

# R7F0C30x、 R7F0C31x

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 R7F0C30x、R7F0C31x 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。
- 目的 本手册以帮助用户理解下述结构中所示的功能为目的。
- 构成 R7F0C30x、R7F0C31x 的用户手册分为用户手册硬件篇（本手册）和用户手册指令篇（78K/0 系列通用）共 2 本。

R7F0C30x、R7F0C31x  
用户手册 硬件篇

78K/0 系列  
用户手册 指令篇

- 引脚功能
  - 内部块功能
  - 中断
  - 其他的内部外围功能
  - 电特性
- CPU 功能
  - 指令集
  - 指令的说明

阅读方法 阅读本手册的读者应具备电气、逻辑电路以及微控制器的基础知识。

- 要理解全部功能时  
→ 请按照目录的顺序阅读本手册。
- 寄存器格式的阅读方法  
→ 关于方框 (<>) 内的位号，其位名称在 RA78K0 中被定义为保留字，而在 CC78K0 中被 #pragma sfr 指令定义为 sfr 变量。
- 要详细了解 78K0 微控制器的指令功能时  
→ 请参照另一本手册《78K/0 系列 用户手册 指令》(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

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

产品的相关资料

资料名称	资料号
R7F0C30x、R7F0C31x 用户手册 硬件篇	本手册
78K/0 系列 用户手册 指令	U12326C
78K0/Kx2 Flash 存储器编程器（编程器）应用笔记	U17739C
78K0/Kx2 Flash Memory Self Programming User's Manual	U17516E
78K0 Microcontrollers Self Programming Library Type01 User's Manual	U18274E

闪存编程的相关资料（用户手册）

资料名称	资料号
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）

资料名称	资料号
QB-MINI2 On-chip Debug Emulator with Programming User's Manual	R20UT0449E

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

开发工具（软件）的相关资料

资料名称		资料号
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 plus Ver.5.20 注 3 User's Manual		U16934E
PM+ Ver.6.30 注 4 User's Manual		U18416E

- 注 1. 在安装 RA78K0 Ver.4.01 时，该文件与工具一起安装到电脑内。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. 在安装 CC78K0 Ver.4.00 时，该文件与工具一起安装到电脑内。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 plus 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。可管理多种不同版本的软件工具（汇编器、C 编译器、调试器以及仿真器）产品。

其他资料

资料名称	资料号
Renesas Semiconductor Package Mount Manual	注
Quality Grades on NEC Semiconductor Devices	C11531E
Semiconductor Reliability Handbook	R51ZZ0001E
Guide to Prevent Damage for Semiconductor Derices by Electrostatic Discharge (ESD)	C11892E

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

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

所有商标及注册商标分别归属于其所有者。

本用户手册仅为参考译文，对应的日文版和英文版具有正式效力。

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

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

# 目 录

第 1 章 概述 .....	1
1.1 特点 .....	1
1.2 产品型号一览表 .....	3
1.3 引脚连接图（俯视图） .....	4
1.3.1 16 引脚产品 .....	4
1.3.2 20 引脚产品 .....	5
1.4 框图 .....	6
1.4.1 16 引脚产品 .....	6
1.4.2 20 引脚产品 .....	7
1.5 功能概要 .....	8
1.5.1 16 引脚产品 .....	8
1.5.2 20 引脚产品 .....	9
第 2 章 引脚功能 .....	10
2.1 引脚功能一览 .....	10
2.1.1 16 引脚产品 .....	10
2.1.2 20 引脚产品 .....	13
2.2 引脚功能的说明 .....	15
2.2.1 P20 ~ P27（Port2） .....	15
2.2.2 P30 ~ P34（Port3） .....	16
2.2.3 P121、P122、P125（Port12） .....	18
2.2.4 $V_{REF}$ 、 $V_{DD}$ 、 $V_{SS}$ .....	20
2.2.5 REGC .....	20
2.3 引脚的输入 / 输出电路和未使用引脚的处理 .....	21
第 3 章 CPU 体系结构 .....	25
3.1 存储空间 .....	25
3.1.1 内部程序存储空间 .....	29
3.1.2 内部数据存储空间 .....	31
3.1.3 特殊功能寄存器（SFR: Special Function Register）区域 .....	31
3.1.4 数据存储寄存器寻址 .....	32
3.2 处理器寄存器 .....	35
3.2.1 控制寄存器 .....	35
3.2.2 通用寄存器 .....	39
3.2.3 特殊功能寄存器（SFR: Special Function Register） .....	40
3.3 指令地址寻址 .....	45
3.3.1 相对寻址 .....	45
3.3.2 立即寻址 .....	46
3.3.3 表间接寻址 .....	47
3.3.4 寄存器寻址 .....	47
3.4 操作数地址寻址 .....	48
3.4.1 隐含寻址 .....	48
3.4.2 寄存器寻址 .....	49
3.4.3 直接寻址 .....	50
3.4.4 短直接寻址 .....	51
3.4.5 特殊功能寄存器（SFR）寻址 .....	52
3.4.6 寄存器间接寻址 .....	53
3.4.7 基址寻址 .....	54
3.4.8 基址变址寻址 .....	55
3.4.9 堆栈寻址 .....	56



第 4 章	端口功能 .....	57
4.1	端口功能 .....	57
4.2	端口结构 .....	59
4.2.1	端口 2 .....	59
4.2.2	端口 3 .....	74
4.2.3	端口 12 .....	82
4.3	端口功能的控制寄存器 .....	87
4.4	端口功能的运行 .....	93
4.4.1	输入 / 输出端口的写入 .....	93
4.4.2	输入 / 输出端口的读取 .....	93
4.4.3	输入 / 输出端口的运算 .....	93
4.5	使用复用功能时的端口模式寄存器和输出锁寄存器的设置 .....	94
4.6	端口寄存器 n (Pn) 的 1 位存储器操作指令的注意事项 .....	98
第 5 章	时钟发生电路 .....	99
5.1	时钟发生电路的功能 .....	99
5.2	时钟发生电路的结构 .....	100
5.3	时钟发生电路的控制寄存器 .....	103
5.4	系统时钟振荡电路 .....	110
5.4.1	X1 振荡电路 .....	110
5.4.2	高速内部振荡电路 .....	112
5.4.3	低速内部振荡电路 .....	112
5.4.4	预分频器 .....	112
5.5	时钟发生电路的运行 .....	113
5.6	时钟的控制 .....	116
5.6.1	高速系统时钟的控制示例 .....	116
5.6.2	高速内部振荡时钟的控制示例 .....	119
5.6.3	低速内部振荡时钟的控制示例 .....	121
5.6.4	作为 CPU 时钟和外围硬件时钟提供的时钟 .....	121
5.6.5	CPU 时钟状态转移图 .....	122
5.6.6	CPU 时钟的转移前条件和转移后处理 .....	125
5.6.7	主系统时钟的切换所需时间 .....	125
5.6.8	时钟振荡停止前的条件 .....	126
5.6.9	外围硬件和源时钟 .....	126
第 6 章	16 位定时器 / 事件计数器 00 .....	127
6.1	16 位定时器 / 事件计数器 00 的功能 .....	127
6.2	16 位定时器 / 事件计数器 00 的结构 .....	128
6.3	16 位定时器 / 事件计数器 00 的控制寄存器 .....	133
6.4	16 位定时器 / 事件计数器 00 的运行 .....	141
6.4.1	作为间隔定时器的运行 .....	141
6.4.2	方波的输出运行 .....	144
6.4.3	作为外部事件计数器的运行 .....	148
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 .....	151
6.4.5	作为自由运行定时器的运行 .....	163
6.4.6	PPG 的输出运行 .....	171
6.4.7	单次脉冲的输出运行 .....	174
6.4.8	脉冲的测量运行 .....	178
6.5	TM00 的特殊使用方法 .....	185
6.5.1	在 TM00 运行期间改写 CR010 .....	185
6.5.2	LVS00 和 LVR00 的设置 .....	185
6.6	16 位定时器 / 事件计数器 00 的注意事项 .....	187

<b>第 7 章 8 位定时器 / 事件计数器 51</b> .....	191
7.1 8 位定时器 / 事件计数器 51 的功能 .....	191
7.2 8 位定时器 / 事件计数器 51 的结构 .....	191
7.3 8 位定时器 / 事件计数器 51 的控制寄存器 .....	193
7.4 8 位定时器 / 事件计数器 51 的运行 .....	196
7.4.1 作为间隔定时器的运行 .....	196
7.4.2 作为外部事件计数器的运行 .....	198
7.4.3 方波的输出运行 .....	199
7.4.4 PWM 的输出运行 .....	200
7.5 8 位定时器 / 事件计数器 51 的注意事项 .....	203
<b>第 8 章 8 位定时器 H1</b> .....	204
8.1 8 位定时器 H1 的功能 .....	204
8.2 8 位定时器 H1 的结构 .....	204
8.3 8 位定时器 H1 的控制寄存器 .....	207
8.4 8 位定时器 H1 的运行 .....	210
8.4.1 作为间隔定时器 / 方波输出的运行 .....	210
8.4.2 PWM 的输出运行 .....	213
8.4.3 作为载波发生器的运行 .....	219
<b>第 9 章 看门狗定时器</b> .....	225
9.1 看门狗定时器的功能 .....	225
9.2 看门狗定时器的结构 .....	226
9.3 看门狗定时器的控制寄存器 .....	227
9.4 看门狗定时器的运行 .....	228
9.4.1 看门狗定时器的运行控制 .....	228
9.4.2 看门狗定时器的上溢时间设置 .....	229
9.4.3 看门狗定时器的窗口打开期间设置 .....	230
<b>第 10 章 A/D 转换器</b> .....	231
10.1 A/D 转换器的功能 .....	231
10.2 A/D 转换器的结构 .....	233
10.3 A/D 转换器使用的寄存器 .....	235
10.4 A/D 转换器的运行 .....	244
10.4.1 A/D 转换器的基本运行 .....	244
10.4.2 输入电压和转换结果 .....	245
10.4.3 A/D 转换器的运行模式 .....	246
10.5 A/D 转换器特性表的阅读方法 .....	248
10.6 A/D 转换器的注意事项 .....	252
<b>第 11 章 运算放大器</b> .....	255
11.1 运算放大器的功能 .....	255
11.2 运算放大器的结构 .....	255
11.3 运算放大器使用的寄存器 .....	257
11.4 运算放大器的运行 .....	263
<b>第 12 章 比较器</b> .....	264
12.1 比较器的功能 .....	264
12.2 比较器的结构 .....	265
12.3 比较器的控制寄存器 .....	266

12.4	比较器的运行 .....	270
12.4.1	开始比较器的运行（内部基准电压用作基准电压的情况） .....	270
12.4.2	开始比较器的运行（CMPCOM 引脚的输入电压用作基准电压的情况） .....	271
12.4.3	停止比较器的运行 .....	271
<b>第 13 章</b>	<b>串行接口 UART0 .....</b>	<b>272</b>
13.1	串行接口 UART0 的功能 .....	272
13.2	串行接口 UART0 的结构 .....	273
13.3	串行接口 UART0 的控制寄存器 .....	277
13.4	串行接口 UART0 的运行 .....	282
13.4.1	运行停止模式 .....	282
13.4.2	异步串行接口（UART）模式 .....	283
13.4.3	专用波特率发生器 .....	289
13.4.4	波特率的计算 .....	290
<b>第 14 章</b>	<b>中断功能 .....</b>	<b>294</b>
14.1	中断功能的种类 .....	294
14.2	中断源和结构 .....	295
14.3	中断功能的控制寄存器 .....	299
14.4	中断处理的运行 .....	305
14.4.1	可屏蔽中断请求的接受运行 .....	305
14.4.2	软件中断请求的接受运行 .....	307
14.4.3	多重中断处理 .....	308
14.4.4	中断请求的保留 .....	311
<b>第 15 章</b>	<b>待机功能 .....</b>	<b>312</b>
15.1	待机功能和结构 .....	312
15.1.1	待机功能 .....	312
15.1.2	待机功能的控制寄存器 .....	313
15.2	待机功能的运行 .....	315
15.2.1	HALT 模式 .....	315
15.2.2	STOP 模式 .....	318
<b>第 16 章</b>	<b>复位功能 .....</b>	<b>323</b>
16.1	复位源的确认寄存器 .....	330
<b>第 17 章</b>	<b>上电清除电路 .....</b>	<b>331</b>
17.1	上电清除电路的功能 .....	331
17.2	上电清除电路的结构 .....	331
17.3	上电清除电路的运行 .....	332
17.4	上电清除电路的注意事项 .....	334
<b>第 18 章</b>	<b>低电压检测电路 .....</b>	<b>336</b>
18.1	低电压检测电路的功能 .....	336
18.2	低电压检测电路的结构 .....	337
18.3	低电压检测电路的控制寄存器 .....	338
18.4	低电压检测电路的运行 .....	340
18.4.1	用作复位时的设置 .....	340
18.4.2	用作中断时的设置 .....	343
18.5	低电压检测电路的注意事项 .....	346
<b>第 19 章</b>	<b>稳压器 .....</b>	<b>349</b>
19.1	稳压器的概要 .....	349

第 20 章 选项字节 .....	350
20.1 选项字节的功能 .....	350
20.2 选项字节的格式 .....	352
第 21 章 闪存 .....	355
21.1 存储容量切换寄存器 .....	355
21.2 闪存编程器的编程方法 .....	356
21.3 编程环境 .....	356
21.4 板上引脚处理 .....	357
21.4.1 <u>TOOL</u> 引脚 .....	357
21.4.2 <u>RESET</u> 引脚 .....	358
21.4.3 端口引脚 .....	358
21.4.4 <u>REGC</u> 引脚 .....	358
21.4.5 其他信号的引脚 .....	358
21.4.6 电源 .....	358
21.4.7 连接晶体 / 陶瓷谐振器时的板上编程 .....	359
21.5 编程方法 .....	360
21.5.1 闪存控制 .....	360
21.5.2 闪存编程模式 .....	360
21.5.3 通信命令 .....	360
21.6 安全设置 .....	362
21.7 通过自编程进行的闪存编程 .....	364
21.7.1 自编程模式控制寄存器 .....	364
21.7.2 自编程（闪存的改写）的流程 .....	365
21.7.3 引导交换功能 .....	366
第 22 章 片上调试功能 .....	368
22.1 <u>QB-MINI2</u> 与 <u>R7F0C30x</u> 、 <u>R7F0C31x</u> 的连接 .....	368
22.2 片上调试安全 ID .....	370
22.3 保留用户资源 .....	370
第 23 章 指令集 .....	371
23.1 凡例 .....	371
23.1.1 操作数的标识符和描述方法 .....	371
23.1.2 操作栏的说明 .....	372
23.1.3 标志栏的说明 .....	372
23.2 操作一览 .....	373
23.3 各寻址的指令一览 .....	381
第 24 章 电特性 .....	384
第 25 章 封装尺寸图 .....	399
25.1 16 引脚产品 .....	399
25.2 20 引脚产品 .....	401
第 26 章 等待的注意事项 .....	403
26.1 等待的注意事项 .....	403
26.2 产生等待的外围硬件 .....	403
附录 A 修订记录 .....	404

## 第 1 章 概述

### 1.1 特点

- 搭载78K0 CPU内核
- I/O端口、ROM容量、RAM容量

产品名称	项目	I/O 端口	程序存储器 (闪存)	数据存储器 (内部高速 RAM)
16 引 脚 产 品	R7F0C30042ESP、R7F0C30642ESN	12 个 (CMOS 输入 / 输出: 11 个、 COMS 输入: 1 个)	4K 字节	512 字节
	R7F0C30142ESP、R7F0C30742ESN		8K 字节	512 字节
	R7F0C30242ESP、R7F0C30842ESN		16K 字节	768 字节
	R7F0C30342ESP、R7F0C30942ESN		4K 字节	512 字节
	R7F0C30442ESP、R7F0C31042ESN		8K 字节	512 字节
	R7F0C30542ESP、R7F0C31142ESN		16K 字节	768 字节
20 引 脚 产 品	R7F0C30062ESP、R7F0C30662ESN	16 个 (CMOS 输入 / 输出: 15 个、 COMS 输入: 1 个)	4K 字节	512 字节
	R7F0C30162ESP、R7F0C30762ESN		8K 字节	512 字节
	R7F0C30262ESP、R7F0C30862ESN		16K 字节	768 字节
	R7F0C30362ESP、R7F0C30962ESN		4K 字节	512 字节
	R7F0C30462ESP、R7F0C31062ESN		8K 字节	512 字节
	R7F0C30562ESP、R7F0C31162ESN		16K 字节	768 字节

- 低功耗
  - 高速内部振荡器运行模式: 1.2mA (TYP.) (以  $V_{DD}=3.0V$  运行)
  - STOP 模式: 1.2 $\mu$ A (TYP.) (以  $V_{DD}=5.0V$  运行)
- 时钟
  - 高速系统时钟...可从以下3种时钟源中选择
    - 陶瓷/晶体谐振器: 1~10MHz
    - 外部时钟: 1~10MHz
    - 高速内部振荡器:
      - SSOP封装产品的情况: 8MHz $\pm$ 3% ( $T_A=-40\sim+85^{\circ}C$ )、8MHz $\pm$ 2% ( $T_A=-20\sim+70^{\circ}C$ )
      - SOP封装产品的情况: 8MHz $\pm$ 5% ( $T_A=-40\sim+85^{\circ}C$ )、8MHz $\pm$ 2% ( $T_A=0\sim+40^{\circ}C$ )
  - 低速内部振荡器 (看门狗定时器、间歇运行时的定时器的时钟):
    - 2.7V  $\leq$   $V_{DD}$   $\leq$  5.5V: 240kHz $\pm$ 10%
    - 1.8V  $\leq$   $V_{DD}$  < 2.7V: 240kHz-20%、240kHz+10%
- 内置上电清除 (POC) 电路
- 内置低电压检测 (LVI) 电路 (在变为检测电压时, 发生中断/复位 (可选))
  - 检测电压: 可从1.93~4.24V间的16个级别中选择
- 单电源闪存
  - 可自编程
  - 软件保护功能...防止第三方复制 (无闪存读取命令)

- 安全功能
  - 可通过独立于CPU的时钟运行的看门狗定时器  
...即使在系统时钟停止运行时，也能监视失控。
  - 可通过LVI检测出电源电压下降  
...可在电源电压下降到低于工作电压之前，执行相应处理。
  - 选项字节功能  
...通过硬件进行重要系统的运行设置。
- 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H ...PWM输出、可通过低速内部振荡时钟运行
  - 8位定时器/事件计数器5 ...外部事件计数器输入
  - 看门狗定时器 ...可通过低速内部振荡时钟运行

项目	16位定时器 / 事件计数器	8位定时器	看门狗定时器
产品名称			
16引脚产品	1个通道	定时器 H: 1个通道	1个通道
20引脚产品		定时器 5: 1个通道	

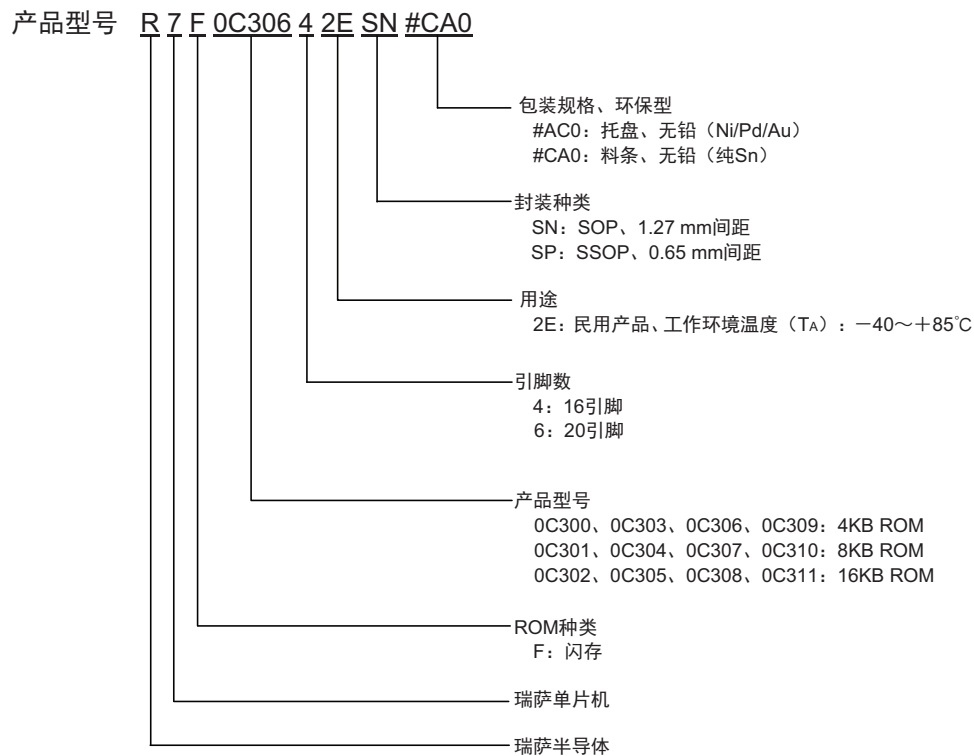
- 串行接口
  - UART ...支持异步通信的双线式串行接口

项目	UART
产品名称	
16引脚产品	1个通道
20引脚产品	

- 10位分辨率A/D转换器
  - 16引脚产品: 6个通道
  - 20引脚产品: 5个通道
- 内置运算放大器: 2个通道  
(只限R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x)
- 内置比较器: 1个通道  
(只限R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x)
- 片上调试功能 ...可控制目标设备的执行和存储器访问
- 支持汇编程序和C语言
- 开发工具
  - 支持简易仿真器 (MINICUBE2)
- 电源电压:  $V_{DD}=1.8\sim 5.5V$
- 工作环境温度:  $T_A=-40\sim +85^{\circ}C$

## 1.2 产品型号一览表

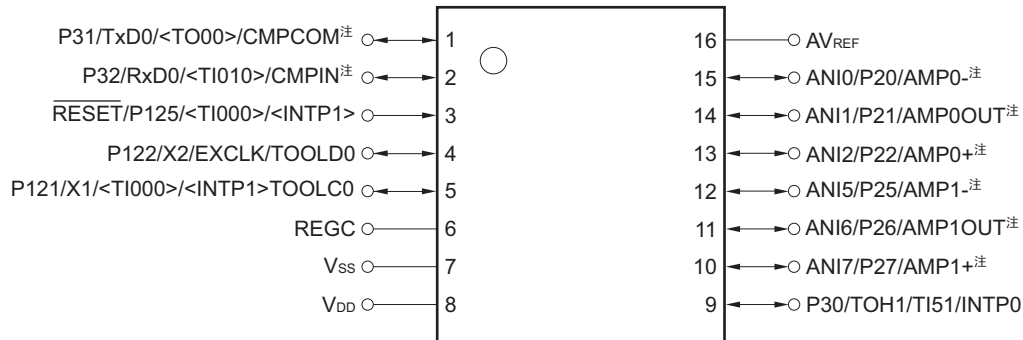
图 1-1 产品型号、存储容量和封装



引脚数	封装	ROM	RAM	包装规格、环保型	订购型号
16 引脚	16 引脚塑封 SSOP (5.72mm(225))	4KB	512B	托盘、无铅 (Ni/Pd/Au)	R7F0C30042ESP#AC0、R7F0C30342ESP#AC0
		8KB	512B		R7F0C30142ESP#AC0、R7F0C30442ESP#AC0
		16KB	768B		R7F0C30242ESP#AC0、R7F0C30542ESP#AC0
	16 引脚塑封 SOP (3.9mm×9.9mm)	4KB	512B	料条、无铅 (纯 Sn)	R7F0C30642ESN#CA0、R7F0C30942ESN#CA0
		8KB	512B		R7F0C30742ESN#CA0、R7F0C31042ESN#CA0
		16KB	768B		R7F0C30842ESN#CA0、R7F0C31142ESN#CA0
20 引脚	20 引脚塑封 SSOP (7.62mm(300))	4KB	512B	托盘、无铅 (Ni/Pd/Au)	R7F0C30062ESP#AC0、R7F0C30362ESP#AC0
		8KB	512B		R7F0C30162ESP#AC0、R7F0C30462ESP#AC0
		16KB	768B		R7F0C30262ESP#AC0、R7F0C30562ESP#AC0
	20 引脚塑封 SOP (7.5mm×12.8mm)	4KB	512B	料条、无铅 (纯 Sn)	R7F0C30662ESN#CA0、R7F0C30962ESN#CA0
		8KB	512B		R7F0C30762ESN#CA0、R7F0C31062ESN#CA0
		16KB	768B		R7F0C30862ESN#CA0、R7F0C31162ESN#CA0

## 1.3 引脚连接图（俯视图）

### 1.3.1 16 引脚产品



AMP0-、AMP0+、	: Amplifier Input	P20 ~ P22、P25 ~ P27	: Port2
AMP1-、AMP1+ 注		P30 ~ P32	: Port3
AMP0OUT、	: Amplifier Output	P121、P122、P125	: Port12
AMP1OUT 注		REGC	: Regulator Capacitance
ANI0 ~ ANI2	: Analog Input	RESET	: Reset
ANI5 ~ ANI7		RxD0	: Receive Data
AVREF	: Analog Reference Voltage	TI000、TI010、TI51	: Timer Input
		TO00、TOH1	: Timer Output
CMPCOM 注	: Comparator Common Input	TOOLC0	: Clock Input for Tool
		TOOLD0	: Data Input/Output for Tool
CMPIN 注	: Comparator Input	TxD0	: Transmit Data
EXCLK	: External Clock Input	VDD	: Power Supply
	(Main System Clock)	Vss	: Ground
INTP0、INTP1	: External Interrupt Input	X1、X2	: Crystal Oscillator
			(Main System Clock)

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114（内置运算放大器 / 比较器的产品）。

注意 1. Vss 复用 A/D 转换器的接地电位，并且必须连接稳定的 GND (=0V)。

2. REGC 通过电容器（0.47 ~ 1μF）连接 Vss。

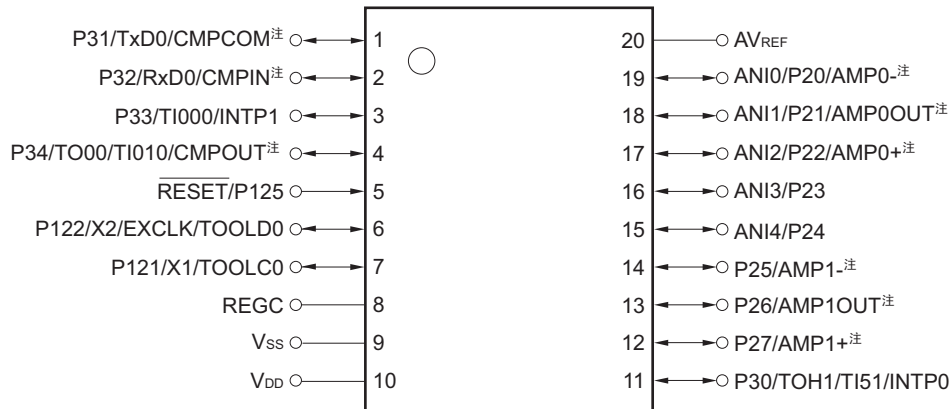
3. 复位解除后，ANI0/P20/AMP0-、ANI1/P21/AMP0OUT、ANI2/P22/AMP0+、ANI5/P25/AMP1-、ANI6/P26/AMP1OUT、ANI7/P27/AMP1+、P31/TxD0/<TO00>/CMPCOM、P32/RxD0/<TI010>/CMPIN 为模拟输入模式。

4. 复位解除后的 RESET/P125 为外部复位输入引脚。

备注 可通过设置端口复用切换控制寄存器（MUXSEL）指定上图 <> 内的功能。



## 1.3.2 20 引脚产品



AMP0-、AMP0+、	: Amplifier Input	P20 ~ P27	: Port2
AMP1-、AMP1+注		P30 ~ P34	: Port3
AMP0OUT、	: Amplifier Output	P121、P122、P125	: Port12
AMP1OUT注		REGC	: Regulator Capacitance
ANI0 ~ ANI4	: Analog Input	RESET	: Reset
AVREF	: Analog Reference Voltage	RxD0	: Receive Data
		TI000、TI010、TI51	: Timer Input
CMPCOM注	: Comparator Common Input	TO00、TOH1	: Timer Output
		TOOLC0	: Clock Input for Tool
CMPIN注	: Comparator Input	TOOLD0	: Data Input/Output for Tool
CMPOUT注	: Comparator Output	TxD0	: Transmit Data
EXCLK	: External Clock Input	VDD	: Power Supply
	(Main System Clock)	Vss	: Ground
INTP0、INTP1	: External Interrupt Input	X1、X2	: Crystal Oscillator
			(Main System Clock)

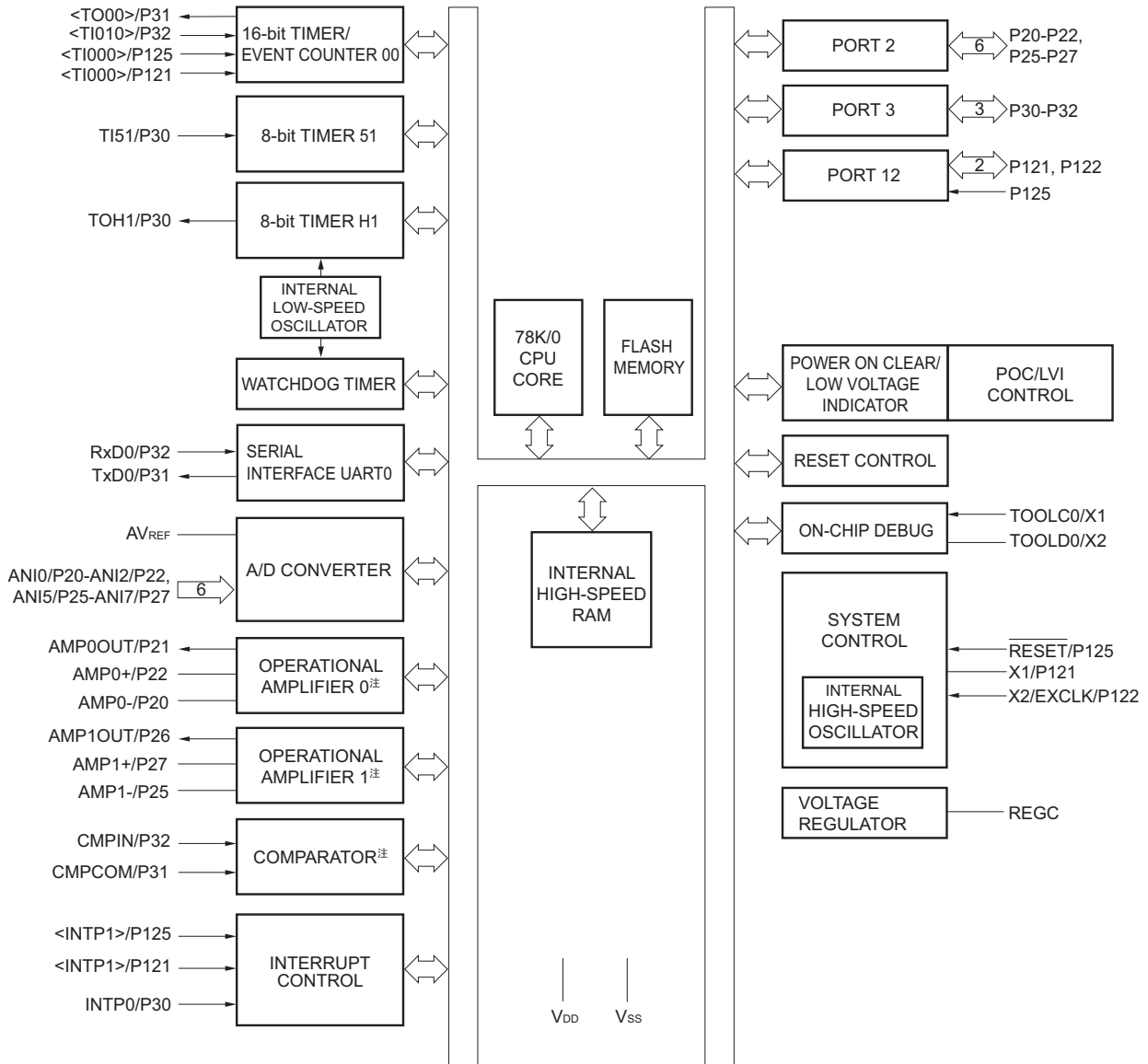
注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116（内置运算放大器 / 比较器的产品）。

注意 1. Vss 复用 A/D 转换器的接地电位，并且必须连接稳定的 GND (=0V)。

2. REGC 通过电容器（0.47 ~ 1μF）连接 Vss。
3. 复位解除后，ANI0/P20/AMP0-、ANI1/P21/AMP0OUT、ANI2/P22/AMP0+、ANI3/P23、ANI4/P24、P31/TxD0/CMPCOM、P32/RxD0/CMPIN 为模拟输入模式。
4. 复位解除后的 RESET/P125 为外部复位输入引脚。

## 1.4 框图

## 1.4.1 16 引脚产品



注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114（内置运算放大器 / 比较器的产品）。

注意 1.  $V_{SS}$  复用 A/D 转换器的接地电位，并且必须连接稳定的 GND (=0V)。

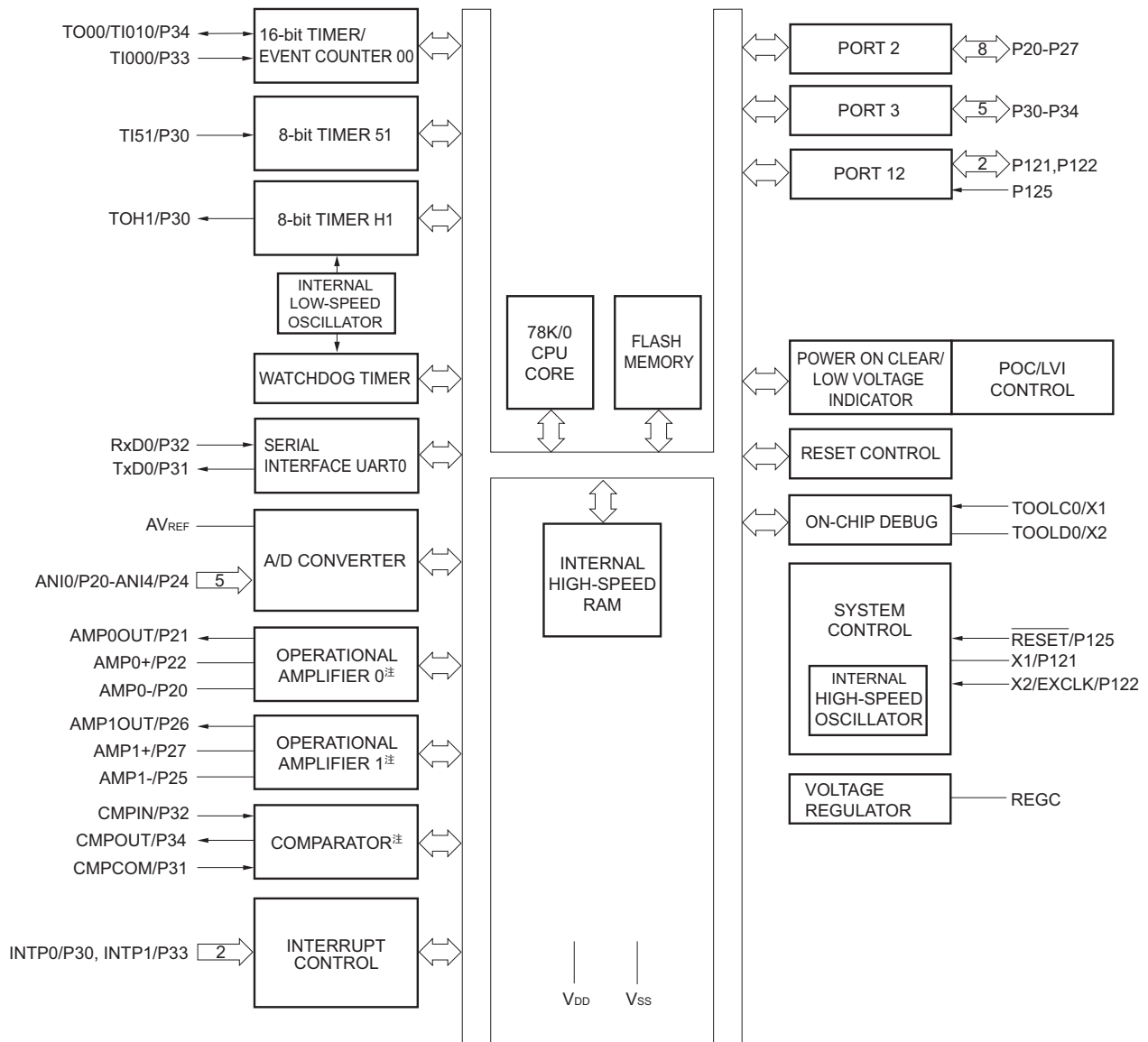
2. REGC 通过电容器（0.47 ~ 1 $\mu$ F）连接  $V_{SS}$ 。

3. 复位解除后，ANI0/P20/AMP0-、ANI1/P21/AMP0OUT、ANI2/P22/AMP0+、ANI5/P25/AMP1-、ANI6/P26/AMP1OUT、ANI7/P27/AMP1+、P31/TxD0/<TO00>/CMPCOM、P32/RxD0/<TI010>/CMPIN 为模拟输入模式。

4. 复位解除后的  $\overline{\text{RESET}}/P125$  为外部复位输入引脚。

备注 可通过设置端口复用切换控制寄存器（MUXSEL）指定上图 <> 内的功能。

## 1.4.2 20 引脚产品



注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116（内置运算放大器 / 比较器的产品）。

注意 1.  $V_{SS}$  复用 A/D 转换器的接地电位，并且必须连接稳定的 GND (=0V)。

2. REGC 通过电容器（0.47 ~ 1 $\mu$ F）连接  $V_{SS}$ 。

3. 复位解除后，ANI0/P20/AMP0-、ANI1/P21/AMP0OUT、ANI2/P22/AMP0+、ANI3/P23、ANI4/P24、P31/TxD0/CMPCOM、P32/RxD0/CMPIN 为模拟输入模式。

4. 复位解除后的  $\overline{\text{RESET}}$ /P125 为外部复位输入引脚。

## 1.5 功能概要

## 1.5.1 16 引脚产品

项目		R7F0C3004、 R7F0C3064	R7F0C3014、 R7F0C3074	R7F0C3024、 R7F0C3084	R7F0C3034、 R7F0C3094	R7F0C3044、 R7F0C3104	R7F0C3054、 R7F0C3114
内部存储器	闪存 (支持自编程)	4K 字节	8K 字节	16K 字节	4K 字节	8K 字节	16K 字节
	高速 RAM	512 字节	512 字节	768 字节	512 字节	512 字节	768 字节
存储空间		64K 字节					
时钟	主时钟	高速系统 (陶瓷 / 晶体振荡、 外部时钟输入)	1 ~ 10MHz: $V_{DD}=2.7 \sim 5.5V$ /1 ~ 5MHz: $V_{DD}=1.8 \sim 5.5V$				
		高速内部振荡	<ul style="list-style-type: none"> <li>16 引脚塑封 SSOP: <math>8MHz \pm 3\%</math> (<math>T_A=-40 \sim +85^\circ C</math>)、 <math>8MHz \pm 2\%</math> (<math>T_A=-20 \sim +70^\circ C</math>)</li> <li>16 引脚塑封 SOP: <math>8MHz \pm 5\%</math> (<math>T_A=-40 \sim +85^\circ C</math>)、 <math>8MHz \pm 2\%</math> (<math>T_A=0 \sim +40^\circ C</math>)</li> </ul>				
	低速内部振荡	<ul style="list-style-type: none"> <li><math>2.7V \leq V_{DD} \leq 5.5V</math>: <math>240kHz \pm 10\%</math></li> <li><math>1.8V \leq V_{DD} &lt; 2.7V</math>: <math>240kHz - 20\%</math>、<math>240kHz + 10\%</math></li> </ul>					
通用寄存器		8 位 $\times$ 32 个寄存器 (8 位 $\times$ 8 个寄存器 $\times$ 4 组)					
指令集		<ul style="list-style-type: none"> <li>8 位运算、16 位运算</li> <li>乘除运算 (8 位 <math>\times</math> 8 位、16 位 <math>\div</math> 8 位)</li> <li>位操作 (置位、复位、测试、布尔运算)</li> <li>BCD 校正等</li> </ul>					
I/O 端口 (总计)		12 个					
		CMOS 输入 / 输出	11 个				
		CMOS 输入	1 个				
定时器	16 位 (TM0)	1 个通道 (PPG 输出: 1 个、捕捉输入: 2 个)					
	8 位 (TM5)	1 个通道					
	8 位 (TMH)	1 个通道 (PWM 输出: 1 个)					
	看门狗 (WDT)	1 个通道					
串行接口	UART	1 个通道					
10 位 A/D 转换器		6 个通道 ( $AV_{REF}=2.3 \sim 5.5V$ )					
运算放大器		—			2 个通道 ( $V_{DD}=2.2 \sim 5.5V$ )		
比较器		—			1 个通道		
向量 中断源	外部	2 个			3 个		
	内部	8 个					
复位		<ul style="list-style-type: none"> <li>通过 RESET 引脚进行的复位</li> <li>通过看门狗定时器进行的内部复位</li> <li>通过上电清除 (POC) 进行的内部复位</li> <li>通过低电压检测 (LVI) 电路进行的内部复位</li> </ul>					
片上调试功能		有					
电源电压		$V_{DD}=1.8 \sim 5.5V$					
工作环境温度		$T_A=-40 \sim +85^\circ C$					
封装		<ul style="list-style-type: none"> <li>16 引脚塑封 SSOP</li> <li>16 引脚塑封 SOP</li> </ul>					

## 1.5.2 20 引脚产品

项目		R7F0C3006、 R7F0C3066	R7F0C3016、 R7F0C3076	R7F0C3026、 R7F0C3086	R7F0C3036、 R7F0C3096	R7F0C3046、 R7F0C3106	R7F0C3056、 R7F0C3116
内部存储器	闪存 (支持自编程)	4K 字节	8K 字节	16K 字节	4K 字节	8K 字节	16K 字节
	高速 RAM	512 字节	512 字节	768 字节	512 字节	512 字节	768 字节
存储空间		64K 字节					
时钟	主时钟	高速系统 (陶瓷 / 晶体振荡、 外部时钟输入)	1 ~ 10MHz: $V_{DD}=2.7 \sim 5.5V$ /1 ~ 5MHz: $V_{DD}=1.8 \sim 5.5V$				
		高速内部振荡	<ul style="list-style-type: none"> <li>20 引脚塑封 SSOP: 8MHz±3% (<math>T_A=-40 \sim +85^\circ C</math>)、 8MHz±2% (<math>T_A=-20 \sim +70^\circ C</math>)</li> <li>20 引脚塑封 SOP: 8MHz±5% (<math>T_A=-40 \sim +85^\circ C</math>)、 8MHz±2% (<math>T_A=0 \sim +40^\circ C</math>)</li> </ul>				
	低速内部振荡	<ul style="list-style-type: none"> <li><math>2.7V \leq V_{DD} \leq 5.5V</math>: 240kHz±10%</li> <li><math>1.8V \leq V_{DD} &lt; 2.7V</math>: 240kHz-20%、240kHz+10%</li> </ul>					
通用寄存器		8 位 × 32 个寄存器 (8 位 × 8 个寄存器 × 4 组)					
指令集		<ul style="list-style-type: none"> <li>8 位运算、16 位运算</li> <li>乘除运算 (8 位 × 8 位、16 位 ÷ 8 位)</li> <li>位操作 (置位、复位、测试、布尔运算)</li> <li>BCD 校正等</li> </ul>					
I/O 端口 (总计)		16 个					
	CMOS 输入 / 输出	15 个					
	CMOS 输入	1 个					
定时器	16 位 (TM0)	1 个通道 (PPG 输出: 1 个、捕捉输入: 2 个)					
	8 位 (TM5)	1 个通道					
	8 位 (TMH)	1 个通道 (PWM 输出: 1 个)					
	看门狗 (WDT)	1 个通道					
串行接口	UART	1 个通道					
10 位 A/D 转换器		5 个通道 ( $AV_{REF}=2.3 \sim 5.5V$ )					
运算放大器		—			2 个通道 ( $V_{DD}=2.2 \sim 5.5V$ )		
比较器		—			1 个通道		
向量 中断源	外部	2 个			3 个		
	内部	8 个					
复位		<ul style="list-style-type: none"> <li>通过 <math>\overline{RESET}</math> 引脚进行的复位</li> <li>通过看门狗定时器进行的内部复位</li> <li>通过上电清除 (POC) 进行的内部复位</li> <li>通过低电压检测 (LVI) 电路进行的内部复位</li> </ul>					
片上调试功能		有					
电源电压		$V_{DD}=1.8 \sim 5.5V$					
工作环境温度		$T_A=-40 \sim +85^\circ C$					
封装		<ul style="list-style-type: none"> <li>20 引脚塑封 SSOP</li> <li>20 引脚塑封 SOP</li> </ul>					

## 第 2 章 引脚功能

### 2.1 引脚功能一览

引脚的输入 / 输出缓冲器电源分为  $AV_{REF}$  和  $V_{DD}$ 。各电源和引脚的关系如下所示。

表 2-1 各引脚的输入 / 输出缓冲器电源

电源	对应引脚
$AV_{REF}$	P20 ~ P27 注
$V_{DD}$	P20 ~ P27 注以外的引脚

注 16 引脚产品：P20 ~ P22、P25 ~ P27

20 引脚产品：P20 ~ P27

#### 2.1.1 16 引脚产品

##### (1) 端口功能：16 引脚产品

功能名称	输入 / 输出	功能	复位时	复用功能
P20	输入 / 输出	端口 2 6 位输入 / 输出端口 可以 1 位单位指定输入 / 输出	模拟输入	ANI0/AMP0- 注
P21				ANI1/AMP0OUT 注
P22				ANI2/AMP0+ 注
P25				ANI5/AMP1- 注
P26				ANI6/AMP1OUT 注
P27				ANI7/AMP1+ 注
P30	输入 / 输出	端口 3 3 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	TOH1/TI51/INTP0
P31			模拟输入	TxD0/<TO00>/ CMPCOM 注
P32				RxD0/<TI010>/ CMPIN 注
P121	输入 / 输出	端口 12 P121、P122 为 2 位输入 / 输出端口 P125 为 1 位输入专用端口 仅 P125 可通过软件设置，使用内部上拉电阻。	输入端口	X1/<TI000>/ <INTP1>/TOOLC0
P122				X2/EXCLK/TOOLD0
P125	输入		复位输入	<TI000>/<INTP1>/ RESET

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114（内置运算放大器 / 比较器的产品）。

备注 可通过设置端口复用切换控制寄存器（MUXSEL）指定上表 < > 内的功能。

## (2) 非端口功能：16 引脚产品 (1/2)

功能名称	输入 / 输出	功能	复位时	复用功能
ANI0	输入	A/D 转换器的模拟输入	模拟输入	P20/AMP0- 注
ANI1				P21/AMP0OUT 注
ANI2				P22/AMP0+ 注
ANI5				P25/AMP1- 注
ANI6				P26/AMP1OUT 注
ANI7				P27/AMP1+ 注
AMP0- 注				输入
AMP0+ 注	P22/ANI2			
AMP0OUT 注	输出	运算放大器 0 输出		P21/ANI1
AMP1- 注	输入	运算放大器 1 输入	模拟输入	P25/ANI5
AMP1+ 注				P27/ANI7
AMP1OUT 注	输出	运算放大器 1 输出		P26/ANI6
CMPCOM 注	输入	比较器公共输入	模拟输入	P31/TxD0/<TO00>
CMPIN 注		比较器输入		P32/RxD0/<TI010>
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	P30/TOH1/TI51
INTP1				P121/X1/<TI000>/ TOOLC0 P125/<TI000>/RESET
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 ~ 1μF）连接到 V <sub>SS</sub> 。	—	—
RESET	输入	系统复位输入	复位输入	P125/<TI000>/<INTP1>
RxD0	输入	UART0 的串行数据输入	输入端口	P32/<TI010>/CMPIN 注
TxD0	输出	UART0 的串行数据输出		P31/<TO00>/ CMPCOM 注
TI000	输入	16 位定时器 / 事件计数器 00 的外部计数时钟输入	输入端口	P121/X1/<INTP1>/ TOOLC0
		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		P125/<INTP1>/ RESET
TI010		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		P32/RxD0/CMPIN 注
TI51	输入	8 位定时器 / 事件计数器 51 的外部计数时钟输入	输入端口	P30/TOH1/INTP0
TO00	输出	16 位定时器 / 事件计数器 00 输出	输入端口	P31/TxD0/CMPCOM 注
TOH1	输出	8 位定时器 H1 输出	输入端口	P30/TI51/INTP0
X1	—	连接用于主系统时钟的谐振器	输入端口	P121/<TI000>/ <INTP1>/TOOLC0
X2				P122/EXCLK/TOOLD0
EXCLK	输入	主系统时钟的外部时钟输入	输入端口	P122/X2/TOOLD0

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114（内置运算放大器 / 比较器的产品）。

备注 可通过设置端口复用切换控制寄存器（MUXSEL）指定上表 <> 内的功能。

## (2) 非端口功能：16 引脚产品 (2/2)

功能名称	输入 / 输出	功能	复位时	复用功能
V <sub>DD</sub>	—	端口 2 以外的正电源	—	—
AV <sub>REF</sub>	—	A/D 转换器的基准电压输入和端口 2、A/D 转换器的正电源		
V <sub>SS</sub>	—	接地电位	—	—
TOOLC0	输入	闪存编程器 / 片上调试的时钟输入	输入端口	P121/X1/<TI000>/<INTP1>
TOOLD0	输入 / 输出	闪存编程器 / 片上调试的数据输入 / 输出		P122/X2/EXCLK

备注 可通过设置端口复用切换控制寄存器 (MUXSEL) 指定上表 < > 内的功能。



## 2.1.2 20 引脚产品

## (1) 端口功能：20 引脚产品

功能名称	输入 / 输出	功能	复位时	复用功能
P20	输入 / 输出	端口 2 8 位输入 / 输出端口 可以 1 位单位指定输入 / 输出	模拟输入	ANI0/AMP0- 注
P21				ANI1/AMP0OUT 注
P22				ANI2/AMP0+ 注
P23				ANI3
P24				ANI4
P25				AMP1- 注
P26				AMP1OUT 注
P27				AMP1+ 注
P30	输入 / 输出	端口 3 5 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	TOH1/TI51/INTP0
P31			模拟输入	TxD0/CMPCOM 注
P32				RxD0/CMPIN 注
P33			输入端口	TI000/INTP1
P34				TO00/TI010/ CMPOUT 注
P121	输入 / 输出	端口 12 P121、P122 为 2 位输入 / 输出端口	输入端口	X1/TOOLC0
P122				X2/EXCLK/TOOLD0
P125	输入	P125 为 1 位输入专用端口 仅 P125 可通过软件设置，使用内部上拉电阻。	复位输入	RESET

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116（内置运算放大器 / 比较器的产品）。

## (2) 非端口功能：20 引脚产品（1/2）

功能名称	输入 / 输出	功能	复位时	复用功能
ANI0	输入	A/D 转换器的模拟输入	模拟输入	P20/AMP0- 注
ANI1				P21/AMP0OUT 注
ANI2				P22/AMP0+ 注
ANI3				P23
ANI4				P24
AMP0- 注	输入	运算放大器 0 输入	模拟输入	P20/ANI0
AMP0+ 注				P22/ANI2
AMP0OUT 注	输出	运算放大器 0 输出		P21/ANI1
AMP1- 注	输入	运算放大器 1 输入	模拟输入	P25
AMP1+ 注				P27
AMP1OUT 注				输出
CMPCOM 注	输入	比较器公共输入	模拟输入	P31/TxD0
CMPIN 注		比较器输入		P32/RxD0
CMPOUT 注	输出	比较器输出		P34/TO00/TI010

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116（内置运算放大器 / 比较器的产品）。

## (2) 非端口功能：20 引脚产品 (2/2)

功能名称	输入 / 输出	功能	复位时	复用功能
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	P30/TOH1/TI51
INTP1				P33/TI000
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 $\mu$ F）连接到 V <sub>SS</sub> 。	—	—
RESET	输入	系统复位输入	复位输入	P125
RxD0	输入	UART0 的串行数据输入	输入端口	P32/CMPIN 注
TxD0	输出	UART0 的串行数据输出		P31/CMPCOM 注
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	P33/INTP1
TI010		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		P34/TO00/CMPOUT 注
TI51	输入	8 位定时器 / 事件计数器 51 的外部计数时钟输入	输入端口	P30/TOH1/INTP0
TO00	输出	16 位定时器 / 事件计数器 00 输出	输入端口	P34/TI010/CMPOUT 注
TOH1	输出	8 位定时器 H1 输出	输入端口	P30/TI51/INTP0
X1	—	连接用于主系统时钟的谐振器	输入端口	P121/TOOLC0
X2				P122/EXCLK/TOOLD0
EXCLK	输入	主系统时钟的外部时钟输入	输入端口	P122/X2/TOOLD0
V <sub>DD</sub>	—	端口 2 以外的正电源	—	—
AV <sub>REF</sub>		A/D 转换器的基准电压输入和端口 2、A/D 转换器的正电源		
V <sub>SS</sub>	—	接地电位	—	—
TOOLC0	输入	闪存编程器 / 片上调试的时钟输入	输入端口	P121/X1
TOOLD0	输入 / 输出	闪存编程器 / 片上调试的数据输入 / 输出		P122/X2/EXCLK

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116（内置运算放大器 / 比较器的产品）。

## 2.2 引脚功能的说明

备注 搭载的引脚因产品而不同。请参照“1.3 引脚连接图（俯视图）”、“2.1 引脚功能一览”。

### 2.2.1 P20 ~ P27 (Port2)

P20 ~ P27 即可用作输入 / 输出端口，也可用作 A/D 转换器的模拟输入、运算放大器的输入 / 输出。

16 引脚产品	20 引脚产品
P20/ANI0/AMP0- 注 1	P20/ANI0/AMP0- 注 2
P21/ANI1/AMP0OUT 注 1	P21/ANI1/AMP0OUT 注 2
P22/ANI2/AMP0+ 注 1	P22/ANI2/AMP0+ 注 2
—	P23/ANI3
—	P24/ANI4
P25/ANI5/AMP1- 注 1	P25/AMP1- 注 2
P26/ANI6/AMP1OUT 注 1	P26/AMP1OUT 注 2
P27/ANI7/AMP1+ 注 1	P27/AMP1+ 注 2

注 1. 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114（内置运算放大器 / 比较器的产品）。

2. 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116（内置运算放大器 / 比较器的产品）。

可以以 1 位单位指定以下运行模式。

#### (1) 端口模式

P20 ~ P27 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

P20 ~ P27 用作 A/D 转换器的模拟输入和运算放大器的输入 / 输出。

##### (a) ANI0 ~ ANI7

A/D 转换器的模拟输入引脚。在用作模拟输入引脚时，请参照“10.6 A/D 转换器的注意事项”的“(5) ANI0/P20 ~ ANI7/P27”。

##### (b) AMP0+、AMP0-

运算放大器 0 的输入引脚。

##### (c) AMP0OUT

运算放大器 0 的输出引脚。

##### (d) AMP1+、AMP1-

运算放大器 1 的输入引脚。

##### (e) AMP1OUT

运算放大器 1 的输出引脚。

注意 复位解除后，ANI0/P20 ~ ANI7/P27 为模拟输入模式。

## 2.2.2 P30 ~ P34 (Port3)

P30 ~ P34 即可用作输入 / 输出端口，也可用作外部中断请求输入、定时器的输入 / 输出、串行接口的数据输入 / 输出、比较器输出、比较器模拟输入以及比较器公共输入。

16 引脚产品	20 引脚产品
P30/TOH1/TI51/INTP0	P30/TOH1/TI51/INTP0
P31/TxD0/<TO00>/CMPCOM 注 1	P31/TxD0/CMPCOM 注 2
P32/RxD0/<TI010>/CMPIN 注 1	P32/RxD0/CMPIN 注 2
—	P33/TI000/INTP1
—	P34/TO00/TI010/CMPOUT 注 2

- 注 1. 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114（内置运算放大器 / 比较器的产品）。
2. 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116（内置运算放大器 / 比较器的产品）。

备注 可通过设置端口复用切换控制寄存器（MUXSEL）指定上表 < > 内的功能。

可以以 1 位单位指定以下运行模式。

### (1) 端口模式

P30 ~ P34 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

P30 ~ P34 用作外部中断请求输入、定时器的输入 / 输出、串行接口的数据输入 / 输出、比较器输出、比较器模拟输入以及比较器公共输入。

#### (a) INTP0、INTP1

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引脚。

#### (b) TI000

16 位定时器 / 事件计数器 00 的外部计数时钟输入引脚和 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。

#### (c) TI010

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。

#### (d) TO00

16 位定时器 / 事件计数器 00 的定时器输出引脚。

#### (e) TI51

8 位定时器 / 事件计数器 51 的外部计数时钟输入引脚。

#### (f) TOH1

8 位定时器 H1 的定时器输出引脚。

(g) RxD0

串行接口 UART0 的串行数据输入引脚。

(h) TxD0

串行接口 UART0 的串行数据输出引脚。

(i) CMPCOM

比较器公共输入引脚。

(j) CMPIN

比较器输入引脚。

(k) CMPOUT

比较器输出引脚。

### 2.2.3 P121、P122、P125 (Port12)

P121、P122 用作输入 / 输出端口。P125 用作输入端口。此外还用作主系统时钟的谐振器连接、主系统时钟的外部时钟输入、外部复位输入、闪存编程器 / 片上调试的时钟输入、数据输入 / 输出、外部中断请求输入以及定时器输入。

P125/RESET 用作输入端口时，将复位引脚模式寄存器 (RSTMASK) 的 bit5 (RSTM) 置“1”；用作外部复位输入时，将 RSTM 置“0”。

16 引脚产品	20 引脚产品
P121/X1/<TI000>/<INTP1>/TOOLC0	P121/X1/TOOLC0
P122/X2//EXCLK/TOOLD0	P122/X2/EXCLK/TOOLD0
P125/<TI000>/<INTP1>/RESET	P125/RESET

备注 可通过设置端口复用切换控制寄存器 (MUXSEL) 指定上表 <> 内的功能。

可以以 1 位单位指定以下运行模式。

#### (1) 端口模式

P121、P122 用作输入 / 输出端口。可通过设置端口模式寄存器 12 (PM12)，指定为输入端口或者输出端口。仅 P125 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

P121、P122 和 P125 用作主系统时钟的谐振器连接、主系统时钟的外部时钟输入、外部复位输入、闪存编程器 / 片上调试的时钟输入、数据输入 / 输出、外部中断请求输入以及定时器输入。

##### (a) X1、X2

主系统时钟的谐振器连接引脚。

##### (b) EXCLK

主系统时钟的外部时钟输入引脚。

##### (c) $\overline{\text{RESET}}$

低电平有效的系统复位输入引脚。

##### (d) TOOLC0

闪存编程器 / 片上调试的时钟输入引脚。

##### (e) TOOLD0

闪存编程器 / 片上调试的数据输入 / 输出引脚。

##### (f) INTP1

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引脚。

## (g) TI000

16 位定时器 / 事件计数器 00 的外部计数时钟输入引脚和 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。

**注意** 复位解除后的  $\overline{\text{RESET}}/\text{P125}$  为外部复位输入，因此，如果在低电平输入期间发生复位，就保持复位状态直到变为高电平输入为止。

**备注** 有关将 TOOLC0/X1 和 TOOLD0/X2 连接至闪存编程器的方法，请参照“第 21 章 闪存”；有关将 TOOLC0/X1 和 TOOLD0/X2 连接至片上调试仿真器的方法，请参照“第 22 章 片上调试功能”。

## 2.2.4 $AV_{REF}$ 、 $V_{DD}$ 、 $V_{SS}$

$AV_{REF}$ 、 $V_{DD}$ 、 $V_{SS}$  为电源 / 接地引脚。

### (a) $AV_{REF}$

$AV_{REF}$  为 A/D 转换器的基准电压输入以及提供端口 2 和 A/D 转换器的正电源的引脚。  
在不使用 A/D 转换器时，必须直接连接  $V_{DD}$  注。

注 在端口 2 用作数字端口时，将  $AV_{REF}$  设置为与  $V_{DD}$  相同的电位。

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

$V_{DD}$  为提供正电源的引脚。

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

$V_{SS}$  为接地电位引脚注。

注  $V_{SS}$  复用 A/D 转换器的接地电位。必须将  $V_{SS}$  连接到稳定的 GND (=0V)。

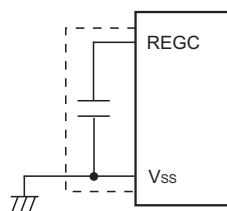
## 2.2.5 REGC

REGC 为稳压器的电容连接引脚和内部连接引脚。

### (a) REGC

REGC 为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 $\mu$ F) 连接到  $V_{SS}$ 。  
但是，如果在高速内部振荡时钟、外部主系统时钟运行时使用 STOP 模式，推荐 0.47 $\mu$ F。

另外，在用于稳定内部电压时，请使用性能良好的电容器。



注意 上图虚线部分的布线尽量使用最短布线。



## 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、表 2-3 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（16 引脚产品）

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
ANI0/P20/AMP0- 注 3	11-P	输入 / 输出	< 设置为数字输入时 > 通过电阻，分别连接到 $AV_{REF}$ 或者 $V_{SS}$ 。 < 设置为数字输出和设置为模拟输入时 > 必须置为开路。注 1
ANI1/P21/AMP0OUT 注 3	11-S		
ANI2/P22/AMP0+ 注 3	11-N		
ANI5/P25/AMP1- 注 3	11-P		
ANI6/P26/AMP1OUT 注 3	11-S		
ANI7/P27/AMP1+ 注 3	11-N		
P30/TOH1/TI151/INTP0	5-AH		输入时：通过电阻，分别连接到 $V_{DD}$ 或者 $V_{SS}$ 。 输出时：必须置为开路。
P31/TxD0/<TO00>/ CMPCOM 注 3	5-BD		< 设置为数字输入时 > 通过电阻，分别连接到 $V_{DD}$ 或者 $V_{SS}$ 。 < 设置为数字输出和设置为模拟输入时 > 必须置为开路。注 1
P32/RxD0/<TI010>/ CMPIN 注 3	5-BB		< 设置为数字输入时 > 通过电阻，分别连接到 $V_{DD}$ 或者 $V_{SS}$ 。 < 设置为数字输出和设置为模拟输入时 > 必须置为开路。注 1
P121/X1/<TI000>/<INTP1>/ TOOLC0 注 2	37-H		输入时：通过电阻，分别连接到 $V_{DD}$ 或者 $V_{SS}$ 。 输出时：必须置为开路。
P122/X2/EXCLK/TOOLD0 注 2			
RESET/P125/<TI000>/ <INTP1>	42-B	输入	直接连接到 $V_{DD}$ 或者通过电阻连接到 $V_{DD}$ 。
$AV_{REF}$	—	—	直接连接到 $V_{DD}$ 。

- 注 1. 如果在设置为模拟输入时置为开路，输入电压电平就不定，因此，推荐在设置为数字输出时置为开路。
2. 未使用时，在输入端口模式（参照“图 5-2 时钟运行模式选择寄存器（OSCCTL）的格式”）下执行上述推荐的连接方法。
3. 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114（内置运算放大器 / 比较器的产品）。

- 注意 1. 在复位解除后，ANI0/P20/AMP0-、ANI1/P21/AMP0OUT、ANI2/P22/AMP0+、ANI5/P25/AMP1-、ANI6/P26/AMP1OUT、ANI7/P27/AMP1+、P31/TxD0/<TO00>/CMPCOM、P32/RxD0/<TI010>/CMPIN 为模拟输入模式。
2. 复位解除后的 RESET/P125 为外部复位输入，因此，如果在低电平输入期间发生复位，就保持复位状态直到变为高电平输入为止。

备注 可通过设置端口复用切换控制寄存器（MUXSEL）指定上表 < > 内的功能。

表 2-3 各引脚的输入 / 输出电路类型 (20 引脚产品)

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	
ANI0/P20/AMP0- 注 3	11-P	输入 / 输出	< 设置为数字输入时 > 通过电阻, 分别连接到 $AV_{REF}$ 或者 $V_{SS}$ 。 < 设置为数字输出和设置为模拟输入时 > 必须置为开路。注 1	
ANI1/P21/AMP0OUT 注 3	11-S			
ANI2/P22/AMP0+ 注 3	11-N			
ANI3/P23	11-G			
ANI4/P24				
P25/AMP1- 注 3	11-P			
P26/AMP1OUT 注 3	11-S			
P27/AMP1+ 注 3	11-N			
P30/TOH1/TI151/INTP0	5-AH			输入时: 通过电阻, 分别连接到 $V_{DD}$ 或者 $V_{SS}$ 。 输出时: 必须置为开路。
P31/TxD0/CMPCOM 注 3	5-BD			< 设置为数字输入时 > 通过电阻, 分别连接到 $V_{DD}$ 或者 $V_{SS}$ 。 < 设置为数字输出和设置为模拟输入时 > 必须置为开路。注 1
P32/RxD0/CMPIN 注 3	5-BB			
P33/TI000/INTP1	5-AH			输入时: 通过电阻, 分别连接到 $V_{DD}$ 或者 $V_{SS}$ 。 输出时: 必须置为开路。
P34/TO00/TI010/CMPOUT 注 3				
P121/X1/TOOLC0 注 2	37-H			
P122/X2/EXCLK/TOOLD0 注 2				
RESET/P125	42-B	输入	直接连接到 $V_{DD}$ 或者通过电阻连接到 $V_{DD}$ 。	
$AV_{REF}$	—	—	直接连接到 $V_{DD}$ 。	

- 注 1. 如果在设置为模拟输入时置为开路, 输入电压电平就不定, 因此, 推荐在设置为数字输出时置为开路。
2. 未使用时, 在输入端口模式 (参照“图 5-2 时钟运行模式选择寄存器 (OSCCTL) 的格式”) 下执行上述推荐的连接方法。
3. 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置运算放大器 / 比较器的产品)。

- 注意 1. 在复位解除后, ANI0/P20/AMP0-、ANI1/P21/AMP0OUT、ANI2/P22/AMP0+、ANI3/P23、ANI4/P24、P31/TxD0/CMPCOM、P32/RxD0/CMPIN 为模拟输入模式。
2. 复位解除后的  $\overline{\text{RESET}}$ /P125 为外部复位输入, 因此, 如果在低电平输入期间发生复位, 就保持复位状态直到变为高电平输入为止。

图 2-1 引脚的输入 / 输出电路一览 (1/2)

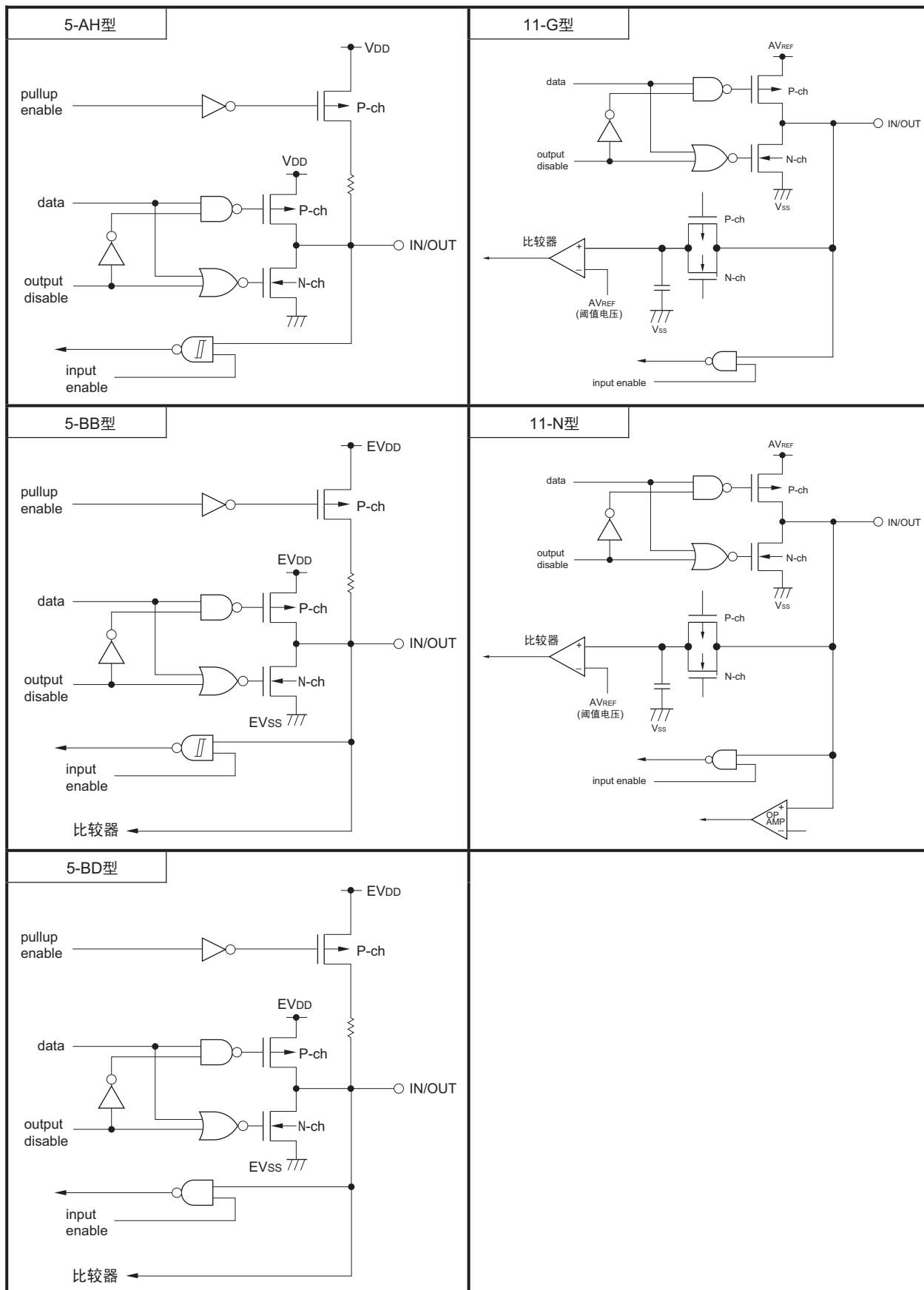
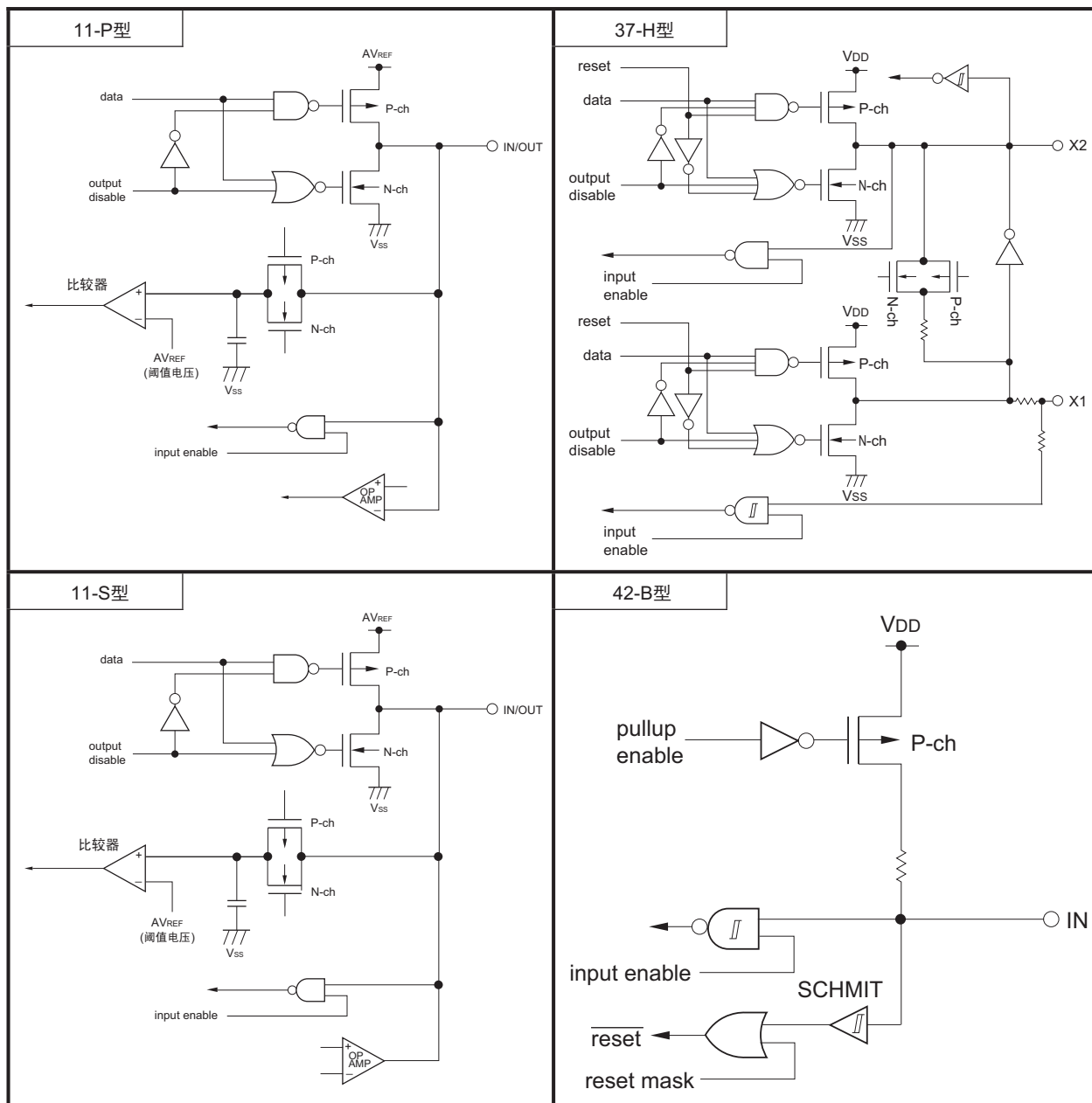


图 2-1 引脚的输入 / 输出电路一览 (2/2)



## 第3章 CPU 体系结构

### 3.1 存储空间

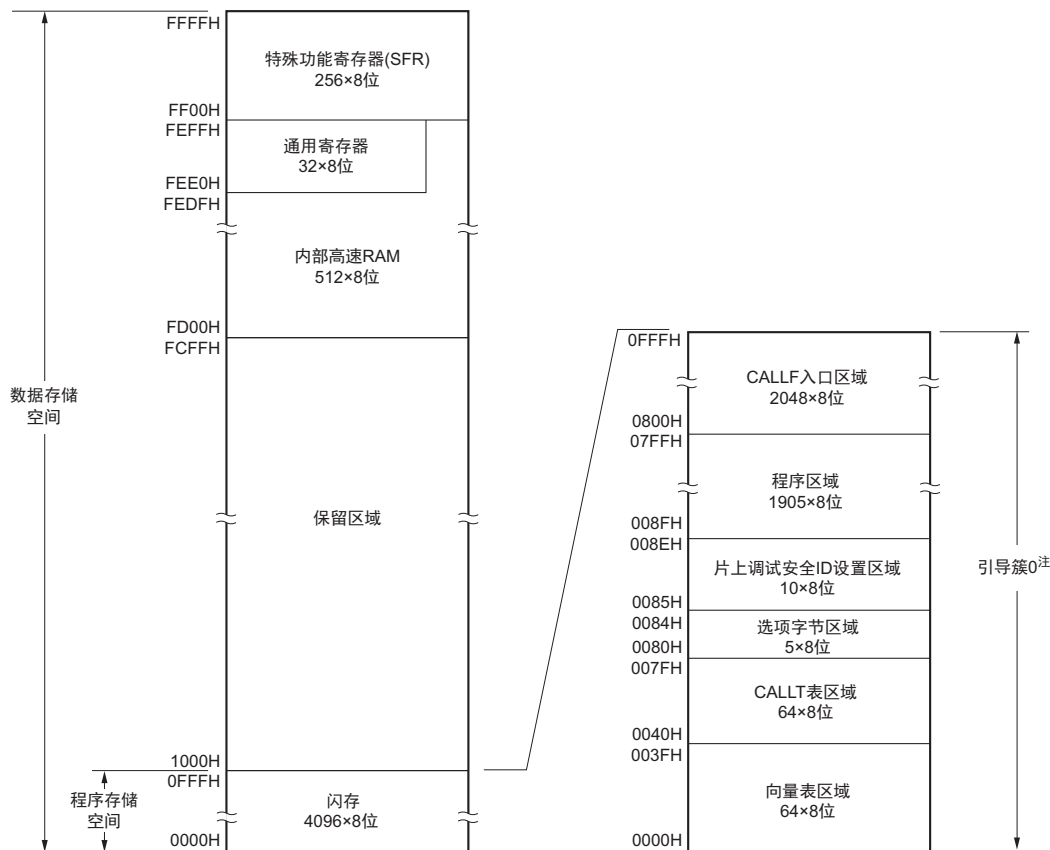
R7F0C30x、R7F0C31x 可存取 64K 字节的存储空间。存储器映射图如图 3-1 ~ 图 3-3 所示。

注意 由于复位时 ROM 区域的设置不定，因此，复位解除后，必须对每个产品设置以下所示的值。

表 3-1 存储容量切换寄存器（IMS）的设置值

产品	IMS	ROM 容量	内部高速 RAM 容量
R7F0C3004、R7F0C3034、R7F0C3006、R7F0C3036、 R7F0C3064、R7F0C3094、R7F0C3066、R7F0C3096	41H	4K 字节	512 字节
R7F0C3014、R7F0C3044、R7F0C3016、R7F0C3046、 R7F0C3074、R7F0C3104、R7F0C3076、R7F0C3106、	42H	8K 字节	512 字节
R7F0C3024、R7F0C3054、R7F0C3026、R7F0C3056、 R7F0C3084、R7F0C3114、R7F0C3086、R7F0C3116	04H	16K 字节	768 字节

图 3-1 存储器映射图 (R7F0C300x、R7F0C303x、R7F0C306x、R7F0C309x)

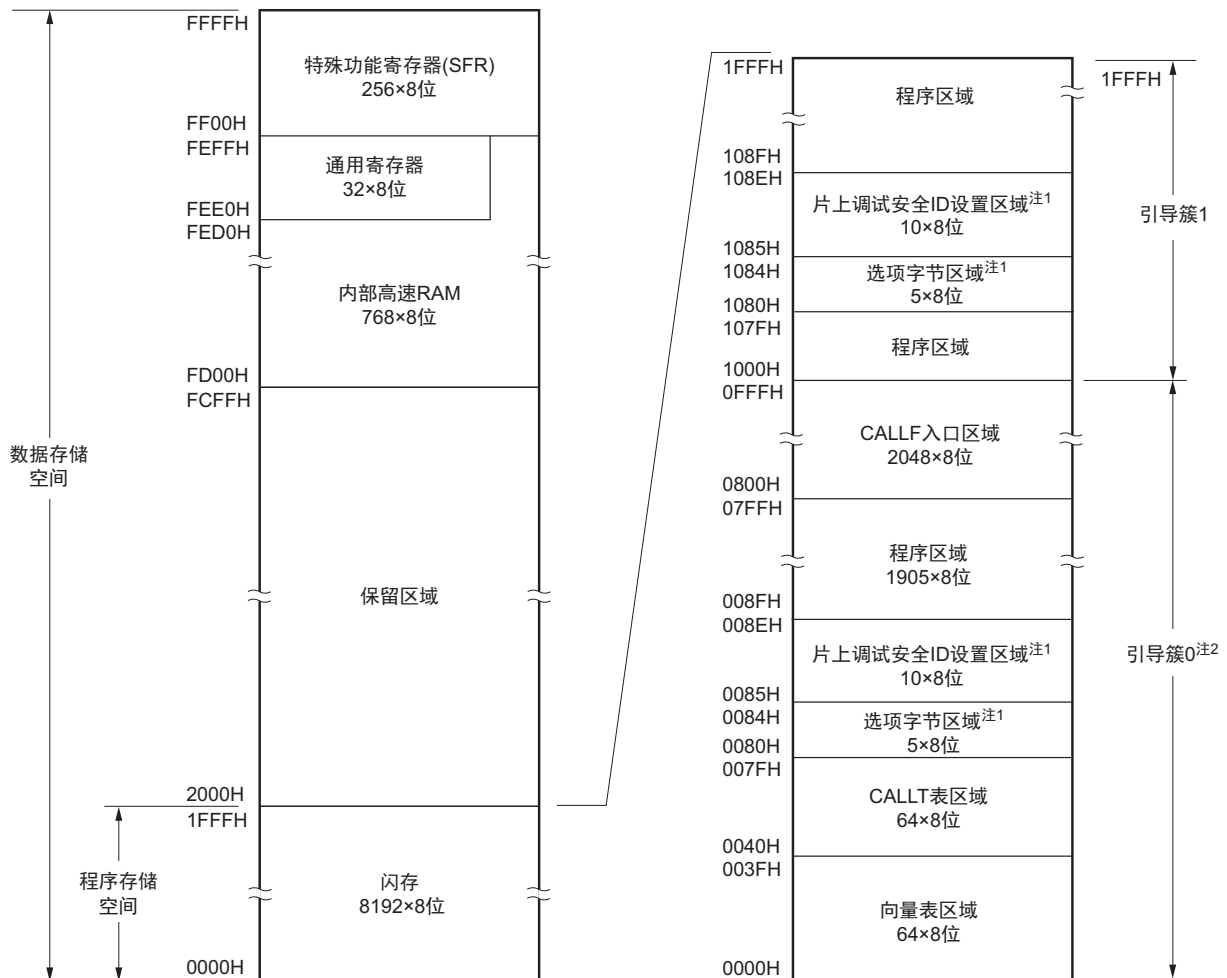


注 可通过安全设置，禁止改写引导簇 0（参照“21.6 安全设置”）。

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”。

0FFFH	块03H	1K字节
0C00H	块02H	
0800H	块01H	
0400H	块00H	
0000H		

图 3-2 存储器映射图 (R7F0C301x、R7F0C304x、R7F0C307x、R7F0C310x)

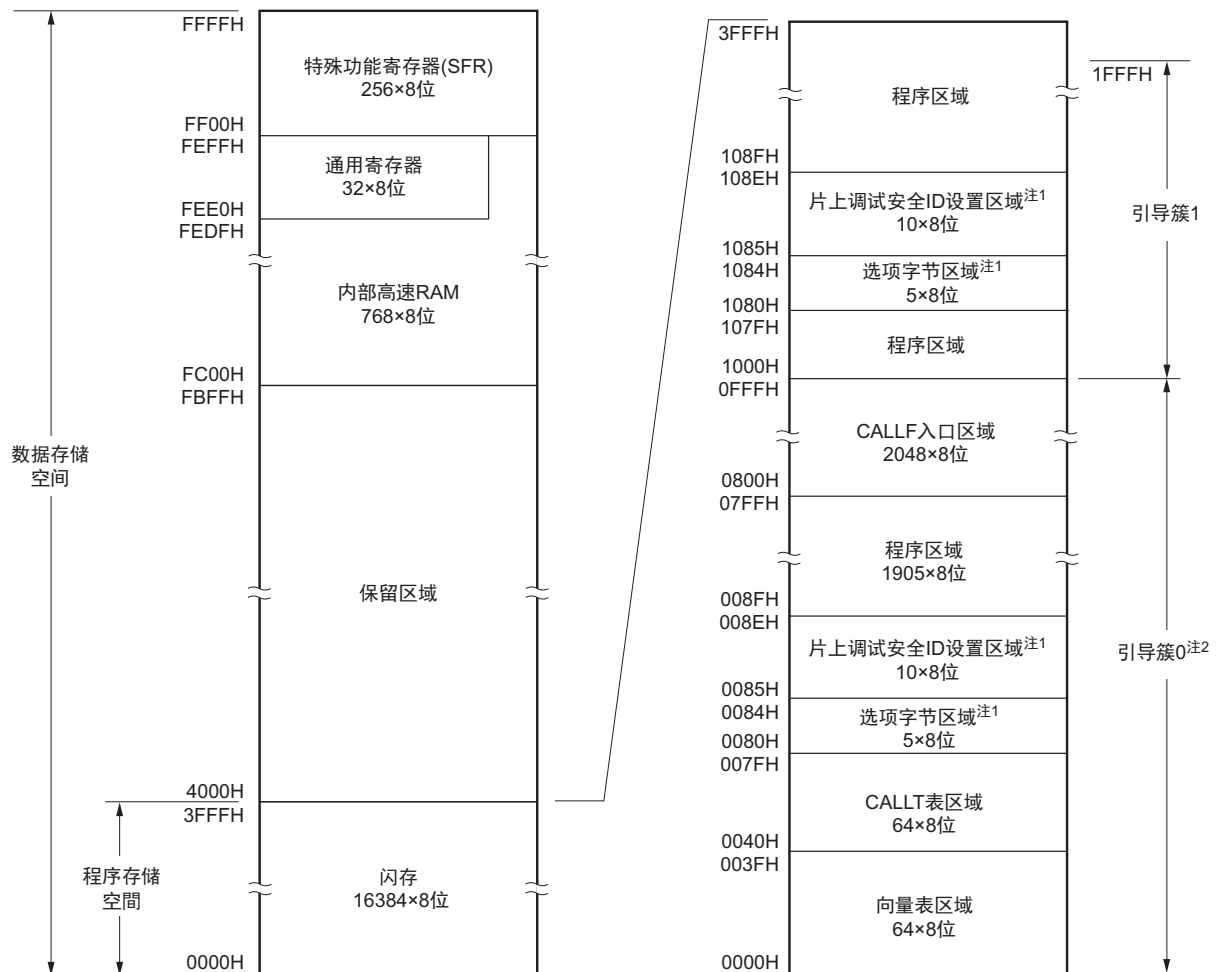


- 注 1. 不使用引导交换功能时：在 0080H ~ 0084H 设置选项字节，在 0085H ~ 008EH 设置片上调试安全 ID。  
使用引导交换功能时：在 0080H ~ 0084H 和 1080H ~ 1084H 设置选项字节，在 0085H ~ 008EH 和 1085H ~ 108EH 设置片上调试安全 ID。
2. 可通过安全设置，禁止改写引导簇 0（参照“21.6 安全设置”）。

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”。

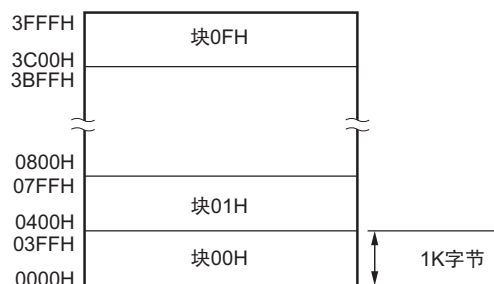


图 3-3 存储器映射图 (R7F0C302x、R7F0C305x、R7F0C308x、R7F0C311x)



- 注 1. 不使用引导交换功能时：在 0080H ~ 0084H 设置选项字节，在 0085H ~ 008EH 设置片上调试安全 ID。  
使用引导交换功能时：在 0080H ~ 0084H 和 1080H ~ 1084H 设置选项字节，在 0085H ~ 008EH 和 1085H ~ 108EH 设置片上调试安全 ID。
2. 可通过安全设置，禁止改写引导簇 0（请参照“21.6 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
0000H ~ 03FFH	00H
0400H ~ 07FFH	01H
0800H ~ 0BFFH	02H
0C00H ~ 0FFFH	03H
1000H ~ 13FFH	04H
1400H ~ 17FFH	05H
1800H ~ 1BFFH	06H
1C00H ~ 1FFFH	07H
2000H ~ 23FFH	08H
2400H ~ 27FFH	09H
2800H ~ 2BFFH	0AH
2C00H ~ 2FFFH	0BH
3000H ~ 33FFH	0CH
3400H ~ 37FFH	0DH
3800H ~ 3BFFH	0EH
3C00H ~ 3FFFH	0FH

备注 R7F0C3004、R7F0C3034、R7F0C3006、R7F0C3036、  
R7F0C3064、R7F0C3094、R7F0C3066、R7F0C3096：块号 00H ~ 03H  
R7F0C3014、R7F0C3044、R7F0C3016、R7F0C3046、  
R7F0C3074、R7F0C3104、R7F0C3076、R7F0C3106：块号 00H ~ 07H  
R7F0C3024、R7F0C3054、R7F0C3026、R7F0C3056、  
R7F0C3084、R7F0C3114、R7F0C3086、R7F0C3116：块号 00H ~ 0FH

### 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。  
R7F0C30x、R7F0C31x 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	
	结构	容量
R7F0C3004、R7F0C3034、R7F0C3006、R7F0C3036、 R7F0C3064、R7F0C3094、R7F0C3066、R7F0C3096	闪存	4096×8 位（0000H ~ 0FFFH）
R7F0C3014、R7F0C3044、R7F0C3016、R7F0C3046、 R7F0C3074、R7F0C3104、R7F0C3076、R7F0C3106、		8192×8 位（0000H ~ 1FFFH）
R7F0C3024、R7F0C3054、R7F0C3026、R7F0C3056、 R7F0C3084、R7F0C3114、R7F0C3086、R7F0C3116		16384×8 位（0000H ~ 3FFFH）

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 注	INTCMP 注
0014H	INTSR0
0016H	INTST0
001AH	INTTMH1
0020H	INTTM000
0022H	INTTM010
0024H	INTAD
002AH	INTTM51
003EH	BRK

注 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x（内置比较器产品）。

## (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 和 1080H ~ 1084H 的 5 字节区域中预备选项字节区域。不使用引导交换功能时，在 0080H ~ 0084H 设置选项字节；使用引导交换功能时，在 0080H ~ 0084H 和 1080H ~ 1084H 设置选项字节。详细内容请参照“第 20 章 选项字节”。

## (4) 片上调试安全 ID 设置区域

在 0085H ~ 008EH 和 1085H ~ 108EH 的 10 字节区域中预备片上调试安全 ID 设定区域。不使用引导交换功能时，在 0085H ~ 008EH 设置 10 字节的片上调试安全 ID；使用引导交换功能时，在 0085H ~ 008EH 和 1085H ~ 108EH 设置 10 字节的片上调试安全 ID。详细内容请参照“第 22 章 片上调试功能”。

## (5) CALLT 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

R7F0C30x、R7F0C31x 内置以下 RAM。

#### (1) 内部高速 RAM

表 3-5 内部高速 RAM 容量

产品	内部高速 RAM 容量
R7F0C3004、R7F0C3034、R7F0C3006、R7F0C3036、 R7F0C3064、R7F0C3094、R7F0C3066、R7F0C3096	512×8 位 (FD00H ~ FEFFH)
R7F0C3014、R7F0C3044、R7F0C3016、R7F0C3046、 R7F0C3074、R7F0C3104、R7F0C3076、R7F0C3106、	512×8 位 (FD00H ~ FEFFH)
R7F0C3024、R7F0C3054、R7F0C3026、R7F0C3056、 R7F0C3084、R7F0C3114、R7F0C3086、R7F0C3116	768×8 位 (FC00H ~ FEFFH)

在 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。该区域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。

### 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 数据存储器和寻址

所谓寻址，就是下一条执行指令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。

R7F0C30x、R7F0C31x 考虑到存储器寻址的可操作性等，设置了多种寻址方式，用于执行指令时作为操作对象的存储器寻址。尤其是在内部数据存储器区域，可通过配合特殊功能寄存器（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（R7F0C300x、R7F0C303x、R7F0C306x、R7F0C309x）

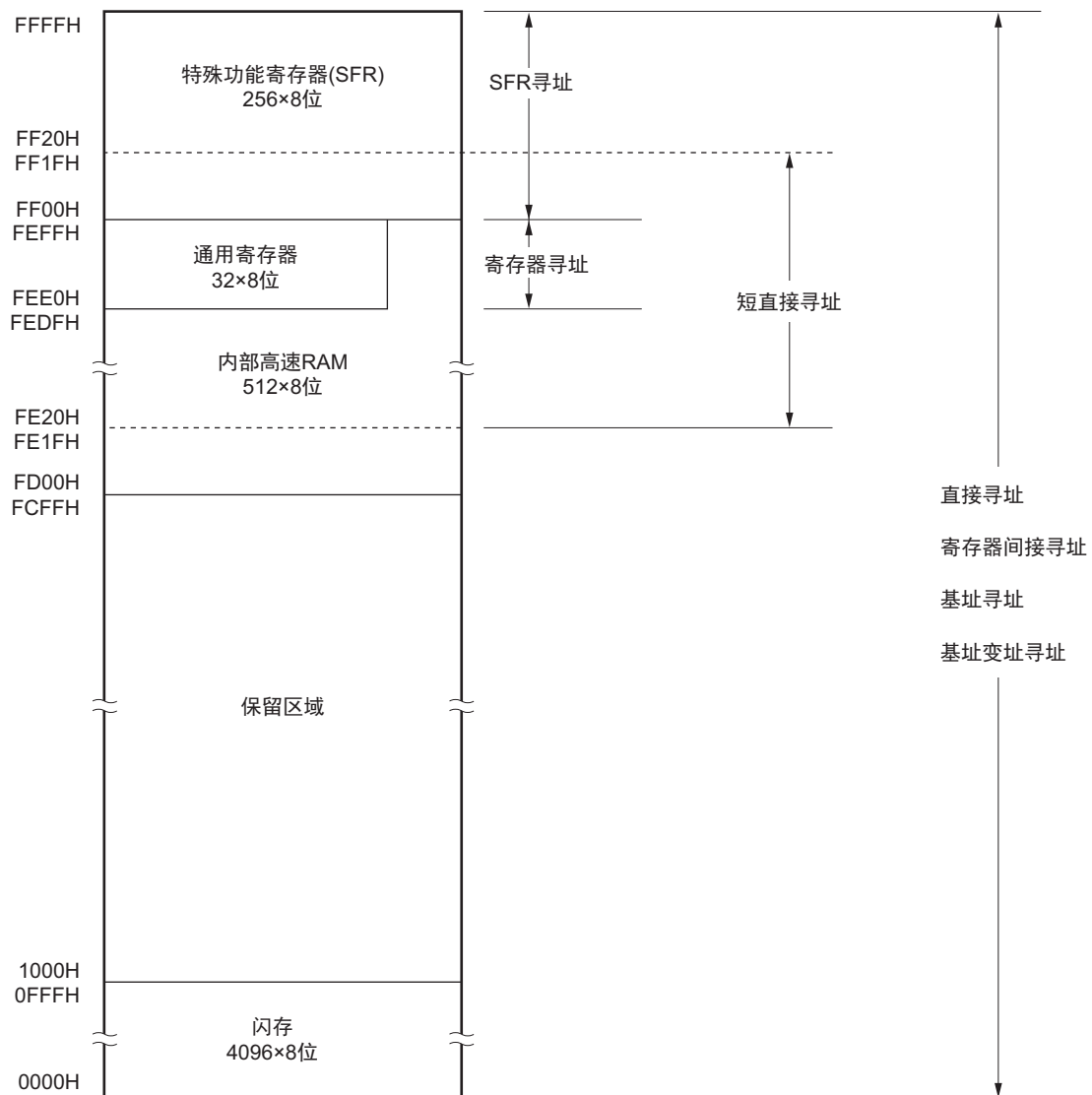


图 3-5 数据存储与寻址的对应 (R7F0C301x、R7F0C304x、R7F0C307x、R7F0C310x)

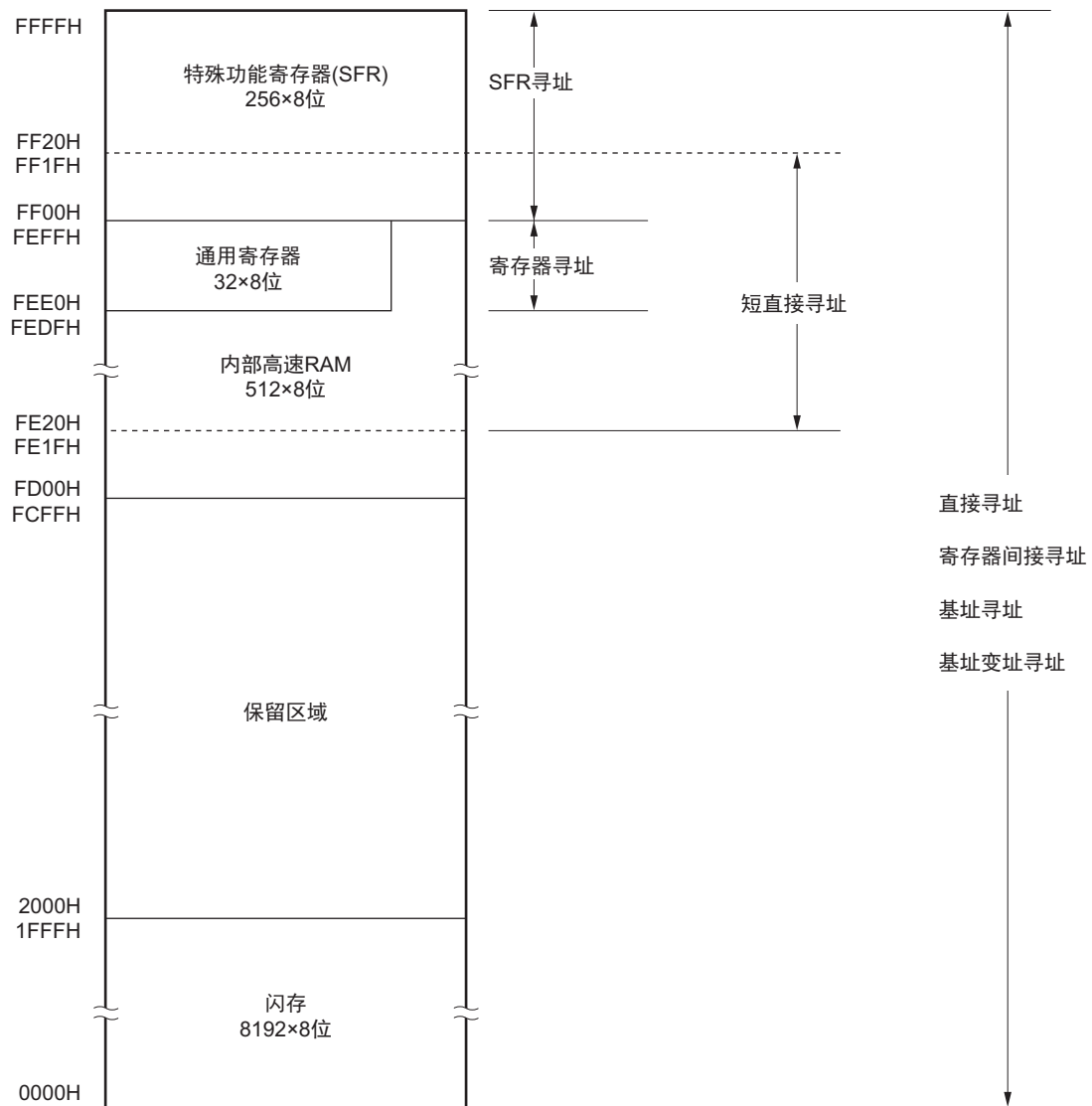
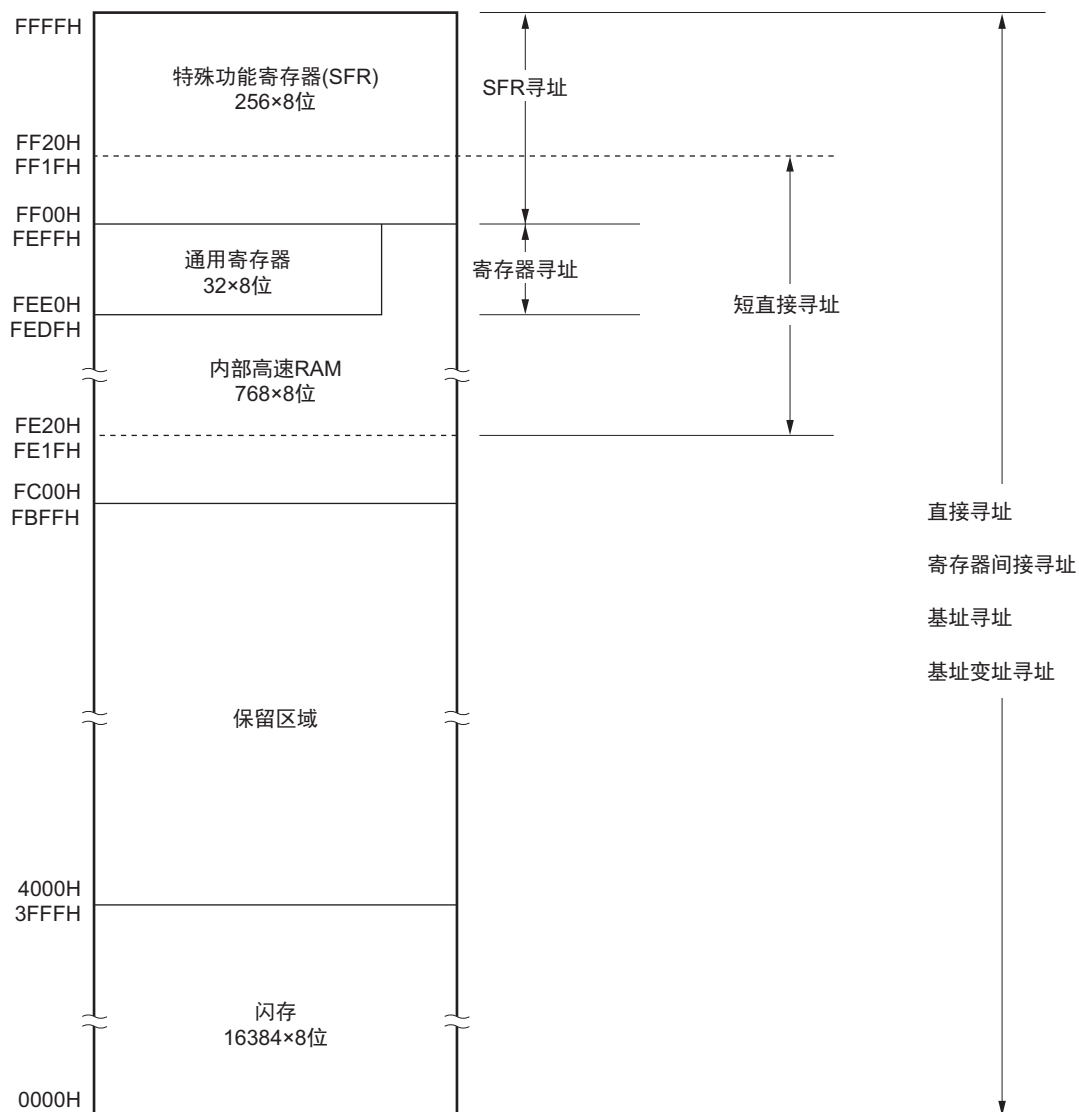


图 3-6 数据存储与寻址的对应 (R7F0C302x、R7F0C305x、R7F0C308x、R7F0C311x)



## 3.2 处理器寄存器

R7F0C30x、R7F0C31x 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）。

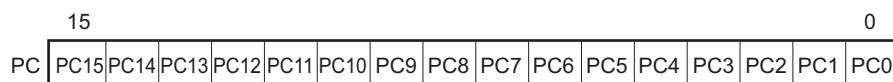
#### (1) 程序计数器（PC）

程序计数器为 16 位寄存器，用于保存下一条执行程序的地址信息。

在普通运行时，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。

在产生复位信号时，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



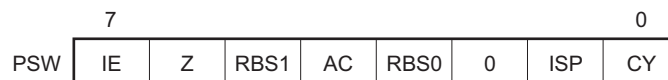
#### (2) 程序状态字（PSW）

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。

在产生复位信号时，PSW 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) (参照 “14.3(3) 优先级指定标志寄存器 (PR0L、PR0H、PR1L)”) 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。

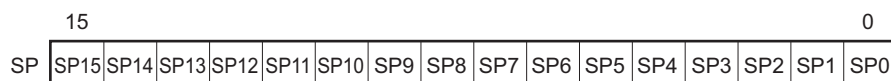
## (f) 进位标志 (CY)

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

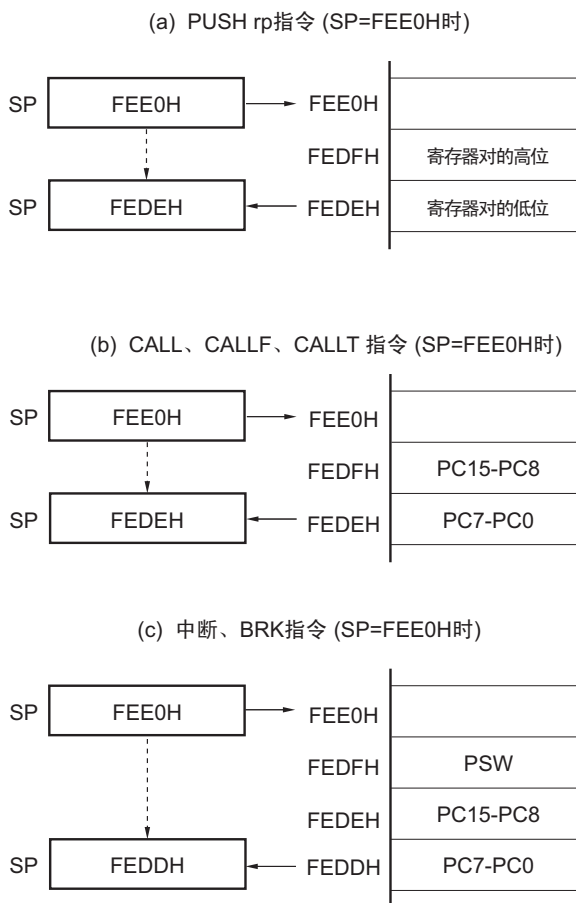
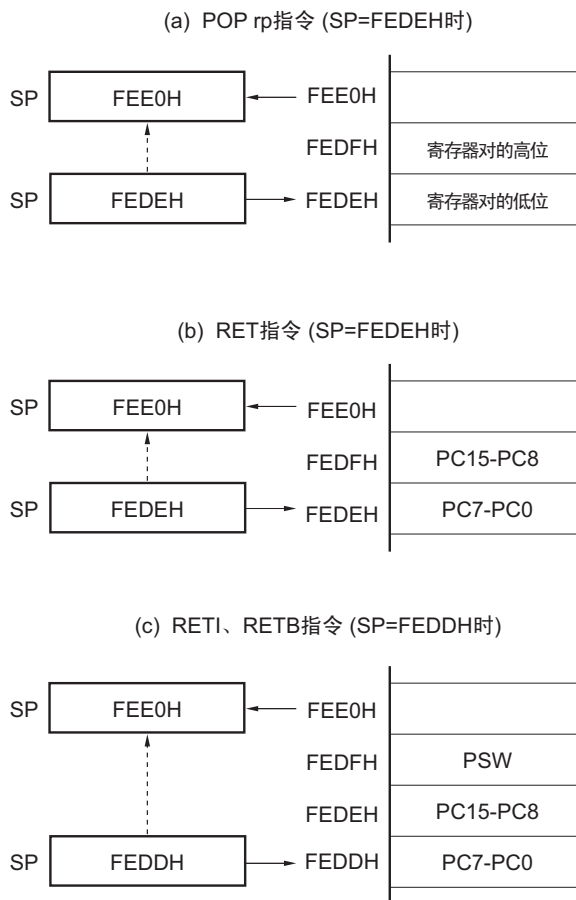


图 3-11 从堆栈存储器恢复的数据



### 3.2.2 通用寄存器

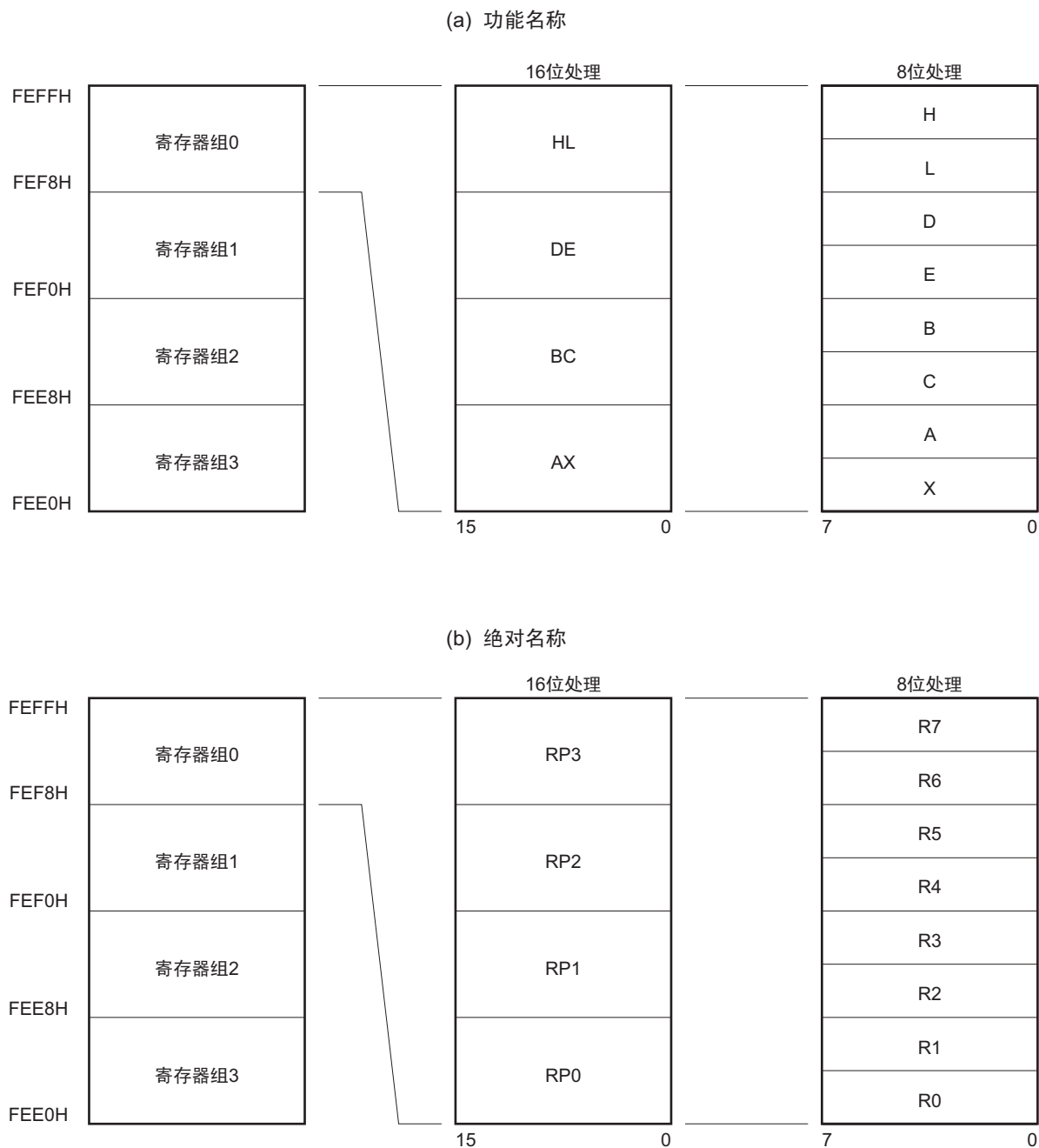
通用寄存器被映射到数据存储器的特定地址（FEE0H ~ FEFH），并且由以 8 个 8 位寄存器（X、A、C、B、E、D、L 和 H）为 1 个寄存器组的 4 个寄存器组构成。

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普通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 (1/4)

地址	符号	位号								R/W	可操作的位单位			复位时	参照页码
		7	6	5	4	3	2	1	0		1	8	16		
FF00H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF01H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF02H	P2	P27	P26	P25	P24注	P23注	P22	P21	P20	R/W	○	○	—	00H	
FF03H	P3	0	0	0	P34注	P33注	P32	P31	P30	R/W	○	○	—	00H	
FF04H ~ FF07H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF08H	AD	—	—	0	0	0	0	0	0	R	—	—	○	0000H	
FF09H	CR   ADCRH	—	—	—	—	—	—	—	—	R	—	○	—	00H	
FF0AH	RXB0	—	—	—	—	—	—	—	—	R	—	○	—	FFH	
FF0BH	TXS0	—	—	—	—	—	—	—	—	R/W	—	○	—	FFH	
FF0CH	P12	0	0	P125	0	0	P122	P121	0	R/W	○	○	—	00H	
FF0DH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF0EH	ADS	0	0	0	0	0	<ADS2>	<ADS1>	<ADS0>	R/W	○	○	—	00H	
FF0FH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF10H	TM00	—	—	—	—	—	—	—	—	R	—	—	○	0000H	
FF11H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF12H	CR000	—	—	—	—	—	—	—	—	R/W	—	—	○	0000H	
FF13H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF14H	CR010	—	—	—	—	—	—	—	—	R/W	—	—	○	0000H	
FF15H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF16H ~ FF19H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF1AH	CMP01	—	—	—	—	—	—	—	—	R/W	—	○	—	00H	
FF1BH	CMP11	—	—	—	—	—	—	—	—	R/W	—	○	—	00H	
FF1CH ~ FF1EH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF1FH	TM51	—	—	—	—	—	—	—	—	R	—	○	—	00H	
FF20H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF21H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF22H	PM2	PM27	PM26	PM25	PM24注	PM23注	PM22	PM21	PM20	R/W	○	○	—	FFH	
FF23H	PM3	1	1	1	PM34注	PM33注	PM32	PM31	PM30	R/W	○	○	—	FFH	
FF24H ~ FF2AH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

注 只限 20 引脚产品。

备注 标有“<>”的位号，在 RA78K0 中定义为保留字；在 CC78K0 中通过 #pragma sfr 指令定义为 sfr 变量。

表 3-6 特殊功能寄存器 (2/4)

地址	符号	位号								R/W	可操作的位单位			复位时	参照页码	
		7	6	5	4	3	2	1	0		1	8	16			
FF2BH	FPCTL	0	0	0	0	0	0	0	0	<FLMDP UP>	R/W	○	○	—	00H	
FF2CH	PM12	1	1	1	1	1	PM122	PM121	1		R/W	○	○	—	FFH	
FF2DH	RSTMASK	0	0	RSTM	0	0	0	0	0		R/W	○	○	—	00H	
FF2EH ~ FF32H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF33H	PU3	0	0	0	PU34 注1	PU33 注1	PU32	PU31	PU30		R/W	○	○	—	00H	
FF34H ~ FF38H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF39H	MUXSEL 注3	<INTP1 SEL1>	<INTP1 SEL0>	<TM00 SEL1>	<TM00 SEL0>	0	0	<TI010 SEL>	<TO00 SEL>		R/W	○	○	—	00H	
FF3AH 、 FF3BH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF3CH	PU12	0	0	PU125	0	0	0	0	0		R/W	○	○	—	20H	
FF3DH ~ FF47H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF48H	EGP	0	0	0	0	0	EGP2 注2	EGP1	EGP0		R/W	○	○	—	00H	
FF49H	EGN	0	0	0	0	0	EGN2 注2	EGN1	EGN0		R/W	○	○	—	00H	
FF4AH ~ FF4FH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF50H	ASIM0	<POWER0>	<TXE0>	<RXE0>	PS01	PS00	CL0	SL0	1		R/W	○	○	—	01H	
FF51H	BRGC0	TPS01	TPS00	0	MDL04	MDL03	MDL02	MDL01	MDL00		R/W	—	○	—	1FH	
FF52H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF53H	ASIS0	0	0	0	0	0	PE0	FE0	OVE0		R	—	○	—	00H	
FF54H ~ FF5FH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF60H	AMPM 注2	<OPAMP0E>	0	0	0	<OPAMP1E>	0	0	0		R/W	○	○	—	00H	
FF61H	CMPCTL 注2	<CMP0EN>	<CDFS1>	<CDFS0>	<CMPOUTEN> 注1	<CREGSEL>	<CFLG>	<COUTEN>	<CINV>		R/W	○	○	—	00H	
FF62H	CMPPC	0	0	0	0	0	0	<CMPCOMPC>	<CMPI NPC>		R/W	○	○	—	00H	
FF63H ~ FF6FH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

注 1. 只限 20 引脚产品。

2. 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x (内置运算放大器 / 比较器的产品)。

3. 只限 16 引脚产品。

备注 标有“<>”的位号, 在 RA78K0 中定义为保留字; 在 CC78K0 中通过 #pragma sfr 指令定义为 sfr 变量。

表 3-6 特殊功能寄存器 (3/4)

地址	符号	位号								R/W	可操作的位单位			复位时	参照页码
		7	6	5	4	3	2	1	0		1	8	16		
FF70H	TMHMD1	<TMHE1>	CKS12	CKS11	CKS10	TMMD11	TMMD10	<TOLEV1>	<TOEN1>	R/W	○	○	—	00H	
FF71H	TMCYC1	0	0	0	0	0	RMC1	NRZB1	<NRZ1>	R/W	○	○	—	00H	
FF72H ~ FF85H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF86H	TMC00	0	0	0	0	TMC003	TMC002	TMC001	<OVF00>	R/W	○	○	—	00H	
FF87H	PRM00	ES110	ES100	ES010	ES000	0	0	PRM001	PRM000	R/W	○	○	—	00H	
FF88H	CRC00	0	0	0	0	0	CRC002	CRC001	CRC000	R/W	○	○	—	00H	
FF89H	TOC00	0	<OSPT00>	<OSPE00>	TOC004	<LVSO0>	<LVRO0>	<TOC001>	<TOE00>	R/W	○	○	—	00H	
FF8AH ~ FF8FH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF90H	ADM	<ADCS>	0	FR2	FR1	FR0	LV1	LV0	<ADCE>	R/W	○	○	—	00H	
FF91H ~ FF96H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF97H	ADPC	ADPC7	ADPC6	ADPC5	ADPC4 注1	ADPC3 注1	ADPC2	ADPC1	ADPC0	R/W	○	○	—	00H	
FF98H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF99H	WDTE	—	—	—	—	—	—	—	—	R/W	—	○	—	1AH/ 9AH 注2	
FF9AH ~ FF9EH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF9FH	OSCCTL	<EXCLK>	<OSCEL>	0	0	0	0	0	0	R/W	○	○	—	00H	
FFA0H	RCM	<RSTS>	0	0	0	0	0	<LSR STOP>	<RSTOP>	R/W	○	○	—	80H 注3	
FFA1H	MCM	0	0	0	0	0	<XSEL>	<MCS>	<MCM0>	R/W	○	○	—	00H	
FFA2H	MOC	<MSTOP>	0	0	0	0	0	0	0	R/W	○	○	—	80H	
FFA3H	OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16	R	○	○	—	00H	
FFA4H	OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	R/W	—	○	—	05H	
FFA5H ~ FFABH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

注 1. 只限 20 引脚产品。

2. WDTE 的复位值由选项字节的设置决定。

3. 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”。

备注 标有“<>”的位号，在 RA78K0 中定义为保留字；在 CC78K0 中通过 #pragma sfr 指令定义为 sfr 变量。

表 3-6 特殊功能寄存器 (4/4)

地址	符号		位号								R/W	可操作的位单位			复位时	参照页码
			7	6	5	4	3	2	1	0		1	8	16		
FFACH	RESF		0	0	0	WDTRF	0	0	0	LVIRF	R	—	○	—	00H 注1	
FFADH ~ FFB0H	—		—	—	—	—	—	—	—	—	—	—	—	—	—	—
FFB1H	CR51		—	—	—	—	—	—	—	—	R/W	—	○	—	00H	
FFB2H	TCL51		0	0	0	0	0	TCL512	TCL511	TCL510	R/W	○	○	—	00H	
FFB3H	TMC51		<TCE51>	TMC516	0	0	0	0	TMC511	0	R/W	○	○	—	00H	
FFB4H ~ FFBDH	—		—	—	—	—	—	—	—	—	—	—	—	—	—	—
FFBEH	LVIM		<LVION>	0	0	0	0	0	<LVIMD>	<LVIF>	R/W	○	○	—	00H 注2	
FFBFH	LVIS		0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0	R/W	○	○	—	00H 注3	
FFC0H ~ FFDFH	—		—	—	—	—	—	—	—	—	—	—	—	—	—	—
FFE0H	IF0	IF0L	0	0	0	0	<CMPIF> 注5	<PIF1>	<PIF0>	<LVIF>	R/W	○	○	○	00H	
FFE1H		IF0H	<TMIF01 0>	<TMIF0 00>	0	0	<TMIFH1>	0	<STIF0>	<SRIF0>	R/W	○	○	○	00H	
FFE2H	IF1	IF1L	0	0	0	0	<TMIF51>	0	0	<ADIF>	R/W	○	○	○	00H	
FFE3H		—	0	0	0	0	0	0	0	0	—	—	—	○	00H	
FFE4H	MK0	MK0L	1	1	1	1	<CMPMK> 注5	<PMK1>	<PMK0>	<LVIMK>	R/W	○	○	○	FFH	
FFE5H		MK0H	<TMMK0 10>	<TMMK 000>	1	1	<TMMKH1>	1	<STMK0>	<SRMK0>	R/W	○	○	○	FFH	
FFE6H	MK1	MK1L	1	1	1	1	<TMMK51>	1	1	<ADMK>	R/W	○	○	○	FFH	
FFE7H		—	1	1	1	1	1	1	1	1	—	—	—	○	FFH	
FFE8H	PR0	PR0L	1	1	1	1	<CMPPR> 注5	<PPR1>	<PPR0>	<LVIPR>	R/W	○	○	○	FFH	
FFE9H		PR0H	<TMPR0 10>	<TMPR 000>	1	1	<TMPRH1>	1	<STPR0>	<SRPR0>	R/W	○	○	○	FFH	
FFEAH	PR1	PR1L	1	1	1	1	<TMPR51>	1	1	<ADPR>	R/W	○	○	○	FFH	
FFEBH		—	1	1	1	1	1	1	1	1	—	—	—	○	FFH	
FFECH ~ FFEFH	—		—	—	—	—	—	—	—	—	—	—	—	—	—	—
FFF0H	IMS		RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0	R/W	—	○	—	CFH 注4	
FFF1H ~ FFFAH	—		—	—	—	—	—	—	—	—	—	—	—	—	—	—
FFFBH	PCC		0	0	0	0	0	PCC2	PCC1	PCC0	R/W	○	○	—	01H	

注 1. RESF 的复位值因复位源而改变。

2. LVIM 的复位值因复位源和选项字节的设置而改变。

3. LVIS 的复位值因复位源而改变。

4. 因为复位时的 ROM 区域的设置不定，所以在复位解除后，必须对各产品设置表 3-1 所示的值。

5. 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x（内置比较器的产品）。

备注 标有“<>”的位号，在 RA78K0 中定义为保留字；在 CC78K0 中通过 #pragma sfr 指令定义为 sfr 变量。



### 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系列用户手册 指令（U12326C）”）。

#### 3.3.1 相对寻址

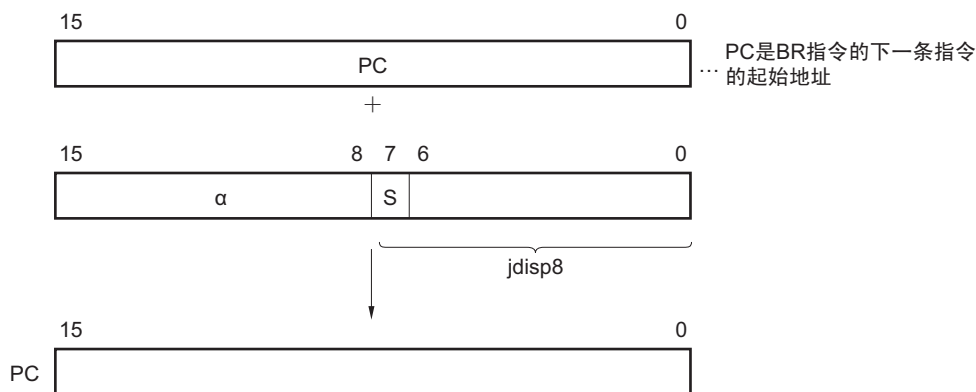
##### [功能]

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。

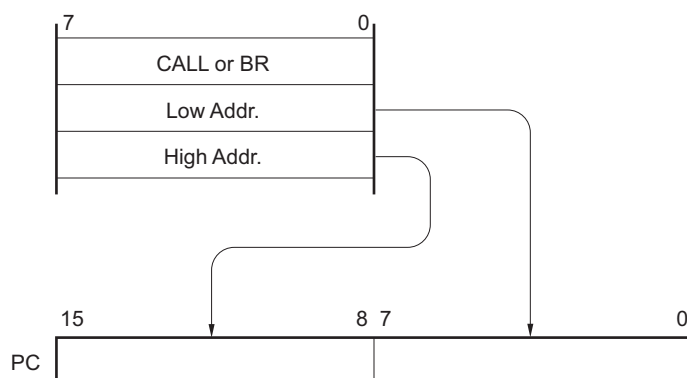
在执行 CALL !addr16、BR !addr16 或者 CALLF !addr11 指令时，进行立即寻址。

通过 CALL !addr16 指令和 BR !addr16 指令可转移到全部的程序存储空间。

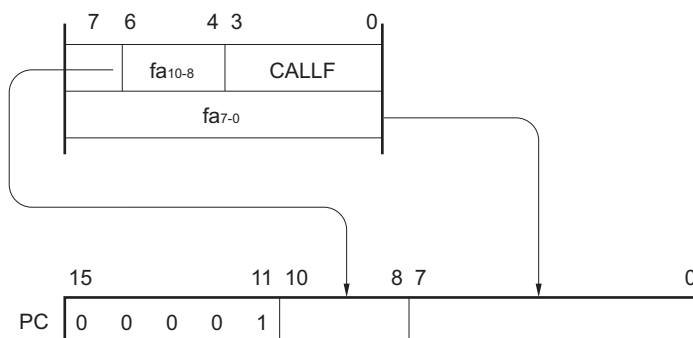
通过 CALLF !addr11 指令可转移到 0800H ~ 0FFFH 的区域。

#### [ 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

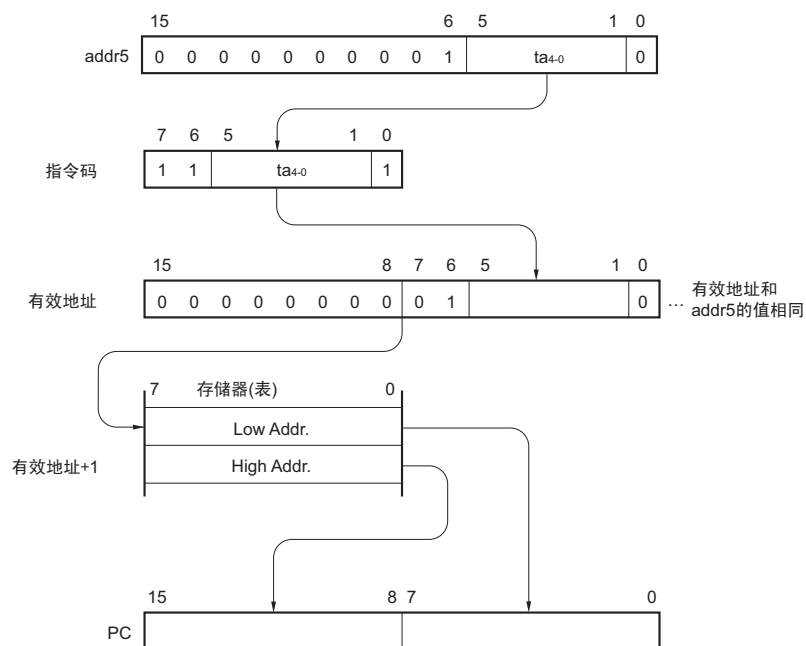
#### [ 功能 ]

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。

#### [ 图示 ]



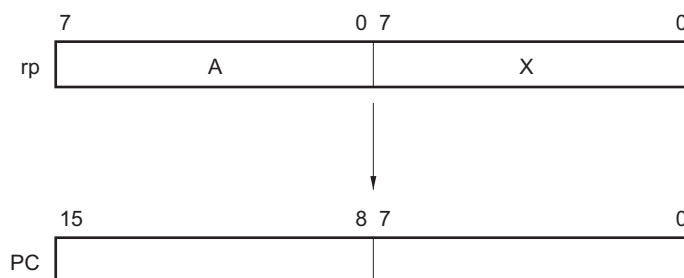
### 3.3.4 寄存器寻址

#### [ 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 R7F0C30x、R7F0C31x 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。

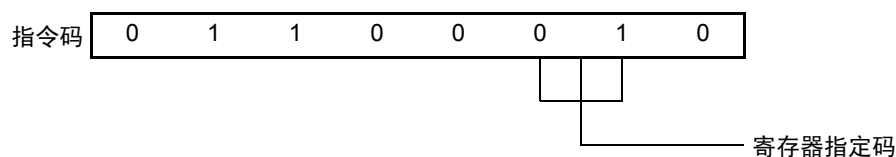
#### [ 操作数格式 ]

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

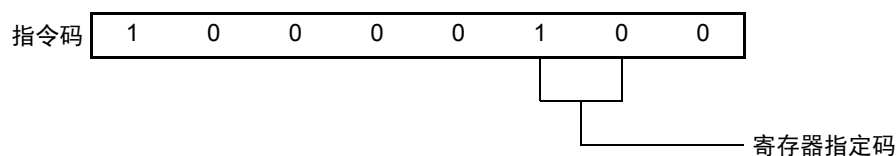
“r”和“rp”可用功能名称（X、A、C、B、E、D、L、H、AX、BC、DE 和 HL）和绝对名称（R0～R7 和 RP0～RP3）进行描述。

#### [ 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 ]

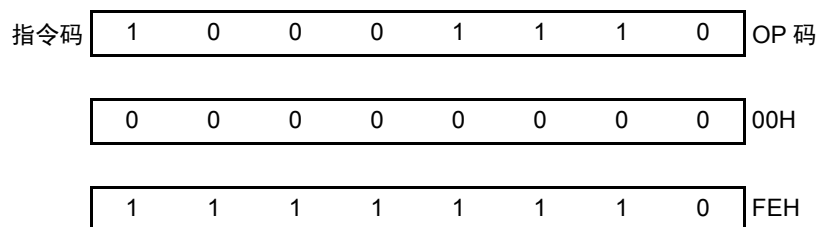
直接寻址是指对指令字中的立即数所示的存储器进行直接寻址。  
可对全部的存储空间进行寻址。

#### [ 操作数格式 ]

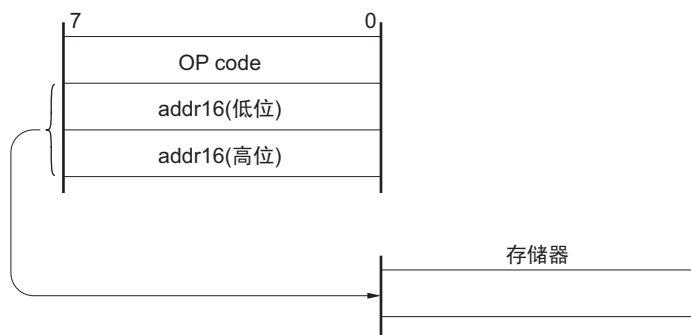
标识符	描述方法
addr16	标号或 16 位立即数

#### [ 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 位立即数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，并且内部高速 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 ]

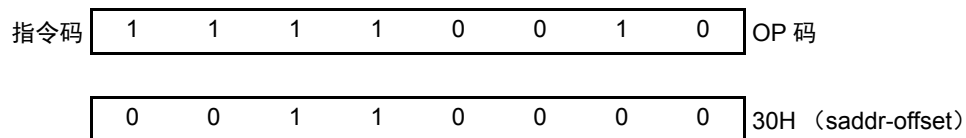
标识符	描述方法
saddr	标号或者表示 FE20H ~ FF1FH 的立即数
saddrp	标号或者表示 FE20H ~ FF1FH 的立即数 (只限偶数地址)

#### [ 描述示例 ]

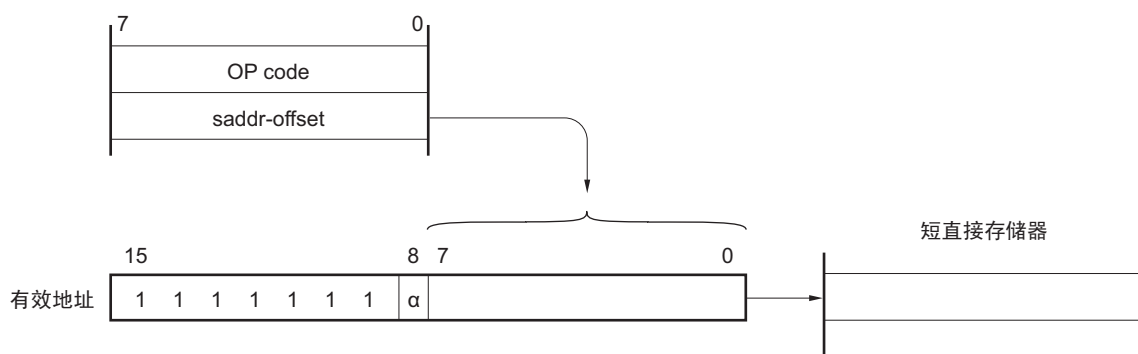
LB1 EQU 0FE30H ; 由 LB1 定义 FE30H。

:

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， $\alpha=0$ 。

8 位立即数为 00H ~ 1FH 时， $\alpha=1$ 。

### 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。

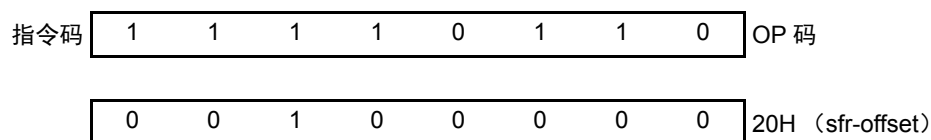
该寻址适用于 FF00H ~ FFCFH 和 FFE0H ~ FFFFH 的 240 字节区域。但是，映射在 FF00H ~ FF1FH 的特殊功能寄存器（SFR）的寻址，也能通过短直接寻址进行。

#### [ 操作数格式 ]

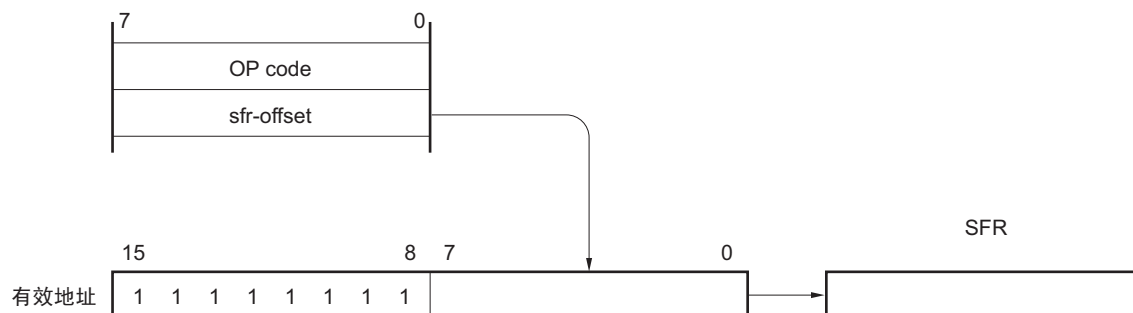
标识符	描述方法
sfr	特殊功能寄存器名称
sfrp	可 16 位操作的特殊功能寄存器名称（只限偶数地址）

#### [ 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。

#### [ 操作数格式 ]

标识符	描述方法
—	[DE], [HL]

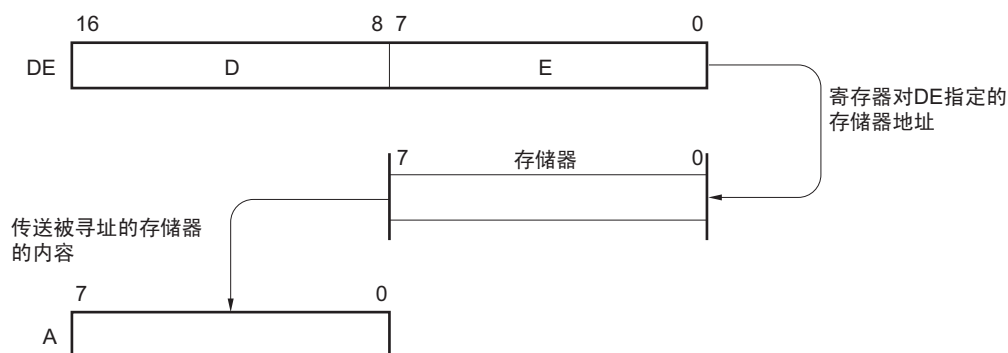
#### [ 描述示例 ]

MOV A, [DE] ; 当选择 [DE] 作为寄存器对时

指令码 

1	0	0	0	0	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。

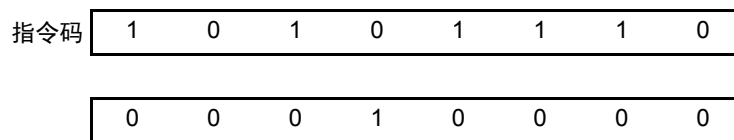
可对全部的存储空间进行寻址。

#### [ 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。

#### [ 操作数格式 ]

标识符	描述方法
—	[HL+B], [HL+C]

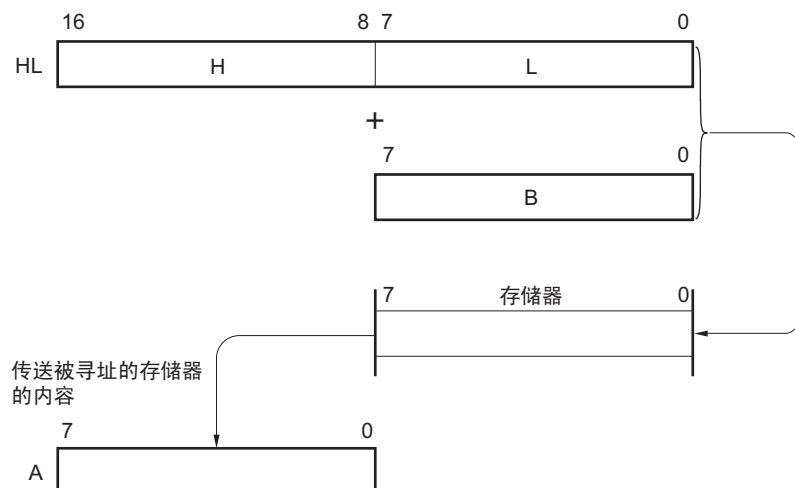
#### [ 描述示例 ]

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。

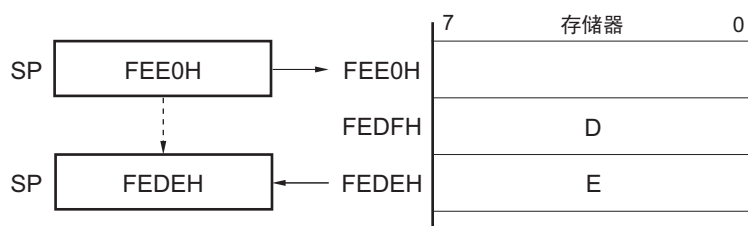
#### [ 描述示例 ]

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 / 输出缓冲器电源有  $AV_{REF}$  和  $V_{DD}$  两种。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
$AV_{REF}$	P20 ~ P27 注
$V_{DD}$	P20 ~ P27 注以外的引脚

注 16 引脚产品：P20 ~ P22、P25 ~ P27

20 引脚产品：P20 ~ P27

R7F0C30x、R7F0C31x 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 和表 4-3 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（16 引脚产品）

功能名称	输入 / 输出	功能	复位时	复用功能
P20	输入 / 输出	端口 2 6 位输入 / 输出端口 可以 1 位单位指定输入 / 输出	模拟输入	ANI0/AMP0- 注
P21				ANI1/AMP0OUT 注
P22				ANI2/AMP0+ 注
P25				ANI5/AMP1- 注
P26				AN16/AMP1OUT 注
P27				AN17/AMP1+ 注
P30	输入 / 输出	端口 3 3 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	TOH1/TI51/INTP0
P31				模拟输入
P32			RxD0/<TI010>/ CMPIN 注	
P121	输入 / 输出	端口 12 P121、P122 为 2 位输入 / 输出端口 P125 为 1 位输入专用端口	输入端口	X1/<TI000>/<INTP1>/ TOOLC0
P122				X2/EXCLK/TOOLD0
P125	输入	仅 P125 可通过软件设置，使用内部上拉电阻。	复位输入	<TI000>/<INTP1>/ RESET

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114（内置运算放大器 / 比较器的产品）。

备注 可通过设置端口复用切换控制寄存器（MUXSEL）指定上表 <> 内的功能。

表 4-3 端口功能 (20 引脚产品)

功能名称	输入 / 输出	功能	复位时	复用功能
P20	输入 / 输出	端口 2 8 位输入 / 输出端口 可以 1 位单位指定输入 / 输出	模拟输入	ANI0/AMP0- 注
P21				ANI1/AMP0OUT 注
P22				ANI2/AMP0+ 注
P23				ANI3
P24				ANI4
P25				AMP1- 注
P26				AMP1OUT 注
P27				AMP1+ 注
P30	输入 / 输出	端口 3 5 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	TOH1/TI51/INTP0
P31			模拟输入	TxD0/CMPCOM 注
P32				RxD0/CMPIN 注
P33			输入端口	TI000/INTP1
P34				TO00/TI010/ CMPOUT 注
P121	输入 / 输出	端口 12 P121、P122 为 2 位输入 / 输出端口	输入端口	X1/TOOLC0
P122				X2/EXCLK/TOOLD0
P125	输入	P125 为 1 位输入专用端口 仅 P125 可通过软件设置, 使用内部上拉电阻。	复位输入	RESET

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置运算放大器 / 比较器的产品)。

## 4.2 端口结构

端口由以下硬件构成。

表 4-4 端口结构

项目	结构
控制寄存器	端口模式寄存器 (PMxx)：PM2、PM3、PM12 端口寄存器 (Pxx)：P2、P3、P12 上拉电阻选择寄存器 (PUxx)：PU3、PU12 复位引脚模式寄存器 (RSTMASK) A/D 端口配置寄存器 (ADPC) 端口复用切换控制寄存器 (MUXSEL) (只限 16 引脚产品)
端口	16 引脚产品：总计 12 个 (CMOS 输入 / 输出：11 个、CMOS 输入：1 个) 20 引脚产品：总计 16 个 (CMOS 输入 / 输出：15 个、CMOS 输入：1 个)
上拉电阻	16 引脚产品：总计 4 个 20 引脚产品：总计 6 个

### 4.2.1 端口 2

#### (1) 16 引脚产品

16 引脚产品
P20/ANI0/AMP0- 注
P21/ANI1/AMP0OUT 注
P22/ANI2/AMP0+ 注
—
—
P25/ANI5/AMP1- 注
P26/ANI6/AMP1OUT 注
P27/ANI7/AMP1+ 注

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 (内置运算放大器的产品)。

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。

作为复用功能，该端口还可用作 A/D 转换器的模拟输入和运算放大器的输入 / 输出。

使用 P20/AMP0-/ANI0 ~ P22/AMP0+/ANI2、P25/AMP1-/ANI5 ~ P27/AMP1+/ANI7 时，可根据使用的引脚功设置寄存器 (参照表 4-5、表 4-6)。

将 P20/AMP0-/ANI0 ~ P22/AMP0+/ANI2、P25/AMP1-/ANI5 ~ P27/AMP1+/ANI7 用作数字输入或者数字输出时，推荐从离  $AV_{REF}$  最远的引脚 (P27/ANI7) 开始执行；将 P20/AMP0-/ANI0 ~ P22/AMP0+/ANI2、P25/AMP1-/ANI5 ~ P27/AMP1+/ANI7 用作模拟输入时，推荐从离  $V_{SS}$  最近的引脚 (P27/ANI7) 开始执行。

在产生复位信号时，端口 2 变为模拟输入。

端口 2 的框图如图 4-1 ~ 图 4-3 所示。

注意 在端口 2 用作数字端口时，将  $AV_{REF}$  设置为与  $V_{DD}$  相同的电位。

表 4-5 P20/ANI0/AMP0-、P22/ANI2/AMP0+、P25/ANI5/AMP1-、P27/ANI7/AMP1+ 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMPmE 位 (m=0、1)	ADS 寄存器 (n=0、2、5、7)	P20/ANI0/AMP0-、 P22/ANI2/AMP0+、 P25/ANI5/AMP1-、 P27/ANI7/AMP1+ 引脚
选择数字输入 / 输出	输入模式	—	选择 ANIn	禁止设置
			不选择 ANIn	数字输入
	输出模式	—	选择 ANIn	禁止设置
			不选择 ANIn	数字输出
选择模拟输入	输入模式	0	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)
		1	选择 ANIn	模拟输入 (A/D 转换对象)、 运算放大器 m 输入
			不选择 ANIn	运算放大器 m 输入
	输出模式	—	—	禁止设置

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMPmE: 运算放大器控制寄存器 (AMPM) 的 bit3 和 bit7。  
 ADS: 模拟输入通道指定寄存器

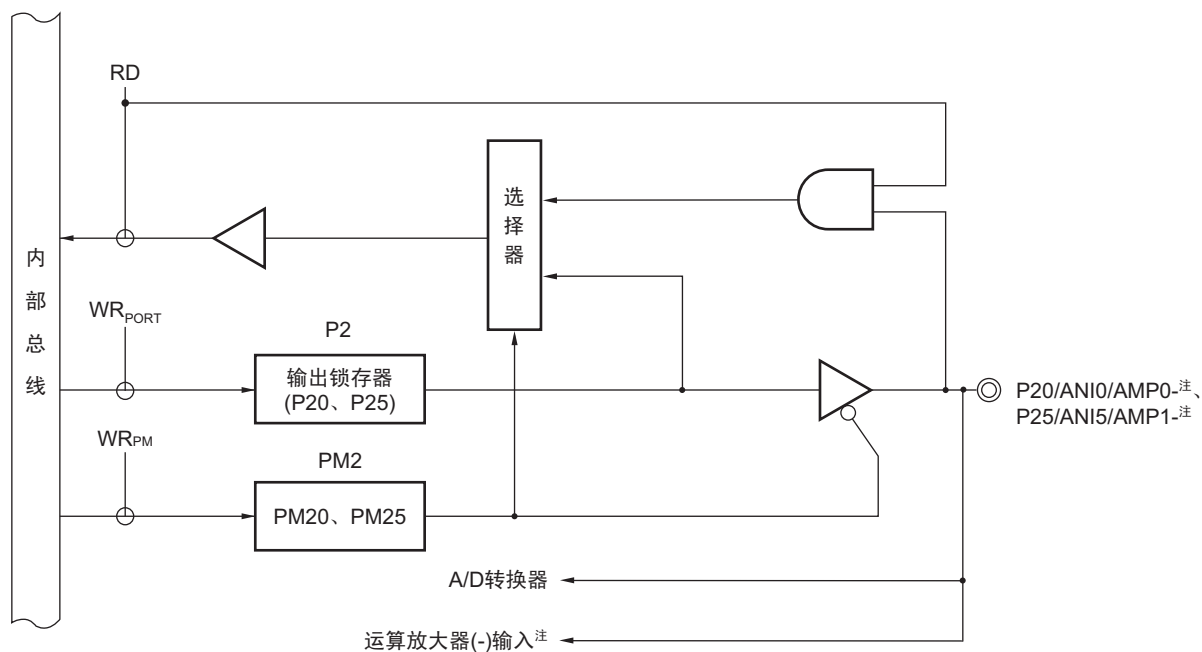
表 4-6 P21/ANI1/AMP0OUT、P26/ANI6/AMP1OUT 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMPmE 位 (m=0、1)	ADS 寄存器 (n=1、6)	P21/ANI1/AMP0OUT、 P26/ANI6/AMP1OUT 引脚
选择数字输入 / 输出	输入模式	0	选择 ANIn	禁止设置
			不选择 ANIn	数字输入
		1	—	禁止设置
			输出模式	0
	不选择 ANIn	数字输出		
	1	—	禁止设置	
选择模拟输入 / 输出		输入模式	0	选择 ANIn
	不选择 ANIn			模拟输入 (非 A/D 转换对象)
	1		选择 ANIn	运算放大器 m 输出 (A/D 转换对象)
			不选择 ANIn	运算放大器 m 输出 (非 A/D 转换对象)
	输出模式	—	—	禁止设置

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMPmE: 运算放大器控制寄存器 (AMPM) 的 bit3 和 bit7。  
 ADS: 模拟输入通道指定寄存器



图 4-1 P20、P25 的框图



P2: 端口寄存器 2

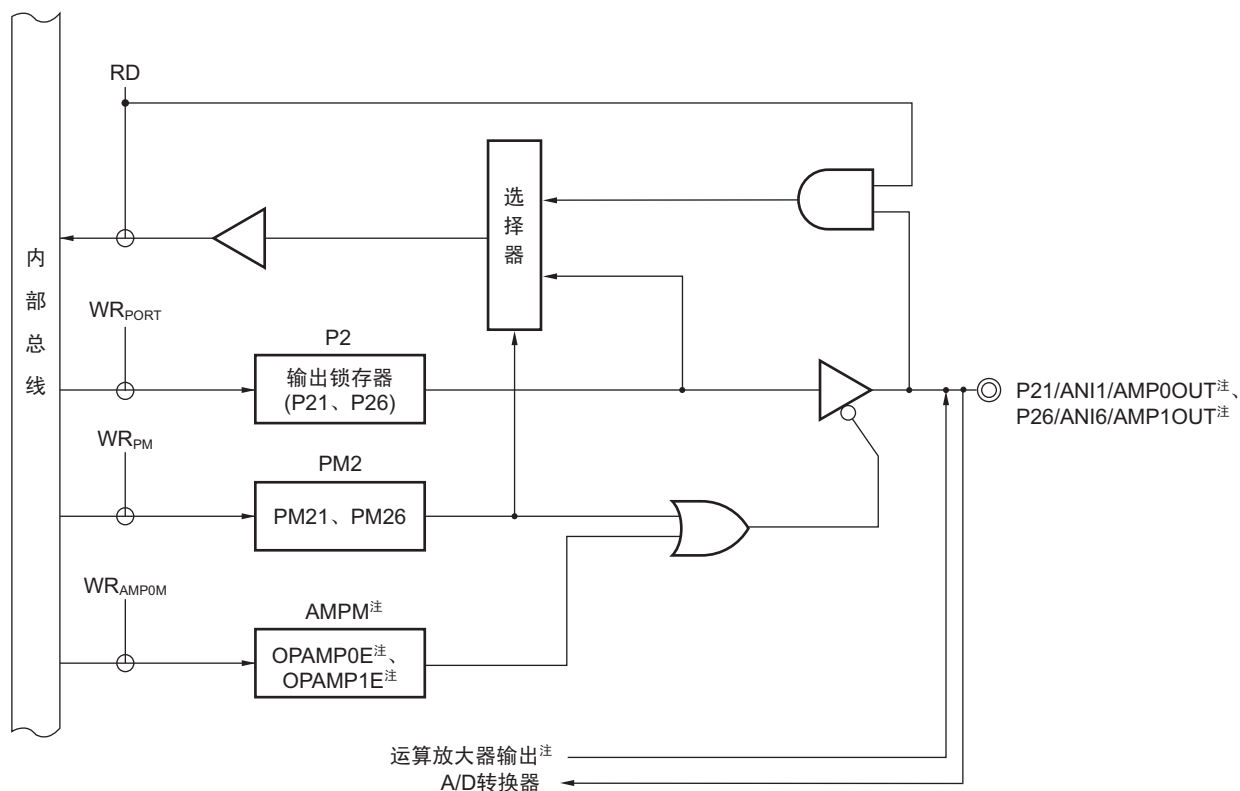
PM2: 端口模式寄存器 2

RD: 读信号

WR<sub>xx</sub>: 写信号

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 (内置运算放大器的产品)。

图 4-2 P21、P26 的框图



P2: 端口寄存器 2

PM2: 端口模式寄存器 2

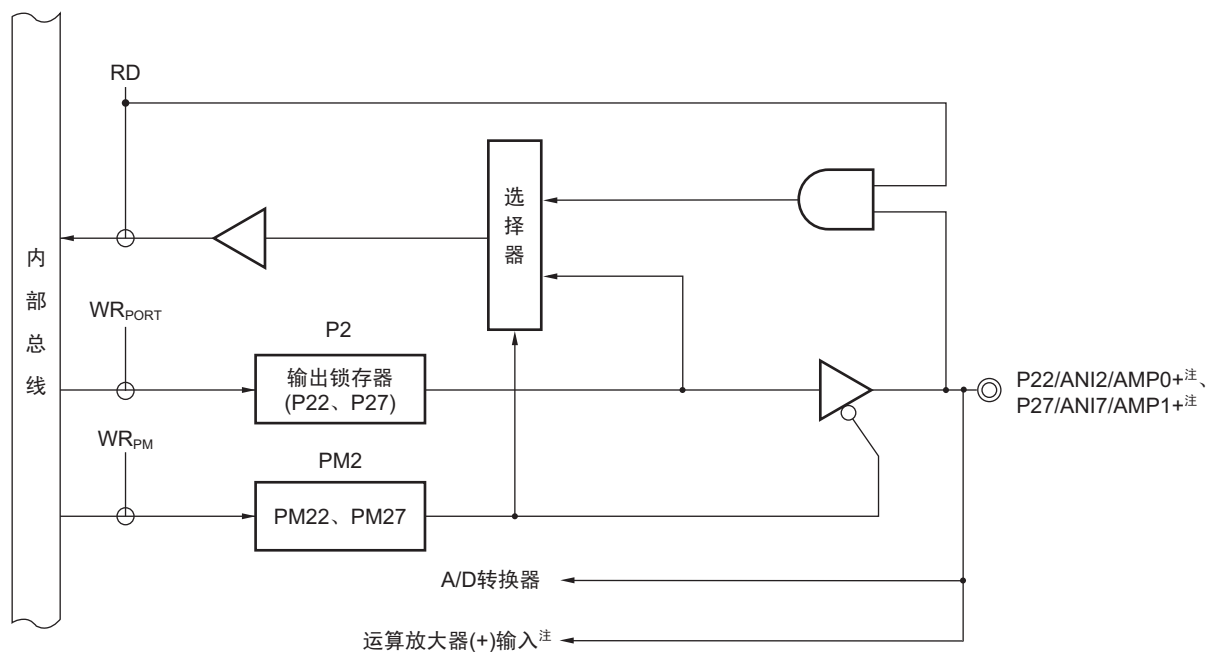
AMPM: 运算放大器控制寄存器

RD: 读信号

WRxx: 写信号

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114（内置运算放大器的产品）。

图 4-3 P22、P27 的框图



P2: 端口寄存器 2

PM2: 端口模式寄存器 2

RD: 读信号

WR<sub>xx</sub>: 写信号

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 (内置运算放大器的产品)。

## (2) 20 引脚产品

20 引脚产品
P20/ANI0/AMP0- 注
P21/ANI1/AMP0OUT 注
P22/ANI2/AMP0+ 注
P23/ANI3
P24/ANI4
P25/AMP1- 注
P26/AMP1OUT 注
P27/AMP1+ 注

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116（内置运算放大器的产品）。

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。

作为复用功能，该端口还可用作 A/D 转换器的模拟输入和运算放大器的输入 / 输出。

使用 P20/AMP0-/ANI0 ~ P24/ANI4 时，可根据使用的引脚功设置寄存器（参照表 4-7 ~ 表 4-11）。

将 P20/AMP0-/ANI0 ~ P24/ANI4 用作数字输入或者数字输出时，推荐从离  $AV_{REF}$  最远的引脚（P24/ANI4）开始执行；将 P20/AMP0-/ANI0 ~ P24/ANI4 用作模拟输入时，推荐从离  $V_{SS}$  最近的引脚（P24/ANI4）开始执行。

在产生复位信号时，端口 2 变为模拟输入。

端口 2 的框图如图 4-4 ~ 图 4-10 所示。

注意 在端口 2 用作数字端口时，将  $AV_{REF}$  设置为与  $V_{DD}$  相同的电位。

表 4-7 P20/ANI0/AMP0-、P22/ANI2/AMP0+ 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMP0E 位	ADS 寄存器 (n=0、2)	P20/ANI0/AMP0-、 P22/ANI2/AMP0+ 引脚
选择数字输入 / 输出	输入模式	—	选择 ANIn	禁止设置
			不选择 ANIn	数字输入
	输出模式	—	选择 ANIn	禁止设置
			不选择 ANIn	数字输出
选择模拟输入	输入模式	0	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）
		1	选择 ANIn	模拟输入（A/D 转换对象）、 运算放大器 0 输入
			不选择 ANIn	运算放大器 0 输入
	输出模式	—	—	禁止设置

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMP0E: 运算放大器控制寄存器（AMPM）的 bit7。  
 ADS: 模拟输入通道指定寄存器

表 4-8 P21/ANI1/AMP0OUT 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMP0E 位	ADS 寄存器	P21/ANI1/AMP0OUT 引脚
选择数字输入 / 输出	输入模式	0	选择 ANI1	禁止设置
			不选择 ANI1	数字输入
		1	—	禁止设置
	输出模式	0	选择 ANI1	禁止设置
			不选择 ANI1	数字输出
		1	—	禁止设置
选择模拟输入 / 输出	输入模式	0	选择 ANI1	模拟输入 (A/D 转换对象)
			不选择 ANI1	模拟输入 (非 A/D 转换对象)
		1	选择 ANI1	运算放大器 0 输出 (A/D 转换对象)
			不选择 ANI1	运算放大器 0 输出 (非 A/D 转换对象)
	输出模式	—	—	禁止设置

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMP0E: 运算放大器控制寄存器 (AMP0) 的 bit7  
 ADS: 模拟输入通道指定寄存器

表 4-9 P23/ANI3、P24/ANI4 引脚功能的设置

ADPC 寄存器	PM2 寄存器	ADS 寄存器 (n=3、4)	P23/ANI3、 P24/ANI4 引脚
选择数字输入 / 输出	输入模式	选择 ANIn	禁止设置
		不选择 ANIn	数字输入
	输出模式	选择 ANIn	禁止设置
		不选择 ANIn	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	—	禁止设置

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 ADS: 模拟输入通道指定寄存器

表 4-10 P25/AMP1-、P27/AMP1+ 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMP1E 位	P25/AMP1-、P27/AMP1+ 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	0	模拟输入（非 A/D 转换对象）
		1	运算放大器 1 输入
	输出模式	—	禁止设置

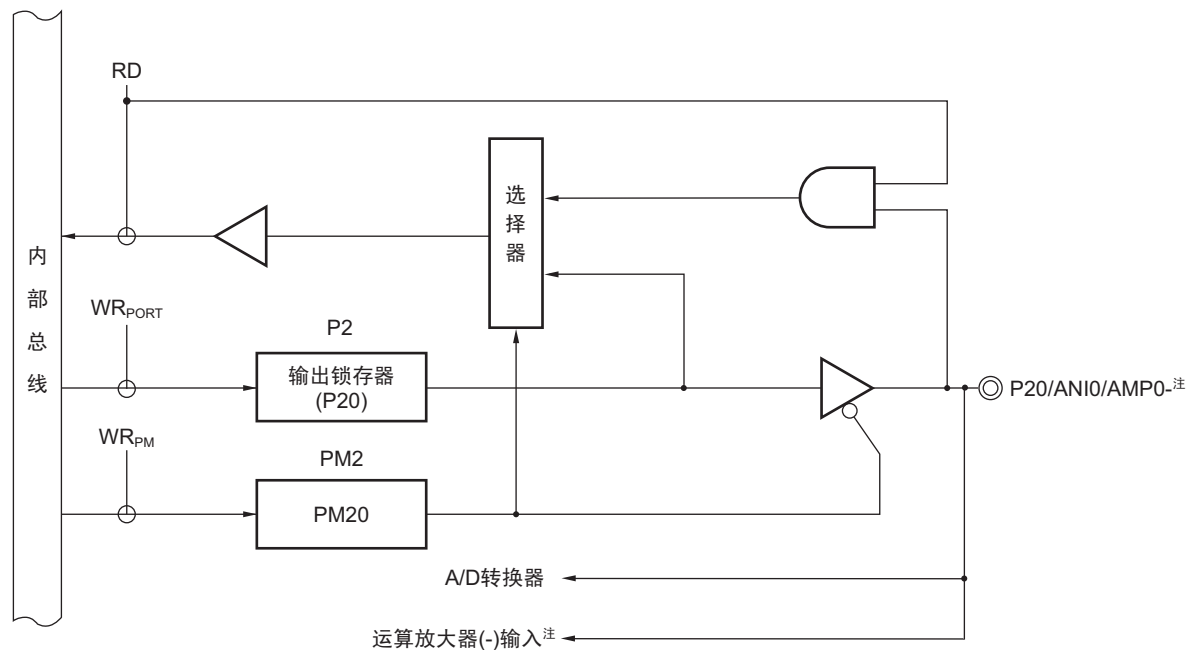
备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMP1E: 运算放大器控制寄存器 (AMPM) 的 bit3  
 ADS: 模拟输入通道指定寄存器

表 4-11 P26/AMP1OUT 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMP1E 位	P26/AMP1OUT 引脚
选择数字输入 / 输出	输入模式	0	数字输入
		1	禁止设置
	输出模式	0	数字输出
		1	禁止设置
选择模拟输入 / 输出	输入模式	0	模拟输入（非 A/D 转换对象）
		1	运算放大器 1 输出
	输出模式	—	禁止设置

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMP1E: 运算放大器控制寄存器 (AMPM) 的 bit3

图 4-4 P20 的框图



P2: 端口寄存器 2

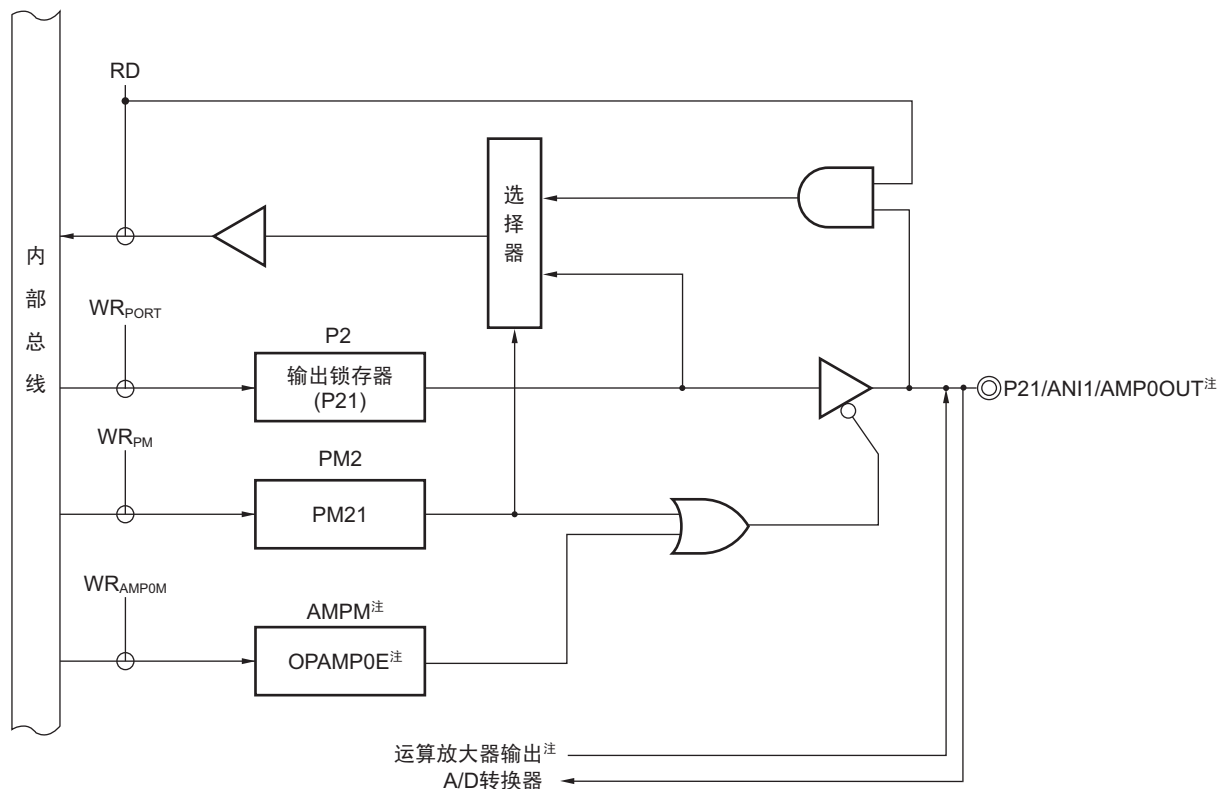
PM2: 端口模式寄存器 2

RD: 读信号

WR<sub>xx</sub>: 写信号

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置运算放大器的产品)。

图 4-5 P21 的框图



P2: 端口寄存器 2

PM2: 端口模式寄存器 2

AMPM: 运算放大器控制寄存器

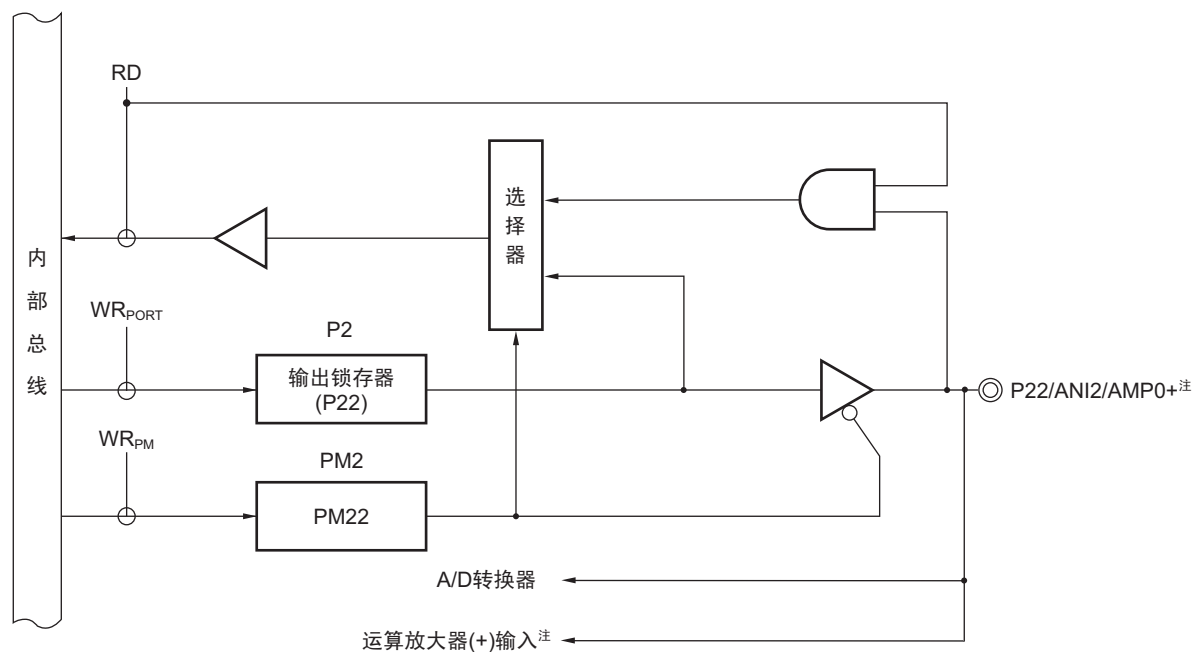
RD: 读信号

WRxx: 写信号

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置运算放大器的产品)。



图 4-6 P22 的框图



P2: 端口寄存器 2

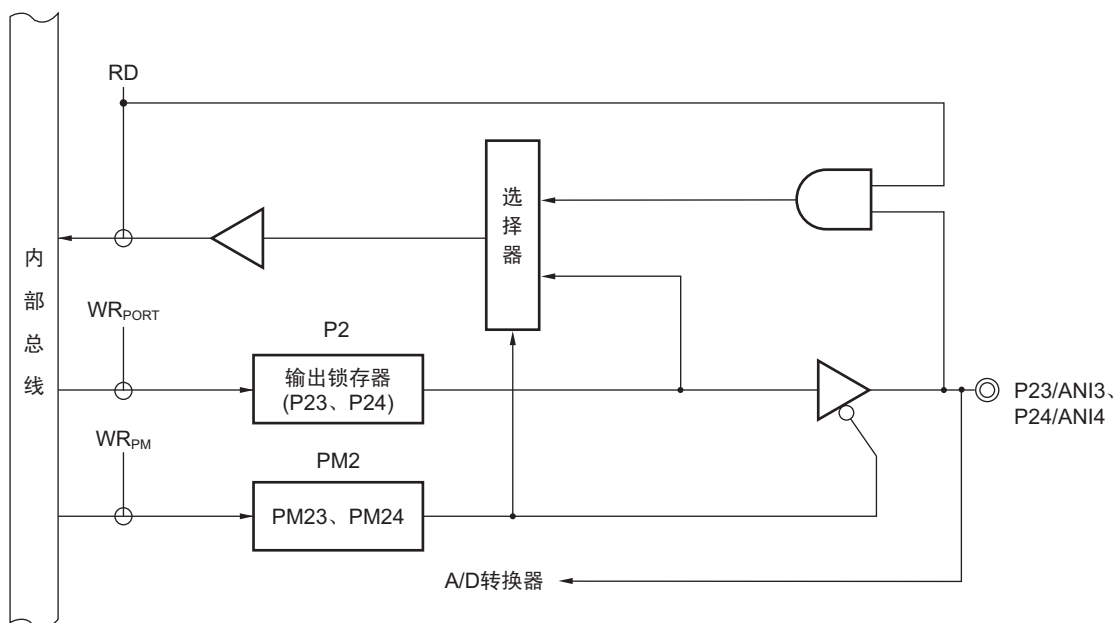
PM2: 端口模式寄存器 2

RD: 读信号

WR<sub>xx</sub>: 写信号

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置运算放大器的产品)。

图 4-7 P23、P24 的框图



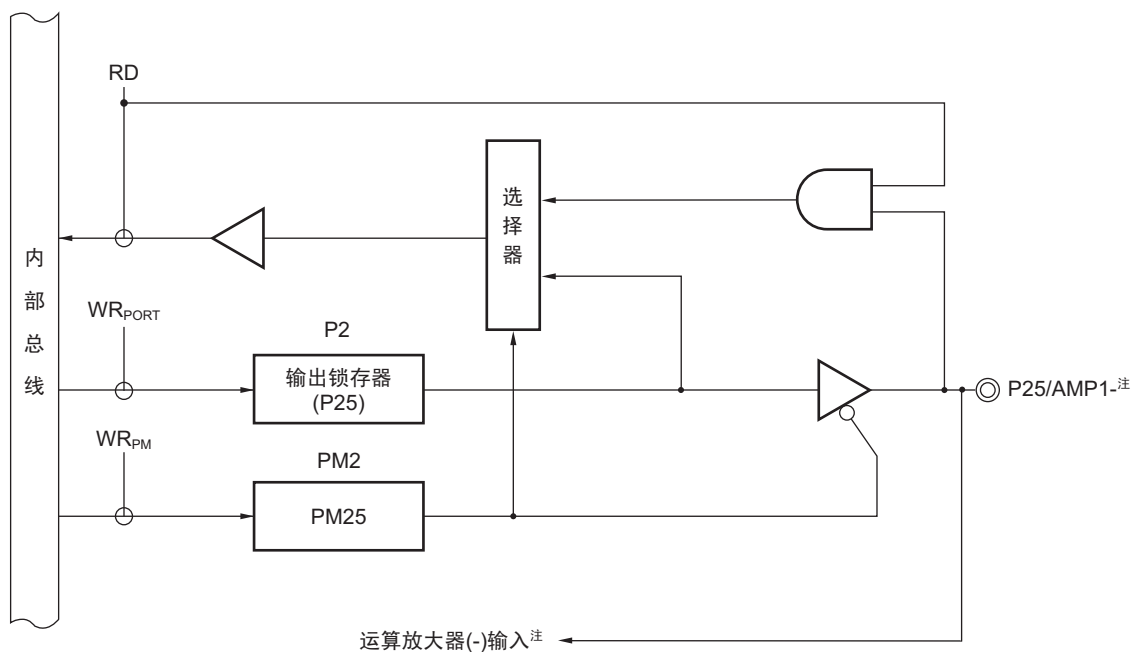
P2: 端口寄存器 2

PM2: 端口模式寄存器 2

RD: 读信号

WR<sub>xx</sub>: 写信号

图 4-8 P25 的框图



P2: 端口寄存器 2

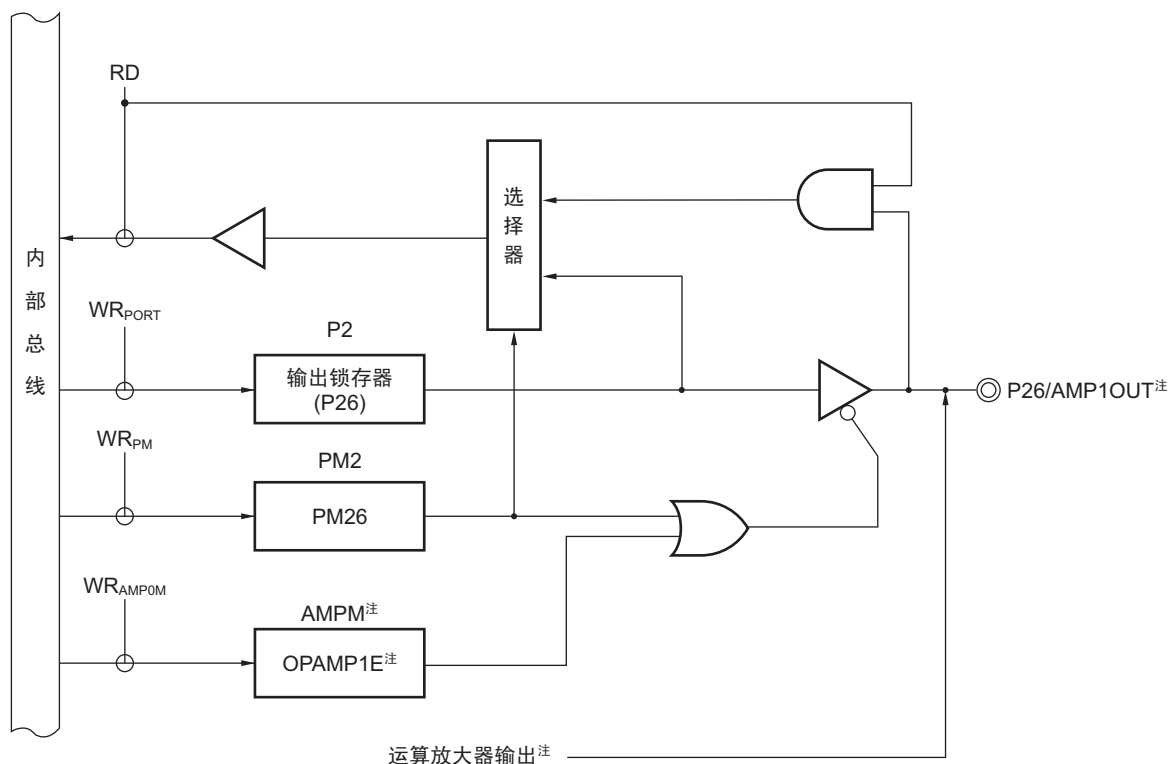
PM2: 端口模式寄存器 2

RD: 读信号

WR<sub>xx</sub>: 写信号

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置运算放大器的产品)。

图 4-9 P26 的框图



P2: 端口寄存器 2

PM2: 端口模式寄存器 2

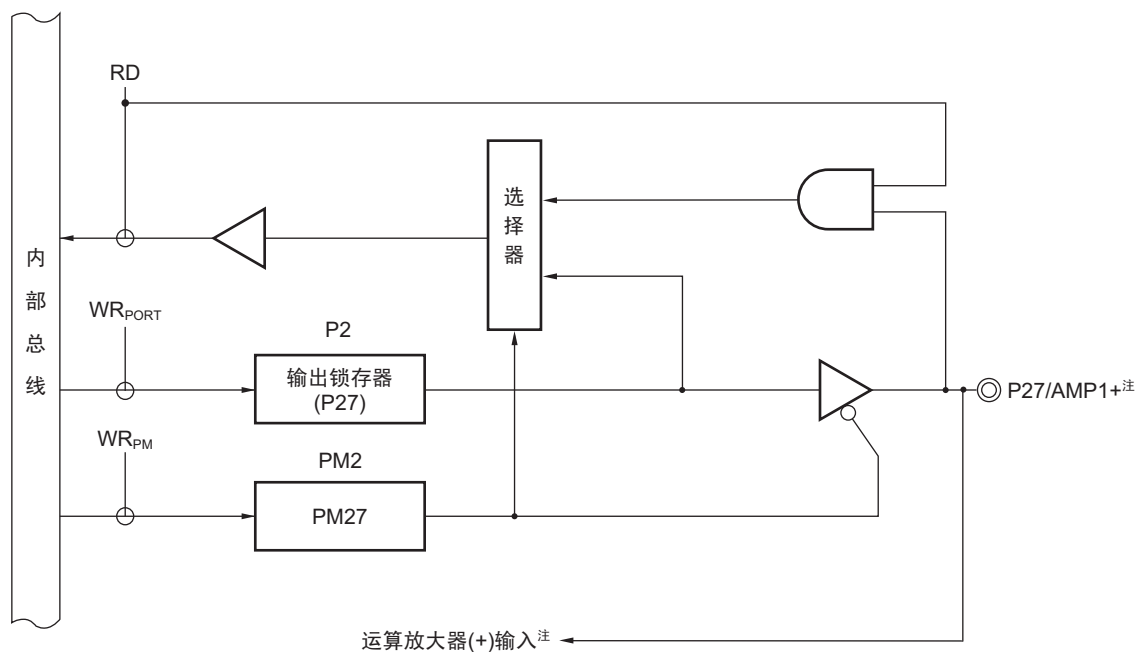
AMPM: 运算放大器控制寄存器

RD: 读信号

WR<sub>xx</sub>: 写信号

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置运算放大器的产品)。

图 4-10 P27 的框图



P2: 端口寄存器 2

PM2: 端口模式寄存器 2

RD: 读信号

WR<sub>xx</sub>: 写信号

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置运算放大器的产品)。

## 4.2.2 端口 3

16 引脚产品	20 引脚产品
P30/TOH1/TI51/INTP0	P30/TOH1/TI51/INTP0
P31/TxD0/<TO00>/CMPCOM 注 1	P31/TxD0/CMPCOM 注 2
P32/RxD0/<TI010>/CMPIN 注 1	P32/RxD0/CMPIN 注 2
—	P33/TI000/INTP1
—	P34/TO00/TI010/CMPOUT 注 2

- 注 1. 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114（内置比较器的产品）。
2. 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116（内置比较器的产品）。

备注 可通过设置端口复用切换控制寄存器（MUXSEL）指定上表 <> 内的功能。

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。在 P30 ~ P34 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。

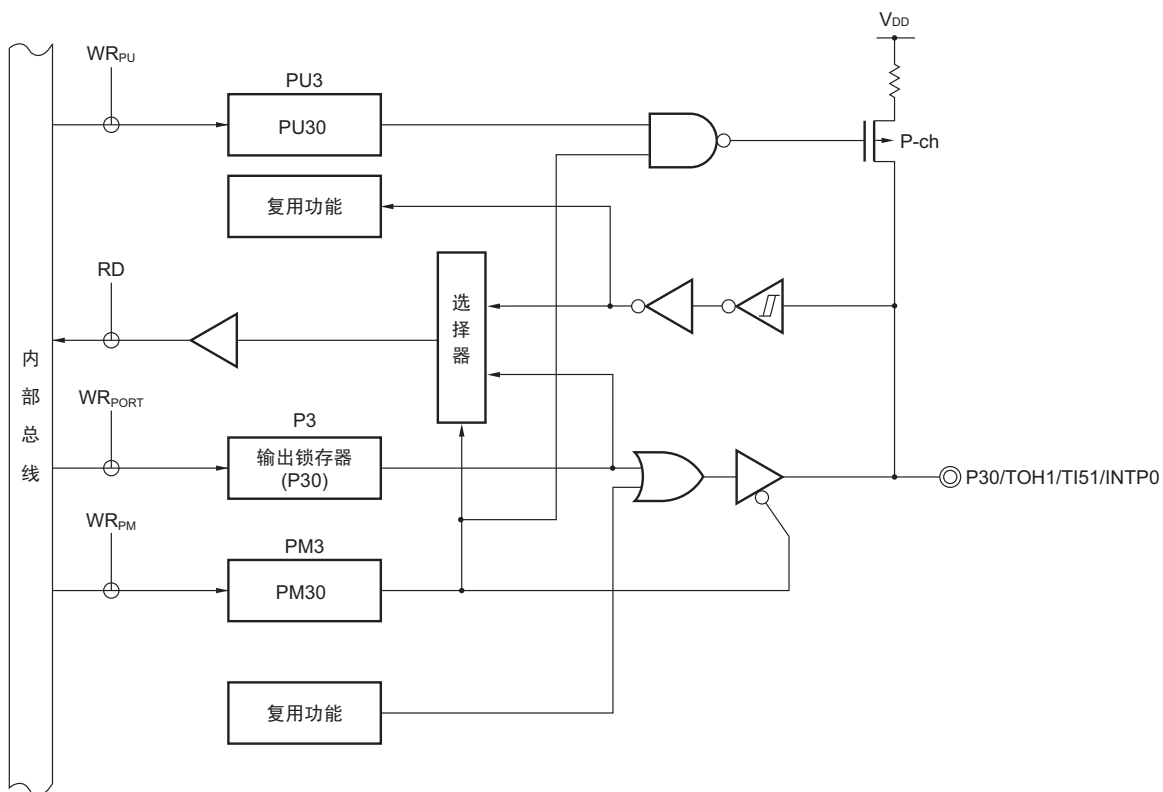
作为复用功能，该端口还可用作外部中断请求输入、定时器的输入 / 输出、串行接口的数据输入 / 输出、比较器输出、比较器模拟输入以及比较器公共输入。

在产生复位信号时，端口 3 变为输入模式。

端口 3 的框图如图 4-11 ~ 图 4-15 所示。

P31/TxD0/(TO00)/CMPCOM 和 P32/RxD0/(TI010)/CMPIN 的引脚功能设定请参照“12.3 比较器的控制寄存器”的表 12-2 和表 12-3。

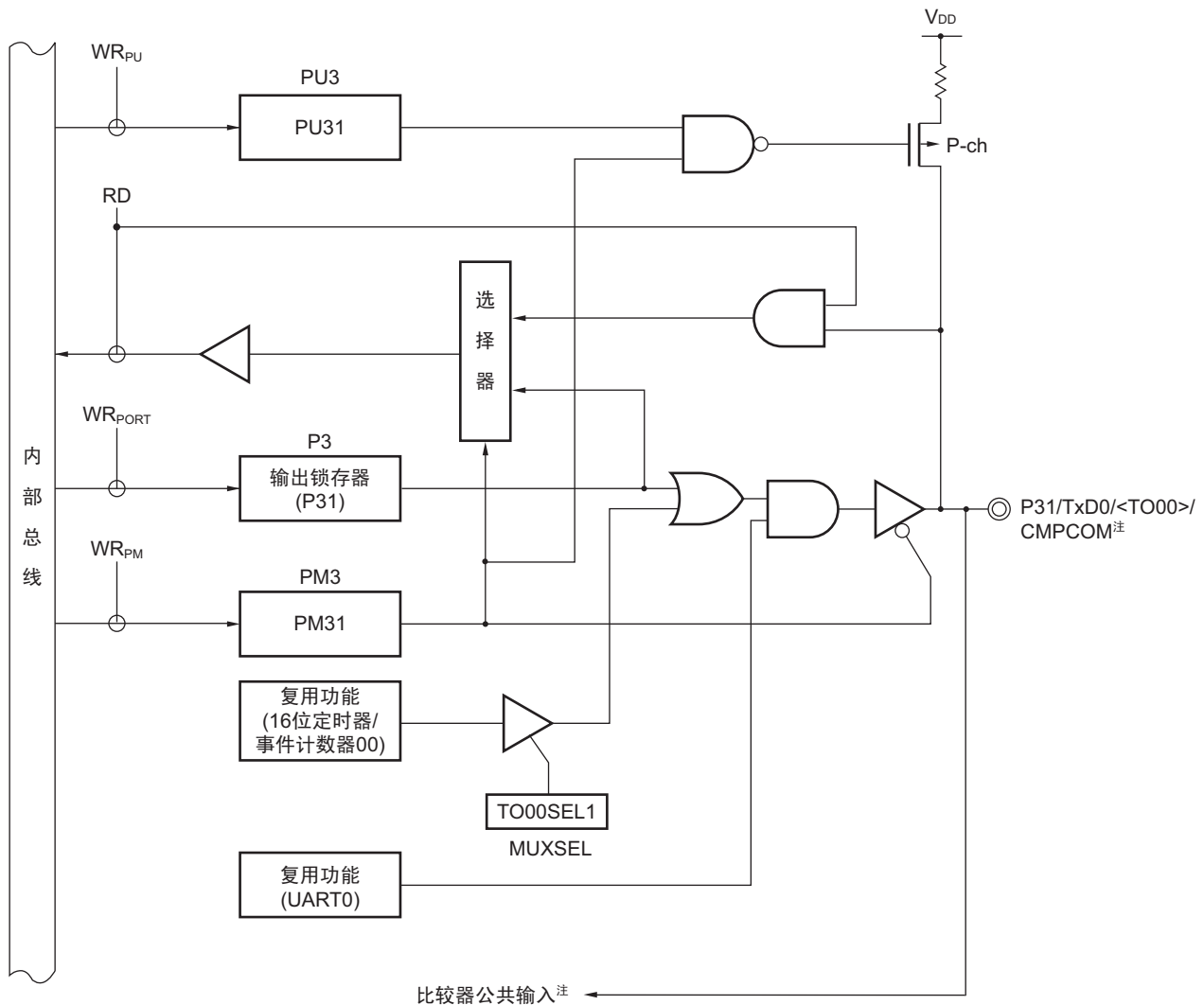
图 4-11 P30 的框图



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

图 4-12 P31 的框图 (1/2)

## (1) 16 引脚产品



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

MUXSEL: 端口复用切换控制寄存器

RD: 读信号

WR<sub>xx</sub>: 写信号

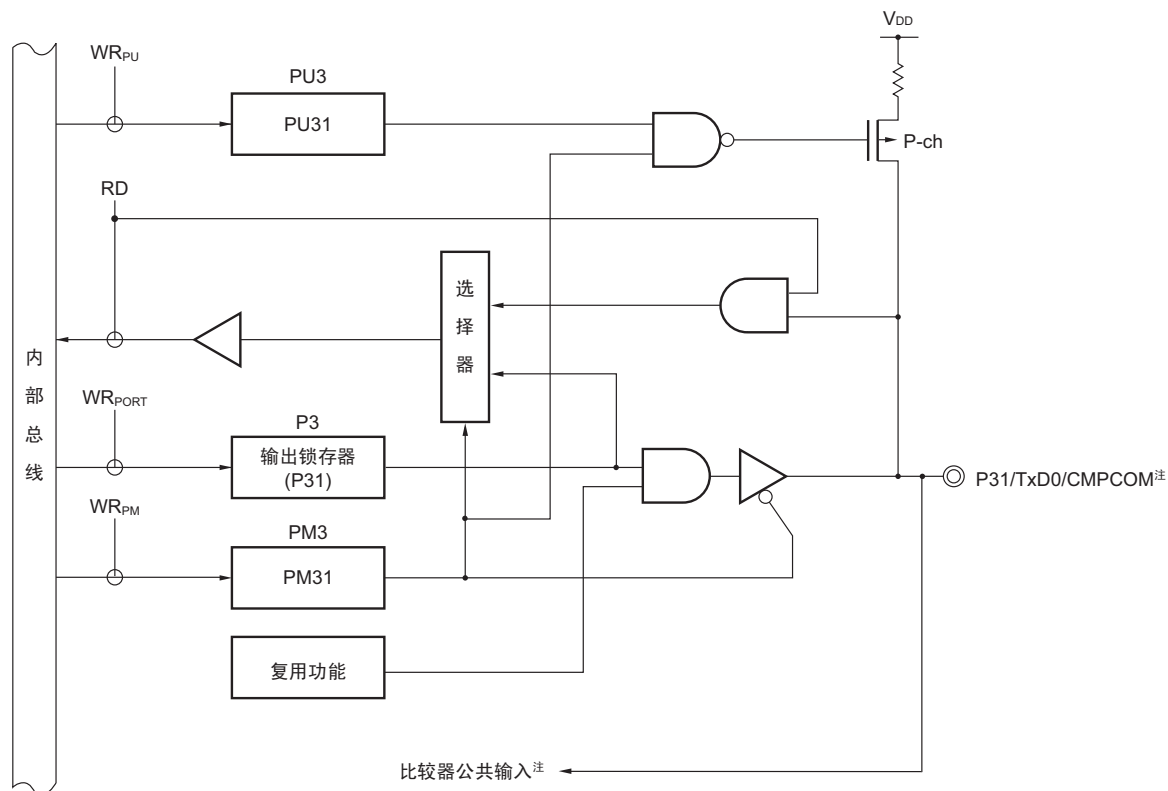
注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 (内置比较器的产品)。

备注 可通过设置端口复用切换控制寄存器 (MUXSEL) 指定上图 <> 内的功能。



图 4-12 P31 的框图 (2/2)

## (2) 20 引脚产品



P3: 端口寄存器 3

PU3: 上拉电阻选择寄存器 3

PM3: 端口模式寄存器 3

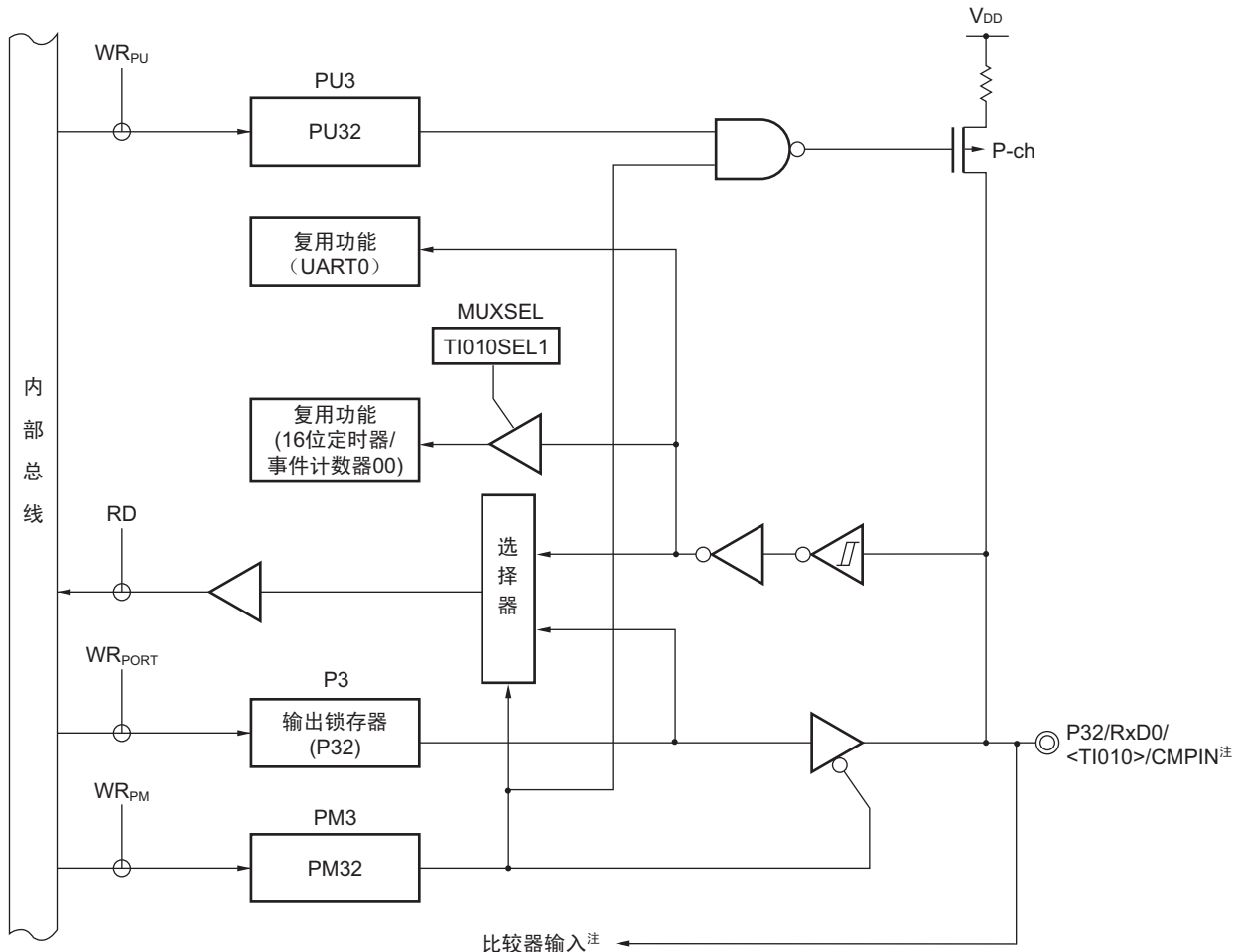
RD: 读信号

WR<sub>xx</sub>: 写信号

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置比较器的产品)。

图 4-13 P32 的框图 (1/2)

## (1) 16 引脚产品



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

MUXSEL: 端口复用切换控制寄存器

RD: 读信号

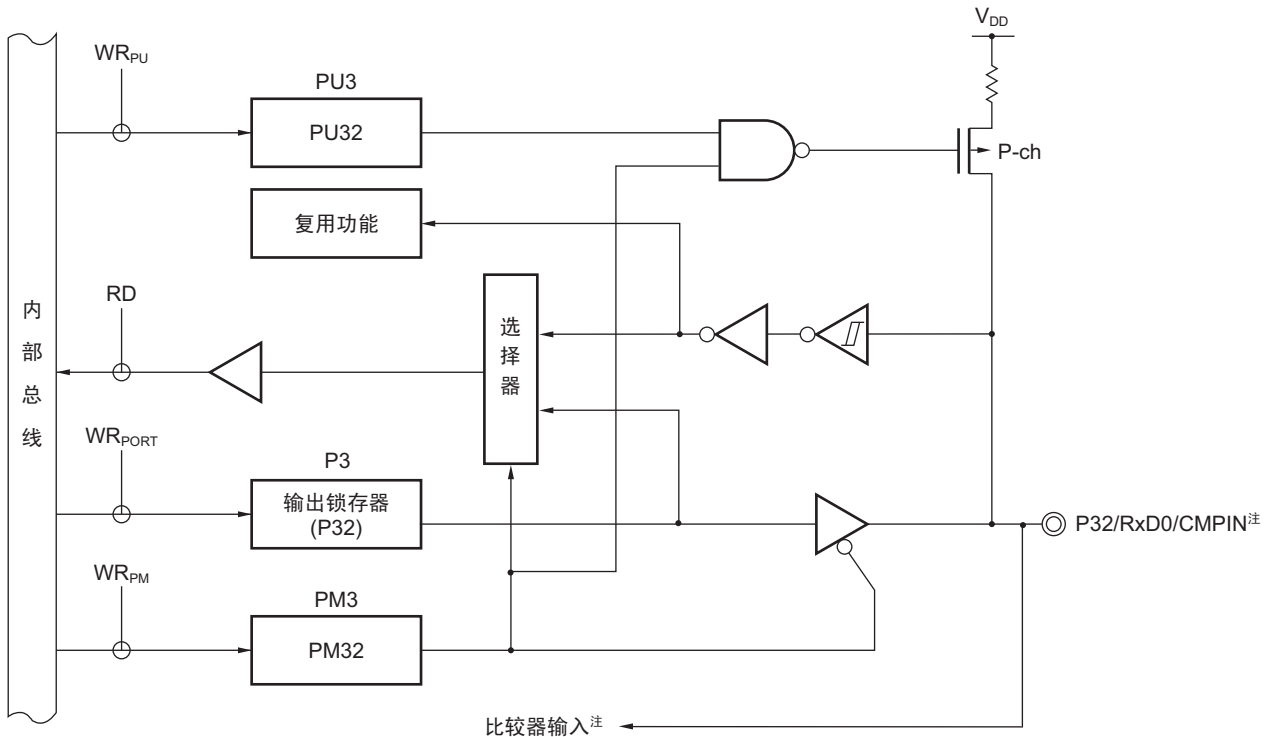
WR<sub>xx</sub>: 写信号

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 (内置比较器的产品)。

备注 可通过设置端口复用切换控制寄存器 (MUXSEL) 指定上图 < > 内的功能。

图 4-13 P32 的框图 (2/2)

## (2) 20 引脚产品



P3: 端口寄存器 3

PU3: 上拉电阻选择寄存器 3

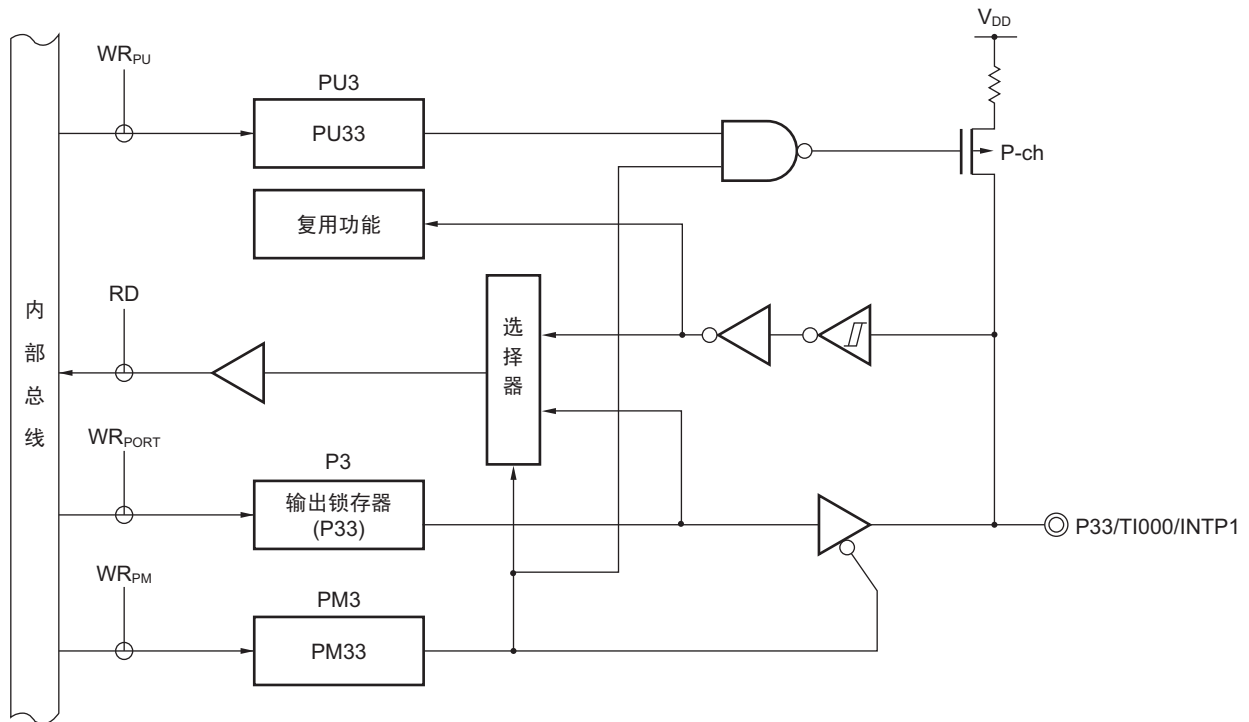
PM3: 端口模式寄存器 3

RD: 读信号

WRxx: 写信号

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置比较器的产品)。

图 4-14 P33 的框图



P3: 端口寄存器 3

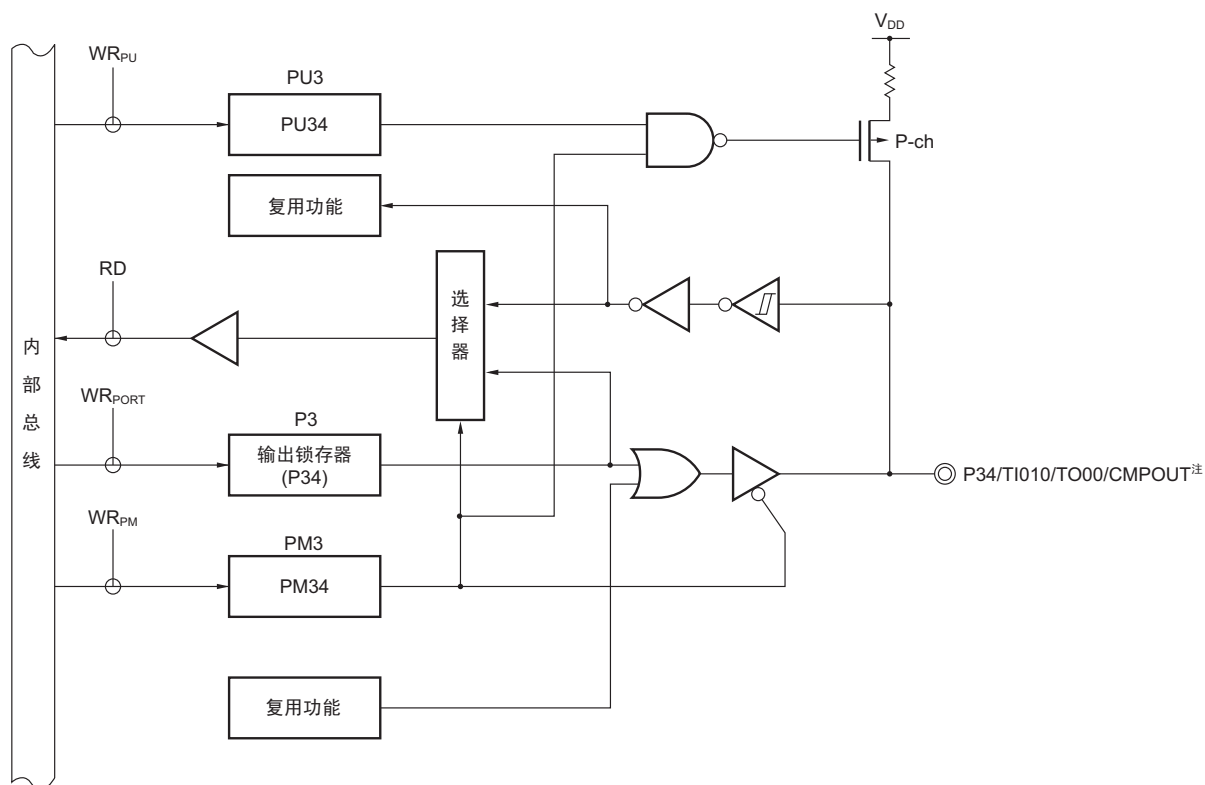
PU3: 上拉电阻选择寄存器 3

PM3: 端口模式寄存器 3

RD: 读信号

WR<sub>xx</sub>: 写信号

图 4-15 P34 的框图



P3: 端口寄存器 3

PU3: 上拉电阻选择寄存器 3

PM3: 端口模式寄存器 3

RD: 读信号

WR<sub>xx</sub>: 写信号

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116（内置比较器的产品）。

## 4.2.3 端口 12

16 引脚产品	20 引脚产品
P121/X1/<TI000>/<INTP1>/TOOLC0	P121/X1/TOOLC0
P122/X2/EXCLK/TOOLD0	P122/X2/EXCLK/TOOLD0
P125/<TI000>/<INTP1>/RESET	P125/RESET

备注 可通过设置端口复用切换控制寄存器（MUXSEL）指定上表 <> 内的功能。

P121、P122 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。

P125 是输入端口。

在 P125 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。

作为复用功能，该端口还可用作与主系统时钟的谐振器连接、主系统时钟的外部时钟输入、外部复位输入、闪存编程器 / 片上调试的时钟输入、数据输入 / 输出、外部中断请求输入以及定时器输入。

如果要将 P125/RESET 用作输入端口，必须将复位引脚模式寄存器（RSTMASK）的 bit5（RSTM）置“1”；如果要将 P125/RESET 用作外部复位输入，必须将复位引脚模式寄存器（RSTMASK）的 bit5（RSTM）置“0”

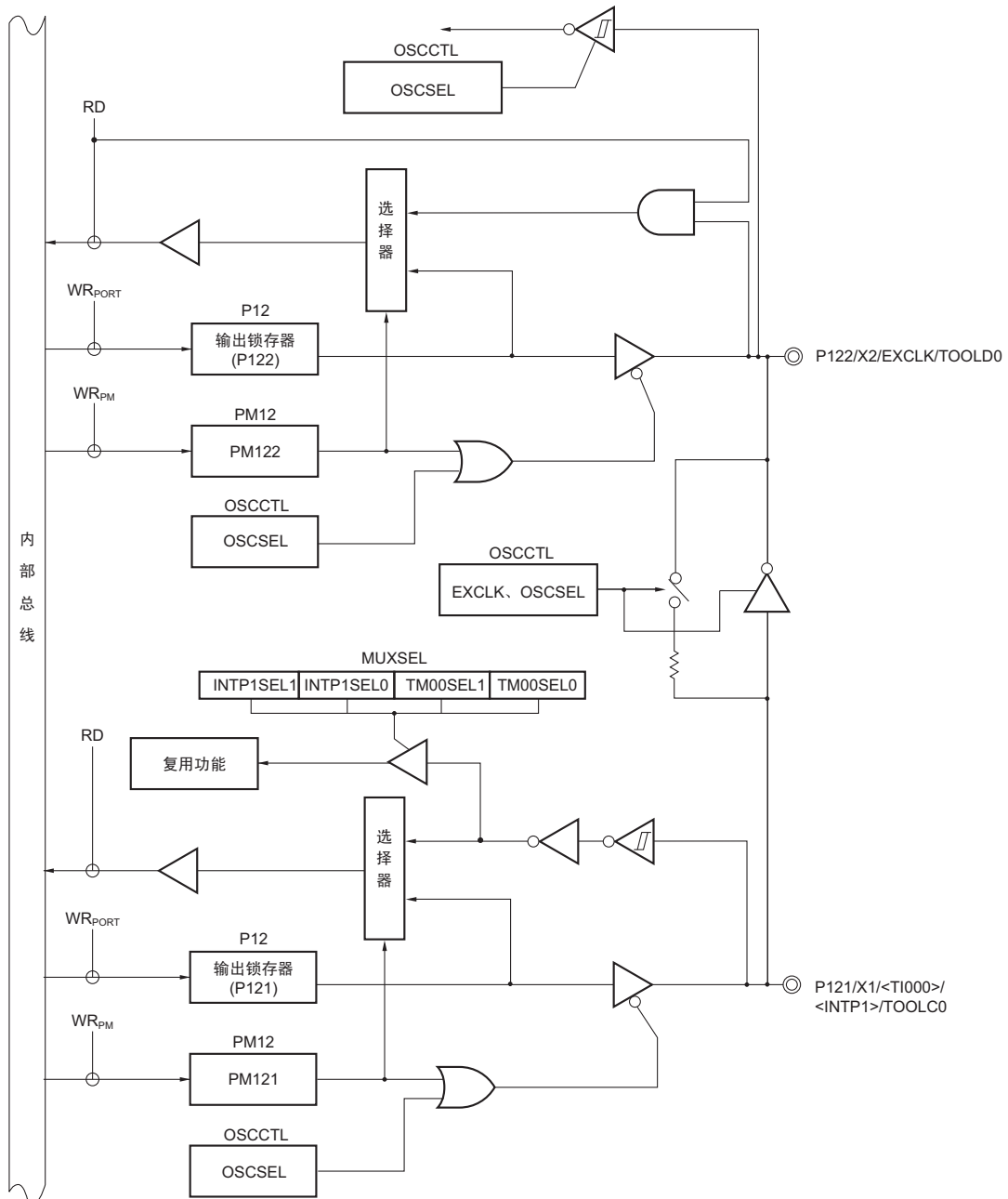
在产生复位信号时，端口 12 变为输入模式。

端口 12 的框图如图 4-16、图 4-17 所示。

- 注意 1. 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 全部为输入端口）。
2. 复位解除后，RESET/P125 的引脚功能为外部复位输入。
  3. 因为复位解除后的 RESET/P125 为外部复位输入，所以如果在低电平输入期间发生复位，就保持复位状态直到变为高电平输入为止。

图 4-16 P121、P122 的框图 (1/2)

## (1) 16 引脚产品



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

MUXSEL: 端口复用切换控制寄存器

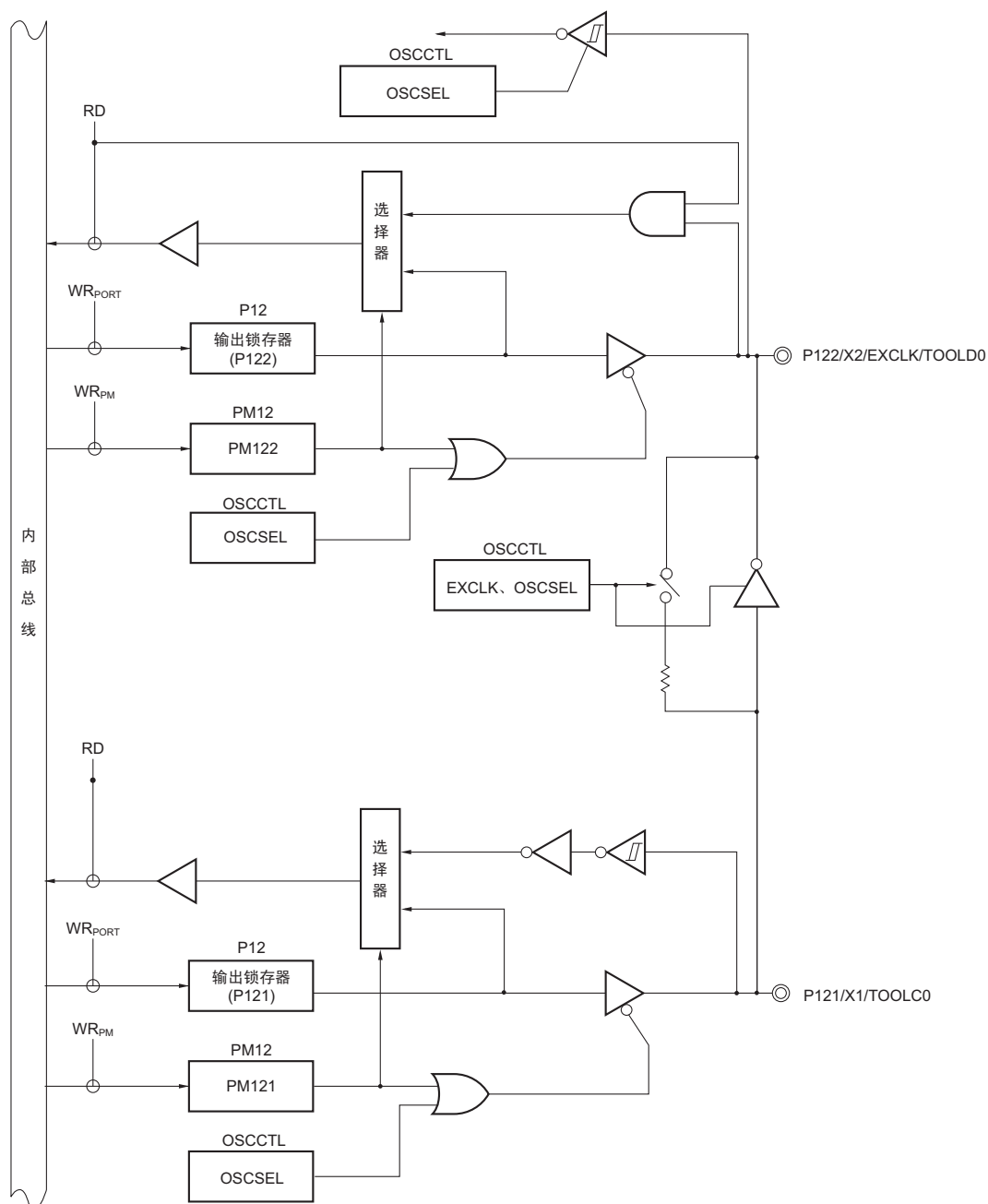
RD: 读信号

WRxx: 写信号

备注 可通过设置端口复用切换控制寄存器 (MUXSEL) 指定上图 <> 内的功能。

图 4-16 P121、P122 的框图 (2/2)

## (2) 20 引脚产品



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

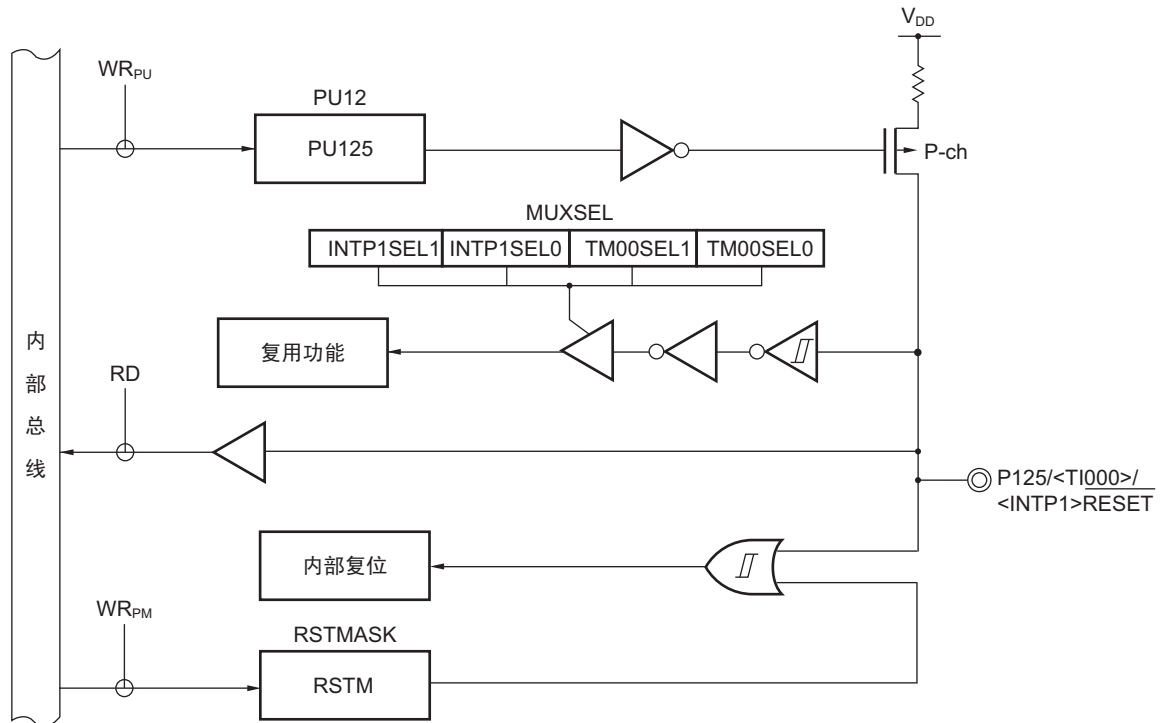
RD: 读信号

WR<sub>xx</sub>: 写信号



图 4-17 P125 的框图

## (1) 16 引脚产品



PU12: 上拉电阻选择寄存器 12

RD: 读信号

RSTMASK: 复位引脚模式寄存器

MUXSEL: 端口复用切换控制寄存器

WR<sub>xx</sub>: 写信号

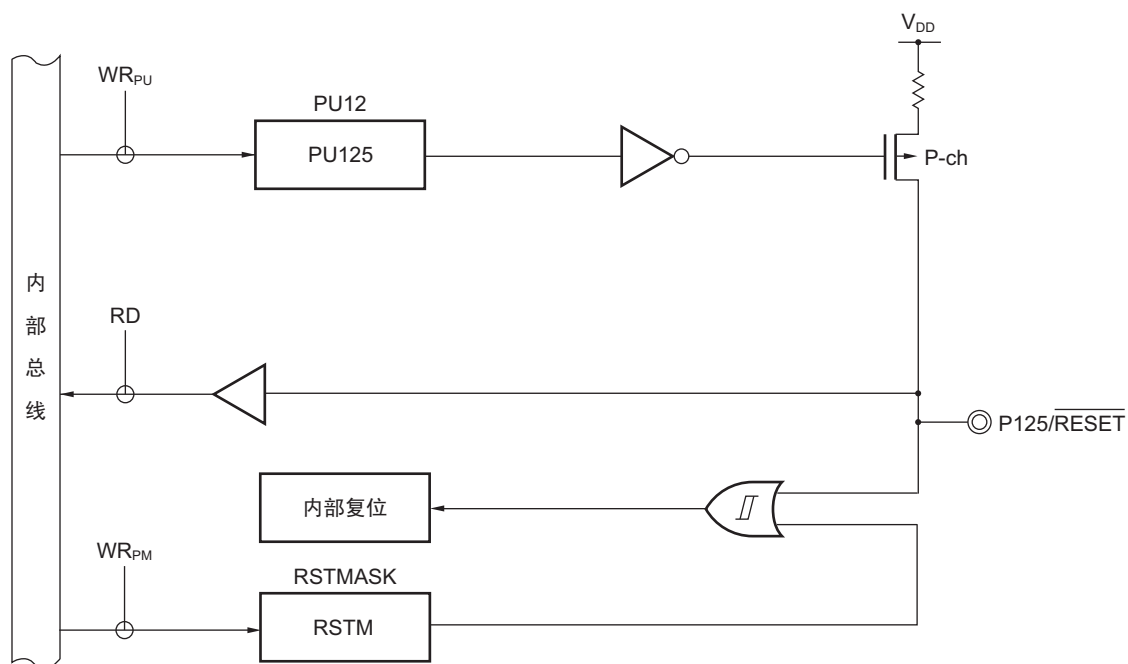
**注意** 因为复位解除后的  $\overline{\text{RESET}}/\text{P125}$  为外部复位输入，所以如果在低电平输入期间发生复位，就保持复位状态直到变为高电平输入为止。

**备注 1.** 复位时，外部复位功能和上拉电阻有效（RSTM=0、PU125=1）。用作端口功能时，必须将 RSTM 位置“1”。

**2.** 可通过设置端口复用切换控制寄存器（MUXSEL）指定上图 <> 内的功能。

图 4-17 P125 的框图 (2/2)

## (2) 20 引脚产品



PU12: 上拉电阻选择寄存器 12

RSTMASK: 复位引脚模式寄存器

RD: 读信号

WR<sub>xx</sub>: 写信号

**注意** 因为复位解除后的  $\overline{\text{RESET/P125}}$  为外部复位输入，所以如果在低电平输入期间发生复位，就保持复位状态直到变为高电平输入为止。

**备注** 复位时，外部复位功能和上拉电阻有效（RSTM=0、PU125=1）。用作端口功能时，必须将 RSTM 位置“1”。

### 4.3 端口功能的控制寄存器

通过以下 6 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)
- 复位引脚模式寄存器 (RSTMASK)
- A/D 端口配置寄存器 (ADPC)
- 端口复用切换控制寄存器 (MUXSEL) (只限 16 引脚产品)

#### (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。

在产生复位信号时，端口模式寄存器为“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 4-18 端口模式寄存器的格式 (16 引脚产品)

符号	7	6	5	4	3	2	1	0	地址	复位时	R/W
PM2	PM27 注 1	PM26 注 1	PM25 注 1	1	1	PM22 注 1	PM21 注 1	PM20 注 1	FF22H	FFH	R/W
PM3	1	1	1	1	1	PM32 注 2	PM31 注 2	PM30	FF23H	FFH	R/W
PM12	1	1	1	1	1	PM122	PM121	1	FF2CH	FFH	R/W
PMmn	选择 Pmn 引脚的输入 / 输出模式 (m=2、3、12；n=0~2、5~7)										
0	输出模式 (输出缓冲器 ON)										
1	输入模式 (输出缓冲器 OFF)										

注 1. 通过 ADPC 寄存器设置为模拟输入时，必须设置为输入模式。

2. 通过 CMPPC 寄存器设置为模拟输入时，必须设置为输入模式 (只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 (内置比较器的产品))。

注意 将 PM2 的 bit3 和 bit4、PM3 的 bit3 ~ bit7 以及 PM12 的 bit0 和 bit3 ~ bit7 置“1”。

图 4-19 端口模式寄存器的格式（20 引脚产品）

符号	7	6	5	4	3	2	1	0	地址	复位时	R/W
PM2	PM27 注 1	PM26 注 1	PM25 注 1	PM24 注 1	PM23 注 1	PM22 注 1	PM21 注 1	PM20 注 1	FF22H	FFH	R/W
PM3	1	1	1	PM34	PM33	PM32 注 2	PM31 注 2	PM30	FF23H	FFH	R/W
PM12	1	1	1	1	1	PM122	PM121	1	FF2CH	FFH	R/W
PMmn	选择 Pmn 引脚的输入 / 输出模式 (m = 2、3、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 1. 通过 ADPC 寄存器设置为模拟输入时，必须设置为输入模式。  
 2. 通过 CMPPC 寄存器设置为模拟输入时，必须设置为输入模式（只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116（内置比较器的产品））。

注意 将 PM3 的 bit5 ~ bit7 以及 PM12 的 bit0 和 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。

在产生复位信号时，端口寄存器为“00H”。

图 4-20 端口寄存器的格式（16 引脚产品）

符号	7	6	5	4	3	2	1	0	地址	复位时	R/W
P2	P27 注 1	P26 注 1	P25 注 1	0	0	P22 注 1	P21 注 1	P20 注 1	FF02H	00H (输出锁存器)	R/W
P3	0	0	0	0	0	P32 注 1	P31 注 1	P30	FF03H	00H (输出锁存器)	R/W
P12	0	0	P125	0	0	P122 注 2	P121 注 2	0	FF0CH	00H (输出锁存器)	R/W
Pmn	m=2、3、12；n=0~2、5~7										
	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 1. 在设置为模拟输入并且设置为输入模式时，不能存取输出锁存器。  
 2. 在 X1 振荡模式和时钟输入模式下使用的引脚的输出锁存器的读取值总是为“0”。

图 4-21 端口寄存器格式（20 引脚产品）

符号	7	6	5	4	3	2	1	0	地址	复位时	R/W
P2	P27 注 1	P26 注 1	P25 注 1	P24 注 1	P23 注 1	P22 注 1	P21 注 1	P20 注 1	FF02H	00H (输出锁存器)	R/W
P3	0	0	0	P34	P33	P32 注 1	P31 注 1	P30	FF03H	00H (输出锁存器)	R/W
P12	0	0	P125	0	0	P122 注 2	P121 注 2	0	FF0CH	00H (输出锁存器)	R/W
Pmn	m=2、3、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 1. 在设置为模拟输入并且设置为输入模式时，不能存取输出锁存器。  
 2. 在 X1 振荡模式和时钟输入模式下使用的引脚的输出锁存器的读取值总是为“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。

在产生复位信号时，上拉电阻选择寄存器为“00H”（仅 PU12 为“20H”）。

图 4-22 上拉电阻选择寄存器的格式（16 引脚产品）

符号	7	6	5	4	3	2	1	0	地址	复位时	R/W
PU3	0	0	0	0	0	PU32	PU31	PU30	FF33H	00H	R/W
PU12	0	0	PU125	0	0	0	0	0	FF3CH	20H	R/W
PUmn	选择 Pmn 的内部上拉电阻 (m=3、12；n=0~2、5)										
0	不连接内部上拉电阻										
1	连接内部上拉电阻										

图 4-23 上拉电阻选择寄存器的格式（20 引脚产品）

符号	7	6	5	4	3	2	1	0	地址	复位时	R/W
PU3	0	0	0	PU34	PU33	PU32	PU31	PU30	FF33H	00H	R/W
PU12	0	0	PU125	0	0	0	0	0	FF3CH	20H	R/W
PUmn	选择 Pmn 的内部上拉电阻 (m=3、12；n=0~5)										
0	不连接内部上拉电阻										
1	连接内部上拉电阻										

## (4) 复位引脚模式寄存器 (RSTMASK)

复位引脚模式寄存器设置  $\overline{\text{RESET}}/\text{P125}$  的引脚功能（外部复位输入 / 输入专用端口）。

通过 1 位或 8 位存储器操作指令设置复位引脚模式寄存器。

在产生复位信号时，复位引脚模式寄存器为“00H”。

图 4-24 复位引脚模式寄存器 (RSTMASK) 的格式

符号	7	6	5	4	3	2	1	0	地址	复位时	R/W
RSTMASK	0	0	RSTM	0	0	0	0	0	FF2DH	00H	R/W

RSTM	选择 $\overline{\text{RESET}}/\text{P125}$ 的引脚功能
0	用作外部复位输入 ( $\overline{\text{RESET}}$ )
1	用作输入专用端口 (P125)

## (5) A/D 端口配置寄存器 (ADPC)

A/D 端口配置寄存器将 P20/AMP0-/ANI0 ~ P24/ANI4（16 引脚产品时：P20/AMP0-/ANI0 ~ P22/AMP0+/ANI2、P25/AMP1-/ANI5 ~ P27/AMP1+/ANI7）切换为端口的数字输入 / 输出或者模拟输入 / 输出。ADPC 的各位分别对应端口 2 的每个引脚，并且可以 1 位单位进行指定。

通过 1 位或 8 位存储器操作指令设置 ADPC。

在产生复位信号时，ADPC 为“00H”。

图 4-25 A/D 端口配置寄存器 (ADPC) 的格式 (16 引脚产品)

符号	7	6	5	4	3	2	1	0	地址	复位时	R/W
ADPC	ADPC7	ADPC6	ADPC5	0	0	ADPC2	ADPC1	ADPC0	FF97H	00H	R/W

ADPCn	选择数据输入 / 输出或者模拟输入 / 输出 (n=0 ~ 2、5 ~ 7)
0	模拟输入 / 输出
1	数字输入 / 输出

注意 1. 通过端口模式寄存器 2 (PM2)，将设置为模拟输入的引脚置为输入模式。

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 26 章 等待的注意事项”。

图 4-26 A/D 端口配置寄存器 (ADPC) 的格式 (20 引脚产品)

符号	7	6	5	4	3	2	1	0	地址	复位时	R/W
ADPC	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	FF97H	00H	R/W
	选择数据输入 / 输出或者模拟输入 / 输出 (n=0 ~ 7)										
	0	模拟输入 / 输出									
	1	数字输入 / 输出									

注意 1. 通过端口模式寄存器 2 (PM2)，将设置为模拟输入的引脚置为输入模式。

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 26 章 等待的注意事项”。

#### (6) 端口复用切换控制寄存器 (MUXSEL) (只限 16 引脚产品)

端口复用切换控制寄存器设置引脚功能的分配。

通过 1 位或 8 位存储器操作指令设置 MUXSEL。

在产生复位信号时，MUXSEL 为“00H”。

图 4-27 端口复用切换控制寄存器 (MUXSEL) 的格式

符号	<7>	<6>	<5>	<4>	3	2	<1>	<0>	地址	复位时	R/W
MUXSEL	INTP1SEL1	INPT1SEL0	TM00SEL1	TM00SEL0	0	0	TI010SEL	TO00SEL	FF39H	00H	R/W

INTP1SEL1	INPT1SEL0	外部中断输入 (INTP1) 的引脚分配
0	0	(默认)
0	1	P121/INTP1
1	0	P125/INTP1
1	1	禁止设置

TM00SEL1	TM00SEL0	16 位定时器 / 事件计数器 00 的输入 (TI000) 引脚分配
0	0	(默认)
0	1	P121/TI000
1	0	P125/TI000
1	1	禁止设置

TI010SEL	16 位定时器 / 事件计数器 00 的输入 (TI010) 引脚分配
0	(默认)
1	P32/TI010

TO00SEL	16 位定时器 / 事件计数器 00 的输出 (TO00) 引脚分配
0	(默认)
1	P31/TO00



## 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。  
在产生复位信号时，清除输出锁存器的数据。

#### (2) 输入模式的情况

通过传送指令将值写入输出锁存器。因为输出缓冲器处于关闭状态，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。  
在产生复位信号时，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。  
在产生复位信号时，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关闭状态，所以引脚状态不会发生变化。  
在产生复位信号时，清除输出锁存器的数据。

#### 4.5 使用复用功能时的端口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 4-12 和表 4-13 设置端口模式寄存器和输出锁存器。

表 4-12 使用复用功能时的端口模式寄存器和输出锁存器的设置（16 引脚产品）（1/2）

引脚名称	复用功能		PMxx	Pxx
	名称	输入 / 输出		
P20	ANI0 注 1	输入	1	x
	AMP0- 注 1、注 2	输入	1	x
P21	ANI1 注 1	输入	1	x
	AMP0OUT 注 1、注 2	输出	1	x
P22	ANI2 注 1	输入	1	x
	AMP0+ 注 1、注 2	输入	1	x
P25	ANI5 注 1	输入	1	x
	AMP1- 注 1、注 2	输入	1	x
P26	ANI6 注 1	输入	1	x
	AMP1OUT 注 1、注 2	输出	1	x
P27	ANI7 注 1	输入	1	x
	AMP1+ 注 1、注 2	输入	1	x
P30	INTP0	输入	1	x
	TI51	输入	1	x
	TOH1	输出	0	0
P31	TxD0 注 3	输出	0	1
	<TO00> 注 3	输出	0	0
	CMPCOM 注 2、注 3	输入	1	x
P32	RxD0 注 3	输入	1	x
	<TI010> 注 3	输入	1	x
	CMPIN 注 2、注 3	输入	1	x

- 注 1. 由 ADPC 寄存器、PM2 寄存器、ADS 寄存器和 AMPM 寄存器决定引脚的功能。请参照表 4-5 和表 4-6。  
 2. 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114（内置运算放大器 / 比较器的产品）。  
 3. 由 CMPPC 寄存器、PM3 寄存器和 CMPCTL 寄存器决定引脚的功能。请参照表 12-2 和表 12-3。

备注 1. x: 忽略

PMxx: 端口模式寄存器

Pxx: 端口的输出锁存器

2. 可通过设置端口复用切换控制寄存器（MUXSEL）指定上表 <> 内的功能。

表 4-12 使用复用功能时的端口模式寄存器和输出锁存器的设置（16 引脚产品）（2/2）

引脚名称	复用功能		PMxx	Pxx
	名称	输入 / 输出		
P121	X1 注 1	—	x	x
	<TI000>	输入	1	x
	<INTP1>	输入	1	x
	TOOLC0	输入	x	x
P122	X2 注 1	—	x	x
	EXCLK 注 1	输入	x	x
	TOOLD0	输入 / 输出	x	x
P125	RESET 注 2	输入	x	x
	<TI000>	输入	1	x
	<INTP1>	输入	1	x

- 注 1. 在将 P121 和 P122 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 全部为输入端口）。
2. 在将 P125 用作外部复位输入（ $\overline{\text{RESET}}$ ）时，必须将 RSTM 位（RSTMASK 寄存器的 bit5）置“0”。

备注 1. x: 忽略

PMxx: 端口模式寄存器

Pxx: 端口的输出锁存器

2. 可通过设置端口复用切换控制寄存器（MUXSEL）指定上表 <> 内的功能。

表 4-13 使用复用功能时的端口模式寄存器和输出锁存器的设置 (20 引脚产品) (1/2)

引脚名称	复用功能		PMxx	Pxx
	名称	输入 / 输出		
P20	ANI0 注 1	输入	1	x
	AMP0- 注 1、注 2	输入	1	x
P21	ANI1 注 1	输入	1	x
	AMP0OUT 注 1、注 2	输出	1	x
P22	ANI2 注 1	输入	1	x
	AMP0+ 注 1、注 2	输入	1	x
P23、P24	ANI3, ANI4 注 1	输入	1	x
P25	AMP1- 注 2	输入	1	x
P26	AMP1OUT 注 2	输出	1	x
P27	AMP1+ 注 2	输入	1	x
P30	INTP0	输入	1	x
	TI51	输入	1	x
	TOH1	输出	0	0
P31	TxD0 注 3	输出	0	1
	CMPCOM 注 2、注 3	输入	1	x
P32	RxD0 注 3	输入	1	x
	CMPIN 注 2、注 3	输入	1	x
P33	INTP1	输入	1	x
	TI000	输入	1	x
P34	TI010	输入	1	x
	TO00	输出	0	0
	CMPOUT 注 2	输出	0	0

- 注 1. 由 ADPC 寄存器、PM2 寄存器、ADS 寄存器和 AMPM 寄存器决定引脚的功能。请参照表 4-7 ~ 表 4-11。  
 2. 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置运算放大器 / 比较器的产品)。  
 3. 由 CMPPC 寄存器、PM3 寄存器和 CMPCTL 寄存器决定引脚的功能。请参照表 12-2 和表 12-3。

备注 x: 忽略  
 PMxx: 端口模式寄存器  
 Pxx: 端口的输出锁存器

表 4-13 使用复用功能时的端口模式寄存器和输出锁存器的设置（20 引脚产品）（2/2）

引脚名称	复用功能		PMxx	Pxx
	名称	输入 / 输出		
P121	X1 注 1	—	x	x
	TOOLC0	输入	x	x
P122	X2 注 1	—	x	x
	EXCLK 注 1	输入	x	x
	TOOLD0	输入 / 输出	x	x
P125	$\overline{\text{RESET}}$ 注 2	输入	x	x

- 注 1. 在将 P121 和 P122 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 全部为输入端口）。
2. 在将 P125 用作外部复位输入（ $\overline{\text{RESET}}$ ）时，必须将 RSTM 位（RSTMASK 寄存器的 bit5）置“0”。

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 P20 为输出端口，P21 ~ P27 为输入端口（引脚状态全部为高电平），并且端口 2 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 P20 的输出从低电平置为高电平，端口 2 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 R7F0C30x、R7F0C31x 内部按以下顺序执行 1 位存储器操作指令。

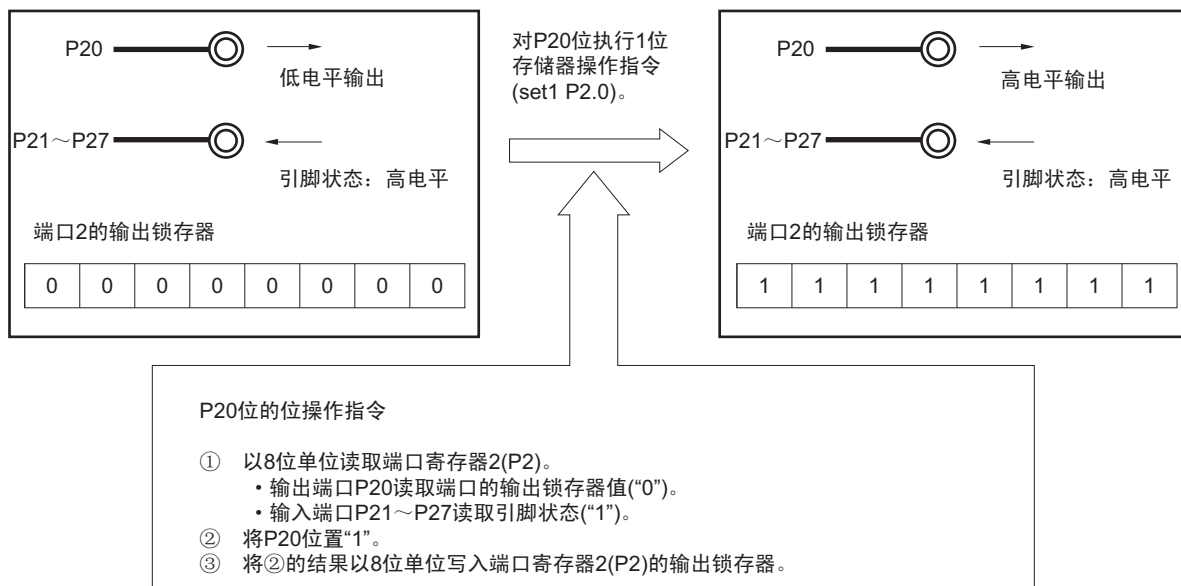
- <1> 以 8 位单位读取 Pn 寄存器。
- <2> 对作为对象的 1 位进行操作。
- <3> 以 8 位单位写入 Pn 寄存器。

在步骤<1>中，输出端口 P20 读取输出锁存的值（“0”），输入端口 P21 ~ P27 读取引脚状态。此时，如果 P21 ~ P27 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 4-28 1 位存储器操作指令（P20 的情况）



备注 以下指令为 1 位存储器操作指令。

- MOV1、AND1、OR1、XOR1、SET1、CLR1、NOT1

## 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  $f_X=1 \sim 10$  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  $f_{IH}=8$  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（ $f_{EXCLK}=1 \sim 10$  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。

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

#### (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  $f_{IL}=240$  kHz(TYP.) 的振荡时钟。在复位解除后，低速内部振荡时钟必须开始运行。

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
- 8 位定时器 H1（选择  $f_{IL}$ 、 $f_{IL}/2^7$  或者  $f_{IL}/2^9$  时）

备注  $f_{IL}$ : 低速内部振荡时钟频率

## 5.2 时钟发生电路的结构

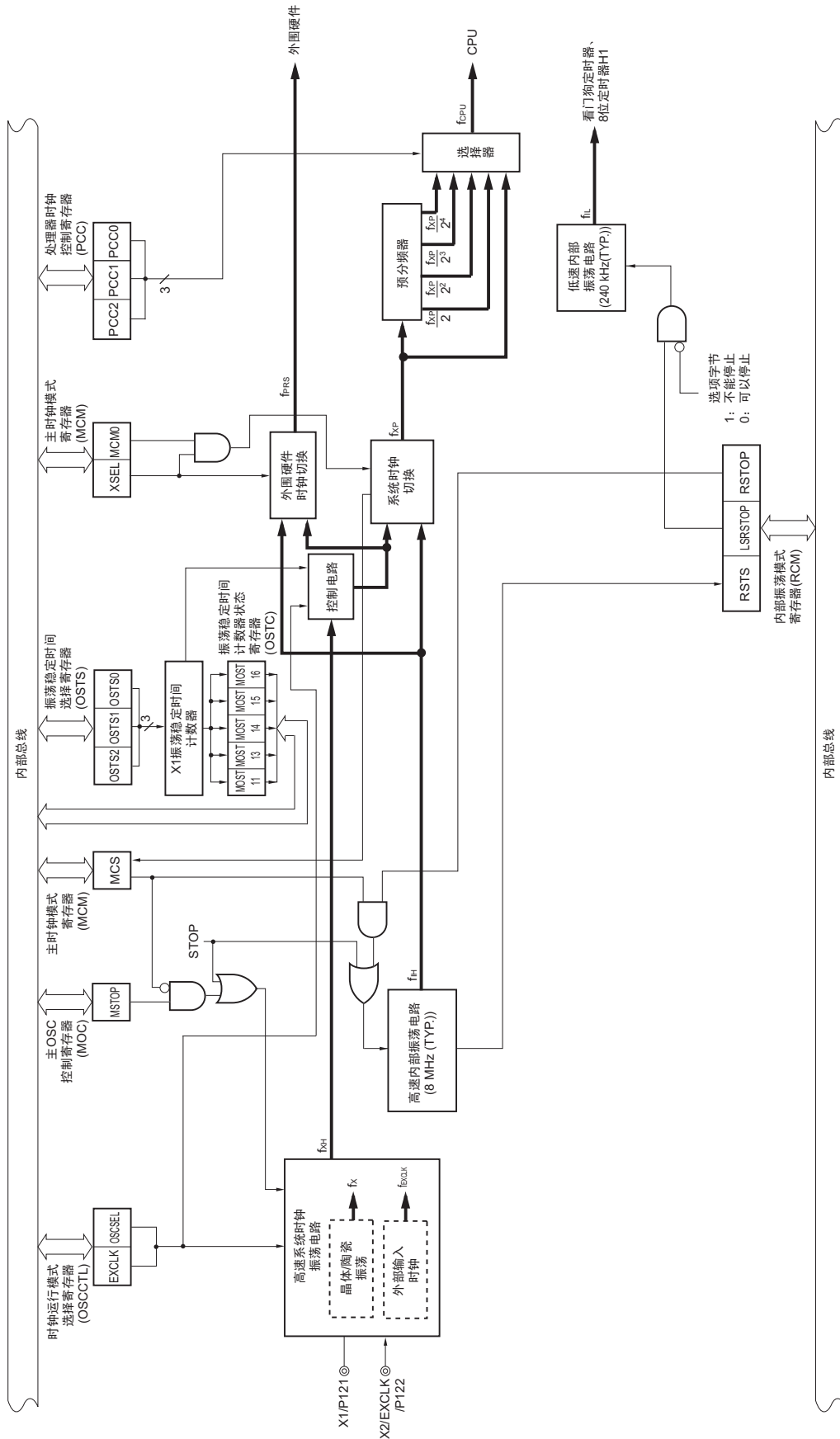
时钟发生电路由以下硬件构成。

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 时钟发生器系统图



---

备注	$f_X$ :	X1 时钟振荡频率
	$f_{IH}$ :	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
	$f_{PRS}$ :	外围硬件时钟频率
	$f_{CPU}$ :	CPU 时钟频率
	$f_{IL}$ :	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 (OSTS)
- 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。

在产生复位信号时，OSCCTL 为“00H”。

图 5-2 时钟运行模式选择寄存器 (OSCCTL) 的格式

地址: FF9FH 复位时: 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	输入端口模式	输入端口	
0	1	X1 振荡模式	连接晶体 / 陶瓷谐振器	
1	0	输入端口模式	输入端口	
1	1	外部时钟输入模式	输入端口	外部时钟输入

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”。

## (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。

在产生复位信号时，PCC 为“01H”。

图 5-3 处理器时钟控制寄存器 (PCC) 的格式

地址：FFF<sub>BH</sub> 复位时：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 R7F0C30x、R7F0C31x 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 $\mu$ s	0.25 $\mu$ s(TYP.)
$f_{XP}/2$	0.4 $\mu$ s	0.5 $\mu$ s(TYP.)
$f_{XP}/2^2$	0.8 $\mu$ s	1.0 $\mu$ s(TYP.)
$f_{XP}/2^3$	1.6 $\mu$ s	2.0 $\mu$ s(TYP.)
$f_{XP}/2^4$	3.2 $\mu$ s	4.0 $\mu$ 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。

在产生复位信号时，RCM为“80H”注1。

图 5-4 内部振荡模式寄存器 (RCM) 的格式

地址：FFA0H 复位时：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。

在产生复位信号时，MOC 为“80H”。

图 5-5 主 OSC 控制寄存器 (MOC) 的格式

地址：FFA2H 复位时：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。

在产生复位信号时，MCM 为“00H”。

图 5-6 主时钟模式寄存器 (MCM) 的格式

地址：FFA1H 复位时：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 ( $f_{IH}$ )	高速内部振荡时钟 ( $f_{IH}$ )
0	1		高速系统时钟 ( $f_{XH}$ )
1	0	高速系统时钟 ( $f_{XH}$ )	
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“ $f_{IL}$ ”、“ $f_{IL}/2^7$ ”或“ $f_{IL}/2^9$ ”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（通过 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”时，OSTC 为“00H”。

图 5-7 振荡稳定时间计数器状态寄存器 (OSTC) 的格式

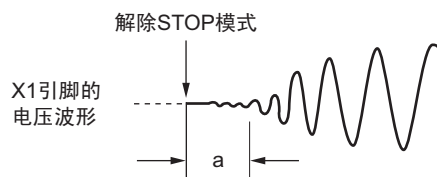
地址: FFA3H 复位时: 00H R

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	
						$f_X=10\text{MHz}$
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”，并且一直保持为“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。

在产生复位信号时，OSTS 为“05H”。

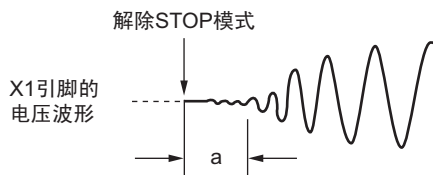
图 5-8 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFA4H 复位时: 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 $\mu\text{s}$
0	1	0	$2^{13}/f_X$	819.2 $\mu\text{s}$
0	1	1	$2^{14}/f_X$	1.64 $\mu\text{s}$
1	0	0	$2^{15}/f_X$	3.27 $\mu\text{s}$
1	0	1	$2^{16}/f_X$	6.55 $\mu\text{s}$
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

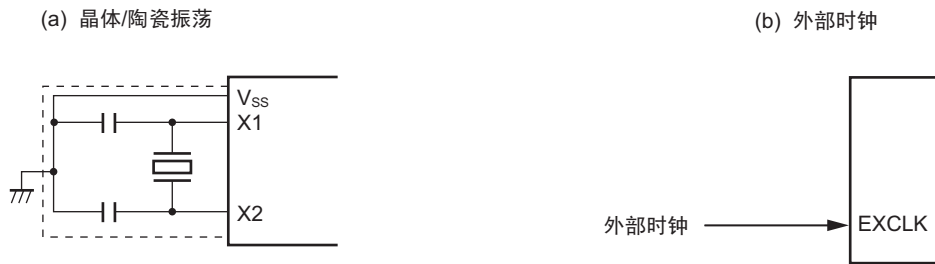
## 5.4 系统时钟振荡电路

### 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 (1/2)

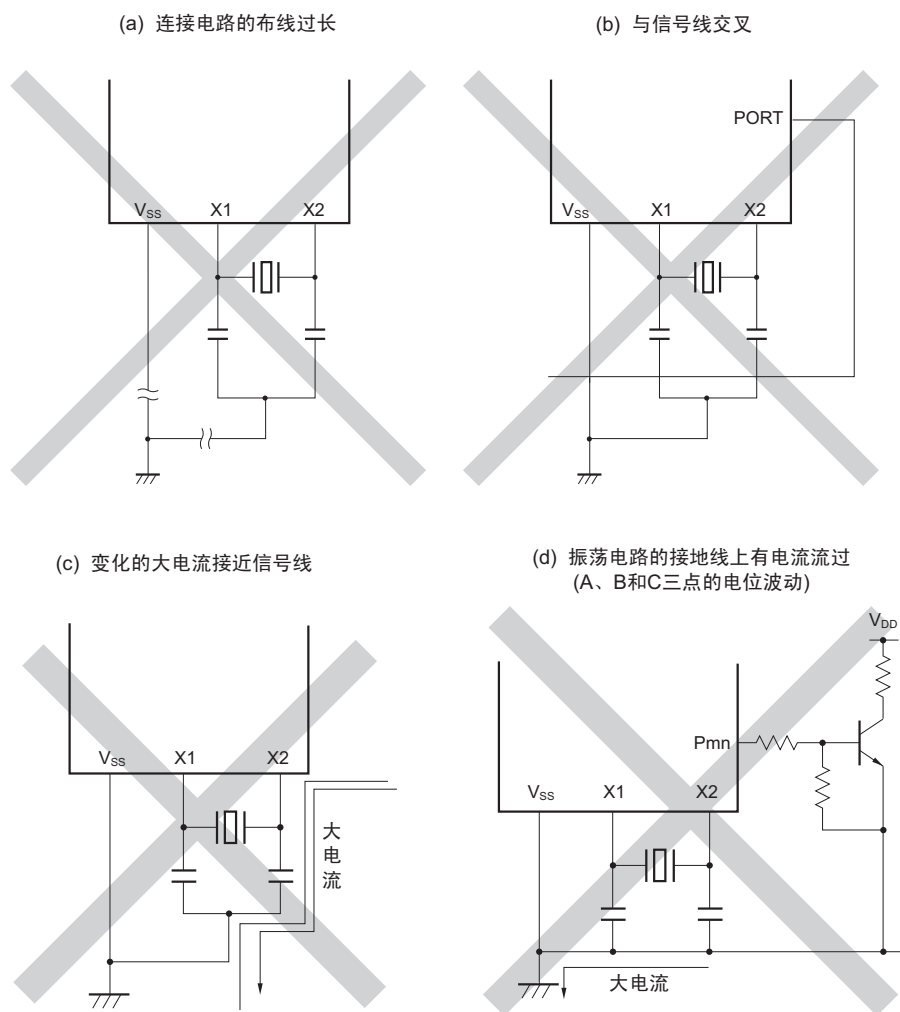
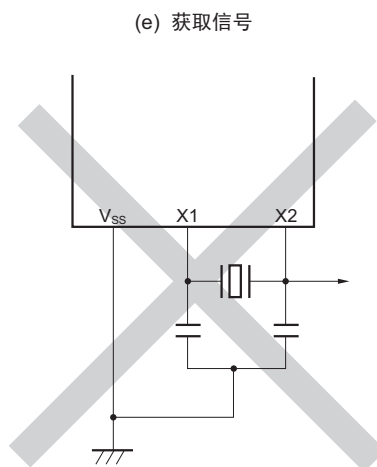


图 5-10 谐振器的错误连接示例 (2/2)



### 5.4.2 高速内部振荡电路

R7F0C30x、R7F0C31x 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

R7F0C30x、R7F0C31x 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  $f_{IH}$
- 低速内部振荡时钟  $f_{IL}$
- CPU 时钟  $f_{CPU}$
- 外围硬件时钟  $f_{PRS}$

在 R7F0C30x、R7F0C31x 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

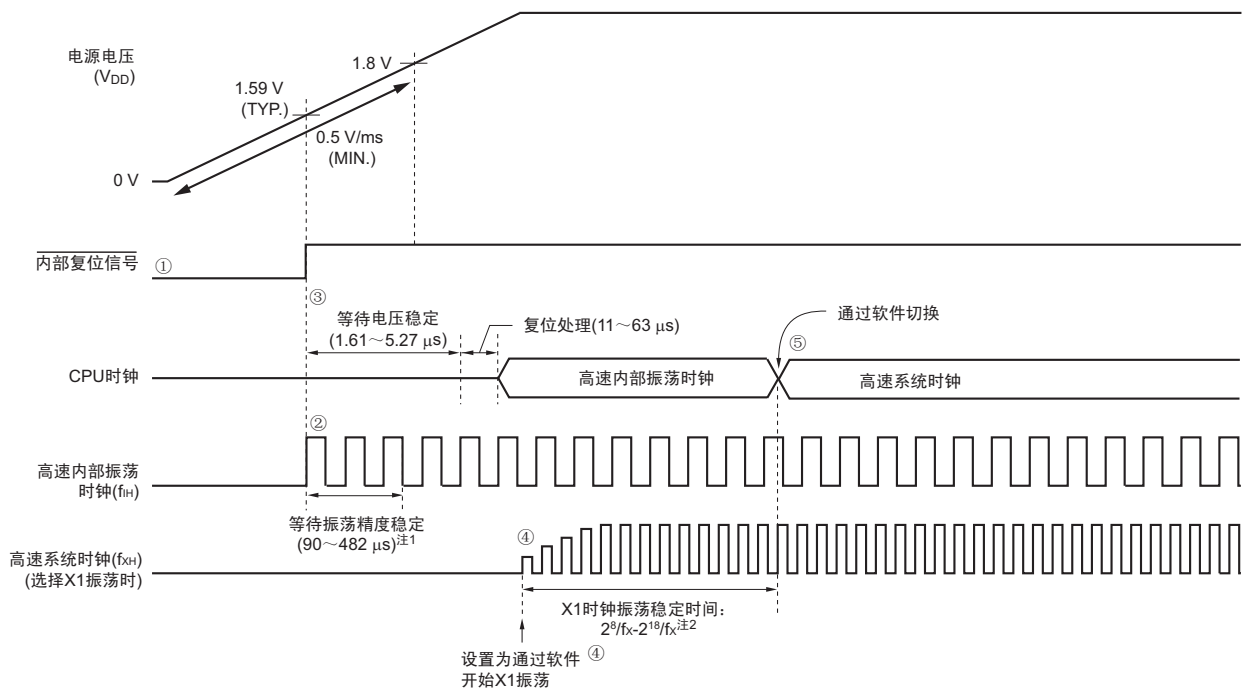
在复位解除后因损坏或连接不良等原因导致 X1 时钟不运行时，如果默认 CPU 时钟为 X1 时钟，设备就从此时刻开始无法运行；但如果 CPU 的启动时钟为高速内部振荡时钟，设备就可在复位解除后通过高速内部振荡时钟启动。因此，可通过如由软件识别复位源、执行异常时的安全处理等最低限度的运行，安全结束系统。

### (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“LVI 默认启动功能停止”的情况 (选项字节: LVISTART=0))



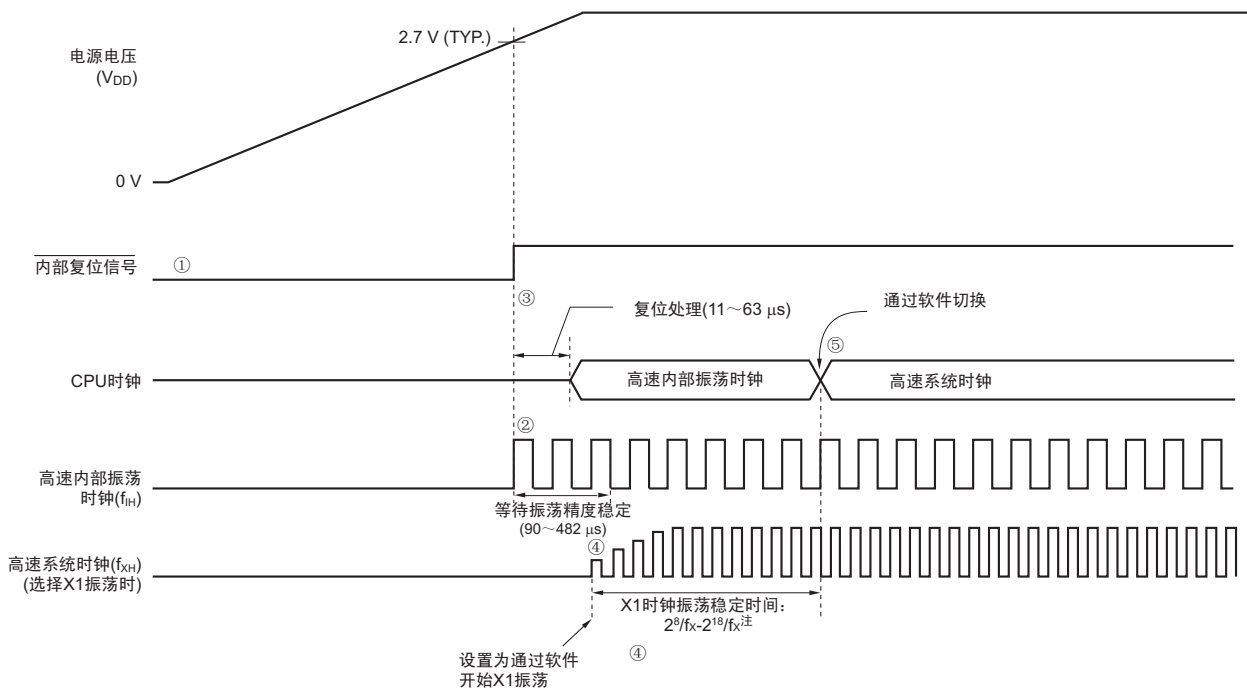
- ① 接通电源后，通过上电清除（POC）电路产生内部复位信号。
- ② 如果电源电压超过 1.59 V(TYP.)，就解除复位，并且高速内部振荡器自动开始振荡。
- ③ 如果电源电压以 0.5V/ms 的斜率上升，就在复位解除并且经过电源 / 稳压器的电压稳定等待时间后，执行复位处理，之后 CPU 以高速内部振荡时钟开始运行。
- ④ 设置为通过软件开始 X1 时钟振荡（参照“5.6.1 高速系统时钟的控制示例”的(1)）。
- ⑤ 如果要将在 CPU 时钟切换为 X1 时钟，必须在等待时钟振荡稳定后，通过软件设置切换（参照“5.6.1 高速系统时钟的控制示例”的(3)）。

- 注 1. 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。
2. 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. 如果从接通电源到达到 1.8V 时的电压的上升斜率低于 0.5V/ms(MIN.)，就在接通电源到达到 1.8V 期间，输入低电平到  $\overline{\text{RESET}}$  引脚，或者通过选项字节设置为“LVI 默认启动功能运行”（LVISTART=1）（参照图 5-12）。在输入低电平到  $\overline{\text{RESET}}$  引脚直到电压达到 1.8V 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. 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“LVI 默认启动功能运行”的情况 (选项字节: LVISTART=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，需要电压稳定等待时间 (1.61 ~ 5.27ms)。如果电压从 1.59V(TYP.) 达到到 2.7V(TYP.) 的时间比电压稳定等待时间短，就在复位处理前自动产生电源稳定等待时间。
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 引脚可用作输入端口。

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

下列情况的设置步骤示例如下所述。

- (1) 开始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) 开始 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 1.** 在 X1 时钟运行时，不能改写 EXCLK 和 OSCSEL。

2. 在电源电压达到使用时钟的工作电压（参照“第 24 章 电特性”）后，设置 X1 时钟。

#### (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 1.** 在外部主系统时钟运行时，不能改写 EXCLK 和 OSCSEL。

2. 在电源电压达到使用时钟的工作电压（参照“第 24 章 电特性”）后，设置外部主系统时钟。



## (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) 开始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 选择主系统时钟作为CPU时钟，分频比的选择 (PCC 寄存器)

主系统时钟提供给CPU。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15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（PCC 和 MCM 寄存器）

通过 CLS 和 MCS 确认 CPU 时钟是否以高速系统时钟以外的时钟运行。

在 CLS=0、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 或者 CLS=1。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<sup>注1</sup>

- 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”<sup>注2</sup>。

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<sup>注</sup>。  
(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<sup>注1</sup>”)。
  - 高速系统时钟振荡<sup>注</sup>  
(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) 开始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 ( $f_{IH}$ )	高速内部振荡时钟 ( $f_{IH}$ )
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）  
主系统时钟提供给CPU。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15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（PCC和MCM寄存器）**

通过CLS和MCS确认CPU时钟以高速内部振荡时钟以外的时钟运行。

在CLS=0、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或者CLS=1。此外，必须停止以高速系统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  $f_{IL}$  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

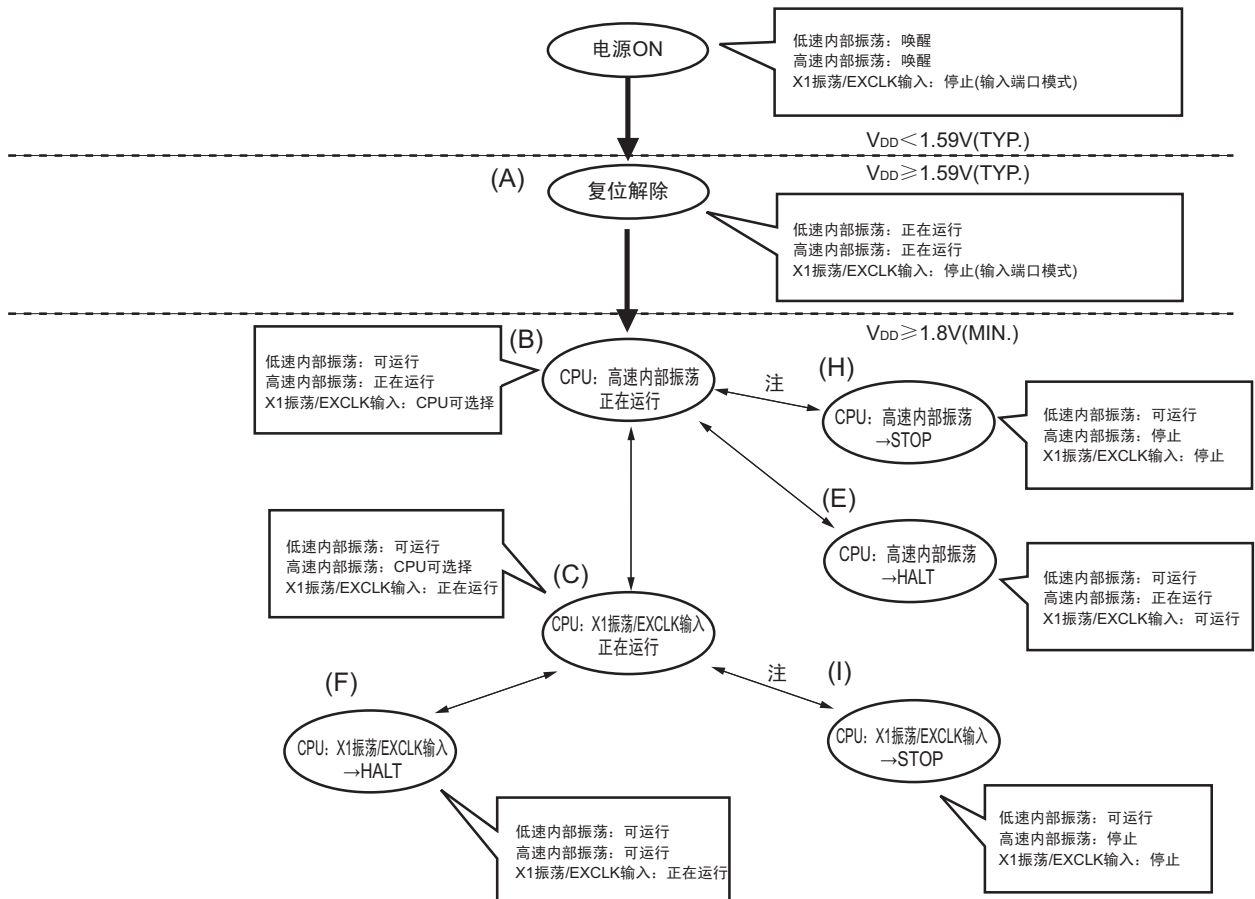
提供的时钟		XSEL	MCM0	EXCLK
作为 CPU 时钟提供的时钟	作为外围硬件时钟提供的时钟			
高速内部振荡时钟		0	x	x
高速内部振荡时钟	X1 时钟	1	0	0
	外部主系统时钟	1	0	1
X1 时钟		1	1	0
外部主系统时钟		1	1	1

**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“LVI 默认启动模式功能停止”的情况 (选项字节: LVISTART=0))



**注意** 在设置为“LVI 默认启动功能运行”时 (选项字节: LVISTART=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 ~ 63μs)，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)	0	1	0	必须确认	1	1
(A)→(B)→(C) (外部主系统时钟)	1	1	0	不必确认	1	1

注意 在电源电压达到设置时钟的工作电压后, 设置时钟 (参照“第 24 章 电特性”)。

(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 <sup>注</sup>	MCM0
(B)→(C) (X1时钟)	0	1	0	必须确认	1	1
(B)→(C) (外部主系统时钟)	1	1	0	不必确认	1	1

如果已设置,  
就不需要再设置。

如果CPU正在以高速系统  
时钟运行, 就不需要设置。

注 复位解除后, 只能进行 1 次设置。如果已设置, 就无需再设置。

注意 在电源电压达到设置时钟的工作电压 (参照“第 24 章 电特性”)后, 设置时钟。

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

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CPU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

(SFR寄存器的设置顺序) →

状态转移	设置	
(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



### 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 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 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	1	0	1	0	0	1	1	1	0	0
0	0	0	8 个时钟			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			4 个时钟		
0	1	1	2 个时钟			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			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		$1 + 2f_{IH}/f_{XH}$ 个时钟
1	$1 + 2f_{XH}/f_{IH}$ 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  $f_{IH}=8\text{MHz}$ 、 $f_{XH}=10\text{MHz}$  振荡时）

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

### 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 (CPU 时钟以高速内部振荡时钟运行)	MSTOP=1
外部主系统时钟		

### 5.6.9 外围硬件和源时钟

R7F0C30x、R7F0C31x 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 ( $f_{IL}$ )	外围硬件引脚的 外部时钟
16 位定时器 / 事件计数器 00		○	x	○ (TI000 引脚) 注
8 位定时器 / 事件计数器	51	○	x	○ (TI51 引脚) 注
8 位定时器	H1	○	○	x
看门狗定时器		x	○	x
A/D 转换器		○	x	x
串行接口	UART0	○	x	x

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

R7F0C30x、R7F0C31x 的所有产品都装有 16 位定时器 / 事件计数器 00。

16 位定时器 / 事件计数器 00 具有以下功能。

**注意** 关于 16 引脚产品，只有在通过 MUXSEL 寄存器指定了 TI000、TI010、TO00 功能时才能使用定时器输入 / 输出引脚。

#### (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。

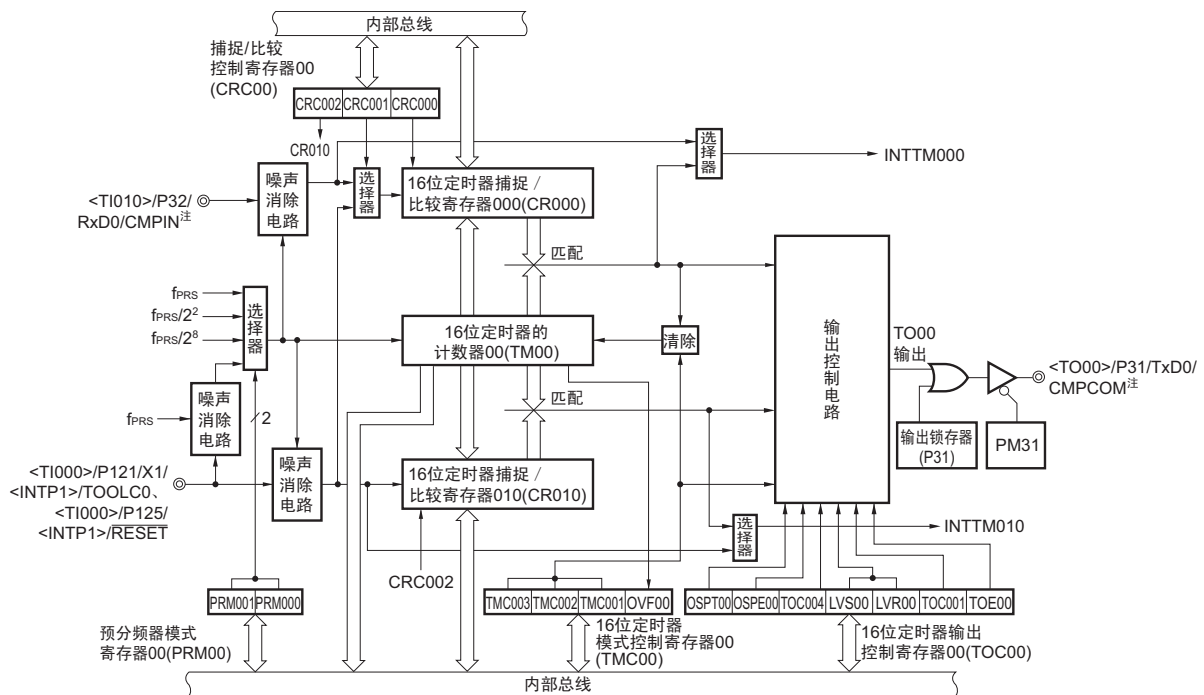
表 6-1 16 位定时器 / 事件计数器 00 的结构

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 位定时器模式控制器寄存器 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) 端口复用切换控制寄存器 (MUXSEL) (只限 16 引脚产品) 端口模式寄存器 3、12 (PM3、PM12) 注 端口寄存器 3、12 (P3、P12) 注

注 16 引脚产品: PM3、PM12、P3、P12  
20 引脚产品: PM3、P3

图 6-1 16 位定时器 / 事件计数器 00 的框图 (1/2)

(1) 16 引脚产品



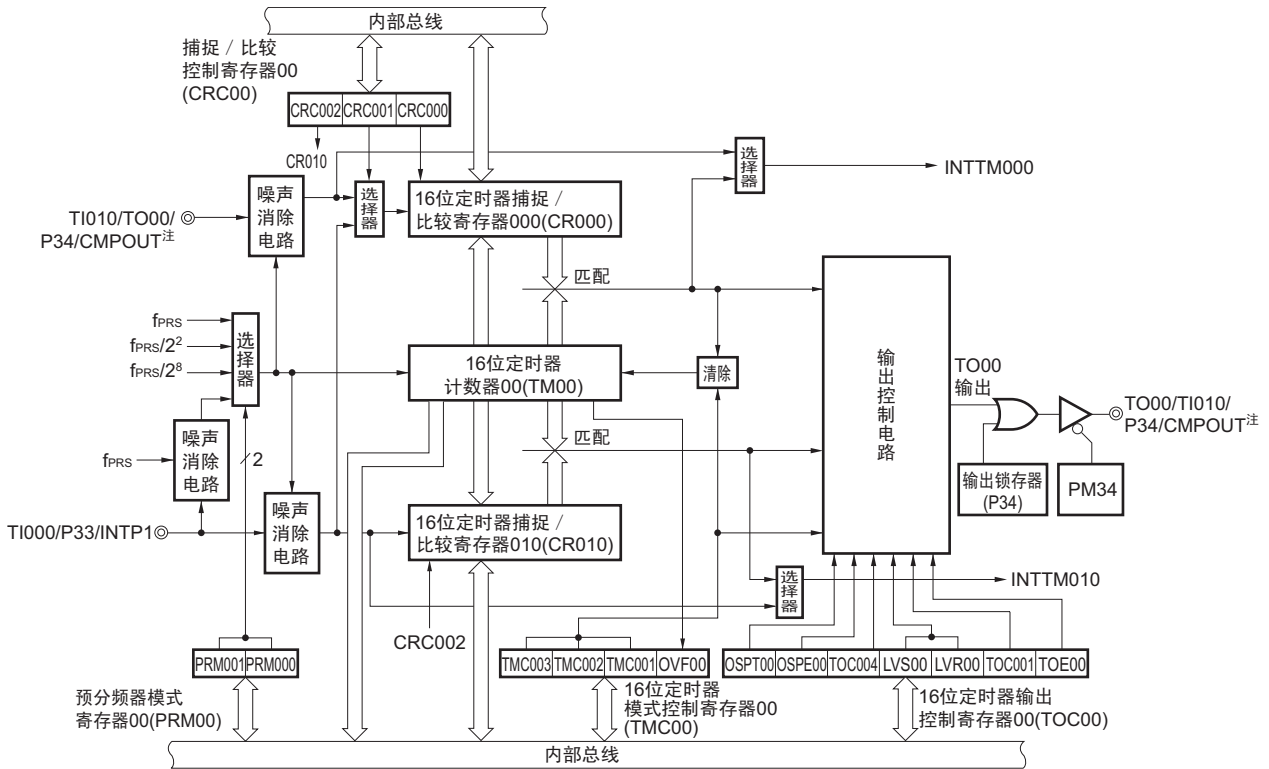
注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 (内置比较器的产品)。

- 注意 1. 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。
- 2. 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。

备注 可通过设置端口复用切换控制寄存器 (MUXSEL) 指定上图 < > 内的功能。

图 6-1 16 位定时器 / 事件计数器 00 的框图 (2/2)

(2) 20 引脚产品



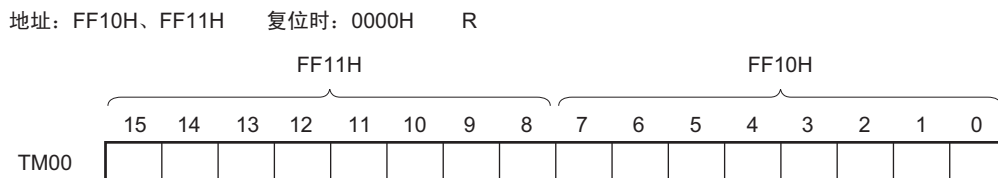
注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置比较器的产品)。

- 注意 1. P34 引脚不能同时使用 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 (TMC0) 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 CR00。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 (TMC003、TMC002) 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。

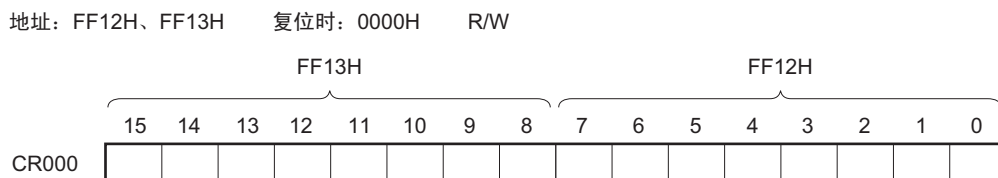
在定时器停止 (TMC003 和 TMC002=00) 时改写 CR000。

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。

在产生复位信号时，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 (INTTM00)。保持该值直到被改写为止。

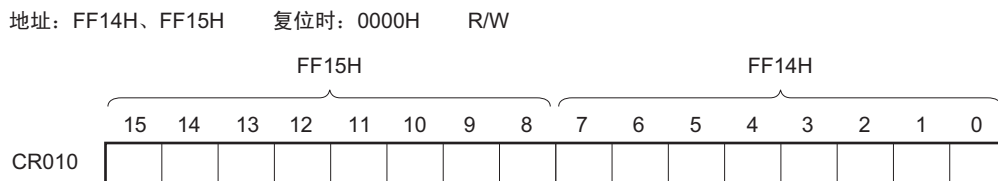
**注意** 即使输入捕捉触发，设置为比较模式的 CR000 也不执行捕捉运行。

## (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。

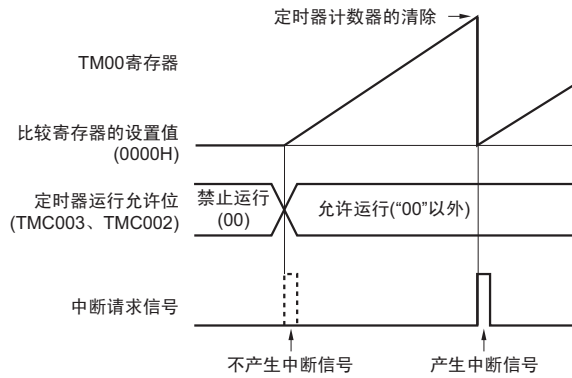
## (iii) CR000、CR010 用作比较寄存器时的设置范围

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 ≤ FFFFH	0000H 注 ≤ M ≤ 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 注 ≤ N ≤ FFFFH	0000H 注 ≤ M ≤ FFFFH
作为自由运行定时器的运行		
PPG 的输出运行	M < N ≤ FFFFH	0000H 注 ≤ M < N
单次脉冲的输出运行	0000H 注 ≤ N ≤ FFFFH (N ≠ M)	0000H 注 ≤ M ≤ FFFFH (M ≠ 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 <sup>注</sup>	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)
- 端口复用切换控制寄存器 (MUXSEL) (只限 16 引脚产品)
- 端口模式寄存器 3、12 (PM3、PM12) 注
- 端口寄存器 3、12 (P3、P12) 注

注 16 引脚产品: PM3、PM12、P3、P12

20 引脚产品: PM3、P3

#### (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。

在产生复位信号时, TMC00 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 复位时: 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。

在产生复位信号时, CRC00 为“00H”。

图 6-6 捕捉 / 比较控制寄存器 00 (CRC00) 的格式

地址: FF88H 复位时: 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 <sup>注</sup>

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。

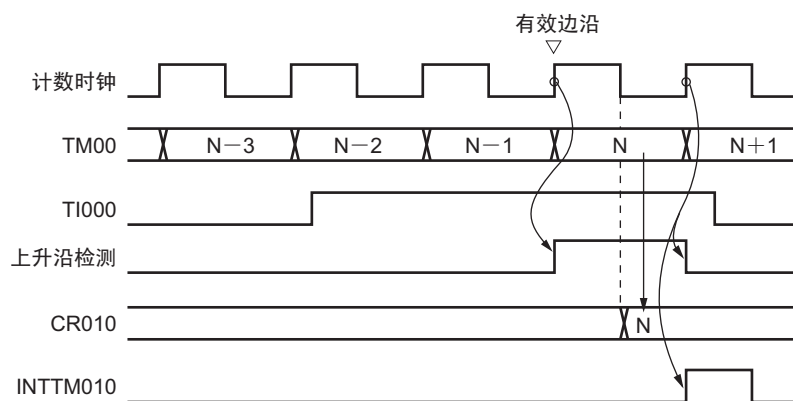
CRC000	选择 CR000 的运行模式
0	作为比较寄存器运行
1	作为捕捉寄存器运行

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)

TOCC0 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。

在产生复位信号时，TOC00 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 复位时: 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 复位时: 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”。
- 在 16 引脚产品中, 实际的 <TO00>/TxD0/CMPCOM 注 1/P31 引脚的 TO00 输出取决于 PM31 和 P31。
- 在 20 引脚产品中, 实际的 TO00/TI010/CMPOUT 注 2/P34 引脚的 TO00 输出取决于 PM34 和 P34。

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

- 注 1. 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 (内置比较器的产品)。
- 注 2. 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置比较器的产品)。

## (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。

在产生复位信号时, PRM00 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. 20 引脚产品的 P34 引脚不能同时用作 TI010 有效边沿和定时器输出 (TO00)。只能选择其中一项功能。

图 6-9 预分频器模式寄存器 00 (PRM00) 的格式

地址: FF87H 复位时: 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	计数时钟的选择注 1		
			$f_{PRS}=2\text{MHz}$	$f_{PRS}=5\text{MHz}$
0	0	$f_{PRS}$	2MHz	5MHz
0	1	$f_{PRS}/2^2$	500kHz	1.25MHz
1	0	$f_{PRS}/2^8$	7.81kHz	19.53kHz
1	1	TI000 有效边沿注 2、注 3		

注 1. 在外围硬件时钟 ( $f_{PRS}$ ) 以高速系统时钟 ( $f_{XH}$ ) 运行 (XSEL=1) 时,  $f_{PRS}$  的工作频率因电源电压而不同。

- $V_{DD}=2.7\sim 5.5\text{V}$ :  $f_{PRS}\leq 10\text{MHz}$
- $V_{DD}=1.8\sim 2.7\text{V}$ :  $f_{PRS}\leq 5\text{MHz}$

2. 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。

3. 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) 端口复用切换控制寄存器 (MUXSEL) (仅限 16 引脚产品)

MUXSEL 设置 16 引脚产品中 TI000 引脚、TI010 引脚和 TO00 引脚的分配。

通过 1 位或 8 位存储器操作指令设置 MUXSEL。

在产生复位信号时，MUXSEL 为“00H”。

图 6-10 端口复用切换控制寄存器 (MUXSEL) 的格式

地址: FF39H 复位时: 00H R/W

符号	<7>	<6>	<5>	<4>	3	2	<1>	<0>
MUXSEL	INTP1SEL1	INTP1SELO	TM00SEL1	TM00SELO	0	0	TI010SEL	TO00SEL

TM00SEL1	TM00SELO	16 位定时器 / 事件计数器 00 的输入引脚 (TI000) 分配
0	0	(默认)
0	1	P121/TI000
1	0	P125/TI000
1	1	禁止设置

TI010SEL	16 位定时器 / 事件计数器 00 的输入引脚 (TI010) 分配
0	(默认)
1	P32/TI010

TO00SEL	16 位定时器 / 事件计数器 00 的输出引脚 (TO00) 分配
0	(默认)
1	P31/TO00

## (6) 端口模式寄存器 3、12 (PM3、PM12)

PM3 和 PM12 以 1 位单位设置端口 3 和端口 12 的输入 / 输出。

- 16 引脚产品

将 P31/TxD0/<TO00>/CMPCOM 注引脚用作定时器输出时，必须将 PM31 和 P31 的输出锁存器置“0”。

将 P32/RxD0/<TI010>/CMPIN 注、P121/X1/<TI000>/<INTP1>/TOOLC0 引脚用作定时器输入时，必须将 PM32 和 PM121 置“1”。此时，P32 和 P121 的输出锁存器可以是“0”也可以是“1”。

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114（内置比较器的产品）。

备注 通过端口复用切换控制寄存器（MUXSEL）的设置将 P125 引脚分配为定时器输入（TI000）功能时，无需设置端口模式寄存器和端口寄存器。

- 20 引脚产品

将 P34/TO00/TI010/CMPOUT 注引脚用作定时器输出时，必须将 PM34 和 P34 的输出锁存器置“0”。

将 P33/TI000/INTP1、P34/TI010/TO00/CMPOUT 注引脚用作定时器输入时，必须将 PM33 和 PM34 置“1”。此时，P33 和 P34 的输出锁存器可以是“0”也可以是“1”。

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116（内置比较器的产品）。

通过 1 位或 8 位存储器操作指令设置 PM3 和 PM12。

在产生复位信号时，PM3 和 PM12 为“FFH”。

图 6-11 端口模式寄存器 3、12 (PM3、PM12) 的格式

地址：FF23H 复位时：FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34 注	PM33 注	PM32	PM31	PM30

PM3n	P3n 引脚的输入 / 输出模式选择 (n=0 ~ 4)
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

地址：FF2CH 复位时：FFH R/W

符号	7	6	5	4	3	2	1	0
PM12	1	1	1	1	1	PM122	PM121	1

PM12n	P12n 引脚的输入 / 输出模式选择 (n=1、2)
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

注 只限 20 引脚产品。



## 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 (6) 端口模式寄存器 3、12 (PM3、PM12)”。
- 2. 有关 INTTM000 信号的中断允许, 请参照“第 14 章 中断功能”。

图 6-12 间隔定时器运行的框图

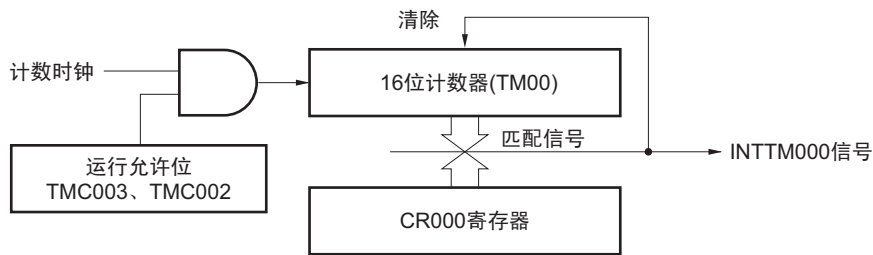


图 6-13 间隔定时器运行的基本时序示例

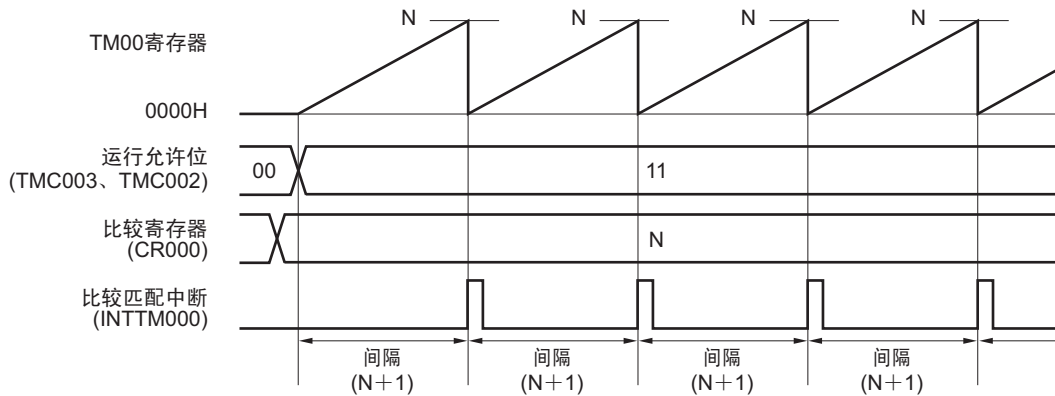
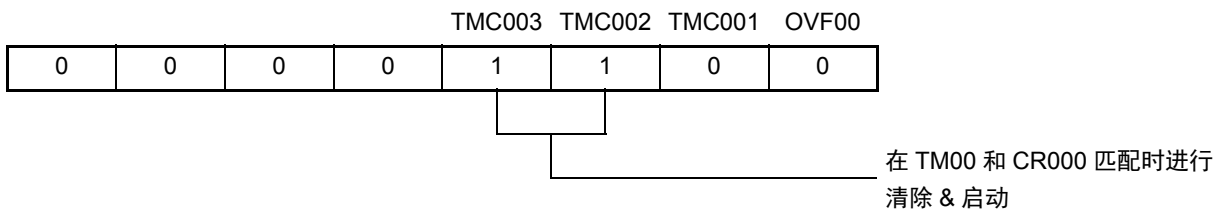
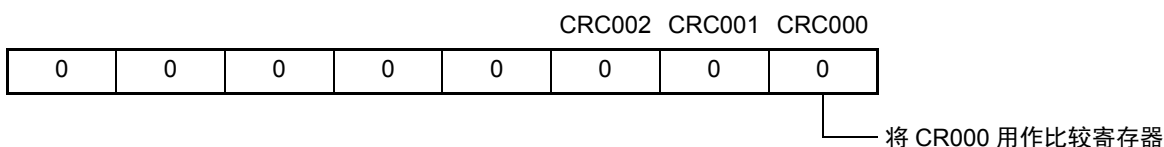


图 6-14 间隔定时器运行时的寄存器设置内容示例

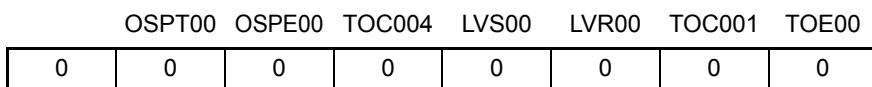
## (a) 16 位定时器模式控制寄存器 00 (TMC00)



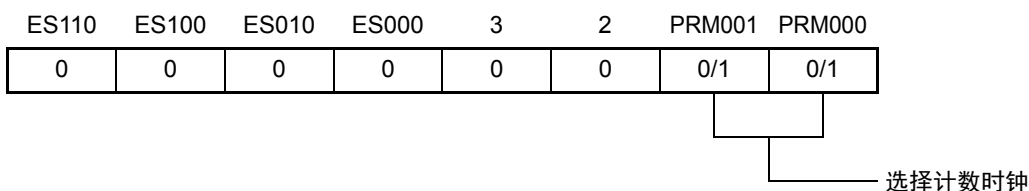
## (b) 捕捉 / 比较控制寄存器 00 (CRC00)



## (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

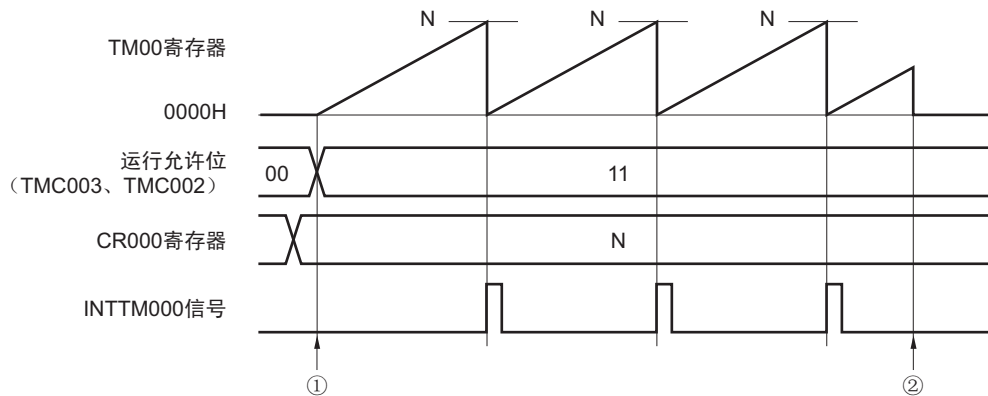
禁止将 CR000 设置为“0000H”。

## (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。

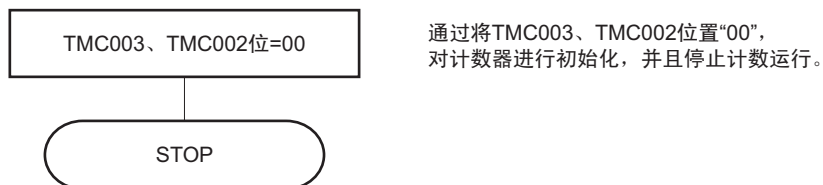
图 6-15 间隔定时器功能期间的软件处理示例



## 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 (6) 端口模式寄存器 3、12（PM3、PM12）”。

2. 有关 INTTM000 信号的中断允许，请参照“第 14 章 中断功能”。

图 6-16 方波输出运行的框图

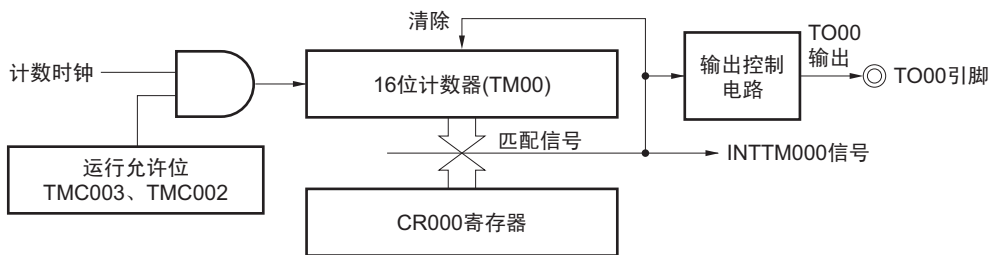


图 6-17 方波输出运行的基本时序示例

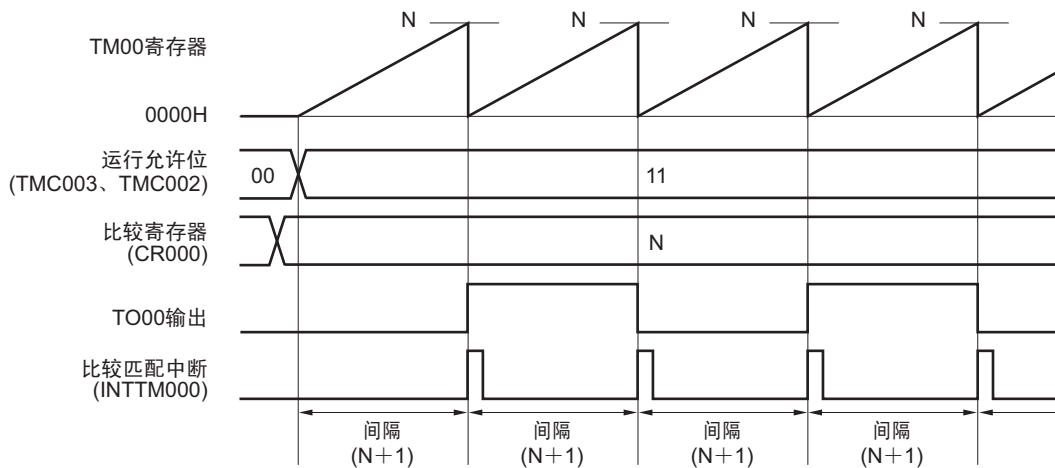
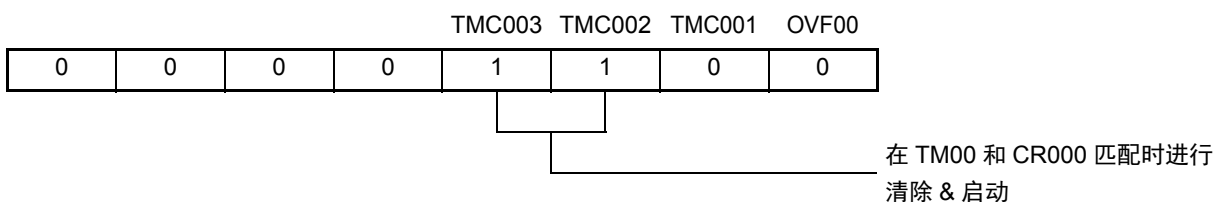
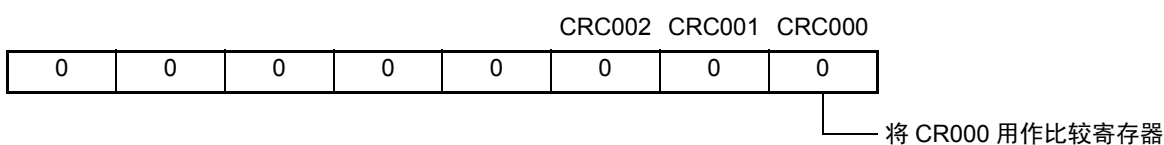


图 6-18 方波输出运行时的寄存器设置内容示例 (1/2)

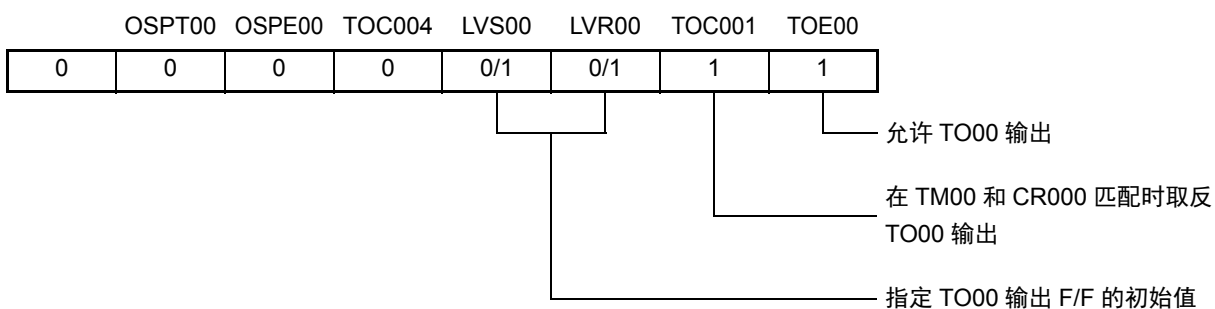
## (a) 16 位定时器模式控制寄存器 00 (TMC00)



## (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)

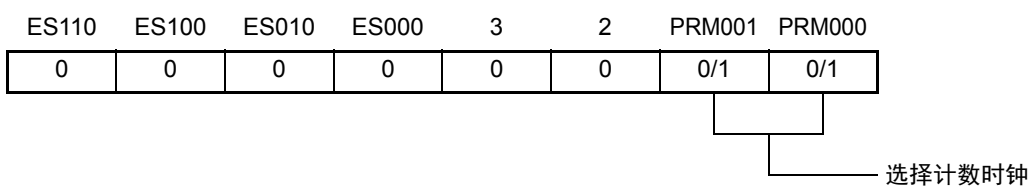


图 6-18 方波输出运行时的寄存器设置内容示例 (2/2)

## (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 =  $1 \div \{2 \times (M+1) \times \text{计数时钟周期}\}$

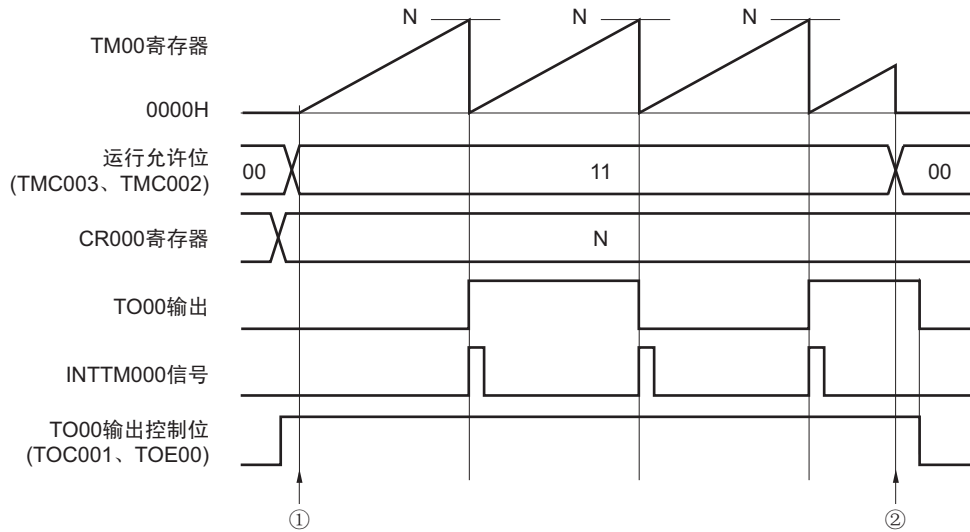
禁止将 CR000 设置为“0000H”。

## (g) 16 位捕捉 / 比较寄存器 010 (CR010)

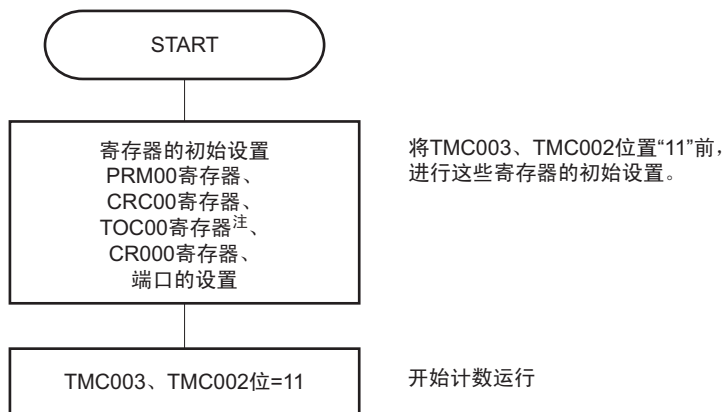
CR010 一般不用于方波输出功能。但是，如果 CR010 的设置值与 TM00 的值匹配，就发生比较匹配中断 (INTTM010)。

因此，通过中断屏蔽标志 (TMMK010) 进行屏蔽设置。

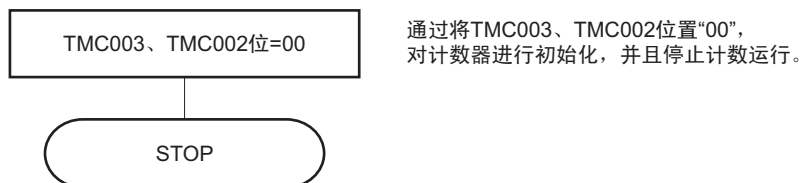
图 6-19 方波输出功能期间的软件处理示例



## 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 (6) 端口模式寄存器 3、12 (PM3、PM12)”。

2. 有关 INTTM000 信号的中断允许, 请参照“第 14 章 中断功能”。

图 6-20 作为外部事件计数器运行的框图

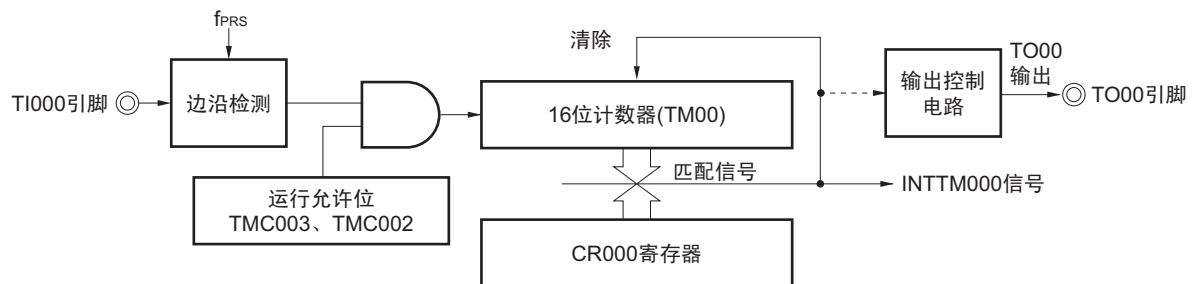
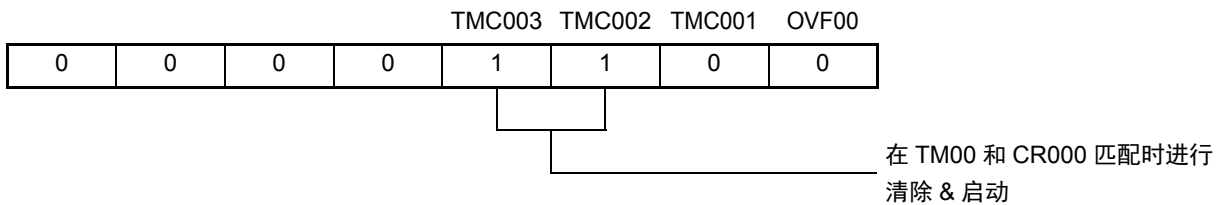


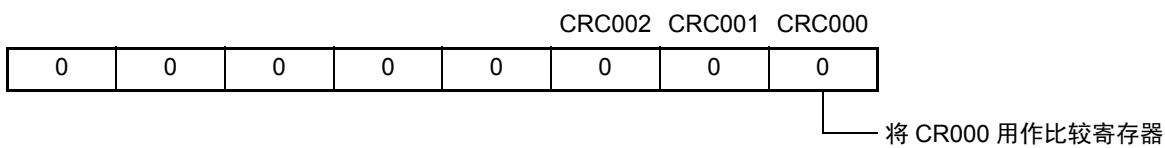


图 6-21 外部事件计数器运行时的寄存器设置内容示例 (1/2)

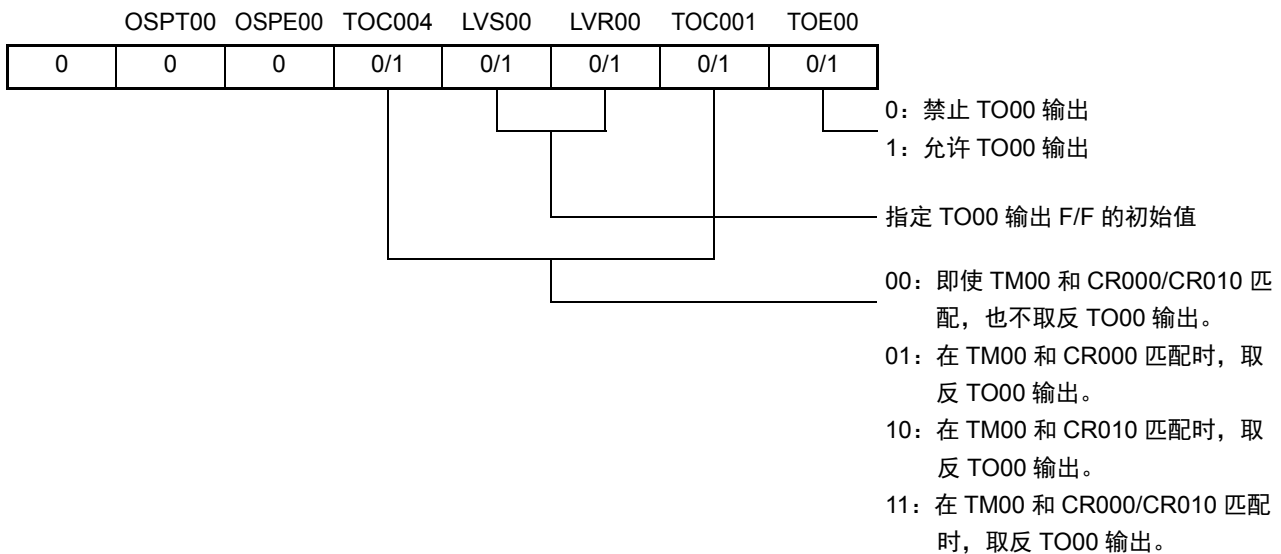
## (a) 16 位定时器模式控制寄存器 00 (TMC00)



## (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)

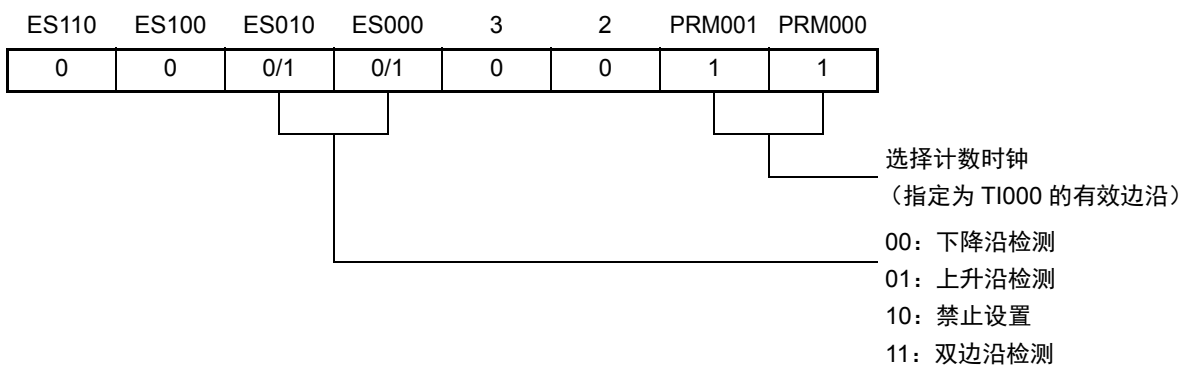


图 6-21 外部事件计数器运行时的寄存器设置内容示例 (2/2)

## (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)

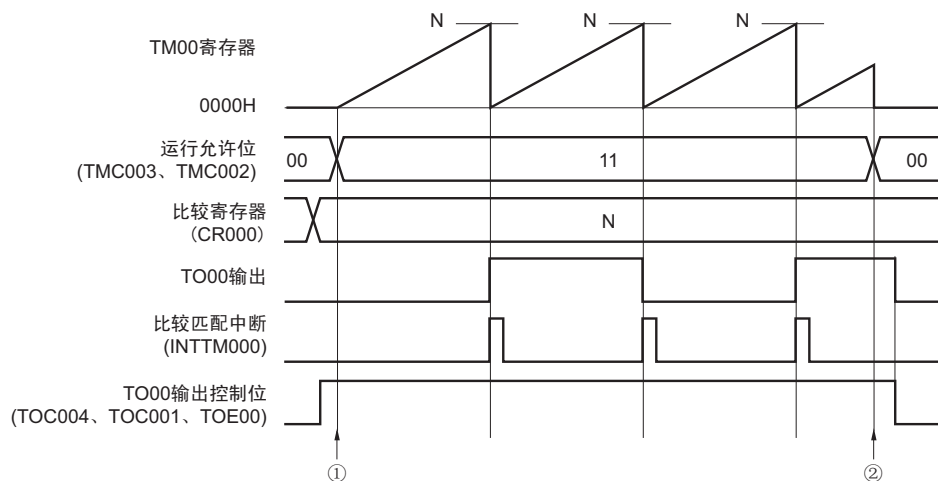
将 M 设置到 CR000 时，如果输入 M+1 次外部事件，就产生中断信号 (INTTM000)。禁止将 CR000 设置为“0000H”。

## (g) 16 位捕捉 / 比较寄存器 010 (CR010)

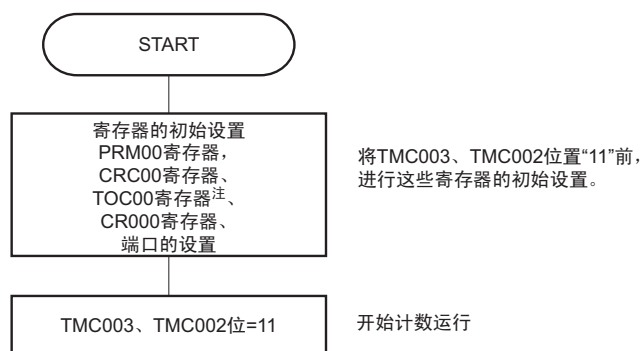
CR010 一般不作为部事件计数器运行。但是，如果 CR010 的设置值与 TM00 的值匹配，就发生比较匹配中断 (INTTM010)。

因此，通过中断屏蔽标志 (TMMK010) 进行屏蔽设置。

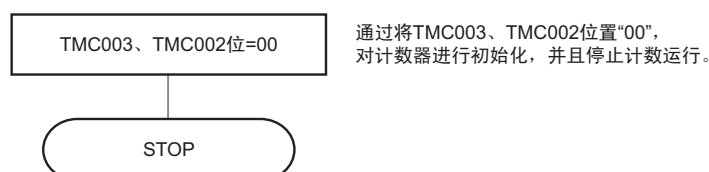
图 6-22 外部事件计数器运行时的软件处理示例



## 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 (6) 端口模式寄存器 3、12 (PM3、PM12)”。

**备注 2.** 有关 INTTM000 信号的中断允许, 请参照“第 14 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 6-23 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)

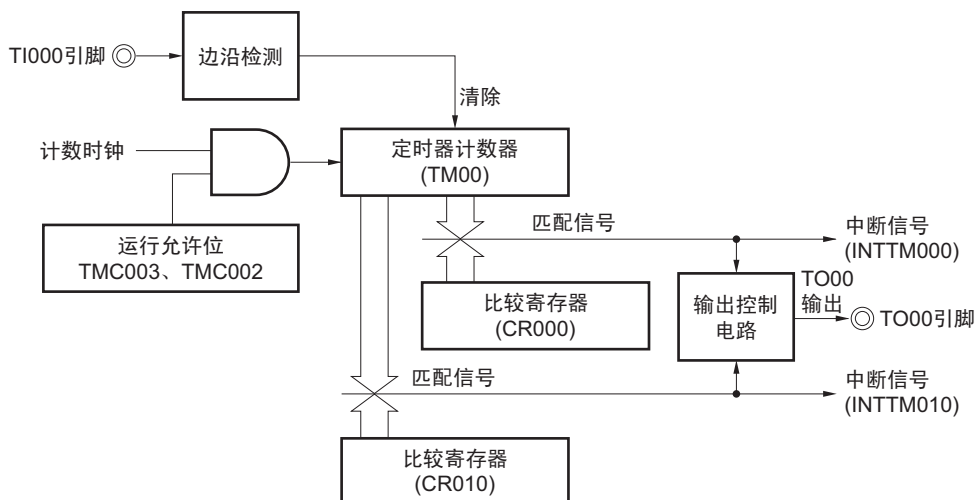
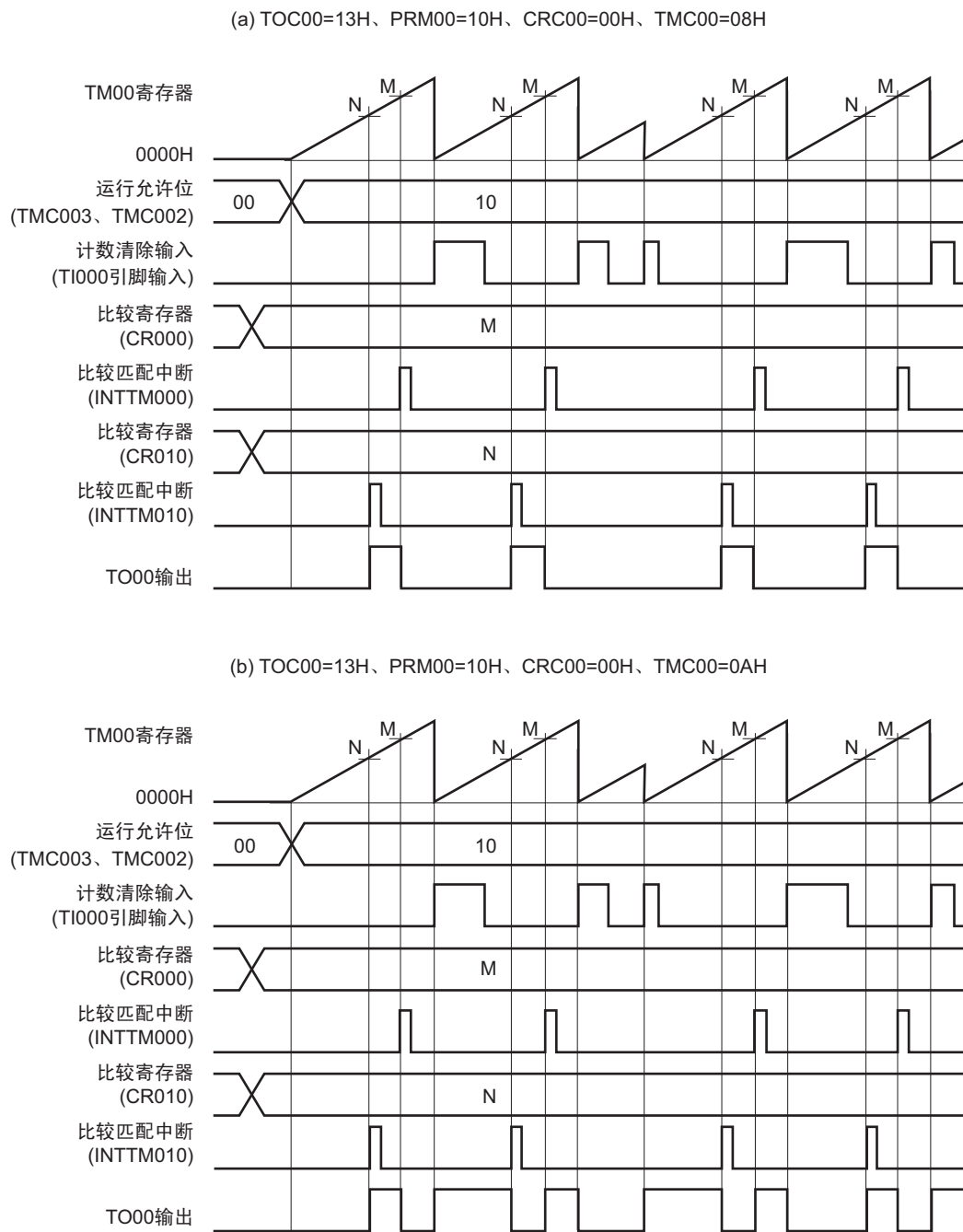


图 6-24 输入 TI000 引脚有效边沿时的清除 & 启动模式的时序示例  
(CR000: 比较寄存器、CR010: 比较寄存器)



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 6-25 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)

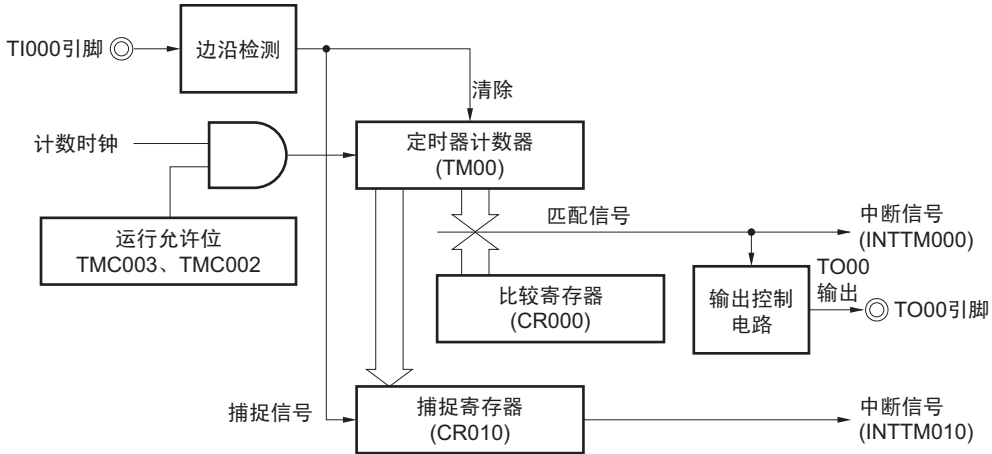
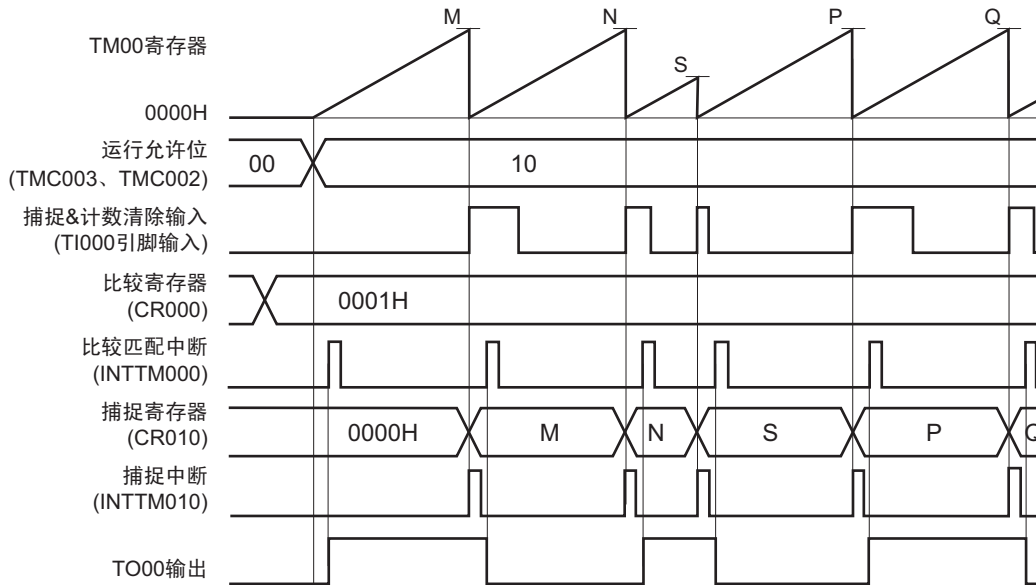


图 6-26 输入 TI000 引脚的有效边沿时的清除 & 启动模式的时序示例  
(CR000: 比较寄存器、CR010: 捕捉寄存器) (1/2)

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

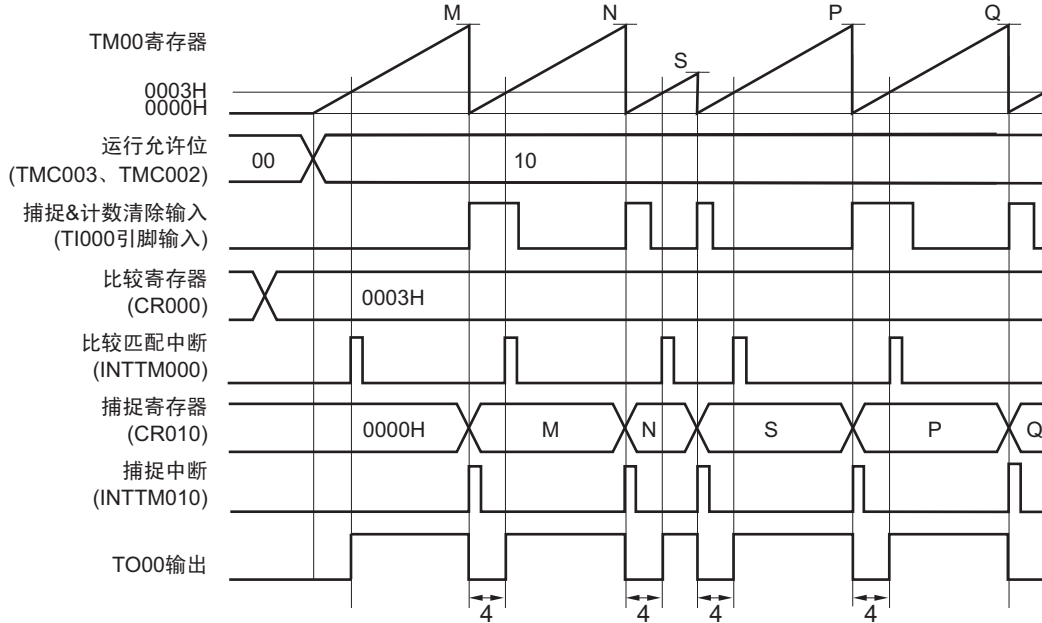


这是在进行捕捉 & 清除后，要取反 TO00 输出电平时的应用示例。

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 6-26 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 6-27 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)

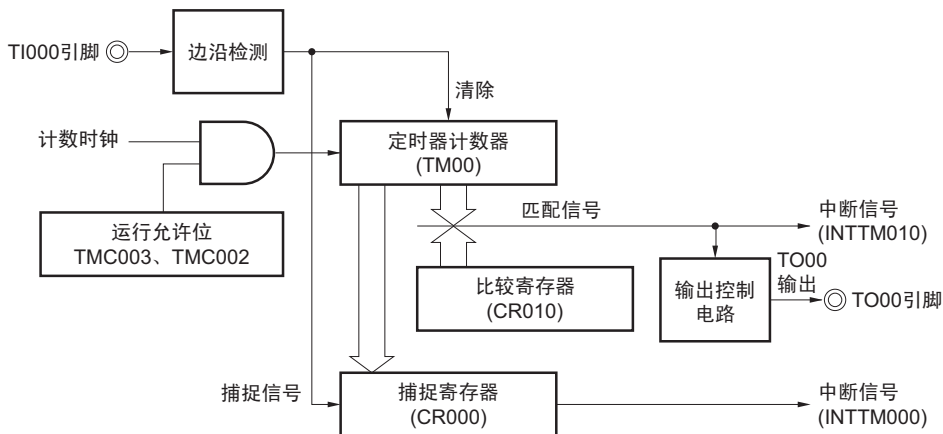
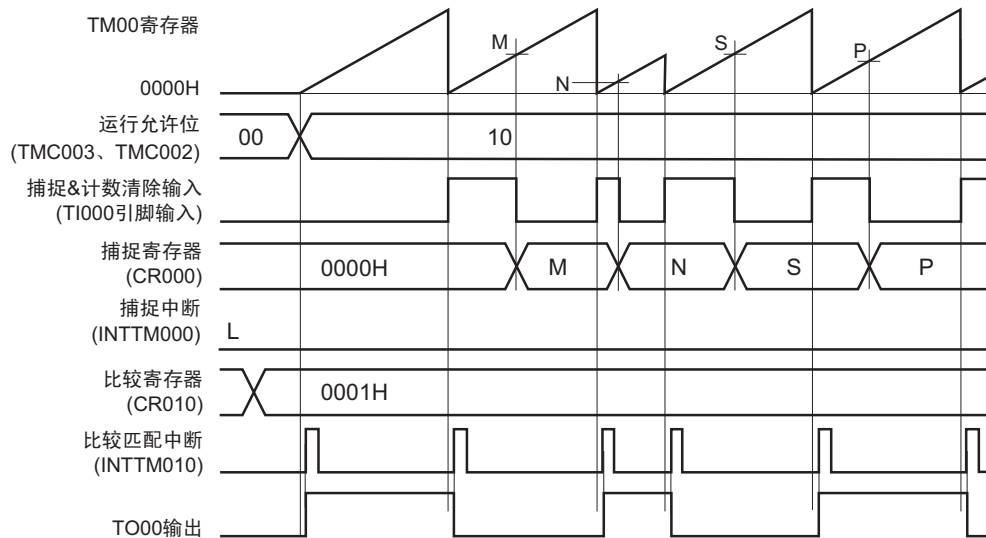


图 6-28 输入 TI000 引脚有效边沿时的清除 & 启动模式的时序示例  
(CR000: 捕捉寄存器、CR010: 比较寄存器) (1/2)

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



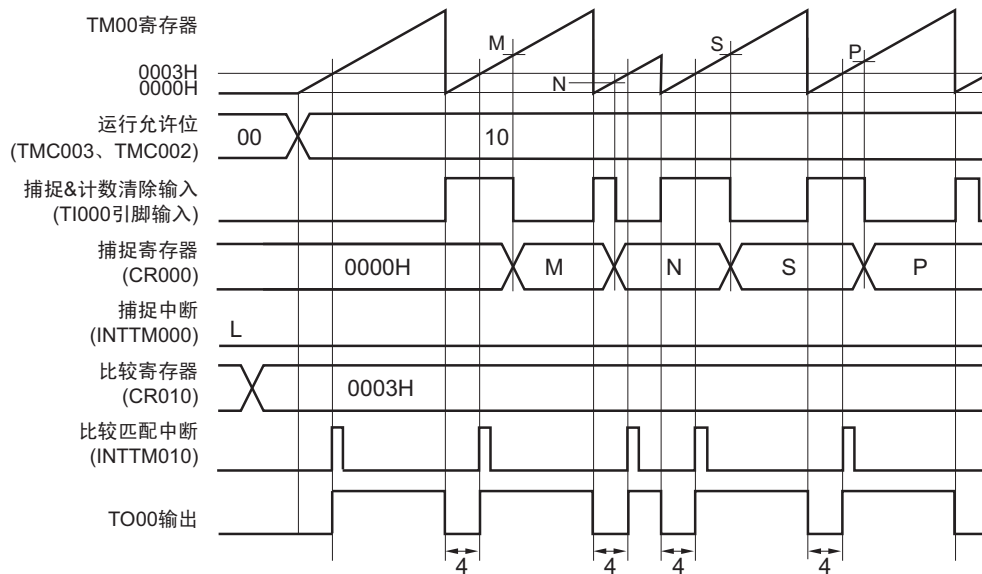
这是在进行捕捉 & 清除后，要取反 TO00 输出电平时的应用示例。

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 6-28 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。

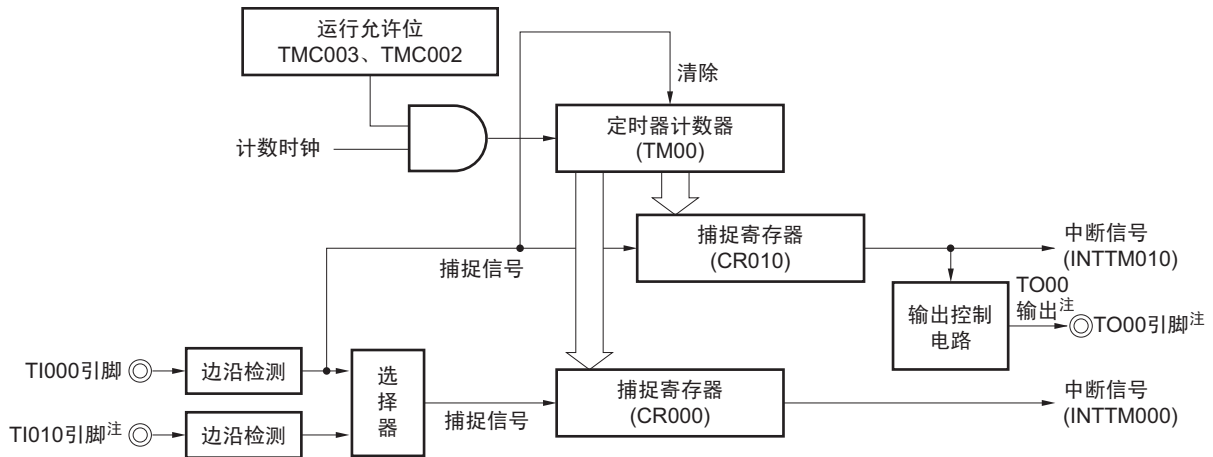
通过 TI000 引脚的上升沿检测，清除 TM00（0000H）。通过 TI000 引脚的下降沿检测，将计数值捕捉到 CR000。在因 TI000 引脚的上升沿检测引起的 TM00 清除（0000H）或者 TM00 和比较寄存器（CR010）匹配时取反 TO00 输出。

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)

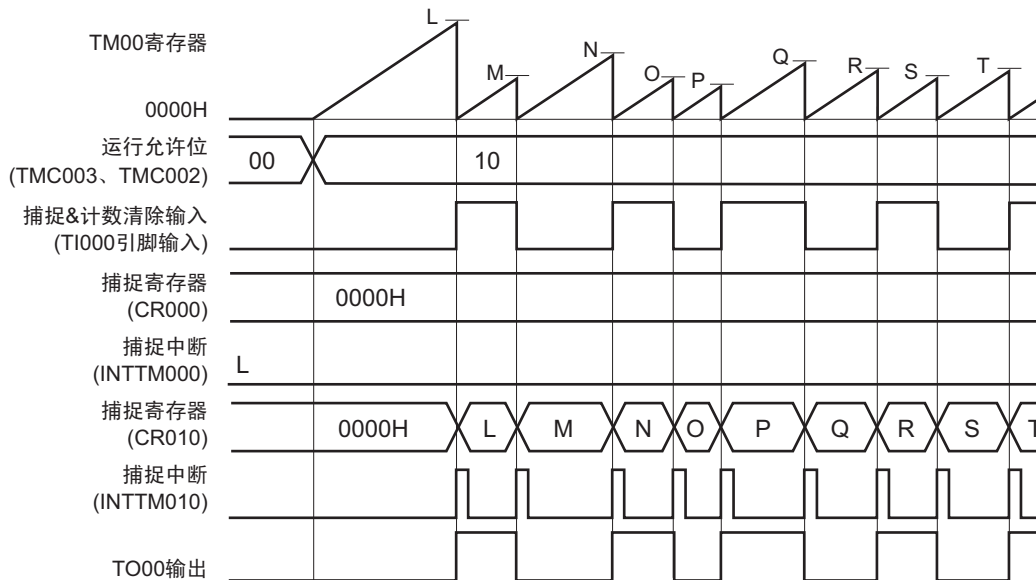
图 6-29 输入 TI000 引脚有效边沿时的清除 & 启动模式的框图  
(CR000: 捕捉寄存器、CR010: 捕捉寄存器)



注 在 20 引脚产品中使用 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)

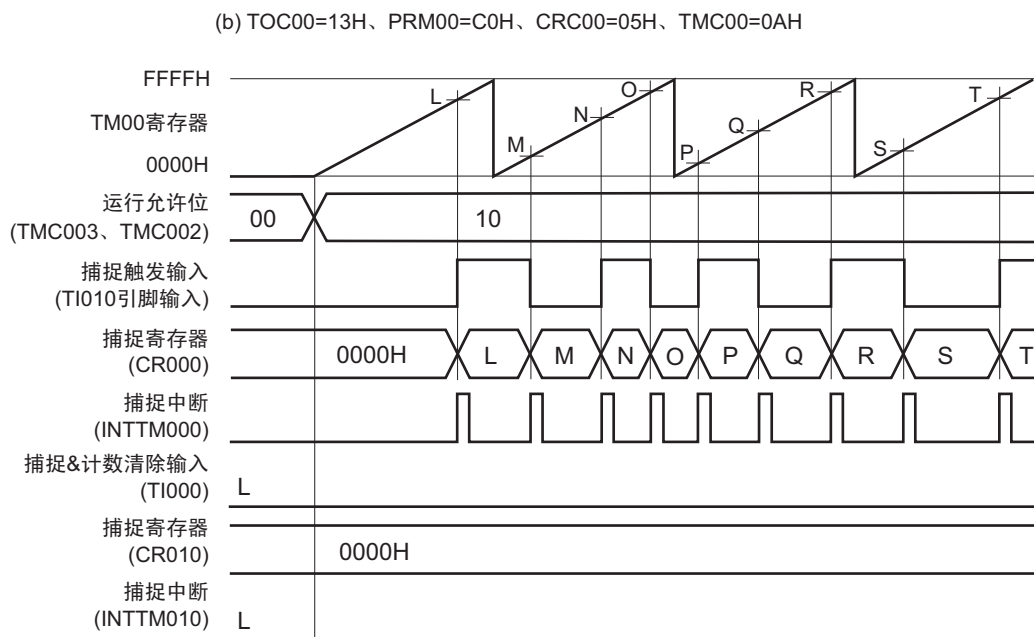
(a) TOC00=13H、PRM00=30H、CRC00=05H、TMC00=0AH



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。

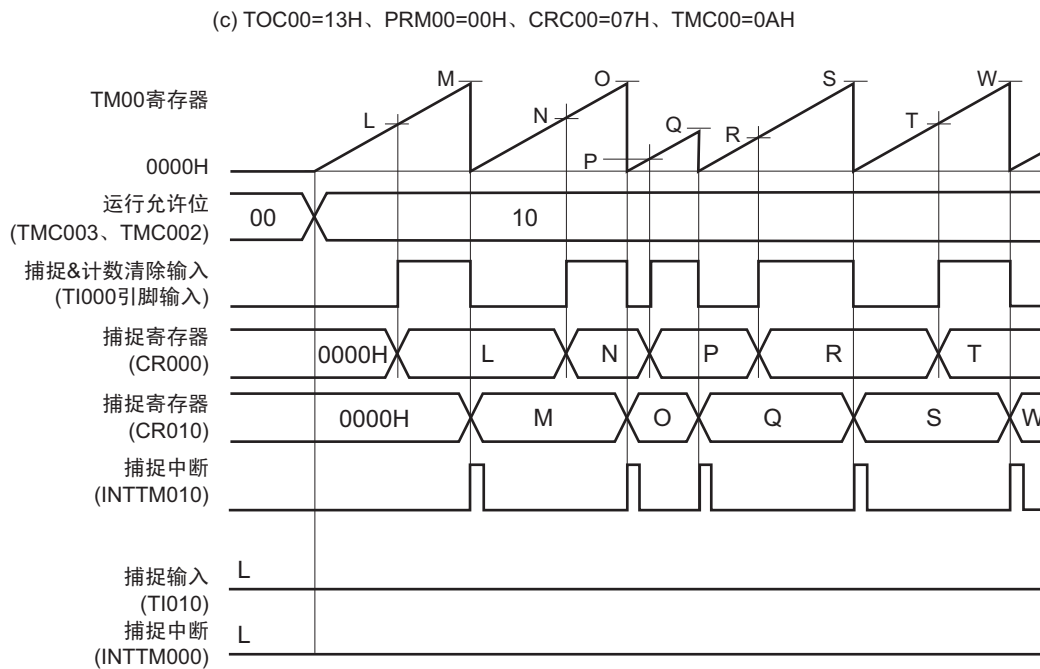
在检测到 TI010 引脚的边沿时，产生中断信号（INTTM000）。如果不使用 INTTM000 信号，必须将其屏蔽。

图 6-30 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 6-30 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。

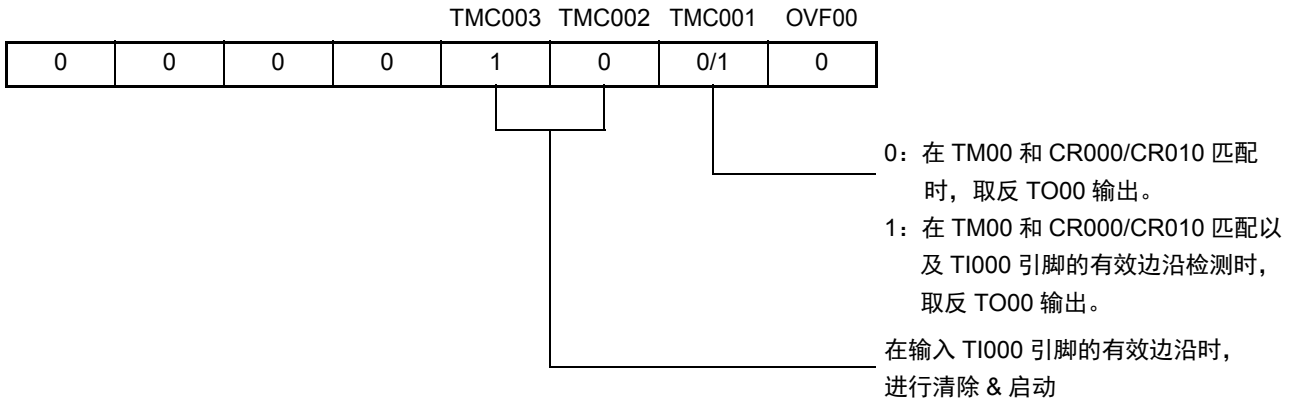
- 高电平宽度=[CR010值] - [CR000值] × [计数时钟周期]
- 低电平宽度=[CR000值] × [计数时钟周期]

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。

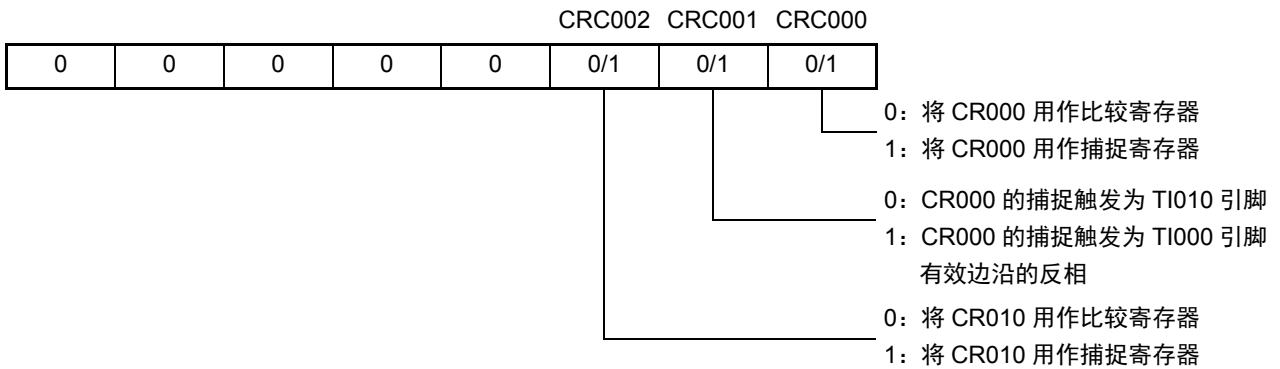
但是，如果将预分频器模式寄存器 00（PRM00）的 bit6 和 bit5（ES110 和 ES100）指定的有效边沿输入到 TI010 引脚，就不进行捕捉运行，但是产生 INTTM000 信号。测量 TI000 引脚的脉宽时，如果不使用 INTTM000 信号，必须将其屏蔽。

图 6-31 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)

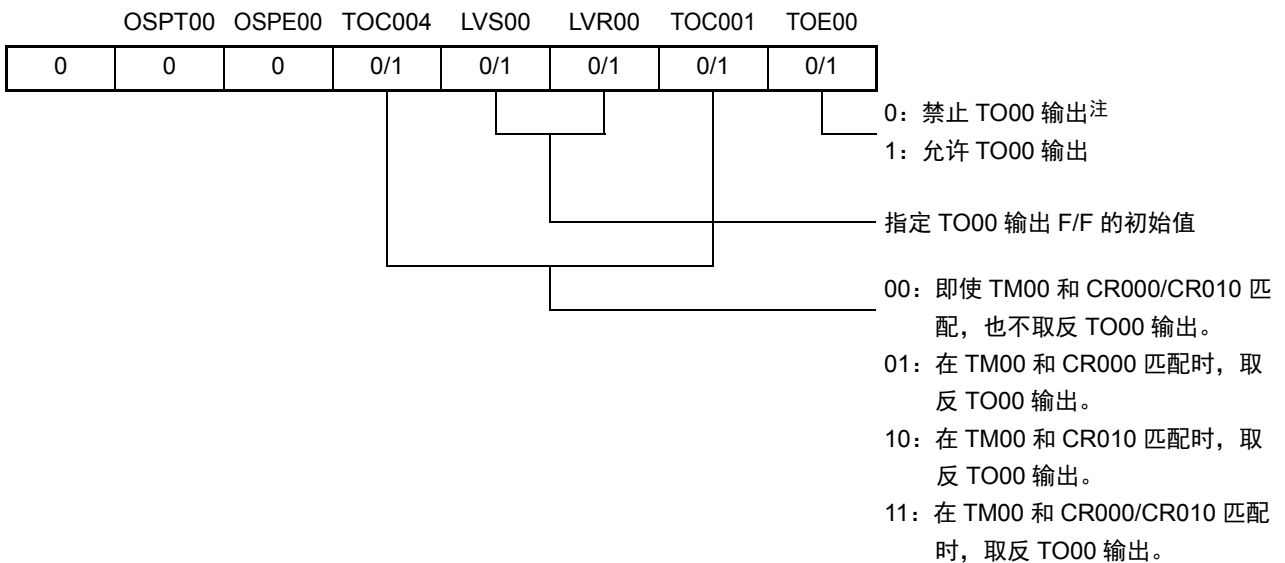
(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕捉 / 比较控制寄存器 00 (CRC00)



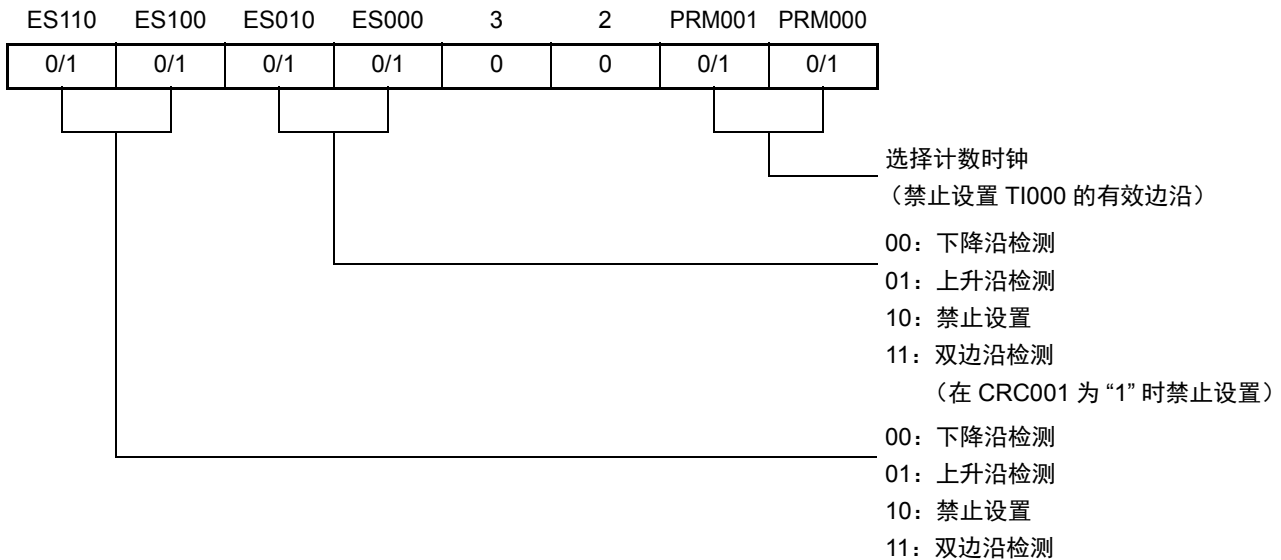
(c) 16 位定时器输出控制寄存器 00 (TOC00)



注 在 20 引脚产品中使用 TI010 引脚的有效边沿检测时, 不能使用定时器输出 (TO00)。

图 6-31 输入 TI000 引脚有效边沿时的清除 &amp; 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。

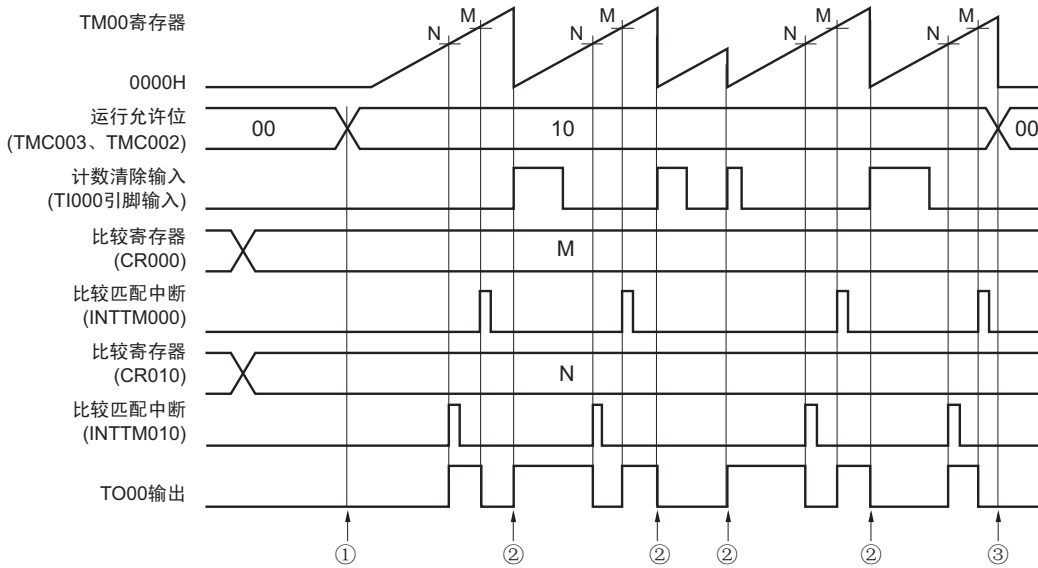
注 在 20 引脚产品中使用 TI010 引脚的有效边沿检测时，不能使用定时器输出 (TO00)。

## (g) 16 位捕捉 / 比较寄存器 010 (CR010)

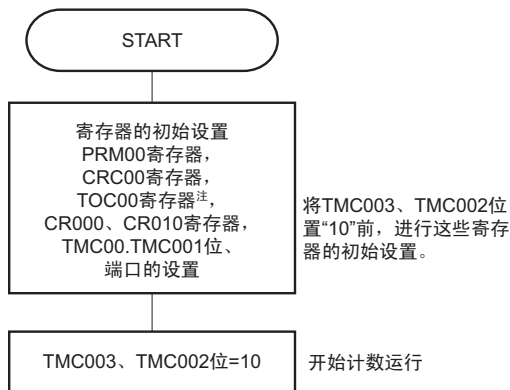
用作比较寄存器时，在与 TM00 的值匹配时，产生中断信号 (INTTM010)。不清除 TM00 的计数值。

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。

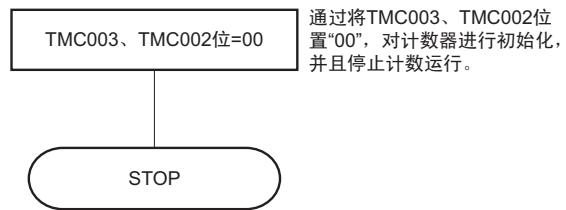
图 6-32 输入 TI000 引脚有效边沿时的清除 & 启动模式运行期间的软件处理示例



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 (6) 端口模式寄存器 3、12 (PM3、PM12)”。

2. 有关 INTTM000 信号的中断允许, 请参照“第 14 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 6-33 自由运行定时器模式的框图  
(CR000: 比较寄存器、CR010: 比较寄存器)

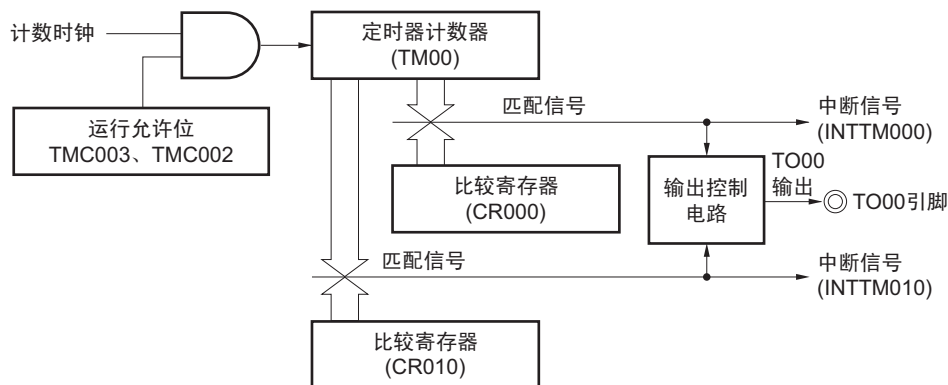
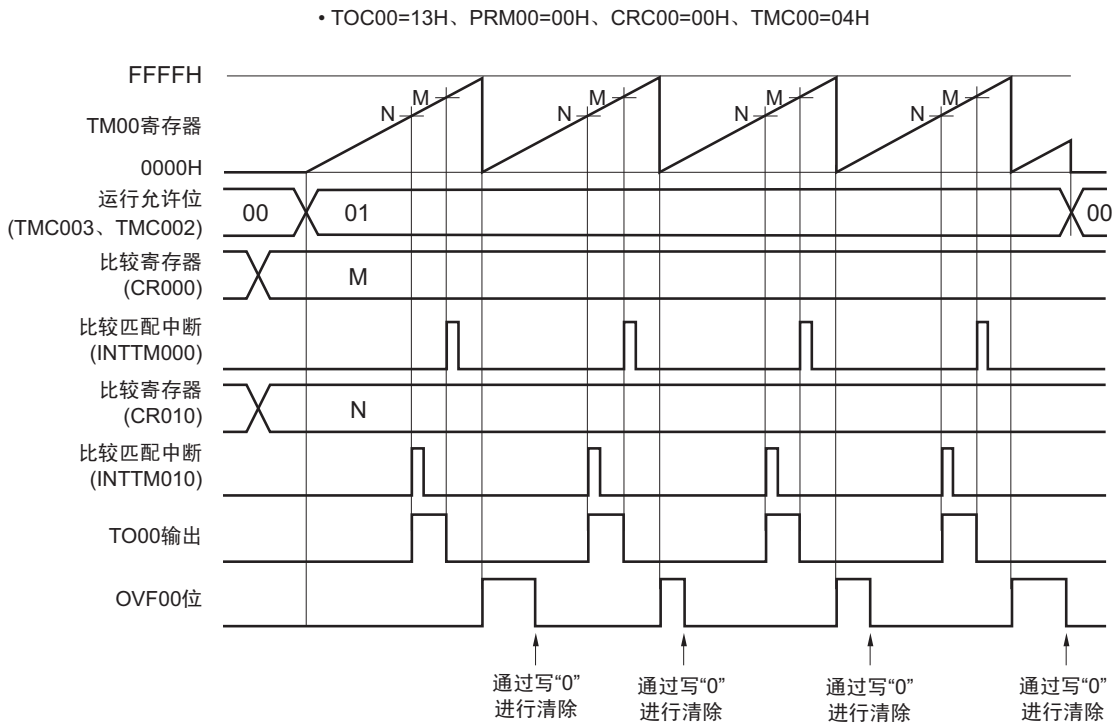


图 6-34 自由运行定时器模式的时序示例  
(CR000: 比较寄存器、CR010: 比较寄存器)



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 6-35 自由运行定时器模式的框图  
(CR000: 比较寄存器、CR010: 捕捉寄存器)

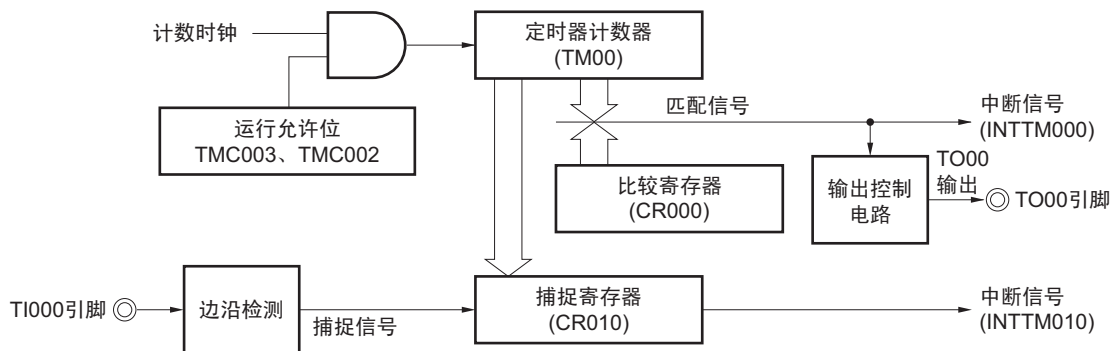
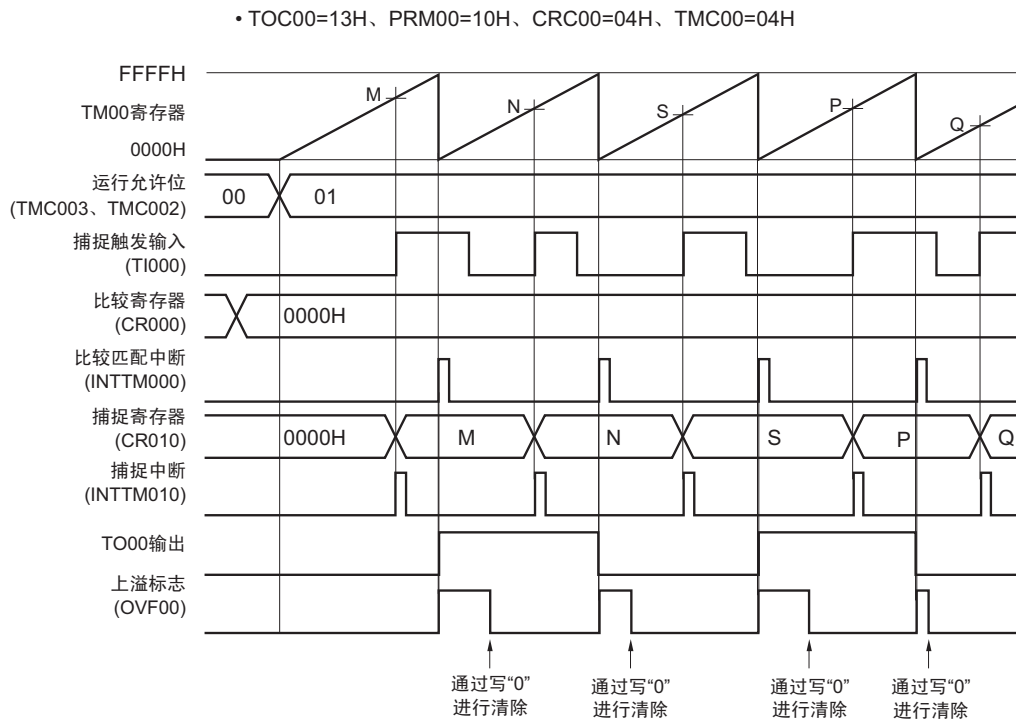




图 6-36 自由运行定时器模式的时序示例  
(CR000: 比较寄存器、CR010: 捕捉寄存器)

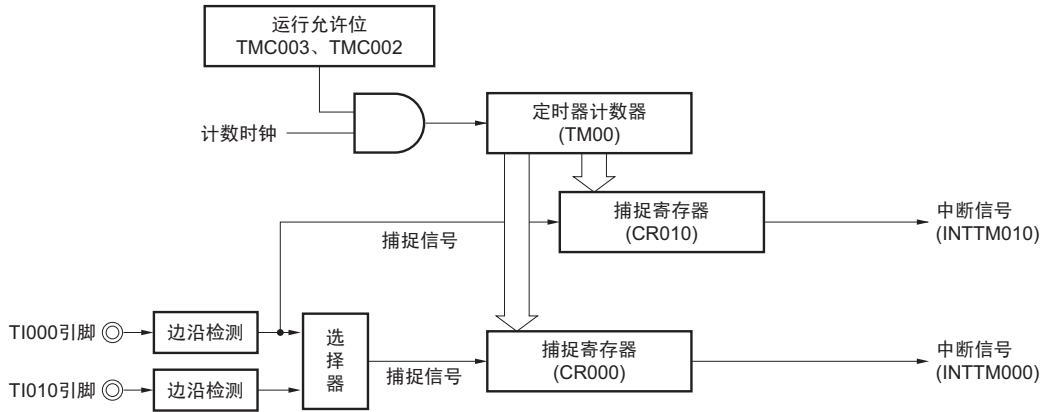


这是在自由运行定时器模式中分别用作比较寄存器和捕捉寄存器的应用示例。

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。

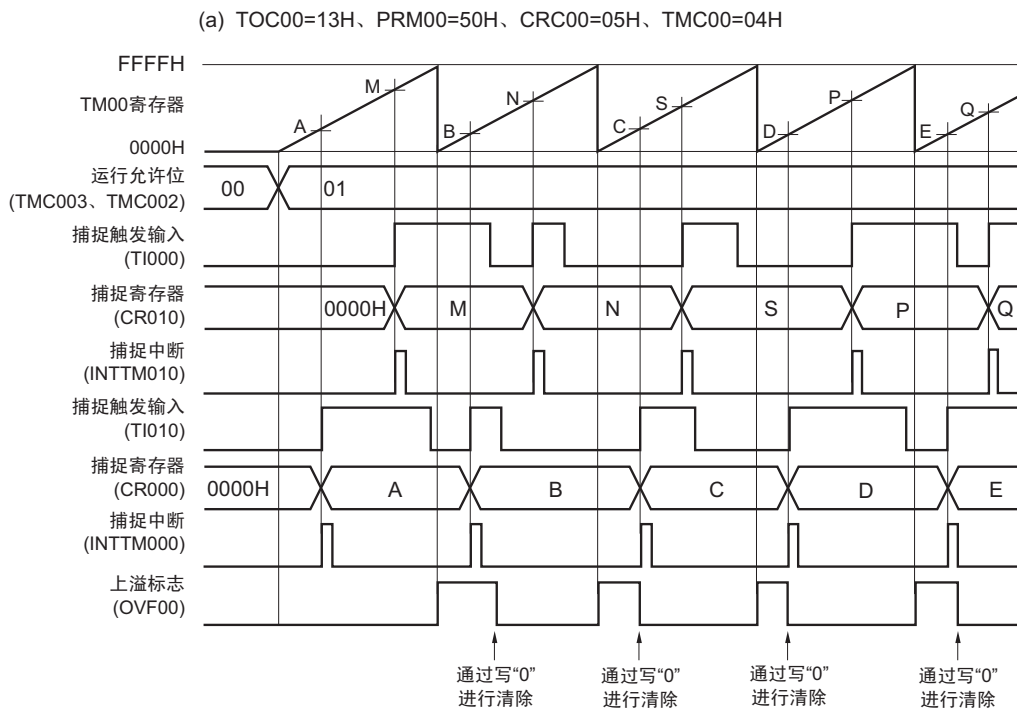
(3) 自由运行定时器模式的运行  
(CR000: 捕捉寄存器、CR010: 捕捉寄存器)

图 6-37 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。

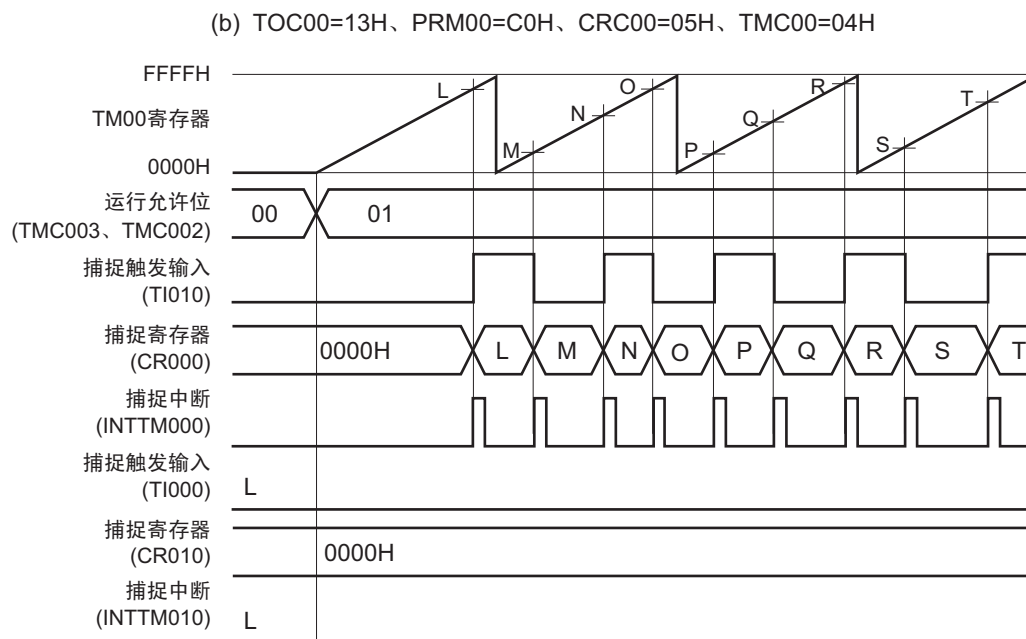
图 6-38 自由运行定时器模式的时序示例  
(CR000: 捕捉寄存器、CR010: 捕捉寄存器) (1/2)



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。

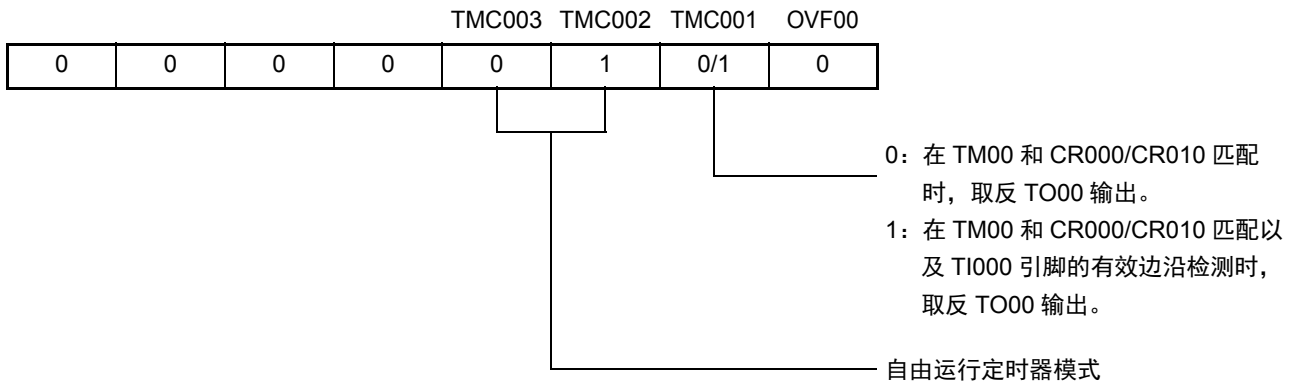
图 6-38 自由运行定时器模式的时序示例  
(CR000: 捕捉寄存器、CR010: 捕捉寄存器) (2/2)



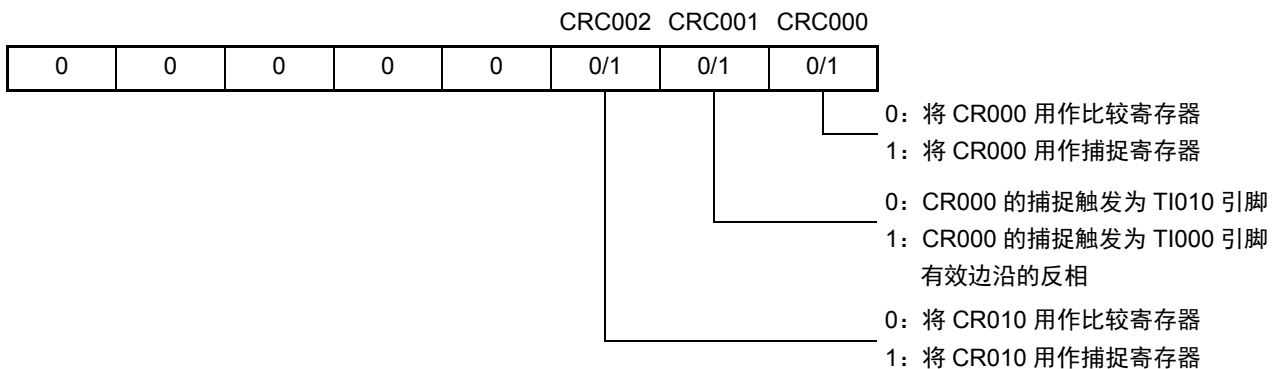
这是在自由运行定时器模式中设置为 TI010 引脚的双边沿检测，并且捕捉计数值到 CR000 的应用示例。CR000 和 CR010 都用作捕捉寄存器并且只能检测出 TI010 引脚的有效边沿时，不能捕捉计数值到 CR010。

图 6-39 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)

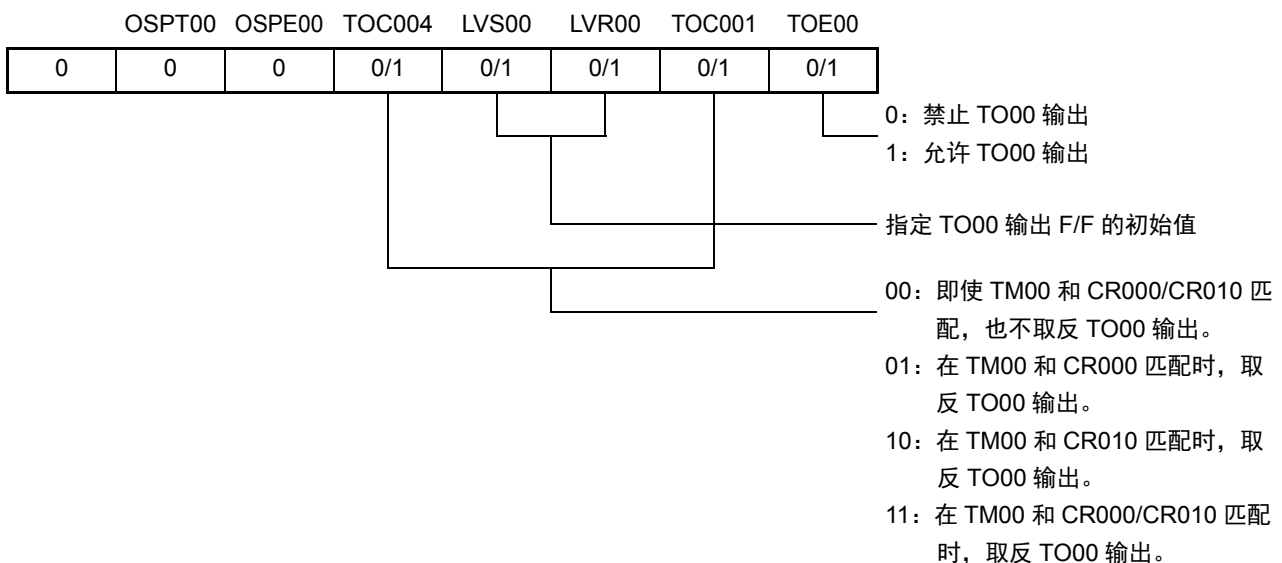
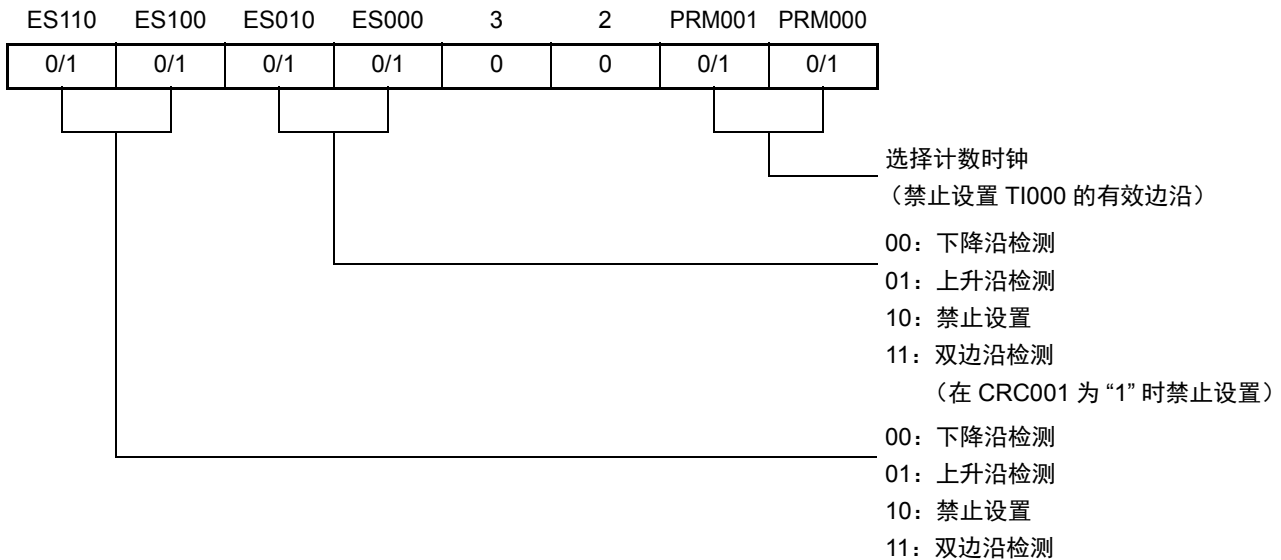


图 6-39 自由运行定时器模式运行期间的寄存器设置内容示例 (2/2)

## (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。

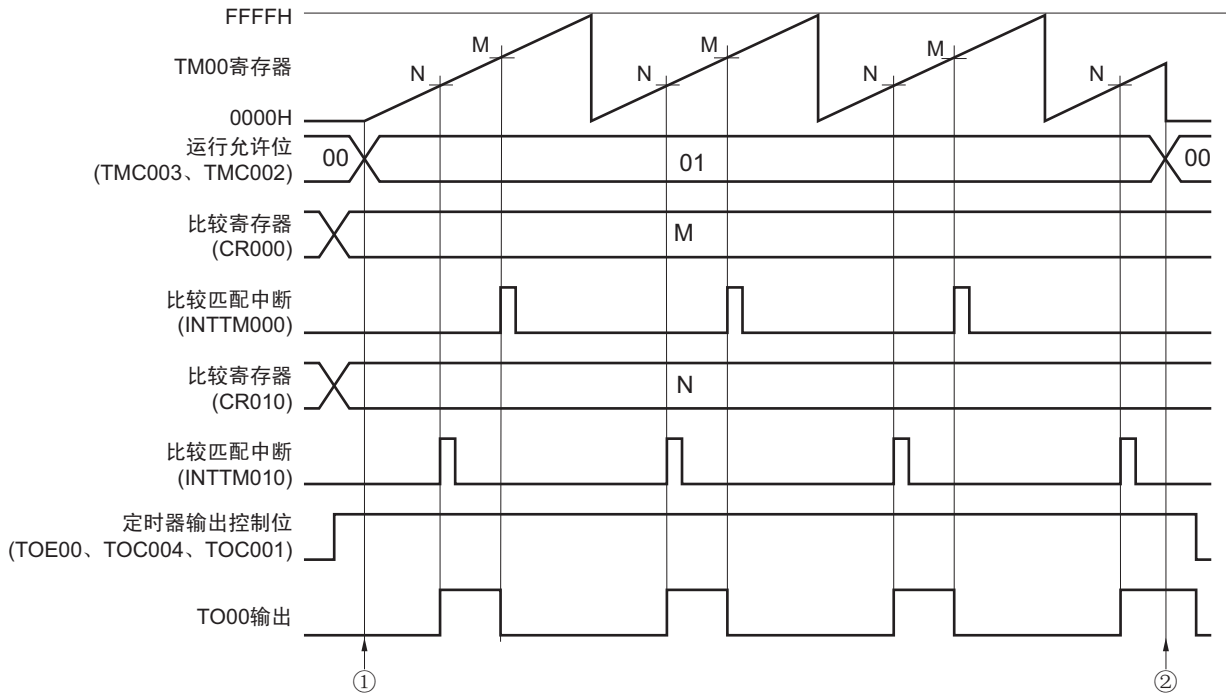
用作捕捉寄存器时，必须设置 TI000 引脚输入或者 TI010 引脚输入作为捕捉触发。在检测出捕捉触发的有效边沿时，将 TM00 的计数值保存到 CR000。

## (g) 16 位捕捉 / 比较寄存器 010 (CR010)

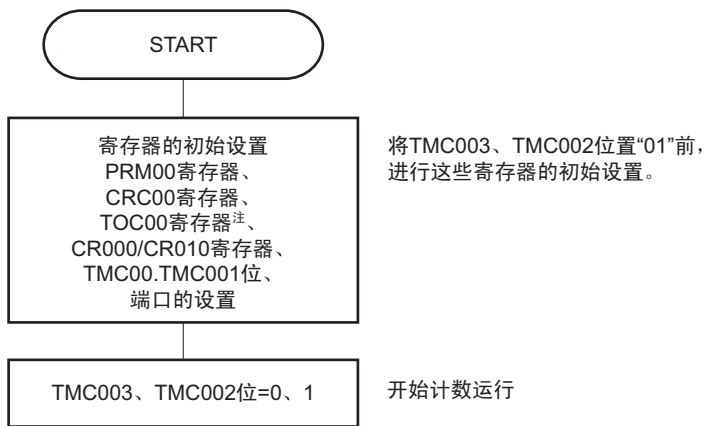
用作比较寄存器时，在与 TM00 的值匹配时，产生中断信号 (INTTM010)。不清除 TM00 的计数值。

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。

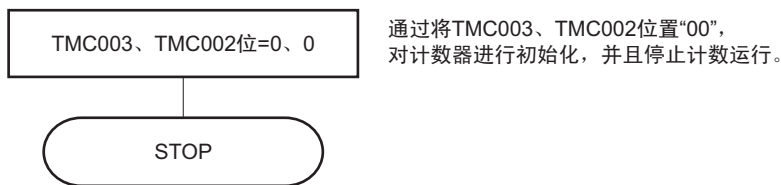
图 6-40 自由运行定时器模式运行期间的软件处理示例



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 (6) 端口模式寄存器 3、12 (PM3、PM12)”。

**2.** 有关 INTTM000 信号的中断允许, 请参照“第 14 章 中断功能”。

图 6-41 PPG 输出运行的框图

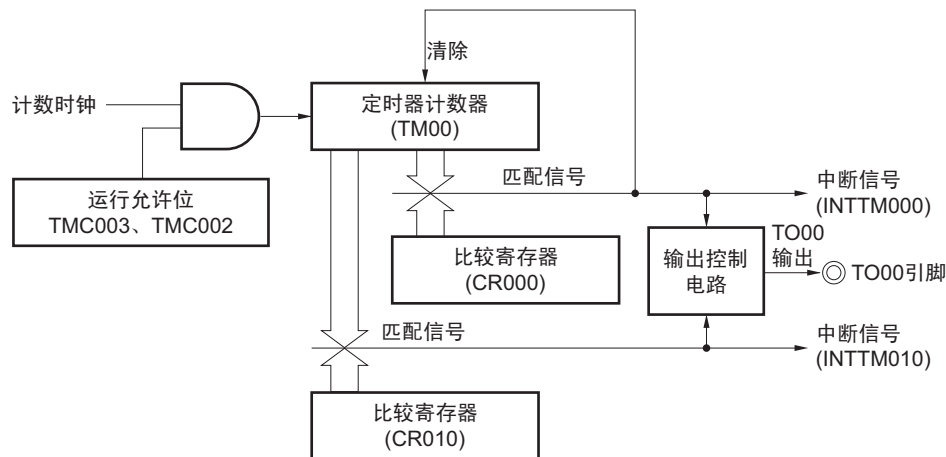
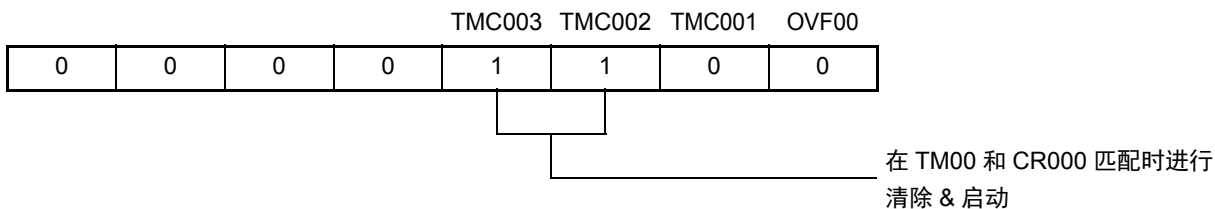
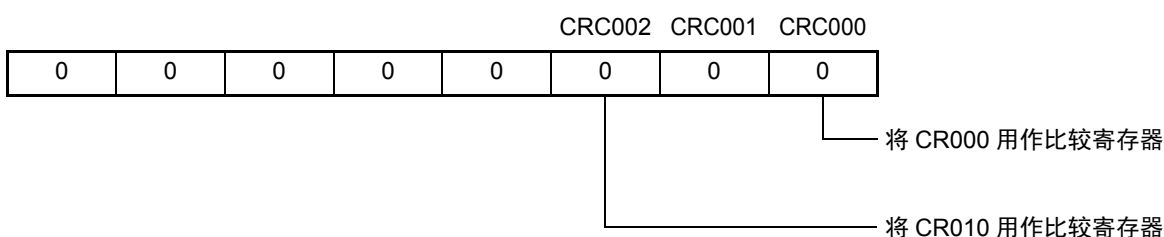


图 6-42 PPG 输出运行时的寄存器设置内容示例 (1/2)

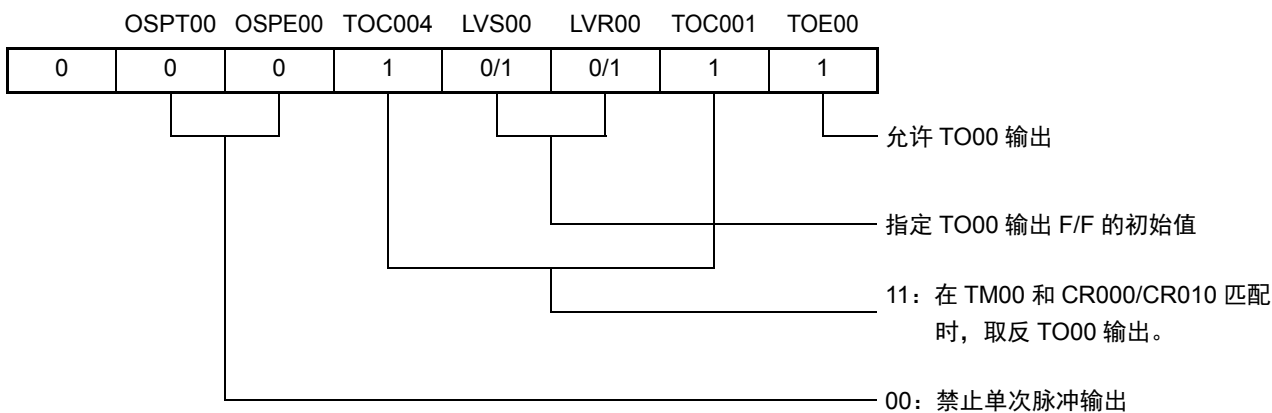
(a) 16 位定时器模式控制寄存器 00 (TMC00)



(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)

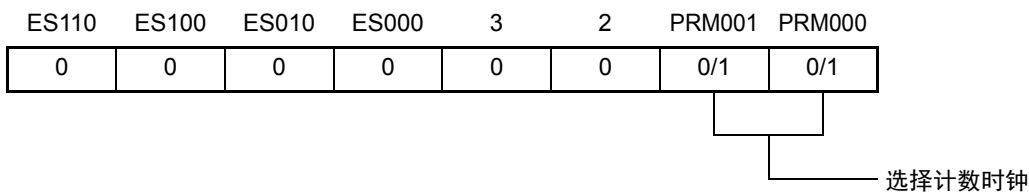




图 6-42 PPG 输出运行时的寄存器设置内容示例 (2/2)

(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)

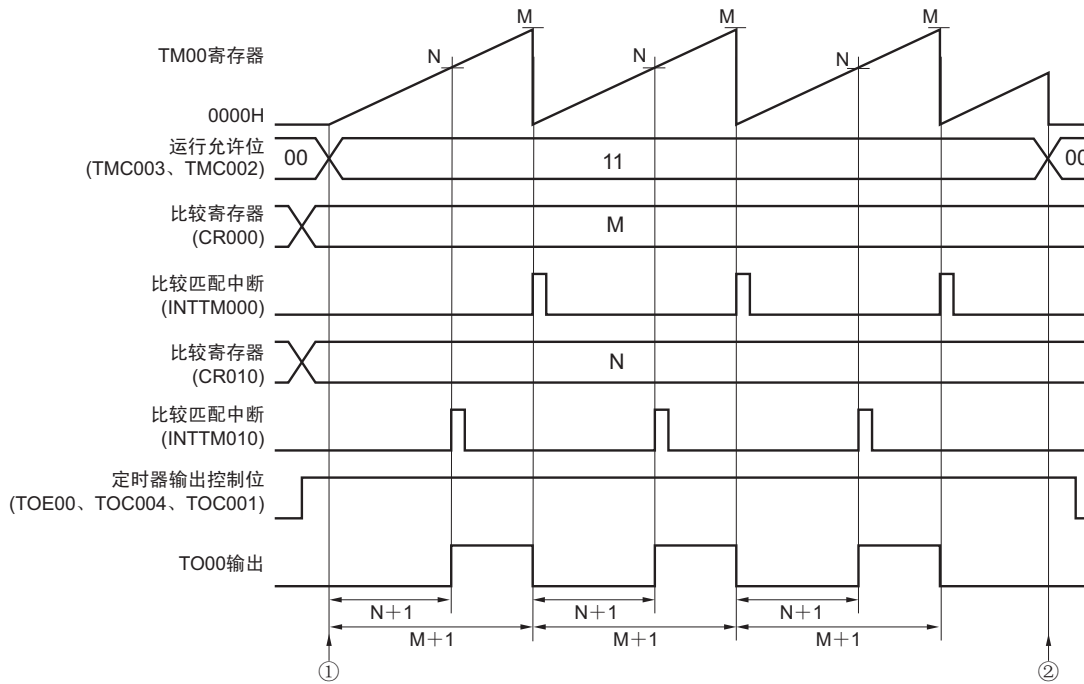
在与 TM00 匹配时，产生中断信号 (INTTM000)。不清除 TM00 的计数值。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

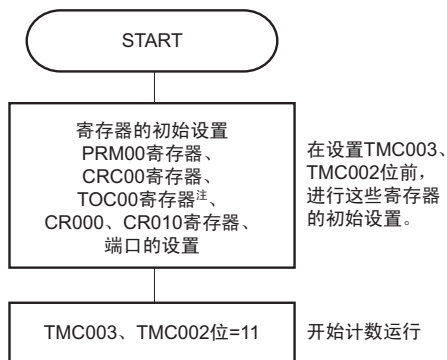
在与 TM00 匹配时，产生中断信号 (INTTM010)。不清除 TM00 的计数值。

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

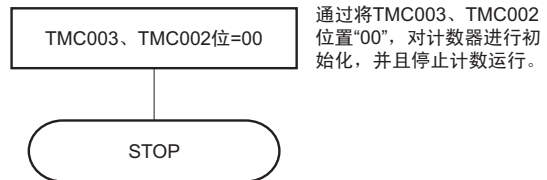
图 6-43 PPG 输出运行时的软件处理示例



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 (6) 端口模式寄存器 3、12 (PM3、PM12)”。
2. 有关 INTTM000 信号的中断允许, 请参照“第 14 章 中断功能”。

图 6-44 单次脉冲输出运行的框图

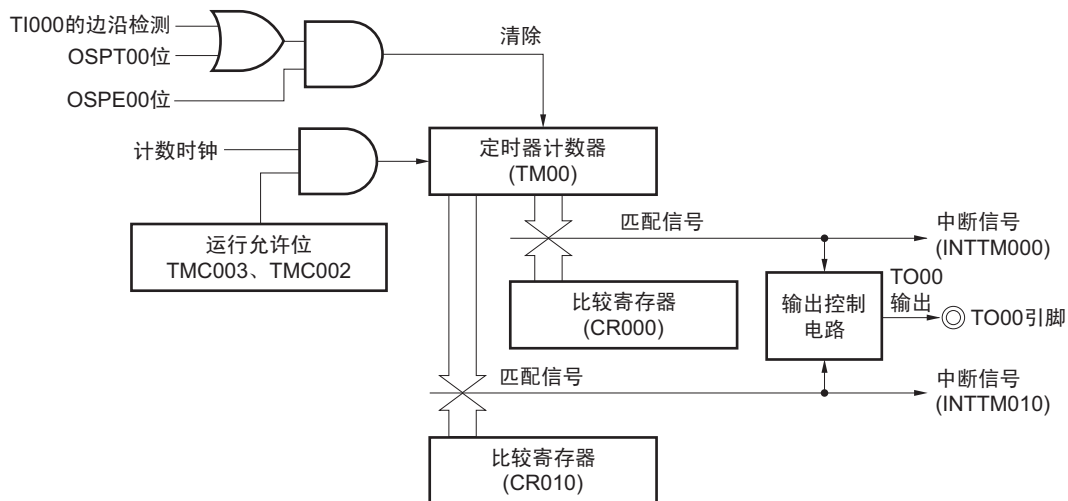
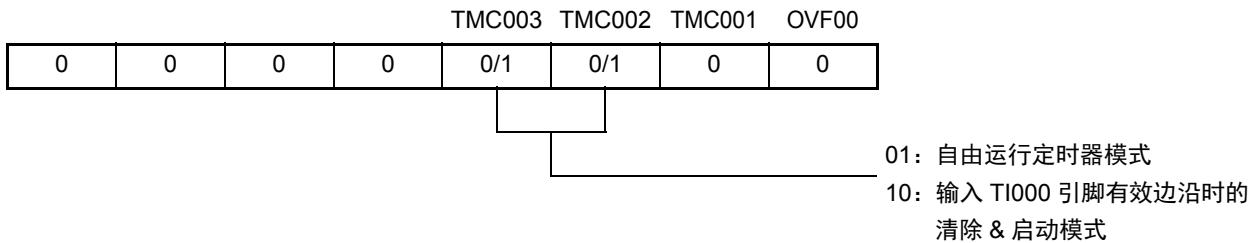
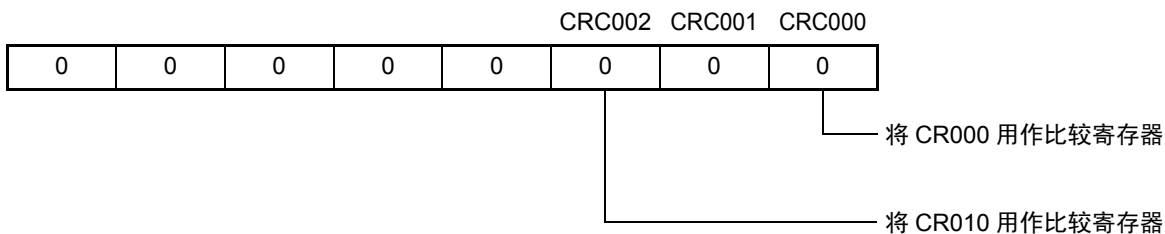


图 6-45 单次脉冲输出运行时的寄存器设置内容示例 (1/2)

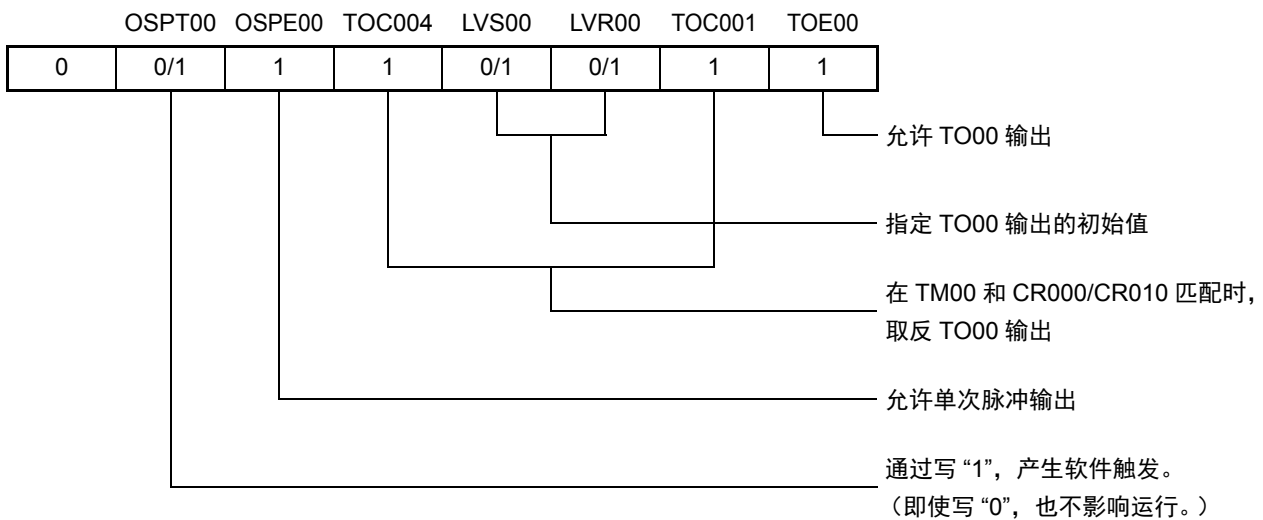
## (a) 16 位定时器模式控制寄存器 00 (TMC00)



## (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)

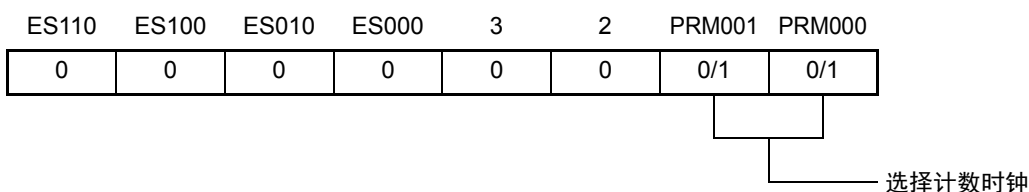


图 6-45 单次脉冲输出运行时的寄存器设置示例 (2/2)

## (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)

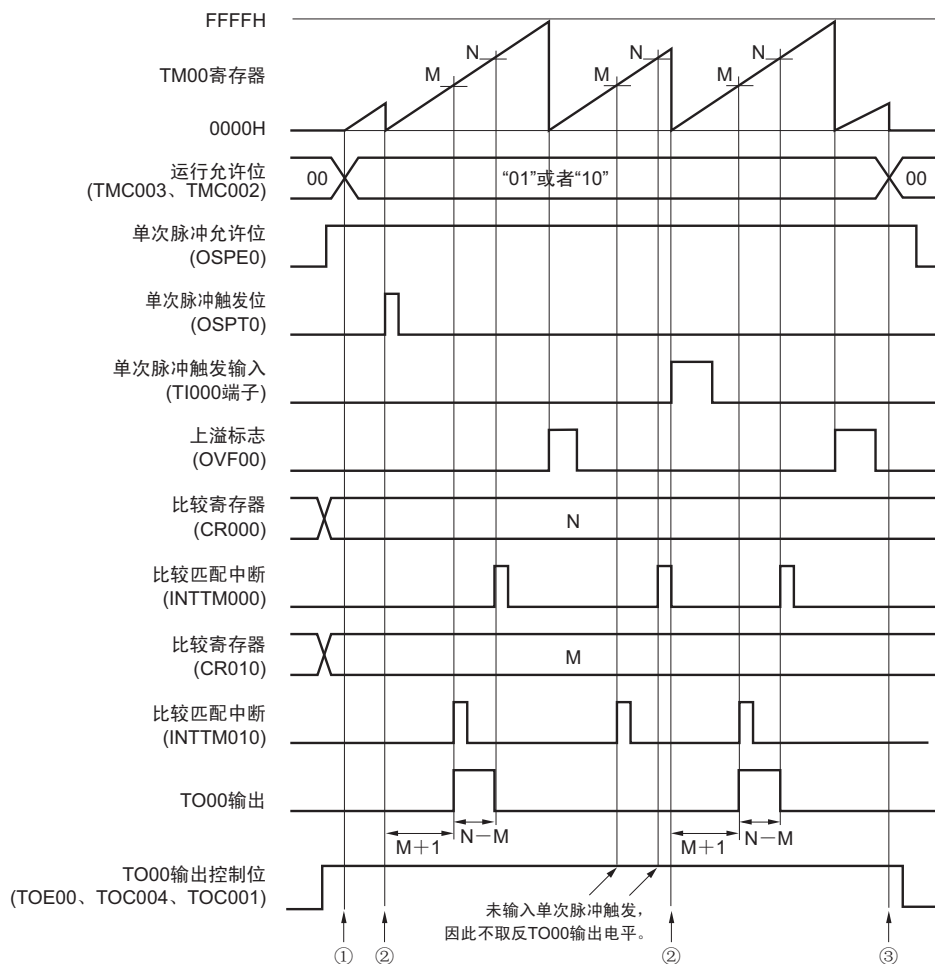
用作单次脉冲输出的比较寄存器。一旦 TM00 的值与 CR000 的值匹配，就产生中断信号 (INTTM000)，并且取反 TO00 输出电平。

## (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。

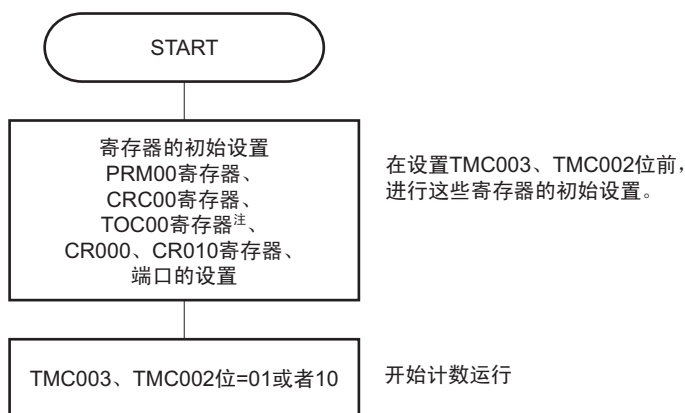
图 6-46 单次脉冲输出运行时的软件处理示例 (1/2)



- 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 6-46 单次脉冲输出运行时的软件处理示例 (2/2)

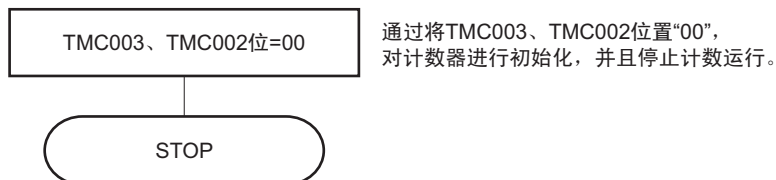
## ① 开始计数运行的流程



## 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 6-47 脉宽测量（自由运行定时器模式）的框图

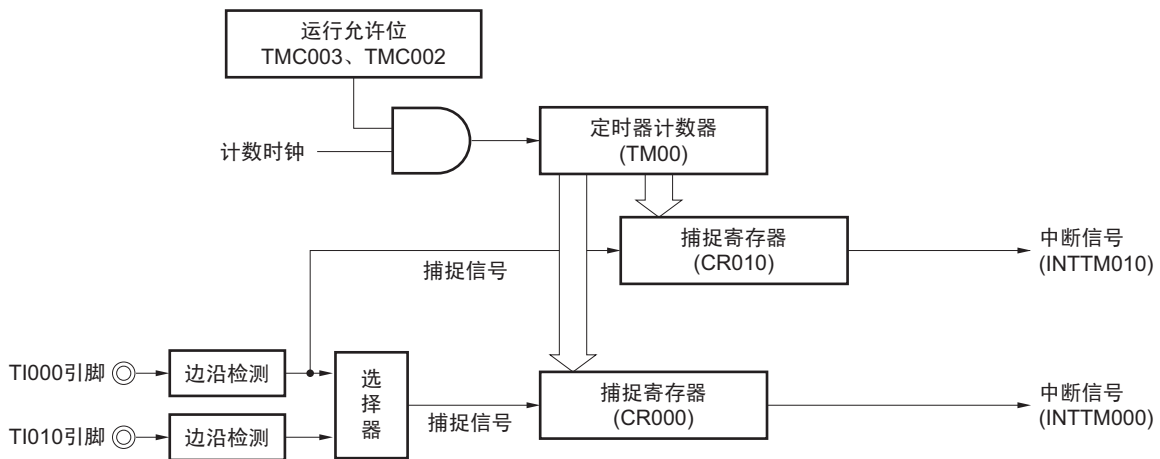
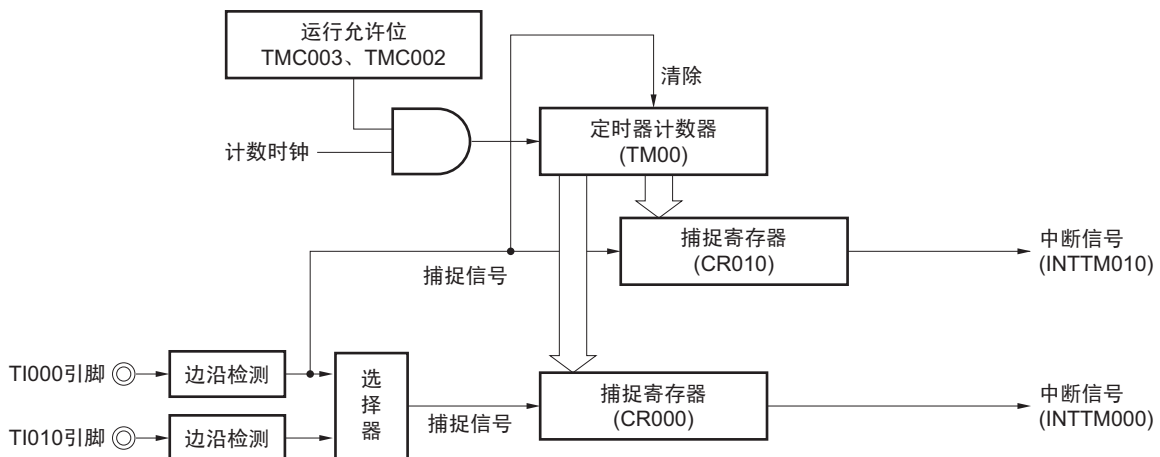


图 6-48 脉宽测量（输入 TI000 引脚有效边沿时的清除 & 启动模式）的框图



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 (6) 端口模式寄存器 3、12（PM3、PM12）”。

2. 有关 INTTM000 信号的中断允许，请参照“第 14 章 中断功能”。

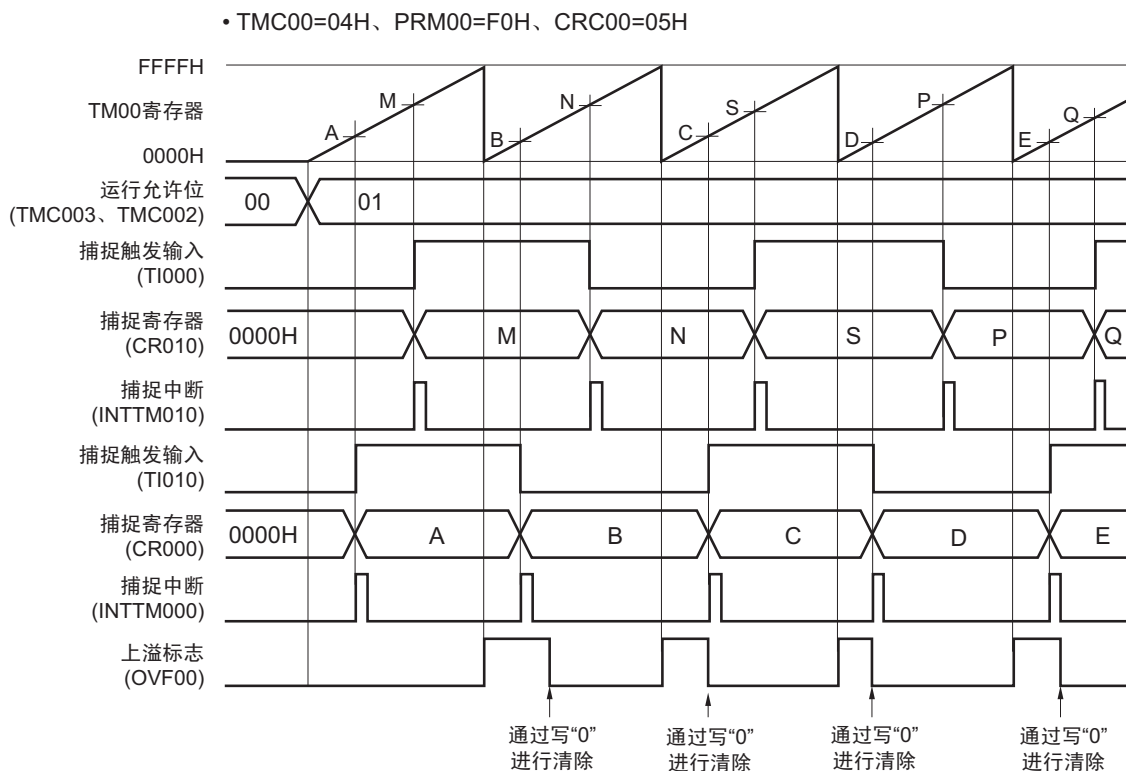
## (1) 通过 TI000 与 TI010 引脚的 2 个输入信号测量脉宽（自由运行定时器模式）

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 6-49 测量脉宽的时序示例（1）



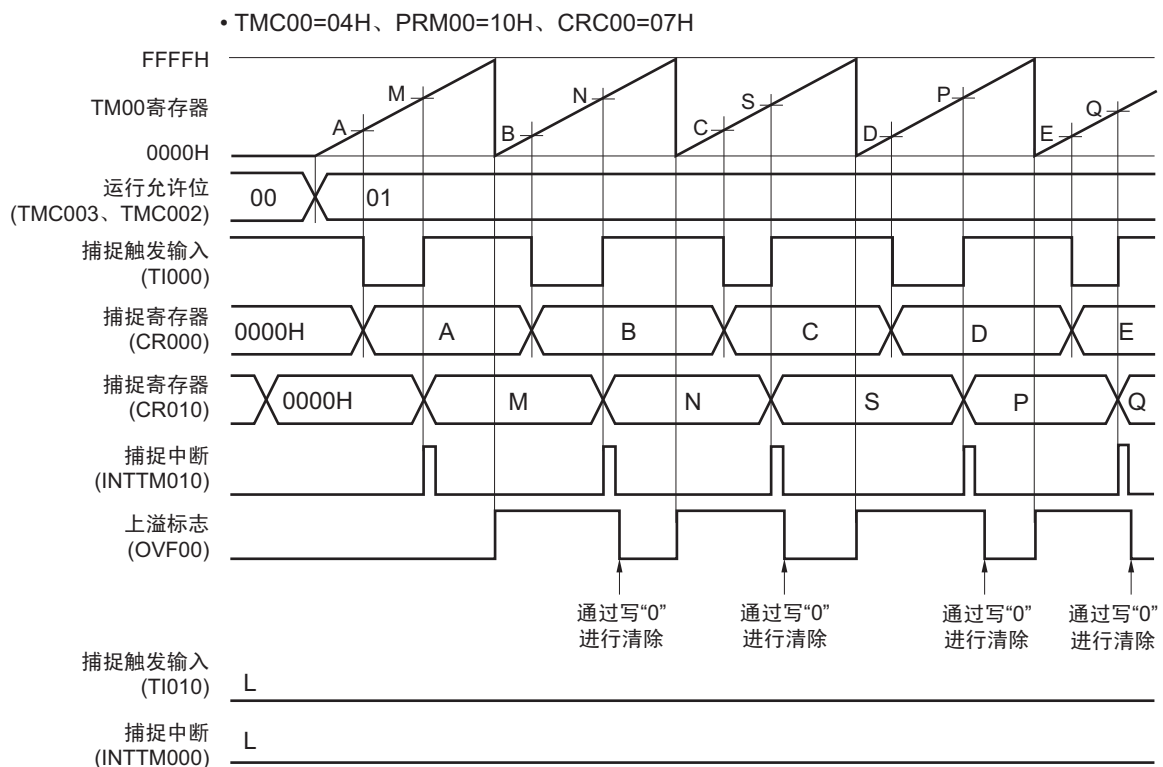
## (2) 通过 TI000 引脚的 1 个输入信号测量脉宽（自由运行定时器模式）

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 6-50 测量脉宽的时序示例（2）



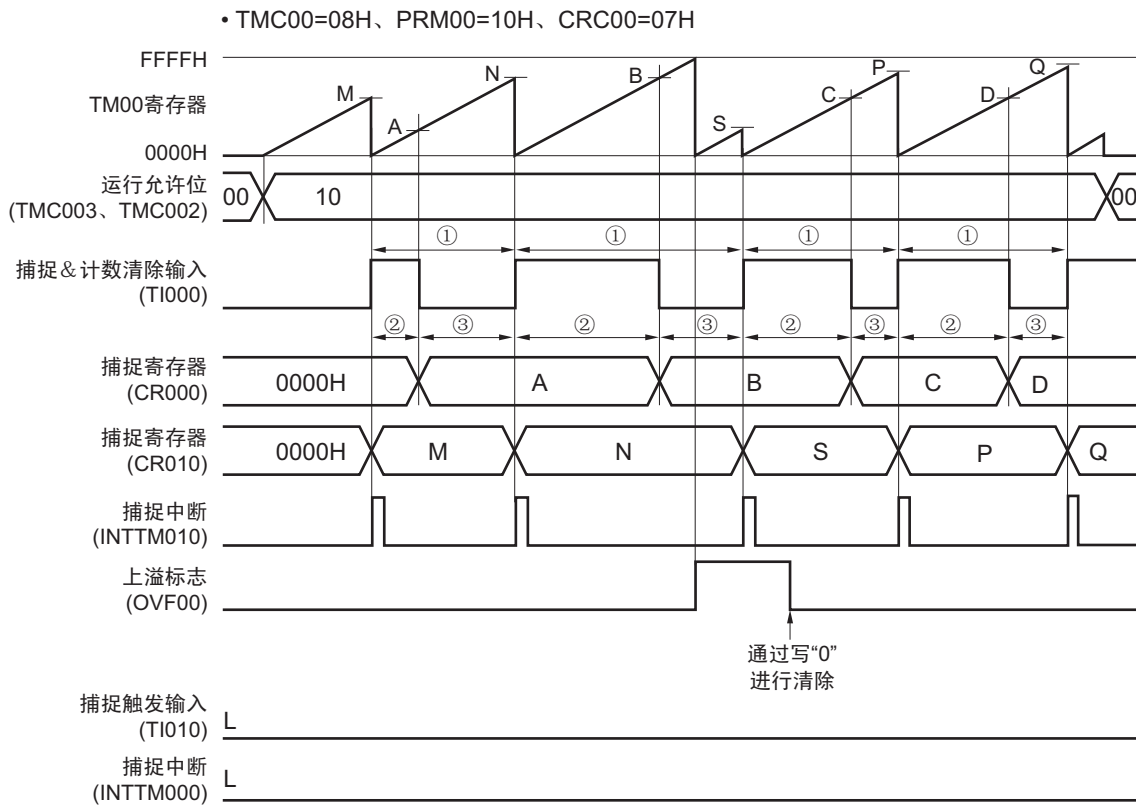


(3) 通过 TI000 引脚的 1 个输入信号测量脉宽 (输入 TI000 引脚有效边沿时的清除 & 启动模式)

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”。

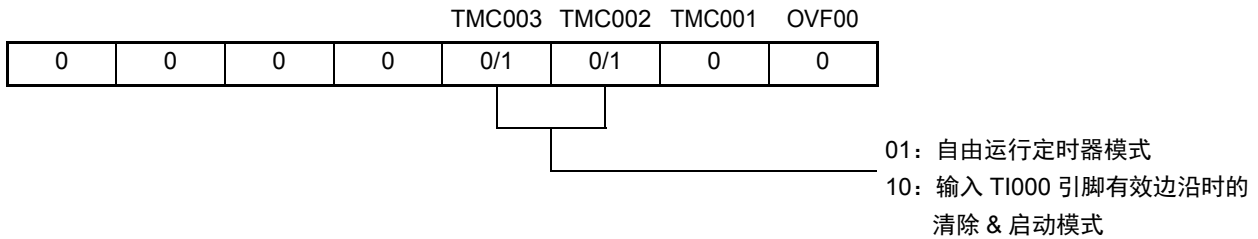
图 6-51 测量脉宽的时序示例 (3)



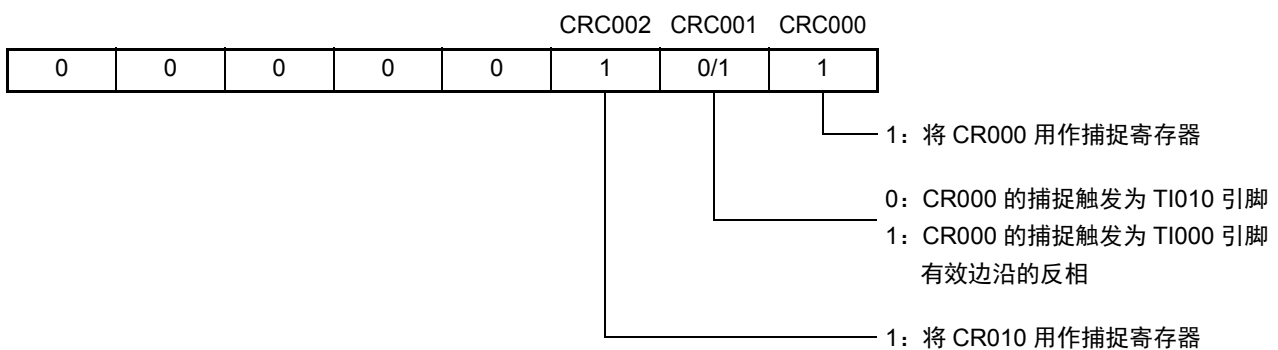
- ① 脉冲周期= (10000H×OVF00位被置“1”的次数+CR010的捕捉值)×计数时钟周期
- ② 脉冲的高电平宽度= (10000H×OVF00位被置“1”的次数+ CR000的捕捉值)×计数时钟周期
- ③ 脉冲的低电平宽度= (脉冲的周期-脉冲的高电平宽度)

图 6-52 测量脉宽时的寄存器设置内容示例 (1/2)

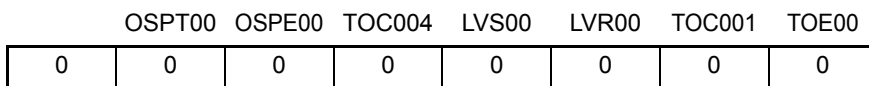
## (a) 16 位定时器模式控制寄存器 00 (TMC00)



## (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)

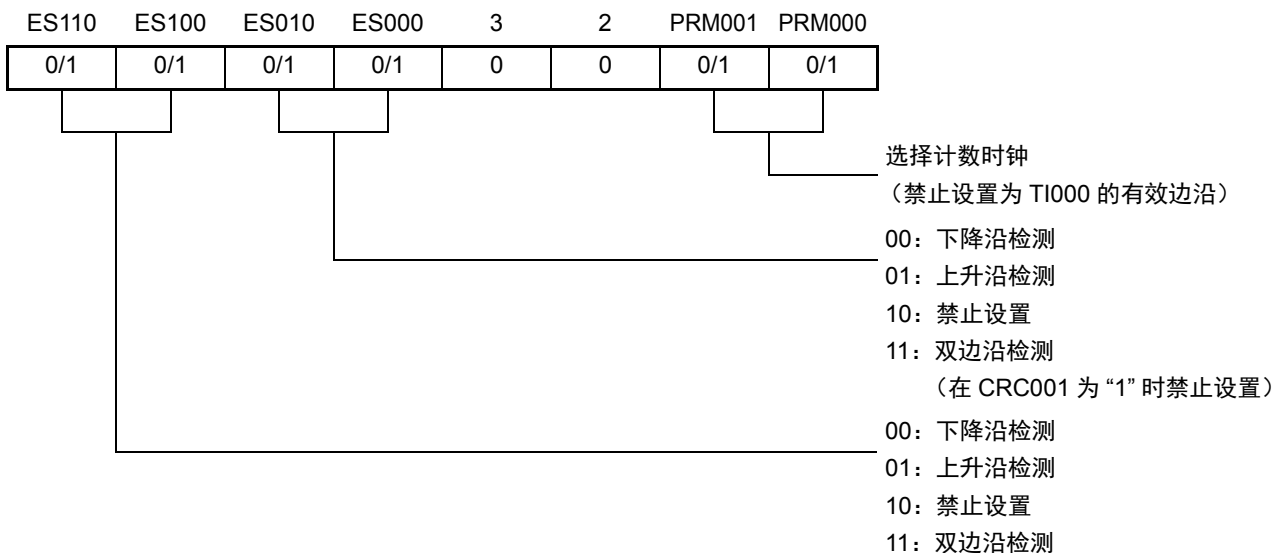


图 6-52 测量脉宽时的寄存器设置内容示例 (2/2)

## (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 6-53 测量脉宽时的软件处理示例 (1/2)

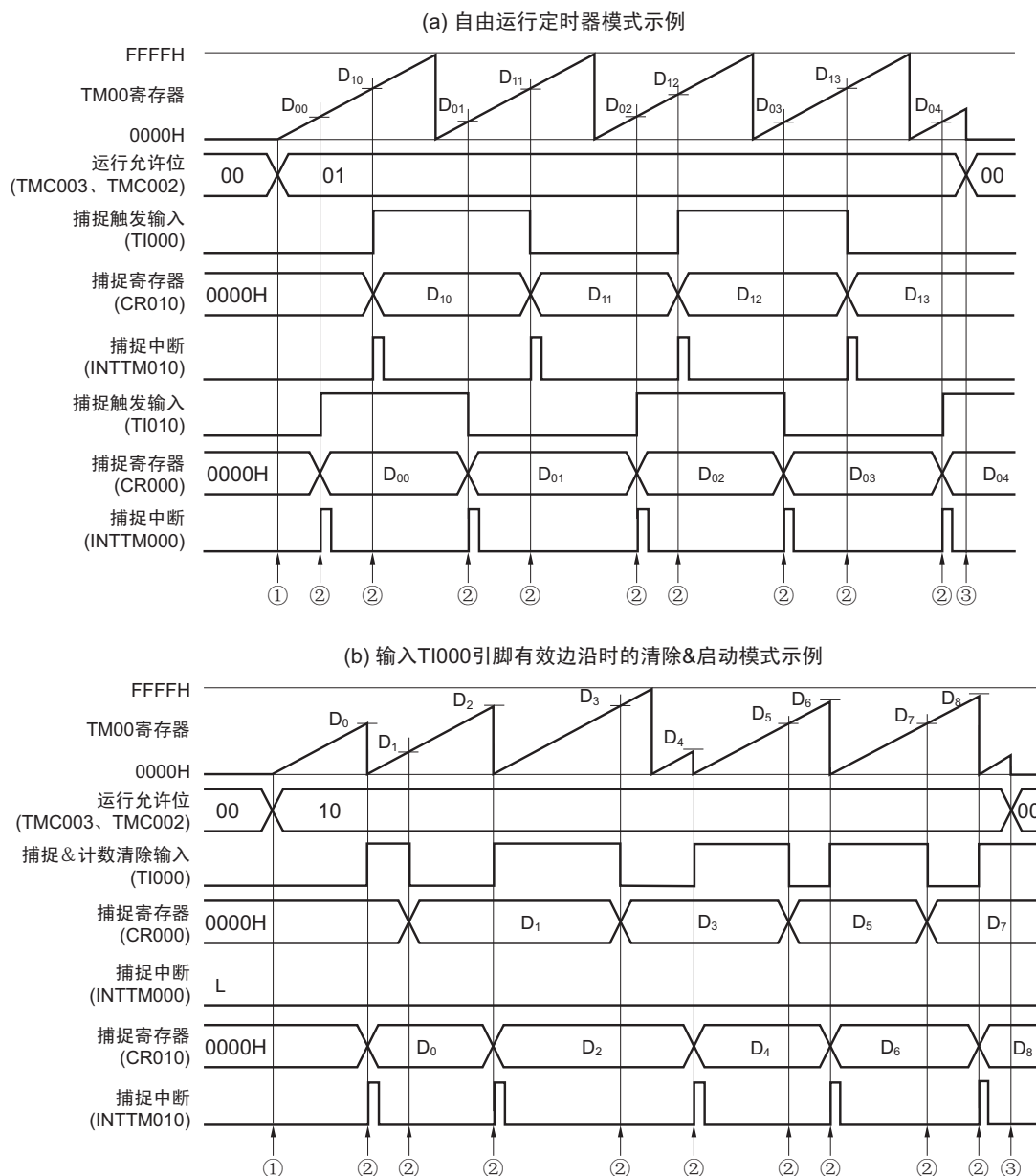
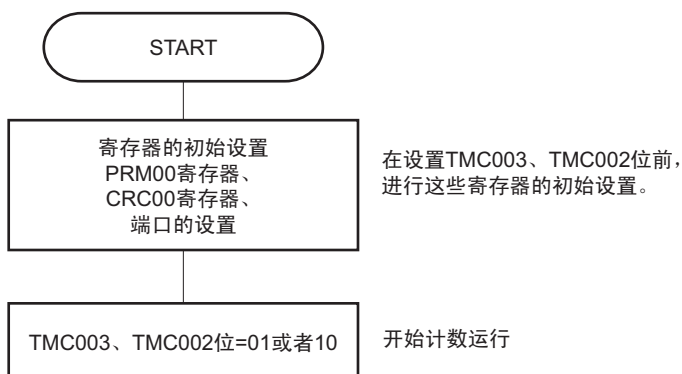
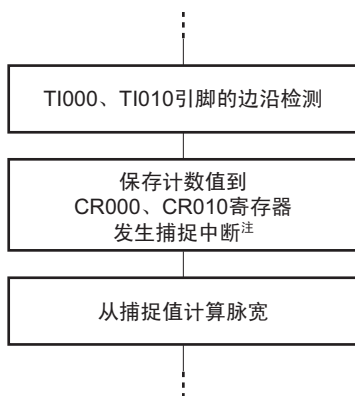


图 6-53 测量脉宽时的软件处理示例 (2/2)

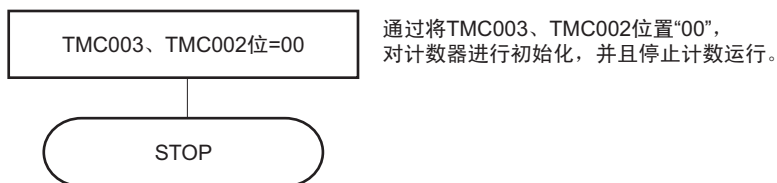
## ① 开始计数运行的流程



## 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（INTTM000）。

## 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，禁止改写作比较寄存器的 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 14 章 中断功能”。

### 6.5.2 LVS00 和 LVR00 的设置

#### (1) LVS00 和 LVR00 的使用用途

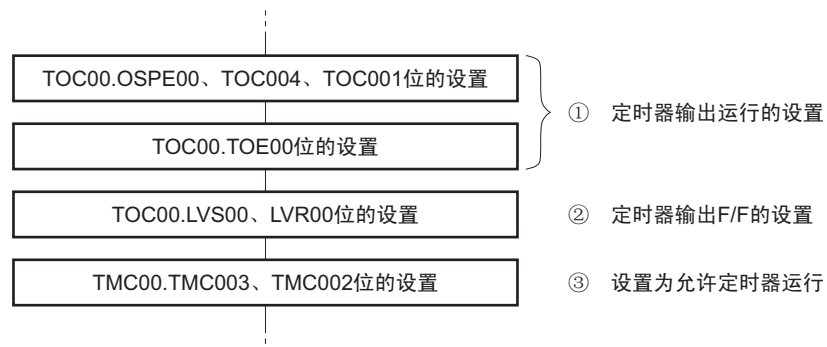
在设置 TO00 输出的初始值，或者在不允许定时器运行（TMC003 与 TMC002=00）的状态下取反定时器输出时，使用 LVS00 和 LVR00。不需要软件控制时，必须将 LVS00 和 LVR00 置“00”（初始值为低电平输出）。

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。

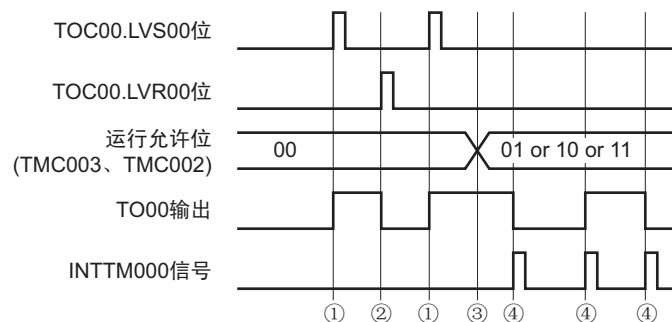
图 6-54 LVS00 和 LVR00 位的设置流程示例



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 6-55 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。

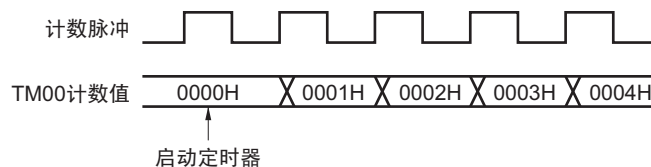
表 6-3 16 位定时器 / 事件计数器 00 的各通道限制事项

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	$0000H \leq CR010 < CR000 \leq FFFFH$
单次脉冲的输出运行	禁止给 CR000 和 CR010 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 6-56 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）

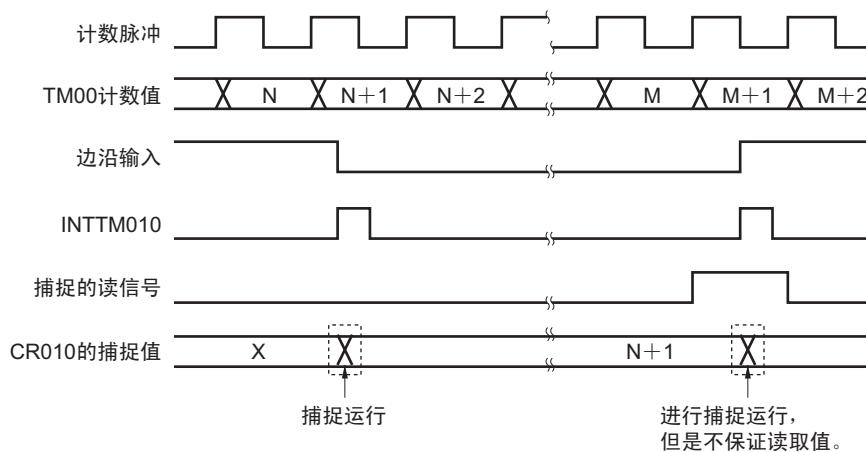
必须给 CR000 和 CR010 设置“0000H”以外的值（用作外部事件计数器时，不能进行 1 个脉冲的计数运行）。

## (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 6-57 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 ES001 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。

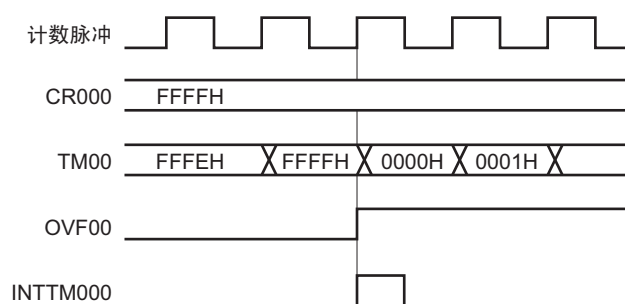
↓

将 CR000 置“FFFFH”。

↓

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

图 6-58 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

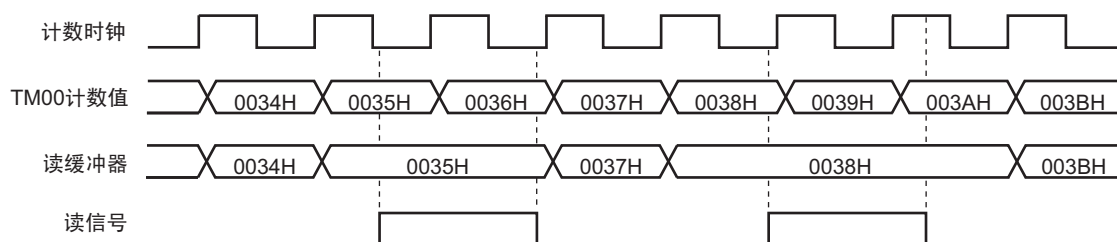
与 CPU 的运行模式无关，只要定时器停止，就不接受 TI000/TI010 引脚的输入信号。

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 6-59 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51

### 7.1 8位定时器/事件计数器51的功能

R7F0C30x、R7F0C31x的所有产品都装有8位定时器/事件计数器51。

8位定时器/事件计数器51具有以下功能。

1. 间隔定时器
2. 外部事件计数器
3. 方波输出
4. PWM输出

备注 方波输出和PWM输出仅可用作UART0的基本时钟。

### 7.2 8位定时器/事件计数器51的结构

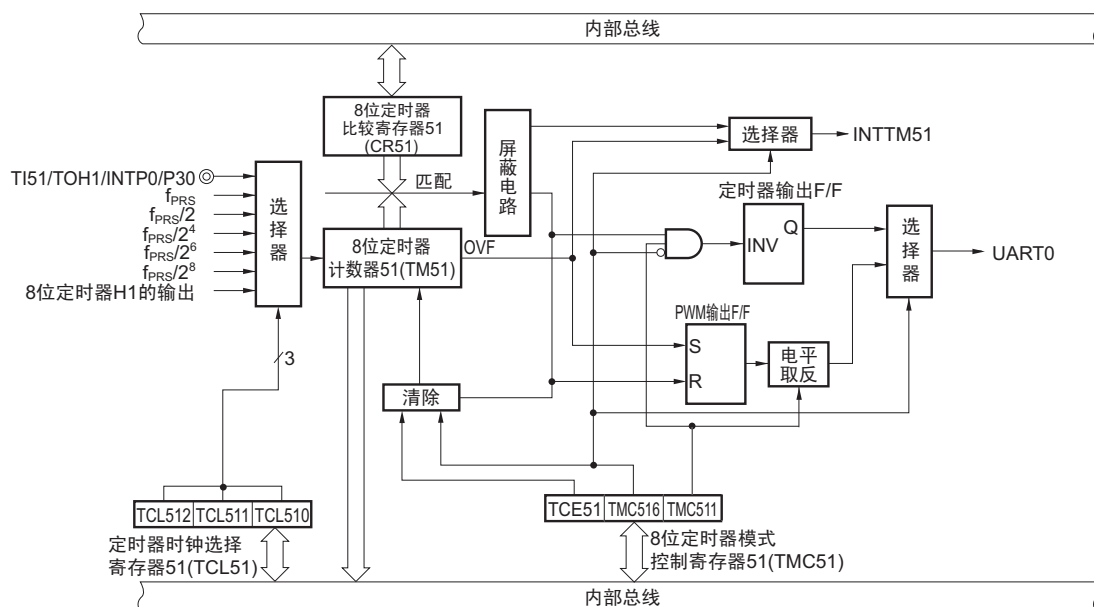
8位定时器/事件计数器51由以下硬件构成。

表 7-1 8位定时器/事件计数器51的结构

项目	结构
定时器寄存器	8位定时器计数器51 (TM51)
定时器输入	TI51
寄存器	8位定时器比较寄存器51 (CR51)
控制寄存器	定时器时钟选择寄存器51 (TCL51) 8位定时器模式控制寄存器51 (TMC51) 端口模式寄存器3 (PM3) 端口寄存器3 (P3)

8位定时器/事件计数器51的框图如图7-1所示。

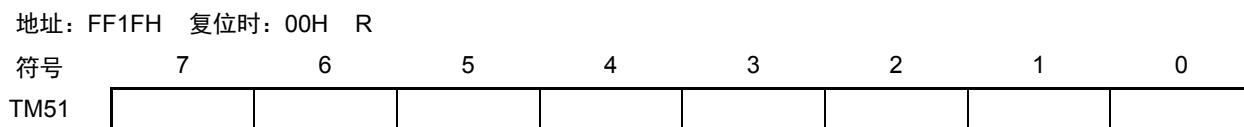
图 7-1 8位定时器/事件计数器51的框图



## (1) 8位定时器计数器 51 (TM51)

TM51 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 7-2 8 位定时器计数器 51 (TM51) 的格式



在以下情况下，计数值为“00H”。

- ① 产生复位信号
- ② 清除 TCE51
- ③ 在 TM51 和 CR51 匹配时的清除 & 启动模式中 TM51 和 CR51 匹配

## (2) 8位定时器比较寄存器 51 (CR51)

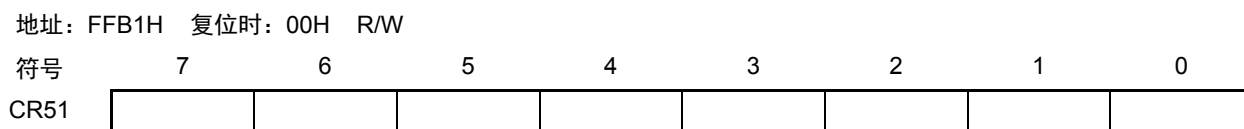
通过 8 位存储器操作指令读写 CR51。

CR51 的设置值总是与 8 位定时器计数器 51 (TM51) 的计数值进行比较，如果两个值匹配，就产生中断请求 (INTTM51)。

在“00H ~ FFH”范围内设置 CR51 的值。

在产生复位信号时，CR51 为“00H”。

图 7-3 8 位定时器比较寄存器 51 (CR51) 的格式



注意 在 TM51 与 CR51 匹配时的清除 & 启动模式 (TMC516=0) 中，不能在运行期间给 CR51 写入不同的值。

### 7.3 8位定时器/事件计数器 51 的控制寄存器

8位定时器/事件计数器 51 的控制寄存器有以下 4 种。

- 定时器时钟选择寄存器 51 (TCL51)
- 8位定时器模式控制寄存器 51 (TMC51)
- 端口模式寄存器 3 (PM3)
- 端口寄存器 3 (P3)

#### (1) 定时器时钟选择寄存器 51 (TCL51)

定时器时钟选择寄存器 51 设置 8 位定时器/事件计数器 51 的计数时钟和 TI51 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 TCL51。  
在产生复位信号时，TCL51 为“00H”。

图 7-4 定时器时钟选择寄存器 51 (TCL51) 的格式

地址: FFB2H 复位时: 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	计数时钟的选择注 1			
			$f_{PRS}=2\text{MHz}$	$f_{PRS}=5\text{MHz}$	$f_{PRS}=10\text{MHz}$	
0	0	0	TI51 引脚的下降沿注 2			
0	0	1	TI51 引脚的上升沿注 2			
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	TMH1 输出			

注 1. 在外围硬件时钟 ( $f_{PRS}$ ) 以高速系统时钟 ( $f_{XH}$ ) 运行 (XSEL=1) 时,  $f_{PRS}$  的工作频率因电源电压而不同。

- $V_{DD}=2.7\sim 5.5\text{V}$ :  $f_{PRS}\leq 10\text{MHz}$
- $V_{DD}=1.8\sim 2.7\text{V}$ :  $f_{PRS}\leq 5\text{MHz}$

2. 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 51 (TMC51)

8位定时器模式控制寄存器 51 控制 8位定时器计数器 51 (TM51) 的计数运行。

通过 1 位或 8 位存储器操作指令设置 TMC51。

在产生复位信号时，TMC51 为“00H”。

图 7-5 8位定时器模式控制寄存器 51 (TMC51) 的格式

地址：FFB3H 复位时：00H R/W

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

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 与 CR51 匹配时的清除 & 启动模式
1	PWM（自由运行）模式

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

注意 1. 按照以下步骤进行设置，禁止同时设置。

- ① 设置 TMC511、TMC516：设置运行模式
- ② 设置 TCE51

2. 在 TCE51 为“1”时，禁止设置 TMC51 的其他位。

备注 在 PWM 模式中，通过将 TCE51 置“0”，PWM 输出变为无效电平。

## (3) 端口模式寄存器 3 (PM3)

端口模式寄存器 3 以 1 位单位设置端口 3 的输入 / 输出。

通过 1 位或 8 位存储器操作指令设置 PM3。

在产生复位信号时，PM3 为“FFH”。

将 P30/TI51/TOH1/INTP0 引脚用作定时器输入时，必须将 PM30 置“1”。此时，P30 的输出锁存器即可是“0”也可能是“1”。

图 7-6 端口模式寄存器 3 (PM3) 的格式

地址：FF23H 复位时：FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34 注	PM33 注	PM32	PM31	PM30

PM3n	选择 P3n 引脚的输入 / 输出模式 (n=0 ~ 4)
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

注 只限 20 引脚产品。

## 7.4 8 位定时器 / 事件计数器 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 51 (CR51) 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 51 (TM51) 的计数值与 CR51 的设置值匹配时, 将 TM51 清“0”, 并且在继续计数的同时, 产生中断请求信号 (INTTM51)。

可通过定时器时钟选择寄存器 51 (TCL51) 的 bit0 ~ bit2 (TCL510 ~ TCL512) 选择 TM51 的计数时钟。

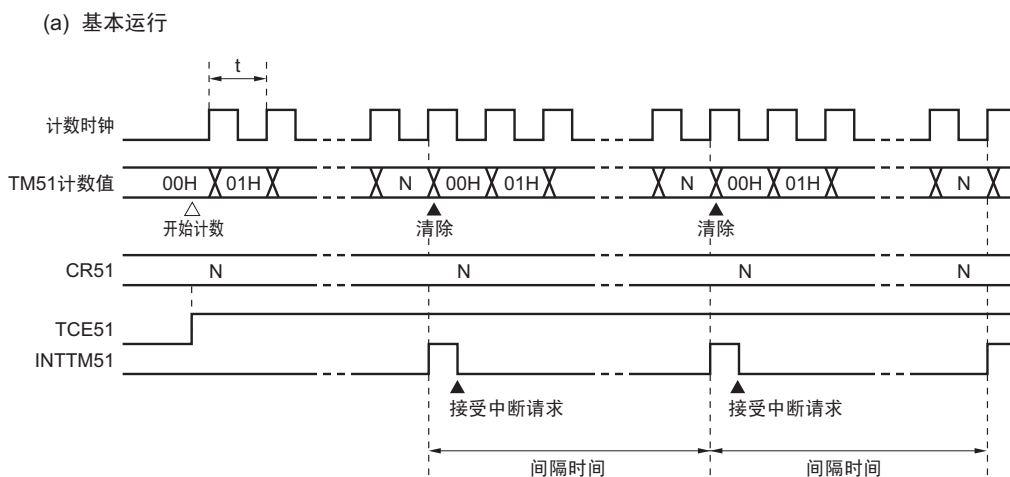
#### 设置方法

- ① 设置各寄存器。
  - TCL51: 选择计数时钟
  - CR51: 比较值
  - TMC51: 停止计数运行、选择 TM51 与 CR51 匹配时的清除 & 启动模式  
(TMC51=0000xx0B x: 忽略)
- ② 如果将 TCE51 置“1”, 开始计数运行。
- ③ 如果 TM51 与 CR51 的值匹配, 就产生 INTTM51 (TM51 被清“00H”)。
- ④ 之后, 以相同的时间间隔重复产生 INTTM51。在停止计数运行时, 必须将 TCE51 置“0”。

**注意** 运行期间不能给 CR51 写入不同的值。

**备注** 有关 INTTM51 信号的中断允许, 请参照“第 14 章 中断功能”。

图 7-7 间隔定时器运行的时序 (1/2)

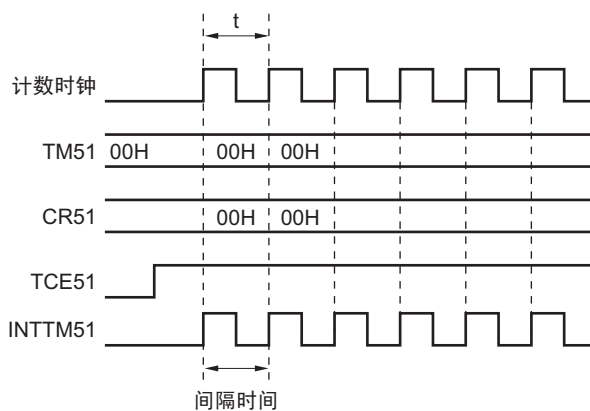


**备注** 间隔时间  $= (N+1) \times t$ 、 $N=01H \sim FFH$

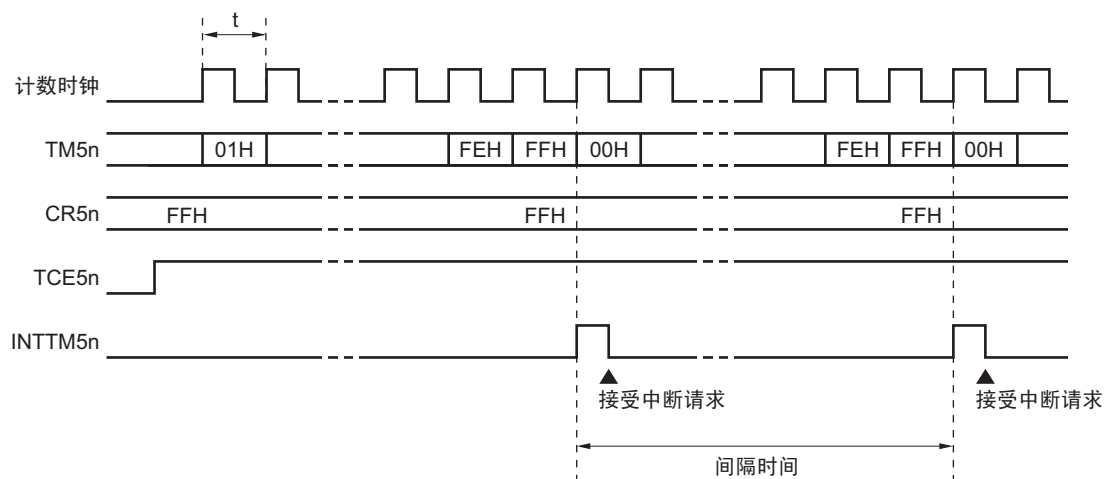


图 7-7 间隔定时器运行的时序 (2/2)

(b) 当CR51=00H时



(c) 当CR51=FFH时



## 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 51 (TM51) 对输入到 TI51 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 51 (TCL51) 指定的有效边沿, TM51 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 TM51 的计数值与 8 位定时器比较寄存器 51 (CR51) 的值匹配, TM51 就清“0”, 并且产生中断请求信号 (INTTM51)。

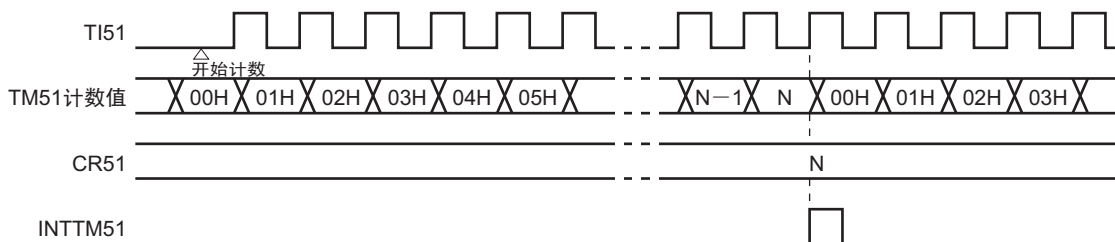
之后, 每当 TM51 的值与 CR51 的值匹配, 就产生 INTTM51。

### 设置方法

- ① 设置各寄存器。
  - 将端口模式寄存器 (PM30) 置“1”
  - TCL51: 选择 TI51 引脚输入的边沿  
TI51 引脚的下降沿 → TCL51=00H  
TI51 引脚的上升沿 → TCL51=01H
  - CR51: 比较值
  - TMC51: 停止计数、选择 TM51 与 CR51 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。  
(TMC51=00000000B)
- ② 如果将 TCE51 置“1”, 就对 TI51 引脚的输入脉冲进行计数。
- ③ 如果 TM51 的值与 CR51 的值匹配, 就产生 INTTM51 (TM51 被清“00H”)。
- ④ 之后, 每当 TM51 的值与 CR51 的值匹配, 就将产生 INTTM51。

备注 有关 INTTM51 信号的中断允许, 请参照“第 14 章 中断功能”。

图 7-8 外部事件计数器运行的时序 (指定上升沿时)



备注 N=00H ~ FFH

### 7.4.3 方波的输出运行

以输出 8 位定时器比较寄存器 51 (CR51) 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 51 (TMC51) 的 bit0 (TOE51) 置“1”，以 CR51 预置的计数值决定的间隔取反定时器 51 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。

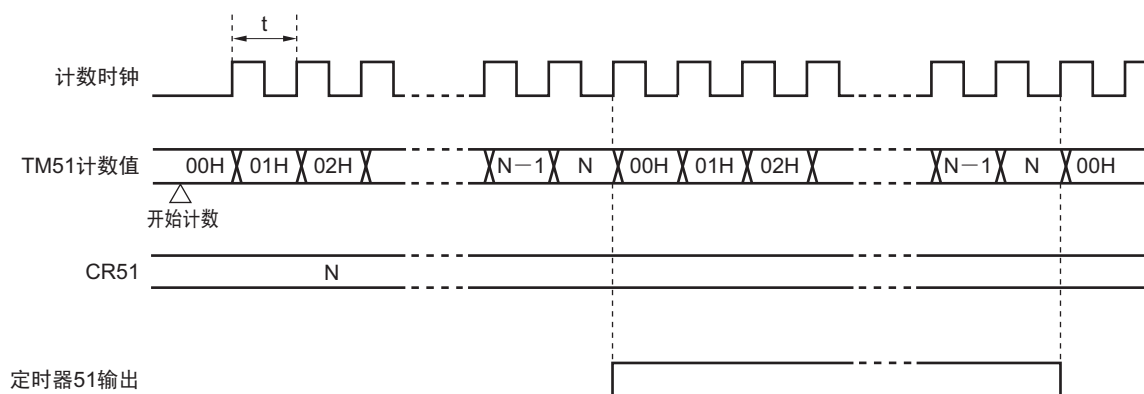
#### 设置方法

- ① 设置各寄存器。
  - 将端口输出锁存器 (P30) 和端口模式寄存器 (PM30) 置“0”
  - TCLK51: 选择计数时钟
  - CR51: 比较值
  - TMC51: 停止计数运行、选择TM51与CR51匹配时的清除&启动模式 (TMC51=00000010B)
- ② 如果将TCE51置“1”，开始计数运行。
- ③ 如果TM51与CR51的值匹配，取反定时器输出F/F。另外，产生INTTM51，并且TM51被清“00H”。
- ④ 之后，以相同间隔取反定时器输出F/F，并且输出方波。频率如下所示。
  - 频率=1/2t(N+1)  
(N: 00H ~ FFH)

**注意** 运行期间不能给 CR51 写入不同的值。

**备注** 有关 INTTM51 信号的中断允许，请参照“第 14 章 中断功能”。

图 7-9 方波输出运行的时序



### 7.4.4 PWM 的输出运行

通过将 8 位定时器模式控制寄存器 51 (TMC51) 的 bit6 (TMC516) 置“1”，进行 PWM 输出运行。输出由 8 位定时器比较寄存器 51 (CR51) 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 CR51，并且可通过 TMC51 的 bit1 (TMC511) 选择有效电平。可通过定时器时钟选择寄存器 51 (TCL51) 的 bit0 ~ bit2 (TCL510 ~ TCL512)，选择计数时钟。

**注意** 在 PWM 模式中，CR51 的改写间隔至少为 3 个计数时钟 (由 TCL51 选择的时钟)。

**备注** PWM 输出仅可用作 UART0 的基本时钟。

#### (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 (P30) 和端口模式寄存器 (PM30) 置“0”
  - TCL51: 选择计数时钟
  - CR51: 比较值
  - TMC51: 停止计数运行、选择 PWM 模式

TMC511	选择有效电平
0	高电平有效
1	低电平有效

(TMC51=01000000B 或者 01000010B)

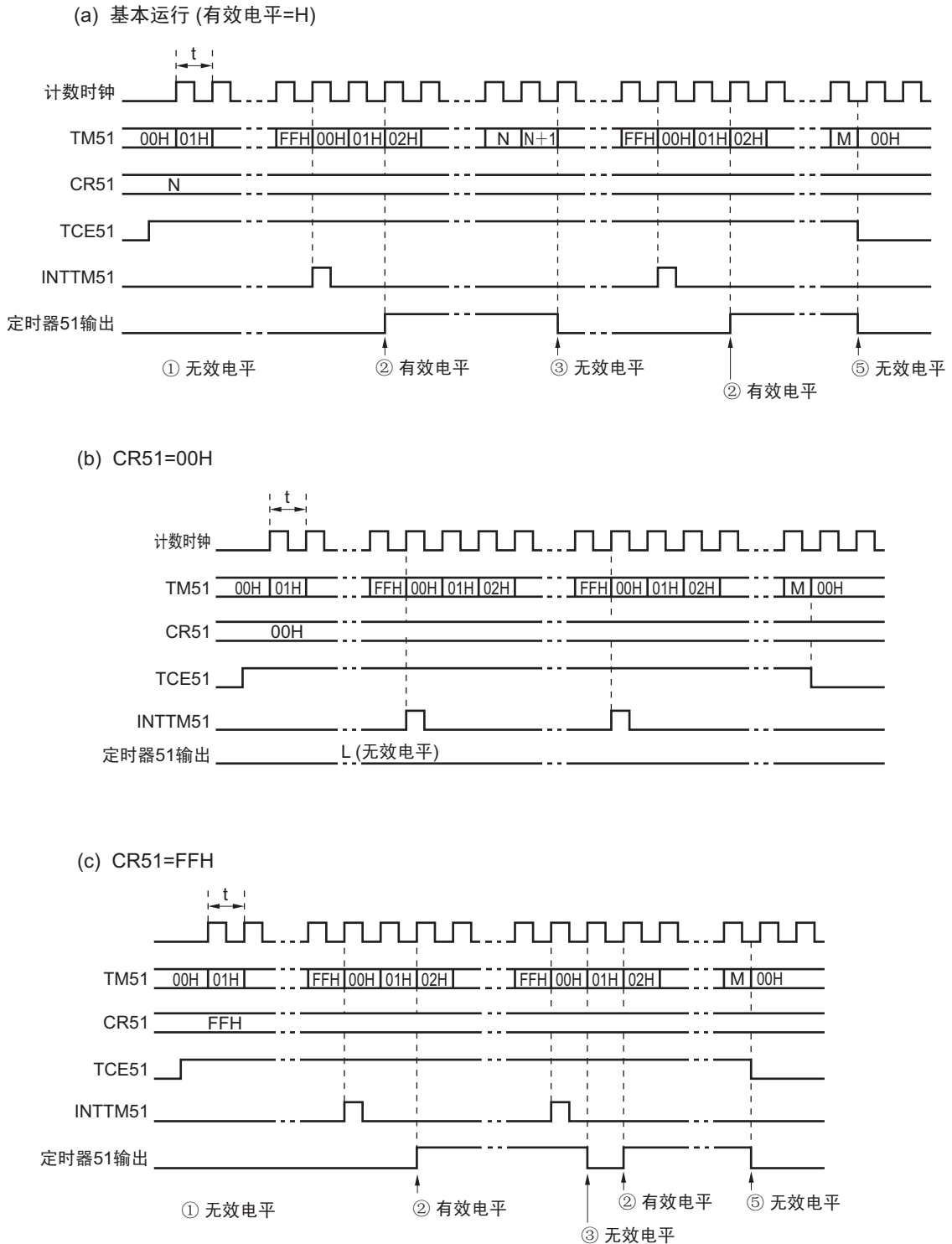
- ② 如果将 TCE51 置“1”，就开始计数运行。  
如果要停止计数，必须将 TCE51 置“0”。

##### PWM 输出

- ① 直到发生上溢前，PWM 输出 (定时器 51 输出) 为无效电平输出。
- ② 如果发生上溢，输出有效电平，直到 CR51 与 8 位定时器计数器 51 (TM51) 的计数值匹配。
- ③ 如果 CR51 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 TCE51 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 7-10 和图 7-11。  
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 =  $N/2^8$   
( $N=00H \sim FFH$ )

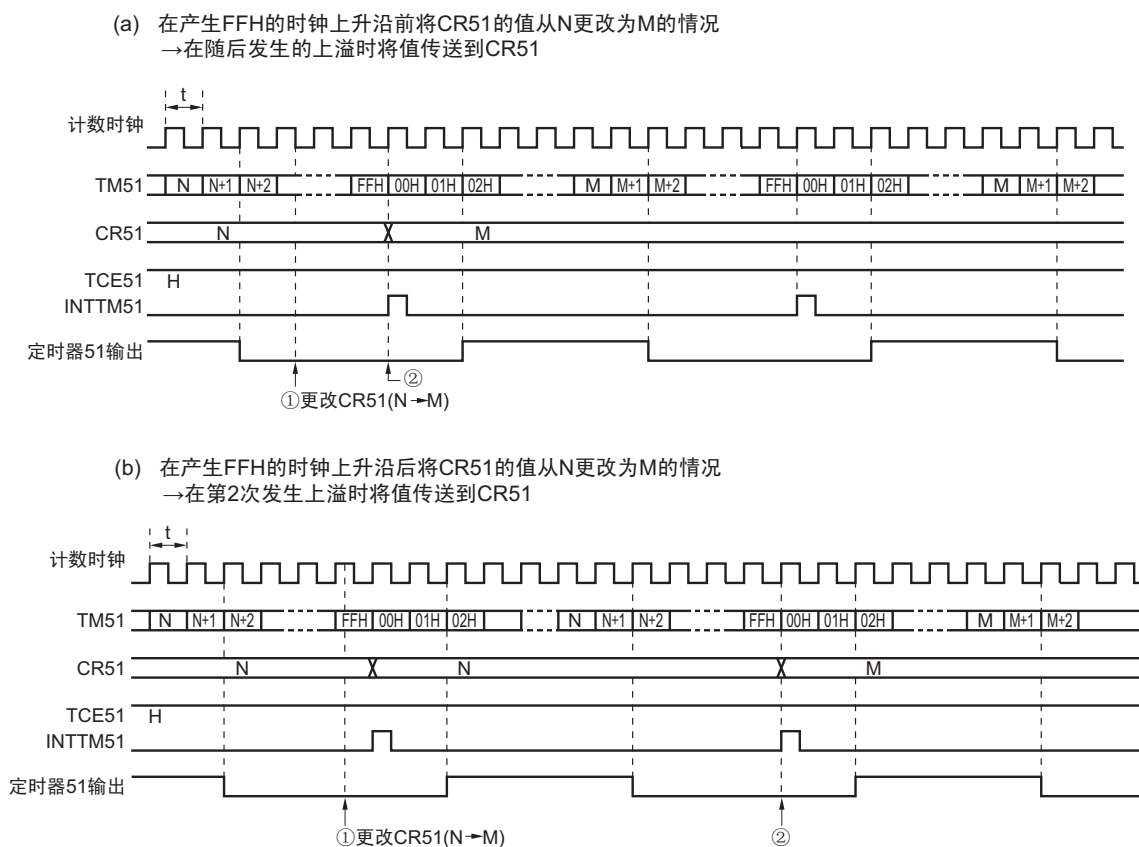
图 7-10 PWM 输出运行的时序



备注 图 7-10(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) 更改 CR51 的运行

图 7-11 更改 CR51 的运行时序



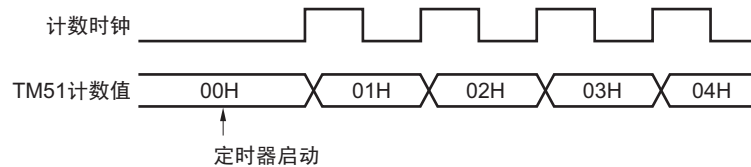
注意 在图 7-11 中的①与②之间读 CR51 时，读取值与实际运行的值不同（读取值：M、实际的 CR51 值：N）。

## 7.5 8 位定时器 / 事件计数器 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 51（TM51）没有与计数时钟同步启动而导致的。

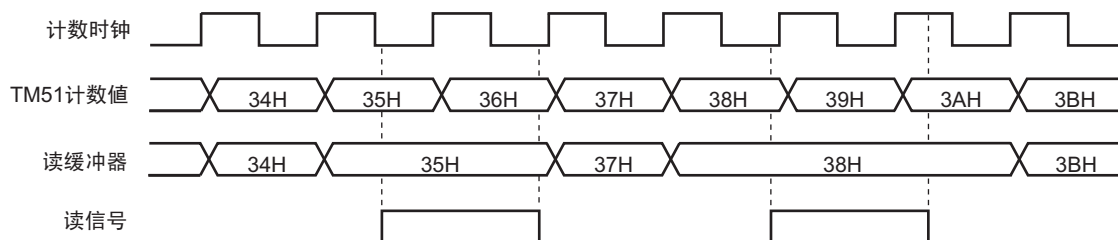
图 7-12 8 位定时器计数器 51（TM51）的启动时序



### (2) 8 位定时器计数器 51（TM51）的读取

因为是在将捕捉到缓冲器的计数值固定后读取 TM51，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 7-13 8 位定时器计数器 51（TM51）的读取时序



## 第 8 章 8 位定时器 H1

### 8.1 8 位定时器 H1 的功能

R7F0C30x、R7F0C31x 的所有产品都装有 8 位定时器 H1。

8 位定时器 H1 具有以下功能。

- 间隔定时器
- 方波输出
- PWM 输出
- 载波发生器

### 8.2 8 位定时器 H1 的结构

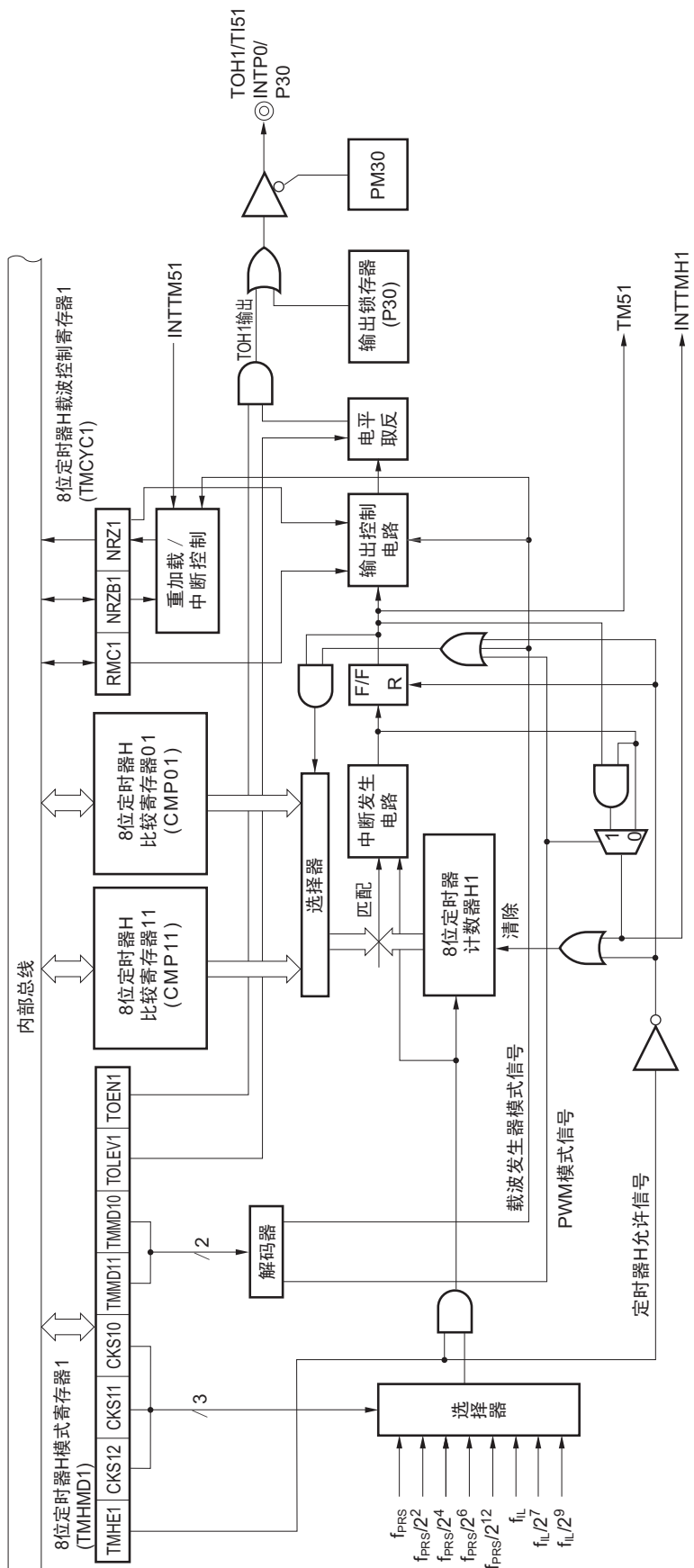
8 位定时器 H1 由以下硬件构成。

表 8-1 8 位定时器 H1 的结构

项目	结构
定时器寄存器	8 位定时器计数器 H1
寄存器	8 位定时器 H 比较寄存器 01 (CMP01) 8 位定时器 H 比较寄存器 11 (CMP11)
定时器输出	TOH1、输出控制电路
控制寄存器	8 位定时器 H 模式寄存器 1 (TMHMD1) 8 位定时器 H 载波控制寄存器 1 (TMCYC1) 端口模式寄存器 3 (PM3) 端口寄存器 3 (P3)



图 8-1 8 位定时器 H1 的框图



**(1) 8 位定时器 H 比较寄存器 01 (CMP01)**

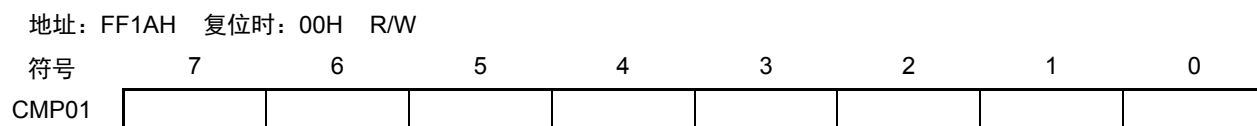
8 位定时器 H 比较寄存器 01 是可通过 8 位存储器操作指令读写的寄存器。用于所有定时器运行模式。

CMP01 的设置值总是和 8 位定时器计数器 H1 的计数值进行比较，如果两个值匹配，就产生中断请求信号 (INTTMH1)，并且取反 TOH1 的输出电平。

在定时器停止 (TMHE1=0) 期间改写 CMP01。

在产生复位信号时，CMP01 为“00H”。

图 8-2 8 位定时器 H 比较寄存器 01 (CMP01) 的格式



**注意** 不能在定时器计数运行期间改写 CMP01。但是，可在定时器计数运行期间刷新 CMP01 (写入相同的值)。

**(2) 8 位定时器 H 比较寄存器 11 (CMP11)**

8 位定时器 H 比较寄存器 11 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，CMP11 的设置值总是和 8 位定时器计数器 H1 的计数值进行比较，如果两个值匹配，就取反 TOH1 的输出电平，不产生中断请求信号。

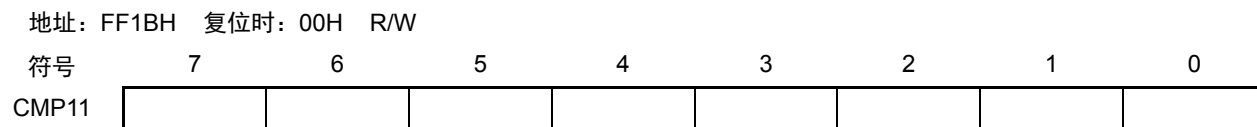
在载波发生器模式中，CMP11 的设置值总是和 8 位定时器计数器 H1 的计数值进行比较，如果两个值匹配，就产生中断请求信号 (INTTMH1)，并且同时清除计数值。

在定时器计数运行期间，可刷新 (写入相同的值) 和改写 CMP11。

如果在定时器计数运行期间改写 CMP11 的值，就锁存该值，并且在计数值与更改前的 CMP11 值匹配时传送到 CMP11，由此更改 CMP11 的值。如果计数值和 CMP11 值的匹配时序与 CMP11 的写入时序发生竞争，就不更改 CMP11 的值。

在产生复位信号时，CMP11 为“00H”。

图 8-3 8 位定时器 H 比较寄存器 11 (CMP11) 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 (TMHE1=0) 后重新开始定时器计数运行 (TMHE1=1)，必须设置 CMP11 (即使设置与 CMP11 相同的值，也必须重新设置)。

### 8.3 8 位定时器 H1 的控制寄存器

8 位定时器 H1 的控制寄存器有以下 4 种。

- 8 位定时器 H 模式寄存器 1 (TMHMD1)
- 8 位定时器 H 载波控制寄存器 1 (TMCYC1)
- 端口模式寄存器 3 (PM3)
- 端口寄存器 3 (P3)

#### (1) 8 位定时器 H 模式寄存器 1 (TMHMD1)

8 位定时器 H 模式寄存器 1 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 TMHMD1。  
在产生复位信号时，TMHMD1 为“00H”。

图 8-4 8 位定时器 H 模式寄存器 1 (TMHMD1) 的格式

地址: FF6CH 复位时: 00H R/W

符号	<7>	6	5	4	3	2	<1>	<0>
CRC00	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	选择计数时钟注			
			$f_{PRS}=2\text{MHz}$	$f_{PRS}=5\text{MHz}$	$f_{PRS}=10\text{MHz}$	
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	$f_{IL}/2^7$	1.88kHz(TYP.)		
1	1	0	$f_{IL}/2^9$	0.47kHz(TYP.)		
1	1	1	$f_{IL}$	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 外围硬件时钟 ( $f_{PRS}$ ) 以高速系统时钟 ( $f_{XH}$ ) 运行 ( $XSEL=1$ ) 时,  $f_{PRS}$  的工作频率因电源电压而不同。

- $V_{DD}=2.7 \sim 5.5\text{V}$ :  $f_{PRS} \leq 10\text{MHz}$
- $V_{DD}=1.8 \sim 2.7\text{V}$ :  $f_{PRS} \leq 5\text{MHz}$

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)。

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 (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)。
3. 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。
4. 实际的 TOH1/TI51/INTP0/P30 引脚的 TOH1 输出取决于 PM30 和 P30。

备注 1.  $f_{PRS}$ : 外围硬件时钟频率

2.  $f_{IL}$ : 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。

在产生复位信号时，TMCYC1 为“00H”。

图 8-5 8 位定时器 H 载波控制寄存器 1 (TMCYC1) 的格式

地址：FF71H 复位时：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 3 (PM3)

端口模式寄存器 3 以 1 位单位设置端口 3 的输入 / 输出。

在将 P30/TOH1/TI51/INTP0 引脚用作定时器输出时，必须将 PM30 和 P30 的输出锁存器置“0”。

通过 1 位或 8 位存储器操作指令设置 PM3。

在产生复位信号时，PM3 为“FFH”。

图 8-6 端口模式寄存器 3 (PM3) 的格式

地址：FF23H 复位时：FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34 注	PM33 注	PM32	PM31	PM30

PM3n	选择 P3n 引脚的输入 / 输出模式 (n=0 ~ 4)
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

注 只限 20 引脚产品。

## 8.4 8 位定时器 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 H1 和比较寄存器 01 (CMP01) 匹配时, 产生中断请求信号 (INTTMH1), 并且将 8 位定时器计数器 H1 清“00H”。

在间隔定时器模式中不使用比较寄存器 11 (CMP11)。因此, 即使设置 CMP11 寄存器, 也不能检测出 8 位定时器计数器 H1 和 CMP11 寄存器匹配, 所以不影响定时器输出。

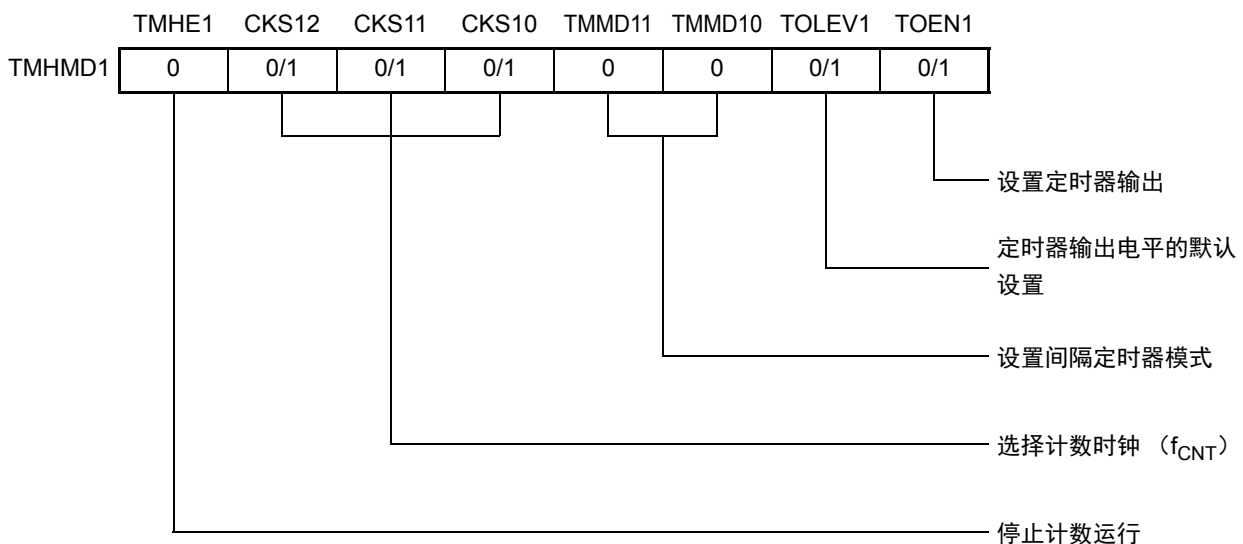
通过将定时器 H 模式寄存器 1 (TMHMD1) 的 bit0 (TOEN1) 置“1”, 从 TOH1 输出任意频率的方波 (占空比 =50%)。

#### 设置方法

- ① 设置各寄存器。

图 8-7 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 1 (TMHMD1)



- (ii) 设置 CMP01 寄存器

将 N 设置为比较值时, 间隔时间如下所示。

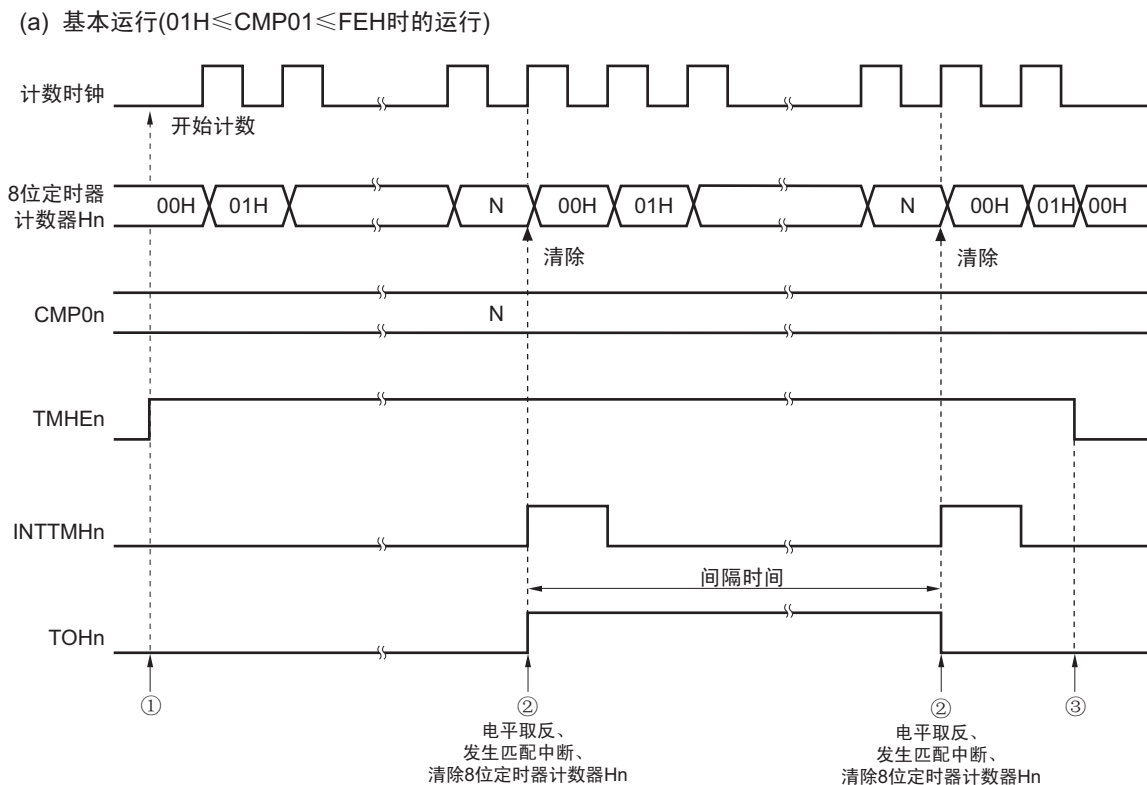
- 间隔时间 =  $(N+1)/f_{CNT}$

- ② 通过将 TMHE1 置“1”, 开始计数运行。
- ③ 如果 8 位定时器计数器 H1 和 CMP01 寄存器的值匹配, 就产生 INTTMH1 信号, 并且将 8 位定时器计数器 H1 清“00H”。
- ④ 之后, 以相同间隔产生 INTTMH1 信号。如果要停止计数运行, 将 TMHE1 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 3 (PM3)”。

2. 有关 INTTMH1 信号的中断允许, 请参照“第 14 章 中断功能”。

图 8-8 间隔定时器 / 方波输出运行的时序 (1/2)



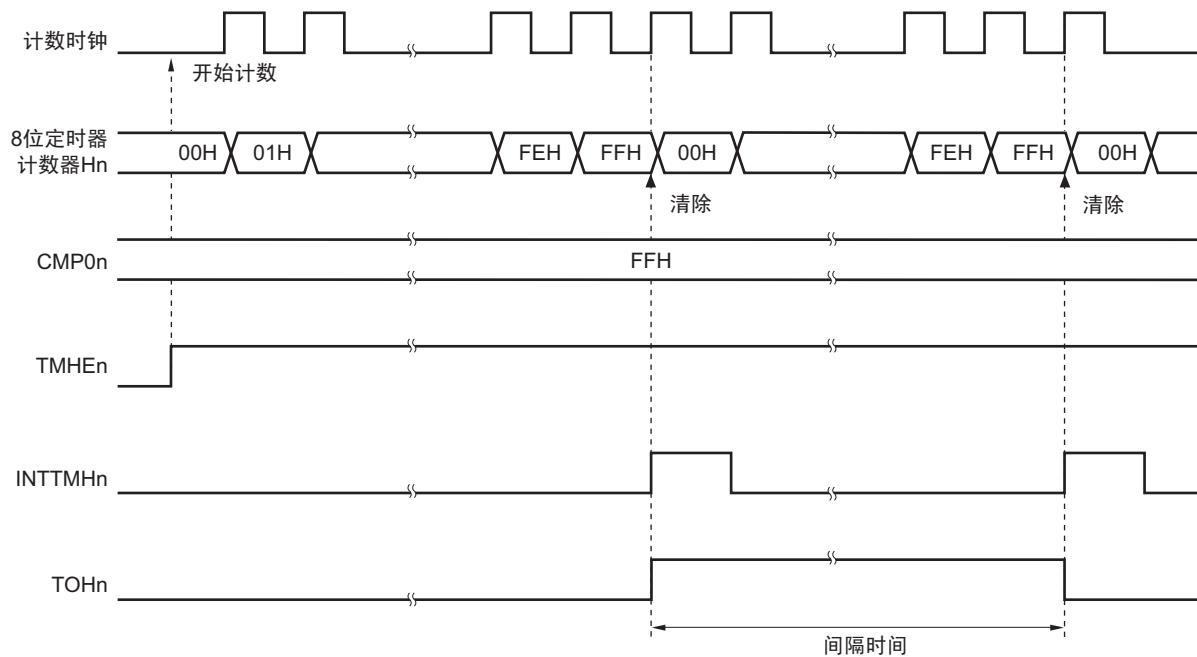
- ① 通过将TMHE1位从“0”置“1”，进入允许计数运行的状态。在允许运行后，计数时钟最迟1个时钟后开始计数。
- ② 如果8位定时器计数器H1的值与CMP01寄存器的值匹配，就清除8位定时器计数器H1的值，并且取反TOH1输出的电平。另外，在计数时钟上升时产生INTTMH1信号。
- ③ 如果在定时器H运行期间将TMHE1位置“0”，INTTMH1信号和TOH1输出就为默认状态。如果在将TMHE1位置“0”前，已经是与默认状态相同的状态，就保持电平。

备注 1.  $01H \leq N \leq FEH$

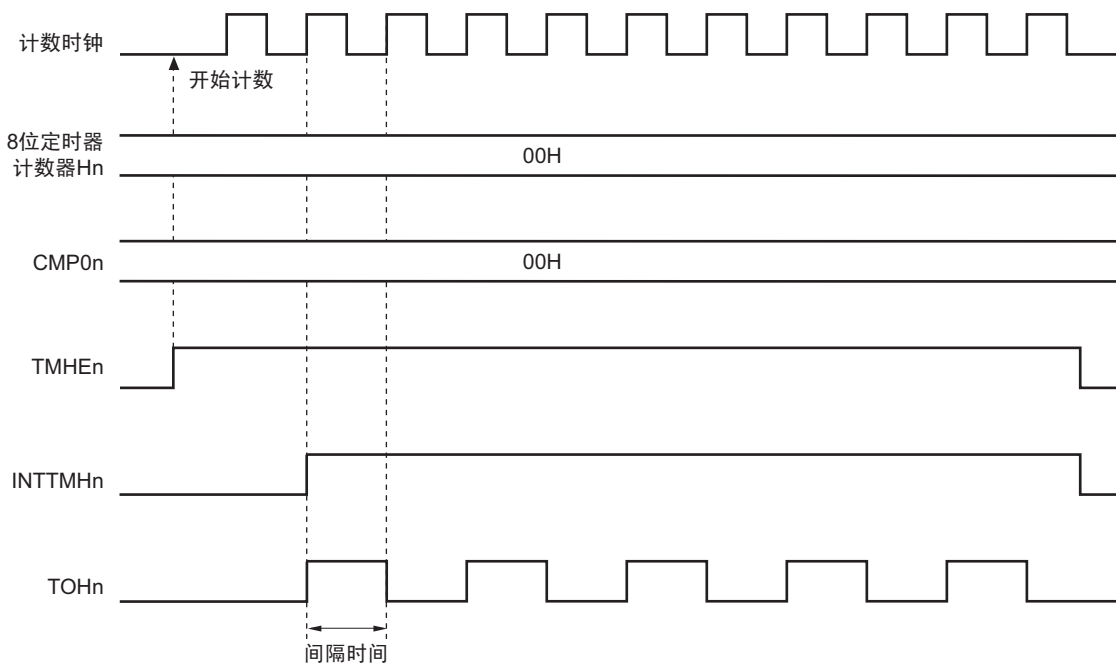
2.  $n=1$

图 8-8 间隔定时器 / 方波输出运行的时序 (2/2)

## (b) CMP01=FFH时的运行



## (c) CMP01=00H时的运行



备注 n=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 01 (CMP01) 控制定时器输出 (TOH1) 的周期。定时器运行期间禁止改写 CMP01 寄存器。

8 位定时器比较寄存器 11 (CMP11) 控制定时器输出 (TOH1) 的占空比。可在定时器运行作期间改写 CMP11 寄存器。

在 PWM 输出模式中的运行如下所示。

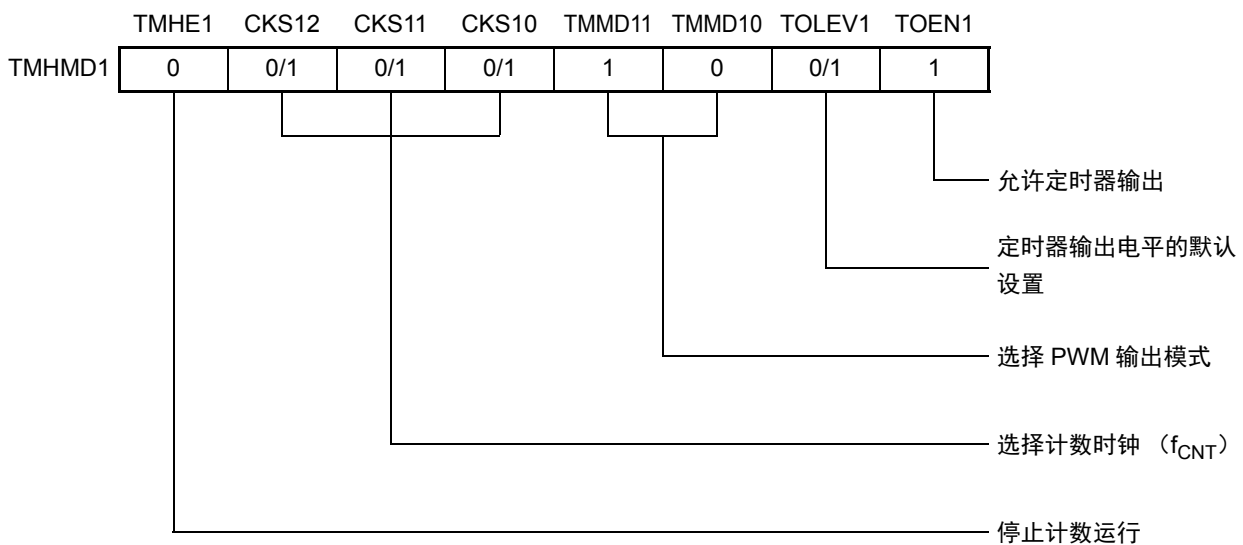
开始定时器计数后, 如果 8 位定时器计数器 H1 与 CMP01 寄存器的值匹配, PWM 输出 (TOH1 输出) 就为有效电平输出, 并且将 8 位定时器计数器 H1 清“0”。如果 8 位定时器计数器 H1 与 CMP11 寄存器的值匹配, PWM 输出 (TOH1 输出) 就为无效电平输出。

### 设置方法

- ① 设置各寄存器。

图 8-9 PWM 输出模式中的寄存器设置

- (i) 设置定时器 H 模式寄存器 1 (TMHMD1)



- (ii) 设置 CMP01 寄存器

- 比较值(N): 设置周期

- (iii) 设置 CMP11 寄存器

- 比较值(M): 设置占空比

备注  $00H \leq \text{CMP11}(M) < \text{CMP01}(N) \leq \text{FFH}$

- ② 通过将 TMHE1 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 CMP01 寄存器。如果 8 位定时器计数器 H1 与 CMP01 寄存器的值匹配，就清除 8 位定时器计数器 H1，产生中断请求信号（INTTMH1），并且输出有效电平。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。
- ④ 如果 8 位定时器计数器 H1 与 CMP11 寄存器匹配，输出无效电平。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。此时，不清除 8 位定时器计数器 H1，也不产生 INTTMH1 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 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 PWM 脉冲输出周期和占空比如下所示。

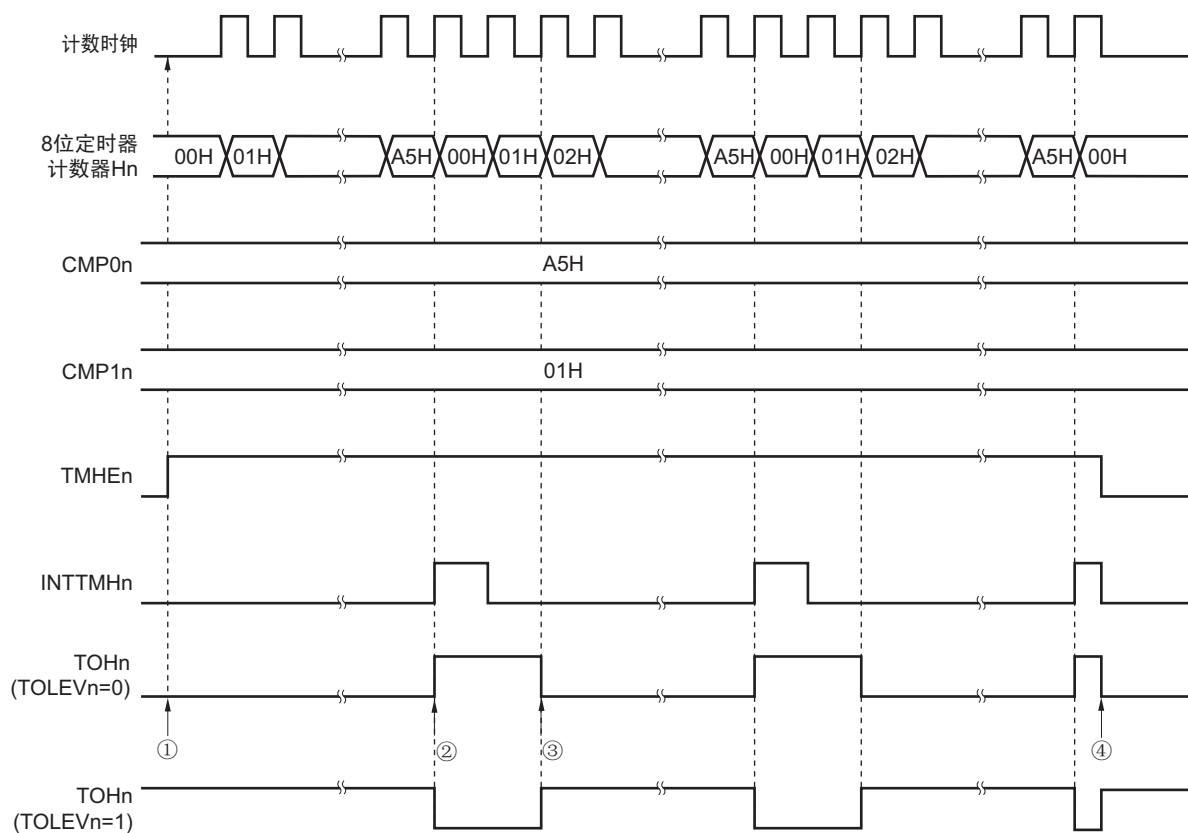
- PWM 脉冲输出周期  $= (N+1)/f_{CNT}$
- 占空比  $= (M+1)/(N+1)$

- 注意 1. 在定时器计数运行期间，可更改 CMP11 寄存器的设置值。但是，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 个运行时钟（通过 TMHMD1 寄存器的 CKS12 ~ CKS10 位选择的信号）。
2. 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（即使设定与 CMP11 相同的值，也必须重新设置）。
  3. CMP11 寄存器的设定值（M）和 CMP01 寄存器的设定值（N）必须在以下范围内。  
 $00H \leq \text{CMP11}(M) < \text{CMP01}(N) \leq FFH$

- 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 3（PM3）”。
2. 有关 INTTMH1 信号的中断允许，请参照“第 14 章 中断功能”。

图 8-10 PWM 输出运行的时序 (1/4)

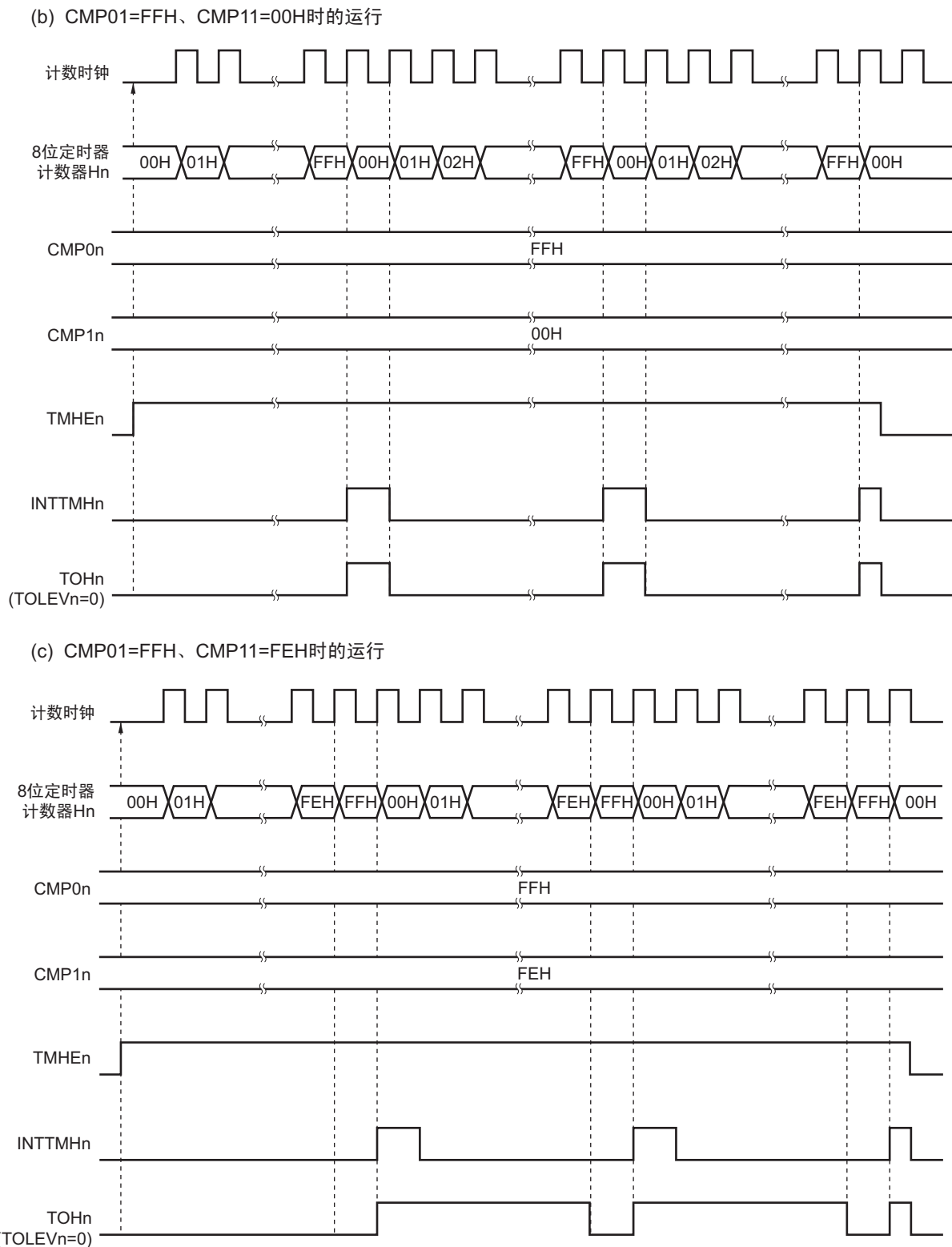
## (a) 基本运行



- ① 通过将TMHE1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H1，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H1的值与CMP01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H1，并且产生INTTMH1信号。
- ③ 如果8位定时器计数器H1的值与CMP11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INTTMH1信号。
- ④ 通过在定时器H1运行期间将TMHE1位置“0”，INTTMH1信号为默认状态，并且PWM输出为无效电平输出。

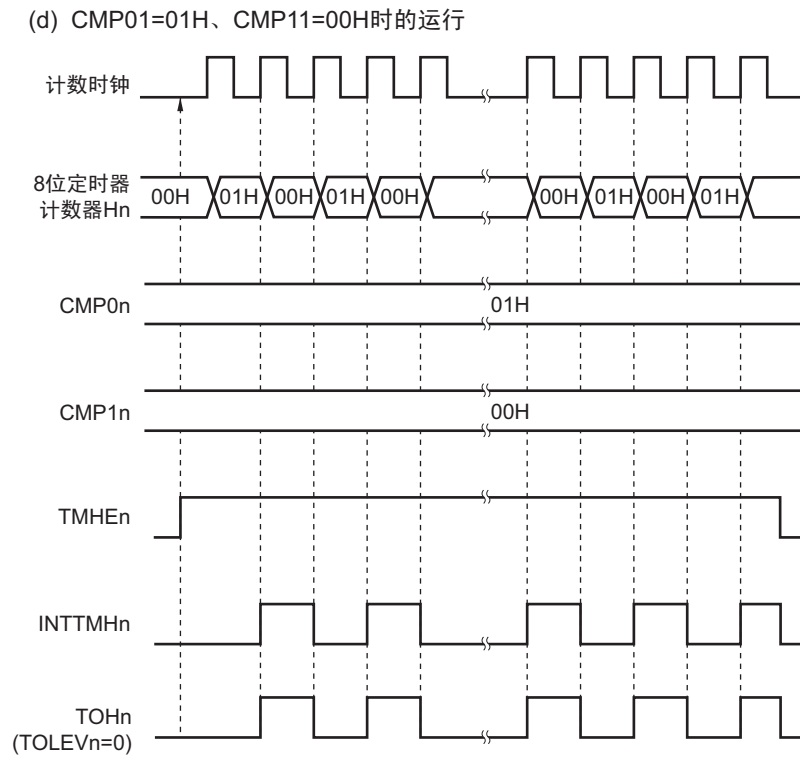
备注 n=1

图 8-10 PWM 输出运行的时序 (2/4)



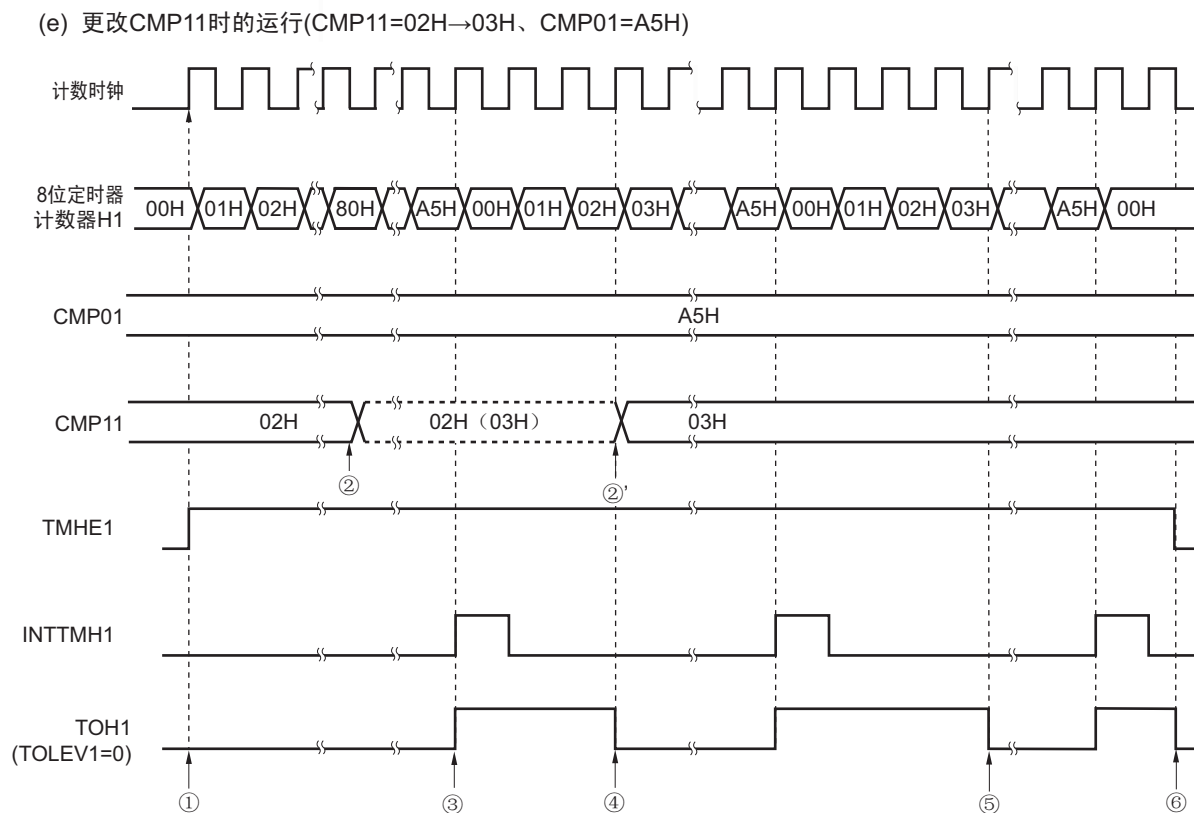
备注 n=1

图 8-10 PWM 输出运行的时序 (3/4)



备注 n=1

图 8-10 PWM 输出运行的时序 (4/4)



- ① 通过将TMHE1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H1，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CMP11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H1的值与CMP01寄存器的值匹配，就清除8位定时器计数器H1，输出有效电平，并且产生INTTMH1信号。
- ④ 即使更改CMP11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H1的值与CMP11寄存器更改前的值匹配，就将值传送到CMP11寄存器，从而更改CMP11寄存器的值（②'）。但是，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，也不能传送更改值到寄存器。
- ⑤ 如果8位定时器计数器H1的值与更改后的CMP11寄存器的值匹配，输出无效电平。不清除8位定时器计数器H1，也不产生INTTMH1信号。
- ⑥ 通过在定时器H1运行期间将TMHE1位置“0”，INTTMH1信号为默认状态，PWM输出为无效电平输出。

备注 n=1

### 8.4.3 作为载波发生器的运行

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。

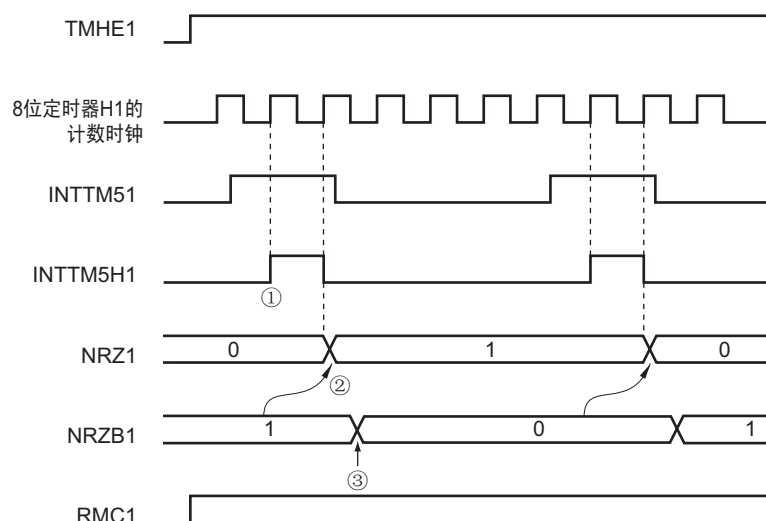
#### (2) 载波的输出控制

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 8-11 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。

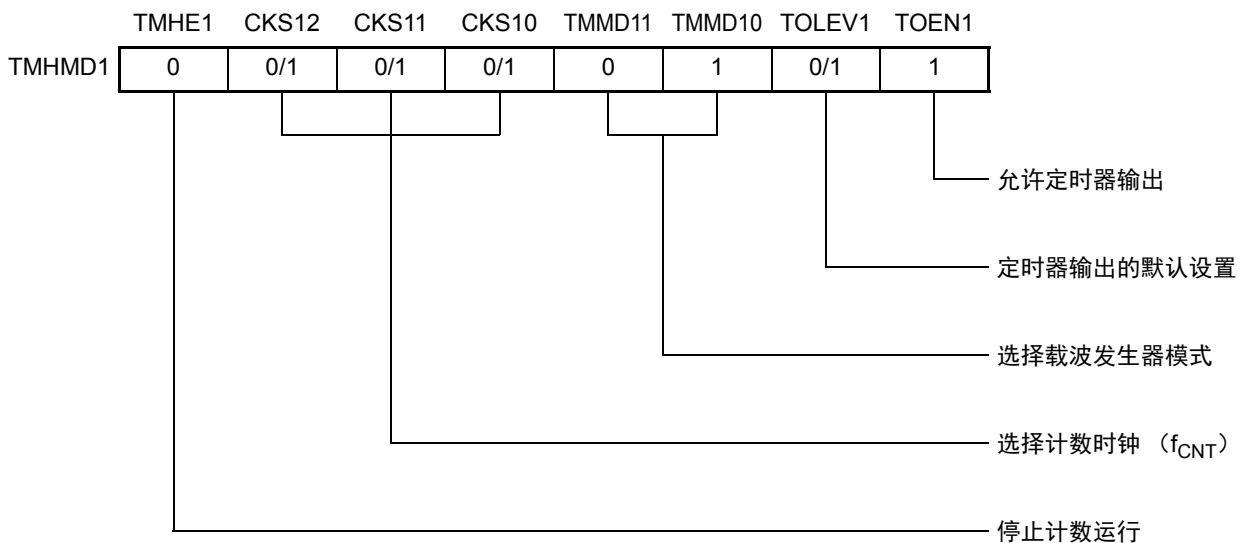
备注 INTTM5H1 是内部信号，不是中断源。

### 设置方法

- ① 设置各寄存器。

图 8-12 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 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”。

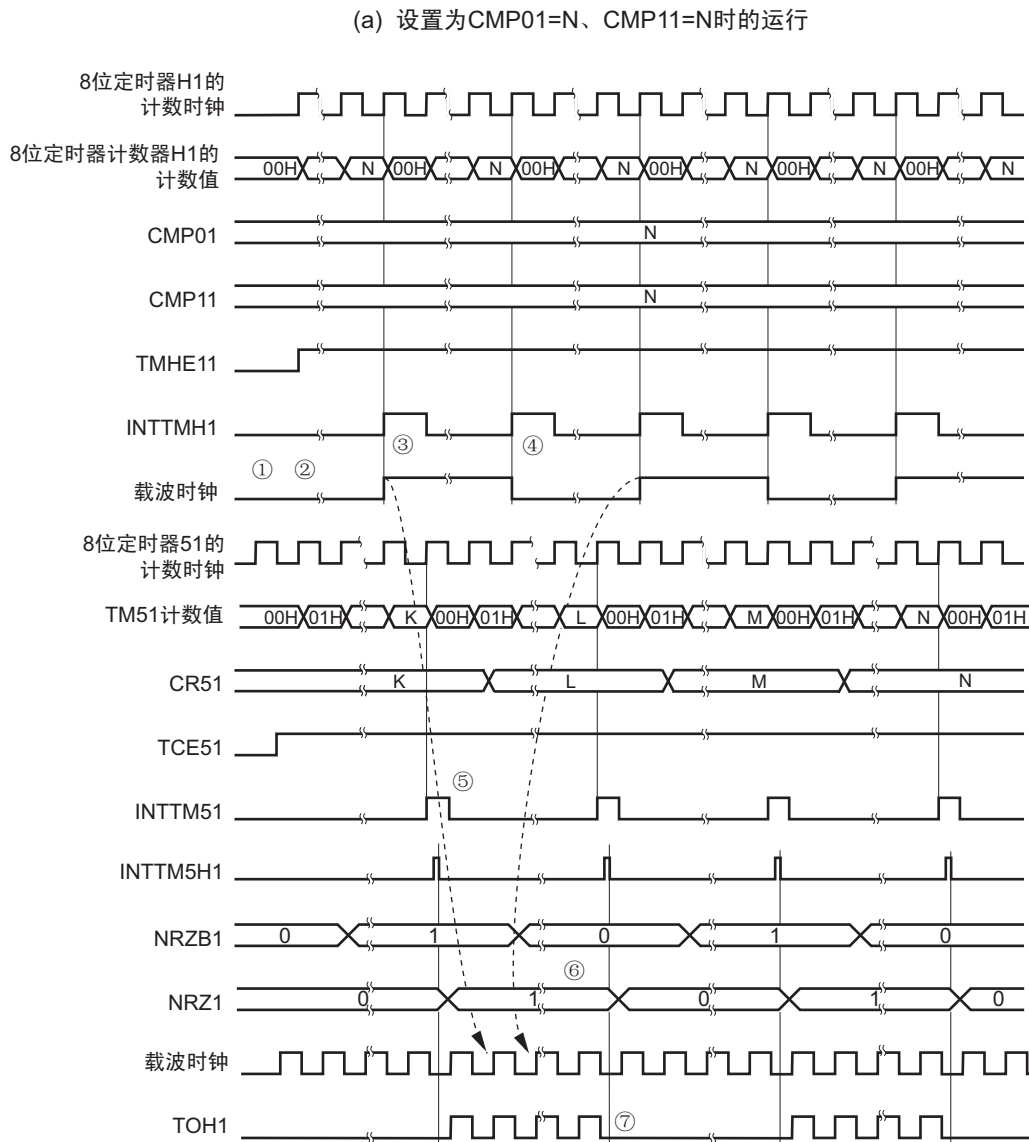
设 CMP01 寄存器的设置值为 N、CMP11 寄存器的设置值为 M、计数时钟频率为  $f_{CNT}$ ，则载波时钟输出周期和占空比如下所示。

- 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 3（PM3）”。
2. 有关 INTTMH1 信号的中断允许，请参照“第 14 章 中断功能”。

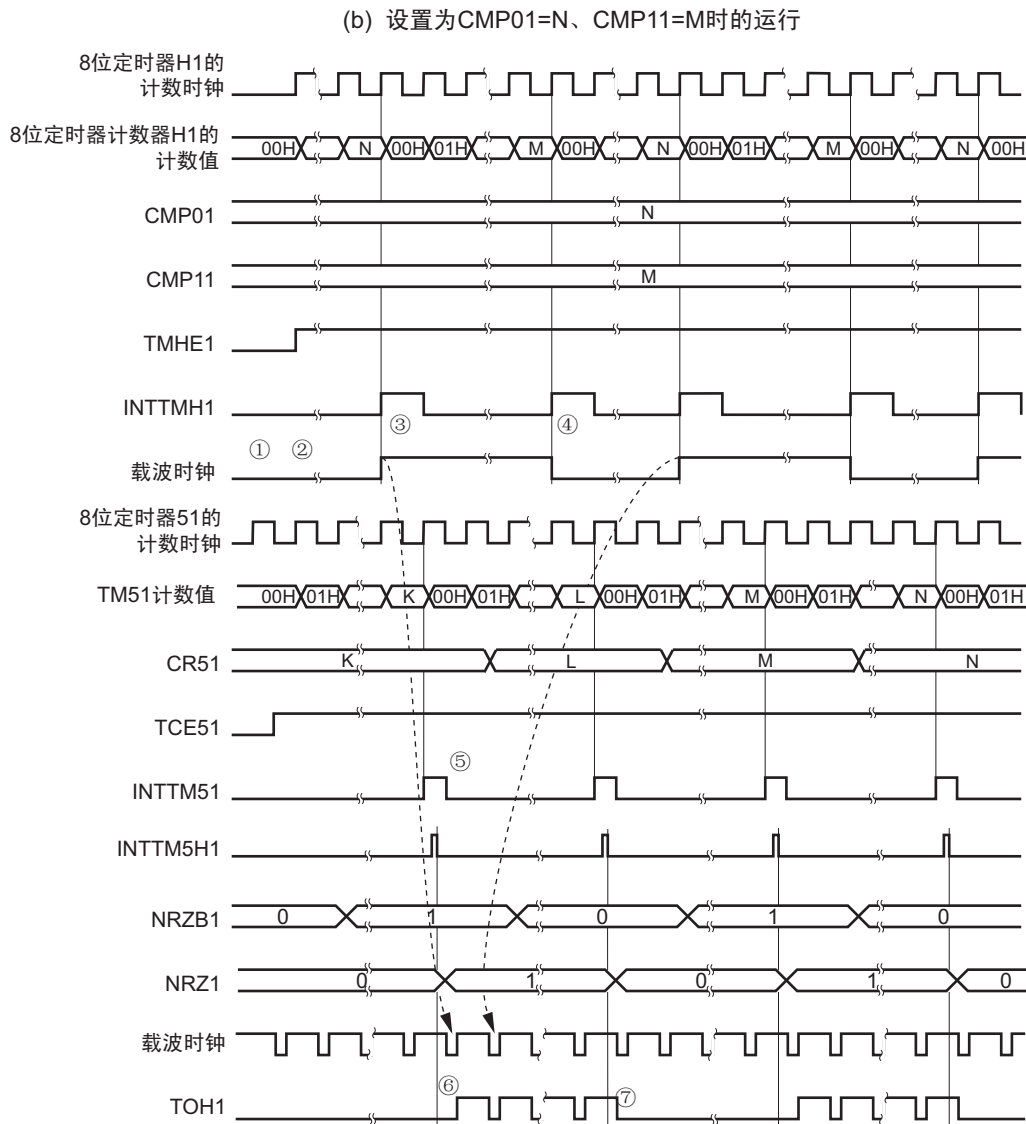
图 8-13 载波发生器模式的运行时序 (1/3)



- 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。

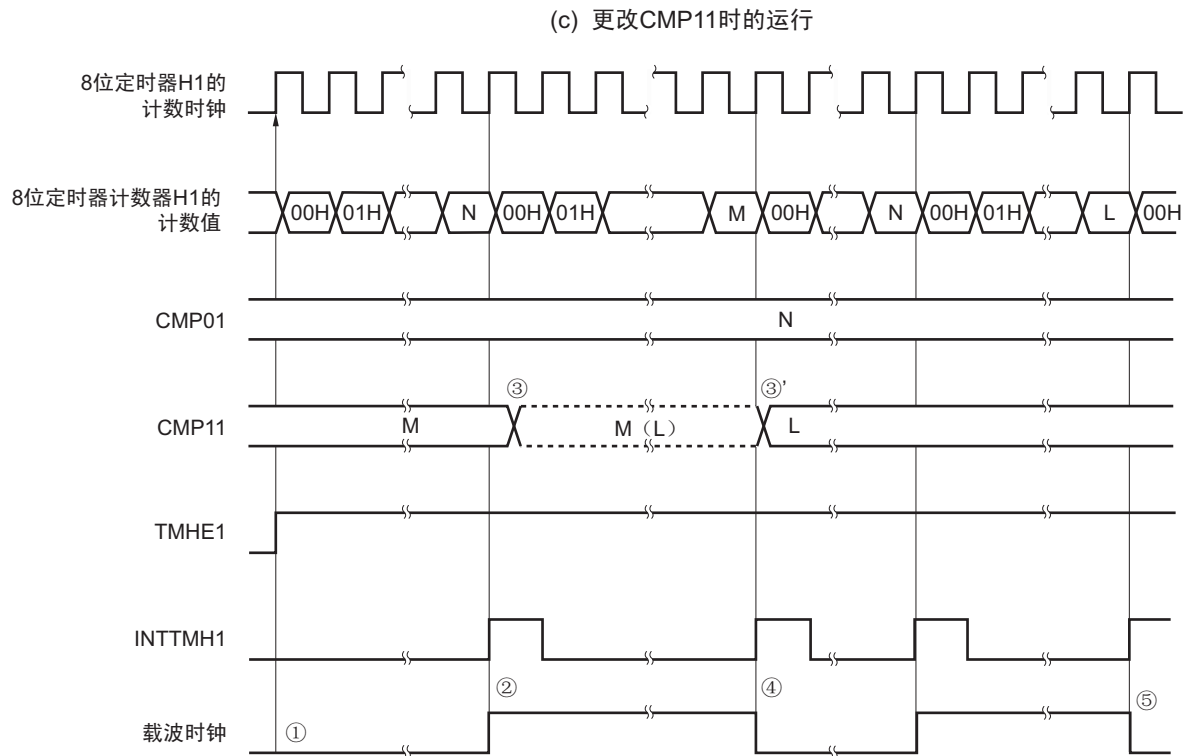
图 8-13 载波发生器模式的运行时序 (2/3)



- 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 8-13 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 看门狗定时器

### 9.1 看门狗定时器的功能

看门狗定时器通过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～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16章 复位功能”。

## 9.2 看门狗定时器的结构

看门狗定时器由以下硬件构成。

表 9-1 看门狗定时器的结构

项目	结构
控制寄存器	看门狗定时器允许寄存器 (WDTE)

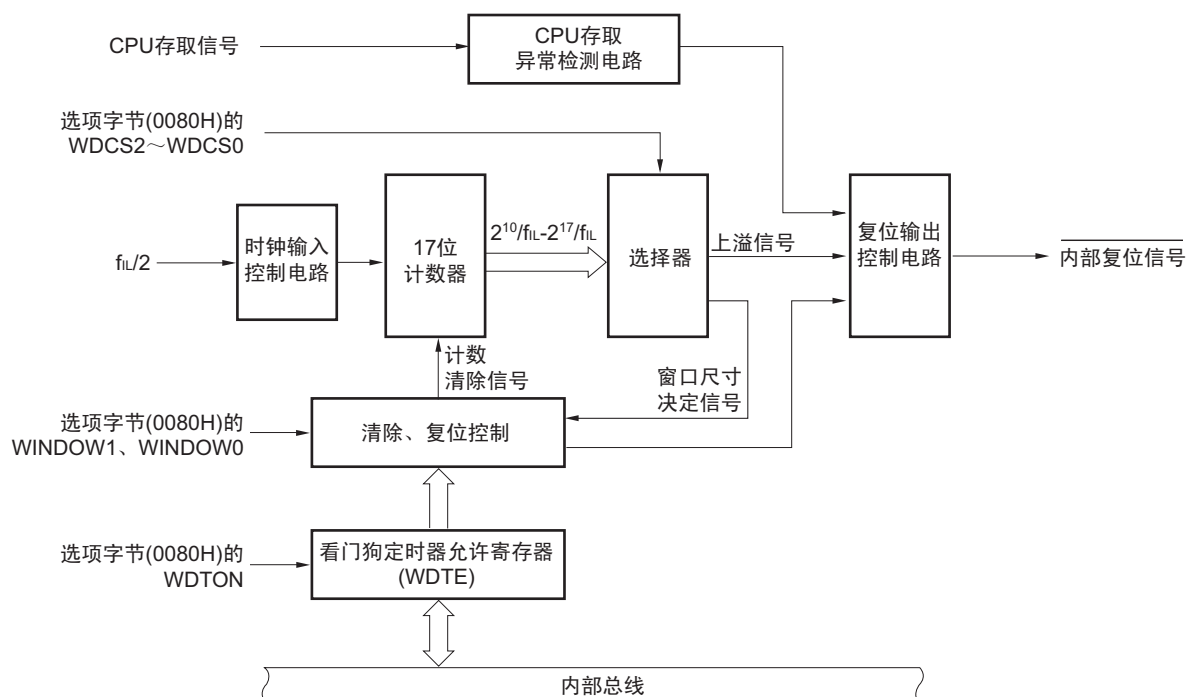
通过选项字节，进行计数器运行的控制，以及上溢时间和窗口打开期间的设置。

表 9-2 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20章 选项字节”。

图 9-1 看门狗定时器的框图



### 9.3 看门狗定时器的控制寄存器

通过看门狗定时器允许寄存器（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。

在产生复位信号时，WDTE 为“9AH”或者“1AH”注。

图 9-2 看门狗定时器允许寄存器（WDTE）的格式

地址：FF99H 复位时：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）。

## 9.4 看门狗定时器的运行

### 9.4.1 看门狗定时器的运行控制

- 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20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9.4.3和第20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9.4.3和第20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寄存器设置的区域（FB00H～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 $2/f_{IL}$ 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”。

- 即使在闪存的自编程期间，看门狗定时器也持续运行。但是，在此处理期间，中断的接受时间会出现延迟，因此，在设置上溢时间和窗口尺寸时，应充分考虑该延迟时间。



## 9.4.2 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 9-3 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	$2^{10}/f_{iL}$ (3.88ms)
0	0	1	$2^{11}/f_{iL}$ (7.76ms)
0	1	0	$2^{12}/f_{iL}$ (15.52ms)
0	1	1	$2^{13}/f_{iL}$ (31.03ms)
1	0	0	$2^{14}/f_{iL}$ (62.06ms)
1	0	1	$2^{15}/f_{iL}$ (124.12ms)
1	1	0	$2^{16}/f_{iL}$ (248.24ms)
1	1	1	$2^{17}/f_{iL}$ (496.48ms)

注意 1. 禁止 WDCS2=WDCS1=WDCS0=0 和 WINDOW1=WINDOW0=0 的组合设置。

- 即使在闪存的自编程期间，看门狗定时器也持续运行。但是，在此处理期间，中断的接受时间会出现延迟，因此，在设置上溢时间和窗口尺寸时，应充分考虑该延迟时间。

备注 1.  $f_{iL}$ ：低速内部振荡时钟频率

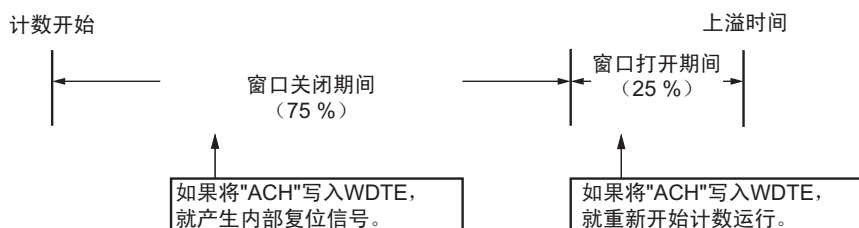
- ( ) 内： $f_{iL}=33\text{kHz (MAX.)}$

### 9.4.3 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 9-4 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 1.** 禁止 WDCS2=WDCS1=WDCS0=0 和 WINDOW1=WINDOW0=0 的组合设置。

2. 即使在闪存的自编程期间，看门狗定时器也持续运行。但是，在此处理期间，中断的接受时间会出现延迟，因此在设置上溢时间和窗口尺寸时，应充分考虑该延迟时间。

**备注** 如果将上溢时间设置为  $2^{11}/f_{IL}$ ，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：  
 $2^{11}/f_{IL}(\text{MAX.})=2^{11}/264\text{kHz}(\text{MAX.})=7.76\text{ms}$
- 窗口关闭时间：  
 $0 \sim 2^{11}/f_{IL}(\text{MIN.}) \times (1 - 0.25) = 0 \sim 2^{11}/216\text{kHz}(\text{MIN.}) \times 0.75 = 0 \sim 7.11\text{ms}$
- 窗口打开时间：  
 $2^{11}/f_{IL}(\text{MIN.}) \times (1 - 0.25) \sim 2^{11}/f_{IL}(\text{MAX.}) = 2^{11}/216\text{kHz}(\text{MIN.}) \times 0.75 \sim 2^{11}/264\text{kHz}(\text{MAX.}) = 7.11 \sim 7.76\text{ms}$

## 第 10 章 A/D 转换器

项目	16 引脚	20 引脚
10 位 A/D 转换器	6ch	5ch

### 10.1 A/D 转换器的功能

A/D 转换器是将模拟输入转换为数字值的 10 位分辨率转换器。最多可控制 6 个通道（ANI0 ~ ANI2、ANI5 ~ ANI7）的模拟输入。

在内置运算放大器的产品中，ANI1 复用运算放大器 0 输出（AMP0OUT）的引脚（16 引脚产品时，ANI6 复用运算放大器 1 输出（AMP1OUT）的引脚）。因此，可将运算放大器输出用作模拟输入源。

A/D 转换器具有以下功能。

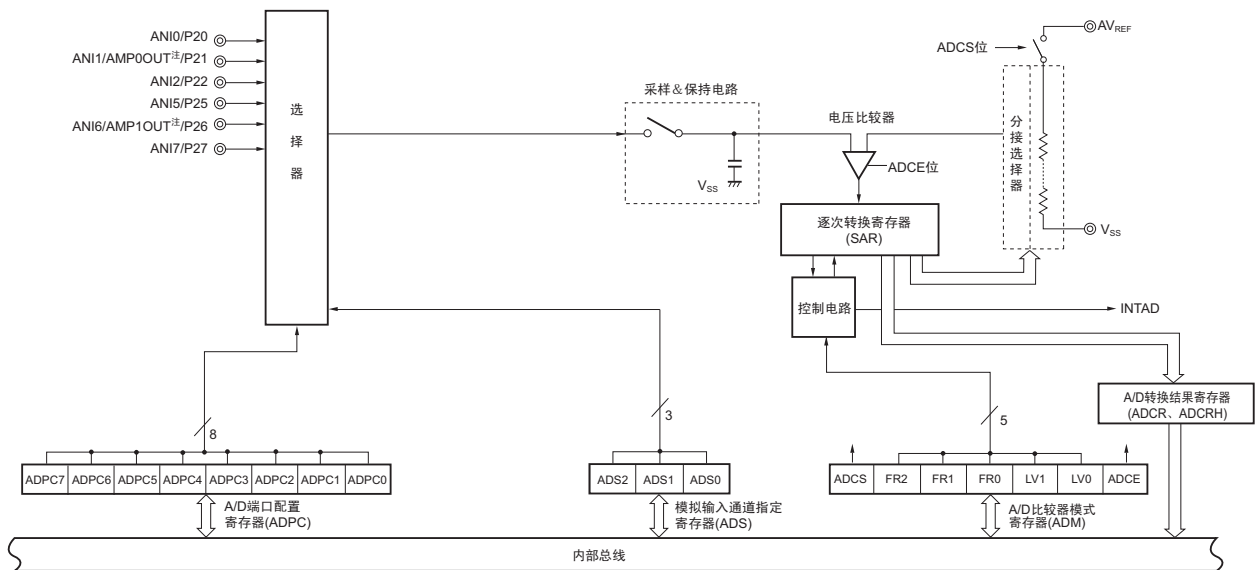
- 10 位分辨率 A/D 转换  
从 ANI0 ~ ANI7 和运算放大器输出中选择 1 个通道的模拟输入，并且重复执行 10 位分辨率的 A/D 转换运行。在每次 A/D 转换结束后，产生中断请求（INTAD）。

备注 A/D 转换器的模拟输入引脚因产品而不同。

- 16 引脚产品：ANI0 ~ ANI2、ANI5 ~ ANI7
- 20 引脚产品：ANI0 ~ ANI4

图 10-1 A/D 转换器的框图

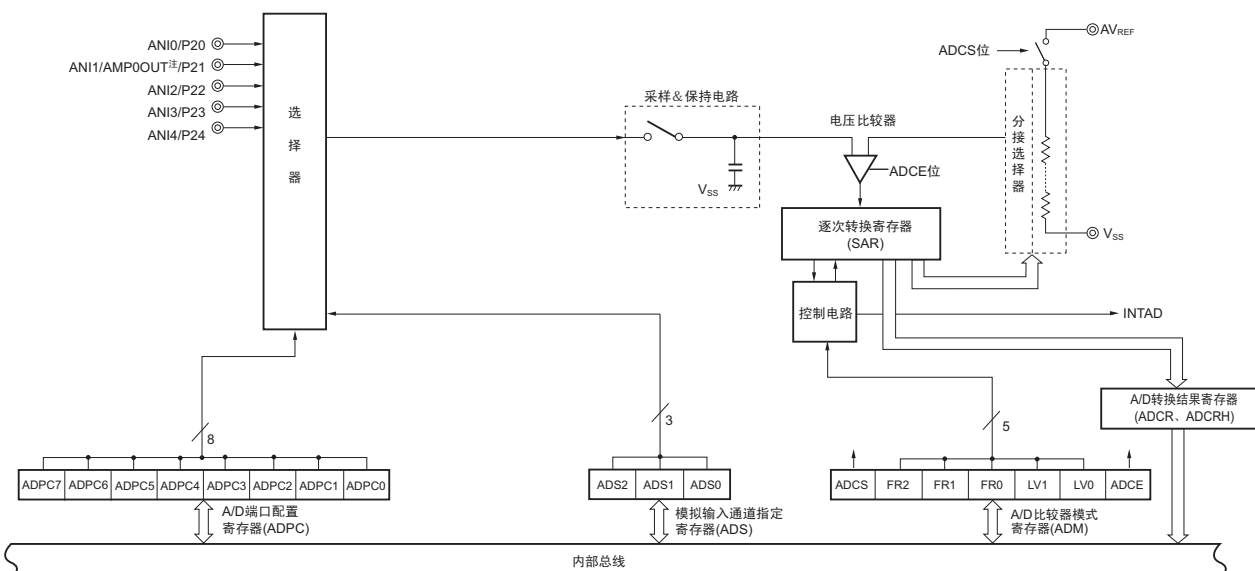
## (1) 16 引脚产品



注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114（内置运算放大器的产品）。

注意  $V_{SS}$  复用 A/D 转换器的接地电位。必须将  $V_{SS}$  连接至稳定的 GND (=0V)。

## (2) 20 引脚产品



注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116（内置运算放大器的产品）。

注意  $V_{SS}$  复用 A/D 转换器的接地电位。必须将  $V_{SS}$  连接至稳定的 GND (=0V)。

## 10.2 A/D 转换器的结构

A/D 转换器由以下硬件构成。

### (1) ANI0 ~ ANI7 引脚

这些是 A/D 转换器的 8 个通道的模拟输入引脚，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。

备注 A/D 转换器的模拟输入引脚因产品而不同。

- 16 引脚产品：ANI0 ~ ANI2、ANI5 ~ ANI7
- 20 引脚产品：ANI0 ~ ANI4

### (2) AMP0OUT 引脚（只限内置运算放大器的产品）

AMP0OUT 是运算放大器 0 的输出引脚。

AMP0OUT 复用 ANI1。A/D 转换器选择运算放大器 0 的输出信号作为模拟输入源来进行 A/D 转换。

### (3) AMP1OUT 引脚（只限 16 引脚产品中内置运算放大器的产品）

AMP1OUT 是运算放大器 1 的输出引脚。

AMP1OUT 复用 ANI6。A/D 转换器选择运算放大器 1 的输出信号作为模拟输入源来进行 A/D 转换。

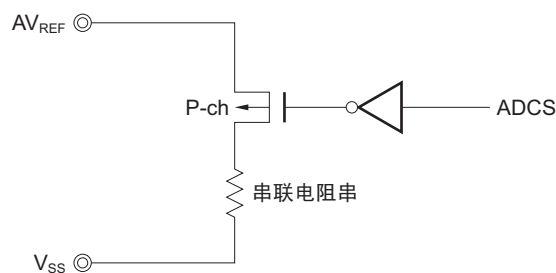
### (4) 采样 & 保持电路

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。

### (5) 串联电阻串

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

图 10-2 串联电阻串的电路结构



### (6) 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。

### (7) 逐次转换寄存器（SAR）

逐次转换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）。

### (8) 10 位 A/D 转换结果寄存器 (ADCR)

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 A/D 转换结果保存到高 10 位（低 6 位固定为“0”）。

### (9) 8 位 A/D 转换结果寄存器 (ADCRH)

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 26 章 等待的注意事项”。

### (10) 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。

### (11) $AV_{REF}$ 引脚

$AV_{REF}$  引脚是 A/D 转换器的模拟电源引脚 / 基准电压的输入引脚。将端口 2 用作数字端口时，必须设置与  $V_{DD}$  相同的电位。

根据  $AV_{REF}$  与  $V_{SS}$  之间所需的电压，将 ANI0 ~ ANI7 的输入信号转换成数字信号。

### (12) $V_{SS}$ 引脚

$V_{SS}$  引脚是接地电位引脚，并且复用 A/D 转换器的接地电位引脚。必须将  $V_{SS}$  连接至稳定的 GND (=0V)。

### 10.3 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)
- 10 位 A/D 转换结果寄存器 (ADCR)
- 8 位 A/D 转换结果寄存器 H (ADCRH)
- 模拟输入通道指定寄存器 (ADS)
- A/D 端口配置寄存器 (ADPC)
- 端口模式寄存器 2 (PM2)

#### (1) A/D 转换器模式寄存器 (ADM)

A/D 转换器模式寄存器设置 A/D 转换的模拟输入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。

在产生复位信号时，ADM 为“00H”。

图 10-3 A/D 转换器模式寄存器 (ADM) 的格式

地址：FF90H 复位时：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 10-2 A/D 转换时间的选择”。

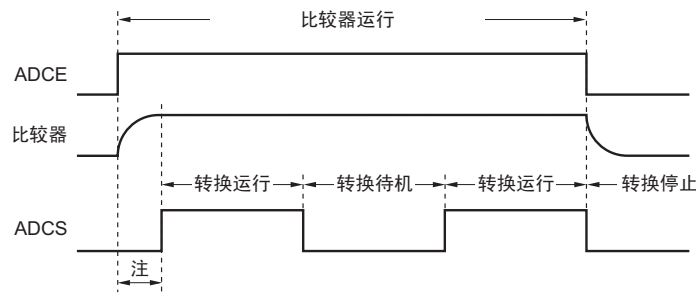
2. 通过 ADCS 和 ADCE 控制电压比较器的运行，从开始运行到运行稳定需要 1 $\mu$ s。因此，从将 ADCE 置“1”开始到至少经过 1 $\mu$ s 后，如果将 ADCS 置“1”，就从第 1 个转换数据开始有效。在未等到 1 $\mu$ s 就将 ADCS 置“1”时，忽略第 1 个转换数据。

表 10-1 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 10-4 使用比较器时的时序图



注 为了稳定内部电路，从将 ADCE 置“1”到将 ADCS 置“1”所需时间至少为 1 $\mu$ 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 26 章 等待的注意事项”。

表 10-2 A/D 转换时间的选择

(1)  $2.7V \leq AV_{REF} \leq 5.5V$  ( $LV0=0$ )

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	$f_{PRS}=2MHz$	$f_{PRS}=10MHz$		
0	0	0	0	0	$264/f_{PRS}$	禁止设置	26.4 $\mu$ s	$f_{PRS}/12$
0	0	1	0	0	$176/f_{PRS}$		17.6 $\mu$ s	$f_{PRS}/8$
0	1	0	0	0	$132/f_{PRS}$		13.2 $\mu$ s	$f_{PRS}/6$
0	1	1	0	0	$88/f_{PRS}$		8.8 $\mu$ s 注	$f_{PRS}/4$
1	0	0	0	0	$66/f_{PRS}$	33.0 $\mu$ s	6.6 $\mu$ s 注	$f_{PRS}/3$
1	0	1	0	0	$44/f_{PRS}$	22.0 $\mu$ s	禁止设置	$f_{PRS}/2$
上述以外					禁止设置			

注 只在  $4.0V \leq AV_{REF} \leq 5.5V$  时能设置。

(2)  $2.3V \leq AV_{REF} < 2.7V$  ( $LV0=1$ )

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	$f_{PRS}=2MHz$	$f_{PRS}=5MHz$		
0	0	0	0	1	$480/f_{PRS}$	禁止设置	禁止设置	$f_{PRS}/12$
0	0	1	0	1	$320/f_{PRS}$		64.0 $\mu$ s	$f_{PRS}/8$
0	1	0	0	1	$240/f_{PRS}$		48.0 $\mu$ s	$f_{PRS}/6$
0	1	1	0	1	$160/f_{PRS}$		32.0 $\mu$ s	$f_{PRS}/4$
1	0	0	0	1	$120/f_{PRS}$	60.0 $\mu$ s	禁止设置	$f_{PRS}/3$
1	0	1	0	1	$80/f_{PRS}$	40.0 $\mu$ 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。

- $4.0V \leq AV_{REF} \leq 5.5V$  时:  $f_{AD}=0.6V \sim 3.6MHz$
- $2.7V \leq AV_{REF} < 4.0V$  时:  $f_{AD}=0.6V \sim 1.8MHz$
- $2.3V \leq AV_{REF} < 2.7V$  时:  $f_{AD}=0.6V \sim 1.48MHz$

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 ( $ADCS=0$ )。

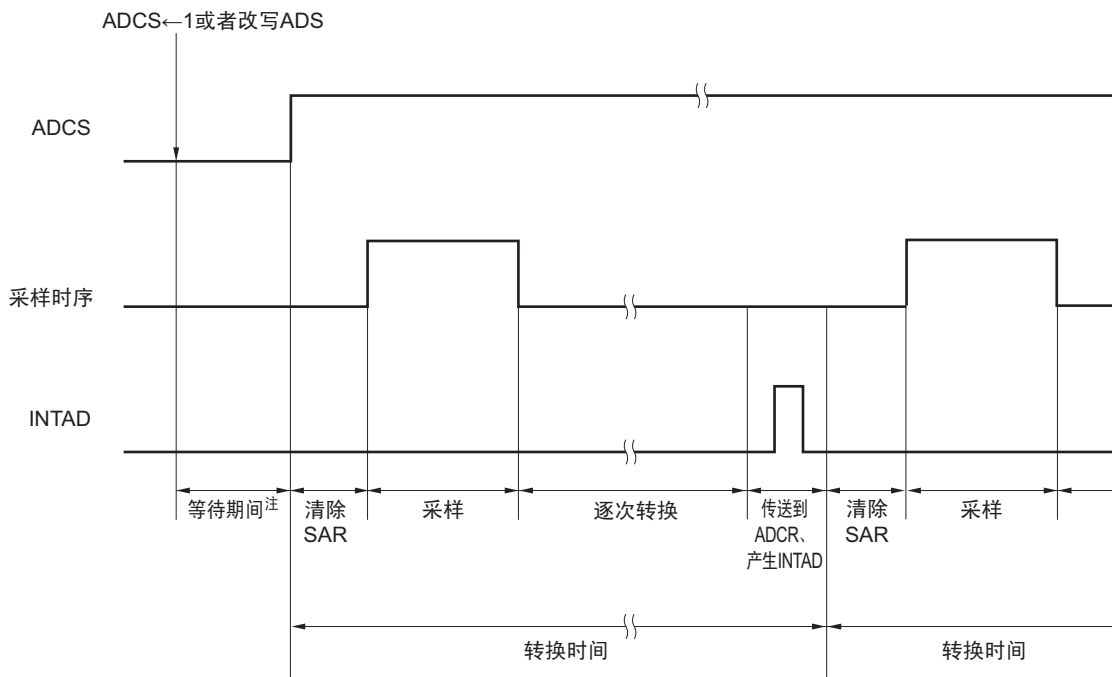
3. 在  $2.3 \leq AV_{REF} < 2.7V$  时，从默认值开始更改 LV0。

4. 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 10-5 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 26 章 等待的注意事项”。

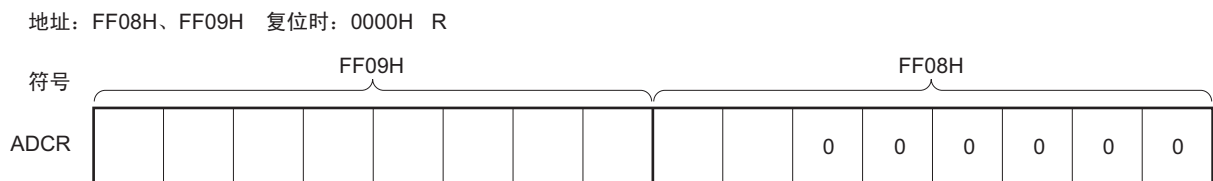
## (2) 10 位 A/D 转换结果寄存器 (ADCR)

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。

在产生复位信号时，ADCR 为“0000H”。

图 10-6 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 26 章 等待的注意事项”。

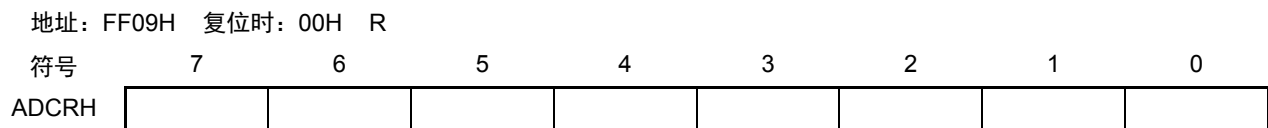
### (3) 8 位 A/D 转换结果寄存器 H (ADCRH)

8 位 A/D 转换结果寄存器 H 是保存 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。

在产生复位信号时，ADCRH 为“00H”。

图 10-7 8 位 A/D 转换结果寄存器 (ADCRH) 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，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 26 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。

在产生复位信号时，ADS 为“00H”。

图 10-8 模拟输入通道指定寄存器 (ADS) 的格式

## (a) 16 引脚产品

地址: FF0EH 复位时: 00H R/W

符号	7	6	5	4	3	<2>	<1>	<0>
ADS	0	0	0	0	0	ADS2	ADS1	ADS0

ADS2	ADS1	ADS0	模拟输入通道	输入源
0	0	0	ANI0	P20/ANI0 引脚
0	0	1	ANI1	P21/ANI1 引脚或者运算放大器 0 输出信号注
0	1	0	ANI2	P22/ANI2 引脚
1	0	1	ANI5	P25/ANI5 引脚
1	1	0	ANI6	P26/ANI6 引脚或者运算放大器 1 输出信号注
1	1	1	ANI7	P27/ANI7 引脚
上述以外			禁止设置	

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 (内置运算放大器的产品)。

注意 1. 必须将 bit3 ~ 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 26 章 等待的注意事项”。

## (b) 20 引脚产品

地址: FF0EH 复位时: 00H R/W

符号	7	6	5	4	3	<2>	<1>	<0>
ADS	0	0	0	0	0	ADS2	ADS1	ADS0

ADS2	ADS1	ADS0	模拟输入通道	输入源
0	0	0	ANI0	P20/ANI0 引脚
0	0	1	ANI1	P21/ANI1 引脚或者运算放大器 0 输出信号注
0	1	0	ANI2	P22/ANI2 引脚
0	1	1	ANI3	P23/ANI3 引脚
1	0	0	ANI4	P24/ANI4 引脚
上述以外			禁止设置	

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置运算放大器的产品)。

注意 1. 必须将 bit3 ~ 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 26 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 P20/AMP0-/ANI0 ~ P24/ANI4 (16 引脚产品时, P20/AMP0-/ANI0 ~ P22/ANI2/AMP0+、P25/ANI5/AMP1- ~ P27/ANI7/AMP1+) 切换为端口的数字输入 / 输出或者模拟输入 / 输出。ADPC 各位分别对应端口 2 的每个引脚, 并且可以 1 位单位进行指定。

通过 1 位或 8 位存储器操作指令设置 ADPC。

在产生复位信号时, ADPC 为“00H”。

图 10-9 A/D 端口配置寄存器 (ADPC) 的格式

地址: FF97H 复位时: 00H R/W

符号	7	6	5	4	3	2	1	0
ADPC	ADPC7	ADPC6	ADPC5	ADPC4注	ADPC3注	ADPC2	ADPC1	ADPC0

ADPCn	数字输入 / 输出或者模拟输入 / 输出的选择 (n=0 ~ 7)
0	模拟输入 / 输出
1	数字输入 / 输出

注 只限 20 引脚产品。

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 26 章 等待的注意事项”。

## (6) 端口模式寄存器 2 (PM2)

## (a) 16 引脚产品

将 ANI0/AMP0-/P20 ~ ANI2/AMP0+/P22、ANI5/AMP1-/P25 ~ ANI7/AMP1+/P27 引脚用作模拟输入端口时, 必须将 PM20 ~ PM22、PM25 ~ PM27 分别置“1”。此时, P20 ~ P22、P25 ~ P27 的输出锁存器即可是“0”也可是“1”。

如果将 PM20 ~ PM22、PM25 ~ PM27 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。

在产生复位信号时, PM2 为“FFH”。

图 10-10 端口模式寄存器 2 (PM2) 的格式 (16 引脚产品)

地址: FF22H 复位时: FFH R/W

符号	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	1	1	PM22	PM21	PM20

PM2n	选择 P2n 引脚的输入 / 输出模式 (n=0 ~ 2、5 ~ 7)
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

在使用 P20/AMP0-/ANI0 ~ P22/ANI2/AMP0+、P25/ANI5/AMP1- ~ P27/ANI7/AMP1+ 时，必须根据使用的引脚功能设置寄存器（参照表 10-3、表 10-4）。

表 10-3 P20/AMP0-/ANI0、P22/ANI2/AMP0+、P25/ANI5/AMP1-、P27/ANI7/AMP1+ 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMPmE 位 (m=0、1)	ADS 寄存器 (n=0、2、5、7)	P20/AMP0-/ANI0、 P22/ANI2/AMP0+、 P25/ANI5/AMP1-、 P27/ANI7/AMP1+ 引脚
选择模拟输入	输入模式	0	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)
		1	选择 ANIn	模拟输入 (A/D 转换对象)、 运算放大器 m 输入
			不选择 ANIn	运算放大器 m 输入
输出模式	—	—	禁止设置	
选择数字输入 / 输出	输入模式	—	选择 ANIn	禁止设置
			不选择 ANIn	数字输入
	输出模式	—	选择 ANIn	禁止设置
			不选择 ANIn	数字输出

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMPmE: 运算放大器控制寄存器 (AMPm) 的 bit3、bit7  
 ADS: 模拟输入通道指定寄存器

表 10-4 P21/ANI1/AMP0OUT、P26/ANI6/AMP1OUT 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMPmE 位 (m=0、1)	ADS 寄存器 (n=1、6)	P21/ANI1/AMP0OUT、 P26/ANI6/AMP1OUT 引脚
选择模拟输入 / 输出	输入模式	0	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)
		1	选择 ANIn	运算放大器 m 输出 (A/D 转换对象)
			不选择 ANIn	运算放大器 m 输出 (非 A/D 转换对象)
输出模式	—	—	禁止设置	
选择数字输入 / 输出	输入模式	0	选择 ANIn	禁止设置
			不选择 ANIn	数字输入
		1	—	禁止设置
	输出模式	0	选择 ANIn	禁止设置
			不选择 ANIn	数字输出
		1	—	禁止设置

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMPmE: 运算放大器控制寄存器 (AMPm) 的 bit3、bit7  
 ADS: 模拟输入通道指定寄存器

## (b) 20 引脚产品

将 ANI0/AMP0-/P20 ~ ANI4/P24 引脚用作模拟输入端口时，必须将 PM20 ~ PM24 分别置“1”。此时，P20 ~ P24 的输出锁存器即可是“0”也可能是“1”。

如果将 PM20 ~ PM24 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。

在产生复位信号时，PM2 为“FFH”。

图 10-11 端口模式寄存器 2 (PM2) 的格式 (20 引脚产品)

地址: FF22H 复位时: FFH R/W

符号	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

PM2n	选择 P2n 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)

在使用 P20/AMP0-/ANI0 ~ P24/ANI4 时，必须根据使用的引脚功能设置寄存器 (参照表 10-5 ~ 表 10-7)。

表 10-5 P20/ANI0/AMP0-、P22/ANI2/AMP0+ 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMP0E 位	ADS 寄存器 (n=0、2)	P20/ANI0/AMP0-、 P22/ANI2/AMP0+ 引脚
选择模拟输入	输入模式	0	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)
		1	选择 ANIn	模拟输入 (A/D 转换对象)、 运算放大器 0 输入
			不选择 ANIn	运算放大器 0 输入
	输出模式	—	—	禁止设置
选择数字输入 / 输出	输入模式	—	选择 ANIn	禁止设置
			不选择 ANIn	数字输入
	输出模式	—	选择 ANIn	禁止设置
			不选择 ANIn	数字输出

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMP0E: 运算放大器控制寄存器 (AMP0) 的 bit7  
 ADS: 模拟输入通道指定寄存器

表 10-6 P21/ANI1/AMP0OUT 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMP0E 位	ADS 寄存器	P21/ANI1/AMP0OUT 引脚
选择模拟输入 / 输出	输入模式	0	选择 ANI1	模拟输入 (A/D 转换对象)
			不选择 ANI1	模拟输入 (非 A/D 转换对象)
		1	选择 ANI1	运算放大器 0 输出 (A/D 转换对象)
			不选择 ANI1	运算放大器 0 输出 (非 A/D 转换对象)
输出模式	—	—	禁止设置	
选择数字输入 / 输出	输入模式	0	选择 ANI1	禁止设置
			不选择 ANI1	数字输入
		1	—	禁止设置
	输出模式	0	选择 ANI1	禁止设置
			不选择 ANI1	数字输出
		1	—	禁止设置

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMP0E: 运算放大器控制寄存器 (AMPM) 的 bit7  
 ADS: 模拟输入通道指定寄存器

表 10-7 P23/ANI3、P24/ANI4 引脚功能的设置

ADPC 寄存器	PM2 寄存器	ADS 寄存器 (n=3、4)	P23/ANI3、P24/ANI4 引脚
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	—	禁止设置
选择数字输入 / 输出	输入模式	选择 ANIn	禁止设置
		不选择 ANIn	数字输入
	输出模式	选择 ANIn	禁止设置
		不选择 ANIn	数字输出

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 ADS: 模拟输入通道指定寄存器

## 10.4 A/D 转换器的运行

### 10.4.1 A/D 转换器的基本运行

1. 通过 A/D 转换器模式寄存器 (ADM) 的 bit5 ~ bit1 (FR2 ~ FR0、LV1、LV0) 设置 A/D 转换时间和运行模式。
2. 将 ADM 的 bit0 (ADCE) 置“1”，开始电压比较器的运行。
3. 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/输出，并且通过端口模式寄存器 2 (PM2) 设置为输入模式。
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 13 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. 设置逐次转换寄存器 (SAR) 的 bit9 后，串联电阻串输出  $(1/2)AV_{REF}$  电压。
9. 通过电压比较器比较串联电阻串的输出电压和采样电压之间的电压差。如果模拟输入电压大于  $(1/2)AV_{REF}$ ，SAR 的 MSB 保持设置；如果模拟输入电压小于  $(1/2)AV_{REF}$ ，SAR 的 MSB 发生复位。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输出电压。
  - bit9=1:  $(3/4)AV_{REF}$
  - bit9=0:  $(1/4)AV_{REF}$
 比较该串联电阻串的输出电压和采样电压，根据其结果如下设置 SAR 的 bit8。
  - 采样电压  $\geq$  串联电阻串的输出电压: bit8=1
  - 采样电压  $<$  串联电阻串的输出电压: bit8=0
10. 持续进行上述比较，直到比较到 SAR 的 bit0。
11. 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)，然后进行锁存。  
同时，还可产生 A/D 转换结束中断请求 (INTAD)。
12. 重复步骤 6 ~ 步骤 12 的运行，直到 ADCS 变为“0”。

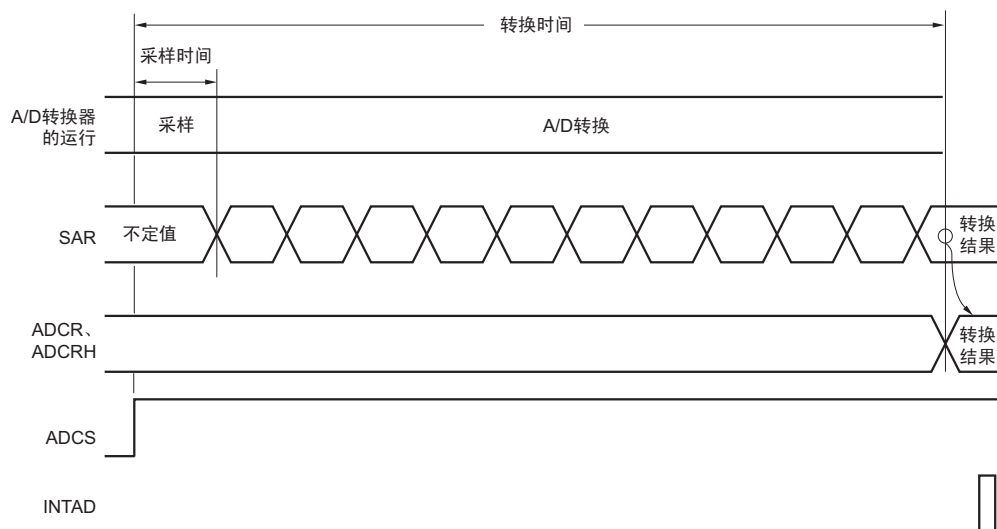
如果要停止 A/D 转换器，必须将 ADCS 置“0”。

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 1. 在步骤 2 到步骤 5 之间至少空出  $1\mu\text{s}$ 。
2. 只要是在步骤 4 之前，可在任何时序执行步骤 2。



图 10-12 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。

在产生复位信号时，A/D 转换结果寄存器（ADCR、ADCRH）为“0000H”或者“00H”。

#### 10.4.2 输入电压和转换结果

输入到模拟输入引脚（ANI0 ~ ANI7）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。

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

或者

$$(\text{ADCR} - 0.5) \times \frac{V_{\text{REF}}}{1024} \leq V_{\text{AIN}} < (\text{ADCR} + 0.5) \times \frac{V_{\text{REF}}}{1024}$$

INT(): 该函数返回 () 中的值的整数部分

$V_{\text{AIN}}$ : 模拟输入电压

$V_{\text{REF}}$ :  $V_{\text{REF}}$  引脚电压

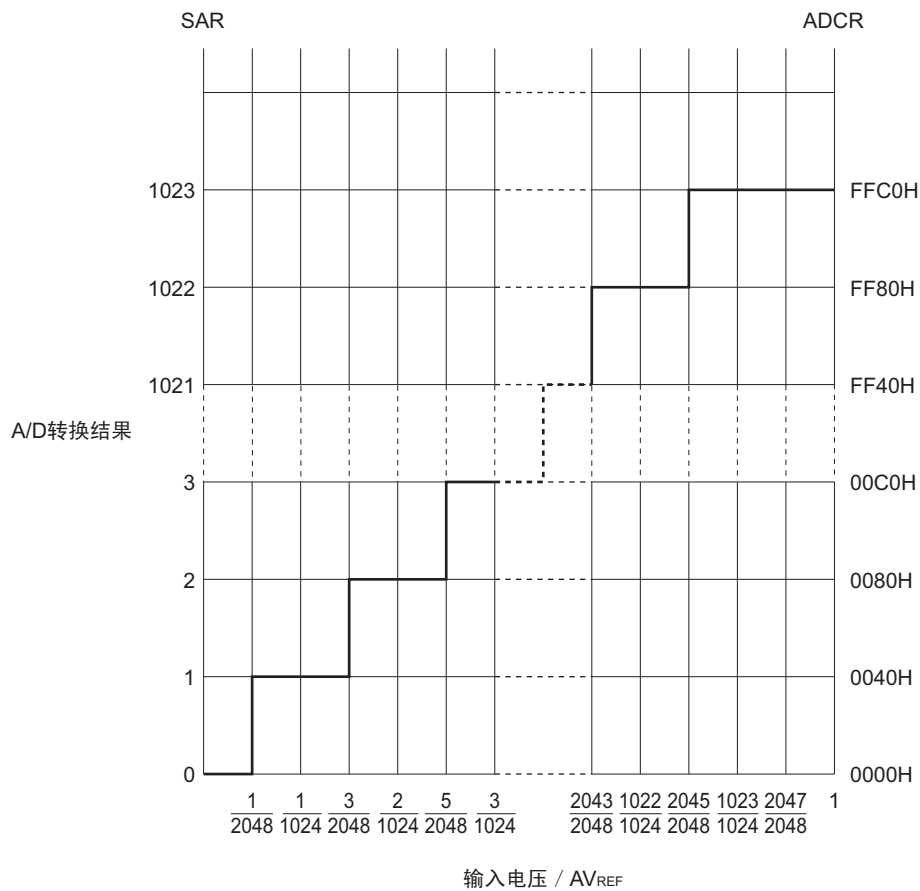
ADCR: 10 位 A/D 转换结果寄存器（ADCR）的值

备注 A/D 转换器的模拟输入引脚因产品而不同。

- 16 引脚产品：ANI0 ~ ANI2、ANI5 ~ ANI7
- 20 引脚产品：ANI0 ~ ANI4

模拟输入电压和 A/D 转换结果的关系如图 10-13 所示。

图 10-13 模拟输入电压和 A/D 转换结果之间的关系



### 10.4.3 A/D 转换器的运行模式

通过模拟输入通道指定寄存器 (ADS)，从 ANI0 ~ ANI7 中选择 1 个通道的模拟输入进行 A/D 转换。

备注 A/D 转换器的模拟输入引脚因产品而不同。

- 16 引脚产品：ANI0 ~ ANI2、ANI5 ~ ANI7
- 20 引脚产品：ANI0 ~ ANI4

#### (1) A/D 转换运行

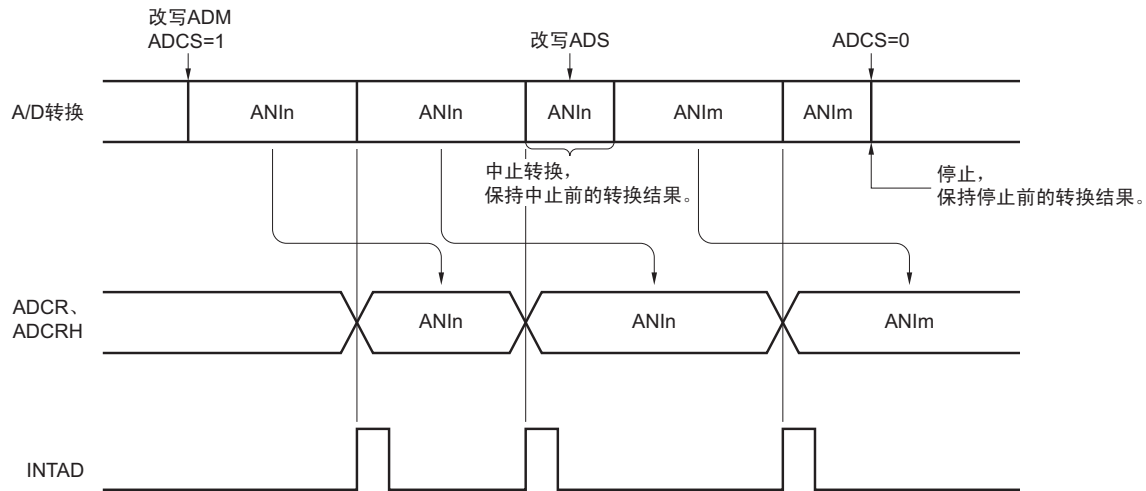
通过将 A/D 转换器模式寄存器 (ADM) 的 bit7 (ADCS) 置“1”，开始模拟输入通道指定寄存器 (ADS) 指定的模拟输入引脚外加电压的 A/D 转换运行。

一旦 A/D 转换运行结束，就将转换结果保存到 A/D 转换结果寄存器 (ADCR、ADCRH)，并且产生中断请求信号 (INTAD)。如果 1 个 A/D 转换结束，就立即开始下一个 A/D 转换运行。

如果在 A/D 转换运行期间改写 ADS，就中止当前正在执行中的 A/D 转换，并且重新从头开始 A/D 转换运行。

如果在 A/D 转换期间给 ADCS 写“0”，就立即停止 A/D 转换运行，并且保持停止之前的转换结果。

图 10-14 A/D 转换运行



- 备注 1. n=0 ~ 7 (因产品而不同)  
2. m=0 ~ 7 (因产品而不同)

设置方法如下所述。

1. 通过 A/D 转换器模式寄存器 (ADM) 的 bit5 ~ bit1 (FR2 ~ FR0、LV1、LV0) 选择 A/D 转换时间和运行模式。
2. 将 ADM 的 bit0 (ADCE) 置“1”。
3. 通过 A/D 端口配置寄存器 (ADPC) 和端口模式寄存器 2 (PM2) 将使用的通道设置为模拟输入。
4. 通过模拟输入通道指定寄存器 (ADS) 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. 将中断屏蔽标志寄存器 1L (MK1L) 的 bit0 (ADMK) 置“1”注。
9. 通过 ADS 更改通道，开始 A/D 转换。
10. 将中断请求标志寄存器 1L (IF1L) 的 bit0 (ADIF) 清“0”。
11. 将 ADMK 清“0”注。
12. 一旦 1 次 A/D 转换结束，就产生中断请求信号 (INTAD)。
13. 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>

14. 将 ADCS 清“0”。
15. 将 ADCE 清“0”。

注 只在 A/D 转换中使用中断处理时，执行此步骤。

注意 1. 步骤 2 到步骤 5 之间至少空出 1 $\mu$ s。

2. 只要是在步骤 4 之前，可在任何时序执行步骤 2。
3. 可省略步骤 2。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 12 的时间与 ADM 的 bit5 ~ bit1 (FR2 ~ FR0、LV1、LV0) 设置的转换时间不同。步骤 9 到步骤 12 的时间为 FR2 ~ FR0、LV1、LV0 设置的转换时间。

## 10.5 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 10-15 总误差

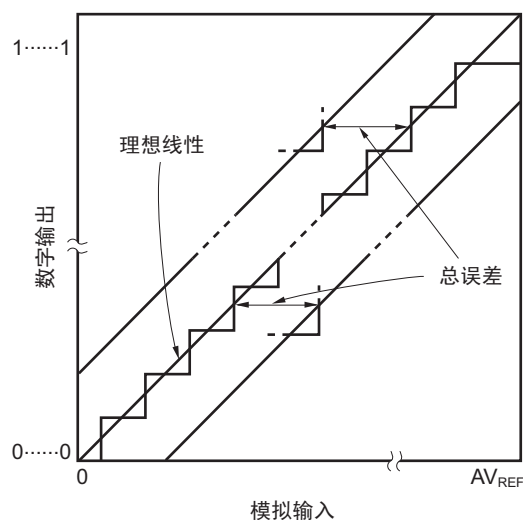
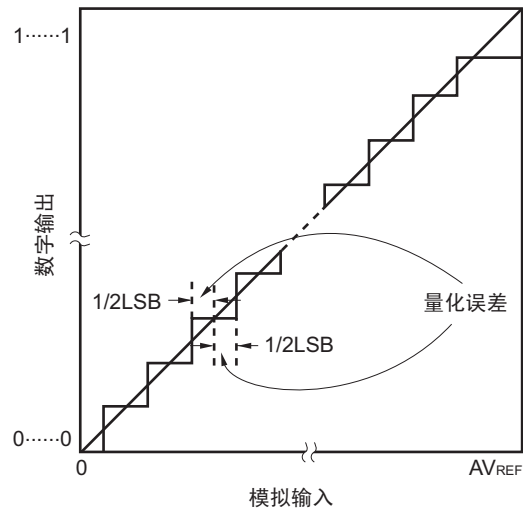


图 10-16 量化误差



#### (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  $1\text{LSB}$  时的代码输出宽度的实际测量值与理想值之差。

图 10-17 零刻度误差

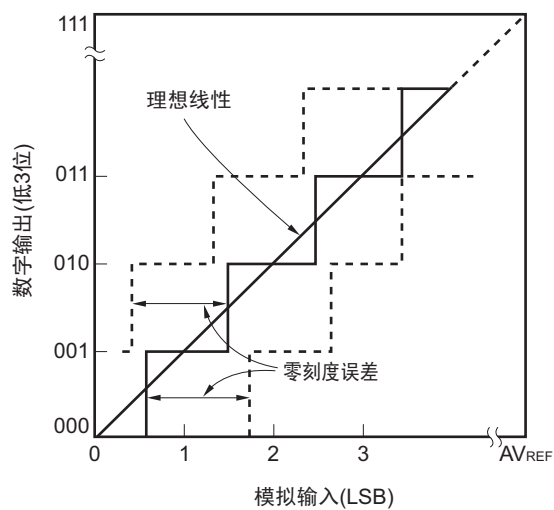


图 10-18 满刻度误差

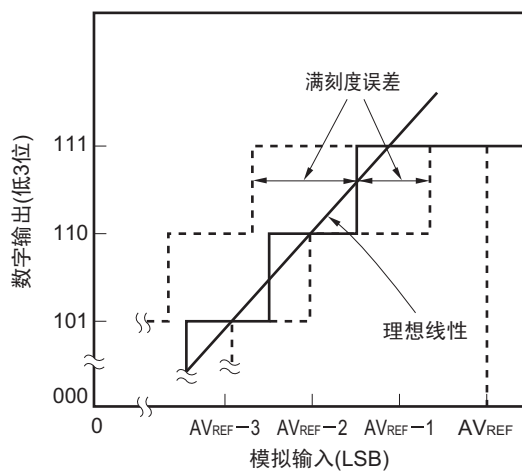


图 10-19 积分线性误差

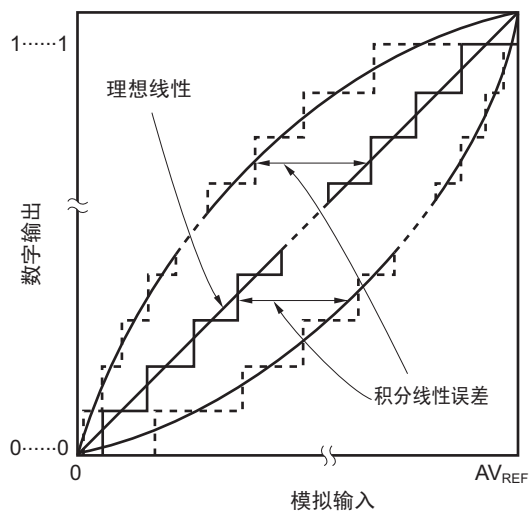
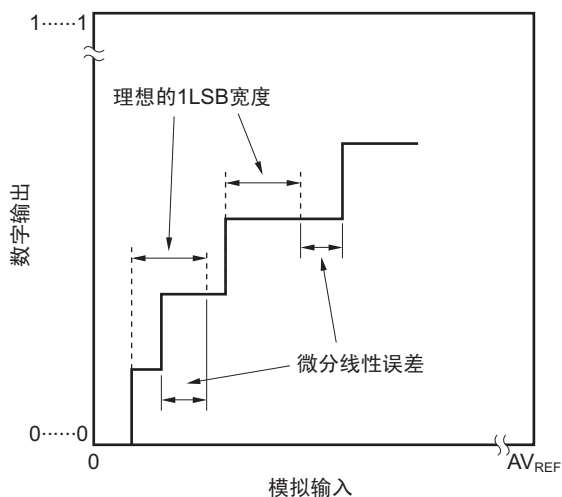


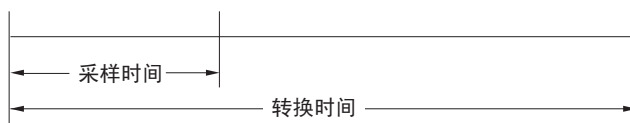
图 10-20 微分线性误差

**(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。



## 10.6 A/D 转换器的注意事项

### (1) STOP 模式中的工作电流

为了满足 STOP 模式中的电源电流的 DC 特性，在执行 STOP 指令前，必须将 A/D 转换器模式寄存器 (ADM) 的 bit7 (ADCS) 和 bit0 (ADCE) 置“0”。

如果要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 ~ ANI7 的输入范围

在规格范围内使用 ANI0 ~ ANI7 的输入电压。尤其是一旦输入大于  $AV_{REF}$  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

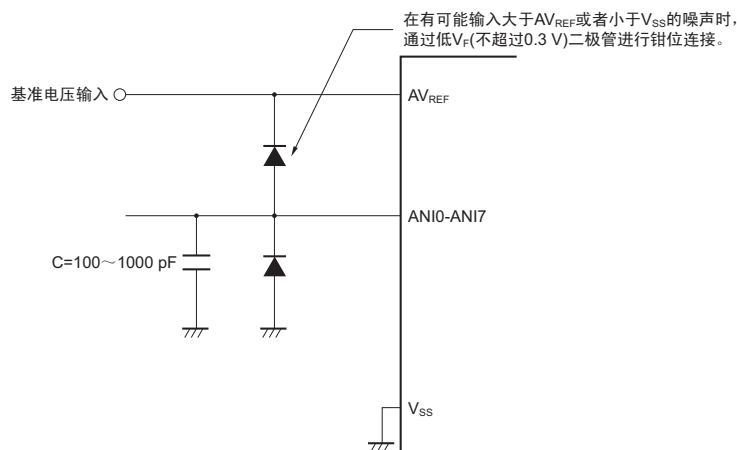
为了保持 10 位分辨率，必须注意影响  $AV_{REF}$ 、ANI0 ~ ANI7 引脚的噪声。

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 10-21 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。

备注 A/D 转换器的模拟输入引脚因产品而不同。

- 16 引脚产品：ANI0 ~ ANI2、ANI5 ~ ANI7
- 20 引脚产品：ANI0 ~ ANI4

图 10-21 模拟输入引脚的处理





### (5) ANI0/P20 ~ ANI7/P27

1. 模拟输入引脚（ANI0~ANI7）复用数字输入/输出端口（P20~P27）引脚。选择 ANI0~ANI7 中的任意一个进行 A/D 转换时，转换过程期间不能存取 P20~P27，否则，可能会降低转换分辨率。
2. 将 ANI0/P20~ANI7/P27 用作数字输入/输出端口时，推荐从离  $AV_{REF}$  最远的引脚（16 引脚产品：ANI7/P27 引脚、20 引脚产品：ANI4/P24 引脚）开始执行；用作模拟输入时，推荐从离  $V_{SS}$  最近的引脚（16 引脚产品：ANI7/P27 引脚、20 引脚产品：ANI4/P24 引脚）开始执行。
3. 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 ~ ANI7 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泄露电流流过，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 ~ ANI7 引脚（参照图 10-21）。

备注 A/D 转换器的模拟输入引脚因产品而不同。

- 16 引脚产品：ANI0~ANI2、ANI5~ANI7
- 20 引脚产品：ANI0~ANI4

### (7) $AV_{REF}$ 引脚的输入抗租

在  $AV_{REF}$  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  $AV_{REF}$  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。

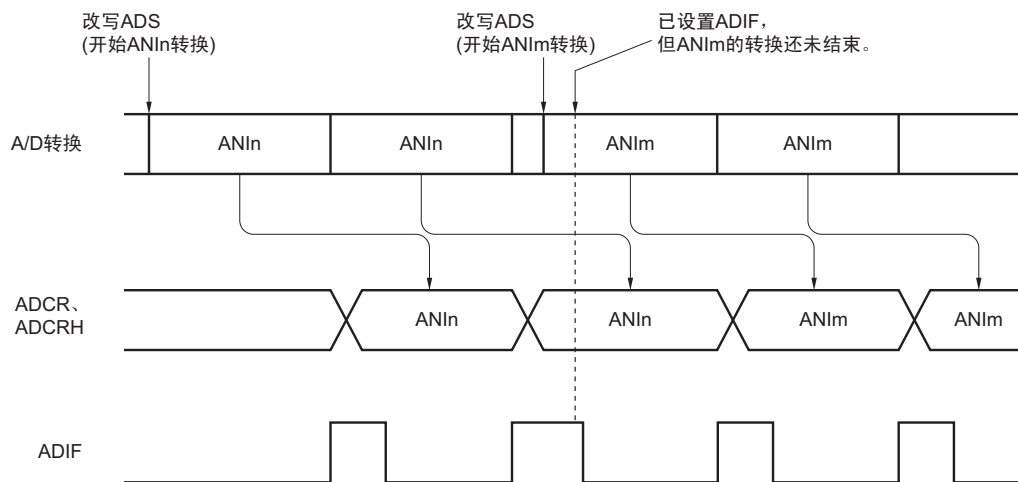
### (8) 中断请求标志（ADIF）

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 转换后又重新开始时，必须在重新开始前先将 ADIF 清“0”。

图 10-22 A/D 转换结束中断请求的发生时序



- 备注 1.  $n=0 \sim 7$  (因产品而不同)  
 2.  $m=0 \sim 7$  (因产品而不同)

### (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 10-23 ANIn 引脚的内部等效电路

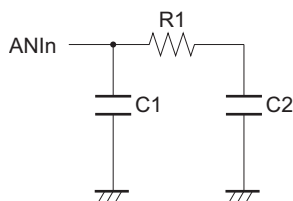


表 10-8 等效电路的各电阻和电容值 (参考值)

$AV_{REF}$	R1	C1	C2
$2.3\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$	11.5 k $\Omega$	8.0 pF	8.0 pF

- 备注 1. 表 10-8 中的各电阻和电容值并非保证值。  
 2.  $n=0 \sim 7$  (因产品而不同)

## 第 11 章 运算放大器

### 11.1 运算放大器的功能

在 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x 中装有运算放大器 0 和运算放大器 1。

运算放大器有 2 个输入引脚（AMPn- 引脚和 AMPn+ 引脚）和 1 个输出引脚（AMPnOUT 引脚），可用作与外部连接的单电源放大器。

AMP0OUT 引脚复用 A/D 转换器的模拟输入引脚（R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 时，AMP1OUT 引脚复用 A/D 转换器的模拟输入引脚），因此，放大后的电压可用作 A/D 转换器的模拟输入。

备注 n=0、1

### 11.2 运算放大器的结构

运算放大器由以下硬件构成。

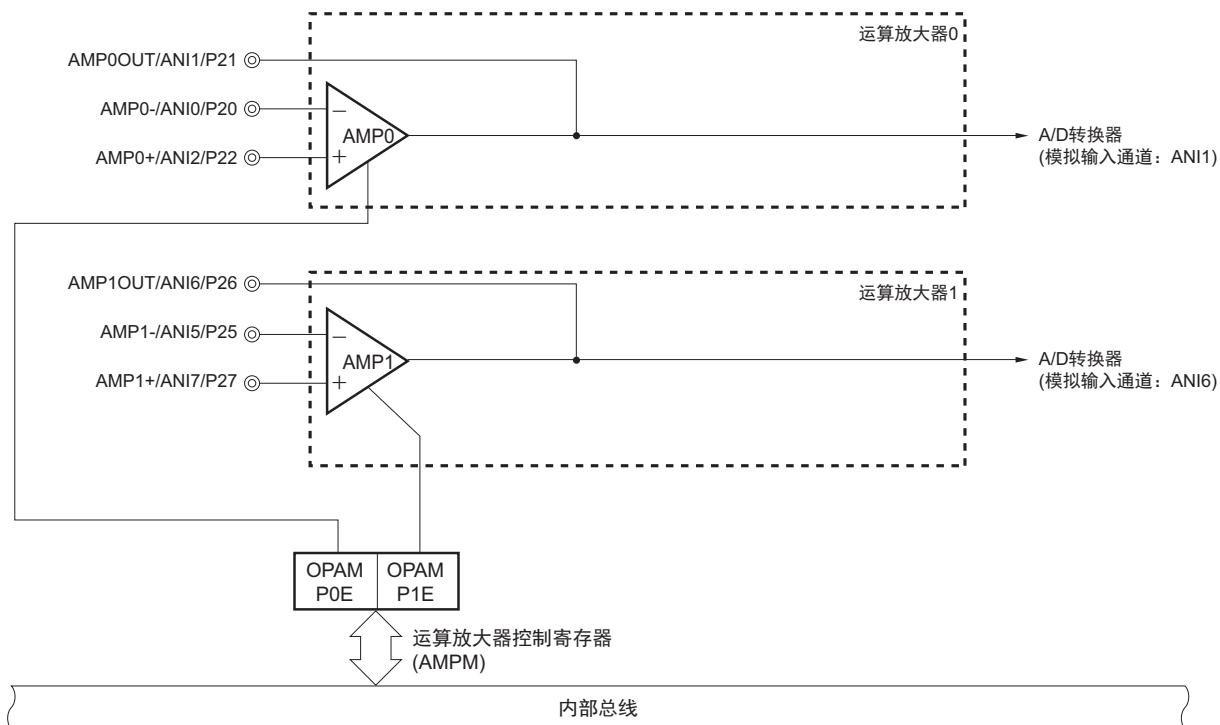
表 11-1 运算放大器的结构

项目	结构
运算放大器输入	AMPn- 引脚、AMPn+ 引脚
运算放大器输出	AMPnOUT 引脚
控制寄存器	运算放大器控制寄存器（AMPM） 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）

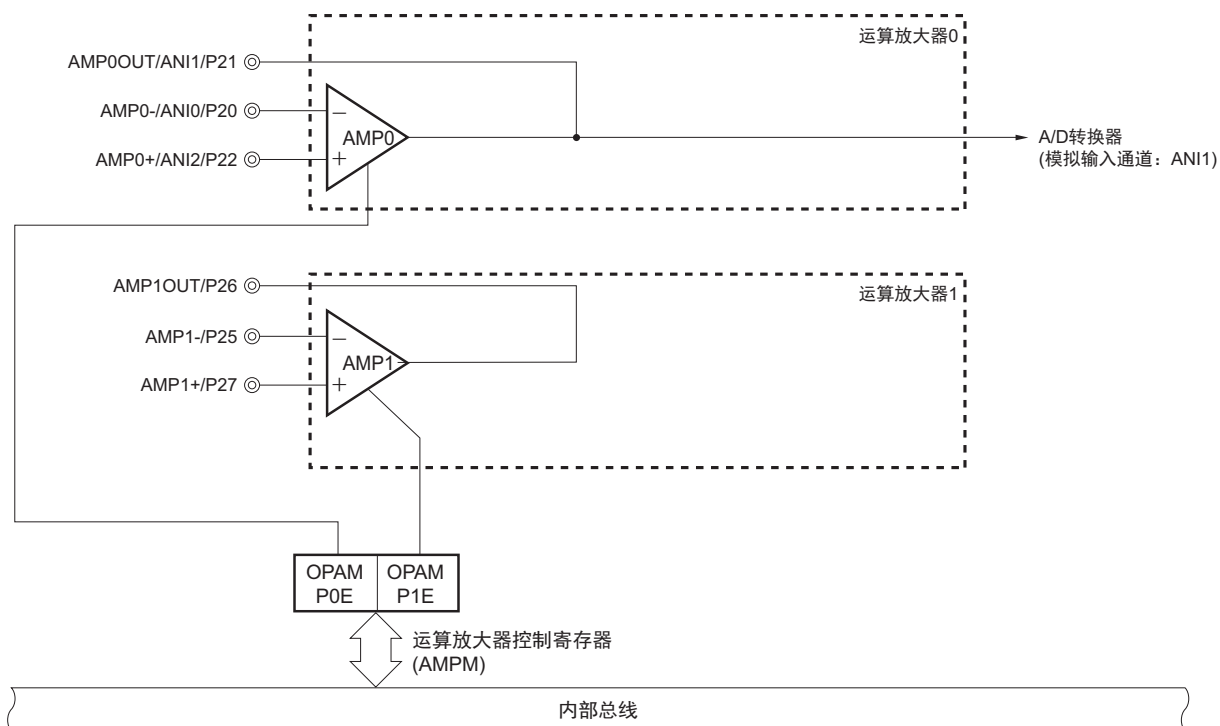
备注 n=0、1

图 11-1 运算放大器的框图

(1) R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114



(2) R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116



### 11.3 运算放大器使用的寄存器

运算放大器使用以下 4 种寄存器。

- 运算放大器控制寄存器 (AMPM)
- 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)

#### (1) 运算放大器控制寄存器 (AMPM)

运算放大器控制寄存器控制运算放大器 0 和 1 的运行。

通过 1 位或 8 位存储器操作指令设置 AMPM。

在产生复位信号时，AMPM 为“00H”。

图 11-2 运算放大器控制寄存器 (AMPM) 的格式

地址: FF60H 复位时: 00H R/W

符号	<7>	6	5	4	<3>	2	1	0
AMPM	OPAMP0E	0	0	0	OPAMP1E	0	0	0

OPAMP0E	控制运算放大器 0 的运行
0	停止运算放大器 0 的运行
1	允许运算放大器 0 的运行

OPAMP1E	控制运算放大器 1 的运行
0	停止运算放大器 1 的运行
1	允许运算放大器 1 的运行

**注意** 使用运算放大器时，如果将不用于运算放大器的端口 2 的引脚用作数字输入，为了防止 A/D 转换分辨率降低，必须固定数字输入端口的输入电平。

## (2) A/D 端口配置寄存器 (ADPC)

A/D 端口配置寄存器将 P20/AMP0-/ANI0 ~ P24/AMP4 (R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 时, P20/AMP0-/ANI0 ~ P22/ANI2/AMP0+、P25/ANI5/AMP1- ~ P27/ANI7/AMP1+) 切换到端口的数字输入 / 输出或者模拟输入 / 输出。ADPC 的各位分别对应端口 2 的每个引脚, 并且可以 1 位单位进行指定。

通过 1 位或 8 位存储器操作指令设置 ADPC。

在产生复位信号时, ADPC 为“00H”。

图 11-3 A/D 端口配置寄存器 (ADPC) 的格式

地址: FF97H 复位时: 00H R/W

符号	7	6	5	4	3	2	1	0
ADPC	ADPC7	ADPC6	ADPC5	ADPC4注	ADPC3注	ADPC2	ADPC1	ADPC0
	数字输入 / 输出或者模拟输入 / 输出的选择 (n=0 ~ 7)							
	0	模拟输入 / 输出						
	1	数字输入 / 输出						

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116。

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。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 26 章 等待的注意事项”。

## (3) 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。

可按 1 位或 8 位存储器操作指令设置 ADS。

在产生复位信号时，ADS 为“00H”。

图 11-4 模拟输入通道指定寄存器 (ADS) 的格式

## (a) R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114

地址：FF0EH 复位时：00H R/W

符号	7	6	5	4	3	<2>	<1>	<0>
ADS	0	0	0	0	0	ADS2	ADS1	ADS0

ADS2	ADS1	ADS0	模拟输入通道	输入源
0	0	0	ANI0	P20/ANI0 引脚
0	0	1	ANI1	P21/ANI1 引脚或运算放大器 0 输出信号
0	1	0	ANI2	P22/ANI2 引脚
1	0	1	ANI5	P25/ANI5 引脚
1	1	0	ANI6	P26/ANI6 引脚或运算放大器 1 输出信号
1	1	1	ANI7	P27/ANI7 引脚
上述以外			禁止设置	

## (b) R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116

地址：FF0EH 复位时：00H R/W

符号	7	6	5	4	3	<2>	<1>	<0>
ADS	0	0	0	0	0	ADS2	ADS1	ADS0

ADS2	ADS1	ADS0	模拟输入通道	输入源
0	0	0	ANI0	P20/ANI0 引脚
0	0	1	ANI1	P21/ANI1 引脚或运算放大器 0 输出信号
0	1	0	ANI2	P22/ANI2 引脚
0	1	1	ANI3	P23/ANI3 引脚
1	0	0	ANI4	P24/ANI4 引脚
上述以外			禁止设置	

注意 1. 必须将 bit3 ~ bit7 置“0”。

2. 通过端口模式寄存器 2 (PM2)，将 A/D 转换使用的通道设置为输入模式。

## (4) 端口模式寄存器 2 (PM2)

## (a) R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114

运算放大器 0 使用 AMP0-/ANI0/P20、AMP0OUT/ANI1/P21 和 AMP0+/ANI2/P22 引脚时，将 PM20 ~ PM22 分别置“1”。

运算放大器 1 使用 AMP1-/ANI5/P25、AMP1OUT/ANI6/P26 和 AMP1+/ANI7/P27 的引脚时，将 PM25 ~ PM27 分别置“1”。

此时，P20 ~ P22 和 P25 ~ P27 的输出锁存器即可是“0”也可能是“1”。

如果将 PM20 ~ PM22 和 PM25 ~ PM27 分别置“0”，就不能用作运算放大器 0 和运算放大器 1 的引脚。

通过 1 位或 8 位存储器操作指令设置 PM2。

在产生复位信号时，PM2 为“FFH”。

图 11-5 端口模式寄存器 2 (PM2) 的格式

地址: FF22H 复位时: FFH R/W

符号	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	1	1	PM22	PM21	PM20

PM2n	选择 P2n 引脚的输入 / 输出模式 (n=0 ~ 2、5 ~ 7)
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

使用 P20/ANI0/AMP0- ~ P22/ANI2/AMP0+、P25/ANI5/AMP1- ~ P27/ANI7/AMP0+ 时，根据使用的引脚功能设置寄存器。(参照表 11-2、表 11-3)。

表 11-2 P20/ANI0/AMP0-、P22/ANI2/AMP0+、P25/ANI5/AMP1-、P27/ANI7/AMP0+ 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMPmE 位 (m=0、2)	ADS 寄存器 (n=0、2、5、7)	P20/ANI0/AMP0-、 P22/ANI2/AMP0+、 P25/ANI5/AMP1-、 P27/ANI7/AMP0+ 引脚
选择模拟输入	输入模式	0	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)
		1	选择 ANIn	模拟输入 (A/D 转换对象)、 运算放大器 m 输入
			不选择 ANIn	运算放大器 m 输入
	输出模式	—	—	禁止设置
选择数字输入 / 输出	输入模式	—	选择 ANIn	禁止设置
			不选择 ANIn	数字输入
	输出模式	—	选择 ANIn	禁止设置
			不选择 ANIn	数字输出



表 11-3 P21/ANI1/AMP0OUT、P26/ANI6/AMP1OUT 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMPmE 位 (m=0、1)	ADS 寄存器 (n=1、6)	P21/ANI1/AMP0OUT、 P26/ANI6/AMP1OUT 引脚
选择模拟输入 / 输出	输入模式	0	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)
		1	选择 ANIn	运算放大器 m 输出 (A/D 转换对象)
			不选择 ANIn	运算放大器 m 输出 (非 A/D 转换对象)
输出模式	—	—	禁止设置	
选择数字输入 / 输出	输入模式	0	选择 ANIn	禁止设置
			不选择 ANIn	数字输入
		1	—	禁止设置
	输出模式	0	选择 ANIn	禁止设置
			不选择 ANIn	数字输出
		1	—	禁止设置

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMPmE: 运算放大器控制寄存器 (AMPm) 的 bit7 和 bit3  
 ADS: 模拟输入通道指定寄存器

## (b) R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116

运算放大器 0 使用 AMP0-/ANI0/P20、AMP0OUT/ANI1/P21 和 AMP0+/ANI2/P22 引脚时，将 PM20 ~ PM22 分别置“1”。

运算放大器 1 使用 AMP1-/P25、AMP1OUT/P26 和 AMP1+/P27 引脚时，将 PM25 ~ PM27 分别置“1”。此时，P20 ~ P22 和 P25 ~ P27 的输出锁存器即可是“0”也可是“1”。

如果将 PM20 ~ PM22 和 PM25 ~ PM27 分别置“0”，就不能用作运算放大器 0 和运算放大器 1 的引脚。通过 1 位或 8 位存储器操作指令设置 PM2。

在产生复位信号时，PM2 为“FFH”。

图 11-6 端口模式寄存器 2 (PM2) 的格式

地址: FF22H 复位时: FFH R/W

符号	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

PM2n	选择 P2n 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)

使用 P20/ANI0/AMP0-、P21/ANI1/AMP0OUT、P22/ANI2/AMP0+、P25/AMP1-、P26/AMP1OUT、P27/AMP1+ 时，根据使用的引脚功能设置寄存器。(参照表 11-4 ~ 表 11-7)。

表 11-4 P20/ANI0/AMP0- 和 P22/ANI2/AMP0+ 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMP0E 位	ADS 寄存器 (n=0、2)	P20/ANI0/AMP0-、 P22/ANI2/AMP0+ 引脚
选择模拟输入	输入模式	0	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)
		1	选择 ANIn	模拟输入 (A/D 转换对象)、 运算放大器 0 输入
			不选择 ANIn	运算放大器 0 输入
输出模式	—	—	禁止设置	
选择数字输入 / 输出	输入模式	—	选择 ANIn	禁止设置
			不选择 ANIn	数字输入
	输出模式	—	选择 ANIn	禁止设置
			不选择 ANIn	数字输出

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMP0E: 运算放大器控制寄存器 (AMP0) 的 bit7  
 ADS: 模拟输入通道指定寄存器

表 11-5 P21/ANI1/AMP0OUT 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMP0E 位	ADS 寄存器	P21/ANI1/AMP0OUT 引脚
选择模拟输入 / 输出	输入模式	0	选择 ANI1	模拟输入 (A/D 转换对象)
			不选择 ANI1	模拟输入 (非 A/D 转换对象)
		1	选择 ANI1	运算放大器 0 输出 (A/D 转换对象)
			不选择 ANI1	运算放大器 0 输出 (非 A/D 转换对象)
输出模式	—	—	禁止设置	
选择数字输入 / 输出	输入模式	0	选择 ANI1	禁止设置
			不选择 ANI1	数字输入
		1	—	禁止设置
	输出模式	0	选择 ANI1	禁止设置
			不选择 ANI1	数字输出
		1	—	禁止设置

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMP0E: 运算放大器控制寄存器 (AMP0) 的 bit7  
 ADS: 模拟输入通道指定寄存器

表 11-6 P25/AMP1-、P27/AMP1+ 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMP1E 位	P25/AMP1-、P27/AMP1+ 引脚
选择模拟输入	输入模式	0	模拟输入 (A/D 转换对象)
		1	运算放大器 1 输出
	输出模式	—	禁止设置
选择数字输入 / 输出	输入模式	—	数字输入
	输出模式	—	数字输出

表 11-7 P26/AMP1OUT 引脚功能的设置

ADPC 寄存器	PM2 寄存器	OPAMP1E 位	P26/AMP1OUT 引脚
选择模拟输入 / 输出	输入模式	0	模拟输入 (非 A/D 转换对象)
		1	运算放大器 1 输出
	输出模式	—	禁止设置
选择数字输入 / 输出	输入模式	0	数字输入
		1	禁止设置
	输出模式	0	数字输出
		1	禁止设置

备注 ADPC: A/D 端口配置寄存器  
 PM2: 端口模式寄存器 2  
 OPAMP1E: 运算放大器控制寄存器 (AMPn) 的 bit3

## 11.4 运算放大器的运行

运算放大器有 2 个输入引脚 (AMPn- 引脚和 AMPn+ 引脚) 和 1 个输出引脚 (AMPnOUT 引脚), 可用作与外部连接的单电源放大器。

AMP0OUT 引脚复用 A/D 转换器的模拟输入引脚 (R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 时, AMP1OUT 引脚复用 A/D 转换器的模拟输入引脚), 因此, 放大后的电压可用作 A/D 转换器的模拟输入。

开始运行的步骤如下所示。

1. 通过 ADPC 寄存器将使用的引脚 (AMPn-、AMPn+、AMPnOUT) 设置为模拟输入/输出。
2. 通过 PM2 寄存器将使用的引脚 (AMPn-、AMPn+、AMPnOUT) 设置为输入模式。
3. 将 OPAMPnE 位置“1”, 允许运行。

注意 将放大后的电压用作 A/D 转换器的输入时, 必须在通过 ADS 寄存器选择模拟输入通道前, 允许运行。

备注 n=0、1

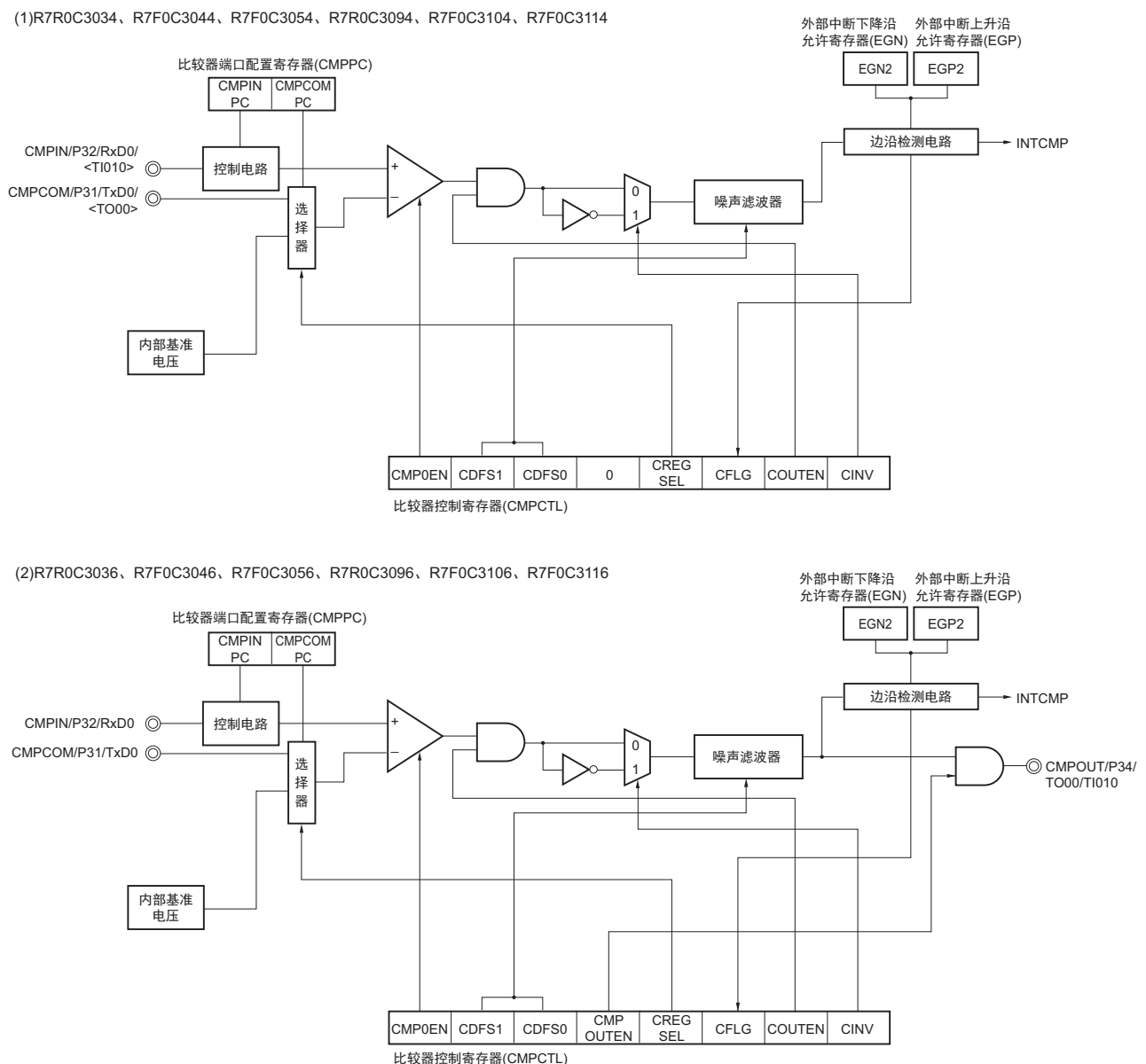
## 第 12 章 比较器

### 12.1 比较器的功能

在 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x 中装有比较器。比较器具有以下功能。

- 可选择以下基准电压。
  1. 内部基准电压
  2. 比较器公共引脚（CMPCOM）的输入电压
- 可检测比较器输出的有效边沿，并且产生中断信号。可通过 EGP2 位和 EGN2 位设置有效边沿（参照“第 14 章 中断功能”）。
- 可选择噪声消除数字滤波器的消除宽度。

图 12-1 比较器的框图



## 12.2 比较器的结构

比较器由以下硬件构成。

表 12-1 比较器的结构

项目	结构
控制寄存器	比较器控制寄存器 (CMPCTL) 比较器端口配置寄存器 (CMPPC) 端口模式寄存器 3 (PM3) 端口寄存器 3 (P3)

## 12.3 比较器的控制寄存器

比较器使用以下 3 种寄存器。

- 比较器控制寄存器（CMPCTL）
- 比较器端口配置寄存器（CMPPC）
- 端口模式寄存器 3（PM3）

### (1) 比较器控制寄存器（CMPCTL）

比较器控制寄存器控制比较器的运行，以及设置比较器输出的允许 / 禁止、输出取反、噪声消除宽度和基准电压。

通过 1 位或 8 位存储器操作指令设置 CMPCTL。

在产生复位信号时，CMPCTL 为“00H”。

图 12-2 比较器控制寄存器（CMPCTL）的格式（1/2）

地址：FF61H 复位时：00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
CMPCTL	CMP0EN	CDFS1	CDFS0	CMPOUTEN <sup>注</sup>	CREGSEL	CFLG	COUTEN	CINV

CMP0EN	比较器 0 的运行控制
0	停止运行
1	允许运行 允许比较器 0 的 + 侧外部引脚（CMPIN）输入

CDFS1	CDFS0	设置噪声消除宽度
0	0	不使用噪声滤波器
0	1	$2f_{prs}$
1	0	$2^2f_{prs}$
1	1	$2^3f_{prs}$

CMPOUTEN <sup>注</sup>	控制比较器输出引脚
0	禁止比较器输出的外部引脚输出
1	允许比较器输出的外部引脚输出

CREGSEL	选择基准电压
0	外部基准电压 CMPCOM
1	内部基准电压（稳压器电压：参照“第 19 章 稳压器”）

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116。  
当为 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 时，必须将此位置“0”。

图 12-2 比较器控制寄存器 (CMPCTL) 的格式 (2/2)

CFLG	监视比较器输出的标志
0	比较器输出值为“0”
1	比较器输出值为“1”
COUTEN	允许 / 禁止比较器输出
0	禁止输出 (输出信号: 固定为低电平)
1	允许输出
CINV	设置输出取反
0	正相
1	反相

- 注意 1. 在设置为禁止比较器运行 (CMP0EN=0) 的状态下, 改写 CDFS1、CDFS0、CMPOUTEN、CREGSEL、COUTEN、CINV。
2. 根据设置值, 有可能多消除 1 个外围硬件时钟频率 ( $f_{PRS}$ ) 时钟的噪声消除宽度。
3. 如果比较器输出噪声的间隔在“设置的噪声消除宽度 + 1 个时钟”以内, 就可能输出非法波形。

备注  $f_{PRS}$ : 外围硬件时钟频率

## (2) 比较器端口配置寄存器 (CMPPC)

比较器端口配置寄存器进行比较器输入引脚的模拟 / 数字切换。

通过 1 位或 8 位存储器操作指令设置 CMPPC。

在产生复位信号时, CMPPC 为“00H”。对于没有内置比较器的产品 (R7F0C300x、R7F0C301x、R7F0C302x、R7F0C306x、R7F0C307x、R7F0C308x), 复位后数字功能处于无效状态。因此在用作数字功能时, 必须将 CMPCOMPC 和 CMPINPC 置“1”。

图 12-3 比较器端口配置寄存器 (CMPPC) 的格式

地址: FF62H 复位时: 00H R/W

符号	7	6	5	4	3	2	<1>	<0>
CMPPC	0	0	0	0	0	0	CMPCOMPC	CMPINPC
CMPCOMPC	P31/TxD0/CMPCOM/<TO00> 注的模拟 (A) / 数字 (D) 选择							
0	用作模拟 (A) (默认)							
1	用作数字 (D)							
CMPINPC	P32/RxD0/CMPIN/<TI010> 注的模拟 (A) / 数字 (D) 选择							
0	用作模拟 (A) (默认)							
1	用作数字 (D)							

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114。

注意 因为 P31 和 P32 在默认状态下为模拟模式, 所以当 P31 和 P32 用作数字功能时, 必须将 CMPCOMPC 和 CMPINPC 置“1”。

## (3) 端口模式寄存器 3 (PM3)

端口模式寄存器 3 以 1 位单位设置端口 3 的输入 / 输出。

将 CMPIN/P32/RxD0/<TI010> 注引脚和 CMPCOM/P31/TxD0/<TO00> 注引脚分别用作比较器输入和比较器公共输入时，必须将 PM31 位和 PM32 位置“1”。此时，P31 和 P32 的输出锁存器即可是“0”也可能是“1”。

通过 1 位或 8 位存储器操作指令设置 PM3。

在产生复位信号时，PM3 为“FFH”。

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114。

图 12-4 端口模式寄存器 3 (PM3) 的格式

地址: FF23H 复位时: FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34 注	PM33 注	PM32	PM31	PM30

注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116。

注意 必须将 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 的 bit3 ~ bit7 和 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 的 bit5 ~ bit7 置“1”。

PM3n	选择 P3n 引脚的输入 / 输出模式 (n=0 ~ 4)
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

注意 使用内部基准电压时，可在输入模式中使用 CMPCOM 引脚复用的端口功能。但是，禁止在输出模式中使用。另外，禁止存取端口寄存器 3 (P3)。

在使用 P32/RxD0/CMPIN/<TI010> 注和 P31/TxD0/CMPCOM/<TO00> 注时，根据使用的引脚功能设置寄存器 (参照表 12-2、表 12-3)。

表 12-2 P32/RxD0/CMPIN/&lt;TI010&gt; 注引脚功能的设置

CMPPC 寄存器	PM3 寄存器	CMP0EN 位	P32/RxD0/CMPIN/<TI010> 注引脚
选择数字输入 / 输出	输入模式	0	数字输入
		1	禁止设置
	输出模式	0	数字输出
		1	禁止设置
选择模拟输入	输入模式	0	比较器输入 (禁止)
		1	比较器输入 (允许)
	输出模式	—	禁止设置

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114。

备注 CMPPC: 比较器端口配置寄存器  
 PM3: 端口模式寄存器 3  
 CMP0EN: 比较器控制寄存器 (CMPCTL) 的 bit7  
 ADS: 模拟输入通道指定寄存器



表 12-3 P31/TxD0/CMPCOM/&lt;TO00&gt; 注引脚功能的设置

CMPPC 寄存器	PM3 寄存器	CMP0EN 位	CREGSEL 位	P31/TxD0/CMPCOM/<TO00> 注引脚
选择数字输入 / 输出	输入模式	0	0	数字输入
			1	
		1	0	禁止设置
			1	数字输入 / 比较器公共输入（禁止）
	输出模式	0	0	数字输出
			1	
1		0	禁止设置	
		1	数字输出 / 比较器公共输入（禁止）	
选择模拟输入	输入模式	0	0	比较器公共输入（禁止）
			1	
		1	0	比较器公共输入（禁止）
			1	比较器公共输入（允许）
	输出模式	—	—	禁止设置

注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114。

备注 CMPPC: 比较器端口配置寄存器  
 PM3: 端口模式寄存器 3  
 CMP0EN: 比较器控制寄存器（CMPCTL）的 bit7  
 ADS: 模拟输入通道指定寄存器

## 12.4 比较器的运行

### 12.4.1 开始比较器的运行（内部基准电压用作基准电压的情况）

图 12-5 开始比较器运行的步骤示例（内部基准电压用作基准电压的情况）



注 必须在设置 CREGSEL 位后至少经过 20 $\mu$ s 后再进行设置。

## 12.4.2 开始比较器的运行（CMPCOM 引脚的输入电压用作基准电压的情况）

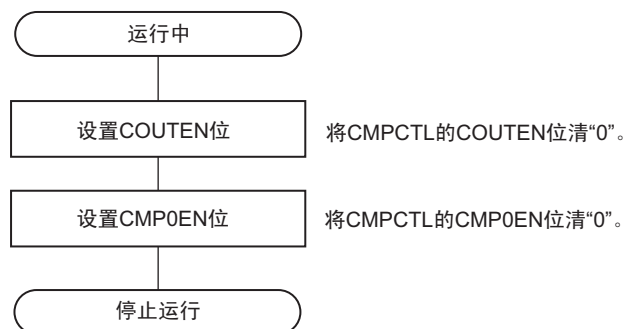
图 12-6 开始比较器运行的步骤示例  
（比较器公共引脚（CMPCOM）的输入电压用作基准电压的情况）



注 必须在设置 CMP0EN 位后至少经过 1μs 后再进行设置。

## 12.4.3 停止比较器的运行

图 12-7 停止比较器运行的步骤示例



## 第 13 章 串行接口 UART0

### 13.1 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“13.4.1 运行停止模式”。

#### (2) 异步串行接口 (UART) 模式

异步串行接口模式的功能概要如下所示。

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 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。

## 13.2 串行接口 UART0 的结构

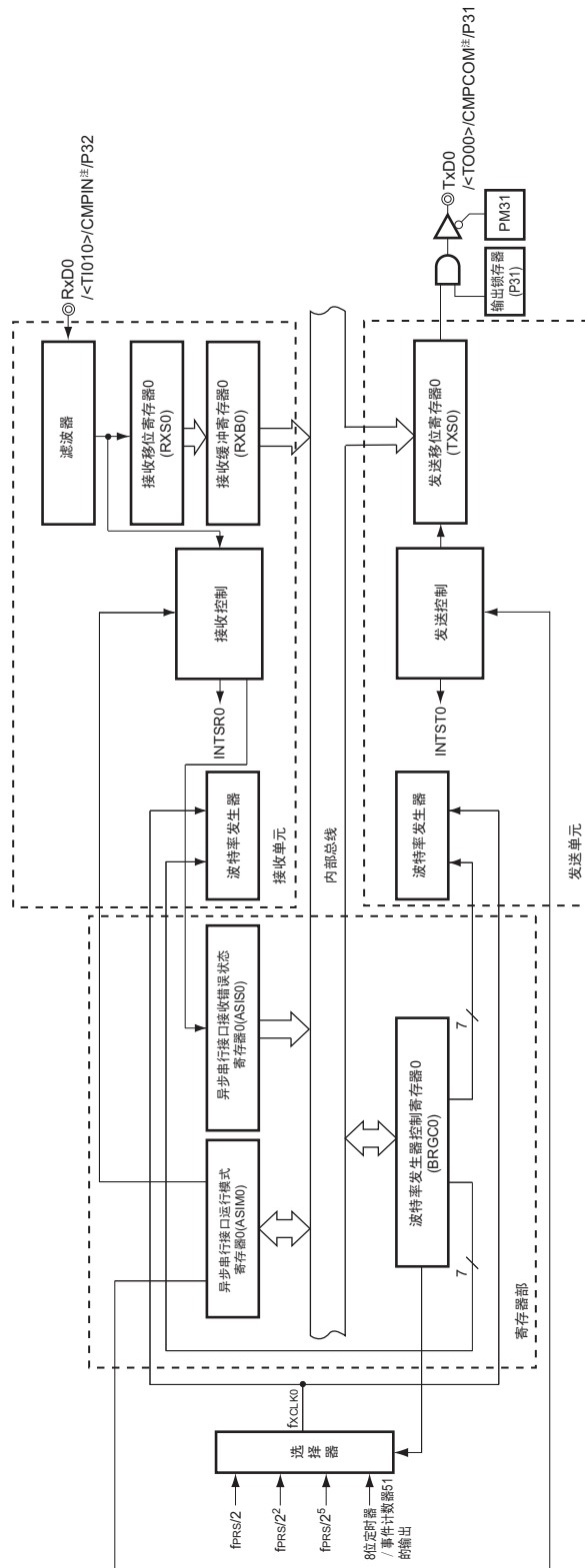
串行接口 UART0 由以下硬件构成。

表 13-1 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 3 (PM3) 端口寄存器 3 (P3)

图 13-1 串行接口 UART0 的框图 (1/2)

(1) 16 引脚产品

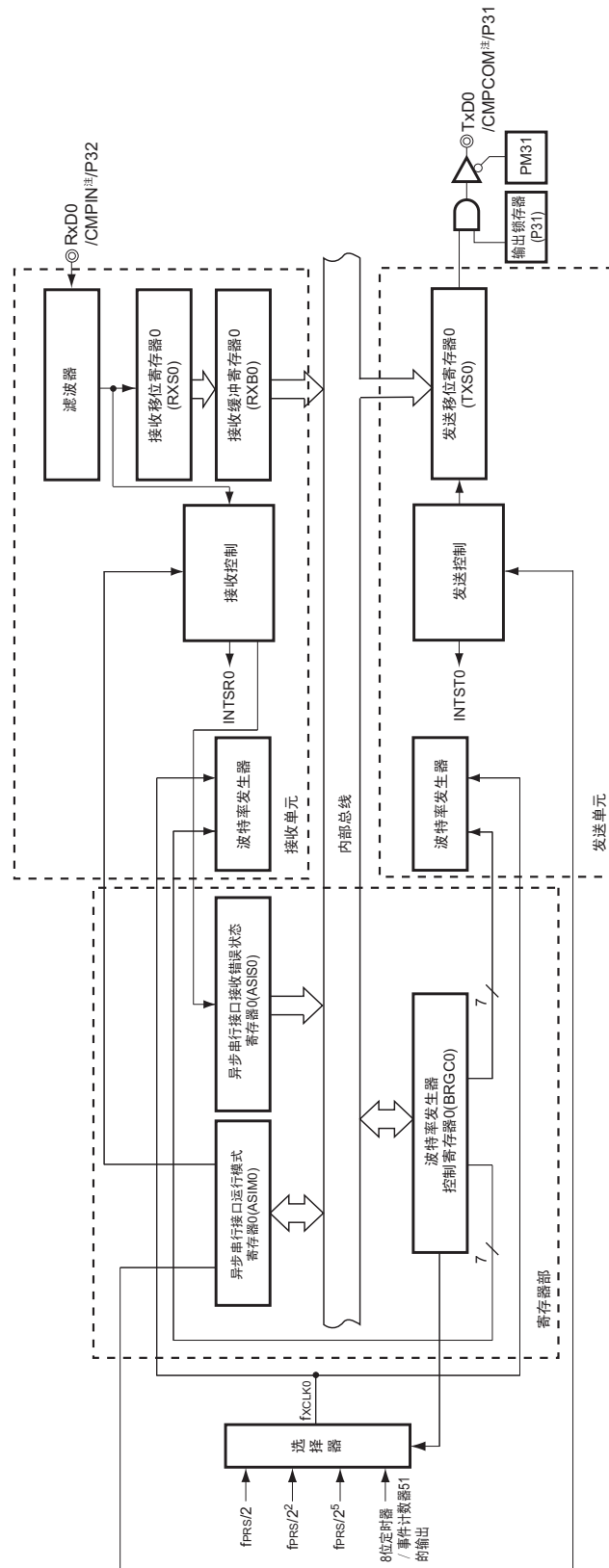


注 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 (内置比较器的产品)。

备注 可通过设置端口复用切换控制寄存器 (MUXSEL) 指定上图 <> 内的功能。

图 13-1 串行接口 UART0 的框图 (2/2)

(2) 20 引脚产品



注 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置比较器的产品)。

### (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。

在产生复位信号并且 POWER0=0 时，RXB0 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 TxD0 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。



### 13.3 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 3 (PM3)
- 端口寄存器 3 (P3)

#### (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。  
在产生复位信号时，ASIM0 为“01H”。

图 13-2 异步串行接口运行模式寄存器 0 (ASIM0) 的格式 (1/2)

地址: FF50H 复位时: 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 13-2 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 <sup>注</sup>
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。

在产生复位信号并且 ASIM0 的 bit7 (POWER0) =0、bit5 (RXE0) =0 时, ASIS0 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 13-3 异步串行接口接收错误状态寄存器 0 (ASIS0) 的格式

地址: FF53H 复位时: 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<sub>PRS</sub>) 停止时, 不能从 ASIS0 读取数据。详细内容请参照“第 26 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。

在产生复位信号时, BRGC0 为“1FH”。

图 13-4 波特率发生器控制寄存器 0 (BRGC0) 的格式

地址: FF51H 复位时: 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	TM51 的输出注			
0	1	$f_{PRS}/2$	1MHz	2.5MHz	5MHz
1	0	$f_{PRS}/2^3$	250kHz	625kHz	1.25MHz
1	1	$f_{PRS}/2^5$	62.5kHz	156.25kHz	312.5kHz

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 TM51 输出作为基本时钟时的注意事项如下所示。

- TM51 与 CR51 匹配时的清除 & 启动模式 (TMC516=0)  
允许定时器 F/F 的取反运行 (TMC511=1), 事先开始 8 位定时器 / 事件计数器 51 的运行。
- PWM 模式 (TMC516=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 51 的运行。

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: 通过 MDL04 ~ MDL00 位设置的值 (k = 8、9、10、...31)

4. x: 任意

5. TMC516: 8 位定时器模式控制寄存器 51 (TMC51) 的 bit6

TMC511: TMC51 的 bit1

**(4) 端口模式寄存器 3 (PM3)**

端口模式寄存器 3 以 1 位单位设置端口 3 的输入 / 输出。

将 P31/TxD0/CMPCOM<sup>注 1</sup> (16 引脚产品时, P31/TxD0/<TO00>/CMPCOM<sup>注 2</sup>) 引脚用作串行接口的数据输出时, 必须将 PM31 置“0”, 将 P31 的输出锁存器置“1”。

将 P32/RxD0/CMPIN<sup>注 1</sup> (16 引脚产品时, P32/RxD0/<TI010>/CMPIN<sup>注 2</sup>) 引脚用作串行接口的数据输入时, 必须将 PM32 置“1”。此时, P32 的输出锁存器即可是“0”也可是“1”。

通过 1 位或 8 位存储器操作指令设置 PM3。

在产生复位信号时, PM3 为“FFH”。

图 13-5 端口模式寄存器 3 (PM3) 的格式

地址: FF23H 复位时: FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34 <sup>注 3</sup>	PM33 <sup>注 3</sup>	PM32	PM31	PM30

PM3n	选择 P3n 引脚的输入 / 输出模式 (n=0 ~ 4)
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

- 注 1. 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置比较器的产品)。
2. 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 (内置比较器的产品)。
3. 只限 20 引脚产品。

备注 可通过设置端口复用切换控制寄存器 (MUXSEL) 指定上述 < > 内的功能。

## 13.4 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

### 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。

在产生复位信号时，ASIM0 为“01H”。

地址：FF50H 复位时：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/CMPIN/P32 和 TxD0/CMPCOM/P31（16 引脚产品时，RxD0/<TI010>/CMPIN/P32 和 TxD0/<TO00>/CMPCOM/P31）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 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 3（PM3）
- 端口寄存器 3（P3）

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 13-4）
2. 设置 ASIM0 寄存器的 bit1 ~ bit4（SL0、CL0、PS00、PS01）（参照图 13-2）
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 13-2 寄存器设置与引脚的关系（16 引脚产品）

POWER0	TXE0	RXE0	PM31	P31	PM32	P32	UART0 的运行	引脚功能	
								TxD0/<TO00>/ CMPCOM 注 1/P31	RxD0/<TI010>/ CMPIN 注 1/P32
0	0	0	x 注 2	x 注 2	x 注 2	x 注 2	停止	<TO00>/ CMPCOM 注 1/P31	<TI010>/ CMPIN 注 1/P32
1	0	1	x 注 2	x 注 2	1	x	接收	<TO00>/ CMPCOM 注 1/P31	RxD0
	1	0	0	1	x 注 2	x 注 2	发送	TxD0	<TI010>/ CMPIN 注 1/P32
	1	1	0	1	1	x	发送 / 接收	TxD0	RxD0

表 13-3 寄存器设置与引脚的关系 (20 引脚产品)

POWER0	TXE0	RXE0	PM31	P31	PM32	P32	UART0 的运行	引脚功能	
								TxD0/ CMPCOM 注 3/P31	RxD0/ CMPIN 注 3/P32
0	0	0	x 注 4	x 注 4	x 注 4	x 注 4	停止	CMPCOM 注 3/P31	CMPIN 注 3/P32
1	0	1	x 注 4	x 注 4	1	x	接收	CMPCOM 注 3/P31	RxD0
	1	0	0	1	x 注 4	x 注 4	发送	TxD0	CMPIN 注 3/P32
	1	1	0	1	1	x	发送 / 接收	TxD0	RxD0

- 注 1. 只限 R7F0C3034、R7F0C3044、R7F0C3054、R7F0C3094、R7F0C3104、R7F0C3114 (内置比较器的产品)。
2. 可作为端口功能、比较器或者定时器设置。
3. 只限 R7F0C3036、R7F0C3046、R7F0C3056、R7F0C3096、R7F0C3106、R7F0C3116 (内置比较器的产品)。
4. 可作为端口功能或者比较器设置。

备注 1. 可通过设置端口复用切换控制寄存器 (MUXSEL) 指定上表 < > 内的功能。

2. 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

PM3x: 端口模式寄存器

P3x: 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 13-6 和图 13-7 所示。

图 13-6 正常 UART 发送 / 接收数据的格式



1 个数据帧由以下各位组成。

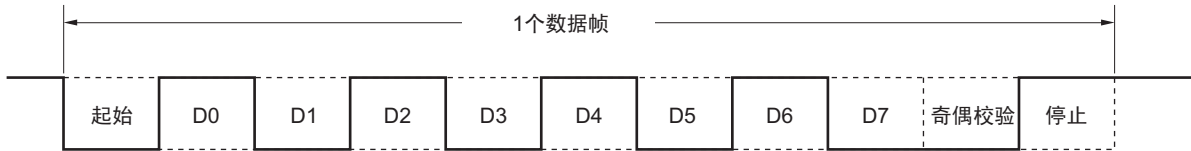
- 起始位 ...1 位
- 字符位 ...7 位或者 8 位 (LSB 优先)
- 奇偶校验位 ... 偶校验 / 奇校验 / 零校验 / 无奇偶校验
- 停止位 ...1 位或者 2 位

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。

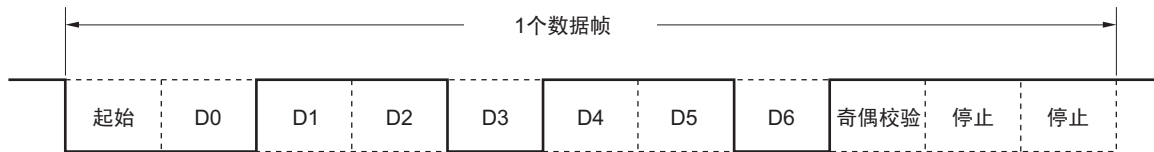


图 13-7 正常 UART 发送 / 接收数据的波形示例

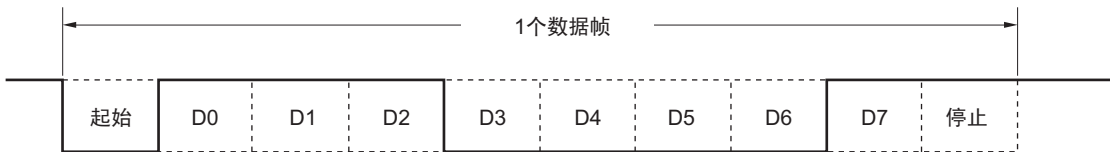
1. 数据长度：8 位、奇偶校验：偶校验、停止位：1 位、通信数据：55H



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)。

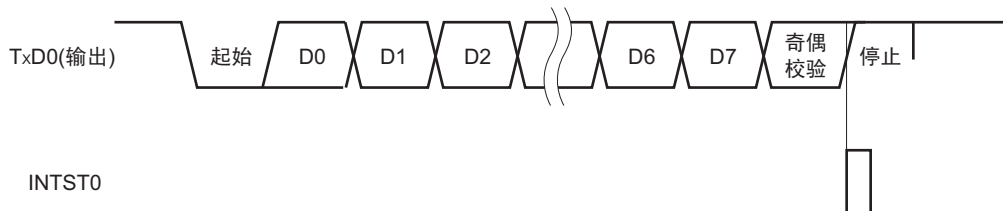
直到下一个发送数据写入 TXS0，才能中止发送运行。

产生发送结束中断请求 (INTST0) 的时序如图 13-8 所示。在输出最后的停止位的同时产生 INTST0。

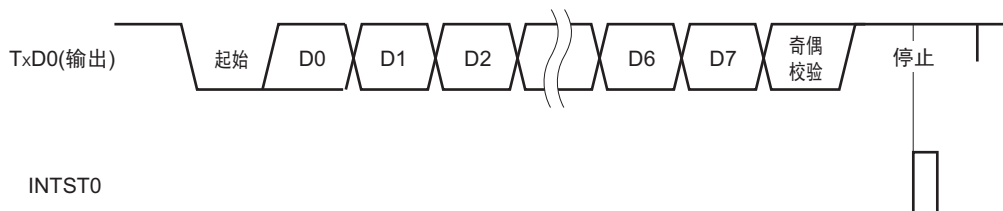
**注意** 从发送数据写入 TXS0 到产生发送结束中断信号 (INTST0) 前，不能写入下一个发送数据。

图 13-8 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。

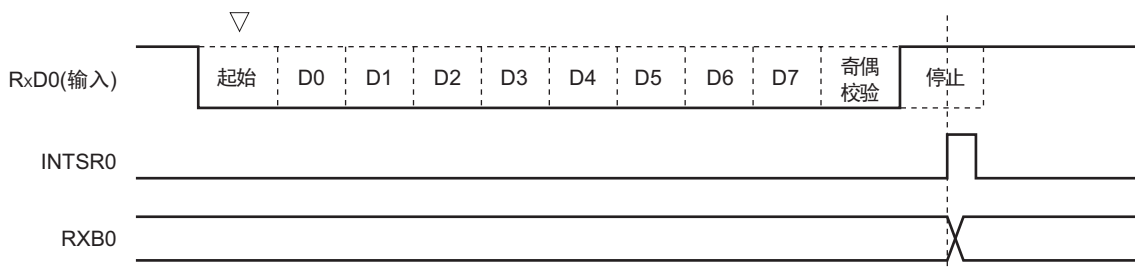
如果检测出 RxD0 引脚输入的下沿，波特率发生器的 5 位计数器就开始计数，在对波特率发生器控制寄存器 0 (BRGC0) 的设置值进行计数后，再次进行 RxD0 引脚输入的采样（如图 13-9 中的“▽”）。如果采样后的 RxD0 引脚为低电平，就识别为起始位。

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 13-9 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 13-3）

通过读取 ASIS0，可将 ASIS0 的内容清“0”。

表 13-4 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

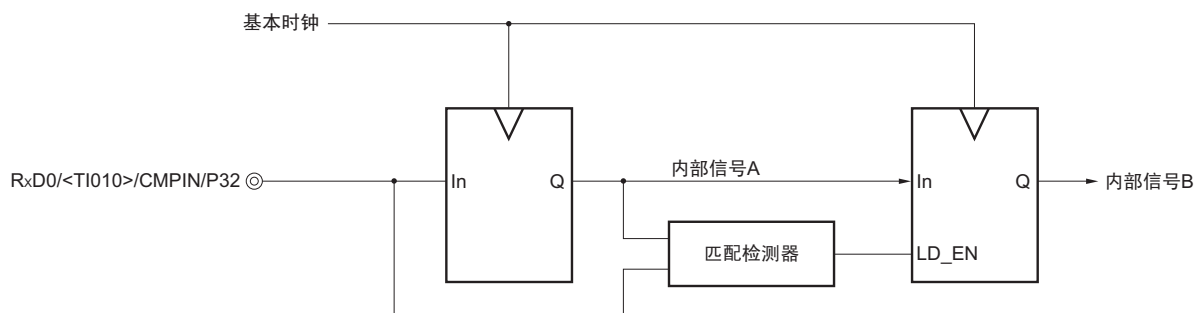
接收数据的噪声滤波器以预分频器输出的基本时钟对 RxD0 信号进行采样。

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。

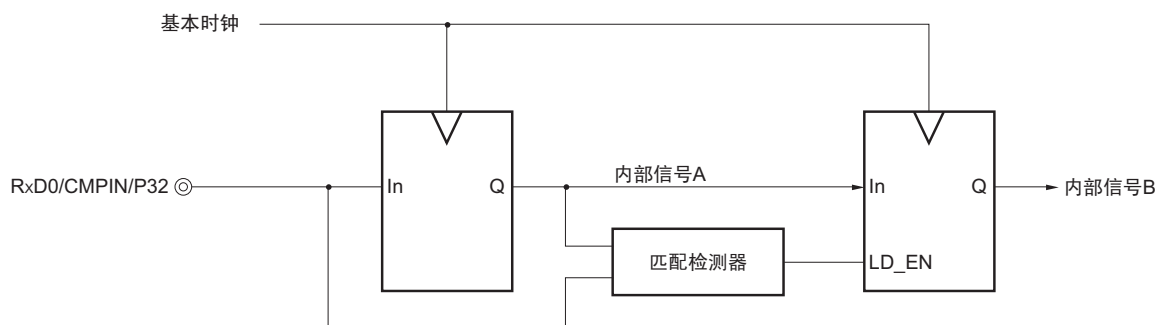
在如图 13-10 所示的电路中，接收运行的内部处理比外部的信号状态迟 2 个时钟。

图 13-10 噪声滤波器电路

## (1) 16 引脚产品



## (2) 20 引脚产品



备注 可通过设置端口复用切换控制寄存器 (MUXSEL) 指定上图 <> 内的功能。

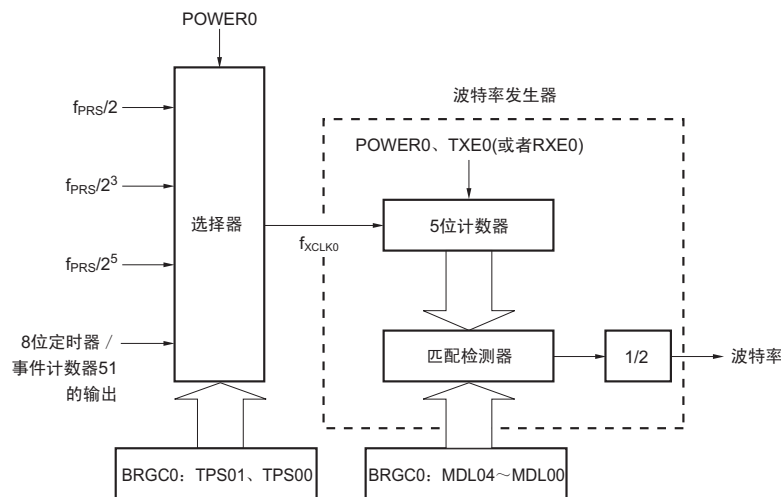
### 13.4.3 专用波特率发生器

专用波特率发生器由源时钟选择器和 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 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 13-11 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 ( $f_{XCLK0}/8 \sim f_{XCLK0}/31$ )。

### 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。

$$\bullet \text{ 波特率} = \frac{f_{\text{XCLK0}}}{2 \times k} \text{ [bps]}$$

$f_{\text{XCLK0}}$ : 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 13-5 TPS01 和 TPS00 的设置值

TPS01	TPS00	选择基本时钟 ( $f_{\text{XCLK0}}$ )			
		$f_{\text{PRS}}=2\text{MHz}$	$f_{\text{PRS}}=5\text{MHz}$	$f_{\text{PRS}}=10\text{MHz}$	
0	0	TM51 的输出注			
0	1	$f_{\text{PRS}}/2$	1MHz	2.5MHz	5MHz
1	0	$f_{\text{PRS}}/2^3$	250kHz	625kHz	1.25MHz
1	1	$f_{\text{PRS}}/2^5$	62.5kHz	156.25kHz	312.5kHz

注 选择 TM51 输出作为基本时钟时的注意事项如下所示。

- TM51 与 CR51 匹配时的清除 & 启动模式 (TMC516=0)  
允许定时器 F/F 的取反运行 (TMC511=1)，事先开始 8 位定时器 / 事件计数器 51 的运行。
- PWM 模式 (TMC516=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 51 的运行。

#### (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 = 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 13-6 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

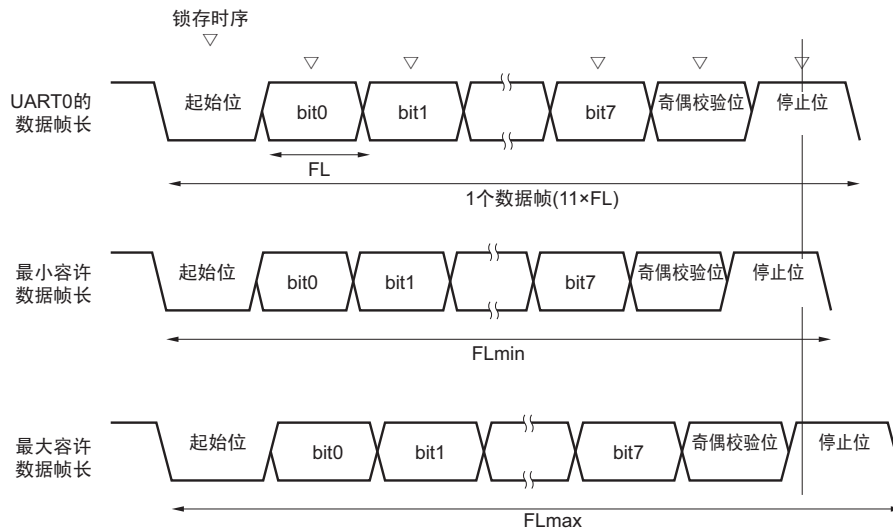
ERR: 波特率误差

## (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 13-12 接收时的波特率允许范围



如图 13-2 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 13-7 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 14 章 中断功能

### 14.1 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（**PROL**、**PROH**、**PRIL**），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 14-1。

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。

## 14.2 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 14-1）。

表 14-1 中断源一览（1/2）

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	引脚输入边沿检测	0006H				
			2	INTP1		0008H				
			3	INTCMP 注 4	比较器边沿检测注 4	000AH 注 4				
	—	—	—	4	—	—	000CH			
				5	—		000EH			
				6	—		0010H			
				7	—		0012H			
	内部	(A)	8	INTSR0	UART0 的接收结束 / 接收错误发生	0014H				
			9	INTST0	UART0 的发送结束	0016H				
	—	—	—	10	—	—	0018H			
	内部	(A)	11	INTTMH1	TMH1 和 CMP01 匹配 (指定比较寄存器时)	001AH				
	—	—	—	12	—	—	001CH			
				13			001EH			
	内部	(A)	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
	—	—	—	17	—	—	0026H			
				18			0028H			
	内部	(A)	19	INTTM51 注 5	TM51 和 CR51 匹配 (指定比较寄存器时)	002AH				
—	—	—	20	—	—	002CH				

- 注 1. 基本结构类型 (A) ~ (C) 分别对应图 14-1 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，“28”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. 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x（内置比较器的产品）。
5. 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 8-11 传送时序”）。

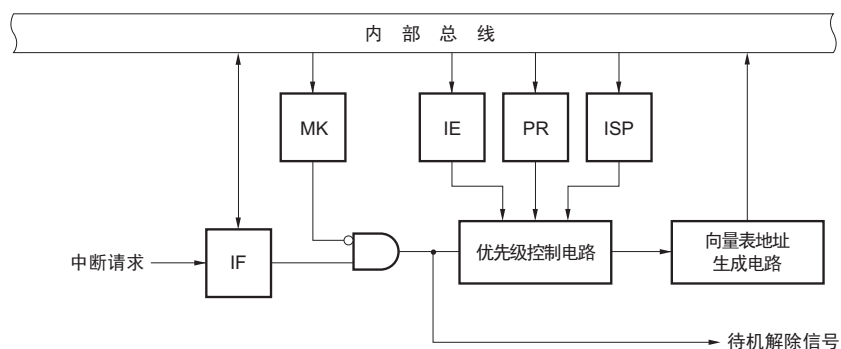
表 14-1 中断源一览 (2/2)

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	—	—	21	—	—	002EH
			22			0030H
			23			0032H
			24			0034H
			25			0036H
			26			0038H
			27			003AH
			28			003CH
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	

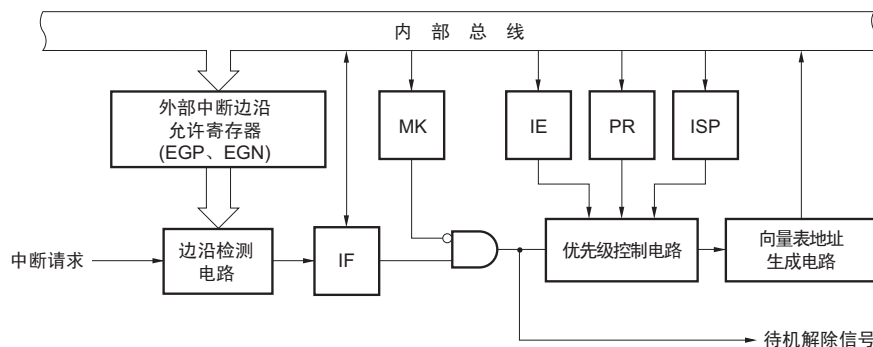
- 注 1. 基本结构类型 (A) ~ (C) 分别对应图 14-1 中的 (A) ~ (C)。
2. 在同时产生多个可屏蔽中断时，默认优先级决定优先处理的向量中断。“0”为最高优先级，“28”为最低优先级。
3. 选择低电压检测寄存器 (LVIM) 的 bit1 (LVIMD) 为“1”时的情况。

图 14-1 中断功能的基本结构 (1/2)

(A) 内部可屏蔽中断



(B) 外部可屏蔽中断(INTPn、INTCMP注)



注 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x (内置比较器的产品)。

备注 n=0、1

IF: 中断请求标志

IE: 中断允许标志

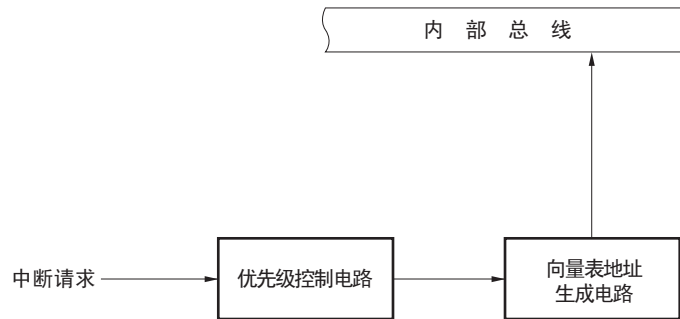
ISP: 优先级控制标志

MK: 中断屏蔽标志

PR: 优先级指定标志

图 14-1 中断功能的基本结构 (2/2)

## (C) 软件中断



- 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

### 14.3 中断功能的控制寄存器

通过以下 6 种寄存器控制中断功能。

- 中断请求标志寄存器 (IF0L、IF0H、IF1L)
- 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L)
- 优先级指定标志寄存器 (PROL、PROH、PR1L)
- 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 14-2 所示。

表 14-2 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	IF0L	LVIMK	MK0L	LVIPR	PROL
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTCMP 注 1	CMPIF 注 1		CMPMK 注 1		CMPPR 注 1	
INTSR0	SRIF0	IF0H	SRMK0	MK0H	SRPR0	PROH
INTST0	STIF0		STMK0		STPR0	
INTTMH1	TMIFH1		TMMKH1		TMPRH1	
INTTM000	TMIF000		TMMK000		TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010	
INTAD	ADIF	IF1L	ADMK	MK1L	ADPR	PR1L
INTTM51 注 2	TMIF51		TMMK51		TMPR51	

- 注 1. 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x (内置比较器的产品)。
- 注 2. 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 8-11 传送时序”)。

## (1) 中断请求标志寄存器 (IF0L、IF0H、IF1L)

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 和 IF1L。另外，将 IF0L 与 IF0H 的组合用作 16 位寄存器 IF0 时，通过 16 位存储器操作指令设置 IF0。

在产生复位信号时，IF0L、IF0H 和 IF1L 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 14-2 中断请求标志寄存器 (IF0L、IF0H、IF1L) 的格式

地址: FFE0H 复位时: 00H R/W

符号	7	6	5	4	<3>	<2>	<1>	<0>
IF0L	0	0	0	0	CMPIF 注	PIF1	PIF0	LVIIIF

地址: FFE1H 复位时: 00H R/W

符号	<7>	<6>	5	4	<3>	2	<1>	<0>
IF0H	TMIF010	TMIF000	0	0	TMIFH1	0	STIF0	SRIF0

地址: FFE2H 复位时: 00H R/W

符号	7	6	5	4	<3>	2	1	<0>
IF1L	0	0	0	0	TMIF51	0	0	ADIF

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 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x (内置比较器的产品)。

注意 必须将 IF0L 的 bit4 ~ bit7、IF0H 的 bit2、bit4、bit5、IF1L 的 bit1、bit2、bit4 ~ bit7 置“0”。



## (2) 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L)

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 和 MK1L。另外，在将 MK0L 和 MK0H 的组合用作 16 位寄存器 MK0 时，通过 16 位存储器操作指令设置 MK0。

在产生复位信号时，MK0L、MK0H 和 MK1L 为“FFH”。

图 14-3 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L) 的格式

地址：FFE4H 复位时：FFH R/W

符号	7	6	5	4	<3>	<2>	<1>	<0>
MK0L	1	1	1	1	CMPMK 注	PMK1	PMK0	LVIMK

地址：FFE5H 复位时：FFH R/W

符号	<7>	<6>	5	4	<3>	2	<1>	<0>
MK0H	TMMK010	TMMK000	1	1	TMMKH1	1	STMK0	SRMK0

地址：FFE6H 复位时：FFH R/W

符号	7	6	5	4	<3>	2	1	<0>
MK1L	1	1	1	1	TMMK51	1	1	ADMK

XXMKX	中断处理的控制	
0	允许中断处理	
1	禁止中断处理	

注 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x (内置比较器的产品)。

注意 必须将 MK0L 的 bit4 ~ bit7、MK0H 的 bit2、bit4、bit5、MK1L 的 bit1、bit2、bit4 ~ bit7 置“1”。

## (3) 优先级指定标志寄存器 (PR0L、PR0H、PR1L)

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 和 PR1L。另外，在将 PR0L 和 PR0H 的组合用作 16 位寄存器 PR0 时，通过 16 位存储器操作指令设置 PR0。

产生复位信号时，PR0L、PR0H 和 PR1L 为“FFH”。

图 14-4 优先级指定标志寄存器 (PR0L、PR0H、PR1L) 的格式

地址: FFE8H 复位时: FFH R/W

符号	7	6	5	4	<3>	<2>	<1>	<0>
PR0L	1	1	1	1	CMPPR 注	PPR1	PPR0	LVIPR

地址: FFE9H 复位时: FFH R/W

符号	<7>	<6>	5	4	<3>	2	<1>	<0>
PR0H	TMPR010	TMPR000	1	1	TMPRH1	1	STPR0	SRPR0

地址: FFEAH 复位时: FFH R/W

符号	7	6	5	4	<3>	2	1	<0>
PR1L	1	1	1	1	TMPR51	1	1	ADPR

XXPRX	优先级的选择
0	高优先级
1	低优先级

注 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x (内置比较器的产品)。

注意 必须将 PR0L 的 bit4 ~ bit7、PR0H 的 bit2、bit4、bit5、PR1L 的 bit1、bit2、bit4 ~ bit7 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<sub>n</sub> 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 为“00H”。

图 14-5 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 复位时: 00H R/W

符号	7	6	5	4	3	2	1	0
EGP	0	0	0	0	0	EGP2	EGP1	EGP0

地址: FF49H 复位时: 00H R/W

符号	7	6	5	4	3	2	1	0
EGN	0	0	0	0	0	EGN2	EGN1	EGN0

EGP <sub>n</sub>	EGN <sub>n</sub>	选择 INTP <sub>n</sub> 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 必须将 EGP 和 EGN 的 bit3 ~ bit7 置“0”。

备注 n=0 ~ 2

EGP<sub>n</sub> 和 EGN<sub>n</sub> 的对应端口如表 14-3 所示。

表 14-3 与 EGP<sub>n</sub> 和 EGN<sub>n</sub> 对应的端口

检测允许位		边沿检测端口	中断请求信号
EGP0	EGN0	P30	INTP0
EGP1	EGN1	P33	INTP1
EGP2	EGN2	—	INTCMP 注

注 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x (内置比较器的产品)。

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<sub>n</sub> 和 EGN<sub>n</sub> 置“0”后切换到端口模式。

备注 n=0 ~ 2

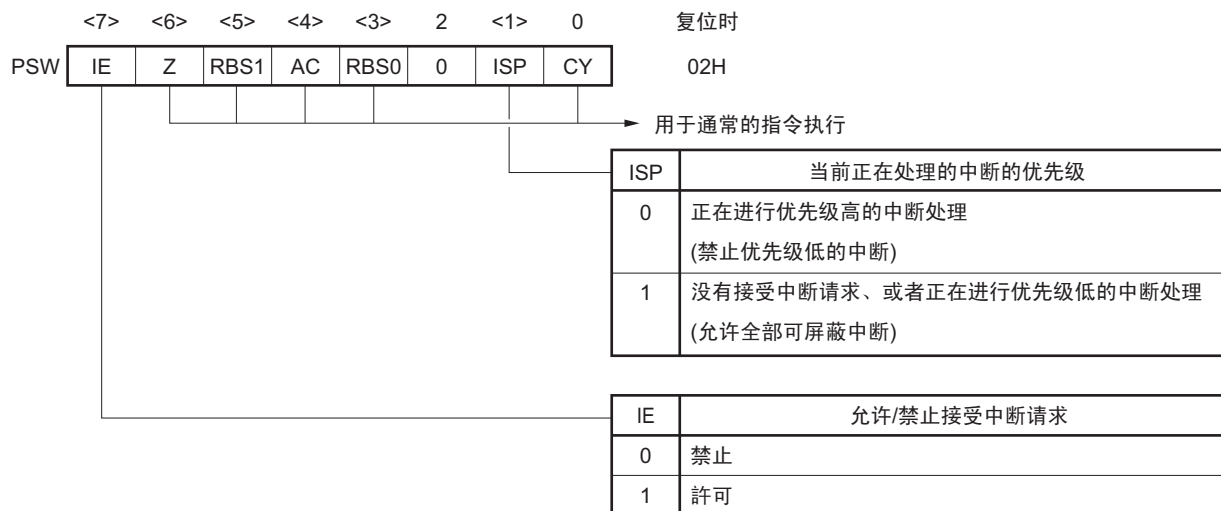
### (5) 程序状态字 (PSW)

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。

在产生复位信号时，PSW 为“02H”。

图 14-6 程序状态字的格式



## 14.4 中断处理的运行

### 14.4.1 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 14-4 所示。

中断请求的接受时序请参照图 14-8 和图 14-9。

表 14-4 从产生可屏蔽中断请求到处理的时间

	最短时间	最长时间 <sup>注</sup>
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<sub>CPU</sub>（f<sub>CPU</sub>：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。

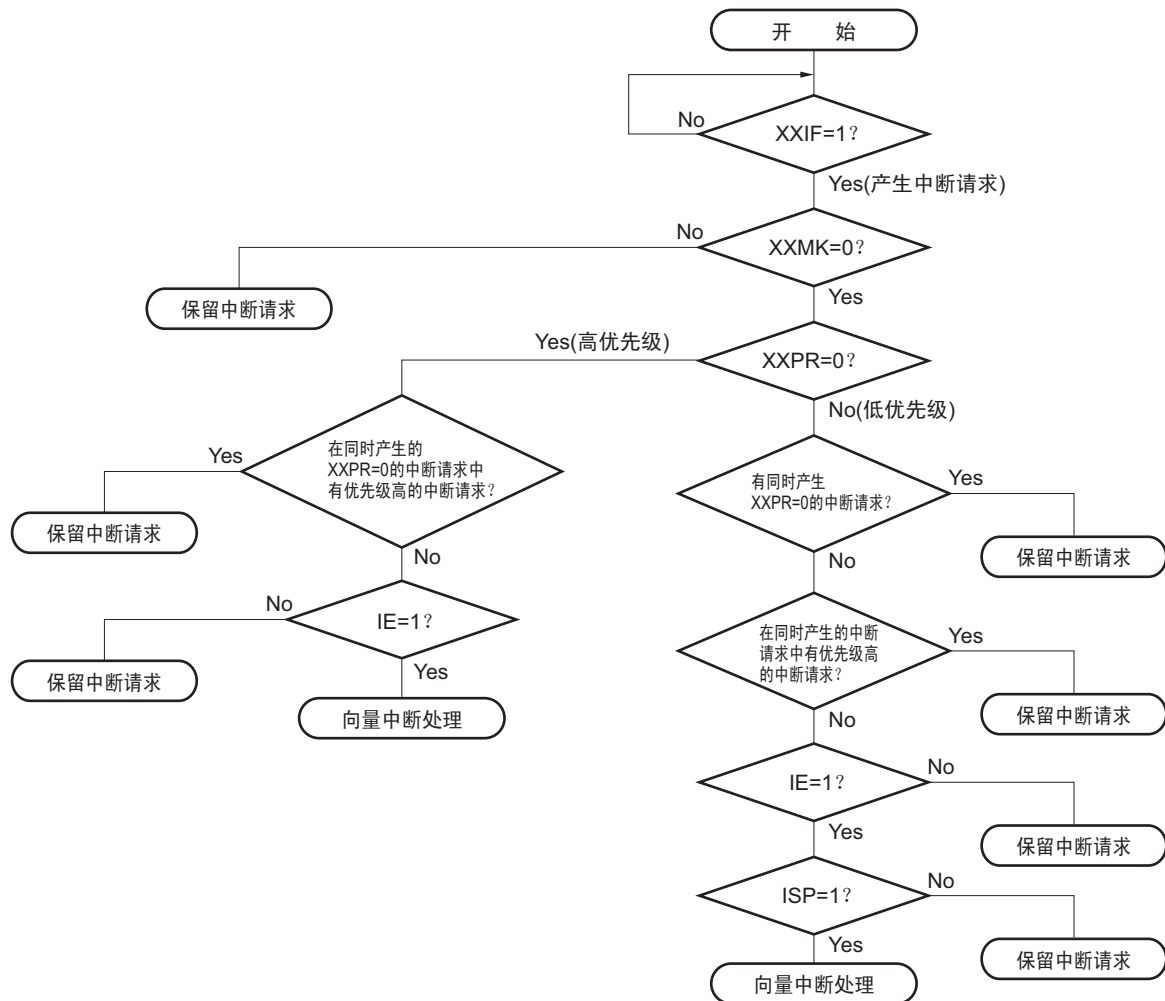
一旦被保留的中断请求进入可接受状态，就可进行接受。

接受中断请求的算法如图 14-7 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 14-7 中断请求的接受处理算法



XXIF: 中断请求标志

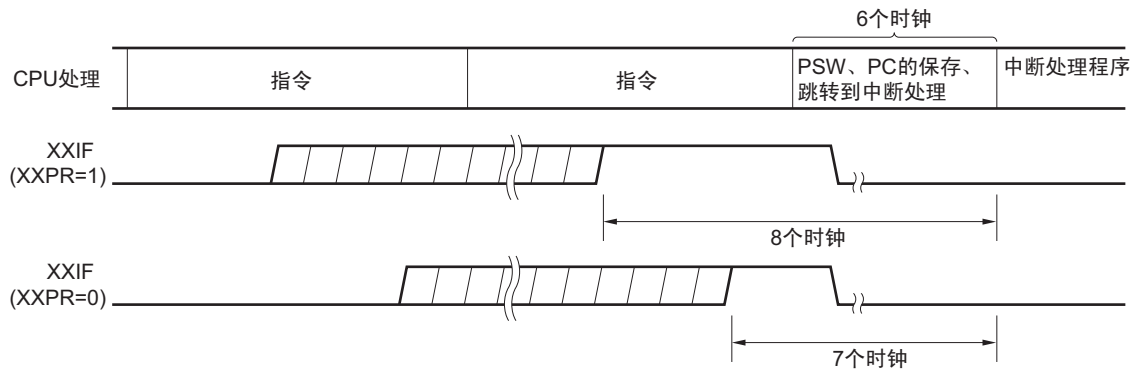
XXMK: 中断屏蔽标志

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)

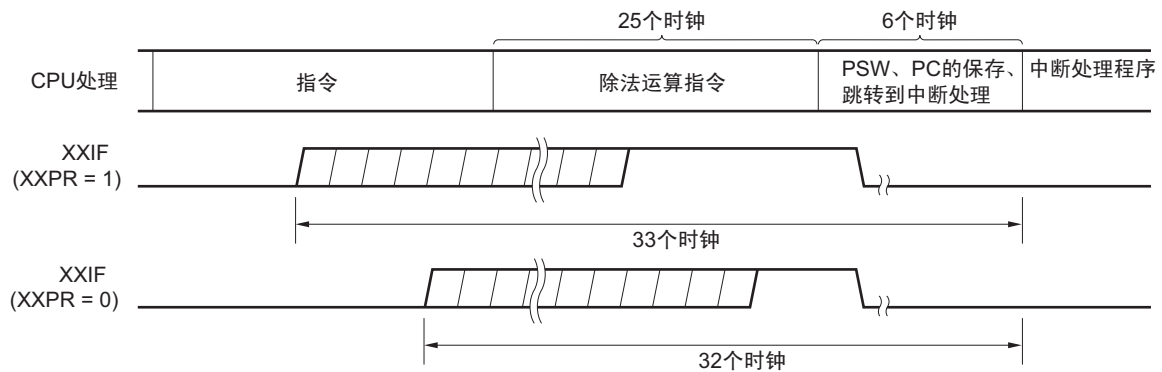
ISP: 表示当前正在处理的中断的优先级标志 (0= 正在进行高优先级的中断处理、1= 没有接受中断请求, 或者正在进行低优先级的中断处理。)

图 14-8 中断请求的接受时序（最短时间）



备注 1 个时钟:  $1/f_{\text{CPU}}$  ( $f_{\text{CPU}}$ : CPU 时钟)

图 14-9 中断请求的接受时序（最长时间）



备注 1 个时钟:  $1/f_{\text{CPU}}$  ( $f_{\text{CPU}}$ : CPU 时钟)

#### 14.4.2 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。

### 14.4.3 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 14-5 所示，多重中断的示例如图 14-10 所示。

表 14-5 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 和 PR1L 中的标志。

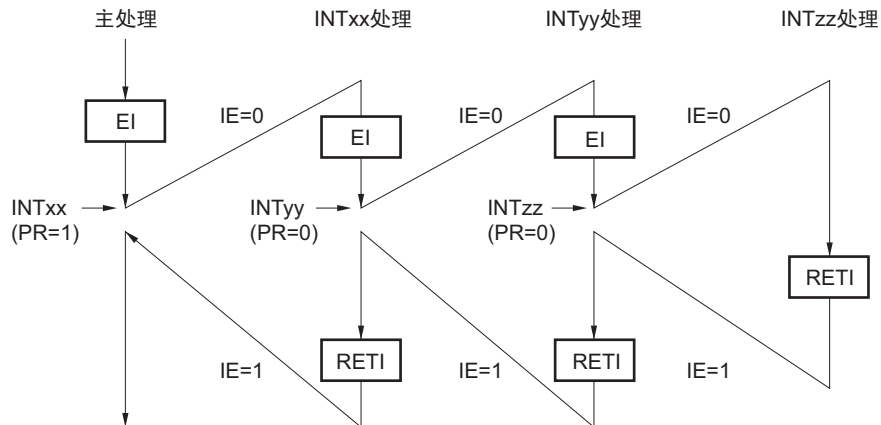
PR=0：高优先级

PR=1：低优先级



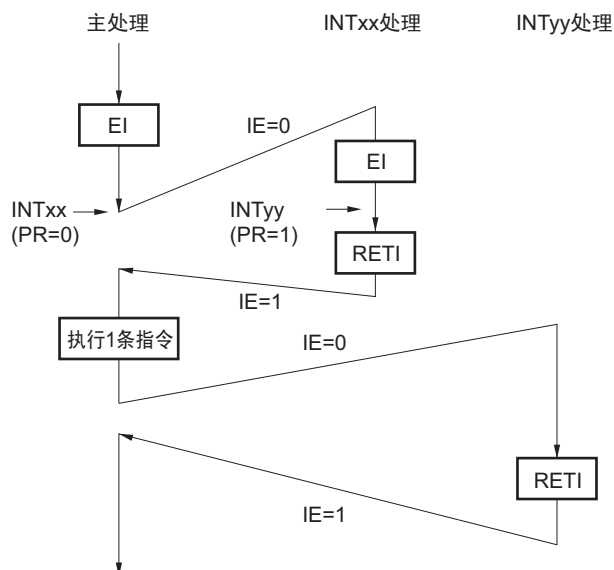
图 14-10 多重中断的示例 (1/2)

## 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。

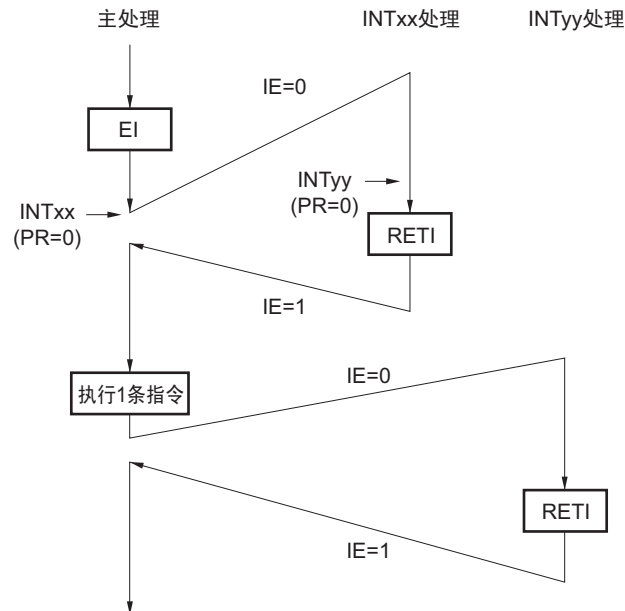
PR=0: 高优先级

PR=1: 低优先级

IE=0: 禁止接受中断请求

图 14-10 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

#### 14.4.4 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、MK0L、MK0H、MK1L、PROL、PROH 和 PR1L 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 14-11 所示。

图 14-11 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. XXPR（中断请求）的运行不受 XXPR（优先级）值的影响。

## 第 15 章 待机功能

### 15.1 待机功能和结构

#### 15.1.1 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 0 (ADM0) 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。
3. 在执行 STOP 指令前，必须将运算放大器设置为停止状态。

### 15.1.2 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（通过 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”时，OSTC 变为“00H”。

图 15-1 振荡稳定时间计数器状态寄存器（OSTC）的格式

地址：FFA3H 复位时：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	
					$f_X=10\text{MHz}$	
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	$\geq 2^{14}/f_X$	$\geq 1.64\mu\text{s}$
1	1	1	1	0	$\geq 2^{15}/f_X$	$\geq 3.27\mu\text{s}$
1	1	1	1	1	$\geq 2^{16}/f_X$	$\geq 6.55\mu\text{s}$

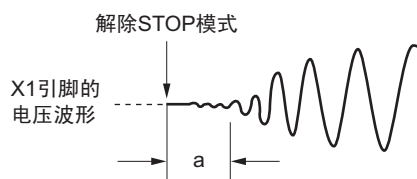
注意 1. 经过上述时间后，从 MOST11 开始逐次置“1”，并且一直保持“1”。

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。

在产生复位信号时，OSTS 为“05H”。

图 15-2 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFA4H 复位时: 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 $\mu\text{s}$
0	1	0	$2^{13}/f_X$	819.2 $\mu\text{s}$
0	1	1	$2^{14}/f_X$	1.64 $\mu\text{s}$
1	0	0	$2^{15}/f_X$	3.27 $\mu\text{s}$
1	0	1	$2^{16}/f_X$	6.55 $\mu\text{s}$
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。

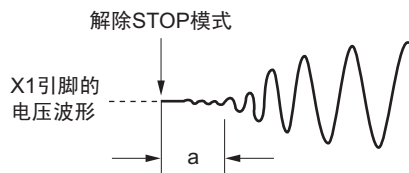
2. 在 X1 时钟的振荡稳定时间内，不能更改 OSTS 寄存器。

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

## 15.2 待机功能的运行

### 15.2.1 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 15-1 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 ( $f_{IH}$ ) 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	$f_{IH}$	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)
	$f_{IL}$	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 51				
8 位定时器 H1				
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		
运算放大器 0、1 注				
比较器注				
串行接口 UART0				
上电清除功能				
低电压检测功能				
外部中断				

注 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x (内置运算放大器 / 比较器的产品)。

备注  $f_{IH}$ : 高速内部振荡时钟  
 $f_{EXCLK}$ : 外部主系统时钟  
 $f_X$ : X1 时钟  
 $f_{IL}$ : 低速内部振荡时钟

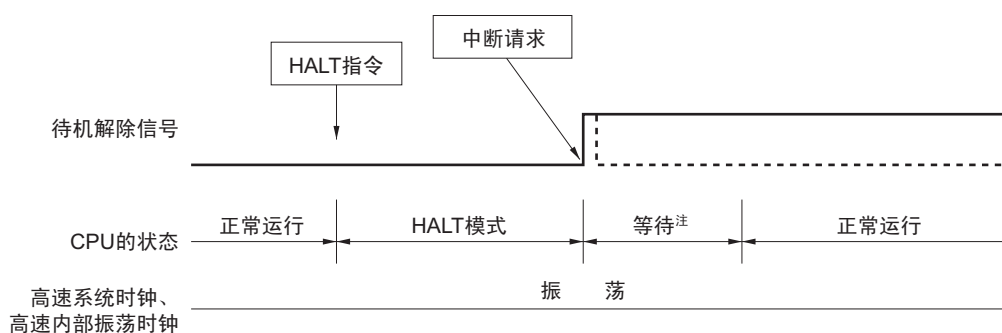
## (2) HALT 模式的解除

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 15-3 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。

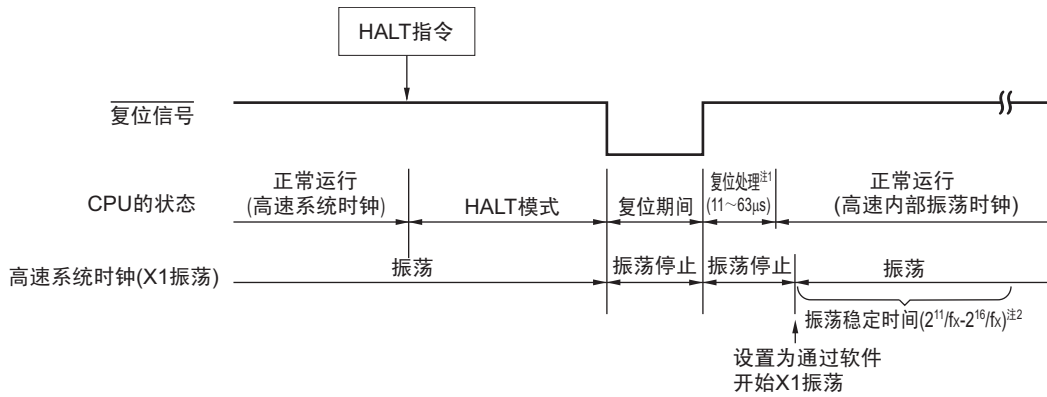
## (b) 通过产生复位信号解除 HALT 模式

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。



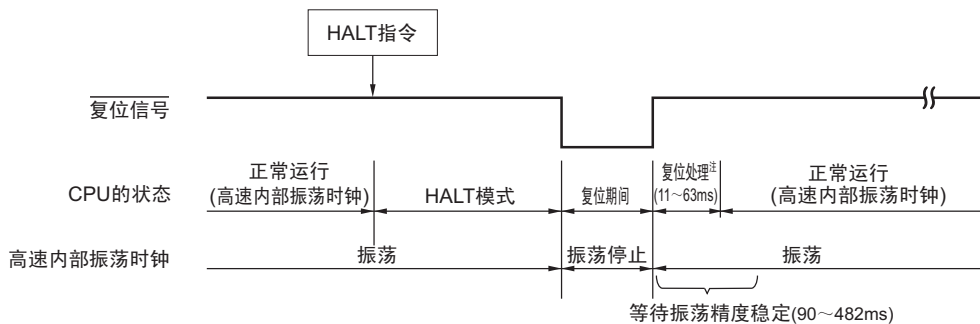
图 15-4 通过 HALT 模式的复位解除

(1) CPU 时钟为高速系统时钟的情况



- 注 1. 在通过 RESET 引脚输入或者 POC 进行复位时，复位处理前需要经过电压稳定等待时间（1.61 ~ 5.27ms）。  
 2. 在外部主系统时钟（f<sub>EXCLK</sub>）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



- 注 在通过 RESET 引脚输入或者 POC 进行复位时，复位处理前需要经过电压稳定等待时间（1.61 ~ 5.27ms）。

表 15-2 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

## 15.2.2 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 15-3 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 ( $f_{IH}$ ) 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	$f_{IH}$	停止运行		
	$f_X$			
	$f_{EXCLK}$	输入无效		
	$f_{IL}$	保持设置 STOP 模式前的状态		
CPU		停止运行		
闪存				
RAM		保持设置 STOP 模式前的状态		
端口（锁存器）				
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 51		只在选择 TI51 作为计数时钟时可运行		
8 位定时器 H1		只在选择 $f_{IL}$ , $f_{IL}/2^7$ , $f_{IL}/2^9$ 作为计数时钟时可运行		
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		
运算放大器 0、1 注		停止运行		
比较器注		禁止运行		
串行接口 UART0		在 8 位定时器 / 事件计数器 51 运行期间，只在选择 TM51 输出作为串行时钟时可运行。		
上电清除功能		可运行		
低电压检测功能				
外部中断				

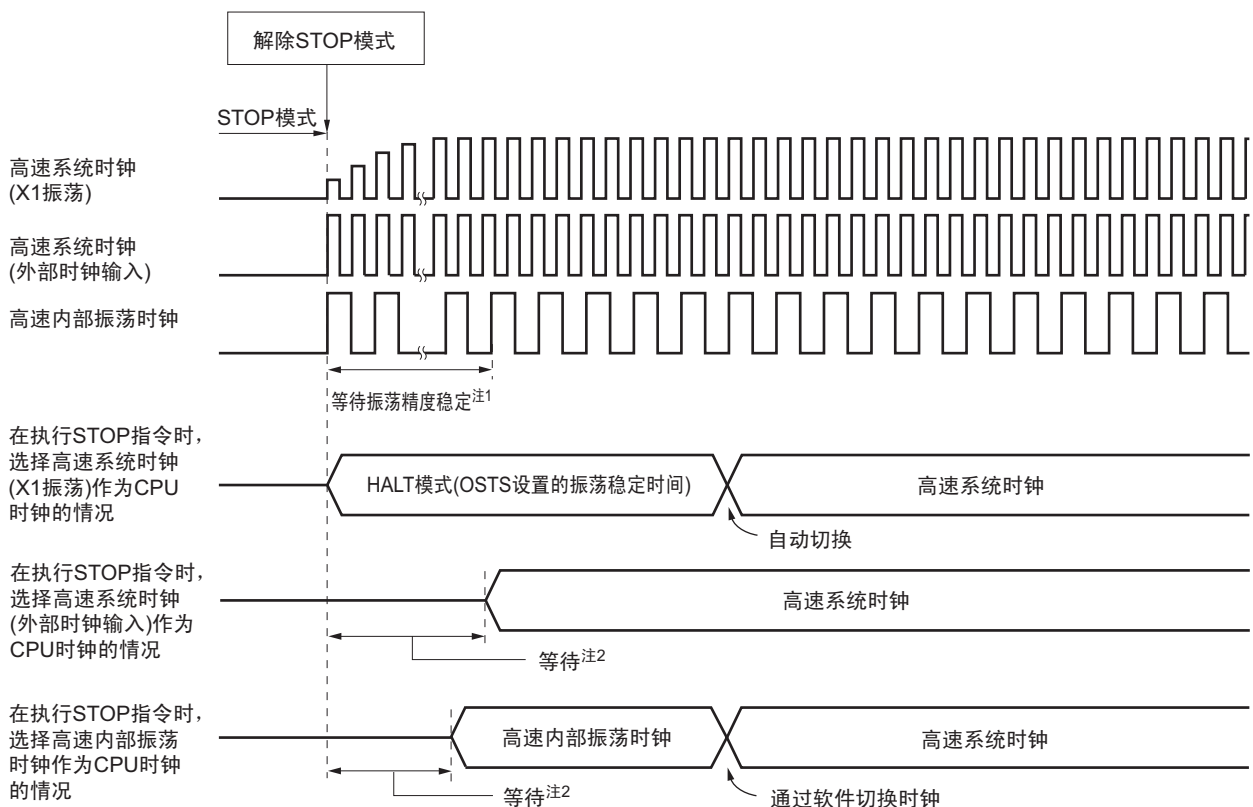
注 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x（内置运算放大器 / 比较器的产品）。

备注  $f_{IH}$ : 高速内部振荡时钟  
 $f_{EXCLK}$ : 外部主系统时钟  
 $f_X$ : X1 时钟  
 $f_{IL}$ : 低速内部振荡时钟

- 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 15-5 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 1. 振荡精确稳定等待时间：90 ~ 482 $\mu$ s
2. 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

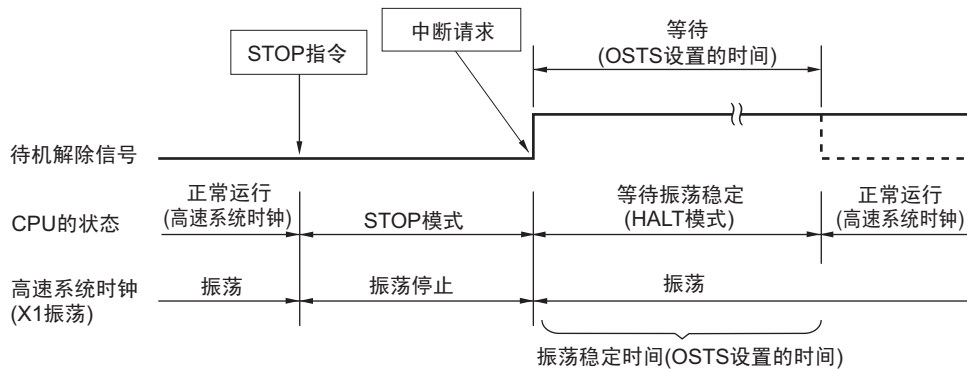
可通过以下两种解除源解除 STOP 模式。

## (a) 通过非屏蔽中断请求解除 STOP 模式

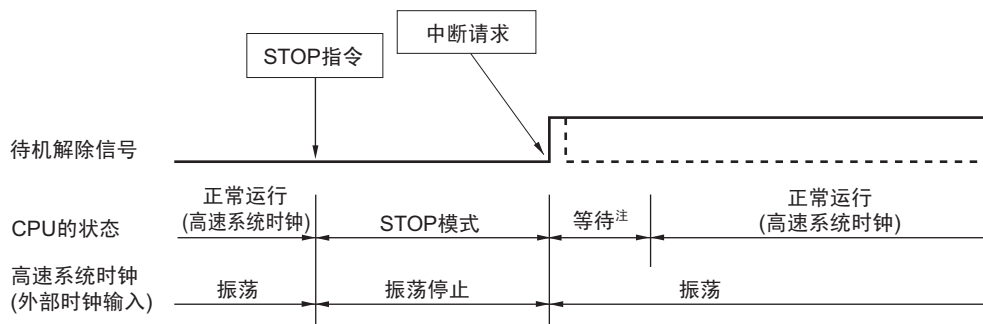
一旦产生非屏蔽中断请求，就解除 STOP 模式。如果为中断接受允许状态，就执行向量中断处理；如果为中断接受禁止状态，就执行下一条地址指令。

图 15-6 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



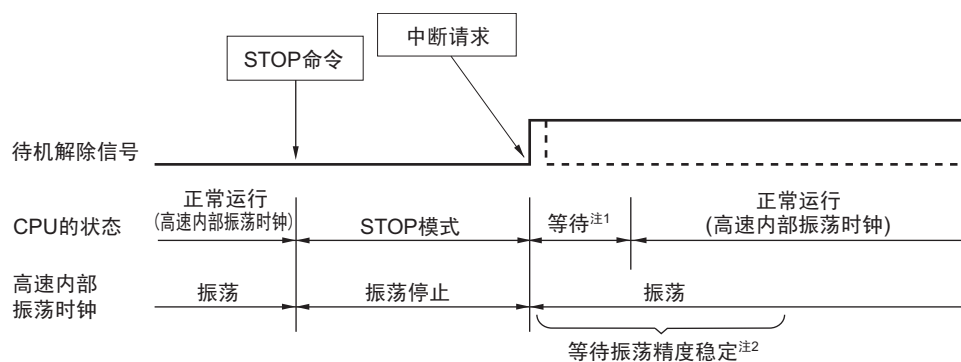
注 待机时间如下所示。

- 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 15-6 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 1. 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

2. 振荡精确稳定等待时间：90 ~ 482 $\mu$ s

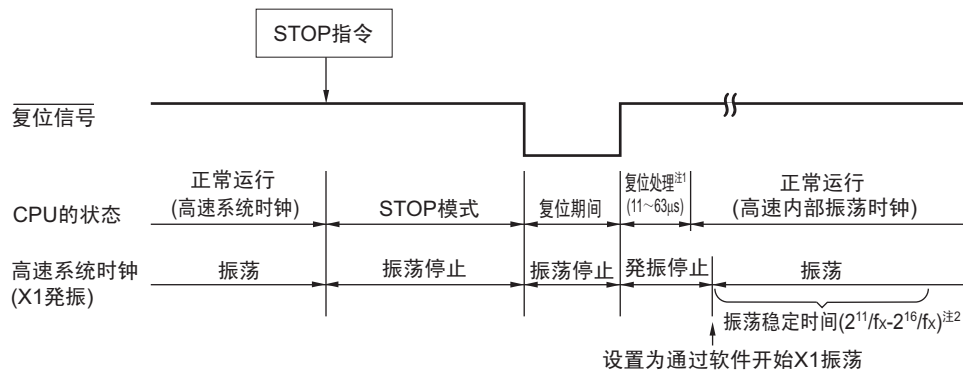
备注 虚线表示接受解除待机的中断请求后的情况。

## (b) 通过产生复位信号解除 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 15-7 通过 STOP 的复位解除

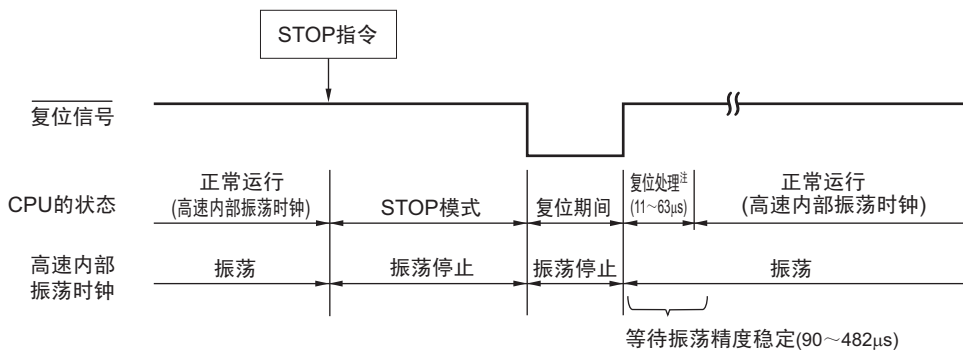
## (1) CPU 时钟为高速系统时钟的情况



- 注 1. 在通过  $\overline{\text{RESET}}$  引脚输入或者 POC 进行复位时，复位处理前需要经过电压稳定等待时间（1.61 ~ 5.27ms）。
2. 在外部主系统时钟（ $f_{\text{EXCLK}}$ ）用作高速系统时钟时，不需要振荡稳定时间。

备注  $f_x$ : X1 时钟振荡频率

## (2) CPU 时钟为高速内部振荡时钟的情况



- 注 在通过  $\overline{\text{RESET}}$  引脚输入或者 POC 进行复位时，复位处理前需要经过电压稳定等待时间（1.61 ~ 5.27ms）。

表 15-4 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 16 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，在产生复位信号时，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 16-1 和表 16-2 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。

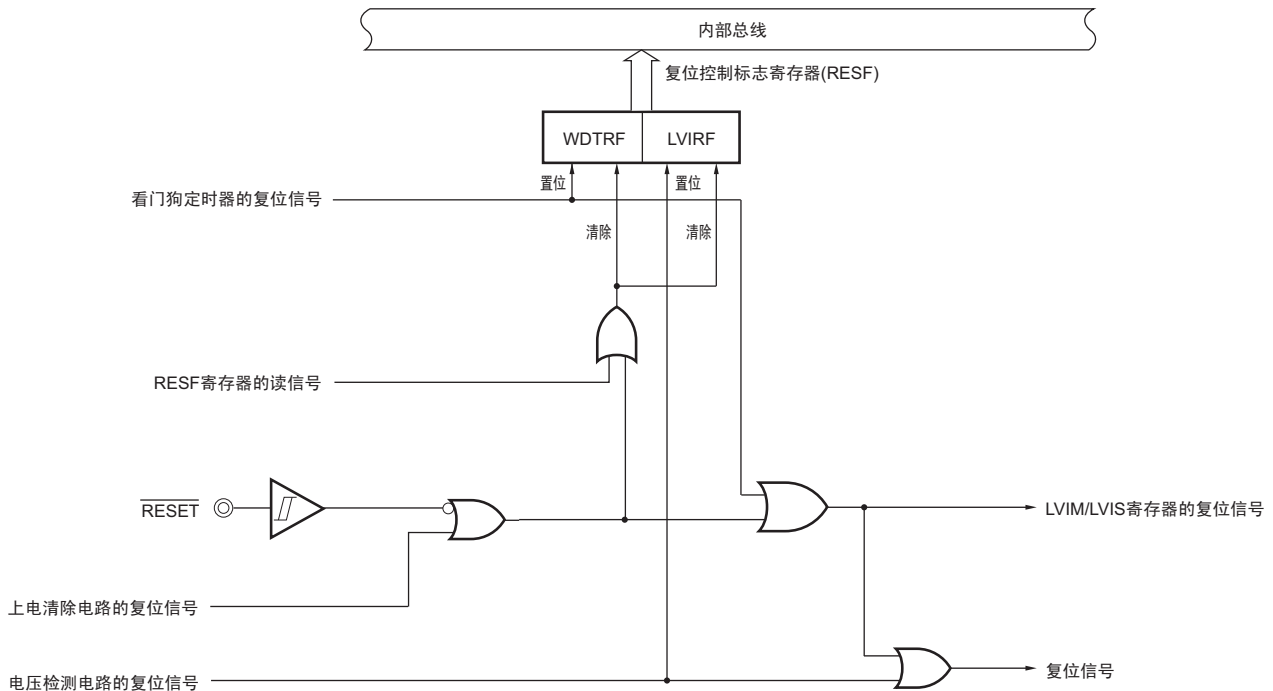
如果向  $\overline{\text{RESET}}$  引脚输入低电平，就发生复位；如果向  $\overline{\text{RESET}}$  引脚输入高电平，就解除复位，并且在复位处理后通过高速内部振荡时钟开始程序的执行。自动解除通过看门狗定时器进行的复位，并且在复位处理后通过高速内部振荡时钟开始程序的执行（参考图 16-2 ~ 图 16-4）。在执行通过 POC 电路和 LVI 电路的电源检测进行的复位后，如果  $V_{DD} \geq V_{POR}$  或者  $V_{DD} \geq V_{LVI}$  时，就解除复位，并且在复位处理后通过高速内部振荡时钟开始执行程序（参照“第 17 章 上电清除电路”、“第 18 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 $\mu$ s 的低电平。

（如果在电源上升时进行外部复位，工作电压范围外（ $V_{DD} < 1.8\text{V}$ ）的期间就不计算在 10 $\mu$ s 内。但是，可在 POC 解除前持续进行低电平输入。）

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 RAM 内容。但是，由于各 SFR 被初始化，所以端口引脚变为高阻抗。

图 16-1 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 16-2 通过 RESET 输入进行复位的时序

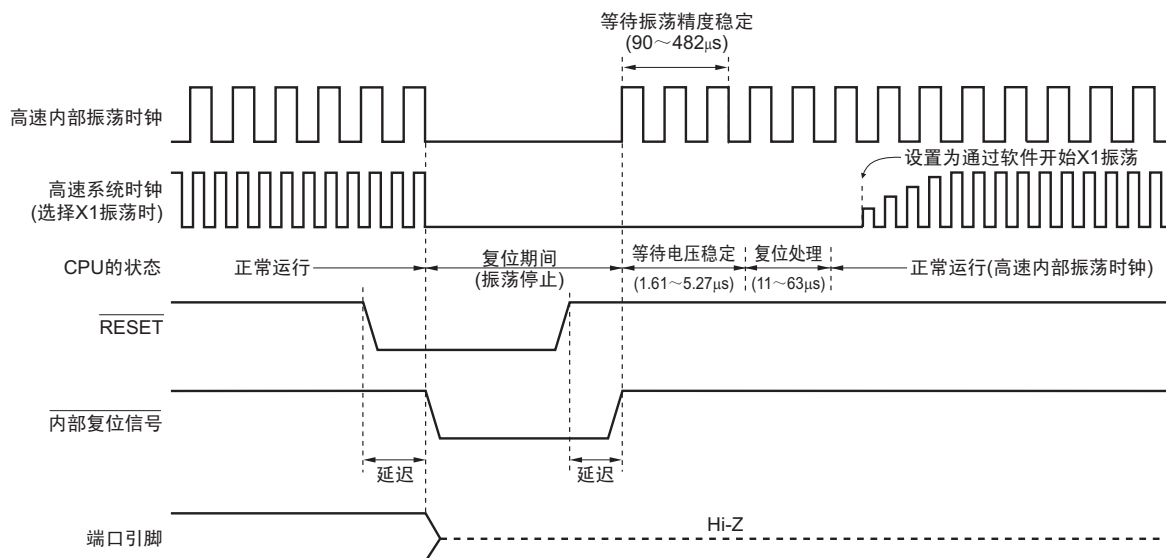
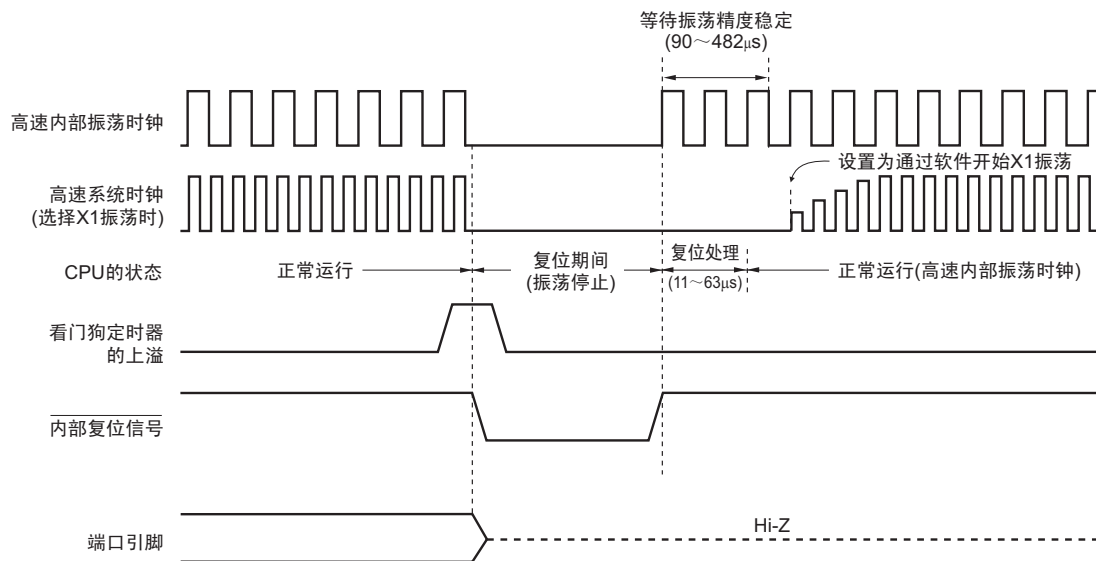
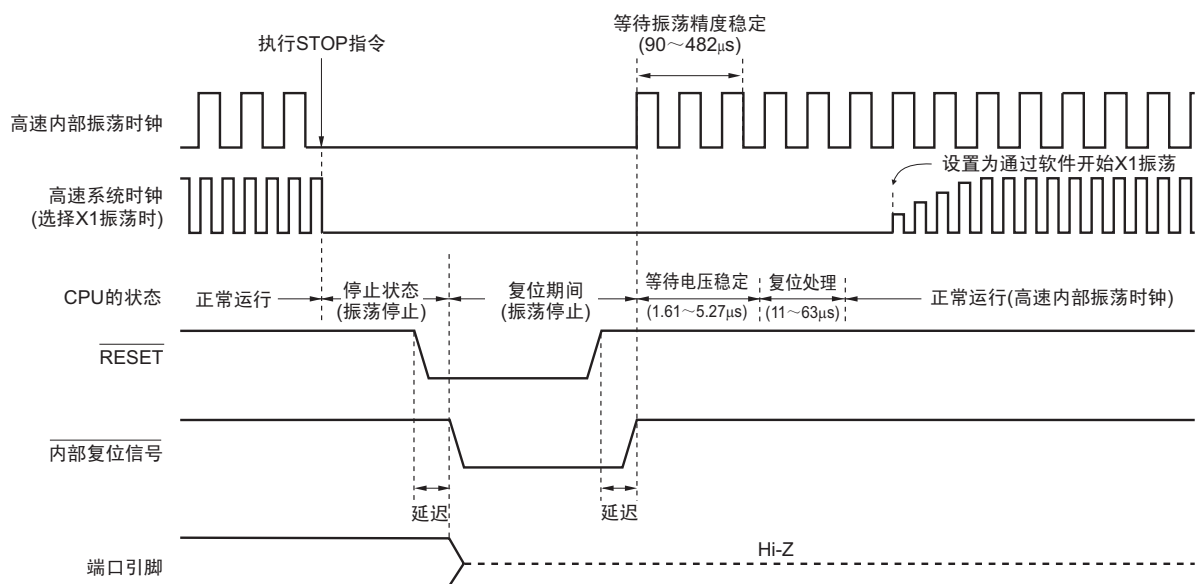




图 16-3 通过看门狗定时器上溢进行的时序



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 16-4 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 17 章 上电清除电路”和“第 18 章 低电压检测电路”。

表 16-1 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	$f_{IH}$	停止运行
	$f_X$	停止运行 (X1 和 X2 引脚为输入端口模式)
	$f_{EXCLK}$	时钟的输入无效 (EXCLK 引脚为输入端口模式)
$f_{IL}$	停止运行	
CPU	停止运行 (但是, 在高于上电清除检测电压时保持值。)	
闪存		
RAM		
端口 (锁存器)		
16 位定时器 / 事件计数器 00		
8 位定时器 / 事件计数器 51		
8 位定时器 H1		
看门狗定时器		
A/D 转换器		
运算放大器 0 (AMP0) 注		
运算放大器 1 (AMP1) 注		
比较器注		
串行接口 UART0		
外部中断		
上电清除功能	可运行	
低电压检测功能	停止运行 (但是, 在 LVI 复位时保持运行)	
片上调试功能	停止运行	

注 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x (内置运算放大器 / 比较器的产品)。

备注  $f_{IH}$ : 高速内部振荡时钟  
 $f_{EXCLK}$ : 外部主系统时钟  
 $f_X$ : X1 时钟  
 $f_{IL}$ : 低速内部振荡时钟

表 16-2 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 2、3、12 (P2、P3、P12) (输出锁存器)		00H
端口模式寄存器 2、3、12 (PM2、PM3、PM12)		FFH
上拉电阻选择寄存器 3 (PU3)		00H
上拉电阻选择寄存器 12 (PU12)		20H
复位引脚模式寄存器 (RSTMASK)		00H
端口复用切换控制寄存器 (MUXSEL) 注 3		00H
存储容量切换寄存器 (IMS)		CFH 注 4

- 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. 只限 16 引脚产品。
3. 因为复位时的 ROM 区域设置为不定，所以在复位解除后，必须给各产品设置如下所示的值。

产品	IMS	ROM 容量	内部高速 RAM 容量
R7F0C3004、R7F0C3034、R7F0C3006、R7F0C3036、 R7F0C3064、R7F0C3094、R7F0C3066、R7F0C3096	41H	4K 字节	512 字节
R7F0C3014、R7F0C3044、R7F0C3016、R7F0C3046、 R7F0C3074、R7F0C3104、R7F0C3076、R7F0C3106、	42H	8K 字节	512 字节
R7F0C3024、R7F0C3054、R7F0C3026、R7F0C3056、 R7F0C3084、R7F0C3114、R7F0C3086、R7F0C3116	04H	16K 字节	768 字节

表 16-2 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 51	定时器计数器 51 (TM51)	00H
	比较寄存器 51 (CR51)	00H
	定时器时钟选择寄存器 51 (TCL51)	00H
	模式控制寄存器 51 (TMC51)	00H
8 位定时器 H1	比较寄存器 01、11 (CMP01、CMP11)	00H
	模式寄存器 (TMHMD1)	00H
	载波控制寄存器 1 (TMCYC1)	00H
看门狗定时器	允许寄存器 (WDTE)	1AH/9AH 注 2
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 H (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
运算放大器 0 (AMP0) 注 3	运算放大器控制寄存器 (AMPM)	00H
运算放大器 1 (AMP1) 注 3		
比较器注 3	比较器控制寄存器 (CMPCTL)	00H
	比较器端口配置寄存器 (CMPPC) 注 4	00H
串行接口 UART0	接收缓冲寄存器 0 (RXB0)	FFH
	发送缓冲寄存器 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. WDTE 的复位值取决于选项字节的设置。
3. 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x (内置运算放大器 / 比较器的产品)。
4. CMPPC 寄存器对所有产品都公开。

表 16-2 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
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 (IF0L、IF0H、IF1L)	00H
	屏蔽标志寄存器 0L、0H、1L (MK0L、MK0H、MK1L)	FFH
	优先级指定标志寄存器 0L、0H、1L (PR0L、PR0H、PR1L)	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 (通过 LVI 默认启动功能进行的复位除外)	通过 LVI 默认启动功能进行的复位
RESF	WDTRF 标志	清除 (0)	清除 (0)	设置 (1)	保持	清除 (0)
	LVIRF 标志			保持	设置 (1)	
LVIM		清除 (00H)	清除 (00H)	清除 (00H)	保持	清除 (00H)
LVIS						

## 16.1 复位源的确认寄存器

R7F0C30x、R7F0C31x 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。

在通过 RESET 输入、上电清除（POC）电路进行的复位和读 RESF 的数据时，RESF 为“00H”。

图 16-5 复位控制标志寄存器（RESF）的格式

地址：FFACH 复位时：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 16-3 所示。

表 16-3 产生复位请求时的 RESF 状态。

标志 \ 复位源	RESET 输入	POC 复位	WDT 复位	LVI 复位 (通过 LVI 默认启动功能进行的复位除外)	通过 LVI 默认启动功能进行的复位
WDTRF	清除 (0)	清除 (0)	置位 (1)	保持	清除 (0)
LVIRF			保持	置位 (1)	

## 第 17 章 上电清除电路

### 17.1 上电清除电路的功能

上电清除电路（POC）具有以下功能。

- (1) 设置为LVI默认启动功能停止时（选项字节：LVISTART=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LVI默认启动功能运行时（选项字节：LVISTART=1）
  - 在接通电源时产生内部复位信号，并且在电源电压（ $V_{DD}$ ）大于接通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。

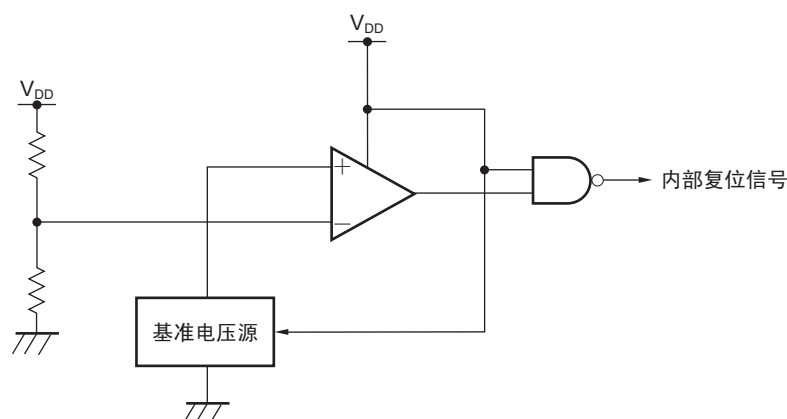
**注意** 在 POC 电路中产生内部复位信号时，复位控制标志寄存器（RESF）清“00H”。

**备注** R7F0C30x、R7F0C31x 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（00H）RESF，但是 RESF 标志被置“1”。有关 RESF 的详细内容，请参照“第 16 章 复位功能”。

### 17.2 上电清除电路的结构

上电清除电路的框图如图 17-1 所示。

图 17-1 上电清除电路的框图



### 17.3 上电清除电路的运行

- 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。

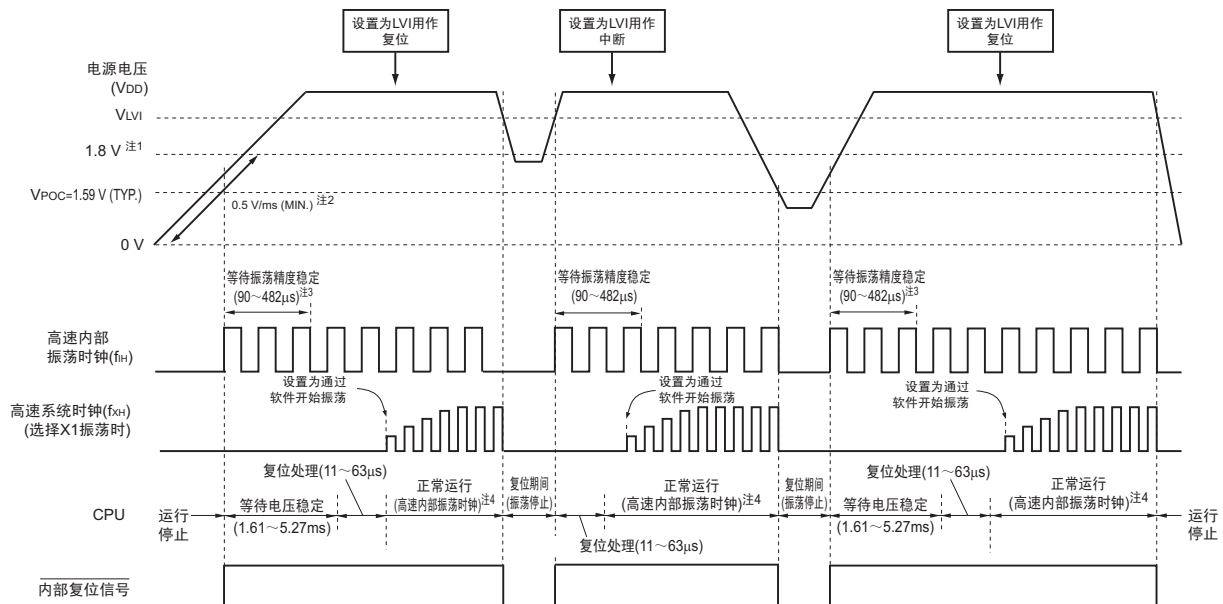
**注意** 如果通过选项字节设置为低电压检测 (LVI) 电路默认启动功能运行，在电源电压 ( $V_{DD}$ ) 超过  $2.7V\pm 0.1V$  前不能解除复位。

- 比较电源电压 ( $V_{DD}$ ) 和检测电压 ( $V_{PDR}=1.59V\pm 0.15V$ )，在  $V_{DD} < V_{PDR}$  时，产生内部复位信号。

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 17-2 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) 电源上升时的 LVI 为 OFF 的情况 (选项字节: LVISTART=0)



- 注**
- 运行保证范围为  $1.8V \leq V_{DD} \leq 5.5V$ 。如果要在电源下降时将“小于 1.8V”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。
  - 如果从接通电源到上升到 1.8V 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。
  - 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。
  - 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。

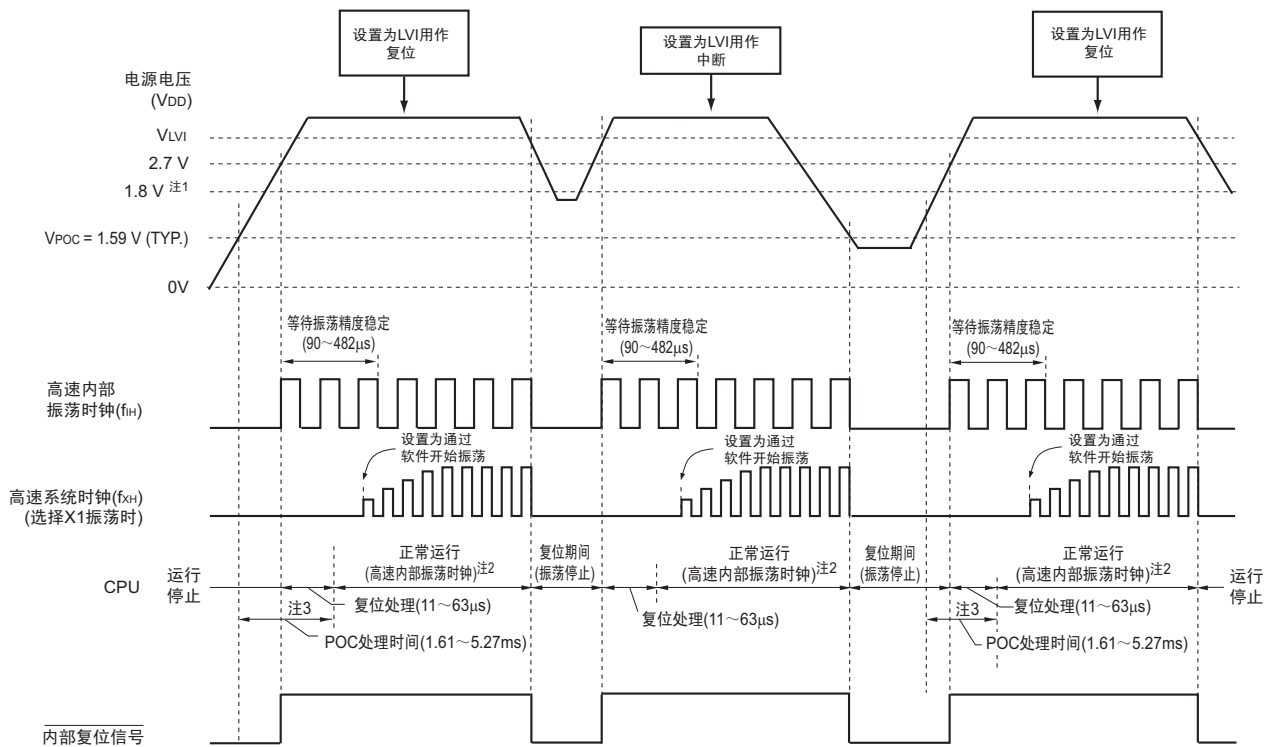
**注意** 必须在复位解除后通过软件设置低电压检测电路 (参照“第 18 章 低电压检测电路”)。

**备注**  $V_{LVI}$ : LVI 检测电压  
 $V_{POC}$ : 检测电压



图 17-2 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) 电源上升时的 LVI 为 ON 的情况 (选项字节: LVISTART=1)



- 注 1. 运行保证范围为  $1.8V \leq V_{DD} \leq 5.5V$ 。如果要在电源下降时将“小于 1.8V”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  $\overline{\text{RESET}}$  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。
3. 从达到检测电压 (1.59V(TYP.)) 到开始正常运行所需的时间如下所示。
- 从 1.59V(TYP.) 到达到 2.7V(TYP.) 的时间 < 5.27ms 时：
    - 从 1.59V(TYP.) 到正常运行大约需要 1.6 ~ 5.3ms 的 POC 处理时间。
  - 从 1.59V(TYP.) 到达到 2.7V(TYP.) 的时间 > 5.27ms 时：
    - 从 2.7V(TYP.) 到正常运行大约需要 11 ~ 63μs 的 POC 处理时间。

注意 必须在复位解除后通过软件更改低电压检测电路的默认设置 (参照“第 18 章 低电压检测电路”)。

备注 V<sub>LV1</sub>: LVI 检测电压  
V<sub>POC</sub>: 检测电压

## 17.4 上电清除电路的注意事项

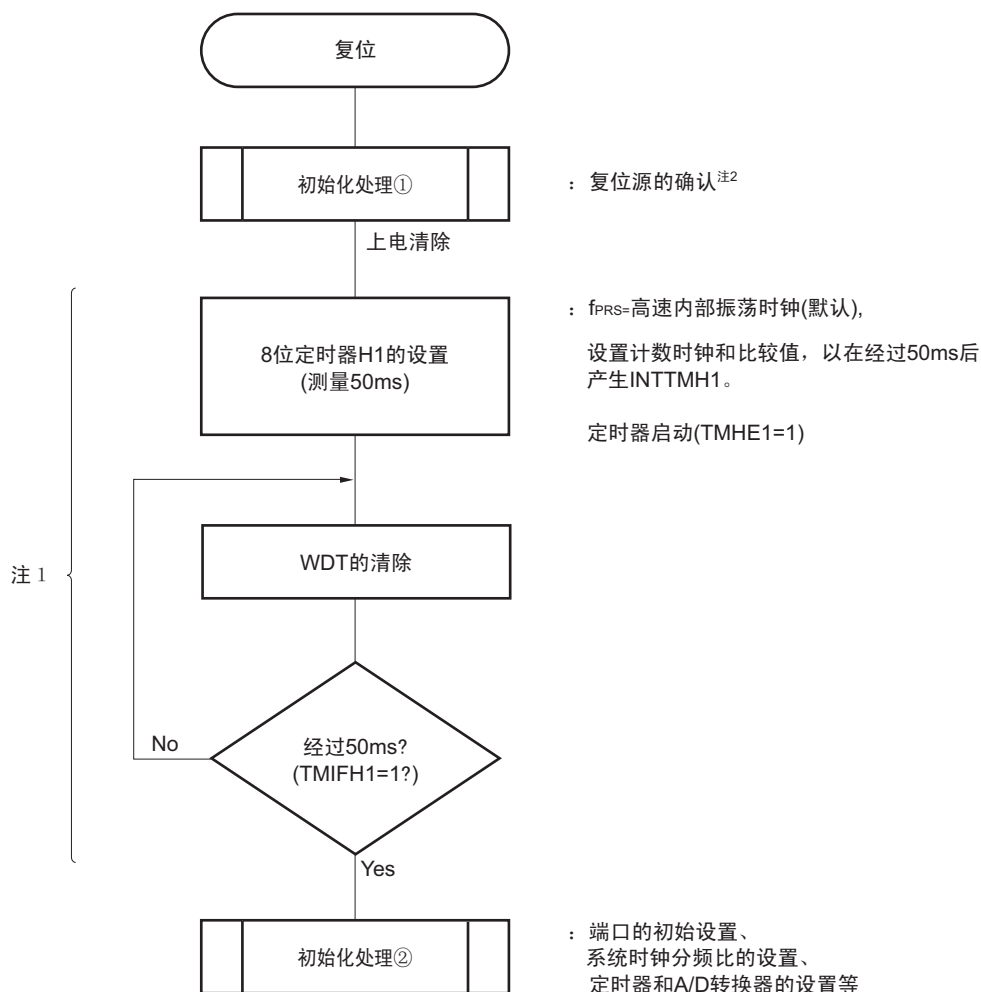
在因电源电压 ( $V_{DD}$ ) 接近检测电压 ( $V_{POC}$ ) 导致某期间出现波动的系统中, 可能会重复出现复位状态 / 复位解除状态。可通过以下措施任意设置从复位解除到单片机运行的时间。

### < 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 17-3 复位解除后的软件处理示例 (1/2)

- 接近检测电压的电源电压波动为 50ms 以下时

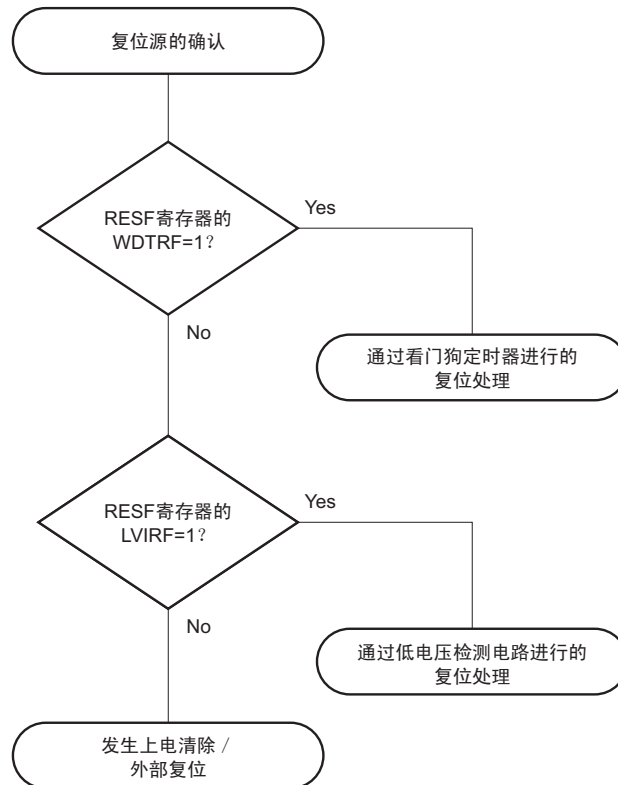


备注 1. 如果在此期间再次发生复位, 就不转移到初始化处理②。

- 流程图如下一页所示。

图 17-3 复位解除后的软件处理示例 (2/2)

- 复位源的确认



## 第 18 章 低电压检测电路

### 18.1 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 LVI 检测电压 ( $V_{LVI}$ )，产生内部复位或者内部中断信号。
- 可通过选项字节设置为低电压检测电路 (LVI) 默认启动功能运行。如果设置为低电压检测电路 (LVI) 默认启动功能运行，并且电源从低于检测电压 ( $V_{POC}=1.59V(TYP.)$ ) 的电压开始上升，就在电源电压 ( $V_{DD}$ ) 小于 LVI 检测电压 ( $V_{LVI}=2.7V\pm 0.1V$ ) 时产生内部复位信号。此后，当电源电压 ( $V_{DD}$ ) 小于 LVI 检测电压 ( $V_{LVI}=2.7V\pm 0.1V$ ) 时，就产生复位信号。
- 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 16 级检测电平中选择电源电压的 LVI 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。

备注 LVISEL: 低电压检测寄存器 (LVIM) 的 bit2

LVIMD: LVIM 的 bit1

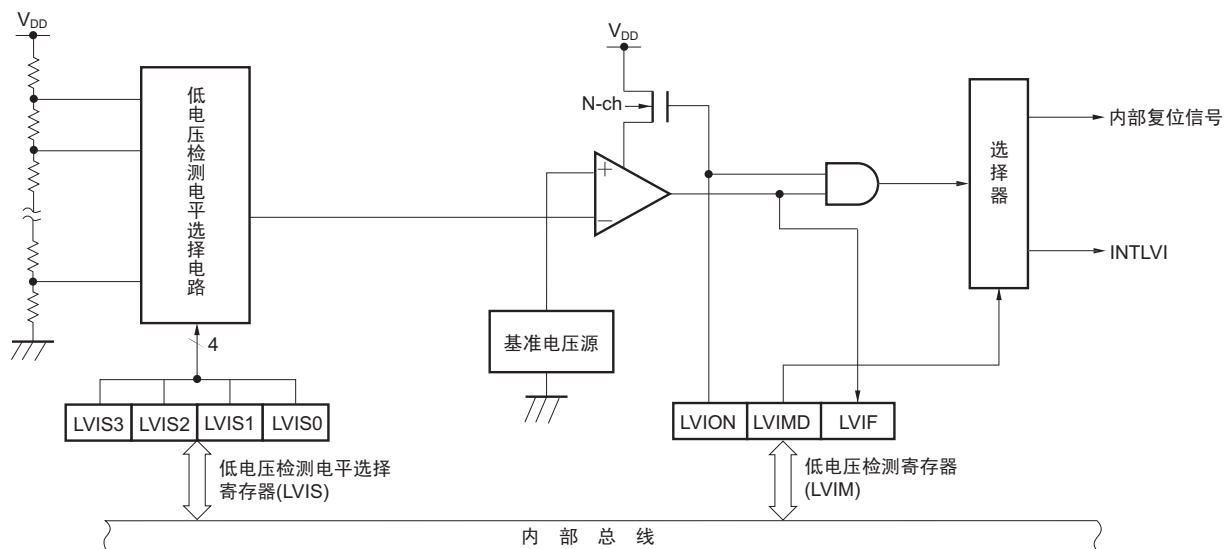
在低电压检测电路运行期间，可通过读低电压检测标志 (LVIF: LVIM 的 bit0)，确认电源电压或者外部输入引脚的输入电压是大于等于检测电平还是小于检测电平。

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 16 章 复位功能”。

## 18.2 低电压检测电路的结构

低电压检测电路的框图如图 18-1 所示。

图 18-1 低电压检测电路的框图



### 18.3 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)

#### (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。

在产生 LVI 复位以外的复位信号时，LVIM 为“00H”。

图 18-2 低电压检测寄存器 (LVIM) 的格式

地址: FFBEH 复位时: 00H 注 1 R/W 注 2

符号	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVION	0	0	0	0	0	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

LVIMD 注 3	选择低电压检测的运行模式 (中断 / 复位)
0	在电压下降至电源电压 ( $V_{DD}$ ) 小于 LVI 检测电压 ( $V_{LVI}$ ) 或者电压上升至电源电压 ( $V_{DD}$ ) 大于等于 LVI 检测电压 ( $V_{LVI}$ ) 时, 产生内部中断信号。
1	在电源电压 ( $V_{DD}$ ) 小于 LVI 检测电压 ( $V_{LVI}$ ) 时, 发生内部复位; 在电源电压 ( $V_{DD}$ ) 大于等于 LVI 检测电压 ( $V_{LVI}$ ) 时, 解除内部复位。

LVIF	低电压检测标志
0	电源电压 ( $V_{DD}$ ) $\geq$ LVI 检测电压 ( $V_{LVI}$ ) 或者禁止运行时
1	电源电压 ( $V_{DD}$ ) $<$ LVI 检测电压 ( $V_{LVI}$ )

注 1. 复位值根据复位源和选项字节的设置而变。

在 LVI 复位 (通过 LVI 默认启动功能进行的复位除外) 时, 该寄存器不被清“00H”。

在其他复位时, 该寄存器清“00H”。

2. bit0 为只读位。

3. LVION 和 LVIMD 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 $\mu$ s(MAX.))。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 $\mu$ s 的外部输入 (最小脉宽: 200 $\mu$ 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. 如果在中断模式 (LVIMD=0) 中使用 LVI, 并且设置为在电源电压 ( $V_{DD}$ ) 小于等于检测电压 ( $V_{LVI}$ ) 时禁止 LVI 运行 (清除 LVION), 就有可能产生中断请求信号 (INTLVI), 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。

在产生 LVI 复位以外的复位信号时，LVIS 为“00H”。

图 18-3 低电压检测电平选择寄存器 (LVIS) 的格式

地址：FFBFH 复位时：00H<sup>注</sup> R/W

符号	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	检测电平
0	0	0	0	$V_{LVI0} (4.24 \pm 0.1V)$
0	0	0	1	$V_{LVI1} (4.09 \pm 0.1V)$
0	0	1	0	$V_{LVI2} (3.93 \pm 0.1V)$
0	0	1	1	$V_{LVI3} (3.78 \pm 0.1V)$
0	1	0	0	$V_{LVI4} (3.62 \pm 0.1V)$
0	1	0	1	$V_{LVI5} (3.47 \pm 0.1V)$
0	1	1	0	$V_{LVI6} (3.32 \pm 0.1V)$
0	1	1	1	$V_{LVI7} (3.16 \pm 0.1V)$
1	0	0	0	$V_{LVI8} (3.01 \pm 0.1V)$
1	0	0	1	$V_{LVI9} (2.85 \pm 0.1V)$
1	0	1	0	$V_{LVI10} (2.70 \pm 0.1V)$
1	0	1	1	$V_{LVI10} (2.55 \pm 0.1V)$
1	1	0	0	$V_{LVI10} (2.39 \pm 0.1V)$
1	1	0	1	$V_{LVI10} (2.24 \pm 0.1V)$
1	1	1	0	$V_{LVI10} (2.08 \pm 0.1V)$
1	1	1	1	$V_{LVI10} (1.93 \pm 0.1V)$

注 复位值因复位源而变。

在发生 LVI 复位（通过 LVI 默认启动功能进行的复位除外）时，LVIS 寄存器的值不被复位，保持当前的值。在其他复位时，LVIS 寄存器的值被复位为“00H”。

注意 1. 必须将 bit4 ~ bit7 置“0”。

2. 在 LVI 运行期间，不能更改 LVIS 的值。

## 18.4 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)

- 比较电源电压 ( $V_{DD}$ ) 和 LVI 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。

**备注** 可通过选项字节设置为低电压检测电路 (LVI) 默认启动功能运行。如果设置为低电压检测电路 (LVI) 默认启动功能运行，并且电源从低于检测电压 ( $V_{POC}=1.59V(TYP.)$ ) 的电压开始上升，就在电源电压 ( $V_{DD}$ ) 小于 LVI 检测电压 ( $V_{LVI}=2.7V\pm 0.1V$ ) 时产生内部复位信号。

### (2) 用作中断 (LVIMD=0)

- 比较电源电压 ( $V_{DD}$ ) 和 LVI 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。

### 18.4.1 用作复位时的设置

#### (1) 设置为 LVI 默认启动功能停止时 (LVISTART=0)

- 开始运行时
  - 屏蔽 LVI 的中断 (LVIMK=1)。
  - 通过低电压检测电平选择寄存器 (LVIS) 的 bit3 ~ bit0 (LVIS3 ~ LVIS0) 设置 LVI 检测电压。
  - 将 LVIM 的 bit7 (LVION) 置“1” (允许 LVI 运行)。
  - 通过软件等待运行稳定时间 (10 $\mu$ s(MAX.))。
  - 保持等待直到通过 LVIM 的 bit0 确认“电源电压 ( $V_{DD}$ ) 大于等于 LVI 检测电压 ( $V_{LVI}$ )”。
  - 将 LVIM 的 bit1 (LVIMD) 置“1” (在电平检测时发生复位)。

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 18-4 所示。

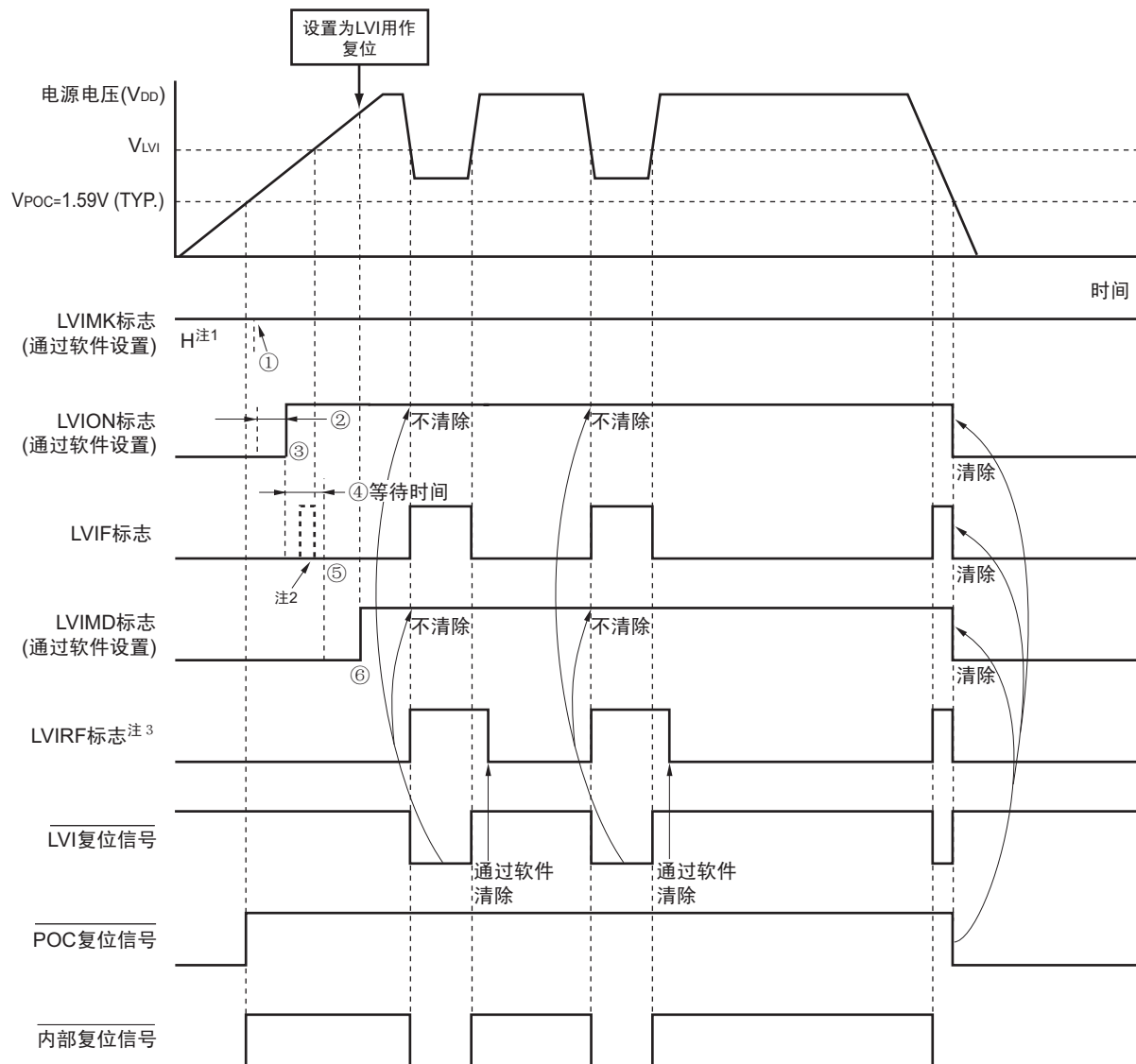
**注意 1.** 必须执行①。如果 LVIMK 为“0”，有可能在执行③的处理后发生中断。

**2.** 在 LVIMD 为“1”时，如果电源电压 ( $V_{DD}$ ) 大于等于 LVI 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 18-4 内部复位信号的产生时序 (LVISTART=0)



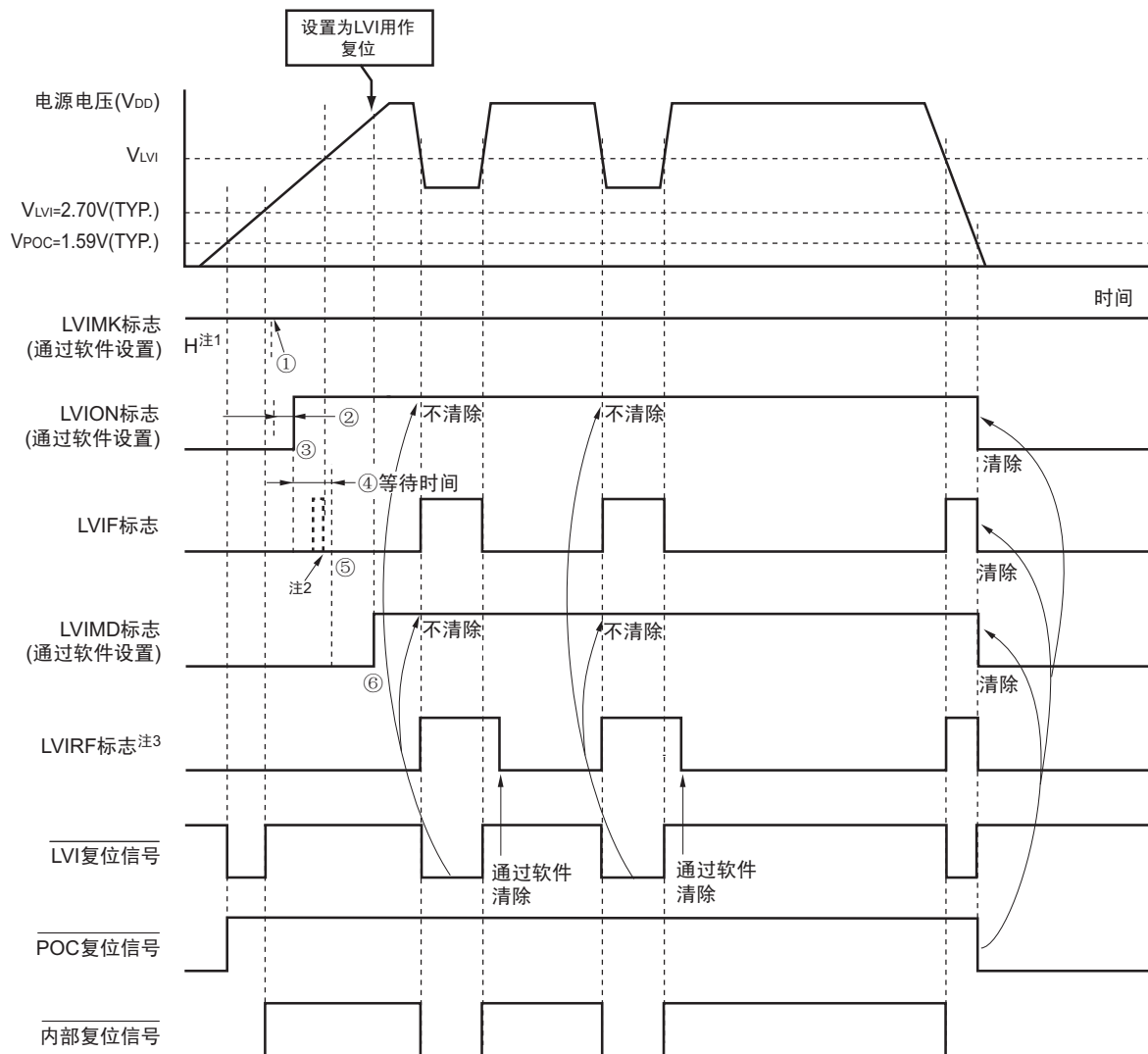
- 注 1. 通过产生复位信号，使 LVIMK 标志置“1”。
2. 中断请求标志寄存器的 LVIF 标志和 LVIMD 标志可能会置“1”。
3. LVIRF 是复位控制标志寄存器 (RESF) 的 bit0。有关 RESF 的详细内容，请参照“第 16 章 复位功能”。

- 备注 1. 图 18-4 中的①~⑥，与“18.4.1 (1) 设置为 LVI 默认启动功能停止时 (LVISTART=0)”中“开始运行时”的步骤①~⑥对应。
2. V<sub>POC</sub>: 检测电压

## (2) 设置为 LVI 默认启动功能运行时 (LVISTART=1)

开始运行以及停止运行时的设置与“18.4.1 (1) 设置为 LVI 默认启动功能停止时 (LVISTART=0)”相同。

图 18-5 内部复位信号的产生时序 (LVISTART=1)



- 注 1. 通过产生复位信号，使 LVIMK 标志置“1”。
2. 中断请求标志寄存器的 LVIF 标志和 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 16 章 复位功能”。

- 备注 1. 图 18-5 中的①~⑥，与“18.4.1 (1) 设置为 LVI 默认启动功能停止时 (LVISTART=0)”中“开始运行时”的步骤①~⑥对应。
2.  $V_{POC}$ : 检测电压

## 18.4.2 用作中断时的设置

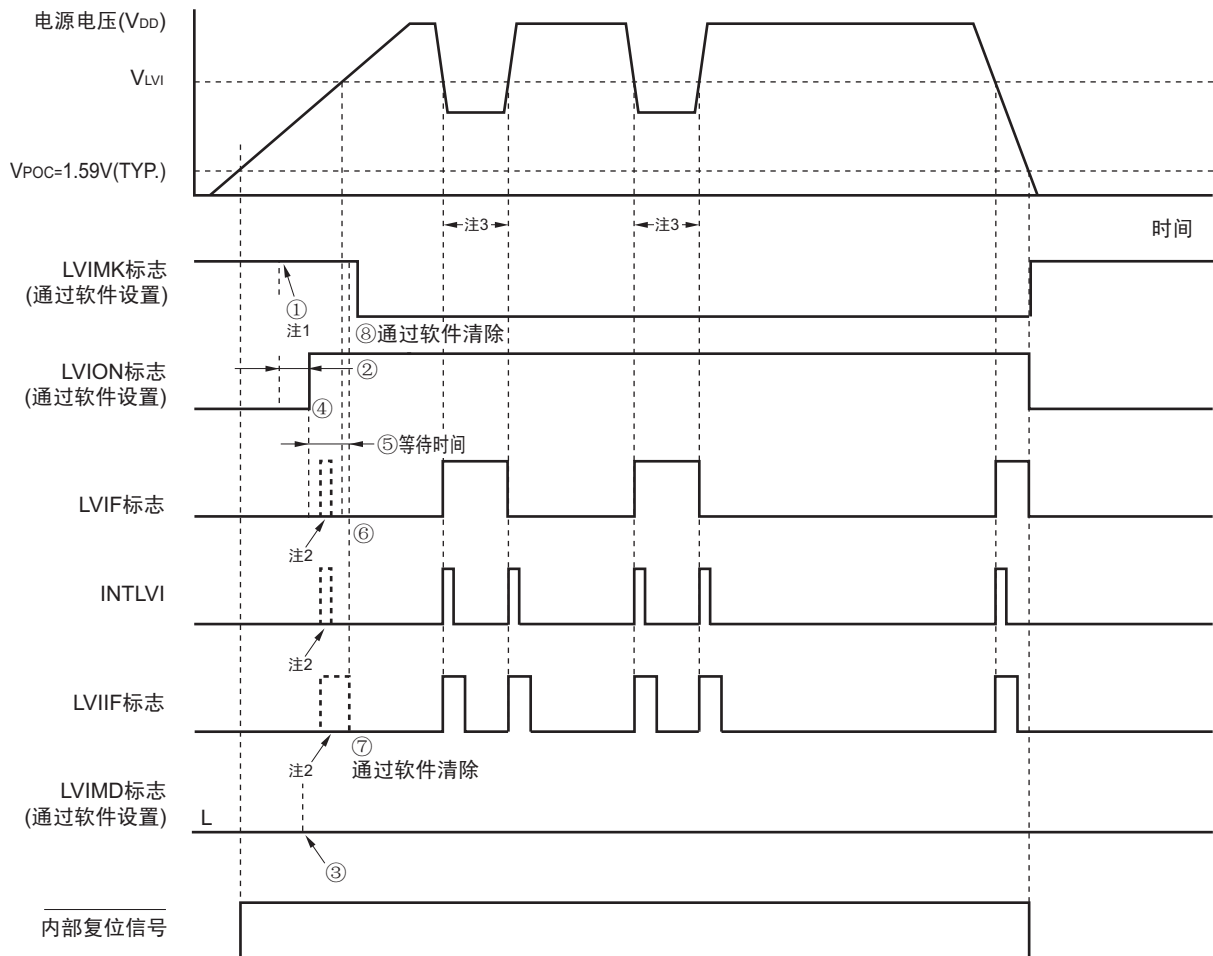
### (1) 设置为 LVI 默认启动功能停止时 (LVISTART=0)

- 开始运行时
  - ① 屏蔽LVI的中断 (LVIMK=1)。
  - ② 通过低电压检测电平选择寄存器 (LVIS) 的 bit3 ~ bit0 (LVIS3 ~ LVIS0) 设置LVI检测电压。
  - 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 $\mu$ s(MAX.))。
  - ⑥ 通过LVIM的 bit0 (LVIF) 确认在检测到下降沿时“电源电压 ( $V_{DD}$ ) 大于等于LVI检测电压 ( $V_{LVI}$ )”，在检测到上升沿时“电源电压 ( $V_{DD}$ ) 小于LVI检测电压 ( $V_{LVI}$ )”。
  - 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 18-6 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 18-6 中断信号的产生时序 (LVISTART=0)



- 注 1. 通过产生复位信号，使 LVIMK 标志置“1”。
2. 可能产生中断请求信号 (INTLVI)，并且 LVIIIF 标志和 LVIF 标志可能置“1”。
3. 如果在电源电压 (V<sub>DD</sub>) 小于等于 LVI 检测电压 (V<sub>LVI</sub>) 时禁止 LVI 运行 (清除 LVION)，就可能产生中断请求信号 (INTLVI)，并且 LVIIIF 可能置“1”。

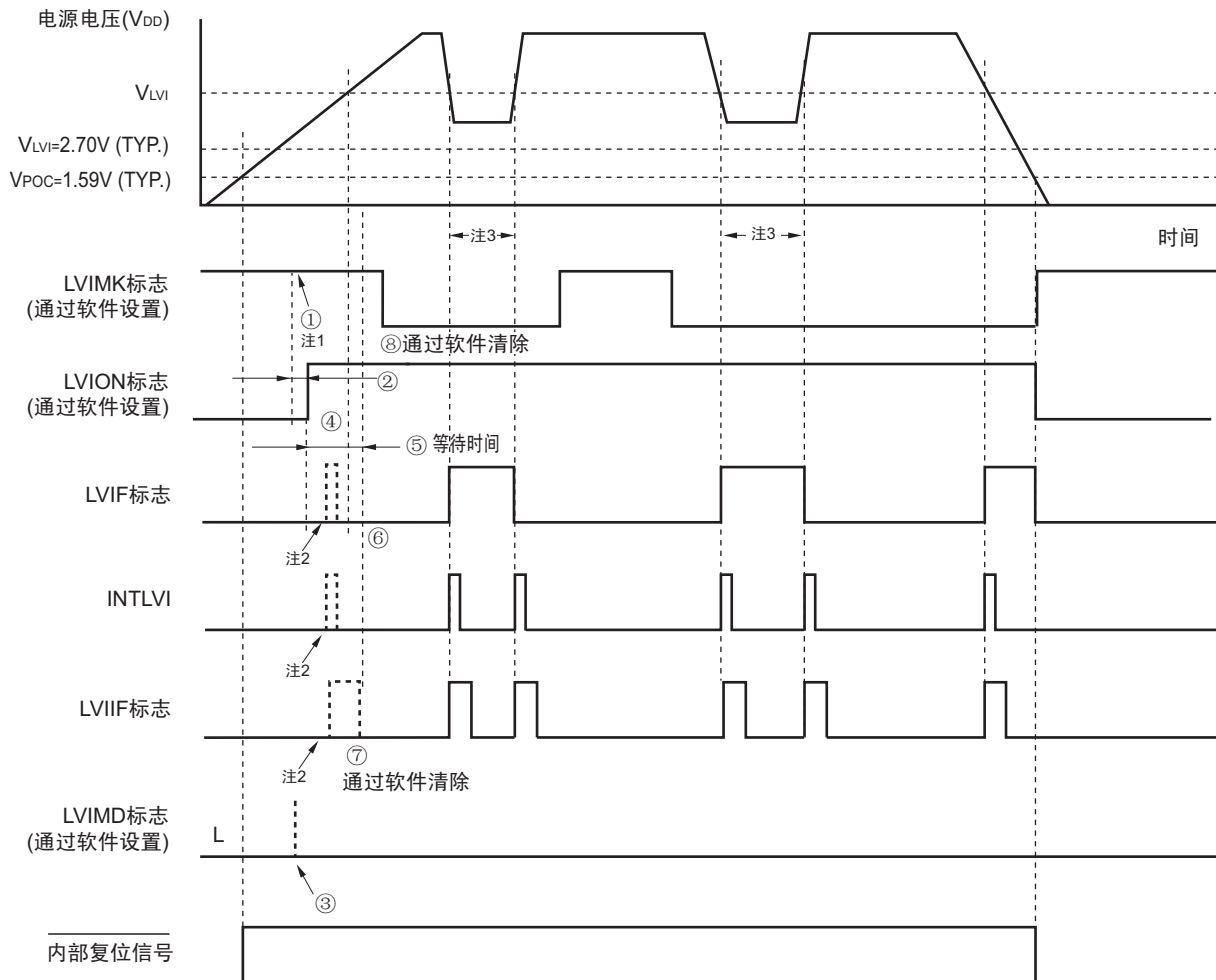
备注 1. 图 18-6 中的①~⑧，与“18.4.2 (1) 设置为 LVI 默认启动功能停止时 (LVISTART=0)”中“开始运行时”的步骤①~⑧对应。

2. V<sub>POC</sub>: 检测电压

## (2) 设置为 LVI 默认启动功能运行时 (LVISTART=1)

开始运行以及停止运行时的设置与“18.4.2 (1) 设置为 LVI 默认启动功能停止时 (LVISTART=0)”相同。

图 18-7 中断信号的产生时序 (LVISTART=1)



- 注 1. 通过产生复位信号，使 LVIMK 标志置“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. 如果在电源电压 (V<sub>DD</sub>) 小于等于 LVI 检测电压 (V<sub>LVI</sub>) 时禁止 LVI 运行 (清除 LVION)，就产生中断请求信号 (INTLVI)，并且 LVIIF 可能置“1”。

备注 1. 图 18-7 中的①~⑧，与“18.4.2 (1) 设置为 LVI 默认启动功能停止时 (LVISTART=0)”中“开始运行时”的步骤①~⑧对应。

2. V<sub>POC</sub>: 检测电压

## 18.5 低电压检测电路的注意事项

在因电源电压 ( $V_{DD}$ ) 接近 LVI 检测电压 ( $V_{LVI}$ ) 导致某期间出现波动的系统中, 根据低电压检测电路的使用方法可进行如下运行。

### 运行示例 1: 用作复位时

可能反复出现复位状态 / 复位解除状态。

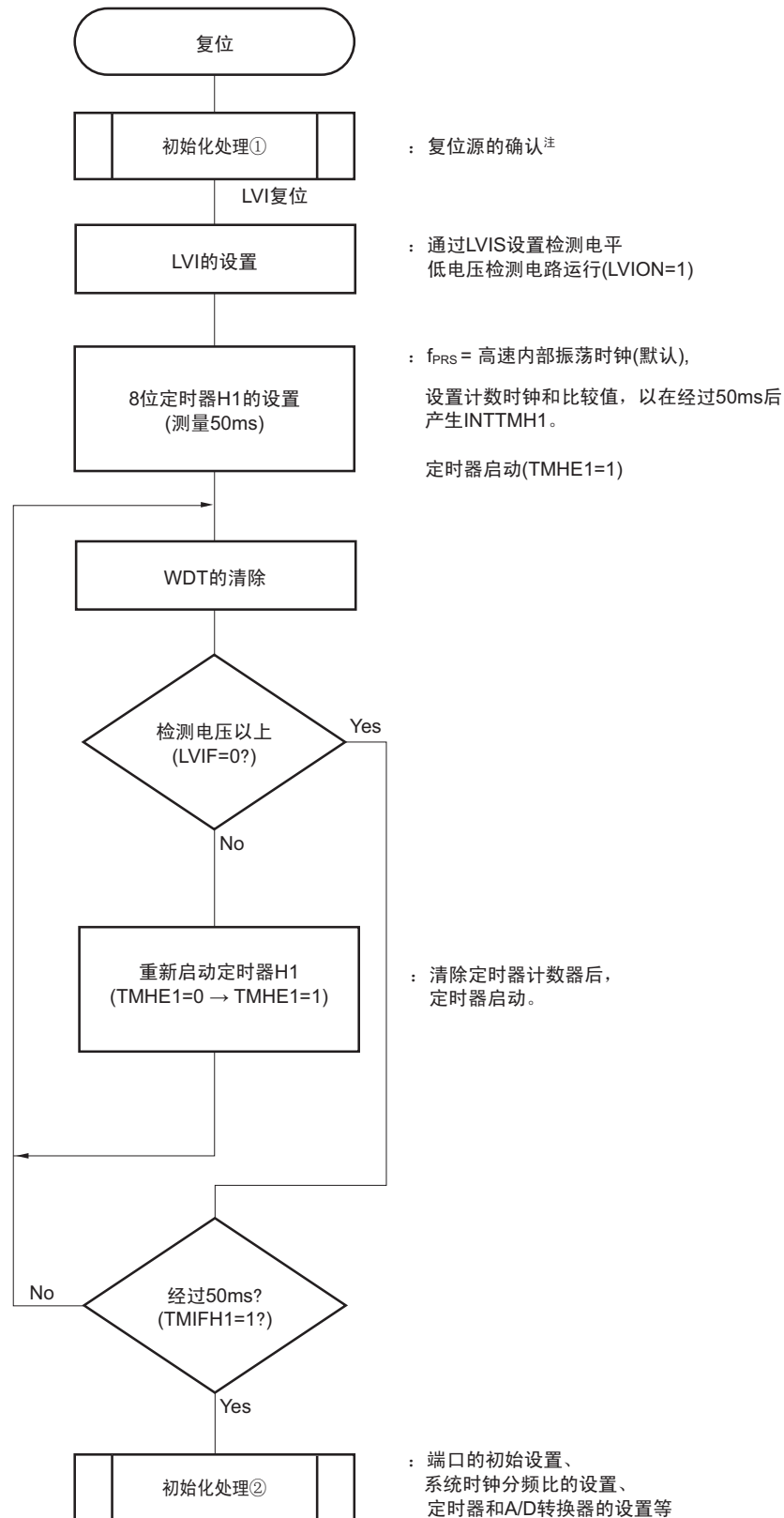
可通过以下措施, 任意设置从复位解除到单片机开始运行的时间。

#### < 措施 >

复位解除后, 通过使用定时器等的软件计数器, 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 18-8)。

图 18-8 复位解除后的软件处理示例 (1/2)

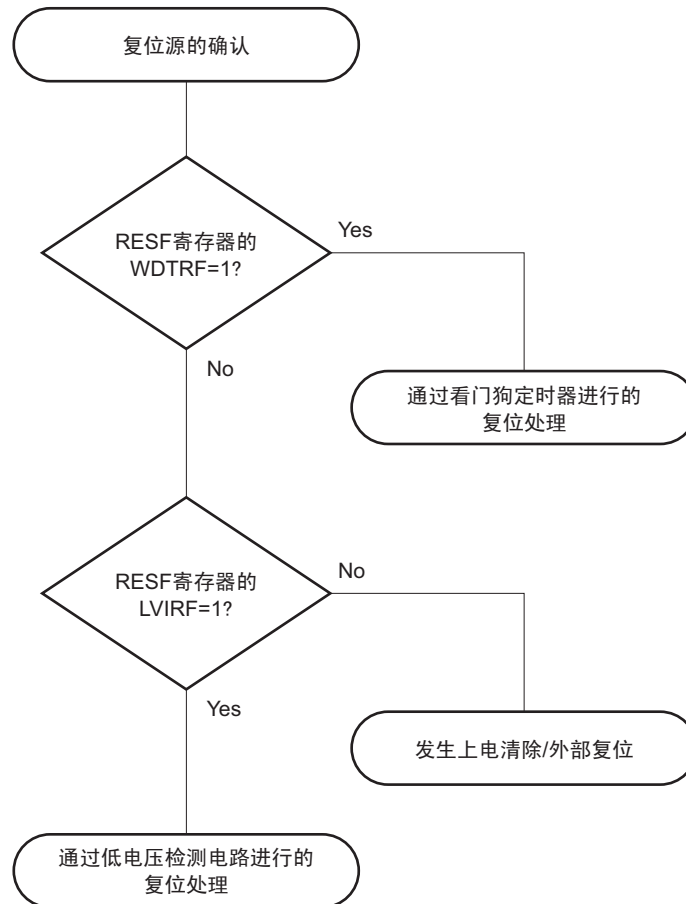
- 接近LVI检测电压的电源电压波动为50ms以下时



注 流程图如下一页所示。

图 18-8 复位解除后的软件处理示例 (2/2)

- 复位源的确认



### 运行示例 2：用作中断时

可能会频繁产生中断请求。

执行以下措施。

#### < 措施 >

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（ $V_{DD}$ ）大于等于 LVI 检测电压（ $V_{LVI}$ ）”，在检测到上升沿时“电源电压（ $V_{DD}$ ）小于 LVI 检测电压（ $V_{LVI}$ ）”，并且将中断请求标志寄存器 0L（IF0L）的 bit1（LVIF）清“0”。

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执行上述措施。



## 第 19 章 稳压器

### 19.1 稳压器的概要

R7F0C30x、R7F0C31x 内置使设备内部以恒电压运行的电路。此时，为了稳定稳压器的输出电压，通过电容器（0.47 ~ 1 $\mu$ F）将 REGC 引脚连接到 V<sub>SS</sub>。但是，如果在高速内部振荡时钟、外部主系统时钟运行时使用 STOP 模式，推荐 0.47 $\mu$ F。另外，因为用于稳定内部电压，推荐使用性能良好的电容器。

## 第 20 章 选项字节

### 20.1 选项字节的功能

R7F0C30x、R7F0C31x 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。

如果自编程时使用引导交换功能，1080H ~ 1084H 就切换到 0080H ~ 0084H。因此，必须事先给 1080H ~ 1084H 和 0080H ~ 0084H 设置相同的值。

#### (1) 0080H/1080H

- 低速内部振荡器的运行
  - 可通过软件停止
  - 不可停止
- 看门狗定时器的间隔时间设置
- 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

**注意** 在引导交换时，0080H 和 1080H 进行切换，因此，必须事先给 0080H 和 1080H 设置相同的值。

#### (2) 0081H/1081H

- LVI 默认启动的运行控制
  - LVI 默认启动功能运行时 (LVISTART=1)  
从复位解除或者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。  
如果从复位解除或者接通电源开始到达到 2.7V(TYP.) 前的电压上升斜率低于 0.5V/ms(MIN.)，推荐运行 LVI 默认启动功能。
  - LVI 默认启动功能停止时 (LVISTART=0)  
从复位解除或者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。

**注意** LVISTART 只能通过专用闪存编程器进行改写。不能在自编程和自编程的引导交换运行中进行设置或者更改。但是，在引导交换时，0081H 和 1081H 进行切换，因此，必须事先给 0081H 和 1081H 设置相同的值。

#### (3) 0082H/1082H

必须设置为“00H”。

**注意** 在引导交换时，0082H 和 1082H 进行切换，因此，必须事先给 1082H 设置“00H”。

## (4) 0083H/1083H

- 片上调试模式的选择
  - 不设置为片上调试模式
  - 强制设置为片上调试模式
- 在执行 STOP 指令时向 UART0 提供时钟。
  - 提供
  - 停止

注意 在引导交换时，0083H 和 1083H 进行切换，因此，必须事先给 0083H 和 1083H 设置相同的值。

## (5) 0084H/1084H

- 片上调试器的运行控制
  - 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。

注意 在引导交换时，0084H 和 1084H 进行切换，因此，必须事先给 0084H 和 1084H 设置相同的值。

## 20.2 选项字节的格式

选项字节的格式如下所示。

图 20-1 选项字节的格式 (1/3)

地址：0080H/1080H注

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	$2^{10}/f_{IL}$ (4.27ms)
0	0	1	$2^{11}/f_{IL}$ (8.53ms)
0	1	0	$2^{12}/f_{IL}$ (17.07ms)
0	1	1	$2^{13}/f_{IL}$ (34.13ms)
1	0	0	$2^{14}/f_{IL}$ (68.27ms)
1	0	1	$2^{15}/f_{IL}$ (136.53ms)
1	1	0	$2^{16}/f_{IL}$ (273.07ms)
1	1	1	$2^{17}/f_{IL}$ (546.13ms)

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 bit1（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 在引导交换时，0080H 和 1080H 进行切换，因此，必须事先给 0080H 和 1080H 设置相同的值。

注意 1. 禁止 WDCS2=WDCS1=WDCS0=0 和 WINDOW1=WINDOW0=0 的组合设置。

2. 在  $1.8V \leq V_{DD} < 2.7V$  下使用时，禁止设置 WINDOW1=WINDOW0=0。

3. 在闪存的自编程期间，看门狗定时器持续运行。但是，在进行这些处理期间，会延迟中断的接受时间，所以在设置上溢时间和窗口尺寸时，应考虑到此延迟。

4. 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。

5. 必须给 bit7 写“0”。

备注 1.  $f_{IL}$ ：低速内部振荡时钟频率

2. ( )： $f_{IL}=240\text{kHz}$  (TYP.) 的情况

图 20-1 选项字节的格式 (2/3)

地址：0081H/1081H 注 1、注 2

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	LVISTART

LVISTART	LVI 默认启动功能的运行控制
0	在电源上升时，LVI 默认启动功能停止。
1	在电源上升时，LVI 默认启动功能运行。

- 注 1. LVISTART 只能通过专用闪存编程器进行改写。不能在自编程和自编程的引导交换运行中进行设置。但是，在引导交换时，0081H 和 1081H 进行切换，因此，必须事先给 0081H 和 1081H 设置相同的值。
2. 在更改 LVI 默认启动的设置内容时，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/1082H 注

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

注 在引导交换时，0082H 和 1082H 进行切换，因此，必须事先给 1082H 设置“00H”。

注意 必须设置为“00H”。

地址：0083H/1083H 注

7	6	5	4	3	2	1	0
0	0	0	OCDCKSTP	1	1	1	OCDONB

OCDCKSTP	在片上调试模式中执行 STOP 指令时向 UART0 提供时钟
0	即使执行 STOP 指令，也不能停止内部振荡器，继续向 UART0 提供时钟。CPU 和外围硬件停止运行。
1	停止内部振荡器的运行，并且停止向 UART0 提供时钟。

OCDONB	片上调试模式
0	不设置为片上调试模式
1	强制设置为片上调试模式

注 在引导交换时，0083H 和 1083H 进行切换，因此，必须事先给 0083H 和 1083H 设置相同的值。

注意 必须给 bit7 ~ bit5、bit0 写“0”，给 bit3 ~ bit1 写“1”。

图 20-1 选项字节的格式 (3/3)

地址：0084H/1084H 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，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，不擦除闪存数据。

注 在引导交换时，0084H 和 1084H 进行切换，因此，必须事先给 0084H 与 1084H 设置相同的值。

注意 必须给 bit7 ~ bit2 写“0”。

备注 有关片上调试安全 ID，请参照“第 22 章 片上调试功能”。

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 $2^{10}/f_{IL}$ 、 ; 可通过软件停止低速内部振荡器
	DB	00H	; 停止 LVI 默认启动功能
	DB	00H	; 内部高速振荡时钟频率为 8MHz(TYP.)
	DB	1EH	; 使用 TOOLC0/X1、TOOLD0/X2 引脚
	DB	02H	; 允许运行、在片上调试安全 ID 认证失败时不擦除闪存数据

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 16 章 复位功能”。

## 第 21 章 闪存

R7F0C30x、R7F0C31x 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。

### 21.1 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。

在产生复位信号时，IMS 为“CFH”。

**注意** 由于复位时 ROM 区域的设置不定，所以在复位解除后，必须给各产品设置表 21-1 所示的值。

图 21-1 存储容量切换寄存器（IMS）的格式

地址：FFF0H 复位时：CFH R/W

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
0	1	0	512 字节
1	1	0	(默认值)
上述以外			禁止设置

ROM3	ROM2	ROM1	ROM0	内部 ROM 容量的选择
0	0	0	1	4K 字节
0	0	1	0	8K 字节
0	1	0	0	16K 字节
1	1	1	1	(默认值)
上述以外				禁止设置

表 21-1 存储容量切换寄存器（IMS）的设置值

产品	IMS 的设置值
R7F0C3004、R7F0C3034、R7F0C3006、R7F0C3036、 R7F0C3064、R7F0C3094、R7F0C3066、R7F0C3096	41H
R7F0C3014、R7F0C3044、R7F0C3016、R7F0C3046、 R7F0C3074、R7F0C3104、R7F0C3076、R7F0C3106、	42H
R7F0C3024、R7F0C3054、R7F0C3026、R7F0C3056、 R7F0C3084、R7F0C3114、R7F0C3086、R7F0C3116	04H

## 21.2 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 R7F0C30x、R7F0C31x 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

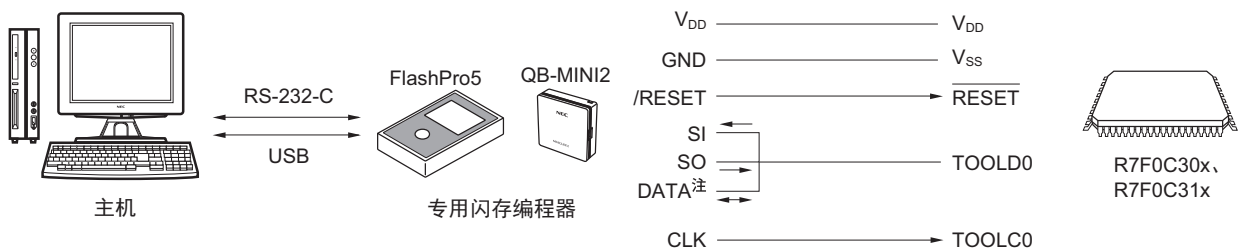
在将 R7F0C30x、R7F0C31x 安装到目标系统前，通过专用程序适配器（FA 系列）等进行闪存编程。

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。只支持 SSOP 封装产品。

## 21.3 编程环境

对 R7F0C30x、R7F0C31x 的闪存进行编程所需的环境如下所示。

图 21-2 闪存编程的环境



注 只限QB-MINI2

需要控制专用闪存编程器的主机。

专用闪存编程器与 R7F0C30x、R7F0C31x 的接口使用 TOOLD0 引脚，通过专用单线 UART 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）。

表 21-2 引脚连接一览

专用闪存编程器			R7F0C30x、R7F0C31x
信号名称	输入 / 输出	引脚功能	引脚名称
CLK	输出	向 R7F0C30x、R7F0C31x 输出时钟	TOOLC0
SI	输入	接收信号	TOOLD0
SO	输出	发送信号	
DATA 注	输入 / 输出	用于调试期间数据通信的输入 / 输出信号	RESET
/RESET	输出	复位信号	
V <sub>DD</sub>	输入 / 输出	V <sub>DD</sub> 电压生成 / 电压监视	V <sub>DD</sub>
GND	—	接地	V <sub>SS</sub>

注 只限 QB-MINI2



## 21.4 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。

### 21.4.1 TOOL 引脚

在闪存编程模式中使用的通信引脚如下所示。

表 21-3 在闪存编程模式中使用的通信引脚

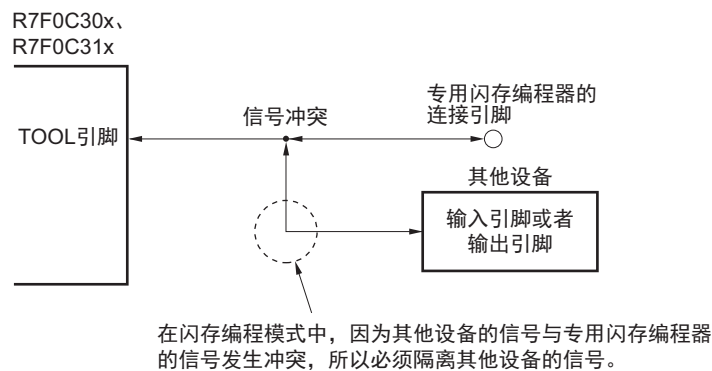
引脚名称	引脚处理
TOOLC0	直接连接专用闪存编程器，或者通过电阻（10kΩ）连接 V <sub>SS</sub> 。
TOOLD0	直接连接专用闪存编程器，或者通过电阻（3kΩ ~ 10kΩ）连接 V <sub>DD</sub> 。

在板上连接其他设备和专用闪存编程器时，必须注意信号冲突和其他设备的异常运行等。

#### (1) 信号冲突

在与其他设备连接的 TOOL 引脚上再连接专用闪存编程器时，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 21-3 信号冲突（TOOL 引脚）

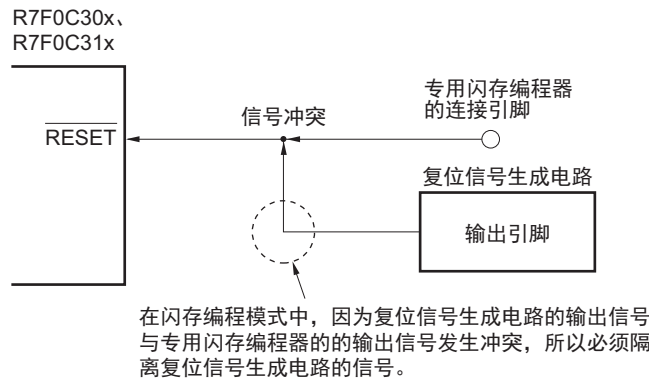


### 21.4.2 $\overline{\text{RESET}}$ 引脚

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 21-4 信号冲突 ( $\overline{\text{RESET}}$  引脚)



### 21.4.3 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。

### 21.4.4 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  $V_{SS}$ 。但是，如果在高速内部振荡时钟、外部主系统时钟运行时使用 STOP 模式，推荐  $0.47\mu\text{F}$ 。另外，因为用于稳定内部电压，所以请使用性能良好的电容器。

### 21.4.5 其他信号的引脚

X1、X2 使用与正常运行模式时相同的状态进行连接。

备注 在闪存编程模式中，使用高速内部振荡时钟 ( $f_{IH}$ )。

### 21.4.6 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。

其他电源 ( $AV_{REF}$ 、 $V_{SS}$ ) 提供与正常运行模式相同的电源。

### 21.4.7 连接晶体 / 陶瓷谐振器时的板上编程

在进行板上编程时，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。

在自编程模式中，引脚的状态与 HALT 模式中的状态相同。

在将 X1 (TOOLC0) 和 X2 (TOOLD0) 引脚分别用作闪存编程器的串行接口时，如果连接外部设备，就发生信号冲突。为避免发生信号冲突，必须隔离外部设备的连接。

同样，在给 X1 和 X2 引脚连接电容器时，由于通信期间的波形发生变化，所以根据电容器的容量有可能无法进行通信。在进行闪存编程时，必须隔离电容器的连接。

在选择晶体 / 陶瓷谐振作为系统时钟，并且因为很难隔离谐振器而在安装谐振器的状态下进行板上编程时，对在安装谐振器状态下的编程进行评估后，必须执行以下处理。

- 在设备和谐振器之间尽量安装小型测试焊盘，通过测试焊盘连接编程器。并且布线尽量要短（参照图 21-5 和表 21-4）。

图 21-5 测试焊盘的安装示例

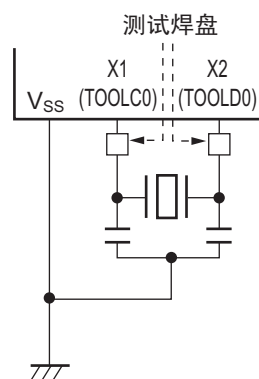


表 21-4 使用时钟和测试焊盘的安装

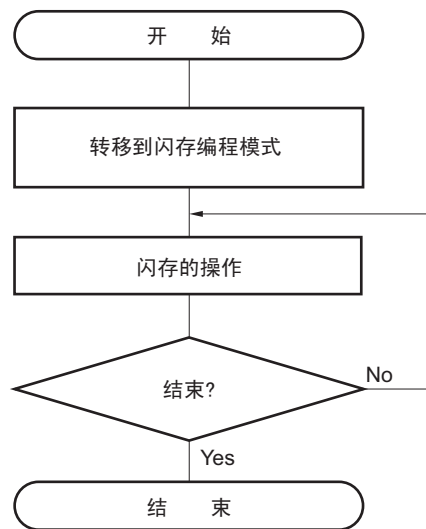
使用时钟		测试焊盘的安装
高速内部振荡时钟		不需要
外部时钟		
晶体 / 陶瓷振荡时钟	安装谐振器前	
	安装谐振器后	需要

## 21.5 编程方法

### 21.5.1 闪存控制

闪存的操作步骤如下所示。

图 21-6 闪存的操作步骤



### 21.5.2 闪存编程模式

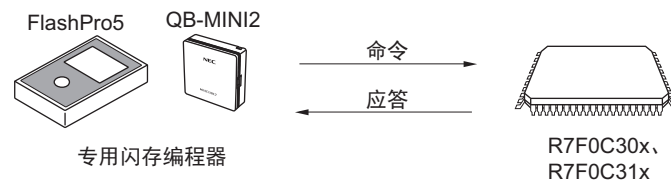
在使用专用闪存编程器改写闪存的内容时，必须将 R7F0C30x、R7F0C31x 设置为闪存编程模式。如果连接专用闪存编程器，并且开始通信，就切换到闪存编程模式。

在进行板上编程时，通过跳线等切换模式。

### 21.5.3 通信命令

R7F0C30x、R7F0C31x 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 R7F0C30x、R7F0C31x 发送的信号称为“命令”，从 R7F0C30x、R7F0C31x 向专用闪存编程器发送的信号称为“应答”。

图 21-7 通信命令



R7F0C30x、R7F0C31x 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，R7F0C30x、R7F0C31x 根据命令进行相应的处理。

表 21-5 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	Silicon Signature	获取 R7F0C30x、R7F0C31x 信息（产品名称、闪存结构等）
	Version Get	获取 R7F0C30x、R7F0C31x 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
	Baud Rate Set	设置选择 UART 时的波特率

R7F0C30x、R7F0C31x 对专用闪存编程器发行的命令返回应答。R7F0C30x、R7F0C31x 发出的应答名称如下所示。

表 21-6 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

## 21.6 安全设置

R7F0C30x、R7F0C31x 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。但是，可进行自编程时的块擦除。
- 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。但是，可进行自编程时的编程。
- 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。

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，解除所有安全设置。

在 R7F0C30x、R7F0C31x 的安全功能有效时，擦除和编程命令的关系如表 21-7 所示。

表 21-7 在安全功能有效时与命令的关系

## (1) 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。

## (2) 自编程时

安全有效	执行命令	
	块擦除	编程
禁止批量擦除（片擦除）	能块擦除	能编程
禁止块擦除		
禁止编程		
禁止改写引导簇 0	不能擦除引导簇 0	不能编程引导簇 0

各编程模式中的安全设置方法如表 21-8 所示。

表 21-8 各编程模式中的安全设置方法

## (1) 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

## (2) 自编程

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）	通过设置信息库设置	设置后不能置为无效
禁止块擦除		在进行板上 / 板外编程时，执行批量擦除（片擦除）命令（在自编程时不能置为无效）。
禁止编程		
禁止改写引导簇 0		

## 21.7 通过自编程进行的闪存编程

R7F0C30x、R7F0C31x 支持在用户程序改写闪存的自编程功能。由于该功能通过使用 R7F0C30x、R7F0C31x 的自编程库在用户应用改写闪存，因此可用于现场程序的升级等。

另外，如果在自编程期间发生中断，可暂时停止自编程而执行中断处理。如果在 EI 状态下产生非屏蔽中断请求，则从自编程库直接转移到中断程序。之后可再次转移到自编程模式，恢复自编程。但是，中断响应时间与正常运行模式中的时间不同。

注意 1. 如果要在自编程期间禁止中断，就必须与正常运行模式相同，在通过 DI 指令将 IE 标志清“0”的状态下执行自编程库。

如果要允许中断，就必须通过 EI 指令将 IE 标志置“1”的状态下，将接受中断的中断屏蔽标志清“0”，然后执行自编程库。

2. 改写闪存时，能进行改写的电源电压范围为  $2.7V \leq V_{DD} \leq 5.5V$ 。

备注 有关自编程功能和自编程库的详细内容，请参照“78K0 Microcontrollers Self Programming Library Type01 User's Manual (U18274E)”和“78K0 Microcontrollers Self Programming Library Type01 Ver.3.10 Operating Precautions (notification document) (ZUD-CD-09-0122)”。

### 21.7.1 自编程模式控制寄存器

自编程模式控制寄存器（FPCTL）控制自编程的模式。

通过 1 位或 8 位存储器操作指令设置 FPCTL。

在产生复位信号时，FPCTL 为“00H”。

图 21-8 自编程模式控制寄存器（FPCTL）的格式

地址：FF2BH 复位时：00H R/W

符号	7	6	5	4	3	2	1	<0>
FPCTL	0	0	0	0	0	0	0	FLMDPUP 注

FLMDPUP 注	自编程模式的控制
0	正常运行模式
1	自编程模式

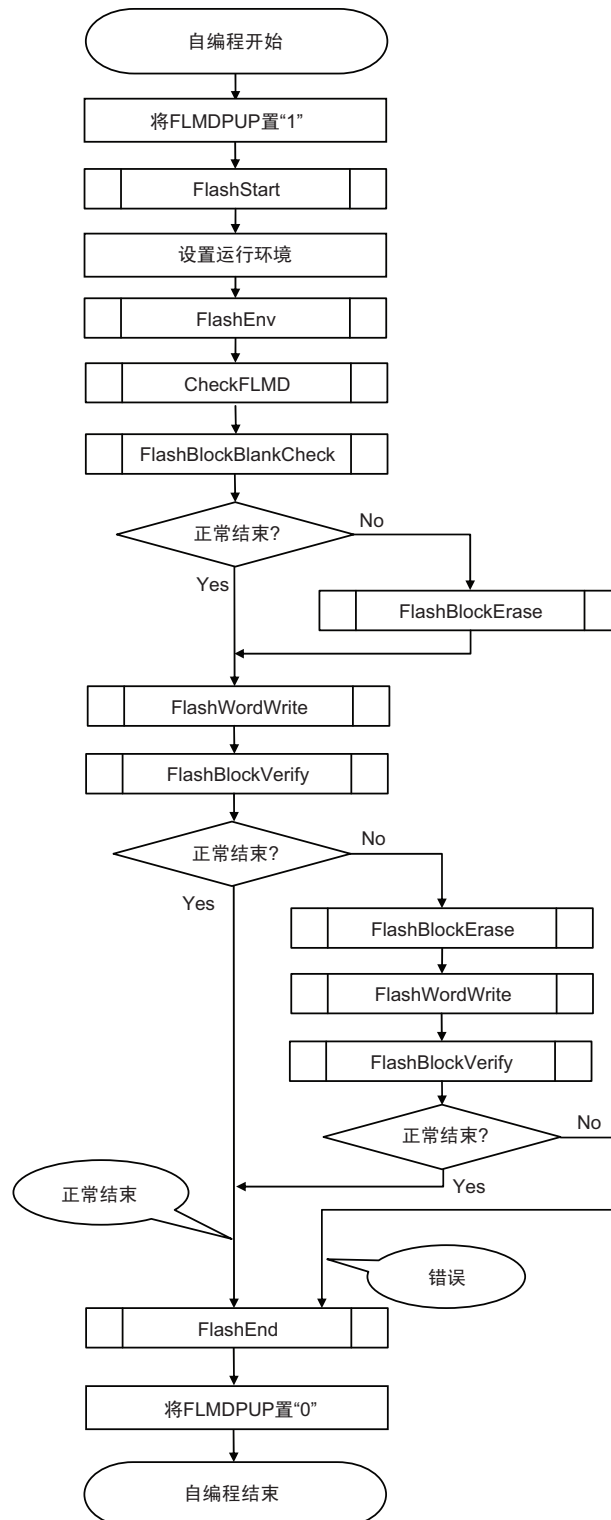
注 在执行通常的用户程序时，FLMDPUP 位必须置“0”（正常运行模式），在执行自编程时，FLMDPUP 位必须置“1”（自编程模式）。在正常运行模式中，由于闪存的改写电路不运行，因此，即使用于改写的固件和软件运行，也不执行实际的改写。



## 21.7.2 自编程（闪存的改写）的流程

使用自编程库改写闪存的流程如下所示。

图 21-9 自编程（闪存的改写）流程



备注 有关自编程功能和自编程库的详细内容，请参照“78K0 Microcontrollers Self Programming Library Type01 User's Manual (U18274E)”和“78K0 Microcontrollers Self Programming Library Type01 Ver.3.10 Operating Precautions (notification document) (ZUD-CD-09-0122)”。

### 21.7.3 引导交换功能

在通过自编程改写引导区域期间，如果因瞬间断电等导致改写失败，就会破坏引导区域的数据，并且无法通过复位重启或者改写程序。

引导交换功能用于避免上述问题。

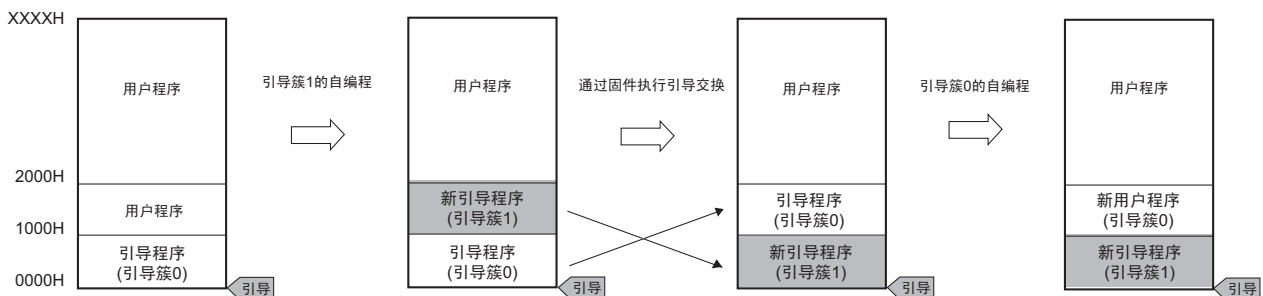
在通过自编程执行引导程序区域即引导簇 0 注的擦除前，必须事先将新的引导程序写入引导簇 1。在引导簇 1 的写操作正常结束后，通过 R7F0C30x、R7F0C31x 内置的固件的设置信息功能，进行该引导簇 1 和引导簇 0 的交换，并且引导簇 1 变为引导程序区域。此后，对原本的引导程序区域即引导簇 0 执行擦除或者编程。

通过上述运行，即使在引导程序区域的改写期间发生瞬间断电，由于从交换对象的引导簇 1 进行交换，所以下一个复位启动程序正常运行。

**注** 引导簇为 4K 字节区域，通过引导交换功能交换引导簇 0 和引导簇 1。

**注意** 4K 字节的 ROM 产品不能使用引导交换功能。

图 21-10 引导交换功能

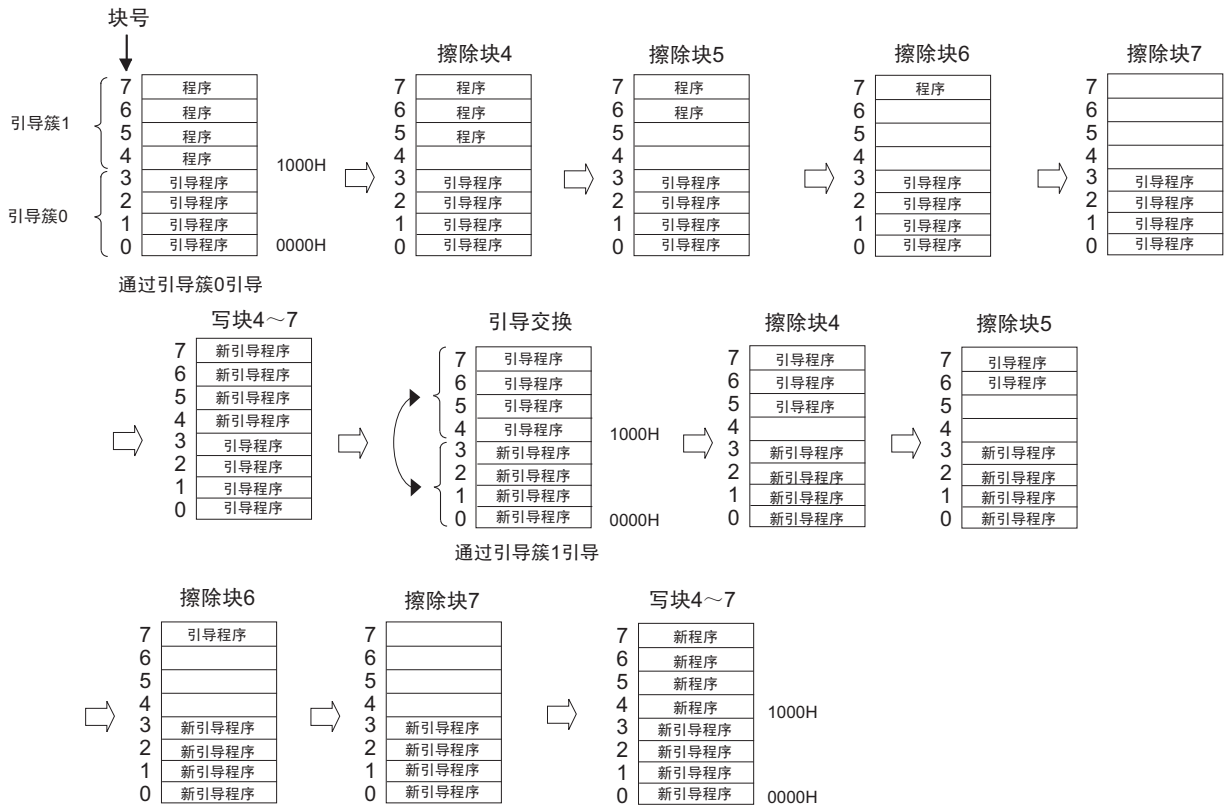


在上图示例中，

引导簇 0 是引导交换前的引导程序区域、

引导簇 1 是引导交换后的引导程序区域。

图 21-11 引导交换的执行示例



## 第 22 章 片上调试功能

### 22.1 QB-MINI2 与 R7F0C30x、R7F0C31x 的连接

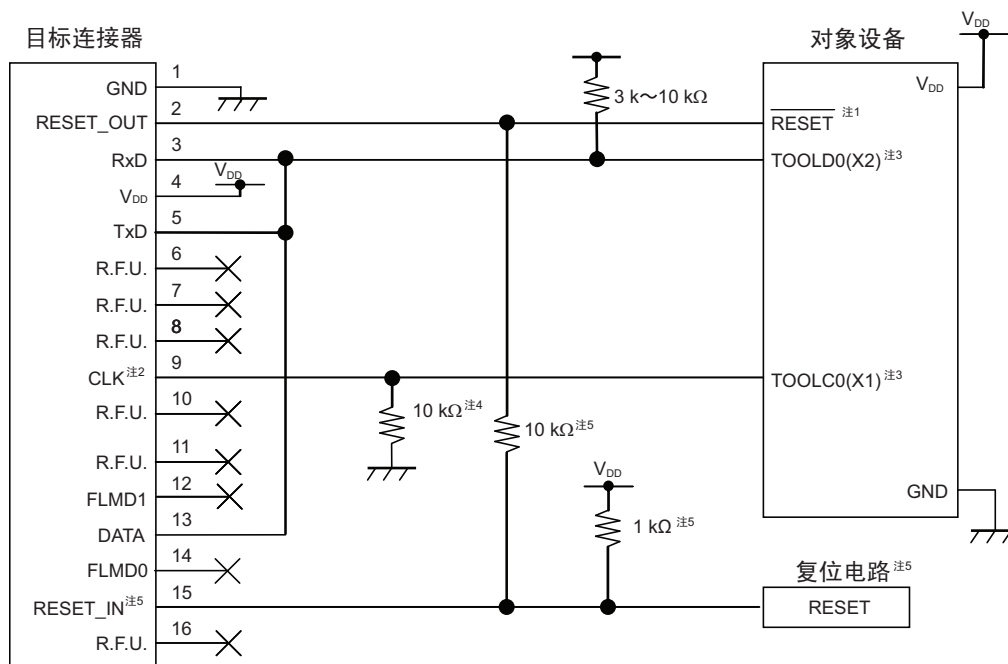
R7F0C30x、R7F0C31x 在通过支持片上调试功能的片上调试仿真器（QB-MINI2）进行与主机间的通信时，使用  $V_{DD}$ 、RESET、TOOLC0/X1、TOOLD0/X2 和  $V_{SS}$  引脚。

注意 1. R7F0C30x、R7F0C31x 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 22-1 QB-MINI2 与 R7F0C30x、R7F0C31x 的连接示例（1/2）

(1) 使用 TOOLC0、TOOLD0 引脚时（未使用 X1 振荡时钟或者 EXCLK 输入时钟，并且调试和编程都执行）



注 1. 如果含有电容器等的电容成分，则片上调试功能有可能无法正常运行。

2. 在片上调试期间，可使用在 78K0-OCD 板上安装的时钟、在 QB-MINI2 内部生成的 4/8MHz 时钟以及设备的高速内部振荡器，作为对象设备的运行时钟。

在闪存编程期间，只能使用设备的高速内部振荡器。

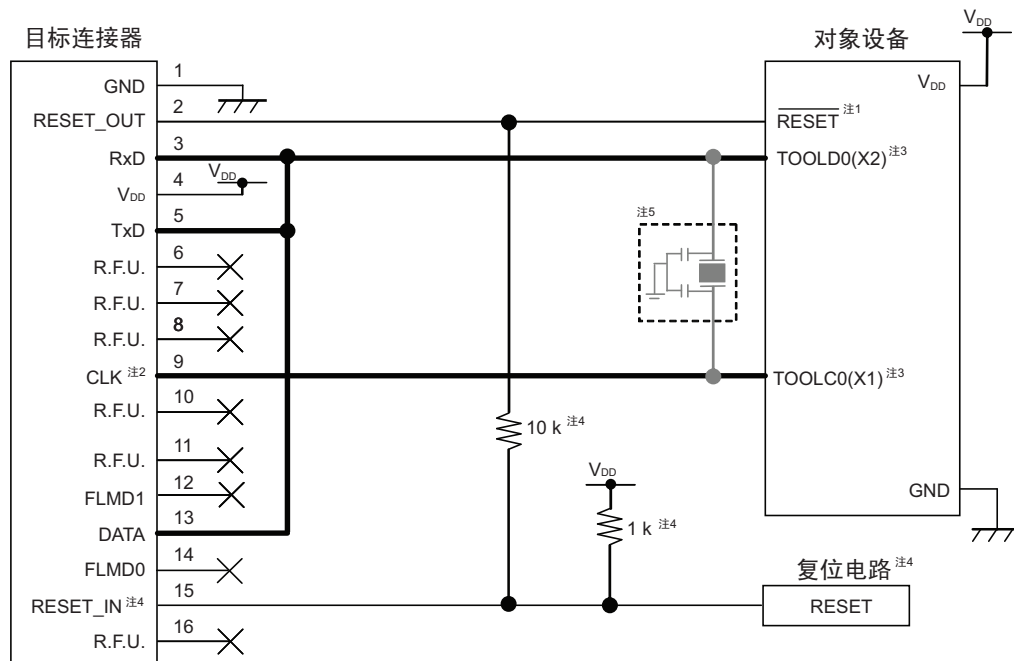
3. 在片上调试期间，由于这些引脚用作片上调试专用引脚，所以忽略在用户程序中的设置，但是在设置为输入引脚时，必须进行引脚处理（在未连接 QB-MINI2 时，必须置为开路）。

4. 这是在对象设备中运行时（未连接 QB-MINI2），未使用引脚（输入开路）的引脚处理（使用振荡电路时不需要）。

5. 这是将复位电路的输出假设为 N-ch 漏极开路缓冲器（输出电阻：100Ω 以下）的电路连接。

图 22-1 QB-MINI2 与 R7F0C30x、R7F0C31x 的连接示例 (2/2)

(2) 使用 TOOLC0、TOOLD0 引脚时 (使用 X1/X2 振荡时钟, 并且调试和编程都执行)



- 注 1. 如果含有电容器等的电容成分, 则片上调试功能有可能无法正常运行。
2. 在片上调试期间, 可使用在 78K0-OCD 板上安装的时钟、在 QB-MINI2 内部生成的 4/8MHz 时钟以及设备的高速内部振荡器, 作为目标设备的运行时钟。  
在闪存编程期间, 只能使用设备的高速内部振荡器。
3. 在片上调试期间, 由于这些引脚用作片上调试专用引脚, 所以忽略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 (在未连接 QB-MINI2 时, 必须置为开路)。
4. 这是将复位电路的输出假设为 N-ch 漏极开路缓冲器 (输出电阻: 100Ω 以下) 的电路连接。详细内容请参照“配备 QB-MINI2 编程功能的片上调试仿真器用户手册 (U18371J)”的“4.1.3 复位引脚的处理”。
5. 在片上调试和闪存编程期间, 禁止安装振荡电路。  
在目标设备中运行时 (未连接 QB-MINI2), 为了防止振荡电路因布线容量等而无法振荡的情况发生, 应采取如通过跨线等隔离目标连接器和振荡电路的措施。  
通过调试下载的程序, 在没有连接 QB-MINI2 时不能运行。

注意 对上图中粗线表示的线路 (TOOLD0 和 TOOLC0) 进行设计时, 从设备的引脚到 QB-MINI2 的连接器的长度不得超过 30mm, 或者进行 GND 屏蔽。

## 22.2 片上调试安全 ID

在 R7F0C30x、R7F0C31x 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（参阅“第 20 章 选项字节”），在地址 0085H ~ 008EH 中预留片上调试安全 ID 设置区域。

在自编程期间使用引导交换功能时，因为 0084H、0085H ~ 008EH 和 1084H、1085H ~ 108EH 之间会进行交换，所以必须事先给 1084H、1085H ~ 108EH 设置相同的值。

有关片上调试安全 ID 的详细内容，请参照“配备 QB-MINI2 编程功能的片上调试仿真器用户手册（U18371J）”。

表 22-1 片上调试安全 ID

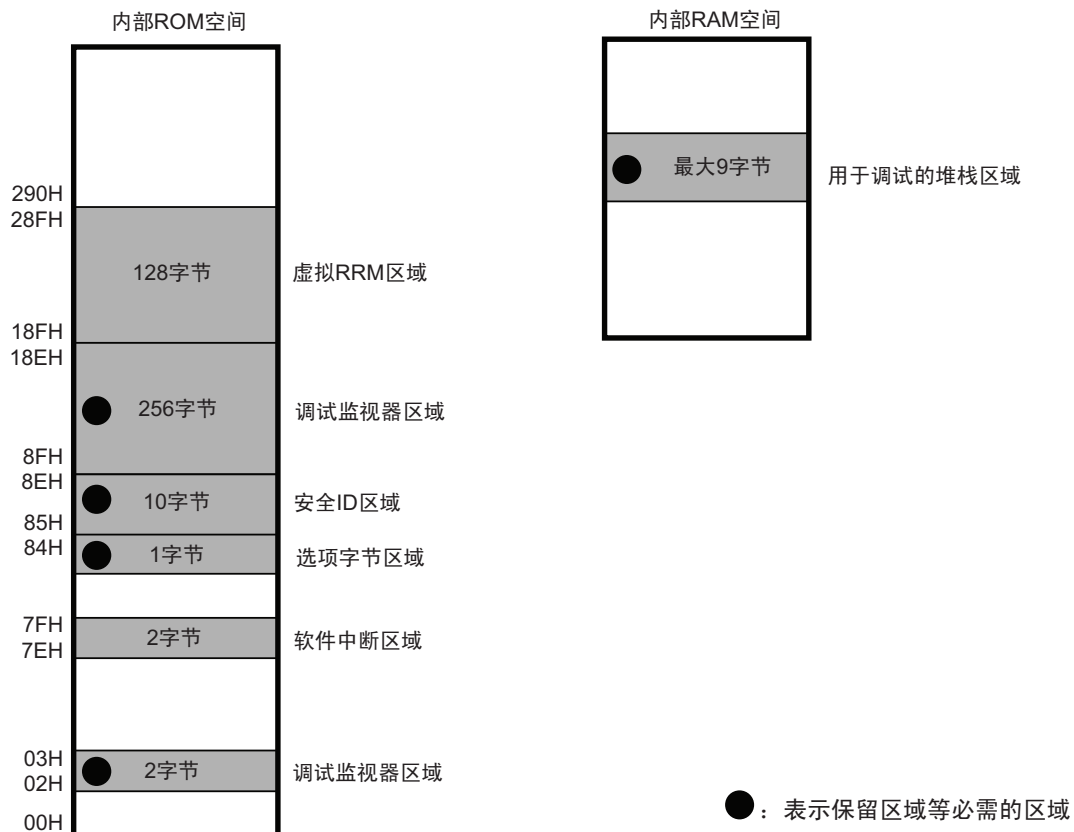
地址	片上调试安全 ID
0085H ~ 008EH	任意 10 字节的 ID 码
1085H ~ 108EH	

## 22.3 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 22-2 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“配备 QB-MINI2 编程功能的片上调试仿真器用户手册（U18371J）”。

图 22-2 QB-MINI2 使用的保留区域



## 第 23 章 指令集

通过一览表表示 R7F0C30x、R7F0C31x 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 用户手册 指令（U12326C）”。

### 23.1 凡例

#### 23.1.1 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 23-1 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 <sup>注</sup>
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） <sup>注</sup>
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”。

### 23.1.2 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 <sub>H</sub> 、X <sub>L</sub> :	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)

### 23.1.3 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



## 23.2 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（ $f_{\text{CPU}}$ ）。

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 ← word			
		saddrp, #word	4	8	10	(saddrp) ← word			
		sfrp, #word	4	—	10	sfrp ← word			
		AX, saddrp	2	6	8	AX ← (saddrp)			
		saddrp, AX	2	6	8	(saddrp) ← AX			
		AX, sfrp	2	—	8	AX ← sfrp			
		sfrp, AX	2	—	8	sfrp ← AX			
		AX, rp <sup>注3</sup>	1	4	—	AX ← rp			
		rp, AX <sup>注3</sup>	1	4	—	rp ← AX			
		AX, !addr16	3	10	12	AX ← (addr16)			
	!addr16, AX	3	10	12	(addr16) ← AX				
XCHW	AX, rp <sup>注3</sup>	1	4	—	AX ↔ rp				
8 位运算	ADD	A, #byte	2	4	—	A, CY ← A + byte	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	x	x	x
		A, r <sup>注4</sup>	2	4	—	A, CY ← A + r	x	x	x
		r, A	2	4	—	r, CY ← r + A	x	x	x
		A, saddr	2	4	5	A, CY ← A + (saddr)	x	x	x
		A, !addr16	3	8	9	A, CY ← A + (addr16)	x	x	x
		A, [HL]	1	4	5	A, CY ← A + (HL)	x	x	x
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	x	x	x
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	x	x	x
	A, [HL + C]	2	8	9	A, CY ← A + (HL + C)	x	x	x	
	ADDC	A, #byte	2	4	—	A, CY ← A + byte + CY	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	x	x	x
		A, r <sup>注4</sup>	2	4	—	A, CY ← A + r + CY	x	x	x
		r, A	2	4	—	r, CY ← r + A + CY	x	x	x
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	x	x	x
		A, !addr16	3	8	9	A, CY ← A + (addr16) + CY	x	x	x
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	x	x	x
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	x	x	x
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	x	x	x
A, [HL + C]		2	8	9	A, CY ← A + (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 ( $f_{\text{CPU}}$ )。

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 <sup>注3</sup>	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 <sup>注3</sup>	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 <sup>注3</sup>	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 ( $f_{\text{CPU}}$ )。  
 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	$CY \leftarrow CY \vee (\text{saddr.bit})$			x
		CY, sfr.bit	3	—	7	$CY \leftarrow CY \vee \text{sfr.bit}$			x
		CY, A.bit	2	4	—	$CY \leftarrow CY \vee A.\text{bit}$			x
		CY, PSW.bit	3	—	7	$CY \leftarrow CY \vee \text{PSW.bit}$			x
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \vee (\text{HL}).\text{bit}$			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	—	$(\text{SP}-1) \leftarrow (\text{PC}+3)_\text{H}, (\text{SP}-2) \leftarrow (\text{PC}+3)_\text{L},$ $\text{PC} \leftarrow \text{addr16}, \text{SP} \leftarrow \text{SP}-2$		
CALLF		!addr11	2	5	—	$(\text{SP}-1) \leftarrow (\text{PC}+2)_\text{H}, (\text{SP}-2) \leftarrow (\text{PC}+2)_\text{L},$ $\text{PC}_{15-11} \leftarrow 00001, \text{PC}_{10-0} \leftarrow \text{addr11},$ $\text{SP} \leftarrow \text{SP}-2$			
CALLT		[addr5]	1	6	—	$(\text{SP}-1) \leftarrow (\text{PC}+1)_\text{H}, (\text{SP}-2) \leftarrow (\text{PC}+1)_\text{L},$ $\text{PC}_\text{H} \leftarrow (\text{addr5}+1), \text{PC}_\text{L} \leftarrow (\text{addr5}),$ $\text{SP} \leftarrow \text{SP}-2$			
BRK			1	6	—	$(\text{SP}-1) \leftarrow \text{PSW}, (\text{SP}-2) \leftarrow (\text{PC}+1)_\text{H},$ $(\text{SP}-3) \leftarrow (\text{PC} + 1)_\text{L}, \text{PCH} \leftarrow (003\text{FH}),$ $\text{PC}_\text{L} \leftarrow (003\text{EH}), \text{SP} \leftarrow \text{SP} - 3, \text{IE} \leftarrow 0$			
RET			1	6	—	$\text{PC}_\text{H} \leftarrow (\text{SP}+1), \text{PC}_\text{L} \leftarrow (\text{SP}),$ $\text{SP} \leftarrow \text{SP}+2$			
RETI			1	6	—	$\text{PC}_\text{H} \leftarrow (\text{SP}+1), \text{PC}_\text{L} \leftarrow (\text{SP}),$ $\text{PSW} \leftarrow (\text{SP}+2), \text{SP} \leftarrow \text{SP}+3$	R	R	R
RETB			1	6	—	$\text{PC}_\text{H} \leftarrow (\text{SP}+1), \text{PC}_\text{L} \leftarrow (\text{SP}),$ $\text{PSW} \leftarrow (\text{SP}+2), \text{SP} \leftarrow \text{SP}+3$	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 ( $f_{\text{CPU}}$ )。

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				
		BNC	\$addr16	2	6	—	$PC \leftarrow PC+2+jdisp8$ if CY = 0			
		BZ	\$addr16	2	6	—	$PC \leftarrow PC+2+jdisp8$ if Z = 1			
		BNZ	\$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				
		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。



## 23.3 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 24 章 电特性

注意 1. R7F0C30x、R7F0C31x 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. 搭载的引脚根据产品而不同，如下所示。

### (1) 端口功能

端口	16 引脚	20 引脚
端口 2	P20 ~ P22、P25 ~ P27	P20 ~ P27
端口 3	P30 ~ P32	P30 ~ P34
端口 12	P121、P122、P125	P121、P122、P125

### (2) 端口以外的功能

功能	16 引脚	20 引脚
电源、接地	$V_{DD}$ 、 $V_{SS}$ 、 $AV_{REF}$	
稳压器	REGC	
复位	$\overline{RESET}$	
时钟振荡	EXCLK	
中断	INTP0、INTP1	
定时器	TM00	TI000、TI010、TO00
	TM51	TI51
	TMH1	TOH1
A/D 转换器	ANI0 ~ ANI2、ANI5 ~ ANI7	ANI0 ~ ANI7
运算放大器注	AMP0+、AMP0-、AMP0OUT、AMP1+、AMP1-、AMP1OUT	
比较器注	CMPCOM、CMPIN	CMPCOM、CMPIN、CMPOUT
片上调试功能	TOOLC0、TOOLD0	

注 只限 R7F0C303x、R7F0C304x、R7F0C305x、R7F0C309x、R7F0C310x、R7F0C311x（内置运算放大器 / 比较器的产品）。

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 ~ +6.5	V
	$V_{SS}$		-0.5 ~ +0.3	V
	$AV_{REF}$		-0.5 ~ $V_{DD}+0.3$ 注 1	V
REGC 引脚输入电压注 2	$V_{IREGC}$		-0.5 ~ +3.6 并且 -0.5 ~ $V_{DD}$	V
输入电压	$V_{I1}$	P30 ~ P34、P121、P122、P125、X1、X2、RESET	-0.3 ~ $V_{DD}+0.3$ 注 1	V
	$V_{I2}$	P20 ~ P27	-0.3 ~ $AV_{REF}+0.3$ 注 1 并且 -0.3 ~ $V_{DD}+0.3$ 注 1	V
输出电压	$V_{O1}$	P30 ~ P34、P121、P122	-0.3 ~ $V_{DD}+0.3$ 注 1	V
	$V_{O2}$	P20 ~ P27	-0.3 ~ $AV_{REF}+0.3$	V

注 1. 小于等于 6.5V。

2. REGC 引脚通过电容器 (0.47 ~ 1mF) 连接  $V_{SS}$ 。这是规定 REGC 引脚的绝对最大额定值的值。不能在外加电压后使用。

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	P31 ~ P34	-10	mA	
		引脚总计		-25	mA	
	$I_{OH2}$	1 个引脚	P30	-10	mA	
		引脚总计		-10	mA	
	$I_{OH3}$	1 个引脚	P20 ~ P27	-0.5	mA	
		引脚总计		-2	mA	
	$I_{OH4}$	1 个引脚	P121、P122	-1	mA	
		引脚总计		-4	mA	
	低电平输出电流	$I_{OL1}$	1 个引脚	P31 ~ P34	30	mA
			引脚总计		55	mA
$I_{OL2}$		1 个引脚	P30	30	mA	
		引脚总计		30	mA	
$I_{OL3}$		1 个引脚	P20 ~ P27	1	mA	
		引脚总计		5	mA	
$I_{OL4}$		1 个引脚	P121、P122	4	mA	
		引脚总计		10	mA	
工作环境温度		$T_A$		-40 ~ +85	$^\circ\text{C}$	
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。必须在不出超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<sub>A</sub>=-40 ~ +85°C、1.8V ≤ V<sub>DD</sub> ≤ 5.5V、V<sub>SS</sub>=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 <sub>X</sub> ) 注	2.7V ≤ V <sub>DD</sub> ≤ 5.5V	1.0		10.0	MHz
			1.8V ≤ V <sub>DD</sub> ≤ 5.5V	1.0		5.0	MHz

注 只表示振荡电路的特性。指令执行时间请参照“AC 特性”。

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<sub>SS</sub> 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。

备注 有关谐振器的选择和振荡电路的常数，需要用户自行对振荡进行评估，或者委托谐振器厂商进行评估。

## 高速内部振荡电路特性

(T<sub>A</sub>=-40 ~ +85°C、1.8V ≤ V<sub>DD</sub> ≤ 5.5V、V<sub>SS</sub>=0V)

谐振器	项目	条件		MIN.	TYP.	MAX.	单位	
高速内部振荡器	振荡频率 (f <sub>IH</sub> =8MHz) 注	RSTS=1	SSOP 产品	T <sub>A</sub> =-20 ~ +70°C (±2%)	7.84	8	8.16	MHz
				T <sub>A</sub> =-40 ~ +85°C (±3%)	7.76	8	8.24	MHz
			SOP 产品	T <sub>A</sub> =0 ~ +40°C (±2%)	7.84	8	8.16	MHz
				T <sub>A</sub> =-40 ~ +85°C (±5%)	7.6	8	8.4	MHz

注 只表示振荡电路的特性。指令执行时间请参照“AC 特性”。

## 低速内部振荡电路特性

(T<sub>A</sub>=-40 ~ +85°C、1.8V ≤ V<sub>DD</sub> ≤ 5.5V、V<sub>SS</sub>=0V)

谐振器	项目	条件	MIN.	TYP.	MAX.	单位
低速内部振荡器	振荡频率 (f <sub>IL</sub> )	2.7V ≤ V <sub>DD</sub> ≤ 5.5V	216	240	264	kHz
		1.8V ≤ V <sub>DD</sub> < 2.7V	192	240	264	kHz

## DC 特性 (1/6) (16 引脚产品)

(T<sub>A</sub>=-40 ~ +85°C、1.8V ≤ V<sub>DD</sub> ≤ 5.5V、AV<sub>REF</sub> ≤ V<sub>DD</sub>、V<sub>SS</sub>=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位	
高电平输出电流注 1	I <sub>OH1</sub>	P31、P32 1 个引脚	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			-3.0	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			-2.5	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			-1.0	mA
		P31、P32 总计注 3	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			-4.5	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			-4.0	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			-1.5	mA
	I <sub>OH2</sub>	P30 1 个引脚	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			-3.0	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			-2.5	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			-1.0	mA
	I <sub>OH3</sub>	P20 ~ P22、P25 ~ P27 1 个引脚	AV <sub>REF</sub> =V <sub>DD</sub>			-100	μA
P121、P122 1 个引脚					-100	μA	
低电平输出电流注 2	I <sub>OL1</sub>	P31、P32 1 个引脚	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			8.5	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			5.0	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			1.0	mA
		P31、P32 总计注 3	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			15.0	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			8.0	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			1.5	mA
	I <sub>OL2</sub>	P30 1 个引脚	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			8.5	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			5.0	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			1.0	mA
	I <sub>OL3</sub>	P122 1 个引脚	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			8.5	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			5.0	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			1.0	mA
	I <sub>OL4</sub>	P20 ~ P22、P25 ~ P27 1 个引脚	AV <sub>REF</sub> =V <sub>DD</sub>			400	μA
			P121 1 个引脚			400	μA

- 注 1. 这是即使电流从 V<sub>DD</sub> 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% 的引脚的总输出电流可通过以下计算式算出。
- I<sub>OH</sub> 的占空比为 n% 的情况：引脚的总输出电流=(I<sub>OH</sub>×0.7)/(n×0.01)
- <计算示例> 当占空比=50%、I<sub>OH</sub>=-20.0mA 时  
 引脚的总输出电流=(-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 (2/6) (20 引脚产品)

(T<sub>A</sub>=-40 ~ +85°C、1.8V ≤ V<sub>DD</sub> ≤ 5.5V、AV<sub>REF</sub> ≤ V<sub>DD</sub>、V<sub>SS</sub>=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位	
高电平输出电流注 1	I <sub>OH1</sub>	P31 ~ P34 1 个引脚	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			-3.0	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			-2.5	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			-1.0	mA
		P31 ~ P34 总计注 3	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			-6.0	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			-4.5	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			-3.5	mA
	I <sub>OH2</sub>	P30 1 个引脚	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			-3.0	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			-2.5	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			-1.0	mA
	I <sub>OH3</sub>	P20 ~ P27 1 个引脚	AV <sub>REF</sub> =V <sub>DD</sub>			-100	μA
P121、P122 1 个引脚					-100	μA	
低电平输出电流注 2	I <sub>OL1</sub>	P31 ~ P34 1 个引脚	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			8.5	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			5.0	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			1.0	mA
		P31 ~ P34 总计注 3	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			15.0	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			10.0	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			3.5	mA
	I <sub>OL2</sub>	P30 1 个引脚	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			8.5	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			5.0	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			1.0	mA
	I <sub>OL3</sub>	P122 1 个引脚	4.0V ≤ V <sub>DD</sub> ≤ 5.5V			8.5	mA
			2.7V ≤ V <sub>DD</sub> < 4.0V			5.0	mA
			1.8V ≤ V <sub>DD</sub> < 2.7V			1.0	mA
	I <sub>OL4</sub>	P20 ~ P27 1 个引脚	AV <sub>REF</sub> =V <sub>DD</sub>			400	μA
		P121 1 个引脚				400	μA

- 注 1. 这是即使电流从 V<sub>DD</sub> 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% 的引脚的总输出电流可通过以下计算式算出。
- I<sub>OH</sub> 的占空比为 n% 的情况：引脚的总输出电流=(I<sub>OH</sub>×0.7)/(n×0.01)
- <计算示例> 当占空比=50%、I<sub>OH</sub>=-20.0mA 时  
 引脚的总输出电流=(-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 (3/6)

(T<sub>A</sub>=-40 ~ +85°C、1.8V ≤ V<sub>DD</sub> ≤ 5.5V、AV<sub>REF</sub> ≤ V<sub>DD</sub>、V<sub>SS</sub>=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
高电平输入电压	V <sub>IH1</sub>	P122注、P31	0.7V <sub>DD</sub>		V <sub>DD</sub>	V
	V <sub>IH2</sub>	P20 ~ P27	AV <sub>REF</sub> =V <sub>DD</sub>		AV <sub>REF</sub>	V
	V <sub>IH3</sub>	P121、P125	0.7V <sub>DD</sub>		V <sub>DD</sub>	V
	V <sub>IH4</sub>	P30、P32 ~ P34、 $\overline{\text{RESET}}$ 、EXCLK	0.8V <sub>DD</sub>		V <sub>DD</sub>	V
	V <sub>IH5</sub>	X1、X2	V <sub>DD</sub> -0.1		V <sub>DD</sub>	V
低电平输入电压	V <sub>IL1</sub>	P122注、P31	0		0.3V <sub>DD</sub>	V
	V <sub>IL2</sub>	P20 ~ P27	AV <sub>REF</sub> =V <sub>DD</sub>		0.3AV <sub>REF</sub>	V
	V <sub>IL3</sub>	P121、P125	0		0.3V <sub>DD</sub>	V
	V <sub>IL4</sub>	P30、P32 ~ P34、 $\overline{\text{RESET}}$ 、EXCLK	0		0.2V <sub>DD</sub>	V
	V <sub>IL5</sub>	X1、X2	0		0.1	V
高电平输出电压	V <sub>OH1</sub>	P30 ~ P34	4.0V ≤ V <sub>DD</sub> ≤ 5.5V、 I <sub>OH1</sub> =-3.0mA		V <sub>DD</sub> -0.7	V
			2.7V ≤ V <sub>DD</sub> < 4.0V、 I <sub>OH1</sub> =-2.5mA		V <sub>DD</sub> -0.5	V
			1.8V ≤ V <sub>DD</sub> < 2.7V、 I <sub>OH1</sub> =-1.0mA		V <sub>DD</sub> -0.5	V
	V <sub>OH2</sub>	P20 ~ P27	AV <sub>REF</sub> =V <sub>DD</sub> 、 I <sub>OH2</sub> =-100μA		V <sub>DD</sub> -0.5	V
			P121、P122	I <sub>OH2</sub> =-100μA		V <sub>DD</sub> -0.5
低电平输出电压	V <sub>OL1</sub>	P30 ~ P34、P122	4.0V ≤ V <sub>DD</sub> ≤ 5.5V、 I <sub>OL1</sub> =8.5mA		0.8	V
			2.7V ≤ V <sub>DD</sub> < 4.0V、 I <sub>OL1</sub> =5.0mA		0.7	V
			1.8V ≤ V <sub>DD</sub> < 2.7V、 I <sub>OL1</sub> =1.0mA		0.5	V
	V <sub>OL2</sub>	P20 ~ P27	AV <sub>REF</sub> =V <sub>DD</sub> 、 I <sub>OL2</sub> =400μA		0.4	V
			P121	I <sub>OL2</sub> =400μA		0.4

注 P122/EXCLK 的 V<sub>IH</sub>/V<sub>IL</sub> 因输入端口模式和外部时钟模式而不同。

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 (4/6)

(T<sub>A</sub>=-40 ~ +85°C、1.8V ≤ V<sub>DD</sub> ≤ 5.5V、AV<sub>REF</sub> ≤ V<sub>DD</sub>、V<sub>SS</sub>=0V)

项目	符号	条件		MIN.	TYP.	MAX.	单位		
高电平输入泄露电流	I <sub>LIH1</sub>	P30 ~ P34、P125/RESET		V <sub>I</sub> =V <sub>DD</sub>		3	μA		
	I <sub>LIH2</sub>	P20 ~ P27		V <sub>I</sub> =AV <sub>REF</sub> =V <sub>DD</sub>		3	μA		
	I <sub>LIH3</sub>	P121、P122		V <sub>I</sub> =	I/O 端口模式	3	μA		
		X1、X2		V <sub>DD</sub>	OSC 模式	20	μA		
低电平输入泄露电流	I <sub>LIL1</sub>	P30 ~ P34、P125/RESET		V <sub>I</sub> =V <sub>SS</sub>		-3	μA		
	I <sub>LIL2</sub>	P20 ~ P27		V <sub>I</sub> =V <sub>SS</sub> 、AV <sub>REF</sub> =V <sub>DD</sub>		-3	μA		
	I <sub>LIL3</sub>	P121、P122		V <sub>I</sub> =	I/O 端口模式	-3	μA		
		X1、X2		V <sub>SS</sub>	OSC 模式	-20	μA		
上拉电阻值	R <sub>PLU1</sub>	P30 ~ P34		V <sub>I</sub> =V <sub>SS</sub>		10	20	100	kΩ
	R <sub>PLU2</sub>	P125/RESET				75	150	300	kΩ

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 (5/6)

(T<sub>A</sub>=-40 ~ +85°C、1.8V ≤ V<sub>DD</sub> ≤ 5.5V、AV<sub>REF</sub> ≤ V<sub>DD</sub>、V<sub>SS</sub>=0V)

项目	符号	条件		MIN.	TYP.	MAX.	单位	
电源电流 <sup>注</sup>	I <sub>DD1</sub>	运行模式	f <sub>XH</sub> =10MHz	方波输入		1.1	2.7	mA
				谐振器连接		1.3	3.2	mA
			f <sub>XH</sub> =5MHz	方波输入		0.63	1.6	mA
				谐振器连接		0.8	2.0	mA
			f <sub>IH</sub> =8MHz			1.2	2.7	mA
			I <sub>DD2</sub>	HALT 模式	f <sub>XH</sub> =10MHz	方波输入		0.13
	谐振器连接					0.36	2.4	mA
	f <sub>XH</sub> =5MHz	方波输入				0.09	0.65	mA
		谐振器连接				0.27	1.1	mA
	f <sub>IH</sub> =8MHz			0.22	1.2	mA		
I <sub>DD3</sub>	STOP 模式	V <sub>DD</sub> =5.0V、 仅 POC 运行时			1.2	10	μA	

注 这是流入内部电源 (V<sub>DD</sub>、AV<sub>REF</sub>) 的总电流。包括将输入引脚固定为 V<sub>DD</sub> 或者 V<sub>SS</sub> 状态下的输入泄露电流。但是，不包括端口的上拉电阻、下拉电阻和输出电流。TYP. 是在 V<sub>DD</sub>=3.0V 状态下仅 CPU 运行时的电流。MAX. 包括外围工作电流。但是，A/D 转换器、看门狗定时器、LVI、运算放大器以及比较器停止。

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 (6/6)

( $T_A = -40 \sim +85^\circ\text{C}$ 、 $1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $AV_{REF} \leq V_{DD}$ 、 $V_{SS} = 0\text{V}$ )

项目	符号	条件		MIN.	TYP.	MAX.	单位
看门狗定时器工作电流注 1	$I_{WDT}$	$V_{DD} = 3.0\text{V}$	以 240kHz 低速内置振荡时钟运行时		5	10	$\mu\text{A}$
LVI 工作电流注 2	$I_{LVI}$				9	18	$\mu\text{A}$
A/D 转换器工作电流注 3	$I_{ADC}$				0.86	1.9	$\text{mA}$
运算放大器工作电流注 3	$I_{AMP}$	1 个运算放大器运行时	$AV_{REF} = V_{DD} = 5.0\text{V}$		250	380	$\mu\text{A}$
			$AV_{REF} = V_{DD} = 3.0\text{V}$		230	321	$\mu\text{A}$
比较器工作电流注 3	$I_{CMP}$	$AV_{REF} = V_{DD} = 5.0\text{V}$			80	240	$\mu\text{A}$
		$AV_{REF} = V_{DD} = 3.0\text{V}$			70	200	$\mu\text{A}$

注 1. 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,  $I_{DD1}$ 、 $I_{DD2}$  或者  $I_{DD3}$  与  $I_{WDT}$  进行加法运算的值为 R7F0C30x、R7F0C31x 的电流值。

2. 这只是流入 LVI 电路的电流。在 LVI 电路运行期间,  $I_{DD1}$ 、 $I_{DD2}$  或者  $I_{DD3}$  与  $I_{LVI}$  进行加法运算的值为 R7F0C30x、R7F0C31x 的电流值。

3. 这是流入  $AV_{REF}$  的电流。

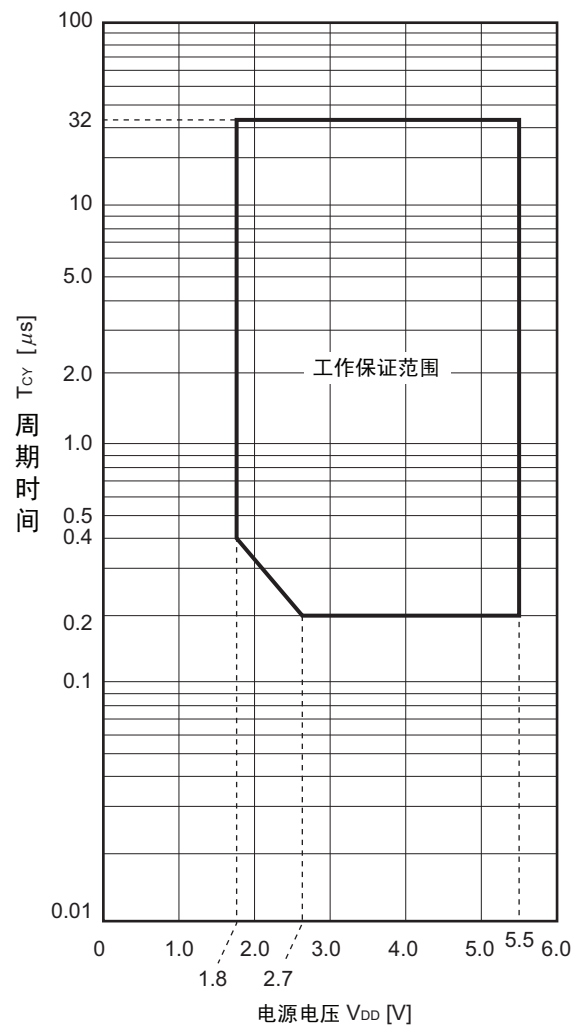
## AC 特性

## (1) 基本运行

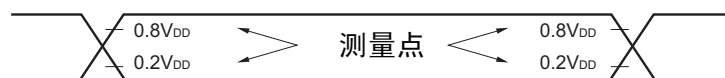
( $T_A = -40 \sim +85^\circ\text{C}$ 、 $1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$ )

项目	符号	条件		MIN.	TYP.	MAX.	单位
指令周期 (最短指令执行时间)	$T_{CY}$	主系统时钟 ( $f_{XP}$ ) 运行	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	0.2		32	$\mu\text{s}$
			$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$	0.4		32	$\mu\text{s}$
外围硬件时钟频率	$f_{PRS}$	$f_{PRS} = f_{XP}$	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$			10	$\text{MHz}$
			$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$			5	$\text{MHz}$
		$f_{PRS} = f_{IH}$	7.76		8.24	$\text{MHz}$	
外部主系统时钟频率	$f_{EXCLK}$	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$		1.0		10.0	$\text{MHz}$
		$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$		1.0		5.0	$\text{MHz}$
外部主系统时钟输入高电平和输入低电平的宽度	$t_{EXCLKH}$ 、 $t_{EXCLKL}$			$(1/f_{EXCLK} \times 1/2) - 1$			$\text{ns}$
TI000、TI010 输入高电平和输入低电平的宽度	$t_{TIH0}$ 、 $t_{TIL0}$	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$		$2/f_{sam} + 0.2$ 注			$\mu\text{s}$
		$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$		$2/f_{sam} + 0.5$ 注			$\mu\text{s}$
TI51 输入频率	$f_{TI5}$	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$				10.0	$\text{MHz}$
		$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$				5.0	$\text{MHz}$
TI51 输入高电平和输入低电平的宽度	$t_{TIH5}$	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$		50			$\text{ns}$
		$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$		100			$\text{ns}$
中断输入高电平和低电平的宽度	$t_{INTH}$ 、 $t_{INTL}$			1			$\mu\text{s}$
RESET 低电平宽度	$t_{RSL}$			10			$\mu\text{s}$

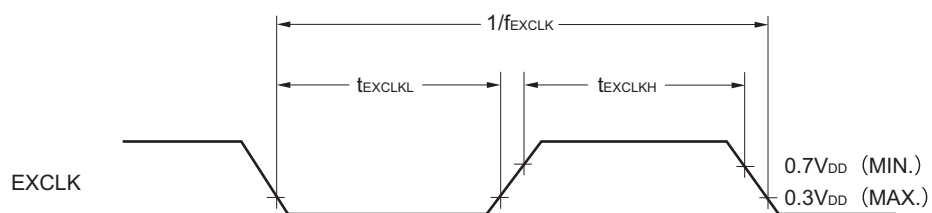
注 通过预分频器模式寄存器 00 (PRM00) 的 bit0 和 bit1 (PRM000、PRM001), 可选择  $f_{sam}$  为  $f_{PRS}$ 、 $f_{PRS}/4$  或者  $f_{PRS}/256$ 。但是, 在选择 TI000 有效边沿作为计数时钟时,  $f_{sam}$  为  $f_{PRS}$ 。

$T_{CY}$  vs  $V_{DD}$  (主系统时钟运行)

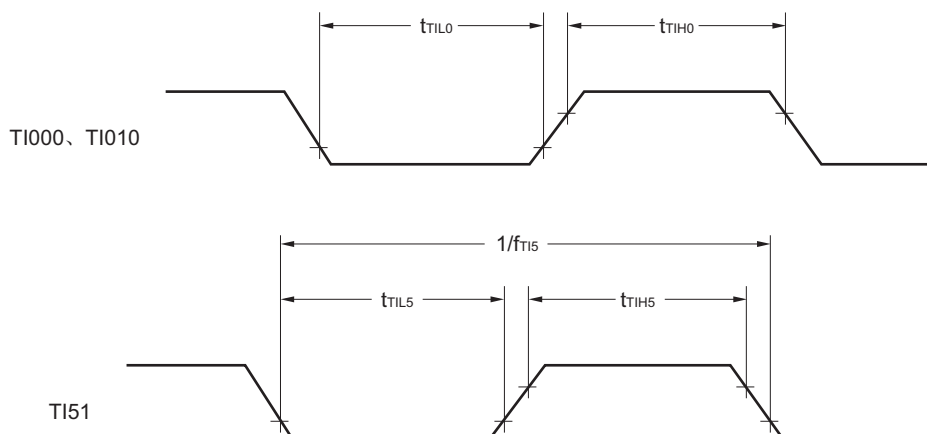
## AC 时序测量点



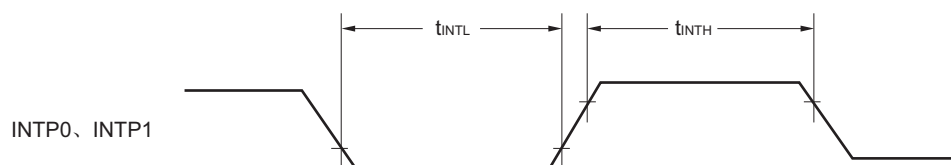
## 外部主系统时钟时序



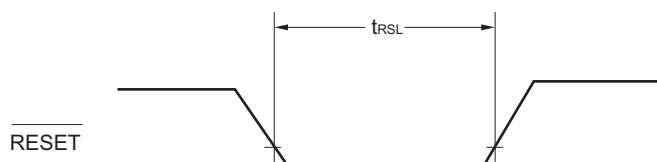
## TI 时序



## 中断请求输入时序



## RESET 输入时序



## (2) 串行接口

( $T_A = -40 \sim +85^\circ\text{C}$ 、 $1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$ )

## (a) 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					312.5	kbps

## (b) OCD (UART0)

项目	符号	条件	MIN.	TYP.	MAX.	单位
传送速率			$f_{CLK}/32$		$f_{CLK}/8$	bps
		OCD 模式中 ( $f_{CLK}=4\text{M}$ 、 $V_{DD} \geq 2.7\text{V}$ 、 $C_b=50\text{pF}$ )	125/250/500			kbps
		编程模式中 ( $f_{CLK}=4\text{M}$ 、 $V_{DD} \geq 2.7\text{V}$ 、 $C_b=50\text{pF}$ )	125/250/500			kbps

## 模拟特性

## (1) A/D 转换器 (SSOP 封装产品)

(T<sub>A</sub> = -40 ~ +85°C、2.3V ≤ AV<sub>REF</sub> ≤ V<sub>DD</sub> ≤ 5.5V、V<sub>SS</sub> = 0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	R <sub>ES</sub>				10	bit
总误差注 1、注 2	A <sub>INL</sub>	4.0V ≤ AV <sub>REF</sub> ≤ 5.5V			±0.4	%FSR
		2.7V ≤ AV <sub>REF</sub> ≤ 5.5V			±0.6	%FSR
		2.3V ≤ AV <sub>REF</sub> < 2.7V			±1.2	%FSR
转换时间	t <sub>CONV</sub>	4.0V ≤ AV <sub>REF</sub> ≤ 5.5V	6.1		36.7	μs
		2.7V ≤ AV <sub>REF</sub> ≤ 5.5V	6.1		36.7	μs
		2.3V ≤ AV <sub>REF</sub> < 2.7V	27		66.6	μs
零刻度误差注 1、注 2	E <sub>ZS</sub>	4.0V ≤ AV <sub>REF</sub> ≤ 5.5V			±0.4	%FSR
		2.7V ≤ AV <sub>REF</sub> ≤ 5.5V			±0.6	%FSR
		2.3V ≤ AV <sub>REF</sub> < 2.7V			±0.6	%FSR
满刻度误差注 1、注 2	E <sub>PS</sub>	4.0V ≤ AV <sub>REF</sub> ≤ 5.5V			±0.4	%FSR
		2.7V ≤ AV <sub>REF</sub> ≤ 5.5V			±0.6	%FSR
		2.3V ≤ AV <sub>REF</sub> < 2.7V			±0.6	%FSR
模拟输入电压	V <sub>AIN</sub>		AV <sub>SS</sub>		AV <sub>REF</sub>	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。

## (2) A/D 转换器 (SOP 封装产品)

(T<sub>A</sub> = -40 ~ +85°C、2.3V ≤ AV<sub>REF</sub> ≤ V<sub>DD</sub> ≤ 5.5V、V<sub>SS</sub> = 0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	R <sub>ES</sub>				10	bit
总误差注 1、注 2	A <sub>INL</sub>	4.0V ≤ AV <sub>REF</sub> ≤ 5.5V			±0.2 注 3	%FSR
		2.7V ≤ AV <sub>REF</sub> ≤ 5.5V			±0.3 注 3	%FSR
		2.3V ≤ AV <sub>REF</sub> < 2.7V			±0.6 注 3	%FSR
转换时间	t <sub>CONV</sub>	4.0V ≤ AV <sub>REF</sub> ≤ 5.5V	6.1		36.7	μs
		2.7V ≤ AV <sub>REF</sub> ≤ 5.5V	6.1		36.7	μs
		2.3V ≤ AV <sub>REF</sub> < 2.7V	27		66.6	μs
零刻度误差注 1、注 2	E <sub>ZS</sub>	4.0V ≤ AV <sub>REF</sub> ≤ 5.5V			±0.2 注 3	%FSR
		2.7V ≤ AV <sub>REF</sub> ≤ 5.5V			±0.3 注 3	%FSR
		2.3V ≤ AV <sub>REF</sub> < 2.7V			±0.3 注 3	%FSR
满刻度误差注 1、注 2	E <sub>PS</sub>	4.0V ≤ AV <sub>REF</sub> ≤ 5.5V			±0.2 注 3	%FSR
		2.7V ≤ AV <sub>REF</sub> ≤ 5.5V			±0.3 注 3	%FSR
		2.3V ≤ AV <sub>REF</sub> < 2.7V			±0.3 注 3	%FSR
模拟输入电压	V <sub>AIN</sub>		AV <sub>SS</sub>		AV <sub>REF</sub>	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。  
 3. MAX. 值是正态分布平均值 ±3σ 的值。未进行出厂检查。

## (3) 运算放大器 0 和运算放大器 1

( $T_A = -40 \sim +85^\circ\text{C}$ 、 $2.2\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $2.2\text{V} \leq AV_{REF} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$ 、输出负载:  $R_L = 47\text{k}\Omega$ 、 $C_L = 50\text{pF}$ )

项目	符号	条件	MIN.	TYP.	MAX.	单位
输入偏移电压注 1	$V_{IOP0}$	$V_{BIAS} = 1/2V_{DD}$ 、 $AV_{REF} = 3.0\text{V}$			$\pm 3$	mA
电源电压抑制比	$PSRR_{OP0}$	$AV_{REF} = 3.0\text{V}$		70		dB
高电平输出电压	$V_{OHOP0}$	$AV_{REF} = 3.0\text{V}/2.2\text{V}$ 、 $I_{OH} = -500\mu\text{A}$	$AV_{REF} - 0.2$			V
低电平输出电压	$V_{OLOP0}$	$AV_{REF} = 3.0\text{V}/2.2\text{V}$ 、 $I_{OL} = 500\mu\text{A}$			0.1	V
同相输入电压	$V_{ICMOP0}$	$AV_{REF} = 3.0\text{V}/2.2\text{V}$	0		$AV_{REF} - 0.6$	V
转换速率	$SR_{OP0}$	$AV_{REF} = 3.0\text{V}$		1.8		V/ $\mu\text{s}$
		$AV_{REF} = 5.0\text{V}$		2.0		V/ $\mu\text{s}$
输入噪声密度 (Inoise)		$AV_{REF} = 3.0\text{V}$ 、 $V_{IN} = 0.1\text{V}$ 、 $f = 1\text{kHz}$		73		nV/ $\sqrt{\text{Hz}}$
		$AV_{REF} = 3.0\text{V}$ 、 $V_{IN} = AV_{REF}/2\text{V}$ 、 $f = 1\text{kHz}$		60		
		$AV_{REF} = 3.0\text{V}$ 、 $V_{IN} = AV_{REF} - 0.6\text{V}$ 、 $f = 1\text{kHz}$		55		
相位裕度		$AV_{REF} = 3.0\text{V}$		40		deg
大振幅电压增益	$AV_{OP0}$	$AV_{REF} = 3.0\text{V}$		100		dB
增益带宽积	$GBW_{OP0}$	$AV_{REF} = 5.0\text{V}/3.0\text{V}/2.2\text{V}$		3.0		MHz
运行稳定等待时间注 2	$t_{OP0}$	$AV_{REF} = 3.0\text{V}$		10		$\mu\text{s}$

注 1. 这是特性评估结果。没有进行出厂检查。

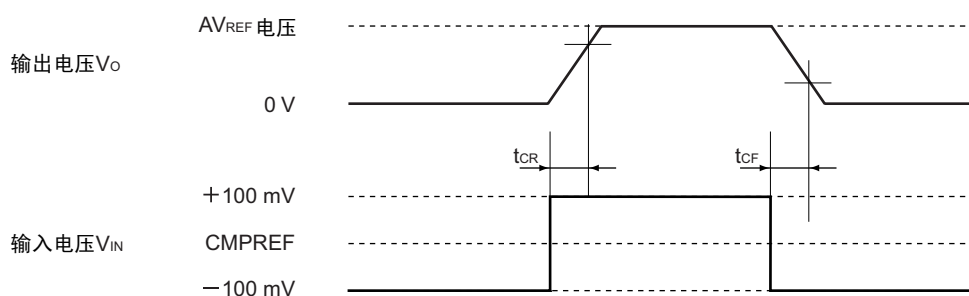
2. 这是从运算放大器 (OPAMP0E/OPAMP1E=1) 运行后, 到达到满足运算放大器运行的 DC/AC 特性的状态所需的时间。

## (4) 比较器

( $T_A = -40 \sim +85^\circ\text{C}$ 、 $2.7\text{V} \leq V_{DD} \leq AV_{REF} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$ )

项目	符号	条件	MIN.	TYP.	MAX.	单位
输入偏移电压	$V_{IOCMP}$			$\pm 5$	$\pm 40$	mV
输入电压范围	$V_{ICMP}$	CMPIN	0		$AV_{REF}$	V
		CMPCOM	0.045		$0.9AV_{REF}$	V
响应时间	$t_{CR}$ 、 $t_{CF}$	输入振荡宽度 $\pm 100\text{mV}$		70	150	ns
运行稳定等待时间注	$t_{CMP}$				1	$\mu\text{s}$
比较器输入高电平和低电平的宽度	$t_{CMPL}$		125			ns

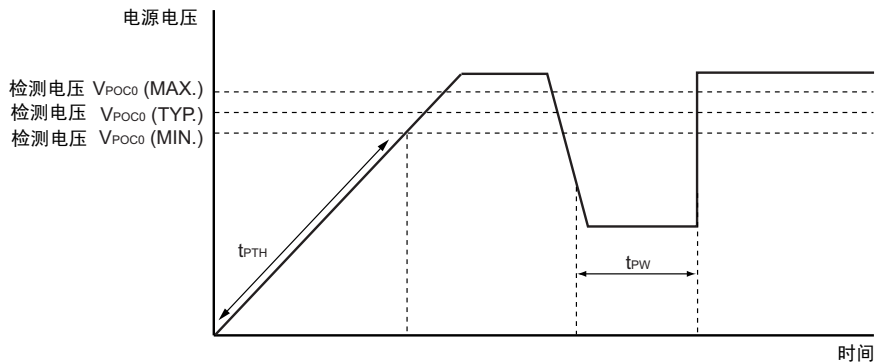
注 这是从允许比较器 (CMP0EN 位 = 1) 运行后, 到达到满足比较器的 DC/AC 特性的状态所需的时间。



(5) POC  
( $T_A = -40 \sim +85^\circ\text{C}$ 、 $V_{SS} = 0\text{V}$ )

项目	符号	条件	MIN.	TYP.	MAX.	单位
检测电压	$V_{POC0}$		1.44	1.59	1.74	V
电源电压上升斜率	$t_{PTH}$	$V_{DD}: 0\text{V} \rightarrow V_{POC0}$ 的变化斜率	0.5			V/ms
最小脉宽	$t_{PW}$	电源下降时	200			$\mu\text{s}$

## POC 电路时序

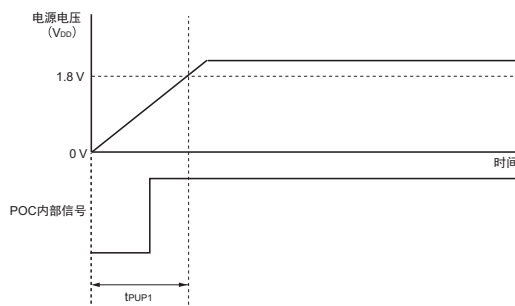
(6) 电源电压上升时间  
( $T_A = -40 \sim +85^\circ\text{C}$ 、 $V_{SS} = 0\text{V}$ )

项目	符号	条件	MIN.	TYP.	MAX.	单位
上升到 1.8V( $V_{DD}(\text{MIN.})$ ) 的最长时间注 ( $V_{DD}: 0\text{V} \rightarrow 1.8\text{V}$ )	$t_{PUP1}$	LVI 默认启动无效 ( $\text{LVISTART}$ (选项字节) = 0)、 未使用 $\overline{\text{RESET}}$ 输入时			3.6	ms
上升到 1.8V( $V_{DD}(\text{MIN.})$ ) 的最长时间注 ( $\overline{\text{RESET}}$ 输入解除 $\rightarrow V_{DD}: 1.8\text{V}$ )	$t_{PUP2}$	LVI 默认启动无效 ( $\text{LVISTART}$ (选项字节) = 0)、 使用 $\overline{\text{RESET}}$ 输入时			1.9	ms

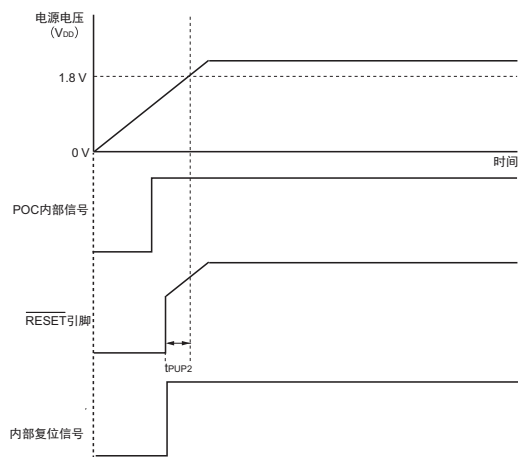
注 必须在比该时间短的时间内上电。

## 电源电压上升时间的时序

• 未使用  $\overline{\text{RESET}}$  引脚输入时



• 使用  $\overline{\text{RESET}}$  引脚输入时(在解除POC后,解除因RESET引脚引起的外部复位)





## (7) LVI

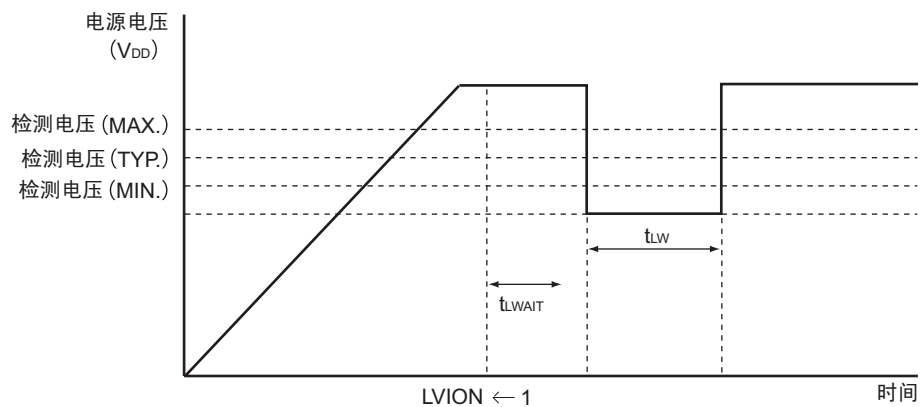
(T<sub>A</sub>=-40 ~ +85°C、V<sub>POC</sub> ≤ V<sub>DD</sub> ≤ 5.5V、V<sub>SS</sub>=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位	
检测电压	检测电压电平	V <sub>LVI0</sub>		4.24±0.1		V	
		V <sub>LVI1</sub>		4.09±0.1		V	
		V <sub>LVI2</sub>		3.93±0.1		V	
		V <sub>LVI3</sub>		3.78±0.1		V	
		V <sub>LVI4</sub>		3.62±0.1		V	
		V <sub>LVI5</sub>		3.47±0.1		V	
		V <sub>LVI6</sub>		3.32±0.1		V	
		V <sub>LVI7</sub>		3.16±0.1		V	
		V <sub>LVI8</sub>		3.01±0.1		V	
		V <sub>LVI9</sub>		2.85±0.1		V	
		V <sub>LVI10</sub>		2.70±0.1		V	
		V <sub>LVI11</sub>		2.55±0.1		V	
		V <sub>LVI12</sub>		2.39±0.1		V	
		V <sub>LVI13</sub>		2.24±0.1		V	
		V <sub>LVI14</sub>		2.08±0.1		V	
V <sub>LVI15</sub>		1.93±0.1		V			
	电源上升时的 电源电压	V <sub>DDLVI</sub>	LVI 默认启动功能运行时 (LVISTART=1)	2.5	2.7	2.9	V
最小脉宽	t <sub>LW</sub>		200			μs	
运行稳定等待时间	t <sub>LWAIT</sub>				10	μs	

注 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<sub>LVI(n-1)</sub> > V<sub>LVI n</sub>: n=1 ~ 15

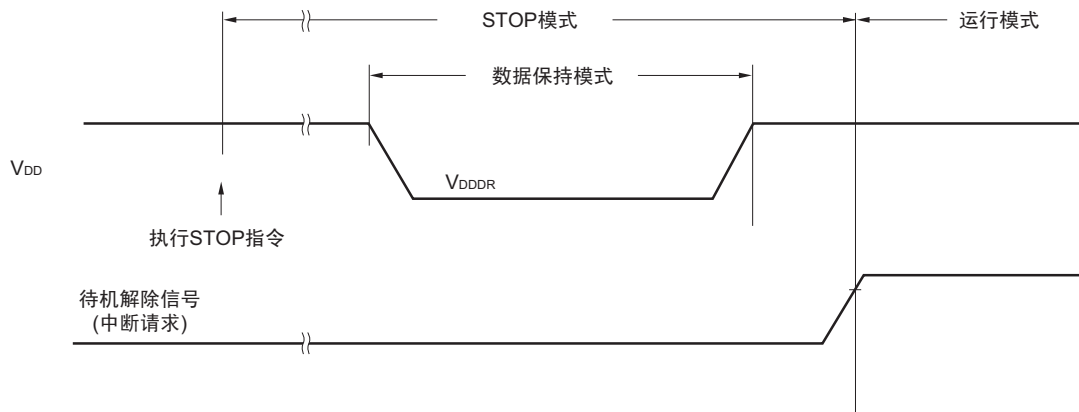
## LVI 电路时序



数据存储器 STOP 模式低电源电压数据保持特性 ( $T_A=-40 \sim +85^\circ\text{C}$ )

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

( $T_A=-40 \sim +85^\circ\text{C}$ 、 $2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS}=0\text{V}$ )

## • 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
擦除时间	片单位	$t_{eraca}$				20	200	ms
	扇区单位	$t_{erasa}$				20	200	ms
编程时间 (8 位单位)	$t_{wrwa}$					10	100	$\mu\text{s}$
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	使用闪存编程器和瑞萨电子提供的自编程库时	保持 15 年	100			次
工作温度		使用闪存编程器时: $10 \sim 40^\circ\text{C}$ 、使用自编程时: $-40 \sim +85^\circ\text{C}$						

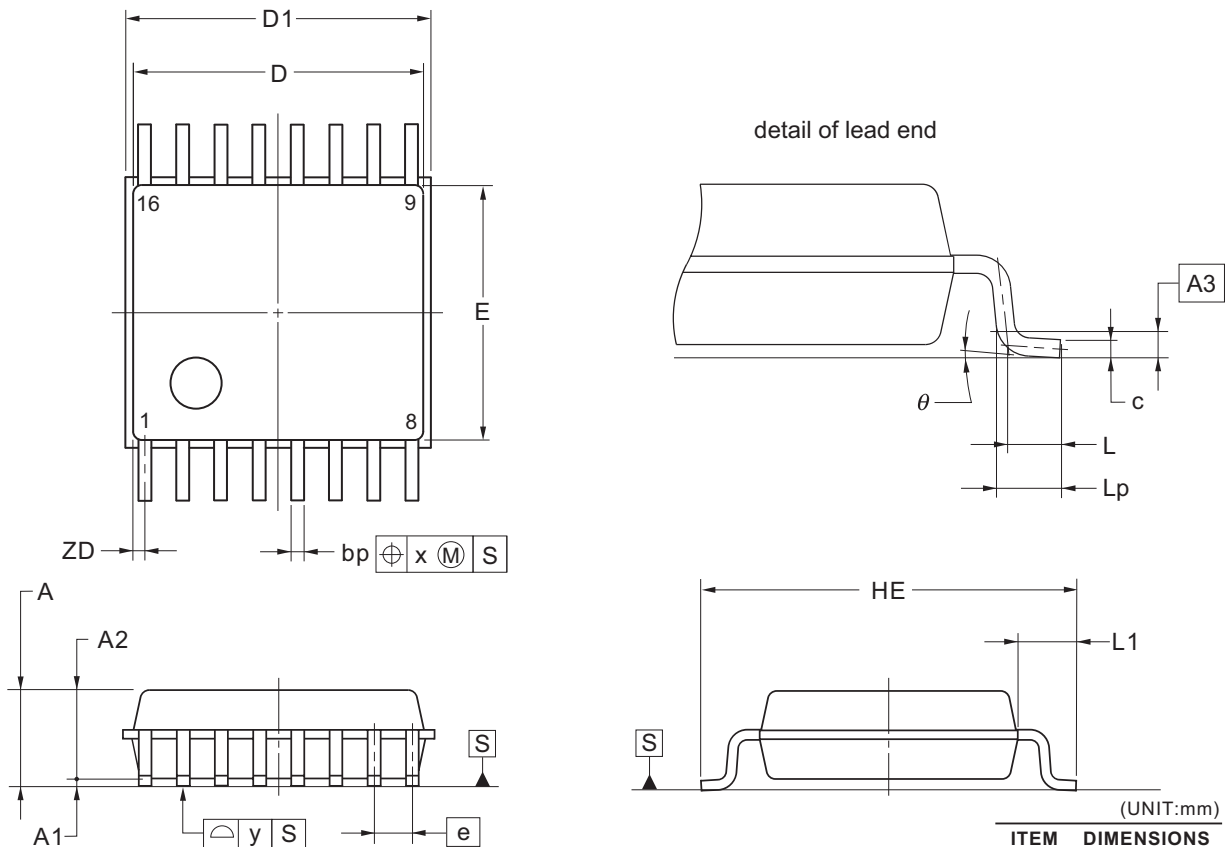
注 在对出厂产品进行第一次编程时，不管是“擦除→编程”还是“只编程”，都被作为 1 次改写。

## 第 25 章 封装尺寸图

## 25.1 16 引脚产品

- R7F0C30042ESP、R7F0C30342ESP、R7F0C30142ESP、R7F0C30442ESP、R7F0C30242ESP、R7F0C30542ESP

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-SSOP16-4.4x5-0.65	PRSP0016JC-A	P16MA-65-FAA-2	0.08



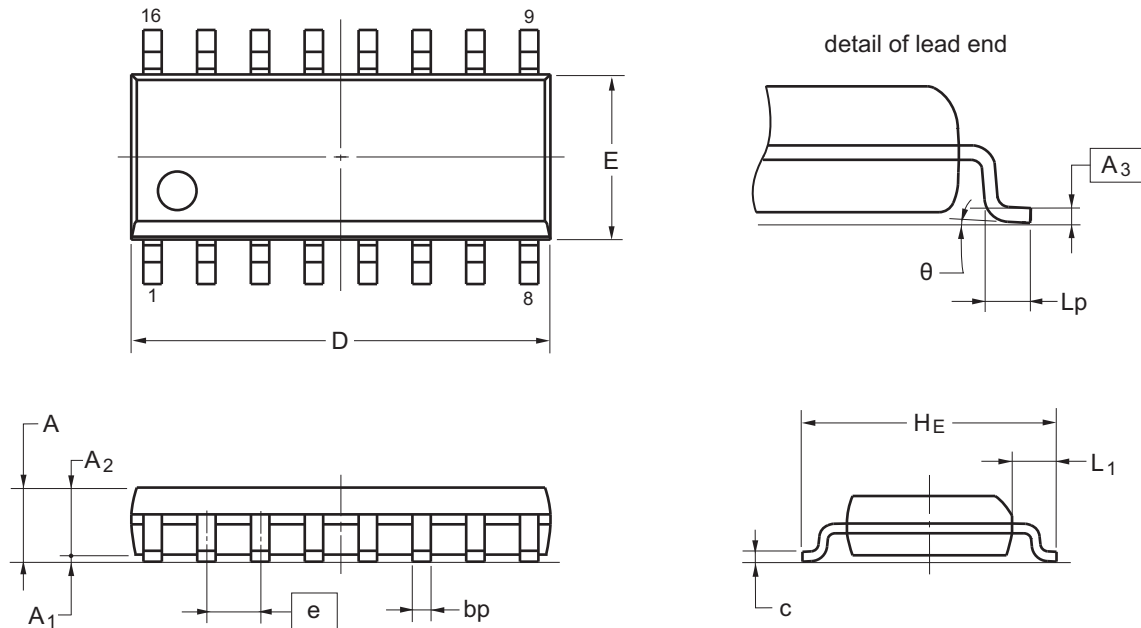
(UNIT:mm)

ITEM	DIMENSIONS
D	5.00±0.15
D1	5.20±0.15
E	4.40±0.20
HE	6.40±0.20
A	1.725 MAX.
A1	0.125±0.05
A2	1.50
A3	0.25
e	0.65
bp	0.22 <sup>+0.08</sup> <sub>-0.07</sub>
c	0.15 <sup>+0.03</sup> <sub>-0.04</sub>
L	0.50
Lp	0.60±0.10
L1	1.00±0.20
x	0.13
y	0.10
θ	3° <sup>+5°</sup> <sub>-3°</sub>
ZD	0.325

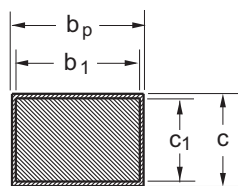
©2012 Renesas Electronics Corporation. All rights reserved.

- R7F0C30642ESN、R7F0C30942ESN、R7F0C30742ESN、R7F0C31042ESN、R7F0C30842ESN、R7F0C31142ESN

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-SOP16-3.9x9.9-1.27	PRSP0016DP-A	T16MF-127-DAA	0.15



Terminal cross section

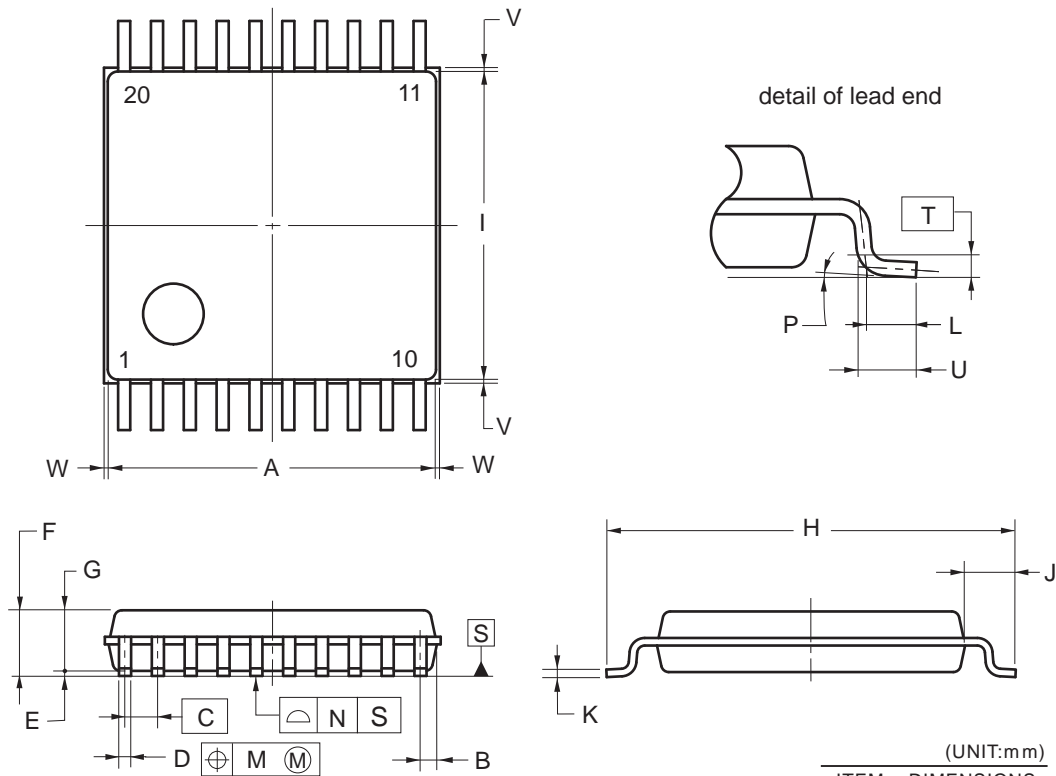


Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	9.70	9.90	10.10
E	3.70	3.90	4.10
A <sub>2</sub>	1.30	1.40	1.50
A <sub>1</sub>	0.10	—	0.225
A	—	—	1.75
A <sub>3</sub>	—	0.25	—
bp	0.39	—	0.48
b <sub>1</sub>	0.38	0.41	0.43
c	0.21	—	0.26
c <sub>1</sub>	0.19	0.20	0.21
θ	0°	—	8°
H <sub>E</sub>	5.80	6.00	6.20
L <sub>p</sub>	0.50	—	0.80
e	—	1.27	—
L <sub>1</sub>	—	1.05	—

## 25.2 20 引脚产品

- R7F0C30062ESP、R7F0C30362ESP、R7F0C30162ESP、R7F0C30462ESP、R7F0C30262ESP、R7F0C30562ESP

## 20-PIN PLASTIC SSOP (7.62 mm (300))



## NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

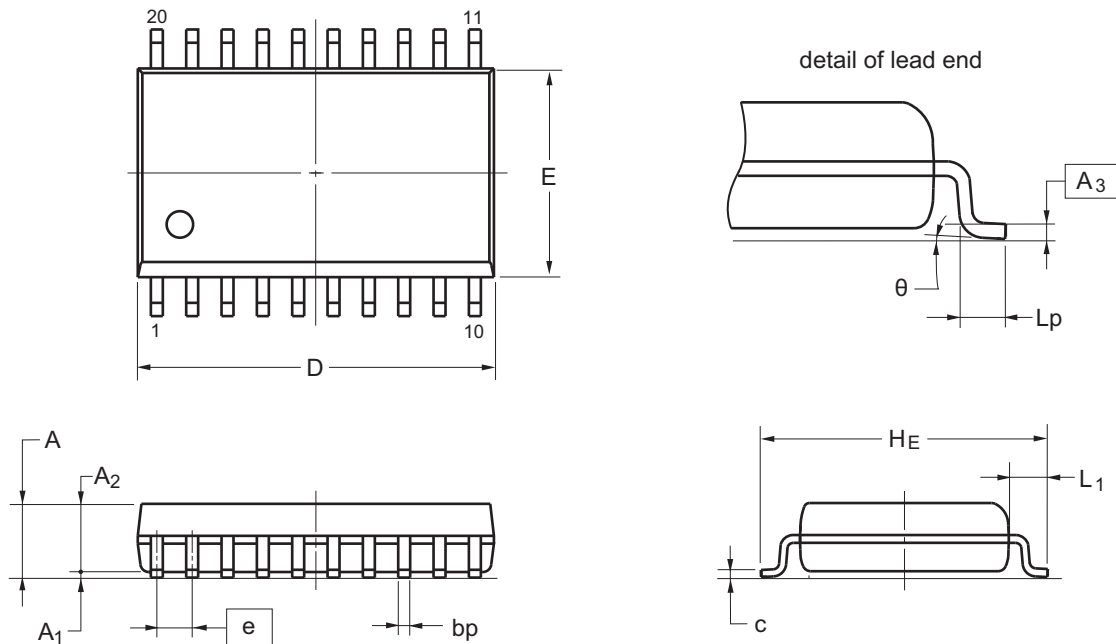
(UNIT:mm)	
ITEM	DIMENSIONS
A	6.50±0.10
B	0.325
C	0.65 (T.P.)
D	0.22 <sup>+0.10</sup> <sub>-0.05</sub>
E	0.10±0.05
F	1.30±0.10
G	1.20
H	8.10±0.20
I	6.10±0.10
J	1.00±0.20
K	0.15 <sup>+0.05</sup> <sub>-0.01</sub>
L	0.50
M	0.13
N	0.10
P	3° <sup>+5°</sup> <sub>-3°</sub>
T	0.25(T.P)
U	0.60±0.15
V	0.25 MAX.
W	0.15 MAX.

P20MC-65-CAA-1

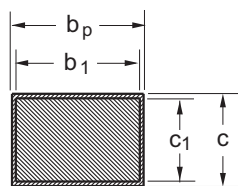
©2010 Renesas Electronics Corporation. All rights reserved.

- R7F0C30662ESN、R7F0C30962ESN、R7F0C30762ESN、R7F0C31062ESN、R7F0C30862ESN、R7F0C31162ESN

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-SOP20-7.5x12.8-1.27	PRSP0020DP-A	T20MR-127-PAB	0.54



Terminal cross section



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	12.60	12.80	13.00
E	7.30	7.50	7.70
A <sub>2</sub>	2.25	2.30	2.35
A <sub>1</sub>	0.10	—	0.30
A	—	—	2.65
A <sub>3</sub>	—	0.25	—
bp	0.35	—	0.44
b <sub>1</sub>	0.34	0.37	0.39
c	0.25	—	0.31
c <sub>1</sub>	0.24	0.25	0.26
θ	0°	—	8°
H <sub>E</sub>	10.10	10.30	10.50
L <sub>p</sub>	0.70	—	1.00
e	—	1.27	—
L <sub>1</sub>	—	1.40	—

## 第 26 章 等待的注意事项

### 26.1 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 26-1）。在要求执行实时处理时，必须注意。

### 26.2 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 26-1 所示。

表 26-1 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）
A/D 转换器	ADM0	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）
	ADCRH	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 $\leq 0.5$ 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 ( $f_{PRS} \sim f_{PRS}/12$ ) $f_{CPU}$ : 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 ( $f_{RPS}/12$ ) • 最小次数: CPU 最低速 ( $f_{SUB}$ )、A/D 转换时钟最高速 ( $f_{RPS}$ )			

**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 ( $f_{CPU}$ )。

## 附录 A 修订记录

Rev.	发行日	修订内容	
		页	修订处
1.00	2012.12.28	—	初版发行
1.10	2014.01.21	全文 全文 3 11 14 20 349 364 399 ~ 402	将“R7F0C30xx”更改为“R7F0C30x、R7F0C31x”。 追加 SOP 封装产品。 追加图 1-1。 删除 2.1.1(2)(1/2) 表中 REGC 功能中的“1.9V”。 删除 2.1.2(2)(2/2) 表中 REGC 功能中的“1.9V”。 删除 2.2.5(a) 中的“1.9V”。 删除 19.1 的第二段和 19.2。 删除 21.7 的注意 2，更改注意 3。 替换第 25 章。
1.20	2014.10.22	42 74  87 88 89  94 96 267  328	删除表 3-6(2/4) 中“FF62H”的注 2。 在 4.2.2 中追加“P31/TxD0/(TO00)/CMPCOM 和 P32/RxD0/(TI010)/CMPIN 的引脚功能设定请参照“12.3 比较器的控制寄存器”的表 12-2 和表 12-3。”。 在图 4-18 中的“PM31”下追加“注 2”。 删除图 4-19 中“PM33”下的“注 2”，并且在“P31”下追加“注 2”。 在图 4-20 中的“PM31”下追加“注 1”；删除图 4-21 中“PM33”下的“注 1”，并且在“P31”下追加“注 1”。 在表 4-12(1/2) 中追加注 3。 在表 4-13(1/2) 中追加注 3。 在 12.3(2) 中追加“对于没有内置比较器的产品（R7F0C300x、R7F0C301x、R7F0C302x、R7F0C306x、R7F0C307x、R7F0C308x），复位后数字功能处于无效状态。因此在用作数字功能时，必须将 CMPCOMPC 和 CMPINPC 置“1”。”；在图 12-3 下追加注意。 在表 16-2(2/3) 中追加注 4。



---

R7F0C30x、R7F0C31x  
用户手册 硬件篇

Publication Date: Rev.1.20 Oct 22, 2014

Published by: Renesas Electronics Corporation

---

**SALES OFFICES****Renesas Electronics Corporation**<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.**Renesas Electronics America Inc.**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 301, Tower A, Central Towers, 555 LanGao Rd., Putuo District, Shanghai, China  
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999**Renesas Electronics Hong Kong Limited**Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2886-9318, 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.**12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

R7F0C30x、 R7F0C31x



瑞萨电子株式会社

R01UH0389CJ0120