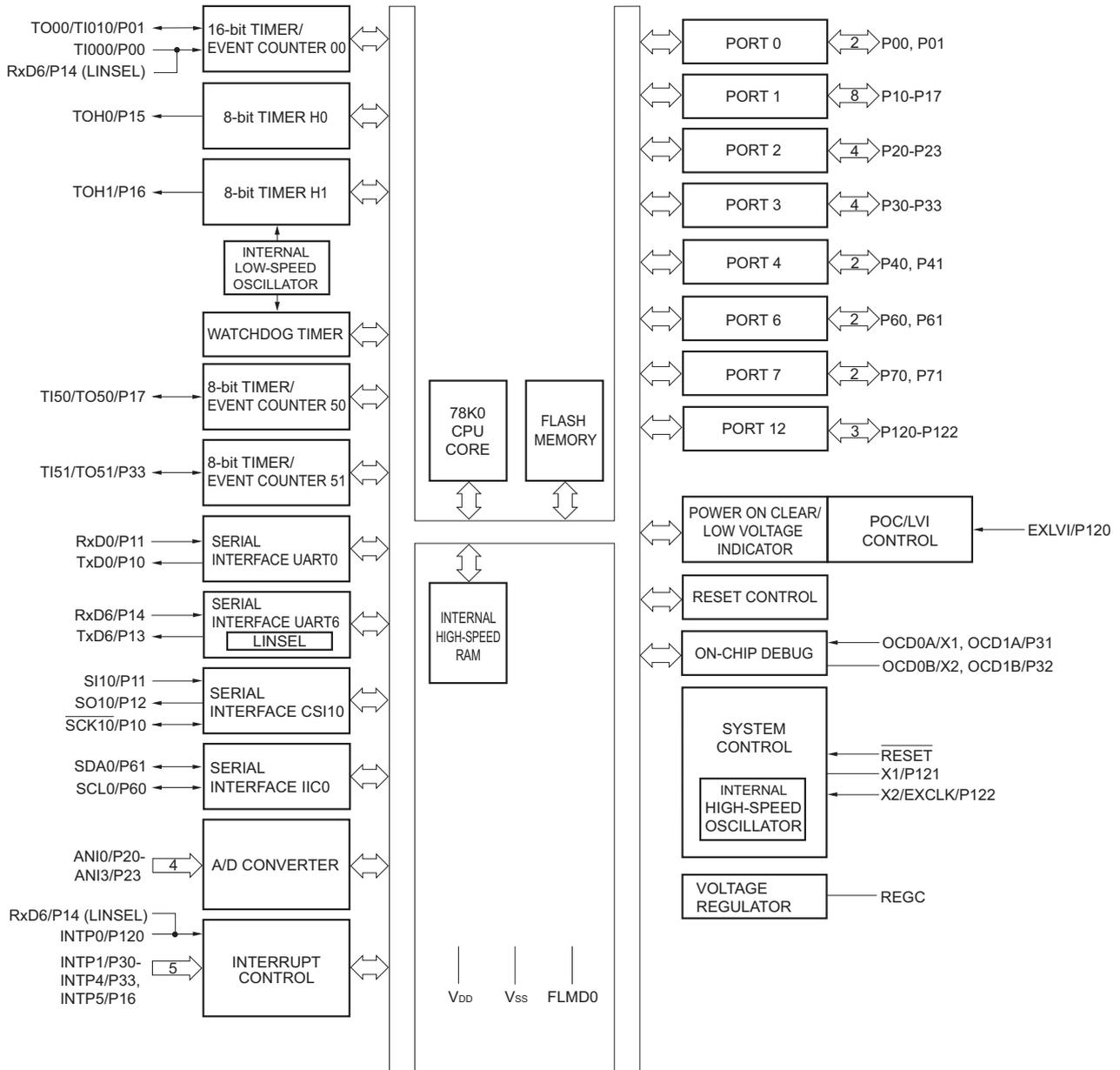
ANI0 ~ ANI3:	Analog Input	REGC:	Regulator Capacitance
EXCLK:	External Clock Input	RESET:	Reset
EXLVI:	External Potential Input for Low-voltage detector	RxD0、RxD6:	Receive Data
FLMD0:	Flash Programming Mode	SCK10:	Serial Clock Input/Output
INTP0、INTP1:	External Interrupt Input	SCL0:	Serial Clock Input/Output
INTP2 ~ INTP5:	External Interrupt Input	SDA0:	Serial Data Input/Output
OCD0A、OCD0B:	On-chip Debug Input/Output	SI10:	Serial Data Input
OCD1A、OCD1B:	On-chip Debug Input/Output	SO10:	Serial Data Output
P00、P01:	Port0	TI000、TI010:	Timer Input
P10 ~ P17:	Port1	TI50、TI51:	Timer Input
P20 ~ P23:	Port2	TO00、TO01:	Timer Output
P30 ~ P33:	Port3	TO50、TO51:	Timer Output
P40、P41:	Port4	TOH0、TOH1:	Timer Output
P60、P61:	Port6	TxD0、TxD6:	Transmit Data
P70、P71:	Port7	V _{DD} :	Power Supply
P120 ~ P122:	Port12	V _{SS} :	Ground
		X1、X2:	Crystal Oscillator (System Clock)

1.6 框图

(1) R7F0C011B、R7F0C012B、R7F0C013B



(2) R7F0C999B



1.7 功能概要

项目	R7F0C011B2001DFP	R7F0C012B2001DFP	R7F0C013B2001DFP	R7F0C999B2DFP 注 1
闪存 (K 字节)	16	24	32	32
高速 RAM (K 字节)	0.75	1	1	1
电源电压	V _{DD} =4.0 ~ 5.5V			
稳压器	内置			
最短指令执行时间	0.2μs (10MHz: V _{DD} =4.0 ~ 5.5V)			
时钟	高速系统	10MHz: V _{DD} =4.0 ~ 5.5V		
	高速内部振荡器	8MHz(TYP.): V _{DD} =4.0 ~ 5.5V		
	低速内部振荡器	240kHz(TYP.): V _{DD} =4.0 ~ 5.5V		
端口	总计	27		
	N-ch O.D. (6V 耐压)	2		
定时器	16 位 (TM0)	1ch		
	8 位 (TM5)	2ch		
	8 位 (TMH)	2ch		
	用于时钟	1ch		
	看门狗 (WDT)	1ch		
串行接口	3 线式 SCI	—		
	自动发送 / 接收 3 线式 SCI	—		
	UART/3 线式 CSI 注 2	1ch		
	支持 LIN-bus 的 UART	1ch		
	I ² C 总线	1ch		
10 位 A/D 转换器	4ch			
中断	外部	6		
	内部	14		
复位	RESET 引脚	有		
	POC	1.59V±0.15V		
	LVI	能选择电源电压的检测电平		
	WDT	有		
片上调试功能	— 注 3			有
工作环境温度	T _A =-40 ~ +85°C			

注 1. R7F0C999B2DFP 是仅为使用片上调试功能的产品。

2. 因为引脚复用, 选择其中一个使用。

3. 通过 R7F0C999B2DFP 对应。

定时器的概要如下所示

		16 位定时器 / 事件计数器 00	8 位定时器 / 事件 计数器 50、51		8 位定时器 H0、H1		时钟定时器	看门狗定时器
		TM00	TM50	TM51	TMH0	TMH1		
功能	间隔定时器	1 个通道	1 个通道	1 个通道	1 个通道	1 个通道	1 个通道注 1	—
	外部事件计数器	1 个通道	1 个通道	1 个通道	—	—	—	—
	PPG 输出	1 个输出	—	—	—	—	—	—
	PWM 输出	—	1 个输出	1 个输出	1 个输出	1 个输出	—	—
	脉宽测量	2 个输入	—	—	—	—	—	—
	方波输出	1 个输出	1 个输出	1 个输出	1 个输出	1 个输出	—	—
	载波发生器	—	—	—	—	1 个输出注 2	—	—
	时钟定时器	—	—	—	—	—	1 个通道注 1	—
	看门狗定时器	—	—	—	—	—	—	1 个通道
中断源		2	1	1	1	1	1	—

- 注 1. 时钟定时器可同时使用定时器和间隔定时器的功能。
 2. 通过组合 TM51 和 TMH1，可作为载波发生器模式使用。

第 2 章 引脚功能

2.1 引脚功能一览

引脚的输入 / 输出缓冲器电源只有 V_{DD} 。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

电源	对应引脚
V_{DD}	全部引脚

(1) 端口功能

功能名称	输入 / 输出	功能	复位时	复用功能
P00	输入 / 输出	端口 0 2 位输入 / 输出端口 可以 1 位单位指定输入 / 输出 可通过软件设置, 使用内部上拉电阻。	输入端口	TI000
P01				TI010/TO00
P10	输入 / 输出	端口 1 8 位输入 / 输出端口 可以 1 位单位指定输入 / 输出 可通过软件设置, 使用内部上拉电阻。	输入端口	SCK10/TxD0
P11				SI10/RxD0
P12				SO10
P13				TxD6
P14				RxD6
P15				TOH0
P16				TOH1/INTP5
P17				TI50/TO50
P20 ~ P23	输入 / 输出	端口 2 4 位输入 / 输出端口 可以 1 位单位指定输入 / 输出	模拟输入	ANI0 ~ ANI3
P30	输入 / 输出	端口 3 4 位输入 / 输出端口 可以 1 位单位指定输入 / 输出 可通过软件设置, 使用内部上拉电阻。	输入端口	INTP1
P31				INTP2 注 1
P32				INTP3 注 2
P33				INTP4/TI51/TO51
P40、P41	输入 / 输出	端口 4 2 位输入 / 输出端口 可以 1 位单位指定输入 / 输出 可通过软件设置, 使用内部上拉电阻。	输入端口	—
P60	输入 / 输出	端口 6 2 位输入 / 输出端口 输出为 N 沟道漏极开路输出 (6V 耐压) 可以 1 位单位指定输入 / 输出	输入端口	SCL0
P61				SDA0
P70、P71	输入 / 输出	端口 7 2 位输入 / 输出端口 可以 1 位单位指定输入 / 输出 可通过软件设置, 使用内部上拉电阻。	输入端口	—
P120	输入 / 输出	端口 12 3 位输入 / 输出端口 可以 1 位单位指定输入 / 输出 仅 P120 可通过软件设置, 使用内部上拉电阻。	输入端口	INTP0/EXLVI
P121				X1 注 3
P122				X2/EXCLK 注 4

- 注 1. 在 R7F0C999B2DFP 时, P31 与 INTP2/OCD1A 复用。
 2. 在 R7F0C999B2DFP 时, P32 与 INTP3/OCD1B 复用。
 3. 在 R7F0C999B2DFP 时, P121 与 X1/OCD0A 复用。
 4. 在 R7F0C999B2DFP 时, P122 与 X2/EXCLK/OCD0B 复用。

(2) 非端口功能 (1/2)

功能名称	输入 / 输出	功能	复位时	复用功能
ANI0 ~ ANI3	输入	A/D 转换器的模拟输入	模拟输入	P20 ~ P23
EXLVI	输入	输入检测外部低电压的电位	输入端口	P120/INTP0
FLMD0	—	转移到闪存编程模式	—	—
INTP0	输入	输入可指定有效边沿（上升沿、下降沿、或者双边沿）的外部中断请求。	输入端口	P120/EXLVI
INTP1				P30
INTP2				P31
INTP3				P32
INTP4				P33/TI51/TO51
INTP5				P16/TOH1
REGC	—	连接内部运行稳压器输出（2.5 V）的稳定电容。通过电容器（0.47 ~ 1 μ F）连接到 V _{SS} 。	—	—
RESET	输入	系统复位输入	—	—
RxD0	输入	UART0 的串行数据输入	输入端口	P11/SI10
RxD6		UART6 的串行数据输入		P14
TxD0	输出	UART0 的串行数据输出	输入端口	P10/ $\overline{\text{SCK10}}$
TxD6		UART6 的串行数据输出		P13
$\overline{\text{SCK10}}$	输入 / 输出	CSI10 的时钟输入 / 输出	输入端口	P10/TxD0
SI10	输入	CSI10 的串行数据输入		P11/RxD0
SO10	输出	CSI10 的串行数据输出		P12
SCL0	输入 / 输出	I ² C 的时钟输入 / 输出	输入端口	P60
SDA0		I ² C 的串行数据输入 / 输出		P61
TI000	输入	16 位定时器 / 事件计数器 00 的外部计数时钟输入 16 位定时器 / 事件计数器 00 的捕捉寄存器（CR000、CR010）的捕捉触发输入	输入端口	P00
TI010	输入	16 位定时器 / 事件计数器 00 的捕捉寄存器（CR000）的捕捉触发输入	输入端口	P01/TO00
TI50	输入	8 位定时器 / 事件计数器 50 的外部计数时钟输入	输入端口	P17/TO50
TI51		8 位定时器 / 事件计数器 51 的外部计数时钟输入		P33/TO51/INTP4
TO00	输出	16 位定时器 / 事件计数器 00 输出	输入端口	P01/TO00
TO50	输出	8 位定时器 / 事件计数器 50 输出	输入端口	P17/TO50
TO51		8 位定时器 / 事件计数器 51 输出		P33/TO51/INTP4
TOH0	输出	8 位定时器 H0 输出	输入端口	P15
TOH1		8 位定时器 H1 输出		P16/INTP5
X1	—	连接用于主系统时钟的谐振器	输入端口	P121
X2	—			P122/EXCLK
EXCLK	输入	主系统时钟的外部时钟输入	输入端口	P122/X2
VDD	—	正电源、A/D 转换器的基准电压输入	—	—
VSS	—	接地电位	—	—

(2) 非端口功能 (2/2)

功能名称	输入 / 输出	功能	复位时	复用功能
OCD0A 注	输入	连接片上调试器	输入端口	P121/X1
OCD0B 注	—	连接片上调试器	输入端口	P122/X2/EXCLK
OCD1A 注	输入	连接片上调试器	输入端口	P31/INTP2
OCD1B 注	—	连接片上调试器	输入端口	P32/INTP3

注 只限 R7F0C999B2DFP。

2.2 引脚功能的说明

2.2.1 P00、P01 (Port0)

P00 和 P01 既可用作输入 / 输出端口，也可用作定时器输入 / 输出。
可以以 1 位单位指定以下运行模式。

(1) 端口模式

用作输入 / 输出端口。可通过设置端口模式寄存器 0 (PM0)，以 1 位单位指定为输入端口或者输出端口。可通过设置上拉电阻选择寄存器 0 (PU0) 使用内部上拉电阻。

(2) 控制模式

用作定时器输入 / 输出。

(a) TI000

这是 16 位定时器 / 事件计数器 00 的外部计数时钟输入引脚以及 16 位定时器 / 事件计数器 00 的捕捉寄存器 (CR000、CR010) 的捕捉触发信号输入引脚。

(b) TI010

这是 16 位定时器 / 事件计数器 00 的捕捉寄存器 (CR000) 的捕捉触发信号输入引脚。

(c) TO00

这是 16 位定时器 / 事件计数器 00 的定时器输出引脚。

2.2.2 P10 ~ P17 (Port1)

P10 ~ P17 既可用作输入 / 输出端口，也可用作串行接口的数据输入 / 输出、时钟输入 / 输出、定时器输入 / 输出以及外部中断请求输入。

可以以 1 位单位指定以下运行模式。

(1) 端口模式

用作输入 / 输出端口。可通过设置端口模式寄存器 1 (PM1)，以 1 位单位指定为输入端口或者输出端口。可通过设置上拉电阻选择寄存器 1 (PU1) 使用内部上拉电阻。

(2) 控制模式

用作串行接口的数据输入 / 输出、时钟输入 / 输出、定时器输入 / 输出以及外部中断请求输入。

(a) SI10

这是串行接口 CSI10 的串行数据输入引脚。

(b) SO10

这是串行接口 CSI10 的串行数据输出引脚。

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

这是串行接口 CSI10 的串行时钟输入 / 输出引脚。

(d) RxD0

这是串行接口 UART0 的串行数据输入引脚。

(e) RxD6

这是串行接口 UART6 的串行数据输入引脚。

(f) TxD0

这是串行接口 UART0 的串行数据输出引脚。

(g) TxD6

这是串行接口 UART6 的串行数据输出引脚。

(h) TI50

这是 8 位定时器 / 事件计数器 50 的外部计数时钟输入引脚。

(i) TO50

这是 8 位定时器 / 事件计数器 50 的定时器输出引脚。

(j) TOH0、TOH1

这是 8 位定时器 H0、H1 的定时器输出引脚。

(k) INTP5

这是可指定有效边沿（上升沿、下降沿或者双边沿）的外部中断请求的输入引脚。

2.2.3 P20 ~ P23 (Port2)

P20 ~ P23 既可用于输入 / 输出端口，也可用作 A/D 转换器的模拟输入。

可以以 1 位单位指定以下运行模式。

(1) 端口模式

用作输入 / 输出端口。可通过设置端口模式寄存器 2 (PM2)，以 1 位单位指定为输入端口或者输出端口。

(2) 控制模式

用作 A/D 转换器的模拟输入引脚 (ANI0 ~ ANI3)。在用作模拟输入引脚时，请参照“11.6 A/D 转换器的注意事项”的“(5) ANI0/P20 ~ ANI3/P23”。

注意 ANI0/P20 ~ ANI3/P23 在复位解除后变为模拟输入模式。

2.2.4 P30 ~ P33 (Port3)

P30 ~ P33 既可作为输入 / 输出端口，也可用作外部中断请求输入、定时器输入 / 输出以及片上调试器输入 / 输出。

可以以 1 位单位指定以下运行模式。

(1) 端口模式

用作输入 / 输出端口。可通过设置端口模式寄存器 3 (PM3)，以 1 位单位指定为输入端口或者输出端口。可通过设置上拉电阻选择寄存器 3 (PU3) 使用内部上拉电阻。

(2) 控制模式

用作外部中断请求输入、定时器输入 / 输出以及片上调试器输入 / 输出。

(a) INTP2 ~ INTP4

这是可指定有效边沿（上升沿、下降沿或者双边沿）的外部中断请求的输入引脚。

(b) TI51

这是 8 位定时器 / 事件计数器 51 的外部计数时钟输入引脚。

(c) TO51

这是 8 位定时器 / 事件计数器 51 的定时器输出引脚。

(d) INTP1

这是可指定有效边沿（上升沿、下降沿或者双边沿）的外部中断请求的输入引脚。

(e) OCD1A、OCD1B (仅限 R7F0C999B2DFP)

这是用于片上调试的连接引脚。

2.2.5 P40、P41 (Port4)

P40 和 P41 用作输入 / 输出端口。可通过设置端口模式寄存器 4 (PM4)，以 1 位单位指定为输入端口或者输出端口。可通过设置上拉电阻选择寄存器 4 (PU4) 使用内部上拉电阻。

2.2.6 P60、P61 (Port6)

P60 和 P61 既可作为输入 / 输出端口，也可用作串行接口的数据输入 / 输出以及时钟输入 / 输出。

可以以 1 位单位指定以下运行模式。

(1) 端口模式

用作输入 / 输出端口。可通过设置端口模式寄存器 6 (PM6)，以 1 位单位指定为输入端口或者输出端口。

P60 和 P61 的输出为 N 沟道漏极开路输出（6V 耐压）。

(2) 控制模式

用作串行接口的数据输入 / 输出和时钟输入输出。

(a) SDA0

这是串行接口 IIC0 的串行数据输入 / 输出引脚。

(b) SCL0

这是串行接口 IIC0 的串行时钟输入 / 输出引脚。

2.2.7 P70、P71 (Port7)

P70 和 P71 用作输入 / 输出端口。可通过设置端口模式寄存器 7 (PM7)，以 1 位单位指定为输入端口或者输出端口。可通过设置上拉电阻选择寄存器 7 (PU7) 使用内部上拉电阻。

2.2.8 P120 ~ P122 (Port12)

P120 ~ P122 既可用作输入 / 输出端口，也可用作外部中断请求输入、检测外部低电压的电位输入、主系统时钟的谐振器连接、主系统时钟的外部时钟输入以及片上调试器的连接。

可以以 1 位单位指定以下运行模式。

(1) 端口模式

用作输入 / 输出端口。可通过设置端口模式寄存器 12 (PM12)，以 1 位单位指定为输入端口或者输出端口。只有 P120 可通过设置上拉电阻选择寄存器 12 (PU12) 使用内部上拉电阻。

(2) 控制模式

用作外部中断请求输入、检测外部低电压的电位输入、主系统时钟的谐振器连接、主系统时钟的外部时钟输入以及片上调试器的连接。

(a) INTP0

这是可指定有效边沿（上升沿、下降沿或者双边沿）的外部中断请求的输入引脚。

(b) EXLVI

这是检测外部低电压的电位输入引脚。

(c) X1、X2

这是主系统时钟的谐振器连接引脚。

(d) EXCLK

这是主系统时钟的外部时钟输入引脚。

(e) OCD0A、OCD0B (仅限 R7F0C999B2DFP)

这是用于片上调试的连接引脚。

2.2.9 V_{DD} 、 V_{SS}

V_{DD} 、 V_{SS} 为电源 / 接地引脚。

(a) V_{DD}

V_{DD} 是提供正电源的引脚。

(b) V_{SS}

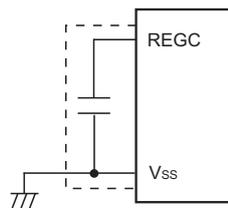
V_{SS} 是接地电位引脚。

2.2.10 $\overline{\text{RESET}}$

这是低电平有效的系统复位输入引脚。

2.2.11 REGC

这是内部运行稳压器输出（2.5 V）的稳定电容连接引脚。通过电容器（0.47 ~ 1 μ F）连接到 V_{SS} 。



注意 上图虚线部分的布线尽量使用最短布线。

2.2.12 FLMD0

这是用于转移到闪存编程模式的引脚。

在通常的运行模式中，FLMD0 必须连接到 V_{SS} 。

在闪存编程模式中，FLMD0 必须连接闪存编程器。

2.3 引脚的输入 / 输出电路和未使用引脚的处理

各引脚的输入 / 输出类型和未使用引脚的处理如表 2-2 所示。

另外，各类型的输入 / 输出电路结构请参照图 2-1。

表 2-2 各引脚的输入 / 输出电路类型

引脚名称	输入 / 输出电路类型	输入 / 输出	未使用时的推荐连接方法
P00/TI00	5-AQ	输入 / 输出	输入时：分别通过电阻连接到 EV_{DD} 或者 EV_{SS} 。 输出时：置为开路。
P01/TI010/TO00			
P10/SCK10/TxD0	5-AQ		
P11/SI10/RxD0			
P12/SO10	5-AG		
P13/TxD6			
P14/RxD6	5-AQ		
P15/TOH0	5-AG		
P16/TOH1/INTP5	5-AQ		
P17/TI50/TO50			
ANI0/P20 ~ ANI3/P23 注 1	11-G		< 设置为数字输入和设置为模拟输入时 > 分别通过电阻连接到 AV_{REF} 和 AV_{SS} 。 < 设置为数字输出时 > 置为开路。
P30/INTP1	5-AQ	输入 / 输出	输入时：分别通过电阻连接到 EV_{DD} 或者 EV_{SS} 。 输出时：置为开路。
P31/INTP2			
P32/INTP3			
P33/TI51/TO51/INTP4			
P40、P41	5-AG		
P60/SCL0	13-AI		输入时：分别通过电阻连接到 EV_{DD} 或者 EV_{SS} ，或者直接连接到 EV_{SS} 。 输出时：将端口的输出锁存器置“0”，并且通过输出低电平置为开路。
P61/SDA0			
P70、P71	5-AQ		输入时：分别通过电阻连接到 EV_{DD} 或者 EV_{SS} 。 输出时：置为开路。
P120/INTP0/EXLVI			
P121/X1 注 2	37		输入时：分别通过电阻连接到 V_{DD} 或者 V_{SS} 。 输出时：置为开路。
P122/X2/EXCLK 注 2			
FLMD0	38-A	—	连接到 EV_{SS} 或者 V_{SS} 。注 3
RESET	2	输入	直接或者通过电阻连接到 EV_{DD} 。
REGC	—	—	通过电容器 (0.47 ~ 1 μ F) 连接到 V_{SS} 。

注 1. ANI0/P20 ~ ANI3/P23 在复位解除后变为模拟输入模式。

2. 未使用时，在输入 / 输出端口模式（参照“图 5-2 时钟运行模式选择寄存器（OSCCTL）的格式”）中执行上述推荐的连接方法。

3. FLMD0 是在将数据写入闪存时使用的引脚。如果在板上进行闪存数据的改写时，必须通过电阻（推荐 10k Ω ）连接到 V_{SS} 。

图 2-1 引脚的输入 / 输出电路一览 (1/2)

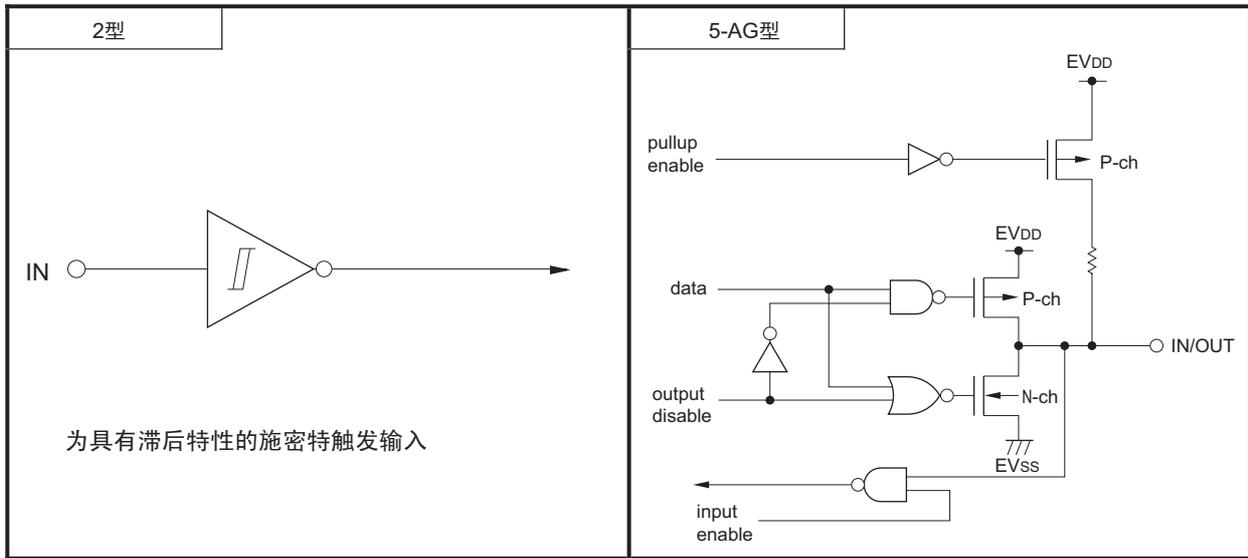
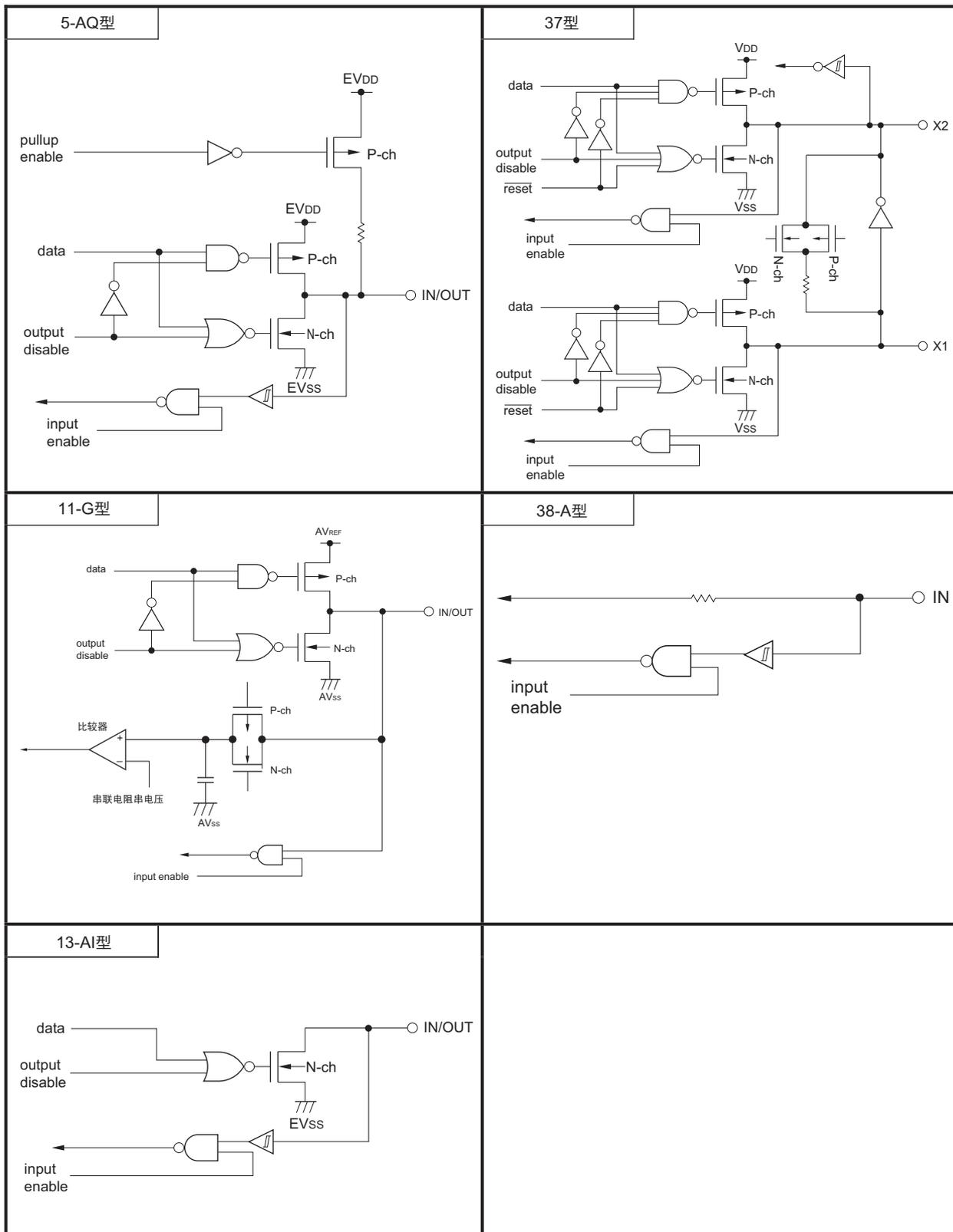


图 2-1 引脚的输入 / 输出电路一览 (2/2)



第3章 CPU 体系结构

3.1 存储空间

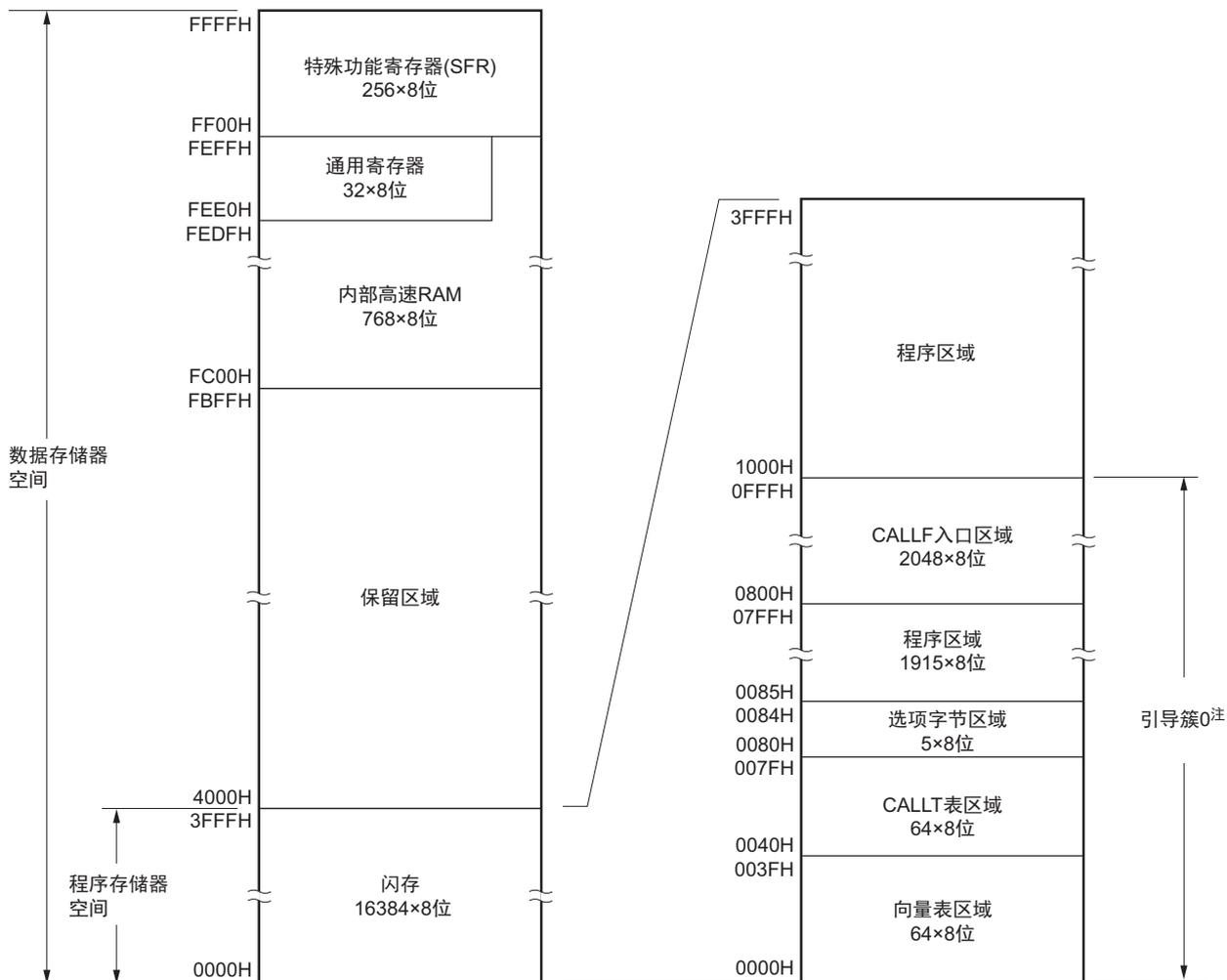
R7F0C011B、R7F0C012B、R7F01013B 以及 R7F0C999B 可存取 64K 字节的存储空间。存储器映射图如图 3-1 ~ 图 3-3 所示。

注意 因为复位解除后的存储容量切换寄存器（IMS）的初始值与内部存储容量无关，所以 R7F0C011B、R7F0C012B、R7F01013B 以及 R7F0C999B 的全部产品的 IMS 初始值固定为“CFH”。因此，复位解除后必须根据产品设置如下所示的值。

表 3-1 存储容量切换寄存器（IMS）的设置值

产品	IMS	ROM 容量
R7F0C011B	04H	16K 字节
R7F0C012B	C6H	24K 字节
R7F0C013B	C8H	32K 字节
R7F0C999B	C8H	32K 字节

图 3-1 存储器映射图 (R7F0C011B)



注 可通过安全设置，禁止改写引导簇 0（参照“22.7 安全设置”）。

备注 按块分割闪存（1 块 =1K 字节）。有关地址值和块号，请参照“表 3-2 闪存的地址值与块号的对应”。

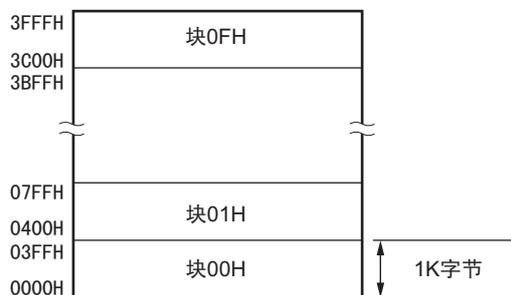
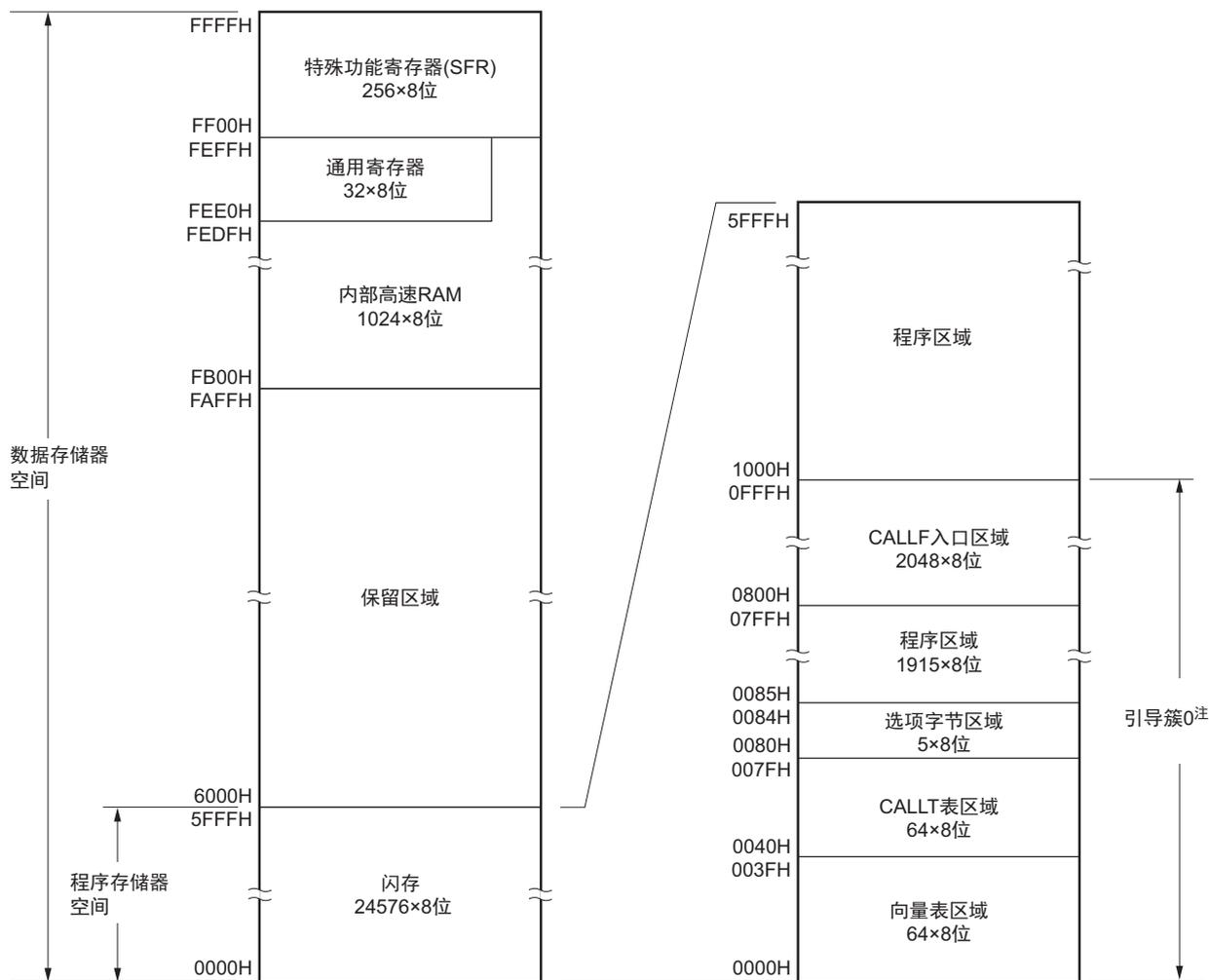


图 3-2 存储器映射图 (R7F0C012B)



注 可通过安全设置，禁止改写引导簇 0（参照“22.7 安全设置”）。

备注 按块分割闪存（1 块 =1K 字节）。有关地址值和块号，请参照“表 3-2 闪存的地址值与块号的对应”。

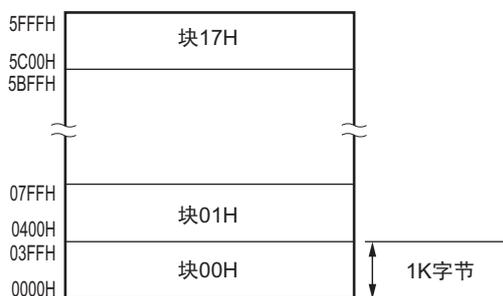
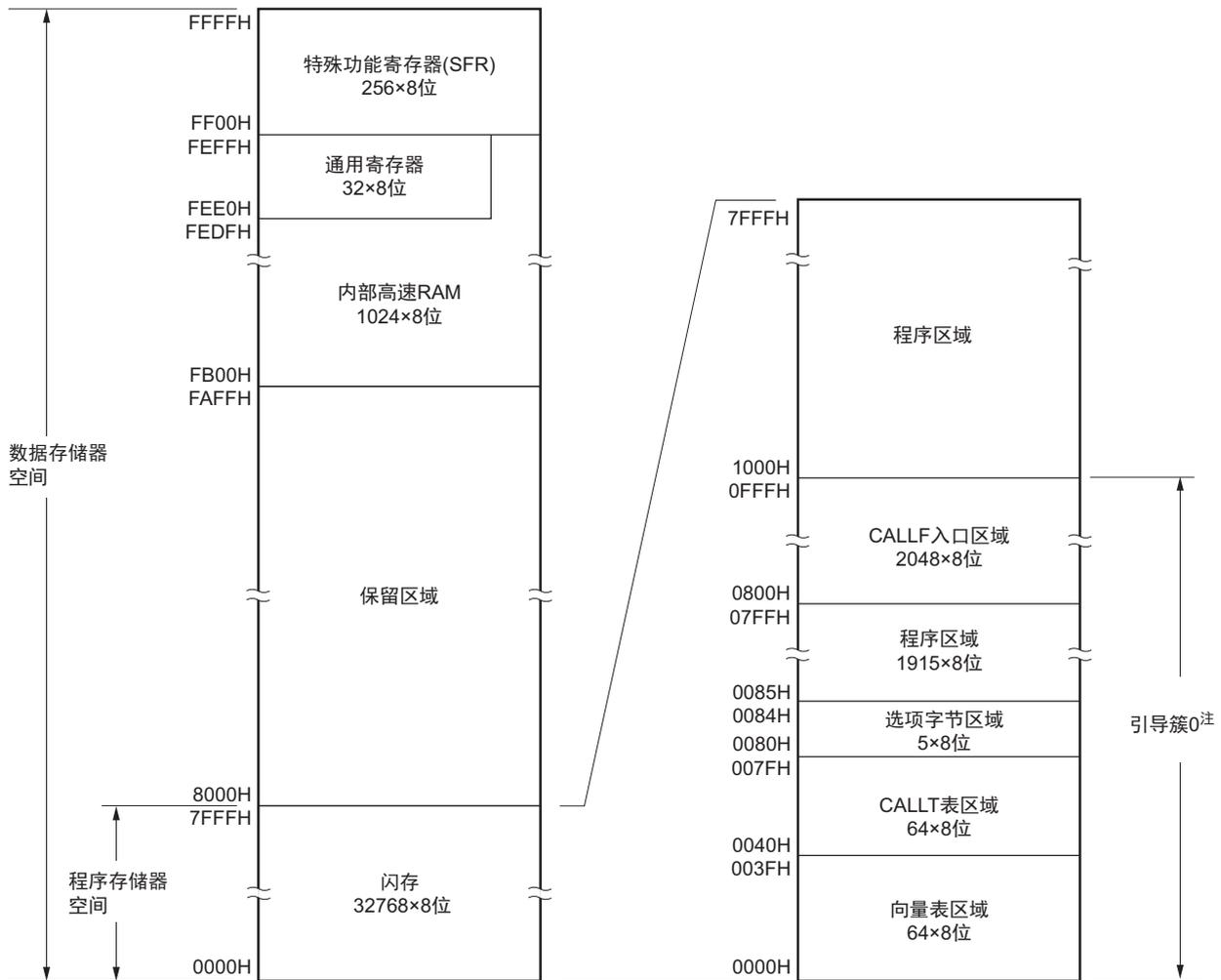
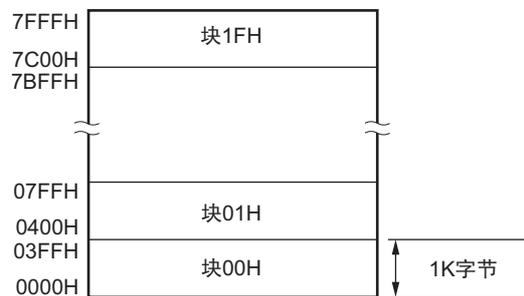


图 3-3 存储器映射图 (R7F0C013B、R7F0C999B)



注 可通过安全设置，禁止改写引导簇0（请参照“22.7 安全设置”）。

备注 按块分割闪存（1块=1K字节）。有关地址值和块号，请参照“表 3-2 闪存的地址值与块号的对应”。



闪存的地址值与块号的对应如下所示。

表 3-2 闪存的地址值与块号的对应

地址值	块号	地址值	块号
0000H ~ 03FFH	00H	4000H ~ 43FFH	10H
0400H ~ 07FFH	01H	4400H ~ 47FFH	11H
0800H ~ 0BFFH	02H	4800H ~ 4BFFH	12H
0C00H ~ 0FFFH	03H	4C00H ~ 4FFFH	13H
1000H ~ 13FFH	04H	5000H ~ 53FFH	14H
1400H ~ 17FFH	05H	5400H ~ 57FFH	15H
1800H ~ 1BFFH	06H	5800H ~ 5BFFH	16H
1C00H ~ 1FFFH	07H	5C00H ~ 5FFFH	17H
2000H ~ 23FFH	08H	6000H ~ 63FFH	18H
2400H ~ 27FFH	09H	6400H ~ 67FFH	19H
2800H ~ 2BFFH	0AH	6800H ~ 6BFFH	1AH
2C00H ~ 2FFFH	0BH	6C00H ~ 6FFFH	1BH
3000H ~ 33FFH	0CH	7000H ~ 73FFH	1CH
3400H ~ 37FFH	0DH	7400H ~ 77FFH	1DH
3800H ~ 3BFFH	0EH	7800H ~ 7BFFH	1EH
3C00H ~ 3FFFH	0FH	7C00H ~ 7FFFH	1FH

备注 R7F0C011B: 块号 00H ~ 0FH
 R7F0C012B: 块号 00H ~ 17H
 R7F0C013B: 块号 00H ~ 1FH
 R7F0C999B: 块号 00H ~ 1FH

3.1.1 内部程序存储空间

内部程序存储空间用于保存程序和表数据。一般由程序计数器（PC）进行寻址。

R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 按产品内置以下的内部 ROM（闪存）。

表 3-3 内部 ROM 容量

产品	内部 ROM（闪存）
R7F0C011B	16384×8 位（0000H ~ 3FFFH）
R7F0C012B	24576×8 位（0000H ~ 5FFFH）
R7F0C013B	32768×8 位（0000H ~ 7FFFH）
R7F0C999B	32768×8 位（0000H ~ 7FFFH）

内部程序存储空间分为以下区域。

(1) 向量表区域

0000H ~ 003FH 的 64 字节区域是保留的向量表区域。向量表区域保存因产生复位或者各中断请求而发生转移时的程序起始地址。

16 位地址中的低 8 位保存于偶数地址，高 8 位保存于奇数地址。

表 3-4 向量表

向量表地址	中断源
0000H	$\overline{\text{RESET}}$ 输入、POC、LVI、WDT
0004H	INTLVI
0006H	INTP0
0008H	INTP1
000AH	INTP2
000CH	INTP3
000EH	INTP4
0010H	INTP5
0012H	INTSRE6
0014H	INTSR6
0016H	INTST6
0018H	INTCSI10/INTST0
001AH	INTTMH1
001CH	INTTMH0
001EH	INTTM50
0020H	INTTM000
0022H	INTTM010
0024H	INTAD
0026H	INTSR0
0028H	INTWTI
002AH	INTTM51
002EH	INTWT
0034H	INTIIC0
003EH	BRK

(2) CALLT 指令表区域

0040H ~ 007FH 的 64 字节区域可保存 1 字节调用指令 (CALLT) 的子程序入口地址。

(3) 选项字节区域

在 0080H ~ 0084H 的 5 字节区域中预备选项字节区域。在 0080H ~ 0084H 设置选项字节。详细内容请参考“第 21 章 选项字节”。

(4) CALLF 指令入口区域

0800H ~ 0FFFH 的区域可通过 2 字节调用指令 (CALLF) 直接进行子程序调用。

3.1.2 内部数据存储空间

R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 内置以下 RAM。

(1) 内部高速 RAM

在内部高速 RAM 的 FEE0H ~ FEFFH 的 32 字节区域中分配 4 组通用寄存器。每组通用寄存器由 8 个 8 位寄存器组成。

内部高速 RAM 不能作为程序区域进行指令的写和执行。

另外，堆栈存储器使用内部高速 RAM。

表 3-5 内部高速 RAM 容量

产品	内部高速 RAM
R7F0C011B	768×8 位 (FC00H ~ FEFFH)
R7F0C012B	1024×8 位 (FB00H ~ FEFFH)
R7F0C013B	
R7F0C999B	

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

在 FF00H ~ FFFFH 区域中分配内部外围硬件的特殊功能寄存器 (SFR) (参照“3.2.3 特殊功能寄存器 (SFR: Special Function Register)”的“表 3-6 特殊功能寄存器”)。

注意 禁止访问未分配 SFR 的地址。

3.1.4 数据存储寻址

所谓寻址，就是指定下一条执行指令的地址或者执行指令时作为操作对象的寄存器或存储器等的地址的指定方法。

考虑到执行指令时作为操作对象的存储器寻址的可操作性等，在 R7F0C011B、R7F0C012B、R7F01013B 以及 R7F0C999B 中配置了多种寻址方式。尤其是在内部数据存储器区域，可根据特殊功能寄存器（SFR）或者通用寄存器等各功能进行特别寻址。数据存储器与寻址的对应如图 3-4 ~ 图 3-6 所示。有关各寻址的详细内容，请参照“3.4 操作数地址寻址”。

图 3-4 数据存储器与寻址的对应（R7F0C011B）

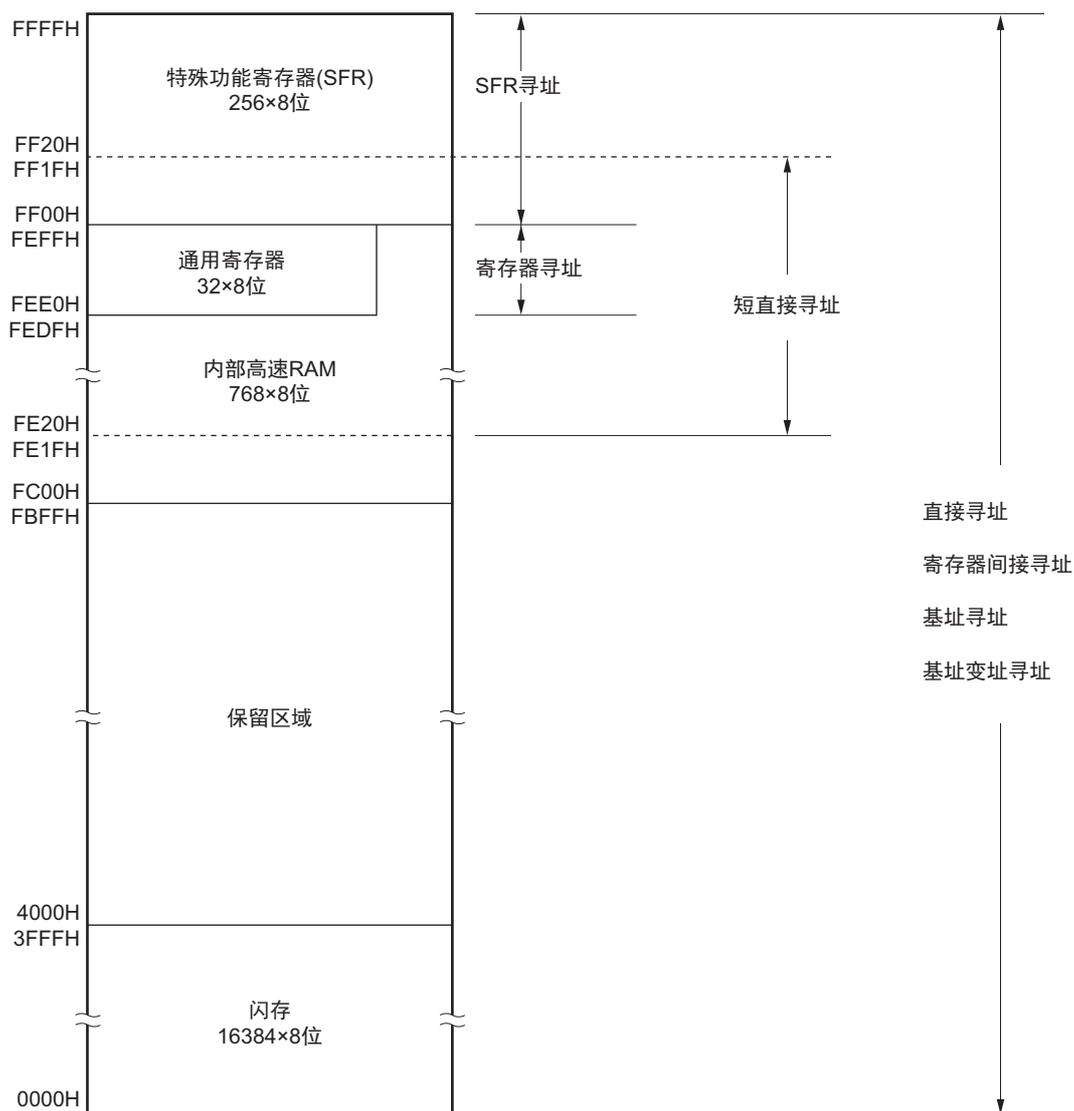


图 3-5 数据存储器和寻址的对应 (R7F0C012B)

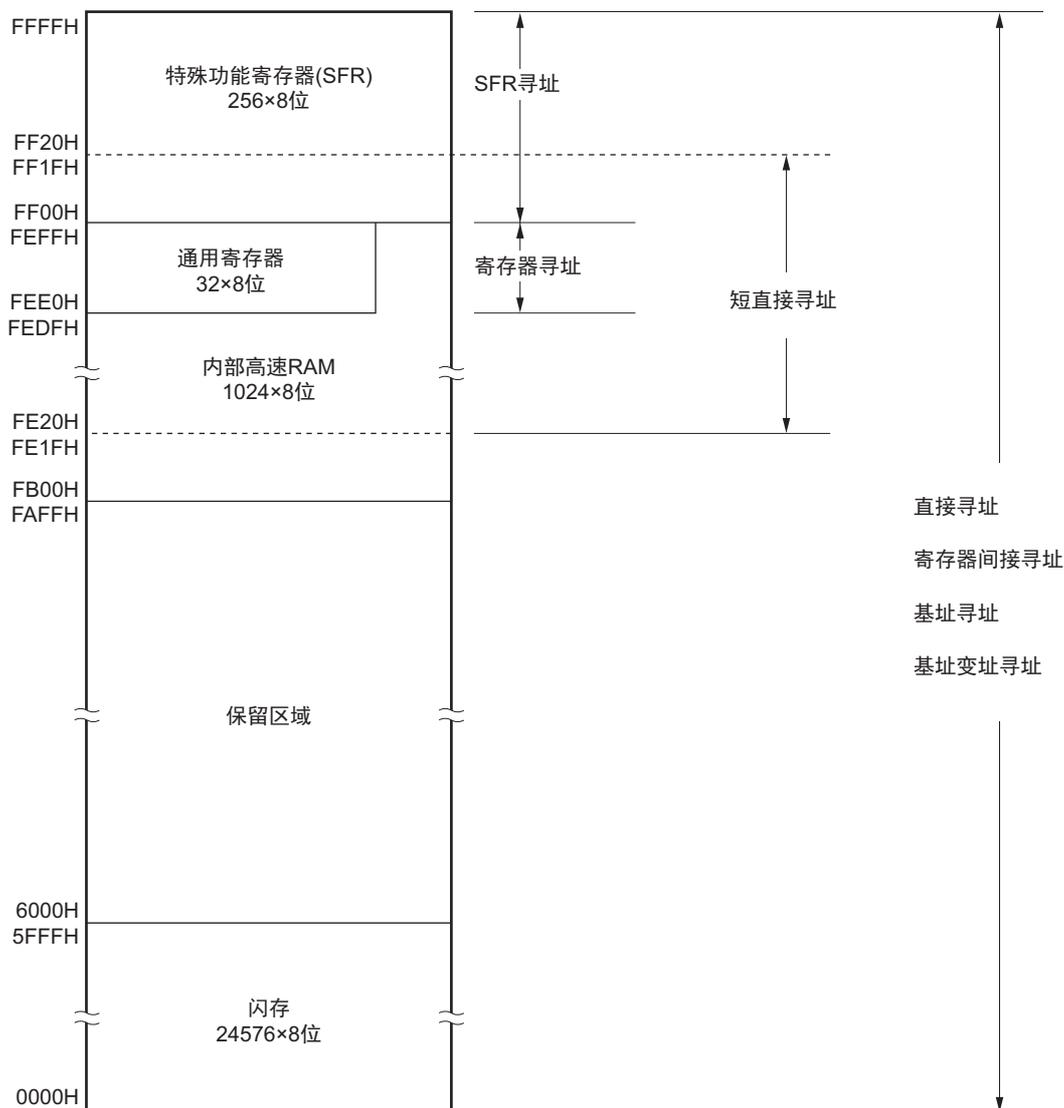
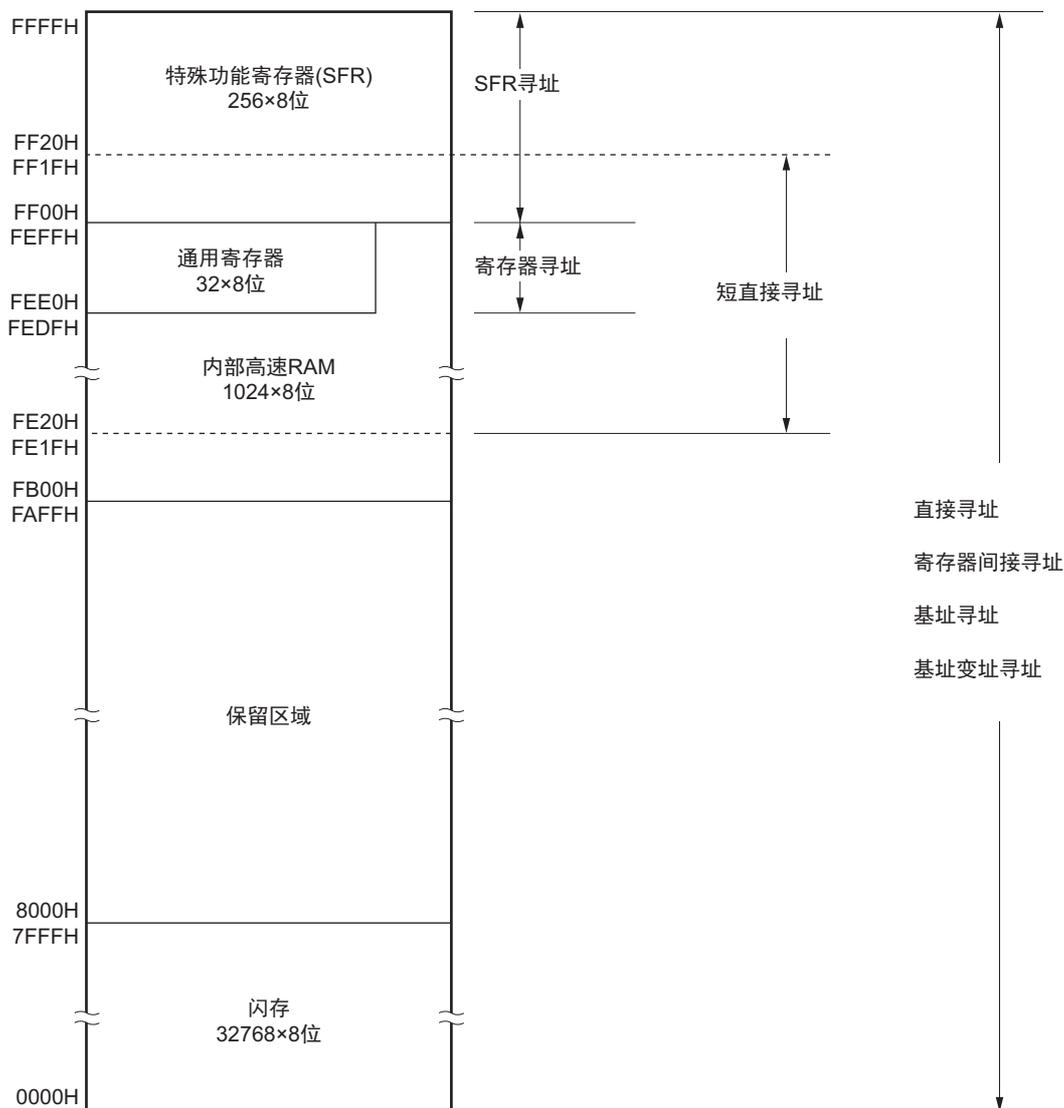


图 3-6 数据存储器和寻址的对应 (R7F0C013B、R7F0C999B)



3.2 处理器寄存器

R7F0C011B、R7F0C012B、R7F0C013B、R7F0C999B 内置以下处理器寄存器。

3.2.1 控制寄存器

控制寄存器具有控制程序顺序、状态和堆栈存储器等专用功能，并且具备程序计数器（PC）、程序状态字（PSW）和堆栈指针（SP）。

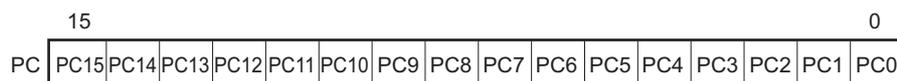
(1) 程序计数器（PC）

程序计数器为 16 位寄存器，用于保存下一个执行程序的地址信息。

在通常运行时，程序计数器（PC）根据取指令的字节数，自动进行递增计数。在执行转移指令时，立即数或者寄存器的内容被设置到程序计数器。

在产生复位信号后，地址 0000H 和 0001H 的复位向量表的值被设置到程序计数器。

图 3-7 程序计数器的结构



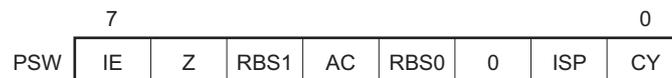
(2) 程序状态字（PSW）

程序状态字是 8 位寄存器，由通过执行指令而被置位或者复位的各种标志构成。

程序状态字的内容在接受向量中断请求以及执行 PUSH PSW 指令时入栈；在执行 RETB、RETI 指令和 POP PSW 指令时出栈。

在产生复位信号后，PSW 变为“02H”。

图 3-8 程序状态字的结构



(a) 中断允许标志（IE）

该标志控制 CPU 中断请求的接受运行。

当 IE 标志为“0”时，为禁止中断（DI）的状态，禁止所有可屏蔽中断。

当 IE 标志为“1”时，为允许中断（EI）的状态。此时，通过优先级控制标志（ISP）、各种中断源对应的中断屏蔽标志以及优先级指定标志来控制中断请求的接受。

IE 标志在执行 DI 指令或者接受中断请求时复位（“0”），在执行 EI 指令时置位（“1”）。

(b) 零标志（Z）

零标志在运算结果为“0”时置位（“1”），在“0”以外的情况下复位（“0”）。

(c) 寄存器组选择标志 (RBS0、RBS1)

该标志为 2 位标志，用于选择 4 个寄存器组中的 1 个寄存器组。
通过执行 SEL RBn 指令选择的寄存器组的 2 位信息保存在该标志中。

(d) 辅助进位标志 (AC)

该标志在 bit3 根据运算结果发生进位或者借位时置位 (“1”)，在上述以外的情况下复位 (“0”)。

(e) 优先级控制标志 (ISP)

该标志管理能接受的可屏蔽向量中断的优先级。当 ISP 标志为 “0” 时，禁止接受通过优先级指定标志寄存器 (PR0L、PR0H、PR1L、PR1H) (参照 “16.3(3) 优先级指定标志寄存器 (PR0L、PR0H、PR1L、PR1H)”) 指定为低优先级的向量中断请求。但是，实际是否接受中断请求则通过中断允许标志 (IE) 的状态来控制。

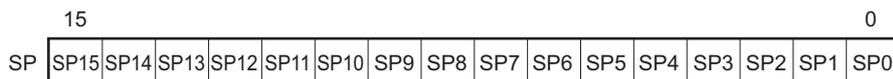
(f) 进位标志 (CY)

该标志存储执行加减运算指令时的上溢或者下溢以及执行循环指令时被移位的值，并且在执行位运算指令时用作位累加器。

(3) 堆栈指针 (SP)

堆栈指针为 16 位寄存器，用于保存存储器堆栈区域的起始地址。只有内部高速 RAM 区域可设置为堆栈区。

图 3-9 堆栈指针的结构



在进行堆栈存储器的写 (保存) 操作前，堆栈指针进行递减计数；在进行堆栈存储器的读 (恢复) 操作后，堆栈指针进行递增计数。

通过各堆栈运行而被保存 / 恢复的数据如图 3-10 和图 3-11 所示。

注意 SP 的内容因复位信号的产生而不定，因此，必须在使用堆栈前进行初始化。

图 3-10 被保存到堆栈存储器的数据

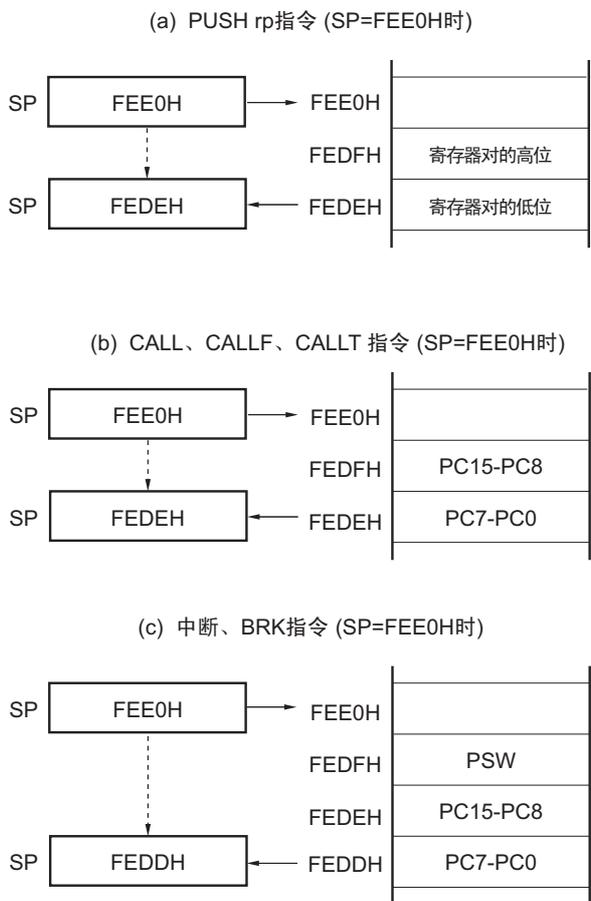
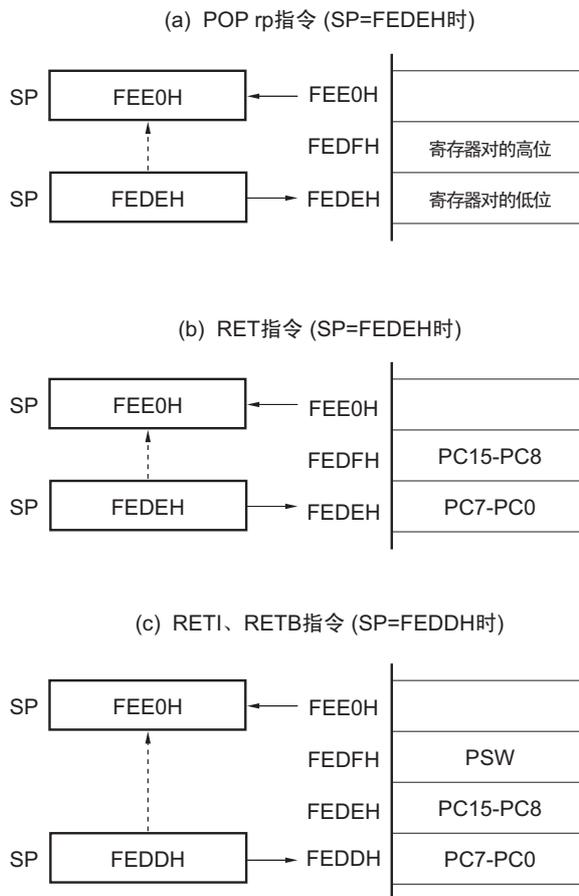


图 3-11 从堆栈存储器恢复的数据



3.2.2 通用寄存器

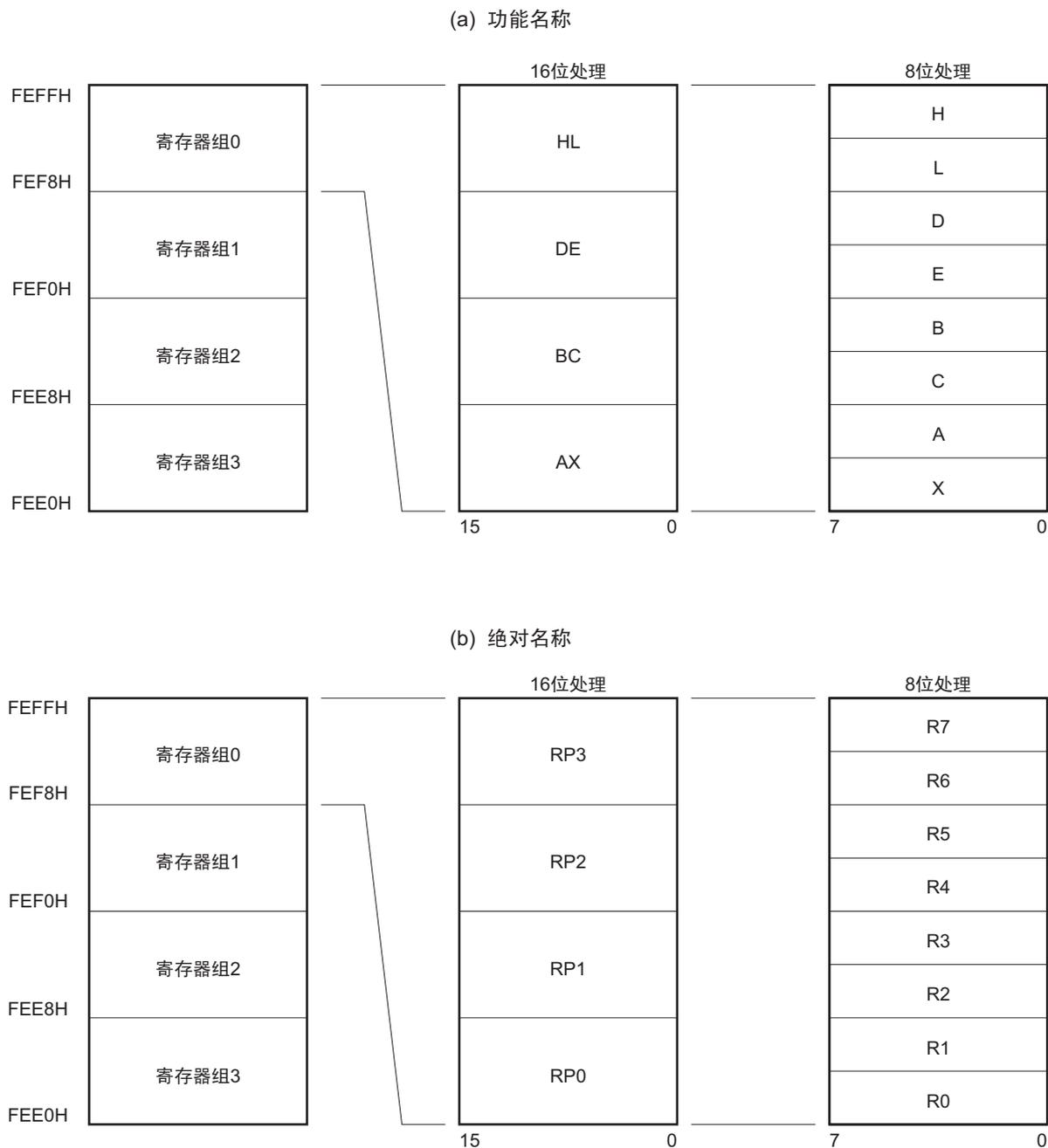
通用寄存器被映射到数据存储器的特定地址（FEE0H ~ FEFH），并且由以 8 个 8 位寄存器（X、A、C、B、E、D、L 和 H）为 1 个寄存器组的 4 个寄存器组构成。

各寄存器可分别用作 8 位寄存器，每 2 个寄存器还可作为一对用作 16 位寄存器（AX、BC、DE 和 HL）。

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

通过 CPU 控制指令（SEL RBn）设置执行指令时使用的寄存器组。因为通用寄存器是由 4 个寄存器组构成，所以可通过以组为单位切换通常处理时使用的寄存器和中断时使用的寄存器，创建高效程序。

图 3-12 通用寄存器的结构



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

特殊功能寄存不同于通用寄存器，分别具有各种特别功能。

特殊功能寄存器分配在 FF00H ~ FFFFH 的区域。

特殊功能寄存器与通用寄存器一样，可通过运算指令、传送指令以及位操作指令进行操作。可操作的位单位（1 位、8 位、16 位）因各特殊功能寄存器而不同。

各操作位单位的指定方法如下所示。

- 1 位操作
将通过汇编程序定义的符号作为 1 位操作指令的操作数（sfr.bit）描述。
也可指定地址。
- 8 位操作
将通过汇编程序定义的符号作为 8 位操作指令的操作数（sfr）描述。
也可指定地址。
- 16 位操作
将通过汇编程序定义的符号作为 16 位操作指令的操作数（sfrp）描述。
指定地址时，描述偶数地址。

特殊功能寄存器一览如表 3-6 所示。表中各项目的含义如下所示。

- 符号
符号表示特殊功能寄存器的地址。在 RA78K0 中定义为保留字；在 CC78K0 中通过 #pragma sfr 指令定义为 sfr 变量。在使用 RA78K0、ID78K0-QB 以及系统仿真器时，可作为指令的操作数描述。
- R/W
表示对应的特殊功能寄存器可否读（Read）/写（Write）。
R/W: 可读/写
R: 只读
W: 只写
- 可操作的位单位
“○”表示可操作的位单位（1 位、8 位或者 16 位）。“—”表示不可操作的位单位。
- 复位时
表示产生复位信号时的各寄存器的状态。

表 3-6 特殊功能寄存器 (1/3)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作的位单位			复位后
				1	8	16	
FF00H	端口寄存器 0	P0	R/W	○	○	—	00H
FF01H	端口寄存器 1	P1	R/W	○	○	—	00H
FF02H	端口寄存器 2	P2	R/W	○	○	—	00H
FF03H	端口寄存器 3	P3	R/W	○	○	—	00H
FF04H	端口寄存器 4	P4	R/W	○	○	—	00H
FF06H	端口寄存器 6	P6	R/W	○	○	—	00H
FF07H	端口寄存器 7	P7	R/W	○	○	—	00H
FF08H	10 位 A/D 转换结果寄存器	ADCR	R	—	—	○	0000H
FF09H	8 位 A/D 转换结果寄存器	ADCRH	R	—	○	—	00H
FF0AH	接收缓冲寄存器 6	RXB6	R	—	○	—	FFH
FF0BH	发送缓冲寄存器 6	TXB6	R/W	—	○	—	FFH
FF0CH	端口寄存器 12	P12	R/W	○	○	—	00H
FF0FH	串行 I/O 移位寄存器 10	SIO10	R	—	○	—	00H
FF10H	16 位定时器计数器 00	TM00	R	—	—	○	0000H
FF11H							
FF12H	16 位定时器捕捉 / 比较寄存器 000	CR000	R/W	—	—	○	0000H
FF13H							
FF14H	16 位定时器捕捉 / 比较寄存器 010	CR010	R/W	—	—	○	0000H
FF15H							
FF16H	8 位定时器计数器 50	TM50	R	—	○	—	00H
FF17H	8 位定时器比较寄存器 50	CR50	R/W	—	○	—	00H
FF18H	8 位定时器 H 比较寄存器 00	CMP00	R/W	—	○	—	00H
FF19H	8 位定时器 H 比较寄存器 10	CMP10	R/W	—	○	—	00H
FF1AH	8 位定时器 H 比较寄存器 01	CMP01	R/W	—	○	—	00H
FF1BH	8 位定时器 H 比较寄存器 11	CMP11	R/W	—	○	—	00H
FF1FH	8 位定时器计数器 51	TM51	R	—	○	—	00H
FF20H	端口模式寄存器 0	PM0	R/W	○	○	—	FFH
FF21H	端口模式寄存器 1	PM1	R/W	○	○	—	FFH
FF22H	端口模式寄存器 2	PM2	R/W	○	○	—	FFH
FF23H	端口模式寄存器 3	PM3	R/W	○	○	—	FFH
FF24H	端口模式寄存器 4	PM4	R/W	○	○	—	FFH
FF26H	端口模式寄存器 6	PM6	R/W	○	○	—	FFH
FF27H	端口模式寄存器 7	PM7	R/W	○	○	—	FFH
FF28H	A/D 转换器模式寄存器	ADM	R/W	○	○	—	00H
FF29H	模拟输入通道指定寄存器	ADS	R/W	○	○	—	00H
FF2CH	端口模式寄存器 12	PM12	R/W	○	○	—	FFH
FF2FH	A/D 端口配置寄存器	ADPC	R/W	○	○	—	00H
FF30H	上拉电阻选择寄存器 0	PU0	R/W	○	○	—	00H

表 3-6 特殊功能寄存器 (2/3)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作的位单位			复位后
				1	8	16	
FF31H	上拉电阻选择寄存器 1	PU1	R/W	○	○	—	00H
FF33H	上拉电阻选择寄存器 3	PU3	R/W	○	○	—	00H
FF34H	上拉电阻选择寄存器 4	PU4	R/W	○	○	—	00H
FF37H	上拉电阻选择寄存器 7	PU7	R/W	○	○	—	00H
FF3CH	上拉电阻选择寄存器 12	PU12	R/W	○	○	—	00H
FF41H	8 位定时器比较寄存器 51	CR51	R/W	—	○	—	00H
FF43H	8 位定时器模式控制寄存器 51	TMC51	R/W	○	○	—	00H
FF48H	外部中断上升沿允许寄存器	EGP	R/W	○	○	—	00H
FF49H	外部中断下降沿允许寄存器	EGN	R/W	○	○	—	00H
FF4FH	输入切换控制寄存器	ISC	R/W	○	○	—	00H
FF50H	异步串行接口运行模式寄存器 6	ASIM6	R/W	○	○	—	01H
FF53H	异步串行接口接收错误状态寄存器 6	ASIS6	R	—	○	—	00H
FF55H	异步串行接口发送状态寄存器 6	ASIF6	R	—	○	—	00H
FF56H	时钟选择寄存器 6	CKSR6	R/W	—	○	—	00H
FF57H	波特率发生器控制寄存器 6	BRGC6	R/W	—	○	—	FFH
FF58H	异步串行接口控制寄存器 6	ASICL6	R/W	○	○	—	16H
FF69H	8 位定时器 H 模式寄存器 0	TMHMD0	R/W	○	○	—	00H
FF6AH	定时器时钟选择寄存器 50	TCL50	R/W	○	○	—	00H
FF6BH	8 位定时器模式控制寄存器 50	TMC50	R/W	○	○	—	00H
FF6CH	8 位定时器 H 模式寄存器 1	TMHMD0	R/W	○	○	—	00H
FF6DH	8 位定时器 H 载波控制寄存器 1	TMCYC1	R/W	○	○	—	00H
FF6FH	时钟定时器运行模式寄存器	WTM	R/W	○	○	—	00H
FF70H	异步串行接口运行模式寄存器 0	ASIM0	R/W	○	○	—	01H
FF71H	波特率发生器控制寄存器 0	BRGC0	R/W	—	○	—	1FH
FF72H	接收缓冲寄存器 0	RXB0	R	—	○	—	FFH
FF73H	异步串行接口接收错误状态寄存器 0	ASIS0	R	—	○	—	00H
FF74H	发送移位寄存器 0	TXS0	W	—	○	—	FFH
FF80H	串行运行模式寄存器 10	CSIM10	R/W	○	○	—	00H
FF81H	串行时钟选择寄存器 10	CSIC10	R/W	○	○	—	00H
FF84H	发送缓冲寄存器 10	SOTB10	R/W	—	○	—	00H
FF8CH	定时器时钟选择寄存器 51	TCL51	R/W	○	○	—	00H
FF99H	看门狗定时器运行寄存器	WDTE	R/W	—	○	—	1AH/9AH 注 1
FF9FH	时钟运行模式选择寄存器	OSCCTL	R/W	○	○	—	00H
FFA0H	内部振荡模式寄存器	RCM	R/W	○	○	—	80H 注 2
FFA1H	主时钟模式寄存器	MCM	R/W	○	○	—	00H
FFA2H	主 OSC 控制寄存器	MOC	R/W	○	○	—	80H

注 1. WDTE 的复位值取决于选项字节的设置。

2. 复位解除后为“00H”，但是在等待高速内部振荡器的振荡精度稳定后，自动切换为“80H”。

表 3-6 特殊功能寄存器 (3/3)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作的位单位			复位后	
				1	8	16		
FFA3H	振荡稳定时间计数器状态寄存器	OSTC	R	○	○	—	00H	
FFA4H	振荡稳定时间选择寄存器	OSTS	R/W	—	○	—	05H	
FFA5H	IIC 移位寄存器 0	IIC0	R/W	—	○	—	00H	
FFA6H	IIC 控制寄存器 0	IICC0	R/W	○	○	—	00H	
FFA7H	从属地址寄存器 0	SVA0	R/W	—	○	—	00H	
FFA8H	IIC 时钟选择寄存器 0	IICCL0	R/W	○	○	—	00H	
FFA9H	IIC 功能扩展寄存器 0	IICX0	R/W	○	○	—	00H	
FFAAH	IIC 状态寄存器 0	IICS0	R	○	○	—	00H	
FFABH	IIC 标志寄存器 0	IICF0	R/W	○	○	—	00H	
FFACH	复位控制标志寄存器	RESF	R	—	○	—	00H 注 1	
FFBAH	16 位定时器模式控制寄存器 00	TMC00	R/W	○	○	—	00H	
FFBBH	预分频器模式寄存器 00	PRM00	R/W	○	○	—	00H	
FFBCH	捕捉 / 比较控制寄存器 00	CRC00	R/W	○	○	—	00H	
FFBDH	16 位定时器输出控制寄存器 00	TOC00	R/W	○	○	—	00H	
FFBEH	低电压检测寄存器	LVIM	R/W	○	○	—	00H 注 2	
FFBFH	低电压检测电平选择寄存器	LVIS	R/W	○	○	—	00H 注 2	
FFE0H	中断请求标志寄存器 0L	IF0	IF0L	R/W	○	○	○	00H
FFE1H	中断请求标志寄存器 0H		IF0H	R/W	○	○		00H
FFE2H	中断请求标志寄存器 1L	IF1	IF1L	R/W	○	○	○	00H
FFE3H	中断请求标志寄存器 1H		IF1H	R/W	○	○		00H
FFE4H	中断屏蔽标志寄存器 0L	MK0	MK0L	R/W	○	○	○	FFH
FFE5H	中断屏蔽标志寄存器 0H		MK0H	R/W	○	○		FFH
FFE6H	中断屏蔽标志寄存器 1L	MK1	MK1L	R/W	○	○	○	FFH
FFE7H	中断屏蔽标志寄存器 1H		MK1H	R/W	○	○		FFH
FFE8H	优先级指定标志寄存器 0L	PR0	PR0L	R/W	○	○	○	FFH
FFE9H	优先级指定标志寄存器 0H		PR1L	R/W	○	○		FFH
FFEAH	优先级指定标志寄存器 1L	PR1	PR1L	R/W	○	○	○	FFH
FFEBH	优先级指定标志寄存器 1H		PR1H	R/W	○	○		FFH
FFF0H	存储器尺寸切换寄存器注 3	IMS	R/W	—	○	—	CFH	
FFF4H	内部扩展 RAM 尺寸切换寄存器注 3	IXS	R/W	—	○	—	0CH	
FFFBH	处理器时钟控制寄存器	PCC	R/W	○	○	—	01H	

注 1. RESF 的复位值因复位源而变。

2. LVIM、LVIS 的复位值因复位源而变。

3. 在 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 的全部产品中，IMS 和 IXS 的复位解除后的初始值与内部存储容量无关，为固定值（IMS=CFH、IXS=0CH）。因此在复位解除后，必须给各产品设置表 3-1 所示的值。

3.3 指令地址寻址

指令地址取决于程序计数器（PC）的内容。一般情况下，每执行1条指令，程序计数器（PC）的内容就根据取指令的字节数自动进行递增（每1字节+1）计数。但是，在执行转移指令时，通过以下所示的寻址方式，将转移目标地址信息设置到程序计数器（PC）后进行转移（有关各指令的详细内容，请参照《78K/0系列指令用户手册》（U12326C））。

3.3.1 相对寻址

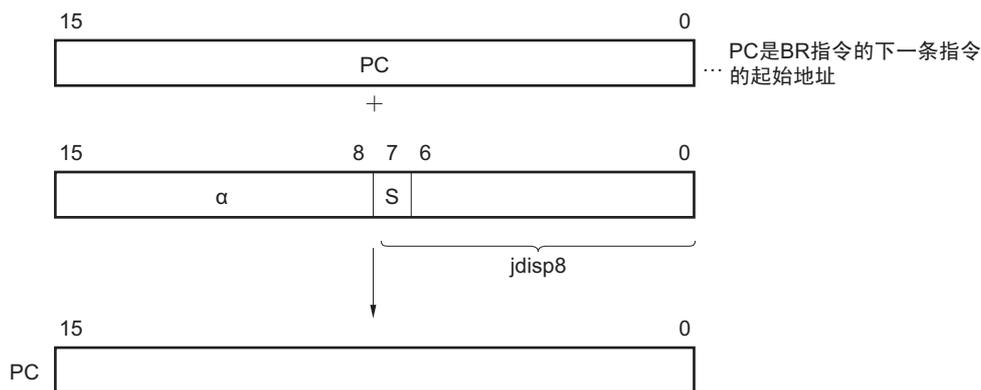
[功能]

将指令码的8位立即数（偏移量：jdisp8）与下一条指令的起始地址相加后的值传送到程序计数器（PC）后进行转移。偏移量作为带符号的2的补码数据（-128～+127）处理，bit7为符号位。

即，相对寻址是指从下一条指令的起始地址相对转移到-128至+127范围内。

在执行BR \$addr16指令和带条件的转移指令时，进行相对寻址。

[图示]



S=0时， α 的全部位为“0”。

S=1时， α 的全部位为“1”。

3.3.2 立即寻址

[功能]

将指令字中的立即数传送到程序计数器（PC）后进行转移。

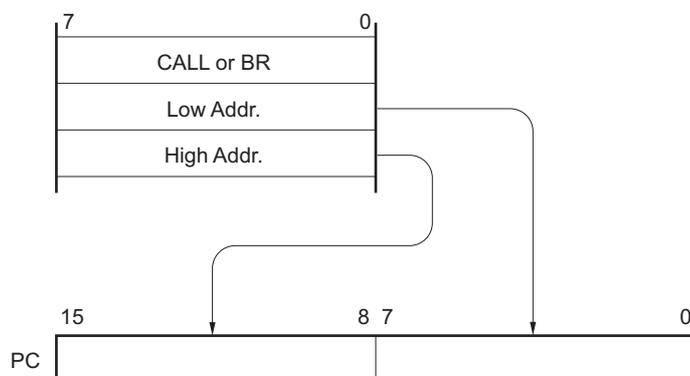
在执行 CALL !addr16、BR !addr16 或者 CALLF !addr11 指令时，进行立即寻址。

通过 CALL !addr16 指令和 BR !addr16 指令可转移到全部的程序存储空间。但是，在转移到没有通过存储器组选择寄存（BANK）设置的存储器组时，必须先通过 BANK 切换到存储器组的设置后再进行转移。

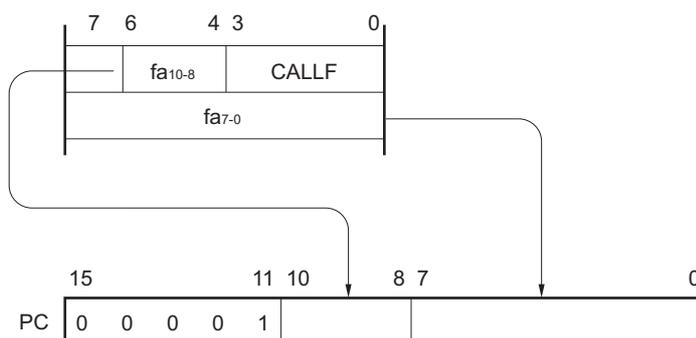
通过 CALLF !addr11 指令可转移到 0800H ~ 0FFFH 的区域。

[图示]

执行CALL !addr16和BR !addr16指令的情况



执行CALLF !addr11指令的情况



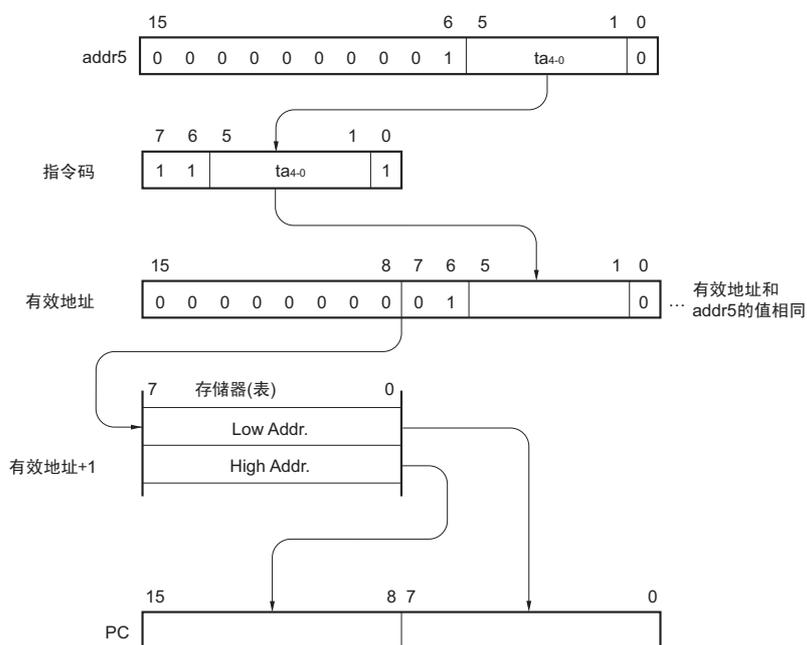
3.3.3 表间接寻址

[功能]

将通过指令码的 bit1 ~ bit5 的立即数进行寻址的特定区域的表内容（转移目标地址）传送到程序计数器（PC）后进行转移。

在执行 CALLT [addr5] 指令时，进行表间接寻址。可通过该指令访问保存在 0040H ~ 007FH 的存储器表中的地址，并且转移到全部的程序存储空间。但是，在转移到没有通过存储器组选择寄存（BANK）设置的存储器组时，必须先通过 BANK 切换到存储器组的设置后再进行转移。

[图示]



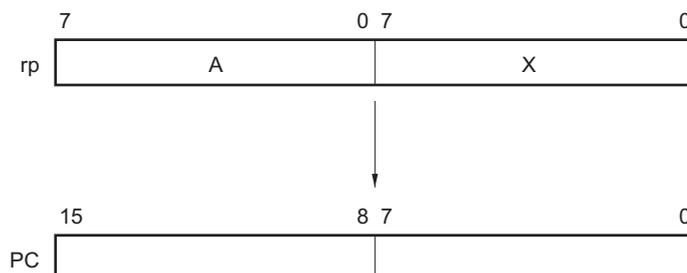
3.3.4 寄存器寻址

[功能]

将通过指令字指定的寄存器对（AX）的内容传送到程序计数器（PC）后进行转移。

在执行 BR AX 指令时，进行寄存器寻址。

[图示]



3.4 操作数地址寻址

在执行指令时作为操作对象的寄存器或者存储器等的地址指定方法（寻址）有以下几种。

3.4.1 隐含寻址

[功能]

隐含寻址是指用作通用寄存器区域中的累加器（A 和 AX）的寄存器自动（隐含）进行寻址。

在 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 的指令字中，使用隐含寻址的指令如下所示。

指令	隐含寻址指定的寄存器
MULU	A 寄存器作为被乘数的寄存器，AX 寄存器作为保存积的寄存器。
DIVUW	AX 寄存器作为被除数和保存商的寄存器。
ADJBA/ADJBS	A 寄存器作为保存 10 进制校正的对象数值的寄存器。
ROR4/ROL4	A 寄存器作为保存数字循环的对象数字数据的寄存器。

[操作数格式]

因为根据指令自动决定，所以没有特定的操作数格式。

[描述示例]

MULU X 的情况

在执行 8 位×8 位的乘法运算指令时，A 寄存器和 X 寄存器的积保存在 AX 寄存器。此时，A 寄存器和 AX 寄存器通过隐含寻址指定。

3.4.2 寄存器寻址

[功能]

寄存器寻址是指对用作操作数的通用寄存器进行的寻址。通过寄存器组选择标志（RBS0、RBS1）和指令码中的寄存器指定码指定被寻址的通用寄存器。

在执行具有以下操作数格式的指令时，进行寄存器寻址。指定 8 位寄存器时，通过指令码中的 3 位指定 8 个寄存器中的 1 个。

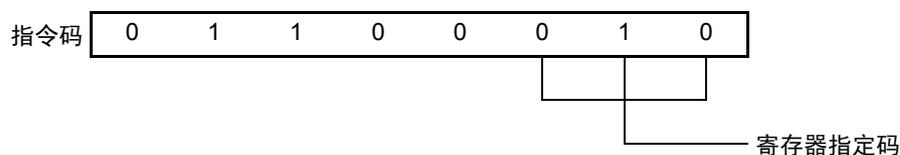
[操作数格式]

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

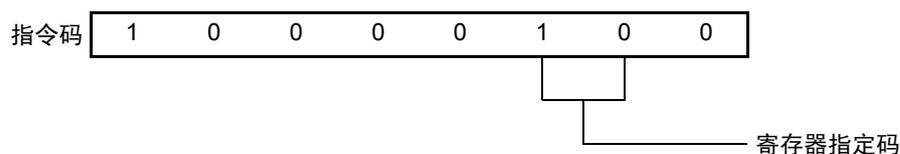
“r”和“rp”可用功能名称（X、A、C、B、E、D、L、H、AX、BC、DE 和 HL）和绝对名称（R0～R7 和 RP0～RP3）进行描述。

[描述示例]

MOV A, C ; 当选择 C 寄存器作为 r 时



INCW DE ; 当选择 DE 寄存器对作为 rp 时



3.4.3 直接寻址

[功能]

直接寻址是指对指令字中的立即数所示的存储器进行直接寻址。

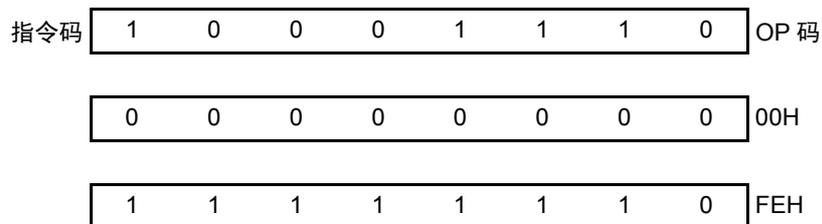
可对全部的存储空间进行寻址。但是，在对没有通过存储器组选择寄存（BANK）设置的存储器组进行寻址时，必须先通过 BANK 切换到存储器组的设置后再进行寻址。

[操作数格式]

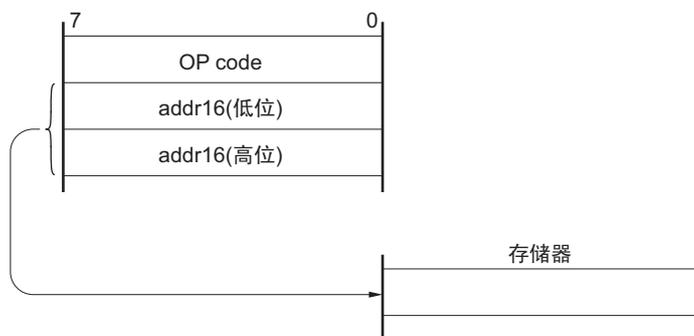
标识符	描述方法
addr16	标号或 16 位立即数

[描述示例]

MOV A, !0FE00H ; !addr16 为 FE00H 的情况



[图示]



3.4.4 短直接寻址

[功能]

短直接寻址是指通过指令字中的 8 位数据对固定区域的操作对象存储器进行直接寻址。

适用于该寻址的固定区域为 FE20H ~ FF1FH 的 256 字节区域，并且内部 RAM 和特殊功能寄存器 (SFR) 分别映射在 FE20H ~ FEFFH 和 FF00H ~ FF1FH 的区域。

SFR 区域中的部分区域 (FF00H ~ FF1FH 区域) 适用于短直接寻址。程序中经常被存取的端口、定时器 / 事件计数器的比较寄存器和捕捉寄存器都映射在该区域，因此，可用短字节数和短时钟数操作这些特殊功能寄存器 (SFR)。

在 8 位立即数为 20H ~ FFH 时，有效地址的 bit8 为“0”，在 8 位立即数为 00H ~ 1FH 时，有效地址的 bit8 为“1”。请参照 “[图示]”。

[操作数格式]

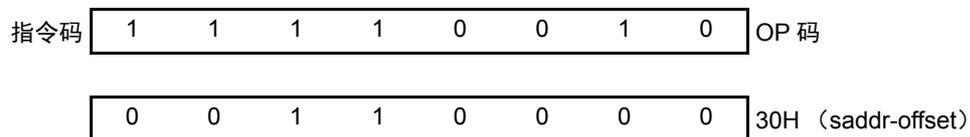
标识符	描述方法
saddr	标号或者表示 FE20H ~ FF1FH 的立即数
saddrp	标号或者表示 FE20H ~ FF1FH 的立即数 (只限偶数地址)

[描述示例]

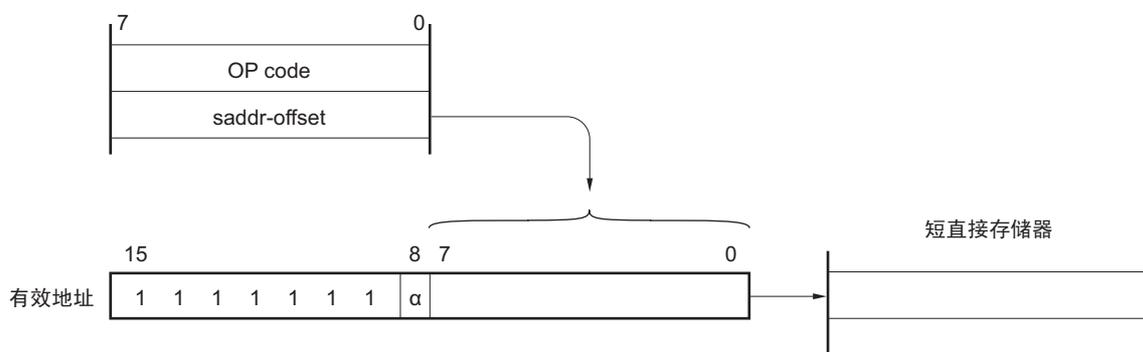
LB1 EQU 0FE30H ; 由 LB1 定义 FE30H。

:

MOV LB1, A ; LB1 表示 saddr 区域的 FE30H，并且将 A 寄存器的值传送给该地址时



[图示]



8 位立即数为 20H ~ FFH 时，α=0。

8 位立即数为 00H ~ 1FH 时，α=1。

3.4.5 特殊功能寄存器（SFR）寻址

[功能]

特殊功能寄存器寻址是指通过指令字中的 8 位立即数对存储器映射的特殊功能寄存器（SFR）进行寻址。

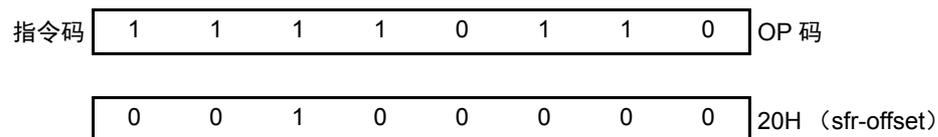
该寻址适用于 FF00H ~ FFCFH 和 FFE0H ~ FFFFH 的 240 字节区域。但是，映射在 FF00H ~ FF1FH 的特殊功能寄存器（SFR）的寻址，也能通过短直接寻址进行。

[操作数格式]

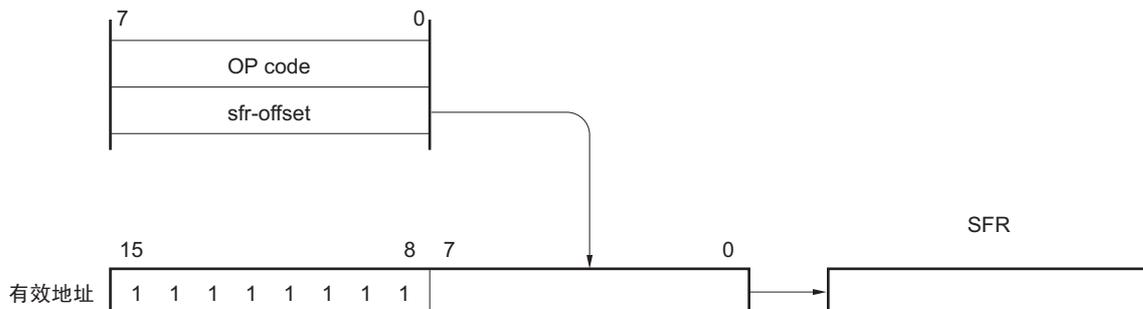
标识符	描述方法
sfr	特殊功能寄存器名称
sfrp	可 16 位操作的特殊功能寄存器名称（只限偶数地址）

[描述示例]

MOV PM0, A ; 当选择 PM0（FF20H）作为 sfr 时



[图示]



3.4.6 寄存器间接寻址

[功能]

寄存器间接寻址是指通过作为操作数指定的寄存器对的内容对存储器进行寻址。通过寄存器组选择标志（RBS0 和 RBS1）和指令字中的寄存器对指定码指定被存取的寄存器对。

可对全部的存储空间进行寻址。但是，在对没有通过存储器组选择寄存（BANK）设置的存储器组进行寻址时，必须先通过 BANK 切换到存储器组的设置后再进行寻址。

[操作数格式]

标识符	描述方法
—	[DE], [HL]

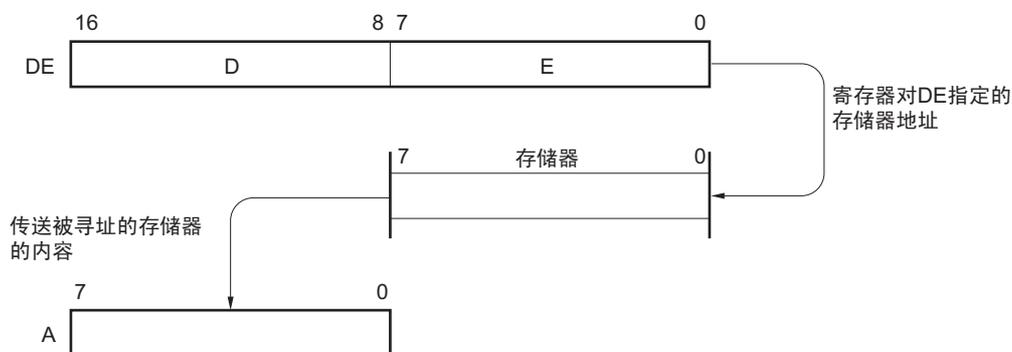
[描述示例]

MOV A, [DE]; 当选择 [DE] 作为寄存器对时

指令码

1	0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---	---

[图示]



3.4.7 基址寻址

[功能]

基址寻址是指通过将 8 位立即数与作为基址寄存器的 HL 寄存器对的内容相加后的结果对存储器进行寻址。通过寄存器组选择标志（RBS0 和 RBS1）指定寄存器组中被访问的 HL 寄存器对。通过将作为正数的偏移数据扩展到 16 位，执行加法运算。忽略从第 16 位的进位。

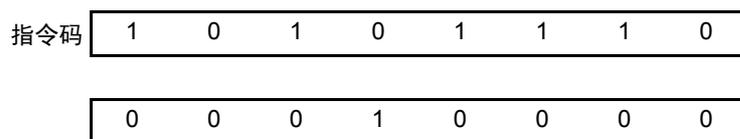
可对全部的存储空间进行寻址。但是，在对没有通过存储器组选择寄存（BANK）设置的存储器组进行寻址时，必须先通过 BANK 切换到存储器组的设置后再进行寻址。

[操作数格式]

标识符	描述方法
—	[HL+byte]

[描述示例]

MOV A, [HL + 10H] ; byte 为“10H”时



[图示]



3.4.8 基址变址寻址

[功能]

基址变址寻址是指通过将指令字指定的 B 寄存器或者 C 寄存器的内容与作为基址寄存器的 HL 寄存器对的内容相加后的结果对存储器进行寻址。通过寄存器组选择标志（RBS0 至 RBS1）指定寄存器组中的被存取 HL、B、C 寄存器。通过将作为正数的 B 寄存器或者 C 寄存器的内容扩展到 16 位，执行加法运算。忽略从第 16 位的进位。

可对全部的存储空间进行寻址。但是，在对没有通过存储器组选择寄存（BANK）设置的存储器组进行寻址时，必须先通过 BANK 切换到存储器组的设置后再进行寻址。

[操作数格式]

标识符	描述方法
—	[HL+B], [HL+C]

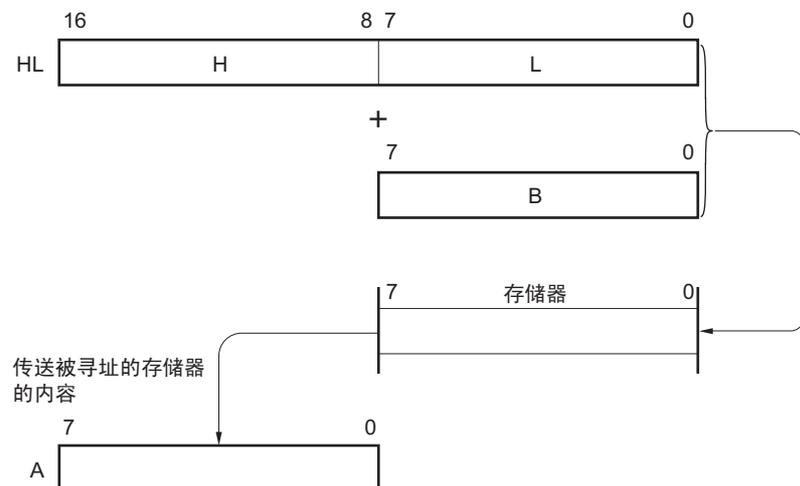
[描述示例]

MOV A, [HL+B]; 选择 B 寄存器时

指令码

1	0	1	0	1	0	1	1
---	---	---	---	---	---	---	---

[图示]



3.4.9 堆栈寻址

[功能]

堆栈寻址是指通过堆栈指针（SP）的内容对堆栈区域进行间接寻址。

在执行 PUSH、POP、子程序调用和返回指令或者因产生中断请求而保存 / 恢复寄存器时，自动进行堆栈寻址。

堆栈寻址只能存取内部高速 RAM 区域。

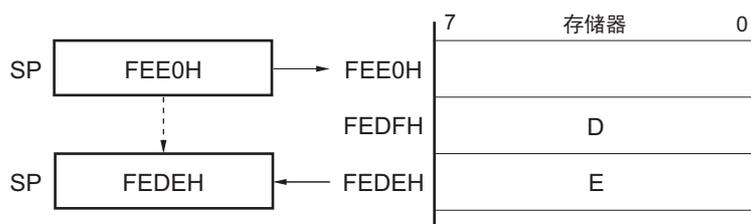
[描述示例]

PUSH DE ; 保存 DE 寄存器时

指令码

1	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

[图示]



第 4 章 端口功能

4.1 端口功能

端口引脚的输入 / 输出缓冲器电源只有 V_{DD} 一种。各电源和引脚的关系如下所示：

表 4-1 各引脚的输入 / 输出缓冲器电源

电源	对应引脚
V_{DD}	全部引脚

R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 具备数字输入 / 输出端口，可进行多种控制。各端口功能如表 4-2 所示。

除数字输入 / 输出端口功能以外，还具备各种复用功能。有关复用功能，请参照“第 2 章 引脚功能”。

表 4-2 端口功能

功能名称	输入 / 输出	功能	复位时	复用功能
P00	输入 / 输出	端口 0 2 位输入 / 输出端口 可以 1 位单位指定输入 / 输出 可通过软件设置，使用内部上拉电阻。	输入端口	TI000
P01				TI010/TO00
P10	输入 / 输出	端口 1 8 位输入 / 输出端口 可以 1 位单位指定输入 / 输出 可通过软件设置，使用内部上拉电阻。	输入端口	SCK10/TxD0
P11				SI10/RxD0
P12				SO10
P13				TxD6
P14				RxD6
P15				TOH0
P16				TOH1/INTP5
P17				TI50/TO50
P20 ~ P23	输入 / 输出	端口 2 4 位输入 / 输出端口 可以 1 位单位指定输入 / 输出	模拟输入	ANI0 ~ ANI3
P30	输入 / 输出	端口 3 4 位输入 / 输出端口 可以 1 位单位指定输入 / 输出 可通过软件设置，使用内部上拉电阻。	输入端口	INTP1
P31				INTP2 注 1
P32				INTP3 注 2
P33				INTP4/TI51/TO51
P40、P41	输入 / 输出	端口 4 2 位输入 / 输出端口 可以 1 位单位指定输入 / 输出 可通过软件设置，使用内部上拉电阻。	输入端口	—
P60	输入 / 输出	端口 6 2 位输入 / 输出端口 输出为 N 沟道漏极开路输出（6V 耐压） 可以 1 位单位指定输入 / 输出	输入端口	SCL0
P61				SDA0
P70、P71	输入 / 输出	端口 7 2 位输入 / 输出端口 可以 1 位单位指定输入 / 输出 可通过软件设置，使用内部上拉电阻。	输入端口	—
P120	输入 / 输出	端口 12 3 位输入 / 输出端口 可以 1 位单位指定输入 / 输出 仅 P120 可通过软件设置，使用内部上拉电阻。	输入端口	INTP0/EXLVI
P121				X1 注 3
P122				X2/EXCLK 注 4

- 注 1. 在 R7F0C999B2DFP 时，P31 与 INTP2/OCD1A 复用。
 2. 在 R7F0C999B2DFP 时，P32 与 INTP3/OCD1B 复用。
 3. 在 R7F0C999B2DFP 时，P121 与 X1/OCD0A 复用。
 4. 在 R7F0C999B2DFP 时，P122 与 X2/EXCLK/OCD0B 复用。

4.2 端口结构

端口由以下硬件构成。

表 4-3 端口结构

项目	结构
控制寄存器	端口模式寄存器 (PMxx): PM0 ~ PM4、PM6、PM7、PM12 端口寄存器 (Pxx): P0 ~ P4、P6、P7、P12 上拉电阻选择寄存器 (PUxx): PU0、PU1、PU3、PU4、PU7、PU12 A/D 端口配置寄存器 (ADPC)
端口	总计 27 个 (CMOS 输入 / 输出: 25 个、N 沟道漏极开路输入 / 输出: 2 个)
上拉电阻	总计 19 个

4.2.1 端口 0

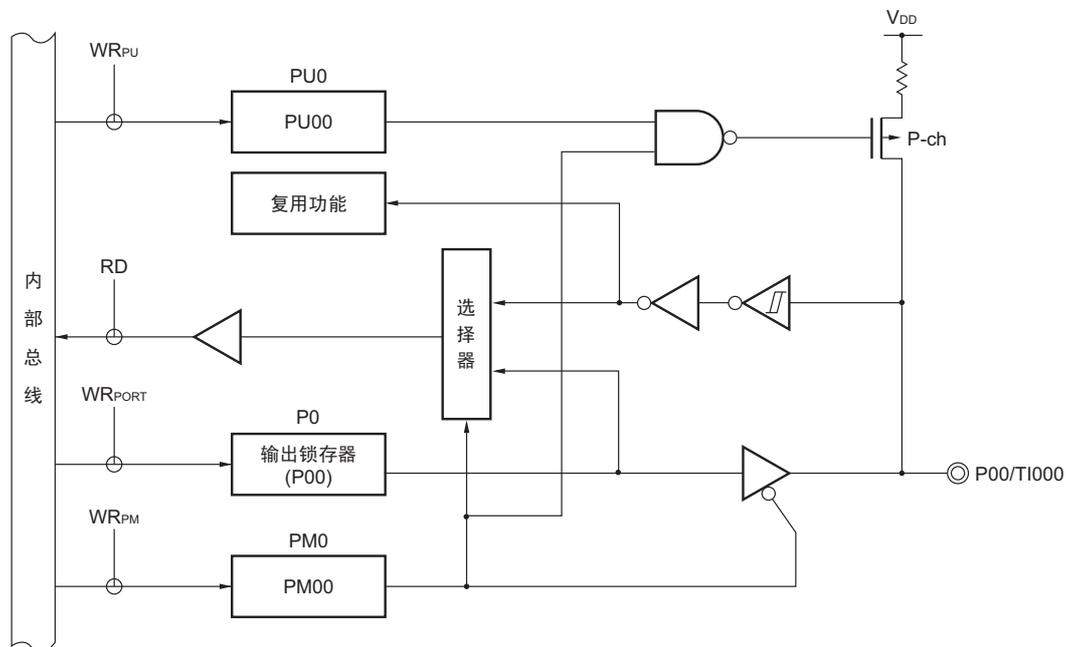
端口 0 是带输出锁存器的输入 / 输出端口。可通过端口模式寄存器 0 (PM0) 以 1 位单位指定为输入模式或者输出模式。将 P00、P01 引脚用作输入端口时, 可通过上拉电阻选择寄存器 0 (PU0) 以 1 位单位使用内部上拉电阻。

另外, 端口 0 还可复用作定时器输入 / 输出。

在产生复位信号后, 端口 0 变为输入模式。

端口 0 的框图如图 4-1、图 4-2 所示。

图 4-1 P00 的框图



P0: 端口寄存器 0

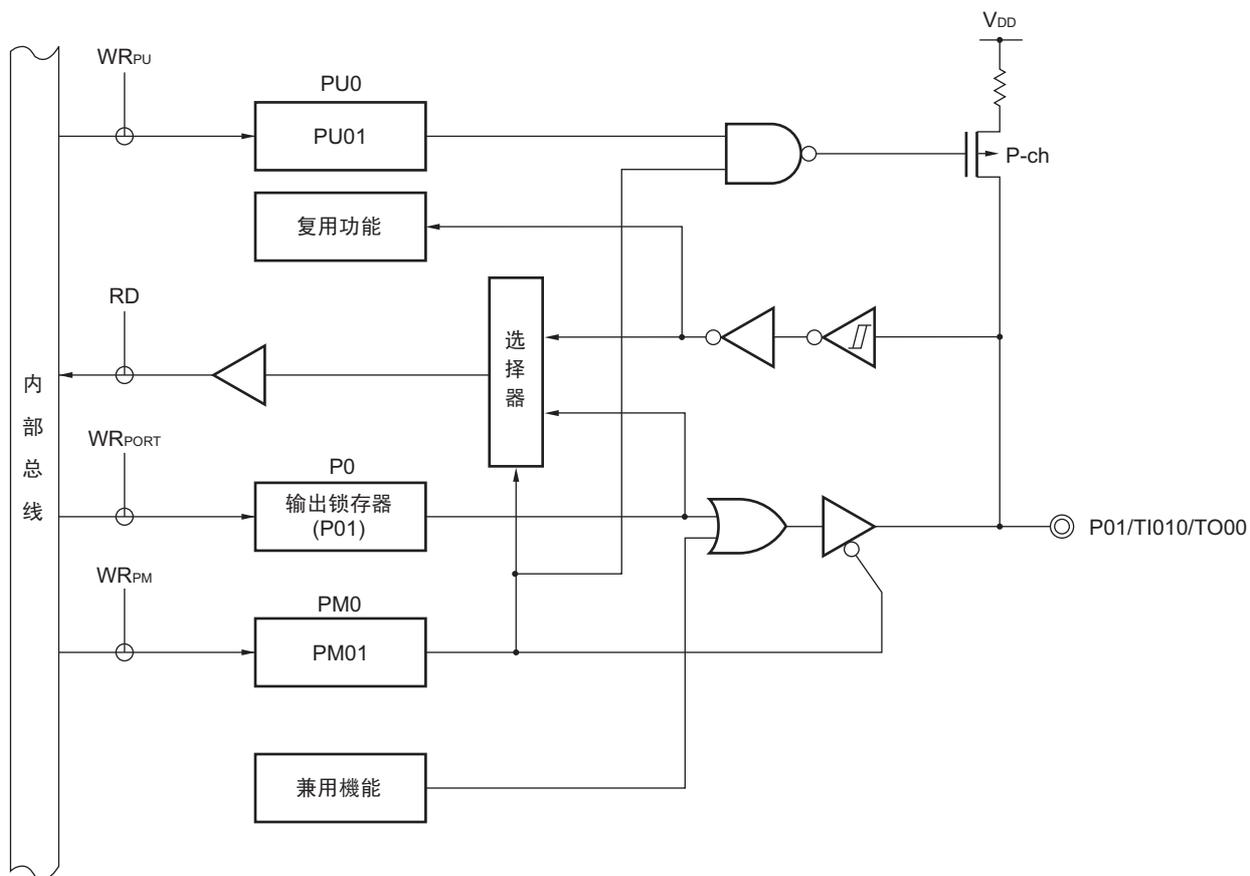
PU0: 上拉电阻选择寄存器 0

PM0: 端口模式寄存器 0

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

RD: 读信号

WR_{xx}: 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。可通过端口模式寄存器 1 (PM1) 以 1 位单位指定为输入模式或者输出模式。将 P10 ~ P17 引脚用作输入端口时, 可通过上拉电阻选择寄存器 1 (PU1) 以 1 位单位使用内部上拉电阻。

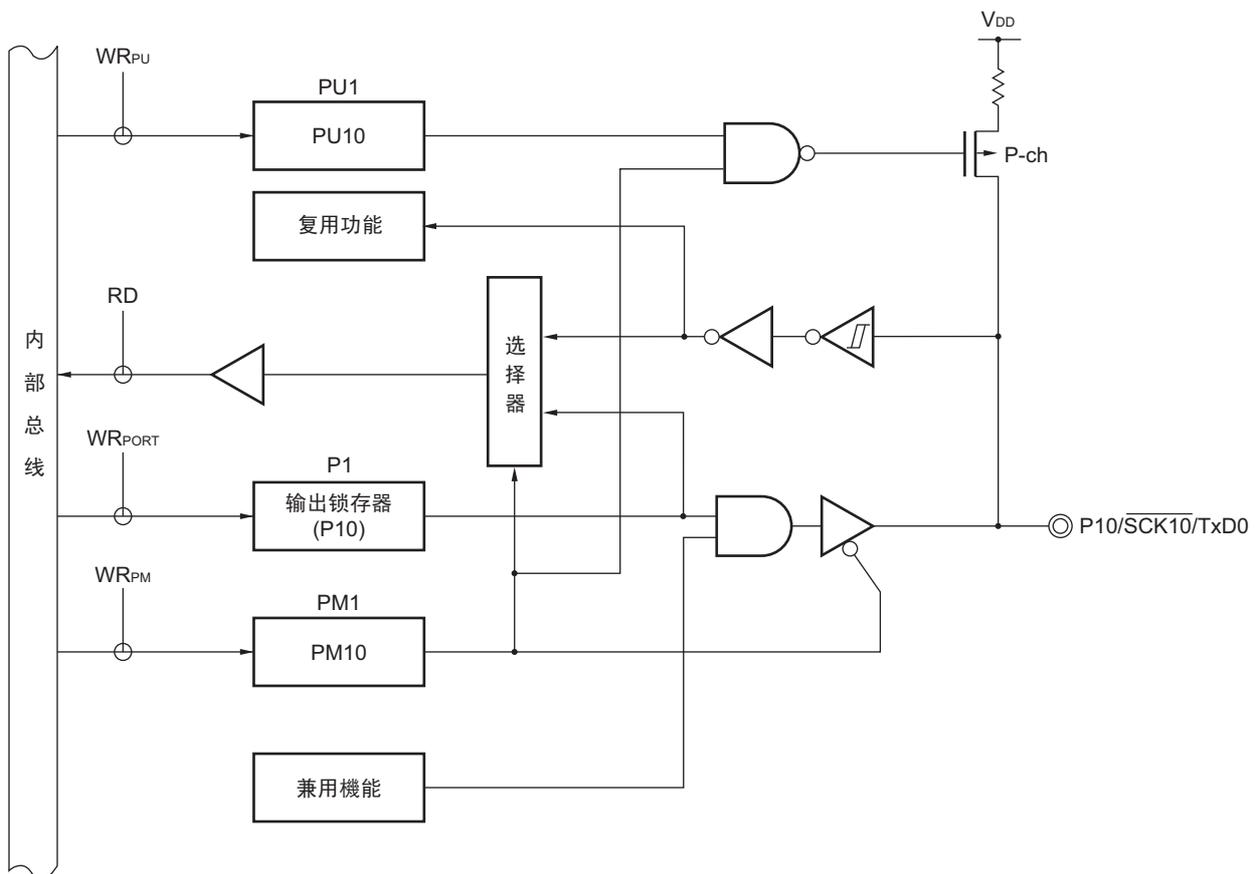
另外, 端口 1 还可复用作串行接口的数据输入 / 输出, 时钟输入 / 输出、定时器输入 / 输出以及外部中断请求输入。

在产生复位信号后, 端口 1 变为输入模式。

端口 1 的框图如图 4-3 ~ 图 4-7 所示。

- 注意 1. 将 P10/SCK10/TxD0、P12/SO10 用作通用端口时, 必须将串行运行模式寄存器 10 (CSIM10) 和串行时钟选择寄存器 10 (CSIC10) 设置为与初始状态相同 (00H)。
2. 将 P13/TxD6 用作通用端口时, 必须将异步串行接口控制寄存器 6 (ASICL6) 的 bit0 (TXDLV6) 置“0” (TxD6 通常输出)。

图 4-3 P10 的框图



P1: 端口寄存器 1

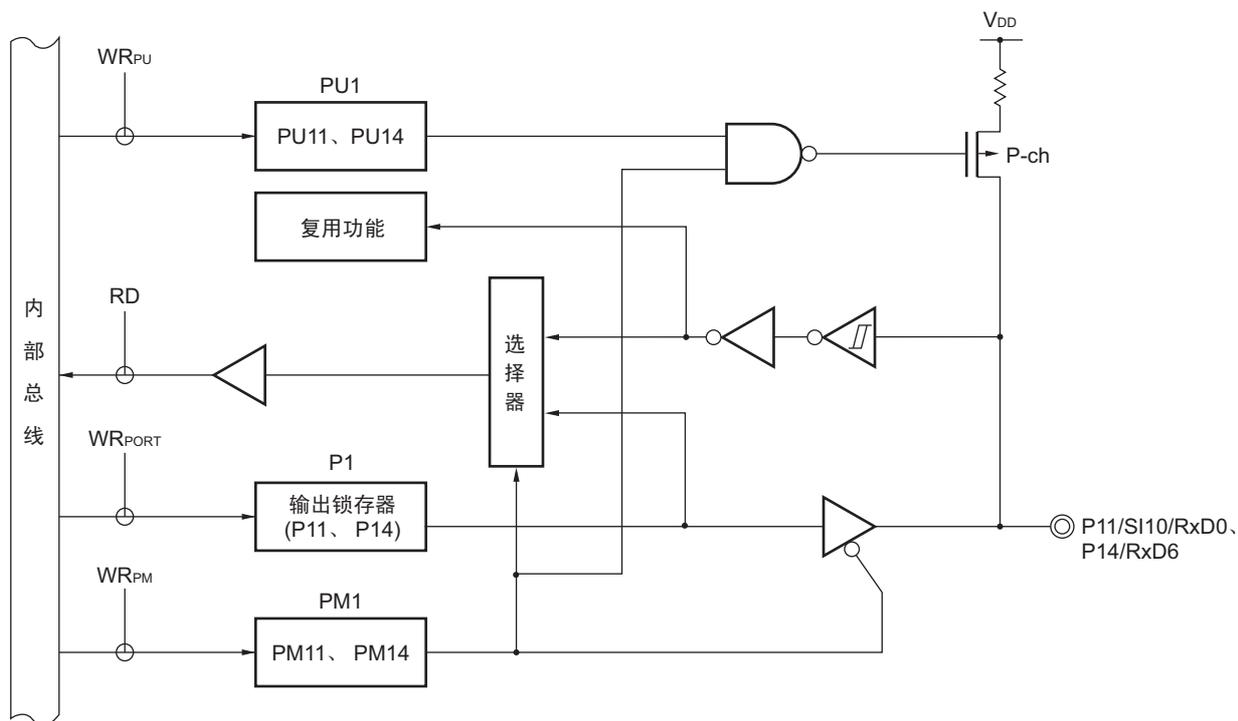
PU1: 上拉电阻选择寄存器 1

PM1: 端口模式寄存器 1

RD: 读信号

WR_{xx}: 写信号

图 4-4 P11、P14 的框图



P1: 端口寄存器 1

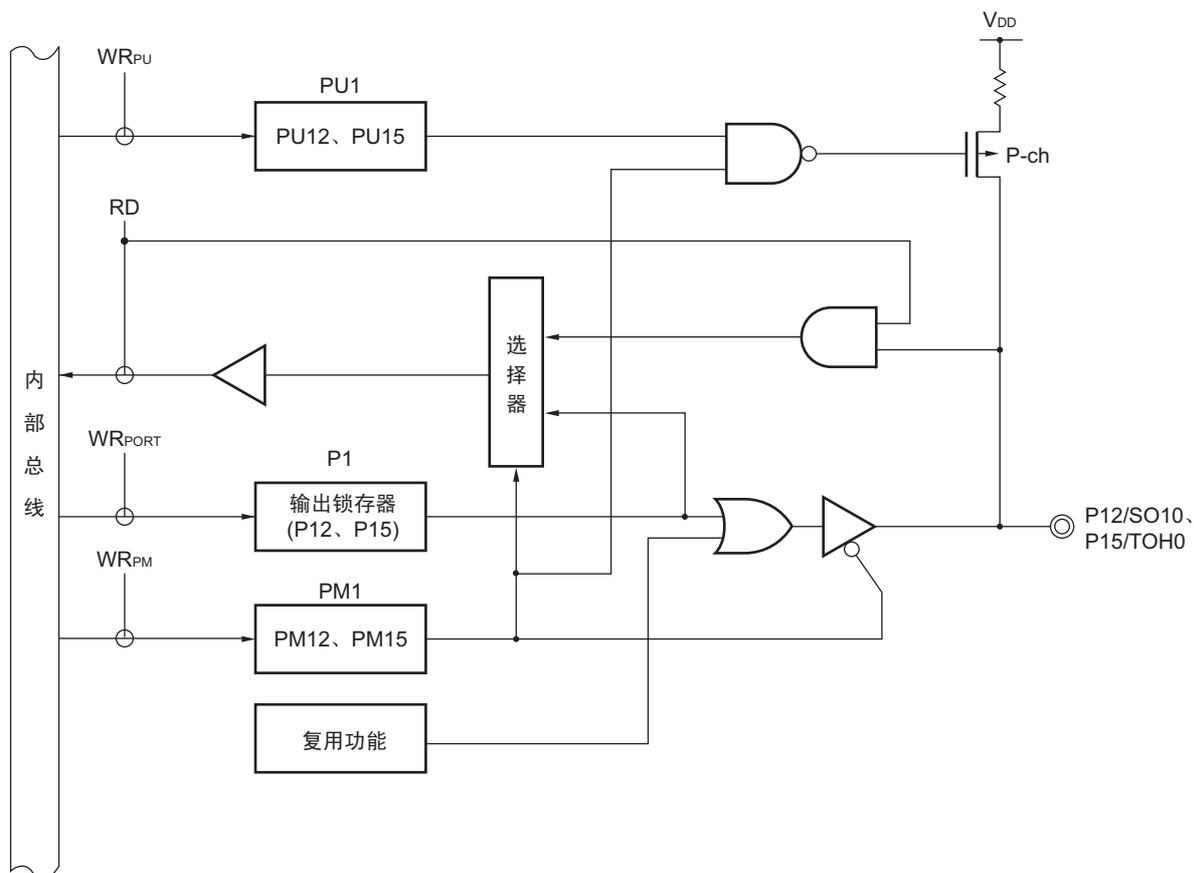
PU1: 上拉电阻选择寄存器 1

PM1: 端口模式寄存器 1

RD: 读信号

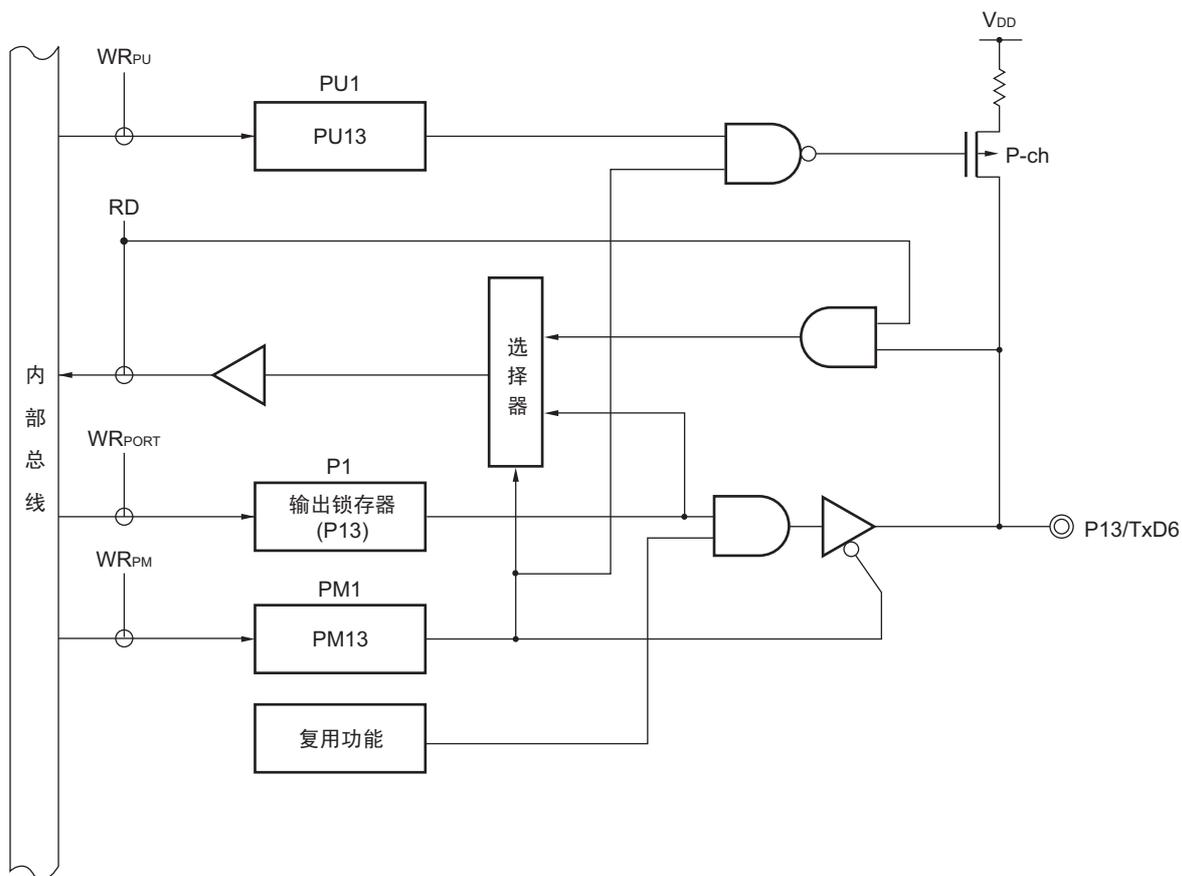
WR_{xx}: 写信号

图 4-5 P12、P15 的框图



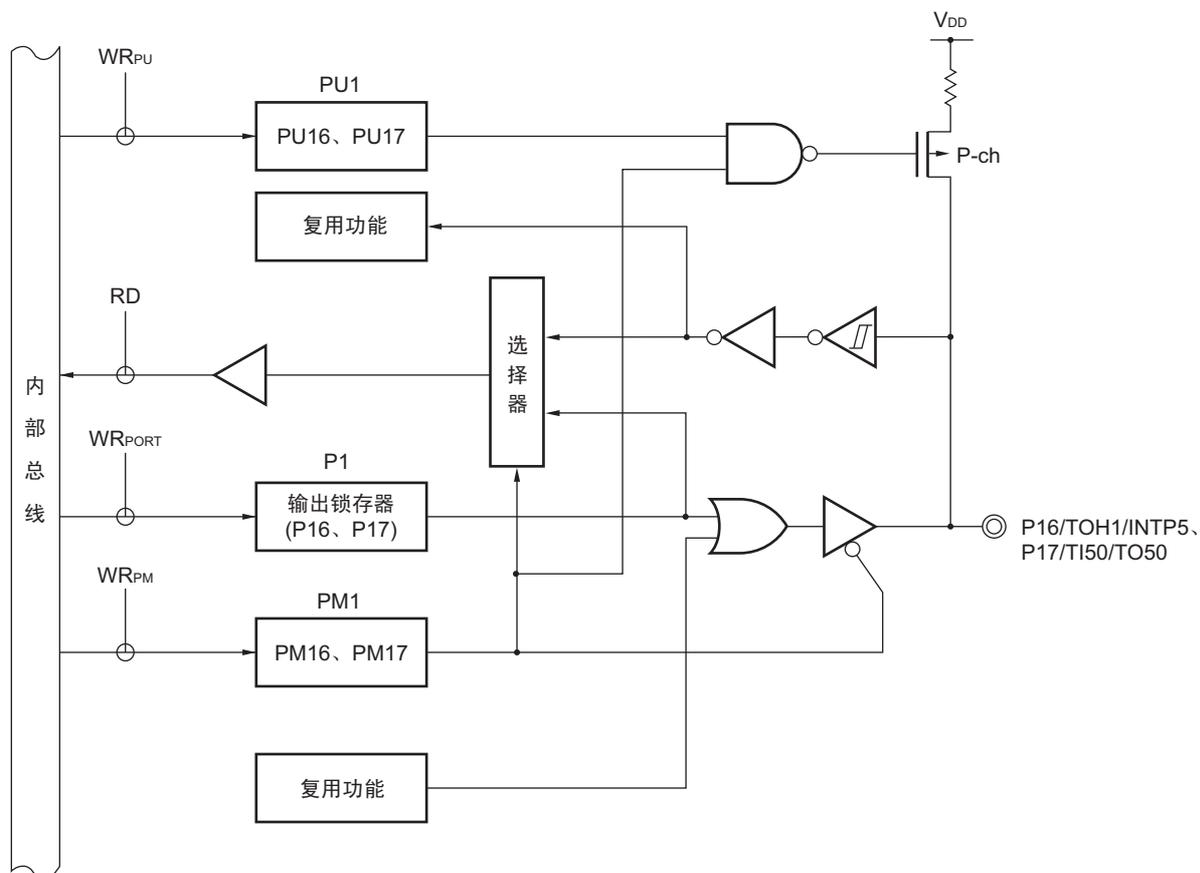
- P1: 端口寄存器 1
 PU1: 上拉电阻选择寄存器 1
 PM1: 端口模式寄存器 1
 RD: 读信号
 WR_{xx}: 写信号

图 4-6 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
- RD: 读信号
- WR_{xx}: 写信号

图 4-7 P16、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
 RD: 读信号
 WR_{xx}: 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。可通过端口模式寄存器 2 (PM2) 以 1 位单位指定为输入模式或者输出模式。

另外，端口 2 还可复用作 A/D 转换器的模拟输入。

如果要将 P20/ANI0 ~ P23/ANI3 用作数字输入，必须在通过 A/D 端口配置寄存器 (ADPC) 设置为数字输入 / 输出并且通过 PM2 设置为输入模式后，从低位开始使用。

如果要将 P20/ANI0 ~ P23/ANI3 用作数字输出，必须通过 ADPC 设置为数字输入 / 输出并且通过 PM2 设置为输出模式。

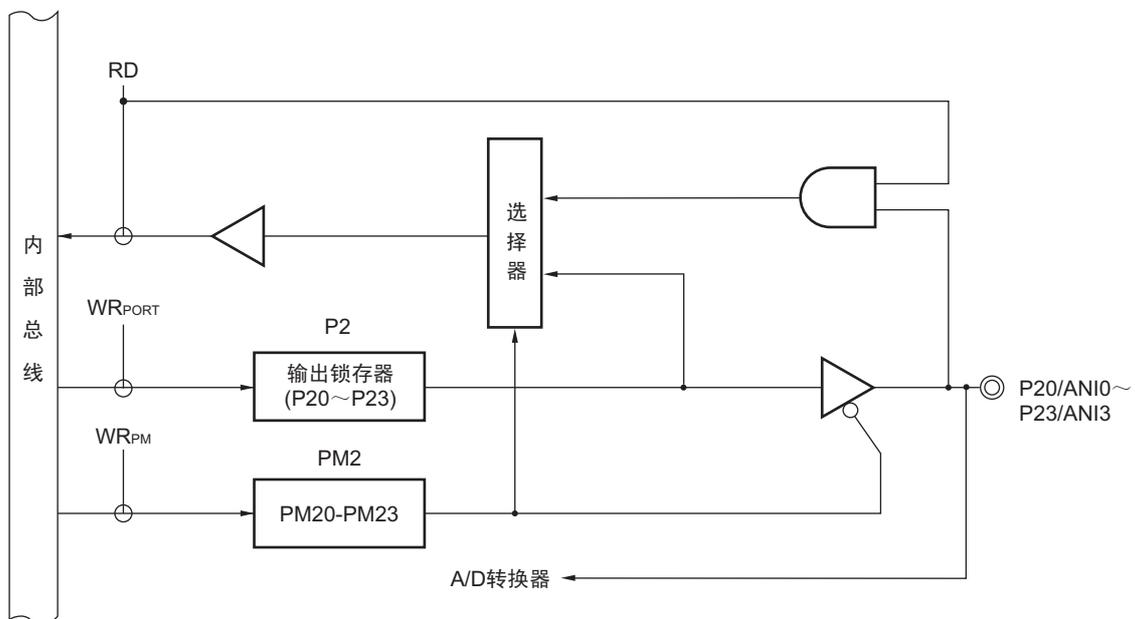
表 4-4 P20/ANI0 ~ P23/ANI3 引脚功能的设置

ADPC 寄存器	PM2 寄存器	ADS 寄存器	P20/ANI0 ~ P23/ANI3 引脚
选择数字输入 / 输出	输入模式	—	数字输入
	输出模式	—	数字输出
选择模拟输入	输入模式	选择 ANIn	模拟输入 (A/D 转换对象)
		不选择 ANIn	模拟输入 (非 A/D 转换对象)
	输出模式	选择 ANIn	禁止设置
		不选择 ANIn	

在产生复位信号后，P20/ANI0 ~ P23/ANI3 全部变为模拟输入。

端口 2 的框图如图 4-8 所示。

图 4-8 P20 ~ P23 的框图



P2: 端口寄存器 2

PM2: 端口模式寄存器 2

RD: 读信号

WRxx: 写信号

4.2.4 端口 3

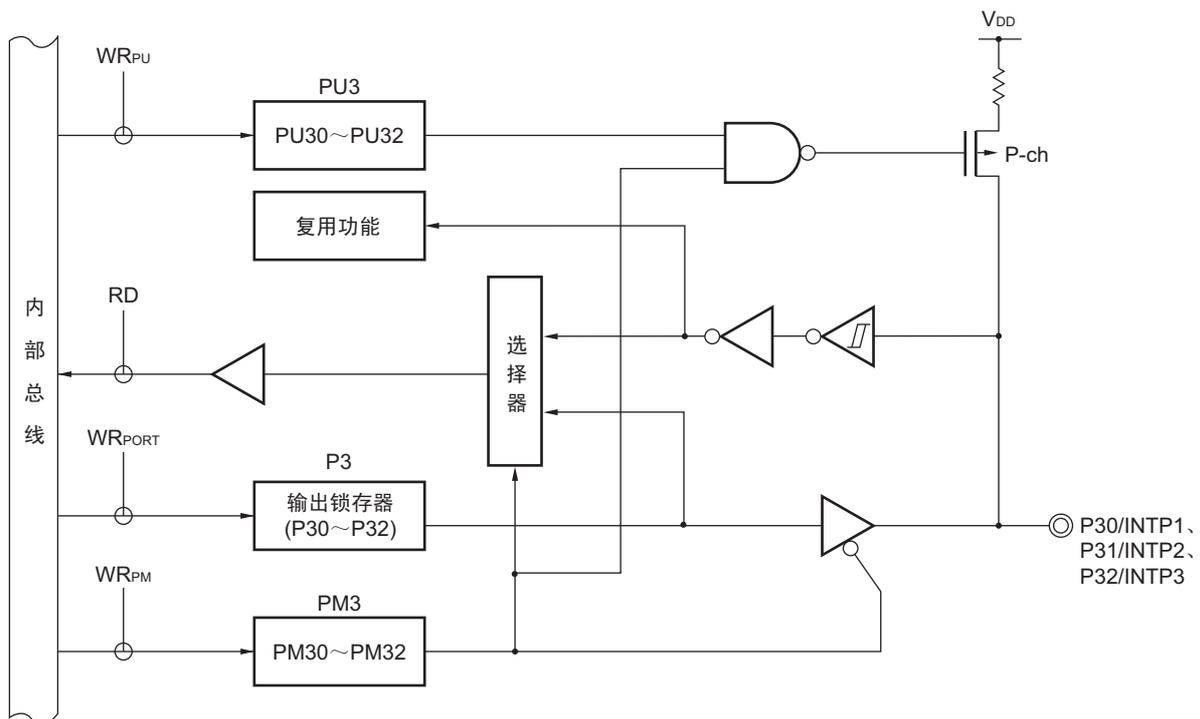
端口 3 是带输出锁存器的输入 / 输出端口。可通过端口模式寄存器 3 (PM3) 以 1 位单位指定为输入模式或者输出模式。将 P30 ~ P33 引脚用作输入端口时, 可通过上拉电阻选择寄存器 3 (PU3) 以 1 位单位使用内部上拉电阻。

另外, 端口 3 还可复用作外部中断请求输入和定时器的输入 / 输出。

在产生复位信号后, 端口 3 变为输入模式。

端口 1 的框图如图 4-9、图 4-10 所示。

图 4-9 P30 ~ P32 的框图



P3: 端口寄存器 3

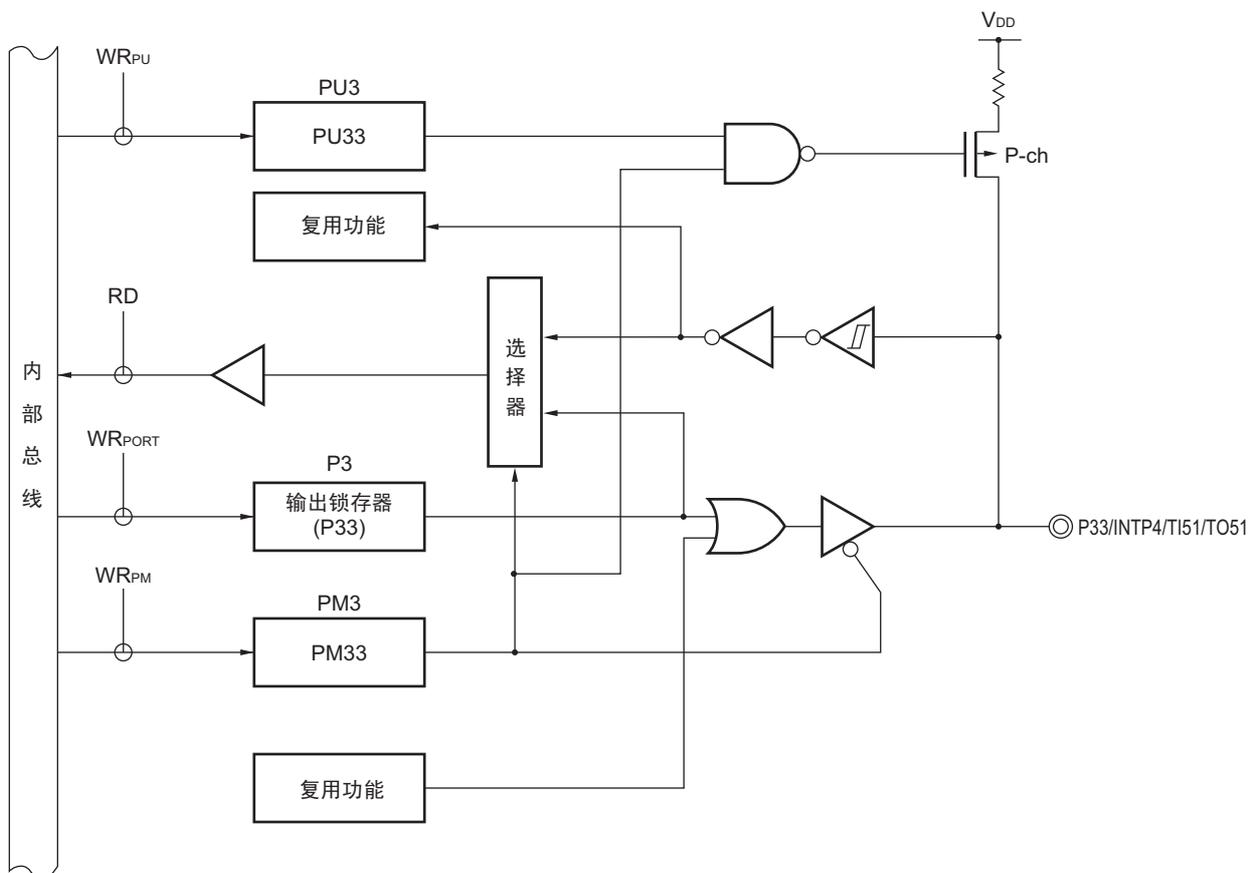
PU3: 上拉电阻选择寄存器 3

PM3: 端口模式寄存器 3

RD: 读信号

WR_{xx}: 写信号

图 4-10 P33 的框图



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

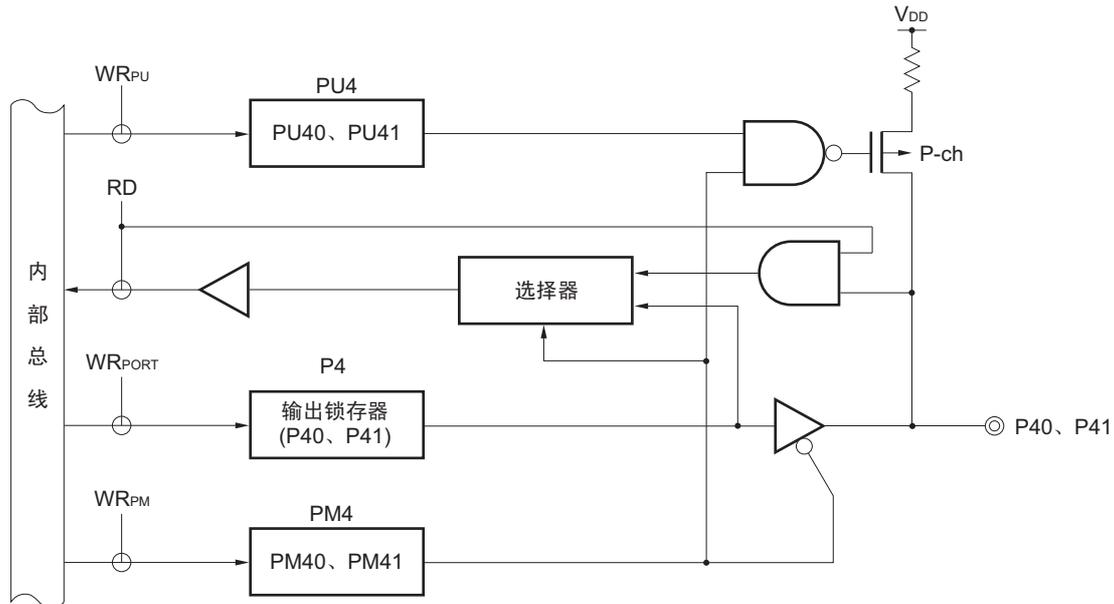
4.2.5 端口 4

端口 4 是带输出锁存器的输入 / 输出端口。可通过端口模式寄存器 4 (PM4) 以 1 位单位指定为输入模式或者输出模式。将 P40、P41 引脚用作输入端口时, 可通过上拉电阻选择寄存器 4 (PU4) 以 1 位单位使用内部上拉电阻。

在产生复位信号后, 端口 4 变为输入模式。

端口 4 的框图如图 4-11 所示。

图 4-11 P40、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

RD: 读信号

WR_{xx}: 写信号

4.2.6 端口 6

端口 6 是带输出锁存器的输入 / 输出端口。可通过端口模式寄存器 6 (PM6) 以 1 位单位指定为输入模式或者输出模式。

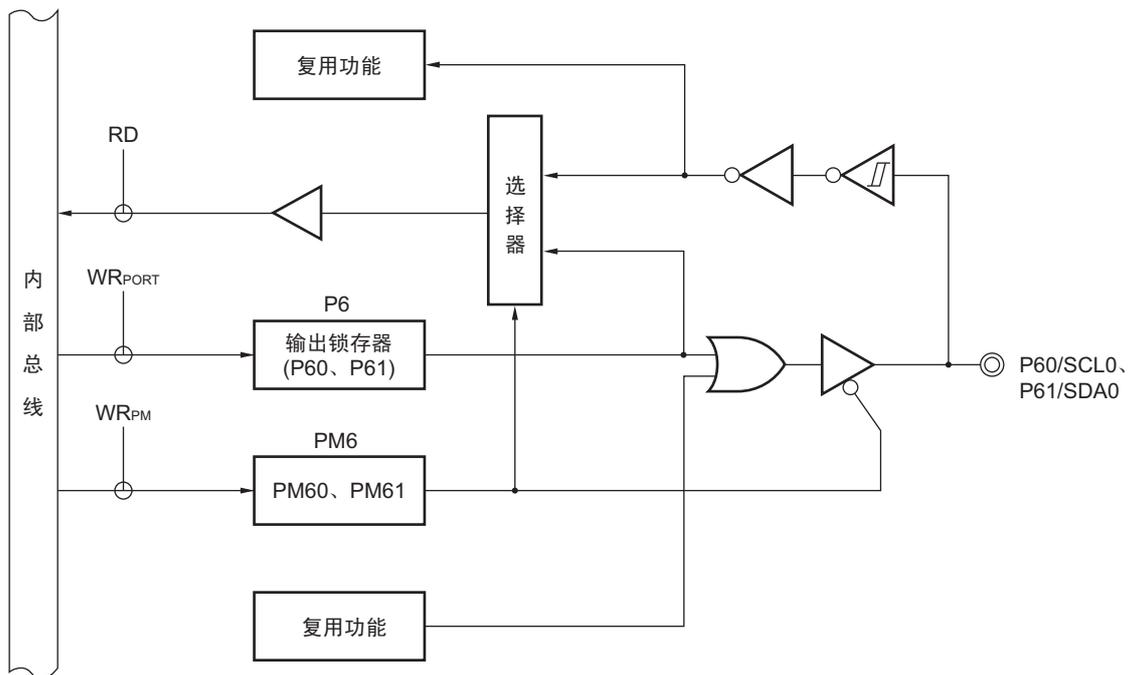
P60 和 P61 的输出为 N 沟道漏极开路输出 (6V 耐压)。

另外, 端口 6 还可复用作串行接口的数据输入 / 输出和时钟的输入 / 输出。

在产生复位信号后, 端口 6 变为输入模式。

端口 6 的框图如图 4-12 所示。

图 4-12 P60、P61 的框图



P6: 端口寄存器 6

PM6: 端口模式寄存器 6

RD: 读信号

WR_{xx}: 写信号

注意 因为在 P60 和 P61 为输出模式时, 输入缓冲器也为 ON, 所以在输入中间电位时, 有直通电流流过。因此, 在 P60 和 P61 为输出模式时, 不要输入中间电位。

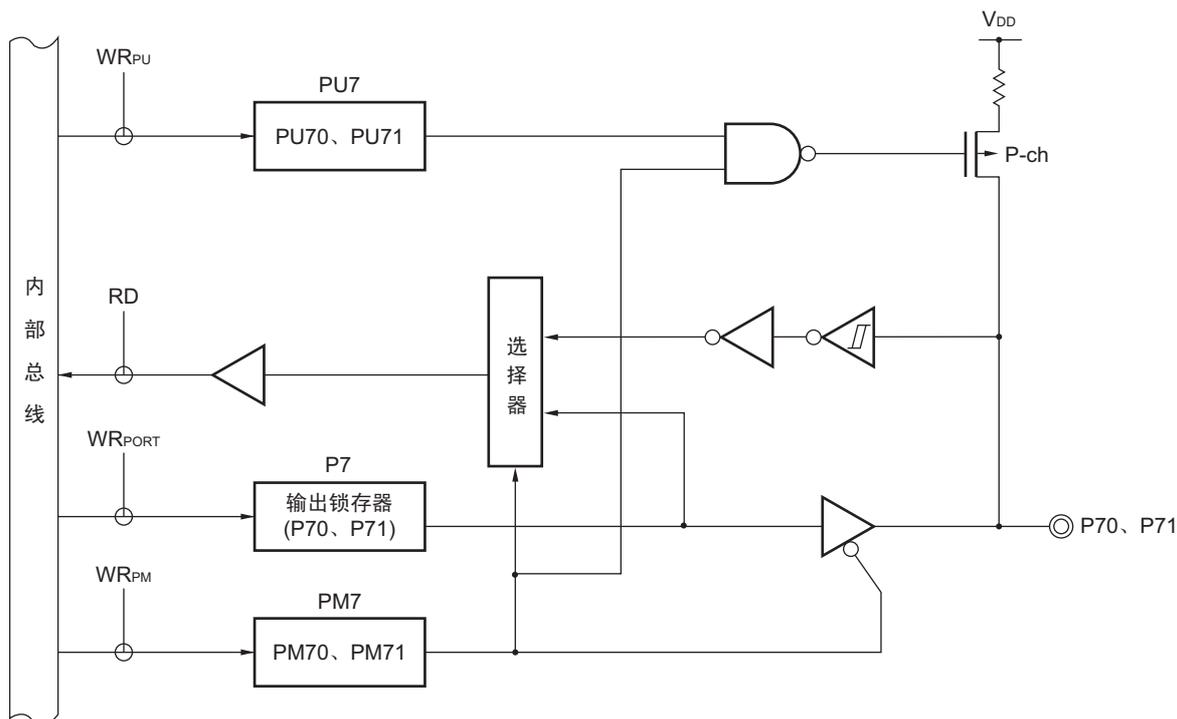
4.2.7 端口 7

端口 7 是带输出锁存器的输入 / 输出端口。可通过端口模式寄存器 7 (PM7) 以 1 位单位指定为输入模式或者输出模式。将 P70、P71 引脚用作输入端口时, 可通过上拉电阻选择寄存器 7 (PU7) 以 1 位单位使用内部上拉电阻。

在产生复位信号后, 端口 7 变为输入模式。

端口 7 的框图如图 4-13 所示。

图 4-13 P70、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

RD: 读信号

WRxx: 写信号

4.2.8 端口 12

端口 12 是带输出锁存器的输入 / 输出端口。可通过端口模式寄存器 12 (PM12) 以 1 位单位指定为输入模式或者输出模式。仅 P120 在用作输入端口时, 可通过上拉电阻选择寄存器 12 (PU12) 使用内部上拉电阻。

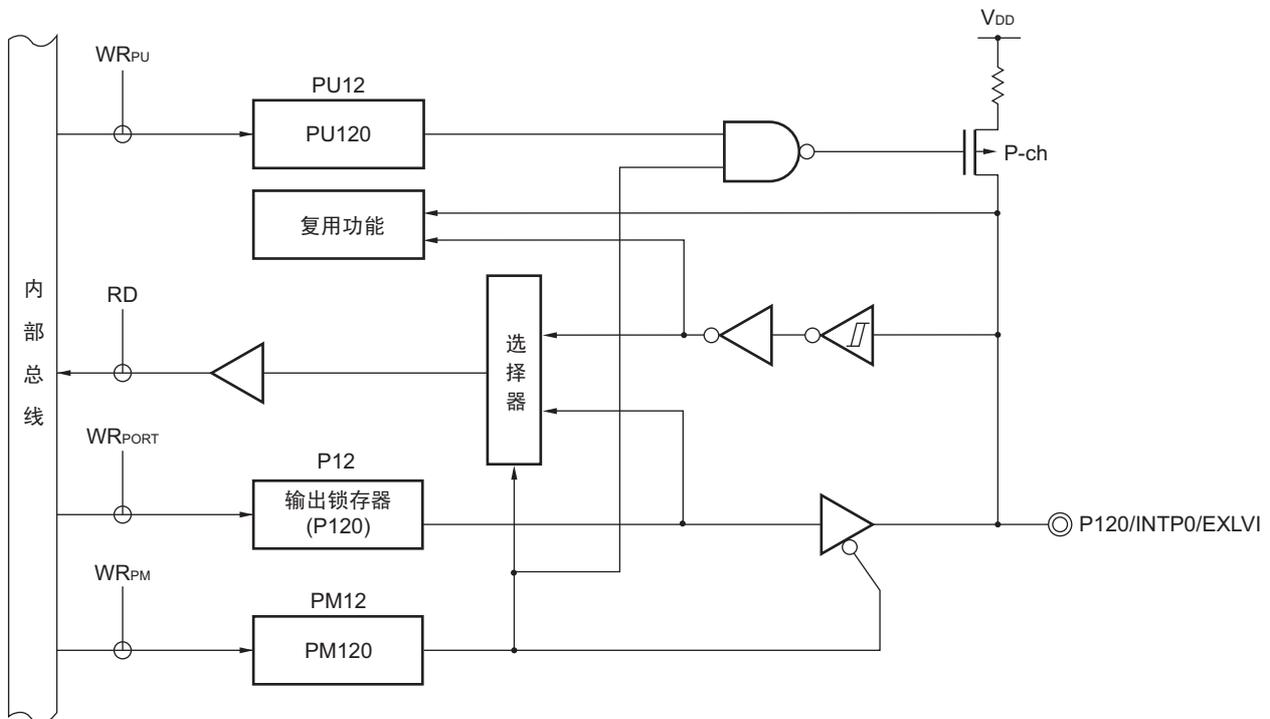
端口 12 还可复用作外部中断请求输入、外部低电压检测电位输入、主系统时钟的谐振器连接以及主系统时钟的外部时钟输入。

在产生复位信号后, 端口 12 变为输入模式。

端口 12 的框图如图 4-14、图 4-15 所示。

注意 如果要将 P121、P122 用作与主系统时钟 (X1、X2) 谐振器的连接, 或者主系统时钟的外部时钟输入 (EXCLK), 必须通过时钟运行模式选择寄存器 (OSCCTL) 设置为 X1 振荡模式或者外部时钟输入模式 (详细内容请参照“5.3 (1) 时钟运行模式选择寄存器 (OSCCTL)”)。OSCCTL 的复位值为“00H” (P121 和 P122 全部为输入 / 输出端口)。此时, 不需要设置 PM121、PM122、P121 以及 P122。

图 4-14 P120 的框图



P12: 端口寄存器 12

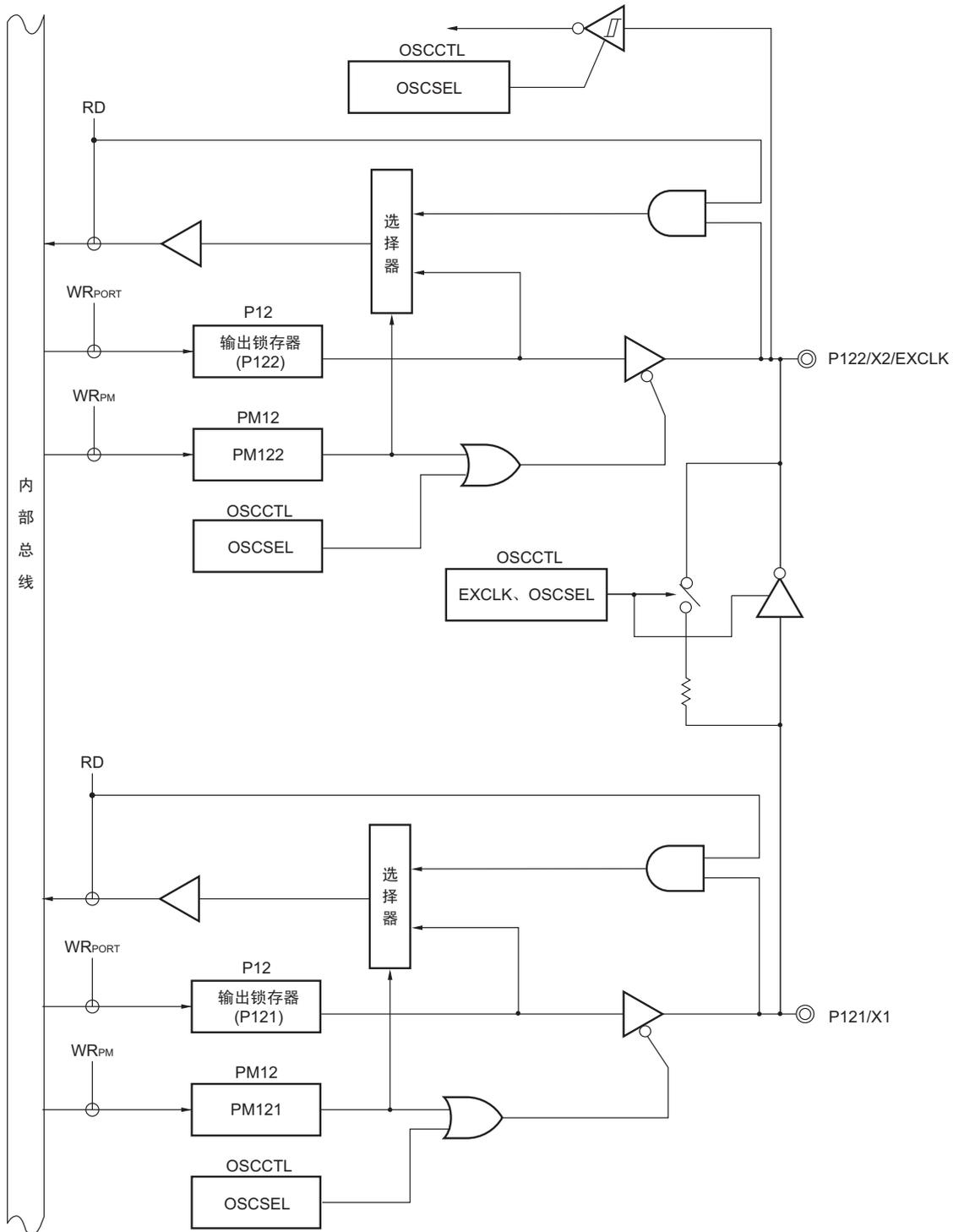
PU12: 上拉电阻选择寄存器 12

PM12: 端口模式寄存器 12

RD: 读信号

WRxx: 写信号

图 4-15 P121、P122 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
 OSCCTL: 时钟运行模式选择寄存器
 RD: 读信号
 WRxx: 写信号

4.3 端口功能的控制寄存器

通过以下 4 种寄存器控制端口。

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

(1) 端口模式寄存器 (PMxx)

端口模式寄存器可以 1 位单位设置端口的输入或者输出。

通过 1 位或 8 位存储器操作指令设置端口模式寄存器。

在产生复位信号后，端口模式寄存器变为“FFH”。

将端口引脚用作复用功能的引脚时，请参照“4.5 使用复用功能时的端口模式寄存器和输出锁存器的设置”进行设置。

图 4-16 端口模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	1	1	1	1	1	PM01	PM00	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	1	1	1	1	PM23	PM122	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM4	1	1	1	1	1	1	PM41	PM40	FF24H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FF26H	FFH	R/W
PM7	1	1	1	1	1	1	PM71	PM70	FF27H	FFH	R/W
PM12	1	1	1	1	1	PM122	PM121	PM120	FF2CH	FFH	R/W
PMmn	选择 Pmn 引脚的输入 / 输出模式 (m=0 ~ 4、6、7、12；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 将 PM0 的 bit2 ~ bit7、PM2 的 bit4 ~ bit7、PM3 的 bit4 ~ bit7、PM4 的 bit2 ~ bit7、PM6 的 bit2 ~ bit7、PM7 的 bit2 ~ bit7 以及 PM12 的 bit3 ~ bit7 置“1”。

(2) 端口寄存器 (Pxx)

端口寄存器在端口输出时写片外输出数据。

如果在输入模式下读端口寄存器，则读取引脚电平；如果在输出模式下读端口寄存器，则读取端口输出锁存器的值。

通过 1 位或 8 位存储器操作指令设置端口寄存器。

在产生复位信号后，端口寄存器变为“00H”。

图 4-17 端口寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
P0	0	0	0	0	0	0	P01	P00	FF00H	00H (输出锁存器)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FF01H	00H (输出锁存器)	R/W
P2	0	0	0	0	P23	P22	P21	P20	FF02H	00H (输出锁存器)	R/W
P3	0	0	0	0	P33	P32	P31	P30	FF03H	00H (输出锁存器)	R/W
P4	0	0	0	0	0	0	P41	P40	FF04H	00H (输出锁存器)	R/W
P6	0	0	0	0	0	0	P61	P60	FF06H	00H (输出锁存器)	R/W
P7	0	0	0	0	0	0	P71	P70	FF07H	00H (输出锁存器)	R/W
P12	0	0	0	0	0	P122注	P121注	P120	FF0CH	00H (输出锁存器)	R/W

Pmn	m=0 ~ 4、6、7、12；n=0 ~ 7	
	输出数据的控制 (输出模式下)	输入数据的读取 (输入模式下)
0	输出“0”	输入低电平
1	输出“1”	输入高电平

注 在引脚模式为外部时钟输入模式时，P121 和 P122 的输出锁存器的读取值总是为“0”。

注意 将 P0 的 bit2 ~ bit7、P2 的 bit4 ~ bit7、P3 的 bit4 ~ bit7、P4 的 bit2 ~ bit7、P6 的 bit2 ~ bit7、P7 的 bit2 ~ bit7 以及 P12 的 bit3 ~ bit7 置“0”。

(3) 上拉电阻选择寄存器 (PUxx)

上拉电阻选择寄存器设置是否使用内部上拉电阻。在指定通过上拉电阻选择寄存器使用内部上拉电阻的引脚中，只有设置为输入模式的位，可以 1 位单位使用内部上拉电阻。设置为输出模式的位，与上拉电阻选择寄存器的设置无关，不能连接内部上拉电阻。用作复用功能的输出引脚时也相同。

通过 1 位或 8 位存储器操作指令设置上拉电阻选择寄存器。

在产生复位信号后，上拉电阻选择寄存器变为“00H”。

图 4-18 上拉电阻选择寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU0	0	0	0	0	0	0	PU01	PU00	FF30H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W
PU3	0	0	0	0	PU33	PU32	PU31	PU30	FF33H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	FF34H	00H	R/W
PU7	0	0	0	0	0	0	PU71	PU70	FF37H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	FF3CH	00H	R/W
PUmn	选择 Pmn 的内部上拉电阻 (m=0、1、3、4、7、12；n=0~7)										
0	不连接内部上拉电阻										
1	连接内部上拉电阻										

(4) A/D 端口配置寄存器 (ADPC)

A/D 端口配置寄存器将 P20/ANI0 ~ P23/ANI3 引脚切换为端口的数字输入 / 输出或者 A/D 转换器的模拟输入。

通过 1 位或 8 位存储器操作指令设置 ADPC。

在产生复位信号后，ADPC 变为“00H”。

图 4-19 A/D 端口配置寄存器 (ADPC) 的格式

地址: FF2FH	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	切换为数字输入 / 输出 (D) 或者模式输入 (A)			
				P23/ANI3	P22/ANI2	P21/ANI1	P20/ANI0
0	0	0	0	A	A	A	A
0	0	0	1	A	A	A	D
0	0	1	0	A	A	D	D
0	0	1	1	A	D	D	D
0	1	0	0	D	D	D	D
1	0	0	0	D	D	D	D
上述以外				禁止设置			

- 注 1. 通过端口模式寄存器 2 (PM2)，将 A/D 转换时使用的通道设置为输入模式。
2. 如果将数据写入 ADPC，就产生等待。当外围硬件时钟停止时，不能将数据写入 ADPC。详细内容请参照“第 27 章 等待的注意事项”。

4.4 端口功能的运行

端口的运行因如下所示的输入 / 输出模式的设置而不同。

4.4.1 输入 / 输出端口的写入

(1) 输出模式的情况

通过传送指令将值写入输出锁存器。输出锁存器的内容从引脚输出。
一旦数据被写入输出锁存器，就保持该数据直到再次将数据写入输出锁存器。
在产生复位信号后，清除输出锁存器的数据。

(2) 输入模式的情况

通过传送指令将值写入输出锁存器。因为输出缓冲器处于 OFF 状态，所以引脚状态不会发生变化。
一旦数据被写入输出锁存器，就保持该数据直到再次将数据写入输出锁存器。
在产生复位信号后，清除输出锁存器的数据。

4.4.2 输入 / 输出端口的读取

(1) 输出模式的情况

通过传送指令读取输出锁存器的内容。输出锁存器的内容不发生变化。

(2) 输入模式的情况

通过传送指令读取引脚的状态。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

与输出锁存器的内容进行运算，结果写入输出锁存器。锁存器的内容从引脚输出。
一旦数据被写入输出锁存器，就保持该数据直到再次将数据写入输出锁存器。
在产生复位信号后，清除输出锁存器的数据。

(2) 输入模式的情况

在读取引脚电平后，与其内容进行运算，结果写入输出锁存器。因为输出缓冲器处于 OFF 状态，所以引脚状态不会发生变化。
在产生复位信号后，清除输出锁存器的数据。

4.5 使用复用功能时的端口模式寄存器和输出锁存器的设置

在将端口引脚作为复用功能的引脚使用时，必须按表 4-5 设置端口模式寄存器和输出锁存器。

表 4-5 使用复用功能时的端口模式寄存器和输出锁存器的设置 (1/2)

引脚名称	复用功能		PMxx	Pxx
	名称	输入 / 输出		
P00	TI000	输入	1	x
P01	TI010	输入	1	x
	TO00	输出	0	0
P10	SCK10	输入	1	x
		输出	0	1
	TxD0	输出	0	1
P11	SI10	输入	1	x
	RxD0	输入	1	x
P12	SO10	输出	0	0
P13	TxD6	输出	0	1
P14	RxD6	输入	1	x
P15	TOH0	输出	0	0
P16	TOH1	输出	0	0
	INTP5	输入	1	x
P17	TI50	输入	1	x
	TO50	输出	0	0
P20 ~ P23 注	ANI0 ~ ANI3 注	输入	1	x

注 ANI0/P20 ~ ANI3/P23 引脚的功能取决于 A/D 端口配置寄存器 (ADPC)、模拟输入通道指定寄存器 (ADS) 以及 PM2 寄存器的设置。

ADPC	PM2	ADS	ANI0/P20 ~ ANI3/P23 引脚
选择模拟输入	输入模式	选择 ANI	模拟输入 (A/D 转换对象)
		不选择 ANI	模拟输入 (非 A/D 转换对象)
	输出模式	选择 ANI	禁止设置
		不选择 ANI	
选择数字输入 / 输出	输入模式	—	数字输入
	输出模式	—	数字输出

备注 x: 忽略

PMxx: 端口模式寄存

Pxx: 端口的输出锁存器

表 4-5 使用复用功能时的端口模式寄存器和输出锁存器的设置 (2/2)

引脚名称	复用功能		PMxx	Pxx
	名称	输入 / 输出		
P30 ~ P32	INTP1 ~ INTP3	输入	1	x
P33	INTP4	输入	1	x
	TI51	输入	1	x
	TO51	输出	0	0
P60	SCL0	输入 / 输出	0	0
P61	SDA0	输入 / 输出	0	0
P120	INTP0	输入	1	x
	EXLVI	输入	1	x
P121	X1 注	—	x	x
P122	X2 注	—	x	x
	EXCLK 注	输入	x	x

注 在将 P121 和 P122 引脚用作与主系统时钟的谐振器连接 (X1、X2) 或者主系统时钟的外部时钟输入 (EXCLK) 时, 必须通过 OSCCTL 寄存器设置为 X1 振荡模式或者外部时钟输入模式 (详细内容请参照“5.3 (1) 时钟运行模式选择寄存器 (OSCCTL)”)。OSCCTL 寄存器的复位值为“00H” (P121 和 P122 全部为输入 / 输出端口)。此时, 不需要设置 PM121、PM122、P121 和 P122。

备注 x: 忽略

PMxx: 端口模式寄存器

Pxx: 端口的输出锁存器

4.6 端口寄存器 n (Pn) 的 1 位存储器操作指令的注意事项

在对输入 / 输出端口执行 1 位存储器操作指令时，不仅是操作对象位，非操作对象的输入端口的输出锁存值也有可能被改写。

因此，在将任意的端口从输入模式切换为输出模式前，推荐重写输出锁存器的值。

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

说明：PMnm 位为“1”的端口的 Pn 寄存器的写入对象和读取对象分别为输出锁存器和引脚状态。

在 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 内部按以下顺序执行 1 位存储器操作指令。

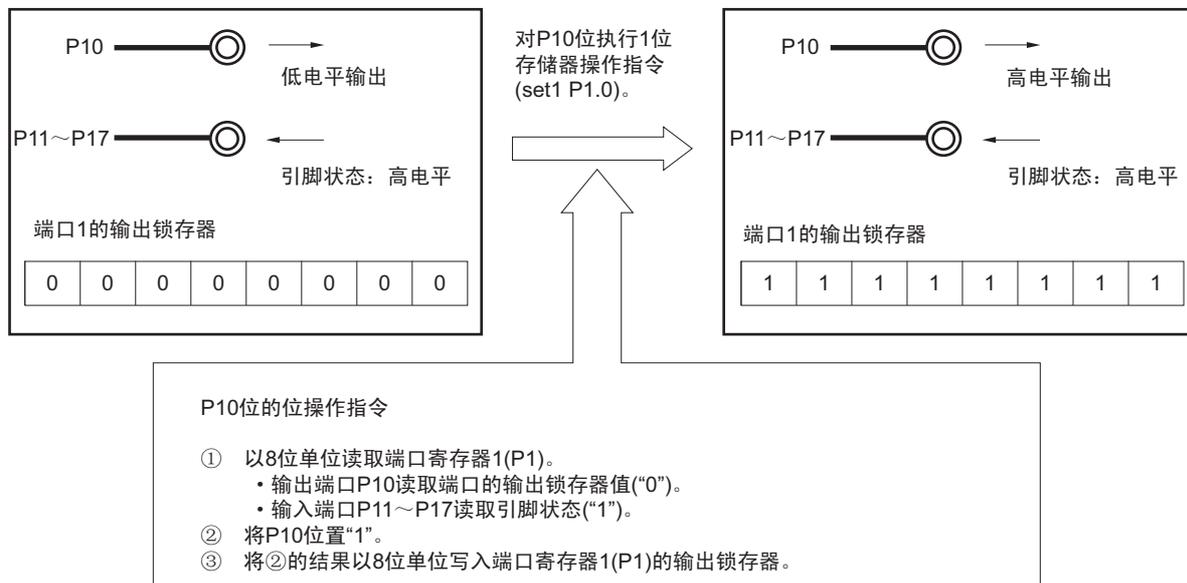
- <1> 以 8 位单位读取 Pn 寄存器。
- <2> 对作为对象的 1 位进行操作。
- <3> 以 8 位单位写入 Pn 寄存器。

在步骤<1>中，输出端口 P10 读取输出锁存的值（“0”），输入端口 P11 ~ P17 读取引脚状态。此时，如果 P11 ~ P17 的引脚状态为高电平，读取值就为“FEH”。

通过步骤<2> 读取值变为“FFH”。

通过步骤<3>，将“FFH”写入输出锁存器。

图 4-20 1 位存储器操作指令（P10 的情况）



第 5 章 时钟发生电路

5.1 时钟发生电路的功能

时钟发生电路是产生提供给 CPU 和外围硬件的时钟的电路。
系统时钟和系统振荡电路的种类如下所示。

(1) 主系统时钟

① X1 振荡电路

通过给 X1、X2 连接谐振器，产生 $f_X=1 \sim 10 \text{ MHz}$ 的振荡时钟。

可通过执行 STOP 指令或者设置主 OSC 控制寄存器（MOC），停止时钟振荡。

② 高速内部振荡电路

产生 $f_{RH}=8\text{MHz(TYP)}$ 的振荡时钟。在复位解除后，CPU 必须以该高速内部振荡时钟开始运行。

可通过执行 STOP 指令或者设置内部振荡模式寄存器（RCM），停止时钟振荡。

另外，可从 EXCLK/X2/P122 引脚提供外部主系统时钟（ $f_{EXCLK}=1 \sim 10\text{MHz}$ ）。通过执行 STOP 指令或者设置 RCM，使外部主系统时钟输入无效。

主系统时钟可通过主时钟模式寄存器（MCM）在高速系统时钟（X1 时钟或者外部主系统时钟）和高速内部振荡时钟间切换。

(2) 低速内部振荡时钟（看门狗定时器时钟）

• 低速内部振荡电路

产生 $f_{RL}=240\text{kHz(TYP)}$ 的振荡时钟。在复位解除后，必须开始低速内部振荡时钟的运行。

在通过选项字节设置为“可通过软件停止低速内部振荡器”时，可通过设置内部振荡模式寄存器（RCM）停止时钟振荡。

低速内部振荡时钟不能用作 CPU 时钟。以低速内部振荡时钟运行的硬件如下所示。

- 看门狗定时器
- TMH1（选择 f_{RL} 、 $f_{RL}/2^7$ 或者 $f_{RL}/2^9$ 时）

备注 f_X : X1 时钟振荡频率
 f_{RH} : 高速内部振荡时钟频率
 f_{EXCLK} : 外部主系统时钟频率
 f_{RL} : 低速内部振荡时钟频率

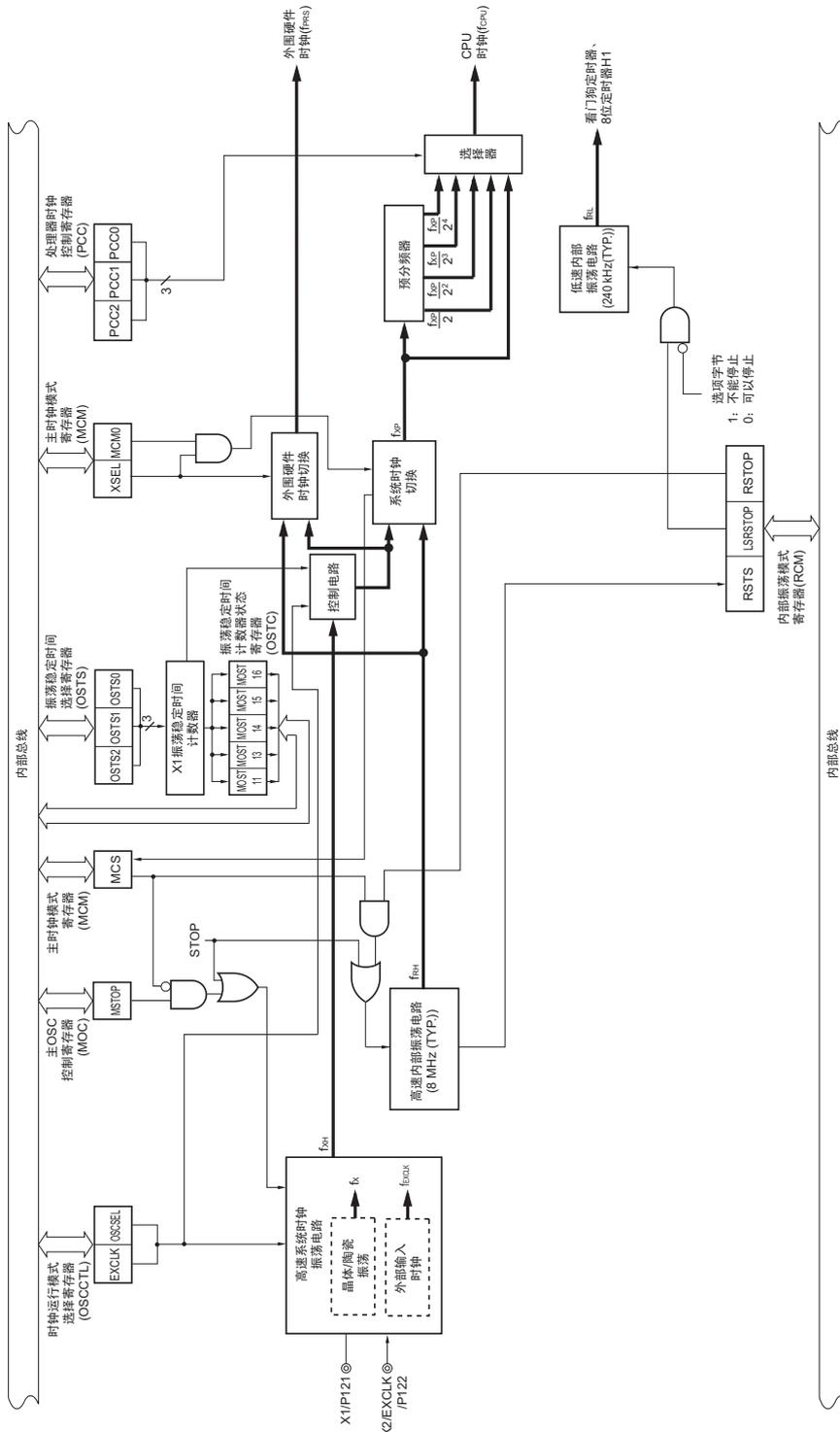
5.2 时钟发生电路的结构

时钟发生电路由以下硬件构成。

表 5-1 时钟发生电路的结构

项目	结构
控制寄存器	时钟运行模式选择寄存器 (OSCCTL) 处理器时钟控制寄存器 (PCC) 内部振荡模式寄存器 (RCM) 主 OSC 控制寄存器 (MOC) 主时钟模式寄存器 (MCM) 振荡稳定时间计数器状态寄存器 (OSTC) 振荡稳定时间选择寄存器 (OSTS)
振荡电路	X1 振荡电路 高速内部振荡电路 低速内部振荡电路

图 5-1 时钟发生电路的框图



- 备注
- f_X : X1 时钟振荡频率
 - f_{RH} : 高速内部振荡时钟频率
 - f_{EXCLK} : 外部主系统时钟频率
 - f_{XH} : 高速系统时钟频率
 - f_{XP} : 主系统时钟频率
 - f_{PR} : 外围硬件时钟频率
 - f_{CPU} : CPU 时钟频率
 - f_{RL} : 低速内部振荡时钟频率

5.3 时钟发生电路的控制寄存器

通过以下 7 种寄存器控制时钟发生电路。

- 时钟运行模式选择寄存器（OSCCTL）
- 处理器时钟控制寄存器（PCC）
- 内部振荡模式寄存器（RCM）
- 主 OSC 控制寄存器（MOC）
- 主时钟模式寄存器（MCM）
- 振荡稳定时间计数器状态寄存器（OSTC）
- 振荡稳定时间选择寄存器（OSTS）

(1) 时钟运行模式选择寄存器（OSCCTL）

时钟运行模式选择寄存器选择高速系统时钟的运行模式。

通过 1 位或 8 位存储器操作指令设置 OSCCTL。

在产生复位信号后，OSCCTL 变为“00H”。

图 5-2 时钟运行模式选择寄存器（OSCCTL）的格式

地址：FF9FH 复位后：00H R/W

符号	<7>	<6>	5	4	3	2	1	0
OSCCTL	EXCLK	OSCSEL	0	0	0	0	0	0

EXCLK	OSCSEL	高速系统时钟引脚的运行模式	P121/X1 引脚	P122/X2/EXCLK 引脚
0	0	输入 / 输出端口模式	输入 / 输出端口	
0	1	X1 振荡模式	连接晶体 / 陶瓷谐振器	
1	0	输入 / 输出端口模式	输入 / 输出端口	
1	1	外部时钟输入模式	输入 / 输出端口	外部时钟输入

注意 1. 如果要改写 EXCLK 和 OSCSEL，必须确认主 OSC 控制寄存器（MOC）的 bit7（MSTOP）为“1”（X1 振荡电路停止或者 EXCLK 引脚的外部时钟无效）。

2. 必须将 bit0 ~ bit5 置“0”。

备注 f_{XH} ：高速系统时钟频率

(2) 处理器时钟控制寄存器 (PCC)

处理器时钟控制寄存器进行 CPU 时钟的选择和分频比的设置。

通过 1 位或 8 位存储器操作指令设置 PCC。

在产生复位信号后，PCC 变为“01H”。

图 5-3 处理器时钟控制寄存器 (PCC) 的格式

地址: FFFBH 复位后: 01H R/W

符号	7	6	5	4	3	2	1	0
PCC	0	0	0	0	0	PCC2	PCC1	PCC0

PCC2	PCC1	PCC0	CPU 时钟 (f_{CPU}) 的选择
0	0	0	f_{XP}
0	0	1	$f_{XP}/2$ (默认)
0	1	0	$f_{XP}/2^2$
0	1	1	$f_{XP}/2^3$
1	0	0	$f_{XP}/2^4$
上述以外			禁止设置

注意 1. 必须将 bit3 ~ bit7 置“0”。

2. 设置 PCC 的分频比时，外围硬件时钟 (f_{PRS}) 不被分频。

备注 f_{XP} : 主系统时钟频率

在 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 中最快的指令执行时间为 2 个 CPU 时钟。

CPU 时钟 (f_{CPU}) 与最短指令执行时间的关系如表 5-2 所示。

表 5-2 CPU 时钟与最短指令执行时间的关系

CPU 时钟 (f_{CPU})	最短指令执行时间: $2/f_{CPU}$	
	主系统时钟	
	高速系统时钟注	高速内部振荡时钟注
	以 10MHz 运行时	以 8MHz(TYP.) 运行时
f_{XP}	0.2 μ s	0.25 μ s(TYP.)
$f_{XP}/2$	0.4 μ s	0.5 μ s(TYP.)
$f_{XP}/2^2$	0.8 μ s	1.0 μ s(TYP.)
$f_{XP}/2^3$	1.6 μ s	2.0 μ s(TYP.)
$f_{XP}/2^4$	3.2 μ s	4.0 μ s(TYP.)

注 通过主时钟模式寄存器 (MCM) 设置作为 CPU 时钟提供的主系统时钟 (高速系统时钟 / 高速内部振荡时钟) (参照图 5-6)。

(3) 内部振荡模式寄存器 (RCM)

内部振荡模式寄存器设置内部振荡器的运行模式。

通过 1 位或 8 位存储器操作指令设置 RCM。

在产生复位信号后，RCM 变为“80H”注 1。

图 5-4 内部振荡模式寄存器 (RCM) 的格式

地址：FFA0H 复位后：80H 注 1 R/W 注 2

符号	<7>	6	5	4	3	2	<1>	<0>
RCM	RSTS	0	0	0	0	0	LSRSTOP	RSTOP

RSTS	高速内部振荡器的状态
0	等待内部高速振荡器的振荡精度稳定
1	高速内部振荡器稳定运行

LSRSTOP	低速内部振荡器的振荡 / 停止
0	低速内部振荡器振荡
1	低速内部振荡器停止

RSTOP	高速内部振荡器振荡 / 停止
0	高速内部振荡器振荡
1	高速内部振荡器停止

- 注 1. 复位解除后的值为“00H”，但是，在等到高速内部振荡器的振荡精度稳定后，自动切换为“80H”。
2. bit7 为只读位。

注意 如果要将在 RSTOP 置“1”，必须确认 CPU 时钟是以高速内部振荡时钟以外的时钟运行。具体条件如下所示。

- MCS=1 时 (CPU 时钟以高速系统时钟运作)

另外，必须在以高速内部振荡时钟运行的外围硬件停止后，将 RSTOP 置“1”。

(4) 主 OSC 控制寄存器 (MOC)

主 OSC 控制寄存器选择高速系统时钟的运行模式。

在 CPU 以高速系统时钟以外的时钟运行时，该寄存器用来停止 X1 振荡电路或者将 EXCLK 引脚的外部时钟置为无效。

通过 1 位或 8 位存储器操作指令设置 MOC。

在产生复位信号后，MOC 变为“80H”。

图 5-5 主 OSC 控制寄存器 (MOC) 的格式

地址：FFA2H 复位后：80H R/W

符号	<7>	6	5	4	3	2	1	0
MOC	MSTOP	0	0	0	0	0	0	0

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
0	X1 振荡电路运行	EXCLK 引脚的外部时钟有效
1	X1 振荡电路停止	EXCLK 引脚的外部时钟无效

注意 1. 如果要将 MSTOP 置“1”，必须确认 CPU 时钟是以高速系统时钟以外的时钟运行。具体条件如下所示。

- MCS=0 时 (CPU 时钟以高速内部振荡时钟运作)

另外，必须在以高速系统时钟运行的外围硬件停止后，将 MSTOP 置“1”。

2. 在时钟运行模式选择寄存器 (OSCCTL) 的 bit6 (OSCSEL) 为“0” (输入端口模式) 时，不能将 MSTOP 置“0”。
3. 如果外围硬件时钟停止，外围硬件就无法运行。如果要使停止后的外围硬件时钟重新开始振荡，必须对外围硬件进行初始化。

(5) 主时钟模式寄存器 (MCM)

主时钟模式寄存器选择作为 CPU 时钟提供的主系统时钟和作为外围硬件时钟提供的时钟。

通过 1 位或 8 位存储器操作指令设置 MCM。

在产生复位信号后，MCM 变为“00H”。

图 5-6 主时钟模式寄存器 (MCM) 的格式

地址：FFA1H 复位后：00H R/W注

符号	7	6	5	4	3	<2>	<1>	<0>
MCM	0	0	0	0	0	XSEL	MCS	MCM0

XSEL	MCM0	选择主系统时钟和提供给外围硬件的时钟	
		主系统时钟 (f_{XP})	外围硬件时钟 (f_{PRS})
0	0	高速内部振荡时钟 (f_{RH})	高速内部振荡时钟 (f_{RH})
0	1		高速系统时钟 (f_{XH})
1	0		
1	1	高速系统时钟 (f_{XH})	

MCS	主系统时钟的状态
0	以高速内部振荡时钟运行
1	以高速系统时钟运行

注 bit1 位为只读位。

注意 1. 复位解除后，只能进行 1 次 XSEL 设置。

2. 与 XSEL 和 MCM0 的设置无关，对以下外围功能提供 f_{PRS} 以外的时钟。

- 看门狗定时器（以低速内部振荡时钟运行）
- 选择“ f_{RL} ”、“ $f_{RL}/2^7$ ”或“ $f_{RL}/2^9$ ”作为 8 位定时器 H1 的计数时钟时（以低速内部振荡时钟运行）
- 选择外部时钟作为时钟源的外围硬件
（但是，选择 TM00 的外部计数时钟时（TI000 引脚的有效边沿）除外。）

(6) 振荡稳定时间计数器状态寄存器 (OSTC)

振荡稳定时间计数器状态寄存器显示 X1 时钟振的荡稳定时间计数器的计数状态。在 CPU 时钟为高速内部振荡时钟并且 X1 时钟开始振荡时，可确认 X1 时钟的振荡稳定时间。

通过 1 位或 8 位存储器操作指令设置 OSTC。

在产生复位信号后（因 RESET 输入、POC、LVI 和 WDT 引起的复位），执行 STOP 指令或者将 MSTOP (MOC 寄存器的 bit7) 置“1”，OSTC 变为“00H”。

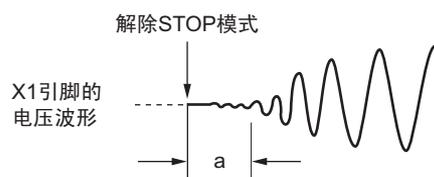
图 5-7 振荡稳定时间计数器状态寄存器 (OSTC) 的格式

地址：FFA3H 复位后：00H R

符号	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	振荡稳定时间的状态	
					$\geq 2^n/f_X$	$f_X=10\text{MHz}$
1	0	0	0	0	$\geq 2^{11}/f_X$	$\geq 204.8\mu\text{s}$
1	1	0	0	0	$\geq 2^{13}/f_X$	$\geq 819.2\mu\text{s}$
1	1	1	0	0	$\geq 2^{14}/f_X$	$\geq 1.64\text{ms}$
1	1	1	1	0	$\geq 2^{15}/f_X$	$\geq 3.27\text{ms}$
1	1	1	1	1	$\geq 2^{16}/f_X$	$\geq 6.55\text{ms}$

- 注意 1. 在经过上述时间后，从 MOST11 开始逐次置“1”，并且一直保持“1”。
2. 振荡稳定时间计数器只能在 OSTC 设置的振荡稳定时间内进行计数。在 CPU 时钟为高速内部振荡时钟时，进入 STOP 模式。如果要解除该模式，必须如下设置 OSTC 的振荡稳定时间。
- 期待的 OSTC 的振荡稳定时间 \leq 通过 OSTC 设置的振荡稳定时间
- 因此，必须注意，解除 STOP 模式后的 OSTC 的设置状态，只能为 OSTC 设置的振荡稳定时间内的状态。
3. X1 时钟的振荡稳定时间不包括开始时钟振荡前的时间（下图中“a”所表示的部分）。



备注 f_X : X1 时钟振荡频率

(7) 振荡稳定时间选择寄存器 (OSTS)

振荡稳定时间选择寄存器选择解除 STOP 模式时的 X1 时钟的振荡稳定时间。

如果选择 X1 时钟作为 CPU 时钟，就在解除 STOP 模式后，等待 OSTS 设置的时间。

如果选择高速内部振荡时钟作为 CPU 时钟，就在解除 STOP 模式后，通过 OSTC 确认是否经过振荡稳定时间。OSTC 可在事先通过 OSTS 设置的时间内进行确认。

通过 8 位存储器操作指令设置 OSTS。

在产生复位信号后，OSTS 变为“05H”。

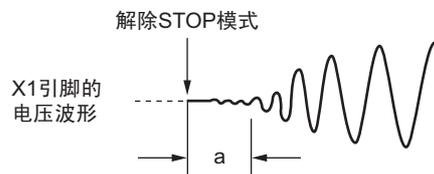
图 5-8 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFA4H 复位后: 05H 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	
				$f_X=10\text{MHz}$
0	0	1	$2^{11}/f_X$	204.8 μs
0	1	0	$2^{13}/f_X$	819.2 μs
0	1	1	$2^{14}/f_X$	1.64ms
1	0	0	$2^{15}/f_X$	3.27ms
1	0	1	$2^{16}/f_X$	6.55ms
上述以外			禁止设置	

- 注意 1. 如果要在 CPU 时钟为 X1 时钟时转移到 STOP 模式，必须在执行 STOP 指令前设置 OSTS。
2. 不能在 X1 时钟的振荡稳定时间内更改 OSTS 寄存器。
3. 振荡稳定时间计数器只能在 OSTS 设置的振荡稳定时间内进行计数。在 CPU 时钟为高速内部振荡时钟时，进入 STOP 模式。如果要解除该模式，必须如下设置 OSTS 的振荡稳定时间。
- 期待的 OSTC 的振荡稳定时间 \leq 通过 OSTS 设置的振荡稳定时间
- 因此，必须注意，解除 STOP 模式后的 OSTC 的设置状态，只能为 OSTS 设置的振荡稳定时间内的状态。
4. X1 时钟的振荡稳定时间不包括开始时钟振荡前的时间（下图中“a”所表示的部分）。



备注 f_X : X1 时钟振荡频率

5.4 系统时钟振荡电路

5.4.1 X1 振荡电路

X1 振荡电路通过连接到 X1 和 X2 引脚的晶体谐振器或者陶瓷谐振器（1 ~ 10MHz）振荡。也可输入外部时钟。此时，向 EXCLK 引脚输入时钟信号。

X1 振荡电路的外接电路示例如图 5-9 所示。

图 5-9 X1 振荡电路的外接电路示例

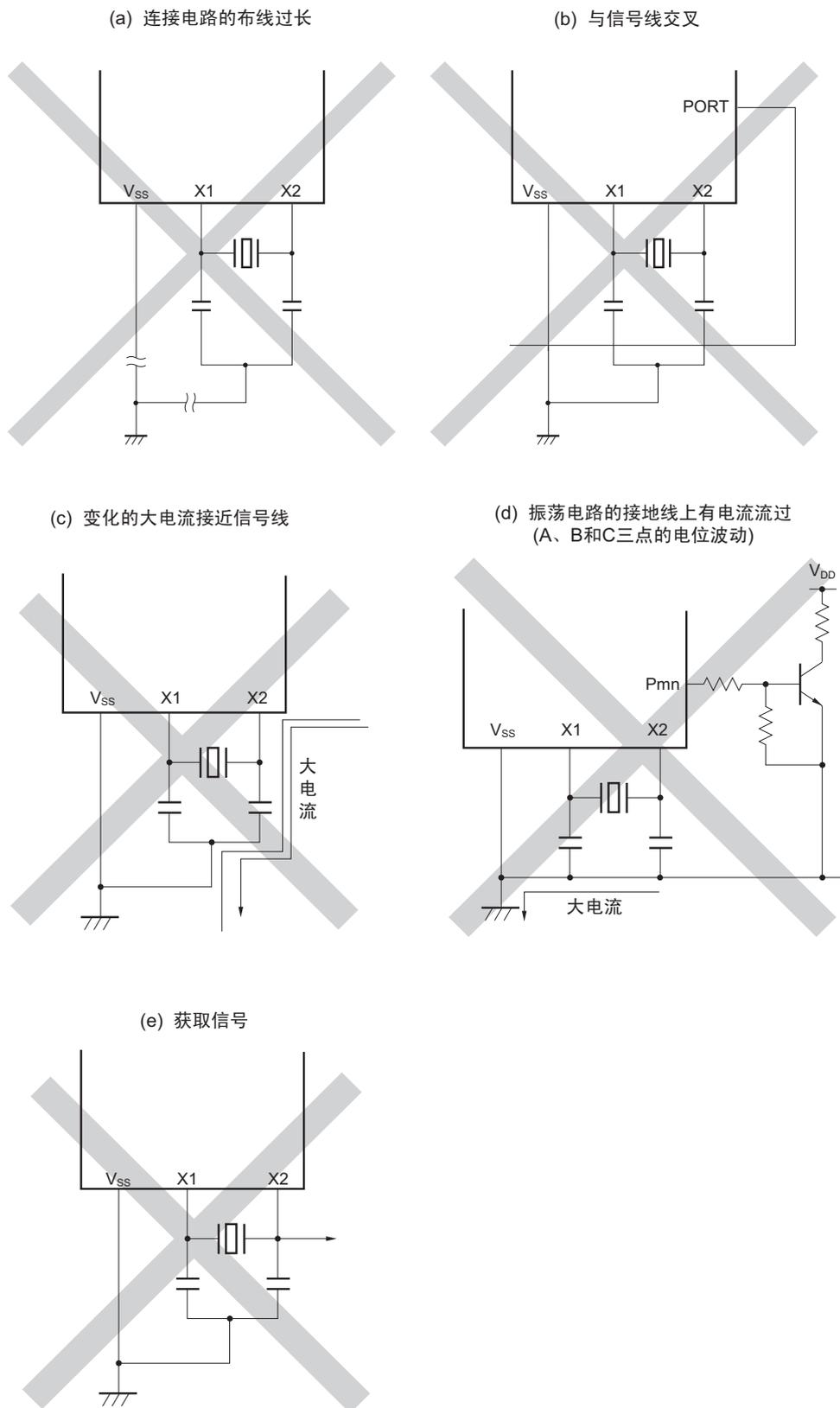


注意 在使用 X1 振荡电路时，对图 5-9 中虚线的部分进行如下布线，从而避免布线电容等的影响。

- 布线要尽量短。
- 不与其他信号线交叉，并且远离会流过发生变化的大电流的线路。
- 振荡电路的电容器接地点必须总是与 V_{SS} 保持相同电位，并且不能与流过大电流的接地图案进行接地。
- 不能从振荡电路获取信号。

谐振器的错误连接示例如图 5-10 所示。

图 5-10 谐振器的错误连接示例



5.4.2 高速内部振荡电路

R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 内置高速内部振荡电路。由内部振荡模式寄存器 (RCM) 控制高速内部振荡电路的振荡。

复位解除后，高速内部振荡电路自动开始振荡。

5.4.3 低速内部振荡电路

R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 内置低速内部振荡电路。

低速内部振荡时钟只能用作看门狗定时器和 8 位定时器 H1 的时钟，不能用作 CPU 时钟。

可通过选项字节选择“可通过软件停止”或者“不可停止”。在选择“可通过软件停止”时，可由内部振荡模式寄存器 (RCM) 控制低速内部振荡电路的振荡。

复位解除后，低速内部振荡电路自动开始振荡，如果通过选项字节设置为“允许看门狗定时器运行”，就启动看门狗定时器 (240 kHz(TYP.))。

5.4.4 预分频器

在选择主系统时钟作为提供给 CPU 的时钟时，预分频器对主系统时钟进行分频，并且生成时钟。

5.5 时钟发生电路的运行

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

- 主系统时钟 f_{XP}
 - 高速系统时钟 f_{XH}
 - X1 时钟 f_X
 - 外部主系统时钟 f_{EXCLK}
 - 高速内部振荡时钟 f_{RH}
- 低速内部振荡时钟 f_{RL}
- CPU 时钟 f_{CPU}
- 外围硬件时钟 f_{PRS}

在 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 中，CPU 在复位解除后通过高速内部振荡电路的输出开始运行，并且具有以下特点。

(1) 强化安全功能

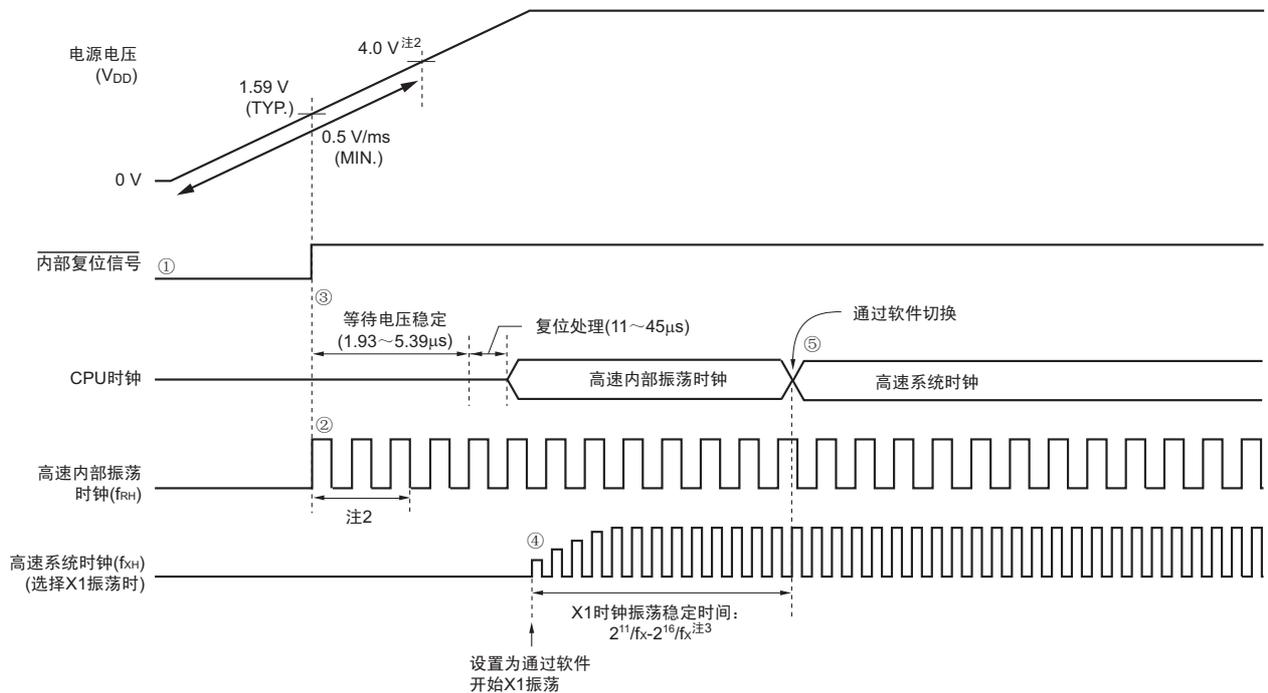
在复位解除后因损坏或连接不良等原因导致 X1 时钟不运行时，如果默认 CPU 时钟为 X1 时钟，设备就从此时刻开始无法运行；但如果 CPU 的启动时钟为高速内部振荡时钟，设备就可在复位解除后通过高速内部振荡时钟启动。因此，可通过如由软件识别复位源、执行异常时的安全处理等最低限度的运行，安全结束系统。

(2) 提高性能

因为可以无需等待 X1 时钟的振荡稳定时间就能启动 CPU，所以实现提高总性能。

输入电源电压时的时钟发生电路的运行如图 5-11 和图 5-12 所示。

图 5-11 输入电源电压时的时钟发生电路的运行
(设置为“1.59V POC 模式”的情况 (选项字节: POCMODE=0))



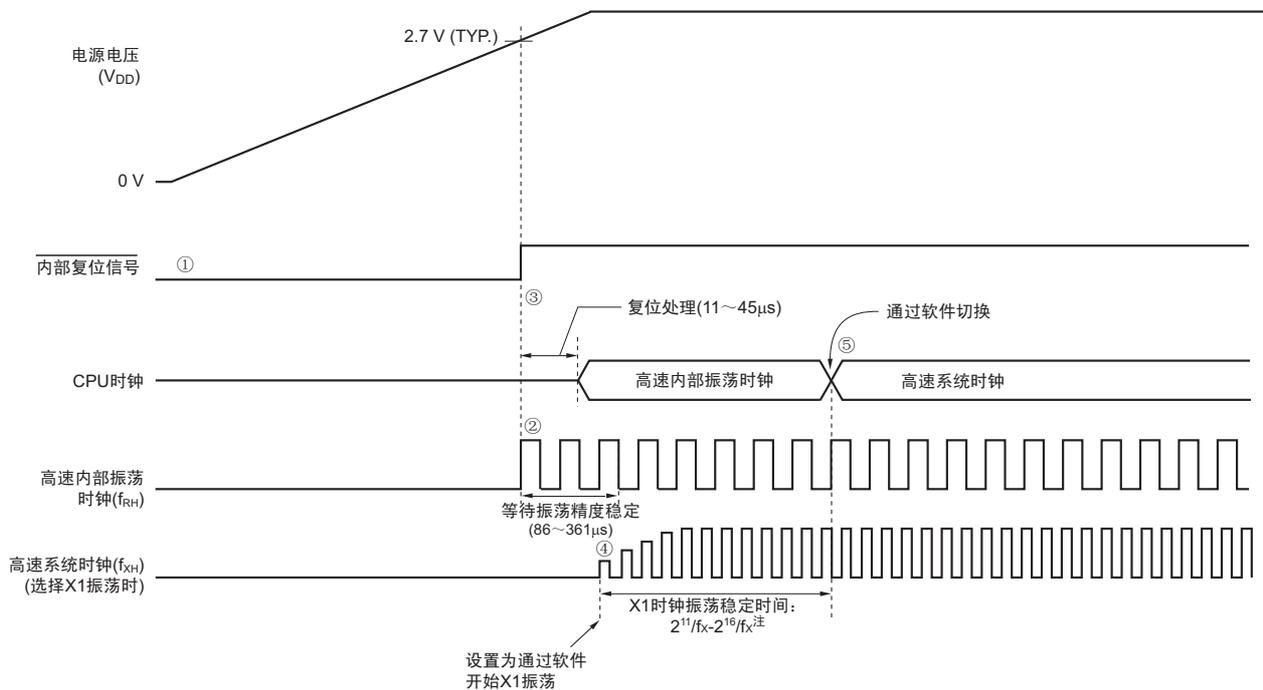
- ① 接通电源后，通过上电清除（POC）电路产生内部复位信号。
- ② 如果电源电压超过 1.59 V(TYP.)，就解除复位，并且高速内部振荡器自动开始振荡。
- ③ 如果电源电压以 0.5V/ms 的斜率上升，就在复位解除并且经过电源 / 稳压器的电压稳定等待时间后，执行复位处理，之后 CPU 以高速内部振荡时钟开始运行。
- ④ 设置为通过软件开始 X1 时钟振荡（参照“5.6.1 高速系统时钟的控制示例”的(1)）。
- ⑤ 如果要将在 CPU 时钟切换为 X1 时钟，必须在等待时钟振荡稳定后，通过软件设置切换（参照“5.6.1 高速系统时钟的控制示例”的(3)）。

- 注 1. 如果从接通电源到达到 4.0V 时的电压的上升斜率低于 0.5V/ms(MIN.)，就在接通电源到达到 4.0V 期间，输入低电平到 $\overline{\text{RESET}}$ 引脚，或者通过选项字节设置为“2.7V/1.59V POC 模式”（POCMODE=1）（参照图 5-12）。在输入低电平到 $\overline{\text{RESET}}$ 引脚直到电压达到 4.0V 时，解除 $\overline{\text{RESET}}$ 引脚复位后的运行时序，与图 5-11 的②以后的时序相同。
2. 内部电压稳定时间包括高速内部振荡时钟的振荡精度稳定等待时间。
 3. 在复位解除（如上图所示）以及解除 CPU 时钟为高速内部振荡时钟时的 STOP 模式时，必须通过振荡稳定时间计数器状态寄存器（OSTC）确认 X1 时钟的振荡稳定时间。如果 CPU 时钟为高速系统时钟（X1 振荡），就通过振荡稳定时间选择寄存器（OSTS）设置解除 STOP 模式时的振荡稳定时间。

注意 使用 EXCLK 引脚的外部时钟输入时，无需振荡稳定等待时间。

备注 在单片机运行时，可通过软件设置停止未作为 CPU 时钟使用的时钟。可通过执行 STOP 指令，停止高速内部振荡时钟和高速系统时钟（参照“5.6.1 高速系统时钟的控制示例”的(4)和“5.6.2 高速内部振荡时钟的控制示例”的(3)）。

图 5-12 输入电源电压时的时钟发生电路的运行
(设置为“2.7V/1.59V POC 模式”的情况 (选项字节: POCMODE=1))



- ① 接通电源后，通过上电清除（POC）电路产生内部复位信号。
- ② 如果电源电压超过 2.7 V(TYP.)，就解除复位，并且高速内部振荡器自动开始振荡。
- ③ 复位解除后，CPU 在执行复位处理后以高速内部振荡时钟开始运行。
- ④ 设置为通过软件开始 X1 时钟振荡（参照“5.6.1 高速系统时钟的控制示例”的(1)）。
- ⑤ 如果要将 CPU 时钟切换为 X1 时钟，必须在等待时钟振荡稳定后，通过软件设置切换（参阅“5.6.1 高速系统时钟的控制示例”的(3)）。

注 在复位解除（如上图所示）以及解除 CPU 时钟为高速内部振荡时钟时的 STOP 模式时，必须通过振荡稳定时间计数器状态寄存器（OSTC）确认 X1 时钟的振荡稳定时间。如果 CPU 时钟为高速系统时钟（X1 振荡），就通过振荡稳定时间选择寄存器（OSTS）设置解除 STOP 模式时的振荡稳定时间。

- 注意 1. 在电源电压达到 1.59V(TYP.) 后，需要 1.93 ~ 5.39ms 的电压稳定等待时间。如果电压从 1.59V(TYP.) 到达到 2.7V(TYP.) 的时间在 1.93ms 以内，就在复位处理前自动产生 0 ~ 5.39ms 的电源稳定等待时间。
2. 使用 EXCLK 引脚的外部时钟输入时，无需振荡稳定等待时间。

备注 在单片机运行时，可通过软件设置停止未作为 CPU 时钟使用的时钟。可通过执行 STOP 指令，停止高速内部振荡时钟和高速系统时钟（参照“5.6.1 高速系统时钟的控制示例”的(4)和“5.6.2 高速内部振荡时钟的控制示例”的(3)）。

5.6 时钟的控制

5.6.1 高速系统时钟的控制示例

高速系统时钟有以下两种。

- X1 时钟：给 X1、X2 引脚连接晶体 / 陶瓷谐振器
- 外部主系统时钟：给 EXCLK 引脚输入外部时钟

不使用时，X1/P121、X2/EXCLK/P122 引脚可用作输入 / 输出端口。

注意 复位解除时的 X1/P121 和 X2/EXCLK/P122 引脚为输入 / 输出端口模式。

下列情况的设置步骤示例如下所述。

- (1) X1 时钟振荡的情况
- (2) 使用外部主系统时钟的情况
- (3) 高速系统时钟用作 CPU 时钟和外围硬件时钟的情况
- (4) 停止高速系统时钟的情况

(1) X1 时钟振荡时的设置步骤示例

- ① P121/X1、P122/X2/EXCLK 引脚的设置，运行模式的选择（OSCCTL 寄存器）
如果将 EXCLK 置“0”、OSCSEL 置“1”，就从端口模式切换到 X1 振荡模式。

EXCLK	OSCSEL	高速系统时钟引脚的运行模式	P121/X1 引脚	P122/X2/EXCLK 引脚
0	1	X1 振荡模式	连接晶体 / 陶瓷谐振器	

- ② 控制 X1 时钟的振荡（MOC 寄存器）
如果将 MSTOP 置“0”，X1 振荡电路就开始振荡。
- ③ 等待 X1 时钟的振荡稳定
确认 OSTC 寄存器，并且等待所需的时间。
在等待过程中，可通过高速内部振荡时钟执行其他软件处理。

注意 在 X1 时钟运行时，不能改写 EXCLK 和 OSCSEL。

(2) 使用外部主系统时钟时的设置步骤示例

- ① P121/X1、P122/X2/EXCLK 引脚的设置，运行模式的选择（OSCCTL 寄存器）
如果将 EXCLK 和 OSCSEL 都置“1”，就从端口模式切换到外部时钟输入模式。

EXCLK	OSCSEL	高速系统时钟引脚的操作模式	P121/X1 引脚	P122/X2/EXCLK 引脚
1	1	外部时钟输入模式	输入 / 输出端口	外部时钟输入

- ② 控制外部主系统时钟的输入（MOC 寄存器）
如果将 MSTOP 置“0”，外部主系统时钟的输入就有效。

注意 在外部主系统时钟运行时，不能改写 EXCLK 和 OSCSEL。

(3) 高速系统时钟用作 CPU 时钟和外围硬件时钟时的设置步骤示例

① 设置高速系统时钟振荡注

(参照“5.6.1 (1) X1时钟振荡时的设置步骤示例”和“(2) 使用外部主系统时钟时的设置步骤示例”。)

注 在高速系统时钟运行时，不需要步骤①。

② 设置高速系统时钟为主系统时钟 (MCM 寄存器)

如果将 XSEL 和 MCM0 都置“1”，高速系统时钟就用作主系统时钟和外围硬件时钟。

XSEL	MCM0	选择主系统时钟和外围硬件时钟	
		主系统时钟 (f_{XP})	外围硬件时钟 (f_{PRS})
1	1	高速系统时钟 (f_{XH})	高速系统时钟 (f_{XH})

注意 如果选择高速系统时钟作为主系统时钟，就不能将高速系统时钟以外的时钟设置为外围硬件时钟。

③ 分频比的选择 (PCC 寄存器)

可通过 PCC0、PCC1 和 PCC2 选择 CPU 时钟的分频比。

PCC2	PCC1	PCC0	选择 CPU 时钟 (f_{CPU})
0	0	0	f_{XP}
0	0	1	$f_{XP}/2$ (默认)
0	1	0	$f_{XP}/2^2$
0	1	1	$f_{XP}/2^3$
1	0	0	$f_{XP}/2^4$
上述以外			禁止设置

(4) 停止高速系统时钟时的设置步骤示例

可以使用以下两种方法，停止高速系统时钟。

- 执行 STOP 指令，停止 X1 振荡 (如果使用外部时钟，则时钟输入无效)。
- 将 MSTOP 置“1”，停止 X1 振荡 (如果使用外部时钟，则时钟输入无效)。

(a) 执行 STOP 指令的情况

① 外围硬件的停止设置

停止 STOP 模式中不能使用的的所有外围硬件 (有关 STOP 模式中不能使用的外围硬件，请参照“第 17 章 待机功能”。)。

② 设置待机解除后的 X1 时钟振荡稳定时间。

CPU 以 X1 时钟运行时，在执行 STOP 指令前设置 OSTS 寄存器的值。

③ 执行 STOP 指令

如果执行 STOP 指令，就转移到 STOP 模式，并且停止 X1 振荡 (外部时钟输入无效)。

(b) 通过将 MSTOP 置“1”停止 X1 振荡（外部时钟输入无效）的情况。

① 确认 CPU 时钟的状态（MCM 寄存器）

通过 MCS 确认 CPU 时钟是否以高速系统时钟以外的时钟运行。

在 MCS=1 时，提供高速系统时钟到 CPU，因此必须将 CPU 时钟更改为高速系统时钟以外的时钟。

MCS	CPU 时钟的状态
0	高速内部振荡时钟
1	高速系统时钟

② 停止高速系统时钟（MOC 寄存器）

如果将 MSTOP 置“1”，就停止 X1 振荡（外部时钟输入无效）。

注意 在将 MSTOP 置“1”时，必须确认 MCS=0。此外，必须停止以高速系统时钟运行的外围硬件。

5.6.2 高速内部振荡时钟的控制示例

下列情况的设置步骤示例如下所述。

- (1) 重新开始高速内部振荡时钟振荡的情况。
- (2) 高速内部振荡时钟用作 CPU 时钟，高速内部振荡时钟或者高速系统时钟用作外围硬件时钟的情况
- (3) 停止高速内部振荡时钟的情况

(1) 重新开始高速内部振荡时钟振荡时的设置步骤示例注 1

- ① 重新开始高速内部振荡时钟振荡的设置（RCM 寄存器）
如果将 RSTOP 置“0”，高速内部振荡时钟就重新开始振荡。
- ② 高速内部振荡时钟的振荡精度稳定等待时间（RCM 寄存器）
等到 RSTS 置“1”注 2。

注 1. 复位解除后，高速内部振荡器自动开始振荡。选择高速内部振荡时钟作为 CPU 时钟。
2. 如果 CPU 时钟和外围硬件时钟不需要精度，则可省略等待时间。

(2) 高速内部振荡时钟用作 CPU 时钟，高速内部振荡时钟或者高速系统时钟用作外围硬件时钟时的设置步骤示例

- ①
 - 重新开始高速内部振荡时钟的振荡注。
(参照“5.6.2 (1) 重新开始高速内部振荡时钟振荡时的设置步骤示例”)。
 - 高速系统时钟振荡注
(在高速系统时钟用作外围硬件时钟时，需要进行该设置。)
(参照“5.6.1 (1) X1 时钟振荡时的设置步骤示例”和“(2) 使用外部主系统时钟时的设置步骤示例”。)

注 在高速内部振荡时钟、高速系统时钟运行时，不需要步骤①。

- ② 主系统时钟和外围硬件时钟的选择（MCM 寄存器）
通过 XSEL 和 MCM0 设置主系统时钟和外围硬件时钟。

XSEL	MCM0	选择主系统时钟和外围硬件时钟	
		主系统时钟 (f_{XP})	外围硬件时钟 (f_{PRS})
0	0	高速内部振荡时钟 (f_{RH})	高速内部振荡时钟 (f_{RH})
0	1		
1	0		高速系统时钟 (f_{XH})

- ③ 选择 CPU 时钟的分频比（PCC 寄存器）
通过 PCC0、PCC1 和 PCC2，选择 CPU 时钟的分频比。

PCC2	PCC1	PCC0	CPU 时钟 (f_{CPU}) 选择
0	0	0	f_{XP}
0	0	1	$f_{XP}/2$ (默认)
0	1	0	$f_{XP}/2^2$
0	1	1	$f_{XP}/2^3$
1	0	0	$f_{XP}/2^4$
上述以外			禁止设置

(3) 停止高速内部振荡时钟时的设置步骤示例

可以使用以下两种方法，停止高速内部振荡时钟。

- 执行STOP指令，转移到STOP模式。
- 将RSTOP置“1”，停止高速内部振荡时钟。

(a) 执行STOP指令的情况**① 外围硬件的设置**

停止STOP模式中不能使用的**所有**外围硬件（有关STOP模式中不能使用的外围硬件，请参照“第17章 待机功能”）。。

② 设置待机解除后的X1时钟振荡稳定时间

CPU以X1时钟运行时，在执行STOP指令前设置OSTS寄存器的值。如果要在STOP模式解除后立即进行CPU运行，必须先将MCM0置“0”，再将CPU时钟切换为高速内部振荡时钟，然后确认RSTS为“1”。

③ 执行STOP指令

如果执行STOP指令，就转移到STOP模式，并且停止高速内部振荡时钟。

(b) 通过将RSTOP置“1”停止高速内部振荡时钟的情况**① 确认CPU时钟的状态（MCM寄存器）**

通过MCS确认CPU时钟以高速内部振荡时钟以外的时钟运行。

在MCS=0时，提供高速内部振荡时钟到CPU，因此必须将CPU时钟更改为高速内部振荡时钟以外的时钟。

MCS	CPU 时钟的状态
0	高速内部振荡时钟
1	高速系统时钟

② 停止高速内部振荡时钟（RCM寄存器）

如果将RSTOP置“1”，就停止高速内部振荡时钟。

注意 在将RSTOP置“1”时，必须确认MCS=1。此外，必须停止以高速内部振荡时钟运行的外围硬件。

5.6.3 低速内部振荡时钟的控制示例

低速内部振荡时钟不能用作 CPU 时钟。

该时钟只能运行以下外围硬件。

- 看门狗定时器
- 8 位定时器 H1（选择 f_{RL} 作为计数时钟时）

另外，可通过选项字节选择以下运行模式。

- 不能停止低速内部振荡器的振荡。
- 可通过软件停止低速内部振荡器的振荡。

复位解除后，低速内部振荡器自动开始振荡。如果通过选项字节设置为“允许看门狗定时器运行”，就启动看门狗定时器（240 kHz (TYP.)）。

(1) 停止低速内部振荡时钟时的设置步骤示例

- ① 将 LSRSTOP 置“1”（RCM 寄存器）

如果将 LSRSTOP 置“1”，就停止低速内部振荡时钟的振荡。

(2) 重新开始低速内部振荡时钟振荡的设置步骤示例

- ① 将 LSRSTOP 置“0”（RCM 寄存器）

如果将 LSRSTOP 置“0”，就重新开始低速内部振荡时钟的振荡。

注意 如果通过选项字节设置为“不能停止低速内部振荡器的振荡”，就无法控制低速内部振荡时钟的振荡。

5.6.4 作为 CPU 时钟和外围硬件时钟提供的时钟

作为 CPU 时钟和外围硬件时钟提供的时钟与寄存器的设置如下所示。

表 5-3 作为 CPU 时钟和外围硬件时钟提供的时钟与寄存器的设置

提供的时钟		XSEL	MCM0	EXCLK
作为 CPU 时钟提供的时钟	作为外围硬件时钟提供的时钟			
高速内部振荡时钟		0	x	x
高速内部振荡时钟	X1 时钟	1	0	0
	外部主系统时钟	1	0	1
X1 时钟		1	1	0
外部主系统时钟		1	1	1

备注 XSEL：主时钟模式寄存器（MCM）的 bit2

MCM0：MCM 的 bit0

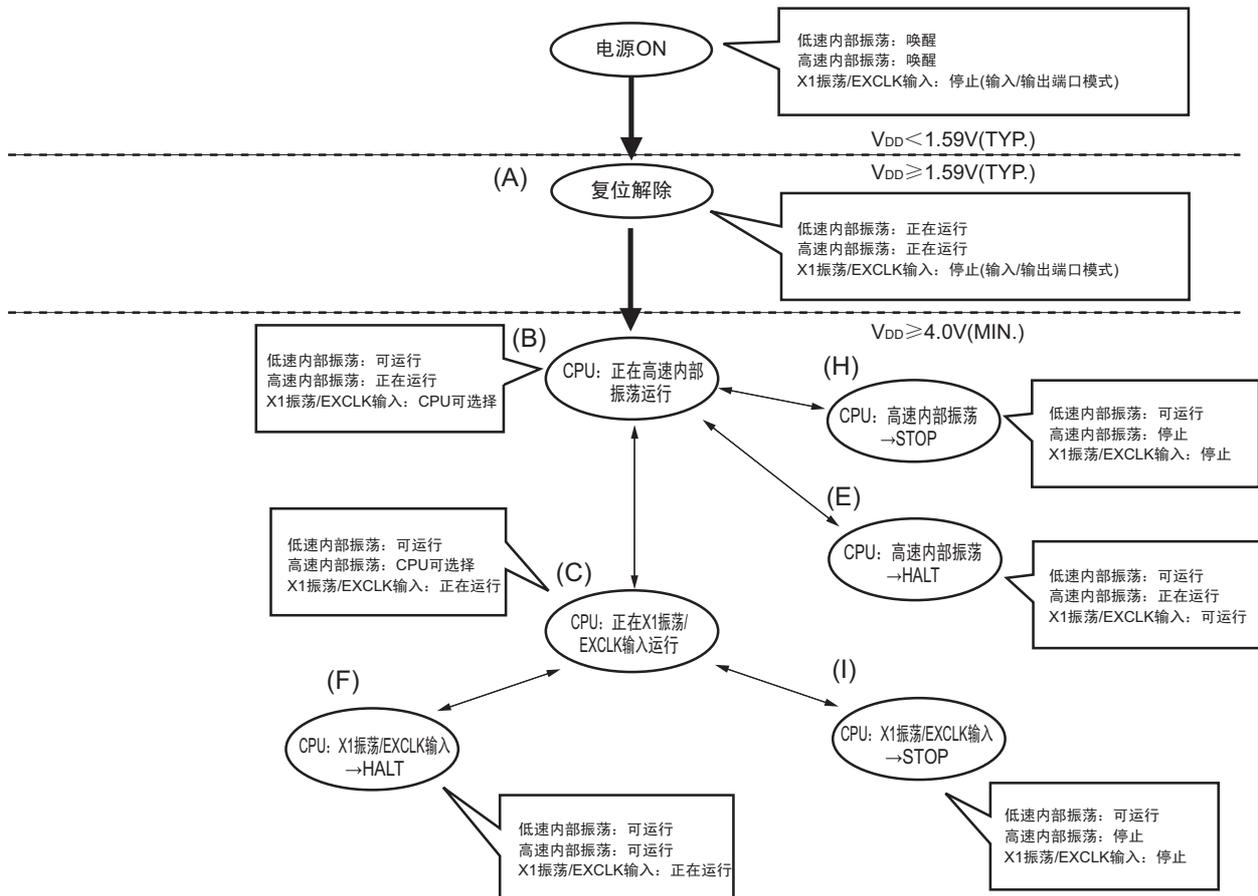
EXCLK：时钟运行模式选择寄存器（OSCCTL）的 bit7

x：忽略

5.6.5 CPU 时钟状态转移图

本产品的 CPU 时钟状态转移图如图 5-13 所示。

图 5-13 CPU 时钟状态转移图
(设置为“1.59V POC 模式”的情况 (选项字节: POCMODE=0))



备注 在设置为“2.7V/1.59V POC 模式”时 (选项字节: POCMODE=1)，接通电源后，如果电源电压超过 2.7V(TYP.)，就转移到上图的 (A)，复位处理后 (11 ~ 45μs)，转移到上图的 (B)。

CPU 时钟的转移和 SFR 寄存器的设置示例如表 5-4 所示。

表 5-4 CPU 时钟的转移和 SFR 寄存器的设置示例 (1/2)

(1) 复位解除后 (A) 向 CPU 以高速内部振荡时钟运行 (B) 转移

状态转换	SFR 寄存器的设置
(A) → (B)	无需设置 SFR 寄存器 (复位解除后的初始状态)

(2) 复位解除后 (A) 向 CPU 以高速系统时钟运行 (C) 转移

(复位解除后, CPU 以高速内部振荡时钟运行 (B)。)

(SFR寄存器的设置顺序) →

状态转换 / SFR寄存器的设置标志	EXCLK	OSCSEL	MSTOP	OSTC 寄存器	XSEL	MCM0
(A)→(B)→(C) (X1时钟: $1\text{MHz} \leq f_{XH} \leq 10\text{MHz}$)	0	1	0	必须确认	1	1
(A)→(B)→(C) (外部主系统时钟: $1\text{MHz} \leq f_{XH} \leq 10\text{MHz}$)	1	1	0	不必确认	1	1

(3) 从 CPU 以高速内部振荡时钟运行 (B) 向 CPU 以高速系统时钟运行 (C) 转移

(SFR寄存器的设置顺序) →

状态转换 / SFR寄存器的设置标志	EXCLK	OSCSEL	MSTOP	OSTC 寄存器	XSEL	MCM0
(B)→(C) (X1时钟: $1\text{MHz} \leq f_{XH} \leq 10\text{MHz}$)	0	1	0	必须确认	1	1
(B)→(C) (外部主系统时钟: $1\text{MHz} \leq f_{XH} \leq 10\text{MHz}$)	1	1	0	不必确认	1	1

如果已设置,
就不需要再设置。

如果以高速系统时钟运行,
就不需要设置。

备注 1. 表 5-4 中的 (A) ~ (I) 对应图 5-13 的 (A) ~ (I)。

2. EXCLK、OSCSEL: 时钟运行模式选择寄存器 (OSCCTL) 的 bit7 和 bit6

MSTOP: 主 OSC 控制寄存器 (MOC) 的 bit7

XSEL、MCM0: 主时钟模式寄存器 (MCM) 的 bit2 和 bit0

X: 忽略

表 5-4 CPU 时钟的转移和 SFR 寄存器的设置示例 (2/2)

(4) 从 CPU 以高速系统时钟运行 (C) 向 CPU 以高速内部振荡时钟运行 (B) 转移

(SFR寄存器的设置顺序) →

状态转移	SFR寄存器的设置标志	RSTOP	RSTS	MCM0
(C)→(B)		0	确认该标志为“1”。	0

如果正在以高速内部振荡时钟运行,就不需要设置。

- (5)
- 从 CPU 正在以高速内部振荡时钟运行 (B) 向 HALT 模式 (E) 转移
 - 从 CPU 正在以高速系统时钟运行 (C) 向 HALT 模式 (F) 转移

状态转移	设置内容
(B)→(E)	执行 HALT 指令
(C)→(F)	

- (6)
- 从 CPU 正在以高速内部振荡时钟运行的 (B) 向 STOP 模式 (H) 转移
 - 从 CPU 正在以高速系统时钟运行的 (C) 向 STOP 模式 (I) 转移

(设置顺序) →

状态转移	设置内容	
(B)→(H) (C)→(I)	停止在STOP模式中不能运行的 的外围功能	执行STOP指令

备注 1. 表 5-4 中的 (A) ~ (I) 对应图 5-13 的 (A) ~ (I)。

2. MCM0: 主时钟模式寄存器 (MCM) 的 bit0
 RSTS、RSTOP: 内部振荡模式寄存器 (RCM) bit7 和 bit0
 X: 忽略

5.6.6 CPU 时钟的转移前条件和转移后处理

CPU 时钟的转移前条件和转移后处理如下所示。

表 5-5 CPU 时钟的转移

CPU 时钟		转移前的条件	转移后的处理
转移前	转移后		
高速内部振荡时钟	X1 时钟	X1 振荡稳定 • MSTOP=0、OSCSEL=1、EXCLK=0 • 经过振荡稳定时间后	可停止高速内部振荡器 (RSTOP=1)
	外部主系统时钟	EXCLK 引脚的外部时钟输入有效 • MSTOP=0、OSCSEL=1、EXCLK=1	可停止高速内部振荡器 (RSTOP=1)
X1 时钟	高速内部振荡	高速内部振荡器振荡 • RSTOP=0	可停止 X1 振荡 (MSTOP=1)
外部主系统时钟	时钟		可设置为外部主系统时钟输入无效 (MSTOP=1)

5.6.7 CPU 时钟的切换和主系统时钟的切换所需时间

可通过设置处理器时钟控制寄存器 (PCC) 的 bit0 ~ bit2 (PCC0 ~ PCC2)，更改主系统时钟的分频比。

实际的切换运行并不是在改写 PCC 后立即就以切换后的时钟运行，而是在改写 PCC 后的数个时钟期间仍以切换前的时钟运行 (参照表 5-6)。

表 5-6 CPU 时钟的切换和更改主系统时钟分频比所需的最长时间

切换前的设置值			切换后的设置值														
PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0
0	0	0	0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0	0	16 个时钟			16 个时钟			16 个时钟			16 个时钟					
0	0	1	8 个时钟			8 个时钟			8 个时钟			8 个时钟					
0	1	0	4 个时钟			4 个时钟			4 个时钟			4 个时钟					
0	1	1	2 个时钟			2 个时钟			2 个时钟			2 个时钟					
1	0	0	1 个时钟			1 个时钟			1 个时钟			1 个时钟					

备注 表 5-6 中的时钟数为切换前的 CPU 时钟数。

可通过设置主时钟模式寄存器 (MCM) 的 bit0 (MCM0)，切换主系统时钟 (在高速内部振荡时钟和高速系统时钟之间进行切换)。

实际的切换运行并不是在改写 MCM0 后立即就以切换后的时钟运行，而是在改写 MCM0 后的数个时钟期间仍以切换前的时钟运行 (参照表 5-7)。

可通过 MCM 的 bit1 确认 CPU 时钟是以高速内部振荡时钟运行还是以高速系统时钟运行。

表 5-7 切换主系统时钟所需的最长时间

切换前的设置值	切换后的设置值	
MCM0	MCM0	
	0	1
0		$1 + 2f_{RH}/f_{XH}$ 个时钟
1	$1 + 2f_{XH}/f_{RH}$ 个时钟	

注意 从高速内部振荡时钟切换到高速系统时钟时，必须事先将 MCM 的 bit2 (XSEL) 置“1”。复位解除后，只能设置 1 次 XSEL。

备注 1. 表 5-7 中的时钟数为切换前的主系统时钟数。

2. 表 5-7 中的时钟数为舍去小数点以后的时钟数。

示例 主系统时钟从高速内部振荡时钟切换到高速系统时钟的情况 (以 $f_{RH}=8\text{MHz}$ 、 $f_{XH}=10\text{MHz}$ 振荡时)

$$1 + 2f_{RH}/f_{XH} = 1 + 2 \times 8/10 = 1 + 2 \times 0.8 = 1 + 1.6 = 2.6 \rightarrow 2 \text{ 个时钟}$$

5.6.8 时钟振荡停止前的条件

停止时钟振荡 (外部时钟输入无效) 的寄存器的标志设置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置

时钟	时钟振荡停止 (外部时钟输入无效) 前的条件	SFR 寄存器的标志设置
高速内部振荡时钟	MCS=1 (CPU 时钟以高速系统时钟运行)	RSTOP=1
X1 时钟	MCS=0	MSTOP=1
外部主系统时钟	(CPU 时钟以高速内部振荡时钟运行)	

5.6.9 外围硬件和源时钟

R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 内置的外围硬件和源时钟如下所示。

表 5-9 外围硬件和源时钟

外围硬件	源时钟	外围硬件时钟 (f _{PRS})	低速内部振荡时钟 (f _{RL})	TM50 输出	外围硬件引脚的 外部时钟
16 位定时器 / 事件计数器 00	00	○	x	x	○ (TI000 引脚) 注
8 位定时器 / 事件计数器	50	○	x	x	○ (TI50 引脚) 注
	51	○	x	x	○ (TI51 引脚) 注
8 位定时器	H0	○	x	○	x
	H1	○	○	x	x
时钟定时器		○	x	x	x
看门狗定时器		x	○	x	x
A/D 转换器		○	x	x	x
串行接口	UART0	○	x	○	x
	UART6	○	x	○	x
	CSI10	○	x	x	○ (SCK10 引脚) 注
	IIC0	○	x	x	○ (SCL0 引脚) 注

注 在 STOP 模式中，不能以外围硬件引脚的外部时钟运行外围硬件。

备注 ○：可选，x：不可选

第 6 章 16 位定时器 / 事件计数器 00

6.1 16 位定时器 / 事件计数器 00 的功能

16 位定时器 / 事件计数器 00 具有以下功能。

(1) 间隔定时器

以事先设置的任意时间间隔发生中断。

(2) 方波输出

可输出任意频率的方波。

(3) 外部事件计数器

可测量外部输入信号的脉冲数。

(4) 单次脉冲输出

可输出任意输出脉宽单次脉冲。

(5) PPG 输出

可输出任意频率和输出脉宽的方波。

(6) 脉宽测量

可测量外部输入信号的脉宽。

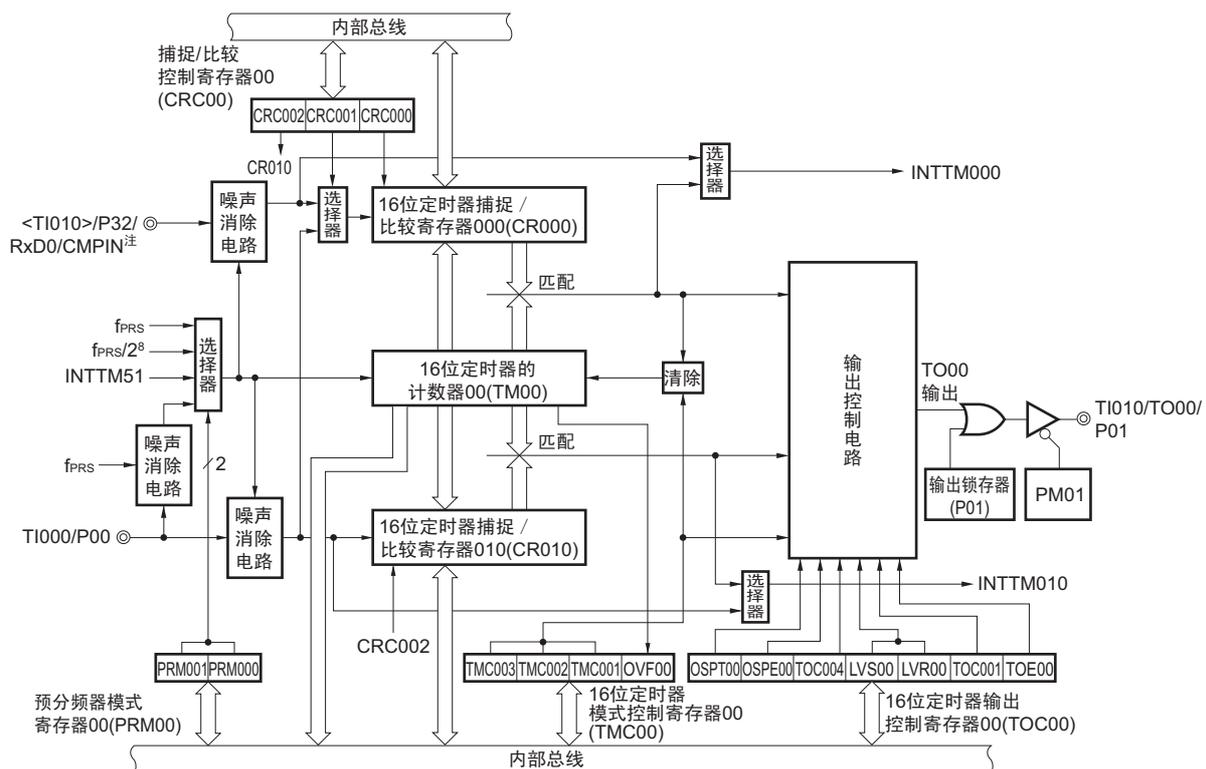
6.2 16 位定时器 / 事件计数器 00 的结构

16 位定时器 / 事件计数器 00 由以下硬件构成。

表 6-1 16 位定时器 / 事件计数器 00 的结构

项目	结构
定时器 / 计数器	16 位定时器计数器 00 (TM00)
寄存器	16 位定时器捕捉 / 比较寄存器 000、010 (CR000、CR010)
定时器输入	TI000、TI010
定时器输出	TO00、输出控制电路
控制寄存器	16 位定时器模式控制寄存器 00 (TMC00) 捕捉 / 比较控制寄存器 00 (CRC00) 16 位定时器输出控制寄存器 00 (TOC00) 预分频器模式寄存器 00 (PRM00) 端口模式寄存器 0 (PM0) 端口寄存器 0 (P0)

图 6-1 16 位定时器 / 事件计数器 00 的框图

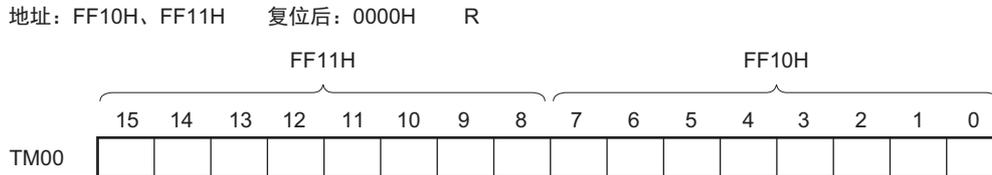


- 注意 1. P01 引脚不能同时用于 TI010 有效边沿和定时器输出 (TO00)。只能选择其中一项功能使用。
2. 如果 16 位定时器模式控制寄存器 00 (TMC00) 的 bit3 和 bit2 (TMC003、TMC002) 置“00”的时序与捕捉触发的输入发生竞争，捕捉数据就为不定值。
3. 如果要从捕捉模式切换到比较模式，必须先将 TMC003 和 TMC002 位置“00”，然后再进行切换。只要不进行复位，捕捉到的值就一直被保存在 CR000。如果切换到比较模式，就必须设置比较值。

(1) 16 位定时器计数器 00 (TM00)

TM00 是 16 位只读寄存器，对计数脉冲进行计数。
计数器与计数时钟的上升沿同步递增。

图 6-2 16 位定时器计数器 00 (TM00) 的格式



可通过在 16 位定时器模式控制寄存器 00 (TMC00) 的 bit2 和 bit3 (TMC002、TMC003) 不为“00”的状态下读 TM00，来读取 TM00 的计数值。如果在 TMC003 和 TMC002 为“00”的状态下读 TM00，TM00 的读取值就为“0000H”。

在以下情况下，计数值为“0000H”。

- 产生复位信号时
- 清除 TMC003 和 TMC002 (“00”) 时
- 在输入 TI000 引脚有效边沿时的清除 & 启动模式中，向 TI000 引脚输入有效边沿时
- 在 TM00 和 CR000 匹配时的清除 & 启动模式中，TM00 和 CR000 匹配时
- 在单次脉冲输出模式中，将 OSPT00 置“1”或者向 TI000 引脚输入有效边沿时

注意 即使读 TM00，读取值也不捕捉到 CR010。

(2) 16 位定时器捕捉 / 比较寄存器 000 (CR000)、16 位定时器捕捉 / 比较寄存器 010 (CR010)

CR000 和 CR010 是可切换使用捕捉功能和比较功能的 16 位寄存器。通过 CRC00 切换捕捉功能和比较功能。

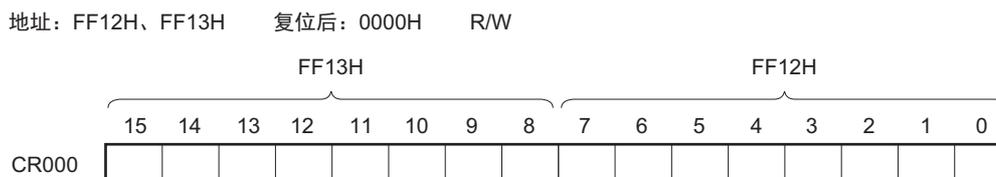
在定时器停止 (TMC003 和 TMC002=00) 时改写 CR000。

只有按规定的方法进行设置后，可在计数器运行期间改写 CR010。详细内容请参照“6.5.1 在 TM00 运行期间改写 CR010”。

可以 16 位单位读 / 写 CR000 和 CR010。

在产生复位信号后，CR000 和 CR010 为“0000H”。

图 6-3 16 位定时器捕捉 / 比较寄存器 000 (CR000) 的格式



(i) CR000 用作比较寄存器

不断比较 CR000 的设置值与 TM00 的计数值，一旦两者匹配，就产生中断信号（INTTM000）。保持该值直到被改写为止。

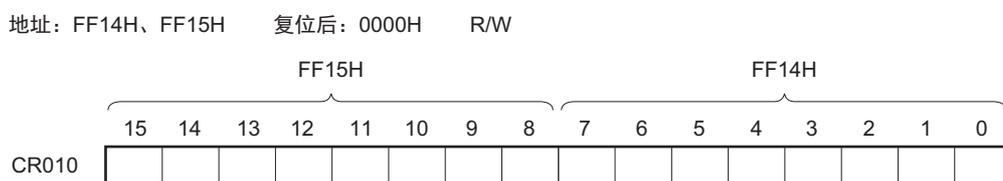
注意 即使输入捕捉触发，设置为比较模式的 CR000 也不执行捕捉运行。

(ii) CR000 用作捕捉寄存器

通过输入捕捉触发，将 TM00 的计数值捕捉到 CR000。

可选择 TI000 引脚的有效边沿的反相或者 TI010 引脚的有效边沿作为捕捉触发。通过 CRC00 和 PRM00 选择捕捉触发。

图 6-4 16 位定时器捕捉 / 比较寄存器 010（CR010）的格式



(i) CR010 用作比较寄存器

不断比较 CR010 的设置值与 TM00 的计数值，一旦两者匹配，就产生中断信号（INTTM010）。

注意 即使输入捕捉触发，设置为比较模式的 CR010 也不执行捕捉运行。

(ii) CR010 用作捕捉寄存器

通过输入捕捉触发，将 TM00 的计数值捕捉到 CR010。

可选择 TI000 引脚的有效边沿作为捕捉触发。通过 PRM00 设置 TI000 引脚的有效边沿。

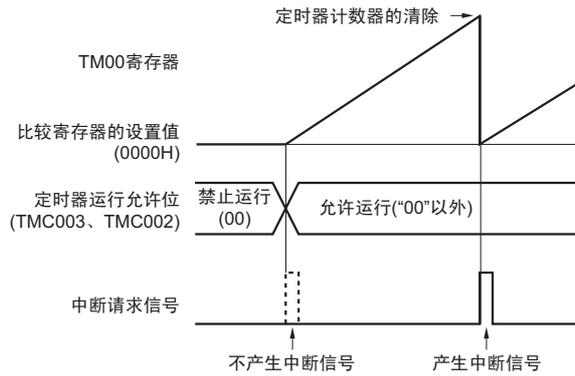
(iii) CR000、CR010 用作比较寄存器时的设置范围

在 CR000、CR010 用作比较寄存器时，设置值必须在以下范围内。

运行	CR000 的设置范围	CR010 的设置范围
作为间隔定时器的运行	$0000H < N \leq FFFFH$	$0000H \text{注} \leq M \leq FFFFH$
方波的输出运行		一般不使用。
作为外部事件计数器的运行		屏蔽匹配中断信号（INTTM010）。
作为输入 TI000 引脚有效边沿时的清除 & 启动模式的运行	$0000H \text{注} \leq N \leq FFFFH$	$0000H \text{注} \leq M \leq FFFFH$
作为自由运行定时器的运行		
PPG 的输出运行	$M < N \leq FFFFH$	$0000H \text{注} \leq M < N$
单次脉冲的输出运行	$0000H \text{注} \leq N \leq FFFFH (N \neq M)$	$0000H \text{注} \leq M \leq FFFFH (M \neq N)$

注 在设置为“0000H”时，不发生定时器运行后的匹配中断，也不改变定时器输出，并且第一个匹配时序如下所示。在定时器计数器（TM00 寄存器）从“0000H”变为“0001H”时，发生匹配中断。

- 因上溢引起定时器计数器清除时
- 因 TI000 引脚的有效边沿引起定时器计数器清除时（输入 TI000 引脚有效边沿时的清除 & 启动模式）
- 因比较匹配引起定时器计数器清除时（TM00 和 CR000 匹配时的清除 & 启动模式（CR000=0000H 以外的值、CR010=0000H））



备注 1. N: CR000 的设置值、M: CR010 的设置值

- 有关运行允许位 (TMC00 的 bit3 和 bit2 (TMC003 和 TMC002) 的详细内容, 请参照“6.3 (1) 16 位定时器模式控制寄存器 00 (TMC00)”。

表 6-2 CR000 和 CR010 的捕捉运行

外部输入信号	TI000 引脚输入		TI010 引脚输入	
捕捉运行				
CR000 的捕捉运行	CRC001=1 TI000 引脚输入 (反相)	ES010 和 ES000 的设置值 捕捉边沿的位置	CRC001 位=0 TI010 引脚输入	ES110 和 ES100 的设置值 捕捉边沿的位置
		01: 上升沿		01: 上升沿
		00: 下降沿		00: 下降沿
	11: 双边沿 (不能捕捉)	11: 双边沿		
中断信号	即使捕捉, 也不产生 INTTM000 信号。		中断信号	在每次捕捉时, 产生 INTTM000 信号。
CR010 的捕捉运行	TI000 引脚输入 ^注	ES010 和 ES000 的设置值 捕捉边沿的位置		
		01: 上升沿		
		00: 下降沿		
	11: 双边沿			
中断信号	在每次捕捉时, 产生 INTTM010 信号。			

注 CRC001 位的设置不影响 CR010 的捕捉运行。

注意 如果通过 TI000 引脚输入的反相将 TM00 寄存器的计数值捕捉到 CR000 寄存器, 则捕捉后不产生中断请求信号 (INTTM000)。如果在此运行期间检测到 TI010 引脚的有效边沿, 就不执行捕捉运行, 而产生 INTTM000 信号作为外部中断信号。不使用外部中断时, 必须屏蔽 INTTM000 信号。

备注 CRC001: 参照“6.3 (2) 捕捉 / 比较控制寄存器 00 (CRC00)”。
ES110、ES100、ES010、ES000: 参照“6.3 (4) 预分频器模式寄存器 00 (PRM00)”。

6.3 16 位定时器 / 事件计数器 00 的控制寄存器

16 位定时器 / 事件计数器 00 的控制寄存器如下所示。

- 16 位定时器模式控制寄存器 00 (TMC00)
- 捕捉/比较控制寄存器 00 (CRC00)
- 16 位定时器输出控制寄存器 00 (TOC00)
- 预分频器模式寄存器 00 (PRM00)
- 端口模式寄存器 0 (PM0)
- 端口寄存器 0 (P0)

(1) 16 位定时器模式控制寄存器 00 (TMC00)

TMC00 是 8 位寄存器，用于设置 16 位定时器 / 事件计数器 00 的运行模式、TM00 的清除模式、输出时序以及检测上溢。

运行期间 (TMC003 和 TMC002 不为“00”时) 禁止改写 TMC00。

但是，如果将 TMC003 和 TMC002 置“00” (停止运行)，在 OVF00 为“0”的情况下就可改写 TMC00。

通过 1 位或 8 位存储器操作指令设置 TMC00。

在产生复位信号后，TMC00 变为“00H”。

注意 在 TMC003 和 TMC002 设置为“00” (运行停止模式) 以外的值时，16 位定时器 / 事件计数器 00 开始运行。如果要停止运行，必须将 TMC003 和 TMC002 置“00”。

图 6-5 16 位定时器模式控制寄存器 00 (TMC00) 的格式

地址: FF86H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	<0>
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

TMC003	TMC002	允许 16 位定时器 / 事件计数器 00 运行
0	0	禁止 16 位定时器 / 事件计数器 00 运行 停止提供运行时钟 清除 16 位定时器计数器 00 (TM00)
0	1	自由运行定时器模式
1	0	输入 TI000 引脚有效边沿时的清除 & 启动模式
1	1	TM00 与 CR000 匹配时的清除 & 启动模式

TMC001	定时器输出 (TO00) 的取反条件
0	• TM00 与 CR000 匹配、TM00 与 CR010 匹配
1	• TM00 与 CR000 匹配、TM00 与 CR010 匹配 • TI000 引脚有效边沿的触发输入

OVF00	TM00 的上溢标志
清除 (0)	OVF00 清“0”或者 TMC003 和 TMC002=00
设置 (1)	发生上溢
在所有运行模式下 (自由运行定时器模式、输入 TI000 引脚有效边沿时的清除 & 启动模式以及 TM00 与 CR000 匹配时的清除 & 启动模式)，如果 TM00 的值从“FFFFH”变为“0000H”，OVF00 就被置“1”。也可通过给 OVF00 写“1”将 OVF00 置“1”。	

注 通过预分频器模式寄存器 00 (PRM00) 的 bit5 和 bit4 (ES010 和 ES000) 设置 TI000 引脚的有效边沿。

(2) 捕捉 / 比较控制寄存器 00 (CRC00)

CRC00 控制 CR000 与 CR010 的运行。

运行期间 (TMC003 和 TMC002 不为“00”时) 禁止改写 CRC00。

通过 1 位或 8 位存储器操作指令设置 CRC00。

在产生复位信号后, CRC00 变为“00H”。

图 6-6 捕捉 / 比较控制寄存器 00 (CRC00) 的格式

地址: FF88H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
CRC00	0	0	0	0	0	CRC002	CRC001	CRC000

CRC002	选择 CR010 的运行模式
0	作为比较寄存器运行
1	作为捕捉寄存器运行

CRC001	选择 CR000 的捕捉触发
0	在 TI010 引脚的有效边沿进行捕捉
1	在 TI000 引脚的有效边沿的反相进行捕捉注

通过 PRM00 设置 TI010 和 TI000 引脚的有效边沿。
如果在 CRC001 为“1”时将 ES010 和 ES000 置“11” (双边沿), 就不能检测 TI000 引脚的有效边沿。

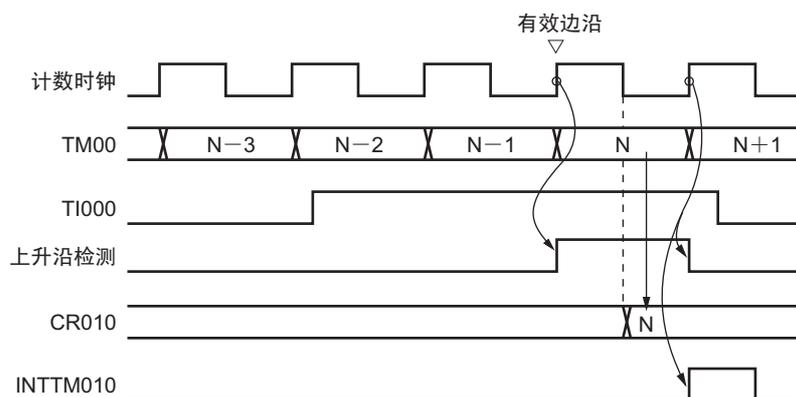
CRC000	选择 CR000 的运行模式
0	作为比较寄存器运行
1	作为捕捉寄存器运行

如果将 TMC003 和 TMC002 置“11” (TM00 和 CR000 匹配时的清除 & 启动模式), 就必须将 CRC000 置“0”。

注 如果检测到 TI010 引脚的有效边沿, 就不执行捕捉运行, 但是产生作为外部中断信号的 INTTM000 信号。

注意 为执行捕捉, 捕捉触发的脉冲必须大于 2 个预分频器模式寄存器 00 (PRM00) 选择的计数时钟周期。

图 6-7 CR010 的捕捉运行示例 (指定上升沿时)



(3) 16 位定时器输出控制寄存器 00 (TOC00)

TOC00 是 8 位寄存器，用于控制 TO00 输出。

运行期间 (TMC003 和 TMC002 不为“00”时) 只能改写 TOC00 的 OSPT00 位，禁止改写其他位。

定时器运行期间可改写作为更改 CR010 值手段的 TOC004 (请参照“6.5.1 在 TM00 运行期间改写 CR010”)。

通过 1 位或 8 位存储器操作指令设置 TOC00。

在产生复位信号后，TOC00 变为“00H”。

注意 必须按照以下步骤设置 TOC00。

①将 TOC004 和 TOC001 置“1”

②单独将 TOE00 置“1”

③将 LVS00 或者 LVR00 置“1”

图 6-8 16 位定时器输出控制寄存器 00 (TOC00) 的格式 (1/2)

地址: FF89H 复位后: 00H R/W

符号	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00

OSPT00	通过软件产生的单次脉冲输出触发
0	—
1	单次脉冲输出

读取值总是为“0”。在非单次脉冲输出模式下，不能置“1”。
如果置“1”，TM00 就执行清除和启动。

OSPE00	单次脉冲输出的运行控制
0	连续脉冲输出
1	单次脉冲输出

在自由运行定时器模式或者输入 TI000 引脚的效边沿时的清除 & 启动模式下，单次脉冲的输出运行正常。
在 TM00 与 CR000 匹配时的清除 & 启动模式下，不能进行单次脉冲输出。

TOC004	CR010 和 TM00 匹配时的 TO00 输出控制
0	禁止取反
1	允许取反

即使 TOC004=0，也产生中断信号 (INTTM010)。

图 6-8 16 位定时器输出控制寄存器 00 (TOC00) 的格式 (2/2)

地址: FF89H 复位后: 00H R/W

符号	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00

LVS00	LVR00	TO00 输出状态的设置
0	0	不变化
0	1	TO00 输出初始值为低电平 (将 TO00 输出清“0”)。
1	0	TO00 输出初始值为高电平 (将 TO00 输出置“1”)。
1	1	禁止设置

- LVS00 与 LVR00 可设置 TO00 输出电平的初始值。无需设置时, 将 LVS00 和 LVR00 保持为“00”。
- 必须在 TOE00=1 时设置 LVS00 和 LVR00。
禁止将 LVS00 或者 LVR00 和 TOE00 同时置“1”。
- LVS00 和 LVR00 是触发位。通过将这两位置“1”, 设置 TO00 输出电平的初始值。即使将这两位清“0”, 也不影响 TO00 输出。
- LVS00 和 LVR00 的读取值总是为“0”。
- 有关 LVS00 和 LVR00 的设置方法, 请参照“6.5.2 LVS00 和 LVR00 的设置”。
- 实际的 TO00/TI010/P01 引脚的 TO00 输出取决于 PM01 和 P01。

TOC001	CR000 和 TM00 匹配时的 TO00 输出控制
0	禁止取反
1	允许取反

即使 TOC001=0, 也产生中断信号 (INTTM000)。

TOE00	TO00 输出控制
0	禁止输出 (TO00 输出固定为低电平)
1	允许输出

(4) 预分频器模式寄存器 00 (PRM00)

PRM00 寄存器设置 TM00 的计数时钟以及 TI000 和 TI010 引脚输入的有效边沿。

运行期间 (TMC003 和 TMC002 不为“00”时) 禁止改写 PRM00。

通过 1 位或 8 位存储器操作指令设置 PRM00。

在产生复位信号后, PRM00 变为“00H”。

注意 1. 在将 PRM001 和 PRM000 位置“11” (将计数时钟指定为 TI000 引脚的有效边沿) 时, 禁止进行以下设置。

- 输入 TI000 引脚有效边沿时的清除 & 启动模式
 - 设置 TI000 引脚为捕捉触发
2. 复位后, 在 TI000 引脚或者 TI010 引脚处于高电平状态, 并且指定上升沿或双边沿为 TI000 引脚或者 TI010 引脚的有效边沿时, 如果允许 16 位定时器 / 事件计数器 00 运行, 则检测出的该高电平为上升沿。上拉 TI000 和 TI010 引脚时等要特别注意。如果允许一旦停止的运行重新启动, 则检测不出上升沿。
 3. P34 引脚不能同时用于 TI010 有效边沿和定时器输出 (TO00)。只能选择其中一项功能使用。

图 6-9 预分频器模式寄存器 00 (PRM00) 的格式

地址: FF87H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PRM00	ES110	ES100	ES010	ES000	0	0	PRM001	PRM000

ES110	ES100	TI010 引脚的有效边沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升和下降的双边沿

ES010	ES000	TI000 引脚的有效边沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升和下降的双边沿

PRM001	PRM000	计数时钟的选择		
			$f_{PRS}=2\text{MHz}$	$f_{PRS}=5\text{MHz}$
0	0	f_{PRS}	2MHz	5MHz
0	1	$f_{PRS}/2^8$	7.81kHz	19.53kHz
1	0	INTTM51		
1	1	TI000 有效边沿注 1、注 2		

注 1. TI000 引脚的外部时钟的脉冲必须大于 2 个外围硬件时钟 (f_{PRS}) 周期。

2. 在 STOP 模式下, 不能通过 TI000 引脚的外部时钟开始定时器运行。

备注 f_{PRS} : 外围硬件时钟频率

(5) 端口模式寄存器 0 (PM0)

PM0 以 1 位单位设置端口 0 的输入 / 输出。

将 P01/TO00/TI010 引脚用作定时器输出时，必须将 PM01 的输出锁存器置“0”。

将 P00/TI000、P01/TI010/TO00 引脚用作定时器输入时，必须将 PM00 和 PM01 置“1”。此时，P00 和 P01 的输出锁存器可以是“0”也可以是“1”。

通过 1 位或 8 位存储器操作指令设置 PM0。

在产生复位信号后，PM0 变为“FFH”。

图 6-10 端口模式寄存器 0 (PM0) 的格式

地址: FF20H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM0	1	1	1	1	1	1	PM01	PM00

PM0n	P0n 引脚的输入 / 输出模式选择 (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)

6.4 16 位定时器 / 事件计数器 00 的运行

6.4.1 作为间隔定时器的运行

如果将 16 位定时器模式控制寄存器 (TMC00) 的 bit3 和 bit2 (TMC003 和 TMC002) 置“11” (TM00 和 CR000 匹配时的计数清除 & 启动模式), 就与计数时钟同步开始计数运行。

此后, 如果 TM00 的值和 CR000 的值匹配, 就将 TM00 清“0000H”, 并且产生匹配中断信号 (INTTM000)。通过以一定间隔产生的 INTTM000 信号, 16 位定时器 / 事件计数器 00 作为间隔定时器运行。

备注 1. 有关输入 / 输出引脚的设置, 请参照“6.3 (5) 端口模式寄存器 0 (PM0)”。

2. 有关 INTTM000 信号的中断允许, 请参照“第 16 章 中断功能”。

图 6-11 间隔定时器运行的框图

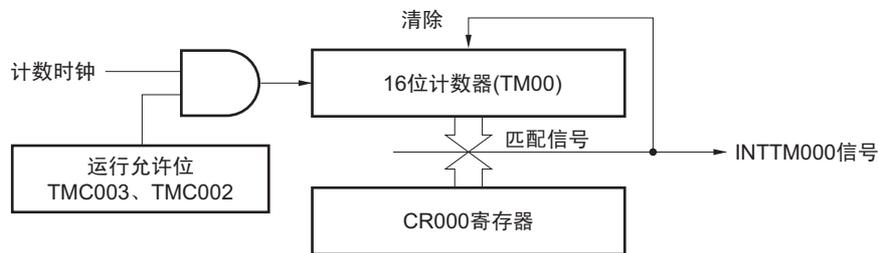


图 6-12 间隔定时器运行的基本时序示例

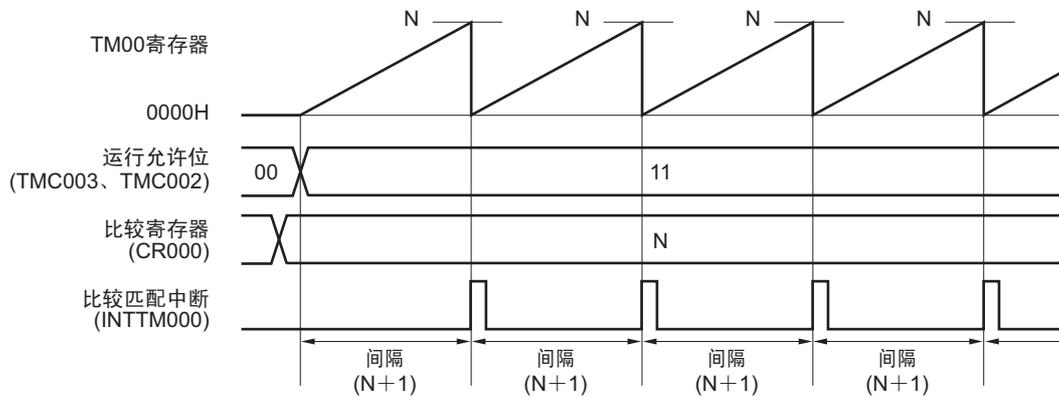
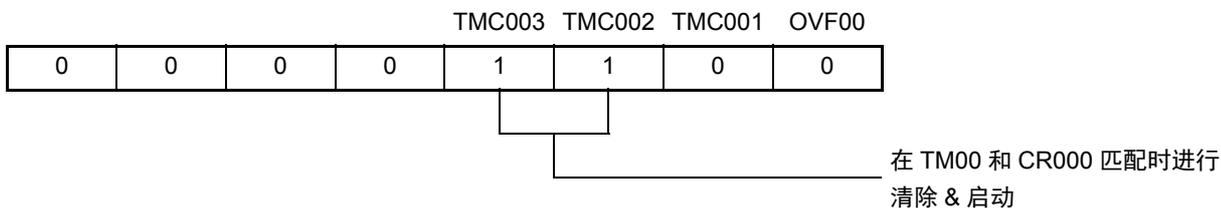
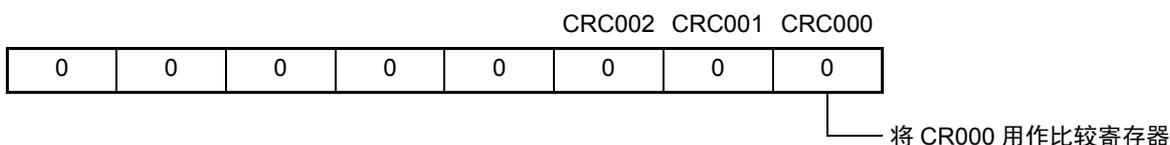


图 6-13 间隔定时器运行时的寄存器设置内容示例

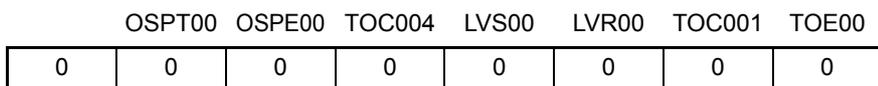
(a) 16 位定时器模式控制寄存器 00 (TMC00)



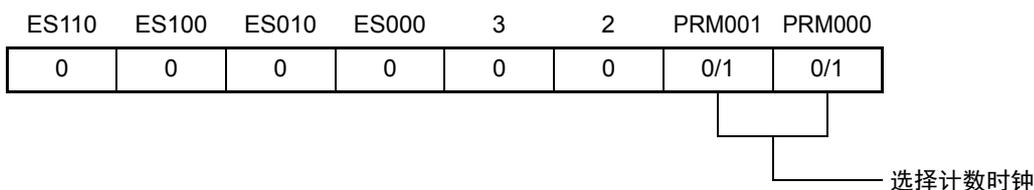
(b) 捕捉 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频器模式寄存器 00 (PRM00)



(e) 16 位定时器计数器 00 (TM00)

通过读 TM00，读取计数器的值。

(f) 16 位捕捉 / 比较寄存器 000 (CR000)

将 M 设置到 CR000 时，间隔时间如下：

- 间隔时间 = (M+1) × 计数时钟周期

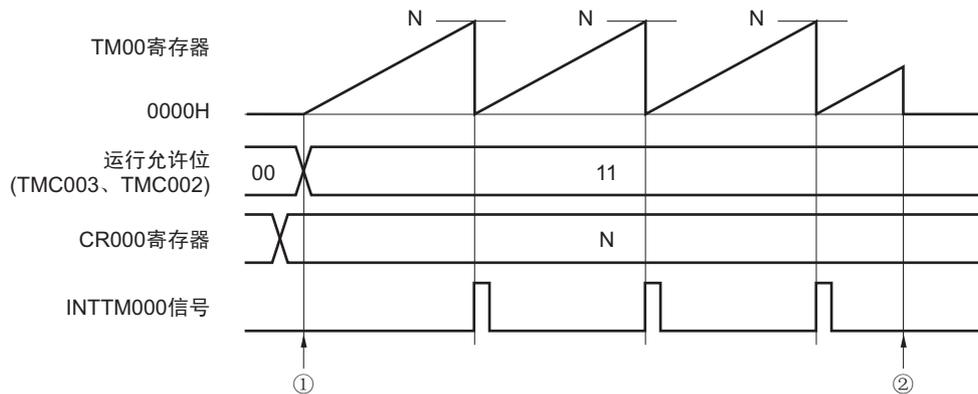
禁止将 CR000 设置为“0000H”。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

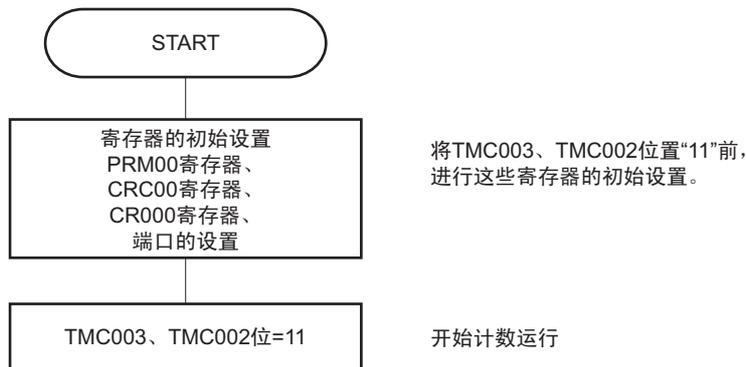
CR010 一般不用作间隔定时器。但是，如果 CR010 的设置值与 TM00 的值匹配，就发生比较匹配中断 (INTTM010)。

因此，通过中断屏蔽标志 (TMMK010) 进行屏蔽设置。

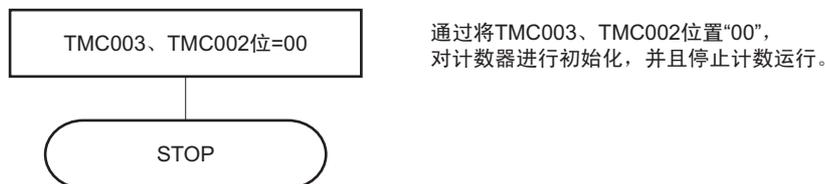
图 6-14 间隔定时器功能期间的软件处理示例



① 开始计数运行的流程



② 停止计数运行的流程



6.4.2 方波的输出运行

当 16 位定时器 / 事件计数器 00 作为间隔定时器（参照 6.4.1）运行时，可通过将 16 位定时器输出控制寄存器 00（TOC00）置“03H”，从 TO00 引脚输出方波。

如果将 TMC003 和 TMC002 置“11”（TM00 和 CR000 匹配时的清除 & 启动模式），就与计数时钟同步开始计数运行。

此后，如果 TM00 的值和 CR000 的值匹配，就将 TM00 清“0000H”，然后产生中断信号（INTTM000）并且取反 TO00 输出。通过以一定间隔取反 TO00 输出，16 位定时器 / 事件计数器 00 进行方波的输出运行。

备注 1. 有关输入 / 输出引脚的设置，请参照“6.3 (5) 端口模式寄存器 0（PM0）”。

2. 有关 INTTM000 信号的中断允许，请参照“第 16 章 中断功能”。

图 6-15 方波输出运行的框图

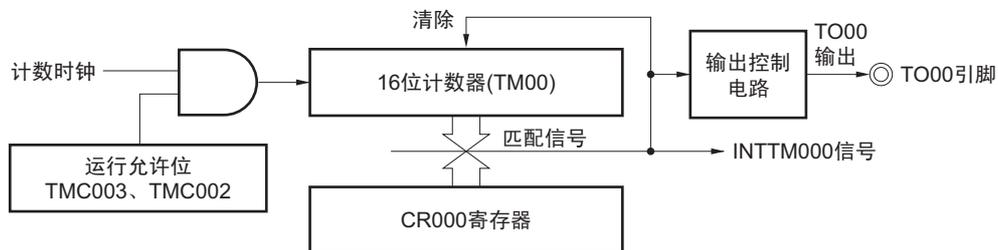


图 6-16 方波输出运行的基本时序示例

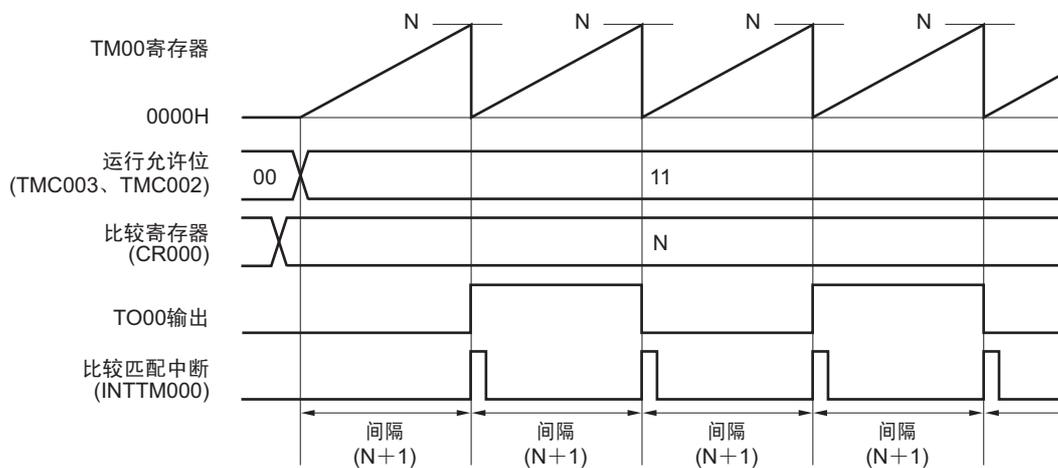
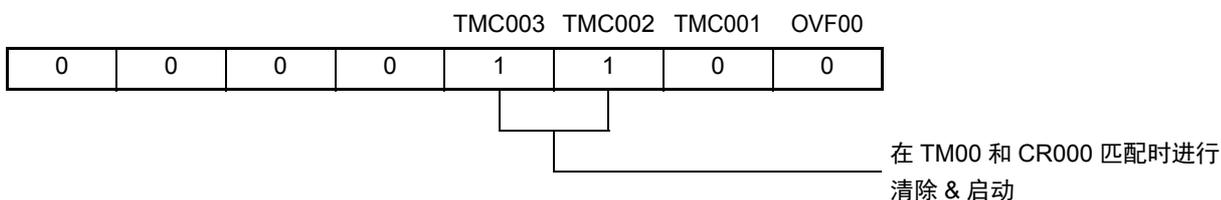
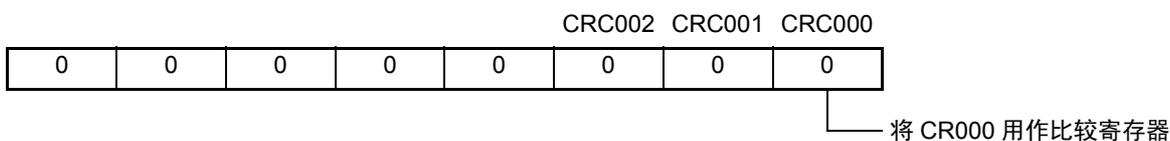


图 6-17 方波输出运行时的寄存器设置内容示例 (1/2)

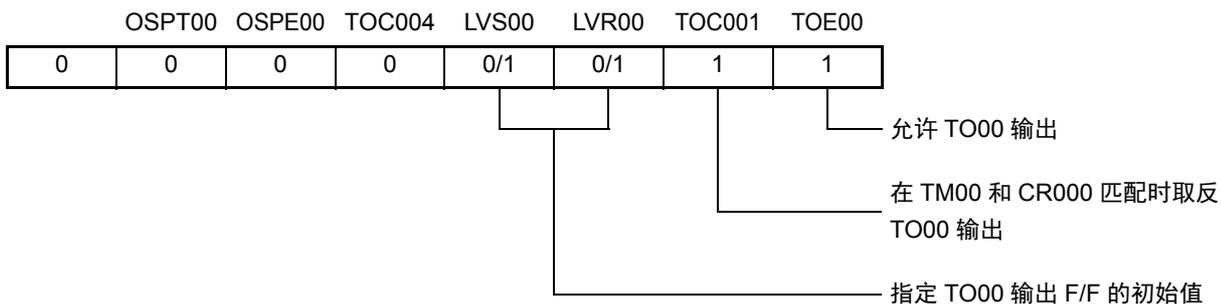
(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕捉 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频器模式寄存器 00 (PRM00)

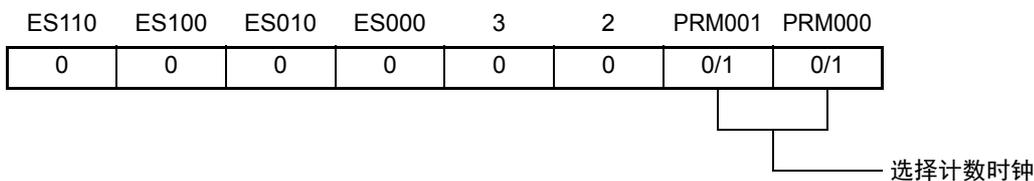


图 6-17 方波输出运行时的寄存器设置内容示例 (2/2)

(e) 16 位定时器计数器 00 (TM00)

通过读 TM00，读取计数器的值。

(f) 16 位捕捉 / 比较寄存器 000 (CR000)

将 M 设置到 CR000 时，方波频率如下：

- 方波频率 = $1 \div \{2 \times (M+1) \times \text{计数时钟周期}\}$

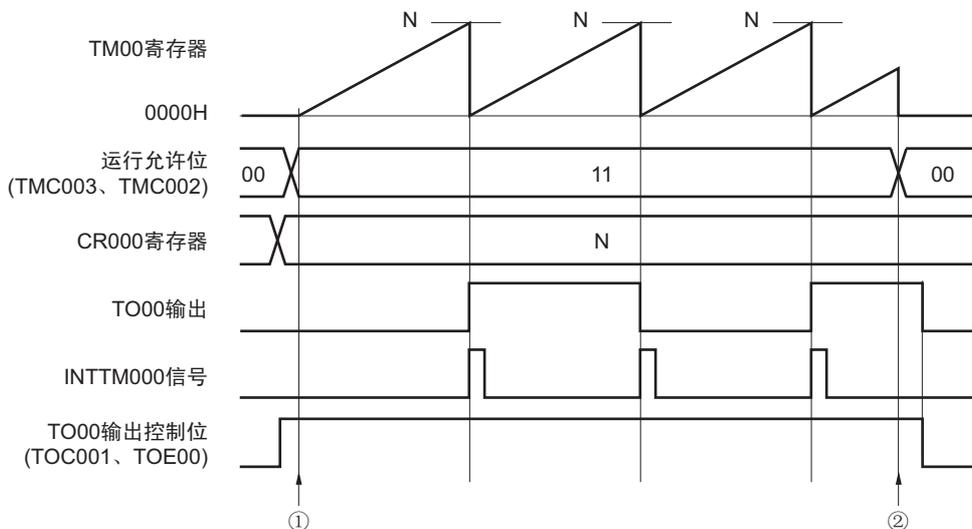
禁止将 CR000 设置为“0000H”。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

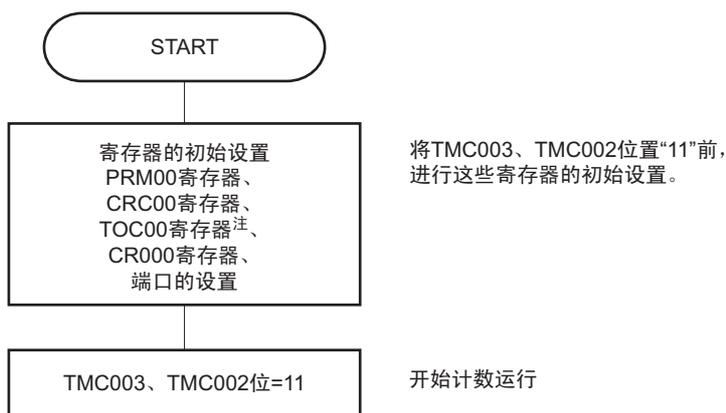
CR010 一般不用于方波输出功能。但是，如果 CR010 的设置值与 TM00 的值匹配，就发生比较匹配中断 (INTTM010)。

因此，通过中断屏蔽标志 (TMMK010) 进行屏蔽设置。

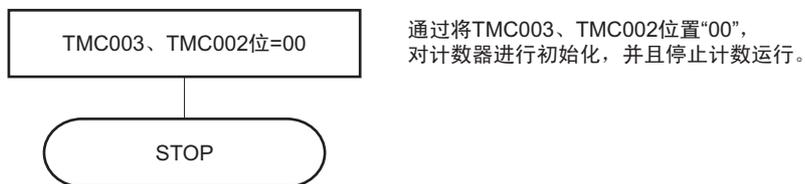
图 6-18 方波输出功能期间的软件处理示例



① 开始计数运行的流程



② 停止计数运行的流程



注 设置 TOC00 时，应特别注意。详细内容请参照“6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)”。

6.4.3 作为外部事件计数器的运行

如果将预分频器模式寄存器 00 (PRM00) 的 bit1 和 bit0 (PRM001 和 PRM000) 置“11” (通过 TI000 引脚的有效边沿进行递增计数), 将 16 位定时器模式控制寄存器 00 (TMC00) 的 bit3 和 bit2 (TMC003 和 TMC002) 置“11”, 就对外部事件输入的有效边沿进行计数, 并且产生 TM00 和 CR000 的匹配中断信号 (INTTM000)。

使用 TI000 引脚作为外部事件的输入引脚。因此, 在输入 TI000 有效边沿时的清除 & 启动模式 (TMC003 和 TMC002=10) 中, 16 位定时器 / 事件计数器 00 不能用作外部事件计数器。

在以下时序产生 INTTM000 信号。

- INTTM000 信号的产生时序 (第 2 次以及第 2 次以后)
= 外部事件输入的有效边沿检测次数 × (CR000 设置值 + 1)

但是, 只在以下时序发生运行开始后的第 1 次中断。

- INTTM000 信号的产生时序 (仅第 1 次)
= 外部事件输入的有效边沿检测次数 × (CR000 设定值 + 2)

以 f_{PRS} 时钟周期对 TI000 引脚输入信号进行采样, 在连续两次检测出有效边沿时, 才被认为检测出有效边沿。因此, 可消除窄脉宽的噪音。

备注 1. 有关输入 / 输出引脚的设置, 请参照“6.3 (5) 端口模式寄存器 0 (PM0)”。

2. 有关 INTTM000 信号的中断允许, 请参照“第 16 章 中断功能”。

图 6-19 作为外部事件计数器运行的框图

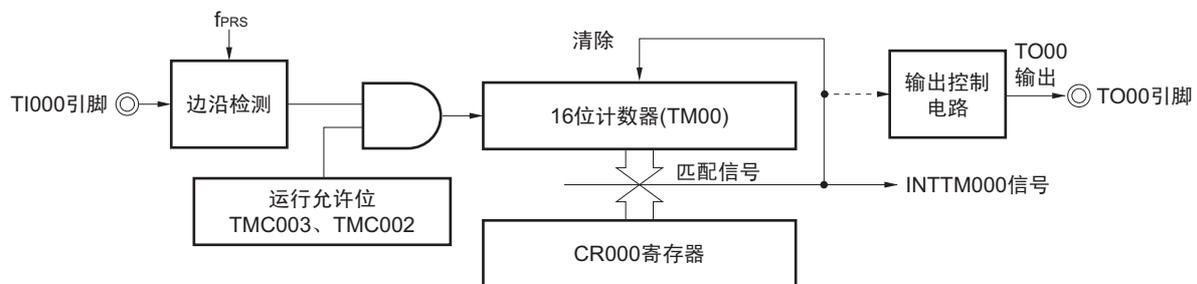
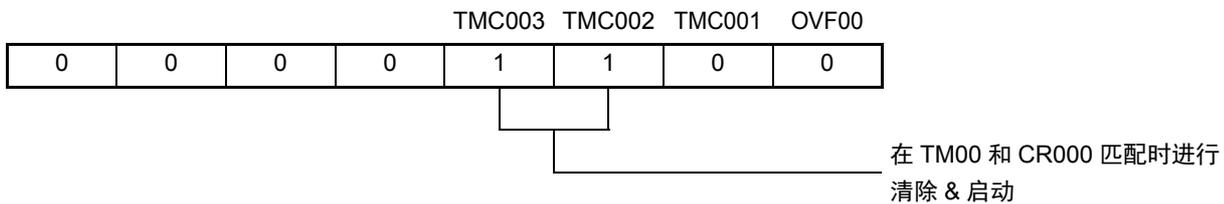
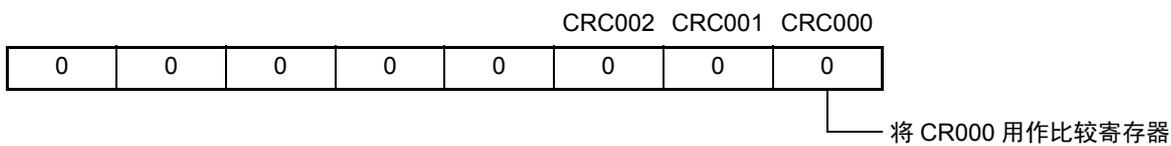


图 6-20 外部事件计数器运行时的寄存器设置内容示例 (1/2)

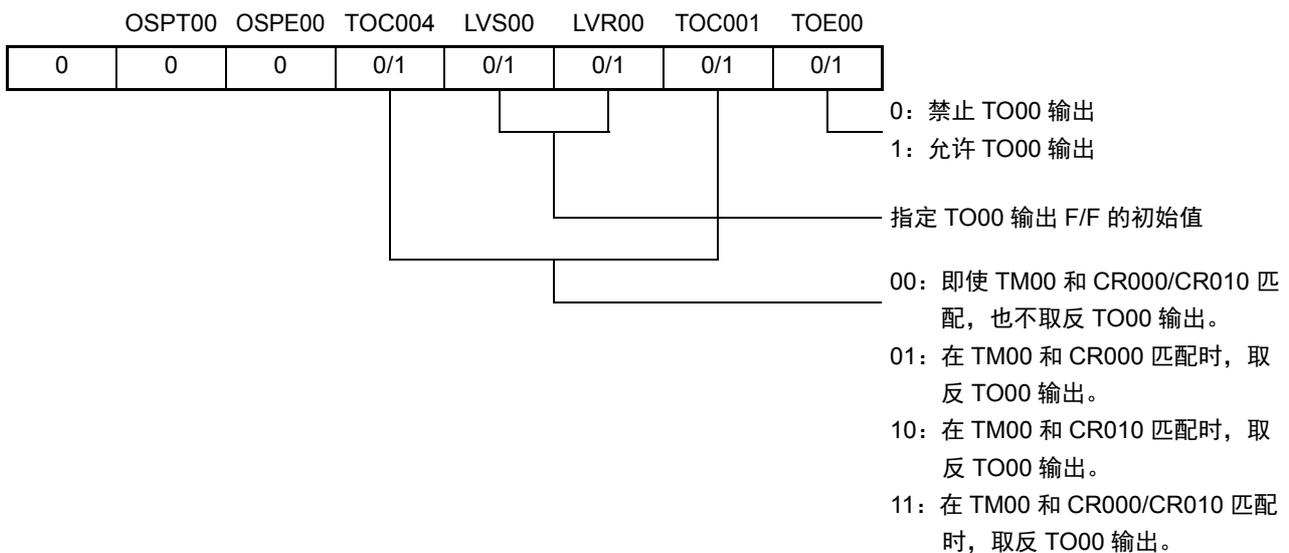
(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕捉 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频器模式寄存器 00 (PRM00)

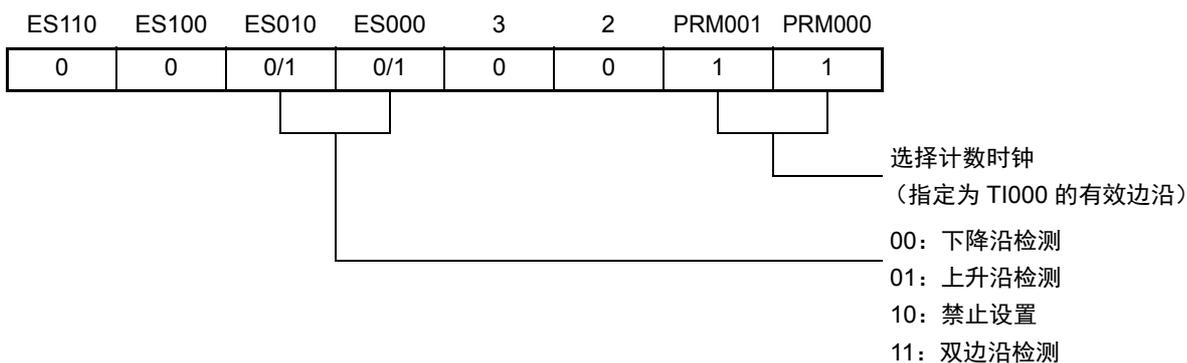


图 6-20 外部事件计数器运行时的寄存器设置内容示例 (2/2)

(e) 16 位定时器计数器 00 (TM00)

通过读 TM00，读取计数器的值。

(f) 16 位捕捉 / 比较寄存器 000 (CR000)

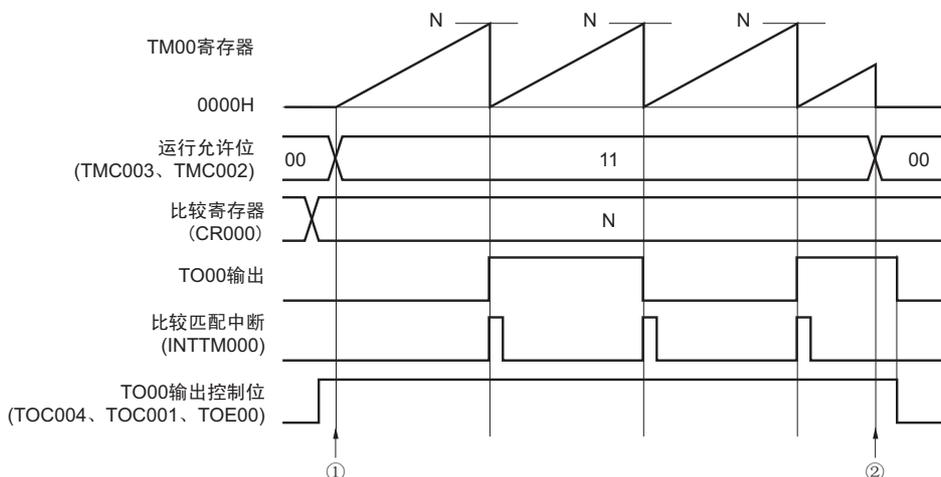
将 M 设置到 CR000 时，如果输入 M+1 次外部事件，就产生中断信号 (INTTM000)。禁止将 CR000 设置为“0000H”。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

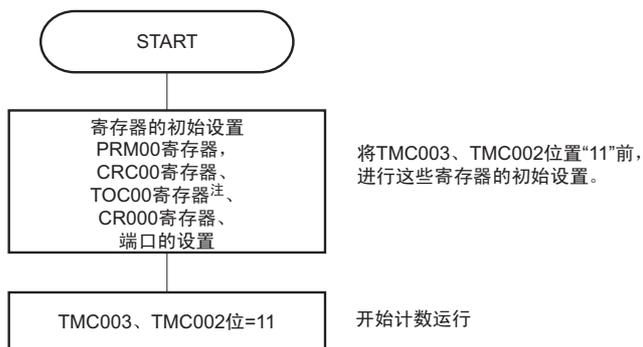
CR010 一般不作为部事件计数器运行。但是，如果 CR010 的设置值与 TM00 的值匹配，就发生比较匹配中断 (INTTM010)。

因此，通过中断屏蔽标志 (TMMK010) 进行屏蔽设置。

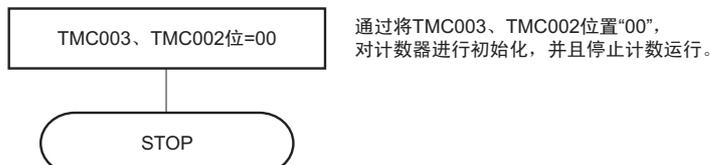
图 6-21 外部事件计数器运行时的软件处理示例



① 开始计数运行的流程



② 停止计数运行的流程



注 设置 TOC00 时，应特别注意。详细内容请参照“6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)”。

6.4.4 作为输入 TI000 引脚有效边沿时的清除 & 启动模式的运行

在将 16 位定时器模式控制寄存器 00 (TMC00) 的 bit3 和 bit2 (TMC003 和 TMC002) 置“10” (输入 TI000 引脚有效边沿时的清除 & 启动模式) 时, 如果提供计数时钟 (通过 PRM00 设置), TM00 就开始递增计数。一旦在计数过程中检测出 TI000 引脚的有效边沿, 就将 TM00 清“0000H”, 然后重新开始递增计数。如果没有检测出 TI000 引脚的有效边沿, TM00 就发生上溢, 并且持续计数。

TI000 引脚的有效边沿为 TM00 的清除源, 并不控制运行开始后立即开始计数的启动。
CR000 和 CR010 可用作比较寄存器和捕捉寄存器。

(a) CR000 和 CR010 用作比较寄存器

在 TM00 的值与 CR000、CR010 的值匹配时, 产生 INTTM000、INTTM010 信号。

(b) CR000 和 CR010 用作捕捉寄存器

如果将有效边沿输入到 TI010 引脚 (或者将有效边沿的反相输入到 TI000 引脚), 就捕捉 TM00 的计数值到 CR000, 并产生 INTTM000 信号。

如果将有效边沿输入到 TI000 引脚, 就捕捉 TM00 的计数值到 CR010, 并产生 INTTM010 信号。在 TI000 引脚的有效边沿进行捕捉运行的同时, 将计数器清“0000H”。

注意 不能将计数时钟设置为 TI000 引脚的有效边沿 (PRM001 和 PRM000=11)。如果将 PRM001 和 PRM000 置“11”, 就清除 TM00。

备注 1. 有关输入 / 输出引脚的设置, 请参照“6.3 (5) 端口模式寄存器 0 (PM0)”。

2. 有关 INTTM000 信号的中断允许, 请参照“第 16 章 中断功能”。

(1) 输入 TI000 引脚有效边沿时的清除 & 启动模式的运行 (CR000: 比较寄存器、CR010: 比较寄存器)

图 6-22 输入 TI000 引脚有效边沿时的清除 & 启动模式的框图
(CR000: 比较寄存器、CR010: 比较寄存器)

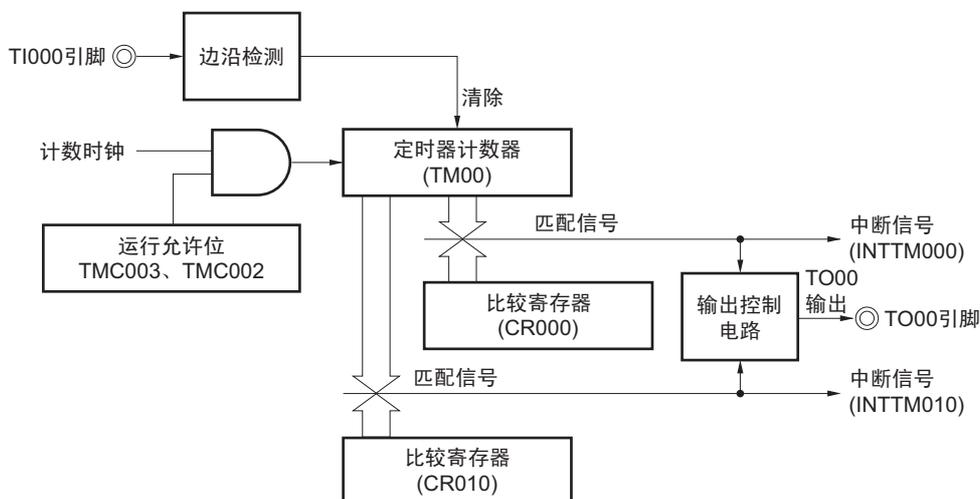
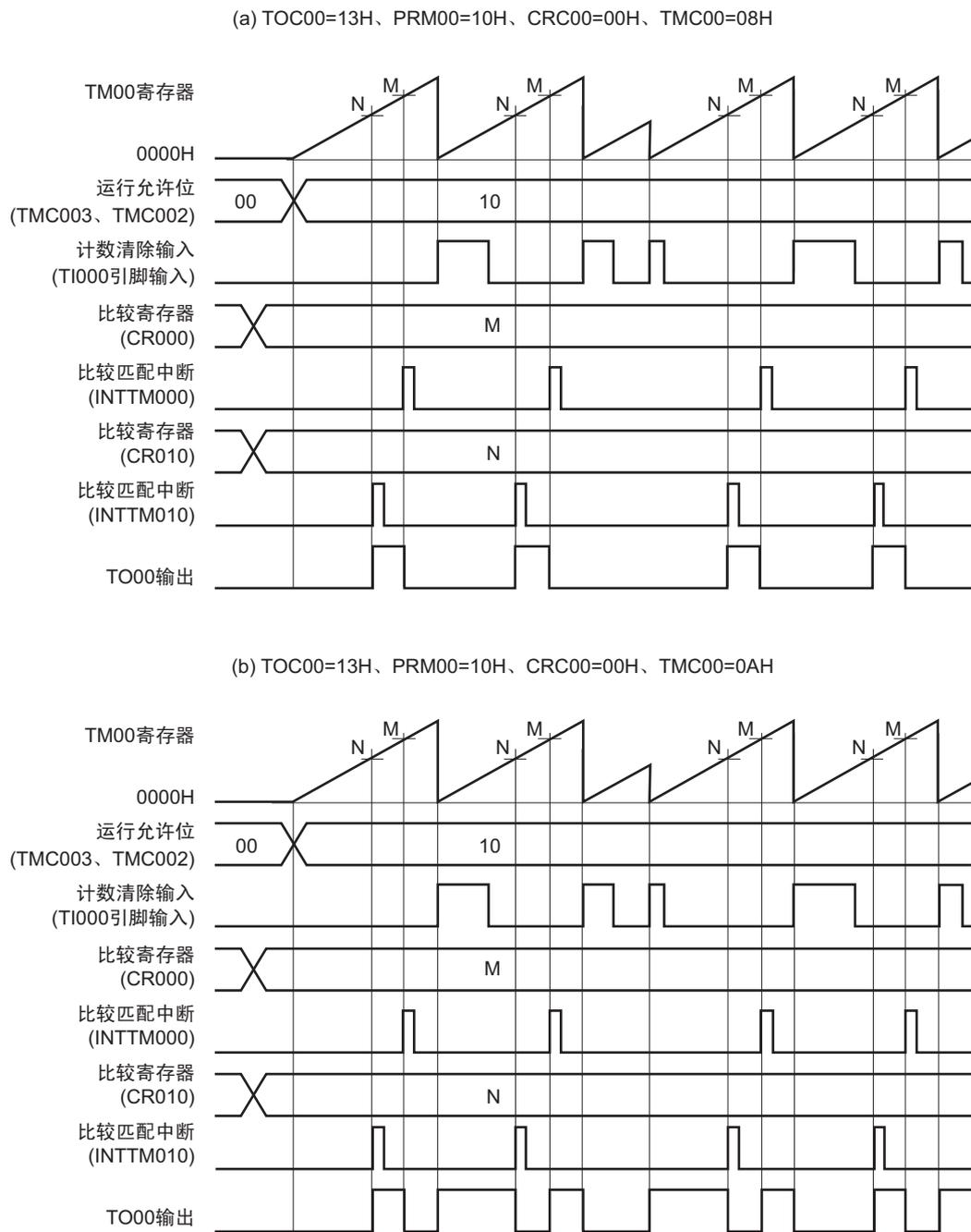


图 6-23 输入 TI000 引脚有效边沿时的清除 & 启动模式的时序示例
(CR000: 比较寄存器、CR010: 比较寄存器)



根据 16 位定时器模式控制寄存器 00 (TMC00) 的 bit1 (TMC001) 的设置, (a) 和 (b) 有如下不同。

(a) 在 TM00 与比较寄存器匹配时, 取反 TO00 的输出电平。

(b) 在 TM00 与比较寄存器匹配或者检测出 TI000 引脚的有效边沿时, 取反 TO00 的输出电平。

(2) 输入 TI000 引脚有效边沿时的清除 & 启动模式的运行
(CR000: 比较寄存器、CR010: 捕捉寄存器)

图 6-24 输入 TI000 引脚有效边沿时的清除 & 启动模式的框图
(CR000: 比较寄存器、CR010: 捕捉寄存器)

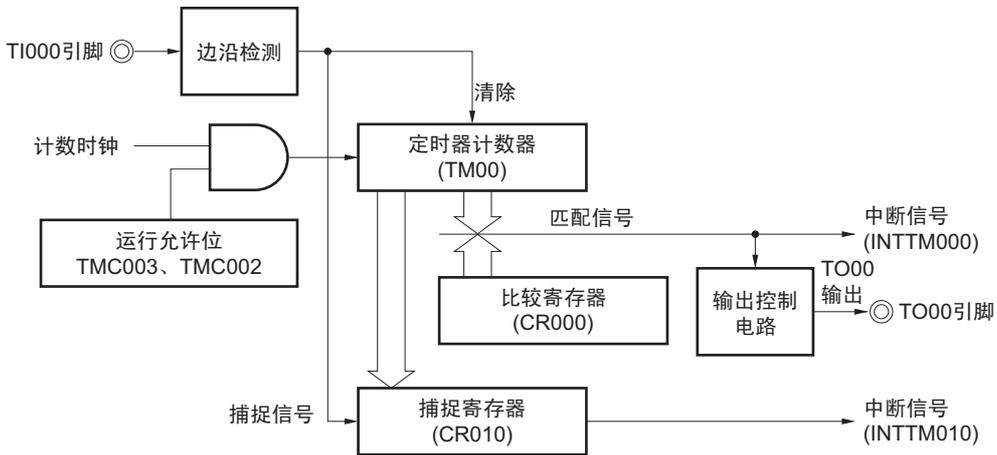
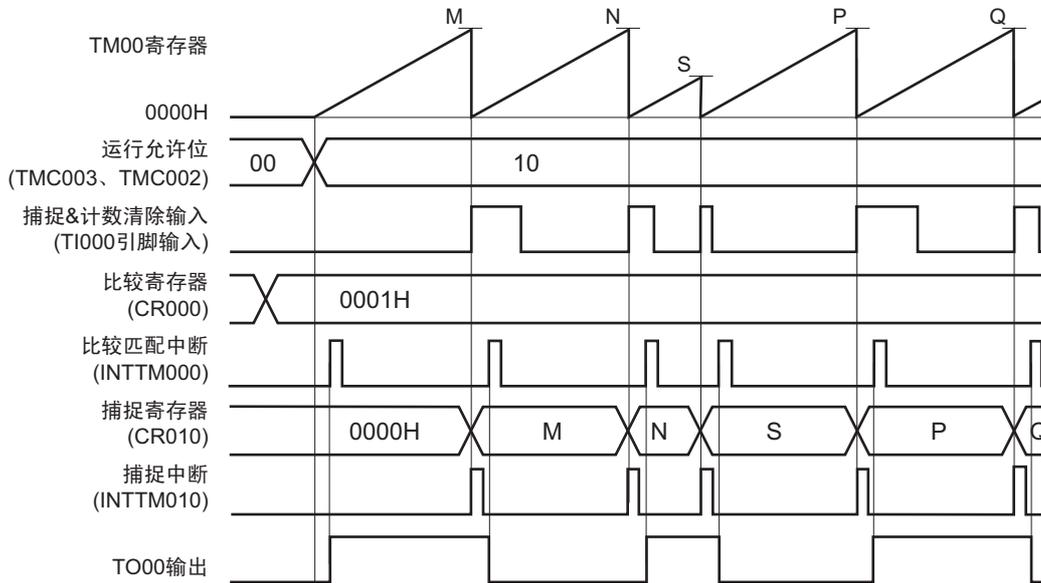


图 6-25 输入 TI000 引脚的有效边沿时的清除 & 启动模式的时序示例
(CR000: 比较寄存器、CR010: 捕捉寄存器) (1/2)

(a) TOC00=13H、PRM00=10H、CRC00=04H、TMC00=08H、CR000=0001H

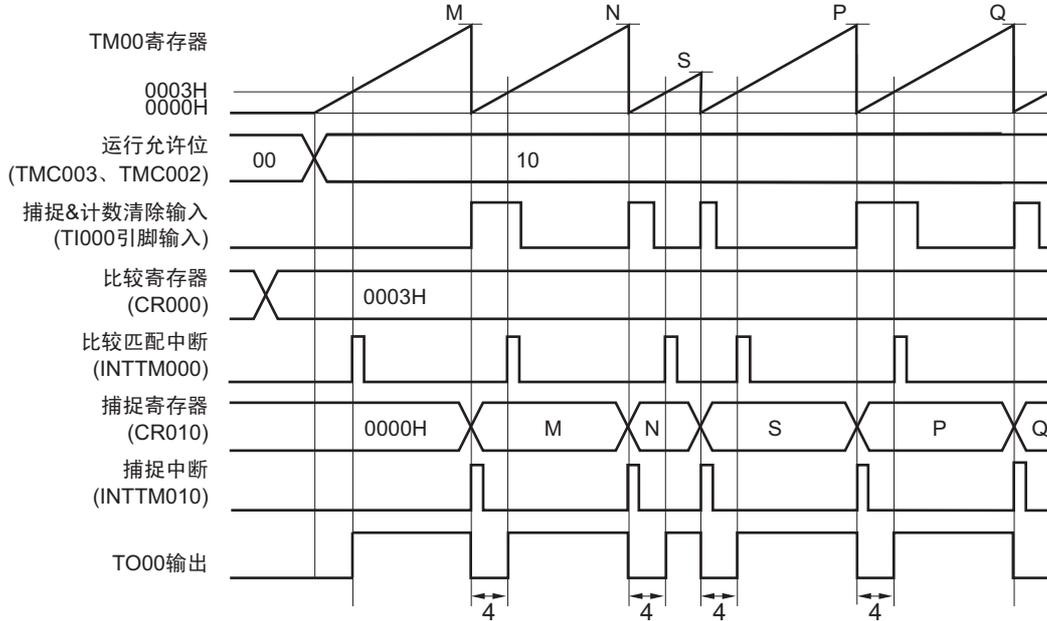


这是在进行捕捉 & 清除后，要取反 TO00 输出电平时的应用示例。

通过 TI000 引脚的有效边沿检测，捕捉计数值到 CR010，并且清除 TM00 (0000H)。一旦 TM00 的计数值变为“0001H”，就产生比较匹配中断信号 (INTTM000)，并且取反 TO00 输出电平。

图 6-25 输入 TI000 引脚有效边沿时的清除 & 启动模式的时序示例
(CR000: 比较寄存器、CR010: 捕捉寄存器) (2/2)

(b) TOC00=13H, PRM00=10H, CRC00=04H, TMC00=0AH, CR000=0003H



这是在进行捕捉 & 清除后，要从 TO00 引脚输出 CR000 设置的宽度（此例中为 4 个时钟）时的应用示例。

通过 TI000 引脚的有效边沿检测，捕捉计数值到 CR010，并且在产生捕捉中断信号（INTTM010）后，清除 TM00（0000H），然后取反 TO00 输出。一旦 TM00 的计数值变为“0003H”（对 4 个时钟计数），就产生比较匹配中断信号（INTTM000），并且取反 TO00 输出电平。

(3) 输入 TI000 引脚有效边沿时的清除 & 启动模式的运行
(CR000: 捕捉寄存器、CR010: 比较寄存器)

图 6-26 输入 TI000 引脚有效边沿时的清除 & 启动模式的框图
(CR000: 捕捉寄存器、CR010: 比较寄存器)

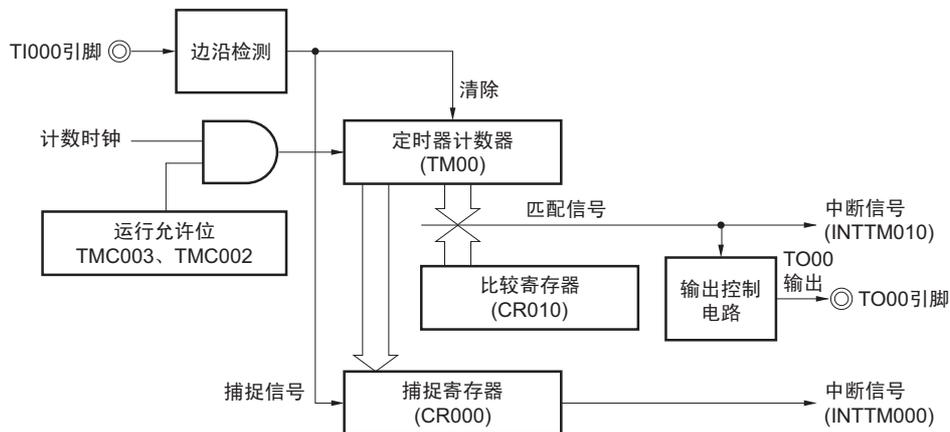
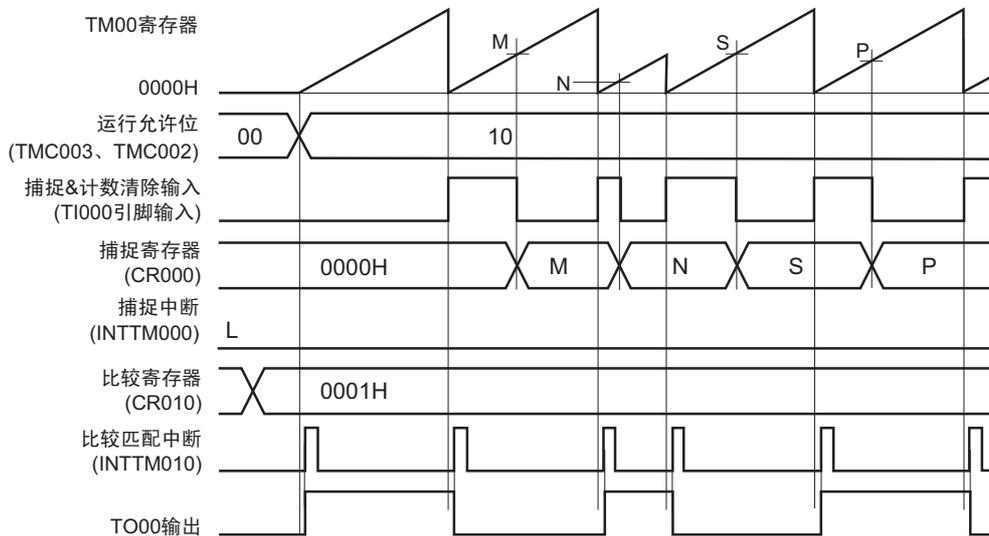


图 6-27 输入 TI000 引脚有效边沿时的清除 & 启动模式的时序示例
(CR000: 捕捉寄存器、CR010: 比较寄存器) (1/2)

(a) TOC00=13H、PRM00=10H、CRC00=03H、TMC00=08H、CR010=0001H



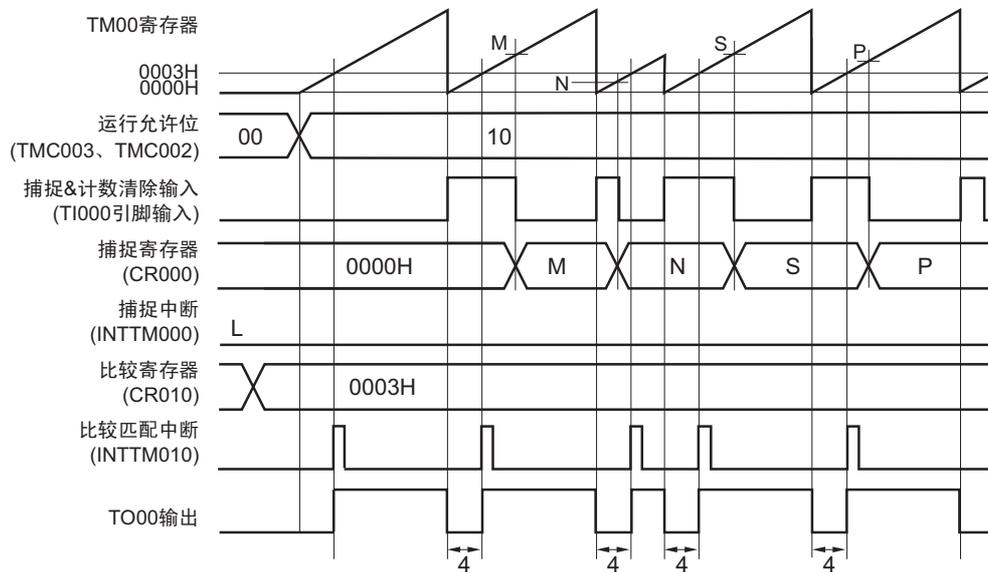
这是在进行捕捉 & 清除后，要取反 TO00 输出电平时的应用示例。

通过 TI000 引脚的上升沿检测，清除 TM00。通过 TI000 引脚的下降沿检测，将计数值捕捉到 CR000。

在将捕捉 / 比较控制寄存器 00 (CRC00) 的 bit1 (CRC001) 置“1”时，通过 TI000 引脚输入的反相捕捉 TM00 的计数值到 CR000，但是不产生捕捉中断信号 (INTTM000)。在检测出 TI010 引脚的有效边沿时，产生 INTTM000 信号。如果不使用 INTTM000 信号，必须将其屏蔽。

图 6-27 输入 TI000 引脚有效边沿时的清除 & 启动模式的时序示例
(CR000: 捕捉寄存器、CR010: 比较寄存器) (2/2)

(b) TOC00=13H、PRM00=10H、CRC00=03H、TMC00=0AH、CR010=0003H



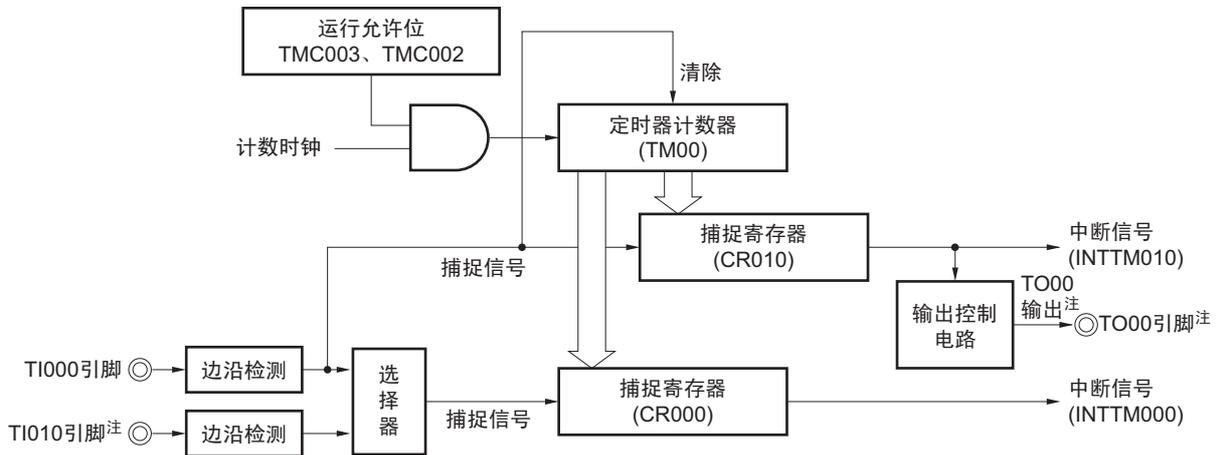
这是在进行捕捉 & 清除后，要从 TO00 引脚输出 CR010 设置的宽度（此例中为 4 个时钟）时的应用示例。

通过 TI000 引脚的上升沿检测，清除 TM00（0000H）。通过 TI000 引脚的下降沿检测，将计数值捕捉到 CR000。在因 TI000 引脚的上升沿检测引起的 TM00 清除（0000H）或者 TM00 和比较寄存器（CR010）匹配时取反 TO00 输出。

在将捕捉 / 比较控制寄存器 00（CRC00）的 bit1（CRC001）置“1”时，通过 TI000 引脚输入的反相捕捉 TM00 的计数值到 CR000，但是不产生捕捉中断信号（INTTM000）。在检测出 TI010 引脚的有效边沿时，产生 INTTM000 信号。如果不使用 INTTM000 信号，必须将其屏蔽。

(4) 输入 TI000 引脚有效边沿时的清除 & 启动模式的运行
(CR000: 捕捉寄存器、CR010: 捕捉寄存器)

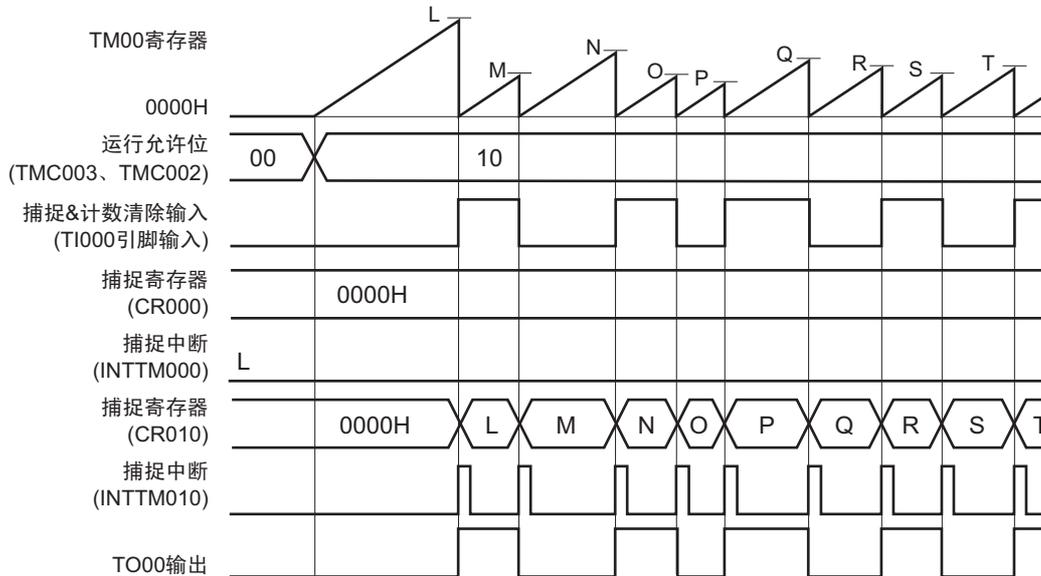
图 6-28 输入 TI000 引脚有效边沿时的清除 & 启动模式的框图
(CR000: 捕捉寄存器、CR010: 捕捉寄存器)



注 使用 TI010 引脚的有效边沿检测时，不能使用定时器输出（TO00）。

图 6-29 输入 TI000 引脚有效边沿时的清除 & 启动模式的时序示例
(CR000: 捕捉寄存器、CR010: 捕捉寄存器) (1/3)

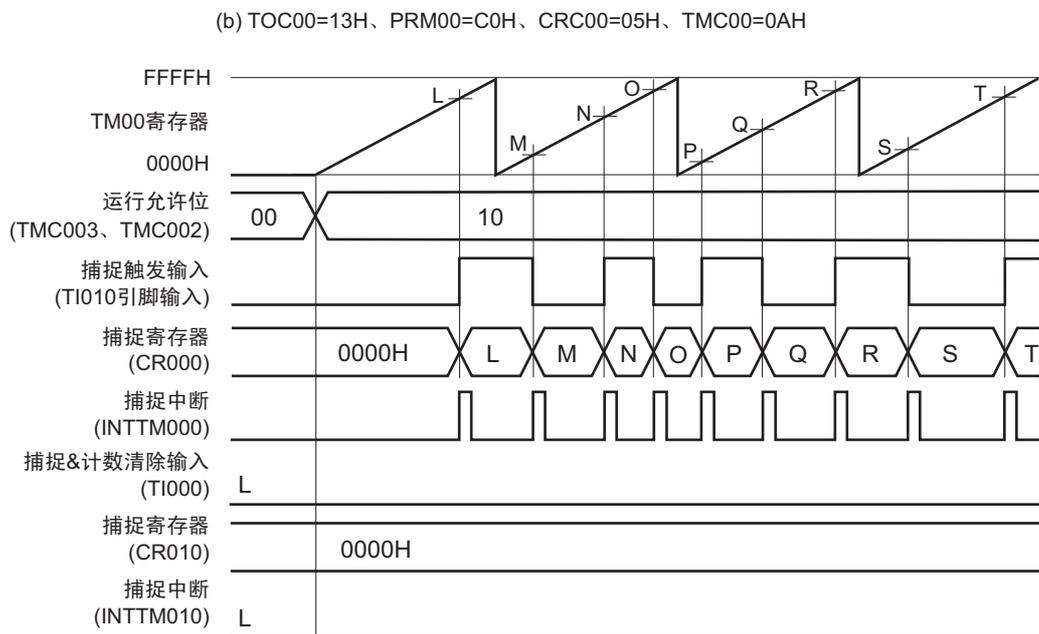
(a) TOC00=13H、PRM00=30H、CRC00=05H、TMC00=0AH



这是在检测到 TI000 引脚的上升沿或者下降沿时，捕捉计数值到 CR010，清除 TM00，并且取反 TO00 输出的应用示例。

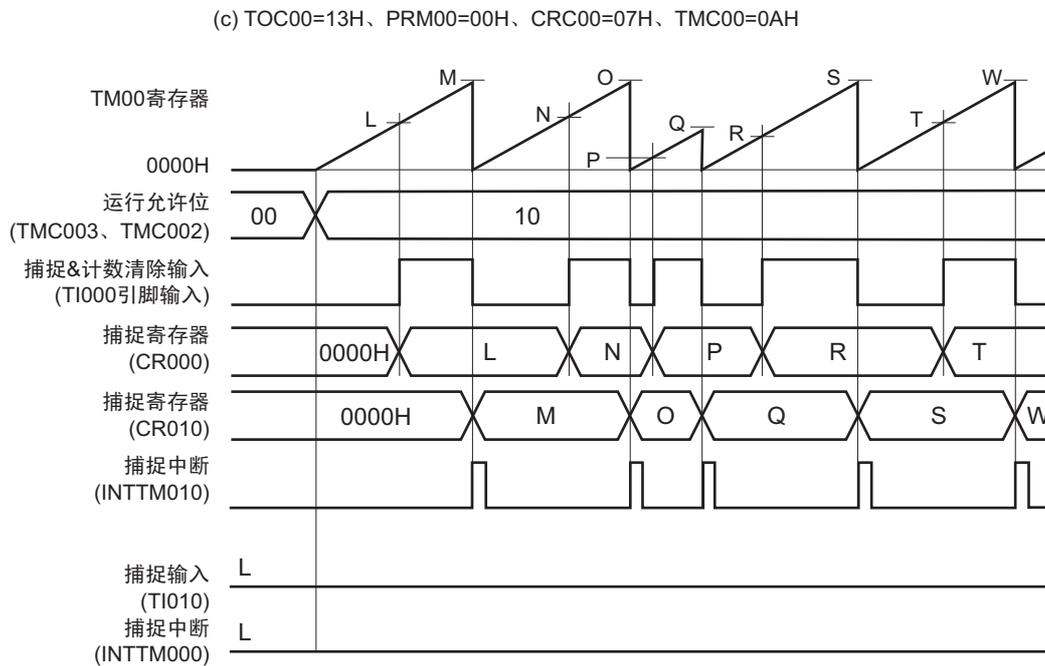
在检测到 TI010 引脚的边沿时，产生中断信号（INTTM000）。如果不使用 INTTM000 信号，必须将其屏蔽。

图 6-29 输入 TI000 引脚有效边沿时的清除 & 启动模式的时序示例
 (CR000: 捕捉寄存器、CR010: 捕捉寄存器) (2/3)



这是在检测出 TI010 引脚的上升沿或者下降沿的情况下，捕捉计数值到 CR000 的过程中，未输入边沿到 TI000 引脚时的时序示例。

图 6-29 输入 TI000 引脚有效边沿时的清除 & 启动模式的时序示例
(CR000: 捕捉寄存器、CR010: 捕捉寄存器) (3/3)



这是测量 TI000 引脚输入信号的脉宽时的应用示例。

通过设置 CRC00，在检测出 TI000 引脚的下降沿的反相（上升沿）时捕捉计数值到 CR000；而在检测出 TI000 引脚的下降沿时捕捉计数值到 CR010。

可通过以下表达式计算输入脉冲的高电平宽度和低电平宽度。

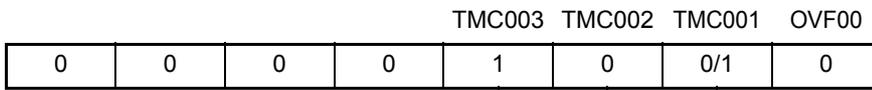
- 高电平宽度=[CR010值] - [CR000值] × [计数时钟周期]
- 低电平宽度=[CR000值] × [计数时钟周期]

在选择 TI000 引脚有效边沿的反相作为 CR000 的捕捉触发时，不产生 INTTM000 信号。必须在产生 INTTM010 信号后，立即读取用于测量脉宽的 CR000 和 CR010 的值。

但是，如果将预分频器模式寄存器 00（PRM00）的 bit6 和 bit5（ES110 和 ES100）指定的有效边沿输入到 TI010 引脚，就不进行捕捉运行，但是产生 INTTM000 信号。测量 TI000 引脚的脉宽时，如果不使用 INTTM000 信号，必须将其屏蔽。

图 6-30 输入 TI000 引脚有效边沿时的清除 & 启动模式运行期间的寄存器设置内容示例 (1/2)

(a) 16 位定时器模式控制寄存器 00 (TMC00)



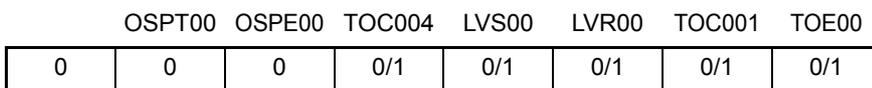
0: 在 TM00 和 CR000/CR010 匹配时, 取反 TO00 输出。
 1: 在 TM00 和 CR000/CR010 匹配以及 TI000 引脚的有效边沿检测时, 取反 TO00 输出。
 在输入 TI000 引脚的有效边沿时, 进行清除 & 启动

(b) 捕捉 / 比较控制寄存器 00 (CRC00)



0: 将 CR000 用作比较寄存器
 1: 将 CR000 用作捕捉寄存器
 0: CR000 的捕捉触发为 TI010 引脚
 1: CR000 的捕捉触发为 TI000 引脚有效边沿的反相
 0: 将 CR010 用作比较寄存器
 1: 将 CR010 用作捕捉寄存器

(c) 16 位定时器输出控制寄存器 00 (TOC00)

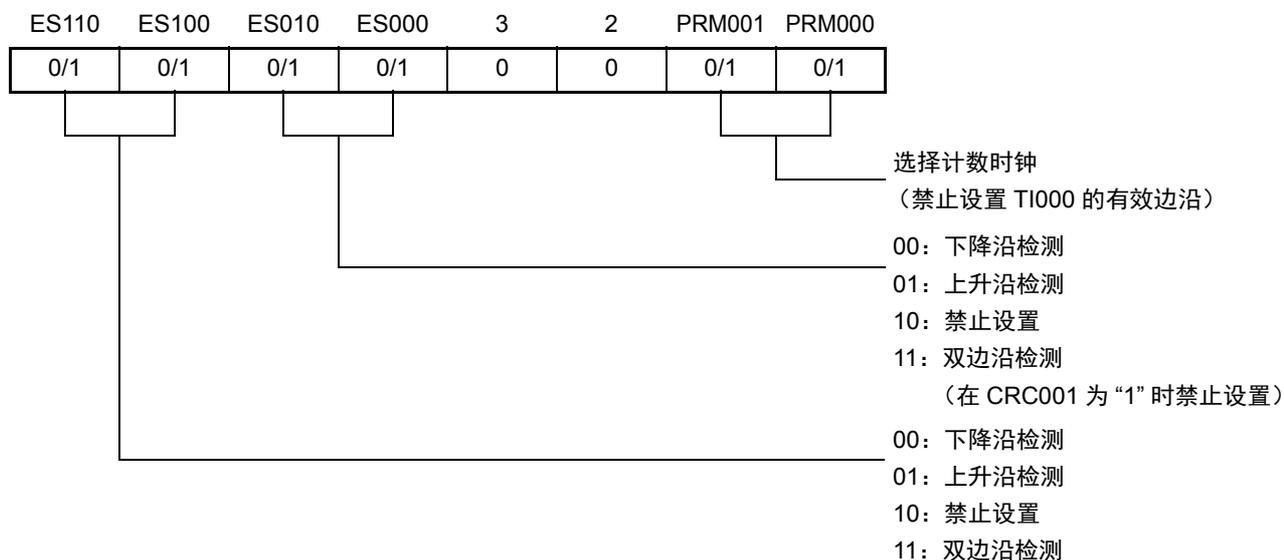


0: 禁止 TO00 输出注
 1: 允许 TO00 输出
 指定 TO00 输出 F/F 的初始值
 00: 即使 TM00 和 CR000/CR010 匹配, 也不取反 TO00 输出。
 01: 在 TM00 和 CR000 匹配时, 取反 TO00 输出。
 10: 在 TM00 和 CR010 匹配时, 取反 TO00 输出。
 11: 在 TM00 和 CR000/CR010 匹配时, 取反 TO00 输出。

注 使用 TI010 引脚的有效边沿检测时, 不能使用定时器输出 (TO00)。

图 6-30 输入 TI000 引脚有效边沿时的清除 & 启动模式运行期间的寄存器设置内容示例 (2/2)

(d) 预分频器模式寄存器 00 (PRM00)



(e) 16 位定时器计数器 00 (TM00)

通过读 TM00，读取计数器的值。

(f) 16 位捕捉 / 比较寄存器 000 (CR000)

用作比较寄存器时，在与 TM00 的值匹配时，产生中断信号 (INTTM000)。不清除 TM00 的计数值。

用作捕捉寄存器时，必须设置 TI000 引脚输入或者 TI010 引脚输入作为捕捉触发。在检测出捕捉触发的有效边沿时，将 TM00 的计数值保存到 CR000。

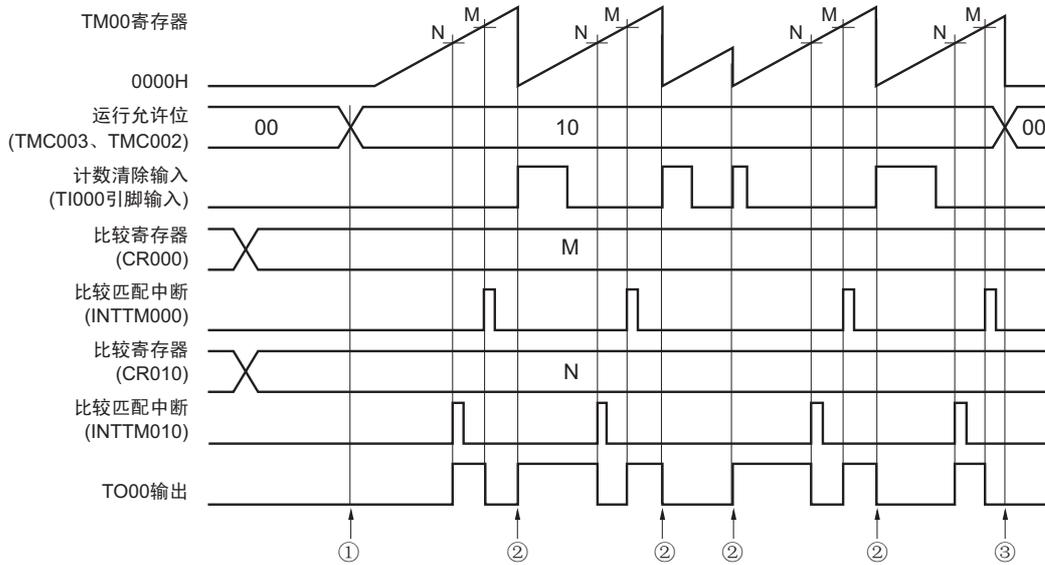
注 使用 TI010 引脚的有效边沿检测时，不能使用定时器输出 (TO00)。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

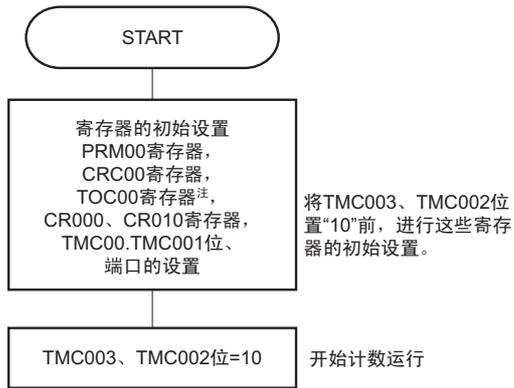
用作比较寄存器时，在与 TM00 的值匹配时，产生中断信号 (INTTM010)。不清除 TM00 的计数值。

用作捕捉寄存器时，TI000 引脚输入作为捕捉触发。在检测出捕捉触发的有效边沿时，将 TM00 的计数值保存到 CR010。

图 6-31 输入 TI000 引脚有效边沿时的清除 & 启动模式运行期间的软件处理示例



① 开始计数运行的流程



③ 停止计数运行的流程



② TM00寄存器清除&启动的流程



注 设置 TOC00 时，应特别注意。详细内容请参照“6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)”。

6.4.5 作为自由运行定时器的运行

如果将 16 位定时器模式控制寄存器 00 (TMC00) 的 bit3 和 bit2 (TMC003 和 TMC002) 置“01” (自由运行定时器模式), 就与计数时钟同步持续递增计数。一旦计数值为“FFFFH”, 就在下一个时钟将上溢标志 (OVF00) 置“1”的同时, 清除 TM00 (0000H), 并且持续计数。通过软件执行 CLR 指令, 将 OVF00 清“0”。

作为自由运行定时器的运行有以下 3 种。

- CR000 和 CR010 都用作比较寄存器
- CR000 和 CR010 中的一个用作比较寄存器, 另一个用作捕捉寄存器。
- CR000 和 CR010 都用作捕捉寄存器

备注 1. 有关输入 / 输出引脚的设置, 请参照“6.3 (5) 端口模式寄存器 0 (PM0)”。

2. 有关 INTTM000 信号的中断允许, 请参照“第 16 章 中断功能”。

(1) 自由运行定时器模式的运行

(CR000: 比较寄存器、CR010: 比较寄存器)

图 6-32 自由运行定时器模式的框图
(CR000: 比较寄存器、CR010: 比较寄存器)

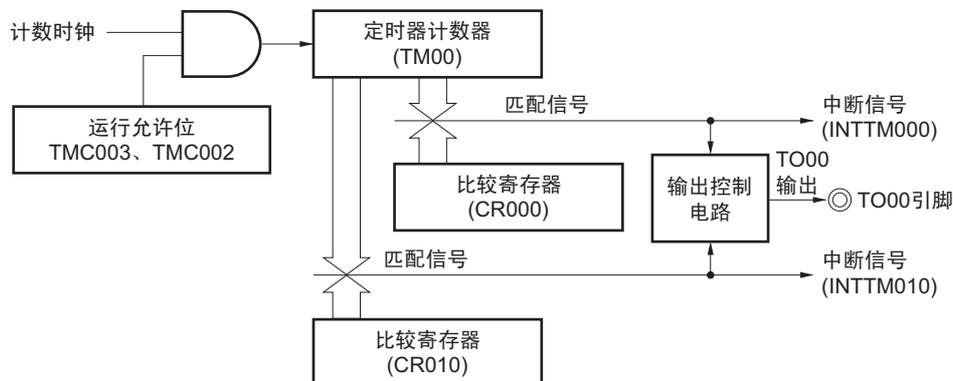
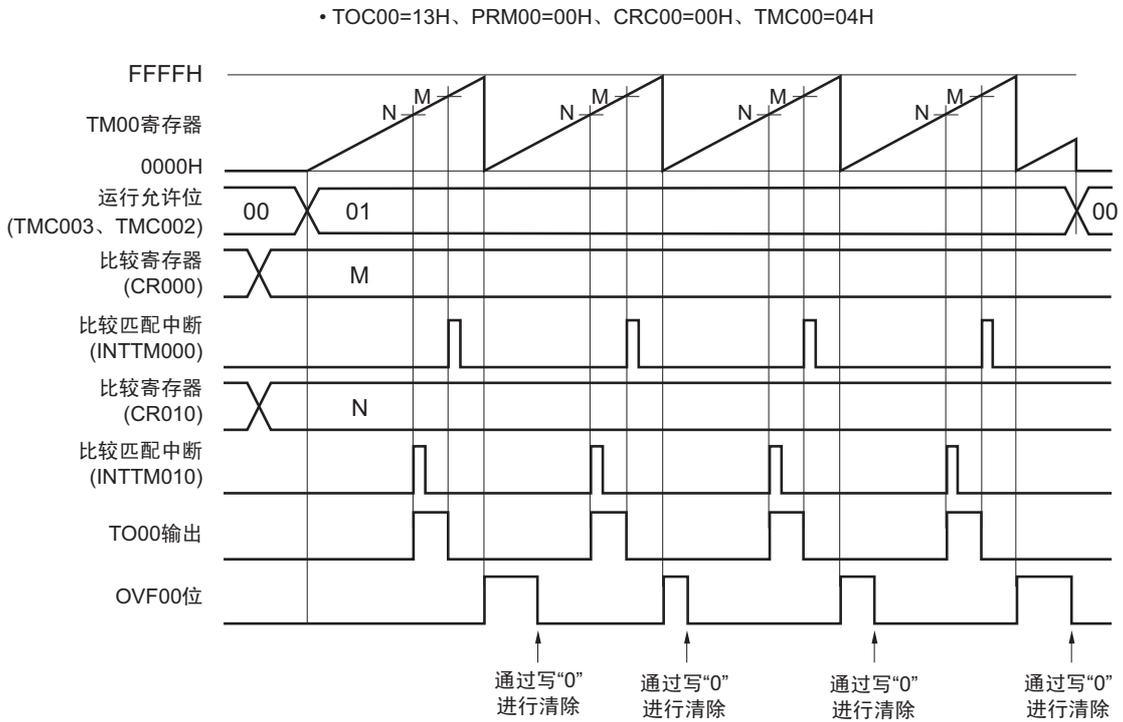


图 6-33 自由运行定时器模式的时序示例
(CR000: 比较寄存器、CR010: 比较寄存器)



这是在自由运行定时器模式中使用 2 个比较寄存器的应用示例。

每当 TM00 的计数值与 CR000、CR010 的设置值匹配时，取反 TO00 输出电平，并且分别产生 INTTM000 信号和 INTTM010 信号。

(2) 自由运行定时器模式的运行
(CR000: 比较寄存器、CR010: 捕捉寄存器)

图 6-34 自由运行定时器模式的框图
(CR000: 比较寄存器、CR010: 捕捉寄存器)

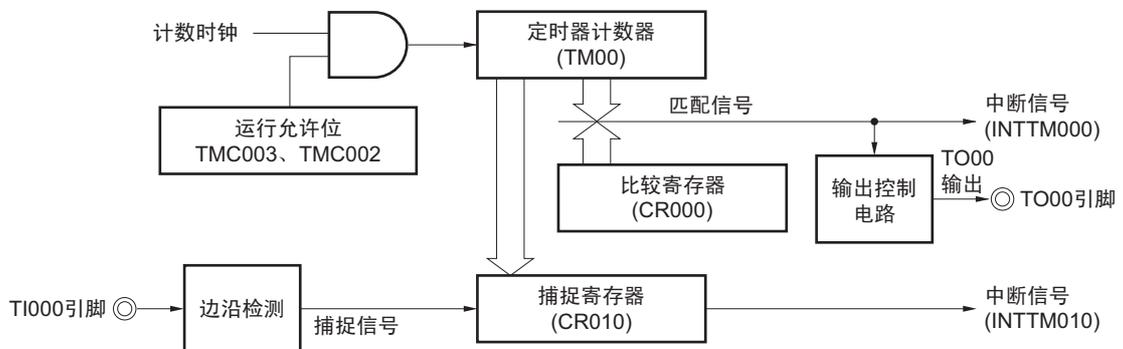
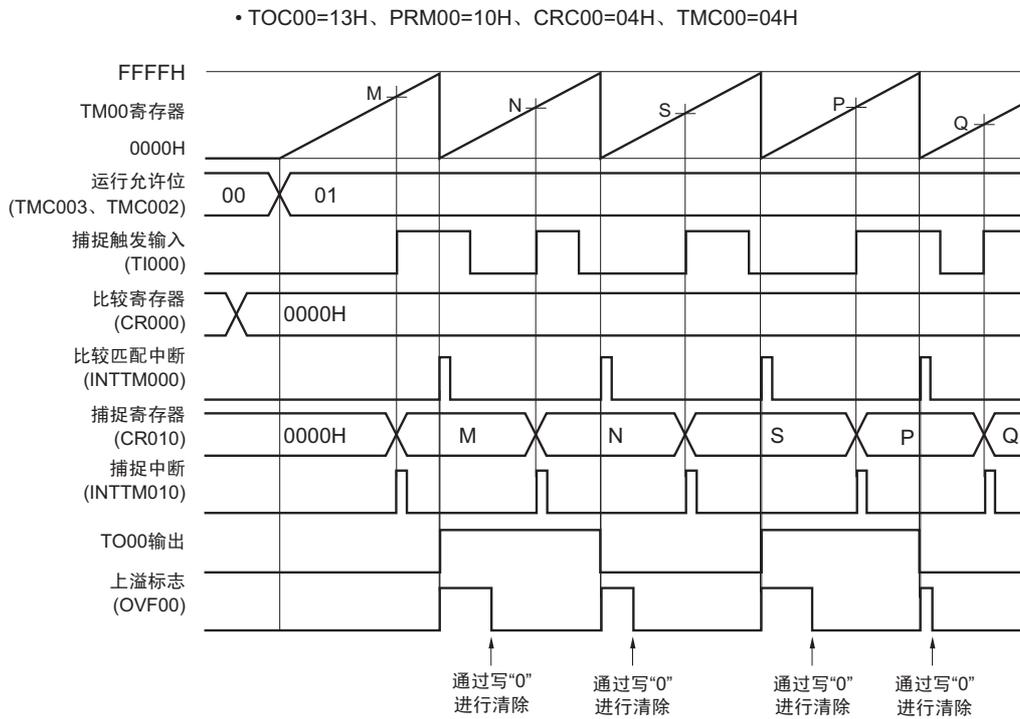


图 6-35 自由运行定时器模式的时序示例
(CR000: 比较寄存器、CR010: 捕捉寄存器)

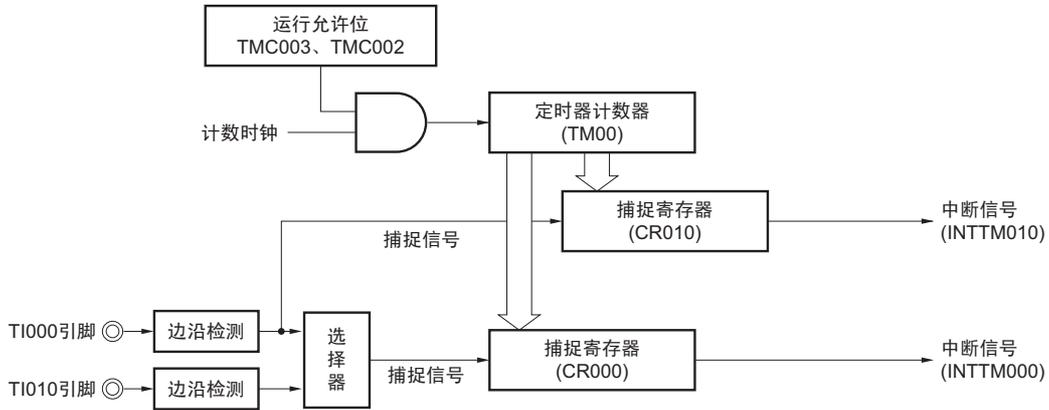


这是在自由运行定时器模式中分别用作比较寄存器和捕捉寄存器的应用示例。

在此例中，每当 TM00 的计数值和 CR000（比较寄存器）的设置值匹配时，产生 INTTM000 信号，并且取反 TO00 输出。另外，每当检测出 TI000 引脚的有效边沿，就产生 INTTM010 信号，并且捕捉 TM00 的计数值到 CR010。

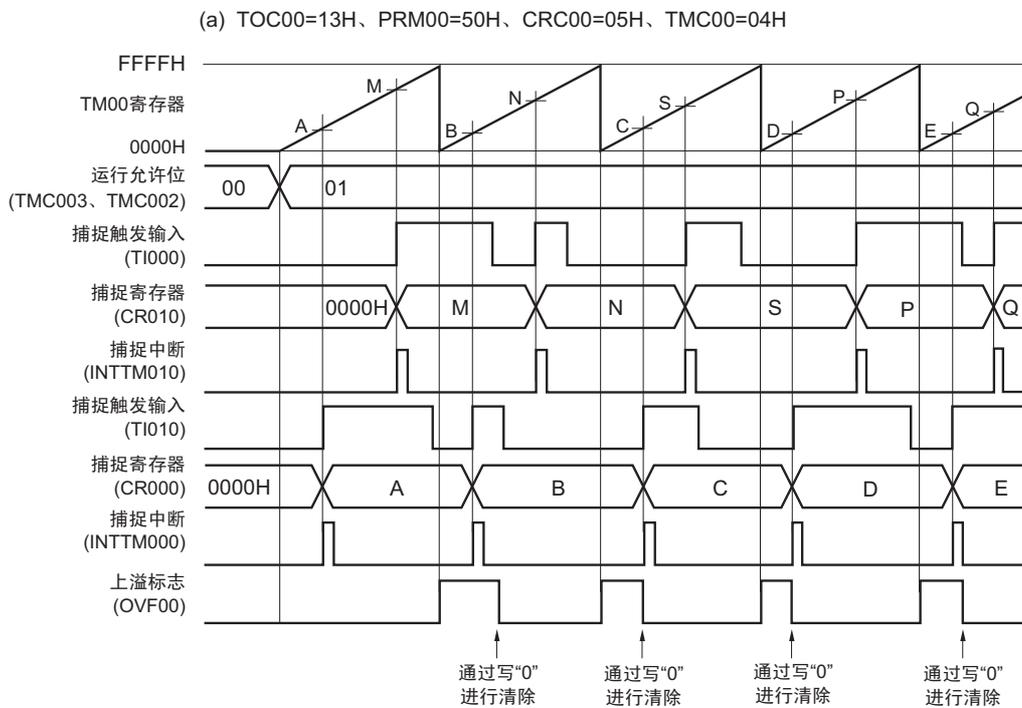
(3) 自由运行定时器模式的运行
(CR000: 捕捉寄存器、CR010: 捕捉寄存器)

图 6-36 自由运行定时器模式的框图
(CR000: 捕捉寄存器、CR010: 捕捉寄存器)



备注 在自由运行定时器模式中将 CR000 和 CR010 都用作捕捉寄存器时，不取反 TO00 输出电平。但是，通过将 16 位定时器模式控制寄存器 00 (TMC00) 的 bit1 (TMC001) 置“1”，每检测出 TI000 引脚的有效边沿，就可取反 TO00 输出电平。

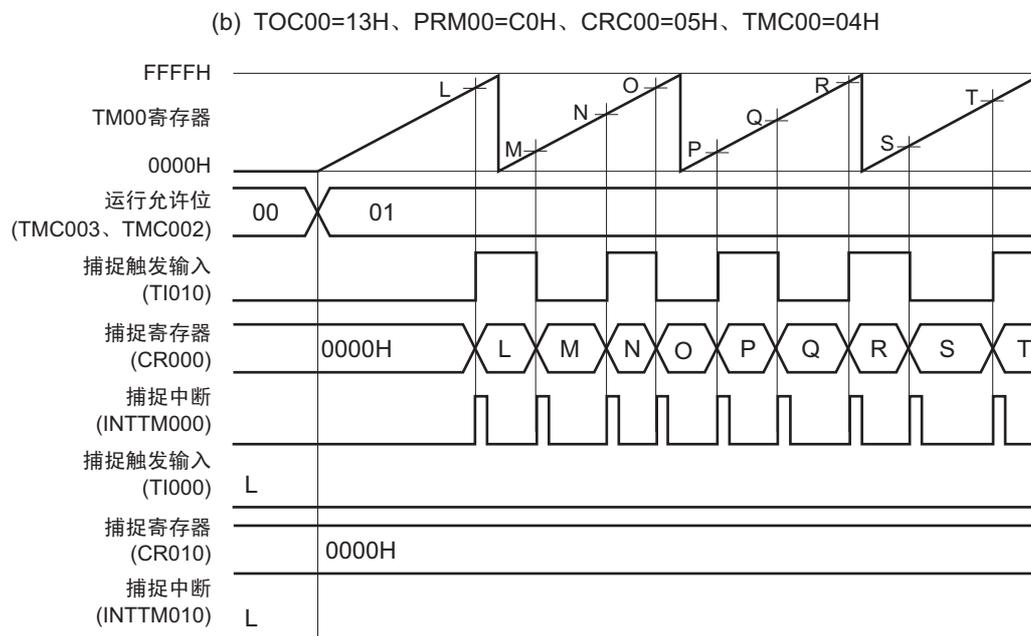
图 6-37 自由运行定时器模式的时序示例
(CR000: 捕捉寄存器、CR010: 捕捉寄存器) (1/2)



这是在自由运行定时器模式中将在各捕捉触发输入的有效边沿捕捉的计数值保存到各捕捉寄存器的应用示例。

在检测出 TI000 引脚输入的有效边沿时，捕捉计数值到 CR010；在检测出 TI010 引脚输入的有效边沿时，捕捉计数值到 CR000。

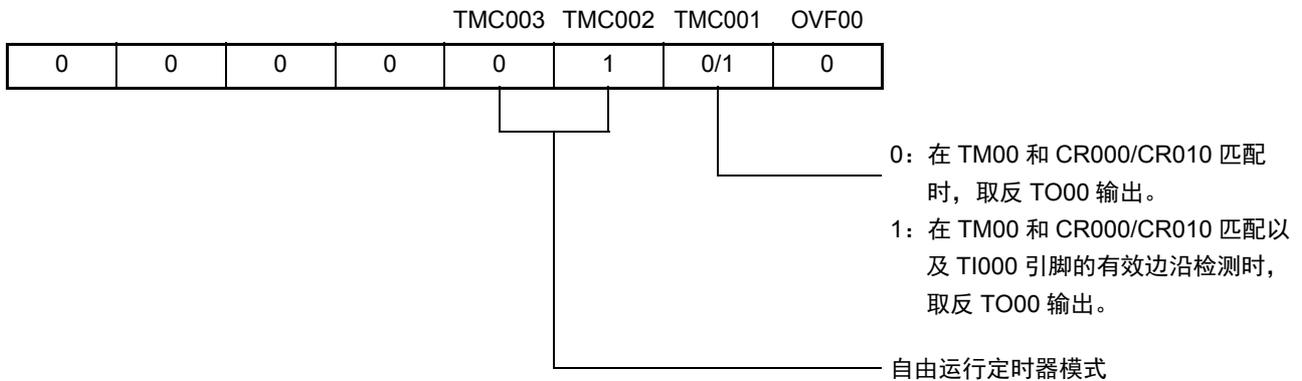
图 6-37 自由运行定时器模式的时序示例
(CR000: 捕捉寄存器、CR010: 捕捉寄存器) (2/2)



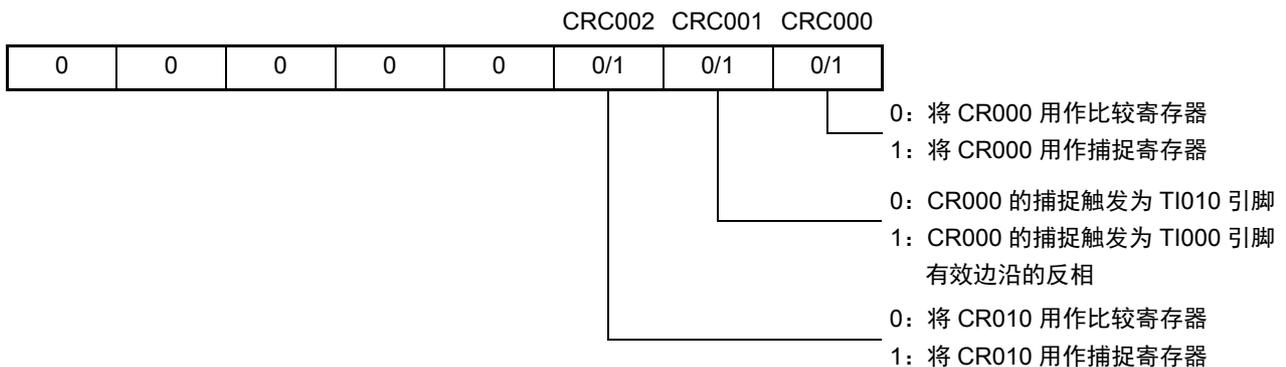
这是在自由运行定时器模式中设置为 TI010 引脚的双边沿检测，并且捕捉计数值到 CR000 的应用示例。CR000 和 CR010 都用作捕捉寄存器并且只能检测出 TI010 引脚的有效边沿时，不能捕捉计数值到 CR010。

图 6-38 自由运行定时器模式运行期间的寄存器设置内容示例 (1/2)

(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕捉 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)

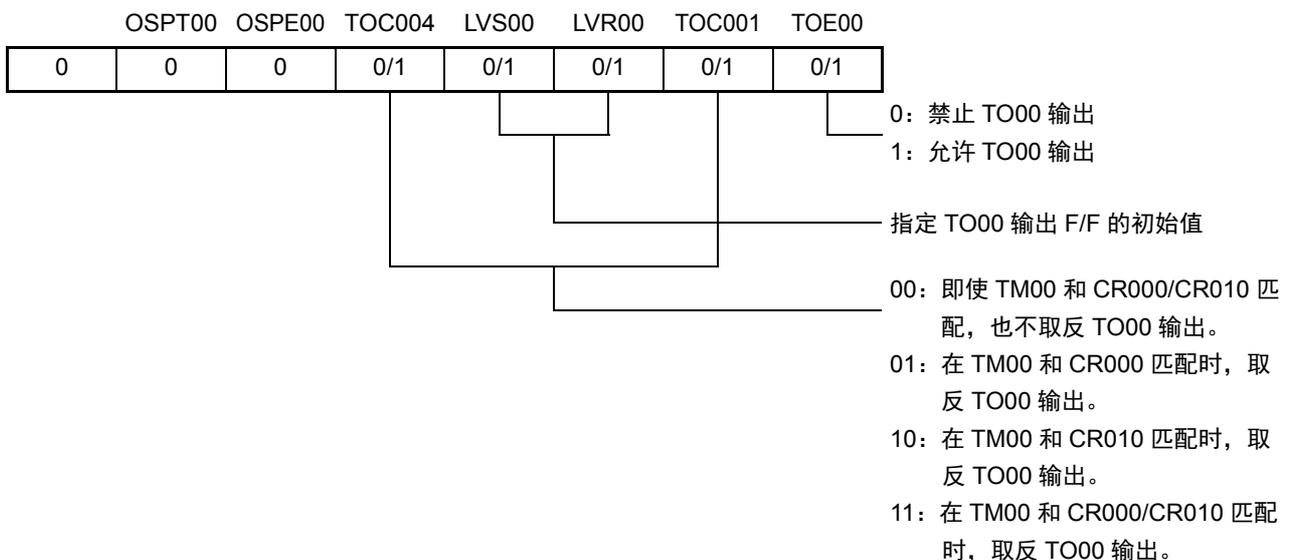
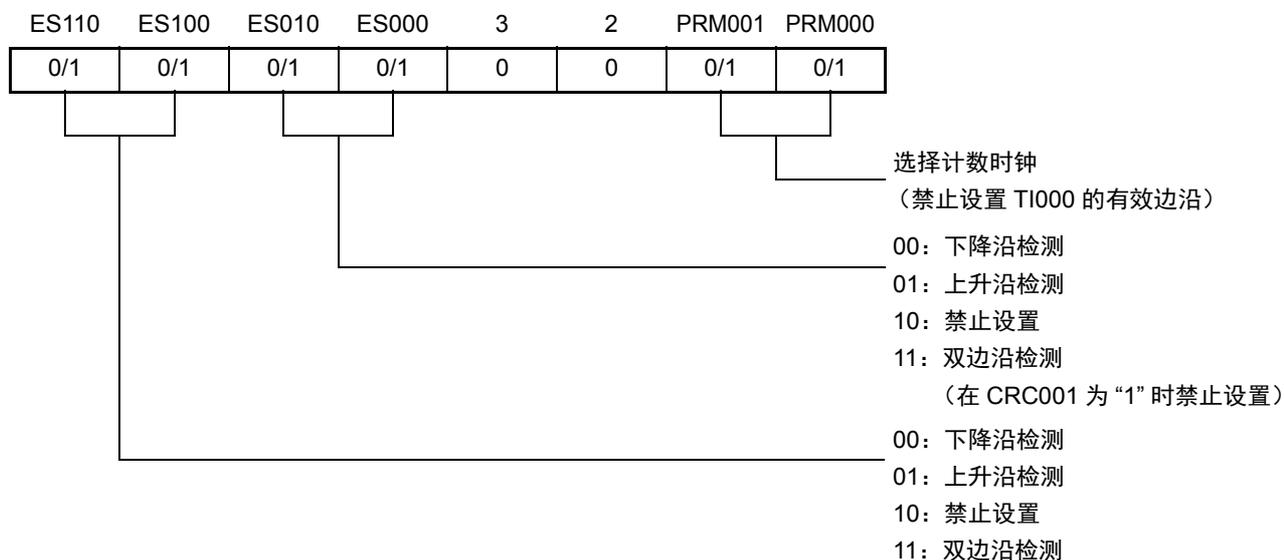


图 6-38 自由运行定时器模式运行期间的寄存器设置内容示例 (2/2)

(d) 预分频模式寄存器 00 (PRM00)



(e) 16 位定时器计数器 00 (TM00)

通过读 TM00，读取计数器的值。

(f) 16 位捕捉 / 比较寄存器 000 (CR000)

用作比较寄存器时，在与 TM00 的值匹配时，产生中断信号 (INTTM000)。不清除 TM00 的计数值。

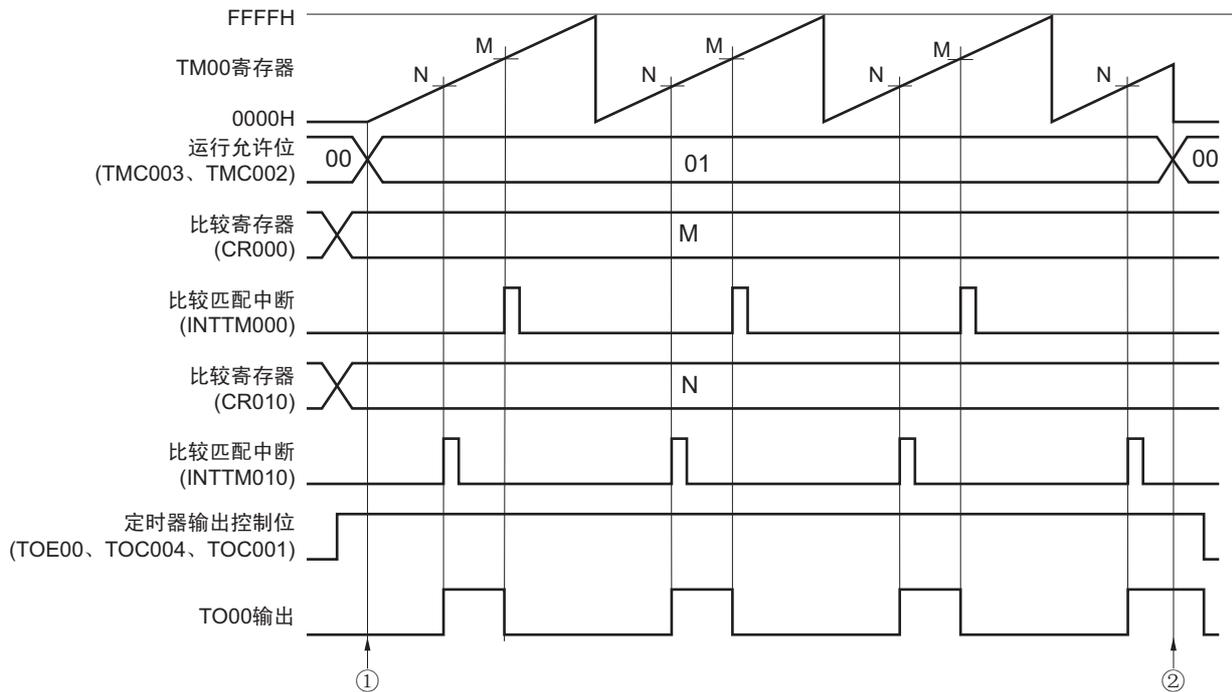
用作捕捉寄存器时，必须设置 TI000 引脚输入或者 TI010 引脚输入作为捕捉触发。在检测出捕捉触发的有效边沿时，将 TM00 的计数值保存到 CR000。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

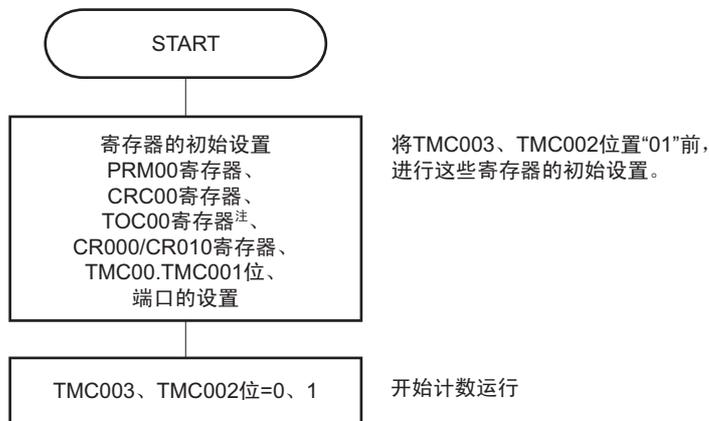
用作比较寄存器时，在与 TM00 的值匹配时，产生中断信号 (INTTM010)。不清除 TM00 的计数值。

用作捕捉寄存器时，TI000 引脚输入作为捕捉触发。在检测出捕捉触发的有效边沿时，将 TM00 的计数值保存到 CR010。

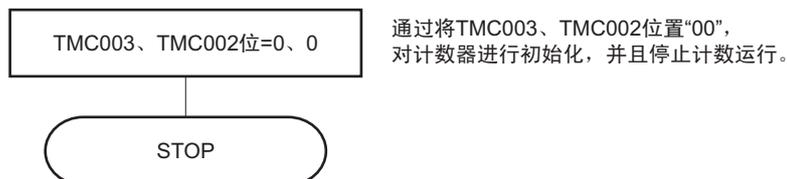
图 6-39 自由运行定时器模式运行期间的软件处理示例



① 开始计数运行的流程



② 停止计数运行的流程



注 设置 TOC00 时，应特别注意。详细内容请参照“6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)”。

6.4.6 PPG 的输出运行

将 16 位定时器模式控制寄存器 00 (TMC00) 的 bit3 和 bit2 (TMC003 和 TMC002) 置“11” (在 TM00 和 CR000 相匹配时进行清除 & 启动)，从 TO00 引脚输出周期和脉宽分别为事先设置的 CR000 值和 CR010 值的方波作为 PPG (Programmable Pulse Generator) 的输出运行。

通过 PPG 输出产生的脉冲周期和占空比如下所示。

- 脉冲周期 = (CR000 的设置值 + 1) × 计数时钟周期
- 占空比 = (CR010 的设置值 + 1) / (CR000 的设置值 + 1)

注意 有关运行期间占空比 (CR010 值) 的更改方法, 请参照“6.5.1 在 TM00 运行期间改写 CR010”。

备注 1. 有关输入 / 输出引脚的设置, 请参照“6.3 (5) 端口模式寄存器 0 (PM0)”。

2. 有关 INTTM000 信号的中断允许, 请参照“第 16 章 中断功能”。

图 6-40 PPG 输出运行的框图

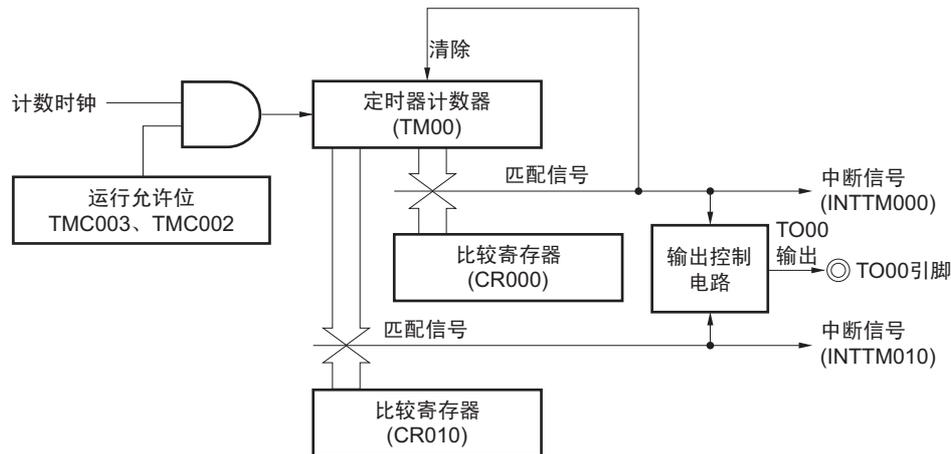
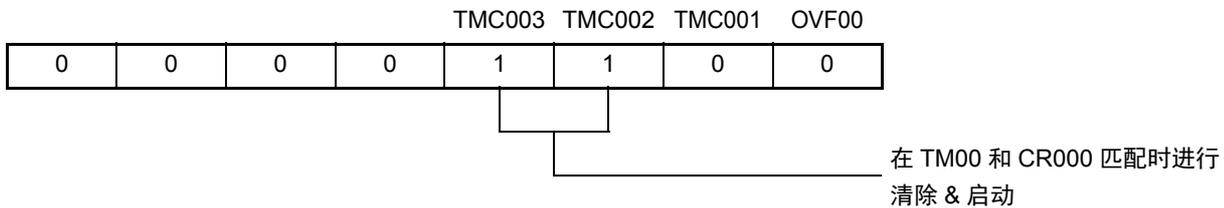
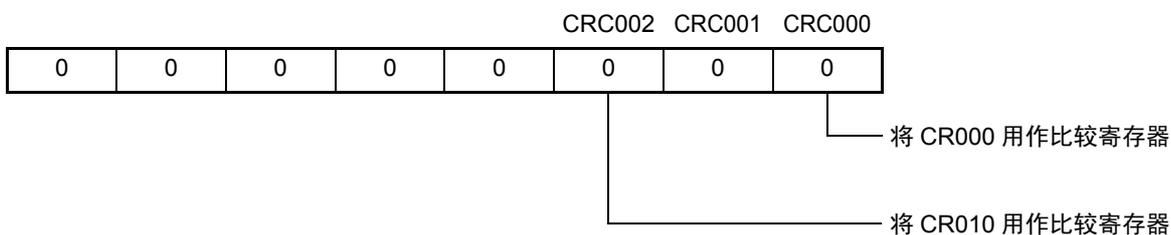


图 6-41 PPG 输出运行时的寄存器设置内容示例 (1/2)

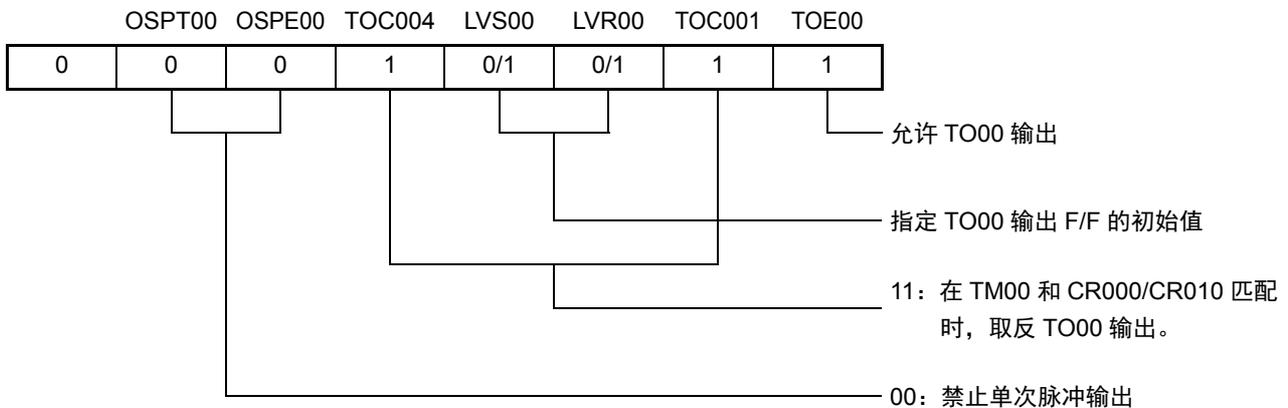
(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕捉 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频模式寄存器 00 (PRM00)

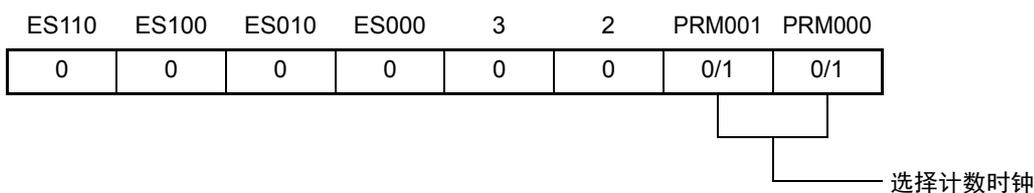


图 6-41 PPG 输出运行时的寄存器设置内容示例 (2/2)

(e) 16 位定时器计数器 00 (TM00)

通过读 TM00，读取计数器的值。

(f) 16 位捕捉 / 比较寄存器 000 (CR000)

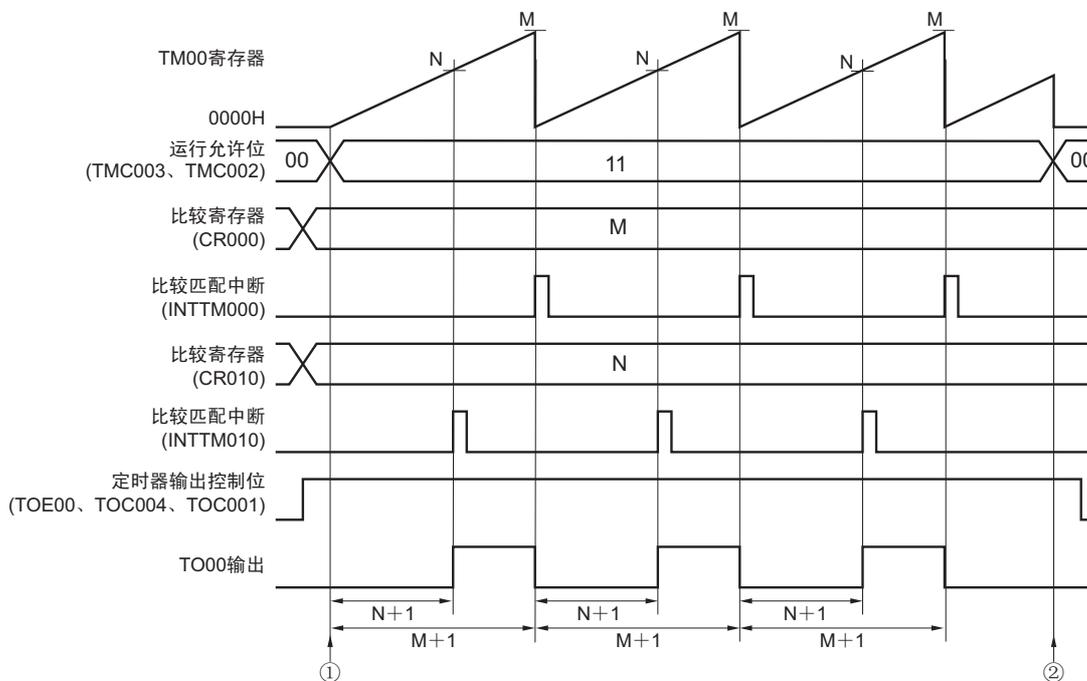
在与 TM00 匹配时，产生中断信号 (INTTM000)。不清除 TM00 的计数值。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

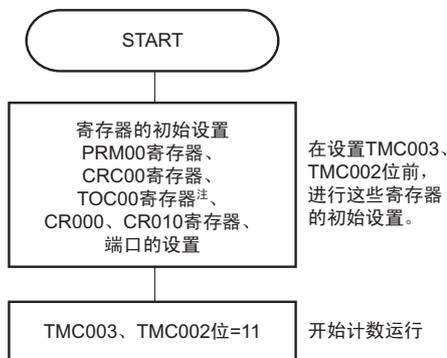
在与 TM00 匹配时，产生中断信号 (INTTM010)。不清除 TM00 的计数值。

注意 必须给 CR000 和 CR010 设置满足 $0000H \leq CR010 < CR000 \leq FFFFH$ 的值。

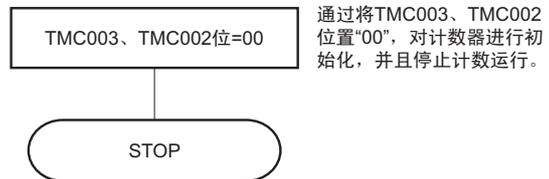
图 6-42 PPG 输出运行时的软件处理示例



① 开始计数运行的流程



② 停止计数运行的流程



注 设置 TOC00 时，应特别注意。详细内容请参照“6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)”。

注意 PPG 的脉冲周期 $= (M+1) \times$ 计数时钟周期

PPG 的占空比 $= (N+1)/(M+1)$

6.4.7 单次脉冲的输出运行

可通过将 16 位定时器模式控制寄存器 00 (TMC00) 的 bit3 和 bit2 (TMC003 和 TMC002) 置“01” (自由运行定时器模式) 或者置“10” (输入 TI000 引脚有效边沿时的清除 & 启动模式), 将 16 位定时器输出控制寄存器 00 (TOC00) 的 bit5 (OSPE00) 置“1”, 输出单次脉冲。

如果在定时器运行期间将 TOC00 的 bit6 (OSPT00) 置“1” 或者输入有效边沿到 TI000 引脚, 就被作为触发, 并且在进行 TM00 的清除 & 启动后, 仅从 TO00 引脚输出 1 次 CR000 和 CR010 的设置值差的脉冲。

注意 1. 在输出单次脉冲期间, 不能输入触发 (将 OSPT00 置“1” 或者 TI000 引脚的有效边沿检测)。只有在当前的单次脉冲输出结束后产生触发, 才可再次输出单次脉冲。

2. 仅将 OSPT00 置“1” 作为单次脉冲输出的触发时, 不能更改 TI000 引脚或者其复用端口引脚的电平。否则, 会在意想不到的时序输出脉冲。

备注 1. 有关输入 / 输出引脚的设置, 请参照“6.3 (5) 端口模式寄存器 0 (PM0)”。

2. 有关 INTTM000 信号的中断允许, 请参照“第 16 章 中断功能”。

图 6-43 单次脉冲输出运行的框图

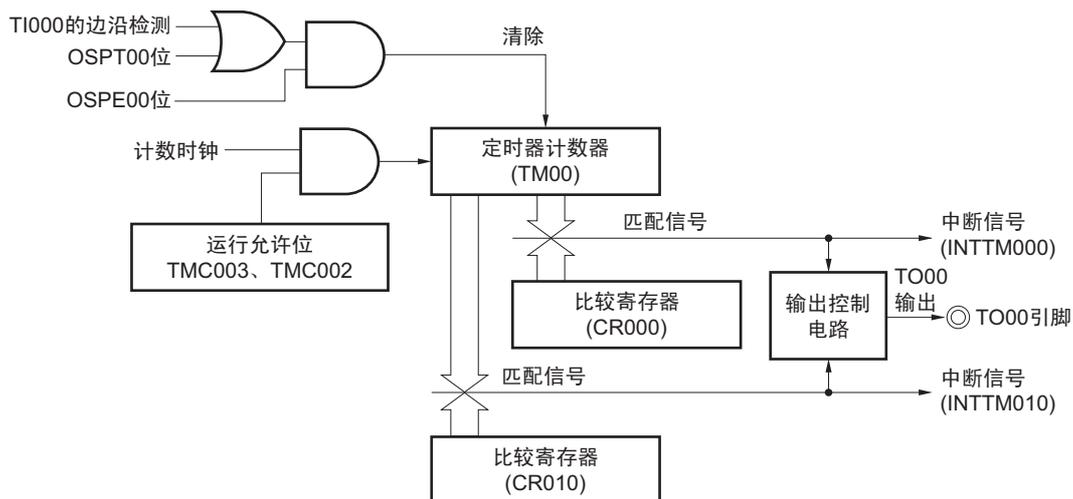
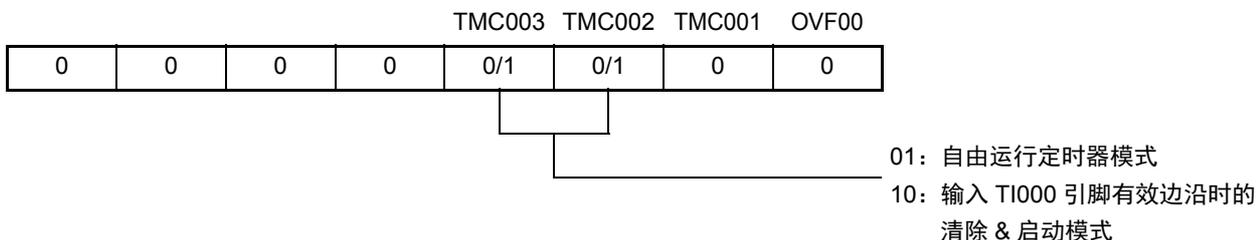
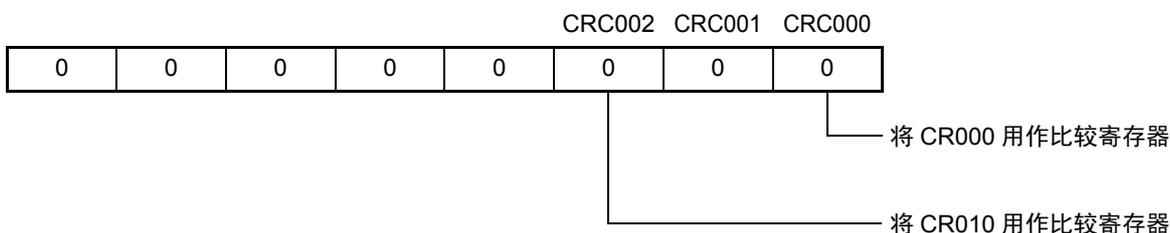


图 6-44 单次脉冲输出运行时的寄存器设置内容示例 (1/2)

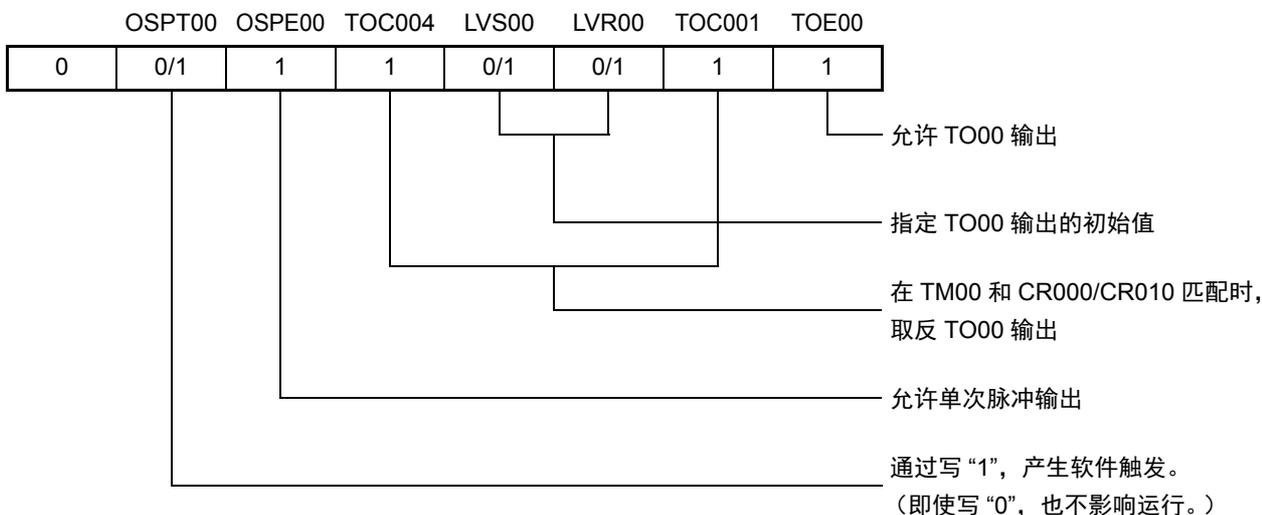
(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕捉 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频模式寄存器 00 (PRM00)

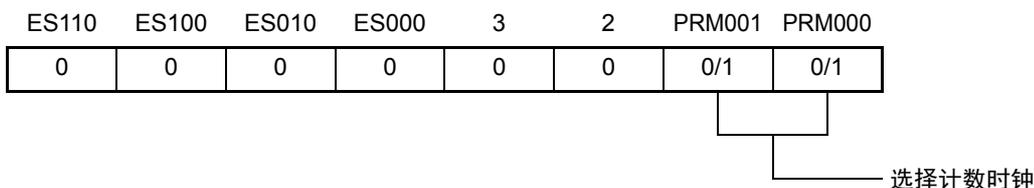


图 6-44 单次脉冲输出运行时的寄存器设置示例 (2/2)

(e) 16 位定时器计数器 00 (TM00)

通过读 TM00，读取计数器的值。

(f) 16 位捕捉 / 比较寄存器 000 (CR000)

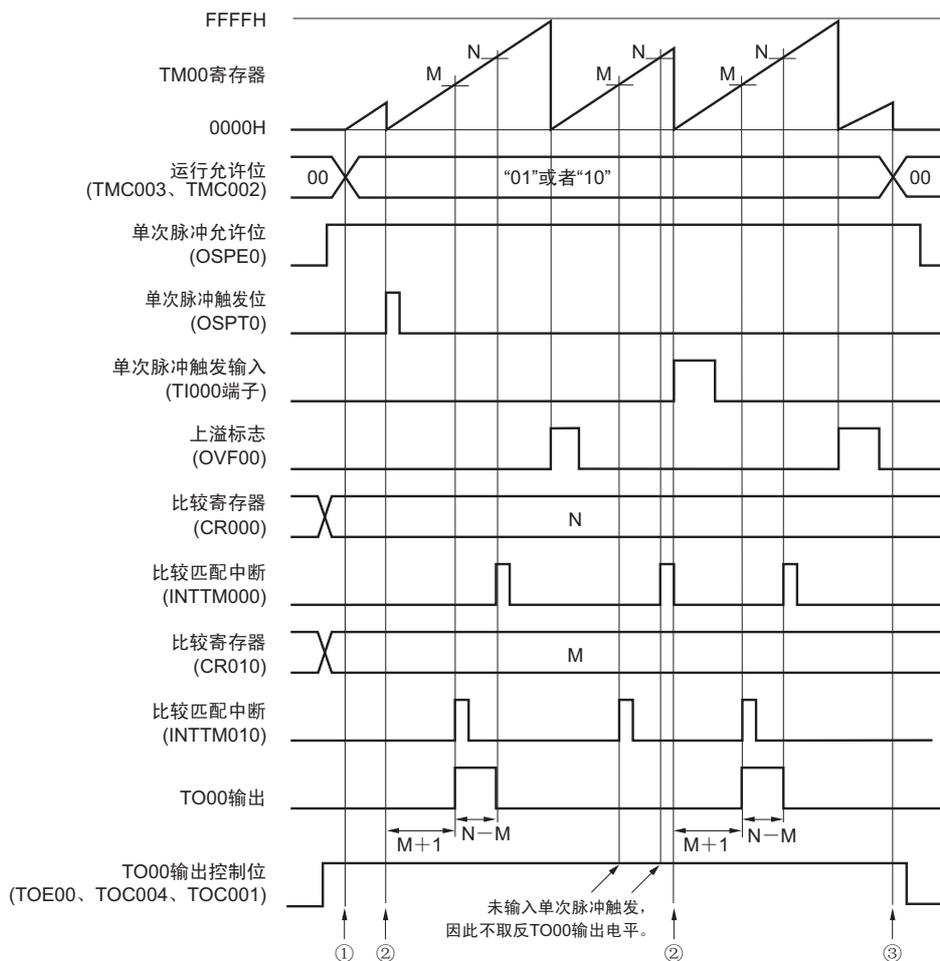
用作单次脉冲输出的比较寄存器。一旦 TM00 的值与 CR000 的值匹配，就产生中断信号 (INTTM000)，并且取反 TO00 输出电平。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

用作单次脉冲输出的比较寄存器。一旦 TM00 的值与 CR010 的值匹配，就产生中断信号 (INTTM010)，并且取反 TO00 输出电平。

注意 不能将 CR000 和 CR010 设置为相同的值。

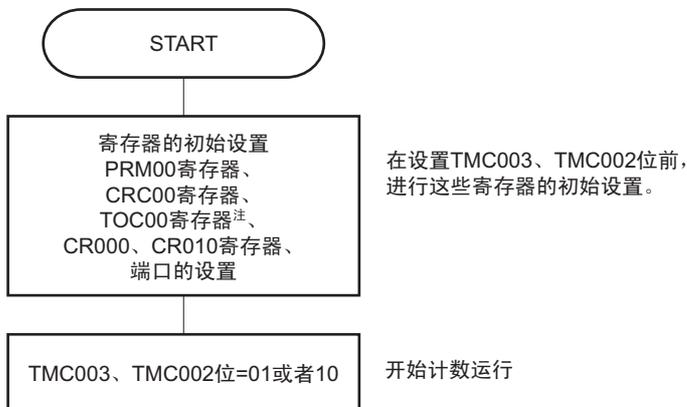
图 6-45 单次脉冲输出运行时的软件处理示例 (1/2)



- 从输入单次脉冲触发到输出单次脉冲的时间
 $= (M+1) \times \text{计数时钟周期}$
- 单次脉冲输出有效电平宽度
 $= (N-M) \times \text{计数时钟周期}$

图 6-45 单次脉冲输出运行时的软件处理示例 (2/2)

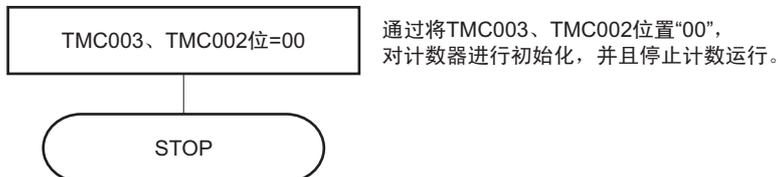
① 开始计数运行的流程



② 单次脉冲触发的输入流程



③ 停止计数运行的流程



注 设置 TOC00 时，应特别注意。详细内容请参照“6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)”。

6.4.8 脉冲的测量运行

使用 TM00 测量输入到 TI000 和 TI010 引脚的信号的脉宽。

可通过在自由运行定时器模式中运行 16 位定时器 / 事件计数器 00 来进行测量，也可通过与输入到 TI000 引脚的信号的边沿同步重新启动定时器来进行测量。

发生中断后，读取有效捕捉寄存器的值，并且测量脉宽。确认 16 位定时器模式控制寄存器 00（TMC00）的 bit0（OVF00），如果被置“1”，就必须通过软件清“0”。

图 6-46 脉宽测量（自由运行定时器模式）的框图

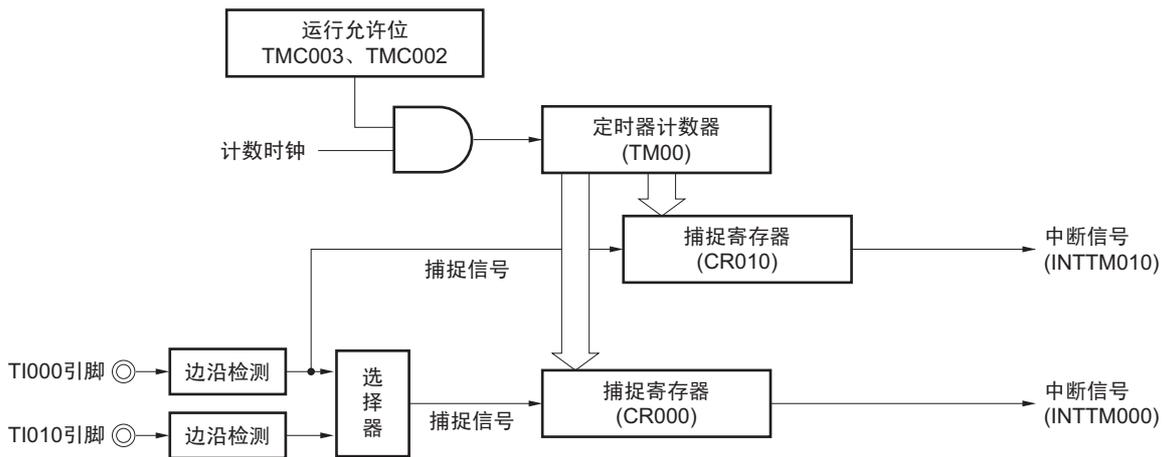
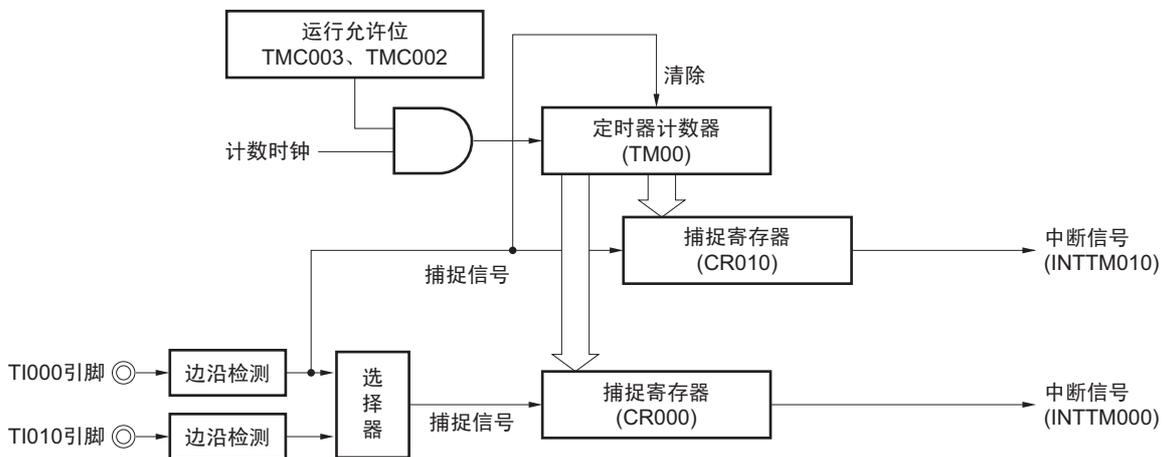


图 6-47 脉宽测量（输入 TI000 引脚有效边沿时的清除 & 启动模式）的框图



通过以下三种方法测量脉宽。

- 通过 TI000 和 TI010 引脚的 2 个输入信号测量脉宽（自由运行定时器模式）
- 通过 TI000 引脚的 1 个输入信号测量脉宽（自由运行定时器模式）
- 通过 TI000 引脚的 1 个输入信号测量脉宽（输入 TI000 引脚有效边沿时的清除 & 启动模式）

备注 1. 有关输入 / 输出引脚的设置，请参照“6.3 (5) 端口模式寄存器 0（PM0）”。

2. 有关 INTTM000 信号的中断允许，请参照“第 16 章 中断功能”。

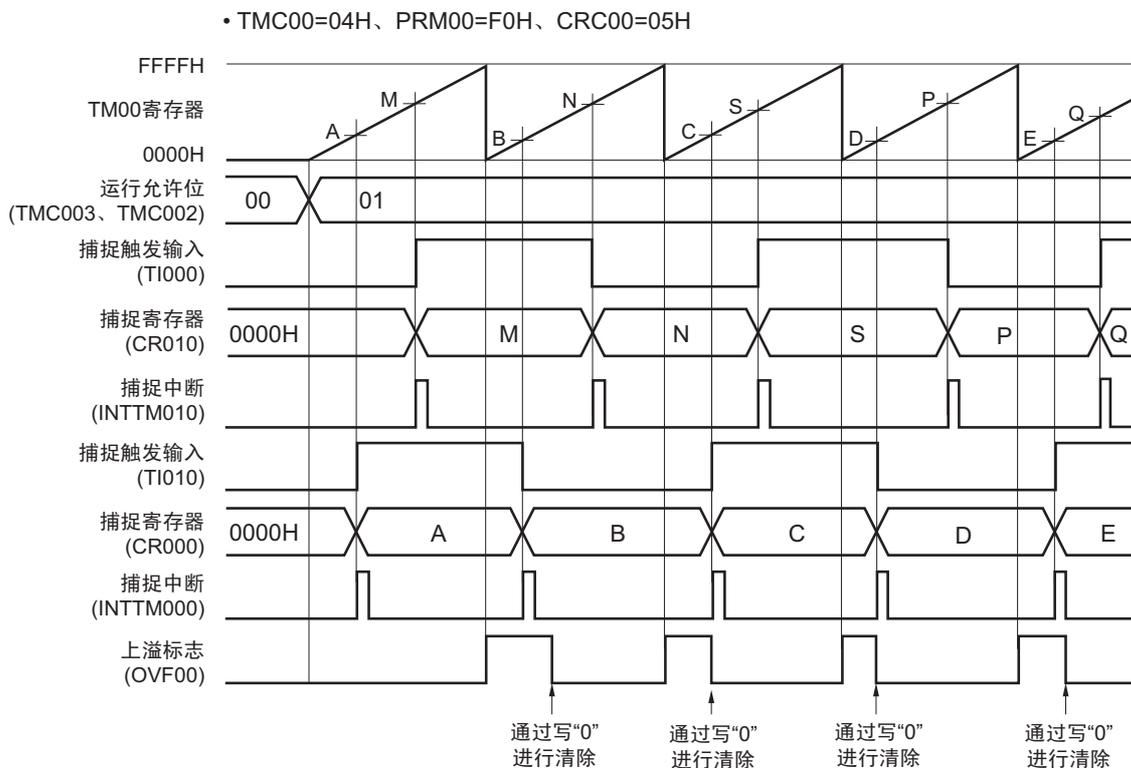
(1) 通过 TI000 与 TI010 引脚的 2 个输入信号测量脉宽（自由运行定时器模式）

设置为自由运行定时器模式（TMC003 和 TMC002=01）。在检测出 TI000 引脚的有效边沿时，捕捉 TM00 的计数值到 CR010；在检测出 TI010 引脚的有效边沿时，捕捉 TM00 的计数值到 CR000。TI000 引脚和 TI010 引脚的边沿检测设置为双边沿。

该测量方法是从各输入信号边沿的捕捉值中减去上一次的捕捉值。因此，必须事先将上一次的捕捉值保存到其他寄存器。

发生上溢时，如果只进行单纯的减法运算，其结果为负，因此，会发生借位（程序状态字（PSW）的 bit0（CY）置“1”）。此时，必须忽略 CY，并且将计算值作为脉宽处理。另外，将 16 位定时器模式控制寄存器 00（TMC00）的 bit0（OVF00）清“0”。

图 6-48 测量脉宽的时序示例（1）



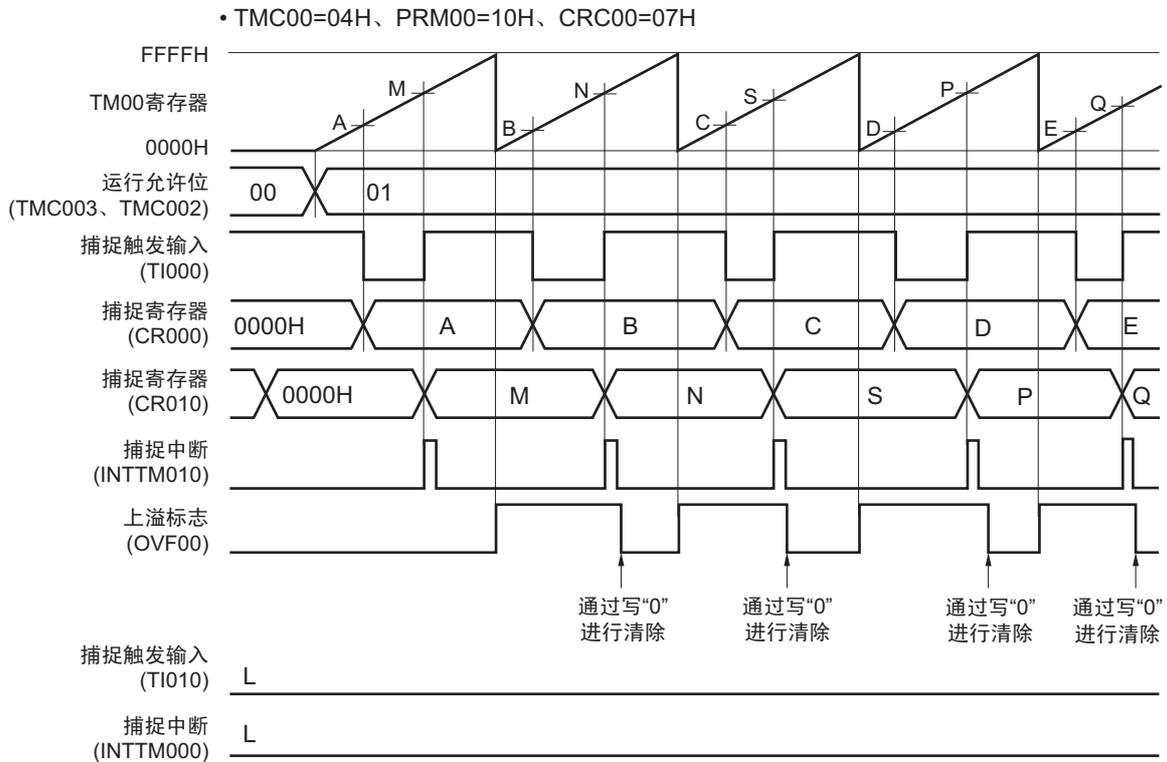
(2) 通过 TI000 引脚的 1 个输入信号测量脉宽 (自由运行定时器模式)

设置为自由运行定时器模式 (TMC003 和 TMC002=01)。在检测出 TI000 引脚的有效边沿的反相时, 捕捉 TM00 的计数值到 CR000; 在检测出 TI000 引脚的有效边沿时, 捕捉 TM00 的计数值到 CR010。

该测量方法在测量边沿与边沿间的宽度时保存捕捉值到各捕捉寄存器, 因此无需保存捕捉值。通过对 2 个捕捉寄存器的值进行减法运算, 算出高电平宽度、低电平宽度以及周期。

发生上溢时, 如果只进行单纯的减法运算, 其结果为负, 因此, 会发生借位 (程序状态字 (PSW) 的 bit0 (CY) 置“1”)。此时, 必须忽略 CY, 并且将计算值作为脉宽处理。另外, 将 16 位定时器模式控制寄存器 00 (TMC00) 的 bit0 (OVF00) 清“0”。

图 6-49 测量脉宽的时序示例 (2)

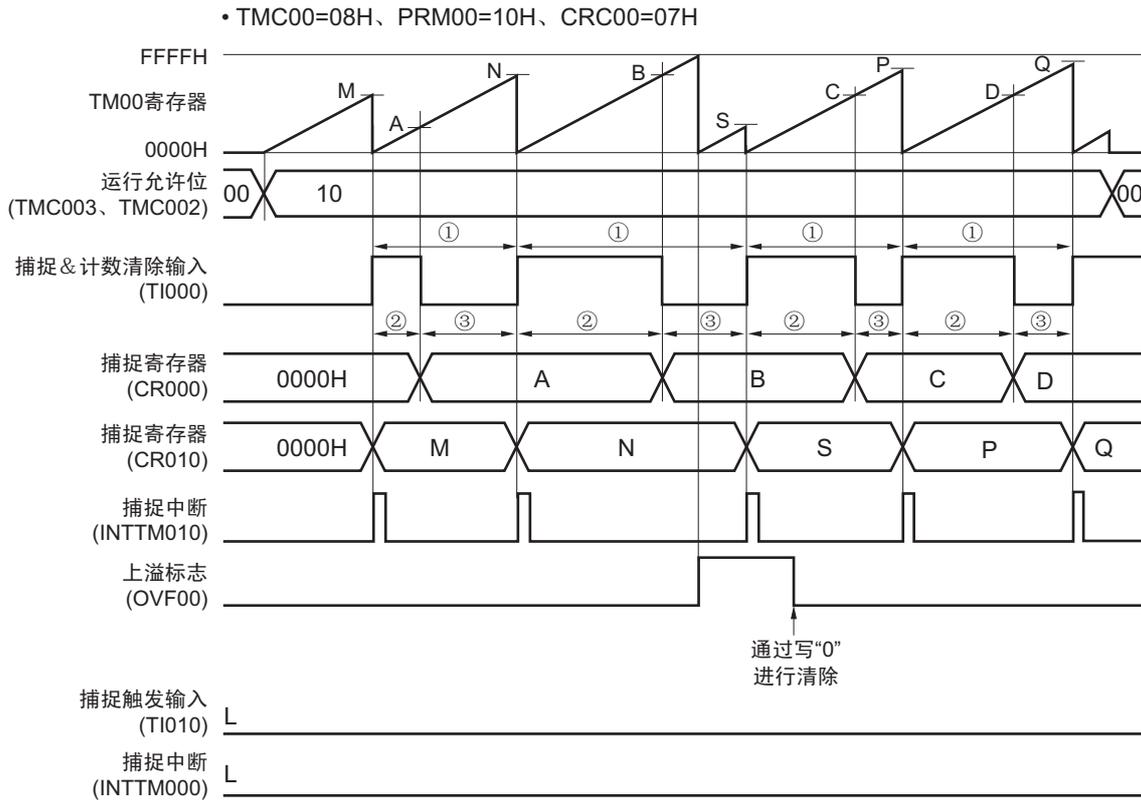


(3) 通过 TI000 引脚的 1 个输入信号测量脉宽 (输入 TI000 引脚有效边沿时的清除 & 启动模式)

设置为输入 TI000 引脚有效边沿时的清除 & 启动模式 (TMC003 和 TMC002=10)。在检测出 TI000 引脚的有效边沿的反相时, 捕捉 TM00 的计数值到 CR000; 在检测出 TI000 引脚的有效边沿时, 捕捉 TM00 的计数值到 CR010, 并且清除 TM00 (0000H)。因此, 只要 TM00 不发生上溢, 就保存周期到 CR010。

如果发生上溢, 就将 CR010 的保存值与“10000H”相加后的值作为周期处理。另外, 将 16 位定时器模式控制寄存器 00 (TMC00) 的 bit0 (OVF00) 清“0”。

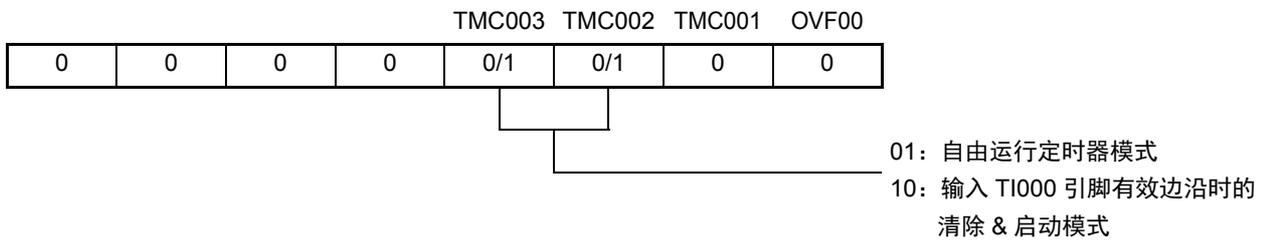
图 6-50 测量脉宽的时序示例 (3)



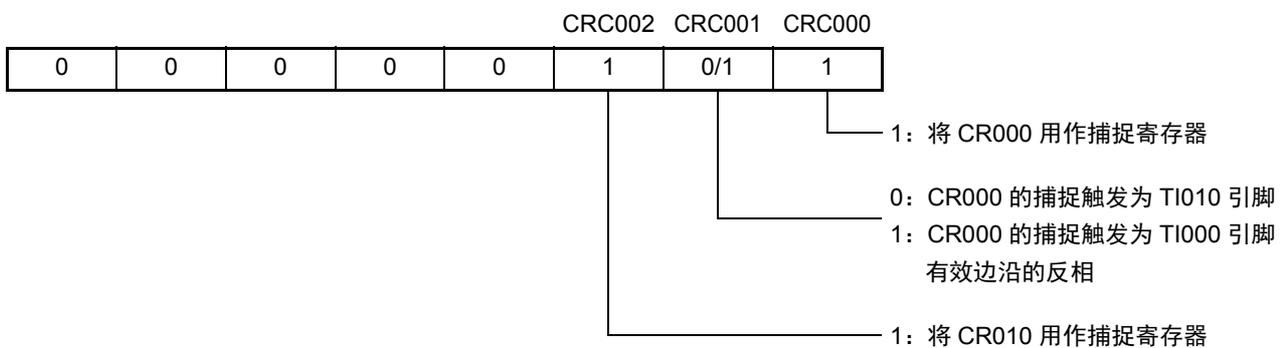
- ① 脉冲周期 = (10000H × OVF00 位被置“1”的次数 + CR010 的捕捉值) × 计数时钟周期
- ② 脉冲的高电平宽度 = (10000H × OVF00 位被置“1”的次数 + CR000 的捕捉值) × 计数时钟周期
- ③ 脉冲的低电平宽度 = (脉冲的周期 - 脉冲的高电平宽度)

图 6-51 测量脉宽时的寄存器设置内容示例 (1/2)

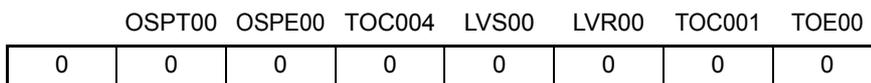
(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕捉 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频器模式寄存器 00 (PRM00)

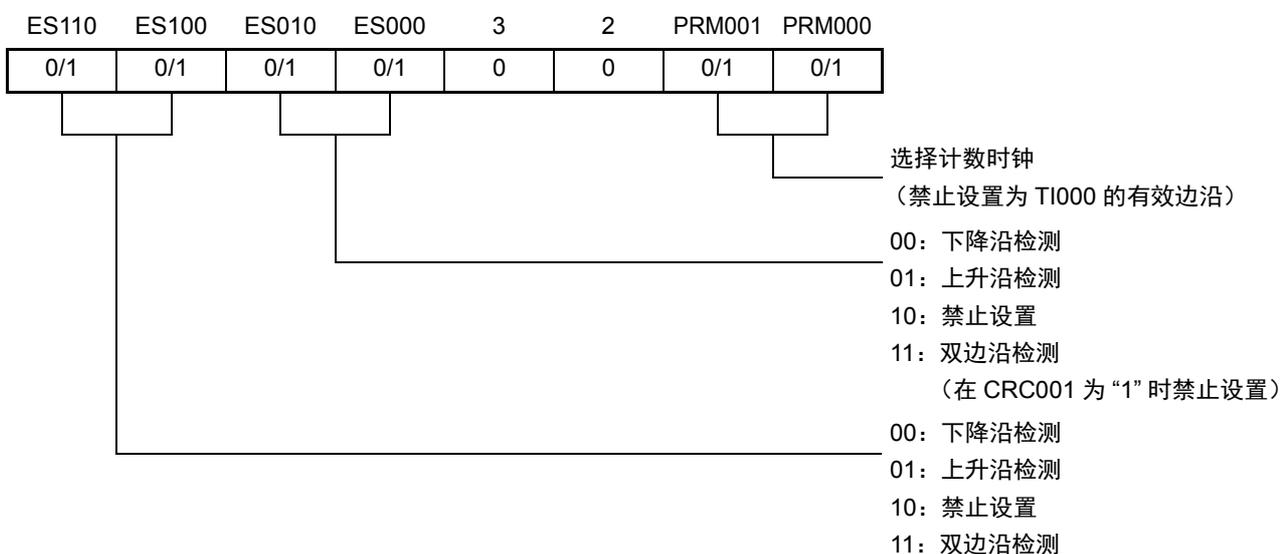


图 6-51 测量脉宽时的寄存器设置内容示例 (2/2)

(e) 16 位定时器计数器 00 (TM00)

通过读 TM00，读取计数器的值。

(f) 16 位捕捉 / 比较寄存器 000 (CR000)

用作捕捉寄存器。设置 TI000 引脚输入或者 TI010 引脚输入为捕捉触发，通过捕捉触发的边沿检测，将 TM00 的计数值保存到 CR000。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

用作捕捉寄存器。设置 TI000 引脚输入为捕捉触发，通过捕捉触发的边沿检测，将 TM00 的计数值保存到 CR010。

图 6-52 测量脉宽时的软件处理示例 (1/2)

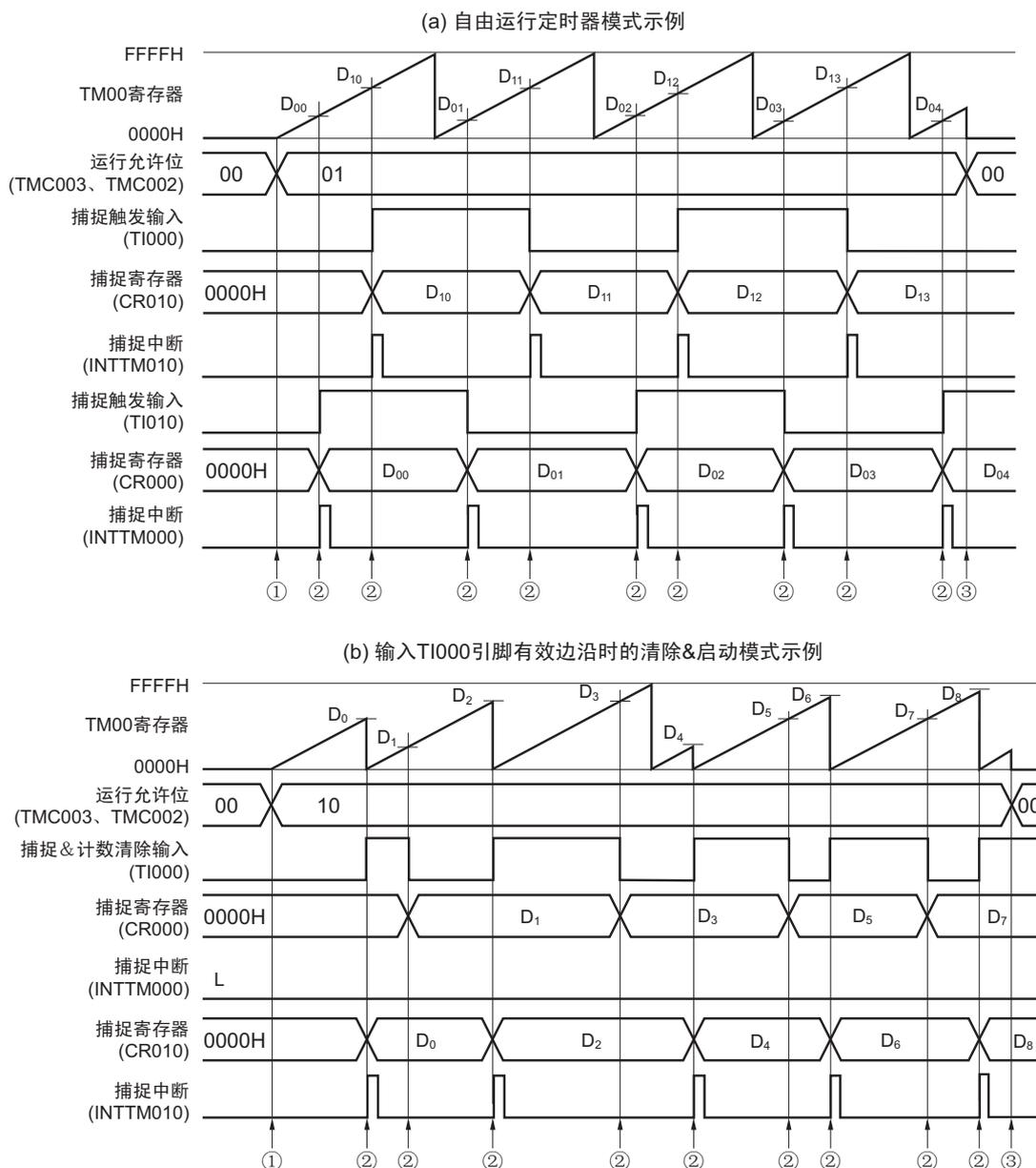
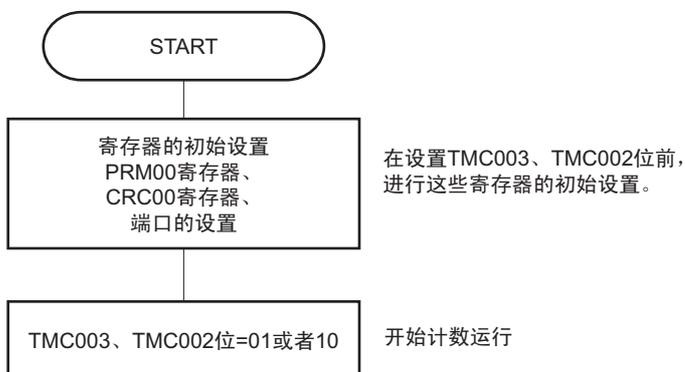
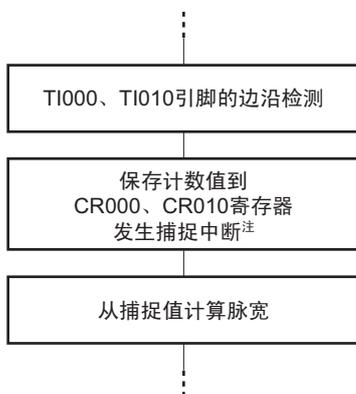


图 6-52 测量脉宽时的软件处理示例 (2/2)

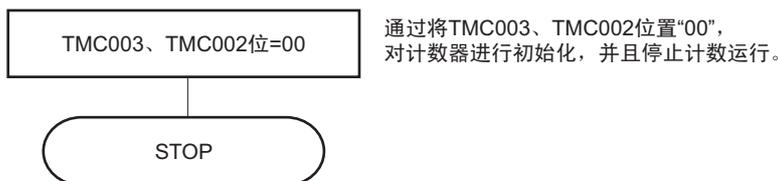
① 开始计数运行的流程



② 捕捉触发的输入流程



③ 停止计数运行的流程



注 选择 TI000 引脚输入的反相为 CR000 的有效边沿时，不产生捕捉中断信号 (INTTM00)。

6.5 TM00 的特殊使用方法

6.5.1 在 TM00 运行期间改写 CR010

原则上，在 TM00 运行（TMC003 和 TMC002 不为“00”时）期间，禁止改用作比较寄存器的 CR000 和 CR010。

但是，要在作为 PPG 输出的定时器运行期间更改占空比时，如果按照以下步骤设置，即使在 TM00 运行期间，也可改写 CR010。（当 CR010 的值小于当前的设置值时，在 CR010 与 TM00 匹配后，立即改写 CR010 的值。当 CR010 的值大于当前的设置值时，在 CR000 与 TM000 匹配后，立即改写 CR010 的值。如果在 CR010 与 TM00 或者 CR000 与 TM00 匹配前进行改写，就可能发生意想不到运行。）

CR010 的改写步骤

- ① 禁止 INTTM010 的中断（TMMK010=1）
- ② 禁止在 TM00 与 CR010 匹配时取反定时器输出（TOC004=0）
- ③ 改写 CR010
- ④ 等待 1 个 TM00 的计数时钟周期
- ⑤ 允许在 TM00 与 CR010 匹配时取反定时器输出（TOC004=1）
- ⑥ 将 INTTM010 的中断标志清“0”（TMIF010=0）
- ⑦ 允许 INTTM010 的中断（TMMK010=0）

备注 有关 TMIF010 和 TMMK010，请参照“第 16 章 中断功能”。

6.5.2 LVS00 和 LVR00 的设置

(1) LVS00 和 LVR00 的使用用途

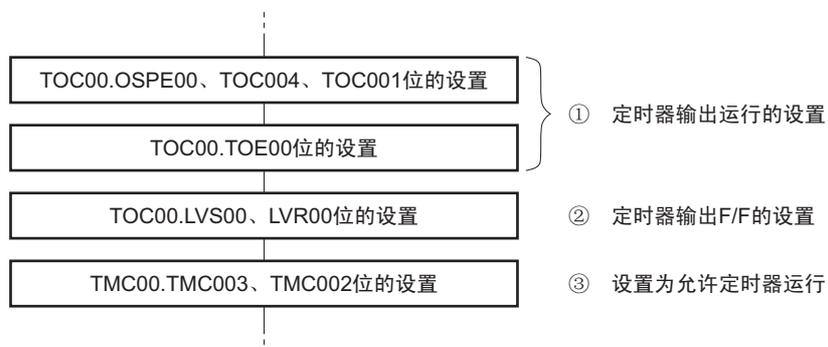
在设置 TO00 输出的初始值，或者在不允许定时器运行（TMC003 与 TMC002=00）的状态下取反定时器输出时，使用 LVS00 和 LVR00。不需要软件控制时，必须将 LVS00 和 LVR00 置“00”（初始值为低电平输出）。

LVS00	LVR00	定时器输出的状态
0	0	无变化（低电平输出）
0	1	清除（低电平输出）
1	0	设置（高电平输出）
1	1	禁止设置

(2) LVS00 和 LVR00 的设置方法

按照以下步骤设置 LVS00 和 LVR00。

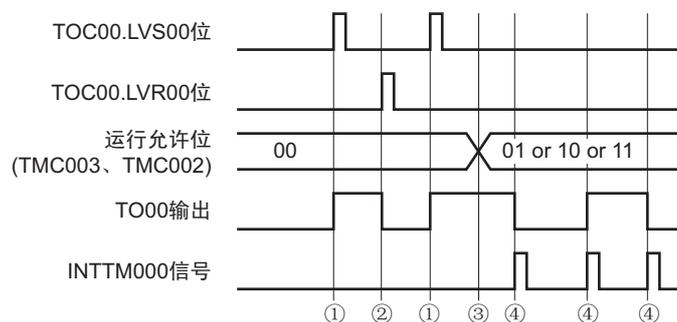
图 6-53 LVS00 和 LVR00 位的设置流程示例



注意 必须按上述①、②、③的步骤设置 LVS00 和 LVR00。

只要在①的设置后③的设置前，就可进行②的设置。

图 6-54 LVR00 和 LVS00 的时序示例



- ① 通过将 LVS00、LVR00 置“10”，TO00 输出变为高电平。
- ② 通过将 LVS00、LVR00 置“01”，TO00 输出变为低电平。
(即使将 LVS00、LVR00 置“00”，TO00 输出也不变化，保持高电平。)
- ③ 通过将 TMC003、TMC002 置“01”、“10”或者“11”，开始定时器运行。因为开始运行前的 LVS00 和 LVR00 的设置值为“10”，所以 TO00 输出从高电平开始。从定时器开始运行到 TMC003 和 TMC002 变为“00”（禁止定时器运行）前，禁止设置 LVS00 和 LVR00。
- ④ 每当产生中断信号 (INTTM000)，取反 TO00 输出的电平。

6.6 16 位定时器 / 事件计数器 00 的注意事项

(1) 16 位定时器 / 事件计数器 00 的各通道限制事项

各通道的限制事项如表 6-3 所示。

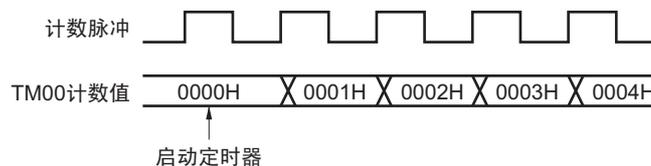
表 6-3 16 位定时器 / 事件计数器 00 的各通道限制事项

运行	限制事项
作为间隔定时器的运行	—
方波的输出运行	
作为外部事件计数器的运行	
作为输入 TI000 引脚有效边沿时的清除 & 启动模式的运行	使用 TI010 引脚的有效边沿检测时，禁止使用定时器输出（TO00）。（设置 TOC00=00H）
作为自由运行定时器模式的运行	—
PPG 的输出运行	$0000H \leq CP010 < CR000 \leq FFFFH$
单次脉冲的输出运行	禁止给 CR000 和 CP010 设置相同的值。
脉宽的测量运行	禁止使用定时器输出（TO00）（设置 TOC00=00H）

(2) 定时器启动时的误差

由于没有与计数脉冲同步开始 TM00 的计数，导致从启动定时器到产生匹配信号前的时间产生最大 1 个时钟的误差。

图 6-55 TM00 的计数开始时序



(3) CR000 和 CR010 的设置（TM00 和 CR000 匹配时的清除 & 启动模式）

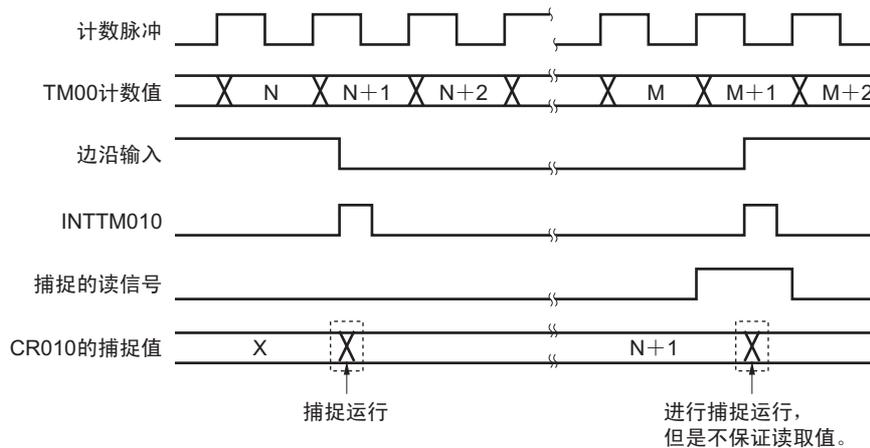
必须给 CR000 和 CR010 设置“0000H”以外的值（用作外部事件计数器时，不能进行 1 个脉冲的计数运行）。

(4) 捕捉寄存器的数据保持时序

- (a) 在读取 CR000/CR010 期间输入 TI000/TI010 引脚的有效边沿，或者检测出 TI000 引脚有效边沿的反相时，CR010 执行捕捉运行，但是不保证 CR000/CR010 的读取值。此时，如果检测出 TI000/TI010 引脚的有效边沿，就产生中断信号（INTTM000/INTTM010）（如果检测出 TI000 引脚有效边沿的反相，就不产生中断信号）。

如果在检测出 TI000/TI010 引脚的有效边沿时进行捕捉，必须在产生 INTTM000/INTTM010 信号后读取 CR000/CR010 的值。

图 6-56 捕捉寄存器的数据保持时序



- (b) 不保证 16 位定时器 / 事件计数器 00 停止后的 CR000 和 CR010 的值。

(5) 有效边沿的设置

必须在定时器运行的停止（TMC003 和 TMC002=00）期间，通过 ES000 和 ES010 设置 TI000 引脚的有效边沿。

(6) 单次脉冲的重新触发

在单次脉冲输出模式的有效电平输出期间，不能产生触发。必须在当前的有效电平输出结束后输入下一个触发。

(7) OVF00 标志的变化

(a) OVF00 标志的设置 (1)

在以下情况 (TM00 发生上溢时除外) 下, OVF00 标志置“1”。

选择 TM00 与 CR000 匹配时的清除 & 启动模式。

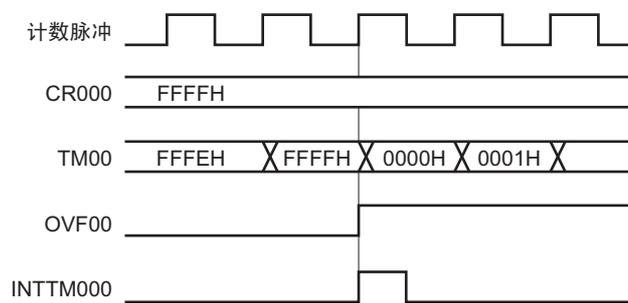
↓

将 CR000 置“FFFFH”。

↓

在 TM00 与 CR000 相匹配时从“FFFFH”清“0000H”的情况。

图 6-57 OVF00 标志的变化时序



(b) OVF00 标志的清除

从 TM00 发生上溢到下一个计数时钟被计数 (TM00 变为“0001H”) 前, 即使将 OVF00 标志清“0”, OVF00 也会重新被置“1”, 清除无效。

(8) 单次脉冲输出

单次脉冲输出在自由运行定时器模式或者输入 TI000 引脚有效边沿时的清除 & 启动模式中正常运行。在 TM00 和 CR000 匹配时的清除 & 启动模式中, 不能输出单次脉冲。

(9) 捕捉运行

(a) 指定 TI000 的有效边沿为计数时钟的情况

指定 TI000 的有效边沿为计数时钟时, 将 TI000 指定为触发的捕捉寄存器不能正常运行。

(b) 通过 TI010 和 TI000 引脚的输入信号确保捕捉的脉宽

作为确保捕捉的捕捉触发, 输入到 TI000 和 TI010 引脚的脉冲的脉宽必须大于 2 个 PRM00 所选计数时钟 (参照图 6-7)。

(c) 中断信号的产生

在计数时钟的下降沿执行捕捉运行, 但是在下一个计数时钟的上升沿产生中断信号 (INTTM000 和 INTTM010) (参照图 6-7)。

(d) 将 CRC001 (捕捉/比较控制寄存器 00 (CRC00) 的 bit1) 置“1”时的注意事项

在通过 TI000 引脚输入的反相捕捉 TM00 的计数值到 CR000 寄存器时, 捕捉后不产生中断请求信号 (INTTM000)。如果在此运行期间检测出 TI010 引脚的有效边沿, 就不执行捕捉, 但是产生作为外部中断信号的 INTTM000 信号。在不使用外部中断时, 必须屏蔽 INTTM000 信号。

(10) 边沿检测

(a) 复位后的有效边沿指定

复位后，如果在 TI000 引脚或者 TI010 引脚为高电平状态下，指定上升沿或者双边沿为 TI000 引脚或者 TI010 引脚的有效边沿，并且允许 16 位定时器/事件计数器 00 的运行，该高电平就作为上升沿被检测。上拉 TI000 引脚或者 TI010 引脚时等要特别注意。但是，如果允许停止运行后的重新运行，就检测不出上升沿。

(b) 用于消除噪声的采样时钟

在 TI000 的有效边沿用作计数时钟和用作捕捉触发时，用于消除噪声的采样时钟不同。前者的采样时钟固定为 f_{PRS} 。后者以 PRM00 所选的计数时钟进行采样。

对 TI000 引脚的输入信号进行采样，如果连续 2 次检测出有效电平，才判断为有效边沿。因此，能消除窄脉宽的噪声（参照图 6-7）。

(11) 有关定时器的运行

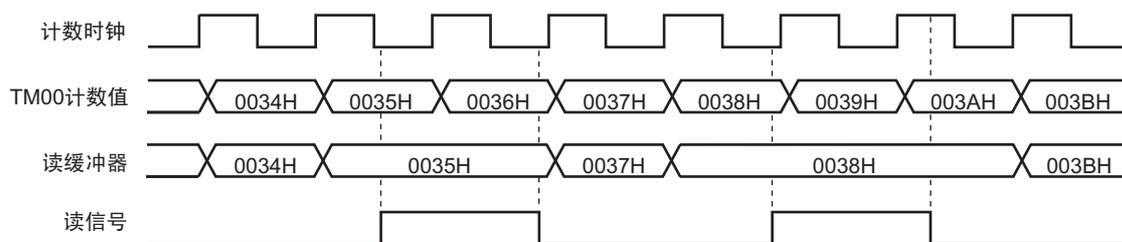
与 CPU 的运行模式无关，只要定时器停止，就不接受 TI000/TI010 引脚的输入信号。

备注 f_{PRS} ：外围硬件时钟频率

(12) 有关 16 位定时器计数器 00 (TM00) 的读取

因为通过固定保存到缓冲器的计数值来读取 TM00，所以可在没有停止计数器运行的情况下进行读取。但是在计数器进行递增计数时会更新缓冲器，因此，如果在递增计数前进行读取，缓冲器就有可能未被更新。

图 6-58 16 位定时器计数器 00 (TM00) 的读取时序



第 7 章 8 位定时器 / 事件计数器 50、51

7.1 8 位定时器 / 事件计数器 50、51 的功能

8 位定时器 / 事件计数器 50、51 具有以下功能。

- 间隔定时器
- 外部事件计数器
- 方波输出
- PWM 输出

7.2 8 位定时器 / 事件计数器 50、51 的结构

8 位定时器 / 事件计数器 50、51 由以下硬件构成。

表 7-1 8 位定时器 / 事件计数器 50、51 的结构

项目	结构
定时器寄存器	8 位定时器计数器 5n (TM5n)
寄存器	8 位定时器比较寄存 5n (CR5n)
定时器输入	TI5n
定时器输出	TO5n
控制寄存器	定时器时钟选择寄存器 5n (TCL5n) 8 位定时器模式控制寄存器 5n (TMC5n) 端口模式寄存器 1 (PM1) 或者端口模式寄存器 3 (PM3) 端口寄存器 1 (P1) 或者端口寄存器 3 (P3)

8 位定时器 / 事件计数器 50、51 的框图如图 7-1、图 7-2 所示。

图 7-1 8 位定时器 / 事件计数器 50 的框图

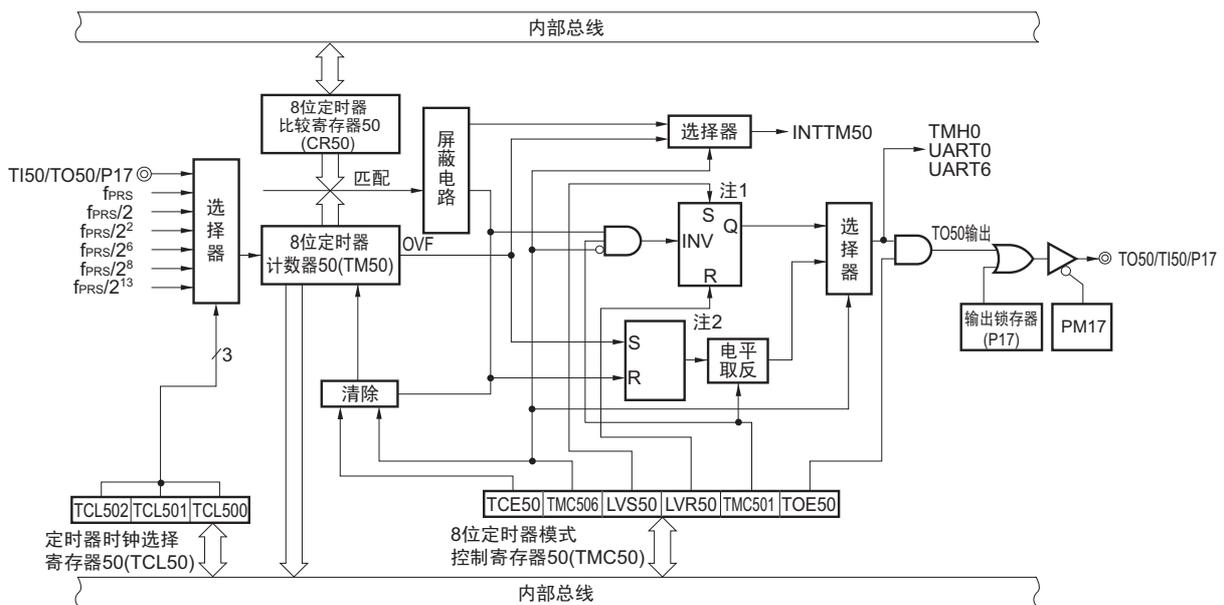
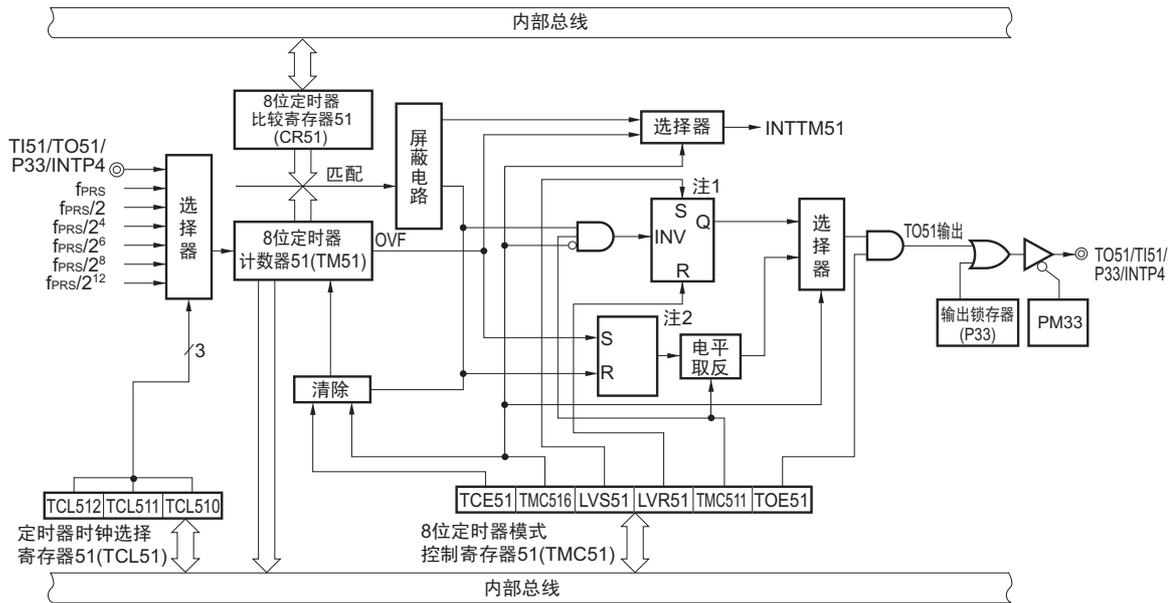


图 7-2 8 位定时器 / 事件计数器 51 的框图



- 注 1. 定时器输出 F/F
- 2. PWM 输出 F/F

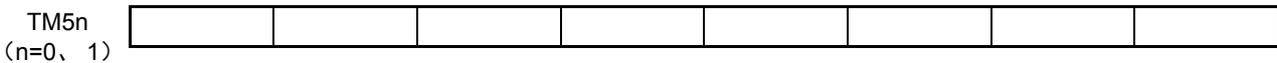
(1) 8 位定时器计数器 5n (TM5n)

TM5n 是 8 位只读寄存器，对计数脉冲进行计数。
计数器与计数时钟的上升沿同步进行递增计数。

图 7-3 8 位定时器计数器 5n (TM5n) 的格式

地址: FF16H (TM50)、FF1FH (TM51) 复位后: 00H R

符号 7 6 5 4 3 2 1 0



在以下情况下，计数值为“00H”。

- ① 产生复位信号
- ② 清除 TCE5n
- ③ 在 TM5n 和 CR5n 匹配时的清除 & 启动模式中 TM5n 和 CR5n 匹配

(2) 8位定时器比较寄存器 5n (CR5n)

通过 8 位存储器操作指令读写 CR5n。

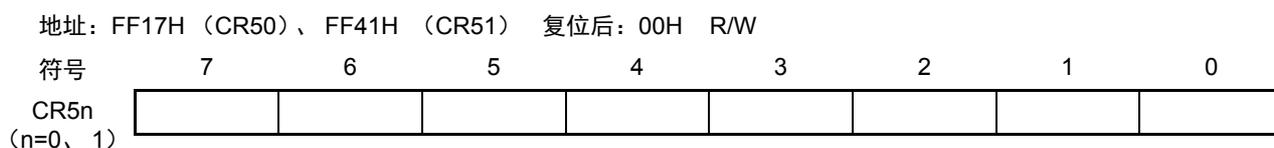
在非 PWM 模式中总是将 CR5n 的设置值与 8 位定时器计数器 5n (TM5n) 的计数值进行比较，如果两个值匹配，就产生中断请求 (INTTM5n)。

在 PWM 模式中，如果 TM5n 的值与 CR5n 的值匹配，TO5n 输出就变为非有效电平输出，并且不发生中断。

在“00H ~ FFH”范围内设置 CR5n 的值。

在产生复位信号后，CR5n 变为“00H”。

图 7-4 8 位定时器比较寄存器 5n (CR5n) 的格式



注意 1. 在 TM5n 与 CR5n 匹配时的清除 & 启动模式 (TMC5n6=0) 中，不能在运行期间给 CR5n 写入不同的值。

2. 在 PWM 模式中，CR5n 的改写间隔必须大于等于 3 个通过 TCL5n 选择的计数时钟。

备注 n=0、1

7.3 8位定时器/事件计数器 50、51 的控制寄存器

8位定时器/事件计数器 50、51 的控制寄存器有以下 4 种。

- 定时器时钟选择寄存器 5n (TCL5n)
- 8位定时器模式控制寄存器 5n (TMC5n)
- 端口模式寄存器 1 (PM1) 或者端口模式寄存器 3 (PM3)
- 端口寄存器 1 (P1) 或者端口寄存器 3 (P3)

(1) 定时器时钟选择寄存器 5n (TCL5n)

定时器时钟选择寄存器 5n 设置 8 位定时器/事件计数器 5n 的计数时钟和 TI5n 引脚输入的有效边沿。

通过 1 位或 8 位存储器操作指令设置 TCL5n。

在产生复位信号后，TCL5n 变为“00H”。

备注 n=0、1

图 7-5 定时器时钟选择寄存器 50 (TCL50) 的格式

地址: FF6AH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500

TCL502	TCL501	TCL500	计数时钟的选择			
				$f_{PRS}=2\text{MHz}$	$f_{PRS}=5\text{MHz}$	$f_{PRS}=10\text{MHz}$
0	0	0	TI50 引脚的下降沿注			
0	0	1	TI50 引脚的上升沿注			
0	1	0	f_{PRS}	2MHz	5MHz	10MHz
0	1	1	$f_{PRS}/2$	1MHz	2.5MHz	5MHz
1	0	0	$f_{PRS}/2^4$	500kHz	1.25MHz	2.5MHz
1	0	1	$f_{PRS}/2^6$	31.25kHz	78.13kHz	156.25kHz
1	1	0	$f_{PRS}/2^8$	7.81kHz	19.53kHz	39.06kHz
1	1	1	$f_{PRS}/2^{13}$	0.24kHz	0.61kHz	1.22kHz

注 在 STOP 模式中，不能以 TI50 引脚的外部时钟开始定时器的运行。

注意 1. 如果要给 TCL50 改写不同数据，必须先停止定时器运行后再改写。

2. bit3 ~ bit7 必须设置为“0”。

备注 f_{PRS} : 外围硬件时钟频率

图 7-6 定时器时钟选择寄存器 51 (TCL51) 的格式

地址: FF8CH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	计数时钟的选择			
				$f_{PRS}=2\text{MHz}$	$f_{PRS}=5\text{MHz}$	$f_{PRS}=10\text{MHz}$
0	0	0	TI51 引脚的下降沿注			
0	0	1	TI51 引脚的上升沿注			
0	1	0	f_{PRS}	2MHz	5MHz	10MHz
0	1	1	$f_{PRS}/2$	1MHz	2.5MHz	5MHz
1	0	0	$f_{PRS}/2^4$	125kHz	312.5kHz	625kHz
1	0	1	$f_{PRS}/2^6$	31.25kHz	78.13kHz	156.25kHz
1	1	0	$f_{PRS}/2^8$	7.81kHz	19.53kHz	39.06kHz
1	1	1	$f_{PRS}/2^{12}$	0.49kHz	1.22kHz	2.44kHz

注 在 STOP 模式中，不能以 TI51 引脚的外部时钟开始定时器的运行。

注意 1. 如果要给 TCL51 改写不同数据，必须先停止定时器运行后再改写。

2. bit3 ~ bit7 必须设置为“0”。

备注 f_{PRS} : 外围硬件时钟频率

(2) 8位定时器模式控制寄存器 5n (TMC5n)

TMC5n 可进行以下 5 种设置。

1. 控制8位定时器计数器5n (TM5n) 的计数运行
2. 选择8位定时器计数器5n (TM5n) 的运行模式
3. 设置定时器输出F/F (触发器) 的状态
4. 控制定时器F/F或者选择PWM (自由运行) 模式中的有效电平
5. 控制定时器输出

通过 1 位或 8 位存储器操作指令设置 TMC5n。

在产生复位信号后，TMC5n 为“00H”。

备注 n=0、1

图 7-7 8 位定时器模式控制寄存器 50 (TMC50) 的格式

地址: FF6BH 复位后: 00H R/W注

符号	<7>	6	5	4	<3>	<2>	1	<0>
TMC50	TCE50	TMC506	0	0	LVS50	LVR50	TMC501	TOE50

TCE50	控制 TM50 的计数运行
0	将计数器清“0”后，禁止计数运行 (计数器停止)
1	开始计数运行

TMC506	选择 TM50 的运行模式
0	TM50 和 CR50 匹配时的清除 & 启动模式
1	PWM (自由运行) 模式

LVS50	LVR50	设置定时器输出 F/F 的状态
0	0	不发生变化
0	1	将定时器输出 F/F 清“0” (TO50 输出初始值为低电平)
1	0	将定时器输出 F/F 清“1” (TO50 输出初始值为高电平)
1	1	禁止设置

TMC501	非 PWM 模式 (TMC506=0)	PWM 模式 (TMC506=1)
	控制定时器 F/F	选择有效电平
0	禁止取反运行	高电平有效
1	允许取反运行	低电平有效

TOE50	控制定时器输出
0	禁止输出 (TO50 输出为低电平输出)
1	允许输出

注 bit2 和 bit3 为只写位。

图 7-8 8 位定时器模式控制寄存器 51 (TMC51) 的格式

地址: FF43H 复位后: 00H R/W注

符号	<7>	6	5	4	<3>	<2>	1	<0>
TMC51	TCE51	TMC516	0	0	LVS51	LVR51	TMC511	TOE51

TCE51	控制 TM51 的计数运行	
0	将计数器清“0”后, 禁止计数运行 (计数器停止)	
1	开始计数运行	

TMC516	选择 TM51 的运行模式	
0	TM51 和 CR51 匹配时的清除 & 启动模式	
1	PWM (自由运行) 模式	

LVS51	LVR51	设置定时器输出 F/F 的状态
0	0	不发生变化
0	1	将定时器输出 F/F 清“0” (TO51 输出初始值为低电平)
1	0	将定时器输出 F/F 清“1” (TO51 输出初始值为高电平)
1	1	禁止设置

TMC511	非 PWM 模式 (TMC516=0)	PWM 模式 (TMC516=1)
	控制定时器 F/F	
0	禁止取反运行	高电平有效
1	允许取反运行	低电平有效

TOE51	控制定时器输出
0	禁止输出 (TO51 输出为低电平输出)
1	允许输出

注 bit2 和 bit3 为只写位。

注意 1. 在非 PWM 模式中, LVS5n 和 LVR5n 的设置有效。

2. 按照以下步骤进行设置, 禁止同时设置①~④。

① 设置 TMC5n1、TMC5n6: 设置运行模式

② 允许输出时, 设置 TOE5n: 允许定时器输出

③ 设置 LVS5n、LVR5n (注意 1): 设置定时器 F/F

④ 设置 TCE5n

3. 在 TCE5n 为“1”时, 禁止设置 TMC5n 的其他位。

4. 实际的 TO50/TI50/P17、TO51/TI51/P33/INTP4 引脚的 TO5n 输出取决于 PM17 和 P17 以及 PM33 和 P33。

备注 1. 在 PWM 模式中, 通过将 TCE5n 置“0”, PWM 输出变为无效电平。

2. LVS5n 和 LVR5n 的读取值为“0”。

3. TMC5n6、LVS5n、LVR5n、TMC5n1、TOE5n 的各位的值与 TCE5n 的值无关, 反映到 TO5n 输出。

4. n=0、1

(3) 端口模式寄存器 1、3 (PM1、PM3)

端口模式寄存器 1、3 以 1 位单位设置端口 1、3 的输入 / 输出。

将 P17/TO50/TI50、P33/TO51/TI51/INTP4 引脚用作定时器输出时，必须将 PM17、PM33、P17 以及 P33 的输出锁存器设置为“0”。

将 P17/TO50/TI50、P33/TO51/TI51/INTP4 引脚用作定时器输入时，必须将 PM17、PM33 置“1”。此时，P17、P33 的输出锁存器既可是“0”也可是“1”。

通过 1 位或 8 位存储器操作指令设置 PM1、PM3。

在产生复位信号后，PM1、PM3 变为“FFH”。

图 7-9 端口模式寄存器 1 (PM1) 的格式

地址：FF21H 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
PM1n	选择 P1n 引脚的输入 / 输出模式 (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)							

图 7-10 端口模式寄存器 3 (PM3) 的格式

地址：FF23H 复位后：FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30
PM3n	选择 P3n 引脚的输入 / 输出模式 (n=0 ~ 3)							
0	输出模式 (输出缓冲器 ON)							
1	输入模式 (输出缓冲器 OFF)							

7.4 8位定时器/事件计数器 50、51 的运行

7.4.1 作为间隔定时器的运行

将8位定时器比较寄存器5n (CR5n) 中预置的计数值作为间隔，并且作为重复产生中断请求的间隔定时器运行。

在8位定时器计数器5n (TM5n) 的计数值与CR5n 的设置值匹配时，将TM5n 清“0”，并且在继续计数的同时，产生中断请求信号 (INTTM5n)。

可通过定时器时钟选择寄存器5n (TCL5n) 的bit0 ~ bit2 (TCL5n0 ~ TCL5n2) 选择TM5n 的计数时钟。

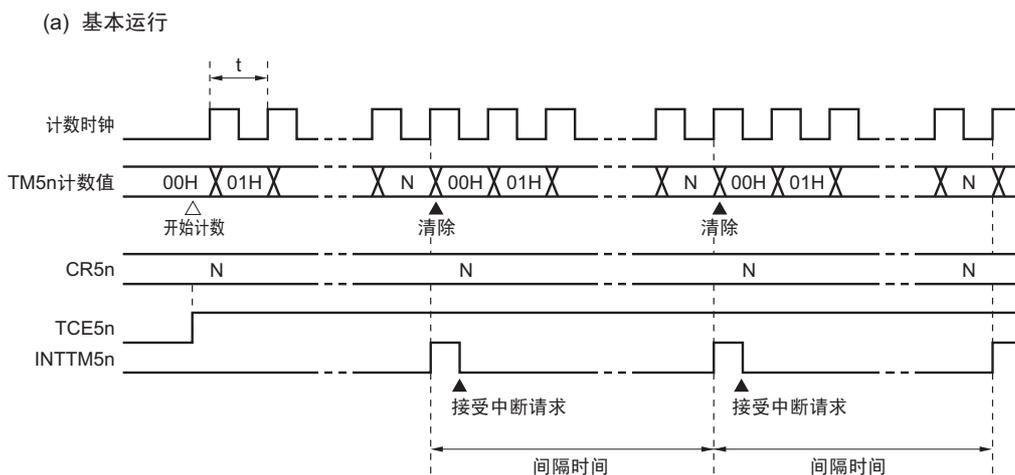
设置方法

- ① 设置各寄存器。
 - TCL5n: 选择计数时钟
 - CR5n: 比较值
 - TMC5n: 停止计数运行、选择TM5n 与CR5n 匹配时的清除&启动模式
(TMC5n=0000xxx0B x: 忽略)
- ② 如果将TCE5n 置“1”，开始计数运行。
- ③ 如果TM5n 与CR5n 的值匹配，就产生INTTM5n (TM5n 被清“00H”)。
- ④ 之后，以相同的时间间隔重复产生INTTM5n。在停止计数运行时，必须将TCE5n 置“0”。

注意 运行期间不能给CR5n 写入不同的值。

备注 有关INTTM5n 信号的中断允许，请参照“第16章 中断功能”。

图 7-11 间隔定时器运行的时序 (1/2)



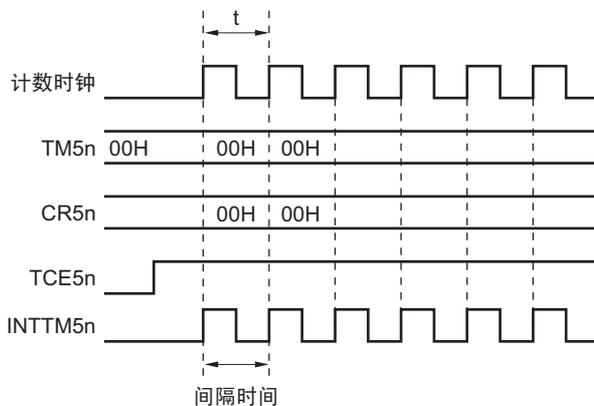
备注 间隔时间 $= (N+1) \times t$

$N=01H \sim FFH$

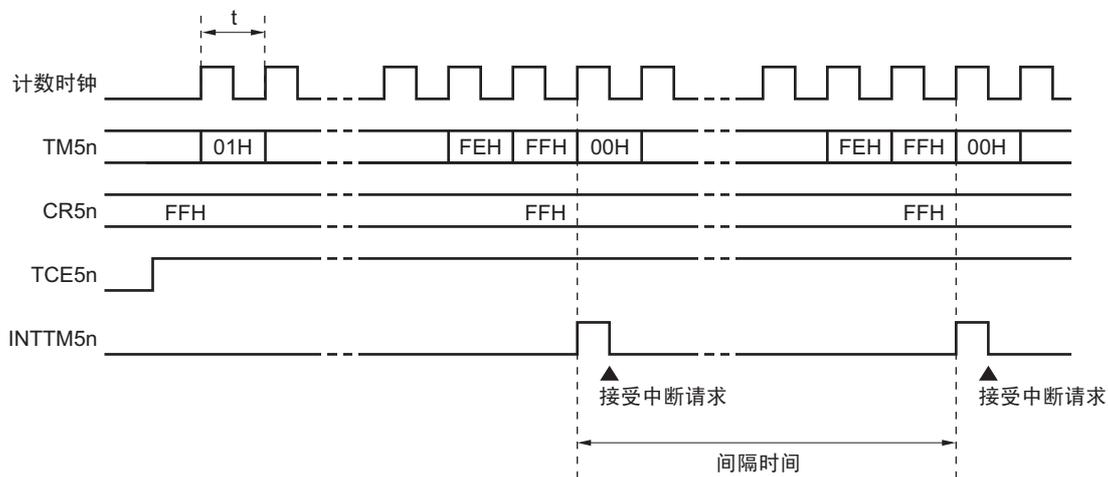
$n=0, 1$

图 7-11 间隔定时器运行的时序 (2/2)

(b) 当CR5n=00H时



(c) 当CR5n=FFH时



备注 n=0、1

7.4.2 作为外部事件计数器的运行

外部事件计数器通过8位定时器计数器5n (TM5n)对输入到TI5n引脚的外部时钟脉冲进行计数。

每当输入由定时器时钟选择寄存器5n (TCL5n)指定的有效边沿, TM5n就进行递增计数。指定的边沿可从上升沿和下降沿中选择。

如果TM5n的计数值与8位定时器比较寄存器5n (CR5n)的值匹配, TM5n就清“0”, 并且产生中断请求信号 (INTTM5n)。

之后, 每当TM5n的值与CR5n的值匹配, 就产生INTTM5n。

设置方法

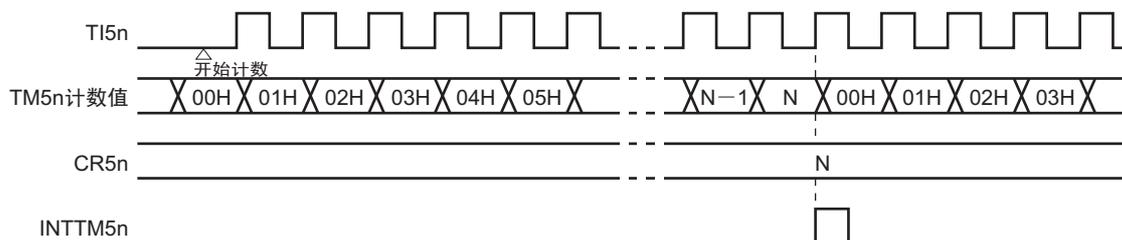
- ① 设置各寄存器。
 - 将端口模式寄存器 (PM17、PM33) 注置“1”
 - TCL5n: 选择TI5n引脚输入的边沿
TI5n引脚的下降沿 → TCL5n=00H
TI5n引脚的上升沿 → TCL5n=01H
 - CR5n: 比较值
 - TMC5n: 停止计数、选择TM5n与CR5n匹配时的清除&启动模式、禁止定时器F/F的取反运行、禁止定时器输出。
(TMC5n=00000000B)
- ② 如果将TCE5n置“1”, 就对TI5n引脚的输入脉冲进行计数。
- ③ 如果TM5n的值与CR5n的值匹配, 就产生INTTM5n (TM5n被清“00H”)。
- ④ 之后, 每当TM5n的值与CR5n的值匹配, 就将产生INTTM5n。

注 8位定时器/事件计数器 50: PM17

8位定时器/事件计数器 51: PM33

备注 有关INTTM5n信号的中断允许, 请参照“第16章 中断功能”。

图 7-12 外部事件计数器运行的时序 (指定上升沿时)



备注 N=00H ~ FFH

n=0、1

7.4.3 方波的输出运行

以输出 8 位定时器比较寄存器 5n (CR5n) 的预置值决定的间隔的任意频率方波作为方波输出运行。

通过将 8 位定时器模式控制寄存器 5n (TMC5n) 的 bit0 (TOE5n) 置“1”，以 CR5n 预置的计数值决定的间隔取反 TO5n 的输出状态。由此输出任意频率的方波 (占空比 = 50%)。

备注 方波输出仅可用作 UART0 的基本时钟。

设置方法

- ① 设置各寄存器。
 - 将端口输出锁存器 (P17、P33) 注和端口模式寄存器 (PM17、PM33) 注置“0”
 - TCL5n: 选择计数时钟
 - CR5n: 比较值
 - TMC5n: 停止计数运行、选择 TM5n 与 CR5n 匹配时的清除 & 启动模式

LVS5n	LVR5n	设置定时器输出 F/F 的状态
0	1	将定时器输出 F/F 清“0” (TO5n 输出初始值为低电平)
1	0	将定时器输出 F/F 置“1” (TO5n 输出初始值为高电平)

允许定时器输出

(TMC5n=00001011B 或者 00000111B)

- ② 如果将 TCE5n 置“1”，开始计数运行。
- ③ 如果 TM5n 与 CR5n 的值匹配，取反定时器输出 F/F。
另外，产生 INTTM5n，并且 TM5n 被清“00H”。
- ④ 之后，以相同间隔取反定时器输出 F/F，并且从 TO5n 输出方波。
频率如下所示。
 - 频率 = $1/2t(N+1)$
(N: 00H ~ FFH)

注 8 位定时器 / 事件计数器 50: P17、PM17

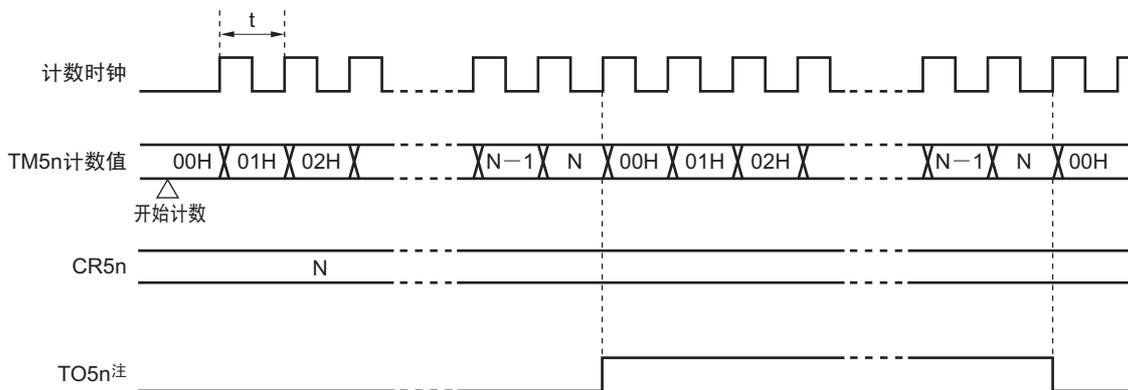
8 位定时器 / 事件计数器 51: P33、PM33

注意 运行期间不能给 CR5n 写入不同的值。

备注 1. 有关 INTTM51n 信号的中断允许，请参照“第 16 章 中断功能”。

2. n=0、1

图 7-13 方波输出运行的时序



注 可通过 8 位定时器模式控制寄存器 5n (TMC5n) 的 bit2、bit3 (LVR5n、LVS5n) 设置 TO5n 输出的初始值。

7.4.4 PWM 的输出运行

通过将 8 位定时器模式控制寄存器 5n (TMC5n) 的 bit6 (TMC5n6) 置“1”，进行 PWM 输出运行。
 从 TO5n 输出由 8 位定时器比较寄存器 5n (CR5n) 的设置值决定的占空比的脉冲。
 将 PWM 脉冲的有效电平宽度设置到 CR5n，并且可通过 TMC5n 的 bit1 (TMC5n1) 选择有效电平。
 可通过定时器时钟选择寄存器 5n (TCL5n) 的 bit0 ~ bit2 (TCL5n0 ~ TCL5n2)，选择计数时钟。
 可通过 TMC5n 的 bit0 (TOE5n) 选择允许/禁止 PWM 输出。

注意 在 PWM 模式中，CR5n 的改写间隔至少为 3 个计数时钟 (由 TCL5n 选择的时钟)。

备注 n=0、1

(1) PWM 输出的基本运行

设置方法

① 设置各寄存器。

- 将端口输出锁存器 (P17、P33) 注和端口模式寄存器 (PM17、PM33) 注置“0”
- TCL5n: 选择计数时钟
- CR5n: 比较值
- TMC5n: 停止计数运行、选择 PWM 模式

TMC5n1	选择有效电平
0	高电平有效
1	低电平有效

允许定时输出

(TMC5n=01000001B 或者 01000011B)

- ② 如果将 TCE5n 置“1”，就开始计数运行。
 如果要停止计数，必须将 TCE5n 置“0”。

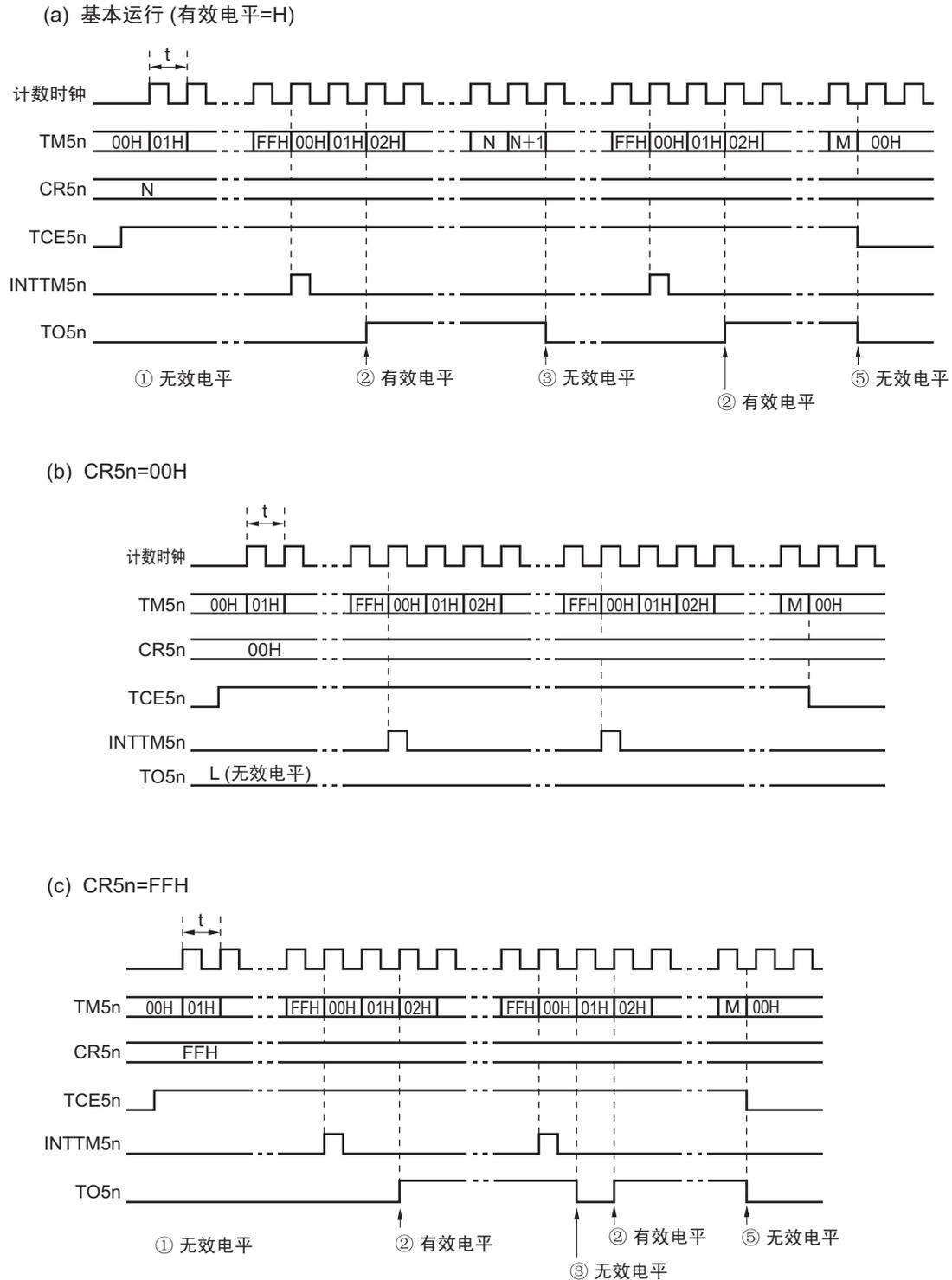
注 8 位定时器 / 事件计数器 50: P17、PM17
 8 位定时器 / 事件计数器 51: P33、PM33

PWM 输出

- ① 直到发生上溢前，PWM 输出 (TO5n 输出) 为无效电平输出。
- ② 如果发生上溢，输出有效电平，直到 CR5n 与 8 位定时器计数器 5n (TM5n) 的计数值匹配。
- ③ 如果 CR5n 与计数值匹配，就输出无效电平，直到再次发生上溢。
- ④ 之后，重复②和③，直到计数停止。
- ⑤ 如果通过将 TCE5n 置“0”停止计数，PWM 输出变为无效电平输出。
 有关详细的时序，请参照图 7-14 和图 7-15。
 周期、有效电平宽度和占空比如下所示。
 - 周期 = $2^8 t$
 - 有效电平宽度 = Nt
 - 占空比 = $N/2^8$
 (N=00H ~ FFH)

备注 n=0、1

图 7-14 PWM 输出运行的时序

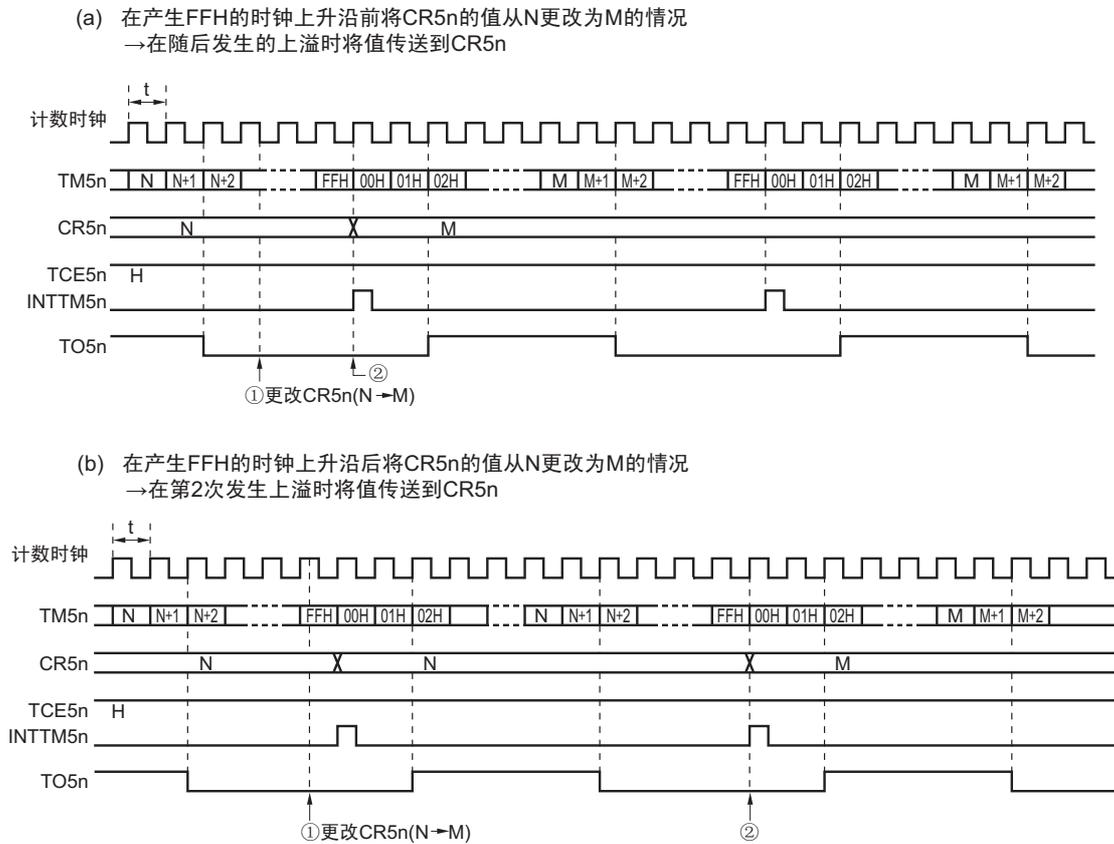


备注 1. 图 7-14(a) 和 (c) 中的①~③、⑤分别对应“7.4.4 (1) PWM 输出的基本运行”的“PWM 输出”中的①~③、⑤。

2. $n=0, 1$

(2) 更改 CR5n 的运行

图 7-15 更改 CR5n 的运行时序



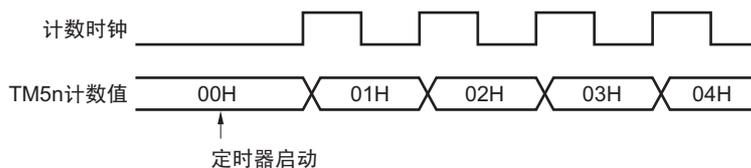
注意 在图 7-15 中的①与②之间读 CR5n 时，读取值与实际运行的值不同（读取值：M、实际的 CR5n 值：N）。

7.5 8 位定时器 / 事件计数器 50、51 的注意事项

(1) 定时器启动时的误差

从定时器启动到匹配信号产生前，产生最大 1 个时钟的误差。这是因为 8 位定时器计数器 50、51 (TM50、TM51) 没有与计数时钟同步启动而导致的。

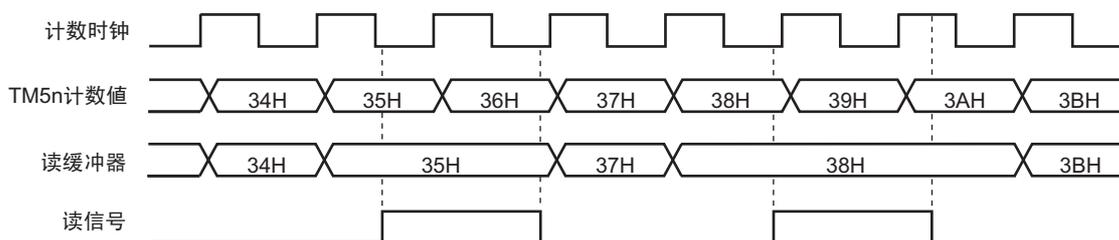
图 7-16 8 位定时器计数器 5n (TM5n) 的启动时序



(2) 8 位定时器计数器 5n (TM5n) 的读取

因为是在将捕捉到缓冲器的计数值固定后读取 TM5n，所以能在没有停止实际计数器的情况下进行读取。但是，缓冲器在计数器的递增计数时被更新，因此，如果在递增计数之前读取 TM51，缓冲器有可能不被更新。

图 7-17 8 位定时器计数器 5n (TM5n) 的读取时序



备注 n=0、1

第 8 章 8 位定时器 H0、H1

8.1 8 位定时器 H0、H1 的功能

8 位定时器 H0、H1 具有以下功能。

- 间隔定时器
- 方波输出
- PWM 输出
- 载波发生器（只限 8 位定时器 H1）

8.2 8 位定时器 H0、H1 的结构

8 位定时器 H0、H1 由以下硬件构成。

表 8-1 8 位定时器 H0、H1 的结构

项目	结构
定时器寄存器	8 位定时器计数器 Hn
寄存器	8 位定时器 H 比较寄存器 0n (CMP0n) 8 位定时器 H 比较寄存器 1n (CMP1n)
定时器输出	TOHn、输出控制电路
控制寄存器	8 位定时器 H 模式寄存器 n (TMHMDn) 8 位定时器 H 载波控制寄存器 1 (TMCYC1) 注 端口模式寄存器 1 (PM1) 端口寄存器 1 (P1)

注 只限 8 位定时器 H1。

备注 n=0、1

框图如图 8-1 和图 8-2 所示。

图 8-1 8 位定时器 H0 的框图

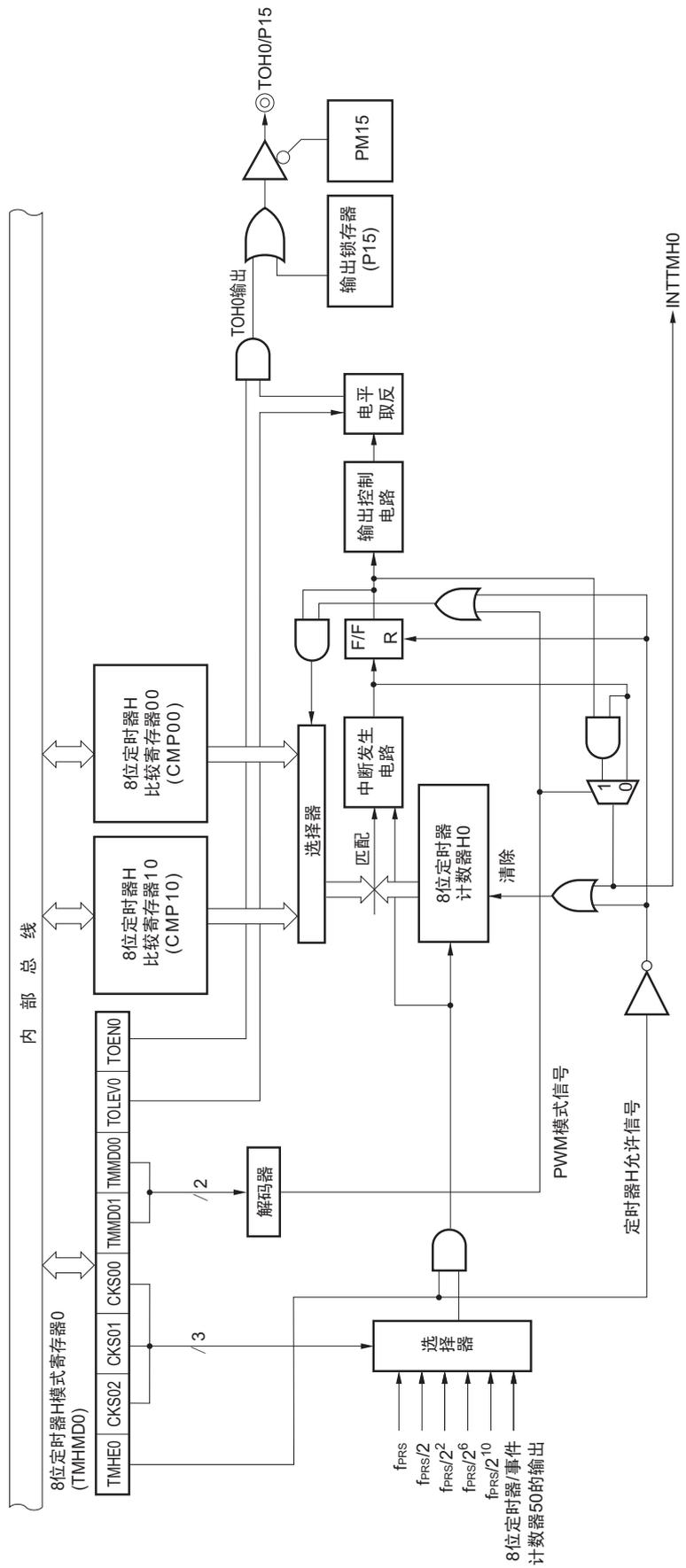
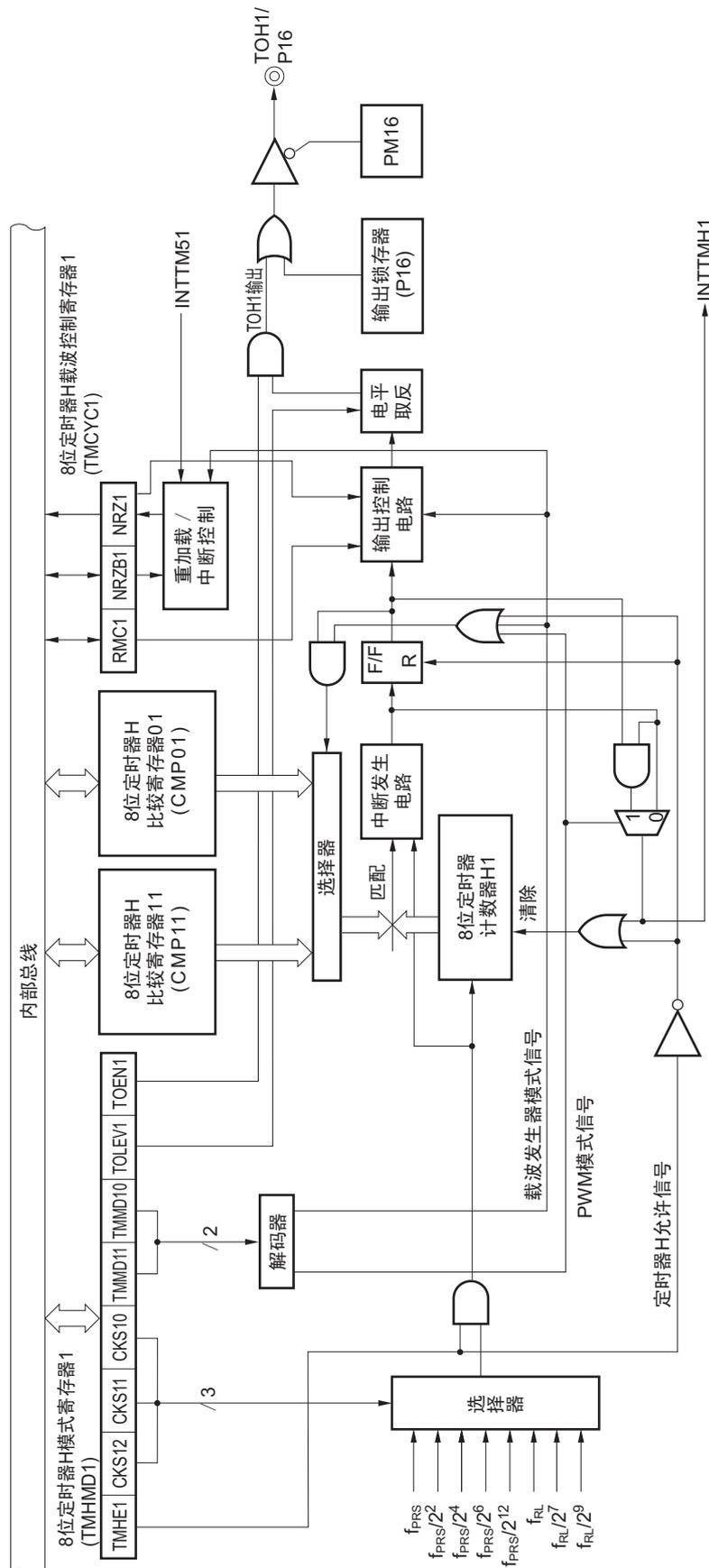


图 8-2 8 位定时器 H1 的框图



(1) 8 位定时器 H 比较寄存器 0n (CMP0n)

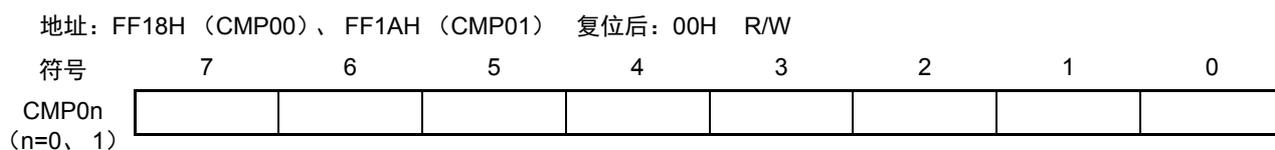
8 位定时器 H 比较寄存器 0n 是可通过 8 位存储器操作指令读写的寄存器。用于所有定时器运行模式。

CMP0n 的设置值总是和 8 位定时器计数器 Hn 的计数值进行比较，如果两个值匹配，就产生中断请求信号 (INTTMHn)，并且取反 TOHn 的输出电平。

在定时器停止 (TMHEn=0) 期间改写 CMP0n。

在产生复位信号后，CMP0n 变为“00H”。

图 8-3 8 位定时器 H 比较寄存器 0n (CMP0n) 的格式



注意 不能在定时器计数运行期间改写 CMP0n。但是，可在定时器计数运行期间刷新 CMP0n (写入相同的值)。

(2) 8 位定时器 H 比较寄存器 1n (CMP1n)

8 位定时器 H 比较寄存器 1n 是可通过 8 位存储器操作指令读写的寄存器。用于 PWM 输出模式和载波发生器模式。

在 PWM 输出模式中，CMP1n 的设置值总是和 8 位定时器计数器 Hn 的计数值进行比较，如果两个值匹配，就取反 TOHn 的输出电平，不产生中断请求信号。

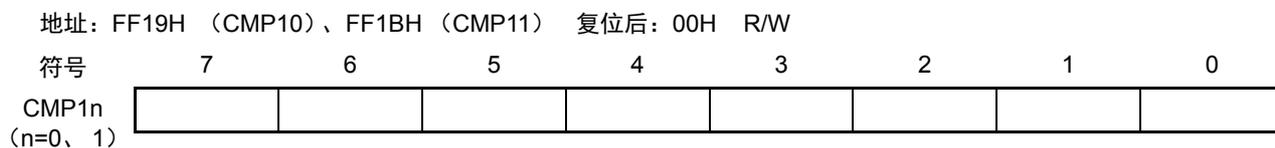
在载波发生器模式中，CMP1n 的设置值总是和 8 位定时器计数器 Hn 的计数值进行比较，如果两个值匹配，就产生中断请求信号 (INTTMHn)，并且同时清除计数值。

在定时器计数运行期间，可刷新 (写入相同的值) 和改写 CMP1n。

如果在定时器计数运行期间改写 CMP1n 的值，就锁存该值，并且在计数值与更改前的 CMP1n 值匹配时传送到 CMP1n，由此更改 CMP1n 的值。如果计数值和 CMP1n 值的匹配时序与 CMP1n 的写入时序发生竞争，就不更改 CMP1n 的值。

在产生复位信号后，CMP1n 变为“00H”。

图 8-4 8 位定时器 H 比较寄存器 1n (CMP1n) 的格式



注意 在 PWM 输出模式和载波发生器模式中，如果要在定时器计数运行停止 (TMHEn=0) 后重新开始定时器计数运行 (TMHEn=1)，必须设置 CMP1n (即使设置与 CMP1n 相同的值，也必须重新设置)。

备注 n=0、1

8.3 8 位定时器 H0、H1 的控制寄存器

8 位定时器 H0、H1 的控制寄存器有以下 4 种。

- 8 位定时器 H 模式寄存器 n (TMHMDn)
- 8 位定时器 H 载波控制寄存器 1 (TMCYC1) 注
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

注 只限 8 位定时器 H1。

(1) 8 位定时器 H 模式寄存器 n (TMHMDn)

8 位定时器 H 模式寄存器 n 控制定时器 H 的模式。
通过 1 位或 8 位存储器操作指令设置 TMHMDn。
在产生复位信号后，TMHMDn 变为“00H”。

备注 n=0、1

图 8-5 8 位定时器 H 模式寄存器 0 (TMHMD0) 的格式

地址: FF69H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	<1>	<0>
TMHMD0	TMHE0	CKS02	CKS01	CKS00	TMMD01	TMMD00	TOLEV0	TOEN0

TMHE0	允许定时器运行
0	停止定时器计数运行 (计数器清“0”)
1	允许定时器计数运行 (通过输入时钟开始计数运行)

CKS02	CKS01	CKS00	选择计数时钟			
			$f_{PRS}=2\text{MHz}$	$f_{PRS}=5\text{MHz}$	$f_{PRS}=10\text{MHz}$	
0	0	0	f_{PRS}	2MHz	5MHz	10MHz
0	0	1	$f_{PRS}/2$	1MHz	2.5MHz	5MHz
0	1	0	$f_{PRS}/2^2$	500kHz	1.25MHz	2.5MHz
0	1	1	$f_{PRS}/2^6$	31.25kHz	78.13kHz	156.25kHz
1	0	0	$f_{PRS}/2^{10}$	1.95kHz	4.88kHz	9.77kHz
1	0	1	TM50 的输出注			
上述以外			禁止设置			

TMMD01	TMMD00	定时器运行模式
0	0	间隔定时器模式
1	0	PWM 输出模式
上述以外		禁止设置

TOLEV0	控制定时器输出电平 (默认时)
0	低电平
1	高电平

TOEN0	控制定时器输出
0	禁止输出
1	允许输出

注 选择 TM50 的输出作为计数时钟时, 必须注意一下内容。

- TM50 和 CR50 匹配时的清除 & 启动模式 (TMC506=0)

允许定时器 F/F 的取反运行 (TMC501=1), 并且事先开始 8 位定时器 / 事件计数器 50 的运行。

- PWM 模式 (TMC506=1)

设置为占空比 50% 的时钟, 并且事先开始 8 位定时器 / 事件计数器 50 的运行。

在其中任一模式下, 都不需要允许 TO50 输出 (TOE50=1)。

注意 1. TMHE0=1 时, 禁止设置 TMHMD0 的其他位。但是, 可刷新 (写入相同的值)。

2. 在 PWM 输出模式和载波发生器模式中, 如果要在定时器计数运行停止 (TMHE0=0) 后重新开始定时器计数运行 (TMHE0=1), 必须设置 8 位定时器 H 比较寄存器 10 (CMP10) (即使设定与 CMP10 相同的值, 也必须重新设置)。
3. 实际的 TOH0/P15 引脚的 TOH0 输出取决于 PM15 和 P5。

备注 1. f_{PRS} : 外围硬件时钟频率

2. TMC506: 8 位定时器模式控制寄存器 50 (TMC50) 的 bit6

3. TMC501: TMC50 的 bit1

图 8-6 8 位定时器 H 模式寄存器 1 (TMHMD1) 的格式

地址: FF6CH 复位后: 00H R/W

符号	<7>	6	5	4	3	2	<1>	<0>
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	允许定时器运行
0	停止定时器计数运行 (计数器清“0”)
1	允许定时器计数运行 (通过输入时钟开始计数运行)

CKS12	CKS11	CKS10		选择计数时钟		
				$f_{PRS}=2\text{MHz}$	$f_{PRS}=5\text{MHz}$	$f_{PRS}=10\text{MHz}$
0	0	0	f_{PRS}	2MHz	5MHz	10MHz
0	0	1	$f_{PRS}/2^2$	500kHz	1.25MHz	2.5MHz
0	1	0	$f_{PRS}/2^4$	125kHz	312.5kHz	625kHz
0	1	1	$f_{PRS}/2^6$	31.25kHz	78.13kHz	156.25kHz
1	0	0	$f_{PRS}/2^{12}$	0.49kHz	1.22kHz	2.44kHz
1	0	1	$f_{RL}/2^7$	1.88kHz(TYP.)		
1	1	0	$f_{RL}/2^9$	0.47kHz(TYP.)		
1	1	1	f_{RL}	240kHz(TYP.)		

TMMD11	TMMD10	定时器运行模式
0	0	间隔定时器模式
0	1	载波发生器模式
1	0	PWM 输出模式
1	1	禁止设置

TOLEV1	控制定时器输出电平 (默认时)
0	低电平
1	高电平

TOEN1	控制定时器输出
0	禁止输出
1	允许输出

注意 1. TMHE1=1 时, 禁止设置 TMHMD1 的其他位。但是, 可刷新 (写入相同的值)。

- 在 PWM 输出模式和载波发生器模式中, 如果要在定时器计数运行停止 (TMHE1=0) 后重新开始定时器计数运行 (TMHE1=1), 必须设置 8 位定时器 H 比较寄存器 11 (CMP11) (即使设定与 CMP11 相同的值, 也必须重新设置)。
- 使用载波发生器模式时, 必须将 TMH1 的计数时钟频率设置为至少 6 倍的 TM51 计数时钟频率。
- 实际的 TOH1/P16 引脚的 TOH1 输出取决于 PM16 和 P16。

备注 1. f_{PRS} : 外围硬件时钟频率

- f_{RL} : 低速内部振荡时钟频率

(2) 8 位定时器 H 载波控制寄存器 1 (TMCYC1)

8 位定时器 H 载波控制寄存器控制 8 位定时器 H1 的遥控输出和载波脉冲输出的状态。

通过 1 位或 8 位存储器操作指令设置 TMCYC1。

在产生复位信号后，TMCYC1 变为“00H”。

图 8-7 8 位定时器 H 载波控制寄存器 1 (TMCYC1) 的格式

地址: FF6DH 复位后: 00H R/W 注

符号	7	6	5	4	3	2	1	<0>
TMCYC1	0	0	0	0	0	RMC1	NRZB1	NRZ1

RMC1	NRZB1	遥控输出
0	0	低电平输出
0	1	在 INTTM51 信号输入的上升沿进行高电平输出
1	0	低电平输出
1	1	在 INTTM51 信号输入的上升沿进行载波脉冲输出

NRZ1	载波脉冲输出状态标志
0	禁止载波输出的状态 (低电平状态)
1	允许载波输出的状态 (RMC1=1: 载波脉冲输出, RMC1=0: 高电平状态)

注 bit0 为只读位。

注意 在 TMHE=1 时，不能改写 RMC1。但是，可刷新 TMCYC1 (写入相同的值)。

(3) 端口模式寄存器 1 (PM1)

端口模式寄存器 1 以 1 位单位设置端口 1 的输入 / 输出。

在将 P15/TOH0 引脚用作定时器输出时，必须将 PM15 和 P15 的输出锁存器置“0”。

在将 P16/TOH1 引脚用作定时器输出时，必须将 PM16 和 P16 的输出锁存器置“0”。

通过 1 位或 8 位存储器操作指令设置 PM1。

在产生复位信号后，PM1 变为“FFH”。

图 8-8 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 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

PM1n	选择 P1n 引脚的输入 / 输出模式 (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)

8.4 8 位定时器 H0、H1 的运行

8.4.1 作为间隔定时器 / 方波输出的运行

在 8 位定时器计数器 H_n 和比较寄存器 0_n (CMP0n) 匹配时, 产生中断请求信号 (INTTMHn), 并且将 8 位定时器计数器 H_n 清“00H”。

在间隔定时器模式中不使用比较寄存器 1_n (CMP1n)。因此, 即使设置 CMP1n 寄存器, 也不能检测出 8 位定时器计数器 H_n 和 CMP1n 寄存器匹配, 所以不影响定时器输出。

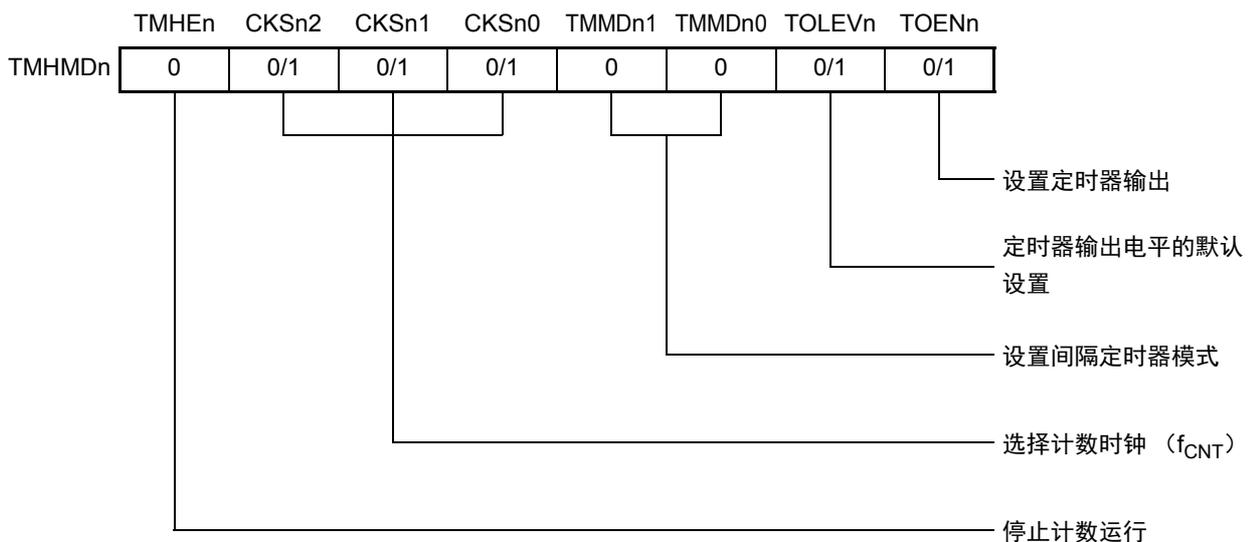
通过将定时器 H 模式寄存器 1 (TMHMD1) 的 bit0 (TOEN1) 置“1”, 从 TOH1 输出任意频率的方波 (占空比 =50%)。

设置方法

- ① 设置各寄存器。

图 8-9 间隔定时器 / 方波输出运行时的寄存器设置

- (i) 设置定时器 H 模式寄存器 n (TMHMDn)



- (ii) 设置 CMP0n 寄存器

将 N 设置为比较值时, 间隔时间如下所示。

- 间隔时间 = $(N+1)/f_{CNT}$

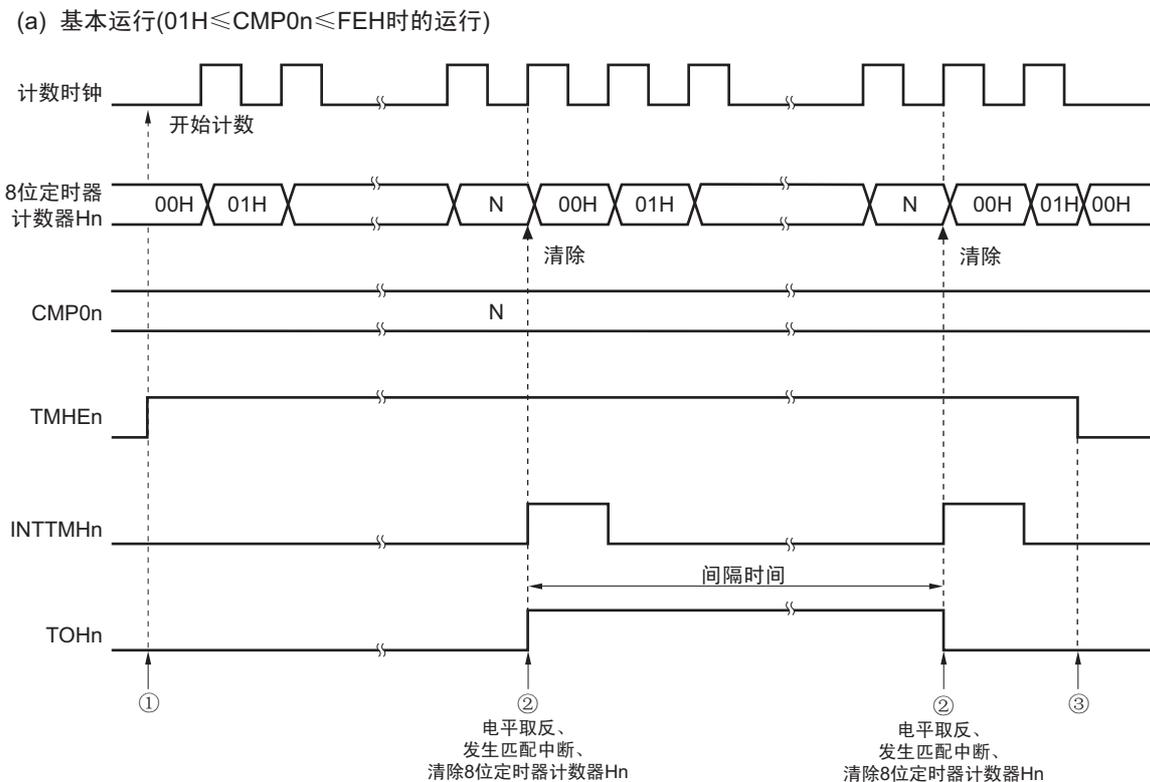
- ② 通过将 TMHE_n 置“1”, 开始计数运行。
- ③ 如果 8 位定时器计数器 H_n 和 CMP0n 寄存器的值匹配, 就产生 INTTMH_n 信号, 并且将 8 位定时器计数器 H_n 清“00H”。
- ④ 之后, 以相同间隔产生 INTTMH_n 信号。如果要停止计数运行, 将 TMHE_n 置“0”。

备注 1. 有关输出引脚的设置, 请参照“8.3 (3) 端口模式寄存器 1 (PM1)”。

2. 有关 INTTMH_n 信号的中断允许, 请参照“第 16 章 中断功能”。

3. n=0、1

图 8-10 间隔定时器 / 方波输出运行的时序 (1/2)

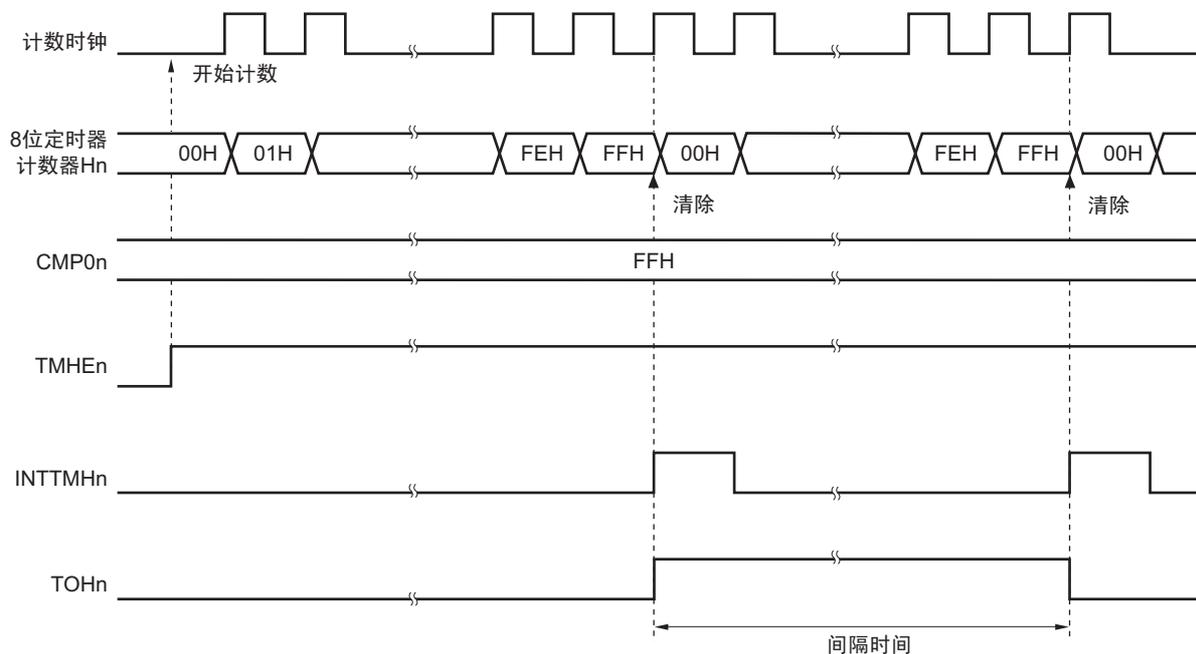


- ① 通过将 $TMHE_n$ 位从“0”置“1”，进入允许计数运行的状态。在允许运行后，计数时钟最迟 1 个时钟后开始计数。
- ② 如果 8 位定时器计数器 H_n 的值与 $CMP0_n$ 寄存器的值匹配，就清除 8 位定时器计数器 H_n 的值，并且取反 TOH_n 输出的电平。另外，在计数时钟上升时产生 $INTTMH_n$ 信号。
- ③ 如果在定时器 H 运行期间将 $TMHE_n$ 位置“0”， $INTTMH_n$ 信号和 TOH_n 输出就为默认状态。如果在将 $TMHE_n$ 位置“0”前，已经是与默认状态相同的状态，就保持电平。

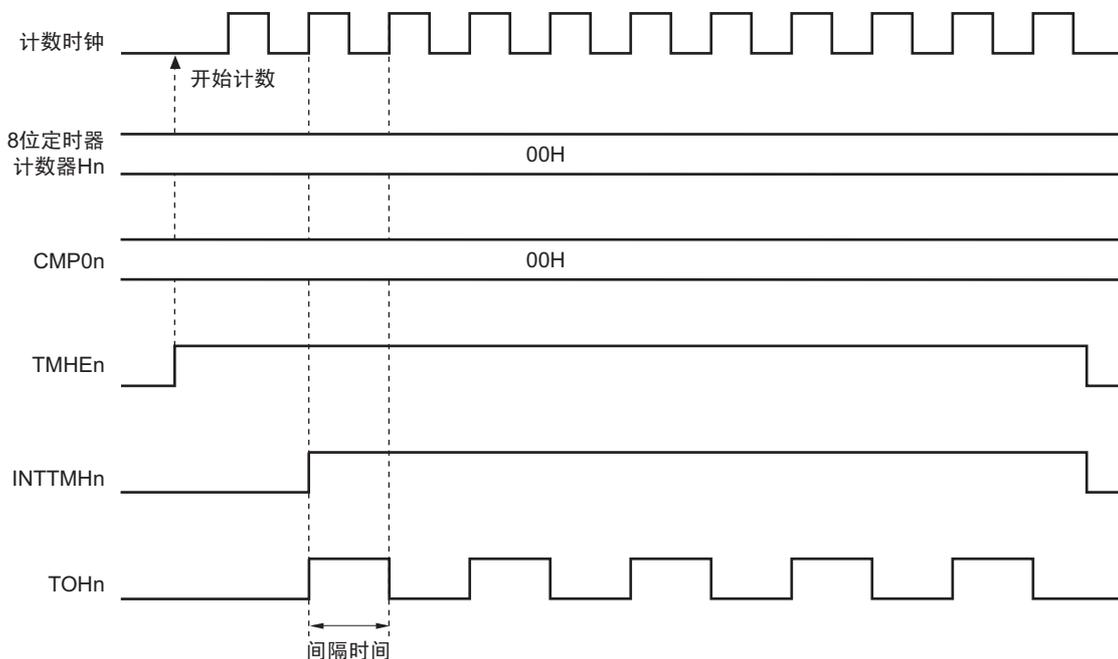
备注 $01H \leq N \leq FEH$
 $n=0、1$

图 8-10 间隔定时器 / 方波输出运行的时序 (2/2)

(b) CMP0n=FFH时的运行



(c) $CMP0n=00H$ 时的运行



备注 n=0、1

8.4.2 PWM 的输出运行

可在 PWM 输出模式中输出任意占空比和周期的脉冲。

8 位定时器比较寄存器 0n (CMP0n) 控制定时器输出 (TOHn) 的周期。定时器运行期间禁止改写 CMP0n 寄存器。

8 位定时器比较寄存器 1n (CMP1n) 控制定时器输出 (TOHn) 的占空比。可在定时器运行作期间改写 CMP1n 寄存器。

在 PWM 输出模式中的运行如下所示。

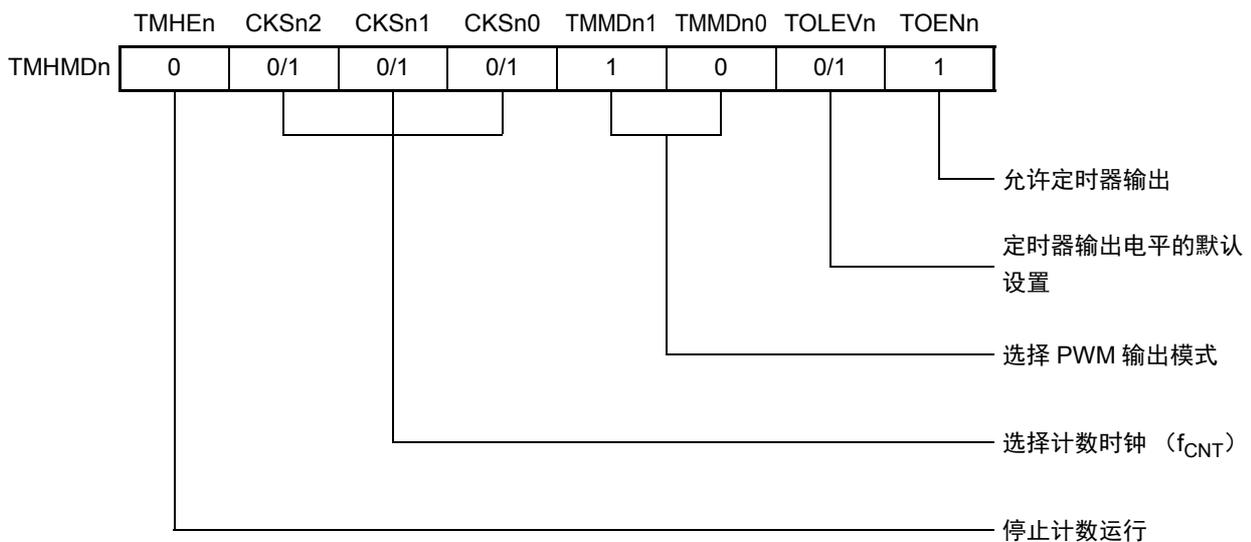
开始定时器计数后，如果 8 位定时器计数器 Hn 与 CMP0n 寄存器的值匹配，PWM 输出 (TOHn 输出) 就为有效电平输出，并且将 8 位定时器计数器 Hn 清“0”。如果 8 位定时器计数器 Hn 与 CMP1n 寄存器的值匹配，PWM 输出 (TOHn 输出) 就为无效电平输出。

设置方法

- ① 设置各寄存器。

图 8-11 PWM 输出模式中的寄存器设置

- (i) 设置定时器 H 模式寄存器 n (TMHMDn)



- (ii) 设置 CMP0n 寄存器

- 比较值(N): 设置周期

- (iii) 设置 CMP1n 寄存器

- 比较值(M): 设置占空比

备注 $00H \leq \text{CMP1n}(M) < \text{CMP0n}(N) \leq \text{FFH}$
n=0、1

- ② 通过将 TMHEn 置“1”，开始计数运行。
- ③ 在允许计数运行后，第 1 个用于比较的比较寄存器是 CMP0n 寄存器。如果 8 位定时器计数器 Hn 与 CMP0n 寄存器的值匹配，就清除 8 位定时器计数器 Hn，产生中断请求信号（INTTMHn），并且输出有效电平。同时，将与 8 位定时器计数器 Hn 比较的比较寄存器从 CMP0n 寄存器切换到 CMP1n 寄存器。
- ④ 如果 8 位定时器计数器 Hn 与 CMP1n 寄存器匹配，输出无效电平。同时，将与 8 位定时器计数器 Hn 比较的比较寄存器从 CMP1n 寄存器切换到 CMP0n 寄存器。此时，不清除 8 位定时器计数器 Hn，也不产生 INTTMHn 信号。
- ⑤ 通过重复执行上述步骤③和④，可获取任意占空比的脉冲。
- ⑥ 如果要停止计数运行，将 TMHEn 置“0”。

设 CMP0n 寄存器的设置值为 N、CMP1n 寄存器的设置值为 M、计数时钟频率为 f_{CNT} ，则 PWM 脉冲输出周期和占空比如下所示。

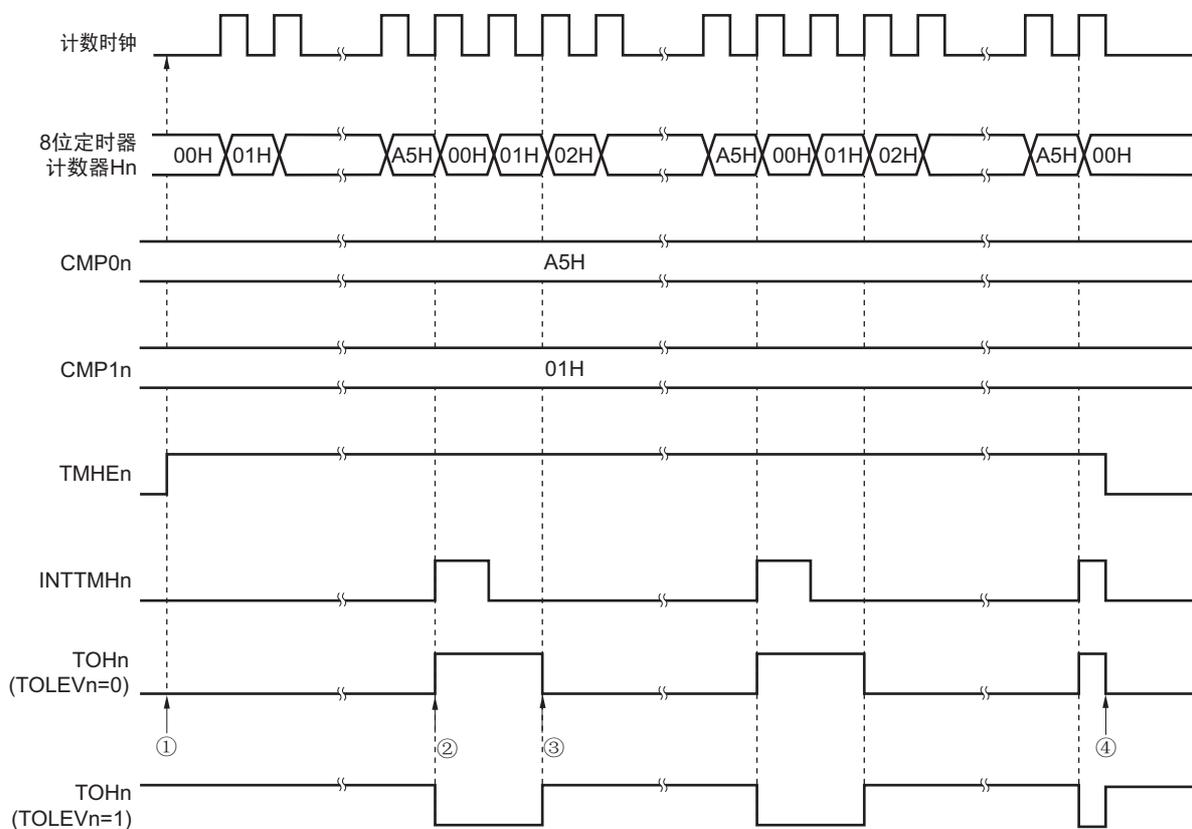
- PWM 脉冲输出周期 $= (N+1)/f_{CNT}$
- 占空比 $= (M+1)/(N+1)$

- 注意 1. 在定时器计数运行期间，可更改 CMP1n 寄存器的设置值。但是，从更改 CMP1n 寄存器的值到将该值传送到寄存器前，至少需要 3 个运行时钟（通过 TMHMDn 寄存器的 CKSn2 ~ CKSn0 位选择的信号）。
2. 要在定时器计数运行停止（TMHEn=0）后重新开始定时器计数运行（TMHEn=1）时，必须设置 CMP1n 寄存器（即使设定与 CMP1n 相同的值，也必须重新设置）。
3. CMP1n 寄存器的设定值（M）和 CMP0n 寄存器的设定值（N）必须在以下范围内。
- $$00H \leq \text{CMP1n}(M) < \text{CMP0n}(N) \leq \text{FFH}$$

- 备注 1. 有关输出引脚的设置，请参照“8.3 (3) 端口模式寄存器 1（PM1）”。
2. 有关 INTTMHn 信号的中断允许，请参照“第 16 章 中断功能”。
3. n=0、1

图 8-12 PWM 输出运行的时序 (1/4)

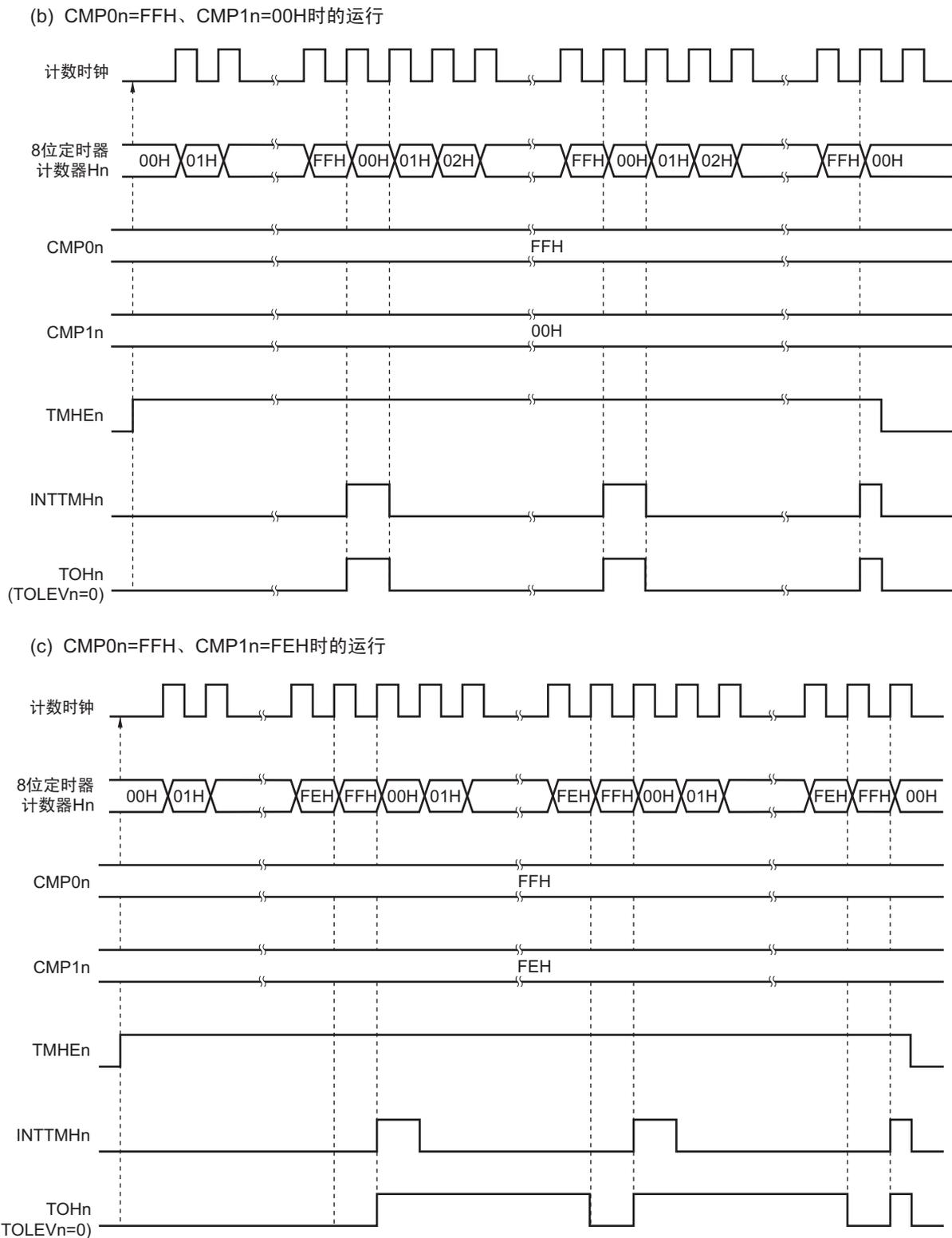
(a) 基本运行



- ① 通过将 $TMHEn$ 置“1”，进入允许计数运行的状态。通过屏蔽 1 个计数时钟，启动 8 位定时器计数器 H_n ，并且进行递增计数。此时，PWM 输出为无效电平输出。
- ② 如果 8 位定时器计数器 H_n 的值与 $CMP0n$ 寄存器的值匹配，输出有效电平。此时，清除 8 位定时器计数器 H_n ，并且产生 $INTTMH_n$ 信号。
- ③ 如果 8 位定时器计数器 H_n 的值与 $CMP1n$ 寄存器的值匹配，输出无效电平。此时，不清除 8 位定时器计数器的值，也不产生 $INTTMH_n$ 信号。
- ④ 通过在定时器 H_n 运行期间将 $TMHEn$ 位置“0”， $INTTMH_n$ 信号为默认状态，并且 PWM 输出为无效电平输出。

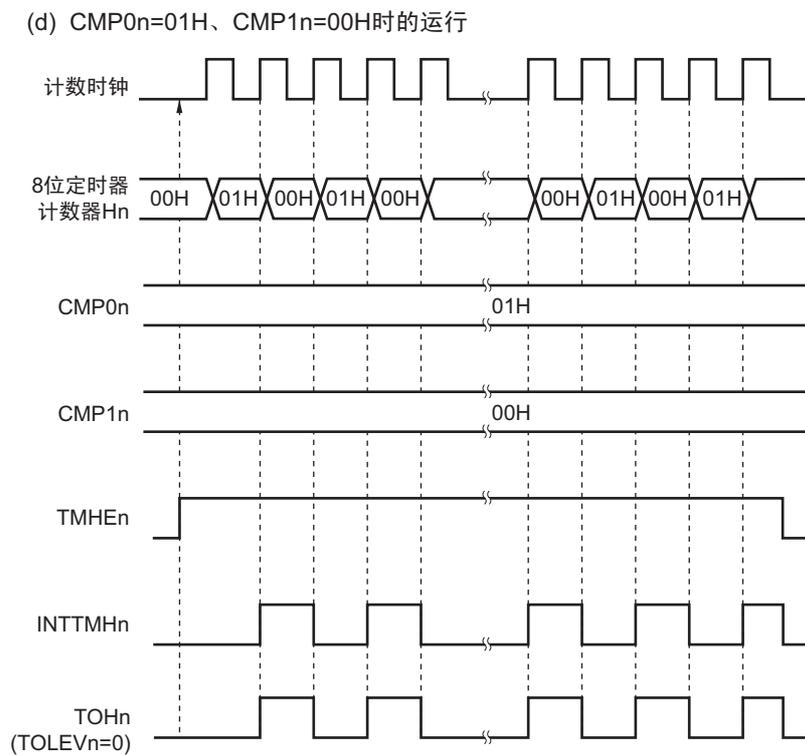
备注 n=0、1

图 8-12 PWM 输出运行的时序 (2/4)



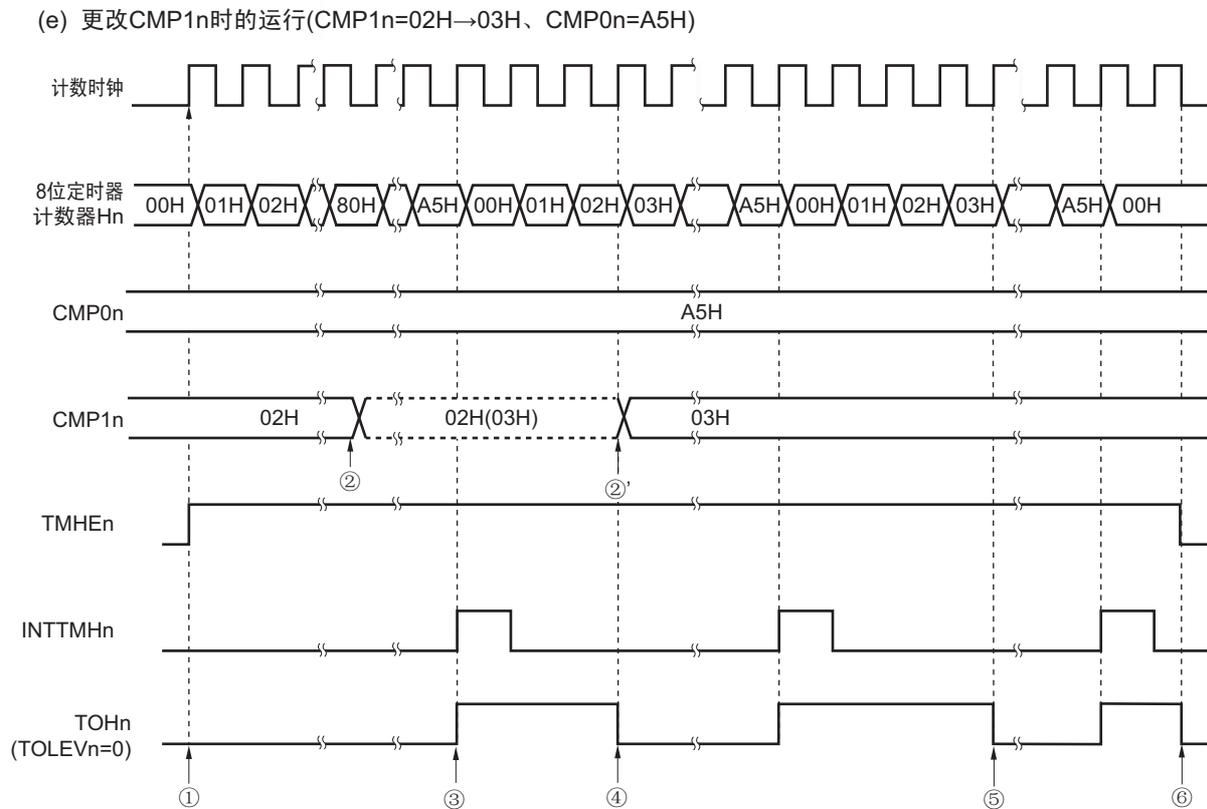
备注 n=0、1

图 8-12 PWM 输出运行的时序 (3/4)



备注 n=0、1

图 8-12 PWM 输出运行的时序 (4/4)



- ① 通过将TMHEn置“1”，进入允许计数运行的状态。通过屏蔽1个计数时钟，启动8位定时器计数器Hn，并且进行递增计数。此时，PWM输出为无效电平输出。
- ② 在定时器计数器运行期间，可更改CMP1n寄存器的设置值。该运行不与计数时钟同步。
- ③ 如果8位定时器计数器Hn的值与CMP0n寄存器的值匹配，就清除8位定时器计数器Hn，输出有效电平，并且产生INTTMHn信号。
- ④ 即使更改CMP1n寄存器的值，也将该值锁存，并且不传送到寄存器。如果8位定时器计数器Hn的值与CMP1n寄存器更改前的值匹配，就将值传送到CMP1n寄存器，从而更改CMP1n寄存器的值（②'）。但是，从更改CMP1n寄存器的值到将该值传送到寄存器前，至少需要3个计数时钟。即使在3个计数时钟内产生匹配信号，也不能传送更改值到寄存器。
- ⑤ 如果8位定时器计数器Hn的值与更改后的CMP1n寄存器的值匹配，输出无效电平。不清除8位定时器计数器Hn，也不产生INTTMHn信号。
- ⑥ 通过在定时器Hn运行期间将TMHEn位置“0”，INTTMHn信号为默认状态，PWM输出为无效电平输出。

备注 n=0、1

8.4.3 作为载波发生器的运行（只限 8 位定时器 H1）

在载波发生器模式中，8 位定时器 H1 用于生成红外线遥控的载波时钟信号，8 位定时器 / 事件计数器 51 用于生成红外线遥控信号（时间计数）。

以 8 位定时器 / 事件计数器 51 设置的周期输出由 8 位定时器 H1 生成的载波时钟。

在载波发生器模式中，通过 8 位定时器 / 事件计数器 51 控制 8 位定时器 H1 的载波脉冲输出量，并且从 TOH1 输出载波脉冲。

(1) 载波的生成

在载波发生器模式中，8 位定时器 H 比较寄存器 01（CMP01）生成载波脉冲的低电平宽度波形，8 位定时器 H 比较寄存器 11（CMP11）生成载波脉冲的高电平宽度波形。

在 8 位定时器 H1 运行期间，可改写 CMP11 寄存器，但禁止改写 CMP01 寄存器。

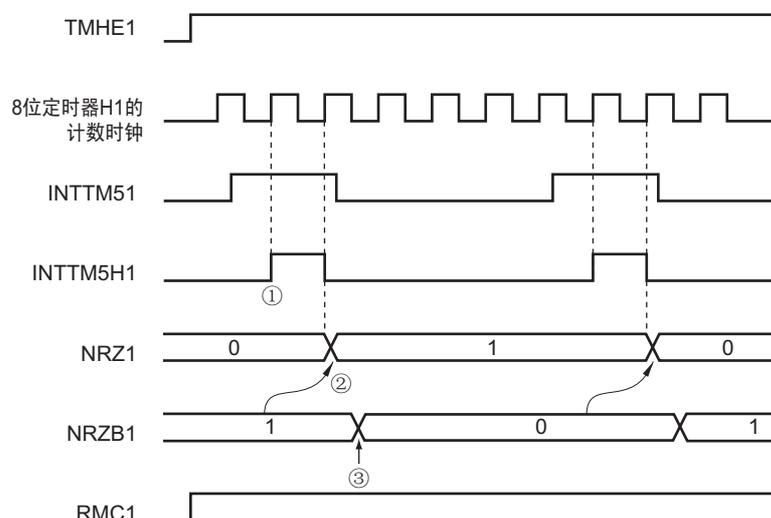
(2) 载波的输出控制

通过 8 位定时器 / 事件计数器 51 的中断请求信号（INTTM51）和 8 位定时器 H 载波控制寄存器（TMCYC1）的 NRZB1 位和 RMC1 位控制载波的输出。输出的关系如下所示。

RMC1 位	NRZB1 位	输出
0	0	低电平输出
0	1	在 INTTM51 信号输入的上升沿进行高电平输出
1	0	低电平输出
1	1	在 INTTM51 信号输入的上升沿进行载波脉冲输出

为了在计数运行期间控制载波脉冲输出，TMCYC1 寄存器的 NRZ1 位和 NRZB1 位由主位和从位构成。NRZ1 位只读，而 NRZB1 位可读写。通过 8 位定时器 H1 计数时钟同步 INTTM51 信号后，作为 INTTM5H1 信号输出。INTTM5H1 信号作为 NRZ1 位的数据传送信号，将 NRZB1 位的值传送到 NRZ1 位。从 NRZB1 位到 NRZ1 位的传送时序如下所示。

图 8-13 传送时序



- ① 通过 8 位定时器 H1 的计数时钟同步 INTTM51 信号后，作为 INTTM5H1 信号输出。
- ② 在从 INTTM5H1 信号的上升沿开始的第 2 个时钟处，将 NRZB1 位的值传送到 NRZ1 位。
- ③ 在通过 INTTM5H1 中断启动的中断处理编程期间，或者在通过轮询中断请求标志确认时序后，将下一个值写入 NRZB1 位。另外，将下一个时间的计数数据写入 CR51 寄存器。

- 注意 1. 从改写 NRZB1 位的值到第 2 个时钟前，不能再次改写 NRZB1 位的值。如果改写，不能保证从 NRZB1 位到 NRZ1 位的传送运行。
2. 如果在载波发生器模式中使用 8 位定时器 / 事件计数器 51，就在①的时序发生中断。如果在载波发生器模式以外的模式中使用 8 位定时器 / 事件计数器 51，则中断的发生时序不同。

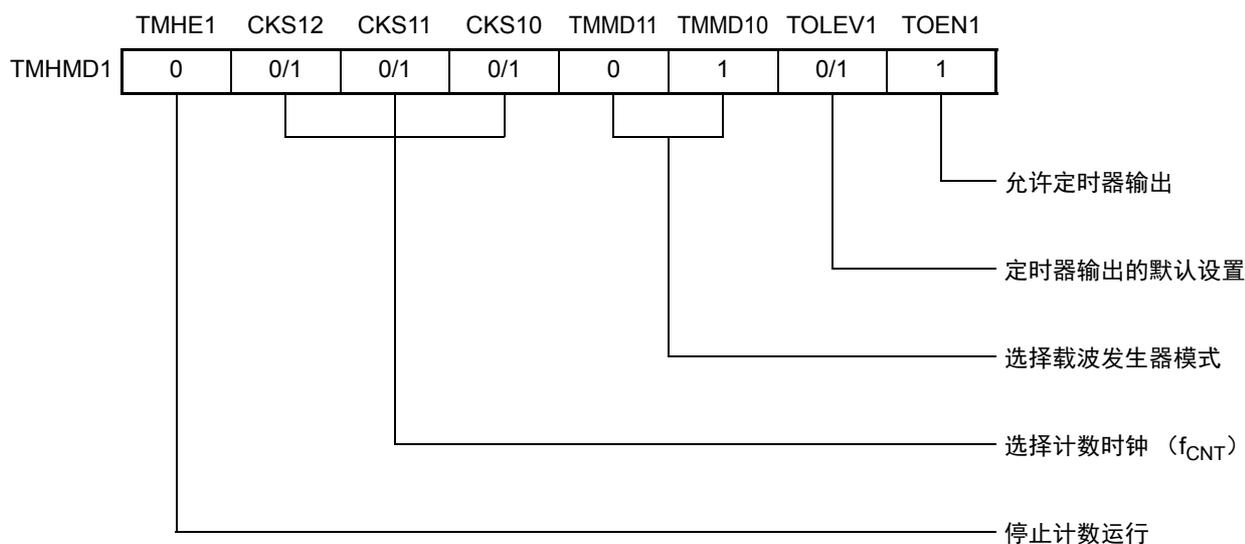
备注 INTTM5H1 是内部信号，不是中断源。

设置方法

- ① 设置各寄存器。

图 8-14 载波发生器模式中的寄存器设置

- (i) 设置 8 位定时器 H 模式寄存器 1 (TMHMD1)



- (ii) 设置 CMP01 寄存器
 - 比较值
- (iii) 设置 CMP11 寄存器
 - 比较值
- (iv) 设置 TMCYC1 寄存器
 - RMC1=1... 遥控输出允许位
 - NRZB1=0/1... 载波输出允许位
- (v) 设置 TCL51 和 TMC51 寄存器
 - 参照“7.3 8 位定时器 / 事件计数器 50、51 的控制寄存器”。

- ② 通过将 TMHE1 置“1”，开始 8 位定时器 H1 的计数运行。
- ③ 如果将 8 位定时器模式控制寄存器 51（TMC51）的 TCE51 置“1”，就开始 8 位定时器/事件计数器 51 的计数运行。
- ④ 在允许计数运行后，第 1 个用于比较的比较寄存器是 CMP01 寄存器。如果 8 位定时器计数器 H1 的计数值与 CMP01 寄存器的值匹配，就产生 INTTMH1，并且清除 8 位定时器计数器 H1。同时，将与 8 位定时器计数器 H1 比较的比较寄存器从 CMP01 寄存器切换到 CMP11 寄存器。
- ⑤ 如果 8 位定时器计数器 H1 的计数值与 CMP11 寄存器的值匹配，产生 INTTMH1 信号，并且清除 8 位定时器计数器 H1。同时，将与 8 位定时器计数器 H1 比较的比较寄存器从 CMP11 寄存器切换到 CMP01 寄存器。
- ⑥ 通过重复执行上述步骤④和⑤，生成载波时钟。
- ⑦ 通过 8 位定时器 H1 的计数时钟同步 INTTM51 信号后，作为 INTTM5H1 信号输出。该信号为 NRZB1 的数据传送信号，将 NRZB1 位的值传送到 NRZ1 位。
- ⑧ 在通过 INTTM5H1 中断启动的中断处理编程期间，或者在通过轮询中断请求标志确认时序后，将下一个值写入 NRZB1 位。另外，将下一个时间的计数数据写入 CR51 寄存器。
- ⑨ 在 NRZ1 位为高电平时，从 TOH1 输出载波时钟。
- ⑩ 通过重复执行上述步骤，可获取任意载波时钟。如果要停止计数运行，将 TMHE1 置“0”。

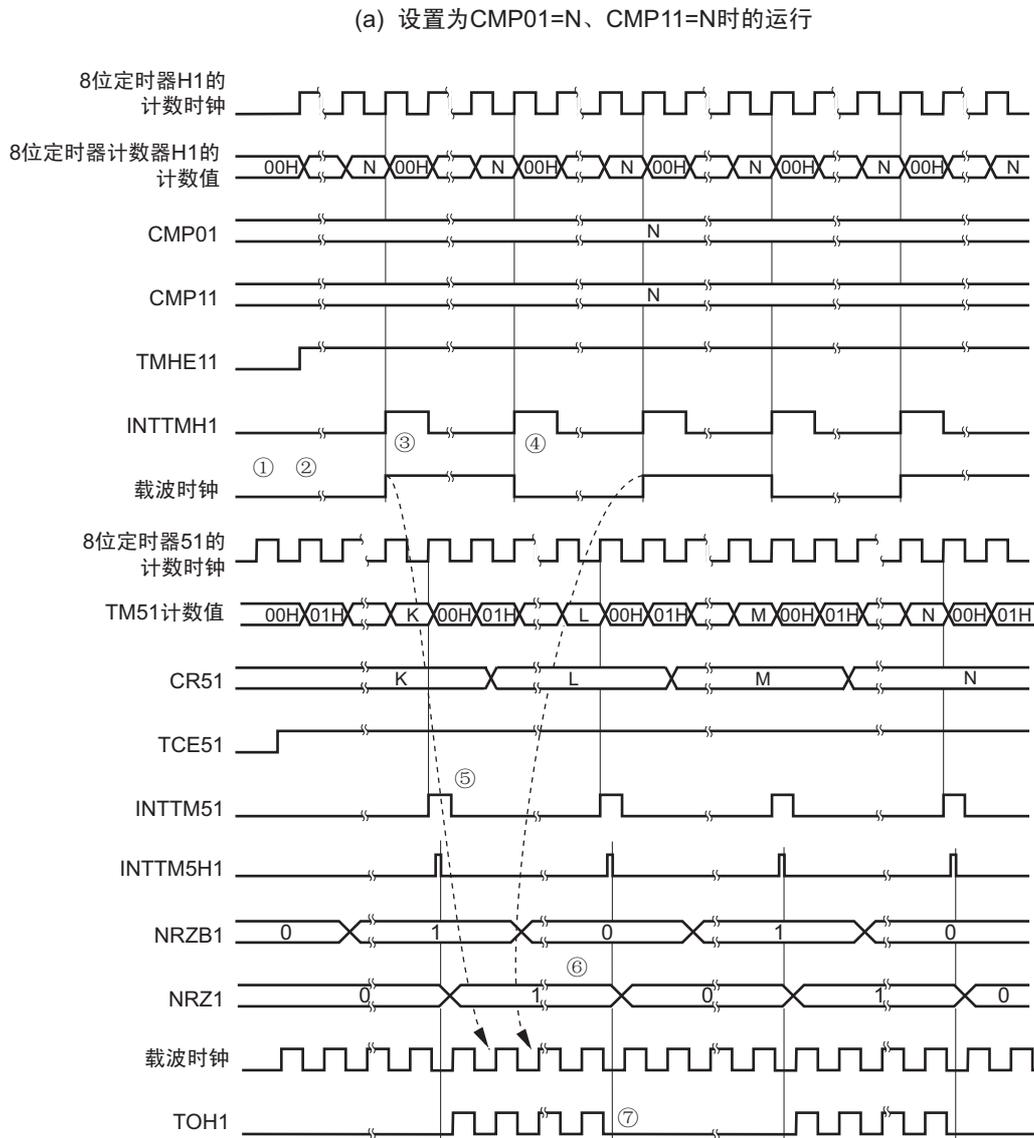
设 CMP01 寄存器的设置值为 N、CMP11 寄存器的设置值为 M、计数时钟频率为 f_{CNT} ，则载波时钟输出周期和占空比如下所示。

- 载波时钟输出周期 $= (N+M+2)/f_{CNT}$
- 占空比 $= \text{高电平宽度} / \text{载波时钟输出宽度} = (M+1)/(N+M+2)$

- 注意 1. 要在定时器计数运行停止（TMHE1=0）后重新开始定时器计数运行（TMHE1=1）时，必须设置 CMP11 寄存器（即使设置与 CMP11 相同的值，也必须重新设置）。
2. 必须将 TMH1 的计数时钟频率设置为至少 6 倍的 TM51 的计数时钟频率。
 3. CMP01 和 CMP11 寄存器的值必须设置在 01H ~ FFH 的范围内。
 4. 在定时器计数运行期间，可更改 CMP11 寄存器的设定值。但是，从更改 CMP11 的值到将该值传送到寄存器前，至少需要 3 个运行时钟（通过 TMHMD1 寄存器的 CKS12 ~ CKS10 位选择的信号）。
 5. 必须在开始计数运行前设置 RMC1 位。

- 备注 1. 有关输出引脚的设置，请参照“8.3 (3) 端口模式寄存器 1（PM1）”。
2. 有关 INTTMH1 信号的中断允许，请参照“第 16 章 中断功能”。

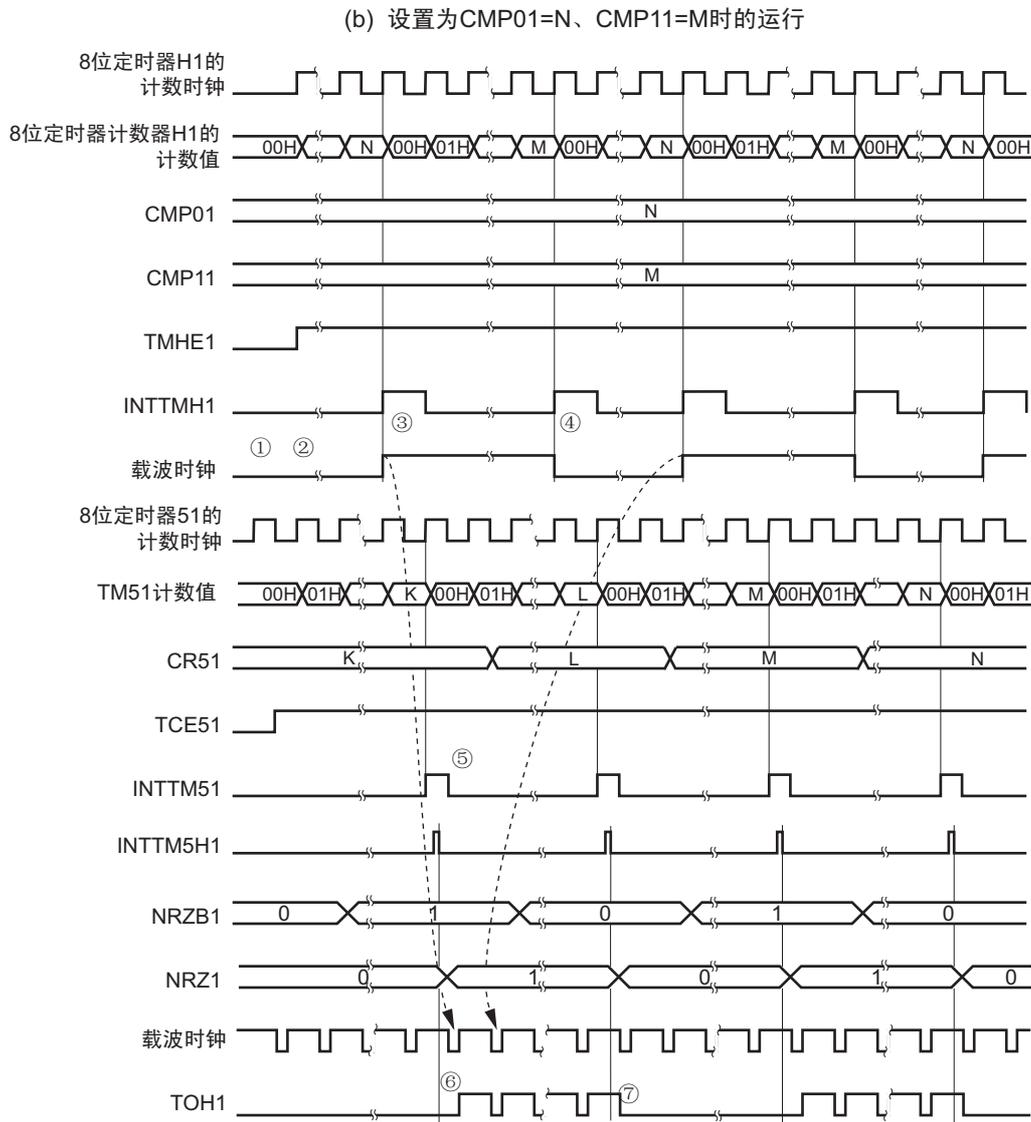
图 8-15 载波发生器模式的运行时序 (1/3)



- ① 在 $TMHE1=0$ 和 $TCE51=0$ 时，8 位定时器计数器 H1 的运行停止状态。
- ② 如果将 $TMHE1$ 置“1”，8 位定时器计数器 H1 就开始计数运行。此时，载波时钟保持默认状态。
- ③ 如果 8 位定时器计数器 H1 的计数值与 $CMP01$ 寄存器的值匹配，产生第 1 个 $INTTMH1$ 信号，取反载波时钟信号，并且将与 8 位定时器计数器 H1 比较的比较寄存器从 $CMP01$ 寄存器切换到 $CMP11$ 寄存器。同时，8 位定时器计数器 H1 被清“00H”。
- ④ 如果 8 位定时器计数器 H1 的计数值与 $CMP11$ 寄存器的值匹配，产生 $INTTMH1$ 信号，取反载波时钟信号，并且将与 8 位定时器计数器 H1 比较的比较寄存器从 $CMP11$ 切换到 $CMP01$ 寄存器。同时，8 位定时器计数器 H1 被清“00H”。通过重复执行步骤③和④，生成占空比固定为 50% 的载波时钟。
- ⑤ 一旦产生 $INTTM51$ 信号，就通过 8 位定时器 H1 的计数时钟同步该信号，然后作为 $INTTM5H1$ 信号输出。
- ⑥ $INTTM5H1$ 信号为 $NRZB1$ 位的数据传送信号，将 $NRZB1$ 位的值传送到 $NRZ1$ 位。
- ⑦ 通过将 $NRZ1$ 置“0”， $TOH1$ 输出为低电平输出。

备注 $INTTM5H1$ 是内部信号，不是中断源。

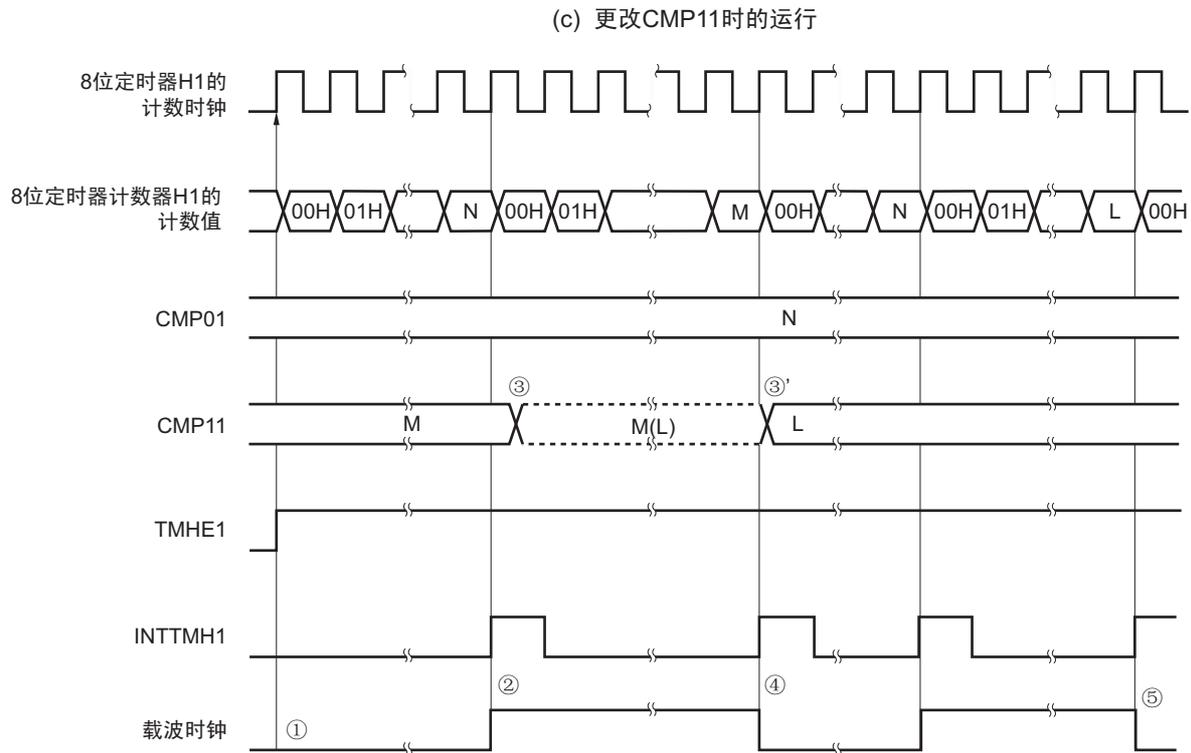
图 8-15 载波发生器模式的运行时序 (2/3)



- ① 在 $TMHE1=0$ 和 $TCE51=0$ 时，8 位定时器计数器 H1 的运行停止状态。
- ② 如果将 $TMHE1$ 置“1”，8 位定时器计数器 H1 就开始计数运行。此时，载波时钟保持默认状态。
- ③ 如果 8 位定时器计数器 H1 的计数值与 $CMP01$ 寄存器的值匹配，产生第 1 个 $INTTMH1$ 信号，取反载波时钟信号，并且将与 8 位定时器计数器 H1 比较的比较寄存器从 $CMP01$ 寄存器切换到 $CMP11$ 寄存器。同时，8 位定时器计数器 H1 被清“00H”。
- ④ 如果 8 位定时器计数器 H1 的计数值与 $CMP11$ 寄存器的值匹配，产生 $INTTMH1$ 信号，取反载波时钟信号，并且将与 8 位定时器计数器 H1 比较的比较寄存器从 $CMP11$ 切换到 $CMP01$ 寄存器。同时，8 位定时器计数器 H1 被清“00H”。通过重复执行步骤③和④，生成固定占空比（50% 除外）的载波时钟。
- ⑤ 一旦产生 $INTTM51$ 信号，就通过 8 位定时器 H1 的计数时钟同步该信号，然后作为 $INTTM5H1$ 信号输出。
- ⑥ 通过将 $NRZ1$ 置“1”，从第 1 个载波时钟的上升沿开始输出载波。
- ⑦ 通过将 $NRZ1$ 置“0”，在载波时钟的高电平期间， $TOH1$ 输出保持高电平，不转变为低电平（通过步骤在⑥和⑦，可保证载波波形的高电平宽度）。

备注 $INTTM5H1$ 是内部信号，不是中断源。

图 8-15 载波发生器模式的运行时序 (3/3)



- ① 如果将TMHE1置“1”，就开始计数运行。此时，载波时钟保持默认状态。
- ② 如果8位定时器计数器H1的计数值与CMP01寄存器的值匹配，产生INTTMH1信号，取反载波时钟信号，并且将8位定时器计数器H1清“00H”。同时，将与8位定时器计数器H1比较的比较寄存器从CMP01寄存器切换到CMP11寄存器。
- ③ CMP11寄存器不与计数时钟同步，可在8位定时器H1运行期间改写CMP11寄存器。但是，锁存该更改值（L）。在8位定时器计数器H1的计数值与CMP11寄存器更改前的值（M）匹配时，更改CMP11寄存器的值（③'）。
从更改CMP11寄存器的值到将该值传送到寄存器前，至少需要3个计数时钟。即使在3个计数时钟内产生匹配信号，也不能将更改值传送到寄存器。
- ④ 如果8位定时器计数器H1的计数值与更改前的CMP11寄存器的值（M）匹配，产生INTTMH1信号，取反载波时钟信号，并且将8位定时器计数器H1清“00H”。同时，将与8位定时器计数器H1比较的比较寄存器从CMP11切换到CMP01寄存器。
- ⑤ 在8位定时器计数器H1的计数值变为更改后的值（L）时，8位定时器计数器H1的计数值与CMP11寄存器的值再次匹配。

第9章 时钟定时器

9.1 时钟定时器的功能

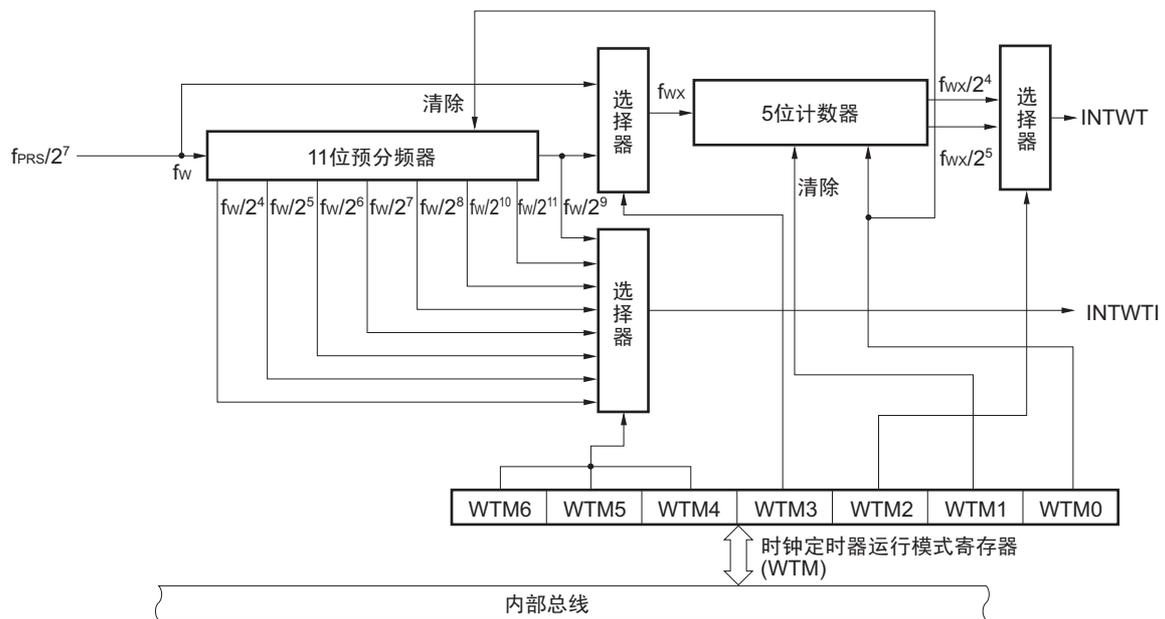
时钟定时器具有以下功能。

- 时钟定时器
- 间隔定时器

时钟定时器和间隔定时器可同时使用。

时钟定时器的框图如图9-1所示。

图9-1 时钟定时器的框图



- 备注 f_{PRS} : 外围硬件时钟频率
 f_W : 时钟定时器的时钟频率 ($f_{PRS}/2^7$)
 f_{WX} : f_W 或者 $f_W/2^9$

(1) 时钟定时器

可通过使用外围硬件时钟，以事先设置的时间间隔产生中断请求信号（INTWT）。

表 9-1 时钟定时器的中断时间

中断时间	以 $f_{PRS}=2\text{MHz}$ 运行时	以 $f_{PRS}=5\text{MHz}$ 运行时	以 $f_{PRS}=10\text{MHz}$ 运行时
$2^4/f_W$	1.02ms	410 μs	205 μs
$2^5/f_W$	2.05ms	819 μs	410 μs
$2^{13}/f_W$	0.52ms	0.210s	0.105s
$2^{14}/f_W$	1.05ms	0.419s	0.210s

备注 f_{PRS} : 外围硬件时钟频率

f_W : 时钟定时器的时钟频率 ($f_{PRS}/2^7$)

(2) 间隔定时器

以事先设置的时间间隔产生中断请求信号（INTWTI）。

表 9-2 间隔定时器的间隔时间

间隔时间	以 $f_{PRS}=2\text{MHz}$ 运行时	以 $f_{PRS}=5\text{MHz}$ 运行时	以 $f_{PRS}=10\text{MHz}$ 运行时
$2^4/f_W$	1.02ms	410 μs	205 μs
$2^5/f_W$	2.05ms	820 μs	410 μs
$2^6/f_W$	4.10ms	1.64ms	820 μs
$2^7/f_W$	8.20ms	3.28ms	1.64ms
$2^8/f_W$	16.4ms	6.55ms	3.28ms
$2^9/f_W$	32.8ms	13.1ms	6.55ms
$2^{10}/f_W$	65.5ms	26.2ms	13.1ms
$2^{11}/f_W$	131.1ms	52.4ms	26.2ms

备注 f_{PRS} : 外围硬件时钟频率

f_W : 时钟定时器的时钟频率 ($f_{PRS}/2^7$)

9.2 时钟定时器的结构

时钟定时器由以下硬件构成。

表 9-3 时钟定时器的结构

项目	结构
计数器	5 位 × 1 个
预分频器	11 位 × 1 个
控制寄存器	时钟定时器运行模式寄存器 (WTM)

9.3 时钟定时器的控制寄存器

时钟定时器的控制寄存器有时钟定时器运行模式寄存器（WTM）。

- 时钟定时器运行模式寄存器（WTM）

时钟定时器运行模式寄存器允许或者禁止时钟定时器的计数时钟运行，设置预分频器的间隔时间以及控制5位计数器的运行。

通过1位或8位存储器操作指令设置WTM。

在产生复位信号后，WTM变为“00H”。

图 9-2 时钟定时器运行模式寄存器（WTM）的格式

地址：FF6FH 复位后：00H R/W

符号	7	6	5	4	3	2	<1>	<0>
WTM	0	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM6	WTM5	WTM4	选择预分频器的间隔时间
0	0	0	$2^4/f_W$
0	0	1	$2^5/f_W$
0	1	0	$2^6/f_W$
0	1	1	$2^7/f_W$
1	0	0	$2^8/f_W$
1	0	1	$2^9/f_W$
1	1	0	$2^{10}/f_W$
1	1	1	$2^{11}/f_W$

WTM3	WTM2	选择时钟定时器的中断时间
0	0	$2^{14}/f_W$
0	1	$2^{13}/f_W$
1	0	$2^5/f_W$
1	1	$2^4/f_W$

WTM1	控制5位计数器的运行
0	运行停止后清除
1	启动

WTM0	允许时钟定时器的运行
0	停止运行（清除预分频器和5位计数器）
1	允许运行

- 注意 1. 在时钟定时器运行期间，不能更改计数时钟和间隔时间（通过WTM的bit4～bit6（WTM4～WTM6）设置）。
2. 必须将bit7置“0”。

备注 f_{PRS} ：外围硬件时钟频率

f_W ：时钟定时器的时钟频率（ $f_{PRS}/2^7$ ）

9.4 时钟定时器的运行

9.4.1 作为时钟定时器的运行

时钟定时器通过使用外围硬件时钟以固定的时间间隔产生中断请求信号（INTWT）。

如果给时钟定时器运行模式寄存器（WTM）的 bit0（WTM0）和 bit1（WTM1）置“1”，时钟定时器就开始计数运行，如果置“0”，就清除 5 位计数器，并且停止计数运行。

另外，在同时运行间隔定时器时，可通过给 WTM1 置“0”，以零秒启动时钟定时器。但是，此情况下不清除 11 位预分频器，因此，零秒启动时钟定时器后的第 1 个上溢（INTWT）产生最大 $2^9 \times 1/f_W$ 秒的误差。

中断请求信号的时间间隔如下所示。

表 9-4 时钟定时器的中断时间

WTM3	WTM2	中断时间的选择	以 $f_{PRS}=2\text{MHz}$ 运行时	以 $f_{PRS}=5\text{MHz}$ 运行时	以 $f_{PRS}=10\text{MHz}$ 运行时
0	0	$2^{14}/f_W$	1.05s	0.419s	0.210s
0	1	$2^{13}/f_W$	0.52s	0.210s	0.105s
1	0	$2^5/f_W$	2.05ms	819 μs	410 μs
1	1	$2^4/f_W$	1.02ms	410 μs	205 μs

备注 f_{PRS} : 外围硬件时钟频率

f_W : 时钟定时器的时钟频率 ($f_{PRS}/2^7$)

9.4.2 作为间隔定时器的运行

间隔定时器以事先设定的计数值为间隔，重复产生中断请求信号（INTWTI）。

可通过时钟定时器运行模式寄存器（WTM）的 bit4 ~ bit6（WTM4 ~ WTM6）选择间隔时间。如果将 WTM 的 bit0（WTM0）置“1”，间隔定时器就开始计数运行，如果置“0”，就停止计数运行。

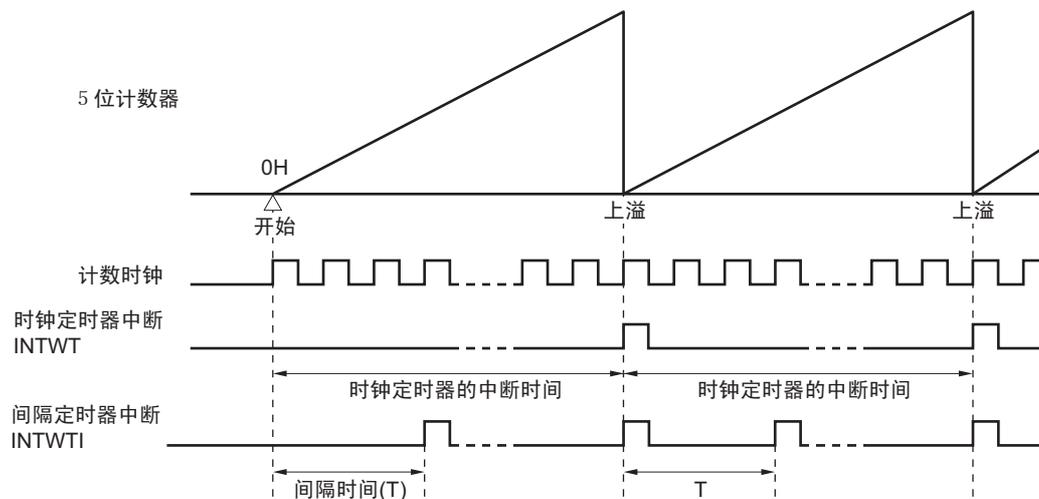
表 9-5 间隔定时器的间隔时间

WTM6	WTM5	WTM4	间隔时间	以 $f_{PRS}=2\text{MHz}$ 运行时	以 $f_{PRS}=5\text{MHz}$ 运行时	以 $f_{PRS}=10\text{MHz}$ 运行时
0	0	0	$2^4/f_W$	1.02ms	410 μs	205 μs
0	0	1	$2^5/f_W$	2.05ms	820 μs	410 μs
0	1	0	$2^6/f_W$	4.10ms	1.64ms	820 μs
0	1	1	$2^7/f_W$	8.20ms	3.28ms	1.64ms
1	0	0	$2^8/f_W$	16.4ms	6.55ms	3.28ms
1	0	1	$2^9/f_W$	32.8ms	13.1ms	6.55ms
1	1	0	$2^{10}/f_W$	65.5ms	26.2ms	13.1ms
1	1	1	$2^{11}/f_W$	131.1ms	52.4ms	26.2ms

备注 f_{PRS} : 外围硬件时钟频率

f_W : 时钟定时器的时钟频率 ($f_{PRS}/2^7$)

图 9-3 时钟定时器 / 间隔定时器的运行时序



9.5 时钟定时器的注意事项

在通过时钟定时器模式控制寄存器 (WTM) 允许 (将 WTM 的 bit0 (WTM0) 和 bit1 (WTM1) 置“1”) 时钟定时器和 5 位计数器运行时, 设置后到产生第 1 个中断请求信号 (INTWT) 的时间不为 WTM 的 bit2 和 bit3 (WTM2、WTM3) 的设置时间, 从第 2 次开始按照设置时间产生 INTWT 信号。

第 10 章 看门狗定时器

10.1 看门狗定时器的功能

看门狗定时器通过低速内部振荡时钟运行。

看门狗定时器用于检测程序的失控。如果检测出程序失控，就产生内部复位信号。

以下情况判断为程序失控。

- 看门狗定时器计数器发生上溢时
- 看门狗定时器允许寄存器（WDTE）使用 1 位操作指令时
- 给 WDTE 写“ACH”以外的数据时
- 在窗口关闭期间给 WDTE 写入数据时
- 从未由 IMS 寄存器设置的区域取指令时
（检测出 CPU 失控时的无效检查）
- 通过 CPU 的读/写指令存取未由 IMS 寄存器设置的区域（FB00H～FFCFH、FFE0H～FFFFH 除外）时
（检测出 CPU 失控时的异常存取）

因看门狗定时器发生复位时，复位控制标志寄存器（RESF）的 bit4（WDTRF）被置“1”。有关 RESF 的详细内容，请参照“第 18 章 复位功能”。

10.2 看门狗定时器的结构

看门狗定时器由以下硬件构成。

表 10-1 看门狗定时器的结构

项目	结构
控制寄存器	看门狗定时器允许寄存器 (WDTE)

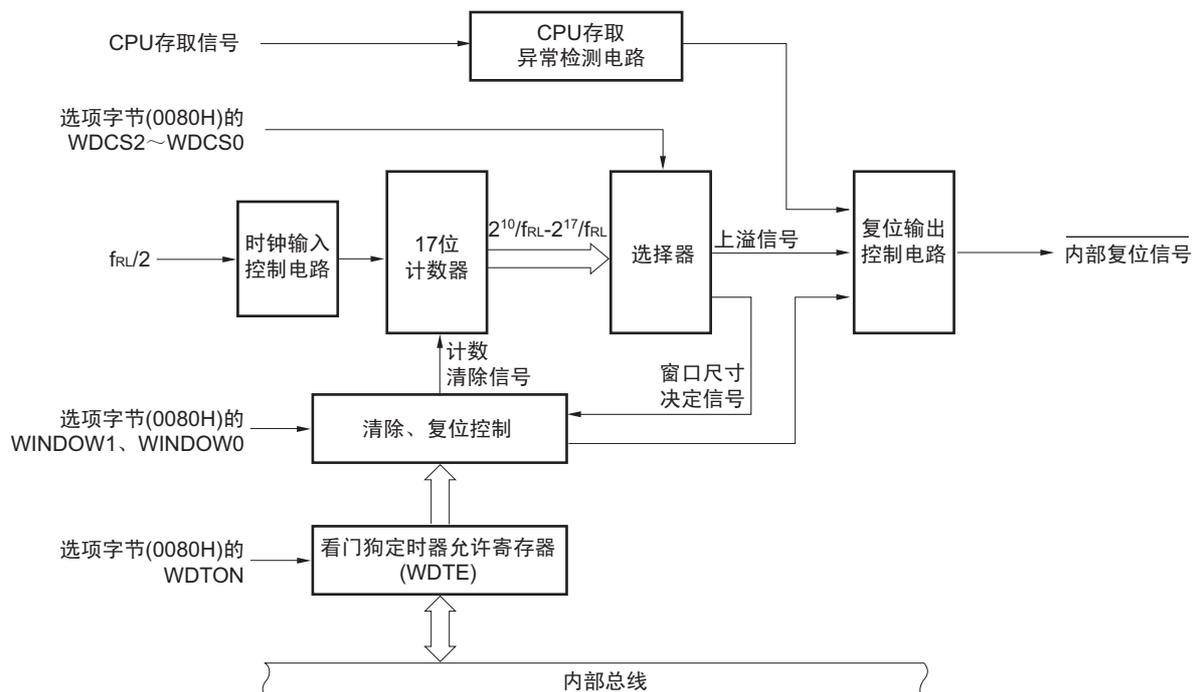
通过选项字节，进行计数器运行的控制，以及上溢时间和窗口打开期间的设置。

表 10-2 选项字节和看门狗定时器的设置内容

看门狗定时器的设置内容	选项字节 (0080H)
设置窗口打开期间	bit6、bit5 (WINDOW1、WINDOW0)
控制看门狗定时器的计数器运行	bit4 (WDTON)
设置看门狗定时器的上溢时间	bit3 ~ bit1 (WDCS2 ~ WDCS0)

备注 有关选项字节，详参照“第 21 章 选项字节”。

图 10-1 看门狗定时器的框图



10.3 看门狗定时器的控制寄存器

通过看门狗定时器允许寄存器（WDTE）控制看门狗定时器。

(1) 看门狗定时器允许寄存器（WDTE）

通过将“ACH”写入 WDTE，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置 WDTE。

在产生复位信号后，WDTE 变为“9AH”或者“1AH”注。

图 10-2 看门狗定时器允许寄存器（WDTE）的格式

地址：FF99H 复位后：9AH/1AH注 R/W

符号	7	6	5	4	3	2	1	0
WDTE								

注 WDTE 的复位值因选项字节（0080H）的 WDTON 的设定值而不同。如果要运行看门狗定时器，必须将 WDTON 置“1”。

WDTON 的设定值	WDTE 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. 如果将“ACH”以外的值写入 WDTE，就产生内部复位信号。但是，如果看门狗定时器的源时钟停止，就在看门狗定时器的源时钟重新开始运行时，产生内部复位信号。

2. 在对 WDTE 执行 1 位存储器操作指令时，产生内部复位信号。但是，如果看门狗定时器的源时钟停止，就在看门狗定时器的源时钟重新开始运行时，产生内部复位信号。

3. WDTE 的读取值为“9AH/1AH”（与写入值（“ACH”）不同的值）。

10.4 看门狗定时器的运行

10.4.1 看门狗定时器的运行控制

- 使用看门狗定时器时，通过选项字节（0080H）设置以下内容。
 - 将选项字节（0080H）的bit4（WDTON）置“1”，允许看门狗定时器的计数运行（复位解除后，计数器开始计数）（详细内容请参照第21章）。

WDTON	看门狗定时器的计数器 / 非法存取检测的运行控制
0	禁止计数运行（复位解除后停止计数）、禁止非法存取检测运行
1	允许计数运行（复位解除后开始计数）、允许非法存取检测运行

- 通过选项字节（0080H）的bit3～bit1（WDCS2～WDCS0），设置上溢时间（详细内容请参照10.4.3和第21章）。
 - 通过选项字节（0080H）的bit6和bit5（WINDOW1和WINDOW0），设置窗口打开期间（详细内容请参照10.4.3和第21章）。
- 复位解除后，看门狗计数器开始计数运行。
 - 在开始计数后到选项字节设置的上溢时间前，通过将“ACH”写入WDTE，清除看门狗定时器，并且重新开始计数运行。
 - 之后，必须在窗口打开期间进行复位解除后的第2次以及第2次以后的WDTE的写操作。如果在窗口关闭期间进行WDTE的写操作，就产生内部复位信号。
 - 如果在经过上溢时间后，还没有将“ACH”写入WDTE，就产生内部复位信号。

另外，在以下情况下，也产生内部复位信号。

- 对看门狗定时器允许寄存器（WDTE）使用1位操作指令时
- 将“ACH”以外的数据写入WDTE时
- 从未由IMS寄存器设置的区域取指令时（检测出CPU失控时的无效检查）
- 通过CPU的读/写指令存取未由IMS寄存器和IXS寄存器设置的区域（FB00H～FFCFH、FFE0H～FFFFH除外）时（检测出CPU失控时的异常存取）

- 注意 1. 只要是在上溢时间前，不管在任何时序执行复位解除后的第1次WDTE的写操作，都能清除看门狗定时器，并且重新开始计数运行。
- 在将“ACH”写入WDTE，清除看门狗定时器时，可能在实际的上溢时间与通过选项字节设置的上溢时间之间产生最长 $2/f_{RL}$ 秒的误差。
 - 在计数值发生上溢（FFFFH）前，看门狗定时器的清除都有效。
 - 根据选项字节的bit0（LSROSC）的设定值，看门狗定时器在HALT模式和STOP模式中的运行有以下不同。

	LSROSC=0 (可通过软件停止低速内部振荡器)	LSROSC=1 (不可停止低速内部振荡器)
HALT 模式下	停止看门狗定时器运行	持续看门狗定时器运行
STOP 模式下		

如果LSROSC=0，就在解除HALT模式和STOP模式后，重新开始看门狗定时器的计数。此时，计数器不清“0”，从停止前的值开始计数。

另外，在设置为LSROSC=0时，如果将LSRSTOP（内部振荡模式寄存器（RCM）的bit1）置“1”，停止低速内部振荡器的振荡，看门狗定时器的运行也停止。此时，计数器不清“0”。

10.4.2 看门狗定时器的上溢时间设置

通过选项字节（0080H）的 bit3 ~ bit1（WDCS2 ~ WDCS0）设置看门狗定时器的上溢时间。

如果发生上溢，就产生内部复位信号。如果在上溢时间前的窗口打开期间将“ACH”写入 WDTE，就清除计数，并且重新开始计数。

上溢时间的设置如下所示。

表 10-3 看门狗定时器的上溢时间设置

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间
0	0	0	$2^{10}/f_{RL}$ (3.88ms)
0	0	1	$2^{11}/f_{RL}$ (7.76ms)
0	1	0	$2^{12}/f_{RL}$ (15.52ms)
0	1	1	$2^{13}/f_{RL}$ (31.03ms)
1	0	0	$2^{14}/f_{RL}$ (62.06ms)
1	0	1	$2^{15}/f_{RL}$ (124.12ms)
1	1	0	$2^{16}/f_{RL}$ (248.24ms)
1	1	1	$2^{17}/f_{RL}$ (496.48ms)

注意 禁止 WDCS2=WDCS1=WDCS0=0 和 WINDOW1=WINDOW0=0 的组合设置。

备注 1. f_{RL} : 低速内部振荡时钟频率

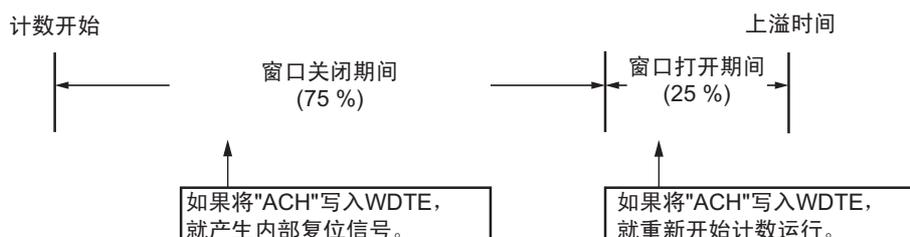
2. ()内: $f_{RL}=264\text{kHz (MAX.)}$

10.4.3 看门狗定时器的窗口打开期间设置

通过选项字节（0080H）的 bit5 和 bit6（WINDOW1 和 WINDOW0）设置看门狗定时器的窗口打开期间。窗口概述如下所示。

- 如果在窗口打开期间将“ACH”写入 WDTE，就清除看门狗定时器，并且重新开始计数。
- 即使在窗口关闭期间将“ACH”写入 WDTE，也会检测出异常，并且产生内部复位信号。

窗口打开期间为 25% 的情况



注意 只要是在上溢时间前，不管在任何时序执行复位解除后的第 1 次 WDTE 的写操作，都能清除看门狗定时器，并且重新开始计数运行。

窗口打开期间的设置如下所示。

表 10-4 看门狗定时器的窗口打开期间设置

WINDOW1	WINDOW0	看门狗定时器的窗口打开期间
0	0	25%
0	1	50%
1	0	75%
1	1	100%

注意 禁止 WDCS2=WDCS1=WDCS0=0 和 WINDOW1=WINDOW0=0 的组合设置。

备注 如果将上溢时间设置为 $2^{11}/f_{RL}$ ，则窗口关闭时间和打开时间如下所示。

	窗口打开期间的设置			
	25%	50%	75%	100%
窗口关闭时间	0 ~ 7.11ms	0 ~ 4.74ms	0 ~ 2.37ms	无
窗口打开时间	7.11 ~ 7.76ms	4.74 ~ 7.76ms	2.37 ~ 7.76ms	0 ~ 7.76ms

<窗口打开期间为 25% 时>

- 上溢时间：
 $2^{11}/f_{RL}(\text{MAX.})=2^{11}/264\text{kHz}(\text{MAX.})=7.76\text{ms}$
- 窗口关闭时间：
 $0 \sim 2^{11}/f_{RL}(\text{MIN.}) \times (1 - 0.25) = 0 \sim 2^{11}/216\text{kHz}(\text{MIN.}) \times 0.75 = 0 \sim 7.11\text{ms}$
- 窗口打开时间：
 $2^{11}/f_{RL}(\text{MIN.}) \times (1 - 0.25) \sim 2^{11}/f_{RL}(\text{MAX.}) = 2^{11}/216\text{kHz}(\text{MIN.}) \times 0.75 \sim 2^{11}/264\text{kHz}(\text{MAX.}) = 7.11 \sim 7.76\text{ms}$

第 11 章 A/D 转换器

11.1 A/D 转换器的功能

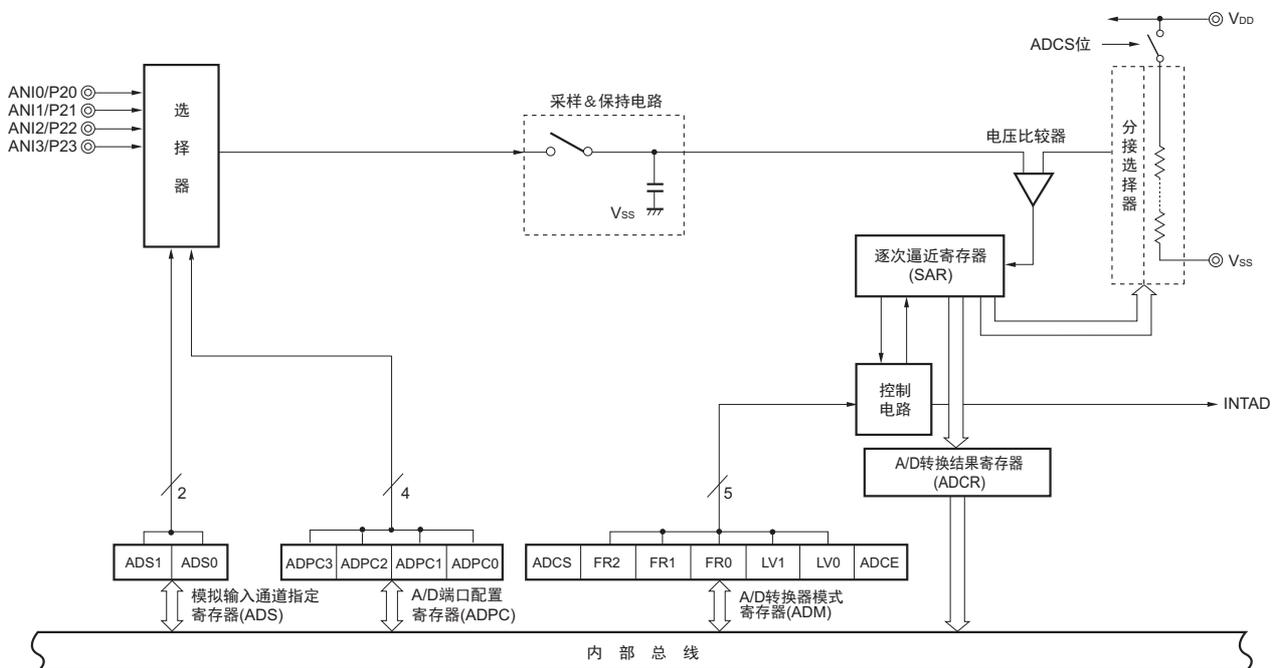
A/D 转换器是将模拟输入转换为数字值的 10 位分辨率转换器。最多可控制 4 个通道（ANI0 ~ ANI3）的模拟输入。

A/D 转换器具有以下功能。

- 10 位分辨率 A/D 转换

从 ANI0 ~ ANI3 中选择 1 个通道的模拟输入，并且重复执行 10 位分辨率的 A/D 转换运行。在每次 A/D 转换结束后，产生中断请求（INTAD）。

图 11-1 A/D 转换器的框图



11.2 A/D 转换器的结构

A/D 转换器由以下硬件构成。

(1) ANI0 ~ ANI3 引脚

这些是 A/D 转换器的 4 个通道的模拟输入引脚，用于输入进行 A/D 转换的模拟信号。除选作模拟输入的引脚之外，其他引脚均用作输入 / 输出端口。

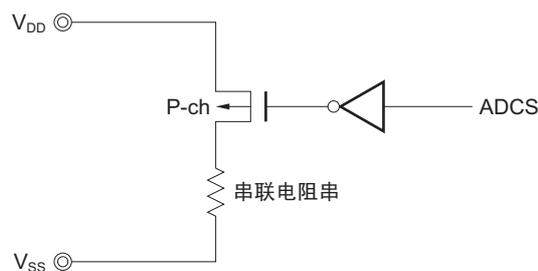
(2) 采样 & 保持电路

采样 & 保持电路在 A/D 转换开始时对选择器选择的模拟输入引脚的输入电压进行采样，并且在 A/D 转换期间保持所采样的电压值。

(3) 串联电阻串

串联电阻串连接在 V_{DD} 和 V_{SS} 之间，用于产生与采样电压值进行比较的电压。

图 11-2 串联电阻串的电路结构



(4) 电压比较器

电压比较器比较采样电压值和串联电阻串的输出电压。

(5) 逐次逼近寄存器 (SAR)

逐次逼近寄存器从最高有效位 (MSB) 开始对通过电压比较器比较的结果进行转换。

将比较结果转换成数字值直到最低有效位 (LSB) 为止 (A/D 转换结束)，SAR 寄存器的内容被传送到 A/D 转换结果寄存器 (ADCR)。

(6) 10 位 A/D 转换结果寄存器 (ADCR)

每当 A/D 转换结束，就从逐次逼近寄存器加载转换结果，并且将 A/D 转换结果保存到高 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，就从逐次逼近寄存器加载转换结果，并且保存 A/D 转换结果的高 8 位。

注意 从 ADCR、ADCRH 读取数据时，产生等待。在外围硬件时钟 (f_{PRS}) 停止时，不能从 ADCR、ADCRH 读取数据。详细内容请参照“第 27 章 等待的注意事项”。

(8) 控制电路

控制电路控制 A/D 转换的模拟输入转换时间、转换运行的开始 / 停止等。在 A/D 转换结束时，产生 INTAD。

(9) V_{DD} 引脚

V_{DD} 引脚是 A/D 转换器的模拟电源引脚 / 基准电压的输入引脚。

根据 V_{DD} 与 V_{SS} 之间所需的电压，将 ANI0 ~ ANI3 的输入信号转换成数字信号。

(10) V_{SS} 引脚

V_{SS} 引脚是 A/D 转换器的接地电位引脚。

(11) A/D 转换器模式寄存器 (ADM)

A/D 转换器模式寄存器设置进行 A/D 转换的模拟输入的转换时间以及转换运行的开始 / 停止。

(12) A/D 端口配置寄存器 (ADPC)

A/D 端口配置寄存器将 ANI0/P20 ~ ANI3/P23 引脚切换为 A/D 转换器的模拟输入 / 端口的数字输入 / 输出。

(13) 模拟输入通道指定寄存器 (ADS)

模拟输入通道指定寄存器指定进行 A/D 转换的模拟电压的输入端口。

(14) 端口模式寄存器 2 (PM2)

端口模式寄存器 2 对 ANI0/P20 ~ ANI3/P23 引脚进行输入 / 输出切换。

11.3 A/D 转换器使用的寄存器

A/D 转换器使用以下 6 种寄存器。

- A/D 转换器模式寄存器 (ADM)
- A/D 端口配置寄存器 (ADPC)
- 模拟输入通道指定寄存器 (ADS)
- 端口模式寄存器 2 (PM2)
- 10 位 A/D 转换结果寄存器 (ADCR)
- 8 位 A/D 转换结果寄存器 (ADCRH)

(1) A/D 转换器模式寄存器 (ADM)

A/D 转换器模式寄存器设置进行 A/D 转换的模拟输入的转换时间以及转换运行的开始 / 停止。

通过 1 位或 8 位存储器操作指令设置 ADM。

在产生复位信号后，ADM 变为“00H”。

图 11-3 A/D 转换器模式寄存器 (ADM) 的格式

地址：FF28H 复位后：00H R/W

符号	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	FR2 注 1	FR1 注 1	FR0 注 1	LV1 注 1	LV0 注 1	ADCE

ADCS	控制 A/D 转换运行
0	停止转换运行
1	允许转换运行

ADCE	控制电压比较器的运行注 2
0	停止电压比较器的运行
1	允许电压比较器的运行

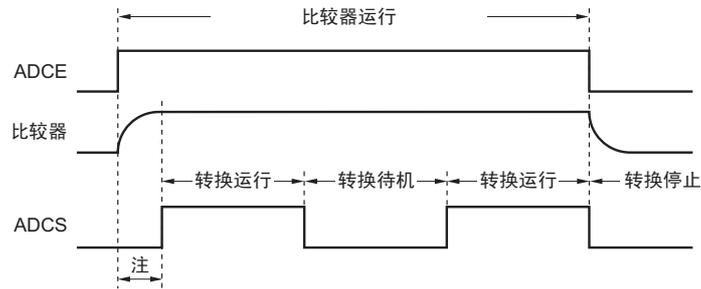
- 注 1. 有关 FR2 ~ FR0、LV1、LV0 和 A/D 转换的详细内容，请参照“表 11-2 A/D 转换时间的选择”。
2. 通过 ADCS 和 ADCE 控制电压比较器的运行，从开始运行到运行稳定需要 1 μ s。因此，从将 ADCE 置“1”开始到至少经过 1 μ s 后，如果将 ADCS 置“1”，就从第 1 个转换数据开始有效。在未等到 1 μ s 就将 ADCS 置“1”时，忽略第 1 个转换数据。

表 11-1 ADCS 和 ADCE 的设置

ADCS	ADCE	A/D 转换运行
0	0	停止状态 (不存在直流功耗途径)
0	1	转换待机模式 (电压比较器运行、只有电压比较器产生功耗)
1	0	转换模式 (电压比较器的运行停止注)
1	1	转换模式 (电压比较器运行)

注 忽略第 1 个转换数据。

图 11-4 使用比较器时的时序图



注 为了稳定内部电路，从 ADCE 的上升沿到 ADCS 的上升沿的所需时间至少为 1 μ s。

注意 1. 将 FR0 ~ FR2、LV1 和 LV0 改写成不同数据时，必须先停止 A/D 转换运行。

2. 如果给 ADM 写数据，就产生等待。在外围硬件时钟 (f_{PRS}) 停止时，不能将数据写到 ADM。详细内容请参照“第 27 章 等待的注意事项”。

表 11-2 A/D 转换时间的选择

A/D 转换器模式寄存器 (ADM)					转换时间的选择			转换时钟 (f_{AD})
FR2	FR1	FR0	LV1	LV0	$f_{PRS}=2\text{MHz}$	$f_{PRS}=10\text{MHz}$		
0	0	0	0	0	$264/f_{PRS}$	禁止设置	26.4 μ s	$f_{PRS}/12$
0	0	1	0	0	$176/f_{PRS}$		17.6 μ s	$f_{PRS}/8$
0	1	0	0	0	$132/f_{PRS}$	66.0 μ s	13.2 μ s	$f_{PRS}/6$
0	1	1	0	0	$88/f_{PRS}$	44.0 μ s	8.8 μ s 注	$f_{PRS}/4$
1	0	0	0	0	$66/f_{PRS}$	33.0 μ s	6.6 μ s 注	$f_{PRS}/3$
1	0	1	0	0	$44/f_{PRS}$	22.0 μ s	禁止设置	$f_{PRS}/2$
上述以外					禁止设置			

注意 1. 在以下条件下，设置转换时间。

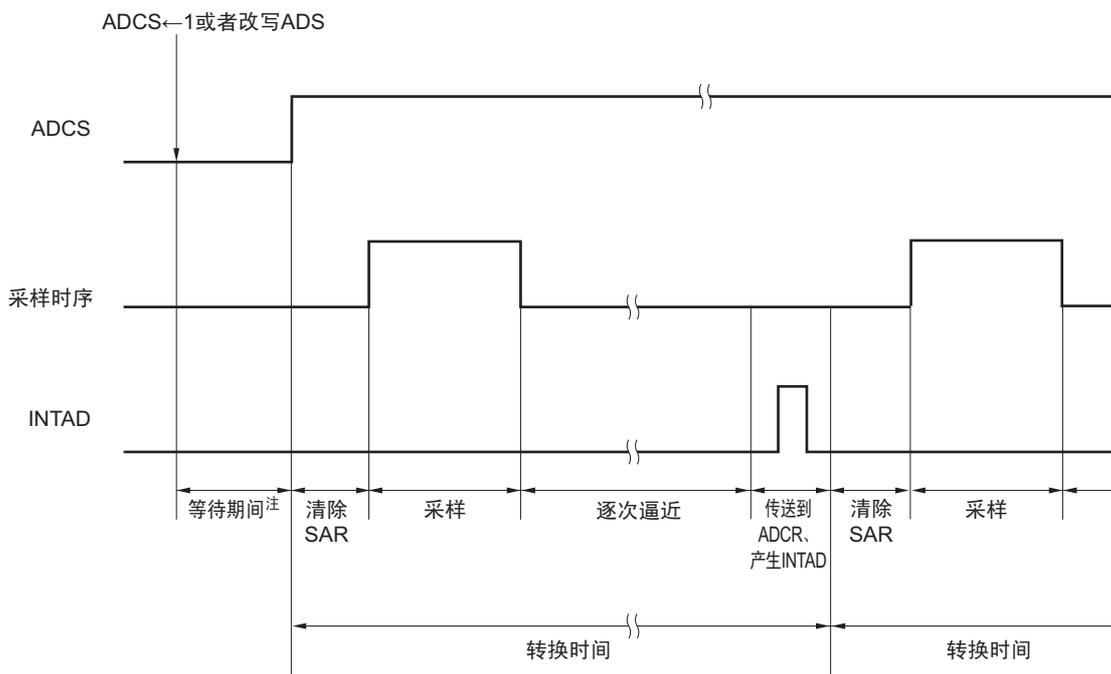
• 4.0V \leq V_{DD} \leq 5.5V 时: $f_{AD}=0.33 \sim 3.6\text{MHz}$

2. 将 FR0 ~ FR2、LV1 和 LV0 改写成不同数据时，必须先停止 A/D 转换运行 (ADCS=0)。

3. 上述转换时间不包括时钟频率误差。选择转换时间时，必须考虑时钟频率误差。

备注 f_{PRS} : 外围硬件时钟频率

图 11-5 A/D 转换器的采样和 A/D 转换时序



注 有关等待期间的详细内容，请参照“第 27 章 等待的注意事项”。

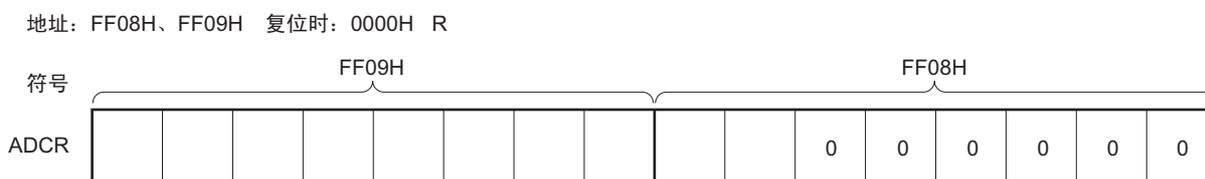
(2) 10 位 A/D 转换结果寄存器 (ADCR)

10 位 A/D 转换结果寄存器是保存 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，就从逐次转换寄存器加载转换结果，并且将转换结果的高 8 位保存到“FF09H”，低 2 位保存到“FF08H”的高 2 位。

通过 16 位存储器操作指令读取 ADCR。

在产生复位信号后，ADCR 变为“0000H”。

图 11-6 10 位 A/D 转换结果寄存器 (ADCR) 的格式



注意 1. 对 A/D 转换器模式寄存器 (ADM)、模拟输入通道指定寄存器 (ADS) 和 A/D 端口配置寄存器 (ADPC) 执行写操作时，ADCR 的内容有可能为不定值。必须在转换结束后，ADM、ADS、ADPC 的写操作执行前，读取转换结果。如果在上述时序以外的时序进行读取，就有可能无法读取正确的转换结果。

2. 如果从 ADCR 读取数据，就产生等待。在外围硬件时钟 (f_{PRS}) 停止时，不能从 ADCR 读取数据。详细内容请参照“第 27 章 等待的注意事项”。

(3) 8 位 A/D 转换结果寄存器 (ADCRH)

8 位 A/D 转换结果寄存器是保存 10 位分辨率的高 8 位 A/D 转换结果的 8 位寄存器。

通过 8 位存储器操作指令读取 ADCRH。

在产生复位信号后，ADCRH 变为“00H”。

图 11-7 8 位 A/D 转换结果寄存器 (ADCRH) 的格式

地址: FF09H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ADCRH								

注意 1. 对 A/D 转换器模式寄存器 (ADM)、模拟输入通道指定寄存器 (ADS) 和 A/D 端口配置寄存器 (ADPC) 执行写操作时，ADCRH 的内容有可能为不定值。必须在转换结束后，ADM、ADS、ADPC 的写操作执行前，读取转换结果。如果在上述时序以外的时序进行读取，就有可能无法读取正确的转换结果。

2. 如果从 ADCRH 读取数据，就产生等待。在外围硬件时钟 (f_{PRS}) 停止时，不能从 ADCRH 读取数据。详细内容请参照“第 27 章 等待的注意事项”。

(4) 模拟输入通道指定寄存器 (ADS)

模拟输入通道指定寄存器指定进行 A/D 转换的模拟电压的输入通道。

通过 1 位或 8 位存储器操作指令设置 ADS。

在产生复位信号后，ADS 变为“00H”。

图 11-8 模拟输入通道指定寄存器 (ADS) 的格式

地址: FF29H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADS	0	0	0	0	0	0	ADS1	ADS0

ADS1	ADS0	指定模拟输入通道
0	0	ANI0
0	1	ANI1
1	0	ANI2
1	1	ANI3

注意 1. 必须将 bit2 ~ bit7 置“0”。

2. 通过端口模式寄存器 2 (PM2)，将 A/D 转换时使用的通道设置为输入模式。

3. 如果给 ADS 写数据，就产生等待。在外围硬件时钟 (f_{PRS}) 停止时，不能将数据写到 ADS。详细内容请参照“第 27 章 等待的注意事项”。

(5) A/D 端口配置寄存器 (ADPC)

A/D 端口配置寄存器将 ANI0/P20 ~ ANI3/P23 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 1 位或 8 位存储器操作指令设置 ADPC。

在产生复位信号后，ADPC 变为“00H”。

图 11-9 A/D 端口配置寄存器 (ADPC) 的格式

地址: FF2FH 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	数字输入 / 输出 (D) 或者模拟输入 (A) 的切换			
				P23/ANI3	P22/ANI2	P21/ANI1	P20/ANI0
0	0	0	0	A	A	A	A
0	0	0	1	A	A	A	D
0	0	1	0	A	A	D	D
0	0	1	1	A	D	D	D
0	1	0	0	D	D	D	D
1	0	0	0	D	D	D	D
上述以外				禁止设置			

注意 1. 通过端口模式寄存器 2 (PM2)，将设置为模拟输入 / 输出的引脚设置为输入模式。

2. 如果给 ADPC 写数据，就产生等待。在外围硬件时钟 (f_{PRS}) 停止时，不能将数据写到 ADPC。详细内容请参照“第 27 章 等待的注意事项”。

(6) 端口模式寄存器 2 (PM2)

将 ANI0/P20 ~ ANI3/P23 引脚用作模拟输入端口时，必须将 PM20 ~ PM23 分别置“1”。此时，P20 ~ P23 的输出锁存器既可是“0”也可是“1”。

如果将 PM20 ~ PM23 分别置“0”，就不能用作模拟输入端口。

通过 1 位或 8 位存储器操作指令设置 PM2。

在产生复位信号后，PM2 变为“FFH”。

图 11-10 端口模式寄存器 2 (PM2) 的格式

地址: FF22H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM2	1	1	1	1	PM23	PM22	PM21	PM20

PM2n	选择 P2n 引脚的输入 / 输出模式 (n=0 ~ 3)
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

注意 必须将 bit4 ~ bit7 置“1”。

ANI0/P20 ~ ANI3/P23 引脚的功能取决于 ADPC、ADS 以及 PM2 的设置。

表 11-3 ANI0/P20 ~ ANI3/P23 引脚功能的设置

ADPC	PM2	ADS	ANI0/P20 ~ ANI3/P23 引脚
选择模拟输入	输入模式	选择 ANI	模拟输入 (A/D 转换对象)
		不选择 ANI	模拟输入 (非 A/D 转换对象)
	输出模式	选择 ANI	禁止设置
		不选择 ANI	
选择数字输入 / 输出	输入模式	—	数字输入
	输出模式	—	数字输出

11.4 A/D 转换器的运行

11.4.1 A/D 转换器的基本运行

1. 将 A/D 转换器模式寄存器 (ADM) 的 bit0 (ADCE) 置“1”，开始比较器的运行。
2. 通过 A/D 端口配置寄存器 (ADPC) 将 A/D 转换通道设置为模拟输入，并且通过端口模式寄存器 2 (PM2) 设置为输入模式。
3. 通过 ADM 的 bit5 ~ bit1 (FR2 ~ FR0、LV1、LV0) 设置 A/D 转换时间。
4. 通过模拟输入通道指定寄存器 (ADS) 选择 1 个 A/D 转换通道。
5. 将 ADM 的 bit7 (ADCS) 置“1”，开始转换运行。
(步骤 6 ~ 步骤 12 为在硬件中的运行)
6. 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。
7. 经过一定时间的采样后，采样 & 保持电路进入保持状态，保持被采样的电压直到 A/D 转换结束。
8. 设置逐次逼近寄存器 (SAR) 的 bit9 后，分接选择器将串联电阻串的分接头电压置为 $(1/2)V_{DD}$ 。
9. 通过电压比较器比较串联电阻串的分接头电压和采样电压之间的电压差。如果模拟输入电压大于 $(1/2)V_{DD}$ ，SAR 的 MSB 保持设置；如果模拟输入电压小于 $(1/2)V_{DD}$ ，SAR 的 MSB 发生复位。
10. 接着自动设置 SAR 的 bit8，进入下一个比较。此时，根据已设置结果的 bit9 的值，如下选择串联电阻串的分接头电压。
 - bit9=1: $(3/4)V_{DD}$
 - bit9=0: $(1/4)V_{DD}$
 比较该分接头电压和采样电压，根据其结果如下设置 SAR 的 bit8。
 - 采样电压 \geq 分接头电压: bit8=1
 - 采样电压 $<$ 分接头电压: bit8=0
11. 持续进行上述比较，直到比较到 SAR 的 bit0。
12. 10 位的比较全部结束后，在 SAR 中保留有效数字的结果，并且将该结果值传送到 A/D 转换结果寄存器 (ADCR、ADCRH)，进行锁存。
同时，还可产生 A/D 转换结束中断请求 (INTAD)。
13. 重复步骤 6 ~ 步骤 12 的运行，直到 ADCS 变为“0”。

如果要停止 A/D 转换器，必须将 ADCS 置“0”。

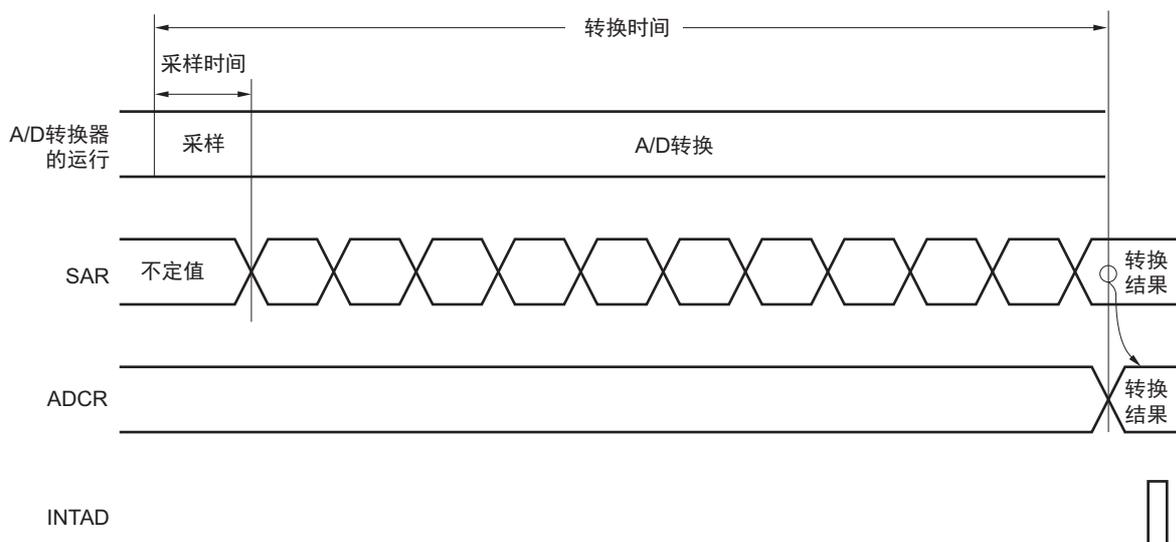
在 ADCE 为“1”的状态下重新开始 A/D 转换时，必须从步骤 5 开始执行。在 ADCE 为“0”的状态下重新开始 A/D 转换时，必须将 ADCE 置“1”，并且在等待至少 $1\mu\text{s}$ 后，从步骤 5 开始执行。另外，如果要更改 A/D 转换的通道，必须从步骤 4 开始执行。

注意 在步骤 1 到步骤 5 之间至少空出 $1\mu\text{s}$ 。

备注 A/D 转换结果寄存器有 2 种。

- ADCR (16 位): 保存 10 位 A/D 转换值
- ADCRH (8 位): 保存 8 位 A/D 转换值

图 11-11 A/D 转换器的基本操作



持续执行 A/D 转换运行，直到通过软件将 A/D 转换器模式寄存器（ADM）的 bit7（ADCS）复位为“0”。在 A/D 转换运行期间，如果对模拟输入通道指定寄存器（ADS）进行写操作，转换运行就被初始化，如果 ADCS 位被置“1”，就从头开始执行转换。

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

11.4.2 输入电压和转换结果

输入到模拟输入引脚（AN10 ~ AN13）的模拟输入电压与理论上的 A/D 转换结果（10 位 A/D 转换结果寄存器（ADCR））之间的关系如下所示。

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

$$ADCR = SAR \times 64$$

或者

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

INT(): 该函数返回 () 中的值的整数部分

V_{AIN} : 模拟输入电压

V_{DD} : V_{DD} 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 11-12 所示。

图 11-12 模拟输入电压和 A/D 转换结果之间的关系

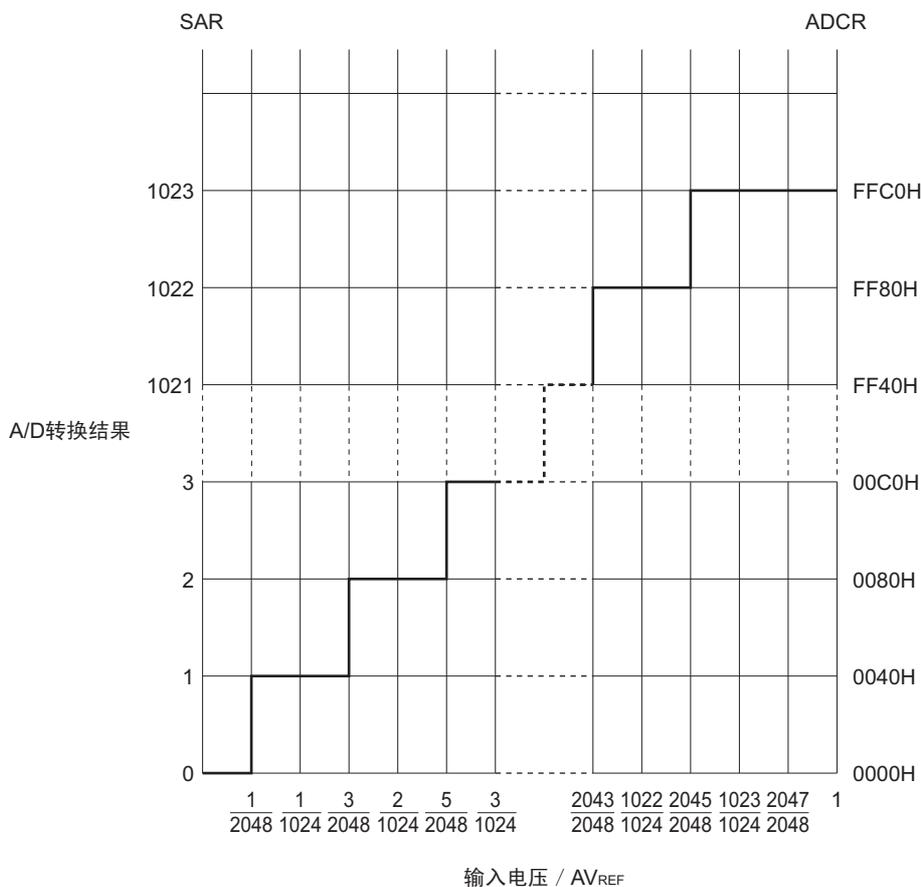
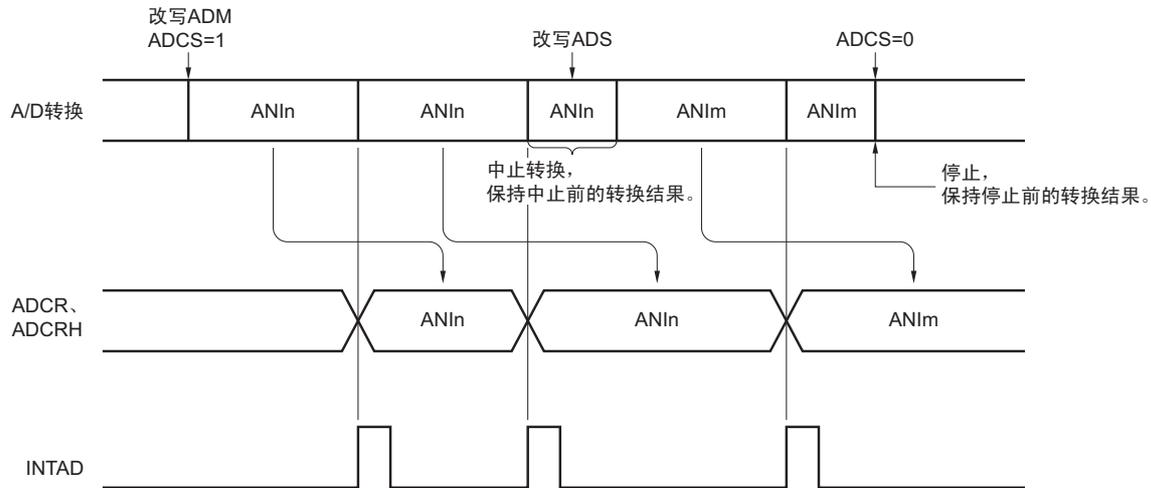


图 11-13 A/D 转换运行



备注 1. $n=0 \sim 3$

2. $m=0 \sim 3$

设置方法如下所述。

1. 将 A/D 转换器模式寄存器 (ADM) 的 bit0 (ADCE) 置“1”。
2. 通过 A/D 端口配置寄存器 (ADPC) 的 bit3 ~ bit0 (ADPC3 ~ ADPC0) 和端口模式寄存器 2 (PM2) 的 bit3 ~ bit0 (PM23 ~ PM20) 将使用的通道设置为模拟输入。
3. 通过 ADM 的 bit5 ~ bit1 (FR2 ~ FR0、LV1、LV0) 选择转换时间。
4. 通过模拟输入通道指定寄存器 (ADS) 的 bit1、bit0 (ADS1、ADS0) 选择使用的通道。
5. 将 ADM 的 bit7 (ADCS) 置“1”，开始 A/D 转换。
6. 一旦 1 次 A/D 转换结束，就产生中断请求信号 (INTAD)。
7. 将 A/D 转换数据传送到 A/D 转换结果寄存器 (ADCR、ADCRH)。

<更改通道>

8. 通过 ADS 的 bit1、bit0 (ADS1、ADS0) 更改通道，开始 A/D 转换。
9. 一旦 1 次 A/D 转换结束，就产生中断请求信号 (INTAD)。
10. 将 A/D 转换数据传送到 A/D 转换结果寄存器 (ADCR、ADCRH)。

<结束 A/D 转换>

11. 将 ADCS 清“0”。
12. 将 ADCE 清“0”。

注意 1. 步骤 1 到步骤 5 之间至少空出 1 μ s。

2. 也可在步骤 2 到步骤 4 之间执行步骤 1。
3. 可省略步骤 1。但在这种情况下执行步骤 5 后，忽略第 1 个转换数据。
4. 步骤 6 到步骤 9 的时间与 ADM 的 bit5 ~ bit1 (FR2 ~ FR0、LV1、LV0) 设置的转换时间不同。步骤 8 到步骤 9 的时间为 FR2 ~ FR0、LV1、LV0 设置的转换时间。

11.5 A/D 转换器特性表的阅读方法

以下对 A/D 转换器中的专用术语进行说明。

(1) 分辨率

分辨率是可识别的最小模拟输入电压。即，将 1 位数字输出的模拟输入电压的比率称之为 1LSB（Least Significant Bit（最低有效位））。用“%FSR”（Full Scale Range（满刻度范围）表示 1LSB 的满刻度比率。

10 位分辨率时：

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

精度与分辨率无关，取决于总误差。

(2) 总误差

总误差是指实际测量值与理论值之间的最大误差。

零刻度误差、满刻度误差、积分线性误差、微分线性误差以及由这些误差的组合产生的误差的综合称为总误差。

特性表的总误差中不包含量化误差。

(3) 量化误差

量化误差是在将模拟值转换成数字值时肯定产生的 $\pm 1/2\text{LSB}$ 误差。在 A/D 转换器中，因为是将 $\pm 1/2\text{LSB}$ 范围内的模拟输入电压转换成相同数字代码，所以量化误差是不可避免的。

特性表的总误差、零刻度误差、满刻度误差、积分线性误差和微分线性误差中都不包含量化误差。

图 11-14 总误差

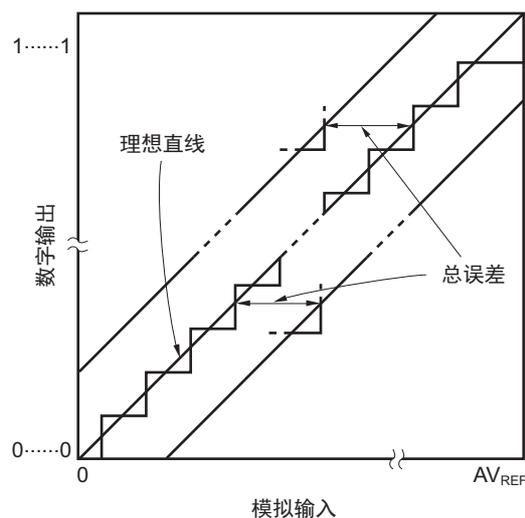
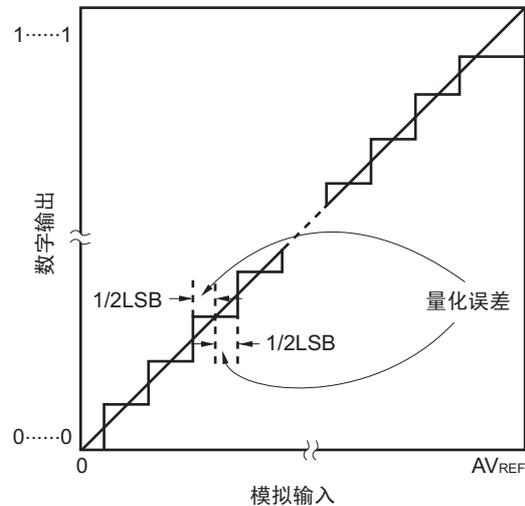


图 11-15 量化误差



(4) 零刻度误差

零刻度误差表示在数字输出从 0.....000 变为 0.....001 时的模拟输入电压的实际测量值与理论值 ($1/2\text{LSB}$) 之差。如果实际测量值大于理论值，零刻度误差表示数字输出从 0.....001 变为 0.....010 时的模拟输入电压的实际测量值与理论值 ($3/2\text{LSB}$) 之差。

(5) 满刻度误差

满刻度误差表示在数字输出从 1.....110 变为 1.....111 时的模拟输入电压的实际测量值与理论值 (满刻度 - $3/2\text{LSB}$) 之差。

(6) 积分线性误差

积分线性误差表示转换特性偏离理想线性关系的程度，以及表示在零刻度误差和满刻度误差均为“0”时的实际测量值与理论值之间的最大差值。

(7) 微分线性误差

微分线性误差表示在代码的输出理想宽度为 1LSB 时的代码输出宽度的实际测量值与理想值之差。

图 11-16 零刻度误差

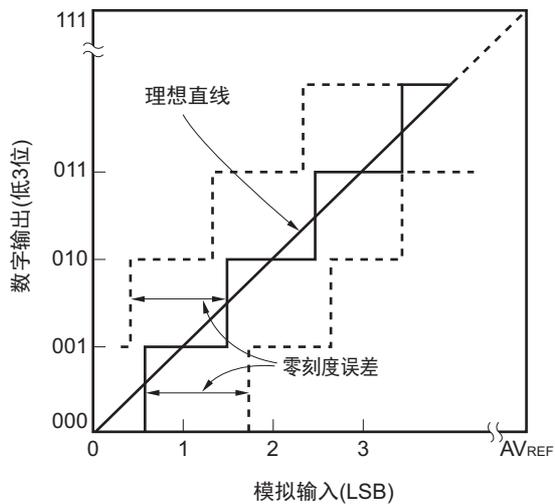


图 11-17 满刻度误差

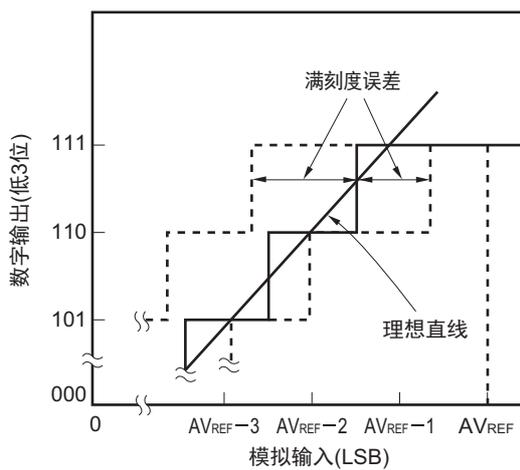


图 11-18 积分线性误差

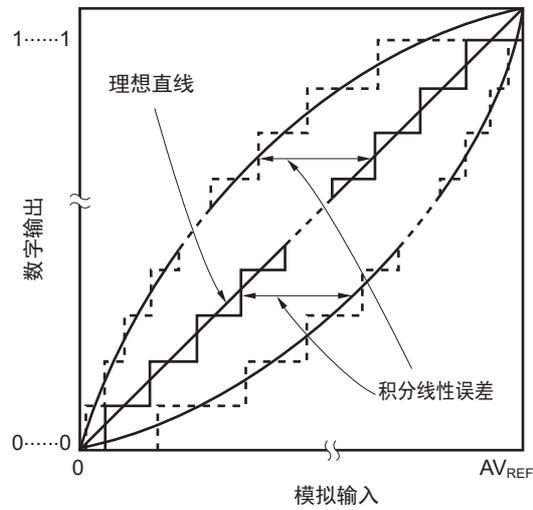
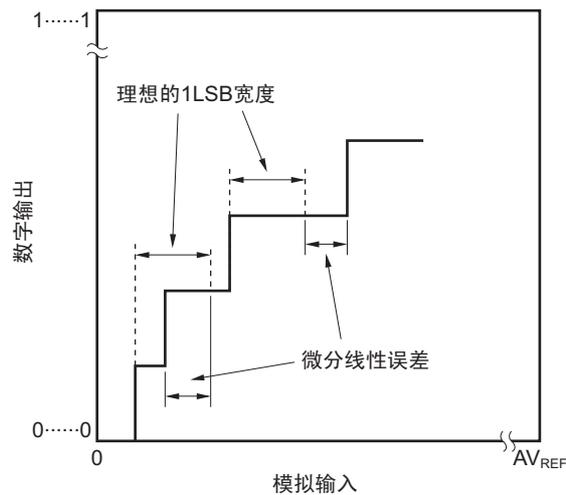


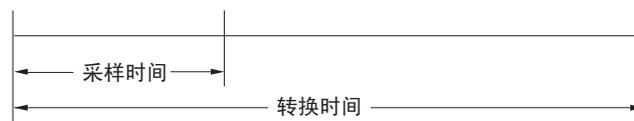
图 11-19 微分线性误差

**(8) 转换时间**

转换时间表示从开始采样到获取数字输出的时间。
特性表的转换时间包括采样时间。

(9) 采样时间

采样时间是指为了将模拟电压取到采样 & 保持电路而保持模拟开关打开的时间。



11.6 A/D 转换器的注意事项

(1) STOP 模式中的工作电流

在 STOP 模式中 A/D 转换器停止运行。此时，可通过将 A/D 转换器模式寄存器（ADM）的 bit7（ADCS）和 bit0（ADCE）置“0”，降低工作电流。

如果要从待机状态重新启动运行，就必须先将中断请求标志寄存器 1L（IF1L）的 bit0（ADIF）清“0”，再开始运行。

(2) ANI0 ~ ANI3 的输入范围

在规格范围内使用 ANI0 ~ ANI3 的输入电压。尤其是一旦输入大于 V_{DD} 或者小于 V_{SS} 的电压（即使在绝对最大额定值范围内），该通道的转换值就为不定值，而且还可能会影响到其他通道的转换值。

(3) 竞争运行

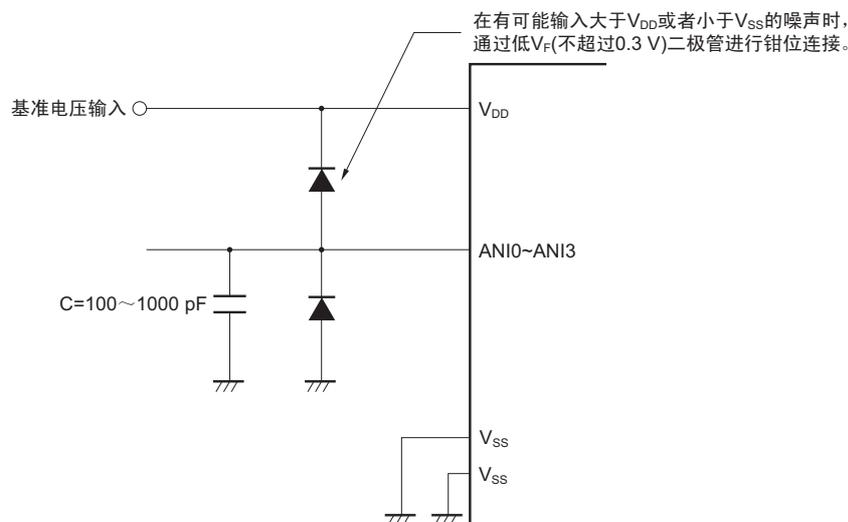
1. 当转换结束时的 A/D 转换结果寄存器（ADCR、ADCRH）的写操作与通过指令进行 ADCR、ADCRH 的读操作发生竞争时
优先读 ADCR、ADCRH，读取后，将新的转换结果写入 ADCR、ADCRH。
2. 当转换结束时的 ADCR、ADCRH 的写操作与 A/D 转换器模式寄存器（ADM）、模拟输入通道指定寄存器（ADS）或者 A/D 端口配置寄存器（ADPC）的写操作发生竞争时
优先写 ADM、ADS 或者 ADPC。不执行 ADCR、ADCRH 的写操作，也不产生转换结束中断信号（INTAD）。

(4) 噪声对策

为了保持 10 位分辨率，必须注意影响 V_{DD} 、ANI0 ~ ANI3 引脚的噪声。

1. 电源需连接具有低等效电阻且频率响应良好的电容器。
2. 模拟输入源的输出阻抗越高，影响就越大。为了减小噪声，推荐如图 11-20 所示的外接 C。
3. 转换期间不能切换其他引脚。
4. 如果在转换开始后立即设置为 HALT 模式，就可以提高精度。

图 11-20 模拟输入引脚的处理



(5) ANI0/P20 ~ ANI3/P23

1. 模拟输入引脚（ANI0 ~ ANI3）复用数字输入/输出端口（P20 ~ P23）引脚。选择 ANI0 ~ ANI3 中的任意一个进行 A/D 转换时，转换过程期间不能存取 P20 ~ P23，否则，可能会降低转换分辨率。另外，选择作为 P20 ~ P23 使用的引脚时，推荐从离 V_{DD} 最远的 ANI0/P20 引脚开始执行。
2. 如果在与 A/D 转换中的引脚的相邻引脚上外加数字脉冲，就可能因耦合噪声而得不到预期的 A/D 转换值。因此，不要在 A/D 转换中的引脚的相邻引脚上外加脉冲。

(6) ANI0 ~ ANI3 引脚的输入阻抗

在 A/D 转换器中，通过在采样时间内向内部采样电容器充电来进行采样。

因此，在不进行采样时只有漏电流流过，而在进行采样时还有向电容器充电的电流流过，从而输入阻抗在采样期间和非采样期间发生波动。

为了充分的进行采样，推荐模拟输入源的输出阻抗不超过 $10k\Omega$ ，并且在高输出阻抗时连接 $100pF$ 左右的电容器到 ANI0 ~ ANI3 引脚（参照图 11-20）。

(7) V_{DD} 引脚的输入抗租

在 V_{DD} 引脚和 V_{SS} 引脚之间连接数十 $k\Omega$ 的串联电阻串。

因此，在基准电压源的输出阻抗高时，造成输出阻抗与 V_{DD} 引脚和 V_{SS} 引脚之间的串联电阻串串联，从而导致基准电压的误差变大。

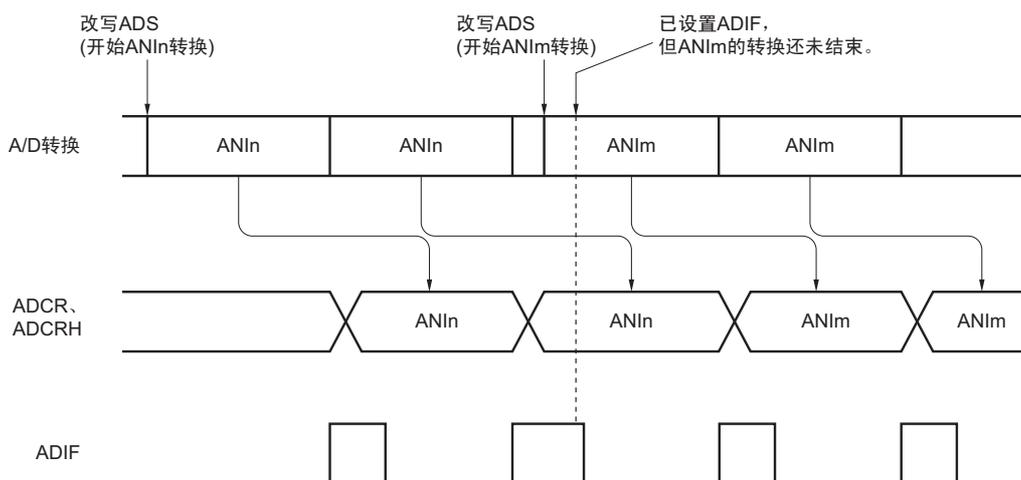
(8) 中断请求标志（ADIF）

即使更改模拟输入通道指定寄存器（ADS），中断请求标志（ADIF）也不被清“0”。

因此，在 A/D 转换期间更改模拟输入引脚后，改写 ADS 前，有可能传送更改前的模拟输入的 A/D 转换结果和置位 ADIF 标志。需要注意的是，如果在改写 ADS 后立即读取 ADIF，即使更改后的模拟输入的 A/D 转换没有结束，也置位 ADIF。

另外，如果在停止 A/D 转换后又要重新开始时，必须在重新开始前先将 ADIF 清“0”。

图 11-21 A/D 转换结束中断请求的发生时序



备注 1. $n=0 \sim 3$

2. $m=0 \sim 3$

(9) A/D 转换刚开始后的转换结果

如果将 ADCE 位置“1”，再在 $1\mu\text{s}$ 内将 ADCS 位置“1”，或者在 ADCE=0 的状态下将 ADCS 位置“1”，A/D 转换刚开始后的 A/D 转换值就有可能不在额定值范围内。可采取轮询 A/D 转换结束中断请求 (INTAD)，舍弃第 1 个转换结果等措施。

(10) A/D 转换结果寄存器 (ADCR、ADCRH) 的读取

在进行 A/D 转换器模式寄存器 (ADM)、模拟输入通道指定寄存器 (ADS) 和 A/D 端口配置寄存器 (ADPC) 的写操作时，ADCR、ADCRH 的内容有可能为不定值。必须在转换结束后，进行 ADM、ADS、ADPC 的写操作前，读取转换结果。如果在上述时序外读取，就有可能无法读取正确的转换结果。

(11) 内部等效电路

模拟输入部的等效电路如下所示。

图 11-22 ANIn 引脚的内部等效电路

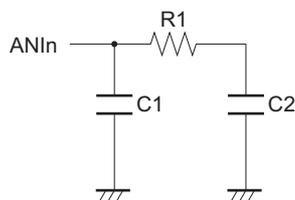


表 11-4 等效电路的各电阻和电容值 (参考值)

V_{DD}	R1	C1	C2
$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	8.1 k Ω	8 pF	5 pF

备注 1. 表 11-4 中的各电阻和电容值并非保证值。

2. $n=0 \sim 3$

第 12 章 串行接口 UART0

12.1 串行接口 UART0 的功能

串行接口 UART0 有以下两种模式。

(1) 运行停止模式

这是在不执行串行通信时使用的模式。能降低功耗。

详细内容请参照“12.4.1 运行停止模式”。

(2) 异步串行接口 (UART) 模式

异步串行接口模式的功能概要如下所示。

详细内容请参照“12.4.2 异步串行接口 (UART) 模式”和“12.4.3 专用波特率发生器”。

- 最大传送速度：625kbps
- 双引脚结构 TxD0：发送数据的输出引脚
RxD0：接收数据的输入引脚
- 可选择7位或者8位的通信数据长度
- 可通过内置专用5位波特率发生器，设置任意的波特率
- 可独立进行发送运行和接收运行（全双工运行）
- 通信数据的起始位固定为LSB

- 注意 1. 在提供给串行接口 UART0 的时钟未停止时（例如，HALT 模式），可继续正常运行。在提供给串行接口 UART0 的时钟停止时（例如，STOP 模式），各寄存器保持时钟停止前的值，并且停止运行。TxD0 引脚输出也相同，保持时钟停止前的值，并且输出。但是，不保证重新提供时钟后的运行，因此，在重新开始提供时钟后，必须将 POWER0、RXE0、TXE0 置“0”，进行电路复位。
2. 通过将 POWER0 置“1”，再将 TXE0 置“1”（发送）或者将 RXE0 置“1”（接收），开始通信。
 3. 通过 BRGC0 设置的基本时钟（ f_{XCLK0} ）同步 TXE0 与 RXE0。在允许重新开始传送或者接收时，必须在将 TXE0 或者 RXE0 置“0”后至少经过 2 个基本时钟，再将 TXE0 或者 RXE0 置“1”。如果在 2 个基本时钟内将 TXE0 或者 RXE0 置“1”，就有可能无法对发送电路或者接收电路进行初始化。
 4. 将 TXE0 置“1”后，至少等待 1 个基本时钟（ f_{XCLK0} ），才能将发送数据设置到 TXS0。

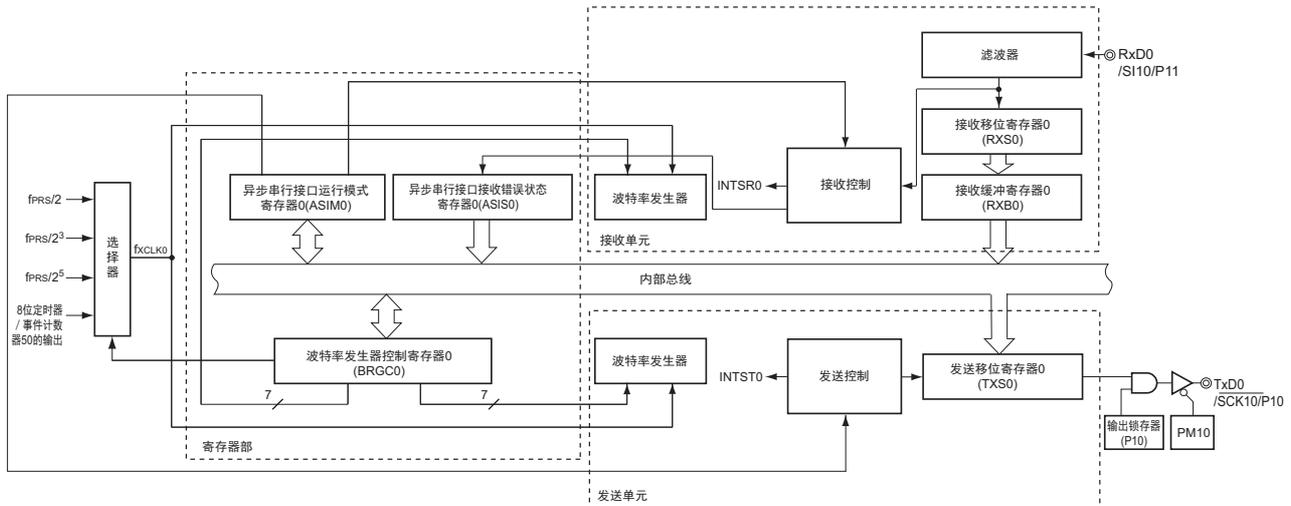
12.2 串行接口 UART0 的结构

串行接口 UART0 由以下硬件构成。

表 12-1 串行接口 UART0 的结构

项目	结构
寄存器	接收缓冲寄存器 0 (RXB0) 接收移位寄存器 0 (RXS0) 发送移位寄存器 0 (TXS0)
控制寄存器	异步串行接口运行模式寄存器 0 (ASIM0) 异步串行接口接收错误状态寄存器 0 (ASIS0) 波特率发生器控制寄存器 0 (BRGC0) 端口模式寄存器 1 (PM1) 端口寄存器 1 (P1)

图 12-1 串行接口 UART0 的框图



(1) 接收缓冲寄存器 0 (RXB0)

接收缓冲寄存器 0 是 8 位寄存器，用于保存通过接收移位寄存器 0 (RXS0) 转换的并行数据。

每接收 1 字节的数据，就传送新的接收数据。

在将数据长度指定为 7 位时，接收数据被传送到 RXB0 的 bit0 ~ bit6，并且 RXB0 的 MSB 总是为“0”。

如果发生溢出错误 (OVE0)，此时的接收数据就不被传送到 RXB0。

可通过 8 位存储器操作指令读 RXB0，但是不能写。

在产生复位信号后并且 POWER0=0，RXB0 变为“FFH”。

(2) 接收移位寄存器 0 (RXS0)

接收移位寄存器 0 将输入到 RxD0 引脚的串行数据转换为并行数据。

不能通过程序直接操作 RXS0。

(3) 发送移位寄存器 0 (TXS0)

发送移位寄存器 0 设置发送数据。通过将数据写入 TXS0，启动发送运行，并且从 Tx/D0 引脚发送串行数据。

可通过 8 位存储器操作指令写 TXS0，但是不能读。

在产生复位信号并且 POWER0=0、TXE0=0 时，TXS0 为“FFH”。

注意 1. 在将 TXE0 置“1”后，至少等待 1 个基本时钟 (f_{XCLK0})，才能将发送数据设置到 TXS0。

2. 从发送数据写入 TXS0 到产生发送结束中断信号 (INTST0) 前，不能写入下一个发送数据。

12.3 串行接口 UART0 的控制寄存器

通过以下 5 种寄存器控制串行接口 UART0。

- 异步串行接口运行模式寄存器 0 (ASIM0)
- 异步串行接口接收错误状态寄存器 0 (ASIS0)
- 波特率发生器控制寄存器 0 (BRGC0)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

(1) 异步串行接口运行模式寄存器 0 (ASIM0)

异步串行接口运行模式寄存器 0 是 8 位寄存器，用于控制串行接口 UART0 的串行通信运行。

通过 1 位或 8 位存储器操作指令设置 ASIM0。

在产生复位信号后，ASIM0 为“01H”。

图 12-2 异步串行接口运行模式寄存器 0 (ASIM0) 的格式 (1/2)

地址: FF70H 复位后: 01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	允许 / 禁止内部运行时钟的运行
0 注 1	禁止内部运行时钟的运行 (固定为低电平), 并且异步复位内部电路注 2。
1	允许内部运行时钟的运行

TXE0	允许 / 禁止发送运行
0	禁止发送运行 (同步复位发送电路)
1	允许发送运行

RXE0	允许 / 禁止接收运行
0	禁止接收运行 (同步复位接收电路)
1	允许接收运行

- 注 1. 通过将 POWER0 置“0”，RxD0 引脚的输入被固定为高电平。
2. 异步串行接口接收错误状态寄存器 0 (ASIS0)、发送移位寄存器 0 (TXS0) 和接收缓冲寄存器 0 (RXB0) 被复位。

图 12-2 异步串行接口运行模式寄存器 0 (ASIM0) 的格式 (2/2)

PS01	PS00	发送运行	接收运行
0	0	不输出奇偶校验位	无奇偶校验的接收
0	1	输出零校验	作为零校验接收 ^注
1	0	输出奇校验	作为奇校验进行判断
1	1	输出偶校验	作为偶校验进行判断

CL0	指定发送 / 接收数据的字符长度
0	数据的字符长度 =7 位
1	数据的字符长度 =8 位

SL0	指定发送数据的停止位个数
0	停止位个数 =1
1	停止位个数 =2

注 如果设置为“作为零校验接收”，就不进行奇偶校验的判断。因此，异步串行接口接收错误状态寄存器 0 (ASIS0) 的 bit2 (PE0) 不被置位，从而也不发生错误中断。

- 注意 1. 将 POWER0 置“1”，再将 TXE0 置“1”，开始发送。将 TXE0 置“0”，再将 POWER0 置“0”，停止发送。
2. 将 POWER0 置“1”，再将 RXE0 置“1”，开始接收。将 RXE0 置“0”，再将 POWER0 置“0”，停止接收。
3. 在将高电平输入 RxD0 引脚的状态下将置 POWER0 置“1”，再将 RXE0 置“1”。如果在低电平的状态下将 POWER0 置“1”，再将 RXE0 置“1”，就开始接收。
4. 通过 BRGC0 设置的基本时钟 (f_{XCLK0}) 同步 TXE0 与 RXE0。为了允许重新发送或者接收，必须在将 TXE0 或者 RXE0 清“0”后，至少等待 2 个基本时钟，再将 TXE0 或者 RXE0 置“1”。如果在 2 个基本时钟内将 TXE0 或者 RXE0 置“1”，就有可能无法对发送电路或者接收电路进行初始化。
5. 在将 TXE0 置“1”后，至少等待 1 个基本时钟 (f_{XCLK0})，才能将发送数据设置到 TXS0。
6. 在将 TXE0 位和 RXE0 位清“0”后，才能改写 PS01 位、PS00 位和 CL0 位。
7. 在将 TXE0 位清“0”后，才能改写 SL0 位。另外，因为只检查接收数据停止位中的第 1 个停止位，所以不受 SL0 位的设置值的影响。
8. 必须将 bit0 置“1”。

(2) 异步串行接口接收错误状态寄存器 0 (ASIS0)

异步串行接口接收错误状态寄存器表示串行接口 UART0 的接收结束时的错误状态。由 3 位错误标志 (PE0、FE0、OVE0) 构成。

只能通过 8 位存储器操作指令读 ASIS0。

在产生复位信号后并且 ASIM0 的 bit7 (POWER0) =0、bit5 (RXE0) =0, ASIS0 变为“00H”。另外, 读取值也为“00H”。如果发生接收错误, 就在读 ASIS0 后读取接收缓存寄存器 0 (RXB0), 并且清除错误标志。

图 12-3 异步串行接口接收错误状态寄存器 0 (ASIS0) 的格式

地址: FF73H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	表示奇偶校验错误的状态标志
0	在将 POWER0 和 RXE0 置“0”或者读 ASIS0 寄存器时
1	接收结束后发送数据的奇偶校验与奇偶校验位不匹配时

FE0	表示帧错误的状态标志
0	在将 POWER0 和 RXE0 置“0”或者读 ASIS0 寄存器时
1	接收结束后未检测出停止位时

OVE0	表示溢出错误的状态标志
0	在将 POWER0 和 RXE0 置“0”或者读 ASIS0 寄存器时
1	将接收数据设置到 RXB0 寄存器并且在读取前结束下一个接收运行时

- 注意 1. PE0 位的运行因异步串行接口运行模式寄存器 0 (ASIM0) 的 PS01 位和 PS00 位的设置值而不同。
- 接收数据的停止位与停止位个数无关只检查第 1 个位。
 - 在发生溢出错误时, 下一个接收数据不被写入接收缓冲寄存器 0 (RXB0), 而被舍弃。
 - 如果从 ASIS0 读取数据, 就产生等待。在外围硬件时钟 (f_{PRS}) 停止时, 不能从 ASIS0 读取数据。详细内容请参照“第 27 章 等待的注意事项”。

(3) 波特率发生器控制寄存器 0 (BRGC0)

波特率发生器控制寄存器选择串行接口 UART0 的基本时钟和设置 5 位计数器的分频值。

通过 8 位存储器操作指令设置 BRGC0。

在产生复位信号后, BRGC0 变为“1FH”。

图 12-4 波特率发生器控制寄存器 0 (BRGC0) 的格式

地址: FF71H 复位后: 1FH R/W

符号	7	6	5	4	3	2	1	0
BRGC0	TPS01	TPS00	0	MDL04	MDL03	MDL02	MDL01	MDL00

TPS01	TPS00	选择基本时钟 (f_{XCLK0})		
		$f_{PRS}=2\text{MHz}$	$f_{PRS}=5\text{MHz}$	$f_{PRS}=10\text{MHz}$
0	0	TM50 的输出注		
0	1	$f_{PRS}/2$	1MHz	2.5MHz
1	0	$f_{PRS}/2^3$	250kHz	625kHz
1	1	$f_{PRS}/2^5$	62.5kHz	156.25kHz

MDL04	MDL03	MDL02	MDL01	MDL00	k	选择 5 位计数器的输出时钟
0	0	x	x	x	x	禁止设置
0	1	0	0	0	8	$f_{XCLK0}/8$
0	1	0	0	1	9	$f_{XCLK0}/9$
0	1	0	1	0	10	$f_{XCLK0}/10$
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	0	1	0	26	$f_{XCLK0}/26$
1	1	0	1	1	27	$f_{XCLK0}/27$
1	1	1	0	0	28	$f_{XCLK0}/28$
1	1	1	0	1	29	$f_{XCLK0}/29$
1	1	1	1	0	30	$f_{XCLK0}/30$
1	1	1	1	1	31	$f_{XCLK0}/31$

注 选择 TM50 输出作为基本时钟时的注意事项如下所示。

- TM50 与 CR50 匹配时的清除 & 启动模式 (TMC506=0)
允许定时器 F/F 的取反运行 (TMC501=1), 事先开始 8 位定时器 / 事件计数器 50 的运行。
- PWM 模式 (TMC506=1)
设置占空比为 50% 的时钟, 事先开始 8 位定时器 / 事件计数器 50 的运行。
在其中任一模式中, 都不需要允许 TO50 输出 (TOE50=1)。

- 注意 1. 在将 ASIM0 寄存器的 bit6 (TXE0) 和 bit5 (RXE0) 置“0”后, 才能改写 MDL04 ~ MDL00 位。
2. 在将 ASIM0 寄存器的 bit7 (POWER0) 置“0”后, 才能改写 TPS01 位和 TPS00 位。
3. 对 5 位计数器的输出时钟进行 1/2 分频后的值为波特率值。

- 备注 1. f_{XCLK0} : 通过 TPS01 位和 TPS00 位选择的基本时钟的频率
2. f_{PRS} : 外围硬件时钟频率
3. k: 通 MDL04 ~ MDL00 位设置的值 (k = 8、9、10、...31)
4. x: 任意
5. TMC506: 8 位定时器模式控制寄存器 50 (TMC50) 的 bit6
TMC501: TMC50 的 bit1

(4) 端口模式寄存器 1 (PM1)

端口模式寄存器 1 以 1 位单位设置端口 1 的输入 / 输出。

将 P10/TxD0/SCK10 引脚用作串行接口的数据输出时，必须将 PM10 置“0”，将 P10 的输出锁存器置“1”。

将 P11/RxD0/SI10 引脚用作串行接口的数据输入时，必须将 PM11 置“1”。此时，P11 的输出锁存器既可是“0”也可是“1”。

通过 1 位或 8 位存储器操作指令设置 PM1。

在产生复位信号后，PM1 变为“FFH”。

图 12-5 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 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

PM1n	选择 P1n 引脚的输入 / 输出模式 (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)

12.4 串行接口 UART0 的运行

串行接口 UART0 有以下两种模式。

- 运行停止模式
- 异步串行接口（UART）模式

12.4.1 运行停止模式

在运行停止模式中，不执行串行通信，从而可降低功耗。此外，引脚通常可用作通用端口。将 ASIM0 的 bit7 ~ bit5（POWER0、TXE0 和 RXE0）置“0”，设置为运行停止模式。

(1) 使用的寄存器

通过异步串行接口运行模式寄存器 0（ASIM0）设置运行停止模式。

通过 1 位或 8 位存储器操作指令设置 ASIM0。

在产生复位信号后，ASIM0 变为“01H”。

地址：FF70H 复位后：01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	允许 / 禁止内部运行时钟的运行
0 注 1	禁止内部运行时钟的运行（固定为低电平），并且异步复位内部电路注 2。

TXE0	允许 / 禁止发送运行
0	禁止发送运行（同步复位发送电路）

RXE0	允许 / 禁止接收运行
0	禁止接收运行（同步复位接收电路）

- 注 1. 通过将 POWER0 置“0”，RxD0 引脚的输入被固定为高电平。
2. 异步串行接口接收错误状态寄存器 0（ASIS0）、发送移位寄存器 0（TXS0）和接收缓冲寄存器 0（RXB0）被复位。

注意 将 TXE0 和 RXE0 置“0”，再将 POWER0 置“0”，停止运行。
将 POWER0 置“1”，再将 TXE0 和 RXE0 置“1”，开始通信。

备注 将 RxD0/SI10/P11 和 TxD0/SCK10/P10 引脚用作通用端口时，参照“第 4 章 端口功能”。

12.4.2 异步串行接口 (UART) 模式

异步串行接口 (UART) 模式是发送 / 接收紧接着起始位的 1 字节数据的模式。可执行全双工运行。因为内置 UART 专用波特率发生器, 所以能以广范围的波特率进行通信。

(1) 使用的寄存器

- 异步串行接口运行模式寄存器 0 (ASIM0)
- 异步串行接口接收错误状态寄存器 0 (ASIS0)
- 波特率发生器控制寄存器 0 (BRGC0)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

UART 模式的基本运行设置步骤示例如下所示。

1. 设置 BRGC0 寄存器 (参照图 12-4)
2. 设置 ASIM0 寄存器的 bit1 ~ bit4 (SL0、CL0、PS00、PS01) (参照图 12-2)
3. 将 ASIM0 寄存器的 bit7 (POWER0) 置“1”
4. 将 ASIM0 寄存器 bit6 (TXE0) 置“1”→可发送
将 ASIM0 寄存器的 bit5 (RXE0) 置“1”→可接收
5. 将数据写入 TXS0 寄存器 → 开始数据发送

注意 必须在考虑与通信方的关系后, 设置端口模式寄存器和端口寄存器。

寄存器设置与引脚的关系如下所示。

表 12-2 寄存器设置与引脚的关系

POWER0	TXE0	RXE0	PM10	P10	PM11	P11	UART0 的运行	引脚功能	
								TxD0/SCK10/P10	RxD0/SI10/P11
0	0	0	x 注	x 注	x 注	x 注	停止	SCK10/P10	SI10/P11
1	0	1	x 注	x 注	1	x	接收	SCK10/P10	RxD0
	1	0	0	1	x 注	x 注	发送	TxD0	SI10/P11
	1	1	0	1	1	x	发送 / 接收	TxD0	RxD0

注 可作为端口功能或者串行接口 SCI10 设置。

备注 x: 忽略
 POWER0: 异步串行接口运行模式寄存器 0 (ASIM0) 的 bit7
 TXE0: ASIM0 的 bit6
 RXE0: ASIM0 的 bit5
 PM1x: 端口模式寄存器
 P1x: 端口的输出锁存器

(2) 通信运行

(a) 正常发送 / 接收数据的格式和波形示例

正常发送 / 接收数据的格式和波形示例如图 12-6 和图 12-7 所示。

图 12-6 正常 UART 发送 / 接收数据的格式



1 个数据帧由以下各位组成。

- 起始位 ...1 位
- 字符位 ...7 位或者 8 位 (LSB 优先)
- 奇偶校验位 ... 偶校验 / 奇校验 / 零校验 / 无奇偶校验
- 停止位 ...1 位或者 2 位

通过异步串行接口运行模式寄存器 0 (ASIM0) 指定 1 个数据帧中的字符位长度、停止位长度以及选择奇偶校验。

图 12-7 正常 UART 发送 / 接收数据的波形示例

1. 数据长度：8 位、奇偶校验：偶校验、停止位：1 位、通信数据：55H



2. 数据长度：7 位、奇偶校验：奇校验、停止位：2 位、通信数据：36H



3. 数据长度：8 位、奇偶校验：无奇偶校验、停止位：1 位、通信数据：87H



(b) 奇偶校验的种类和运行

奇偶校验位检测通信数据的位错误。通常，发送端和接收端使用相同种类的奇偶校验位。偶校验和奇校验都可检测出 1 位（奇数）错误。在零校验和无奇偶校验时，不能检测出错误。

(i) 偶校验

• 发送时

将包括奇偶校验位的发送数据中值为“1”的位个数控制为偶数。

奇偶校验位的值如下所示。

发送数据中值为“1”的位个数为奇数：1

发送数据中值为“1”的位个数为偶数：0

• 接收时

对包括奇偶校验位的接收数据中值为“1”的位个数进行计数，如果为奇数，就发生奇偶校验错误。

(ii) 奇校验

• 发送时

与偶校验相反，将包括奇偶校验位的发送数据中值为“1”的位个数控制为奇数。

奇偶校验位的值如下所示。

发送数据中值为“1”的位个数为奇数：0

发送数据中值为“1”的位个数为偶数：1

• 接收时

对包括奇偶校验位的接收数据中值为“1”的位个数进行计数，如果为偶数，就发生奇偶校验错误。

(iii) 零校验

发送期间，与发送数据无关，将奇偶校验位置“0”。

接收期间，不进行奇偶校验位的检测。因此，无论校验位为“0”或者“1”，都不发生奇偶校验错误。

(iv) 无奇偶校验

发送数据中无奇偶校验位。

接收期间，以无奇偶位进行接收。因为没有奇偶校验位，所以不发生奇偶校验错误。

(c) 发送

通过将异步串行接口运行模式寄存器 0（ASIM0）的 bit7（POWER0）置“1”、将 ASIM0 的 bit6（TXE0）置“1”，进入允许发送状态，并且通过将发送数据写入发送移位寄存器 0（TXS0），启动发送运行。自动附加起始位、奇偶校验位以及停止位。

开始发送运行后，先从 TxD0 引脚输出起始位，然后从 LSB 开始依次输出发送数据。一旦发送结束，就附加由 ASIM0 设置的奇偶校验位和停止位，最后产生发送结束中断请求（INTST0）。

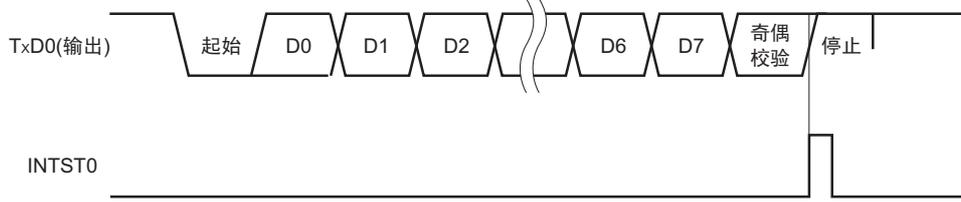
直到下一个发送数据写入 TXS0，才能中止发送运行。

产生发送结束中断请求（INTST0）的时序如图 12-8 所示。在输出最后的停止位的同时产生 INTST0。

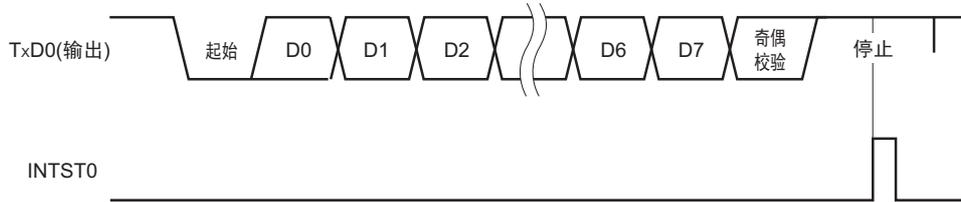
注意 从发送数据写入 TXS0 到产生发送结束中断信号（INTST0）前，不能写入下一个发送数据。

图 12-8 发送结束中断请求时序

1. 停止位长度：1



2. 停止位长度：2



(d) 接收

如果将异步串行接口运行模式寄存器 0 (ASIM0) 的 bit7 (POWER0) 置“1”，再将 ASIM0 的 bit5 (RXE0) 置“1”，就进入允许接收状态，并且进行 RxD0 引脚输入的采样。

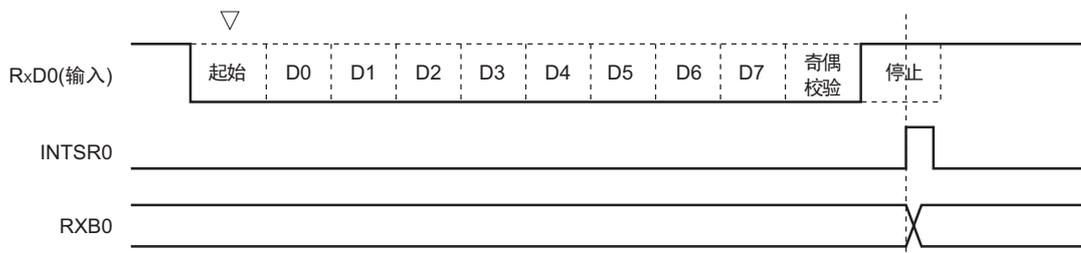
如果检测出 RxD0 引脚输入的下沿，波特率发生器的 5 位计数器就开始计数，在对波特率发生器控制寄存器 0 (BRGC0) 的设置值进行计数后，再次进行 RxD0 引脚输入的采样（如图 12-9 中的“▽”）。如果采样后的 RxD0 引脚为低电平，就识别为起始位。

在检测出起始位后，开始接收运行，并且根据设置的波特率，将串行数据依次保存到接收移位寄存器 0 (RXS0)。一旦接收到停止位，就在发生接收结束中断 (INTSR0) 的同时，将 RXS0 的数据写入接收缓冲寄存器 0 (RXB0)。但是，在发生溢出错误 (OVE0) 时，不能将接收的数据写入 RXB0。

即使在接收过程中发生奇偶校验错误 (PE0)，也继续接收，直到停止位的接收位置，并且在接收结束后发生接收错误中断 (INTSR0)。

在接收结束和接收错误时发生 INTSR0。

图 12-9 接收结束中断请求时序



注意 1. 如果发生接收错误，在读取异步串行接口接收错误状态寄存器 0 (ASIS0) 后，读取接收缓冲寄存器 0 (RXB0)，并且清除错误标志。

如果不读取 RXB0，就在接收下一个数据时发生溢出错误，而且将一直持续接收错误的状态。

2. 接收运行总是以“停止位个数 =1”进行运行。忽略第 2 个停止位。

(e) 接收错误

接收期间的错误有 3 种，分别是奇偶校验错误、帧错误和溢出错误。如果数据接收的结果错误标志在异步串行接口接收错误状态寄存器 0（ASIS0）中置位，就发生接收错误中断（INTSR0）。

在接收错误中断（INTSR0）处理中，可通过读取 ASIS0 的内容，检测出在接收期间发生了哪种错误（参照图 12-3）

通过读取 ASIS0，可将 ASIS0 的内容清“0”。

表 12-3 接收错误源

接收错误	接收错误源
奇偶校验错误	指定的发送时的奇偶校验与接收数据的奇偶校验不匹配
帧错误	未检测出停止位
溢出错误	在从接收缓冲寄存器 0（RXB0）读取数据前结束下一个数据的接收

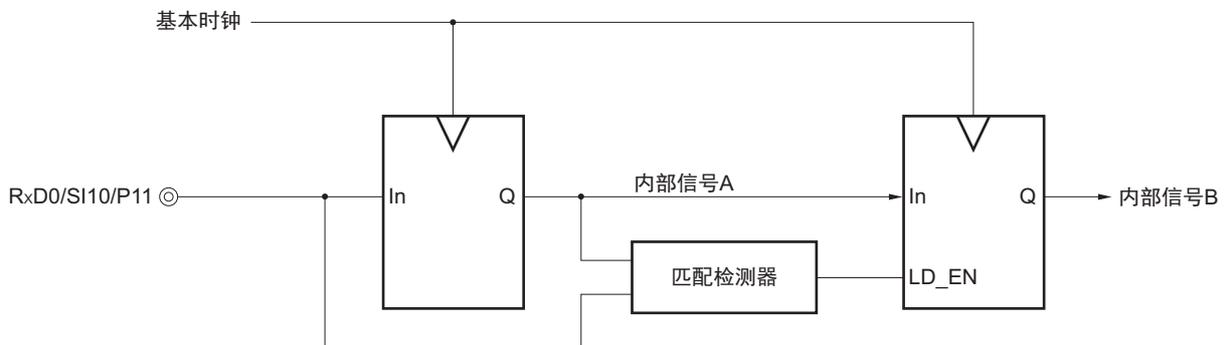
(f) 接收数据的噪声滤波器

接收数据的噪声滤波器以预分频器输出的基本时钟对 RxD0 信号进行采样。

如果对相同的值进行 2 次采样，匹配检测器的输出就发生变化，并且作为输入数据被采样。

在如图 12-10 所示的电路中，接收运行的内部处理比外部的信号状态迟 2 个时钟。

图 12-10 噪声滤波器电路



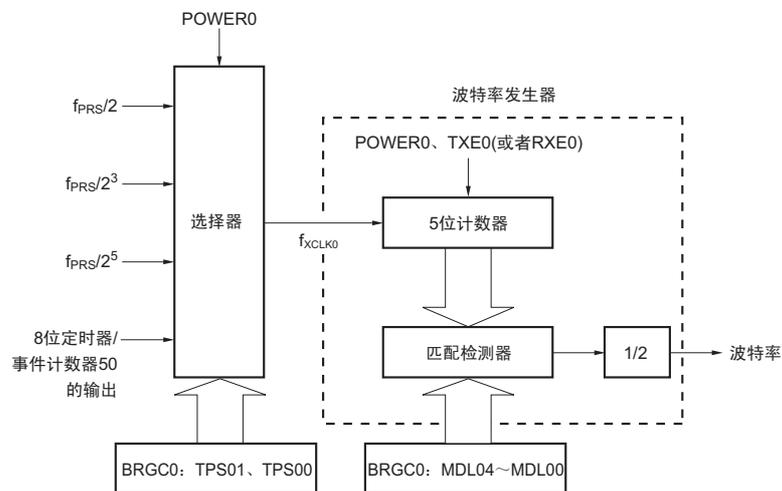
12.4.3 专用波特率发生器

专用波特率发生器由源时钟选择器和 5 位可编程计数器构成，生成 UART0 发送 / 接收时的串行时钟。发送和接收分别使用不同的的 5 位计数器。

(1) 波特率发生器的结构

- 基本时钟**
 在异步串行接口运行模式寄存器 0 (ASIM0) 的 bit7 (POWER0) 为“1”时，向各模块提供由波特率发生器控制寄存器 0 (BRGC0) 的 bit7 和 bit6 (TPS01 和 TPS00) 选择的时钟。该时钟被称为基本时钟，其频率用 f_{XCLK0} 表示。在 POWER0=0 时，基本时钟固定为低电平。
- 发送计数器**
 在异步串行接口运行模式寄存器 0 (ASIM0) 的 bit7 (POWER0) 或者 bit6 (TXE0) 为“0”时，发送计数器在清“0”的状态下停止运行。
 通过将 POWER0 和 TXE0 置“1”，计数器开始计数。
 在进行第 1 个发送时，通过将数据写入发送移位寄存器 0 (TXS0)，计数器清“0”。
- 接收计数器**
 在异步串行接口运行模式寄存器 0 (ASIM0) 的 bit7 (POWER0) 或者 bit5 (RXE0) 为“0”时，接收计数器在清“0”的状态下停止运行。
 在检测出起始位时，计数器开始计数。
 在接收 1 帧的数据后，计数器停止计数，直到检测出下一个起始位。

图 12-11 波特率发生器的结构



备注 POWER0: 异步串行接口运行模式寄存器 0 (ASIM0) 的 bit7

TXE0: ASIM0 的 bit6

RXE0: ASIM0 的 bit5

BRGC0: 波特率发生器控制寄存器 0

(2) 串行时钟的生成

通过设置波特率发生器控制寄存器 0 (BRGC0) 指定生成的串行时钟。

可通过 BRGC0 的 bit7 和 bit6 (TPS01 和 TPS00) 选择 5 位计数器的输入时钟，通过 BRGC0 的 bit4 ~ bit0 (MDL04 ~ MDL00) 设置 5 位计数器的分频值 ($f_{XCLK0}/8 \sim f_{XCLK0}/31$)。

12.4.4 波特率的计算

(1) 波特率的计算公式

可通过以下公式计算波特率。

$$\bullet \text{ 波特率} = \frac{f_{\text{XCLK0}}}{2 \times k} \text{ [bps]}$$

f_{XCLK0} : 通过 BRGC0 寄存器的 TPS01 位和 TPS00 位选择的基本时钟的频率

k : 通过 BRGC0 寄存器的 MDL04 ~ MDL00 位设置的值 ($k = 8、9、10、\dots 31$)

表 12-4 TPS01 和 TPS00 的设置值

TPS01	TPS00	选择基本时钟 (f_{XCLK0})		
		$f_{\text{PRS}}=2\text{MHz}$	$f_{\text{PRS}}=5\text{MHz}$	$f_{\text{PRS}}=10\text{MHz}$
0	0	TM50 的输出注		
0	1	$f_{\text{PRS}}/2$	1MHz	2.5MHz
1	0	$f_{\text{PRS}}/2^3$	250kHz	625kHz
1	1	$f_{\text{PRS}}/2^5$	62.5kHz	156.25kHz

注 选择 TM50 输出作为基本时钟时的注意事项如下所示。

- TM50 与 CR50 匹配时的清除 & 启动模式 (TMC506=0)
允许定时器 F/F 的取反运行 (TMC501=1)，事先开始 8 位定时器 / 事件计数器 50 的运行。
- PWM 模式 (TMC506=1)
设置占空比为 50% 的时钟，事先开始 8 位定时器 / 事件计数器 50 的运行。
在其中任一模式中，都不需要允许 TO50 输出 (TOE50=1)。

(2) 波特率的误差

可通过以下公式计算波特率的误差。

$$\text{误差}(\%) = \left[\frac{\text{实际波特率(有误差的波特率)}}{\text{理想波特率(正确波特率)}} - 1 \right] \times 100 [\%]$$

注意 1. 发送时的波特率误差必须在接收方的允许误差范围内。

2. 接收时的波特率误差必须满足“(4) 接收时的波特率允许范围”中所示的范围。

示例: 基本时钟频率 = 2.5MHz = 2500000Hz

BRGC0 寄存器的 MDL04 ~ MDL00 位的设置值 = 10000B ($k=16$)

理想波特率 = 76800bps

波特率 = $2.5\text{M}/(2 \times 16)$

= $2500000/(2 \times 16) = 78125$ [bps]

错误 = $(78125/76800 - 1) \times 100$

= 1.725[%]

(3) 波特率的设置示例

表 12-5 波特率发生器的设置数据

波特率 [bps]	f _{PRS} =2.0 MHz				f _{PRS} =5.0 MHz				f _{PRS} =10.0 MHz			
	TPS01、 TPS00	k	计算值	ERR[%]	TPS01、 TPS00	k	计算值	ERR[%]	TPS01、 TPS00	k	计算值	ERR[%]
4800	2H	26	4808	0.16	3H	16	4883	1.73	—	—	—	—
9600	2H	13	9615	0.16	3H	8	9766	1.73	3H	16	9766	1.73
10400	2H	12	10417	0.16	2H	30	10417	0.16	3H	15	10417	0.16
19200	1H	26	19231	0.16	2H	16	19531	1.73	3H	8	19531	1.73
24000	1H	21	23810	-0.79	2H	13	24038	0.16	2H	26	24038	0.16
31250	1H	16	31250	0	2H	10	31250	0	2H	20	31250	0
33600	1H	15	33333	-0.79	2H	9	34722	3.34	2H	19	32895	-2.1
38400	1H	13	38462	0.16	2H	8	39063	1.73	2H	16	39063	1.73
56000	1H	9	55556	-0.79	1H	22	56818	1.46	2H	11	56818	1.46
62500	1H	8	62500	0	1H	20	62500	0	2H	10	62500	0
76800	—	—	—	—	1H	16	78125	1.73	2H	8	78125	1.73
115200	—	—	—	—	1H	11	113636	-1.36	1H	22	113636	-1.36
153600	—	—	—	—	1H	8	156250	1.73	1H	16	156250	1.73
312500	—	—	—	—	—	—	—	—	1H	8	312500	0
625000	—	—	—	—	—	—	—	—	—	—	—	—

备注 TPS01、TPS00: 波特率发生器控制寄存器 0 (BRGC0) 的 bit6 和 bit7 (设置基本时钟 (f_{XCLK0}))

k: 通过 BRGC0 的 MDL04 ~ MDL00 位设置的值 (k=8、9、10、...31)

f_{PRS}: 外围硬件时钟频率

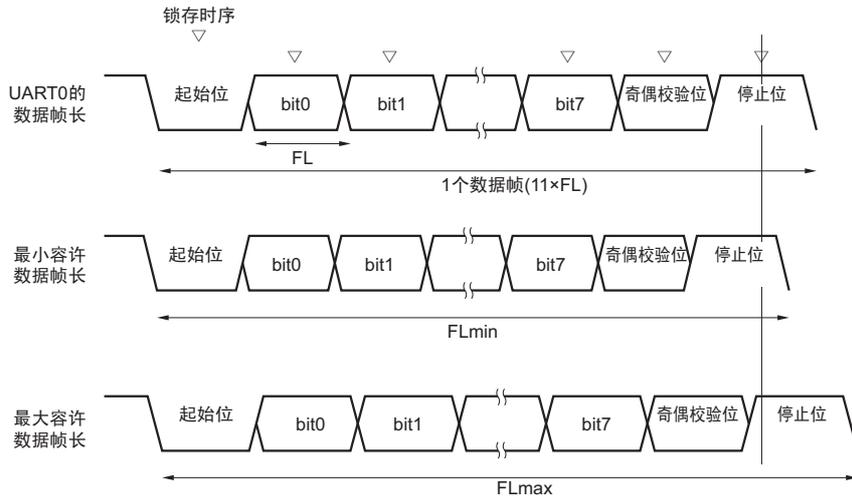
ERR: 波特率误差

(4) 接收时的波特率允许范围

接收期间能容许的发送方的波特率范围如下所示。

注意 通过以下计算式设置接收期间的波特率误差，使其在允许误差范围内。

图 12-12 接收时的波特率允许范围



如图 12-2 所示，在检测出起始位后，通过波特率发生器控制寄存器 0（BRGC0）设置的计数器决定接收数据的锁存时序。如果该时序在最后数据（停止位）前，就能进行正常接收。

如果将上述内容用于 11 位接收，则理论上可得出以下结果。

$$FL = (\text{Brate})^{-1}$$

Brate: UART0 波特率

k: BRGC0 的设置值

FL: 1 位数据长度

锁存时序的容限: 2 个时钟

$$\text{最小允许数据帧长: } FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

因此，可接收的发送目标的最大波特率如下所示。

$$BR_{\max} = (FL_{\min}/11)^{-1} = \frac{22k}{21k+2} \text{ Brate}$$

同样，最大允许数据帧长的计算如下所示。

$$\begin{aligned} \frac{10}{11} \times FL_{\max} &= 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL \\ FL_{\max} &= \frac{21k-2}{20k} FL \times 11 \end{aligned}$$

因此，可接收的发送目标的最小波特率如下所示。

$$BR_{\min} = (FL_{\max}/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

从上述的最小 / 最大波特率值的计算式中，求 UART0 和发送目标间的波特率容许误差，如下所示。

表 12-6 最大 / 最小波特率的允许误差

分频比 (k)	最大波特率的允许误差	最小波特率允许误差
8	+3.53%	-3.61%
16	+4.14%	-4.19%
24	+4.34%	-4.38%
31	+4.44%	-4.47%

备注 1. 接收时的允许误差取决于每帧的位数、输入时钟频率和分频比 (k)。输入时钟频率和分频比 (k) 越大，允许误差就越大。

2. k: BRGC0 的设置值

第 13 章 串行接口 UART6

13.1 串行接口 UART6 的功能

串行接口 UART6 有以下两种模式。

(1) 运行停止模式

这是在不执行串行通信时使用的模式。能降低功耗。

详细内容请参照“13.4.1 运行停止模式”。

(2) 异步串行接口 (UART) 模式

异步串行接口 (UART) 模式支持 LIN (Local Interconnect Network) -bus。功能概要如下所示。

详细内容请参照“13.4.2 异步串行接口 (UART) 模式”和“13.4.3 专用波特率发生器”。

- 最大传送速度：625kbps
- 双引脚结构 TxD6：发送数据的输出引脚
Rx6D6：接收数据的输入引脚
- 可选择7位或者8位的通信数据长度
- 可通过内置专用8位波特率发生器，设置任意的波特率
- 可独立进行发送运行和接收运行（全双工运行）
- 可选择MSB/LSB优先通信
- 可进行发送反相运行
- 同步间隔场的发送可从13位到20位中选择
- 可识别至少11位的同步间隔场的接收（有SBF接收标志）

注意 1. Tx6D6 输出取反功能只能取反发送侧，不能取反接收侧，因此，在使用 Tx6D6 输出取反功能时，对方也必须在取反电平下进行接收。

2. 在提供给串行接口 UART6 的时钟未停止时（例如，HALT 模式），可继续正常运行。在提供给串行接口 UART6 的时钟停止时（例如，STOP 模式），各寄存器保持时钟停止前的值，并且停止运行。Tx6D6 引脚输出也相同，保持时钟停止前的值，并且输出。但是，不保证重新提供时钟后的运行，因此，在重新开始提供时钟后，必须将 POWER6、RXE6、TXE6 置“0”，进行电路复位。
3. 通过将 POWER6 置“1”，再将 TXE6 置“1”（发送）或者将 RXE6 置“1”（接收），开始通信。
4. 通过 CKSR6 设置的基本时钟 (f_{XCLK6}) 同步 TXE6 与 RXE6。在允许重新开始传送或者接收时，必须在将 TXE6 或者 RXE6 置“0”后至少经过 2 个基本时钟，再将 TXE6 或者 RXE6 置“1”。如果在 2 个基本时钟内将 TXE6 或者 RXE6 置“1”，就有可能无法对发送电路或者接收电路进行初始化。
5. 将 TXE6 置“1”后，至少等待 1 个基本时钟 (f_{XCLK6})，才能将发送数据设置到 TXB6。
6. 连续发送时，从停止位到一个起始位的通信时序比正常时延迟 2 个宏运行时钟。但是，通过检测起始位，接收侧进行时序的初始化，因此不会影响通信结果。另外，在 LIN 通信运行下使用 UART6 时，不能使用连续发送功能。

备注 LIN 为 Local Interconnect Network 的简称，是以降低车载网络成本为目的的低速（1 ~ 20kbps）的串行通信协议。

LIN 通信为单主控通信，对 1 个主控设备最多可连接 15 个从属设备。

LIN 的从属设备用于开关、执行器、传感器等的控制，这些设备通过 LIN 的网络连接到 LIN 的主控设备。

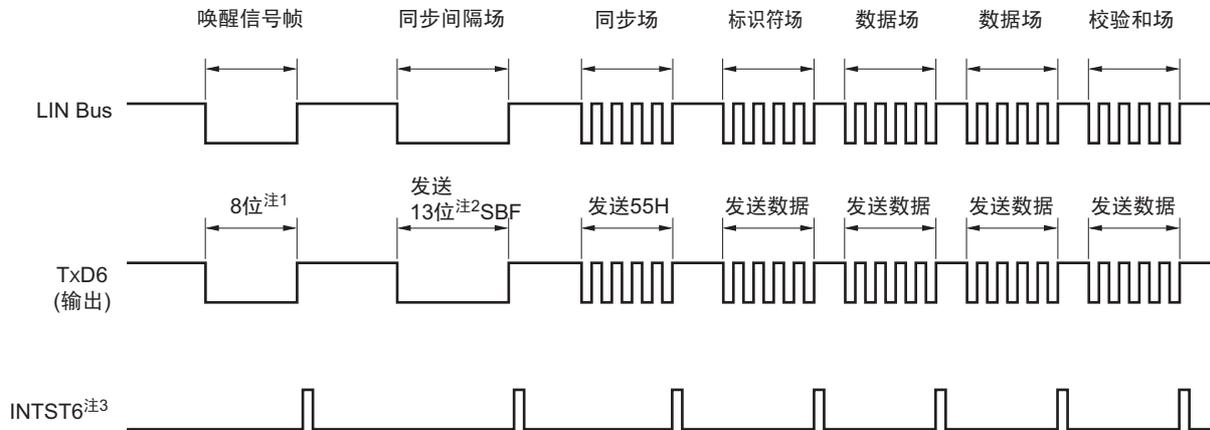
LIN 的主控设备通常连接在 CAN (Controller Area Network) 等的网络。

另外，LIN 总线为单线式，通过符合 ISO9141 标准的收发器连接各节点。

在 LIN 协议中，主控设备发送附加波特率信息的帧，而从属设备接收此帧，并且校正与主控设备的波特率误差。因此，只要从属设备的波特率误差不超过 $\pm 15\%$ ，就可进行通信。

LIN 的发送操作和接收操作的概略如图 13-1、图 13-2 所示。

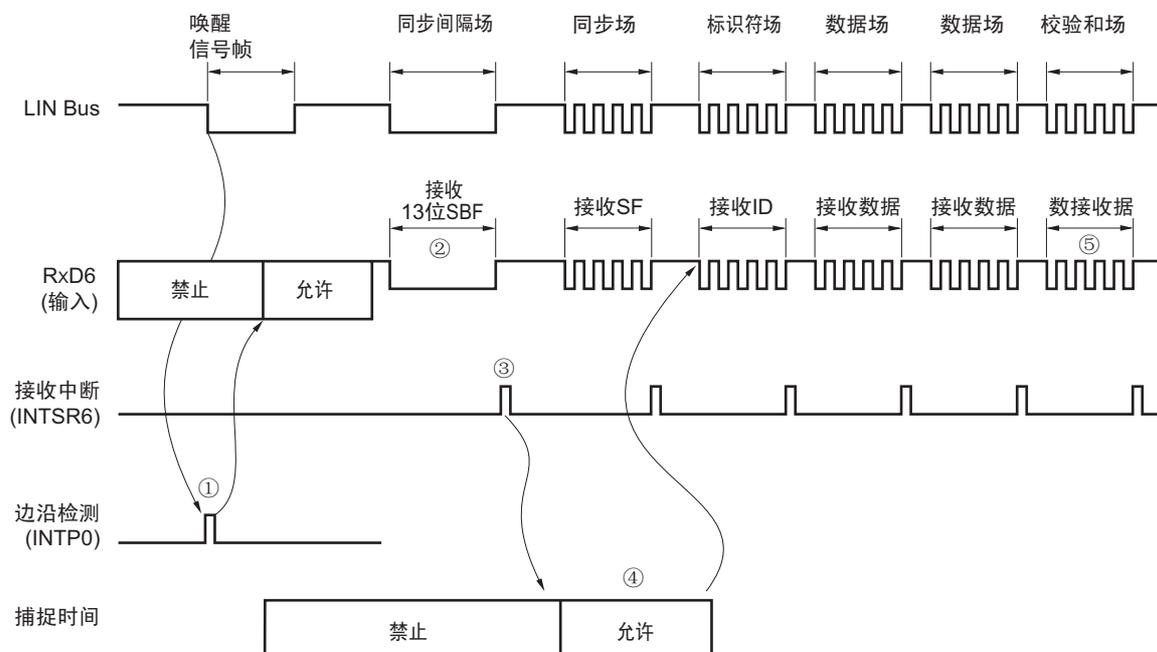
图 13-1 LIN 的发送操作



- 注 1. 用 8 位模式的“80H”发送代替唤醒信号帧。
 2. 通过硬件执行同步间隔场的输出。输出宽度为异步串行接口控制寄存器 6 (ASICL6) 的 bit4 ~ bit2 (SBL62 ~ SBL60) 设置的位宽 (参照“13.4.2 (2) (h) SBF 发送”)。
 3. 在各发送结束时输出 INTST6。在发送 SBF 时也输出 INTST6。

备注 通过软件设置各场间的间隔。

图 13-2 LIN 的接收操作



接收处理的流程如下所示。

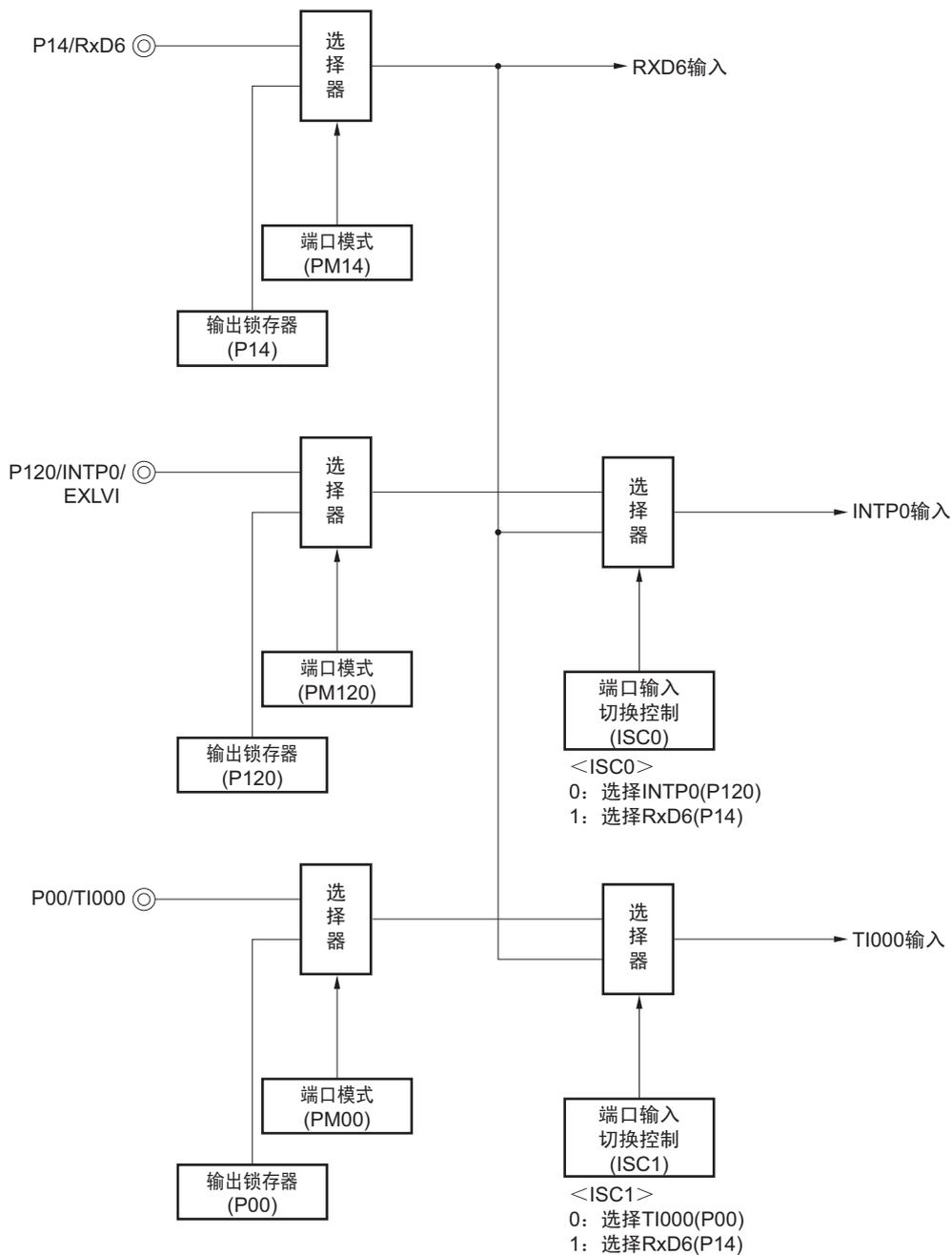
1. 通过引脚的边沿检测进行唤醒信号的检测。通过唤醒信号允许UART6，并且设置为SBF接收模式。
2. 进行接收运行直到检测出STOP位。如果通过SBF检测出至少11位的低电平数据，就判断为SBF接收正常结束，并且输出中断信号。如果通过SBF检测出不满11位的低电平数据，就判断为SBF接收错误，并且不输出中断信号而返回SBF接收模式。
3. 在正常结束SBF接收时，输出中断信号。通过SBF接收结束中断处理启动16位定时器/事件计数器00，测量同步场的位间隔（脉宽）（参照“6.4.8 脉冲的测量运行”）。另外，OVE6、PE6、FE6的各错误检测被抑制，UART通信的错误检测处理以及移位寄存器和RXB6的数据传送也不执行。移位寄存器保持复位值“FFH”。
4. 从同步场的位间隔算出波特率误差，并且在SF接收后停止UART6的运行，重新设置波特率发生器控制寄存器6（BRGC6）。
5. 通过软件执行校验和场的区别。在接收校验和场后，初始化UART6，并且通过软件执行重新设置为SBF接收模式的处理。

用于 LIN 的接收操作的端口结构图如图 13-3 所示。

通过外部中断（INTP0）的边沿检测，接收从 LIN 的主控设备发送来的唤醒信号。另外，可通过 16 位定时器 / 事件计数器 00 的外部事件捕捉运行，测量从 LIN 的主控设备发送来的同步场的宽度，并且计算波特率误差。

可通过控制端口输入的切换（ISC0/ISC1），在外部不需要 RxD6 和 INTP0、TI000 的接线的情况下，将接收端口输入（RxD6）的输入源输入到外部中断（INTP0）和 16 位定时器 / 事件计数器 00。

图 13-3 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、bit1（参照图 13-11）。

在 LIN 通信运行中使用的外围功能如下所示。

<使用的外围功能>

- 外部中断（INTP0）：检测唤醒信号
用途：检测唤醒信号的边沿和是否开始通信。
- 16 位定时器/事件计数器 00（TI000）：检测波特率误差
用途：检测同步场（SF）的宽度，并且通过与位数相除检测波特率误差（通过捕捉模式测量 TI000 输入边沿的间隔）。
- 串行接口 UART6

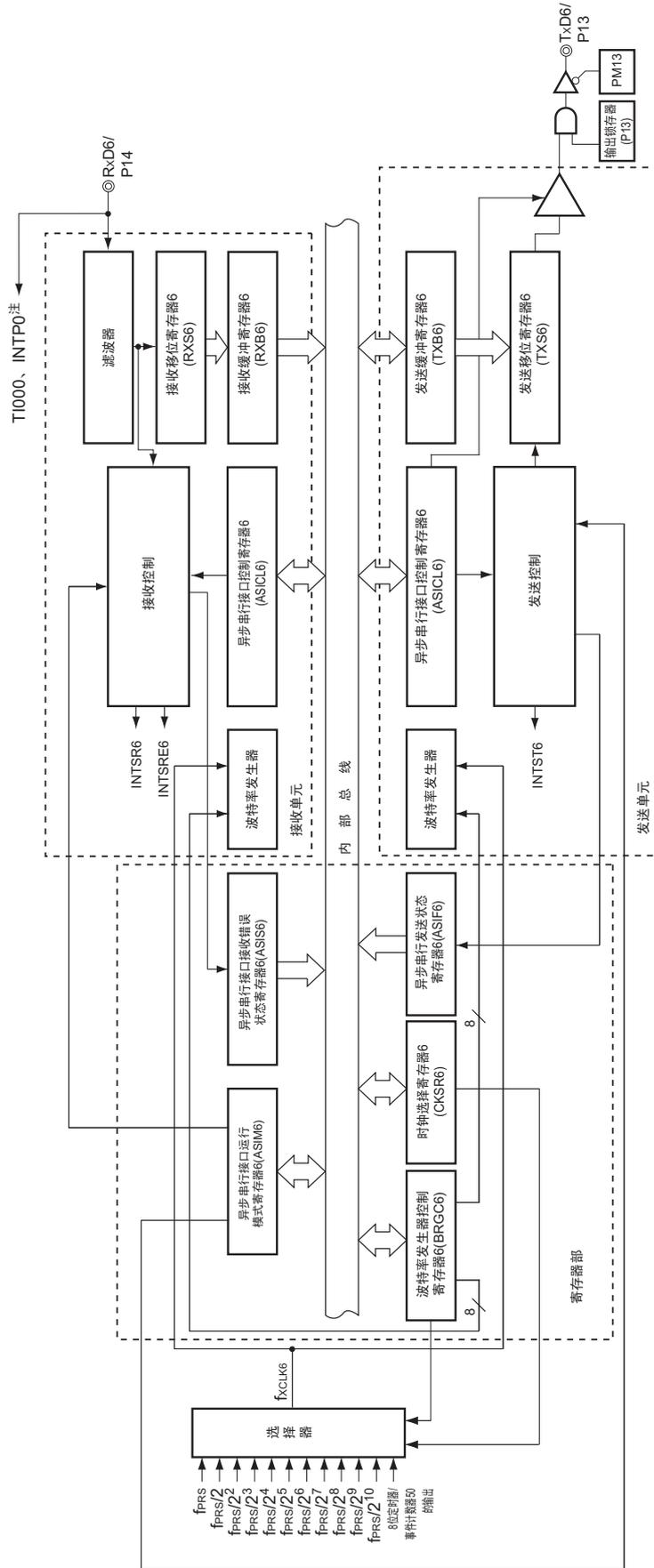
13.2 串行接口 UART6 的结构

串行接口 UART6 由以下硬件构成。

表 13-1 串行接口 UART6 的结构

项目	结构
寄存器	接收缓冲寄存器 6 (RXB6) 接收移位寄存器 6 (RXS6) 发送缓冲寄存器 6 (TXB6) 发送移位寄存器 6 (TXS6)
控制寄存器	异步串行接口运行模式寄存器 6 (ASIM6) 异步串行接口接收错误状态寄存器 6 (ASIS6) 异步串行接口发送状态寄存器 6 (ASIF6) 时钟选择寄存器 6 (CKSR6) 波特率发生器控制寄存器 6 (BRGC6) 异步串行接口控制寄存器 6 (ASICL6) 输入切换控制寄存器 (ISC) 端口模式寄存器 1 (PM1) 端口寄存器 1 (P1)

图 13-4 串行接口 UART6 的框图



(1) 接收缓冲寄存器 6 (RXB6)

接收缓冲寄存器 6 是 8 位寄存器，用于保存通过接收移位寄存器 6 (RXS6) 转换的并行数据。

每接收 1 字节的数据，就从 RXS6 传送来新的接收数据。

在将数据长度指定为 7 位时，RXB6 如下变化。

- 在进行 LSB 优先接收时，接收数据被传送到 RXB6 的 bit0 ~ bit6，并且 RXB6 的 MSB 总是为“0”。
- 在进行 MSB 优先接收时，接收数据被传送到 RXB6 的 bit1 ~ bit7，并且 RXB6 的 LSB 总是为“0”。

如果发生溢出错误 (OVE6)，此时的接收数据就不被传送到 RXB6。

可通过 8 位存储器操作指令读 RXB6，但是不能写。

在产生复位信号后，RXB6 变为“FFH”。

(2) 接收移位寄存器 6 (RXS6)

接收移位寄存器 6 将输入到 RxD6 引脚的串行数据转换为并行数据。

不能通过程序直接操作 RXS6。

(3) 发送缓冲寄存器 6 (TXB6)

发送缓冲寄存器 6 是设置发送数据的寄存器。通过给 TXB6 写发送数据，开始发送运行。

可通过 8 位存储器操作指令读写 TXB6。

在产生复位信号后，TXB6 为“FFH”。

注意 1. 在异步串行接口发送状态寄存器 6 (ASIF6) 的 bit1 (TXBF6) 为“1”时，不能给 TXB6 写数据。

2. 在通信运行过程中 (异步串行接口运行模式寄存器 6 (ASIM6) 的 bit7、bit6 (POWER6、TXE6) 都为“1”或者 ASIM6 的 bit7、bit5 都为“1”)，能通过软件执行 TXB6 的刷新 (写相同的值) 运行。

3. 在将 TXE6 置“1”后，至少等待 1 个基本时钟 (f_{XCLK6})，才能给 TXB6 设置发送数据。

(4) 发送移位寄存器 6 (TXS6)

发送移位寄存器 6 从 TxD6 引脚将来自 TXB6 的数据作为串行数据进行发送。在第 1 次发送时，TXB6 的数据传送时序为写 TXB6 之后立即进行传送。在连续发送时，TXB6 的数据传送时序为在发送 1 帧后马上产生 INTST6 前进行传送。另外，TXB6 的数据传送和 TxD6 引脚的发送都在基本时钟的下降沿时执行。

不能通过程序直接操作 TXS6。

13.3 串行接口 UART6 的控制寄存器

通过以下 9 种寄存器控制串行接口 UART6。

- 异步串行接口运行模式寄存器 6 (ASIM6)
- 异步串行接口接收错误状态寄存器 6 (ASIS6)
- 异步串行接口发送状态寄存器 6 (ASIF6)
- 时钟选择寄存器 6 (CKSR6)
- 波特率发生器控制寄存器 6 (BRGC6)
- 异步串行接口控制寄存器 6 (ASICL6)
- 输入切换控制寄存器 (ISC)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

(1) 异步串行接口运行模式寄存器 6 (ASIM6)

异步串行接口运行模式寄存器 6 是 8 位寄存器，用于控制串行接口 UART6 的串行通信运行。

通过 1 位或 8 位存储器操作指令设置 ASIM6。

在产生复位信号后，ASIM6 变为“01H”。

备注 在通信运行过程中（异步串行接口运行模式寄存器 6 (ASIM6) 的 bit7、bit6 (POWER6、TXE6) 都为“1”或者 ASIM6 的 bit7、bit5 (POWER6、RXE6) 都为“1”)，能通过软件执行 TXB6 的刷新（写相同的值）运行。

图 13-5 异步串行接口运行模式寄存器 6 (ASIM6) 的格式 (1/2)

地址: FF50H 复位后: 01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	允许 / 禁止内部运行时钟的运行
0 注 1	禁止内部运行时钟的运行（固定为低电平），并且异步复位内部电路注 2。
1	允许内部运行时钟的运行

TXE6	允许 / 禁止发送运行
0	禁止发送运行（同步复位发送电路）
1	允许发送运行

RXE6	允许 / 禁止接收运行
0	禁止接收运行（同步复位接收电路）
1	允许接收运行

注 1. 通过将 POWER6 置“0”，TxD6 引脚的输出被固定为高电平（TXDLV6=0 时）。另外，RxD6 引脚的输入也被固定为高电平。

2. 异步串行接口接收错误状态寄存器 6 (ASIS6)、异步串行接口发送状态寄存器 6 (ASIF6)、异步串行接口控制寄存器 6 (ASICL6) 的 bit7 (SBRF6) 和 bit6 (SBRT6) 以及接收缓冲寄存器 6 (RXB6) 发生复位。

图 13-5 异步串行接口运行模式寄存器 6 (ASIM6) 的格式 (2/2)

PS61	PS60	发送运行	接收运行
0	0	不输出奇偶校验位	无奇偶校验的接收
0	1	输出零校验	作为零校验接收 ^注
1	0	输出奇校验	作为奇校验进行判断
1	1	输出偶校验	作为偶校验进行判断

CL6	指定发送 / 接收数据的字符长度
0	数据的字符长度 =7 位
1	数据的字符长度 =8 位

SL6	指定发送数据的停止位个数
0	停止位个数 =1
1	停止位个数 =2

ISRM6	在发生错误时允许 / 禁止发生接收结束中断
0	在发生错误时发生中断“INTSRE6”（此时不发生 INTSR6）
1	在发生错误时发生中断“INTSR6”（此时不发生 INTSRE6）

注 如果设置为“作为零校验接收”，就不进行奇偶校验的判断。因此，异步串行接口接收错误状态寄存器 6 (ASIS6) 的 bit2 (PE6) 不被置位，从而也不发生错误中断。

- 注意 1. 将 POWER6 置“1”，再将 TXE6 置“1”，开始发送。将 TXE6 置“0”，再将 POWER6 置“0”，停止发送。
2. 将 POWER6 置“1”，再将 RXE6 置“1”，开始接收。将 RXE6 置“0”，再将 POWER6 置“0”，停止接收。
3. 在将高电平输入 RxD6 引脚的状态下将置 POWER6 置“1”，再将 RXE6 置“1”。如果在低电平的状态下将 POWER6 置“1”，再将 RXE6 置“1”，就开始接收。
4. 通过 CKSR6 设置的基本时钟 (f_{XCLK6}) 同步 TXE6 与 RXE6。为了允许重新发送或者接收，必须在将 TXE6 或者 RXE6 清“0”后，至少等待 2 个基本时钟，再将 TXE6 或者 RXE6 置“1”。如果在 2 个基本时钟内将 TXE6 或者 RXE6 置“1”，就有可能无法对发送电路或者接收电路进行初始化。
5. 在将 TXE6 置“1”后，至少等待 1 个基本时钟 (f_{XCLK6})，才能将发送数据设置到 TXB6。
6. 在将 TXE6 位和 RXE6 位清“0”后，才能改写 PS61 位、PS60 位和 CL6 位。
7. 在 LIN 通信运行下使用时，必须将 PS61、PS60 位固定为“0”。
8. 在将 TXE6 位清“0”后，才能改写 SL6 位。另外，因为只检查接收数据停止位中的第 1 个停止位，所以不受 SL6 位的设置值的影响。
9. 在将 RXE6 位清“0”后，才能改写 ISRM6 位。

(2) 异步串行接口接收错误状态寄存器 6 (ASIS6)

异步串行接口接收错误状态寄存器 6 表示串行接口 UART6 的接收结束时的错误状态。由 3 位错误标志 (PE6、FE6、OVE6) 构成。

只能通过 8 位存储器操作指令读 ASIS6。

在产生复位信号后并且 ASIM6 的 bit7 (POWER6) =0、bit5 (RXE6) =0, ASIS6 变为“00H”。另外, 读取值也为“00H”。如果发生接收错误, 就在读 ASIS6 后读取接收缓冲寄存器 6 (RXB6), 并且清除错误标志。

图 13-6 异步串行接口接收错误状态寄存器 6 (ASIS6) 的格式

地址: FF53H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIS6	0	0	0	0	0	PE6	FE6	OVE6

PE6	表示奇偶校验错误的状态标志
0	在将 POWER6 和 RXE6 置“0”或者读 ASIS6 寄存器时
1	接收结束后发送数据的奇偶校验与奇偶校验位不匹配时

FE6	表示帧错误的状态标志
0	在将 POWER6 和 RXE6 置“0”或者读 ASIS6 寄存器时
1	接收结束后未检测出停止位时

OVE6	表示溢出错误的状态标志
0	在将 POWER6 和 RXE6 置“0”或者读 ASIS6 寄存器时
1	将接收数据设置到 RXB6 寄存器并且在读取前结束下一个接收运行时

- 注意 1. PE6 位的运行因异步串行接口运行模式寄存器 6 (ASIM6) 的 PS61 位和 PS60 位的设置值而不同。
- 接收数据的停止位与停止位数无关只检查第 1 个位。
 - 在发生溢出错误时, 下一个接收数据不被写入接收缓冲寄存器 6 (RXB6), 而被舍弃。
 - 如果从 ASIS6 读取数据, 就产生等待。在外围硬件时钟 (f_{PRS}) 停止时, 不能从 ASIS6 读取数据。详细内容请参照“第 27 章 等待的注意事项”。

(3) 异步串行接口发送状态寄存器 6 (ASIF6)

异步串行接口发送状态寄存器 6 表示串行接口 UART6 的发送时的状态。由 2 位状态标志 (TXBF6、TXSF6) 构成。

在将数据从 TXB6 寄存器传送到 TXS6 寄存器后, 可通过给 TXB6 寄存器写下一个数据, 持续执行发送, 即使在中断期间也不会间断。

只能通过 8 位存储器操作指令读 ASIF6。

在产生复位信号后并且 ASIM6 的 bit7 (POWER6) =0、bit6 (TXE6) =0, ASIF6 变为“00H”。

图 13-7 异步串行接口发送状态寄存器 6 (ASIF6) 的格式

地址: FF55H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIF6	0	0	0	0	0	0	TXBF6	TXSF6

TXBF6	发送缓冲器的数据标志
0	在将 POWER6 或者 TXE6 置“0”, 或者向发送移位寄存器 6 (TXS6) 传送数据时
1	在给发送缓冲寄存器 6 (TXB6) 写数据时 (TXB6 中有数据时)

TXSF6	发送移位寄存器的数据标志
0	在将 POWER6 或者 TXE6 置“0”, 或者数据传送后没有从发送缓冲寄存器 6 (TXB6) 传送下一个数据时
1	从发送缓冲寄存器 6 (TXB6) 传送来数据时 (数据传送中)

- 注意 1. 在进行连续发送时, 必须在将第 1 个发送数据 (第 1 字节) 写入 TXB6 寄存器后确认 TXBF6 标志为“0”, 之后才能将下一个发送数据 (第 2 字节) 写入 TXB6。如果在 TXBF6 标志为“1”时将数据写入 TXB6 寄存器, 就不保证发送数据。
2. 如果要在连续发送结束时初始化发送单元, 必须在发生发送结束中断后确认 TXSF6 标志为“0”, 之后再行初始化。如果在 TXSF6 标志为“1”时进行初始化, 就不保证发送数据。

(4) 时钟选择寄存器 6 (CKSR6)

时钟选择寄存器 6 选择串行接口 UART6 的基本时钟。

通过 8 位存储器操作指令设置 CKSR6。

在产生复位信号后，CKSR6 变为“00H”。

备注 在通信运行过程中（异步串行接口运行模式寄存器 6 (ASIM6) 的 bit7、bit6 (POWER6、TXE6) 都为“1”或者 ASIM6 的 bit7、bit5 (POWER6、RXE6) 都为“1”)，能通过软件执行 CKSR6 的刷新（写相同的值）运行。

图 13-8 时钟选择寄存器 6 (CKSR6) 的格式

地址: FF56H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60

TPS63	TPS62	TPS61	TPS60		选择基本时钟 (f_{XCLK6})		
					$f_{PRS}=2\text{MHz}$	$f_{PRS}=5\text{MHz}$	$f_{PRS}=10\text{MHz}$
0	0	0	0	f_{PRS}	2MHz	5MHz	10MHz
0	0	0	1	$f_{PRS}/2$	1MHz	2.5MHz	5MHz
0	0	1	0	$f_{PRS}/2^2$	500kHz	1.25MHz	2.5MHz
0	0	1	1	$f_{PRS}/2^3$	250kHz	625kHz	1.25MHz
0	1	0	0	$f_{PRS}/2^4$	125kHz	312.5kHz	625kHz
0	1	0	1	$f_{PRS}/2^5$	62.5kHz	156.25kHz	312.5kHz
0	1	1	0	$f_{PRS}/2^6$	31.25kHz	78.13kHz	156.25kHz
0	1	1	1	$f_{PRS}/2^7$	15.625kHz	39.06kHz	78.13kHz
1	0	0	0	$f_{PRS}/2^8$	7.813kHz	19.53kHz	39.06kHz
1	0	0	1	$f_{PRS}/2^9$	3.906kHz	9.77kHz	19.53kHz
1	0	1	0	$f_{PRS}/2^{10}$	1.953kHz	4.88kHz	9.77kHz
1	0	1	1	TM50 的输出注			
其他				禁止设置			

注 选择 TM50 输出作为基本时钟时，必须注意以下内容。

- TM50 与 CR50 匹配时的清除 & 启动模式 (TMC506=0)
允许定时器 F/F 的取反运行 (TMC501=1)，事先开始 8 位定时器 / 事件计数器 50 的运行。
- PWM 模式 (TMC506=1)
设置占空比为 50% 的时钟，事先开始 8 位定时器 / 事件计数器 50 的运行。
在其中任一模式中，都不需要允许 TO50 输出 (TOE50=1)。

注意 在将 POWER6 置“0”后，才能改写 TPS63 ~ TPS60。

备注 1. f_{PRS} : 外围硬件时钟频率

2. TMC506: 8 位定时器模式控制寄存器 50 (TMC50) 的 bit6
TMC501: TMC50 的 bit1

(5) 波特率发生器控制寄存器 6 (BRGC6)

波特率发生器控制寄存器 6 设置串行接口 UART6 的 8 位计数器的分频值。

通过 8 位存储器操作指令设置 BRGC6。

在产生复位信号后，BRGC6 变为“FFH”。

备注 在通信运行过程中（异步串行接口运行模式寄存器 6 (ASIM6) 的 bit7、bit6 (POWER6、TXE6) 都为“1”或者 ASIM6 的 bit7、bit5 (POWER6、RXE6) 都为“1”)，能通过软件执行 BRGC6 的刷新（写相同的值）运行。

图 13-9 波特率发生器控制寄存器 6 (BRGC6) 的格式

地址: FF57H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
BRGC6	MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	k	选择 8 位计数器的输出时钟
0	0	0	0	0	0	x	x	x	禁止设置
0	0	0	0	0	1	0	0	4	$f_{XCLK6}/4$
0	0	0	0	0	1	0	1	5	$f_{XCLK6}/5$
0	0	0	1	0	1	1	0	6	$f_{XCLK6}/6$
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	$f_{XCLK6}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK6}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK6}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK6}/255$

注意 1. 在将 ASIM6 寄存器的 bit6 (TXE6) 和 bit5 (RXE6) 置“0”后，才能改写 MDL67 ~ MDL60 位。

2. 对 8 位计数器的输出时钟进行 1/2 分频后的值为波特率值。

备注 1. f_{XCLK6} : 通过 CKSR6 寄存器的 TPS63 ~ TPS60 位选择的基本时钟的频率

2. k: 通过 MDL67 ~ MDL60 位设置的值 (k = 4、5、6、...255)

3. x: 任意

(6) 异步串行接口控制寄存器 6 (ASICL6)

异步串行接口控制寄存器 6 控制串行接口 UART6 的串行通信运行。

能通过 1 位或者 8 位存储器操作指令设置 ASICL6。

在产生复位信号后，ASICL6 变为“16H”。

备注 在通信运行过程中（异步串行接口运行模式寄存器 6 (ASIM6) 的 bit7、bit6 (POWER6、TXE6) 都为“1”或者 ASIM6 的 bit7、bit5 (POWER6、RXE6) 都为“1”)，能通过软件执行 ASICL6 的刷新（写相同的值）运行。但是，在 SBF 接收过程 (SBRF6=1) 或者 SBF 发送过程（从 SBTT6 置“1”后到产生 INTST6 之间）中，如果通过刷新运行将 SBRT6 和 SBTT6 置“1”，就成为重新进行 SBF 发送、SBF 接收的触发源，因此不能进行设置。

图 13-10 异步串行接口控制寄存器 6 (ASICL6) 的格式 (1/2)

地址：FF58H 复位后：16H R/W注

符号	<7>	<6>	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6

SBRF6	SBF 接收状态标志
0	将 POWER 和 RXE6 置“0”或者 SBF 接收正常结束时
1	正在进行 SBF 接收

SBRT6	SBF 接收触发
0	—
1	SBF 接收触发

SBTT6	SBF 发送触发
0	—
1	SBF 发送触发

注 bit7 为只读位。

图 13-10 异步串行接口控制寄存器 6 (ASICL6) 的格式 (2/2)

SBL62	SBL61	SBL60	控制 SBF 发送的输出宽度
1	0	1	SBF 以 13 位输出
1	1	0	SBF 以 14 位输出
1	1	1	SBF 以 15 位输出
0	0	0	SBF 以 16 位输出
0	0	1	SBF 以 17 位输出
0	1	0	SBF 以 18 位输出
0	1	1	SBF 以 19 位输出
1	0	0	SBF 以 20 位输出

DIR6	指定起始位
0	MSB
1	LSB

TXDLV6	允许 / 禁止 TxD6 输出取反
0	TxD6 正常输出
1	TxD6 取反输出

- 注意 1. 发生 SBF 接收错误时, 重新返回 SBF 接收模式。SBRF6 标志保持为“1”的状态。
- 在将 ASIM6 的 bit7 (POWER6) 和 bit5 (RXE6) 置“1”后, 才能将 SBRT6 位置“1”。另外, 在 SBRT6 位置“1”后到 SBF 接收结束 (产生中断请求信号) 前, 不能将 SBRT6 位清“0”。
 - SBRT6 位的读取值总是为“0”。在 SBF 接收正常结束后, SBRT6 位自动清“0”。
 - 在将 ASIM6 的 bit7 (POWER6) 和 bit6 (TXE6) 置“1”后, 才能将 SBTT6 位置“1”。另外, 在 SBTT6 位置“1”后到 SBF 发送结束 (产生中断请求信号) 前, 不能将 SBTT6 位清“0”。
 - SBTT6 位的读取值总是为“0”。在 SBF 发送结束后, SBTT6 位自动清“0”。
 - 在接收运行过程中, 不能将 SBRT6 位置“1”; 在发送运行过程中, 不能将 SBTT6 位置“1”。
 - 在将 TXE6、RXE6 位清“0”后, 才能改写 DIR6、TXDLV6 位。
 - 在 TXDLV6 位为“1” (TxD6 取反输出) 时, 与 POWER6、TXE6 的设置无关, TxD6/P13 引脚不能用作通用端口。如果要将 TxD6/P13 引脚用作通用端口, 必须将 TXDLV6 位置“0” (TxD6 正常输出)。

(7) 输入切换控制寄存器 (ISC)

在 LIN (Local Interconnect Network) 接收时, 输入切换控制寄存器 (ISC) 用于接收从主控设备发送来的状态信号。

通过将 ISC0、ISC1 置“1”, INTP0 和 TI000 的输入源切换为 P14/RxD6 引脚的输入信号。

能通过 1 位或 8 位存储器操作指令设置 ISC。

在产生复位信号后, ISC 变为“00H”。

图 13-11 输入切换控制寄存器 (ISC) 的格式

地址: FF4FH 复位后: 00H R/W

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

ISC1	选择 TI000 的输入源
0	TI000 (P00)
1	RxD6 (P14)

ISC0	选择 INTP0 的输入源
0	INTP0 (P120)
1	RxD6 (P14)

(8) 端口模式寄存器 1 (PM1)

端口模式寄存器 1 以 1 位单位设置端口 1 的输入 / 输出。

将 P13/TxD6 引脚用作串行接口的数据输出时, 必须将 PM13 置“0”, 将 P13 的输出锁存器置“1”。

将 P14/RxD6 引脚用作串行接口的数据输入时, 必须将 PM14 置“1”。此时, P14 的输出锁存器既可是“0”也可是“1”。

通过 1 位或 8 位存储器操作指令设置 PM1。

在产生复位信号后, PM1 变为“FFH”。

图 13-12 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 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

PM1n	选择 P1n 引脚的输入 / 输出模式 (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)

13.4 串行接口 UART6 的运行

串行接口 UART6 有以下两种模式。

- 运行停止模式
- 异步串行接口（UART）模式

13.4.1 运行停止模式

在运行停止模式中，不执行串行通信，从而可降低功耗。此外，引脚通常可用作通用端口。将 ASIM6 的 bit7 ~ bit5（POWER6、TXE6 和 RXE6）置“0”，设置为运行停止模式。

(1) 使用的寄存器

通过异步串行接口运行模式寄存器 6（ASIM6）设置运行停止模式。

通过 1 位或 8 位存储器操作指令设置 ASIM6。

在产生复位信号后，ASIM6 变为“01H”。

地址：FF50H 复位后：01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	允许 / 禁止内部运行时钟的运行
0 注 1	禁止内部运行时钟的运行（固定为低电平），并且异步复位内部电路注 2。

TXE6	允许 / 禁止发送运行
0	禁止发送运行（同步复位发送电路）

RXE6	允许 / 禁止接收运行
0	禁止接收运行（同步复位接收电路）

- 注 1. 通过将 POWER6 置“0”，TxD6 引脚的输出被固定为高电平（TXDLV6=0 时）。另外，RxD6 引脚的输入也被固定为高电平。
2. 异步串行接口接收错误状态寄存器 6（ASIS6）、异步串行接口发送状态寄存器 6（ASIF6）、异步串行接口控制寄存器 6（ASICL6）的 bit7（SBRF6）和 bit6（SBRT6）以及接收缓冲寄存器 6（RXB6）发生复位。

注意 将 TXE6 和 RXE6 置“0”，再将 POWER6 置“0”，停止运行。
将 POWER6 置“1”，再将 TXE6 和 RXE6 置“1”，开始通信。

备注 将 RxD6/P14 和 TxD6/P13 引脚用作通用端口时，参照“第 4 章 端口功能”。

13.4.2 异步串行接口（UART）模式

异步串行接口（UART）模式是发送 / 接收紧接着起始位的 1 字节数据的模式。可执行全双工运行。因为内置 UART 专用波特率发生器，所以能以广范围的波特率进行通信。

(1) 使用的寄存器

- 异步串行接口运行模式寄存器 6（ASIM6）
- 异步串行接口接收错误状态寄存器 6（ASIS6）
- 异步串行接口发送状态寄存器 6（ASIF6）
- 时钟选择寄存器 6（CKSR6）
- 波特率发生器控制寄存器 6（BRGC6）
- 异步串行接口控制寄存器 6（ASICL6）
- 输入切换控制寄存器（ISC）
- 端口模式寄存器 1（PM1）
- 端口寄存器 1（P1）

UART 模式的基本运行设置步骤示例如下所示。

1. 设置 CKSR6 寄存器（参照图 13-8）
2. 设置 BRGC6 寄存器（参照图 13-9）
3. 设置 ASIM6 寄存器的 bit0 ~ bit4（ISRM6、SL6、CL6、PS60、PS61）（参照图 13-5）
4. 设置 ASICL6 寄存器的 bit0、bit1（TXDLV6、DIR6）（参照图 13-10）
5. 将 ASIM6 寄存器的 bit7（POWER6）置“1”
6. 将 ASIM6 寄存器 bit6（TXE6）置“1”→可发送
将 ASIM6 寄存器的 bit5（RXE6）置“1”→可接收
7. 将数据写入发送缓冲寄存器 6（TXB6）→开始数据发送

注意 必须在考虑与通信方的关系后，设置端口模式寄存器和端口寄存器。

寄存器设置与引脚的关系如下所示。

表 13-2 寄存器设置与引脚的关系

POWER6	TXE6	RXE6	PM13	P13	PM14	P14	UART6 的运行	引脚功能	
								TxD6/P13	RxD6/P14
0	0	0	x 注	x 注	x 注	x 注	停止	P13	P14
1	0	1	x 注	x 注	1	x	接收	P13	RxD6
	1	0	0	1	x 注	x 注	发送	TxD6	P14
	1	1	0	1	1	x	发送 / 接收	TxD6	RxD6

注 可作为端口功能设置。

备注 x: 忽略
 POWER6: 异步串行接口运行模式寄存器 6（ASIM6）的 bit7
 TXE6: ASIM6 的 bit6
 RXE6: ASIM6 的 bit5
 PM1x: 端口模式寄存器
 P1x: 端口的输出锁存器

(2) 通信运行

(a) 正常发送 / 接收数据的格式和波形示例

正常发送 / 接收数据的格式和波形示例如图 13-13 和图 13-14 所示。

图 13-13 正常 UART 发送 / 接收数据的格式

1. LSB 优先的情况



2. MSB 优先的情况



1 个数据帧由以下各位组成。

- 起始位 ...1 位
- 字符位 ...7 位/8 位
- 奇偶校验位 ... 偶校验/奇校验/零校验/无奇偶校验
- 停止位 ...1 位/2 位

通过异步串行接口运行模式寄存器 6 (ASIM6) 指定 1 个数据帧中的字符位长度、停止位长度以及选择奇偶校验。

通过异步串行接口控制寄存器 6 (ASICL6) 的 bit1 (DIR6) 设置 LSB 优先 / MSB 优先后进行数据通信。另外，通过 ASICL6 的 bit0 (TXDLV6) 设置 TxD6 引脚的正常输出 / 取反输出。

图 13-14 正常 UART 发送 / 接收数据的波形示例

1. 数据长度：8 位、LSB 优先、奇偶校验：偶校验、停止位：1 位、通信数据：55H



2. 数据长度：8 位、MSB 优先、奇偶校验：偶校验、停止位：1 位、通信数据：55H



3. 数据长度：8 位、MSB 优先、奇偶校验：偶校验、停止位：1 位、通信数据：55H、TxD6 引脚取反输出



4. 数据长度：7 位、LSB 优先、奇偶校验：奇校验、停止位：2 位、通信数据：36H



5. 数据长度：8 位、LSB 优先、奇偶校验：无奇偶校验、停止位：1 位、通信数据：87H



(b) 奇偶校验的种类和运行

奇偶校验位检测通信数据的位错误。通常，发送端和接收端使用相同种类的奇偶校验位。偶校验和奇校验都可检测出 1 位（奇数）错误。在零校验和无奇偶校验时，不能检测出错误。

注意 在 LIN 通信运行下使用时，PS61、PS60 位必须固定为“0”。

(i) 偶校验

• 发送时

将包括奇偶校验位的发送数据中值为“1”的位个数控制为偶数。
奇偶校验位的值如下所示。

发送数据中值为“1”的位个数为奇数：1
发送数据中值为“1”的位个数为偶数：0

• 接收时

对包括奇偶校验位的接收数据中值为“1”的位个数进行计数，如果为奇数，就发生奇偶校验错误。

(ii) 奇校验

• 发送时

与偶校验相反，将包括奇偶校验位的发送数据中值为“1”的位个数控制为奇数。
奇偶校验位的值如下所示。

发送数据中值为“1”的位个数为奇数：0
发送数据中值为“1”的位个数为偶数：1

• 接收时

对包括奇偶校验位的接收数据中值为“1”的位个数进行计数，如果为偶数，就发生奇偶校验错误。

(iii) 零校验

发送期间，与发送数据无关，将奇偶校验位置“0”。

接收期间，不进行奇偶校验位的检测。因此，无论校验位为“0”或者“1”，都不发生奇偶校验错误。

(iv) 无奇偶校验

发送数据中无奇偶校验位。

接收期间，以无奇偶位进行接收。因为没有奇偶校验位，所以不发生奇偶校验错误。

(c) 正常发送

通过将异步串行接口运行模式寄存器 6（ASIM6）的 bit7（POWER6）置“1”、将 ASIM6 的 bit6（TXE6）置“1”，进入允许发送状态，并且通过将发送数据写入发送缓冲寄存器 6（TXB6），启动发送运行。自动附加起始位、奇偶校验位以及停止位。

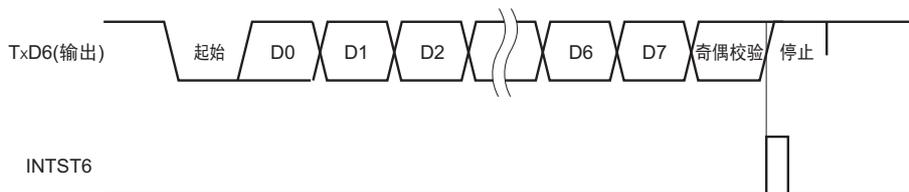
开始发送运行后，TXB6 中的数据被传送到发送移位寄存器 6（TXS6）。之后，发送数据从 TXS6 开始依次输出到 TxD6 引脚。一旦发送结束，就附加由 ASIM6 设置的奇偶校验位和停止位，最后产生发送结束中断请求（INTST6）。

直到下一个发送数据写入 TXB6，才能中止发送运行。

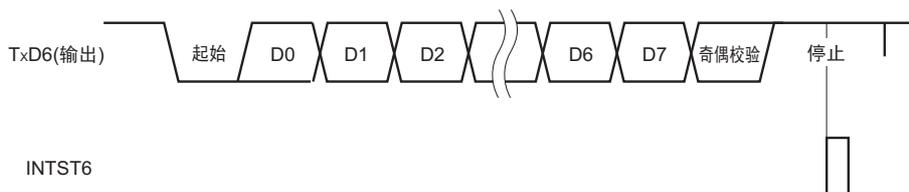
产生发送结束中断请求（INTST6）的时序如图 13-15 所示。在输出最后的停止位的同时产生 INTST6。

图 13-15 正常发送结束中断请求时序

1. 停止位长度：1



2. 停止位长度：2



(d) 连续发送

发送移位寄存器 6 (TXS6) 在移位运行开始时, 可将下一个发送数据写入发送缓冲寄存器 6 (TXB6)。由此, 即使在发送 1 个数据帧后的 INTST6 中断处理过程中也可进行连续发送, 从而实现高效通信速率。另外, 在发生发送结束中断后, 通过读异步串行接口发送状态寄存器 6 (ASIF6) 的 bit0 (TXSF6), 无需等待 1 个数据帧的发送时间, 就能高效写 2 次 (2 字节) TXB6 寄存器。

在进行连续发送时, 必须在参照 ASIF6 寄存器, 确认发送状态和是否可写 TXB6 寄存器后, 才能写数据。

注意 1. 在连续发送时, ASIF6 寄存器的 TXBF6、TXSF6 标志按照“10”→“11”→“01”的顺序变化。因此, 在确认状态时, 不能通过 TXBF6、TXSF6 标志的组合进行判断。在执行连续发送时, 只能读 TXBF6 标志。

2. 在 LIN 通信运行下使用时, 不能使用连续发送功能。必须在确认异步串行接口发送状态寄存器 6 (ASIF6) 为“00H”后, 将发送数据写入发送缓冲寄存器 6 (TXB6)。

TXBF6	是否可写 TXB6 寄存器
0	可写
1	不可写

注意 在进行连续发送时, 必须在将第 1 个发送数据 (第 1 字节) 写入 TXB6 寄存器后确认 TXBF6 标志为“0”, 之后才能将下一个发送数据 (第 2 字节) 写入 TXB6。如果在 TXBF6 标志为“1”时将数据写入 TXB6 寄存器, 就不保证发送数据。

可通过 TXSF6 标志确认通信状态。

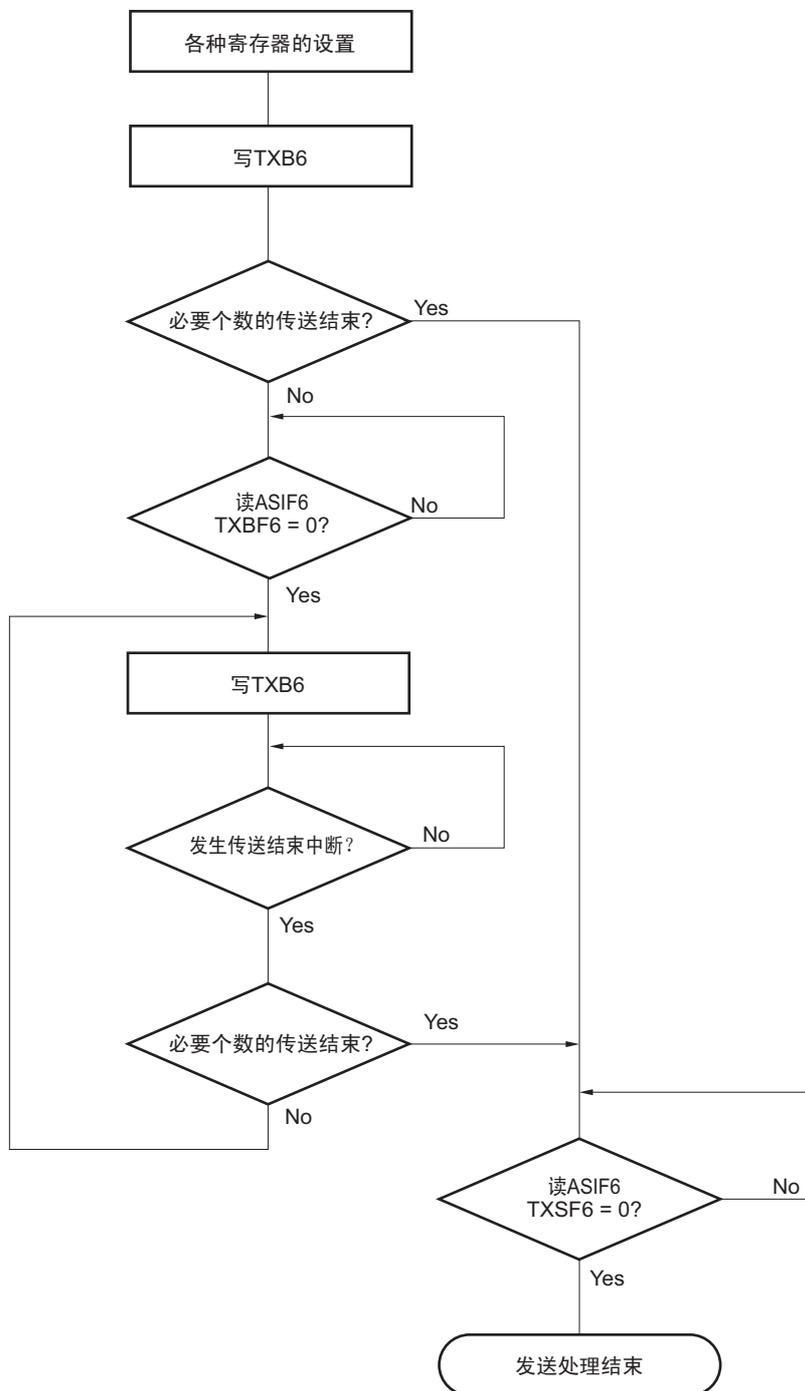
TXSF6	发送状态
0	发送结束
1	正在发送

注意 1. 如果要在连续发送结束时初始化发送单元, 必须在发生发送结束中断后确认 TXSF6 标志为“0”, 之后再行初始化。如果在 TXSF6 标志为“1”时进行初始化, 就不保证发送数据。

2. 连续发送时, 在执行发送 1 个数据帧后的 INTST6 中断处理前, 下一个发送有可能已结束。可通过编程对发送数据进行计数以及参照 TXSF6 标志等措施进行检测。

连续发送的处理流程示例如图 13-16 所示。

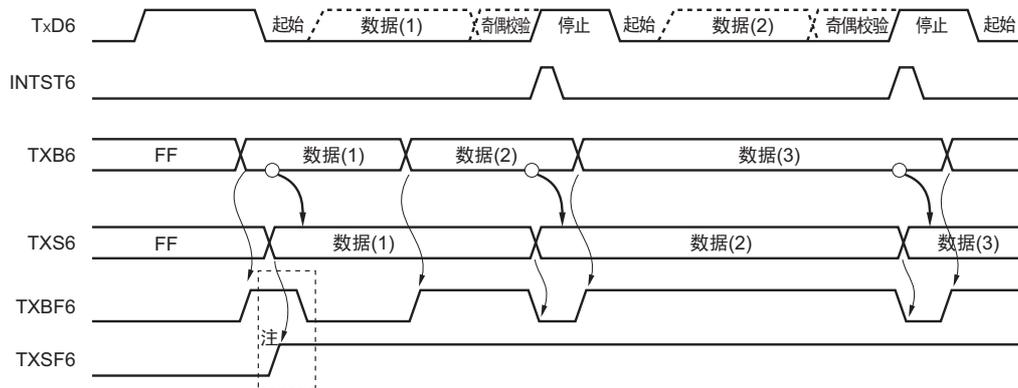
图 13-16 连续发送的处理流程示例



- 备注 TXB6: 发送缓冲寄存器 6
 ASIF6: 异步串行接口发送状态寄存器 6
 TXBF6: ASIF6 的 bit1 (发送缓冲器的数据标志)
 TXSF6: ASIF6 的 bit0 (发送移位寄存器的数据标志)

开始连续发送时的时序如图 13-17 所示。结束连续发送时的时序如图 13-18 所示。

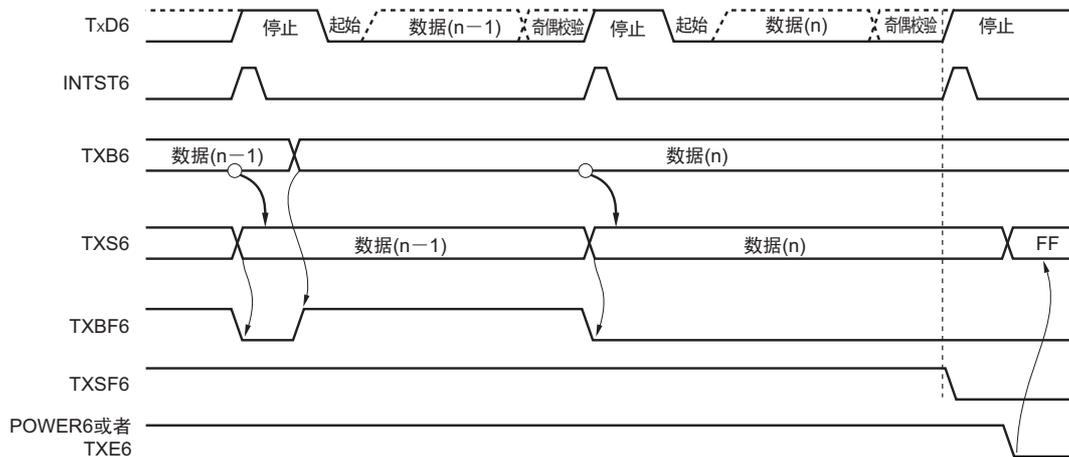
图 13-17 开始连续发送时的时序



注 在读 ASIF6 时，存在 TXBF6、TXSF6 都为“1”的期间。因此，只能通过 TXBF6 位判断是否可写。

- 备注 TxD6: TxD6 引脚（输出）
 INTST6: 中断请求信号
 TXB6: 发送缓冲寄存器 6
 TXS6: 发送移位寄存器 6
 ASIF6: 异步串行接口发送状态寄存器 6
 TXBF6: ASIF6 的 bit1
 TXSF6: ASIF6 的 bit0

图 13-18 结束连续发送时的时序



- 备注 TxD6: TxD6 引脚（输出）
 INTST6: 中断请求信号
 TXB6: 发送缓冲寄存器 6
 TXS6: 发送移位寄存器 6
 ASIF6: 异步串行接口发送状态寄存器 6
 TXBF6: ASIF6 的 bit1
 TXSF6: ASIF6 的 bit0
 POWER6: 异步串行接口运行模式寄存器 6（ASIM6）的 bit7
 TXE6: 异步串行接口运行模式寄存器 6（ASIM6）的 bit6

(e) 正常接收

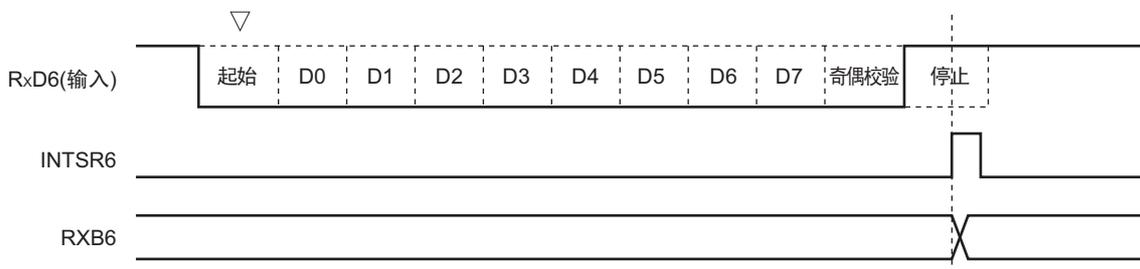
如果将异步串行接口运行模式寄存器 6 (ASIM6) 的 bit7 (POWER6) 置“1”，再将 ASIM6 的 bit5 (RXE6) 置“1”，就进入允许接收状态，并且进行 RxD6 引脚输入的采样。

如果检测出 RxD6 引脚输入的下沿，波特率发生器的 8 位计数器就开始计数，在对波特率发生器控制寄存器 6 (BRGC6) 的设置值进行计数后，再次进行 RxD6 引脚输入的采样 (如图 13-19 中的“▽”)。如果采样后的 RxD6 引脚为低电平，就识别为起始位。

在检测出起始位后，开始接收运行，并且根据设置的波特率，将串行数据依次保存到接收移位寄存器 6 (RXS6)。一旦接收到停止位，就在发生接收结束中断 (INTSR6) 的同时，将 RXS6 的数据写入接收缓冲寄存器 6 (RXB6)。但是，在发生溢出错误 (OVE6) 时，不能将接收的数据写入 RXB6。

即使在接收过程中发生奇偶校验错误 (PE6)，也继续接收，直到停止位的接收位置，并且在接收结束后发生接收错误中断 (INTSR6/INTSRE6)。

图 13-19 接收结束中断请求时序



- 注意 1. 如果发生接收错误，在读 ASIS6 后，读 RXB6，并且清除错误标志。如果不读 RXB6，就在接收下一个数据时发生溢出错误，而且将一直持续接收错误的状态。
2. 接收运行总是以“停止位个数=1”进行运行。忽略第 2 个停止位。
3. 在读 RXB6 前，必须先读异步串行接口接收错误状态寄存器 6 (ASIS6)。

(f) 接收错误

接收运行期间的错误有 3 种，分别是奇偶校验错误、帧错误和溢出错误。如果数据接收的结果错误标志在异步串行接口接收错误状态寄存器 6 (ASIS6) 中置位，就发生接收错误中断 (INTSR6/INTSRE6)。

在接收错误中断 (INTSR6/INTSRE6) 处理中，可通过读取 ASIS6 的内容，检测出在接收期间发生了哪种错误 (参照图 13-6)。

通过读取 ASIS6，可将 ASIS6 的内容清“0”。

表 13-3 接收错误源

接收错误	接收错误源
奇偶校验错误	指定的发送时的奇偶校验与接收数据的奇偶校验不匹配
帧错误	未检测出停止位
溢出错误	在从接收缓冲寄存器 6 (RXB6) 读取数据前结束下一个数据的接收

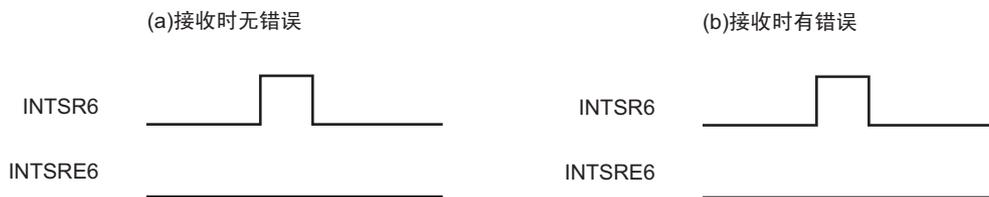
可通过给异步串行接口运行模式寄存器 6 (ASIM6) 的 bit0 (ISRM6) 置“0”，将接收错误中断分离为接收结束中断 (INTSR6) 和错误中断 (INTSRE6)。

图 13-20 接收错误中断

1. 给 ISRM6 置“0”的情况（分离为接收结束中断（INTSR6）和错误中断（INTSRE6））



2. 给 ISRM6 置“1”的情况（INTSR6 包括错误中断）



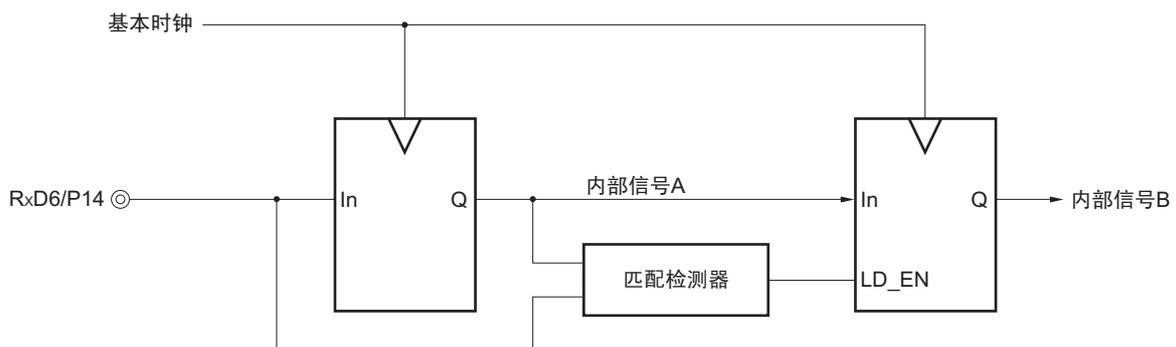
(g) 接收数据的噪声滤波器

接收数据的噪声滤波器以预分频器输出的基本时钟对 RxD6 信号进行采样。

如果对相同的值进行 2 次采样，匹配检测器的输出就发生变化，并且作为输入数据被采样。

在如图 13-21 所示的电路中，接收运行的内部处理比外部的信号状态迟 2 个时钟。

图 13-21 噪声滤波器电路



(h) SBF 发送

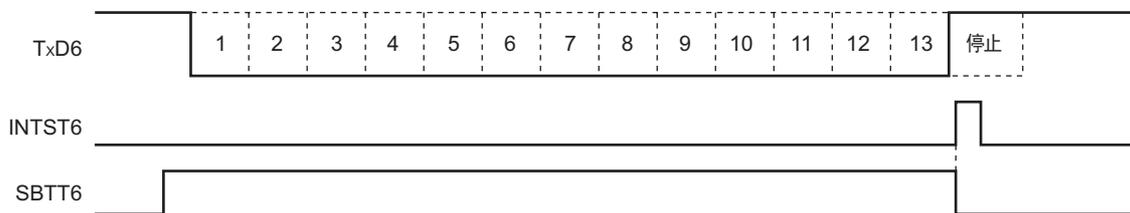
在 LIN 通信运行下使用时，SBF（Synchronous Break Field）发送控制功能用于发送。有关 LIN 的发送操作，请参照“图 13-1 LIN 的发送操作”。

如果将异步串行接口运行模式寄存器 6（ASIM6）的 bit7（POWER6）置“1”，就从 TxD6 引脚输出高电平。接着如果将 ASIM6 的 bit6（TXE6）置“1”，就进入允许发送状态，并且通过将异步串行接口控制寄存器 6（ASICL6）的 bit5（SBTT6）置“1”，启动 SBF 发送运行。

启动后，输出 13 位到 20 位的低电平（通过 ASICL6 的 bit4 ~ bit2（SBL62 ~ SBL60）设置）。如果 SBF 发送结束，就产生发送结束中断请求（INTST6），并且自动清除 SBTT6。结束 SBF 发送后，返回正常发送模式。

中止发送运行，直到将下一个发送数据写入发送缓冲寄存器 6（TXB6）或者将 SBTT6 置“1”。

图 13-22 SBF 发送



备注 TxD6: TxD6 引脚（输出）
INTST6: 发送结束中断请求
SBTT6: 异步串行接口控制寄存器 6（ASICL6）的 bit5

(i) SBF 接收

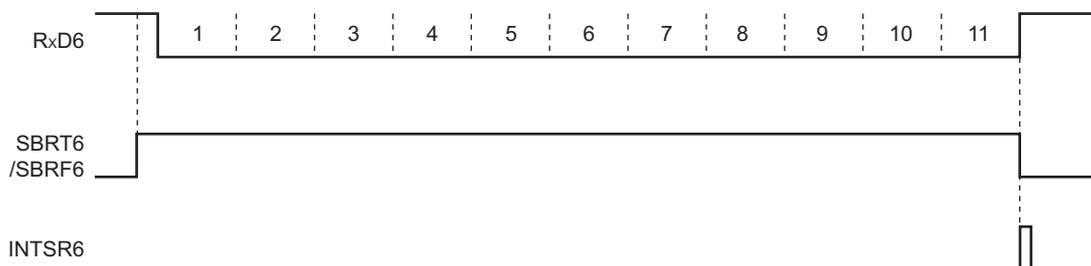
在 LIN 通信运行下使用时，SBF（Synchronous Break Field）接收控制功能用于接收。有关 LIN 的接收操作，请参照“图 13-2 LIN 的接收操作”。

如果将异步串行接口运行模式寄存器 6（ASIM6）的 bit7（POWER6）置“1”，接着将 ASIM6 的 bit5（RXE6）置“1”，就进入允许接收状态，之后如果将异步串行接口控制寄存器 6（ASICL6）的 bit6（SBRT6）置“1”，就进入允许 SBF 接收状态。允许 SBF 接收的状态和通常的接收状态相同，对 RxD6 引脚进行采样，并且执行起始位的检测。

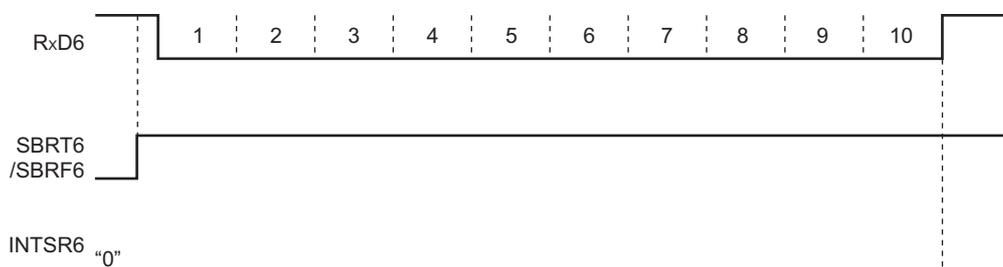
如果检测出起始位，就开始接收运行，根据设置的波特率依次将串行数据保存到接收移位寄存器 6（RXS6）。如果接收到停止位，就在 SBF 的宽度至少为 11 位时，作为正常处理，产生 SBF 接收结束中断请求（INTSR6）。此时，自动清除 SBRF6、SBRT6 位，结束 SBF 接收。OVE6、PE6 和 FE6（异步串行接口接收错误状态寄存器 6（ASIS6）的 bit0 ~ bit2）的各错误检测被抑制，UART 通信的错误检测处理也不执行。而且不执行接收移位寄存器 6（RXS6）和接收缓冲寄存器 6（RXB6）的数据传送，保持复位值“FFH”。在 SBF 的宽度最多为 10 位时，如果接收到停止位，就作为错误处理，在不发生中断下结束接收，并且重新返回 SBF 接收模式。此时，不清除 SBRF6、SBRT6 位。

图 13-23 SBF 接收

1. 正常 SBF 接收（大于 10.5 位并且检测出停止位）



2. SBF 接收错误（小于等于 10.5 位并且检测出停止位）



- 备注 RxD6: RxD6 引脚（输入）
 SBRT6: 异步串行接口控制寄存器 6（ASICL6）的 bit6
 SBRF6: ASICL6 的 bit7
 INTSR6: 接收结束中断请求

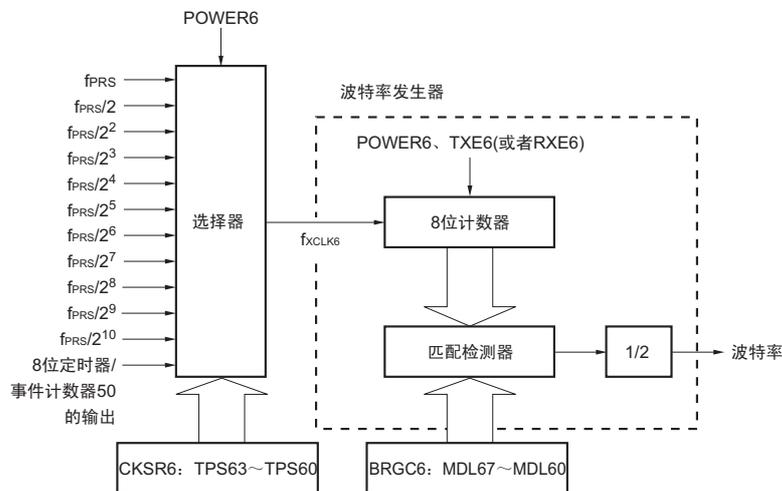
13.4.3 专用波特率发生器

专用波特率发生器由源时钟选择器和 8 位可编程计数器构成，生成 UART6 发送 / 接收时的串行时钟。发送和接收分别使用不同的 8 位计数器。

(1) 波特率发生器的结构

- 基本时钟
在异步串行接口运行模式寄存器 6 (ASIM6) 的 bit7 (POWER6) 为“1”时，向各模块提供由时钟选择寄存器 6 (CKSR6) 的 bit3 ~ bit0 (TPS63 ~ TPS60) 选择的时钟。该时钟被称为基本时钟，其频率频用 f_{XCLK6} 表示。在 POWER6=0 时，基本时钟固定为低电平。
- 发送计数器
在异步串行接口运行模式寄存器 6 (ASIM6) 的 bit7 (POWER6) 或者 bit6 (TXE6) 为“0”时，发送计数器在清“0”的状态下停止运行。
通过将 POWER6 和 TXE6 置“1”，计数器开始计数。
在进行第 1 个发送时，通过将数据写入发送缓冲寄存器 6 (TXB6)，计数器清“0”。
在连续发送时，一旦 1 帧的数据发送结束，计数器就重新被清“0”。如果没有下一个发送数据，计数器就不被清“0”，持续计数运行，直到 POWER6 或者 TXE6 清“0”。
- 接收计数器
在异步串行接口运行模式寄存器 6 (ASIM6) 的 bit7 (POWER6) 或者 bit5 (RXE6) 为“0”时，接收计数器在清“0”的状态下停止运行。
在检测出起始位时，计数器开始计数。
在接收 1 帧的数据后，计数器停止计数，直到检测出下一个起始位。

图 13-24 波特率发生器的结构



备注 POWER6: 异步串行接口运行模式寄存器 6 (ASIM6) 的 bit7

TXE6: ASIM6 的 bit6

RXE6: ASIM6 的 bit5

CKSR6: 时钟选择寄存器 6

BRGC6: 波特率发生器控制寄存器 6

(2) 串行时钟的生成

通过设置时钟选择寄存器 6 (CKSR6) 和波特率发生器控制寄存器 6 (BRGC6) 指定生成的串行时钟。

可通过 CKSR6 的 bit3 ~ bit0 (TPS63 ~ TPS60) 选择 8 位计数器的输入时钟，通过 BRGC6 的 bit7 ~ bit0 (MDL67 ~ MDL60) 设置 8 位计数器的分频值 ($f_{XCLK6}/4 \sim f_{XCLK6}/255$)。

13.4.4 波特率的计算

(1) 波特率的计算公式

可通过以下公式计算波特率。

$$\bullet \text{ 波特率} = \frac{f_{\text{XCLK6}}}{2 \times k} \text{ [bps]}$$

f_{XCLK6} : 通过 CKSR6 寄存器的 TPS63 ~ TPS60 位选择的基本时钟的频率

k : 通过 BRGC6 寄存器的 MDL67 ~ MDL60 位设置的值 ($k = 4、5、6、\dots 255$)

表 13-4 TPS63 ~ TPS60 的设置值

TPS63	TPS62	TPS61	TPS60		选择基本时钟 (f_{XCLK6})		
					$f_{\text{PRS}}=2\text{MHz}$	$f_{\text{PRS}}=5\text{MHz}$	$f_{\text{PRS}}=10\text{MHz}$
0	0	0	0	f_{PRS}	2MHz	5MHz	10MHz
0	0	0	1	$f_{\text{PRS}}/2$	1MHz	2.5MHz	5MHz
0	0	1	0	$f_{\text{PRS}}/2^2$	500kHz	1.25MHz	2.5MHz
0	0	1	1	$f_{\text{PRS}}/2^3$	250kHz	625kHz	1.25MHz
0	1	0	0	$f_{\text{PRS}}/2^4$	125kHz	312.5kHz	625kHz
0	1	0	1	$f_{\text{PRS}}/2^5$	62.5kHz	156.25kHz	312.5kHz
0	1	1	0	$f_{\text{PRS}}/2^6$	31.25kHz	78.13kHz	156.25kHz
0	1	1	1	$f_{\text{PRS}}/2^7$	15.625kHz	39.06kHz	78.13kHz
1	0	0	0	$f_{\text{PRS}}/2^8$	7.813kHz	19.53kHz	39.06kHz
1	0	0	1	$f_{\text{PRS}}/2^9$	3.906kHz	9.77kHz	19.53kHz
1	0	1	0	$f_{\text{PRS}}/2^{10}$	1.953kHz	4.88kHz	9.77kHz
1	0	1	1	TM50 的输出注			
其他				禁止设置			

注 选择 TM50 输出作为基本时钟时的注意事项如下所示。

- TM50 与 CR50 匹配时的清除 & 启动模式 (TMC506=0)
允许定时器 F/F 的取反运行 (TMC501=1)，事先开始 8 位定时器 / 事件计数器 50 的运行。
- PWM 模式 (TMC506=1)
设置占空比为 50% 的时钟，事先开始 8 位定时器 / 事件计数器 50 的运行。
在其中任一模式中，都不需要允许 TO50 输出 (TOE50=1)。

(2) 波特率的误差

可通过以下公式计算波特率的误差。

$$\text{误差 (\%)} = \left[\frac{\text{实际波特率 (有误差的波特率)}}{\text{理想波特率 (正确波特率)}} - 1 \right] \times 100 \text{ [\%]}$$

注意 1. 发送时的波特率误差必须在接收方的允许误差范围内。

2. 接收时的波特率误差必须满足“(4) 接收时的波特率允许范围”中所示的范围。

示例：基本时钟频率 = 10MHz = 10000000Hz
 BRGC6 寄存器的 MDL67 ~ MDL60 位的设置值 = 00100001B (k=33)
 理想波特率 = 153600bps
 波特率 = $10M/(2 \times 33)$
 = $10000000/(2 \times 33) = 151515[\text{bps}]$
 误差 = $(151515/153600 - 1) \times 100$
 = $-1.357[\%]$

(3) 波特率的设置示例

表 13-5 波特率发生器的设置数据

波特率 [bps]	f _{PRS} =2.0 MHz				f _{PRS} =5.0 MHz				f _{PRS} =10.0 MHz			
	TPS63 ~ TPS60	k	计算值	ERR[%]	TPS63 ~ TPS60	k	计算值	ERR[%]	TPS63 ~ TPS60	k	计算值	ERR[%]
300	8H	13	301	0.16	7H	65	301	0.16	8H	65	301	0.16
600	7H	13	601	0.16	6H	65	601	0.16	7H	65	601	0.16
1200	6H	13	1202	0.16	5H	65	1202	0.16	6H	65	1202	0.16
2400	5H	13	2404	0.16	4H	65	2404	0.16	5H	65	2404	0.16
4800	4H	13	4808	0.16	3H	65	4808	0.16	4H	65	4808	0.16
9600	3H	13	9615	0.16	2H	65	9615	0.16	3H	65	9615	0.16
19200	2H	13	19231	0.16	1H	65	19231	0.16	2H	65	19231	0.16
24000	1H	21	23810	-0.79	3H	13	24038	0.16	4H	13	24038	0.16
31250	1H	16	31250	0	4H	5	31250	0	5H	5	31250	0
38400	1H	13	38462	0.16	0H	65	38462	0.16	1H	65	38462	0.16
48000	0H	21	47619	-0.79	2H	13	48077	0.16	3H	13	48077	0.16
76800	0H	13	76923	0.16	0H	33	75758	-1.36	0H	65	76923	0.16
115200	0H	9	111111	-3.55	1H	11	113636	-1.36	0H	43	116279	0.94
153600	—	—	—	—	1H	8	156250	1.73	0H	33	151515	-1.36
312500	—	—	—	—	0H	8	312500	0	1H	8	312500	0
625000	—	—	—	—	0H	4	625000	0	1H	4	625000	0

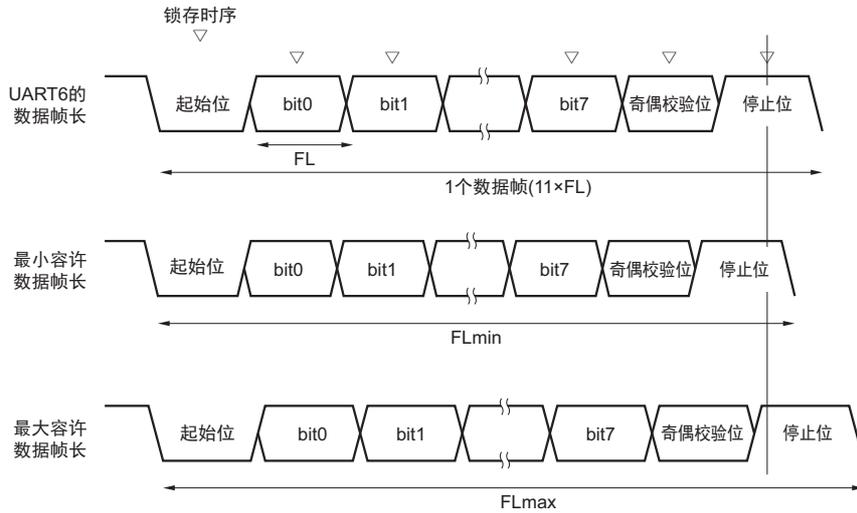
备注 TPS63 ~ TPS60: 时钟选择寄存器 6 (CKSR6) 的 bit3 ~ bit0 (设置基本时钟 (f_{XCLK6}))
 k: 通过波特率发生器控制寄存器 6 (BRGC6) 的 MDL67 ~ MDL60 位设置的值 (k=4、5、6、...255)
 f_{PRS}: 外围硬件时钟频率
 ERR: 波特率误差

(4) 接收时的波特率允许范围

接收期间能容许的发送方的波特率范围如下所示。

注意 通过以下计算式设置接收期间的波特率误差，使其在允许误差范围内。

图 13-25 接收时的波特率允许范围



如图 13-25 所示，在检测出起始位后，通过波特率发生器控制寄存器 6 (BRGC6) 设置的计数器决定接收数据的锁存时序。如果该时序在最后数据 (停止位) 前，就能进行正常接收。

如果将上述内容用于 11 位接收，则理论上可得出以下结果。

$$FL = (\text{Brate})^{-1}$$

Brate: UART6 的波特率

k: BRGC6 的设置值

FL: 1 位数据长度

锁存时序的容限: 2 个时钟

$$\text{最小允许数据帧长: } FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

因此，可接收的发送目标的最大波特率如下所示。

$$BR_{\max} = (FL_{\min}/11)^{-1} = \frac{22k}{21k+2} \text{ Brate}$$

同样，最大允许数据帧长的计算如下所示。

$$\begin{aligned} \frac{10}{11} \times FL_{\max} &= 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL \\ FL_{\max} &= \frac{21k-2}{20k} FL \times 11 \end{aligned}$$

因此，可接收的发送目标的最小波特率如下所示。

$$BR_{\min} = (FL_{\max}/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

从上述的最小 / 最大波特率值的计算式中，求 UART6 和发送目标间的波特率容许误差，如下所示。

表 13-6 最大 / 最小波特率的允许误差

分频比 (k)	最大波特率的允许误差	最小波特率允许误差
4	+2.33%	-2.44%
8	+3.53%	-3.61%
20	+4.26%	-4.31%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.73%

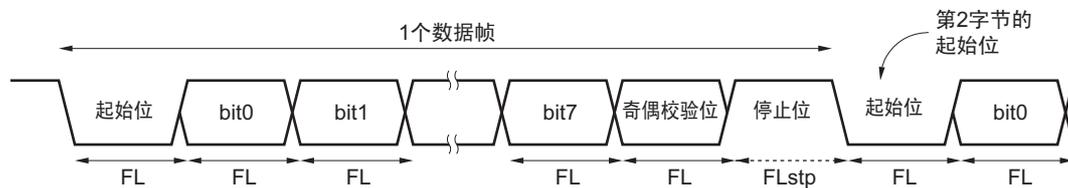
备注 1. 接收时的允许误差取决于每帧的位数、输入时钟频率和分频比 (k)。输入时钟频率和分频比 (k) 越大，允许误差就越大。

2. k: BRGC6 的设置值

(5) 连续发送时的数据帧长

连续发送时，从停止位到下一个停止位的数据帧长比正常时延迟 2 个基本时钟。但是，通过检测起始位，接收方对时序进行初始化，因此不影响通信结果。

图 13-26 连续发送时的数据帧长



如果 1 位数据帧长为 FL、停止位长为 FLstp、基本时钟频率为 f_{XCLK6} ，以下计算式成立。

$$FLstp = FL + 2/f_{XCLK6}$$

由此，连续发送时的数据帧长如下所示。

$$\text{数据帧长} = 11 \times FL + 2/f_{XCLK6}$$

第 14 章 串行接口 CSI10

14.1 串行接口 CSI10 的功能

串行接口 CSI10 有以下两种模式。

(1) 运行停止模式

这是在不执行串行通信时使用的模式。能降低功耗。

详细内容请参照“14.4.1 运行停止模式”。

(2) 3 线式串行 I/O 模式（可进行 MSB/LSB 优先切换）

3 线式串行 I/O 模式通过串行时钟（ $\overline{\text{SCK10}}$ ）和串行数据（SI10、SO10）的 3 条线进行 8 位数据的通信。

3 线式串行 I/O 模式可同时进行发送 / 接收运行，因此能缩短数据通信的处理时间。

因为串行通信的 8 位数据的起始位可切换为 MSB 或者 LSB，所以能与任一起始位的设备连接。

在与内置时钟同步串行接口的外围 IC 或者显示控制器等连接时可使用 3 线式串行 I/O 模式。

详细内容请参照“14.4.2 3 线式串行 I/O 模式”。

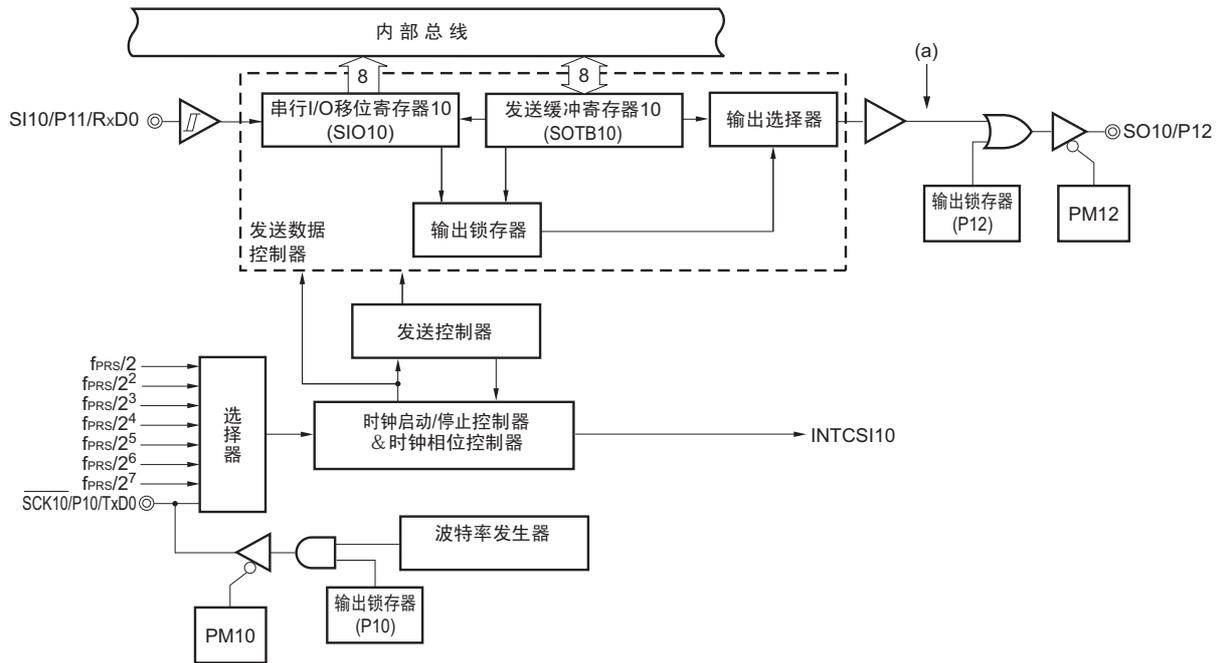
14.2 串行接口 CSI10 的结构

串行接口 CSI10 由以下硬件构成。

表 14-1 串行接口 CSI10 的结构

项目	结构
控制电路	发送控制器 时钟启动 / 停止控制器 & 时钟相位控制器
寄存器	发送缓冲寄存器 10（SOTB10） 串行 I/O 移位寄存器 10（SIO10）
控制寄存器	串行运行模式寄存器 10（CSIM10） 串行时钟选择寄存器 10（CSIC10） 端口模式寄存器 1（PM1） 端口寄存器 1（P1）

图 14-1 串行接口 CSI10 的框图



备注 (a): SO10 输出

(1) 发送缓冲寄存器 10 (SOTB10)

发送缓冲寄存器 10 设置发送数据。

在串行运行模式选择寄存器 10 (CSIM10) 的 bit7 (CSIE10) 和 bit6 (TRMD10) 为“1”时，通过给 SOTB10 写数据，开始发送 / 接收运行。

写入 SOTB10 的数据通过串行 I/O 移位寄存器 10 从并行数据转换为串行数据，并且被输出到串行输出引脚 (SO10)。

通过 8 位存储器操作指令读写 SOTB10。

在产生复位信号后，SOTB10 变为“00H”。

注意 在 CSOT10=1 (串行通信中) 时，不能存取 SOTB10。

(2) 串行 I/O 移位寄存器 10 (SIO10)

串行 I/O 移位寄存器是执行并行 - 串行间转换的 8 位寄存器。

可通过 8 位存储器操作指令读 SIO10。

在串行运行模式寄存器 10 (CSIM10) 的 bit6 (TRMD10) 为“0”时，通过从 SIO10 读数据，开始接收运行。

接收时，数据从串行输入 (SI10) 读取到 SIO10。

在产生复位信号后，SIO10 变为“00H”。

注意 在 CSOT10=1 (串行通信中) 时，不能存取 SIO10。

14.3 串行接口 CSI10 的控制寄存器

通过以下 4 种寄存器控制串行接口 CSI10。

- 串行运行模式寄存器 10 (CSIM10)
- 串行时钟选择寄存器 10 (CSIC10)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

(1) 串行运行模式寄存器 10 (CSIM10)

串行运行模式寄存器 10 设置运行模式以及允许 / 禁止运行。

通过 1 位或 8 位存储器操作指令设置 CSIM10。

在产生复位信号后，CSIM10 变为“00H”。

图 14-2 串行运行模式寄存器 10 (CSIM10) 的格式

地址: FF80H 复位后: 00H R/W^{注1}

符号	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10	控制 3 线式串行 I/O 模式中的运行
0	禁止运行 ^{注2} ，并且异步复位内部电路 ^{注3} 。
1	允许运行

TRMD10 ^{注4}	控制发送 / 接收模式
0 ^{注5}	接收模式 (禁止发送)
1	发送 / 接收模式

DIR10 ^{注6}	指定起始位
0	MSB
1	LSB

CSOT10	通信状态标志
0	通信停止
1	正在通信

- 注
1. bit0 为只读位。
 2. 将 P10/ $\overline{\text{SCK10}}$ /TxD0、P12/SO10 用作通用端口时，CSIM10 的设置 (00H) 必须与初始状态相同。
 3. CSIM10 的 bit0 (CSOT10) 和串行 I/O 移位寄存器 10 (SIO10) 发生复位。
 4. 在 CSOT10=1 (串行通信中) 时，不能改写 TRMD10。
 5. 在 TRMD10 为“0”时，SO10 输出 (参照图 14-1 的 (a)) 固定为低电平。如果从 SIO10 读取数据，就开始接收运行。
 6. 在 CSOT10=1 (串行通信中) 时，不能改写 DIR10。

注意 必须将 bit5 置“0”。

(2) 串行时钟选择寄存器 10 (CSIC10)

串行时钟选择寄存器 10 指定数据发送 / 接收的时序以及设置串行时钟。

通过 1 位或 8 位存储器操作指令设置 CSIC10。

在产生复位信号后，CSIC10 变为“00H”。

图 14-3 串行时钟选择寄存器 10 (CSIC10) 的格式

地址: FF81H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
CSIC10	0	0	0	CKP10	DAP10	CKS102	CKS101	CKS100

CKP10	DAP10	指定数据发送 / 接收的时序	类型
0	0		1
0	1		2
1	0		3
1	1		4

CKS102	CKS101	CKS100	选择 CS110 的串行时钟 ^{注 1}			模式	
			$f_{PRS}=2\text{MHz}$	$f_{PRS}=5\text{MHz}$	$f_{PRS}=10\text{MHz}$		
0	0	0	$f_{PRS}/2$	1MHz	2.5MHz	5MHz	主控模式
0	0	1	$f_{PRS}/2^2$	500kHz	1.25MHz	2.5MHz	
0	1	0	$f_{PRS}/2^3$	250kHz	625kHz	1.25MHz	
0	1	1	$f_{PRS}/2^4$	125kHz	312.5kHz	625kHz	
1	0	0	$f_{PRS}/2^5$	62.5kHz	156.25kHz	312.5kHz	
1	0	1	$f_{PRS}/2^6$	31.25kHz	78.13kHz	156.25kHz	
1	1	0	$f_{PRS}/2^7$	15.63kHz	39.06kHz	78.13kHz	
1	1	1	SCK10 的外部时钟输入 ^{注 2}			从属模式	

注 1. 串行时钟的设置必须满足：串行时钟 $\leq 5\text{MHz}$

2. 在 STOP 模式中，不能通过 SCK10 引脚的外部时钟开始通信运行。

注意 1. 在 CSIE10=1（允许运行）时，不能写 CSIC10。

2. 将 P10/SCK10/TxD0、P12/SO10 用作通用端口时，将 CSIC10 的设置（00H）必须与初始状态相同。

3. 复位后的数据时钟的相位类型为类型 1。

备注 f_{PRS} : 外围硬件时钟频率

(3) 端口模式寄存器 1 (PM1)

端口模式寄存器 1 以 1 位单位设置端口 1 的输入 / 输出。

将 P10//SCK10/TxD0 引脚用作串行接口的数据输出时，必须将 PM10 置“0”，将 P10 的输出锁存器置“1”。

将 P12//SO10 引脚用作串行接口的数据输出时，必须将 PM12 和 P12 的输出锁存器置“0”。

将 P10//SCK10/TxD0 引脚用作串行接口的时钟输入、P11//SI10/RxD0 引脚用作串行接口的数据输入时，必须将 PM10、PM11 置“1”。此时，P10 和 P11 的输出锁存器既可是“0”也可是“1”。

通过 1 位或 8 位存储器操作指令设置 PM1。

在产生复位信号后，PM1 变为“FFH”。

图 14-4 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM1	PM17	PM16	1PM15	PM14	PM13	PM12	PM11	PM10
PM1n	选择 P1n 引脚的输入 / 输出模式 (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)							

14.4 串行接口 CSI10 的运行

串行接口 CSI10 有以下两种模式。

- 运行停止模式
- 3线式串行 I/O 模式

14.4.1 运行停止模式

在运行停止模式中，不执行串行通信，从而可降低功耗。此外，P10/SCK10/TxD0、P11/SI10/RxD0、P12/SO10 引脚可用作通用的输入 / 输出端口。

(1) 使用的寄存器

通过串行运行模式寄存器 10 (CSIM10) 设置运行停止模式。

在设定为运行停止模式时，必须将 CSIM10 的 bit7 (CSIE10) 置“0”。

(a) 串行运行模式寄存器 10 (CSIM10)

通过 1 位或 8 位存储器操作指令设置 CSIM10。

在产生复位信号后，CSIM10 变为“00H”。

地址：FF80H 复位后：00H R/W

符号	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10
	CSIE10	控制 3 线式串行 I/O 模式中的运行						
	0	禁止运行注 1，并且异步复位内部电路注 2。						

注 1. 将 P10/SCK10/TxD0、P12/SO10 用作通用端口时，CSIM10 的设置 (00H) 必须与初始状态相同。

2. CSIM10 的 bit0 (CSOT10) 和串行 I/O 移位寄存器 10 (SIO10) 发生复位。

14.4.2 3 线式串行 I/O 模式

连接内置时钟同步串行接口的外围 IC 和显示控制器等时使用 3 线式串行 I/O 模式。
通过串行时钟（SCK10）、串行输出（SO10）、串行输入（SI10）的 3 条线进行通信。

(1) 使用的寄存器

- 串行运行模式寄存器 10（CSIM0）
- 串行时钟选择寄存器 10（CSIC10）
- 端口模式寄存器 1（PM1）
- 端口寄存器 1（P1）

3 线式串行 I/O 模式的基本运行设置步骤示例如下所示。

1. 设置 CSIC10 寄存器（参照高速内部振荡时钟）
2. 设置 CSIM10 寄存器的 bit0、bit4、bit6（CSOT10、DIR10、TRMD10）（参照图 14-2）
3. 将 CSIM10 寄存器的 bit7（CSIE10）置“1”→可发送/接收
4. 将数据写入发送缓冲寄存器 10（SOTB10）→开始发送/接收数据
从串行 I/O 移位寄存器 10（SIO10）读取数据→开始接收数据

注意 必须在考虑与通信方的关系后，设置端口模式寄存器和端口寄存器。

寄存器设置与引脚的关系如下所示。

表 14-2 寄存器设置与引脚的关系

CSIE10	TRMD10	PM11	P11	PM12	P12	PM10	P10	CSI10 的运行	引脚功能		
									SI10/RxD0/P11	SO10/P12	SCK10/TxD0/P10
0	x	x 注 1	x 注 1	x 注 1	x 注 1	x 注 1	x 注 1	停止	RxD0/P11	P12	TxD0/P10 注 2
1	0	1	x	x 注 1	x 注 1	1	x	从属接收注 3	SI10	P12	SCK10（输入）注 3
1	1	x 注 1	x 注 1	0	0	1	x	从属发送注 3	RxD0/P11	SO10	SCK10（输入）注 3
1	1	1	x	0	0	1	x	从属发送 / 接收 注 3	SI10	SO10	SCK10（输入）注 3
1	0	1	x	x 注 1	x 注 1	0	1	主控接收	SI10	P12	SCK10（输出）
1	1	x 注 1	x 注 1	0	0	0	1	主控发送	RxD0/P11	SO10	SCK10（输出）
1	1	1	x	0	0	0	1	主控发送 / 接收	SI10	SO10	SCK10（输出）

- 注 1. 可作为端口功能设置。
2. 将 P10/SCK10/TxD0 用作端口功能时，必须将 CKP10 置“0”。
3. 用作从属功能时，必须将 CKS102、CKS101 和 CKS100 都置“1”。

备注 x: 忽略
CSIE10: 串行运行模式寄存器 10（CSIM10）的 bit7
TRMD10: CSIM10 的 bit6
CKP10: 串行时钟选择寄存器 10（CSIC10）的 bit4
CKS102、CKS101、CKS100: CSIC10 的 bit2 ~ bit0
PM1x: 端口模式寄存器
P1x: 端口的输出锁存器

(2) 通信运行

3 线式串行 I/O 模式以 8 位单位进行数据的发送 / 接收。数据与串行时钟同步以 1 位单位发送 / 接收。

在串行运行模式寄存器 10 (CSIM10) 的 bit6 (TRMD10) 为“1”时, 可进行数据的发送 / 接收。通过将值写入发送缓冲寄存器 10 (SOTB10), 开始发送 / 接收运行。另外, 在串行运行模式寄存器 10 (CSIM10) 的 bit6 (TRMD10) 为“0”时, 可进行数据的接收。通过从串行 I/O 移位寄存器 10 (SIO10) 读取数据, 开始接收运行。

通信开始后, CSIM10 的 bit0 (CSOT10) 变为“1”。一旦 8 位通信结束, 通信结束中断请求标志 (CSIIF10) 就被置位, 并且 CSOT10 被清“0”。接着可进行如下通信。

注意 在 CSOT10=1 (串行通信中) 时, 不能存取控制寄存器和数据寄存器。

图 14-5 3 线式串行 I/O 模式的时序 (1/2)

(a) 发送 / 接收时序 (类型 1: TRMD10=1、DIR10=0、CKP10=0、DAP10=0)

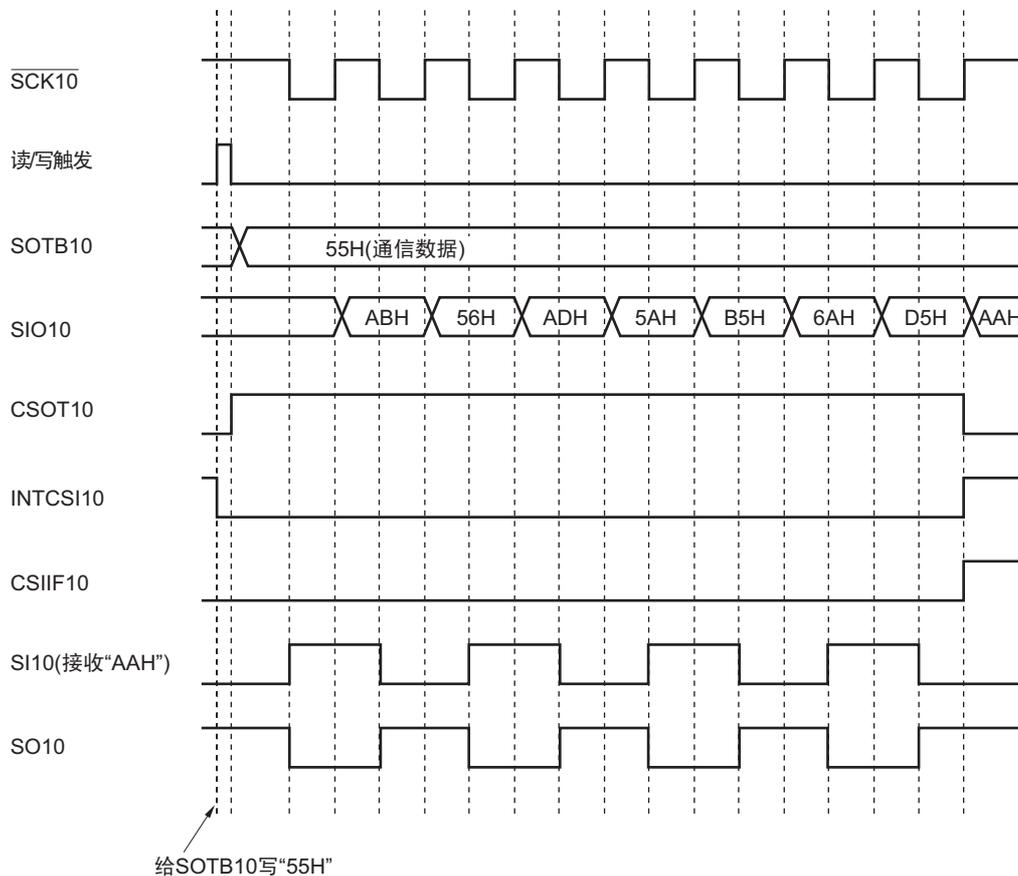


图 14-5 3 线式串行 I/O 模式的时序 (2/2)

(b) 发送 / 接收时序 (类型 2: TRMD10=1、DIR10=0、CKP10=0、DAP10=1)

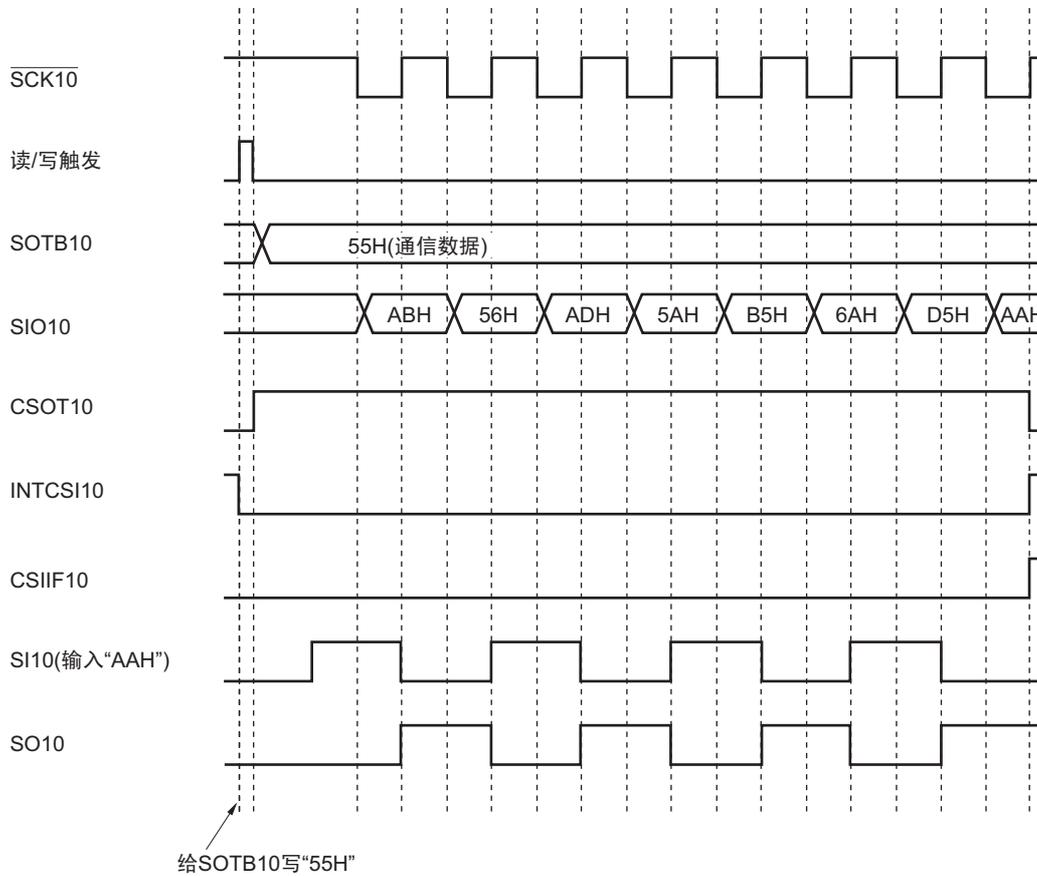
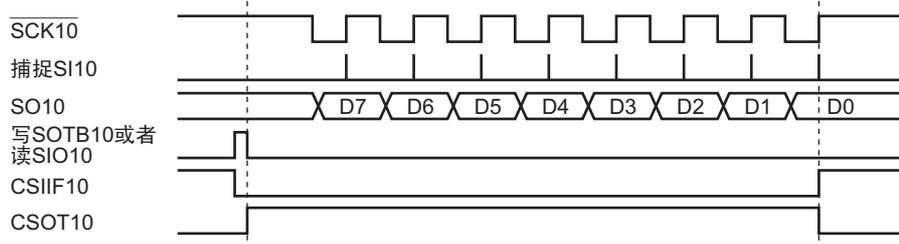


图 14-6 时钟 / 数据相位的时序 (1/2)

(a) 类型 1: CKP10=0、DAP10=0、DIR10=0



(b) 类型 2: CKP10=0、DAP10=1、DIR10=0

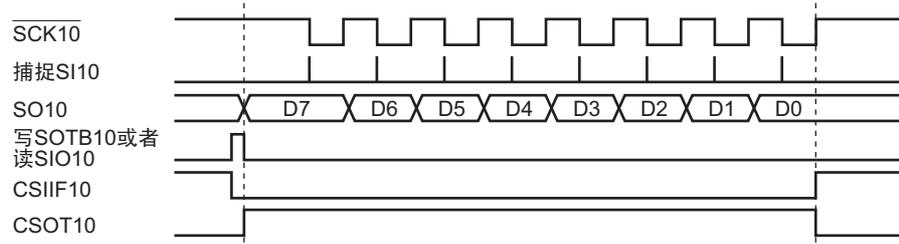
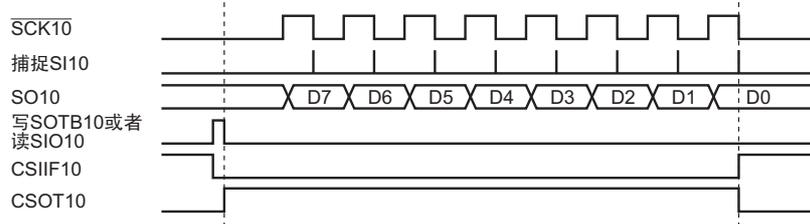
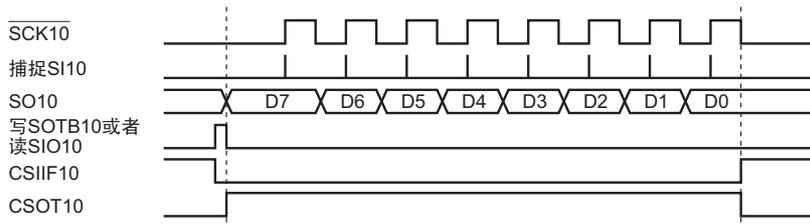


图 14-6 时钟 / 数据相位的时序 (2/2)

(c) 类型 3: CKP10=1、DAP10=0、DIR10=0



(d) 类型 4: CKP10=1、DAP10=1、DIR10=0



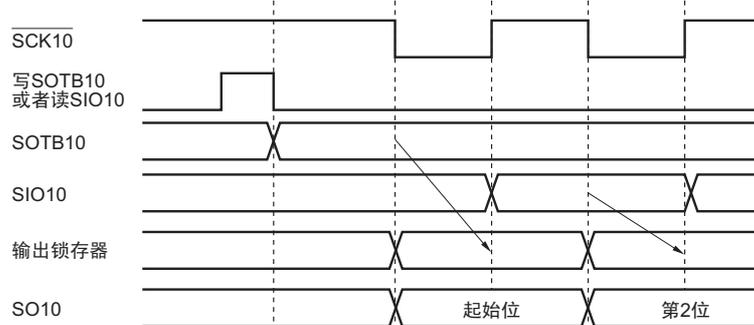
备注 上图为 MSB 优先的通信运行。

(3) SO10 引脚的输出时序 (起始位)

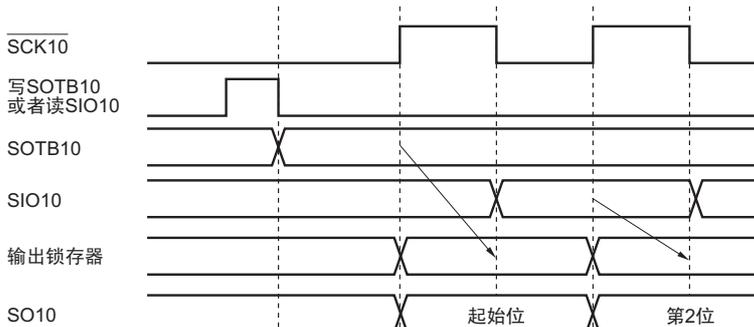
开始通信时, 从 SO10 引脚输出发送缓冲寄存器 10 (SOTB10) 的值。此时, 起始位的输出运行如下所示。

图 14-7 起始位的输出运行 (1/2)

(a) 类型 1: CKP10=0、DAP10=0



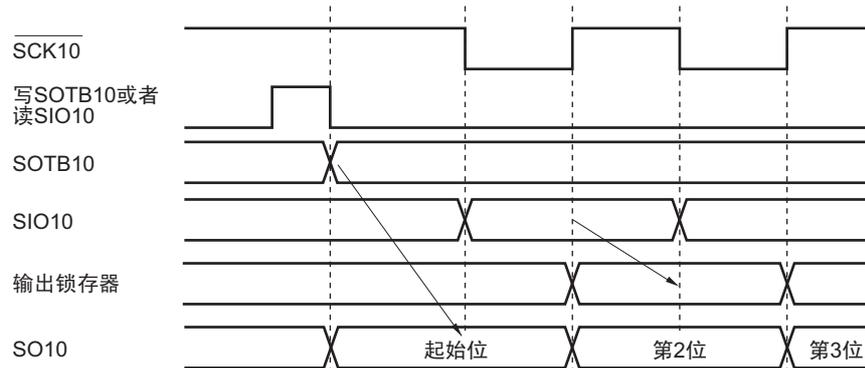
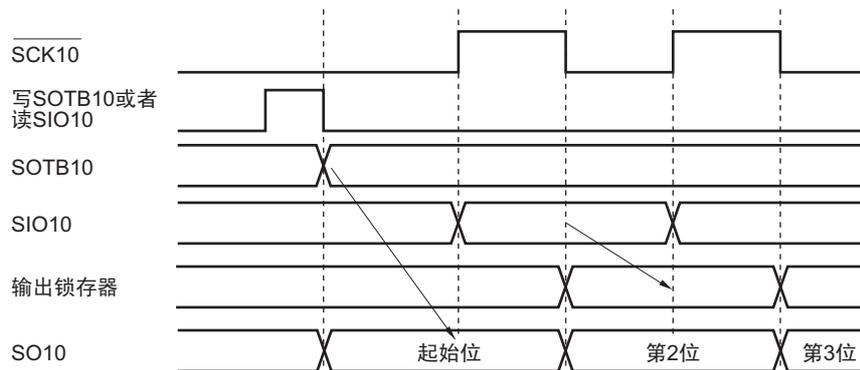
(b) 类型 3: CKP10=1、DAP10=0



在 $\overline{\text{SCK10}}$ 的下降（或者上升）沿直接将起始位从 SOTB10 寄存器锁存到输出锁存器，然后通过输出选择器从 SO10 引脚输出。在下一个 $\overline{\text{SCK10}}$ 的上升（或者下降）沿将 SOTB10 寄存器的值传送到 SIO10 寄存器，并且进行 1 位移位。于此同时，通过 SI10 引脚，将接收数据的起始位保存到 SIO10 寄存器。

在下一个 $\overline{\text{SCK10}}$ 的下降（或者上升）沿将第 2 位以后（包括第 2 位）的位从 SIO10 寄存器锁存到输出锁存器，并且从 SO10 引脚输出数据。

图 14-7 起始位的输出运行 (2/2)

(c) 类型 2: $\text{CKP10}=0$ 、 $\text{DAP10}=1$ (d) 类型 4: $\text{CKP10}=1$ 、 $\text{DAP10}=1$ 

在 SOTB10 寄存器的写信号或者 SIO10 寄存器的读信号的下降沿，直接将起始位写入 SOTB10 寄存器，然后通过输出选择器从 SO10 引脚输出。在下一个 $\overline{\text{SCK10}}$ 的下降（或者上升）沿将 SOTB10 寄存器的值传送到 SIO10 寄存器，并且进行 1 位移位。于此同时，通过 SI10 引脚，将接收数据的起始位保存到 SIO10 寄存器。

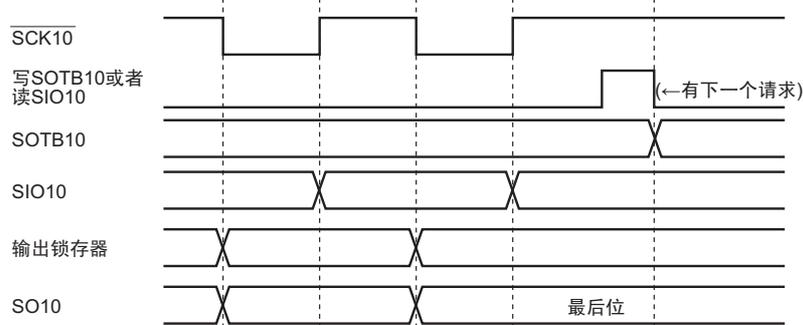
在下一个 $\overline{\text{SCK10}}$ 的上升（或者下降）沿将第 2 位以后（包括第 2 位）的位从 SIO10 寄存器锁存到输出锁存器，然后从 SO10 引脚输出数据。

(4) SO10 引脚的输出值（最后位）

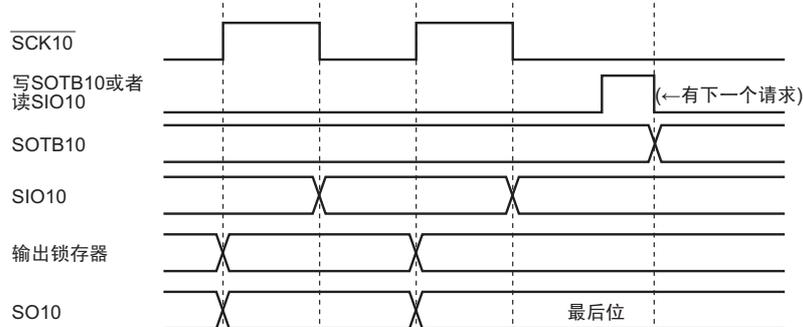
通信结束后，SO10 引脚的输出保持最后位的输出值。

图 14-8 SO10 引脚的输出值（最后位）

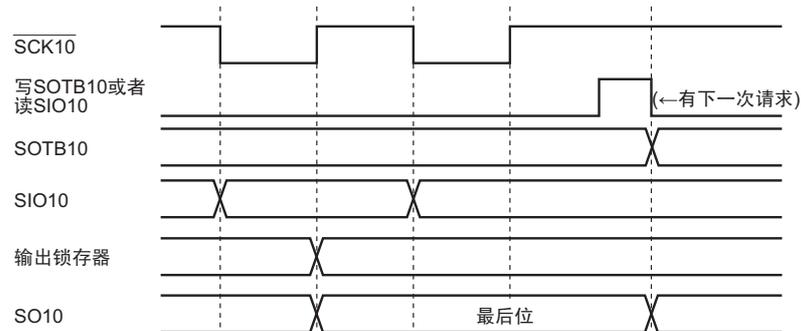
(a) 类型 1: CKP10=0、DPA10=0



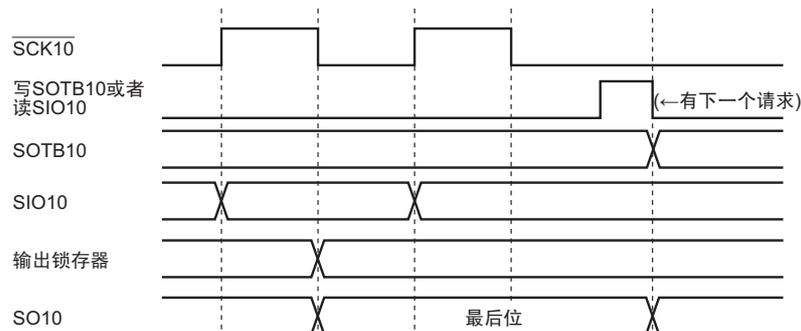
(b) 类型 3: CKP10=1、DAP10=0



(c) 类型 2: CKP10=0、DAP10=1



(d) 类型 4: CKP10=1、DAP10=1



(5) SO10 输出（参照图 14-1(a)）

如果将串行运行模式寄存器 10（CSIM10）的 bit7（CSIE10）置“0”，SO10 输出的状态如下所示。

表 14-3 SO10 输出的状态

TRMD10	DAP10	DIR10	SO10 输出注 1
TRMD10=0 注 2	—	—	低电平输出注 2
TRMD10=1	DAP10=0	—	SO10 锁存器的值 (低电平输出)
	DAP10=1	DIR10=0	SOTB10 的 bit7 的值
		DIR10=1	SOTB10 的 bit0 的值

注 1. 实际的 SO10/P12 引脚的 SO10 输出取决于 PM12 和 P12。
2. 复位时的状态。

注意 如果将值写入 TRMD10、DAP10、DIR10，SO10 的输出值就发生变化。

第 15 章 串行接口 IIC0

15.1 串行接口 IIC0 的功能

串行接口 IIC0 有以下 2 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）

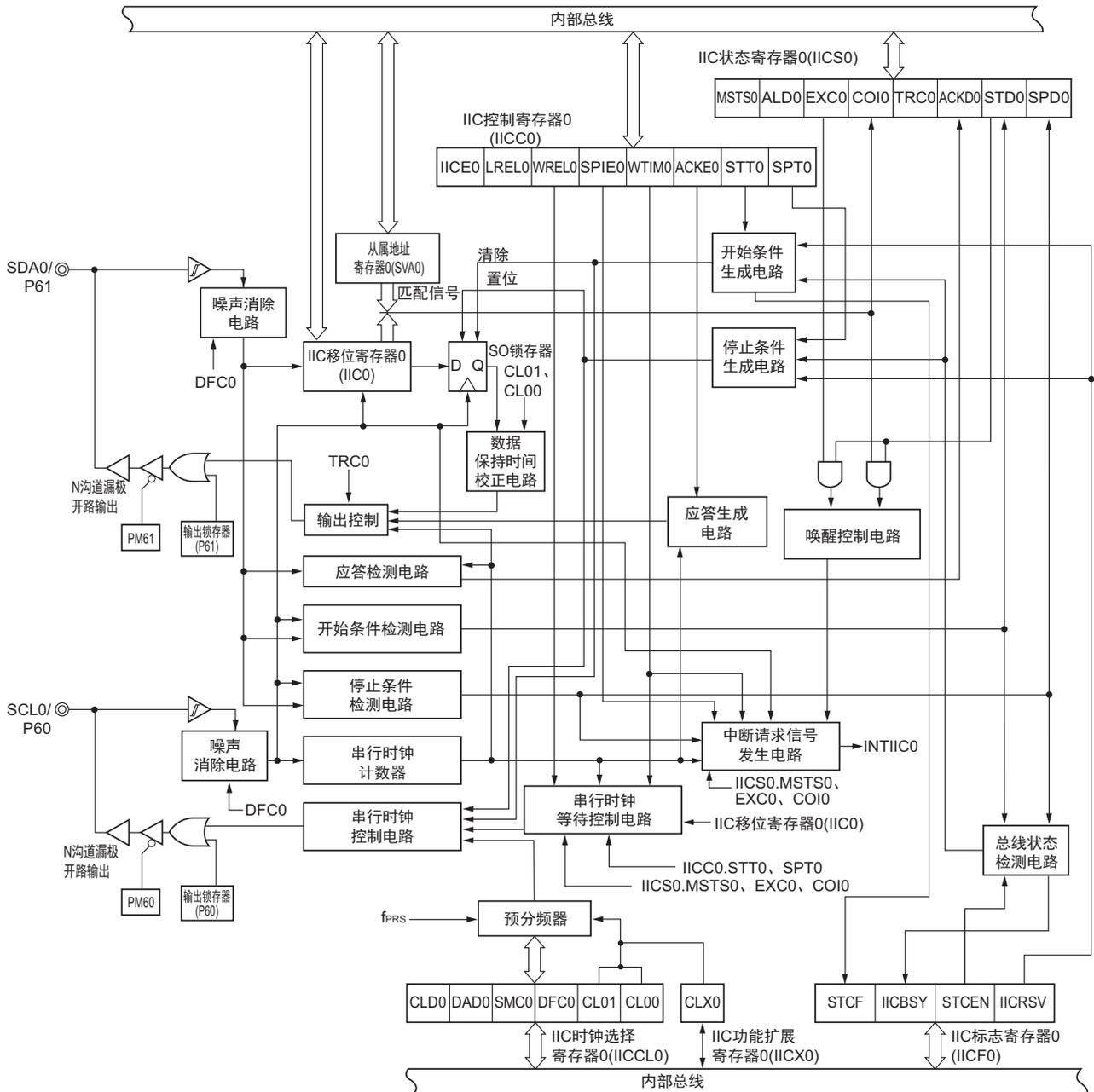
此模式通过串行时钟（SCL0）和串行数据总线（SDA0）的 2 条线，与多个设备进行 8 位数据传送。

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。

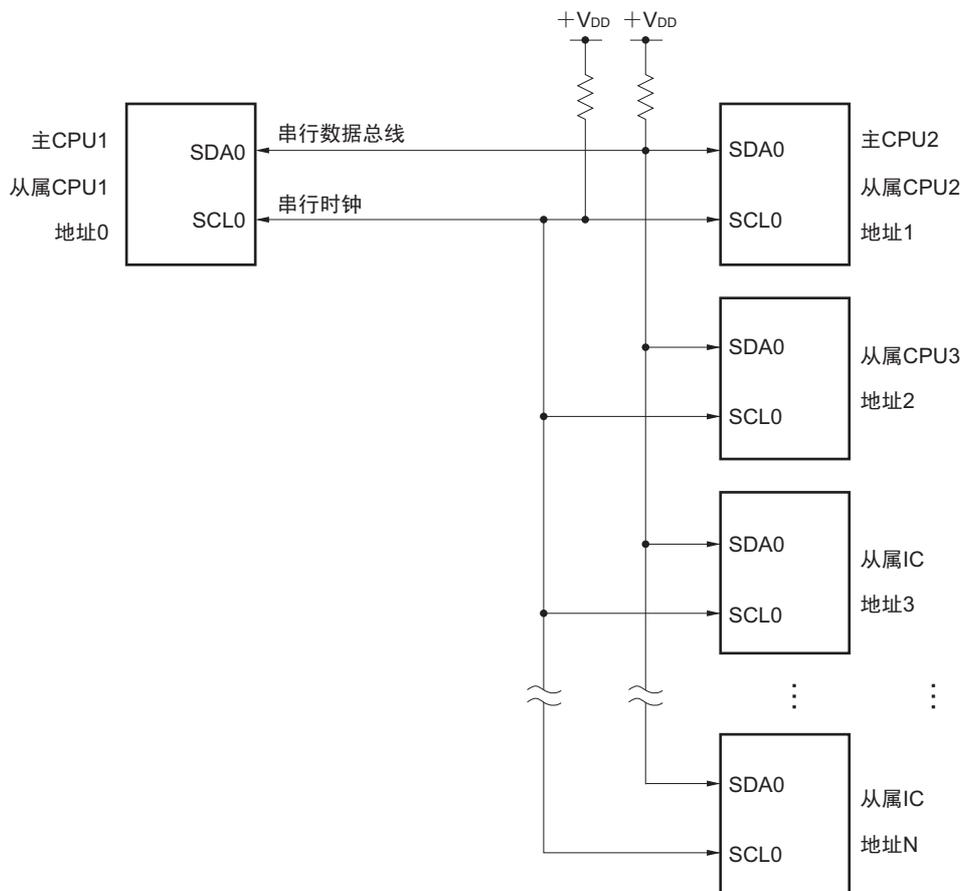
因为串行接口 IIC0 的 SCL0 引脚和 SDA0 引脚用作漏极开路输出，所以串行时钟线和串行数据总线需要上拉电阻。

串行接口 IIC0 的框图如图 15-1 所示。

图 15-1 串行接口 IIC0 的框图



串行总线的结构如图 15-2 所示。

图 15-2 I²C 总线的串行总线结构例子

15.2 串行接口 IIC0 的结构

串行接口 IIC0 由以下硬件构成。

表 15-1 串行接口 IIC0 的结构

项目	结构
寄存器	IIC 移位寄存器 0 (IIC0) 从属地址寄存器 0 (SVA0)
控制寄存器	IIC 控制寄存器 0 (IICC0) IIC 状态寄存器 0 (IICS0) IIC 标志寄存器 0 (IICF0) IIC 时钟选择寄存器 0 (IICCL0) IIC 功能扩展寄存器 0 (IICX0) 端口模式寄存器 6 (PM6) 端口寄存器 6 (P6)

(1) IIC 移位寄存器 0 (IIC0)

IIC0 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 IIC0 寄存器来控制实际的接收和发送。

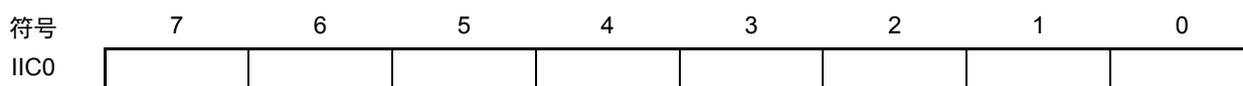
在等待期间，通过写 IIC0 寄存器来解除等待，开始传送数据。

通过 8 位存储器操作指令设置 IIC0。

在产生复位信号后，IIC0 变为“00H”。

图 15-3 IIC 移位寄存器 0 (IIC0) 的格式

地址：FFA5H 复位后：00H R/W



注意 1. 在数据传送过程中，不能给 IIC0 写数据。

2. 只能在等待期间读写 IIC0。除了等待期间以外，禁止在通信状态下存取 IIC0。但是，在主控设备的情况下，能在将通信触发位 (STT0) 置“1”后写一次 IIC0 寄存器。

3. 当保留预约时，必须在检测到由停止条件产生的中断后给 IIC0 寄存器写数据。

(2) 从属地址寄存器 0 (SVA0)

这是在用作从属设备时保存 7 位本地站地址 {A6, A5, A4, A3, A2, A1, A0} 的寄存器。

通过 8 位存储器操作指令设置 SVA0 寄存器。但是，在 STD0 位为“1”（检测到开始条件）时，禁止改写此寄存器。

在产生复位信号后，SVA0 变为“00H”。

图 15-4 从属地址寄存器 0 (SVA0) 的格式

地址：FFA7H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
SVA0								0注

注 bit0 固定为“0”。

(3) SO 锁存器

SO 锁存器保持 SDA0 引脚的输出电平。

(4) 唤醒控制电路

当设定在从属地址寄存器 0 (SVA0) 的地址值和接收到的地址相同时或者当接收到扩展码时，此电路产生中断请求 (INTIIC0)。

(5) 预分频器

预分频器选择使用的采样时钟。

(6) 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，并且检查是否进行了 8 位数据的发送和接收。

(7) 中断请求信号发生电路

此电路控制产生中断请求信号 (INTIIC0)。

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 WTIM0 位进行设置）
- 因检测到停止条件而产生中断请求（通过 SPIE0 位进行设置）

备注 WTIM0 位：IIC 控制寄存器 0 (IICC0) 的 bit3

SPIE0 位：IIC 控制寄存器 0 (IICC0) 的 bit4

(8) 串行时钟控制电路

在主控模式中，此电路从采样时钟生成输出到 SCL0 引脚的时钟。

(9) 串行时钟等待控制电路

此电路控制等待时序。

(10) 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。

(11) 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。

(12) 开始条件生成电路

如果将 STT0 位置“1”，此电路就生成开始条件。

但是，在禁止预约通信的状态下（IICRSV 位 =1）并且没释放总线（IICBSY 位 =1）时，忽视开始条件请求并且将 STCF 位置“1”。

(13) 停止条件生成电路

如果将 SPT0 位置“1”，此电路就生成停止条件。

(14) 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 STCEN 位设置总线状态检测电路的初始状态。

备注	STT0 位:	IIC 控制寄存器 0 (IICC0) 的 bit1
	SPT0 位:	IIC 控制寄存器 0 (IICC0) 的 bit0
	IICRSV 位:	IIC 标志寄存器 0 (IICF0) 的 bit0
	IICBSY 位:	IIC 标志寄存器 0 (IICF0) 的 bit6
	STCF 位:	IIC 标志寄存器 0 (IICF0) 的 bit7
	STCEN 位:	IIC 标志寄存器 0 (IICF0) 的 bit1

15.3 串行接口 IIC0 的控制寄存器

通过以下 7 种寄存器控制串行接口 IIC0。

- IIC 控制寄存器 0 (IICC0)
- IIC 标志寄存器 0 (IICF0)
- IIC 状态寄存器 0 (IICS0)
- IIC 时钟选择寄存器 0 (IICCL0)
- IIC 功能扩展寄存器 0 (IICX0)
- 端口模式寄存器 6 (PM6)
- 端口寄存器 6 (P6)

(1) IIC 控制寄存器 0 (IICC0)

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置 IICC0 寄存器。但是，必须在 IICE0 位为“0”时或者等待期间设置 SPIE0 位、WTIM0 位和 ACKE0 位，而且在将 IICE0 位从“0”置为“1”时能同设置这些位。

在产生复位信号后，IIC0 变为“00H”。

图 15-5 IIC 控制寄存器 0 (IICC0) 的格式 (1/4)

地址: FFA6H 复位: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICC0	IICE0	LRELO	WRELO	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I ² C 运行的允许
0	停止运行。对 IIC 状态寄存器 0 (IICS0) 进行复位 ^{注 1} ，并且停止内部运行。
1	允许运行。
必须在 SCL0 线和 SDA0 线为高电平的状态下将此位置“1”。	
清除条件 (IICE0=0)	置位条件 (IICE0=1)
<ul style="list-style-type: none"> 通过指令清除。 当复位时 	<ul style="list-style-type: none"> 通过指令置位。

LRELO ^{注 2、3}	通信的退出
0	通常运行
1	退出当前的通信，进入待机状态。执行后自动清“0”。 在接收到与本站无关的扩展码等情况时使用。 SCL0 线和 SDA0 线变为高阻状态。 IIC 控制寄存器 0 (IICC0) 和 IIC 状态寄存器 0 (IICS0) 中的以下标志被清“0”： •STT0 •SPT0 •MSTS0 •EXC0 •COI0 •TRC0 •ACKD0 •STD0
变为退出通信的待机状态，保持到满足以下的通信参加条件为止。	
<ul style="list-style-type: none"> 在检测到停止条件后作为主控设备启动。 在检测到开始条件后地址匹配或者接收到扩展码。 	
清除条件 (LRELO=0)	置位条件 (LRELO=1)
<ul style="list-style-type: none"> 在执行后自动清除。 当复位时 	<ul style="list-style-type: none"> 通过指令置位。

WRELO ^{注 2、3}	等待的解除
0	不解除等待。
1	解除等待。在解除等待后自动清除。
如果在发送状态下 (TRC0=1) 的第 9 个时钟等待期间将 WRELO 位 (解除等待) 置位，SDA0 线就变为高阻抗状态 (TRC0=0)。	
清除条件 (WRELO=0)	置位条件 (WRELO=1)
<ul style="list-style-type: none"> 在执行后自动清除。 当复位时 	<ul style="list-style-type: none"> 通过指令置位。

- 注 1. IICS0 寄存器、IICF0 寄存器的 STCF0 位和 IICBSY0 位以及 IICCL0 寄存器的 CLD0 位和 DAD0 位发生复位。
2. 在 IICE0 位为“0”的状态下，此位的信号无效。
3. LRELO 位和 WRELO 位的读取值总是“0”。

注意 如果在 SCL0 线为高电平、SDA0 线为低电平并且数字滤波器为 ON (IICCL0 寄存器的 DFC0=1) 时允许 I²C 运行 (IICE0=1)，就立即检测开始条件。此时，必须在允许 I²C 运行 (IICE0=1) 后连续通过 1 位存储器操作指令将 LRELO 位置“1”。

图 15-5 IIC 控制寄存器 0 (IICC0) 的格式 (2/4)

SPIE0 注 1	允许或者禁止停止条件检测产生的中断请求	
0	禁止	
1	允许	
清除条件 (SPIE0=0)		置位条件 (SPIE0=1)
<ul style="list-style-type: none"> 通过指令清除。 当复位时 		<ul style="list-style-type: none"> 通过指令置位。

WTIM0 注 1	等待和中断请求的控制	
0	<p>在第 8 个时钟的下降沿产生中断请求信号。</p> <p>主控设备：在输出 8 个时钟后，将时钟输出置为低电平进行等待。</p> <p>从属设备：在输入 8 个时钟后，将时钟置为低电平，然后等待主控设备。</p>	
1	<p>在第 9 个时钟的下降沿产生中断请求信号。</p> <p>主控设备：在输出 9 个时钟后，将时钟输出置为低电平进行等待。</p> <p>从属设备：在输入 9 个时钟后，将时钟置为低电平，然后等待主控设备。</p>	
<p>在地址传送期间，与此位的设定无关，在第 9 个时钟的下降沿产生中断；在地址传送结束后，此位的设定有效。主控设备在地址传送期间的第 9 个时钟下降沿进入等待。接收到本地站地址的从属设备在产生应答 (ACK) 后的第 9 个时钟下降沿进入等待，但是接收到扩展码的从属设备在第 8 个时钟下降沿进入等待。</p>		
清除条件 (WTIM0=0)		置位条件 (WTIM0=1)
<ul style="list-style-type: none"> 通过指令清除。 当复位时 		<ul style="list-style-type: none"> 通过指令置位。

ACKE0 注 1、2	应答控制	
0	禁止应答。	
1	允许应答。在第 9 个时钟期间将 SDA0 线置为低电平。	
清除条件 (ACKE0=0)		置位条件 (ACKE0=1)
<ul style="list-style-type: none"> 通过指令清除。 当复位时 		<ul style="list-style-type: none"> 通过指令置位。

- 注 1. 在 IICE0 位为“0”的状态下，此位的信号无效。必须在此期间设定此位。
2. 在地址传送过程中并且不是扩展码时，设定值无效。
- 当为从属设备并且地址匹配时，与设定值无关而生成应答。

图 15-5 IIC 控制寄存器 0 (IICC0) 的格式 (3/4)

STT0 注	开始条件的触发
0	不生成开始条件。
1	<p>当总线被释放时（待机状态，IICBSY 位为“0”）： 如果将此位置“1”，就生成开始条件（作为主控设备的启动）。</p> <p>当第三方正在通信时：</p> <ul style="list-style-type: none"> • 允许通信预约功能的情况（IICRSV=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（IICRSV=1） 即使将此位置“1”，也清除 STT0 位并且将 STT0 清除标志（STCF）置“1”，不生成开始条件。 <p>等待状态（主控设备）： 在解除等待后生成重新开始条件。</p>
<p>有关置位时序的注意事项：</p> <ul style="list-style-type: none"> • 主控接收：禁止在传送过程中将此位置“1”。只有在将 ACKE0 位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 • 主控发送：在应答期间，可能不能正常生成开始条件。必须在输出第 9 个时钟后的等待期间将此位置“1”。 • 禁止与停止条件的触发（SPT0）同时置“1”。 • 在将 STT0 位置“1”后，禁止在清“0”前再次将此位“1”。 	
清除条件（STT0=0）	置位条件（STT0=1）
<ul style="list-style-type: none"> • 在禁止通信预约的状态下将 STT0 位置“1”。 • 在仲裁失败时 • 主控设备生成开始条件。 • 因 LRELO 位为“1”（退出通信）而进行的清除 • 当 IICE0 位为“0”（停止运行）时 • 当复位时 	<ul style="list-style-type: none"> • 通过指令置位。

注 在 IICE0 位为“0”的状态下，此位的信号无效。

备注 1. 如果在设定数据后读 bit1（STT0），此位就变为“0”。

2. IICRSV：IIC 标志寄存器 0（IICF0）的 bit0

STCF：IIC 标志寄存器 0（IICF0）的 bit7

图 15-5 IIC 控制寄存器 0 (IICC0) 的格式 (4/4)

SPT0	停止条件的触发	
0	不生成停止条件。	
1	生成停止条件 (作为主控设备的传送结束)。	
有关置位时序的注意事项： <ul style="list-style-type: none"> • 主控接收：禁止在传送过程中将此位置“1”。只有在将 ACKE0 位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 • 主控发送：在应答期间，可能不能正常生成停止条件。必须在输出第 9 个时钟后的等待期间将此位置“1”。 • 禁止与开始条件的触发 (STT0) 同时置“1”。 • 只有在主控设备的情况下才能将 SPT0 位置“1”。 • 在 WTIM0 位为“0”时，必须注意：如果在输出 8 个时钟后的等待期间将 SPT0 位置“1”，就在解除等待后的第 9 个时钟的高电平期间生成停止条件。必须在输出 8 个时钟后的等待期间将 WTIM0 位从“0”置为“1”并且在输出第 9 个时钟后的等待期间将 SPT0 位置“1”。 • 在将 SPT0 位置“1”后，禁止在清“0”前再次将此位置“1”。 		
清除条件 (SPT0=0)		置位条件 (SPT0=1)
<ul style="list-style-type: none"> • 当仲裁失败时 • 在检测到停止条件后自动清除。 • 因 LREL0 位为“1” (退出通信) 而进行的清除 • 当 IICE0 位为“0” (停止运行) 时 • 当复位时 		<ul style="list-style-type: none"> • 通过指令置位。

注意 在 IIC 状态寄存器 0 (IICS0) 的 bit3 (TRC0) 为“1” (发送状态) 时，如果在第 9 个时钟将 IICC0 寄存器的 bit5 (WREL0) 置“1”来解除等待，就在清除 TRC0 位 (接收状态) 后将 SDA0 线置为高阻抗。必须通过写 IIC 移位寄存器 0 (IIC0) 进行 TRC 位为“1” (发送状态) 时的等待解除。

备注 如果在设定数据后读 bit0 (SPT0)，此位就变为“0”。

(2) IIC 状态寄存器 0 (IICS0)

这是表示 I²C 状态的寄存器。

只有在 STT0 位为“1”并且等待期间，才能通过 1 位或者 8 位存储器操作指令读 IICS0。

在产生复位信号后，IICS0 变为“00H”。

注意 如果从 IICS0 寄存器读取数据，就产生等待。在外围硬件时钟（f_{PRS}）停止时，不能从 IICS0 寄存器读取数据。详细内容请参照“第 27 章 等待的注意事项”。

图 15-6 IIC 状态寄存器 0 (IICS0) 的格式 (1/3)

地址：FFAAH 复位后：00H R

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	主控状态
0	从属状态或者通信待机状态
1	主控通信状态
清除条件 (MSTS0=0)	置位条件 (MSTS0=1)
<ul style="list-style-type: none"> 当检测到停止条件时 当 ALD0 位为“1”（仲裁失败）时 因 LREL0 位为“1”（退出通信）而进行的清除 当 IICE0 位从“1”变为“0”（停止运行）时 当复位时 	<ul style="list-style-type: none"> 当生成开始条件时

ALD0	仲裁失败的检测
0	表示未发生仲裁或者赢得仲裁。
1	表示仲裁失败。清除 MSTS0 位。
清除条件 (ALD0=0)	置位条件 (ALD0=1)
<ul style="list-style-type: none"> 在读 IICS0 寄存器后自动清除注。 当 IICE0 位从“1”变为“0”（停止运行）时 当复位时 	<ul style="list-style-type: none"> 当仲裁失败时

注 即使对 IICS0 寄存器以外的位执行 1 位存储器操作指令，也清除此位。因此，在使用 ALD0 位时，必须在读其他位前先读 ALD0 位的数据。

备注 LREL0: IIC 控制寄存器 0 (IICC0) 的 bit6

IICE0: IIC 控制寄存器 0 (IICC0) 的 bit7

图 15-6 IIC 状态寄存器 0 (IICS0) 的格式 (2/3)

EXC0	扩展码的接收检测	
0	未接收到扩展码。	
1	接收到扩展码。	
清除条件 (EXC0=0)		置位条件 (EXC0=1)
<ul style="list-style-type: none"> 当检测到开始条件时 当检测到停止条件时 因 LREL0 位为“1” (退出通信) 而进行的清除 当 IICE0 位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当接收的地址数据的高 4 位为“0000”或者“1111”时 (在第 8 个时钟的上升沿进行置位)。

COI0	匹配地址的检测	
0	地址不同。	
1	地址相同。	
清除条件 (COI0=0)		置位条件 (COI0=1)
<ul style="list-style-type: none"> 当检测到开始条件时 当检测到停止条件时 因 LREL0 位为“1” (退出通信) 而进行的清除 当 IICE0 位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当接收地址和本地站地址 (从属地址寄存器 0 (SVA0)) 相同时 (在第 8 个时钟的上升沿置位)。

TRC0	发送 / 接收的状态检测	
0	接收状态 (发送状态除外)。将 SDA0 线置为高阻抗。	
1	发送状态。设定为能将 SO0 锁存器的值输出到 SDA0 线 (在第 1 字节的第 9 个时钟的下降沿以后有效)。	
清除条件 (TRC0=0)		置位条件 (TRC0=1)
<主控设备和从属设备> <ul style="list-style-type: none"> 当检测到停止条件时 因 LREL0 位为“1” (退出通信) 而进行的清除 当 IICE0 位从“1”变为“0” (停止运行) 时 因 WREL0 位为“1” (解除等待) 而进行的清除注 当 ALD0 位从“0”变为“1” (仲裁失败) 时 当复位时 不参加通信的情况 (MSTS0、EXC0、COI0=0) <主控设备> <ul style="list-style-type: none"> 当第 1 字节的 LSB (传送方向指示位) 输出“1”时。 <从属设备> <ul style="list-style-type: none"> 当检测到开始条件时 当第 1 字节的 LSB (传送方向指示位) 输入“0”时 		<主控设备> <ul style="list-style-type: none"> 当生成开始条件时 当将第 1 字节 (地址传送) 的 LSB (传送方向指示位) 输出“0” (主控发送) 时 <从属设备> <ul style="list-style-type: none"> 当来自主控设备的第 1 字节 (地址传送) 的 LSB (传送方向指示位) 输入“1” (从属发送) 时

注 在 IIC 状态寄存器 0 (IICS0) 的 bit3 (TRC0) 为“1” (发送状态) 时, 如果在第 9 个时钟将 IIC 控制寄存器 0 (IICC0) 的 bit5 (WREL0) 置“1”来解除等待, 就在清除 TRC0 位 (接收状态) 后将 SDA0 线置为高阻抗。必须通过写 IIC 移位寄存器 0 进行 TRC0 位为“1” (发送状态) 时的等待解除。

备注 LREL0: IIC 控制寄存器 0 (IICC0) 的 bit6
IICE0: IIC 控制寄存器 0 (IICC0) 的 bit7

图 15-6 IIC 状态寄存器 0 (IICS0) 的格式 (3/3)

ACKD0	应答 (ACK) 的检测	
0	未检测到应答。	
1	检测到应答。	
清除条件 (ACKD0=0)		置位条件 (ACKD0=1)
<ul style="list-style-type: none"> 当检测到停止条件时 当下一字节的第 1 个时钟上升时 因 LREL0 位为“1” (退出通信) 而进行的清除 当 IICE0 位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 在 SCL0 线的第 9 个时钟上升沿将 SDA0 线为低电平时

STD0	开始条件的检测	
0	未检测到开始条件。	
1	检测到开始条件, 表示处于地址传送期间。	
清除条件 (STD0=0)		置位条件 (STD0=1)
<ul style="list-style-type: none"> 当检测到停止条件时 在地址传送后的下一个字节的第 1 个时钟上升时 因 LREL0 位为“1” (退出通信) 而进行的清除 当 IICE0 位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当检测到开始条件时

SPD0	停止条件的检测	
0	未检测到停止条件。	
1	检测到停止条件, 主控设备结束通信并且已释放总线。	
清除条件 (SPD0=0)		置位条件 (SPD0=1)
<ul style="list-style-type: none"> 在将此位置位后, 在检测到开始条件后的地址传送位的第 1 个时钟上升时 当 IICE0 位从“1”变为“0” (停止运行) 当复位时 		<ul style="list-style-type: none"> 当检测到停止条件时

备注 LREL0: IIC 控制寄存器 0 (IICC0) 的 bit6
IICE0: IIC 控制寄存器 0 (IICC0) 的 bit7

(3) IIC 标志寄存器 0 (IICF0)

这是设定 I²C 运行模式以及表示 I²C 总线状态的寄存器。

通过 1 位或者 8 位存储器操作指令设置 IICF0。但是, STT0 清除标志 (STCF) 和 I²C 总线状态标志 (IICBSY) 位为只读位。

通过 IICRSV 位设置允许或者禁止通信预约功能, 并且通过 STCEN 位设置 IICBSY 位的初始值。

只有在禁止 I²C 运行 (IIC 控制寄存器 0 (IICC0) 的 bit7 (IICE0) =0) 时才能写 IICRSV 位和 STCEN 位。在允许运行后, IICF0 寄存器为只读寄存器。

在产生复位信号后, IICF0 变为“00H”。

图 15-7 IIC 标志寄存器 0 (IICF0) 的格式

地址: FFABH 复位后: 00H R/W注

符号	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	STT0 清除标志
0	发行开始条件。
1	无法发行开始条件而清除 STT0 标志。
清除条件 (STCF=0)	置位条件 (STCF=1)
<ul style="list-style-type: none"> 因 STT0 位为“1”而进行的清除 当 IICE0 位为“0” (停止运行) 时 当复位时 	<ul style="list-style-type: none"> 在设定为禁止通信预约 (IICRSV=1) 的状态下无法发行开始条件而将 STT0 位清“0”时

IICBSY	I ² C 总线状态标志
0	总线释放状态 (STCEN=1 时的通信初始状态)
1	总线通信状态 (STCEN=0 时的通信初始状态)
清除条件 (IICBSY=0)	置位条件 (IICBSY=1)
<ul style="list-style-type: none"> 当检测到停止条件时 当 IICE0 位为“0” (停止运行) 时 当复位时 	<ul style="list-style-type: none"> 当检测到开始条件时 STCEN 位为“0” 时的 IICE0 位的置位

STCEN	初始开始允许触发
0	在允许运行 (IICE0=1) 后, 通过检测停止条件而允许生成开始条件。
1	在允许运行 (IICE0=1) 后, 不通过检测停止条件而允许而生成开始条件。
清除条件 (STCEN=0)	置位条件 (STCEN=1)
<ul style="list-style-type: none"> 当检测到开始条件时 当复位时 	<ul style="list-style-type: none"> 通过指令置位。

IICRSV	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 (IICRSV=0)	置位条件 (IICRSV=1)
<ul style="list-style-type: none"> 通过指令清除。 当复位时 	<ul style="list-style-type: none"> 通过指令置位。

注 bit6 和 bit7 是只读位。

注意 1. 只有在停止运行 (IICE0=0) 时才能写 STCEN 位。

2. 如果 STCEN 位为“1”, 就与实际的总线状态无关而认为总线为释放状态 (IICBSY=0), 因此为了在发行第 1 个开始条件 (STT0=1) 时避免破坏其他通信, 必须确认没有正在通信的第三方。

3. 只有在停止运行 (IICE0=0) 时才能写 IICRSV。

备注 STT0: IIC 控制寄存器 0 (IICC0) 的 bit1

IICE0: IIC 控制寄存器 0 (IICC0) 的 bit7

(4) IIC 时钟选择寄存器 (IICCL0)

这是用于设置 I²C 的传送时钟的寄存器。

通过 1 位或者 8 位存储器操作指令设置 IICCL0。但是，CLD0 位和 DAD0 位是只读位。通过与 IIC 功能扩展寄存器 0 (IICX0) 的 bit0 (CLX0) 的组合设置 SMC0、CL01、CL00 位 (参照“15.3(6) I²C 的传送时钟的设置方法”)。

在 IIC 控制寄存器 0 (IICC0) 的 bit7 (IICE0) 为“0”时，才能设置 IICCL0。

在产生复位信号后，IICCL0 变为“00H”。

图 15-8 IIC 时钟选择寄存器 0 (IICCL0) 的格式

地址: FFA8H 复位后: 00H R/W 注

符号	7	6	<5>	<4>	<3>	<2>	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	CL01	CL00

CLD0	SCL0 引脚的电平检测 (只在 IICE0=1 时有效)
0	检测出 SCL0 引脚为低电平。
1	检测出 SCL0 引脚为高电平。
清除条件 (CLD0=0)	
<ul style="list-style-type: none"> 当 SCL0 引脚为低电平时 当 IICE0=0 时 (运行停止) 当复位时 	
置位条件 (CLD0=1)	
<ul style="list-style-type: none"> 当 SCL0 引脚为高电平时 	

DAD0	SDA0 引脚的电平检测 (只在 IICE0=1 时有效)
0	检测出 SDA0 引脚为低电平。
1	检测出 SDA0 引脚为高电平。
清除条件 (DAD0=0)	
<ul style="list-style-type: none"> 当 SDA0 引脚为低电平时 当 IICE0=0 时 (运行停止) 当复位时 	
置位条件 (DAD0=1)	
<ul style="list-style-type: none"> 当 SDA0 引脚为高电平时 	

SMC0	运行模式的切换
0	在标志模式下运行
1	在高速模式下运行

DFC0	控制数字滤波器的运行
0	数字滤波器 OFF
1	数字滤波器 ON
只能在高速模式中使用数字滤波器。	
在高速模式中，不管是将 DFC0 置“1”或者清“0”，传送时钟都不发生变化。	
在高速模式中，数字滤波器用于消除噪声。	

注 bit4 和 bit5 为只读位。

备注 IICE0: IIC 控制寄存器 0 (IICC0) 的 bit7。

(5) IIC 功能扩展寄存器 0 (IICX0)

这是用于设置 I²C 的功能扩展的寄存器。

通过 1 位或者 8 位存储器操作指令设置 IICX0。通过与 IIC 时钟选择寄存器 0 (IICCL0) 的 bit3、bit1、bit0 (SMC0、CL01、CL00) 的组合设置 CLX0 位 (参照“15.3(6) I²C 的传送时钟的设置方法”)。

在 IIC 控制寄存器 0 (IICC0) 的 bit7 (IICE0) 为“0”时, 才能设置 IICX0。

在产生复位信号后, IICX0 变为“00H”。

图 15-9 IIC 功能扩展寄存器 0 (IICX0) 的格式

地址: FFA9H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	<0>
IICX0	0	0	0	0	0	0	0	CLX0

(6) I²C 的传送时钟的设置方法

通过以下计算式求 I²C 的传送时钟频率 (f_{SCL})。

$$f_{SCL} = 1 / (m \times T + t_R + t_F)$$

$m = 12、18、24、44、66、86$ (参照“表 15-2 选择时钟的设置”)

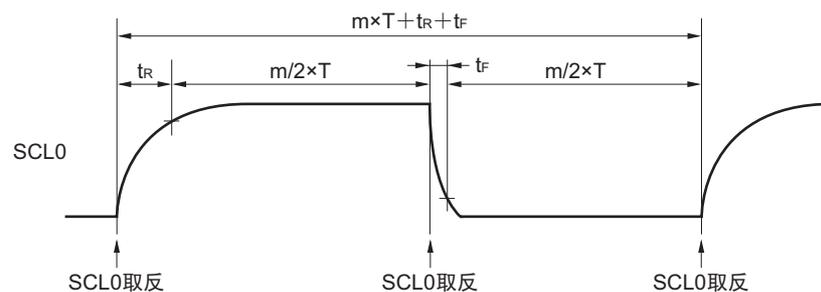
$T: 1/f_W$

t_R : SCL0 的上升时间

t_F : SCL0 的下降时间

假设 $f_W = f_{PRS} / 2 = 4.19\text{MHz}$ 、 $m = 86$ 、 $t_R = 200\text{ns}$ 、 $t_F = 50\text{ns}$, 通过以下计算式求 I²C 的传送时钟频率 (f_{SCL})。

$$f_{SCL} = 1 / (86 \times 238.7\text{ns} + 200\text{ns} + 50\text{ns}) \approx 48.1\text{kHz}$$



通过 IIC 时钟选择寄存器 0 (IICCL0) 的 bit3、bit1、bit0 (SMC0、CL01、CL00) 和 IIC 功能扩展寄存器 0 (IICX0) 的 bit0 (CLX0) 的组合, 设置选择的时钟。

表 15-2 选择时钟的设置

IICX0	IICCL0			选择时钟 (f_W) 注	传送时钟 (f_W/m)	可设置的选择时钟 (f_W) 的范围	运行模式	
	bit0	bit3	bit1					bit0
CLX0	SMC0	CL01	CL00					
0	0	0	0	$f_{PRS}/2$	$f_W/44$	2.00MHz ~ 4.19MHz	标准模式 (SMC0 位 =0)	
0	0	0	1	$f_{PRS}/2$	$f_W/86$	4.19MHz ~ 8.38MHz		
0	0	1	0	$f_{PRS}/4$	$f_W/86$			
0	0	1	1	禁止设置				
0	1	0	x	$f_{PRS}/2$	$f_W/24$	4.00MHz ~ 8.38MHz	高速模式 (SMC0 位 =0)	
0	1	1	0	$f_{PRS}/4$	$f_W/24$			
0	1	1	1	禁止设置				
1	0	x	x	禁止设置				
1	1	0	x	$f_{PRS}/2$	$f_W/12$	4.00MHz ~ 4.19MHz	高速模式 (SMC0 位 =1)	
1	1	1	0	$f_{PRS}/4$	$f_W/12$			
1	1	1	1	禁止设置				

注 在外围硬件时钟 (f_{PRS}) 以高速内部振荡时钟 (f_{XH}) 运行 (XSEL=0) 时, 如下设置 CLX0、SMC0、CL01 和 CL00。

IICX0	IICCL0			选择时钟 (f_W)	传送时钟 (f_W/m)	可设置的选择时钟 (f_W) 的范围	运行模式
	bit0	bit3	bit1				
CLX0	SMC0	CL01	CL00				
0	0	0	0	$f_{PRS}/2$	$f_W/44$	3.8MHz ~ 4.2MHz	标准模式 (SMC0 位 =0)
0	1	0	x	$f_{PRS}/2$	$f_W/44$		高速模式 (SMC0 位 =1)

注意 在允许运行 (IIC 控制寄存器 0 (IIC0) 的 bit7 (IICE0) =1) 前, 必须先通过 CLX0、SMC0、CL01 和 CL00 决定 I²C 的传送时钟频率。如果要更改传送时钟频率, 必须将 IICE0 清“0”。

备注 1. x: 忽略

2. f_{PRS} : 外围硬件时钟频率

(7) 端口模式寄存器 6 (PM6)

此寄存器以 1 位为单位设置端口 6 的输入 / 输出。

在将 P60/SCL0 引脚用作时钟输入 / 输出并且将 P61/SDA0 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 IICE0 (IIC 控制寄存器 0 (IICC0) 的 bit7) 为“0”时，P60/SCL0 引脚和 P61/SDA0 引脚为低电平输出 (固定)，因此必须在将 IICE0 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置 PM6。

在产生复位信号后，PM6 变为“FFH”。

图 15-10 端口模式寄存器 6 (PM6) 的格式

地址: FF26H 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（SCL0）和串行数据总线引脚（SDA0）的结构如下。

(1) SCL0..... 串行时钟的输入/输出引脚

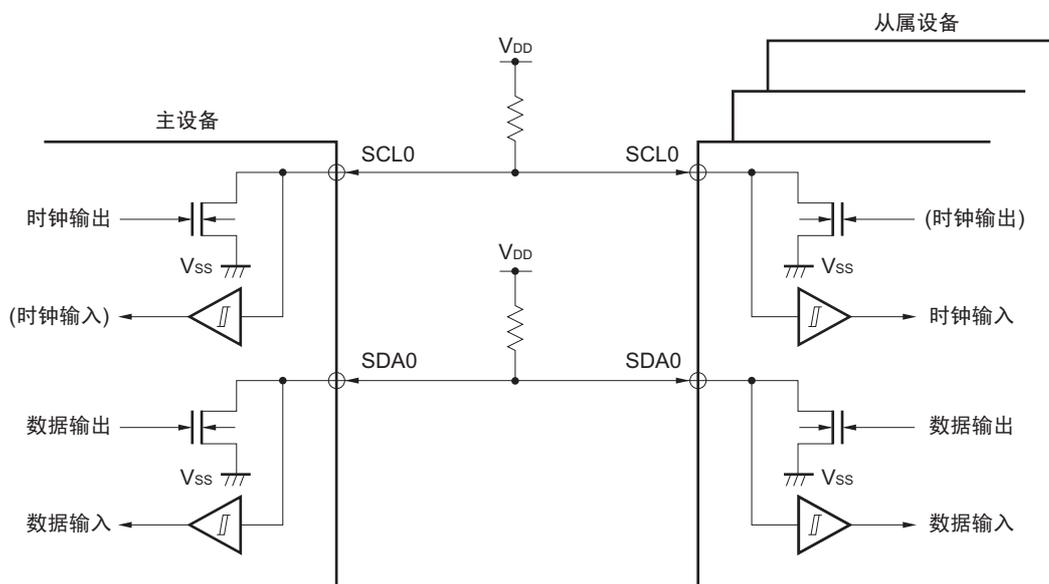
主控设备和从属设备的输出都为N沟道漏极开路输出，输入都为施密特输入。

(2) SDA0..... 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 15-11 引脚结构图

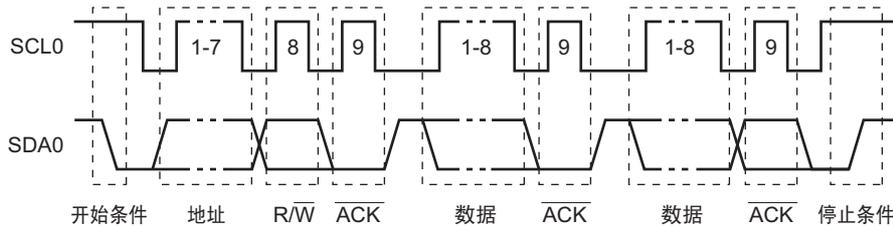


15.5 I²C 总线的定义和控制方法

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 15-12 所示。

图 15-12 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。

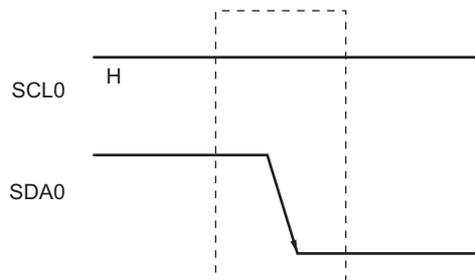
主控设备和从属设备都能生成应答（ACK）（在一般情况下，接收方输出 8 位数据）。

主控设备连续输出串行时钟（SCL0）。但是，从属设备能延长 SCL0 的低电平期间并且插入等待。

15.5.1 开始条件

在 SCL0 引脚为高电平时，如果 SDA0 引脚从高电平变为低电平，就生成开始条件。SCL0 引脚和 SDA0 引脚的开始条件是在主控设备对从属设备开始串行传送时生成的信号。从属设备能检测到开始条件。

图 15-13 开始条件



在检测到停止条件（SPD0: IIC 状态寄存器 0 (IICS0) 的 bit0=1) 的状态下，如果将 IIC 控制寄存器 0 (IICC0) 的 bit1 (STT0) 置“1”，就输出开始条件。如果检测到开始条件，就将 IICS0 寄存器的 bit1 (STD0) 置“1”。

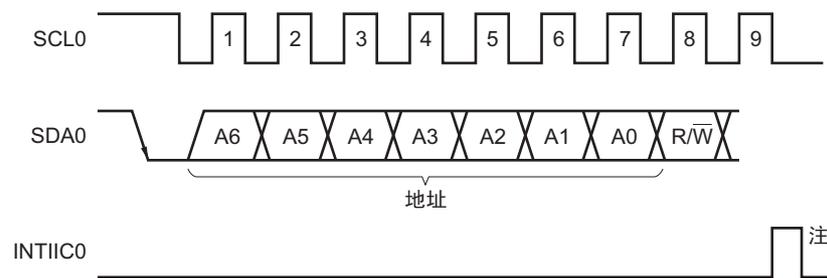
15.5.2 地址

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 0（SVA0）的内容相同。此时，如果 7 位数据和 SVA0 寄存器的值相同，该从属设备就被选中，在 7 位数据生成开始条件或者停止条件前，与主控设备进行通信。

图 15-14 地址



注 如果在从属设备运行时接收到本地站地址或者扩展码以外的数据，就不产生 INTIIC0。

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 IIC 移位寄存器 0（IIC0），就输出地址。接收到的地址被写到 IIC0 寄存器。

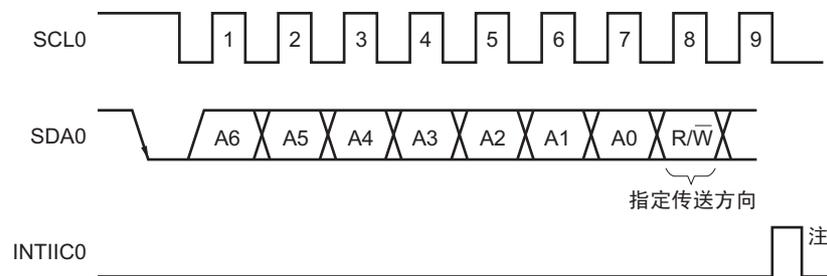
从属地址分配在 IIC0 寄存器的高 7 位。

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 15-15 传送方向的指定



注 如果在从属设备运行时接收到本地站地址或者扩展码以外的数据，就不产生 INTIIC0。

15.5.4 应答 ($\overline{\text{ACK}}$)

能通过应答 ($\overline{\text{ACK}}$) 确认发送方和接收方的串行数据状态。

接收方在每次接收到 8 位数据时返回应答。

通常，发送方在发送 8 位数据后接收应答。当接收方返回应答时，认为已正常接收，继续处理。能通过 IIC 状态寄存器 0 (IICS0) 的 bit2 (ACKD0) 确认应答的检测。

在主控设备为接收状态下接收到最后的数据时，不返回应答而生成停止条件。在从属设备接收数据后不返回应答时，主控设备输出停止条件或者重新开始条件，中止传送。不返回应答的原因如下：

- ① 没有正常接收。
- ② 已结束最后数据的接收。
- ③ 不存在地址指定的接收方。

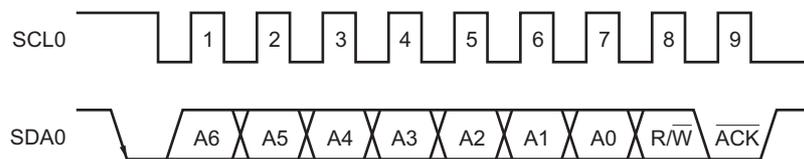
接收方在第 9 个时钟将 SDA0 线置为低电平，生成应答（正常接收）。

通过将 IIC 控制寄存器 0 (IICC0) 的 bit2 (ACKE0) 置“1”，变为能够自动生成应答的状态。能通过 7 位地址信息后续的第 8 位数据设定 IICS0 寄存器的 bit3 (TRC0)。在接收 (TRC0=0) 的情况下，通常必须将 ACEK0 位置“1”。

在从属接收运行过程中 (TRC0=0) 不能接收数据或者不需要下一个数据时，必须将 ACEK0 位清“0”，通知主控方不能接收数据。

在从属接收运行过程中 (TRC0=0) 不需要下一个数据时，为了不生成应答，必须将 ACEK0 位清“0”，通知从属发送方数据的结束（停止发送）。

图 15-16 应答



当接收到本地站的地址时，与 ACEK0 位的值无关，自动生成应答；当接收到非本地站的地址时，不生成应答 (NACK)。

通过事先将 ACEK0 位置“1”，在接收到扩展码时生成应答。

接收数据时的应答生成方法因等待时序的设定而不同，如下所示。

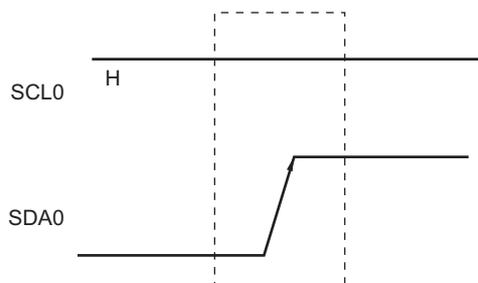
- 当选择 8 个时钟等待时 (IICC0 寄存器的 bit3 (WTIM0) = 0)：通过在解除等待前将 ACEK0 位置“1”，与 SCL0 引脚的第 8 个时钟下降同步生成应答。
- 当选择 9 个时钟等待时 (IICC0 寄存器的 bit3 (WTIM0) = 1)：通过事先将 ACEK0 位置“1”，生成应答。

15.5.5 停止条件

在 SCL0 引脚为高电平时，如果 SDA0 引脚从低电平变为高电平，就生成停止条件。

停止条件是在主控设备结束对从属设备的串行传送时生成的信号。从属设备能检测到停止条件。

图 15-17 停止条件



如果将 IIC 控制寄存器 0 (IICC0) 的 bit0 (SPT0) 置“1”，就生成停止条件。如果检测到停止条件，就将 IIC 状态寄存器 0 (IICS0) 的 bit0 (SPD0) 置“1”，并且在 IICC0 寄存器的 bit4 (SPIE0) 为“1”时产生 INTIIC0。

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接收 / 发送（等待状态）。

通过将 SCL0 引脚置为低电平，通知对方处于等待状态。如果主控设备和从属设备的等待状态都解除，就能开始下一次传送。

图 15-18 等待 (1/2)

- (1) 主控设备为 9 个时钟等待，从属设备为 8 个时钟等待的情况
(主控设备：发送，从属设备：接收，ACKE0=1)

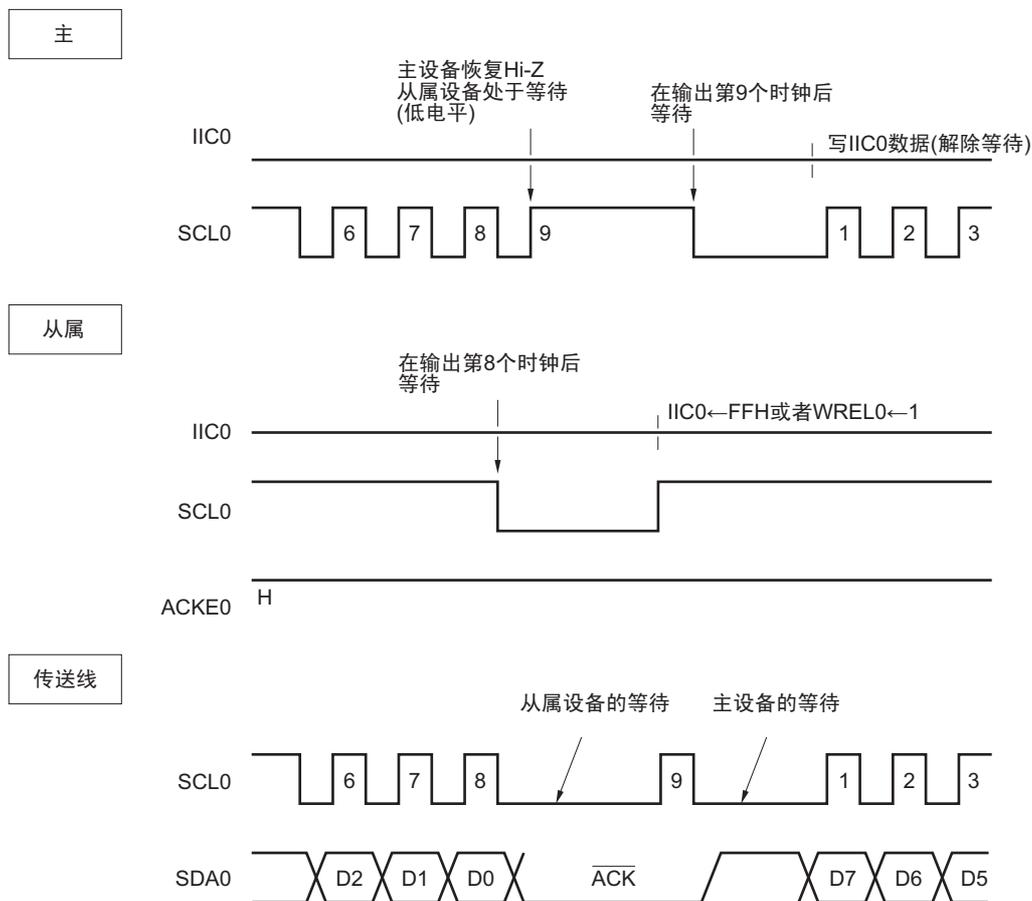
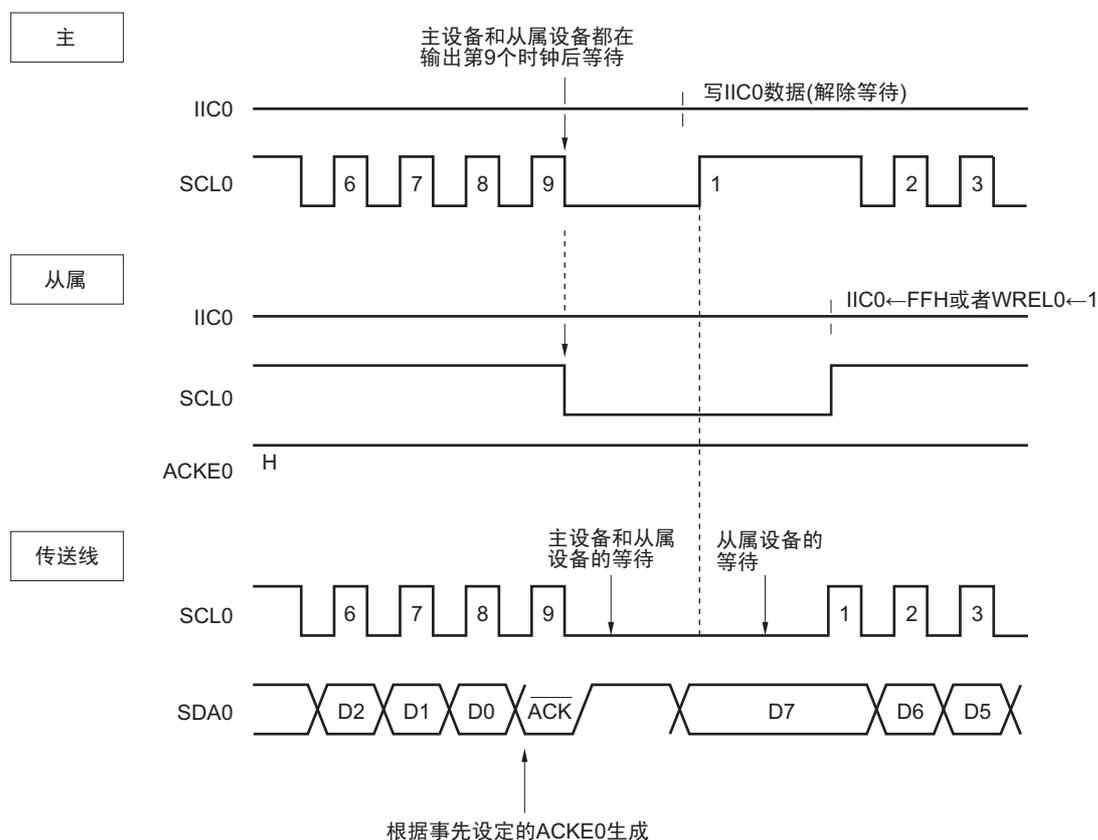


图 15-18 等待 (2/2)

- (2) 主控设备和从属设备都为 9 个时钟等待的情况
(主控设备: 发送, 从属设备: 接收, ACKE0=1)



备注 ACKE0: IIC 控制寄存器 0 (IICC0) 的 bit2
WRELO: IIC 控制寄存器 0 (IICC0) 的 bit5

通过设定 IIC 控制寄存器 0 (IICC0) 的 bit3 (WTIM0) 自动产生等待。

通常, 在接收方, 如果 IICC0 寄存器的 bit5 (WRELO) 为“1” 或者给 IIC 移位寄存器 0 (IIC0) 写“FFH”, 就解除等待; 在发送方, 如果给 IIC0 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 IICC0 寄存器的 bit1 (STT0) 置“1”。
- 将 IICC0 寄存器的 bit0 (SPT0) 置“1”。

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 IIC 移位寄存器 0 (IIC0) 写数据。
- 设置 IIC 控制寄存器 0 (IICC0) 的 bit5 (WREL0) (解除等待)。
- 设置 IICC0 寄存器的 bit1 (STT0) (生成开始条件) 注。
- 设置 IICC0 寄存器的 bit0 (SPT0) (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 IIC0 寄存器写数据。

要在解除等待后接收数据或者结束发送数据，就必须将 IIC 控制寄存器 0 (IICC0) 的 bit5 (WREL0) 置“1”。

要在解除等待后生成重新开始条件，就必须将 IICC0 寄存器的 bit1 (STT0) 置“1”。

要在解除等待后生成停止条件，就必须将 IICC0 寄存器的 bit0 (SPT0) 置“1”。

对于一次等待只能执行一次解除处理。

例如，如果在通过将 WREL0 位置“1”来解除等待后给 IIC0 寄存器写数据，SDA0 线的变化时序与 IIC0 寄存器的写时序就可能发生冲突，导致将错误的值输出到 SDA0 线。

除了这些处理以外，在中途中止通信的情况下，如果将 IICE0 位清“0”，就停止通信，因此能解除等待。

在 I²C 总线状态因噪声而被死锁的情况下，如果将 IICC0 寄存器的 bit6 (LREL0) 置“1”，就退出通信，因此能解除等待。

15.5.8 中断请求（INTIIC0）的产生时序和等待控制

通过设定IIC控制寄存器0（IICC0）的bit3（WTIM0），在表15-3所示的时序产生INTIIC0并且进行等待控制。

表 15-3 INTIIC0 的产生时序和等待控制

WTIM0	从属设备			主控设备		
	地址	数据接收	数据发送	地址	数据接收	数据发送
0	9 注 1、2	8 注 2	8 注 2	9	8	8
1	9 注 1、2	9 注 2	9 注 2	9	9	9

注 1. 只有在接收的地址和从属地址寄存器 0（SVA0）的设定值相同时，从属设备才在第 9 个时钟的下降沿产生 INTIIC0 信号并且进入等待状态。

此时，与 IICC0 寄存器 bit2（ACKE0）的设定无关，生成应答。接收到扩展码的从属设备在第 8 个时钟的下降沿产生 INTIIC0。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 INTIIC0，但是不进入等待状态。

2. 如果接收的地址和从属地址寄存器 0（SVA0）的内容不同并且未接收到扩展码，就不产生 INTIIC0 并且也不进入等待状态。

备注 表中的数字表示串行时钟的时钟数。中断请求和等待控制都与串行时钟的下降沿同步。

(1) 地址的发送和接收

- 从属设备：与 WTIM0 位无关，通过上述注 1 和注 2 的条件决定中断和等待的时序。
- 主控设备：与 WTIM0 位无关，在第 9 个时钟的下降沿产生中断和等待的时序。

(2) 数据的接收

- 主控设备/从属设备：通过 WTIM0 位决定中断和等待的时序。

(3) 数据的发送

- 主控设备/从属设备：通过 WTIM0 位决定中断和等待的时序。

(4) 等待的解除方法

等待的解除方法有以下 4 种：

- 给 IIC 移位寄存器 0 (IIC0) 写数据。
- 设置 IIC 控制寄存器 0 (IICC0) 的 bit5 (WREL0) (解除等待)。
- 设置 IICC0 寄存器的 bit1 (STT0) 置位 (生成开始条件) 注。
- 设置 IICC0 寄存器的 bit0 (SPT0) 置位 (生成停止条件) 注。

注 只限于主控设备。

当选择 8 个时钟等待 (WTIM0=0) 时，需要在解除等待前决定是否生成应答。

(5) 停止条件的检测

如果检测到停止条件，就产生 INTIIC0 (只限于 SPIE0=1)。

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。

通过硬件自动检测地址匹配。当主控设备送来的从属地址和从属地址寄存器 0 (SVA0) 的设定地址相同或者接收到扩展码时，产生 INTIIC0 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 (SDA0) 的状态被取到发送器件的 IIC 移位寄存器 0 (IIC0)，所以能通过将开始发送前和发送结束的 IIC0 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。

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（EXC0）置“1”，并且在第8个时钟的下降沿产生中断请求（INTIIC0）。
不影响保存在从属地址寄存器0（SVA0）的本地站地址。
- (2) 当SVA0寄存器的设定值为“11110xx0”时，如果通过10位地址传送从主控设备发送“11110xx0”，就发生以下的置位。但是，在第8个时钟的下降沿产生中断请求（INTIIC0）。
 - 高4位数据相同：EXC0=1
 - 7位数据相同：COI0=1

备注 EXC0：IIC 状态寄存器 0（IICS0）的 bit5

COI0：IIC 状态寄存器 0（IICS0）的 bit4

- (3) 中断请求发生后的处理因扩展码的后续数据而不同，通过软件进行处理。
如果在从属设备运行时接收到扩展码，即使地址不同也在参加通信。
例如，在接收到扩展码后不想作为从属设备运行时，必须将IIC控制寄存器0（IICC0）的bit6（LREL0）置“1”，进入下次通信的待机状态。

表 15-4 主要扩展码的位定义

从属地址	R/W 位	说明
0 0 0 0 0 0 0	0	全呼地址
1 1 1 1 0 x x	0	10 位从属地址的指定（地址认证时）
1 1 1 1 0 x x	1	10 位从属地址的指定（在地址相同后发行读命令时）

备注 有关上述以外的扩展码，请参照 NXP 公司发行的 I²C 总线规格书。

15.5.12 仲裁

当多个主控设备同时生成开始条件时（在 STD0 位变为“1”前将 STT0 位置“1”的情况），边调整时钟边进行主控设备的通信，直到数据不同为止。此运行称为仲裁。

在仲裁失败时，仲裁失败的主控设备将 IIC 状态寄存器 0（IICS0）的仲裁失败标志（ALD0）置“1”，并且将 SCL0 线和 SDA0 线都置为高阻抗状态，以释放总线。

在发生下一次中断请求时（例如：在第 8 或者第 9 个时钟检测到停止条件），用软件通过 ALD0 位为“1”来检测仲裁的失败。

有关中断请求的产生时序，请参照“15.5.17 I²C 中断请求（IITIC0）的产生时序”。

备注 STD0: IIC 状态寄存器 0（IICS0）的 bit1

STT0: IIC 控制寄存器 0（IICC0）的 bit1

图 15-19 仲裁时序例子

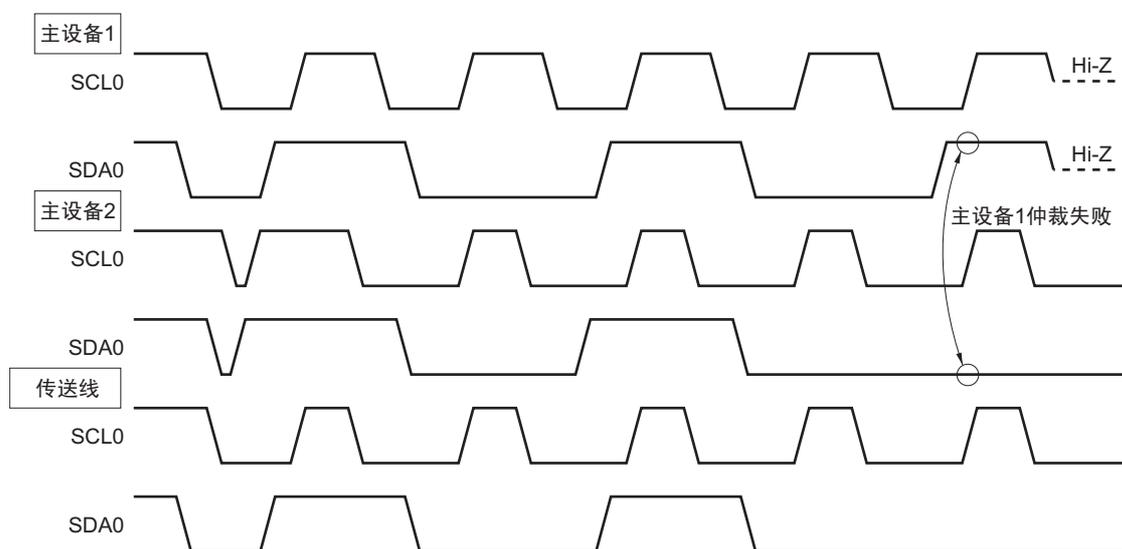


表 15-5 发生仲裁时的状态和中断请求的产生时序

发生仲裁时的状态	中断请求的产生时序
地址发送过程中	在字节传送后的第 8 或者第 9 个时钟的下降沿注 1
发送地址后的读写信息	
扩展码发送过程中	
发送扩展码后的读写信息	
数据发送过程中	
发送数据后的应答传送过程中	
在数据传送过程中检测到重新开始条件。	
在数据传送过程中检测到停止条件。	在生成停止条件时 (SPIE0=1) 注 2
要生成重新开始条件, 但是数据为低电平。	在字节传送后的第 8 或者第 9 个时钟的下降沿注 1
要生成重新开始条件, 但是检测到停止条件。	在生成停止条件时 (SPIE0=1) 注 2
要生成停止条件, 但是数据为低电平。	在字节传送后的第 8 或者第 9 个时钟的下降沿注 1
要生成重新开始条件, 但是 SCL0 为低电平。	

注 1. 当 WTIM0 位 (IIC 控制寄存器 0 (IICC0) 的 bit3) 为“1”时, 在第 9 个时钟的下降沿产生中断请求。当 WTIM0 位为“0”并且接收到扩展码的从属地址时, 在第 8 个时钟的下降沿产生中断请求。

2. 当有可能发生仲裁时, 必须在主控设备时将 SPIE0 位置“1”。

备注 SPIE0: IIC 控制寄存器 0 (IICC0) 的 bit4

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（INTIIC0）的功能。

在地址不同的情况下不产生不需要的 INTIIC0 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。

如果检测到停止条件，与唤醒功能无关，根据 IIC 控制寄存器 0（IICC0）的 bit4（SPIE0）的设置，决定允许 / 禁止产生中断请求。

15.5.14 通信预约

(1) 允许通信预约功能的情况（IIC 标志寄存器 0（IICF0）的 bit0（IICRSV）=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（不返回应答而通过将 IIC 控制寄存器 0（IICC0）的 bit6（LREL0）置“1”释放了总线）。

如果在不加入总线的状态下将 IICC0 寄存器的 bit1（STT0）置“1”，就在释放总线后（检测到停止条件）自动生成开始条件，进入等待状态。

将 IICC0 寄存器的 bit4（SPIE0）置“1”，在通过产生的中断请求信号（INTIIC0）检测到总线的释放（检测到停止条件）后，如果给 IIC 移位寄存器 0（IIC0）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 IIC0 寄存器写的的数据无效。

当将 STT0 位置“1”时，根据总线状态决定是作为开始条件运行还是作为通信预约运行。

- 总线处于释放状态时.....生成开始条件
- 总线未处于释放状态（待机状态）时.....通信预约

在将 STT0 位置“1”并且经过等待时间后，通过 MST50 位（IIC 状态寄存器 0（IICS0）的 bit7）确认是否作为通信预约运行。

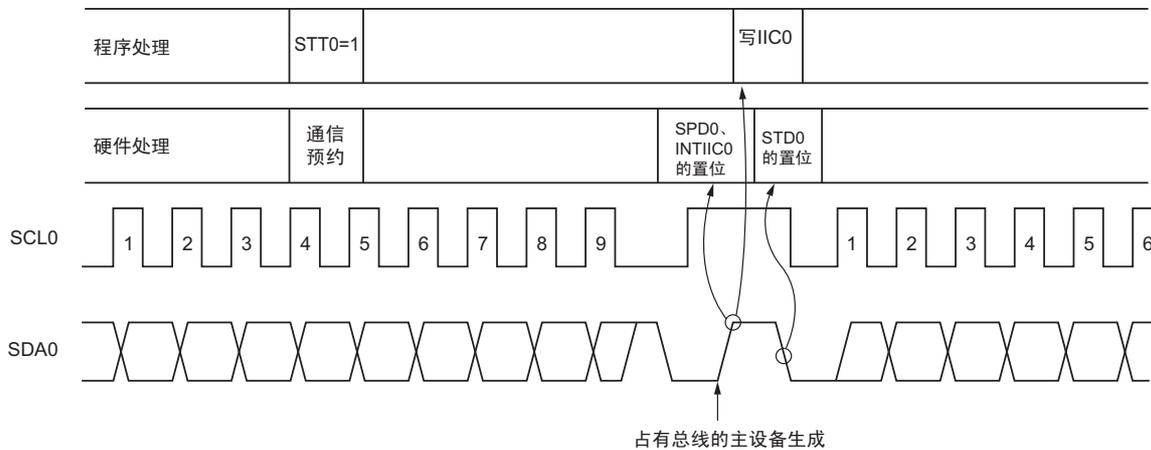
必须通过软件确保表 15-6 所示的等待时间。

表 15-6 等待时间

CLX0	SMC0	CL01	CL00	等待时间
0	0	0	0	46 个时钟
0	0	0	1	86 个时钟
0	0	1	0	172 个时钟
0	0	1	1	34 个时钟
0	1	0	0	30 个时钟
0	1	0	1	
0	1	1	0	60 个时钟
0	1	1	1	12 个时钟
1	1	0	0	18 个时钟
1	1	0	1	
1	1	1	0	36

通信预约的时序如图 15-20 所示。

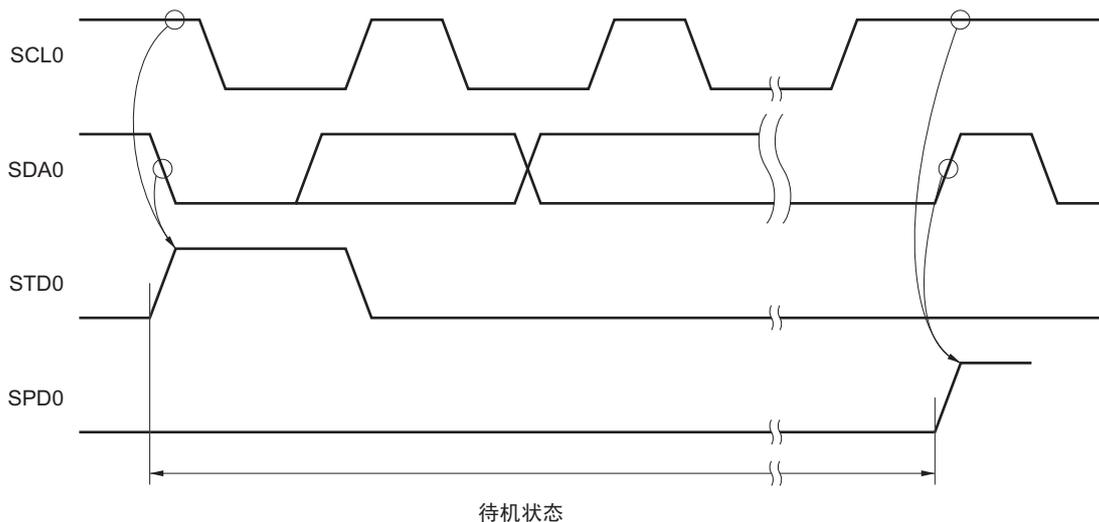
图 15-20 通信预约的时序



备注 IIC0: IIC 移位寄存器 0
 STT0: IIC 控制寄存器 0 (IICC0) 的 bit1
 STD0: IIC 状态寄存器 0 (IICS0) 的 bit1
 SPD0: IIC 状态寄存器 0 (IICS0) 的 bit0

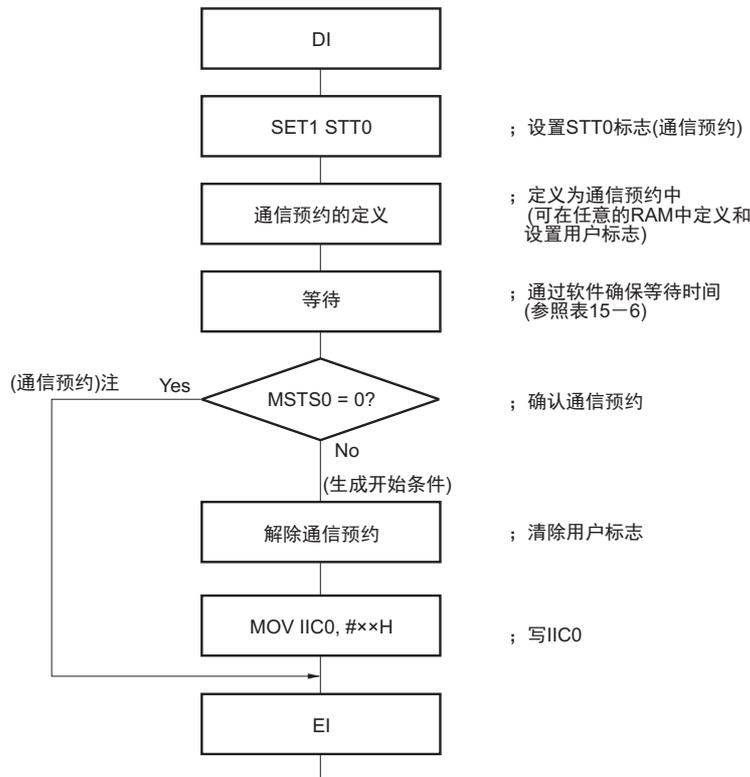
通过图 15-21 所示的时序接受通信预约。在 IIC 状态寄存器 0 (IICS0) 的 bit1 (STD0) 变为“1”后并且在检测到停止条件前，将 IIC 控制寄存器 0 (IICC0) 的 bit1 (STT0) 置“1”进行通信预约。

图 15-21 通信预约的接受时序



通信预约步骤如图 15-22 所示。

图 15-22 通信预约步骤



注 在通信预约运行时，通过停止条件中断请求写 IIC 移位寄存器 0（IIC0）。

备注 STT0: IIC 控制寄存器 0（IICC0）的 bit1
 MSTS0: IIC 状态寄存器 0（IICS0）的 bit7
 IIC0: IIC 移位寄存器 0

(2) 禁止通信预约功能的情况（IIC 标志寄存器 0（IICF0）的 bit0（IICRSV）=1）

在总线通信过程中，如果不参加此通信的状态下将 IIC 控制寄存器 0（IICC0）的 bit1（STT0）置“1”，就拒绝此请求而且不生成开始条件。此时的不加入总线包括以下 2 种状态：

- 在仲裁结果既不是主控设备也不是从属设备时
- 在接收到扩展码后不作为从属设备运行时（不返回应答而将 IICC0 寄存器的 bit6（LRELO）置“1”，退出通信后释放了总线）。

通过 STCF 标志（IICF0 寄存器的 bit7）来确认是生成了开始条件还是拒绝了请求。因为从 STT0 位为“1”到将 STCF 标志置“1”需要表 15-7 所示的时间，所以必须通过软件确保此时间。

表 15-7 等待时间

CL01	CL00	等待时间
0	0	6 个时钟
0	1	6 个时钟
1	0	12 个时钟
1	1	3 个时钟

15.5.15 其他注意事项

- (1) STCEN 位 (IIC 标志寄存器 0 (IICF0) 的 bit1) 为“0”的情况
在刚允许 I²C 运行 (IICE0=1) 后, 与实际的总线状态无关而视为通信状态 (IICBSY 标志 (IICF0 的 bit6) =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 设置 IIC 时钟选择寄存器 0 (IICCL0)。
 - ② 将 IIC 控制寄存器 0 (IICC0) 的 bit7 (IICE0) 置“1”。
 - ③ 将 IICC0 寄存器的 bit0 (SPT0) 置“1”。
- (2) STCEN 位为“1”的情况
在刚允许 I²C 运行 (IICE0=1) 后, 与实际的总线状态无关而视为释放状态 (IICBSY=0)。因此在生成第 1 个开始条件 (STT0 (IIC 控制寄存器 0 (IICC0) 的 bit1) =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 SDA0 引脚为低电平并且 SCL0 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 SDA0 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 IICC0 寄存器的 bit4 (SPIE0) 清“0”, 禁止在检测到停止条件时产生中断请求信号 (INTIIC0)。
 - ② 将 IICC0 寄存器的 bit7 (IICE0) 置“1”, 允许 I²C 运行。
 - ③ 等待检测到开始条件。
 - ④ 在返回应答前 (将 IICE0 位置“1”后的 4~80 个时钟内) 将 IICC0 寄存器的 bit6 (LREL0) 置“1”, 强制将检测置为无效。
- (4) 在允许运行 (IICE0=1) 前, 必须通过 SMC0、CL01、CL00 位 (IICL0 寄存器的 bit3、bit1、bit0) 和 CLX0 位 (IICX0 寄存器的 bit0) 决定传送时钟频率。如果要更改传送时钟频率, 必须将 IICE0 位清“0”。
- (5) 在设置 STT0、SPT0 位 (IICC0 寄存器的 bit1、bit0) 后, 禁止清“0”前的再设置。
- (6) 如果进行了通信预约, 就必须将 SPIE0 位 (IICC0 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 IIC 状态寄存器 0 (IICS0) 写通信数据, 开始传送。如果在检测到停止条件时不发生中断, 就因为开始通信时不产生中断请求而停止在等待状态。但是, 当通过软件检测 MST0 位 (IIC 状态寄存器 0 (IICS0) 的 bit7) 时, 不需要将 SPIE0 位置“1”。

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

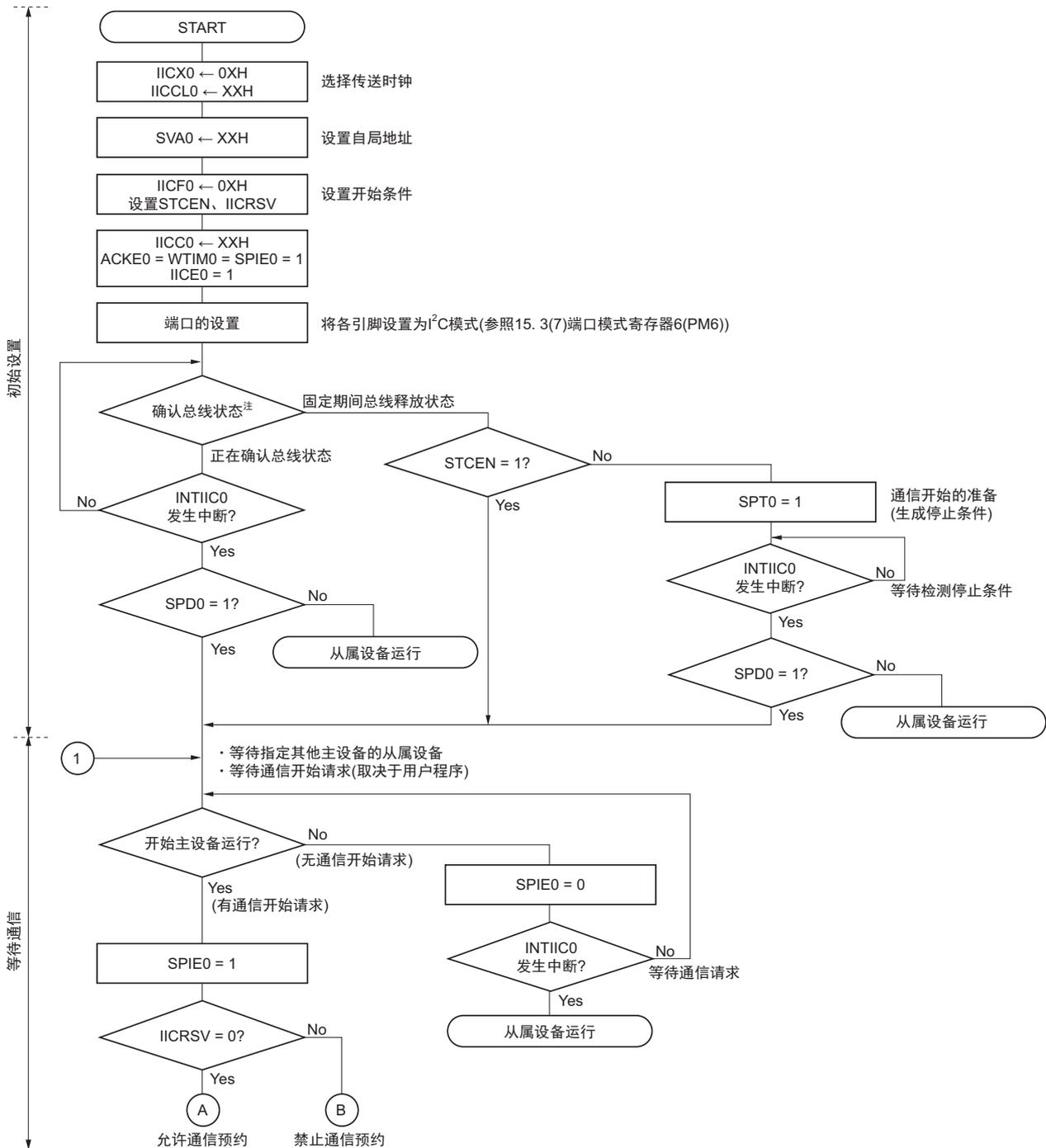
用作 I²C 总线从属设备的例子如下所示。

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 INTIIC0 中断的发生。如果发生 INTIIC0 中断，就判断通信状态并且将标志传递给主处理部。

通过检查各标志，进行所要的“通信处理”。

(2) 多主控系统的主控运行

图 15-24 多主控系统的主控运行 (1/3)



注 必须确认在一定时间内（例如，1 帧）总线为释放状态（CLD0 位 = 1、DAD0 位 = 1）。当 SDA0 引脚固定为低电平时，必须根据通信中的产品的规格，判断是否释放 I²C 总线（SCL0 引脚和 SDA0 引脚为高电平）。

图 15-24 多主控系统的主控运行 (2/3)

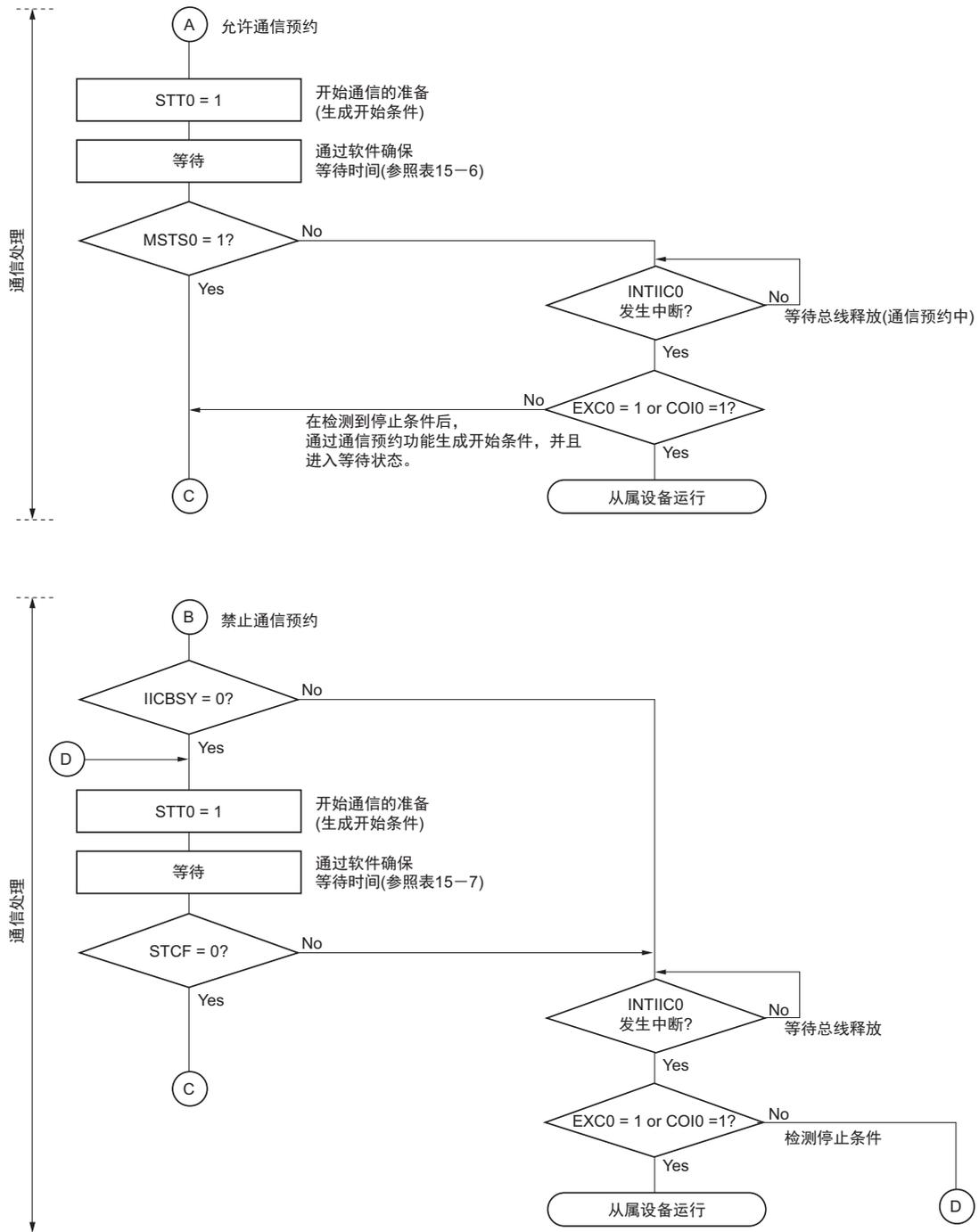
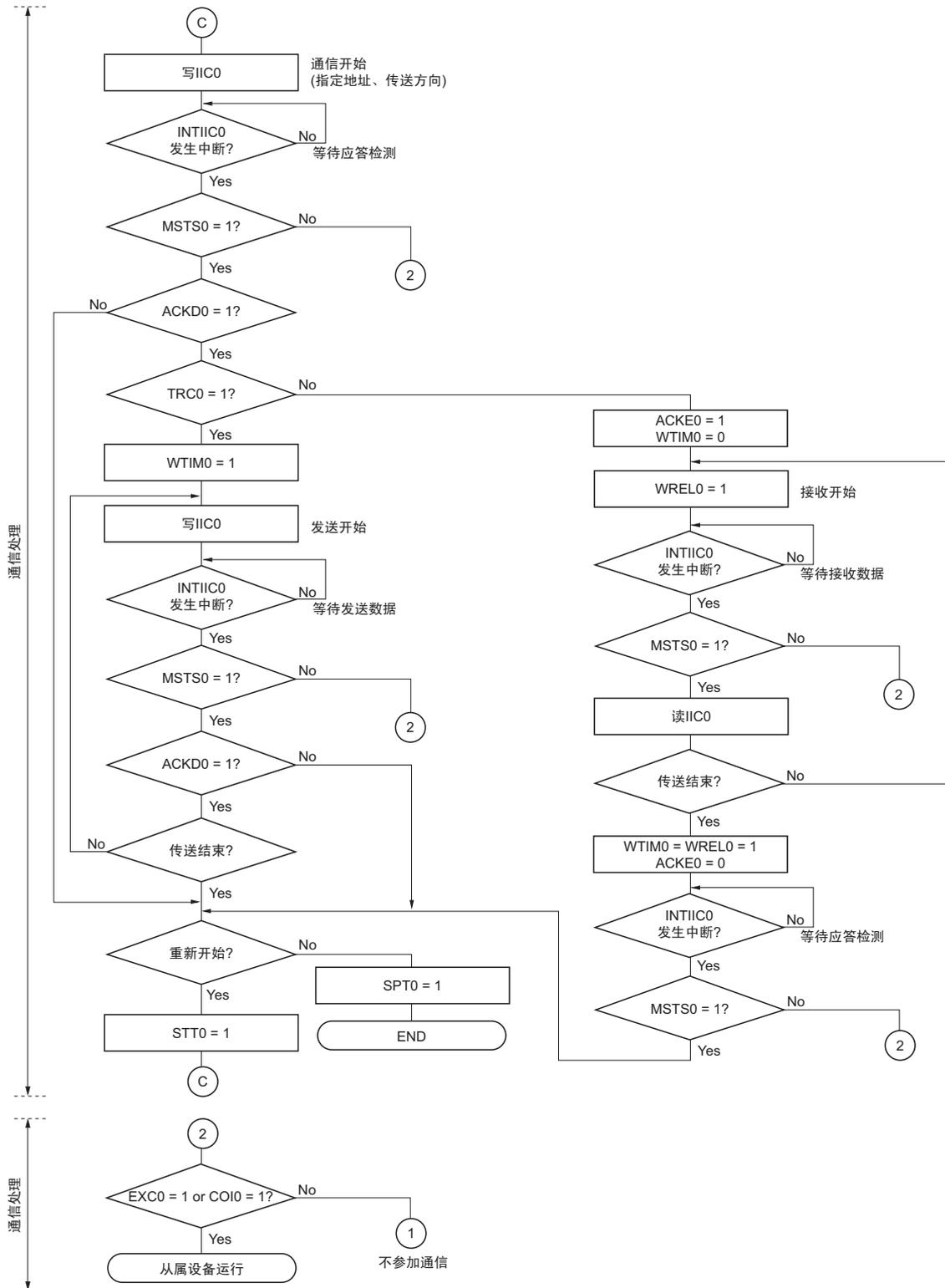


图 15-24 多主控系统的主控运行 (3/3)



备注 1. 传送和接收格式必须符合通信中的产品的规格。

2. 在多主控系统中用作主控设备的情况下，必须在每次发生 INTIIC0 中断时读 MSTS0 位，确认仲裁结果。

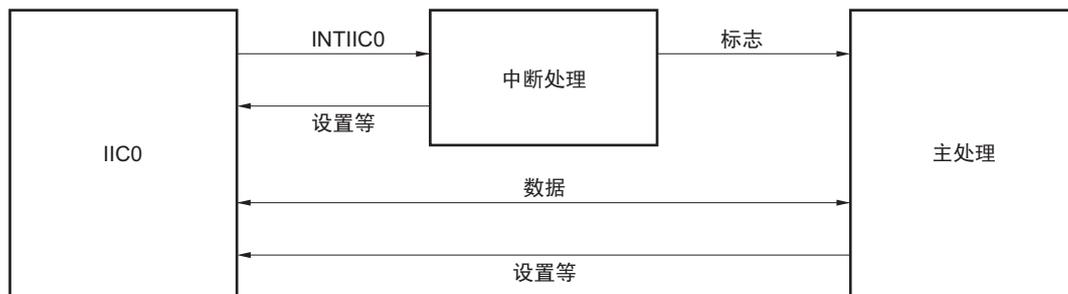
3. 在多主控系统中用作从属设备的情况下，必须在每次发生 INTIIC0 中断时通过 IICS0 寄存器和 IICF0 寄存器确认状态，决定以后的处理。

(3) 从属运行

从属运行的处理步骤如下所示。

从属运行基本上由事件驱动，因此需要通过 INTIIC0 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，INTIIC0 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 INTIIC0 将标志传递给主处理部，进行数据通信处理。

① 通信模式标志

此标志表示以下 2 种通信状态：

- 清除模式：不在进行数据通信的状态
- 通信模式：正在进行数据通信的状态（有效地址的检测～停止条件的检测，未检测到主控设备的应答，地址不同）

② 就绪标志

此标志表示可进行数据通信。在通常的数据通信中，和 INTIIC0 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 TRC0 位的值相同。

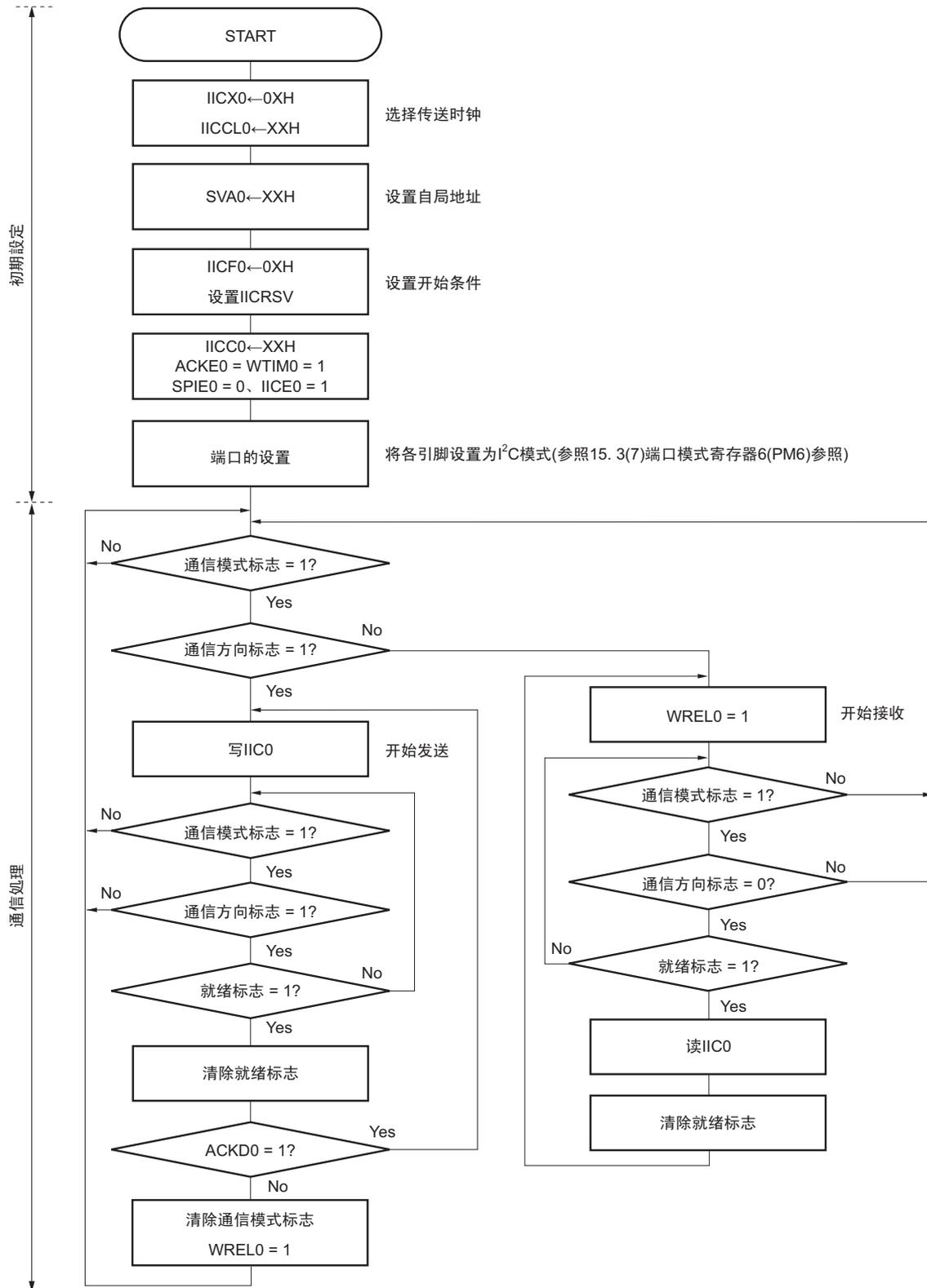
从属运行的主处理部的运行如下所示。

启动串行接口 IIC0，等待变为可通信的状态。如果变为可通信的状态，就使用通信模式标志和就绪标志进行通信（通过中断进行停止条件和开始条件的处理，在此通过标志确认状态）。

在发送时，重复发送，直到主控设备不返回应答为止。如果主控设备不返回应答，就结束通信。

在接收时，接收所需数量的数据。如果通信结束，就在下一个数据时不返回应答。此后，主控设备生成停止条件或者重新开始条件，从而退出通信状态。

图 15-25 从属运行步骤 (1)



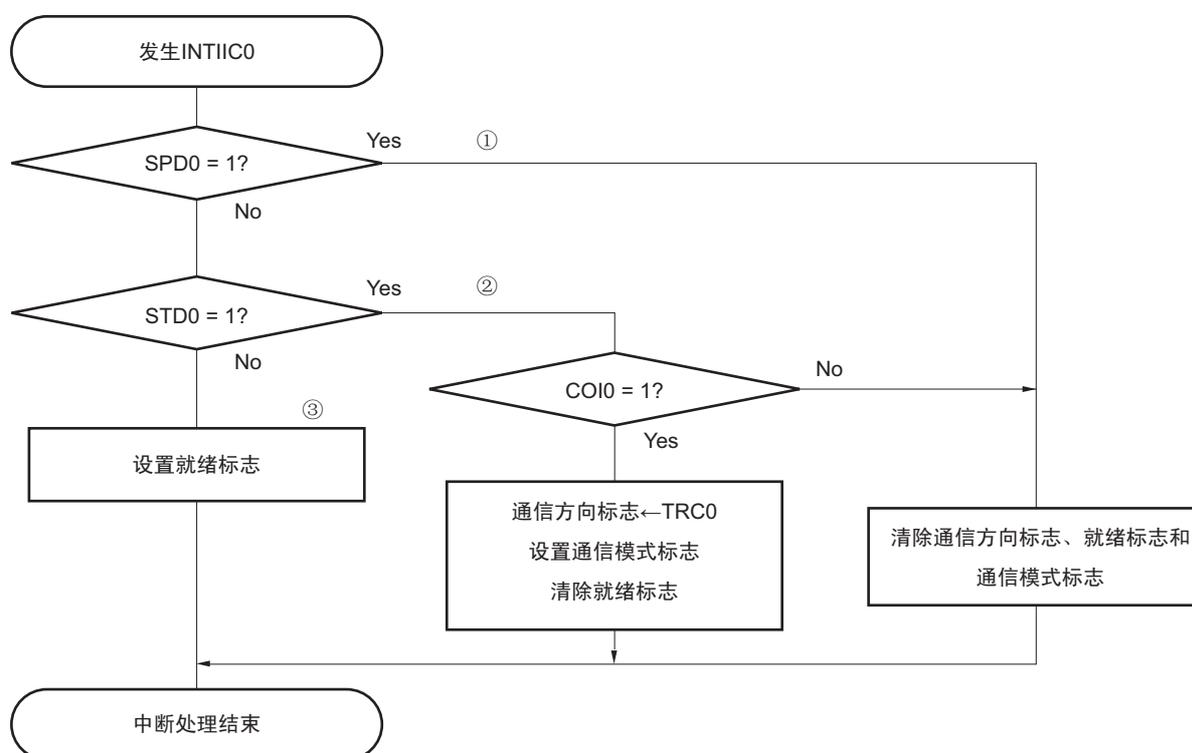
备注 传送和接收格式必须符合通信中的产品的规格。

从属设备通过 INTIIC0 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 INTIIC0 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 15-26 从属运行步骤（2）”的①~③。

图 15-26 从属运行步骤（2）



15.5.17 I²C 中断请求 (INTIIC0) 的产生时序

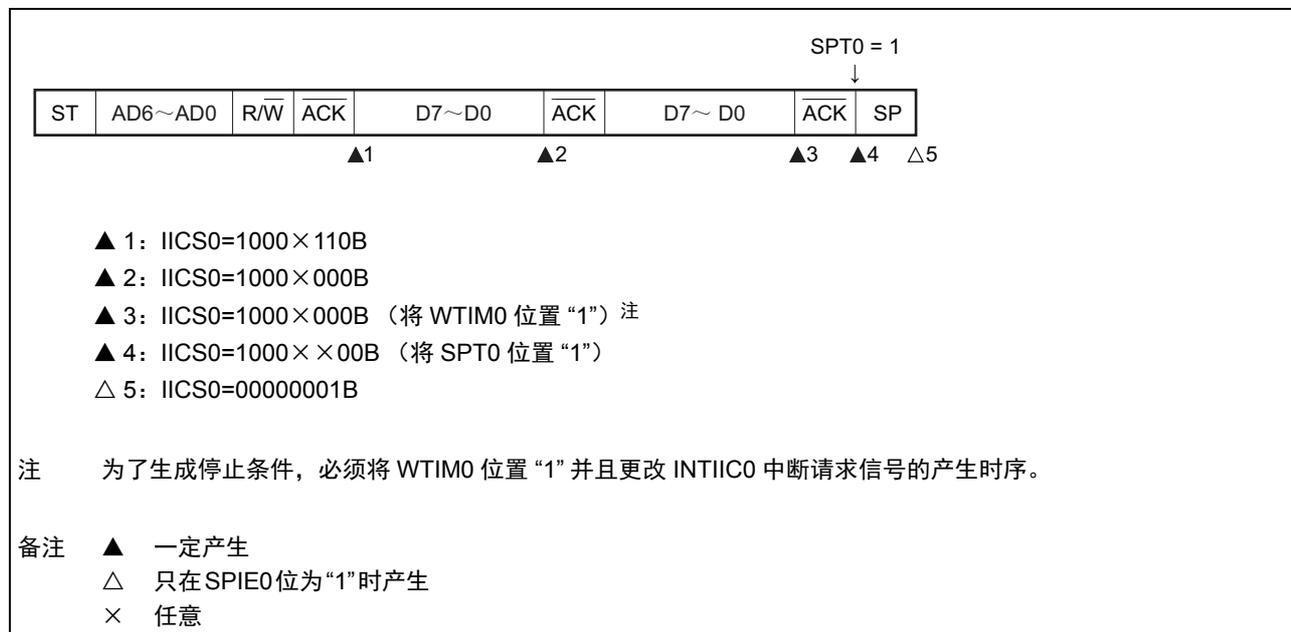
数据的发送和接收时序、INTIIC0 中断请求信号的产生时序以及产生 INTIIC0 信号时的 IICS0 寄存器的值如下所示。

备注	ST:	开始条件
	AD6~AD0:	地址
	$\overline{R/W}$:	传送方向的指定
	\overline{ACK} :	应答
	D7~D0:	数据
	SP:	停止条件

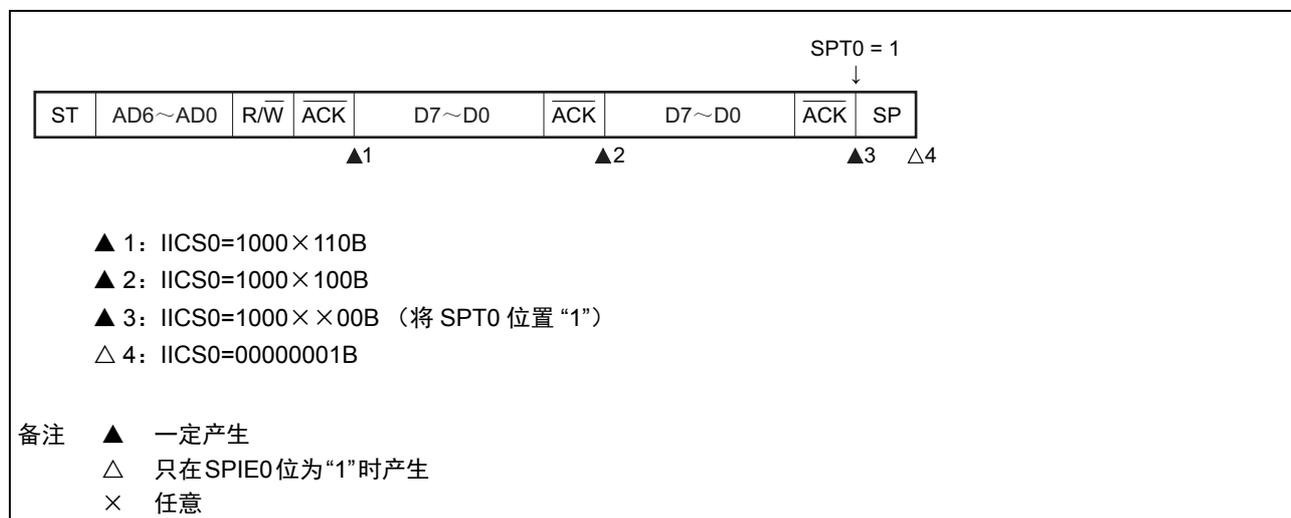
(1) 主控运行

(a) Start ~ Address ~ Data ~ Data ~ Stop (发送和接收)

(i) WTIM0=0的情况

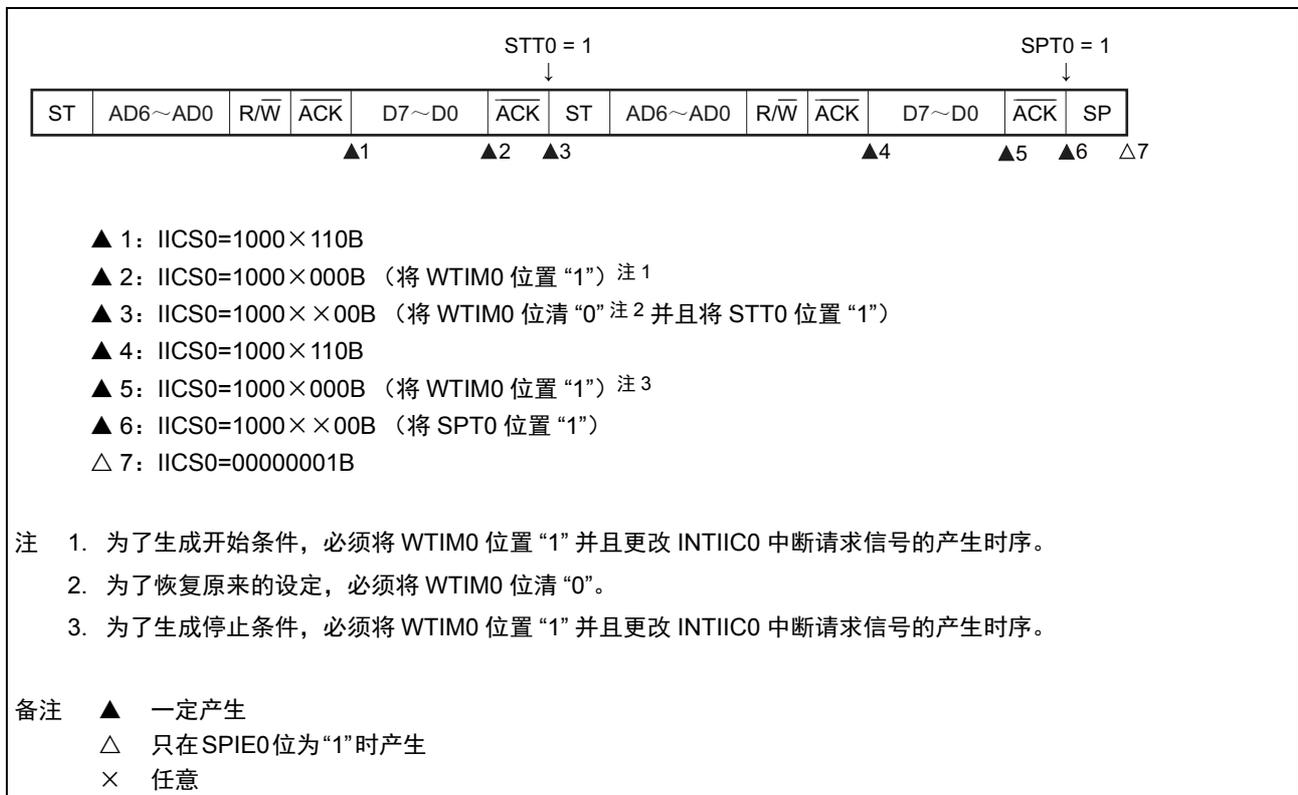


(ii) WTIM0=1的情况

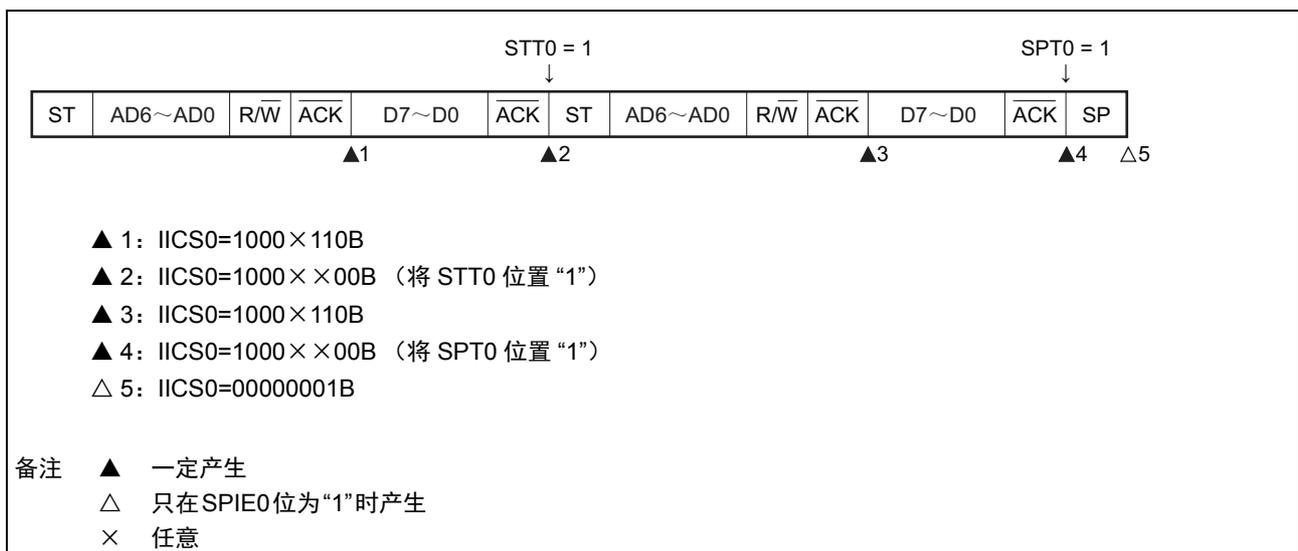


(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (重新开始)

(i) WTIM0=0的情况

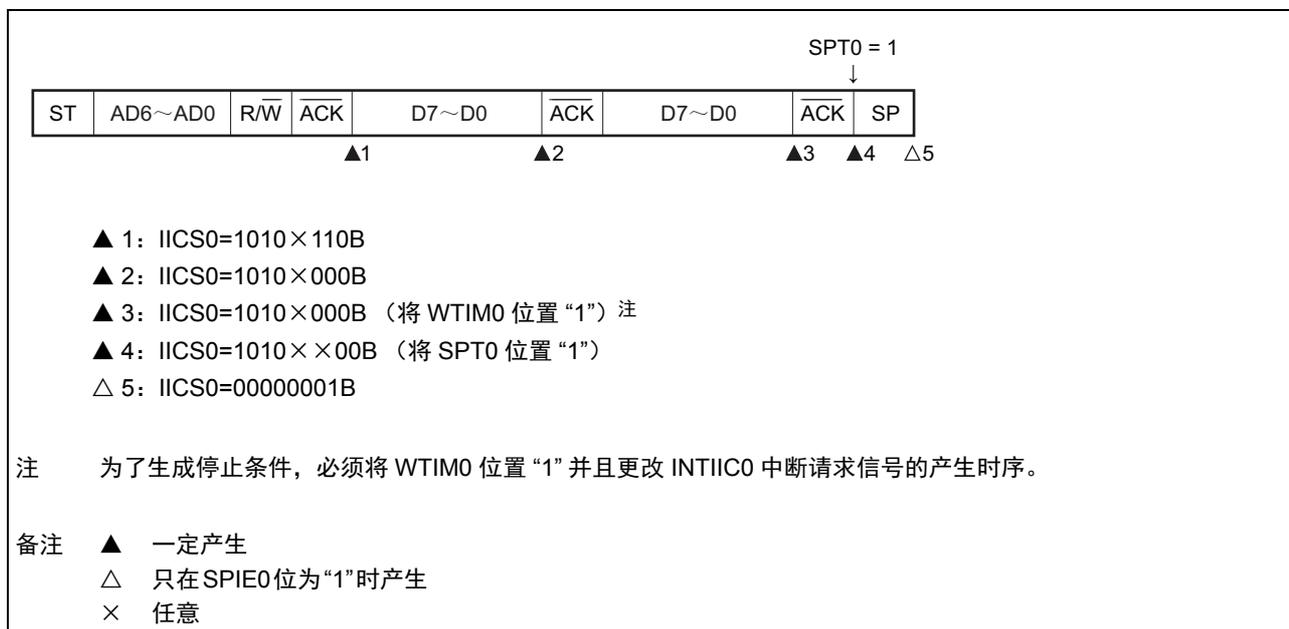


(ii) WTIM0=1的情况

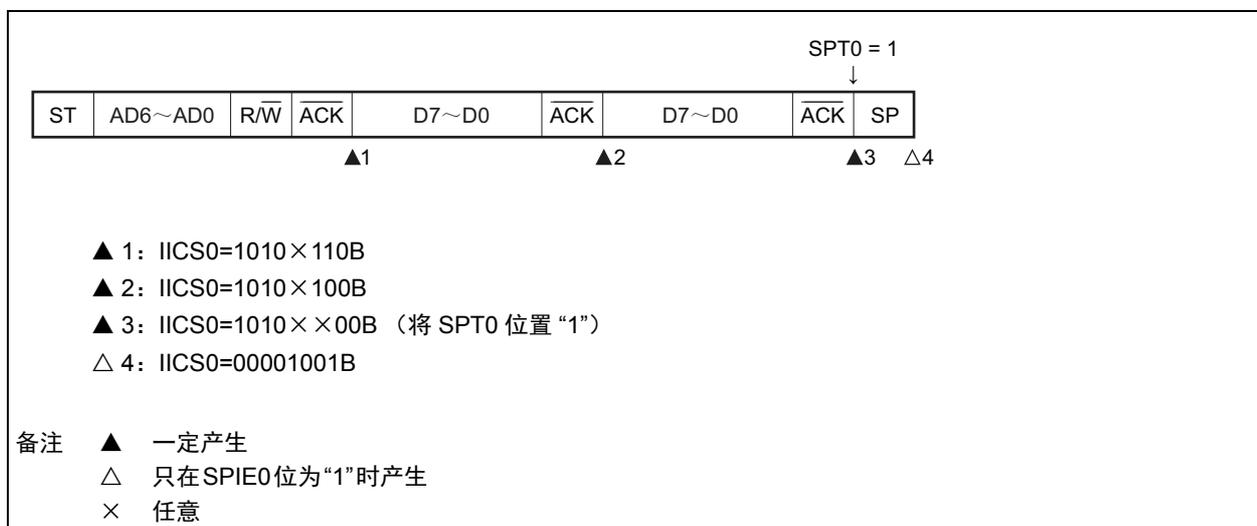


(c) Start ~ Code ~ Data ~ Data ~ Stop (发送扩展码)

(i) WTIM0=0的情况



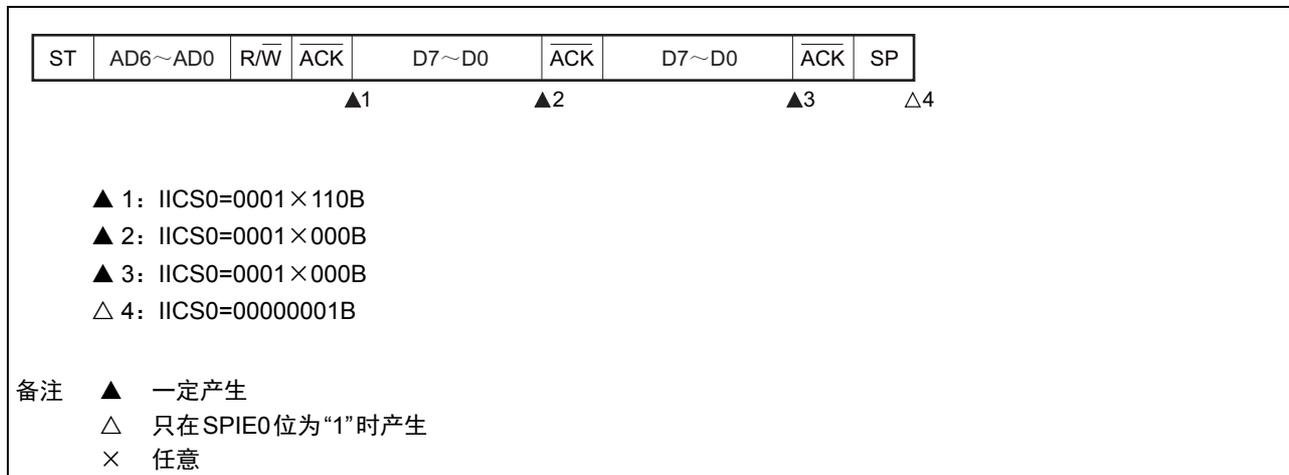
(ii) WTIM0=1的情况



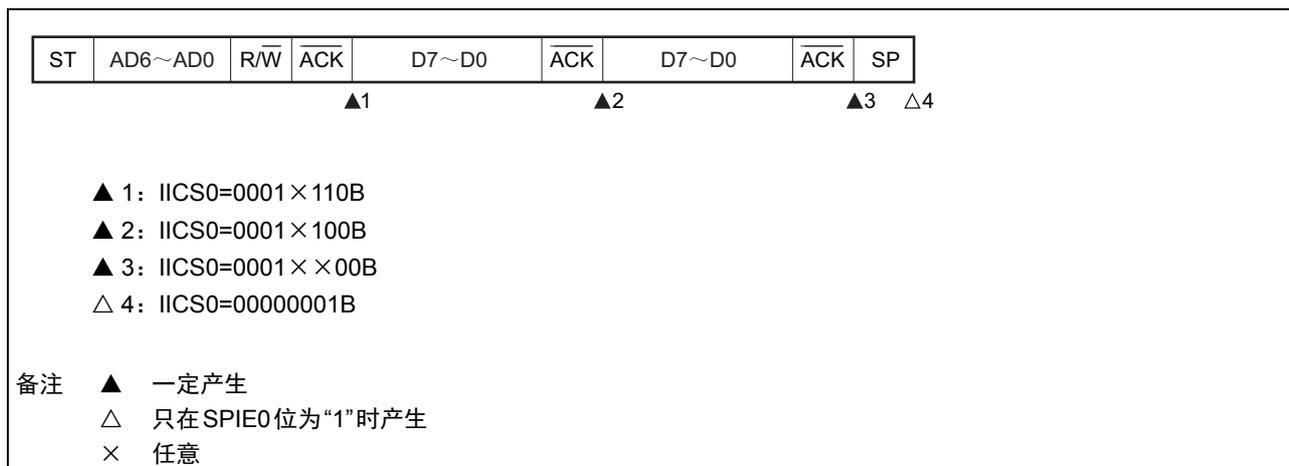
(2) 从属运行（接收从属地址的情况）

(a) Start ~ Address ~ Data ~ Data ~ Stop

(i) WTIM0=0的情况

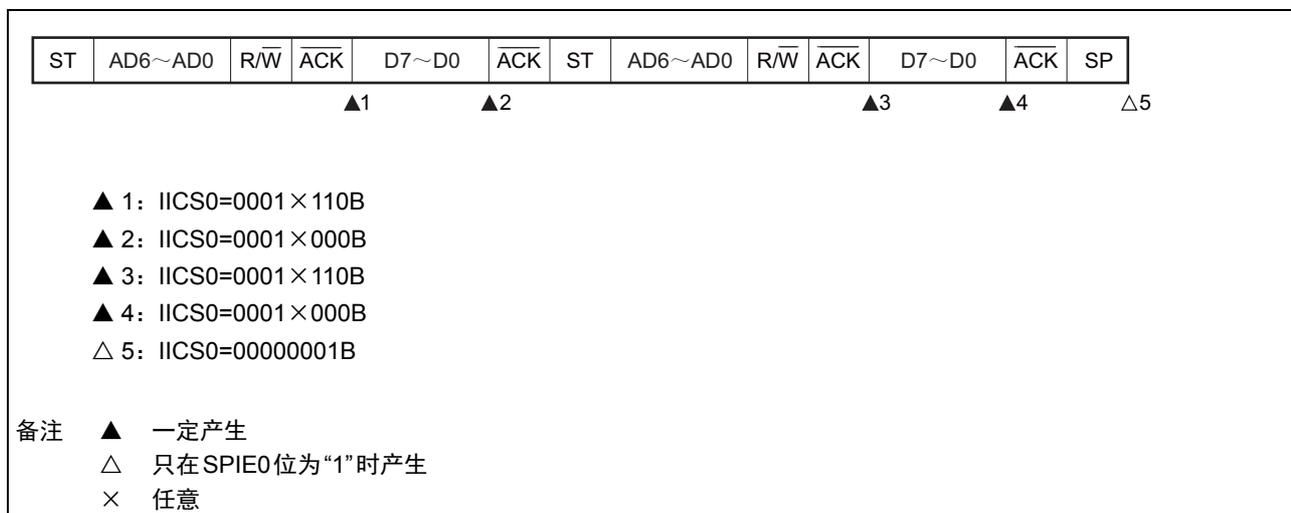


(ii) WTIM0=1的情况

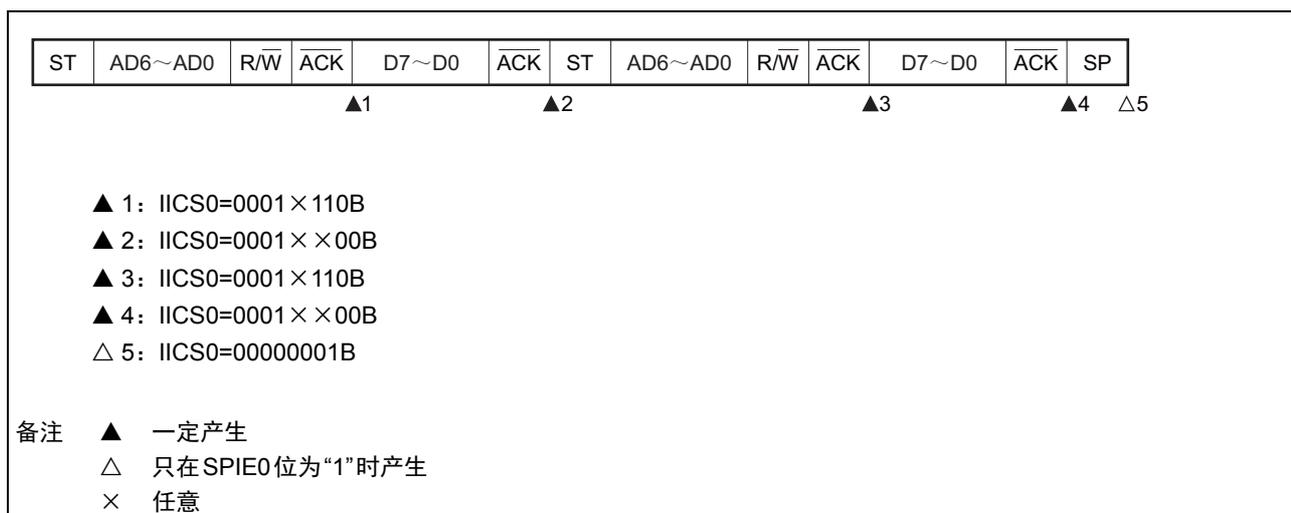


(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIM0=0的情况 (在重新开始后SVA0相同)

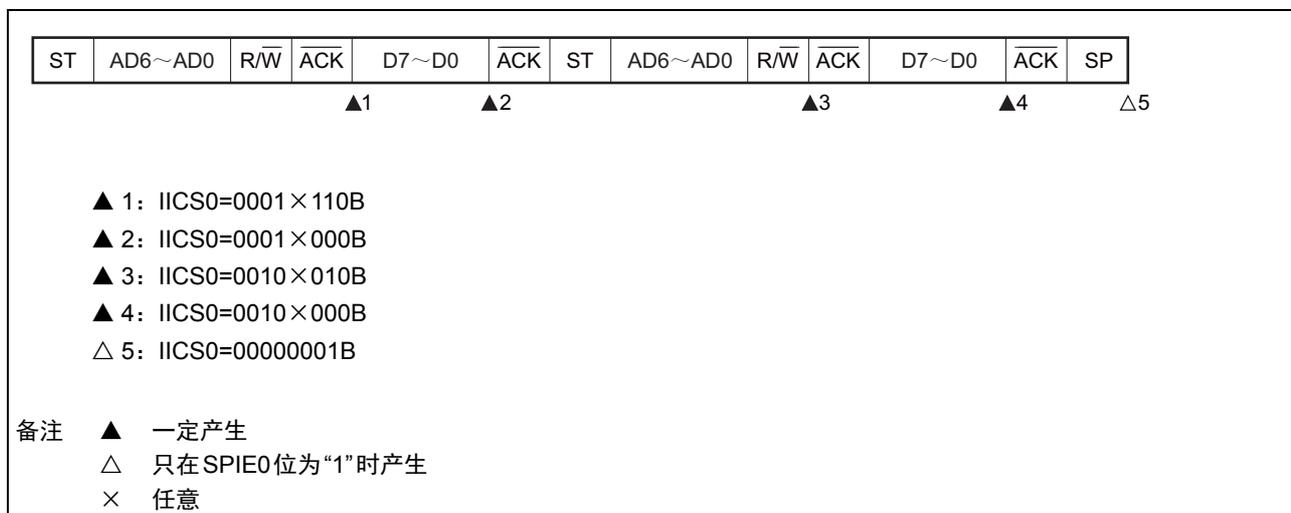


(ii) WTIM0=1的情况 (在重新开始后SVA0相同)

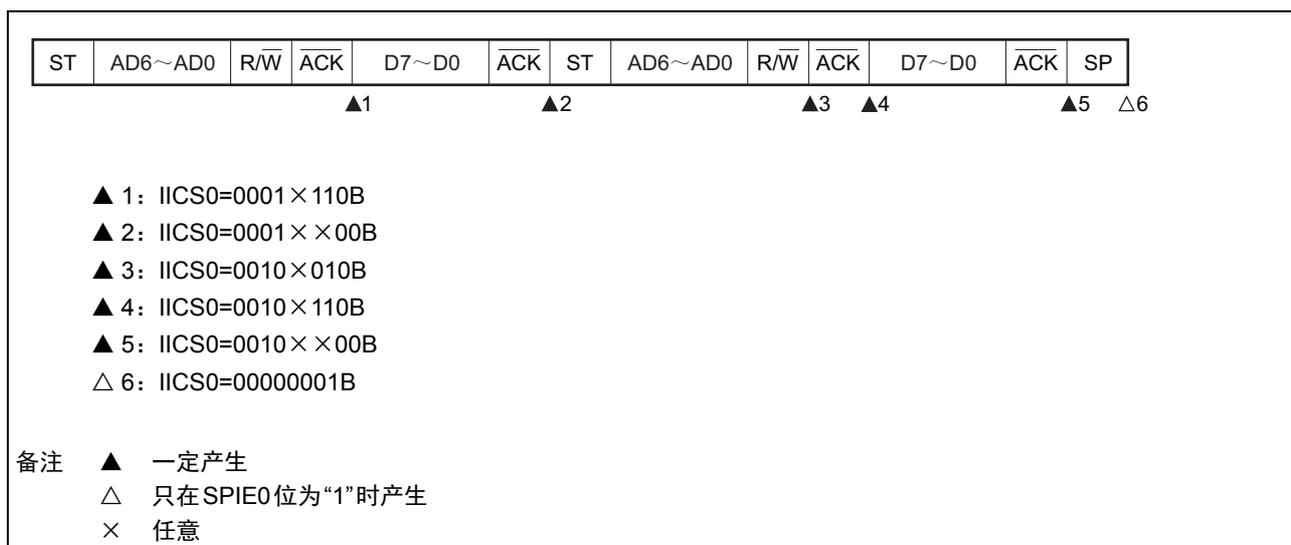


(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) WTIM0=0的情况 (在重新开始后地址不同 (扩展码))

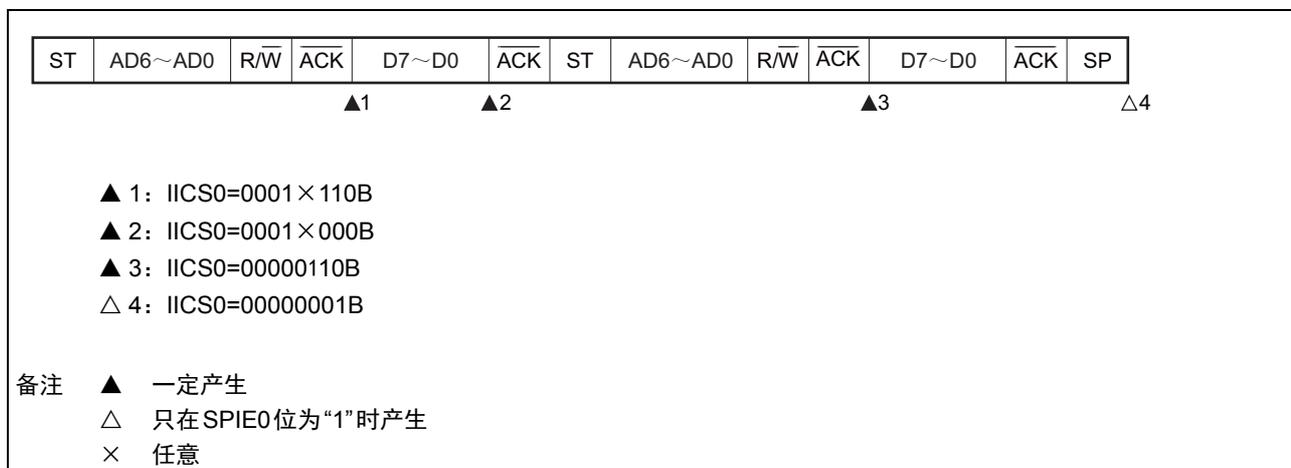


(ii) WTIM0=1的情况 (在重新开始后地址不同 (扩展码))

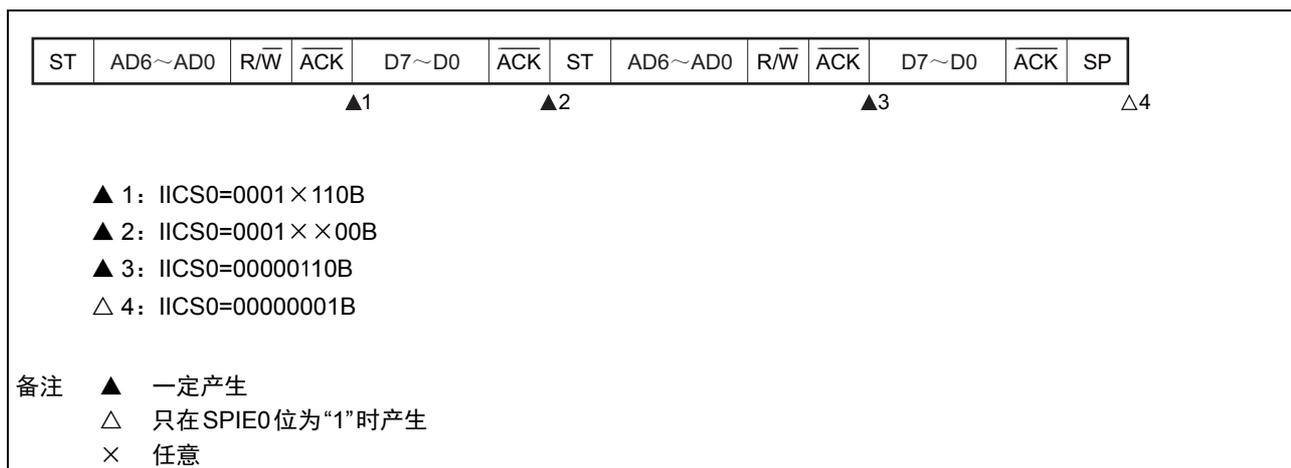


(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIM0=0的情况 (在重新开始后地址不同 (非扩展码))



(ii) WTIM0=1的情况 (在重新开始后地址不同 (非扩展码))

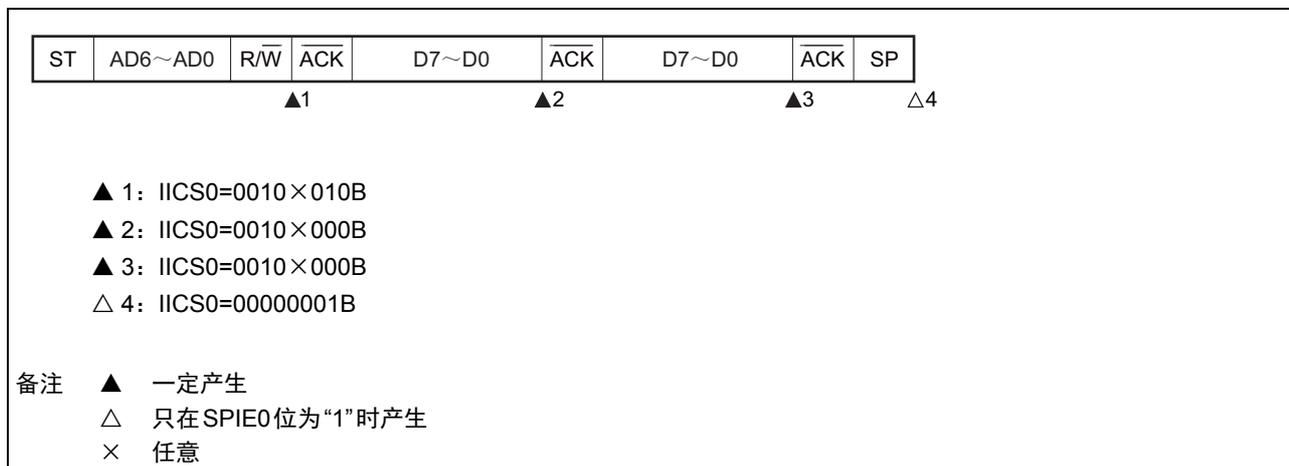


(3) 从属运行（接收扩展码的情况）

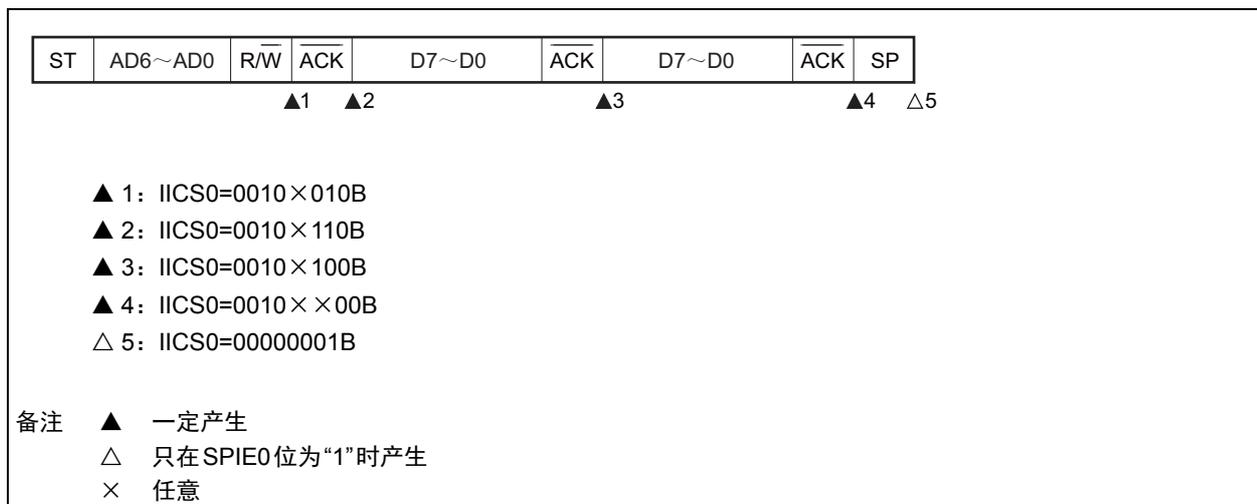
在接收扩展码时，始终参加通信。

(a) Start ~ Code ~ Data ~ Data ~ Stop

(i) WTIM0=0的情况

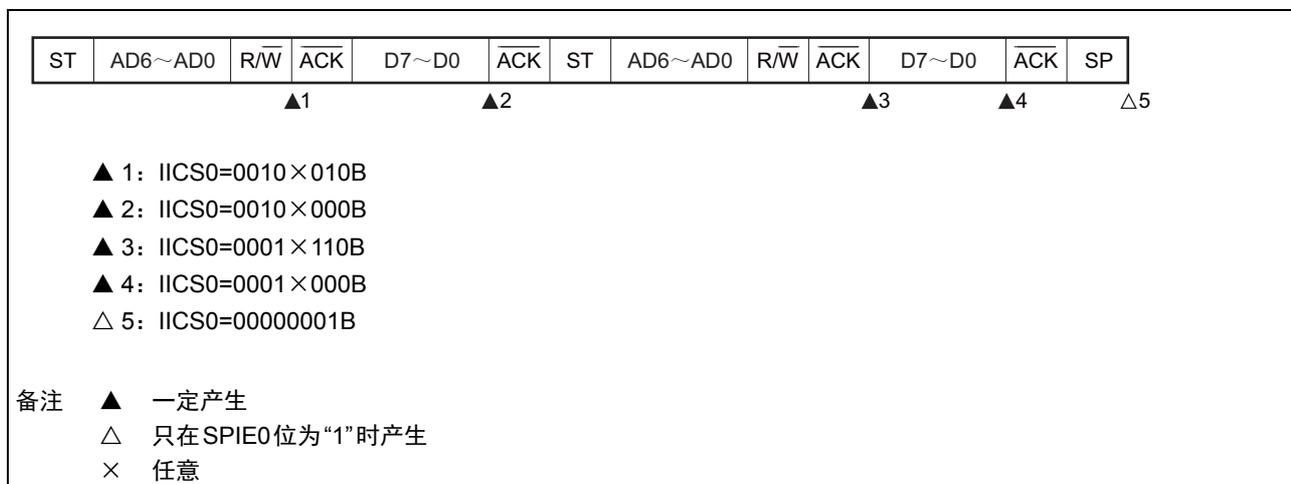


(ii) WTIM0=1的情况

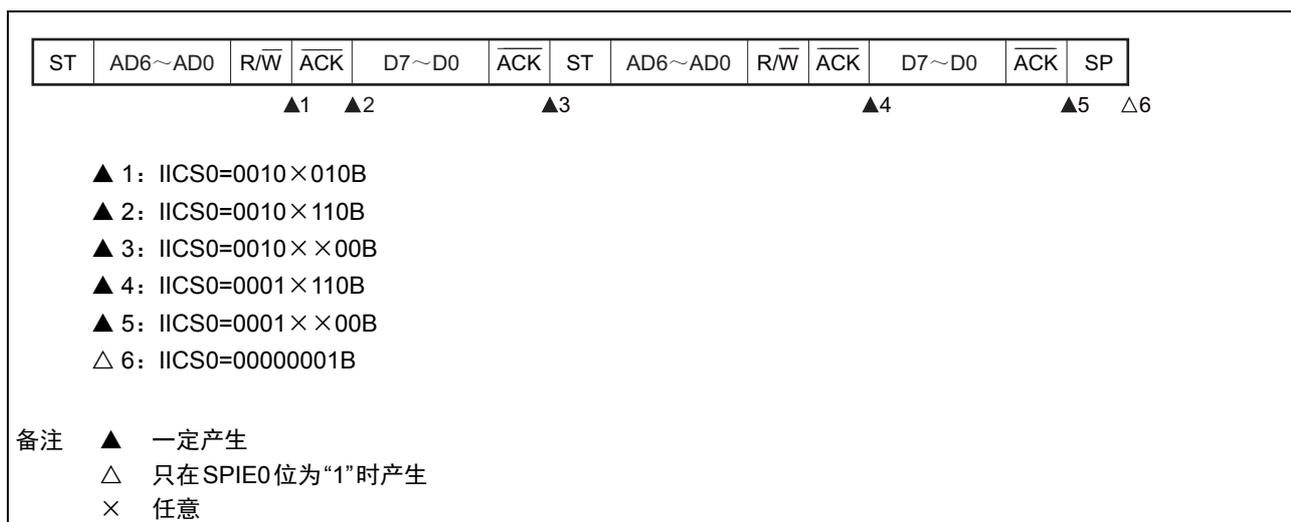


(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIM0=0的情况 (在重新开始后SVA0相同)

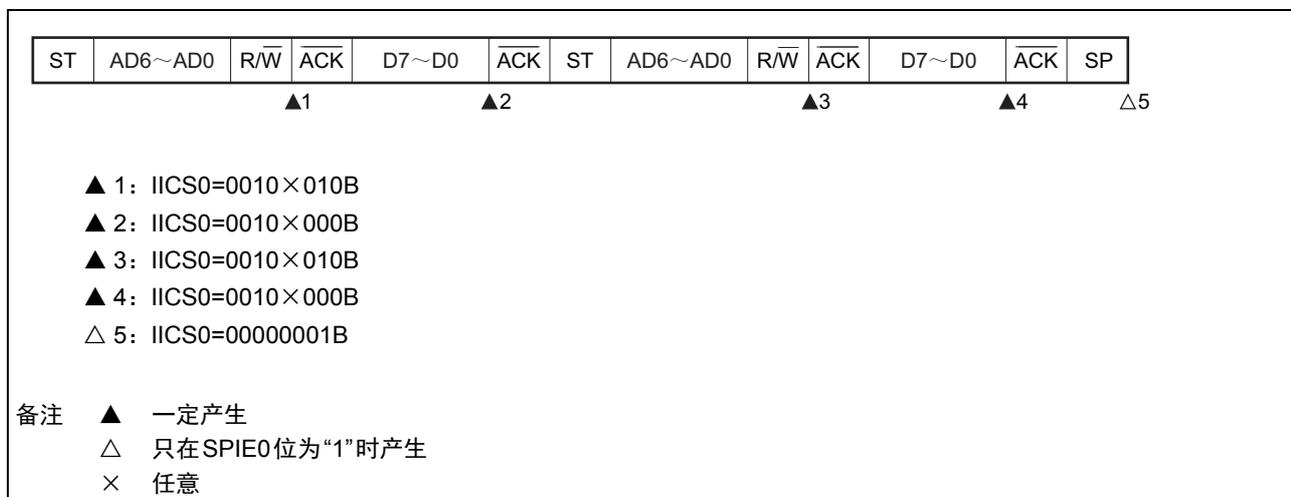


(ii) WTIM0=1的情况 (在重新开始后SVA0相同)

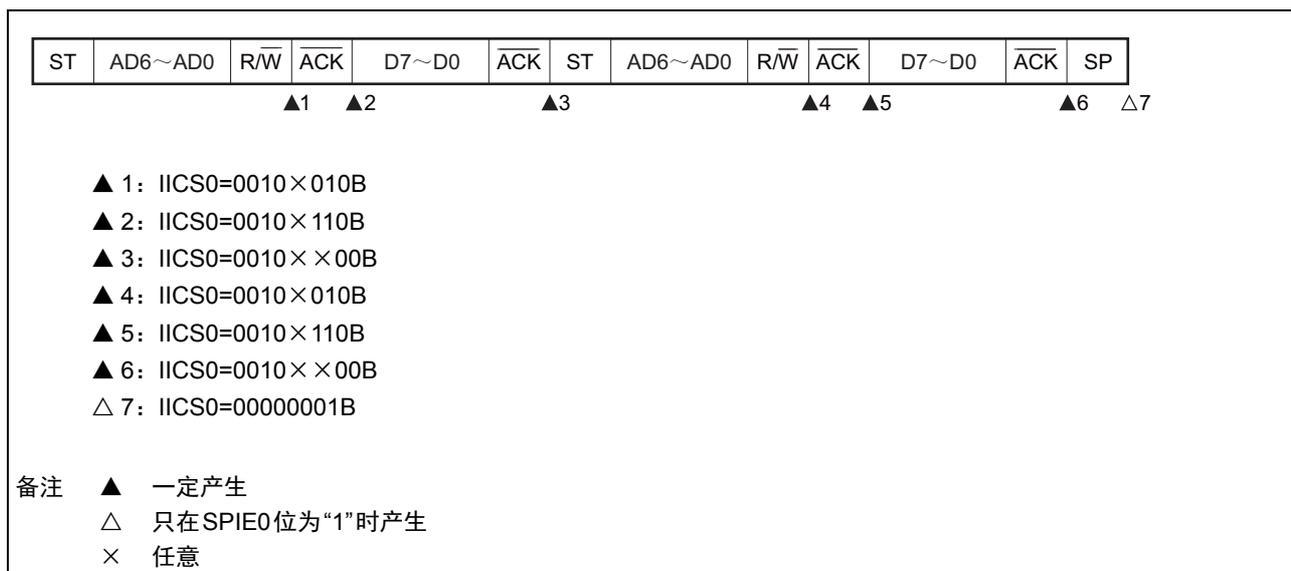


(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) WTIM0=0的情况 (在重新开始后接收扩展码)

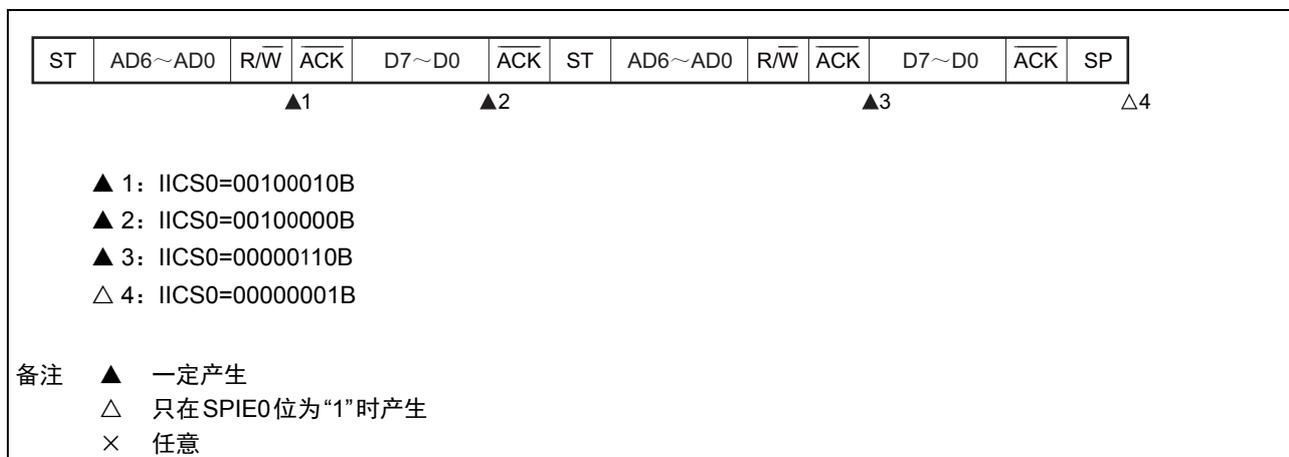


(ii) WTIM0=1的情况 (在重新开始后接收扩展码)

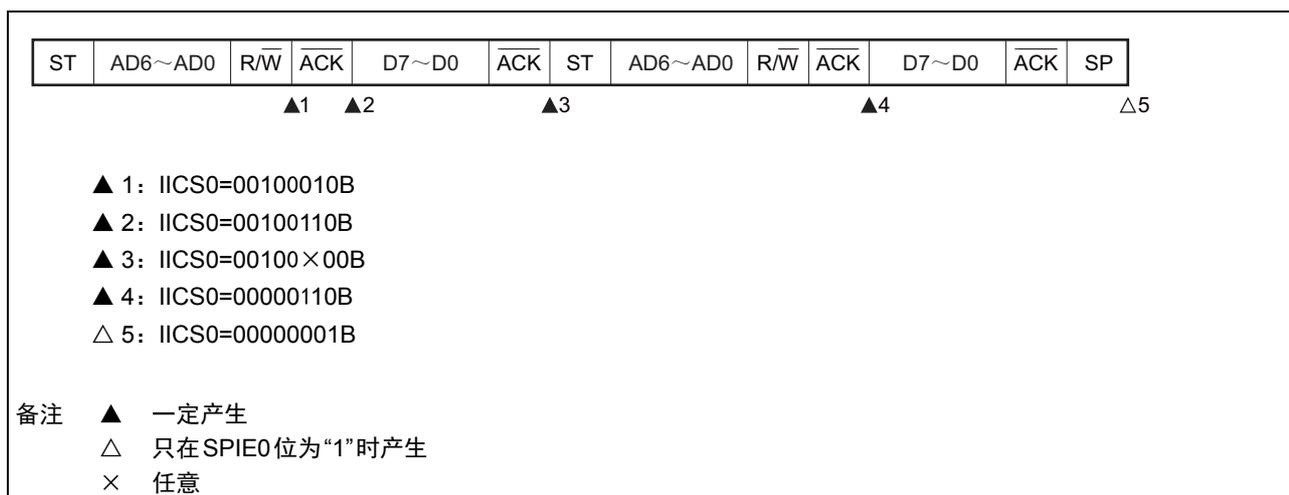


(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIM0=0的情况 (在重新开始后地址不同 (非扩展码))

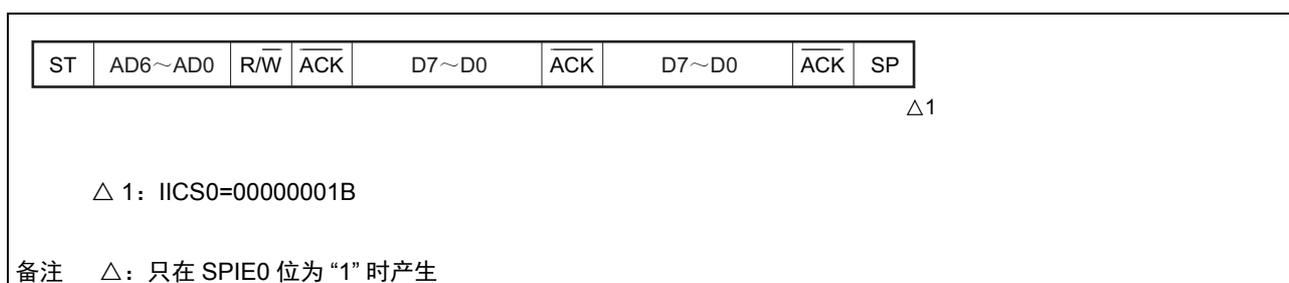


(ii) WTIM0=1的情况 (在重新开始后地址不同 (非扩展码))



(4) 不参加通信的运行

(a) Start ~ Code ~ Data ~ Data ~ Stop

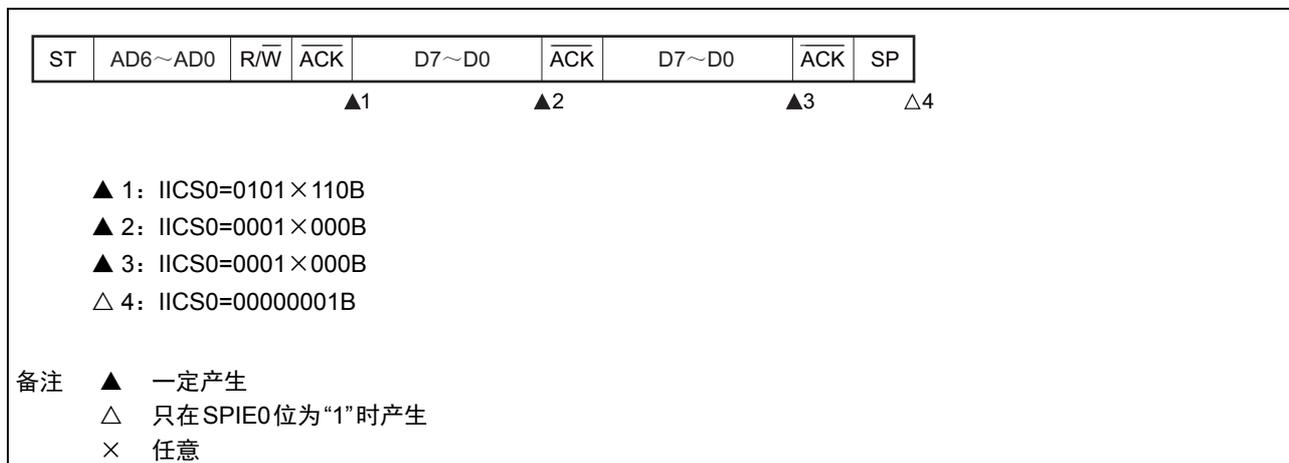


(5) 仲裁失败的运行（在仲裁失败后作为从属设备运行）

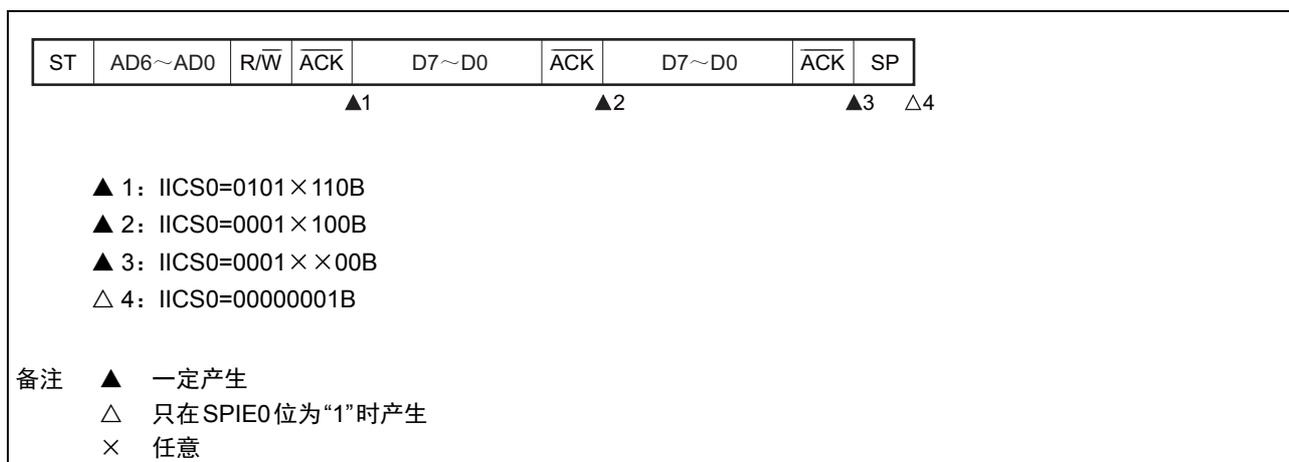
在多主控系统中用作主控设备时，必须在每次产生 INTIIC0 中断请求信号时读 MSTS0 位，确认仲裁结果。

(a) 在发送从属地址数据的过程中仲裁失败的情况

(i) WTIM0=0的情况

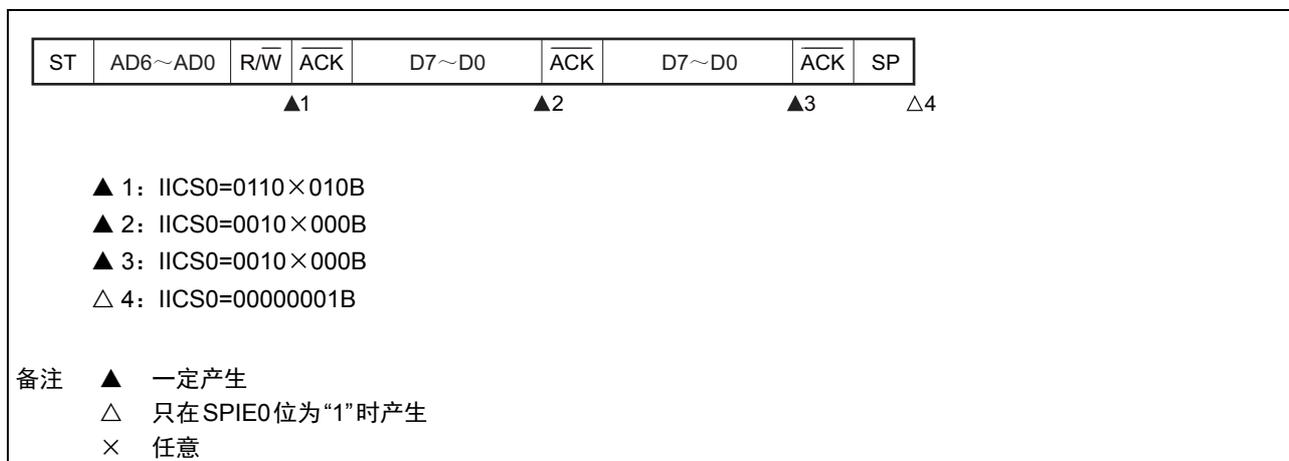


(ii) WTIM0=1的情况

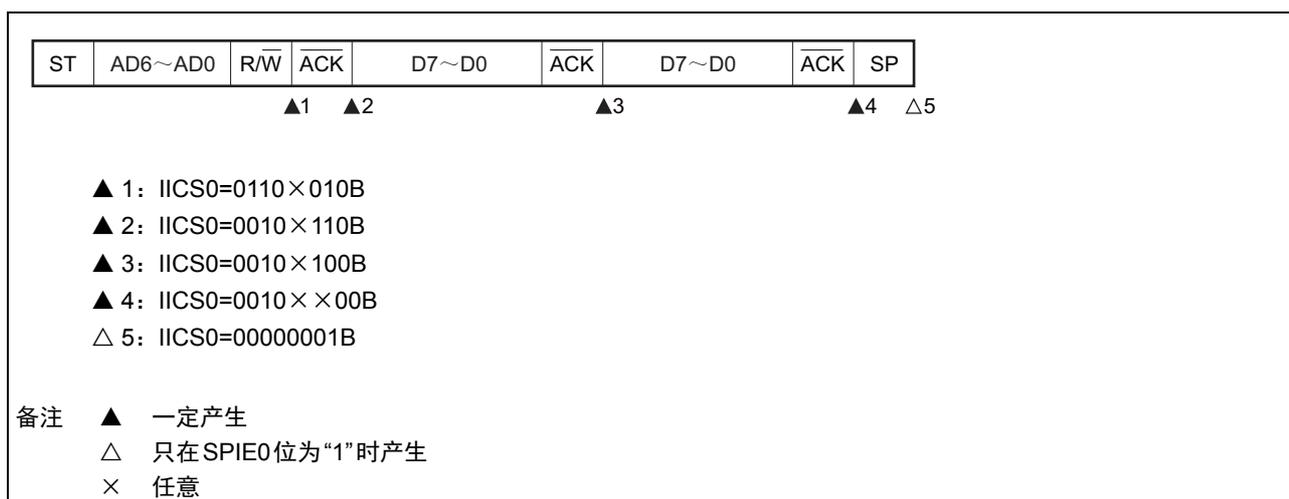


(b) 在发送扩展码的过程中仲裁失败的情况

(i) WTIM0=0的情况



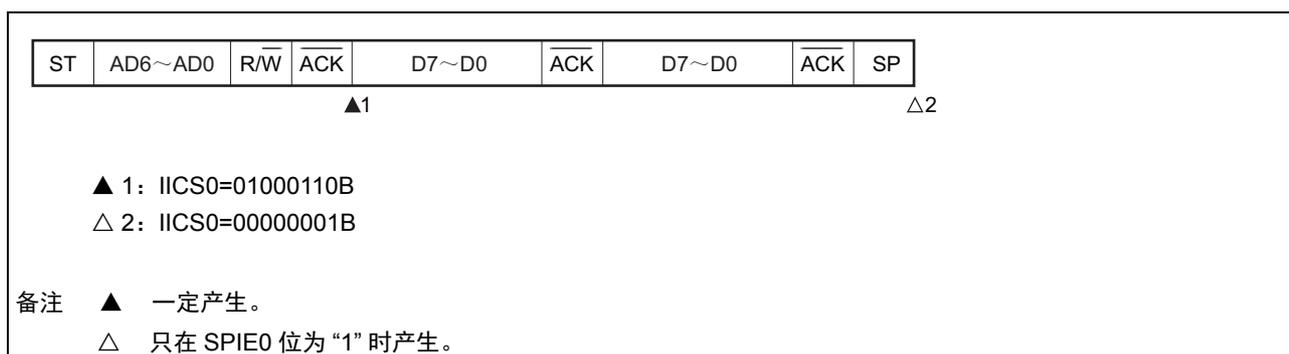
(ii) WTIM0=1的情况



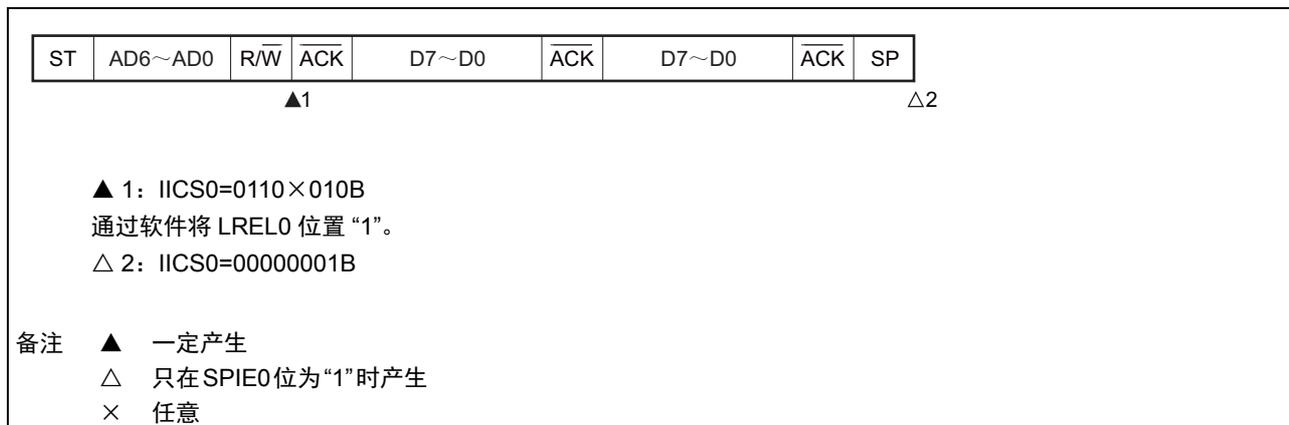
(6) 仲裁失败的运行（在仲裁失败后不参加通信）

在多主控系统中用作主控设备时，必须在每次产生 INTIIC0 中断请求信号时读 MSTS0 位，确认仲裁结果。

(a) 在发送从属地址数据的过程中仲裁失败的情况（WTIM0=1）

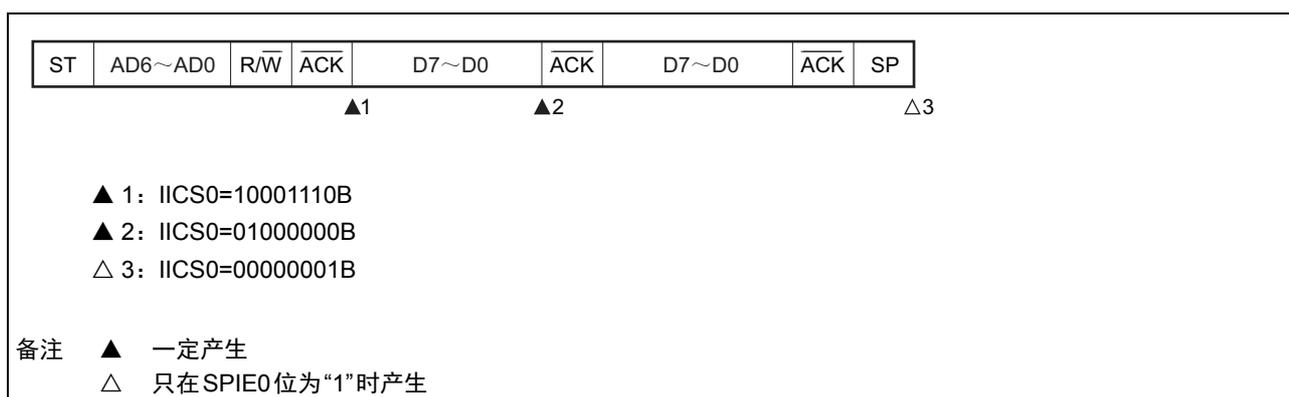


(b) 在发送扩展码的过程中仲裁失败的情况

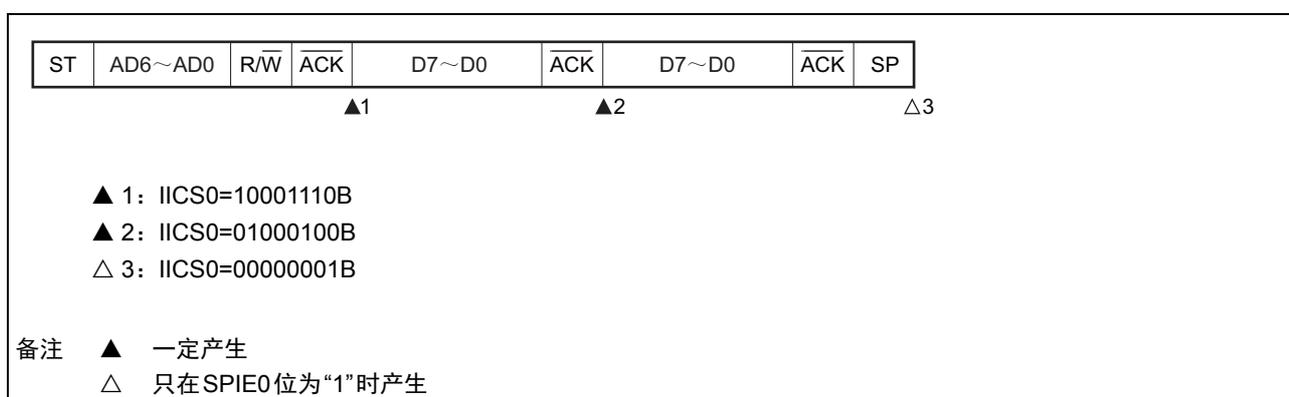


(c) 在发送数据的过程中仲裁失败的情况

(i) WTIM0=0 的情况

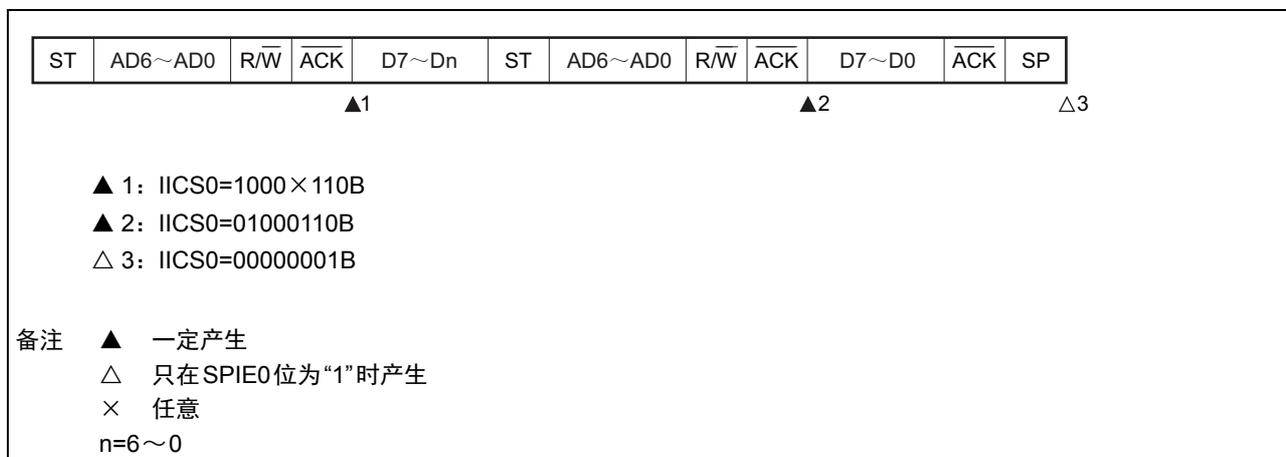


(ii) WTIM0=1

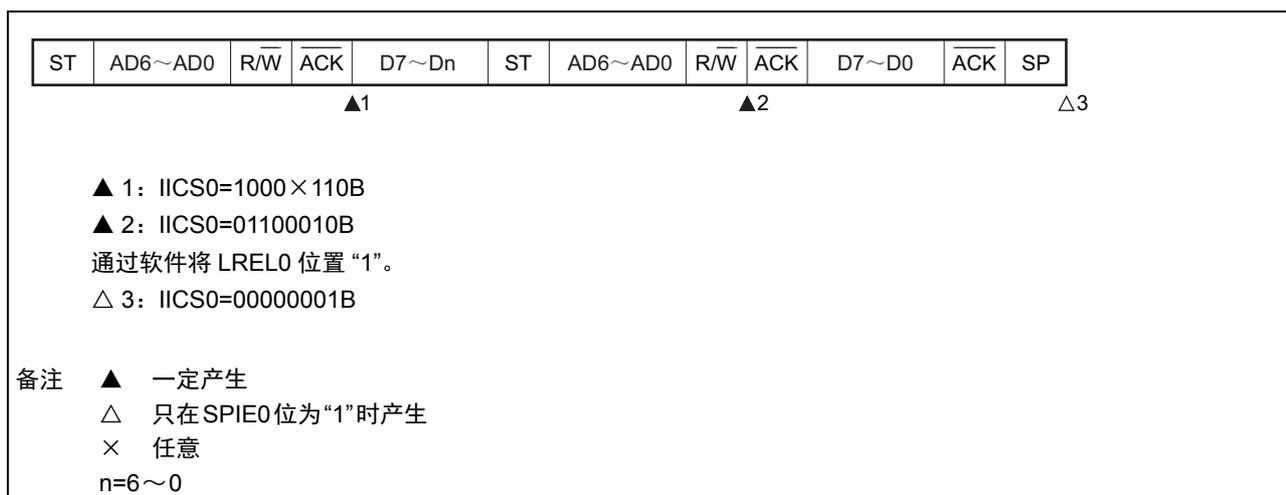


(d) 在传送数据时因重新开始条件而仲裁失败的情况

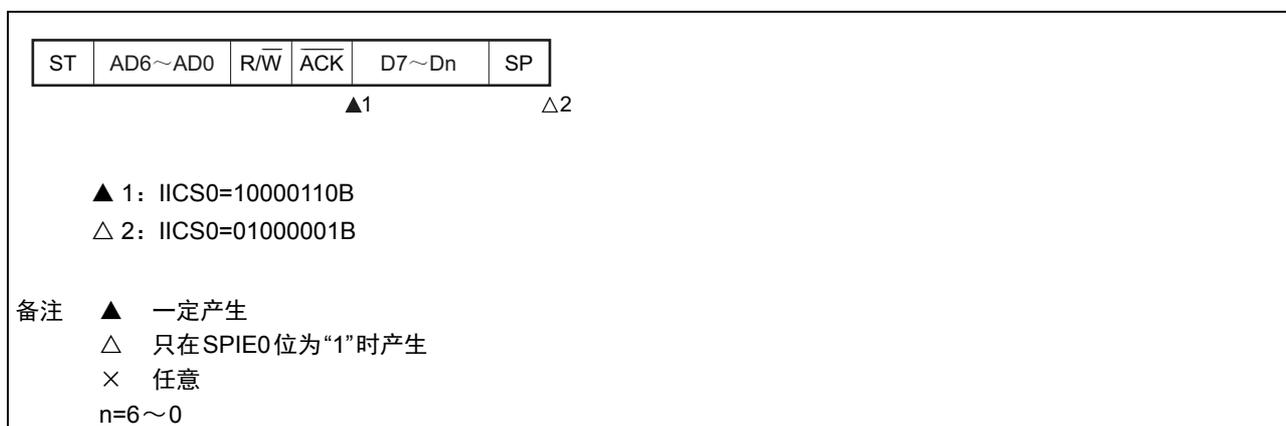
(i) 非扩展码（例如，SVA0不同）



(ii) 扩展码

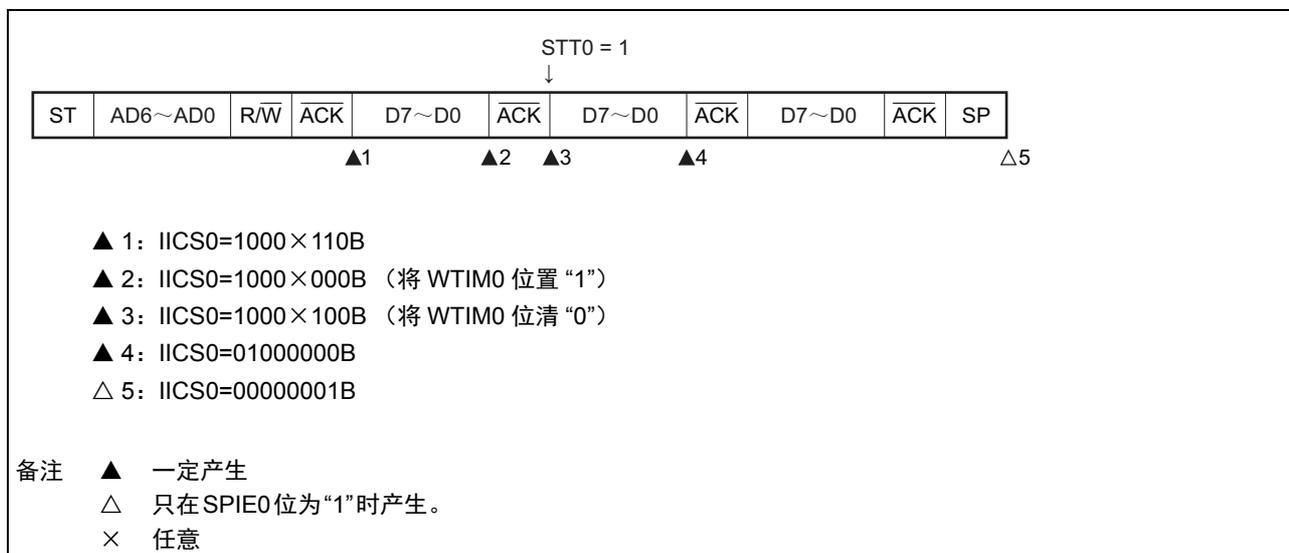


(e) 在传送数据时因停止条件而仲裁失败的情况

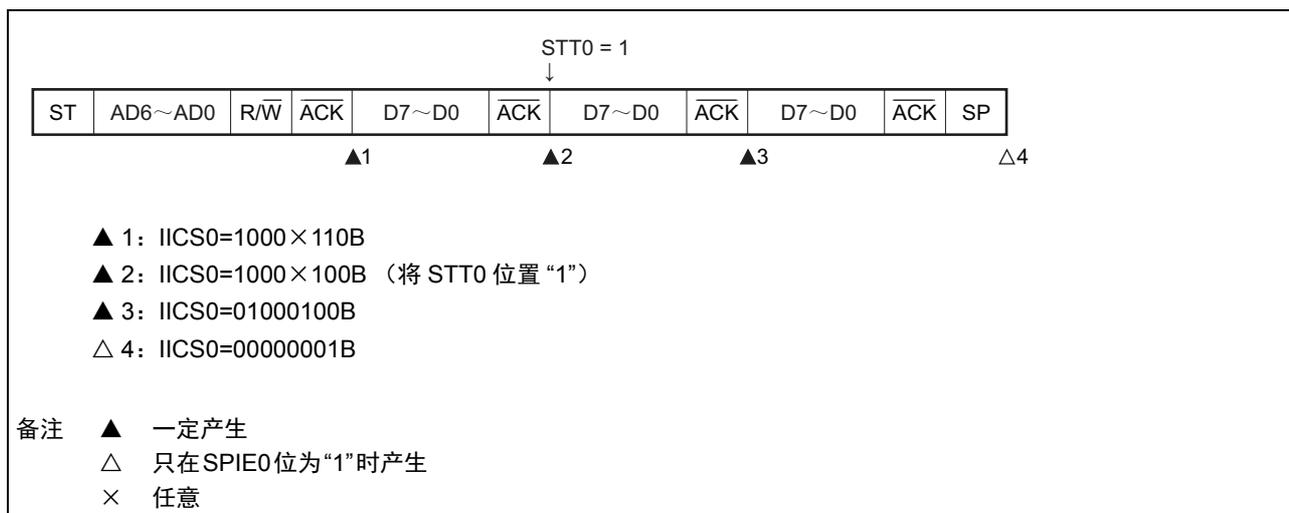


(f) 在想生成重新开始条件时因数据为低电平而仲裁失败的情况

(i) WTIM0=0的情况

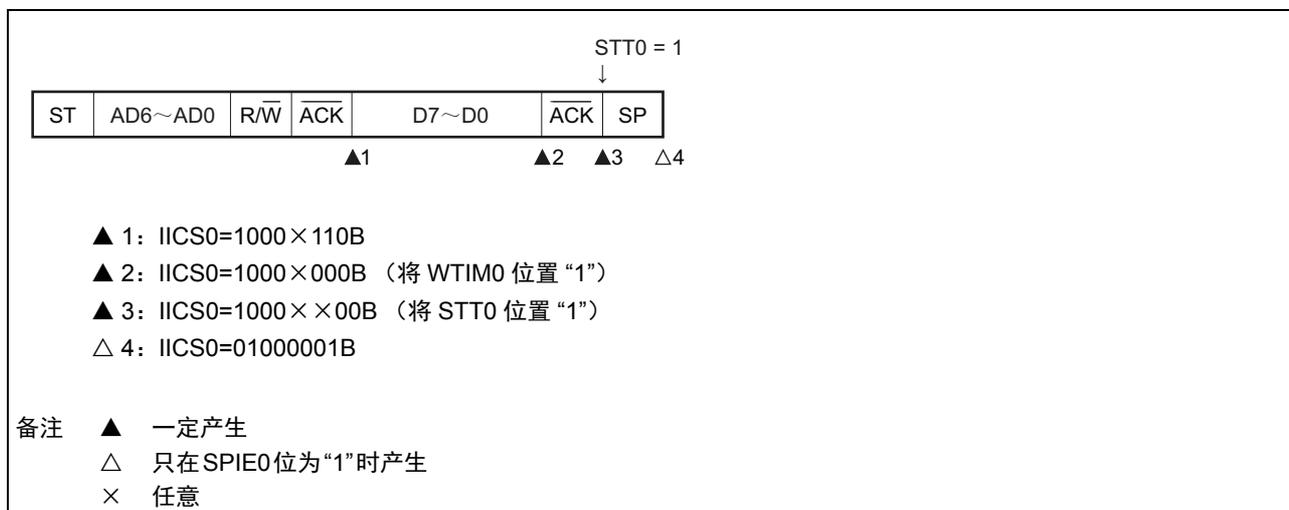


(ii) WTIM0=1的情况

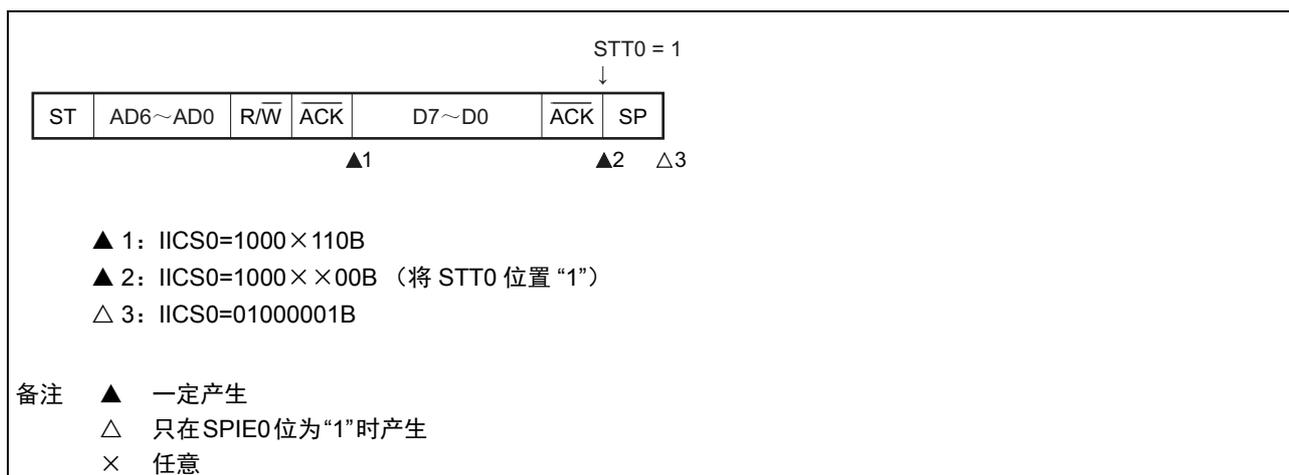


(g) 在想生成重新开始条件时因停止条件而仲裁失败的情况

(i) WTIM0=0的情况

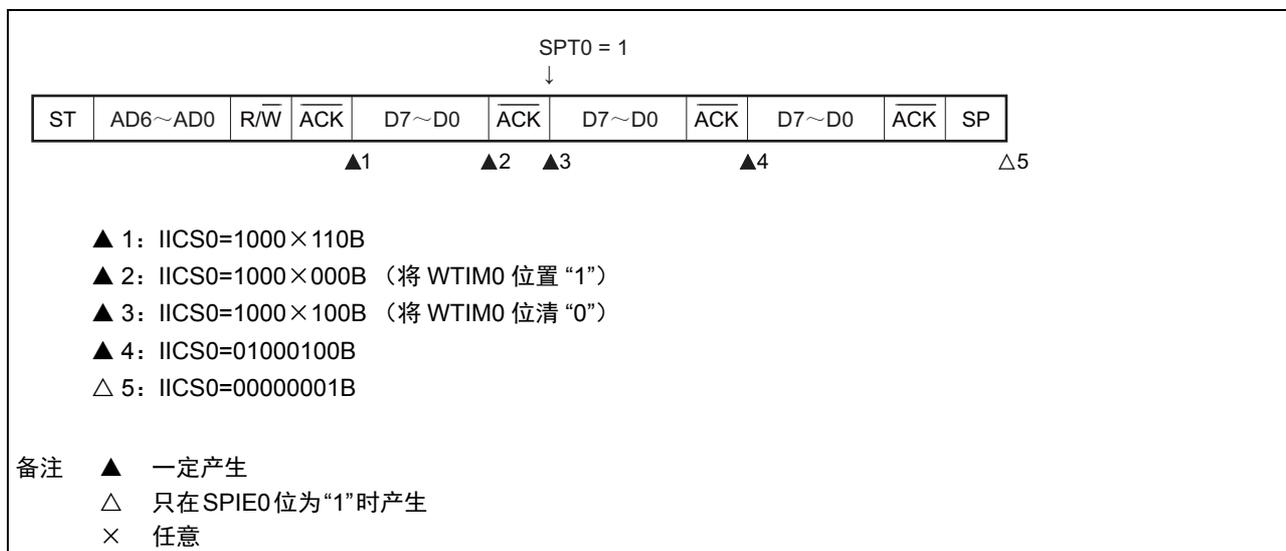


(ii) WTIM0=1的情况

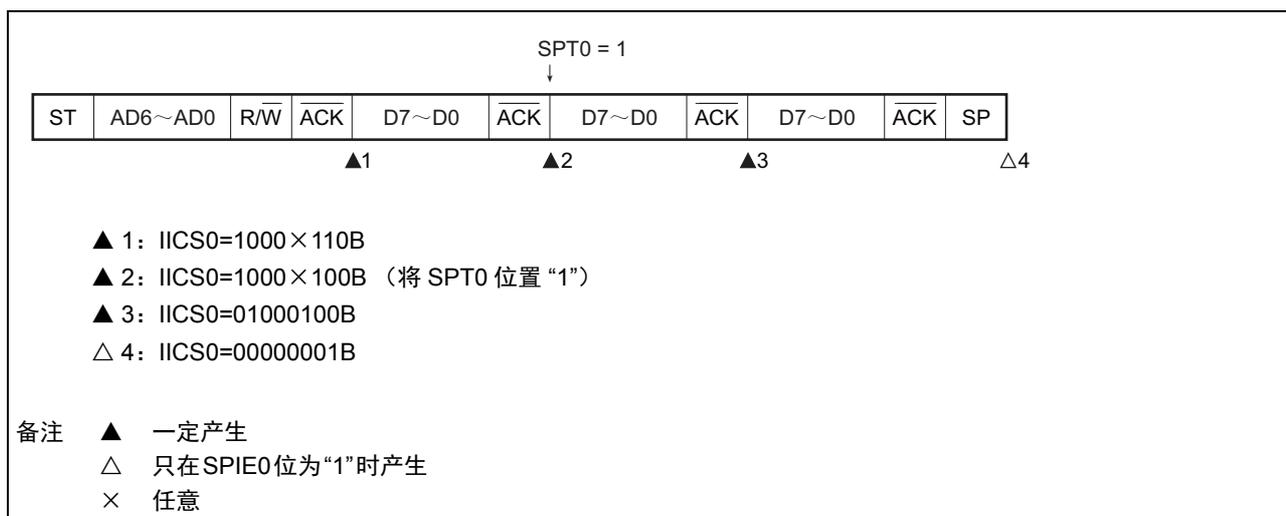


(h) 在想生成停止条件时因数据为低电平而仲裁失败的情况

(i) WTIM0=0的情况



(ii) WTIM0=1的情况



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，从多个从属设备选择一个通信对象的从属设备。

主控设备在从属设备地址之后发送表示数据传送方向的 TRC0 位（IIC 状态寄存器 0（IICS0）的 bit3），开始与从属设备进行串行通信。

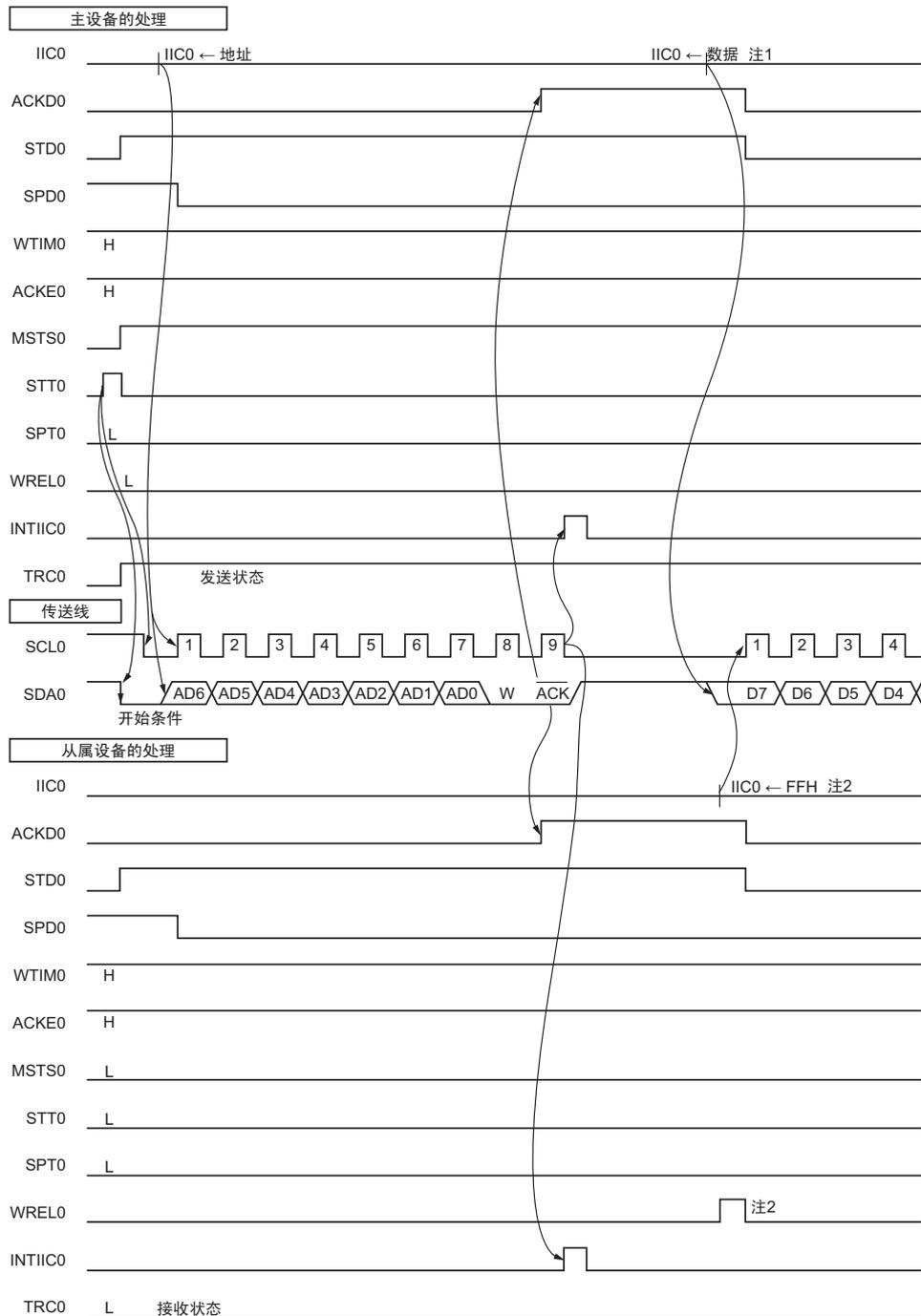
数据通信的时序图如图 15-27 和图 15-28 所示。

与串行时钟（SCL0）的下降沿同步进行 IIC 移位寄存器 0（IIC0）的移位，并且将发送数据传送到 SO0 锁存器，以 MSB 优先从 SDA0 引脚输出数据。

在 SCL0 的上升沿将 SDA0 引脚输入的数据取到 IIC0。

图 15-27 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) (1/3)

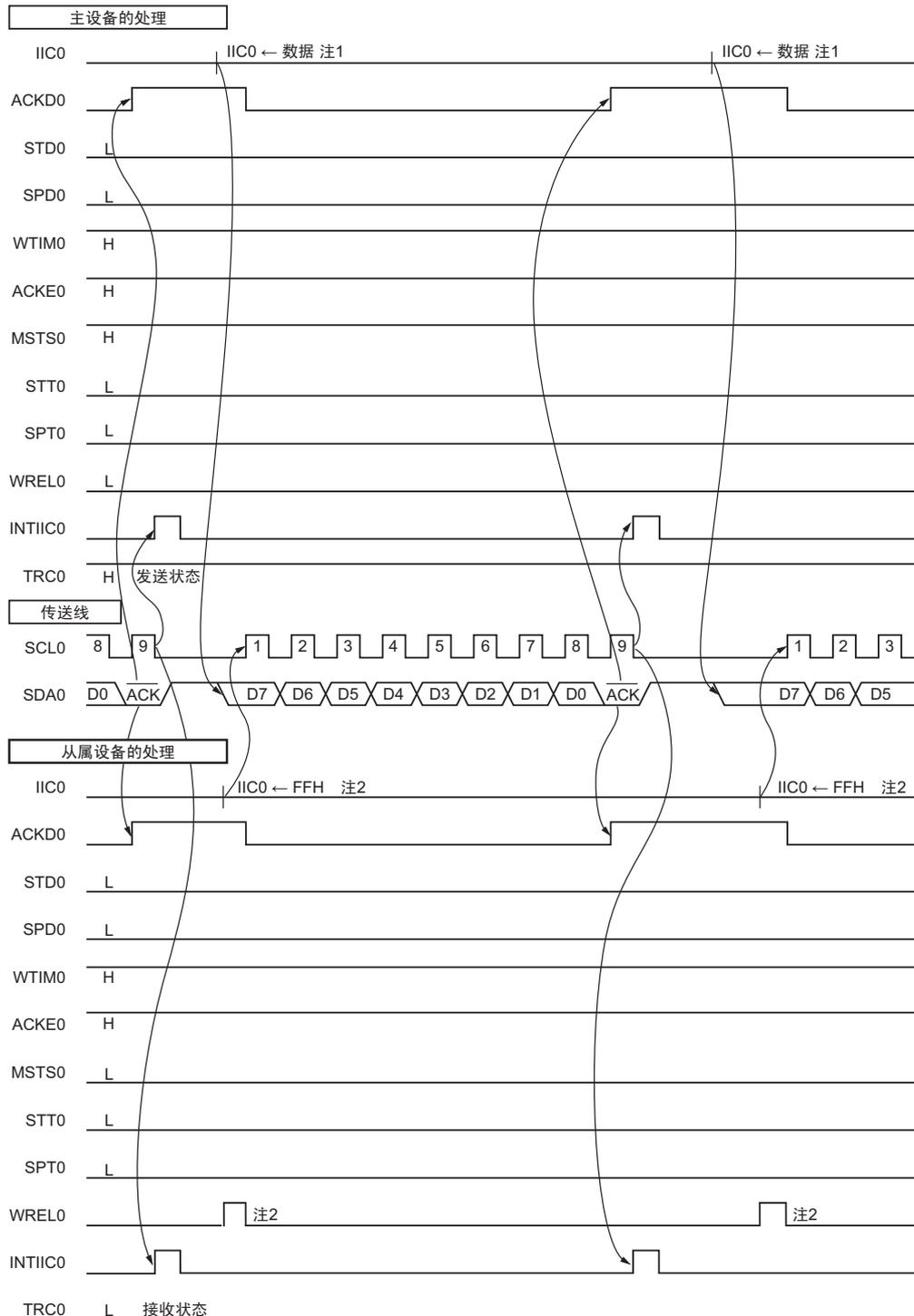
(1) 开始条件~地址



- 注 1. 要在主控方的发送期间解除等待时，必须给 IIC0 写数据而不是将 WRELO 位置位。
 2. 要在从属方的接收期间解除等待时，必须将 IIC0 置“FFH”或者将 WRELO 位置位。

图 15-27 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) (2/3)

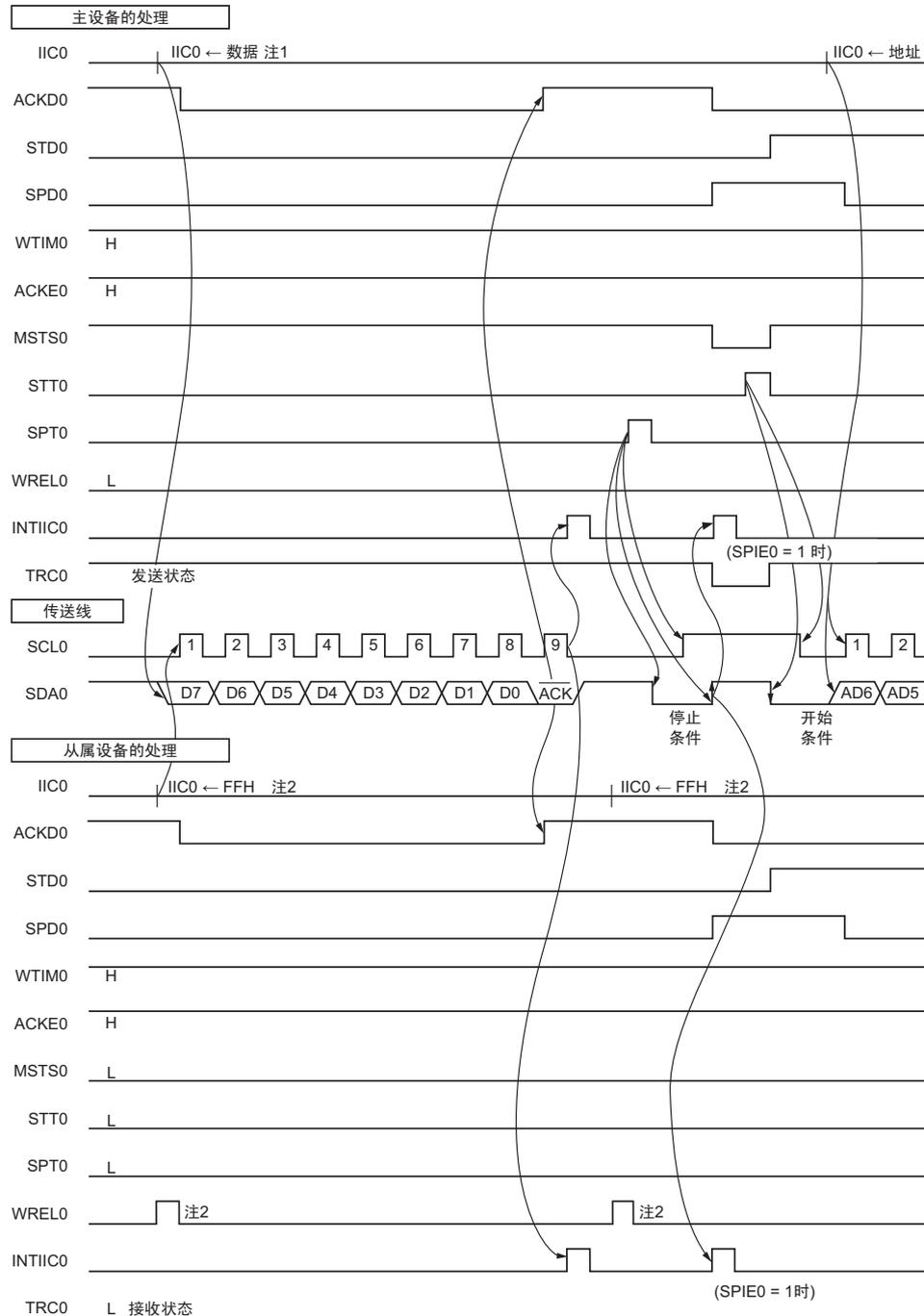
(2) 数据



- 注 1. 要在主控方的发送期间解除等待时，必须给 IIC0 写数据而不是将 WRELO 位置位。
 2. 要在从属方的接收期间解除等待时，必须将 IIC0 置“FFH”或者将 WRELO 位置位。

图 15-27 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) (3/3)

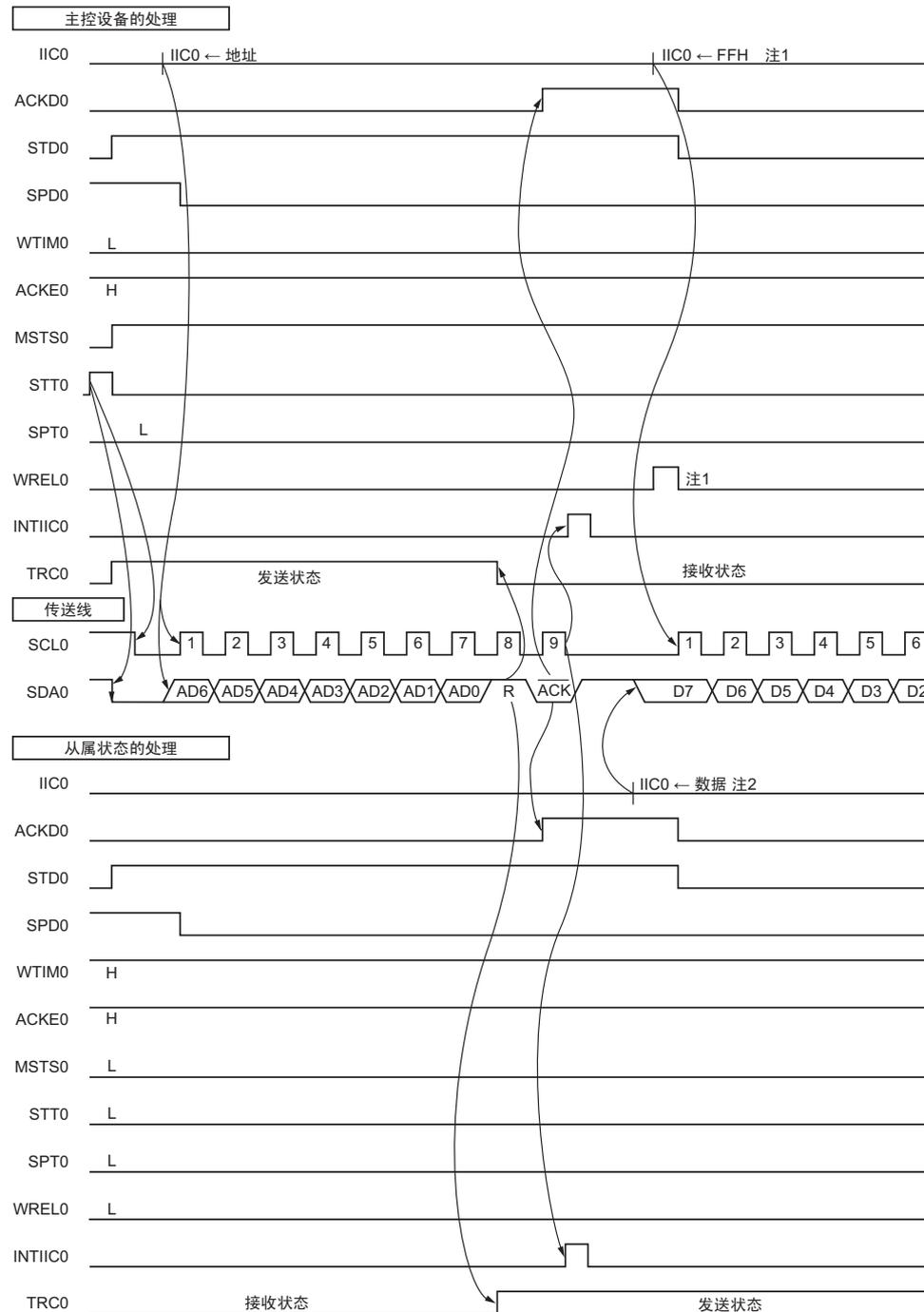
(3) 停止条件



- 注 1. 要在主控方的发送期间解除等待时, 必须给 IIC0 写数据而不是将 WRELO 位置位。
 注 2. 要在从属方的接收期间解除等待时, 必须将 IIC0 置“FFH”或者将 WRELO 位置位。

图 15-28 从属设备 → 主控设备的通信例子
 (主控设备：选择 8 个时钟等待，从属设备：选择 9 个时钟等待) (1/3)

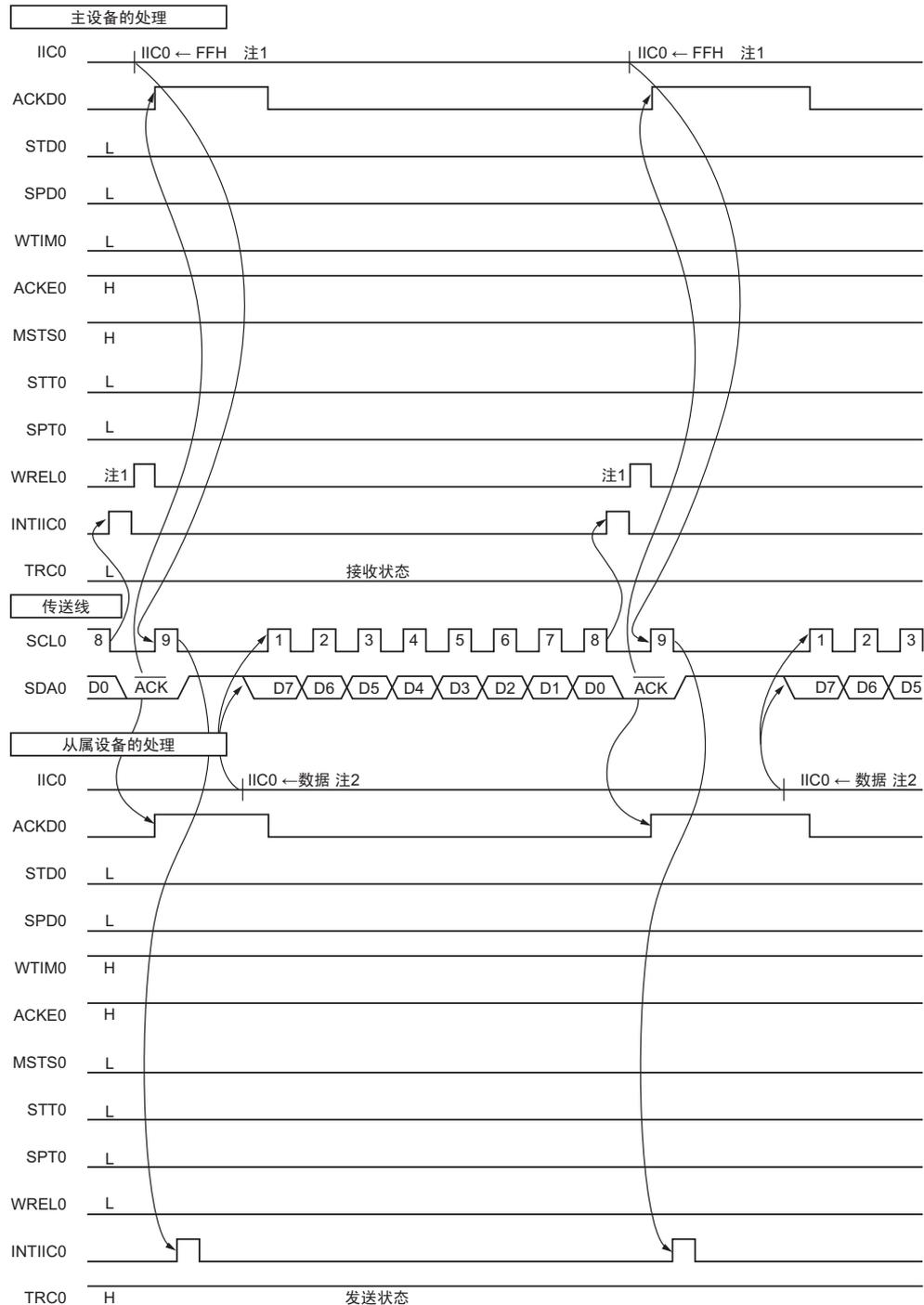
(1) 开始条件～地址



- 注 1. 要在主控方的接收期间解除等待时，必须将 IIC0 置 "FFH" 或者将 WRELO 位置位。
 注 2. 要在从属方的发送期间解除等待时，必须给 IIC0 写数据而不是将 WRELO 位置位。

图 15-28 从属设备 → 主控设备的通信例子
 (主控设备：选择 8 个时钟等待，从属设备：选择 9 个时钟等待) (2/3)

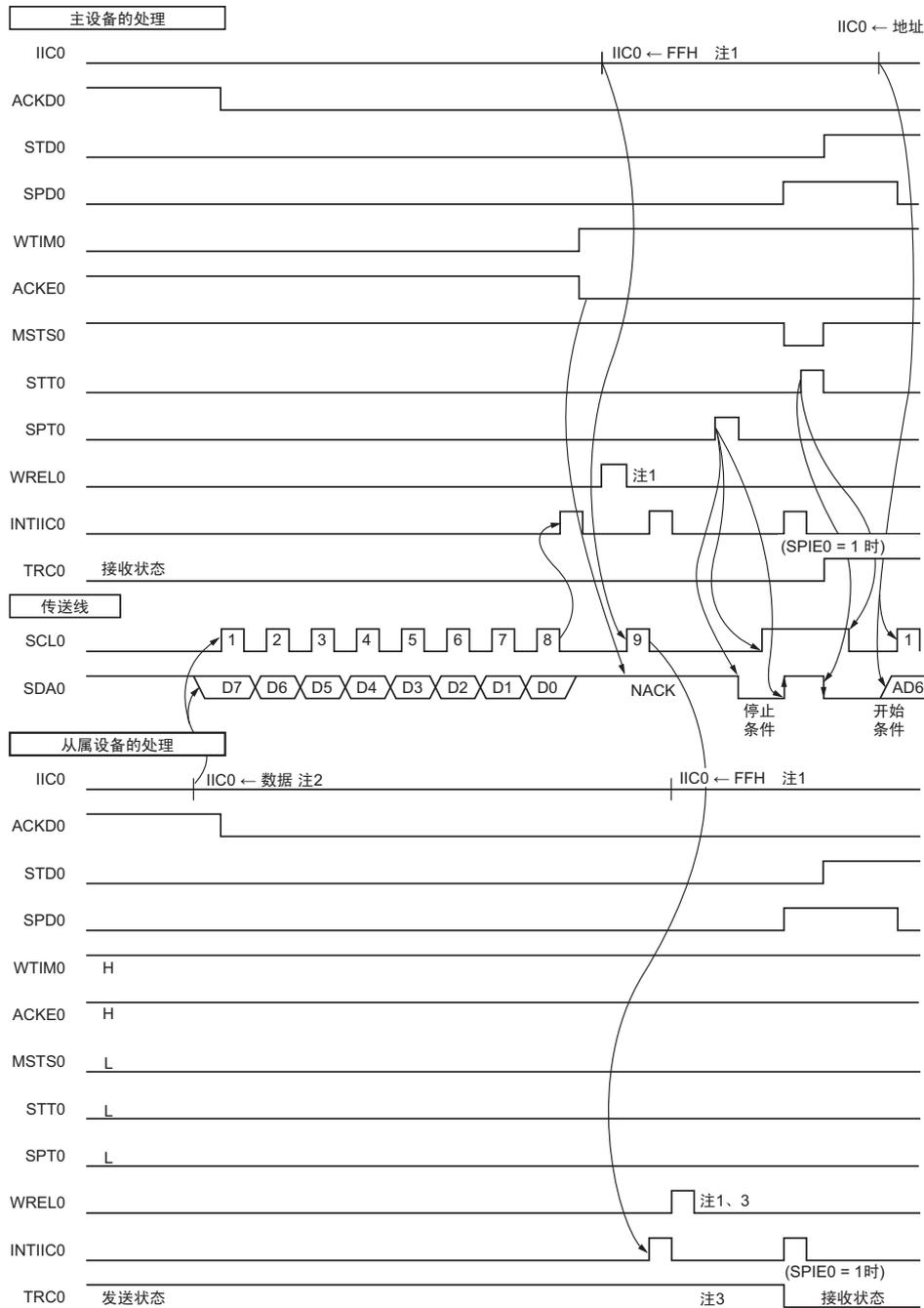
(2) 数据



- 注 1. 要在主控方的接收期间解除等待时，必须将 IIC0 置“FFH”或者将 WREL0 位置位。
 2. 要在从属方的发送期间解除等待时，必须给 IIC0 写数据而不是将 WREL0 位置位。

图 15-28 从属设备 → 主控设备的通信例子
 (主控设备：选择 8 个 → 9 个时钟等待，从属设备：选择 9 个时钟等待) (3/3)

(3) 停止条件



- 注 1. 要解除等待时，必须将 IIC0 置“FFH”或者将 WRELO 位置位。
 注 2. 要在从属方的发送期间解除等待时，必须给 IIC0 写数据而不是将 WRELO 位置位。
 注 3. 在从属方的发送期间，如果通过 WRELO 位的置位来解除等待，就清除 TRC0 位。

第 16 章 中断功能

16.1 中断功能的种类

中断功能有以下两种。

(1) 可屏蔽中断

这是接受屏蔽控制的中断。可通过设置优先级指定标志寄存器（PR0L、PR0H、PR1L、PR1H），将中断优先级分为高优先级中断组和低优先级中断组。高优先级中断可对低优先级中断进行多重中断。另外，如果同时产生多个相同优先级的中断请求，就根据向量中断处理的优先级进行处理。优先级请参照表 16-1。

在产生待机解除信号时，解除 STOP 模式和 HALT 模式。

可屏蔽中断有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 指令发生的向量中断。即使在禁止中断时也能被接受。软件中断不受中断优先级控制。

16.2 中断源和结构

中断源分为可屏蔽中断的中断源和软件中断的中断源。另外，还有最多 4 种的复位源（参照表 16-1）。

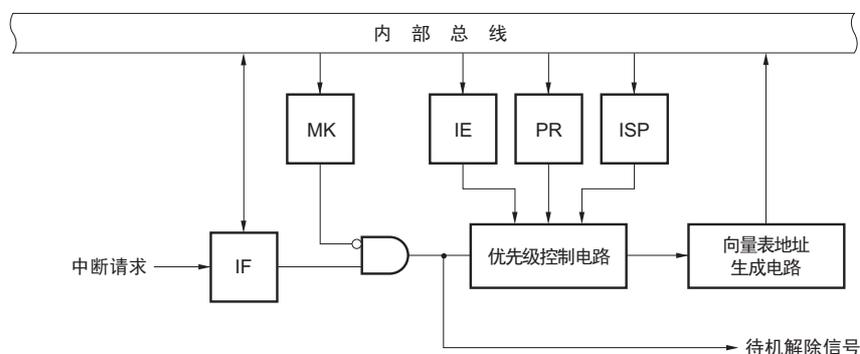
表 16-1 中断源一览

中断种类	内部 / 外部	基本结构类型注 1	默认优先级注 2	中断源		向量表地址
				名称	触发	
可屏蔽	内部	(A)	0	INTLVI	低电压检测注 3	0004H
			外部	(B)	1	INTP0
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	内部	(A)	7	INTSRE6	发生 UART6 的接收错误	0012H
			8	INTSR6	UART6 的接收结束	0014H
			9	INTST6	UART6 的发送结束	0016H
			10	INTCSI10/ INTST0	CSI10 的通信结束 / UART0 的发送结束	0018H
			11	INTTMH1	TMH1 和 CMP01 匹配（指定比较寄存器时）	001AH
			12	INTTMH0	TMH0 和 CMP00 匹配（指定比较寄存器时）	001CH
			13	INTTM50	TM50 和 CR50 匹配（指定比较寄存器时）	001EH
			14	INTTM000	TM00 和 CR000 匹配（指定比较寄存器时）、 TI010 引脚的有效边沿检测（指定捕捉寄存器时）	0020H
			15	INTTM010	TM00 和 CR010 匹配（指定比较寄存器时）、 TI000 引脚的有效边沿检测（指定捕捉寄存器时）	0022H
			16	INTAD	A/D 转换结束	0024H
			17	INTSR0	UART0 接收结束或者发生接收错误	0026H
			18	INTWTI	时钟定时器的基准时间间隔信号	0028H
			19	INTTM51注 4	TM51 和 CR51 匹配（指定比较寄存器时）	002AH
			21	INTWT	时钟定时器的上溢	002EH
	22	INTIIC0	IIC0 通信结束	0034H		
软件	—	(C)	—	BRK	执行 BRK 指令	003EH
复位	—	—	—	RESET	复位输入	0000H
				POC	上电清除	
				LVI	低电压检测注 3	
				WDT	WDT 的上溢	

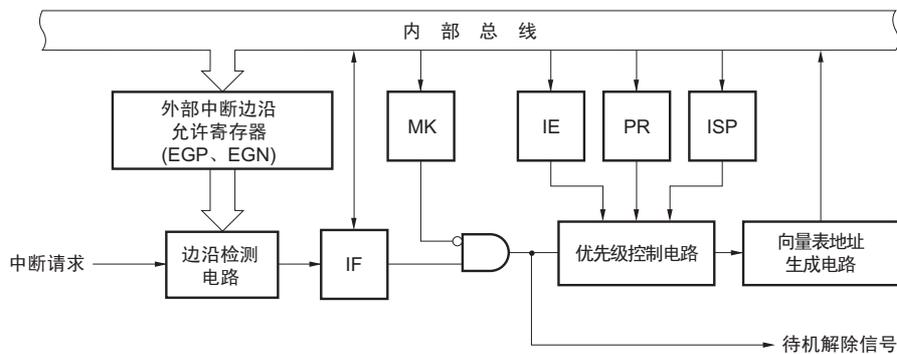
- 注 1. 基本结构类型 (A) ~ (C) 分别对应图 16-1 中的 (A) ~ (C)。
2. 在同时产生多个可屏蔽中断时，默认优先级决定优先处理的向量中断。“0”为最高优先级，“22”为最低优先级。
3. 选择低电压检测寄存器（LVIM）的 bit1（LVIMD）为“0”时的情况。
4. 在载波发生器模式中使用 8 位定时器 / 事件计数器 51 时，在 INTTM5H1 信号的中断时序发生中断（参照“图 8-13 传送时序”）。

图 16-1 中断功能的基本结构

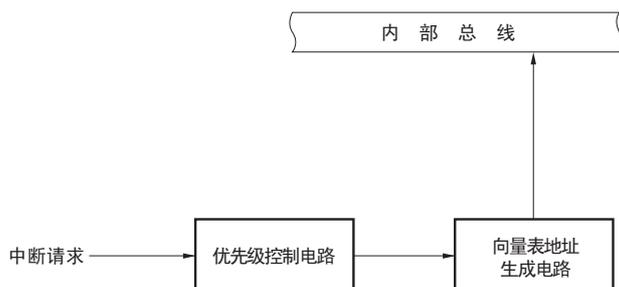
(A) 内部可屏蔽中断



(B) 外部可屏蔽中断(INTPn)



(C) 软件中断



备注 n=0 ~ 5

IF: 中断请求标志

IE: 中断允许标志

ISP: 优先级控制标志

MK: 中断屏蔽标志

PR: 优先级指定标志

16.3 中断功能的控制寄存器

通过以下 6 种寄存器控制中断功能。

- 中断请求标志寄存器 (IF0L、IF0H、IF1L、IF1H)
- 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L、MK1H)
- 优先级指定标志寄存器 (PR0L、PR0H、PR1L、PR1H)
- 外部中断上升沿允许寄存器 (EGP)
- 外部中断下降沿允许寄存器 (EGN)
- 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 16-2 所示。

表 16-2 对应中断请求源的各标志

中断源	中断请求标志		中断屏蔽标志		优先级指定标志	
		寄存器		寄存器		寄存器
INTLVI	LVIF		LVIMK		LVIPR	
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTP5	PIF5		PMK5		PPR5	
INTSRE6	SREIF6		SREMK6		SREPR6	
INTSR6	SRIF6		SRMK6		SRPR6	
INTST6	STIF6		STMK6		STPR6	
INTCSI10	CSIF10 注 1	DUALIF0 注 1	CSIMK10 注 2	DUALMK0 注 2	CSIPR10 注 3	DUALPR0 注 3
INTST0	STIF0 注 1		STMK0 注 2		STPR0 注 3	
INTTMH1	TMIFH1		TMMKH1		TMPRH1	
INTTMH0	TMIFH0		TMMKH0		TMPRH0	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM000	TMIF000		TMMK000		TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010	
INTAD	ADIF		ADMK		ADPR	
INTSR0	SRIF0		SRMK0		SRPR0	
INTWTI	WTIF		WTIMK		WTIPR	
INTTM51 ^{注 4}	TMIF51		TMMK51		TMPR51	
INTWT	WTIF		WTMK		WTPR	
INTIIC0	IICIF0		IICMK0		IICRP0	

注 1. 如果产生中断源 INTCSI10 或者 INTST0，IF0H 的 bit2 就置“1”。

2. MK0H 的 bit2 对应中断源 INTCSI10 和 INTST0。

3. PR0H 的 bit2 对应中断源 INTCSI10 和 INTST0。

4. 在载波发生器模式中使用 8 位定时器 / 事件计数器 51 时，在 INTTM5H1 信号的中断时序发生中断（参照“图 8-13 传送时序”）。

(1) 中断请求标志寄存器 (IF0L、IF0H、IF1L、IF1H)

在产生对应的中断请求或者执行指令时，中断请求标志置“1”；如果接受中断请求，在产生复位信号或者执行指令时，中断请求标志清“0”。

一旦接受中断请求，就在自动清除中断请求标志后，进入中断程序。

通过 1 位或 8 位存储器操作指令设置 IF0L、IF0H、IF1L 和 IF1H。另外，将 IF0L 与 IF0H、IF1L 与 IF1H 的组合用作 16 位寄存器 IF0、IF1 时，通过 16 位存储器操作指令设置。

在产生复位信号后，IF0L、IF0H、IF1L 和 IF1H 为“00H”。

注意 1. 如果要在解除待机后运行定时器、串行接口、A/D 转换器等等时，必须先清除中断请求标志，再进行运行。有可能因噪声等使中断请求标志置位。

2. 使用 1 位存储器操作指令 (CLR1) 进行中断请求标志寄存器的标志操作。使用 C 语言描述时，由于编译的汇编程序必须是 1 位存储器操作指令 (CLR1)，所以应使用如“IF0L.0 = 0;”或者“_asm(“clr1 IF0L,0”);”的位操作指令。

如果通过如“IF0L &= 0xfe;”的 8 位存储器操作指令作为 C 语言进行描述，在进行编译后将成为 3 条汇编指令。

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

此时，即使在“mov a, IF0L”和“mov IF0L, a”之间将同一个中断请求标志寄存器 (IF0L) 的其他位的请求标志置“1”，也通过“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。

图 16-2 中断请求标志寄存器 (IF0L、IF0H、IF1L、IF1H) 的格式

地址: FFE0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	SREIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIF

地址: FFE1H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	TMIF010	TMIF000	TMIF50	TMIFH0	TMIFH1	DUALIF0 CSIIF10 STIF0	STIF6	SRIF6

地址: FFE2H 复位后: 00H R/W

符号	7	6	<5>	4	<3>	<2>	<1>	<0>
IF1L	0	0	WTIF	0	TMIF51	WTIF	SRIF0	ADIF

地址: FFE3H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	<0>
IF1H	0	0	0	0	0	0	0	IICIF0

XXIFX	中断请求标志
0	不产生中断请求信号。
1	产生中断请求信号，进入中断请求状态。

注意 IF1L 的 bit4、bit6、bit7 和 IF1H 的 bit ~ bit7 必须置“0”。

(2) 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L、MK1H)

中断屏蔽标志寄存器设置允许 / 禁止对应的可屏蔽中断处理。

通过 1 位或 8 位存储器操作指令设置 MK0L、MK0H、MK1L 和 MK1H。另外，在将 MK0L 和 MK0H、MK1L 和 MK1H 的组合用作 16 位寄存器 MK0、MK1 时，通过 16 位存储器操作指令设置。

在产生复位信号后，MK0L、MK0H、MK1L 和 MK1H 变为“FFH”。

图 16-3 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L、MK1H) 的格式

地址: FFE4H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	SREMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

地址: FFE5H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	TMMK010	TMMK000	TMMK50	TMMKH0	TMMKH1	DUALMK0 CSIMK10 STMK0	STMK6	SRMK6

地址: FFE6H 复位后: FFH R/W

符号	7	6	<5>	4	<3>	<2>	<1>	<0>
MK1L	1	1	WTMK	1	TMMK51	WTMK	SRMK0	ADMK

地址: FFE7H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	<0>
MK1H	1	1	1	1	1	1	1	IICMK0

XXMKX	中断处理的控制
0	允许中断处理
1	禁止中断处理

注意 MK1L 的 bit4、bit6、bit7 和 MK1H 的 bit1 ~ bit7 必须置“1”。

(3) 优先级指定标志寄存器 (PR0L、PR0H、PR1L、PR1H)

优先级指定标志寄存器设置对应的可屏蔽中断的优先级。

通过 1 位或 8 位存储器操作指令设置 PR0L、PR0H、PR1L 和 PR1H。另外，在将 PR0L 和 PR0H、PR1L 和 PR1H 的组合用作 16 位寄存器 PR0、PR1 时，通过 16 位存储器操作指令设置。

产生复位信号时，PR0L、PR0H、PR1L 和 PR1H 变为“FFH”。

图 16-4 优先级指定标志寄存器 (PR0L、PR0H、PR1L、PR1H) 的格式

地址: FFE8H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0L	SREPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	LVIPR

地址: FFE9H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0H	TMPR010	TMPR000	TMPR50	TMPRH0	TMPRH1	DUALPR0 CSIPR10 STPR0	STPR6	SRPR6

地址: FFEAH 复位后: FFH R/W

符号	7	6	<5>	4	<3>	<2>	<1>	<0>
PR1L	1	1	WTPR	1	TMPR51	WTIPR	SRPR0	ADPR

地址: FFE BH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	<0>
PR1H	1	1	1	1	1	1	1	IICPR0

XXPRX	优先级的选择
0	高优先级
1	低优先级

注意 PR1L 的 bit4、bit6、bit7 和 PR1H 的 bit1 ~ bit7 必须置“1”。

(4) 外部中断上升沿允许寄存器 (EGP)、外部中断下降沿允许寄存器 (EGN)

外部中断上升沿允许寄存器和外部中断下降沿允许寄存器设置 INTP_n 的有效边沿。

通过 1 位或 8 位存储器操作指令设置 EGP 和 EGN。

产生复位信号时，EGP 和 EGN 变为“00H”。

备注 n=0 ~ 5

图 16-5 外部中断上升沿允许寄存器 (EGP) 和外部中断下降沿允许寄存器 (EGN) 的格式

地址: FF48H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGP	0	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

地址: FF49H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGN	0	0	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGP _n	EGN _n	选择 INTP _n 引脚的有效边沿
0	0	禁止边沿检测
0	1	下降沿
1	0	上升沿
1	1	双边沿

注意 EGP 和 EGN 的 bit6、bit7 必须置“0”。

备注 n=0 ~ 5

EGP_n 和 EGN_n 的对应端口如表 16-3 所示。

表 16-3 与 EGP_n 和 EGN_n 对应的端口

检测允许寄存器		边沿检测端口	中断请求信号
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P30	INTP1
EGP2	EGN2	P31	INTP2
EGP3	EGN3	P32	INTP3
EGP4	EGN4	P33	INTP4
EGP5	EGN5	P16	INTP5

注意 将外部中断功能切换到端口功能时，有可能进行边沿检测，因此，必须在将 EGP_n 和 EGN_n 置“0”后切换到端口模式。

备注 n=0 ~ 5

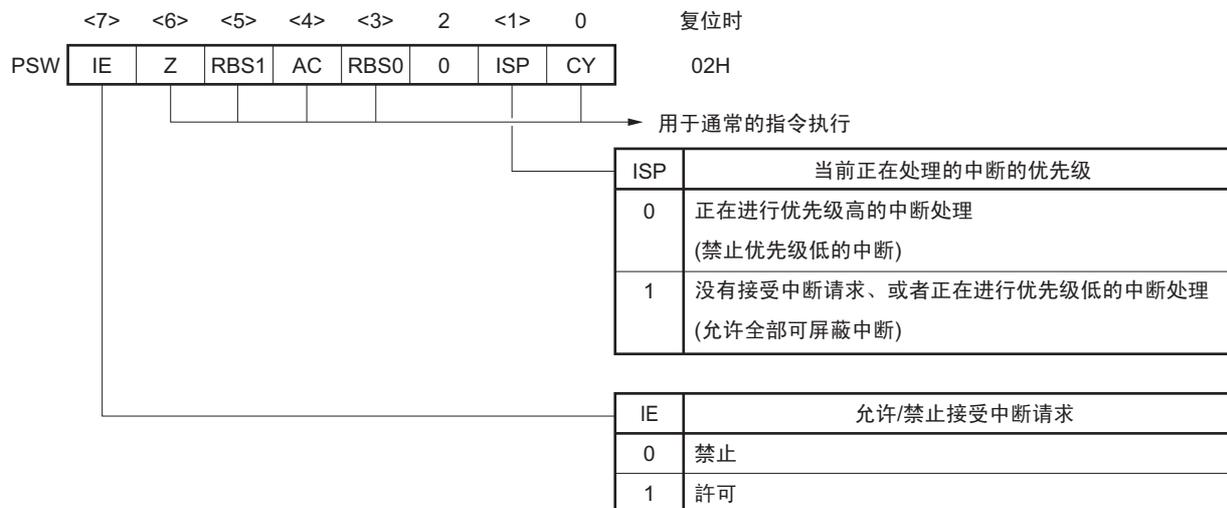
(5) 程序状态字 (PSW)

程序状态字是保存指令的执行结果和中断请求的当前状态的寄存器。PSW 映射设置允许 / 禁止可屏蔽中断的 IE 标志和控制多重中断处理的 ISP 标志。

除了可以 8 位单位读 / 写 PSW 外，还可通过位操作指令和专用指令 (EI 和 DI) 操作 PSW。另外，在接受向量中断请求和执行 BRK 指令时，PSW 的内容自动保存到堆栈，并且 IE 标志复位为“0”。在接受可屏蔽中断请求时，将已接受中断的优先级指定标志的内容传送到 ISP 标志。还可通过 PUSH PSW 指令将 PSW 的内容保存到堆栈，并且可通过 RETI、RETB 和 POP PSW 指令将 PSW 内容从堆栈中恢复。

在产生复位信号后，PSW 变为“02H”。

图 16-6 程序状态字的格式



16.4 中断处理的运行

16.4.1 可屏蔽中断请求的接受运行

在中断请求标志置“1”并且该中断请求的屏蔽（MK）标志清“0”时，为可接受可屏蔽中断请求的状态。在中断允许状态（IE 标志为“1”时）下可接受向量中断请求。但是，在进行高优先级的中断处理期间（ISP 标志为“0”时），不能接受指定为低优先级的中断请求。

在产生可屏蔽中断请求到执行向量中断处理的时间如表 16-4 所示。

中断请求的接受时序请参照图 16-8 和图 16-9。

表 16-4 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
XXPR=0 时	7 个时钟	32 个时钟
XXPR=1 时	8 个时钟	33 个时钟

注 在除法运算指令前产生中断请求时，等待时间最长。

备注 1 个时钟：1/f_{CPU}（f_{CPU}：CPU 时钟）

如果同时产生多个可屏蔽中断请求，就从通过优先级指定标志指定为高优先级的中断请求开始接受。如果通过优先级指定标志指定的优先级相同，就从默认优先级高的中断请求开始接受。

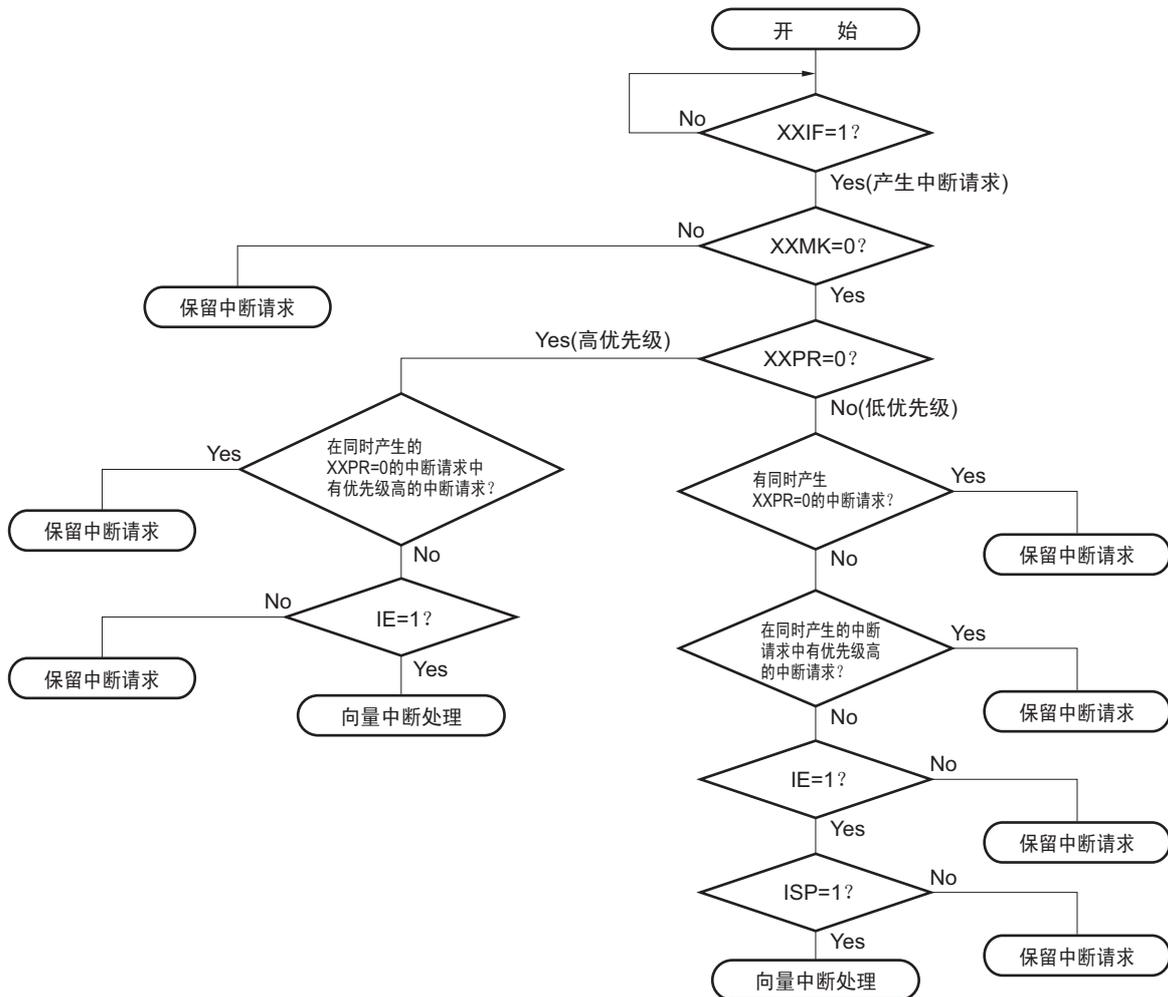
一旦被保留的中断请求进入可接受状态，就可进行接受。

接受中断请求的算法如图 16-7 所示。

一旦接受可屏蔽中断请求，就按先程序状态字（PSW）再程序计数器（PC）的顺序将其内容保存到堆栈，并且将 IE 标志复位为“0”，将已接受中断的优先级指定标志的内容传送到 ISP 标志。而且，还将根据各中断请求决定的向量表中的数据加载到 PC 后进行转移。

可通过 RETI 指令，从中断恢复。

图 16-7 中断请求的接受处理算法



XXIF: 中断请求标志

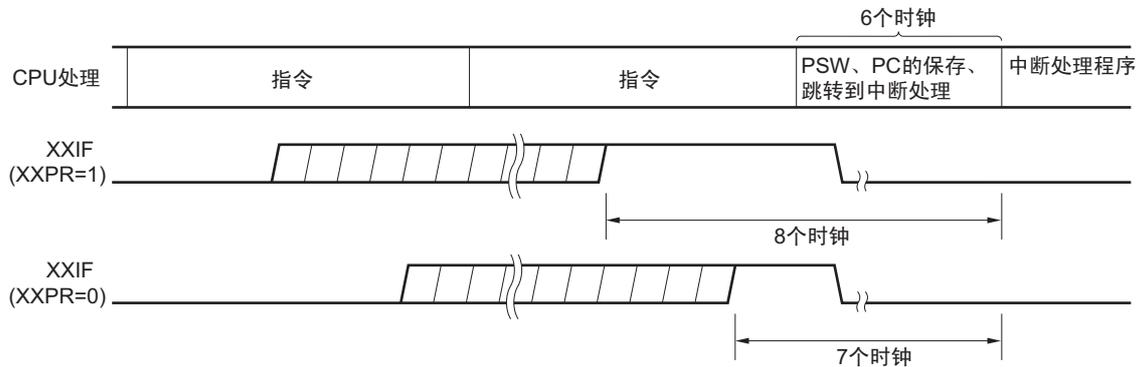
XXMK: 中断屏蔽标志

XXPR: 优先级指定标志

IE: 控制可屏蔽中断请求接受的标志 (1= 允许、0= 禁止)

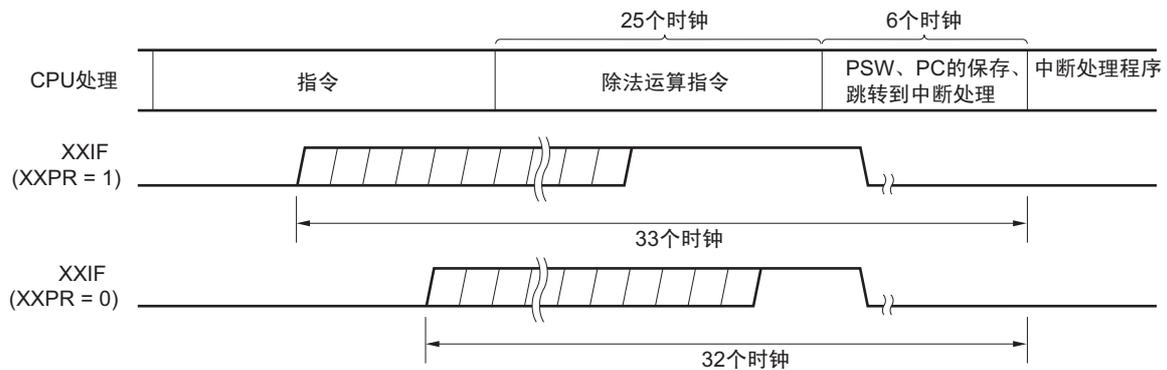
ISP: 表示当前正在处理的中断的优先级标志 (0= 正在进行高优先级的中断处理、1= 没有接受中断请求, 或者正在进行低优先级的中断处理。)

图 16-8 中断请求的接受时序（最短时间）



备注 1 个时钟: $1/f_{\text{CPU}}$ (f_{CPU} : CPU 时钟)

图 16-9 中断请求的接受时序（最长时间）



备注 1 个时钟: $1/f_{\text{CPU}}$ (f_{CPU} : CPU 时钟)

16.4.2 软件中断请求的接受运行

通过执行 BRK 指令接受软件中断请求。不可禁止软件中断。

一旦接受软件中断请求，就按先程序状态字（PSW）再程序计数器（PC）的顺序将其内容保存到堆栈，并且将 IE 标志复位为“0”，将向量表（003EH、003FH）的内容加载到 PC 后进行转移。

可通过 RETB 指令，从软件中断恢复。

注意 禁止使用 RETI 指令从软件中断恢复。

16.4.3 多重中断处理

将在中断处理期间又接受其他中断请求的情况称为多重中断。

只有在中断请求接受允许状态（IE=1）下才能发生多重中断。在接受中断请求后，进入中断请求接受禁止状态（IE=0）。因此，要允许多重中断时，必须在中断处理期间通过 EI 指令将 IE 标志置“1”，设置为中断允许状态。

另外，即使在中断允许状态，也有可能不允许多重中断。这是因为受了中断优先级的控制。中断优先级有两种，分别是默认优先级和可编程优先级。通过控制可编程优先级执行多重中断的控制。

在中断允许状态下产生与当前正在处理的中断的优先级相同的中断请求，或者高于当前正在处理的中断的优先级的中断请求时，可作为多重中断接受。如果是产生低于当前正在处理的中断的优先级的中断请求，就不能作为多重中断接受。

保留因禁止中断或者优先级低而未被允许的多重中断的中断请求。在当前的中断处理结束并且至少执行 1 条主处理的指令后，可接受被保留的中断请求。

可多重中断的中断请求间关系如表 16-5 所示，多重中断的示例如图 16-10 所示。

表 16-5 在中断处理期间可多重中断的中断请求间关系

多重中断请求		可屏蔽中断请求				软件中断请求
		PR=0		PR=1		
		IE=1	IE=0	IE=1	IE=0	
正在处理的中断						
可屏蔽中断	ISP=0	○	X	X	X	○
	ISP=1	○	X	○	X	○
软件中断		○	X	○	X	○

备注 1. ○：可多重中断

2. X：不可多重中断

3. ISP 和 IE 为 PSW 中的标志。

ISP=0：正在进行高优先级的中断处理

ISP=1：没有接受中断请求，或者正在进行低优先级的中断处理。

IE=0：禁止接受中断请求

IE=1：允许接受中断请求

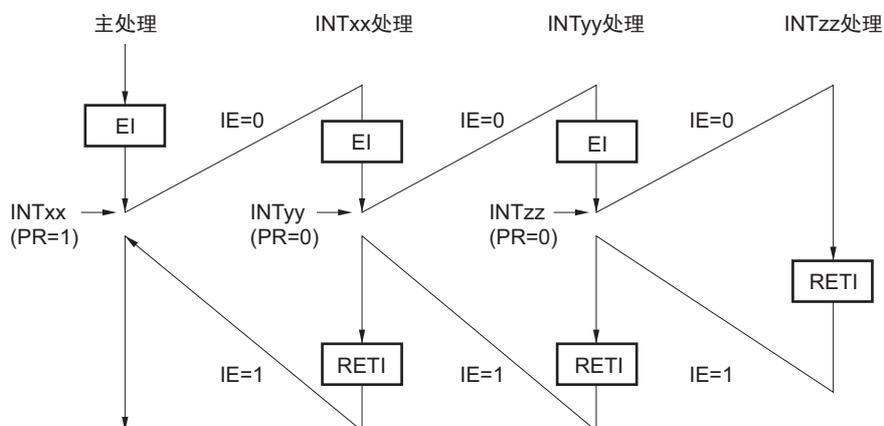
4. PR 为 PR0L、PR0H、PR1L 和 PR1H 中的标志。

PR=0：高优先级

PR=1：低优先级

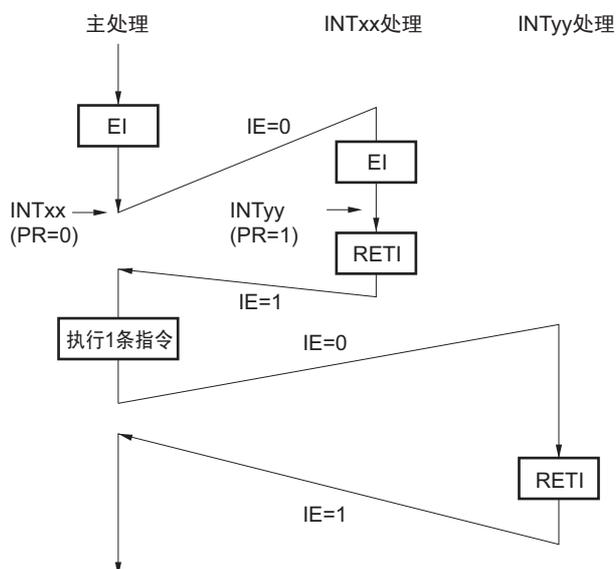
图 16-10 多重中断的示例 (1/2)

示例 1. 发生 2 次多重中断的示例



在中断 INTxx 处理期间，接受 2 次中断请求 INTyy 和 INTzz 后，发生多重中断。在接受各中断请求前，必须通过发行 EI 指令，进入中断请求接受允许状态。

示例 2. 因优先级控制而没有发生多重中断的示例



因为在中断 INTxx 处理期间产生的中断请求 INTyy 的中断优先级低于 INTxx，所以不接受中断请求 INTyy，也不发生多重中断。但是，保留中断请求 INTyy，并且在执行 1 条主处理指令后可接受中断请求 INTyy。

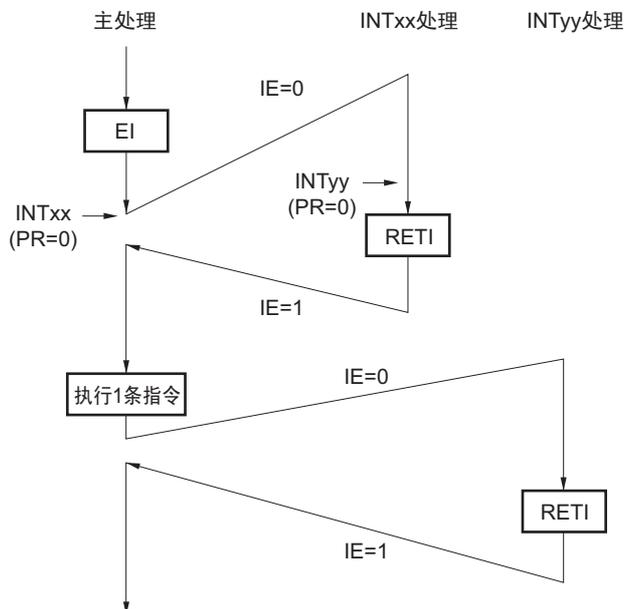
PR=0: 高优先级

PR=1: 低优先级

IE=0: 禁止接受中断请求

图 16-10 多重中断的示例 (2/2)

示例 3. 因未允许中断而没有发生多重中断的示例



因为未允许中断 INTxx 处理期间的中断（未发行 EI 指令），所以不接受中断请求 INTyy，也不发生多重中断。但是，保留中断请求 INTyy，并且在执行 1 条主处理指令后可接受中断请求 INTyy。

PR=0: 高优先级

IE=0: 禁止接受中断请求

16.4.4 中断请求的保留

在某些指令的执行期间，即使产生中断请求，也保留接受该中断请求，直到下一条指令执行结束。这类指令（中断请求的保留指令）如下所示。

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW. bit, CY
- MOV1 CY, PSW. bit
- AND1 CY, PSW. bit
- OR1 CY, PSW. bit
- XOR1 CY, PSW. bit
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW. bit, \$addr16
- BF PSW. bit, \$addr16
- BTCLR PSW. bit, \$addr16
- EI
- DI
- IF0L、IF0H、IF1L、IF1H、MK0L、MK0H、MK1L、MK1H、PROL、PROH、PR1L和PR1H寄存器的操作指令

注意 BRK 指令不为上述中断请求的保留指令。但是，在通过执行 BRK 指令启动的软件中断中，IE 标志被清“0”。因此，即使在执行 BRK 指令期间产生可屏蔽中断请求，也不接受该中断请求。

中断请求的保留时序如图 16-11 所示。

图 16-11 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: 除中断请求的保留指令以外的指令
3. XXIF（中断请求）的运行不受 XXPR（优先级）值的影响。

第 17 章 待机功能

17.1 待机功能和结构

17.1.1 待机功能

待机功能用于降低系统的工作电流，有以下两种模式。

(1) HALT 模式

通过执行 HALT 指令进入 HALT 模式。HALT 模式是停止 CPU 的运行时钟的模式。如果在设置 HALT 模式前高速系统时钟振荡电路、高速内部振荡电路、低速内部振荡电路运行，设置后的各振荡电路的时钟持续振荡。HALT 模式可降低的工作电流不如 STOP 模式多，适用于通过中断请求立即重新开始处理或者频繁进行间歇运行等情况。

(2) STOP 模式

通过执行 STOP 指令进入 STOP 模式。STOP 模式是通过停止高速系统时钟振荡电路和高速内部振荡电路来停止整个系统运行的模式。STOP 模式可大幅降低 CPU 的工作电流。

可通过中断请求解除 STOP 模式，因此，也能用于间歇运行。但是，在 X1 时钟时，解除 STOP 模式后需要确保振荡稳定的等待时间，因此，想要通过中断请求立即开始处理时，应选择 HALT 模式。

在这两种模式下，保存设置为待机模式前的寄存器、标志和数据存储器的所有内容。而且，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. 必须在停止通过主系统时钟运行的外围硬件的运行后，执行 STOP 指令，才能进行 STOP 模式的转移。

2. 为了降低 A/D 转换器的工作电流，必须将 A/D 转换器模式寄存器 (ADM) 的 bit7 (ADCS) 和 bit0 (ADCE) 清“0”，并且在停止 A/D 转换运行后执行 STOP 指令。

17.1.2 待机功能的控制寄存器

待机功能的控制寄存器有以下两种。

- 振荡稳定时间计数器状态寄存器（OSTC）
- 振荡稳定时间选择寄存器（OSTS）

注意 有关控制时钟的运行/停止以及切换的寄存器，请参照“第 5 章 时钟发生电路”。

(1) 振荡稳定时间计数器状态寄存器（OSTC）

振荡稳定时间计数器状态寄存器表示 X1 时钟振荡稳定时间计数器的计数状态。在 CPU 时钟为高速内部振荡时钟并且 X1 时钟开始振荡时，可确认 X1 时钟的振荡稳定时间。

通过 1 位或 8 位存储器操作指令读 OSTC。

在产生复位信号后（通过 $\overline{\text{RESET}}$ 输入、POC、LVI 和 WDT 进行的复位）、执行 STOP 指令或者将 MSTOP（MOC 寄存器的 bit7）置“1”，OSTC 变为“00H”。

图 17-1 振荡稳定时间计数器状态寄存器（OSTC）的格式

地址：FFA3H 复位后：00H R

符号	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	振荡稳定时间的状态	
					$\geq 2^{11}/f_X$	$\geq 204.8\mu\text{s}$
1	0	0	0	0	$\geq 2^{11}/f_X$	$\geq 204.8\mu\text{s}$
1	1	0	0	0	$\geq 2^{13}/f_X$	$\geq 819.2\mu\text{s}$
1	1	1	0	0	$\geq 2^{14}/f_X$	$\geq 1.64\text{ms}$
1	1	1	1	0	$\geq 2^{15}/f_X$	$\geq 3.27\text{ms}$
1	1	1	1	1	$\geq 2^{16}/f_X$	$\geq 6.55\text{ms}$

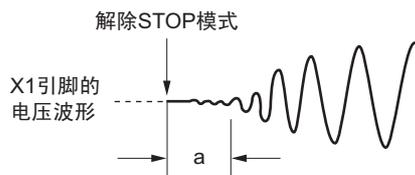
注意 1. 经过上述时间后，从 MOST11 开始逐次置“1”，并且一直保持“1”。

2. 振荡稳定时间计数器只能在 OSTS 设置的振荡稳定时间内进行计数。在 CPU 时钟为高速内部振荡时钟时，进入 STOP 模式。如果要解除该模式，必须如下设置 OSTS 的振荡稳定时间。

- 期待的 OSTC 的振荡稳定时间 \leq 通过 OSTS 设置的振荡稳定时间

因此，必须注意，解除 STOP 模式后的 OSTC 的设置状态，只能为 OSTS 设置的振荡稳定时间内的状态。

3. X1 时钟的振荡稳定时间不包括开始时钟振荡前的时间（下图中“a”所表示的部分）。



备注 f_X : X1 时钟振荡频率

(2) 振荡稳定时间选择寄存器 (OSTS)

振荡稳定时间选择寄存器选择解除 STOP 模式时的 X1 时钟的振荡稳定时间。

如果选择 X1 时钟作为 CPU 时钟，在解除 STOP 模式后，需等待 OSTS 设置的时间。

如果选择高速内部振荡时钟作为 CPU 时钟，在解除 STOP 模式后，必须通过 OSTC 确认是否经过振荡稳定时间。OSTC 可在事先通过 OSTS 设置的时间内进行确认。

通过 8 位存储器操作指令设置 OSTS。

在产生复位信号后，OSTS 变为“05H”。

图 17-2 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFA4H 复位后: 05H 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	
				$f_X=10\text{MHz}$
0	0	1	$2^{11}/f_X$	204.8 μs
0	1	0	$2^{13}/f_X$	819.2 μs
0	1	1	$2^{14}/f_X$	1.64ms
1	0	0	$2^{15}/f_X$	3.27ms
1	0	1	$2^{16}/f_X$	6.55ms
上述以外			禁止设置	

注意 1. 如果要在 CPU 时钟为 X1 时钟时转移到 STOP 模式，必须在执行 STOP 指令前设置 OSTS。

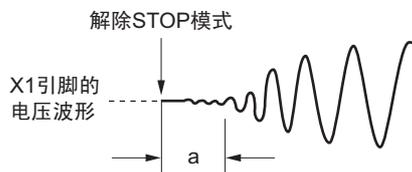
2. 在 X1 时钟的振荡稳定时间内，不能更改 OSTS 寄存器。

3. 振荡稳定时间计数器只能在 OSTS 设置的振荡稳定时间内进行计数。在 CPU 时钟为高速内部振荡时钟时，进入 STOP 模式。如果要解除该模式，必须如下设置 OSTS 的振荡稳定时间。

• 期待的 OSTC 的振荡稳定时间 \leq 通过 OSTS 设置的振荡稳定时间

因此，必须注意，解除 STOP 模式后的 OSTC 的设置状态，只能为 OSTS 设置的振荡稳定时间内的状态。

4. X1 时钟的振荡稳定时间不包括开始时钟振荡前的时间（下图中“a”所表示的部分）。



备注 f_X : X1 时钟振荡频率

17.2 待机功能的运行

17.2.1 HALT 模式

(1) HALT 模式

通过执行 HALT 指令设置为 HALT 模式。只要设置前的 CPU 时钟是高速系统时钟或者高速内部振荡时钟，就可设置为 HALT 模式。

HALT 模式下的运行状态如下所示。

表 17-1 HALT 模式下的运行状态

HALT 模式的设置		在以主系统时钟进行 CPU 运行时执行 HALT 指令的情况		
		以高速内部振荡时钟 (f_{RH}) 进行 CPU 运行时	以 X1 时钟 (f_X) 进行 CPU 运行时	以外部主系统时钟 (f_{EXCLK}) 进行 CPU 运行时
系统时钟		停止向 CPU 提供时钟		
主系统时钟	f_{RH}	继续运行 (不能停止)	保持设置 HALT 模式前的状态	
	f_X	保持设置 HALT 模式前的状态	继续运行 (不能停止)	保持设置 HALT 模式前的状态
	f_{EXCLK}	通过输入外部时钟运行或者停止		继续运行 (不能停止)
f_{RL}		保持设置 HALT 模式前的状态		
CPU		停止运行		
闪存				
RAM		保持设置 HALT 模式前的状态		
端口 (锁存器)				
16 位定时器 / 事件计数器	00	可运行		
8 位定时器 / 事件计数器	50			
	51			
8 位定时器	H0			
	H1			
时钟定时器				
看门狗定时器		可运行 但是，在通过选项字节设置为“可通过软件停止低速内部振荡器”时，停止向看门狗定时器提供时钟。		
A/D 转换器		可运行		
串行接口	UART0			
	UART6			
	CSI10			
	IIC0			
上电清除功能				
低电压检测功能				
外部中断				

备注 1. f_{RH} : 高速内部振荡时钟

f_{EXCLK} : 外部主系统时钟

f_X : X1 时钟

f_{RL} : 低速内部振荡时钟

2. 配置的功能因产品而不同。参照“1.6 框图”、“1.7 功能概要”。

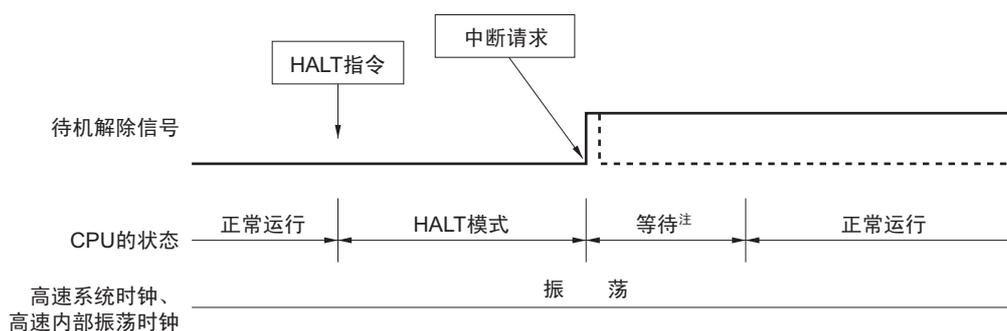
(2) HALT 模式的解除

可通过以下两种解除源解除 HALT 模式。

(a) 通过非屏蔽中断请求解除 HALT 模式

一旦产生非屏蔽中断请求，就解除 HALT 模式。如果为中断接受允许状态，就执行向量中断处理。如果为中断接受禁止状态，就执行下一条地址指令。

图 17-3 通过产生 HALT 模式的中断请求解除



注 等待时间如下所示。

- 执行向量中断处理时：11 ~ 12 个时钟
- 不执行向量中断处理时：4 ~ 5 个时钟

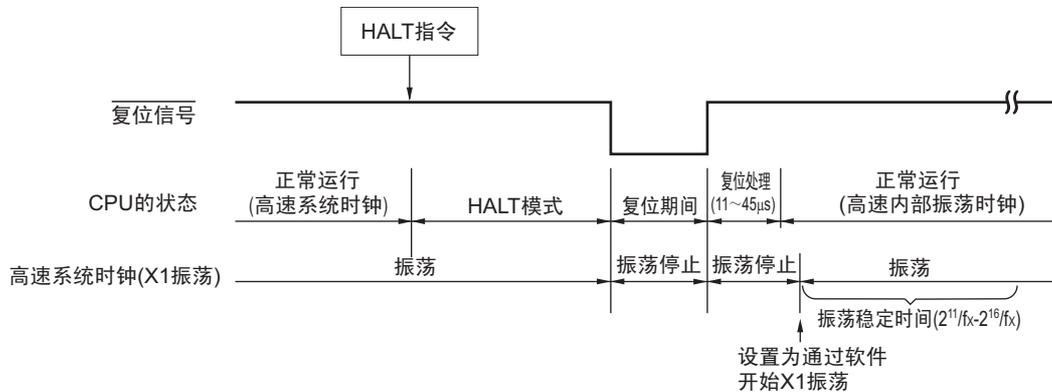
注意 虚线表示接受解除待机的中断请求后的情况。

(b) 在产生复位信号后，解除 HALT 模式

一旦产生复位信号，就解除 HALT 模式。与通常的复位运行一样在转移到复位向量地址后，开始执行程序。

图 17-4 通过 HALT 模式的复位解除

(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时钟的情况

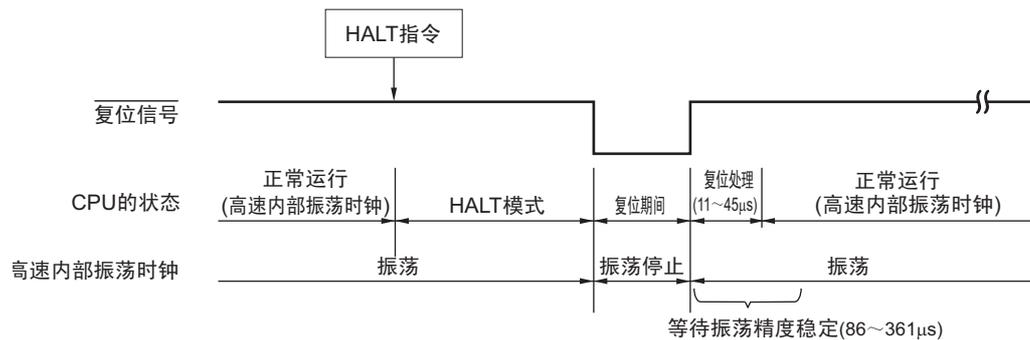
备注 f_x : X1 时钟振荡频率

表 17-2 与 HALT 模式中的中断请求对应的运行

解除源	MKXX	PRXX	IE	ISP	运行
可屏蔽中断请求	0	0	0	x	执行下一条地址指令
	0	0	1	x	执行中断处理
	0	1	0	1	执行下一个地址指令
	0	1	x	0	
	0	1	1	1	执行中断处理务
	1	x	x	x	保持 HALT 模式
复位	—	—	x	x	复位处理

x: 忽略

17.2.2 STOP 模式

(1) STOP 模式的设置和运行状态

通过执行 STOP 指令设置为 STOP 模式。只在设置前的 CPU 时钟为主系统时钟时可设置为 STOP 模式。

注意 因为中断请求信号用于解除待机模式，所以在产生中断请求标志置位或者中断屏蔽标志复位的中断源时，即使进入待机模式也立即被解除。因此，在 STOP 模式中，执行 STOP 指令后立即进入 HALT 模式，并且在等待振荡稳定时间选择寄存器（OSTS）设置的时间后，返回运行模式。

STOP 模式下的运行状态如下所示。

表 17-3 STOP 模式下的运行状态

STOP 模式的设置		在以主系统时钟进行 CPU 运行时执行 STOP 指令的情况		
		以高速内部振荡时钟 (f_{RH}) 进行 CPU 运行时	以 X1 时钟 (f_X) 进行 CPU 运行时	以外部主系统时钟 (f_{EXCLK}) 进行 CPU 运行时
系统时钟		停止向 CPU 提供时钟		
主系统时钟	f_{RH}	停止运行		
	f_X	停止运行		
	f_{EXCLK}	输入无效		
f_{RL}		保持设置 STOP 模式前的状态		
CPU		停止运行		
闪存		停止运行		
RAM		保持设置 STOP 模式前的状态		
端口（锁存器）		保持设置 STOP 模式前的状态		
16 位定时器 / 事件计数器	00	停止运行		
8 位定时器 / 事件计数器	50 注	只在选择 TI50 作为计数时钟时可运行		
	51 注	只在选择 TI51 作为计数时钟时可运行		
8 位定时器	H0	当 8 位定时器 / 事件计数器 50 运行时，只在选择 TM50 输出作为计数时钟时可运行。		
	H1	只在选择 f_{RL} , $f_{RL}/2^7$, $f_{RL}/2^9$ 作为计数时钟时可运行		
时钟定时器		停止运行		
看门狗定时器		可运行 但是，在通过选项字节设置为“可通过软件停止低速内部振荡器”时，停止向看门狗定时器提供时钟。		
A/D 转换器		停止运行		
串行接口	UART0	当 8 位定时器 / 事件计数器 50 运行时，只在选择 TM50 输出作为串行时钟时可运行。		
	UART6	当 8 位定时器 / 事件计数器 50 运行时，只在选择 TM50 输出作为串行时钟时可运行。		
	CSI10 注	只在选择外部时钟作为串行时钟时可运行		
	IIC0	停止运行		
上电清除功能		可运行		
低电压检测功能		可运行		
外部中断		可运行		

注 在 STOP 模式中，不能通过外围硬件引脚的外部时钟开始运行。

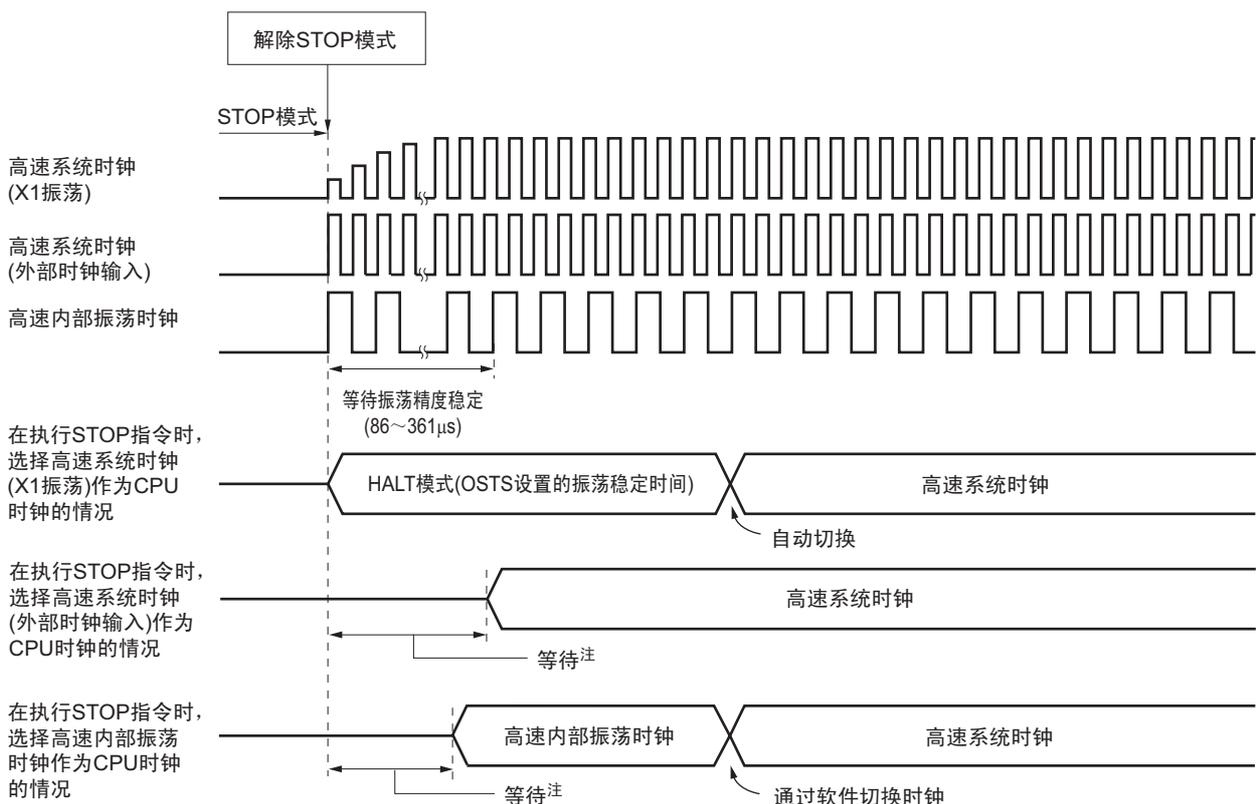
备注 1. f_{RH} : 高速内部振荡时钟 f_{EXCLK} : 外部主系统时钟
 f_X : X1 时钟 f_{RL} : 低速内部振荡时钟

2. 搭载的功能因产品而不同。参照“1.6 框图”、“1.7 功能概要”。

- 注意 1. 如果要在解除 STOP 模式后使用在 STOP 模式中停止运行以及选择时钟振荡停止的外围硬件，必须重新启动外围硬件。
2. 即使通过选项字节选择“可通过软件停止低速内部振荡器”，在 STOP 模式中，低速内部振荡时钟仍保持 STOP 模式设置前的状态。如果要在 STOP 模式中停止低速内部振荡时钟，必须先通过软件停止低速内部振荡器的振荡，再执行 STOP 指令。
3. 如果要在以高速系统时钟（X1 振荡）进行 CPU 运行时，缩短解除 STOP 模式后的振荡稳定时间，必须在执行 STOP 指令前按照以下步骤将 CPU 时钟切换到高速内部振荡时钟。
- ①将 RSTOP 置“0”（开始高速内部振荡器的振荡）→ ②将 MCM0 置“0”（将 CPU 从 X1 振荡切换到高速内部振荡）→ ③确认 MCS 为“0”（确认 CPU 时钟）→ ④确认 RSTS 为“1”（确认高速内部振荡运行）→ ⑤执行 STOP 指令
- 要在解除 STOP 模式后，将 CPU 时钟从高速内部振荡时钟切换到高速系统时钟（X1 振荡）时，必须先通过振荡稳定时间计数器状态寄存器（OSTC）确认振荡稳定时间。
4. 必须在确认高速内部振荡器稳定运行（RSTS=1）后，执行 STOP 指令。

(2) STOP 模式的解除

图 17-5 解除 STOP 模式时的运行时序（通过非屏蔽中断请求解除的情况）



注 等待时间如下所示。

- 执行向量中断处理时：17 ~ 18 个时钟
- 不执行向量中断处理时：11 ~ 12 个时钟

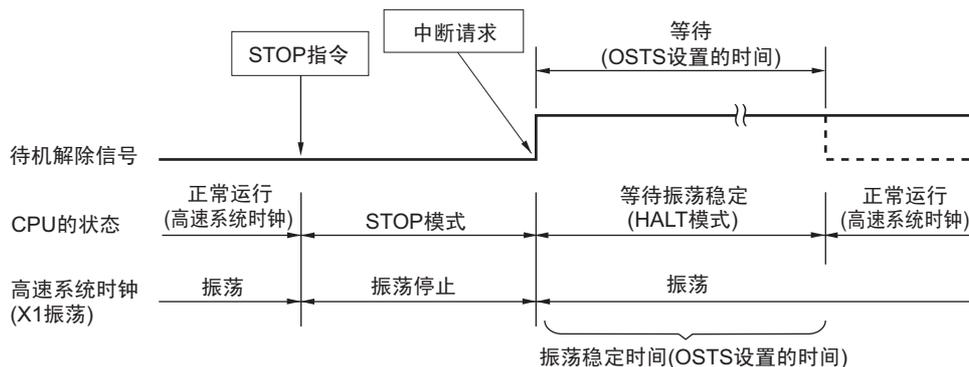
可通过以下两种解除源解除 STOP 模式。

(a) 通过非屏蔽中断请求解除 STOP 模式

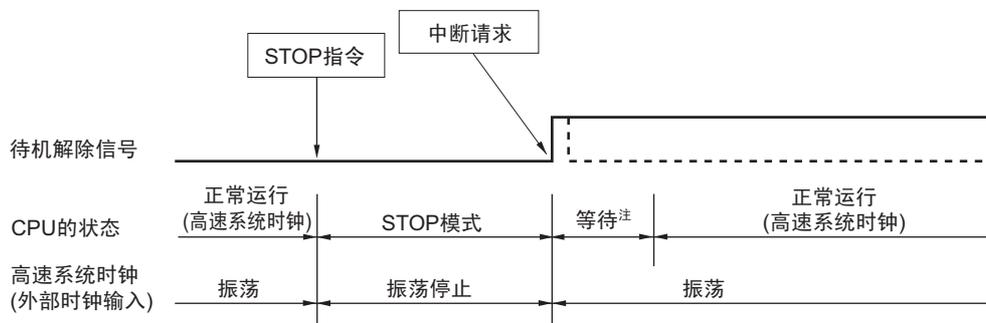
一旦产生非屏蔽中断请求，就解除 STOP 模式。如果为中断接受允许状态，就执行向量中断处理；如果为中断接受禁止状态，就执行下一条地址指令。

图 17-6 通过产生 STOP 模式的中断请求解除 (1/2)

(1) CPU 时钟为高速系统时钟 (X1 振荡) 的情况



(2) CPU 时钟为高速系统时钟 (外部时钟输入) 的情况



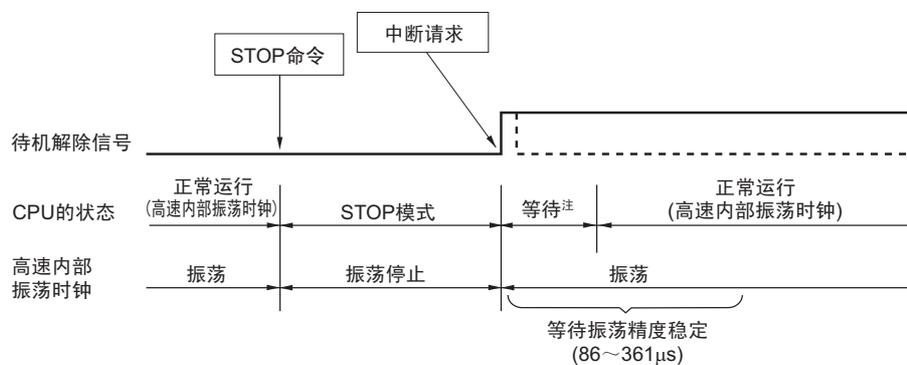
注 待机时间如下所示。

- 执行向量中断处理时：17~18个时钟
- 不执行向量中断处理时：11~12个时钟

备注 虚线表示接受解除待机的中断请求后的情况。

图 17-6 通过产生 STOP 模式的中断请求解除 (2/2)

(3) CPU 时钟为高速内部振荡时钟的情况



注 待机时间如下所示。

- 执行向量中断处理时：17~18个时钟
- 不执行向量中断处理时：11~12个时钟

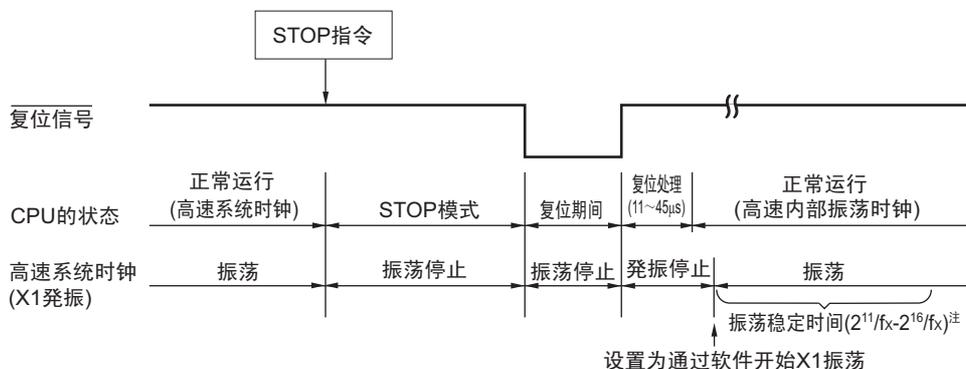
备注 虚线表示接受解除待机的中断请求后的情况。

(b) 在产生复位信号后，解除 STOP 模式

一旦产生复位信号，就解除 STOP 模式。与通常的复位运行一样在转移到复位向量地址后，开始执行程序。

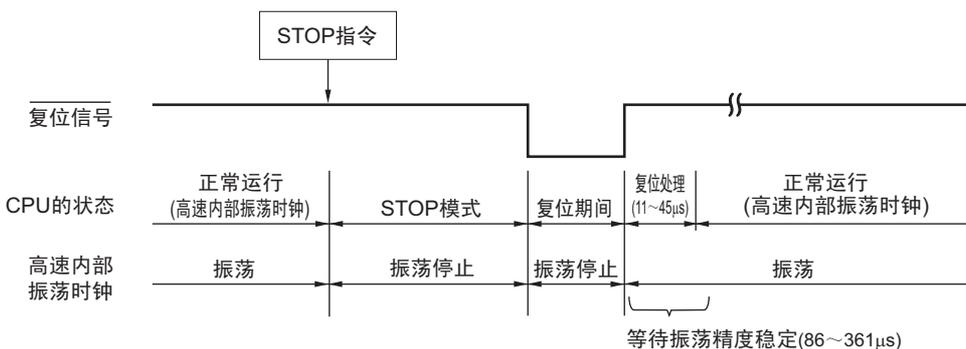
图 17-7 通过 STOP 模式的复位解除

(1) CPU 时钟为高速系统时钟的情况



注 在外部主系统时钟 (f_{EXCLK}) 用作高速系统时钟时, 不需要振荡稳定时间。

(2) CPU 时钟为高速内部振荡时钟的情况



备注 f_x : X1 时钟振荡频率

表 17-4 与 STOP 模式中的中断请求对应的运行

解除源	MKXX	PRXX	IE	ISP	运行
可屏蔽中断请求	0	0	0	x	执行下一条地址指令
	0	0	1	x	执行中断处理
	0	1	0	1	执行下一条地址指令
	0	1	x	0	
	0	1	1	1	执行中断处理
	1	x	x	x	保持 STOP 模式
复位	—	—	x	x	复位处理

x: 忽略

第 18 章 复位功能

产生复位信号的方法有以下 4 种。

1. 通过 $\overline{\text{RESET}}$ 引脚进行的外部复位输入
2. 通过看门狗定时器的程序失控检测进行的内部复位
3. 通过上电清除（POC）电路的电源电压与检测电压的比较进行的内部复位
4. 通过低电源检测电路（LVI）的电源电压与检测电压的比较进行的内部复位

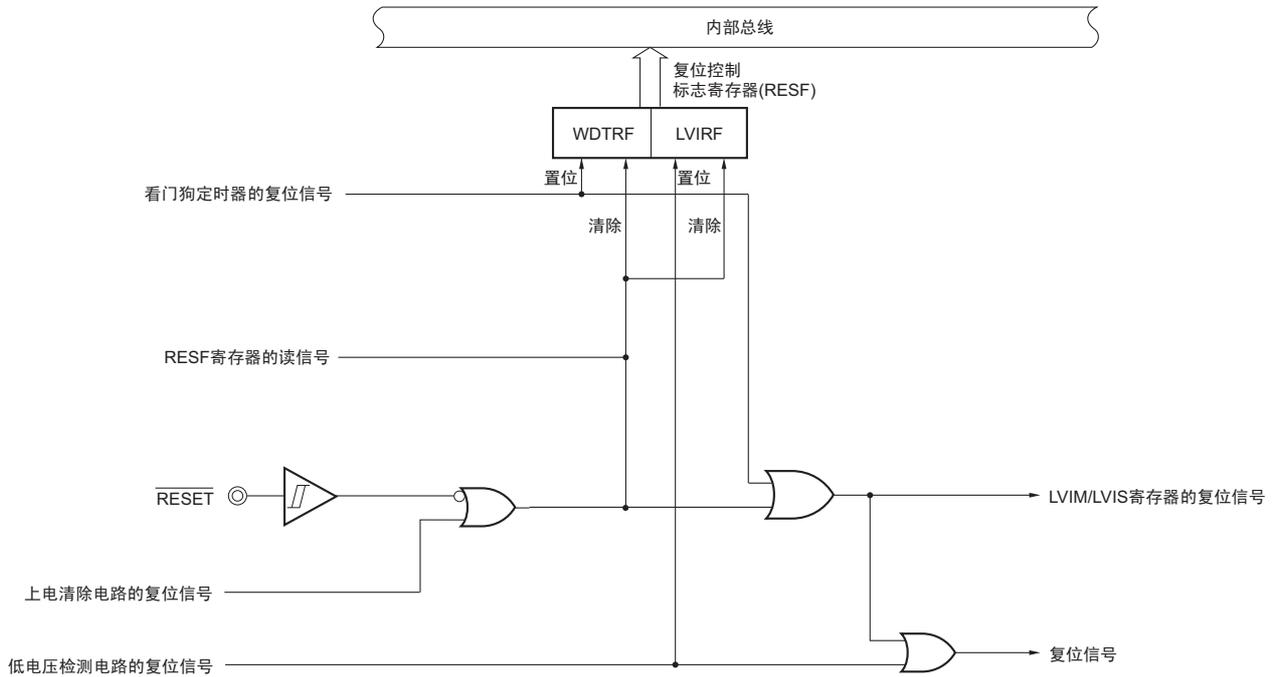
外部复位与内部复位一样，在产生复位信号后，从地址 0000H 和 0001H 中的起始地址开始执行程序。

通过 $\overline{\text{RESET}}$ 引脚的低电平输入、看门狗定时器的程序失控检测、或者 POC 电路和 LVI 电路的电压检测，发生复位，并且各硬件的状态如表 18-1 和表 18-2 所示。在产生复位信号期间和复位解除后的振荡稳定时间内，各引脚的状态为高阻抗状态。

如果向 $\overline{\text{RESET}}$ 引脚输入低电平，就发生复位；如果向 $\overline{\text{RESET}}$ 引脚输入高电平，就解除复位，并且在复位处理后通过高速内部振荡时钟开始程序的执行。自动解除通过看门狗定时器进行的复位，并且在复位处理后通过高速内部振荡时钟开始程序的执行（参考图 18-2 ~ 图 18-4）。在执行通过 POC 电路和 LVI 电路的电源检测进行的复位后，如果 $V_{DD} \geq V_{POC}$ 或者 $V_{DD} \geq V_{LVI}$ 时，就解除复位，并且在复位处理后通过高速内部振荡时钟开始执行程序（参照“第 19 章 上电清除电路”、“第 20 章 低电压检测电路”）。

- 注意 1. 在进行外部复位时，必须向 $\overline{\text{RESET}}$ 引脚输入不低于 10 μs 的低电平。
2. 在产生复位信号期间，X1 时钟、高速内部振荡时钟和低速内部振荡时钟停止振荡。外部主系统时钟的输入无效。
 3. 通过复位解除 STOP 模式时，复位输入期间保持 STOP 模式中的内容。但是，端口引脚变为高阻抗。

图 18-1 复位功能的框图



注意 在进行 LVI 电路的内部复位时，不对 LVI 电路进行复位。

- 备注 1. LVIM: 低电压检测寄存器
2. LVIS: 低电压检测电平选择寄存器

图 18-2 通过 RESET 输入进行复位的时序

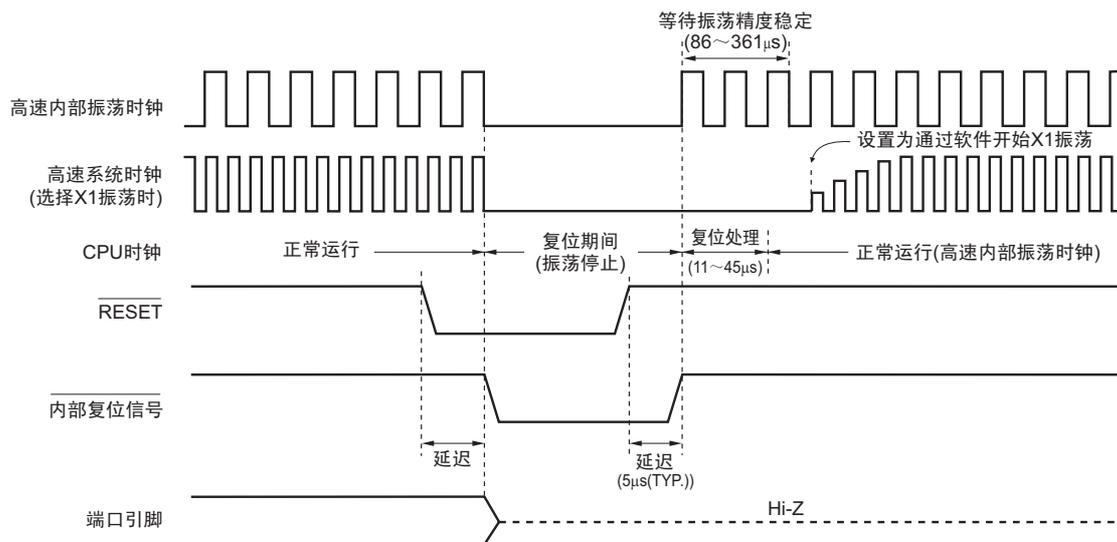
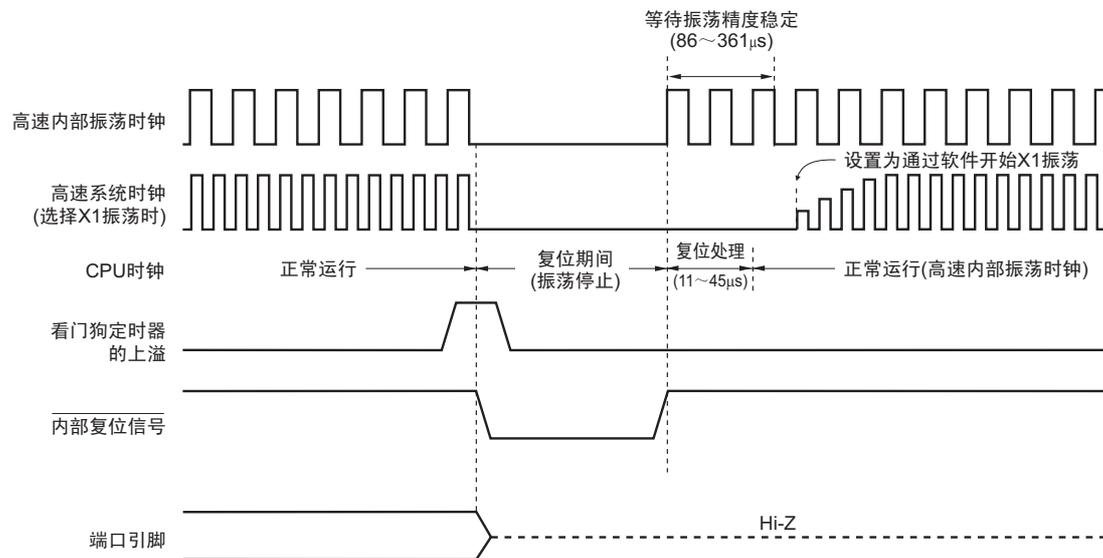
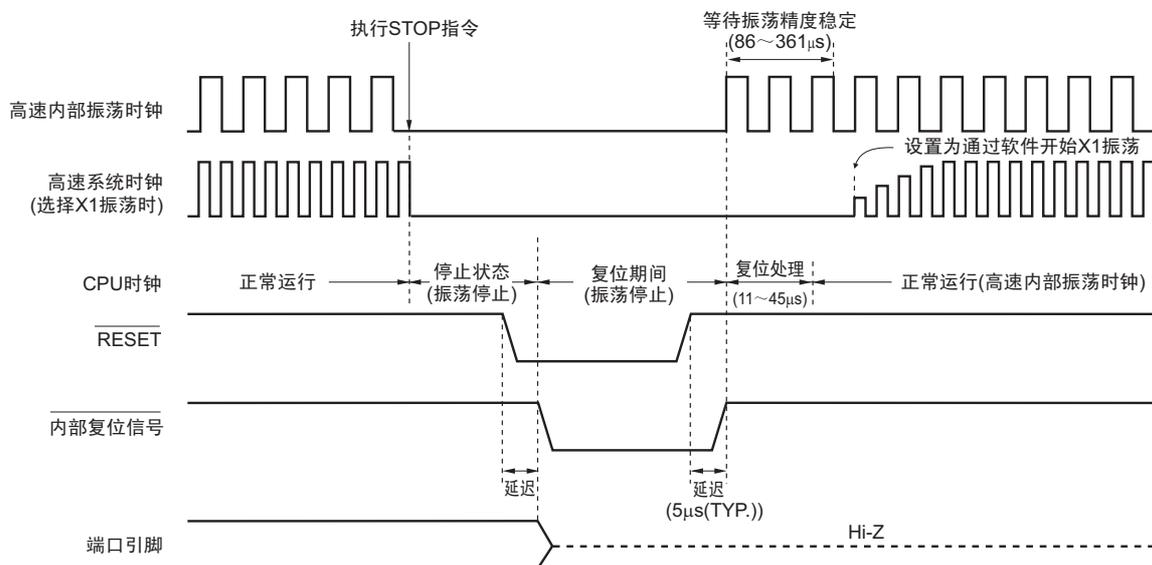


图 18-3 通过看门狗定时器上溢进行的时序



注意 在进行看门狗定时器的内部复位时，看门狗定时器也被复位。

图 18-4 通过 STOP 模式中的 $\overline{\text{RESET}}$ 输入进行复位的时序

备注 有关上电清除电路和低电压检测电路的复位时序，请参照“第 19 章 上电清除电路”和“第 20 章 低电压检测电路”。

表 18-1 复位期间的运行状态

项目		复位期间
系统时钟		停止向 CPU 提供时钟
主系统时钟	f_{RH}	停止运行
	f_X	停止运行（引脚为输入 / 输出端口模式）
	f_{EXCLK}	时钟的输入无效（引脚为输入 / 输出端口模式）
f_{RL}	停止运行	
CPU		
闪存		
RAM		
端口（锁存器）		停止运行（但是，在高于上电清除检测电压时保持值。）
16 位定时器 / 事件计数器	00	
8 位定时器 / 事件计数器	50	
	51	
8 位定时器	H0	
	H1	
时钟定时器		
看门狗定时器		
A/D 转换器		
串行接口	UART0	
	UART6	
	CSI10	
	IIC0	
上电清除功能		可运行
低电压检测功能		停止运行
外部中断		

备注 1. f_{RH} : 高速内部振荡时钟

f_{EXCLK} : 外部主系统时钟

f_X : X1 时钟

f_{RL} : 低速内部振荡时钟

2. 搭载的功能因产品而不同。参照“1.6 框图”、“1.7 功能概要”。

表 18-2 各硬件接受复位后的状态 (1/3)

硬件		接受复位后的状态注 1
程序计数器 (PC)		设置复位向量表的内容 (0000H、0001H)
堆栈指针 (SP)		不定
程序状态字 (PSW)		02H
RAM	数据存储器	不定注 2
	通用寄存器	不定注 2
端口寄存器 (P0 ~ P4、P6、P7、P12) (输出锁存器)		00H
端口模式寄存器 (PM0 ~ PM4、PM6、PM7、PM12)		FFH
上拉电阻选择寄存器 (PU0、PU1、PU3、PU4、PU7、PU12)		00H
存储容量切换寄存器 (IMS)		CFH 注 3

- 注 1. 在产生复位信号期间和等待振荡稳定时间内的各硬件状态中，只有 PC 的内容不定。其他硬件的状态与复位后的状态相同。
2. 保持待机模式中的复位后的状态。
3. 与内部存储容量无关，复位解除存储容量切换寄存器 (IMS) 后的初始值在 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 全部产品中固定 (IMS=CFH)。因此，复位解除后，复位时的 ROM 区域设置为，必须给各产品设置如表 3-1 所示的值。

表 18-2 各硬件接受复位后的状态 (2/3)

硬件		接受复位后的状态注 ¹
时钟运行模式选择寄存器 (OSCCTL)		00H
处理器时钟控制寄存器 (PCC)		01H
内部振荡模式寄存器 (RCM)		80H
主 OSC 控制寄存器 (MOC)		80H
主时钟模式寄存器 (MCM)		00H
振荡稳定时间计数器状态寄存器 (OSTC)		00H
振荡稳定时间选择寄存器 (OSTS)		05H
16 位定时器 / 事件计数器 00	定时器计数器 00 (TM00)	0000H
	捕捉 / 比较寄存器 000、010 (CR000、CR010)	0000H
	模式控制寄存器 00 (TMC00)	00H
	预分频器模式寄存器 00 (PRM00)	00H
	捕捉 / 比较控制寄存器 00 (CRC00)	00H
	定时器输出控制寄存器 00 (TOC00)	00H
8 位定时器 / 事件计数器 50、51	定时器计数器 50、51 (TM50、TM51)	00H
	比较寄存器 50、51 (CR50、CR51)	00H
	定时器时钟选择寄存器 50、51 (TCL50、TCL51)	00H
	模式控制寄存器 50、51 (TMC50、TMC51)	00H
8 位定时器 H0、H1	比较寄存器 00、10、01、11 (CMP00、CMP10、CMP01、CMP11)	00H
	模式寄存器 (TMHMD0、TMHMD1)	00H
	载波控制寄存器 1 (TMCYC1) 注 ²	00H
时钟定时器	运行模式寄存器 (WTM)	00H
看门狗定时器	允许寄存器 (WDTE)	1AH/9AH 注 ³
A/D 转换器	10 位 A/D 转换结果寄存器 (ADCR)	0000H
	8 位 A/D 转换结果寄存器 (ADCRH)	00H
	模式寄存器 (ADM)	00H
	模拟输入通道指定寄存器 (ADS)	00H
	A/D 端口配置寄存器 (ADPC)	00H
串行接口 UART0	接收缓冲寄存器 0 (RXB0)	FFH
	发送移位寄存器 0 (TXS0)	FFH
	异步串行接口运行模式寄存器 0 (ASIM0)	01H
	异步串行接口接收错误状态寄存器 0 (ASIS0)	00H
	波特率发生器控制寄存器 0 (BRGC0)	1FH

注 1. 在产生复位信号期间和等待振荡稳定时间内的各硬件状态中，只有 PC 的内容不定。其他硬件的状态与复位后的状态相同。

2. 只限 8 位定时器 H1。

3. WDTE 的复位值取决于选项字节的设置。

表 18-2 各硬件接受复位后的状态 (3/3)

硬件		接受复位后的状态注 1
串行接口 UART6	接收缓冲寄存器 6 (RXB6)	FFH
	发送缓冲寄存器 6 (TXB6)	FFH
	异步串行接口运行模式寄存器 6 (ASIM6)	01H
	异步串行接口接收错误状态寄存器 6 (ASIS6)	00H
	异步串行接口发送状态寄存器 6 (ASIF6)	00H
	时钟选择寄存器 6 (CKSR6)	00H
	波特率生成器控制寄存器 6 (BRGC6)	FFH
	异步串行接口控制寄存器 6 (ASICL6)	16H
	输入切换控制寄存器 (ISC)	00H
串行接口 CSI10	发送缓冲寄存器 10 (SOTB10)	00H
	串行 I/O 移位寄存器 10 (SIO10)	00H
	串行运行模式寄存器 10 (CSIM10)	00H
	串行时钟选择寄存器 10 (CSIC10)	00H
串行接口 IIC0	移位寄存器 0 (IIC0)	00H
	控制寄存器 0 (IICC0)	00H
	从属地址寄存器 0 (SVA0)	00H
	时钟选择寄存器 0 (IICCL0)	00H
	功能扩展寄存器 0 (IICX0)	00H
	状态寄存器 0 (IICS0)	00H
	标志寄存器 0 (IICF0)	00H
复位功能	复位控制标志寄存器 (RESF)	00H 注 2
低电压检测电路	低电压检测寄存器 (LVIM)	00H 注 2
	低电压检测电平选择寄存器 (LVIS)	00H 注 2
中断	请求标志寄存器 0L、0H、1L、1H (IF0L、IF0H、IF1L、IF1H)	00H
	屏蔽标志寄存器 0L、0H、1L、1H (MK0L、MK0H、MK1L、MK1H)	FFH
	优先级指定标志寄存器 0L、0H、1L、1H (PR0L、PR0H、PR1L、PR1H)	FFH
	外部中断上升沿允许寄存器 (EGP)	00H
	外部中断下降沿允许寄存器 (EGN)	00H

注 1. 在产生复位信号期间和等待振荡稳定时间内的各硬件状态中，只有 PC 的内容不定。其他硬件的状态与复位后的状态相同。

2. 复位源如下变化。

复位源		RESET 输入	POC 复位	WDT 复位	LVI 复位
RESF	WDTRF 标志	清除 (0)	清除 (0)	设置 (1)	保持
	LVIRF 标志			保持	设置 (1)
LVIM		清除 (00H)	清除 (00H)	清除 (00H)	保持
LVIS					

18.1 复位源的确认寄存器

R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 存在多个内部复位发生源。复位控制标志寄存器 (RESF) 保存是由哪个复位源产生的复位请求。

可通过 8 位存储器操作指令读 RESF。

通过 RESET 输入、上电清除 (POC) 电路进行的复位和读 RESF 的数据, RESF 变为“00H”。

图 18-5 复位控制标志寄存器 (RESF) 的格式

地址: FFACH 复位后: 00H 注 R

符号	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LVIRF

WDTRF	通过看门狗定时器 (WDT) 产生内部复位请求
0	不产生内部复位请求, 或者清除 RESF。
1	产生内部复位请求

LVIRF	通过低电压检测电路产生内部复位请求 (LVI)
0	不产生内部复位请求, 或者除 RESF。
1	产生内部复位请求

注 因复位源而不同。

注意 不能通过 1 位存储器操作指令读取数据。

产生复位请求时的 RESF 状态如表 18-3 所示。

表 18-3 产生复位请求时的 RESF 状态。

复位源	RESET 输入	POC 复位	WDT 复位	LVI 复位
标志				
WDTRF	清除 (0)	清除 (0)	设置 (1)	保持
LVIRF			保持	设置 (1)

第 19 章 上电清除电路

19.1 上电清除电路的功能

上电清除电路（POC）具有以下功能。

- 在接通电源时产生内部复位信号
在设置为 1.59V POC 模式（选项字节：POCMODE=0）时，如果电源电压（ V_{DD} ）大于 $1.59V \pm 0.15V$ ，就解除复位。
在设置为 2.7V/1.59V POC 模式（选项字节：POCMODE=1）时，如果电源电压（ V_{DD} ）大于 $2.7V \pm 0.2V$ ，就解除复位。
- 比较电源电压（ V_{DD} ）和检测电压（ $V_{POC}=1.59V \pm 0.15V$ ），在 $V_{DD} < V_{POC}$ 时，产生内部复位信号。

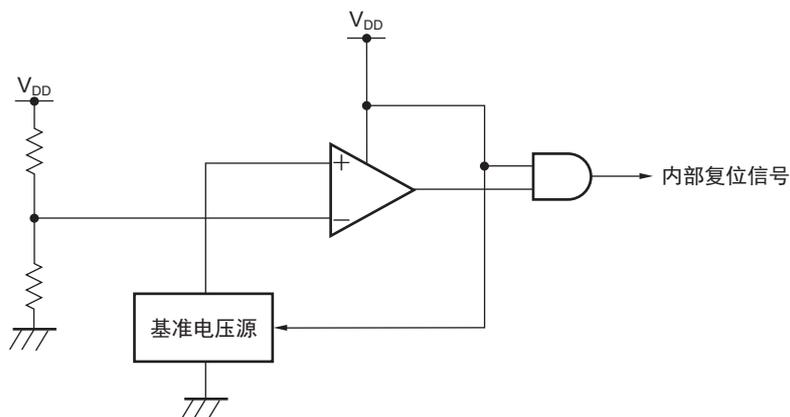
注意 在 POC 电路中产生内部复位信号时，复位控制标志寄存器（RESF）清“00H”。

备注 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 内置产生内部复位信号的多个硬件。在通过看门狗定时器（WDT）或者低电压检测器（LVI）电路产生内部复位信号时，将表示该复位源的标志配置到复位控制标志寄存器（RESF）。在通过 WDT 或者 LVI 产生内部复位信号时，不清除 RESF（00H），但是 RESF 标志被置“1”。有关 RESF 的详细内容，请参照“第 18 章 复位功能”。

19.2 上电清除电路的结构

上电清除电路的框图如图 19-1 所示。

图 19-1 上电清除电路的框图



19.3 上电清除电路的运行

(1) 设置为 1.59V POC 模式时 (选项字节: POCMODE=0)

- 在接通电源时产生内部复位信号, 并且在电源电压 (V_{DD}) 大于检测电压 ($V_{POC}=1.59V\pm 0.15V$) 时, 解除复位。
- 比较电源电压 (V_{DD}) 和检测电压 ($V_{POC}=1.59V\pm 0.15V$), 在 $V_{DD} < V_{POC}$ 时, 产生内部复位信号; 在 $V_{DD} \geq V_{POC}$ 时, 解除复位。

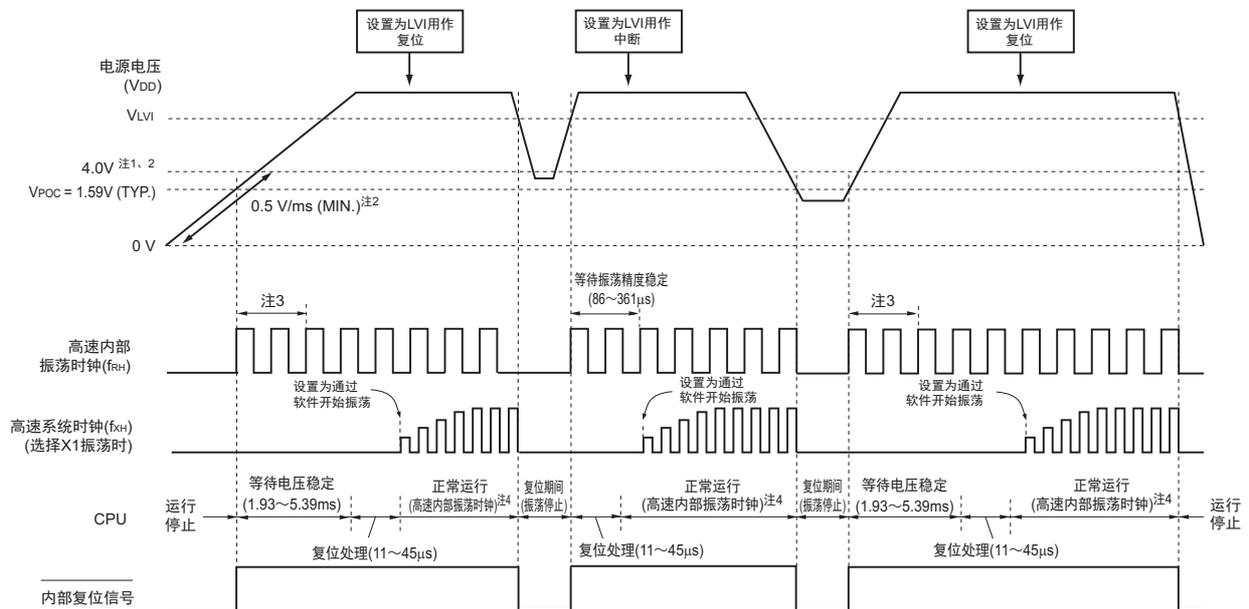
(2) 设置为 2.7V/1.59V POC 模式时 (选项字节: POCMODE=1)

- 在接通电源时产生内部复位信号, 并且在电源电压 (V_{DD}) 大于输入电源电压时的检测电压 ($V_{DDPOC}=2.7V\pm 0.2V$) 时, 解除复位。
- 比较电源电压 (V_{DD}) 和检测电压 ($V_{POC}=1.59V\pm 0.15V$), 在 $V_{DD} < V_{POC}$ 时, 产生内部复位信号; 在 $V_{DD} \geq V_{DDPOC}$ 时, 解除复位。

上电清除电路和低电压检测电路的内部复位信号产生时序如下所示。

图 19-2 上电清除电路和低电压检测电路的内部复位信号产生时序 (1/2)

(1) 设置为 1.59V POC 模式的情况 (选项字节: POCMODE=0)



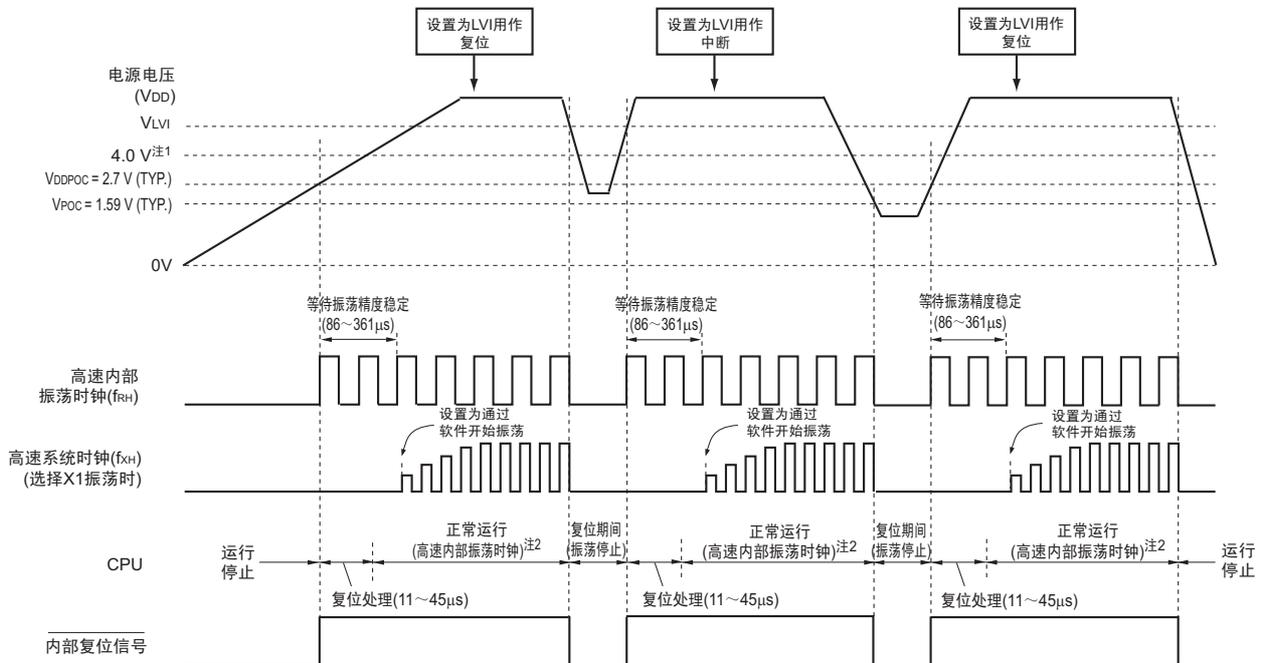
- 注 1. 运行保证范围为 $4.0V \leq V_{DD} \leq 5.5V$ 。如果要在电源下降时将非运行保证范围的电压范围设置为复位状态，使用低电压检测电路的复位功能，或者将低电平输入到 \overline{RESET} 引脚。
2. 如果从接通电源到上升到 4.0V 的电压上升斜率小于 0.5V/ms(MIN.)，就在此期间输入低电平到 \overline{RESET} 引脚，或者通过选项字节设置为 2.7V/1.59V POC 模式 (POCMODE=1)。
3. 高速内部振荡时钟的振荡精度稳定等待时间包括在内部的电压稳定等待时间内。
4. 可将 CPU 时钟从高速内部振荡时钟切换到高速系统时钟。如果使用 X1 时钟，必须在通过 OSTC 寄存器确认振荡稳定时间后再切换。

注意 必须在复位解除后通过软件设置低电压检测电路 (参照“第 20 章 低电压检测电路”)。

备注 V_{LVI}: LVI 检测电压
V_{POC}: POC 检测电压

图 19-2 上电清除电路和低电压检测电路的内部复位信号产生时序 (2/2)

(2) 设置为 2.7V/1.59V POC 模式的情况 (选项字节: POCMODE=1)



- 注 1. 运行保证范围为 $4.0V \leq V_{DD} \leq 5.5V$ 。如果要在电源下降时将非运行保证范围的电压范围设置为复位状态, 使用低电压检测电路的复位功能, 或者将低电平输入到 \overline{RESET} 引脚。
2. 可将 CPU 时钟从高速内部振荡时钟切换到高速系统时钟。如果使用 X1 时钟, 必须在通过 OSTC 寄存器确认振荡稳定时间后再切换。

注意 1. 必须在复位解除后通过软件进行低电压检测电路的设置 (参照“第 20 章 低电压检测电路”)。

2. 在电源电压上升到 1.59V(TYP.) 后, 需要 1.93 ~ 5.39ms 的电压稳定等待时间。如果从 1.59V(TYP.) 上升到 2.7V(TYP.) 的时间在 1.93ms 以内, 就在复位处理前自动产生 0 ~ 5.39ms 的电源稳定等待时间。

备注 V_{LVI} : LVI 检测电压
 V_{POC} : POC 检测电压

19.4 上电清除电路的注意事项

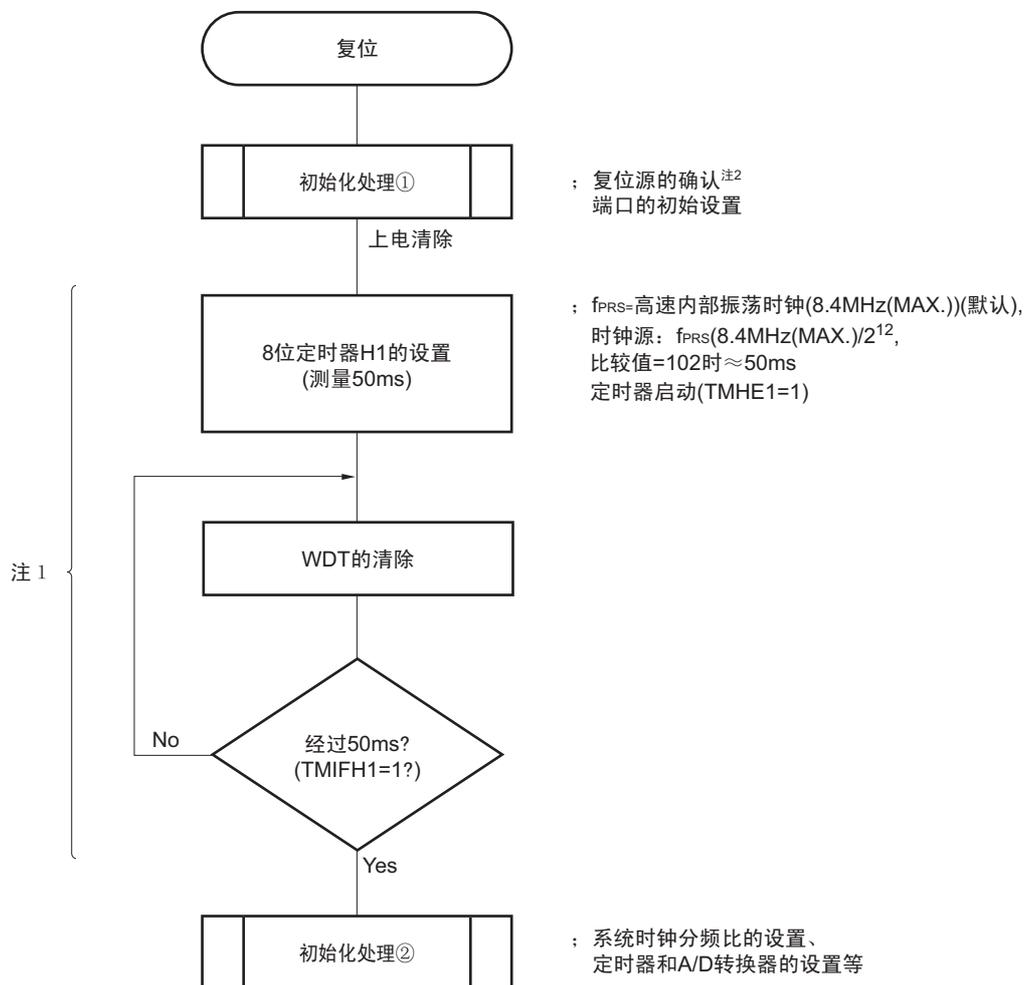
在因电源电压 (V_{DD}) 接近 POC 检测电压 (V_{POC}) 导致某期间出现波动的系统中, 可能会重复出现复位状态 / 复位解除状态。可通过以下措施任意设置从复位解除到单片机运行的时间。

< 措施 >

复位解除后, 通过使用定时器等软件计数器, 等待因系统而不同的电源电压波动期间后, 对端口等进行初始设置。

图 19-3 复位解除后的软件处理示例 (1/2)

- 接近 POC 检测电压的电源电压波动为 50ms 以下时

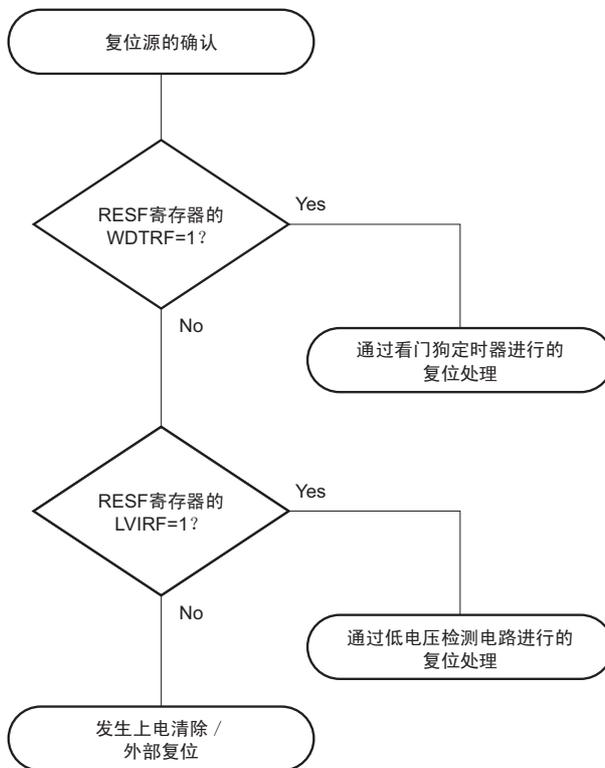


备注 1. 如果在此期间再次发生复位, 就不转移到初始化处理②。

2. 流程图如下一页所示。

图 19-3 复位解除后的软件处理示例 (2/2)

- 复位源的确认



第 20 章 低电压检测电路

20.1 低电压检测电路的功能

低电压检测电路具有以下功能。

- 比较电源电压 (V_{DD}) 和检测电压 (V_{LVI})，或者比较外部输入引脚的输入电压 ($EXLVI$) 和检测电压 ($V_{EXLVI}=1.21V(TYP.)$: 固定)，产生内部复位或者内部中断信号。
- 可通过软件选择电源电压 (V_{DD}) / 外部输入引脚的输入电压 ($EXLVI$)。
- 可通过软件选择检测后是发生复位还是发生中断。
- 可通过软件从 2 种检测电平中选择电源电压的检测电压 (V_{LVI})。
- 可在 STOP 模式中运行。

可通过软件选择产生以下的复位和中断信号。

选择电源电压 (V_{DD}) 的电平检测 (LVISEL=0)		选择外部输入引脚的输入电压 ($EXLVI$) 的电平检测 (LVISEL=1)	
选择复位 (LVIMD=1)	选择中断 (LVIMD=0)	选择复位 (LVIMD=1)	选择中断 (LVIMD=0)
在 $V_{DD} < V_{LVI}$ 时，发生内部复位；在 $V_{DD} \geq V_{LVI}$ 时，解除内部复位。	在电源电压下降至 $V_{DD} < V_{LVI}$ 或者电源电压上升至 $V_{DD} \geq V_{LVI}$ 时，产生内部中断信号。	在 $EXLVI < V_{EXLVI}$ 时，发生内部复位；在 $EXLVI \geq V_{EXLVI}$ 时，解除内部复位。	在输入电压下降至 $EXLVI < V_{EXLVI}$ 或者输入电压上升至 $EXLVI \geq V_{EXLVI}$ 时，产生内部中断信号。

备注 LVISEL: 低电压检测寄存器 (LVIM) 的 bit2

LVIMD: LVIM 的 bit1

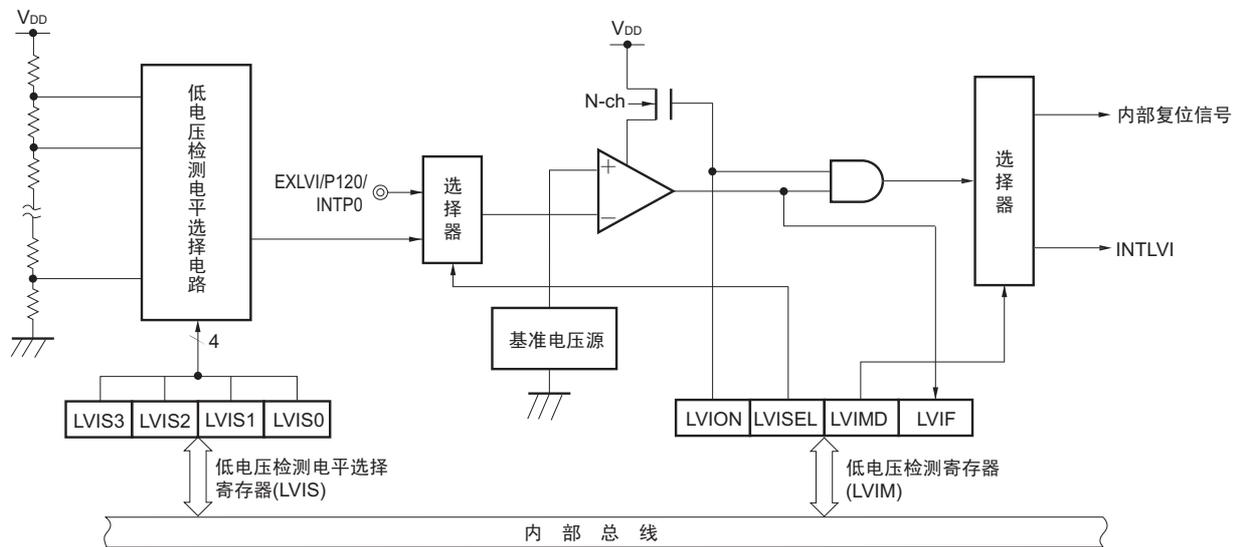
在低电压检测电路运行期间，可通过读低电压检测标志 (LVIF: LVIM 的 bit0)，确认电源电压或者外部输入引脚的输入电压是大于等于检测电平还是小于检测电平。

在将低电压检测电路用作复位时，一旦发生复位，复位控制标志寄存器 (RESF) 的 bit0 (LVIRF) 就被置“1”。有关 RESF 的详细内容，请参照“第 18 章 复位功能”。

20.2 低电压检测电路的结构

低电压检测电路的框图如图 20-1 所示。

图 20-1 低电压检测电路的框图



20.3 低电压检测电路的控制寄存器

低电压检测电路由以下寄存器控制。

- 低电压检测寄存器 (LVIM)
- 低电压检测电平选择寄存器 (LVIS)
- 端口模式寄存器 12 (PM12)

(1) 低电压检测寄存器 (LVIM)

低电压检测寄存器设置低电压检测和运行模式。

通过 1 位或 8 位存储器操作指令设置 LVIM。

通过产生 LVI 复位以外的复位信号时，LVIM 变为“00H”。

图 20-2 低电压检测寄存器 (LVIM) 的格式

地址: FFBEH 复位后: 00H^{注1} R/W^{注2}

符号	<7>	6	5	4	3	<2>	<1>	<0>
LVIM	LVION	0	0	0	0	LVISEL	LVIMD	LVIF

LVION 注3、注4	允许低电压检测运行
0	禁止运行
1	允许运行

LVISEL ^{注3}	选择电压检测
0	检测电源电压 (V_{DD}) 的电平
1	检测外部输入引脚的输入电压 ($EXLVI$) 的电平

LVIMD ^{注3}	选择低电压检测的运行模式 (中断 / 复位)
0	<ul style="list-style-type: none"> 当 LVISEL=0 时, 在电压下降至电源电压 (V_{DD}) < 检测电压 (V_{LVI}) 或者电压上升至 $V_{DD} \geq V_{LVI}$ 时, 产生内部中断信号。 当 LVISEL=1 时, 在电压下降至外部输入引脚的输入电压 ($EXLVI$) < 检测电压 (V_{EXLVI}) 或者电压上升 $EXLVI \geq V_{EXLVI}$ 时, 产生内部中断信号。
1	<ul style="list-style-type: none"> 当 LVISEL=0 时, 在电源电压 (V_{DD}) < 检测电压 (V_{LVI}) 时发生内部复位, 在 $V_{DD} \geq V_{LVI}$ 时解除内部复位。 当 LVISEL=1 时, 在外部输入引脚的输入电压 ($EXLVI$) < 检测电压 (V_{EXLVI}) 时发生内部复位, 在 $EXLVI \geq V_{EXLVI}$ 时解除内部复位。

LVIF	低电压检测标志
0	<ul style="list-style-type: none"> 如果 LVISEL=0, 电源电压 (V_{DD}) \geq 检测电压 (V_{LVI}) 或者禁止运行时 如果 LVISEL=1, 外部输入引脚的输入电压 ($EXLVI$) \geq 检测电压 (V_{EXLVI}) 或者禁止运行时
1	<ul style="list-style-type: none"> 如果 LVISEL=0, 电源电压 (V_{DD}) < 检测电压 (V_{LVI}) 如果 LVISEL=1, 外部输入引脚的输入电压 ($EXLVI$) < 检测电压 (V_{EXLVI})

- 注 1. 在 LVI 复位以外的复位时, 该寄存器清“00H”。
2. bit0 为只读位。
3. LVION、LVIMD、LVISEL 在 LVI 复位以外的复位时清“0”, 在 LVI 复位时, 不清“0”。
4. 如果将 LVION 置“1”, LVI 电路内的比较器就开始运行。通过软件等待从 LVION 置“1”到运行稳定的时间 (10 μ s(MIN.))。运行稳定后, 从小于 LVI 检测电压的状态到 LVIF 置“1”前, 需要至少 200 μ s 的外部输入 (最小脉宽: 200 μ s(MIN.))。

注意 1. 可通过执行以下任一步骤, 停止 LVI。

- 8 位存储器操作指令时: 将“00H”写入 LVIM
 - 1 位存储器操作指令时: 将 LVION 清“0”
2. 外部输入引脚的输入电压 ($EXLVI$) 必须满足 $EXLVI < V_{DD}$ 。
3. 将 LVI 用作中断时, 如果在小于 LVI 检测电压的状态下将 LVION 清“0”, 就产生 INTLVI 信号, 并且 LVIF 变为“1”。

(2) 低电压检测电平选择寄存器 (LVIS)

低电压检测电平选择寄存器选择低电压检测电平。

通过 1 位或 8 位存储器操作指令设置 LVIS。

通过产生 LVI 复位以外的复位信号，LVIS 变为“00H”。

图 20-3 低电压检测电平选择寄存器 (LVIS) 的格式

地址: FFBFH 复位后: 00H^注 R/W

符号	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	0	0	LVIS0

LVIS0	检测电平
0	$V_{LV10} (4.24 \pm 0.1V)$
1	$V_{LV11} (4.09 \pm 0.1V)$

注 在发生 LVI 复位时，LVIS 寄存器的值不被复位，保持当前的值。在其他复位时，LVIS 寄存器的值被清“00H”。

注意 1. 必须将 bit1 ~ bit7 置“0”。

2. 在 LVI 运行期间，不能更改 LVIS 的值。

3. 如果检测到外部输入引脚的输入电压 (EXLVI)，检测电压就固定 ($V_{EXLVI}=1.21V(TYP.)$)。因为不需要设置 LVIS。

(3) 端口模式寄存器 12 (PM12)

将 P120/EXLVI/INTP0 引脚用作外部低电压检测的电位输入时，必须将 PM120 置“1”。此时，P120 的输出锁存器既可是“0”也可是“1”。

通过 1 位或 8 位存储器操作指令设置 PM12。

在产生复位信号后，PM12 变为“FFH”。

图 20-4 端口模式寄存器 12 (PM12) 的格式

地址: FF2CH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM12	1	1	1	1	1	PM122	PM121	PM120

PM12n	选择 P12n 引脚的输入 / 输出模式 (n=0 ~ 2)
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

20.4 低电压检测电路的运行

低电压检测电路有以下两种运行模式。

(1) 用作复位 (LVIMD=1)

- 在LVISEL=0时, 比较电源电压 (V_{DD}) 和检测电压 (V_{LVI}), 在 $V_{DD} < V_{LVI}$ 时发生内部复位; 在 $V_{DD} \geq V_{LVI}$ 时解除内部复位。
- 在LVISEL=1时, 比较外部输入引脚的输入电压 (EXLVI) 和检测电压 ($V_{EXLVI}=1.21V(TYP.)$), 在 $EXLVI < V_{EXLVI}$ 时发生内部复位; 在 $EXLVI \geq V_{EXLVI}$ 时解除内部复位。

(2) 用作中断 (LVIMD=0)

- 在LVISEL=0时, 比较电源电压 (V_{DD}) 和检测电压 (V_{LVI}), 在电压下降至 $V_{DD} < V_{LVI}$ 或者电压上升至 $V_{DD} \geq V_{LVI}$ 时, 产生中断信号 (INTLVI)。
- 在LVISEL=1时, 比较外部输入引脚的输入电压 (EXLVI) 和检测电压 ($V_{EXLVI}=1.21V(TYP.)$), 在电压下降至 $EXLVI < V_{EXLVI}$ 或者电压上升至 $EXLVI \geq V_{EXLVI}$ 时, 产生中断信号 (INTLVI)。

在低电压检测电路运行期间, 可通过读低电压检测标志 (LVIF: LVIM 的 bit0), 确认电源电压是大于等于检测电平还是小于检测电平。

备注 LVIMD: 低电压检测寄存器 (LVIM) 的 bit1。

LVISEL: LVIM 的 bit2

20.4.1 用作复位时的设置

(1) 检测电源电压 (V_{DD}) 的电平时

- 开始运行时
 - ① 屏蔽LVI的中断 (LVIMK=1)。
 - ② 将低电压检测寄存器 (LVIM) 的 bit2 (LVISEL) 置“0” (检测电源电压 (V_{DD}) 的电平) (默认值)。
 - ③ 通过低电压检测电平选择寄存器 (LVIS) 的 bit0 (LVIS0) 设置检测电压。
 - ④ 将LVIM的 bit7 (LVION) 置“1” (允许LVI运行)。
 - ⑤ 通过软件等待运行稳定时间 (10 μ s(MIN.))。
 - ⑥ 保持等待直到通过LVIM的 bit0 (LVIF) 确认“电源电压 (V_{DD}) \geq 检测电压 (V_{LVI})”。
 - ⑦ 将LVIM的 bit1 (LVIMD) 置“1” (在检测出电平时发生复位)。

与步骤①~⑦对应的低电压检测电路的内部复位信号产生时序如图 20-5 所示。

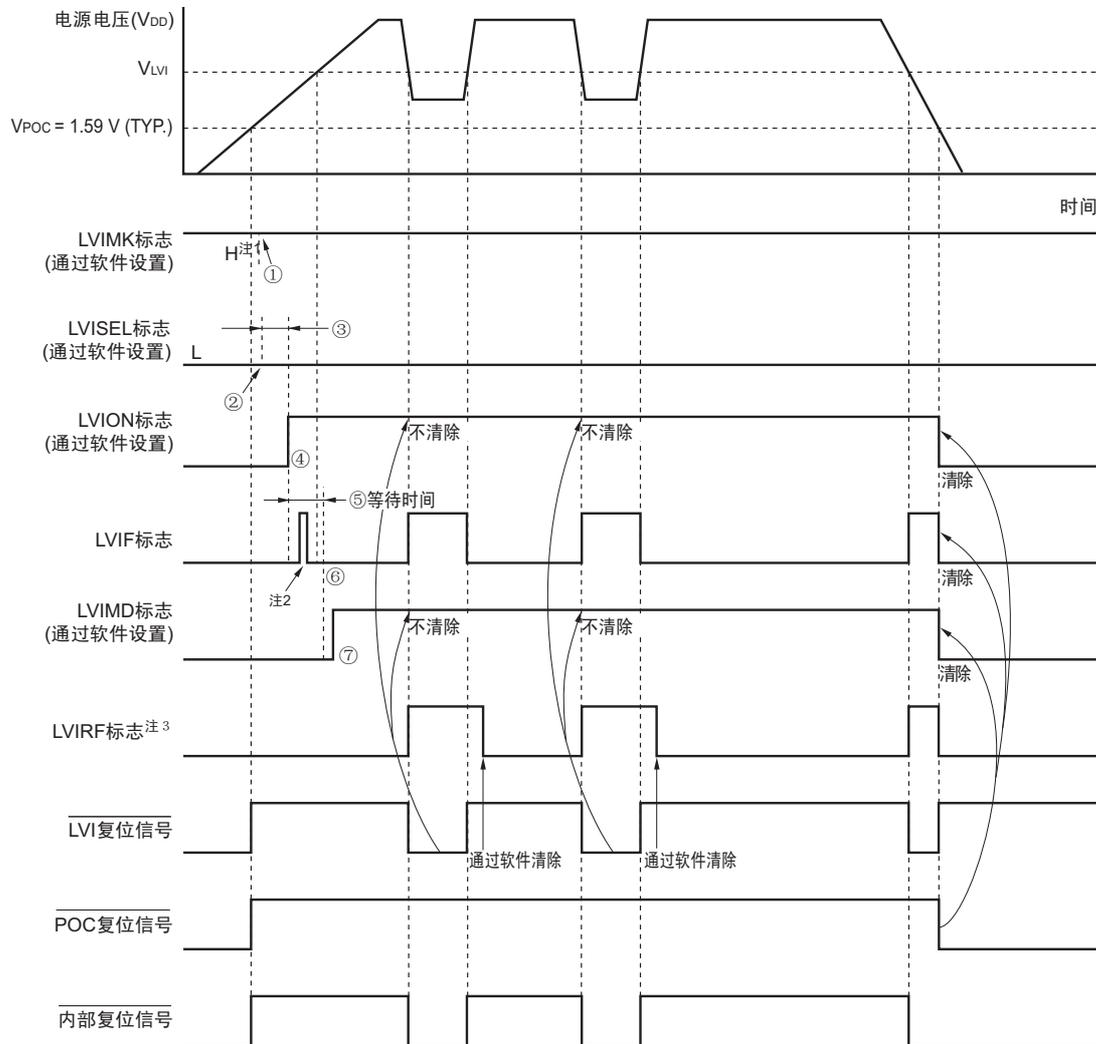
注意 1. 必须执行①。如果 LVIMK 为“0”, 有可能在执行④的处理后发生中断。

2. 在 LVIMD 为“1”时, 如果电源电压 (V_{DD}) \geq 检测电压 (V_{LVI}), 就不产生内部复位信号。

- 停止运行时
必须执行以下任一步骤。
 - 8位存储器操作指令时:
将“00H”写入LVIM。
 - 1位存储器操作指令时:
将LVIMD清“0”后, 将LVION清“0”。

图 20-5 低电压检测电路的内部复位信号的产生时序（检测电源电压（ V_{DD} ）的电平）（1/2）

(1) 设置为 1.59V POC 模式的情况（选项字节：POCMODE=0）

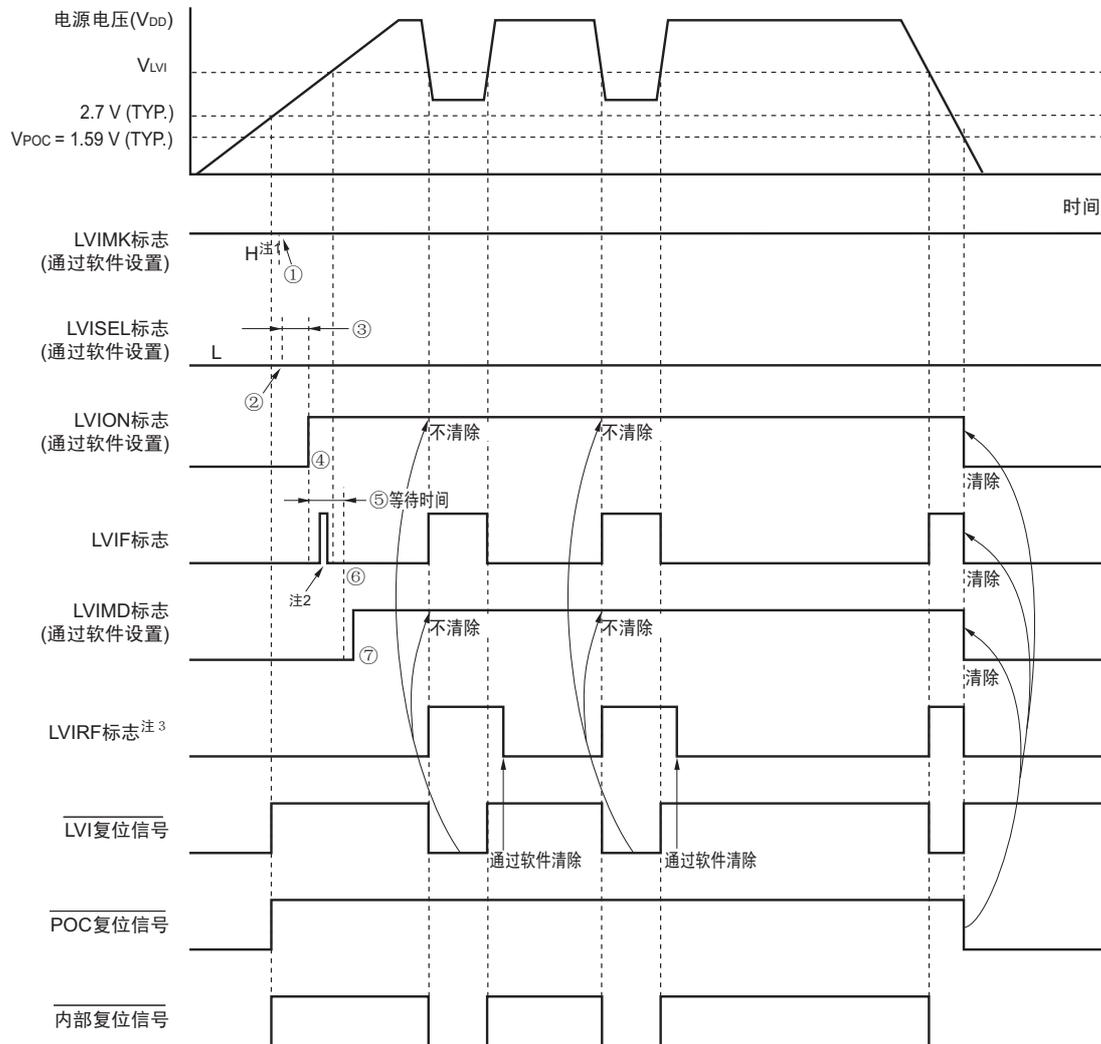


- 注 1. 在产生复位信号后，LVIMK 标志变为“1”。
2. LVIF 标志可能会置“1”。
3. LVIRF 是复位控制标志寄存器（RESF）的 bit0。有关 RESF 的详细内容，请参照“第 18 章 复位功能”。

备注 图 20-5 中的①~⑦，与“20.4.1 (1) 检测电源电压（ V_{DD} ）的电平时”中“开始运行时”的步骤①~⑦对应。

图 20-5 低电压检测电路的内部复位信号的产生时序（检测电源电压（ V_{DD} ）的电平）（2/2）

(2) 设置为 2.7V/1.59V POC 模式的情况（选项字节：POCMODE=1）



- 注 1. 在产生复位信号后，LVIMK 标志变为“1”。
2. LVIF 标志可能会置“1”。
3. LVIRF 是复位控制标志寄存器（RESF）的 bit0。有关 RESF 的详细内容，请参照“第 18 章 复位功能”。

备注 图 20-5 中的①~⑦，与“20.4.1 (1) 检测电源电压（ V_{DD} ）的电平时”中“开始运行时”的步骤①~⑦对应。

(2) 检测外部输入引脚的输入电压 (EXLVI) 的电平时

- 开始运行时
 - ① 屏蔽LVI的中断 (LVIMK=1)。
 - ② 将低电压检测寄存器 (LVIM) 的bit2 (LVISEL) 置“1” (检测外部输入引脚的输入电压 (EXLVI) 的电平)。
 - ③ 将LVIM的bit7 (LVION) 置“1” (允许LVI运行)。
 - ④ 通过软件等待运行稳定时间 (10 μ s(MIN.))。
 - ⑤ 保持等待直到通过LVIM的bit0 (LVIF) 确认“外部输入引脚的输入电压 (EXLVI) \geq 检测电压 ($V_{EXLVI}=1.21V(TYP.)$) ”。
 - ⑥ 将LVIM的bit1 (LVIMD) 置“1” (在检测出电平时发生复位)。

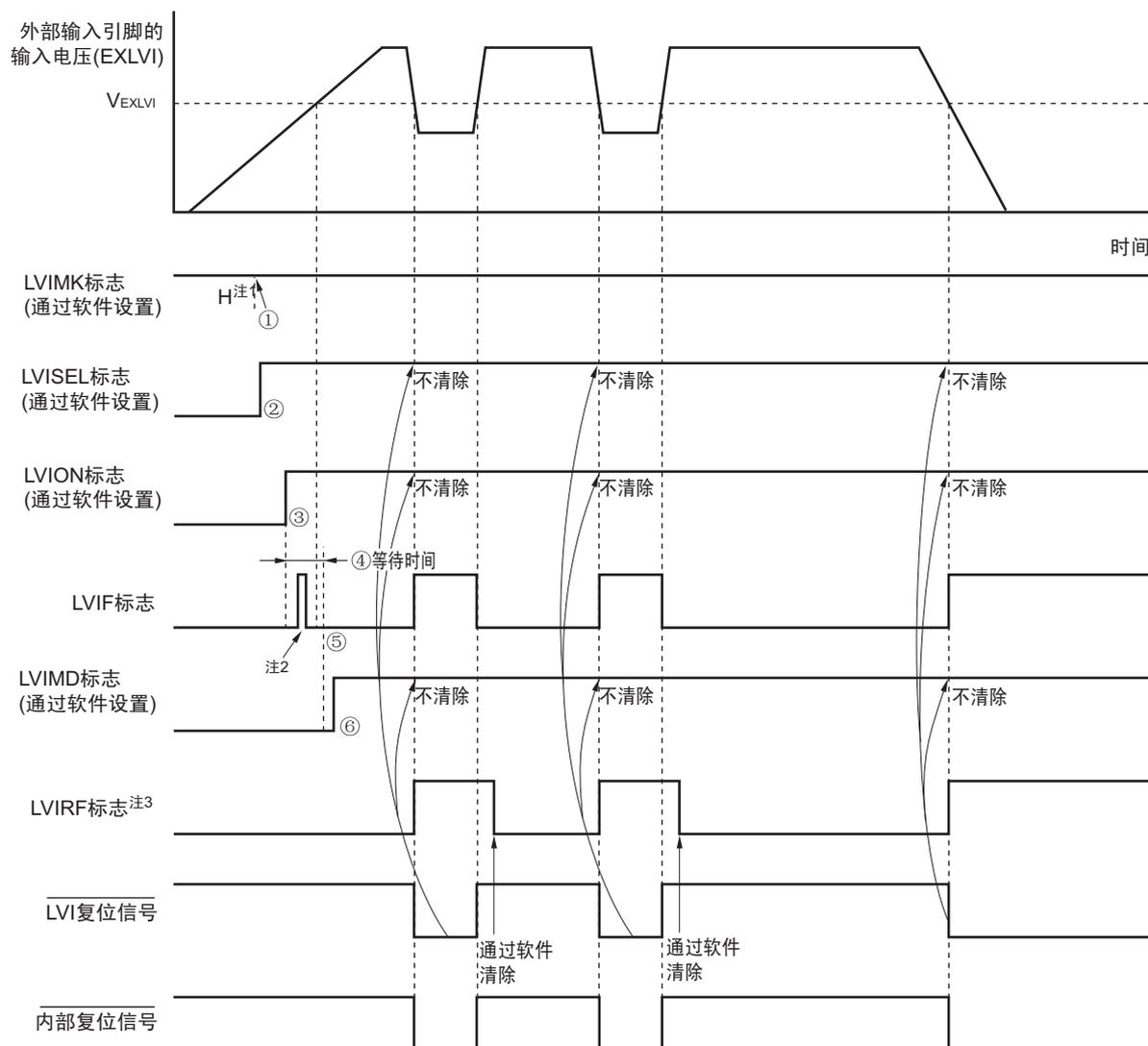
与步骤①~⑥对应的低电压检测电路的内部复位信号产生时序如图 20-6 所示。

注意 1. 必须执行①。如果 LVIMK 为“0”，有可能在执行③的处理后发生中断。

2. 在 LVIMD 为“1”时，如果外部输入引脚的输入电压 (EXLVI) \geq 检测电压 ($V_{EXLVI}=1.21V(TYP.)$)，就不产生内部复位信号。
3. 外部输入引脚的输入电压必须满足 $EXLVI < V_{DD}$ 。

- 停止运行时
必须执行以下任一步骤。
 - 8位存储器操作指令时：
将“00H”写入LVIM。
 - 1位存储器操作指令时：
将LVIMD清“0”后，将LVION清“0”。

图 20-6 低电压检测电路的内部复位信号的产生时序
(检测外部输入引脚的输入电压 (EXLVI) 的电平)



- 注 1. 在产生复位信号后，LVIMK 标志变为“1”。
 2. LVIF 标志可能会置“1”。
 3. LVIRF 是复位控制标志寄存器 (RESF) 的 bit0。有关 RESF 的详细内容，请参照“第 18 章 复位功能”。

备注 图 20-6 中的①~⑥，与“20.4.1 (2) 检测外部输入引脚的输入电压 (EXLVI) 的电平时”中“开始运行时”的步骤①~⑥对应。

20.4.2 用作中断时的设置

(1) 检测电源电压 (V_{DD}) 的电平时

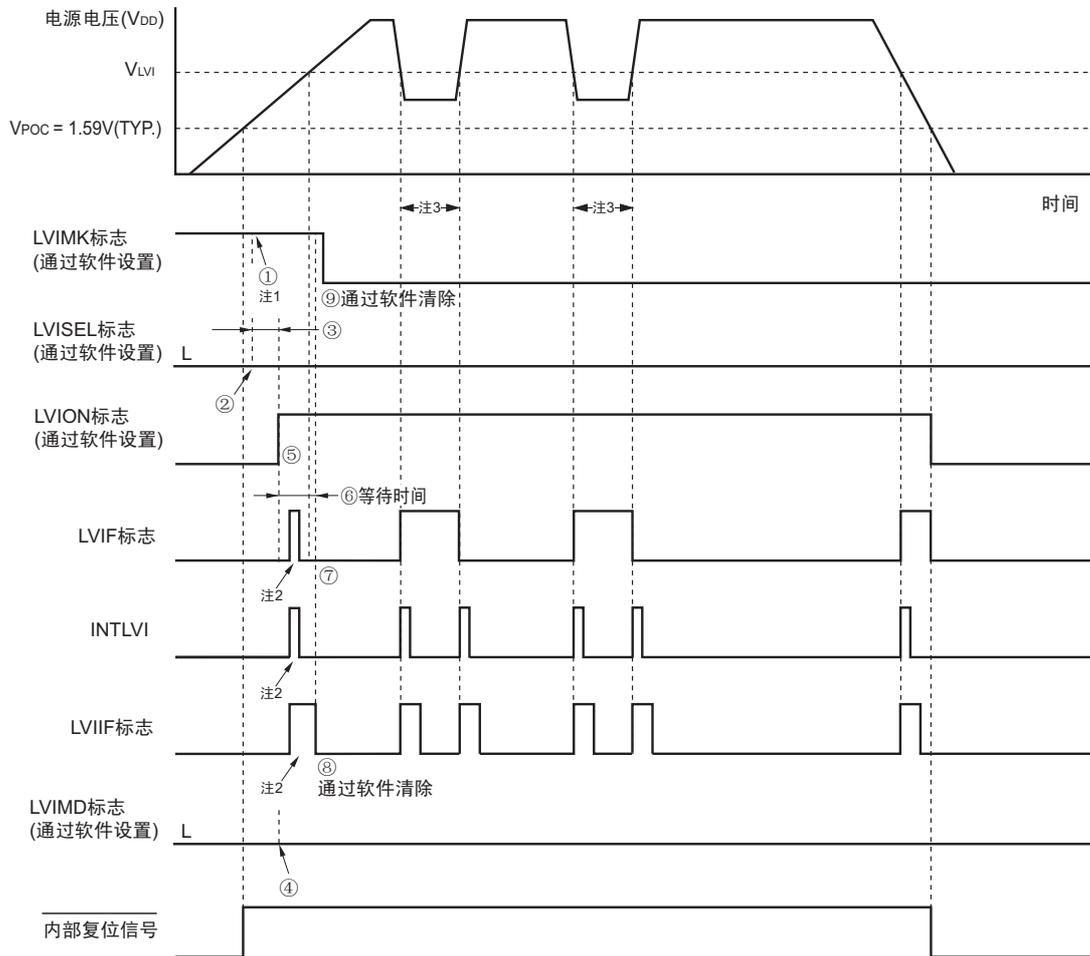
- 开始运行时
 - ① 屏蔽LVI的中断 ($LVIMK=1$)。
 - ② 将低电压检测寄存器 (LVIM) 的bit2 ($LVISEL$) 置“0” (检测电源电压 (V_{DD}) 的电平) (默认值)。
 - ③ 通过低电压检测电平选择寄存器 (LVIS) 的bit0 ($LVIS0$) 设置检测电压。
 - ④ 将LVIM的bit1 ($LVIMD$) 置“0” (在电平检测时产生中断信号) (默认值)。
 - ⑤ 将LVIM的bit7 ($LVION$) 置“1” (允许LVI运行)。
 - ⑥ 通过软件等待运行稳定时间 ($10\mu s(MIN.)$)。
 - ⑦ 通过LVIM的bit0 ($LVIF$) 确认在检测到下降沿时“电源电压 (V_{DD}) \geq 检测电压 (V_{LVI}) ”, 在检测到上升沿时“电源电压 (V_{DD}) $<$ 检测电压 (V_{LVI}) ”。
 - ⑧ 将LVI的中断请求标志 ($LVIF$) 清“0”。
 - ⑨ 解除LVI的中断屏蔽标志 ($LVIMK$)。
 - ⑩ 执行EI指令 (使用向量中断时)。

与步骤①~⑨对应的低电压检测电路的中断信号产生时序如图 20-7 所示。

- 停止运行时
必须执行以下任一步骤。
 - 8位存储器操作指令时:
将“00H”写入LVIM。
 - 1位存储器操作指令时:
将LVION清“0”。

图 20-7 低电压检测电路的中断信号的产生时序（检测电源电压（ V_{DD} ）的电平）（1/2）

(1) 设置为 1.59V POC 模式的情况（选项字节：POCMODE=0）

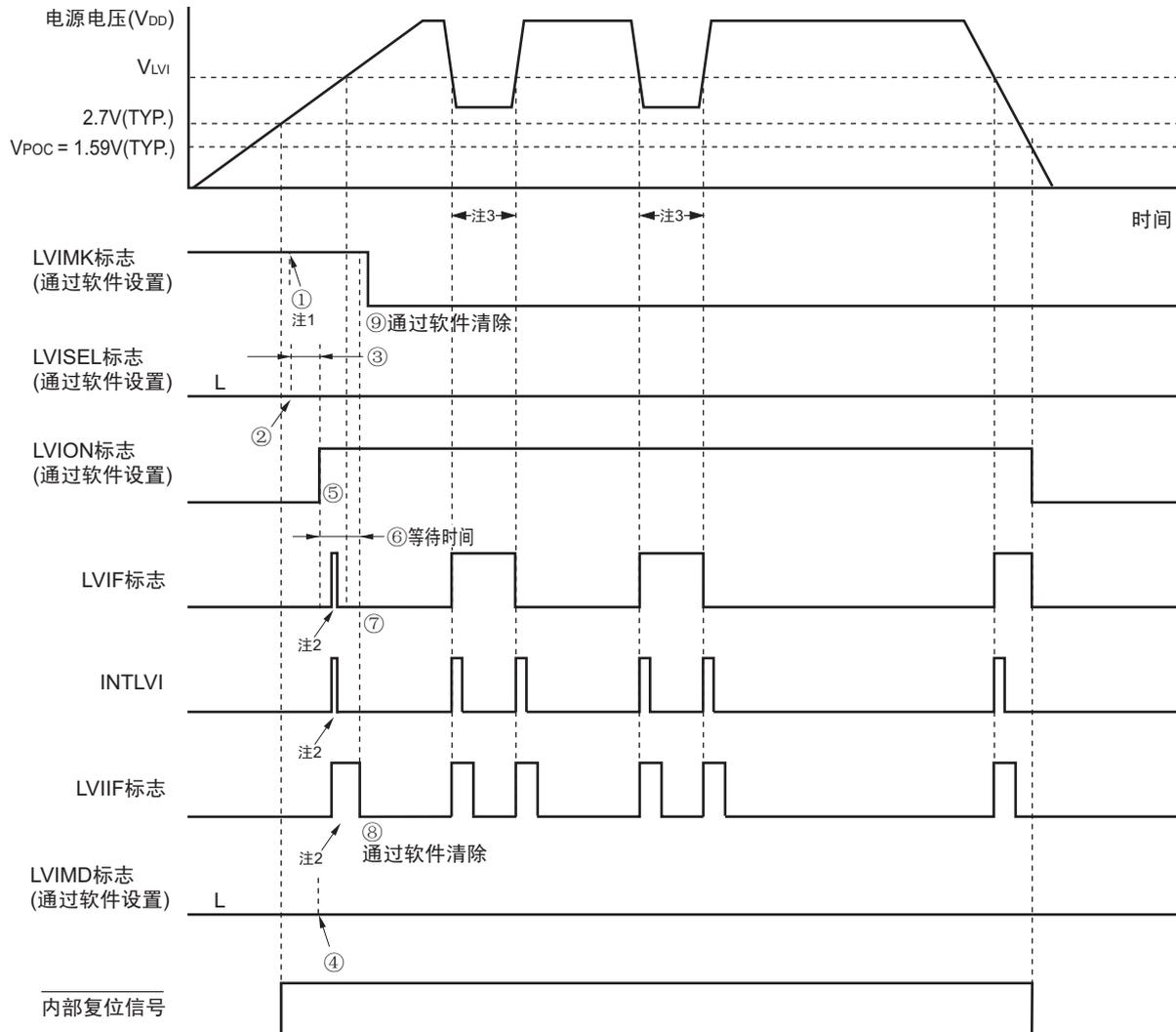


- 注 1. 在产生复位信号后，LVIMK 标志变为“1”。
2. 可能产生中断请求信号（INTLVI），并且 LVIIF 标志和 LVIF 标志可能置“1”。
3. 如果在小于 LVI 检测电压（ V_{LVI} ）的状态下将 LVION 清“0”，就产生 INTLVI 信号，并且 LVIIF 变为“1”。

备注 图 20-7 中的①~⑨，与“20.4.2 (1) 检测电源电压（ V_{DD} ）的电平时”中“开始运行时”的步骤①~⑨对应。

图 20-7 低电压检测电路的中断信号的产生时序（检测电源电压（ V_{DD} ）的电平）（2/2）

(2) 设置为 2.7V/1.59V POC 模式的情况（选项字节：POCMODE=1）



- 注 1. 在产生复位信号后，LVIMK 标志变为“1”。
2. 可能产生中断请求信号（INTLVI），并且 LVIIF 标志和 LVIF 标志可能置“1”。
3. 如果在小于 LVI 检测电压（ V_{LVI} ）的状态下将 LVION 清“0”，就产生 INTLVI 信号，并且 LVIIF 变为“1”。

备注 图 20-7 中的①~⑨，与“20.4.2 (1) 检测电源电压（ V_{DD} ）的电平时”中“开始运行时”的步骤①~⑨对应。

(2) 检测外部输入引脚的输入电压 (EXLVI) 的电平时

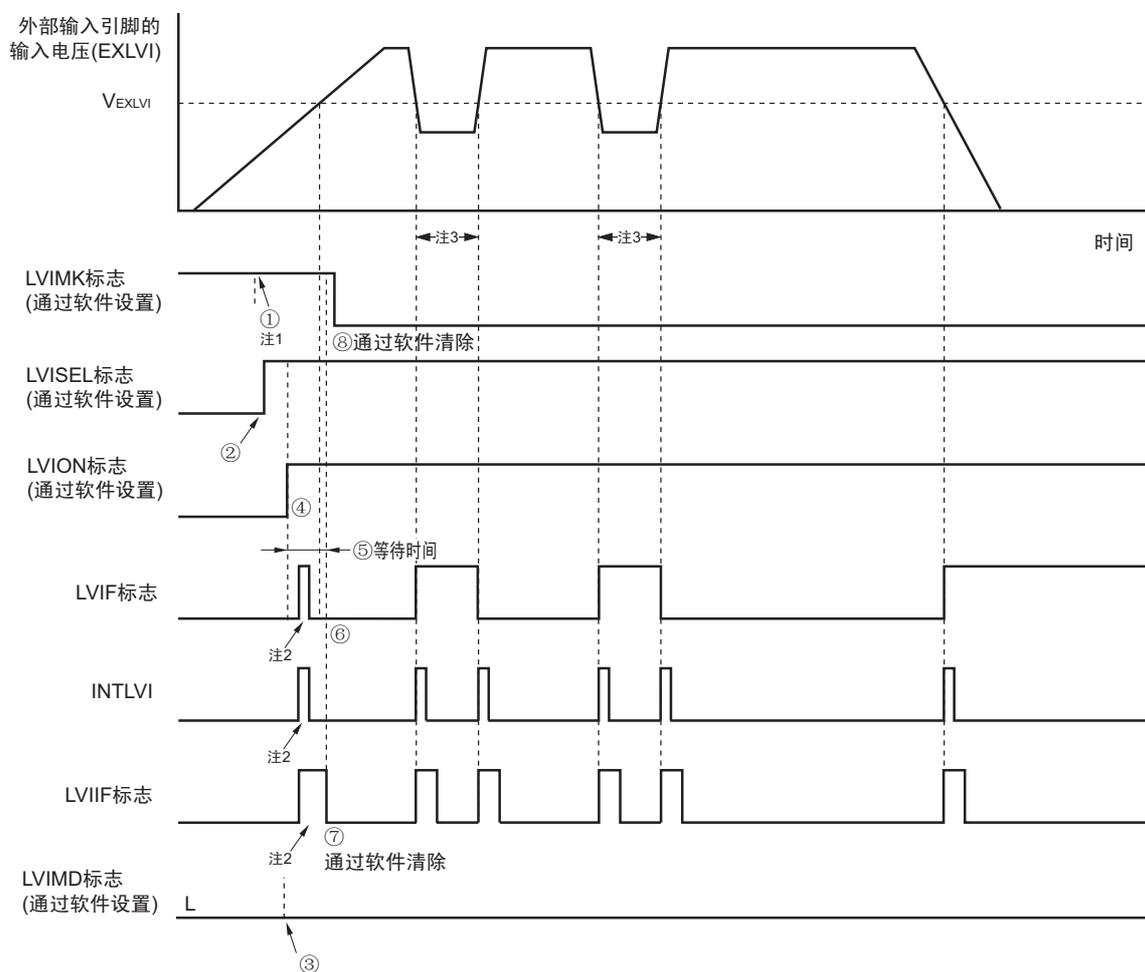
- 开始运行时
 - ① 屏蔽LVI的中断 (LVIMK=1)。
 - ② 将低电压检测寄存器 (LVIM) 的bit2 (LVISEL) 置“1” (检测外部输入引脚的输入电压 (EXLVI) 的电平)。
 - ③ 将LVIM的bit1 (LVIMD) 置“0” (在电平检测时产生中断信号) (默认值)。
 - ④ 将LVIM的bit7 (LVION) 置“1” (允许LVI运行)。
 - ⑤ 通过软件等待运行稳定时间 (10 μ s(MIN.))。
 - ⑥ 通过LVIM的bit0 (LVIF) 确认在检测到下降沿时“外部输入引脚的输入电压 (EXLVI) \geq 检测电压 ($V_{EXLVI}=1.21V(TYP.)$) ”, 在检测到上升沿时“外部输入引脚的输入电压 (EXLVI) $<$ 检测电压 ($V_{EXLVI}=1.21V(TYP.)$) ”。
 - ⑦ 将LVI的中断请求标志 (LVIF) 清“0”。
 - ⑧ 解除LVI的中断屏蔽标志 (LVIMK)。
 - ⑨ 执行EI指令 (使用向量中断时)。

与步骤①~⑧对应的低电压检测电路的中断信号产生时序如图 20-8 所示。

注意 外部输入引脚的输入电压 (EXLVI) 必须满足 $EXLVI < V_{DD}$ 。

- 停止运行时
必须执行以下任一步骤。
 - 8位存储器操作指令时:
将“00H”写入LVIM。
 - 1位存储器操作指令时:
将LVION清“0”。

图 20-8 低电压检测电路的中断信号的产生时序
(检测外部输入引脚的输入电压 (EXLVI) 的电平)



- 注 1. 在产生复位信号后，LVIMK 标志变为“1”。
- 注 2. 产生中断请求信号 (INTLVI)，并且 LVIIF 标志和 LVIF 标志可能置“1”。
- 注 3. 如果在小于 LVI 检测电压 (V_{LVI}) 的状态下将 LVION 清“0”，就产生 INTLVI 信号，并且 LVIIF 变为“1”。

备注 图 20-8 中的①~⑧，与“20.4.2 (2) 检测外部输入引脚的输入电压 (EXLVI) 的电平时”中“开始运行时”的步骤①~⑧对应。

20.5 低电压检测电路的注意事项

某期间的电源电压 (V_{DD}) 在 LVD 检测电压附近波动的系统中, 根据低电压检测电路的使用方法可进行如下运行。

(1) 用作复位的情况

有可能反复出现复位状态 / 复位解除状态。

可通过下述的措施 (1) 所示的处理, 任意设置从复位解除到开始单片机运行的时间。

(2) 用作中断的情况

有可能频繁产生中断请求。请执行下述措施 (2) 中 (b) 所示的处理。

< 措施 >

(1) 用作复位的情况

复位解除后, 通过使用定时器等软件计数器, 等待因系统而不同的电源电压波动期间后, 对端口进行初始设置 (参照图 20-9)。

(2) 用作中断的情况

(a) 在 LVI 中断处理程序中, 通过低电压检测寄存器 (LVIM) 的 bit0 (LVIF), 确认在检测到下降沿时“电源电压 (V_{DD}) \geq 检测电压 (V_{LVI})”, 在检测到上升沿时“电源电压 (V_{DD}) $<$ 检测电压 (V_{LVI})”, 并且将中断请求标志寄存器 0L (IF0L) 的 bit0 (LVIIF) 清“0”。

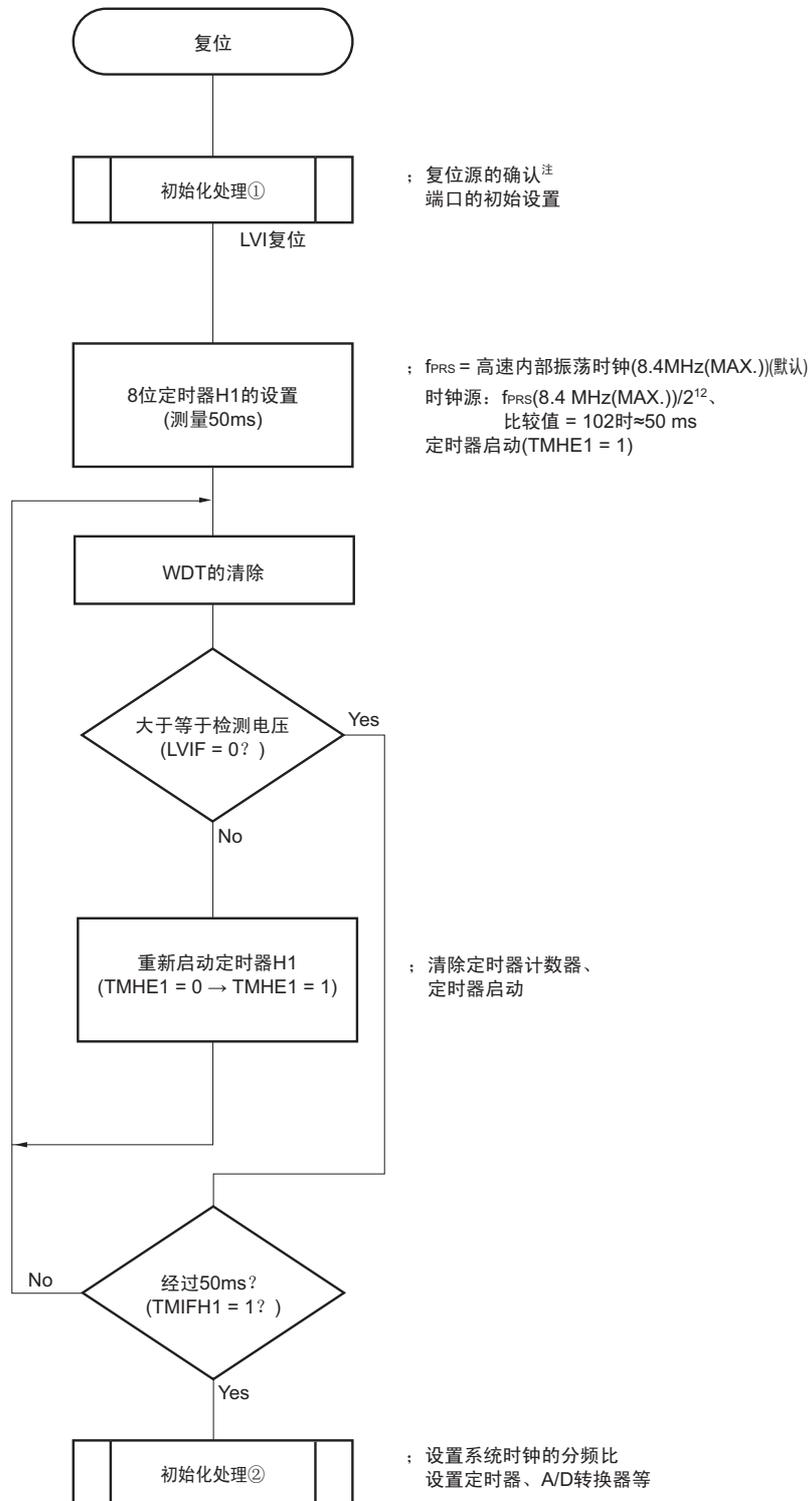
(b) 在接近 LVI 检测电压的电源电压波动期间长的系统中, 必须在等待电源电压变化期间后, 通过 LVIF 标志, 确认在检测到下降沿时“电源电压 (V_{DD}) \geq 检测电压 (V_{LVI})”, 在检测到上升沿时“电源电压 (V_{DD}) $<$ 检测电压 (V_{LVI})”, 并且将 LVIIF 标志清“0”。

备注 在将低电压检测寄存器 (LVIM) 的 bit2 (LVISEL) 置“1”时, 上述措施中的“电源电压”和“检测电压”替换为以下电压。

- 电源电压 (V_{DD}) \rightarrow 外部输入引脚的输入电压 ($EXLVI$)
- 检测电压 (V_{LVI}) \rightarrow 检测电压 ($V_{EXLVI}=1.21V$)

图 20-9 复位解除后的软件处理示例 (1/2)

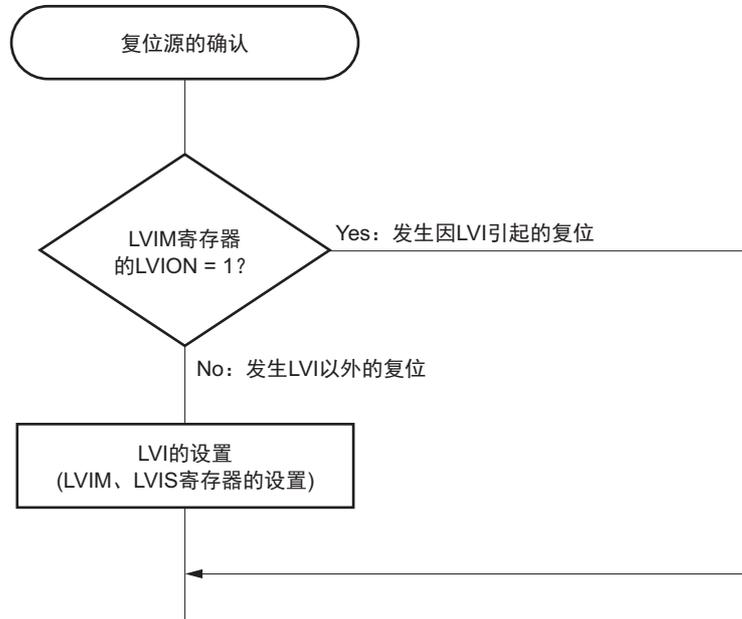
- 接近LVI检测电压的电源电压波动为50ms以下时



注 流程图如下一页所示。

图 20-9 复位解除后的软件处理示例 (2/2)

- 复位源的确认



第 21 章 选项字节

21.1 选项字节的功能

R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 的闪存中的地址 0080H ~ 0084H 为选项字节区域。在接通电源或者通过复位启动时，自动参考选项字节，并且设置指定的功能。在使用本产品时，必须使用选项字节设置如下所示的功能。

(1) 0080H

- 低速内部振荡器的运行
 - 可通过软件停止
 - 不可停止
- 看门狗定时器的上溢时间设置
- 看门狗定时器的计数器运行
 - 允许计数器运行
 - 禁止计数器运行
- 看门狗定时器的窗口打开期间设置

(2) 0081H

- 选择 POC 模式
 - 2.7V/1.59V POC 模式运行时 (POCMODE=1)
从接通电源开始到达到 2.7V(TYP.) 的期间处于复位状态，一旦超过 2.7V(TYP.)，就解除复位。之后，不执行 2.7V 下的 POC 检测，执行 1.59V(TYP.) 下的 POC 检测。
如果从接通电源开始到达到 4.0V 的电压的上升斜率低于 0.5V/ms(MIN.)，推荐使用 2.7V/1.59V POC 模式。
 - 1.59V POC 模式运行时 (POCMODE=0)
从接通电源开始到达到 1.59V(TYP.) 的期间处于复位状态，一旦超过 1.59V(TYP.)，就解除复位。之后，与接通电源时相同，执行 1.59V(TYP.) 下的 POC 检测。

(3) 0082H

必须设置为“00H”。

(4) 0083H

必须设置为“00H”。

(5) 0084H

- 片上调试器的运行控制 (只限 R7F0C999B)
 - 禁止片上调试器运行
 - 允许片上调试器运行，并且在片上调试安全 ID 认证失败时擦除闪存数据。
 - 允许片上调试器运行，并且在片上调试安全 ID 认证失败时不擦除闪存数据。

21.2 选项字节的格式

选项字节的格式如下所示。

图 21-1 选项字节的格式 (1/2)

地址: 0080H

7	6	5	4	3	2	1	0
0	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	LSROSC
WINDOW1	WINDOW0	看门狗定时器的窗口打开期间					
0	0	25%					
0	1	50%					
1	0	75%					
1	1	100%					
WDTON	看门狗定时器计数器 / 非法存取检测的运行控制						
0	禁止计数器运行 (复位解除后停止计数)、禁止非法存取检测运行						
1	允许计数器运行 (复位解除后开始计数)、允许非法存取检测运行						
WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间				
0	0	0	$2^{10}/f_{RL}$ (3.88ms)				
0	0	1	$2^{11}/f_{RL}$ (7.76ms)				
0	1	0	$2^{12}/f_{RL}$ (15.52ms)				
0	1	1	$2^{13}/f_{RL}$ (31.03ms)				
1	0	0	$2^{14}/f_{RL}$ (62.06ms)				
1	0	1	$2^{15}/f_{RL}$ (124.12ms)				
1	1	0	$2^{16}/f_{RL}$ (248.24ms)				
1	1	1	$2^{17}/f_{RL}$ (496.48ms)				
LSROSC	低速内部振荡器的运行						
0	可通过软件停止 (通过给 RCM 寄存器的 bit0 (LSRSTOP) 写“1”, 停止运行。)						
1	不可停止 (即使给 LSRSTOP 位写“1”, 也不停止运行。)						

注意 1. 禁止 WDCS2=WDCS1=WDCS0=0 和 WINDOW1=WINDOW0=0 的组合设置。

- 如果 LSROSC 为“0” (可通过软件停止), 与内部振荡模式寄存器 (RCM) 的 bit0 (LSRSTOP) 的设置无关, 在 HALT 模式和 STOP 模式中, 不向看门狗定时器提供计数时钟。

但是, 在以低速内部振荡时钟进行 8 位定时器 H1 运行时, 即使在 HALT/STOP 模式中, 也向 8 位定时器 H1 提供计数时钟。

- 必须给 bit7 写“0”。

备注 1. f_{RL} : 低速内部振荡时钟频率

- (): $f_{RL}=264\text{kHz}$ (MAX.) 的情况

图 21-1 选项字节的格式 (2/2)

地址: 0081H 注

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	POCMODE

POCMODE	选择 POC 模式
0	1.59V POC 模式 (默认)
1	2.7V/1.59V POC 模式

注 在更改 POC 模式的设置内容时, 如果批量擦除 (片擦除) 闪存后, 就必须重新给 0081H 设置值。指定块的存储器擦除后的设置更改无效。

注意 必须给 bit7 ~ bit1 写“0”。

地址: 0082H 注

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

注 0082H 为保留区域, 因此必须设置为“00H”。

地址: 0083H 注

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

注 0083H 为保留区域, 因此必须设置为“00H”。

地址: 0084H 注

7	6	5	4	3	2	1	0
0	0	0	0	0	0	OCDEN1	OCDEN0

OCDEN1	OCDEN0	片上调试器的运行控制
0	0	禁止运行
0	1	禁止设置
1	0	允许运行。在片上调试安全 ID 认证失败时, 不擦除闪存数据。
1	1	允许运行。在片上调试安全 ID 认证失败时, 擦除闪存数据。

注 在 R7F0C011B、R7F0C012B、R7F0C013B 产品中, 0084H 为保留区域, 因此必须设置为“00H”。

注意 在 R7F0C999B 产品中, 必须给 bit7 ~ bit2 写“0”。

备注 有关片上调试安全 ID, 请参照“第 23 章 片上调试功能 (只限 R7F0C999B) ”。

选项字节设置的软件描述示例如下所示。

OPT	CSEG	AT 0080H	
OPTION:	DB	30H	; 允许看门狗定时器运行（非法存取检测运行）、 ; 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 $2^{10}/f_{RL}$ 、 ; 可通过软件停止低速内部振荡器
	DB	00H	; 1.59V POC 模式
	DB	00H	; 保留区域
	DB	00H	; 保留区域
	DB	00H	; 禁止片上调试器运行

备注 在复位处理期间执行选项字节的参考。有关复位处理的时序，请参照“第 18 章 复位功能”。

第 22 章 闪存

R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 内置可在安装在电路板上的状态下进行编程、擦除和改写的闪存。

22.1 存储容量切换寄存器

存储容量切换寄存器（IMS）选择内部存储器的容量。

通过 8 位存储器操作指令设置 IMS。

在产生复位信号后，IMS 变为“CFH”。

注意 在复位解除后，必须给各产品设置表 22-1 所示的值。

图 22-1 存储容量切换寄存器（IMS）的格式

地址：FFF0H 复位后：CFH RW

符号	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM2	RAM1	RAM0	内部高速 RAM 容量的选择
0	0	0	768 字节
1	1	0	1024 字节
上述以外			禁止设置

ROM3	ROM2	ROM1	ROM0	内部 ROM 容量的选择
0	1	0	0	16K 字节
0	1	1	0	24K 字节
1	0	0	0	32K 字节
上述以外				禁止设置

表 22-1 存储容量切换寄存器（IMS）的设置值

产品	IMS 的设置值
R7F0C011B	04H
R7F0C012B	C6H
R7F0C013B	C8H
R7F0C999B	C8H

22.2 闪存编程器的编程方法

可通过专用闪存编程器进行板上或者板外编程。

(1) 板上编程

在目标系统上安装 R7F0C011B、R7F0C012B、R7F0C013B、R7F0C999B 后，可改写闪存内容。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

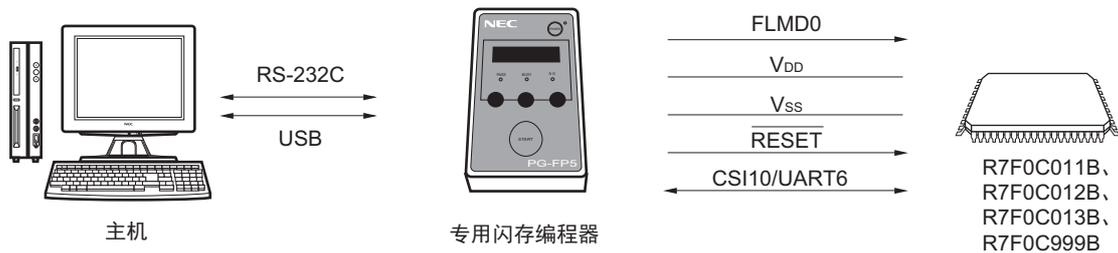
在将 R7F0C011B、R7F0C012B、R7F0C013B、R7F0C999B 安装到目标系统前，通过专用程序适配器（FA 系列）等进行闪存编程。

备注 FA 系列是株式会社内藤电诚町田制作所（Naito Densai Machida Mfg. Co., Ltd.）的产品。

22.3 编程环境

对 R7F0C011B、R7F0C012B、R7F0C013B、R7F0C999B 的闪存进行编程所需的环境如下所示。

图 22-2 闪存编程的环境



需要控制专用闪存编程器的主机。

专用闪存编程器与 R7F0C011B、R7F0C012B、R7F0C013B、R7F0C999B 的接口使用 CSI10 或者 UART6 进行编程和擦除。在进行板外编程时，需要专用程序适配器（FA 系列）。

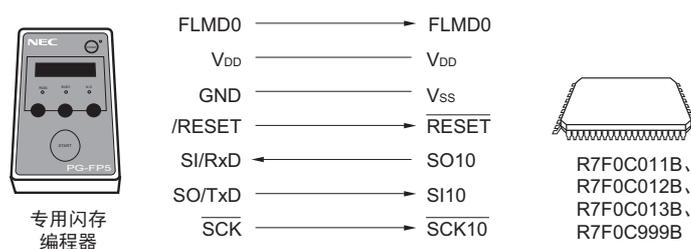
22.4 通信方式

通过 R7F0C011B、R7F0C012B、R7F0C013B、R7F0C999B 的 CSI10 或者 UART6 进行专用闪存编程器与 R7F0C011B、R7F0C012B、R7F0C013B、R7F0C999B 的串行通信。

(1) CSI10

传送速率：2.4kHz~2.5MHz

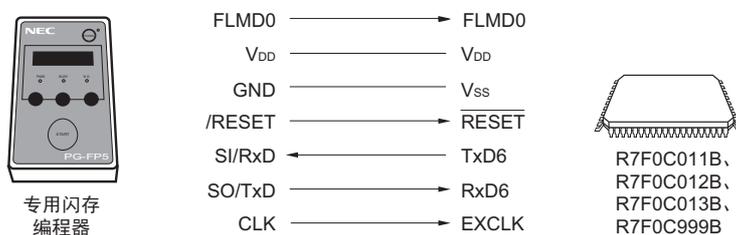
图 22-3 与专用闪存编程器的通信 (CSI10)



(2) UART6

传送速率：115200bps

图 22-4 与专用闪存编程器的通信 (UART6)



专用闪存编程器对 R7F0C011B、R7F0C012B、R7F0C013B、R7F0C999B 生成以下信号。详细内容请参照 PG-FP5、FL-PR5 的手册。

表 22-2 引脚连接一览

专用闪存编程器			R7F0C011B、R7F0C012B、 R7F0C013B、R7F0C999B	连接时的处理	
信号名称	输入 / 输出	引脚功能	引脚名称	CSI10	UART6
FLMD0	输出	模式信号	FLMD0	◎	◎
V _{DD}	输入 / 输出	V _{DD} 电压生成 / 电压监视	V _{DD}	◎	◎
GND	—	接地	V _{SS}	◎	◎
CLK	输出	R7F0C011B、R7F0C012B、 R7F0C013B、R7F0C999B 的时钟输出	EXCLK/X2/P122	X ^{注1}	○ ^{注2}
/RESET	输出	复位信号	RESET	◎	◎
SI/RxD	输入	接收信号	SO10/TxD6	◎	◎
SO/TxD	输出	发送信号	SI10/RxD6	◎	◎
SCK	输出	传送时钟	SCK10	◎	X

- 注 1. 在使用 CSI10 时，只能使用高速内部振荡时钟 (f_{RH})。
 2. 在使用 UART6 时，只能使用 X1 时钟 (f_X) 或者外部主系统时钟 (f_{EXCLK})。

备注 ◎：必须连接。
 ○：如果在目标电路板上生成，就不需要连接。
 X：不需要连接。

使用专用程序适配器（FA 系列）时，对于未使用的引脚，必须执行“表 2-2”或者“表 22-3”所示的处理。

表 22-3 连接编程闪存的适配器时的未使用引脚的处理（必须）

引脚名称	引脚处理
P00、P01	必须分别通过电阻连接到 V _{SS} ^{注1} 。
P10、P11	必须分别通过电阻连接到 V _{SS} ^{注2} 。
P14	必须分别通过电阻连接到 V _{SS} ^{注3} 。
P16、P17	必须分别通过电阻连接到 V _{SS} ^{注1} 。
P30 ~ P33	
P60、P61	必须分别通过电阻连接到 V _{SS} ，或者直接连接到 V _{SS} 。
P70、P71	必须分别通过电阻连接到 V _{SS} ^{注1} 。
P120	

- 注 1. 在设计成不能在编程闪存的适配器电路板上进行闪存编程过程中的通常运行模式转移时，可不通过电阻直接连接到 V_{SS}。
 2. 通过 CSI10 进行与专用闪存编程器的串行通信时，必须连接编程器。
 3. 通过 UART6 进行与专用闪存编程器的串行通信时，必须连接编程器。

22.5 板上引脚处理

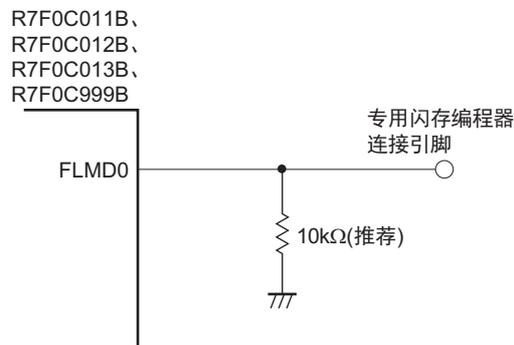
在进行板上编程时，必须在目标系统上安装连接专用闪存编程器的连接器。另外，必须在板上设置从正常运行模式切换到闪存编程模式的切换功能。

如果转移到闪存编程模式，所有不用于闪存编程的引脚的状态与复位后的状态相同。因此，在外部设备不识别复位后的状态时，必须进行引脚处理。

22.5.1 FLMD0 引脚的连接例子

在通常运行模式中，向 FLMD0 引脚输入 0V 电压。在闪存编程模式中，向 FLMD0 引脚提供 V_{DD} 电平的编程电压。FLMD0 引脚的连接例子如下所示。

图 22-5 FLMD0 引脚的连接例子



22.5.2 串行接口引脚

各串行接口使用的引脚如下所示。

表 22-4 各串行接口使用的引脚

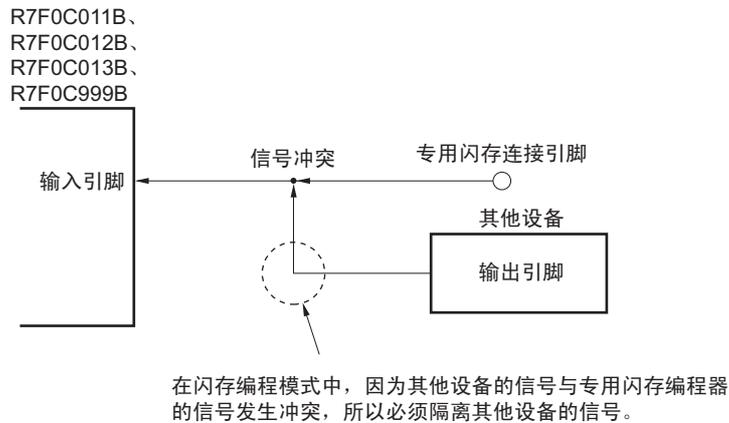
串行接口	使用引脚
CSI10	SO10、SI10、 $\overline{SCK10}$
UART6	TxD6、RxD6

将专用闪存编程器连接到在板上与其他设备连接的串行接口的引脚时，必须注意信号冲突和其他设备的异常运行等。

(1) 信号冲突

将专用闪存编程器连接到与其他设备（输出）连接的串行接口的引脚（输入）时，会发生信号冲突。为了避免发生信号冲突，必须隔离其他设备的连接，或者将其他设备置为高阻抗状态。

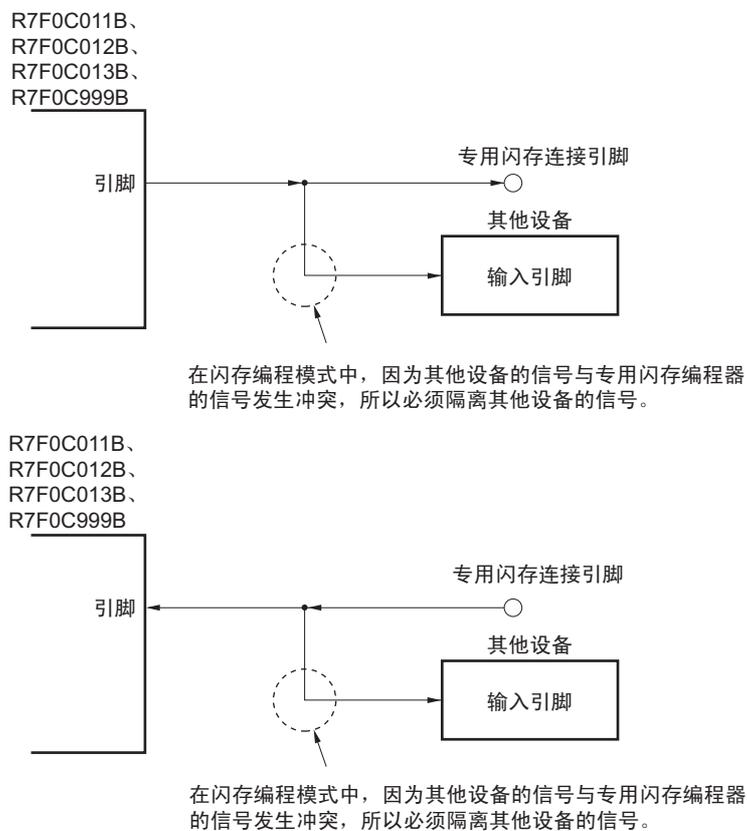
图 22-6 信号冲突（串行接口的输入引脚）



(2) 其他设备的异常运行

将专用闪存编程器（输出或者输入）连接到与其他设备（输入）连接的串行接口的引脚（输入或者输出）时，有可能从其他设备输出信号，并且发生异常运行。为了避免该异常运行，必须隔离其他设备的连接。

图 22-7 其他设备的异常运行

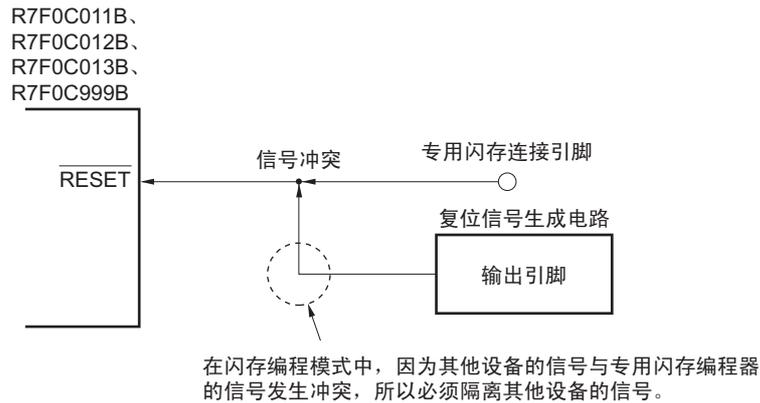


22.5.3 $\overline{\text{RESET}}$ 引脚

在板上将专用闪存编程器的复位信号连接到与复位信号生成电路连接的 $\overline{\text{RESET}}$ 引脚时，发生信号冲突。为了避免发生信号冲突，必须隔离复位信号生成电路的连接。

如果在闪存编程模式中从用户系统输入复位信号，就不能执行正常的编程，因此，不能输入专用闪存编程器复位信号以外的信号。

图 22-8 信号冲突 ($\overline{\text{RESET}}$ 引脚)



22.5.4 端口引脚

如果转移到闪存编程模式，所有不用于闪存编程的引脚的状态与复位后的状态相同。因此，在连接各端口的的外部设备不识别复位后的状态时，必须进行通过电阻连接 V_{DD} 或者 V_{SS} 等引脚处理。

22.5.5 REGC 引脚

与正常运行时相同，REGC 引脚必须通过电容器 ($0.47 \sim 1\mu\text{F}$) 连接 V_{SS} 。

22.5.6 其他信号的引脚

在使用板上的时钟时，X1、X2 以与正常运行模式时相同的状态进行连接。

但是，从专用闪存编程器输入运行时时钟时，编程器的 CLK 必须与 EXCLK/X2/P122 连接。

- 注意 1. 在使用 CSI10 时，只能使用高速内部振荡时钟 (f_{RH})。
2. 在使用 UART6 时，只能使用 X1 时钟 (f_X) 或者外部主系统时钟 (f_{EXCLK})。

22.5.7 电源

在使用闪存编程器的电源输出时，必须将 V_{DD} 引脚连接到闪存编程器的 V_{DD} ，将 V_{SS} 引脚连接到闪存编程器的 GND。

在使用板上电源时，必须符合正常运行模式时的标准进行连接。

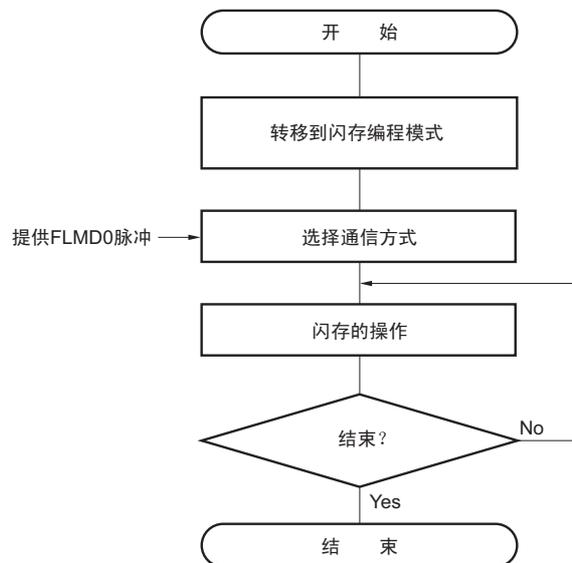
但是，即使在使用板上电源时，也通过闪存编程器监视电压，因此， V_{DD} 引脚和 V_{SS} 引脚必须分别连接闪存编程器的 V_{DD} 和 GND。

22.6 编程方法

22.6.1 闪存控制

闪存的操作步骤如下所示。

图 22-9 闪存的操作步骤



22.6.2 闪存编程模式

在使用专用闪存编程器改写闪存的内容时，必须将 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 设置为闪存编程模式。如果要进行模式转移，将 FLMD0 引脚设置为 V_{DD} 后，解除复位。在进行板上编程时，通过跳线等切换模式。

图 22-10 闪存编程模式

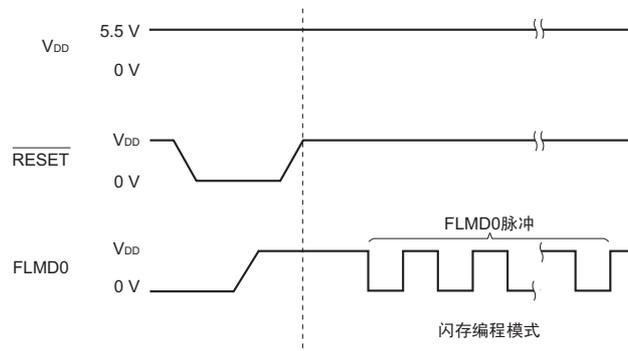


表 22-5 与复位解除时的 FLMD0 引脚的运行模式的关系

FLMD0	运行模式
0	正常运行模式
V_{DD}	闪存编程模式

22.6.3 通信方式的选择

在 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 转移到闪存编程模式后，通过向 FLMD0 引脚输入脉冲选择通信方式。该 FLMD0 脉冲由专用闪存编程器生成。

脉冲数和通信方式的关系如下所示。

表 22-6 通信方式一览

通信方式	Standard 设置注 1				使用引脚	外围时钟	FLMD0 脉冲数
	Port	Speed	Frequency	Multiply Rate			
UART (UART6)	UART-Ext-Osc	115200bps 注 3	2M ~ 20MHz 注 2	1.0	TxD6、RxD6	f_X	0
	UART-Ext-FP5CK					f_{EXCLK}	3
3 线式串行 I/O (CSI10)	CSI-Internal-OSC	2.4kHz ~ 25MHz	—		SO10、SI10、SCK10	f_{RH}	8

- 注 1. 这是在闪存编程器的 GUI 上的 Standard 设置中的设置项目。
 2. 可设置的范围因电压而不同。详细内容请参照电特性。
 3. UART 通信除了波特率误差，还受信号波形变钝等影响，因此必须在进行评估后再使用。

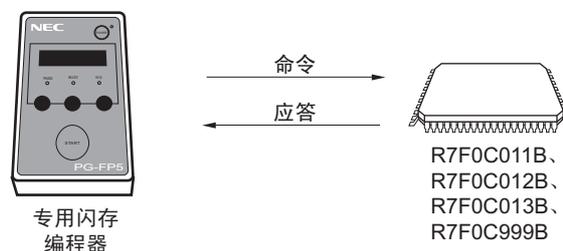
注意 如果选择 UART6，就在接受 FLMD0 脉冲后，以从专用闪存编程器发送来的复位命令为基准计算接收时钟。

备注 f_X : X1 时钟
 f_{EXCLK} : 外部主系统时钟
 f_{RH} : 高速内部振荡时钟

22.6.4 通信命令

R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 和专用闪存编程器通过命令进行通信。从专用闪存编程器向 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 发送的信号称为“命令”，从 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 向专用闪存编程器发送的信号称为“应答”。

图 22-11 通信命令



R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 的闪存控制命令如下表所示。所有这些命令都由编程器发行，R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 根据命令进行相应的处理。

表 22-7 闪存控制命令

类别	命令名称	功能
检验	Verify	比较闪存指定区域的内容与编程器的发送数据
擦除	Chip Erase	擦除整个闪存
	Block Erase	擦除指定区域的闪存
空白检查	Block Blank Check	检查指定块的闪存擦除状态
编程	Programming	将数据写入闪存的指定区域
取信息	Status	获取当前的运行状况（状态、数据）
	Silicon Signature	获取 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 信息（产品名称、闪存结构等）
	Version Get	获取产品版本、固件版本
	Checksum	获取指定区域的校验和数据
安全	Security Set	设置安全信息
其他	Reset	用于通信的同步检测
	Oscillating Frequency Set	指定振荡频率

R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 对专用闪存编程器发行的命令返回应答。
R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 发出的应答名称如下所示。

表 22-8 应答名称

应答名称	功能
ACK	命令 / 数据等的应答
NAK	非法命令 / 数据等的应答

22.7 安全设置

R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 支持禁止改写闪存中写入的用户程序的安全功能，因此，可防止第三方篡改程序等。

可通过 Security Set（安全设置）命令，执行以下操作。安全设置从下一个编程模式开始有效。

- 禁止批量擦除（片擦除）
在进行板上/板外编程时，禁止对闪存内的全部块执行块擦除命令和批量擦除（片擦除）命令。一旦设置为禁止，就不能执行批量擦除（片擦除）命令，因此，所有的禁止设置（包括禁止批量擦除（片擦除））都无法解除。

注意 在进行批量擦除的安全设置后，就不能再对该设备进行擦除。另外，即使执行编程命令，因为擦除命令无效，也不能写与已写入闪存的数据不同的数据。

- 禁止块擦除
在进行板上/板外编程时，禁止执行闪存内的块擦除命令。
- 禁止编程
在进行板上/板外编程时，禁止对闪存内的全部块执行编程命令和块擦除命令。
- 禁止改写引导簇 0
禁止对闪存内的引导簇 0（0000H ~ 0FFFH）执行块擦除命令和编程命令。另外，还禁止执行批量擦除（片擦除）命令。

注意 在进行了引导簇 0 的改写安全设置后，就无法对该设备进行引导簇 0 的改写，以及批量擦除（片擦除）。

出厂时的初始状态为允许所有批量擦除（片擦除）/块擦除/编程/引导簇 0 的改写。可在板上/板外编程和自编程时进行安全设置。可同时组合使用各安全设置。

可通过执行批量擦除（片擦除）命令，解除禁止块擦除和禁止编程。

在 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 的安全功能有效时，擦除和编程命令的关系如表 22-9 所示。

表 22-9 在安全功能有效时与命令的关系

●板上 / 板外编程时

安全有效	执行命令		
	批量擦除（片擦除）	块擦除	编程
禁止批量擦除（片擦除）	不能批量擦除	不能块擦除	能编程 ^注
禁止块擦除	能批量擦除		能编程
禁止编程			不能编程
禁止改写引导簇 0	不能批量擦除	不能擦除引导簇 0	不能编程引导簇 0

注 确认编程区域中没有写入数据。因为在设置禁止批量擦除（片擦除）后就不能进行擦除，所以在数据还没有被擦除时，不能写入数据。

各编程模式中的安全设置方法如表 22-10 所示。

表 22-10 各编程模式中的安全设置方法

●板上 / 板外编程

安全	安全设置方法	将安全设置为无效的方法
禁止批量擦除（片擦除）	在专用闪存编程器的 GUI 上等设置	设置后不能置为无效
禁止块擦除		执行批量擦除（片擦除）命令
禁止编程		
禁止改写引导簇 0		设置后不能置为无效

第 23 章 片上调试功能（只限 R7F0C999B）

23.1 QB-MINI2 与 R7F0C999B 的连接

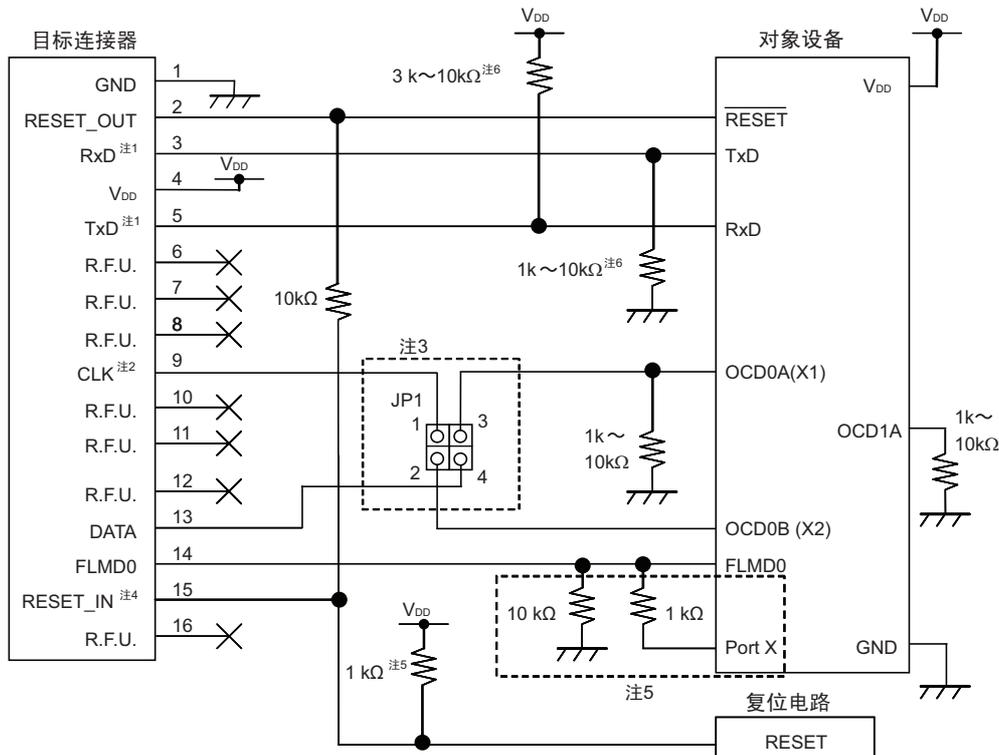
R7F0C999B 在通过支持片上调试功能的片上调试仿真器（QB-MINI2）进行与主机间的通信时，使用 V_{DD} 、 $\overline{\text{RESET}}$ 、 V_{SS} 引脚和 OCD0A/X1、OCD1A/P31、OCD0B/X2 或者 OCD1B/P32 引脚。

注意 1. R7F0C999B 搭载用于开发 / 评估的片上调试功能。在使用片上调试功能时，有可能发生超过闪存所定改写次数的情况，从而无法保证产品的可靠性，因此，请勿将本功能用于量化的产品。瑞萨电子公司不接受任何使用了片上调试功能的产品的投诉。

2. 在片上调试期间转移到 STOP 模式时，虽然高速内部振荡器继续振荡，但是不影响片上调试运行。

图 23-1 QB-MINI2 与 R7F0C999B 的连接示例（1/4）

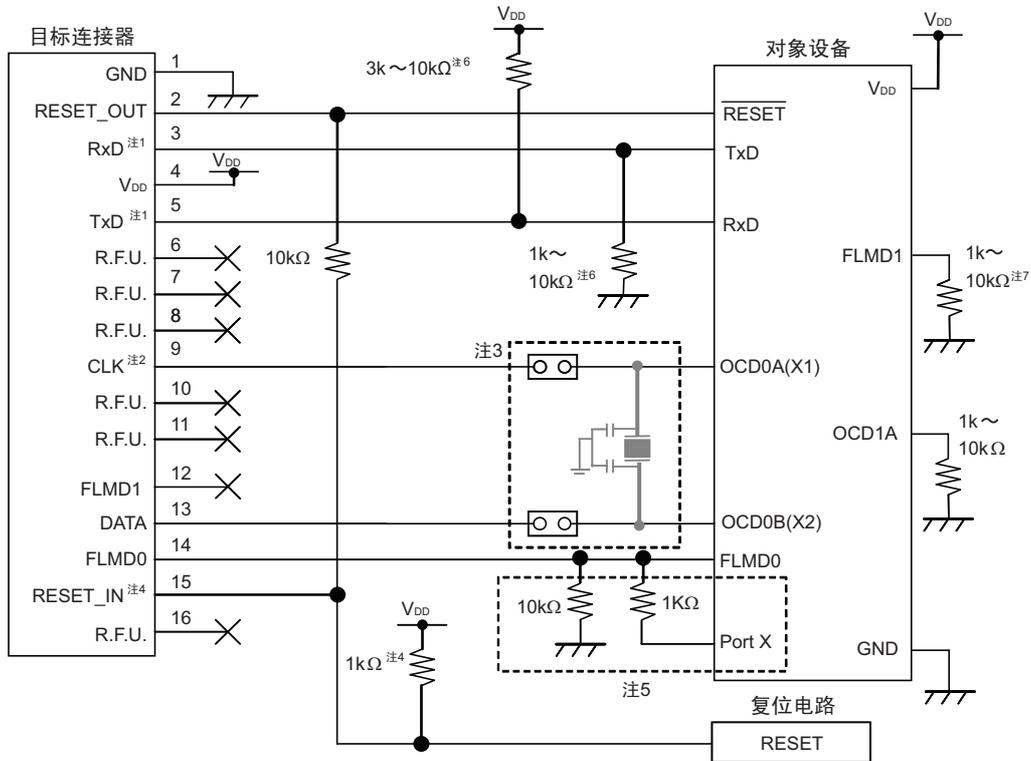
(1) 调试和编程都执行的情况（未使用 OCD0A、OCD0B 通信 / X1 振荡电路）



- 注 1. 对象设备的 TxD（发送方）连接目标连接器的 RxD（接收方），目标连接器的 TxD（发送方）连接对象设备的 RxD（接收方）。
2. 在片上调试期间，可使用 X1 振荡电路以外的时钟作为对象设备的运行时钟。在闪存编程期间，只能使用在 QB-MINI2 内部生成的 4/8/16MHz 的时钟。
3. 在片上调试期间，JP1 的 1-3 必须置为短路，JP1 的 2-4 必须置为短路。在闪存编程期间，JP1 的 1-2 必须置为短路，3-4 必须置为开路。在对象设备中运行时（未连接 QB-MINI2），JP1 必须置为开路。
4. 这是将复位电路的输出假设为 N 沟道漏极开路缓冲器（输出电阻：100Ω 以下）的电路连接。
5. 这是闪存自编程时在用户程序中控制 FLMD0 引脚时的电路。任意 1 个可输出端口通过电阻连接 FLMD0。在不使用闪存自编程时，根据设备的规格进行引脚处理。
6. 这是在对象设备中运行时（未连接 QB-MINI2），未使用（输入开路）时的引脚处理。

图 23-1 QB-MINI2 与 R7F0C999B 的连接示例（2/4）

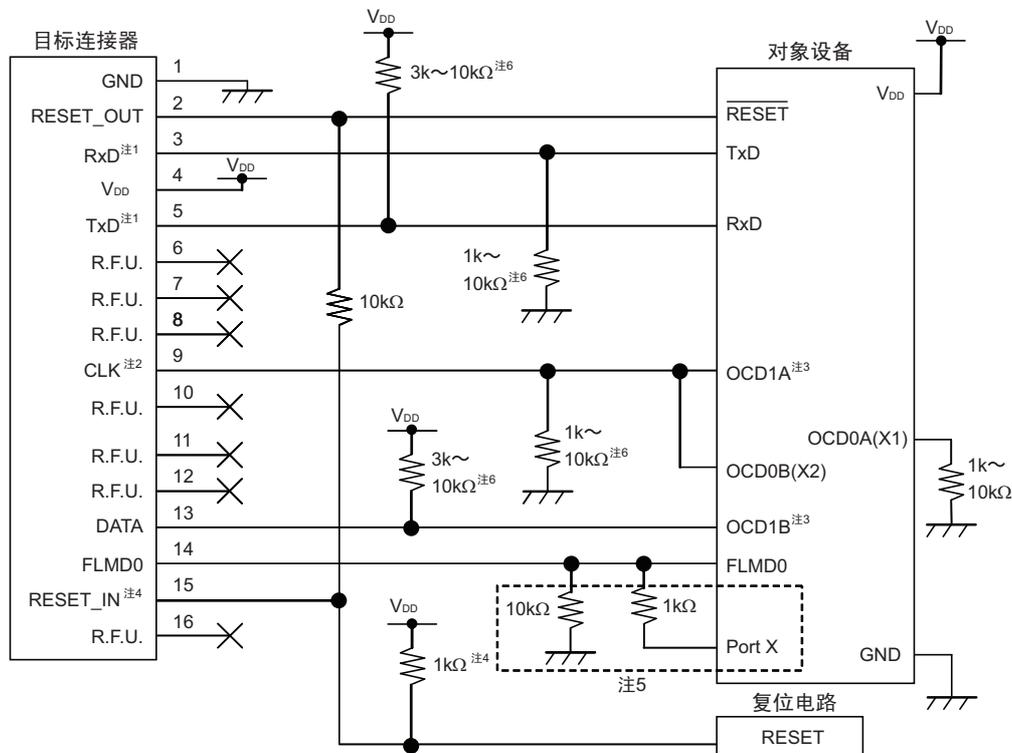
(2) 调试和编程都执行的情况（使用 OCD0A、OCD0B 通信 /X1 振荡电路）



- 注 1. 对象设备的 TxD（发送方）连接目标连接器的 RxD（接收方），目标连接器的 TxD（发送方）连接对象设备的 RxD（接收方）。
2. 在片上调试期间，不能使用目标系统中的 X1 振荡电路作为对象设备的运行时钟。
必须使用由 QB-MINI2 提供的时钟（也可用安装在 78K0-OCD 板上的振荡器或者作为系统时钟选择的 4/8/16MHz 时钟代替）。
3. 在片上调试期间，必须将跨接线置为短路，并且不安装时钟电路。
在闪存编程期间，必须在对象设备中运行时（未连接 QB-MINI2）将跨接线置为开路，并且安装时钟电路。
4. 这是将复位电路的输出假设为 N 沟道漏极开路缓冲器（输出电阻：100Ω 以下）的电路连接。
5. 这是闪存自编程时在用户程序中控制 FLMD0 引脚时的电路。任意 1 个可输出端口通过电阻连接 FLMD0。在不使用闪存自编程时，根据设备的规格进行引脚处理。
6. 这是在对象设备中运行时（未连接 QB-MINI2），未使用（输入开路）时的引脚处理。
7. 如果要使用 FLMD1，必须与 QB-MINI2 的 FLMD1 连接。

图 23-1 QB-MINI2 与 R7F0C999B 的连接示例（3/4）

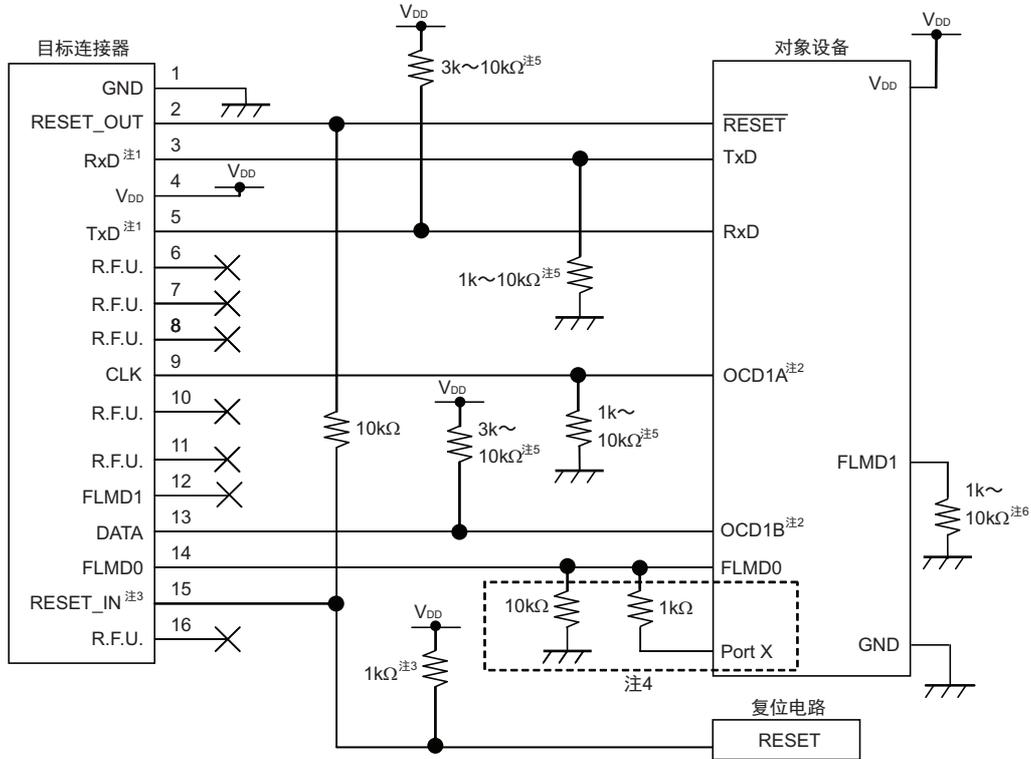
(3) 调试和编程都执行的情况（不使用 OCD1A、OCD1B 通信 /X1 振荡电路）



- 注 1. 对象设备的 TxD（发送方）连接目标连接器的 RxD（接收方），目标连接器的 TxD（发送方）连接对象设备的 RxD（接收方）。
2. 在片上调试期间，可使用 X1 振荡电路以外的时钟作为对象设备的运行时钟。
3. 在片上调试期间，这些引脚用作片上调试专用引脚，所以忽视在用户程序中的设置，但是，在设置为输入引脚时，必须进行引脚处理（因为在未连接 QB-MINI2 时置为开路）。
4. 这是将复位电路的输出假设为 N 沟道漏极开路缓冲器（输出电阻：100Ω 以下）的电路连接。
5. 这是闪存自编程时在用户程序中控制 FLMD0 引脚时的电路。任意 1 个可输出端口通过电阻连接 FLMD0。在不使用闪存自编程时，根据设备的规格进行引脚处理。
6. 这是在对象设备中运行时（未连接 QB-MINI2），未使用（输入开路）时的引脚处理。

图 23-1 QB-MINI2 与 R7F0C999B 的连接示例（4/4）

(4) 调试和编程都执行的情况（使用 OCD1A、OCD1B 通信 /X1 振荡电路）



- 注 1. 对象设备的 TxD（发送方）连接目标连接器的 RxD（接收方），目标连接器的 TxD（发送方）连接对象设备的 RxD（接收方）。
- 注 2. 在片上调试期间，这些引脚用作片上调试专用引脚，所以忽视在用户程序中的设置，但是，在设置为输入引脚时，必须进行引脚处理（因为在未连接 QB-MINI2 时置为开路）。
- 注 3. 这是将复位电路的输出假设为 N 沟道漏极开路缓冲器（输出电阻：100Ω 以下）的电路连接。
- 注 4. 这是闪存自编程时在用户程序中控制 FLMD0 引脚时的电路。任意 1 个可输出端口通过电阻连接 FLMD0。在不使用闪存自编程时，根据设备的规格进行引脚处理。
- 注 5. 这是在对象设备中运行时（未连接 QB-MINI2），未使用（输入开路）时的引脚处理。
- 注 6. 如果要使用 FLMD1，必须与 QB-MINI2 的 FLMD1 连接。

23.2 片上调试安全 ID

在 R7F0C999B 中，为了防止第三方读取存储器的内容，在闪存的地址 0084H 中预留片上调试运行控制位（参照“第 21 章 选项字节”），在地址 0085H ~ 008EH 中预留片上调试安全 ID 设置区域。

有关片上调试安全 ID 的详细内容，请参照《QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual》（R20UT0449EJ）。

表 23-1 片上调试安全 ID

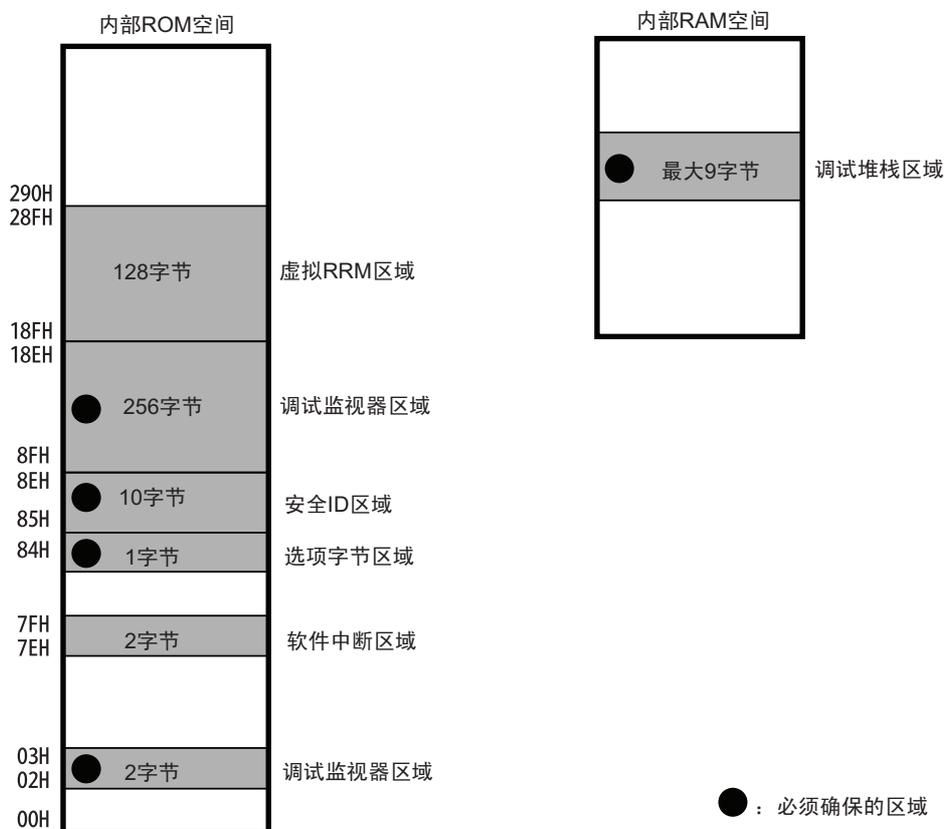
地址	片上调试安全 ID
0085H ~ 008EH	任意 10 字节的 ID 码
1085H ~ 108EH	

23.3 保留用户资源

QB-MINI2 为了实现与对象设备的通信或者各调试功能，使用图 23-2 中灰色部分所示的用户存储空间。图中以“●”表示的区域在调试期间必须使用，其他空间根据使用的调试功能进行使用。

通过用户程序或者连接程序选项来保留这些区域。详细内容请参照《QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual》（R20UT0449EJ）。

图 23-2 QB-MINI2 使用的保留区域



第 24 章 指令集

通过一览表表示 R7F0C011B、R7F0C012B、R7F0C013B 以及 R7F0C999B 的指令集。有关各指令的详细运行和机器指令（指令码），请参照《78K/0 系列指令用户手册》（U12326C）。

24.1 凡例

24.1.1 操作数的标识符和描述方法

在各指令的操作数栏中，根据其指令的操作数标识符对应的描述方法描述操作数（详细内容参照汇编程序规范）。如果有多种描述方法，就选择其中一种。大写字母和符号 #、!、\$、[] 为关键字，描述时应保持原样。各符号含义如下所示。

- #: 指定立即数
- !: 指定绝对地址
- \$: 指定相对地址
- []: 指定间接地址

立即数可以由适当的数值或者标号来描述。在通过标号描述时，必须使用符号 #、!、\$ 和 []。

操作数的寄存器的标识符 r 和 rp，可以使用功能名称（X、A、C 等）或者绝对名称（下表中括号内的名称，R0、R1、R2 等）进行描述。

表 24-1 操作数的标识符和描述方法

标识符	描述方法
r	X(R0)、A(R1)、C(R2)、B(R3)、E(R4)、D(R5)、L(R6)、H(R7)、AX(RP0)、
rp	BC(RP1)、DE(RP2)、HL(RP3)
sfr	特殊功能寄存器符号 ^注
sfrp	特殊功能寄存器符号（只限可 16 位操作的寄存器的偶数地址） ^注
saddr	FE20H ~ FF1FH 立即数或者标号
saddrp	FE20H ~ FF1FH 立即数或标号（只限偶数地址）
addr16	0000H ~ FFFFH 立即数或者标号 （只限 16 位数据传送指令时的偶数地址）
addr11	0800H ~ 0FFFH 立即数或者标号
addr5	0040H ~ 007FH 立即数或者标号（只限偶数地址）
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 地址 FFD0H ~ FFD7H 不能进行寻址。

备注 有关特殊功能寄存器的符号请参照“表 3-6 特殊功能寄存器”。

24.1.2 操作栏的说明

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
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:	载波标志
AC:	辅助载波标志
Z:	零标志
RBS:	寄存器组选择标志
IE:	中断请求允许标志
():	通过括号中的地址或者寄存器的内容表示的存储器内容
X _H 、X _L :	16 位寄存器的高 8 位和低 8 位
∧:	逻辑与 (AND)
∨:	逻辑或 (OR)
⊕:	异或 (exclusive OR)
—:	数据取反
addr16:	16 位立即数或者标号
jdisp8:	带符号的 8 位数据 (偏移量)

24.1.3 标志栏的说明

(Blank):	无变化
0:	清“0”
1:	置“1”
x:	根据结果设置 / 清除
R:	恢复之前保存的值

24.2 操作一览

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位 数 据 传 送	MOV	r, #byte	2	4	—	$r \leftarrow \text{byte}$			
		saddr, #byte	3	6	7	$(\text{saddr}) \leftarrow \text{byte}$			
		sfr, #byte	3	—	7	$\text{sfr} \leftarrow \text{byte}$			
		A, r 注 3	1	2	—	$A \leftarrow r$			
		r, A 注 3	1	2	—	$r \leftarrow A$			
		A, saddr	2	4	5	$A \leftarrow (\text{saddr})$			
		saddr, A	2	4	5	$(\text{saddr}) \leftarrow A$			
		A, sfr	2	—	5	$A \leftarrow \text{sfr}$			
		sfr, A	2	—	5	$\text{sfr} \leftarrow A$			
		A, !addr16	3	8	9	$A \leftarrow (\text{addr16})$			
		!addr16, A	3	8	9	$(\text{addr16}) \leftarrow A$			
		PSW, #byte	3	—	7	$\text{PSW} \leftarrow \text{byte}$	x	x	x
		A, PSW	2	—	5	$A \leftarrow \text{PSW}$			
		PSW, A	2	—	5	$\text{PSW} \leftarrow A$	x	x	x
		A, [DE]	1	4	5	$A \leftarrow (\text{DE})$			
		[DE], A	1	4	5	$(\text{DE}) \leftarrow A$			
		A, [HL]	1	4	5	$A \leftarrow (\text{HL})$			
		[HL], A	1	4	5	$(\text{HL}) \leftarrow A$			
		A, [HL + byte]	2	8	9	$A \leftarrow (\text{HL} + \text{byte})$			
		[HL + byte], A	2	8	9	$(\text{HL} + \text{byte}) \leftarrow A$			
	A, [HL + B]	1	6	7	$A \leftarrow (\text{HL} + \text{B})$				
	[HL + B], A	1	6	7	$(\text{HL} + \text{B}) \leftarrow A$				
	A, [HL + C]	1	6	7	$A \leftarrow (\text{HL} + \text{C})$				
	[HL + C], A	1	6	7	$(\text{HL} + \text{C}) \leftarrow A$				
	XCH	A, r 注 3	1	2	—	$A \leftrightarrow r$			
		A, saddr	2	4	6	$A \leftrightarrow (\text{saddr})$			
		A, sfr	2	—	6	$A \leftrightarrow (\text{sfr})$			
		A, !addr16	3	8	10	$A \leftrightarrow (\text{addr16})$			
A, [DE]		1	4	6	$A \leftrightarrow (\text{DE})$				
A, [HL]		1	4	6	$A \leftrightarrow (\text{HL})$				
A, [HL + byte]		2	8	10	$A \leftrightarrow (\text{HL} + \text{byte})$				
A, [HL + B]		2	8	10	$A \leftrightarrow (\text{HL} + \text{B})$				
A, [HL + C]	2	8	10	$A \leftrightarrow (\text{HL} + \text{C})$					

注 1. 这是存取内部高速 RAM 区域时或者执行不进行数据存取的指令时的时钟数。

2. 这是存取内部高速 RAM 以外区域时的时钟数。

3. “r=A” 除外

备注 1. 1 个指令时钟是指 1 个通过处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU})。

2. 这是在内部 ROM 区域执行程序时的时钟数。

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	rp, #word	3	6	—	$rp \leftarrow \text{word}$			
		saddrp, #word	4	8	10	$(saddrp) \leftarrow \text{word}$			
		sfrp, #word	4	—	10	$sfrp \leftarrow \text{word}$			
		AX, saddrp	2	6	8	$AX \leftarrow (saddrp)$			
		saddrp, AX	2	6	8	$(saddrp) \leftarrow AX$			
		AX, sfrp	2	—	8	$AX \leftarrow sfrp$			
		sfrp, AX	2	—	8	$sfrp \leftarrow AX$			
		AX, rp ^{注3}	1	4	—	$AX \leftarrow rp$			
		rp, AX ^{注3}	1	4	—	$rp \leftarrow AX$			
		AX, !addr16	3	10	12	$AX \leftarrow (\text{addr16})$			
	!addr16, AX	3	10	12	$(\text{addr16}) \leftarrow AX$				
XCHW	AX, rp ^{注3}	1	4	—	$AX \leftrightarrow rp$				
8 位运算	ADD	A, #byte	2	4	—	$A, CY \leftarrow A + \text{byte}$	x	x	x
		saddr, #byte	3	6	8	$(saddr), CY \leftarrow (saddr) + \text{byte}$	x	x	x
		A, r ^{注4}	2	4	—	$A, CY \leftarrow A + r$	x	x	x
		r, A	2	4	—	$r, CY \leftarrow r + A$	x	x	x
		A, saddr	2	4	5	$A, CY \leftarrow A + (saddr)$	x	x	x
		A, !addr16	3	8	9	$A, CY \leftarrow A + (\text{addr16})$	x	x	x
		A, [HL]	1	4	5	$A, CY \leftarrow A + (\text{HL})$	x	x	x
		A, [HL + byte]	2	8	9	$A, CY \leftarrow A + (\text{HL} + \text{byte})$	x	x	x
		A, [HL + B]	2	8	9	$A, CY \leftarrow A + (\text{HL} + B)$	x	x	x
	A, [HL + C]	2	8	9	$A, CY \leftarrow A + (\text{HL} + C)$	x	x	x	
	ADDC	A, #byte	2	4	—	$A, CY \leftarrow A + \text{byte} + CY$	x	x	x
		saddr, #byte	3	6	8	$(saddr), CY \leftarrow (saddr) + \text{byte} + CY$	x	x	x
		A, r ^{注4}	2	4	—	$A, CY \leftarrow A + r + CY$	x	x	x
		r, A	2	4	—	$r, CY \leftarrow r + A + CY$	x	x	x
		A, saddr	2	4	5	$A, CY \leftarrow A + (saddr) + CY$	x	x	x
		A, !addr16	3	8	9	$A, CY \leftarrow A + (\text{addr16}) + CY$	x	x	x
		A, [HL]	1	4	5	$A, CY \leftarrow A + (\text{HL}) + CY$	x	x	x
		A, [HL + byte]	2	8	9	$A, CY \leftarrow A + (\text{HL} + \text{byte}) + CY$	x	x	x
		A, [HL + B]	2	8	9	$A, CY \leftarrow A + (\text{HL} + B) + CY$	x	x	x
A, [HL + C]		2	8	9	$A, CY \leftarrow A + (\text{HL} + C) + CY$	x	x	x	

注 1. 这是存取内部高速 RAM 区域时或者执行不进行数据存取的指令时的时钟数。

2. 这是存取内部高速 RAM 以外区域时的时钟数。

3. 只在 rp=BC、DE 或者 HL 时

4. “r=A” 除外

备注 1. 1 个指令时钟是指 1 个通过处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU})。

2. 这是在内部 ROM 区域执行程序时的时钟数。

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	SUB	A, #byte	2	4	—	$A, CY \leftarrow A - \text{byte}$	x	x	x
		saddr, #byte	3	6	8	$(saddr), CY \leftarrow (saddr) - \text{byte}$	x	x	x
		A, r 注 3	2	4	—	$A, CY \leftarrow A - r$	x	x	x
		r, A	2	4	—	$r, CY \leftarrow r - A$	x	x	x
		A, saddr	2	4	5	$A, CY \leftarrow A - (saddr)$	x	x	x
		A, !addr16	3	8	9	$A, CY \leftarrow A - (\text{addr16})$	x	x	x
		A, [HL]	1	4	5	$A, CY \leftarrow A - (\text{HL})$	x	x	x
		A, [HL + byte]	2	8	9	$A, CY \leftarrow A - (\text{HL} + \text{byte})$	x	x	x
		A, [HL + B]	2	8	9	$A, CY \leftarrow A - (\text{HL} + B)$	x	x	x
		A, [HL + C]	2	8	9	$A, CY \leftarrow A - (\text{HL} + C)$	x	x	x
	SUBC	A, #byte	2	4	—	$A, CY \leftarrow A - \text{byte} - CY$	x	x	x
		saddr, #byte	3	6	8	$(saddr), CY \leftarrow (saddr) - \text{byte} - CY$	x	x	x
		A, r 注 3	2	4	—	$A, CY \leftarrow A - r - CY$	x	x	x
		r, A	2	4	—	$r, CY \leftarrow r - A - CY$	x	x	x
		A, saddr	2	4	5	$A, CY \leftarrow A - (saddr) - CY$	x	x	x
		A, !addr16	3	8	9	$A, CY \leftarrow A - (\text{addr16}) - CY$	x	x	x
		A, [HL]	1	4	5	$A, CY \leftarrow A - (\text{HL}) - CY$	x	x	x
		A, [HL + byte]	2	8	9	$A, CY \leftarrow A - (\text{HL} + \text{byte}) - CY$	x	x	x
		A, [HL + B]	2	8	9	$A, CY \leftarrow A - (\text{HL} + B) - CY$	x	x	x
		A, [HL + C]	2	8	9	$A, CY \leftarrow A - (\text{HL} + C) - CY$	x	x	x
	AND	A, #byte	2	4	—	$A \leftarrow A \wedge \text{byte}$	x		
		saddr, #byte	3	6	8	$(saddr) \leftarrow (saddr) \wedge \text{byte}$	x		
		A, r 注 3	2	4	—	$A \leftarrow A \wedge r$	x		
		r, A	2	4	—	$r \leftarrow r \wedge A$	x		
		A, saddr	2	4	5	$A \leftarrow A \wedge (saddr)$	x		
		A, !addr16	3	8	9	$A \leftarrow A \wedge (\text{addr16})$	x		
		A, [HL]	1	4	5	$A \leftarrow A \wedge (\text{HL})$	x		
		A, [HL + byte]	2	8	9	$A \leftarrow A \wedge (\text{HL} + \text{byte})$	x		
		A, [HL + B]	2	8	9	$A \leftarrow A \wedge (\text{HL} + B)$	x		
		A, [HL + C]	2	8	9	$A \leftarrow A \wedge (\text{HL} + C)$	x		

注 1. 这是存取内部高速 RAM 区域时或者执行不进行数据存取的指令时的时钟数。

2. 这是存取内部高速 RAM 以外区域时的时钟数。

3. “r=A” 除外

备注 1. 1 个指令时钟是指 1 个通过处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU})。

2. 这是在内部 ROM 区域执行程序时的时钟数。

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	4	—	$A \leftarrow A \vee \text{byte}$	x		
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	x		
		A, r ^{注 3}	2	4	—	$A \leftarrow A \vee r$	x		
		r, A	2	4	—	$r \leftarrow r \vee A$	x		
		A, saddr	2	4	5	$A \leftarrow A \vee (\text{saddr})$	x		
		A, !addr16	3	8	9	$A \leftarrow A \vee (\text{addr}16)$	x		
		A, [HL]	1	4	5	$A \leftarrow A \vee (\text{HL})$	x		
		A, [HL+byte]	2	8	9	$A \leftarrow A \vee (\text{HL} + \text{byte})$	x		
		A, [HL+B]	2	8	9	$A \leftarrow A \vee (\text{HL} + B)$	x		
	A, [HL+C]	2	8	9	$A \leftarrow A \vee (\text{HL} + C)$	x			
	XOR	A, #byte	2	4	—	$A \leftarrow A \vee \text{byte}$	x		
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	x		
		A, r ^{注 3}	2	4	—	$A \leftarrow A \vee r$	x		
		r, A	2	4	—	$r \leftarrow r \vee A$	x		
		A, saddr	2	4	5	$A \leftarrow A \vee (\text{saddr})$	x		
		A, !addr16	3	8	9	$A \leftarrow A \vee (\text{addr}16)$	x		
		A, [HL]	1	4	5	$A \leftarrow A \vee (\text{HL})$	x		
		A, [HL+byte]	2	8	9	$A \leftarrow A \vee (\text{HL} + \text{byte})$	x		
		A, [HL+B]	2	8	9	$A \leftarrow A \vee (\text{HL} + B)$	x		
	A, [HL+C]	2	8	9	$A \leftarrow A \vee (\text{HL} + C)$	x			
	CMP	A, #byte	2	4	—	$A - \text{byte}$	x	x	x
		saddr, #byte	3	6	8	$(\text{saddr}) - \text{byte}$	x	x	x
		A, r ^{注 3}	2	4	—	$A - r$	x	x	x
		r, A	2	4	—	$r - A$	x	x	x
		A, saddr	2	4	5	$A - (\text{saddr})$	x	x	x
		A, !addr16	3	8	9	$A - (\text{addr}16)$	x	x	x
		A, [HL]	1	4	5	$A - (\text{HL})$	x	x	x
A, [HL+byte]		2	8	9	$A - (\text{HL} + \text{byte})$	x	x	x	
A, [HL+B]		2	8	9	$A - (\text{HL} + B)$	x	x	x	
A, [HL+C]	2	8	9	$A - (\text{HL} + C)$	x	x	x		
16 位运算	ADDW	AX, #word	3	6	—	$AX, CY \leftarrow AX + \text{word}$	x	x	x
	SUBW	AX, #word	3	6	—	$AX, CY \leftarrow AX - \text{word}$	x	x	x
	CMPW	AX, #word	3	6	—	$AX - \text{word}$	x	x	x
乘除运算	MULU	X	2	16	—	$AX \leftarrow A \times X$			
	DIVUW	C	2	25	—	$AX(\text{商}), C(\text{余数}) \leftarrow AX \div C$			

- 注 1. 这是存取内部高速 RAM 区域时或者执行不进行数据存取的指令时的时钟数。
 2. 这是存取内部高速 RAM 以外区域时的时钟数。
 3. “r=A” 除外

- 备注 1. 1 个指令时钟是指 1 个通过处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU})。
 2. 这是在内部 ROM 区域执行程序时的时钟数。

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	2	—	$r \leftarrow r+1$	x	x	
		saddr	2	4	6	$(saddr) \leftarrow (saddr)+1$	x	x	
	DEC	r	1	2	—	$r \leftarrow r-1$	x	x	
		saddr	2	4	6	$(saddr) \leftarrow (saddr)-1$	x	x	
	INCW	rp	1	4	—	$rp \leftarrow rp+1$			
DECW	rp	1	4	—	$rp \leftarrow rp-1$				
循环	ROR	A, 1	1	2	—	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$ 次			x
	ROL	A, 1	1	2	—	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$ 次			x
	RORC	A, 1	1	2	—	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$ 次			x
	ROLC	A, 1	1	2	—	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$ 次			x
	ROR4	[HL]	2	10	12	$A_{3-0} \leftarrow (HL)_{3-0}, (HL)_{7-4} \leftarrow A_{3-0}, (HL)_{3-0} \leftarrow (HL)_{7-4}$			
	ROL4	[HL]	2	10	12	$A_{3-0} \leftarrow (HL)_{7-4}, (HL)_{3-0} \leftarrow A_{3-0}, (HL)_{7-4} \leftarrow (HL)_{3-0}$			
BCD 校正	ADJBA		2	4	—	Decimal Adjust Accumulator after Addition	x	x	x
	ADJBS		2	4	—	Decimal Adjust Accumulator after Subtract	x	x	x
位 操 作	MOV1	CY, saddr.bit	3	6	7	$CY \leftarrow (saddr.bit)$			x
		CY, sfr.bit	3	—	7	$CY \leftarrow sfr.bit$			x
		CY, A.bit	2	4	—	$CY \leftarrow A.bit$			x
		CY, PSW.bit	3	—	7	$CY \leftarrow PSW.bit$			x
		CY,[HL].bit	2	6	7	$CY \leftarrow (HL).bit$			x
		saddr.bit, CY	3	6	8	$(saddr.bit) \leftarrow CY$			
		sfr.bit, CY	3	—	8	$sfr.bit \leftarrow CY$			
		A.bit, CY	2	4	—	$A.bit \leftarrow CY$			
		PSW.bit, CY	3	—	8	$PSW.bit \leftarrow CY$	x	x	
	[HL].bit, CY	2	6	8	$(HL).bit \leftarrow CY$				
	AND1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \wedge (saddr.bit)$			x
		CY, sfr.bit	3	—	7	$CY \leftarrow CY \wedge sfr.bit$			x
		CY, A.bit	2	4	—	$CY \leftarrow CY \wedge A.bit$			x
		CY, PSW.bit	3	—	7	$CY \leftarrow CY \wedge PSW.bit$			x
		CY,[HL].bit	2	6	7	$CY \leftarrow CY \wedge (HL).bit$			x
	OR1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \vee (saddr.bit)$			x
		CY, sfr.bit	3	—	7	$CY \leftarrow CY \vee sfr.bit$			x
		CY, A.bit	2	4	—	$CY \leftarrow CY \vee A.bit$			x
		CY, PSW.bit	3	—	7	$CY \leftarrow CY \vee PSW.bit$			x
		CY,[HL].bit	2	6	7	$CY \leftarrow CY \vee (HL).bit$			x

注 1. 这是存取内部高速 RAM 区域时或者执行不进行数据存取的指令时的时钟数。

2. 这是存取内部高速 RAM 以外区域时的时钟数。

备注 1. 1 个指令时钟是指 1 个通过处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU})。

2. 这是在内部 ROM 区域执行程序时的时钟数。

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, saddr.bit	3	6	7	$CY \leftarrow CY \nabla (\text{saddr.bit})$			x
		CY, sfr.bit	3	—	7	$CY \leftarrow CY \nabla \text{sfr.bit}$			x
		CY, A.bit	2	4	—	$CY \leftarrow CY \nabla A.\text{bit}$			x
		CY, PSW.bit	3	—	7	$CY \leftarrow CY \nabla \text{PSW.bit}$			x
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \nabla (\text{HL}).\text{bit}$			x
	SET1	saddr.bit	2	4	6	$(\text{saddr.bit}) \leftarrow 1$			
		sfr.bit	3	—	8	$\text{sfr.bit} \leftarrow 1$			
		A.bit	2	4	—	$A.\text{bit} \leftarrow 1$			
		PSW.bit	2	—	6	$\text{PSW.bit} \leftarrow 1$	x	x	x
		[HL].bit	2	6	8	$(\text{HL}).\text{bit} \leftarrow 1$			
	CLR1	saddr.bit	2	4	6	$(\text{saddr.bit}) \leftarrow 0$			
		sfr.bit	3	—	8	$\text{sfr.bit} \leftarrow 0$			
		A.bit	2	4	—	$A.\text{bit} \leftarrow 0$			
		PSW.bit	2	—	6	$\text{PSW.bit} \leftarrow 0$	x	x	x
		[HL].bit	2	6	8	$(\text{HL}).\text{bit} \leftarrow 0$			
	SET1	CY	1	2	—	$CY \leftarrow 1$			1
	CLR1	CY	1	2	—	$CY \leftarrow 0$			0
	NOT1	CY	1	2	—	$CY \leftarrow \overline{CY}$			x
	调用返回	CALL	!addr16	3	7	—	$(\text{SP}-1) \leftarrow (\text{PC}+3)_H, (\text{SP}-2) \leftarrow (\text{PC}+3)_L,$ $\text{PC} \leftarrow \text{addr16}, \text{SP} \leftarrow \text{SP}-2$		
CALLF		!addr11	2	5	—	$(\text{SP}-1) \leftarrow (\text{PC}+2)_H, (\text{SP}-2) \leftarrow (\text{PC}+2)_L,$ $\text{PC}_{15-11} \leftarrow 00001, \text{PC}_{10-0} \leftarrow \text{addr11},$ $\text{SP} \leftarrow \text{SP}-2$			
CALLT		[addr5]	1	6	—	$(\text{SP}-1) \leftarrow (\text{PC}+1)_H, (\text{SP}-2) \leftarrow (\text{PC}+1)_L,$ $\text{PC}_H \leftarrow (\text{addr5}+1), \text{PC}_L \leftarrow (\text{addr5}),$ $\text{SP} \leftarrow \text{SP}-2$			
BRK			1	6	—	$(\text{SP}-1) \leftarrow \text{PSW}, (\text{SP}-2) \leftarrow (\text{PC}+1)_H,$ $(\text{SP}-3) \leftarrow (\text{PC}+1)_L, \text{PC}_H \leftarrow (003FH),$ $\text{PC}_L \leftarrow (003EH), \text{SP} \leftarrow \text{SP}-3, \text{IE} \leftarrow 0$			
RET			1	6	—	$\text{PC}_H \leftarrow (\text{SP}+1), \text{PC}_L \leftarrow (\text{SP}), \text{SP} \leftarrow \text{SP}+2$			
RETI			1	6	—	$\text{PC}_H \leftarrow (\text{SP}+1), \text{PC}_L \leftarrow (\text{SP}),$ $\text{PSW} \leftarrow (\text{SP}+2), \text{SP} \leftarrow \text{SP}+3$	R	R	R
RETB			1	6	—	$\text{PC}_H \leftarrow (\text{SP}+1), \text{PC}_L \leftarrow (\text{SP}),$ $\text{PSW} \leftarrow (\text{SP}+2), \text{SP} \leftarrow \text{SP}+3$	R	R	R

注 1. 这是存取内部高速 RAM 区域时或者执行不进行数据存取的指令时的时钟数。

2. 这是存取内部高速 RAM 以外区域时的时钟数。

备注 1. 1 个指令时钟是指 1 个通过处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU})。

2. 这是在内部 ROM 区域执行程序时的时钟数。

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	1	2	—	$(SP-1) \leftarrow PSW, SP \leftarrow SP-1$			
		rp	1	4	—	$(SP-1) \leftarrow rp_H, (SP-2) \leftarrow rp_L, SP \leftarrow SP-2$			
	POP	PSW	1	2	—	$PSW \leftarrow (SP), SP \leftarrow SP+1$	R	R	R
		rp	1	4	—	$rp_H \leftarrow (SP+1), rp_L \leftarrow (SP), SP \leftarrow SP+2$			
	MOVW	SP, #word	4	—	10	$SP \leftarrow word$			
		SP, AX	2	—	8	$SP \leftarrow AX$			
AX, SP		2	—	8	$AX \leftarrow SP$				
无条件转移	BR	!addr16	3	6	—	$PC \leftarrow addr16$			
		\$addr16	2	6	—	$PC \leftarrow PC+2+jdisp8$			
		AX	2	8	—	$PC_H \leftarrow A, PC_L \leftarrow X$			
带条件的转移	BC	\$addr16	2	6	—	$PC \leftarrow PC+2+jdisp8$ if CY = 1			
		\$addr16	2	6	—	$PC \leftarrow PC+2+jdisp8$ if CY = 0			
	BZ	\$addr16	2	6	—	$PC \leftarrow PC+2+jdisp8$ if Z = 1			
		\$addr16	2	6	—	$PC \leftarrow PC+2+jdisp8$ if Z = 0			
	BT	saddr.bit, \$addr16	3	8	9	$PC \leftarrow PC+3+jdisp8$ if (saddr.bit) = 1			
		sfr.bit, \$addr16	4	—	11	$PC \leftarrow PC+4+jdisp8$ if sfr.bit = 1			
		A.bit, \$addr16	3	8	—	$PC \leftarrow PC+3+jdisp8$ if A.bit = 1			
		PSW.bit, \$addr16	3	—	9	$PC \leftarrow PC+3+jdisp8$ if PSW.bit = 1			
		[HL].bit, \$addr16	3	10	11	$PC \leftarrow PC+3+jdisp8$ if (HL).bit = 1			
	BF	saddr.bit, \$addr16	4	10	11	$PC \leftarrow PC+4+jdisp8$ if (saddr.bit) = 0			
		sfr.bit, \$addr16	4	—	11	$PC \leftarrow PC+4+jdisp8$ if sfr.bit = 0			
		A.bit, \$addr16	3	8	—	$PC \leftarrow PC+3+jdisp8$ if A.bit = 0			
		PSW.bit, \$addr16	4	—	11	$PC \leftarrow PC+4+jdisp8$ if PSW.bit = 0			
		[HL].bit, \$addr16	3	10	11	$PC \leftarrow PC+3+jdisp8$ if (HL).bit = 0			
	BTCLR	saddr.bit, \$addr16	4	10	12	$PC \leftarrow PC+4+jdisp8$ if (saddr.bit) = 1 then reset (saddr.bit)			
		sfr.bit, \$addr16	4	—	12	$PC \leftarrow PC+4+jdisp8$ if sfr.bit = 1 then reset sfr.bit			
		A.bit, \$addr16	3	8	—	$PC \leftarrow PC+3+jdisp8$ if A.bit = 1 then reset A.bit			
		PSW.bit, \$addr16	4	—	12	$PC \leftarrow PC+4+jdisp8$ if PSW.bit = 1 then reset PSW.bit	x	x	x
		[HL].bit, \$addr16	3	10	12	$PC \leftarrow PC+3+jdisp8$ if (HL).bit = 1 then reset(HL).bit			

注 1. 这是存取内部高速 RAM 区域时或者执行不进行数据存取的指令时的时钟数。

2. 这是存取内部高速 RAM 以外区域时的时钟数。

备注 1. 1 个指令时钟是指 1 个通过处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU})。

2. 这是在内部 ROM 区域执行程序时的时钟数。

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
带条件的转移	DBNZ	B, \$addr16	2	6	—	B ← B-1, then PC ← PC+2+jdisp8 if B≠0			
		C, \$addr16	2	6	—	C ← C-1, then PC ← PC+2+jdisp8 if C≠0			
		saddr, \$addr16	3	8	10	(saddr) ← (saddr)-1, then PC ← PC+3+jdisp8 if(saddr)≠0			
CPU 控制	SEL	RBn	2	4	—	RBS1, 0 ← n			
	NOP		1	2	—	No Operation			
	EI		2	—	6	IE ← 1(Enable Interrupt)			
	DI		2	—	6	IE ← 0(Disable Interrupt)			
	HALT		2	6	—	Set HALT Mode			
	STOP		2	6	—	Set STOP Mode			

- 注 1. 这是存取内部高速 RAM 区域时或者执行不进行数据存储取的指令时的时钟数。
 2. 这是存取内部高速 RAM 以外区域时的时钟数。

- 备注 1. 1 个指令时钟是指 1 个通过处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU})。
 2. 这是在内部 ROM 区域执行程序时的时钟数。

24.3 各寻址的指令一览

(1) 8 位指令

MOV、XCH、ADD、ADDC、SUB、SUBC、AND、OR、XOR、CMP、MULU、DIVUW、INC、DEC、ROR、ROL、RORC、ROLC、ROR4、ROL4、PUSH、POP、DBNZ

第 2 操作数 第 1 操作数	#byte	A	r 注	sfr	saddr	laddr16	PSW	[DE]	[HL]	[HL+byte] [HL+B] [HL+C]	\$addr16	1	无
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B,C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL+byte] [HL+B] [HL+C]		MOV											
X													MULU
C													DIVUW

注 “r=A” 除外

(2) 16 位指令

MOVW、XCHW、ADDW、SUBW、CMPW、PUSH、POP、INCW、DECW

第 2 操作数 \ 第 1 操作数	#word	AX	rp 注	sfrp	saddrp	!addr16	SP	无
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW 注						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

注 只在 rp=BC、DE 或者 HL 时

(3) 位操作指令

MOV1、AND1、OR1、XOR1、SET1、CLR1、NOT1、BT、BF、BTCLR

第 2 操作数 \ 第 1 操作数	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	\$addr16	无
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) 调用指令 / 转移指令

CALL、CALLF、CALLT、BR、BC、BNC、BZ、BNZ、BT、BF、BTCLR、DBNZ

第 2 操作数 第 1 操作数	AX	!addr16	!addr11	[addr5]	\$addr16
基本指令	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
复合指令					BT BF BTCLR DBNZ

(5) 其他指令

ADJBA、ADJBS、BRK、RET、RETI、RETB、SEL、NOP、EI、DI、HALT、STOP

第 25 章 电特性 (目标)

注意 该电特性为目标值，在对设备评估后有可能进行更改。

绝对最大额定值 ($T_A=25^\circ\text{C}$) (1/2)

项目	符号	条件	额定值	单位
电源电源	V_{DD}		$-0.5 \sim +6.5$	V
	V_{SS}		$-0.5 \sim +0.3$	V
REGC 引脚输入电压	V_{IREGC}		$-0.5 \sim +3.6$ 并且 $-0.5 \sim V_{DD}$	V
输入电压	V_{I1}	P00、P01、P10 ~ P17、P20 ~ P23、P30 ~ P33、P40、P41、P70、P71、P120 ~ P122、X1、X2、RESET、FLMD0	$-0.3 \sim V_{DD}+0.3$ 注	V
	V_{I2}	P60、P61 (N 沟道漏极开路)	$-0.3 \sim +6.5$	V
输出电压	V_O		$-0.3 \sim V_{DD}+0.3$ 注	V
模拟输入电压	V_{AN}	ANI0 ~ ANI3	$-0.3 \sim V_{DD}+0.3$ 注	V

注 小于等于 6.5V。

注意 各项目中只要有一项即使一瞬间超出绝对最大额定值，就有可能对产品的品质造成损害。即绝对最大额定值是有可能对产品造成物理损害的额定值。必须在不出超额定值的状态下使用产品。

备注 在没有特别指定时，复用引脚的特性与端口引脚的特性相同。

绝对最大额定值 ($T_A=25^{\circ}\text{C}$) (2/2)

项目	符号	条件		额定值	单位
高电平输出电流	I_{OH1}	1 个引脚	P00、P01、P10 ~ P17、P30 ~ P33、P40、P41、P70、P71、P120	-10	mA
		引脚总计 -80mA	P00、P01、P40、P41、P120	-25	mA
			P10 ~ P17、P30 ~ P33、P70、P71	-55	
	I_{OH2}	1 个引脚	P20 ~ P23	-0.5	mA
		引脚总计		-2	mA
	I_{OH3}	1 个引脚	P121、P122	-1	mA
		引脚总计		-2	mA
低电平输出电流	I_{OL1}	1 个引脚	P00、P01、P10 ~ P17、P30 ~ P33、P40、P41、P60、P61、P70、P71、P120	30	mA
		引脚总计 200mA	P00、P01、P40、P41、P120	60	mA
			P10 ~ P17、P30 ~ P33、P60、P61、P70、P71	140	
	I_{OL2}	1 个引脚	P20 ~ P23	1	mA
		引脚总计		4	mA
	I_{OL3}	1 个引脚	P121、P122	4	mA
		引脚总计		8	mA
工作环境温度	T_A	通常运行时		-40 ~ +85	$^{\circ}\text{C}$
		闪存编程时			
保存温度	T_{stg}			-65 ~ +150	$^{\circ}\text{C}$

注意 1. 各项目中只要有一项即使一瞬间超出绝对最大额定值, 就有可能对产品的品质造成损害。绝对最大额定值是有可能对产品造成物理损害的额定值。必须在不出额定值的状态下使用产品。

2. 每个引脚可流过的电流值必须即满足每个引脚的电流值也满足引脚总计的电流值。

备注 在没有特别指定时, 复用引脚的特性与端口引脚的特性相同。

X1 振荡电路特性

(T_A=-40 ~ +85°C、4.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

项目	推荐电路	项目	条件	MIN.	TYP.	MAX.	单位
陶瓷谐振器、 晶体谐振器		X1 时钟振荡频率 (f _X) 注 1		1.0 注 2		10.0	MHz

注 1. 只表示振荡电路的特性。指令执行时间请参照“AC 特性”。

2. 如果在板上编程时使用 UART6, 为 2.0MHz(MIN.)。

注意 1. 在使用 X1 振荡电路时, 为了避免受布线电容等的影响, 必须如下进行上图中虚线部分的布线。

- 布线尽可能短
 - 不能与其他信号线交叉
 - 不能靠近有波动大的大电流流过的布线
 - 振荡电路的电容器接地点总是与 V_{SS} 的电位相同
 - 有大电流流过的接地图案不能接地
 - 不能从振荡电路取信号
2. 在复位解除后, 为了通过高速内部振荡时钟启动 CPU, 用户必须通过振荡稳定时间计数器状态寄存器 (OSTC) 确认 X1 时钟的振荡稳定时间。通过使用的谐振器对振荡稳定时间进行充分的评估后, 再决定 OSTC 寄存器和振荡稳定时间选择寄存器 (OSTS) 的振荡稳定时间。

内部振荡电路特性

(T_A=-40 ~ +85°C、4.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

谐振器	项目	条件	MIN.	TYP.	MAX.	单位
8MHz 内部振荡器	高速内部振荡时钟 频率 (f _{RH}) 注	RSTS=1	7.6	8.0	8.4	MHz
		RSTS=0	2.48	5.6	9.86	MHz
240kHz 内部振荡器	低速内部振荡时钟 频率 (f _{RL})		216	240	264	MHz

注 只表示振荡电路的特性。指令执行时间请参照“AC 特性”。

备注 RSTS: 内部振荡模式寄存器 (RCM) 的 bit7

DC 特性 (1/4)

(T_A=-40 ~ +85°C、4.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
高电平输出电流注 1	I _{OH1}	P00、P01、P10 ~ P17、P30 ~ P33、P40、P41、P70、P71、P120 1 个引脚			-3.0	mA
		P00、P01、P40、P41、P120 总计注 3			-20.0	mA
		P10 ~ P17、P30 ~ P33、P70、P71 总计注 3			-30.0	mA
		上述全部引脚 总计注 3			-50.0	mA
	I _{OH2}	P20 ~ P23 1 个引脚			-0.1	mA
		P121、P122 1 个引脚			-0.1	mA
低电平输出电流注 2	I _{OL1}	P00、P01、P10 ~ P17、P30 ~ P33、P40、P41、P70、P71、P120 1 个引脚			8.5	mA
		P60、P61 1 个引脚			15.0	mA
		P00、P01、P40、P41、P120 总计注 3			20.0	mA
		P10 ~ P17、P30 ~ P33、P60、P61、P70、P71 总计注 3			45.0	mA
		上述全部引脚 总计注 3			65.0	mA
	I _{OL2}	P20 ~ P23 1 个引脚			0.4	mA
		P121、P122 1 个引脚			0.4	mA

- 注 1. 这是即使电流从 V_{DD} 流向输出引脚，也能保证设备运行的电流值。
2. 这是即使电流从输出引脚流入 GND，也能保证设备运行的电流值。
3. 这是在占空比为 70% 的条件 (假设某一定的时间为 t、输出电流时间为 0.7×t、未输出电流时间为 0.3×t 的情况) 下的规格。占空比不为 70% 的引脚总计的输出电流可通过以下计算式算出。
- I_{OH} 的占空比为 n% 的情况: 引脚总计的输出电流 = (I_{OH} × 0.7) / (n × 0.01)
- < 计算示例 > 当占空比 = 50%、I_{OH} = -20.0mA 时
- 引脚总计的输出电流 = (-20.0 × 0.7) / (50 × 0.01) = -28.0mA
- 但是，每个引脚的输出电流不会因占空比而变。另外，不能输出超出绝对最大额定值的电流。

备注 在没有特别指定的情况下，复用引脚的特性与端口引脚的特性相同。

DC 特性 (2/4)

(T_A=-40 ~ +85°C、4.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
高电平输入电压	V _{IH1}	P40、P41、P121、P122	0.7V _{DD}		V _{DD}	V
	V _{IH2}	P00、P01、P10 ~ P17、P30 ~ P33、P70、P71、P120、 $\overline{\text{RESET}}$	0.8V _{DD}		V _{DD}	V
	V _{IH3}	P20 ~ P23	0.7V _{DD}		V _{DD}	V
	V _{IH4}	P60、P61	0.7V _{DD}		6.0	V
低电平输入电压	V _{IL1}	P40、P41、P60、P61、P121、P122	0		0.3V _{DD}	V
	V _{IL2}	P00、P01、P10 ~ P17、P30 ~ P33、P70、P71、P120、 $\overline{\text{RESET}}$	0		0.2V _{DD}	V
	V _{IL3}	P20 ~ P23	0		0.3V _{DD}	V
高电平输出电压	V _{OH1}	P00、P01、P10 ~ P17、P30 ~ P33、P40、P41、P70、P71、P120	I _{OH1} =-3.0mA		V _{DD} -0.7	V
	V _{OH2}	P20 ~ P23 P121、P122	I _{OH2} =-100μA		V _{DD} -0.5	V
低电平输出电压	V _{OL1}	P00、P01、P10 ~ P17、P30 ~ P33、P40、P41、P70、P71、P120	I _{OL1} =8.5mA		0.7	V
	V _{OL2}	P20 ~ P23 P121、P122	I _{OL2} =0.4mA		0.4	V
	V _{OL3}	P60、P61	I _{OL1} =15.0mA I _{OL1} =5.0mA		2.0 0.4	V V
高电平输入漏电流	I _{LIH1}	P00、P01、P10 ~ P17、P30 ~ P33、P40、P41、P60、P61、P70、P71、P120、FLMD0、 $\overline{\text{RESET}}$	V _I =V _{DD}		1	μA
	I _{LIH2}	P20、P23	V _I =V _{DD}		1	μA
	I _{LIH3}	P121、P122 (X1、X2)	V _I =V _{DD}	I/O 端口模式 OSC 模式	1 20	μA μA
低电平输入漏电流	I _{LIL1}	P00、P01、P10 ~ P17、P30 ~ P33、P40、P41、P60、P61、P70、P71、P120、FLMD0、 $\overline{\text{RESET}}$	V _I =V _{SS}		-1	μA
	I _{LIL2}	P20 ~ P23	V _I =V _{SS}		-1	μA
	I _{LIL3}	P121、P122 (X1、X2)	V _I =V _{SS}	I/O 端口模式 OSC 模式	-1 -20	μA μA
上拉电阻值	R _U	V _I =V _{SS}	10	20	100	kΩ
FLMD0 电源电压	V _{IL}	通常运行时	0		0.2V _{DD}	V
	V _{IH}		0.8V _{DD}		V _{DD}	V

备注 在没有特别指定的情况下，复用引脚的特性与端口引脚的特性相同。

DC 特性 (3/4)

(T_A=-40 ~ +85°C、4.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

项目	符号	条件		MIN.	TYP.	MAX.	单位	
电源电流 ^{注 1}	I _{DD1}	运行模式	f _{XH} =10MHz、 V _{DD} =5.0V ^{注 2}	方波输入		1.6	2.8	mA
				谐振器连接		2.3	3.9	mA
			f _{RH} =8MHz、V _{DD} =5.0V ^{注 3}		1.4	2.5	mA	
	I _{DD2}	HALT 模式	f _{XH} =10MHz、 V _{DD} =5.0V ^{注 2}	方波输入		0.4	1.3	mA
				谐振器连接		1.0	2.4	mA
			f _{RH} =8MHz、V _{DD} =5.0V ^{注 3}		0.4	1.2	mA	
I _{DD3} ^{注 4}	STOP 模式	T _A =+70 ~ +85°C			—	20	μA	
		T _A =-40 ~ +70°C			1	10	μA	

- 注 1. 这是流入内部电源 (V_{DD}) 的总电流。包括将输入引脚固定为 V_{DD} 或者 V_{SS} 状态下的输入漏电流以及外围工作电流。但是, 不包括端口的上拉电阻和输出电流。
2. 不包括 8MHz 内部振荡器和 240kHz 内部振荡器的工作电流, 以及流入 A/D 转换器、看门狗定时器、LVI 电路的电流。
3. 不包括 X1 振荡电路和 240kHz 内部振荡器的工作电流, 以及流入 A/D 转换器、看门狗定时器、LVI 电路的电流。
4. 不包括 240kHz 内部振荡器的工作电流以及流入 A/D 转换器、看门狗定时器、LVI 电路的电流。

备注 1. f_{XH}: 高速系统时钟频率 (X1 时钟振荡频率数或者外部主系统时钟频率数)2. f_{RH}: 高速内部振荡时钟频率

DC 特性 (4/4)

(T_A=-40 ~ +85°C、4.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
A/D 转换器工作电流	I _{ADC} ^{注 1}	ADCS=1		0.86	1.9	mA
看门狗定时器工作电流	I _{WDT} ^{注 2}	240kHz 低速内部振荡时钟运行时		5	10	μA
LVI 工作电流	I _{LVI} ^{注 3}			9	18	μA

- 注 1. 这是只流入 A/D 转换器的电流。在运行模式或者 HALT 模式中 A/D 转换器运行期间, R7F0C011B、R7F0C012B、R7F0C013B 的电流值为 I_{DD1} 或者 I_{DD2} 与 I_{ADC} 相加后的值。
2. 这只是流入看门狗定时器的电流 (包括 240kHz 内部振荡器的工作电流)。在看门狗定时器运行期间, R7F0C011B、R7F0C012B、R7F0C013B 的电流值为 I_{DD1}、I_{DD2} 或者 I_{DD3} 与 I_{WDT} 相加后的值。
3. 这只是流入 LVI 电路的电流。在 LVI 电路运行期间, R7F0C011B、R7F0C012B、R7F0C013B 的电流值为 I_{DD1}、I_{DD2} 或者 I_{DD3} 与 I_{LVI} 相加后的值。

备注 1. f_{XH}: 高速系统时钟频率 (X1 时钟振荡频率数或者外部主系统时钟频率数)2. f_{RH}: 高速内部振荡时钟频率

AC 特性

(1) 基本运行

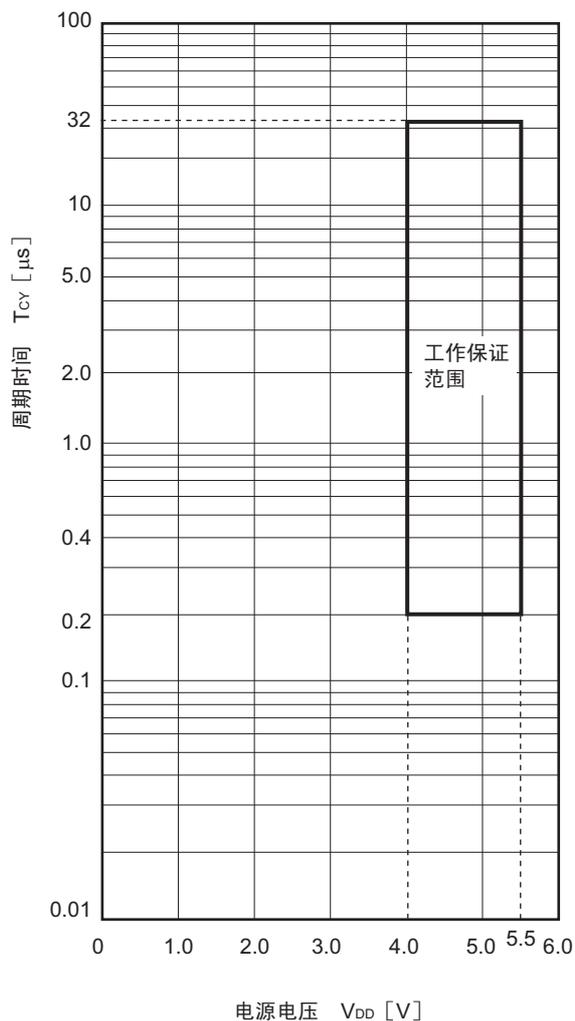
(T_A=-40 ~ +85°C、4.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
指令周期 (最短指令执行时间)	T _{CY}	主系统时钟 (f _{XP}) 运行	0.2		32	μs
外围硬件时钟频率	f _{PRS}	f _{PRS} =f _{XH} (XSEL=1)			10	MHz
		f _{PRS} =f _{RH} (XSEL=0)	7.6		8.4	MHz
外部主系统时钟频率	f _{EXCLK}		1.0 注 1		10.0	MHz
外部主系统时钟输入高电平和输入低电平的宽度	t _{EXCLKH} 、 t _{EXCLKL}		48			ns
TI000、TI010 输入高电平和输入低电平的宽度	t _{TIH0} 、 t _{TIL0}		2/f _{sam} +0.1 注 2			μs
TI50、TI51 输入频率	f _{TI5}				10	MHz
TI50、TI51 输入高电平和输入低电平的宽度	t _{TIH5} 、 t _{TIL5}		50			ns
中断输入高电平和低电平的宽度	t _{INTH} 、 t _{INTL}		1			μs
RESET 低电平宽度	t _{RSL}		10			μs

注 1. 如果在板上编程时使用 UART6, 为 2.0MHz(MIN.)。

2. 通过预分频器模式寄存器 00 (PRM00) 的 bit0 和 bit1 (PRM000、PRM001), 可选择 f_{sam} 为 f_{PRS}、f_{PRS}/4、f_{PRS}/256 或者 f_{PRS}、f_{PRS}/16、f_{PRS}/64。但是, 在选择 TI000 有效边沿作为计数时钟时, f_{sam} 为 f_{PRS}。

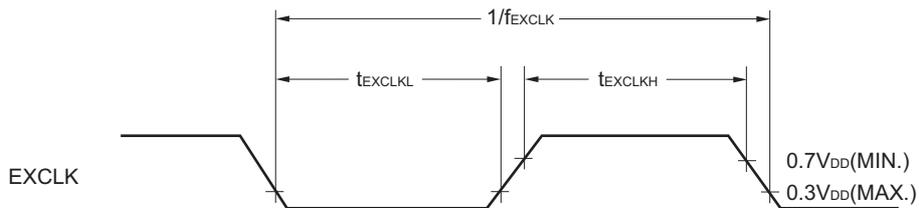
T_{CY} vs V_{DD} (主系统时钟运行)



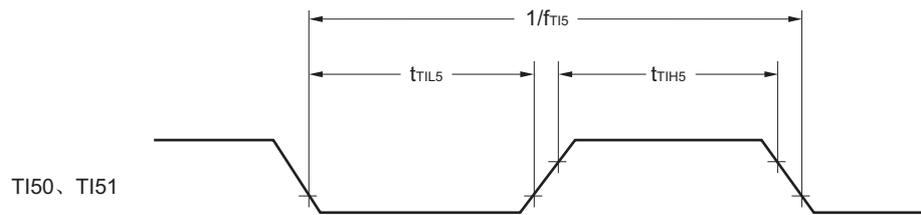
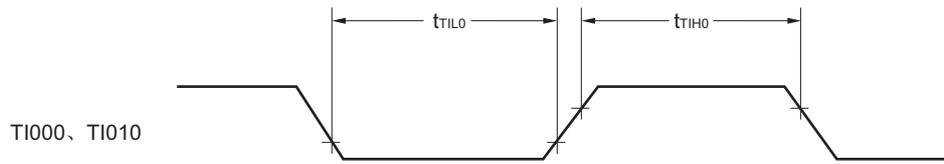
AC 时序测量点



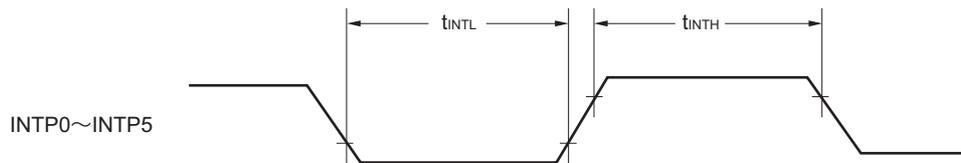
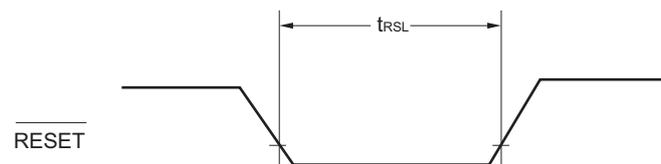
外部主系统时钟时序



TI 时序



中断请求输入时序

 $\overline{\text{RESET}}$ 输入时序

(2) 串行接口

(T_A=−40 ~ +85°C、4.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

(a) UART6 (专用波特率发生器输出)

项目	符号	条件	MIN.	TYP.	MAX.	单位
传送速率					625	kbps

(b) UART0 (专用波特率发生器输出)

项目	符号	条件	MIN.	TYP.	MAX.	单位
传送速率					625	kbps

(c) IIC0

项目	符号	条件	标准模式		高速模式		单位	
			MIN.	MAX.	MIN.	MAX.		
SCL0 时钟频率	f _{SCL}		0	100	0	400	kHz	
重新开始条件的准备时间	t _{SU: STA}		4.7	—	0.6	—	μs	
保持时间注 1	t _{HD: STA}		4.0	—	0.6	—	μs	
SCL0="L" 电平的保持时间	t _{LOW}	内部时钟运行	4.7	—	1.3	—	μs	
SCL0="H" 电平的保持时间	t _{HIGH}		4.0	—	0.6	—	μs	
数据准备时间 (接收时)	t _{SU: DAT}		250	—	100	—	ns	
数据保持时间 (发送时) 注 2	t _{HD: DAT}	选择 f _W =f _{XH} /2 ^N 时注 3	DFC0=0	0	3.45	0	0.9 注 4	μs
							1.00 注 5	
		DFC0=1		—	—	0	0.9 注 6	μs
							1.125 注 7	
		选择 f _W =f _{RH} /2 ^N 时注 3	DFC0=0	0	3.45	0	1.05	μs
			DFC0=1	—	—	0	1.184	
停止条件的准备时间	t _{SU: STO}		4.0	—	0.6	—	μs	
总线释放时间	t _{BUF}		4.7	—	1.3	—	μs	

注 1. 开始 / 重新开始条件下, 在此间后生成第一个时钟脉冲。

2. t_{HD: DAT} 的最大值 (MAX.) 为通常传送时的数值。在 $\overline{\text{ACK}}$ (应答) 时序期间, 需要等待。3. f_W 表示 IICL0 寄存器选择的 IIC0 的传送时钟。4. 选择 f_W ≥ 4.4MHz 时5. 选择 f_W < 4.4MHz 时6. 选择 f_W ≥ 5.0MHz 时7. 选择 f_W < 5.0MHz 时

(d) CSI10 (主模式、 $\overline{\text{SCK10}}$... 内部时钟输出)

项目	符号	条件	MIN.	TYP.	MAX.	单位
$\overline{\text{SCK10}}$ 周期时间	t_{KCY1}		200			ns
$\overline{\text{SCK10}}$ 高 / 低电平宽度	t_{KH1} 、 t_{KL1}		$t_{\text{KCY1}}/2-15$ 注 1			ns
SI10 准备时间 (对 $\overline{\text{SCK10}}\uparrow$)	t_{SIK1}		55			ns
SI10 保持时间 (对 $\overline{\text{SCK10}}\uparrow$)	t_{KSI1}		30			ns
$\overline{\text{SCK10}}\downarrow\rightarrow\text{SO10}$ 输出延迟时间	t_{KSO1}	C=50pF 注 2			40	ns

注 1. 这是使用高速系统时钟 (f_{XH}) 时的数值。

2. C 是 $\overline{\text{SCK10}}$ 、SO10 输出线的负载电容。

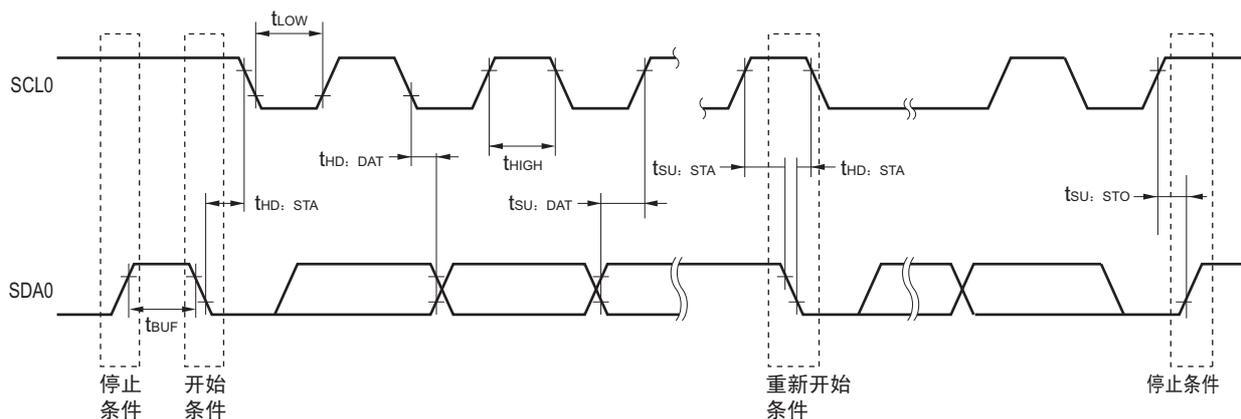
(e) CSI10 (从属模式、 $\overline{\text{SCK10}}$... 外部时钟输入)

项目	符号	条件	MIN.	TYP.	MAX.	单位
$\overline{\text{SCK10}}$ 周期时间	t_{KCY2}		400			ns
$\overline{\text{SCK10}}$ 高 / 低电平宽度	t_{KH2} 、 t_{KL2}		$t_{\text{KCY2}}/2$			ns
SI10 准备时间 (对 $\overline{\text{SCK10}}\uparrow$)	t_{SIK2}		80			ns
SI10 保持时间 (对 $\overline{\text{SCK10}}\uparrow$)	t_{KSI2}		50			ns
$\overline{\text{SCK10}}\downarrow\rightarrow\text{SO10}$ 输出延迟时间	t_{KSO2}	C=50pF 注			120	ns

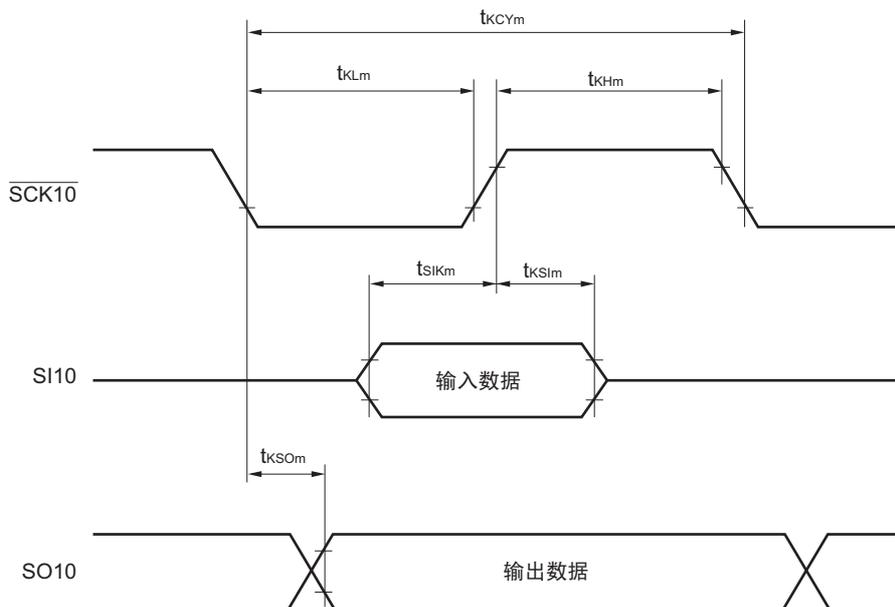
注 C 是 SO10 输出线的负载电容。

串行传送时序

IIC0:



CSI10:



备注 m=1、2

A/D 转换器特性

(T_A = -40 ~ +85°C, 4.0V ≤ V_{DD} ≤ 5.5V, V_{SS} = 0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	R _{ES}		10			bit
总误差注 1、注 2	A _{INL}				±0.4	%FSR
转换时间	t _{CONV}		6.1		66.6	μs
模拟输入电压	V _{AIN}		V _{SS}		V _{DD}	V

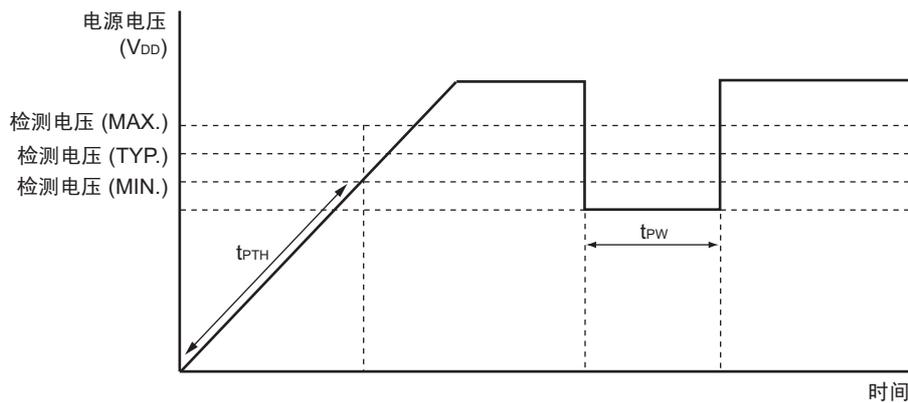
- 注 1. 包括量化误差 (±1/2LSB)。
 2. 通过对应满刻度值的比率 (%FSR) 表示。

1.59V POC 电路特性

(T_A=-40 ~ +85°C、V_{SS}=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
检测电压	V _{POC}		1.44	1.59	1.74	V
电源电压上升斜率	t _{PTH}	V _{DD} : 0V → V _{POC} 的变化斜率	0.5			V/ms
最小脉宽	t _{PW}		200			μs

1.59V POC 电路时序



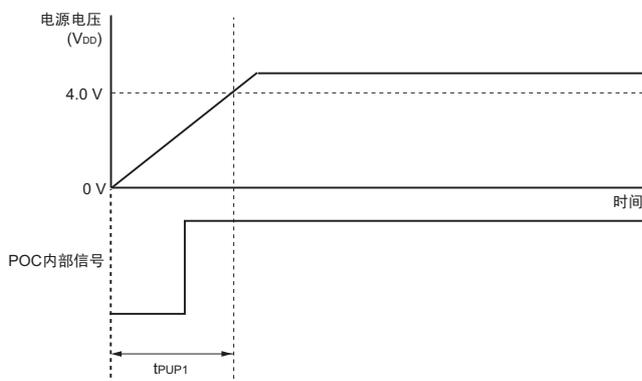
电源电压上升时间

(T_A=-40 ~ +85°C、V_{SS}=0V)

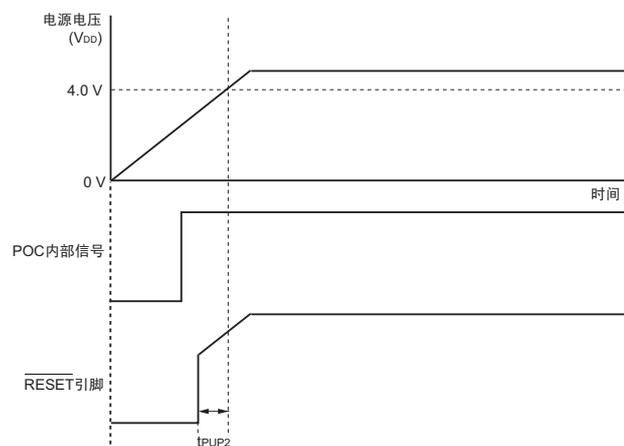
项目	符号	条件	MIN.	TYP.	MAX.	单位
上升到 4.0V(V _{DD} (MIN.)) 的最长时间 (V _{DD} : 0V→4.0V)	t _{PUP1}	POCMODE (选项字节) =0、 未使用 RESET 输入时			3.6	ms
上升到 4.0V(V _{DD} (MIN.)) 的最长时间 (RESET 输入解除 →V _{DD} : 4.0V)	t _{PUP2}	POCMODE (选项字节) =0、 使用 RESET 输入时			1.9	ms

电源电压上升时序

- 未使用 RESET 引脚输入时



- 使用 RESET 引脚输入时



2.7V POC 电路特性

(T_A=-40 ~ +85°C、V_{SS}=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
接通电源电压时的检测电压	V _{DDPOC}	POCMODE (选项字节) =1	2.50	2.70	2.90	V

备注 POC 电路的运行根据 POCMODE (选项字节) 的设置如下变化。

选项字节的设置	POC 模式	运行
POCMODE=0	1.59V 模式运行	从接通电源到达到 V _{POC} =1.59V(TYP.) 前为复位状态, 一旦大于 V _{POC} 就解除复位。此后, 和接通电源时相同, 通过 V _{POC} 进行 POC 检测。 当 POCMODE=0 时, 必须在 t _{PUP1} 或者 t _{PUP2} 的时间内上升电源电压。
POCMODE=1	2.7V/1.59V 模式运行	从接通电源到达到 V _{DDPOC} =2.7V(TYP.) 前为复位状态, 一旦大于 V _{DDPOC} 就解除复位。此后, 不通过 V _{DDPOC} 进行 POC 检测, 通过 V _{POC} =1.59V(TYP.) 进行 POC 检测。 如果从接通电源到达到 4.0V 前的电压上升比 t _{PTH} 缓慢, 建议使用 2.7V/1.59V POC 模式。

LVI 电路特性

(T_A=-40 ~ +85°C、V_{POC} ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

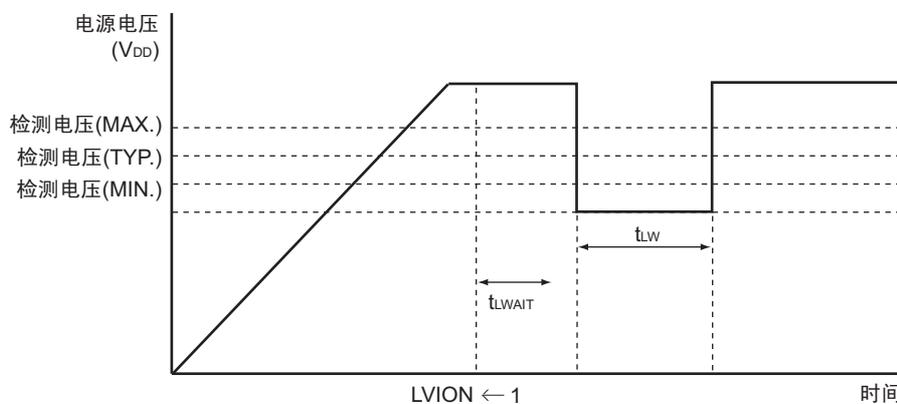
项目	符号	条件	MIN.	TYP.	MAX.	单位
检测电压	检测电压电平	V _{LVI0}	4.14	4.24	4.34	V
		V _{LVI1}	3.99	4.09	4.19	V
	外部输入引脚注 1	EXLVI	EXLVI < V _{DD} 、4.0V ≤ V _{DD} ≤ 5.5V	1.11	1.21	1.31
最小脉宽	t _{LW}		200			μs
运行稳定等待时间注 2	t _{LWAIT}		10			μs

注 1. 使用 EXLVI/P120/INTP0 引脚

2. 将低电压检测寄存器 (LVIM) 的 bit7 (LVION) 置“1”到运行稳定为止的时间。

备注 V_{LVI0} > V_{LVI1}

LVI 电路时序

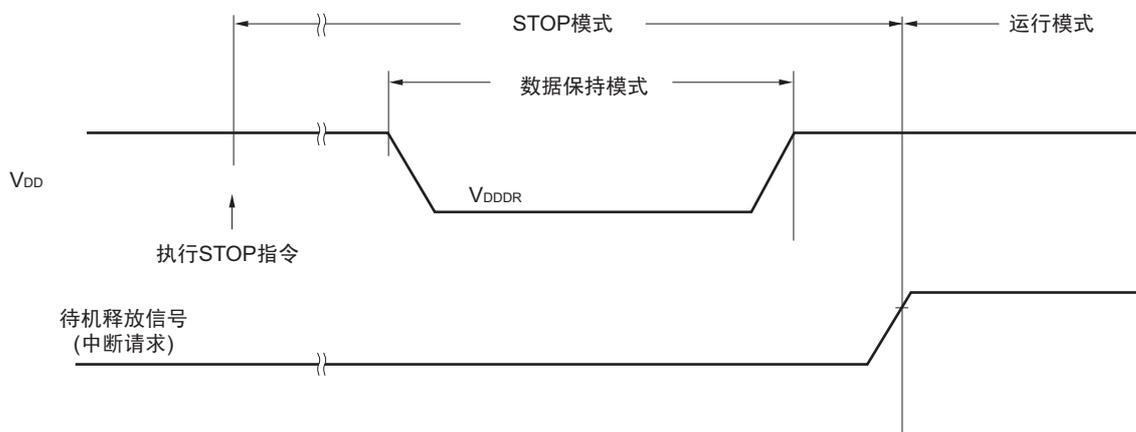


数据存储器 STOP 模式低电源电压数据保持特性

(T_A=-40 ~ +85°C)

项目	符号	条件	MIN.	TYP.	MAX.	单位
数据保持电源电压	V _{DDDR}		1.44 注		5.5	V

注 取决于 POC 检测电压。在电压下降时，保持数据直到 POC 复位。不保持 POC 复位后的数据。



闪存编程特性

(T_A=-40 ~ +85°C、4.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

• 基本特性

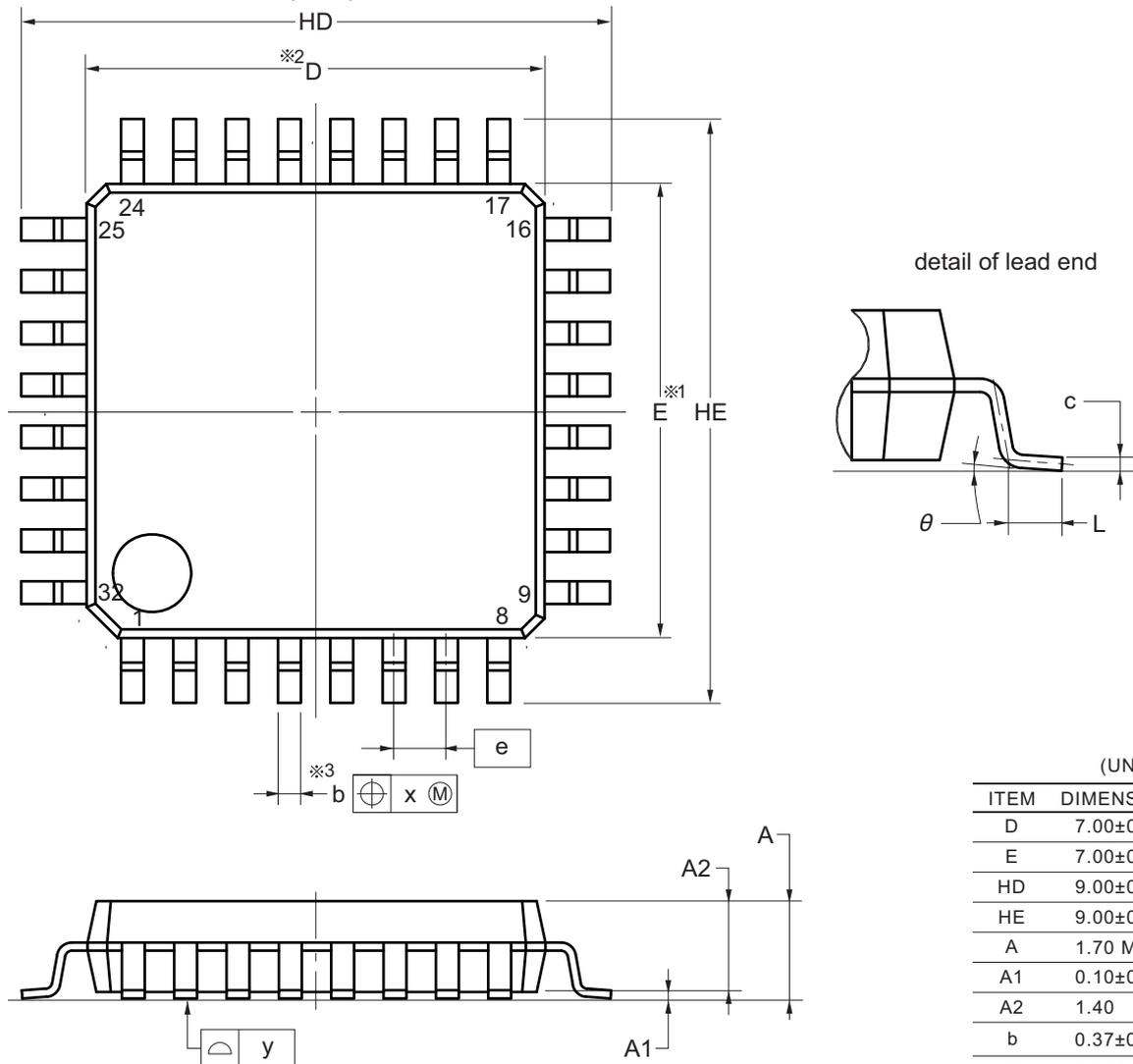
项目	符号	条件		MIN.	TYP.	MAX.	单位
V _{DD} 电源电流	I _{DD}				4.5	11.0	mA
每个芯片的改写次数注	C _{enwr}	1 次擦除 + 1 次擦除后的编程 = 1 次改写注	使用闪存编程器时，用于程序更新。	保持 15 年	10		次

注 在对出厂产品进行第一次编程时，不管是“擦除→编程”还是“只编程”，都被作为 1 次改写。

第 26 章 封装尺寸图

- R7F0C011B2001DFP
- R7F0C012B2001DFP
- R7F0C013B2001DFP
- R7F0C999B2DFP

32-PIN PLASTIC LQFP(7x7)



NOTE

- Dimensions "※1" and "※2" do not include mold flash.
- Dimension "※3" does not include trim offset.

第 27 章 等待的注意事项

27.1 等待的注意事项

本产品内置 2 种系统总线。

一种是 CPU 总线，另一种是连接低速外围硬件的外围总线。

因为 CPU 总线的时钟和外围总线的时钟为异步关系，所以在 CPU 和外围硬件的存取期间发生竞争时，有可能传送预想不到的非法数据。

因此，在存取有可能发生竞争的外围硬件时，重复执行处理直到 CPU 传送正确的数据。

其结果，CPU 不转移到下一条指令的处理，而是进入等待状态作为 CPU 处理，因此，在发生该等待时，指令的执行时钟数会多出等待时钟数的数量（等待时钟数请参照表 27-1）。在要求执行实时处理时，必须注意。

27.2 产生等待的外围硬件

在进行 CPU 存取时产生等待请求的寄存器和 CPU 的等待时钟数如表 27-1 所示。

表 27-1 产生等待的寄存器和 CPU 的等待时钟数

外围硬件	对象寄存器	对象存取	等待时钟数
串行接口 UART0	ASIS0	读	1 个时钟（固定）
串行接口 UART6	ASIS6	读	1 个时钟（固定）
串行接口 IIC0	IICS0	读	1 个时钟（固定）
A/D 转换器	ADM	写	1 ~ 5 个时钟（选择 $f_{AD}=f_{PRS}/2$ 时）
	ADS	写	1 ~ 7 个时钟（选择 $f_{AD}=f_{PRS}/3$ 时）
	ADPC	写	1 ~ 9 个时钟（选择 $f_{AD}=f_{PRS}/4$ 时）
	ADCR	读	2 ~ 13 个时钟（选择 $f_{AD}=f_{PRS}/6$ 时） 2 ~ 17 个时钟（选择 $f_{AD}=f_{PRS}/8$ 时） 2 ~ 25 个时钟（选择 $f_{AD}=f_{PRS}/12$ 时）
上述的时钟数是给 f_{CPU} 和 f_{PRS} 选择相同源时钟时的示例。可通过以下计算式和条件，计算等待时钟数。 < 等待时钟数的计算式 > • 等待时钟数 = $2f_{CPU}/f_{AD}+1$ - 在等待时钟数 ≤ 0.5 时，舍弃小数点以后的部分；在等待时钟数 > 0.5 时，进行四舍五入。 f_{AD} : A/D 转换时钟频率 ($f_{PRS}/2 \sim f_{PRS}/12$) f_{CPU} : CPU 时钟频率 f_{PRS} : 外围硬件时钟频率 f_{XP} : 主系统时钟频率 < 最大 / 最小等待时钟数的条件 > • 最大次数: CPU 最高速 (f_{XP})、A/D 转换时钟最低速 ($f_{PRS}/12$) • 最小次数: CPU 最低速 ($f_{XP}/16$)、A/D 转换时钟最高速 ($f_{PRS}/2$)			

注意 在外围硬件时钟 (f_{PRS}) 停止时，不能通过产生等待请求的存取方法存取上述对象寄存器。

备注 时钟表示 CPU 时钟 (f_{CPU})。

R7F0C011B、R7F0C012B、R7F0C013B
用户手册 硬件篇

Publication Date: Rev.1.00 Sep 10, 2013

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.
2880 Scott Boulevard Santa Clara, CA 95050-2554, 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: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

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: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, 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.
11F., Samik Lavied'or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

R7F0C011B、 R7F0C012B、 R7F0C013B



瑞萨电子株式会社

R01UH0408CJ0100