

To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. 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 categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
 - “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.
 - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
 - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system manufactured by you.
10. 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.
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.

μ PD179F11x, 179F12x 微控制器

8 位单片微控制器

μ PD179F110

μ PD179F111

μ PD179F112

μ PD179F113

μ PD179F114

μ PD179F122

μ PD179F123

μ PD179F124

[备忘录]

CMOS 设备注意事项

① 输入引脚处的电压波形

输入噪音或一个反射波引起的波形失真可能导致错误发生。如果由于噪音等的影响使CMOS设备的输入电压范围保持在VIL(MAX)和VIH(MIN)之间,设备可能发生错误。在输入电平固定时以及输入电平从VIL(MAX)过渡到VIH(MIN)时的传输期间,要防止散射噪声影响设备。

② 未使用的输入引脚的处理

CMOS设备的输入端保持开路可能导致误操作。如果一个输入引脚未被连接,则由于噪音等原因可能会产生内部输入电平,从而导致误操作。CMOS设备的操作特性与Bipolar或NMOS设备不同。CMOS设备的输入电平必须借助上拉或下拉电路固定在高电平或低电平。每一个未使用引脚都应该通过附加电阻连接到VDD或GND。如果有可能尽量定义为输出引脚。对未使用引脚的处理因设备而异,必须遵循与设备相关的规定和说明。

③ ESD防护措施

如果MOS设备周围有强电场,将会击穿氧化栅极,从而影响设备的运行。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。对于环境必须有适当的控制。如果空气干燥,应当使用增湿器。建议避免使用容易产生静电的绝缘体。半导体设备的存放和运输必须使用抗静电容器、抗静电屏蔽袋或导电材料容器。所有的测试和测量工具包括工作台和工作面必须良好接地。操作员应当佩戴静电消除手带以保证良好接地。不能用手直接接触半导体设备。对于装配有半导体设备的PW板也应采取类似的静电防范措施。

④ 初始化之前的状态

在上电时MOS设备的初始状态是不确定的。在刚刚上电之后,具有复位功能的MOS设备并没有被初始化。因此上电不能保证输出引脚的电平,I/O设置和寄存器的内容。设备在收到复位信号后才进行初始化。具有复位功能的设备在上电后必须立即进行复位操作。

⑤ 电源开关顺序

在一个设备的内部操作和外部接口使用不同的电源的情况下,按照规定,应先在接通内部电源之后再接通外部电源。当关闭电源时,按照规定,先关闭外部电源再关闭内部电源。如果电源开关顺序颠倒,可能会导致设备的内部组件过电压,产生异常电流,从而引起内部组件的误操作和性能的退化。对于每个设备电源的正确开关顺序必须依据设备的规范说明分别进行判断。

⑥ 电源关闭状态下的输入信号

不要向没有加电的设备输入信号或提供I/O上拉电源。因为输入信号或提供I/O上拉电源将引起电流注入,从而引起设备的误操作,并产生异常电流,从而使内部组件退化。每个设备电源关闭时的信号输入必须依据设备的规范说明分别进行判断。

Windows和WindowsNT是Microsoft Corporation在美国及其他国家的注册商标和商标。

PC/AT是International Business Machines Corporation的商标。

HP9000系列700和HP-UX是Hewlett-Packard Company的商标。

SPARCstation是SPARC International, Inc.的商标。

Solaris和SunOS是Sun Microsystems, Inc.的商标。

SuperFlash®是 Silicon Storage Technology, Inc.的一个注册商标，已经在美国和日本等几个国家使用。

注意事项：该产品使用的SuperFlash®技术获得了Silicon Storage Technology, Inc.公司的授权。

- 本文档所登载的内容有效期截止至 2009 年 2 月，信息先于产品的生产周期发布。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。否则因本文档所登载内容引发的错误，本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的，对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或其他人所有的专利、版权以及其它知识产权作出任何明示或默示的许可及授权。
- 本文件中的电路、软件以及相关信息仅用以说明半导体产品的运作和应用实例。用户如在设备设计中应用本文件中的电路、软件以及相关信息，应自行负责。对于用户或其他人因使用了上述电路、软件以及相关信息而引起的任何损失，本公司概不负责。
- 虽然本公司致力于提高半导体产品的质量及可靠性，但用户应同意并知晓，我们仍然无法完全消除出现产品缺陷的可能。为了最大限度地减少因本公司半导体产品故障而引起的对人身、财产造成损害（包括死亡）的危险，用户务必在其设计中采用必要的安全措施，如冗余度、防火和防故障等安全设计。
- 本公司产品质量分为：

“标准等级”、“专业等级”以及“特殊等级”三种质量等级。

“特殊等级”仅适用于为特定用途而根据用户指定的质量保证程序所开发的日电电子产品。另外，各种日电电子产品的推荐用途取决于其质量等级，详见如下。用户在选用本公司的产品时，请事先确认产品的质量等级。

“标准等级”： 计算机，办公自动化设备，通信设备，测试和测量设备，音频·视频设备，家电，加工机械以及产业用机器人。

“专业等级”： 运输设备（汽车、火车、船舶等），交通信号控制设备，防灾装置，防止犯罪装置，各种安全装置以及医疗设备（不包括专门为维持生命而设计的设备）。

“特殊等级”： 航空器械，宇航设备，海底中继设备，原子能控制系统，为了维持生命的医疗设备、用于维持生命的装置或系统等。

除在本公司半导体产品的数据表或数据手册等资料中另有特别规定以外，本公司半导体产品的质量等级均为“标准等级”。如果用户希望在本公司设计意图以外使用本公司半导体产品，务必事先与本公司销售代表联系以确认本公司是否同意为该项应用提供支持。

(注)

- (1) 本声明中的“本公司”是指日本电气电子株式会社（NEC Electronics Corporation）及其控股公司。
- (2) 本声明中的“本公司产品”是指所有由日本电气电子株式会社开发或制造的产品或为日本电气电子株式会社（定义如上）开发或制造的产品。

前言

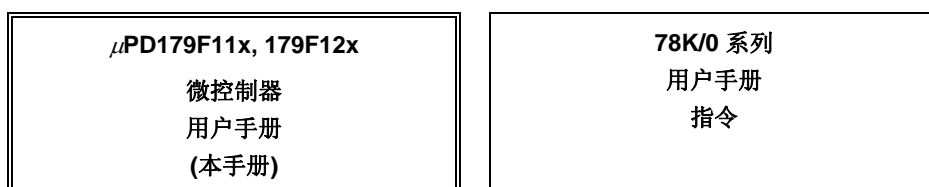
读者 本手册旨在向希望了解 μ PD179F11x, 179F12x 微控制器的功能, 设计, 开发应用系统及用于这些设备的程序的工程师.

目标产品如下所示

μ PD179F11x, 179F12x 微控制器: μ PD179F110, 179F111, 179F112, 179F113,
179F114, 179F122, 179F123, 179F124

目的 本手册旨在提供给用户功能描述的了解, 如下面**组织**中所示

组织 μ PD179F11x, 179F12x 微控制器手册分成两个部分: 本手册和指令手册 (与 78K0 微控制器通用).



- 引脚功能
- 内置 block 功能
- 中断
- 其它片上外围功能
- 电气特性
- CPU 功能
- 指令设置
- 各指令的扩展

如何阅读本手册 在阅读本手册前, 读者应掌握电子工程、逻辑电路和微控制器等电子工程方面的基础知识。

- 要了解功能的概要:
 - 请按**目录**顺序阅读本手册. “<R>”表示主要修点. 在 PDF 文档中通过复制“<R>”, 并指定到查找区能方便的查询到修改点。
- 如何阅读寄存器格式:
 - 尖括号(<>)中的二进制位名称在 RA78K0S 中被定义为保留字, 并且在 CC78K0S 中用#pragma sfr 指令定义为一个 sfr 变量。
- 有关 78K0 微控制器指令的详细内容:
 - 请参考 **78K/0 系列指令用户手册 (U12326E)**.

规则	据规则:	数据的高位部分在左边, 低位部分在右边
	有效低电平表示法:	xxx (在引脚和信号名称上加划一条线)
	注:	文中用 注 标注的相关术语的脚注
	注意事项:	需要特别关注的信息
	备注:	补充信息
	数值的表示:	二进制 ... xxxx 或 xxxxB
		十进制 ... xxxx
		十六进制 ... xxxxH

相关文档 请阅读下列与本手册有关的文献
本手册中提到的相关文档可能包括有初稿版本。但是, 初稿版本没有特别注明。

设备相关文档

文档名称	文档名称
μPD179F11x, 179F12x 微控制器用户手册	本手册
78K/0 系列指令用户手册	U12326E

开发工具 (软件)相关文档 (用户手册)

文档名称	文档名称	文档名称
RA78K0 Ver. 3.80 汇编包	操作篇	U17199E
	语言篇	U17198E
	结构化汇编语言	U17197E
CC78K0 Ver. 3.70 C 编译器	操作篇	U17201E
	语言篇	U17200E
ID78K0-QB Ver. 3.00 综合调试器	操作篇	U18492E
PM+ Ver. 6.00		U17178E

注意事项 上面所列相关文档可能会有新的版本, 请确认使用最新版本的文档进行设计、开发等。

开发工具 (硬件)相关文档 (用户手册)

文档名称	文档名称
QB-179F124 片上仿真器	U18586E
QB-MINI2 具有编程功能的片上调试仿真器	U18371E

Flash 存储器编程器相关文档 (用户手册)

文档名称	文档名称
PG-FP4 Flash 存储器编程器	U15260E
PG-FP5 Flash 存储器编程器	U18865E

其它文档

文档名称	文档名称
半导体选择指南 - 产品和封装	X13769X
半导体设备装配手册	注
NEC 半导体设备的质量等级	C11531E
NEC 半导体设备可靠性/质量控制系统	C10983E
半导体设备防静电 ESD 指南	C11892E

注 可参阅“半导体设备装配手册”网站 (<http://www.necel.com/pkg/en/mount/index.html>)。

注意事项 以上列出的相关文档可能会在无任何声明条件下修改。读者开发设计时，应该使用每个文档的最新版本。

目录

第一章 概述	15
1.1 特征.....	15
1.2 应用.....	15
1.3 订购信息	16
1.4 应用电路.....	16
1.5 引脚配置（俯视图）	17
1.6 μ PD179F11x, 179F12x 微控制器系列	19
1.7 框图.....	19
1.8 功能概述	20
第二章 引脚功能	22
2.1 引脚功能列表	22
2.2 引脚功能描述	24
2.2.1 P00 到 P07（端口 0）	24
2.2.2 P10 到 P17（端口 1）	25
2.2.3 P20 到 P27（端口 2）	25
2.2.4 P30 到 P35（端口 3）（仅适用于 38 引脚产品）	26
2.2.5 P120 到 P123（端口 12）	26
2.2.6 REGC	27
2.2.7 VDD	27
2.2.8 VSS.....	27
2.2.9 FLMD0.....	27
2.3 I/O 引脚电路以及不使用引脚的推荐连接	27
第三章 CPU 架构	30
3.1 存储空间	30
3.1.1 内部程序存储空间.....	37
3.1.2 内部数据存储空间.....	38
3.1.3 特殊功能寄存器（SFR）区域.....	38
3.1.4 数据存储寄存器寻址	38
3.2 处理器寄存器	44
3.2.1 控制寄存器	44
3.2.2 通用寄存器	48
3.2.3 特殊功能寄存器（SFR）	49
3.3 指令地址寻址	53
3.3.1 相对寻址	53
3.3.2 立即寻址.....	54
3.3.3 表间接寻址	55
3.3.4 寄存器寻址	55
3.4 操作数地址寻址	56
3.4.1 隐含寻址.....	56
3.4.2 寄存器寻址	57
3.4.3 直接寻址.....	58

3.4.4	短直接寻址	59
3.4.5	特殊功能寄存器 (SFR) 寻址	60
3.4.6	寄存器间接寻址	61
3.4.7	基址寻址	62
3.4.8	基址变址寻址	63
3.4.9	堆栈寻址	64
第四章	端口功能	65
4.1	端口功能	65
4.2	端口配置	67
4.2.1	端口 0	68
4.2.2	端口 1	73
4.2.3	端口 2	74
4.2.4	端口 3 (仅限 38 引脚产品)	77
4.2.5	端口 12	78
4.3	控制端口功能的寄存器	81
4.4	端口功能操作	85
4.4.1	写入 I/O 端口	85
4.4.2	从 I/O 端口中读取	85
4.4.3	I/O 端口上的操作	85
4.5	使用复用功能时端口模式寄存器, 输出锁存, 上拉电阻选择寄存器以及端口输出模式寄存器的设置	86
4.6	端口寄存器 n (Pn) 的 1 位操作指令的注意事项	87
第五章	时钟发生器	88
5.1	时钟发生器的功能	88
5.2	时钟发生器的配置	89
5.3	控制时钟发生器的寄存器	91
5.4	系统时钟振荡器	99
5.4.1	X1 振荡器	99
5.4.2	内部高速振荡器	101
5.4.3	内部低速振荡器	101
5.4.4	预分频器	102
5.5	时钟发生器的操作	102
5.6	控制时钟	104
5.6.1	控制高速系统时钟的示例	104
5.6.2	控制内部高速振荡时钟的示例	106
5.6.3	控制内部低速振荡时钟的示例	108
5.6.4	提供给 CPU 以及周边硬件的时钟	109
5.6.5	CPU 时钟状态转换图	110
5.6.6	更改 CPU 时钟前的条件以及更改 CPU 时钟后的处理	113
5.6.7	CPU 时钟和主系统时钟间转换所需的时间	113
5.6.8	时钟振荡停止前的条件	114
5.6.9	周边硬件以及源时钟	115
第六章	16 位定时器 / 事件计数器 00	115
6.1	16 位定时器 / 事件计数器 00 的功能	115

6.2	16 位定时器 / 事件计数器 00 的配置.....	116
6.3	控制 16 位定时器 / 事件计数器 00 的寄存器.....	121
6.4	16 位定时器 / 事件计数器 00 的操作.....	129
6.4.1	间隔定时器的操作.....	129
6.4.2	方波输出操作.....	132
6.4.3	外部事件计数器操作.....	135
6.4.4	通过 TI000 引脚有效沿输入进入清除&开始模式的操作.....	138
6.4.5	自由运行定时器操作.....	151
6.4.6	PPG 输出操作.....	160
6.4.7	单脉冲输出操作.....	163
6.4.8	脉宽测量操作.....	168
6.5	TM00 的特殊用途.....	176
6.5.1	TM00 操作期间重写 CR010.....	176
6.5.2	设置 LVS00 和 LVR00.....	176
6.6	16 位定时器 / 事件计数器 00 的注意事项.....	178
第七章 8 位定时器 / 事件计数器 50 和 51		182
7.1	8 位定时器 / 事件计数器 50 和 51 的功能.....	182
7.2	8 位定时器 / 事件计数器 50 和 51 的配置.....	182
7.3	控制 8 位定时器 / 事件计数器 50 和 51 的寄存器.....	185
7.4	8 位定时器 / 事件计数器 50 和 51 的操作.....	190
7.4.1	用作间隔定时器.....	190
7.4.2	用作外部事件计数器.....	192
7.4.3	方波输出操作.....	193
7.4.4	PWM 输出操作.....	194
7.5	8 位定时器 / 事件计数器 50 和 51 的注意事项.....	198
第八章 8 位定时器 H0 和 H1.....		199
8.1	8 位定时器 H0 和 H1 的功能.....	199
8.2	8 位定时器 H0 和 H1 的配置.....	199
8.3	控制 8 位定时器 H0 和 H1 的寄存器.....	203
8.4	8 位定时器 H0 和 H1 的操作.....	208
8.4.1	用作间隔定时器 / 方波输出.....	208
8.4.2	用作 PWM 输出操作.....	211
8.4.3	载波发生器操作（仅用于 8 位定时器 H1）.....	217
第九章 看门狗定时器.....		224
9.1	看门狗定时器的功能.....	224
9.2	看门狗定时器的配置.....	225
9.3	控制看门狗定时器的寄存器.....	226
9.4	看门狗定时器的操作.....	227
9.4.1	看门狗定时器的控制操作.....	227
9.4.2	设置看门狗定时器的溢出时间.....	228
9.4.3	设置看门狗定时器窗口打开周期.....	229
第十章 串行接口 UART6.....		231
10.1	串行接口 UART6 的功能.....	231

10.2 串行接口 UART6 的配置	232
10.3 控制串行接口 UART6 的寄存器.....	235
10.4 串行接口 UART6 的操作	243
10.4.1 操作停止模式.....	243
10.4.2 异步串行接口 (UART) 模式.....	244
10.4.3 专用波特率发生器	255
10.4.4 波特率的计算.....	257
第十一章 中断功能.....	262
11.1 中断功能的类型	262
11.2 中断源及配置	262
11.3 控制中断功能的寄存器	266
11.4 中断服务操作.....	273
11.4.1 可屏蔽中断响应.....	273
11.4.2 软件中断请求响应	275
11.4.3 中断嵌套服务.....	276
11.4.4 中断请求保持.....	279
第十二章 按键中断功能.....	280
12.1 按键中断的功能	280
12.2 按键中断的配置	280
12.3 控制按键中断的寄存器	281
第十三章 待机功能.....	283
13.1 待机功能及配置	283
13.1.1 待机功能.....	283
13.1.2 控制待机功能的寄存器	283
13.2 待机功能的操作	286
13.2.1 HALT 模式.....	286
13.2.2 STOP 模式	288
第十四章 复位功能.....	294
14.1 确认复位源的寄存器	300
第十五章 上电清零电路.....	301
15.1 上电清零电路的功能	301
15.2 上电清零电路的配置	302
15.3 上电清零电路的操作	302
15.4 使用上电清零电路的注意事项.....	304
第十六章 低电压检测器.....	306
16.1 低电压检测器的功能	306
16.2 低电压检测器的配置	307
16.3 控制低电压检测器的寄存器	307
16.4 低电压检测器的操作	310
16.4.1 用作复位时	311

16.4.2 用作中断时	315
16.5 低电压检测器的注意事项	319
16.6 RAM 数据保存检测器	322
第十七章 选项字节	323
17.1 选项字节的功能	323
17.2 选项字节的格式	324
第十八章 FLASH 存储器	326
18.1 内部存储器容量切换寄存器	326
18.2 用 Flash 存储器编程器写入数据	327
18.3 编程环境	329
18.4 通信模式	329
18.5 在线引脚的处理	330
18.5.1 FLMD0 引脚	330
18.5.2 串行接口引脚	331
18.5.3 RESET 引脚.....	332
18.5.4 端口引脚	333
18.5.5 REGC 引脚	333
18.5.6 其他信号引脚	333
18.5.7 电源.....	333
18.6 编程方法	334
18.6.1 控制 Flash 存储器	334
18.6.2 Flash 存储器编程模式.....	334
18.6.3 选择通信模式	335
18.6.4 通信命令	335
18.7 安全设置	336
18.8 通过自编程对 Flash 存储器进行编程	339
第十九章 片上调试功能	342
19.1 连接 QB-MINI2	342
19.2 通过 QB-MINI2 使用的保留区域	344
第二十章 指令集	345
20.1 操作列表使用规则	345
20.1.1 操作数标识符及标识方法	345
20.1.2 操作栏的描述	346
20.1.3 标志操作栏的描述	346
20.2 操作列表	347
20.3 按寻址类型列出的指令	355
第二十一章 电气特性	358
第二十二章 封装图	369
附录 A 开发工具	371

A.1	软件包.....	374
A.2	语言处理软件	374
A.3	控制软件.....	375
A.4	Flash 存储器写入工具	376
	A.4.1 使用 flash 存储器编程器 PG-FP4, FL-PR4, PG-FP5 和 FL-PR5 时	376
	A.4.2 使用带有编程功能的片上调试仿真器 QB-MINI2.....	376
A.5	调试工具（硬件）	377
	A.5.1 使用在线仿真器 QB-179F124 时	377
	A.5.2 使用带有编程功能的片上调试仿真器 QB-MINI2 时	378
A.6	调试工具（软件）	378
附录 B	修订记录.....	379
	B.1 在此版本中的主要修订	379
	B.2 先前版本的修订记录.....	381

第一章 概述

1.1 特征

- 高速（1 μ S: $V_{DD} = 1.8$ 到 3.6 V, 高速系统时钟: @ $f_{XH} = 2$ 或 4 MHz 情况下工作, 0.5 μ S: $V_{DD} = 2.0$ 到 3.6 V, 高速系统时钟: @ $f_{XH} = 4$ MHz 操作）
- 通用寄存器: 8 位 \times 32 个寄存器（8 位 \times 8 个寄存器 \times 4 组）
- ROM, RAM 容量

产品代号 \ 项目	程序内存 (ROM)		数据存储器
			内部高速 RAM [‡]
μ PD179F110	Flash 存储器 [‡]	4 KB	512 字节
μ PD179F111		8 KB	
μ PD179F112, 179F122		16 KB	768 字节
μ PD179F113, 179F123		24 KB	1 KB
μ PD179F114, 179F124		32 KB	

注 通过使用内部存储器容量切换寄存器 (IMS), 可以改变内部 Flash 存储器以及内部高速 RAM 的容量。关于 IMS 的详细信息请参见 **18.1 存储器容量切换寄存器**。

- 片上单电源 Flash 存储器
- 自编程（带有引导交换功能）
- 片上调试功能
- 片上上电清零 (POC) 电路和低压检测器 (LVI)
- 片上看门狗定时器（在片上内部低速振荡时钟下操作）
- 片上按键中断功能
- I/O 端口:
 - 30 引脚产品 (μ PD179F11x 微控制器): 26 (N-ch 漏极开路 / CMOS: 24, P-ch 漏极开路^{‡1} / CMOS: 1)
 - 38 引脚产品 (μ PD179F12x 微控制器): 34 (N-ch 漏极开路: 32, P-ch 漏极开路^{‡1} / CMOS: 1)
- 定时器: 6 个通道
 - 16 位定时器 / 事件计数器: 1 个通道
 - 8 位定时器 / 事件计数器: 2 个通道
 - 8 位定时器: 2 个通道
 - 看门狗定时器: 1 个通道
- 串行接口: 1 个通道
 - UART: 1 个通道
- 电源供电电压: $V_{DD} = 1.8$ 到 3.6 V^{‡2}
- 工作环境温度: $T_A = -40$ 到 $+85^\circ\text{C}$

注

1. 可用于遥控输出
2. 由于上电清零 (POC) 电路检测电压 (V_{POC}) 是 $1.8\text{ V} \pm 0.1\text{ V}$, 所以应在 1.9 到 3.6 V 范围内使用本产品

1.2 应用

- 重新复位遥控输出

1.3 订购信息

<R> • Flash 存储器版本

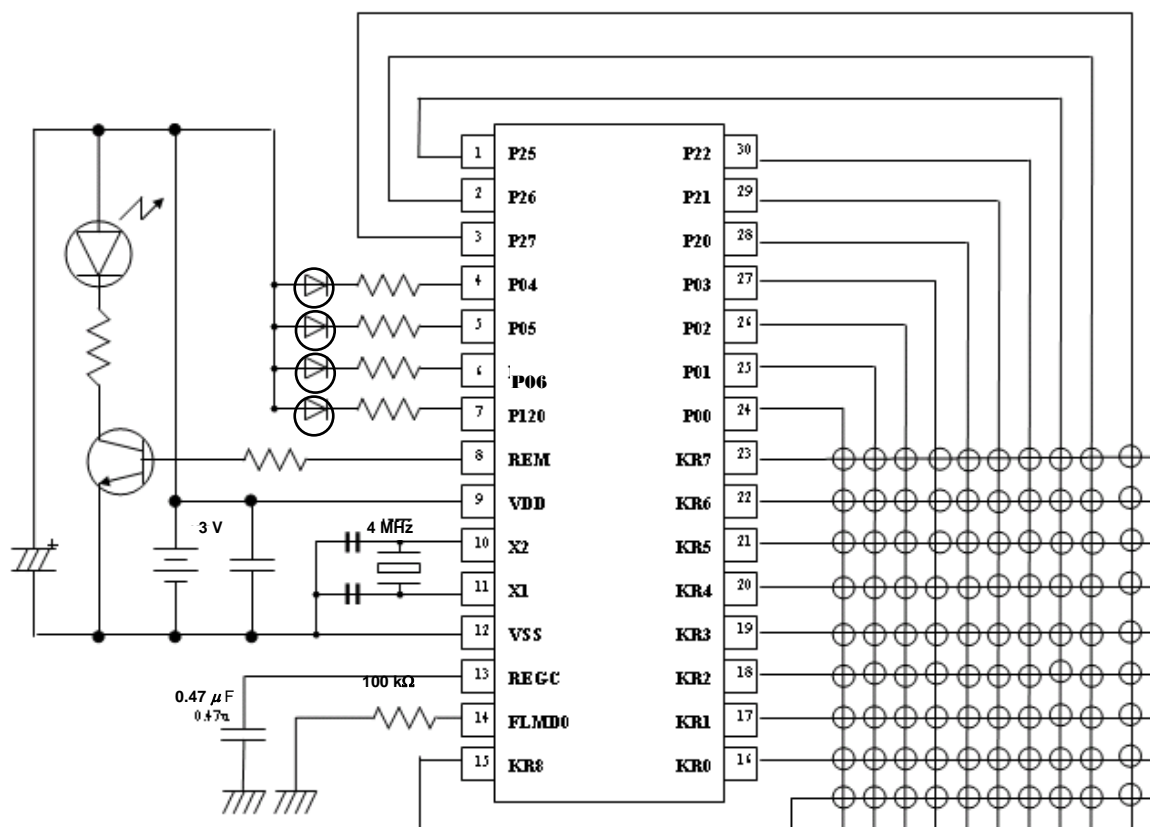
产品代号	封装	质量等级
μ PD179F110MC-CAB-AX ^注	30引脚塑封 SSOP (7.62 mm (300))	标准
μ PD179F111MC-CAB-AX ^注	30引脚塑封 SSOP (7.62 mm (300))	标准
μ PD179F112MC-CAB-AX ^注	30引脚塑封 SSOP (7.62 mm (300))	标准
μ PD179F113MC-CAB-AX ^注	30引脚塑封 SSOP (7.62 mm (300))	标准
μ PD179F114MC-CAB-AX ^注	30引脚塑封 SSOP (7.62 mm (300))	标准
μ PD179F122MC-GAA-AX ^注	38引脚塑封 SSOP (7.62 mm (300))	标准
μ PD179F123MC-GAA-AX ^注	38引脚塑封 SSOP (7.62 mm (300))	标准
μ PD179F124MC-GAA-AX ^注	38引脚塑封 SSOP (7.62 mm (300))	标准

注 开发中

备注 产品代号以-AX 结尾的产品为无铅产品。

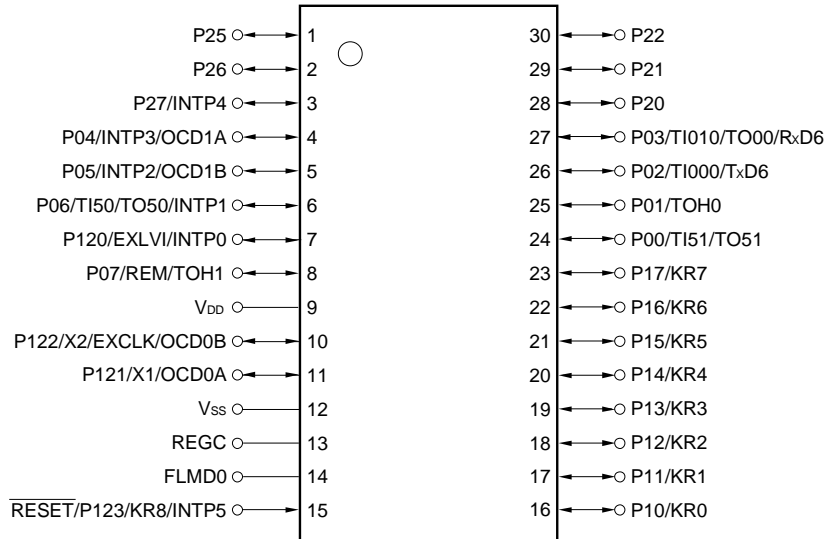
1.4 应用电路

下图展示了按键矩阵的例子：9 x 10 = 90 个键。通过使用 P04 到 P06 以及 P120，可以配置 126 个键的矩阵（在 μ PD179F11x 微控制器情况下）。



1.5 引脚配置（俯视图）

- 30 引脚塑封 SSOP（7.62 mm（300））

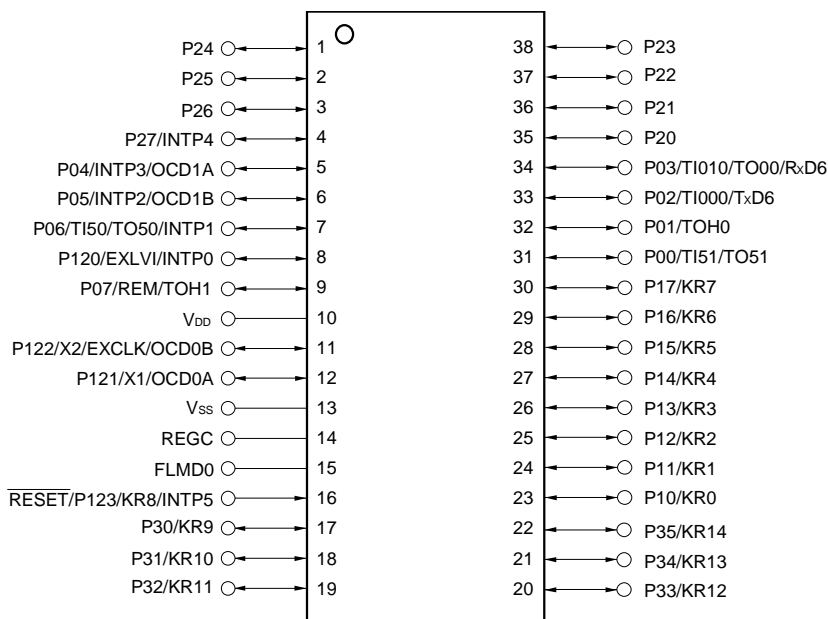


注意事项 通过一个电容将 **REGC** 引脚连接到 **V_{SS}**（0.47 μ F：推荐）。

引脚标识

EXCLK:	外部时钟输入 (主系统时钟)	TO00, TO50, TO51, TOH0, TOH1:	定时器输出
EXLVI:	用于低压检测器的外部电位输入	TxD6:	发送数据
FLMD0:	Flash 编程模式	V _{DD} :	电源
INTP0 到 INTP5:	外部中断输入	V _{SS} :	地
KR0 到 KR8:	按键中断返回	X1, X2:	晶体振荡器 (主系统时钟)
OCD0A, OCD0B, OCD1A, OCD1B:	片上调试输入 / 输出		
P00-P07:	端口 0		
P10-P17:	端口 1		
P20 到 P22, P25 到 P27:	端口 2		
P120 到 P123:	端口 12		
REGC:	调节器电容		
REM:	遥控输出		
RESET:	复位		
RxD6:	接收数据		
TI000, TI010, TI50, TI51:	定时器输入		

- 38 引脚塑封 SSOP (7.62 mm (300))



注意事项 通过一个电容将 **REGC** 引脚连接到 **V_{SS}** (**0.47μF**: 推荐)。

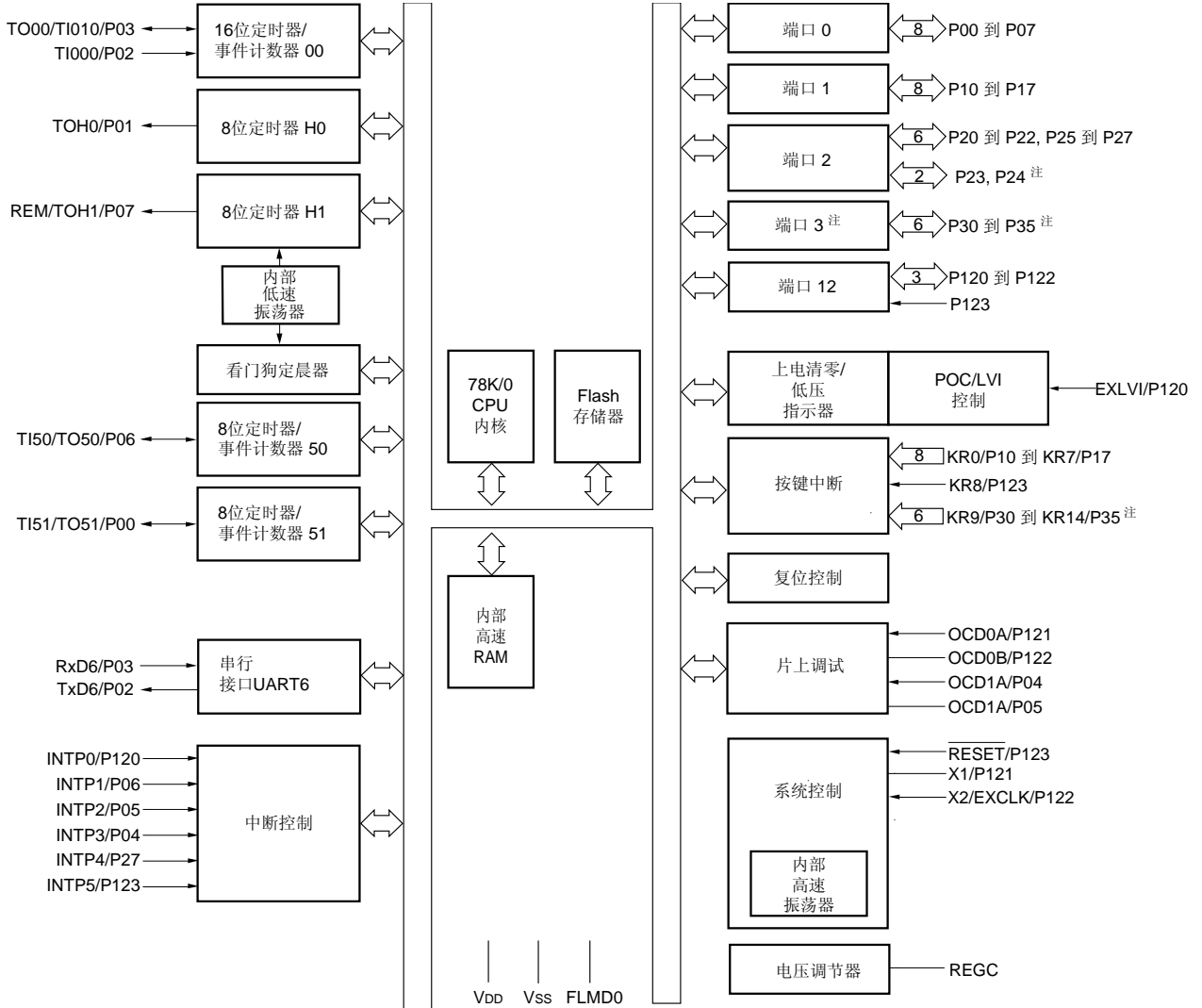
引脚标识

EXCLK:	外部时钟输入 (主系统时钟)	TO00, TO50, TO51, TOH0, TOH1:	定时器输出
EXLVI:	用于低压检测器的外部电位输入	TxD6:	发送数据
FLMD0:	Flash 编程模式	V _{DD} :	电源
INTP0 到 INTP5:	外部中断输入	V _{SS} :	地电平
KR0 到 KR14:	按键中断	X1, X2:	晶体振荡器 (主系统时钟)
OCD0A, OCD0B, OCD1A, OCD1B:	片上调试输入 / 输出		
P00 到 P07:	端口 0		
P10 到 P17:	端口 1		
P20 到 P27:	端口 2		
P120 到 P123:	端口 12		
REGC:	调节器电容		
REM:	遥控输出		
RESET:	复位		
RxD6:	接收数据		
TI000, TI010, TI50, TI51:	定时器输入		

1.6 μ PD179F11x, 179F12x微控制器系列

ROM	RAM	30 引脚	38 引脚
32 KB	1 KB	μ PD179F114	μ PD179F114
24 KB		μ PD179F113	μ PD179F113
16 KB	768 B	μ PD179F112	μ PD179F112
8 KB	512 B	μ PD179F111	-
4 KB		μ PD179F110	-

1.7 框图



注 仅适用于 38 引脚产品

1.8 功能概述

项目		μPD179F110	μPD179F111	μPD179F112	μPD179F113	μPD179F114	μPD179F122	μPD179F123	μPD179F124	
内部存储器 (字节)	Flash 存储器 (支持自编程) [‡]	4 K	8 K	16 K	24 K	32K	16 K	24 K	32K	
	高速 RAM [‡]	512		768	1 K		768	1 K		
存储器空间		64 KB								
主系统时钟 (振荡频率)	高速系统时钟	X1 (晶体 / 陶瓷) 振荡, 外部主系统时钟输入 (EXCLK) 4 MHz: V _{DD} = 1.8 到 3.6 V								
	内部高速振荡时钟	内部振荡 4 MHz ± 2%: V _{DD} = 1.8 到 3.6 V								
内部低速振荡时钟 (用于 WDT)		内部振荡 230 KHz (TYP.): V _{DD} = 1.8 到 3.6 V								
通用寄存器		8 位 × 32 个寄存器 (8 位 × 8 个寄存器 × 4 组)								
最短指令执行时间		<ul style="list-style-type: none"> • 1 μs (V_{DD} = 1.8 到 3.6 V, 高速系统时钟: @ f_{XH} = 2 or 4 MHz 下工作) • 0.5 μs (V_{DD} = 2.0 到 3.6 V, 高速系统时钟: @ f_{XH} = 4 MHz 下工作) 								
指令集		<ul style="list-style-type: none"> • 8 / 16 位操作 • 乘法 / 除法 (8 位 × 8 位, 16 位 ÷ 8 位) • 位操作 (设置, 复位, 测试和布尔操作) • BCD 调整, 等。 								
I/O 端口		总计: 26 (30 引脚产品) 34 (38 引脚产品) N-ch 漏极开路输出 / CMOS I/O: 24 32 P-ch 漏极开路输出 / CMOS I/O: 1 1 CMOS 输入: 1 1								
定时器		<ul style="list-style-type: none"> • 16 位定时器 / 事件计数器: 1 个通道 • 8 位定时器 / 事件计数器: 2 个通道 • 8 位定时器: 2 个通道 • 看门狗定时器: 1 个通道 								
	定时器输出	5 (PWM 输出: 2, 用于遥控的载波发生器输出: 1)								
串行接口		UART 模式: 1 个通道								
中断向量源	内部	10								
	外部	8								
按键中断		按键中断发生在检测到按键输入引脚下降沿时 (KR0 和 KR8)。					按键中断发生在检测到按键输入引脚下降沿时 (KR0 到 KR14)。			
复位		<ul style="list-style-type: none"> • 使用 RESET 引脚复位 • 使用看门狗定时器内部复位 • 使用上电清零内部复位 • 使用低压检测器内部复位 								
片上调试功能		提供								
供电电压		V _{DD} = 1.8 到 3.6 V								
工作环境温度		T _A = -40 到 +85°C								
封装		30 引脚塑封 SSOP (7.62 mm (300))					38 引脚塑封 SSOP (7.62 mm (300))			

注 通过使用内部存储器容量切换寄存器 (IMS), 可以改变内部 Flash 存储器以及内部高速 RAM 的容量。

定时器概况如下所示。

		16 位定时器 / 事件计数器 00	8 位定时器 / 事件计数器 50 and 51		8 位定时器 H0 以及 H1		看门狗定时 器
		TM00	TM50	TM51	TMH0	TMH1	
功能	间隔定时器	1 个通道	1 个通道	1 个通道	1 个通道	1 个通道	-
	外部事件计数器	1 个通道	1 个通道	1 个通道	-	-	-
	PWM 输出	-	-	-	1 输出	1 输出	-
	脉宽测量	2 输入	-	-	-	-	-
	方波输出	1 输出	1 输出	1 输出	-	-	-
	载波发生器	-	-	注	-	1 输出 ^注	-
	看门狗定时器	-	-	-	-	-	1 个通道
中断源		2	1	1	1	1	-

注 TM51 和 TMH1 可以作为载波发生器模式来结合使用。

第二章 引脚功能

2.1 引脚功能列表

(1) 端口功能

功能名称	I/O	功能	复位后	复用功能
P00	I/O	端口 0。 8 位 I/O 端口。 可以以 1 位为单位来指定输入 / 输出模式。 可以以 1 位为单位定义片上上拉电阻的使用。 可以以 1 位为单位将 P00 到 P06 设为 N-ch 漏极开路输出或 CMOS I/O。 可以以 1 位为单位将 P07 设为 P-ch 漏极开路输出或 CMOS I/O。通过指定 P-ch 漏极开路输出，该引脚可用作作用于遥控的载波发生器输出。	输入端口	TI51/TO51
P01				TOH0
P02				TI000/TxD6
P03				TI010/TO00/ RxD6
P04				INTP3/OCD1A
P05				INTP2/OCD1B
P06				TI50/TO50/INTP1
P07				REM/TOH1
P10 到 P17	I/O	端口 1。 8 位 I/O 端口。 可以以 1 位为单位来指定输入 / 输出模式。 可以以 1 位为单位定义片上上拉电阻的使用。 可以以 1 位为单位将 P10 到 P17 设为 N-ch 漏极开路输出或 CMOS I/O。	输入端口	KR0 到 KR7
P20 到 P22	I/O	端口 2。 38 引脚产品：8 位 I/O 端口 30 引脚产品：6 位 I/O 端口 可以以 1 位为单位来指定输入 / 输出模式。 可以以 1 位为单位定义片上上拉电阻的使用。 可以以 1 位为单位将 P20 到 P22，P23 ^注 ，P24 ^注 ，P25 到 P27 设为 N-ch 漏极开路输出或 CMOS I/O。	输入端口	-
P23 ^注 ，P24 ^注				-
P25，P26				-
P27				INTP4
P30 到 P35 ^注	I/O	端口 3 ^注 。 6 位 I/O 端口。 可以以 1 位为单位来指定输入 / 输出模式。 可以以 1 位为单位定义片上上拉电阻的使用。 可以以 1 位为单位将 P30 到 P35 设为 N-ch 漏极开路输出或 CMOS I/O。	输入端口	KR9 到 KR14 ^注
P120	I/O	端口 12。 3 位 I/O 端口和 1 位只输入端口。 在 P120 到 P122 引脚中，可以以 1 位为单位来指定输入 / 输出模式。	输入端口	EXLVI/INTP0
P121				X1/OCD0A
P122				X2/EXCLK/OCD0B
P123				输入 仅在 P120 和 P123 引脚中，可以指定片上上拉电阻的使用。 可以以 1 位为单位将 P120 到 P122 设为 N-ch 漏极开路输出或 CMOS I/O。

注 仅限 38 引脚的产品

(2) 非端口功能 (1/2)

功能名称	I/O	功能	复位后	复用功能
EXLVI	输入	外部低压检测的电位输入	输入端口	P120/INTP0
FLMD0	-	Flash 存储器编程模式设置	-	-
INTP0	输入	可以指定有效沿（上升沿，下降沿，或双边沿）的外部中断请求输入	输入端口	P120/EXLVI
INTP1				P06/TI50/TO50
INTP2				P05/OCD1B
INTP3				P04/OCD1A
INTP4				P27
INTP5				P123/RESET/ KR8
KR0 到 KR7	输入	按键中断输入	输入端口	P10-P17
KR8				P123/RESET/ INTP5
KR9 到 KR15 [‡]				P30 到 P35 [‡]
REGC	-	为内部操作连接调节器输出稳定电容。 通过一个电容器（0.47 μ F：推荐）连接至 V _{ss} 。	-	-
REM	输出	遥控输出	输入端口	P07/TOH1
RESET	输入	系统复位输入	输入端口	P123/KR8/INTP5
RxD6	输入	输入至异步串行接口的串行数据	输入端口	P03/TI010/TO00
TI000	输入	外部计数时钟输入到 16 位定时器 / 事件计数器 00 捕获触发信号输入到 16 位定时器 / 事件计数器 00 的捕获寄存器（CR000, CR010）中	输入端口	P02/TxD6
TI010		捕获触发信号输入到 16-位定时器 / 事件计数器 00 的捕获寄存器（CR000）中		P03/TO00/RxD6
TI50	输入	外部计数时钟输入到 8 位定时器 / 事件计数器 50 中	输入端口	P06/TO50/INTP1
TI51		外部计数时钟输入到 8 位定时器 / 事件计数器 51 中		P00/TO51
TO00	输出	16 位定时器 / 事件计数器 00 输出	输入端口	P03/TI010/RxD6
TO50	输出	8 位定时器 / 事件计数器 50 输出	输入端口	P06/TI50/INTP1
TO51		8 位定时器 / 事件计数器 51 输出		P00/TI51
TOH0	输出	8 位定时器 H0 输出	输入端口	P01
TOH1		8 位定时器 H1 输出		P07/REM
TxD6	输出	从异步串行接口输出的串行数据	输入端口	P02/TI000
X1	输入	为主系统时钟连接谐振器	输入端口	P121/OCD0A
X2	-			P122/EXCLK/ OCD0B
EXCLK	输入	为主系统时钟输入的外部时钟	输入端口	P122/X2/OCD0B
V _{DD}	-	电源正极	-	-
V _{SS}	-	地电平	-	-
OCD0A	输入	片上调试模式设置引脚的连接	输入端口	P121/X1
OCD1A				P04/INTP3
OCD0B	-			P122/X2/EXCLK
OCD1B				P05/INTP2

注 仅限 38 引脚的产品

2.2 引脚功能描述

2.2.1 P00 到 P07（端口 0）

P00 到 P07 作为 8 位 I/O 端口使用。这些引脚也可用于定时器 I/O，串行数据 I/O，外部中断请求输入，遥控输出以及设置片上调试模式的连接。

以下操作模式可以以 1 位为单位进行指定。

(1) 端口模式

P00 到 P07 作为 8 位 I/O 端口使用。P00 到 P07 可以通过使用端口模式寄存器 0（PM0）以 1 位为单位设置为输入或输出端口。可以通过上拉电阻选择寄存器 0（PU0）来指定片上上拉电阻的使用。

通过使用端口输出模式寄存器 0（POM0），可以以 1 位为单位设置 P00 到 P06 为 N-ch 漏极开路输出或 CMOS I/O。

通过使用端口输出模式寄存器 0（POM0），可以以 1 位为单位设置 P07 为 P-ch 漏极开路输出或 CMOS I/O。通过指定 P-ch 漏极开路输出，该引脚可用作用于遥控的载波发生器输出。

(2) 控制模式

P00 到 P07 可用于定时器 I/O，串行数据 I/O，外部中断请求输入，遥控输出以及设置片上调试模式的连接。

(a) TI000

这是用于将外部计数时钟输入到 16 位定时器 / 事件计数器 00 中的引脚，也用于将捕获触发信号输入到 16 位定时器 / 事件计数器 00 的捕获寄存器（CR000，CR010）中。

(b) TI010

这用于将捕获触发信号输入到 16 位定时器 / 事件计数器 00 的捕获寄存器（CR000）中的引脚。

(c) T000

这是 16 位定时器 / 事件计数器 00 的定时器输出引脚。

(d) RxD6

这是串行接口 UART6 的串行数据输入引脚。

(e) TxD6

这是串行接口 UART6 的串行数据输出引脚。

(f) TI50, TI51

这些是用于将外部计数时钟输入到 8 位定时器 / 事件计数器 50 和 51 的引脚。

(g) T050, T051

这些是 8 位定时器 / 事件计数器 50 和 51 的定时器输出引脚。

(h) TOH0, TOH1

这些是 8 位定时器 H0 和 H1 的定时器输出引脚。

(i) INTP1 到 INTP3

这些是可以为有效沿（上升沿，下降沿，或双边沿）指定的外部中断请求输入引脚。

(j) REM

这是用于输出遥控信号的引脚。

(k) OCD1A, OCD1B

这些为片上调试模式的设置连接的引脚。

2.2.2 P10 到 P17（端口 1）

P10 和 P17 用作 8 位 I/O 端口。这些引脚也可用于按键中断输入。

以下操作模式可以以 1 位为单位进行指定。

(1) 端口模式

P10 和 P17 用作 8 位 I/O 端口。P10 和 P17 可以通过使用端口模式寄存器 1（PM1）以 1 位为单位设置为输入或输出端口。可以通过上拉电阻选择寄存器 1（PU1）来指定片上上拉电阻的使用。通过使用端口输出模式寄存器 1（POM1），可以以 1 位为单位设置 P10 到 P17 为 N-ch 漏极开路或 CMOS I/O。

(2) 控制模式**(a) KR0 到 KR7**

P10 到 P17 可用于按键中断输入。

2.2.3 P20 到 P27（端口 2）

在 38 引脚产品中，P20 到 P27 可用作 8 位 I/O 端口。在 30 引脚产品中，P20 到 P22，P25 到 P27 可用作 6 位 I/O 端口。这些引脚也可以用于外部中断请求输入。

以下操作模式可以以 1 位为单位进行指定。

(1) 端口模式

在 38 引脚产品中，P20 到 P27 可用作 8 位 I/O 端口。在 30 引脚产品中，P20 到 P22，P25 到 P27 可用作 6 位 I/O 端口。通过使用端口模式寄存器 2（PM2），这些引脚可以以 1 位为单位设置为输入模式或输出模式。由上拉电阻选择寄存器 2（PU2）来指定片上上拉电阻的使用。通过使用端口输出模式寄存器 2（POM2），这些引脚可以以 1 位为单位设置为 N-ch 漏极开路输出或 CMOS I/O。

(2) 控制模式（仅 P27）**(a) INTP4**

P27 可用作外部中断请求输入引脚。

注意事项 对于 30 引脚产品，确保设置 PM2 的第 3 位和第 4 位为“1”，以及 P2，PU2，和 POM2 第 3 位和第 4 位为“0”。

备注 P23 和 P24 仅存在于 38 引脚产品中。

2.2.4 P30 到 P35（端口 3）（仅适用于 38 引脚产品）

P30 到 P35 可用作 6 位 I/O 端口。这些引脚也用作外部中断请求输入以及定时器 I/O 的引脚。

以下操作模式可以以 1 位为单位进行指定。

(1) 端口模式

P30 到 P35 可用作 6 位 I/O 端口。通过使用端口模式寄存器 3（PM3），可以以 1 位为单位设置 P30 到 P35 为输入输出端口。可以通过上拉电阻选择寄存器 3（PU3）来指定片上上拉电阻的使用。通过使用端口输出模式寄存器 3（POM3），可以以 1 位为单位设置 P30 到 P35 为 N-ch 漏极开路输出或 CMOS I/O。

(2) 控制模式

(a) KR9 到 KR14

P30 到 P35 可用作按键中断输入。

注意事项 对于 30 引脚产品，确保设置 PM3 的第 0 到 5 位为“1”，以及 P3，PU3，和 POM3 第 0 到 5 位为“0”。

2.2.5 P120 到 P123（端口 12）

P120 至 P122 用作 3 位 I/O 端口。P123 只用作输入端口。这些引脚也可用于外部中断请求输入，外部低压检测电压输入，为主系统时钟连接振荡器，主系统时钟的外部时钟输入，系统复位输入，按键中断输入以及片上调试模式的连接设置。

为 P123，INTP5 或 KR8 使用 P123/RESET/INTP5/KR8 引脚时，应将复位引脚模式寄存器（RSTMASK）的第 3 位（RSTM）设置为“1”。

以下操作模式可以以 1 位为单位进行指定。

(1) 端口模式

P120 至 P122 用作 3 位 I/O 端口。P123 只用作输入端口。通过使用端口模式寄存器 12（PM12），可以以 1 位为单位设置 P120 到 P122 为输入或输出端口。仅对于 P120 和 P123，可以通过上拉电阻选择寄存器 12（PU12）指定片上上拉电阻的使用。通过使用端口输出模式寄存器 12（POM12），可以以 1 位为单位设置 P120 到 P122 为 N-ch 漏极开路输出或 CMOS I/O。

(2) 控制模式

P120 到 P123 可用作外部中断请求输入，外部低压检测的电压输入，为主系统时钟连接振荡器，主系统时钟的外部时钟输入，系统复位输入，按键中断输入以及为片上调试模式设置连接的引脚。

(a) INTP0, INTP5

这些是可以为有效沿（上升沿，下降沿，或双边沿）指定的外部中断请求输入引脚。

(b) EXLVI

这是外部低压检测的电平输入引脚。

(c) X1, X2

这些是用于连接主系统时钟谐振器的引脚。

(d) EXCLK

这是主系统时钟的外部时钟输入引脚。

(e) KR8

这是按键中断输入引脚。

(f) $\overline{\text{RESET}}$

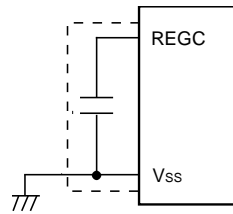
该引脚为有效低电平系统复位输入引脚。

(g) OCD0A, OCD0B

这些为片上调试模式的设置连接的引脚。

2.2.6 REGC

该引脚用于为内部操作连接调节器输出（2.0 V）稳定电容。通过一个电容器（0.47 μF ；推荐）连接至该引脚。



注意事项 在上图中，虚线部分的配线长度应尽可能的短。

2.2.7 V_{DD}

V_{DD} 为正电源引脚。

2.2.8 V_{SS}

V_{SS} 是地电平引脚。

2.2.9 FLMD0

这是用于设置 flash 存储器编程模式的引脚。在线编程或自编程时，将引脚 FLMD0 通过电阻（100 k Ω ）下拉。

在正常工作模式中，保持 FLMD0 与 V_{SS} 具有相同的电位。

在 flash 存储器编程模式中，将该引脚连接到 flash 存储器编程器上。

2.3 I/O 引脚电路以及不使用引脚的推荐连接

表 2-2 显示了 I/O 引脚电路的类型以及不使用的引脚的推荐连接。

各个类型的 I/O 电路和配置请参见图 2-1。

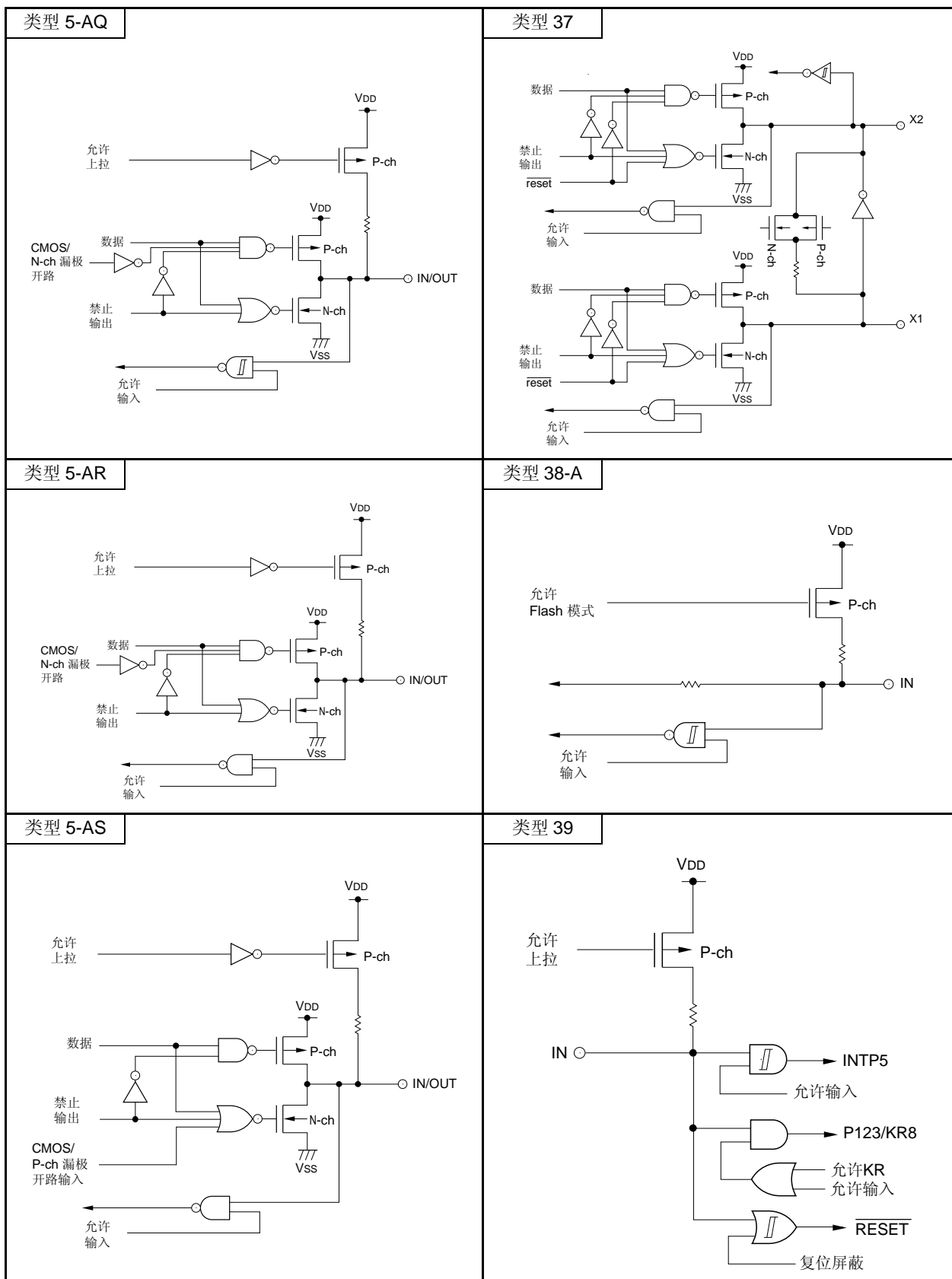
表 2-1. I/O 引脚电路类型

引脚名称	I/O 电路类型	I/O	不使用的引脚的推荐连接
P00/TI51/TO51	5-AQ	I/O	输入： 通过一个电阻单独连接到 V _{DD} 或 V _{SS} 。 输出： 保持开路。
P01/TOH0	5-AR		
P02/TI000/TxD6	5-AQ		
P03/TI010/TO00/RxD6			
P04/INTP3/OCD1A			
P05/INTP2/OCD1B			
P06/TI50/TO50/INTP1			
P07/REM/TOH1			
P10/KR0	5-AR		
P11/KR1			
P12/KR2			
P13/KR3			
P14/KR4			
P15/KR5			
P16/KR6			
P17/KR7			
P20			
P21			
P22			
P23 ^{#1}	5-AQ		
P24 ^{#1}			
P25			
P26			
P27/INTP4	5-AR		
P30/KR9 ^{#1}			
P31/KR10 ^{#1}			
P32/KR11 ^{#1}			
P33/KR12 ^{#1}			
P34/KR13 ^{#1}			
P35/KR14 ^{#1}	37		
P120/EXLVI/INTP0		5-AQ	
P121/X1/OCD0A	37	输入： 通过一个电阻单独连接到 V _{DD} 或 V _{SS} 。 输出： 保持开路。	
P122/X2/EXCLK/ OCD0B			
P123/RESET/KR8/ INTP5	39	输入	通过一个电阻直接连接到 V _{DD} 上。
FLMD0	38-A	-	连接到 V _{SS} ^{#2} 。

注 1. P23, P24, P30 到 P35 仅存在于 38 引脚产品中。

2. FLMD0 是用于将数据写入 flash 存储器中的引脚。要在线重写 flash 存储器的数据，则需通过一个电阻（100kΩ：推荐）将此引脚连接到 V_{SS} 上。

图 2-1. I/O 引脚电路列表 (1/2)



第三章 CPU架构

3.1 存储空间

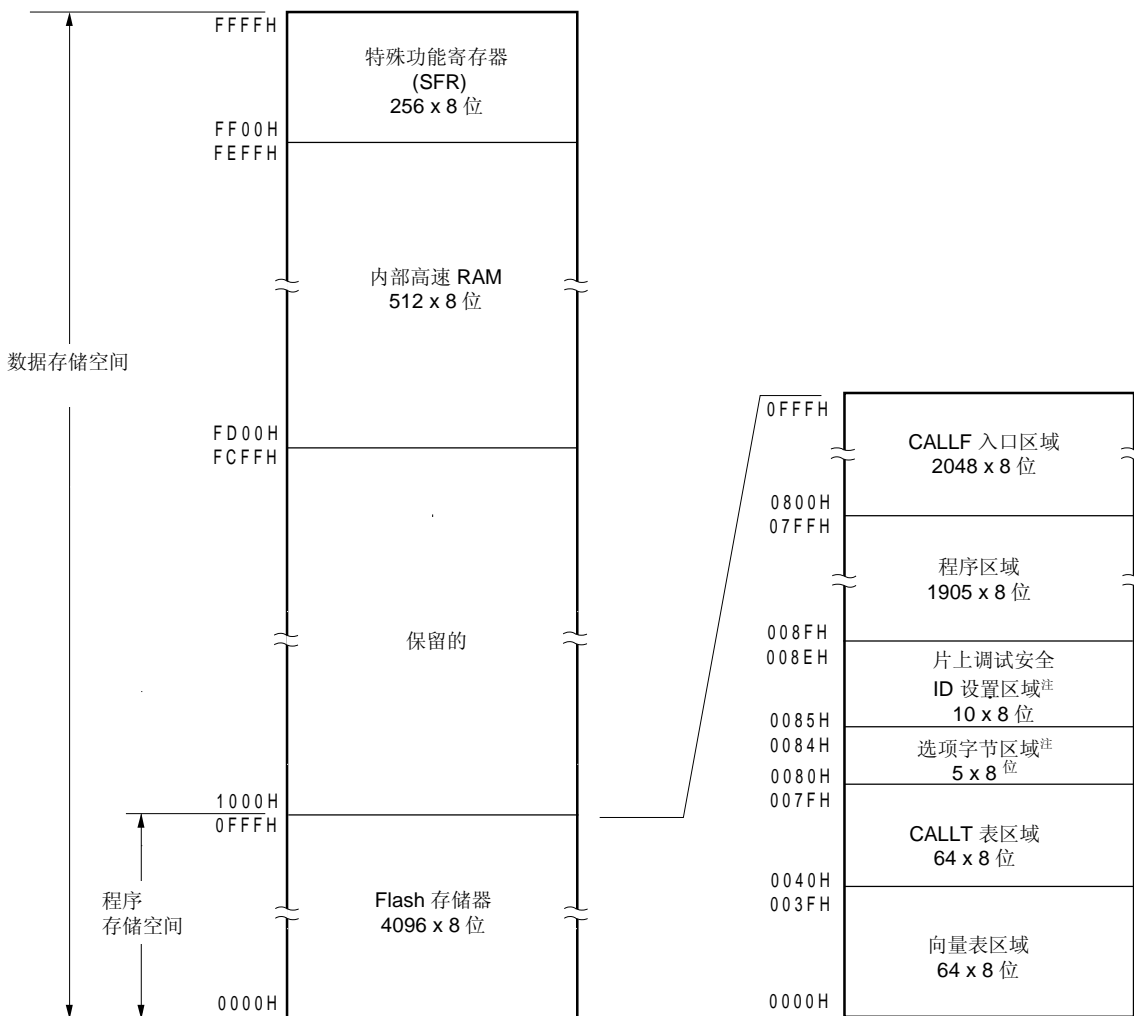
μ PD179F11x, 179F12x 微控制器中的产品可以访问一个 64KB 大小的存储器空间。图 3-1 到 3-5 显示了存储器映射图。

注意事项 不管内部存储器容量有多少, μ PD179F11x, 179F12x 微控制器中所有产品的内部存储器容量切换寄存器 (IMS) 的初始值都是固定的 (IMS = CFH)。因此, 应按以下所显示的内容对各个产品进行相应的值设置。

表 3-1. 内部存储器容量切换寄存器 (IMS) 的设定值

Flash 存储器版本 (μ PD179F11x, 179F12x 微控制器)	IMS	ROM 容量	内部高速 RAM 容量
μ PD179F110	41H	4 KB	512 字节
μ PD179F111	42H	8 KB	
μ PD179F112, 179F122	04H	16 KB	768 字节
μ PD179F113, 179F123	C6H	24 KB	1 KB
μ PD179F114, 179F124	C8H	32 KB	

图 3-1. 存储器映射图 (μPD179F110)



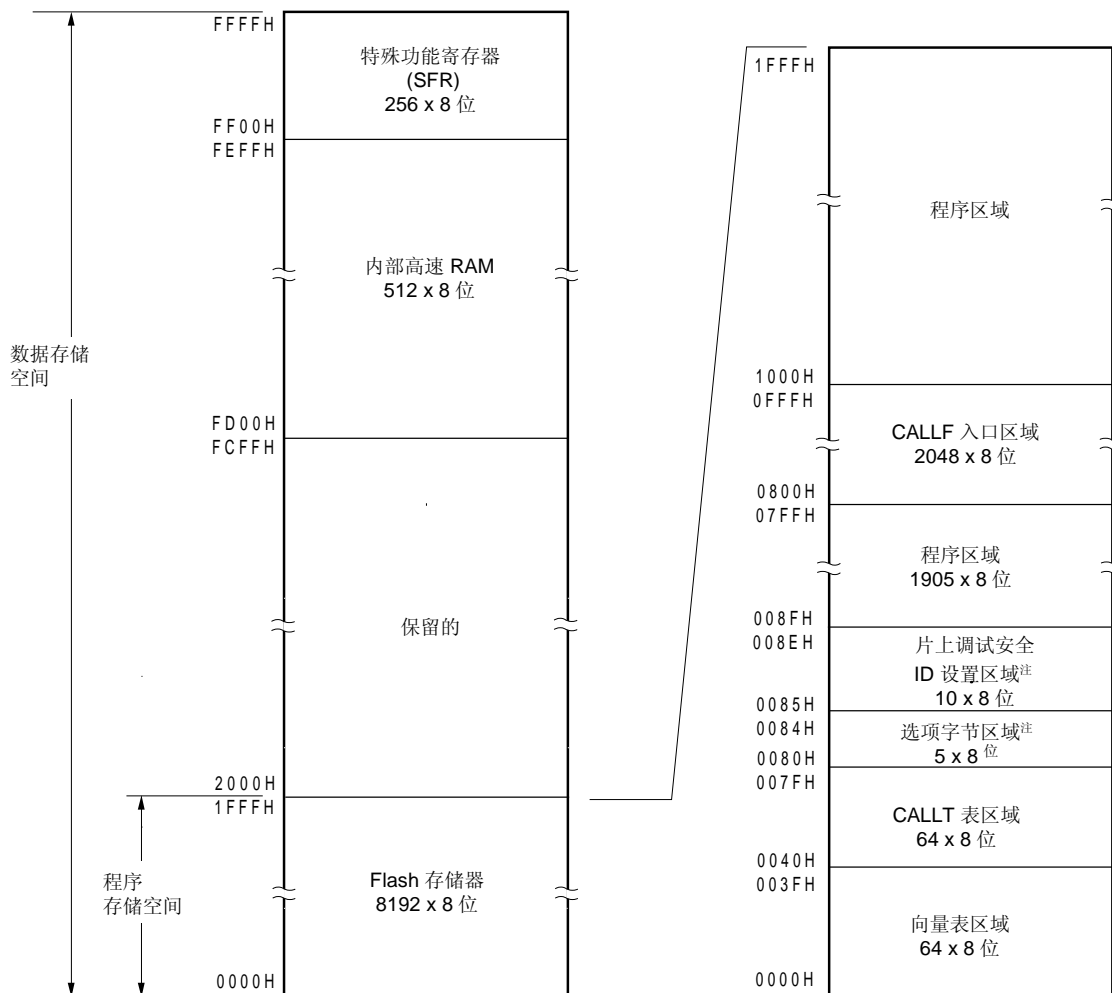
注 将选项字节设为 0080H 至 0084H。
将片上调试安全 ID 设置为 0085H 到 008EH。

备注 flash 存储器被分为多个 block (每个 block = 1 KB)。关于地址值以及 block 编号的详细信息, 请参见表 3-2 Flash 存储器中地址值与 block 编号间的对应关系

0FFFH	Block 03H
0C00H 0BFFH	Block 02H
07FFH	Block 01H
0400H 03FFH	Block 00H
0000H	

↑ 1 KB

图 3-2. 存储器映射图 (μ PD179F111)



注 将选项字节设为 0080H 至 0084H。
将片上调试安全 ID 设置为 0085H 到 008EH。

备注 flash 存储器被分为多个 block (每个 block = 1 KB)。关于地址值以及 block 编号的详细信息, 请参见表 3-2 Flash 存储器中地址值与 block 编号间的对应关系

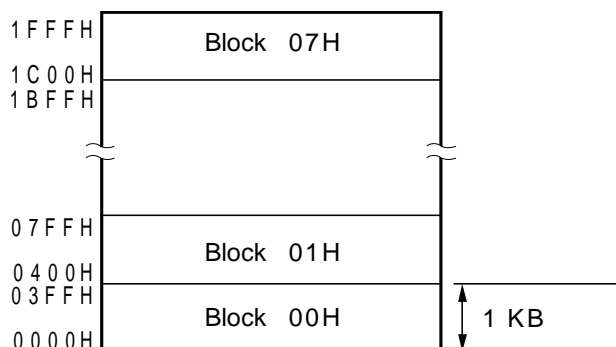
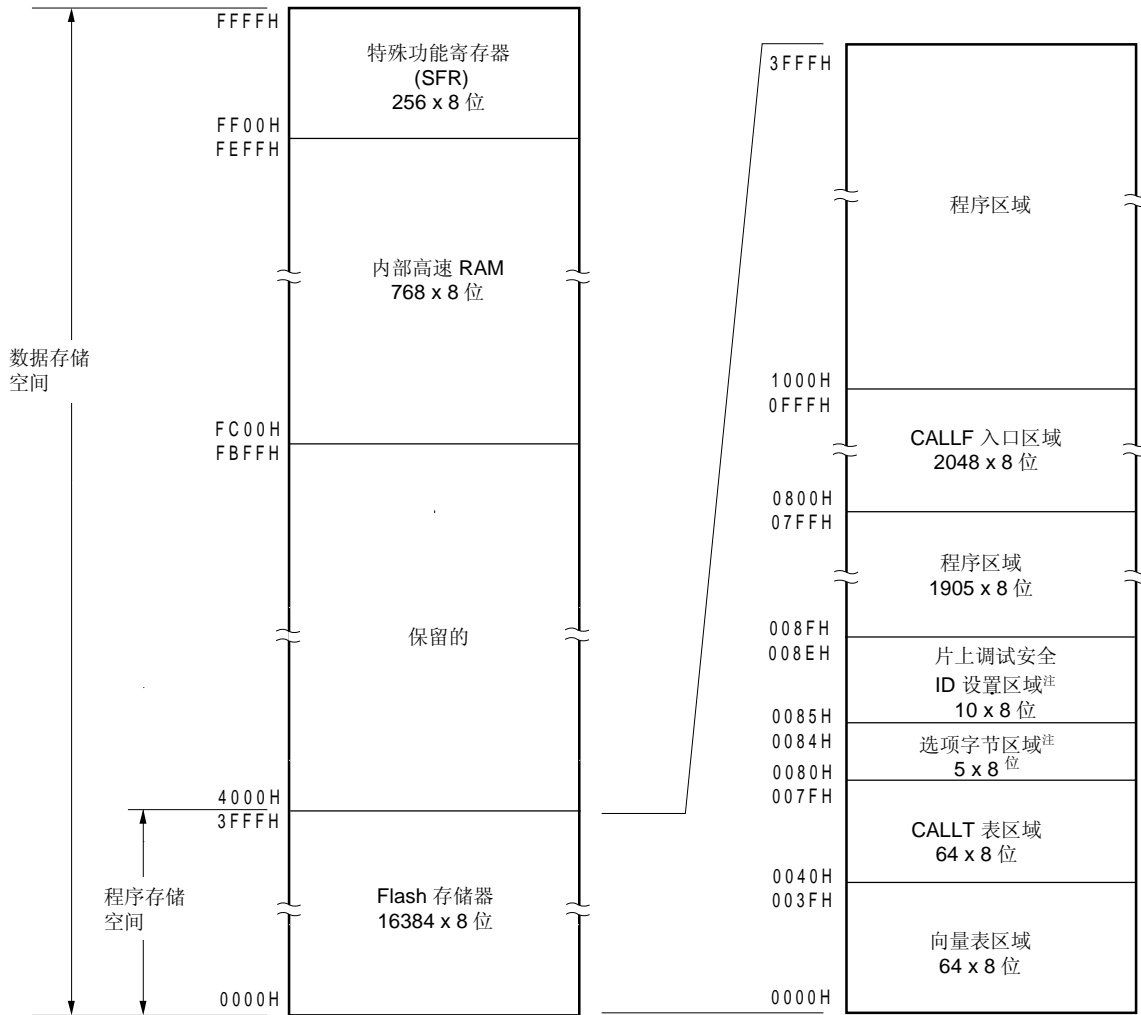


图 3-3. 存储器映射图 (μPD179F112, 179F122)



注 将选项字节设为 0080H 至 0084H。
将片上调试安全 ID 设置为 0085H 到 008EH。

备注 flash 存储器被分为多个 block (每个 block = 1 KB)。关于地址值以及 block 编号的详细信息, 请参见表 3-2 Flash 存储器中地址值与 block 编号间的对应关系

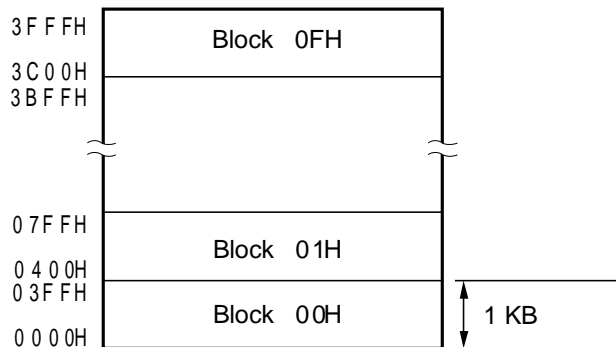
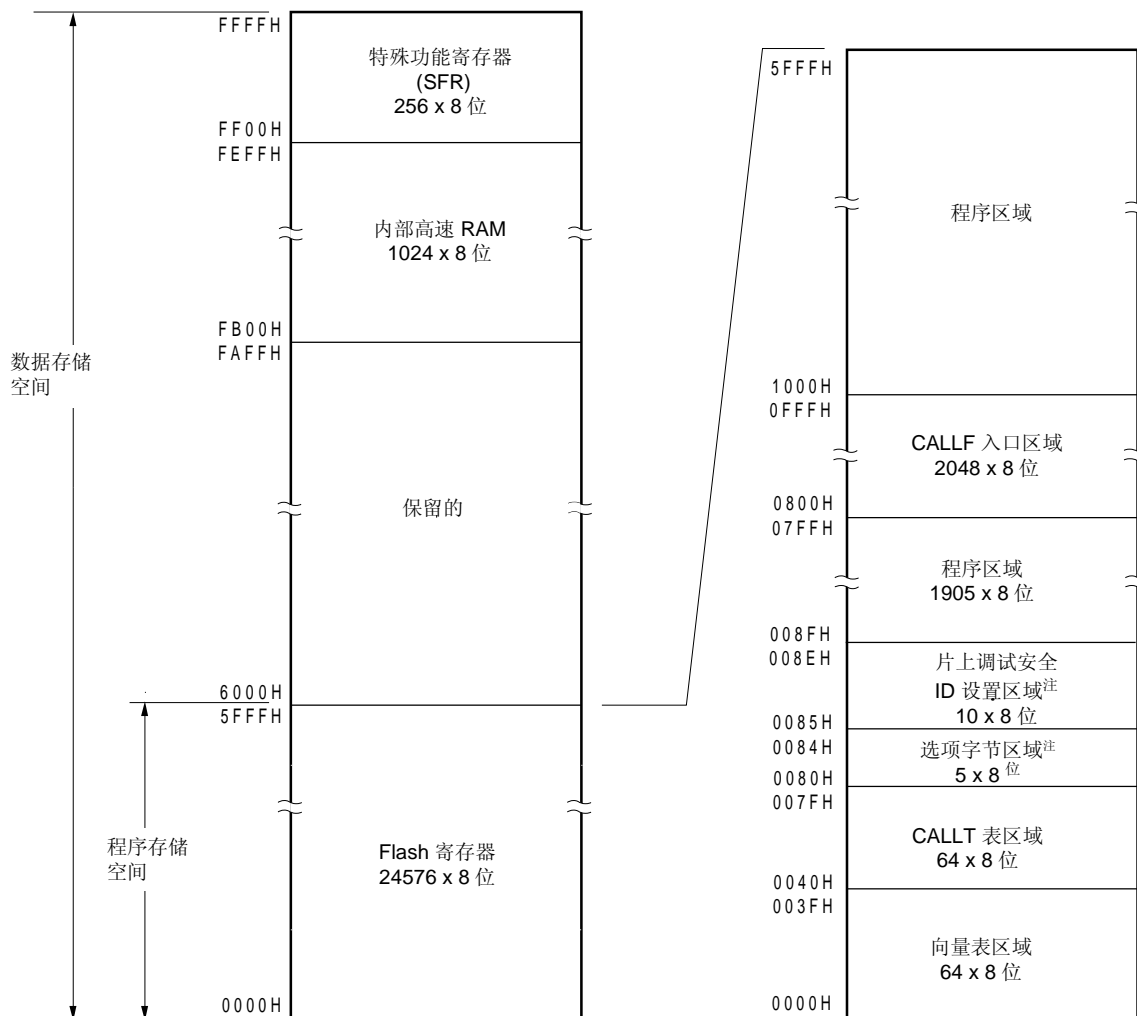


图 3-4. 存储器映射图 (μPD179F113, 179F123)



注 将选项字节设为 0080H 至 0084H。
将片上调试安全 ID 设置为 0085H 到 008EH。

备注 flash 存储器被分为多个 block (每个 block = 1 KB)。关于地址值以及 block 编号的详细信息, 请参见表 3-2 Flash 存储器中地址值与 block 编号间的对应关系

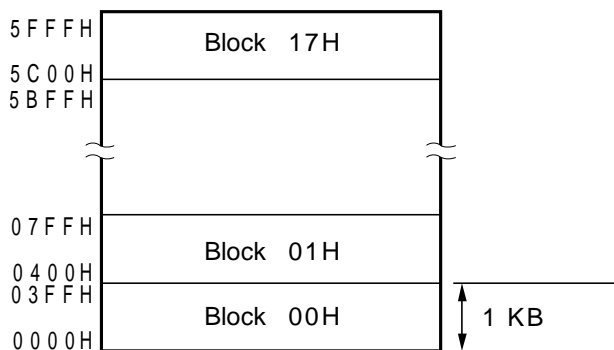
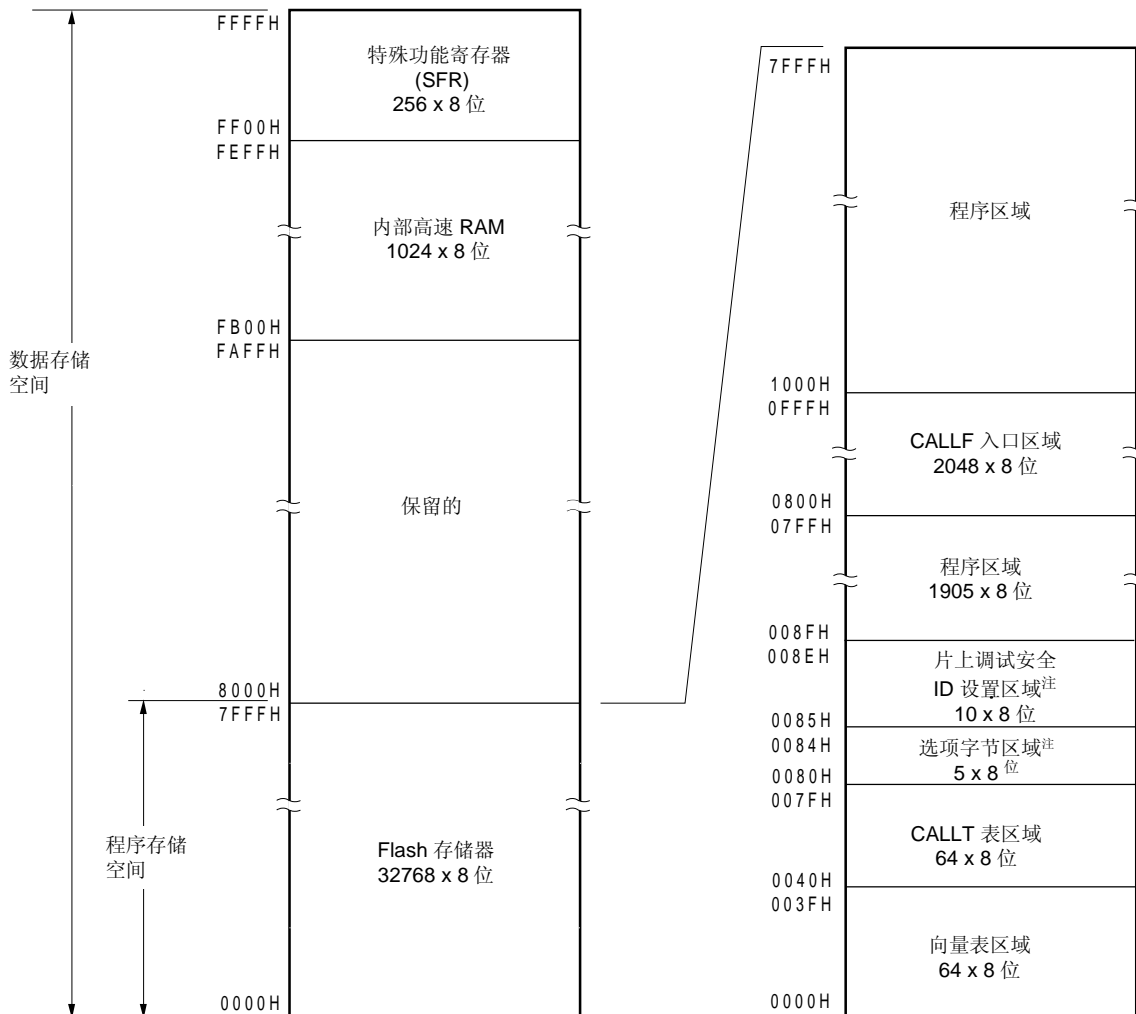
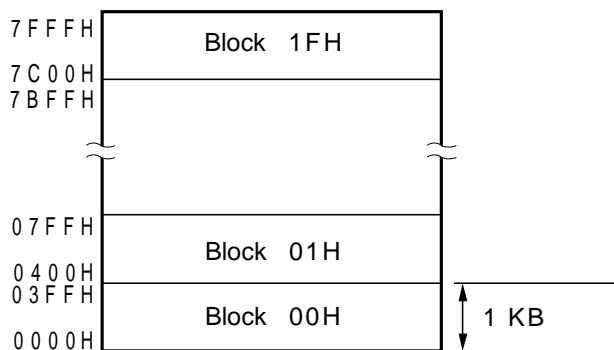


图 3-5. 存储器映射图 (μPD179F114, 179F124)



注 将选项字节设为 0080H 至 0084H。
将片上调试安全 ID 设置为 0085H 到 008EH。

备注 flash 存储器被分为多个 block (每个 block = 1 KB)。关于地址值以及 block 编号的详细信息, 请参见表 3-2 Flash 存储器中地址值与 block 编号间的对应关系



flash 存储器中地址值与 block 编号的对应关系如下所示。

表 3-2. Flash 存储器中地址值与 block 编号的对应关系

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

备注 μPD179F110: block 编号 00H 到 03H
 μPD179F111: block 编号 00H 到 07H
 μPD179F112, 179F122: block 编号 00H 到 0FH
 μPD179F113, 179F123: block 编号 00H 到 17H
 μPD179F114, 179F124: block 编号 00H 到 1FH

3.1.1 内部程序存储空间

内部程序存储空间用于保存程序以及表格数据。通常情况下，它使用程序计数器（PC）进行编址。如下所示， μ PD179F11x, 179F12x 微控制器中包含有内置 ROM（flash 存储器）。

表 3-3. 内置 ROM 容量

产品代号	内部 ROM	
	结构	容量
μ PD179F110	Flash 存储器	4096 × 8 位 (0000H 到 0FFFH)
μ PD179F111		8192 × 8 位 (0000H 到 1FFFH)
μ PD179F112, 179F122		16384 × 8 位 (0000H 到 3FFFH)
μ PD179F113, 179F123		24576 × 8 位 (0000H 到 5FFFH)
μ PD179F114, 179F124		32768 × 8 位 (0000H 到 7FFFH)

内部程序存储空间被分为以下各区域。

(1) 向量表区域

64 字节的区域 0000H 至 003FH 被保留为向量表区域。在发生复位或生成各个中断请求基础上的跳转指令的程序开始地址被保存在向量表区域中。

16 位地址中的低 8 位被保存在偶地址中，而 16 位地址中的高 8 位则被保存在奇地址中。

表 3-4. 向量表

向量表地址	中断源	向量表地址	中断源
0000H	RESET 输入, POC, LVI, WDT	0016H	INTST6
0004H	INTLVI	001AH	INTTMH1
0006H	INTP0	001CH	INTTMH0
0008H	INTP1	001EH	INTTM50
000AH	INTP2	0020H	INTTM000
000CH	INTP3	0022H	INTTM010
000EH	INTP4	002AH	INTTM51
0010H	INTP5	002CH	INTKR
0012H	INTSRE6	002EH	INTWT
0014H	INTSR6	003EH	BRK

(2) CALLT 指令表区域

64 字节区域 0040H 至 007FH 可以用于保存 1 字节调用指令（CALLT）的子程序入口地址。

(3) 选项字节区域

0080H 到 0084H 的 5 字节区域可以用作选项字节区域。详细信息，请参见第十七章 选项字节。

(4) CALLF 指令入口区域

0800H 至 0FFFH 区域可以通过 2 字节调用指令（CALLF）直接执行子程序调用。

(5) 片上调试安全 ID 设置区域

0085H 到 008EH 的 10 字节区域可以用作片上安全 ID 设置区域。详细信息，请参见第十九章 片上调试功能。

3.1.2 内部数据存储空间

μ PD179F11x, 179F12x 微控制器包含以下几种 RAM。

(1) 内部高速 RAM

表 3-5. 内部高速 RAM 容量

产品代号	内部高速 RAM
μ PD179F110	512 × 8 位 (FD00H 到 FEFFH)
μ PD179F111	
μ PD179F112, 179F122	768 × 8 位 (FC00H 到 FEFFH)
μ PD179F113, 179F123	1024 × 8 位 (FB00H 到 FEFFH)
μ PD179F114, 179F124	

32 字节区域 FEE0H 至 FEFFH 被分配到 4 个通用寄存器组中，每个组中包含有 8 个 8 位寄存器。该区域不能被用作指令进行写入和执行的程序区域。内部高速 RAM 也可以用作栈存储器。

3.1.3 特殊功能寄存器 (SFR) 区域

片上周边硬件特殊功能寄存器 (SFR) 位于 FF00H 至 FFFFH 的区域中 (参见 4.2.3 特殊功能寄存器 (SFR) 中的表 3-6 特殊功能寄存器列表)。

注意事项 不要访问 SFR 没有被分配的地址。

3.1.4 数据存储器寻址

寻址是根据指定下一个执行的指令的地址的方法，指定寄存器地址的方法，或者指定与指令执行相关的存储器的方法来执行的。

基于可操作性及其他一些考虑， μ PD179F11x, 179F12x 微控制器中与指令执行相关的存储器寻址具有多种模式。特别是对于那些包含有数据存储器的区域，将可以使用为特殊功能寄存器 (SFR) 以及通用寄存器的功能所设计的特殊的寻址方法。图 3-6 到 3-10 显示了数据存储器与寻址间的对应相系。关于各种寻址模式的详细信息，请参见 3.4 操作数地址寻址。

图 3-6. 数据存储器与寻址间的对应关系 (μ PD179F110)

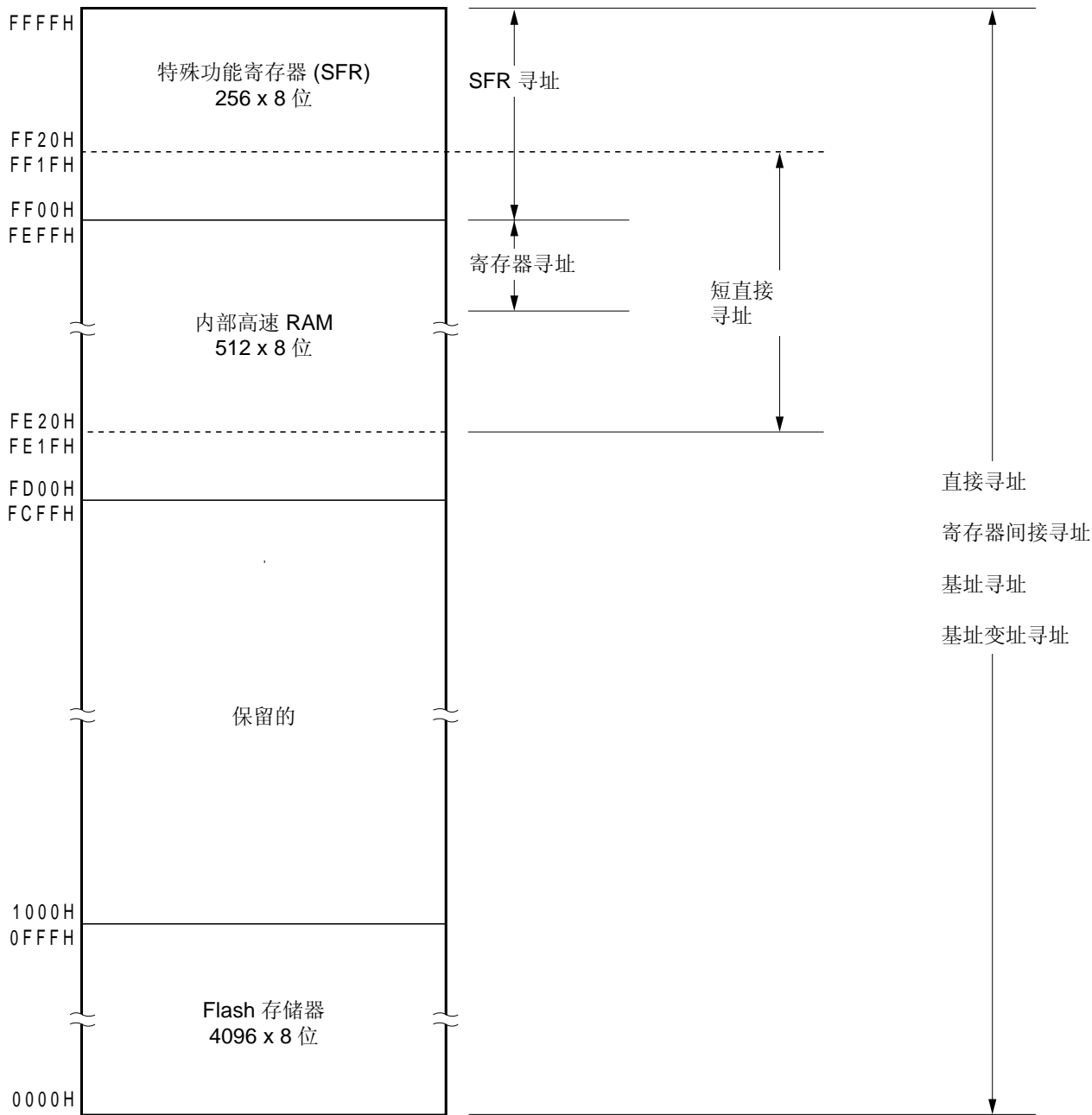


图 3-7. 数据存储器与寻址间的对应关系 (μPD179F111)

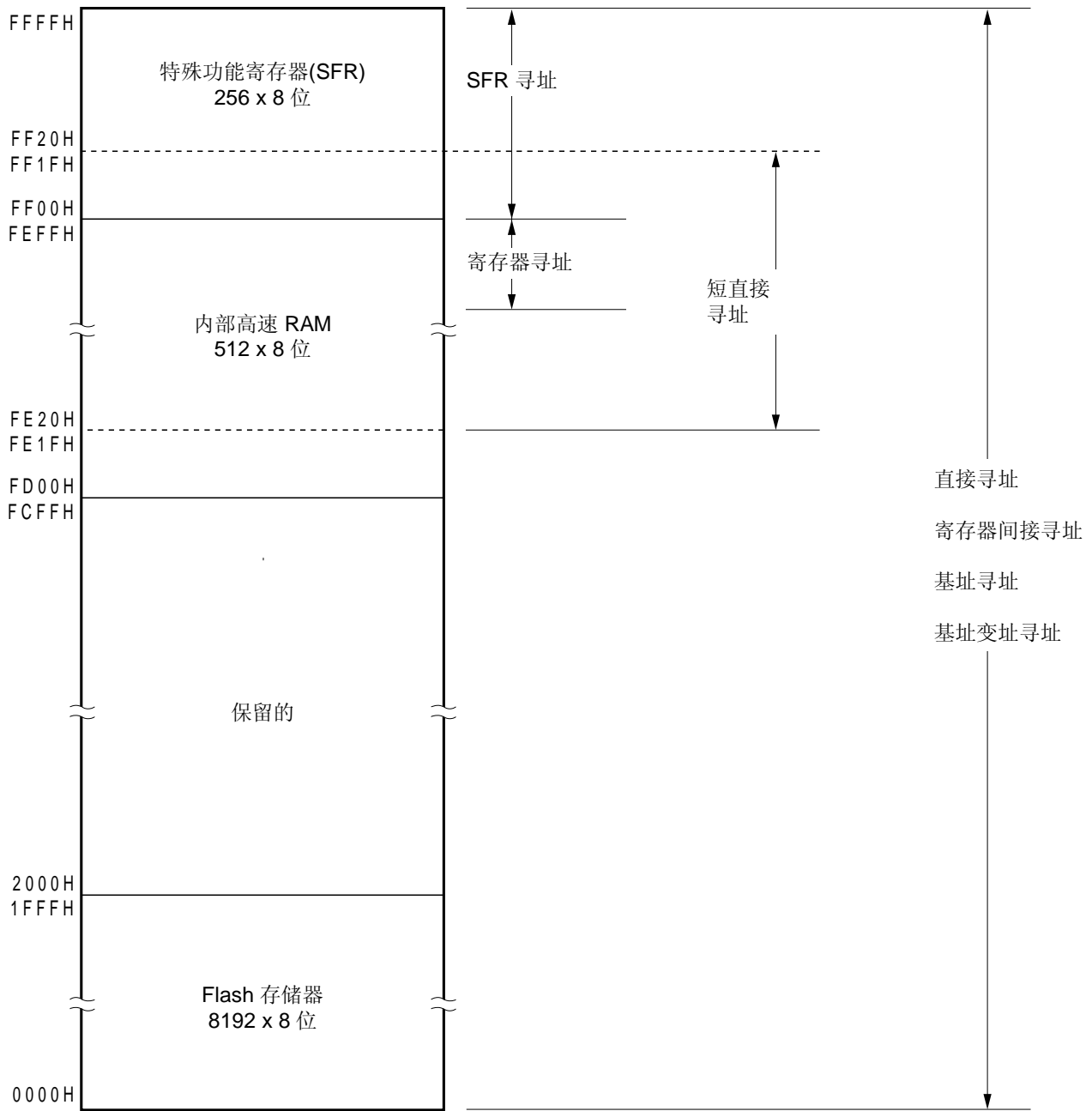


图 3-8. 数据存储器与寻址间的对应关系 (μ PD179F112 和 179F122)

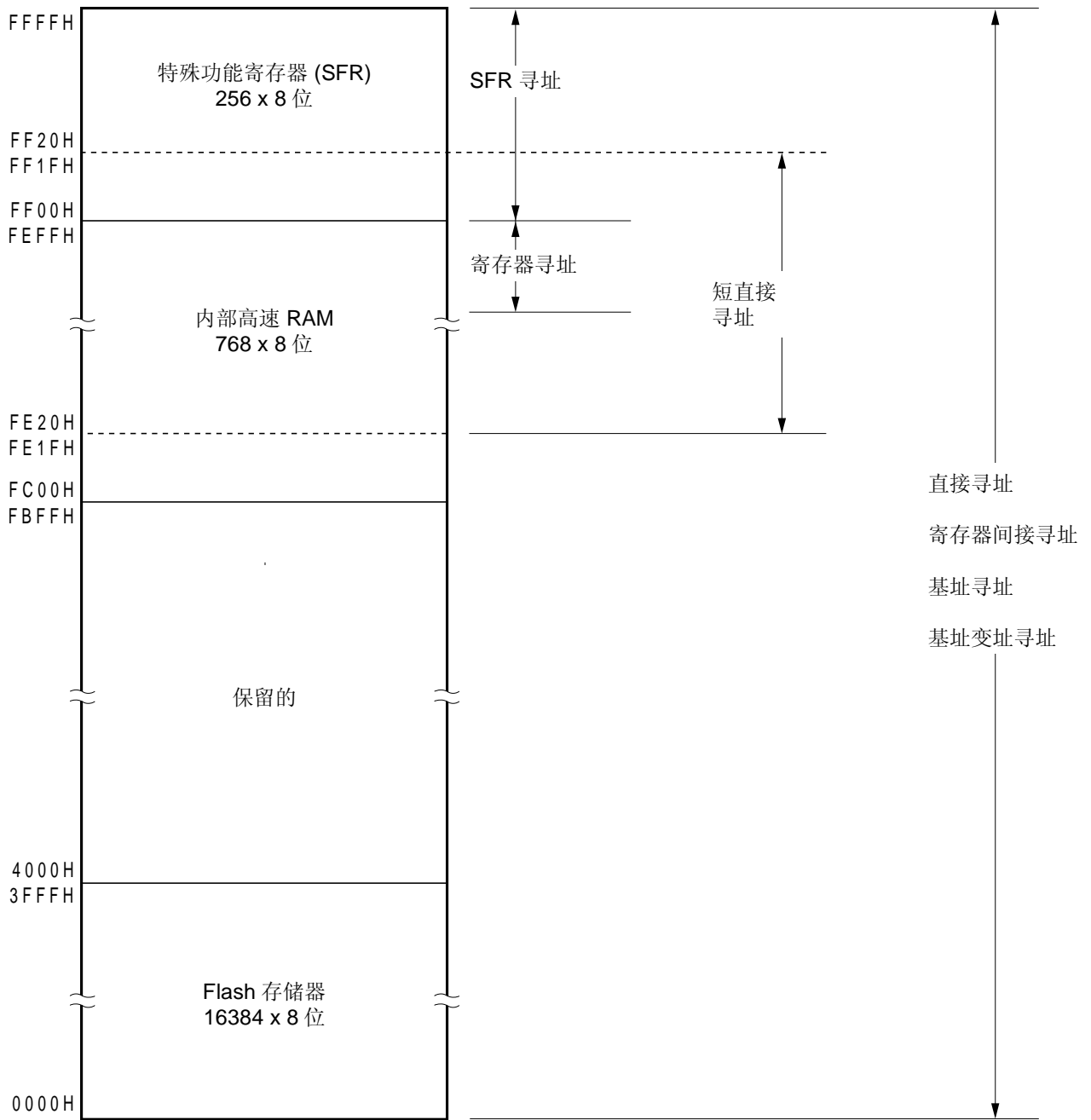


图 3-9.数据存储器与寻址间的对应关系 (μPD179F113 和 179F123)

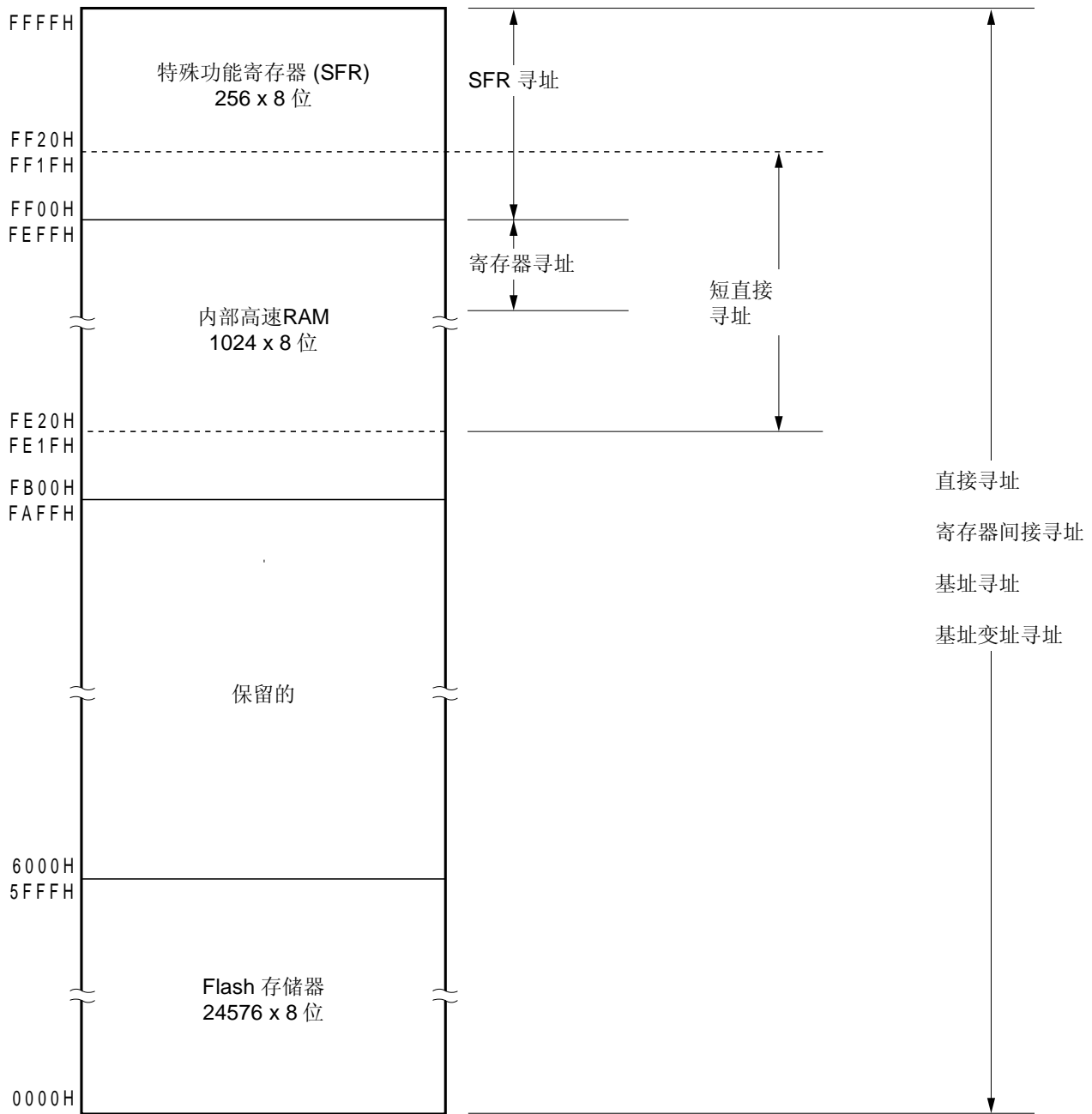
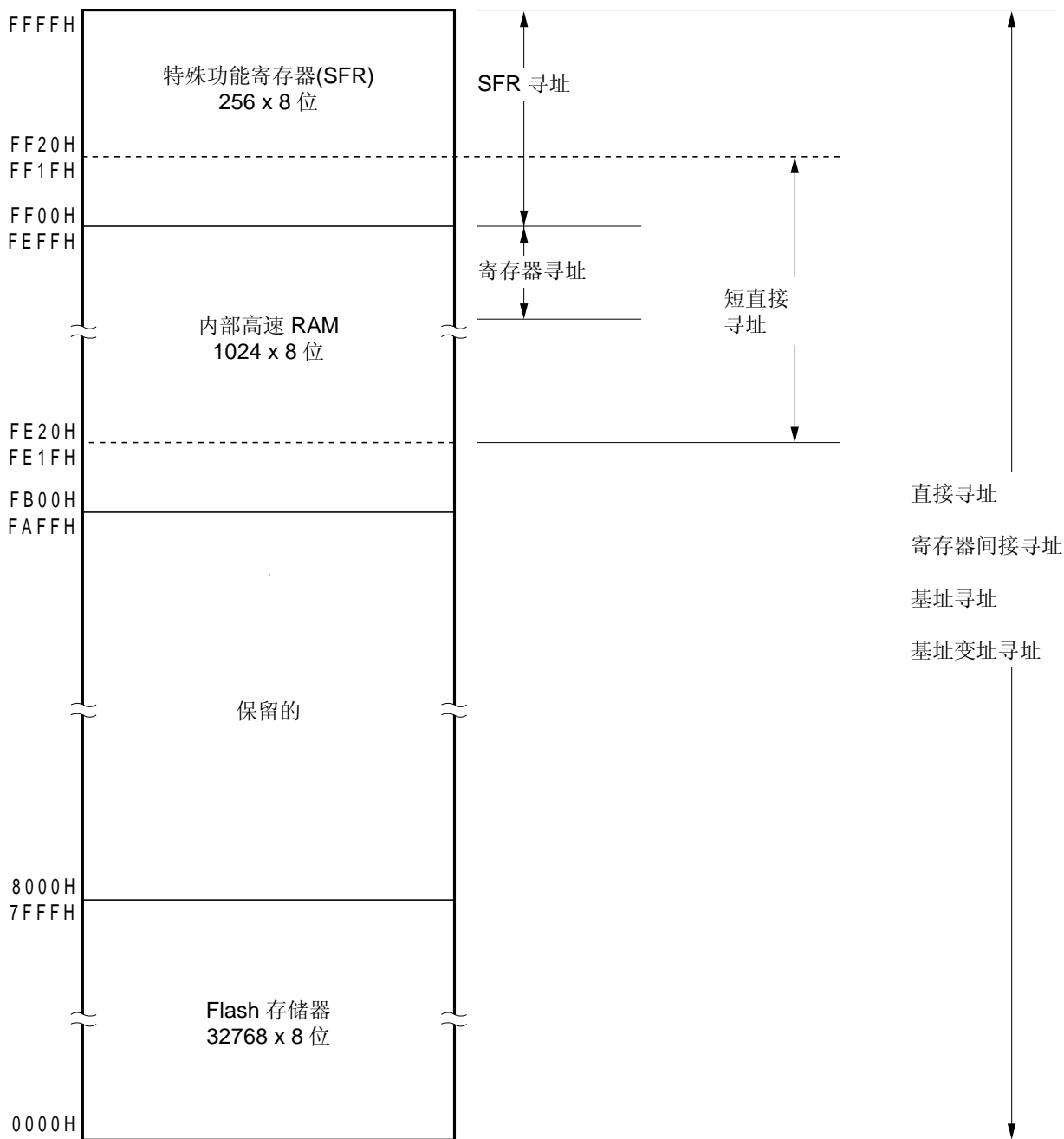


图 3-10. 数据存储器与寻址间的对应关系 (μPD179F114 和 179F124)



3.2 处理器寄存器

μ PD179F11x, 179F12x 微控制器包含有以下几种处理器寄存器。

3.2.1 控制寄存器

控制寄存器用于控制程序次序，状态以及栈存储器。控制寄存器由一个程序计数器（PC），一个程序状态字（PSW）以及一个堆栈指针（SP）组成。

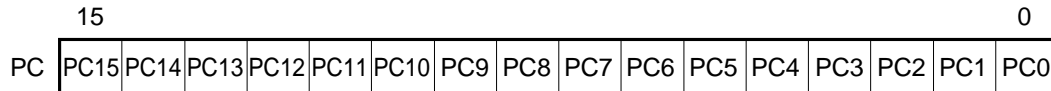
(1) 程序计数器（PC）

程序计数器是一个 16 位寄存器，用于保存下一个即将要执行的程序的地址信息。

在正常运行中，PC 会根据所获取的指令的字节数自动累加。执行分支指令时，则要对立即数据和寄存器内容进行设置。

复位信号生成会将地址 0000H 和 0001H 处的复位向量表的价值赋值到程序计数器中。

图 3-11. 程序计数器的格式



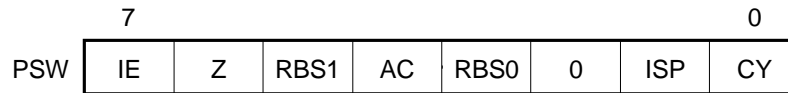
(2) 程序状态字（PSW）

程序状态字是一个 8 位寄存器，由不同的标志集组成，可通过指令执行进行复位。

中断请求生成或在 PUSH PSW 指令执行时程序状态字的内容会被保存在堆栈区域中，而在执行 RETB, RETI 以及 POP PSW 指令时程序状态字的内容将会被恢复。

复位信号生成将 PSW 设为 02H。

图 3-12. 程序状态字的格式



(a) 中断允许标志（IE）

该标志用于控制 CPU 的中断请求响应操作。

当 IE 为 0 时，IE 标志将会被设置为中断禁止（DI）状态，且所有可屏蔽中断请求都将会被禁止。

当 IE 为 1 时，IE 标志将会被设置为中断允许（EI）状态且中断请求响应将会通过服务优先级标志（ISP），各中断源的中断屏蔽标志，以及优先级指定标志来控制。

执行 DI 指令或中断响应时 IE 标志将会被复位为 0，在执行 EI 指令时 IE 标志则会被设置为 1。

(b) 零标志 (Z)

当操作结果为零时，该标志被设置为 1。其他情况下则复位为 0。

(c) 寄存器组选择标志 (RBS0 和 RBS1)

这些是 2 位标志，用于选择 4 个寄存器组中的一个组。

在这些标志中，保存有 2 位信息，2 位信息显示了 SEL RBn 指令执行所选择的寄存器组。

(d) 辅助进位标志 (AC)

如果操作结果中有一个来自第 3 位的进位或在第 3 位上有错位，那么该标志将会被设为 1。其他情况下则复位为 0。

(e) 服务优先级标志 (ISP)

该标志用于管理可屏蔽向量中断响应的优先级。当该标志为 0 时，将不能响应由优先级指定标志寄存器 (PR0L, PR0H, PR1L) 所指定的低级别向量中断请求 (参见 11.3 (3) 优先级指令标志寄存器 (PR0L, PR0H, PR1L))。实际请求的响应是由中断允许标志 (IE) 来控制的。

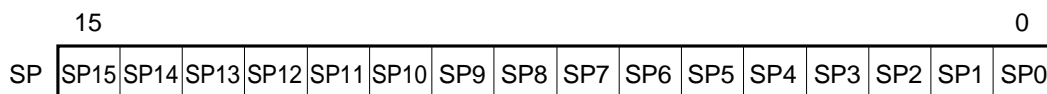
(f) 进位标志 (CY)

该标志用于保存执行加 / 减指令时的上溢和下溢。它也用于保存循环指令执行时的转换值，并可在位操作指令执行过程中用作位累加器。

(3) 堆栈指针 (SP)

这是一个 16 位寄存器，用于保存存储器堆栈区域的开始地址。只有内部高速 RAM 区域可以设置为堆栈区域。

图 3-13. 堆栈指针的格式



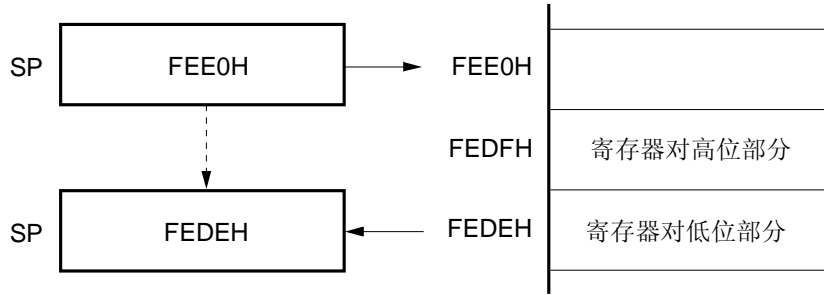
向堆栈存储器进行写 (保存) 操作前 SP 累减，从堆栈存储器进行读取 (恢复) 操作后 SP 累加。

各个堆栈保存 / 恢复数据的操作如图 3-14 和 3-15 所示。

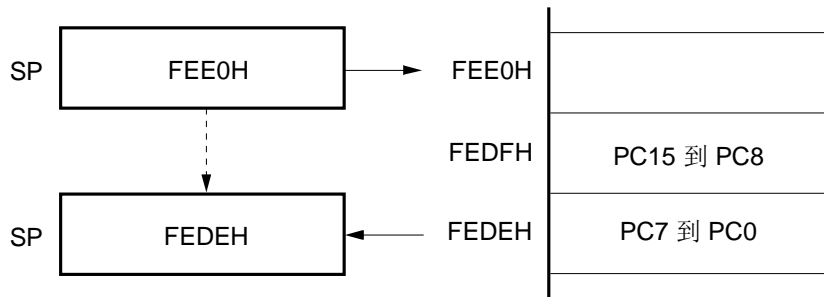
注意事项 由于复位信号生成会使用 SP 内容变为未定义，因此在使用堆栈前请务必初始化 SP。

图 3-14. 保存在堆栈存储器中的数据

(a) PUSH rp 指令 (当 SP = FEE0H 时)



(b) CALL, CALLF, CALLT 指令 (当 SP = FEE0H 时)



(c) 中断, BRK 指令 (当 SP = FEE0H 时)

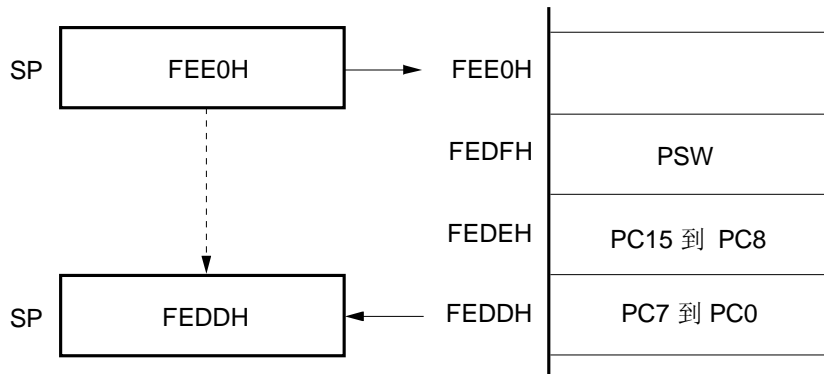
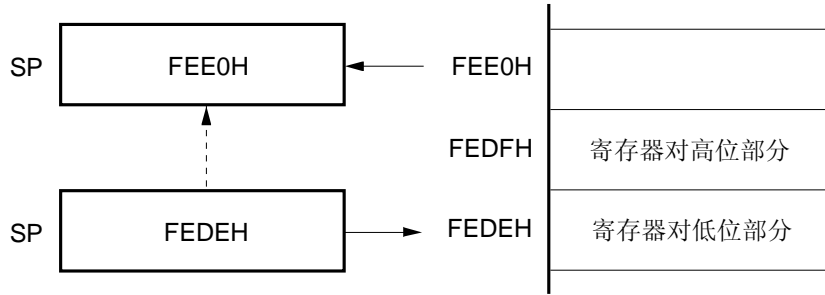
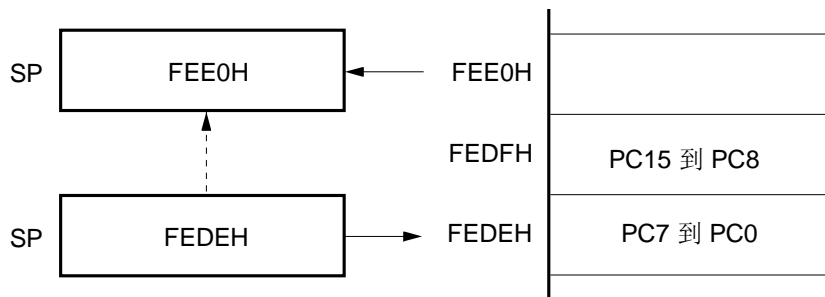


图 3-15. 从堆栈存储器中恢复的数据

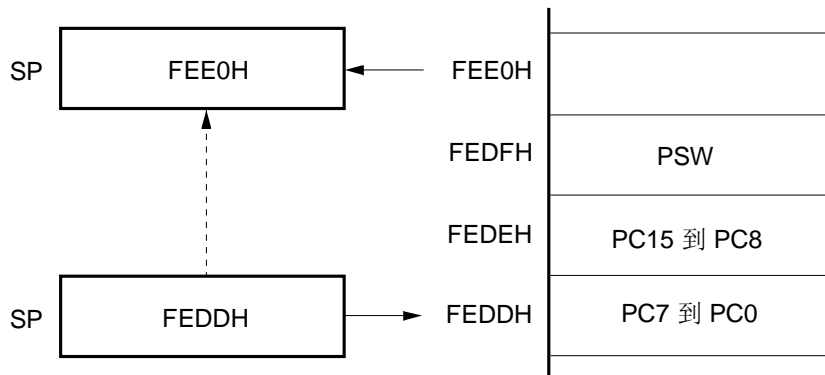
(a) POP rp 指令 (当 SP = FEDEH 时)



(b) RET 指令 (当 SP = FEDEH 时)



(c) RETI, RETB 指令 (当 SP = FEDDH 时)



3.2.2 通用寄存器

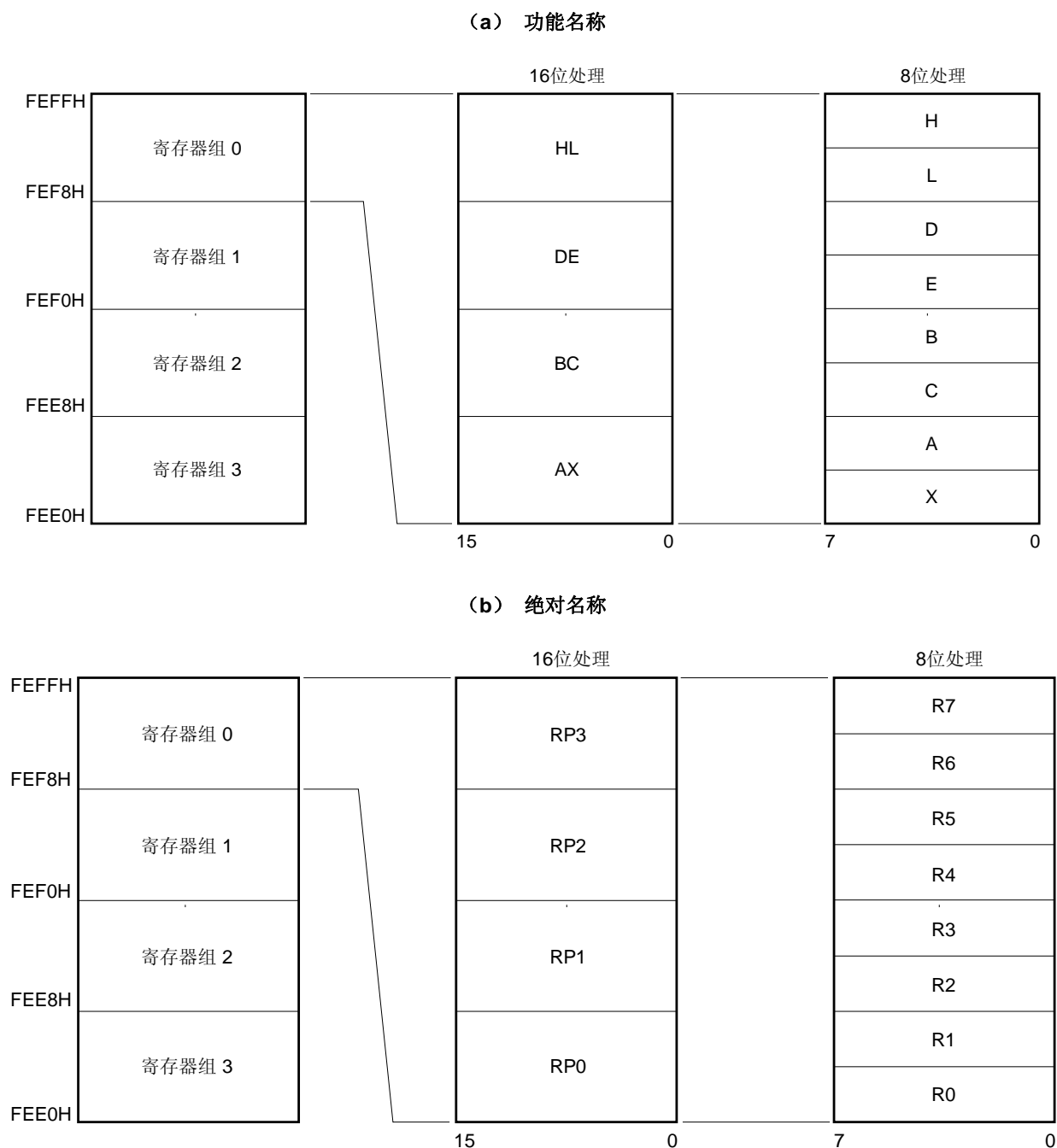
通用寄存器映射在数据存储器的特定地址（FEE0H 至 FEF8H）上。通用寄存器由 4 个组组成，每个组由 8 个 8 位寄存器（X, A, C, B, E, D, L, 以及 H）组成。

每个寄存器都可以用作一个 8 位寄存器，两个 8 位寄存器也可以作为一个 16 位寄存器（AX, BC, DE, 和 HL）来使用。

这些寄存器可以使用功能名称（X, A, C, B, E, D, L, H, AX, BC, DE, 和 HL）和绝对名称（R0 至 R7 以及 RP0 至 RP3）来进行描述。

用于指令执行的寄存器组通过 CPU 控制指令（SEL RBn）来进行设置。根据 4 寄存器组的结构特点，可以通过在用于正常处理的寄存器以及用于各个组的中断的寄存器间进行转换来创建一个有效程序。

图 3-16. 通用寄存器的结构



3.2.3 特殊功能寄存器 (SFR)

与通用寄存器不同，每个特殊功能寄存器都有一个特殊功能。

SFR 被分配到 FF00H 至 FFFFH 区域中。

通过使用操作，转移以及位操作指令，特殊功能寄存器可以像通用寄存器那样进行操作。根据特殊寄存器的类型，可操作位单元可以是 1，8 以及 16。

每个操作位单元可以按如下进行指令。

- 1 位操作
1 位操作指令操作数 (sfr.位) 被描述为汇编程序保留的符号。
该操作也可以通过使用一个地址来指令。
- 8 位操作
8 位操作指令操作数 (sfr) 被描述为汇编程序保留的符号。
该操作也可以通过使用一个地址来指令。
- 16 位操作
16 位操作指令操作数 (sfrp) 被描述为汇编程序保留的符号。
指定一个地址时应表示为一个偶地址。

表 3-6 列出了特殊功能寄存器。表格中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。它是 RA78K0 中的保留字，而在 CC78K0 中则通过使用 #pragma sfr 导引来定义为一个 sfr 变量。使用 RA78K0，ID78K0-QB，和 SM+ for 78K0/KX2 时，符号可以被写作指令操作数。
- R/W
表示相应的特殊功能寄存器是被读取还是被写入。
R/W: 允许读取 / 写入
R: 只读
W: 只写
- 可操作位单元
表示可操作位单元 (1, 8, 或 16)。“-”表示操作不能实现的位单元。
- 复位后
表示复位信号生成时各寄存器的状态。

表 3-6. 特殊功能寄存器列表 (1/3)

地址	特殊功能寄存器 (SFR) 的名称	符号	R/W	可操作位单元			复位后
				1 位	8 位	16 位	
FF00H	端口寄存器 0	P0	R/W	√	√	–	00H
FF01H	端口寄存器 1	P1	R/W	√	√	–	00H
FF02H	端口寄存器 2	P2	R/W	√	√	–	00H
FF03H	端口寄存器 3	P3	R/W	√	√	–	00H
FF0AH	接收缓冲寄存器 6	RXB6	R	–	√	–	FFH
FF0BH	发送缓冲寄存器 6	TXB6	R/W	–	√	–	FFH
FF0CH	端口寄存器 12	P12	R/W	√	√	–	00H
FF10H	16 位定时器计数器 00	TM00	R	–	–	√	0000H
FF11H							
FF12H	16 位定时器捕获 / 比较寄存器 000	CR000	R/W	–	–	√	0000H
FF13H							
FF14H	16 位定时器捕获 / 比较寄存器 010	CR010	R/W	–	–	√	0000H
FF15H							
FF16H	8 位定时器计数器 50	TM50	R	–	√	–	00H
FF17H	8 位定时器比较寄存器 50	CR50	R/W	–	√	–	00H
FF18H	8 位定时器 H 比较寄存器 00	CMP00	R/W	–	√	–	00H
FF19H	8 位定时器 H 比较寄存器 10	CMP10	R/W	–	√	–	00H
FF1AH	8 位定时器 H 比较寄存器 01	CMP01	R/W	–	√	–	00H
FF1BH	8 位定时器 H 比较寄存器 11	CMP11	R/W	–	√	–	00H
FF1FH	8 位定时器计数器 51	TM51	R	–	√	–	00H
FF20H	端口模式寄存器 0	PM0	R/W	√	√	–	FFH
FF21H	端口模式寄存器 1	PM1	R/W	√	√	–	FFH
FF22H	端口模式寄存器 2	PM2	R/W	√	√	–	FFH
FF23H	端口模式寄存器 3	PM3	R/W	√	√	–	FFH
FF2CH	端口模式寄存器 12	PM12	R/W	√	√	–	FFH
FF2EH	复位引脚模式寄存器	RSTMASK	R/W	√	√	–	00H
FF30H	上拉电阻选项寄存器 0	PU0	R/W	√	√	–	00H
FF31H	上拉电阻选项寄存器 1	PU1	R/W	√	√	–	00H
FF32H	上拉电阻选项寄存器 2	PU2	R/W	√	√	–	00H
FF33H	上拉电阻选项寄存器 3	PU3	R/W	√	√	–	00H
FF35H	FLMDO 上拉/下拉控制寄存器	FPCTL	R/W	√	√	–	00H
FF37H	FLMDO 上拉/下拉允许寄存器	FPEN	R/W	√	√	–	00H
FF38H	端口输出模式寄存器 0	POM0	R/W	√	√	–	00H
FF39H	端口输出模式寄存器 1	POM1	R/W	√	√	–	00H
FF3AH	端口输出模式寄存器 2	POM2	R/W	√	√	–	00H
FF3BH	端口输出模式寄存器 3	POM3	R/W	√	√	–	00H
FF3CH	上拉电阻选项寄存器 12	PU12	R/W	√	√	–	08H
FF3EH	端口输出模式寄存器 12	POM12	R/W	√	√	–	00H
FF41H	8 位定时器比较寄存器 51	CR51	R/W	–	√	–	00H
FF43H	8 位定时器模式控制寄存器 51	TMC51	R/W	√	√	–	00H
FF48H	外部中断上升沿允许寄存器	EGP	R/W	√	√	–	00H
FF49H	外部中断下降沿允许寄存器	EGN	R/W	√	√	–	00H

表 3-6. 特殊功能寄存器列表 (2/3)

地址	特殊功能寄存器 (SFR) 的名称	符号	R/W	可操作位单元			复位后	
				1 位	8 位	16 位		
FF50H	异步串行接口操作模式寄存器 6	ASIM6	R/W	√	√	–	01H	
FF53H	异步串行接口接收错误状态寄存器 6	ASIS6	R	–	√	–	00H	
FF55H	异步串行接口发送状态寄存器 6	ASIF6	R	–	√	–	00H	
FF56H	时钟选择寄存器 6	CKSR6	R/W	–	√	–	00H	
FF57H	波特率发生器控制寄存器 6	BRGC6	R/W	–	√	–	FFH	
FF58H	异步串行接口控制寄存器 6	ASICL6	R/W	√	√	–	16H	
FF69H	8 位定时器 H 模式寄存器 0	TMHMD0	R/W	√	√	–	00H	
FF6AH	定时器时钟选择寄存器 50	TCL50	R/W	√	√	–	00H	
FF6BH	8 位定时器模式控制寄存器 50	TMC50	R/W	√	√	–	00H	
FF6CH	8 位定时器 H 模式寄存器 1	TMHMD1	R/W	√	√	–	00H	
FF6DH	8 位定时器 H 载波控制寄存器 1	TMCYC1	R/W	√	√	–	00H	
FF6EH	按键中断模式寄存器 0	KRML	R/W	√	√	–	00H	
FF6FH	按键中断模式寄存器 1	KRMH	R/W	√	√	–	00H	
FF8CH	定时器时钟选择寄存器 51	TCL51	R/W	√	√	–	00H	
FF99H	看门狗定时器允许寄存器	WDTE	R/W	–	√	–	注 1 1AH/9AH	
FF9FH	时钟操作模式选择寄存器	OSCCTL	R/W	√	√	–	00H	
FFA0H	内部振荡模式寄存器	RCM	R/W	√	√	–	80H 注 2	
FFA1H	主时钟模式寄存器	MCM	R/W	√	√	–	00H	
FFA2H	主 OSC 控制寄存器	MOC	R/W	√	√	–	80H	
FFA3H	振荡稳定时间计数状态寄存器	OSTC	R	√	√	–	00H	
FFA4H	振荡稳定时间选择寄存器	OSTS	R/W	–	√	–	05H	
FFACH	复位控制标志寄存器	RESF	R	–	√	–	00H 注 3	
FFB0H	RAM 数据保存控制寄存器	LVDET	R/W	√	√	–	未定义	
FFBAH	16 位定时器模式控制寄存器 00	TMC00	R/W	√	√	–	00H	
FFBBH	预分频模式寄存器 00	PRM00	R/W	√	√	–	00H	
FFBCH	捕获/比较控制寄存器 00	CRC00	R/W	√	√	–	00H	
FFBDH	16 位定时器输出控制寄存器 00	TOC00	R/W	√	√	–	00H	
FFBEH	低电压检测寄存器	LVIM	R/W	√	√	–	00H 注 3	
FFBFH	低电压检测电平选择寄存器	LVIS	R/W	√	√	–	00H 注 3	
FFE0H	中断请求标志寄存器 0L	IF0	IF0L	R/W	√	√	√	00H
FFE1H	中断请求标志寄存器 0H		IF0H	R/W	√	√		00H
FFE2H	中断请求标志寄存器 1L	IF1	IF1L	R/W	√	√	–	00H
FFE4H	中断屏蔽标志寄存器 0L	MK0	MK0L	R/W	√	√	√	FFH
FFE5H	中断屏蔽标志寄存器 0H		MK0H	R/W	√	√		FFH
FFE6H	中断屏蔽标志寄存器 1L	MK1	MK1L	R/W	√	√	–	FFH

- 注
1. WDTE 的复位值由选项字节的设置来决定。
 2. 此寄存器值在复位释放后立即为 00H，但当高速内部振荡器已振荡稳定后自动变为 80H。
 3. RESF, LVIM, t LVIS 的复位值根据复位源的不同而变化。

表 3-6. 特殊功能寄存器列表 (3/3)

地址	特殊功能寄存器 (SFR) 的名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
FFE8H	优先级指定标志寄存器 0L	PR0	PR0L	R/W	√	√	√	FFH
FFE9H	优先级指定标志寄存器 0H		PR0H	R/W	√	√		FFH
FFEAH	优先级指定标志寄存器 1L	PR1	PR1L	R/W	√	√	–	FFH
FFF0H	内部存储器容量切换寄存器 ^注	IMS		R/W	–	√	–	CFH
FFF9H	内部高速振荡器时序寄存器	R4MTRM2		R/W	–	√	–	00H
FFFBH	处理器时钟控制寄存器	PCC		R/W	√	√	–	01H

注 不管内部存储器容量有多少， μ PD179F11x, 179F12x 微控制器中所有产品的内部存储器容量切换寄存器 (IMS) 的初始值都是固定的 (IMS = CFH)。因此，应按以下所显示的内容对各个产品进行相应的值设置。

Flash 存储器版本 (μ PD179F11x, 179F12x 微控制器)	IMS	ROM 容量	内部高速 RAM 容量
μ PD179F110	41H	4 KB	512 字节
μ PD179F111	42H	8 KB	
μ PD179F112, 179F122	04H	16 KB	768 字节
μ PD179F113, 179F123	C6H	24 KB	1 KB
μ PD179F114, 179F124	C8H	32 KB	

3.3 指令地址寻址

指令地址由程序计数器（PC）的内容决定，并且在每次执行另一指令时会根据所获取的指令的字节数自动累加（每个字节+1）。执行转移指令时，转移目的信息会被赋值到 PC 中并且通过以下寻址方法确定地址（关于指令的详细信息，请参照 **78K/0 系列指令用户手册（U12326E）**）。

3.3.1 相对寻址

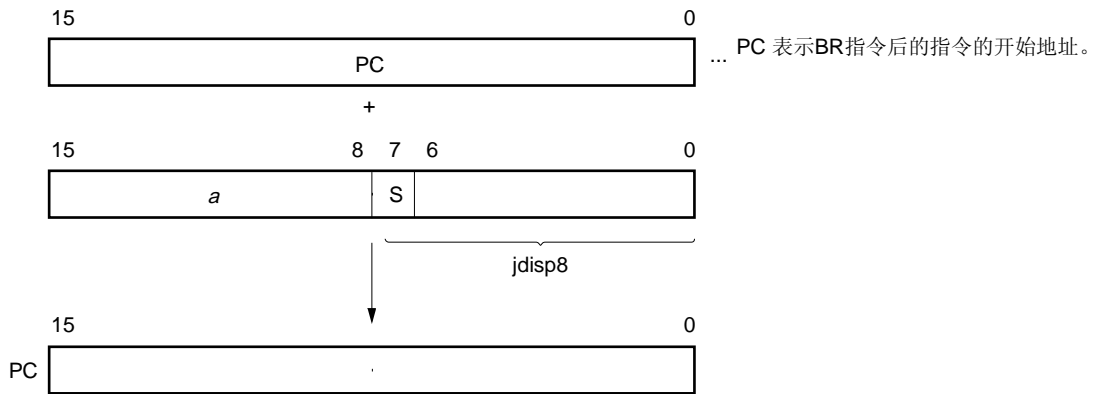
[功能]

将指令代码的 8 位立即数据（替换值：**jdisp8**）与以下指令中的开始地址相加后所得的值传送到程序计数器（PC）中并转向相加结果指向的地址。替换值被当作带符号位的二进制补码（-128 至 +127），而第 7 位则变为一个符号位。

换句话说，相对寻址是由从下一条指令的开始地址开始，范围在 -128 至 +127 内。

该功能可以在执行 **BR \$addr16** 指令或条件转移指令时实现。

[图示]



当S = 0时，a的所有位为0
 当S = 1时，a的所有位为1

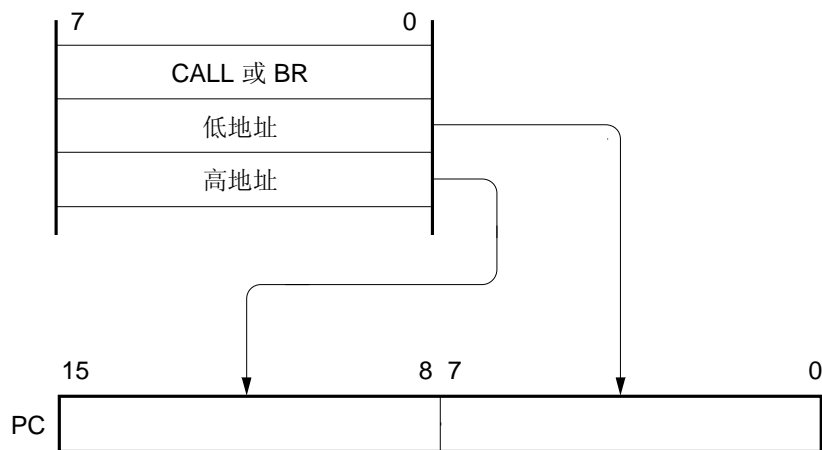
3.3.2 立即寻址

[功能]

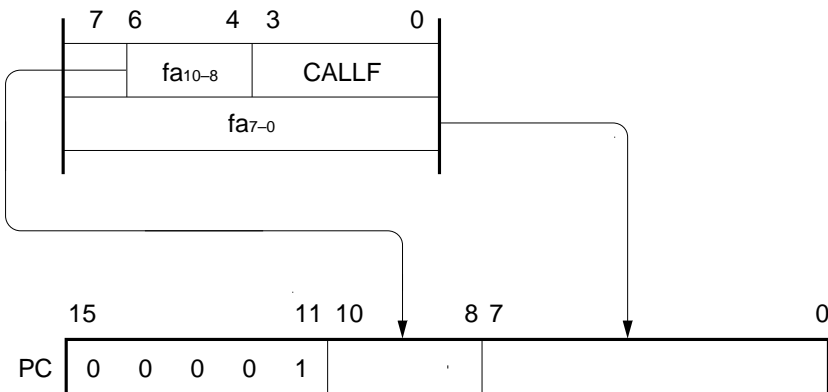
指令字中的立即数据被传送到程序计数器（PC）中并转向该地址。
 该功能可以在执行 CALL !addr16 或 BR !addr16 或 CALLF !addr11 指令时实现。
 CALL !addr16 和 BR !addr16 指令可以转移到整个内存空间中。
 CALLF !addr11 则被转移到 0800H 至 0FFFH 的区域中。

[图示]

使用 CALL !addr16 和 BR !addr16 指令时



使用 CALLF !addr11 指令时



<R> 3.3.3 表间接寻址

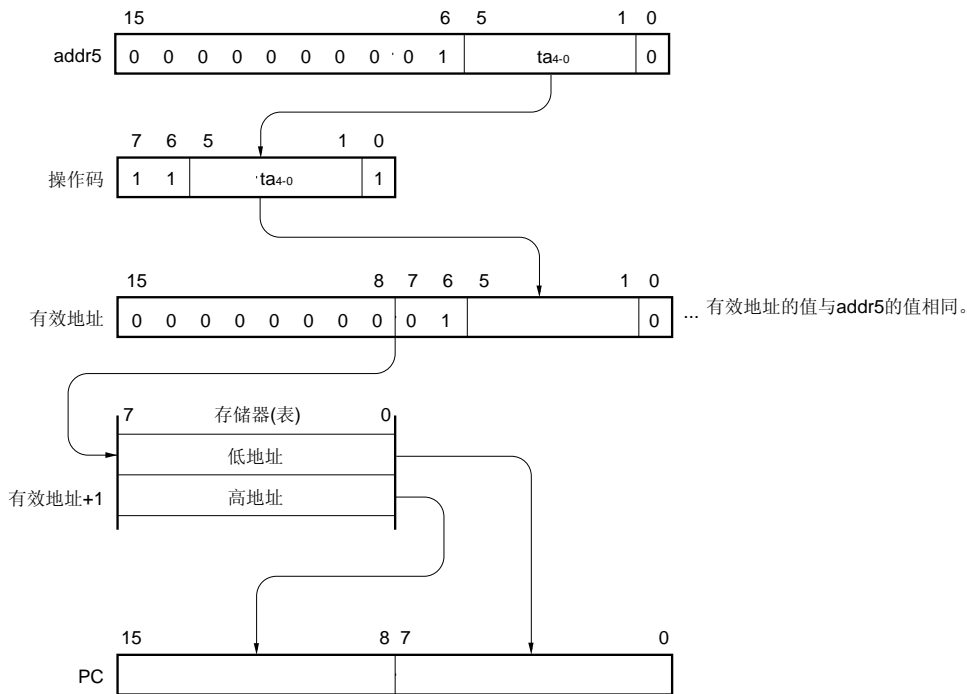
[功能]

通过操作数的立即数据的 1 到 5 位寻址的特定位置的表内容（转移目的地址）被赋值给程序计数器（PC）并转向该地址。

该功能在执行 CALLT [addr5]指令时实现。

该指令参考了由 addr5 表示的并且保存在范围从 0040H 至 007FH 的存储器表的地址中，并且允许跳转至整个内存空间。

[图示]



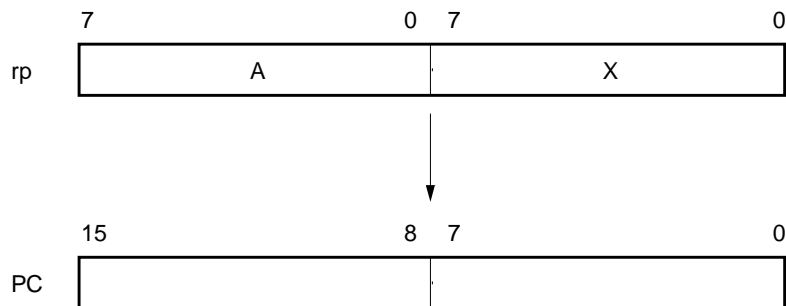
3.3.4 寄存器寻址

[功能]

通过指令字指定的寄存器对（AX）内容可以被赋值给程序计数器（PC）并且转向该地址。

该功能在执行 BR AX 指令时实现。

[图示]



3.4 操作数地址寻址

可以通过以下方法来指定寄存器和存储器（寻址）用以实现指令执行过程中的操作。

3.4.1 隐含寻址

[功能]

在通用寄存器中用作累加器（A 和 AX）的寄存器会自动（隐含）寻址。

在 μ PD179F11x, 179F12x 微控制器指令字中，以下指令使用隐含寻址。

指令	通过隐含寻址指令的寄存器
MULU	A 寄存器用于存放被乘数而 AX 寄存器用于存放运算结果。
DIVUW	AX 寄存器用于存放被除数和商。
ADJBA/ADJBS	A 寄存器用于存放那些进行小数更正的数值。
ROR4/ROL4	A 寄存器用于存放进行数字循环的数字数据。

[操作数格式]

由于隐含寻址可以通过指令自动进行，因此不需要特殊的操作数格式。

[说明示例]

使用 MULU X 时

使用 8 位 \times 8 位乘法指令时，A 寄存器与 X 寄存器的乘积将会被保存在 AX 中。该示例中，A 和 AX 寄存器是通过隐含寻址的方式指定的。

3.4.2 寄存器寻址

[功能]

寄存器寻址方式通过寄存器组选择标志（RBS0 至 RBS1）以及操作代码的寄存器指定代码将指定的通用寄存器作为操作数进行访问。

执行带有以下操作数格式的指令时可以进行寄存器寻址。当指定了一个 8 位寄存器时，操作代码中将会有 3 位用于指定 8 个寄存器中的一个。

[操作数格式]

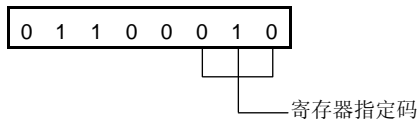
标识符	说明
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

‘r’ 和 ‘rp’ 可以使用绝对名称（R0 至 R7 以及 RP0 至 RP3）进行描述，也可以使用功能名称（X, A, C, B, E, D, L, H, AX, BC, DE, 和 HL）来进行描述。

[说明示例]

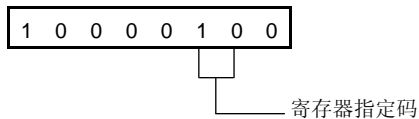
MOV A, C: 将 C 寄存器选择为 r 时

操作代码



INCW DE: 将 DE 寄存器对选作 rp 时

操作代码



3.4.3 直接寻址

[功能]

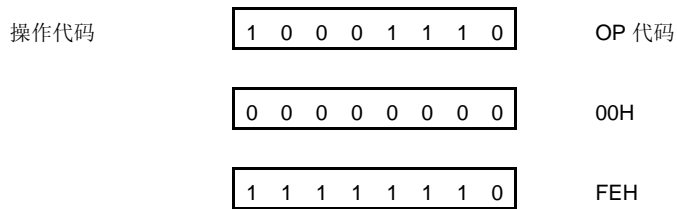
要进行操作的存储器通过变为操作数地址的指令字中的立即数据直接进行寻址。
所有存储空间都可以实现该寻址。

[操作数格式]

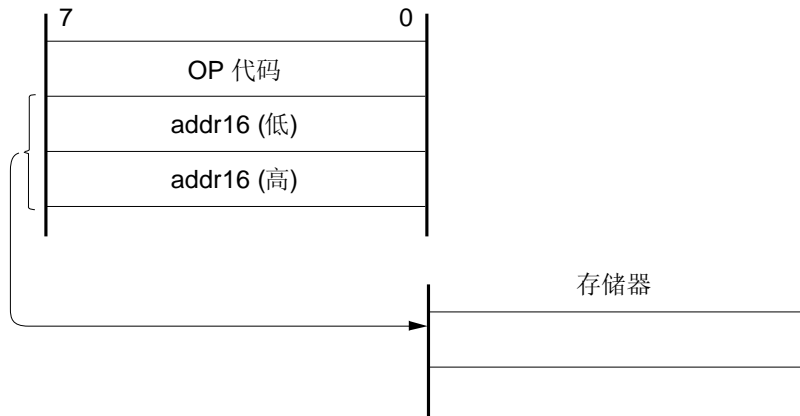
标识符	说明
addr16	标签或 16 位立即数据

[说明示例]

MOV A, !0FE00H; 将 !addr16 设置到 FE00H 中时



[图示]



3.4.4 短直接寻址

[功能]

在固定空间中要进行操作的存储空间可以通过指令字中的 8 位数据直接进行访问。

该寻址方式应用于 256 字节的空间 FE20H 至 FF1FH 中。内部高速 RAM 和特殊功能寄存器 (SFR) 分别映射在 FE20H 至 FEFFH 以及 FF00H 至 FF1FH 的区域。

采用短直接寻址的 SFR 区域 (FF00H 至 FF1FH) 是整个 SFR 区域中的一部分。在程序中快速访问的端口和定时器 / 事件计数的比较和捕获寄存器映射在该区域中, 且 SFR 可以使用少量的字节和时钟进行操作。

当 8 位立即数据在 20H 至 FFH 内时, 有效地址的第 8 位将会被置 0。当 8 位立即数据在 00H 至 1FH 内时, 有效地址的第 8 位则将会被设置为 1。参见以下的[图示]。

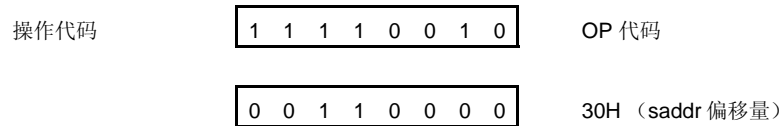
[操作数格式]

标识符	说明
saddr	表示标签或 FE20H 至 FF1FH 的立即数据
saddrp	表示标签或 FE20H 至 FF1FH 的立即数据 (仅限偶地址)

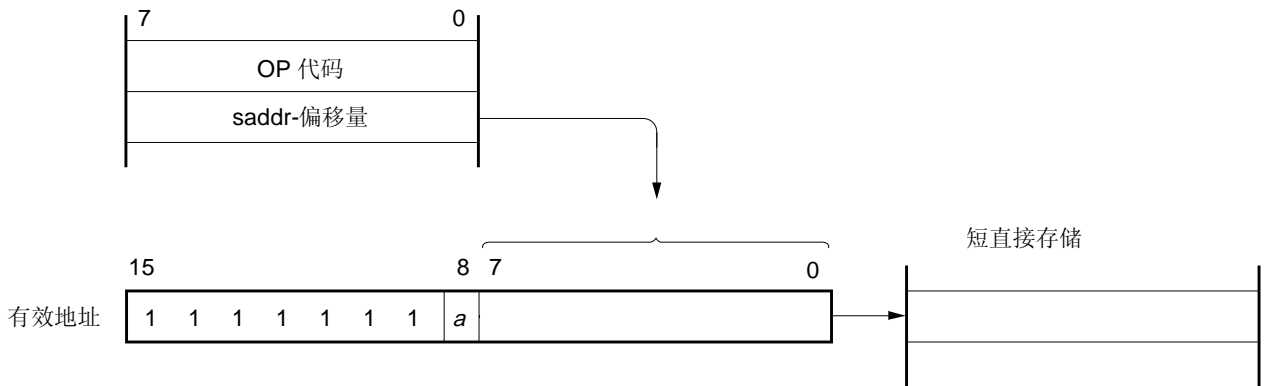
[说明示例]

```

LB1 EQU 0FE30H ;      用 LB1 来定义 FE30H。
:
MOV LB1, A      ;      当 LB1 表示 saddr 区域的 FE30H 且寄存器 A 的值被转移到该地址中时
    
```



[图示]



当 8 位立即数据在 20H 至 FFH 间时, $\alpha = 0$

当 8 位立即数据在 00H 至 1FH 间时, $\alpha = 1$

3.4.5 特殊功能寄存器 (SFR) 寻址

[功能]

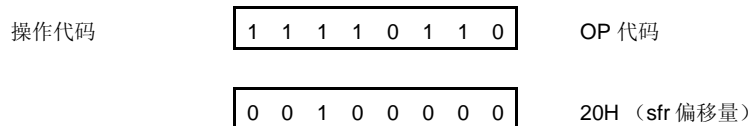
通过指令字中的 8 位立即数据对存储器映射特殊功能寄存器 (SFR) 进行寻址。
 该寻址方式应用于 FF00H 至 FFCFH 以及 FFE0H 至 FFFFH 的 240 个字节的空间内。然而，映射在 FF00H 至 FF1FH 内的 SFR 可以通过短直接寻址方式进行访问。

[操作数格式]

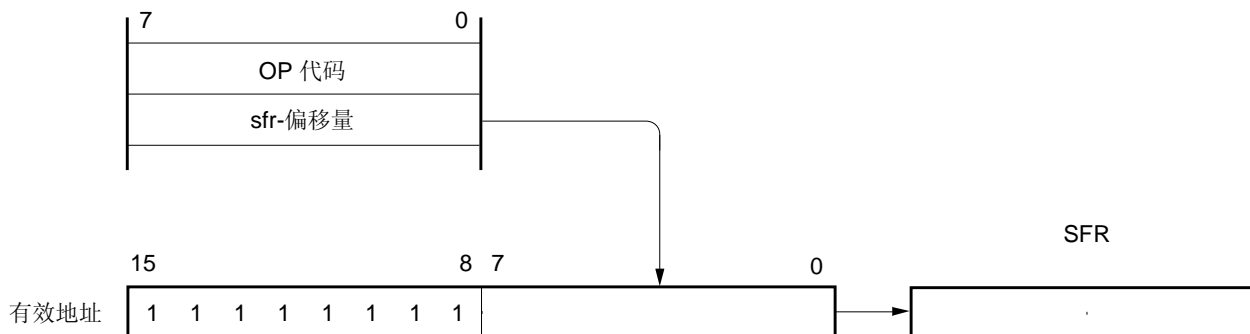
标识符	说明
sfr	特殊功能寄存器名
sfrp	16 位可操作特殊功能寄存器名 (仅限偶地址)

[说明示例]

MOV PM0, A: 当 PM0 (FF20H) 被选作 sfr 时



[图示]



3.4.6 寄存器间接寻址

[功能]

由指令字中的寄存器对指定代码以及寄存器组选择标志（RBS0 和 RBS1）指定的寄存器对的内容用作对存储器进行寻址的操作数地址。
所有存储空间都可以实现该寻址。

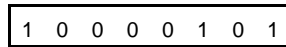
[操作数格式]

标识符	说明
-	[DE], [HL]

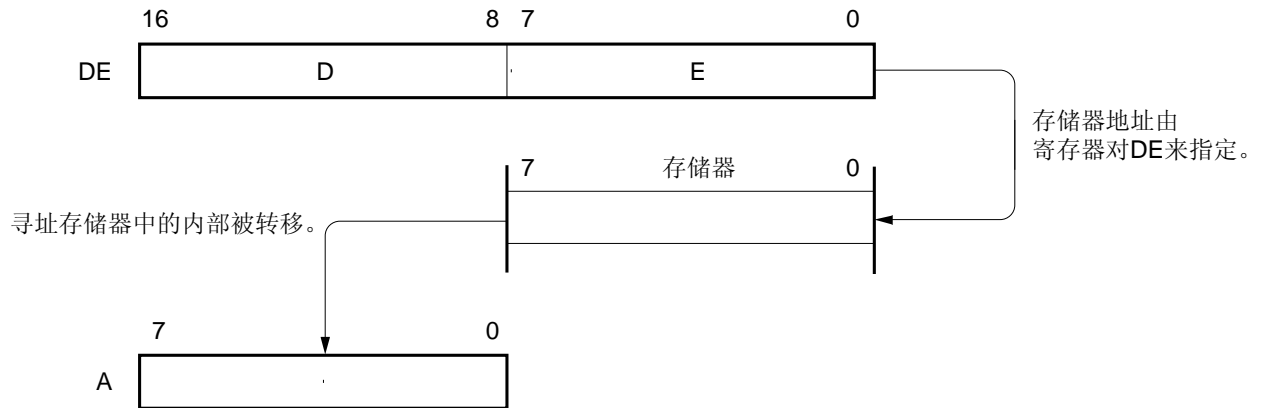
[说明示例]

MOV A, [DE]; 当[DE]被选作寄存器对时

操作代码



[图示]



3.4.7 基址寻址

[功能]

8 位立即数据作为偏移数据被加至基址寄存器的内容中，也就是说寄存器组中的 HL 寄存器对是由寄存器组选择标志（RBS0 和 RBS1）指定的，并会根据相加结果对寄存器进行寻址。加法操作是通过将偏移数据扩展至 16 位的正数来完成的。第 16 位的进位忽略不计。

所有存储空间都可以实现该寻址。

[操作数格式]

标识符	说明
-	[HL + byte]

[说明示例]

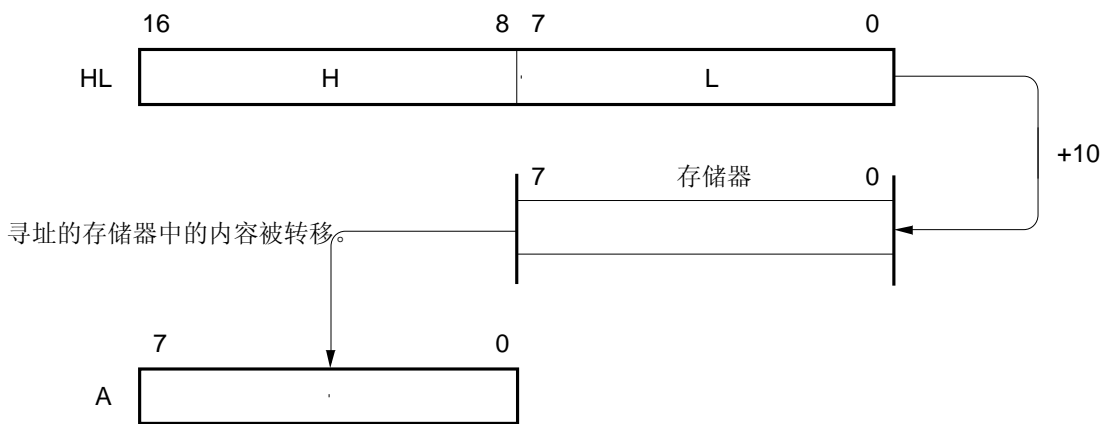
MOV A, [HL + 10H]; 将字节设置为 10H 时

操作代码

1 0 1 0 1 1 1 0

0 0 0 1 0 0 0 0

[图示]



3.4.8 基址变址寻址

[功能]

在指令中指令的 B 或 C 寄存器内容被加至基址寄存器的内容中，也就是说寄存器组中的 HL 寄存器对是由寄存器组选择标志（RBS0 和 RBS1）指定的，并会根据相加结果对寄存器进行寻址。加法操作是通过将 B 或 C 寄存器内容扩展至 16 位的正数来完成的。第 16 位的进位忽略不计。
所有存储空间都可以实现该寻址。

[操作数格式]

标识符	说明
-	[HL + B], [HL + C]

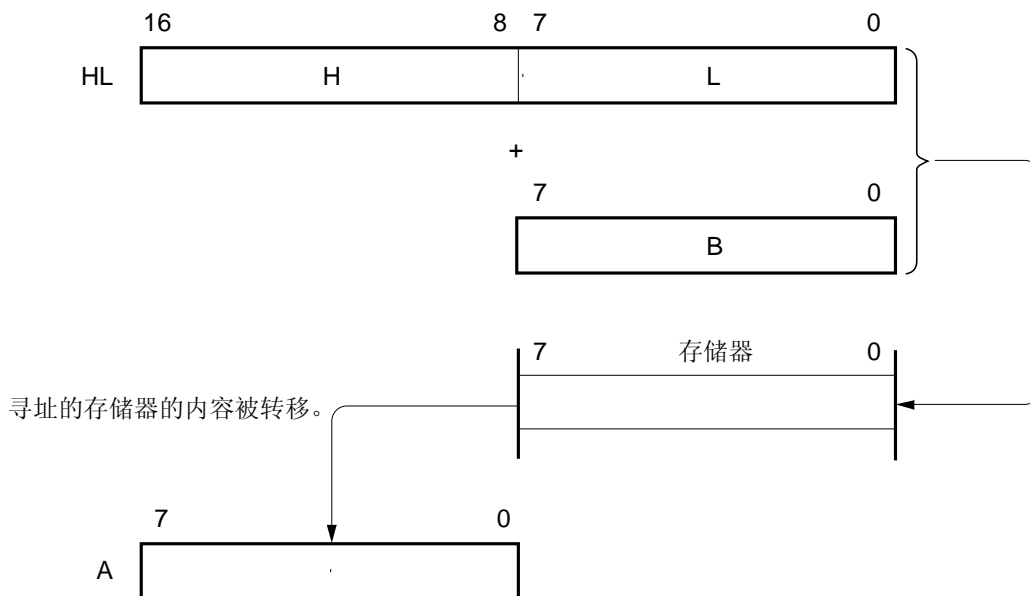
[说明示例]

MOV A, [HL + B]; 选择 B 寄存器时

操作代码

1 0 1 0 1 0 1 1

[图示]



3.4.9 堆栈寻址

[功能]

堆栈区域通过堆栈指针（SP）的内容进行间接寻址。

在执行 PUSH， POP， 子程序调用以及返回指令时， 或者在生成中断请求时对寄存器进行保存 / 复位时， 将会自动采用这种寻址方式。

使用堆栈寻址时， 只能访问内部高速 RAM 区域。

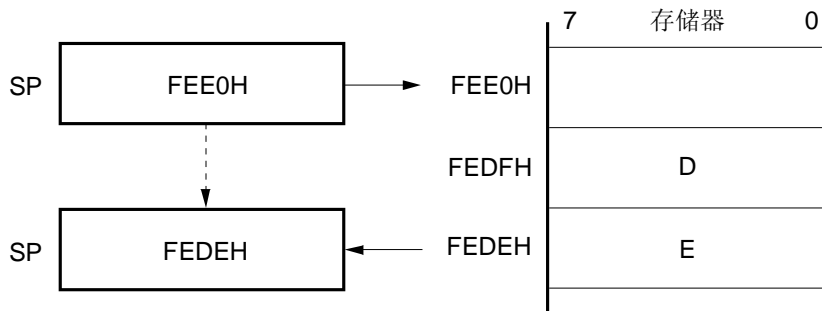
[说明示例]

PUSH DE; 保存 DE 寄存器时

操作代码

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

[图示]



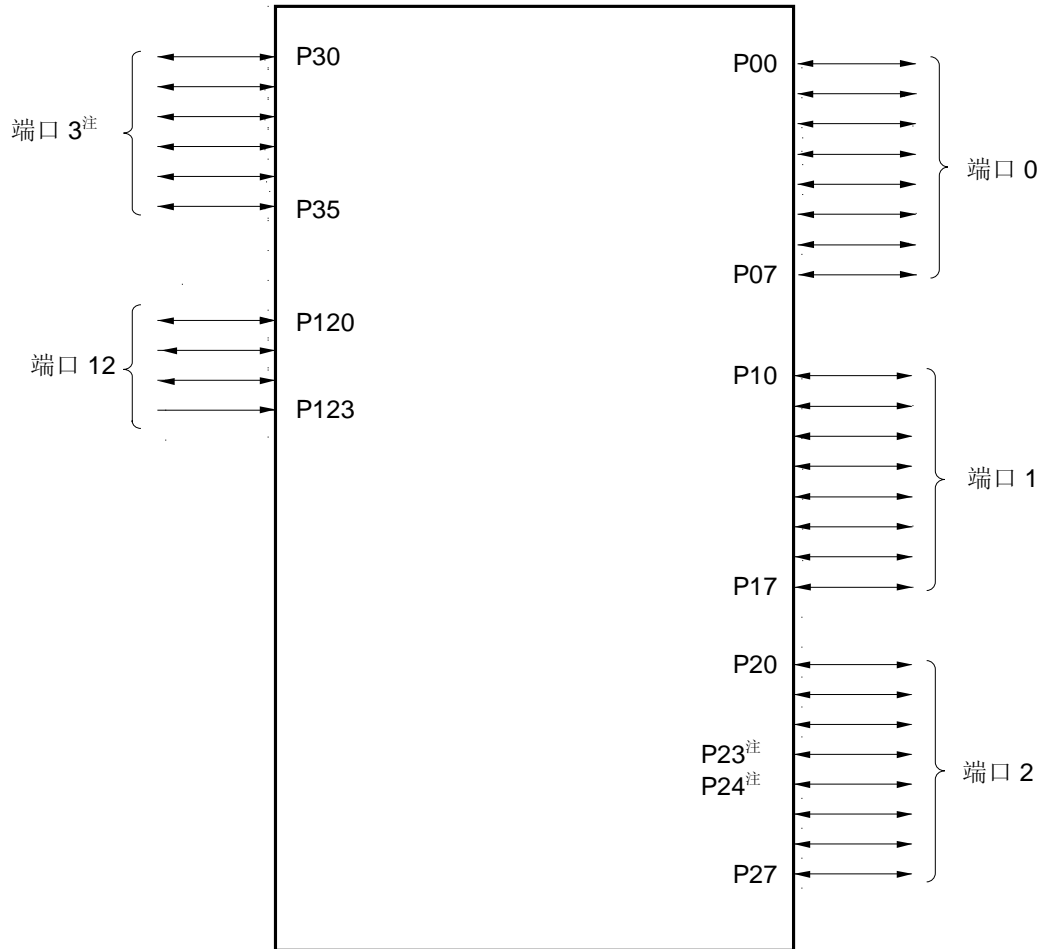
第四章 端口功能

4.1 端口功能

μ PD179F11x, 179F12x 微控制器提供的端口如图 4-1 中所示, 这些端口支持多种控制操作。各个端口的功能如表 4-1 中所示。

除了用作数字 I/O 端口外, 这些端口还具有多个复用功能。关于复用功能的详细信息, 请参见第二章 引脚功能。

图 4-1. 端口类型



注 仅限 38 引脚的产品

表 4-1. 端口功能

功能名称	I/O	功能	复位后	复用功能
P00	I/O	端口 0。 8 位 I/O 端口。 可以以 1 位为单位来指定输入 / 输出模式。 可以以 1 位为单位定义片上上拉电阻的使用。 可以以 1 位为单位将 P00 到 P06 设为 N-ch 漏极开路输出或 CMOS I/O。 可以以 1 位为单位将 P07 设为 P-ch 漏极开路输出或 CMOS I/O。通过指定 P-ch 漏极开路输出，该引脚可作用于遥控的载波发生器输出。	输入端口	TI51/TO51
P01				TOH0
P02				TI000/TxD6
P03				TI010/TO00/ RxD6
P04				INTP3/OCD1A
P05				INTP2/OCD1B
P06				TI50/TO50/INTP1
P07				REM/TOH1
P10 到 P17	I/O	端口 1。 8 位 I/O 端口。 可以以 1 位为单位来指定输入 / 输出模式。 可以以 1 位为单位定义片上上拉电阻的使用。 可以以 1 位为单位将 P10 到 P17 设为 N-ch 漏极开路输出或 CMOS I/O。	输入端口	KR0 to KR7
P20 到 P22	I/O	端口 2。 38 引脚产品：8 位 I/O 端口 30 引脚产品：6 位 I/O 端口 可以以 1 位为单位来指定输入 / 输出模式。 可以以 1 位为单位定义片上上拉电阻的使用。 可以以 1 位为单位将 P20 到 P22, P23 ^注 , P24 ^注 , P25 到 P27 设为 N-ch 漏极开路输出或 CMOS I/O。	输入端口	-
P23 ^注 , P24 ^注				-
P25, P26				-
P27				INTP4
P30 到 P35 ^注	I/O	端口 3 ^注 。 6 位 I/O 端口。 可以以 1 位为单位来指定输入 / 输出模式。 可以以 1 位为单位定义片上上拉电阻的使用。 可以以 1 位为单位将 P30 到 P35 设为 N-ch 漏极开路输出或 CMOS I/O。	输入端口	KR9 到 KR14 ^注
P120	I/O	端口 12。 3 位 I/O 端口和 1 位只输入端口。 在 P120 到 P122 引脚中，可以以 1 位为单位来指定输入 / 输出模式。 仅在 P120 和 P123 引脚中，可以指定片上上拉电阻的使用。 可以以 1 位为单位将 P120 到 P122 设为 N-ch 漏极开路输出或 CMOS I/O。	输入端口	EXLVI/INTP0
P121				X1/OCD0A
P122				X2/EXCLK/OCD0B
P123				RESET/KR8/ INTP5

注 仅限 38 引脚的产品

对于 30 引脚产品，确保设置 PM2 的第 3 位和第 4 位，PM3 的第 0 位到第 5 位为“1”，以及 P2，PU2 和 POM2 第 3 位和第 4 位，P3，PU3 和 POM3 的第 0 位到第 5 位为“0”。

4.2 端口配置

端口包含有以下硬件。

表 4-2. 端口配置

项目	配置
控制寄存器	端口模式寄存器 (PM0 到 PM2, PM3 ^注 , PM12) 端口寄存器 (P0 到 P2, P3 ^注 , P12) 上拉电阻选择寄存器 (PU0 到 PU2, PU3 ^注 , PU12) 端口输出模式寄存器 (POM0 到 POM2, POM3 ^注 , POM12) 复位引脚模式寄存器 (RSTMASK)
端口	<ul style="list-style-type: none"> • 30 引脚产品 总计: 26 (N-ch 漏极开路输出或 CMOS I/O: 24, P-ch 漏极开路输出或 CMOS I/O: 1, CMOS 输入: 1) • 38 引脚产品 总计: 34 (N-ch 漏极开路输出或 CMOS I/O: 32, P-ch 漏极开路输出或 CMOS I/O: 1, CMOS 输入: 1)
上拉电阻	<ul style="list-style-type: none"> • 30 引脚产品 总计: 24 • 38 引脚产品 总计: 32

注 仅限 38 引脚的产品

4.2.1 端口 0

端口 0 是带有输出锁存的 8 位 I/O 端口。端口 0 可以通过使用端口模式寄存器 0 (PM0) 以 1 位为单位设置为输入模式或输出模式。当 P00 到 P07 引脚用作输入端口时，片上上拉电阻可以通过上拉电阻选择寄存器 0 (PU0) 以 1 位为单位进行指定。

通过使用端口输出模式寄存器 0 (POM0)，可以以 1 位为单位设置 P00 到 P06 为 N-ch 漏极开路输出 或 CMOS I/O。

通过使用端口输出模式寄存器 0 (POM0)，可以以 1 位为单位设置 P07 为 P-ch 漏极开路输出或 CMOS I/O。通过指定 P-ch 漏极开路输出，该引脚可用作用于遥控的载波发生器输出。

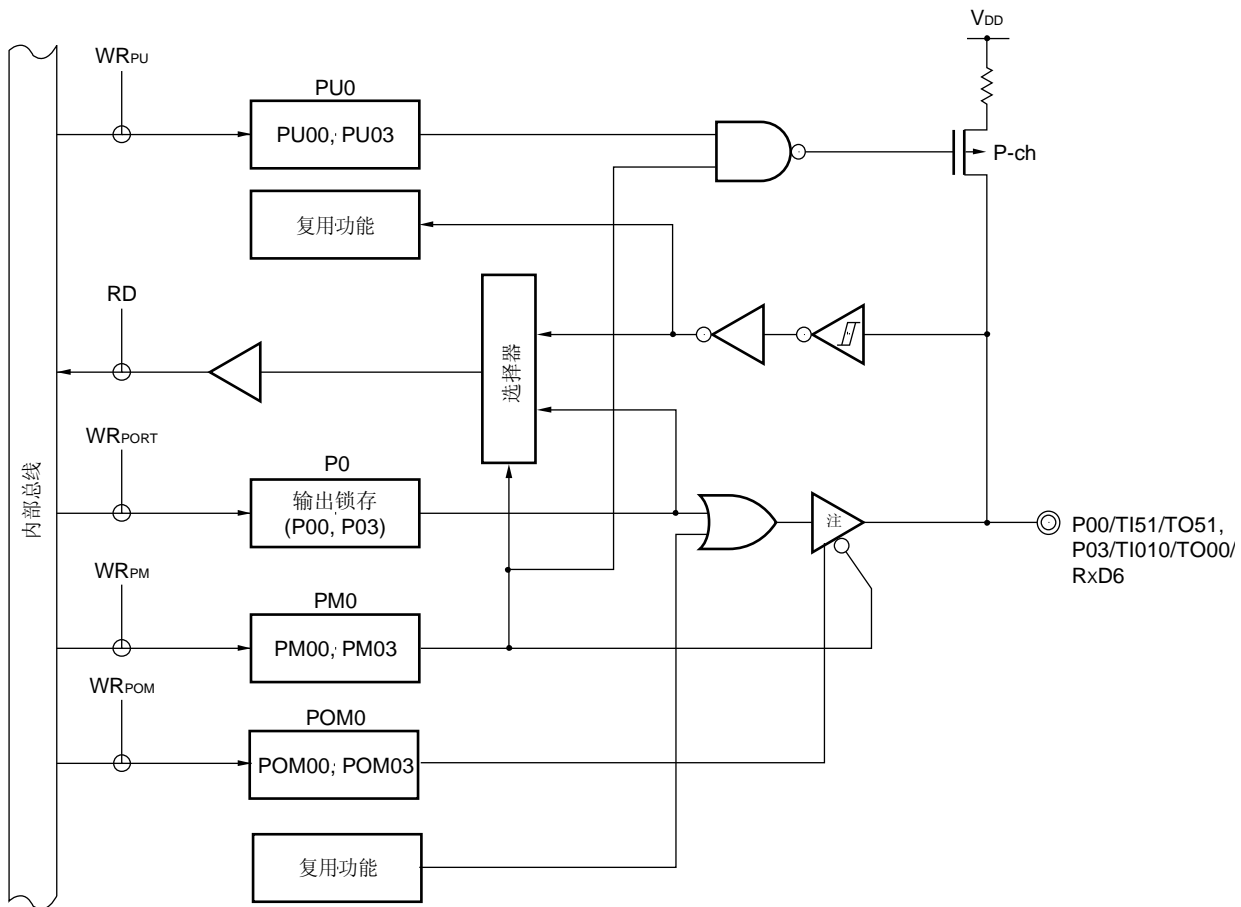
该端口也可用于定时器 I/O，串行数据 I/O，外部中断请求输入，遥控输出以及设置片上调试模式的连接。

复位信号生成将端口 0 设置为输入模式。

图 4-2 至 4-6 显示了端口 0 的框图。

<R>

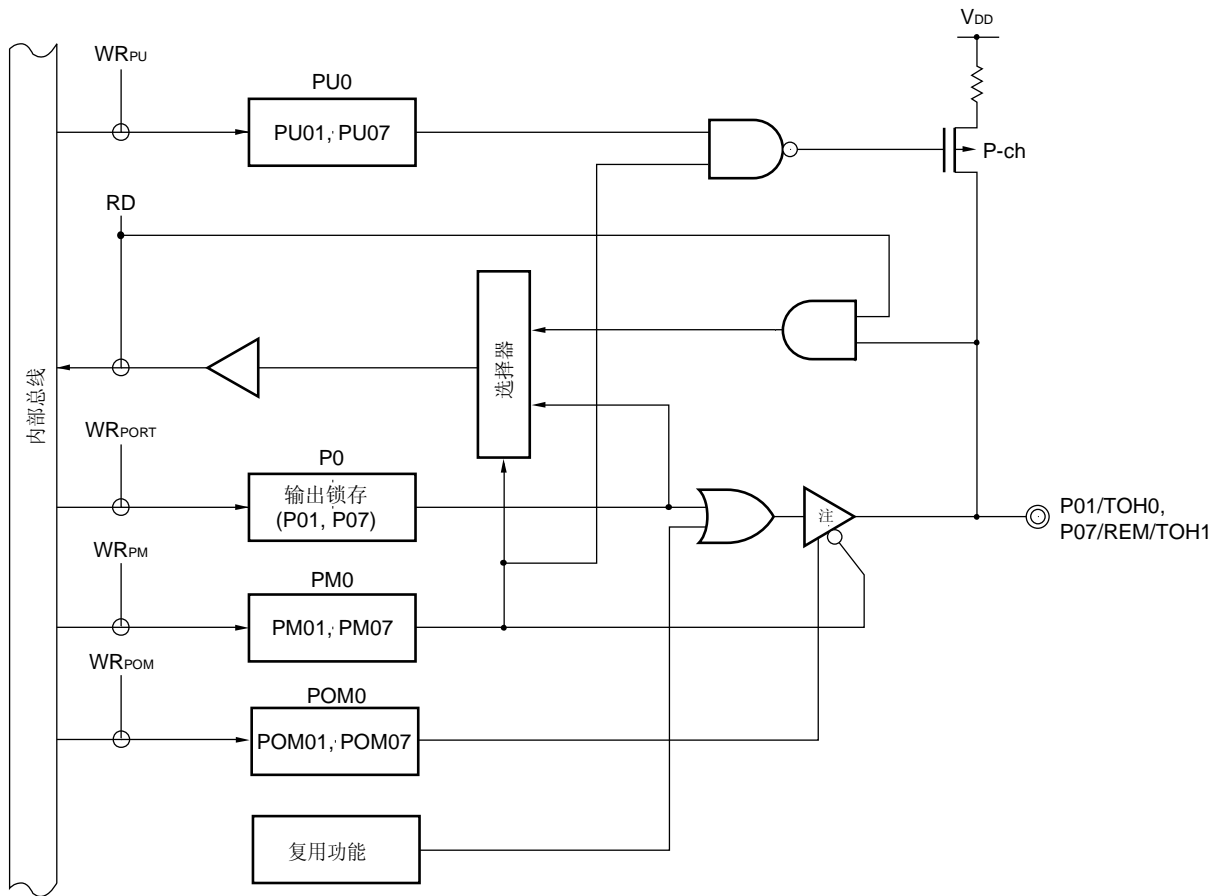
图 4-2. P00 和 P03 的框图



- P0: 端口寄存器 0
- PU0: 上拉电阻选择寄存器 0
- PM0: 端口模式寄存器 0
- POM0: 端口输出模式寄存器 0
- RD: 读取信号
- WR_{xx}: 写入信号

注 可以转换为 CMOS 输出或 N-ch 漏极开路输出的输出缓冲

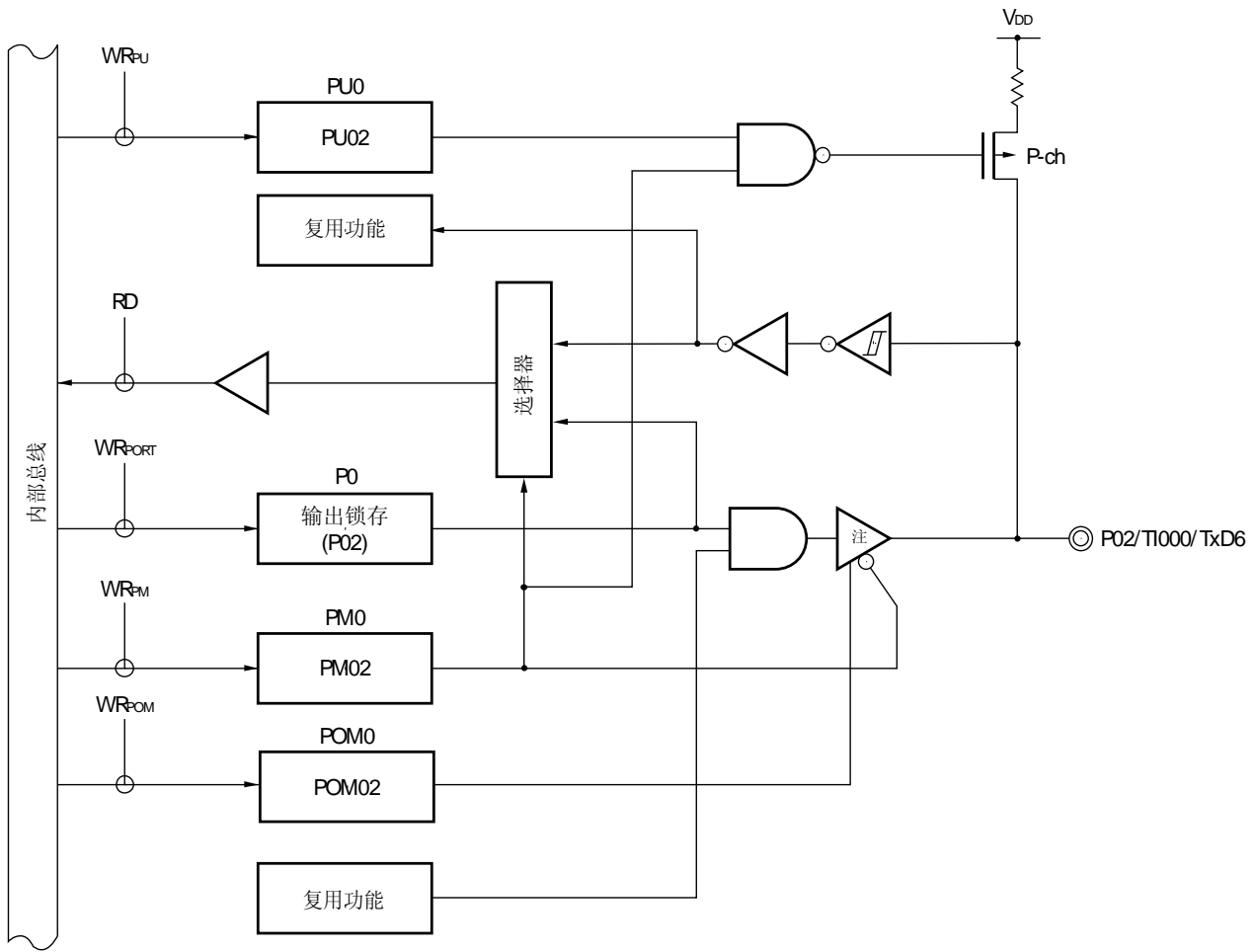
图 4-3. P01 和 P07 的框图



- P0: 端口寄存器 0
- PU0: 上拉电阻选择寄存器 0
- PM0: 端口模式寄存器 0
- POM0: 端口输出模式寄存器 0
- RD: 读取信号
- WR_{xx}: 写入信号

注 对于 P01 来说可以转换为 CMOS 输出或 N-ch 漏极开路输出的输出缓冲。
 对于 P07 来说可以转换为 CMOS 输出或 P-ch 漏极开路输出的输出缓冲。

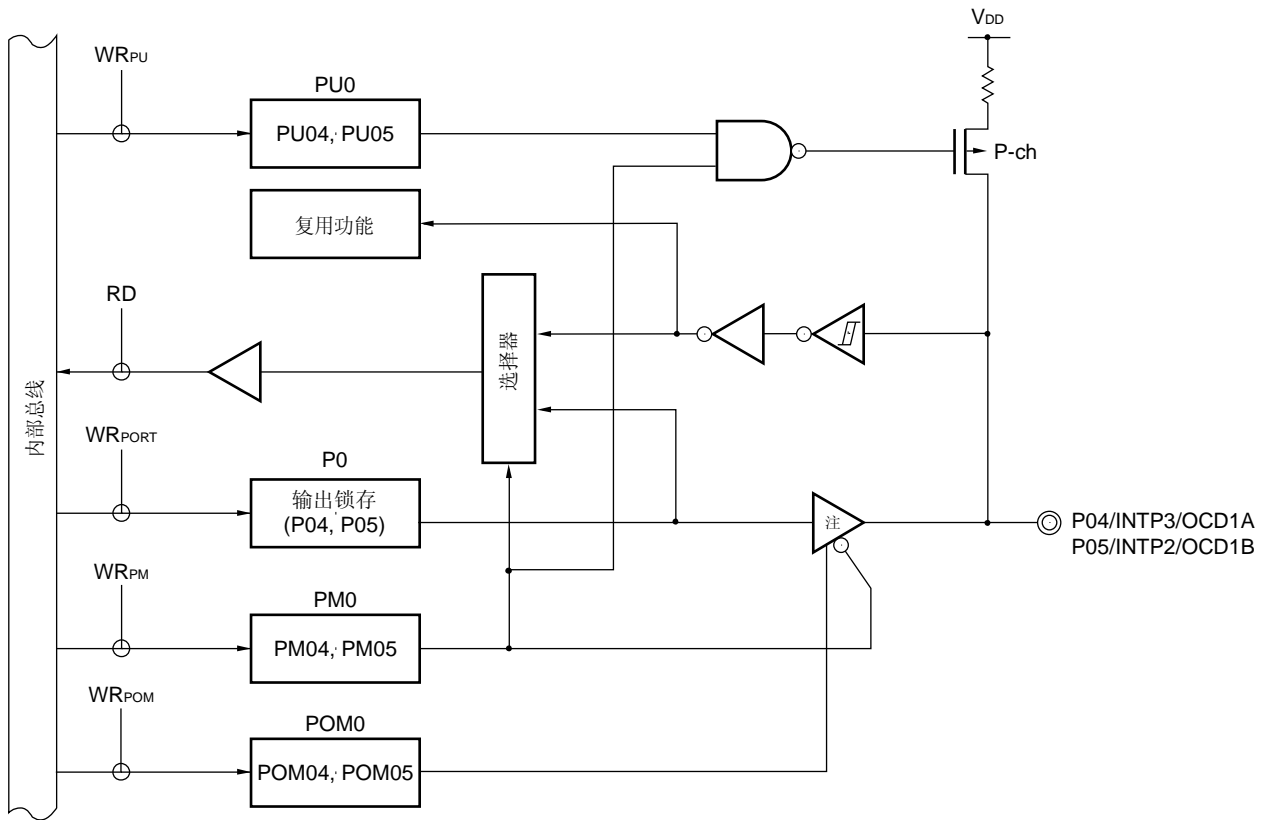
图 4-4. P02 的框图



- P0: 端口寄存器 0
- PU0: 上拉电阻选择寄存器 0
- PM0: 端口模式寄存器 0
- POM0: 端口输出模式寄存器 0
- RD: 读取信号
- WR_{xx}: 写入信号

注 可以转换为 CMOS 输出或 N-ch 漏极开路输出的输出缓冲

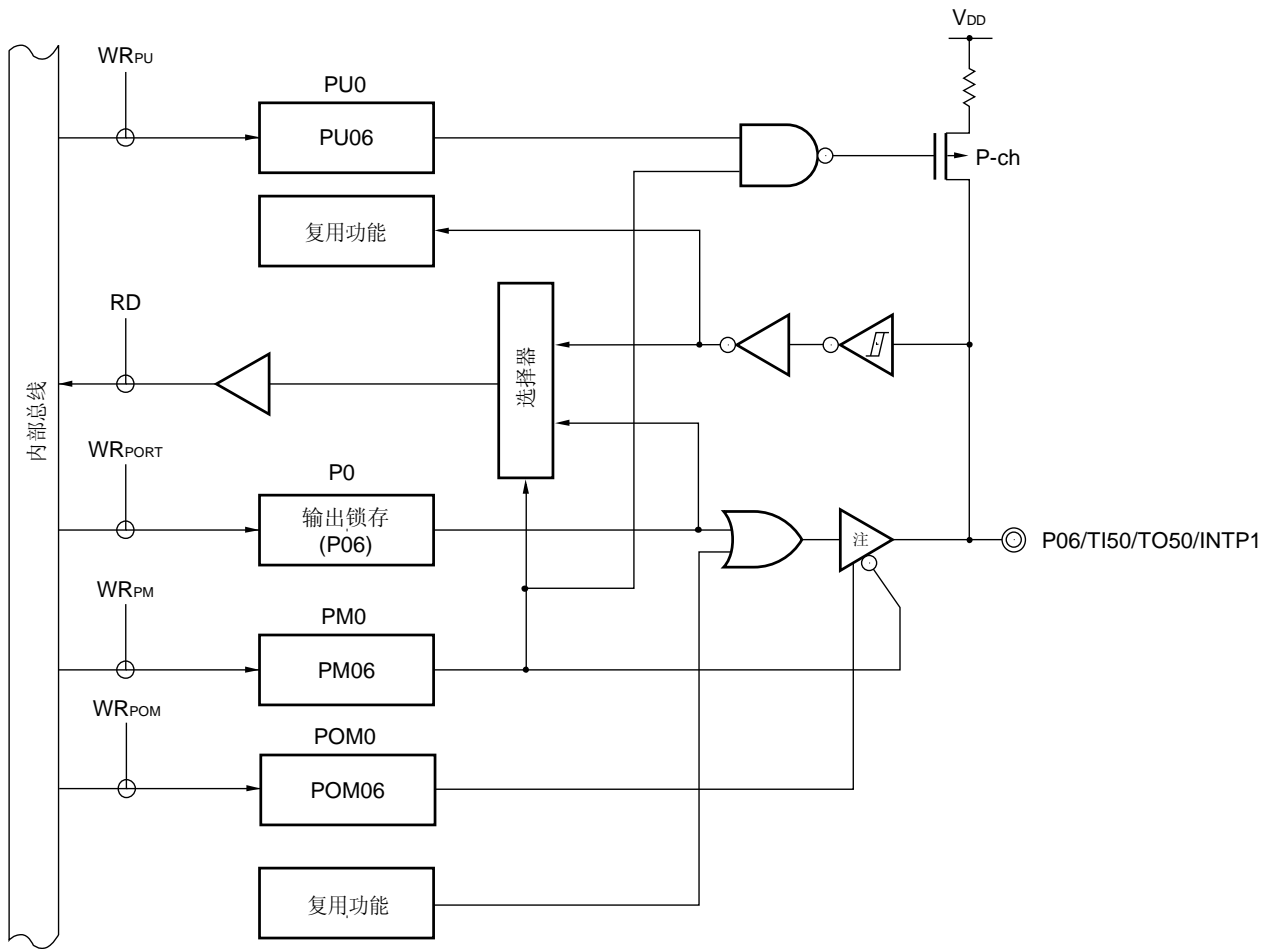
图 4-5. P04 和 P05 的框图



- P0: 端口寄存器 0
- PU0: 上拉电阻选择寄存器 0
- PM0: 端口模式寄存器 0
- POM0: 端口输出模式寄存器 0
- RD: 读取信号
- WR_{xx}: 写入信号

注 可以转换为 CMOS 输出或 N-ch 漏极开路输出的输出缓冲

图 4-6. P06 的框图



- P0: 端口寄存器 0
 PU0: 上拉电阻选择寄存器 0
 PM0: 端口模式寄存器 0
 POM0: 端口输出模式寄存器 0
 RD: 读取信号
 WR_{xx}: 写入信号

注 可以转换为 CMOS 输出或 N-ch 漏极开路输出的输出缓冲

4.2.2 端口 1

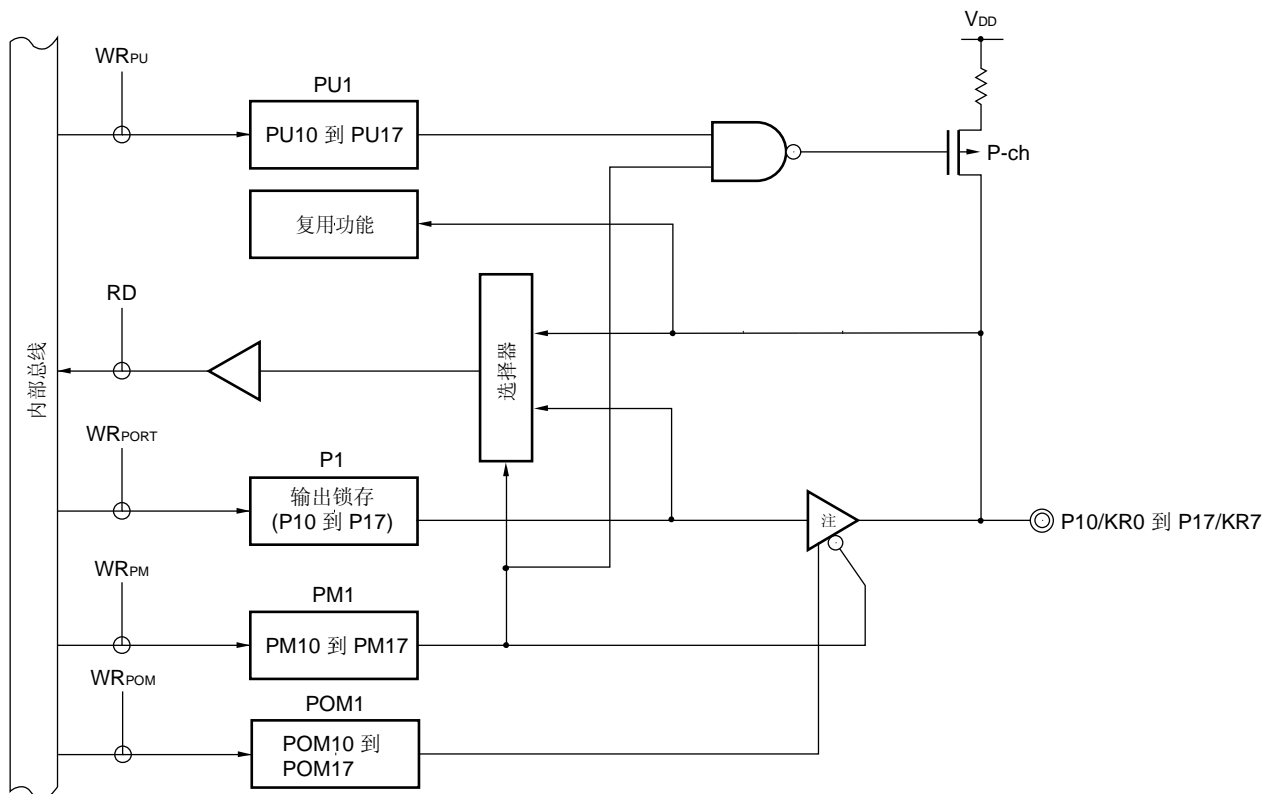
端口 1 是带有输出锁存的 8 位 I/O 端口。端口 1 可以通过使用端口模式寄存器 1 (PM1) 以 1 位为单位设置为输入模式或输出模式。当 P10 至 P17 引脚用作输入端口时，片上上拉电阻可以通过上拉电阻选择寄存器 1 (PU1) 以 1 位为单位进行指定。通过使用端口输出模式寄存器 1 (POM1)，可以以 1 位为单位设置 P10 到 P17 为 N-ch 漏极开路输出或 CMOS I/O。

该端口也可以用于按键中断输入。

复位信号生成将端口 1 设置为输入模式。

图 4-7 显示了端口 1 的框图。

图 4-7. P10 到 P17 的框图



- P1: 端口寄存器 1
- PU1: 上拉电阻选择寄存器 1
- PM1: 端口模式寄存器 1
- POM1: 端口输出模式寄存器 1
- RD: 读取信号
- WR_{xx}: 写入信号

注 可以转换为 CMOS 输出或 N-ch 漏极开路输出的输出缓冲

4.2.3 端口 2

在 30 引脚的产品中，端口 2 是带有输出锁存的 6 位 I/O 端口。在 38 引脚的产品中，端口 2 是带有输出锁存的 8 位 I/O 端口。端口 2 可以通过使用端口模式寄存器 0（PM2）以 1 位为单位设置为输入模式或输出模式。当 P20 到 P27 引脚用作输入端口时，片上上拉电阻的使用可以通过上拉电阻选择寄存器 2（PU2）以 1 位为单位进行指定。通过使用端口输出模式寄存器 2（POM2），这些引脚可以以 1 位为单位设置为 N-ch 漏极开路输出或 CMOS I/O。

只有 P27 可以用于外部中断请求输入。

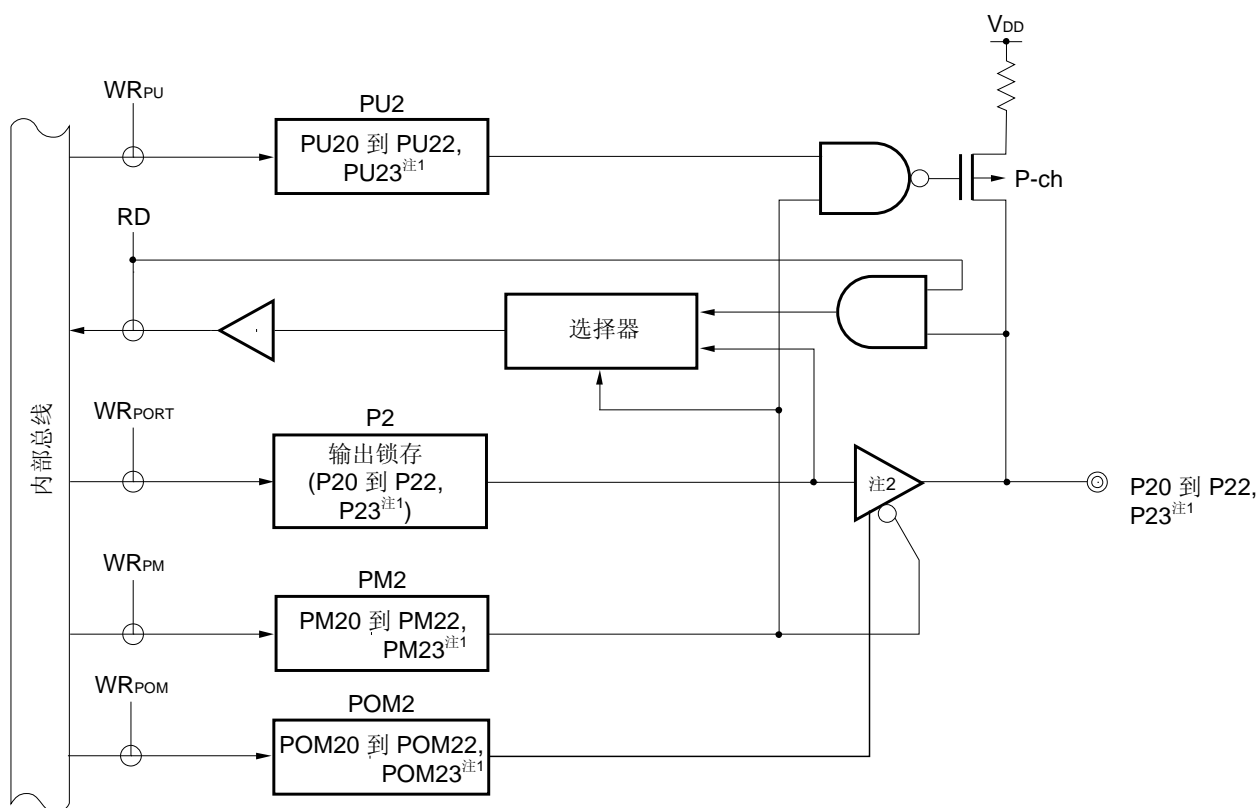
复位信号生成将端口 2 设置为输入模式。

图 4-8 至 4-10 显示了端口 2 的框图。

备注 30 引脚产品： P20 到 P22, P25, P26, P27/INTP4

38 引脚产品： P20 到 P26, P27/INTP4

图 4-8. P20 至 P23 引脚的框图



P2: 端口寄存器 2

PM2: 端口模式寄存器 2

POM2: 端口输出模式寄存器 2

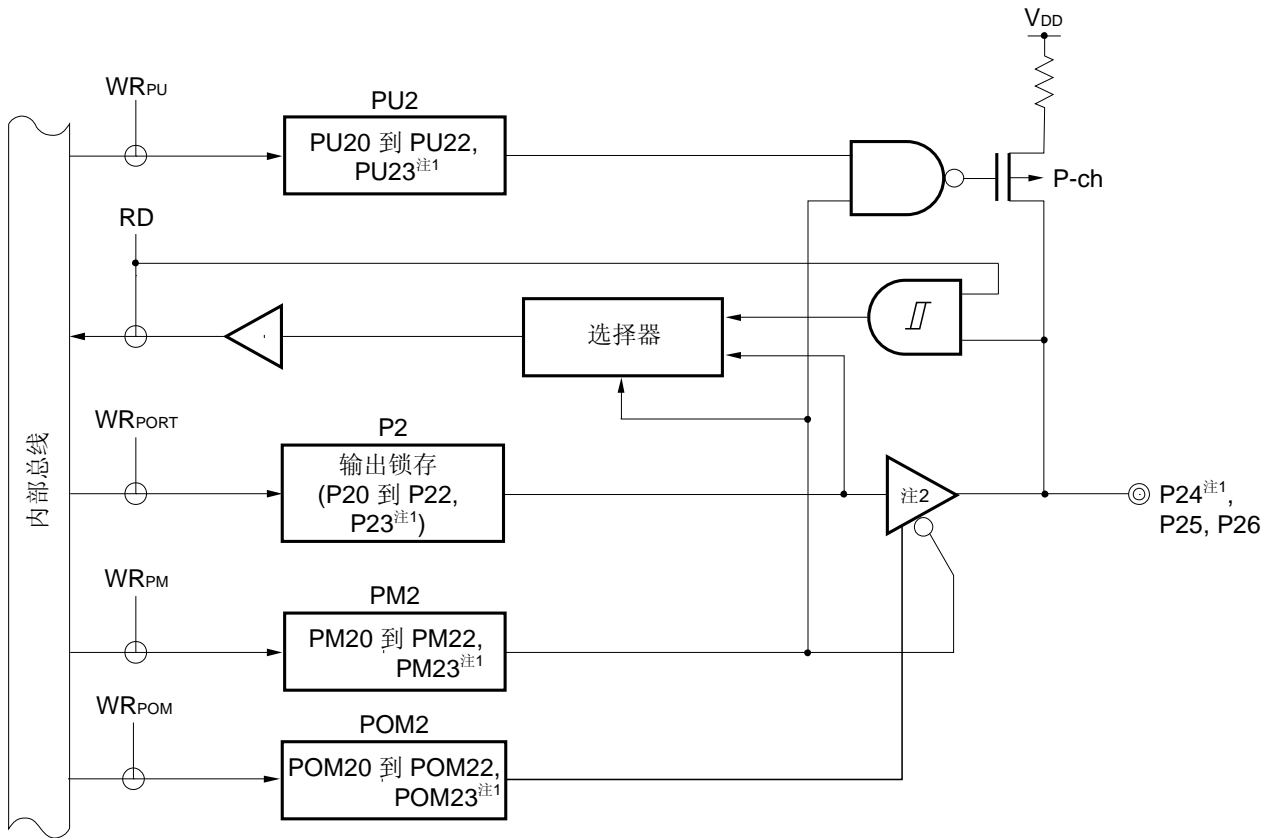
RD: 读取信号

WR_{xx}: 写入信号

注 1. 仅限 38 引脚的产品

2. 可以转换为 CMOS 输出或 N-ch 漏极开路输出的输出缓冲

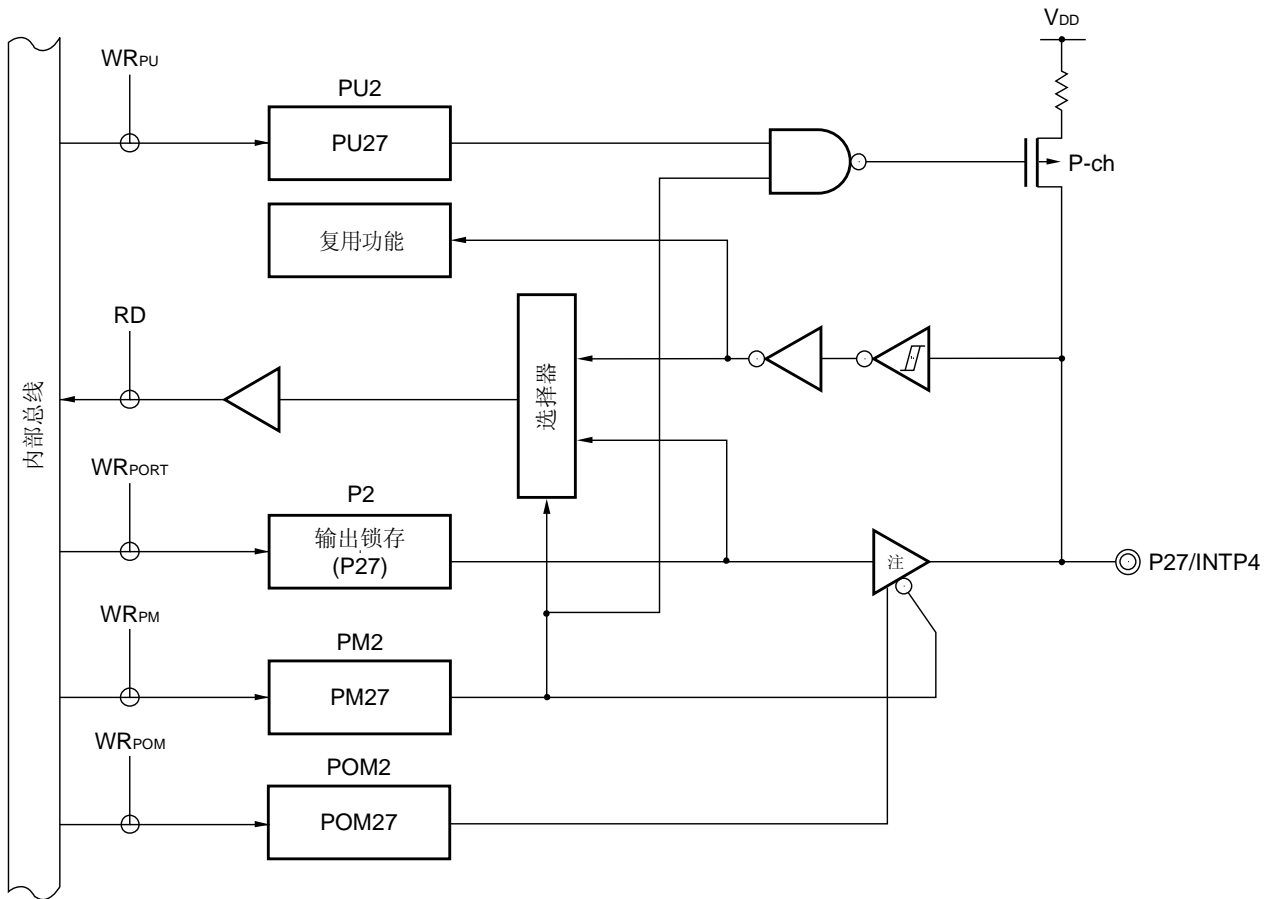
图 4-9. P24 到 P26 的框图



- P2: 端口寄存器 2
- PM2: 端口模式寄存器 2
- POM2: 端口输出模式寄存器 2
- RD: 读取信号
- WR_{xx} : 写入信号

- 注
1. 仅限 38 引脚的产品
 2. 可以转换为 CMOS 输出或 N-ch 漏极开路输出的输出缓冲

图 4-10. P27 的框图



- P2: 端口寄存器 2
- PM2: 端口模式寄存器 2
- POM2: 端口输出模式寄存器 2
- RD: 读取信号
- WR_{xx}: 写入信号

注 可以转换为 CMOS 输出或 N-ch 漏极开路输出的输出缓冲

4.2.4 端口 3（仅限 38 引脚产品）

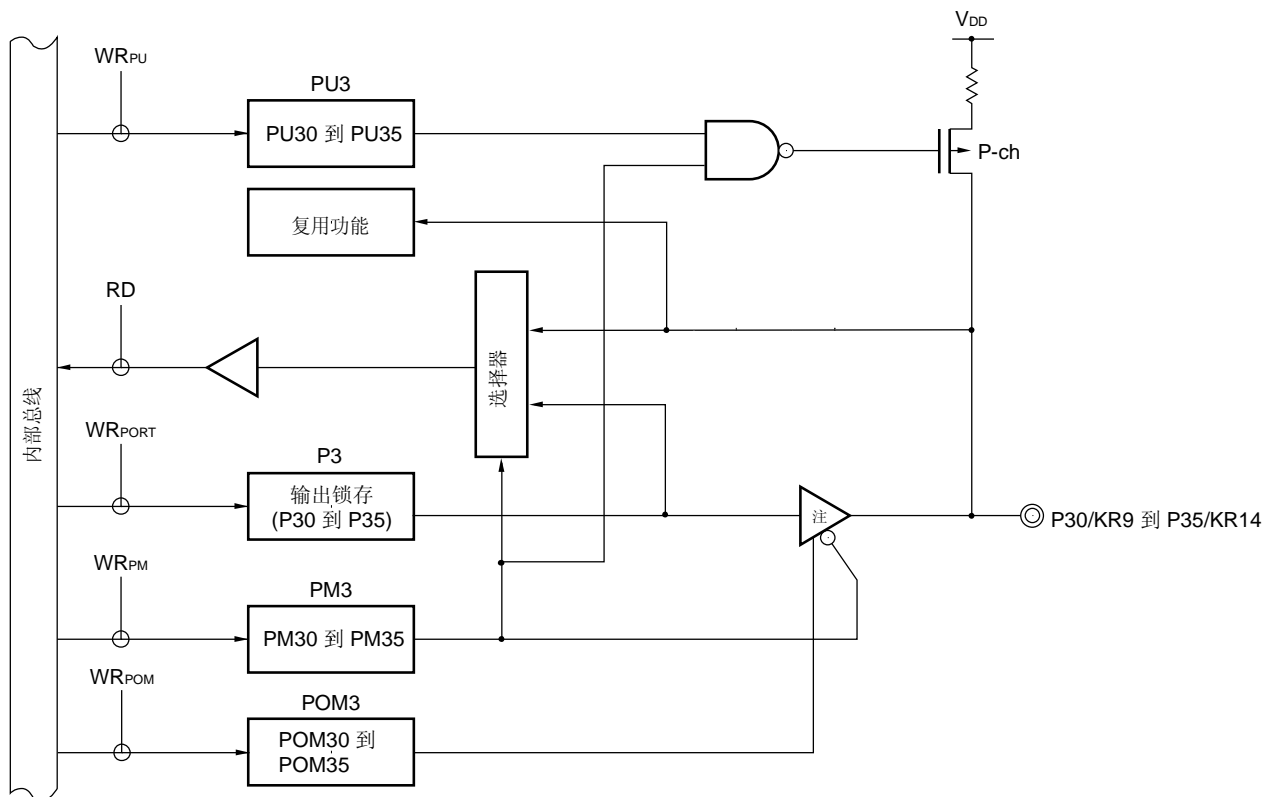
端口 3 是带有输出锁存的 6 位 I/O 端口。端口 3 可以通过使用端口模式寄存器 0（PM3）以 1 位为单位设置为输入模式或输出模式。当 P30 至 P35 引脚用作输入端口时，片上上拉电阻的使用可以通过上拉电阻选择寄存器 3（PU3）以 1 位为单位进行指定。通过使用端口输出模式寄存器 3（POM3），可以以 1 位为单位设置 P30 到 P35 为 N-ch 漏极开路输出或 CMOS I/O。

该端口也可以用于按键中断输入。

复位信号生成将端口 3 设置为输入模式。

图 4-11 显示了端口 3 的框图。

图 4-11. P30 到 P35 引脚的框图



- 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
- POM3: 端口输出模式寄存器 3
- RD: 读取信号
- WR_{xx}: 写入信号

注 可以转换为 CMOS 输出或 N-ch 漏极开路输出的输出缓冲

4.2.5 端口 12

P120 到 P122 是 3 位 I/O 端口，P123 是只输入的端口。端口 P120 到 P122 可以通过使用端口模式寄存器 12 (PM12) 以 1 位为单位设置为输入模式或输出模式。仅为 P120 和 P123 用作输入端口时，可以通过上拉电阻选择寄存器 12 (PU12) 来指定片上上拉电阻的使用。通过使用端口输出模式寄存器 12 (POM12)，可以以 1 位为单位设置 P120 到 P122 为 N-ch 漏极开路输出或 CMOS I/O。

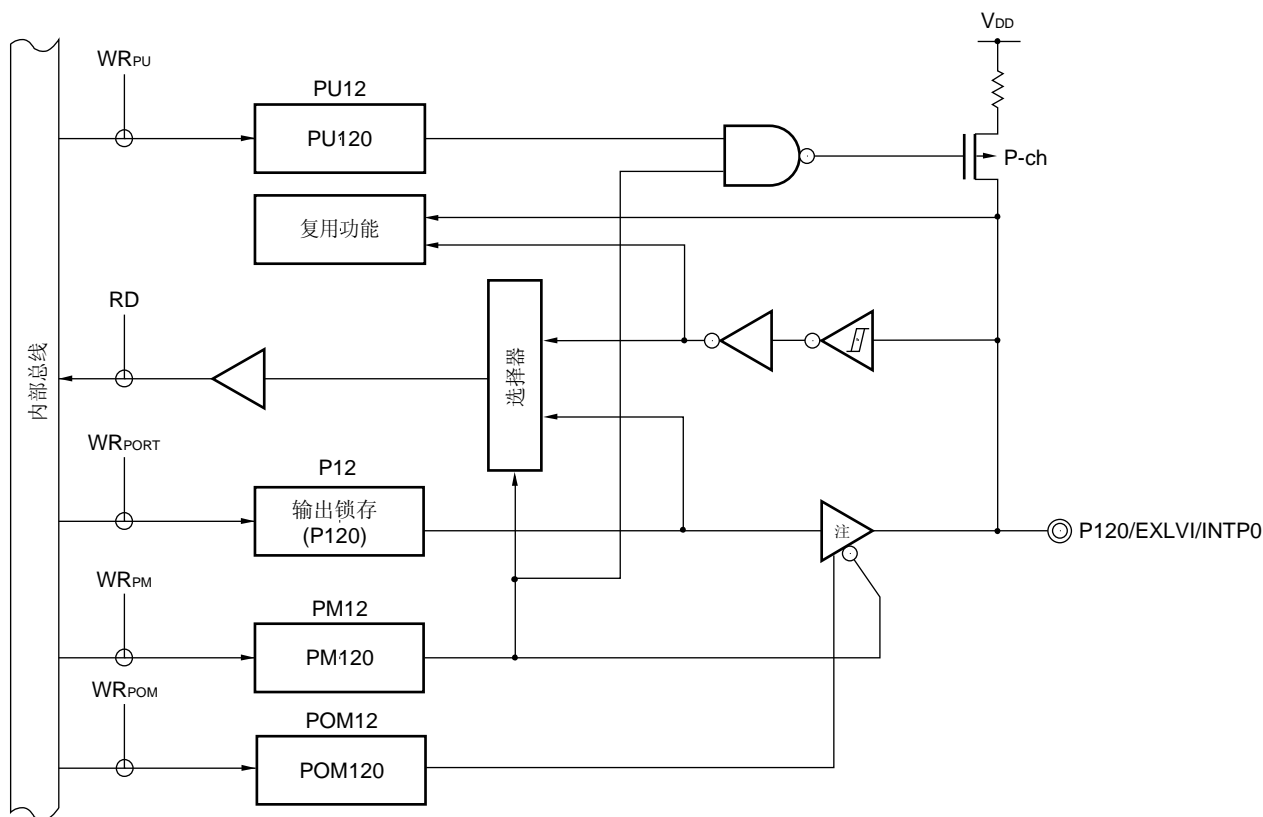
该端口也可用作外部中断请求输入，外部低压检测电压输入，为主系统时钟连接振荡器，主系统时钟的外部时钟输入，系统复位输入，按键中断输入以及为片上调试模式的设置连接的引脚。

复位信号生成将端口 12 设置为输入模式。

图 4-12 和 4-14 显示了端口 12 的框图。

注意事项 在使用 P121 和 P122 引脚连接主系统时钟 (X1, X2) 的谐振器或使用 P121 和 P122 来输入主系统时钟 (EXCLK) 的外部时钟时，必须通过使用时钟操作模式选择寄存器 (OSCCTL) 来设置 X1 振荡模式或外部时钟输入模式 (详细信息，请参见 5.3 (1) 时钟操作模式选择寄存器 (OSCCTL))。OSCCTL 的复位值为 00H (P121 和 P122 引脚都是 I/O 端口引脚)。此时，不需要对 PM121, PM122, P121, 以及 P122 引脚进行设置。

图 4-12. P120 的框图



P12: 端口寄存器 12

PU12: 上拉电阻选择寄存器 12

PM12: 端口模式寄存器 12

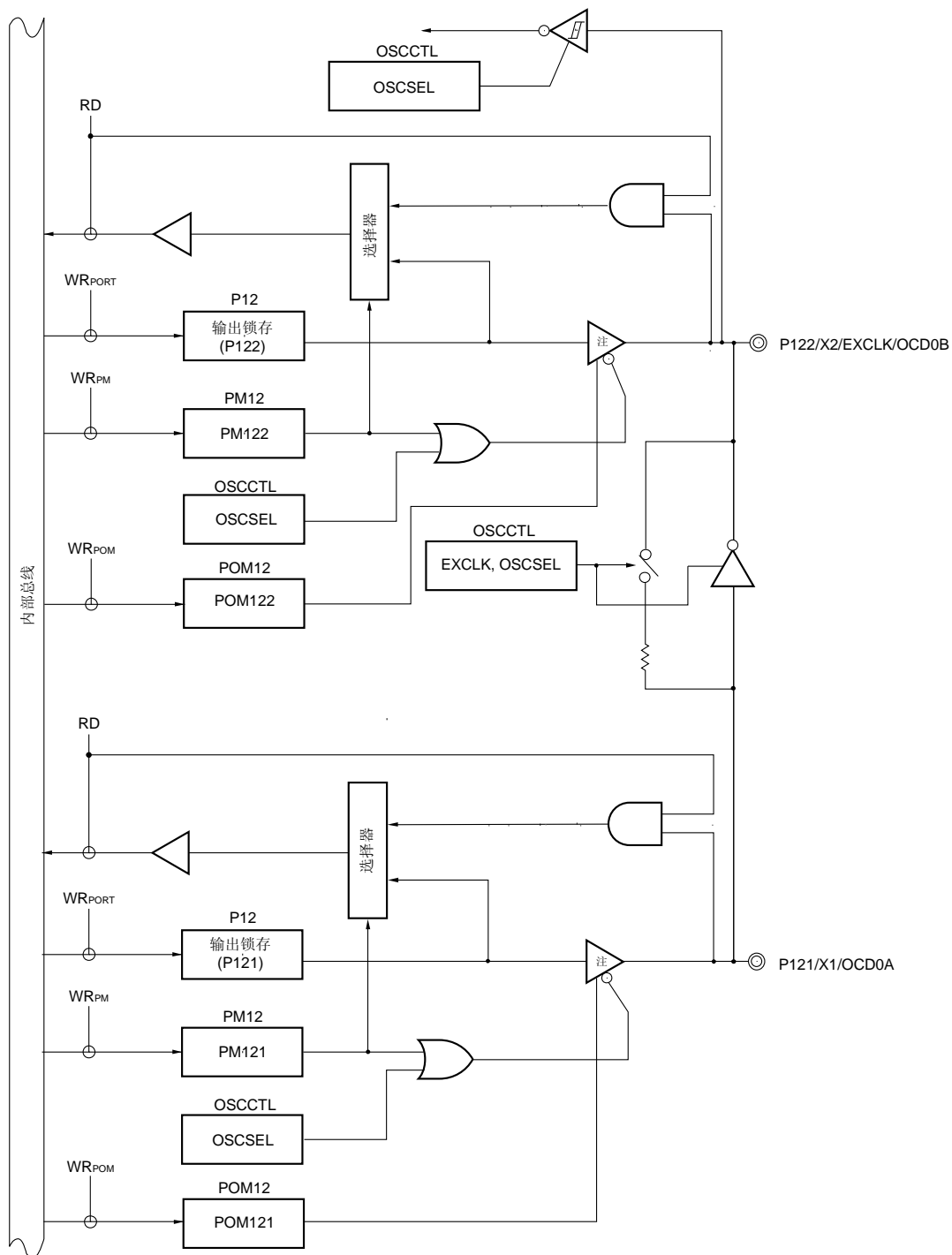
POM12: 端口输出模式寄存器 12

RD: 读取信号

WR_{xx}: 写入信号

注 可以转换为 CMOS 输出或 N-ch 漏极开路输出的输出缓冲

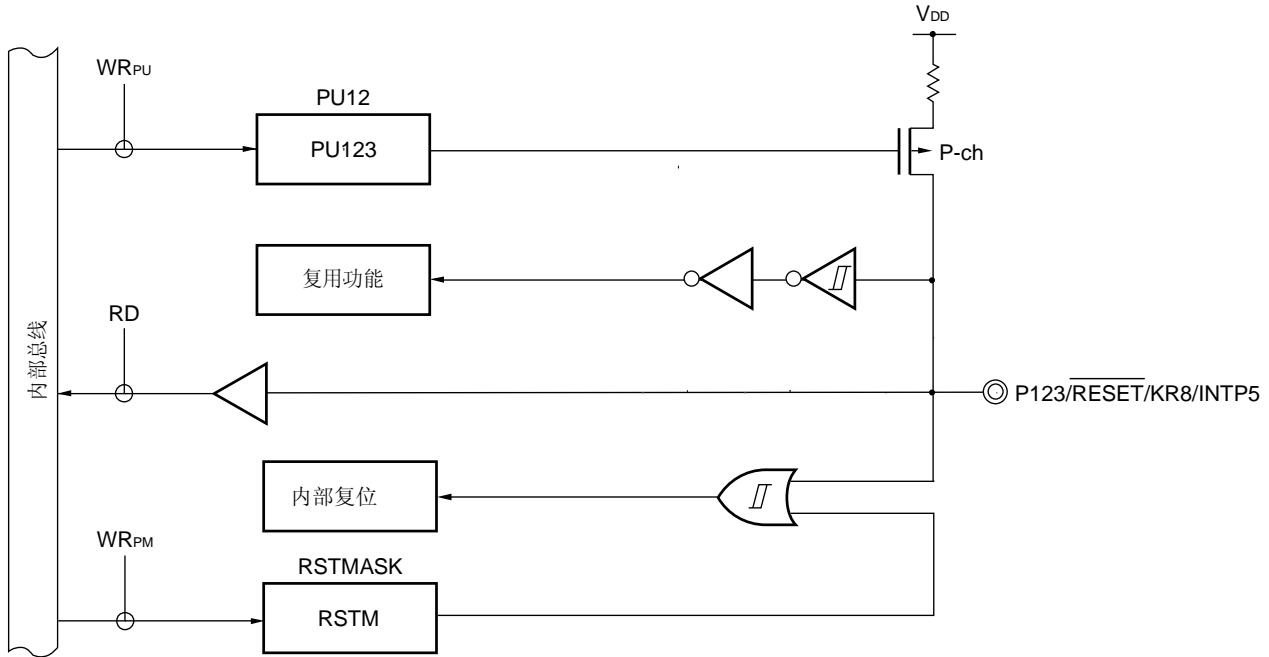
图 4-13. P121 和 P122 的框图



- P12: 端口寄存器 12
- PU12: 上拉电阻选择寄存器 12
- PM12: 端口模式寄存器 12
- OSCCTL: 时钟操作模式选择寄存器
- RD: 读取信号
- WR_{xx}: 写入信号

注 可以转换为 CMOS 输出或 N-ch 漏极开路输出的输出缓冲

图 4-14. P123 的框图



PU12: 上拉电阻选择寄存器 12

RSTMASK: 复位引脚模式寄存器

RD: 读取信号

WR_{xx}: 写入信号

备注 复位后，将可以使用外部复位功能以及上拉电阻（RSTM = 0，PU123 = 1）。
为 P123，KR8，或 INTP5 使用 P123/RESET/KR8/INTP5 引脚时，应将 RSTM 设为“1”。

4.3 控制端口功能的寄存器

端口功能由以下五种类型的寄存器进行控制。

- 端口模式寄存器（PM0 到 PM2, PM3^注, PM12）
- 端口寄存器（P0 到 P2, P3^注, P12）
- 上拉电阻选择寄存器（PU0 到 PU2, PU3^注, PU12）
- 端口输出模式寄存器（POM0 到 POM2, POM3^注, POM12）
- 复位引脚模式寄存器（RSTMASK）

注 仅限 38 引脚的产品

(1) 端口模式寄存器（PM0 到 PM2, PM3^注, PM12）

这些寄存器以 1 位为单位为端口指定输入或输出模式。

这些寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将这些寄存器设置为 FFH。

当端口引脚用作复用功能引脚时，参照 4.5 使用复用功能时的端口模式寄存器，输出锁存，上拉电阻选择寄存器以及端口输出模式寄存器的设置来设置端口模式寄存器。

注 仅限 38 引脚的产品

图 4-15. 端口模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24 ^注	PM23 ^注	PM22	PM21	PM20	FF22H	FFH	R/W
PM3 ^注	1	1	PM35 ^注	PM34 ^注	PM33 ^注	PM32 ^注	PM31 ^注	PM30 ^注	FF23H	FFH	R/W
PM12	1	1	1	1	1	PM122	PM121	PM120	FF2CH	FFH	R/W

PMmn	Pmn 引脚 I/O 模式选择 (m = 0 到 3, 12; n = 0 到 7)
0	输出模式 (输出缓冲开)
1	输入模式 (输出缓冲关)

注 仅限 38 引脚的产品

注意事项 对于 30 引脚产品，确保将 PM2 的第 3 位和第 4 位，PM3 的第 0 位到第 5 位设为“1”。

(2) 端口寄存器 (P0 到 P2, P3^注和 P12)

当数据从端口中输出时, 这些寄存器用于对芯片中输出的数据进行写操作。

如果在输入模式中读取数据, 那么将会读取引脚电平。如果在输出模式中读取, 则将会读取输出锁存的值。

这些寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将这些寄存器设置为 00H。

注 仅限 38 引脚的产品

图 4-16. 端口寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
P0	0	0	0	0	0	0	P01	P00	FF00H	00H (输出锁存)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FF01H	00H (输出锁存)	R/W
P2	P27	P26	P25	P24 ^注	P23 ^注	P22	P21	P20	FF02H	00H (输出锁存)	R/W
P3 ^注	0	0	P35 ^注	P34 ^注	P33 ^注	P32 ^注	P31 ^注	P30 ^注	FF03H	00H (输出锁存)	R/W
P12	0	0	0	0	P123	P122	P121	P120	FF0CH	00H (输出锁存)	R/W

Pmn	m = 0 到 3, 12; n = 0 到 7	
	输出数据控制 (输出模式中)	输入数据读取 (输入模式中)
0	输出 0	输入低电平
1	输出 1	输入高电平

注 仅限 38 引脚的产品

注意事项 对于 30 引脚产品, 确保将 P2 的第 3 位和第 4 位, P3 的第 0 位到第 5 位设为“0”。

(3) 上拉电阻选择寄存器 (PU0 到 PU2, PU3^注和 PU12)

这些寄存器用于指定是否使用 P00 到 P07, P10 到 P17, P20 到 P22, P23, P24^注, P25 到 P27, P30 到 P35^注, P120 以及 P123 的片上上拉电阻。片上上拉电阻只能以 1 位为单位用于那些设置到引脚输入模式的位中, 且对于这些引脚来说, 片上上拉电阻的使用已经在 PU0, PU1, PU3^注以及 PU12 中被指定。不管 PU0, PU1, PU3^注和 PU12 是何设置, 片上上拉电阻都不能被连接到设置为输出模式的位上以及用作复用功能输出引脚的位上。

这些寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。

<R> 复位信号生成将这些寄存器设置为 00H (PU12 中为 08H)。

注 仅限 38 引脚的产品

图 4-17. 上拉电阻选择寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	FF30H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W
PU2	PU27	PU26	PU25	PU24 ^注	PU23 ^注	PU22	PU21	PU20	FF32H	00H	R/W
PU3 ^注	0	0	PU35 ^注	PU34 ^注	PU33 ^注	PU32 ^注	PU31 ^注	PU30 ^注	FF33H	00H	R/W
PU12	0	0	0	0	PU123	0	0	PU120	FF3CH	08H	R/W

PUmn	Pmn 引脚片上上拉电阻选择 (m = 0 到 3, 12; n = 0 到 7)
0	未连接片上上拉电阻
1	已连接片上上拉电阻

注 仅限 38 引脚的产品

注意事项 对于 30 引脚产品, 确保将 PU2 的第 3 位和第 4 位, PU3 的第 0 位到第 5 位设为“0”。

(4) 端口输出模式寄存器 (POM0 到 POM2, POM3^注和 POM12)

这些寄存器可以设置 P00 到 P07, P10 到 P17, P20 到 P22, P23^注, P24^注, P25 到 P27, P30 到 P35^注, P120 到 P122 的输出模式。

这些寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。复位信号生成将这些寄存器设置为 00H。

注 仅限 38 引脚的产品

图 4-18. 端口输出模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
POM0	POM07	POM06	POM05	POM04	POM03	POM02	POM01	POM00	FF38H	00H	R/W
POM1	POM17	POM16	POM15	POM14	POM13	POM12	POM11	POM10	FF39H	00H	R/W
POM2	POM27	POM26	POM25	POM24 ^注	POM23 ^注	POM22	POM21	POM20	FF3AH	00H	R/W
POM3 ^注	0	0	POM35 ^注	POM34 ^注	POM33 ^注	POM32 ^注	POM31 ^注	POM30 ^注	FF3BH	00H	R/W
POM12	0	0	0	0	0	POM122	POM121	POM120	FF3EH	00H	R/W

POMmn	Pmn 引脚输出模式选择 (m = 0 到 3, 12; n = 0 到 7)
0	CMOS 输出
1	N-ch 漏极开路输出 (P07: P-ch 漏极开路输出)

注 仅限 38 引脚的产品

注意事项 对于 30 引脚产品, 确保将 POM2 的第 3 位和第 4 位, POM3 的第 0 位到第 5 位设为“0”。

(5) 复位引脚模式寄存器 (RSTMASK)

RSTMASK 是用于控制允许或禁止 $\overline{\text{RESET}}$ /P123/KR8/INTP5 引脚中除复位功能外的复用功能的寄存器。

该寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。复位信号生成将这些寄存器设置为 00H。

图 4-19. 复位引脚模式寄存器 (RSTMASK) 的格式

地址: FF2EH 复位后: 00H R/W

符号	7	6	5	4	<3>	2	1	0
RSTMASK	0	0	0	0	RSTM	0	0	0

RSTM	允许或禁止复位功能外的复用功能的控制
0	禁止复用功能。(RESET/P123/KR8/INTP5 引脚用作 $\overline{\text{RESET}}$ 引脚。)
1	允许复用功能。(RESET/P123/KR8/INTP5 引脚不用作 $\overline{\text{RESET}}$ 引脚。)

4.4 端口功能操作

使用输入模式或输出模式时的端口操作是不同的，如下所示。

4.4.1 写入I/O端口

(1) 输出模式

通过传送指令对输出锁存写入值，而输出锁存的内容则会从引脚中输出。

一旦数据被写入到输出锁存中，它将会被保留直到数据被再次写入到输出锁存中。

复位信号生成时，输出锁存的数据将会被清除。

(2) 输入模式

通过传送指令对输出锁存写入值，但由于输出缓冲处于关闭状态，因此引脚状态将不会发生变化。

一旦数据被写入到输出锁存中，它将会被保留直到数据被再次写入到输出锁存中。

复位信号生成时，输出锁存的数据将会被清除。

4.4.2 从I/O端口中读取

(1) 输出模式

通过传送指令读取输出锁存的内容。输出锁存的内容不会发生变化。

(2) 输入模式

通过传送指令读取引脚状态。输出锁存的内容不会发生变化。

4.4.3 I/O端口上的操作

(1) 输出模式

操作是在输出锁存的内容上进行的，且操作结果会被写入到输出锁存中。输出锁存的内容为引脚输出。

一旦数据被写入到输出锁存中，它将会被保留直到数据被再次写入到输出锁存中。

复位信号生成时，输出锁存的数据将会被清除。

(2) 输入模式

引脚电平被读取且操作会在其内容上进行。操作结果被写入到输出锁存中，但由于输出缓冲处于关闭状态，因此引脚状态将不会发生变化。

复位信号生成时，输出锁存的数据将会被清除。

<R> 4.5 使用复用功能时端口模式寄存器，输出锁存，上拉电阻选择寄存器以及端口输出模式寄存器的设置

若要使用端口引脚的复用功能，则应按表 4-3 所示对端口模式寄存器，输出锁存，上拉电阻选择寄存器以及端口输出模式寄存器进行设置。

表 4-3. 使用复用功能时端口模式寄存器，输出锁存，上拉电阻选择寄存器以及端口输出模式寄存器的设置

引脚名称	复用功能		PM _{xx}	P _{xx}	PU _{xx}	POM _{xx}
	功能名称	I/O				
P00	TI51	输入	1	×	×	×
	TO51	输出	0	0	×	×
P01	TOH0	输出	0	0	×	×
P02	TI000	输入	1	×	×	×
	TXD6	输出	0	1	×	×
P03	TI010	输入	1	×	×	×
	TO00	输出	0	0	×	×
	RXD6	输入	1	×	×	×
P04	INTP3	输入	1	×	×	×
P05	INTP2	输入	1	×	×	×
P06	TI50	输入	1	×	×	×
	TO50	输出	0	0	×	×
	INTP1	输入	1	×	×	×
P07	REM	输出	0	0	×	1
	TOH1	输出	0	0	×	0
P10 到 P17	KR0 to KR7	输入	1	×	1	×
P27	INTP4	输入	1	×	×	×
P30 到 P35 ^注	KR9 到 KR14 ^注	输入	1	×	1	×
P120	EXLVI	输入	1	×	×	×
	INTP0	输入	1	×	×	×
P121	X1	-	×	×	0	×
P122	X2	-	×	×	0	×
	EXCLK	输入	1	×	×	×
P123	RESET	输入	-	×	×	×
	KR8	输入	-	×	1	×
	INTP5	输入	-	×	×	×

注 仅限 38 引脚的产品

备注 1. ×: 无关的

PM_{xx}: 端口模式寄存器

P_{xx}: 端口输出锁存

PU_{xx}: 上拉电阻选择寄存器

POM_{xx}: 端口输出模式寄存器

2. 当片上调试功能被使用时，P04/INTP3，P05/INTP2，P121/X1 以及 P122/X2/EXCLK 引脚可以用作片上调试模式设置引脚（OCD1A，OCD1B，OCD0A，OCD0B）。关于连接带有编程功能（QB-MINI2）的片上调试仿真器的方法，请参见第十九章 片上调试功能。

4.6 端口寄存器 n (Pn) 的 1 位操作指令的注意事项

当一个 1 位的操作指令在一个同时提供输入和输出功能的端口上执行时，除了目标位以外，不属于操作对象的输入端口的输出锁存的值也可能被写入。

因此，当端口从输入模式切换到输出模式时，推荐重写输出锁存。

<例> 当 P10 引脚为输出端口，P11 至 P17 引脚为输入端口（所有的引脚状态均为高电平），且端口锁存的值为 00H 时，如果通过 1 位操作指令将输出端口 P10 的输出从低电平变为高电平，则端口锁存的值为 FFH。

说明：PMnm 位 = 1 的端口中 Pn 寄存器的写 / 读对象是各自的输出锁存和引脚状态。
 μ PD179F11x, 179F12x 微控制器中，1 位操作指令按照以下顺序执行。

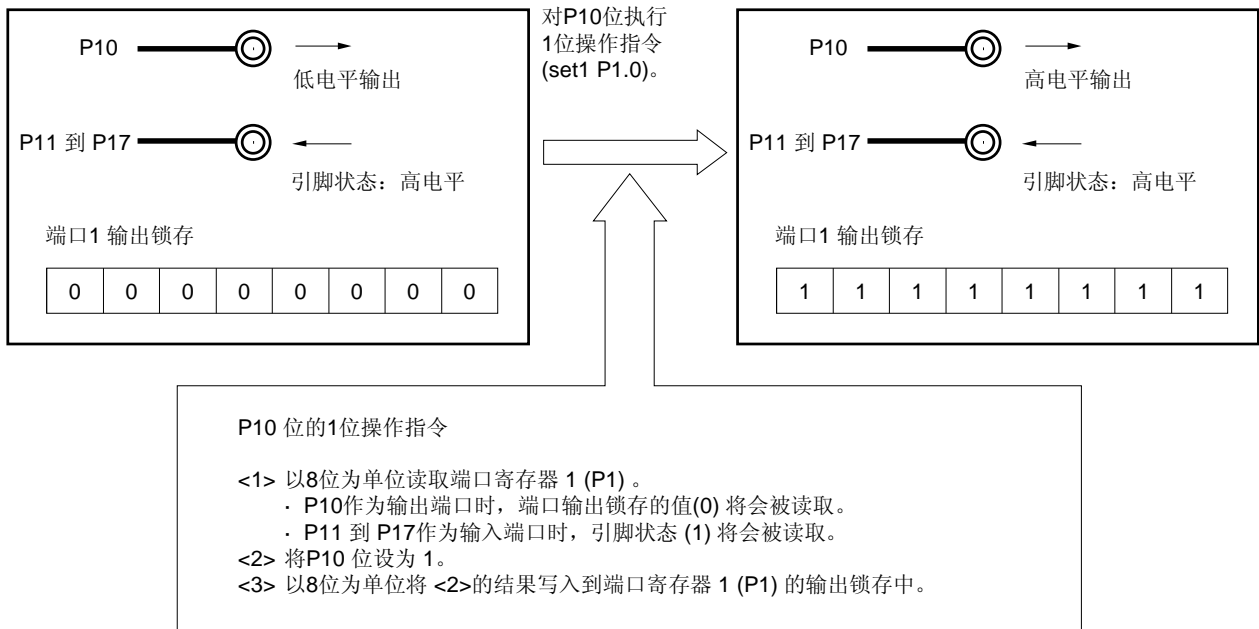
- <1> 以 8 位为单位读取 Pn 寄存器。
- <2> 操作目标位。
- <3> 以 8 位为单位写入 Pn 寄存器。

在步骤 <1>中，当作为输入端口的 P11 至 P17 引脚状态被读取时，作为输出端口的 P10 引脚的输出锁存值 (0) 被读取。如果此时 P11 至 P17 的状态为高电平，则读取的值为 FEH。

通过步骤 <2>中的操作，此值变为 FFH。

通过步骤 <3>中的操作，FFH 被写到输出锁存中。

图 4-20. 位操作指令 (P10)



5.1 时钟发生器的功能

时钟发生器用于生成提供给 CPU 以及周边硬件的时钟。

以下三种系统时钟以及时钟振荡器可供选择。

(1) 主系统时钟

<1> X1 振荡器

该电路通过将谐振器连接到 X1 和 X2 上来提供一个 $f_x = 1$ 至 4 MHz 的时钟。

通过执行 STOP 指令或使用主 OSC 控制寄存器 (MOC) 可以停止振动。

<2> 内部高速振荡器

该电路用于提供一个 $f_{RH} = 4 \text{ MHz} \pm 2\%$ 的时钟。复位释放后, CPU 通常使用该内部高速振荡时钟来开始操作。通过执行 STOP 指令或使用主振荡模式寄存器 (RCM) 可以停止振动。

外部主系统时钟 ($f_{EXCLK} = 1$ 至 4 MHz) 也可以从 EXCLK/X2/P122 引脚中来提供。通过执行 STOP 指令或使用 RCM 可以禁止外部主系统时钟输入。

作为主系统时钟时, 可以通过使用主时钟模式寄存器 (MCM) 来选择高速系统时钟 (X1 时钟或外部主系统时钟) 或内部高速振荡时钟。

(2) 内部低速振荡时钟 (看门狗定时器的时钟)

• 内部低速振荡器

<R> 该电路用于振动一个 $f_{RL} = 240 \text{ kHz}$ (TYP.) 的时钟。通常在复位释放后, 内部低速振荡时钟会开始操作。

当通过选项字节设置为“内部低速振荡器可以通过软件停止”时, 振荡可以通过使用内部振荡模式寄存器 (RCM) 来停止。

内部低速振荡时钟不能被用作 CPU 时钟。以下硬件通过内部低速振荡时钟进行操作。

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

备注	1. f_x :	X1 时钟振动频率
	2. f_{RH} :	内部高速振荡时钟频率
	3. f_{EXCLK} :	外部主系统时钟频率
	4. f_{RL} :	内部低速振荡时钟频率

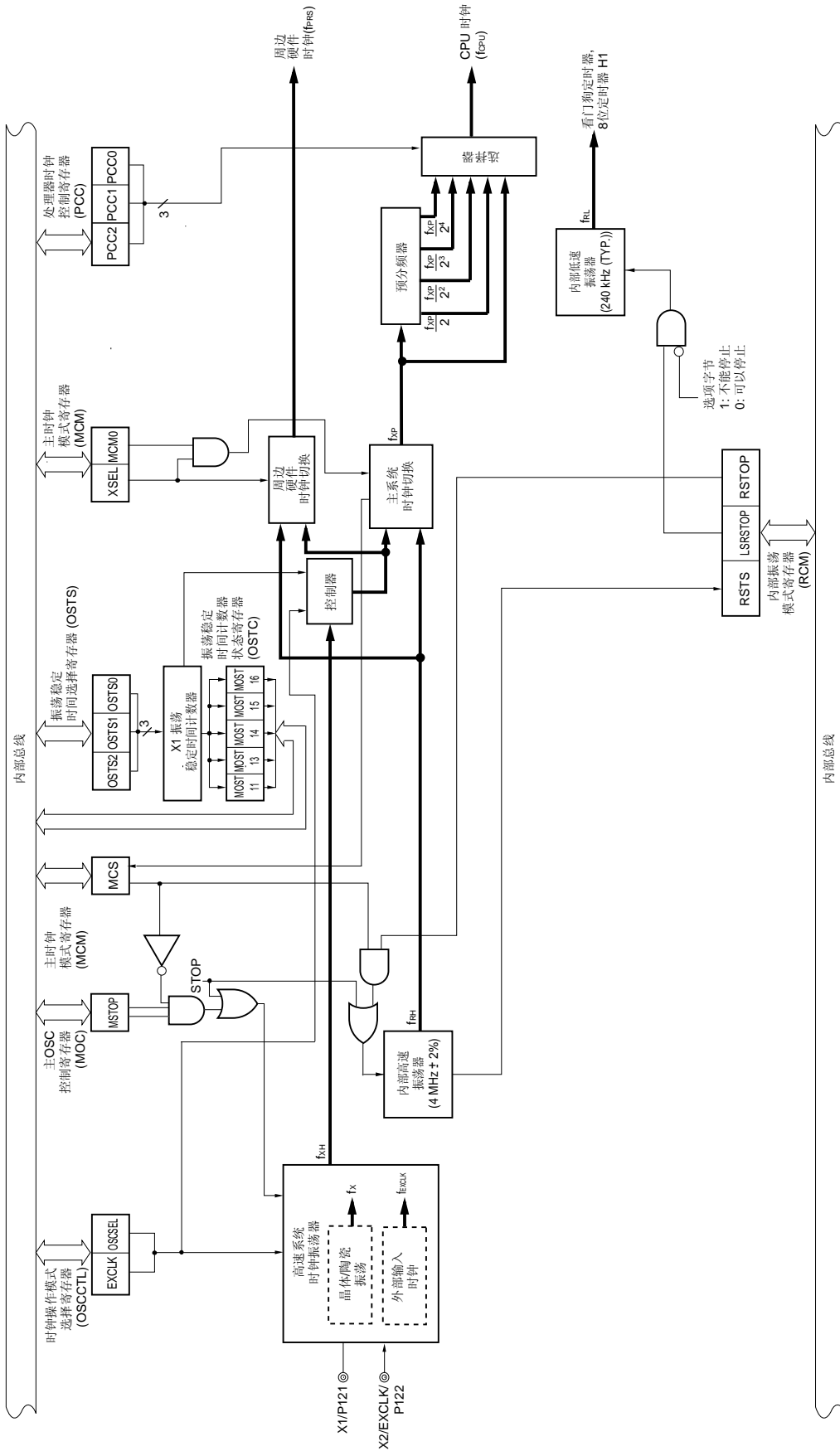
5.2 时钟发生器的配置

时钟发生器包含以下硬件。

表 5-1. 时钟发生器的配置

项目	配置
控制寄存器	时钟操作模式选择寄存器 (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)
振荡器	X1 振荡器 内部高速振荡器 内部低速振荡器

图 5-1. 时钟发生器的框图



备注	1. f_x :	X1 时钟振动频率
	2. f_{RH} :	内部高速振荡时钟频率
	3. f_{EXCLK} :	外部主系统时钟频率
	4. f_{XH} :	高速系统时钟频率
	5. f_{XP} :	主系统时钟频率
	6. f_{PRS} :	周边硬件时钟频率
	7. f_{CPU} :	CPU 时钟频率
	8. f_{RL} :	内部低速振荡时钟频率

5.3 控制时钟发生器的寄存器

以下七种寄存器是用于控制时钟发生器的。

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

(1) 时钟操作模式选择寄存器 (OSCCTL)

该寄存器用于选择高速系统的操作模式以及片上振荡器的增益。

OSCCTL 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器清除为 00H。

图 5-2. 时钟操作模式选择寄存器 (OSCCTL) 的格式

地址: FF9FH 复位后: 00H R/W

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

EXCLK	OSCSEL	高速系统时钟引脚操作模式	P121/X1 引脚	P122/X2/EXCLK 引脚
0	0	I/O 端口模式	I/O 端口	
0	1	X1 振荡模式	晶体 / 陶瓷谐振器连接	
1	0	I/O 端口模式	I/O 端口	
1	1	外部时钟输入模式	I/O 端口	外部时钟输入

<R> **注意事项** 若要更改 EXCLK 和 OSCSEL 的值, 请务必确认主 OSC 控制寄存器 (MOC) 的第 7 位 (MSTOP) 为 1 (停止 X1 振荡器或禁止 EXCLK 引脚的外部时钟)。

备注 f_{xH}: 高速系统时钟频率

(2) 处理器时钟控制寄存器 (PCC)

该寄存器用于选择 CPU 时钟和分频比率。

PCC 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将 PCC 设为 01H。

图 5-3. 处理器时钟控制寄存器 (PCC) 的格式

地址: FFFBH 复位后: 01H R/W[※]

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

PCC2	PCC1	PCC0	CPU 时钟 (f _{CPU}) 选择	
			FLMD0 = 0	FLMD0 = 1
0	0	0	f _{XP}	f _{RH}
0	0	1	f _{XP} /2 (缺省)	f _{RH} /2 (缺省)
0	1	0	f _{XP} /2 ²	f _{RH} /2 ²
0	1	1	f _{XP} /2 ³	f _{RH} /2 ³
1	0	0	f _{XP} /2 ⁴	f _{RH} /2 ⁴
其他情况			禁止设置	

注 第 5 位是只读的。

注意事项 确保将第 3 位至第 7 位清除为“0”。

备注 f_{XP}: 主系统时钟振荡频率

μPD179F11x, 179F12x 微控制器中, 执行速度最快的指令执行时间在 2 个 CPU 时钟以内。因此, 表 5-2 显示了 CPU 时钟 (f_{CPU}) 与最短指令执行时间之间的关系。

表 5-2. CPU 时钟与最短指令执行时间之间的关系

CPU 时钟 (f _{CPU})	最短指定执行时间: 2/f _{CPU}	
	主系统时钟	
	高速系统时钟 ^{※1}	内部高速振荡时钟 ^{※1}
	4 MHz 工作频率	4 MHz ± 2% 工作频率
f _{XP}	0.5 μs ^{※2}	0.5 μs (TYP.) ^{※2}
f _{XP} /2	1 μs	1 μs (TYP.)
f _{XP} /2 ²	2 μs	2 μs (TYP.)
f _{XP} /2 ³	4 μs	4 μs (TYP.)
f _{XP} /2 ⁴	8 μs	8 μs (TYP.)

- 注**
1. 主时钟模式寄存器 (MCM) 用于设置提供给 CPU 时钟 (高速系统时钟 / 内部高速振荡时钟) 的主系统时钟。(参见图 5-6)
 2. 只有 V_{DD} = 2.0 到 3.6 V 时是可设置的。

(3) 内部振荡模式寄存器 (RCM)

该寄存器用于设置内部振荡器的操作模式。

RCM 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器设置为 80H^{#1}。

图 5-4. 内部振荡模式寄存器 (RCM) 的格式

地址: FFA0H 复位后: 80H^{#1} R/W^{#2}

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

RSTS	内部高速振荡器的状态
0	等待内部高速振荡器的准确稳定
1	内部高速振荡器的稳定性操作

LSRSTOP	内部低速振荡器振荡 / 停止
0	内部低速振荡器振荡
1	内部低速振荡器停止振荡

RSTOP	内部高速振荡器振荡 / 停止振荡
0	内部高速振荡器振荡
1	内部高速振荡器停止振荡

- 注
1. 复位释放后该寄存器的值会立即变为 00H，但在内部高速振荡器稳定后寄存器的值将会自动变为 80H。
 2. 第 7 位是只读的。

注意事项 当 RSTOP 被设为 1 时，请务必确认 CPU 是使用除内部高速振荡时钟外的时钟在进行操作。在以下任一条件下进行特定地设置。

- MCS = 1 时 (CPU 使用高速系统时钟进行操作)

另外，应在将 RSTOP 设为 1 前停止在内部高速振荡时钟上运行的周边硬件。

(4) 主 OSC 控制寄存器 (MOC)

该寄存器用于选择高速系统时钟的操作模式。

该寄存器用于在 CPU 使用高速系统时钟外的时钟进行操作时，停止 X1 振荡器或禁止从 EXCLK 引脚中输入的外部时钟。

MOC 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器设置为 80H。

图 5-5. 主 OSC 控制寄存器 (MOC) 的格式

地址: FFA2H 复位后: 80H R/W

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

MSTOP	高速系统时钟操作的控制	
	X1 振荡模式	外部时钟输入模式
0	X1 振荡器工作	允许 EXCLK 引脚的外部时钟
1	X1 振荡器停止	禁止 EXCLK 引脚的外部时钟

- 注意事项**
1. 当 MSTOP 被设为 1 时，请务必确认 CPU 是使用除高速系统时钟外的时钟在进行操作。在以下任一条件下进行特定地设置。
 - MCS = 0 时 (CPU 使用内部高速振荡时钟进行操作)
 2. 当时钟操作模式选择寄存器 (OSCCTL) 的第 6 位 (OSCSEL) 为 0 时 (I/O 端口模式)，不要将 MSTOP 清除为 0。
 3. 周边硬件时钟停止时，周边硬件不能工作。若要在周边硬件时钟停止后恢复周边硬件的操作，则应对周边硬件进行初始化。

(5) 主时钟模式寄存器 (MCM)

该寄存器用于选择提供给 CPU 时钟的主系统时钟以及提供给周边硬件时钟的时钟。

MCM 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器清除为 00H。

图 5-6. 主时钟模式寄存器 (MCM) 的格式

地址: FFA1H 复位后: 00H R/W[#]

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

XSEL	MCM0	提供给主系统时钟和周边硬件的时钟的选择	
		主系统时钟 (f _{XP})	周边硬件时钟 (f _{PRS})
0	0	内部高速振荡时钟 (f _{RH})	内部高速振荡时钟 (f _{RH})
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	主系统时钟状态
0	使用内部高速振荡时钟进行操作
1	使用高速系统时钟进行操作

注 第 1 位是只读的。

注意事项 1. 复位释放后 XSEL 只能改变一次。

2. 不管 XSEL 和 MCM0 是何设置, 不同于 f_{PRS} 的时钟都将会用于以下外围功能。

- 看门狗定时器 (使用内部低速振荡时钟进行操作)
- 当 “f_{RL}” 或 “f_{RL/2⁷}” 被选作 8 位定时器 H1 的计数时钟时 (使用内部低速振荡时钟进行操作)
- 周边硬件将外部时钟选作时钟源 (除了 TM00 的外部计数时钟被选择 (TI000 引脚有效沿) 时)

(6) 振荡稳定时间计数状态寄存器 (OSTC)

该寄存器用于表示 X1 时钟振荡稳定时间计数器的计数状态。当 X1 时钟振荡通过用作 CPU 时钟的内部高速振荡时钟开始工作时，将可以检测 X1 时钟振荡稳定时间。

OSTC 可以通过 1 位或 8 位存储器操作指令来读取。

复位释放时（通过 RESET 输入，POC，LVI，以及 WDT 进行复位），STOP 指令可以将 OSTC 清除为 00H，而 MSTOP（MOC 寄存器的第 7 位）= 1 也可以将 OSTC 清除为 00H。

图 5-7. 振荡稳定时间计数状态寄存器 (OSTC) 的格式

地址： FFA3H 复位后： 00H R

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

MOST11	MOST13	MOST14	MOST15	MOST16	振荡稳定时间的状态			
					$f_x = 1 \text{ MHz}$	$f_x = 2 \text{ MHz}$	$f_x = 4 \text{ MHz}$	
0	0	0	0	0	少于 $2^{11}/f_x$	少于 2.04 ms	少于 1.02 ms	少于 510 μs
1	0	0	0	0	$2^{11}/f_x \text{ min.}$	2.04 ms min	1.02 ms min	510 μs min
1	1	0	0	0	$2^{13}/f_x \text{ min.}$	8.20 ms min	4.10 ms min	2.04 ms min
1	1	1	0	0	$2^{14}/f_x \text{ min.}$	16.38 ms min.	8.19 ms min.	4.10 ms min
1	1	1	1	0	$2^{15}/f_x \text{ min.}$	32.77 ms min.	16.38 ms min.	8.19 ms min.
1	1	1	1	1	$2^{16}/f_x \text{ min.}$	65.45 ms min.	32.77 ms min.	16.38 ms min.

- 注意事项**
- 上述时间过后，各个位将会从 MOST11 开始按顺序设置为 1 并保持为 1。
 - 振荡稳定时间计数器用于累计通过 OSTC 所设置的振荡稳定时间。如果在内部高速振荡时钟被用作 CPU 时钟时先进入 STOP 模式，然后释放，那么将按如下对振荡稳定时间进行设置。
 - OSTC 振荡稳定期望时间 \leq 由 OSTC 设定的振荡稳定时间
 因此，需要注意在 STOP 模式被释放后，只有达到 OSTC 设定的振荡稳定时间时的状态会被赋予 OSTC。
 - X1 时钟振荡稳定等待时间不包括时钟振荡开始之前的时间（以下“a”）。



备注 fx: X1 时钟振动频率

(7) 振荡稳定时间选择寄存器 (OSTS)

该寄存器用于在 STOP 模式被释放时选择 X1 时钟振荡稳定等待时间。

当 X1 时钟被选作 CPU 时钟时，在 STOP 模式被释放后操作将会通过使用 OSTS 来等待一段时间。

当内部高速振荡时钟被选作 CPU 时钟时，在 STOP 模式被释放后应对 OSTC 进行确认，确认期望的振荡稳定时间已经过去。振荡稳定时间可以使用 OSTC 与时间集进行核对。

OSTS 可以通过 8 位存储器操作指令来进行设置。

复位信号生成将 OSTS 设为 05H。

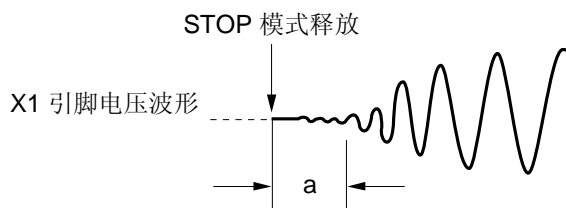
图 5-8. 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFA4H 复位后: 05H R/W

符号	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	振荡稳定时间的选择			
			$f_x = 1 \text{ MHz}$	$f_x = 2 \text{ MHz}$	$f_x = 4 \text{ MHz}$	
0	0	1	$2^{11}/f_x$	2.04 ms	1.02 ms	510 μs
0	1	0	$2^{13}/f_x$	8.19 ms	4.10 ms	2.04 m
0	1	1	$2^{14}/f_x$	16.38 ms	8.19 ms	4.10 ms
1	0	0	$2^{15}/f_x$	32.77 ms	16.38 ms	8.19 ms
1	0	1	$2^{16}/f_x$	65.45 ms	32.77 ms	16.38 ms
其他情况			禁止设置			

- 注意事项 1.** 若要在 X1 时钟被用作 CPU 时钟时设置 STOP 模式，则应在执行 STOP 指令前对 OSTS 进行设置。
- 2.** 在 X1 时钟振荡稳定时间内不要更改 OSTS 寄存器的值。
- 3.** 振荡稳定时间计数器用于累计通过 OSTS 所设置的振荡稳定时间。如果在内部高速振荡时钟被用作 CPU 时钟时先进入 STOP 模式，然后释放，那么将按如下对振荡稳定时间进行设置。
- OSTC 振荡稳定期望时间 \leq 由 OSTS 设定的振荡稳定时间
- 因此，需要注意在 STOP 模式被释放后，只有达到 OSTS 设定的振荡稳定时间时的状态会被赋予 OSTC。
- 4.** X1 时钟振荡稳定等待时间不包括时钟振荡开始之前的时间（以下“a”）。



备注 f_x : X1 时钟振动频率

5.4 系统时钟振荡器

5.4.1 X1 振荡器

X1 振荡器通过连接在 X1 和 X2 引脚上的晶体谐振器或陶瓷谐振器（1 至 4 MHz）来进行振动。也可以输入一个外部时钟。在这种情况下，将时钟信号输入至 EXCLK 引脚中。

图 5-9 显示了 X1 振荡器的外部电路的示例。

图 5-9. X1 振荡器外部电路示例



<R>

注意事项 当使用 X1 振荡器时，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。

- 连接线越短越好。
- 连接线不应与其他信号线交叉。
- 流经的电流变化较大的信号线不要在振荡器周围布线。
- 要保持振荡器电容器的接地点电压与 Vss 相同。
- 不要将电容的地信号接入大电流地。
- 不要从振荡器获取信号。

在图 5-10 中列出了需要注意的事项。

图 5-10. 错误的谐振器连接示例 (1/2)

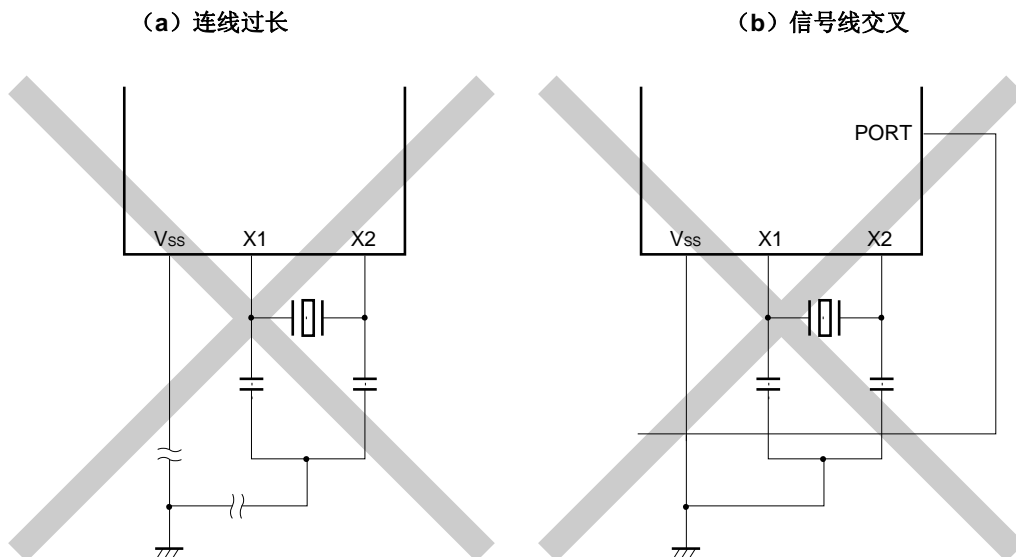
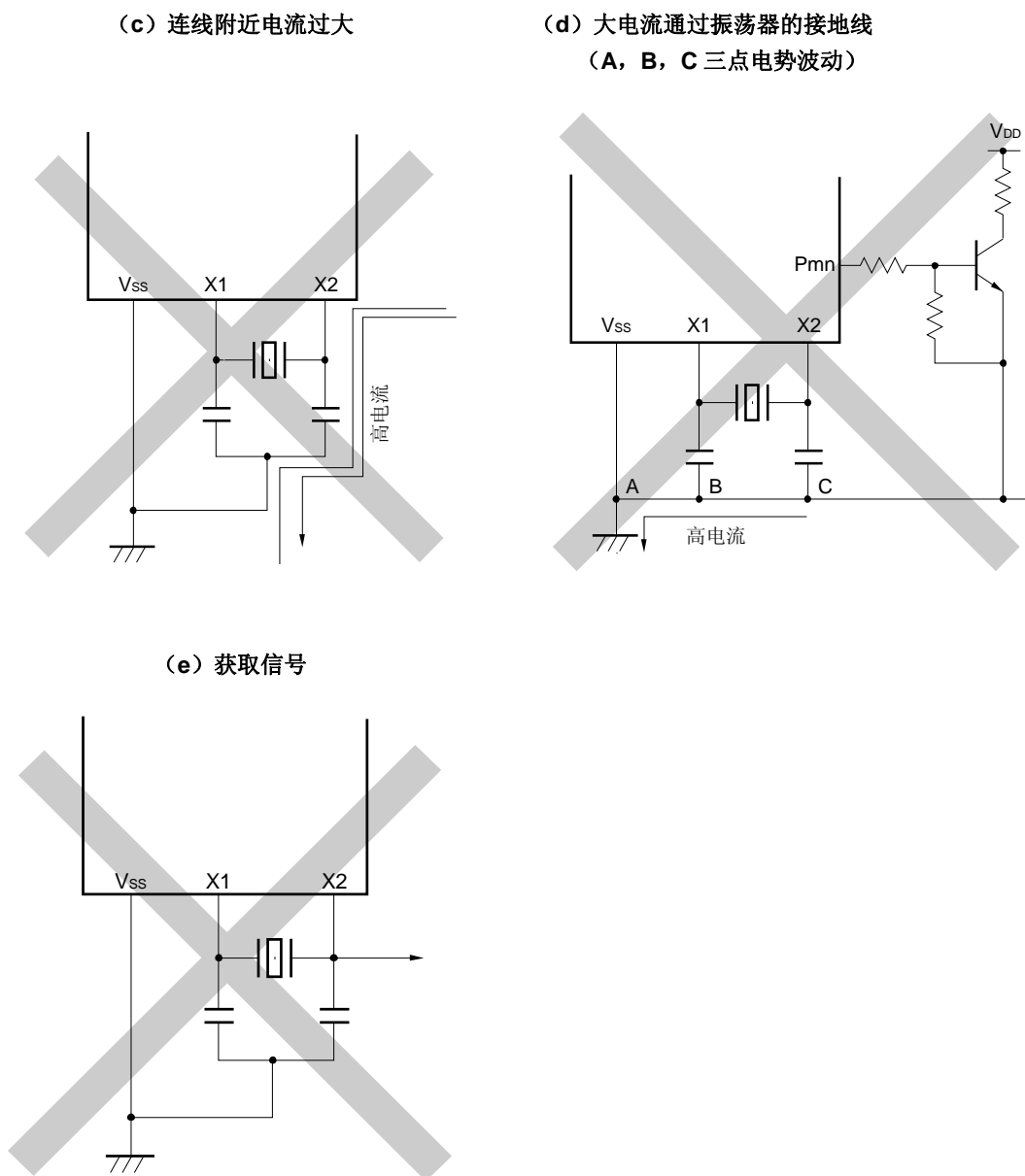


图 5-10. 错误的谐振器连接示例 (2/2)



5.4.2 内部高速振荡器

μ PD179F11x, μ PD 179F12x 微控制器中包含内部高速振荡器。振荡可以通过内部振荡模式寄存器 (RCM) 来进行控制。

复位释放后, 内部高速振荡器会自动开始振荡 ($4 \text{ MHz} \pm 2\%$)。

5.4.3 内部低速振荡器

μ PD179F11x, μ PD 179F12x 微控制器中包含内部低速振荡器。

内部低速振荡时钟只可以用作看门狗定时器以及 8 位定时器 H1 的时钟。内部低速振荡时钟不能被用作 CPU 时钟。

可以通过选项字节来选择“可以通过软件停止”或“不能停止”。当设置为“可以通过软件停止”时, 振荡可以通过内部振荡模式寄存器 (RCM) 来进行控制。

<R> 复位释放后，内部低速振荡器自动产生振荡，并且如果使用选项字节允许看门狗定时器操作，则可以驱动看门狗定时器（240 kHz（TYP.））。

5.4.4 预分频器

预分频器用于在主系统时钟被选作提供给 CPU 的时钟时通过对主系统时钟进行分频来生成不同的时钟。

5.5 时钟发生器的操作

时钟发生器用于生成以下时钟并控制 CPU 的操作模式，例如待机模式（参见图 5-1）。

- 主系统时钟 f_{XP}
- 高速系统时钟 f_{XH}
- $\left\{ \begin{array}{l} X1 \text{ 时钟 } f_x \\ \text{外部主系统时钟 } f_{EXCLK} \end{array} \right.$
- 内部高速振荡时钟 f_{RH}
- 内部低速振荡时钟 f_{RL}
- CPU 时钟 f_{CPU}
- 周边硬件时钟 f_{PRS}

在 μ PD179F11x, 179F12x 微控制器中，当内部高速振荡器在复位释放后开始输出时 CPU 开始工作，因此以下操作被允许。

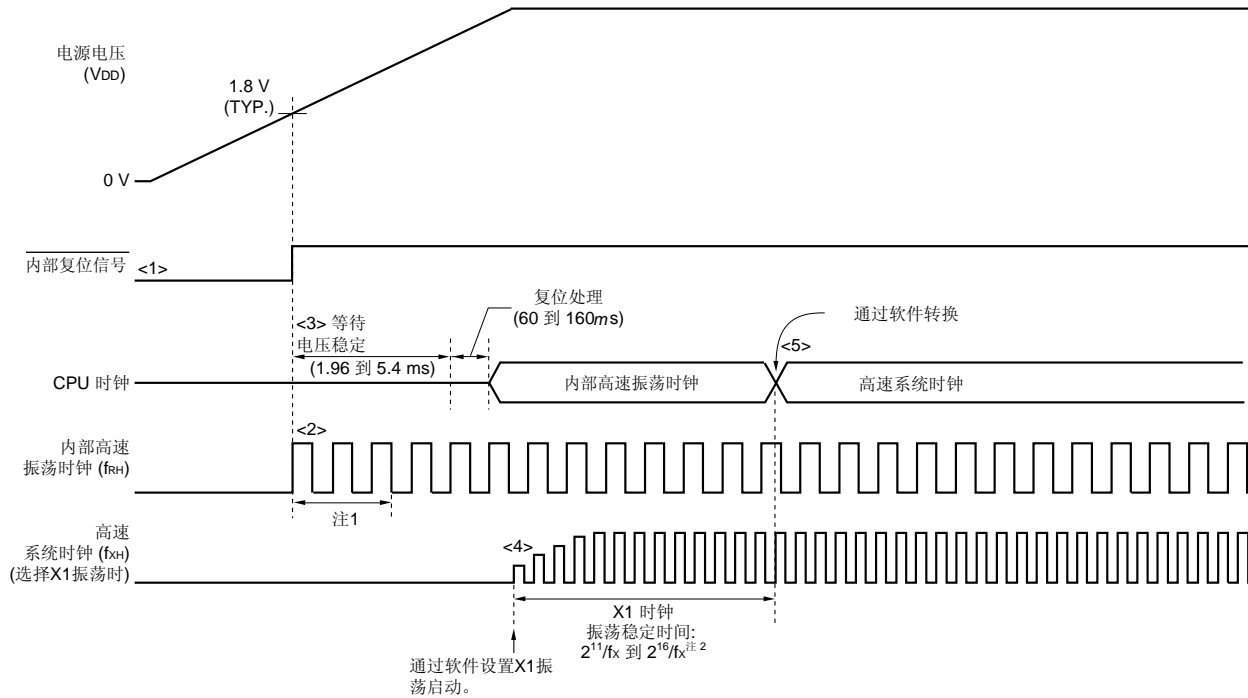
(1) 安全功能的增强

当 X1 时钟按默认设置被设置为 CPU 时钟时，如果 X1 时钟损坏或错误连接，那么设备将不能运行，因此复位释放后不应进行操作。然而，CPU 的开始时钟是内部高速振荡时钟，因此设备可以在复位释放后通过内部高速振荡时钟开始操作。所以，可以通过执行最低操作来安全关闭系统，例如在发生故障时通过软件响应复位源或执行安全处理。

(2) 性能的改进

由于 CPU 无需等待 X1 时钟振荡稳定时间便可开始操作，因此整体的性能将会得到改进。电源打开时，时钟发生器的操作如图 5-11 中所示。

图 5-11 电源打开时时钟发生器的操作



- <1> 电源打开时，内部复位信号通过上电清除（POC）电路来生成。
- <2> 当供电电压超过 1.8 V（TYP.）时，复位将会被释放且内部高速振荡器将自动开始振荡。
- <3> 供电电压和调节器电压的稳定时间过后，CPU 会在复位释放后在内部高速振荡时钟开始操作，然后执行复位处理。
- <4> 通过软件设置 X1 时钟的振荡的开始（参见 5.6.1 控制高速系统时钟示例中的（1））。
- <5> 在将 CPU 时钟转换至 X1 或 XT1 时钟时，应先等待时钟振荡稳定，然后再通过软件进行转换（参见 5.6.1 控制高速系统时钟示例中的（3））。

- 注 1. 内部电压稳定时间包括了内部高速振荡时钟的振荡准确稳定时间。
2. 当 CPU 在内部高速振荡时钟上进行操作时释放复位（上图）或释放 STOP 模式时，使用振荡稳定时间计数状态寄存器（OSTC）来确认 X1 时钟的振荡稳定时间。如果 CPU 在高速系统时钟（X1 振荡）上进行操作，那么应在使用振荡稳定时间选择寄存器（OSTS）时设置振荡稳定时间。

注意事项 使用从 EXCLK 引脚中输入的外部时钟时将不需要等待振荡稳定时间。

备注 微控制器进行操作时，可以通过软件设置来停止没有用作 CPU 时钟的时钟。内部高速振荡时钟和高速系统时钟可以通过执行 STOP 指令来停止（参见 5.6.1 控制高速系统时钟示例中的（4）以及 5.6.2 控制内部高速振荡时钟示例中的（3））。

5.6 控制时钟

5.6.1 控制高速系统时钟的示例

可以使用以下两种类型的高速系统时钟。

- X1 时钟：晶体 / 陶瓷谐振器通过 X1 和 X2 引脚来连接。
- 外部主系统时钟：外部时钟被输入到 EXCLK 引脚中。

未使用高速系统时钟时，X1/P121 和 X2/EXCLK/P122 引脚可以被用作 I/O 端口引脚。

注意事项 复位释放后 X1/P121 和 X2/EXCLK/P122 引脚将处于 I/O 端口模式中。

以下详细说明了下述情况中设置步骤的示例。

- (1) 使用振荡 X1 时钟时
- (2) 使用外部主系统时钟时
- (3) 将高速系统时钟用作 CPU 时钟以及周边硬件时钟时
- (4) 停止高速系统时钟时

(1) 振荡 X1 时钟时的设置步骤示例

- <1> 设置 P121/X1/OCD0A 和 P122/X2/EXCLK/OCD0B 引脚并选择 X1 时钟或外部时钟（OSCCTL 寄存器）
当 EXCLK 被清零且 OSCSEL 被设置为 1 时，模式将会从端口模式转换至 X1 振荡模式。

EXCLK	OSCSEL	高速系统时钟引脚的操作模式	P121/X1 引脚	P122/X2/EXCLK 引脚
0	1	X1 振荡模式	晶体 / 陶瓷谐振器连接	

- <2> 控制 X1 时钟的振荡（MOC 寄存器）
如果 MSTOP 被清除为 0，那么 X1 振荡器将开始振荡。

- <3> 等待 X1 时钟振荡的稳定
检查 OSTC 寄存器并等待所需时间。
在等待期间，其他软件处理可以通过内部高速振荡时钟来完成。

注意事项 1. X1 时钟操作时不要改变 EXCLK 和 OSCSEL 的值。

- 2.** 在供电电压达到所使用的时钟的操作电压时对 X1 时钟进行设置（参见第二十一章 电气特性）。

(2) 使用外部主系统时钟时的设置步骤示例

- <1> 设置 P121/X1/OCD0A 和 P122/X2/EXCLK/OCD0B 引脚并选择操作模式（OSCCTL 寄存器）
当 EXCLK 和 OSCSEL 被设置为 1 时，模式将会从端口模式转换至外部时钟输入模式。

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	I/O 端口	外部时钟输入

- <2> 控制外部主系统时钟输入（MOC 寄存器）
 当 MSTOP 被清零时，将会允许外部主系统时钟的输入。

- 注意事项**
1. 外部主系统时钟运行时不要改变 EXCLK 和 OSCSEL 的值。
 2. 在供电电压达到所使用的时钟的操作电压时对外部主系统时钟进行设置（参见第二十一章 电气特性）。

(3) 将高速系统时钟用作 CPU 时钟以及周边硬件时钟时的设置步骤示例

- <1> 设置高速系统时钟振荡[※]
 （参见 5.6.1 (1) X1 时钟振荡时的设置步骤示例以及 (2) 使用外部主系统时钟时的设置步骤示例。）

注 <1>的设置的高速系统时钟已经运行时不需要。

- <2> 将高速系统时钟设置为主系统时钟（MCM 寄存器）
 当 XSEL 和 MCM0 被设为 1 时，高速系统时钟将会作为主系统时钟和周边硬件时钟来提供。

XSEL	MCM0	主系统时钟以及提供给周边硬件的时钟的选择	
		主系统时钟 (f _{XP})	周边硬件时钟 (f _{PRS})
1	1	高速系统时钟 (f _{XH})	高速系统时钟 (f _{XH})

注意事项 如果高速系统时钟被选作主系统时钟，那么除了高速系统时钟，其他时钟均不能被设置为周边硬件时钟。

- <3> 将主系统时钟设置为 CPU 时钟并选择分频比率（PCC 寄存器）
 当 CSS 被清零时，主系统时钟将会提供时钟给 CPU。若要选择 CPU 时钟的分频比率，则应使用 PCC0，PCC1，以及 PCC2。

PCC2	PCC1	PCC0	CPU 时钟 (f _{CPU}) 选择	
			FLMD0 = 0	FLMD0 = 1
0	0	0	f _{XP}	f _{RH}
0	0	1	f _{XP} /2 (缺省)	f _{RH} /2 (缺省)
0	1	0	f _{XP} /2 ²	f _{RH} /2 ²
0	1	1	f _{XP} /2 ³	f _{RH} /2 ³
1	0	0	f _{XP} /2 ⁴	f _{RH} /2 ⁴
其他情况			禁止设置	

(4) 停止高速系统时钟时的设置步骤示例

高速系统时钟可以通过以下两种方法来停止。

- 执行 STOP 指令并停止 X1 振荡（如果使用了外部时钟，则将禁止时钟输入）
- 将 MSTOP 设为 1 并停止 X1 振荡（如果使用了外部时钟，则将禁止时钟输入）

(a) 若要执行 STOP 指令

- <1> 通过设置来停止周边硬件

停止不能在 STOP 模式中使用的周边硬件（关于不能在 STOP 模式中使用的周边硬件的详细信息，请参见第十三章 待机功能）。

- <2> 待机释放后设置 X1 时钟振荡稳定时间
当 CPU 在 X1 时钟上进行操作时，应在执行 STOP 指令前设置 OSTC 寄存器的值。
- <3> 执行 STOP 指令
执行 STOP 指令时，系统被置于 STOP 模式中，且 X1 振荡停止（禁止外部时钟的输入）。

(b) 若要通过将 MSTOP 置 1 来停止 X1 振荡（禁止外部时钟输入）

- <1> 确认 CPU 时钟的状态（PCC 和 MCM 寄存器）
通过 MCS 来确认 CPU 在除高速系统时钟以外的时钟上运行。
当 MCS = 1 时，高速系统时钟将提供给 CPU，因此应将 CPU 时钟改为内部高速振荡时钟。

MCS	CPU 时钟状态
0	内部高速振荡时钟
1	高速系统时钟

- <2> 停止高速系统时钟（MOC 寄存器）
当 MSTOP 被设为 1 时，X1 振荡停止（禁止外部时钟的输入）。

注意事项 当 MSTOP 被设置为 1 时，请务必确认 MCS = 0。此外，还应停止在高速系统时钟上运行的周边硬件。

5.6.2 控制内部高速振荡时钟的示例

以下详细说明了在下述案例中时钟设置的步骤。

- (1) 重新开始内部高速振荡时钟的振荡时
- (2) 将内部高速振荡时钟用作 CPU 时钟，并且将内部高速振荡时钟或高速系统时钟用作周边硬件时钟时
- (3) 停止内部高速振荡时钟时

(1) 重新开始内部高速振荡时钟的振荡时的设置步骤的示例^{#1}

- <1> 设置内部高速振荡时钟的振荡重新开始（RCM 寄存器）
当 RSTOP 被清零时，内部高速振荡时钟将开始运行。
- <2> 等待内部高速振荡时钟振荡精确稳定的时间（RCM 寄存器）
等待直到 RSTS 被置 1^{#2}。

- 注**
- 1. 复位释放后，内部高速振荡器将自动开始振荡，且内部高速振荡时钟将会被选作 CPU 时钟。
 - 2. 如果 CPU 时钟和周边硬件时钟不需要高精度度，那么将不需要等待时间。

(2) 将内部高速振荡时钟用作 CPU 时钟，并将内部高速振荡时钟或高速系统时钟用作周边硬件时钟时设置步骤示例

<1> 重新开始内部高速振荡时钟的振荡^注

(参见 5.6.2 (1) 重新开始内部高速振荡时钟的振荡时的设置步骤的示例)。

• 高速系统时钟^注

(将高速系统时钟用作周边硬件时钟时需要该设置。参见 5.6.1 (1) X1 时钟振荡时的设置步骤示例以及 (2) 使用外部主系统时钟时的设置步骤示例。)

注 <1>的设置内部高速振荡时钟或高速系统时钟已经运行时是不需要的。

<2> 将提供的时钟选作主系统时钟以及周边硬件时钟 (MCM 寄存器)

使用 XSEL 和 MCM0 来设置主系统时钟以及周边硬件时钟

XSEL	MCM0	主系统时钟以及提供给周边硬件的时钟的选择	
		主系统时钟 (f _{XP})	周边硬件时钟 (f _{PRS})
0	0	内部高速振荡时钟 (f _{RH})	内部高速振荡时钟 (f _{RH})
0	1		
1	0		高速系统时钟 (f _{XH})

<3> 选择 CPU 时钟的分频比率 (PCC 寄存器)

当 CSS 被清零时，主系统时钟将会提供时钟给 CPU。若要选择 CPU 时钟的分频比率，则应使用 PCC0，PCC1，以及 PCC2。

PCC2	PCC1	PCC0	CPU 时钟 (f _{CPU}) 选择	
			FLMD0 = 0	FLMD0 = 1
0	0	0	f _{XP}	f _{RH}
0	0	1	f _{XP} /2 (缺省)	f _{RH} /2 (缺省)
0	1	0	f _{XP} /2 ²	f _{RH} /2 ²
0	1	1	f _{XP} /2 ³	f _{RH} /2 ³
1	0	0	f _{XP} /2 ⁴	f _{RH} /2 ⁴
其他情况			禁止设置	

(3) 停止内部高速振荡时钟时的设置步骤示例

内部高速振荡时钟可以通过以下两种方法来停止。

- 执行 STOP 指令以设置 STOP 模式
- 将 RSTOP 设置为 1 并停止内部高速振荡时钟

(a) 若要执行 STOP 指令

<1> 周边硬件的设置

停止不能在 STOP 模式中使用的周边硬件 (关于不能在 STOP 模式中使用的周边硬件的详细信息，请参见第十三章 待机功能)。

<2> 待机释放后设置 X1 时钟振荡稳定时间

当 CPU 在 X1 时钟上进行操作时，应在执行 STOP 指令前设置 OSTC 寄存器的值。

<3> 执行 STOP 指令

执行 STOP 指令时，系统被置于 STOP 模式中，且内部高速振荡时钟被停止。

(b) 若要通过将 RSTOP 置 1 来停止内部高速振荡时钟

<1> 确认 CPU 时钟的状态（PCC 和 MCM 寄存器）

通过 MCS 来确认 CPU 在除内部高速振荡时钟以外的时钟上运行。

当 MCS = 0 时，内部高速振荡系统时钟将提供给 CPU，因此应将 CPU 时钟改为高速系统时钟。

MCS	CPU 时钟状态
0	内部高速振荡时钟
1	高速系统时钟

<2> 停止内部高速振荡时钟（RCM 寄存器）

当 RSTOP 被设置为 1 时，内部高速振荡时钟将停止。

注意事项 当 RSTOP 被设置为 1 时，请务必确认 MCS = 1。此外，还应停止在高速振荡时钟上运行的周边硬件。

5.6.3 控制内部低速振荡时钟的示例

内部低速振荡时钟不能被用作 CPU 时钟。

只有以下周边硬件可以使用该时钟进行操作。

- 看门狗定时器
- 8 位定时器 H1（如果 frL 被选作计数时钟）

此外，以下操作模式可以通过选项字节来选择。

- 内部低速振荡器不能停止的模式
- 内部低速振荡器可以通过软件停止的模式

复位释放后，内部低速振荡器会自动开始振荡，且如果通过使用选项字节可以允许看门狗定时器的操作，那么看门狗定时器也会启动（240 kHz（TYP.））。

<R>

(1) 停止内部低速振荡时钟时的设置步骤示例

<1> 将 LSRSTOP 设置为 1（RCM 寄存器）

当 LSRSTOP 被设置为 1 时，内部低速振荡时钟将停止。

(2) 重新开始内部低速振荡时钟的振荡时的设置步骤的示例

<1> 将 LSRSTOP 设置为 0（RCM 寄存器）

当 LSRSTOP 被清零时，内部低速振荡时钟将重新开始振荡。

注意事项 如果通过选项字节选择了“内部低速振荡器不能停止”的模式，那么将不能控制内部低速振荡时钟的振荡。

5.6.4 提供给CPU以及周边硬件的时钟

下表显示了提供给 CPU 及周边硬件的时钟间的关系以及寄存器的设置。

表 5-3. 提供给 CPU 及周边硬件的时钟，以及寄存器设置

提供的时钟		XSEL	MCM0	EXCLK
提供给 CPU 的时钟	提供给周边硬件的时钟			
内部高速振荡时钟		0	×	×
内部高速振荡时钟	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**
1. XSEL: 主时钟模式寄存器 (MCM) 的第 2 位
 2. MCM0: MCM 的 0 位
 3. EXCLK: 时钟操作模式选择寄存器 (OSCCTL) 的第 7 位
 4. ×: 不考虑

5.6.5 CPU时钟状态转换图

图 5-12 显示了该产品的 CPU 时钟状态转换图。

图 5-12. CPU 时钟状态转换图

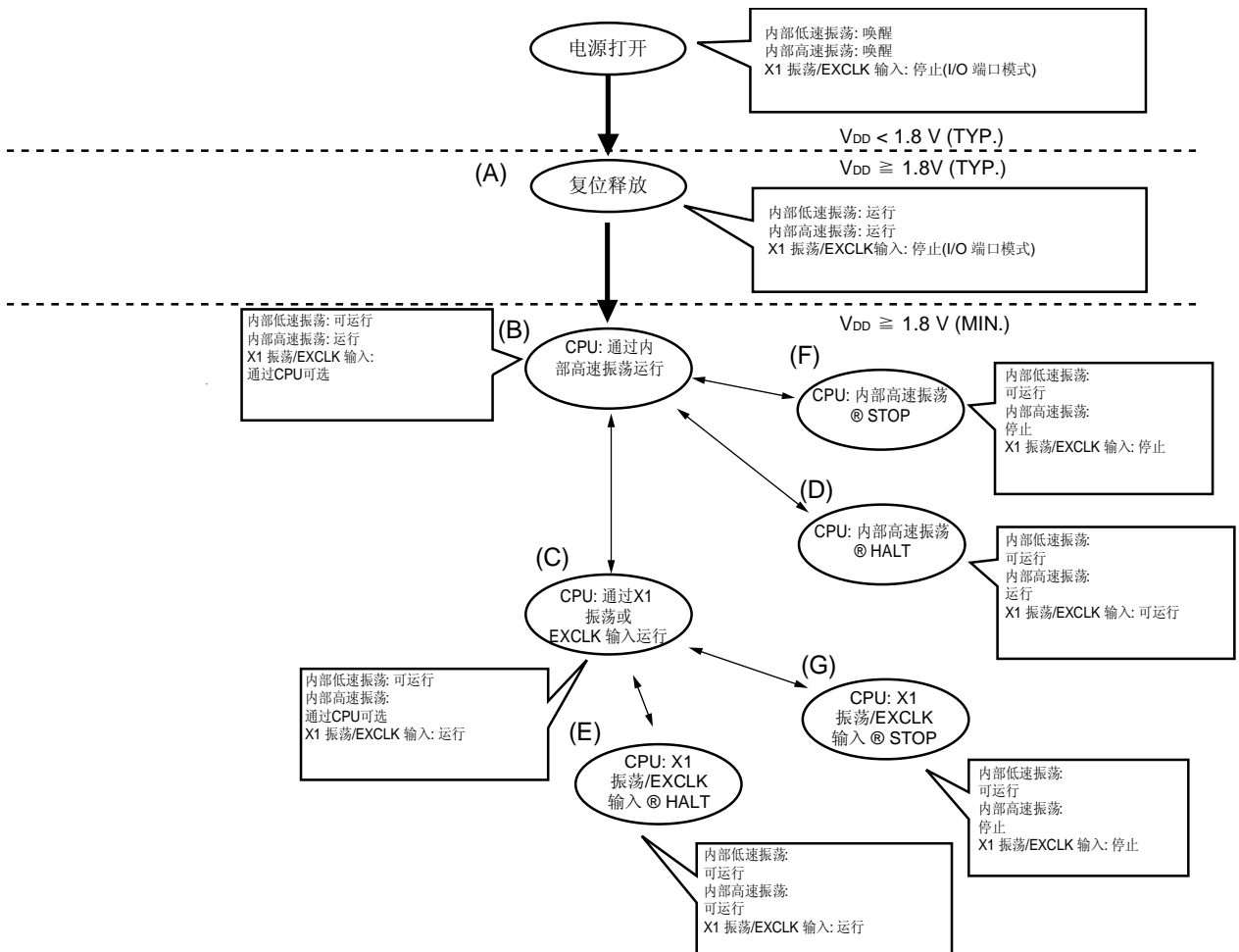


表 5-4 显示了 CPU 时钟的转换以及设置 SFR 寄存器的示例。

表 5-4. CPU 时钟转换以及 SFR 寄存器设置示例 (1/2)

(1) 复位释放 (A) 后通过内部高速振荡时钟 (B) 进行操作的 CPU

状态转换	SFR 寄存器设置
(A) → (B)	SFR 寄存器不必设置 (复位释放后的默认状态)。

(2) 复位释放 (A) 后通过高速系统时钟 (C) 进行操作的 CPU

(复位释放 (B) 后 CPU 立即通过内部高速振荡时钟进行操作)。

(SFR 寄存器的设置次序) →

SFR 寄存器的设置标志	EXCLK	OSCSEL	MSTOP	OSTC 寄存器	XSEL	MCM0
状态转换						
(A) → (B) → (C) (X1 时钟: $1 \text{ MHz} \leq f_{XH} \leq 4 \text{ MHz}$)	0	1	0	必须检测	1	1
(A) → (B) → (C) (外部主时钟: $1 \text{ MHz} \leq f_{XH} \leq 4 \text{ MHz}$)	1	1	0	不必检测	1	1

(3) 从内部高速振荡时钟 (B) 转变为高速系统时钟 (C) 的 CPU 时钟

(SFR 寄存器的设置次序) →

SFR 寄存器的设置标志	EXCLK	OSCSEL	MSTOP	OSTC 寄存器	XSEL ^注	MCM0
状态转换						
(B) → (C) (X1 时钟: $1 \text{ MHz} \leq f_{XH} \leq 4 \text{ MHz}$)	0	1	0	必须检测	1	1
(B) → (C) (外部主时钟: $1 \text{ MHz} \leq f_{XH} \leq 4 \text{ MHz}$)	1	1	0	不必检测	1	1

在这些寄存器已经被设置 CPU 使用高速系统时钟操作时不需要时不需要

注 复位释放后该标志的值只能改变一次。如果它已经被设置, 则不需要该设置。

注意事项 在供电电压达到所使用的时钟的操作电压时对时钟进行设置 (参见第二十一章 电气特性)。

备注

1. 表 5-4 中的 (A) 至 (G) 对应于图 5-12 中的 (A) 至 (G)。
2. EXCLK, OSCSEL: 时钟操作模式选择寄存器 (OSCCTL) 的第 7 位和第 6 位
MSTOP: 主 OSC 控制寄存器 (MOC) 的第 7 位
XSEL, MCM0: 主时钟模式寄存器 (MCM) 的第 2 位以及 0 位
x: 不考虑

表 5-4. CPU 时钟转换以及 SFR 寄存器设置示例 (2/2)

(4) CPU 时钟从高速系统时钟 (C) 转变为内部高速振荡时钟 (B)

(SFR 寄存器的设置次序)

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

CPU 使用内部高速振荡时钟操作时不需要

<R> 注意事项 在供电电压达到要设置的时钟的可操作电压时对时钟进行设置 (参见第二十一章 电气特性)。

(5) CPU 通过内部高速振荡时钟 (B) 进行操作时设置的 HALT 模式 (D) ,
CPU 通过高速系统时钟 (C) 进行操作时设置的 HALT 模式 (E)

状态转换	设置
(B) → (D) (C) → (E)	执行 HALT 指令

(6) CPU 通过内部高速振荡时钟 (B) 进行操作时设置的 STOP 模式 (F) ,
CPU 通过高速系统时钟 (C) 进行操作时设置的 STOP 模式 (G)

(设置次序)

状态转换	设置	
(B) → (F) (C) → (G)	停止不能在 STOP 模式中进行操作的 外围功能	执行 STOP 指令

- 备注
1. 表 5-4 中的 (A) 至 (G) 对应于图 5-12 中的 (A) 至 (G)。
 2. MCM0: 主时钟模式寄存器 (MCM) 的第 0 位
RSTS, RSTOP: 内部振荡模式寄存器 (RCM) 的第 7 位以及 0 位

5.6.6 更改CPU时钟前的条件以及更改CPU时钟后的处理

更改 CPU 时钟前的条件以及更改 CPU 时钟后的处理如下所示。

表 5-5. 更改 CPU 时钟

CPU 时钟		更改前的条件	更改后的处理
更改前	更改后		
内部高速振荡时钟	X1 时钟	X1 振荡稳定 • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • 振荡稳定时间过后	内部高速振荡器可以停止 (RSTOP = 1)。
	外部主系统时钟	允许 EXCLK 引脚中的外部时钟的输入 • MSTOP = 0, OSCSEL = 1, EXCLK = 1	内部高速振荡器可以停止 (RSTOP = 1)。
X1 时钟	内部高速振荡时钟	内部高速振荡器的振荡 • RSTOP = 0	X1 振荡可以停止 (MSTOP = 1)。
外部主系统时钟			外部主系统时钟输入被禁止 (MSTOP = 1)。

5.6.7 CPU时钟和主系统时钟间转换所需的时间

通过设置处理器时钟控制寄存器 (PCC) 的 0 至 2 位 (PCC0 至 PCC2)，可以改变主系统时钟的分频比率。

实际转换操作不会在对 PCC 进行重写后立即执行；操作会在预转换时钟上持续几个时钟 (参见表 5-6)。

表 5-6. CPU 时钟和主系统时钟周期分频因子间转换所需的时间

转换前的设定值			转换后的设定值														
PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0
0	0	0	0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0	0	16 个时钟			16 个时钟			16 个时钟			16 个时钟					
0	0	1	8 个时钟			8 个时钟			8 个时钟			8 个时钟					
0	1	0	4 个时钟			4 个时钟			4 个时钟			4 个时钟					
0	1	1	2 个时钟			2 个时钟			2 个时钟			2 个时钟					
1	0	0	1 个时钟			1 个时钟			1 个时钟			1 个时钟					

备注 表 5-6 中所列出的时钟个数是转换前 CPU 时钟的个数。

通过设置主时钟模式寄存器 (MCM) 的 0 位 (MCM0)，主系统时钟将可以进行转换 (在内部高速振荡时钟和高速系统时钟间进行转换)。

实际转换操作不会在对 MCM0 进行重写后立即执行；操作会在预转换时钟上持续几个时钟 (参见表 5-7)。

CPU 是在内部高速振荡时钟上进行操作还是在高速系统时钟上进行操作可以使用 MCM 的第 1 位 (MCS) 来确定。

表 5-7. 主系统时钟转换所需的最长时间

转换前的设定值	转换后的设定值	
MCM0	MCM0	
	0	1
0		$1 + 2f_{RH}/f_{XH}$ 个时钟
1	$1 + 2f_{XH}/f_{RH}$ 个时钟	

注意事项 内部高速振荡时钟转换为高速系统时钟时，必须事先将 MCM 的第 2 位 (XSEL) 设为 1。复位释放后 XSEL 的值只能改变一次。

备注

1. 表 5-7 中所列出的时钟个数是转换前主系统时钟的个数。
2. 通过除去小数部分来计算表 5-7 中的时钟个数。

例 当主系统时钟从内部高速振荡时钟转换至高速系统时钟时 (@ 在 $f_{RH} = 1 \text{ MHz}$, $f_{XH} = 4 \text{ MHz}$ 时振荡)

那么 $1 + 2f_{RH}/f_{XH} = 1 + 2 \times 1/4 = 1 + 2 \times 0.25 = 1 + 0.5 = 1.5 \rightarrow 1$ 个时钟

5.6.8 时钟振荡停止前的条件

以下列出了用于停止时钟振荡 (禁止外部时钟输入) 而设置的寄存器标志以及时钟振荡停止前的条件。

表 5-8 时钟振荡停止前的条件以及标志设置

时钟	时钟振荡停止前的条件 (禁止外部时钟输入)	SFR 寄存器的标志设置
内部高速振荡时钟	MCS = 1 (CPU 在除内部高速振荡时钟外的时钟上进行操作)	RSTOP = 1
X1 时钟	MCS = 0 (CPU 在除高速系统时钟外的时钟上进行操作)	MSTOP = 1
外部主系统时钟		

5.6.9 周边硬件以及源时钟

以下列出了 μ PD179F11x, 179F12x 微控制器中所包含的周边硬件及源时钟。

表 5-9. 周边硬件和源时钟

源时钟		周边硬件时钟 (fPRS)	内部低速振荡时钟 (fRL)	TM50 输出	周边硬件引脚中的 外部时钟
周边硬件					
16 位定时器 / 事件计数器 00		Y	N	N	Y (TI000 引脚)
8 位定时器 / 事件计数器	50	Y	N	N	Y (TI50 引脚)
	51	Y	N	N	Y (TI51 引脚)
8 位定时器	H0	Y	N	Y	N
	H1	Y	Y	N	N
看门狗定时器		N	Y	N	N
时钟输出 ^{注 2}		Y	N	N	N
串行接口	UART6	Y	N	Y	N

备注 Y: 可选, N: 不可选

6.1 16 位定时器 / 事件计数器 00 的功能

16 位定时器 / 事件计数器 00 具有以下功能。

(1) 间隔定时器

16 位定时器 / 事件计数器 00 可以在预定时间间隔生成一个中断请求。

(2) 方波输出

16 位定时器 / 事件计数器 00 可以通过任何选中的频率输出一个方波。

(3) 外部事件计数器

16 位定时器 / 事件计数器 00 可以测量外部输入信号的脉冲数量。

(4) 单脉冲输出

16 位定时器 / 事件计数器 00 可以输出一个单脉冲，该单脉冲的输出脉宽可以任意设置。

(5) PPG 输出

16 位定时器 / 事件计数器 00 可以输出一个方波，该方波的频率和输出脉宽可以任意设置。

(6) 脉宽测量

16 位定时器 / 事件计数器 00 可以测量外部输入信号的脉宽。

6.2 16 位定时器 / 事件计数器 00 的配置

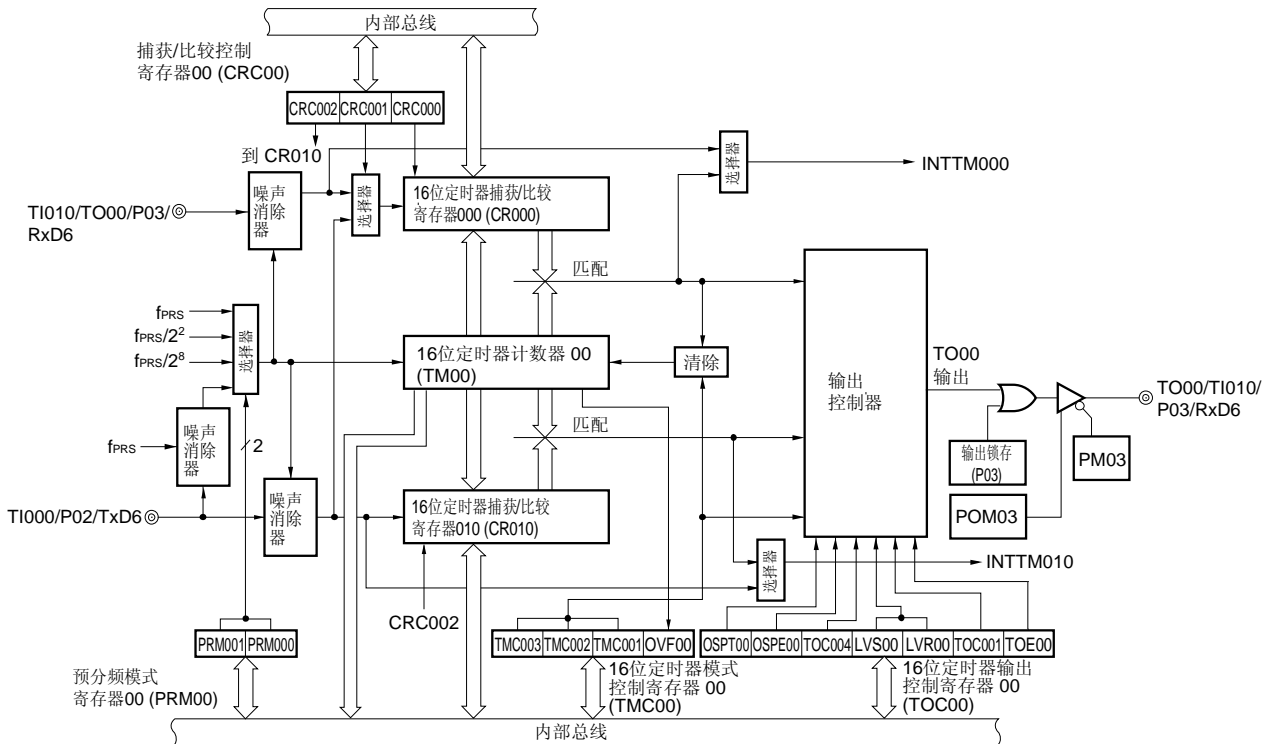
16 位定时器 / 事件计数器 00 包含有以下硬件。

表 6-1. 16 位定时器 / 事件计数器 00 的配置

项目	配置
定时器 / 计数器	16 位定时器计数器 00 (TM00)
寄存器	16 位定时器捕获 / 比较寄存器 000, 010 (CR000, CR010)
定时器输入	TI000, TI010 引脚
定时器输出	TO00 引脚, 输出控制器
控制寄存器	16 位定时器模式控制寄存器 00 (TMC00) 16 位定时器捕获 / 比较控制寄存器 00 (CRC00) 16 位定时器输出控制寄存器 00 (TOC00) 预分频模式寄存器 00 (PRM00) 端口模式寄存器 0 (PM0) 端口寄存器 0 (P0) 端口输出模式寄存器 0 (POM0)

图 6-1 显示了框图。

图 6-1. 16 位定时器 / 事件计数器 00 的框图



注意事项 1. TI010 的有效沿与定时器输出 (TO00) 不能同时用于 P03 引脚。选择任一功能。

- 注意事项 2.** 如果 16 位定时器模式控制寄存器 00 (TMC00) 的第 3 位和第 2 位的清零与捕获触发器的输入冲突, 那么捕获数据将不确定。
- 3.** 若要将模式从捕获模式转变为比较模式, 首先应将 TMC003 和 TMC002 位清零, 然后再更改设置。除非设备被复位, 否则捕获的值将不能保持在 CR000 中。如果模式已经变化比较模式, 则务必设置一个比较值

(1) 16 位定时器计数器 00 (TM00)

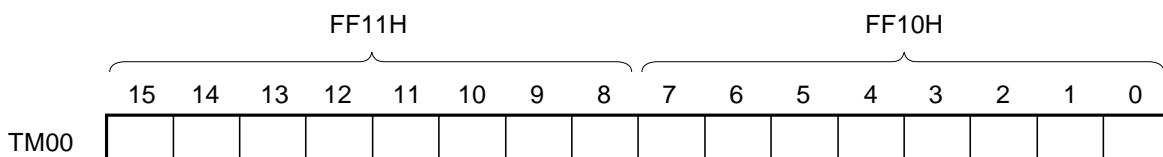
TM00 是一个 16 位只读寄存器, 用于对计数脉冲进行计数。

计数器与计数时钟的上升沿同步累加。

如果在操作过程中先读取计数值, 然后临时停止计数时钟的输入, 则将会读取该点的计数值。

图 6-2. 16 位定时计数器 00 (TM00) 的格式

地址: FF10H, FF11H 复位后: 0000H R



当 16 位定时器模式控制寄存器 00 (TMC00) 中第 3 位和第 2 位 (TMC003 和 TMC002) 的值不等于 00 时, TM00 的计数值可以通过读取 TM00 来读取。如果在 TMC003 和 TMC002 = 00 读取 TM00, 那么 TM00 的值将等于 0000H。

在以下情况中计数值会被复位为 0000H。

- 复位信号生成时
- 如果 TMC003 和 TMC002 被清除为 00
- TI000 引脚的有效沿在发生清除&开始的模式中被输入
- 如果 TM00 和 CR000 在发生清除&开始的模式中匹配时
- OSPT00 在单脉冲输出模式中被设置为 1 或有效沿被输入到 T1000 引脚中

注意事项 1. 即使 TM00 被读取, 也不能通过 CR010 来获取值。

- 2.** 如果在定时器计数期间参考了 TM00, 那么定时器计数会在参考处理期间停止操作, 而在参考处理结束之后, 定时器计数将会恢复。

因此, 如果执行了与 TM00 相关的处理, 那么在定时器计数期间将产生错误。

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

CR000 和 CR010 是 16 位寄存器, 它们通过使用 CRC00 选择的捕获功能或比较功能来进行使用。

定时器停止时 (TMC003 和 TMC002 = 00) 改变 CR000 的值。

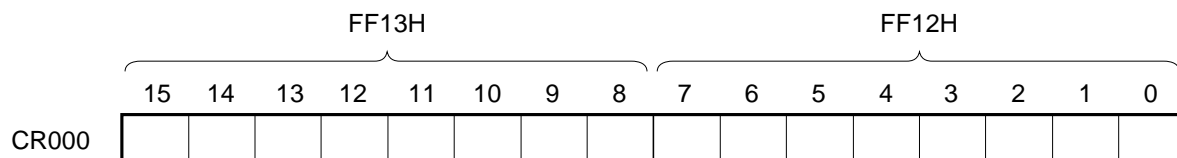
如果 CR010 的值已经通过一个特定的方式进行设置, 那么在操作过程中 CR010 的值可以改变。详细信息, 请参见 **6.5.1 TM00 操作期间重写 CR010**。

这些寄存器可以以 16 位为单位进行读取或写入。

复位信号生成将这些寄存器设置为 0000H。

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

地址: FF12H, FF13H 复位后: 0000H R/W



(i) 当 CR000 被用作比较寄存器时

在 CR000 中设定的值持续地与 TM00 计数值进行比较, 如果两者匹配, 则将生成一个中断请求信号 (INTTM000)。值会一直被保存直到 CR000 被重写。

注意事项 当 CR000 在比较模式中被设置时, 即使捕捉触发被输入到 CR000 中, CR000 也不能执行捕获操作。

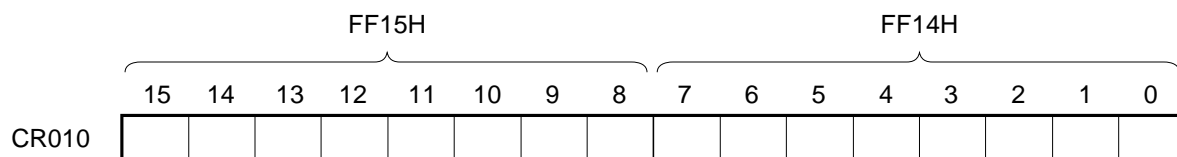
(ii) 当 CR000 被用作捕获寄存器时

TM00 的计数值会在输入捕获触发器时被捕获至 CR000 中。

用作捕获触发器时, 可以通过使用 CRC00 或 PRM00 来选择与 TI000 引脚相位相反的相位边沿或 TI010 引脚的有效沿。

图 6-4. 16 位定时器捕获 / 比较寄存器 010 (CR010) 的格式

地址: FF14H, FF15H 复位后: 0000H R/W



(i) 当 CR010 被用作比较寄存器时

在 CR010 中设定的值持续地与 TM00 计数值进行比较, 如果两者匹配, 则将生成一个中断请求信号 (INTTM010)。

注意事项 当 CR010 在比较模式中被设置时, 即使捕捉触发器被输入到 CR010 中, CR010 也不能执行捕获操作。

(ii) 当 CR010 被用作捕获寄存器时

TM00 的计数值会在输入捕获触发器时被捕获至 CR010 中。

可以将 TI000 引脚的有效沿选作捕获触发器。TI000 引脚的有效沿可以通过 PRM00 来设置。

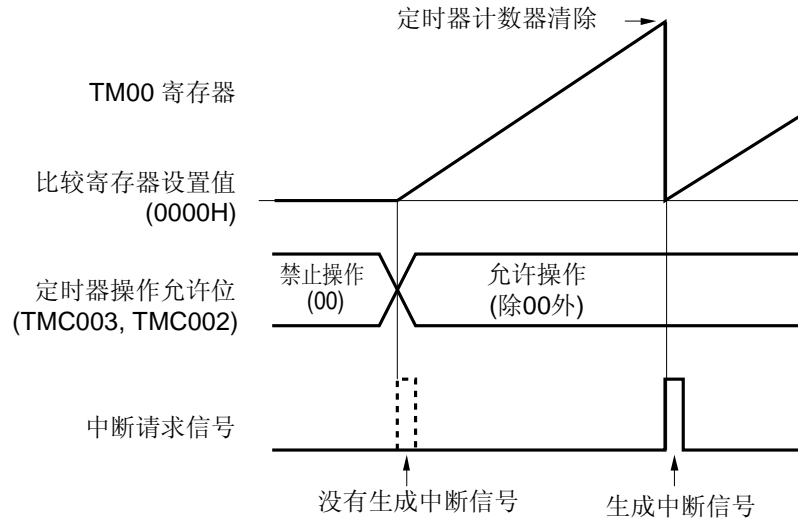
(iii) CR000 或 CR010 用作比较寄存器时的设置范围

CR000 或 CR010 用作比较寄存器时，按如下所示来对其进行设置。

操作	CR000寄存器设置范围	CR010寄存器设置范围
用作间隔定时器	$0000H < N \leq FFFFH$	$0000H^{\#} \leq M \leq FFFFH$
用作方波输出		通常情况下不使用该设置。屏蔽匹配中断信号 (INTTM010)。
用作外部事件计数器		
通过TI000引脚有效沿输入进入的清除&开始模式中的操作	$0000H^{\#} \leq N \leq FFFFH$	$0000H^{\#} \leq M \leq FFFFH$
用作自由运行定时器		
用作PPG输出	$M < N \leq FFFFH$	$0000H^{\#} \leq M < N$
用作单脉冲输出	$0000H^{\#} \leq N \leq FFFFH (N \neq M)$	$0000H^{\#} \leq M \leq FFFFH (M \neq N)$






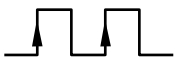

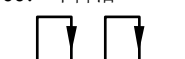
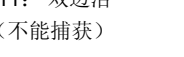





注 设为 0000H 时，定时器操作后不会立即发生匹配中断，定时器输出也不会发生变化，而第一个匹配时序则如下所示。当定时器计数器 (TM00 寄存器) 从 0000H 变为 0001H 时会在时序中产生匹配中断。

- 定时器计数器由于溢出而被清零时
- 定时器计数器由于 TI000 引脚有效沿而被清零时 (通过 TI000 引脚有效沿输入进入清除 & 开始模式时)
- 定时器计数器由于比较匹配而被清零时 (通过 TM00 和 CR000 间匹配进入清除 & 开始模式时 (CR000 不等于 0000H, CR010 等于 0000H))



- 备注**
1. N: CR000 寄存器设定值, M: CR010 寄存器设定值
 2. 关于 TMC003 和 TMC002 的详细信息, 请参见 6.3 (1) 16 位定时器模式控制寄存器 00 (TOC00)。

表 6-2. CR000 和 CR010 的捕获操作

外部输入信号 捕获操作	T1000 引脚输入 	T1010 引脚输入 		
CR000的捕获操作	CRC001 = 1 T1000 引脚输入 (反向) 	ES001 和 ES000 的设置 值 要进行捕获的边沿的位置	CRC001 位 = 0 T1010 引脚输入 	ES101 和 ES100的设置 值 要进行捕获的边沿的位置
		01: 上升沿 		01: 上升沿 
		00: 下降沿 		00: 下降沿 
	11: 双边沿 (不能捕获)	11: 双边沿 	11: 双边沿 	
中断信号	即使捕获到值， INTTM000 信号也不能生 成。	中断信号	每次捕获到值时都会成 生 INTTM000信号。	
CR010的捕获操作	T1000 引脚输入 ^注 	ES001 和 ES000 的设置 设置值 要进行捕获的边沿的位置		
		01: 上升沿 		
		00: 下降沿 		
	11: 双边沿 			
中断信号	每次捕获到值时都会成 生 INTTM010 信号。			

注 CR010 的捕获操作不会受 CRC001 位设置的影响。

注意事项 若使用与输入到 T1000 引脚中的相位相反的相位将 TM00 寄存器的计数值捕获至 CR000 寄存器，那么在值被捕获后将不能生成中断请求信号 (INTTM000)。如果在该操作过程中在 T1010 引脚上检测到有效沿，那么将不会执行捕获操作，但 INTTM000 信号将会生成成为一个外部中断信号。为了不使用外部中断，应屏蔽 INTTM000 信号。

备注 CRC001: 参见 6.3 (2) 捕获 / 比较控制寄存器 00 (CRC00)。
ES101, ES100, ES001, ES000: 参见 6.3 (4) 预分频模式寄存器 00 (PRM00)。

6.3 控制 16 位定时器 / 事件计数器 00 的寄存器

以下显示了用于控制 16 位定时器 / 事件计数器 00 的寄存器。

- 16 位定时器模式控制寄存器 00 (TMC00)
- 捕获 / 比较控制寄存器 00 (CRC00)
- 16 位定时器输出控制寄存器 00 (TOC00)
- 预分频模式寄存器 00 (PRM00)
- 端口模式寄存器 0 (PM0)
- 端口寄存器 0 (P0)
- 端口输出模式寄存器 0 (POM0)

(1) 16 位定时器模式控制寄存器 00 (TMC00)

TMC00 是一个 8 位寄存器，它用于设置 16 位定时器 / 事件计数器 00 的操作模式，TMC00 的清除模式以及输出时序，并且用于检测溢出。

禁止在操作（当 TMC003 和 TMC002 不等于 00）期间重写 TMC00。然后，当 TMC003 和 TMC002 被清除为 00（停止操作）且 OVF00 被清除为 0 时 TMC00 可以更改。

TMC00 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该 TMC00 设置为 00H。

注意事项 16 位定时器 / 事件计数器 00 在 TMC002 和 TMC003 分别被设置不同于 00（操作停止模式）的值时开始操作。将 TMC002 和 TMC003 设置为 00 以停止操作。

图 6-5. 16 位定时器模式控制寄存器 00 (TMC00) 的格式

地址: FFBAH 复位后: 00H R/W

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

TMC003	TMC002	16位定时器 / 事件计数器 00的允许操作
0	0	禁止16位定时器 / 事件计数器00的操作。停止提供操作时钟。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	通过TI000 引脚有效沿进入的清除&开始模式*
1	1	在TM00 与 CR000匹配的情况下进入的清除&开始模式。

TMC001	反转定时器输出的条件 (TO00)
0	<ul style="list-style-type: none"> • TM00与CR000匹配或TM00与CR010匹配
1	<ul style="list-style-type: none"> • TM00与CR000匹配或TM00与CR010匹配 • TI000 引脚有效沿的触发输入

OVF00	TM00溢出标志
清除 (0)	将OVF00清零或TMC003和TMC002 = 00
设置 (1)	发生溢出。
TM00的值在任何操作模式 (自由运行定时器模式, 通过TI000引脚有效沿输入进入的清除&开始模式, 以及在TM00和CR000匹配的基础上进入的清除&开始模式) 中从FFFFH转变为0000H时, OVF00都会被设为1。也可以通过将1写入OVF00将其设为1。	

注 TI000 引脚有效沿通过预分频模式寄存器 00 (PRM00) 的第 5 位和第 4 位 (ES001, ES000) 来进行设置。

(2) 捕获 / 比较控制寄存器 00 (CRC00)

CRC00 是用于控制 CR000 和 CR010 操作的寄存器。

禁止在操作（当 TMC003 和 TMC002 不等于 00）期间改变 CRC00 的值。

CRC00 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该 CRC00 设置为 00H。

图 6-6. 捕获 / 比较控制寄存器 00 (CRC00) 的格式

地址: FFBC H 复位后: 00H R/W

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

CRC002	CR010 操作模式选择
0	作为比较寄存器进行操作
1	作为捕获寄存器进行操作

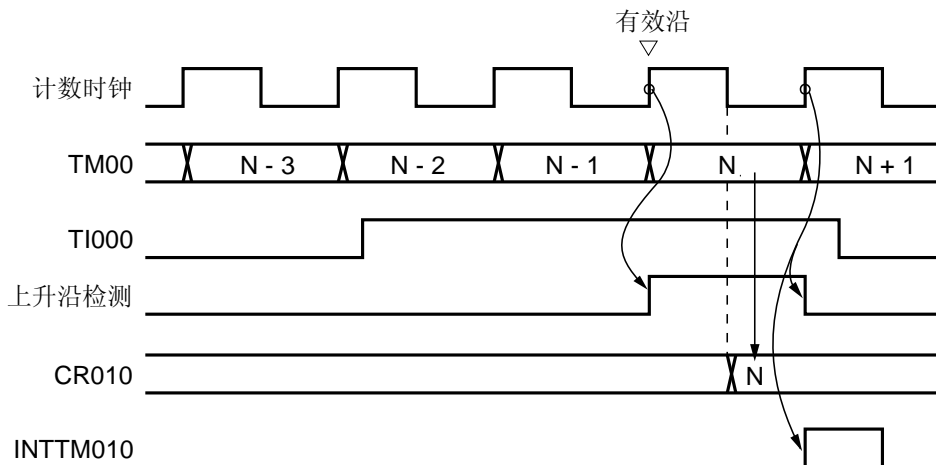
CRC001	CR000 捕获触发器选择
0	在 TI010 引脚的有效沿上进行捕获
1	通过反相在 TI000 引脚的有效沿上进行捕获 ^注
通过 PRM00 对 TI010 和 TI000 的有效沿进行设置。 如果 ES001 和 ES000 在 CRC001 为 1 时被设为 11（双沿），那么 TI000 引脚的有效沿将不能被检测。	

CRC000	CR000 操作模式选择
0	作为比较寄存器进行操作
1	作为捕获寄存器进行操作
如果 TMC003 和 TMC002 被设为 11（在 TM00 和 CR000 匹配的基础上进入的清除&开始模式），请务必将 CRC000 设为 0。	

注 在 TI010 引脚上检测到有效沿时，将不会执行捕获操作，但 INTTM000 信号将会生成成为一个外部中断信号。

注意事项 要确保捕获操作正确执行，捕获触发器需要一个比预分频模式寄存器 00 (PRM00) 所选择的计数时钟长两个周期的脉冲。

图 6-7. CR010 捕获操作示例（上升沿被指定时）



(3) 16 位定时器输出控制寄存器 00 (TOC00)

TOC00 是一个 8 位寄存器，用于控制 TO00 输出。

TOC00 只有在 OSPT00 进行操作时（当 TMC003 和 TMC002 不等于 00 时）才可以重写。禁止在操作过程中对其他位进行重写。

然而，在定时器运行过程中 TOC004 可以按重写 CR010 的方法进行重写（参见 6.5.1 TM00 操作期间重写 CR010）。

TOC00 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该 TOC00 清除为 00H。

注意事项 务必按以下步骤对 TOC00 进行设置。

- <1> 将 TOC004 和 TOC001 设为 1。
- <2> 仅将 TOE00 设为 1。
- <3> 将 LVS00 或 LVR00 设为 1。

图 6-8. 16 位定时器输出控制寄存器 00 (TOC00) 的格式

地址: FFB DH 复位后: 00H R/W

符号	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
OSPT00	通过软件控制的单脉冲输出触发器							
0	-							
1	单脉冲输出							
该位在读取时的值始终为“0”。在除单脉冲输出模式外的其他模式下不要将该位设为 1。如果该位被设为 1，那么 TM00 将会被清除并且启动。								
OSPE00	单脉冲输出操作控制							
0	连续脉冲输出							
1	单脉冲输出							
单脉冲输出在自由运行定时器模式或通过 TI000 引脚有效沿进入的清除&开始模式中正确进行。单脉冲不能在 TM00 与 CR000 匹配的情况下进入的清除&开始模式中输出。								
TOC004	在 CR010 和 TM00 匹配时的 TO00 引脚输出控制							
0	禁止反转操作							
1	允许反转操作							
即使当 TOC004 = 0 时也会生成中断信号 (INTTM010)。								
LVS00	LVR00	TO00 引脚输出状态的设置						
0	0	不变						
0	1	TO00 引脚输出的初始值为低电平 (TO00 引脚输出被清除为 0)。						
1	0	TO00 引脚输出的初始值为高电平 (TO00 引脚输出被设为 1)。						
1	1	禁止设置						
<ul style="list-style-type: none"> LVS00 和 LVR00 可以用于设置 TO00 引脚输出电平的初始值。如果初始值不用进行设置，则将 LVS00 和 LVR00 保持为 00。 当 TOE00 = 1 时请务必对 LVS00 和 LVR00 进行设置。禁止同时将 LVS00, LVR00, 以及 TOE00 设为 1。 LVS00 和 LVR00 为触发位。通过将这些位设置为 1 可以对 TO00 引脚输出电平的初始值进行设置。即使这些位被清除为 0, 也不会影响 TO00 引脚的输出。 当 LVS00 和 LVR00 被读取时, 它们的值始终为 0。 关于设置 LVS00 和 LVR00 的方法, 请参见 6.5.2 设置 LVS00 和 LVR00。 除 TO00 输出外, TO00/TI010/RxD6/P03 引脚的实际输出也是根据 PM03 和 P03 来决定的。 								
TOC001	在 CR000 和 TM00 匹配时的 TO00 引脚输出控制							
0	禁止反转操作							
1	允许反转操作							
即使当 TOC001 = 0 时也会生成中断信号 (INTTM000)。								
TOE00	TO00 输出控制							
0	禁止输出 (TO00 输出固定为低电平)							
1	允许输出							

(4) 预分频模式寄存器 00 (PRM00)

PRM00 是用于设置 TM00 计数时钟以及 TI000 和 TI010 引脚输入有效沿的寄存器。

禁止在操作（当 TMC003 和 TMC002 不等于 00）期间重写 PRM00。

PRM00 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该 PRM00 清除为 00H。

注意事项 1. 将 PRM001 和 PRM000 位设置为 11（用于将 TI000 引脚的有效沿指定为计数时钟）时不要应用以下设置。

- 通过 TI000 引脚有效沿进入清除&开始模式
 - 将 TI000 引脚设置为捕获触发器
2. 如果在 TI000 或 TI010 引脚处于高电平且 TI000 或 TI010 的有效沿被指定为上升沿或双边沿时允许 16 位定时器 / 事件计数器 00 的操作，那么 TI000 或 TI010 引脚的高电平将会被检测为上升沿。在 TI000 或 TI010 引脚上拉时需注意这一点。然而，当定时器操作被停止时将不能检测到上升沿，接着将再次允许 16 位定时器 / 事件计数器 00 的操作。
3. TI010 的有效沿与定时器输出 (TO00) 不能同时用于 P03 引脚。选择任一功能。

图 6-9. 预分频模式寄存器 00 (PRM00) 的格式

地址: FFBH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PRM00	ES101	ES100	ES001	ES000	0	0	PRM001	PRM000

ES101	ES100	TI010 引脚有效沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	双边沿

ES001	ES000	TI000 引脚有效沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	双边沿

PRM001	PRM000	计数时钟选择		
			$f_{PRS} = 2 \text{ MHz}$	$f_{PRS} = 4 \text{ MHz}$
0	0	f_{PRS}	2 MHz	4 MHz
0	1	$f_{PRS}/2^2$	500 kHz	1 MHz
1	0	$f_{PRS}/2^8$	7.81 kHz	15.625 kHz
1	1	TI000 有效沿 ^注		

注 TI000 引脚的外部时钟需要一个比周边硬件时钟 (f_{PRS}) 的两倍周期更长的脉冲。

备注 f_{PRS} : 周边硬件时钟频率

(5) 端口模式寄存器 0 (PM0)

该寄存器以 1 位为单位对端口 0 的输入 / 输出进行设置。

当定时器输出使用 P03/TO00/TI010/RxD6 引脚时, 应将 PM03 以及 P03 的输出锁存设置为 0。

当定时器输入使用 P02/TI000/TxD6 和 P03/TO00/TI010/RxD6 引脚时, 应将 PM02 和 PM03 设置为 1。此时, P02 和 P03 的输出锁存可能为 0 也可能为 1。

PM0 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将 PM0 设置为 FFH。

图 6-10. 端口模式寄存器 0 (PM0) 的格式

地址: FF20H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	P0n 引脚I/O 模式选择 (n = 0 到 7)
0	输出模式 (输出缓冲开)
1	输入模式 (输出缓冲关)

(6) 端口输出模式寄存器 (POM0)

该寄存器用于设置端口 0 的输出模式。

该寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将这些寄存器设置为 00H。

图 6-11. 端口输出模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
POM0	POM07	POM06	POM05	POM04	POM03	POM02	POM01	POM00	FF38H	00H	R/W

POM0n	P0n 引脚输出模式选择 (n = 0 到 7)
0	CMOS 输出
1	N-ch 漏极开路输出 (P07: P-ch 漏极开路输出)

6.4 16 位定时器 / 事件计数器 00 的操作

6.4.1 间隔定时器的操作

如果 16 位定时器模式控制寄存器 (TMC00) 的第 3 位和第 2 位 (TMC003 和 TMC002) 被设置为 11 (在 TM00 与 CR000 匹配时进入的清除&开始模式), 那么计数操作将会与计数时钟同步开始。

当 TM00 的值与 CR000 的值匹配时, TM00 将会被清除为 0000H, 并会生成一个匹配中断信号 (INTTM000)。该 INTTM000 信号允许 TM00 作为间隔定时器来进行操作。

- 备注**
1. 关于 I/O 引脚设置的详细信息, 请参见 6.3 (5) 端口模式寄存器 0 (PM0) 和 (6) 端口输出模式寄存器 (POM0)。
 2. 关于允许 INTTM000 中断的方法, 请参见第十一章 中断功能。

图 6-12. 间隔定时器操作的框图

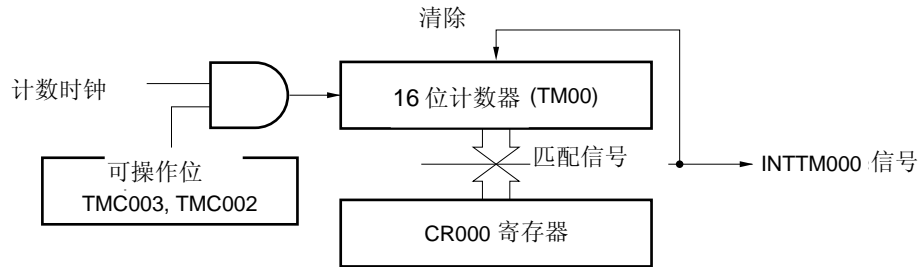


图 6-13. 间隔定时器操作的基本时序示例

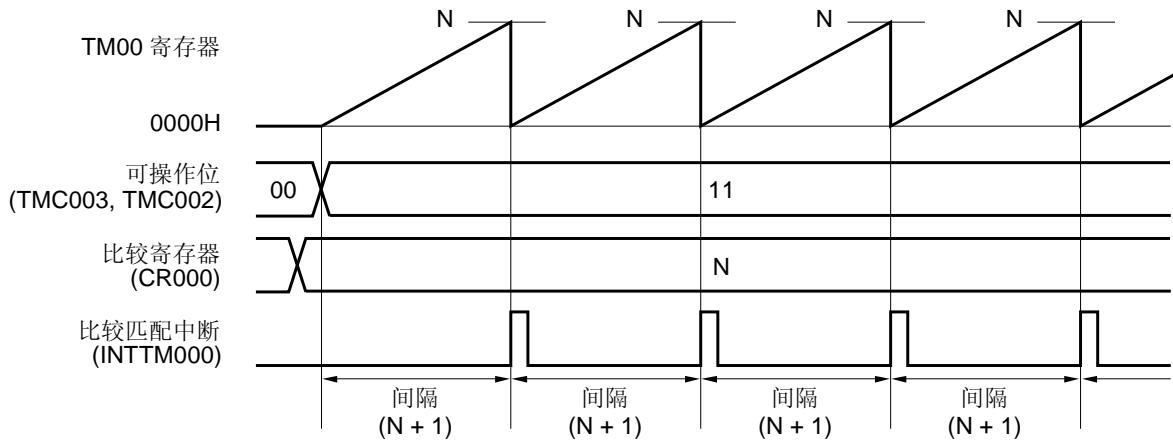
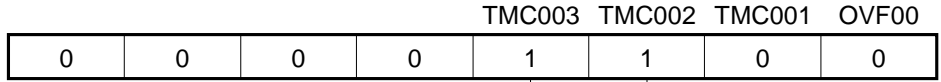


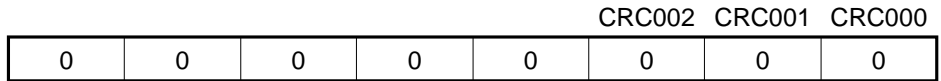
图 6-14. 间隔定时器操作的寄存器设置示例

(a) 16 位定时器模式控制寄存器 00 (TMC00)



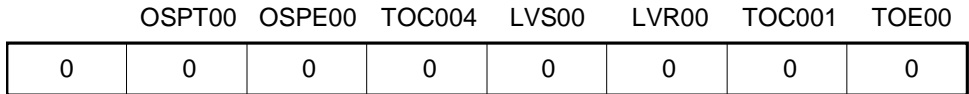
清除并在TM00和CR000匹配时启动。

(b) 捕获 / 比较控制寄存器 00 (CRC00)

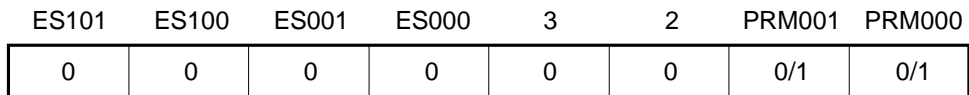


CR000 用作比较寄存器

(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)



选择计数时钟

(e) 16 位定时器计数器 00 (TM00)

计数值可以通过读取 TM00 来读取。

(f) 16 位捕获 / 比较寄存器 000 (CR000)

如果 M 被设置为 CR000，间隔时间如下。

- 间隔时间 = (M + 1) × 计数时钟周期

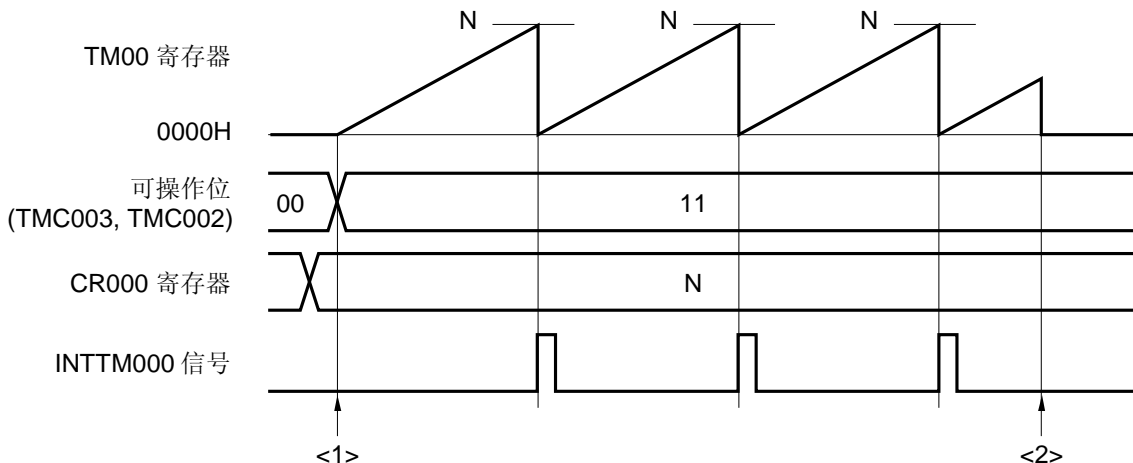
禁止将 CR000 设置为 0000H。

(g) 16 位捕获 / 比较寄存器 010 (CR010)

通常情况下 CR010 不会用于间隔定时器功能。然而，当 CR010 的设定值与 TM00 的值匹配时将会生成一个比较匹配中断 (INTTM010)。

因此，应通过使用中断屏蔽标志 (TMMK010) 来屏蔽中断请求。

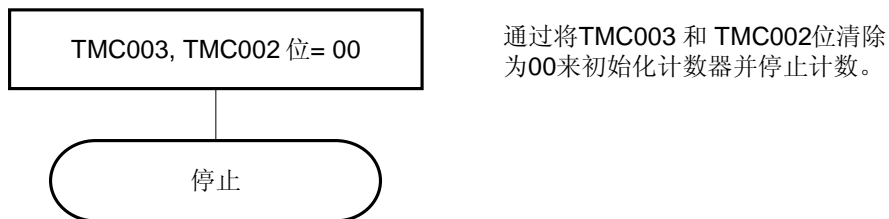
图 6-15. 间隔定时器操作的软件处理示例



<1> 计数操作开始流程



<2> 计数操作停止流程



6.4.2 方波输出操作

当 16 位定时器 / 事件计数器 00 用作间隔定时器时（见 6.4.1），通过将 16 位定时器输出控制寄存器 00（TOC00）设为 03H 可以从 TO00 引脚中输出一个方波。

当 TMC003 和 TMC002 被设为 11 时（TM00 和 CR000 匹配时进入的计数清除&开始模式），计数操作将会与计数时钟同步开始。

当 TM00 的值与 CR000 的值匹配时，TM00 将会被清除为 0000H，并会生成一个中断信号（INTTM000），而 TO00 引脚的输出也会被反转。在固定间隔反转的 TO00 引脚输出允许 TO00 输出一个方波。

- 备注**
1. 关于 I/O 引脚设置的详细信息，请参见 6.3（5）端口模式寄存器 0（PM0）和（6）端口输出模式寄存器（POM0）。
 2. 关于允许 INTTM000 信号中断的方法，请参见第十一章 中断功能。

图 6-16. 方波输出操作的框图

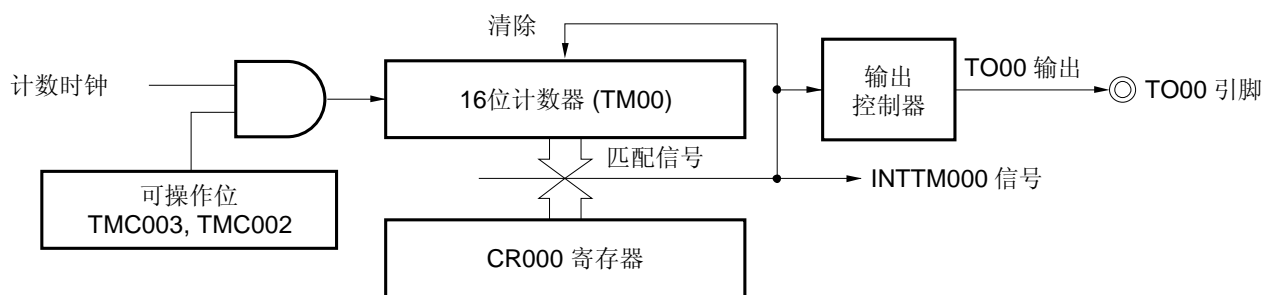


图 6-17. 方波输出操作的基本时序示例

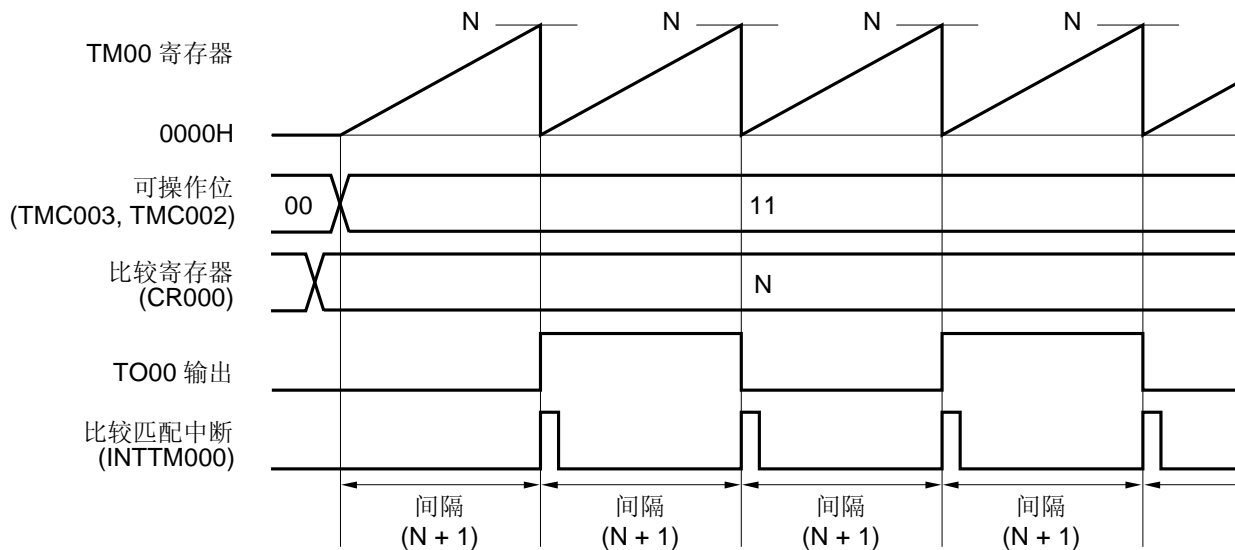
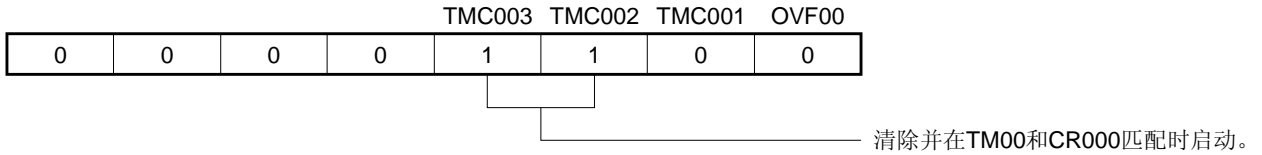
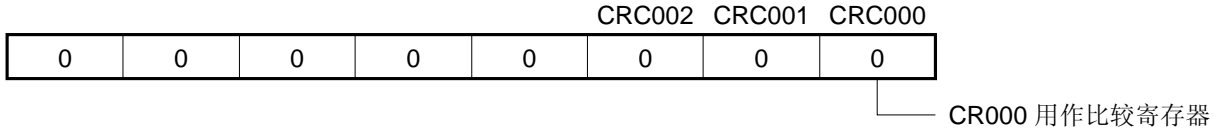


图 6-18. 方波输出操作的寄存器设置示例

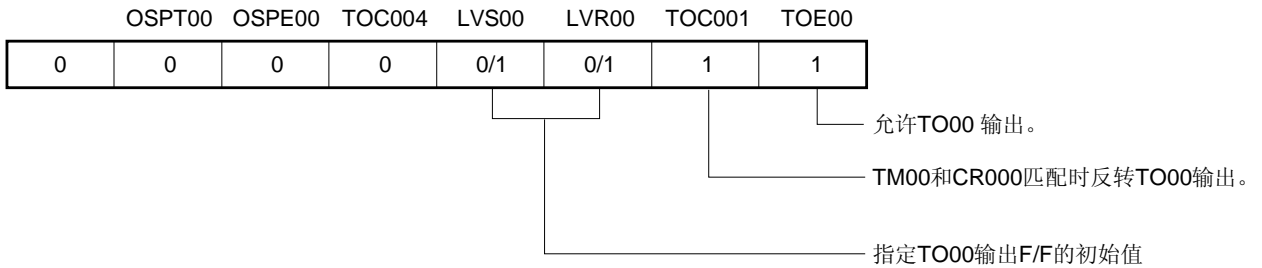
(a) 16 位定时器模式控制寄存器 00 (TMC00)



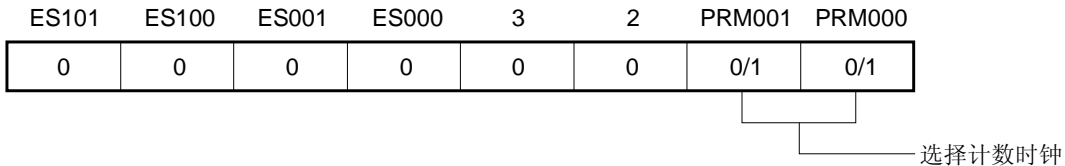
(b) 捕获 / 比较控制寄存器 00 (CRC00)



(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)



(e) 16 位定时器计数器 00 (TM00)

计数值可以通过读取 TM00 来读取。

(f) 16 位捕获 / 比较寄存器 000 (CR000)

如果 M 被设置为 CR000，间隔时间如下。

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

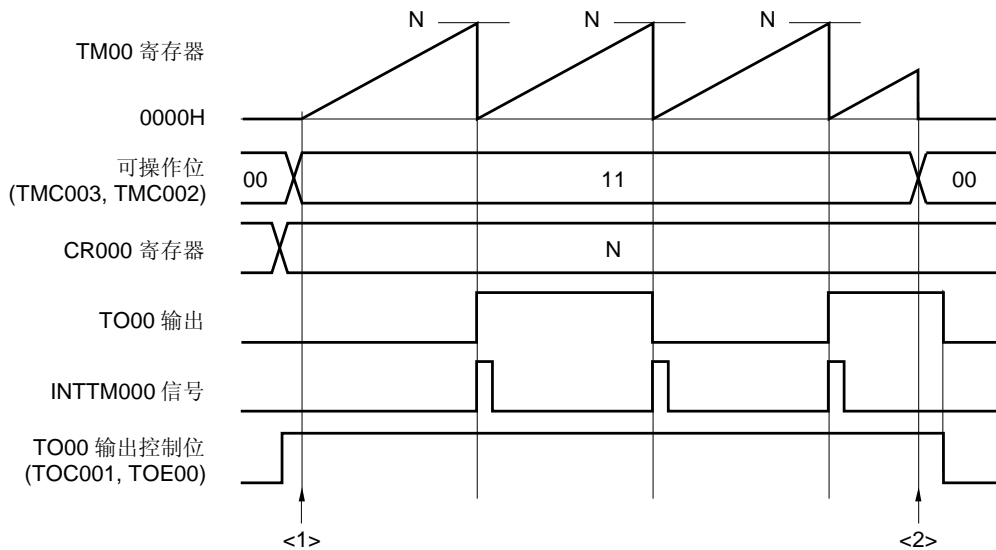
禁止将 CR000 设置为 0000H。

(g) 16 位捕获 / 比较寄存器 010 (CR010)

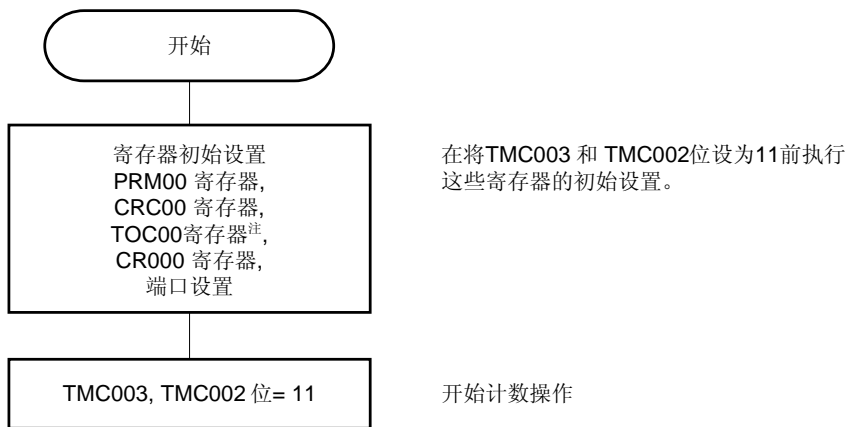
通常情况下 CR010 不会用于方波输出功能。然而，当 CR010 的设定值与 TM00 的值匹配时将会生成一个比较匹配中断 (INTTM010)。

因此，应通过使用中断屏蔽标志 (TMMK010) 来屏蔽中断请求。

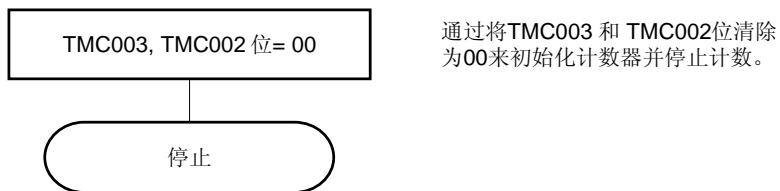
图 6-19. 方波输出功能的软件处理示例



<1> 计数操作开始流程



<2> 计数操作停止流程



注 对 TOC00 进行设置时必须谨慎。详细信息，请参见 6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

6.4.3 外部事件计数器操作

当预分频模式寄存器 00 (PRM00) 的第 1 位和 0 位 (PRM001 和 PRM000) 被设为 11 (使用 TI000 引脚的有效沿进行计数) 且 16 位定时器模式控制寄存器 00 (TMC00) 的第 3 位和第 2 位 (TMC003 和 TMC002) 被设为 11 时, 将会对外部事件输入的有效沿进行计数, 并会生成一个表示 TM00 与 CR000 (INTTM000) 匹配的匹配中断信号。

若要输入外部事件, 则应使用 TI000 引脚。因此在通过 TI000 引脚有效沿输入 (TMC003 和 TMC002 = 10 时) 进入的清除&开始模式中不能将定时器 / 事件计数器用作外部事件计数器。

INTTM000 信号通过以下时序来生成。

- 生成 INTTM000 信号的时序 (第二次或第二次以后)
= 外部事件有效沿的检测次数 × (CR000 的设定值 + 1)

然而, 定时器 / 计数器开始操作后的第一次匹配中断会按以下时序生成。

- 生成 INTTM000 信号的时序 (仅限第一次)
= 外部事件输入有效沿的检测次数 × (CR000 的设定值 + 2)

若要检测有效沿, 则在 fPRS 的时钟周期内对输入到 TI000 引脚中的信号进行采样。有效沿只有在连续被检测出两次时才能被检测到。因此, 可以消除带有短脉宽的噪声。

- 备注**
1. 关于 I/O 引脚设置的详细信息, 请参见 6.3 (5) 端口模式寄存器 0 (PM0)。
 2. 关于允许 INTTM000 信号中断的方法, 请参见第十一章 中断功能。

图 6-20. 外部事件计数器操作的框图

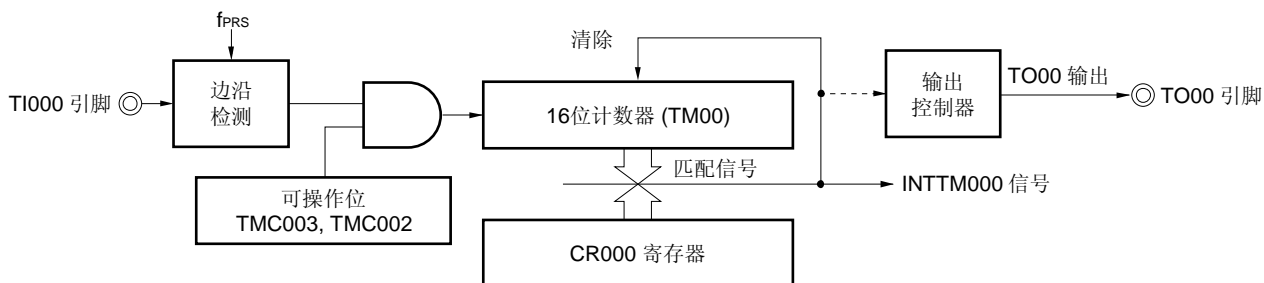
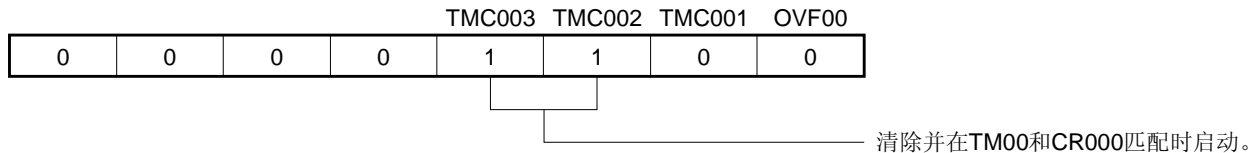
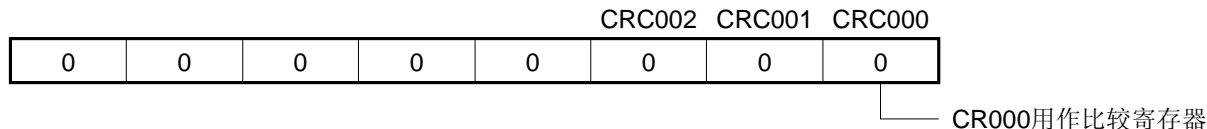


图 6-21. 外部事件计数器模式中的寄存器设置示例 (1/2)

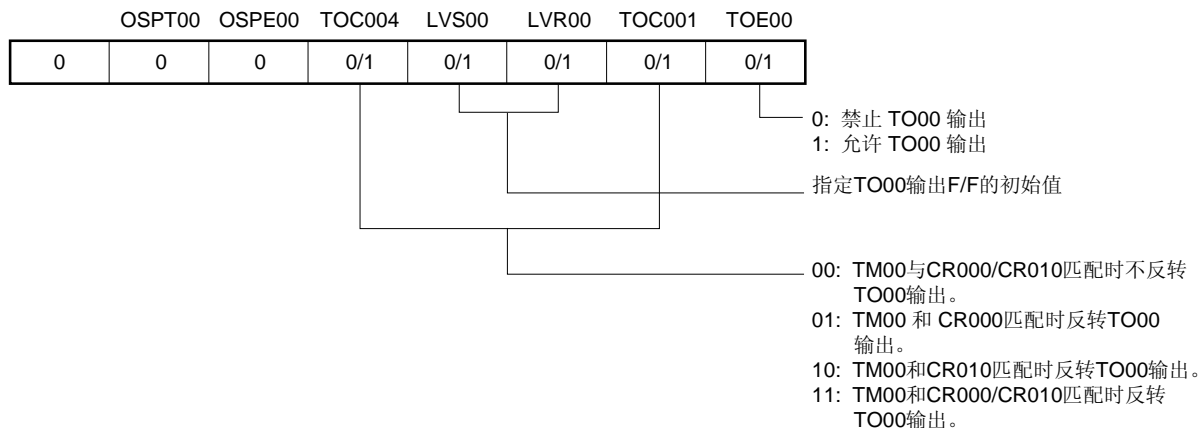
(a) 16 位定时器模式控制寄存器 00 (TMC00)



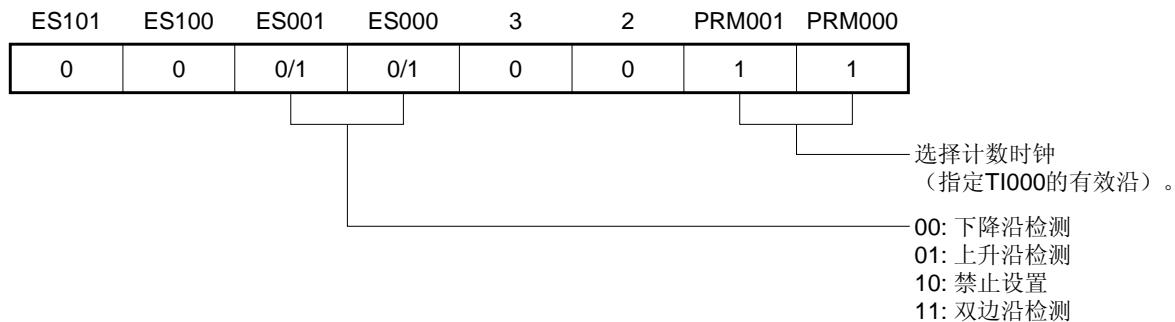
(b) 捕获 / 比较控制寄存器 00 (CRC00)



(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)



(e) 16 位定时器计数器 00 (TM00)

计数值可以通过读取 TM00 来读取。

(f) 16 位捕获 / 比较寄存器 000 (CR000)

如果 M 被设为 CR000, 那么在达到外部事件的个数 (M + 1) 时将会生成中断信号 (INTTM000)。
禁止将 CR000 设置为 0000H。

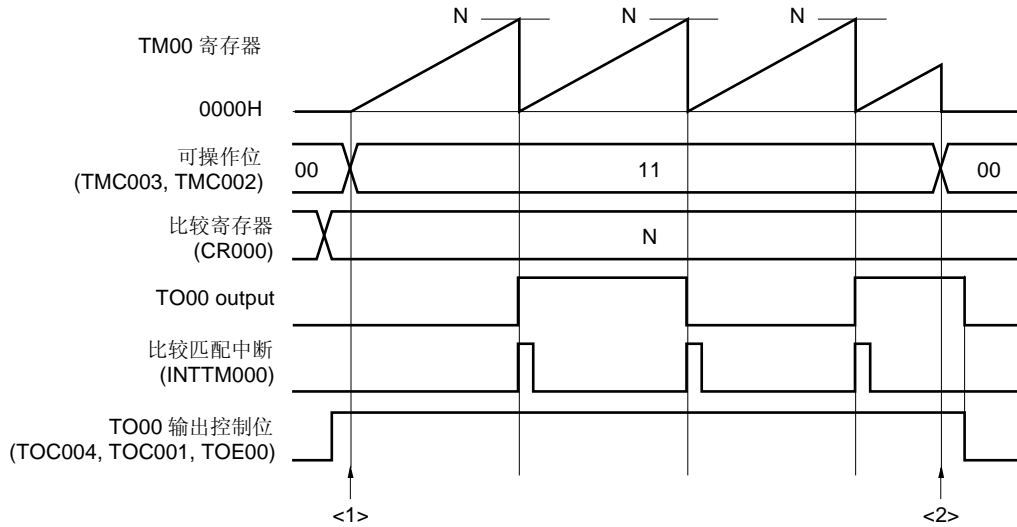
图 6-21. 外部事件计数器模式中的寄存器设置示例 (2/2)

(g) 16 位捕获 / 比较寄存器 010 (CR010)

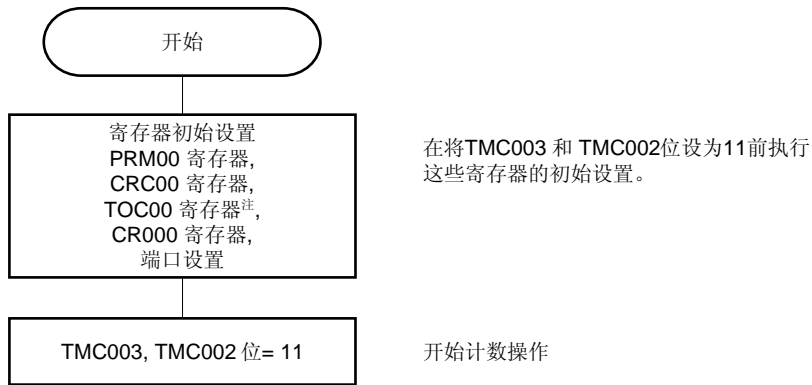
通常情况下，外部事件计数器模式中不使用 CR010 寄存器。然而，当 CR010 的设定值与 TM00 的值匹配时将会生成一个比较匹配中断 (INTTM010)。

因此，应通过使用中断屏蔽标志 (TMMK010) 来屏蔽中断请求。

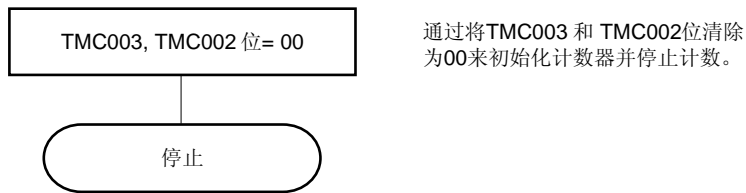
图 6-22. 外部事件计数模式的软件处理示例



<1> 计数操作开始流程



<2> 计数操作停止流程



注 对 TOC00 进行设置时必须需要谨慎。详细信息，请参见 6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

6.4.4 通过 TI000 引脚有效沿输入进入清除&开始模式的操作

当 16 位定时器模式控制寄存器 00 (TMC00) 的第 3 位和第 2 位 (TMC003 和 TMC002) 被设为 10 (通过 TI000 引脚有效沿输入进入清除&开始模式) 且计数时钟 (由 PRM00 设置的) 被提供给定时器 / 事件计数器时, TM00 将开始计数。在计数操作过程中检测到 TI000 引脚的有效沿时, TM00 将会被清除为 0000H 并再次开始计数。如果没有检测到 TI000 引脚的有效沿, 那么 TM00 将会溢出并继续计数。

TI000 引脚的有效沿的有效沿是用于清除 TM00 的原因。开始操作后不能立即控制计数器开始计数。

CR000 和 CR010 被用作比较寄存器和捕获寄存器。

(a) 当 CR000 和 CR010 被用作比较寄存器时

当 TM00 的值与 CR000 和 CR010 的值匹配时将会生成 INTTM000 和 INTTM010 信号。

(b) 当 CR000 和 CR010 被用作捕获寄存器时

当有效沿输入到 TI010 引脚时 (或当与有效沿的相位相反的相位被输入到 TI000 引脚中时) TM00 的计数值将会被捕获到 CR000 中且会生成 INTTM000 信号。

当有效沿被输入到 TI000 引脚中时, TM00 的计数值将会被捕获至 CR010 中并会生成 INTTM010 信号。一旦计数值被捕获, 计数器将会被清除为 0000H。

注意事项 不要将计数时钟设置为 TI000 引脚的有效沿 (PRM001 和 PRM000 = 11)。当 PRM001 和 PRM000 = 11 时, TM00 将会被清除。

备注 1. 关于 I/O 引脚设置的详细信息, 请参见 6.3 (5) 端口模式寄存器 0 (PM0)。

2. 关于允许 INTTM000 信号中断的方法, 请参见第十一章 中断功能。

(1) 通过 TI000 引脚有效沿输入进入清除&开始模式的操作

(CR000: 比较寄存器, CR010: 比较寄存器)

图 6-23. 通过 TI000 引脚有效沿输入进入清除&开始模式框图

(CR000: 比较寄存器, CR010: 比较寄存器)

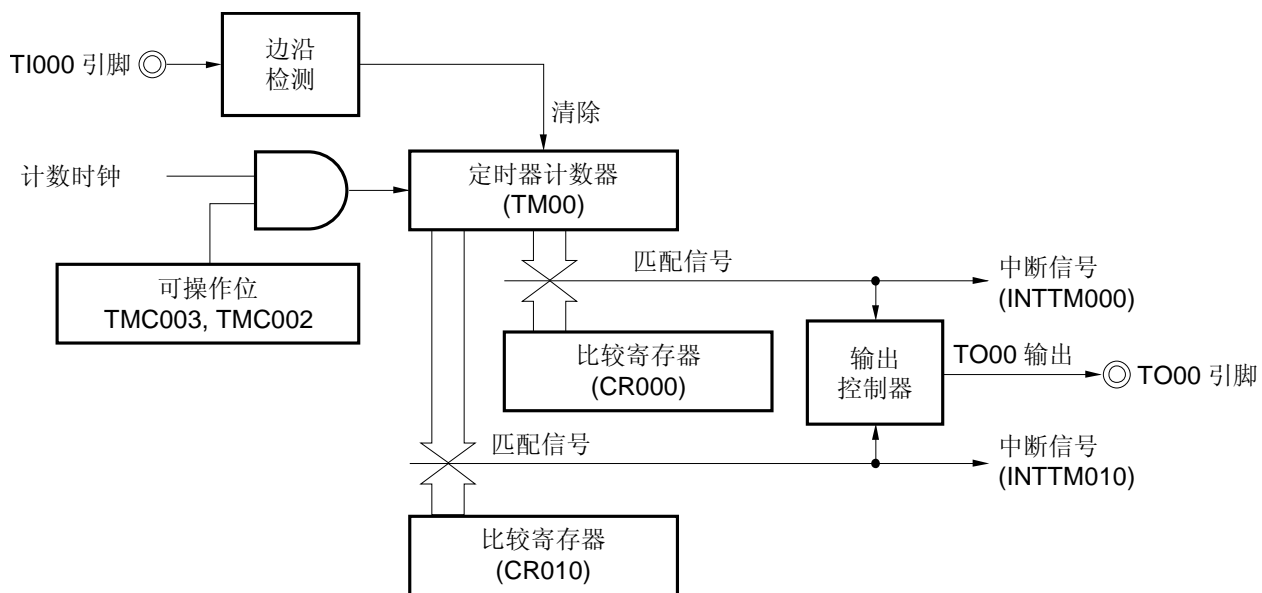
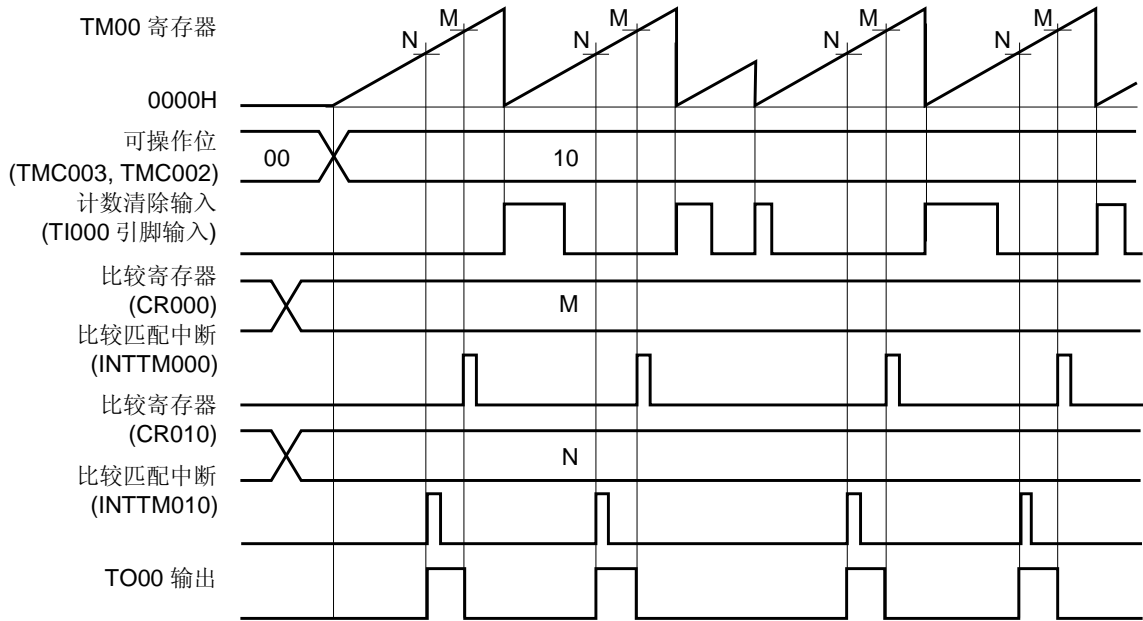
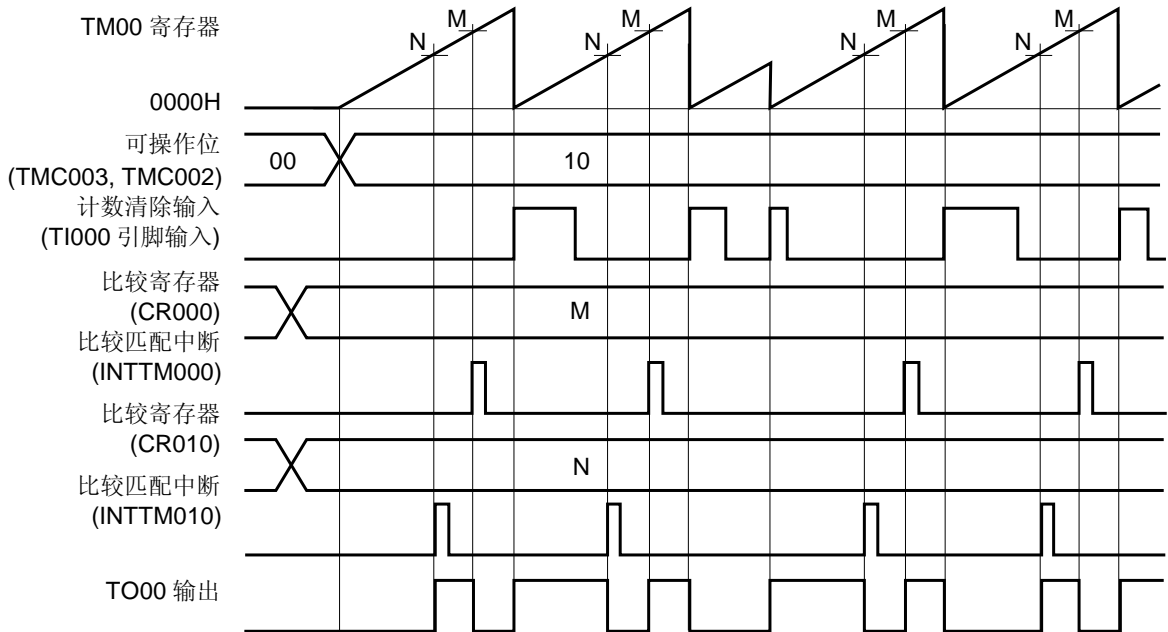


图 6-24. 通过 TI000 引脚有效沿输入进入的清除&开始模式的时序示例
(CR000: 比较寄存器, CR010: 比较寄存器)

(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 00H, TMC00 = 08H



(b) TOC00 = 13H, PRM00 = 10H, CRC00, = 00H, TMC00 = 0AH



根据 16 位定时器模式控制寄存器 01 (TMC00) 的第 1 位 (TMC001) 的不同设置, 如下所示, (a) 与 (b) 也不相同。

- (a) 当 TM00 与比较寄存器匹配时 TO00 引脚的输出电平被反转。
- (b) 当 TM00 与比较寄存器匹配或当 TI000 引脚的有效沿被检测到时 TO00 引脚的输出电平将会被反转。

(2) 通过 TI000 引脚有效沿输入进入清除&开始模式的操作
 (CR000: 比较寄存器, CR010: 捕获寄存器)

图 6-25. 通过 TI000 引脚有效沿输入进入清除&开始模式的框图
 (CR000: 比较寄存器, CR010: 捕获寄存器)

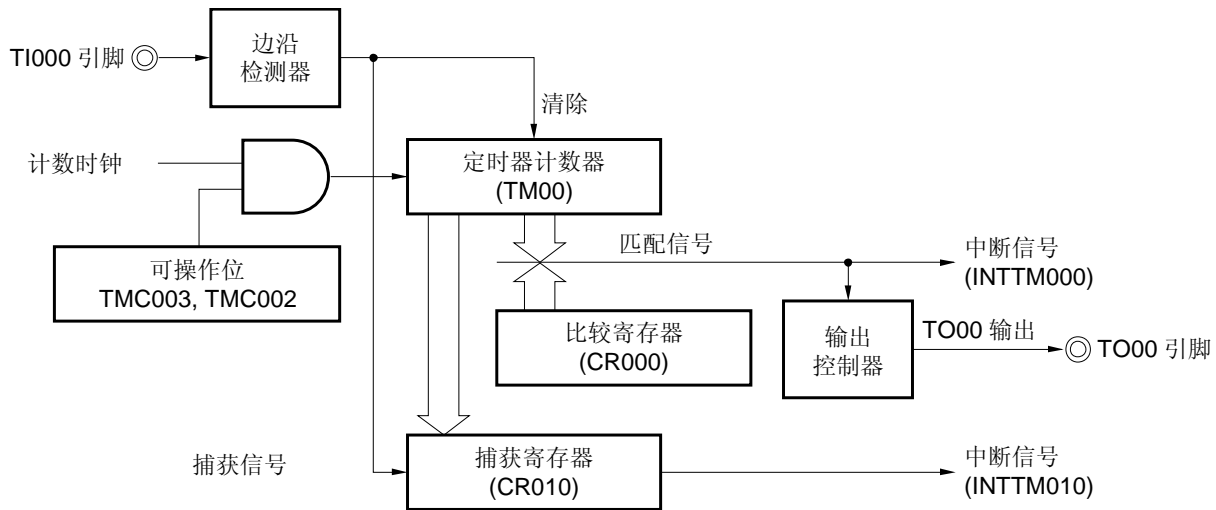
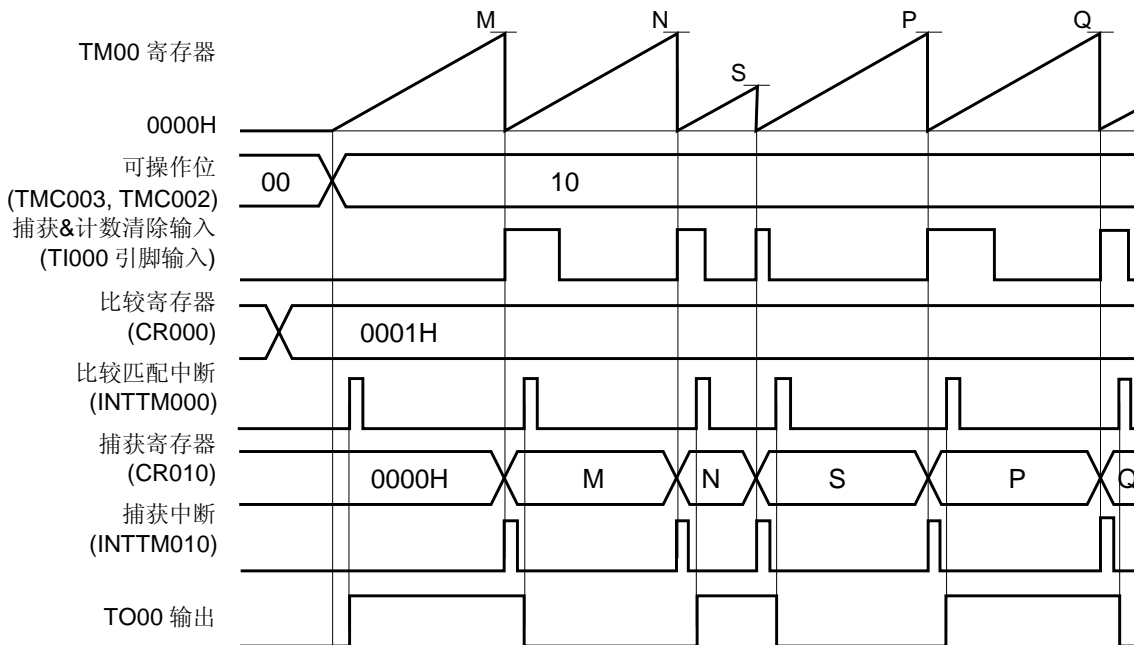


图 6-26. 通过 TI000 引脚有效沿输入进入的清除&开始模式的时序示例
 (CR000: 比较寄存器, CR010: 捕获寄存器) (1/2)

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

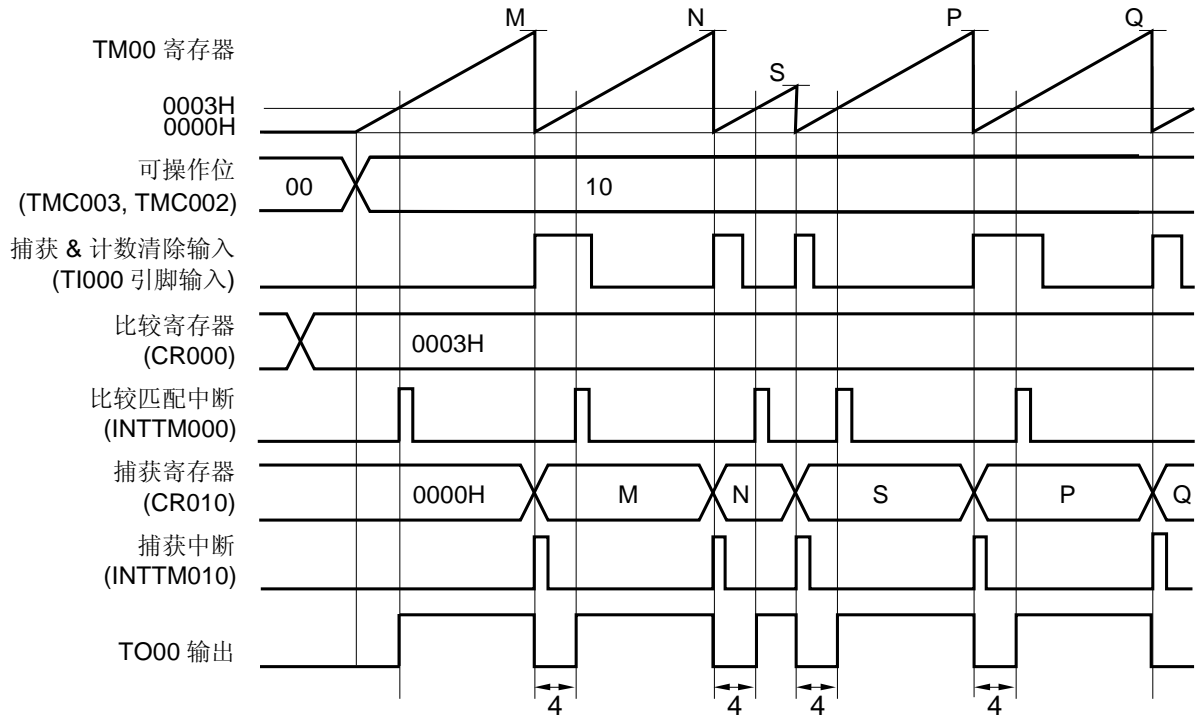


这是计数值被捕获&清除时 TO00 引脚的输出电平发生反转的应用示例。

当检测到 TI000 引脚的有效沿时, 计数值将会被捕获至 CR010 中且 TM00 也会被清除 (为 0000H)。当 TM00 的计数值为 0001H 时, 将会生成一个比较匹配中断信号 (INTTM00), 且 TO00 引脚的输出电平将会被反转。

图 6-26. 通过 TI000 引脚有效沿输入进入的清除&开始模式的时序示例
(CR000: 比较寄存器, CR010: 捕获寄存器) (2/2)

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



这是计数值被捕获&清除时赋值于 CR000 的宽度 (该示例中为 4 个时钟) 从 TO00 引脚中输出的应用示例。当检测到 TI000 引脚的有效沿时, 计数值将会被捕获至 CR010, 并会生成一个捕获中断信号 (INTTM010), TM00 也将会被清除 (为 0000H), 而 TO00 引脚的输出电平也将被反转。当 TM00 的计数值为 0003H 时 (已经计数了 4 个时钟), 将会生成一个比较匹配中断信号 (INTTM000), 且 TO00 引脚的输出电平将会被反转。

- (3) 通过 TI000 引脚有效沿输入进入清除&开始模式的操作
(CR000: 捕获寄存器, CR010: 比较寄存器)

图 6-27. 通过 TI000 引脚有效沿输入进入的清除&开始模式的框图
(CR000: 捕获寄存器, CR010: 比较寄存器)

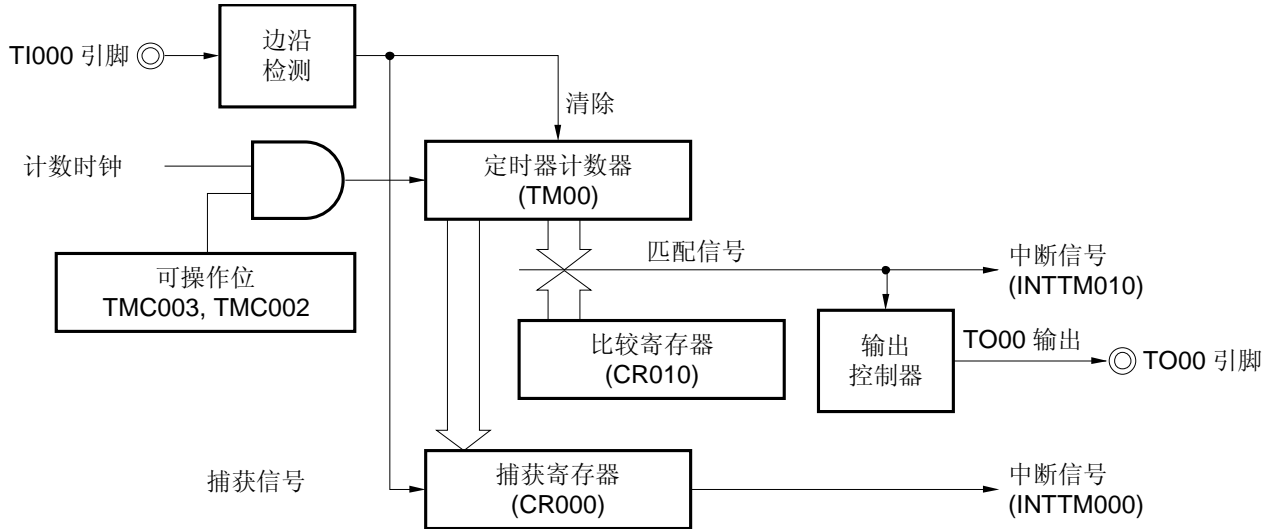
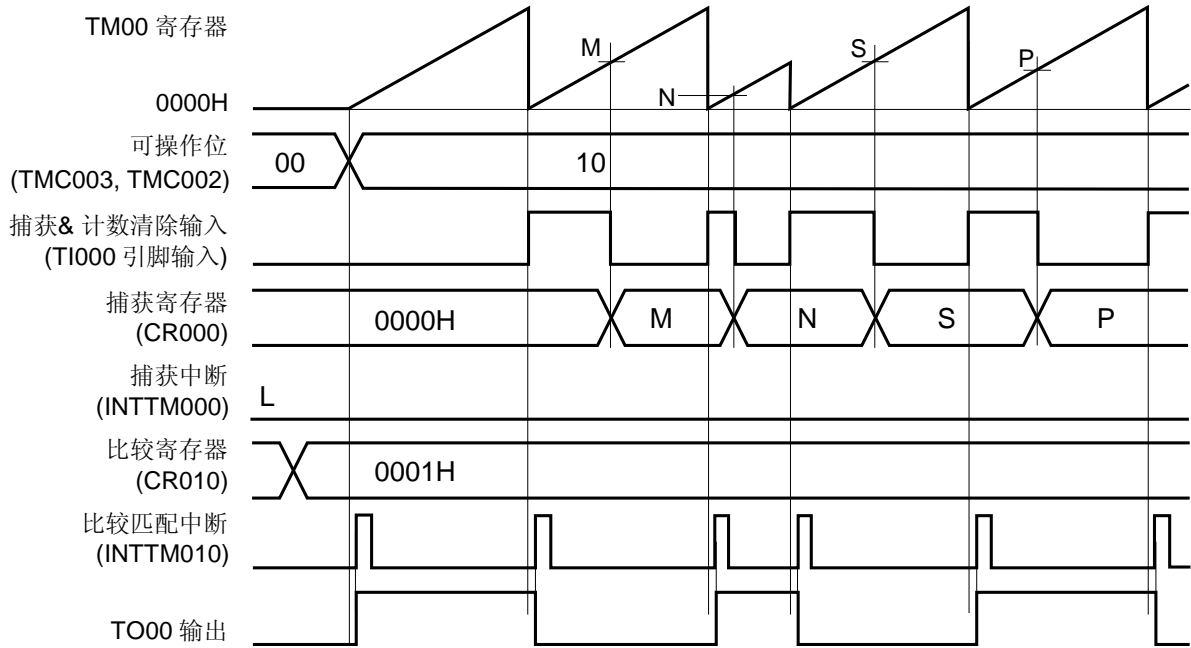


图 6-28. 通过 TI000 引脚有效沿输入进入清除&开始模式的时序示例
(CR000: 捕获寄存器, CR010: 比较寄存器) (1/2)

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



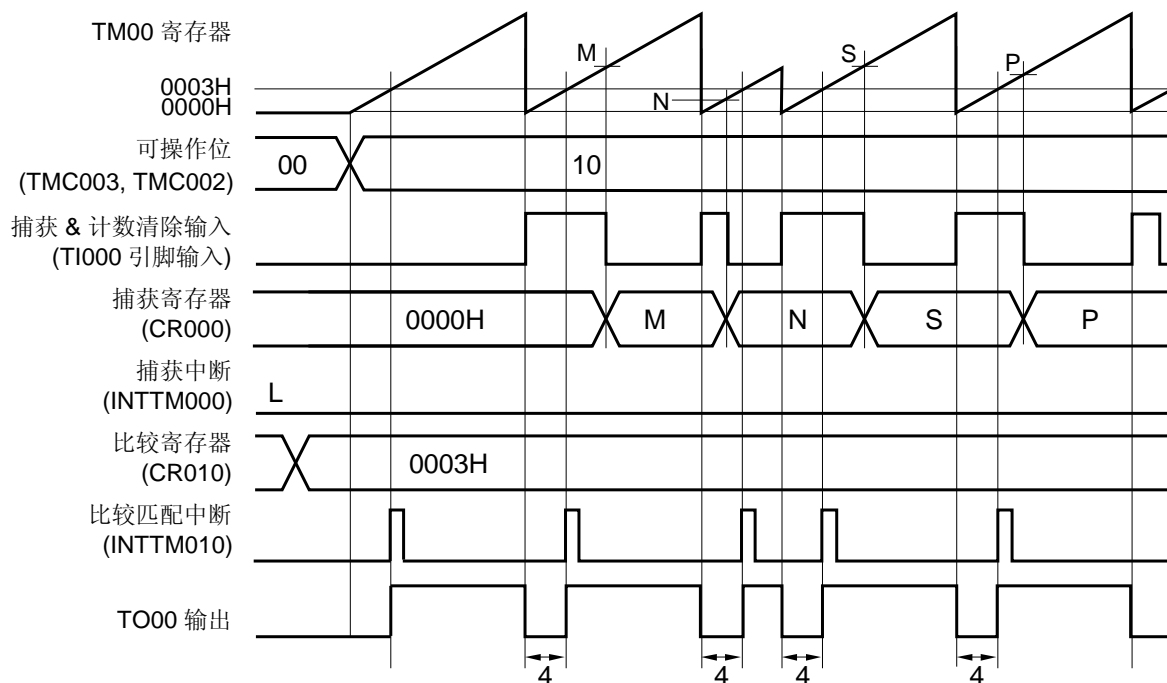
这是计数值被捕获&清除时 TO00 引脚的输出电平发生反转的应用示例。

TM00 在 TI000 引脚上升沿检测时被清除，并会在 TI000 引脚的下降沿检测时被捕获至 CR000。

当捕获 / 比较控制寄存器 00 (CRC00) 的第 1 位 (CRC001) 被设为 1 时，TM00 的计数值将会在与 TI000 引脚信号输入的相位相反的相位中被捕获至 CR000 中，但不会生成捕获中断信号 (INTTM000)。然而，在检测到 TI010 引脚的有效沿时将会生成 INTTM000 信号。在不使用 INTTM000 信号时将其屏蔽。

图 6-28. 通过 TI000 引脚有效沿输入进入清除&开始模式的时序示例
(CR000: 捕获寄存器, CR010: 比较寄存器) (2/2)

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



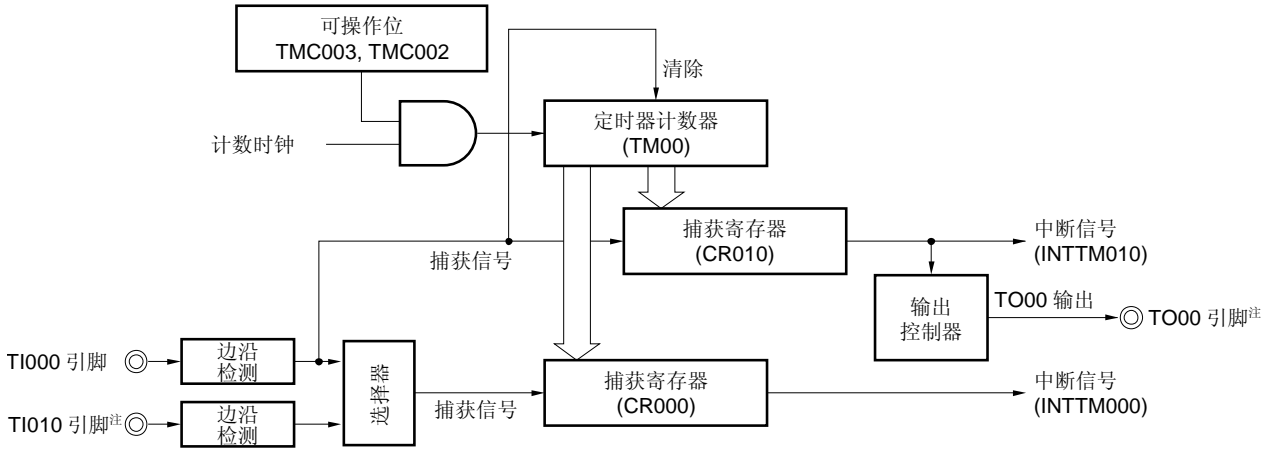
这是计数值被捕获&清除时赋值于 CR010 的宽度 (该示例中为 4 个时钟) 从 TO00 引脚中输出的应用示例。

TM00 在 TI000 引脚上升沿检测时被清除 (为 0000H), 并会在 TI000 引脚的下降沿检测时被捕获至 CR000。当 TM00 被清除 (为 0000H) 时 TO00 引脚的输出电平将会被反转, 这可能是因为检测到 TI000 引脚的上升沿, 也可能是因为 TM00 的值与比较寄存器 (CR010) 的值匹配。

当捕获 / 比较控制寄存器 00 (CRC00) 的第 1 位 (CRC001) 被设为 1 时, TM00 的计数值将会在与 TI000 引脚输入信号的相位相反的相位中被捕获至 CR000 中, 但不会生成捕获中断信号 (INTTM000)。然而, 在检测到 TI010 引脚的有效沿时将会生成 INTTM000 中断。在不使用 INTTM000 信号时将其屏蔽。

(4) 通过 TI000 引脚有效沿输入进入清除&开始模式的操作
(CR000: 捕获寄存器, CR010: 捕获寄存器)

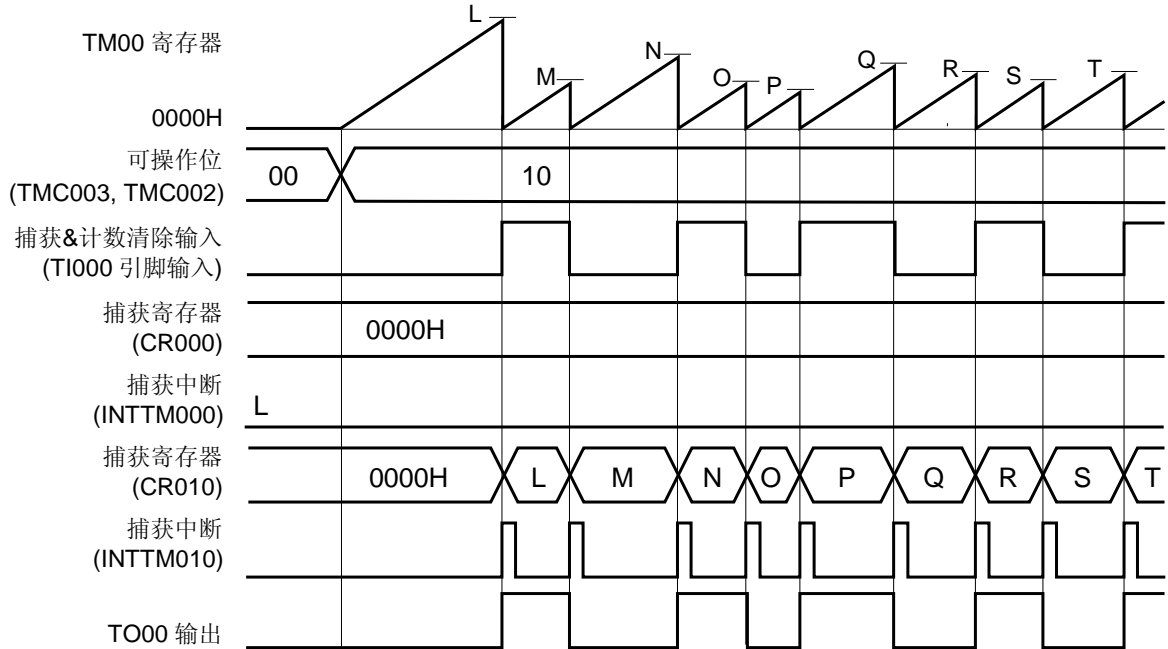
图 6-29. 通过 TI000 引脚有效沿输入进入清除&开始模式的框图
(CR000: 捕获寄存器, CR010: 捕获寄存器)



注 在检测 TI010 引脚的有效沿时不能使用定时器输出 (TO00)。

图 6-30. 通过 TI000 引脚有效沿输入进入清除&开始模式的时序示例
(CR000: 捕获寄存器, CR010: 捕获寄存器) (1/3)

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

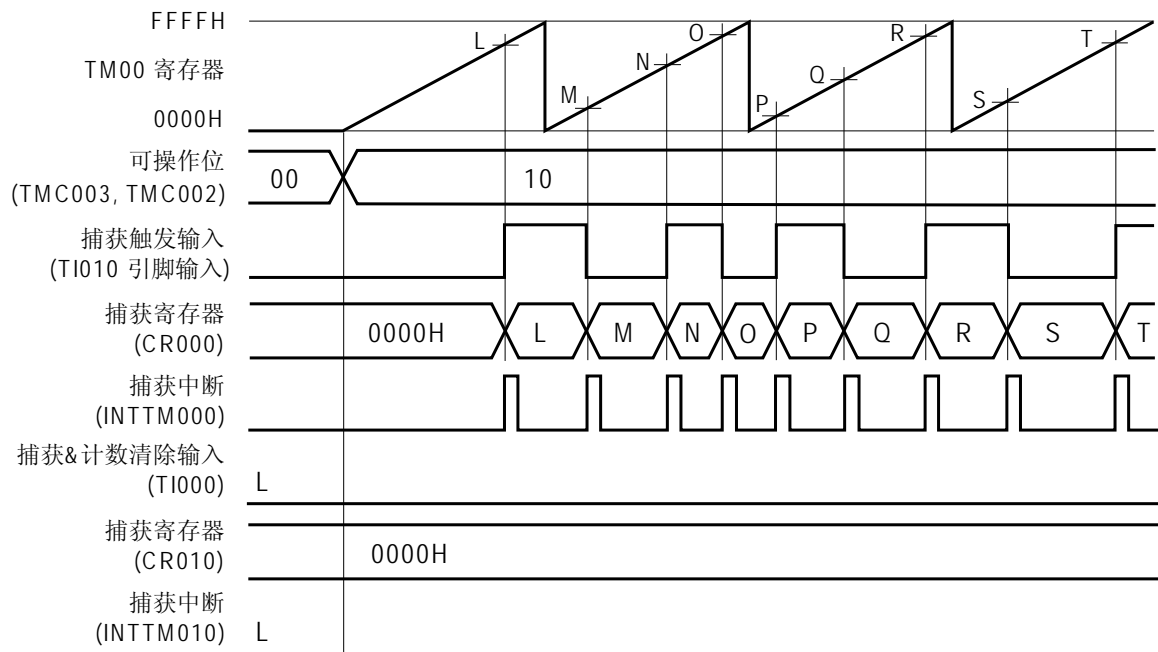


在该示例中，当检测到 TI000 引脚的上升沿或下降沿时，计数值将会被捕获至 CR010，TM00 将被清除，而 TO00 引脚输出也将被反转。

当检测到 TI010 引脚的边沿时，将会生成一个中断信号 (INTTM000)。在不使用 INTTM000 信号时将其屏蔽。

图 6-30. 通过 TI000 引脚有效沿输入进入清除&开始模式的时序示例
 (CR000: 捕获寄存器, CR010: 捕获寄存器) (2/3)

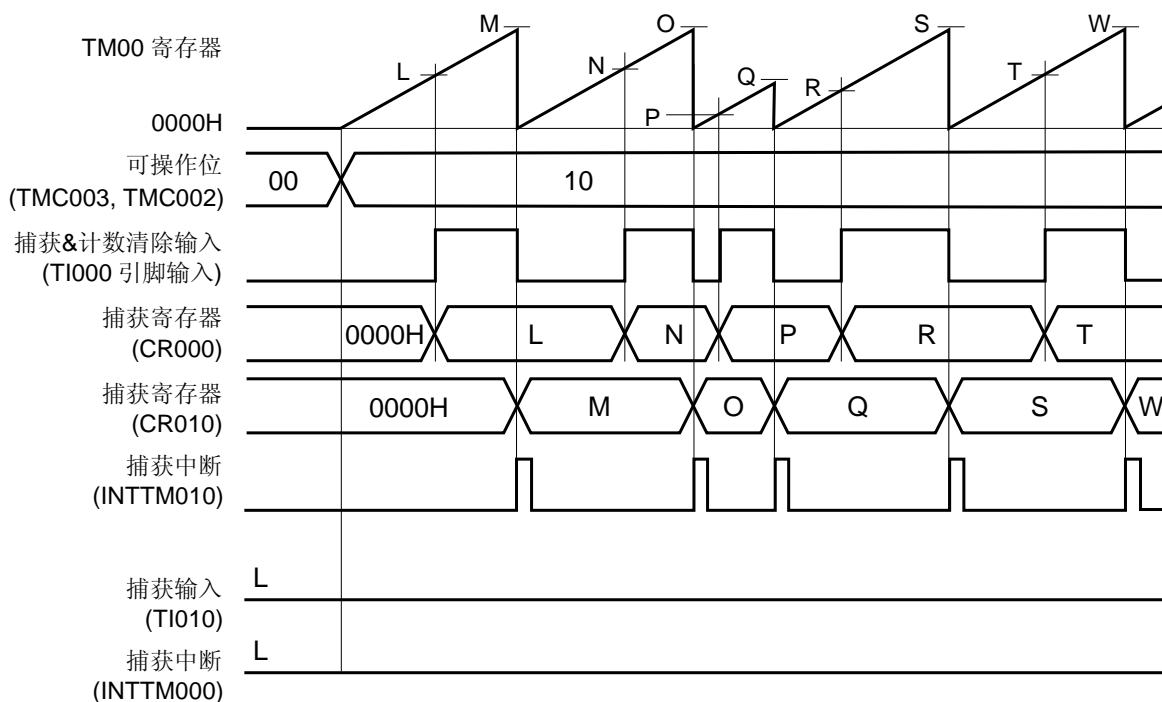
(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 0AH



在该时序示例中，没有边沿被输入到 TI000 引脚中，但在应该示例中，当检测到 TI010 引脚的上升沿或下降时，计数值被捕获至 CR000。

图 6-30. 通过 TI000 引脚有效沿输入进入清除&开始模式的时序示例
(CR000: 捕获寄存器, CR010: 捕获寄存器) (3/3)

(c) TOC00 = 13H, PRM00 = 00H, CRC00 = 07H, TMC00 = 0AH



该应用示例中，输入到 TI000 引脚中的信号脉宽被测量。

通过设置 CRC00，计数值可以在与 TI000 引脚的下降沿相位相反的相位（也就是上升沿）中被捕获至 CR000 以及 TI000 引脚上升沿处的 CR010 中。

可以通过以下公式来计算输入脉冲的高电平宽度和低电平宽度。

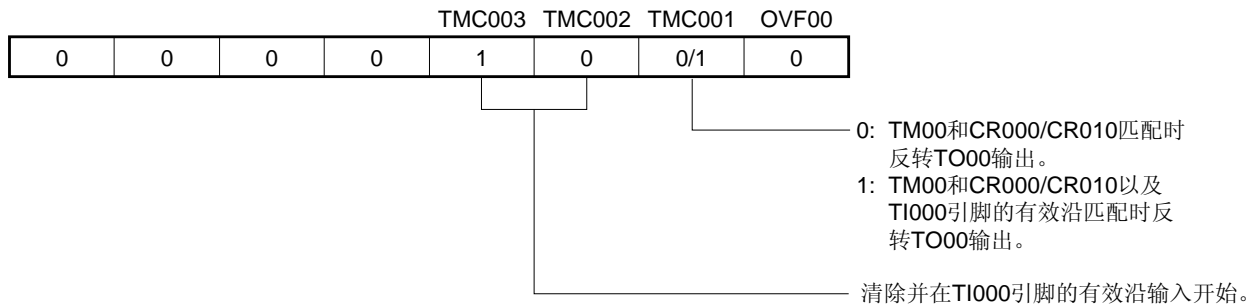
- 高电平宽 = [CR010 值] - [CR000 值] × [计算时钟周期]
- 低电平宽 = [CR000 值] × [计算时钟周期]

如果将 TI000 引脚的反向选作触发来将计数值捕获至 CR000，那么将不能生成 INTTM000 信号。在生成 INTTM010 信号后立即读取 CR000 和 CR010 的值以测量脉宽。

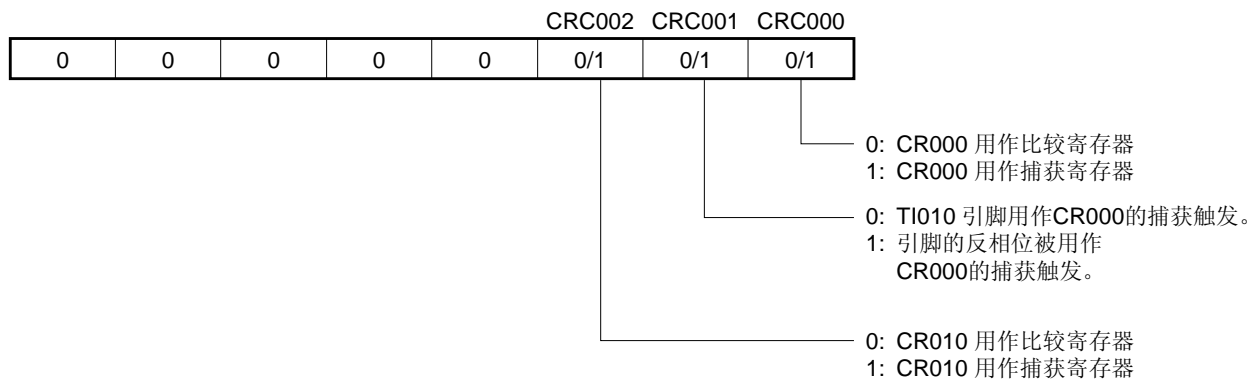
然而，如果由预分频模式寄存器 00 (PRM00) 的第 6 位和第 5 位 (ES101 和 ES100) 指定的有效沿被输入至 TI010 引脚中，那么将不能捕获计数值但会生成 INTTM000 信号。若要测量 TI000 引脚的脉宽，则在不使用 INTTM000 信号时将其屏蔽。

图 6-31. 通过 TI000 引脚有效沿输入进入清除&开始模式的寄存器设置示例 (1/2)

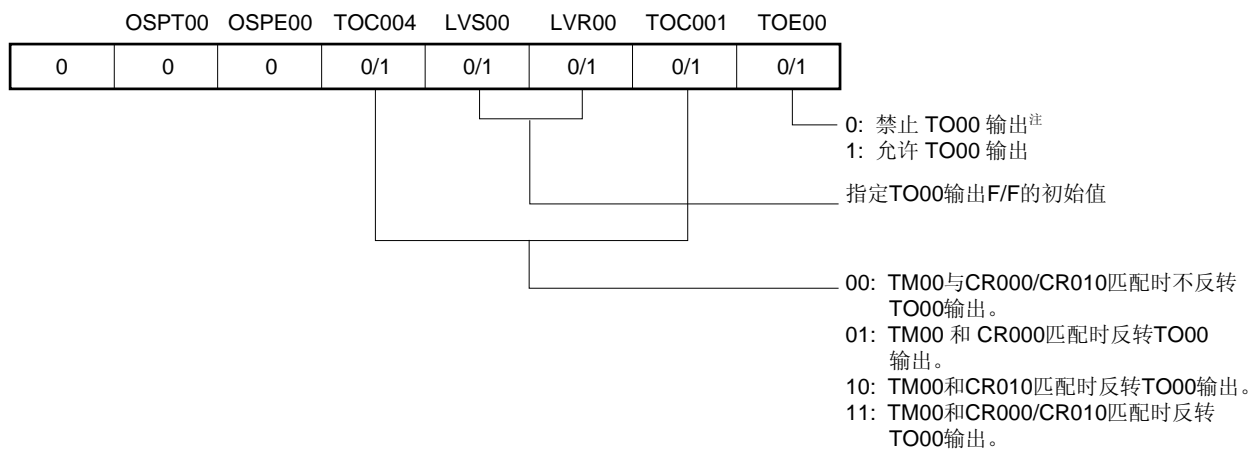
(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕获 / 比较控制寄存器 00 (CRC00)



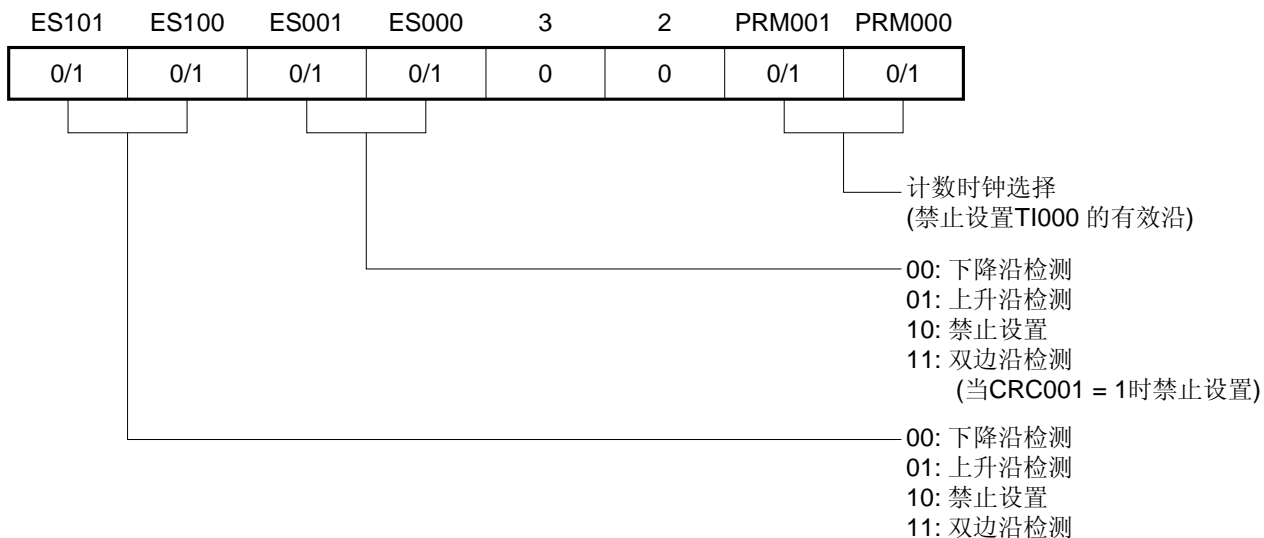
(c) 16 位定时器输出控制寄存器 00 (TOC00)



注 在检测 TI010 引脚的有效沿时不能使用定时器输出 (TO00)。

图 6-31. 通过 TI000 引脚有效沿输入进入清除&开始模式的寄存器设置示例 (2/2)

(d) 预分频模式寄存器 00 (PRM00)



(e) 16 位定时器计数器 00 (TM00)

计数值可以通过读取 TM00 来读取。

(f) 16 位捕获 / 比较寄存器 000 (CR000)

当该寄存器用作比较寄存器且该寄存器的值与 TM00 的计数值匹配时，将会生成一个中断信号 (INTTM000)。TM00 的计数值不被清除。

若要将该寄存器用作捕获寄存器，则应将 TI000 或 TI010 引脚*选作捕获触发器。当检测到捕获触发器的有效沿时，TM00 的计数值将会被保存在 CR000 中。

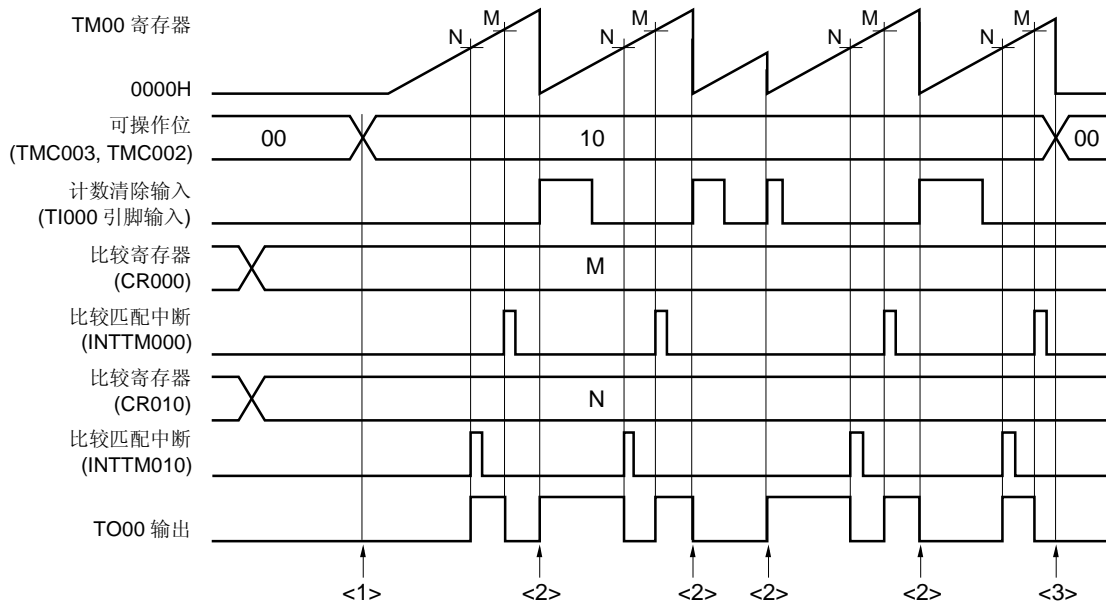
注 在检测 TI010 引脚的有效沿时不能使用定时器输出 (TO00)。

(g) 16 位捕获 / 比较寄存器 010 (CR010)

当该寄存器用作比较寄存器且该寄存器的值与 TM00 的计数值匹配时，将会生成一个中断信号 (INTTM010)。TM00 的计数值不被清除。

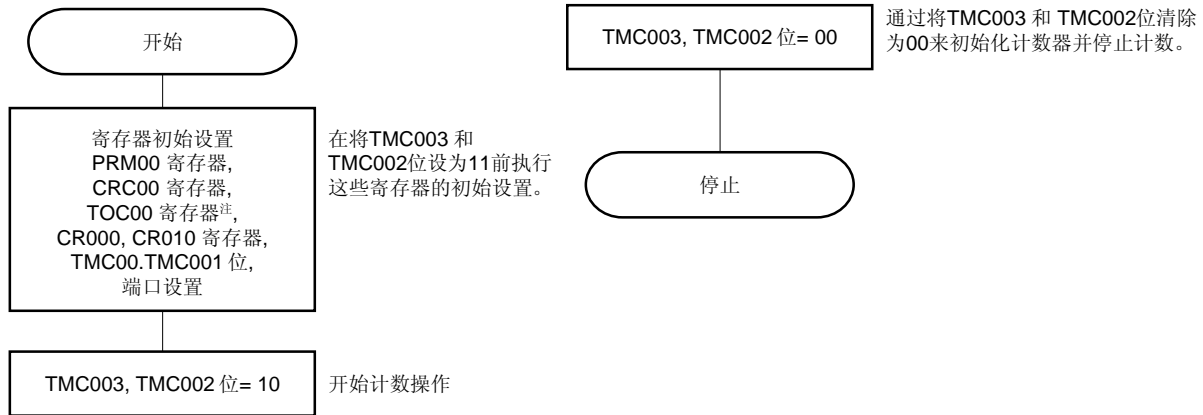
当该寄存器被用作捕获寄存器时，TI000 引脚输入将会被用作捕获触发器。当检测到捕获触发器的有效沿时，TM00 的计数值将会被保存在 CR010 中。

图 6-32. 通过 TI000 引脚有效沿输入进入清除&开始模式的软件处理示例

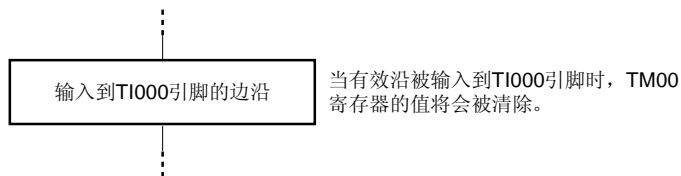


<1> 计数操作开始流程

<3> 计数操作停止流程



<2> TM00寄存器清除&开始流程



注 对 TOC00 进行设置时必须谨慎。详细信息，请参见 6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

6.4.5 自由运行定时器操作

当 16 位定时器模式控制寄存器 00 (TMC00) 的第 3 位和第 2 位被设置为 01 (自由选择定时器模式) 时, 16 位定时器 / 事件计数器 00 会继续与计数时钟同步计数。当已经计数至 FFFFH 时, 溢出标志 (OVF00) 会在下个时钟被设置为 1, 且 TM00 将会被清除 (为 0000H) 并继续计数。通过软件执行 CLR 指令, 用来将 OVF00 清除为 0。

可以使用以下三种类型的自由运行定时器操作。

- CR000 和 CR010 均被用作比较寄存器时。
- CR000 或 CR010 中的一个寄存器被用作比较寄存器, 另一个被用作捕获寄存器。
- CR000 和 CR010 均被用作捕获寄存器。

备注 1. 关于 I/O 引脚设置的详细信息, 请参见 6.3 (5) 端口模式寄存器 0 (PM0) 和 (6) 端口输出模式寄存器 (POM0)。

2. 关于允许 INTTM000 信号中断的方法, 请参见第十一章 中断功能。

(1) 自由运行定时器模式操作

(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)

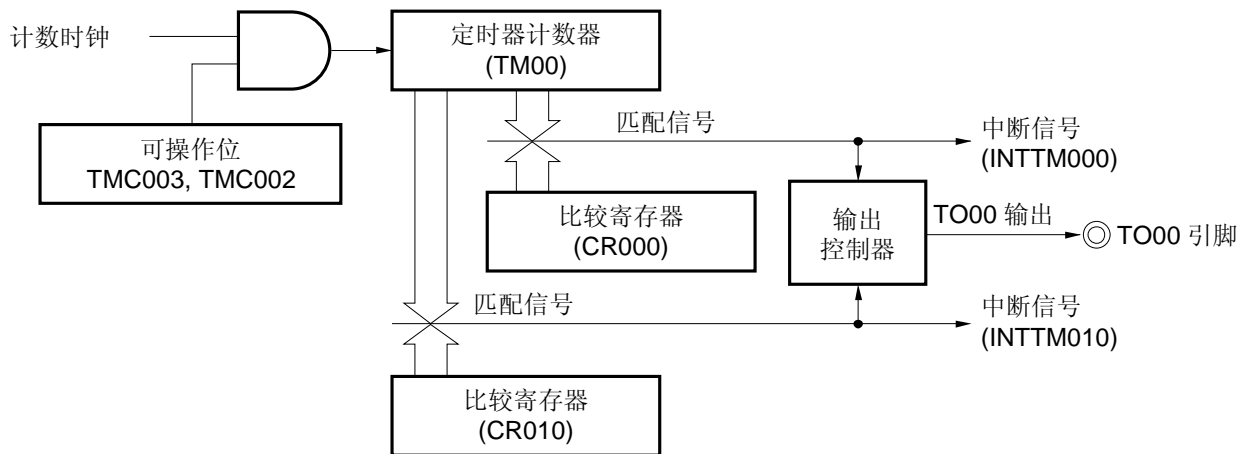
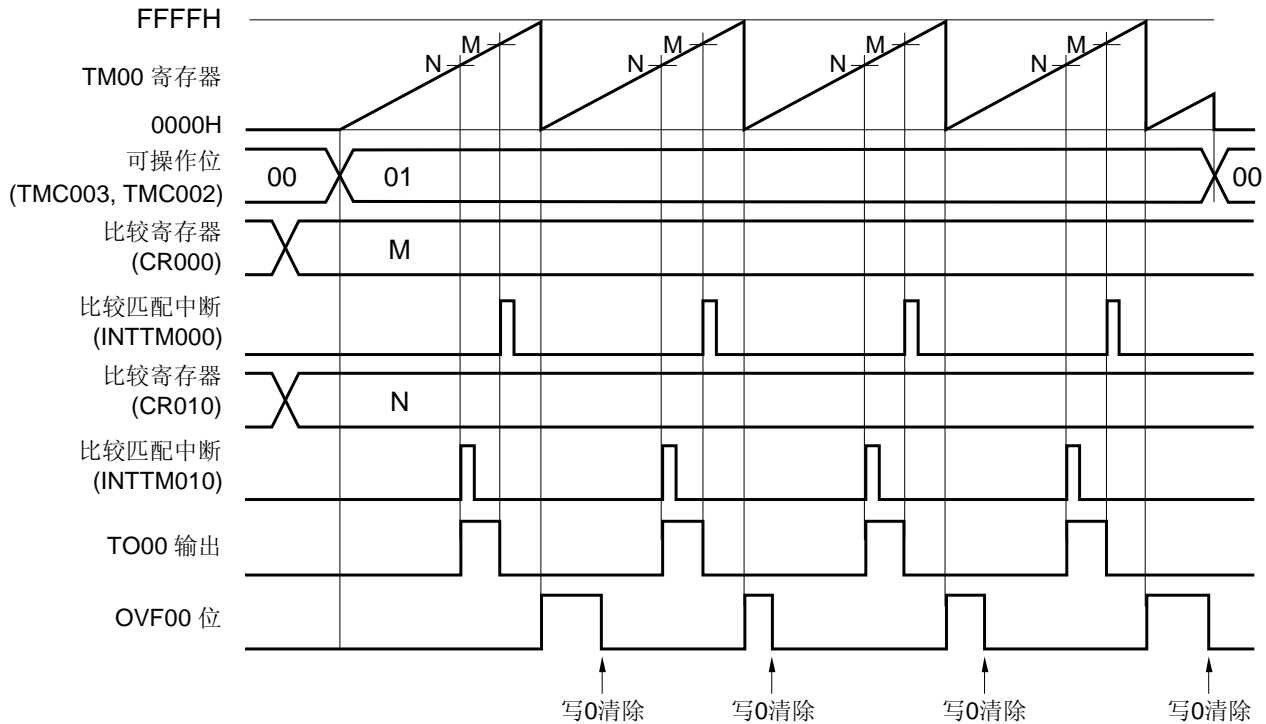


图 6-34. 自由运行定时器模式的时序示例
(CR000: 比较寄存器, CR010: 比较寄存器) (2/2)

• TOC00 = 13H, PRM00 = 00H, CRC00 = 00H, TMC00 = 04H



该应用示例中，两个比较寄存器均在自由运行定时器模式中使用。

每次 TM00 的计数值与 CR000 或 CR010 的设定值匹配时，TO00 引脚的输出电平都会被反转。当计数值与寄存器值匹配时，将会生成 INTTM000 或 INTTM010 信号。

(2) 自由运行定时器模式操作
(CR000: 比较寄存器, CR010: 捕获寄存器)

图 6-35. 自由运行定时器模式的框图
(CR000: 比较寄存器, CR010: 捕获寄存器)

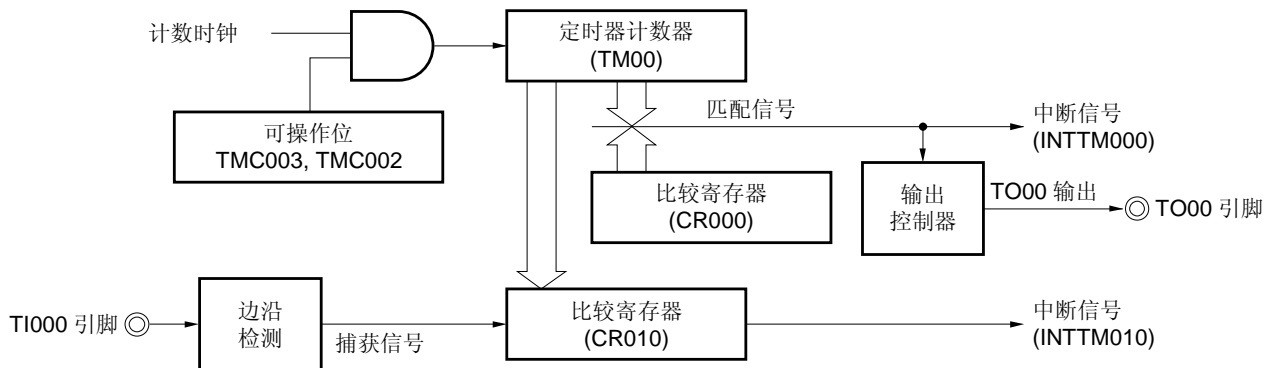
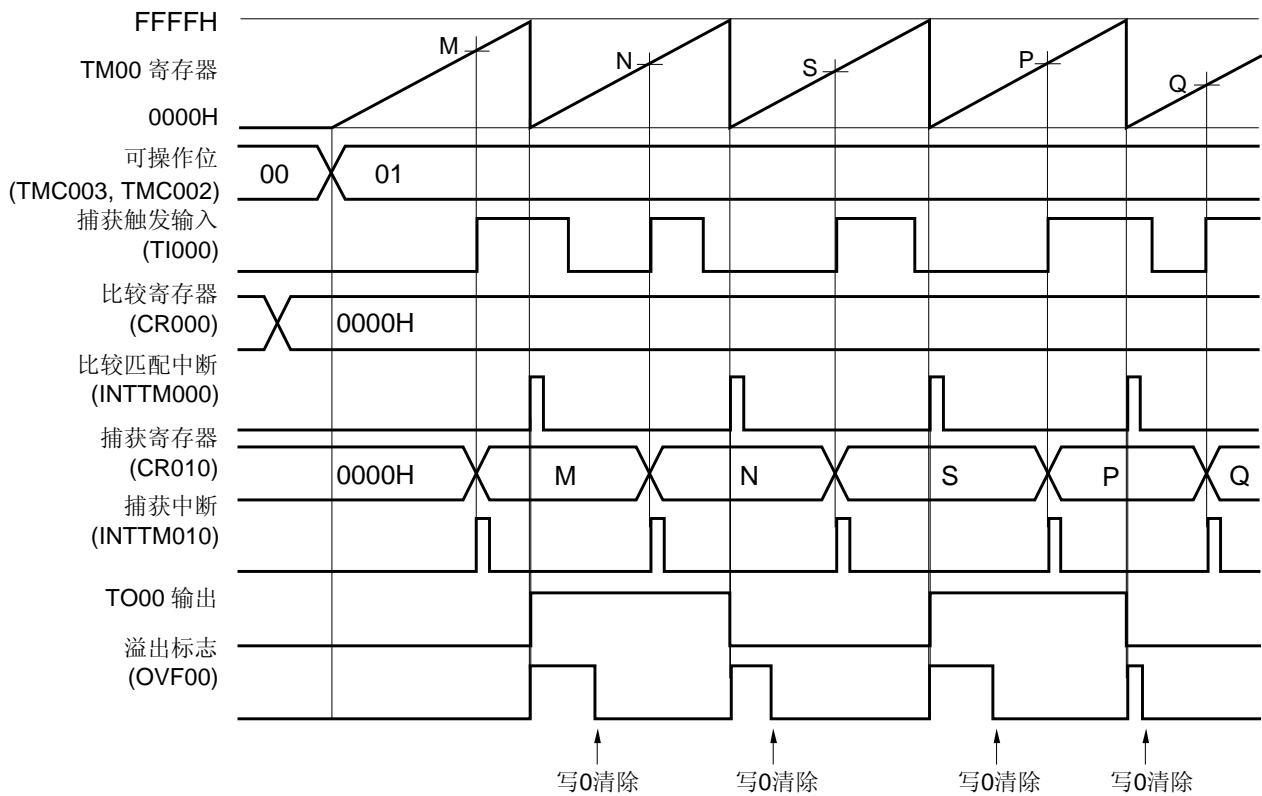


图 6-36. 自由运行定时器模式的时序示例
(CR000: 比较寄存器, CR010: 捕获寄存器) (2/2)

• TOC00 = 13H, PRM00 = 10H, CRC00 = 04H, TMC00 = 04H



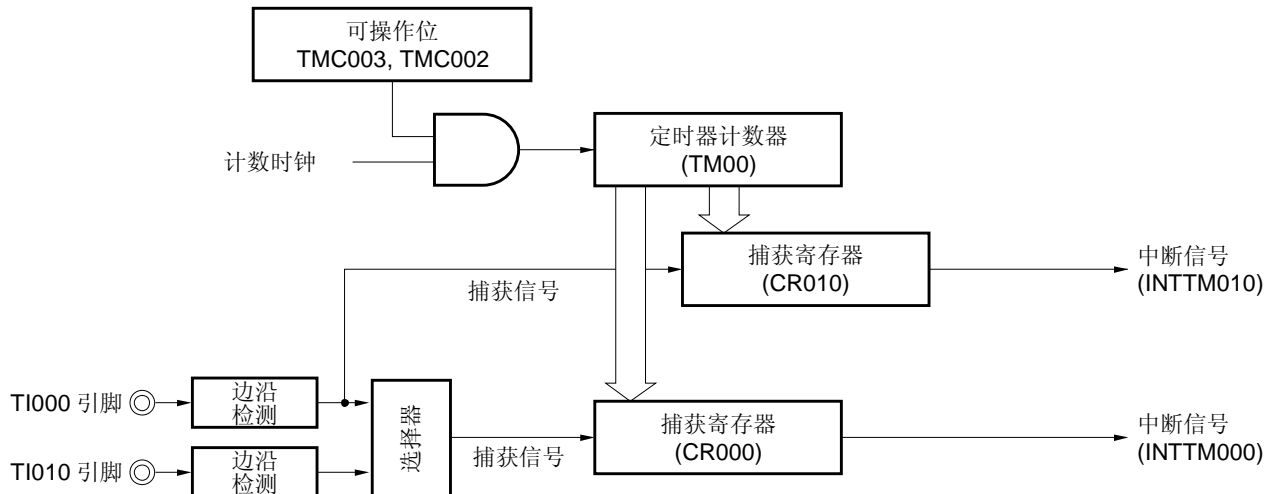
在该应用示例中，比较寄存器和捕获寄存器同时在自由运行定时器模式中使用。

在该示例中，每次 TM00 的计数值与 CR000（比较寄存器）的设定值匹配时都将会生成 INTTM000 信号，且 TO00 引脚的输出电平也会反转。此外，每次检测到 T1000 引脚的有效沿时都会生成 INTTM010 信号，且 TM00 的计数值也将会被捕获至 CR010。

(3) 自由运行定时器模式操作

(CR000: 捕获寄存器, CR010: 捕获寄存器)

图 6-37. 自由运行定时器模式的框图
(CR000: 捕获寄存器, CR010: 捕获寄存器)

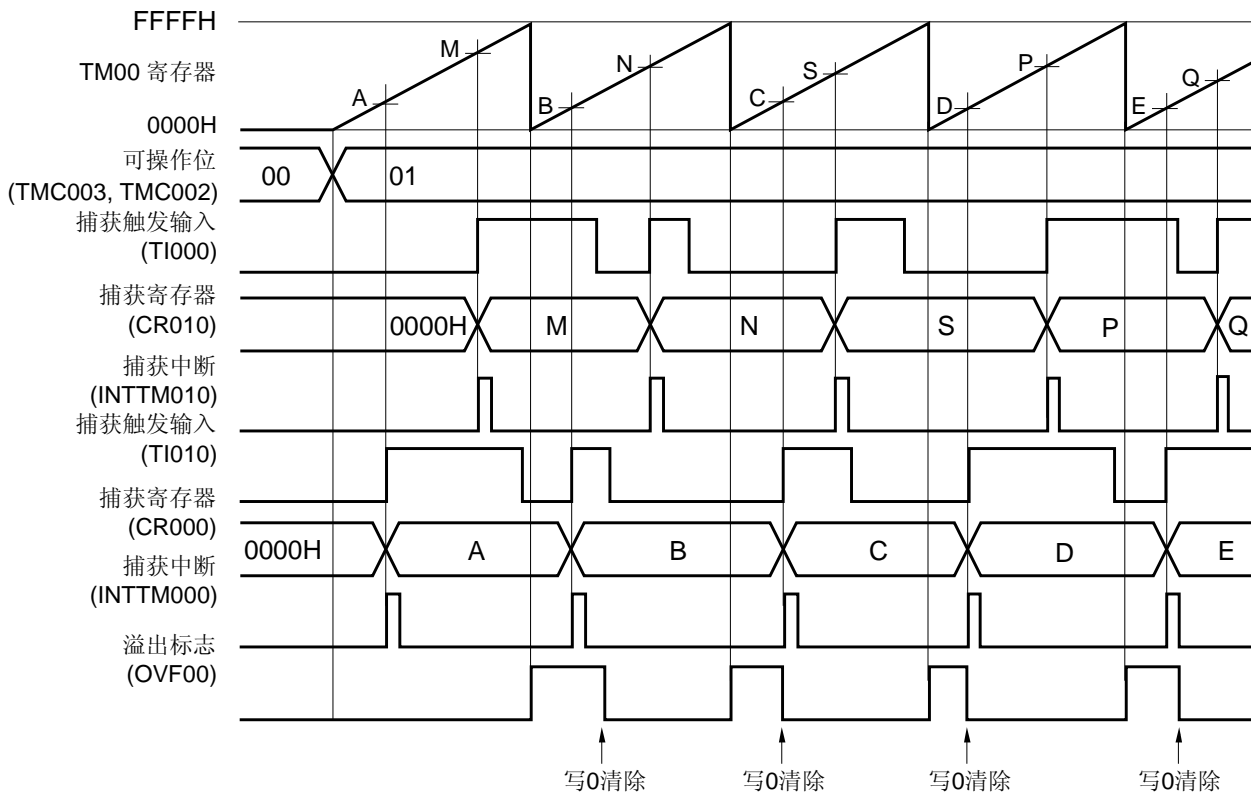


备注 如果 CR000 和 CR010 在自由选择定时器模式中均被用作捕获寄存器，那么 TO00 引脚的输出电平将不会反转。

然而，如果 16 位定时器模式控制寄存器 00 (TMC00) 的第 1 位 (TMC001) 被设为 1，那么每次检测到 TI000 引脚时 TO00 引脚输出电平都可以进行反转。

图 6-38. 自由运行定时器模式的时序示例
 (CR000: 捕获寄存器, CR010: 捕获寄存器) (1/2)

(a) TOC00 = 13H, PRM00 = 50H, CRC00 = 05H, TMC00 = 04H

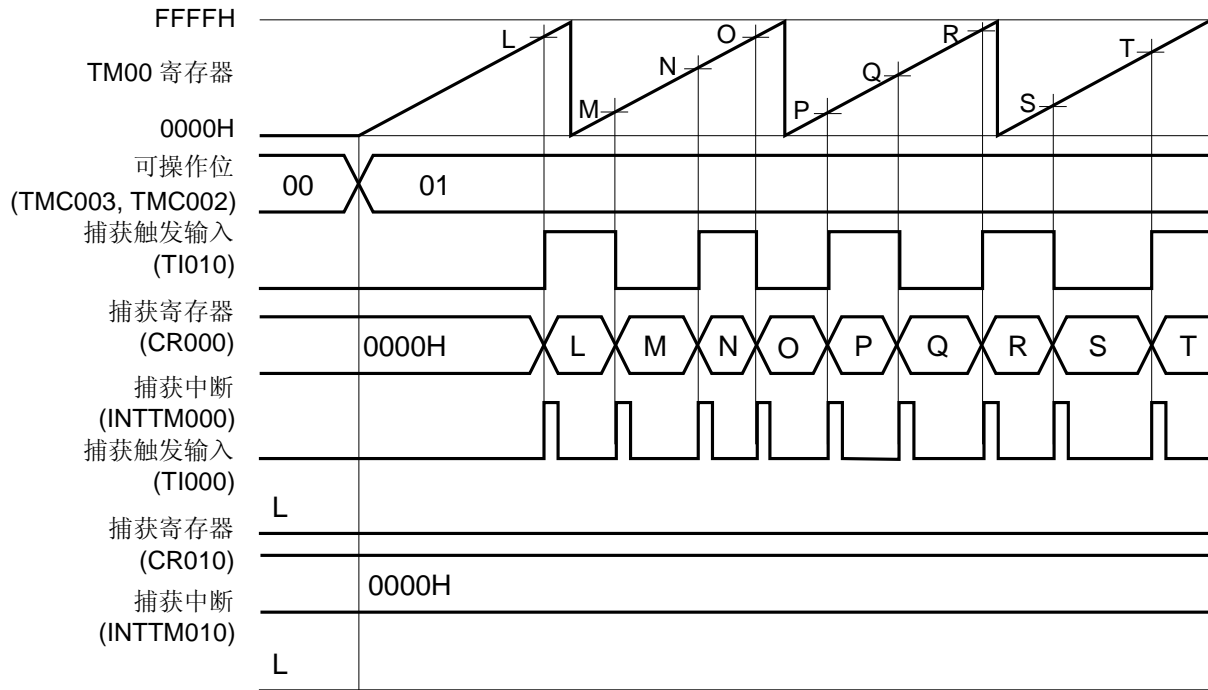


在该示例中，在单独的捕获触发信号的有效沿处捕获的计数值在自由运行定时器模式中被保存在单独的捕获寄存器中。

在检测到 TI000 引脚的有效沿时计数值将会被捕获至 CR010，而在检测到 TI010 引脚输入的有效沿时计数值则将会被捕获至 CR000 中。

图 6-38. 自由运行定时器模式的时序示例
(CR000: 捕获寄存器, CR010: 捕获寄存器) (2/2)

(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 04H



该示例中，TI010 引脚的双边沿均被检测到，且计数值在自由运行定时器模式中被捕获至 CR000。当 CR000 和 CR010 均用作捕获寄存器但只有 TI010 引脚的有效沿被检测到时，计数值将不能被捕获至 CR010。

图 6-39. 自由运行定时器模式的寄存器设置示例 (1/2)

(a) 16 位定时器模式控制寄存器 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	0	1	0/1	0

- 0: TM00和CR000/CR010匹配时反转TO00输出。
- 1: TM00和CR000/CR010以及TI000引脚的有效沿匹配时反转TO00输出。
- 自由运行定时器模式

(b) 捕获 / 比较控制寄存器 00 (CRC00)

					CRC002	CRC001	CRC000
0	0	0	0	0	0/1	0/1	0/1

- 0: CR000 用作比较寄存器
- 1: CR000 用作捕获寄存器
- 0: TI010 引脚用作CR000的捕获触发。
- 1: TI000引脚的反相位被用作CR000的捕获触发。
- 0: CR010 用作比较寄存器
- 1: CR010 用作捕获寄存器

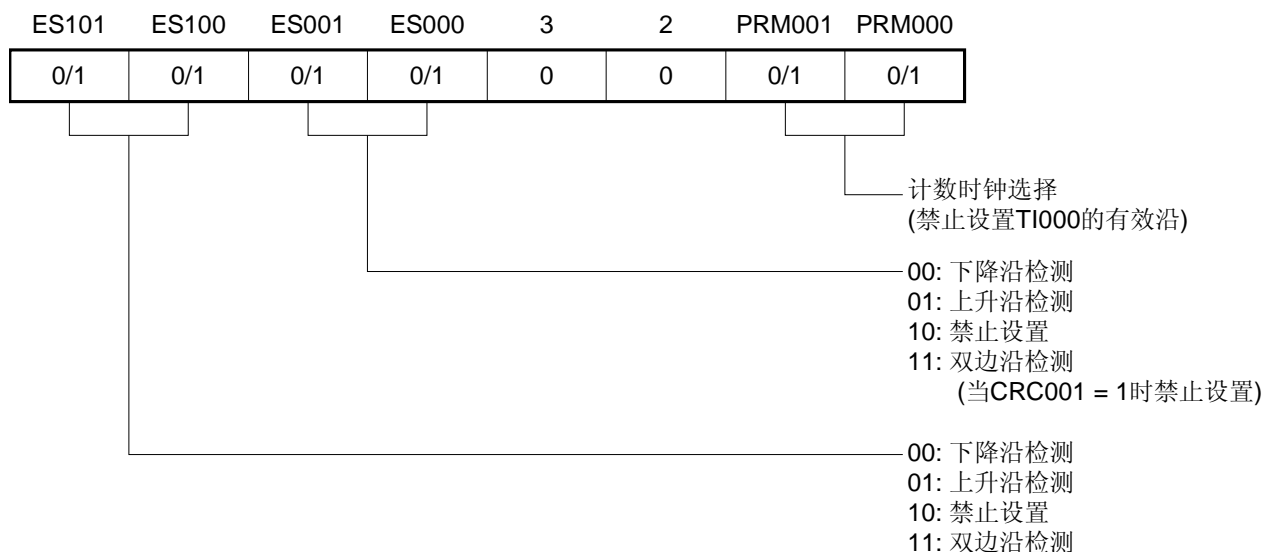
(c) 16 位定时器输出控制寄存器 00 (TOC00)

OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	0/1	0/1	0/1	0/1

- 0: 禁止 TO00 输出
- 1: 允许 TO00 输出
- 指定TO00输出F/F的初始值
- 00: TM00与CR000/CR010匹配时不反转TO00输出。
- 01: TM00 和 CR000匹配时反转TO00输出。
- 10: TM00和CR010匹配时反转TO00输出。
- 11: TM00和CR000/CR010匹配时反转TO00输出。

图 6-39. 自由运行定时器模式的寄存器设置示例 (2/2)

(d) 预分频模式寄存器 00 (PRM00)



(e) 16 位定时器计数器 00 (TM00)

计数值可以通过读取 TM00 来读取。

(f) 16 位捕获 / 比较寄存器 000 (CR000)

当该寄存器用作比较寄存器且该寄存器的值与 TM00 的计数值匹配时，将会生成一个中断信号 (INTTM000)。TM00 的计数值不被清除。

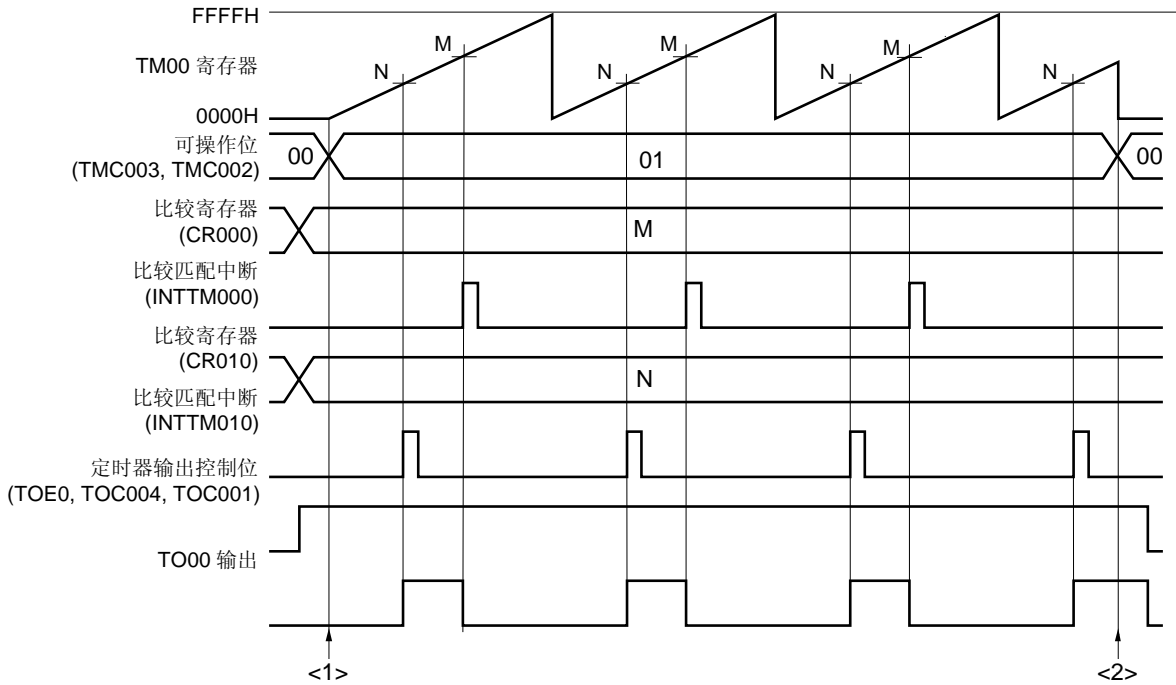
若要将该寄存器用作捕获寄存器，则应将 TI000 或 TI010 引脚选作捕获触发器。当检测到捕获触发器的有效沿时，TM00 的计数值将会被保存在 CR000 中。

(g) 16 位捕获 / 比较寄存器 010 (CR010)

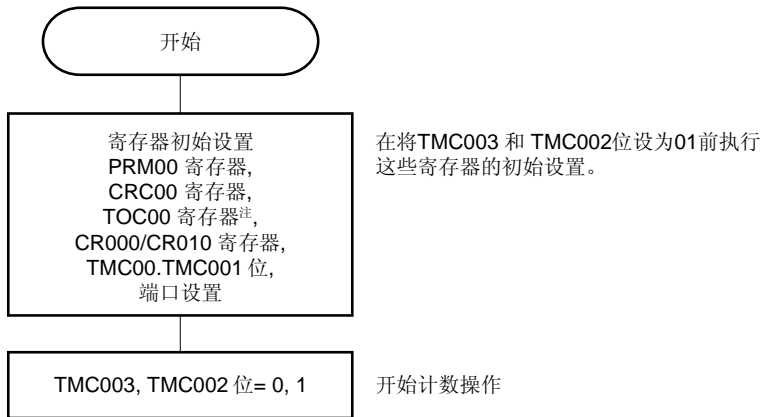
当该寄存器用作比较寄存器且该寄存器的值与 TM00 的计数值匹配时，将会生成一个中断信号 (INTTM010)。TM00 的计数值不被清除。

当该寄存器被用作捕获寄存器时，TI000 引脚输入将会被用作捕获触发器。当检测到捕获触发器的有效沿时，TM00 的计数值将会被保存在 CR010 中。

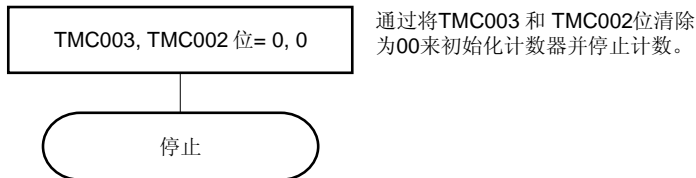
图 6-40. 自由运行定时器模式的软件处理示例



<1> 计数操作开始流程



<2> 计数操作停止流程



注 对 TOC00 进行设置时必需要谨慎。详细信息，请参见 6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

6.4.6 PPG输出操作

当 16 位定时器模式控制寄存器 00 (TMC00) 被设为 11 (TM00 与 CR000 匹配时的清除&开始) 时, 带有由 CR010 事先设定的脉宽的方波将会在 CR000 设定的周期内从 TO00 引脚中输出为 PPG (可编程脉冲发生器) 信号。

生成 PPG 输出脉冲的脉冲周期和占空比系数如下所示。

- 脉冲周期 = (CR000 的设定值 + 1) × 计数时钟周期
- 占空比系统 = (CR010 的设定值 + 1) / (CR000 的设定值 + 1)

注意事项 若要在操作过程中改变占空比系数 (CR010 的值), 请参见 6.5.1 TM00 操作期间重写 CR010。

备注 1. 关于 I/O 引脚设置的详细信息, 请参见 6.3 (5) 端口模式寄存器 0 (PM0) 和 (6) 端口输出模式寄存器 (POM0)。

2. 关于允许 INTTM000 信号中断的方法, 请参见第十一章 中断功能。

图 6-41. PPG 输出操作的框图

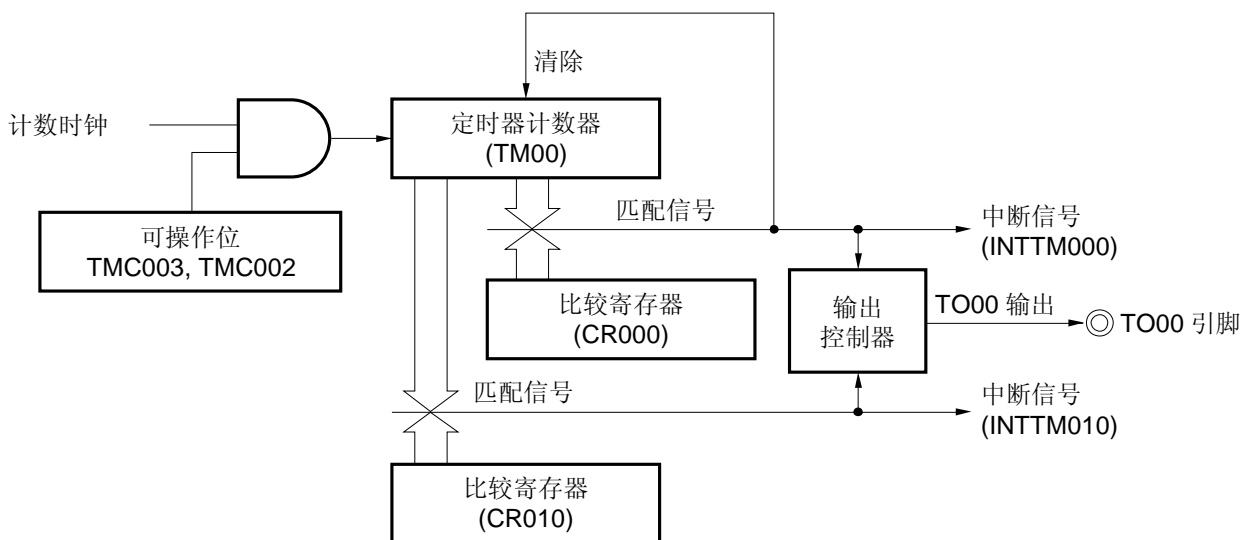
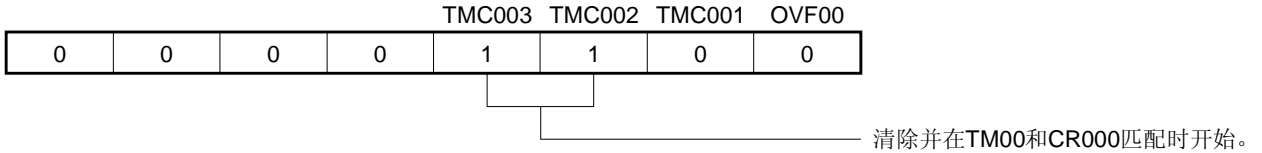
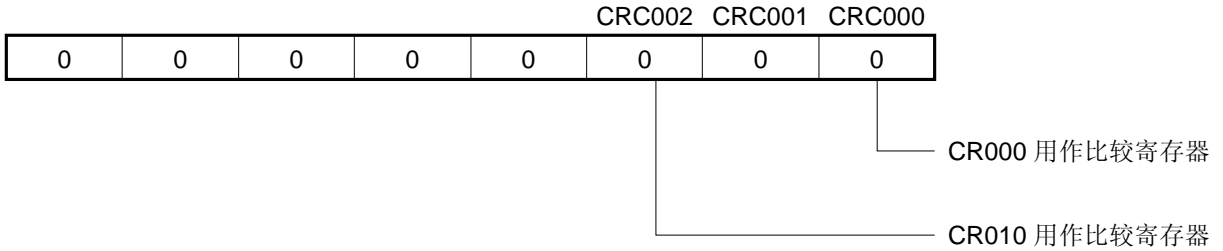


图 6-42. PPG 输出操作的寄存器设置示例

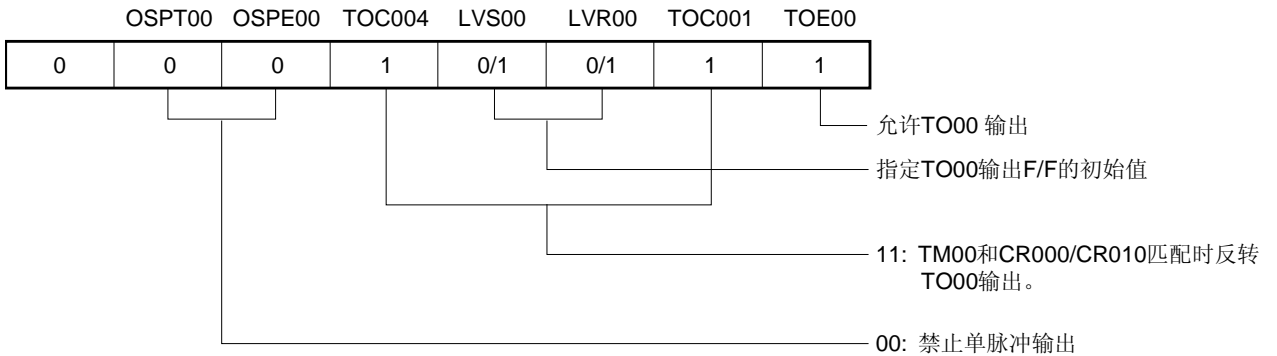
(a) 16 位定时器模式控制寄存器 00 (TMC00)



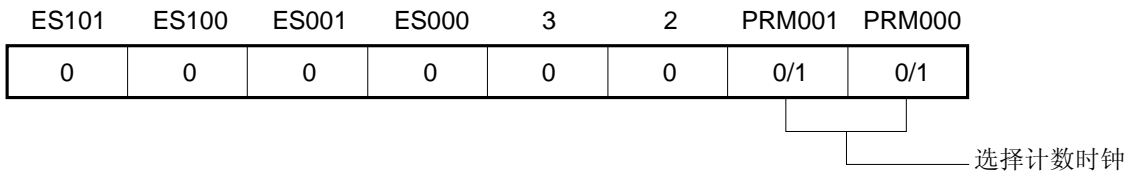
(b) 捕获 / 比较控制寄存器 00 (CRC00)



(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)



(e) 16 位定时器计数器 00 (TM00)

计数值可以通过读取 TM00 来读取。

(f) 16 位捕获 / 比较寄存器 000 (CR000)

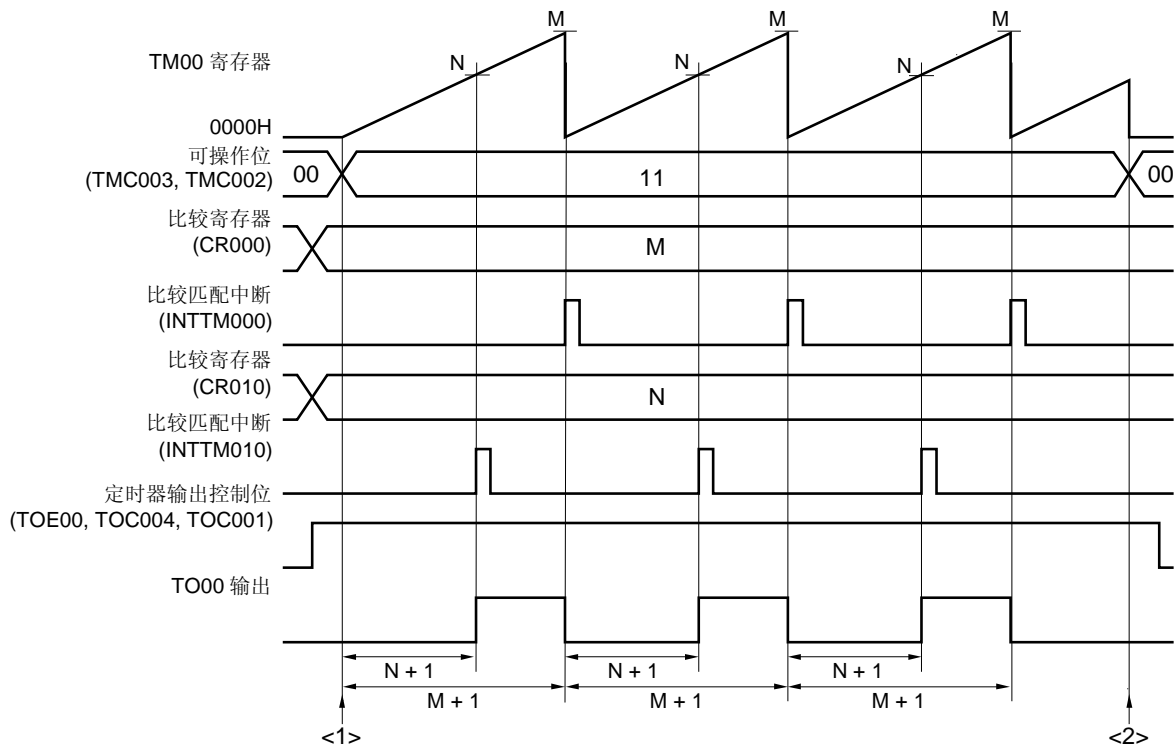
当寄存器的值与 TM00 的计数值匹配时将会生成一个中断信号 (INTTM000)。TM00 的计数值不被清除。

(g) 16 位捕获 / 比较寄存器 010 (CR010)

当该寄存器的值与 TM00 的计数值匹配时将会生成一个中断信号 (INTTM010)。TM00 的计数值不被清除。

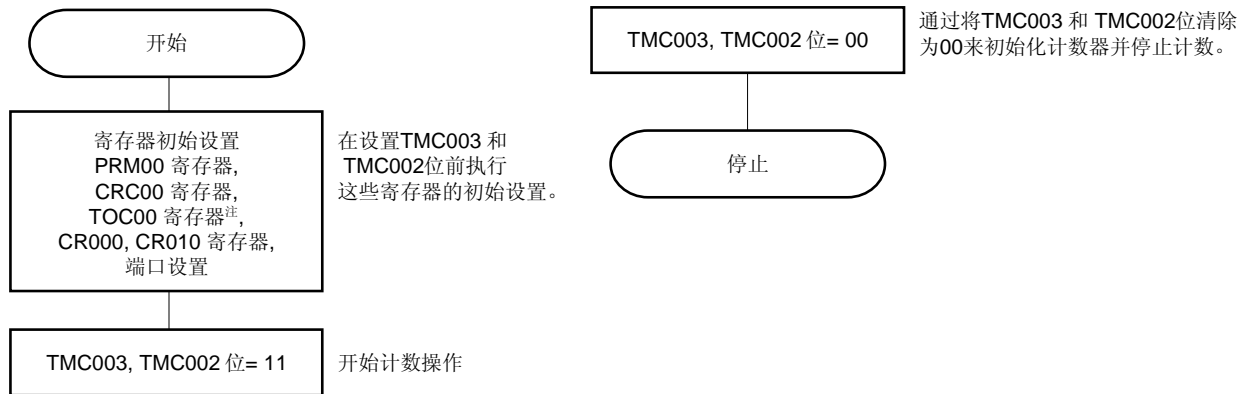
注意事项 在满足 $0000H \leq CR010 < CR000 \leq FFFFH$ 的条件时赋值给 CR000 和 CR010。

图 6-43. PPG 输出操作的软件处理示例



<1> 计数操作开始流程

<2> 计数操作停止流程



注 对 TOC00 进行设置时必须谨慎。详细信息，请参见 6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

备注 PPG 脉冲周期 = (M + 1) × 计数时钟周期

PPG 占空比系数 = (N + 1) / (M + 1)

6.4.7 单脉冲输出操作

通过将 16 位定时器模式控制寄存器 00 (TMC00) 的第 3 位和第 2 位 (TMC003 和 TMC002) 设置为 01 (自由运行定时器模式) 或 10 (通过 TI000 引脚有效沿进入清除&开始模式) 并将 16 位定时器输出控制寄存器 00 (TOC00) 的第 5 位 (OSPE00) 设置为 1 可以输出一个单脉冲。

在 TOC00 的第 6 位 (OSPT00) 被设为 1 或在定时器操作期间将有效沿输入 TI000 引脚时, TM00 的清除&开始模式将会被触发, 且 CR000 和 CR010 的值间相差的脉冲只会从 TO00 引脚中输出一次。

- 注意事项**
1. 输出单脉冲时不要再次输入触发 (将 OSPT00 设为 1 或检测 TI000 引脚的有效沿)。要再次输出单脉冲, 则应在当前单脉冲输出完成后生成触发。
 2. 若只将设置为 1 的 OSPT00 的设置用作单脉冲输出, 则不要改变 TI000 引脚或其复位功能端口引脚的电平。否则, 脉冲将不能按期望的来输出。

- 备注**
1. 关于 I/O 引脚设置的详细信息, 请参见 6.3 (5) 端口模式寄存器 0 (PM0) 和 (6) 端口输出模式寄存器 (POM0)。
 2. 关于允许 INTTM000 信号中断的方法, 请参见第十一章 中断功能。

图 6-44. 单脉冲输出操作的框图

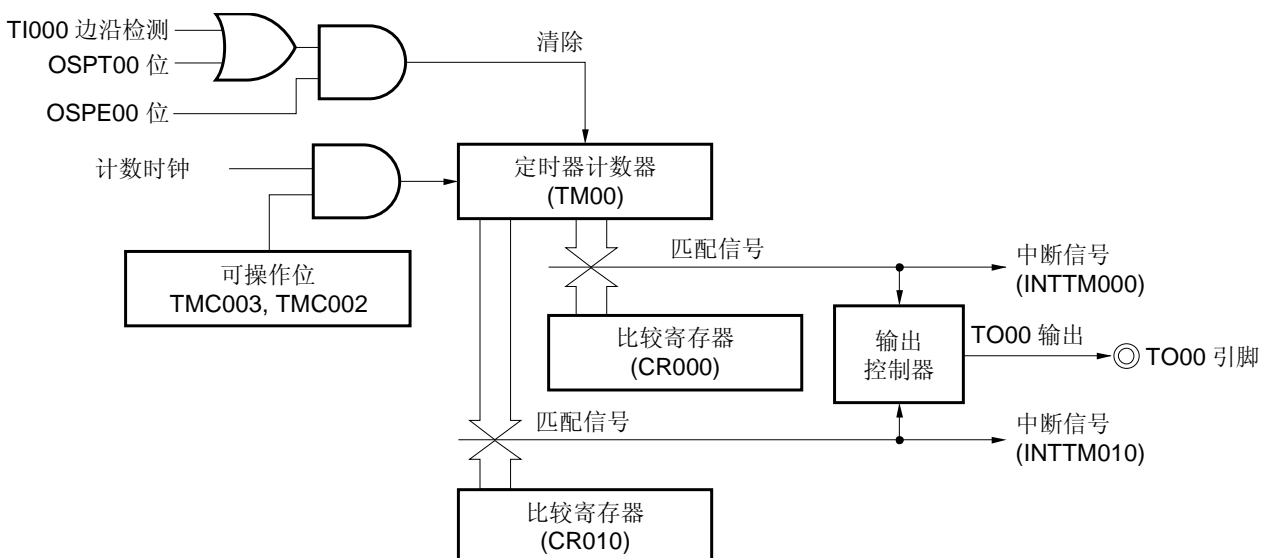
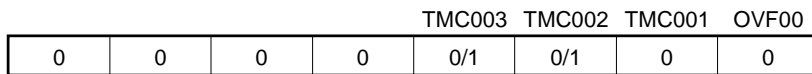


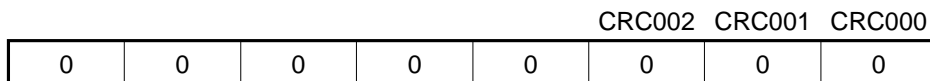
图 6-45. 单脉冲输出操作的寄存器设置示例 (1/2)

(a) 16 位定时器模式控制寄存器 00 (TMC00)



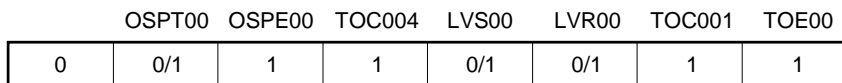
01: 自由运行定时器模式
10: 通过TI000引脚有效沿清除及开始模式。

(b) 捕获 / 比较控制寄存器 00 (CRC00)



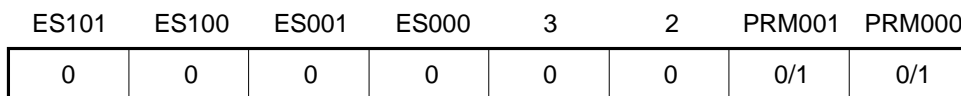
CR000用作比较寄存器
CR010用作比较寄存器

(c) 16 位定时器输出控制寄存器 00 (TOC00)



允许TO00 输出
指定TO00 输出的初始值
TM00和CR000/CR010 匹配时反转TO00 输出
允许单脉冲输出
通过向该位写1来生成软件触发 (即使0被写入该位操作也不会受影响)。

(d) 预分频模式寄存器 00 (PRM00)



选择计数时钟

图 6-45. 单脉冲输出操作的寄存器设置示例 (2/2)

(e) 16 位定时器计数器 00 (TM00)

计数值可以通过读取 TM00 来读取。

(f) 16 位捕获 / 比较寄存器 000 (CR000)

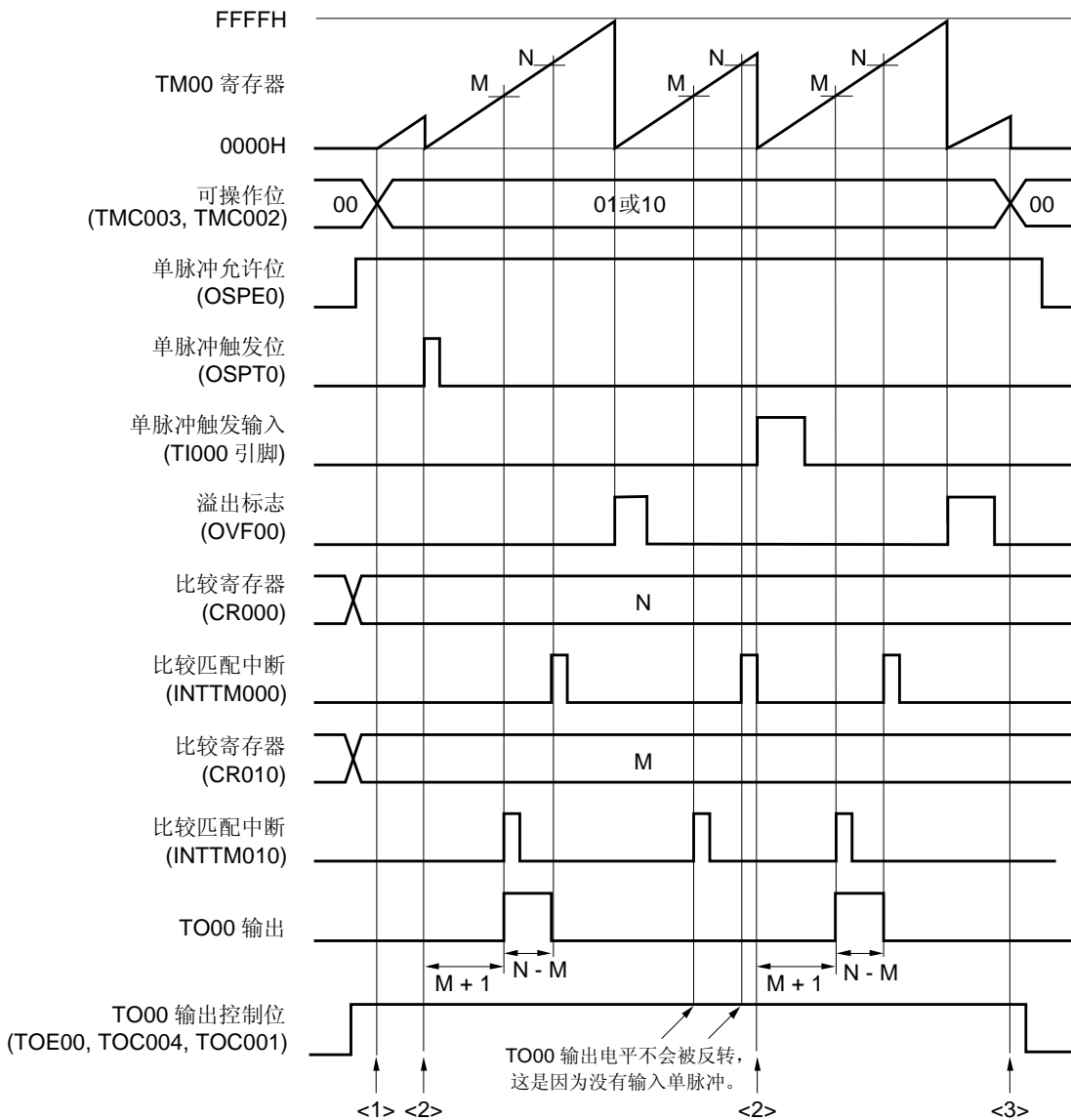
在输出单脉冲时，该寄存器用作比较寄存器。当 TM00 的值与 CR000 的值匹配时，将会生成一个中断信号 (INTTM000)，且 TO00 引脚的输出电平将会被反转。

(g) 16 位捕获 / 比较寄存器 010 (CR010)

在输出单脉冲时，该寄存器用作比较寄存器。当 TM00 的值与 CR010 的值匹配时，将会生成一个中断信号 (INTTM010)，且 TO00 引脚的输出电平将会被反转。

注意事项 不要对 CR000 和 CR010 设置相同的值。

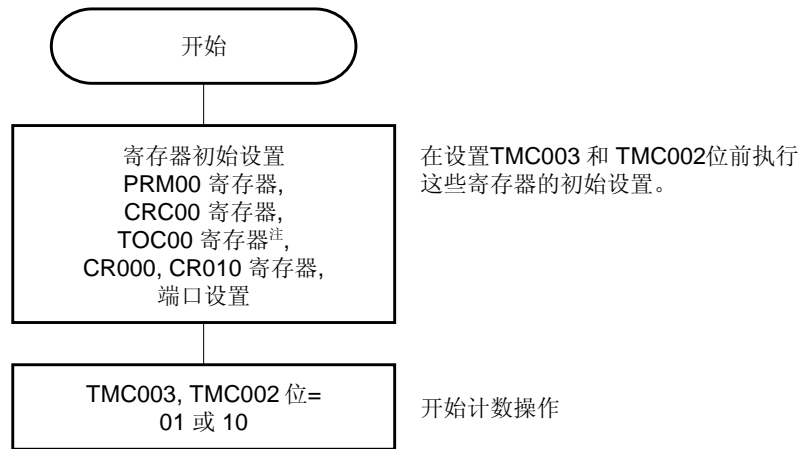
图 6-46. 单脉冲输出操作的软件处理示例 (1/2)



- 从输入单脉冲触发开始直到输出单脉冲为止的时间
= $(M + 1) \times$ 计数时钟周期
- 单脉冲输出有效电平宽度
= $(N - M) \times$ 计数时钟周期

图 6-46. 单脉冲输出操作的软件处理示例 (2/2)

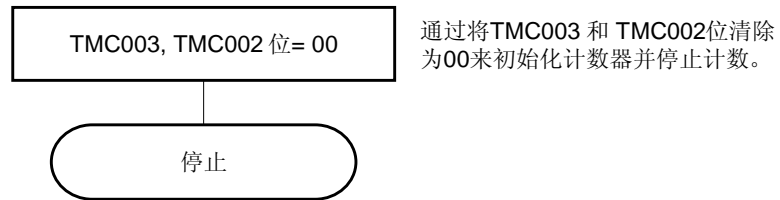
<1> 计数操作开始流程



<2> 单脉冲触发输入流程



<3> 计数操作停止流程



注 对 TOC00 进行设置时必须需要谨慎。详细信息，请参见 6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

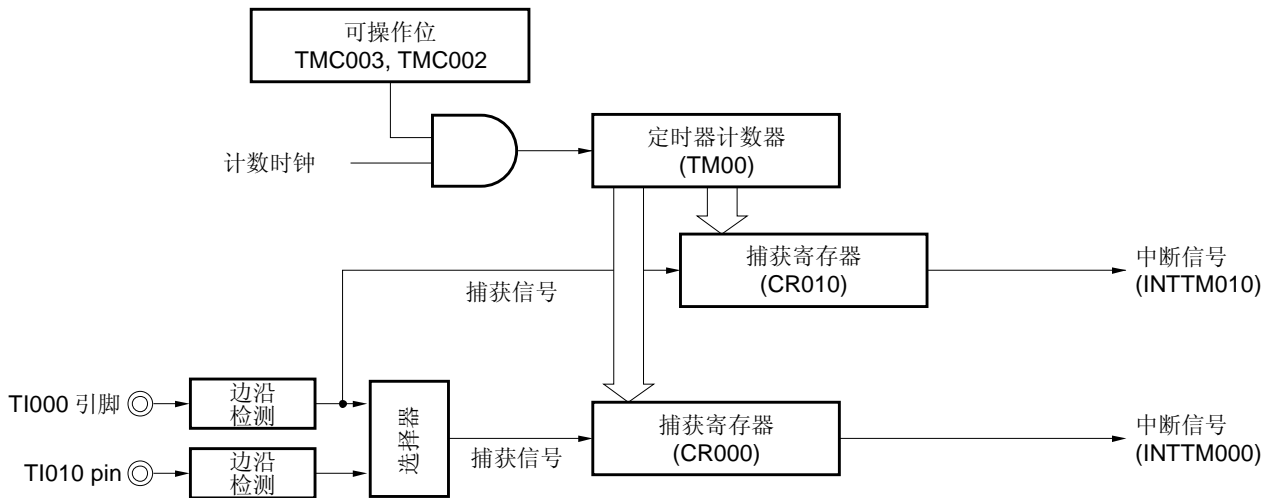
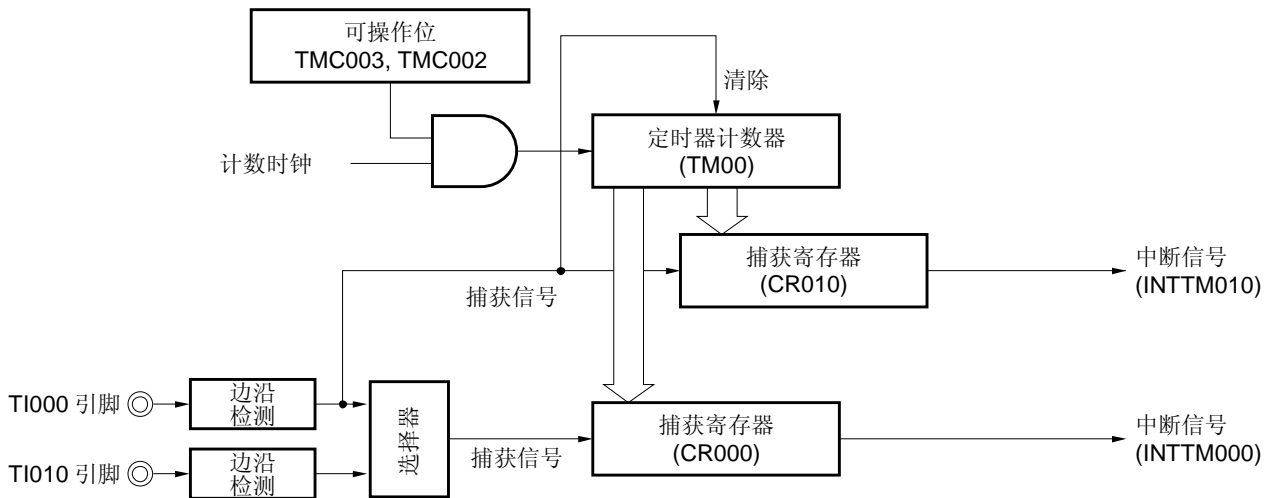
6.4.8 脉宽测量操作

TM00 可以用于测量输入到 TI000 和 TI010 引脚中信号的脉宽。

测量可以通过在自由运行定时器模式中操作 16 位定时器 / 事件计数器 00 来实现，也可以通过与输入到 TI000 引脚中的信号重新同步开始来实现。

当中断产生时，将读取有效捕获寄存器的值并测量脉宽。检测 16 位定时器模式控制寄存器 00（TMC00）的 0 位（OVF00）。如果它被设为 1，则通过软件将其清零。

图 6-47. 脉宽测量（自由运行定时器模式）的框图

图 6-48. 脉宽测量的框图
(通过 TI000 引脚有效沿输入进入的清除&开始模式)

脉宽可以通过以下三种方式进行测量。

- 通过使用 TI000 和 TI010 引脚的两个输入信号来测量脉宽（自由运行定时器模式）
- 通过使用 TI000 引脚的一个输入信号来测量脉宽（自由运行定时器模式）
- 通过使用 TI000 引脚一个输入信号来测量脉宽（通过 TI000 引脚有效沿输入进入的清除&开始模式）

备注

1. 关于 I/O 引脚设置的详细信息，请参见 6.3 (5) 端口模式寄存器 0 (PM0)。
2. 关于允许 INTTM000 信号中断的方法，请参见第十一章 中断功能。

(1) 通过使用 TI000 和 TI010 引脚的两个输入信号来测量脉宽（自由运行定时器模式）

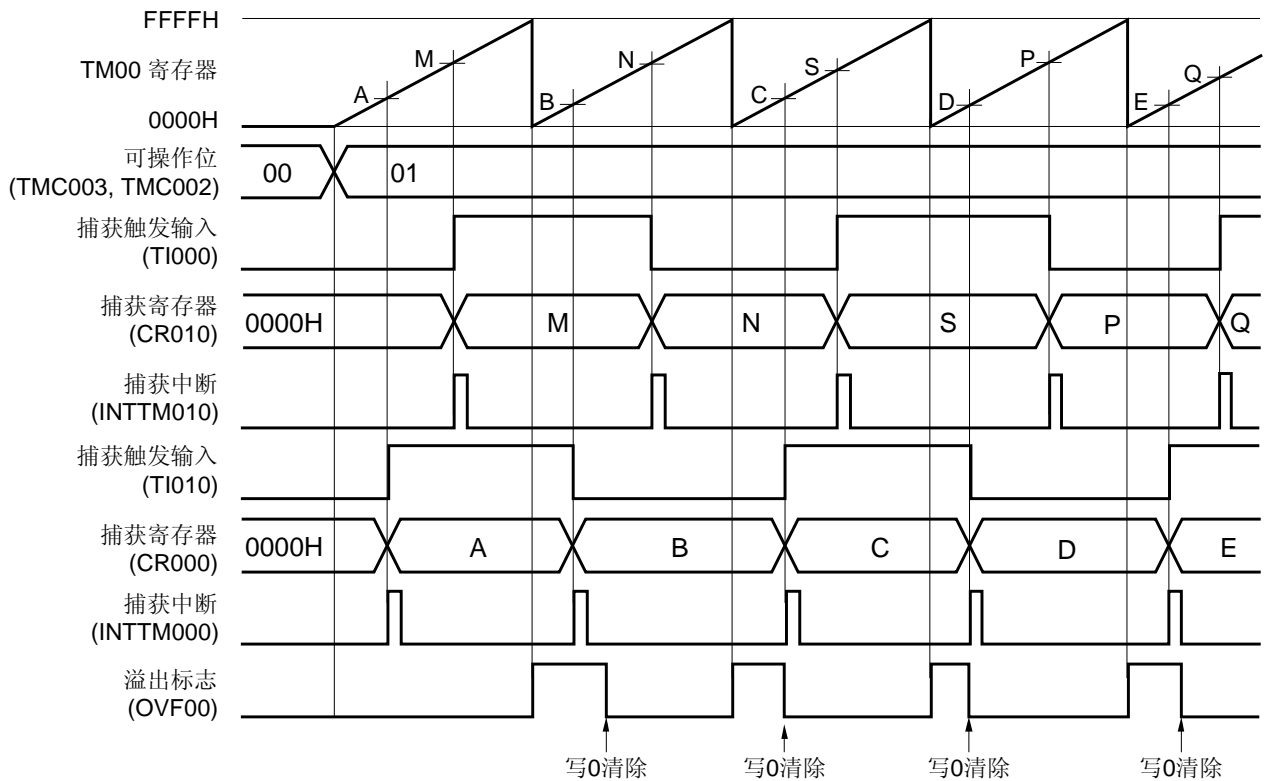
设置自由运行定时器模式 (TMC003 和 TMC002 = 01)。当检测到 TI000 引脚的有效沿时，TM00 的计数值将会被捕获至 CR010 中。当检测到 TI010 引脚的有效沿时，TM00 的计数值将会被捕获至 CR000 中。指定 TI000 和 TI010 引脚的双边沿检测。

通过这种测量方法，先前的计数值将会与各个输入信号的边沿所捕获的计数值相减。因此，需事先将先前捕获的值保存在一个单独的寄存器中。

发生溢出时，如果先前捕获的值只是直接与当前捕获的值相减，那么值将会是负的，也会因此发生借位（程序状态字 (PSW) 的 0 位 (CY) 被设为 1)。如果发生这种情况，则忽略 CY 并将计数结果作为脉宽。此外，应将 16 位定时器模式控制寄存器 00 (TMC00) 的 0 位 (OVF00) 清除为 0。

图 6-49. 脉宽测量的时序示例 (1)

• TMC00 = 04H, PRM00 = F0H, CRC00 = 05H



(2) 通过使用 TI000 引脚的一个输入信号来测量脉宽（自由运行模式）

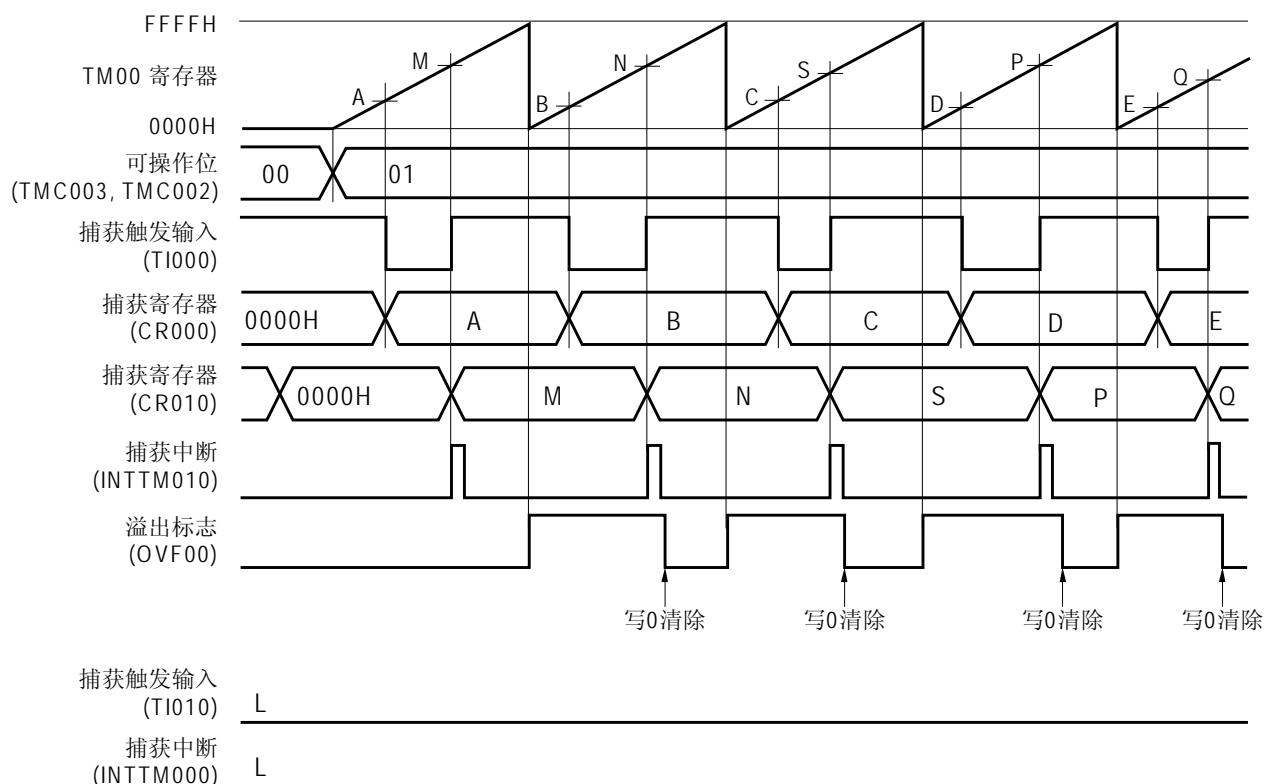
设置自由运行定时器模式（TMC003 和 TMC002 = 01）。TM00 的计数值在与 TI000 引脚上检测的有效沿的相位相反的相位中被捕获至 CR000。当检测到 TI000 引脚的有效沿时，TM00 的计数值将会被捕获至 CR010 中。

使用这种测量方法时，在测量从某一边沿至另一边沿的宽度时，值会被保存在单独的捕获寄存器中。因此，不必保存捕获值。通过将两种捕获寄存器中的值相减，可以计算高电平宽，低电平宽，以及周期。

发生溢出时，如果捕获的值只是直接与另一捕获的值相减，那么值将会是负的，也会因此发生借位（程序状态字（PSW）的 0 位（CY）被设为 1）。如果发生这种情况，则忽略 CY 并将计数结果作为脉宽。此外，应将 16 位定时器模式控制寄存器 00（TMC00）的 0 位（OVF00）清除为 0。

图 6-50. 脉宽测量的时序示例（2）

• TMC00 = 04H, PRM00 = 10H, CRC00 = 07H



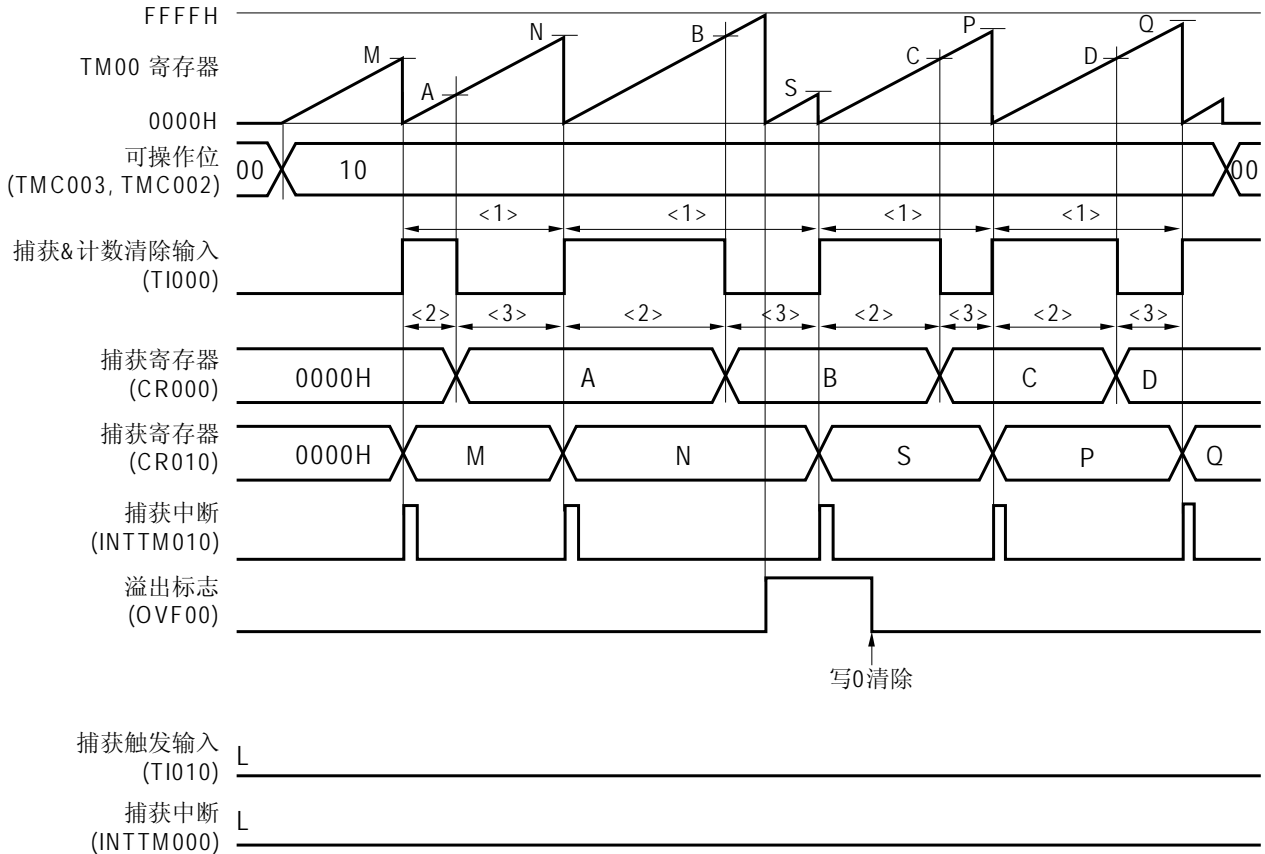
(3) 通过使用 TI000 引脚一个输入信号来测量脉宽 (通过 TI000 引脚有效沿输入进入的清除&开始模式)

设置通过 TI000 引脚有效沿 (TMC003 和 TMC002 = 10) 进入的清除&开始模式。TM00 的计数值在与 TI000 引脚有效边沿的相位相反的相位中被捕获至的 CR000, TM00 的计数值则被捕获至 CR010, 而 TM00 在检测到 TI000 引脚的有效沿时被清除 (为 0000H)。因此, 如果 TM00 不溢出, 那么周期将会保存在 CR010 中。

如果发生溢出, 则将 10000H 加至保存在 CR010 中的值后所得的结果作为周期。将 16 位定时器模式控制寄存器 00 (TMC00) 的 0 位 (OVF00) 清除为 0。

图 6-51. 脉宽测量的时序示例 (3)

• TMC00 = 08H, PRM00 = 10H, CRC00 = 07H



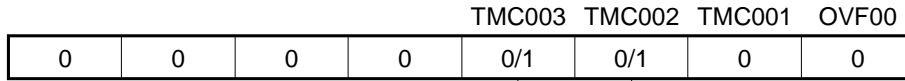
<1> 脉冲周期 = $(10000H \times \text{OVF00 位被设为 1 的次数} + \text{CR010 的捕获值}) \times \text{计数时钟周期}$

<2> 高电平脉宽 = $(10000H \times \text{OVF00 位被设为 1 的次数} + \text{CR000 的捕获值}) \times \text{计数时钟周期}$

<3> 低电平脉宽 = $(\text{脉冲周期} - \text{高电平脉宽})$

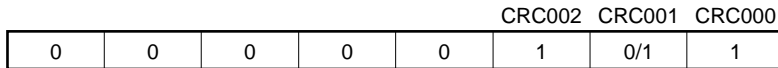
图 6-52. 脉宽测量的寄存器设置示例 (1/2)

(a) 16 位定时器模式控制寄存器 00 (TMC00)



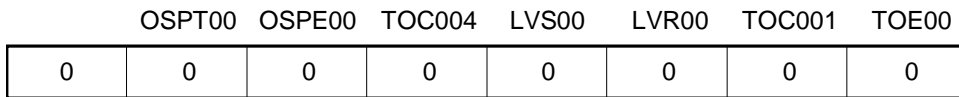
01: 自由运行定时器模式
10: 清除并开始通过TI000引脚有效进行的模式。

(b) 捕获 / 比较控制寄存器 00 (CRC00)

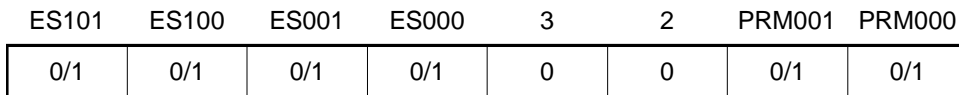


1: CR000用作捕获寄存器
0: TI010 引脚用作CR000的捕获触发。
1: TI000引脚的反相位被用作CR000的捕获触发。
1: CR010 用作捕获寄存器

(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)



选择计数时钟
(禁止设置TI000的有效沿)

00: 下降沿检测
01: 上升沿检测
10: 禁止设置
11: 双边沿检测
(当CRC001 = 1时禁止设置)

00: 下降沿检测
01: 上升沿检测
10: 禁止设置
11: 双边沿检测

图 6-52. 脉宽测量的寄存器设置示例 (2/2)

(e) 16 位定时器计数器 00 (TM00)

计数值可以通过读取 TM00 来读取。

(f) 16 位捕获 / 比较寄存器 000 (CR000)

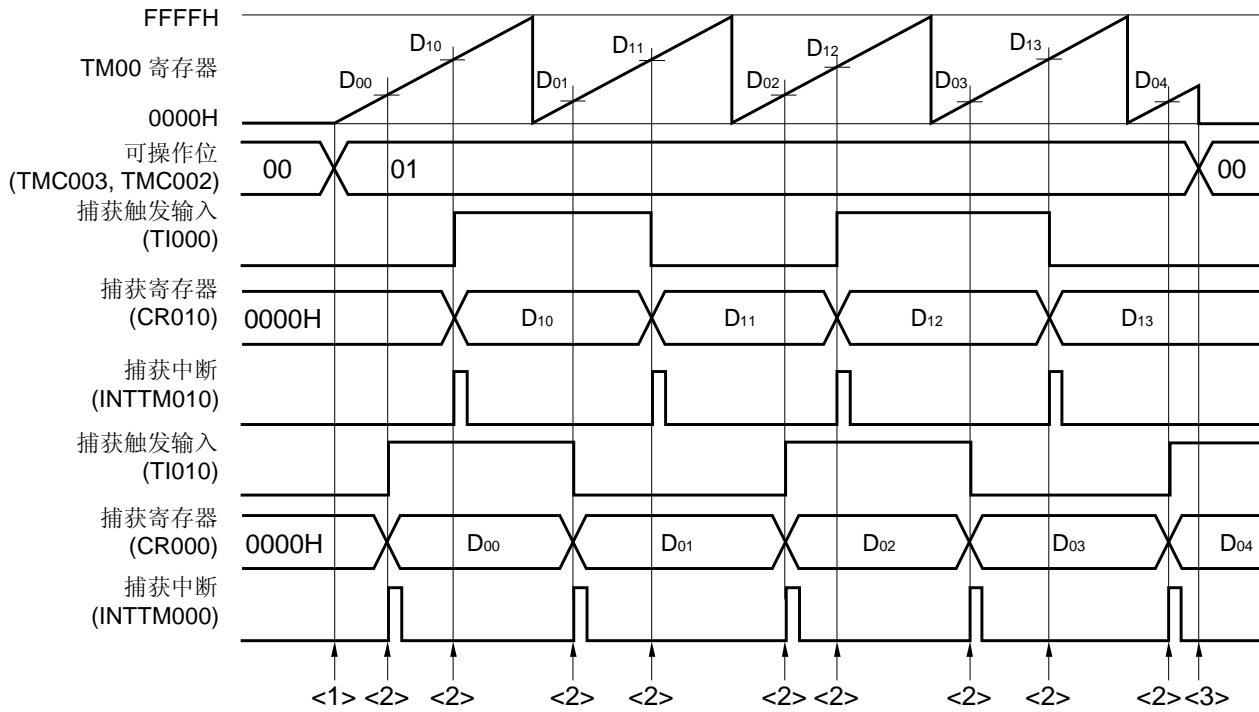
该寄存器用作捕获寄存器。TI000 或 TI010 引脚被选作捕获触发器。当检测到捕获触发器的指定沿时，TM00 的计数值将会被保存在 CR000 中。

(g) 16 位捕获 / 比较寄存器 010 (CR010)

该寄存器用作捕获寄存器。输入到 TI000 引脚的信号被用作捕获触发信号。当检测到捕获触发时，TM00 的计数值将会被保存在 CR010 中。

图 6-53. 脉宽测量的软件处理示例 (1/2)

(a) 自由运行定时器模式的示例



(b) 通过 TI000 引脚有效沿进入的清除&开始模式示例

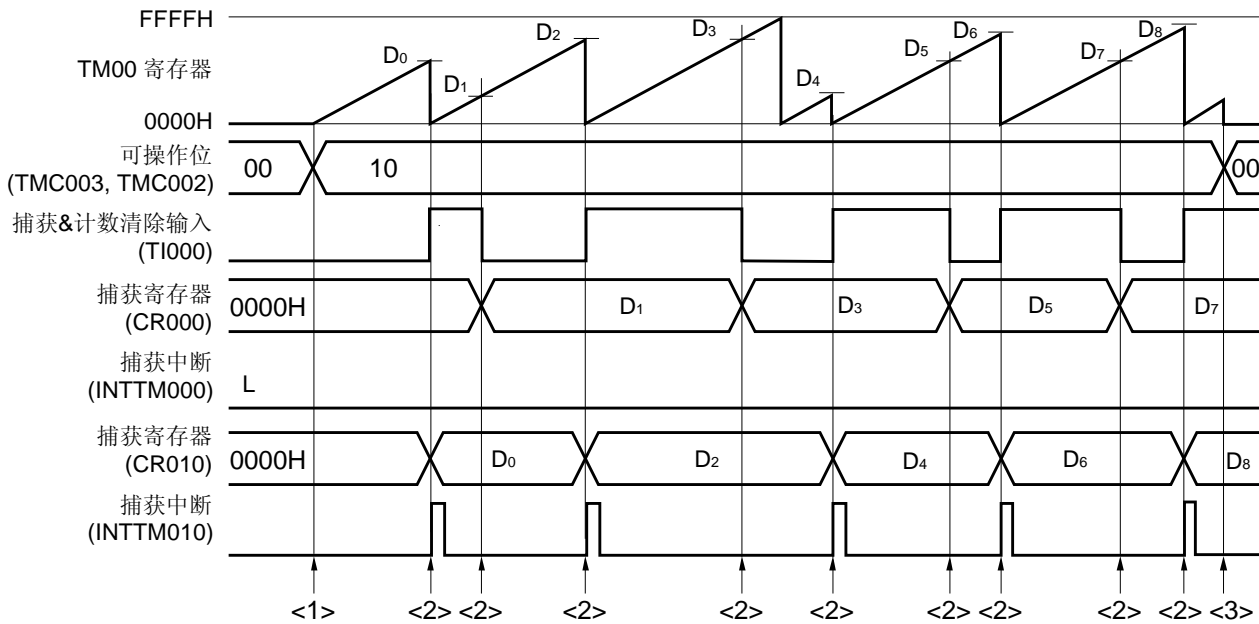
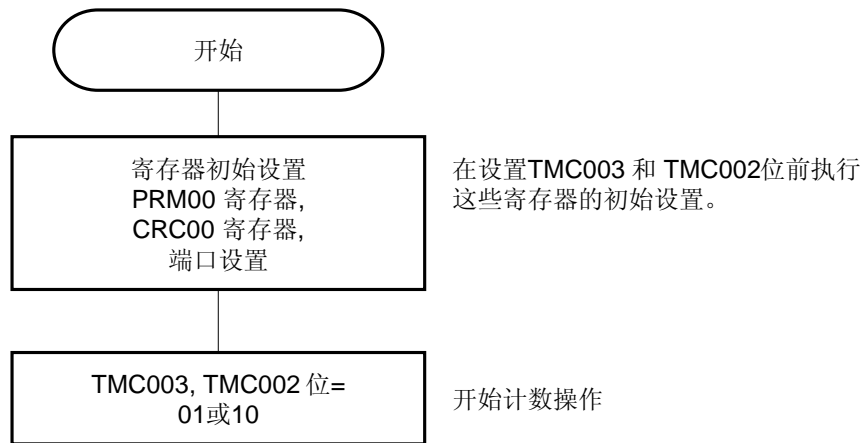
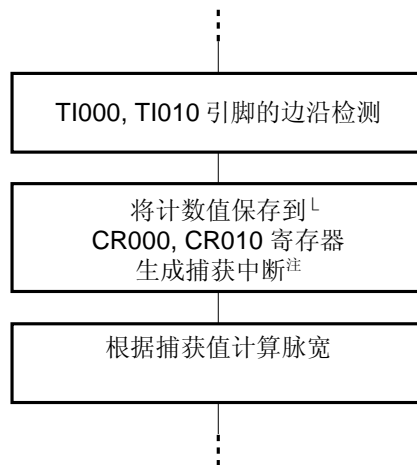


图 6-53. 脉宽测量的软件处理示例 (2/2)

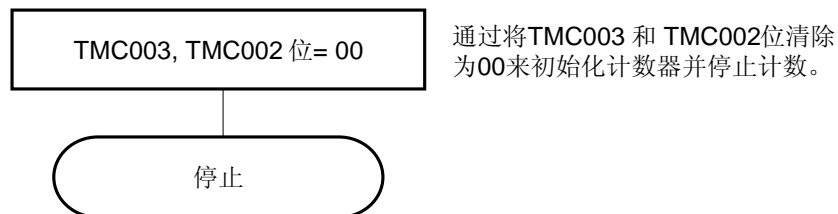
<1> 计数操作开始流程



<2> 捕获触发输入流程



<3> 计数操作停止流程



注 当 TI000 引脚输入的反向边沿被选作 CR000 的有效沿时，将不能生成捕获中断信号（INTTM000）。

6.5 TM00 的特殊用途

6.5.1 TM00 操作期间重写CR010

一般来说，在 μ PD179F11x, 179F12x 微控制器中，当 TM00 进行操作时（TMC003 和 TMC002 不等于 00），禁止在 CR000 和 CR010 用作比较寄存器时对它们进行重写。

然而，如果 CR010 用于 PPG 输出且占空比系数被改变，那么即使 TM00 在进行操作，CR010 的值也可以改变。（将 CR010 的值改为比当前值小的值时，在 CR010 的值与 TM00 的值匹配后立即对其进行改写。将 CR010 的值改为比当前值大的值时，在 CR000 的值与 TM00 的值匹配后立即对其进行改写。如果 CR010 的值在 CR010 与 TM00 匹配前，或在 CR000 与 TM00 匹配前被改写，那么可能会发生意想不到的操作。）。

改变 CR010 的值的步骤

- <1> 禁止中断 INTTM010（TMMK010 = 1）。
- <2> 当 TM00 的值与 CR010 的值（TOC004 = 0）匹配时，禁止定时器输出的反转。
- <3> 更改 CR010 的值。
- <4> 等待 TM00 的计数时钟的一个周期。
- <5> 当 TM00 的值与 CR010 的值（TOC004 = 1）匹配时，允许定时器输出的反转。
- <6> 将 INTTM010（TMIF010 = 0）的中断标志清除为 0。
- <7> 允许中断 INTTM010（TMMK010 = 0）。

备注 关于 TMIF010 和 TMMK010 的详细信息，请参见第十一章 中断功能。

6.5.2 设置LVS00 和 LVR00

(1) LVS00 和 LVR00 的用途

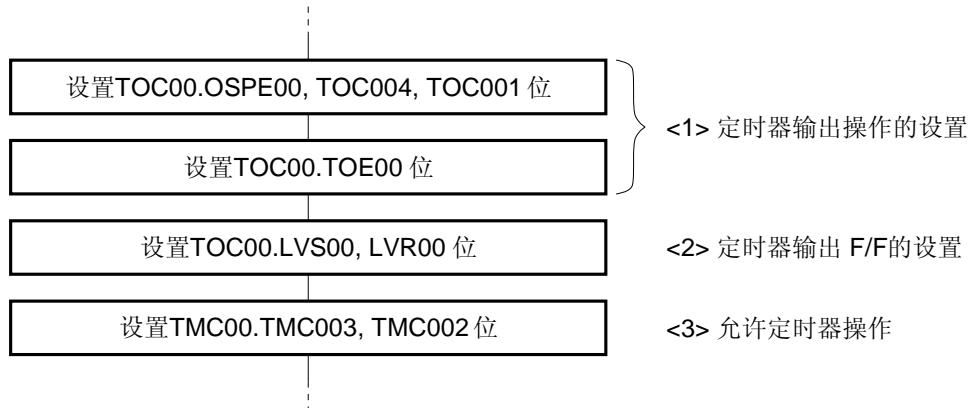
LVS00 和 LVR00 用于设置 TO00 引脚输出的默认值，也用于在没有允许定时器操作（TMC003 和 TMC002 = 00）的情况下反转定时器输出。在不需要软件控制时将 LVS00 和 LVR00 清除为 00（默认值：低电平输出）。

LVS00	LVR00	定时器输出状态
0	0	没有变化（低电平输出）
0	1	已清除（低电平输出）
1	0	设置（高电平输出）
1	1	禁止设置

(2) 设置 LVS00 和 LVR00

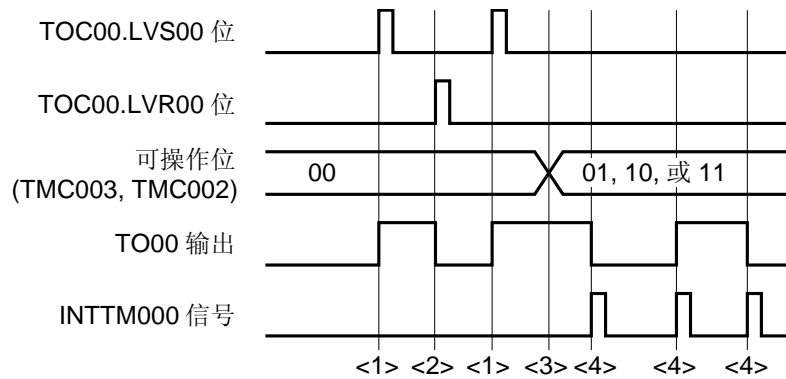
使用以下步骤来设置 LVS00 和 LVR00。

图 6-54. 设置 LVS00 和 LVR00 位的流程示例



注意事项 请务必按照上述步骤<1>, <2>, 和 <3>来对 LVS00 和 LVR00 进行设置。
步骤 <2>可以在<1>之后, <3>之前执行。

图 6-55. LVR00 和 LVS00 的时序示例



- <1> 当 LVS00 和 LVR00 = 10 时 TO00 引脚输出将会变高。
- <2> 当 LVS00 和 LVR00 = 01 时 TO00 引脚输出将会变低（即使 LVS00 和 LVR00 被清除为 00, 引脚输出也会从高电平开始保持不变）。
- <3> 当 TMC003 和 TMC002 被设为 01, 10, 或 11 时定时器开始操作。由于 LVS00 和 LVR00 在操作开始前被设为 10, TO00 引脚输出将会从高电平处开始。定时器开始操作后, 禁止对 LVS00 和 LVR00 进行设置直到 TMC003 和 TMC002 = 00（禁止定时器操作）。
- <4> 每次生成中断信号（INTTM000）时, TO00 引脚的输出电平都将反转。

6.6 16 位定时器 / 事件计数器 00 的注意事项

(1) 对于 16 位定时器 / 事件计数器 00 各个通道的限制

表 6-3 显示了对于各个通道的限制。

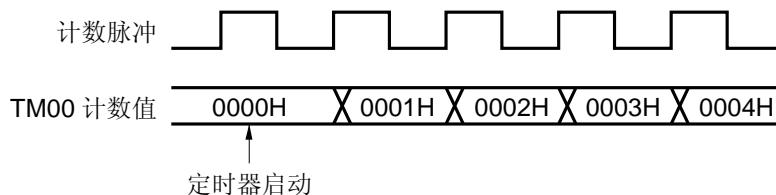
表 6-3. 16 位定时器 / 事件计数器 00 各个通道的限制

操作	限制
用作间隔定时器	-
用作方波输出	
用作外部事件计数器	
用作通过 TI000 引脚有效沿进入的清除&开始模式 ^注	当 TI010 引脚的有效沿进行检测时禁止使用定时器输出 (TO00)。(TOC00 = 00H)
用作自由运行定时器	-
用作 PPG 输出	$0000H \leq CR010 < CR000 \leq FFFFH$
用作单脉冲输出	禁止对 CR000 和 CR010 设置相同的值。
用作脉宽测量	禁止使用定时器输出 (TO00) (TOC00 = 00H)

(2) 定时器启动时的误差

在定时器启动后，生成一个匹配信号所需的时间内至少会发生一个时钟的误差。这是由于对 TM00 的计数没有与计数脉冲同步开始造成的。

图 6-56. TM00 计数的开始时序



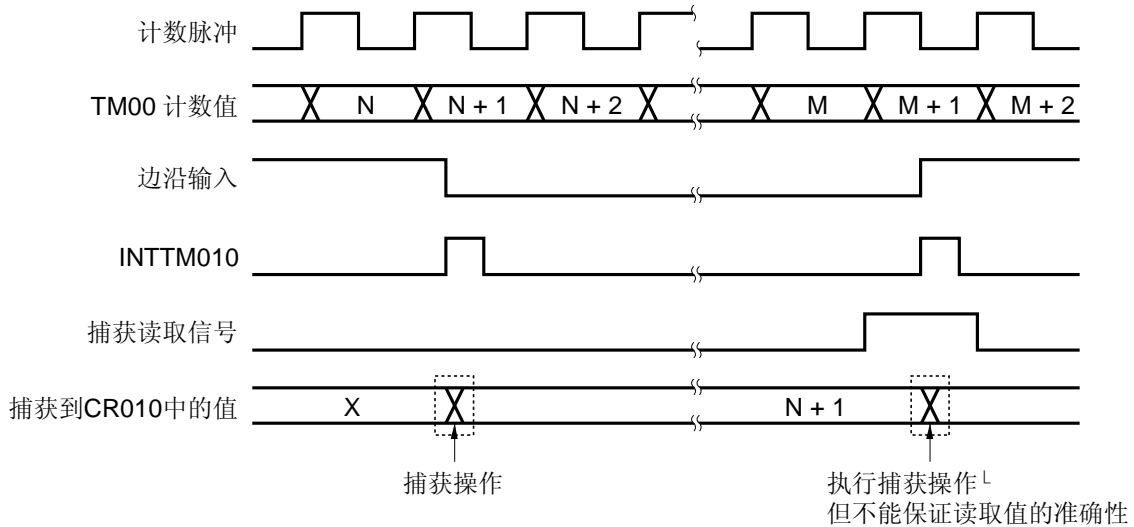
(3) CR000 和 CR010 的设置 (TM00 与 CR000 匹配的基础上进入的清除&开始模式)

向 CR000 和 CR010 中设定一个不同于 0000H 的值 (当 TM00 被用作外部事件计数器时 TM00 不能对脉冲进行计数)。

(4) 通过捕获寄存器保存数据的时序

- (a) 当有效沿被输入到 TI000/TI010 引脚中并且在读取 CR000/CR010 时检测到 TI000 引脚的反相时, CR010 将会执行捕获操作, 但 CR000/CR010 的读取值将不能被保证。此时, 当检测到 TI000/TI010 引脚的有效沿时将会生成一个中断信号 (INTTM000/INTTM010) (在检测到 TI000 引脚的反相边沿时将不会生成中断信号)。当计数值由于检测到 TI000/TI010 引脚的有效沿而被捕获时, 则将会在生成 INTTM000/INTTM010 后读取 CR000/CR010 的值。

图 6-57. 通过捕获寄存器保存数据的时序



- (b) 16 位定时器 / 事件计数器 00 停止后 CR000 和 CR010 的值将不能被保证。

(5) 设置有效沿

在定时器操作停止 (TMC003 和 TMC002 = 00) 时设置 TI000 引脚的有效沿。通过使用 ES000 和 ES001 来设置有效沿。

(6) 重新触发的单脉冲

确定在单脉冲输出模式中触发信号没有在有效电平输出时生成。在当前有效电平输出后请务必输入下一个触发信号。

(7) OVF00 标志的操作**(a) 设置 OVF00 标志 (1)**

OVF00 标志会在以下情况中被设为 1，也会在 TM00 溢出时被设为 1。

选择在 TM00 与 CR000 匹配的情况下进入的清除&开始模式。

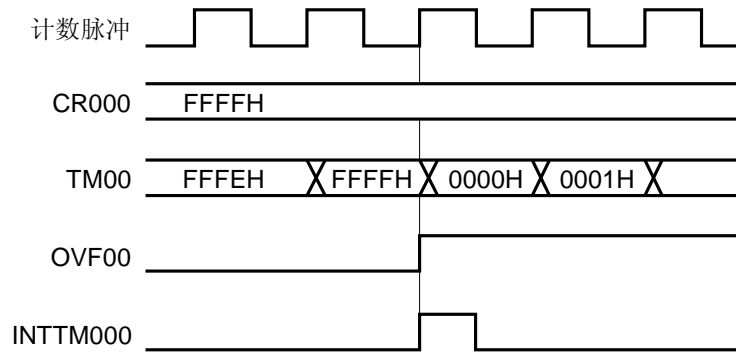
↓

将 CR000 设为 FFFFH。

↓

当 TM00 与 CR000 匹配且 TM00 从 FFFFH 清除为 0000H 时

图 6-58. OVF00 标志的操作时序

**(b) 清除 OVF00 标志**

即使 OVF00 标志在 TM00 溢出后，下个计数时钟被计数前被清除为 0 (TM00 的值变为 0001H 前)，它也将被再次设为 1，清除操作是无效的。

(8) 单脉冲输出

单脉冲输出在自由运行定时器模式或通过 TI000 引脚有效沿进入的清除&开始模式中正确进行。单脉冲不能在 TM00 与 CR000 匹配的情况下进入的清除&开始模式中输出。

(9) 捕获操作**(a) 当 TI000 的有效沿被指定为计数时钟时**

当 TI000 的有效沿被指定为主计数时钟时，TI000 被指定为触发器的捕获寄存器将不能正确运行。

(b) 通过输入到 TI010 和 TI000 引脚中的信号准确捕获值的脉宽

若要准确捕获计数值，那么输入到 TI000 和 TI010 引脚中作为捕获触发的脉冲必须比 PRM00 选择的两个计数时钟更宽（参见图 6-7）。

(c) 中断信号的生成

捕获操作在计数时钟的上升沿处进行，但中断信号（INTTM000 和 INTTM010）则是在下个计数时钟的上升沿处产生（参见图 6-7）。

(d) 当 CRC001（捕获 / 比较控制寄存器 00（CRC00）的第 1 位）被设为 1 时需要注意的事项

当 TM00 寄存器的计数值在与输入到 TI000 引脚中的信号相反的相位中被捕获至 CR000 寄存器时，计数值被捕获后将不能生成中断请求信号（INTTM000）。如果在该操作过程中在 TI010 引脚上检测到有效沿，那么将不会执行捕获操作，但 INTTM000 将会生成成为一个外部中断信号。在不使用外部中断时应将 INTTM000 信号屏蔽。

(10) 边沿检测**(a) 复位后指定有效边沿**

如果在 TI000 或 TI010 引脚处于高电平且 TI000 或 TI010 的有效沿被指定为上升沿或双边沿时允许 16 位定时器 / 事件计数器 00 的操作，那么 TI000 或 TI010 引脚的高电平将会被检测为上升沿。在 TI000 或 TI010 引脚上拉时需注意这一点。然而，一旦操作停止，那么将不能检测到上升沿，然后操作将再次被允许。

(b) 消除噪声的采样时钟

用于消除噪声的采样时钟会根据 TI000 有效沿是用作计数时钟还是捕获时钟而不同。在先前的情况中，采样时钟固定为 f_{PRS} 。接下去，由 PRM00 选择的计数时钟将会用于采样。

当输入到 TI000 引脚中的信号被采样且有效电平被连续检测到两次时，将检测到有效沿。因此，可以消除具有短脉宽的噪声（见图 6-7）。

(11) 定时器操作

当定时器停止时，不管 CPU 的操作模式是何，输入到 TI000/TI010 引脚中的信号均不能被响应。

备注 f_{PRS} : 周边硬件时钟频率

第七章 8 位定时器 / 事件计数器 50 和 51

7.1 8 位定时器 / 事件计数器 50 和 51 的功能

8 位定时器 / 事件计数器 50 和 51 具有以下功能。

- 间隔定时器
- 外部事件计数器
- 方波输出
- PWM 输出

7.2 8 位定时器 / 事件计数器 50 和 51 的配置

8 位定时器 / 事件计数器 50 和 51 包括以下硬件。

表 7-1. 8 位定时器 / 事件计数器 50 和 51 的配置

项目	配置
定时器寄存器	8 位定时器计数器 5n (TM5n)
寄存器	8 位定时器比较寄存器 5n (CR5n)
定时器输入	TI5n
定时器输出	TO5n
控制寄存器	定时器时钟选择寄存器 5n (TCL5n) 8 位定时器模式控制寄存器 5n (TMC5n) 端口模式寄存器 0 (PM0) 端口寄存器 0 (P0) 端口输出模式寄存器 0 (POM0)

图 7-1 和 7-2 显示了 8 位定时器 / 事件计数器 50 和 51 的框图。

图 7-1. 8 位定时器 / 事件计数器 50 的框图

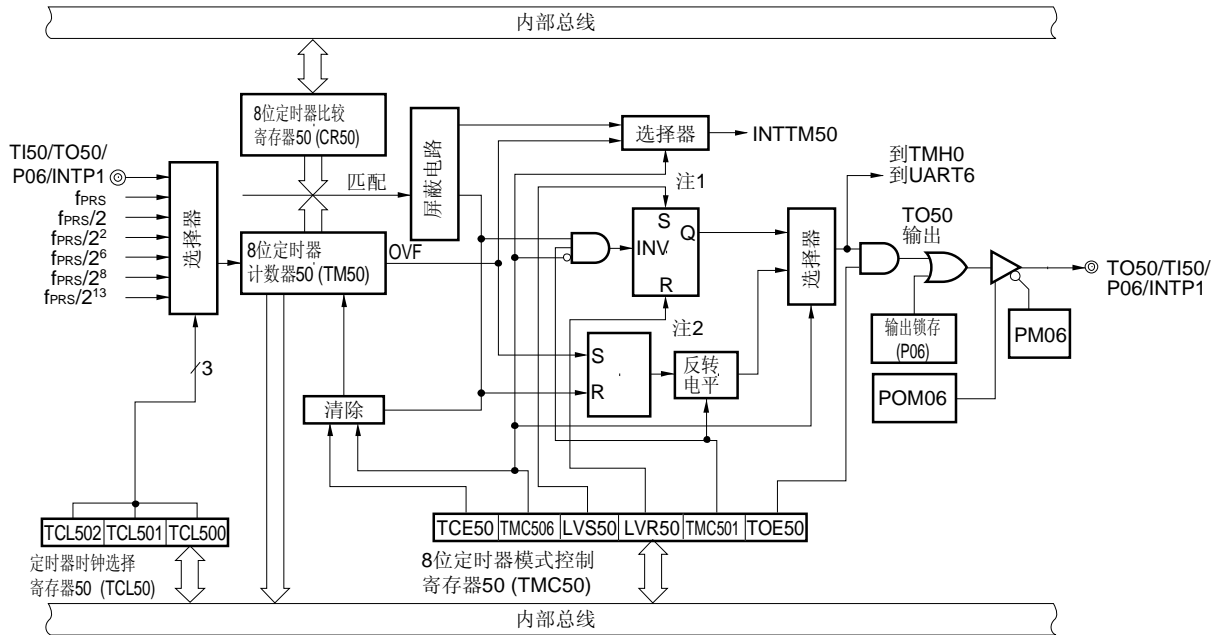
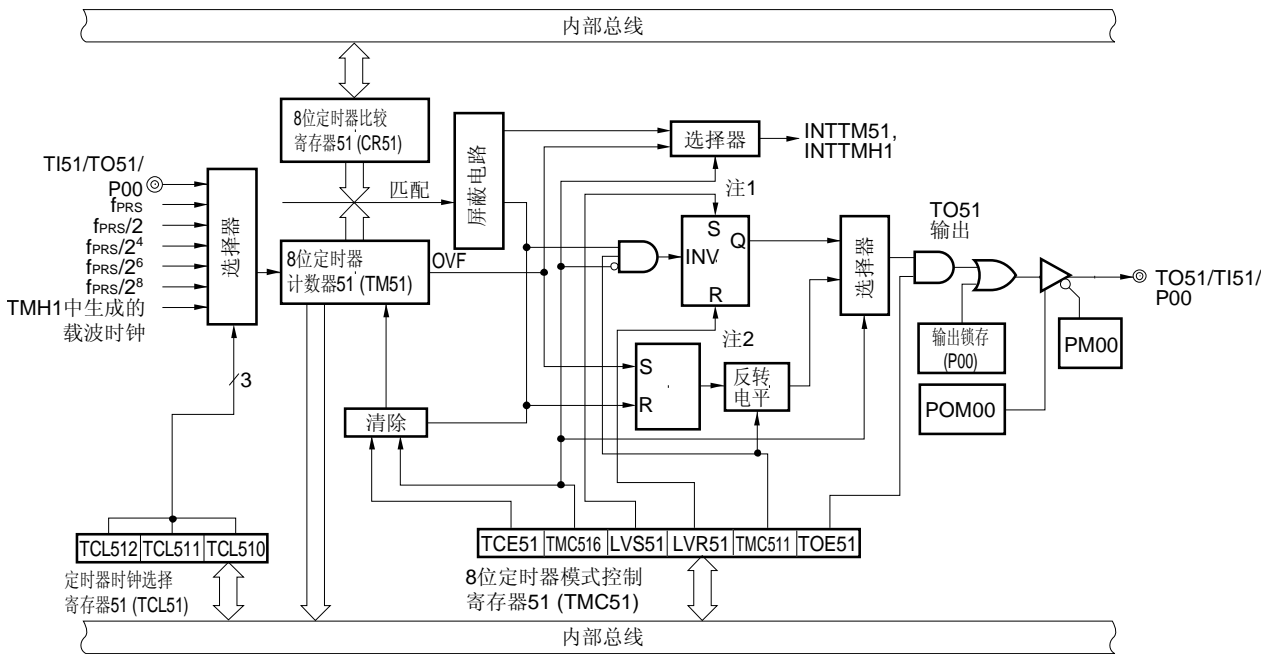


图 7-2. 8 位定时器 / 事件计数器 51 的框图



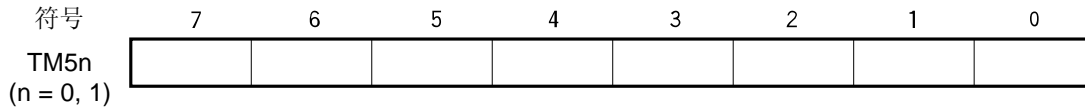
- 注 1. 定时器输出 F/F
 注 2. PWM 输出 F/F

(1) 8 位定时计数器 5n (TM5n)

TM5n 是一个 8 位寄存器，用于对计数脉冲进行计数，它是只读的。
计数器与计数时钟的上升沿同步累加。

图 7-3. 8 位定时计数器 5n (TM5n) 的格式

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



在以下几种情况下将计数值清除为 00H。

- <1> 产生复位信号
- <2> TCE5n 清零
- <3> 在 TM5n 与 CR5n 匹配时清零&启动模式中，当 TM5n 和 CR5n 匹配时。

(2) 8 位定时器比较寄存器 5n (CR5n)

CR5n 可以通过 8 位存储器操作指令来进行读取和写入。

除了 PWM 模式外，CR5n 的值将与 8 位定时器计数器 5n (TM5n) 的计数值频繁比较，如果结果匹配则将产生中断请求 (INTTM5n)。

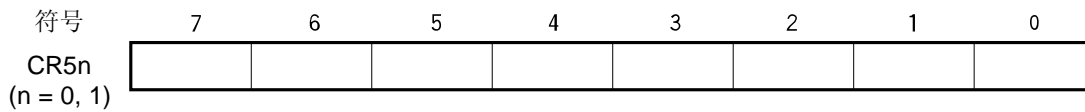
在 PWM 模式下，TM5n 与 CR5n 的值匹配时，TO5n 输出非有效电平，但不产生中断信号。

CR5n 的取值范围为 00H 到 FFH。

复位信号的生成将 CR5n 清除为 00H。

图 7-4. 8 位定时器比较寄存器 5n (CR5n) 的格式

地址: FF17H (CR50), FF41H (CR51) 复位后: 00H R/W



- 注意事项**
1. 在 TM5n 与 CR5n (TMC5n6 = 0) 匹配时清零&启动模式中，不要在操作过程中将其它值写入 CR5n。
 2. 在 PWM 模式中，以至少 3 个计数时钟（由 TCL5n 选择的时钟）间隔对 CR5n 进行重写。

备注 n = 0, 1

7.3 控制 8 位定时器 / 事件计数器 50 和 51 的寄存器

以下四种寄存器用于控制 8 位定时器 / 事件计数器 50 和 51。

- 定时器时钟选择寄存器 5n (TCL5n)
- 8 位定时器模式控制寄存器 5n (TMC5n)
- 端口模式寄存器 0 (PM0)
- 端口寄存器 0 (P0)
- 端口输出模式寄存器 0 (POM0)

(1) 定时器时钟选择寄存器 5n (TCL5n)

该寄存器设置 8 位定时器 / 事件计数器 5n 的计数时钟以及 TI5n 引脚输入的有效沿。

TCL5n 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将 TCL5n 清除为 00H。

备注 n = 0, 1

图 7-5. 定时器时钟选择寄存器 50 (TCL50) 的格式

地址: FF6AH 复位后: 00H R/W

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

TCL502	TCL501	TCL500	计数时钟选择		
				f _{PRS} = 2 MHz	f _{PRS} = 4 MHz
0	0	0	TI50 引脚下降沿		
0	0	1	TI50 引脚上升沿		
0	1	0	f _{PRS}	2 MHz	4 MHz
0	1	1	f _{PRS} /2	1 MHz	2 MHz
1	0	0	f _{PRS} /2 ²	500 kHz	1 MHz
1	0	1	f _{PRS} /2 ⁶	31.25 kHz	62.25 kHz
1	1	0	f _{PRS} /2 ⁸	7.81 kHz	15.6 kHz
1	1	1	f _{PRS} /2 ¹³	0.24 kHz	0.488 kHz

注意事项 1. 如果要向 TCL50 写入其它值, 则必须提前停止定时器操作。

2. 确保将位 3 至位 7 清除为“0”。

备注 f_{PRS}: 周边硬件时钟频率

图 7-6. 定时器时钟选择寄存器 51 (TCL51) 的格式

地址: FF8CH 复位后: 00H R/W

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

TCL512	TCL511	TCL510	计数时钟选择		
				fPRS = 2 MHz	fPRS = 4 MHz
0	0	0	TI51 引脚下降沿		
0	0	1	TI51 引脚上升沿		
0	1	0	fPRS	2 MHz	4MHz
0	1	1	fPRS/2	1 MHz	2MHz
1	0	0	fPRS/2 ⁴	125 kHz	250 kHz
1	0	1	fPRS/2 ⁶	31.25 kHz	62.5 kHz
1	1	0	fPRS/2 ⁸	7.81 kHz	15.6 kHz
1	1	1	TMH1 产生的载波时钟	-	-

- 注意事项
1. 如果要向 TCL51 写入其它值, 则必须提前停止定时器操作。
 2. 确保将位 3 至位 7 清除为 “0”。

备注 fPRS: 周边硬件时钟频率

(2) 8 位定时器模式控制寄存器 5n (TMC5n)

TMC5n 主要执行以下五种设置操作。

- <1> 8 位定时器计数器 5n (TM5n) 计数操作控制
- <2> 8 位定时器计数器 5n (TM5n) 操作模式选择
- <3> 定时器输出 F/F (flip flop) 状态设置
- <4> 定时器 F/F 控制或 PWM (自由运行) 模式中的有效电平选择
- <5> 定时器输出控制

TMC5n 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器清除为 00H。

备注 n = 0, 1

图 7-7. 8 位定时器模式控制寄存器 50 (TMC50) 的格式

地址: FF6BH 复位后: 00H R/W[※]

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

TCE50	TM50 计数操作控制
0	清零后, 禁止计数操作 (计数器停止操作)
1	计数操作开始

TMC506	TM50 操作模式选择
0	TM50 和 CR50 匹配时产生的清零 & 启动模式
1	PWM (自由运行) 模式

LVS50	LVR50	定时器输出 F/F 状态设置
0	0	不变
0	1	定时器输出 F/F 清除 (0) (TO50 输出的默认值: 低电平)
1	0	定时器输出 F/F 设置 (1) (TO50 输出的默认值: 高电平)
1	1	禁止设置

TMC501	其他模式中 (TMC506 = 0)	PWM 模式中 (TMC506 = 1)
	定时器 F/F 控制	
0	禁止反转操作	高电平有效
1	允许反转操作	低电平有效

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

注 第 2 位和第 3 位为只写操作位。

(注意事项和备注 列于下页中。)

图 7-8. 8 位定时器模式控制寄存器 51 (TMC51) 的格式

地址: FF43H 复位后: 00H R/W[※]

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

TCE51	TM51 计数操作控制	
0	清零后禁止计数操作 (计数器停止操作)	
1	计数操作开始	

TMC516	TM51 操作模式选择	
0	TM51 和 CR51 匹配时产生的清零 & 启动模式	
1	PWM (自由运行) 模式	

LVS51	LVR51	定时器输出 F/F 状态设置
0	0	不变
0	1	定时器输出 F/F 清除 (0) (TO51 输出的默认值: 低电平)
1	0	定时器输出 F/F 设置 (1) (TO51 输出的默认值: 高电平)
1	1	禁止设置

TMC511	其他模式中 (TMC516 = 0)	PWM 模式中 (TMC516 = 1)
	定时器 F/F 控制	
0	禁止反转操作	高电平有效
1	允许反转操作	低电平有效

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

注 第 2 位和第 3 位为只写操作位。

- 注意事项
- 除 PWM 模式外, 对 LVS5n 与 LVR5n 的设置有效。
 - 按下列顺序执行<1> 到 <4>, 注意不能同时执行。
 - <1> 设置 TMC5n1, TMC5n6: 操作模式设置
 - <2> 设置 TOE5n, 允许输出: 允许定时器输出
 - <3> 设置 LVS5n, LVR5n (参见注意事项 1): 定时器 F/F 设置
 - <4> 设置 TCE5n
 - 当 TCE5n = 1 时, 禁止设置 TMC5n 其他位。
 - 除 TO5n 输出外, 实际 TO50/TI50/P06/INTP1 和 TO51/TI51/P00 引脚输出由 PM06 和 P06, 以及 PM00 和 P00 决定。

- 备注
- 在 PWM 模式中, 将 TCE5n 清零可禁止 PWM 输出。
 - 如果读取了 LVS5n 和 LVR5n, 则值为 0。
 - 无论 TCE5n 的值是什么, TMC5n6, LVS5n, LVR5n, TMC5n1 和 TOE5n 位的值都会反应在 TO5n 输出中。
 - n = 0, 1

(3) 端口模式寄存器 0 (PM0)

该寄存器以 1 位为单位对端口 0 的输入 / 输出进行设置。

当使用 P06/TO50/TI50/INTP1 和 P00/TO51/TI51 引脚作为定时器输出时，将 PM06 和 PM00 清零并将 P06 和 P00 的输出锁存清零。

当定时器输入使用 P06/TO50/TI50/INTP1 和 P00/TO51/TI51 引脚时，应将 PM06 和 PM00 设为 1。此时，P06 和 P00 的输出锁存可能为 0 或 1。

PM0 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将这些寄存器设置为 FFH。

图 7-9. 端口模式寄存器 0 (PM0) 的格式

地址: FF20H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	P0n 引脚 I/O 模式选择 (n = 0 到 7)
0	输出模式 (输出缓冲开)
1	输入模式 (输出缓冲关)

(4) 端口输出模式寄存器 (POM0)

该寄存器用于设置端口 0 的输出模式。

该寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将这些寄存器设置为 00H。

图 7-10. 上拉电阻选择寄存器的格式

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	POM07	POM06	POM05	POM04	POM03	POM02	POM01	POM00	FF38H	00H	R/W

POM0n	P0n 引脚输出模式选择 (n = 0 到 7)
0	CMOS 输出
1	N-ch 开漏输出 (P07: P-ch 开漏输出)

7.4 8 位定时器 / 事件计数器 50 和 51 的操作

7.4.1 用作间隔定时器

8 位定时器 / 事件计数器 5n 用作间隔定时器，可以在间隔时间内不断产生中断请求，该间隔时间由 8 位定时器比较寄存器 5n (CR5n) 的预置值决定。

当 8 位定时器计数器 5n (TM5n) 中的计数值与 CR5n 中的值相等时，将 TM5n 清零并继续计数，同时产生中断请求信号 (INTTM5n)。

根据定时器时钟选择寄存器 5n (TCL5n) 的第 0 位到第 2 位 (TCL5n0 到 TCL5n2) 可以选择 TM5n 的计数时钟。

设置

<1> 设置寄存器。

- TCL5n: 选择计数时钟。
- CR5n: 比较值。
- TMC5n: 停止计数操作，选择 TM5n 与 CR5n 匹配时清零 & 启动模式。
(TMC5n = 0000xxx0B x = 不考虑)

<2> 将 TCE5n 置 1 后，开始计数操作。

<3> 如果 TM5n 与 CR5n 值匹配，则将产生 INTTM5n (TM5n 被清除为 00H)。

<4> INTTM5n 会在相同的间隔时间内重复生成。

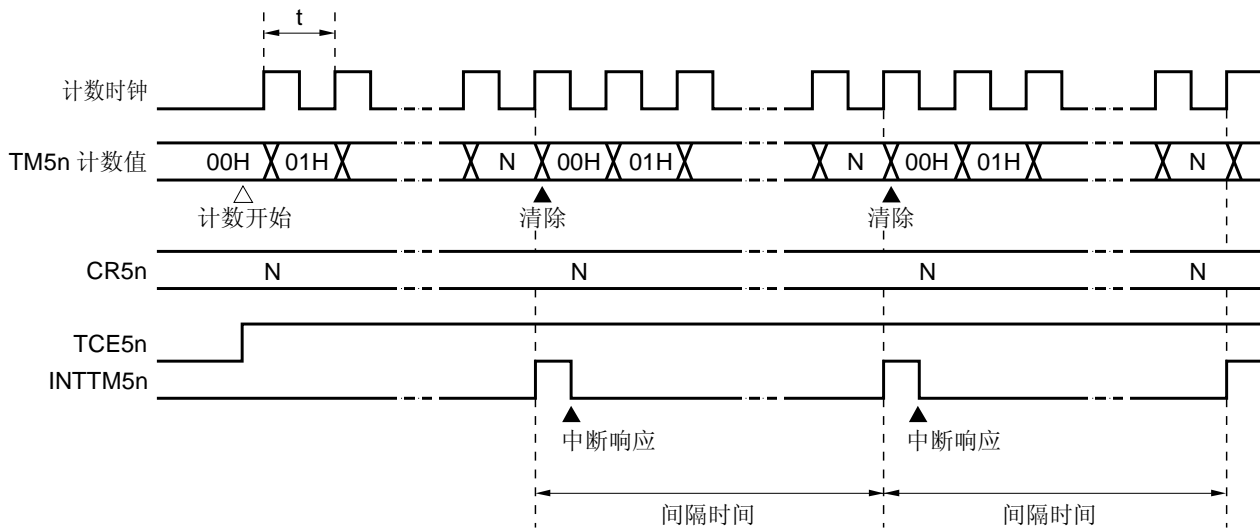
将 TCE5n 清零来停止计数操作。

注意事项 操作期间不要向 CR5n 写入其它值。

- 备注**
1. 如需了解如何允许 INTTM5n 中断，请参见第十一章 中断功能。
 2. n = 0, 1

图 7-11. 间隔定时器操作时序 (1/2)

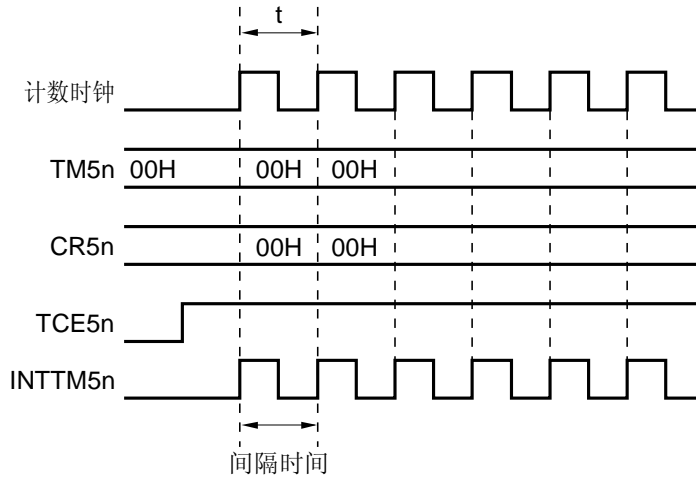
(a) 基本操作



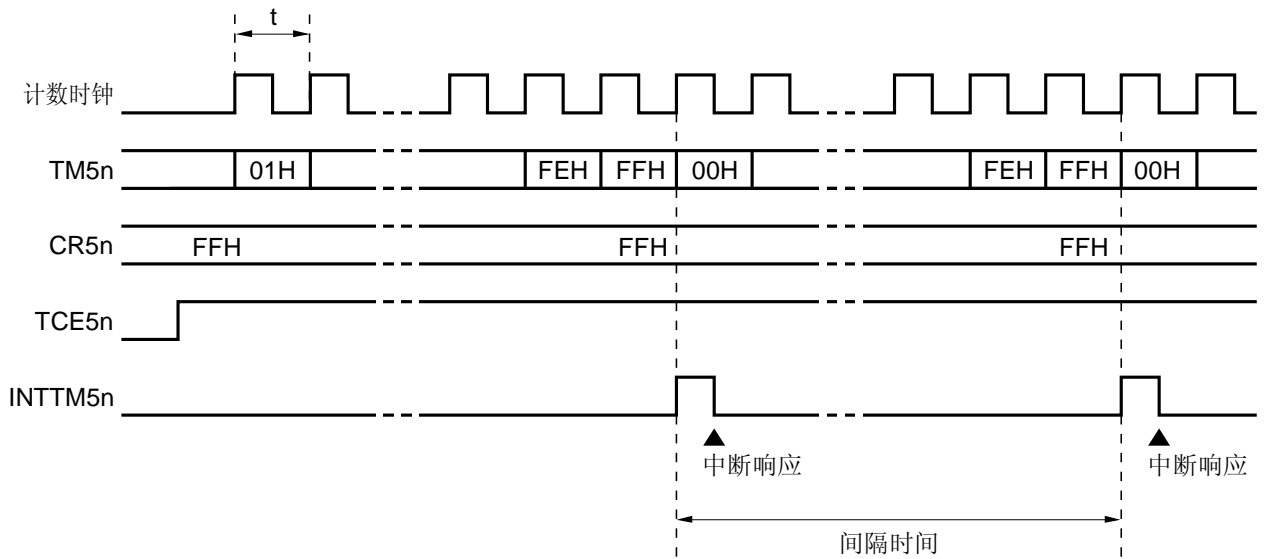
- 备注**
- 间隔时间 = (N + 1) × t
 N = 01H 到 FFH
 n = 0, 1

图 7-11. 间隔定时器操作时序 (2/2)

(b) 当 CR5n = 00H 时



(c) 当 CR5n = FFH 时



备注 n = 0, 1

7.4.2 用作外部事件计数器

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

每次输入由定时器时钟选择寄存器 5n (TCL5n) 指定的有效沿时 TM5n 都将会递增。可以选择上升沿或下降沿。

当 TM5n 中的计数值与 8 位定时器比较寄存器 5n (CR5n) 中的值相匹配时, TM5n 将会被清零, 并产生中断请求信号 (INTTM5n)。

无论在什么时候, 只要 TM5n 的值与 CR5n 的值匹配都将会生成 INTTM5n。

设置

<1> 设置各个寄存器。

- 设置端口模式寄存器 (PM06 或 PM00) ^{*} 的值为 1。
- TCL5n: 选择 TI5n 引脚输入沿。
TI5n 引脚下降沿 → TCL5n = 00H
TI5n 引脚上升沿 → TCL5n = 01H
- CR5n: 比较值
- TMC5n: 停止计数操作, 选择 TM5n 与 CR5n 匹配时发生清零 & 启动的模式, 禁止定时器 F/F 反转操作, 禁止定时器输出。
(TMC5n = 0000000B)

<2> 当设置 TCE5n = 1 时, 开始对从 TI5n 引脚输入的脉冲进行计数。

<3> 当 TM5n 与 CR5n 的值匹配时, 将产生 INTTM5n (TM5n 被清除为 00H)。

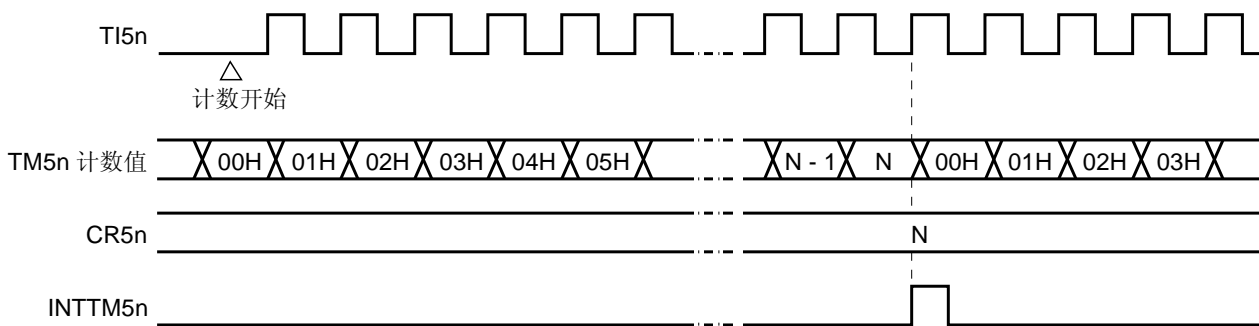
<4> 在经过以上设置后, 每当 TM5n 与 CR5n 的值匹配时都会产生 INTTM5n。

注 8 位定时器 / 事件计数器 50: PM06

8 位定时器 / 事件计数器 51: PM00

备注 如需了解如何允许 INTTM5n 中断, 请参见第十一章 中断功能。

图 7-12. 外部事件计数器操作的时序 (规定上升沿有效)



备注 N = 00H 到 FFH
n = 0, 1

7.4.3 方波输出操作

以 8 位定时器比较寄存器 5n (CR5n) 的预置值决定的间隔时间输出任选频率的方波。

通过将 8 位定时器模式控制寄存器 5n (TMC5n) 的第 0 位 (TOE5n) 置 1, TO5n 引脚输出状态可以在规定的间隔时间内反转, 该间隔时间由 CR5n 的预置值决定。这使得任选频率的方波 (占空比=50%) 都可以被输出。

设置

<1> 设置各个寄存器。

- 将端口输出锁存 (P06 或 P00) ^注 和端口模式寄存器 (PM06 或 PM00) ^注 清零。
- TCL5n: 选择计数时钟。
- CR5n: 比较值
- TMC5n: 停止计数操作, 选择 TM5n 与 CR5n 匹配时发生清零 & 启动的模式。

LVS5n	LVR5n	定时器输出 F/F 状态设置
0	1	定时器输出 F/F 清除 (0) (TO5n 输出的默认值: 低电平)
1	0	定时器输出 F/F 设置 (1) (TO5n 输出的默认值: 高电平)

允许定时器输出

(TMC5n = 00001011B 或 00000111B)

<2> 将 TCE5n 置 1 后, 开始计数操作。

<3> 定时器输出 F/F 通过 TM5n 和 CR5n 的匹配进行反转。生成 INTTM5n 后 TM5n 将会被清除为 00H。

<4> 在经过以上设置后, 定时器输出 F/F 会在相同间隔内反转, 并从 TO5n 输出一个方波。

频率如下。

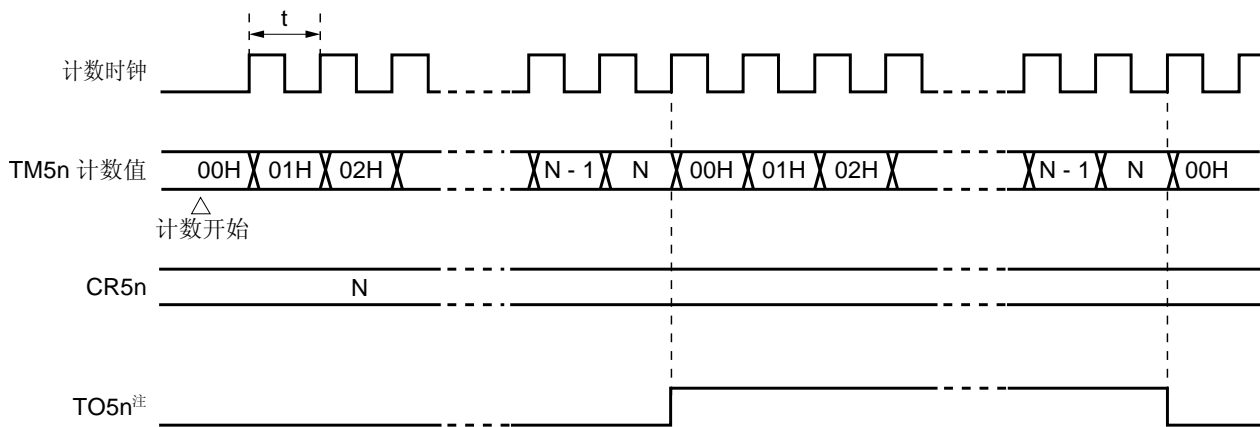
- 频率 = $1/2t (N + 1)$
(N: 00H 到 FFH)

注 8 位定时器 / 事件计数器 50: P06, PM06
8 位定时器 / 事件计数器 51: P00, PM00

注意事项 操作期间不要向 CR5n 写入其它值。

备注 1. 如需了解如何允许 INTTM5n 中断, 请参见第十一章 中断功能。
2. n = 0, 1

图 7-13. 方波输出操作的时序



注 TO5n 输出的初始值可以通过 8 位定时器模式控制寄存器 5n (TMC5n) 的第 2 位和第 3 位 (LVR5n, LVS5n) 来进行设置。

7.4.4 PWM 输出操作

当 8 位定时器模式控制寄存器 5n (TMC5n) 的第 6 位 (TMC5n6) 置 1 时, 8 位定时器 / 事件计数器 5n 将作为 PWM 输出使用。

占空比由 8 位定时器比较寄存器 5n (CR5n) 的值决定, 并从 TO5n 引脚输出。

将 PWM 脉冲的有效电平宽度设置到 CR5n; 可以由 TMC5n 的第 1 位 (TMC5n1) 选择有效电平。

根据定时器时钟选择寄存器 5n (TCL5n) 的第 0 位到第 2 位 (TCL5n0 到 TCL5n2) 可以选择计数时钟。

可以由 TMC5n 的第 0 位 (TOE5n) 决定允许 / 禁止 PWM 输出。

注意事项 在 PWM 模式中, 使 CR5n 的重写周期变为 3 个或更多个计数时钟 (由 TCL5n 选择时钟)。

备注 n = 0, 1

(1) PWM 输出的基本操作

设置

<1> 设置各个寄存器。

- 将端口输出锁存 (P06 或 P00)[※] 和端口模式寄存器 (PM06 或 PM00)[※] 清零。
- TCLK5n: 选择计数时钟。
- CR5n: 比较值
- TMC5n: 停止计数操作, 选择 PWM 模式。
定时器输出 F/F 不会改变。

TMC5n1	有效电平选择
0	高电平有效
1	低电平有效

允许定时器输出

(TMC5n = 01000001B 或 01000011B)

<2> 当 TCE5n = 1 时, 计数开始。

将 TCE5n 清零来停止计数操作。

注 8 位定时器 / 事件计数器 50: P06, PM06

8 位定时器 / 事件计数器 51: P00, PM00

PWM 输出操作

<1> PWM 输出 (TO5n 输出) 一直输出非有效电平直至产生溢出。

<2> 当发生溢出时, 输出有效电平。直至 CR5n 与 8 位定时器计数器 5n (TM5n) 的计数值匹配时才会输出有效电平。

<3> CR5n 与计数值匹配后, 将输出非有效电平直至溢出再次发生。

<4> 重复 <2> 和 <3> 的操作, 直至计数操作停止。

<5> 由于 TCE5n=0 而停止计数操作时, PWM 输出将变为非有效电平。

关于时序的详细信息, 请参见图 7-14 和 7-15。

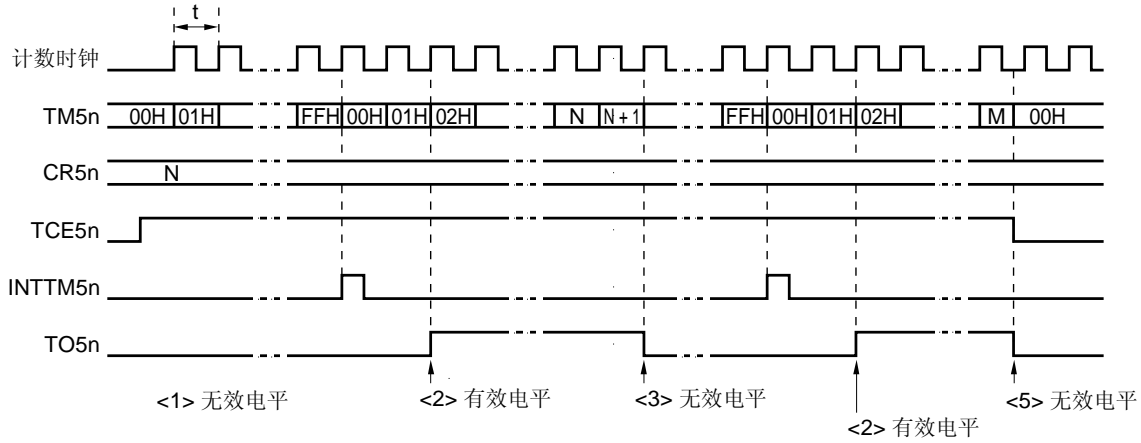
周期, 有效电平宽度以及占空比如下所示。

- 周期 = $2^8 t$
- 有效电平宽度 = Nt
- 占空比 = $N/2^8$
(N = 00H 到 FFH)

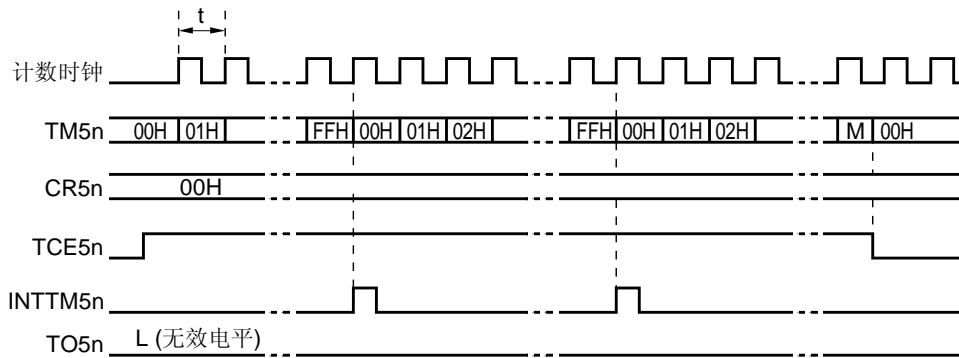
备注 n = 0, 1

图 7-14. PWM 输出操作的时序

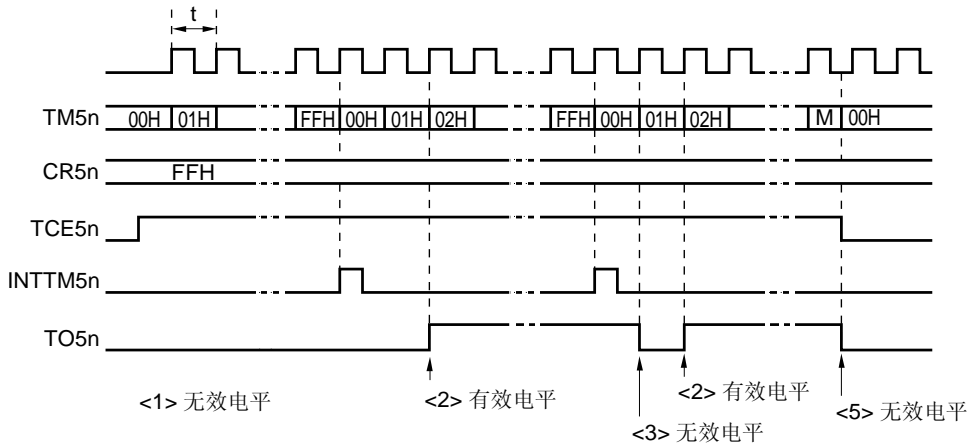
(a) 基本操作 (有效电平 = H)



(b) CR5n = 00H



(c) CR5n = FFH

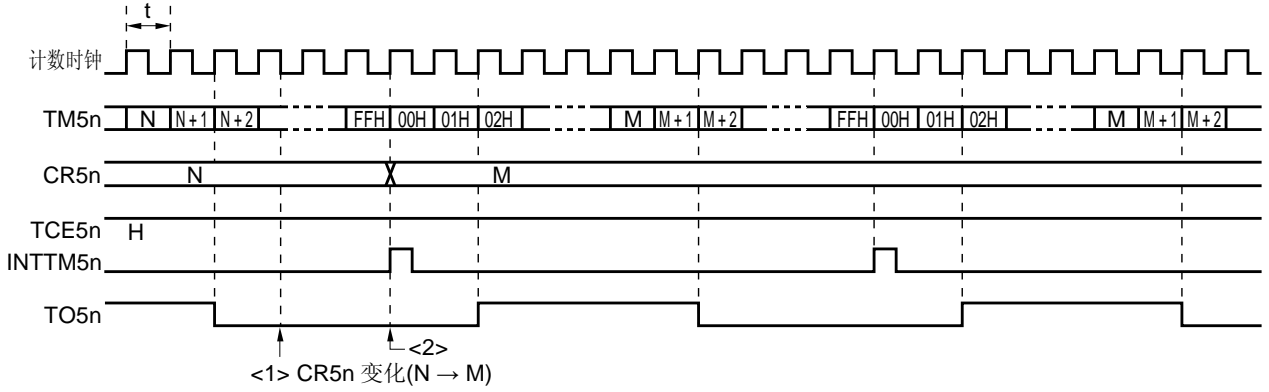


- 备注
1. 图 7-14 (a) 中的<1> 到 <3> 和 <5> 与 7.4.4 (1) PWM 输出基本操作的 PWM 输出操作中<1> 到 <3> 和 <5>相对应。
 2. $n = 0, 1$

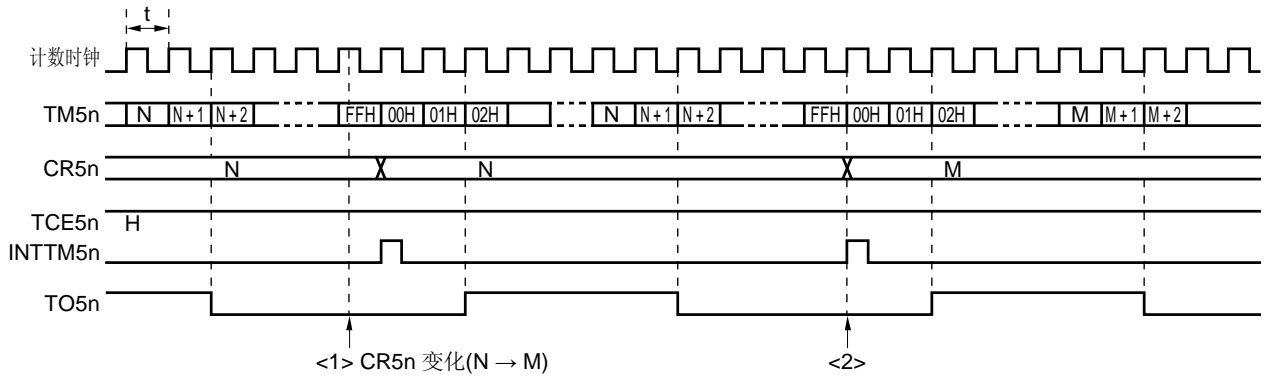
(2) CR5n 改变时的操作

图 7-15. 改变 CR5n 时的操作时序

(a) 在计数值为 FFH 的时钟上升沿出现之前，CR5n 的值由 N 变为 M
 → 改变后，溢出时立即将值传送至 CR5n。



(b) 在计数值为 FFH 的时钟上升沿出现之后，CR5n 的值由 N 变为 M
 → 发生第二次溢出时将值传送 CR5n。



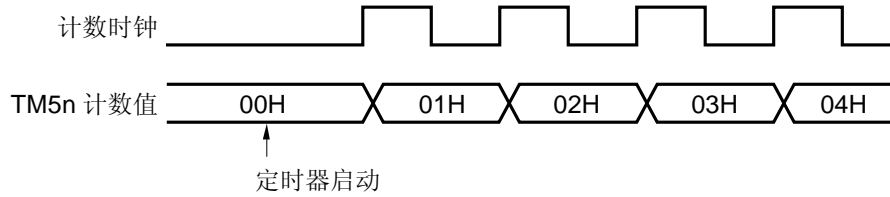
注意事项 在图 7-15 中的<1>与<2>之间读取 CR5n 时，读取的值与实际值是不同的（读取的值：M，CR5n 的实际值：N）。

7.5 8 位定时器 / 事件计数器 50 和 51 的注意事项

(1) 定时器启动时的误差

在定时器启动后，生成一个匹配信号所需的时间内至少会发生一个时钟的误差。这是由于 8 位定时器计数器 50 和 51 (TM50, TM51) 的启动与计数时钟不同步引起的。

图 7-16. 8 位定时器计数器 5n 启动时序



备注 n = 0, 1

第八章 8 位定时器 H0 和 H1

8.1 8 位定时器 H0 和 H1 的功能

8 位定时器 H0 和 H1 具有以下功能。

- 间隔定时器
- 方波输出
- PWM 输出
- 用于遥控的载波发生器模式（仅用于 8 位定时器 H1）

8.2 8 位定时器 H0 和 H1 的配置

8 位定时器 H0 和 H1 包括以下硬件。

表 8-1. 8 位定时器 H0 和 H1 的配置

项目	配置
定时器寄存器	8 位定时器计数器 Hn
寄存器	8 位定时器 H 比较寄存器 0n (CMP0n) 8 位定时器 H 比较寄存器 1n (CMP1n)
定时器输出	TOH0, TOH1/REM, 输出控制器
控制寄存器	8 位定时器 H 模式寄存器 n (TMHMDn) 8 位定时器 H 载波控制寄存器 1 (TMCYC1) ^注 端口模式寄存器 0 (PM0) 端口寄存器 0 (P0) 端口输出模式寄存器 0 (PM0)

注 仅适用于 8 位定时器 H1

备注 n = 0, 1

图 8-1 和 8-2 显示了其框图。

图 8-1. 8 位定时器 H0 的框图

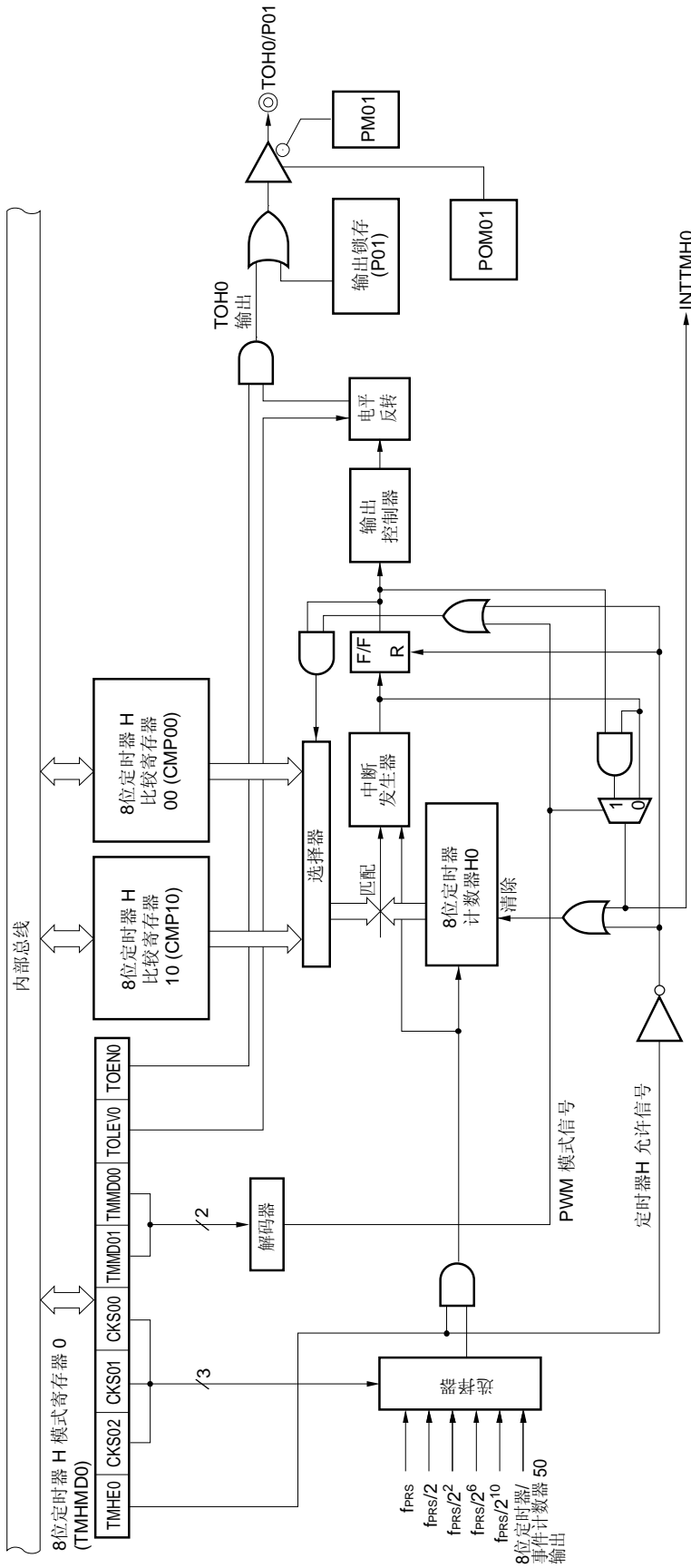
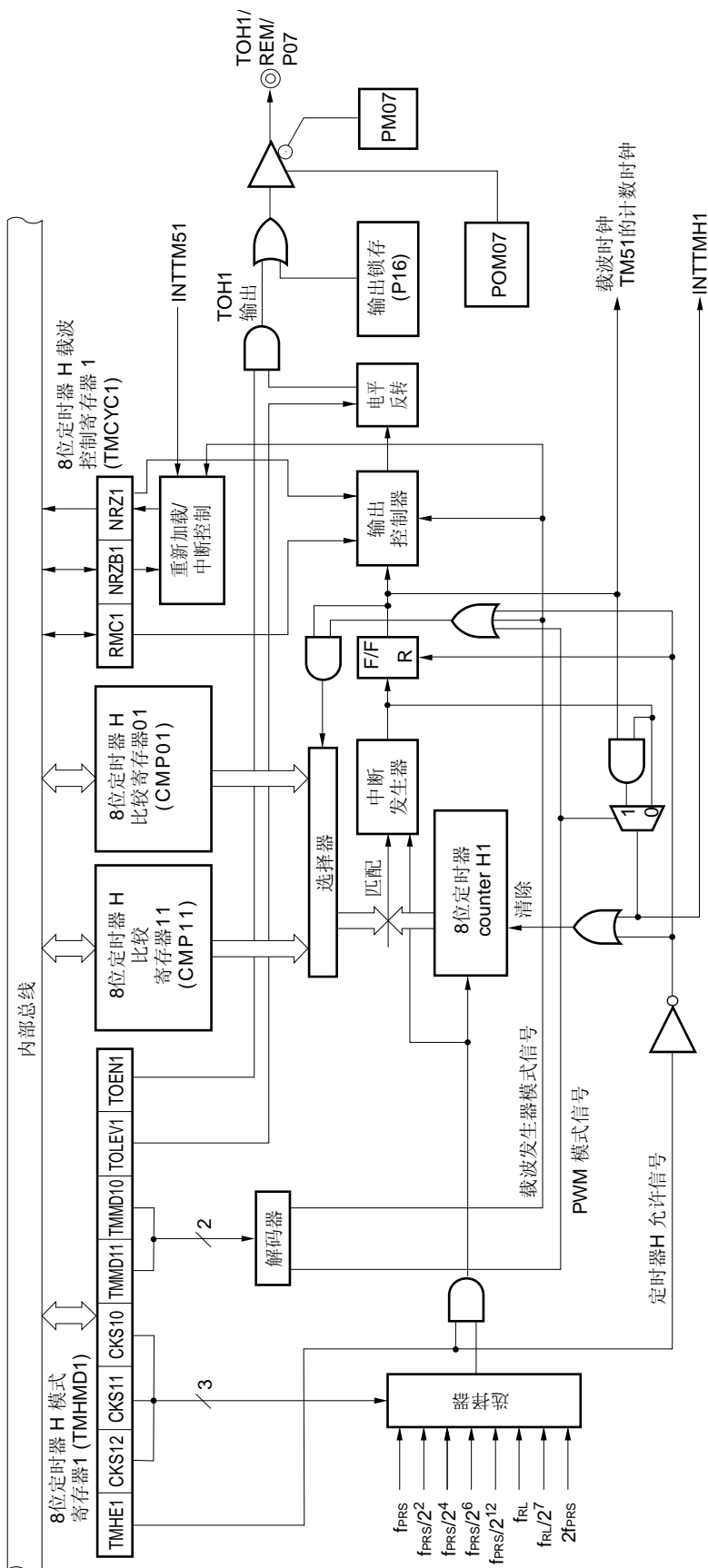


图 8-2. 8 位定时器 H1 的框图



(1) 8 位定时器 H 比较寄存器 0n (CMP0n)

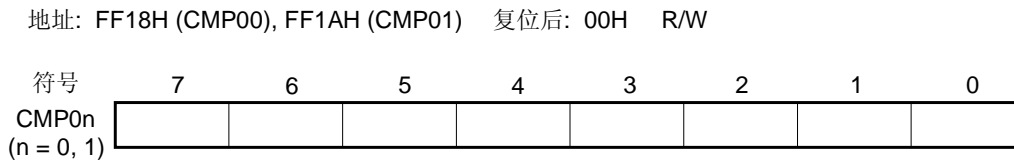
该寄存器可通过 8 位存储器操作指令进行读写。该寄存器用于所有的定时器操作模式。

该寄存器不断地将 CMP0n 的值和 8 位定时器计数器 Hn 的计数值进行比较，并且当两个值匹配时，将会产生一个中断请求信号 (INTTMHn) 并反转 TOHn 的输出电平。

当定时器停止时 (TMHEn = 0) 重写 CMP0n 的值。

复位信号的产生将该寄存器设置为 00H。

图 8-3. 8 位定时器 H 比较寄存器 0n (CMP0n) 的格式



注意事项 CMP0n 在定时器计数操作期间不能被重写。CMP0n 在定时器计数操作期间可以进行更新（写入相同的值）。

(2) 8 位定时器 H 比较寄存器 1n (CMP1n)

该寄存器可通过 8 位存储器操作指令进行读写。该寄存器用于 PWM 输出模式和载波发生器模式中。

在 PWM 输出模式中，该寄存器不断地将 CMP1n 的设置值和 8 位定时器计数器 Hn 的计数值进行比较，当两个值匹配时，反转 TOHn 的输出电平。不产生中断请求信号。

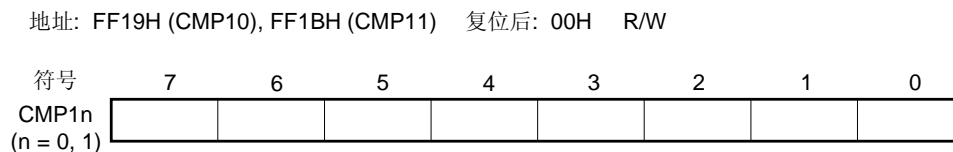
在载波发生器模式中，CMP1n 寄存器不断地将 CMP1n 的设置值和 8 位定时器计数器 Hn 的计数值进行比较，如果两个值匹配，则产生一个中断请求信号 (INTTMHn)。同时，计数值清零。

可以在定时器计数操作期间对 CMP1n 进行更新（写入相同值）和重写。

如果在定时器操作期间重写 CMP1n 的值，则当定时器计数器的值和 CMP1n 以前的值相匹配时，新的值将会被锁存并传输给 CMP1n，然后 CMP1n 的值则将会变为新值。如果计数值和 CMP1n 值的匹配操作与 CMP1n 的写操作发生冲突，那么 CMP1n 的值将保持不变。

复位信号的产生将该寄存器设置为 00H。

图 8-4. 8 位定时器 H 比较寄存器 1n (CMP1n) 的格式



注意事项 在 PWM 输出模式和载波发生器模式下，当停止定时器计数操作 (TMHEn = 0) 后再启动定时器计数操作 (TMHEn = 1) 时，必须设置 CMP1n（即使对 CMP1n 设置相同的值也要再设置一次）。

备注 n = 0, 1

8.3 控制 8 位定时器 H0 和 H1 的寄存器

以下四个寄存器可用于控制 8 位定时器 H0 和 H1。

- 8 位定时器 H 模式寄存器 n (TMHMDn)
- 8 位定时器 H 载波控制寄存器 1 (TMCYC1) ^注
- 端口模式寄存器 0 (PM0)
- 端口寄存器 0 (P0)

注 仅适用于 8 位定时器 H1

(1) 8 位定时器 H 模式寄存器 n (TMHMDn)

该寄存器用于控制定时器 H 的模式。

该寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器清除为 00H。

备注 n = 0, 1

图 8-5. 8 位定时器 H 模式寄存器 0 (TMHMD0) 的格式

地址: FF69H 复位后: 00H R/W

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

TMHE0	允许定时器操作
0	停止定时器计数操作 (计数器被清零)
1	允许定时器计数操作 (通过输入时钟开始操作)

CKS02	CKS01	CKS00	计数时钟选择		
				f _{PRS} = 2 MHz	f _{PRS} = 4 MHz
0	0	0	f _{PRS}	2 MHz	4 MHz
0	0	1	f _{PRS} /2	1 MHz	2 MHz
0	1	0	f _{PRS} /2 ²	500 kHz	1 MHz
0	1	1	f _{PRS} /2 ⁶	31.25 kHz	62.5 kHz
1	0	0	f _{PRS} /2 ¹⁰	1.95 kHz	3.91 kHz
1	0	1	TM50 输出 ^注		
其他情况			禁止设置		

TMMD01	TMMD00	定时器操作模式
0	0	间隔定时器模式
1	0	PWM 输出模式
其他情况		禁止设置

TOLEV0	定时器输出电平控制 (默认模式中)
0	低电平
1	高电平

TOEN0	定时器输出控制
0	禁止输出
1	允许输出

注 当选择 TM50 输出作为计数时钟时, 应注意下面几点。

- 计数时钟通过 TM50 和 CR50 的匹配来清零和启动 (TMC506 = 0) 的模式
首先启动 8 位定时器 / 事件计数器 50 的操作, 然后将允许定时器 F/F 的反转操作。(TMC501 = 1)。
- PWM 模式 (TMC506 = 1)
首先启动 8 位定时器 / 事件计数器 50 的操作, 然后设置计数时钟来使得占空比 = 50%。
任意模式下都没必要允许 (TOE50 = 1) TO50 输出。

注意事项 1. 当 TMHE0 = 1 时, 禁止设置 TMHMD0 中其他的位。然而, TMHMD0 可以进行更新 (写入相同的值)。

- 注意事项**
- 在 PWM 输出模式下，当定时器计数器操作停止（TMHE0 = 0）之后启动定时器计数操作（TMHE0 = 1）时，请务必对 8 位定时器 H 比较寄存器 10（CMP10）进行设置（即使对 CMP10 设置了相同的值也务必再设置一次）。
 - 除 TOH0 输出外，TOH0/P01 引脚的实际输出是根据 PM01 和 P01 来决定的。

- 备注**
- fPRS: 周边硬件时钟频率
 - TMC506: 8 位定时器模式控制寄存器 50（TMC50）的第 6 位
TMC501: TMC50 的第 1 位

图 8-6. 8 位定时器 H 模式寄存器 1（TMHMD1）的格式

地址: FF6CH 复位后: 00H R/W

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

TMHE1	允许定时器操作
0	停止定时器计数操作 (计数器被清零)
1	允许定时器计数操作 (通过输入时钟开始操作)

CKS12	CKS11	CKS10	计数时钟选择		
				fPRS = 2 MHz	fPRS = 4 MHz
0	0	0	fPRS	2 MHz	4 MHz
0	0	1	fPRS/2 ²	500 kHz	1 MHz
0	1	0	fPRS/2 ⁴	125 kHz	250 kHz
0	1	1	fPRS/2 ⁶	31.25 kHz	62.5 kHz
1	0	0	fPRS/2 ¹²	0.49 kHz	0.98 kHz
1	0	1	fRL/2 ⁷	1.80 kHz (TYP.)	
1	1	0	2PRS	4 kHz	8 MHz
1	1	1	fRL	240 kHz (TYP.)	

TMMD11	TMMD10	定时器操作模式
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	定时器输出电平控制 (默认模式中)
0	低电平
1	高电平

TOEN1	定时器输出控制
0	禁止输出
1	允许输出

<R>

- 注意事项**
1. 当 $TMHE1 = 1$ 时，禁止设置 $TMHMD1$ 中其他的位。然而， $TMHMD1$ 可以进行更新（写入相同的值）。
 2. 在 PWM 输出模式和载波发生模式中，当定时器停止计数后（ $TMHE1=0$ ）再次启动计数器操作时（ $TMHE1=1$ ）时，必须对 8 位定时器 H 比较寄存器 11（ $CMP11$ ）进行设置（即使对 $CMP11$ 设置了相同的值，也务必再设置一次）。
 3. 当使用载波发生器模式时，设置 $TMH1$ 的计数时钟频率至少是 $TM51$ 计数时钟频率的 6 倍。
 4. 除 $TOH1$ 输出外， $TOH1/REM/P07$ 引脚的实际输出是根据 $PM07$ 和 $P07$ 来决定的。

- 备注**
1. f_{PRS} : 周边硬件时钟频率
 2. f_{RL} : 内部低速振荡时钟频率

(2) 8 位定时器 H 载波控制寄存器 1（ $TMCYC1$ ）

该寄存器用于控制 8 位定时器 H1 的遥控输出和载波脉冲输出状态。

该寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器清除为 00H。

图 8-7. 8 位定时器 H 载波控制寄存器 1（ $TMCYC1$ ）的格式

地址: FF6DH 复位后: 00H R/W^注

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

RMC1	NRZB1	遥控输出
0	0	低电平输出
0	1	在 $INTTM51$ 信号输入的上升沿输出高电平
1	0	低电平输出
1	1	在 $INTTM51$ 信号输入的上升沿输出载波脉冲

NRZ1	载波脉冲输出状态标志
0	载波输出禁止状态 (低电平状态)
1	载波输出允许状态 ($RMC1 = 1$: 载波脉冲输出, $RMC1 = 0$: 高电平状态)

注 第 0 位是只读的。

注意事项 当 $TMHE1 = 1$ 时，不能重写 $RMC1$ 。然而， $TMCYC1$ 可以更新（写入相同的值）。

(3) 端口模式寄存器 0 (PM0)

该寄存器以 1 位为单位对端口 1 的输入 / 输出进行设置。

在将 P01/TOH0 和 P07/TOH1/REM 引脚用于定时器输出，将载波发生器输出用于遥控时，需将 PM01 和 PM07 以及 P01 和 P07 的输出锁存器清零。

PM0 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器设置为 FFH。

图 8-8. 端口模式寄存器 0 (PM0) 的格式

地址: FF20H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	P0n 引脚 I/O 模式选择 (n = 0 到 7)
0	输出模式 (输出缓冲开)
1	输入模式 (输出缓冲关)

(4) 端口输出模式寄存器 (POM0)

该寄存器用于设置端口 0 的输出模式。当为遥控载波发生器输出使用 P07/TOH1/REM 引脚时，应设置 POM07 为 1。

该寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将这些寄存器设置为 00H。

图 8-9. 上拉电阻选择寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
POM0	POM07	POM06	POM05	POM04	POM03	POM02	POM01	POM00	FF38H	00H	R/W

POM0n	P0n 引脚输出模式选择 (n = 0 到 7)
0	CMOS 输出
1	N-ch 开漏输出 (P07: P-ch 开漏输出)

8.4 8 位定时器 H0 和 H1 的操作

8.4.1 用作间隔定时器 / 方波输出

当 8 位定时器计数器 H_n 与比较寄存器 0_n (CMP0n) 匹配时, 将产生中断请求信号 (INTTMHn) 并将 8 位定时器计数器 H_n 清零 (00H)。

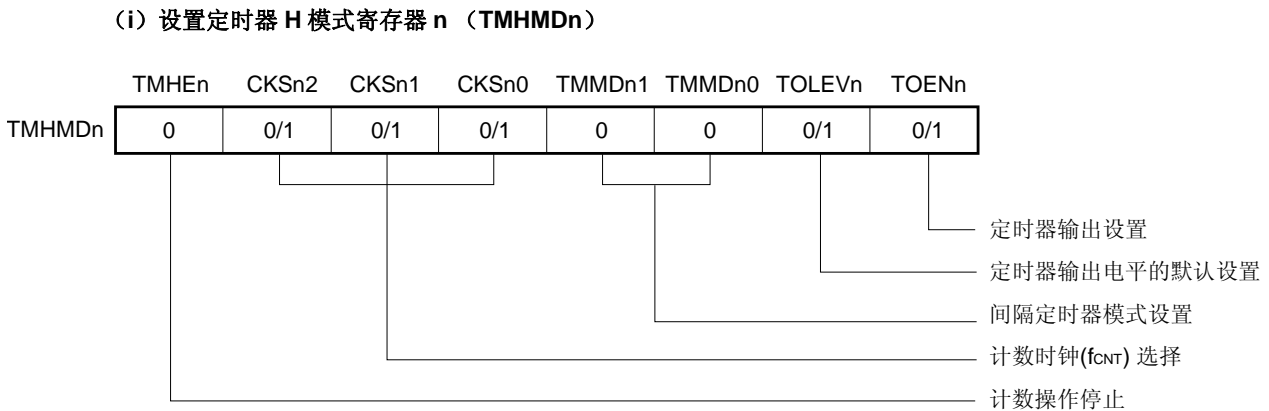
在间隔定时器模式中不使用比较寄存器 1_n (CMP1n)。即使已经对 CMP1n 寄存器进行了设置, 由于不会检测到 8 位定时器计数器 H_n 与 CMP1n 寄存器匹配的情况, 所以不影响定时器的输出。

通过将定时器 H 模式寄存器 n (TMHMDn) 的第 0 位 (TOENn) 置 1, 可以从 TOHn 输出任意频率的方波 (占空比 = 50%)。

设置

<1> 设置各个寄存器。

图 8-10. 间隔定时器 / 方波输出操作期间的寄存器设置



(ii) CMP0n 寄存器设置

如果 N 作为比较值, 则时间间隔如下。

- 间隔时间 = $(N + 1) / f_{CNT}$

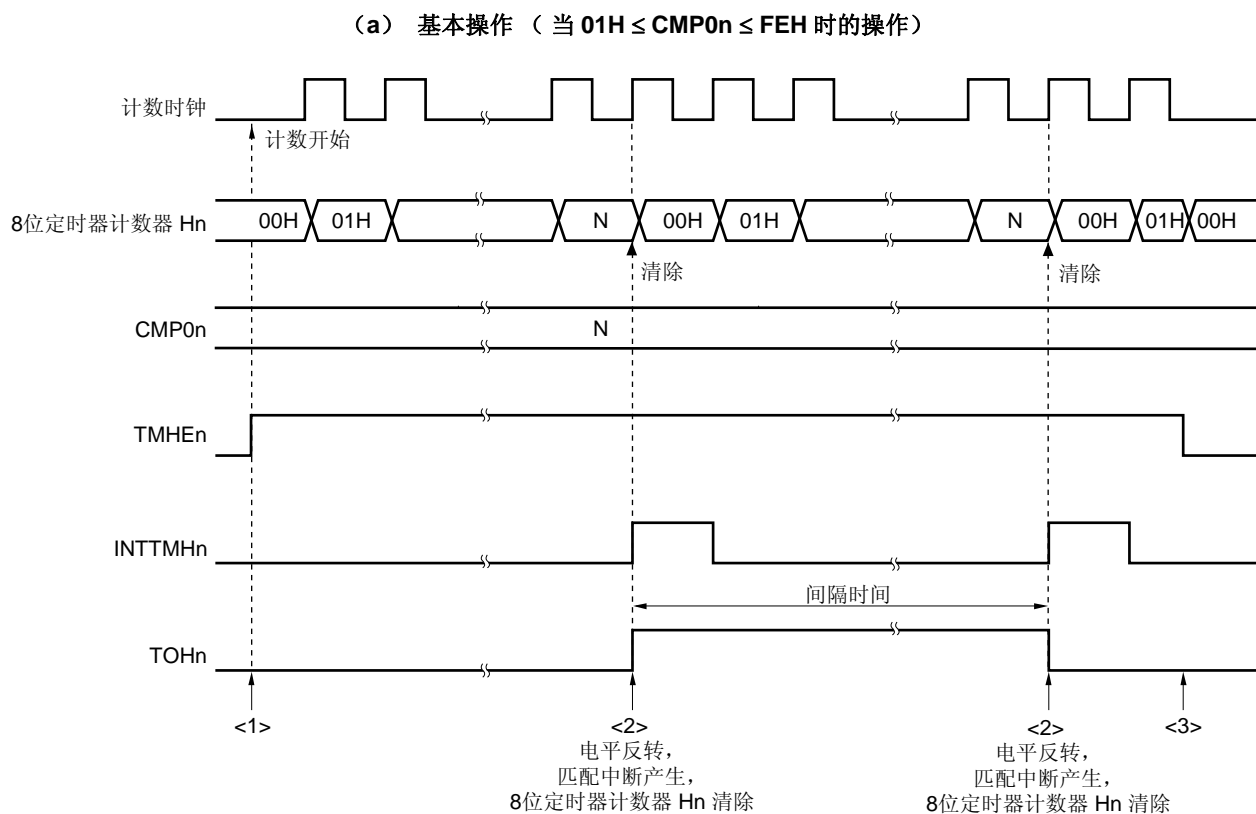
<2> 当 TMHEn = 1 时, 计数开始。

<3> 当 8 位定时器计数器 H_n 与 CMP0n 寄存器的值匹配时, 将产生 INTTMHn 信号, 并将 8 位定时器计数器 H_n 清零 (00H)。

<4> 随后, INTTMHn 信号会以相同的间隔时间重复产生。要停止计数操作, 则将 TMHEn 清零 (0)。

- 备注
1. 关于输出引脚设置的详细信息, 请参见 8.3 (3) 端口模式寄存器 0 (PM0) 和 (4) 端口输出模式寄存器 (POM0)。
 2. 如需了解允许 INTTMHn 信号中断, 请参见第十一章 中断功能。
 3. $n = 0, 1$

图 8-11. 间隔定时器 / 方波输出操作时序 (1/2)

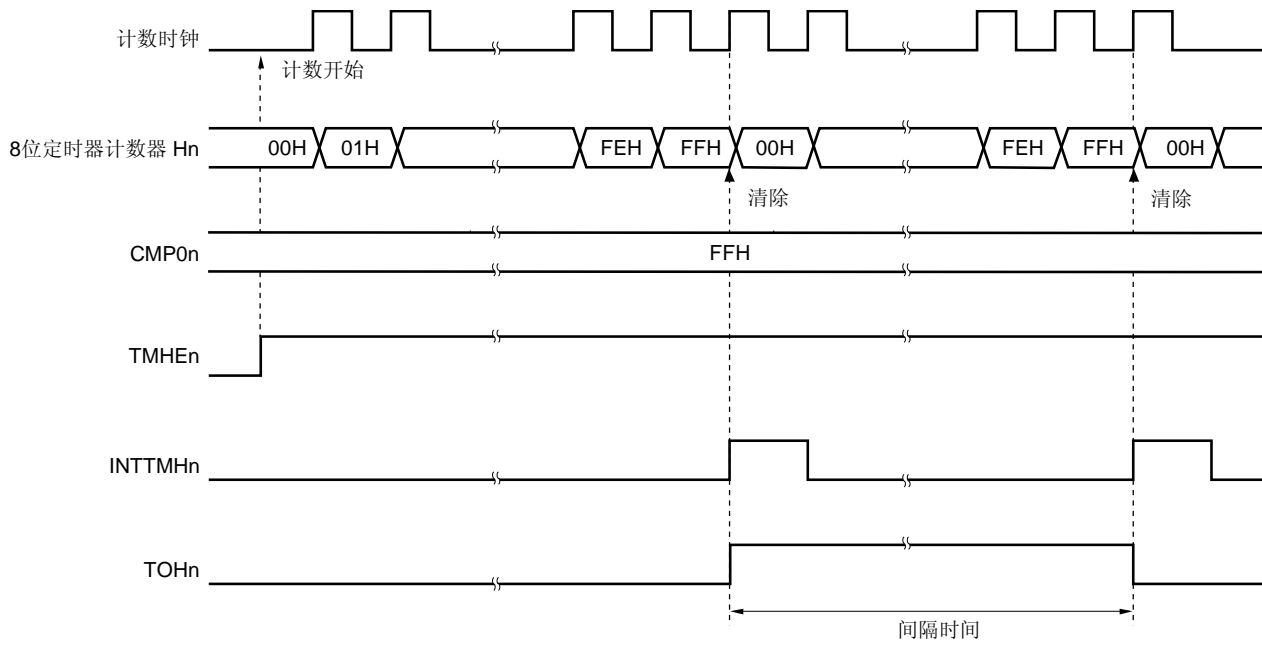


- <1> 通过将 $TMHEn$ 位置 1 可以允许计数操作。计数操作被允许后, 在不到 1 个时钟周期的时间内将会启动计数操作。
- <2> 当 8 位定时器计数器 H_n 与 $CMP0n$ 寄存器的值相等时, 该定时器计数器的值将会被清除, 且 $TOHn$ 的输出电平将会被反转。此外, 在计数时钟的上升沿处将会输出 $INTTMHn$ 信号。
- <3> 如果在定时器 H 操作期间将 $TMHEn$ 位清零, 那么 $INTTMHn$ 信号和 $TOHn$ 输出将会被设置为默认电平。如果在将 $TMHEn$ 位清零 (0) 前它们已经处于默认电平, 则电平保持不变。

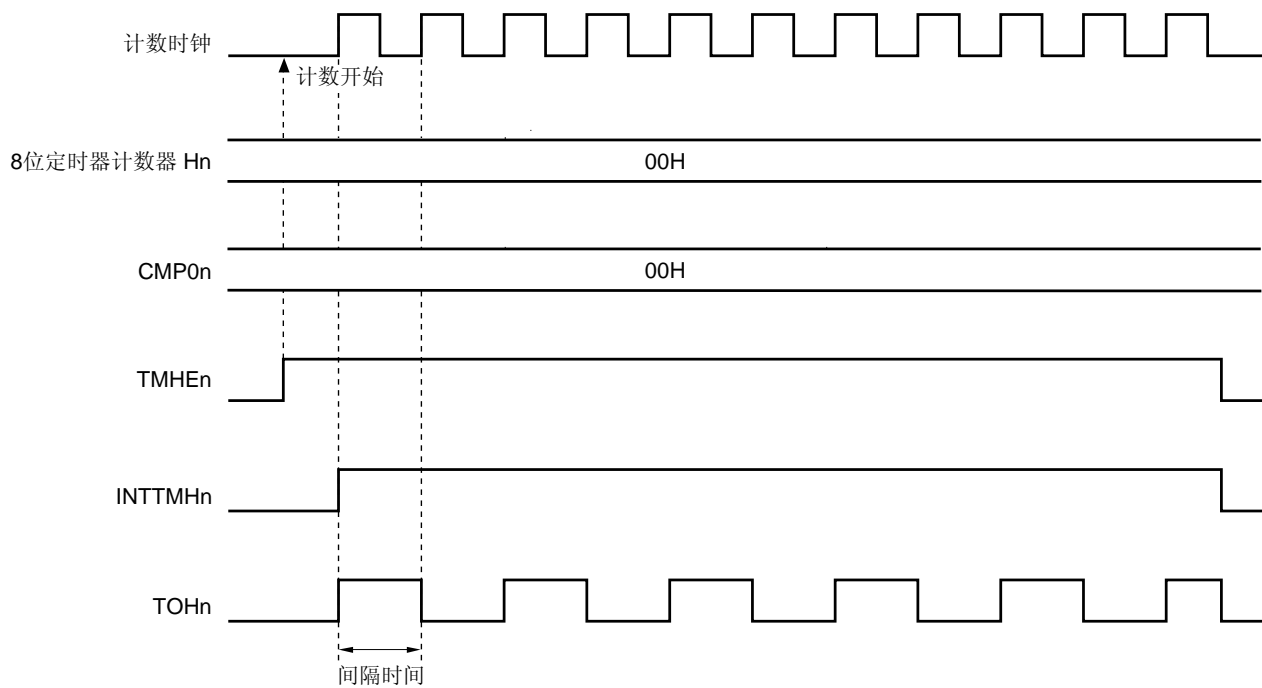
备注 $n = 0, 1$
 $01H \leq N \leq FEH$

图 8-11. 间隔定时器 / 方波输出操作时序 (2/2)

(b) 当 $CMP0n = FFH$ 时的操作



(c) 当 $CMP0n = 00H$ 时的操作



备注 n = 0, 1

8.4.2 用作PWM输出操作

在 PWM 输出模式中，可以输出具有任意占空比和周期的脉冲。

8 位定时器比较寄存器 0n (CMP0n) 用于控制定时器输出 (TOHn) 的周期。禁止在定时器操作期间重写 CMP0n 寄存器。

8 位定时器比较寄存器 1n (CMP1n) 用于控制定时器输出 (TOHn) 的占空比。允许在定时器操作期间重写 CMP1n 寄存器。

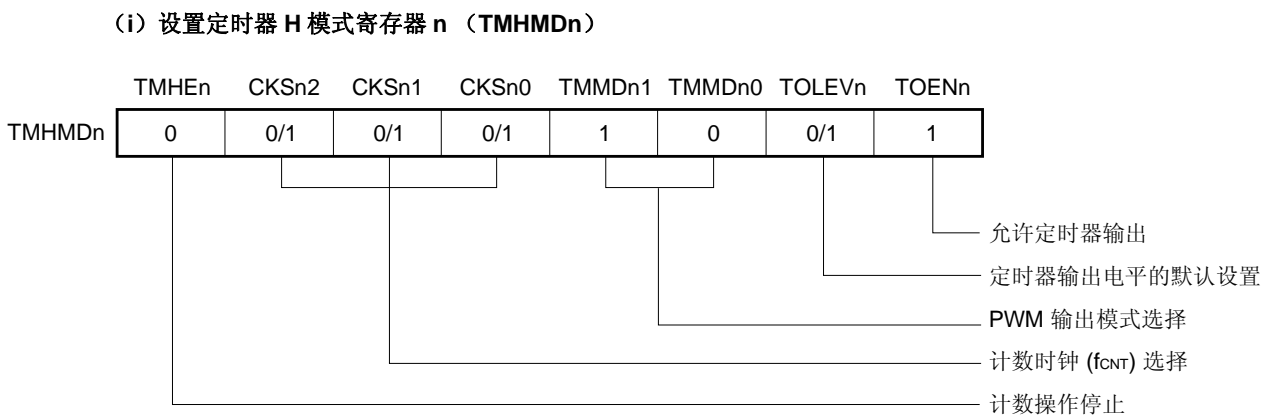
PWM 输出模式中的操作如下所示。

定时器计数启动后，在 8 位定时器 Hn 和 CMP0n 寄存器匹配时，PWM 输出 (TOHn 输出) 将会输出一个有效电平并将 8 位定时器 Hn 清零。当 8 位定时器计数器 Hn 和 CMP1n 寄存器匹配时，PWM 输出 (TOHn 输出) 则会输出一个无效电平。

设置

<1> 设置各个寄存器。

图 8-12. PWM 输出模式下的寄存器设置



(ii) 设置 CMP0n 寄存器

- 比较值 (N)：周期设置

(iii) 设置 CMP1n 寄存器

- 比较值 (M)：占空比设置

- 备注
1. $n = 0, 1$
 2. $00H \leq \text{CMP1n (M)} < \text{CMP0n (N)} \leq \text{FFH}$

<2> 当 TMHEn = 1 时，开始计数。

<3> CMP0n 寄存器是在允许计数器操作后第一个被比较的比较寄存器。当 8 位定时器计数器 Hn 与 CMP0n 寄存器的值匹配时，8 位定时器计数器 Hn 将先被清零，再生成一个中断请求信号 (INTTMHn)，并会输出一个有效电平。同时，要与 8 位定时器计数器 Hn 进行比较的比较寄存器将会从 CMP0n 寄存器变为 CMP1n 寄存器。

<4> 当 8 位定时器计数器 Hn 与 CMP1n 寄存器匹配时，将会输出一个无效电平，而用来与 8 位定时器计数器 Hn 比较的比较寄存器将会从 CMP1n 转变为 CMP0n 寄存器。此时，8 位定时器计数器 Hn 不会被清零，也不会生成 INTTMHn 信号。

<5> 通过重复执行过程<3> 和 <4>, 可以获取具有任意占空比的脉冲。

<6> 要停止计数操作, 则设置 $TMHE_n = 0$ 。

如果 $CMP0_n$ 寄存器的设定值为 N , $CMP1_n$ 寄存器的设定值为 M , 并且计数时钟频率为 f_{CNT} , 那么 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 = $(M + 1) / (N + 1)$

注意事项 1. 可以在定时器计数器操作时改变 $CMP1_n$ 的设置值。然而, 从改变 $CMP1_n$ 寄存器的值到这个值被传输给寄存器需要经历三个操作时钟 (通过 $TMHMD_n$ 寄存器的 $CKSn_2$ 到 $CKSn_0$ 位选择的时钟信号)。

2. 定时器计数操作停止后 ($TMHE_n=0$), 启动定时器计数操作时 ($TMHE_n=1$) 时必须对 $CMP1_n$ 寄存器进行设置 (即使对 $CMP1_n$ 寄存器设置了相同的值, 也务必再设置一次)。

3. 确认 $CMP1_n$ 寄存器的设置值 (M) 和 $CMP0_n$ 寄存器的设置值 (N) 在以下范围内。

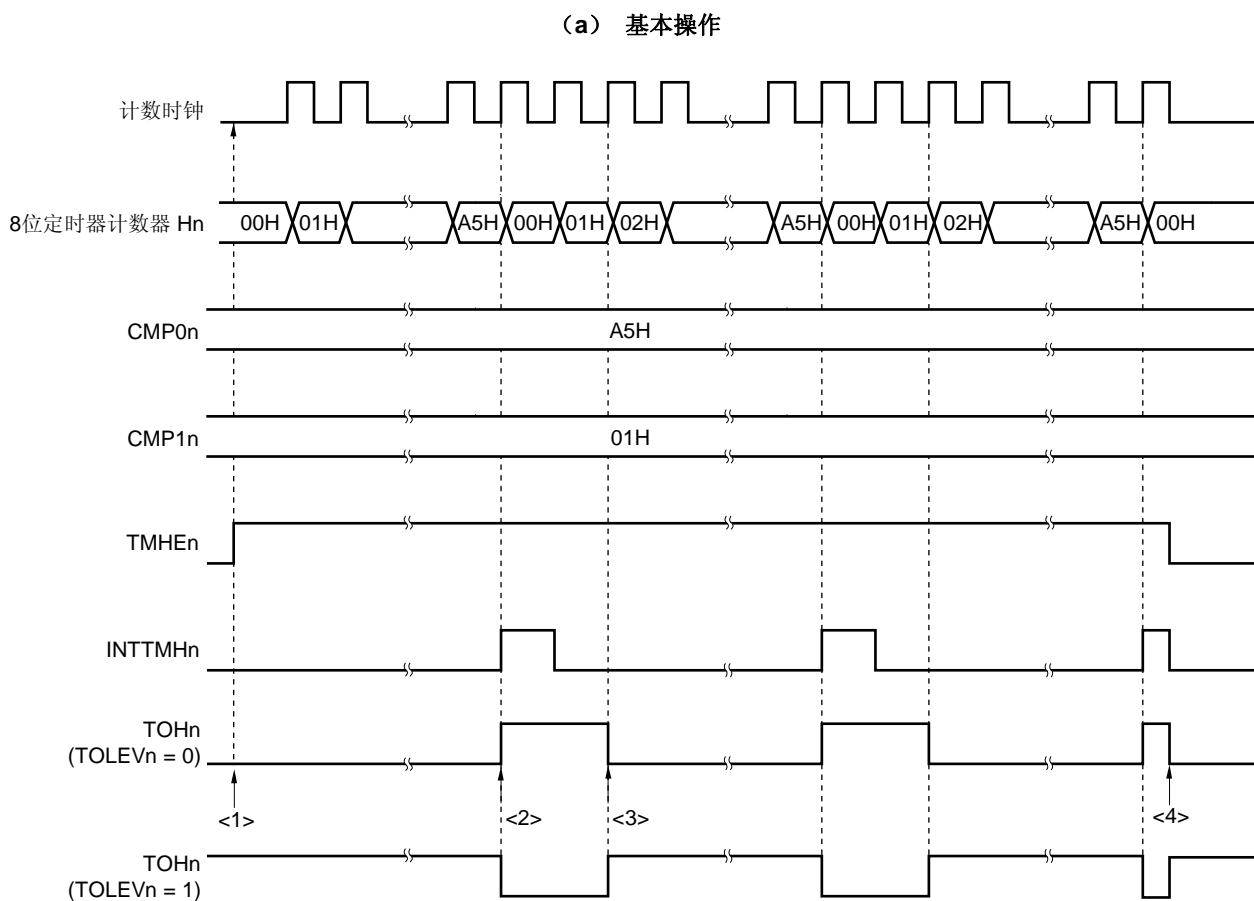
$$00H \leq CMP1_n (M) < CMP0_n (N) \leq FFH$$

备注 1. 关于输出引脚设置的详细信息, 请参见 8.3 (3) 端口模式寄存器 0 (PM0) 和 (4) 端口输出模式寄存器 (POM0)。

2. 如需了解如何允许 $INTTMH_n$ 信号中断, 请参见第十一章 中断功能。

3. $n = 0, 1$

图 8-13. PWM 输出模式下的操作时序 (1/4)

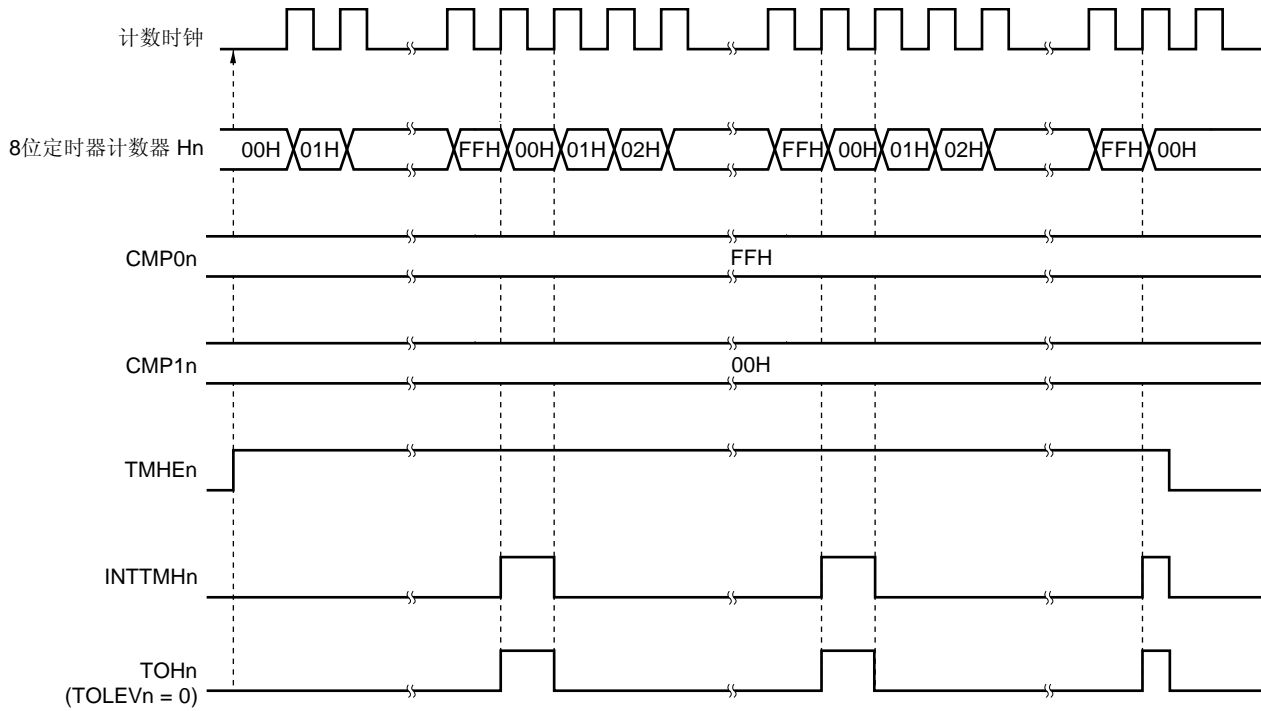


- <1> 通过将 TMHEn 位设为 1 来允许计数操作。通过屏蔽一个计数时钟来启动 8 位定时器计数器 Hn 进行计数。此时，PWM 输出一个无效电平。
- <2> 当 8 位定时器计数器 Hn 与 CMP0n 匹配时将输出一个有效电平。此时，8 位定时器计数器 Hn 的值不会被清零，也不会输出 INTTMHn 信号。
- <3> 当 8 位定时器计数器 Hn 与 CMP1n 寄存器的值匹配时将输出一个无效电平。此时，8 位计数器的值不会被清零，也不会生成 INTTMHn 信号。
- <4> 在定时器 Hn 操作期间若将 TMHEn 位清零，将可以设置 INTTMHn 信号为默认值并设置 PWM 输出为无效电平。

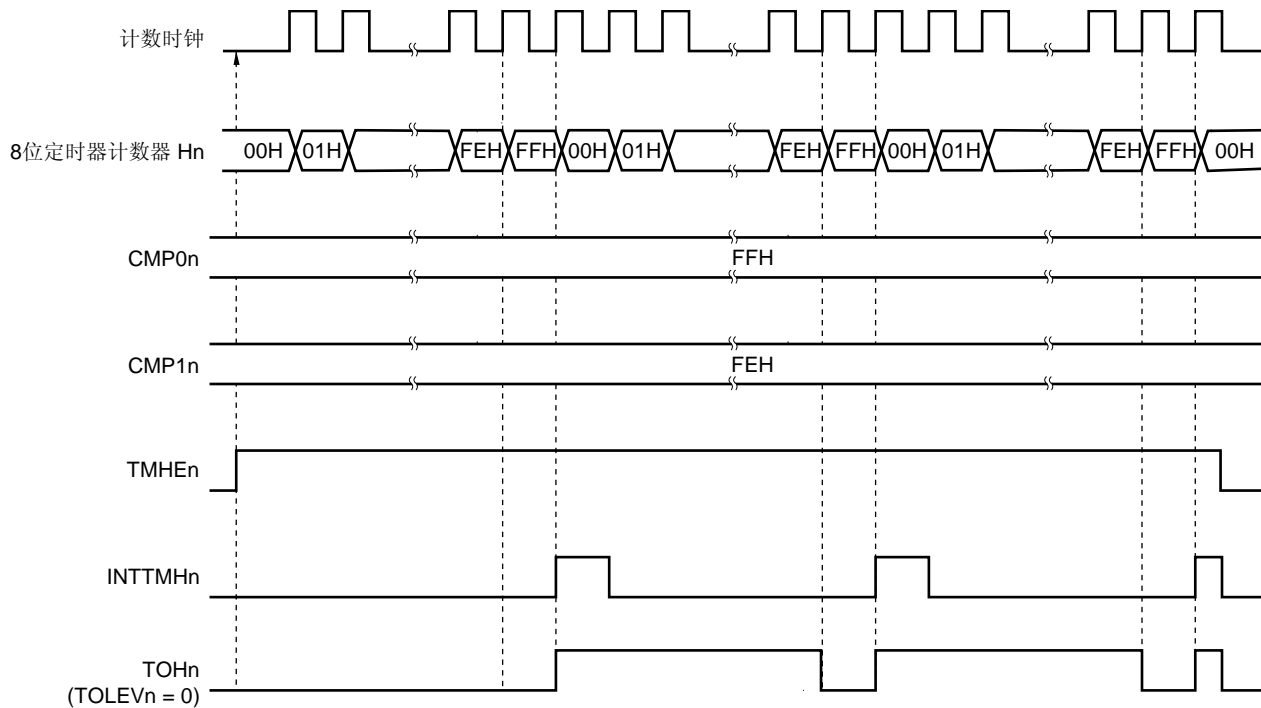
备注 n = 0, 1

图 8-13. PWM 输出模式下的操作时序 (2/4)

(b) 当 $CMP0n = FFH$, $CMP1n = 00H$ 时的操作



(c) 当 $CMP0n = FFH$, $CMP1n = FEH$ 时的操作



备注 $n = 0, 1$

图 8-13. PWM 输出模式下的操作时序 (3/4)

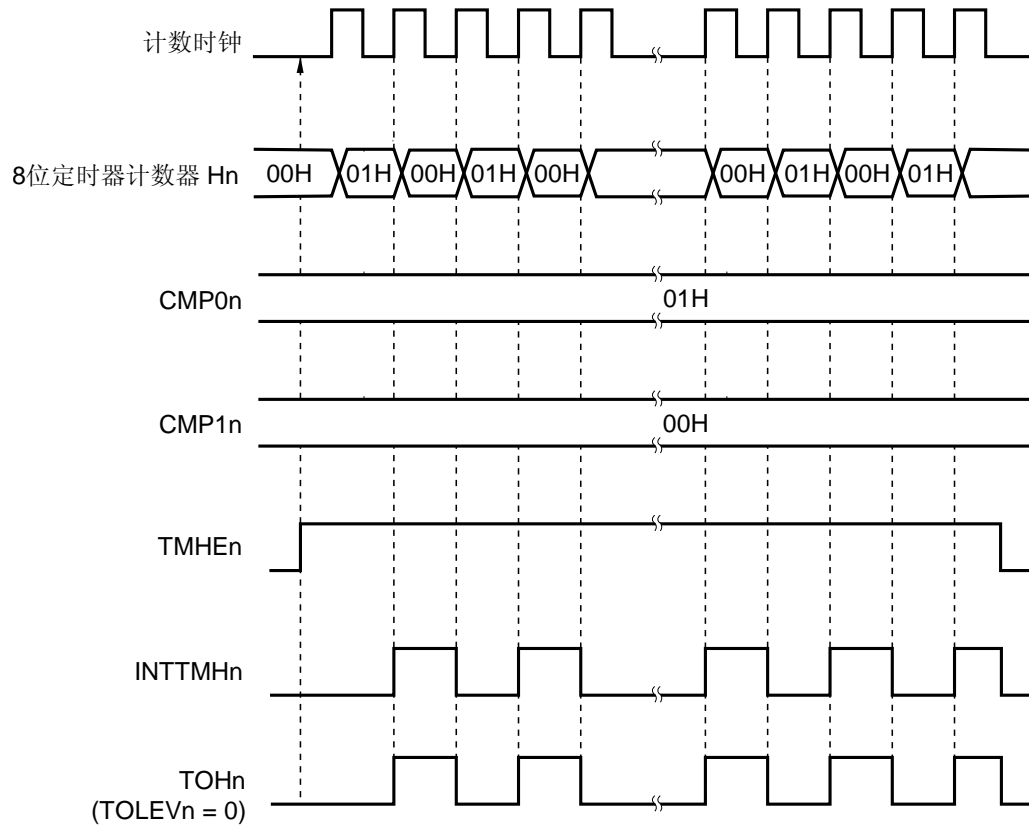
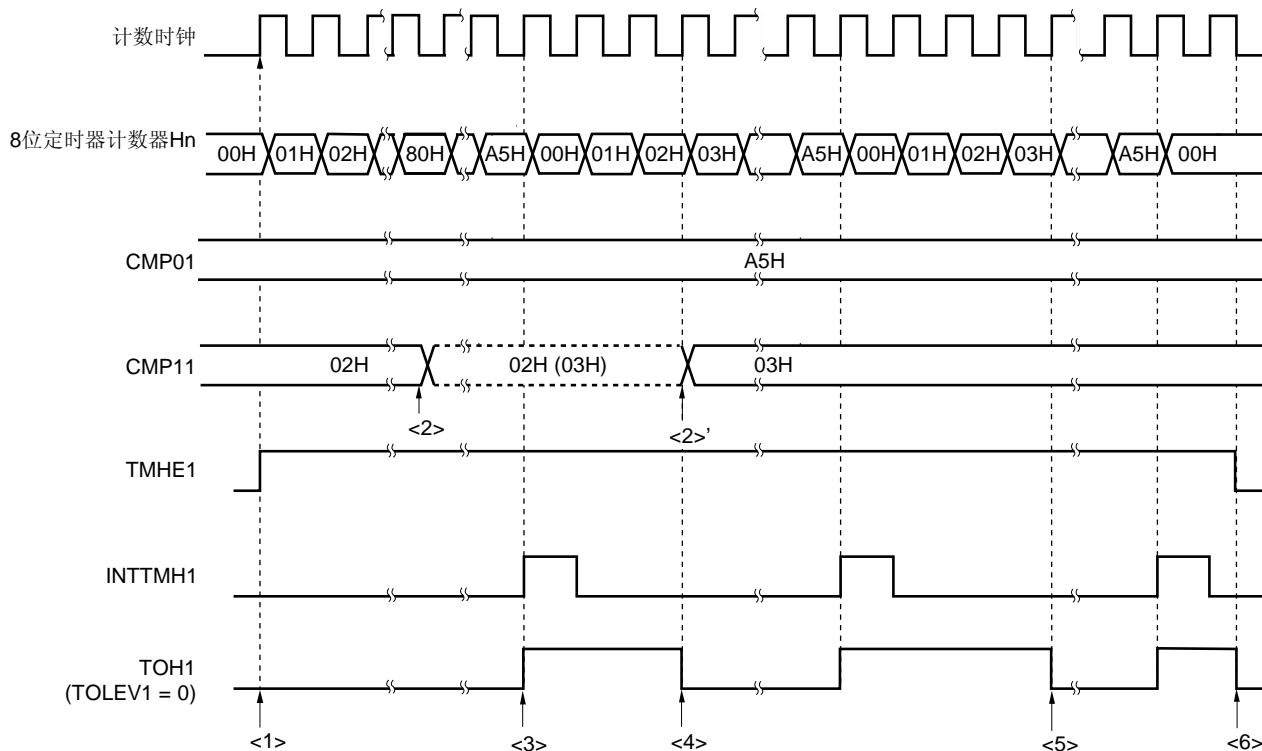
(d) 当 $CMP0n = 01H$, $CMP1n = 00H$ 时的操作备注 $n = 0, 1$

图 8-13. PWM 输出模式下的操作时序 (4/4)

(e) 通过改变 CMP1n 的操作 (CMP1n = 02H → 03H, CMP0n = A5H)



- <1> 通过将 TMHEn 位设为 1 来允许计数操作。通过屏蔽一个计数时钟来启动 8 位定时器计数器 Hn 进行计数。此时, PWM 输出一个无效电平。
- <2> 在定时器计数器操作期间可以修改 CMP1n 的值。该操作与计数时钟不同步。
- <3> 当 8 位定时器计数器 Hn 与 CMP0n 寄存器的值匹配时, 8 位定时器计数器 Hn 的值将被清除, 并会输出一个有效电平, 并且会输出一个 INTTMHn 信号。
- <4> 如果 CMP1n 寄存器的值被修改, 则修改后的值被锁存, 且不会被传送到寄存器。当 8 位定时器计数器 Hn 的值与修改前 CMP1n 寄存器的值匹配时, 则值将会被传送到 CMP1n 寄存器中, 且 CMP1n 寄存器的值将会被修改 (<2>')。
- 然而, 从 CMP1n 寄存器的值被修改到将值传送到寄存器, 至少需要 3 个计数时钟。如果在 3 个计数时钟内产生一个匹配信号, 则不能将修改后的值传送到寄存器。
- <5> 当 8 位定时器计数器 Hn 的值与修改后的 CMP1n 的值匹配时, 将会输出一个无效电平。此时, 8 位定时器计数器 Hn 不会被清零, 也不会生成 INTTMHn 信号。
- <6> 在定时器 Hn 操作期间若将 TMHEn 位清零, 将可以设置 INTTMHn 信号为默认值并设置 PWM 输出为无效电平。

备注 n = 0, 1

8.4.3 载波发生器操作（仅用于 8 位定时器H1）

在载波发生器模式中，8 位定时器 H1 用于产生红外遥控器的载波信号，8 位定时器 / 事件计数器 51 用于红外遥控信号的产生（计时）。

由 8 位定时器 H1 生成的载波时钟会在 8 位定时器 / 事件计数器 51 所设置的周期中输出。

在载波发生器模式中，8 位定时器 H1 载波脉冲由 8 位定时器 / 事件计数器 51 控制输出，且载波脉冲将会从 REM 输出中输出。

(1) 载波的发生

在载波发生器模式中，8 位定时器 H 比较寄存器 01（CMP01）生成了一个低电平宽度的载波脉冲波形，而 8 位定时器 H 比较寄存器 11（CMP11）则生成了一个高电平宽度的载波脉冲波形。

在 8 位定时器 H1 操作期间允许重写 CMP11 寄存器，但禁止重写 CMP01 寄存器。

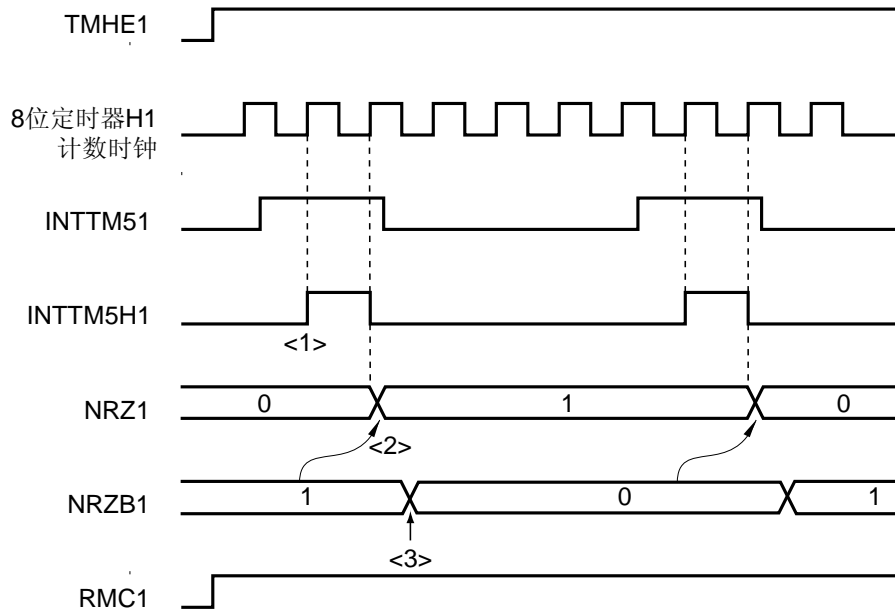
(2) 载波输出控制

载波输出由 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 和 RMC1 位来控制。输出之间的关系如下所示。

RMC1 位	NRZB1 位	输出
0	0	低电平输出
0	1	在 INTTM51 信号输入上升沿处的高电平输出
1	0	低电平输出
1	1	在 INTTM51 信号输入上升沿处的载波脉冲输出

为了在计数操作期间控制载波脉冲输出，TMCYC1 寄存器的 NRZ1 和 NRZB1 位有一个主位和一个从位配置。NRZ1 位只读，而 NRZB1 位可读可写。INTTM51 信号与 8 位定时器 H1 计数时钟同步，并且作为 INTTM5H1 信号输出。INTTM5H1 信号变为 NRZ1 位的数据传送信号，且 NRZB1 位的值被传送至 NRZ1 位。从 NRZB1 位传送到 NRZ1 位的时序如下所示。

图 8-14. 传送时序



<1> INTTM51 信号与 8 位定时器 H1 计数时钟同步，并且作为 INTTM5H1 信号输出。

<2> NRZB1 位的值在 INTTM5H1 信号上升沿的第 2 个时钟处被传送至 NRZ1 位。

<3> 在中断服务程序中将下一个值写入 NRZB1 位，该中断服务程序已由 INTTM5H1 中断启动或通过轮询中断请求标志检测到时序后启动。将下一次要计数的值写入 CR51 寄存器中。

注意事项 1. NRZB1 位被重写后至少到第二个时钟时才能被再次重写，否则，从 NRZB1 位到 NRZ1 位的传送将不能得到保证。

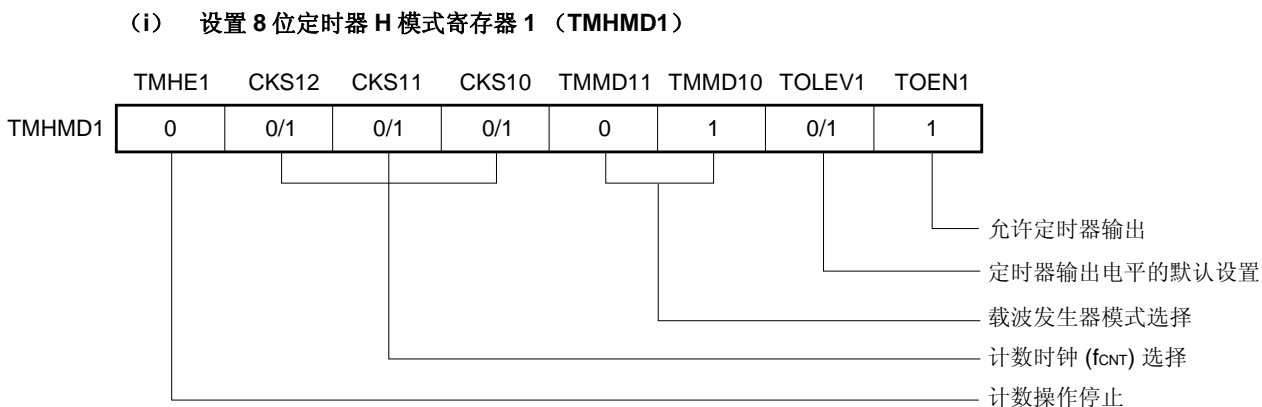
2. 在载波发生器模式中使用 8 位定时器 / 事件计数器 51 时，将会在<1>的时序中产生中断。在除载波发生器模式外的其他模式中使用 8 位定时器 / 事件计数器 51 时，中断产生的时序是不相同的。

备注 INTTM5H1 是内部信号但不是中断源。

设置

<1> 设置各个寄存器。

图 8-15. 在载波发生器模式下的寄存器设置



(ii) CMP01 寄存器设置

- 比较值

(iii) CMP11 寄存器设置

- 比较值

(iv) TMCYC1 寄存器设置

- RMC1 = 1 ...遥控输出允许位
- NRZB1 = 0/1 ...载波输出允许位

(v) TCL51 和 TMC51 的寄存器设置

- 参见 7.3 控制 8 位定时器 / 事件计数器 50 和 51 的寄存器。

<2> 当 TMHE1 = 1 时, 8 位定时器 H1 开始计数。

<3> 当 8 位定时器模式控制寄存器 51 (TMC51) 的 TCE51 被设为 1 时, 8 位定时器 / 事件计数器 51 开始计数。

<4> 允许计数操作后, 第一个用于比较的比较寄存器是 CMP01 寄存器。当 8 位定时器计数器 H1 的计数值与 CMP01 寄存器的值匹配时, 将产生 INTTMH1 信号, 并将 8 位定时器计数器 H1 清零。同时, 与 8 位定时器计数器 H1 比较的比较寄存器将从 CMP01 寄存器切换为 CMP11 寄存器。

<5> 当 8 位定时器计数器 H1 的计数值与 CMP11 寄存器的值匹配时, 将产生 INTTMH1 信号, 并将 8 位定时器计数器 H1 清零。同时, 与 8 位定时器计数器 H1 比较的比较寄存器将从 CMP11 寄存器切换为 CMP01 寄存器。

<6> 重复执行过程<4>和<5>, 将产生 1 个载波时钟。

<7> INTTM51 信号与 8 位定时器 H1 计数时钟同步, 并且作为 INTTM5H1 信号输出。INTTM5H1 信号变为 NRZB1 位的数据传送信号, 且 NRZB1 位的值被传送至 NRZ1 位。

<8> 在中断服务程序中将下一个值写入 NRZB1 位, 该中断服务程序已由 INTTM5H1 中断启动或通过轮询中断请求标志检测到时序后启动。将下一次要计数的值写入 CR51 寄存器中。

<9> 当 NRZ1 位是高电平时, 将通过 REM 输出来输出一个载波时钟。

<10> 通过执行上述步骤，可以获得任意形式的载波时钟。要停止计数操作，则将 TMHE1 清零（0）。

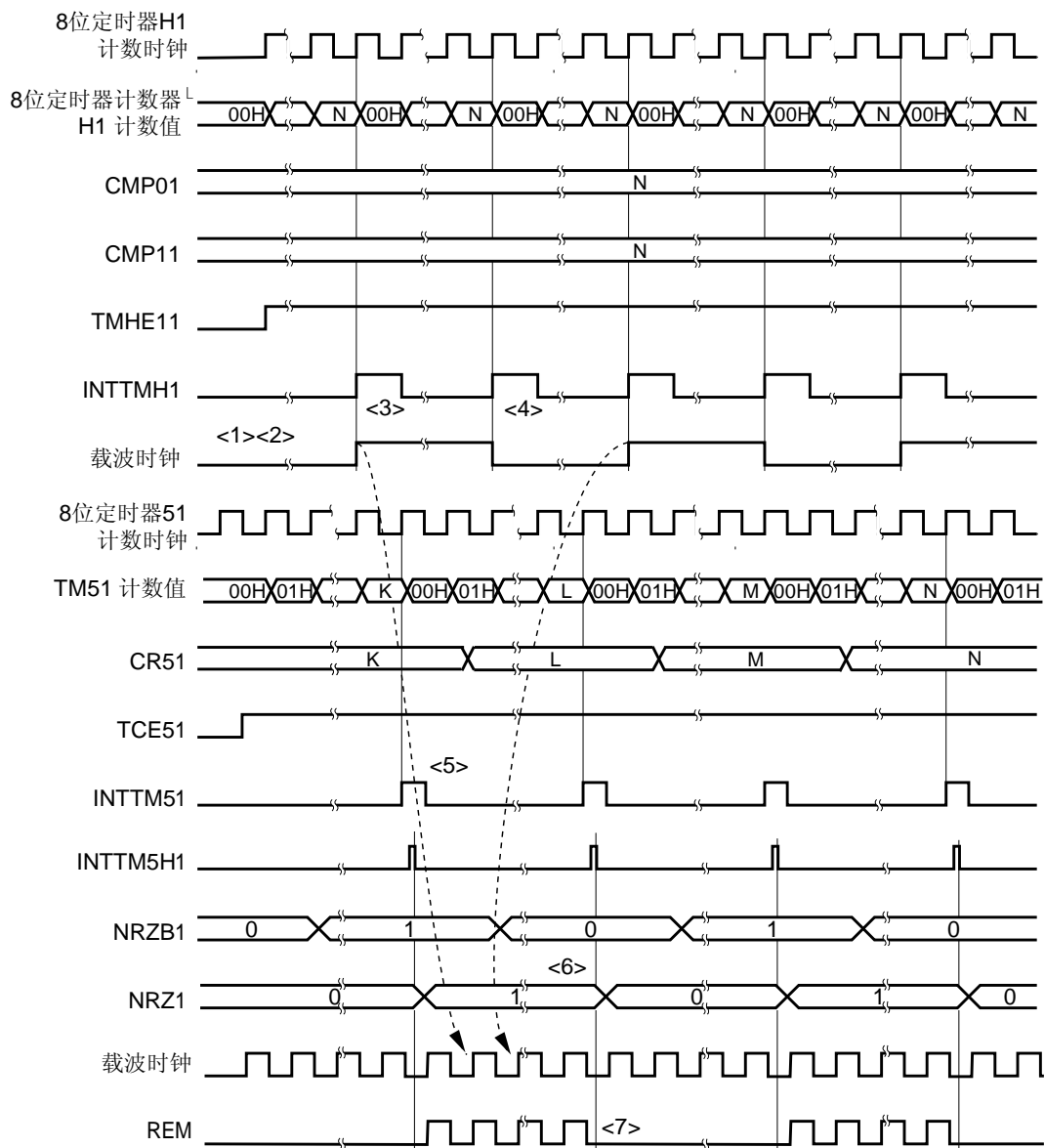
如果 CMP01 寄存器的设定值为 N，CMP11 寄存器的设定值为 M，并且计数时钟频率为 f_{CNT} ，那么载波时钟输出周期和占空比如下所示。

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

- 注意事项**
1. 定时器计数操作停止后（TMHE1=0），启动定时器计数操作时（TMHE1=1）时必须对 CMP11 寄存器进行设置（即使对 CMP11 寄存器设置了相同的值，也务必再设置一次）。
 2. 设置 TMH1 的计数时钟频率至少是 TM51 的计数时钟频率的 6 倍。
 3. CMP01 和 CMP11 寄存器的值必须设置在 01H 和 FFH 之间。
 4. 定时器计数器操作期间可以改变 CMP11 寄存器的值。然而，从改变 CMP11 的值到将该值传输到寄存器需要三个操作时钟（由 TMHMD1 寄存器的 CKS12 到 CKS10 位选择的时钟信号）。
 5. 在计时操作开始前必须设置 RMC1 位。

- 备注**
1. 关于输出引脚设置的详细信息，请参见 8.3（3）端口模式寄存器 0（PM0）和（4）端口输出模式寄存器（POM0）。
 2. 如需了解如何允许 INTTMH1 信号中断，请参见第十一章 中断功能。

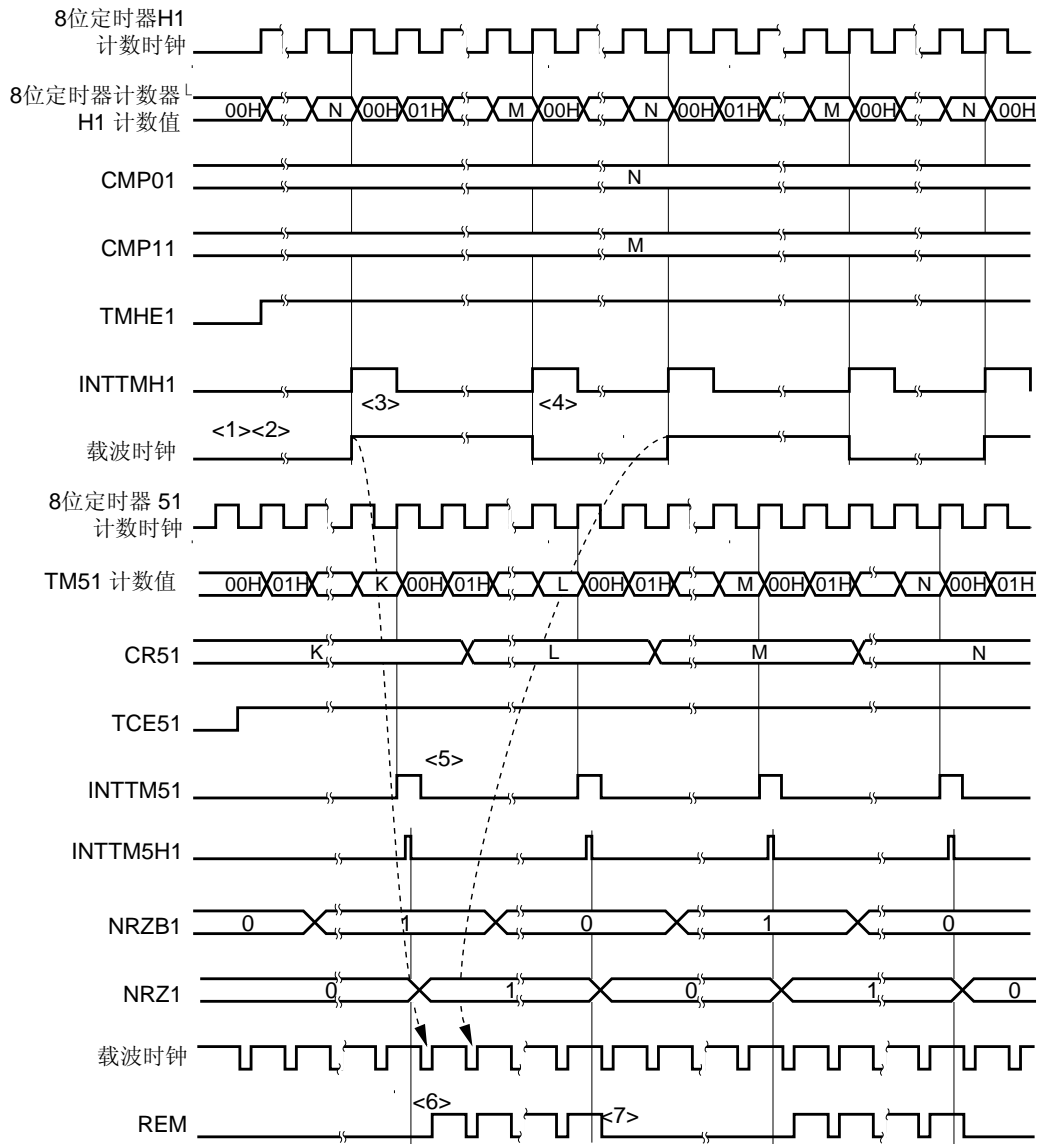
图 8-16. 载波发生器模式操作时序 (1/3)

(a) 当 $CMP01 = N$, $CMP11 = N$ 时的操作

- <1> 当 $TMHE1 = 0$ 且 $TCE51 = 0$ 时, 停止 8 位定时器计数器 H1 的操作。
- <2> 当 $TMHE1 = 1$ 时, 8 位定时器计数器 H1 开始计数操作。此时, 载波时钟保持默认电平状态。
- <3> 当 8 位定时器计数器 H1 的计数值与 $CMP01$ 寄存器的值匹配时, 将会生成第一个 $INTTMH1$ 信号并反转载波时钟信号, 同时与 8 位定时器计数器 H1 比较的比较寄存器将从 $CMP01$ 寄存器切换为 $CMP11$ 寄存器。8 位定时器计数器 H1 的值被清零 (00H)。
- <4> 当 8 位定时器计数器 H1 的计数值与 $CMP11$ 寄存器的值匹配时, 将会生成一个 $INTTMH1$ 信号并反转载波时钟信号, 同时与 8 位定时器计数器 H1 比较的比较寄存器将从 $CMP11$ 寄存器切换为 $CMP01$ 寄存器。8 位定时器计数器 H1 的值被清零 (00H)。重复执行过程<3>和<4>, 将产生 1 个载波时钟, 且占空比恒为 50%。
- <5> 当 $INTTM51$ 信号产生时, 它与 8 位定时器 H1 计数时钟同步, 并作为 $INTTM5H1$ 信号输出。
- <6> $INTTM5H1$ 信号变为 $NRZB1$ 位的数据传送信号, 且 $NRZB1$ 位的值被传送至 $NRZ1$ 位。
- <7> 设置 $NRZ1 = 0$ 时, REM 输出将为低电平。

备注 $INTTM5H1$ 是内部信号但不是中断源。

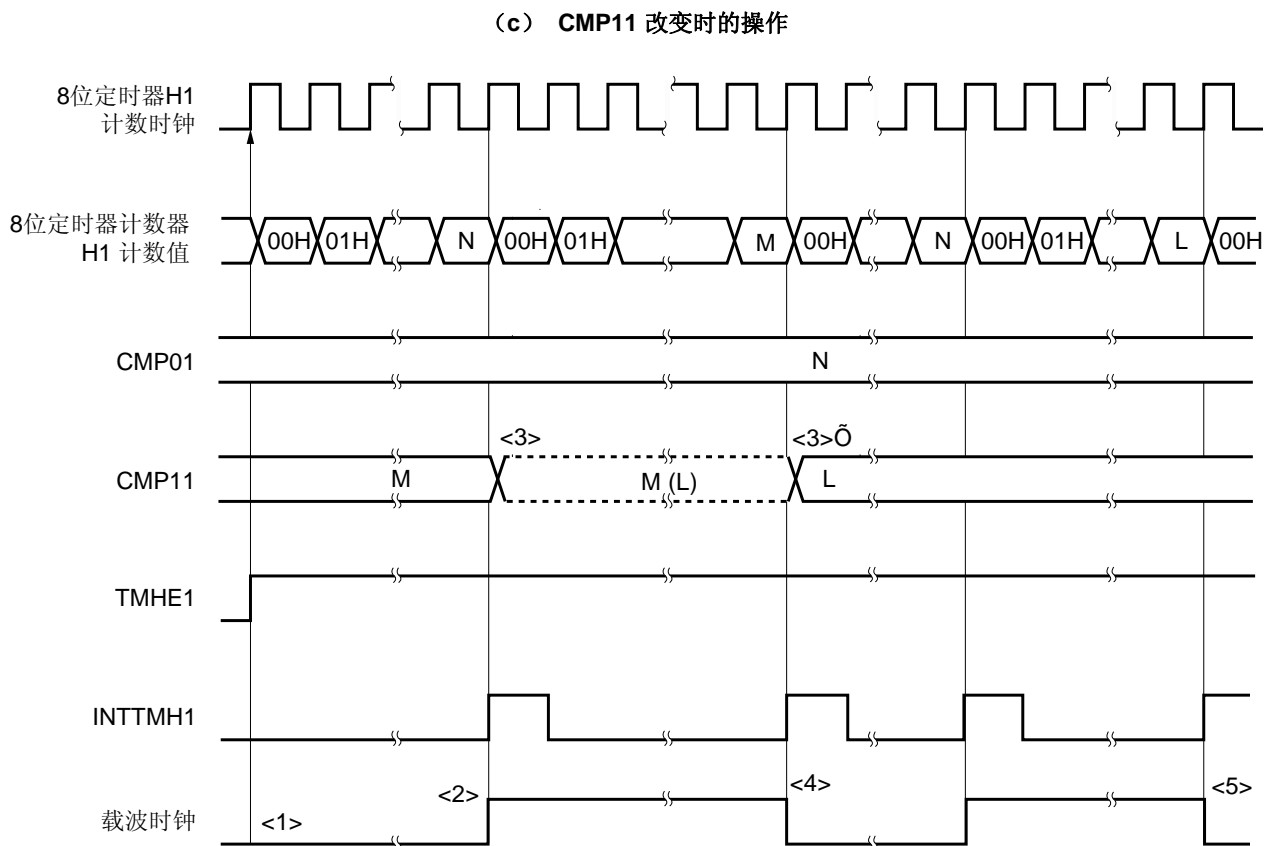
图 8-16. 载波发生器模式操作时序 (2/3)

(b) 当 $CMP01 = N$, $CMP11 = M$ 时的操作

- <1> 当 $TMHE1 = 0$ 且 $TCE51 = 0$ 时，停止 8 位定时器计数器 H1 的操作。
- <2> 当 $TMHE1 = 1$ 时，8 位定时器计数器 H1 开始计数操作。此时，载波时钟保持默认电平状态。
- <3> 当 8 位定时器计数器 H1 的计数值与 $CMP01$ 寄存器的值匹配时，将会生成第一个 $INTTMH1$ 信号并反转载波时钟信号，同时与 8 位定时器计数器 H1 比较的比较寄存器将从 $CMP01$ 寄存器切换为 $CMP11$ 寄存器。8 位定时器计数器 H1 的值被清零 (00H)。
- <4> 当 8 位定时器计数器 H1 的计数值与 $CMP11$ 寄存器的值匹配时，将会生成一个 $INTTMH1$ 信号并反转载波时钟信号，同时与 8 位定时器计数器 H1 比较的比较寄存器将从 $CMP11$ 寄存器切换为 $CMP01$ 寄存器。8 位定时器计数器 H1 的值被清零 (00H)。重复执行过程<3>和<4>，将产生 1 个载波时钟，但占空比不等于 50%。
- <5> 当 $INTTM51$ 信号产生时，它与 8 位定时器 H1 计数时钟同步，并作为 $INTTM5H1$ 信号输出。
- <6> 若 $NRZ1$ 被设为 1，那么将在载波时钟的第 1 个上升沿处输出一个载波信号。
- <7> 当 $NRZ1 = 0$ 时， REM 输出将保持高电平状态，且在载波时钟为高电平时（在过程<6>和<7>中，保证载波时钟波形的高电平宽度）不会转变为低电平。

备注 $INTTM5H1$ 是内部信号但不是中断源。

图 8-16. 载波发生器模式操作时序 (3/3)



- <1> 当设置 $TMHE1 = 1$ 时，8 位定时器 H1 开始计数操作。此时，载波时钟保持默认电平状态。
- <2> 当 8 位定时器计数器 H1 的计数值与 **CMP01** 寄存器的值匹配时，将输出 **INTTMH1** 信号、反转载波信号，并将定时器计数器清零 (00H)。同时，值与 8 位定时器计数器 H1 比较的比较寄存器将从 **CMP01** 寄存器切换为 **CMP11** 寄存器。
- <3> **CMP11** 寄存器与计数时钟不同步，在 8 位定时器 H1 操作期间可以修改 **CMP11** 寄存器的值。寄存器要改变到的新的值 (L) 将会被锁存。当 8 位定时器计数器 H1 的计数值与 **CMP11** 寄存器修改前的值 (M) 匹配时，**CMP11** 寄存器的值将会被改变 (<3>)。然而，从 **CMP11** 寄存器的值改变到将该值传送到寄存器将需要 3 个计数时钟或更长。即使在 3 个计数时钟内产生了匹配信号，新值也不会被传送到寄存器。
- <4> 当 8 位定时器计数器 H1 的计数值与 **CMP11** 寄存器修改前的值 (M) 匹配时，将会输出 **INTTMH1** 信号、反转载波信号，并将定时器计数器清零 (00H)。同时，值与 8 位定时器计数器 H1 比较的比较寄存器将从 **CMP11** 寄存器切换为 **CMP01** 寄存器。
- <5> 8 位定时器计数器 H1 的计数值与 **CMP11** 寄存器的值再次匹配时的时序通过修改后的值 (L) 来表示。

第九章 看门狗定时器

9.1 看门狗定时器的功能

看门狗定时器使用内部低速振荡时钟进行操作。

看门狗定时器用于检测不希望出现的程序循环。如果检测到一个程序循环，将产生一个内部复位信号。出现以下情况时检测程序循环。

- 如果看门狗定时器计数器溢出
- 如果对看门狗定时器允许寄存器 (WDTE) 执行 1 位操作指令。
- 如果将“ACH”以外的数据写入 WDTE
- 如果在窗口关闭期间将数据写入 WDTE
- 如果从没有通过 IMS 寄存器设置的区域获取指令 (在 CPU 挂起时检查是否有无效校验)
- 如果 CPU 通过读取 / 写入指令访问了没有通过 IMS 寄存器设置的区域 (FB00H 到 FFFFH 除外) (CPU 程序循环期间检测是否有非法访问)

当由看门狗定时器产生复位时，复位控制标志寄存器 (RESF) 的第 4 位 (WDTRF) 置 1。关于 RESF 的详细信息，请参见 **第十四章 复位功能**。

9.2 看门狗定时器的配置

看门狗定时器包含以下硬件。

表 9-1. 看门狗定时器的配置

项目	配置
控制寄存器	看门狗定时器允许寄存器 (WDTE)

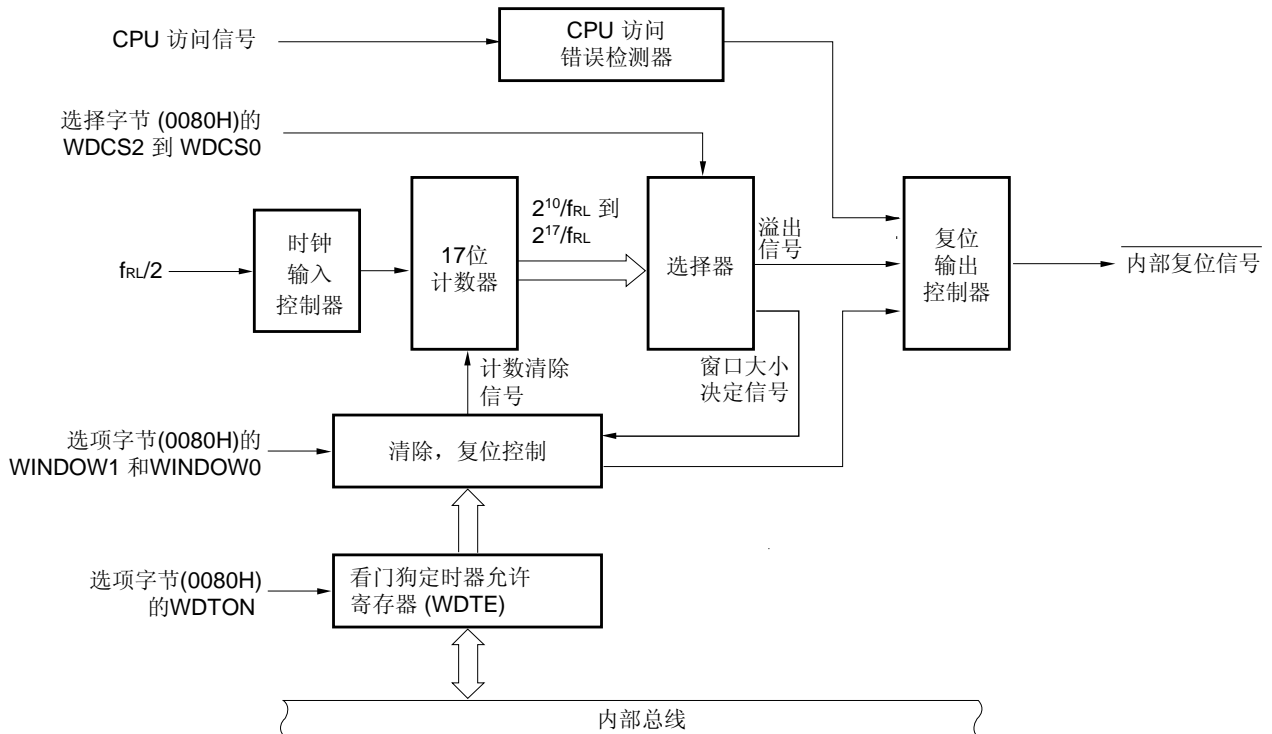
由选项字节设置如何控制计数器的操作、溢出时间和窗口打开周期。

表 9-2. 选项字节和看门狗定时器的设置

看门狗定时器的设置	选项字节 (0080H)
窗口打开周期	第 6 位和第 5 位 (WINDOW1, WINDOW0)
控制看门狗定时器的计数器操作	第 4 位 (WDTON)
看门狗定时器的溢出时间	第 3 位到第 1 位 (WDCS2 到 WDCS0)

备注 关于选项字节的详细信息，请参见第十七章 选项字节。

图 9-1. 看门狗定时器的框图



9.3 控制看门狗定时器的寄存器

看门狗定时器由看门狗定时器允许寄存器（WDTE）来控制。

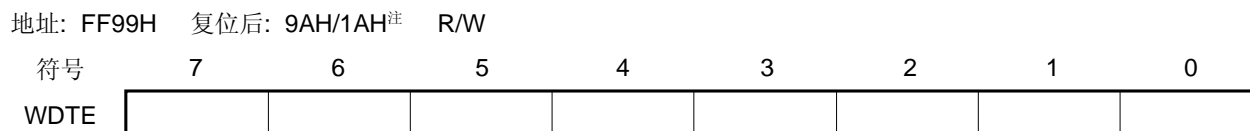
(1) 看门狗定时器允许寄存器（WDTE）

将 ACH 写入 WDTE 可以对看门狗定时器计数器清零并再次开始计数操作。

该寄存器可以通过 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器设置为 9AH 或 1AH^注。

图 9-2. 看门狗定时器允许寄存器（WDTE）的格式



注 WDTE 的复位后的值随选项字节（0080H）的 WDTON 位的设置值而变化。如果要对看门狗定时器进行操作，则设置 WDTON = 1。

WDTON 设置值	WDTE 复位值
0（禁止看门狗定时器计数的操作）	1AH
1（允许看门狗定时器计数的操作）	9AH

- 注意事项**
1. 如果一个非 ACH 的值被写入 WDTE，则将产生一个内部复位信号。然而，如果看门狗定时器的时钟源被停止，则当看门狗定时器的时钟源恢复操作时将会产生一个内部复位信号。
 2. 如果使用 1 位存储器操作指令对 WDTE 进行操作，将产生一个内部复位信号。然而，如果看门狗定时器的时钟源被停止，则当看门狗定时器的时钟源恢复操作时将会产生一个内部复位信号。
 3. 从 WDTE 读取的值为 9AH/1AH（与写入值（ACH）不同）。

9.4 看门狗定时器的操作

9.4.1 看门狗定时器的控制操作

1. 当使用看门狗定时器时，将由选项字节（0080H）来指定其操作。
 - 通过设置选项字节（0080H）的 4 位（WDTON）为 1 来允许看门狗定时器的计数操作（复位释放后计数器开始计数）（详细信息，请参见 第十七章）。

WDTON	看门狗定时器计数器的操作控制 / 非法访问检测
0	禁止计数器操作（复位后停止计数），禁止非法访问检测操作
1	允许计数器操作（复位后开始计数），允许非法访问检测操作

- 通过使用选项字节（0080H）的第 3 位到第 1 位（WDCS2 到 WDSC0）来设置溢出时间（详细信息，请参见 9.4.2 和 第十七章）。
 - 通过使用选项字节（0080H）的第 6 位和第 5 位（WINDOW1 和 WINDOW0）设置窗口打开周期（详细信息，请参见 9.4.3 和 第十七章）。
2. 复位释放后，看门狗定时器开始计数。
 3. 在看门狗定时器开始计数后且在达到选项字节设置的溢出时间之前，通过将“ACH”写入 WDTE，看门狗定时器将被清零并再次开始计数。
 4. 在这之后，在复位释放后的第二次或之后的窗口打开周期期间对 WDTE 进行写入。如果在窗口关闭期间对 WDTE 进行写入，则会产生内部复位信号。
 5. 如果在“ACH”没有写入 WDTE 的情况下到达溢出时间，则将产生一个内部复位信号。内部复位信号会在以下情况中产生。
 - 如果对看门狗定时器允许寄存器（WDTE）执行 1 位操作指令。
 - 如果将“ACH”以外的数据写入 WDTE
 - 如果从没有通过 IMS 寄存器设置的区域获取指令（在 CPU 程序循环期间检查是否有无效校验）
 - 如果 CPU 通过读取 / 写入指令访问了没有通过 IMS 寄存器设置的区域（FB00H 到 FFFFH 除外）（CPU 程序循环期间检测是否有非法访问）

- 注意事项**
1. 复位释放后对 WDTE 进行的第一次写操作将会清零看门狗定时器，如果不管写操作时序是什么，这一操作是在到达溢出时间之前进行的，那和看门狗定时器将再次开始计数。
 2. 如果通过将“ACH”写入 WDTE 来对看门狗定时器清零，则实际的溢出时间可能与由选项字节设置的溢出时间不同，最高误差可达 $2/f_{RL}$ 秒。
 3. 在计数值溢出（FFFFH）之前，可将看门狗定时器立即清零。

注意事项 4. 在 HALT 和 STOP 模式下看门狗定时器的操作依据选项字节第 0 位 (LSROSC) 的设置值的不同而不同, 如下所示。

	LSROSC = 0 (内部低速振荡器可以通过软件来停止)	LSROSC = 1 (内部低速振荡器不能停止)
HALT 模式中	看门狗定时器操作停止	看门狗定时器操作继续
STOP 模式中		

如果 LSROSC = 0, 那么 HALT 或 STOP 模式释放后看门狗定时器将会恢复计数。此时, 计数器不会被清零, 但会从计数器停止时的值开始计数。

如果在 LSROSC = 0 时通过设置 LSRSTOP (内部振荡模式寄存器 (RCM) 的第 1 位=1) 停止内部低速振荡器的振荡, 则看门狗定时器将停止操作。此时, 计数器不会被清零。

<R> 5. 在 flash 存储器的自编程过程中看门狗定时器会继续其操作操作。在处理过程中中断响应时间将会被延迟。设置溢出时间和窗口大小时应将延迟情况考虑在内。

9.4.2 设置看门狗定时器的溢出时间

通过使用选项字节 (0080H) 的第 3 位到第 1 位 (WDSC2 到 WDSC0) 来设置看门狗定时器的溢出时间。

如果出现溢出, 则将产生内部复位信号。在到达溢出时间之前的窗口打开周期内通过将 “ACH” 写入 WDTE, 可对当前计数值清零, 且看门狗定时器再次开始计数。

溢出时间设置如下所示。

表 9-3. 看门狗定时器溢出时间的设置 (当 2.1 V ≤ V_{DD} ≤ 3.6 V 时)

<R>

WDSC2	WDSC1	WDSC0	看门狗定时器的溢出时间
0	0	0	$2^{10}/f_{RL}$ (3.71 ms)
0	0	1	$2^{11}/f_{RL}$ (7.42 ms)
0	1	0	$2^{12}/f_{RL}$ (14.84 ms)
0	1	1	$2^{13}/f_{RL}$ (29.68 ms)
1	0	0	$2^{14}/f_{RL}$ (59.36 ms)
1	0	1	$2^{15}/f_{RL}$ (118.72 ms)
1	1	0	$2^{16}/f_{RL}$ (237.45 ms)
1	1	1	$2^{17}/f_{RL}$ (474.90 ms)

注意事项 1. 禁止出现 WDSC2 = WDSC1 = WDSC0 = 0 且 WINDOW1 = WINDOW0 = 0 的情况。

2. 在 flash 存储器的自编程过程中看门狗定时器会继续其操作操作。在处理过程中中断响应时间将会被延迟。设置溢出时间和窗口大小时应将延迟情况考虑在内。

备注 1. f_{RL} : 内部低速振荡时钟频率
2. () : $f_{RL} = 276 \text{ kHz (MAX.)}$

<R>

表 9-4. 看门狗定时器溢出时间的设置（当 $1.8\text{ V} \leq V_{DD} < 2.1\text{ V}$ 时）

WDCS2	WDCS1	WDCS0	看门狗定时器的溢出时间
0	0	0	$2^{10}/f_{RL}$ (3.41ms)
0	0	1	$2^{11}/f_{RL}$ (6.83 ms)
0	1	0	$2^{12}/f_{RL}$ (13.65 ms)
0	1	1	$2^{13}/f_{RL}$ (27.31 ms)
1	0	0	$2^{14}/f_{RL}$ (54.61 ms)
1	0	1	$2^{15}/f_{RL}$ (109.23 ms)
1	1	0	$2^{16}/f_{RL}$ (218.45 ms)
1	1	1	$2^{17}/f_{RL}$ (436.91 ms)

注意事项 1. 禁止出现 $WDCS2 = WDCS1 = WDCS0 = 0$ 且 $WINDOW1 = WINDOW0 = 0$ 的情况。

2. 在 flash 存储器的自编程过程中看门狗定时器会继续其操作操作。在处理过程中中断响应时间将会被延迟。设置溢出时间和窗口大小时应将延迟情况考虑在内。

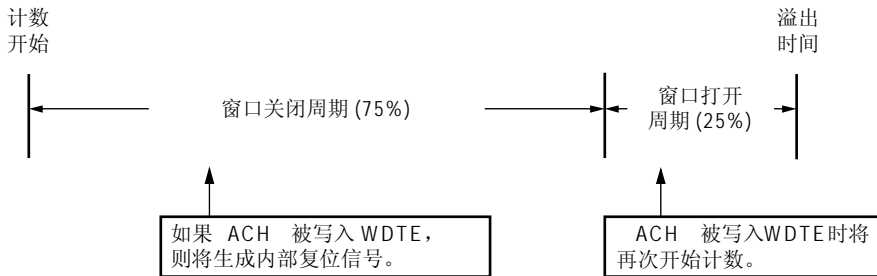
备注 1. f_{RL} : 内部低速振荡时钟频率
2. () : $f_{RL} = 300\text{ kHz (MAX.)}$

9.4.3 设置看门狗定时器窗口打开周期

通过使用选项字节 (0080H) 的第 6 位和第 5 位 (WINDOW1, WINDOW0) 来设置看门狗定时器的窗口打开周期。关于窗口的概述如下所示。

- 如果在窗口打开期间将“ACH”写入 WDTE，则看门狗定时器将会被清零并再次开始计数。
- 在窗口关闭期间即使将“ACH”写入 WDTE，也会检测到异常情况并产生内部复位信号。

例：如果窗口打开周期为 25%



注意事项 复位释放后对 WDTE 进行的第一次写操作将会清零看门狗定时器，如果不管写操作时序是什么，这一操作是在到达溢出时间之前进行的，那和看门狗定时器将再次开始计数。

要进行设置的窗口打开周期如下所示。

表 9-5. 设置看门狗定时器窗口打开周期

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

<R>

注意事项 1. 禁止出现 $WDCS2 = WDCS1 = WDCS0 = 0$ 且 $WINDOW1 = WINDOW0 = 0$ 的情况。
2. 在 flash 存储器的自编程过程中看门狗定时器会继续其操作操作。在处理过程中中断响应时间将会被延迟。设置溢出时间和窗口大小时应将延迟情况考虑在内。

备注 如果将溢出时间设置为 $2^{10}/f_{RL}$ ，则窗口关闭时间和打开时间如下所示。

<R>

(当 $2.1\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ 时)

	窗口打开周期的设置			
	25%	50%	75%	100%
窗口关闭时间	0 到 3.56 ms	0 到 2.37 ms	0 到 1.19 ms	无
窗口打开时间	3.56 到 3.71 ms	2.37 到 3.71 ms	1.19 到 3.71 ms	0 到 3.71 ms

<当窗口打开周期为 25%>

- 溢出时间：
 $2^{10}/f_{RL}$ (MAX.) = $2^{10}/276\text{ kHz}$ (MAX.) = 3.71 ms
- 窗口关闭时间：
 0 到 $2^{10}/f_{RL}$ (MIN.) $\times (1 - 0.25) = 0$ 到 $2^{10}/216\text{ kHz}$ (MIN.) $\times 0.75 = 0$ 到 3.56 ms
- 窗口打开时间：
 $2^{10}/f_{RL}$ (MIN.) $\times (1 - 0.25)$ 到 $2^{10}/f_{RL}$ (MAX.) = $2^{10}/216\text{ kHz}$ (MIN.) $\times 0.75$ 到 $2^{10}/276\text{ kHz}$ (MAX.) = 3.56 到 3.71 ms

<R>

(当 $1.8\text{ V} \leq V_{DD} < 2.1\text{ V}$ 时)

	窗口打开周期的设置			
	25%	50%	75%	100%
窗口关闭时间	禁止设置	0 到 2.84 ms	0 到 1.42 ms	无
窗口打开时间		2.84 到 3.41 ms	1.42 到 3.41 ms	0 到 3.41 ms

<当窗口打开周期为 25%>

- 溢出时间：
 $2^{10}/f_{RL}$ (MAX.) = $2^{10}/300\text{ kHz}$ (MAX.) = 3.41 ms
- 窗口关闭时间：
 0 到 $2^{10}/f_{RL}$ (MIN.) $\times (1 - 0.5) = 0$ 到 $2^{10}/180\text{ kHz}$ (MIN.) $\times 0.5 = 0$ 到 2.84 ms
- 窗口打开时间：
 $2^{10}/f_{RL}$ (MIN.) $\times (1 - 0.5)$ 到 $2^{10}/f_{RL}$ (MAX.) = $2^{10}/180\text{ kHz}$ (MIN.) $\times 0.5$ 到 $2^{10}/300\text{ kHz}$ (MAX.) = 2.84 到 3.41 ms

10.1 串行接口UART6 的功能

串行接口 UART6 有以下两种模式。

(1) 操作停止模式

在不进行串行通信时可采用此模式，以降低功耗。

详细信息，请参见 **10.4.1 操作停止模式**。

(2) 异步串行接口 (UART) 模式

该模式的功能概括如下。

详细信息，请参见 **10.4.2 异步串行接口 (UART) 模式** 和 **10.4.3 专用波特率发生器**。

- 最大传输速率：312.5 kbps
- 双引脚配置
TxD6: 发送数据输出引脚
RxD6: 接收数据输入引脚
- 可选择通信数据的数据宽度为 7 或 8 位。
- 专用内置 8 位波特率发生器允许设置任意大小的波特率。
- 可分别执行发送和接收操作（全双工操作）。
- 可选 MSB 优先或 LSB 优先通信。
- 反向发送操作。

- 注意事项**
1. **TxD6** 输出反转功能仅在发送端进行而不在接收端进行。为了能够使用该功能，接收端应准备好接收反转数据。
 2. 如果提供给串行接口 **UART6** 的时钟未被停止（例如，处于 **HALT** 模式下），则继续正常操作。如果提供给串行接口 **UART6** 的时钟已被停止（例如，处于 **STOP** 模式下），则寄存器停止操作，并且在时钟停止之前及时保存数值。**TXD6** 引脚也在时钟停止之前立即保存数值并将保存的数值输出。然而，时钟恢复后，操作将不能被保证。因此，必须复位电路，使 **POWER6 = 0**，**RXE6 = 0** 以及 **TXE6 = 0**。
 3. 设置 **POWER6 = 1** 然后设置 **TXE6 = 1**（发送）或 **RXE6 = 1**（接收），以启动通信。
 4. 通过由 **CKSR6** 设置的基本时钟（**f_{CLK6}**）使 **TXE6** 与 **RXE6** 同步。若要再次允许发送或接收，必须在 **TXE6** 或 **RXE6** 清零（0）后至少两个基本时钟后将 **TXE6** 或 **RXE6** 设置为 1。如果在两个基本时钟内设置 **TXE6** 或 **RXE6**，则发送电路或接收电路可能不被初始化。
 5. 在设置 **TXE6 = 1** 后至少经历一个基本时钟（**f_{CLK6}**）再为 **TXB6** 设置发送数据。
 6. 如果连续发送数据，从停止位到下一个起始位的通信时序将延长两个宏操作时钟。然而，这不会影响到通信结果，因为接收端会在检测到起始位时会对时序进行初始化。

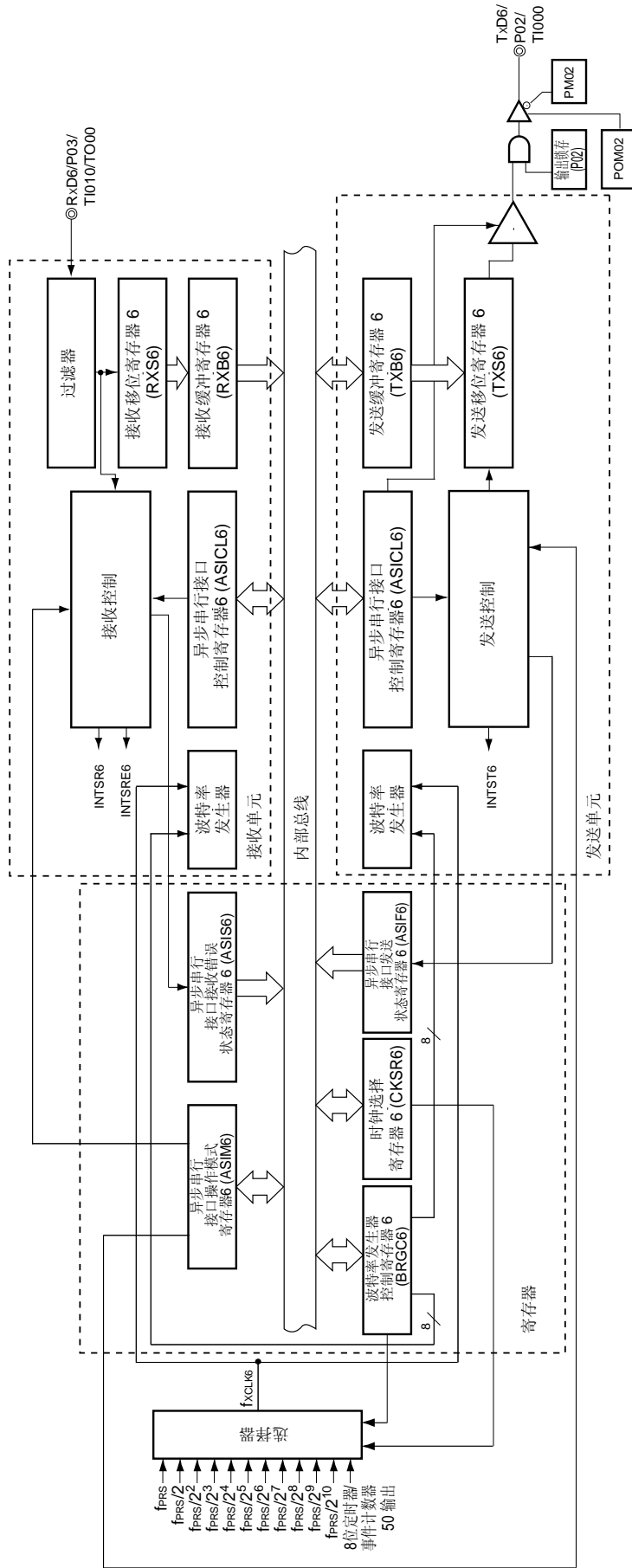
10.2 串行接口UART6 的配置

串行接口 UART6 包括以下硬件。

表 10-1. 串行接口 UART6 的配置

项目	配置
寄存器	接收缓冲寄存器 6 (RXB6) 接收移位寄存器 6 (RXS6) 发送缓冲寄存器 6 (TXB6) 发送移位寄存器 6 (TXS6)
控制寄存器	异步串行接口操作模式寄存器 6 (ASIM6) 异步串行接口接收错误状态寄存器 6 (ASIS6) 异步串行接口发送状态寄存器 6 (ASIF6) 时钟选择寄存器 6 (CKSR6) 波特率发生器控制寄存器 6 (BRGC6) 异步串行接口控制寄存器 6 (ASICL6) 端口模式寄存器 0 (PM0) 端口寄存器 0 (P0) 端口输出模式寄存器 0 (POM0)

图 10-1. 串行接口 UART6 框图



(1) 接收缓冲寄存器 6 (RXB6)

该 8 位寄存器用于存储由接收移位寄存器 (RXS6) 转换的并行数据。

每次接收到 1 字节的数据时, 新的接收数据就会从 RXS6 传送到该寄存器中。如果数据宽度被设定为 7 位, 则按如下方式发送数据。

- 在 LSB 优先接收模式下, 将接收到的数据传送到 RXB6 的 0 到 6 位, 而 MSB 位恒为 0。
- 在 MSB 优先接收模式下, 将接收到的数据传送到 RXB6 的 1 到 7 位, 而 LSB 位恒为 0。

如果出现溢出错误 (OVE6), 则不把接收数据传送到 RXB6 中。

RXB6 可以通过 8 位存储器操作指令来进行读取。不能将数据写入该寄存器中。

复位信号生成将该寄存器设置为 FFH。

(2) 接收移位寄存器 6 (RXS6)

该寄存器将输入到 RxD6 引脚的串行数据转换成并行数据。

RXS6 不能通过程序直接进行操作。

(3) 发送缓冲寄存器 6 (TXB6)

该缓冲寄存器用于设置发送数据。当数据写入 TXB6 时开始发送。该寄存器可通过 8 位存储器操作指令进行读写。

复位信号生成将该寄存器设置为 FFH。

- 注意事项**
1. 当异步串行接口发送状态寄存器 6 (ASIF6) 的第 1 位 (TXBF6) 为 1 时不要将数据写入 TXB6。
 2. 在通信过程中 (当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位和第 6 位 (POWER6, TXE6) 均为 1, 或 ASIM6 的第 7 位和第 5 位 (POWER6, RXE6) 均为 1), 不要通过软件刷新 (写入相同值) TXB6。
 3. 在设置 TXE6 = 1 后至少经历一个基本时钟 (f_{CLK6}) 再为 TXB6 设置发送数据。

(4) 发送移位寄存器 6 (TXS6)

该寄存器将从 TXB6 中传送的数据作为串行数据从 TxD6 引脚中发送。在第一次发送过程中, 数据会在 TXB6 被写入后立即从 TXB6 中发送, 或在连接发送过程中, 在发出一帧后在产生 INTST6 前立即从 TXB6 中发送。数据从 TXB6 中转移, 并在基本时钟的下降沿处从 TxD6 引脚中发送。

TXS6 不能通过程序直接进行操作。

10.3 控制串行接口UART6 的寄存器

串行接口 UART6 由以下九个寄存器控制。

- 异步串行接口操作模式寄存器 6 (ASIM6)
- 异步串行接口接收错误状态寄存器 6 (ASIS6)
- 异步串行接口发送状态寄存器 6 (ASIF6)
- 时钟选择寄存器 6 (CKSR6)
- 波特率发生器控制寄存器 6 (BRGC6)
- 异步串行接口控制寄存器 6 (ASICL6)
- 端口模式寄存器 0 (PM0)
- 端口寄存器 0 (P0)
- 端口输出模式寄存器 0 (POM0)

(1) 异步串行接口操作模式寄存器 6 (ASIM6)

该 8 位寄存器用于控制串行接口 UART6 的串行通信操作。
该寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。
复位信号生成将该寄存器设置为 01H。

备注 在通信过程中（当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位和第 6 位 (POWER6, TXE6) 均为 1, 或 ASIM6 的第 7 位和第 5 位 (POWER6, RXE6) 均为 1), 可以通过软件刷新（写入相同值）ASIM6。

图 10-2. 异步串行接口操作模式寄存器 6 (ASIM6) 的格式 (1/2)

地址: FF50H 复位后: 01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6
POWER6	允许 / 禁止内部操作时钟的操作							
0 ^{注1}	禁止内部操作时钟的操作（时钟固定为低电平）并对内部电路进行异步复位 ^{注2} 。							
1	允许内部操作时钟的操作							
TXE6	允许 / 禁止发送							
0	禁止发送（同步复位发送电路）							
1	允许发送							
RXE6	允许 / 禁止接收							
0	禁止接收（同步复位接收电路）							
1	允许接收							

- 注**
1. 发送期间当 POWER6 = 0 时, TxD6 引脚的输出变为高电平, 而来自 RxD6 引脚的输入则恒为高电平。
 2. 对异步串行接口接收错误状态寄存器 6 (ASIS6), 异步串行接口发送状态寄存器 6 (ASIF6) 以及接收缓冲寄存器 6 (RXB6) 进行复位。

图 10-2. 异步串行接口操作模式寄存器 6 (ASIM6) 的格式 (2/2)

PS61	PS60	发送操作	接收操作
0	0	不输出校验位	无校验接收
0	1	输出零校验	按零校验接收 [*]
1	0	输出奇校验	按奇校验判断
1	1	输出偶校验	按偶校验判断

CL6	指定发送 / 接收数据的字符宽度
0	数据的字符宽度 = 7 位
1	数据的字符宽度 = 8 位

SL6	指定发送数据停止位的个数
0	停止位的个数 = 1
1	停止位的个数 = 2

ISRM6	允许 / 禁止出错时接收完成中断的产生
0	出错时产生“INTSR6”（此时，不产生 INTSR6）
1	出错时产生“INTSR6”（此时，不产生 INTSR6）

注 如果选择“按零校验接收”，则无法判断校验。因此不要设置异步串行接口接收错误状态寄存器 0 (ASIS6) 的第 2 位 (PE6)，这样就不会产生出错中断。

- 注意事项
- 若要开始发送，则将 POWER6 设置为 1，然后将 TXE6 设置为 1。若要停止发送，则先将 TXE6 清零，然后将 POWER6 清零。
 - 若要开始接收，则将 POWER6 设置为 1，然后将 RXE6 设置为 1。若要停止接收，则先将 RXE6 清零，然后将 POWER6 清零。
 - 当向 RxD6 引脚输入高电平时，应先将 POWER6 设置为 1，再将 RXE6 设置为 1。如果在输入低电平时 POWER6 被设置为 1，RXE6 被设置为 1，那么将会启动接收操作。
 - 通过由 CKSR6 设置的基本时钟 (fxCLK6) 使 TXE6 与 RXE6 同步。若要再次允许发送或接收，必须在 TXE6 或 RXE6 清零 (0) 后至少两个基本时钟后将 TXE6 或 RXE6 设置为 1。如果在两个基本时钟内设置 TXE6 或 RXE6，则发送电路或接收电路可能不被初始化。
 - 在设置 TXE6 = 1 后至少经历一个基本时钟 (fxCLK6) 再为 TXB6 设置发送数据。
 - 在对 PS61、PS60 和 CL6 位重写之前，先将 TXE6 和 RXE6 位清零。
 - 在重写 SL6 位前必须将 TXE6 清零。由于总是以“停止位的个数=1”的格式执行接收操作，因此接收不会受到 SL6 设置值的影响。
 - 在重写 ISRM6 位时要确保 RXE6 = 0。

(2) 异步串行接口接收错误状态寄存器 6 (ASIS6)

该寄存器通过串行接口 UART6 来指示接收操作完成时出现的错误状态。它包含有三个错误标志位 (PE6, FE6, OVE6)。

该寄存器可以通过 8 位存储器操作指令来进行只读操作。

复位信号生成, 或将 ASIM6 的第 7 位 (POWER6) 或第 5 位 (RXE6) 清零都可以将该寄存器清除为 00H。读取该寄存器时的读取值为 00H。如果产生接收错误, 那么先读取 ASIS6 再读取接收缓冲寄存器 6 (RXB6) 可将出错标志清零。

图 10-3. 异步串行接口接收错误状态寄存器 6 (ASIS6) 的格式

地址: FF53H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIS6	0	0	0	0	0	PE6	FE6	OVE6

PE6	指示校验错误的状态标志
0	如果 POWER6 = 0 或 RXE6 = 0 时, 或如果 ASIS6 寄存器被读取
1	如果发送数据的校验位与接收完成时的校验位不匹配

FE6	指示帧错误的状态标志
0	如果 POWER6 = 0 或 RXE6 = 0, 或如果 ASIS6 寄存器被读取
1	如果接收完成时未检测到停止位

OVE6	指示溢出错误的状态标志
0	如果 POWER6 = 0 或 RXE6 = 0, 或如果 ASIS6 寄存器被读取
1	如果将接收数据设置到 RXB6 寄存器中且在读取该数据前下一个接收操作已完成

- 注意事项**
1. 根据异步串行接口操作模式寄存器 6 (ASIM6) 的 PS61 和 PS60 位的内容的不同, 对 PE6 位的操作也不同。
 2. 无论停止位的个数是多少, 对于接收数据的停止位来说只会检测第一个停止位。
 3. 如果出现溢出错误, 则下一个接收数据不写入接收缓冲寄存器 6 (RXB6) 中而是被丢弃。
 4. 如果从 ASIS6 读取数据, 将会产生一个等待周期。

(3) 异步串行接口发送状态寄存器6 (ASIF6)

该寄存器通过串行接口 UART6 来指示发送状态，它包含有两个状态标志位 (TXBF6 和 TXSF6)。

当数据从 TXB6 寄存器传送到 TXS6 寄存器中后，通过将下一个发送数据写入 TXB6 寄存器可以使发送能够连续进行而不被中断，即便是在中断产生期间也不受影响。

该寄存器可以通过 8 位存储器操作指令来进行只读操作。

复位信号生成，或将 ASIM6 的第 7 位 (POWER6) 或第 6 位 (TXE6) 清零都可以将该寄存器清除为 00H。

图 10-4. 异步串行接口发送状态寄存器 6 (ASIS6) 的格式

地址: FF55H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIF6	0	0	0	0	0	0	TXBF6	TXSF6

TXBF6	发送缓冲数据标志
0	如果 POWER6 = 0 或 TXE6 = 0, 或如果数据被传送到发送移位寄存器 6 (TXS6) 中
1	如果数据被写入发送缓冲寄存器 6 (TXB6) 中 (如果数据已在于 TXB6 中)

TXSF6	发送移位寄存器数据标志
0	如果 POWER6 = 0 或 TXE6 = 0, 或如果发送完成后下一个数据没有从发送缓冲寄存器 6 (TXB6) 发出
1	如果数据从发送缓冲寄存器 6 (TXB6) 中发出 (如果数据仍在发送过程中)

- 注意事项**
1. 为了能够连续发送数据，应将第一个发送的数据（第一个字节）写入 TXB6 寄存器中。请务必检查 TXBF6 标志位为“0”。如果正确，将下一个发送数据（第二个字节）写入 TXB6 寄存器中。如果 TXBF6 标志为“1”时将数据写入 TXB6 寄存器，则无法保证发送数据的正确性。
 2. 若要在连续发送结束时初始化发送单元，那么在产生发送完成中断后必须核准 TXSF6 标志位为“0”，然后再执行初始化操作。如果在 TXSF6 标志为“1”时执行初始化操作，将无法保证发送数据的正确性。

(4) 时钟选择寄存器 6 (CKSR6)

该寄存器用于选择串行接口 UART6 的基本时钟。

CKSR6 可以通过 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器清除为 00H。

备注 在通信过程中（当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位和第 6 位 (POWER6, TXE6) 均为 1, 或 ASIM6 的第 7 位和第 5 位 (POWER6, RXE6) 均为 1），可以通过软件刷新（写入相同值）CKSR6。

图 10-5. 时钟选择寄存器 6 (CKSR6) 的格式

地址: FF56H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60

TPS63	TPS62	TPS61	TPS60		基本时钟 (f _{CLK6}) 选择	
					f _{PRS} = 2 MHz	f _{PRS} = 4 MHz
0	0	0	0	f _{PRS}	2 MHz	4 MHz
0	0	0	1	f _{PRS} /2	1 MHz	2 MHz
0	0	1	0	f _{PRS} /2 ²	500 kHz	1 MHz
0	0	1	1	f _{PRS} /2 ³	250 kHz	500 kHz
0	1	0	0	f _{PRS} /2 ⁴	125 kHz	250 kHz
0	1	0	1	f _{PRS} /2 ⁵	62.5 kHz	125 kHz
0	1	1	0	f _{PRS} /2 ⁶	31.25 kHz	62.5 kHz
0	1	1	1	f _{PRS} /2 ⁷	15.625 kHz	31.25 kHz
1	0	0	0	f _{PRS} /2 ⁸	7.813 kHz	15.625 kHz
1	0	0	1	f _{PRS} /2 ⁹	3.906 kHz	7.813 kHz
1	0	1	0	f _{PRS} /2 ¹⁰	1.953 kHz	3.906 kHz
1	0	1	1	TM50 输出 [#]		
其他情况				禁止设置		

注 当选择 TM50 输出作为基本时钟时，应注意下面几点。

- 计数时钟通过 TM50 和 CR50 的匹配来清零和启动 (TMC506 = 0) 的模式
首先启动 8 位定时器 / 事件计数器 50 的操作，然后将允许定时器 F/F 的反转操作。(TMC501 = 1)。
- PWM 模式 (TMC506 = 1)
首先启动 8 位定时器 / 事件计数器 50 的操作，然后设置计数时钟来使得占空比= 50%。
任意模式下都没必要允许 (TOE50 = 1) TO50 输出。

注意事项 在重写 TPS63 到 TPS60 位时，必须确保 POWER6 = 0。

- 备注
1. f_{PRS}: 外围硬件时钟频率
 2. TMC506: 8 位定时器模式控制寄存器 50 (TMC50) 的第 6 位
TMC501: TMC50 的第 1 位

(5) 波特率发生器控制寄存器 6 (BRGC6)

该寄存器用于设置串行接口 UART6 的 8 位计数器的分频值。

BRGC6 可以通过 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器设置为 FFH。

备注 在通信过程中（当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位和第 6 位 (POWER6, TXE6) 均为 1，或 ASIM6 的第 7 位和第 5 位 (POWER6, RXE6) 均为 1），可以通过软件刷新（写入相同值）BRGC6。

图 10-6. 波特率发生器控制寄存器 6 (BRGC6) 的格式

地址：FF57H 复位后：FFH R/W

符号	7	6	5	4	3	2	1	0
BRGC6	MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	k	8 位计数器的输出时钟选择
0	0	0	0	0	0	×	×	×	禁止设置
0	0	0	0	0	1	0	0	4	$f_{XCLK6}/4$
0	0	0	0	0	1	0	1	5	$f_{XCLK6}/5$
0	0	0	0	0	1	1	0	6	$f_{XCLK6}/6$
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	$f_{XCLK6}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK6}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK6}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK6}/255$

- 注意事项**
1. 在重写 MDL67 到 MDL60 位时，必须确保 ASIM6 寄存器的第 6 位 (TXE6) 和第 5 位 (RXE6) = 0。
 2. 波特率等于 8 位计数器的输出时钟除以 2。

- 备注**
1. f_{XCLK6} : 根据 CKSR6 寄存器的 TPS63 到 TPS60 位选择的基本时钟的频率
 2. k: 使用 MDL67 到 MDL60 位设置的值 (k = 4, 5, 6, ..., 255)
 3. ×: 不考虑

(6) 异步串行接口控制寄存器 6 (ASICL6)

该寄存器用于控制串行接口 UART6 的串行通信操作。

ASICL6 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器设置为 16H。

注意事项 在通信过程中（当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位和第 6 位 (POWER6, TXE6) 均为 1，或 ASIM6 的第 7 位和第 5 位 (POWER6, RXE6) 均为 1），可以通过软件刷新（写入相同值）ASICL6。

图 10-7. 异步串行接口控制寄存器 6 (ASICL6) 的格式

地址： FF58H 复位后： 16H R/W

符号	7	6	5	4	3	2	1	0
ASICL6	0	0	0	0	0	0	DIR6	TXDLV6

DIR6	首位指定
0	MSB
1	LSB

TXDLV6	允许 / 禁止反转 TxD6 输出
0	TxD6 的正常输出
1	TxD6 的反转输出

注意事项 重新写 DIR6 和 TXDLV6 位之前，应先将 TXE6 和 RXE6 位清除为 0。

(7) 端口模式寄存器 0 (PM0)

该寄存器以 1 位为单位对端口 1 的输入 / 输出进行设置。

当 P02/TxD6/TI000 引脚作为串行接口的数据输出时，将 PM02 清 0 并将 P02 的输出锁存设置为 1。

当 P03/RxD6/TI010/TO00 引脚作为串行接口数据输入时，将 PM03 置 1。此时，P03 输出锁存的值可能是 0 或 1。

PM0 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器设置为 FFH。

图 10-8. 端口模式寄存器 0 (PM0) 的格式

地址: FF20H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	P0n 引脚 I/O 模式选择 (n = 0 到 7)
0	输出模式 (输出缓冲开)
1	输入模式 (输出缓冲关)

(8) 端口输出模式电阻 (POM0)

该寄存器用于设置端口 0 的输出模式。

该寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将这些寄存器设置为 00H。

图 10-9. 上拉电阻选择寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
POM0	POM07	POM06	POM05	POM04	POM03	POM02	POM01	POM00	FF38H	00H	R/W

POM0n	P0n 引脚输出模式选择 (n = 0 到 7)
0	CMOS 输出
1	N-ch 开漏输出 (P07: P-ch 开漏输出)

10.4 串行接口UART6 的操作

串行接口 UART6 有以下两种模式。

- 操作停止模式
- 异步串行接口（UART）模式

10.4.1 操作停止模式

在此模式下，不能进行串行通信，因此，可降低功耗。此外，该模式中引脚可作为普通端口引脚来使用。若要设置操作停止模式，应将 ASIM6 的第 7，6 和 5 位（POWER6，TXE6 和 RXE6）清零。

(1) 使用的寄存器

由异步串行接口操作模式寄存器 6（ASIM6）来设置操作停止模式。

ASIM6 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该寄存器设置为 01H。

地址：FF50H 复位后：01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6
POWER6	允许 / 禁止内部操作时钟的操作							
0 ^{#1}	禁止内部操作时钟的操作（时钟固定为低电平）并对内部电路进行异步复位 ^{#2} 。							
TXE6	允许 / 禁止发送							
0	禁止发送操作（同步复位发送电路）							
RXE6	允许 / 禁止接收							
0	禁止接收（同步复位接收电路）							

- 注**
1. 发送期间当 POWER6 = 0 时，TxD6 引脚的输出变为高电平，而来自 RxD6 引脚的输入则恒为高电平。
 2. 对异步串行接口接收错误状态寄存器 6（ASIS6），异步串行接口发送状态寄存器 6（ASIF6）以及接收缓冲寄存器 6（RXB6）进行复位。

注意事项 将 TXE6 和 RXE6 清零后再对 POWER6 清零可停止操作。

若要启动通信时，则应先将 POWER6 设置为 1，然后将 TXE6 或 RXE6 设置为 1。

备注 若要将 RxD6/P03/TI010/TO00 和 TxD6/P02/TI000 作为通用端口引脚使用，请参见 第四章 端口功能。

10.4.2 异步串行接口 (UART) 模式

在该模式下，在传送了起始位后，紧接着发送 / 接收 1 字节数据，然后可执行全双工操作。
包含有一个专用的 UART 波特率发生器，这样通信选择的波特率范围较大。

(1) 使用的寄存器

- 异步串行接口操作模式寄存器 6 (ASIM6)
- 异步串行接口接收错误状态寄存器 6 (ASIS6)
- 异步串行接口发送状态寄存器 6 (ASIF6)
- 时钟选择寄存器 6 (CKSR6)
- 波特率发生器控制寄存器 6 (BRGC6)
- 异步串行接口控制寄存器 6 (ASICL6)
- 端口模式寄存器 0 (PM0)
- 端口寄存器 0 (P0)
- 端口输出模式寄存器 0 (POM0)

在 UART 模式中设置操作的基本过程如下。

- <1> 设置 CKSR6 寄存器 (参见图 10-5)。
- <2> 设置 BRGC6 寄存器 (参见图 10-6)。
- <3> 设置 ASIM6 寄存器的第 0 位到第 4 位 (ISRM6, SL6, CL6, PS60, PS61) (参见图 10-2)。
- <4> 设置 ASICL6 寄存器的第 0 位和第 1 位 (TXDLV6, DIR6) (参见图 10-7)。
- <5> 设置 ASIM6 寄存器的第 7 位 (POWER6) = 1。
- <6> 设置 ASIM6 寄存器的第 6 位 (TXE6) = 1。 → 允许发送。
设置 ASIM6 寄存器的第 5 位 (RXE6) = 1。 → 允许接收。
- <7> 将数据写入发送缓冲寄存器 6 (TXB6)。 → 开始发送数据。

注意事项 在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。

寄存器设置与引脚之间的关系如下所示。

表 10-2. 寄存器设置与引脚之间的关系

POWER6	TXE6	RXE6	PM02	P02	PM03	P03	UART6 操作	引脚功能	
								TxD6/P02/TI000	RxD6/P03/TI010/TO00
0	0	0	× ^注	× ^注	× ^注	× ^注	停止	P02/TI000	P03/ TI010/TO00
1	0	1	× ^注	× ^注	1	×	接收	P02/TI000	RxD6
	1	0	0	1	× ^注	× ^注	发送	TxD6	P03/TI010/TO00
	1	1	0	1	1	×	发送/ 接收	TxD6	RxD6

注 可以设为端口功能或 16 位定时器 / 事件计数器 00。

- 备注** x: 不考虑
 POWER6: 异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位
 TXE6: ASIM6 的第 6 位
 RXE6: ASIM6 的第 5 位
 PM0x: 端口模式寄存器
 P0x: 端口输出锁存

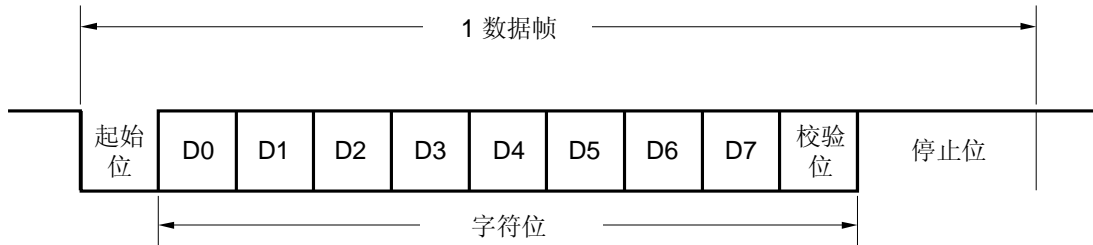
(2) 通信操作

(a) 正常发送 / 接收数据的格式和波形示例

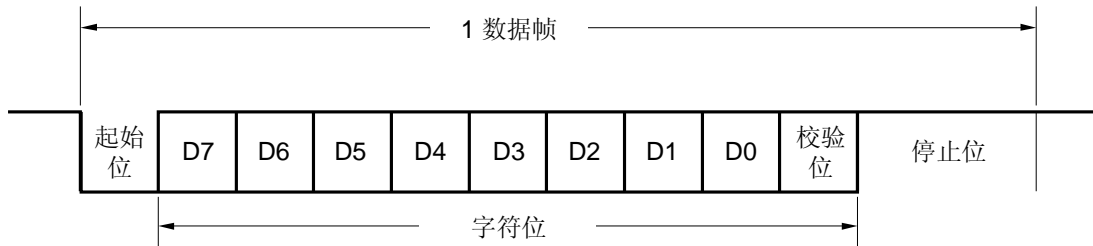
图 10-10 和 10-11 为正常发送 / 接收数据的格式和波形示例。

图 10-10. 正常 UART 发送 / 接收数据的格式

1. LSB 优先发送 / 接收



2. MSB 优先发送 / 接收



一个数据帧由以下各位组成。

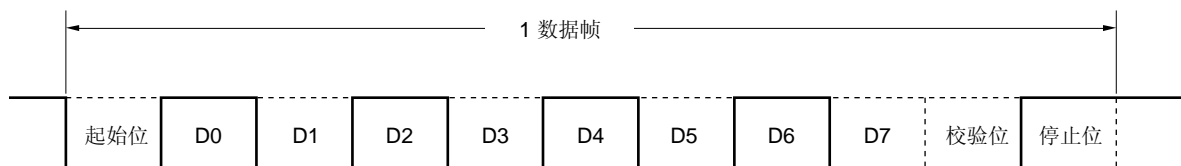
- 起始位 ...1 位
- 字符位 ...7 或 8 位
- 校验位 ...偶校验, 奇校验, 零校验或无校验
- 停止位 ...1 或 2 位

由异步串行接口操作模式寄存器 6 (ASIM6) 来指定一个数据帧中字符位的宽度, 校验方式和停止位的宽度。
 由异步串行接口控制寄存器 6 (ASICL6) 的第 1 位 (DIR6) 来指定数据是通过 LSB 优先还是 MSB 优先来进行通信。

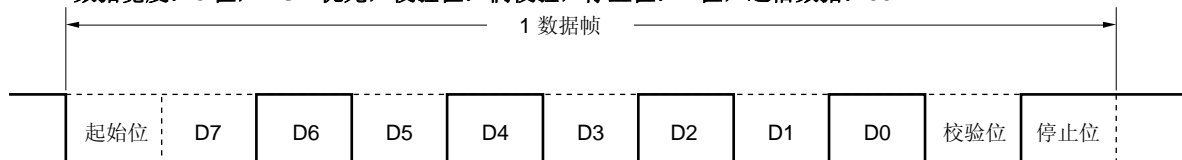
由 ASICL6 的第 0 位 (TXDLV6) 来指定 Tx/D6 引脚输出的是正常数据还是反向数据。

图 10-11. 正常 UART 发送 / 接收数据波形示例

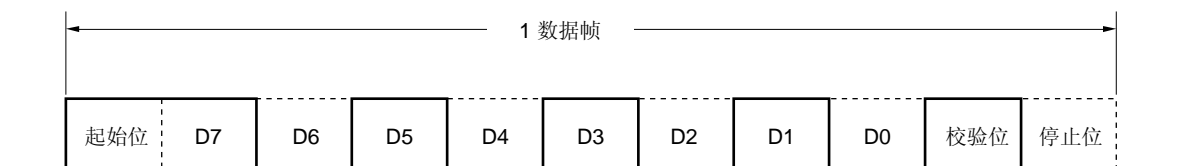
1. 数据宽度：8 位，LSB 优先，校验位：偶校验，停止位：1 位，通信数据：55H



2. 数据宽度：8 位，MSB 优先，校验位：偶校验，停止位：1 位，通信数据：55H



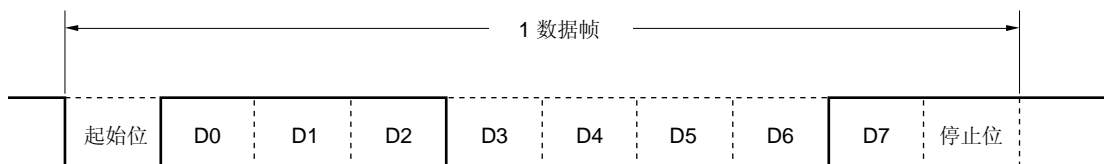
3. 数据宽度：8 位，MSB 优先，校验位：偶校验，停止位：1 位，通信数据：55H，TxD6 引脚反向输出



4. 数据宽度：7 位，LSB 优先，校验位：奇校验，停止位：2 位，通信数据：36H



5. 数据宽度：8 位，LSB 优先，校验位：无，停止位：1 位，通信数据：87H



(b) 校验方式与操作

校验位用于检测数据通信中的位错误。通常在发送和接收两端可采用相同类型的校验位。当采用偶校验和奇校验时，可检测到 1 位（奇数位）错误。而采用零校验和无校验时，则不能检测到错误。

(i) 偶校验

• 发送

控制发送数据，包括校验位，使得为“1”的位的个数为偶数。

校验位的取值如下。

如果发送数据有奇数个位数为“1”： 1

如果发送数据有偶数个位数为“1”： 0

• 接收

计算接收数据包括校验位中为“1”的位的个数。如果是奇数，则产生校验错误。

(ii) 奇校验

• 发送

跟偶校验不同的是，控制发送数据，包括校验位，使得为“1”的位的个数为奇数。

如果发送数据有奇数个位数为“1”： 0

如果发送数据有偶数个位数为“1”： 1

• 接收

计算接收数据包括校验位中为“1”的位的个数。如果是偶数，则产生校验错误。

(iii) 零校验

无论发送数据为何值，数据发送时校验位都将清零。

接收数据时不检测校验位。因此，无论校验位是“0”还是“1”，都不会产生校验错误。

(iv) 无校验

没有校验位被添加到发送数据中。

进行接收操作时是在假设接收数据中没有校验位的情况下执行接收操作的。由于不存在校验位，因此也不会产生校验错误。

(c) 正常发送

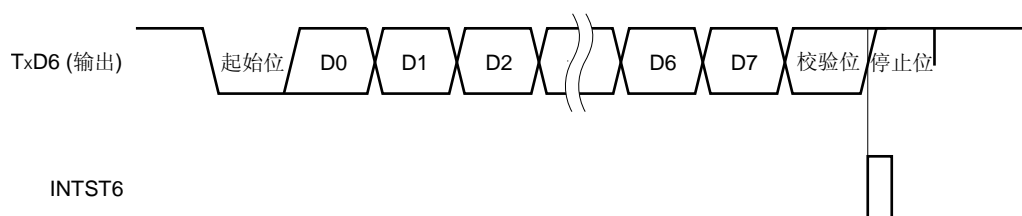
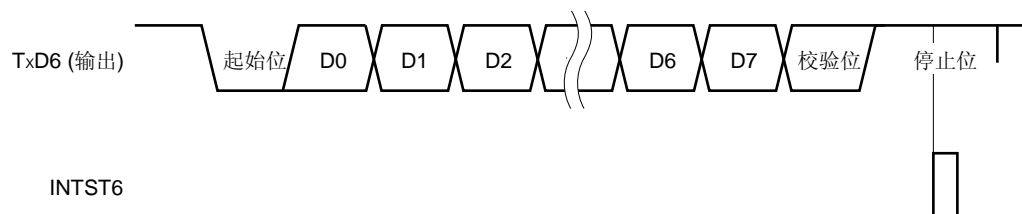
在将异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 设为 1, 然后将 ASIM6 的第 6 位 (TXE6) 设为 1 时, 允许发送。将待发送的数据写入发送缓冲寄存器 6 (TXB6) 中, 可启动发送操作。起始位, 校验位和停止位会自动被添加到数据中。

启动发送操作时, TXB6 中的数据会被传送到发送移位寄存器 6 (TXS6) 中。之后, 发送数据将依次从 TXS6 被输出到 TxD6 引脚。当发送结束时, 由 ASIM6 设置的校验位和停止位被添加到数据中, 并产生发送完成中断请求 (INTST6)。

当下一个发送数据写入 TXB6 时, 发送操作才结束。

图 10-12 显示了发送完成中断请求 (INTST6) 的时序。最后一个停止位输出后马上就产生该中断。

图 10-12. 正常的发送完成中断请求时序

1. 停止位宽度: 1**2. 停止位宽度: 2**

(d) 连续发送

一旦发送移位寄存器 6 (TXS6) 启动移位操作, 下一个发送数据马上可以被写入发送缓冲寄存器 6 (TXB6)。因此, 当发送完一数据帧后, 即使在执行 INTST6 中断服务时也能够连续发送数据, 从而实现高效率的通信速率。此外, 当产生发送完成中断时通过读取异步串行发送状态寄存器 6 (ASIF6) 的第 0 位 (TXSF6), 可以对 TXB6 寄存器进行两次有效的写操作 (2 个字节) 而无须等待发送一数据帧的时间。为了能够连续发送数据, 必须根据 ASIF6 寄存器来检查发送状态, 判断是否可以对 TXB6 寄存器进行写操作, 如果可以, 则写入数据。

注意事项 在连续发送期间, ASIF6 寄存器的 TXBF6 和 TXSF6 标志从“10”变成“11”, 再变成“01”。因此, 若要检查状态, 则不能通过结合使用 TXBF6 和 TXST6 标志来进行判断。在连续发送期间只读到 TXBF6 标志。

TXBF6	写入 TXB6 寄存器
0	允许写入
1	禁止写入

注意事项 为了能够连续发送数据, 应将第一个发送的数据 (第一个字节) 写入 TXB6 寄存器中。请务必检查 TXBF6 标志位为“0”。如果正确, 将下一个发送数据 (第二个字节) 写入 TXB6 寄存器中。如果 TXBF6 标志为“1”时将数据写入 TXB6 寄存器, 则无法保证发送数据的正确性。

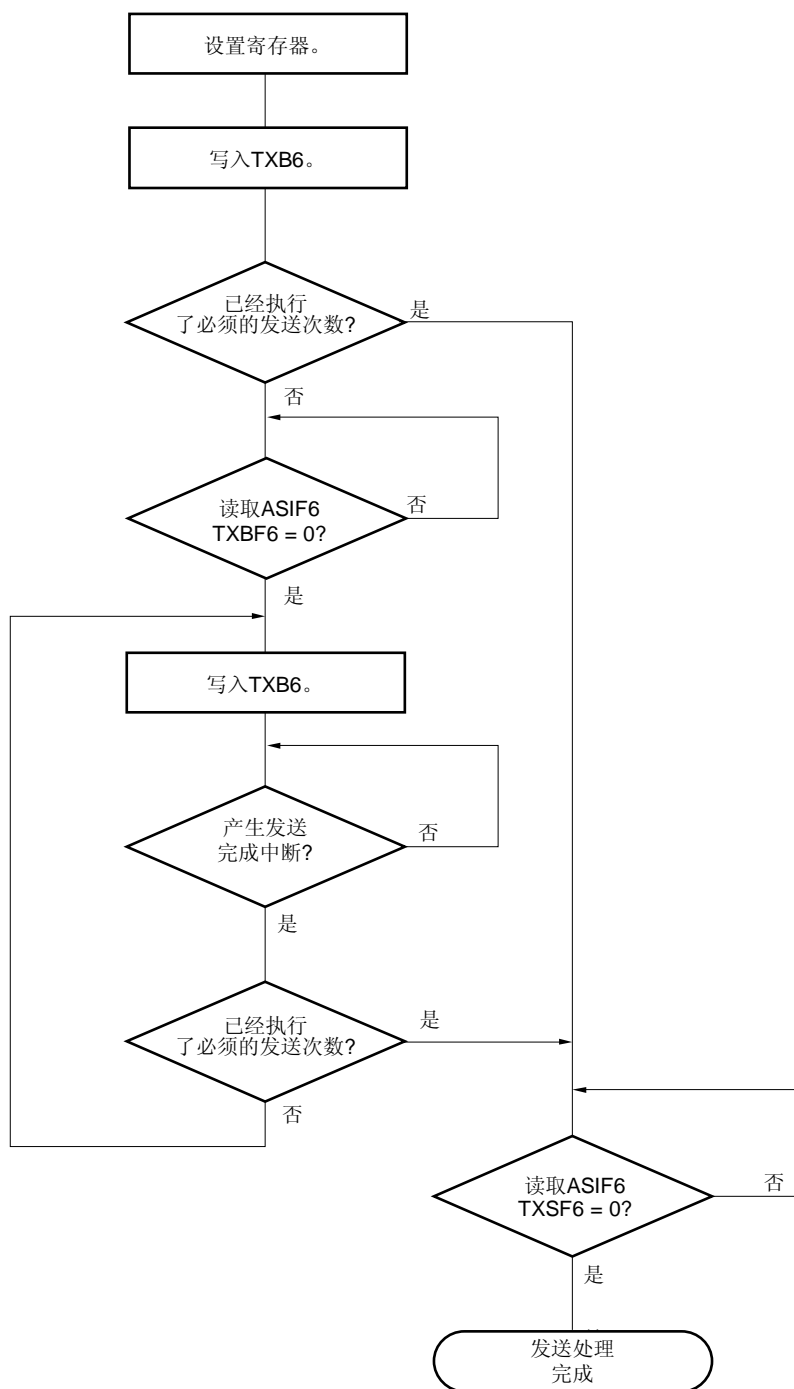
可以使用 TXSF6 标志来检测通信状态。

TXSF6	发送状态
0	发送完成
1	发送正在进行中

- 注意事项**
1. 若要在连续发送结束时初始化发送单元, 那么在产生发送完成中断后必须核准 TXSF6 标志位为“0”, 然后再执行初始化操作。如果在 TXSF6 标志为“1”时执行初始化操作, 将无法保证发送数据的正确性。
 2. 在连续发送期间, 在发送完一数据帧后执行 INTST6 中断服务之前下一个发送操作可能已经完成。计数测量时, 可通过开发一个可以计数发送数据的个数的程序或者通过参考 TXSF6 标志来实现检测。

图 10-13 为连续发送操作流程示例。

图 10-13. 连续发送操作流程示例

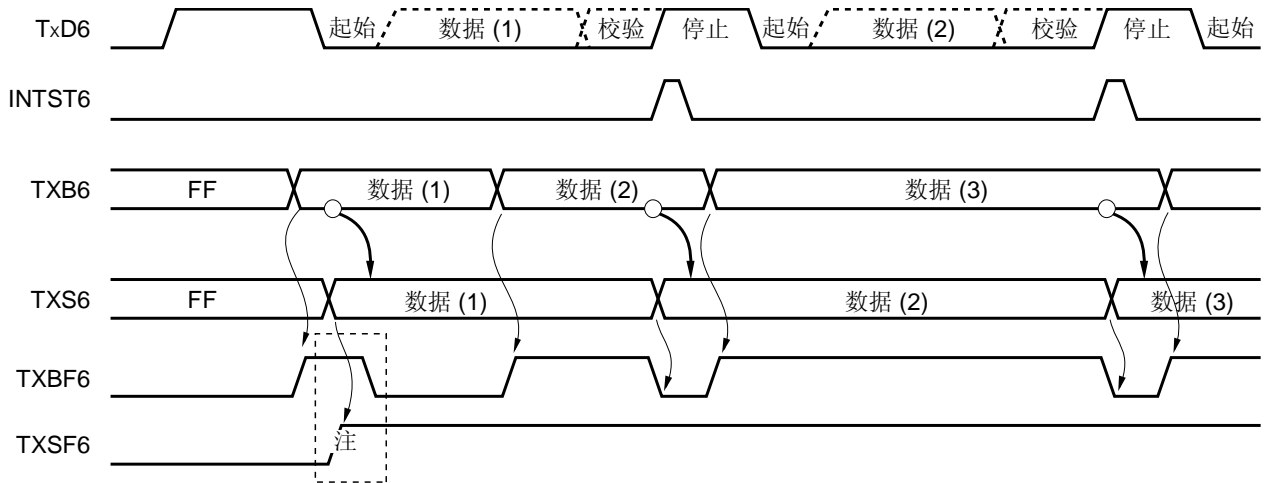


备注

- TXB6: 发送缓冲寄存器 6
- ASIF6: 异步串行接口发送状态寄存器 6
- TXBF6: ASIF6 的第 1 位 (发送缓冲数据标志)
- TXSF6: ASIF6 的第 0 位 (发送移位寄存器数据标志)

图 10-14 显示了启动连续发送操作的时序，而图 10-15 显示了结束连续发送操作的时序。

图 10-14. 启动连续发送操作的时序

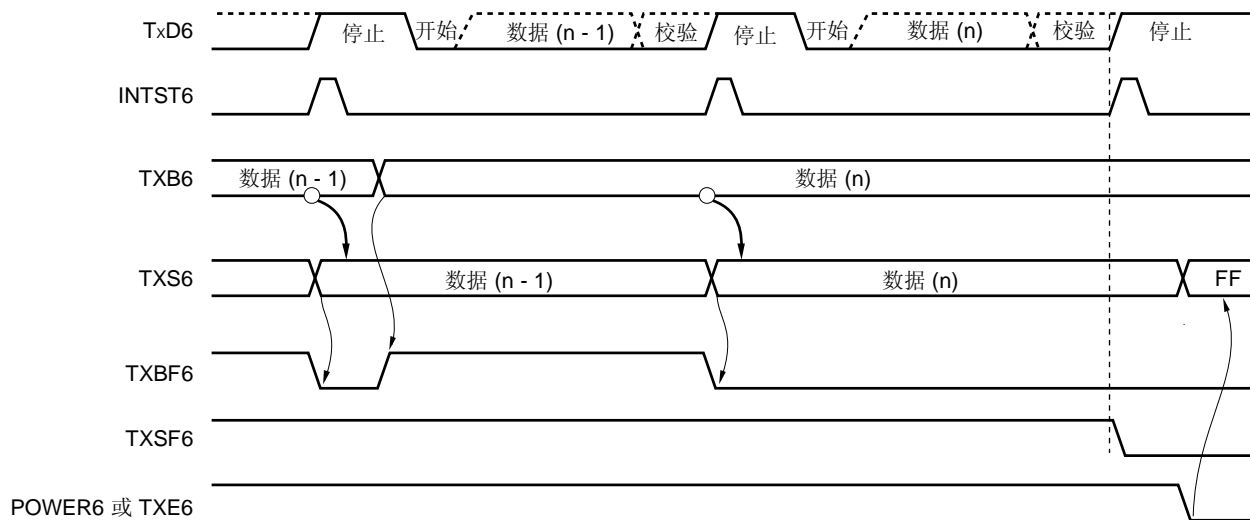


注 在读取 ASIF6 时，存在 TXBF6 和 TXSF6 = 1, 1 的情况。因此，仅用 TXBF6 位就可以判断是否允许进行写操作。

备注

- TxD6: TxD6 引脚 (输出)
- INTST6: 中断请求信号
- TXB6: 发送缓冲寄存器 6
- TXS6: 发送移位寄存器 6
- ASIF6: 异步串行接口发送状态寄存器 6
- TXBF6: ASIF6 的第 1 位
- TXSF6: ASIF6 的第 0 位

图 10-15. 结束连续发送操作的时序



- 备注**
- TxD6: TxD6 引脚（输出）
 - INTST6: 中断请求信号
 - TXB6: 发送缓冲寄存器 6
 - TXS6: 发送移位寄存器 6
 - ASIF6: 异步串行接口发送状态寄存器 6
 - TXBF6: ASIF6 的第 1 位
 - TXSF6: ASIF6 的第 0 位
 - POWER6: 异步串行接口操作模式寄存器（ASIM6）的第 7 位
 - TXE6: 异步串行接口操作模式寄存器（ASIM6）的第 6 位

(e) 正常接收

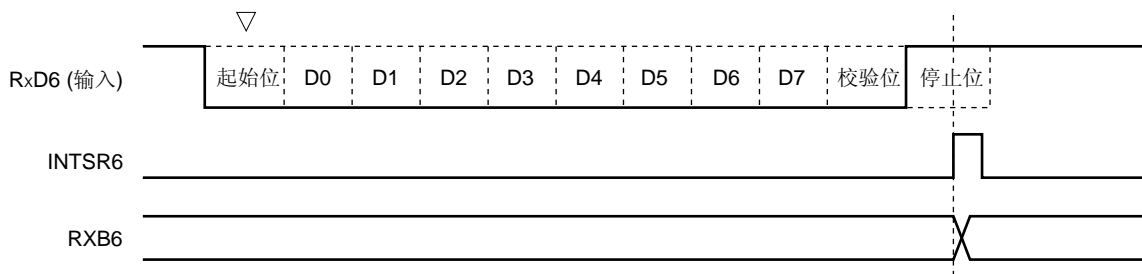
当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 被设为 1, 然后将 ASIM6 的第 5 位 (RXE6) 设为 1 时, 将会允许接收并对 RxD6 引脚输入进行采样。

当检测到 RxD6 引脚输入的下降沿时, 波特率发生器的 8 位计数器开始计数。当计数值达到波特率发生器控制寄存器 6 (BRGC6) 的设置值时, 会再次采样 RxD6 引脚的输入信号 (如图 10-16 中的 ▽)。如果此时 RxD6 引脚为低电平, 则会被认为是起始位。

当检测到起始位时, 开始接收, 以设置的波特率将串行数据依次存入接收移位寄存器 (RXS6) 中。当接收到停止位时, 会产生接收完成中断 (INTSR6), 并且 RXS6 的数据会被写入接收缓冲寄存器 6 (RXB6) 中。然而, 如果出现溢出错误 (OVE6), 则接收到的数据不会被写入 RXB6。

即使在接收过程中出现校验错 (PE6), 接收仍然会继续直至接收到停止位, 而在接收完成后会产生一个接收错误中断 (INTSR6/INTSRE6)。

图 10-16. 接收完成中断请求时序



- 注意事项**
1. 如果出现接收错误, 那么先后读取 ASIS6 和 RXB6 可将出错标志清零。否则, 当接收到下一个数据时将会产生溢出错误, 而且接收错误状态保持不变。
 2. 接收始终按“停止位的个数 = 1”的情况执行。第二个停止位将会被忽略。
 3. 在读取 RXB6 之前, 必须先读取异步串行接口接收错误状态寄存器 6 (ASIS6)。

(f) 接收错误

在接收期间有三种出错情况：校验错误，帧错误或溢出错误。如果作为数据接收的结果，异步串行接口接收错误状态寄存器 6 (ASIS6) 的出错标志位被设置，则将会产生一个接收错误中断请求 (INTSR6/INTSRE6)。在执行接收错误中断 (INTSR6/INTSRE6) 服务时通过读取 ASIS6 的内容，可确定接收期间出现的是何种错误 (参见图 10-3)。

读取 ASIS6 后，其内容将会被清零。

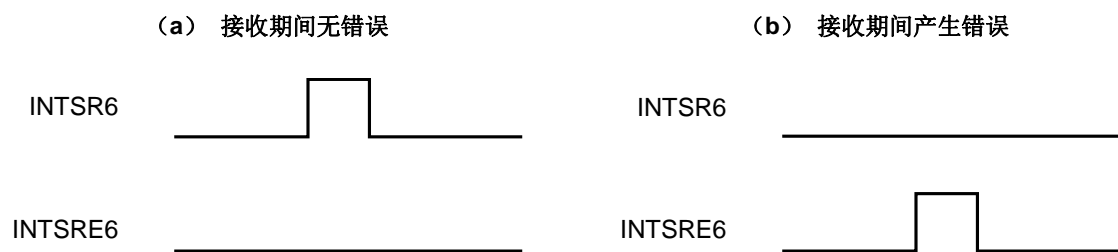
表 10-3. 产生接收错误的原因

接收错误	原因
校验错误	为发送所指定的校验位与接收数据校验位不匹配
帧错误	没检测到停止位
溢出错误	从接收缓冲寄存器 6 (RXB6) 中读取数据前已完成下一个数据的接收操作

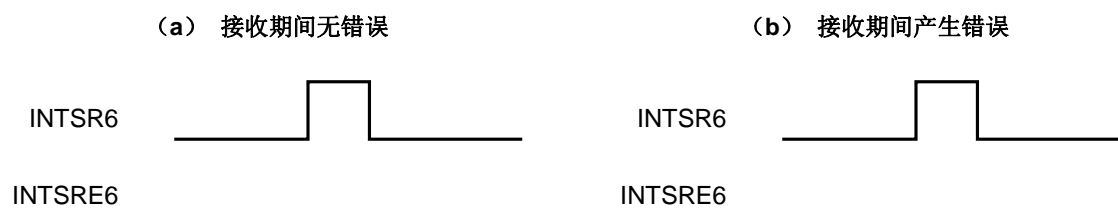
通过将异步串行接口操作模式寄存器 6 (ASIM6) 的第 0 位 (ISRM6) 清零，可以把接收错误中断划分为接收完成中断 (INTSR6) 和错误中断 (INTSRE6) 两部分。

图 10-17. 接收错误中断

1. 如果 ISRM6 被清零 (接收完成中断 (INTSR6) 和接收错误中断 (INTSRE6) 是分开的)



2. 如果 ISRM6 被设为 1 (接收错误中断包含在 INTSR6 中)



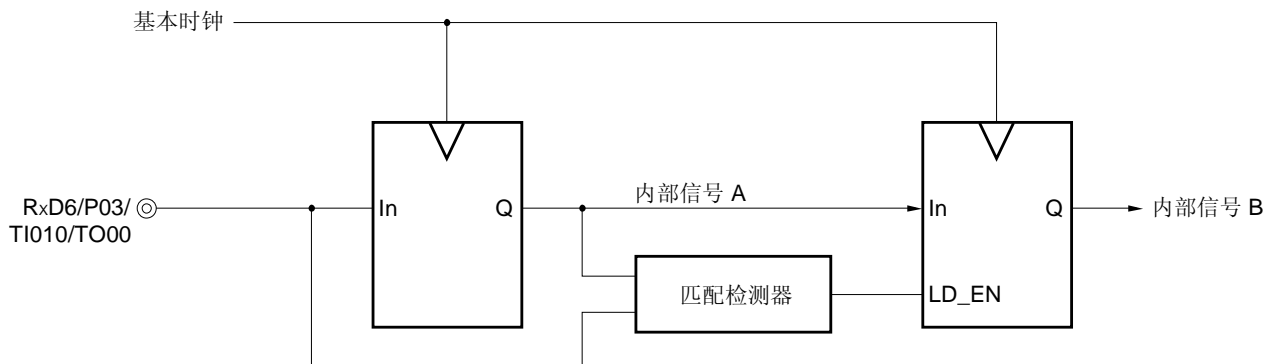
(g) 接收数据的噪声过滤器

使用由预分频模块输出的基本时钟采样 RXD6 信号。

如果两次采样值相同，则匹配检测器的输出会发生变化，并把采样的数据作为输入数据。

基于如图 10-18 所示的电路结构，接收操作的内部处理过程会将外部信号状态开始延迟两个时钟才执行。

图 10-18. 噪声过滤器电路

**10.4.3 专用波特率发生器**

专用波特率发生器由一个源时钟选择器和一个 8 位可编程计数器组成，用于为 UART6 的发送 / 接收生成一个串行时钟。

分别使用不同的 8 位计数器用于发送和接收。

(1) 波特率发生器的配置

- 基本时钟**

当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 等于 1 时，由时钟选择寄存器 6 (CKSR6) 的第 3 位到第 0 位 (TPS63 到 TPS60) 所选择的时钟将会提供给每个模块。该时钟被称为基本时钟，它的频率用 f_{CLK6} 表示。当 $POWER6 = 0$ 时，基本时钟恒为低电平。
- 发送计数器**

当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 或第 6 位 (TXE6) 等于 0 时，该计数器停止计数，并被清零。

当 $POWER6 = 1$ 且 $TXE6 = 1$ 时计数器开始计数。

当第一个发送数据写入发送缓冲寄存器 6 (TXB6) 时，计数器被清零。

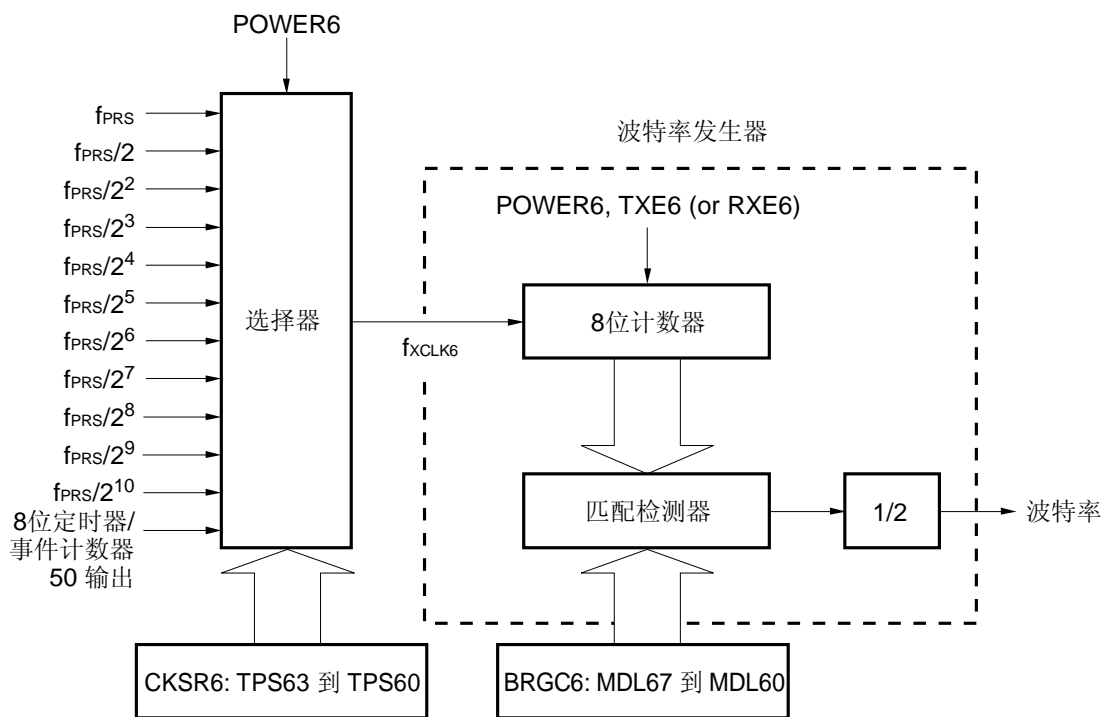
如果连续发送数据，那么当发送完 1 帧数据时计数器将再次被清零。如果后面没有数据要发送，那么计数器不被清零，而是继续计数直至 $POWER6$ 或 $TXE6$ 被清零为止。
- 接收计数器**

当异步串行接口操作模式寄存器 5 (ASIM6) 的第 7 位 (POWER6) 或第 6 位 (RXE6) 等于 0 时，该计数器停止计数，并被清零。

当检测到起始位时，计数器开始计数。

在收到一帧数据后计数器停止计数，直到下一个起始位被检测到时再重新开始计数。

图 10-19. 波特率发生器的配置



备注

- POWER6: 异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位
- TXE6: ASIM6 的第 6 位
- RXE6: ASIM6 的第 5 位
- CKSR6: 时钟选择寄存器 6
- BRGC6: 波特率发生器控制寄存器 6

(2) 串行时钟的产生

通过使用时钟选择寄存器 6 (CKSR6) 和波特率发生器控制寄存器 6 (BRGC6) 可以指定要生成的串行时钟。使用 CKSR6 的第 3 位到第 0 位 (TPS63 到 TPS60) 可以设置输入到 8 位计数器的时钟, 而使用 BRGC6 的第 7 位到第 0 位 (MDL67 到 MDL60) 可以设置 8 位计数器的分频值 ($f_{XCLK6}/4$ 到 $f_{XCLK6}/255$)。

10.4.4 波特率的计算

(1) 波特率计算公式

波特率可通过以下表达式来进行计算。

$$\bullet \text{ 波特率} = \frac{f_{\text{XCLK6}}}{2 \times k} \text{ [bps]}$$

f_{XCLK6} : 根据 CKSR6 的 TPS63 到 TPS60 位选择的基本时钟的频率

k: 根据 BRGC6 寄存器的 MDL67 到 MDL60 位设置的值 ($k = 4, 5, 6, \dots, 255$)

表 10-4. TPS63 到 TPS60 的设定值

TPS63	TPS62	TPS61	TPS60	基本时钟 (f_{XCLK6}) 选择		
				$f_{\text{PRS}} = 2 \text{ MHz}$	$f_{\text{PRS}} = 4 \text{ MHz}$	
0	0	0	0	f_{PRS}	2 MHz	4 MHz
0	0	0	1	$f_{\text{PRS}}/2$	1 MHz	2 MHz
0	0	1	0	$f_{\text{PRS}}/2^2$	500 kHz	1 MHz
0	0	1	1	$f_{\text{PRS}}/2^3$	250 kHz	500 kHz
0	1	0	0	$f_{\text{PRS}}/2^4$	125 kHz	250 kHz
0	1	0	1	$f_{\text{PRS}}/2^5$	62.5 kHz	125 kHz
0	1	1	0	$f_{\text{PRS}}/2^6$	31.25 kHz	62.5 kHz
0	1	1	1	$f_{\text{PRS}}/2^7$	15.625 kHz	31.25 kHz
1	0	0	0	$f_{\text{PRS}}/2^8$	7.813 kHz	15.625 kHz
1	0	0	1	$f_{\text{PRS}}/2^9$	3.906 kHz	7.813 kHz
1	0	1	0	$f_{\text{PRS}}/2^{10}$	1.953 kHz	3.906 kHz
1	0	1	1	TM50 输出 [#]		
其他情况				禁止设置		

注 当选择 TM50 输出作为基本时钟时，应注意以下几点。

- 计数时钟通过 TM50 和 CR50 的匹配来清零和启动 ($\text{TMC506} = 0$) 的模式
首先启动 8 位定时器 / 事件计数器 50 的操作，然后将允许定时器 F/F 的反转操作。 ($\text{TMC501} = 1$)。
- PWM 模式 ($\text{TMC506} = 1$)
首先启动 8 位定时器 / 事件计数器 50 的操作，然后设置计数时钟，来使得占空比= 50%。
任意模式下都没必要允许 ($\text{TOE50} = 1$) TO50 输出。

(2) 波特率误差

波特率误差可通过以下表达式来进行计算。

$$\bullet \text{ 误差 (\%)} = \left[\frac{\text{实际波特率 (有误差的波特率)}}{\text{预期波特率 (正确的波特率)}} - 1 \right] \times 100 [\%]$$

- 注意事项 1. 发送期间必须保持波特率误差在接收端允许的误差范围内。
2. 接收期间, 确保波特率误差满足“(4) 接收期间允许的波特率范围”中所描述的范围。

例: 基本时钟的频率 = 4 MHz = 4, 000, 000 Hz
BRGC6 寄存器 MDL67 到 MDL60 位的设定值 = 00001101B (k = 13)
目标波特率 = 153600 bps

$$\begin{aligned} \text{波特率} &= 4 \text{ M} / (2 \times 13) \\ &= 4000000 / (2 \times 13) = 153846 [\text{bps}] \end{aligned}$$

$$\begin{aligned} \text{误差} &= (153846/153600 - 1) \times 100 \\ &= 0.16 [\%] \end{aligned}$$

(3) 波特率设置示例

表 10-5. 波特率发生器的设置数据

波特率 [bps]	f _{PRS} = 2.0 MHz				f _{PRS} = 4.0 MHz			
	TPS63- TPS60	k	计算值	ERR [%]	TPS63- TPS60	k	计算值	ERR [%]
300	8H	13	301	0.16	8H	26	301	0.16
600	7H	13	601	0.16	7H	26	601	0.16
1200	6H	13	1202	0.16	6H	26	1202	0.16
2400	5H	13	2404	0.16	5H	26	2404	0.16
4800	4H	13	4808	0.16	4H	26	4808	0.16
9600	3H	13	9615	0.16	3H	26	9615	0.16
19200	2H	13	19231	0.16	2H	26	19231	0.16
24000	1H	21	23810	-0.79	1H	42	23810	-0.79
31250	1H	16	31250	0	1H	32	31250	0
38400	1H	13	38462	0.16	1H	26	38462	0.16
48000	0H	21	47619	-0.79	0H	42	47619	-0.79
76800	0H	13	76923	0.16	0H	26	76923	0.16
115200	0H	9	111111	-3.55	0H	17	117647	2.1
153600	-	-	-	-	0H	13	153846	0.16
312500	-	-	-	-	0H	-	-	-
625000	-	-	-	-	0H	-	-	-

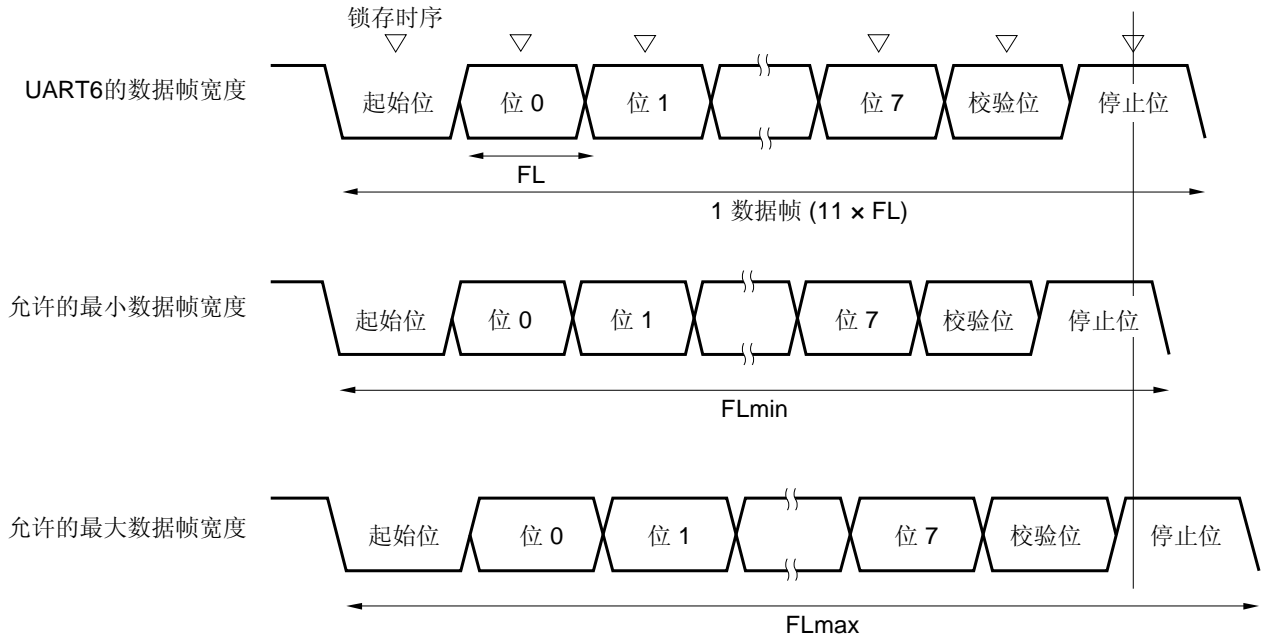
备注 TPS63 到 TPS60: 时钟选择寄存器 6 (CKSR6) 的第 3 位到第 0 位 (基本时钟 (f_{CLK6}) 的设置)
k: 使用波特率发生器控制寄存器 6 (BRGC6) 的 MDL67 到 MDL60 位设置的值 (k = 4, 5, 6, ..., 255)
f_{PRS}: 外围硬件时钟频率
ERR: 波特率误差

(4) 接收期间允许的波特率范围

接收期间来自发送端的波特率可允许的误差如下所示。

注意事项 必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。

图 10-20. 接收期间允许的波特率范围



如图 10-20 所示，检测到起始位后，接收数据的锁存时序将由波特率发生器控制寄存器 6 (BRGC6) 设置的计数器来确定。如果数据的最后一位（停止位）满足该锁存时序，则数据可被正确接收。

假定接收数据为 11 位，理论值可以按以下方式进行计算。

$$FL = (\text{Brate})^{-1}$$

Brate: UART6 的波特率

k: BRGC6 设置值

FL: 1 位数据宽度

锁存时序的极限: 2 个时钟

$$\text{可允许的最小数据帧宽度: } FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

因此，在发送端可接收的最大波特率如下所示。

$$BR_{\max} = (FL_{\min}/11)^{-1} = \frac{22k}{21k+2} \text{ Brate}$$

同样地，允许的最大数据帧宽度可按以下方式进行计算。

$$\frac{10}{11} \times FL_{\max} = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FL_{\max} = \frac{21k-2}{20k} FL \times 11$$

因此，在发送端可接收的最小波特率如下所示。

$$BR_{\min} = (FL_{\max}/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

可通过上述最小和最大波特率公式计算在 UART6 与发送端之间允许的波特率误差，如下所示。

表 10-6. 允许的最大 / 最小波特率误差

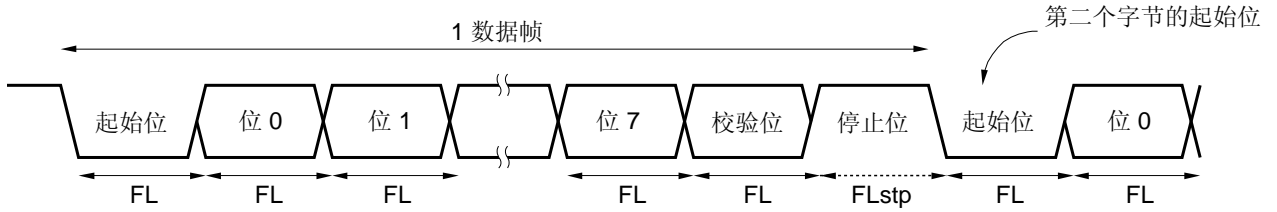
分频比 (k)	允许的最大波特率误差	允许的最小波特率误差
4	+2.33%	-2.44%
8	+3.53%	-3.61%
20	+4.26%	-4.31%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.73%

- 备注**
1. 接收时允许的误差与每帧的位数、输入时钟频率和分频比 (k) 有关。输入时钟频率和分频比 (k) 越高，允许的误差就越大。
 2. k: BRGC6 的设置值

(5) 连续发送期间数据帧的宽度

当连续发送数据时，从停止位到下一个起始位的数据帧宽度比正常值延长了两个基本时钟。然而，由于在检测到起始位时接收端的时序被初始化，因此通信结果不会受到影响。

图 10-21. 连续发送期间数据帧的宽度



当 1 位数据宽度为 FL，停止位宽度为 FLstp，基本时钟频率为 f_{XCLK6} 时，将满足以下表达式。

$$FLstp = FL + 2/f_{XCLK6}$$

因此，连续发送期间的数据帧宽度为：

$$\text{数据帧宽度} = 11 \times FL + 2/f_{XCLK6}$$

第十一章 中断功能

11.1 中断功能的类型

可以使用以下两种类型的中断功能。

(1) 可屏蔽中断

这些中断可进行屏蔽控制。通过设置优先级指定标志寄存器（PR0L， PR0H， PR1L）将可屏蔽中断分为高优先级中断组和低优先级中断组。

高优先级中断生成时中断嵌套服务可以应用到低优先级中断中。如果同时产生两个或两个以上具有相同优先级的中断请求，则将根据向量中断服务的优先级进行处理。关于优先级顺序的详细信息，请参见表 11-1。

产生一个待机释放信号，并释放 STOP 和 HALT 模式。

可屏蔽中断包括外部中断请求和内部中断请求。

- 外部： 8， 内部： 10

(2) 软件中断

这是通过执行 BRK 指令产生的向量中断。即使禁止中断时也可以响应这类中断。软件中断不受中断优先级控制。

11.2 中断源及配置

μ PD179F11x， 179F12x 微控制器产品共有 19 个中断源，包括可屏蔽的中断和软件中断。此外，还具有 4 种复位源（参见表 11-1）。

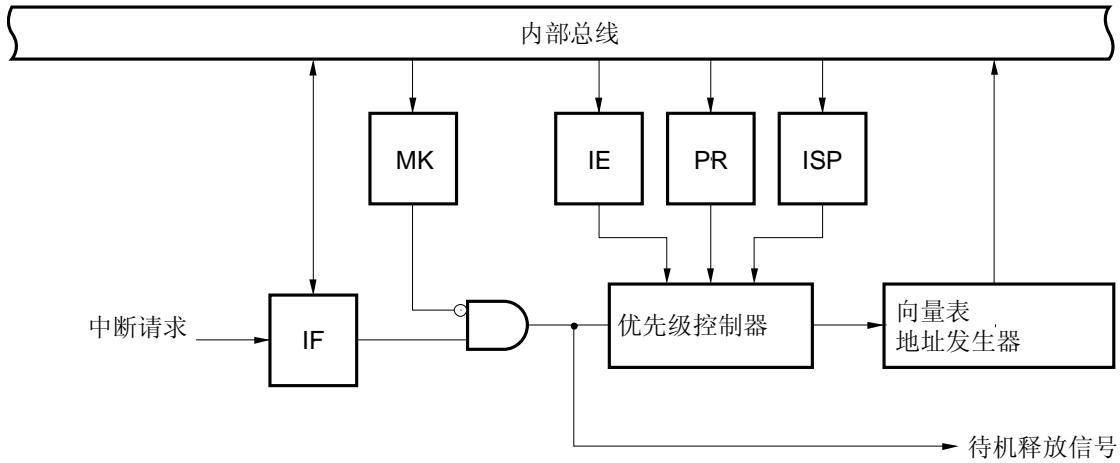
表 11-1. 中断源列表 (1/2)

中断类型	缺省优先级 ^{注1}	中断源		内部 / 外部	向量表地址	基本配置类型 ^{注2}
		名称	触发			
可屏蔽	0	INTLVI	低压检测 ^{注3}	内部	0004H	(A)
	1	INTP0	引脚输入边沿检测	外部	0006H	(B)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTSRE6	UART6 接收错误发生	内部	0012H	(A)
	8	INTSR6	UART6 接收结束		0014H	
	9	INTST6	UART6 发送结束		0016H	
	10	INTTMH1	TMH1 和 CMP01 间匹配 (当比较寄存器被指定时)		001AH	
	11	INTTMH0	TMH0 和 CMP00 间匹配 (当比较寄存器被指定时)		001CH	
	12	INTTM50	TM50 和 CR50 间匹配 (当比较寄存器被指定时)		001EH	
	13	INTTM000	TM00 和 CR000 间匹配 (当比较寄存器被指定时), TI010 引脚有效沿检测 (当捕获寄存器被指定时)		0020H	
	14	INTTM010	TM00 和 CR010 间匹配 (当比较寄存器被指定时), TI000 引脚有效沿检测 (当捕获寄存器被指定时)		0022H	
	15	INTTM51 ^{注4}	TM51 和 CR51 间匹配 (当比较寄存器被指定时)		002AH	
	16	INTKR0	按键中断检测		外部	
17	INTKR1	002EH				
软件	-	BRK	BRK 指令执行	-	003EH	(D)
复位	-	RESET	复位输入	-	0000H	-
		POC	上电清零			
		LVI	低压检测 ^{注5}			
		WDT	WDT 溢出			

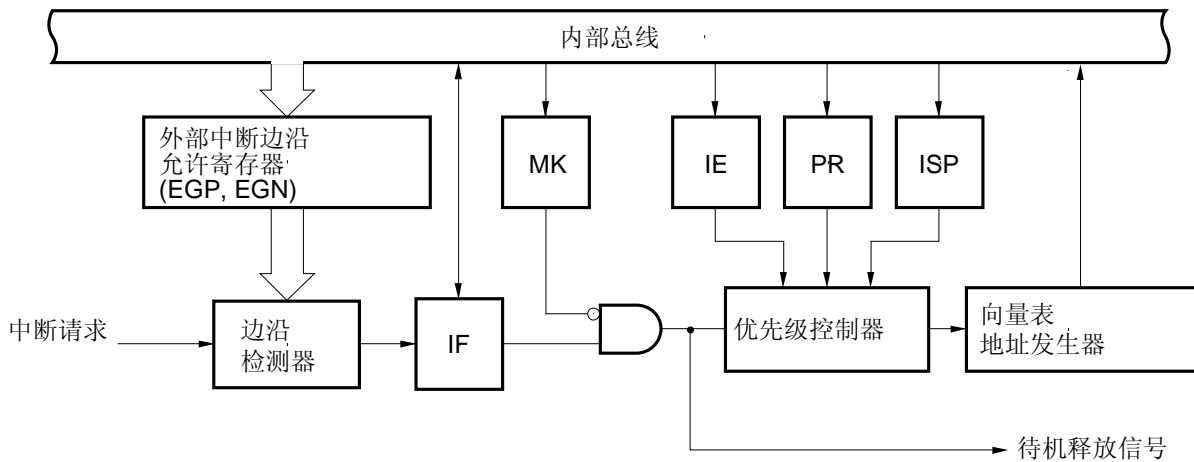
- 注
1. 如果两个或两个以上可屏蔽中断同时产生，那么采用默认的优先级来决定向量中断的处理顺序。零表示优先级最高，而 17 的优先级最低。
 2. 基本配置类型 (A) 到 (D) 与图 11-1 中 (A) 到 (D) 相对应。
 3. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 被清零时。
 4. 当 8 位定时器 / 事件计数器 51 用于载波发生器模式时，中断将会在 INTTM5H1 信号发生的时序中产生 (请参见图 8-16 载波发生器模式操作时序)。
 5. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 被设为 1 时。

图 11-1. 中断功能的基本配置 (1/2)

(A) 内部可屏蔽中断



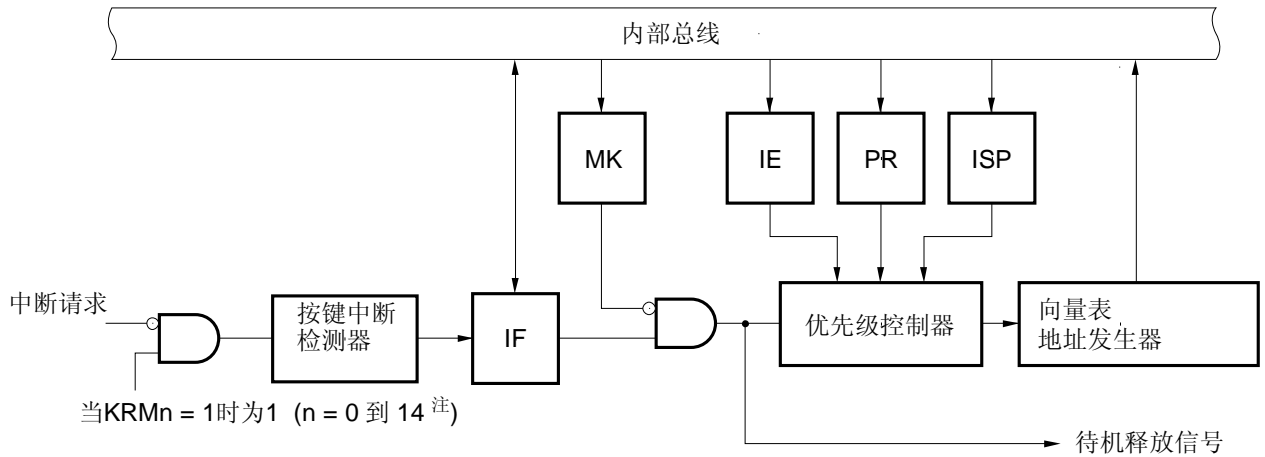
(B) 外部可屏蔽中断 (INTP0 到 INTP5)



- IF: 中断请求标志
- IE: 中断允许标志
- ISP: 当前服务优先级标志
- MK: 中断屏蔽标志
- PR: 优先级指定标志

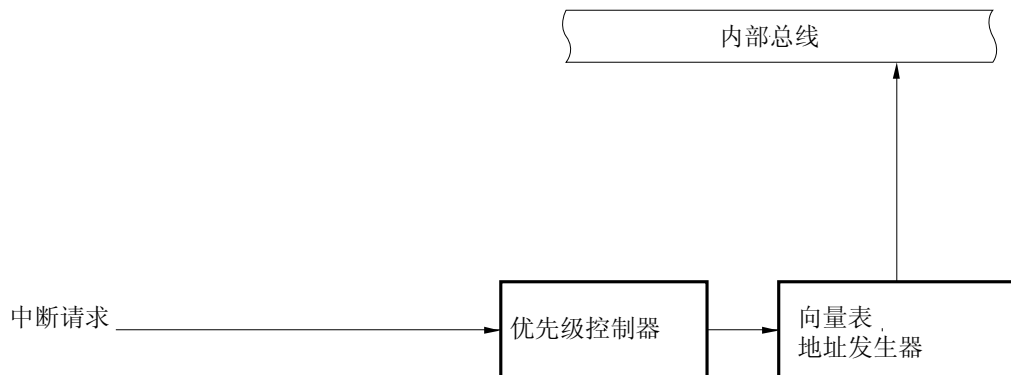
图 11-1. 中断功能的基本配置 (2/2)

(C) 外部可屏蔽中断 (INTKR)



注 KRM9 到 KRM14 仅在 38 引脚产品中提供。

(D) 软件中断



- IF: 中断请求标志
- IE: 中断允许标志
- ISP: 当前服务优先级标志
- MK: 中断屏蔽标志
- PR: 优先级指定标志
- KRM: 按键返回模式寄存器

11.3 控制中断功能的寄存器

以下 6 种类型的寄存器用于控制中断功能。

- 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 (PR0L, PR0H, 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)

表 11-2 列出了与中断请求源相对应的中断请求标志, 中断屏蔽标志和优先级指定标志。

表 11-2. 与中断请求源对应的标志

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	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTP5	PIF5		PMK5		PPR5	
INTSRE6	SREIF6		SREMK6		SREPR6	
INTSR6	SRIF6	IF0H	SRMK6	MK0H	SRPR6	PR0H
INTST6	STIF6		STMK6		STPR6	
INTTMH1	TMIFH1		TMMKH1		TMPRH1	
INTTMH0	TMIFH0		TMMKH0		TMPRH0	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM000	TMIF000		TMMK000		TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010	
INTTM51 ^注	TMIF51		TMMK51		TMPR51	
INTKR0	KRIF0	IF1L	KRMK0	MK1L	KRPR0	PR1L
INTKR1	KRIF1		KRMK1		KRPR1	

注 当 8 位定时器 / 事件计数器 51 用于载波发生器模式时，中断将会在 INTTM5H1 信号发生的时序中产生（请参见图 8-16 载波发生器模式操作时序）。

(1) 中断请求标志寄存器 (IF0L, IF0H, IF1L)

当相关的中断请求产生或指令执行时, 这些中断请求标志被置 1。如果在中断请求响应或复位信号生成时执行指令, 那么这些标志被清零。

当响应中断时, 中断请求标志自动清零, 然后进入中断服务程序。

IF0L, IF0H 和 IF1L 可以通过 1 位或 8 位存储器操作指令来进行设置。当 IF0L 和 IF0H 组合起来形成 16 位寄存器 IF0 时, 可由 16 位存储器操作指令来设置这些寄存器。

复位信号生成将这些寄存器设置为 00H。

图 11-2. 中断请求标志寄存器 (IF0L, IF0H, IF1L) 的格式

地址: FFE0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	SREIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIF

地址: FFE1H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
IF0H	TMIF010	TMIF000	TMIF50	TMIFH0	TMIFH1	0	STIF6	SRIF6

地址: FFE2H 复位后: 00H R/W

符号	7	6	<5>	<4>	<3>	2	1	0
IF1L	0	0	KRIF1	KRIF0	TMIF51	0	0	0

XXIFX	中断请求标志
0	无中断请求信号产生
1	中断请求产生, 中断请求状态

- 注意事项
1. 确保将 IF0H 的第 2 位, IF1L 的 0 位到第 2 位, 第 6 位, 第 7 位清零。
 2. 在待机模式释放后使用定时器时, 将中断请求标志清零后再进行操作。噪音可能会设置中断请求标志。
 3. 当对中断请求标志寄存器中的标志进行操作时, 使用 1 位存储器操作指令 (CLR1)。当用 C 语言描述时, 应使用一个诸如位 “IF0L.0 = 0;” 或 “_asm (“clr1 IF0L, 0”);” 的操作指令, 这是因为编译后的汇编指令必须是 1 位存储操作指令 (CLR1)。
如果使用诸如 “IF0L &= 0xfe;”, 的 8 位存储器操作指令的程序以 C 语言来描述并编译, 那么它将被转换为三条汇编指令。

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

这种情况下, 即使同一中断请求标志寄存器 (IF0L) 的其他位的请求标志在处于 “mov a, IF0L” 和 “mov IF0L, a” 之间的时序时被设置为 1, 该请求标志也将被 “mov IF0L, a” 清零。因此, 在 C 语言中使用 8 位存储器操作指令时必须小心。

(2) 中断屏蔽标志寄存器 (MK0L, MK0H, MK1L)

这些中断屏蔽标志用于允许 / 禁止相关的可屏蔽中断服务。

MK0L, MK0H 和 MK1L 可以通过 1 位或 8 位存储器操作指令来进行设置。当 MK0L 和 MK0H 组合起来形成 16 位寄存器 MK0 时, 可由 16 位存储器操作指令来设置这些寄存器。

复位信号生成将这些寄存器设置为 FFH。

图 11-3. 中断屏蔽标志寄存器 (MK0L, MK0H, MK1L) 的格式

地址: FFE4H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	SREMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

地址: FFE5H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
MK0H	TMMK010	TMMK000	TMMK50	TMMKH0	TMMKH1	1	STMK6	SRMK6

地址: FFE6H 复位后: FFH R/W

符号	7	6	<5>	<4>	<3>	2	1	0
MK1L	1	1	KRMK1	KRMK0	TMMK51	1	1	1

XXMKX	中断服务控制						
0	允许中断服务						
1	禁止中断服务						

注意事项 确保将 MK0H 的第 2 位, MK1L 的 0 位到第 2 位, 第 6 位, 第 7 位设为 1。

(3) 优先级指定标志寄存器 (PR0L, PR0H, PR1L)

这些优先级指定标志寄存器用于设置相关的可屏蔽中断优先级次序。

PR0L, PR0H 和 PR1L 可以通过 1 位或 8 位存储器操作指令来进行设置。如果 PR0L 和 PR0H 组合起来形成 16 位寄存器 PR0, 则可由 16 位存储器操作指令来设置这些寄存器。

复位信号生成将这些寄存器设置为 FFH。

图 11-4. 优先级指定标志寄存器 (PR0L, PR0H, PR1L) 的格式

地址: FFE8H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0L	SREPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	LVIPR

地址: FFE9H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
PR0H	TMPR010	TMPR000	TMPR50	TMPRH0	TMPRH1	1	STPR6	SRPR6

地址: FFEAH 复位后: FFH R/W

符号	7	6	<5>	<4>	<3>	2	1	0
PR1L	1	1	KRPR1	KRPR0	TMPR51	1	1	1

XXPRX	优先级选择
0	高优先级
1	低优先级

注意事项 确保将 PR0H 的第 2 位, PR1L 的 0 位到第 2 位, 第 6 位, 第 7 位设为 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)

这两个寄存器用于指定 INTP0 到 INTP5 的有效沿。

EGP 和 EGN 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将这些寄存器设置为 00H。

图 11-5. 外部中断上升沿允许寄存器 (EGP) 的格式
和外部中断下降沿允许寄存器 (EGN) 的格式

地址: FF48H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGP	0	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

地址: FF49H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGN	0	0	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn 引脚有效沿选择 (n = 0 到 5)
0	0	禁止边沿检测
0	1	下降沿
1	0	上升沿
1	1	兼有上升沿和下降沿

注意事项 一定要将 EGP 和 EGN 的位 6 和位 7 清零。

表 11-3 显示了与 EGPn 和 EGNn 相关的端口。

表 11-3. 与 EGPn 和 EGNn 相关的端口。

检测允许寄存器		边沿检测端口	中断请求信号
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P06	INTP1
EGP2	EGN2	P05	INTP2
EGP3	EGN3	P04	INTP3
EGP4	EGN4	P27	INTP4
EGP5	EGN5	RESET/P123	INTP5

注意事项 在从外部中断功能切换到端口功能时可能会检测到边沿，因此通过将 EGPn 和 EGNn 清零可选择端口模式。

备注 n = 0 到 5

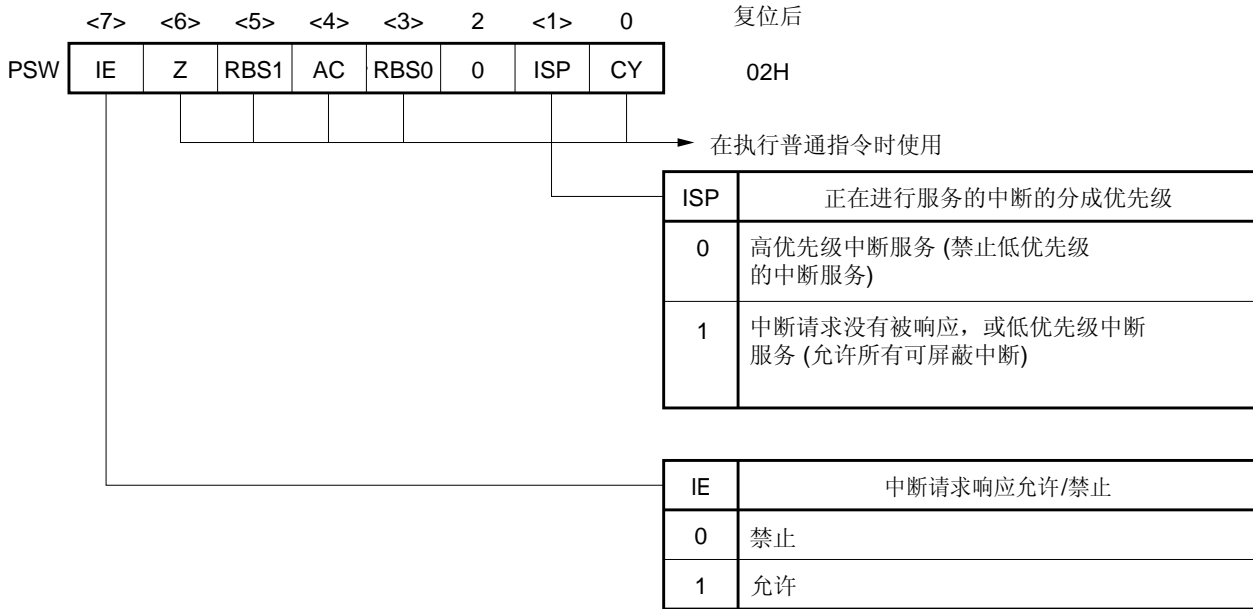
(5) 程序状态字 (PSW)

程序状态字是用于保存指令执行结果和中断请求的当前状态的寄存器。在 PSW 中包含用于设置允许 / 禁止可屏蔽中断的 IE 标志以及用于控制中断嵌套服务的 ISP 标志。

除了 8 位读 / 写操作指令外, 还可使用位操作指令和专用指令 (EI 和 DI) 对该寄存器进行操作。在响应向量中断请求时, 如果执行 BRK 指令, 则将 PSW 的内容自动保存到堆栈中, 并且将 IE 标志复位为 0。如果响应了可屏蔽中断请求, 则将被响应中断的优先级指定标志的内容转移到 ISP 标志中。PSW 的内容也可以通过执行 PUSH PSW 指令来保存到堆栈中。这些内容可以通过执行 RETI, RETB 和 POP PSW 指令从堆栈中恢复。

复位信号生成将 PSW 设为 02H。

图 11-6. 程序状态字的格式



11.4 中断服务操作

11.4.1 可屏蔽中断响应

当中断请求标志被设为 1 且与该中断请求相关的屏蔽标志 (MK) 被清零时, 可以响应这个可屏蔽中断请求。如果中断处于中断允许状态 (IE 标志被设为 1 时), 可以响应向量中断请求。然而, 在一个较高优先级中断请求服务期间 (ISP 标志复位为 0 时), 不会响应低优先级的中断请求。

从一个可屏蔽中断请求产生到向量中断服务执行所经历的时间列于表 11-4 中。

关于中断请求响应时序的详细信息, 请参见图 11-8 和 11-9。

表 11-4. 从可屏蔽中断产生到执行中断服务所需要的时间

	最短时间	最长时间 ^注
当 $\times\times PR = 0$ 时	7 个时钟	32 个时钟
当 $\times\times PR = 1$ 时	8 个时钟	33 个时钟

注 如果是在除法指令执行之前产生一个中断请求, 则等待时间会变长。

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

如果同时产生两个或两个以上的可屏蔽中断请求, 则首先响应优先级指定标志中优先级别高的请求。如果两个或两个以上的中断请求具有相同的优先级别, 则首先响应具有最高默认优先级的中断请求。

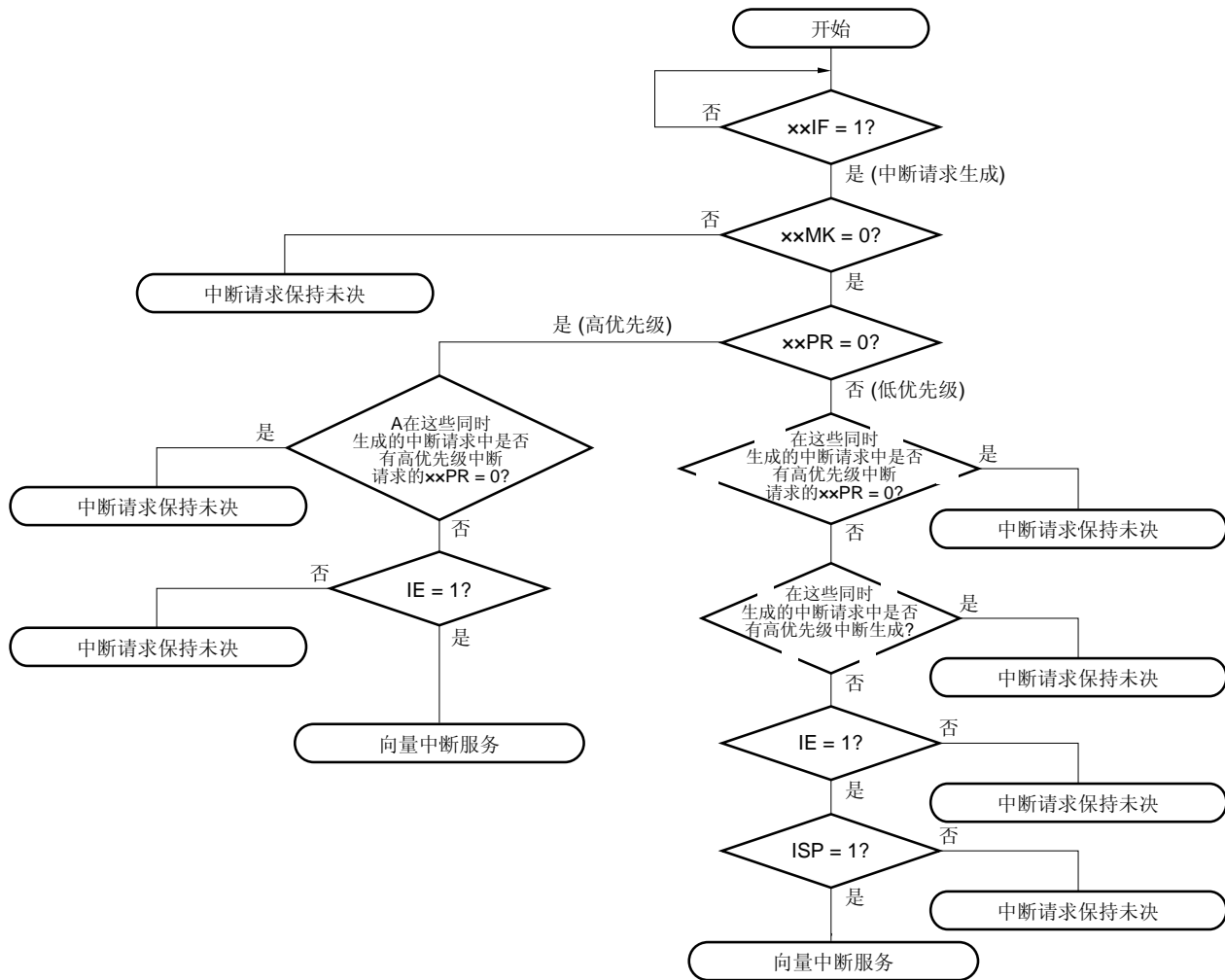
当允许响应中断请求时, 处于等待状态的中断请求被响应。

图 11-7 显示了中断请求响应算法。

如果响应了一个可屏蔽中断请求, 则将 PSW 的内容依次保存到堆栈中, 然后将 PC 的内容依次保存到堆栈中, 而 IE 标志将被复位为 0, 并且将与被响应的中断相关的优先级指定标志的内容传送到 ISP 中。将用于中断请求的向量表数据装载到 PC 中, 并转移。

通过使用 RETI 指令可以实现从中断中返回。

图 11-7. 中断请求响应处理算法



xxIF: 中断请求标志

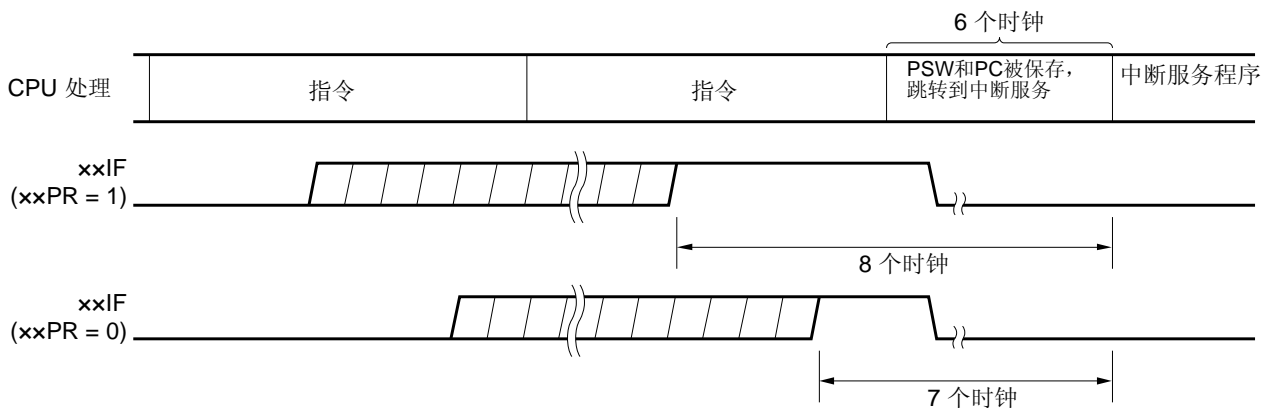
xxMK: 中断屏蔽标志

xxPR: 优先级指定标志

IE: 控制可屏蔽中断请求响应的标志 (1 = 允许, 0 = 禁止)

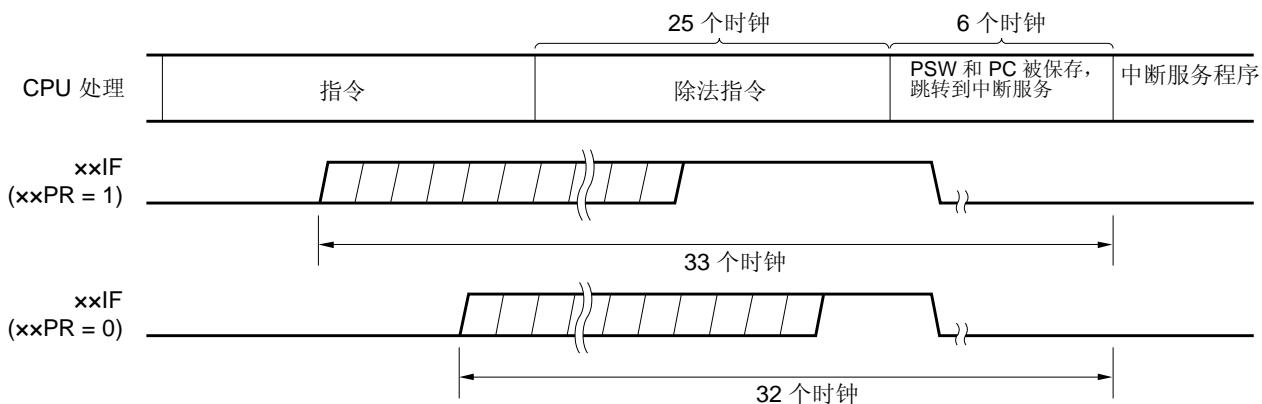
ISP: 指示正在服务的中断优先级级别的标志 (0 = 高优先级的中断服务, 1 = 没有中断请求被响应, 或低优先级的中断服务)

图 11-8. 中断请求响应时序（最短时间）



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

图 11-9. 中断请求响应时序（最长时间）



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

11.4.2 软件中断请求响应

通过执行 BRK 指令可响应软件中断请求。软件中断不能被禁止。

如果响应了一个软件中断请求，则将程序状态字（PSW）和程序计数器（PC）的内容依次保存到堆栈中，然后将 IE 标志复位为 0，并将向量表的内容（003EH，003FH）装载到 PC 中，然后转移。

通过使用 RETB 指令可以实现从软件中断中返回。

注意事项 不要使用 RETI 指令来从软件中断返回。

11.4.3 中断嵌套服务

在中断执行过程中又响应了其他中断请求时会产生中断嵌套服务。

除非选择中断请求响应允许状态 (IE = 1)，否则不会产生中断嵌套服务。在响应一个中断请求时，禁止响应其它中断请求 (IE = 0)。因此，如果要允许中断嵌套服务，必须在中断服务期间通过 EI 指令将 IE 标志设为 1，从而允许响应其它中断请求。

此外，即使允许中断，也不一定允许中断嵌套服务，这是因为受到中断优先级控制的限制。可使用两类优先级控制方式：默认优先级控制和可编程优先级控制。可编程优先级控制用于中断嵌套服务中。

在中断允许状态中，如果产生的中断请求的优先级与正在服务的中断优先级相等或高于它，则响应该中断请求，从而产生中断嵌套服务。如果中断服务期间产生的中断请求的优先级低于正在服务的中断优先级，则不会响应该中断请求而产生中断嵌套服务。由于中断处于中断禁止状态或中断请求的优先级较低而不被允许的中断请求会处于等待状态。若当前的中断服务已结束，则在执行至少一条主程序指令后才可响应处于等待状态的中断请求。

表 11-5 显示了允许中断嵌套服务的中断请求之间的关系，图 11-10 为中断嵌套服务的示例。

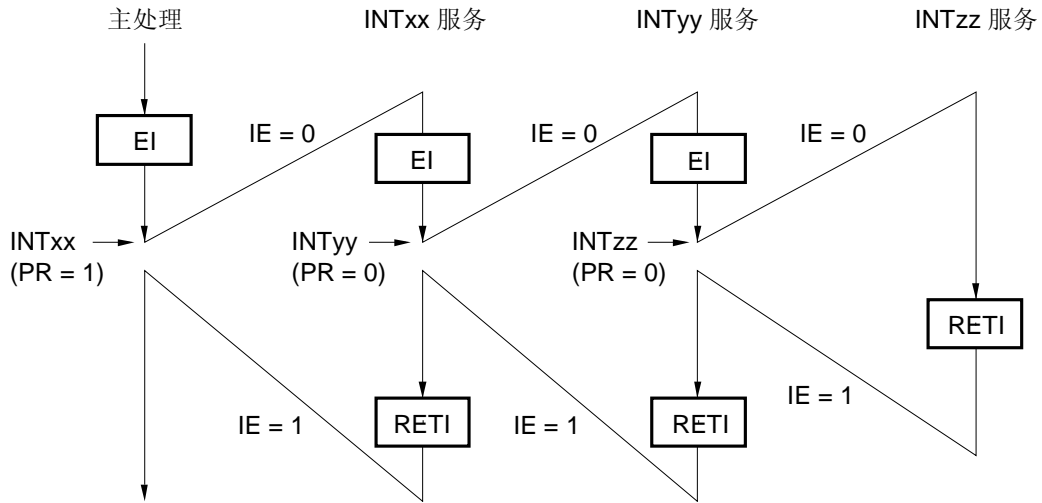
表 11-5. 中断服务期间允许中断嵌套服务的中断请求之间的关系

多得多中断请求		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	
正在服务的中断						
可屏蔽中断	ISP = 0	○	×	×	×	○
	ISP = 1	○	×	○	×	○
软件中断		○	×	○	×	○

- 备注**
- : 允许中断嵌套服务
 - ×: 禁止中断嵌套服务
 - ISP 和 IE 是包含在 PSW 中的标志。
 ISP = 0: 正在服务一个较高优先级的中断。
 ISP = 1: 无中断请求被响应，或正在服务一个较低优先级的中断。
 IE = 0: 禁止响应中断请求。
 IE = 1: 允许响应中断请求。
 - PR 是包含在 PR0L, PR0H 和 PR1L 中的标志。
 PR = 0: 高优先级
 PR = 1: 低优先级

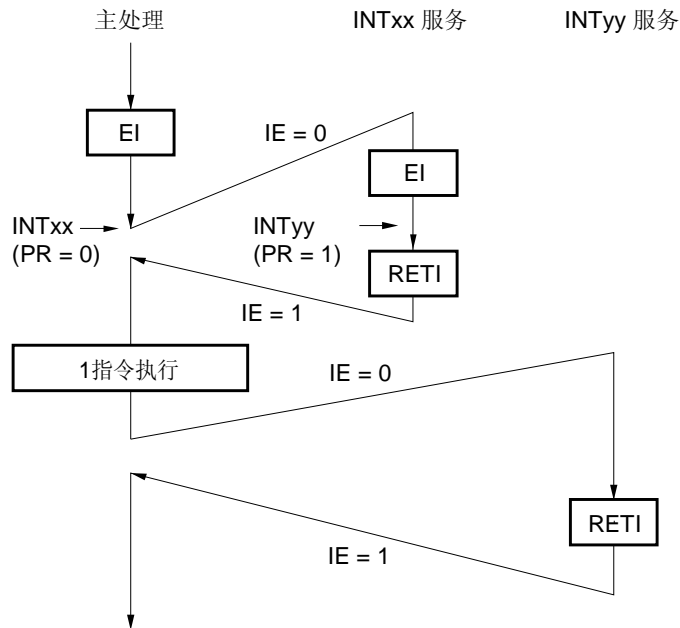
图 11-10. 中断嵌套服务示例 (1/2)

例 1. 发生两次中断嵌套服务



在 INTxx 中断服务期间，响应了两个中断请求，INTyy 和 INTzz，这时就会发生中断嵌套服务。在响应每个中断之前，必须先执行 EI 指令才能允许响应中断请求。

例 2. 由于优先级控制而没有发生中断嵌套服务

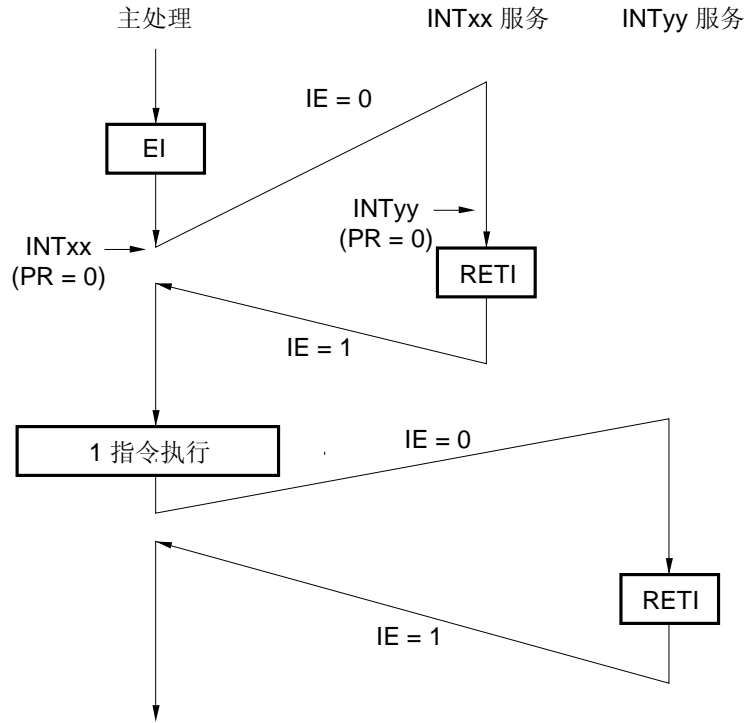


在 INTxx 中断服务期间发出的中断请求 INTyy 没有被响应，这是因为 INTyy 的优先级低于 INTxx 的优先，这样也就不会发生中断嵌套。中断请求 INTyy 处于等待状态，在执行一条主程序指令后会响应该中断请求。

- PR = 0: 高优先级
- PR = 1: 低优先级
- IE = 0: 禁止响应中断请求

图 11-10. 中断嵌套服务示例 (2/2)

例 3. 由于中断不允许而没有产生中断嵌套服务



中断在 INT_{xx} 中断服务期间将不会被允许（没有发出 EI 指令），因此不会响应中断请求 INT_{yy} ，这样也不会产生中断嵌套。中断请求 INT_{yy} 处于等待状态，在执行一条主程序指令后会响应该中断请求。

$PR = 0$: 高优先级

$IE = 0$: 禁止响应中断请求

11.4.4 中断请求保持

存在一些指令，即使在另一个指令执行时为这些指令发出中断请求，这些指令中的中断请求也会保持为等待状态直到下个指令执行结束。以下列出了这些指令（中断请求保持指令）。

- 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
- 用于 IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H 和 PR1L 寄存器的操作指令。

注意事项 **BRK** 指令不属于上述列出的中断请求保持指令。然而，通过执行 **BRK** 指令激活的软件中断会将 **IE** 标志清零。因此，即使在执行 **BRK** 指令期间产生了可屏蔽中断请求，该中断请求也不会被响应。

图 11-11 显示了中断请求处于等待状态的时序。

图 11-11. 中断请求保持



- 备注**
1. 指令 N: 中断请求保持指令
 2. 指令 M: 除中断请求保持指令之外的指令
 3. xxPR（优先级别）的值不会影响xxIF（中断请求）的操作。

第十二章 按键中断功能

12.1 按键中断的功能

可通过设置按键返回模式寄存器 0, 1 (KRML 和 KRMH) 以及向按键中断输入引脚^注输入一个下降沿来生成一个按键中断 (INTKR0 和 INTKR1)。

为 KR8 使用 $\overline{\text{RESET}}/\text{P123}/\text{INTP5}/\text{KR8}$ 引脚时, 应将复位引脚模式寄存器 (RSTMASK) 的第 3 位 (RSTM) 设置为“1”。

注 30 引脚产品: KR0 到 KR8
38 引脚产品: KR0 到 KR14

表 12-1. 按键中断检测引脚的分配

标志	说明
KRMLm	以 1 位为单位控制 KRm 信号
KRMHm	以 1 位为单位控制 KRn 信号

备注 m = 0 到 7, n = 8 到 14

12.2 按键中断的配置

按键中断包含以下硬件。

表 12-2. 按键中断的配置

项目	配置
控制寄存器	按键返回模式寄存器 0 (KRML) 按键返回模式寄存器 1 (KRMH)

图 12-1. 按键中断的框图 (1/2)

(a) 按键中断 (INTKR0)

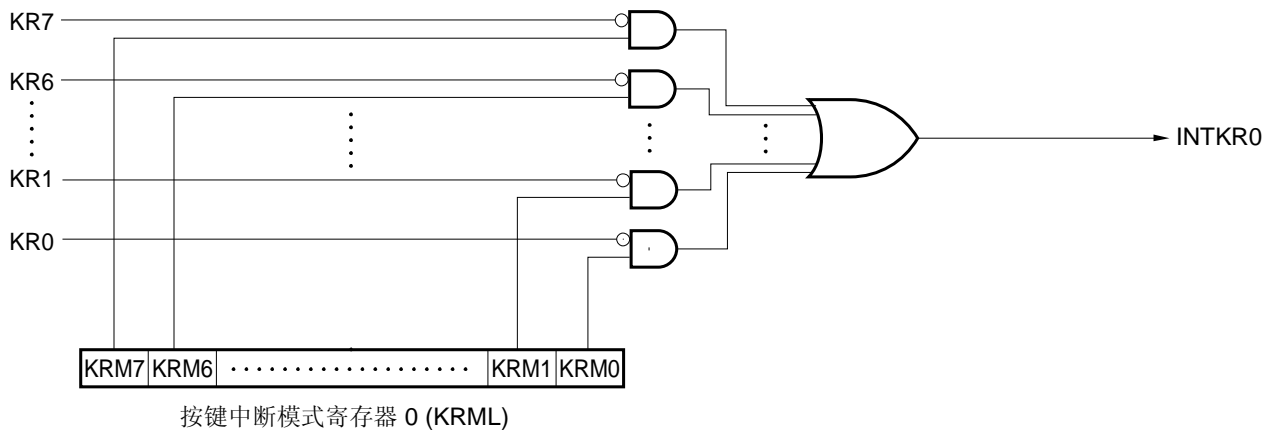
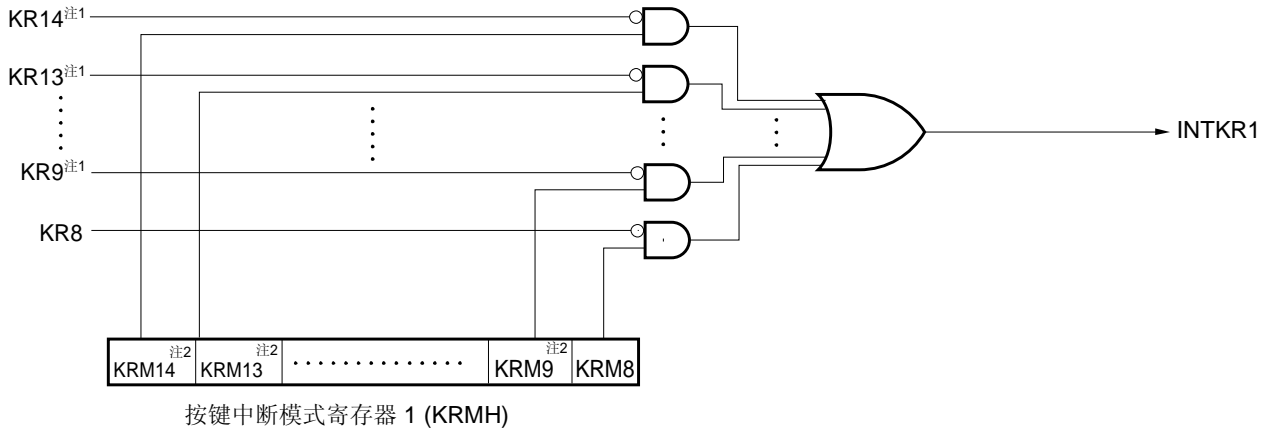


图 12-1. 按键中断的框图 (2/2)

(b) 按键中断 (INTKR1)



- 注 1. 仅限 38 引脚产品
- 2. 在 30 引脚产品中这些位固定为“0”。

12.3 控制按键中断的寄存器

(1) 按键返回模式寄存器 0 (KRML)

该寄存器分别使用 KR0 至 KR7 信号来控制 KRM0 至 KRM7 位。
 KRML 可以通过 1 位或 8 位存储器操作指令来进行设置。
 复位信号生成将该 KRML 设置为 00H。

图 12-2. 按键返回模式寄存器 0 (KRML) 的格式

地址: FF6EH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
KRML	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

PE6	指示校验错误的状态标志
0	不能检测按键中断信号
1	检测按键中断信号

- 注意事项**
1. 如果 KRM0 到 KRM7 位中的任意位被设为 1，则应将上拉电阻寄存器 1 (PU1) 中相应的 0 到 7 位设为 1。
 2. 如果 KRML 发生变化，则中断请求标志可能被设置。因此，先禁止中断然后再修改 KRML 寄存器。清除中断请求标志并允许中断。
 3. 在按键中断模式中没有被使用的位可用作普通端口。

(2) 按键返回模式寄存器 1 (KRMH)

该寄存器分别通过使用 KR8 以及 KR9 到 KR14[≠] 信号来控制 KRM8 以及 KRM9 到 KRM14[≠] 位。

KRMH 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该 KRMH 设置为 00H。

图 12-3. 按键返回模式寄存器 1 (KRMH) 的格式

地址: FF6FH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
KRMH	0	KRM14 [≠]	KRM13 [≠]	KRM12 [≠]	KRM11 [≠]	KRM10 [≠]	KRM9 [≠]	KRM8
PE6	指示校验错误的状态标志							
0	不能检测按键中断信号							
1	检测按键中断信号							

注 仅限 38 引脚产品

- 注意事项
1. 如果 KRM8 到 KRM14 位中的任何位被设置为 1，则应将上拉电阻寄存器 3 (PU3) 中相应的 0 到 5 位为 1 以及相应的上拉电阻寄存器 12 (PU12) 的第 3 位设为 1。
 2. 如果 KRMH 发生变化，则中断请求标志可能被设置。因此，先禁止中断然后再修改 KRMH 寄存器。清除中断请求标志并允许中断。
 3. 在按键中断模式中没有被使用的位可用作正常端口。
 4. 对于 30 引脚产品，确保将 KRMH 和 PM3 的 0 到 6 位设为“0”。

第十三章 待机功能

13.1 待机功能及配置

13.1.1 待机功能

待机功能用于减少系统的工作电流。以下两种模式可用。

(1) HALT 模式

通过执行 HALT 指令设置 HALT 模式。在 HALT 模式中，CPU 操作时钟停止。如果设置 HALT 模式前，高速系统时钟振荡器，内部高速振荡器或内部低速振荡器正在使用，那么每种时钟的振荡将会继续。在此模式中，工作电流不如 STOP 模式中下降得多，但 HALT 模式对于中断请求产生后立即重启操作和频繁进行间断的操作非常有效。

(2) STOP 模式

通过执行 STOP 指令设置 STOP 模式。在 STOP 模式中，高速系统时钟振荡器和内部高速振荡器停止操作，整个系统的操作也停止，这样 CPU 的工作电流将会大幅下降。

由于可以通过中断请求释放该模式，因此它允许被中断的操作继续执行。然而，由于选择 X1 时钟时，在释放 STOP 模式后需要一段等待时间以确保振荡器振荡稳定，因此如果需要在产生中断请求后立即进行处理，则应选择 HALT 模式。

在这两种模式中，寄存器，标志和数据存储器中的所有内容将会保持进入待机模式前的内容。I/O 端口输出锁存器和输出缓冲器的状态也将被保持。

<R> **注意事项** **当切换到 STOP 模式时，在执行 STOP 指令前必须停止使用带有主系统时钟的外围硬件的操作。**

13.1.2 控制待机功能的寄存器

待机功能由以下两个寄存器控制。

- 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)

备注 关于启动，停止或选择时钟的寄存器的详细信息，请参见第五章 时钟发生器。

(1) 振荡稳定时间计数状态寄存器 (OSTC)

该寄存器用于表示 X1 时钟振荡稳定时间计数器的计数状态。当 X1 时钟振荡用作 CPU 时钟的内部高速振荡时钟开始工作时，将可以检测 X1 时钟振荡稳定时间。

OSTC 可以通过 1 位或 8 位存储器操作指令来读取。

复位释放时（通过 $\overline{\text{RESET}}$ 输入，POC，LVI，以及 WDT 进行复位），STOP 指令以及 MSTOP（MOC 寄存器的第 7 位）= 1，可以将 OSTC 清除为 00H。

<R>

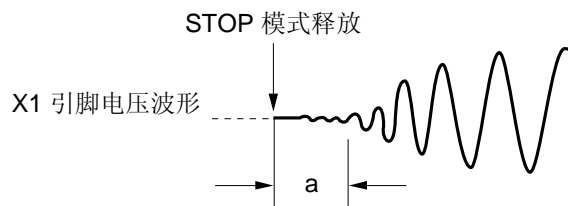
图 13-1. 振荡稳定时间计数状态寄存器 (OSTC) 的格式

地址： FFA3H 复位后： 00H R

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

MOST11	MOST13	MOST14	MOST15	MOST16	振荡稳定时间状态			
					$f_x = 1 \text{ MHz}$	$f_x = 2 \text{ MHz}$	$f_x = 4 \text{ MHz}$	
0	0	0	0	0	少于 $2^{11}/f_x$	少于 2.04 ms	少于 1.02 ms	少于 510 μs
1	0	0	0	0	$2^{11}/f_x \text{ min.}$	2.04 ms min	1.02 ms min	510 μs min
1	1	0	0	0	$2^{13}/f_x \text{ min.}$	8.20 ms min	4.10 ms min	2.04 ms min
1	1	1	0	0	$2^{14}/f_x \text{ min.}$	16.38 ms min.	8.19 ms min.	4.10 ms min
1	1	1	1	0	$2^{15}/f_x \text{ min.}$	32.77 ms min.	16.38 ms min.	8.19 ms min.
1	1	1	1	1	$2^{16}/f_x \text{ min.}$	65.45 ms min.	32.77 ms min.	16.38 ms min.

- 注意事项**
1. 经历上述时间后，从 MOST11 开始按顺序设置为 1 并保持为 1。
 2. 振荡稳定时间计数器用于累计通过 OSTC 所设置的振荡稳定时间。如果在内部高速振荡时钟被用作 CPU 时钟时，先进入 STOP 模式，然后释放，那么将按如下对振荡稳定时间进行设置。
 - OSTC 振荡稳定期望时间 \leq 由 OSTC 设定的振荡稳定时间
 因此，需要注意在 STOP 模式被释放后，只有达到 OSTC 设定的振荡稳定时间时的状态会被赋予 OSTC。
 3. X1 时钟振荡稳定等待时间只有在时钟振荡开始后才会包含时间（以下“a”）。



备注 f_x : X1 时钟振动频率

(2) 振荡稳定时间选择寄存器 (OSTS)

该寄存器用于在 STOP 模式被释放时选择 X1 时钟振荡稳定等待时间。

当 X1 时钟被选作 CPU 时钟时，在 STOP 模式被释放后操作将会通过使用 OSTS 来等待一段时间。

当内部高速振荡时钟被选作 CPU 时钟时，在 STOP 模式被释放后应对 OSTC 进行确认，确认期望的振荡稳定时间已经过去。振荡稳定时间可以使用 OSTC 与时间集进行核对。

OSTS 可以通过 8 位存储器操作指令来进行设置。

复位信号生成将 OSTS 设为 05H。

<R>

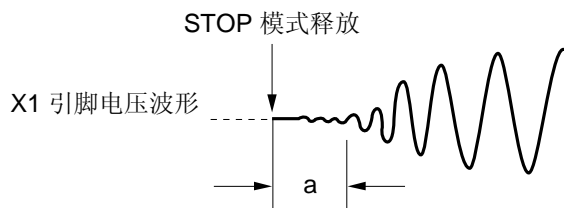
图 13-2. 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFA4H 复位后: 05H R/W

符号	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	振荡稳定时间选择			
			$f_x = 1 \text{ MHz}$	$f_x = 2 \text{ MHz}$	$f_x = 4 \text{ MHz}$	
0	0	1	$2^{11}/f_x$	2.04 ms	1.02 ms	510 μs
0	1	0	$2^{13}/f_x$	8.19 ms	4.10 ms	2.04 ms
0	1	1	$2^{14}/f_x$	16.38 ms	8.19 ms	4.10 ms
1	0	0	$2^{15}/f_x$	32.77 ms	16.38 ms	8.19 ms
1	0	1	$2^{16}/f_x$	65.45 ms	32.77 ms	16.38 ms
其他情况			禁止设置			

- 注意事项**
1. 若要在 X1 时钟被用作 CPU 时钟时设置 STOP 模式，则应在执行 STOP 指令前对 OSTS 进行设置。
 2. 在 X1 时钟振荡稳定时间内不要更改 OSTS 寄存器的值。
 3. 振荡稳定时间计数器用于累计通过 OSTS 所设置的振荡稳定时间。如果在内部高速振荡时钟被用作 CPU 时钟时先进入 STOP 模式，然后释放，那么将按如下对振荡稳定时间进行设置。
 - OSTC 振荡稳定期望时间 \leq 由 OSTS 设定的振荡稳定时间
 因此，需要注意在 STOP 模式被释放后，只有达到 OSTS 设定的振荡稳定时间时的状态会被赋予 OSTC。
 4. X1 时钟振荡稳定等待时间只有在时钟振荡开始后才会包含时间（以下“a”）。



备注 f_x : X1 时钟振动频率

13.2 待机功能的操作

13.2.1 HALT模式

(1) HALT 模式

通过执行 HALT 指令可以设置 HALT 模式。无论设置前 CPU 时钟是高速系统时钟或内部高速振荡时钟，都可以对 HALT 模式进行设置。

HALT 模式中的操作状态如下所示。

表 13-1. HALT 模式中的操作状态

HALT 模式设置		当 CPU 使用主系统时钟并执行 HALT 指令时		
		当 CPU 使用内部高速振荡时钟 (f_{RH}) 时	当 CPU 使用 X1 时钟 (f_x) 时	当 CPU 使用外部主系统时钟 (f_{EXCLK}) 时
项目				
系统时钟		提供给 CPU 的时钟被停止		
主系统时钟	f_{RH}	操作继续 (不能停止)	继续设置 HALT 模式前的状态	
	f_x	保持设置 HALT 模式前的状态	操作继续 (不能停止)	保持设置 HALT 模式前的状态
	f_{EXCLK}	由外部时钟输入进行操作或停止		操作继续 (不能停止)
f_{RL}		继续设置 HALT 模式前的状态		
CPU		操作停止		
Flash 存储器				
RAM		保持设置 HALT 模式前的状态		
端口 (锁存)				
16 位定时器 / 事件计数器 00		可操作		
8 位定时器 / 事件计数器	50			
	51			
8 位定时器	H0			
	H1			
看门狗定时器		可操作。当通过选项字节设置“可以由软件停止内部低速振荡器”时，停止提供给看门狗定时器的时钟。		
串行接口	UART6	可操作		
上电清零功能				
低电压检测功能				
外部中断				

备注

- f_{RH} : 内部高速振荡时钟
- f_x : X1 时钟
- f_{EXCLK} : 外部主系统时钟
- f_{RL} : 内部低速振荡时钟

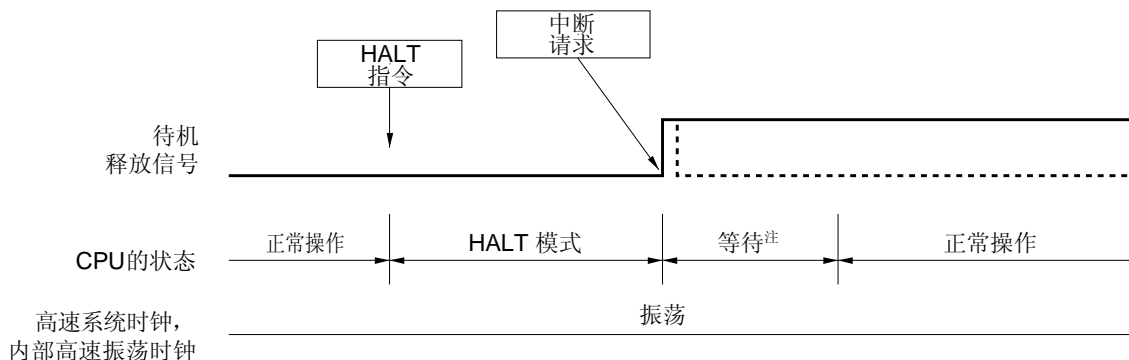
(2) HALT 模式释放

HALT 模式可以通过以下两种源来释放。

(a) 通过没有被屏蔽的中断请求释放

当产生一个没有被屏蔽的中断请求时，HALT 模式将会被释放。如果允许中断响应，则将执行向量中断服务程序。如果禁止中断响应，则将执行下一个地址指令。

图 13-3. 通过产生中断请求释放 HALT 模式



注 等待时间如下所示：

- 执行向量中断服务程序时：8 或 9 个时钟
- 不执行向量中断服务程序时：2 或 3 个时钟

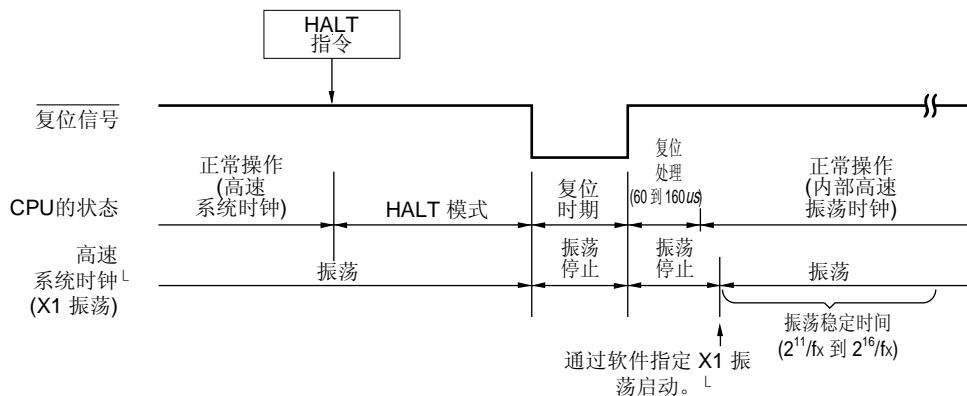
备注 虚线表示释放待机模式的中断请求被响应时的情况。

(b) 通过复位信号生成释放

当复位信号生成时，HALT 模式被释放，然后，在进行正常复位操作时，程序会在跳转到复位向量地址后执行。

图 13-4. 通过复位释放 HALT 模式 (1/2)

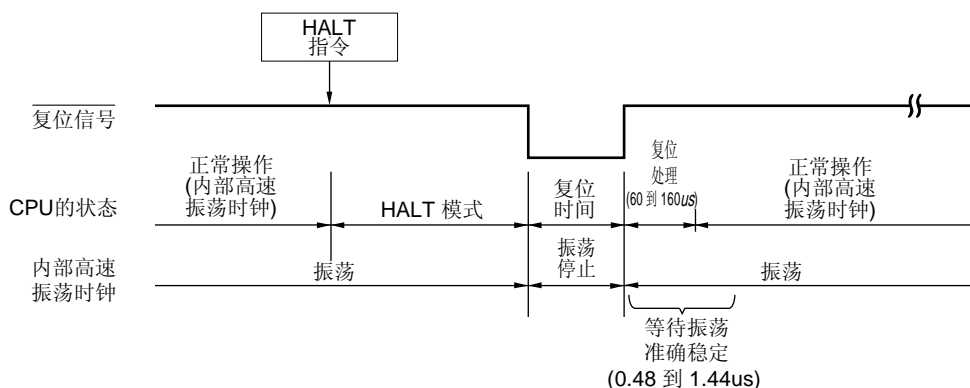
(1) 高速系统时钟被用作 CPU 时钟时



备注 fx: X1 时钟振动频率

图 13-4. 通过复位释放 HALT 模式 (2/2)

(2) 内部高速振荡时钟被用作 CPU 时钟时



备注 fx: X1 时钟振荡频率

表 13-2. 在 HALT 模式中响应中断请求的操作

释放源	MKxx	PRxx	IE	ISP	操作
可屏蔽中断请求	0	0	0	×	执行下一个地址指令
	0	0	1	×	执行中断服务程序
	0	1	0	1	执行下一个地址指令
	0	1	×	0	
	0	1	1	1	执行中断服务程序
	1	×	×	×	保持 HALT 模式
复位	-	-	×	×	复位处理

×: 不考虑

13.2.2 STOP 模式

(1) STOP 模式设置及操作状态

通过执行 STOP 指令来设置 STOP 模式，并且它只有在 CPU 时钟使用主系统时钟前设置。

注意事项 由于中断请求信号用于释放待机模式，因此如果一个带有中断请求标志的中断源被设置且中断屏蔽标志被复位，则立即释放待机模式。因此，在执行 STOP 指令后，STOP 模式将立即复位到 HALT 模式，并且在经历了振荡稳定时间选择寄存器（OSTS）所设置的等待时间后，系统将返回操作模式。

STOP 模式中的操作状态如下所示。

表 13-3. STOP 模式中的操作状态

STOP 模式设置		当 CPU 使用主系统时钟并执行 STOP 指令时		
		当 CPU 使用内部高速振荡时钟 (f_{RH}) 时	当 CPU 使用 X1 时钟 (f_x) 时	当 CPU 使用外部主系统时钟 (f_{EXCLK}) 时
项目				
系统时钟		提供给 CPU 的时钟被停止		
主系统时钟	f_{RH}	停止		
	f_x			
	f_{EXCLK}	输入无效		
	f_{RL}	继续设置 STOP 模式前的状态		
CPU		操作停止		
Flash 存储器				
RAM		保持设置 STOP 模式前的状态		
端口 (锁存)				
16-位定时器 / 事件计数器 00 ^{注1}		操作停止		
8 位定时器 / 事件计数器	50 ^{注1}	仅在选择 TI50 作为计数时钟时可操作		
	51 ^{注1}	当 8 位定时器 H1 在载波发生器模式中, 仅在选择 TI51 或 TMH1 载波时钟作为计数时钟时可操作		
8 位定时器	H0	在 8 位定时器 / 事件计数器 50 操作过程中, 仅在选择 TM50 输出作为计数时钟时可操作		
	H1	仅在选择 f_{RL} 或 $f_{RL}/2^7$ 作为计数时钟时可操作		
看门狗定时器		可操作。当通过选项字节设置“可以由软件停止内部低速振荡器”时, 停止提供给看门狗定时器的时钟。		
串行接口	UART6	在 8 位定时器 / 事件计数器 50 操作过程中, 仅在选择 TM50 输出作为串行时钟时可操作		
上电清零功能		可操作		
低电压检测功能				
外部中断				

注 在停止模式下, 不要使用从外围硬件引脚输入的外部时钟上的这些功能。

备注 f_{RH} : 内部高速振荡时钟
 f_x : X1 时钟
 f_{EXCLK} : 外部主系统时钟
 f_{RL} : 内部低速振荡时钟

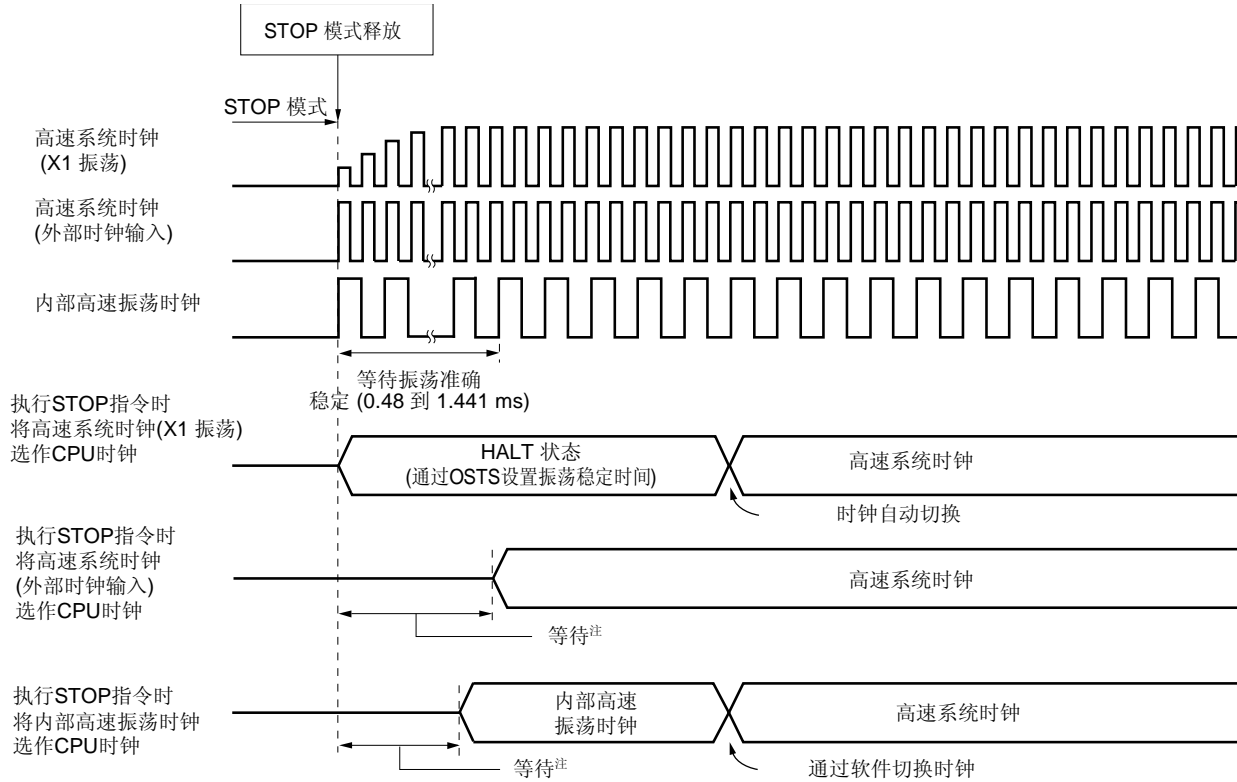
- 注意事项
1. 在 STOP 模式释放后, 要使用在 STOP 模式下停止操作的外围硬件, 以及在 STOP 模式下时钟停止振荡的外围硬件, 必须重新启动外围硬件。
 2. 即使通过选项字节选择“可由软件停止内部低速振荡器”, 在 STOP 模式下, 内部低速振荡时钟仍继续振荡, 并保持 STOP 模式设置前的状态。要在 STOP 模式下停止内部低速振荡器的振荡, 应通过软件停止该振荡器, 然后执行 STOP 指令。
 3. 当 CPU 使用高速系统时钟 (X1 振荡) 时, 为了在 STOP 模式释放后缩短振荡稳定时间, 可在下一次执行 STOP 指令前将 CPU 时钟暂时切换到内部高速振荡时钟。STOP 模式释放后, 在将 CPU 时钟从内部高速振荡时钟切换到高速系统时钟 (X1 振荡) 之前, 应使用振荡稳定时间计数器状态寄存器 (OSTC) 检测振荡稳定时间。

<R>

注意事项 4. 执行 STOP 指令时，必须确认内部高速振荡器的操作稳定（RSTS = 1）。

(2) STOP 模式释放

图 13-5. 释放 STOP 模式时的操作时序（当未屏蔽的中断请求产生时）



注 等待时间如下所示：

- 执行向量中断服务程序时：8 或 9 个时钟
- 不执行向量中断服务程序时：2 或 3 个时钟

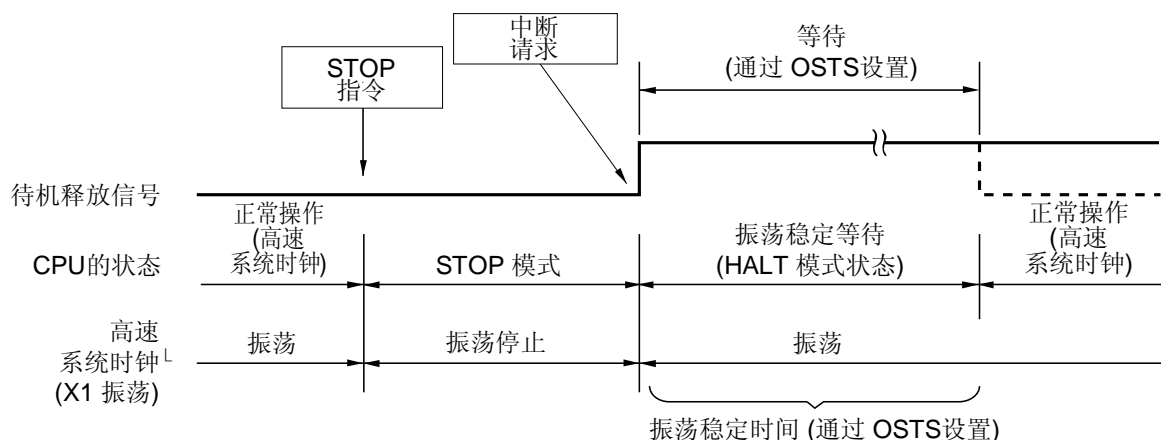
STOP 模式可以通过以下两种源来释放。

(a) 通过没有被屏蔽的中断请求释放

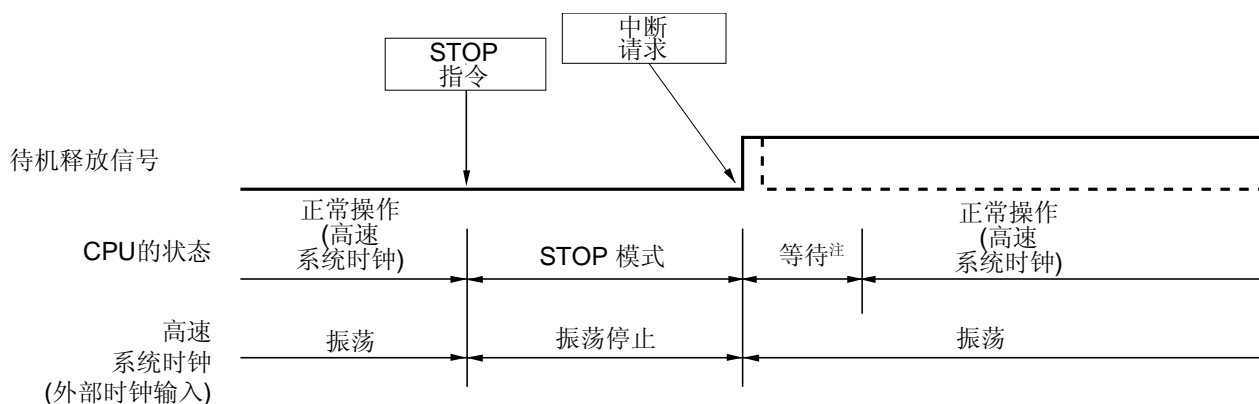
当产生一个没有被屏蔽的中断请求时，STOP 模式将会被释放。如果允许中断响应，则将执行向量中断服务程序。如果禁止中断响应，则将执行下一个地址指令。

图 13-6. 通过产生中断请求释放 STOP 模式 (1/2)

(1) 高速系统时钟 (X1 振荡) 被用作 CPU 时钟时



(2) 高速系统时钟 (外部时钟输入) 被用作 CPU 时钟时



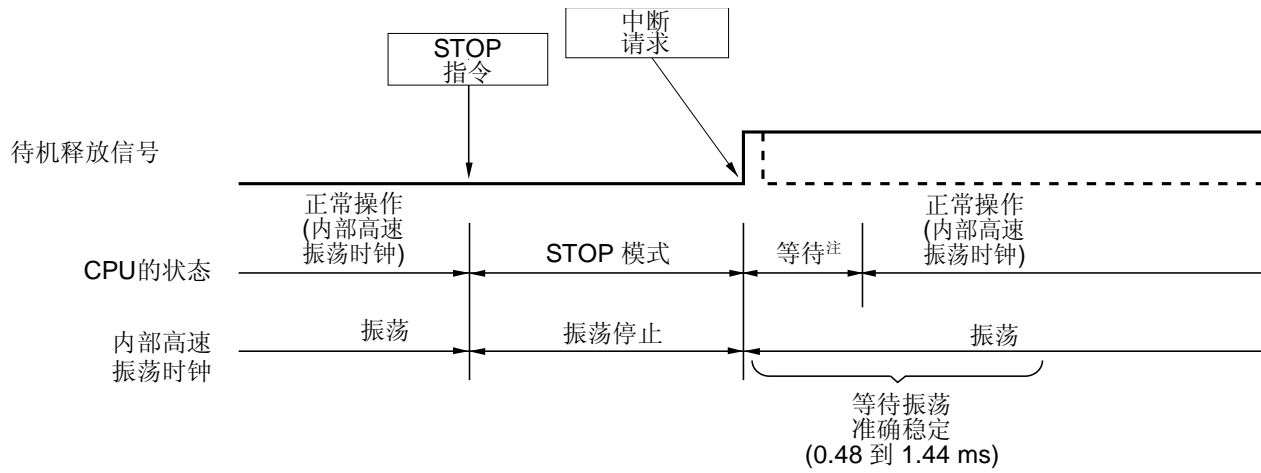
注 等待时间如下所示：

- 执行向量中断服务程序时：8 或 9 个时钟
- 不执行向量中断服务程序时：2 或 3 个时钟

备注 虚线表示释放待机模式的中断请求被响应时的情况。

图 13-6. 通过产生中断请求释放 STOP 模式 (2/2)

(3) 内部高速振荡时钟被用作 CPU 时钟时



注 等待时间如下所示：

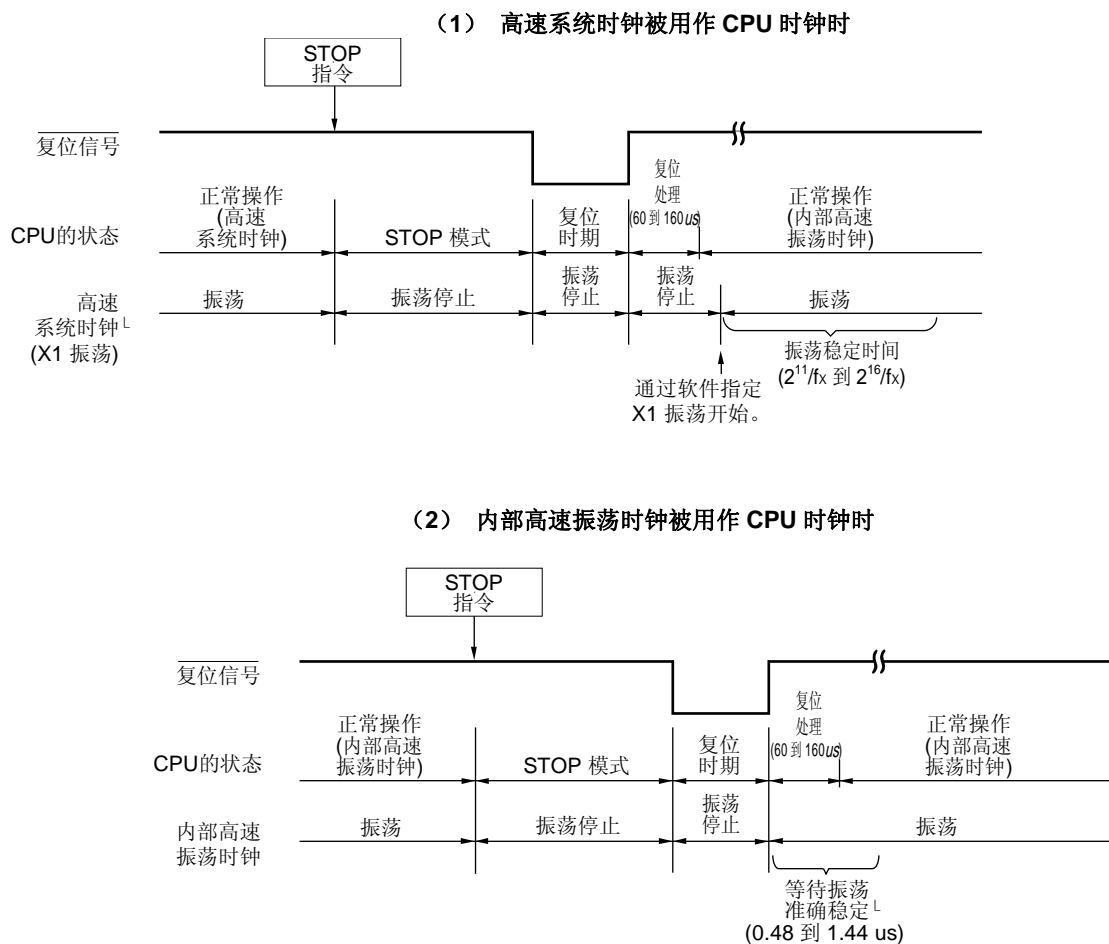
- 执行向量中断服务程序时： 8 或 9 个时钟
- 不执行向量中断服务程序时： 2 或 3 个时钟

备注 虚线表示释放待机模式的中断请求被响应时的情况。

(b) 通过复位信号生成释放

当复位信号生成时，STOP 模式被释放，然后，在进行正常复位操作时，程序会在跳转到复位向量地址后执行。

图 13-7. 由复位释放 STOP 模式



备注 fx: X1 时钟振动频率

表 13-4. 在 STOP 模式中响应中断请求的操作

释放源	MK _{xx}	PR _{xx}	IE	ISP	操作
可屏蔽中断请求	0	0	0	×	执行下一个地址指令
	0	0	1	×	执行中断服务程序
	0	1	0	1	执行下一个地址指令
	0	1	×	0	
	0	1	1	1	执行中断服务程序
	1	×	×	×	保持 STOP 模式
复位	-	-	×	×	复位处理

x: 不考虑

第十四章 复位功能

以下 4 种操作用于产生复位信号。

- (1) 通过 $\overline{\text{RESET}}$ 引脚输入的外部复位
- (2) 通过看门狗定时器程序循环检测引起的内部复位
- (3) 通过比较上电清零 (POC) 电路的检测电压和供电电压引起的内部复位
- (4) 通过比较低电压检测器 (LVI) 的检测电压和供电电压引起的内部复位

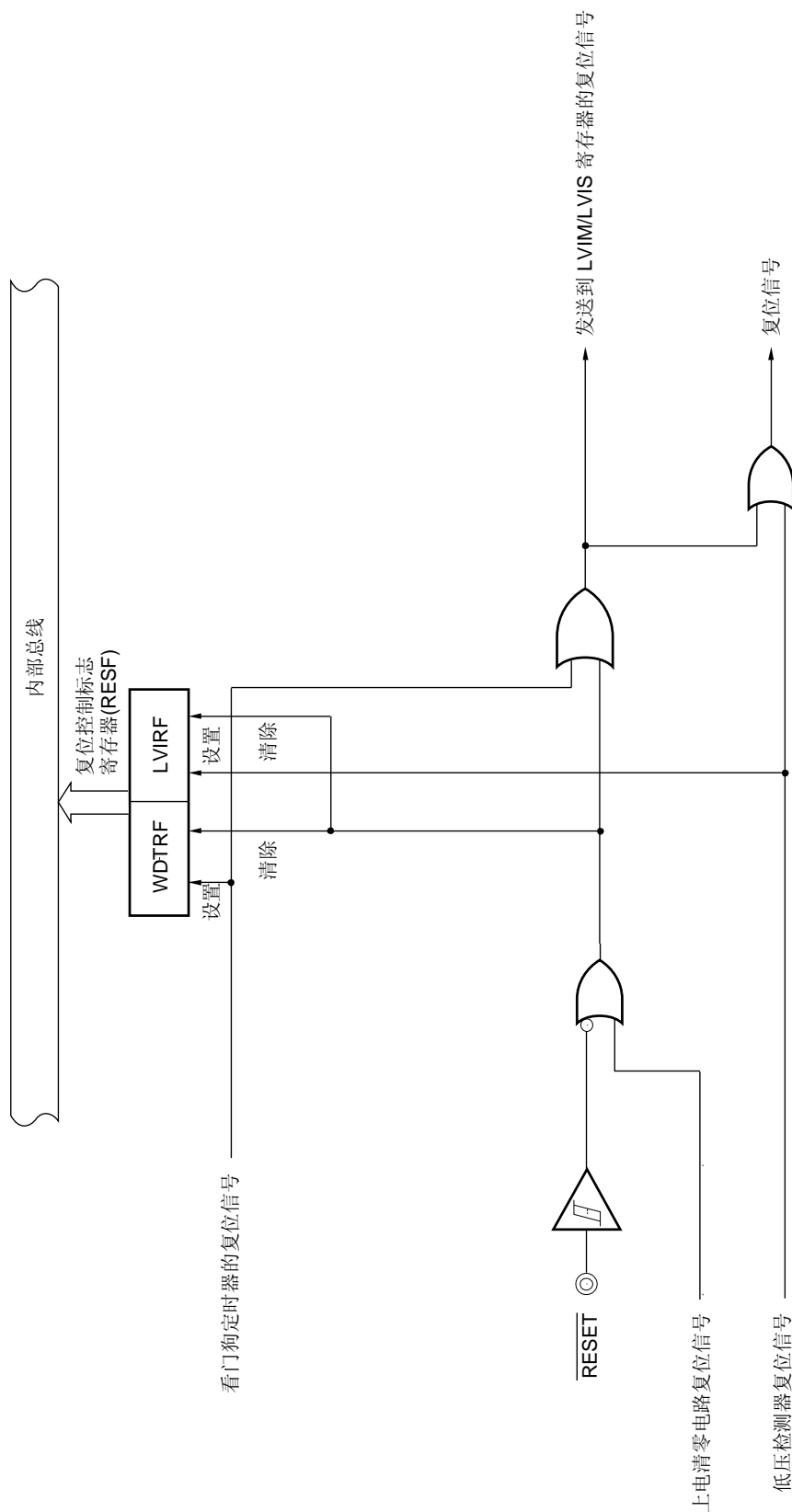
外部复位与内部复位在功能上没有什么区别。在这两种情况中，当产生复位信号时，程序都是从地址 0000H 和 0001H 处开始执行。

当 $\overline{\text{RESET}}$ 引脚输入为低电平，看门狗定时器溢出或 POC 和 LVI 电路的电压检测都可以引起复位，且硬件的各项会设置为如表 14-1 和 14-2 所示的状态。在复位信号产生期间或在复位释放后的振荡稳定时间内，每个引脚均为高阻抗。

当 $\overline{\text{RESET}}$ 引脚输入低电平时，设备被复位。当 $\overline{\text{RESET}}$ 引脚输入高电平时，设备从复位状态释放，并在复位处理后使用内部高速振荡时钟执行程序。由看门狗定时器引起的复位自动释放，并在复位处理后使用内部高速振荡时钟执行程序（参见图 14-2 到 14-4）。在复位后，由 POC 和 LVI 电路供电电压检测引起的复位在 $V_{DD} \geq V_{POC}$ or $V_{DD} \geq V_{LVI}$ 时自动释放，并在复位处理后使用内部高速振荡时钟执行程序（参见第十五章 上电清零电路 和 第十六章 低电压检测器）。

- 注意事项**
1. 对于外部复位，向 $\overline{\text{RESET}}$ 引脚输入低电平的时间应为 10 μs 或更长。
 2. 复位输入期间，X1 时钟，内部高速振荡时钟和内部低速振荡时钟停止振荡。外部主系统时钟输入变为无效。
 3. 在通过复位释放 STOP 模式时，在复位输入期间会保持 STOP 模式的内容。然而，端口引脚会变为高阻抗。

图 14-1. 复位功能的框图



注意事项 LVI 电路内部复位不会复位 LVI 电路。

- 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 14-2. 通过 RESET 输入复位的时序

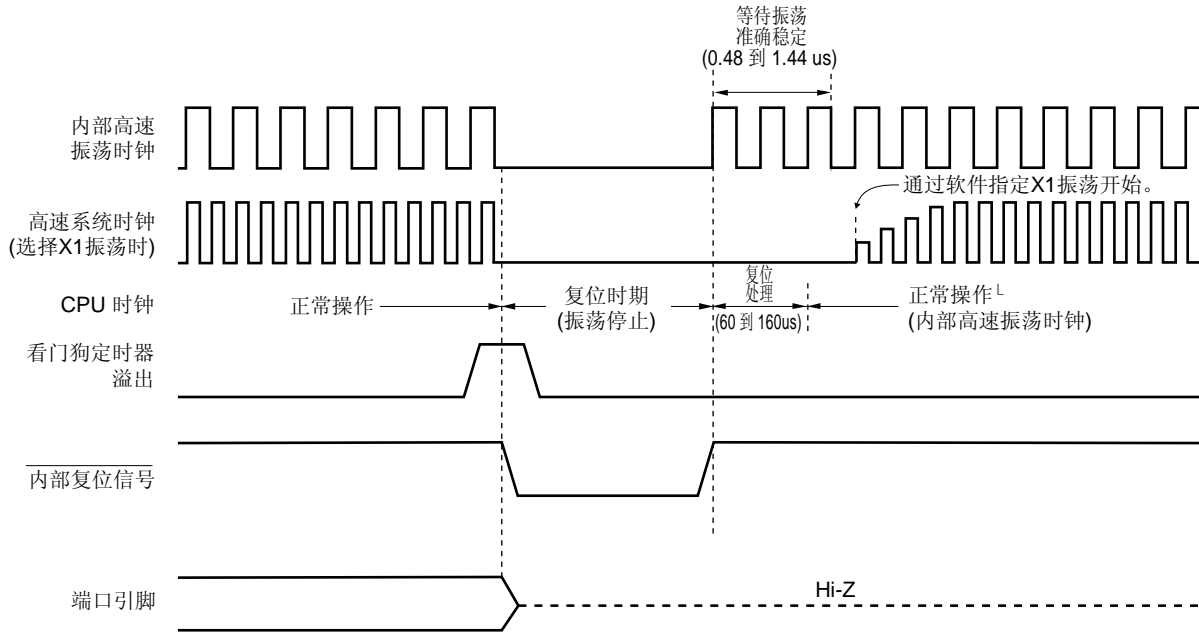
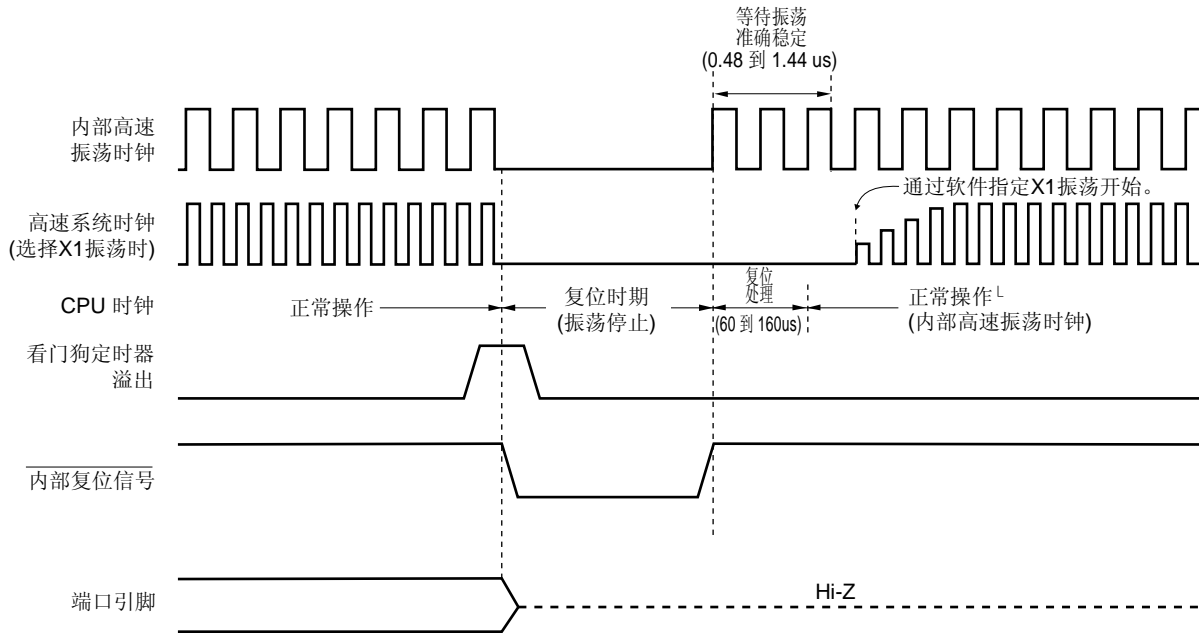
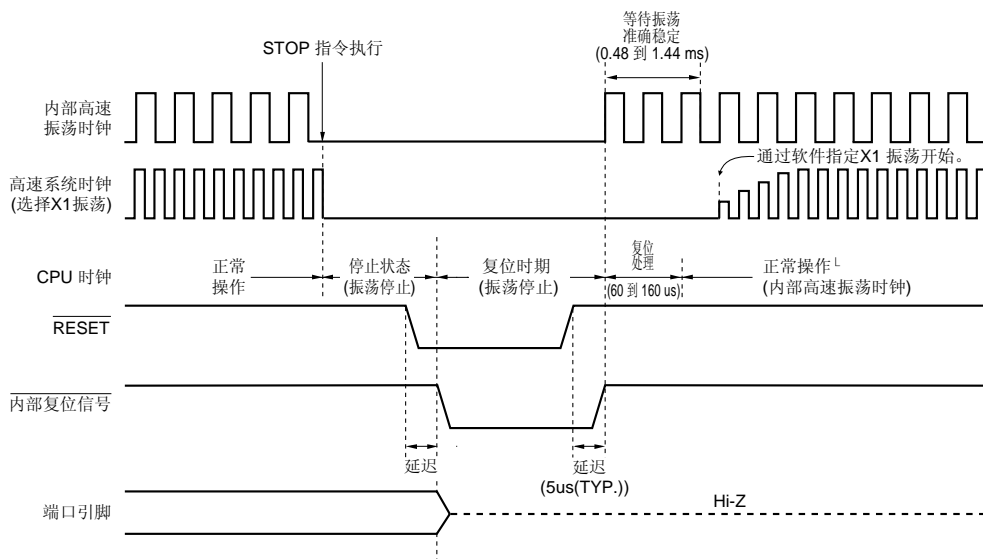


图 14-3. 由于看门狗定时器溢出进行复位的时序



注意事项 看门狗定时器内部复位功能可以复位看门狗定时器。

图 14-4. 在 STOP 模式下通过 RESET 输入复位的时序



备注 关于上电清零电路和低电压检测器的复位时序，请参见第十五章 上电清零电路 和 第十六章 低电压检测器。

表 14-1. 复位期间的操作状态

项目	复位期间	
系统时钟	提供给 CPU 的时钟被停止。	
主系统时钟	f _{RH}	操作停止
	f _X	操作停止 (引脚为 I/O 端口模式)
	f _{EXCLK}	时钟输入无效 (引脚为 I/O 端口模式)
	f _{RL}	操作停止
CPU		
Flash 存储器		
RAM		
端口 (锁存)		
16-位定时器 / 事件计数器 00		
8 位定时器 / 事件计数器	50	
	51	
8 位定时器	H0	
	H1	
看门狗定时器		
串行接口	UART6	
上电清零功能	可操作	
低电压检测功能	操作停止	
外部中断		

备注

- f_{RH}: 内部高速振荡时钟
- f_X: X1 振荡时钟
- f_{EXCLK}: 外部主系统时钟
- f_{RL}: 内部低速振荡时钟

表 14-2. 复位响应后的硬件状态 (1/2)

硬件		复位响应后 ^{注1}
程序计数器 (PC)		复位向量表 (0000H, 0001H) 中的内容被设置。
堆栈指针 (SP)		未定义
程序状态字 (PSW)		02H
RAM	数据存储	未定义 ^{注2}
	通用寄存器	未定义 ^{注2}
端口寄存器 (P0 到 P2, P3 ^{注3} , P12) (输出锁存)		00H
端口模式寄存器 (PM0 到 PM2, PM3 ^{注3} , PM12)		FFH
上拉电阻选项寄存器 (PU0 到 PU2, PU3 ^{注3} , PU12)		00H
端口输出模式电阻 (POM0 到 POM2, POM3 ^{注3} , POM12)		00H
上拉电阻选项寄存器 (PU0 到 PU2, PU3 ^{注3} , PU12)		00H (PU12 中的 08H)
FLMD0 上拉 / 下拉控制寄存器 (FPCTL)		00H
FLMD0 上拉 / 下拉允许寄存器 (FPEN)		00H
内部存储器容量切换寄存器 (IMS)		CFH ^{注4}
时钟操作模式选择寄存器 (OSCCTL)		00H
处理器时钟控制寄存器 (PCC)		01H
内置振荡模式寄存器 (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
	捕获 / 比较寄存器 000, 010 (CR000, CR010)	0000H
	模式控制寄存器 00 (TMC00)	00H
	预分频模式寄存器 00 (PRM00)	00H
	捕获 / 比较控制寄存器 00 (CRC00)	00H
	定时器输出控制寄存器 00 (TOC00)	00H

- 注
1. 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后所有其它硬件状态保持不变。
 2. 当在待机模式中进行复位时，即使复位后也保持复位前的状态。
 3. 仅限 38 引脚的产品
 4. 不管内部存储器容量多大，所有 μ PD179F11x, 179F12x 微控制器产品中的内部存储器容量切换寄存器的初始值在复位释放后都是恒定的 (IMS = CFH)。因此，复位释放后，必须按如下来设置各个产品的值。

Flash 存储器版本 (μ PD179F11x, 179F12x 微控制器)	IMS
μ PD179F110	41H
μ PD179F111	42H
μ PD179F112, 179F122	04H
μ PD179F113, 179F123	C6H
μ PD179F114, 179F124	C8H

表 14-2. 复位响应后的硬件状态 (2/2)

硬件		复位响应后的状态 ^{注1}
8 位定时器 / 事件计数器 50, 51	定时计数器 50, 51 (TM50, TM51)	00H
	比较寄存器 50, 51 (CR50, CR51)	00H
	定时时钟选择寄存器 50, 51 (TCL50, TCL51)	00H
	模式控制寄存器 50, 51 (TMC50, TMC51)	00H
8 位定时器 H0, H1	比较寄存器 00, 10, 01, 11 (CMP00, CMP10, CMP01, CMP11)	00H
	模式寄存器 (TMHMD0, TMHMD1)	00H
	载波控制寄存器 1 (TMCYC1) ^{注2}	00H
看门狗定时器	允许寄存器 (WDTE)	1AH/9AH ^{注3}
串行接口 UART6	接收缓冲寄存器 6 (RXB6)	FFH
	发送缓冲寄存器 6 (TXB6)	FFH
	异步串行接口操作模式寄存器 6 (ASIM6)	01H
	异步串行接口接收错误状态寄存器 6 (ASIS6)	00H
	异步串行接口发送状态寄存器 6 (ASIF6)	00H
	时钟选择寄存器 6 (CKSR6)	00H
	波特率发生器控制寄存器 6 (BRGC6)	FFH
	异步串行接口控制寄存器 6 (ASICL6)	16H
	输入转换控制寄存器 (ISC)	00H
按键中断	按键返回模式寄存器 0, 1 (KRML, KRMH)	00H
复位功能	复位控制标志寄存器 (RESF)	00H ^{注4}
	复位引脚模式寄存器 (RSTMASK)	00H
低压检测器	低电压检测寄存器 (LVIM)	00H ^{注4}
	低电压检测等级选择寄存器 (LVIS)	00H ^{注4}
	RAM 数据保存控制寄存器 (LVDET)	未定义
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. 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后所有其它硬件状态保持不变。
 2. 仅限 8 位定时器 H1
 3. WDTE 的复位值由选项字节的设置来决定。
 4. 这些值根据复位源而变化。

寄存器 \ 复位源		RESET 输入	通过 POC 复位	通过 WDT 复位	通过 LVI 复位
		RESF	清除 (0)	清除 (0)	设置 (1)
	WDTRF 位			保持	设置 (1)
	LVIRF 位				
LVIM		清除 (00H)	清除 (00H)	清除 (00H)	保持
LVIS					

14.1 确认复位源的寄存器

μ PD179F11x, 179F12x 微控制器中存在有许多内部复位发生源。复位控制标志寄存器 (RESF) 用于存储产生复位请求的复位源。

RESF 可以通过 8 位存储器操作指令来进行读取。

RESET 输入, 由上电清零 (POC) 电路引起的复位输入以及读取 RESF 都会将 RESF 设置为 00H。

图 14-5. 复位控制标志寄存器 (RESF) 的格式

地址: FFACH 复位后: 00H^{*} R

符号	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LVIRF

WDTRF	看门狗定时器 (WDT) 产生的内部复位请求
0	不产生内部复位请求, 或 RESF 被清零
1	产生内部复位请求

LVIRF	由低电压检测器 (LVI) 产生的内部复位请求
0	不产生内部复位请求, 或 RESF 被清零
1	产生内部复位请求

注 复位后的值根据复位源的不同而变化。

注意事项 不要通过 1 位存储器操作指令读取数据。

产生复位请求时 RESF 的状态如表 14-3 所示。

表 14-3. 产生复位请求时 RESF 的状态

标志 \ 复位源	RESET 输入	通过 POC 复位	通过 WDT 复位	通过 LVI 复位
WDTRF	清除 (0)	清除 (0)	设置 (1)	保持
LVIRF			保持	设置 (1)

第十五章 上电清零电路

15.1 上电清零电路的功能

上电清零电路（POC）具有以下功能。

- 在上电时产生内部复位信号。
当供电电压（ V_{DD} ）超过 $1.8\text{ V} \pm 0.1\text{ V}$ 时释放复位信号。
- 比较供电电压（ V_{DD} ）和检测电压（ $V_{POC} = 1.8\text{ V} \pm 0.1\text{ V}$ ），并在 $V_{DD} < V_{POC}$ 时产生内部复位信号。

注意事项 如果在 POC 电路中产生内部复位信号，则将复位控制标志寄存器（RESF）清除为 00H。

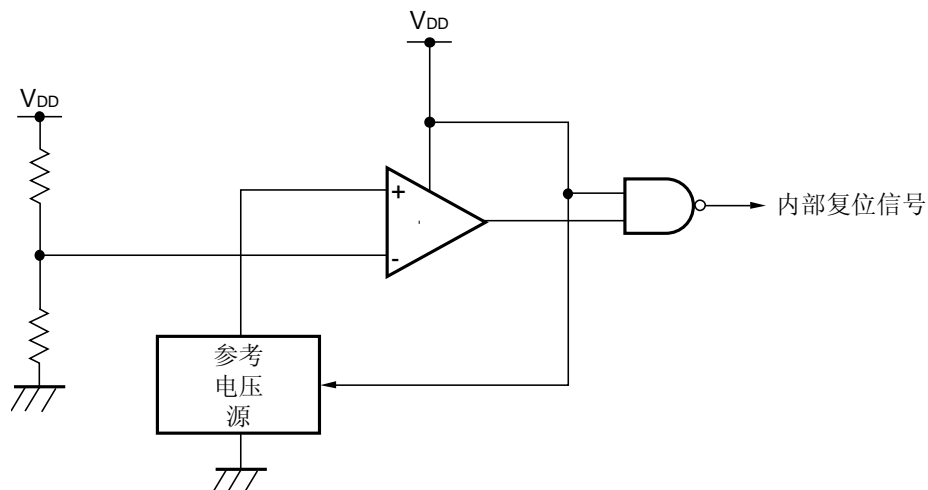
备注 μ PD179F11x, 179F12x 微控制器集成了多种可以产生内部复位信号的硬件功能。复位控制标志寄存器（RESF）中有一个标志用于指示复位源，用于看门狗定时器（WDT）或低电压检测器（LVI）产生的内部复位信号时。当 WDT 或 LVI 产生内部复位信号时，RESF 不会被清除为 00H，而标志则会被设置为 1。

关于 RESF 的详细信息，参见第十四章 复位功能。

15.2 上电清零电路的配置

上电清零电路的框图如图 15-1 所示。

图 15-1. 上电清零电路的框图

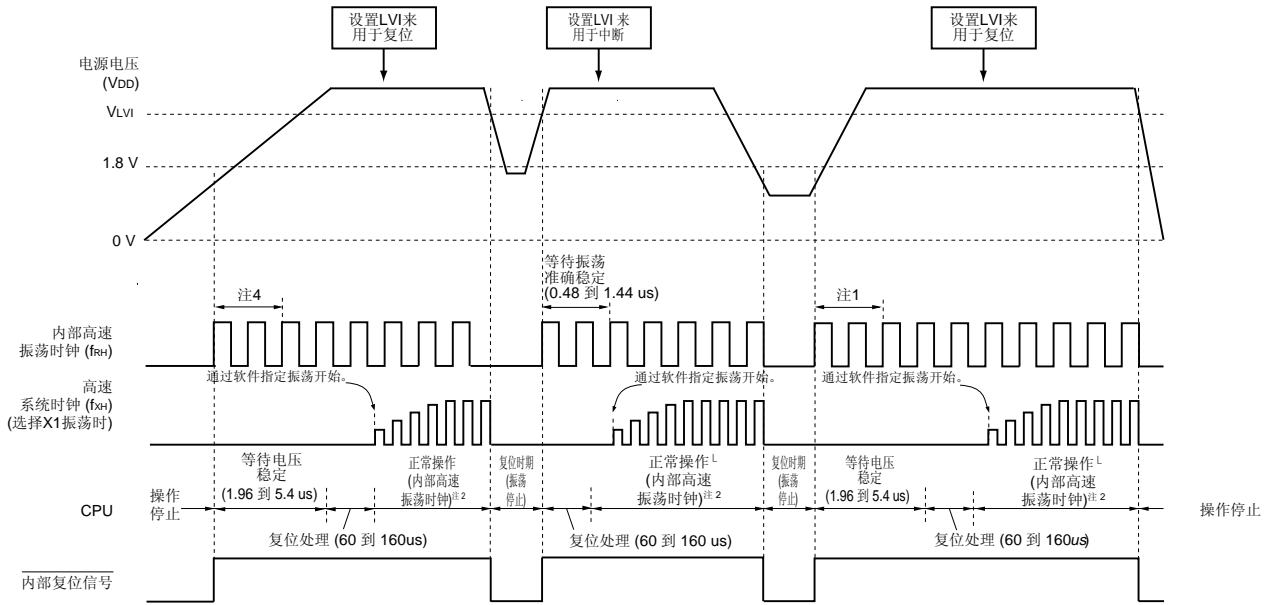


15.3 上电清零电路的操作

- 上电时会产生一个内部复位信号。当供电电压 (V_{DD}) 超过检测电压 ($V_{POC} = 1.8\text{ V} \pm 0.1\text{ V}$) 时，复位状态被释放。
- 供电电压 (V_{DD}) 与检测电压 ($V_{POC} = 1.8\text{ V} \pm 0.1\text{ V}$) 进行比较。当 $V_{DD} < V_{POC}$ 时，产生内部复位信号。当 $V_{DD} \geq V_{POC}$ 时释放内部复位信号。

由上电清零电路和低电压检测器产生内部复位信号的时序如下所示。

图 15-2. 由上电清零电路和低电压检测器产生内部复位信号的时序



- 注
1. 内部高速振荡时钟的振荡准确稳定时间包含在内部电压稳定时间内。
 2. CPU 时钟可以从内部高速振荡时钟切换为高速系统时钟。若要使用 X1 时钟，则应使用 OSTC 寄存器来确定是否经过了振荡稳定时间。

注意事项 在释放复位状态后通过软件设置低电压检测器（参见第十六章 低电压检测器）。

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

15.4 使用上电清零电路的注意事项

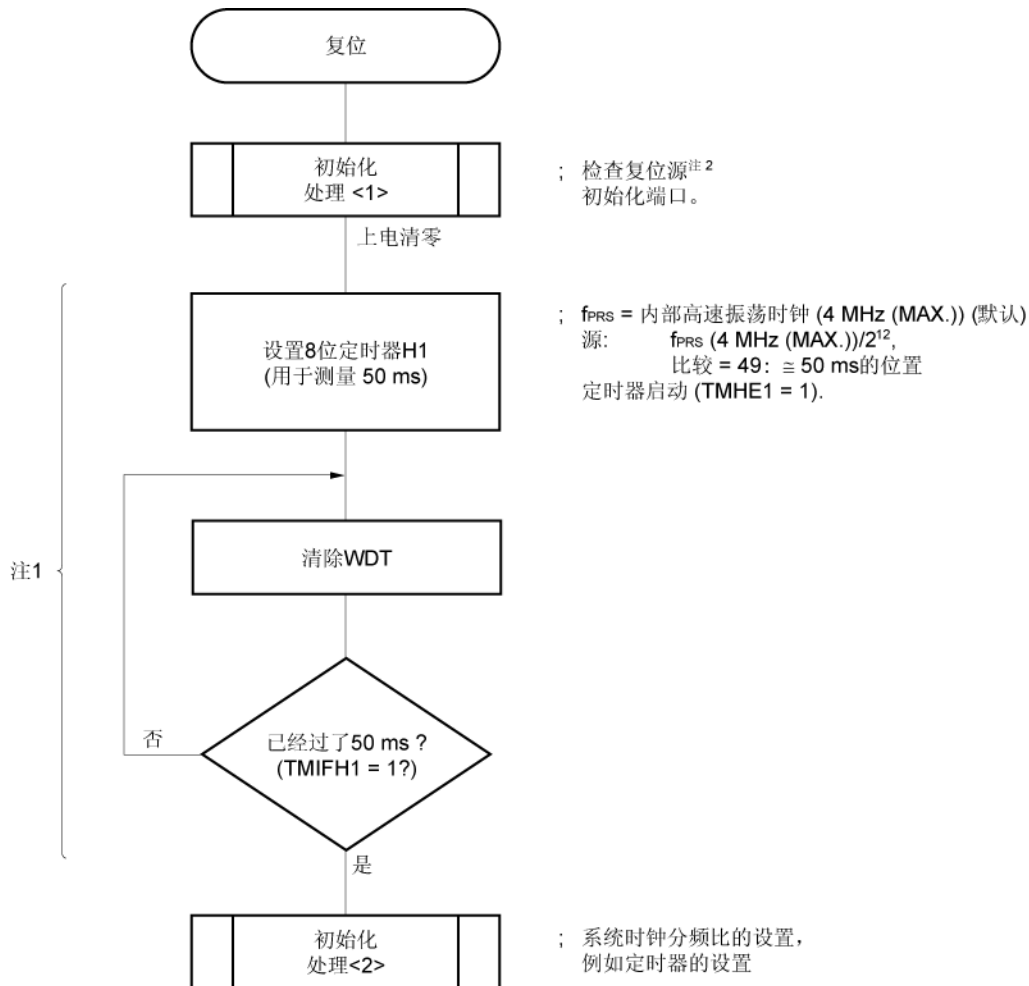
在系统中，如果供电电压（V_{DD}）接近 POC 检测电压（V_{POC}）时的一段时间内产生波动，则系统可能会反复进行复位并从复位状态中释放。在这种情况下，可采用下列方法任意设置从复位释放到微控制器启动所需要的时间。

<方法>

在释放复位信号后，通过使用定时器的软件计数器等待各个系统供电电压的波动期，然后对端口进行初始化。

图 15-3. 复位释放后软件处理过程示例（1/2）

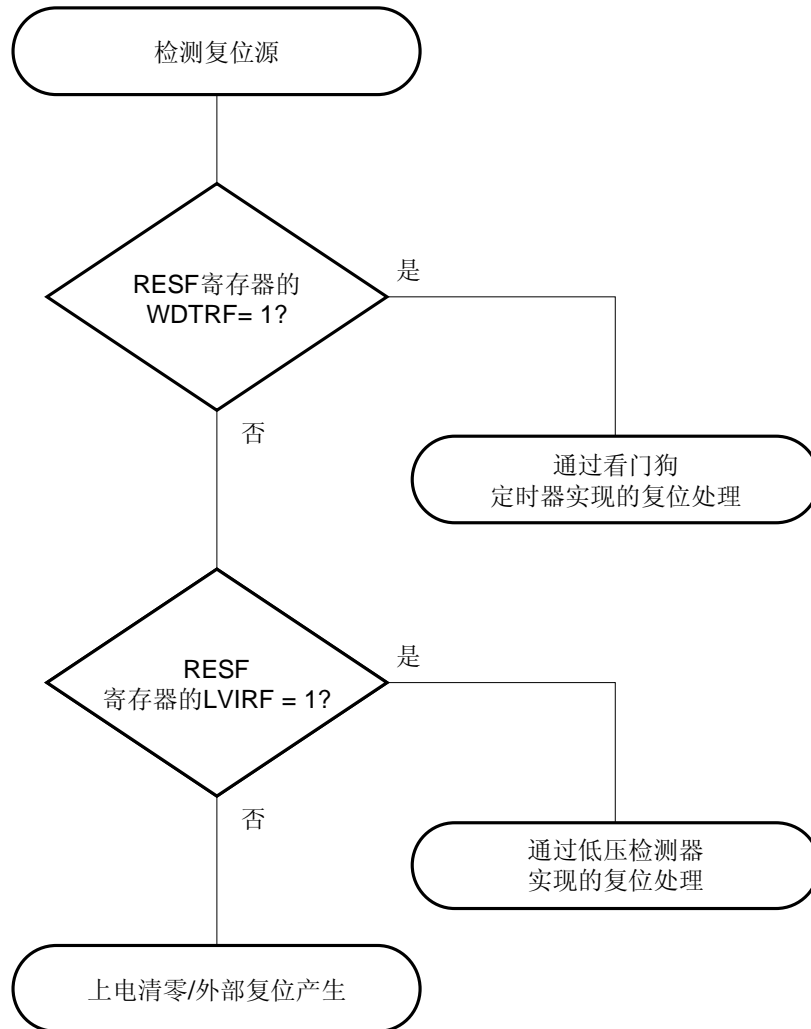
- 如果供电电压接近 POC 检测电压，其波动时间为 50 ms 或更短



- 注 1. 如果在此期间再次产生复位，则不启动初始化操作<2>。
 2. 流程图显示在下页。

图 15-3. 复位释放后软件处理过程示例 (2/2)

• 检测复位源



第十六章 低电压检测器

16.1 低电压检测器的功能

低电压检测器 (LVI) 具有以下功能。

- LVI 电路用于将供电电压 (V_{DD}) 和检测电压 (V_{LVI}) 或带有检测电压 ($V_{EXLVI} = 1.21\text{ V (TYP.)}$): 固定的) 的外部输入引脚 (EXLVI) 中的输入电压进行比较并生成一个内部复位或内部中断信号。
- 可以通过软件选择供电电压 (V_{DD}) 或外部输入引脚的输入电压 (EXLVI)。
- 可以通过软件选择复位或中断功能。
- 可以通过软件来改变供电电压的检测等级 (共 11 级)。
- 在 STOP 模式下可操作。
- RAM 数据保持检测 (参照 16.6 RAM 数据保持检测器)

复位信号和中断信号依据软件的选择按照如下方式产生。

供电电压 (V_{DD}) 电平检测的选择 (LVISEL = 0)		外部输入引脚 (EXLVI) 输入电压 电平检测的选择 (LVISEL = 1)	
选择复位 (LVIMD = 1)。	选择中断 (LVIMD = 0)。	选择复位 (LVIMD = 1)。	选择中断 (LVIMD = 0)。
当 $V_{DD} < V_{LVI}$ 时产生内部复位信号并且当 $V_{DD} \geq V_{LVI}$ 时释放复位信号。	当 V_{DD} 下降至比 V_{LVI} 低 ($V_{DD} < V_{LVI}$) 时或者当 V_{DD} 变成 V_{LVI} 或更高 ($V_{DD} \geq V_{LVI}$) 时产生内部中断信号。	当 $EXLVI < V_{EXLVI}$ 时产生内部复位信号并且当 $EXLVI \geq V_{EXLVI}$ 时释放复位信号。	当 $EXLVI$ 下降至比 V_{EXLVI} 低 ($EXLVI < V_{EXLVI}$) 时或者当 $EXLVI$ 变成 V_{EXLVI} 或更高 ($EXLVI \geq V_{EXLVI}$) 时产生内部中断信号。

备注 LVISEL: 低电压检测寄存器 (LVIM) 的第 2 位
LVIMD: LVIM 的第 1 位

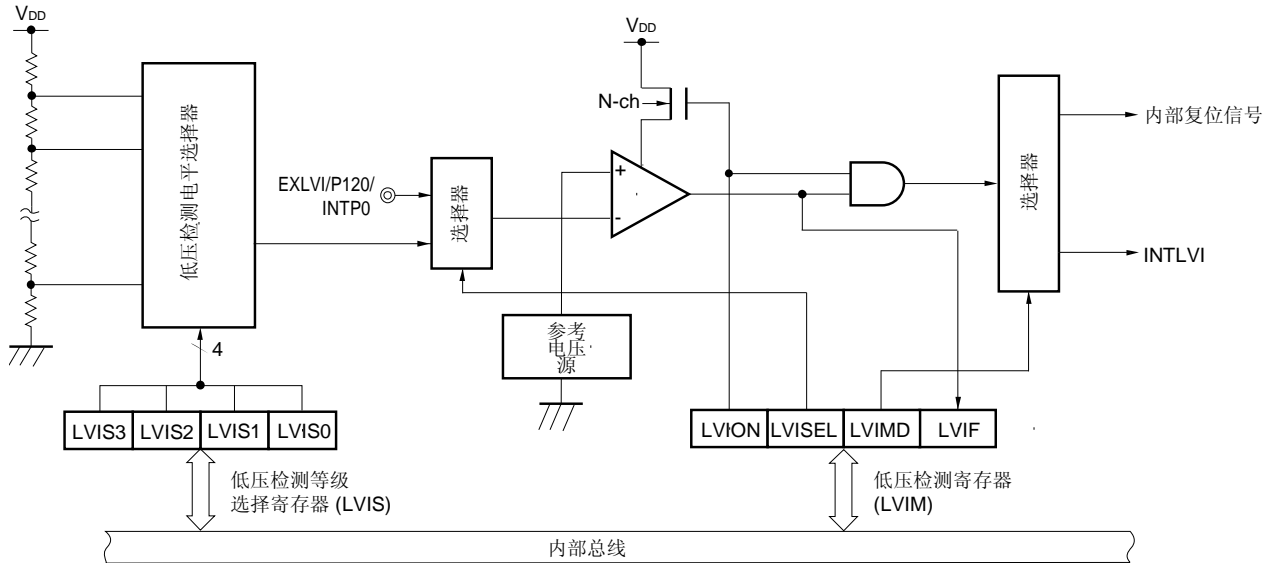
当低电压检测器运行时, 供电电压或者外部输入引脚的输入电压可以通过读取低电压检测标志 (LVIF: LVIM 的位 0) 来得知其是否高于或低于检测电平。

当低电压检测器用于复位时, 如果复位产生, 复位控制标志寄存器 (RESF) 的 0 位 (LVIRF) 将会被设为 1。关于 RESF 的详细信息, 参见第十四章 复位功能。

16.2 低电压检测器的配置

低电压检测器的框图如图 16-1 所示。

图 16-1. 低电压检测器的框图



16.3 控制低电压检测器的寄存器

低电压检测器使用以下寄存器进行控制。

- 低电压检测寄存器 (LVIM)
- 低电压检测等级选择寄存器 (LVIS)
- 端口模式寄存器 12 (PM12)

(1) 低电压检测寄存器 (LVIM)

该寄存器用于设置低电压检测和操作模式。

该寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。

除 LVI 复位外的复位信号的产生会将该寄存器清除为 00H。

图 16-2. 低电压检测寄存器 (LVIM) 的格式

地址: FFBEH 复位后: 00H^{注1} R/W^{注2}

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

LVION ^{注3, 4}	允许低电压检测操作
0	禁止操作
1	允许操作

LVISEL ^{注3}	电压检测的选择
0	检测供电电压 (V _{DD}) 的等级
1	检测外部输入引脚 (EXLVI) 的输入电压的等级

LVIMD ^{注3}	低电压检测操作模式 (中断 / 复位) 的选择
0	<ul style="list-style-type: none"> LVISEL = 0: 当供电电压 (V_{DD}) 下降至比检测电压 (V_{LVI}) 低 (V_{DD} < V_{LVI}) 时或者当 V_{DD} 变成 V_{LVI} 或更高 (V_{DD} ≥ V_{LVI}) 时产生内部中断信号。 LVISEL = 1: 当外部输入引脚 (EXLVI) 的输入电压下降至低于检测电压 (V_{EXLVI}) (EXLVI < V_{EXLVI}) 时或当 EXLVI 变为 V_{EXLVI} 或更高 (EXLVI ≥ V_{EXLVI}) 时产生中断信号。
1	<ul style="list-style-type: none"> LVISEL = 0: 当供电电压 (V_{DD}) < 检测电压 (V_{LVI}) 时产生内部复位信号并且当 V_{DD} ≥ V_{LVI} 时释放复位信号。 LVISEL = 1: 当外部输入引脚 (EXLVI) 的输入电压 < 检测电压 (V_{EXLVI}) 时产生一个内部复位信号并且在 EXLVI ≥ V_{EXLVI} 时释放复位信号。

LVIF ^{注4}	低电压检测标志
0	<ul style="list-style-type: none"> LVISEL = 0: 供电电压 (V_{DD}) ≥ 检测电压 (V_{LVI}) 或当禁止操作时 LVISEL = 1: 外部输入引脚 (EXLVI) 的输入电压 ≥ 检测电压 (V_{EXLVI})，或当禁止操作时
1	<ul style="list-style-type: none"> LVISEL = 0: 供电电压 (V_{DD}) < 检测电压 (V_{LVI}) LVISEL = 1: 外部输入引脚的输入电压 (EXLVI) < 检测电压 (V_{EXLVI})

- 注
- 除 LVI 复位外的复位进行时该位被清除为 00H。
 - 第 0 位是只读的。
 - 在产生除 LVI 复位以外的复位时，LVION，LVIMD 和 LVISEL 被清零。在产生 LVI 复位时这些位不被清零。
 - 当 LVION 被设为 1 时，LVI 电路中的比较器开始操作。使用软件等待从设置 LVION 被设为 1 到操作稳定的操作稳定时间 (10 μs (MAX.))。操作稳定后，从进入低于 LVI 检测电压的状态开始到 LVIF 被设为 1 需要 50 μs (TYP.) 的时间。

- 注意事项
- 若要停止 LVI，则按以下任一操作执行。
 - 当使用 8 位存储器操作指令时：写 00H 到 LVIM 中。
 - 当使用 1 位存储器操作指令时：将 LVION 清除为 0。
 - 外部输入引脚 (EXLVI) 的输入电压必须是 EXLVI < V_{DD}。
 - 生成 LVI 复位后，当 LVION = 1 时不要将值写入 LVIS 和 LVIM 中。
 - 将 LVI 用作中断时，如果 LVION 在低于 LVI 检测电压的状态下被清零，那么将会生成 INTLVI 信号且 LVIF 会被设为 1。

(2) 低电压检测等级选择寄存器 (LVIS)

该寄存器用于选择供电电压 (V_{DD}) 的低电压检测等级。

检测到外部输入引脚 (EXLVI) 的输入电压时，检测电压是固定的 (V_{EXLVI} = 1.21 V (TYP.))。因此，不需要 LVIS 的设置。

该寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。
除 LVI 复位外的复位信号的产生会将该寄存器清除为 00H。

图 16-3. 低电压检测等级选择寄存器 (LVIS) 的格式

地址: FFBFH 复位后: 00H^{注1} 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	检测等级
不同于以下的其他情况				禁止设置
0	1	0	1	V _{LV15} (3.47 V ±0.1 V)
0	1	1	0	V _{LV16} (3.32 V ±0.1 V)
0	1	1	1	V _{LV17} (3.16 V ±0.1 V)
1	0	0	0	V _{LV18} (3.01 V ±0.1 V)
1	0	0	1	V _{LV19} (2.85 V ±0.1 V)
1	0	1	0	V _{LV110} (2.70 V ±0.1 V)
1	0	1	1	V _{LV111} (2.55 V ±0.1 V)
1	1	0	0	V _{LV112} (2.39 V ±0.1 V)
1	1	0	1	V _{LV113} (2.24 V ±0.1 V)
1	1	1	0	V _{LV114} (2.05 V ±0.05 V)
1	1	1	1	V _{LV115} (1.93 V ±0.1 V)

注 由 LVI 引起复位后, LVIS 的数值并不复位, 而是保留该数值。其它原因引起的复位会将其清除为 00H。

- 注意事项
1. 确保将位 4 至位 7 清除为“0”。
 2. LVI 操作期间不要修改 LVIS 的值。
 3. 生成 LVI 复位后, 当 LVION = 1 时不要将值写入 LVIS 和 LVIM 中。

(3) 端口模式寄存器 12 (PM12)

当使用 P120/EXLVI/INTP0 引脚进行外部低电压检测电平输入时，会将 PM120 设为 1。此时，P120 的输出锁存的值可以为 0 或 1。

PM12 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将 PM12 设置为 FFH。

图 16-4. 端口模式寄存器 12 (PM12) 的格式

地址: FF2CH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM12	1	1	1	1	1	PM122	PM121	PM120

PM12n	P12n 引脚 I/O 模式选择 (n = 0 到 2)
0	输出模式 (输出缓冲开)
1	输入模式 (输出缓冲关)

16.4 低电压检测器的操作

低电压检测器能够用于以下两种模式。

(1) 用作复位 (LVIMD = 1)

- 如果 LVISEL = 0，比较供电电压 (V_{DD}) 与检测电压 (V_{LVI})，当 $V_{DD} < V_{LVI}$ 时将产生内部复位信号，而当 $V_{DD} \geq V_{LVI}$ 时则将释放内部复位。
- 如果 LVISEL = 1，比较外部输入引脚 (EXLVI) 的输入电压和检测电压 ($V_{EXLVI} = 1.21 \text{ V (TYP.)}$)，在 $EXLVI < V_{EXLVI}$ 时会产生内部复位信号，而在 $EXLVI \geq V_{EXLVI}$ 时则将释放内部复位。

(2) 用作中断 (LVIMD = 0)

- 如果 LVISEL = 0，比较供电电压 (V_{DD}) 与检测电压 (V_{LVI})。当 V_{DD} 下降至比 V_{LVI} 低 ($V_{DD} < V_{LVI}$) 时或者当 V_{DD} 变成 V_{LVI} 或更高 ($V_{DD} \geq V_{LVI}$) 时会产生一个中断信号 (INTLVI)。
- 如果 LVISEL = 1，比较外部输入引脚 (EXLVI) 的输入电压和检测电压 ($V_{EXLVI} = 1.21 \text{ V (TYP.)}$)。当 EXLVI 下降至比 V_{EXLVI} 低 ($EXLVI < V_{EXLVI}$) 时或者当 EXLVI 变成 V_{EXLVI} 或更高 ($EXLVI \geq V_{EXLVI}$) 时将会产生一个中断信号 (INTLVI)。

当低电压检测器运行时，供电电压或者外部输入引脚的输入电压可以通过读取低电压检测标志 (LVIF: LVIM 的位 0) 来得知其是否高于或低于检测电平。

备注 LVIMD: 低电压检测寄存器 (LVIM) 的第 1 位
LVISEL: LVIM 的第 2 位

16.4.1 用作复位时

(1) 当检测供电电压 (V_{DD}) 的等级时

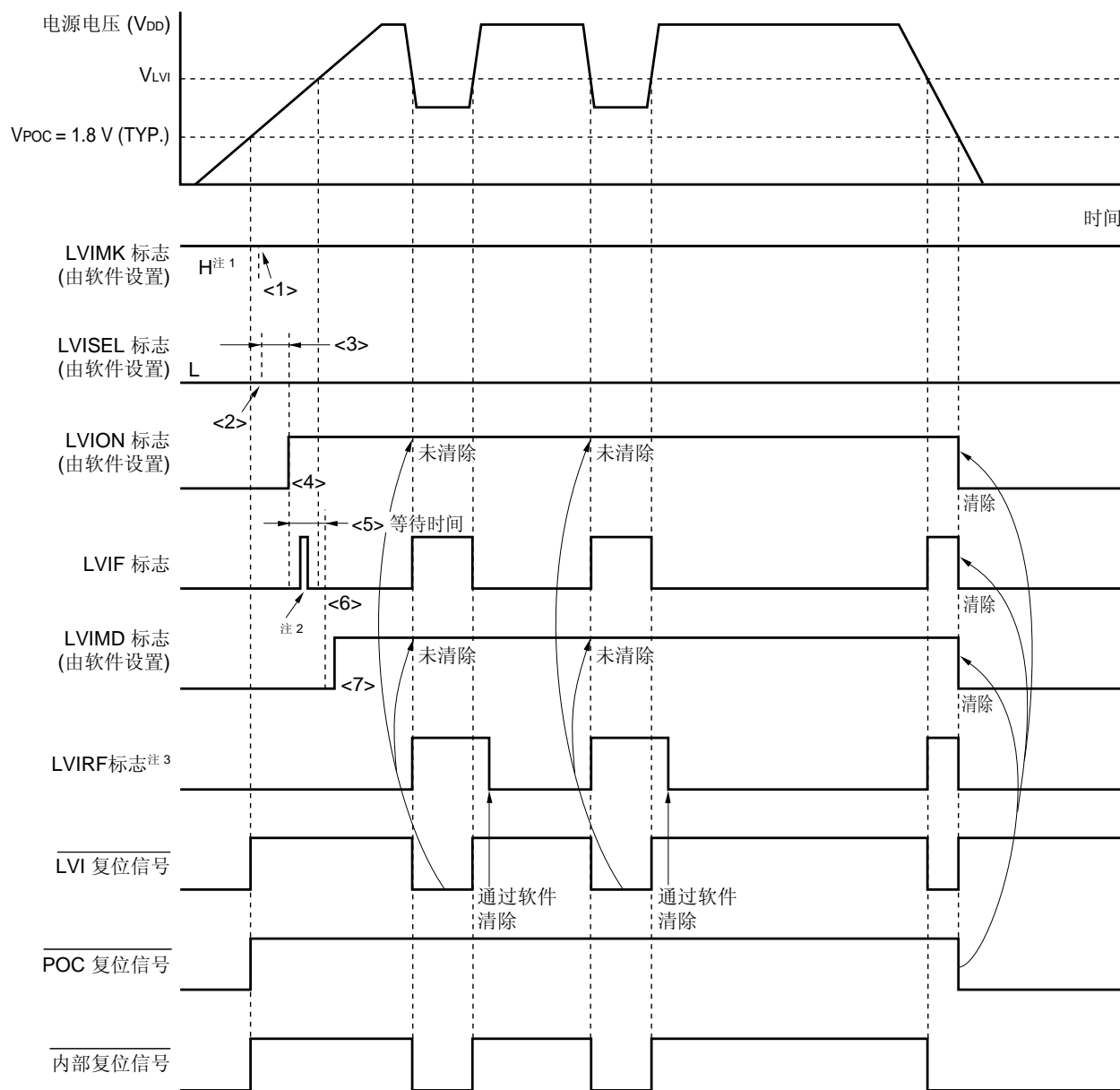
- 启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <2> 将低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 清零 (检测供电电压 (V_{DD}) 的等级) (默认值)。
 - <3> 使用低电压检测等级选择寄存器 (LVIS) 的第 3 位到第 0 位 (LVIS3 到 LVIS0) 来设置检测电压。
 - <4> 设置 LVIM 的第 7 位 (LVION) 为 1 (允许 LVI 的操作)。
 - <5> 使用软件等待操作稳定时间 (10 μ s (MAX.))。
 - <6> 根据 LVIM 的 0 位 (LVIF) 等待直至检测到 (供电电压 (V_{DD}) \geq 检测电压 (V_{LVI}))。
 - <7> 设置 LVIM 的第 1 位 (LVIMD) 为 1 (检测到电平时产生复位)。

图 16-5 显示了由低电压检测器产生的内部复位信号的时序。该时序图中的编号与上面的<1> 到 <7>对应。

- 注意事项**
1. 必须执行过程<1>。当 LVIMK = 0 时, 在过程<4>之后可能立即会产生中断。
 2. 当 LVIMD 被设为 1 时, 如果供电电压 (V_{DD}) \geq 检测电压 (V_{LVI}), 则不会产生内部复位信号。

- 当停止操作时
必须执行以下过程之一。
 - 当使用 8 位存储器操作指令时:
写 00H 到 LVIM 中。
 - 当使用 1 位存储器操作指令时:
依次将 LVIMD 和 LVION 清零。

图 16-5. 低电压检测器内部复位信号产生时序
(检测供电电压 (V_{DD}) 的等级)



- 注
1. 由复位信号产生将 LVIMK 标志设置为“1”。
 2. 可以设置 LVIF 标志为 (1)。
 3. LVIRF 位为复位控制寄存器 (RESF) 的位 0。关于 RESF 的详细信息，参见第十四章 复位功能。

备注 上图 16-5 中的 <1> 到 <7> 与 16.4.1 (1) 当检测供电电压 (V_{DD}) 等级时中“当启动操作时”的描述中的 <1> 到 <7> 相对应。

(2) 当检测外部输入引脚 (EXLVI) 的输入电压等级时

- 启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <2> 将低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 设为 1 (检测外部输入引脚 (EXLVI) 的输入电压等级)。
 - <3> 设置 LVIM 的第 7 位 (LVION) 为 1 (允许 LVI 的操作)。
 - <4> 使用软件等待操作稳定时间 (10 μ s (MAX.))。
 - <5> 根据 LVIM 的 0 位 (LVIF) 等待直至检测到 (外部输入引脚的输入电压 (EXLVI) \geq 检测电压 ($V_{EXLVI} = 1.21$ V (TYP.)))
 - <6> 设置 LVIM 的第 1 位 (LVIMD) 为 1 (检测到电平时产生复位信号)。

图 16-6 显示了由低电压检测器产生的内部复位信号的时序。该时序图中的编号与上面的<1>到<6>对应。

- 注意事项**
1. 必须执行过程<1>。当 LVIMK = 0 时，在过程<3>之后可能立即会产生中断。
 2. 如果在 LVIMD 被设为 1 时外部输入引脚 (EXLVI) 的输入电压 \geq 检测电压 ($V_{EXLVI} = 1.21$ V (TYP.))，则不会产生内部复位信号。
 3. 外部输入引脚 (EXLVI) 的输入电压必须是 $EXLVI \leq V_{DD}$ 。

- 当停止操作时
必须执行以下过程之一。
 - 当使用 8 位存储器操作指令时：
写 00H 到 LVIM 中。
 - 当使用 1 位存储器操作指令时：
依次将 LVIMD 和 LVION 清零。

16.4.2 用作中断时

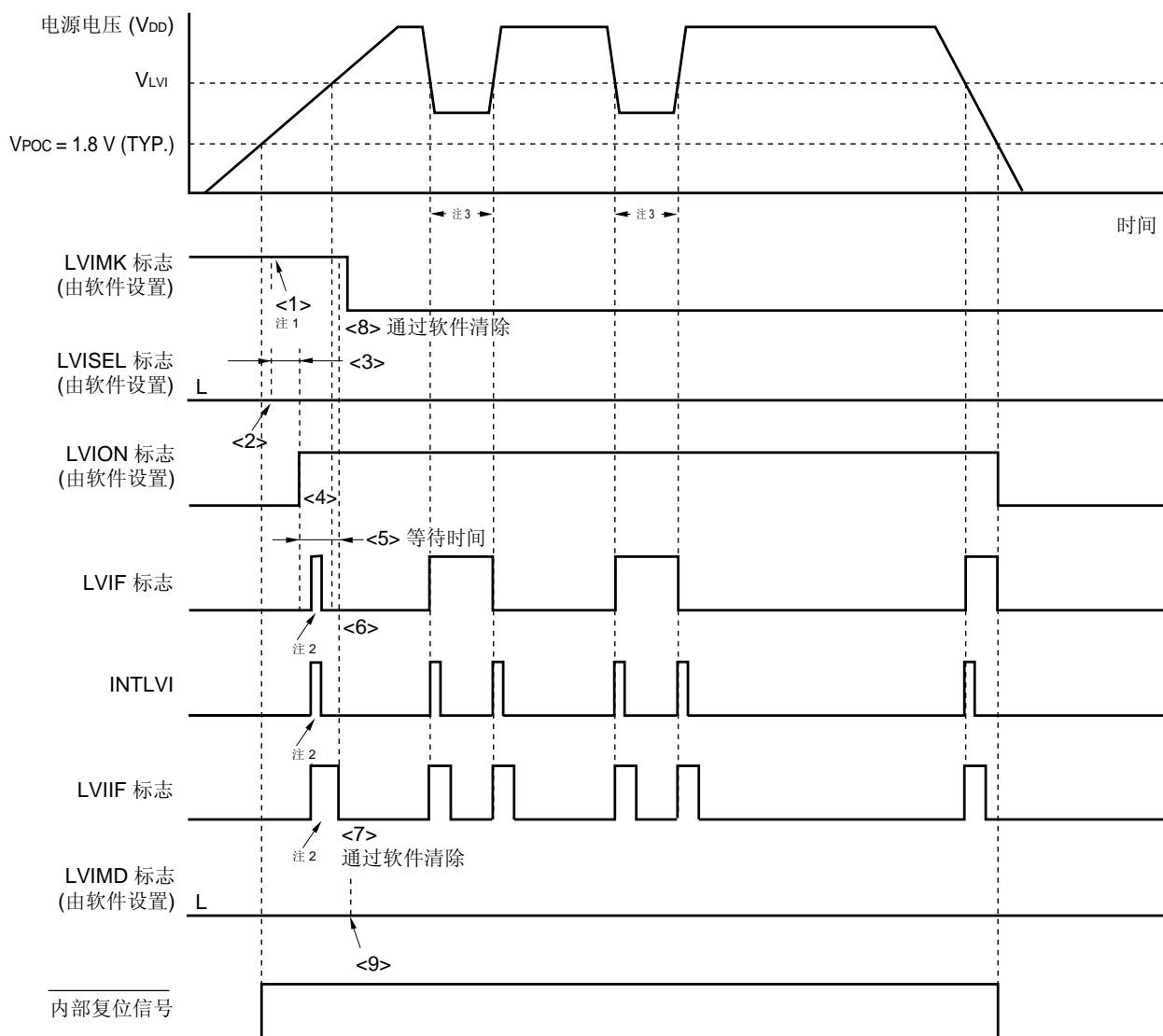
(1) 当检测供电电压 (V_{DD}) 的等级时

- 启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <2> 将低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 清零 (检测供电电压 (V_{DD}) 的等级) (默认值)。
 - <3> 使用低电压检测等级选择寄存器 (LVIS) 的第 3 位到第 0 位 (LVIS3 到 LVIS0) 来设置检测电压。
 - <4> 设置 LVIM 的第 7 位 (LVION) 为 1 (允许 LVI 的操作)。
 - <5> 使用软件等待操作稳定时间 (10 μ s (MAX.))。
 - <6> 在 LVIM 的 0 位 (LVIF) 处检测到 V_{DD} 的下降沿时, 应确认 “供电电压 (V_{DD}) \geq 检测电压 (V_{LVI})”, 在 LVIM 的 0 位 (LVIF) 处检测到 V_{DD} 的上升沿时, 应确认 “供电电压 (V_{DD}) < 检测电压 (V_{LVI})”。
 - <7> 将 LVI 的中断请求标志 (LVIF) 清零。
 - <8> 释放 LVI 的中断屏蔽标志 (LVIMK)。
 - <9> 将 LVIM 的第 1 位 (LVIMD) 清零 (当检测到电平时产生中断信号) (默认值)。
 - <10> 执行 EI 指令 (当使用向量中断时)。

图 16-7 显示了由低电压检测器产生的中断信号的时序。该时序图中的编号与上面的<1> 到 <9>对应。

- 当停止操作时
必须执行以下过程之一。
 - 当使用 8 位存储器操作指令时:
写 00H 到 LVIM 中。
 - 当使用 1 位存储器操作指令时:
将 LVION 清除为 0。

图 16-7. 低电压检测器中断信号产生时序
(检测供电电压 (V_{DD}) 的等级)



- 注
1. 由复位信号产生将 LVIMK 标志设置为“1”。
 2. 产生中断请求信号 (INTLVI)，且可以设置 LVIF 和 LVIIF 标志为 (1)。
 3. 如果 LVION 在低于 LVI 检测电压的状态下被清零，那么将会生成 INTLVI 信号且 LVIIF 会被设为 1。

备注 上图 16-7 中的<1> 到 <9>与 16.4.2 (1) 当检测供电电压 (V_{DD}) 等级时中“当启动操作时”的描述中的<1>到<9>相对应。

(2) 当检测外部输入引脚 (EXLVI) 的输入电压等级时

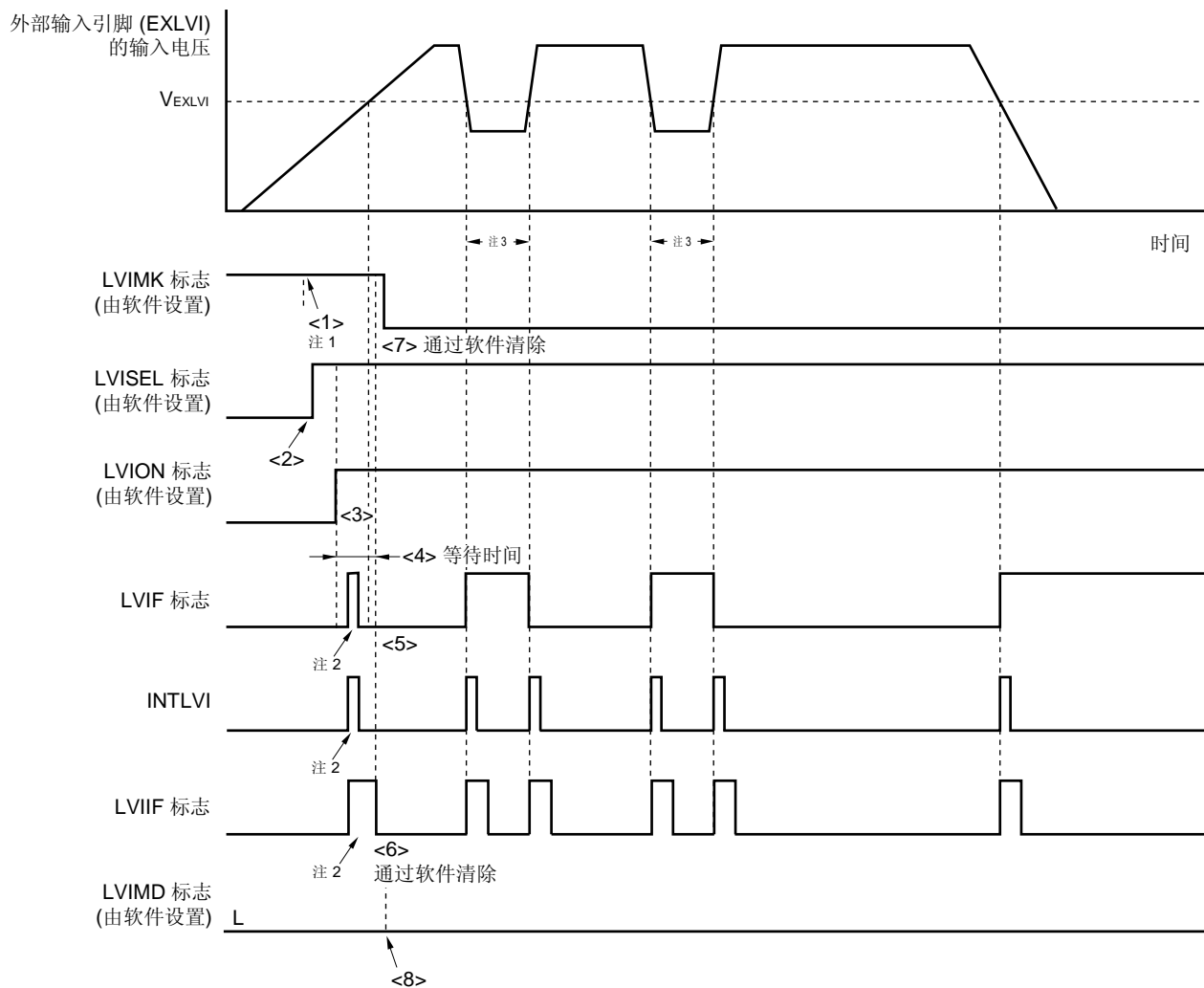
- 启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <2> 将低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 设为 1 (检测外部输入引脚 (EXLVI) 的输入电压等级)。
 - <3> 设置 LVIM 的第 7 位 (LVION) 为 1 (允许 LVI 的操作)。
 - <4> 使用软件等待操作稳定时间 (10 μ s (MAX.))。
 - <5> 在 LVIM 的 0 位 (LVIF) 处检测到 EXLVI 的下降沿时, 应确认“外部输入引脚 (EXLVI) 的输入电压 \geq 检测电压 ($V_{EXLVI} = 1.21$ V (TYP.))”, 在 LVIM 的 0 位 (LVIF) 处检测到 EXLVI 的上升沿时, 应确认“外部输入引脚 (EXLVI) 的输入电压 $<$ 检测电压 ($V_{EXLVI} = 1.21$ V (TYP.))”。
 - <6> 将 LVI 的中断请求标志 (LVIF) 清零。
 - <7> 释放 LVI 的中断屏蔽标志 (LVIMK)。
 - <8> 将 LVIM 的第 1 位 (LVIMD) 清零 (当检测到电平时产生中断信号) (默认值)。
 - <9> 执行 EI 指令 (当使用向量中断时)。

图 16-8 显示了由低电压检测器产生的中断信号的时序。该时序图中的编号与上面的<1> 到 <8>对应。

注意事项 外部输入引脚 (EXLVI) 的输入电压必须是 $EXLVI \leq V_{DD}$ 。

- 当停止操作时
 - 必须执行以下过程之一。
 - 当使用 8 位存储器操作指令时:
 - 写 00H 到 LVIM 中。
 - 当使用 1 位存储器操作指令时:
 - 将 LVION 清除为 0。

图 16-8. 低电压检测器中断信号产生时序
(检测外部输入引脚 (EXLVI) 的输入电压等级)



- 注
1. 由复位信号产生将 LVIMK 标志设置为“1”。
 2. 产生中断请求信号 (INTLVI)，且可以设置 LVIF 和 LVIIF 标志为 (1)。
 3. 如果 LVION 在低于 LVI 检测电压的状态下被清零，那么将会生成 INTLVI 信号且 LVIIF 会被设为 1。

备注 上图 16-8 中的<1> 到 <8>与 16.4.2 (1) 当检测供电电压 (V_{DD}) 等级时中“当启动操作时”的描述中的<1> 到 <8>相对应。

16.5 低电压检测器的注意事项

在系统中，如果供电电压（V_{DD}）在接近 LVI 检测电压（V_{LVI}）的一段时间内产生波动，这时根据低电压检测器的使用情况进行如下操作。

(1) 用作复位时

系统可能会反复进行复位并从复位状态中释放。

在这种情况下，可采用下面的方法（1）任意设置从复位释放到微控制器的启动所经历的时间。

(2) 用作中断时

中断请求可能会频繁产生。可采用以下方法（2）中的（b）。

<方法>

(1) 用作复位时

在释放复位信号后，通过使用定时器的软件计数器等待各个系统供电电压的波动期，然后对端口进行初始化（参见图 16-9）。

(2) 用作中断时

(a) 通过使用低电压检测寄存器（LVIM）的 0 位（LVIF）的 LVI 中断的服务程序中，在检测到 V_{DD} 的下降沿时，应确认“供电电压（V_{DD}）≥ 检测电压（V_{LVI}）”，在检测到 V_{DD} 的上升沿时，应确认“供电电压（V_{DD}）< 检测电压（V_{LVI}）”。把中断请求标志寄存器 0L（IFOL）的第 0 位（LVIIF）清零。

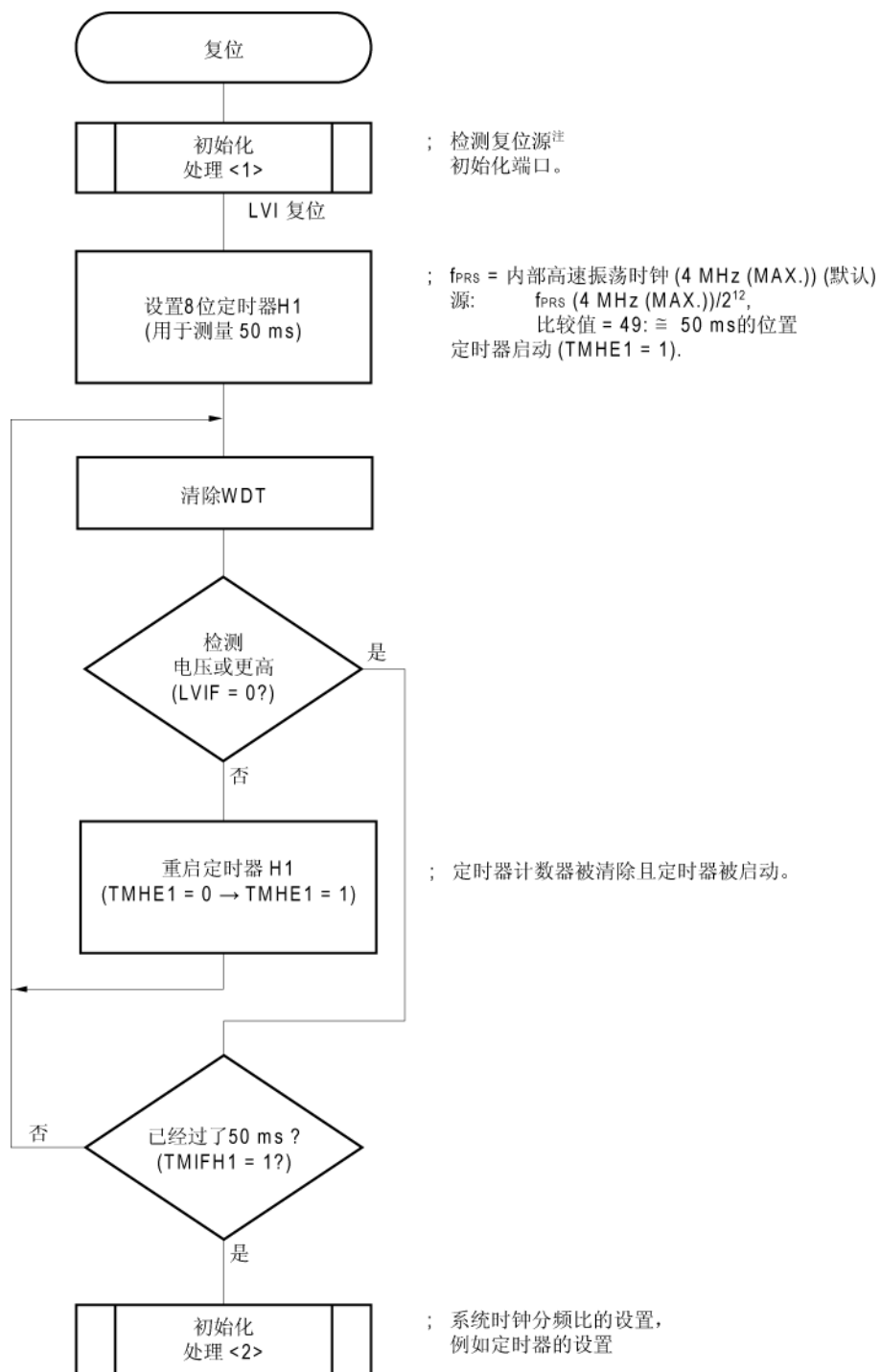
(b) 供电电压在接近 LVI 检测电压的一段时间内产生波动，且波动时间较长的系统中，应通过使用 LVIF 标志等待供电电压波动时间，并确认在检测到 V_{DD} 的下降沿时“供电电压（V_{DD}）≥ 检测电压（V_{LVI}）”，在检测到 V_{DD} 的上升沿时“供电电压（V_{DD}）< 检测电压（V_{LVI}）”，然后把 LVIIF 标志清零。

备注 如果低电压检测寄存器（LVIM）的第 2 位（LVISEL）被设为“1”，那么上述文字的意思将变为如下所示。

- 供电电压（V_{DD}） → 外部输入引脚（EXLVI）的输入电压
- 检测电压（V_{LVI}） → 检测电压（V_{EXLVI} = 1.21 V（TYP.））

图 16-9. 复位释放后软件处理过程示例 (1/2)

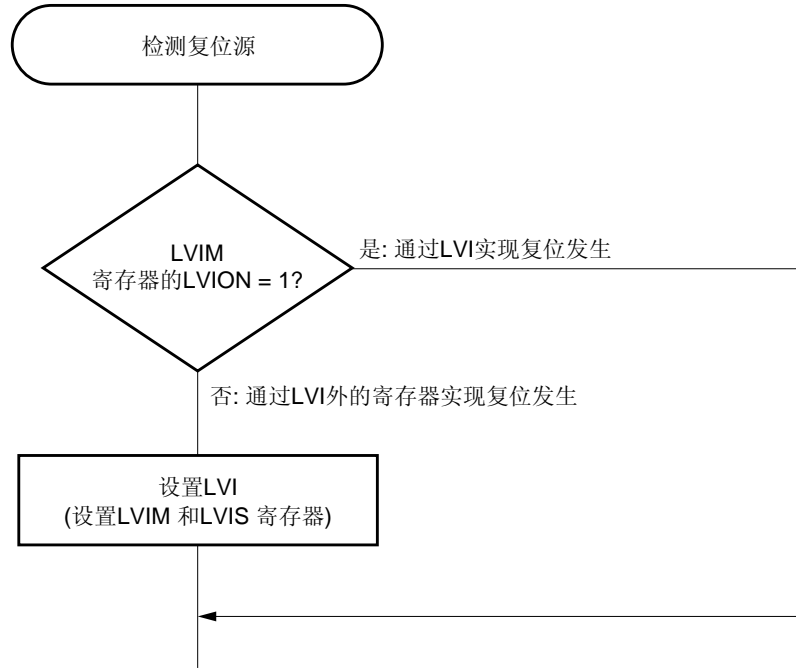
- 如果供电电压接近 LVI 检测电压，其波动时间为 50 ms 或更短



注 流程图显示在下页。

图 16-9. 复位释放后软件处理过程示例 (2/2)

• 检测复位源



16.6 RAM 数据保存检测器

通过使用该电路，在电池变更时可以判断供电电压（ V_{DD} ）是否下降到不能保存 RAM 中数据值的电压以下。RAM 数据保存检测电压（ V_{LD} ）为 $1.4V \pm 0.1V$ 。

RAM 数据保存检测器使用以下寄存器进行控制。

- RAM 数据保存控制寄存器（LVDET）

(1) RAM 数据保存控制寄存器（LVDET）

该寄存器用于设置 RAM 数据保存检测和操作模式。

该寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。

图 16-10. RAM 数据保存控制寄存器（LVDET）的格式

地址：FFB0H 复位后：未定义* R/W

符号	7	6	5	4	3	2	1	0
LVDET	0	0	0	0	0	0	LVDET0	0

LVDET0	RAM数据保存标志
0	RAM 数据可能会变化
1	RAM 数据被保存

注 当 V_{DD} 低于 V_{LD} （ $1.4V$ （Typ.））时，值变为“00H”。其他情况下 LVDET 的值将会被保存。当使用 RAM 数据保存检测器功能时，请将 LVDET0 设为“1”。

当供电电压（ V_{DD} ）下降到 RAM 数据保存检测电压（ V_{LD} ： $1.4\pm 0.1V$ ）以下时，RAM 数据保存标志将会被清零。复位后可以通过检测 RAM 数据保存标志（LVDET0）来判断 RAM 数据值是否被保持。

第十七章 选项字节

17.1 选项字节的功能

μ PD179F11x, 179F12x 微控制器中 Flash 存储器的 0080H 到 0084H 区域为选项字节区域。当打开电源或从复位状态中重启设备时，设备自动参考选项字节并设置指定的功能。使用该产品时，必须使用选项字节设置以下几项功能。

注意事项 一定要对 00H 到 0081H 以及 0083H 进行设置。

(1) 0080H

- 内部低速振荡器的操作
 - 可以通过软件停止
 - 不能停止
- 看门狗定时器间隔时间的设置
- 看门狗定时器计数器的操作
 - 允许计数器的操作
 - 禁止计数器的操作
- 看门狗定时器窗口打开周期的设置

(2) 0084H

- 片上调试操作控制
 - 禁止片上调试操作
 - 一旦片上调试安全 ID 失效，将会允许片上调试操作并擦除 Flash 存储器的数据。
 - 一旦片上调试安全 ID 失效，将会允许片上调试操作但不会擦除 Flash 存储器的数据。

17.2 选项字节的格式

选项字节格式如下所示。

图 17-1. 选项字节的格式 (1/2)

地址: 0080H

7	6	5	4	3	2	1	0
0	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	LSROSC
WINDOW1	WINDOW0	看门狗定时器窗口打开周期					
0	0	25%					
0	1	50%					
1	0	75%					
1	1	100%					
WDTON	看门狗定时器计数器的操作控制 / 非法访问检测						
0	禁止计数器操作 (复位后停止计数), 禁止非法访问检测操作						
1	允许计数器操作 (复位后开始计数), 允许非法访问检测操作						
WDCS2	WDCS1	WDCS0	看门狗定时器溢出时间				
0	0	0	$2^{10}/f_{RL}$ (3.41 ms)				
0	0	1	$2^{11}/f_{RL}$ (6.83 ms)				
0	1	0	$2^{12}/f_{RL}$ (13.65 ms)				
0	1	1	$2^{13}/f_{RL}$ (27.31 ms)				
1	0	0	$2^{14}/f_{RL}$ (54.61 ms)				
1	0	1	$2^{15}/f_{RL}$ (109.23 ms)				
1	1	0	$2^{16}/f_{RL}$ (218.45 ms)				
1	1	1	$2^{17}/f_{RL}$ (436.91 ms)				
LSROSC	内部低速振荡器的操作						
0	可由软件停止 (当将 1 写入 RCM 寄存器的第 1 位 (LSRSTOP) 时停止操作)						
1	不能停止 (即使将 1 写入 LSRSTOP 位也不停止操作)						

<R>

- 注意事项
1. 禁止出现 $WDCS2 = WDCS1 = WDCS0 = 0$ 且 $WINDOW1 = WINDOW0 = 0$ 的情况。
 2. 在 flash 存储器的自编程过程中看门狗定时器会继续其操作操作。在处理过程中中断响应时间将会被延迟。设置溢出时间和窗口大小时应将延迟情况考虑在内。
 3. 如果 $LSROSC = 0$ (可由软件停止振荡), 那么不管内部振荡模式寄存器 (RCM) 的第 1 位 (LSRSTOP) 为何值, 在 HALT 和 STOP 模式中计数数据均不会提供给看门狗定时器。
当 8 位定时器 H1 使用内部低速振荡时钟时, 即使在 HALT/STOP 模式下计数时钟也会提供给 8 位定时器 H1。
 4. 确保将位 7 清除为 "0"。

<R>

- 备注
1. f_{RL} : 内部低速振荡时钟频率
 2. (): $f_{RL} = 300 \text{ kHz (MAX.)}$

<R>

图 17-1. 选项字节的格式 (2/2)

地址: 0081H 到 0083H[#]

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

注 由于这些地址为保留区域，因此必须将 0081H 以及 0083H 设为 00H。

地址: 0084H[#]

7	6	5	4	3	2	1	0
0	0	0	0	0	0	OCDEN1	OCDEN0

OCDEN1	OCDEN0	片上调试操作控制
0	0	禁止操作
0	1	禁止设置
1	0	操作允许。一旦片上调试安全 ID 失效也不会擦除 Flash 存储器的数据。
1	1	操作允许。一旦片上调试安全 ID 失效将会擦除 Flash 存储器的数据。

注 要使用片上调试功能，则需要将 0084H 设为 02H 或 03H。

备注 关于片上调试安全 ID 的详细信息，请参见第十九章 片上调试功能。

此处为设置选项字节的软件描述示例。

```

OPT      CSEG  AT 0080H
OPTION:  DB    30H      ; 允许看门狗定时器的操作（非法访问检测操作），
                       ; 看门狗定时器窗口打开周期：50%，
                       ; 看门狗定时器的溢出时间：210/fRL，
                       ; 可由软件停止内部低速振荡器
          DB    00H      ; 保留区域
          DB    00H      ; 保留区域
          DB    00H      ; 禁止片上调试操作

```

备注 复位处理期间参考选项字节。关于复位处理时序的详细信息，请参见第十四章 复位功能。

第十八章 FLASH存储器

μ PD179F11x, 179F12x 微控制器中包含了 Flash 存储器, Flash 存储器安装在电路板上时可以实现程序的写入, 擦除以及覆盖操作。

18.1 内部存储器容量切换寄存器

可以使用内部存储器容量切换寄存器 (IMS) 选择内部存储器的容量。

IMS 可以通过 8 位存储器操作指令来设置。

复位信号生成将 IMS 设为 CFH。

注意事项 复位释放后必须按照表 18-1 所示来设置每个产品的值。

图 18-1. 内部存储器尺寸切换寄存器 (IMS) 的格式

地址: FFF0H 复位后: 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 容量选择
0	0	0	768 个字节
0	1	0	512 个字节
1	1	0	1024 个字节
其他情况			禁止设置

ROM3	ROM2	ROM1	ROM0	内部 ROM 容量选择
0	0	0	1	4 KB
0	0	1	0	8 KB
0	1	0	0	16 KB
0	1	1	0	24 KB
1	0	0	0	32 KB
其他情况				禁止设置

表 18-1. 内部存储器容量切换寄存器的设置

Flash 存储器版本 (μ PD179F11x, 179F12x 微控制器)	IMS 设置
μ PD179F110	41H
μ PD179F111	42H
μ PD179F112, 179F122	04H
μ PD179F113, 179F123	C6H
μ PD179F114, 179F124	C8H

18.2 用Flash存储器编程器写入数据

可使用专用 Flash 存储器编程器将数据在线写入或离线写入 Flash 存储器。

(1) 在线编程

当目标系统上安装了 μ PD179F11x, 179F12x 微控制器后, Flash 存储器的内容可以被重写。必须将连接专用 Flash 存储器编程器的连接器安装在目标系统中。

(2) 离线编程

在将 μ PD179F11x, 179F12x 微控制器安装到目标系统之前, 可使用专用程序适配器 (FA 系列) 将数据写入 Flash 存储器。

备注 FA 系列是 Naito Densai Machida Mfg. Co., Ltd 的产品。

表 18-2. μ PD179F11x 微控制器和专用 Flash 存储器编程器之间的配线示例

专用 Flash 存储器编程器的引脚配置			μ PD179F11x 微控制器	
信号名称	I/O	引脚功能	引脚名称	引脚编号
SI/RxD	输入	接收信号	TxD6/P13	29
SO/TxD	输出	发送信号	RxD6/P14	28
SCK	输出	发送时钟	—	—
CLK	输出	μ PD179F11x, 179F12x 微控制器的时钟	注	注
/RESET	输出	复位信号	$\overline{\text{RESET}}$	6
FLMD0	输出	模式信号	FLMD0	9
V _{DD}	I/O	V _{DD} 电压生成 / 电源检测	V _{DD}	14
			AV _{REF}	33
GND	—	接地	V _{SS}	13
			AV _{SS}	34

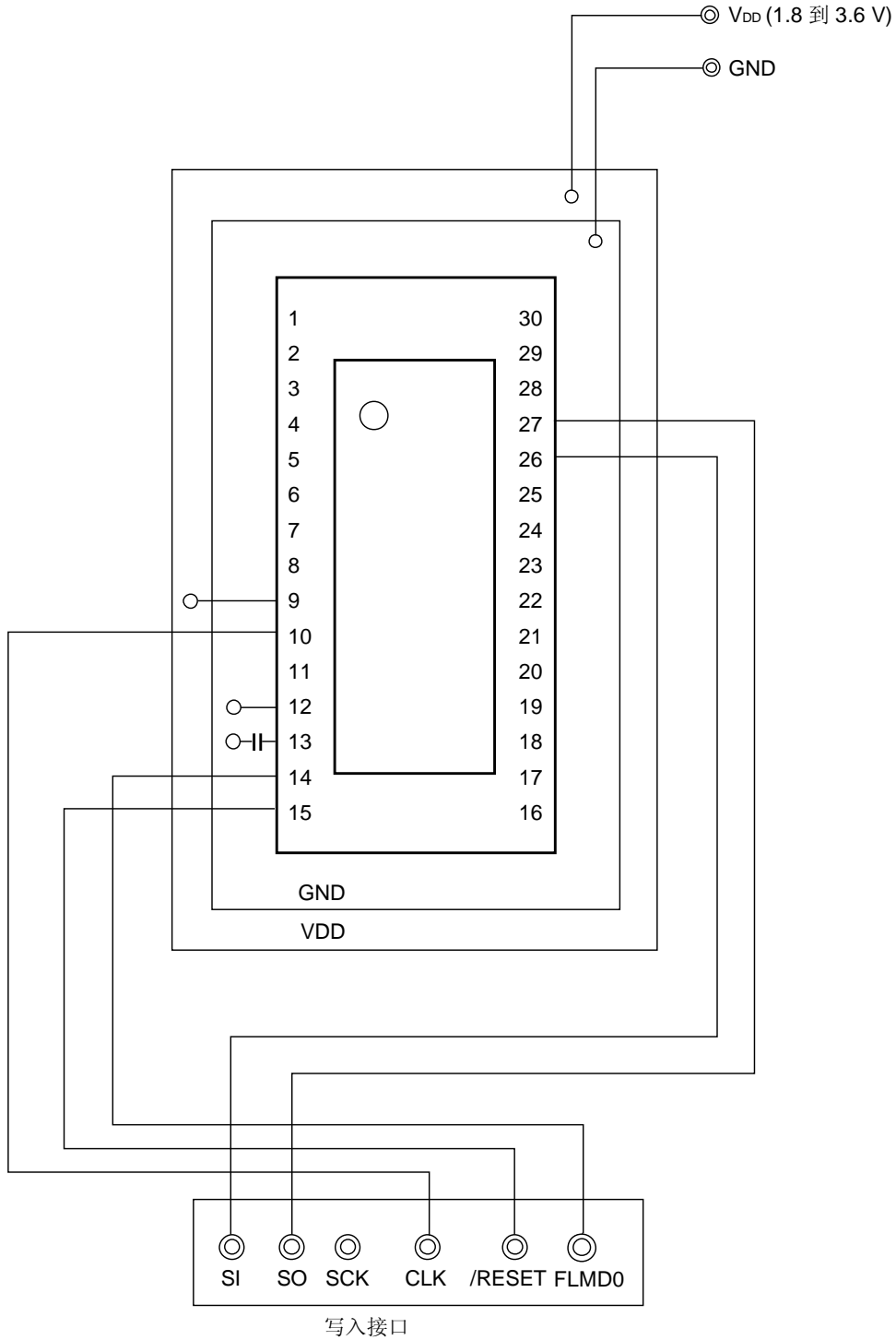
注 只能使用 X1 时钟 (fx) 或外部主系统时钟 (fEXCLK)。使用专用 Flash 存储器编程器的时钟输出时, 引脚连接依据所用的专用 Flash 存储器编程器的类型而变化。

<R>

- PG-FP4, FL-PR4, PG-FP5, FL-PR5: 将编程器的 CLK 连接到 EXCLK/X2/P122/OCD0B (引脚 10)。

使用用于 Flash 存储器写入的适配器时推荐的连接示例如下所示。

图 18-2. 用于 Flash 存储器写入的适配器的配线示例（30 引脚产品）



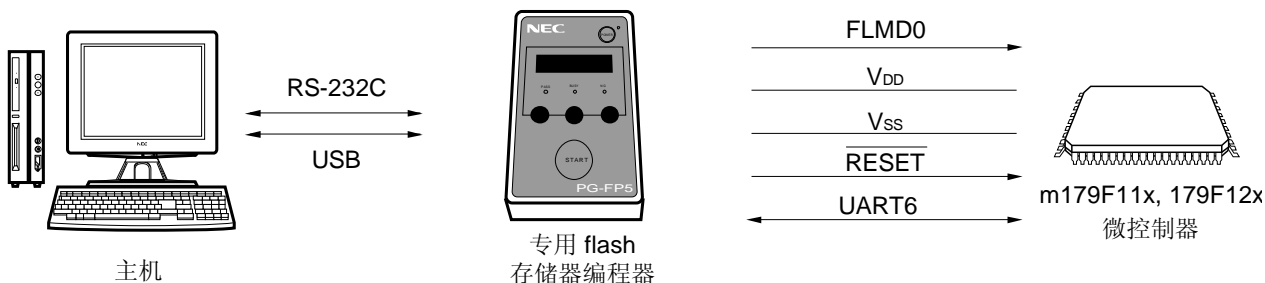
<R> **备注** 上图说明了使用 PG-FP4, FL-PR4, PG-FP5 或 FL-PR5 的时钟输出时的配线示例。

18.3 编程环境

对 μ PD179F11x, 179F12x 微控制器的 Flash 存储器写入程序时所需的环境如下图所示。

<R>

图 18-3. 写程序到 Flash 存储器所需要的环境



需要有一个控制专用 Flash 存储器编程器的主机。

对于专用 Flash 存储器编程器与 μ PD179F11x, 179F12x 微控制器间的端口来说, CSI10 或 UART6 是用于进行写入和擦除等操作的。要对 Flash 存储器进行离线写操作则需要一个专用程序适配器 (FA 系列)。

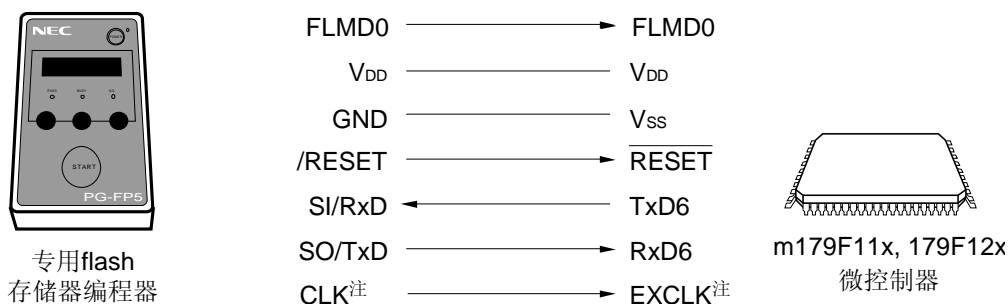
18.4 通信模式

通过 μ PD179F11x, 179F12x 微控制器的串行通信来建立专用 Flash 存储器编程器与 μ PD179F11x, 179F12x 微控制器之间的通信。

<R>

图 18-4. 与专用 Flash 存储器编程器的通信

传输速率: 115200 bps



注 上图说明了使用 PG-FP4, FL-PR4, PG-FP5 或 FL-PR5 的时钟输出时的配线示例。

专用 Flash 存储器编程器为 μ PD179F11x, 179F12x 微控制器生成以下信号。详细信息, 请参考 PG-FP4 或 FL-PR4 的用户手册。

表 18-3. 引脚连接

专用 Flash 存储器编程器			μ PD179F11x, 179F12x 微控制器	连接
信号名称	I/O	引脚功能	引脚名称	
FLMD0	输出	模式信号	FLMD0	◎
V _{DD}	I/O	V _{DD} 电压产生 / 电源检测	V _{DD} , AV _{REF}	◎
GND	-	接地	V _{SS} , AV _{SS}	◎
CLK	输出	μ PD179F11x, 179F12x 微控制器的时钟输出	Note 1	○ ^{‡1}
/RESET	输出	复位信号	RESET	◎
SI/RxD	输入	接收信号	SO10/TxD6	◎
SO/TxD	输出	发送信号	SI10/RxD6	◎
SCK	输出	传输时钟	SCK10	×

注 只能使用 X1 时钟 (fx) 或外部主系统时钟 (fEXCLK)。使用专用 Flash 存储器编程器的时钟输出时, 引脚连接依据所用的专用 Flash 存储器编程器的类型而变化。

<R> • PG-FP4, FL-PR4, PG-FP5, FL-PR5: 将编程器的 CLK 连接到 EXCLK/X2/P122。

备注

- ◎: 确保连接引脚。
- : 如果在目标板上产生信号, 则此引脚不需要连接。
- ×: 不需要连接连接。

18.5 在线引脚的处理

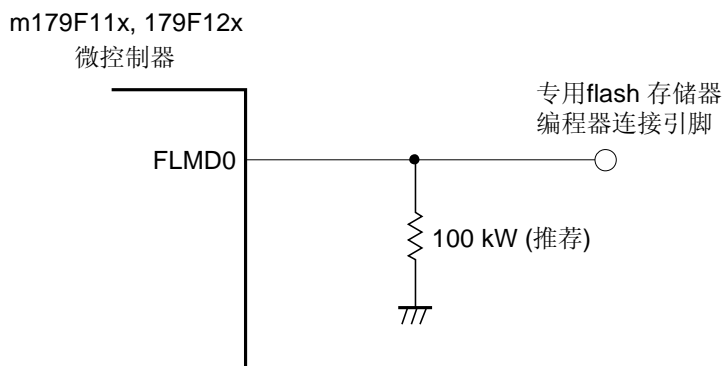
若要对 Flash 存储器进行在线写入, 目标系统必须配有连接专用 Flash 存储器编程器的连接器。电路板上首先要提供一个用于选择正常操作模式或 Flash 存储器编程模式的功能。

当设置 Flash 存储器编程模式时, 那些不用于 Flash 存储器编程的引脚状态与复位后的状态相同。因此, 如果外部设备不能立即识别复位后的状态, 则必须采用如下方式连接引脚。

18.5.1 FLMD0 引脚

在普通操作模式下, 向 FLMD0 引脚输入 0V 电压。在 Flash 存储器编程模式中, 将 V_{DD} 写入电压提供给 FLMD0 引脚。FLMD0 引脚的连接示意图如下所示。

图 18-5. FLMD0 引脚连接示例



18.5.2 串行接口引脚

各串行接口使用的引脚如下所示。

表 18-4. 各串行接口使用的引脚

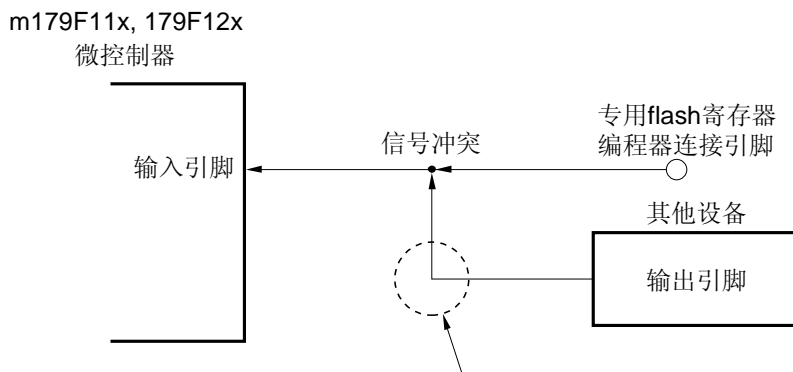
串行接口	使用引脚
UART6	TxD6, RxD6

若要将专用 Flash 存储器编程器与连接到电路板上另一个设备的串行接口的引脚相连接，则必须注意信号之间不能冲突并且其他设备不能出现误操作。

(1) 信号冲突

如果专用 Flash 存储器编程器（输出）与连接到另一个设备（输出）的串行接口的引脚（输入）相连接，那么将会产生信号冲突。为了避免这种冲突，应隔离与其他设备的连接，或者使其他设备处于输出高阻抗状态。

图 18-6. 信号冲突（串行接口的输入引脚）

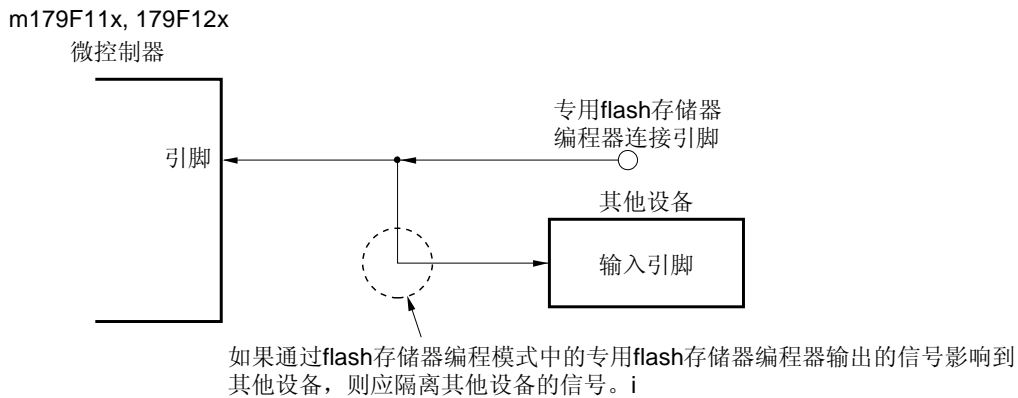
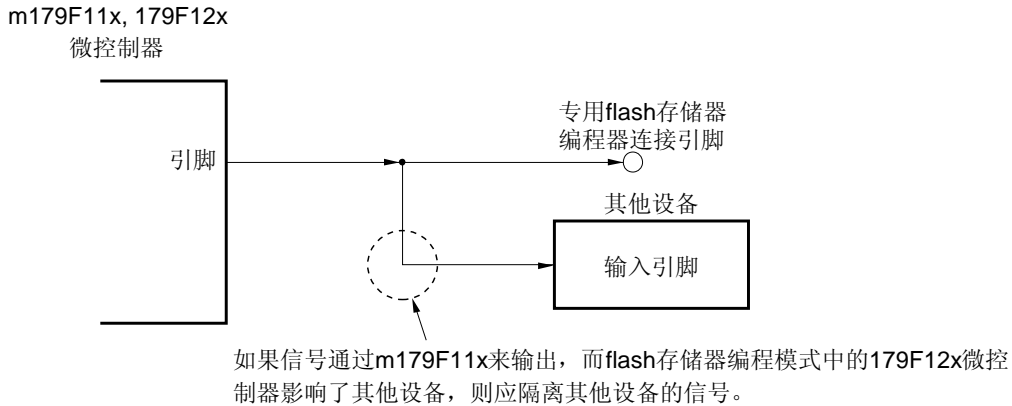


在flash存储器编程模式中，由其他设备输出的信号与专用flash存储器编程器中发出的信号冲突。因此，应隔离其他设备的信号。

(2) 其他设备故障

如果专用 Flash 存储器编程器（输出或输入）与连接到另一个设备（输入）上的串行接口引脚（输入或输出）相连，则信号可能会输出到其他设备中，从而引起该设备的故障。为了避免这种故障，应隔离与其他设备的连接。

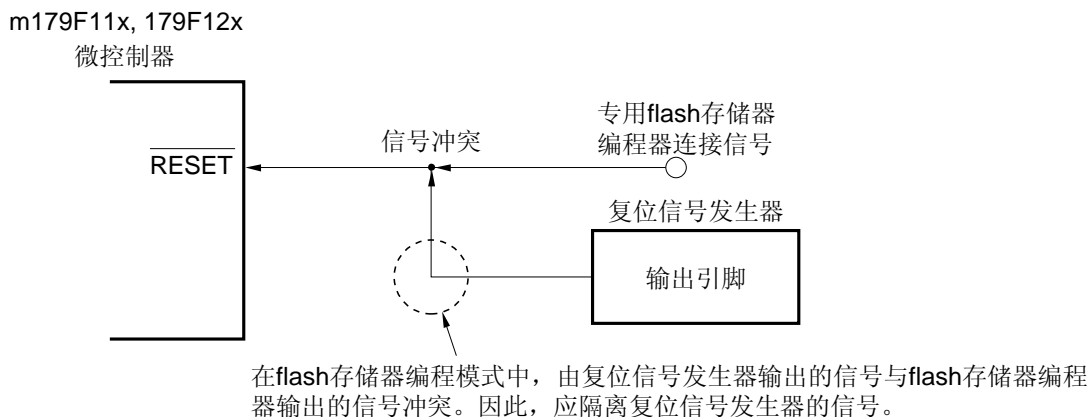
图 18-7. 其他设备故障



18.5.3 $\overline{\text{RESET}}$ 引脚

如果专用 Flash 存储器编程器的复位信号与连接到电路板上复位信号发生器的 $\overline{\text{RESET}}$ 引脚相连接，则将会发生信号冲突。为了避免这种冲突，应隔离与复位信号发生器的连接。

如果在设置 Flash 存储器编程模式时复位信号从用户系统中输入，那么 Flash 存储器将不能正确编程。因此，除了专用 Flash 存储器编程器的复位信号外，不要输入其它信号。

图 18-8. 信号冲突 ($\overline{\text{RESET}}$ 引脚)

18.5.4 端口引脚

当设置 Flash 存储器编程模式时，那些不用于 Flash 存储器编程的引脚将会立即进入与复位后相同的状态。如果连接到端口的外部设备不能立即识别复位后的端口状态，则必须通过一个电阻将端口引脚连接到 V_{DD} 或 V_{SS}。

18.5.5 REGC 引脚

以正常操作相同的方式，通过一个电容（0.47 μ F：推荐）将 REGC 引脚接地。

18.5.6 其他信号引脚

使用在线时钟时，以与正常操作模式下相同的状态连接 X1 和 X2。

然而，若要从专用 Flash 存储器编程器输入操作，则按下列所示连接。

<R> • PG-FP4, FL-PR4, PG-FP5, FL-PR5: 将编程器的 CLK 连接到 EXCLK/X2/P122/OCD0B。

注意事项 只能使用 X1 时钟（fx）或外部主系统时钟（fEXCLK）。

18.5.7 电源

若要使用 Flash 存储器编程器输出的供电电压，则将 V_{DD} 引脚与 Flash 存储器编程器的 V_{DD} 相连，将 V_{SS} 引脚与 Flash 存储器编程器的 GND 相连。

要使用在线供电电压，要按正常操作模式连接。

然而，即使在使用片上供电电压时，也必须将 V_{DD} 和 V_{SS} 引脚与 Flash 存储器编程器的 V_{DD} 和 GND 相连。

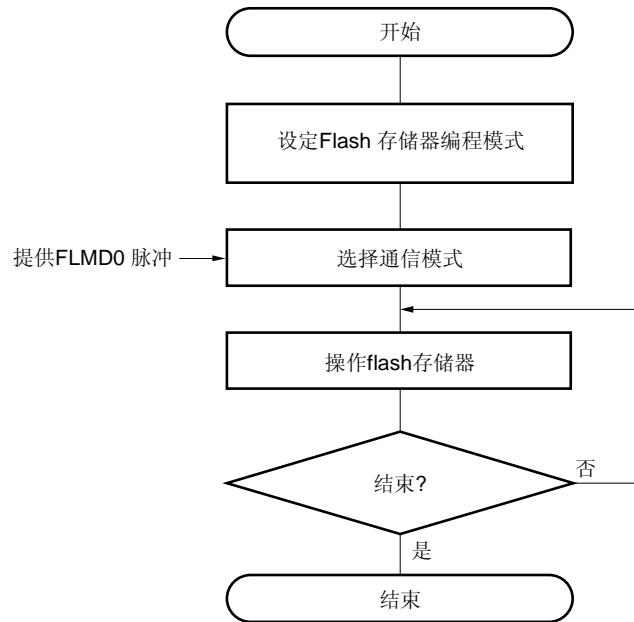
18.6 编程方法

18.6.1 控制Flash 存储器

下图显示了 Flash 存储器的操作过程。

<R>

图 18-9. Flash 存储器操作过程



18.6.2 Flash 存储器编程模式

若要通过使用专用 Flash 存储器编程器来重写 Flash 存储器的内部，则必须在 Flash 存储器编程模式中对 μ PD179F11x, 179F12x 微控制器进行设置。若要设置该模式，则应将 FLMD0 引脚设置到 V_{DD} 并清除复位信号。

在线写入 Flash 存储器时，应通过使用跳线改变模式。

图 18-10. Flash 存储器编程模式

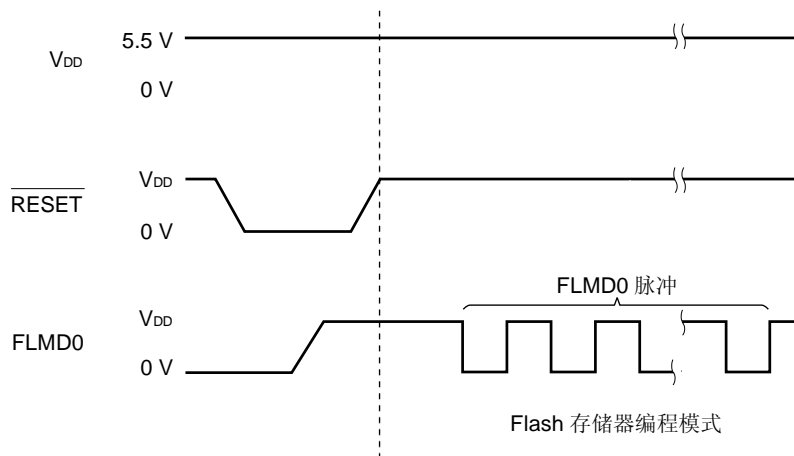


表 18-5. 复位后 FLMD0 引脚与操作模式间的关系

FLMD0	操作模式
0	正常操作模式
V_{DD}	Flash 存储器编程模式

18.6.3 选择通信模式

在 μ PD179F11x, 179F12x 微控制器中, 在进入专用 Flash 存储器编程模式后, 通过将脉冲输入到 FLMD0 引脚来选择通信模式。由 Flash 存储器编程器来生成这些 FLMD0 脉冲。

下表显示了脉冲个数与通信模式之间的关系。

表 18-6. 通信模式

通信模式	标准设置 ^{注1}				使用引脚	外围时钟	FLMD0 脉冲的个数
	端口	速度	频率	倍率			
UART (UART6)	UART-Ext-Osc	115, 200 bps ^{注3}	2 到 4 MHz ^{注2}	1.0	TxD6, RxD6	fx	0
	UART-Ext-FP4CK					fEXCLK	3

- 注
1. Flash 存储器编程器 GUI 标准设置的选择项目。
 2. 可设置范围根据电压的不同而变化。详细信息, 请参见电气特性的章节。
 3. 波特率误差以外的因素, 如信号波形瞬变, 也会影响 UART 通信, 因此必须象测量波特率误差一样严格测量瞬变。

注意事项 在接收到 FLMD0 脉冲后, 会根据专用 Flash 存储器编程器发送的复位命令来计算接收时钟。

备注

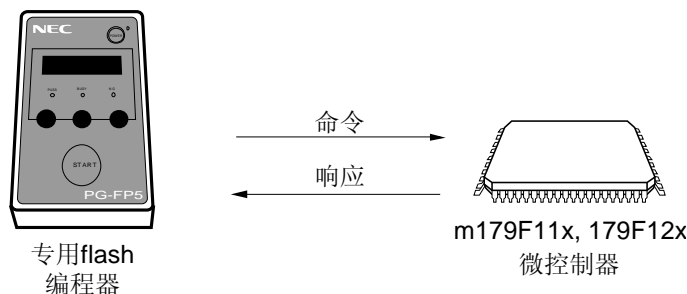
fx: X1 时钟
fEXCLK: 外部主系统时钟
fRH: 内部高速振荡时钟

18.6.4 通信命令

μ PD179F11x, 179F12x 微控制器使用命令与专用 Flash 存储器编程器进行通信。从 Flash 存储器编程器发往 μ PD179F11x, 179F12x 微控制器的信号称为命令, 从 μ PD179F11x, 179F12x 微控制器发往专用 Flash 存储器编程器的信号称为响应。

<R>

图 18-11. 通信命令



μ PD179F11x, 179F12x 微控制器的 Flash 存储器控制命令列于下表中。所有这些命令都从编程器发出, μ PD179F11x, 179F12x 微控制器根据相应的命令进行处理。

表 18-7. Flash 存储器控制命令

类别	命令名称	功能
校验	校验	比较 Flash 存储器中指定区域的内容与从编程器中发出的数据。
擦除	block 擦除	擦除 Flash 区域中的所有区域的内容。
<R> 空白检测	block 空白检测	检测 Flash 存储器中指定 block 是否被正确擦除。
写入	编程	将数据写入 Flash 存储器中的指定区域。
获取信息	状态	获取当前操作状态（状态数据）。
	硅标记	获取设备信息（例如产品代号和 Flash 存储器配置）。
	版本获取	获取设备版本和固件版本。
	校验和	获取指定区域的校验和数据。
安全	安全设置	设置安全信息。
其他	复位	用来检测通信的同步状态。
	振荡频率设置	指定一个振荡频率。

μ PD179F11x, 179F12x 微控制器为专用 Flash 编程器发出的命令返回一个响应。以下列出了 μ PD179F11x, 179F12x 微控制器发出的响应名称。

表 18-8. 响应名称

响应名称	功能
ACK	响应命令 / 数据
NAK	响应非法命令 / 数据

18.7 安全设置

μ PD179F11x, 179F12x 微控制器支持禁止将用户程序写入到内部 Flash 存储器的安全功能，因此程序不能由未经授权的人修改。

以下显示使用安全设定命令能够执行的操作。在设置下一个编程模式时，安全设置有效。

- 禁止批擦除（片擦除）

μ PD179F11x, 179F12x 微控制器禁止的设置。

- 禁止 block 擦除

在线 / 离线编程过程中禁止执行特定 Flash 存储器的 block 擦除命令。然而，block 可以通过自编程方式来擦除。

- 禁止写入

在线 / 离线编程过程中禁止执行对 Flash 存储器中所有 block 的写入命令和 block 擦除命令。然而，block 可以通过自编程方式来写入。

- 禁止重写 block 00H 到 03H

通过该设置可以禁止对 Flash 存储器中 block 00H 到 03H（0000H 到 0FFFH）执行 block 擦除命令和写入命令。

注意事项 如果应用了重写 block 00H 到 03H 的安全设置，该设备的 block 00H 到 03H 将不能被重写。

<R>

装载 Flash 存储器时，在默认设置下允许 block 擦除命令以及写入命令。可以通过在线 / 离线编程来进行安全设置。各项安全设置可以结合使用。

一旦为 μ PD179F11x, 179F12x 微控制器设置了安全禁止设置，该安全设置将不能更改。

表 18-9 显示了允许使用 μ PD179F11x, 179F12x 微控制器的安全功能时，擦除命令与写入命令之间的关系。

表 18-9. 允许使用的安全功能和命令之间的关系

(1) 在线 / 离线编程期间

<R>

有效的安全功能	执行的命令	
	block 擦除 ^注	写入
禁止 block 擦除	block 不能擦除。	可以执行。
禁止写入		不能执行。
禁止重写 block 00H 到 03H		引导簇 0 不能被写入。

注 在线或离线编程模式中，block 擦除命令可以对所有 block 进行一次擦除。

(2) 自编程期间

<R>

有效的安全功能	执行的命令	
	block 擦除 ^注	写入
禁止 block 擦除	block 可以擦除。	可以执行。
禁止写入		
禁止重写 block 00H 到 03H	引导簇 0 不能被擦除。	引导簇 0 不能被写入。

注 自编程模式中 block 擦除命令可以用于以 block（1 KB）为单位进行 block 擦除。

表 18-10 显示了在每一种编程模式下进行安全设置的方法。

表 18-10. 每一种编程模式下的安全设置

(1) 在线 / 离线编程

安全	安全设置	禁止安全设置的方法
禁止 block 擦除	通过专用 Flash 存储器编程器的 GUI 等进行设置。	设置后不能禁止。
禁止写入		
禁止重写 block 00H 到 03H		

(2) 自编程

安全	安全设置	禁止安全设置的方法
禁止 block 擦除	不能设置。	设置后不能禁止。
禁止写入		
禁止重写 block 00H 到 03H		

18.8 通过自编程对Flash存储器进行编程

μ PD179F11x, 179F12x 微控制器支持自编程功能, 该自编程功能可以通过一个用户程序来重写 Flash 存储器。通过使用自编程库该功能可以允许用户应用程序重写 Flash 存储器, 也可以用于升级正在使用的程序。

如果在自编程过程中发生中断, 自编程可以暂时停止并执行中断服务。若要执行中断服务, 则应在自编程停止后恢复正常操作模式, 并执行 EI 指令。在自编程模式被恢复后可以继续执行自编程。

备注 关于自编程功能和自编程库的详细信息, 请参照 78K0 微控制器自编程库类型 01 的用户手册 (U18274E)。

注意事项 1. 在自编程过程中向 FLMD0 引脚输入一个高电平。

(1) 通过外部输入设置

- 将 FPEN0 (FLMD0 上拉/下拉允许寄存器 (FPEN) 位 0) 设为 “0”。
- 向 FLMD0 输入高电平 (100KOhm 下拉)

(2) 通过内部上拉电阻设置

- 通过 100KOhm 下拉 FLMD0 引脚。
- 将 FPEN0 (FLMD0 上拉/下拉允许寄存器 (FPEN) 位 0) 和 FLMDPUP (FLMD0 上拉/下拉控制寄存器 (FPCTL) 位 0) 设为 “1”。通过对其进行设置, 输入 FLMD0 引脚的电平将变为 “高电平”。

图 18-12. 自编程时的 FLMD0 引脚设置

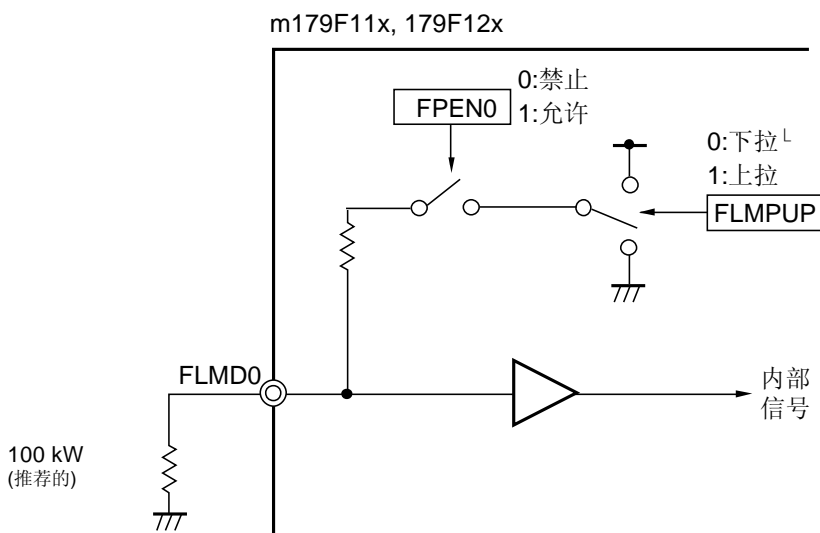


图 18-13. FLMD0 上拉 / 下拉控制寄存器 (FPCTL) 的格式

地址: FF35H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
FPCTL	0	0	0	0	0	0	0	FLMDPUP

FLMDPUP	FLMD0 上拉 / 下拉设置
0	下拉
1	上拉

图 18-14. FLMD0 上拉 / 下拉允许寄存器 (FPEN) 的格式

地址: FF37H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
FPEN	0	0	0	0	0	0	0	FPEN0

FPEN0	FLMD0 上拉 / 下拉允许设置
0	禁止上拉 / 下拉
1	允许上拉 / 下拉

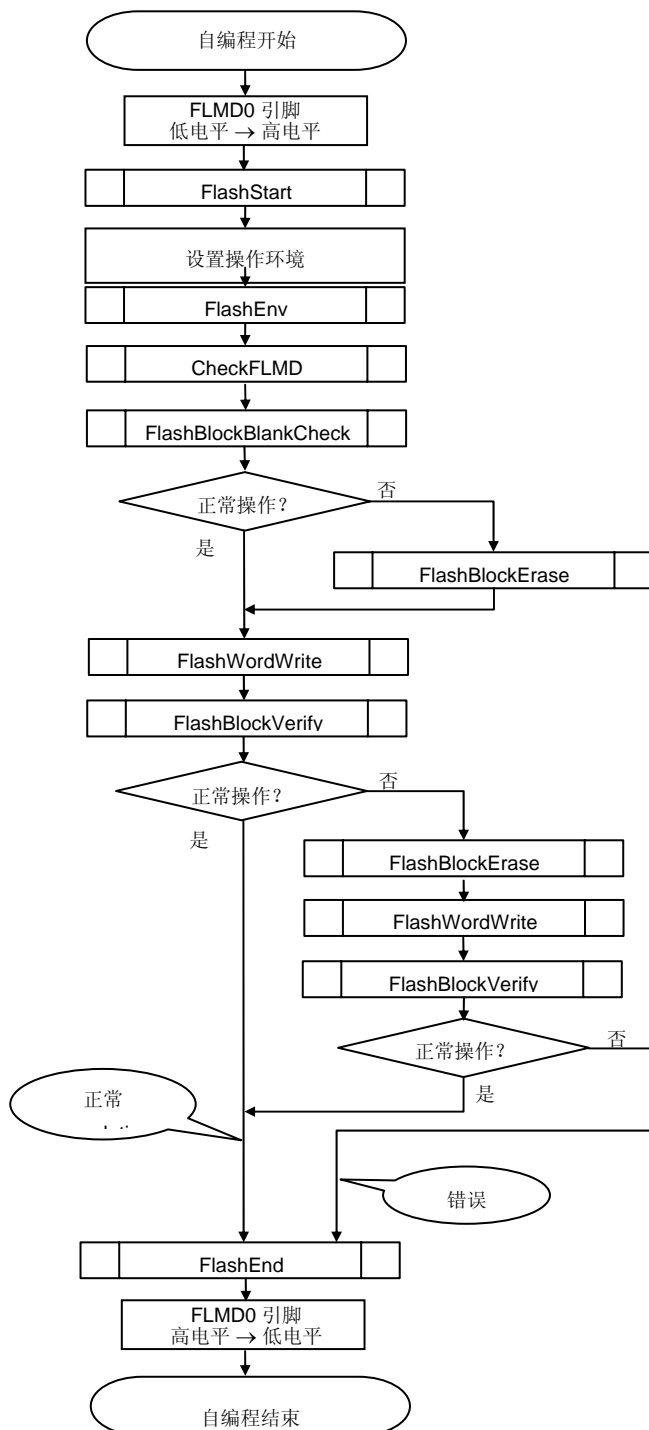
注意事项 2. 在开始自编程前必须执行 DI 指令。

自编程功能可以检测中断请求标志 (IF0L, IF0H, IF1L)。如果生成中断请求, 将会停止自编程。

3. 通过一个未屏蔽的中断请求, 即使是在 DI 状态中, 自编程也会停止。为了避免这种情况, 应通过使用中断屏蔽标志寄存器 (MK0L, MK0H, MK1L) 来屏蔽中断。

下图显示了通过使用自编程库重写 Flash 存储器的流程。

图 18-15. 自编程的流程（重写 Flash 存储器）



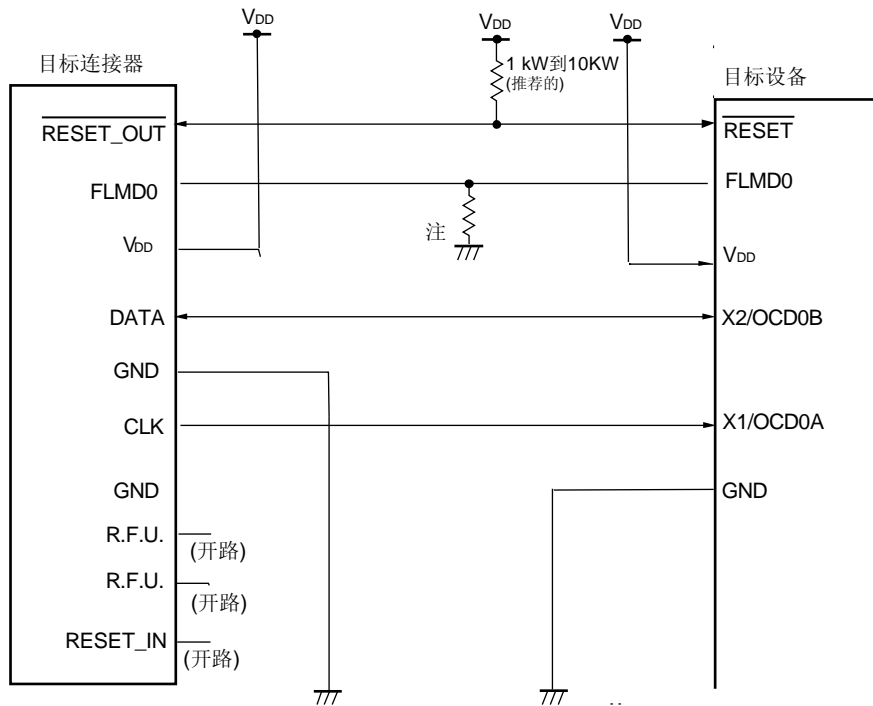
备注 关于自编程库的详细信息，请参见 **78K0 微控制器自编程库类型 01** 的用户手册（U18274E）。

第十九章 片上调试功能

19.1 连接 QB-MINI2

μ PD179F11x, 179F12x 微控制器通过一个片上调试仿真器 (QB-MINI2) 使用 V_{DD} , FLMD0, $\overline{\text{RESET}}$, OCD0A/X1 (或 OCD1A/P04), OCD0B/X2 (或 OCD1B/P05) 以及 V_{SS} 引脚与主机进行通信。可以选择是使用 OCD0A/X1 和 OCD1A/P04, 还是 OCD0B/X2 和 OCD1B/P05。

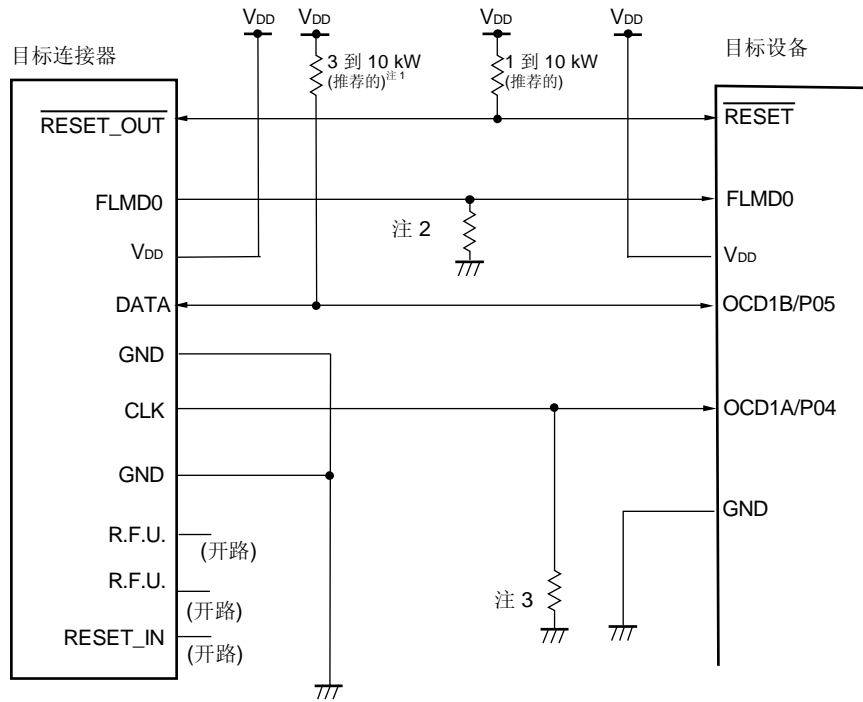
图 19-1. QB-MINI2 和 μ PD179F11x, 179F12x 微控制器连接示例
(当使用 OCD0A/X1 和 OCD0B/X2 时)



注 使下拉电阻的大小等于 100 k Ω 。

- 注意事项
1. 在片上调试过程中从 OCD0A/X1 引脚中输入时钟。
 2. 当 OCD0A/X1 和 OCD0B/X2 用于 OCD 时，在复位信号必须为固定电平（高电平或低电平）期间 OCD1A/P04 和 OCD1B/P05 的电压电平。
 3. 由于 $\overline{\text{RESET}}$ 引脚是通过 QB-MINI2 来使用的，因此复用功能不能用于仿真。

图 19-2. QB-MINI2 和 μ PD179F11x, 179F12x 微控制器连接示例
(当使用 OCD1A/P04 和 OCD1B/P05 时)



- 注
1. 这是 OCD1B/P05 被设置为输入端口（为了在不连接 QB-MINI2 时防止此引脚开路）时的引脚处理。
 2. 使下拉电阻的大小等于 100 k Ω 。
 3. 使下拉电阻的大小等于 10 k Ω （推荐）。

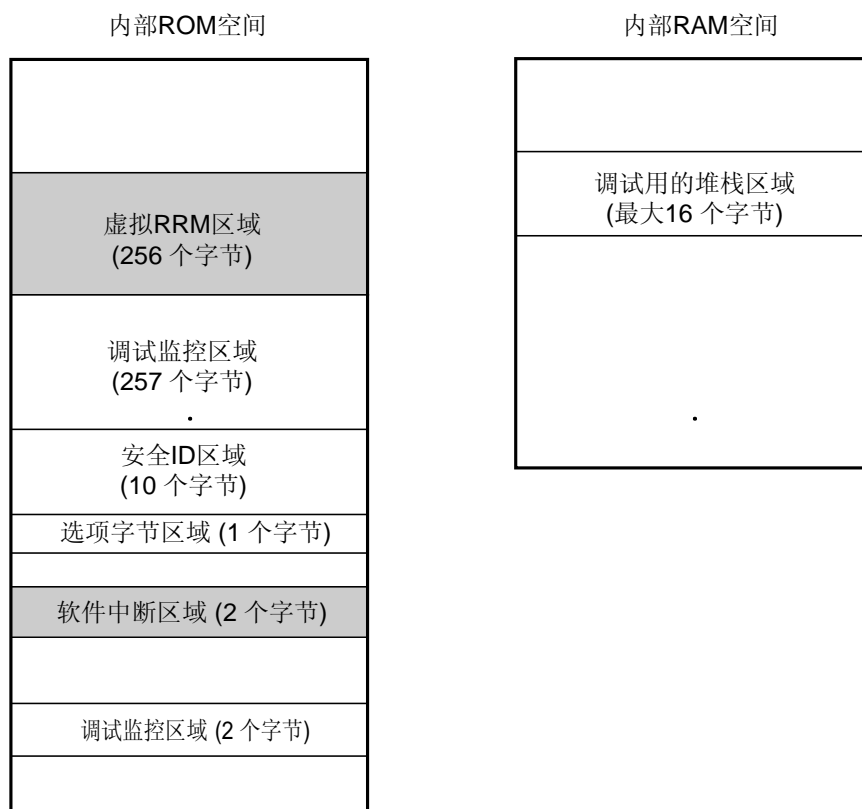
- 注意事项
1. 当 OCD1A/P04, OCD1B/P05 用于片上调试时，不要在片上调试模式中将 P04 和 P05 设为输出模式。
 2. 由于 $\overline{\text{RESET}}$ 引脚是通过 QB-MINI2 来使用的，因此复用功能不能用于仿真。

19.2 通过QB-MINI2 使用的保留区域

QB-MINI2 使用下图 19-3 所示保留区域以完成与目标设备的通信或调试功能。阴影保留区域用于所要使用的调试功能，其它区域则总是用于调试。使用用户程序和编译选项可以确保这些保留区域是安全的。

关于保留区域的详细信息，请参见 **QB-MINI2 用户手册 (U18371E)**。

图 19-3. 通过 QB-MINI2 使用的保留区域



备注 阴影保留区域：用于使用单个调试功能的区域
其他保留区域：始终用于调试的区域

第二十章 指令集

本章节以表格形式列出了 μ PD179F11x, 179F12x 微控制器的各个指令集。关于各种操作和操作代码的详细信息，请参考单独的文档 **78K/0 系列指令用户手册 (U12326E)**。

20.1 操作列表使用规则

20.1.1 操作数标识符及标识方法

根据规范规定的指令操作数标识方法（详细信息请参考汇编程序编程规范），在每种指令的“操作数”栏列出操作数。存在两种或更多的标识方法时，只要选择其中的一种方法。大写字母以及符号 #, !, \$ 和 [] 均为关键字，必须按原样书写。每种符号具有以下含义。

- #: 立即数据标识
- !: 绝对地址标识
- \$: 相对地址标识
- []: 间接地址标识

立即数用来描述一个数值型数据或标号。当使用标号时，请务必加上 #, !, \$, 及 [] 符号。

关于操作数寄存器标识符 r 和 rp、功能名称 (X、A、C, 等等) 或者绝对名称 (下表括号中的名称, R0、R1、R2, 等等) 可以用于标识。

表 20-1. 操作数标识符和标识方法

标识符	标识方法
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	特殊功能寄存器符号 [※]
sfrp	特殊功能寄存器符号 (仅限 16 位可操作寄存器偶地址) [※]
saddr	FE20H 到 FF1FH 立即数或标号
saddrp	FE20H 到 FF1FH 立即数或标号 (仅限偶地址)
addr16	0000H 到 FFFFH 立即数或标号 (仅限 16 位数据传送指令的偶地址)
addr11	0800H 到 0FFFH 立即数或标号
addr5	0040H 到 007FH 立即数或标号 (仅限偶地址)
word	16 位立即数或标号
byte	8 位立即数或标号
bit	3 位立即数或标号
RBn	RB0 到 RB3

注 这些操作数不能访问从 FFD0H 到 FFDFH 的地址。

备注 关于特殊功能寄存器符号的详细信息，请参见表 3-6 特殊功能寄存器列表。

20.1.2 操作栏的描述

- 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: 进位标志
- AC: 辅助进位标志
- Z: 零标志
- RBS: 寄存器 bank 选择标志
- IE: 中断请求允许标志
- (): 由括号中的地址或寄存器内容表示的存储器内容
- XH, XL: 16 位寄存器的高 8 位和低 8 位
- ^: 逻辑与 (AND)
- ∨: 逻辑加 (OR)
- ⊕: 异或逻辑加 (异或)
- : 数据取反
- addr16: 16 位立即数或标号
- jdisp8: 带符号 8 位数据 (位移值)

20.1.3 标志操作栏的描述

- (空): 不受影响
- 0: 清零
- 1: 设置为 1
- ×: 根据结果进行设置 / 清零
- R: 恢复先前保存的值

20.2 操作列表

指令组	助记符	操作数	字节	时钟		操作	标志	
				注 1	注 2		Z ACCY	
8 位数据传送	MOV	r, #byte	2	4	-	r ← byte		
		saddr, #byte	3	6	7	(saddr) ← byte		
		sfr, #byte	3	-	7	sfr ← byte		
		A, r ^{注 3}	1	2	-	A ← r		
		r, A ^{注 3}	1	2	-	r ← A		
		A, saddr	2	4	5	A ← (saddr)		
		saddr, A	2	4	5	(saddr) ← A		
		A, sfr	2	-	5	A ← sfr		
		sfr, A	2	-	5	sfr ← A		
		A, laddr16	3	8	9	A ← (addr16)		
		laddr16, A	3	8	9	(addr16) ← A		
		PSW, #byte	3	-	7	PSW ← byte	x x x	
		A, PSW	2	-	5	A ← PSW		
		PSW, A	2	-	5	PSW ← A	x x x	
		A, [DE]	1	4	5	A ← (DE)		
		[DE], A	1	4	5	(DE) ← A		
		A, [HL]	1	4	5	A ← (HL)		
		[HL], A	1	4	5	(HL) ← A		
		A, [HL + byte]	2	8	9	A ← (HL + byte)		
		[HL + byte], A	2	8	9	(HL + byte) ← A		
	A, [HL + B]	1	6	7	A ← (HL + B)			
	[HL + B], A	1	6	7	(HL + B) ← A			
	A, [HL + C]	1	6	7	A ← (HL + C)			
	[HL + C], A	1	6	7	(HL + C) ← A			
	XCH	^{注 3}	A, r	1	2	-	A ↔ r	
			A, saddr	2	4	6	A ↔ (saddr)	
			A, sfr	2	-	6	A ↔ sfr	
			A, laddr16	3	8	10	A ↔ (addr16)	
			A, [DE]	1	4	6	A ↔ (DE)	
			A, [HL]	1	4	6	A ↔ (HL)	
A, [HL + byte]			2	8	10	A ↔ (HL + byte)		
A, [HL + B]			2	8	10	A ↔ (HL + B)		
A, [HL + C]	2	8	10	A ↔ (HL + C)				

- 注
1. 当内部高速 RAM 区域被访问时或者对于一个没有数据访问的指令。
 2. 当内部高速 RAM 区域以外的区域被访问时
 3. “r = A” 除外

- 备注
1. 一个指令时钟周期是由处理器时钟控制寄存器 (PCC) 所选择的 CPU 时钟 (f_{cpu}) 的一个周期。
 2. 此时钟周期应用于内部 ROM 程序。

指令组	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 ^{注 3}	1	4	-	AX ← rp			
		rp, AX ^{注 3}	1	4	-	rp ← AX			
		AX, !addr16	3	10	12	AX ← (addr16)			
	!addr16, AX	3	10	12	(addr16) ← AX				
	XCHW	AX, rp ^{注 3}	1	4	-	AX ↔ rp			
8 位操作	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 ^{注 4}	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 ^{注 4}	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) + C	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. 当内部高速 RAM 区域被访问时或者对于一个没有数据访问的指令。
 2. 当内部高速 RAM 区域以外的区域被访问时
 3. 仅当 rp = BC, DE 或 HL 时
 4. “r = A” 除外

- 备注
1. 一个指令时钟周期是由处理器时钟控制寄存器 (PCC) 所选择的 CPU 时钟 (fcPU) 的一个周期。
 2. 此时钟周期应用于内部 ROM 程序。

指令组	助记符	操作数	字节	时钟		操作	标志
				注 1	注 2		Z AC CY
8 位操作	SUB	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 ^{注 3}	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	
	SUBC	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 ^{注 3}	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	
	AND	A, #byte	2	4	-	A ← A ∧ byte	x
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	x
		A, r ^{注 3}	2	4	-	A ← A ∧ r	x
		r, A	2	4	-	r ← r ∧ A	x
		A, saddr	2	4	5	A ← A ∧ (saddr)	x
		A, !addr16	3	8	9	A ← A ∧ (addr16)	x
		A, [HL]	1	4	5	A ← A ∧ (HL)	x
		A, [HL + byte]	2	8	9	A ← A ∧ (HL + byte)	x
		A, [HL + B]	2	8	9	A ← A ∧ (HL + B)	x
	A, [HL + C]	2	8	9	A ← A ∧ (HL + C)	x	

- 注
1. 当内部高速 RAM 区域被访问时或者对于一个没有数据访问的指令。
 2. 当内部高速 RAM 区域以外的区域被访问时
 3. “r = A” 除外

- 备注
1. 一个指令时钟周期是由处理器时钟控制寄存器 (PCC) 所选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 此时钟周期应用于内部 ROM 程序。

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	OR	A, #byte	2	4	-	$A \leftarrow A \vee \text{byte}$		x	
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$		x	
		A, r ^{注 3}	2	4	-	$A \leftarrow A \vee r$		x	
		r, A	2	4	-	$r \leftarrow r \vee A$		x	
		A, saddr	2	4	5	$A \leftarrow A \vee (\text{saddr})$		x	
		A, !addr16	3	8	9	$A \leftarrow A \vee (\text{addr16})$		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	-	$A \leftarrow A \nabla \text{byte}$		x	
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \nabla \text{byte}$		x	
		A, r ^{注 3}	2	4	-	$A \leftarrow A \nabla r$		x	
		r, A	2	4	-	$r \leftarrow r \nabla A$		x	
		A, saddr	2	4	5	$A \leftarrow A \nabla (\text{saddr})$		x	
		A, !addr16	3	8	9	$A \leftarrow A \nabla (\text{addr16})$		x	
		A, [HL]	1	4	5	$A \leftarrow A \nabla (\text{HL})$		x	
		A, [HL + byte]	2	8	9	$A \leftarrow A \nabla (\text{HL} + \text{byte})$		x	
		A, [HL + B]	2	8	9	$A \leftarrow A \nabla (\text{HL} + B)$		x	
	A, [HL + C]	2	8	9	$A \leftarrow A \nabla (\text{HL} + C)$		x		
	CMP	A, #byte	2	4	-	$A - \text{byte}$	x	x	x
		saddr, #byte	3	6	8	$(\text{saddr}) - \text{byte}$	x	x	x
		A, r ^{注 3}	2	4	-	$A - r$	x	x	x
		r, A	2	4	-	$r - A$	x	x	x
		A, saddr	2	4	5	$A - (\text{saddr})$	x	x	x
		A, !addr16	3	8	9	$A - (\text{addr16})$	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	

- 注
1. 当内部高速 RAM 区域被访问时或者对于一个没有数据访问的指令。
 2. 当内部高速 RAM 区域以外的区域被访问时
 3. “r = A” 除外

- 备注
1. 一个指令时钟周期是由处理器时钟控制寄存器 (PCC) 所选择的 CPU 时钟 (f_{cpu}) 的一个周期。
 2. 此时钟周期应用于内部 ROM 程序。

指令组	助记符	操作数	字节	时钟		操作	标志
				注 1	注 2		Z AC CY
16 位操作	ADDW	AX, #word	3	6	-	AX, CY ← AX + word	x x x
	SUBW	AX, #word	3	6	-	AX, CY ← AX - word	x x x
	CMPW	AX, #word	3	6	-	AX - word	x x x
乘法 / 除法	MULU	X	2	16	-	AX ← A × X	
	DIVUW	C	2	25	-	AX (商), C (余数) ← AX ÷ C	
递增 / 递减	INC	r	1	2	-	r ← r + 1	x x
		saddr	2	4	6	(saddr) ← (saddr) + 1	x x
	DEC	r	1	2	-	r ← r - 1	x x
		saddr	2	4	6	(saddr) ← (saddr) - 1	x x
	INCW	rp	1	4	-	rp ← rp + 1	
	DECW	rp	1	4	-	rp ← rp - 1	
循环	ROR	A, 1	1	2	-	(CY, A7 ← A0, Am-1 ← Am) × 1 次	x
	ROL	A, 1	1	2	-	(CY, A0 ← A7, Am+1 ← Am) × 1 次	x
	RORC	A, 1	1	2	-	(CY ← A0, A7 ← CY, Am-1 ← Am) × 1 次	x
	ROLC	A, 1	1	2	-	(CY ← A7, A0 ← CY, Am+1 ← Am) × 1 次	x
	ROR4	[HL]	2	10	12	A3-0 ← (HL)3-0, (HL)7-4 ← A3-0, (HL)3-0 ← (HL)7-4	
	ROL4	[HL]	2	10	12	A3-0 ← (HL)7-4, (HL)3-0 ← A3-0, (HL)7-4 ← (HL)3-0	
BCD 调整	ADJBA		2	4	-	加法后的十进制调整累加器	x x x
	ADJBS		2	4	-	减法后的十进制调整累加器	x x x
位操作	MOV1	CY, saddr.bit	3	6	7	CY ← (saddr.bit)	x
		CY, sfr.bit	3	-	7	CY ← sfr.bit	x
		CY, A.bit	2	4	-	CY ← A.bit	x
		CY, PSW.bit	3	-	7	CY ← PSW.bit	x
		CY, [HL].bit	2	6	7	CY ← (HL).bit	x
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY	
		sfr.bit, CY	3	-	8	sfr.bit ← CY	
		A.bit, CY	2	4	-	A.bit ← CY	
		PSW.bit, CY	3	-	8	PSW.bit ← CY	x x
[HL].bit, CY	2	6	8	(HL).bit ← CY			

- 注**
1. 当内部高速 RAM 区域被访问时或者对于一个没有数据访问的指令。
 2. 当内部高速 RAM 区域以外的区域被访问时

- 备注**
1. 一个指令时钟周期是由处理器时钟控制寄存器 (PCC) 所选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 此时钟周期应用于内部 ROM 程序。

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
位操作	AND1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \wedge (\text{saddr.bit})$			x
		CY, sfr.bit	3	-	7	$CY \leftarrow CY \wedge \text{sfr.bit}$			x
		CY, A.bit	2	4	-	$CY \leftarrow CY \wedge A.\text{bit}$			x
		CY, PSW.bit	3	-	7	$CY \leftarrow CY \wedge \text{PSW.bit}$			x
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \wedge (\text{HL}).\text{bit}$			x
	OR1	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
	XOR1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \oplus (\text{saddr.bit})$			x
		CY, sfr.bit	3	-	7	$CY \leftarrow CY \oplus \text{sfr.bit}$			x
		CY, A.bit	2	4	-	$CY \leftarrow CY \oplus A.\text{bit}$			x
		CY, PSW.bit	3	-	7	$CY \leftarrow CY \oplus \text{PSW.bit}$			x
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \oplus (\text{HL}).\text{bit}$			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

- 注
1. 当内部高速 RAM 区域被访问时或者对于一个没有数据访问的指令。
 2. 当内部高速 RAM 区域以外的区域被访问时

- 备注
1. 一个指令时钟周期是由处理器时钟控制寄存器 (PCC) 所选择的 CPU 时钟 (f_{cpu}) 的一个周期。
 2. 此时钟周期应用于内部 ROM 程序。

指令组	助记符	操作数	字节	时钟		操作	标志
				注 1	注 2		Z AC CY
调用 / 返回	CALL	!addr16	3	7	-	$(SP - 1) \leftarrow (PC + 3)_H, (SP - 2) \leftarrow (PC + 3)_L,$ $PC \leftarrow \text{addr16}, SP \leftarrow SP - 2$	
	CALLF	!addr11	2	5	-	$(SP - 1) \leftarrow (PC + 2)_H, (SP - 2) \leftarrow (PC + 2)_L,$ $PC_{15-11} \leftarrow 00001, PC_{10-0} \leftarrow \text{addr11},$ $SP \leftarrow SP - 2$	
	CALLT	[addr5]	1	6	-	$(SP - 1) \leftarrow (PC + 1)_H, (SP - 2) \leftarrow (PC + 1)_L,$ $PC_H \leftarrow (\text{addr5} + 1), PC_L \leftarrow (\text{addr5}),$ $SP \leftarrow SP - 2$	
	BRK		1	6	-	$(SP - 1) \leftarrow PSW, (SP - 2) \leftarrow (PC + 1)_H,$ $(SP - 3) \leftarrow (PC + 1)_L, PC_H \leftarrow (003FH),$ $PC_L \leftarrow (003EH), SP \leftarrow SP - 3, IE \leftarrow 0$	
	RET		1	6	-	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $SP \leftarrow SP + 2$	
	RETI		1	6	-	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $PSW \leftarrow (SP + 2), SP \leftarrow SP + 3$	R R R
	RETB		1	6	-	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $PSW \leftarrow (SP + 2), SP \leftarrow SP + 3$	R R R
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	$SP \leftarrow \text{word}$	
		SP, AX	2	-	8	$SP \leftarrow AX$	
AX, SP		2	-	8	$AX \leftarrow SP$		
无条件转移	BR	!addr16	3	6	-	$PC \leftarrow \text{addr16}$	
		\$addr16	2	6	-	$PC \leftarrow PC + 2 + \text{jdisp8}$	
		AX	2	8	-	$PC_H \leftarrow A, PC_L \leftarrow X$	
条件转移	BC	\$addr16	2	6	-	如果 $CY = 1$, 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	
	BNC	\$addr16	2	6	-	如果 $CY = 0$, 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	
	BZ	\$addr16	2	6	-	如果 $Z = 1$, 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	
	BNZ	\$addr16	2	6	-	如果 $Z = 0$, 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	

- 注
1. 当内部高速 RAM 区域被访问时或者对于一个没有数据访问的指令。
 2. 当内部高速 RAM 区域以外的区域被访问时

- 备注
1. 一个指令时钟周期是由处理器时钟控制寄存器 (PCC) 所选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 此时钟周期应用于内部 ROM 程序。

指令组	助记符	操作数	字节	时钟		操作	标志			
				注 1	注 2		Z	AC	CY	
条件转移	BT	saddr.bit, \$addr16	3	8	9	如果 (saddr.bit) = 1, 则 PC ← PC + 3 + jdisp8				
		sfr.bit, \$addr16	4	-	11	如果 sfr.bit = 1, 则 PC ← PC + 4 + jdisp8				
		A.bit, \$addr16	3	8	-	如果 A.bit = 1, 则 PC ← PC + 3 + jdisp8				
		PSW.bit, \$addr16	3	-	9	如果 PSW.bit = 1, 则 PC ← PC + 3 + jdisp8				
		[HL].bit, \$addr16	3	10	11	如果 (HL).bit = 1, 则 PC ← PC + 3 + jdisp8				
	BF	saddr.bit, \$addr16	4	10	11	如果 (saddr.bit) = 0, 则 PC ← PC + 4 + jdisp8				
		sfr.bit, \$addr16	4	-	11	如果 sfr.bit = 0, 则 PC ← PC + 4 + jdisp8				
		A.bit, \$addr16	3	8	-	如果 A.bit = 0, 则 PC ← PC + 3 + jdisp8				
		PSW.bit, \$addr16	4	-	11	如果 PSW.bit = 0, 则 PC ← PC + 4 + jdisp8				
		[HL].bit, \$addr16	3	10	11	如果 (HL).bit = 0, 则 PC ← PC + 3 + jdisp8				
	BTCLR	saddr.bit, \$addr16	4	10	12	如果 (saddr.bit) = 1, 则 PC ← PC + 4 + jdisp8, 然后复位 (saddr.bit)				
		sfr.bit, \$addr16	4	-	12	如果 sfr.bit = 1, 则 PC ← PC + 4 + jdisp8, 然后复位 sfr.bit				
		A.bit, \$addr16	3	8	-	如果 A.bit = 1, 则 PC ← PC + 3 + jdisp8, 然后复位 A.bit				
		PSW.bit, \$addr16	4	-	12	如果 PSW.bit = 1, 则 PC ← PC + 4 + jdisp8 然后复位 PSW.bit	x	x	x	
		[HL].bit, \$addr16	3	10	12	如果 (HL).bit = 1, 则 PC ← PC + 3 + jdisp8 然后复位 (HL).bit				
	DBNZ	B, \$addr16	2	6	-	B ← B - 1, 那么 如果 B ≠ 0, 则 PC ← PC + 2 + jdisp8				
		C, \$addr16	2	6	-	C ← C - 1, 那么 如果 C ≠ 0, 则 PC ← PC + 2 + jdisp8				
		saddr, \$addr16	3	8	10	(saddr) ← (saddr) - 1, 那么 如果 (saddr) ≠ 0, 则 PC ← PC + 3 + jdisp8				
	CPU 控制	SEL	RBn	2	4	-	RBS1, 0 ← n			
		NOP		1	2	-	无操作			
EI			2	-	6	IE ← 1 (允许中断)				
DI			2	-	6	IE ← 0 (禁止中断)				
HALT			2	6	-	设置 HALT 模式				
STOP			2	6	-	设置 STOP 模式				

- 注
1. 当内部高速 RAM 区域被访问时或者对于一个没有数据访问的指令。
 2. 当内部高速 RAM 区域以外的区域被访问时

- 备注
1. 一个指令时钟周期是由处理器时钟控制寄存器 (PCC) 所选择的 CPU 时钟 (f_{cpu}) 的一个周期。
 2. 此时钟周期应用于内部 ROM 程序。

20.3 按寻址类型列出的指令

(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

第二操作数 第一操作数	#byte	A	r*	sfr	saddr	!addr16	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

第二操作数 第一操作数	#word	AX	rp [#]	sfrp	saddrp	laddr16	SP	无
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW [#]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
laddr16		MOVW						
SP	MOVW	MOVW						

注 仅当 rp = BC, DE, HL 时

(3) 位操作指令

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

第二操作数 第一操作数	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 / 跳转指令

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

第二操作数 第一操作数	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

第二十一章 电气特性

最大绝对额定值 (T_A = 25°C)

参数	符号	条件		额定值	单位
供电电压	V _{DD}			-0.5 到+6.5	V
	V _{SS}			-0.5 到+0.3	V
REGC 引脚输入电压	V _{I_{REGC}}			-0.5 到+3.6 以及 -0.5 到 V _{DD}	V
输入电压	V _{I1}	P00 到 P07, P10 到 P17, P20 到 P22, P23 ^{註2} , P24 ^{註2} , P25 到 P27, P30 到 P35 ^{註2} , P120-P123, FLMD0		-0.3 到 V _{DD} + 0.3 ^{註1}	V
输出电压	V _O			-0.3 到 V _{DD} + 0.3 ^{註1}	V
输出电流, 高	I _{OH1}	每个引脚	P00 到 P06, P10 到 P17, P20 到 P22, P23 ^{註2} , P24 ^{註2} , P25 到 P27, P30 到 P35 ^{註2} , P120 到 P122	-40	mA
		所有引脚		-80	mA
	I _{OH2}	每个引脚	REM	-40	mA
输出电流, 低	I _{OL1}	每个引脚	P00 到 P03, P07, P10 到 P17, P20 到 P22, P23 ^{註2} , P30 到 P35 ^{註2} , P121, P122	40	mA
		所有引脚		150	mA
	I _{OL2}	每个引脚	P04 到 P06, P24 ^{註2} , P25 到 P27, P120	40	mA
		所有引脚		150	mA
工作环境温度	T _A			-40 到+85	°C
存储温度	T _{stg}			-65 到+150	°C

注 1. 必须为 6.5 V 或更低。

2. 仅限 38 引脚的产品

注意事项 任何一项参数哪怕是在瞬间超过最大额定值, 都会使产品质量受到影响。也就是说, 最大额定值是产品濒临物理损坏的临界点, 因而, 必须保证产品在不超过最大额定值的条件下使用。

备注 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

X1 振荡器特性

(TA = -40 到 +85°C, 1.8 V ≤ VDD ≤ 3.6 V, VSS = 0 V)

振荡器	推荐电路	参数	条件	最小.	典型值.	最大.	单位
陶瓷振荡器		X1 时钟振荡频率 (fx) ‡		1.0		4.0	MHz
晶体振荡器		X1 时钟振荡频率 (fx) ‡		1.0		4.0	MHz

注 只显示振荡器特性。关于指令执行时间请参见 AC 特性。

注意事项 1. 当使用 X1 振荡器时，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接电容产生不利影响。

- 连接线越短越好。
- 连接线不应与其他信号线交叉。
- 流经的电流变化较大的信号线不要在振荡器周围布线。
- 要保持振荡器电容器的接地点电压与 VSS 相同。
- 不要将电容的地信号接入大电流地。
- 不要从振荡器获取信号。

2. 由于复位释放后 CPU 通过内部高速振荡时钟被启动，因此用户需要使用振荡稳定时间计数器状态寄存器 (OSTC) 来检查 X1 时钟振荡稳定时间。充分评估要使用的振荡器的振荡稳定时间后，确定 OSTC 寄存器的振荡稳定时间和振荡稳定时间选择寄存器 (OSTS)。

内部振荡器特性

(TA = -40 到 +85°C, 1.8 V ≤ VDD ≤ 3.6 V, VSS = 0 V)

振荡器	参数	条件	最小.	典型值.	最大.	单位
4 MHz 内部振荡器	内部高速振荡时钟频率 (fRH) ‡	RSTS = 1 TA = -10 到 +70°C	3.92	4.0	4.08	MHz
240 kHz 内部振荡器	内部低速振荡时钟频率 (fRL)	2.1 V ≤ VDD ≤ 3.6 V	216	240	276	kHz
		1.8 V ≤ VDD < 2.1 V	180	240	300	kHz

注 只显示振荡器特性。关于指令执行时间请参见 AC 特性。

备注 RSTS: 内部振荡模式寄存器 (RCM) 的第 7 位

DC 特性 (1/2)

(T_A = -40 到 +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

参数	符号	条件	最小.	典型值.	最大.	单位		
输出电流, 高 ^{注1}	I _{OH1}	P00 到 P06, P10 到 P17, P20 到 P22, P23 ^{注2} , P24 ^{注2} , P25 到 P27, P30 到 P35 ^{注2} , P120 到 P122 的每个引脚			-2.0	mA		
		P04 到 P06, P24 ^{注2} , P25 到 P27, P120 的所有引脚			-4.0	mA		
		P00 到 P06, P10 到 P17, P20 到 P22, P23 ^{注2} , P24 ^{注2} , P25 到 P27, P30 到 P35 ^{注2} , P121, P122 的所有引脚			-20.0	mA		
	I _{OH2}	P07 (REM) V _{DD} = 3.0 V, V _{OH2} = 1.0 V	-6.0	-13.0	-25.0	mA		
	所有的引脚 ^{注3}					-49.0	mA	
输出电流, 低 ^{注4}	I _{OL1}	P00 到 P03, P07, P10 到 P17, P20 到 P22, P23 ^{注2} , P30 到 P35 ^{注2} , P121, P122	每个引脚			1.0	mA	
			所有引脚				20.0	mA
	I _{OL2}	P04 到 P06, P24 ^{注2} , P25 到 P27, P120	每个引脚				3.0	mA
			所有引脚				30.0	mA
	所有的引脚 ^{注3}						50.0	mA
输入电压, 高	V _{IH1}	P07, P10 到 P17, P20 到 P22, P23 ^{注2} , P30 到 P35 ^{注2} , P121, P122, P123 (KR8)	0.7V _{DD}		V _{DD}	V		
	V _{IH2}	P00 到 P06, P24 ^{注2} , P25 到 P27, P120, RESET/P123 (INTP5)	0.8V _{DD}		V _{DD}	V		
	V _{IH3}	FLMD0	0.9V _{DD}		V _{DD}	V		
输入电压, 低	V _{IL1}	P07, P10 到 P17, P20 到 P22, P23 ^{注2} , P30 到 P35 ^{注2} , P121, P122, P123 (KR8)	0		0.3V _{DD}	V		
	V _{IL2}	P00 到 P06, P24 ^{注2} , P25 到 P27, P120, RESET/P123 (INTP5)	0		0.2V _{DD}	V		
	V _{IL3}	FLMD0	0		0.1V _{DD}	V		
输出电压, 高	V _{OH1}	P00 到 P06, P10 到 P17, P20 到 P22, P23 ^{注2} , P24 ^{注2} , P25 到 P27, P30 到 P35 ^{注2} , P120 到 P122	V _{DD} = 1.8 V I _{OH1} = -1.0 mA	V _{DD} -0.5		V		
	V _{OH2}	P07	V _{DD} = 3.0 V I _{OH2} = -6.0 mA	1.0		V		
输出电压, 低	V _{OL1}	P00 到 P03, P07, P10 到 P17, P20 到 P22, P23 ^{注2} , P30 到 P35 ^{注2} , P121, P122	V _{DD} = 1.8 V I _{OL1} = 0.5 mA		0.4	V		
	V _{OL2}	P04 到 P06, P24 ^{注2} , P25 到 P27, P120	V _{DD} = 2.0 V I _{OL2} = 1.5 mA		0.1	V		

- 注
- 设备操作的电流值被保证, 即使电流从 V_{DD} 流向一个输出引脚。
 - 仅限 38 引脚的产品
 - 占空比为 70% 的 (输出电流时间为 0.7 × t, 不输出电流时间为 0.3 × t, t 为指定时间) 条件下的描述。占空比不是 70% 时这些引脚总的输出电流可通过下面的表达式计算。
 - I_{OH} 的占空比为 n% 处: 引脚总的输出电流 = (I_{OH} × 0.7) / (n × 0.01)

<例> 占空比为 50% 处, I_{OH} = 20.0 mA
 引脚的总共输出电流 = (20.0 × 0.7) / (50 × 0.01) = 28.0 mA

然而, 允许流入一个引脚的电流不会根据占空比而改变。高于最大绝对额定值的电流不能流入一个引脚。
 - 设备操作的电流值被保证, 即使电流从一个输出引脚流向 GND。

备注 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

DC 特性 (2/2)

(TA = -40 到 +85°C, 1.8 V ≤ VDD ≤ 3.6 V, VSS = 0 V)

参数	符号	条件		最小.	典型值.	最大.	单位
输入泄漏电流, 高	I _{IH1}	P00 到 P07, P10 到 P17, P20 到 P22, P23 ^{‡1} , P24 ^{‡1} , P25 到 P27, P30 到 P35 ^{‡1} , P120, P123	V _I = V _{DD}			3	μA
	I _{IH3}	P121, P122 (X1, X2)	V _I = V _{DD}	I/O 端口模式		3	μA
				OSC 模式		20	μA
输入泄漏电流, 低	I _{L1}	P00 到 P07, P10 到 P17, P20 到 P22, P23 ^{‡1} , P24 ^{‡1} , P25 到 P27, P30 到 P35 ^{‡1} , P120, P123	V _I = V _{SS}			-3	μA
	I _{L2}	P121, P122 (X1, X2)	V _I = V _{SS}	I/O 端口模式		-3	μA
				OSC 模式		-20	μA
上拉电阻	R _{U1}	V _I = V _{SS}	P00 到 P07, P20 到 P22, P23 ^{‡1} , P24 ^{‡1} , P25 到 P27, P120 到 P122	10	20	100	kΩ
	R _{U2}		P10 到 P17, P30 到 P35 ^{‡1} , RESET	75	150	300	kΩ
	R _{U3}		FLMDO 1.8 V ≤ V _{DD} ≤ 3.6 V	10	19.5	52	kΩ
供电电流 ^{‡2}	I _{DD1}	操作模式	f _{XH} = 4 MHz ^{‡3} , V _{DD} = 3.0 V		0.8	1.2	mA
			f _{RH} = 4 MHz ^{‡4} , V _{DD} = 3.0 V		0.7	1.0	mA
	I _{DD2}	HALT 模式	f _{XH} = 4 MHz ^{‡3} , V _{DD} = 3.0 V		0.25	0.33	mA
			f _{RH} = 4 MHz ^{‡4} , V _{DD} = 3.0 V		0.15	0.28	mA
I _{DD3}	STOP 模式 ^{‡4}	V _{DD} = 3.0 V		1	20	μA	
看门狗定时器工作电流	I _{WDT} ^{‡5}	240 kHz 内部低速振荡时钟操作过程中			3.2	6.4	μA
LVI 工作电流	I _{LVI} ^{‡6}				5.8	12	μA

- 注
1. 仅限 38 引脚的产品
 2. 输入引脚电平固定为 V_{DD} 或 V_{SS} 时流入内部电源 (V_{DD}) 的总的电流, 其中包括外围工作电流以及流入的输入泄漏电流。然而, 其中不包括流入上拉电阻的电流以及端口的输出电流。
 3. 不包括 4 MHz 内部振荡器以及 240 kHz 内部振荡器的工作电流, 也不包括流入看门狗定时器以及 LVI 电路的电流。
 4. 不包括 240 kHz 内部振荡器的工作电流, 也不包括流入看门狗定时器以及 LVI 电路的电流。
 5. 仅流入看门狗定时器的电流, 包括 240 kHz 内部振荡器的工作电流。当看门狗定时器在 HALT 或 STOP 模式中运行, μPD179F11x, 179F12x 微控制器工作电流是 I_{DD2} 或 I_{DD3} 与 I_{WDT} 之和。
 6. 仅流入 LVI 电路的电流。当 LVI 电路在 HALT 或 STOP 模式中运行, μPD179F11x, 179F12x 微控制器工作电流是 I_{DD2} 或 I_{DD3} 与 I_{LVI} 之和。

备注 1. f_{XH}: 高速系统时钟频率 (X1 时钟振荡频率或外部主系统时钟频率)

2. f_{RH}: 内部高速振荡时钟频率

AC 特性

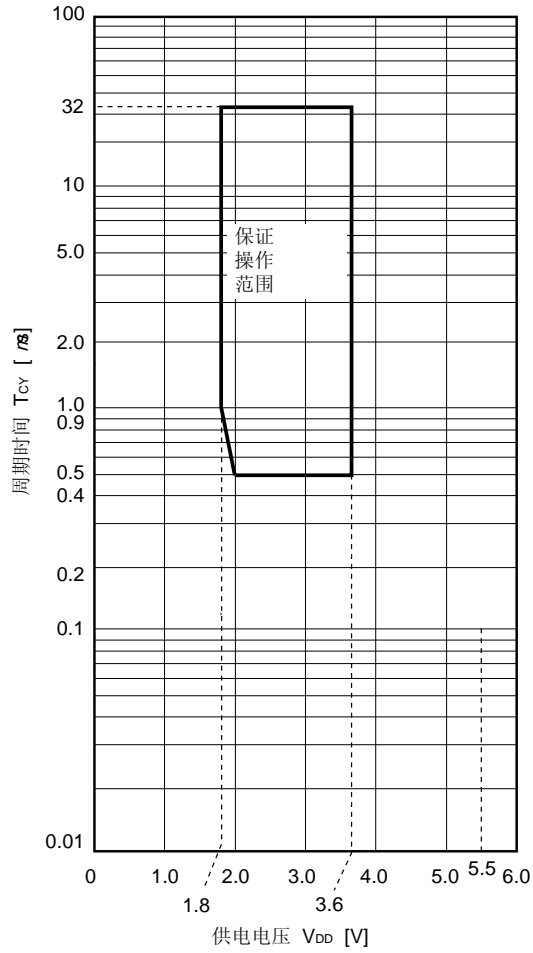
(1) 基本操作

(T_A = -40 到 +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

参数	符号	条件	最小.	典型值.	最大.	单位	
指令周期 (最小指令执行时间)	T _{CY}	主系统 时钟 (f _{XP}) 操作	2.0 V ≤ V _{DD} ≤ 3.6 V	0.5		32	μs
			1.8 V ≤ V _{DD} < 2.0 V	1		32	μs
外部主系统时钟频率	f _{EXCLK}		1.0		4.0	MHz	
外部主系统时钟输入高电平宽度, 低电平宽度	t _{EXCLKH} , t _{EXCLKL}		(1/f _{EXCLK} K×1/2) -1			ns	
TI000, TI010 输入高电平宽度, 低电平宽度	t _{TIH0} , t _{TIL0}	2.1 V ≤ V _{DD} ≤ 3.6 V	2/f _{sam} + 0.2 ^注			μs	
		1.8 V ≤ V _{DD} < 2.1 V	2/f _{sam} + 0.5 ^注			μs	
TI50, TI51 输入频率	f _{TI5}				4	MHz	
TI50, TI51 输入高电平宽度, 低电平宽度	t _{TIH5} , t _{TIL5}		250			ns	
中断输入高电平宽度, 低电平宽度	t _{INTH} , t _{INTL}		1			μs	
按键中断输入低电平宽度	t _{KR}		250			ns	
RESET 低电平宽度	t _{RSL}		10			μs	

注 可以使用预分频模式寄存器 00 (PRM00) 的 0 位和第 1 位 (PRM000, PRM001) 来选择 f_{sam} = f_{PRS}, f_{PRS}/4, f_{PRS}/256。将 TI000 有效沿选作计数时钟时需注意 f_{sam} = f_{PRS}。

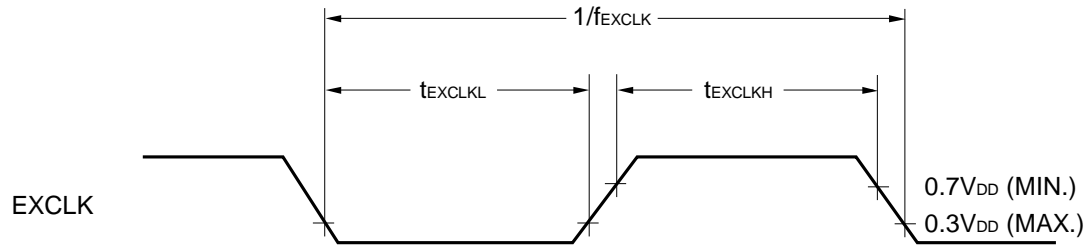
T_{cy} vs. V_{DD} (主系统时钟操作)



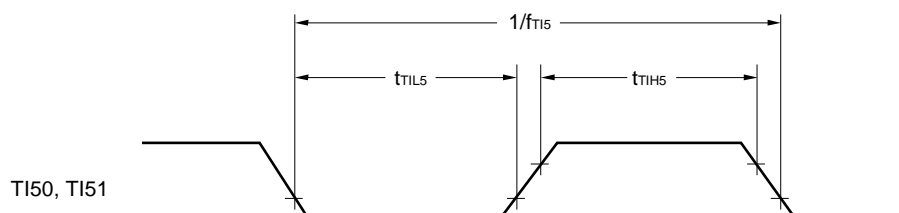
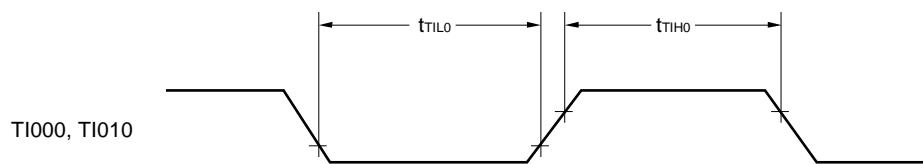
AC 时序测试点



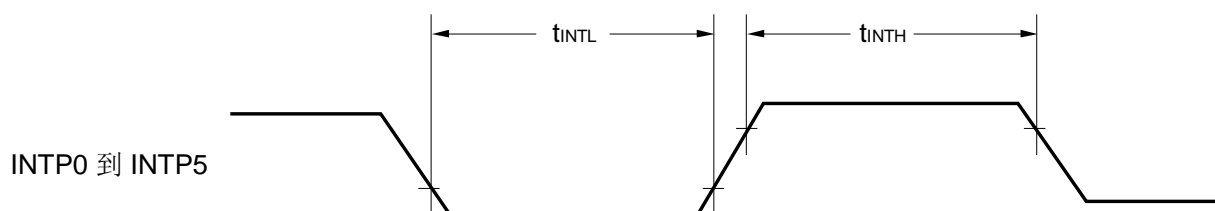
外部主系统时钟时序, 外部副系统时钟时序



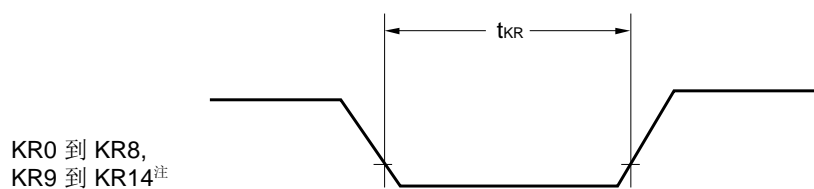
TI 的时序



中断请求输入时序

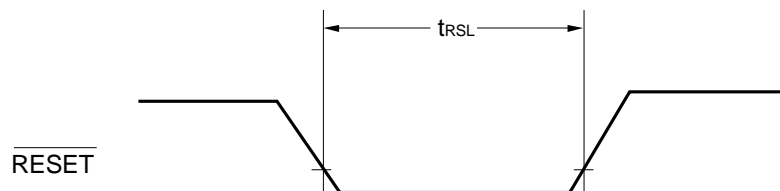


按键中断输入时序



注 仅限 38 引脚的产品

RESET 输入时序



(2) 串行接口

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 3.6\text{ V}$, $V_{SS} = 0\text{ V}$)

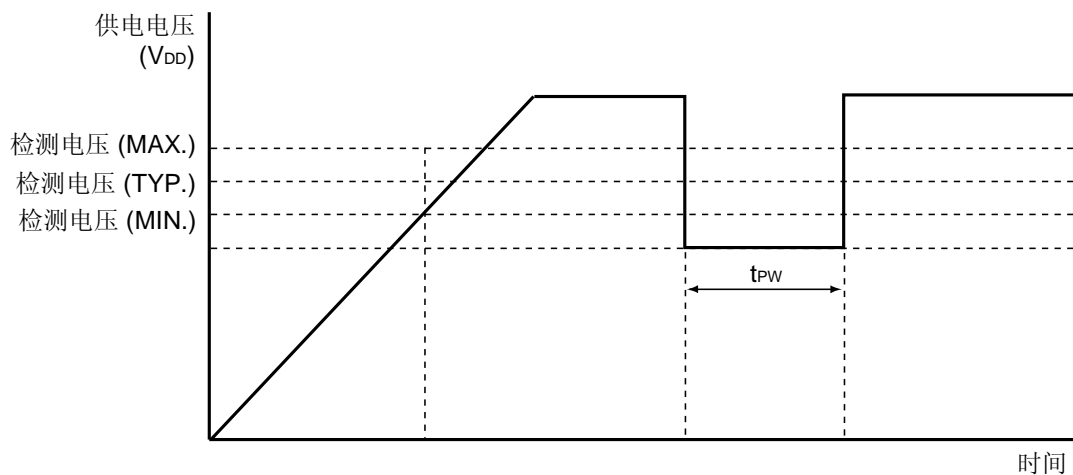
(a) UART6 (专用波特率发生器输出)

参数	符号	条件	最小.	典型值.	最大.	单位
传输速率					312.5	kbps

POC 电路特性 ($T_A = -40$ 到 $+85^{\circ}\text{C}$, $V_{SS} = 0\text{ V}$)

参数	符号	条件	最小.	典型值.	最大.	单位
检测电压	V_{POC}		1.7	1.8	1.9	V
最小脉冲宽度	t_{PW}		200			μs

POC 电路时序



注 在 $V_{DD} < V_{POC}$ 的状态保持 $200\ \mu\text{s}$ 时，将会通过 POC 电路生成内部复位信号。 $200\ \mu\text{s}$ 为保证值，因此它可能在 $T_{PW} < 200\ \mu\text{s}$ 的情况下生成内部复位信号。

POC 检测电压可能低于操作电压范围 ($V_{DD} = 1.8$ 到 3.6V)，但 CPU 不会在此时溢出。

如果操作没有保证的振荡器处于低压中，那么需注意在 POC 复位生成前 X1 振荡可能会停止。

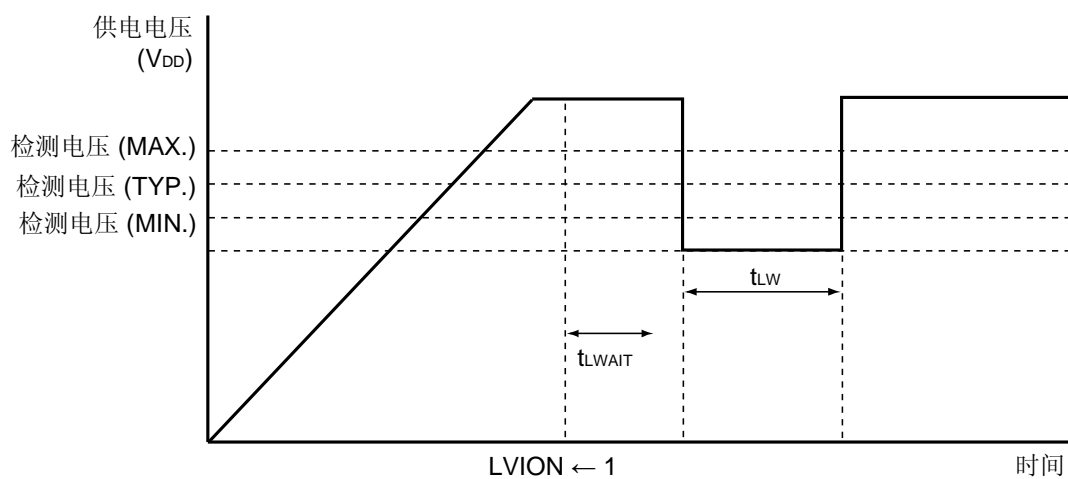
LVI 电路特性 ($T_A = -40$ 到 $+85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 3.6\text{ V}$, $V_{SS} = 0\text{ V}$)

参数		符号	条件	最小.	典型值.	最大.	单位
检测电压	供电电压电平	V_{LV15}		3.37	3.47	3.57	V
		V_{LV16}		3.22	3.32	3.42	V
		V_{LV17}		3.06	3.16	3.26	V
		V_{LV18}		2.91	3.01	3.11	V
		V_{LV19}		2.75	2.85	2.95	V
		V_{LV110}		2.60	2.70	2.80	V
		V_{LV111}		2.45	2.55	2.65	V
		V_{LV112}		2.29	2.39	2.49	V
		V_{LV113}		2.14	2.24	2.34	V
		V_{LV114}		2.00	2.08	2.15	V
		V_{LV115}		1.83	1.93	2.03	V
	外部输入引脚 ^{注1}	EXLVI	EXLVI < V_{DD}		1.21		V
最小脉冲宽度		t_{LW}		200			μs
操作稳定等待时间 ^{注2}		t_{LWAIT}				10	μs

- 注 1. 使用 EXLVI/P120/INTP0 引脚。
 2. 从设置低电压检测寄存器 (LVIM) 的第 7 位 (LVION) 为 1 到工作稳定所需要的时间。

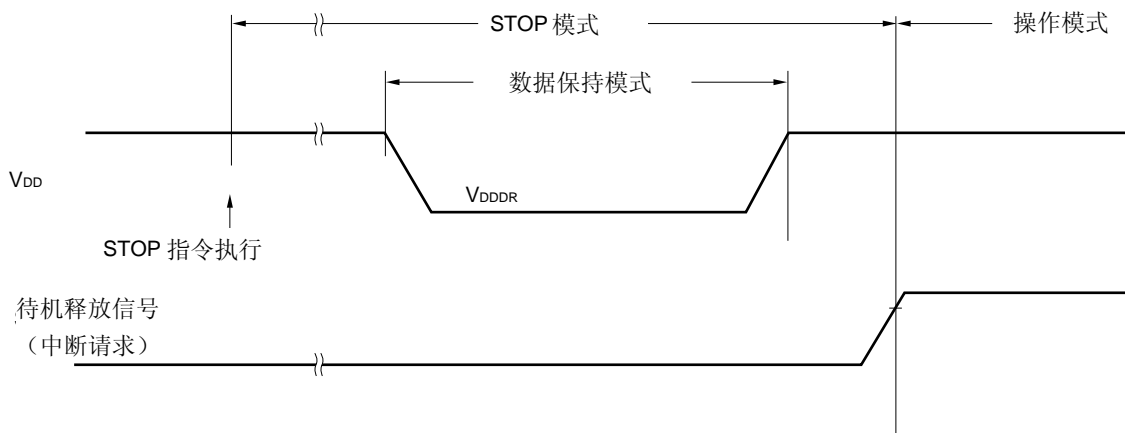
备注 $V_{LV1(n-1)} > V_{LV1n}$: $n = 6$ 到 15

LVI 电路时序



数据存储器 STOP 模式低供电电压数据保持特性 ($T_A = -40$ 至 $+85^\circ\text{C}$)

参数	符号	条件	最小.	典型值.	最大.	单位
数据保持供电电压	V_{DDDR}		1.3		3.6	V

电池更换时数据存储器保持检测电压且供电电压下降 ($T_A = -40$ 到 $+85^\circ\text{C}$)

参数	符号	条件	最小.	典型值.	最大.	单位
RAM 保持的检测电压	V_{LD}	LVDET0 位 “1”→“0”	1.30	1.40	1.50	V

<R> Flash 存储器编程特性 ($T_A = -40$ 到 $+85^\circ\text{C}$, $2.0\text{ V} \leq V_{DD} \leq 3.6\text{ V}$, $V_{SS} = 0\text{ V}$)

• 基本特性

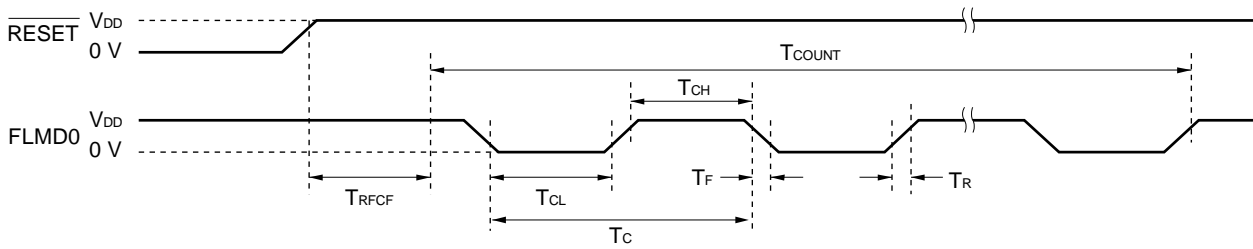
参数	符号	条件	最小.	典型值.	最大.	单位
V_{DD} 供电电流	I_{DD}	$f_{XP} = 4\text{ MHz}$ (TYP.)		4.5	11.0	mA
擦除时间 ^{注1, 2}	所有 block	T_{eraca}		20	200	ms
	Block 单元	T_{erasa}		20	200	ms
写入时间 (以 8 位为单位) ^{注1}	T_{wrwa}			10	100	μs
重写各芯片的次数	C_{enwr}	1 次擦除 + 擦除后写 1 次 = 1 重写 ^{注3}	1000			次数

- 注
1. flash 寄存器的特性。
 2. 不包括擦除前的预写时间以及擦除验证时间 (回写时间)。
 3. 当产品在发货后第一次被写入时, “擦除 → 写入”和“只写入”都被认为一次重写。”

备注 f_{XP} : 主系统时钟振荡频率

• 串行写入操作特性

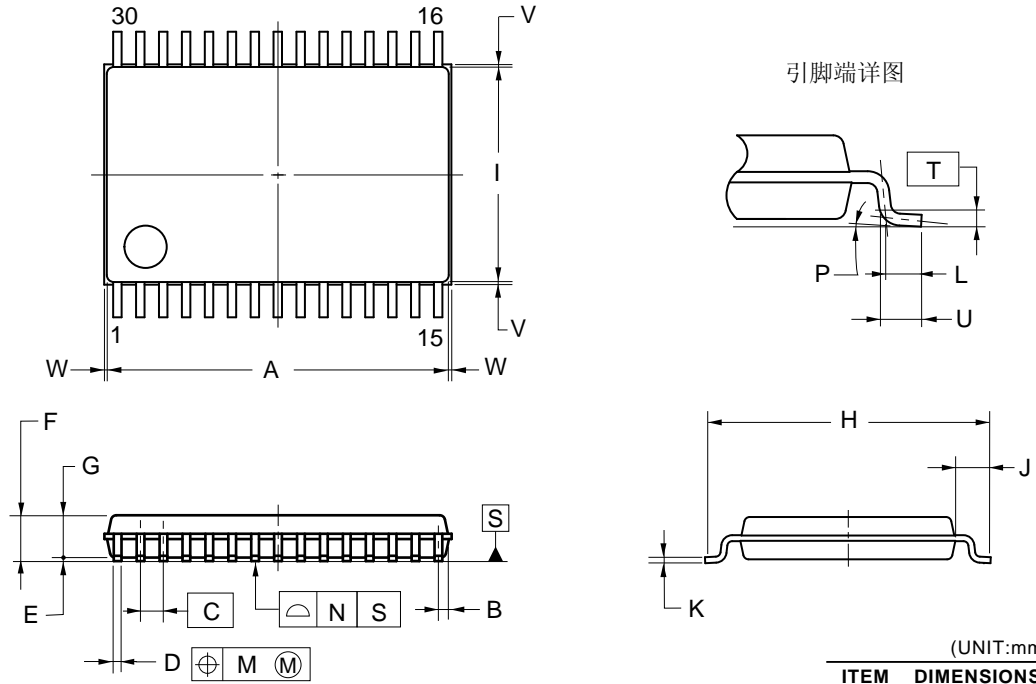
参数	符号	条件	最小.	典型值.	最大.	单位
计数启动时间 从 RESET↑ 到 FLMD0	T_{RFCF}		4.1		17.1	ms
计数执行时间	T_{COUNT}		10.8		13.2	ms
FLMD0 计数器高电平宽度 / 低电平宽度	T_{CH}/ T_{CL}		$T_C \times 0.45$			μs
FLMD0 计数器上升时间 / 下降时间	T_R/ T_F		12.5			μs



第二十二章 封装图

- μ PD179F110MC-CAB-AX, 179F111MC-CAB-AX, 179F112MC-CAB-AX, 179F113MC-CAB-AX, 179F114MC-CAB-AX

38 引脚塑封 SSOP (7.62mm (300))



注

该结构在最大使用材料情况下，每条引脚的中心线位于其实际位置的 0.13mm 内

(UNIT:mm)

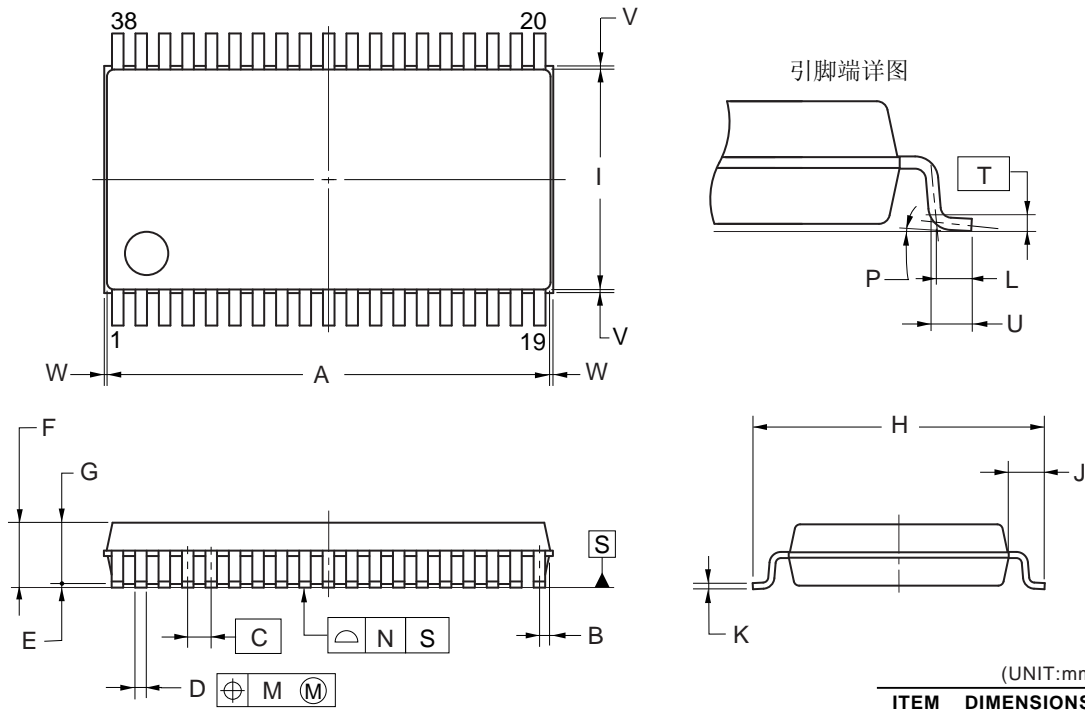
ITEM	DIMENSIONS
A	9.70±0.10
B	0.30
C	0.65 (T.P.)
D	0.22 ^{+0.10} _{-0.05}
E	0.10±0.05
F	1.30±0.10
G	1.20
H	8.10±0.20
I	6.10±0.10
J	1.00±0.20
K	0.15 ^{+0.05} _{-0.01}
L	0.50
M	0.13
N	0.10
P	3° ^{+5°} _{-3°}
T	0.25(T.P.)
U	0.60±0.15
V	0.25 MAX.
W	0.15 MAX.

P30MC-65-CAB

© NEC Electronics Corporation 2005

• μ PD179F122MC-GAA-AX, 179F123MC-GAA-AX, 179F124MC-GAA-AX

38 引脚塑封 SSOP (7.62mm (300))



注

该结构在最大使用材料情况下，每条引脚的中心线位于其实际位置的 0.1mm 内

P38MC-65-GAA

附录 A 开发工具

在使用 μ PD179F11x, 179F12x 微控制器的系统开发中可应用如下开发工具。

图 A-1 所示为开发工具的组成。

- **支持 PC98-NX 系列**

除非特别说明，IBM PC/AT™ 兼容机支持的产品与 PC98-NX 系列计算机是兼容的。当使用 PC98-NX 系列计算机时，参阅 IBM PC/AT 兼容机的使用说明。

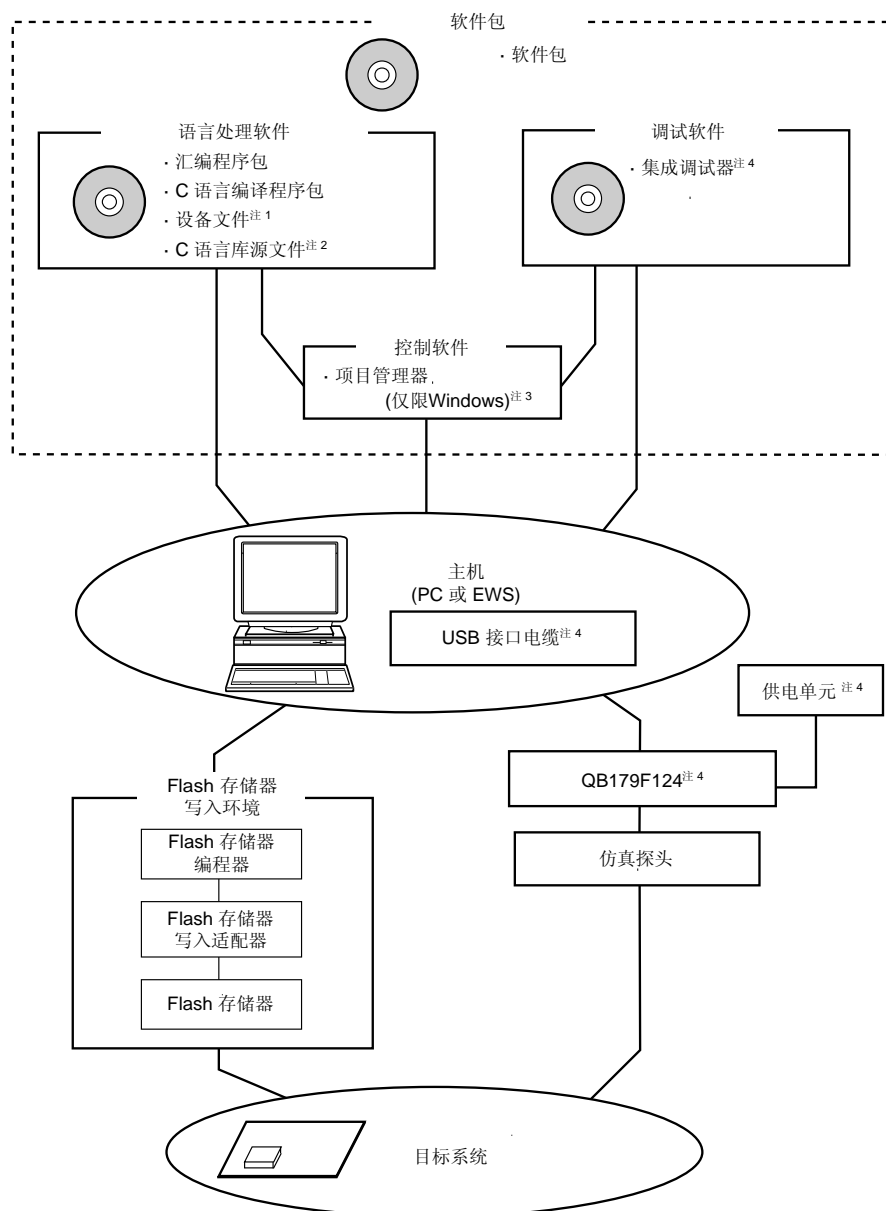
- **Windows™**

除非特别说明，“Windows”指以下几种操作系统。

- Windows 98
- Windows NT™
- Windows 2000
- Windows XP

图 A-1. 开发工具的组成 (1/2)

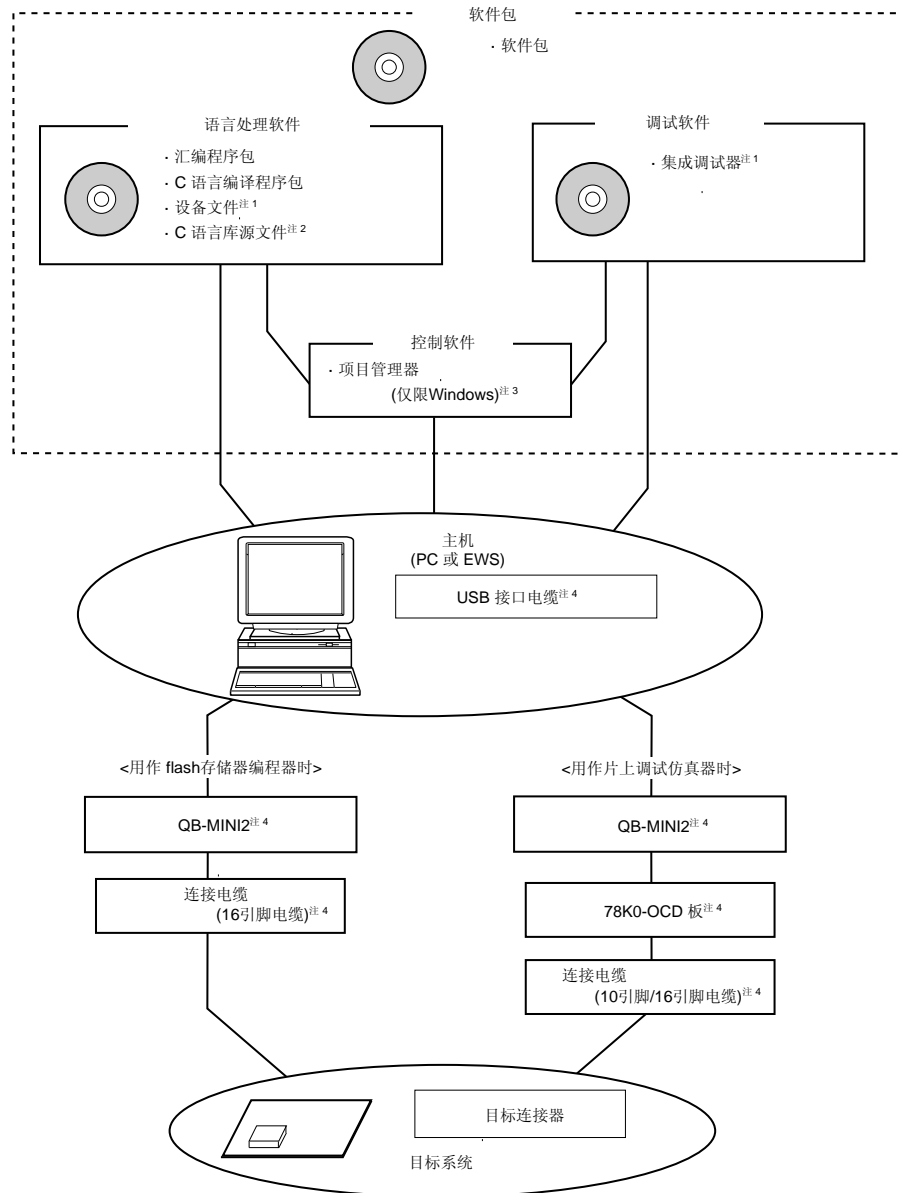
(1) 使用在线仿真器 QB-179F124 时



- 注
1. 从开发工具的下载地址中为 μ PD179F11x, 179F12x 微控制器下载设备文件 (DF179124) (<http://www.necel.com/micro/ods/eng/index.html>)。
 2. C 语言库源文件不包含在软件包中。
 3. 项目管理器 PM+ 包含在汇编程序包中。
PM+ 仅用于 Windows。
 4. QB-179F124 提供了集成调试器 ID78K0-QB, 一个 USB 接口电缆, 一个电源部件, 片上调试仿真器 QB-MINI2, 连接电缆 (10 引脚和 16 引脚电缆), 以及 78K0-OCD 板。所有其他产品都需单独购买。
使用 QB-MINI2 时, 可以从开发工具的下载站点中下载用于操作 QB-MINI2 的软件 (<http://www.necel.com/micro/ods/eng/index.html>)。

图 A-1. 开发工具的组成 (2/2)

(2) 使用带有编程功能的片上调试仿真器 QB-MINI2



- 注
1. 从开发工具的下载地址中为 μ PD179F11x, 179F12x微控制器和集成调试器ID78K0-QB下载设备文件 (DF179124) (<http://www.necel.com/micro/ods/eng/index.html>)。
 2. C 语言库源文件不包含在软件包中。
 3. 项目管理器 PM+包含在汇编程序包中。
PM+ 仅用于 Windows。
 4. 片上调试仿真器QB-MINI2 提供有一个USB接口电缆, 连接电缆 (10 引脚和 16 引脚电缆) 以及 78K0-OCD 板。所有其他产品都需单独购买。从开发工具的下载站点中下载用于操作 QB-MINI2 的软件 (<http://www.necel.com/micro/ods/eng/index.html>)。

A.1 软件包

SP78K0 78K0 微控制器软件包	该软件包包含了适用于 78K0 微控制器以及 μ PD179F11x, 179F12x 微控制器的开发工具 (软件)。
	产品型号: μ SxxxxSP78K0

备注 产品型号中的xxxx 随主机和使用的 OS 而变化。

μ SxxxxSP78K0

xxxx	主机	OS	存储介质
AB17	PC-9800 系列,	Windows (日语版)	CD-ROM
BB17	IBM PC/AT 兼容机	Windows (英语版)	

A.2 语言处理软件

RA78K0 汇编程序包	<p>该汇编器将有助于记符来写的程序转换为微控制器可执行的目标代码。该汇编器还带有自动创建符号表的功能以及优先跳转指令的功能。该汇编器应该同设备文件 (DF179124) (需另外购买) 结合使用。</p> <p><在 PC 环境中使用 RA78K0 时的预防措施></p> <p>该汇编程序包是基于 DOS 的应用程序。然而, 它也可以通过使用项目管理器 (包含于汇编程序包) 在 Windows 中进行使用。</p> <p>产品型号: μSxxxxRA78K0</p>
CC78K0 C 语言编译程序包	<p>该编译器将以 C 语言所写的程序转换为微控制器可执行的目标代码。该汇编器应该同汇编程序包以及设备文件 (均需另外购买) 结合使用。</p> <p><在 PC 环境中使用 CC78K0 时的预防措施></p> <p>该 C 语言编译程序包是基于 DOS 的应用程序。然而, 它也可以通过使用项目管理器 (包含于汇编程序包) 在 Windows 中进行使用。</p> <p>产品型号: μSxxxxCC78K0</p>
DF179124 ^{注1} 设备文件	<p>该文件包含设备特有的信息。该设备文件应该同工具 (RA78K0, CC78K0, 以及 ID78K0-QB) (均需另外购买) 结合使用。</p> <p>相应的 OS 和主机随使用工具而变化。</p> <p>产品型号: μSxxxxDF179124</p>
CC78K0-L ^{注2} C 语言库源文件	<p>该函数源文件用于配置 C 语言编译程序包中所包含的目标库。根据用户规范, 该文件需与 C 语言编译程序包中所包含的目标库匹配。</p> <p>产品型号: μSxxxxCC78K0-L</p>

- 注
1. DF179124 可以和 RA78K0, CC78K0, 和 ID78K0-QB 一起使用。从开发工具的下载网站中下载 DF179124 (<http://www.necel.com/micro/ods/eng/index.html>)。
 2. CC78K0-L 不包含在软件包 (SP78K0) 中。

备注 产品型号中的xxxx 随主机和使用的 OS 而变化。

μSxxxxRA78K0

μSxxxxCC78K0

μSxxxxCC78K0-L

xxxx	主机	OS	存储介质
AB17	PC-9800 系列, IBM PC/AT 兼容机	Windows (日语版)	CD-ROM
BB17		Windows (英语版)	
3P17	HP9000 系列 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARC 工作站™	SunOS™ (Rel. 4.1.4) Solaris™ (Rel. 2.5.1)	

μSxxxxDF179124

xxxx	主机	OS	存储介质
AB13	PC-9800 系列, IBM PC/AT 兼容机	Windows (日语版)	3.5-英寸 2HD FD
BB13		Windows (英语版)	

A.3 控制软件

PM+ 项目管理器	<p>这是一个控制软件，可使用户在 Windows 环境下有效地进行程序开发。用户程序开发中进行的任何操作，如启动编辑器、构造程序和启动调试器，都可以由项目管理器来执行。</p> <p><注意事项> 项目管理器包含于汇编程序包 (RA78K0) 中。 项目管理器仅在 Windows 下使用。</p>
--------------	---

A.4 Flash存储器写入工具

<R> A.4.1 使用flash存储器编程器PG-FP4, FL-PR4, PG-FP5 和 FL-PR5 时

PG-FP4, FL-PR4, PG-FP5, FL-PR5 Flash 存储器编程器	Flash 存储器编程器, 专用于带有片上 flash 存储器的微控制器。
FA-179F114MC-CAB-MX FA-179F124MC-GAA-MX Flash 存储器编程适配器	Flash 存储器写入适配器用于连接 Flash 存储器编程器。 <ul style="list-style-type: none"> • FA-179F114MC-CAB-MX: For 30 引脚塑封 SSOP (MC-CAB 类型) • FA-179F124MC-GAA-MX: For 38 引脚塑封 SSOP (MC-GAA 类型)

- 备注**
1. FL-PR4, FL-PR5, FA-179F114MC-CAB-MX 以及 FA-179F124MC-GAA-MX 是 Naito Densei Machida Mfg. Co., Ltd 的产品。
TEL: +81-42-750-4172 Naito Densei Machida Mfg. Co., Ltd.
 2. 请使用 flash 存储器编程适配器的最新版本。

A.4.2 使用带有编程功能的片上调试仿真器QB-MINI2

QB-MINI2 带有编程功能的片上调试仿真器	这是一个简易 flash 存储器编程器, 专用于带有片上 flash 存储器的微控制器。使用 μ PD179F11x, 179F12x 微控制器开发应用系统时, 它也可以用作调试硬件和软件的片上调试仿真器。用作 flash 存储器编程器时, 该仿真器应与连接电缆 (16 引脚电缆) 以及用于连接主机的 USB 接口电缆结合使用。
目标连接器规格	16 引脚通用连接器 (2.54 mm 脚距)

- 备注**
1. QB-MINI2 提供有一个 USB 接口电缆, 连接电缆 (10 引脚和 16 引脚电缆) 以及 78K0-OCD 板。只有在使用片上调试功能时才会使用连接电缆 (10 引脚电缆) 以及 78K0-OCD 板。
 2. 从开发工具的下载站点中下载用于操作 QB-MINI2 的软件 ([http : //www.necel.com/micro/ods/eng/index.html](http://www.necel.com/micro/ods/eng/index.html)) 。

A.5 调试工具（硬件）

A.5.1 使用在线仿真器QB-179F124 时

QB-179F124 在线仿真器	当使用 μ PD78F073x μ PD179F11x, 179F12x 微控制器进行应用系统开发时, 在线仿真器用来调试硬件和软件。它支持集成调试器 (ID78K0-QB)。该仿真器应与供电单元以及仿真探头结合使用, 而 USB 用于将该仿真器连接到主机上。
QB-144-CA-01 检测引脚适配器	该检测引脚适配器用于使用示波器等波形监视。
QB-80-EP-01T 仿真探头	该仿真探头可灵活使用, 用于连接在线仿真器和目标系统。
QB-30MC-EA-02T, QB-38MC-EA-02T 交换适配器	该交换适配器用于实现从在线仿真器到目标连接器的引脚转换。 <ul style="list-style-type: none"> • QB-30MC-EA-02T: 用于 30 引脚塑封 SSOP (MC-CAB 类型) • QB-38MC-EA-02T: 用于 38 引脚塑封 SSOP (MC-GAA 类型)
QB-30MC-YS-01T, QB-38MC-YS-01T 空间适配器	该空间适配器用于调整目标系统与在线仿真器间的高度。 <ul style="list-style-type: none"> • QB-30MC-YS-01T: 用于 30 引脚塑封 SSOP (MC-CAB 类型) • QB-38MC-YS-01T: 用于 38 引脚塑封 SSOP (MC-GAA 类型)
QB-30MC-YQ-01T, QB-38MC-YQ-01T YQ 连接器	该 YQ 连接器用于连接目标连接器和交换适配器。 <ul style="list-style-type: none"> • QB-30MC-YQ-01T: 用于 30 引脚塑封 SSOP (MC-CAB 类型) • QB-38MC-YQ-01T: 用于 38 引脚塑封 SSOP (MC-GAA 类型)
QB-30MC-HQ-01T, QB-38MC-HQ-01T 安装适配器	该安装适配器用于安装带有插槽的目标设备。 <ul style="list-style-type: none"> • QB-30MC-HQ-01T: 用于 30 引脚塑封 SSOP (MC-CAB 类型) • QB-38MC-HQ-01T: 用于 38 引脚塑封 SSOP (MC-GAA 类型)
QB-30MC-NQ-01T, QB-38MC-NQ-01T 目标连接器	该目标连接器用于安装在目标系统上。 <ul style="list-style-type: none"> • QB-30MC-NQ-01T: 用于 30 引脚塑封 SSOP (MC-CAB 类型) • QB-38MC-NQ-01T: 用于 38 引脚塑封 SSOP (MC-GAA 类型)

备注 1. QB-179F124 提供了集成调试器 ID78K0-QB, 一个 USB 接口电缆, 一个电源部件, 片上调试仿真器 QB-MINI2, 连接电缆 (10 引脚和 16 引脚电缆), 以及 78K0-OCD 板。

使用 QB-MINI2 时, 可以从开发工具的下站点中下载用于操作 QB-MINI2 的软件 (<http://www.necel.com/micro/ods/eng/index.html>)。

2. 封装内容随产品型号而变化, 如下所示。

封装内容 产品代号	在线仿真器	仿真探头	交换适配器	YQ 连接器	目标连接器
QB-179F124-ZZZ	QB-179F124	无			
QB-179F124-T30MC		QB-80-EP-01T	QB-30MC-EA-02T	QB-30MC-YQ-01T	QB-30MC-NQ-01T
QB-179F124-T38MC			QB-38MC-EA-02T	QB-38MC-YQ-01T	QB-38MC-NQ-01T

A.5.2 使用带有编程功能的片上调试仿真器QB-MINI2 时

QB-MINI2 带有编程功能的片上调试仿真器	当使用 μ PD179F11x, 179F12x 微控制器进行应用系统开发时, 片上调试仿真器用来调试硬件和软件。它也可以用作 flash 存储器编程器, 专用于带有片上 flash 存储器的微控制器。用作片上调试仿真器时, 该仿真器应与连接电缆 (10 引脚电缆或 16 引脚电缆) 用于连接主机的 USB 接口电缆, 以及 78K0-OCD 板结合使用。
目标连接器规格	10 引脚通用连接器 (2.54 mm 脚距) 或 16 引脚通用连接器 (2.54 mm 脚距)

- 备注**
1. QB-MINI2 提供有一个 USB 接口电缆, 连接电缆 (10 引脚和 16 引脚电缆) 以及 78K0-OCD 板。只有在使用片上调试功能时才会使用连接电缆 (10 引脚电缆) 以及 78K0-OCD 板。
 2. 从开发工具的下载站点中下载用于操作 QB-MINI2 的软件 ([http : //www.necel.com/micro/ods/eng/index.html](http://www.necel.com/micro/ods/eng/index.html)) 。

A.6 调试工具 (软件)

ID78K0-QB 集成调试器	该调试器支持 78K0 微控制器和 μ PD179F11x, 179F12x 微控制器的在线仿真器。 ID78K0-QB 是基于 Windows 的软件。 它改善了 C 语言兼容的调试功能, 并且能够通过使用结合了源程序, 分别显示以及带有跟踪结果的存储器显示的集成窗口功能来显示源程序的追踪结果。它应该同设备文件 (单独销售) 一起使用。
	产品型号: μ SxxxxID78K0-QB

备注 产品型号中的xxxx随主机和使用的 OS 而变化。

μ SxxxxID78K0-QB

xxxx	主机	OS	存储介质
AB17	PC-9800 系列,	Windows (日语版)	CD-ROM
BB17	IBM PC/AT 兼容机	Windows (英语版)	

附录B 修订记录

B.1 在此版本中的主要修订

(1/2)

所在页	说明
第一章 概述	
p. 16	修改了 1.3 订购信息
p. 20	修改了 1.8 功能概述中的内部低速振荡时钟（用于 WDT）
第二章 引脚功能	
p. 28	修改了表 2-1. I/O 引脚电路类型
第三章 CPU 架构	
p. 55	修改了 3.3.3 表间接寻址
第四章 端口功能	
p. 68	修改了图 4-2. P00 和 P03 的框图
p. 83	修改了 4.3 (3) 上拉电阻选择寄存器（PU0 到 PU2, PU3 和 PU12）
p. 86	修改了 4.5 使用复用功能时端口模式寄存器，输出锁存，上拉电阻选择寄存器以及端口输出模式寄存器的设置
第五章 时钟发生器	
p. 88	修改了 5.1 时钟发生器的功能中的 (2) 内部低速振荡时钟（看门狗定时器的时钟）。
p. 90	修改了图 5-1. 时钟发生器的框图
p. 92	从图 5-2. 时钟操作模式选择寄存器（OSCCTL）的格式中删除了注意事项 1
p. 99	在图 5-9. X1 振荡器外部电路示例中添加了注意事项
p. 101	修改了 5.4.3 内部低速振荡器
p. 107	添加了 5.6.3 控制内部低速振荡时钟的示例
p. 111	在表 5-4. CPU 时钟转换以及 SFR 寄存器设置示例 (2/2) 中添加了注意事项
第八章 8 位定时器 H0 和 H1	
p. 205	修改了图 8-6. 8 位定时器 H 模式寄存器 1（TMHMD1）的格式
第九章 看门狗定时器	
p. 228	修改了 9.4.1 看门狗定时器的控制操作中的注意事项 5
p. 228	修改了表 9-3. 看门狗定时器溢出时间的设置（当 $2.1\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ 时）
p. 229	添加了表 9-4. 看门狗定时器溢出时间的设置（当 $1.8\text{ V} \leq V_{DD} < 2.1\text{ V}$ 时）
p. 230	修改了表 9-5. 设置看门狗定时器窗口打开周期中的注意事项和备注
第十三章 待机功能	
p. 283	删除了 13.1.1 待机功能中的注意事项 1
p. 284	修改了图 13-1. 振荡稳定时间计数状态寄存器（OSTC）的格式
p. 285	修改了图 13-2. 振荡稳定时间选择寄存器（OSTS）的格式
p. 290	在表 13-3. STOP 模式中的操作状态中添加了注意事项 4
第十七章 选项字节	
p. 324	修改了图 17-1. 选项字节的格式 (1/2)

所在页	说明
第十八章 FLASH 存储器	
p. 327	修改了表 18-2. μ PD179F11x 微控制器和专用 Flash 存储器编程器之间的配线示例中的注
p. 328	修改了图 18-2. 用于 Flash 存储器写入的适配器的配线示例 (30 引脚产品) 中的备注
p. 329	修改了图 18-3. 写程序到 Flash 存储器所需要的环境
p. 329	修改了图 18-4. 与专用 Flash 存储器编程器的通信
p. 330	修改了表 18-3. 引脚连接中的注
p. 333	修改了 18.5.6 其他信号引脚
p. 334	修改了图 18-9. Flash 存储器操作过程
p. 335	修改了图 18-11. 通信命令
p. 336	修改了表 18-7. Flash 存储器控制命令
p. 337	修改了 18.7 安全设置
p. 337	修改了表 18-9. 允许使用的安全功能和命令之间的关系
p. 338	修改了表 18-10. 每一种编程模式下的安全设置
第二十章 指令集	
p. 353	修改了 20.2 操作列表
第二十一章 电气特性	
p. 367	修改了 Flash 存储器编程特性 ($T_A = -40$ 到 $+85^\circ\text{C}$, $2.0\text{ V} \leq V_{DD} \leq 3.6\text{ V}$, $V_{SS} = 0\text{ V}$)
附录 A 开发工具	
p. 376	修改了 A.4.1 使用 flash 存储器编程器 PG-FP4, FL-PR4, PG-FP5 和 FL-PR5 时
附录 B 修订记录	
p. 380	添加了 B.2 先前版本的修订记录

<R> B.2 先前版本的修订记录

以下为先前版本的修订记录。章节为各版本中的。

版本	说明	适用于
第二版	添加了对 2.2.5 P120 到 P123 （端口 12）的描述	第二章 引脚功能
	修改了表 3-6. 特殊功能寄存器列表	第三章 CPU 结构
	修改了表 4-2. 端口配置	第四章 端口功能
	修改了图 4-14. P123 的框图	
	添加了对 4.3 控制端口功能的寄存器的描述	
	修改了图 4-17. 上拉电阻选择寄存器的格式	
	添加了 4.3 (5) 复位引脚模式寄存器 (RSTMASK)	
	添加了对 12.1 按键中断功能的描述	
	修改了表 14-2. 复位响应后的硬件状态	第 14 章 复位功能
	添加了对 16.1 F 低电压检测器的功能的描述	第十六章 低电压检测器
	添加了 16.6 RAM 数据保存检测器	
	修改了表 18-7. Flash 存储器控制命令	第十八章 FLASH 存储器
	修改了 18.7 安全设置	
	添加了 18.8 通过自编程进行编程的 Flash 存储器	
	修改了图 19-1. QB-MINI2 和 μPD179F11x, 179F12x 微控制器连接示例 （当使用 OCD0A/X1 和 OCD0B/X2 时）	第十九章 片上调试功能
	修改了图 19-2. QB-MINI2 和 μPD179F11x, 179F12x 微控制器连接示例 （当使用 OCD1A/P04 和 OCD1B/P05 时）	
修改了第二十一章 电气特性	第二十一章 电气特性	

详细信息请联系：

中国区

MCU 技术支持热线：

电话：+86-400-700-0606 (普通话)

服务时间：9:00-12:00，13:00-17:00（不含法定节假日）

网址：

<http://www.cn.necel.com/>（中文）

<http://www.necel.com/>（英文）

[北京]

日电电子（中国）有限公司

中国北京市海淀区知春路 27 号

量子芯座 7，8，9，15 层

电话：（+86）10-8235-1155

传真：（+86）10-8235-7679

[深圳]

日电电子（中国）有限公司深圳分公司

深圳市福田区益田路卓越时代广场大厦 39 楼

3901，3902，3909 室

电话：（+86）755-8282-9800

传真：（+86）755-8282-9899

[上海]

日电电子（中国）有限公司上海分公司

中国上海市浦东新区银城中路 200 号

中银大厦 2409-2412 和 2509-2510 室

电话：（+86）21-5888-5400

传真：（+86）21-5888-5230

[香港]

香港日电电子有限公司

香港九龙旺角太子道西 193 号新世纪广场

第 2 座 16 楼 1601-1613 室

电话：（+852）2886-9318

传真：（+852）2886-9022

2886-9044

上海恩益禧电子国际贸易有限公司

中国上海市浦东新区银城中路 200 号

中银大厦 2511-2512 室

电话：（+86）21-5888-5400

传真：（+86）21-5888-5230

[成都]

日电电子（中国）有限公司成都分公司

成都市二环路南三段 15 号天华大厦 7 楼 703 室

电话：(+86)28-8512-5224

传真：(+86)28-8512-5334

[长春]

日电电子（中国）有限公司长春分公司

吉林省长春市朝阳区

西安大路 727 号中银大厦 A 座 1609 室

电话：(+86)431-8859-7533 / 8859-8533

传真：(+86)431-8680-2944

[大连]

日电电子（中国）有限公司长春分公司

大连市中山路 88 号天安国际大厦 2701 室

电话：(+86)411-8230-8815 / 8230-8825

传真：(+86)411-8230-8835