

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.



用户手册

μ PD78F8024, 78F8025

8 位单片微控制器

μ PD78F8024

μ PD78F8025

文档编号. U18976CA3V0UD00 (第 3 版)

发行日期 2009 年 7 月 CP(N)

© NEC Electronics Corporation 2007

日本印刷

[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。

EEPROM 是 **NEC Electronics Corporation** 的注册商标。

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

SuperFlash® 是 **Silicon Storage Technology, Inc.** 的注册商标，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 7 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。

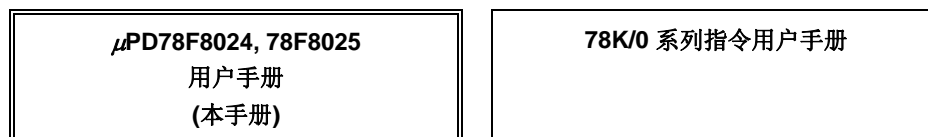
(注)

- (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。

M5 02.11-1

前言

- <R> 读者 本手册旨在使用户了解 μ PD78F8024 和 78F8025 的功能及设计开发应用系统。
- 目的 本手册旨在用户对功能描述的了解，以下为手册的**组织结构**。
- <R> 组织 μ PD78F8024 和 78F8025 手册分为两部分：本手册和指令部分 (与 78K0 微控制器通用)。



- | | |
|--|---|
| <ul style="list-style-type: none">• 引脚功能• 内部模块功能• 中断• 其他片上外围功能• 电气特性 | <ul style="list-style-type: none">• CPU 功能• 指令设置• 各指令说明 |
|--|---|

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。

- 要掌握全部功能:
→ 请按**目录**顺序阅读本手册 标注“<R>”表示主要修改部分。 修订部分可以很方便地通过在 PDF 文件中拷贝“<R>”并通过在“查找:”中指定来搜索查询。
- 如何解释寄存器格式:
→ 尖括号(<>)中的二进制位名称在 RA78K0 中被定义为保留字，并且在 CC78K0 中用#pragma sfr 指令定义为一个 sfr 变量。
- 要查找详细的寄存器名称
→ 请参阅 **附录 C 寄存器索引**
- 要了解 78K0 微控制器指令功能的详细内容
→ 请参考 **78K/0 微控制器指令用户手册 (U12326E)**。

规则

数据规则:	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:	$\overline{\text{xxx}}$ 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

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	文档编号
μPD78F8024, 78F8025 用户手册	本手册
78K0 系列指令用户手册	U12326E
78K0/Kx2 Flash 存储器编程(编程器)应用笔记	U17739E
78K0 微控制器自编程库 Type01 用户手册	U18274E
78K0 微控制器 EEPROM™ 仿真库 Type01 用户手册	U18275E

Flash存储器编程相关文档 (用户手册)

文档名称	文档编号
PG-FP5 存储器编程器	U18865E
PG-FP4 存储器编程器	U15260E
QB-MINI2 具有编程功能的片上调试仿真器	U18371E

开发工具相关文档 (硬件) (用户手册)

文档名称	文档编号
QB-78K0KX2 片上仿真器	U17341E
QB-78F8024-EA-01T 交换适配器	U19135X

注意事项: 以上列出的相关文档可能会在任何声明条件下修改。开发设计时，请使用各文档的最新版本。

开发工具相关文档(软件)

文档名称		文档编号
RA78K0 Ver.3.80 汇编包 用户手册 ^{注 1}	操作篇	U17199E
	语言篇	U17198E
	结构化汇编语言篇	U17197E
78K0 汇编包 RA78K0 Ver.4.01 操作预注意事项(告示文档) ^{注 1}		ZUD-CD-07-0181-E
CC78K0 Ver.3.70 C 编译器 用户手册 ^{注 2}	操作篇	U17201E
	语言篇	U17200E
78K0 C 编译器 CC78K0 Ver. 4.00 操作预注意事项(告示文档) ^{注 2}		ZUD-CD-07-0103-E
SM+ 系统仿真器 用户手册	操作篇	U18601E
	用户开放接口	U18212E
ID78K0-QB Ver.2.94 集成调试器 用户手册	操作篇	U18330E
ID78K0-QB Ver.3.00 集成调试器 用户手册	操作篇	U18492E
PM plus Ver.5.20 用户手册 ^{注 3}		U16934E
PM+ Ver.6.30 用户手册 ^{注 4}		U18416E

- 注**
1. 本手册将在安装工具 RA78K0 Ver. 4.01 时一起安装到 PC 中,有关说明不包含在“78K0 汇编包 RA78K0 Ver. 4.01 操作预注意事项”,请参考 RA78K0 Ver. 3.80 用户手册。
 2. 本手册将在安装工具 CC78K0 Ver. 4.00 时一起安装到 PC 中,有关说明不包含在“78K0 C 编译器 CC78K0 Ver. 4.00 操作预注意事项”,请参考 CC78K0 Ver. 3.70 用户手册。
 3. PM plus Ver. 5.20 是集成开发环境,包含在 RA78K0 Ver. 3.80 的压缩包中。
 4. PM+ Ver. 6.30 是集成开发环境,包含在 RA78K0 Ver. 4.01 的压缩包中,可以管理不同版本的软件工具(汇编器, C 编译器, 调试器和仿真器)。

其他文档

文档名称	文档编号
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 (ESD)	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何声明条件下修改。 开发设计时, 请使用各文档的最新版本。

目录

第一章 概述	16
1.1 特点	16
1.2 应用	17
1.3 订购信息	17
1.4 引脚构成（俯视图）	18
1.5 配置	20
1.6 框图	21
1.7 功能概述	22
第二章 引脚功能	24
2.1 引脚功能列表	24
2.1.1 CPU 部分	24
2.1.2 带有可选降压或升压转换器的恒流驱动器部分	27
2.2 引脚功能描述（CPU 部分）	28
2.2.1 P00 和 P01（端口 0）	28
2.2.2 P10 至 P17（端口 1）	28
2.2.3 P20 至 P23（端口 2）	29
2.2.4 P30 至 P33（端口 3）	30
2.2.5 P60 和 P61（端口 6）	30
2.2.6 P120 至 P122（端口 12）	31
2.2.7 AVREF	31
2.2.8 AVSS	31
2.2.9 RESET	31
2.2.10 REGC	32
2.2.11 VDD	32
2.2.12 VSS	32
2.2.13 FLMD0	32
2.3 引脚功能描述（恒流驱动器部分）	33
2.3.1 EN（内部引脚）	33
2.3.2 PWM0 至 PWM3（内部引脚）	33
2.3.3 SH（内部引脚）	33
2.3.4 DRV0 至 DRV3	33
2.3.5 MODE	33
2.3.6 SENSEA0 至 SENSEA3	33
2.3.7 SENSEB0 至 SENSEB3	33
2.3.8 CCA0 至 CCA3, CCB0 至 CCB3	33
2.3.9 CT	33
2.3.10 RT	33
2.3.11 CVDD0 至 CVDD2	33
2.3.12 VIN	33
2.3.13 AGND	34
2.3.14 PGND0 至 PGND3, PGND11	34
2.3.15 IC0	34
2.4 I/O 引脚电路以及不使用引脚的推荐连接	35

第三章 带有可选降压或升压转换器的恒流驱动器	39
3.1 带有可选降压或升压转换器的恒流驱动器的功能	39
3.2 升压模式	40
3.3 降压模式	41
3.4 工作模式设置步骤	42
第四章 CPU 架构	44
4.1 存储空间	44
4.1.1 内部程序存储空间	47
4.1.2 内部数据存储空间	48
4.1.3 特殊功能寄存器 (SFR) 区域	48
4.1.4 数据存储寄存器寻址	49
4.2 处理器寄存器	52
4.2.1 控制寄存器	52
4.2.2 通用寄存器	56
4.2.3 特殊功能寄存器 (SFR)	57
4.3 指令地址寻址	61
4.3.1 相对寻址	61
4.3.2 立即寻址	62
4.3.3 表间接寻址	63
4.3.4 寄存器寻址	64
4.4 操作数地址寻址	64
4.4.1 隐含寻址	64
4.4.2 寄存器寻址	65
4.4.3 直接寻址	66
4.4.4 短直接寻址	67
4.4.5 特殊功能寄存器 (SFR) 寻址	68
4.4.6 寄存器间接寻址	69
4.4.7 基址寻址	70
4.4.8 基址变址寻址	71
4.4.9 堆栈寻址	72
第五章 端口功能	73
5.1 端口功能	73
5.2 端口配置	75
5.2.1 端口 0	76
5.2.2 端口 1	78
5.2.3 端口 2	84
5.2.4 端口 3	85
5.2.5 端口 6	88
5.2.6 端口 12	89
5.3 寄存器控制端口功能	92
5.4 端口功能操作	96
5.4.1 写入 I/O 端口	96
5.4.2 从 I/O 端口中读取	96
5.4.3 I/O 端口上的操作	96
5.5 使用复用功能时端口模式寄存器以及输出锁存器的设置	97
5.6 端口寄存器 n (Pn) 1 位操作指令的注意事项	99

第六章 时钟发生器	100
6.1 时钟发生器的功能	100
6.2 时钟发生器的配置	101
6.3 控制时钟发生器的寄存器	103
6.4 系统时钟振荡器	111
6.4.1 X1 振荡器	111
6.4.2 内部高速振荡器	113
6.4.3 内部低速振荡器	113
6.4.4 预分频器	113
6.5 时钟发生器的操作	114
6.6 控制时钟	117
6.6.1 控制高速系统时钟的示例	117
6.6.2 控制内部高速振荡时钟的示例	120
6.6.3 控制内部低速振荡时钟的示例	122
6.6.4 提供给 CPU 以及外围硬件的时钟	123
6.6.5 CPU 时钟状态迁移图	124
6.6.6 更改 CPU 时钟前的条件以及更改 CPU 时钟后的处理	127
6.6.7 主系统时钟转换所需的时间	128
6.6.8 时钟振荡停止前的条件	129
6.6.9 外围硬件以及源时钟	129
第七章 16 位定时器 / 事件计数器 00	130
7.1 16 位定时器 / 事件计数器 00 的功能	130
7.2 16 位定时器 / 事件计数器 00 的配置	131
7.3 控制 16 位定时器 / 事件计数器 00 的寄存器	136
7.4 16 位定时器 / 事件计数器 00 的操作	144
7.4.1 间隔定时器操作	144
7.4.2 方波输出操作	147
7.4.3 外部事件计数器操作	150
7.4.4 通过 TI000 引脚有效沿输入进入的清零&启动模式中的操作	153
7.4.5 自由运行定时器操作	166
7.4.6 PPG 输出操作	175
7.4.7 单脉冲输出操作	178
7.4.8 脉冲宽度测量操作	183
7.5 TM00 的特殊用途	191
7.5.1 TM00 操作期间重新写入 CR010	191
7.5.2 设置 LVS00 和 LVR00	191
7.6 16 位定时器 / 事件计数器 00 的注意事项	193
第八章 8 位定时器 / 事件计数器 50 和 51	198
8.1 8 位定时器 / 事件计数器 50 和 51 的功能	198
8.2 8 位定时器 / 事件计数器 50 和 51 的配置	198
8.3 控制 8 位定时器 / 事件计数器 50 和 51 的寄存器	201
8.4 8 位定时器 / 事件计数器 50 和 51 的操作	207
8.4.1 作为间隔定时器工作	207
8.4.2 作为外部事件计数器工作	209
8.4.3 方波输出操作	210
8.4.4 PWM 输出操作	211

8.5 8 位定时器 / 事件计数器 50 和 51 的注意事项	215
第九章 8 位定时器 H0 和 H1	216
9.1 8 位定时器 H0 和 H1 的功能	216
9.2 8 位定时器 H0 和 H1 的配置	216
9.3 控制 8 位定时器 H0 和 H1 的寄存器	220
9.4 8 位定时器 H0 和 H1 的操作	226
9.4.1 作为间隔定时器 / 方波输出工作	226
9.4.2 作为 PWM 输出工作	229
9.4.3 载波发生器操作 (只有 8 位定时器 H1)	235
第十章 看门狗定时器	242
10.1 看门狗定时器的功能	242
10.2 看门狗定时器的配置	242
10.3 控制看门狗定时器的寄存器	244
10.4 看门狗定时器的操作	245
10.4.1 看门狗定时器的控制操作	245
10.4.2 设置看门狗定时器溢出时间	246
10.4.3 设置看门狗定时器的窗口打开周期	247
第十一章 A/D 转换器	249
11.1 A/D 转换器的功能	249
11.2 A/D 转换器的配置	250
11.3 在 A/D 转换器中使用的寄存器	252
11.4 A/D 转换器操作	261
11.4.1 A/D 转换器的基本操作	261
11.4.2 输入电压和转换结果	263
11.4.3 A/D 转换器工作模式	264
11.5 如何阅读 A/D 转换器特性表	266
11.6 A/D 转换器的注意事项	268
第十二章 串行接口 UART0	272
12.1 串行接口 UART0 的功能	272
12.2 串行接口 UART0 的配置	273
12.3 控制串行接口 UART0 的寄存器	276
12.4 串行接口 UART0 的操作	281
12.4.1 操作停止模式	281
12.4.2 异步串行接口 (UART) 模式	282
12.4.3 专用波特率发生器	288
12.4.4 波特率的计算	289
第十三章 串行接口 UART6	293
13.1 串行接口 UART6 的功能	293
13.2 串行接口 UART6 的配置	297
13.3 控制串行接口 UART6 的寄存器	300
13.4 串行接口 UART6 的操作	309
13.4.1 操作停止模式	309
13.4.2 异步串行接口 (UART) 模式	310

13.4.3 专用波特率发生器	323
13.4.4 波特率的计算	325
第十四章 串行接口 CSI10	330
14.1 串行接口 CSI10 的功能	330
14.2 串行接口 CSI10 的配置	331
14.3 控制串行接口 CSI10 的寄存器	333
14.4 串行接口 CSI10 的操作	337
14.4.1 操作停止模式	337
14.4.2 3线串行输入 / 输出模式	337
第十五章 串行接口 IIC0	348
15.1 串行接口 IIC0 的功能	348
15.2 串行接口 IIC0 的配置	351
15.3 控制串行接口 IIC0 的寄存器	354
15.4 I²C 总线模式功能	367
15.4.1 引脚配置	367
15.5 I²C 总线定义和控制方法	368
15.5.1 启动条件	368
15.5.2 地址	369
15.5.3 传输方向指定	369
15.5.4 响应 (ACK)	370
15.5.5 停止条件	371
15.5.6 等待	372
15.5.7 取消等待	374
15.5.8 中断请求 (INTIIC0) 产生时间和等待控制	374
15.5.9 地址匹配检测方法	375
15.5.10 错误检测	375
15.5.11 扩展码	376
15.5.12 仲裁	377
15.5.13 唤醒功能	378
15.5.14 通信预约	379
15.5.15 其它注意事项	382
15.5.16 通信操作	383
15.5.17 I ² C 中断请求 (INTIIC0) 发生的时序	391
15.6 时序图	412
第十六章 中断功能	419
16.1 中断功能类型	419
16.2 中断源和配置	419
16.3 控制中断功能的寄存器	422
16.4 中断服务操作	430
16.4.1 可屏蔽中断响应	430
16.4.2 软件中断请求响应	432
16.4.3 中断嵌套	433
16.4.4 中断请求保持	436
第十七章 待机功能	437

17.1 待机功能和配置	437
17.1.1 待机功能.....	437
17.1.2 控制待机功能的寄存器	437
17.2 待机功能操作.....	440
17.2.1 HALT 模式.....	440
17.2.2 STOP 模式.....	444
第十八章 复位功能	450
18.1 确定复位源的寄存器	458
第十九章 上电清零电路.....	459
19.1 上电清零电路的功能	459
19.2 上电清零电路的配置	460
19.3 上电清零电路的操作	460
19.4 上电清零电路的注意事项.....	463
第二十章 低电压检测电路	465
20.1 低电压检测电路的功能	465
20.2 低电压检测电路的配置	466
20.3 控制低电压检测电路的寄存器.....	466
20.4 低电压检测电路的操作	469
20.4.1 当用作复位时.....	470
20.4.2 当用作中断时.....	475
20.5 低电压检测电路的注意事项	480
20.5.1 当供电电压在 LVI 检测电压附近波动时	480
20.5.2 使用 LVI 作为复位信号时的注意事项.....	481
第二十一章 选项字节	482
21.1 选项字节的功能	482
21.2 字节选项的格式	483
第二十二章 FLASH 存储器	485
22.1 内部存储器容量切换寄存器	485
22.2 用 Flash 存储器编程器写入.....	486
22.3 编程环境	486
22.4 通信模式	486
22.5 板上引脚的操作	488
22.5.1 FLMD0 引脚	488
22.5.2 串行接口引脚.....	489
22.5.3 RESET 引脚	490
22.5.4 端口引脚.....	491
22.5.5 REGC 引脚.....	491
22.5.6 其它信号引脚.....	491
22.5.7 电源.....	491
22.6 编程方法	492
22.6.1 控制 flash 存储器	492
22.6.2 Flash 存储器编程模式	492
22.6.3 选择通信模式.....	493

22.6.4 通信命令.....	494
22.7 安全设置	495
22.8 Flash 存储器的自编程.....	497
22.8.1 启动交换功能	504
第二十三章 指令集.....	506
23.1 在操作列表中使用的约定.....	506
23.1.1 操作数标识符和描述方法.....	506
23.1.2 操作栏的说明	507
23.1.3 标志栏的说明	507
23.2 操作列表	508
23.3 按寻址类型列出指令	516
第二十四章 电气特性	519
24.1 CPU 部分.....	519
24.2 带有可选降压或升压转换器的恒流驱动器部分	540
第二十五章 封装图	543
第二十六章 推荐的焊接条件.....	544
第二十七章 等待注意事项	545
27.1 等待注意事项	545
27.2 产生等待的外围硬件	546
附录 A 开发工具	547
A.1 软件包	549
A.2 语言处理软件	549
A.3 Flash 存储器编程工具	550
A.3.1 使用 flash 存储器编程器 PG-FP5, FL-PR5, PG-FP4, 和 FL-PR4 时.....	550
A.3.2 使用带有编程功能的片上调试仿真器 QB-MINI2	550
A.4 调试工具（硬件）	551
A.5 调试工具（软件）	552
附录 B 目标系统设置时的注意事项.....	553
附录 C 寄存器索引.....	554
C.1 寄存器索引 (按字母排列寄存器名称)	554
C.2 寄存器索引 (按字母排列寄存器符号).....	557
附录 D 修改历史	560
D.1 本版本的主要修改内容.....	560
D.2 前一个版本的修改历史.....	562

第一章 概述

1.1 特点

- 带有可选降压或升压转换器的片上恒流驱动器
- 最短的指令执行时间从高速（0.1 μs : @20 MHz 高速系统时钟操作）至低速（1.6 μs : @20 MHz 高速系统时钟操作）可选
- 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	程序存储器 (ROM)		数据存储器 (内部高速 RAM)
	Flash 存储器		
$\mu\text{PD78F8024}$		8 KB ^{注1}	512 个字节 ^{注1}
$\mu\text{PD78F8025}$		32 KB ^{注1}	1KB ^{注1}

- 片上单电压供电 Flash 存储器
- 自编程（带有引导交换功能）
- 片上上电清零（POC）电路和低压检测器（LVI）
- 片上看门狗定时器（可使用片上内部低速振荡时钟）
- I/O 端口：23（N-ch 漏极开路：2）
- 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：3 个通道
 - UART（支持 LIN（局域互连网）总线）：1 个通道
 - CSI/UART ^{注2}：1 个通道
 - I²C：1 个通道
- 10 位分辨率 A/D 转换器（ $V_{\text{REF}} = 2.3$ 至 5.5 V ）：4 个通道
- 供电电压： $V_{\text{DD}} = 1.8$ 至 5.5 V ^{注3}
- 工作环境温度： $T_{\text{A}} = -40$ 至 $+85^{\circ}\text{C}$

- 注**
1. 复位释放后，通过内部存储器容量转换寄存器（IMS）对使用的内部 flash 存储器和内部高速 RAM 容量进行设置。
 2. 选择这些复用功能引脚功能中的任一功能。
 3. 使用恒流驱动器时： $V_{\text{DD}} = CV_{\text{DD}} = 4.5$ 至 5.5 V
未使用恒流驱动器时： $V_{\text{DD}} = 1.8$ 至 5.5 V

1.2 应用

- 加热器
- 电机
- LCD 背景光
- 照明

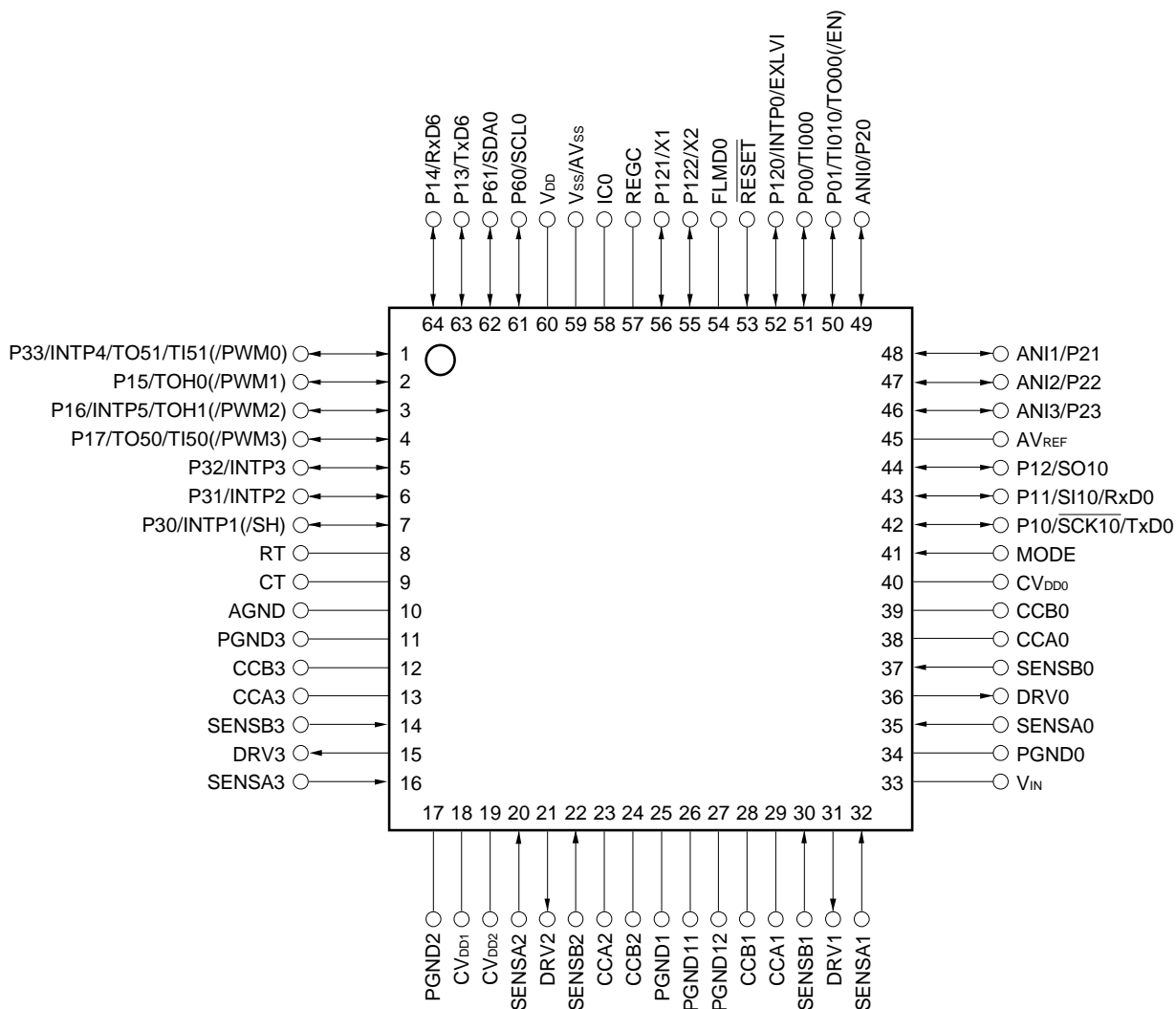
1.3 订购信息

- **Flash 存储器版本（无铅产品）**

产品代号	封装
μ PD78F8024GK-GAJ-AX	64-引脚塑封 LQFP (12x12)
μ PD78F8025GK-GAJ-AX	64-引脚塑封 LQFP (12x12)

1.4 引脚构成（俯视图）

- 64-引脚塑封 LQFP（12x12）



- 注意事项**
1. 通过电容将 REGC 引脚连接到 V_{SS} 上（推荐 0.47 至 1 μF）。
 2. 复位释放后 ANI0/P20 至 ANI3/P23 引脚设置为模拟输入模式。
 3. 直接将 IC0（内部连接）引脚连接至 V_{SS}。

备注 圆括号内的字符为内部引脚。

引脚标识

• CPU 部分

ANI0 至 ANI3:	模拟输入	RESET:	复位
AVREF:	模拟参考电压	RxD0, RxD6:	接收数据
AVSS:	模拟地电平	SCK10:	串行时钟输入 / 输出
EXCLK:	外部时钟输入 (主系统时钟)	SCL0:	串行时钟输入 / 输出
EXLVI:	用于低电压监测的 外部电位输入	SDA0:	串行数据输入 / 输出
FLMD0:	Flash 编程模式	SI10:	串行数据输入
INTP0 至 INTP5:	外部中断输入	SO10:	串行数据输出
P00, P01:	端口 0	TI000, TI010, TI150, TI151:	定时器输入
P10 至 P17:	端口 1	TO00,	
P20 至 P23:	端口 2	TO50, TO51,	
P30 至 P33:	端口 3	TOH0, TOH1:	定时器输出
P60, P61:	端口 6	TxD0, TxD6:	发送数据
P120 至 P122:	端口 12	V _{DD} :	供电电源
REGC:	调节器电容	V _{SS} :	地电平
		X1, X2:	晶体振荡器 (主系统时钟)

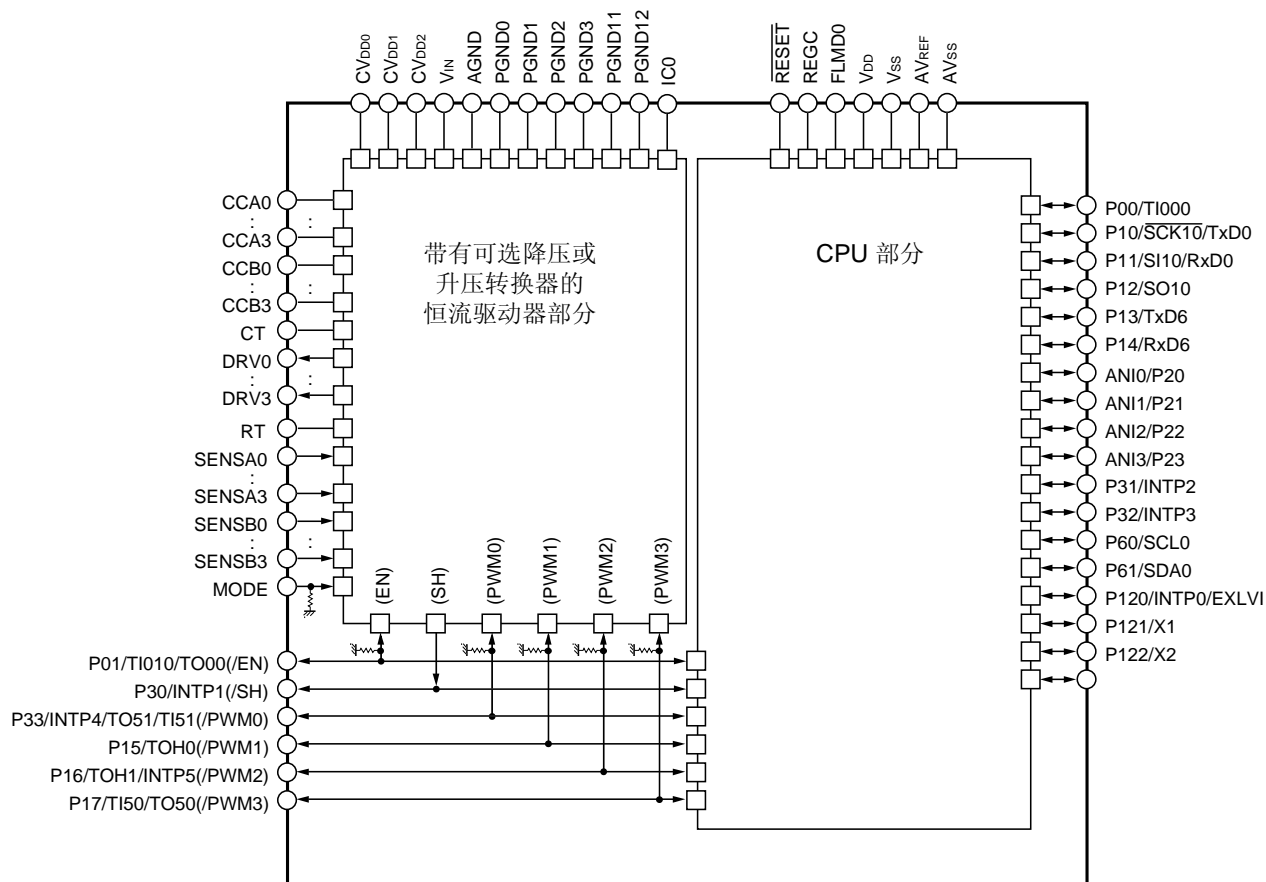
• 带有可选降压或升压转换器的恒流驱动器部分

AGND:	模拟地电平
CCA0-CCA3, CCB0-CCB3:	相位校正的电容
CT:	三角波振荡器的电容
CV _{DD0} -CV _{DD2} :	带有可选降压或升压转换器的恒流驱动器的电源
DRV0-DRV3: (EN):	N-ch MOSFET 的栅极驱动输出 允许下拉操作的使能信号 (200 kΩ (TYP.))
IC0:	内部连接
MODE:	带下拉的降压模式或升压模式选择输入 (200 kΩ (TYP.))
PGND0-PGND3, PGND11, PGND12:	地
(PWM0-PWM3):	使用下拉的调光控制的 PWM 脉冲输入 (200 kΩ (TYP.))
RT:	三角波振荡器的电阻
SENSA0-SENSA3: SENSB0-SENSB3: (SH):	升压模式中各通道的电流检测 热保护报警
V _{IN} :	高电压电源输入

备注 圆括号内的字符为内部引脚。

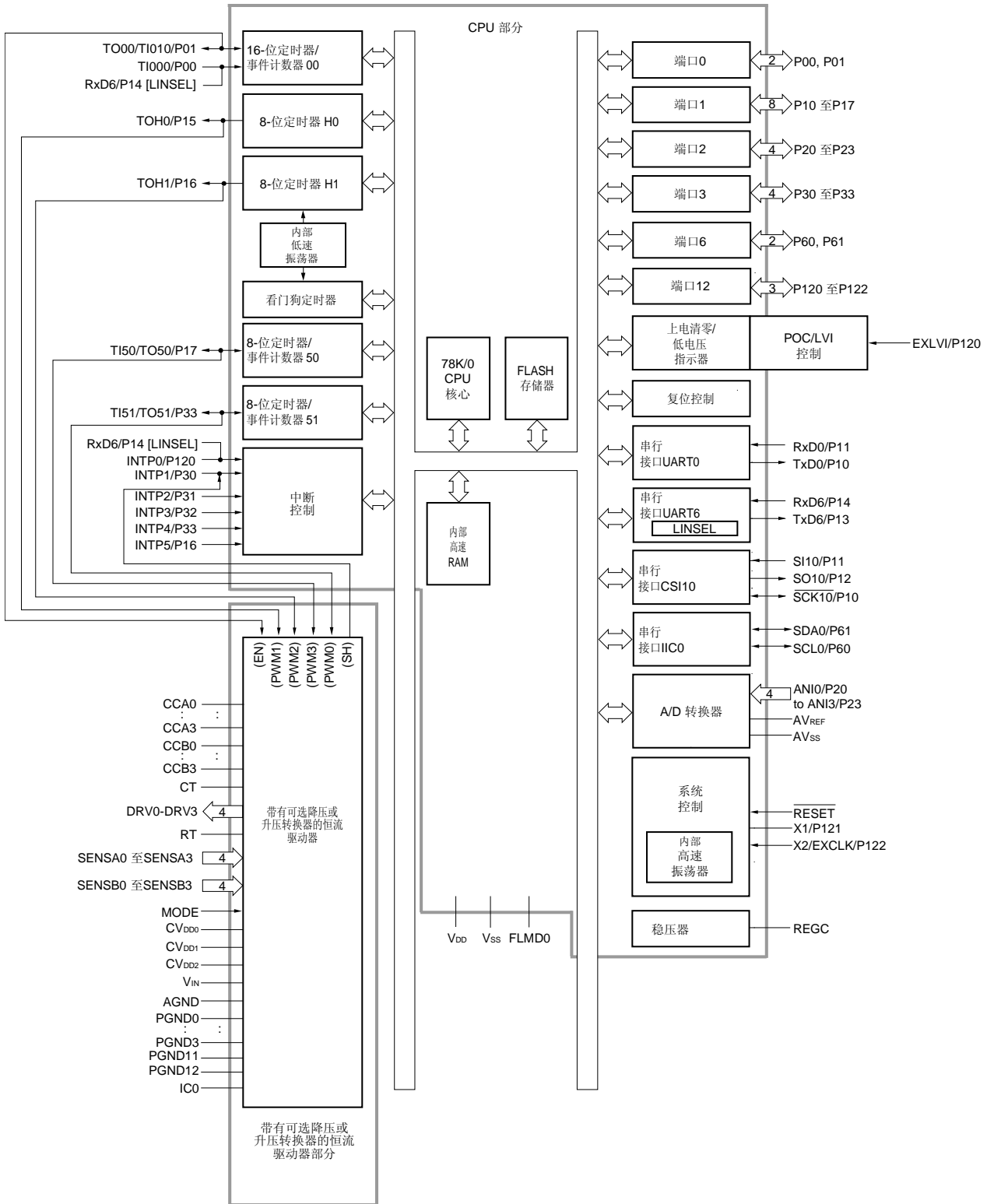
1.5 配置

μ PD78F8024 和 78F8025 为 SIP（系统级封装）产品，它由 CPU 部分以及带有可选降压或升压转换器的恒流驱动器部分组成。



备注 圆括号内的字符为内部引脚。

1.6 框图



备注 圆括号内的字符为内部引脚。

1.7 功能概述

(1/2)

项目		μ PD78F8024	μ PD78F8025
内部存储器	Flash 存储器 (支持自编程)	8 KB ^{注1}	32KB ^{注1}
	高速 RAM	512 个字节 ^{注1}	1KB ^{注1}
存储空间		64 KB	
主系统时钟 (振荡频率)	高速系统	X1 (晶体/陶瓷) 振荡, 外部主系统时钟输入 (EXCLK)	
	时钟	未使用恒流驱动器时	1 至 20 MHz: $V_{DD} = 2.7$ 至 5.5 V, 1 至 5 MHz: $V_{DD} = 1.8$ 至 5.5 V
		使用恒流驱动器时	1 至 20 MHz: $V_{DD} = CV_{DD} = 4.5$ 至 5.5 V
	内部高速振荡	内部振荡	
时钟	未使用恒流驱动器时	8 MHz (TYP.): $V_{DD} = 1.8$ 至 5.5 V	
	使用恒流驱动器时	8 MHz (TYP.): $V_{DD} = CV_{DD} = 4.5$ 至 5.5 V	
内部低速振荡时钟 (对于 TMH1, WDT)	内部振荡		
通用寄存器	未使用恒流驱动器时	240 kHz (TYP.): $V_{DD} = 1.8$ 至 5.5 V	
	使用恒流驱动器时	240 kHz (TYP.): $V_{DD} = CV_{DD} = 4.5$ 至 5.5 V	
通用寄存器	8 位 \times 32 个寄存器 (8 位 \times 8 个寄存器 \times 4 组)		
最短指令执行时间	0.1 μ s (高速系统时钟: @ $f_{XH} = 20$ MHz 操作)		
	0.25 μ s (内部高速振荡时钟: @ $f_{RH} = 8$ MHz (TYP.) 操作)		
指令集	<ul style="list-style-type: none"> • 8 位操作, 16 位操作 • 乘/除 (8 位 \times 8 位, 16 位 \div 8 位) • 位操作 (设置, 复位, 测试, 以及布尔操作) • BCD 调整等 		
I/O 端口	总数: <u>23</u> CMOS I/O: 21 N-ch 漏极开路 I/O (6 V 耐压值): 2		
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: 4, PPG 输出: 1)	
A/D 转换器	10 位分辨率 \times 4 个通道 ($AV_{REF} = 2.3$ 至 5.5 V)		
串行接口	<ul style="list-style-type: none"> • 支持 LIN 总线的 UART: 1 个通道 • 3-线串行 I/O/UART ^{注2}: 1 个通道 • IIC 总线: 1 个通道 		

- 注
1. 复位释放后, 通过设置内部存储器容量转换寄存器 (IMS) 对使用的内部 flash 存储器和内部高速 RAM 容量进行设置。
 2. 选择这些复用功能引脚功能中的任一功能。

项目		μ PD78F8024	μ PD78F8025
带有可选降压或升压转换器的恒流驱动器		<ul style="list-style-type: none"> 升压模式: 4 个通道 降压模式: 4 个通道 	
矢量中断源	内部	14	
	内部	6	
复位		<ul style="list-style-type: none"> 使用 RESET 引脚进行复位 通过看门狗定时器进行内部复位 通过上电清零进行内部复位 通过低电压检测器进行内部复位 	
供电电压	未使用恒流驱动器时	$V_{DD} = 1.8$ 至 5.5 V	
	使用恒流驱动器时	$V_{DD} = CV_{DD} = 4.5$ 至 5.5 V	
工作环境温度		$T_A = -40$ 至 $+85^\circ\text{C}$	
封装		64-引脚塑封 LQFP (12x12)	

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 和 51		8 位定时器 H0 和 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	-	-	-
	PPG 输出	1 次输出	-	-	-	-	-
	PWM 输出	-	1 次输出	1 次输出	1 次输出	1 次输出	-
	脉宽测量	2 次输入	-	-	-	-	-
	方波输出	1 次输出	1 次输出	1 次输出	1 次输出	1 次输出	-
	载频发生器	-	-	-	-	1 次输出 ^注	-
	看门狗定时器	-	-	-	-	-	1 个通道
中断源		2	1	1	1	1	-

注 TM51 和 TMH1 可以结合使用作为载频发生器模式。

第二章 引脚功能

2.1 引脚功能列表

2.1.1 CPU 部分

(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。 2 位 I/O 端口。 可以以 1 位为单位来指定输入 / 输出模式。 可以通过软件设置来指定使用片内上拉电阻。	输入端口	TI000
P01				TI010/TO00 (/EN)
P10	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。 可以通过软件设置来指定使用片内上拉电阻。	输入端口	$\overline{\text{SCK10}}/\text{TxD0}$
P11				S110/RxD0
P12				SO10
P13				TxD6
P14				RxD6
P15				TOH0 (/PWM1)
P16				TOH1/INTP5 (/PWM2)
P17				TI50/TO50 (/PWM3)
P20 至 P23	I/O	端口 2。 4 位 I/O 端口。 可以以 1 位为单位来指定输入 / 输出模式。	模拟输入	ANI0 to ANI3
P30	I/O	端口 3。 4 位 I/O 端口。 可以以 1 位为单位来指定输入 / 输出模式。 可以通过软件设置来指定使用片内上拉电阻。	输入端口	INTP1 (/SH)
P31				INTP2
P32				INTP3
P33				INTP4/TI51/TO51 (/PWM0)
P60	I/O	端口 6。 2 位 I/O 端口。 输出为 N-ch 漏极开路输出 (耐压 6V)。 可以以 1 位为单位来指定输入 / 输出模式。	输入端口	SCL0
P61				SDA0
P120	I/O	端口 12。 3 位 I/O 端口。 可以以 1 位为单位来指定输入 / 输出模式。 只有 P120 可以通过软件设置来指定使用片内上拉电阻。	输入端口	INTP0/EXLVI
P121				X1
P122				X2/EXCLK

备注 圆括号内的字符为内部引脚。

(2) 非端口功能 (1/2)

功能名称	I/O	功能	复位后	复用功能
ANI0 至 ANI3	输入	A/D 转换器模拟输入。	模拟输入	P20 至 P23
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	可以为外部中断请求输入指定有效沿（上升沿，下降沿，或双边沿）	输入端口	P120/EXLVI
INTP1				P30 (/SH)
INTP2				P31
INTP3				P32
INTP4				P33/TI51/TO51 (/PWM0)
INTP5				P16/TOH1 (/PWM2)
REGC	-	为内部操作连接一个稳定电容到稳压器的输出（2.5V）。通过电容连接到 V _{SS} 上（0.47 至 1 μ F：推荐）。	-	-
RESET	输入	系统复位输入	-	-
RxD0	输入	串行数据输入至 UART0	输入端口	P11/SI10
RxD6	输入	串行数据输入至 UART6		P14
TxD0	输出	从 UART0 输出串行数据	输入端口	P10/SCK10
TxD6	输出	从 UART6 输出串行数据		P13
SCK10	I/O	CSI10 的时钟输入 / 输出	输入端口	P10/TxD0
SI10	输入	串行数据输入至 CSI10		P11/RxD0
SO10	输出	从 CSI10 输出串行数据		P12
SCL0	I/O	I ² C 的 CSI10 输入 / 输出	输入端口	P60
SDA0	I/O	I ² C 的串行数据 I/O		P61
TI000	输入	外部计数时钟输入到 16 位定时器 / 事件计数器 00 捕捉触发信号输入到 16 位定时器 / 事件计数器 00 的捕捉寄存器（CR000, CR010）中	输入端口	P00
TI010	输入	捕捉触发信号输入到 16 位定时器 / 事件计数器 00 的捕捉寄存器（CR000）中	输入端口	P01/TO00 (/EN)
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	P17/TO50 (/PWM3)
TI51		外部计数时钟输入到 8 位定时器 / 事件计数器 51 中		P33/TO51/INTP4 (/PWM0)

备注 圆括号内的字符为内部引脚。

(2) 非端口功能 (2/2)

功能名称	I/O	功能	复位后	复用功能
TO00	输出	16 位定时器 / 事件计数器 00 输出	输入端口	P01/TI010 (/EN)
TO50	输出	8 位定时器 / 事件计数器 50 输出	输入端口	P17/TI50 (/PWM3)
TO51		8 位定时器 / 事件计数器 51 输出		P33/TI51/INTP4 (/PWM0)
TOH0	输出	8 位定时器 H0 输出	输入端口	P15 (/PWM1)
TOH1		8 位定时器 H1 输出		P16/INTP5 (/PWM2)
X1	-	为主系统时钟连接振荡器	输入端口	P121
X2	-		输入端口	P122/EXCLK
EXCLK	输入	主系统时钟的外部时钟输入	输入端口	P122/X2
V _{DD}	-	除 P20 至 P23 外的引脚的电源	-	-
AV _{REF}	-	A/D 转换器参考电压输入和 P20 至 P23 引脚以及 A/D 转换器的电源	-	-
V _{SS}	-	除 P20 至 P23 外的引脚的地电平	-	-
AV _{SS}	-	A/D 转换器地电平。使用电平与 V _{SS} 的电平一致。	-	-

备注 圆括号内的字符为内部引脚。

2.1.2 带有可选降压或升压转换器的恒流驱动器部分

功能名称	I/O	功能	复位后	复用功能
(/EN)	输入	带下拉操作的使能信号 (200 k Ω (TYP.))	输入端口	P01/TI010/TO00
(PWM0)	输入	带下拉 (200 k Ω (TYP.)) 的调光控制的 PWM 脉冲输入	输入端口	P33/INTP4/TO51/ TI51
(PWM1)				P15/TOH0
(PWM2)				P16/TOH1/INTP5
(PWM3)				P17/TI50/TO50
(SH)	输出	热保护报警输出	输入端口	P30/INTP1
DRV0 至 DRV3	输出	外部 MOSFET 的栅极驱动输出	输出	–
MODE	输入	使用下拉的降压模式或升压模式选择输入 (200 k Ω (TYP.))	输入	–
SENSA0 至 SENSA3	输入	降压模式中各通道的电流检测/升压模式中过压保护的电压监测	输入	–
SENSB0 至 SENSB3	输入	升压模式中各通道的电流检测	输入	–
CCA0 至 CCA3, CCB0 至 CCB3	–	相位校正的电容连接	–	–
CT	–	三角波振荡器的电容连接	–	–
RT	–	三角波振荡器的电阻连接	–	–
CV _{DD0} 至 CV _{DD2}	–	恒流驱动器的电源	–	–
V _{IN}	–	高压电源	–	–
AGND	–	模拟地电平	–	–
PGND0 至 PGND3, PGND11, PGND12	–	地	–	–
IC0	–	内部连接。直接将该引脚连接至 V _{SS} 。	–	–

备注 圆括号内的字符为内部引脚。

2.2 引脚功能描述（CPU部分）

2.2.1 P00 和 P01（端口 0）

P00 和 P01 用作 2 位 I/O 端口。这些引脚也用作定时器 I/O。

以下操作模式可以以 1 位为单位进行指定。

(1) 端口模式

P00 和 P01 用作 2 位 I/O 端口。P00 和 P01 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）来指定使用片上上拉电阻。

(2) 控制模式

P00 和 P01 用作定时器 I/O。

(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，也用于将捕捉触发信号输入到 16 位定时器 / 事件计数器 00 的捕捉寄存器（CR000，CR010）中。

(b) TI010

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

(c) TO00

这是 16 位定时器 / 事件计数器 00 的定时器输出引脚。

备注 P01/TI010/TO00 引脚可以对恒流驱动器部分的 EN 引脚（内部引脚）进行操作，这是因为它连接在恒流驱动器部分的 EN 引脚的封装之内。

2.2.2 P10 至 P17（端口 1）

P10 和 P17 用作 8 位 I/O 端口。这些引脚也用作外部中断请求输入，串行接口数据 I/O，时钟 I/O，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

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）来指定使用片上上拉电阻。

(2) 控制模式

P10 到 P17 引脚也用作外部中断请求输入，串行接口数据 I/O，时钟 I/O，以及定时器 I/O。

(a) SI10

这是串行接口 CSI10 的串行数据输入引脚。

(b) SO10

这是串行接口 CSI10 的串行数据输出引脚。

(c) SCK10

这是串行接口 CSI10 的串行时钟 I/O 引脚。

(d) RxD0

这是串行接口 UART0 的串行数据输入引脚。

(e) RxD6

这是串行接口 UART6 的串行数据输入引脚。

(f) TxD0

这是串行接口 UART0 的串行数据输出引脚。

(g) TxD6

这是串行接口 UART6 的串行数据输出引脚。

(h) TI50

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

(i) TO50

这是 8 位定时器 / 事件计数器 50 的定时器输出引脚。

(j) TOH0, TOH1

这些是 8 位定时器 H0 和 H1 的定时器输出引脚。

(k) INTP5

这是可以指定有效沿（上升沿，下降沿，或双边沿）的外部中断请求输入引脚。

备注 P15/TOH0, P16/TOH1/INTP5, 以及 P17/TI50/TO50 引脚可以对恒流驱动器部分的 PWM1, PWM2, 以及 PWM3 引脚（内部引脚）进行操作，这是因为它们分别连接在恒流驱动器部分的 PWM1, PWM2, 以及 PWM3 引脚的封装之内。

2.2.3 P20 至 P23（端口 2）

P20 至 P23 用作 4 位 I/O 端口。这些引脚也可以用作 A/D 转换器模拟输入的引脚。

以下操作模式可以以 1 位为单位进行指定。

(1) 端口模式

P20 至 P23 用作 4 位 I/O 端口。P20 至 P23 可以通过使用端口模式寄存器 2（PM2）以 1 位为单位设置为输入或输出端口。

(2) 控制模式

P20 至 P23 可用作 A/D 转换器模拟输入引脚（ANI0 至 ANI3）。将这些引脚用作模拟输入引脚时，请参见 11.6 A/D 转换器的注意事项中的 (5) ANI0/P20 至 ANI3/P23。

注意事项 复位释放后，在模拟输入模式中设置 ANI0/P20 至 ANI3/P23 引脚。

2.2.4 P30 至 P33（端口 3）

P30 和 P33 用作 4 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 和 P33 用作 4 位 I/O 端口。P30 和 P33 可以通过使用端口模式寄存器 3（PM3）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 3（PU3）来指定使用片上上拉电阻。

(2) 控制模式

P30 至 P33 用作外部中断请求输入以及定时器 I/O。

(a) INTP1 至 INTP4

这些可作为外部中断请求输入引脚，可指定有效沿（上升沿，下降沿，或双边沿）。

(b) TI51

这是 8 位定时器 / 事件计数器 51 的外部计数时钟输入引脚。

(c) TO51

这是 8 位定时器 / 事件计数器 51 的定时器输出引脚。

- 备注**
1. 通过 P30/INTP1 引脚可以获取恒流驱动器部分的 SH 引脚（内部引脚）信号，这是因为它连接在恒流驱动器部分的 SH 引脚的封装之内。
 2. P33/INTP4/TO51/TI51 引脚可以对恒流驱动器部分的 PWM0 引脚（内部引脚）进行操作，这是因为它连接在恒流驱动器部分的 PWM0 引脚的封装之内。

2.2.5 P60 和 P61（端口 6）

P60 和 P61 用作 2 位 I/O 端口。这些引脚也用作串行接口数据 I/O 和时钟 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

P60 和 P61 用作 2 位 I/O 端口。P60 和 P61 可以通过使用端口模式寄存器 6（PM6）以 1 位为单位设置为输入或输出端口。

输出为 N-ch 漏极开路输出（耐压 6V）。

(2) 控制模式

P60 和 P61 引脚用作串行接口数据 I/O 和时钟 I/O 的引脚。

(a) SDA0

这是串行接口 IIC0 的串行数据 I/O 引脚。

(b) SCL0

这是串行接口 IIC0 的串行时钟 I/O 引脚。

2.2.6 P120 至 P122（端口 12）

P120 至 P122 为 3 位 I/O 端口。这些引脚用于外部中断请求输入，外部低压检测的电平输入，并且用于连接主系统时钟的振荡器以及主系统时钟的外部时钟输入。以下操作模式可以以 1 位为单位进行指定。

(1) 端口模式

P120 至 P122 用作 3 位 I/O 端口。P120 至 P122 可以通过使用端口模式寄存器 12（PM12）来设置为输入或输出端口。只有 P120 可以通过上拉电阻选择寄存器 12（PU12）来指定使用片上上拉电阻。

(2) 控制模式

P120 至 P122 引脚用于外部中断请求输入，外部低压检测的电平输入，并且用于连接主系统时钟的振荡器以及主系统时钟的外部时钟输入。

(a) INTPO

该引脚可作为外部中断请求输入（INTPO），可指定有效沿（上升沿，下降沿，或双边沿）。

(b) EXLVI

这是外部低压检测的电平输入引脚。

(c) X1, X2

这些可用于连接主系统时钟振荡器的引脚。

(d) EXCLK

这是主系统时钟的外部时钟输入引脚。

2.2.7 AV_{REF}

A/D 转换器参考电压输入引脚和 P20 至 P23 引脚以及 A/D 转换器的电源未使用 A/D 转换器时直接将该引脚连接在 V_{DD}[#]上。

注 当端口 2 被用作数字端口时，将 AV_{REF} 引脚的电平设置为 V_{DD} 引脚的电平。

2.2.8 AV_{SS}

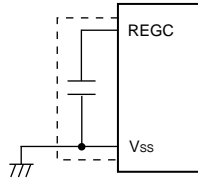
这是 A/D 转换器地电平引脚。即使没有使用 A/D 转换器时，通常也将该引脚的电平与 V_{SS} 引脚的电平保持一致。

2.2.9 $\overline{\text{RESET}}$

该引脚为有效低电平系统复位输入引脚。

2.2.10 REGC

该引脚用于为内部操作的稳压器输出（2.5V）连接一个稳定电容。通过电容将该引脚连接到 V_{SS} 上（0.47 至 1 μF ：推荐）。



注意事项 在上图虚线部分的配线长度应尽可能的短。

2.2.11 V_{DD}

这是除 P20 至 P23 外的引脚的电源引脚。

使用恒流驱动器部分时应在 4.5 至 5.5 V 的电压范围内对 V_{DD} 和 CV_{DD} 进行操作，未使用恒流驱动器部分时则应在 1.8 至 5.5 V 的电压范围内对 V_{DD} 进行操作。

2.2.12 V_{SS}

这是除 P20 至 P23 外的引脚的地电平引脚。

2.2.13 FLMD0

这是用于设置 flash 存储器编程模式的引脚。

在正常操作模式中将该 FLMD0 连接到 V_{SS} 上。

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 引脚功能描述（恒流驱动器部分）

2.3.1 EN（内部引脚）

使用下拉的操作的使能信号引脚（200 k Ω （TYP.））。

EN 引脚被连接在 CPU 部分的 P01/TI010/TO00 引脚的封装之内。因此，控制外部驱动器 IC 时，应使用外部驱动器 IC 来连接该引脚。没有控制外部驱动器 IC 时则将该引脚保持为开路状态。

2.3.2 PWM0 至 PWM3（内部引脚）

使用下拉的调光控制的 PWM 脉冲输入引脚（200 k Ω （TYP.））。

PWM0, PWM1, PWM2, 以及 PWM3 引脚被分别连接在 CPU 部分的 P33/INTP4/TO51/TI51, P15/TOH0, P16/TOH1/INTP5, 以及 P17/TI50/TO50 引脚的封装之内。因此，控制外部驱动器 IC 时，应使用外部驱动器 IC 来连接这些引脚。没有控制外部驱动器 IC 时则将这些引脚保持为开路状态。

此外，没有用作 PWM 控制输入的引脚可以用作通用端口，定时器 I/O，或中断输入。

2.3.3 SH（内部引脚）

这是热保护报警的输出引脚。根据热保护电路的操作，从 SH 引脚输出高电平，恒流驱动器的操作停止。

SH 引脚被连接在 CPU 部分的 P30/INTP1 引脚的封装之内。因此，该引脚应保持开路状态。

2.3.4 DRV0 至 DRV3

这些是外部 MOSFET 的栅极驱动输出引脚。

2.3.5 MODE

使用下拉的降压模式或升压模式选择输入（200 k Ω （TYP.））。

当 MODE 引脚输入电平被设为低电平时，恒流驱动器将会被设为升压模式，当 MODE 引脚输入电平被设为高电平时，恒流驱动器则将被设为降压模式。

2.3.6 SENS0 至 SENS3

这些是降压模式中各个通道的外部电路中的电流检测引脚，以及升压模式中外部电路的过压保护电压监测引脚。

2.3.7 SENS0 至 SENS3

这些是升压模式中各个通道的外部电路中的电流检测引脚。

2.3.8 CCA0 至 CCA3, CCB0 至 CCB3

这些是相位校正的电容连接引脚。

2.3.9 CT

这是三角波振荡器的电容连接引脚。

2.3.10 RT

这是三角波振荡器的电阻连接引脚。

2.3.11 CV_{DD0} 至 CV_{DD2}

这些是恒流驱动器的电源引脚。使用恒流驱动器部分时将会在 4.5 至 5.5 V 的电压范围内对 V_{DD} 和 CV_{DD} 进行操作。

2.3.12 V_{IN}

这是高电压电源引脚。

2.3.13 AGND

这是模拟地电平引脚。

2.3.14 PGND0 至 PGND3, PGND11

这些是地电平引脚。

2.3.15 IC0

这是内部连接引脚。直接将该引脚连接至 V_{SS} 。

2.4 I/O引脚电路以及不使用引脚的推荐连接

表 2-1 和 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 (CPU 部分)

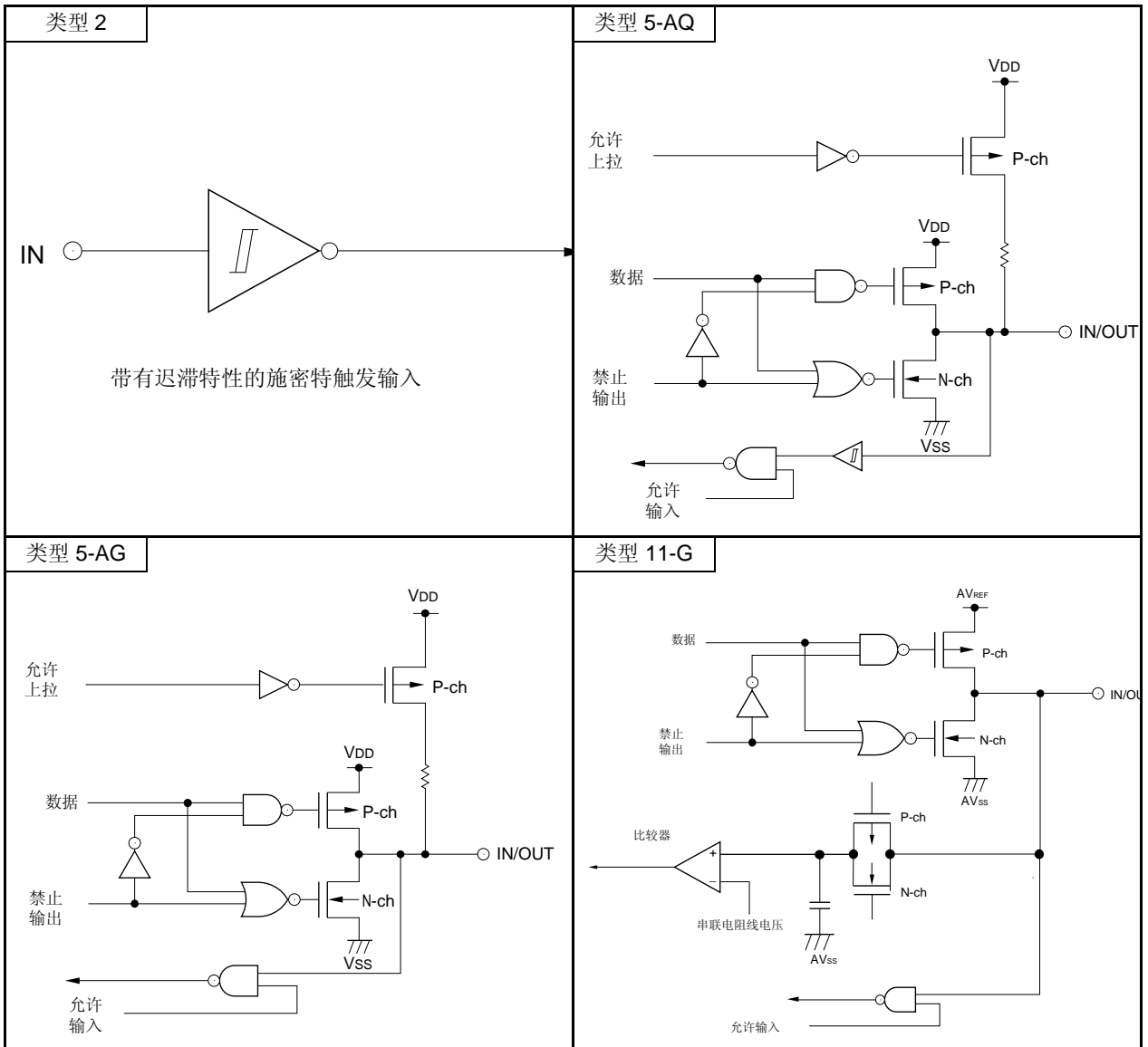
引脚名称	I/O 电路类型	I/O	不使用的引脚的推荐连接
P00/TI000	5-AQ	I/O	输入： 通过一个电阻单独连接到 V _{DD} 或 V _{SS} 。 输出： 保持开路。
P01/TI010/TO00 (/EN)			保持开路。
P10/SCK10/TxD0			输入： 通过一个电阻单独连接到 V _{DD} 或 V _{SS} 。 输出： 保持开路。
P11/SI10/RxD0			
P12/SO10	5-AG		
P13/TxD6			
P14/RxD6	5-AQ		
P15/TOH0 (/PWM1)	5-AG		保持开路。
P16/TOH1/INTP5 (/PWM2)	5-AQ		
P17/TI50/TO50 (/PWM3)			
ANI0/P20 至 ANI3/P23 ^{注1}	11-G	输入	<数字输入设置和模拟输入设置> 通过一个电阻单独连接到 AV _{REF} 或 AV _{SS} 。 <数字输出设置> 保持开路。
P30/INTP1 (/SH)	5-AQ	I/O	保持开路。
P31/INTP2			输入： 通过一个电阻单独连接到 V _{DD} 或 V _{SS} 。 输出： 保持开路。
P32/INTP3			
P33/TI51/TO51/INTP4 (/PWM0)			保持开路。
P60/SCL0	13-AI		输入： 连接到 V _{SS} 。 输出： 在将端口的输出锁存器清除为 0 后，将该引脚在低电平输出时保持为开路。
P61/SDA0			
P120/INTP0/EXLVI	5-AQ		输入： 通过一个电阻单独连接到 V _{DD} 或 V _{SS} 。 输出： 保持开路。
P121/X1 ^{注2}	37		
P122/X2/EXCLK ^{注2}			
AV _{REF}	-	-	直接连接到 V _{DD} ^{注3} 。
AV _{SS}	-	-	直接连接到 V _{SS} 。
FLMD0	38-A	-	连接到 V _{SS} ^{注4} 。
RESET	2	输入	通过一个电阻直接连接到 V _{DD} 上。

- 注
1. 复位释放后 P20/ANI0 至 P23/ANI3 引脚设置为模拟输入模式。
 2. 当这些引脚没有被使用时，在 I/O 端口模式中使用上述的推荐连接（参见图 6-2 时钟操作模式选择寄存器 (OSCCTL) 的格式）。
 3. 当端口 2 被用作数字端口时，将电平设置为 V_{DD} 引脚的电平。
 4. FLMD0 是用于将数据写入 flash 存储器中的引脚。若要重写在线 flash 存储器的数据，则通过一个电阻（10 kΩ：推荐）将该引脚连接至 V_{SS} 上。

表 2-2. I/O 引脚电路类型（恒流驱动器部分）

操作模式	引脚名称	I/O	不使用的引脚的推荐连接	
升压模式， 未使用通道 n 时 (n = 0 至 3)	PWMn	输入	保持开路，或通过一个电阻连接到 PGND 上。	
	SENSAn		连接到 VIN。	
	SENSBn		连接到 PGND。	
	DRVn	输出	保持开路。	
	CCAn			-
	CCBn			-
降压模式， 未使用通道 n 时 (n = 0 至 3)	PWMn	输入	保持开路，或通过一个电阻连接到 PGND 上。	
	SENSAn		连接到 VIN。	
	SENSBn		保持开路。	
	DRVn	输出	保持开路。	
	CCAn			-
	CCBn			-
-	IC0	-	直接连接到 VSS。	

图 2-1. I/O 引脚电路列表 (1/2)



第三章 带有可选降压或升压转换器的恒流驱动器

3.1 带有可选降压或升压转换器的恒流驱动器的功能

恒流驱动器的每一个通道都可以外接功率 MOSFET，搭配合适的外部组件，向负载提供 0.35 A，0.7 A，1 A，或 1.5 A 的恒流驱动。

根据 MODE 引脚中的输入电平（外部设定的）以及外部组件的配置，可以选择升压模式（低电平输入）或降压模式（高电平输入）作为恒流驱动器的工作模式。

上电后，通过将高电平输入到 EN 引脚，然后将高电平输入到 PWM0 至 PWM3 的各个引脚，每个通道的恒流驱动器开始工作。可以通过将 PWM 信号输入至 PWM0 至 PWM3 引脚实现恒流驱动器的间歇运行。

当电压 $CV_{DD} < V_{LUVLO}$ 时，恒流驱动器停止工作。

在恒流驱动器操作工作过程中，从外部电路中检测到过流或过压（仅在升压模式中）时，仅停止有问题的通道的工作。此外，当设备升温（到至少 150°C（TYP.））时，从 SH 引脚将输出一个高电平，且恒流驱动器停止工作。

- 可以选择升压模式或降压模式。
- 提供 4 通道（通道 0 至 3）控制电路，用于驱动恒流负载（0.35 A, 0.7 A, 1 A, 或 1.5 A）。
- 根据 PWM 输入（CPU 部分的 8 位定时器的 PWM 输出）可以实现开关控制以及电流值控制。
- 提供热保护功能。
- 提供过流保护功能。
- 提供过压保护功能（仅在升压模式中）。
- 提供欠压锁定功能（UVLO）。

注意事项 1. EN 引脚被连接在 CPU 的 P01/TI010/TO00 引脚。可以通过操作 CPU 的 P01 来控制恒流驱动器。

要控制外部驱动器 IC，应使用外部驱动器 IC 来连接该引脚。没有控制外部驱动器 IC 时，则将该引脚保持为开路状态。

2. SH 引脚连接在 CPU 的 P30/INTP1 引脚。因此，可以使用 CPU 的中断功能来检测 SH 引脚的输出。该引脚保持为开路状态。

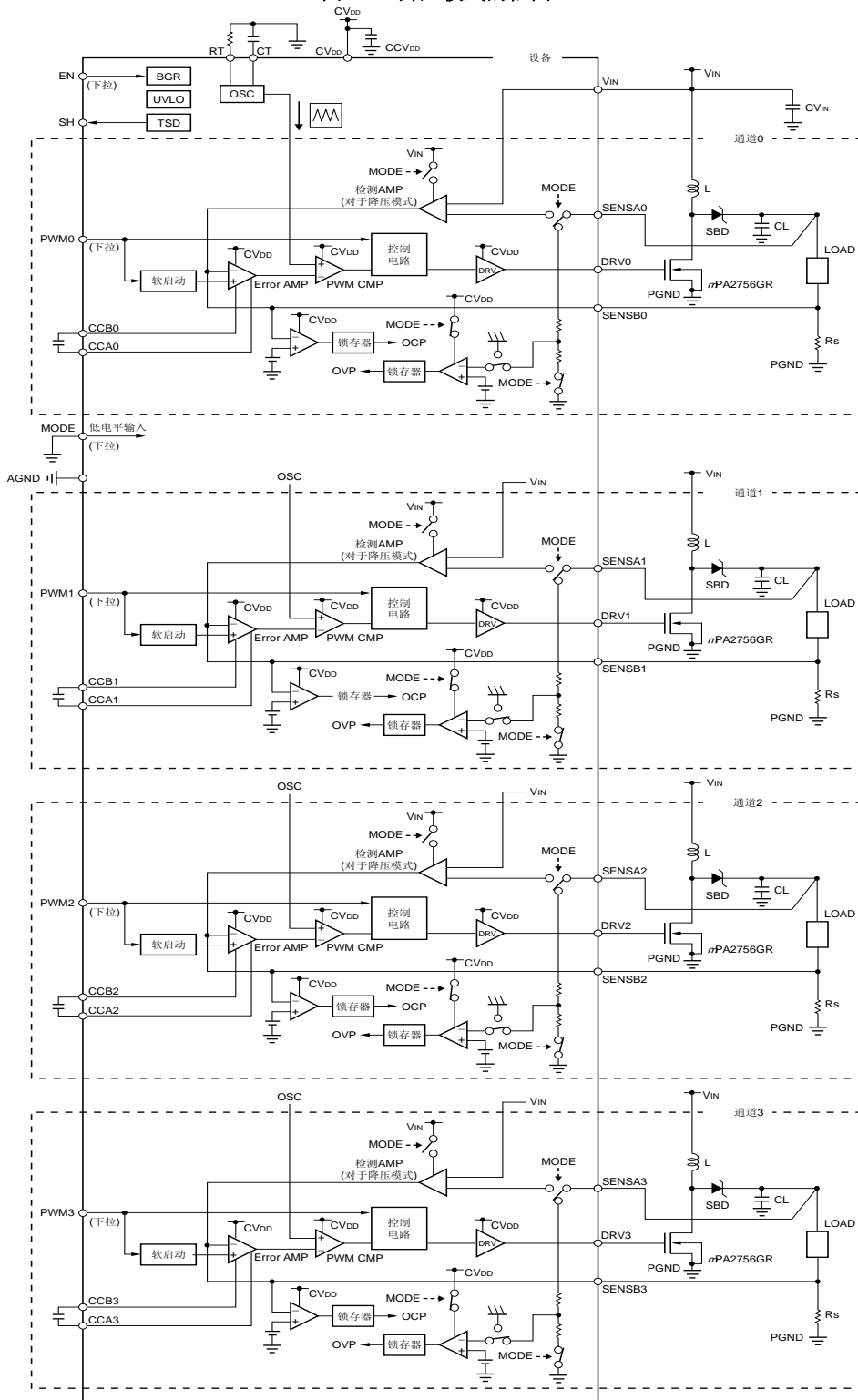
3. PWM0, PWM1, PWM2, 以及 PWM3 引脚分别连接在 CPU 的 P33/INTP4/TO51/TI51, P15/TOH0, P16/TOH1/INTP5, 以及 P17/TI50/TO50 引脚。因此，CPU 的 8 位定时器输出的 PWM 可以输入到恒流驱动器部分中。

要控制外部驱动 IC，应使用外部驱动 IC 来连接这些引脚。没有外部驱动 IC 时，则将这些引脚保持为开路状态。此外，没有用作 PWM 控制输入的引脚可以用作通用端口，定时器 I/O，或中断输入。

3.2 升压模式

下图为升压模式的框图。

图 3-1. 升压模式的框图

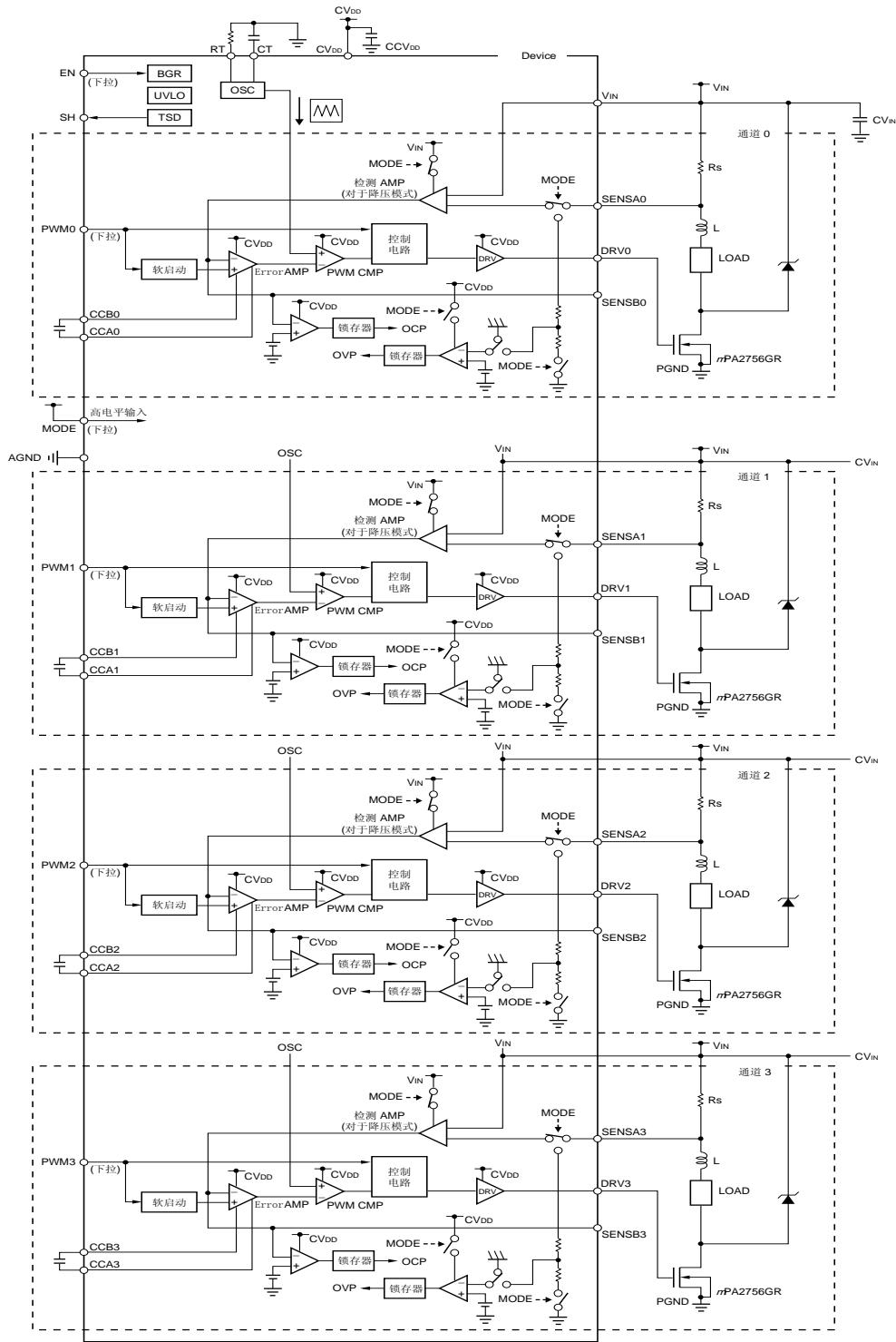


- 注意事项 1. 上述所示电路框图仅供参考，不能保证实际规范。
 2. 设置升压模式时，直接将 MODE 引脚连接至 GND。

3.3 降压模式

下图为降压模式的框图。

图 3-2. 降压模式的框图



- 注意事项 1. 上述模式仅供参考，不能保证实际规范。
 2. 设置降压模式时，直接将 MODE 引脚连接至 CVDD。

备注 1. 负载电流 (I_{LOAD}) 按如下公式确定，这是因为在升压或降压模式中各个通道的检测电压为 0.115 V (TBD)。

$$\text{负载电流 (I}_{\text{LOAD}}) = \frac{0.115\text{V(TBD)}}{\text{电流检测电阻 (R}_s)}$$

建议电流检测电阻 (R_s) 的允许误差范围不超过±1%。

设置示例如下所示。

参考值			
R _s (Ω)	I _{LOAD} (A)	V _{LOAD} (V)	P _{RS} (W)
0.33	0.35	0.115	0.04
0.18	0.64	0.115	0.073
0.12	0.96	0.115	0.11
0.082	1.4	0.115	0.16

2. 根据负载电流 (I_{LOAD}) 的大小，推荐的外部 N-ch MOSFET 产品如下：

负载电流 (I_{LOAD}) ≤ 0.75 A: μ PA2756GR (NEC Electronics 产品)

0.75 A < 负载电流 (I_{LOAD}) ≤ 1.5 A: 2SK2414-Z (NEC Electronics 产品)

3.4 工作模式设置步骤

通过以下步骤在升压模式或降压模式中启动操作。

<1> 为恒流驱动器提供供电电压 (CV_{DD} 引脚 0 V → CV_{DD})。

<2> 将 MODE 引脚中的输入电平固定为高电平或低电平。

- 低电平：升压模式
- 高电平：降压模式

<3> 为恒流驱动器提供高压 (V_{IN} 引脚：0 V → V_{IN})。

<4> 将高电平输入到 EN 引脚中。启动三角波的振荡。

<5> PWM 等待时间 (100 μs (最小)) 过后向 PWM0 至 PWM3 的各个引脚中输入了一个高电平。各个通道的恒流驱动器的操作将会开始。

<6> 将 PWM 信号输入到 PWM0 至 PWM3 引脚中。恒流驱动器将会开始间歇操作。

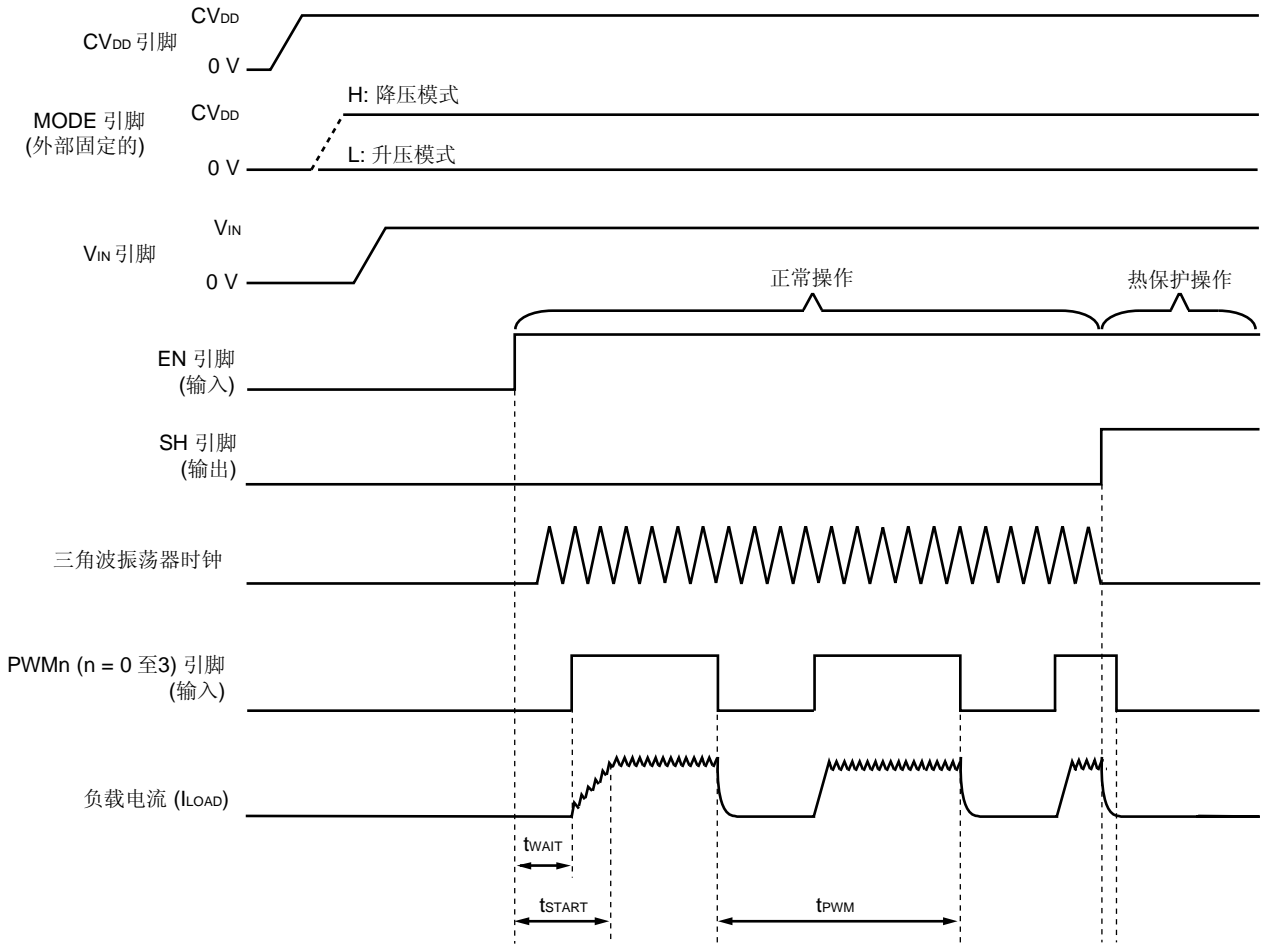
通道 0 至 3 的平均负载电流可以经过相应的 PWM 输入引脚中的 PWM 信号来进行控制。CPU 部分的 8 位定时器中的 PWM 输出的占空比可以实现 256 阶控制，因为它处于 1/256 至 256/256 范围之内。

在恒流驱动器操作过程中，当在外部电路中检测到过流或过压 (仅在升压模式中) 时，仅停止有问题的通道的操作。此时，SH 引脚没有输出。

此外，当恒流驱动器升温 (到至少 150°C (TYP.)) 时，从 SH 引脚中输出一个高电平，且恒流驱动器停止工作。

若要停止恒流驱动器的工作并关闭电源，则应在 V_{IN} 引脚被设为 0V 后将 CV_{DD} 引脚设为 0V。

图 3-3. 设定恒流驱动器操作模式的时序



- 注意事项 1.** 当恒流驱动器升温（到至少 150°C（TYP.））时，将开始启用热保护功能。启用热保护功能后将无法保证设备的可靠性和电气特性，这是因为该温度超过了最大绝对额定值。
- 2.** 热保护功能在 EN 引脚抬高后的 400 μs 后开始启用。
- 3.** 设置 PWM 输入以使 PWM 输出在上述 PWM 等待时间（t_{WAIT}）过后从 CPU 发送。

备注

t _{WAIT} :	PWM 等待时间（EN 引脚抬高→ PWMn 引脚抬高）
t _{START} :	总的升温时间（EN 引脚抬高→ 90% 正常负载电流）
t _{PWM} :	PWM 周期

第四章 CPU架构

4.1 存储空间

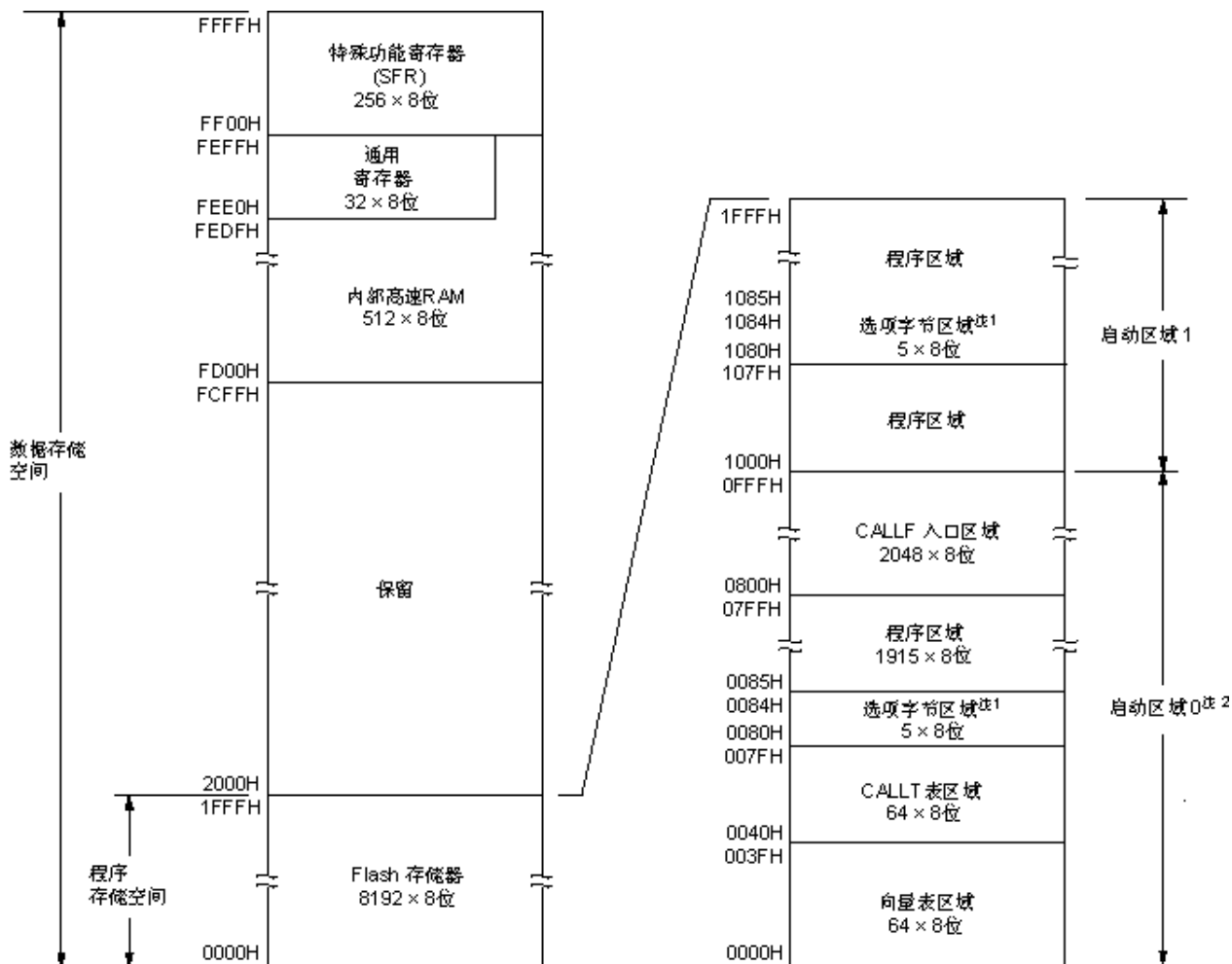
μ PD78F8024 和 μ PD78F8025 可以访问 64KB 的存储空间。图 4-1 和图 4-2 显示了存储器映射图。

注意事项 无论内部存储器的容量是多少，内部存储器容量切换寄存器（IMS）的初始值均为 **CFH**。因此，为如下所示的每一种产品设置相应的值。

表 4-1. 内部存储器容量切换寄存器（IMS）的设定值

产品	IMS	ROM 容量	内部高速 RAM 容量
μ PD78F8024	42H	8 KB	512 字节
μ PD78F8025	C8H	32 KB	1K 字节

图 4-1. 存储器映射图 (μPD78F8024)



- 注
1. 未使用引导交换时：为 0080H 至 0084H 设置选项字节。
使用引导交换时：为 0080H 至 0084H 以及 1080H 至 1084H 设置选项字节。
 2. 通过安全设置可以禁止对引导簇 0 进行写操作（参见 22.7 安全设置）。

备注 flash 存储器被分为多个块（每块 = 1 KB）。关于地址值以及块编号的详细信息，请参见表 4-2 Flash 存储器中地址值与块编号间的对应关系

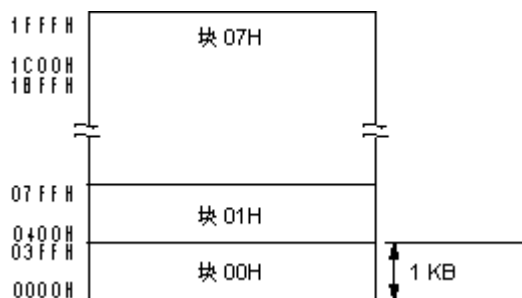
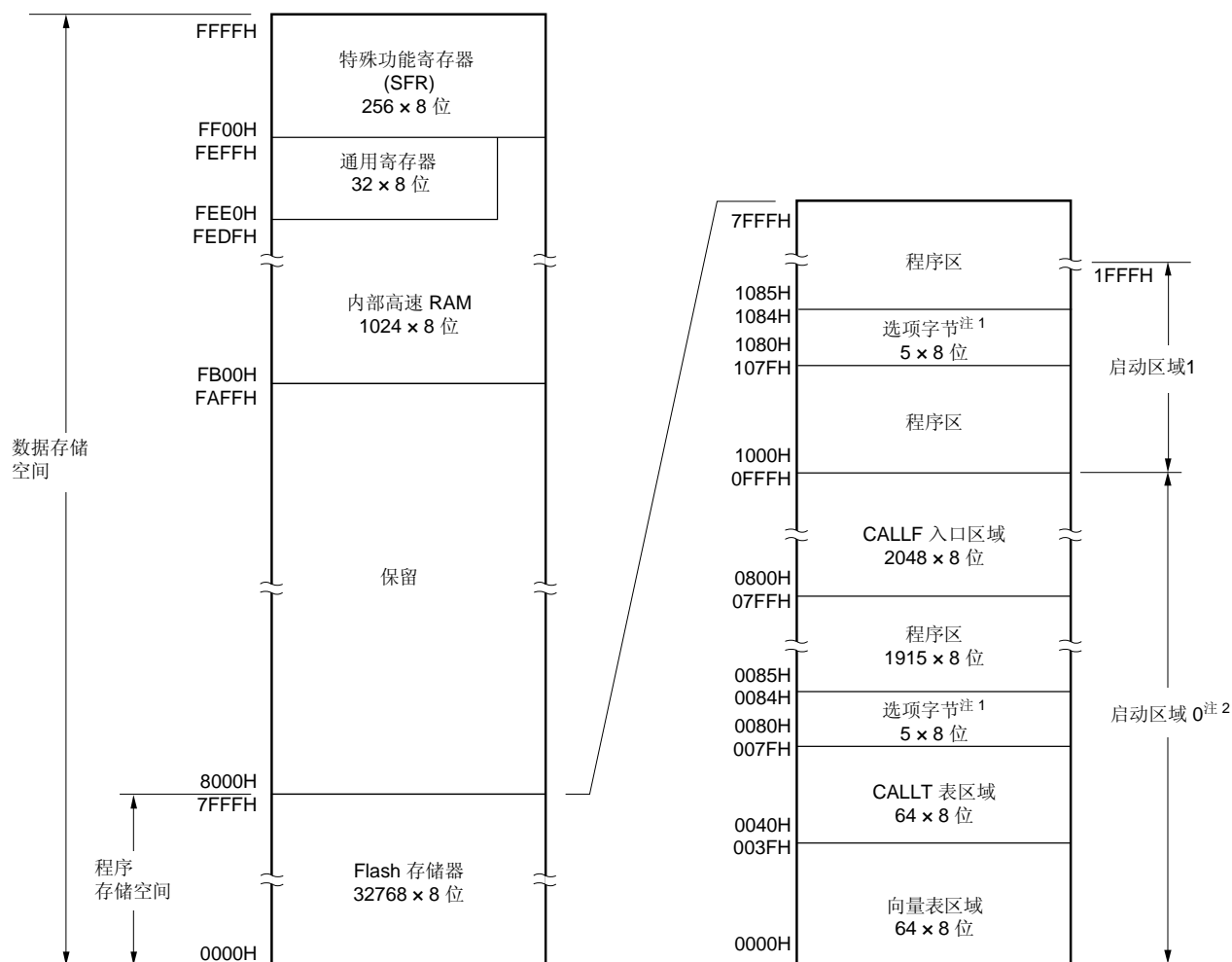


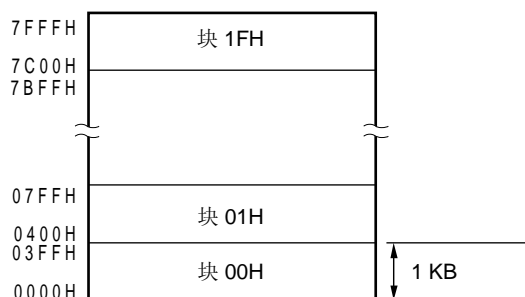
图 4-2. 存储器映射图 (μ PD78F8025)



Notes 1. 未使用引导交换时：为 0080H 至 0084H 设置选项字节。
使用引导交换时：为 0080H 至 0084H 以及 1080H 至 1084H 设置选项字节。

2. 通过安全设置可以禁止对引导簇 0 进行写操作（参见 22.7 安全设置）。

备注 flash 存储器被分为多个块（每块 = 1 KB）。关于地址值以及块编号的详细信息，请参见表 4-2 Flash 存储器中地址值与块编号间的对应关系



flash 存储器中地址值与块编号的对应关系如下所示。

表 4-2. Flash 存储器中地址值与块编号的对应关系

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

备注 μ PD78F8024: 块编号 00H ~ 07H

μ PD78F8025: 块编号 00H ~ 1FH

4.1.1 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，程序计数器（PC）对其寻址。

μ PD78F8024 和 μ PD78F8025 中包含的内置 ROM（flash 存储器）如下所示。

表 4-3. 内置 ROM 容量

产品编号	内部 ROM	
	结构	容量
μ PD78F8024	Flash 存储器	8192 × 8 位 (0000H ~ 1FFFH)
μ PD78F8025		32768 × 8 位 (0000H ~ 7FFFH)

<R>

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 被用作向量表区域。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 位保存在偶地址中，而 16 位地址中的高 8 位则保存在奇地址中。

表 4-4. 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	0018H	INTCSI10/INTST0
0004H	INTLVI	001AH	INTTMH1
0006H	INTP0	001CH	INTTMH0
0008H	INTP1	001EH	INTTM50
000AH	INTP2	0020H	INTTM000
000CH	INTP3	0022H	INTTM010
000EH	INTP4	0024H	INTAD
0010H	INTP5	0026H	INTSR0
0012H	INTSRE6	002AH	INTTM51
0014H	INTSR6	0034H	INTIIC0
0016H	INTST6	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 至 0084H 以及 1080H 至 1084H 的 5 字节区域可以用作选项字节区域。未使用引导交换时对 0080H 至 0084H 处的选项字节进行设置, 使用引导交换时则对 0080H 至 0084H 以及 1080H 至 1084H 处的选项字节进行设置。详细信息, 请参照第 21 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) 执行直接子程序调用。

4.1.2 内部数据存储空间

μ PD78F8024 和 μ PD78F8025 包含的内部 RAM 如下所示。

表 4-5. 内部高速 RAM 容量

产品编号	内部高速 RAM
μ PD78F8024	512 × 8 位 (FD00H ~ FEFFH)
μ PD78F8025	1024 × 8 位 (FB00H ~ FEFFH)

<R>

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 也可以用作堆栈存储器。

4.1.3 特殊功能寄存器 (SFR) 区域

片上外围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 4-6 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。

4.1.4 数据存储器寻址

寻址为指定下一个执行指令的地址或与指令执行相关的寄存器或存储器地址的方法。

基于可操作性及其他一些考虑， μ PD78F8024 和 μ PD78F8025 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 4-3 图 4-4 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 4.4 操作数地址寻址。

图 4-3. 数据存储与寻址间的对应关系 (μPD78F8024)

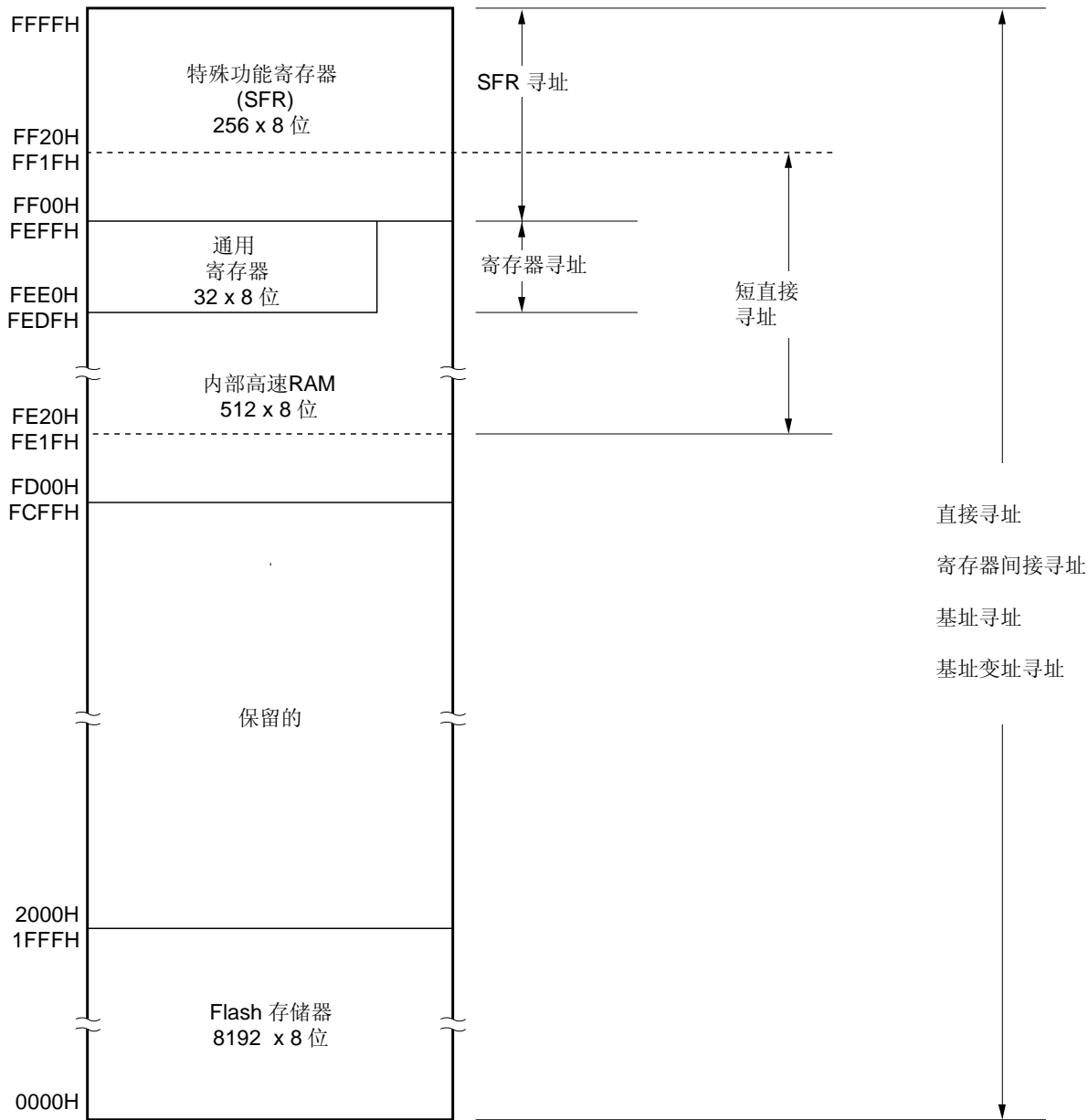
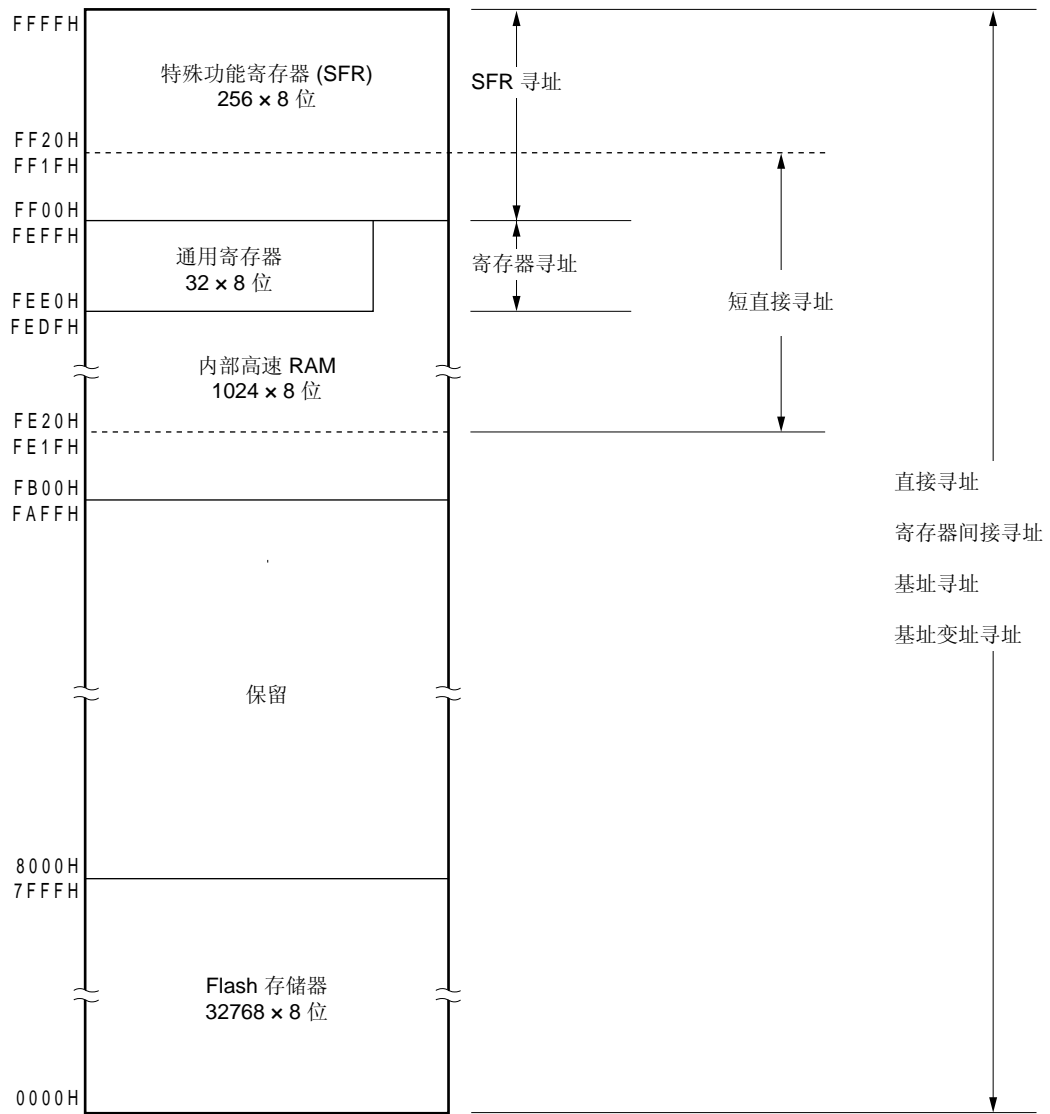


图 4-4. 数据存储与寻址间的对应关系 (μ PD78F8025)



4.2 处理器寄存器

μ PD78F8024 和 μ PD78F8025 中包含有以下处理器寄存器。

4.2.1 控制寄存器

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

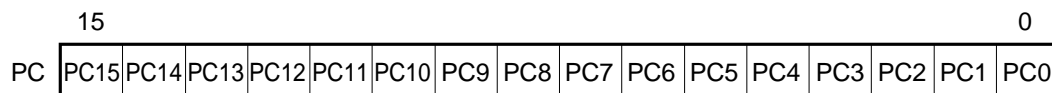
(1) 程序计数器（PC）

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 处的复位向量表的值赋值到程序计数器中。

图 4-5. 程序计数器的格式



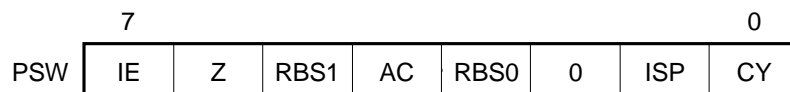
(2) 程序状态字（PSW）

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设置/复位。

响应向量中断时或在 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 4-6. 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 标志将中断状态设置为中断禁止（DI）状态，且禁止所有可屏蔽中断请求。

当 IE 为 1 时，IE 标志中断状态设置为中断允许（EI）状态，且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中断请求响应。

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 R_n 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 位上有借位，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, PR1H) 所指定的低电平向量中断请求 (参见 16.3 (3) 优先级指令标志寄存器 (PR0L, PR0H, PR1L, PR1H))。实际请求的响应是由中断允许标志 (IE) 来控制的。

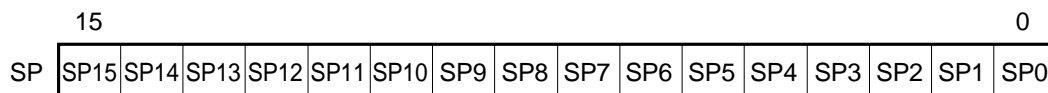
(f) 进位标志 (CY)

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 4-7. 堆栈指针的格式

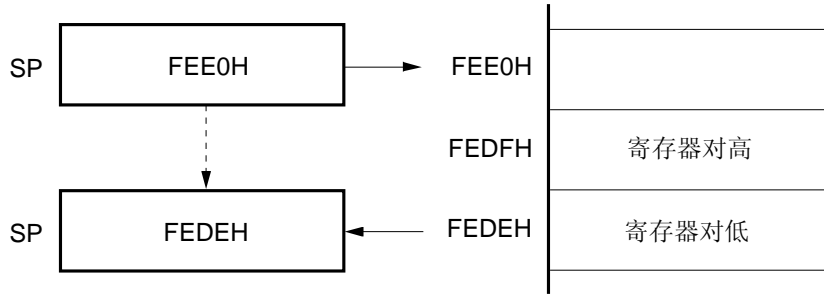


向堆栈存储器进行写 (保存) 操作前 SP 累减，从堆栈存储器进行读取 (恢复) 操作后 SP 累加。各个堆栈保存/恢复数据的操作如图 4-8 和 4-9 所示。

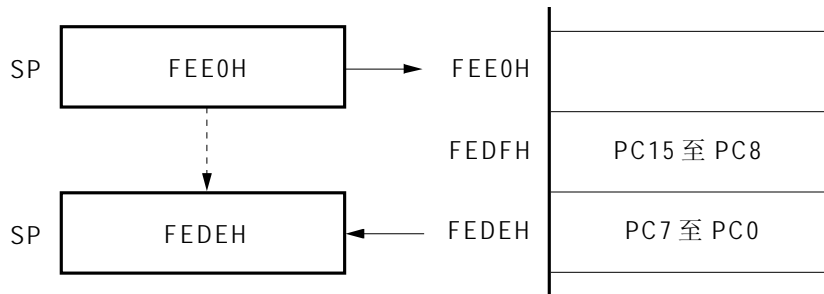
注意事项 由于复位信号生成会使用 SP 内容变为未定义，因此在使用堆栈前请务必初始化 SP。

图 4-8. 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)

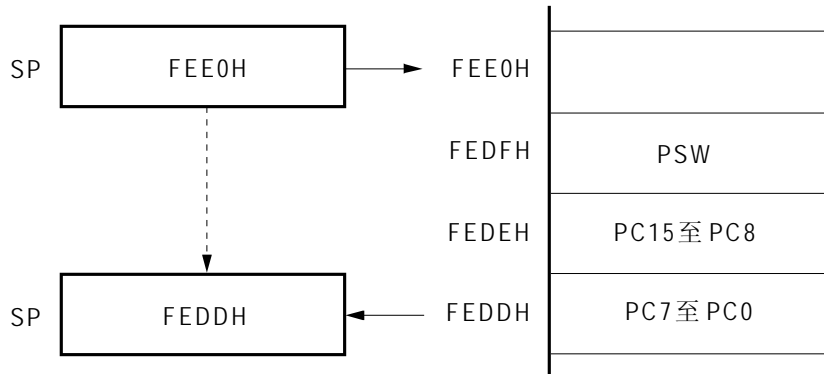
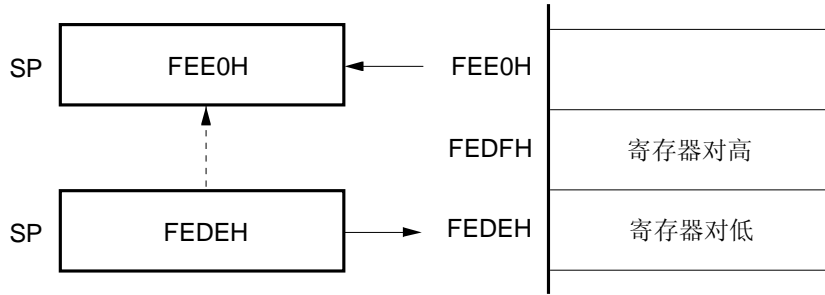
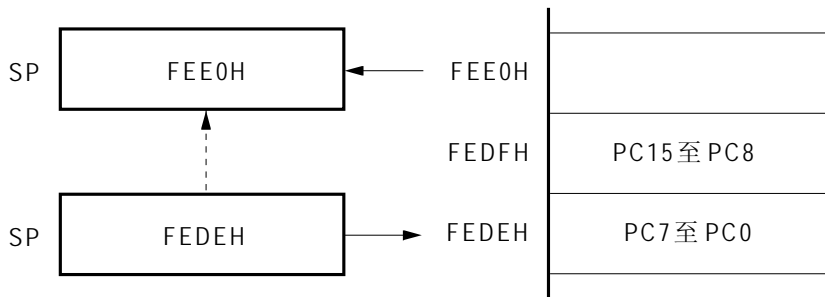


图 4-9. 从堆栈存储器中恢复的数据

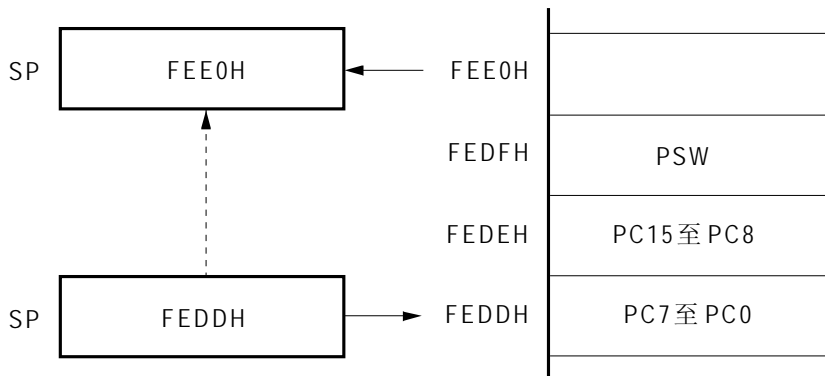
(a) POP rp 指令 (当 SP = FEDEH 时)



(b) RET 指令 (当 SP = FEDEH 时)



(c) RETI, RETB 指令 (当 SP = FEDDH 时)



4.2.2 通用寄存器

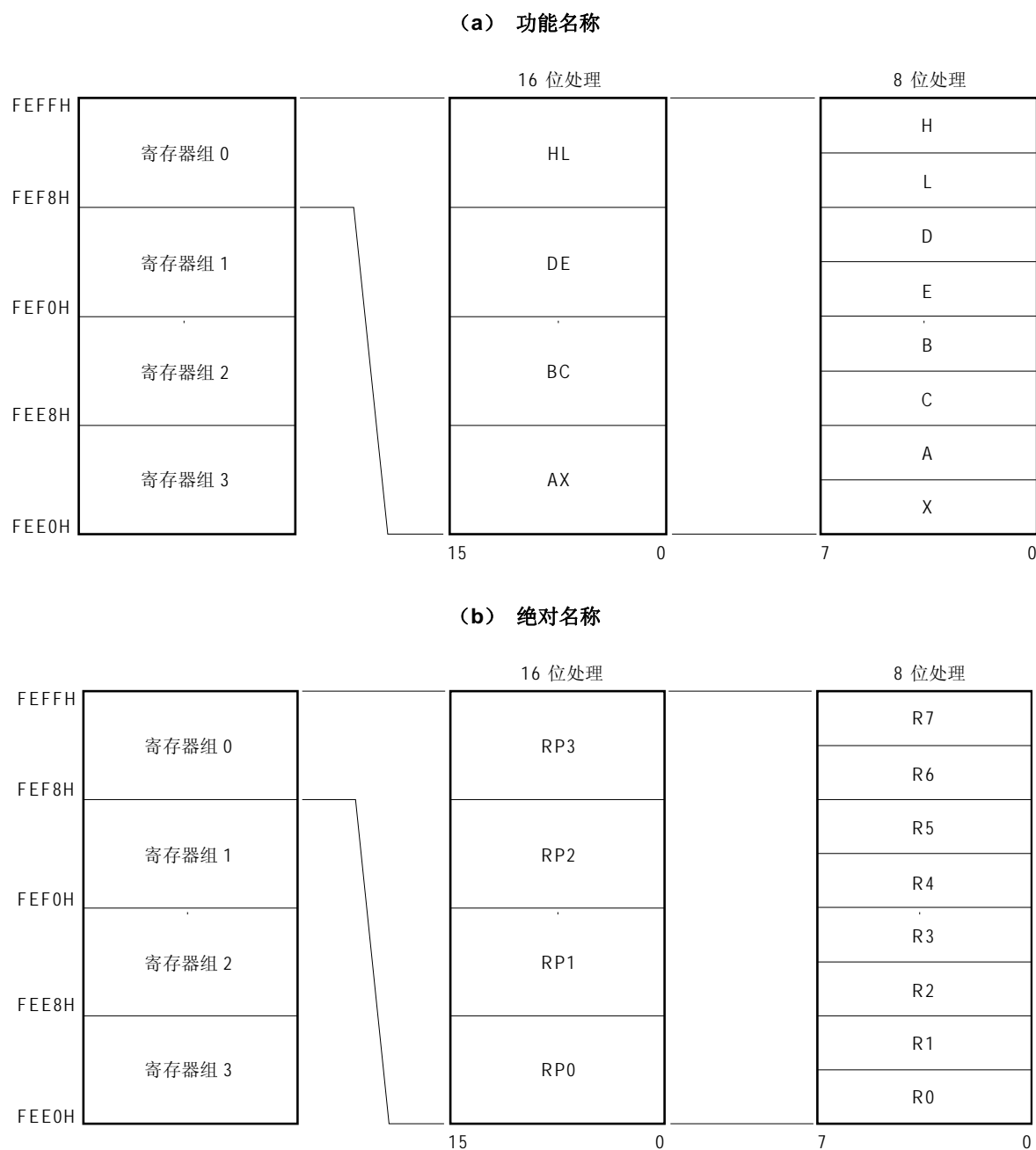
通用寄存器映射在数据存储器的特定地址（FEE0H 至 FEF FH）上。通用寄存器由 4 个寄存器组成，每个寄存器组由 8 个 8 位寄存器（X，A，C，B，E，D，L，以及 H）组成。

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 个寄存器组的结构，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切换使得程序更有效率。

图 4-10. 通用寄存器的结构



4.2.3 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特定功能。

SFR 可被分配到 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.bit**) 描述为汇编程序保留的符号。
该操作也可以通过地址来指定。
- 8 位操作
8 位操作指令操作数 (**sfr**)，描述为汇编程序保留的符号。
该操作也可以通过地址来指定。
- 16 位操作
16 位操作指令操作数 (**sfrp**)，描述为汇编程序保留的符号。
指定地址时，应为偶地址。

表 4-6 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` 声明来定义为 `sfr` 变量。使用 RA78K0，ID78K0-QB，SM+ for 78K0 和 SM+ for 78K0/KX2 时，符号可以写作指令操作数。
- R/W
表示相应的特殊功能寄存器是可读还是可写。
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)。“-”表示不能进行对应位操作。
- 复位后
表示复位信号生成时各寄存器的状态。

表 4-6. 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
FF06H	端口寄存器 6	P6	R/W	√	√	–	00H
FF08H	10-位 A/D 转换结果寄存器	ADCR	R	–	–	√	0000H
FF09H	8 位 A/D 转换结果寄存器	ADCRH	R	–	√	–	00H
FF0AH	接收缓冲寄存器 6	RXB6	R	–	√	–	FFH
FF0BH	发送缓冲寄存器 6	TXB6	R/W	–	√	–	FFH
FF0CH	端口寄存器 12	P12	R/W	√	√	–	00H
FF0FH	串行 I/O 移位寄存器 10	SIO10	R	–	√	–	00H
FF10H	16 位定时器计数器 00	TM00	R	–	–	√	0000H
FF11H							
FF12H	16 位定时器捕捉/比较寄存器 000	CR000	R/W	–	–	√	0000H
FF13H							
FF14H	16 位定时器捕捉/比较寄存器 010	CR010	R/W	–	–	√	0000H
FF15H							
FF16H	8 位定时器计数器 50	TM50	R	–	√	–	00H
FF17H	8 位定时器比较寄存器 50	CR50	R/W	–	√	–	00H
FF18H	8 位定时器 H 比较寄存器 00	CMP00	R/W	–	√	–	00H
FF19H	8 位定时器 H 比较寄存器 10	CMP10	R/W	–	√	–	00H
FF1AH	8 位定时器 H 比较寄存器 01	CMP01	R/W	–	√	–	00H
FF1BH	8 位定时器 H 比较寄存器 11	CMP11	R/W	–	√	–	00H
FF1FH	8 位定时器计数器 51	TM51	R	–	√	–	00H
FF20H	端口模式寄存器 0	PM0	R/W	√	√	–	FFH
FF21H	端口模式寄存器 1	PM1	R/W	√	√	–	FFH
FF22H	端口模式寄存器 2	PM2	R/W	√	√	–	FFH
FF23H	端口模式寄存器 3	PM3	R/W	√	√	–	FFH
FF26H	端口模式寄存器 6	PM6	R/W	√	√	–	FFH
FF28H	A/D 转换器模式寄存器	ADM	R/W	√	√	–	00H
FF29H	模拟输入通道指定寄存器	ADS	R/W	√	√	–	00H
FF2CH	端口模式寄存器 12	PM12	R/W	√	√	–	FFH
FF2FH	A/D 端口配置寄存器	ADPC	R/W	√	√	–	00H
FF30H	上拉电阻选择寄存器 0	PU0	R/W	√	√	–	00H
FF31H	上拉电阻选择寄存器 1	PU1	R/W	√	√	–	00H
FF33H	上拉电阻选择寄存器 3	PU3	R/W	√	√	–	00H
FF3CH	上拉电阻选择寄存器 12	PU12	R/W	√	√	–	00H
FF41H	8 位定时器比较寄存器 51	CR51	R/W	–	√	–	00H
FF43H	8 位定时器模式控制寄存器 51	TMC51	R/W	√	√	–	00H
FF48H	外部中断上升沿允许寄存器	EGP	R/W	√	√	–	00H
FF49H	外部中断下降沿允许寄存器	EGN	R/W	√	√	–	00H

表 4-6. 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	
FF4FH	输入转换控制寄存器	ISC	R/W	√	√	–	00H
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 比较寄存器 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 比较寄存器 1	TMHMD1	R/W	√	√	–	00H
FF6DH	8 位定时器 H 载频控制寄存器 1	TMCYC1	R/W	√	√	–	00H
FF70H	异步串行接口操作模式寄存器 0	ASIM0	R/W	√	√	–	01H
FF71H	波特率发生器控制寄存器 0	BRGC0	R/W	–	√	–	1FH
FF72H	接收缓冲寄存器 0	RXB0	R	–	√	–	FFH
FF73H	异步串行接口接收错误状态寄存器 0	ASIS0	R	–	√	–	00H
FF74H	发送移位寄存器 0	TXS0	W	–	√	–	FFH
FF80H	串行操作模式寄存器 10	CSIM10	R/W	√	√	–	00H
FF81H	串行时钟选择寄存器 10	CSIC10	R/W	√	√	–	00H
FF84H	发送缓冲寄存器 10	SOTB10	R/W	–	√	–	00H
FF8CH	定时器时钟选择寄存器 51	TCL51	R/W	√	√	–	00H
FF99H	看门狗定时器允许寄存器	WDTE	R/W	–	√	–	1AH/9AH ^{注 1}
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

- 注 1. WDTE 的复位值由选项字节的设置来决定。
 2. 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的振荡精度稳定后会变为 80H。

表 4-6. 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	
FFA5H	IIC 移位寄存器 0	IIC0		R/W	–	√	–	00H
FFA6H	IIC 控制寄存器 0	IICC0		R/W	√	√	–	00H
FFA7H	从设备地址寄存器 0	SVA0		R/W	–	√	–	00H
FFA8H	IIC 时钟选择寄存器 0	IICCL0		R/W	√	√	–	00H
FFA9H	IIC 功能扩展寄存器 0	IICX0		R/W	√	√	–	00H
FFAAH	IIC 状态寄存器 0	IICS0		R	√	√	–	00H
FFABH	IIC 标志寄存器 0	IICF0		R/W	√	√	–	00H
FFACH	复位控制标志寄存器	RESF		R	–	√	–	00H ^{注 1}
FFBAH	16 位定时器模式控制寄存器 00	TMC00		R/W	√	√	–	00H
FFBBH	预分频模式寄存器 00	PRM00		R/W	√	√	–	00H
FFBCH	捕捉/比较控制寄存器 00	CRC00		R/W	√	√	–	00H
FFBDH	16 位定时器输出控制寄存器 00	TOC00		R/W	√	√	–	00H
FFBEH	低电压检测寄存器	LVIM		R/W	√	√	–	00H ^{注 2}
FFBFH	低电压检测电平选择寄存器	LVIS		R/W	√	√	–	00H ^{注 2}
FFE0H	中断请求标志寄存器 0L	IF0	IF0L	R/W	√	√	√	00H
FFE1H	中断请求标志寄存器 0H		IF0H	R/W	√	√		00H
FFE2H	中断请求标志寄存器 1L	IF1	IF1L	R/W	√	√	√	00H
FFE3H	中断请求标志寄存器 1H		IF1H	R/W	√	√		00H
FFE4H	中断屏蔽标志寄存器 0L	MK0	MK0L	R/W	√	√	√	FFH
FFE5H	中断屏蔽标志寄存器 0H		MK0H	R/W	√	√		FFH
FFE6H	中断屏蔽标志寄存器 1L	MK1	MK1L	R/W	√	√	√	FFH
FFE7H	中断屏蔽标志寄存器 1H		MK1H	R/W	√	√		FFH
FFE8H	优先级指定标志寄存器 0L	PR0	PR0L	R/W	√	√	√	FFH
FFE9H	优先级指定标志寄存器 0H		PR0H	R/W	√	√		FFH
FFEAH	优先级指定标志寄存器 1L	PR1	PR1L	R/W	√	√	√	FFH
FFEBH	优先级指定标志寄存器 1H		PR1H	R/W	√	√		FFH
FFF0H	内部存储器容量切换寄存器 ^{注 3}	IMS		R/W	–	√	–	CFH
FFFBH	处理器时钟控制寄存器	PCC		R/W	√	√	–	01H

- 注
1. RESF 的复位值根据复位源的不同而变化。
 2. LVIM 和 LVIS 的复位值根据复位源的不同而变化。
 3. 无论内部存储器的容量是多少，内部存储器容量切换寄存器 (IMS) 的初始值均为 CFH。因此，为如下所示的每一个产品设置相应的值。

产品	IMS	ROM 容量	内部高速 RAM 容量
μPD78F8024	42H	8 KB	512 个字节
μPD78F8025	C8H	32 KB	1KB

4.3 指令地址寻址

指令地址由程序计数器（PC）的内容决定，根据执行指令时获取的下一条指令字节数，程序计数器（PC）的内容自动累加（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）**）。

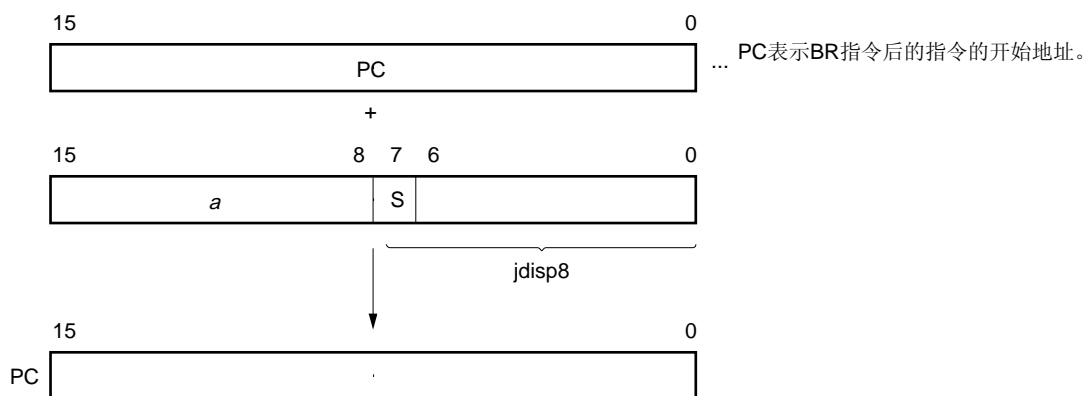
4.3.1 相对寻址

[功能]

将指令代码的 8 位立即数据（偏移量：jdisp8）与下一条指令中的开始地址相加，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。偏移量是带符号位的二进制补码（-128 至 +127），而第 7 位是符号位。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，所有位 a 为 0。
当 $S = 1$ 时，所有位 a 为 1。

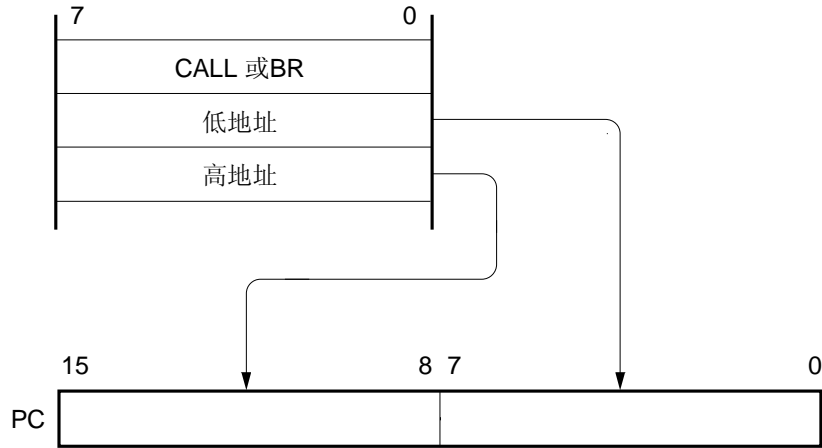
4.3.2 立即寻址

[功能]

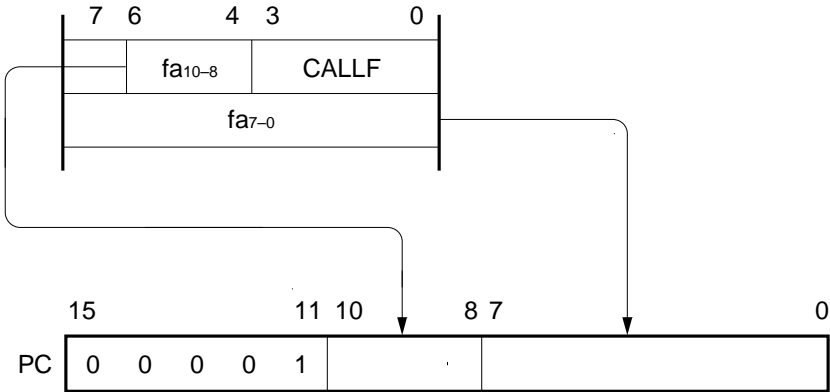
指令字中的立即数据赋给程序计数器（PC），并转向该地址。
 该功能可以在执行 CALL !addr16 或 BR !addr16 或 CALLF !addr11 指令时执行。
 CALL !addr16 和 BR !addr16 指令可以跳转到整个存储器空间中。
 CALLF !addr11 则被转移到 0800H 至 0FFFH 的区域中。

[图示]

使用 CALL !addr16 和 BR !addr16 指令时



使用 CALLF !addr11 指令时



4.3.3 表间接寻址

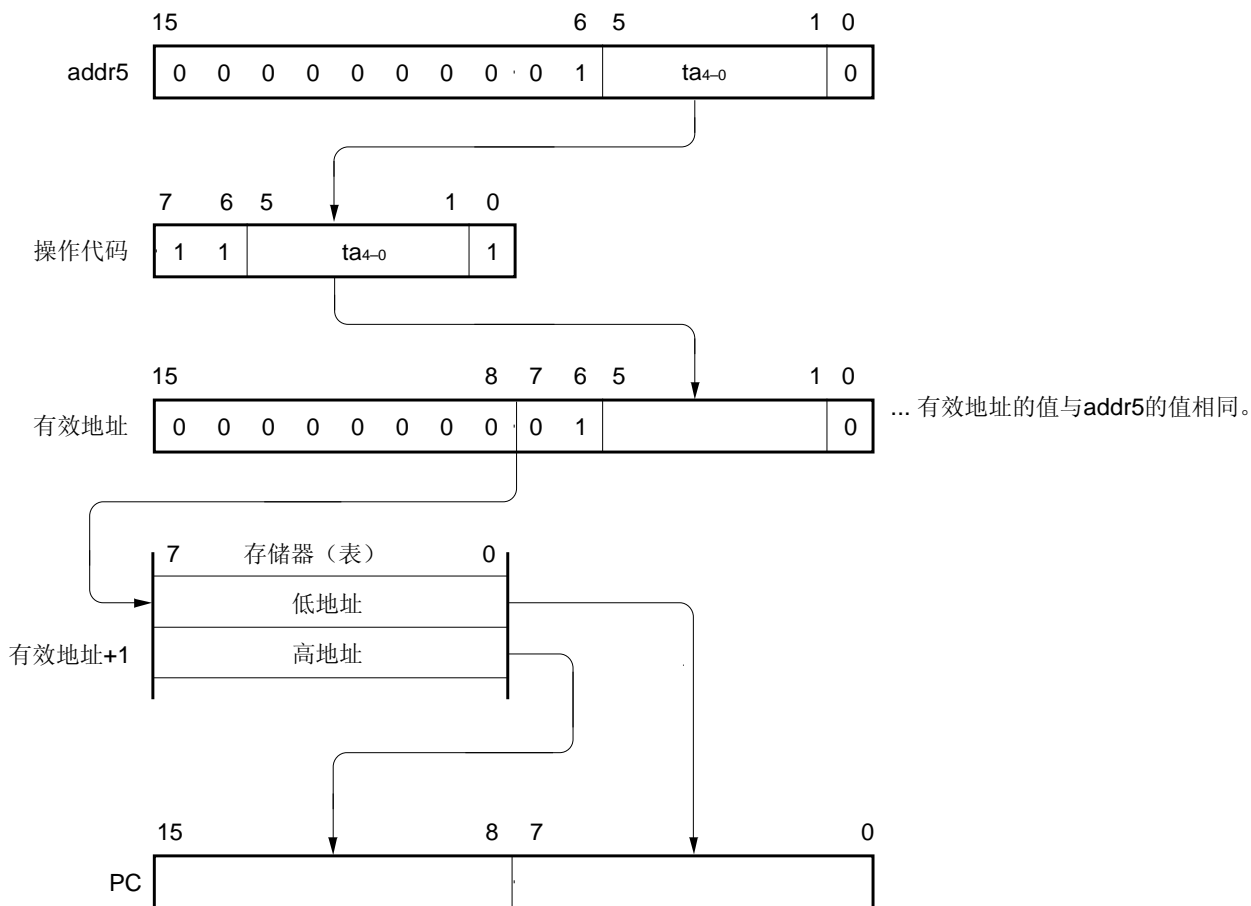
[功能]

通过指令码第 1 位到第 5 位的立即数，访问特定存储区中表的内容（转移目的地址），并将表的内容赋给程序计数器（PC），然后转向该地址执行程序。

该功能在执行 CALLT [addr5]指令时执行。

该指令参考了由 addr5 表示的并且存储在 0040H 至 007FH 的存储器表的地址，并且允许跳转至整个存储器空间。

[图示]

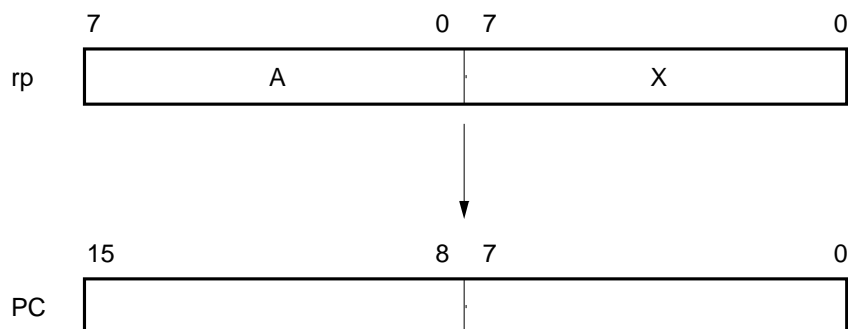


4.3.4 寄存器寻址

[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]



4.4 操作数地址寻址

以下方法来指定指令执行过程中寄存器和存储器（寻址）进行的操作。

4.4.1 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 PD78F8024 和 μ PD78F8025 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 寄存器用于存放进行十进制调整后的数值。
ROR4/ROL4	A 寄存器用于存放进行数字循环的数据。

[操作数格式]

由于隐含寻址可以通过指令自动进行，因此不需要特定的操作数格式。

[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。

4.4.2 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，操作码中将会有 3 位用于表示一个 8 位寄存器。

[操作数格式]

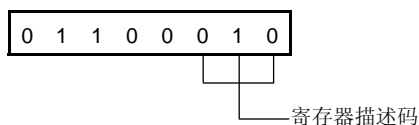
标识符	说明
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

'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]

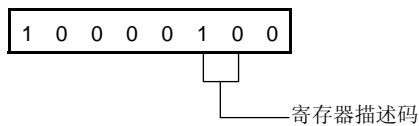
MOV A, C; 将选择 C 寄存器为 r 时

操作代码



INCW DE; 将 DE 寄存器对选作 rp 时

操作代码



4.4.3 直接寻址

[功能]

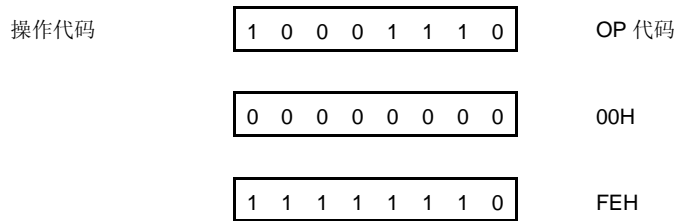
要进行操作的存储器通过变为操作数地址的指令字中的立即数据直接进行寻址。
所有存储器空间都可以实现直接寻址。

[操作数格式]

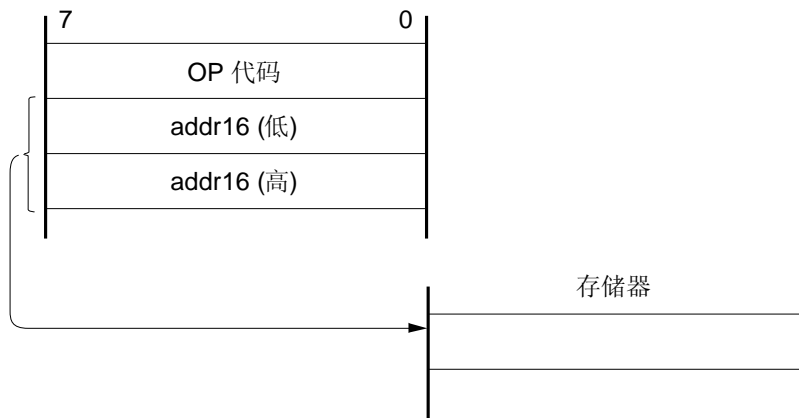
标识符	说明
addr16	标签或 16 位立即数据

[说明示例]

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



[图示]



4.4.4 短直接寻址

[功能]

在固定空间中要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。在程序中经常访问的端口和定时器 / 事件计数的比较和捕捉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

操作代码

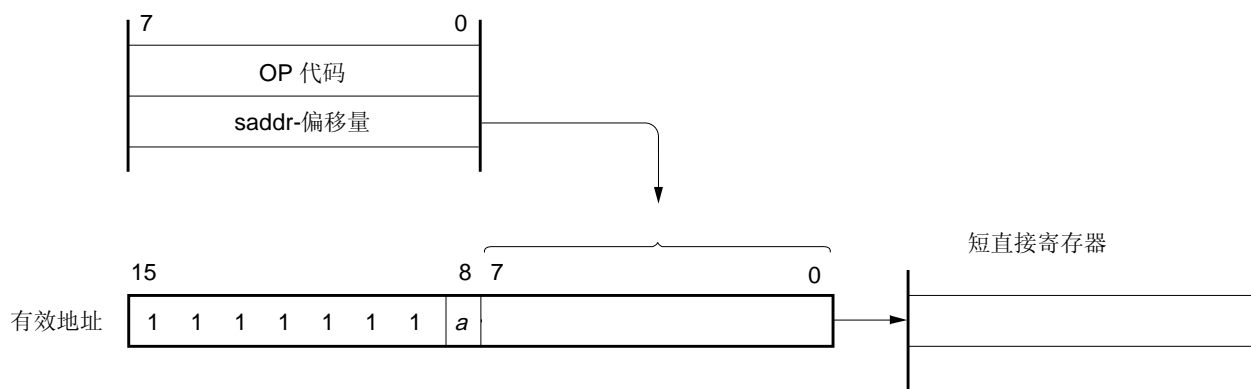
1 1 1 1 0 0 1 0

OP 代码

0 0 1 1 0 0 0 0

30H (saddr-偏移量)

[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, $\alpha = 1$

4.4.5 特殊功能寄存器 (SFR) 寻址

[功能]

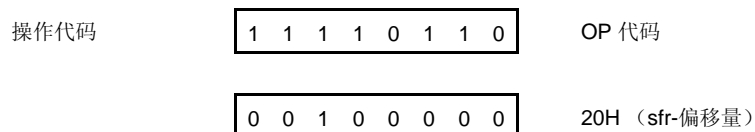
通过指令字中的 8 位立即数据对存储器的特殊功能寄存器 (SFR) 进行寻址。
 该寻址方式应用于 FF00H 至 FFCFH 以及 FFE0H 至 FFFFH 的 240 个字节的空间内。然而，映射在 FF00H 至 FF1FH 内的 SFR 可以通过短直接寻址方式进行访问。

[操作数格式]

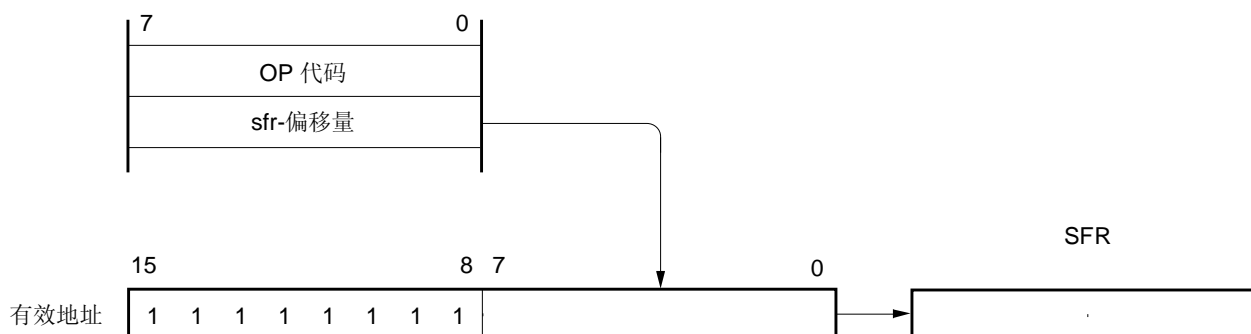
标识符	说明
sfr	特殊功能寄存器名
sfrp	16 位可操作特殊功能寄存器名 (仅限偶地址)

[说明示例]

MOV PM0, A; 当 PM0 (FF20H) 被选作 sfr 时



[图示]



4.4.6 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。

所有存储器空间都可以实现该寻址。

[操作数格式]

标识符	说明
-	[DE], [HL]

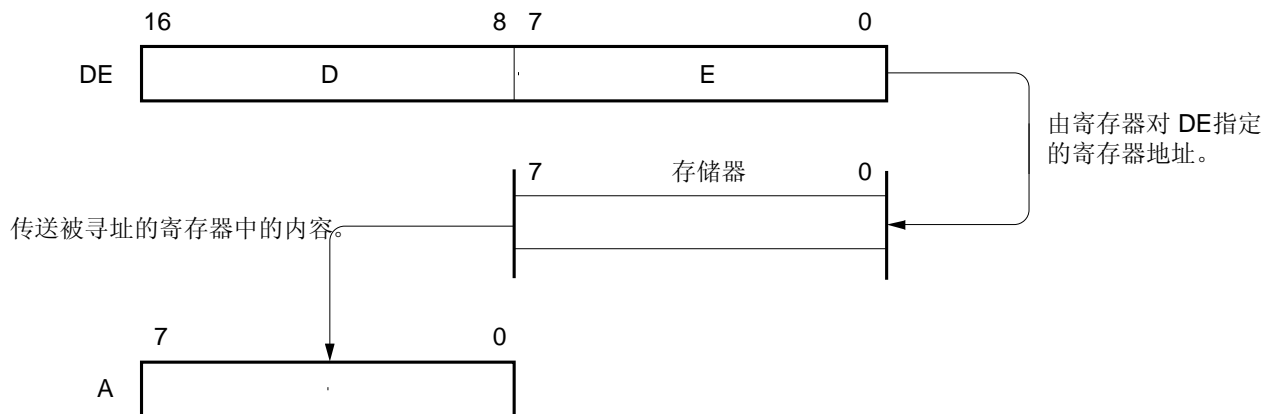
[说明示例]

MOV A, [DE]: 当[DE]被选作寄存器对时

操作代码

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

[图示]



4.4.7 基址寻址

[功能]

8 位立即数据作为偏移数据被加至基地址寄存器的内容中，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为 16 位的正数来完成的。第 16 位的进位忽略不计。
所有存储器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]

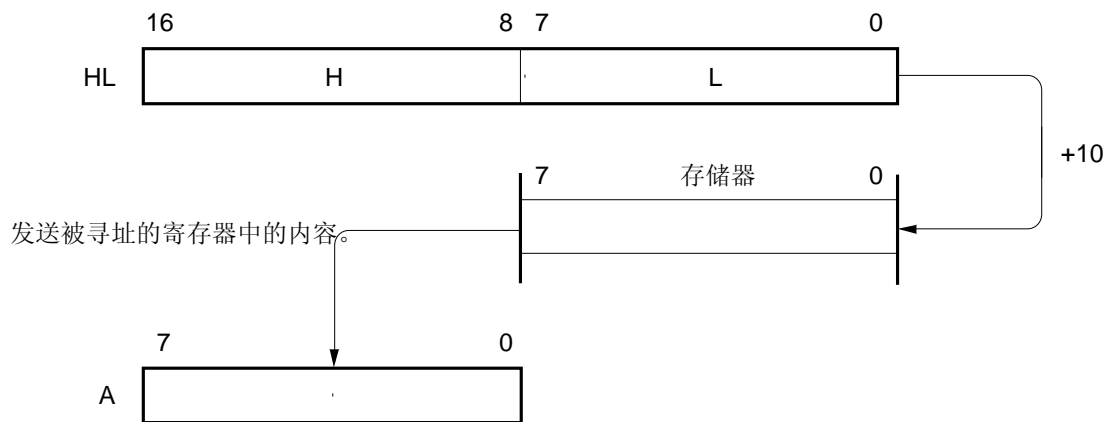
MOV A, [HL + 10H]; 将字节设置为 10H 时

操作代码

1 0 1 0 1 1 1 0

0 0 0 1 0 0 0 0

[图示]



4.4.8 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 寄存器内容扩展为 16 位的正数来完成的。第 16 位的进位忽略不计。
所有存储器空间都可以实现该寻址。

[操作数格式]

标识符	说明
-	[HL + B], [HL + C]

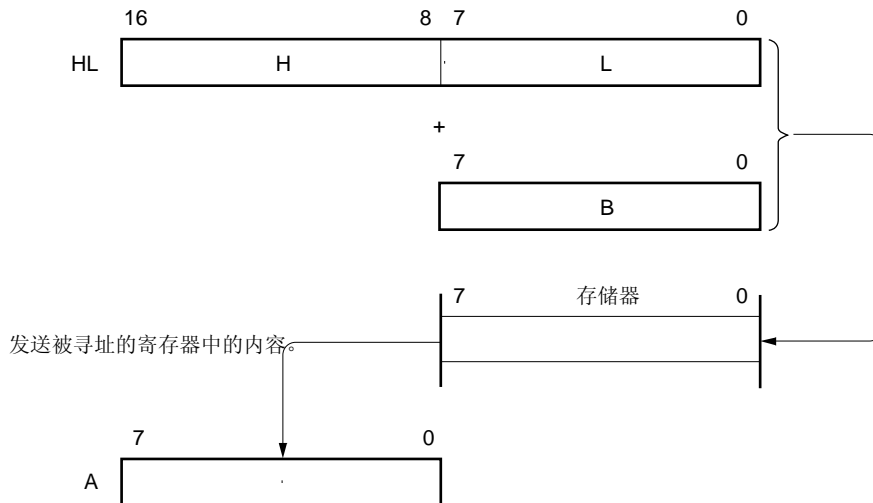
[说明示例]

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]



4.4.9 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/恢复时， 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。

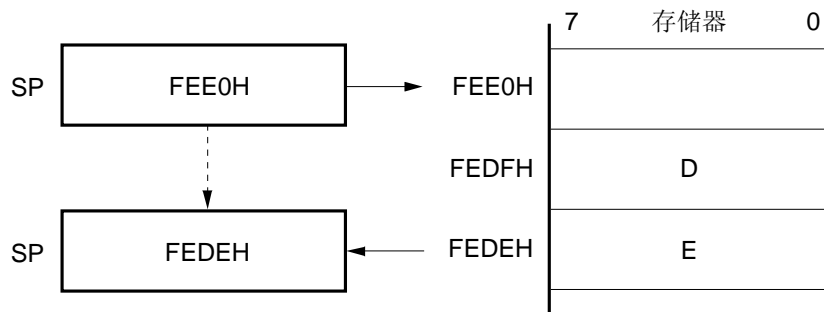
[说明示例]

PUSH DE; 保存 DE 寄存器时

操作代码

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

[图示]



第五章 端口功能

5.1 端口功能

共有两种类型的引脚 I/O 缓冲器供电电源：AVREF 以及 VDD。这些供电电源和引脚间的关系如下所示。

表 5-1. 引脚 I/O 缓冲式供电电源

电源	对应引脚
AVREF	P20 至 P23
VDD	除 P20 至 P23 外的引脚

μ PD78F8024 和 μ PD78F8025 提供的端口如图 5-1 中所示，这些端口支持多种控制操作。各个端口的功能如表 5-2 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 2 章 引脚功能。

图 5-1. 端口类型

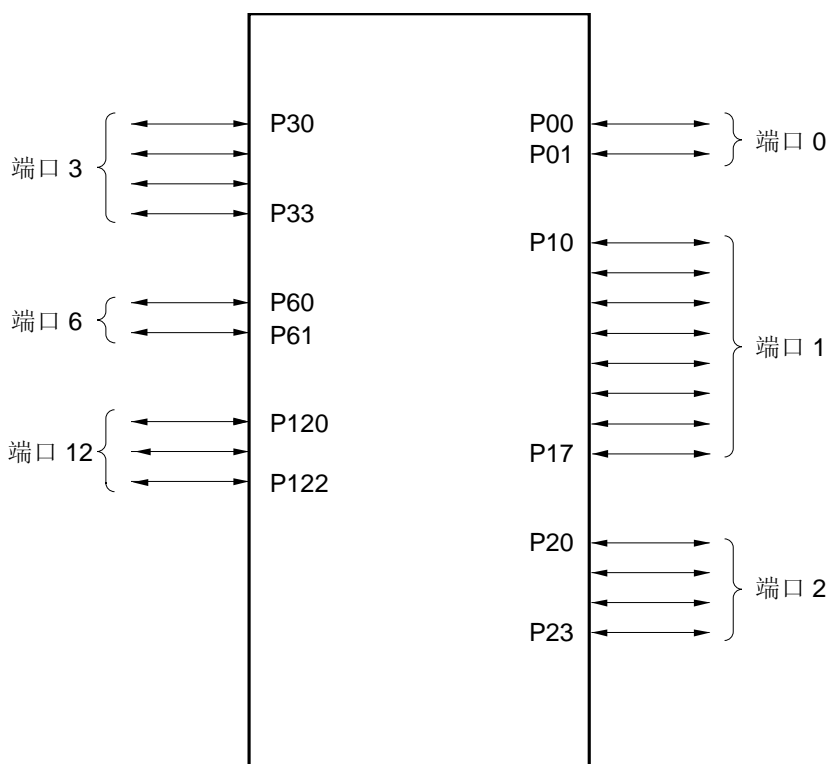


表 5-2. 端口功能

功能名称	I/O	功能	复位后	复用功能
P00	I/O	端口 0。 2 位 I/O 端口。 可以以 1 位为单位指定输入 / 输出。 可以通过软件设置来指定片上上拉电阻的使用。	输入端口	TI000
P01				TI010/TO00 (/EN)
P10	I/O	端口 1。 8 位 I/O 端口。 可以以 1 位为单位指定输入 / 输出。 可以通过软件设置来指定片上上拉电阻的使用。	输入端口	$\overline{\text{SCK10}}/\text{TxD0}$
P11				SI10/RxD0
P12				SO10
P13				TxD6
P14				RxD6
P15				TOH0 (/PWM1)
P16				TOH1/INTP5 (/PWM2)
P17				TI50/TO50 (/PWM3)
P20 至 P23	I/O	端口 2。 4 位 I/O 端口。 可以以 1 位为单位指定输入 / 输出。	模拟输入	ANI0 至 ANI3
P30	I/O	端口 3。 4 位 I/O 端口。 可以以 1 位为单位指定输入 / 输出。 可以通过软件设置来指定片上上拉电阻的使用。	输入端口	INTP1 (/SH)
P31				INTP2
P32				INTP3
P33				INTP4/TI51/TO51 (/PWM0)
P60	I/O	端口 6。 2 位 I/O 端口。 输出为 N-ch 漏极开路输出 (6 V 耐压)。 可以以 1 位为单位指定输入 / 输出。	输入端口	SCL0
P61				SDA0
P120	I/O	端口 12。 3 位 I/O 端口。 可以以 1 位为单位指定输入 / 输出。 只有 P120 可以通过软件设置来指定片上上拉电阻的使用。	输入端口	INTP0/EXLVI
P121				X1
P122				X2/EXCLK

备注 圆括号内的字符为内部引脚。

5.2 端口配置

端口包含有以下硬件。

表 5-3. 端口配置

项目	配置
控制寄存器	端口模式寄存器 (PM0 至 PM3, PM6, PM12) 端口寄存器 (P0 至 P3, P6, P12) 上拉电阻选择寄存器 (PU0, PU1, PU3, PU12) A/D 端口配置寄存器 (ADPC)
端口	总数: 23 (CMOS I/O: 21, N-ch 漏极开路 I/O: 2)
上拉电阻	总数: 15

5.2.1 端口 0

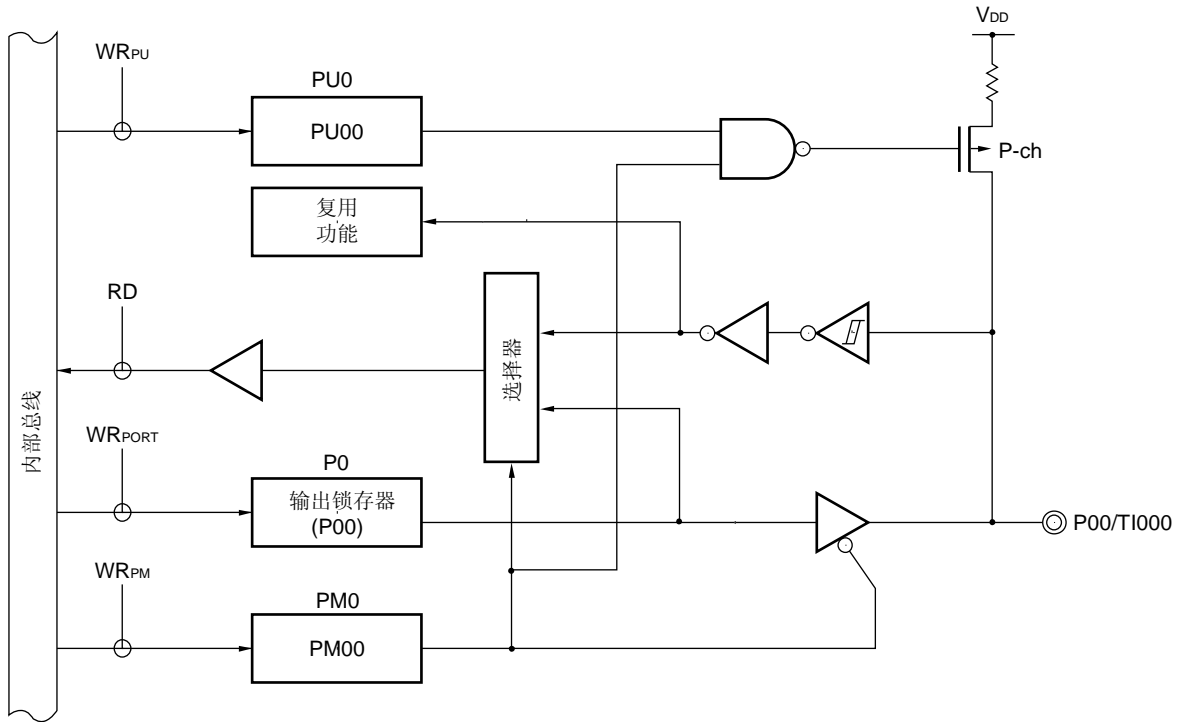
端口 0 是带有输出锁存的 2 位 I/O 端口。端口 0 可以通过使用端口模式寄存器 0 (PM0) 以 1 位为单位设置为输入模式或输出模式。当 P00 和 P01 引脚用作输入端口时，片上上拉电阻可以通过上拉电阻选择寄存器 0 (PU0) 以 1 位为单位进行指定。

该端口也可以用于定时器 I/O。

复位信号将端口 0 设置为输入模式。

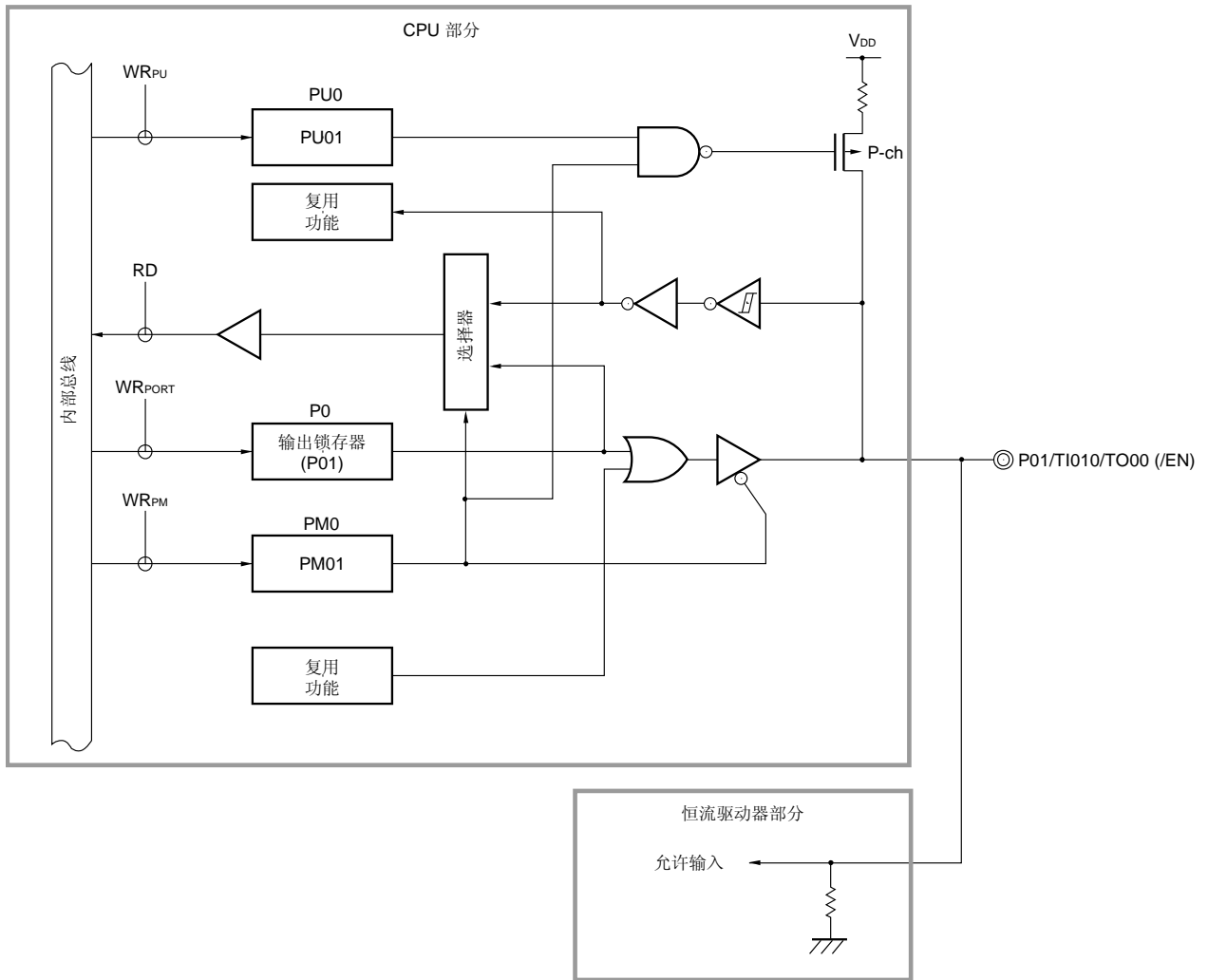
图 5-2 和 5-3 显示了端口 0 的框图。

图 5-2. P00 的框图



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

图 5-3. P01 的框图



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

备注 P01/TI010/TO00 引脚可以对恒流驱动器部分的 EN 引脚（内部引脚）进行操作，这是因为它连接在恒流驱动器部分的 EN 引脚的封装之内。

5.2.2 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。

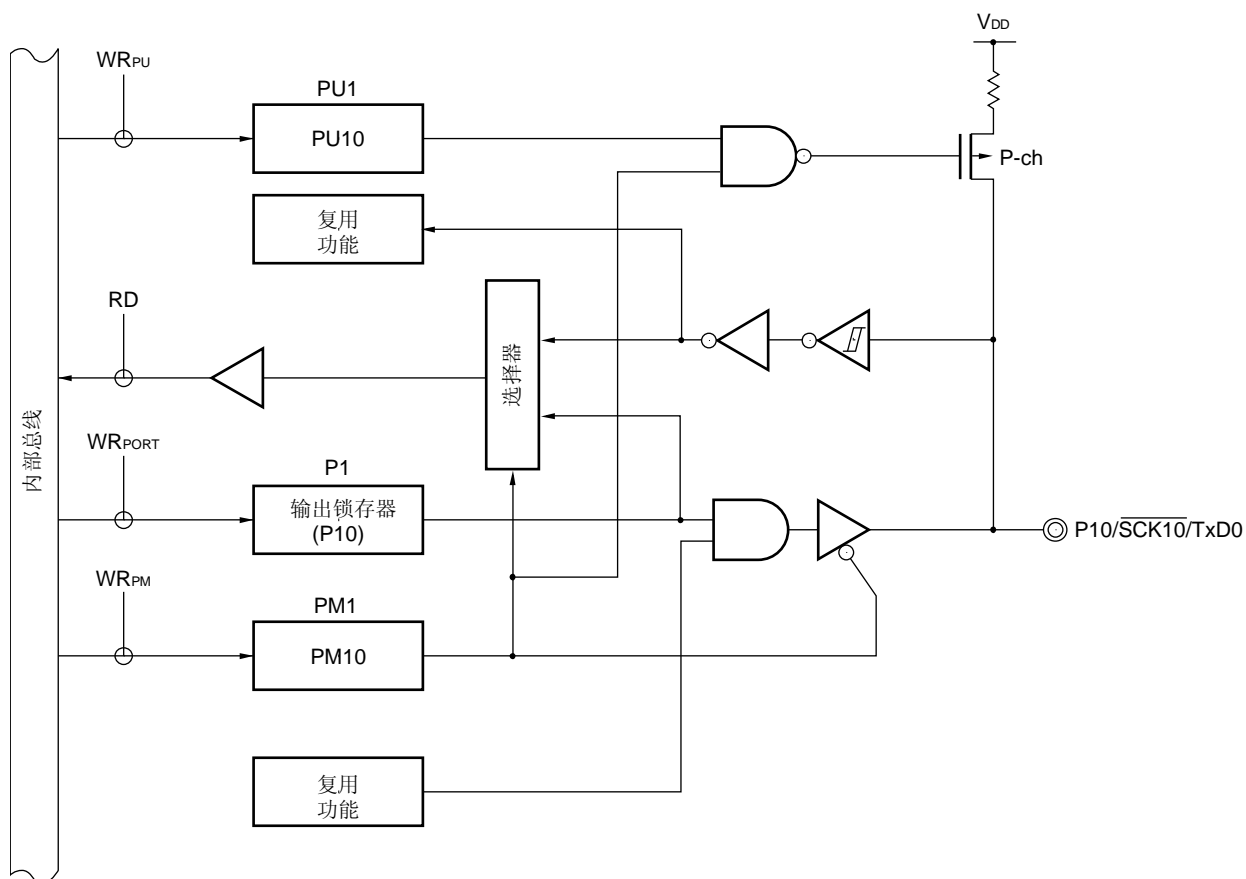
该端口也可用于外部中断请求输入, 串行接口数据 I/O, 时钟 I/O, 以及定时器 I/O。

复位信号将端口 1 设置为输入模式。

图 5-4 至 5-9 显示了端口 1 的框图。

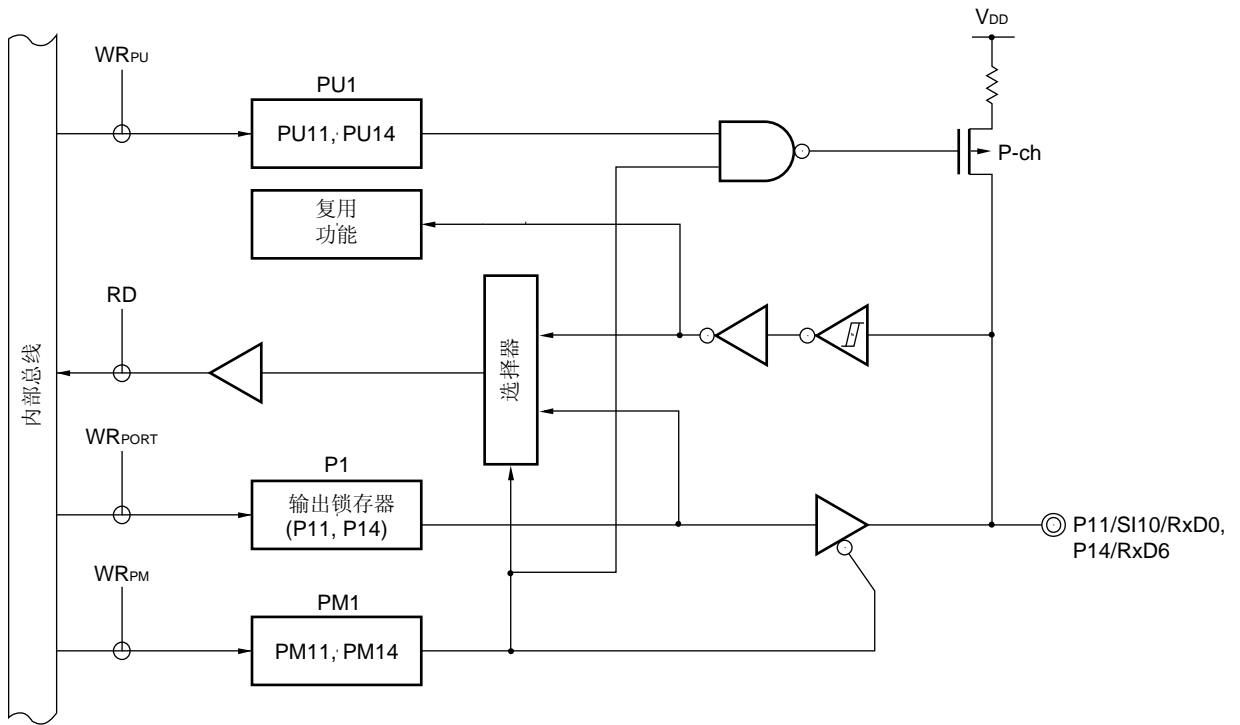
- 注意事项**
1. 若要将 P10/SCK10/TxD0 和 P12/SO10 用作通用端口, 则应将串行模式寄存器 10 (CSIM10) 以及串行时钟选择寄存器 10 (CSIC10) 设置为默认状态 (00H)。
 2. 使用 P13/TxD6 作为通用端口时, 设置异步串行接口控制寄存器 6 (ASICL6) 的位 0 为 0 (TxD6 的通常输出)。

图 5-4. P10 的框图



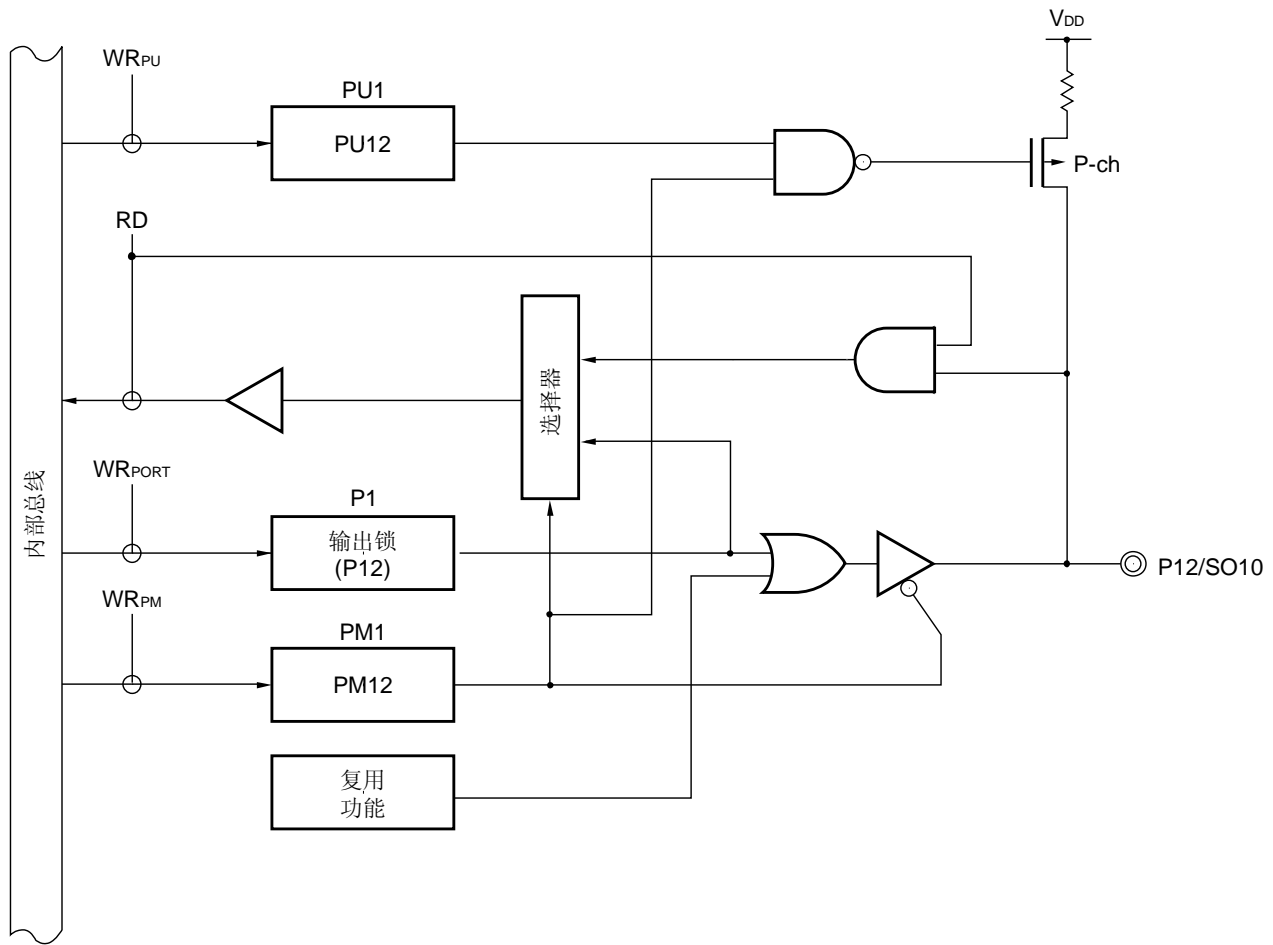
- P1: 端口寄存器 1
 PU1: 上拉电阻选择寄存器 1
 PM1: 端口模式寄存器 1
 RD: 读取信号
 WR_{xx}: 写入信号

图 5-5. P11 和 P14 的框图



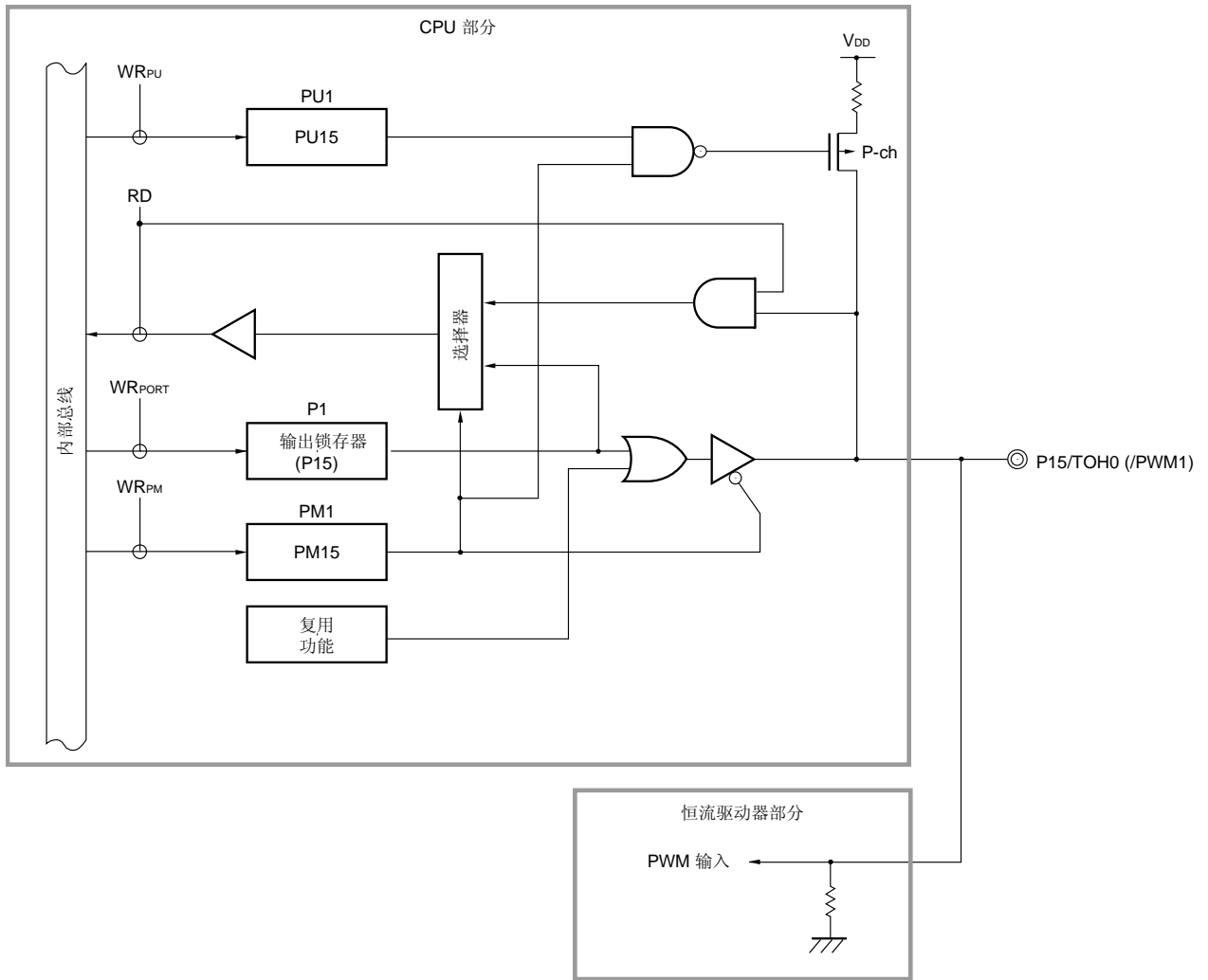
- P1: 端口寄存器 1
- PU1: 上拉电阻选择寄存器 1
- PM1: 端口模式寄存器 1
- RD: 读取信号
- WR_{xx}: 写入信号

图 5-6. P12 的框图



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

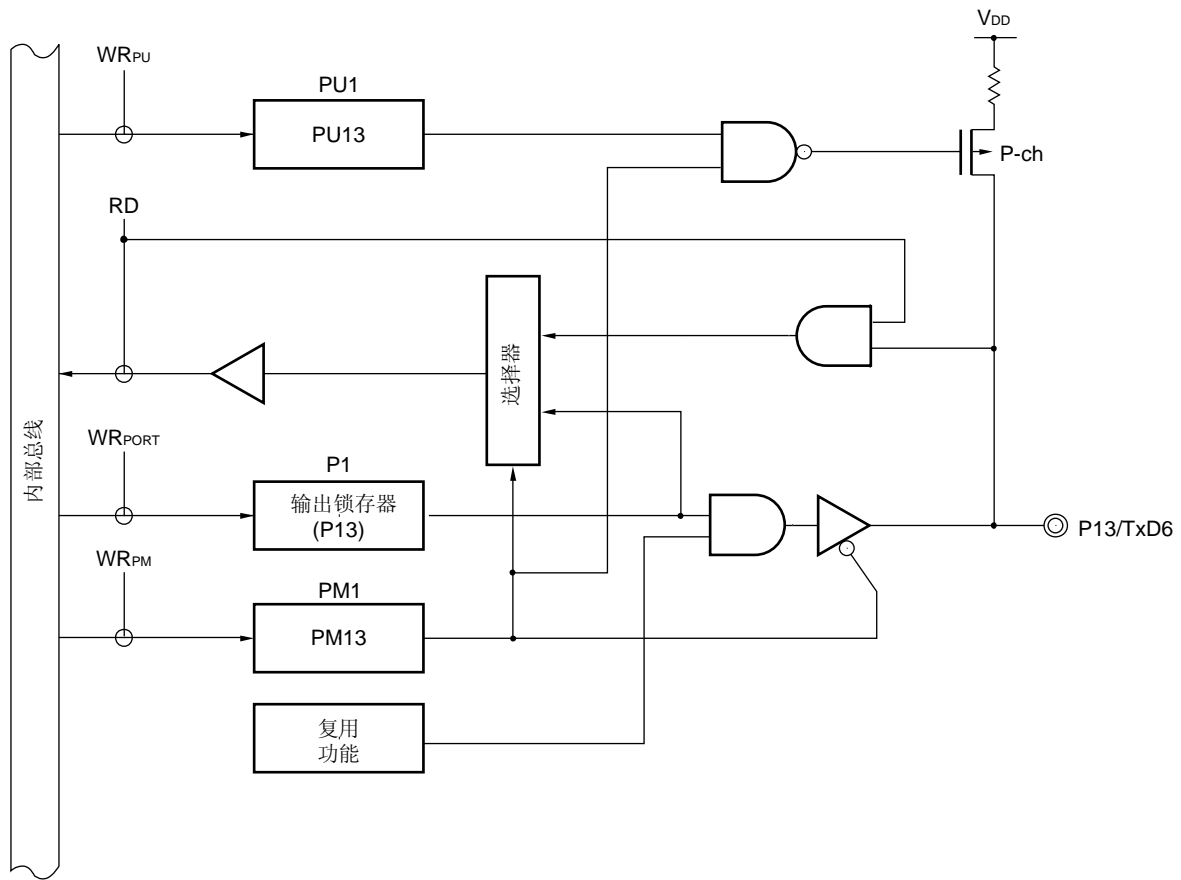
图 5-7. P15 的框图



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

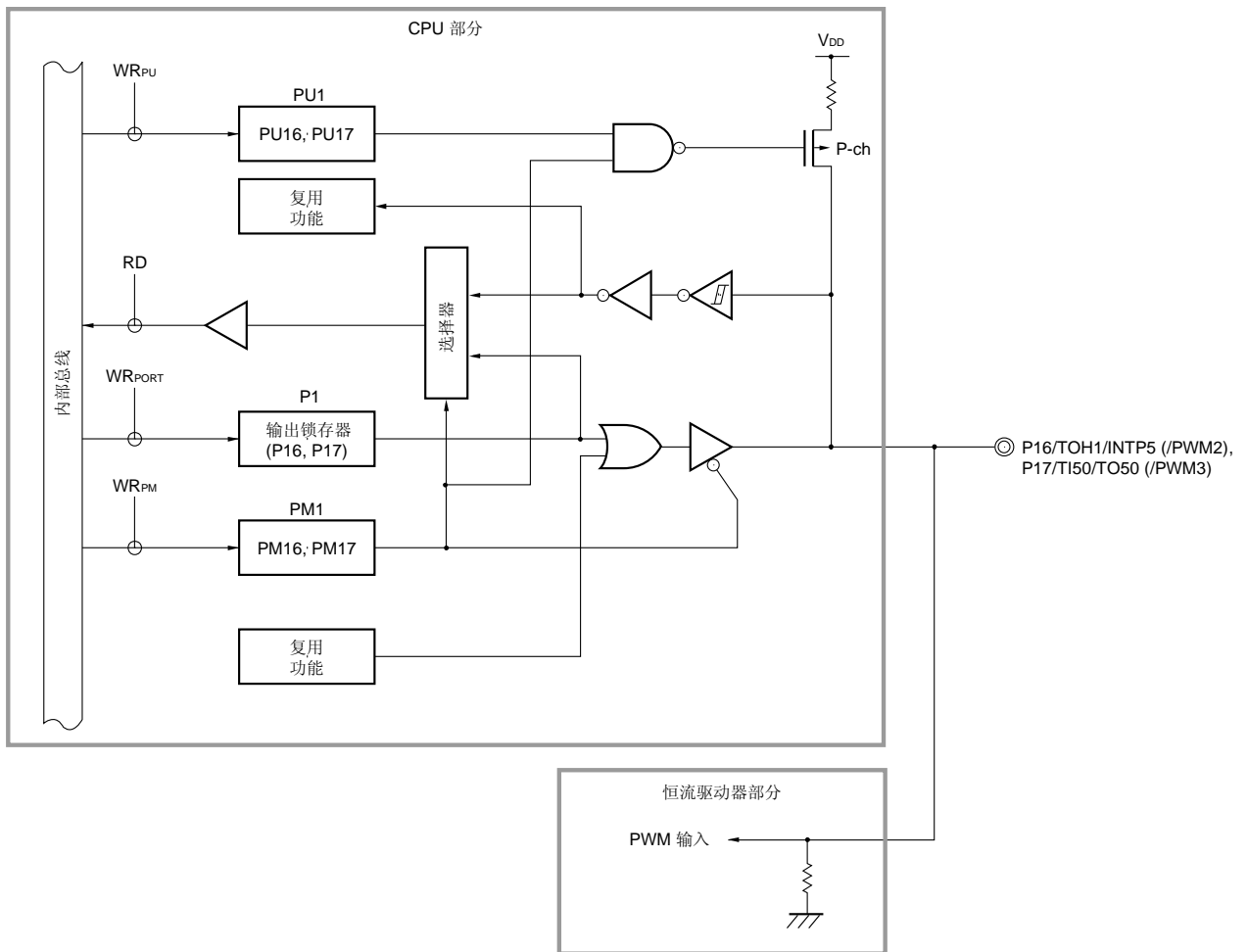
备注 P15/TOH0 引脚可以对恒流驱动器部分的 PWM1 引脚（内部引脚）进行操作，这是因为它们连接在恒流驱动器部分的 PWM1 引脚的封装之内。

图 5-8. P13 的框图



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

图 5-9. P16 和 P17 的框图



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

备注 P16/TOH1/INTP5 以及 P17/TI50/TO50 引脚可以对恒流驱动器部分的 PWM2 以及 PWM3 引脚（内部引脚）进行操作，这是因为它们分别连接在恒流驱动器部分的 PWM2 以及 PWM3 引脚的封装之内。

5.2.3 端口 2

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 2 (PM2) 以 1 位为单位设置为输入模式或输出模式。

该端口也可用于 A/D 转换器模拟输入。

要将 P20/ANI0 至 P23/ANI3 用作数字输入引脚，则应通过使用 A/D 端口配置寄存器 (ADPC) 设置为数字 I/O 模式，并设置 PM2 为输入模式。从低位开始使用这些引脚。

要将 P20/ANI0 至 P23/ANI3 用作数字输出引脚，则应通过使用 ADPC 设置为数字 I/O 模式，并设置 PM2 为输出模式。

表 5-4. P20/ANI0 至 P23/ANI3 引脚的设置功能

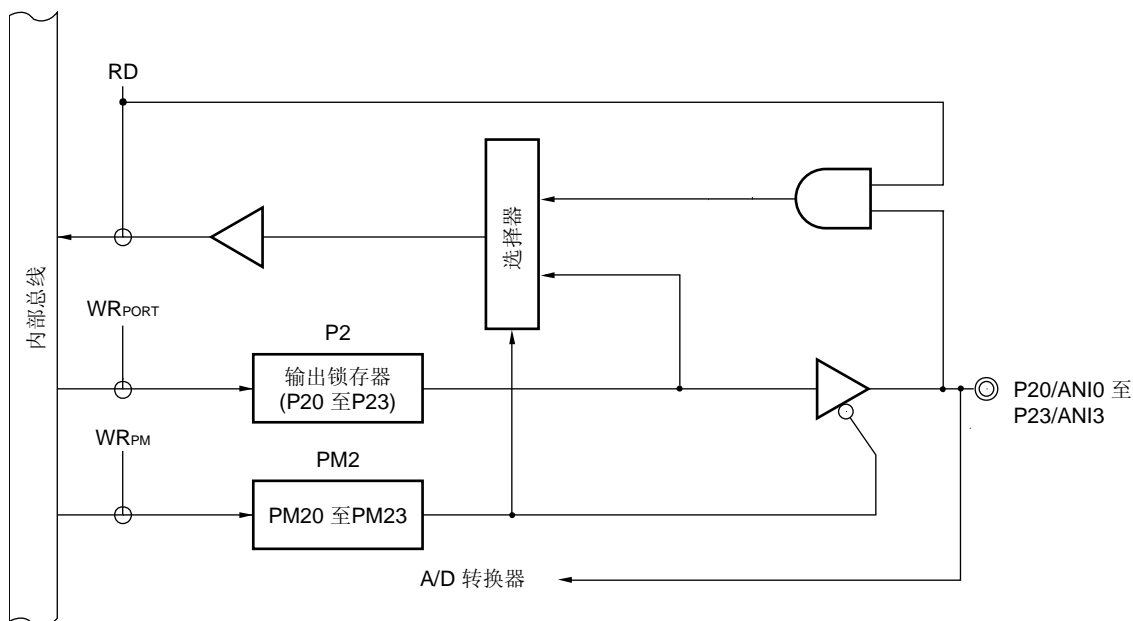
ADPC	PM2	ADS	P20/ANI0 至 P23/ANI3 引脚
数字 I/O 选择	输入模式	-	数字输入
	输出模式	-	数字输出
模拟输入选择	输入模式	选择 ANI。	模拟输入 (要进行转换)
		不选择 ANI。	模拟输入 (不进行转换)
	输出模式	选择 ANI。	禁止设置
		不选择 ANI。	

复位信号的产生时，P20/ANI0 至 P23/ANI3 设置为模拟输入模式

图 5-10 显示了端口 2 的框图。

注意事项 当端口 2 被用作数字端口时，将 AVREF 引脚的电平设置为 VDD 引脚的电平。

图 5-10. P20 至 P23 的框图



- P2: 端口寄存器 2
- PM2: 端口模式寄存器 2
- RD: 读取信号
- WRxx: 写入信号

5.2.4 端口 3

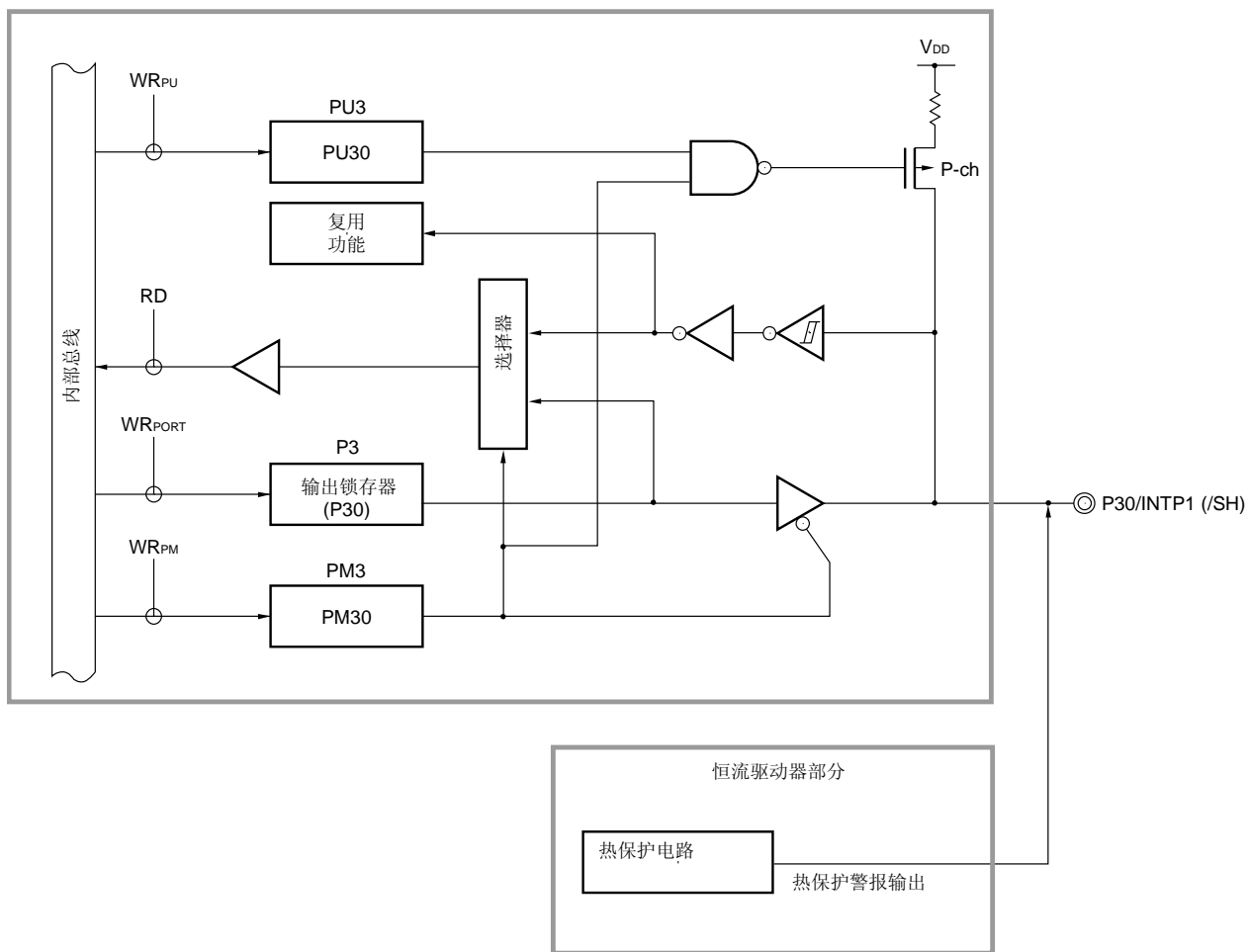
端口 3 是带有输出锁存的 4 位 I/O 端口。端口 3 可以通过使用端口模式寄存器 3 (PM3) 以 1 位为单位设置为输入或输出模式。当 P30 至 P33 引脚用作输入端口时，片上上拉电阻可以通过上拉电阻选择寄存器 3 (PU3) 以 1 位为单位进行指定。

该端口也可用于外部中断请求输入以及定时器 I/O。

复位信号将端口 3 设置为输入模式。

图 5-11 至 5-13 显示了端口 3 的框图。

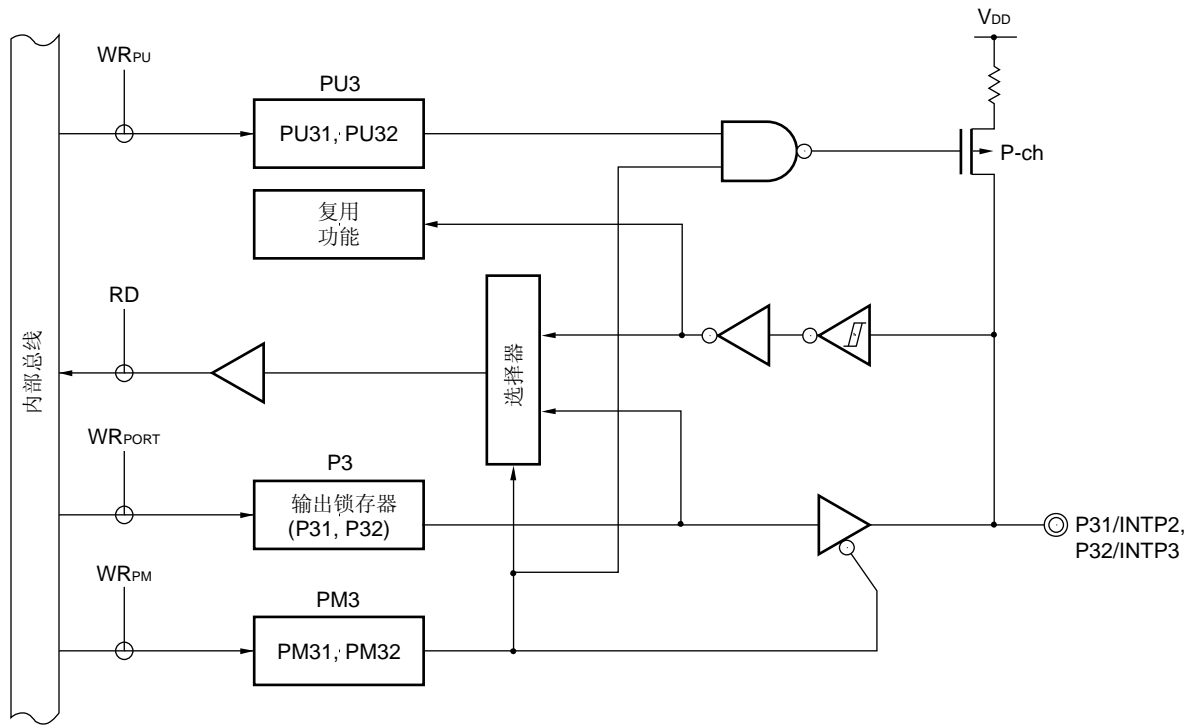
图 5-11. P30 的框图



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

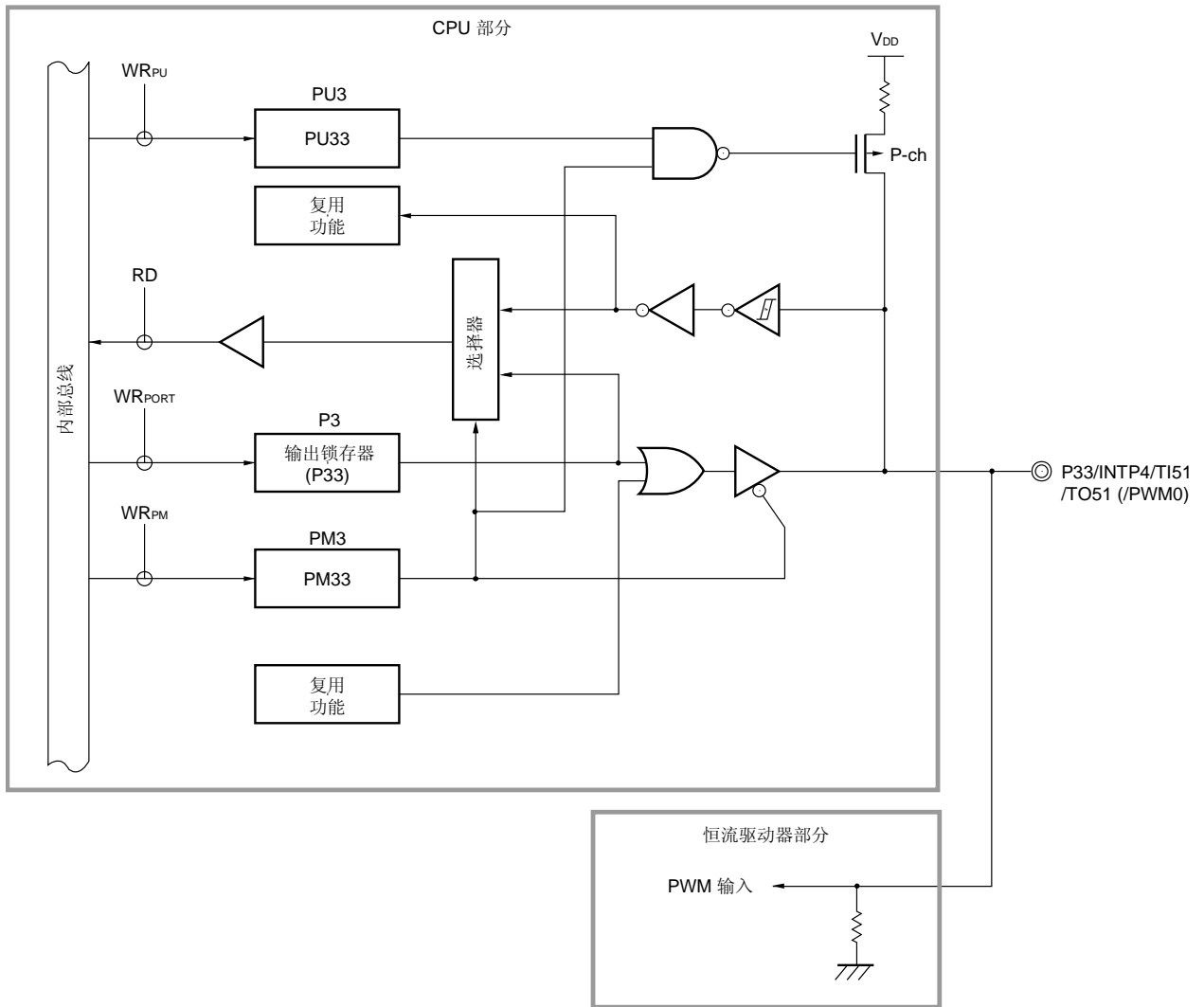
备注 P30/INTP1 引脚可以对恒流驱动器部分的 SH 引脚（内部引脚）进行操作，这是因为它连接在恒流驱动器部分的 SH 引脚的封装之内。

图 5-12. P31 和 P32 的框图



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

图 5-13. P33 的框图



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

备注 P33/INTP4/TO51/TI51 引脚可以对恒流驱动器部分的 PWM0 引脚（内部引脚）进行操作，这是因为它连接在恒流驱动器部分的 PWM0 引脚的封装之内。

5.2.5 端口 6

端口 6 是带有输出锁存的 2 位 I/O 端口。端口 6 可以通过使用端口模式寄存器 6 (PM6) 以 1 位为单位设置为输入模式或输出模式。

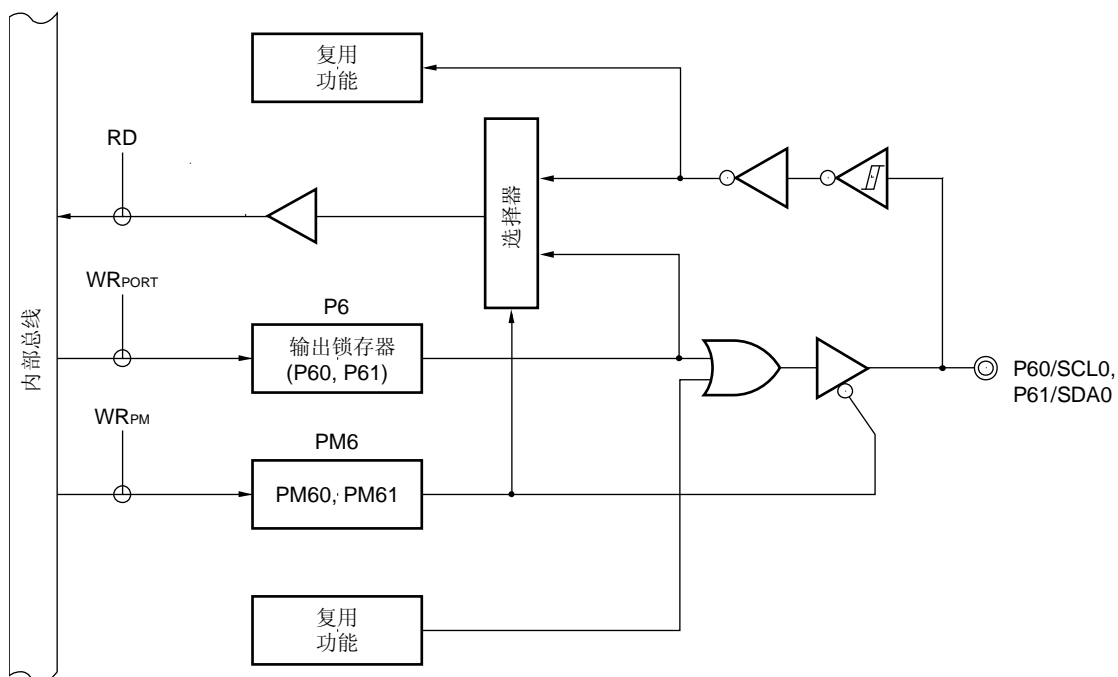
输出为 N-ch 漏极开路输出 (耐压 6V)。

该端口也可用于串行接口数据 I/O 和时钟 I/O。

复位信号将端口 6 设置为输入模式。

图 5-14 显示了端口 6 的框图。

图 5-14. P60 和 P61 的框图



P6: 端口寄存器 6
 PM6: 端口模式寄存器 6
 RD: 读取信号
 WR_{xx}: 写入信号

<R> 注意事项 如果一个中间电压输入到这些引脚, 直通电流将流过 P60 和 P61 引脚, 因为当 P60 和 P61 工作在输出模式时输入缓冲器也是打开的。因此, 当 P60 和 P61 工作在输出模式时不要向这些引脚输入中间电压。

5.2.6 端口 12

端口 12 是带有输出锁存的 3 位 I/O 端口。端口 12 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。当 P120 只用作输入端口时, 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。

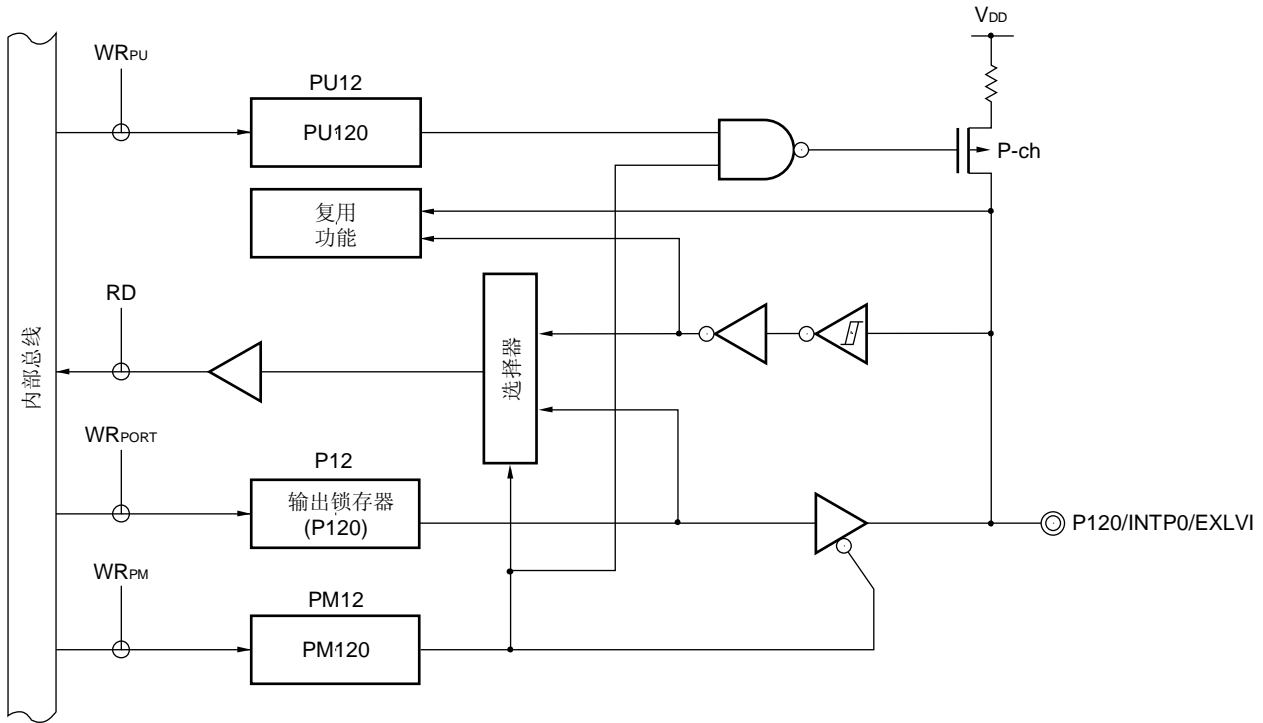
该端口也可用于外部中断请求输入, 外部低压检测的电平输入, 并且用于连接主系统时钟的振荡器以及主系统时钟的外部时钟输入。

复位信号将端口 12 设置为输入模式。

图 5-15 和 5-16 显示了端口 12 的框图。

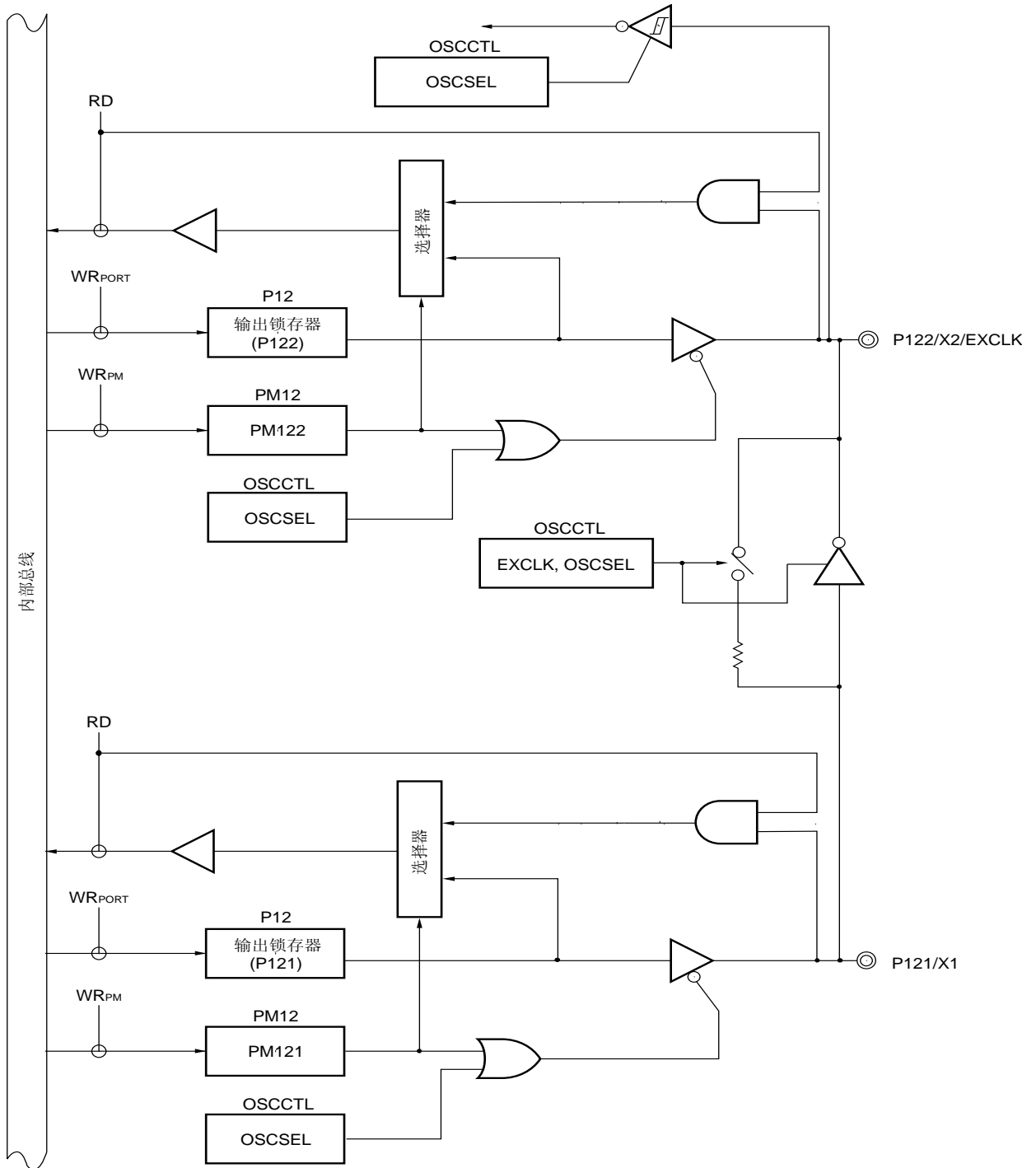
注意事项 在使用 P121 和 P122 引脚连接主系统时钟 (X1, X2) 的振荡器或使用 P121 和 P122 来输入主系统时钟 (EXCLK) 的外部时钟时, 必须通过使用时钟操作模式选择寄存器 (OSCCTL) 来设置 X1 振荡模式或外部时钟输入模式 (详细信息, 请参见 5.3 (1) 时钟操作模式选择寄存器 (OSCCTL))。OSCCTL 的复位值为 00H (P121 和 P122 引脚都是 I/O 端口引脚)。此时, 不需要对 PM121, PM122, P121, 以及 P122 引脚进行设置。

图 5-15. P120 的框图



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

图 5-16. 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

5.3 寄存器控制端口功能

端口功能由以下四种类型的寄存器进行控制。

- 端口模式寄存器 (PM0 至 PM3, PM6, PM12)
- 端口寄存器 (P0 至 P3, P6, P12)
- 上拉电阻选择寄存器 (PU0, PU1, PU3, PU12)
- A/D 端口配置寄存器 (ADPC)

(1) 端口模式寄存器 (PM0 至 PM3, PM6, 以及 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。

复位信号的产生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 5.5 使用复用功能时的端口模式寄存器及输出锁存器的设置来设置端口模式寄存器。

图 5-17. 端口模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	1	1	1	1	1	PM01	PM00	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	1	1	1	1	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FF26H	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 至 3, 6, 12; n = 0 至 7)
0	输出模式 (输出缓冲器开)
1	输入模式 (输出缓冲器关)

注意事项 请务必将 PM0 的第 2 位至第 7 位, PM2 的第 4 位至第 7 位, PM3 的第 4 位至第 7 位, PM6 的第 2 位至第 7 位, PM12 的第 3 位至第 7 位设置为 1。

(2) 端口寄存器 (P0 至 P3, P6, 以及 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，读取的是引脚的电平。如果在输出模式中读取，读取的是输出锁存器的值。

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。

复位信号的产生将这些寄存器设置为 00H。

图 5-18. 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	0	0	0	0	P23	P22	P21	P20	FF02H	00H (输出锁存)	R/W
P3	0	0	0	0	P33	P32	P31	P30	FF03H	00H (输出锁存)	R/W
P6	0	0	0	0	0	0	P61	P60	FF06H	00H (输出锁存)	R/W
P12	0	0	0	0	0	P122 ^注	P121 ^注	P120	FF0CH	00H (输出锁存)	R/W

Pmn	m = 0 至 3, 6, 12; n = 0 至 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 如果引脚处于外部时钟输入模式中，那么通常会从 P121 和 P122 的输出锁存器中读取到“0”。

(3) 上拉电阻选择寄存器 (PU0, PU1, PU3, 和 PU12)

这些寄存器用于指定是否使用 P00, P01, P10 至 P17, P30 至 P33, 或 P120 的片上上拉电阻。片上上拉电阻可以以 1 位单元的指定, 用于那些已设置为输入模式的引脚, 并且这些引脚已通过 PU0, PU1, PU3 和 PU12 指定为使用内部上拉电阻。无论 PU0, PU1, PU3, 和 PU12 设置为何值, 设置为输出模式的引脚和用作复用功能的输出引脚都不连接内部上拉电阻。

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。

复位信号的产生将这些寄存器设置为 00H。

图 5-19. 上拉电阻选择寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU0	0	0	0	0	0	0	PU01	PU00	FF30H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W
PU3	0	0	0	0	PU33	PU32	PU31	PU30	FF33H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	FF3CH	00H	R/W

PU _{mn}	P _{mn} 引脚片上上拉电阻选择 (m = 0, 1, 3, 12; n = 0 至 7)
0	未连接片上上拉电阻
1	已连接片上上拉电阻

(4) A/D 端口配置寄存器 (ADPC)

该寄存器将 P20/ANI0 至 P23/ANI3 引脚在端口的数字 I/O 或 A/D 转换器的模拟输入中切换。

ADPC 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号的产生将该寄存器设置为 00H。

图 5-20. A/D 端口配置寄存器 (ADPC) 的格式

地址: FF2FH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADPC	0	0	0	0	0	ADPC2	ADPC1	ADPC0

ADPC2	ADPC1	ADPC0	数字 I/O (D)/模拟输入 (A) 切换			
			P23/ANI3	P22/ANI2	P21/ANI1	P20/ANI0
0	0	0	A	A	A	A
0	0	1	A	A	A	D
0	1	0	A	A	D	D
0	1	1	A	D	D	D
1	0	0	D	D	D	D
除上述情况			禁止设置			

- 注意事项**
1. 通过使用端口模式寄存器 2 (PM2) 将用于 A/D 转换的通道设置为输入模式。
 2. 如果数据被写入 ADPC, 则将生成一个等待周期。详细信息, 请参见第 27 章 等待注意事项。
 3. 确保将第 3 位至第 7 位清除为 0。

5.4 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。

5.4.1 写入I/O端口

(1) 输出模式

使用传送指令对输出锁存器进行写操作，输出锁存器的内容从引脚输出。

一旦数据写入输出锁存器，它将一直保存直到写入新数据。

复位信号的产生时，输出锁存器的数据被清除。

(2) 输入模式

使用传送指令对输出锁存器进行写操作，因为输出缓冲器处于关闭状态，所以引脚状态不会改变。

一旦数据写入输出锁存器，它将一直保存到新数据被写入。

复位信号的产生时，输出锁存器的数据被清除。

5.4.2 从I/O端口中读取

(1) 输出模式

使用传送指令读取的是输出锁存器的内容。输出锁存器的内容不会改变。

(2) 输入模式

使用传送指令读取的是引脚状态。输出锁存器的内容不会改变。

5.4.3 I/O端口上的操作

(1) 输出模式

对输出锁存器执行一个操作时，操作结果写入输出锁存器。而输出锁存器的内容则从引脚输出。

一旦数据写入输出锁存器，它将一直保存到新数据被写入。

复位信号的产生时，输出锁存器的数据被清除。

(2) 输入模式

读取引脚电平，对它的内容执行操作，操作的结果写入输出锁存器，因为输出缓冲器处于关闭状态，所以引脚状态不会改变。

复位信号的产生时，输出锁存器的数据被清除。

5.5 使用复用功能时端口模式寄存器以及输出锁存器的设置

若要使用端口引脚的复用功能，则应按表 5-5 中所示对端口模式寄存器和输出锁存器进行设置。

表 5-5. 使用复用功能时端口模式寄存器以及输出锁存器的设置

引脚名称	复用功能		PM _{xx}	P _{xx}
	功能名称	I/O		
P00	TI000	输入	1	×
P01	TI010	输入	1	×
	TO00	输出	0	0
P10	SCK10	输入	1	×
		输出	0	1
	TxD0	输出	0	1
P11	SI10	输入	1	×
	RxD0	输入	1	×
P12	SO10	输出	0	0
P13	TxD6	输出	0	1
P14	RxD6	输入	1	×
P15	TOH0	输出	0	0
P16	TOH1	输出	0	0
	INTP5	输入	1	×
P17	TI50	输入	1	×
	TO50	输出	0	0
P20 至 P23 ^{‡1}	ANI0 至 ANI3 ^{‡1}	输入	1	×
P30 至 P32	INTP1 至 INTP3	输入	1	×
P33	INTP4	输入	1	×
	TI51	输入	1	×
	TO51	输出	0	0
P60	SCL0	I/O	0	0
P61	SDA0	I/O	0	0
P120	INTP0	输入	1	×
	EXLVI	输入	1	×
P121	X1 ^{‡2}	–	×	×
P122	X2 ^{‡2}	–	×	×
	EXCLK ^{‡2}	输入	×	×

备注 ×: 不考虑
 PM_{xx}: 端口模式寄存器
 P_{xx}: 端口输出锁存器

(注 1 和 2 列于下一页中。)

- 注 1. ANI0/P20 至 ANI3/P23 引脚的功能可以通过使用 A/D 端口配置寄存器 (ADPC)，模拟输入通道选择寄存器 (ADS) 以及 PM2 来进行选择。

表 5-6. ANI0/P20 至 ANI3/P23 引脚的设置功能

ADPC	PM2	ADS	ANI0/P20 至 ANI3/P23 引脚
模拟输入选择	输入模式	选择 ANI。	模拟输入 (要进行转换)
		不选择 ANI。	模拟输入 (不进行转换)
	输出模式	选择 ANI。	禁止设置
		不选择 ANI。	
数字 I/O 选择	输入模式	-	数字输入
	输出模式	-	数字输出

2. 在使用 P121 和 P122 引脚连接主系统时钟 (X1, X2) 的振荡器或使用 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 6.3 (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 进行设置。

5.6 端口寄存器 n (Pn) 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器的值也可能被写入。

因此，当端口从输入模式切换到输出模式时，推荐重写输出锁存器。

<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器和引脚状态。
 μ PD78F8024 中，1 位操作指令按照以下顺序执行。

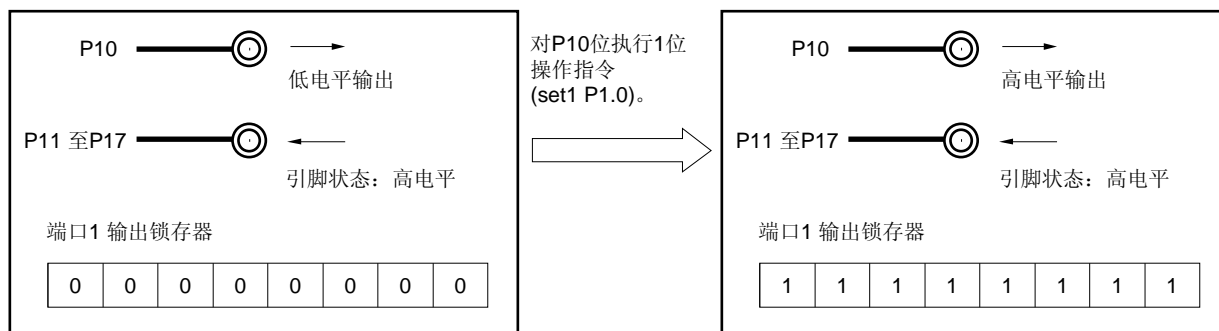
- <1> 以 8 位为单位读取 Pn 寄存器。
- <2> 操作目标位。
- <3> 以 8 位为单位写入 Pn 寄存器。

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 引脚的输出锁存器 (0) 的值被读取。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 被写到输出锁存器中。

图 5-21. 位操作指令 (P10)



P10 位的1位操作指令

- <1> 端口寄存器1 (P1) 以8位为单位进行读取。
 - P10用作输出端口时，端口输出锁存器(0)的值将会被读取。
 - P11 至P17引脚用作输入端口时，引脚状态(1)将会被读取。
- <2> 将P10 位设为 1。
- <3> 将<2>的结果写入端口寄存器1 (P1)输出锁存器中。
 - 以8位为单元

第六章 时钟发生器

6.1 时钟发生器的功能

时钟发生器用于生成提供给 CPU 以及外围硬件的时钟。

以下各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

该电路通过将振荡器连接到 X1 和 X2 上，以产生 $f_x = 1$ 至 20 MHz 的时钟。

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

<2> 内部高速振荡器

该电路用于产生一个 $f_{RH} = 8$ MHz (典型值) 的时钟。复位释放后, CPU 使用该内部高速振荡时钟开始操作。通过执行 STOP 指令或使用主振荡模式寄存器 (RCM) 可以停止振荡。

外部主系统时钟 ($f_{EXCLK} = 1$ 至 20 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

该电路用于产生一个 $f_{RL} = 240$ kHz (典型值) 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, 可以通过使用内部振荡模式寄存器 (RCM) 来停止振荡。

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 (当 f_{RL} , $f_{RL}/2^7$, 或 $f_{RL}/2^9$ 被选择时)

备注	1. f_x :	X1 时钟振荡频率
	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

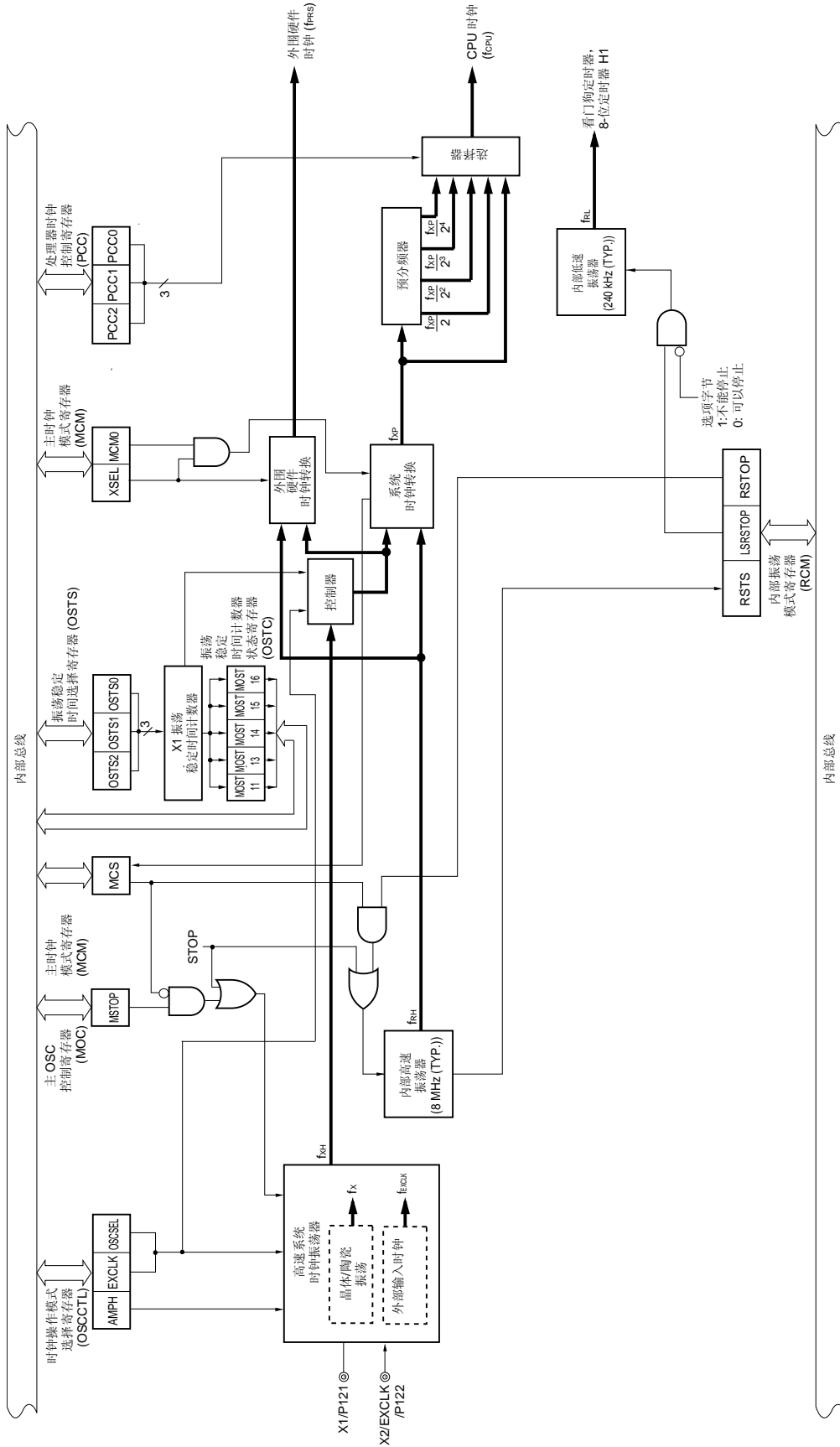
6.2 时钟发生器的配置

时钟发生器包含以下硬件。

表 6-1. 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 6-1. 时钟发生器的框图



备注	1. f_x :	X1 时钟振荡频率
	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} :	高速系统时钟振荡频率
	5. f_{XP} :	主系统时钟振荡频率
	6. f_{PRG} :	外围硬件时钟振荡频率
	7. f_{CPU} :	CPU 时钟振荡频率
	8. f_{RL} :	内部低速振荡时钟频率

6.3 控制时钟发生器的寄存器

以下七种寄存器用于控制时钟发生器。

- 时钟操作模式选择寄存器 (OSCCTL)
- 处理器时钟控制寄存器 (PCC)
- 内置振荡模式寄存器 (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时钟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。

复位信号将该寄存器设置为 00H。

图 6-2. 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	AMPH
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/陶瓷振荡器连接				
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		
AMPH	操作频率控制							
0	1 MHz ≤ f _{XH} ≤ 10 MHz							
1	10 MHz < f _{XH} ≤ 20 MHz							

- 注意事项**
1. 如果高速系统时钟振荡频率超过 10 MHz，请务必将 AMPH 设为 1。
 2. 在设置主时钟模式寄存器（MCM）之前设置 AMPH。
 3. 复位释放后需在设置外围功能之前对 AMPH 进行设置。复位释放后 AMPH 的值只能改变一次。当高速系统时钟（X1 振荡）被选作 CPU 时钟时，在 AMPH 设置为 1 后，CPU 时钟的供给将会停止 4.06 至 16.12 μs。当高速系统时钟（外部时钟输入）被选作 CPU 时钟时，在 AMPH 被设置为 1 后，CPU 时钟的供给则将会停止 160 个外部时钟所持续的时间。
 4. 如果在 AMPH = 1 时执行了 STOP 指令，那么在内部高速振荡时钟被选作 CPU 时钟时 CPU 时钟将会在 STOP 模式释放后停止 4.06 至 16.12 μs，或在高速系统时钟（外部时钟输入）被选作 CPU 时钟时停止 160 个外部时钟所持续的时间。当高速系统时钟（X1 振荡）被选作 CPU 时钟时，振荡稳定时间将会在 STOP 模式释放后开始计数。
 5. 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）。
 6. 确保将第 1 位至第 5 位清除为 0。

备注 f_{XH}: 高速系统时钟振荡频率

(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。

复位信号将 PCC 设为 01H。

图 6-3. 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
0	0	0	f _{XP}
0	0	1	f _{XP} /2 (默认)
0	1	0	f _{XP} /2 ²
0	1	1	f _{XP} /2 ³
1	0	0	f _{XP} /2 ⁴
除上述情况			禁止设置

注意事项 1. 确保将第 3 位至第 7 位清除为 0。

2. 当 PCC 的分频比率被设置时, 外围硬件时钟 (f_{PRS}) 将不会分频。

备注 f_{XP}: 主系统时钟振荡频率

μPD78F8024 和 μPD78F8025 产品中, 最快的指令可以在 2 个 CPU 时钟内执行完毕。因此, 表 6-2 显示了 CPU 时钟 (f_{cpu}) 与最短指令执行时间之间的关系。

表 6-2. CPU 时钟与最短指令执行时间之间的关系

CPU 时钟 (f _{cpu})	最短指令执行时间: 2/f _{cpu}		
	高速系统时钟 ^注		内部高速振荡时钟 ^注
	10 MHz 操作时	20 MHz 操作时	8 MHz (典型值) 操作时
f _{XP}	0.2 μs	0.1 μs	0.25 μs (典型值)
f _{XP} /2	0.4 μs	0.2 μs	0.5 μs (典型值)
f _{XP} /2 ²	0.8 μs	0.4 μs	1.0 μs (典型值)
f _{XP} /2 ³	1.6 μs	0.8 μs	2.0 μs (典型值)
f _{XP} /2 ⁴	3.2 μs	1.6 μs	4.0 μs (典型值)

注 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 6-6)

(3) 内置振荡模式寄存器 (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。

复位信号将该寄存器设置为 80H^{#1}。

图 6-4. 内部振荡模式寄存器 (RCM) 的格式

地址: FFA0H 复位后: 80H^{#1} R/W^{Note 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精确稳定
1	内部高速振荡器的稳定操作

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 是使用高速系统时钟 (MCS = 1) 在进行操作。另外，应在将 RSTOP 设为 1 前停止以内部高速振荡时钟运行的外围硬件。

(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。

复位信号将该寄存器设置为 80H。

图 6-5. 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 是使用内部高速系统时钟 (**MCS = 0**) 在进行操作。
另外, 应在将 **MSTOP** 设为 1 前停止以高速系统时钟运行的外围硬件。
 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. 外围硬件时钟停止时, 外围硬件不能工作。若要在外围硬件时钟停止后恢复外围硬件的操作, 则应对外围硬件进行初始化。

(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外围硬件时钟的时钟。

MCM 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号将该寄存器设置为 00H。

图 6-6. 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外围硬件的时钟的选择	
		主系统时钟 (f _{XP})	外围硬件时钟 (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		
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, f_{PRS} 以外的时钟可用于以下外围功能。
 - 看门狗定时器 (使用内部低速振荡时钟进行操作)
 - 当“f_{RL}”, “f_{RL}/2⁷”, 或 “f_{RL}/2⁹”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)
 - 外围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 (为 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 $\overline{\text{RESET}}$ 输入，POC，LVI，以及 WDT 进行复位），STOP 指令及 MSTOP（MOC 寄存器的第 7 位）= 1 都可以将 OSTC 清除为 00H。

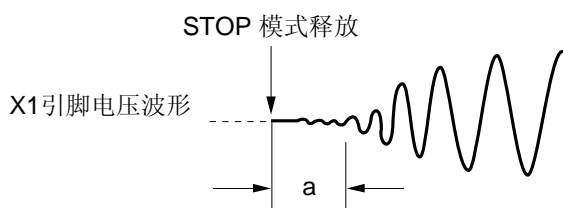
图 6-7. 振荡稳定时间计数状态寄存器 (OSTC) 的格式

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	振荡稳定时间状态		
					$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$	
1	0	0	0	0	$2^{11}/f_x \text{ min.}$	204.8 $\mu\text{s min.}$	102.4 $\mu\text{s min.}$
1	1	0	0	0	$2^{13}/f_x \text{ min.}$	819.2 $\mu\text{s min.}$	409.6 $\mu\text{s min.}$
1	1	1	0	0	$2^{14}/f_x \text{ min.}$	1.64 ms min.	819.2 $\mu\text{s min.}$
1	1	1	1	0	$2^{15}/f_x \text{ min.}$	3.27 ms min.	1.64 ms min.
1	1	1	1	1	$2^{16}/f_x \text{ min.}$	6.55 ms min.	3.27 ms min.

- 注意事项**
- 上述时间过后，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累加到通过 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，已进入 STOP 模式，在释放该模式时，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，仅将达到 OSTC 设定的振荡稳定时间时的状态被赋予 OSTC。
 - X1 时钟振荡稳定等待时间不包括时钟开始振荡前的时间（以下“a”）。



备注 f_x : X1 时钟振荡频率

(7) 振荡稳定时间选择寄存器 (OSTS)

该寄存器用于选择在 STOP 模式被释放后 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，通过使用 OSTS 等待一段时间再操作。

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 设置的时间核对。

OSTS 可以通过 8 位存储器操作指令来进行设置。

复位信号将 OSTS 设为 05H。

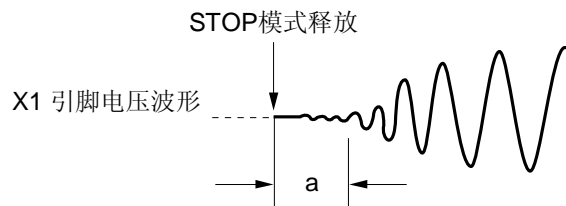
图 6-8. 振荡稳定时间选择寄存器 (OSTS) 的格式

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	振荡稳定时间选择	
			$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$
0	0	1	$2^{11}/f_x$	204.8 μs
0	1	0	$2^{13}/f_x$	819.2 μs
0	1	1	$2^{14}/f_x$	1.64 ms
1	0	0	$2^{15}/f_x$	3.27 ms
1	0	1	$2^{16}/f_x$	6.55 ms
除上述情况			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加到通过 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，已进入 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被赋予 OSTC。
 4. X1 时钟振荡稳定等待时间不包括时钟开始振荡前的时间（以下“a”）。



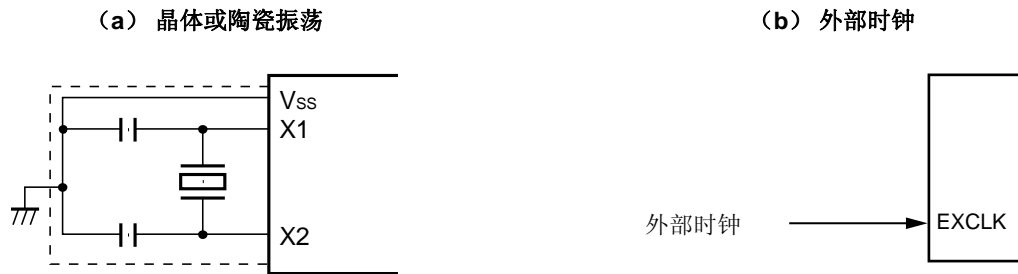
备注 f_x : X1 时钟振荡频率

6.4 系统时钟振荡器

6.4.1 X1 振荡器

X1 振荡器通过连接在 X1 和 X2 引脚上的晶体振荡器或陶瓷振荡器（1 至 20 MHz）来进行振荡。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 6-9 显示了 X1 振荡器的外部电路的示例。

图 6-9. X1 振荡器外部电路示例



注意事项 当使用 X1 振荡器时，图 6-9 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。

图 6-10 显示了错误的振荡器连接方式。

图 6-10. 错误的振荡器连接示例（1/2）

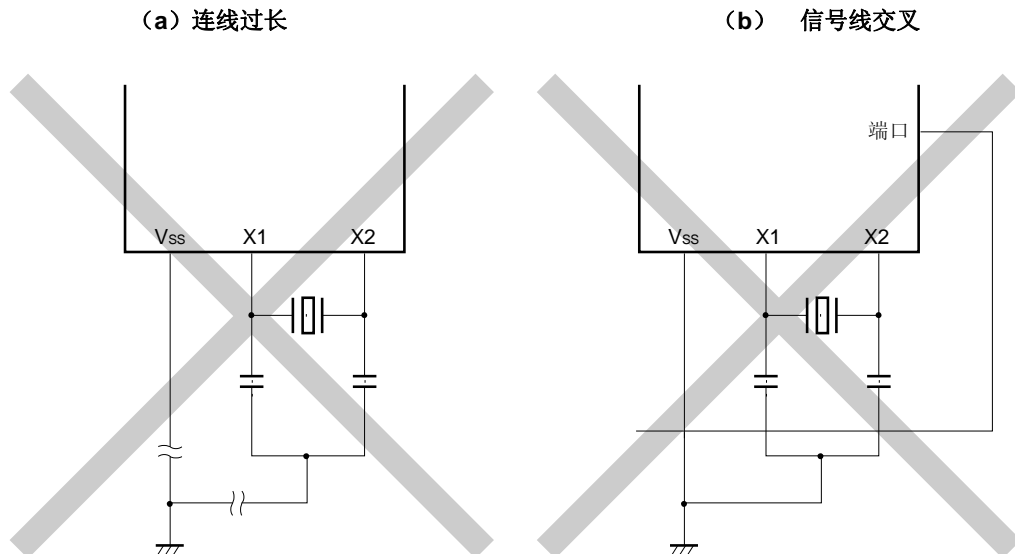
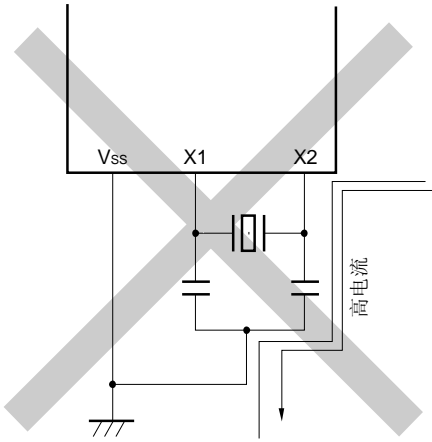
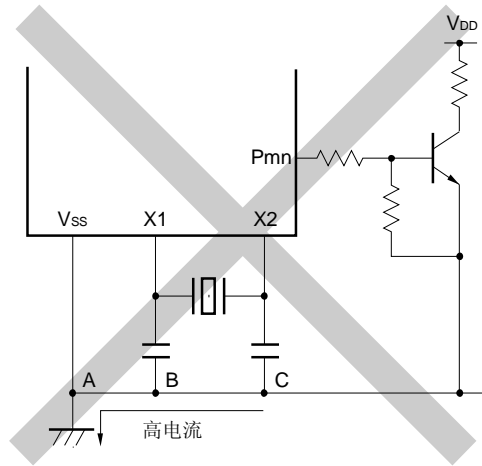


图 6-10. 错误的振荡器连接示例 (2/2)

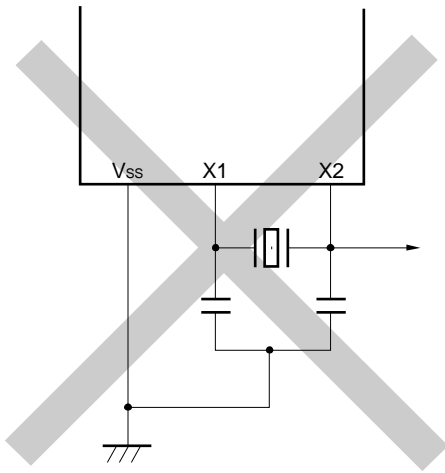
(c) 连线附近电流过大



(d) 大电流通过振荡器的接地线
(A, B, C 三点电压波动)



(e) 从振荡器中获取信号



6.4.2 内部高速振荡器

μ PD78F8024 和 78F8025 产品中含有一个内部高速振荡器。可以通过内部振荡模式寄存器（RCM）控制振荡。复位释放后，内部高速振荡器会自动开始振荡（8 MHz（典型值））。

6.4.3 内部低速振荡器

μ PD78F8024 和 78F8025 产品中含有一个内部低速振荡器。

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，可以通过内部振荡模式寄存器（RCM）控制振荡。

复位释放后，内部低速振荡器会自动开始振荡，且如果设置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（典型值））。

6.4.4 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。

6.5 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 6-1）。

- 主系统时钟 f_{XP}
 - 高速系统时钟 f_{XH}
 - X1 时钟 f_X
 - 外部主系统时钟 f_{EXCLK}
 - 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}
- 外围硬件时钟 f_{PRS}

在 μ PD78F8024 和 78F8025 产品中，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。

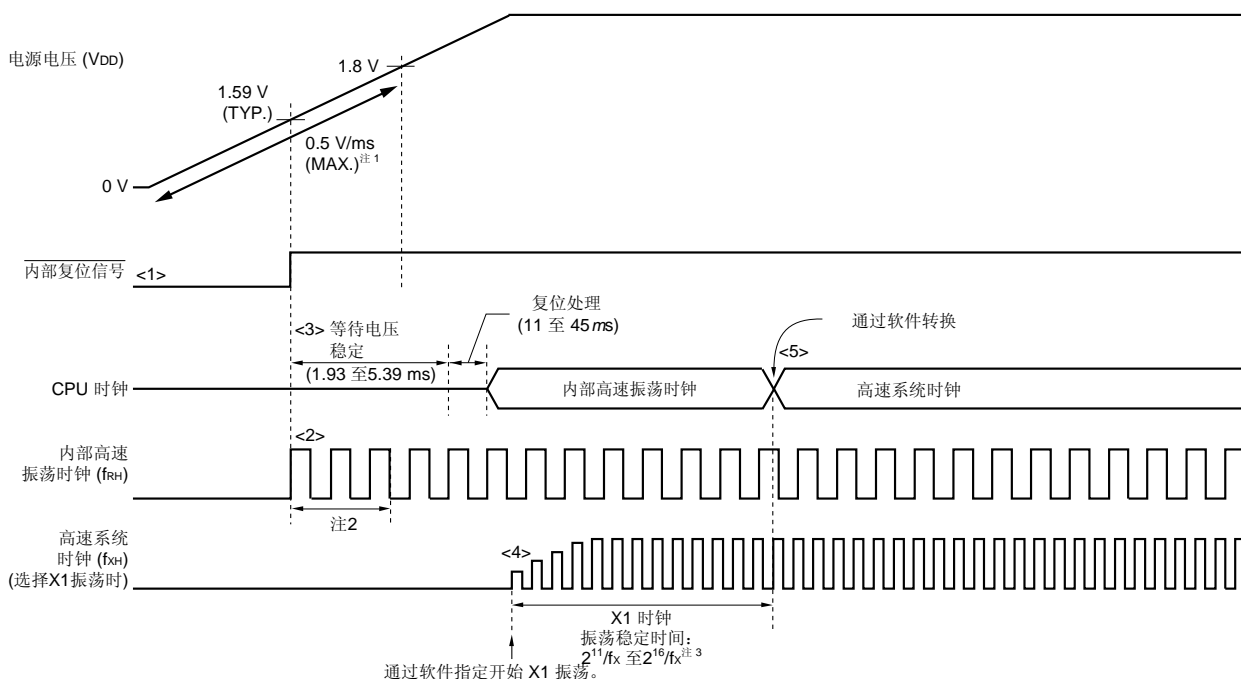
(1) 增强安全功能

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，因此复位释放后不能执行操作。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，系统可以通过执行最低操作安全关闭，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。电源开启时，时钟发生器的操作如图 6-11 中所示。

图 6-11. 电源打开时钟发生器的操作
(当设置为 1.59 V POC 模式时 (选项字节: POCMODE = 0))



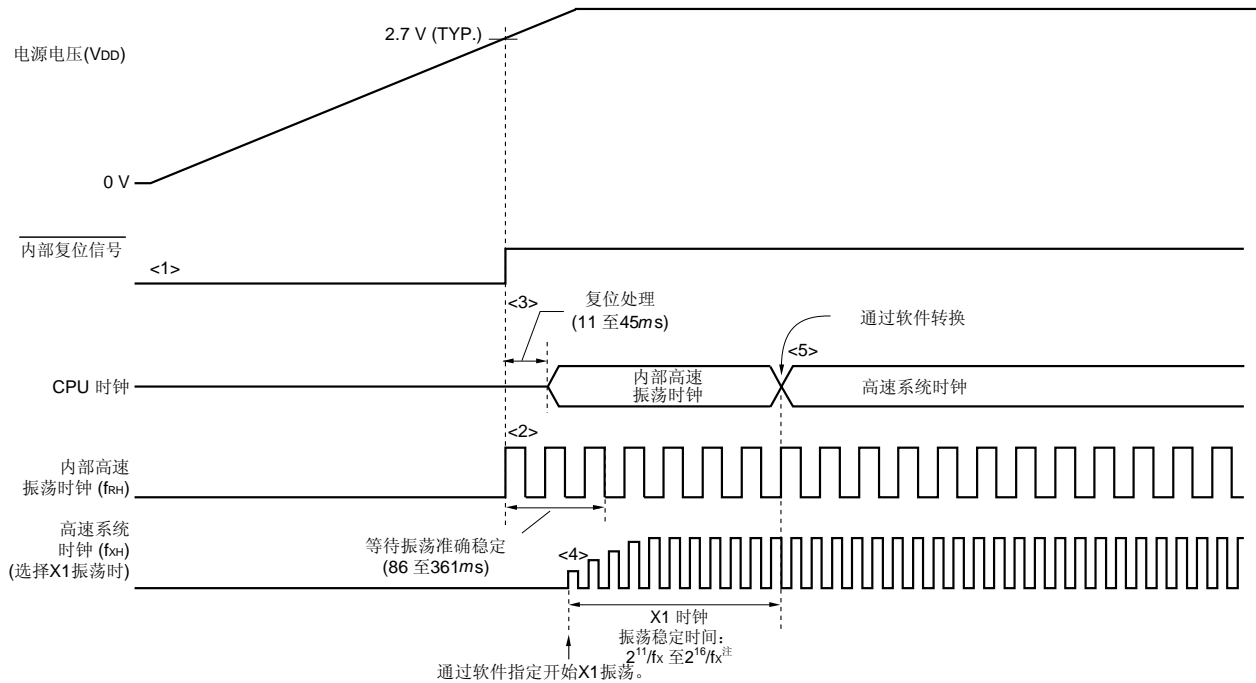
- <1> 电源打开时，通过上电清零（POC）电路产生内部复位信号。
- <2> 当电源电压超过 1.59 V（典型值）时，复位将会被释放且内部高速振荡器将自动开始振荡。
- <3> 当电源电压以 0.5 V/ms（MIN.）的斜率上升时，CPU 将会在复位释放后，在电源及稳压器的电压稳定后在内部高速振荡时钟中进行操作，然后执行复位处理。
- <4> 通过软件设置 X1 时钟的振荡的开始（参见 6.6.1 控制高速系统时钟示例中的（1））。
- <5> 在将 CPU 时钟转换至 X1 时钟时，应先等待时钟振荡稳定，然后再通过软件进行转换（参见 6.6.1 控制高速系统时钟示例中的（3））。

- 注
1. 如果电压从上电开始以小于 0.5 V/ms（最小值）的斜率上升至 1.8 V，那么从上电到电压达到 1.8 V 的过程中应对 RESET 引脚输入一个低电平，或通过使用选项字节（POCMODE = 1）来设置 2.7 V/1.59 V POC 模式（参见图 6-12）。通过 RESET 引脚复位释放后，如果在电压达到 1.8 V 前对 RESET 引脚输入了一个低电平，那么 CPU 将会按与<2>以及之后的图 6-11 中相同的时序来进行操作。
 2. 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。
 3. 在 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 引脚中输出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。可以通过执行 STOP 指令停止内部高速振荡时钟和高速系统时钟（参见 6.6.1 控制高速系统时钟示例中的（4）以及 6.6.2 控制内部高速振荡时钟示例中的（3））。

图 6-12. 电源打开时钟发生器的操作
（当设置为 2.7 V/1.59 V POC 模式时（选项字节：POCMODE = 1））



- <1> 电源打开时，通过上电清零（POC）电路产生内部复位信号。
- <2> 当电源电压超过 2.7 V（典型值）时，复位将会被释放且内部高速振荡器将自动开始振荡。
- <3> 在复位释放后以及复位处理进行时，CPU 将在内部高速振荡时钟上开始操作。
- <4> 通过软件设置 X1 时钟的振荡的开始（参见 6.6.1 控制高速系统时钟示例中的（1））。
- <5> 在将 CPU 时钟转换至 X1 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 6.6.1 控制高速系统时钟示例中的（3））。

注 在 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**
1. 电压达到 1.59 V（典型值）后将会需要 1.93 至 5.39 ms 的电压振荡稳定时间。如果电源电压在 1.93 ms 内从 1.59 V（典型值）上升至 2.7 V（典型值），那么在复位处理前将会自动产生 0 至 5.39 ms 的电源振荡稳定时间。
 2. 使用从 EXCLK 引脚中输出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，可以通过软件设置停止没有用作 CPU 时钟的时钟。可以通过执行 STOP 指令停止内部高速振荡时钟和高速系统时钟（参见 6.6.1 控制高速系统时钟示例中的（4）以及 6.6.2 控制内部高速振荡时钟示例中的（3））。

6.6 控制时钟

6.6.1 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 / 陶瓷振荡器通过 X1 和 X2 引脚来连接。
- 外部主系统时钟：输入外部时钟到 EXCLK 引脚。

未使用高速系统时钟时，X1/P121 和 X2/EXCLK/P122 引脚不能被用作 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）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 时钟以及外围硬件时钟时
- （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> 设置频率（OSCCTL 寄存器）

使用 AMPH，根据所使用的频率来设置片上振荡器的增益。

AMPH ^注	操作频率控制
0	$1 \text{ MHz} \leq f_{XH} \leq 10 \text{ MHz}$
1	$10 \text{ MHz} < f_{XH} \leq 20 \text{ MHz}$

注 复位释放后需在设置外围功能之前对 AMPH 进行设置。复位释放后 AMPH 的值只能改变一次。AMPH 被设置为 1 时，提供给 CPU 的时钟将会停止 4.06 至 16.12 μs 。

备注 f_{XH} ：高速系统时钟振荡频率

- <2> 设置 P121/X1 和 P122/X2/EXCLK 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 2 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/陶瓷振荡器连接	

- <3> 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。
- <4> 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. 在电源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 第二十四章 电气特性）。**

(2) 使用外部主系统时钟时的设置步骤示例

- <1> 设置频率（OSCCTL 寄存器）
使用 AMPH 来设置要使用的频率。

AMPH ^注	操作频率控制
0	$1 \text{ MHz} \leq f_{XH} \leq 10 \text{ MHz}$
1	$10 \text{ MHz} < f_{XH} \leq 20 \text{ MHz}$

注 复位释放后需在设置外围功能之前对 AMPH 进行设置。复位释放后 AMPH 的值只能改变一次。在 AMPH 被设为 1 后提供给 CPU 的时钟将会停止 160 个外部时钟所持续的时间。

备注 f_{XH} : 高速系统时钟振荡频率

- <2> 设置 P121/X1 和 P122/X2/EXCLK 引脚并选择操作模式（OSCCTL 寄存器）
当 EXCLK 和 OSCSEL 被设置为 1 时，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

- <3> 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. 在电源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 第二十四章 电气特性）。**

(3) 将高速系统时钟用作 CPU 时钟以及外围硬件时钟时的设置步骤示例**<1> 设置高速系统时钟振荡[Ⓐ]**

(参见 6.6.1 (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 在高速系统时钟已经运行时不需要<1>的设置。

<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, 高速系统时钟提供主系统时钟和外围硬件时钟。

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

注意事项 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外围硬件时钟。

<3> 选择 CPU 时钟的分频比率 (PCC 寄存器)

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
0	0	0	f _{XP}
0	0	1	f _{XP} /2 (默认)
0	1	0	f _{XP} /2 ²
0	1	1	f _{XP} /2 ³
1	0	0	f _{XP} /2 ⁴
除上述情况			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> 通过设置停止外围硬件**

停止不能在 STOP 模式中使用的外围硬件 (关于不能在 STOP 模式中使用的外围硬件的详细信息, 请参见 第十七章 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 OSTS 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 停止 X1 振荡（禁止外部时钟输入）

<1> 确认 CPU 时钟的状态（MCM 寄存器）

通过 MCS 来确认 CPU 以除高速系统时钟以外的时钟运行。

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外围硬件。

6.6.2 控制内部高速振荡时钟的示例

以下详细说明了下述情况中设置步骤的示例。

- (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外围硬件时钟时
- (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 时钟和外围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，或将内部高速振荡时钟或高速系统时钟用作外围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 6.6.2 (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)。

• 振荡高速系统时钟^注

(将高速系统时钟用作外围硬件时钟时需要该设置。参见 6.6.1 (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 在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> 选择主系统时钟以及外围硬件时钟的时钟源 (MCM 寄存器)

使用 XSEL 和 MCM0 来设置主系统时钟以及外围硬件时钟

XSEL	MCM0	主系统时钟以及提供给外围硬件的时钟的选择	
		主系统时钟 (f _{XP})	外围硬件时钟 (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		内部高速振荡时钟 (f _{RH})
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)

、选择 CPU 时钟的分频比率，设置 PCC0，PCC1，以及 PCC2。

PCC2	PCC1	PCC0	CPU 时钟 (f _{CPU}) 选择
0	0	0	f _{XP}
0	0	1	f _{XP} /2 (默认)
0	1	0	f _{XP} /2 ²
0	1	1	f _{XP} /2 ³
1	0	0	f _{XP} /2 ⁴
除上述情况			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> 外围硬件的设置**

停止不能在 STOP 模式中使用的 外围硬件 (关于不能在 STOP 模式中使用的 外围硬件的详细信息，请参见 第十七章 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。若要在释放 STOP 模式后立即操作 CPU，则应将 MCM0 设为 0，将 CPU 时钟转换为内部高速振荡时钟，然后检查 RSTS 为 1。

<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 (MCM 寄存器)

通过 MCS 来确认 CPU 以高速系统时钟执行操作。

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 设置为 1 时，请务必确认 MCS = 1。此外，还应停止以高速振荡时钟运行的外围硬件。

6.6.3 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。

只有以下外围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 f_{RL} 被选作计数时钟)

此外，可以通过选项字节来选择以下操作模式。

- 不能停止内部低速振荡器
- 可以通过软件停止内部低速振荡器

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 (典型值))。

(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“不能停止内部低速振荡器”的模式，那么将不能控制内部低速振荡时钟的振荡。

6.6.4 提供给CPU以及外围硬件的时钟

下表显示了提供给 CPU 及外围硬件的时钟间的关系以及寄存器的设置。

表 6-3. 提供给 CPU 及外围硬件的时钟，以及寄存器设置

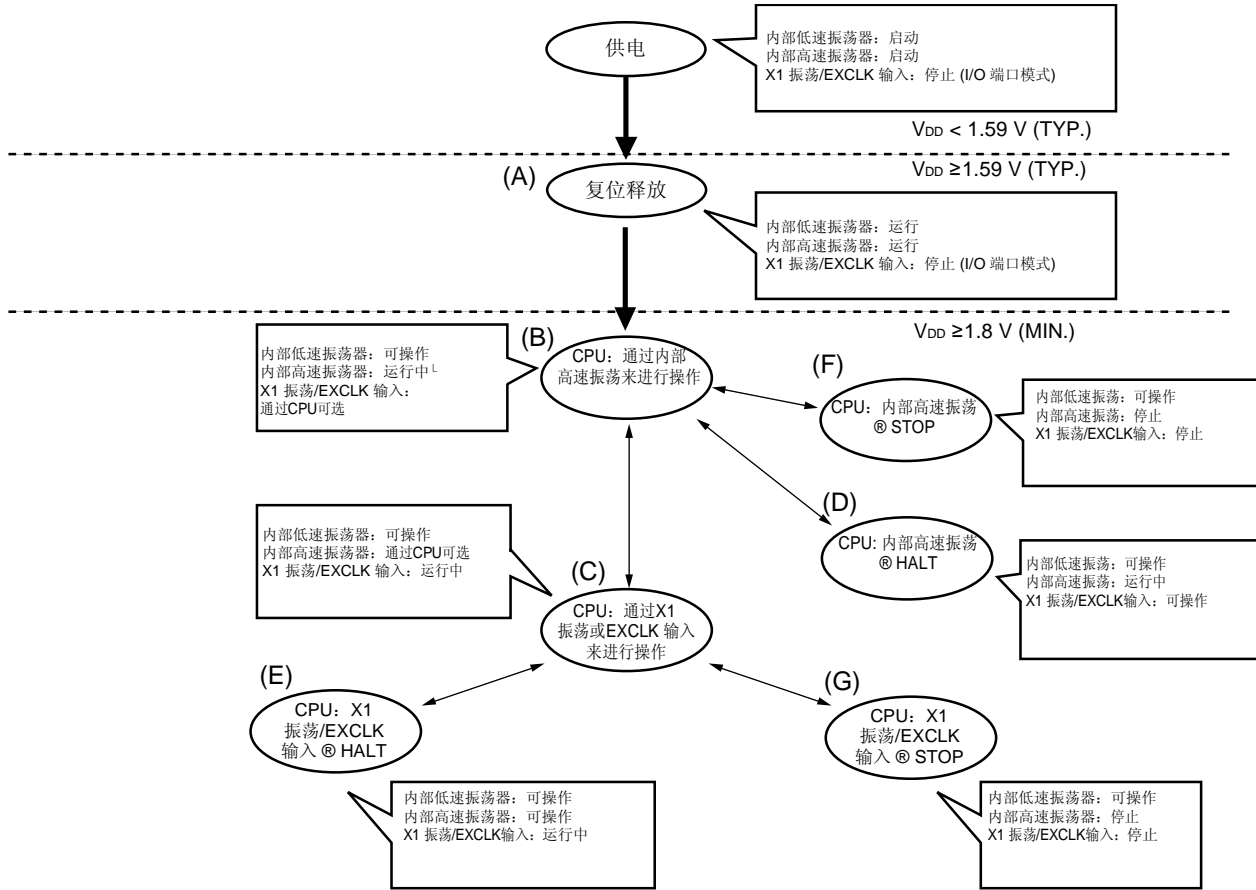
提供的时钟		XSEL	MCM0	EXCLK
提供给 CPU 的时钟	提供给外围硬件的时钟			
内部高速振荡时钟		0	×	×
内部高速振荡时钟	X1 时钟	1	0	0
	外部主系统时钟	1	0	1
X1 时钟		1	1	0
外部主系统时钟		1	1	1

- 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

6.6.5 CPU时钟状态迁移图

图 6-13 显示了该产品的 CPU 时钟状态迁移图。

图 6-13. CPU 时钟状态迁移图
(设置 1.59 V POC 模式时 (选项字节: POCMODE = 0))



备注 在 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1) 中, 当电源电压超过 2.7 V (典型值) 时 CPU 时钟状态将转变为上图中的 (A), 而在复位处理 (11 至 45 μs) 后则将转变为上图中的 (B)。

表 6-4 显示了 CPU 时钟的转移以及设置 SFR 寄存器的示例。

表 6-4. CPU 时钟转换以及 SFR 寄存器设置示例 (1/3)

(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)	复位释放后不必设置 SFR 寄存器 (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 状态转换	AMPH	EXCLK	OSCSEL	MSTOP	OSTC 寄存器	XSEL	MCM0
(A) → (B) → (C) (X1 时钟: $1 \text{ MHz} \leq f_{XH} \leq 10 \text{ MHz}$)	0	0	1	0	必须检测	1	1
(A) → (B) → (C) (外部主时钟: $1 \text{ MHz} \leq f_{XH} \leq 10 \text{ MHz}$)	0	1	1	0	不必检测	1	1
(A) → (B) → (C) (X1 时钟: $10 \text{ MHz} < f_{XH} \leq 20 \text{ MHz}$)	1	0	1	0	必须检测	1	1
(A) → (B) → (C) (外部主时钟: $10 \text{ MHz} < f_{XH} \leq 20 \text{ MHz}$)	1	1	1	0	不必检测	1	1

注意事项 在电源电压达到所使用的时钟的操作电压时, 设置时钟 (参见 第二十四章 电气特性)。

备注 1. 表 6-4 中的 (A) 至 (G) 对应于图 6-13 中的 (A) 至 (G)。

2. EXCLK, OSCSEL, AMPH:

时钟操作模式选择寄存器 (OSCCTL) 的第 7 位, 第 6 位以及 0 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

表 6-4. CPU 时钟转换以及 SFR 寄存器设置示例 (2/3)

(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	AMPH [※]	EXCLK	OSCSEL	MSTOP	OSTC 寄存器	XSEL [※]	MCM0
(B) → (C) (X1 时钟: $1 \text{ MHz} \leq f_{XH} \leq 10 \text{ MHz}$)	0	0	1	0	必须检测	1	1
(B) → (C) (外部主时钟: $1 \text{ MHz} \leq f_{XH} \leq 10 \text{ MHz}$)	0	1	1	0	不必检测	1	1
(B) → (C) (X1 时钟: $10 \text{ MHz} < f_{XH} \leq 20 \text{ MHz}$)	1	0	1	0	必须检测	1	1
(B) → (C) (外部主时钟: $10 \text{ MHz} < f_{XH} \leq 20 \text{ MHz}$)	1	1	1	0	不必检测	1	1

在这些寄存器已经被设置时不需要

CPU 使用高速系统时钟操作时不需要

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 在电源电压达到所使用的时钟的操作电压时对时钟进行设置（参见第二十四章 电气特性）。

(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
(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

备注 1. 表 6-4 中的 (A) 至 (G) 对应于图 6-13 中的 (A) 至 (G)。

2. EXCLK, OSCSEL, AMPH:

时钟操作模式选择寄存器 (OSCCTL) 的第 7 位, 第 6 位以及 0 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

RSTS, RSTOP: 内部振荡模式寄存器 (RCM) 的第 7 位以及 0 位

表 6-4. CPU 时钟转换以及 SFR 寄存器设置示例 (3/3)

- (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 模式中进行操作的功能	执行 STOP 指令

备注 表 6-4 中的 (A) 至 (G) 对应于图 6-13 中的 (A) 至 (G)。

6.6.6 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 6-5. 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)。 • AMPH 被设置为 1 后, 提供给 CPU 的时钟将会停止 4.06 至 16.12 μ s。
	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)。 • 在 AMPH 被设为 1 后提供给 CPU 的时钟将会停止 EXCLK 引脚中 160 个外部时钟所持续的时间。
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)。

6.6.7 主系统时钟转换所需的时间

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 6-6）。

表 6-6. CPU 时钟和主系统时钟周期分频系数间转换所需的时间

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	1	0	1	0	0	1	1	1	0	0
0	0	0	8 个时钟			16 个时钟			16 个时钟			16 个时钟			16 个时钟		
0	0	1				8 个时钟			8 个时钟			8 个时钟			8 个时钟		
0	1	0	4 个时钟			4 个时钟			4 个时钟			4 个时钟			4 个时钟		
0	1	1	2 个时钟			2 个时钟			2 个时钟			2 个时钟			2 个时钟		
1	0	0	1 个时钟			1 个时钟			1 个时钟			1 个时钟			1 个时钟		

备注 表 6-6 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；操作将以预转换时钟持续几个时钟（参见表 6-7）。

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）来确认。

表 6-7. 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} 时钟	
1	1 + 2f _{XH} /f _{RH} 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 6-7 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 6-7 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} = 8 MHz，f_{XH} = 10 MHz 时振荡）

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

6.6.8 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 6-8 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 以高速系统时钟执行操作)	RSTOP = 1
X1 时钟	MCS = 0 (CPU 以高速振荡时钟执行操作)	MSTOP = 1
外部主系统时钟		

6.6.9 外围硬件以及源时钟

以下列出了 μ PD78F8024 和 78F8025 产品中所包含的外围硬件及源时钟。

表 6-9. 外围硬件和源时钟

源时钟		外围硬件时钟 (fPRS)	内部低速振荡时钟 (fRL)	TM50 输出	外围硬件引脚中的外部时钟
外围硬件					
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 (TI150 引脚)
	51	Y	N	N	Y (TI151 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
A/D 转换器		Y	N	N	N
串行接口	UART0	Y	N	Y	N
	UART6	Y	N	Y	N
	CSI10	Y	N	N	Y (SCK10 引脚)
	IIC0	Y	N	N	Y (SCL0 引脚)

备注 Y: 可选, N: 不可选

7.1 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产生一个中断请求。

(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 可以输出任意脉冲宽度的单脉冲。

(5) PPG 输出

16 位定时器 / 事件计数器 00 可以输出一个方形波，该方形波的频率和输出脉冲宽度可以任意设置。

(6) 脉冲宽度测量

16 位定时器 / 事件计数器 00 可以测量外部输入信号的脉冲宽度。

7.2 16 位定时器 / 事件计数器 00 的配置

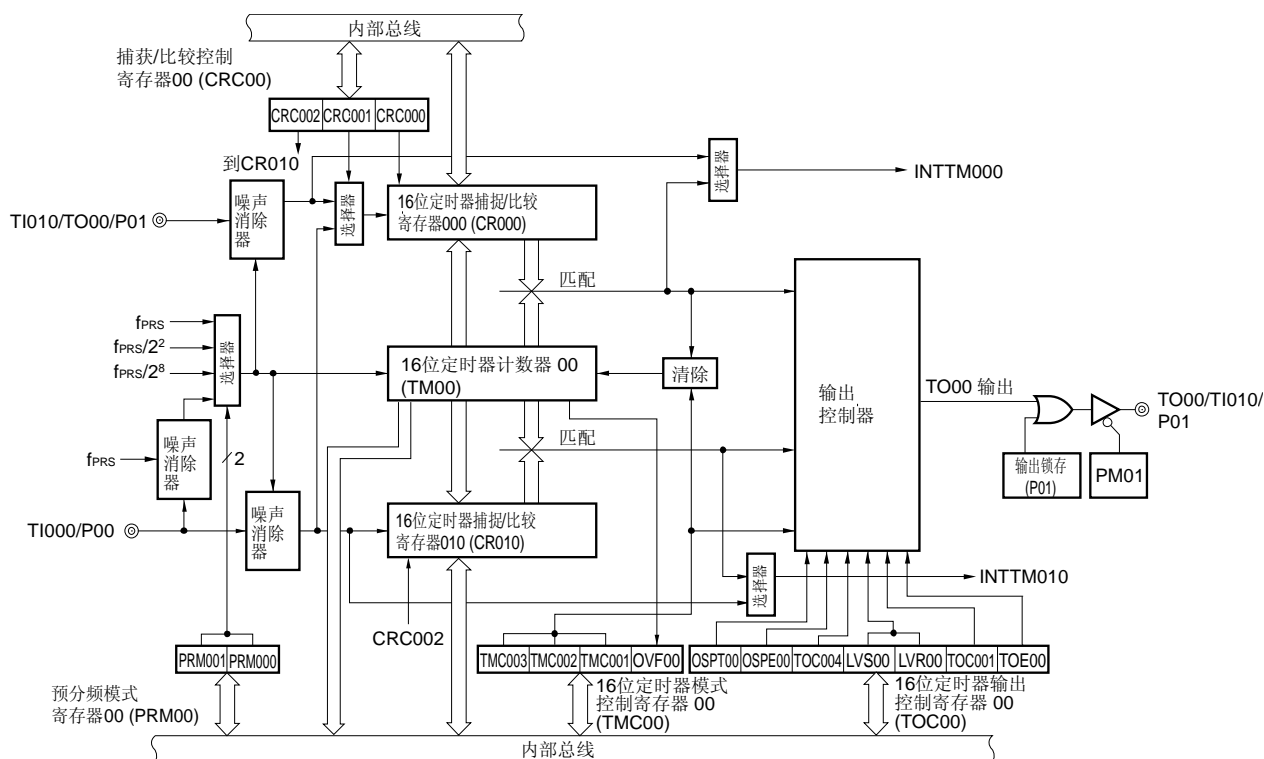
16 位定时器 / 事件计数器 00 包含有以下硬件。

表 7-1. 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 位定时器捕捉 / 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 位定时器捕捉 / 比较控制寄存器 00 (CRC00) 16 位定时器输出控制寄存器 00 (TOC00) 预分频模式寄存器 00 (PRM00) 端口模式寄存器 0 (PM0) 端口寄存器 0 (P0)

图 7-1 显示了框图。

图 7-1. 16 位定时器 / 事件计数器 00 的框图



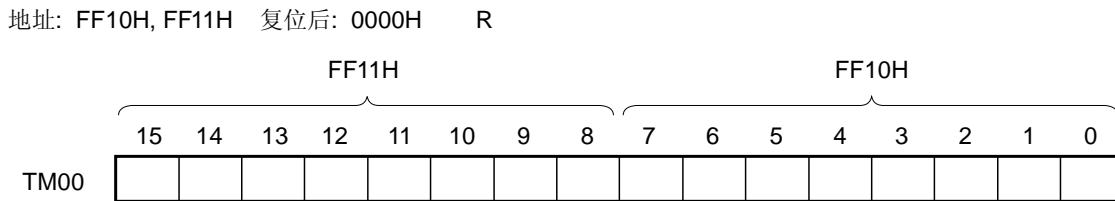
注意事项 1. TI010 的有效沿与定时器输出 (TO00) 不能同时用于 P01 引脚。选择其中一个功能。

- 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 (TMC003 和 TMC002) 的清零与捕捉触发的输入冲突, 那么捕捉数据将不确定。
- 3.** 若要将模式从捕捉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, 否则捕捉的值将不能保持在 CR000 中。如果模式已经改变为比较模式, 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。

图 7-2. 16 位定时计数器 00 (TM00) 的格式



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 获得。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。

- 复位信号的产生时
- 如果 TMC003 和 TMC002 被清除为 00
- 如果在 TI000 引脚输入有效沿产生清零&启动模式中, 在有效沿被输入到 TI000 引脚时
- 如果 TM00 和 CR000 在发生清零&启动的模式中匹配时
- 在单脉冲输出模式中或有效沿被输入到 T1000 引脚时, OSPT00 被设置为 1。

注意事项 即使 TM00 被读取, 也不能通过 CR010 来获取值。

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

CR000 和 CR010 是 16 位寄存器, 通过 CRC00 选择作为捕捉功能或比较功能使用。

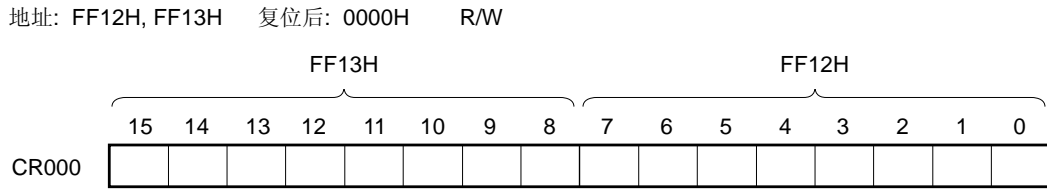
定时器停止时 (TMC003 和 TMC002 = 00) 改变 CR000 的值。

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。关于详细信息, 参见 7.5.1 TM00 操作期间重新写入 CR010。

这些寄存器可以以 16 位为单位进行读取或写入。

复位信号的产生将这些寄存器设置为 0000H。

图 7-3. 16 位定时器捕捉 / 比较寄存器 000 (CR000) 的格式



(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, 则将产生一个中断请求信号 (INTTM000)。值会一直被保存直到 CR000 被重新写入。

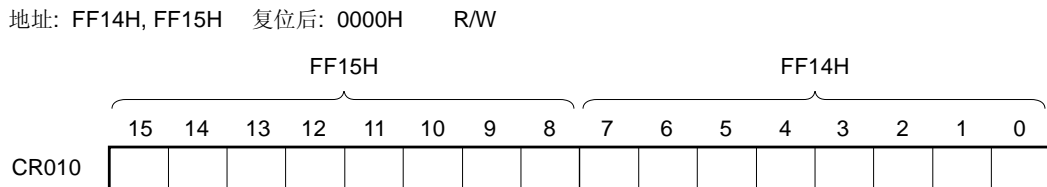
注意事项 当 CR000 被设置为比较模式时, 即使捕捉触发被输入到 CR000 中, 它也不能执行捕捉操作。

(ii) 当 CR000 被用作捕捉寄存器时

TM00 的计数值会在输入捕捉触发时被捕捉至 CR000 中。

用作捕捉触发时, 可以通过使用 CRC00 或 PRM00 来选择与 TI000 引脚相位相反的脉冲沿或 TI010 引脚的有效沿。

图 7-4. 16 位定时器捕捉 / 比较寄存器 010 (CR010) 的格式



(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, 则将产生一个中断请求信号 (INTTM010)。

注意事项 当 CR010 被设置为比较模式时, 即使输入捕捉触发, 它也不能执行捕捉操作。

(ii) 当 CR010 被用作捕捉寄存器时

TM00 的计数值会在输入捕捉触发时被捕捉至 CR010 中。

可以将 TI000 引脚的有效沿选作捕捉触发。TI000 引脚的有效沿可以通过 PRM00 来设置。

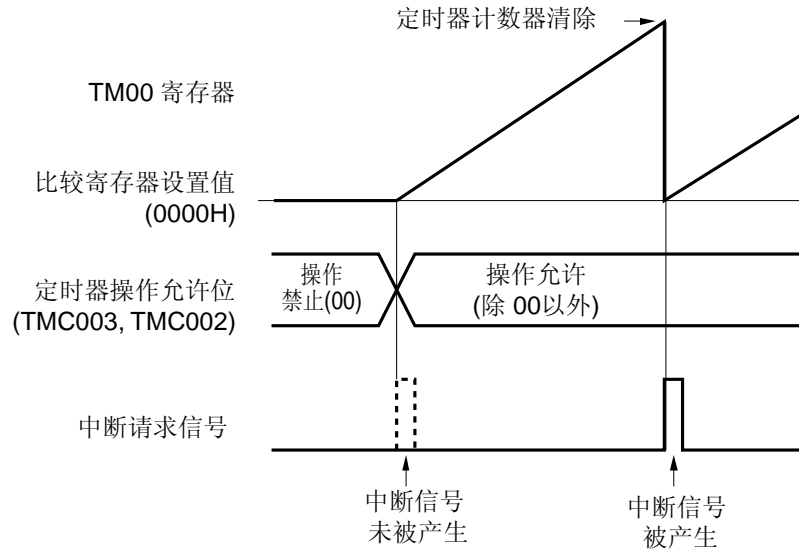
(iii) 当 CR000 或 CR010 被用作比较寄存器时的设置范围

当 CR000 或 CR010 被用作比较寄存器时，按照下面设置。

操作	CR000 寄存器设置范围	CR010 寄存器设置范围
作为间隔定时器操作	0000H < N ≤ FFFFH	0000H [#] ≤ M ≤ FFFFH
作为方波输出操作		一般，不使用这个设置。屏蔽匹配中断信号 (INTTM010)。
作为外部事件计数器操作		
通过TI000引脚有效沿输入进入的清零&启动模式中的操作	0000H [#] ≤ N ≤ FFFFH	0000H [#] ≤ M ≤ FFFFH
作为自由运行定时器操作		
作为PPG输出操作	M < N ≤ FFFFH	0000H [#] ≤ M < N
作为单脉冲输出操作	0000H [#] ≤ N ≤ FFFFH (N ≠ M)	0000H [#] ≤ M ≤ FFFFH (M ≠ N)



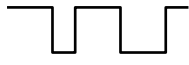



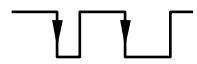
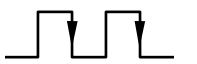
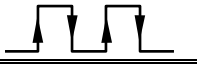




注 当设置 0000H 时，定时器操作后的匹配中断不会发生，定时器输出不改变并且第一个匹配时序如下所示。当定时器计数器 (TM00 寄存器) 从 0000H 改变为 0001H 时，匹配中断发生。

- 当定时器计数器由于溢出被清除时
- 当定时器计数器由于 TI000 引脚的有效沿被清除时 (当通过 TI000 引脚有效沿输入进入清零&启动模式时)
- 当定时器计数器由于比较匹配被清除时 (当通过 TM00 和 CR000 间的匹配进入清零&启动模式时 (CR000 ≠ 0000H, CR010 = 0000H))



- 备注
1. N: CR000 寄存器设置值, M: CR010 检测器的设置值
 2. 关于 TMC003 和 TMC002 的详细信息, 参见 7.3 (1) 16 位定时器模式控制寄存器 00 (TMC00)。

表 7-2. CR000 和 CR010 的捕捉操作

外部输入信号 捕捉操作	TI000 引脚输入 		TI010 引脚输入 	
CR000 的捕捉操作	CRC001 = 1 TI000 引脚输入 (反相) 	ES001 和 ES000 的设定值 要捕捉的边沿的位置	CRC001 位 = 0 TI010 引脚输入 	ES101 和 ES100 的设定值 要捕捉的边沿的位置
		01: 上升 		01: 上升 
00: 下降 		00: 下降 		
		11: 双边沿 (不能捕捉)		11: 双边沿 
	中断信号	即使捕捉了计数值, 也不会产生INTTM000信号。	中断信号	每次捕捉到计数值时都会产生INTTM000信号。
CR010 的捕捉操作	TI000 引脚输入 ^注 	ES001 和 ES000 的设定值 要捕捉的边沿的位置		
		01: 上升 		
00: 下降 				
		11: 双边沿 		
	中断信号	每次捕捉到值时都会产生INTTM010 信号。		

注 CR010 的捕捉操作不会受 CRC001 位设置的影响。

注意事项 若要使用与输入到 T1000 引脚中的信号相反的相位将 TM00 寄存器的计数值捕捉至 CR000 寄存器, 那么在值被捕捉后将不能产生中断请求信号 (INTTM000)。如果在该操作过程中在 TI010 引脚上检测到有效沿, 那么将不会执行捕捉操作, 但 INTTM000 信号将产生一个外部中断信号。为了不使用外部中断, 应屏蔽 INTTM000 信号。

备注 CRC001: 参见 7.3 (2) 捕捉 / 比较控制寄存器 00 (CRC00)。
ES101, ES100, ES001, ES000: 参见 7.3 (4) 预分频模式寄存器 00 (PRM00)。

7.3 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)
- 捕捉 / 比较控制寄存器 00 (CRC00)
- 16 位定时器输出控制寄存器 00 (TOC00)
- 预分频模式寄存器 00 (PRM00)
- 端口模式寄存器 0 (PM0)
- 端口寄存器 0 (P0)

(1) 16 位定时器模式控制寄存器 00 (TMC00)

TMC00 是一个 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及溢出检测。

禁止在操作（当 TMC003 和 TMC002 不等于 00）期间重新写入 TMC00。然而，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。

复位信号的产生将该 TMC00 设置为 00H。

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

图 7-5. 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引脚有效沿输入进入的清零&启动模式 ^注
1	1	在TM00和CR000匹配时进入的清零&启动模式 ^注

TMC001	反转定时器输出的条件 (TO00)
0	<ul style="list-style-type: none"> • TM00和CR000匹配或TM00和CR010匹配
1	<ul style="list-style-type: none"> • TM00和CR000匹配或TM00和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零&启动模式，以及在TM00和CR000匹配时进入的清零&启动模式）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 通过预分频模式寄存器 00 (PRM00) 的第 5 位和第 4 位 (ES001, ES000) 设置 TI000 引脚的有效沿。

(2) 捕捉 / 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。

复位信号的产生将该 CRC00 设置为 00H。

图 7-6. 捕捉 / 比较控制寄存器 00 (CRC00) 的格式

地址: FFBCH 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	作为捕捉寄存器进行操作

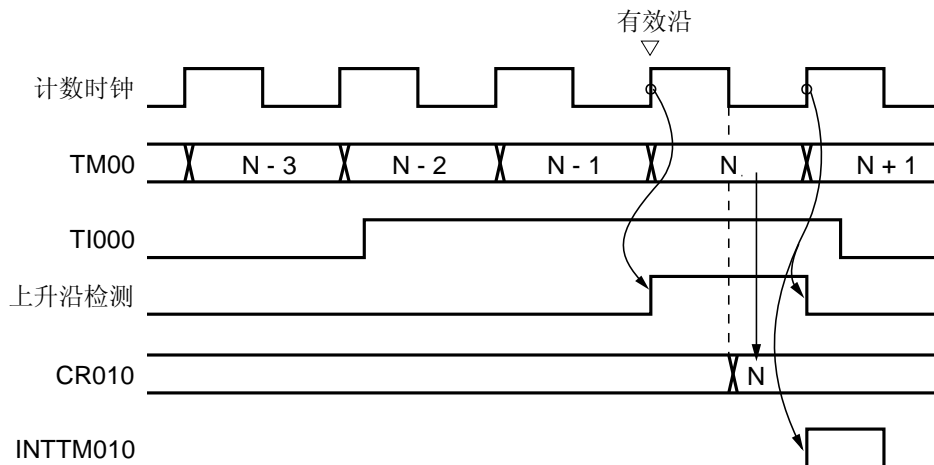
CRC001	CR000 捕捉触发选择
0	在 TI010 引脚的有效沿上进行捕捉
1	在 TI000 引脚的有效沿反向时进行捕捉 ^注
通过 PRM00 对 TI010 和 TI000 的有效沿进行设置。 如果 ES001 和 ES000 在 CRC001 为 1 时被设为 11 (双沿), 那么 TI000 引脚的有效沿将不能被检测。	

CRC000	CR000 操作模式选择
0	作为比较寄存器进行操作
1	作为捕捉寄存器进行操作
如果 TMC003 和 TMC002 被设为 11 (在 TM00 和 CR000 匹配时进入的清零&启动模式), 请务必将 CRC000 设为 0。	

注 在 TI010 引脚上检测到有效沿时, 将不会执行捕捉操作, 但将产生 INTTM000 信号作为外部中断信号。

注意事项 要确保捕捉操作正确执行, 捕捉触发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 7-7. CR010 捕捉操作示例（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 的输出。

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

然而，在定时器运行过程中 TOC004 可以按重新写入 CR010 的方法进行重新写入（参见 7.5.1 **TM00 操作期间重新写入 CR010**）。

TOC00 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号的产生将该 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 7-8. 16 位定时器输出控制寄存器 00 (TOC00) 的格式

地址: FFBDAH 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
0	-
1	单脉冲输出
该位在读取时的值始终为“0”。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 引脚有效沿输入进入的清零&启动模式中正确进行。单脉冲不能在 TM00 与 CR000 匹配的情况下进入的清零&启动模式中输出。	

TOC004	在 CR010 和 TM00 匹配时的 TO00 输出控制
0	禁止反转操作
1	允许反转操作
即使当 TOC004 = 0 时也会产生中断信号 (INTTM010)。	

LVS00	LVR00	TO00 输出状态的设置
0	0	没有变化
0	1	TO00 输出的初始值为低电平 (TO00 输出被清除为 0)。
1	0	TO00 输出的初始值为高电平 (TO00 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 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 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 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，参见 7.5.2 设置 LVS00 和 LVR00。 实际 TO00 / TI010 / P01 引脚的输出由 PM01 和 P01 以及 TO00 输出确定。 		

TOC001	在 CR000 和 TM00 匹配时的 TO00 输出控制
0	禁止反转操作
1	允许反转操作
即使当 TOC001 = 0 时也会产生中断信号 (INTTM000)。	

TOE00	TO00 输出控制
0	禁止输出 (TO00 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）期间重新写入 PRM00。

PRM00 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号的产生将该 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 引脚有效沿进入的清零&启动模式
 - 将 TI000 引脚设置为捕捉触发
 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然后再次允许时将不能检测到上升沿。
 3. TI010 的有效沿与定时器输出 (TO00) 不能同时用于 P01 引脚。选择其中任一功能。

图 7-9. 预分频模式寄存器 00 (PRM00) 的格式

地址: FFBBH 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 引脚有效沿检测				
0	0	下降沿				
0	1	上升沿				
1	0	禁止设置				
1	1	双边沿				

ES001	ES000	TI000 引脚有效沿检测				
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 ^{注1}				
			f _{PRS} = 4 MHz	f _{PRS} = 8 MHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz
0	0	f _{PRS} ^{注2}	4 MHz	8 MHz	10 MHz	20 MHz ^{注3}
0	1	f _{PRS} /2 ²	1 MHz	2 MHz	2.5 MHz	5 MHz
1	0	f _{PRS} /28	15.63 kHz	31.25 kHz	39.06 kHz	78.12 kHz
1	1	TI000 有效沿 ^{注4,5}				

- 注
1. 如果外围硬件时钟 (f_{PRS}) 工作于高速系统时钟 (f_{XH}) (XSEL = 1), f_{PRS} 工作频率根据供电电压改变。
 - V_{DD} = 2.7 到 5.5 V: f_{PRS} ≤ 20 MHz
 - V_{DD} = 1.8 到 2.7 V: f_{PRS} ≤ 5 MHz
 2. 如果外围硬件时钟 (f_{PRS}) 工作于内部高速振荡时钟 (f_{RH}) (XSEL = 0), 当 1.8 V ≤ V_{DD} < 2.7 V 时, 禁止设置 PRM001 = PRM000 = 0 (计数时钟: f_{PRS})。
 3. 只有在 4.0 V ≤ V_{DD} ≤ 5.5 V 时可以设置。
 4. 从 TI000 引脚输入的外部时钟需要一个比两个外围硬件时钟 (f_{PRS}) 周期长的脉冲。
 5. 在 STOP 模式下, 不要启动使用 TI000 引脚输入的外部时钟的定时器操作。

备注 f_{PRS}: 外围硬件时钟频率

(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。

为定时器输出使用 P01 / TO00 / TI010 引脚时，应将 PM01 以及 P01 输出锁存设置为 0。

为定时器输入使用 P00 / TI000 和 P01 / TO00 / TI010 引脚时，应将 PM00 和 PM01 设置为 1。此时，P00 和 P01 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。

复位信号的产生将 PM0 设置为 FFH。

图 7-10. 端口模式寄存器 0 (PM0) 的格式

地址: FF20H 复位后: FFH RW

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

PM0n	P0n 引脚输入/输出模式选择 (n = 0, 1)
0	输出模式 (输出缓冲器开)
1	输入模式 (输出缓冲器关)

7.4 16 位定时器 / 事件计数器 00 的操作

7.4.1 间隔定时器操作

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) 设置为 11 (在 TM00 与 CR000 匹配时进入的清零&启动模式), 那么计数操作将与计数时钟同步开始。

当 TM00 的值与 CR000 的值匹配时, TM00 将被清除为 0000H, 并产生一个匹配中断信号 (INTTM000)。该 INTTM000 信号允许 TM00 作为间隔定时器来进行操作。

- 备注**
1. 关于输入 / 输出引脚设置的详细信息, 参见 7.3 (5) 端口模式寄存器 0 (PM0)。
 2. 关于允许 INTTM000 中断的方法, 参见第十六章 中断功能。

图 7-11. 间隔定时器操作的框图

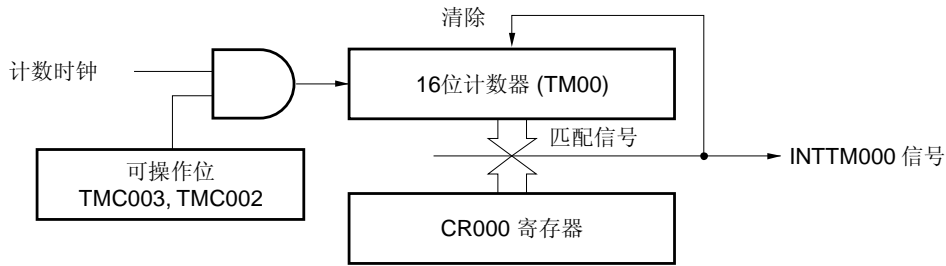


图 7-12. 间隔定时器操作的基本时序示例

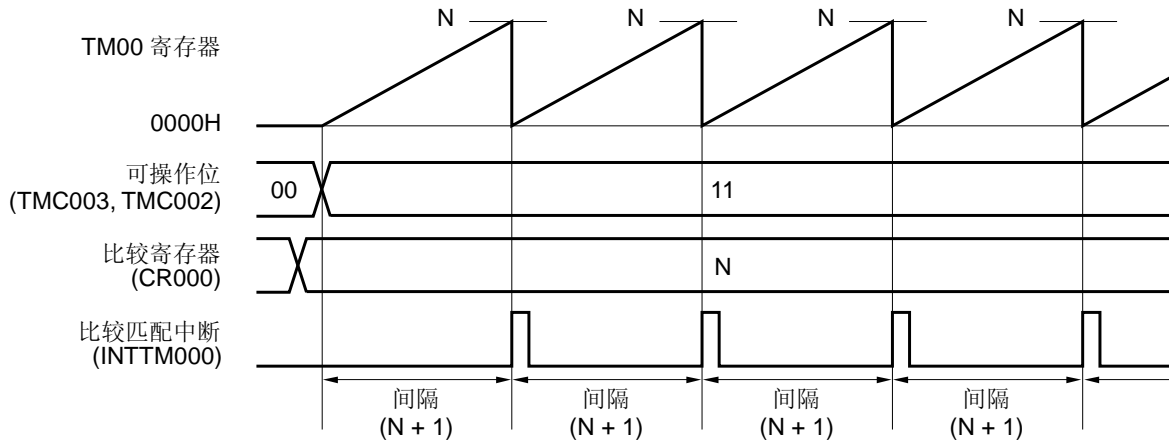


图 7-13. 间隔定时器操作的寄存器设置示例

(a) 16 位定时器模式控制寄存器 00 (TMC00)

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

TM00 和 CR000 匹配时清除并开始。

(b) 捕捉 / 比较控制寄存器 00 (CRC00)

				CRC002		CRC001	CRC000
0	0	0	0	0	0	0	0

CR000 用作比较寄存器

(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	0	0	0

(d) 预分频模式寄存器 00 (PRM00)

ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
0	0	0	0	0	0	0/1	0/1

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 获得。

(f) 16 位捕捉 / 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

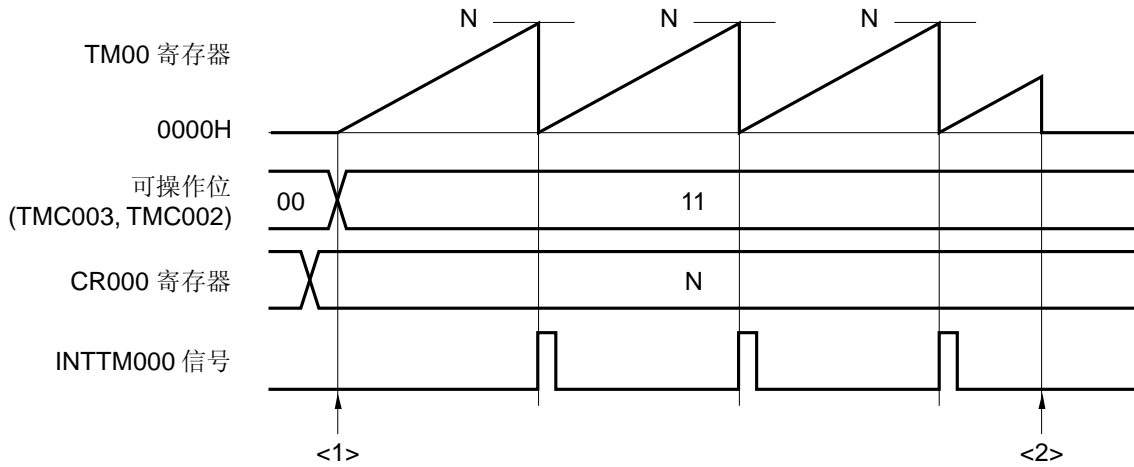
禁止将 CR000 设置为 0000H。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

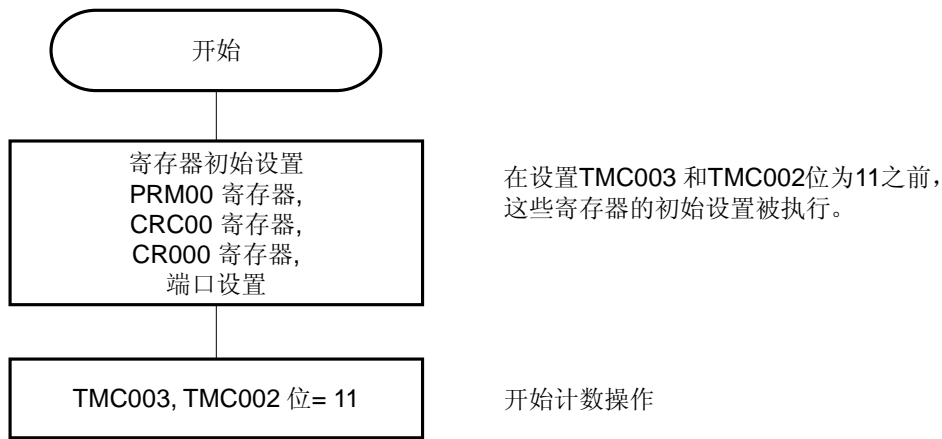
通常情况下 CR010 不会用于间隔定时器功能。然而，当 CR010 的设定值与 TM00 的值匹配时将产生一个比较匹配中断 (INTTM010)。

因此，可使用中断屏蔽标志 (TMMK010) 来屏蔽该中断请求。

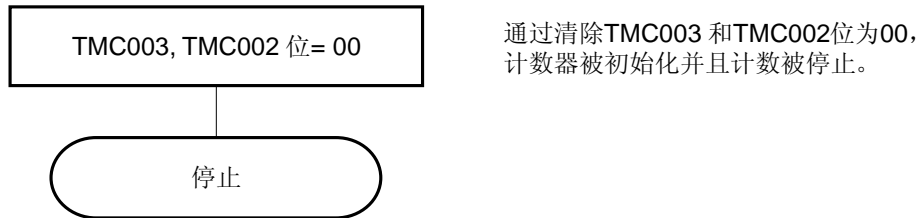
图 7-14. 间隔定时器操作的软件处理示例



<1> 计数操作开始流程



<2> 计数操作停止流程



7.4.2 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（参见 7.4.1），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 设为 11 时（TM00 和 CR000 匹配时进入的计数清零&启动模式），计数操作将与计数时钟同步开始。

当 TM00 的值与 CR000 的值匹配时，TM00 将被清除为 0000H，并产生一个中断信号（INTTM000），而 TO00 输出反转。TO00 固定间隔反转输出使得 TO00 输出一个方波。

- 备注**
1. 关于输入 / 输出引脚设置的详细信息，参见 7.3（5）端口模式寄存器 0（PM0）。
 2. 关于允许 INTTM000 信号中断的方法，参见第十六章 中断功能。

图 7-15. 方波输出操作的框图

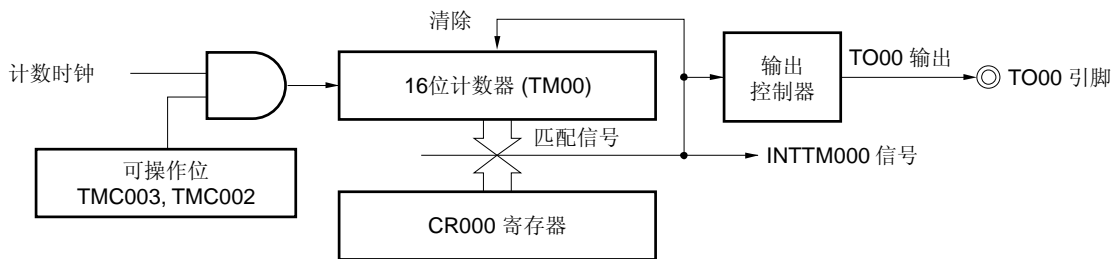


图 7-16. 方波输出操作的基本时序示例

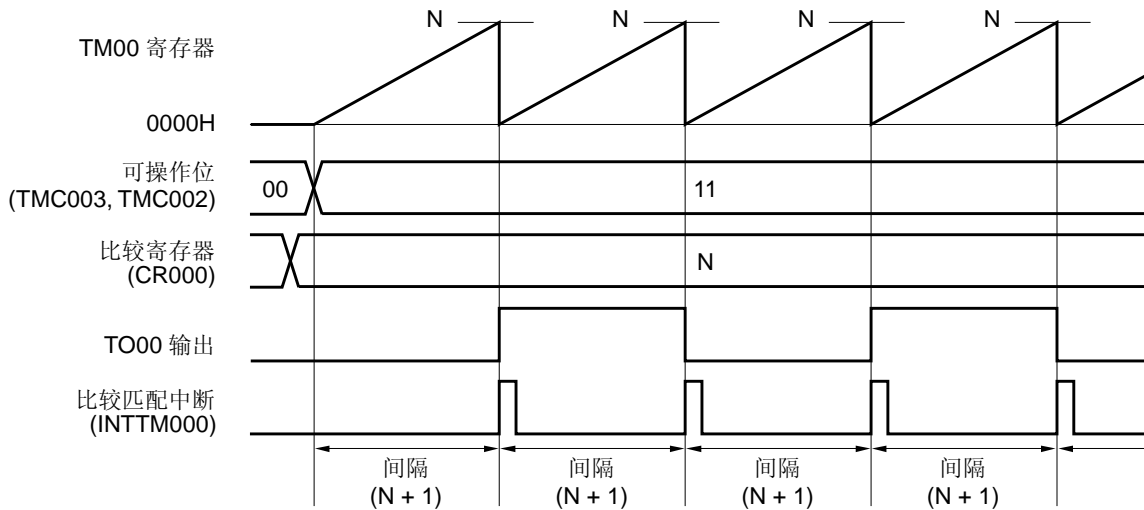
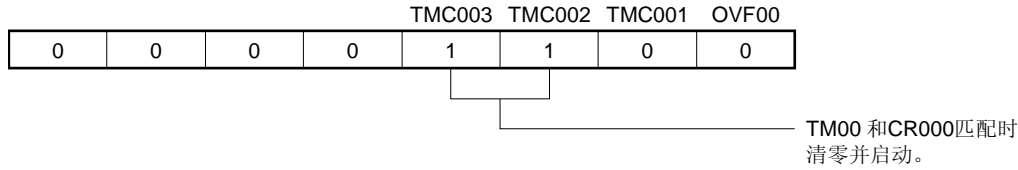
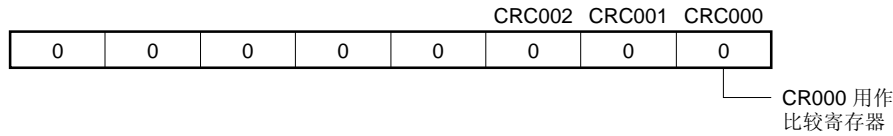


图 7-17. 方波输出操作的寄存器设置示例

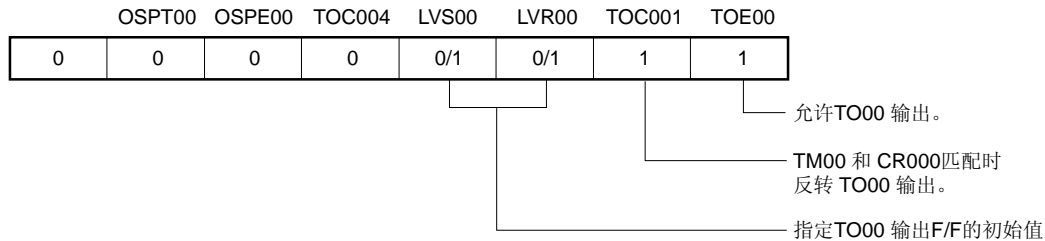
(a) 16 位定时器模式控制寄存器 00 (TMC00)



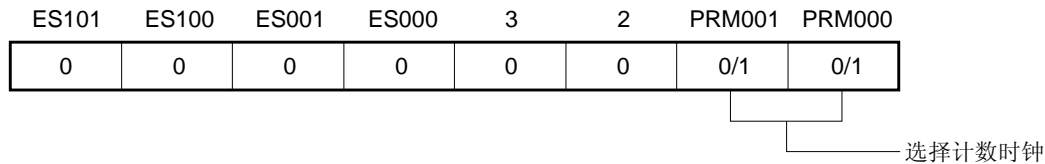
(b) 捕捉 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频模式寄存器 00 (PRM00)



(e) 16 位定时器计数器 00 (TM00)

计数值可以通过读取 TM00 获得。

(f) 16 位捕捉 / 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 = $1 / [2 \times (M + 1) \times \text{计数时钟周期}]$

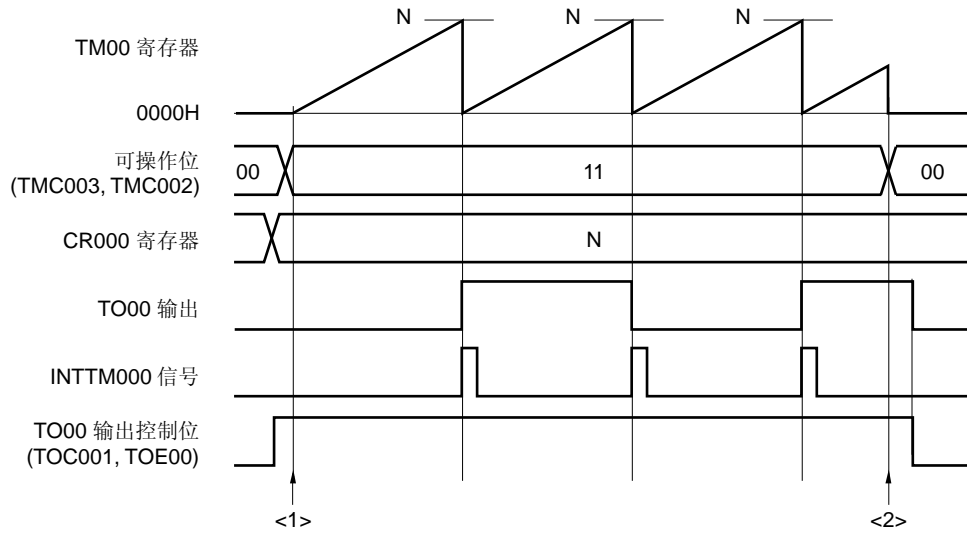
禁止将 CR000 设置为 0000H。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

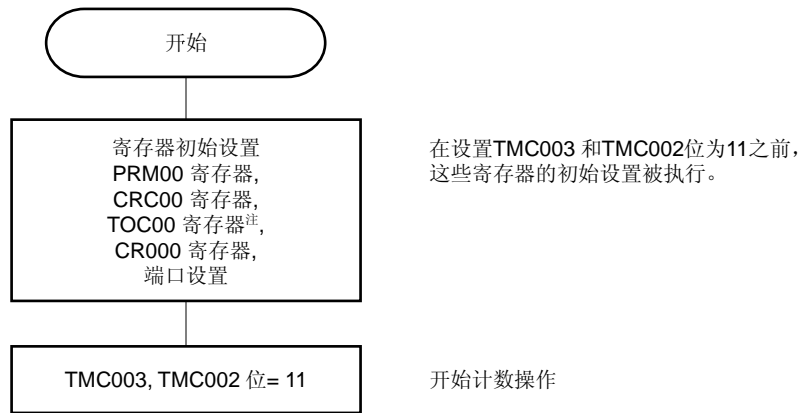
通常情况下 CR010 不会用于方波输出功能。然而，当 CR010 的设定值与 TM00 的值匹配时将产生一个比较匹配中断 (INTTM010)。

因此，应通过使用中断屏蔽标志 (TMMK010) 来屏蔽中断请求。

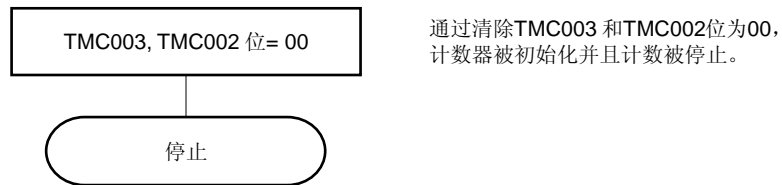
图 7-18. 方波输出功能的软件处理示例



<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。关于详细信息，参见 7.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

7.4.3 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) 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) 和第 3 位和第 2 位 (TMC003 和 TMC002) 被设为 11 时, 将对外部事件输入的有效沿进行计数, 并产生一个表示 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) 进入的清零&启动模式中不能将定时器 / 事件计数器用作外部事件计数器。

INTTM000 信号按以下时序来产生。

- 产生 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产生。

- 产生 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检测到。因此, 可以消除带有短脉冲宽度的噪声。

- 备注**
1. 关于输入 / 输出引脚设置的详细信息, 参见 7.3 (5) 端口模式寄存器 0 (PM0)。
 2. 关于允许 INTTM000 信号中断的方法, 参见第十六章 中断功能。

图 7-19. 外部事件计数器操作的框图

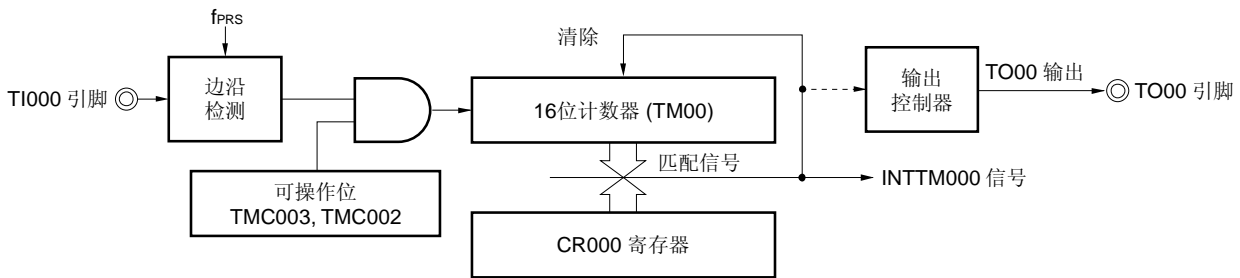
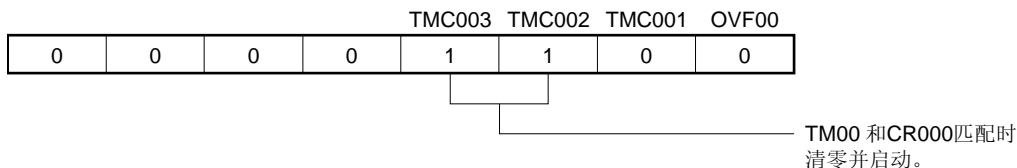
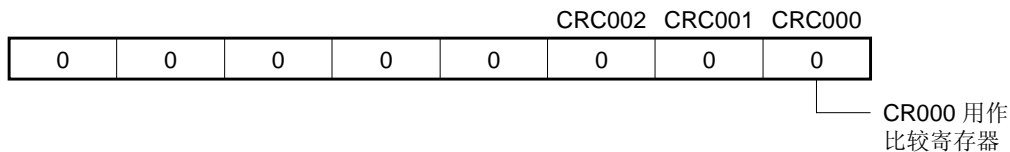


图 7-20. 外部事件计数模式中的寄存器设置示例 (1 / 2)

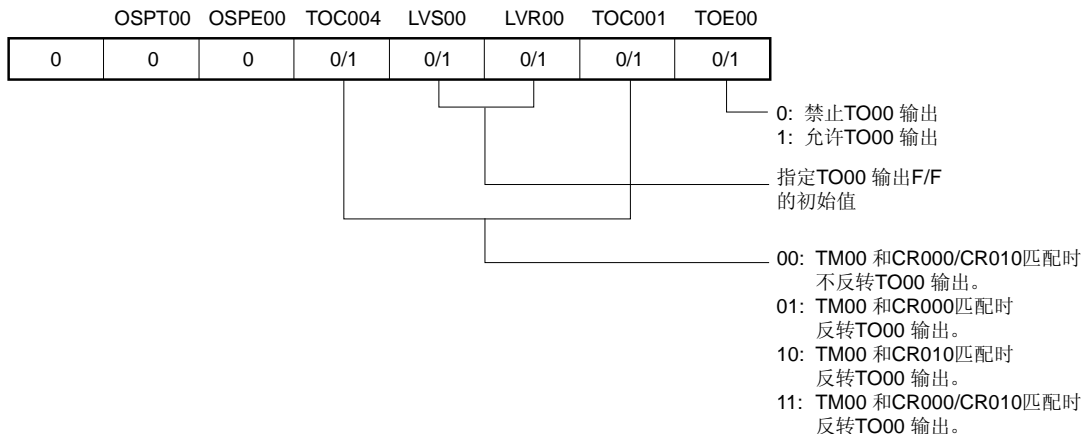
(a) 16 位定时器模式控制寄存器 00 (TMC00)



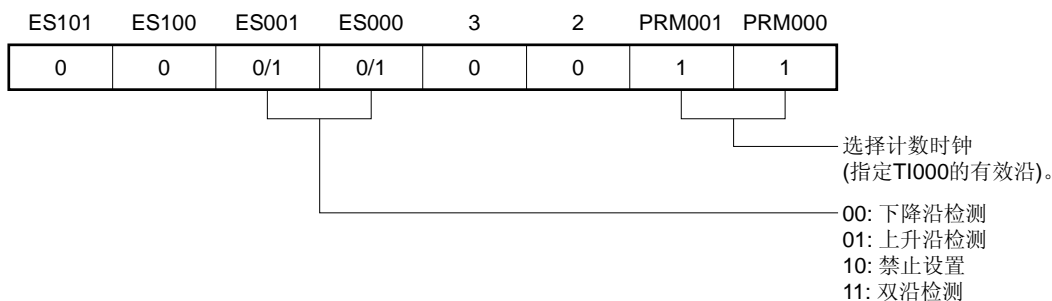
(b) 捕捉 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频模式寄存器 00 (PRM00)



(e) 16 位定时器计数器 00 (TM00)

计数值可以通过读取 TM00 获取。

(f) 16 位捕捉 / 比较寄存器 000 (CR000)

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

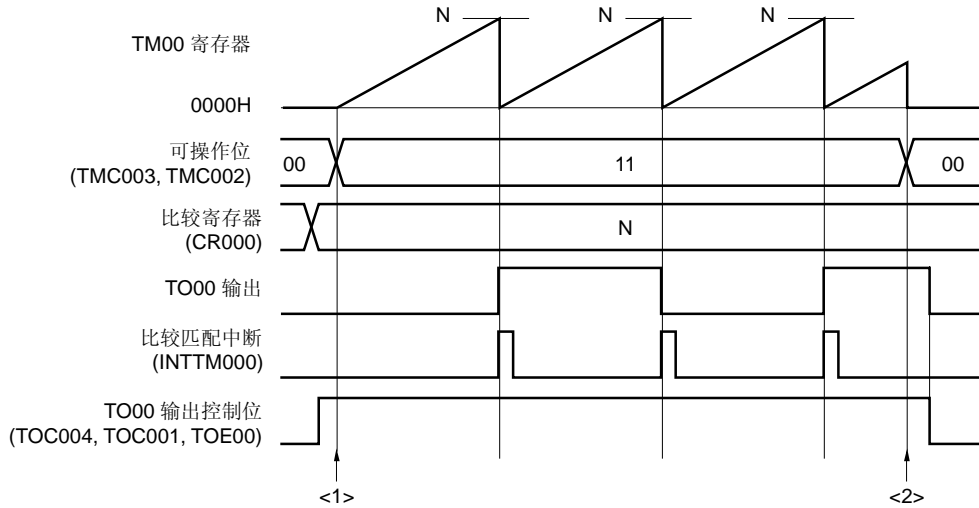
图 7-20. 外部事件计数模式中的寄存器设置示例 (2 / 2)

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

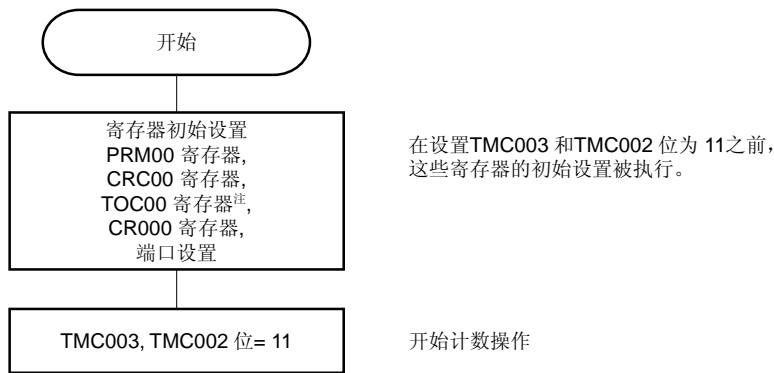
通常情况下，外部事件计数模式中不使用 CR010 寄存器。然而，当 CR010 的设定值与 TM00 的值匹配时将产生一个比较匹配中断 (INTTM010)。

因此，应通过使用中断屏蔽标志 (TMMK010) 来屏蔽中断请求。

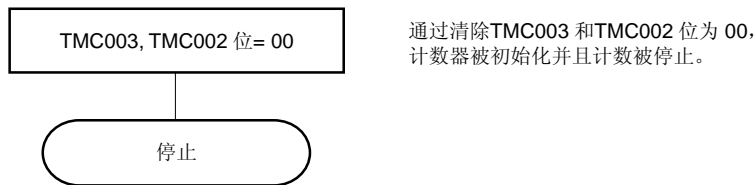
图 7-21. 外部事件计数模式的软件处理示例



<1> 计数操作开始流程



<2> 计数操作停止流程



注 对 TOC00 进行设置时必须谨慎。关于详细信息，参见 7.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

7.4.4 通过 TI000 引脚有效沿输入进入的清零&启动模式中的操作

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) 设为 10 (通过 TI000 引脚有效沿输入进入的清零&启动模式) 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 将溢出并继续计数。

TI000 引脚的有效沿的有效沿是用于清除 TM00。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捉寄存器。

(a) 当 CR000 和 CR010 被用作比较寄存器时

当 TM00 的值与 CR000 和 CR010 的值匹配时将产生 INTTM000 和 INTTM010 信号。

(b) 当 CR000 和 CR010 被用作捕捉寄存器时

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输入到 TI000 引脚中时) TM00 的计数值将被捕捉到 CR000 中且产生 INTTM000 信号。

当有效沿输入到 TI000 引脚中时, TM00 的计数值将被捕捉至 CR010 中并产生 INTTM010 信号。一旦计数值被捕捉, 计数器将被清除为 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 将被清除。

备注 1. 关于输入 / 输出引脚设置的详细信息, 参见 7.3 (5) 端口模式寄存器 0 (PM0)。
2. 关于允许 INTTM000 信号中断的方法, 参见第十六章 中断功能。

(1) 通过 TI000 引脚有效沿输入进入的清零&启动模式中的操作

(CR000: 比较寄存器, CR010: 比较寄存器)

图 7-22. 通过 TI000 引脚有效沿输入进入的清零&启动模式的框图
(CR000: 比较寄存器, CR010: 比较寄存器)

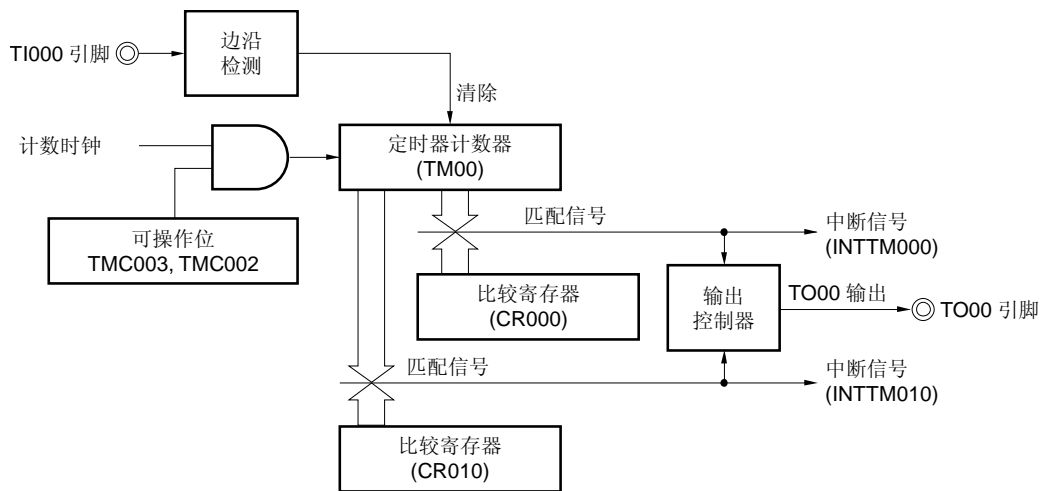
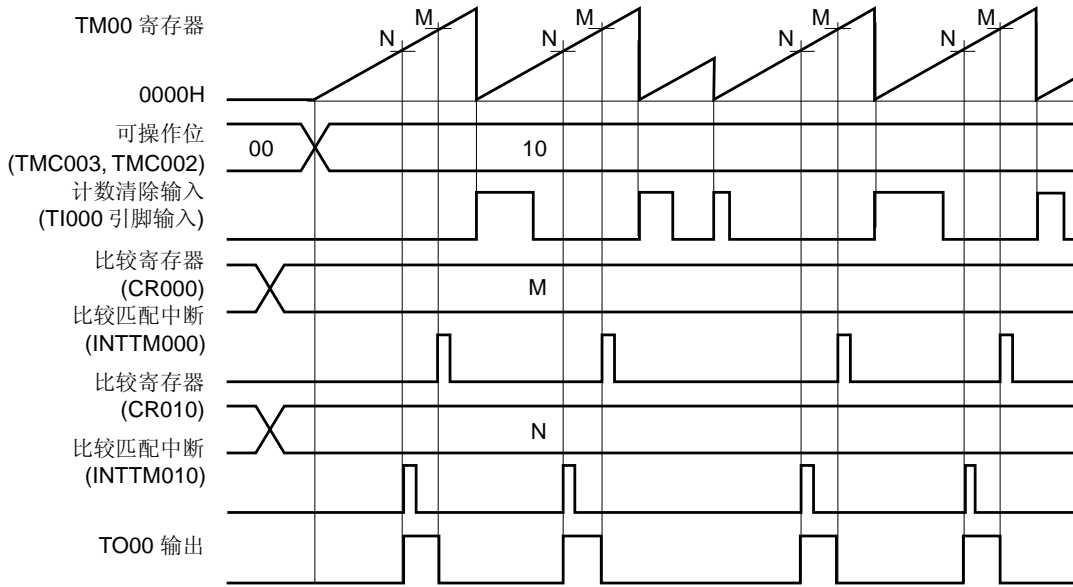
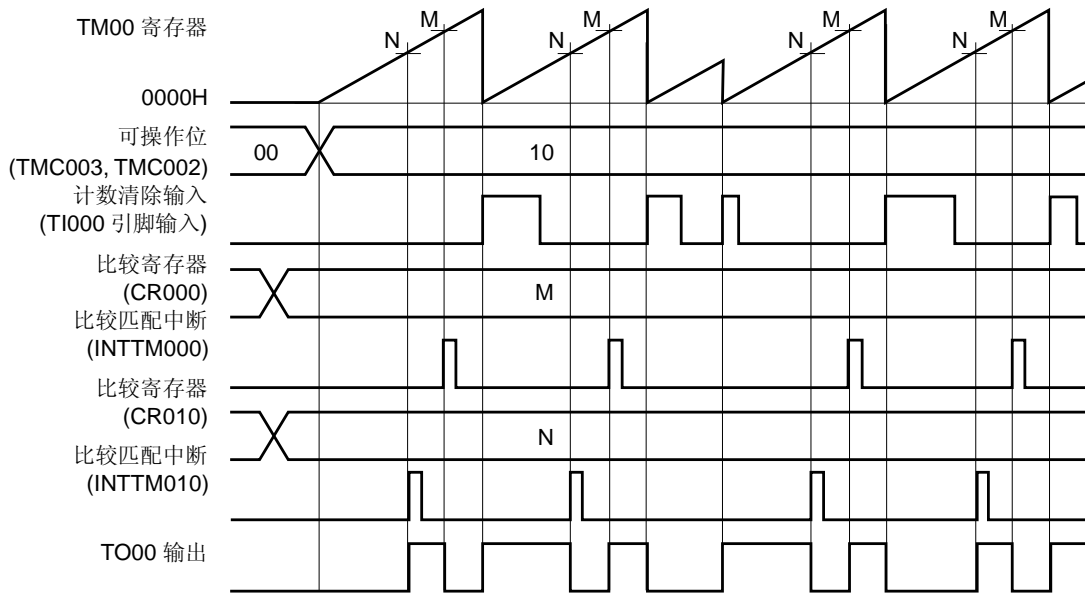


图 7-23. 通过 TI000 引脚有效沿输入进入的清零&启动模式的时序示例
(CR000: 比较寄存器, CR010: 比较寄存器)

(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 00 (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 的输出电平反转。
- (b) 当 TM00 与比较寄存器匹配或当 TI000 的有效沿被检测到时 TO00 的输出电平将被反转。

(2) 通过 TI000 引脚有效沿输入进入的清零&启动模式中的操作
(CR000: 比较寄存器, CR010: 捕捉寄存器)

图 7-24. 通过 TI000 引脚有效沿输入进入的清零&启动模式的框图
(CR000: 比较寄存器, CR010: 捕捉寄存器)

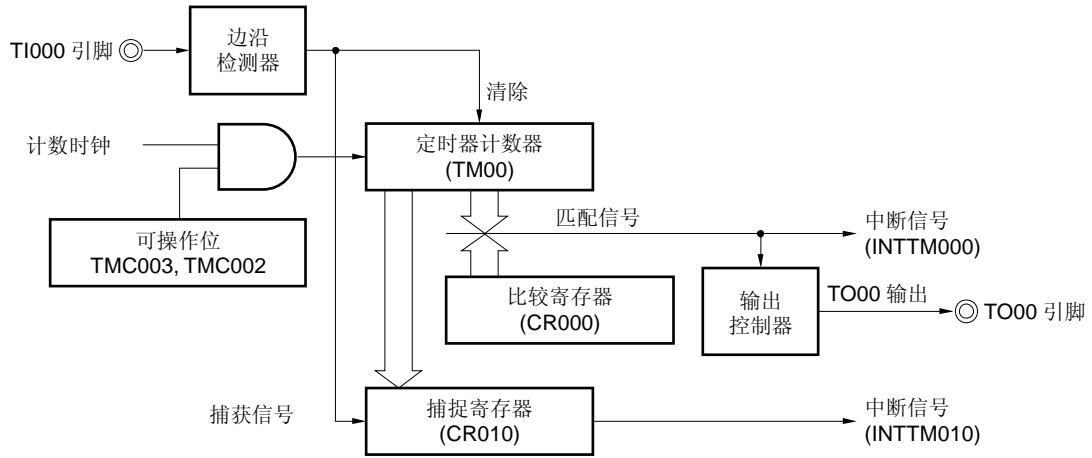
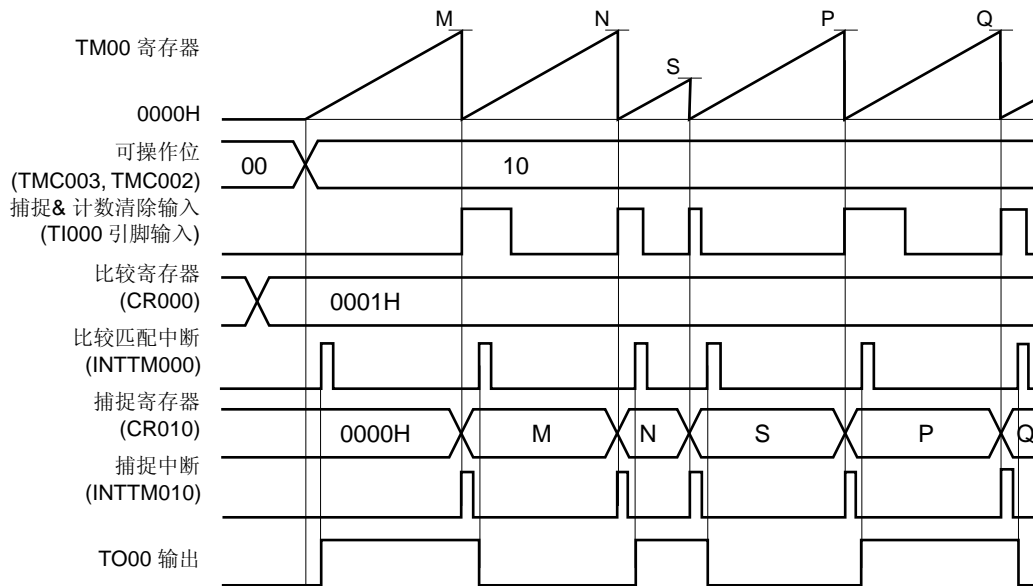


图 7-25. 通过 TI000 引脚有效沿输入进入的清零&启动模式的时序示例
(CR000: 比较寄存器, CR010: 捕捉寄存器) (1 / 2)

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

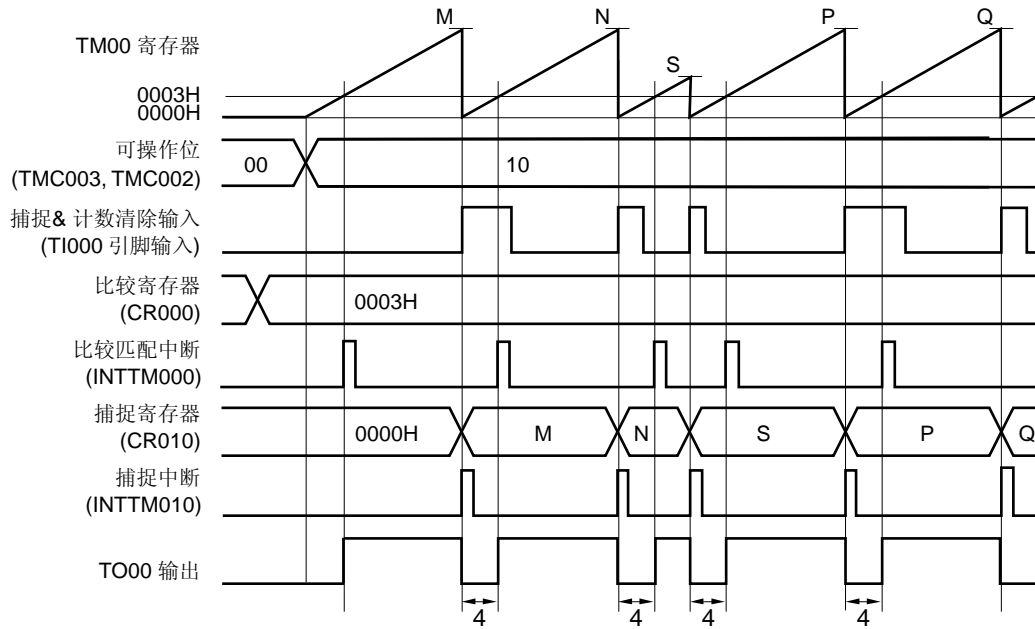


这是计数值被捕捉&清除时 TO00 的输出电平反转的应用示例。

当检测到 TI000 引脚的有效沿时, 计数值将被捕捉至 CR010 中且 TM00 也会被清除 (为 0000H)。当 TM00 的计数值为 0001H 时, 将产生一个比较匹配中断信号 (INTTM000), 且 TO00 的输出电平将被反转。

图 7-25. 通过 TI000 引脚有效沿输入进入的清零&启动模式的时序示例
(CR000: 比较寄存器, CR010: 捕捉寄存器) (2 / 2)

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



这是计数值被捕捉&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, 计数值将被捕捉至 CR010, 并产生一个捕捉中断信号 (INTTM010), TM00 也被清零 (为 0000H), 而 TO00 的输出电平被反转。当 TM00 的计数值为 0003H 时 (已经计数了 4 个时钟), 将产生一个比较匹配中断信号 (INTTM000), 且 TO00 的输出电平反转。

(3) 通过 TI000 引脚有效沿输入进入的清零&启动模式中的操作
 (CR000: 捕捉寄存器, CR010: 比较寄存器)

图 7-26. 通过 TI000 引脚有效沿输入进入的清零&启动模式的框图
 (CR000: 捕捉寄存器, CR010: 比较寄存器)

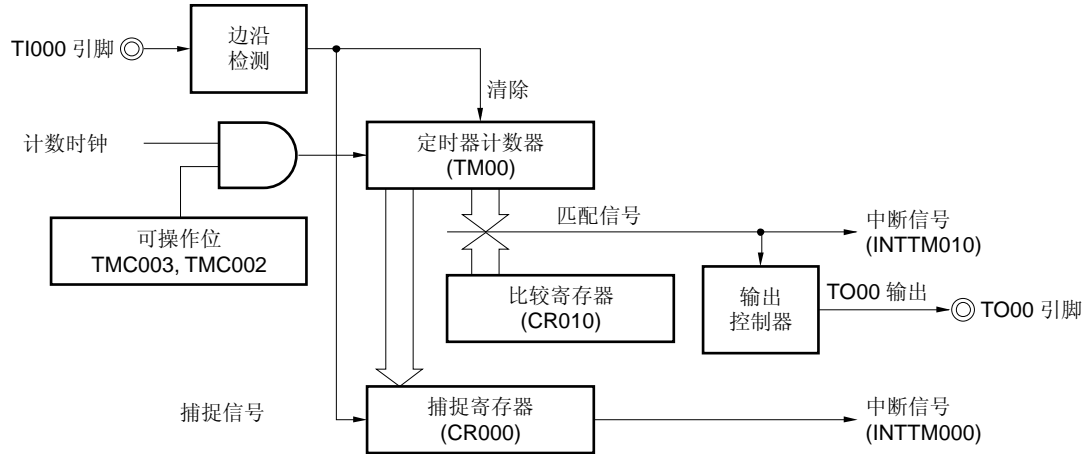
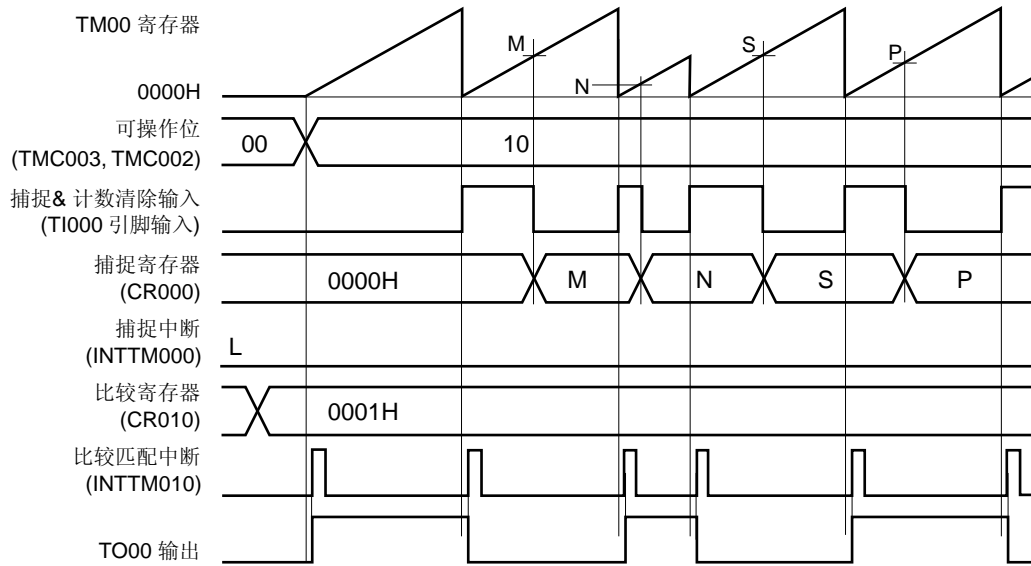


图 7-27. 通过 TI000 引脚有效沿输入进入的清零&启动模式的时序示例
(CR000: 捕捉寄存器, CR010: 比较寄存器) (1 / 2)

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



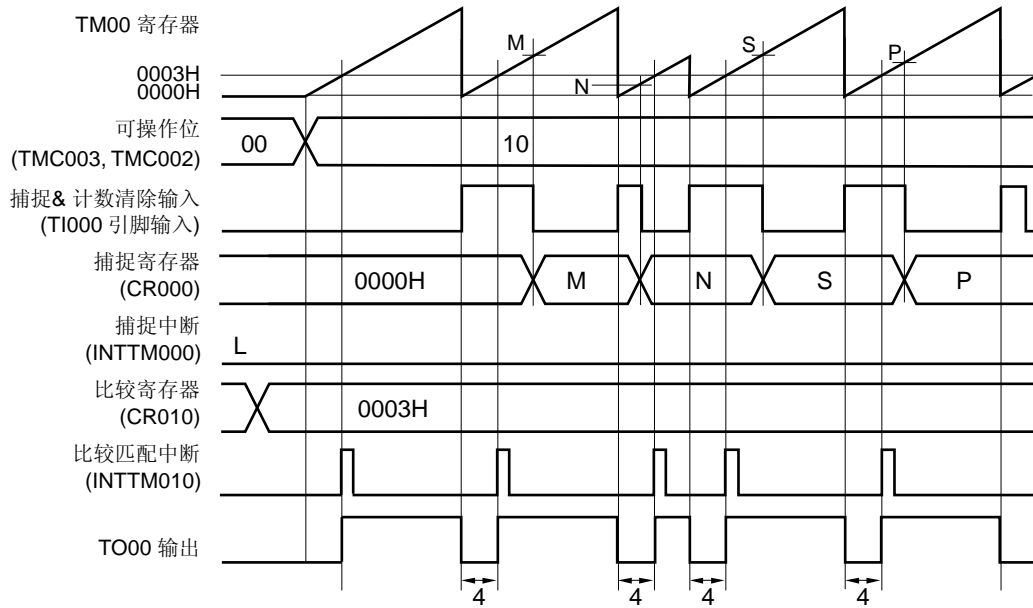
这是计数值被捕捉&清除时 TO00 的输出电平发生反转的应用示例。

TM00 在 TI000 引脚上升沿检测时清除, 并会在 TI000 引脚的下降沿检测时被捕捉至 CR000。

当捕捉 / 比较控制寄存器 00 (CRC00) 的第 1 位 (CRC001) 被设为 1 时, TM00 的计数值将在与输入到 TI000 引脚的信号相反的相位中捕捉至 CR000 中, 但不产生捕捉中断信号 (INTTM000)。然而, 在检测到 TI010 引脚的有效沿时将产生 INTTM000 信号。在不使用 INTTM000 信号时将其屏蔽。

图 7-27. 通过 TI000 引脚有效沿输入进入的清零&启动模式的时序示例
(CR000: 捕捉寄存器, CR010: 比较寄存器) (2 / 2)

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



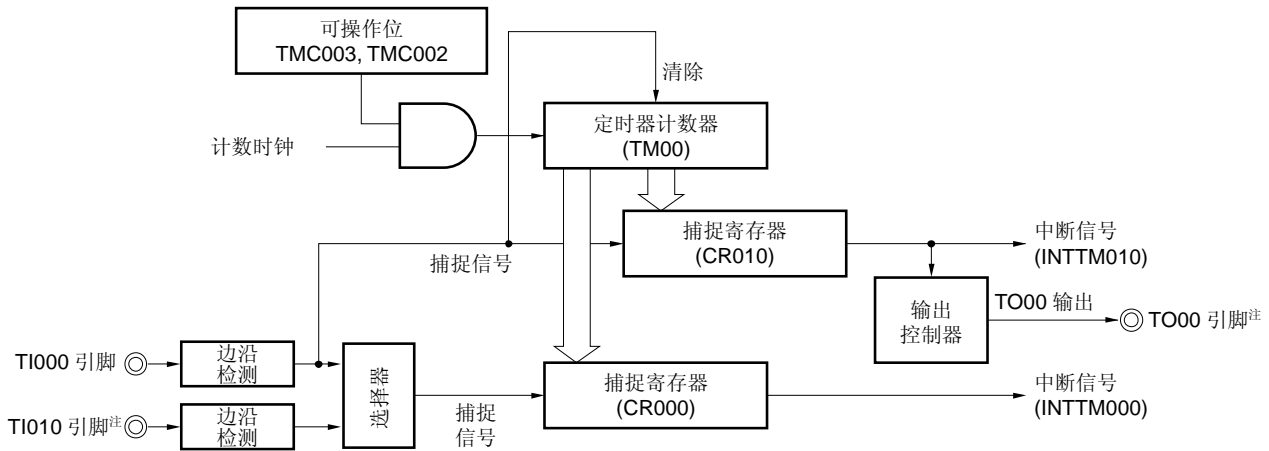
这是计数值被捕捉&清除时赋值于 CR010 的宽度 (该示例中为 4 个时钟) 从 TO00 引脚中输出的应用示例。

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捉至 CR000。当 TM00 被清除 (为 0000H) 时 TO00 的输出电平将被反转, 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。

当捕捉 / 比较控制寄存器 00 (CRC00) 的第 1 位 (CRC001) 被设为 1 时, TM00 的计数值将在与 TI000 引脚输入信号相反的相位中被捕捉至 CR000 中, 但不会产生捕捉中断信号 (INTTM000)。然而, 在检测到 TI010 引脚的有效沿时将产生 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的清零&启动模式中的操作
(CR000: 捕捉寄存器, CR010: 捕捉寄存器)

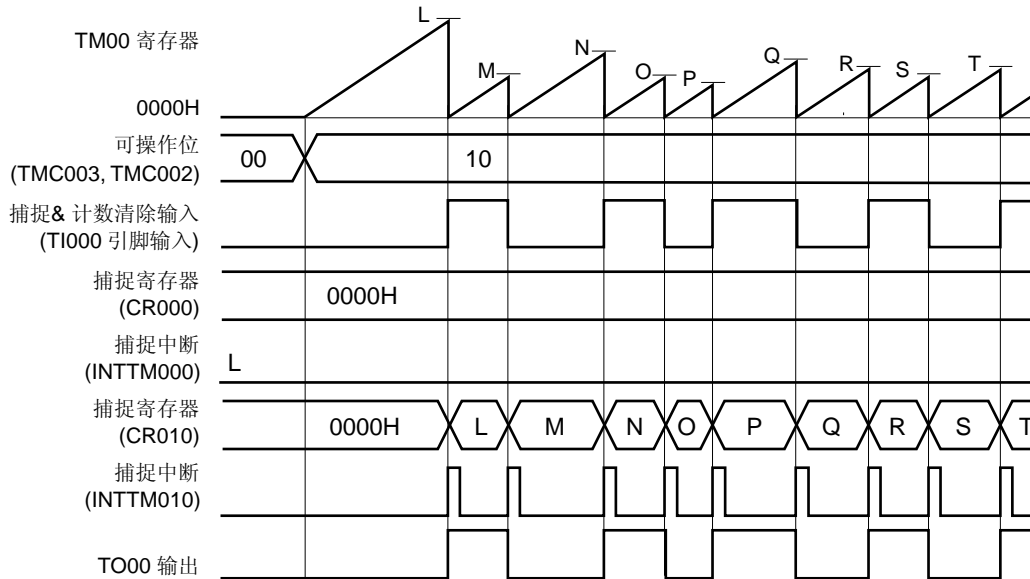
图 7-28. 通过 TI000 引脚有效沿输入进入的清零&启动模式的框图
(CR000: 捕捉寄存器, CR010: 捕捉寄存器)



注 在检测 TI010 引脚的有效沿时不能使用定时器输出 (TO00)。

图 7-29. 通过 TI000 引脚有效沿输入进入的清零&启动模式的时序示例
(CR000: 捕捉寄存器, CR010: 捕捉寄存器) (1 / 3)

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

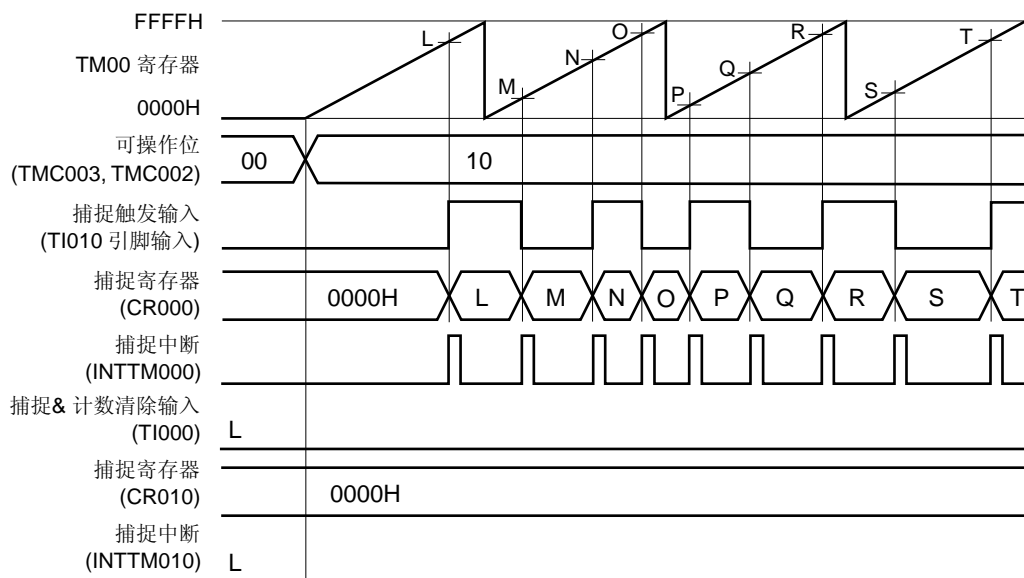


在该示例中, 当检测到 TI000 的上升沿或下降沿时, 计数值将被捕捉至 CR010, TM00 将被清除, 而 TO00 引脚输出也将被反转。

当检测到 TI010 引脚的边沿时, 将产生一个中断信号 (INTTM000)。在不使用 INTTM000 信号时将其屏蔽。

图 7-29. 通过 TI000 引脚有效沿输入进入的清零&启动模式的时序示例
(CR000: 捕捉寄存器, CR010: 捕捉寄存器) (2 / 3)

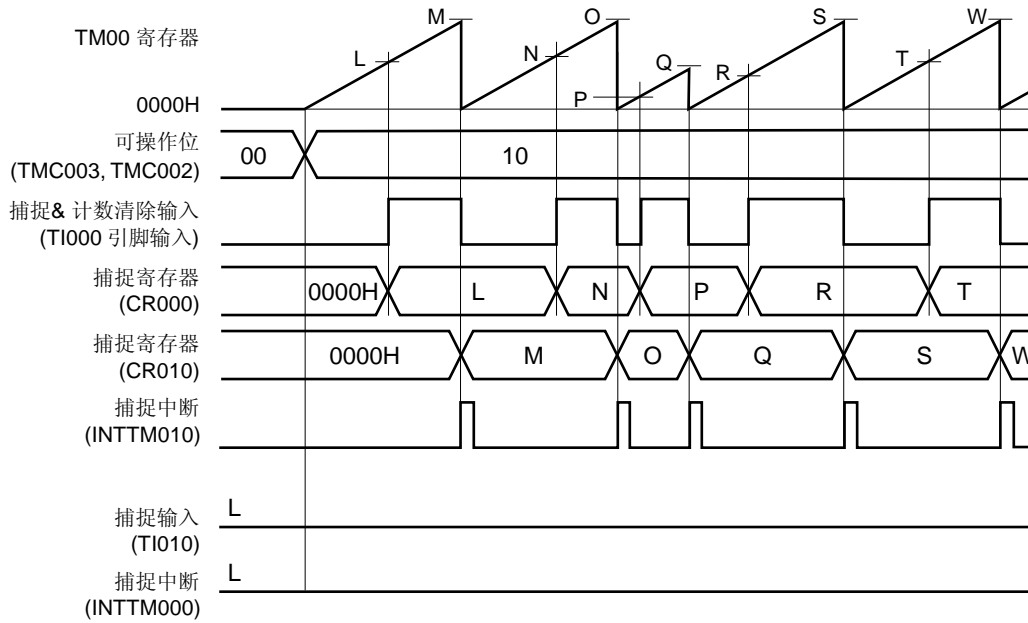
(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 0AH



在该时序示例中，没有边沿被输入到 TI000 引脚中，实际应用中当检测到 TI010 引脚的上升沿或下降时，计数值被捕捉至 CR000。

图 7-29. 通过 TI000 引脚有效沿输入进入的清零&启动模式的时序示例
(CR000: 捕捉寄存器, CR010: 捕捉寄存器) (3 / 3)

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



该应用示例中，测量输入到 TI000 引脚中的信号脉冲宽度。

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）中被捕捉至 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。

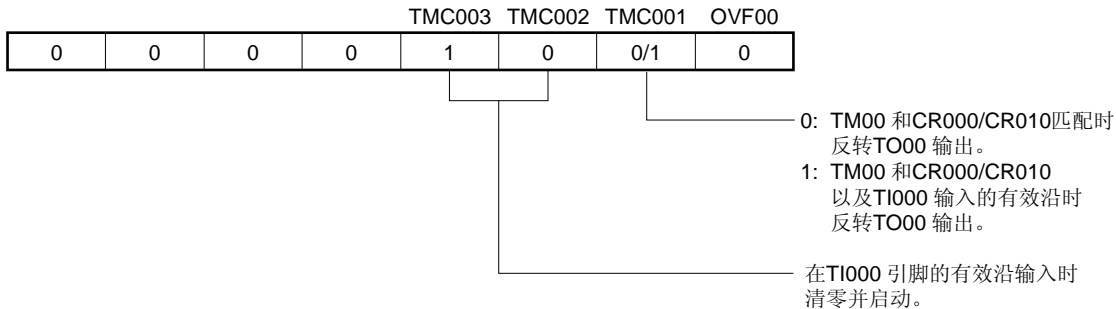
- 高电平宽 = [CR010 值] - [CR000 值] × [计算时钟周期]
- 低电平宽 = [CR000 值] × [计算时钟周期]

如果将 TI000 引脚的反向作为触发将计数值捕捉至 CR000，那么将不能产生 INTTM000 信号。在产生 INTTM010 信号后立即读取 CR000 和 CR010 的值以测量脉冲宽度。

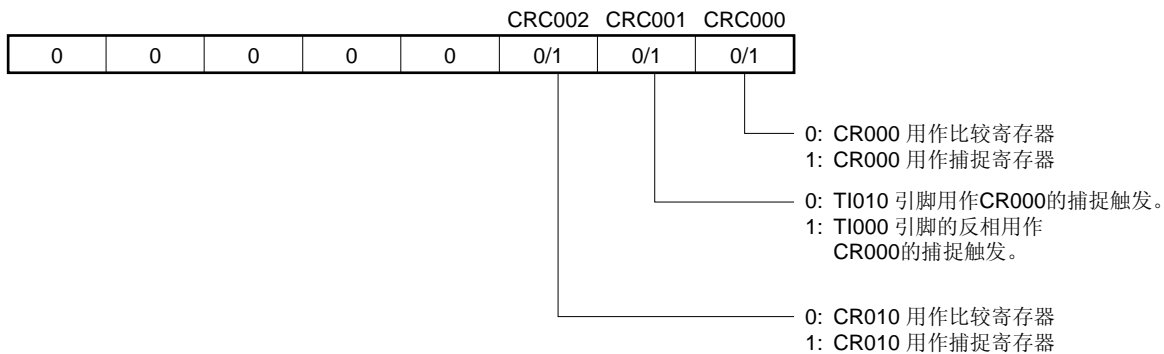
然而，如果由预分频模式寄存器 00 (PRM00) 的第 6 位和第 5 位 (ES101 和 ES100) 指定的有效沿被输入至 TI010 引脚，那么计数值不捕捉但会产生 INTTM000 信号。若要测量 TI000 引脚的脉冲宽度，则在不使用 INTTM000 信号时将其屏蔽。

图 7-30. 通过 TI000 引脚有效沿输入进入的清零&启动模式的寄存器设置示例 (1 / 2)

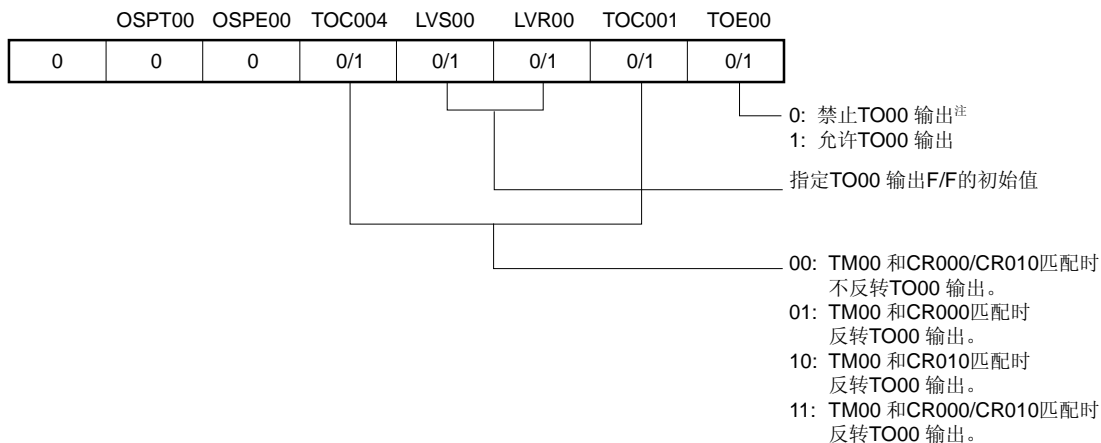
(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕捉 / 比较控制寄存器 00 (CRC00)



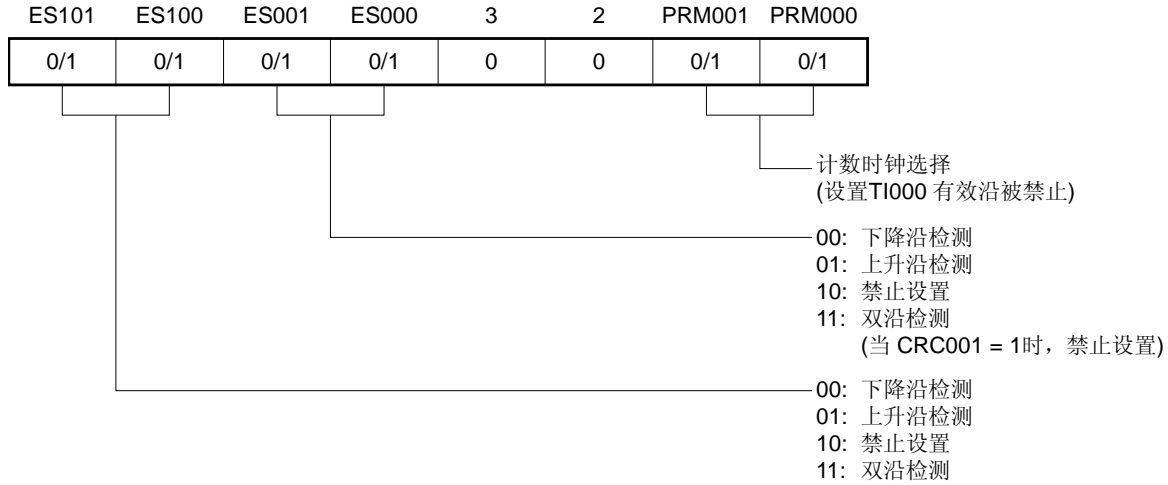
(c) 16 位定时器输出控制寄存器 00 (TOC00)



注 在检测 TI010 引脚的有效沿时不能使用定时器输出 (TO00)。

图 7-30. 通过 TI000 引脚有效沿输入进入的清零&启动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 获得。

(f) 16 位捕捉 / 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, 将产生一个中断信号 (INTTM000)。TM00 的计数值被清除。

若要将该寄存器用作捕捉寄存器, 则应将 TI000 或 TI010 引脚^注选作捕捉触发。当检测到捕捉触发的有效沿时, TM00 的计数值将被保存在 CR000 中。

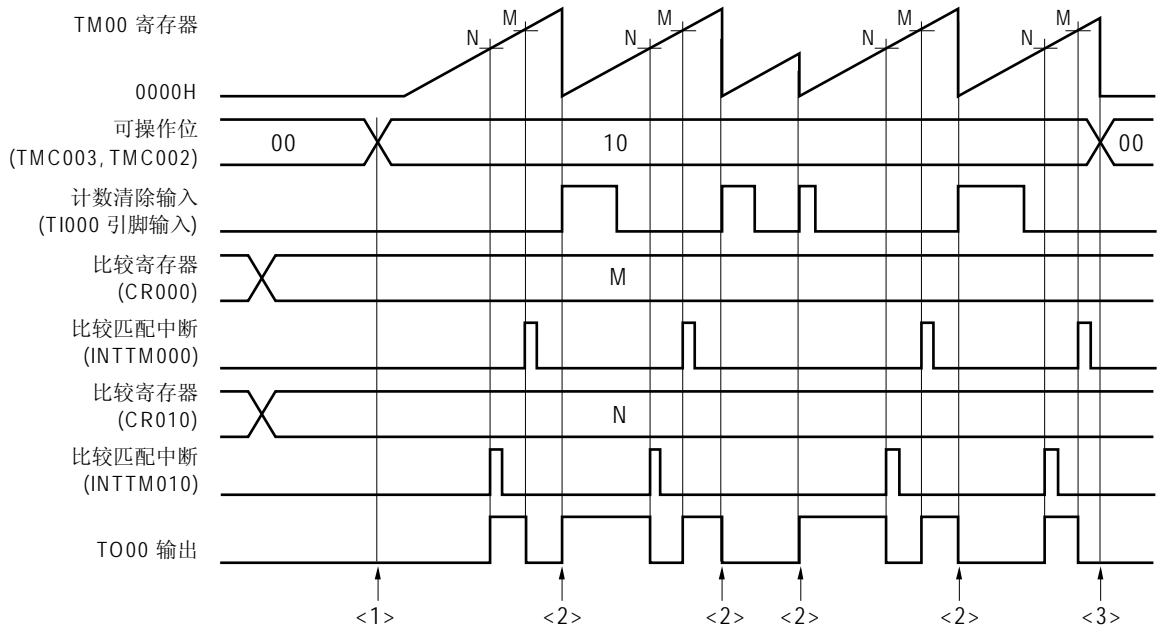
注 在检测 TI010 引脚的有效沿时不能使用定时器输出 (TO00)。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

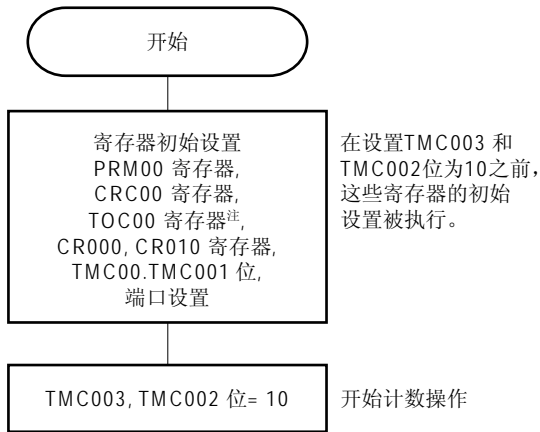
当该寄存器用作比较寄存器且该寄存器的值与 TM00 的计数值匹配时, 将产生一个中断信号 (INTTM010)。TM00 的计数值被清除。

当该寄存器被用作捕捉寄存器时, TI000 引脚输入将被用作捕捉触发。当检测到捕捉触发的有效沿时, TM00 的计数值将被保存在 CR010 中。

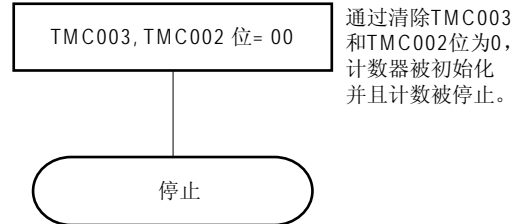
图 7-31. 通过 TI000 引脚有效沿输入进入的清零&启动模式的软件处理示例



<1> 计数操作开始流程



<3> 计数操作停止流程



<2> TM00 寄存器清零 & 启动流程



注 对 TOC00 进行设置时必须谨慎。关于详细信息，参见 7.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

7.4.5 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 (自由运行定时器模式) 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 被清零 (为 0000H) 并继续计数。可通过软件执行 CLR 指令, 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 均用作比较寄存器。
- CR000 或 CR010 中的一个寄存器用作比较寄存器, 另一个用作捕捉寄存器。
- CR000 和 CR010 均用作捕捉寄存器。

备注

1. 关于输入 / 输出引脚设置的详细信息, 参见 7.3 (5) 端口模式寄存器 0 (PM0)。
2. 关于允许 INTTM000 信号中断的方法, 参见第十六章 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 7-32. 自由运行定时器模式的框图
(CR000: 比较寄存器, CR010: 比较寄存器)

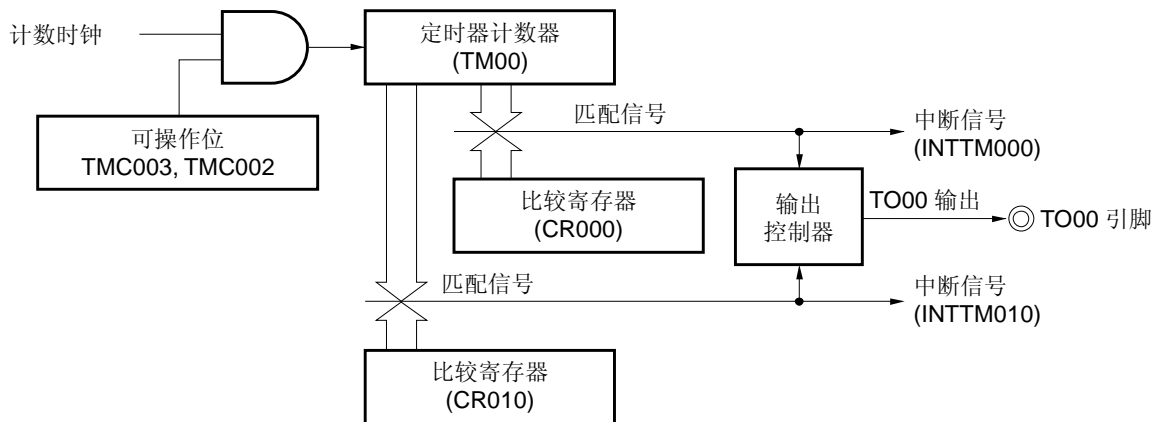
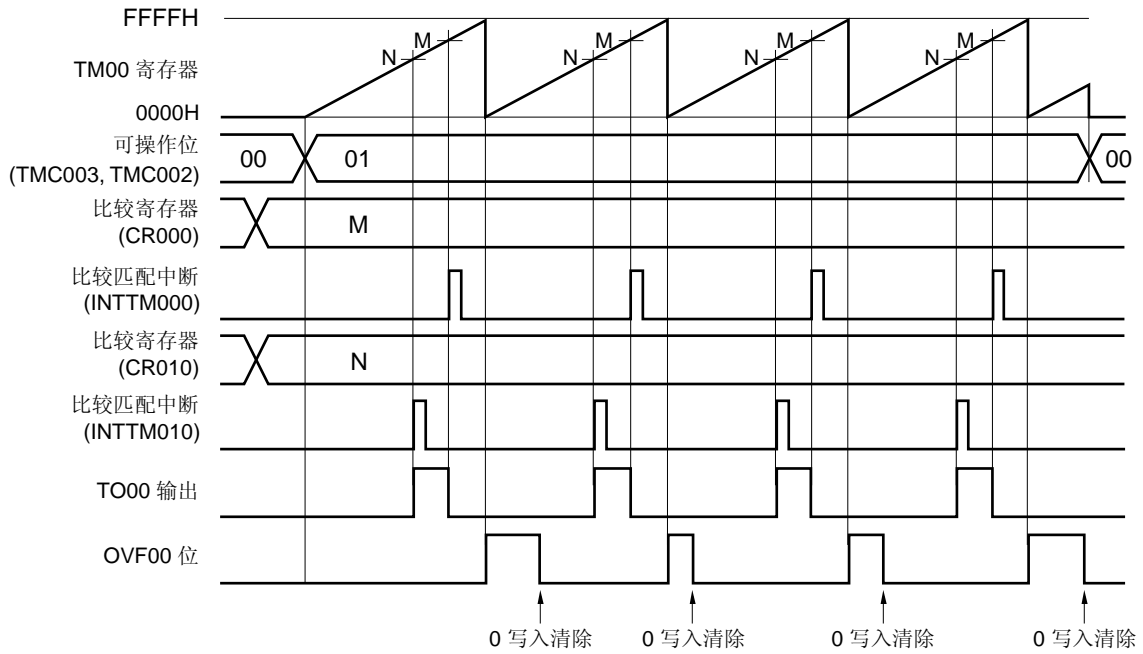


图 7-33. 自由运行定时器模式的时序示例
(CR000: 比较寄存器, CR010: 比较寄存器)

• 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 的输出电平都会反转。当计数值与寄存器值匹配时，将产生 INTTM000 或 INTTM010 信号。

(2) 自由运行定时器模式操作

(CR000: 比较寄存器, CR010: 捕捉寄存器)

图 7-34. 自由运行定时器模式的框图
(CR000: 比较寄存器, CR010: 捕捉寄存器)

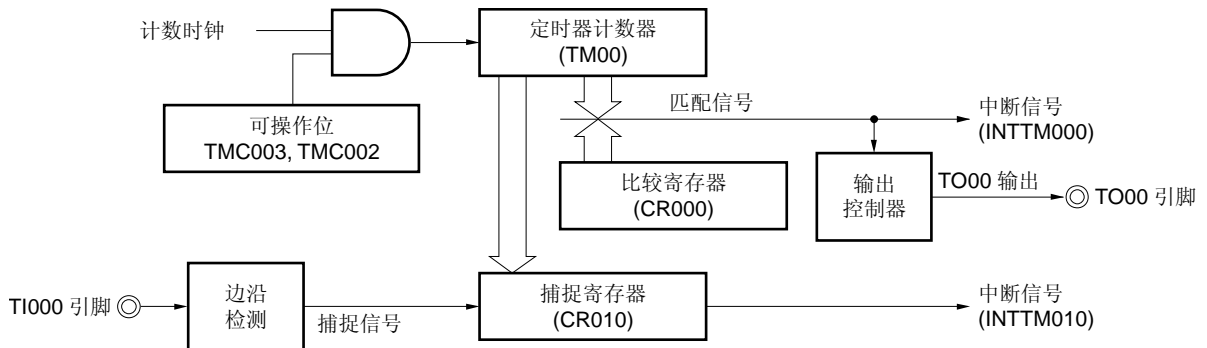
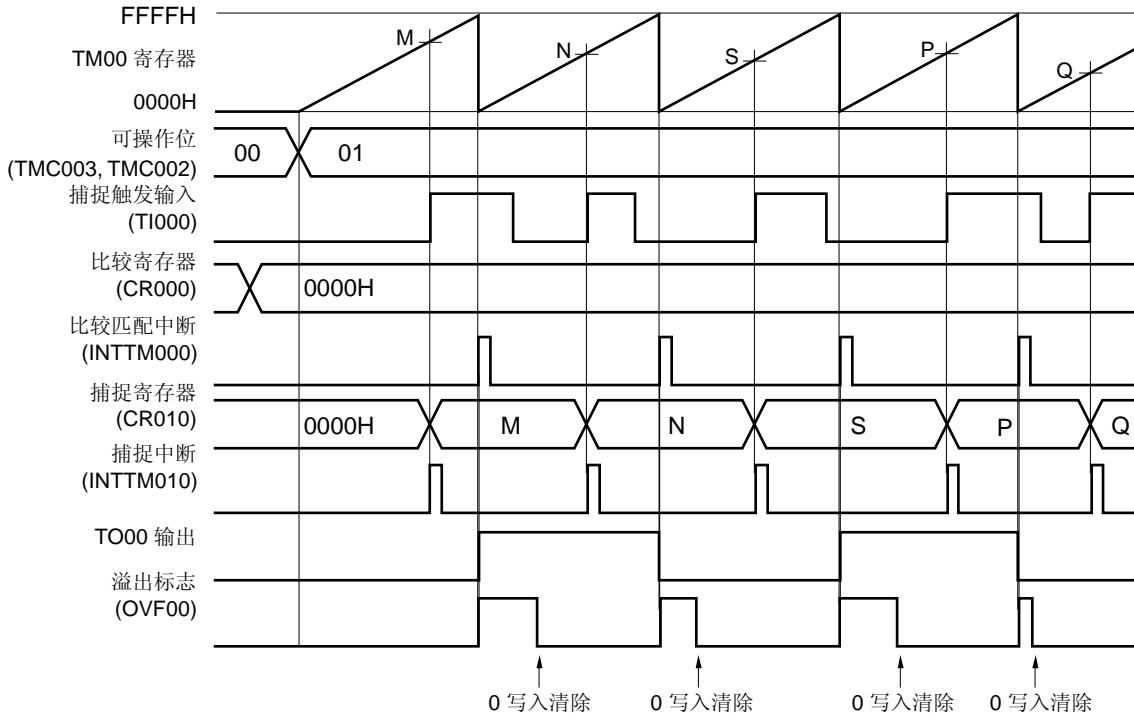


图 7-35. 自由运行定时器模式的时序示例
(CR000: 比较寄存器, CR010: 捕捉寄存器)

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



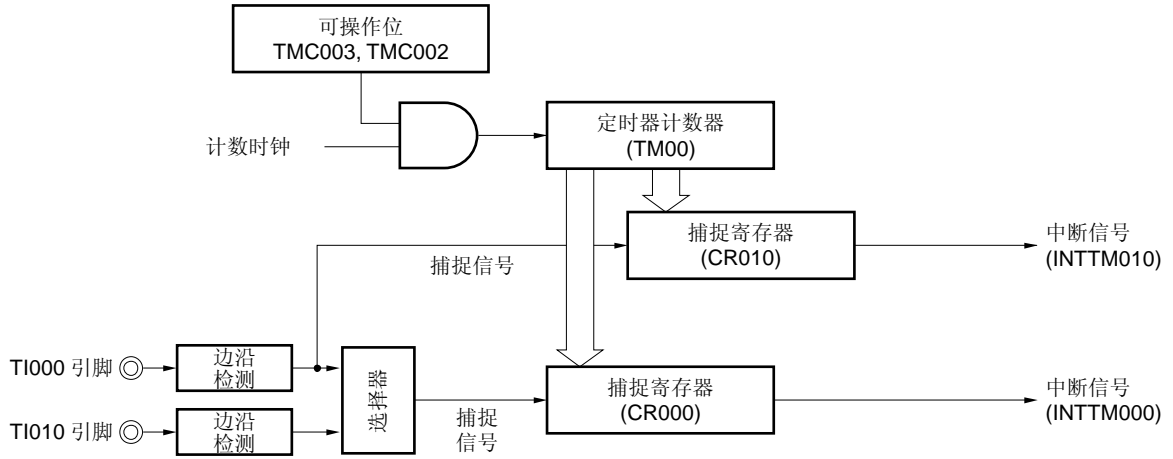
在该应用示例中，比较寄存器和捕捉寄存器同时在自由运行定时器模式中使用。

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产生 INTTM000 信号，且 TO00 的输出电平反转。此外，每次检测到 TI000 引脚的有效沿时都产生 INTTM010 信号，且 TM00 的计数值也捕捉至 CR010。

(3) 自由运行定时器模式操作

(CR000: 捕捉寄存器, CR010: 捕捉寄存器)

图 7-36. 自由运行定时器模式的框图
(CR000: 捕捉寄存器, CR010: 捕捉寄存器)

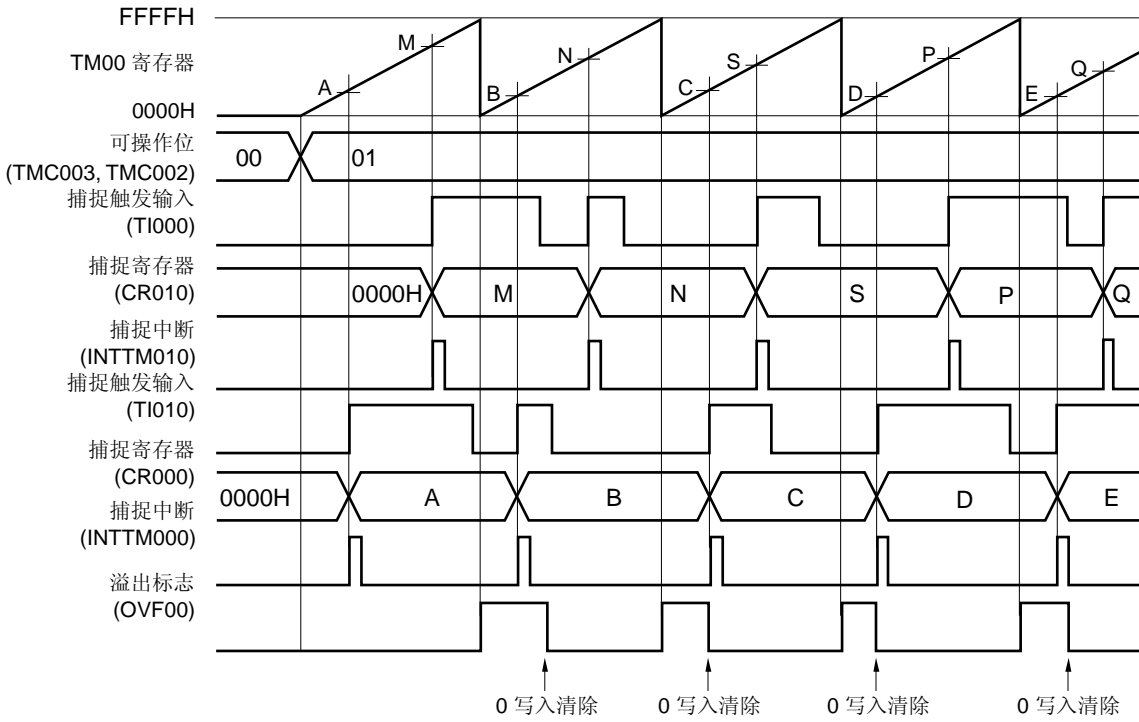


备注 如果 CR000 和 CR010 在自由选择定时器模式中均被用作捕捉寄存器，那么 TO00 的输出电平将不会反转。

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 引脚有效沿时 TO00 引脚输出电平反转。

图 7-37. 自由运行定时器模式的时序示例
(CR000: 捕捉寄存器, CR010: 捕捉寄存器) (1 / 2)

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

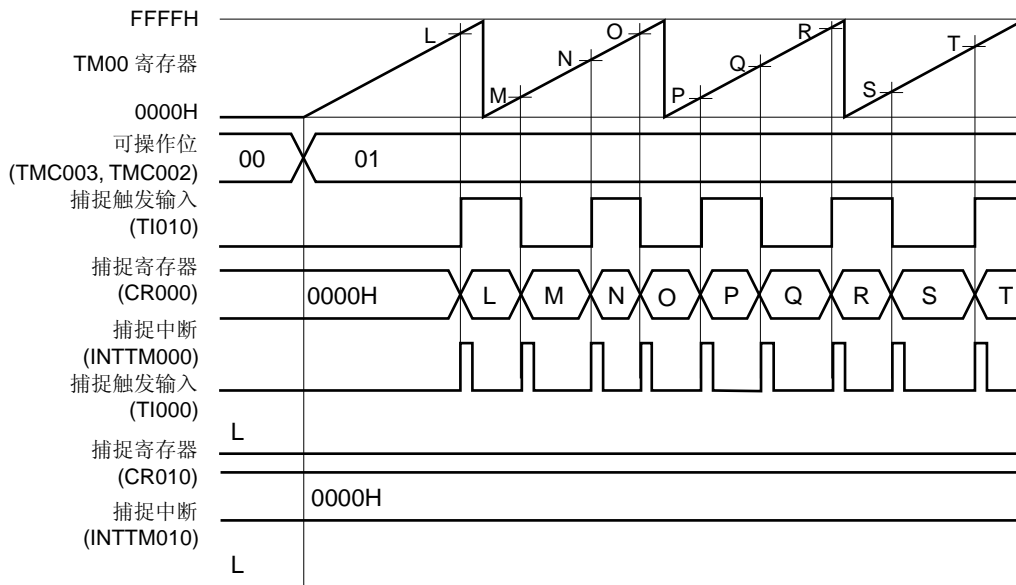


在该示例中，在单独的捕捉触发信号的有效沿处捕捉的计数值在自由运行定时器模式中被保存在单独的捕捉寄存器中。

在检测到 TI000 引脚的有效沿时计数值捕捉至 CR010，而在检测到 TI010 引脚输入的有效沿时计数值则捕捉至 CR000 中。

图 7-37. 自由运行定时器模式的时序示例
 (CR000: 捕捉寄存器, CR010: 捕捉寄存器) (2 / 2)

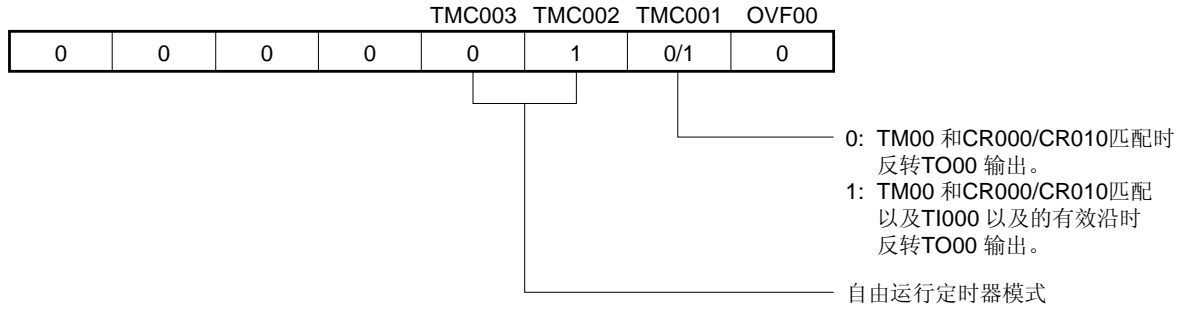
(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 04H



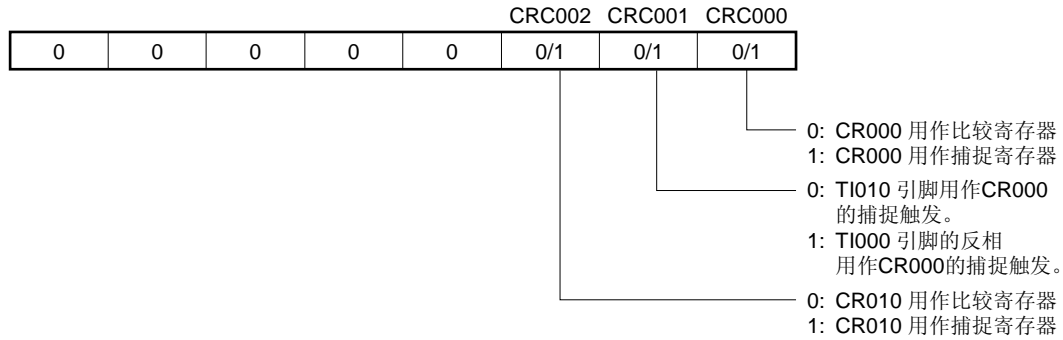
该示例中，TI010 引脚的双边沿均检测，且计数值在自由运行定时器模式中被捕捉至 CR000、
 当 CR000 和 CR010 均用作捕捉寄存器但只有 TI010 引脚的有效沿被检测到时，计数值不捕捉至 CR010。

图 7-38. 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)



(b) 捕捉 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)

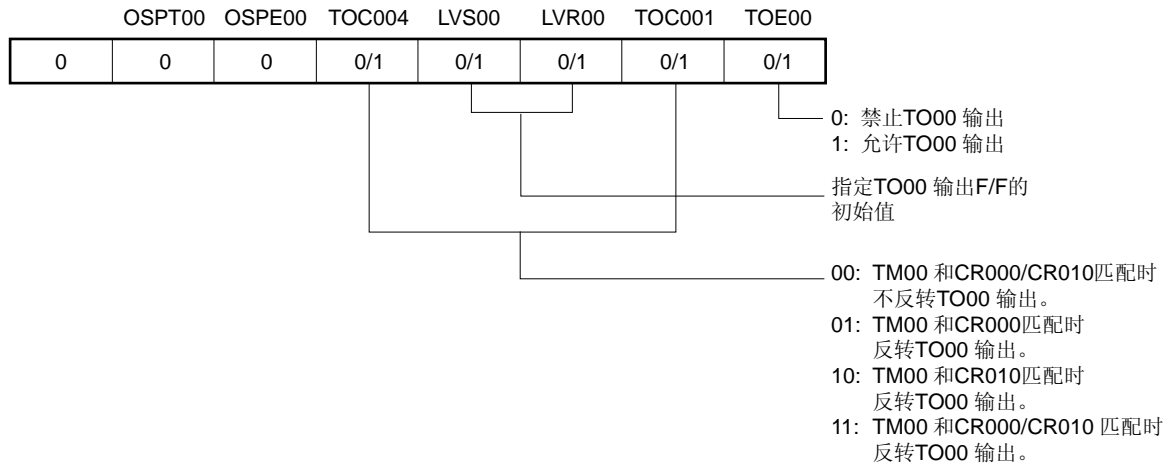
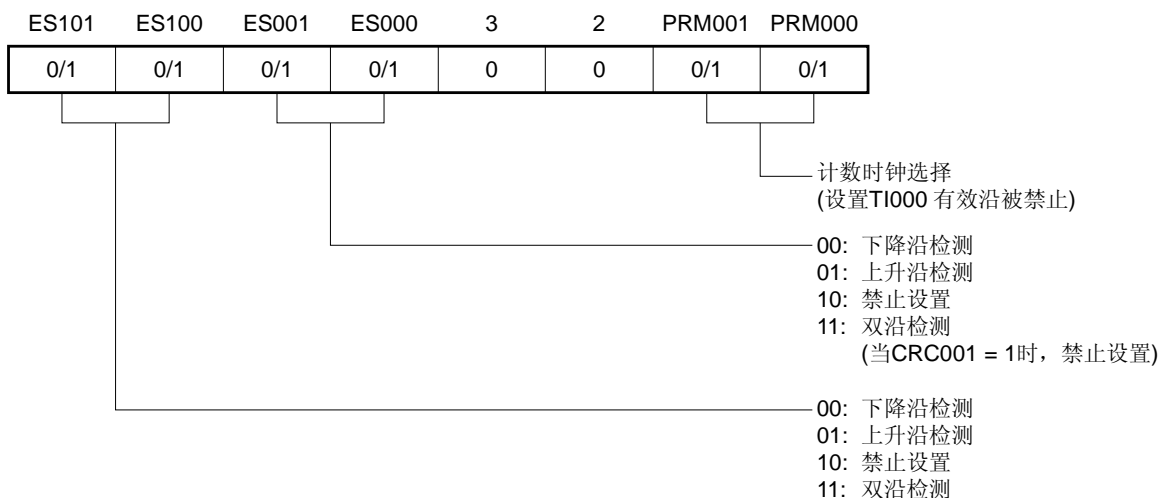


图 7-38. 自由运行定时器模式的寄存器设置示例 (2 / 2)

(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 获得。

(f) 16 位捕捉 / 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, 将产生一个中断信号 (INTTM000)。TM00 的计数值被清除。

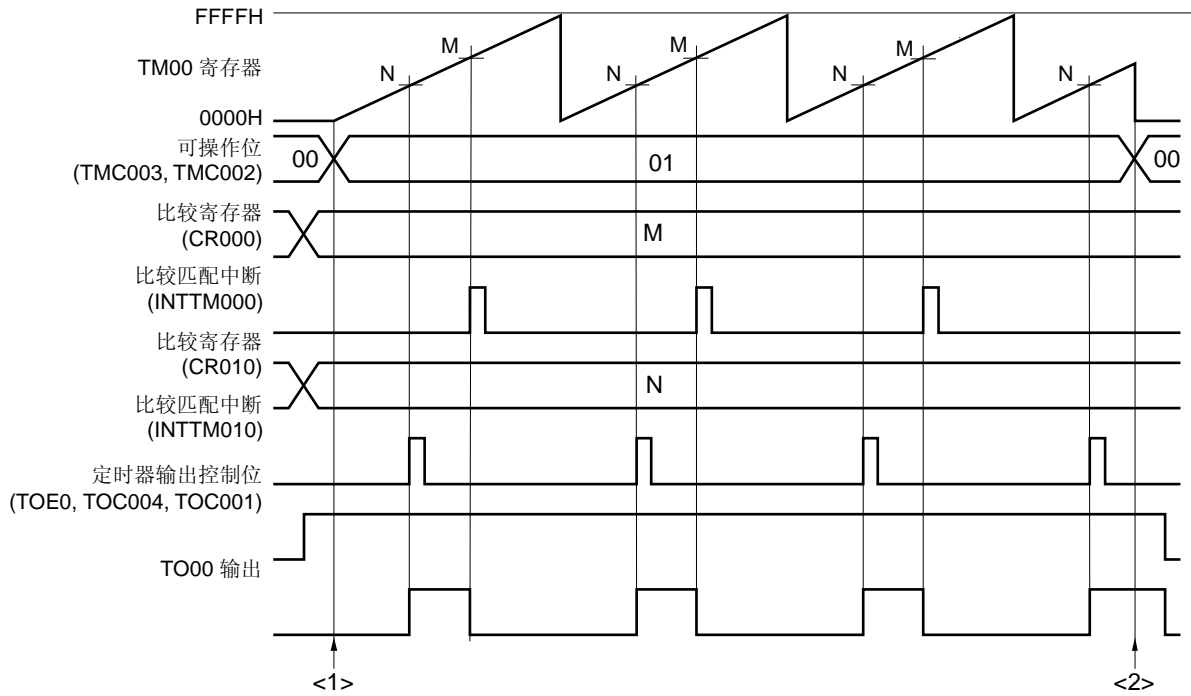
若要将该寄存器用作捕捉寄存器, 则应将 TI000 或 TI010 引脚选作捕捉触发。当检测到捕捉触发的有效沿时, TM00 的计数值将被保存在 CR000 中。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

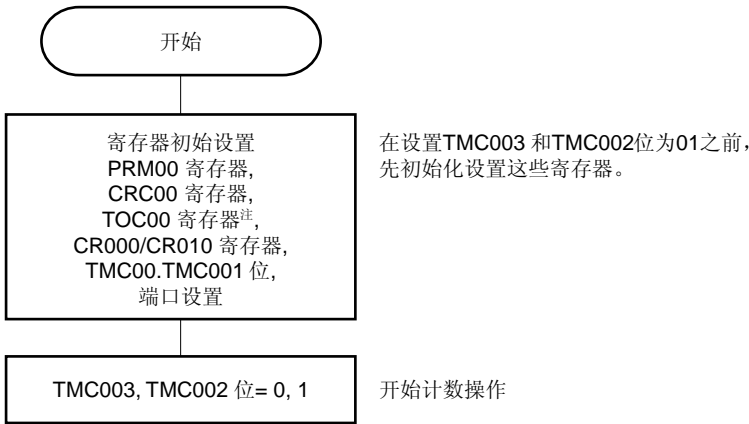
当该寄存器用作比较寄存器且该寄存器的值与 TM00 的计数值匹配时, 将产生一个中断信号 (INTTM010)。TM00 的计数值被清除。

当该寄存器被用作捕捉寄存器时, TI000 引脚输入将被用作捕捉触发。当检测到捕捉触发的有效沿时, TM00 的计数值将被保存在 CR010 中。

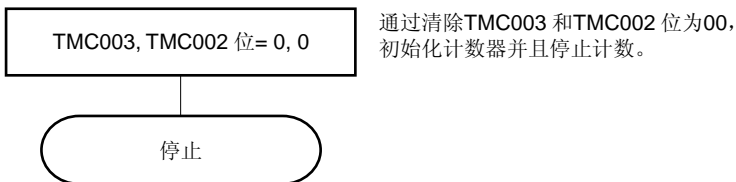
图 7-39. 自由运行定时器模式的软件处理示例



<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。关于详细信息，参见 7.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

7.4.6 PPG输出操作

当 16 位定时器模式控制寄存器 00 (TMC00) 设为 11 (TM00 与 CR000 匹配时的清零&启动) 时, 由 CR010 事先设定的脉冲宽度的方波作为 PPG 信号 (可编程脉冲发生器) 在 CR000 设定的周期内从 TO00 引脚输出。

作为 PPG 输出的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
- 占空比系数 = (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), 参见 7.5.1 TM00 操作期间重新写入 CR010。

- 备注**
1. 关于输入 / 输出引脚设置的详细信息, 参见 7.3 (5) 端口模式寄存器 0 (PM0)。
 2. 关于允许 INTTM000 信号中断的方法, 参见第十六章 中断功能。

图 7-40. PPG 输出操作的框图

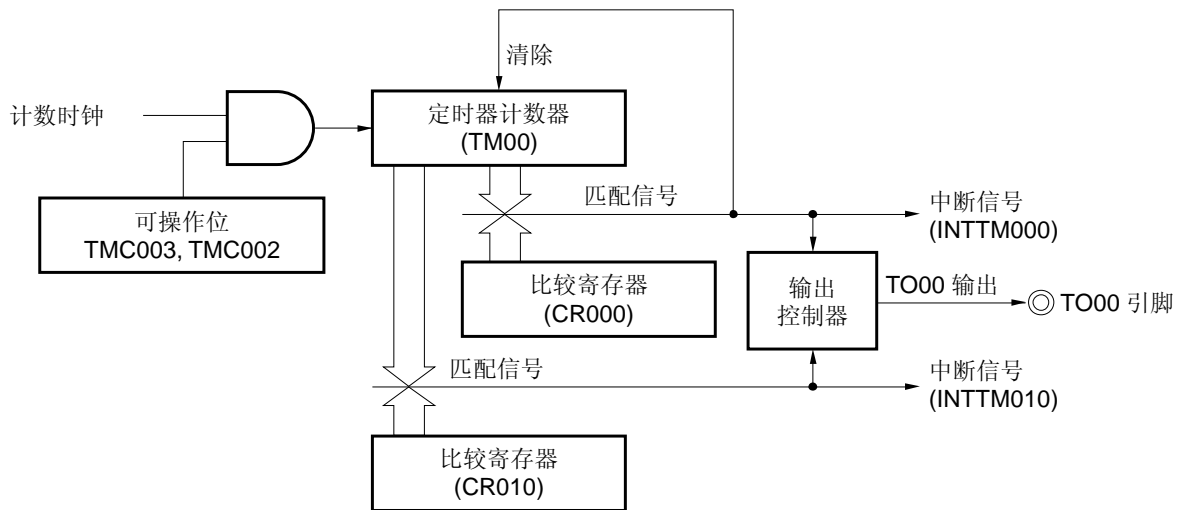
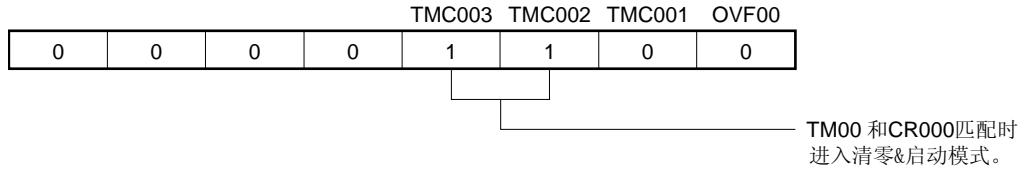
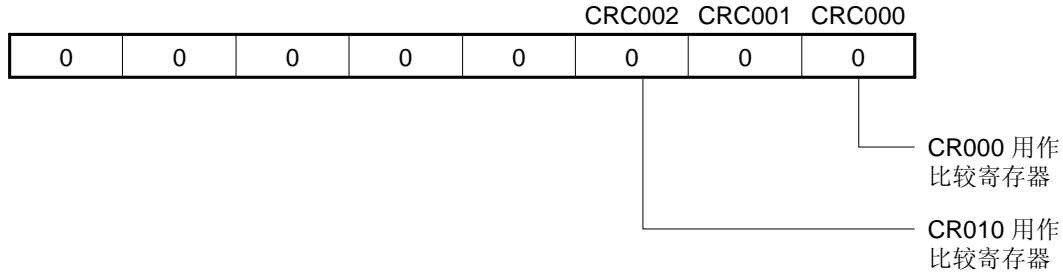


图 7-41. PPG 输出操作的寄存器设置示例

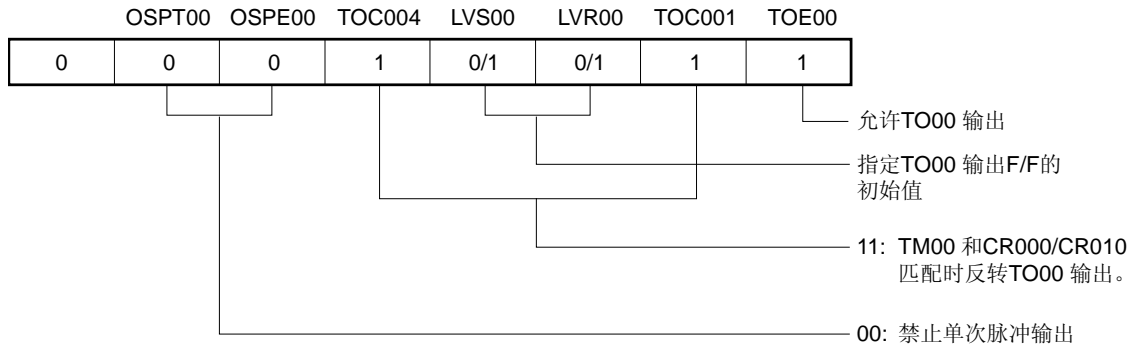
(a) 16 位定时器模式控制寄存器 00 (TMC00)



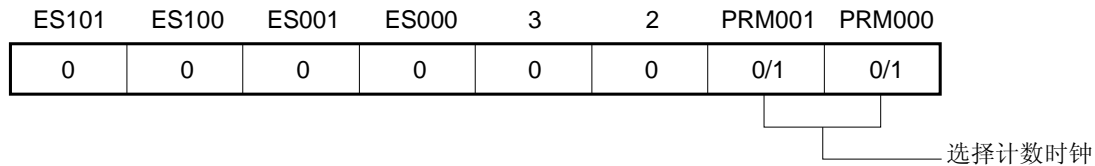
(b) 捕捉 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频模式寄存器 00 (PRM00)



(e) 16 位定时器计数器 00 (TM00)

计数值可以通过读取 TM00 获得。

(f) 16 位捕捉 / 比较寄存器 000 (CR000)

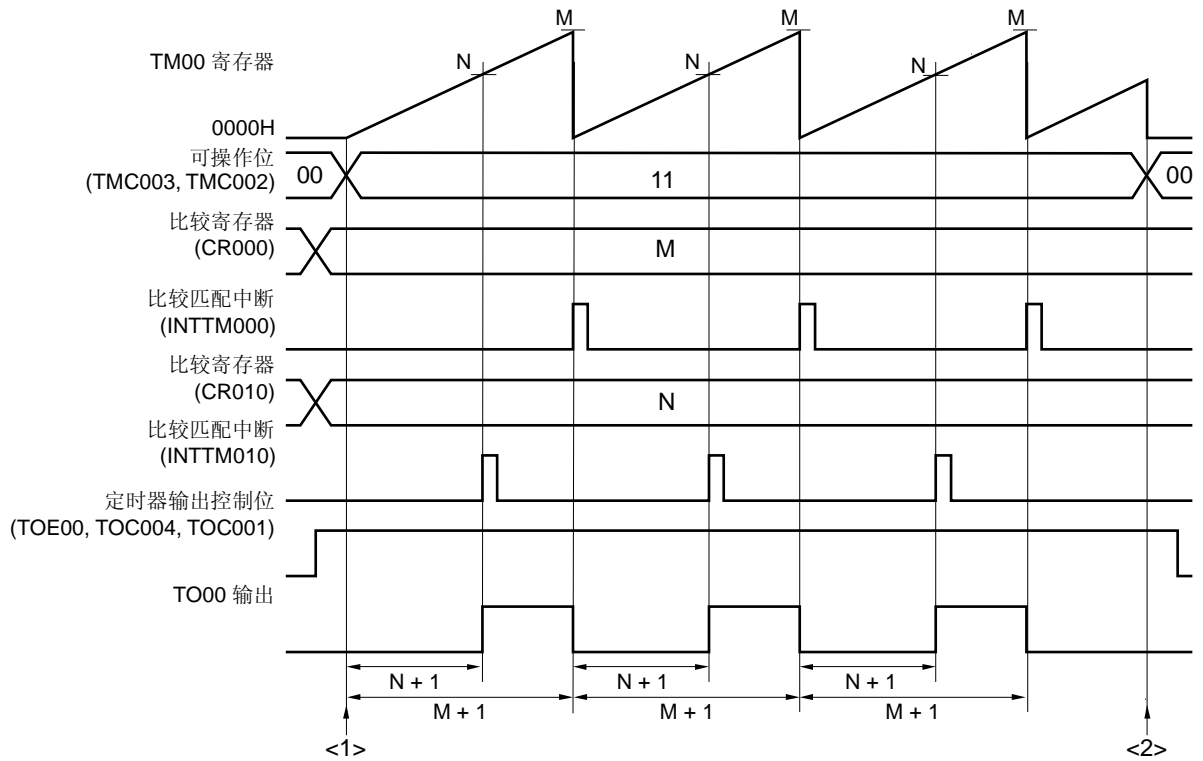
当寄存器的值与 TM00 的计数值匹配时将产生一个中断信号 (INTTM000)。TM00 的计数值清除。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

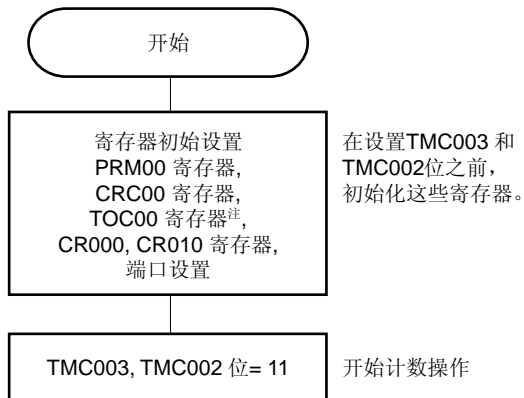
当该寄存器的值与 TM00 的计数值匹配时将产生一个中断信号 (INTTM010)。TM00 的计数值清除。

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

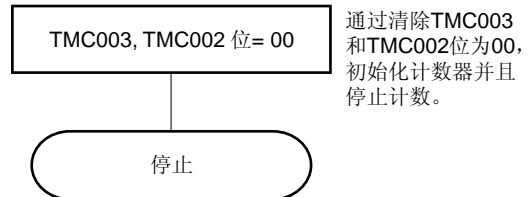
图 7-42. PPG 输出操作的软件处理示例



<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。关于详细信息，参见 7.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

备注 PWM 脉冲周期 = (M + 1) × 计数时钟周期
PPG 占空比系数 = (N + 1) / (M + 1)

7.4.7 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 引脚有效沿进入的清零&启动模式) 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, 将触发 TM00 的清零&启动模式, 且 TO00 引脚只输出一次反映 CR000 和 CR010 的值相差的脉冲。

- 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产生触发。
 2. 若只设置 OSPT00=1, 用作单脉冲输出, 则不要改变 TI000 引脚或其复用功能端口引脚的电平。否则, 将不能输出期望的脉冲。

- 备注**
1. 关于输入 / 输出引脚设置的详细信息, 参见 7.3 (5) 端口模式寄存器 0 (PM0)。
 2. 关于允许 INTTM000 信号中断的方法, 参见第十六章 中断功能。

图 7-43. 单脉冲输出操作的框图

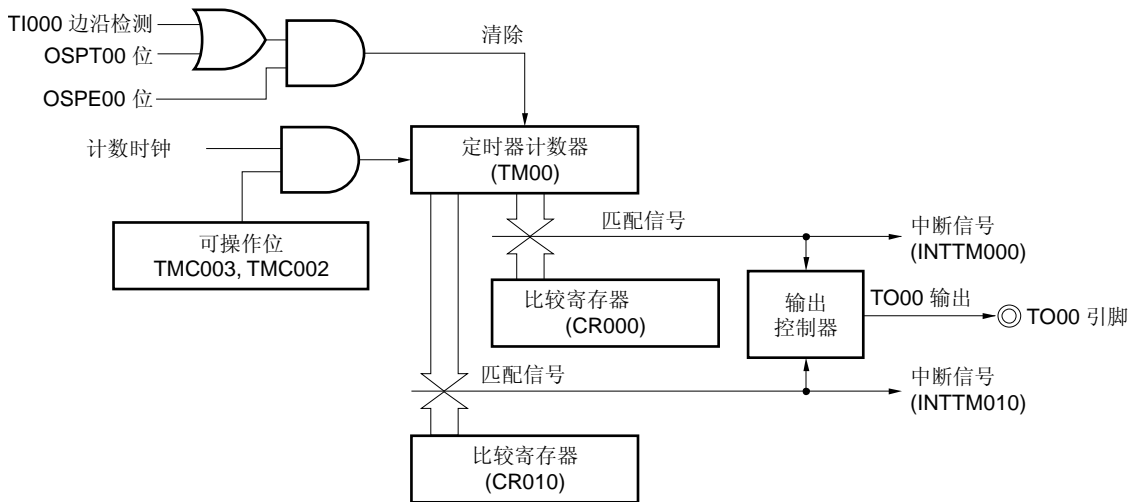
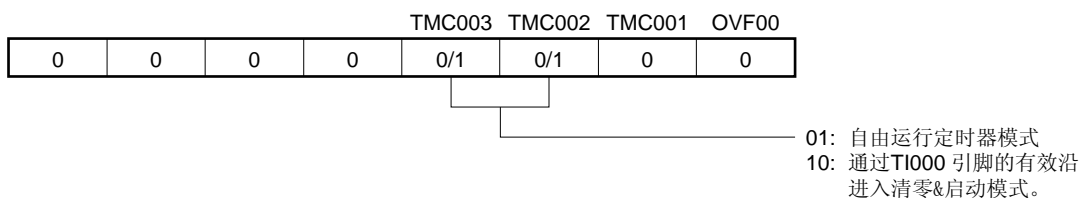
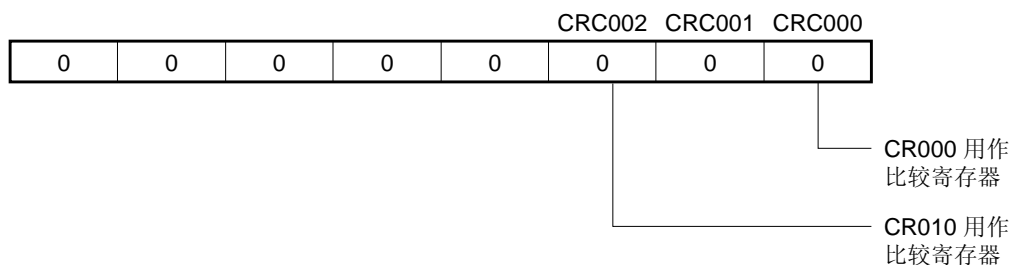


图 7-44. 单脉冲输出操作的寄存器设置示例 (1 / 2)

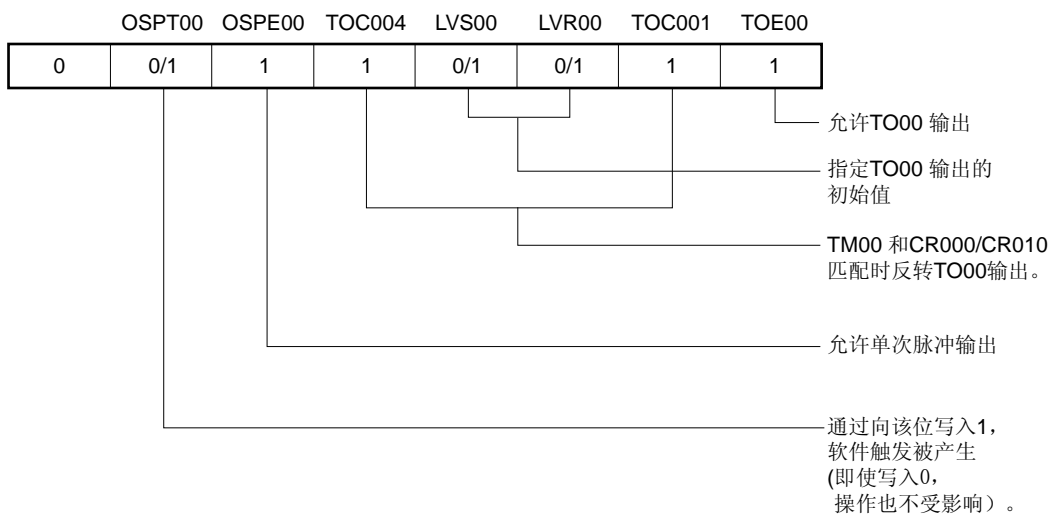
(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕捉 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频模式寄存器 00 (PRM00)

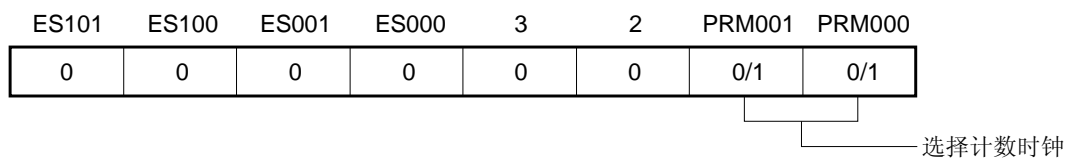


图 7-44. 单脉冲输出操作的寄存器设置示例 (2 / 2)

(e) 16 位定时器计数器 00 (TM00)

计数值可以通过读取 TM00 获得。

(f) 16 位捕捉 / 比较寄存器 000 (CR000)

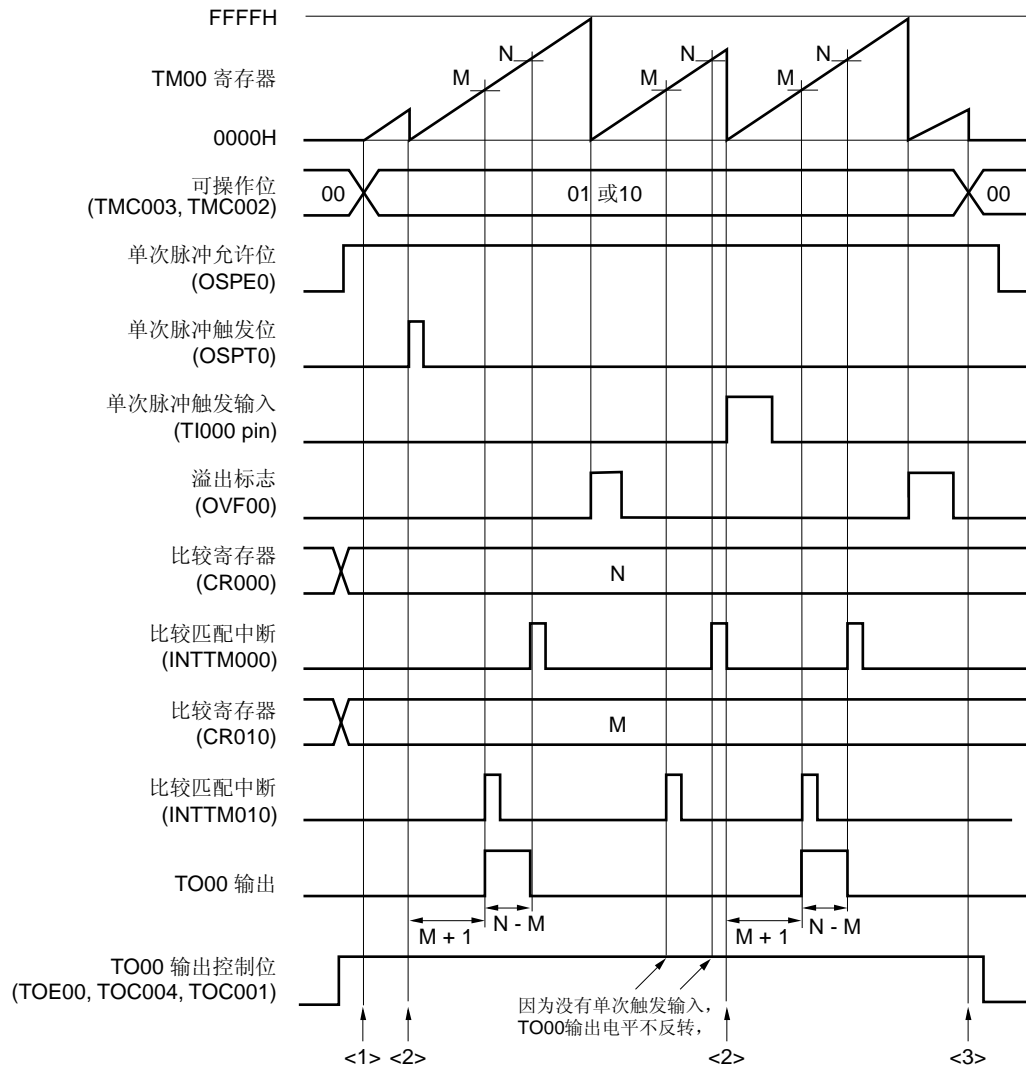
在输出单脉冲时，该寄存器用作比较寄存器。当 TM00 的值与 CR000 的值匹配时，将产生一个中断信号 (INTTM000)，且 TO00 的输出电平反转。

(g) 16 位捕捉 / 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，将产生一个中断信号 (INTTM010)，且 TO00 的输出电平反转。

注意事项 不要对 CR000 和 CR010 设置相同的值。

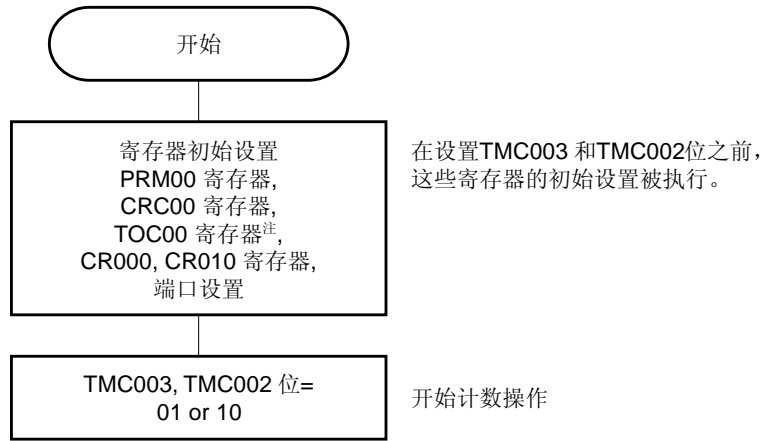
图 7-45. 单脉冲输出操作的软件处理示例 (1 / 2)



- 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 7-45. 单脉冲输出操作的软件处理示例 (2 / 2)

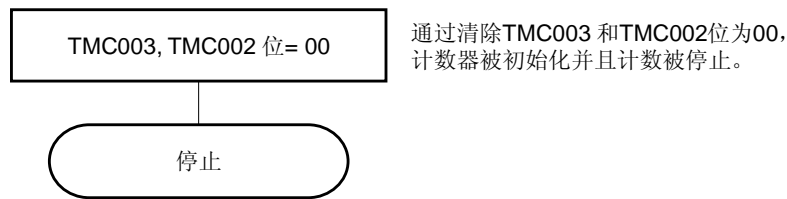
<1> 计数操作开始流程



<2> 单次触发输入流程



<3> 计数操作停止流程



注 对 TOC00 进行设置时必须需要谨慎。关于详细信息，参见 7.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

7.4.8 脉冲宽度测量操作

TM00 可以用于测量输入到 TI000 和 TI010 引脚信号的脉冲宽度。

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。

当产生中断时，将读取有效捕捉寄存器的值并测量脉冲宽度。检测 16 位定时器模式控制寄存器 00（TMC00）的 0 位（OVF00）。如果它设为 1，则通过软件将其清零。

图 7-46. 脉冲宽度测量（自由运行定时器模式）的框图

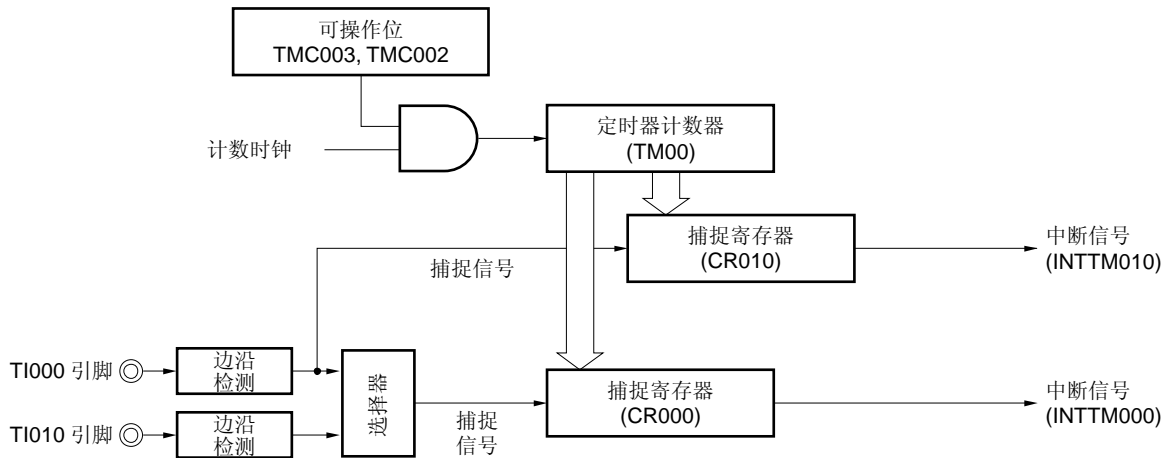
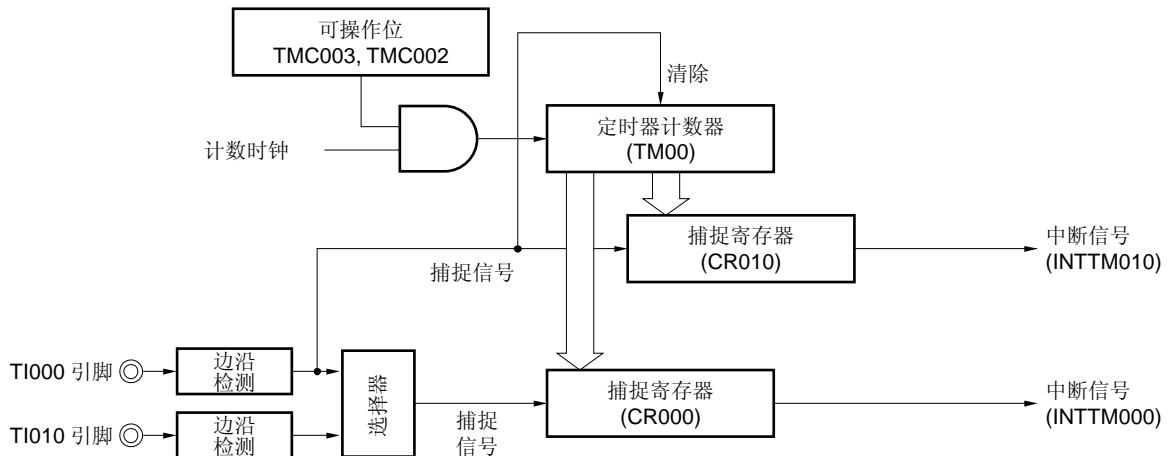


图 7-47. 脉冲宽度测量（通过 TI000 引脚有效沿输入进入的清零&启动模式）的框图



脉冲宽度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冲宽度（自由运行定时器模式）
- 通过使用 TI000 引脚的输入信号来测量脉冲宽度（自由运行定时器模式）
- 通过使用 TI000 引脚输入信号来测量脉冲宽度（通过 TI000 引脚有效沿输入进入的清零&启动模式）

备注

1. 关于输入 / 输出引脚设置的详细信息，参见 7.3 (5) 端口模式寄存器 0 (PM0)。
2. 关于允许 INTTM000 信号中断的方法，参见第十六章 中断功能。

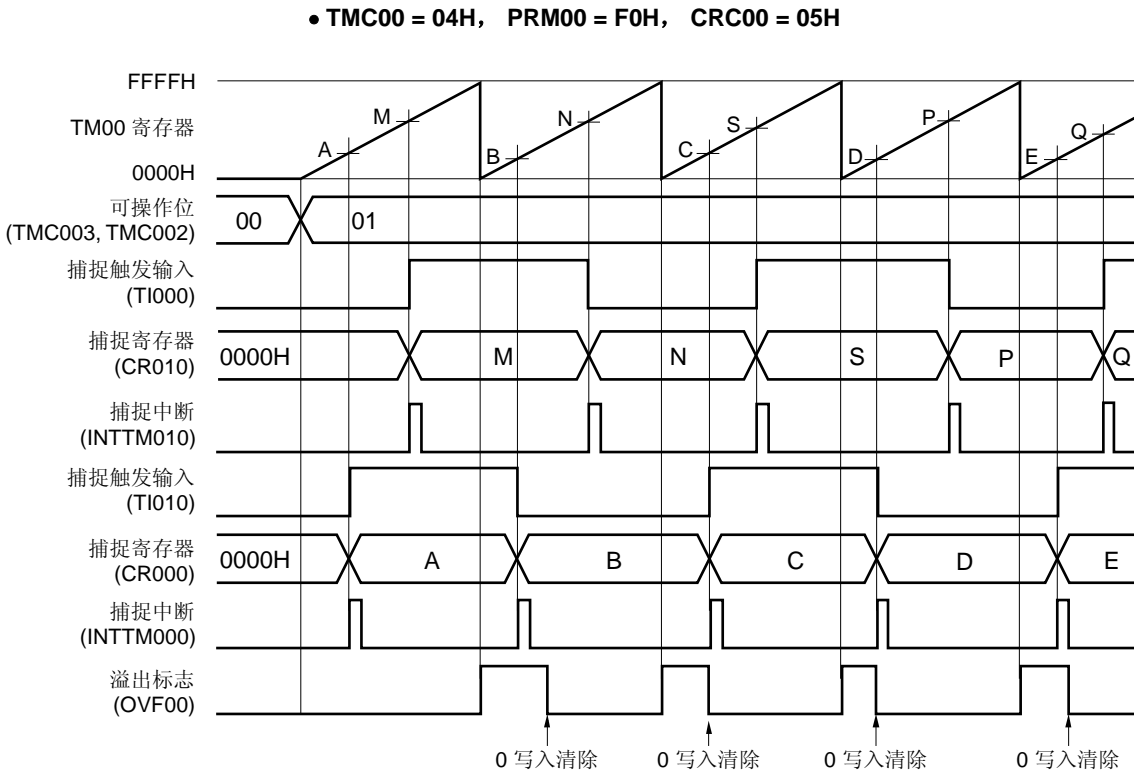
(1) 通过使用 TI000 和 TI010 引脚的两个输入信号来测量脉冲宽度（自由运行定时器模式）

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 的计数值将被捕捉至 CR010 中。当检测到 TI010 引脚的有效沿时，TM00 的计数值将被捕捉至 CR000 中。指定 TI000 和 TI010 引脚的双边沿检测。

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

发生溢出时，如果先前捕捉的值只是直接与当前捕捉的值相减，则值为负，也因此发生借位（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冲宽度。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 7-48. 脉冲宽度测量的时序示例 (1)



(2) 通过使用 TI000 引脚的一个输入信号来测量脉冲宽度 (自由运行定时器模式)

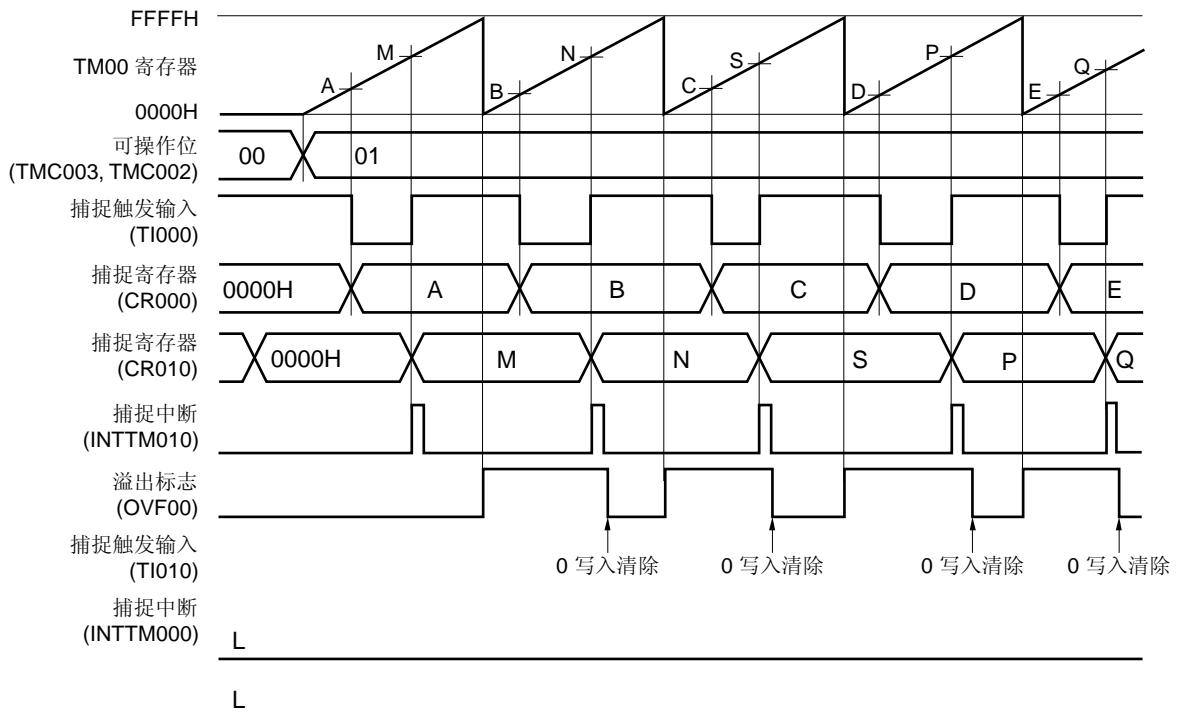
设置自由运行定时器模式 (TMC003 和 TMC002 = 01)。TM00 的计数值在与 TI000 引脚上检测的有效沿的相位相反的边沿中捕捉至 CR000。当检测到 TI000 引脚的有效沿时, TM00 的计数值将被捕捉至 CR010 中。

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捉寄存器中。因此, 不必保存捕捉值。通过将两个捕捉寄存器中的值相减, 可以计算高电平的宽度, 低电平的宽度, 以及周期。

发生溢出时, 如果捕捉的值只是直接与另一捕捉的值相减, 则值为负, 也因此发生借位 (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冲宽度。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 7-49. 脉冲宽度测量的时序示例 (2)

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



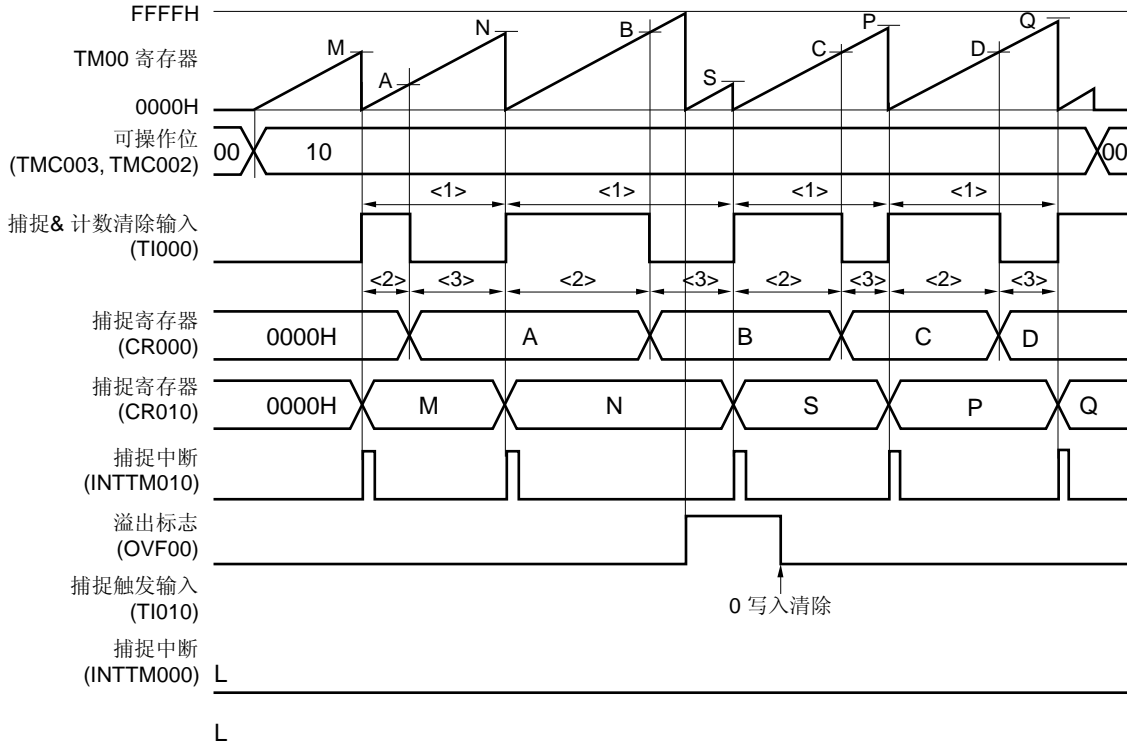
(3) 通过使用 TI000 引脚输入信号来测量脉冲宽度 (通过 TI000 引脚有效沿输入进入的清零&启动模式)

设置通过 TI000 引脚有效沿 (TMC003 和 TMC002 = 10) 进入的清零&启动模式。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捉至 CR000, TM00 的计数值则被捕捉至 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 7-50. 脉冲宽度测量的时序示例 (3)

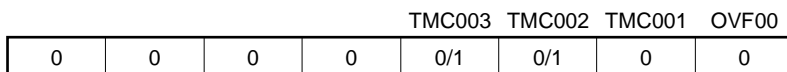
• TMC00 = 08H, PRM00 = 10H, CRC00 = 07H



- <1> 脉冲周期 = (10000H × OVF00 位被设为 1 的次数 + CR010 的捕捉值) × 计数时钟周期
- <2> 高电平脉冲宽度 = (10000H × OVF00 位被设为 1 的次数 + CR000 的捕捉值) × 计数时钟周期
- <3> 低电平脉冲宽度 = (脉冲周期 - 高电平脉冲宽度)

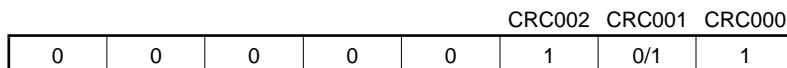
图 7-51. 脉冲宽度测量的寄存器设置示例 (1 / 2)

(a) 16 位定时器模式控制寄存器 00 (TMC00)



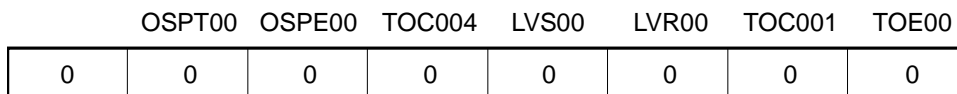
01: 自由运行定时器模式
10: 通过TI000 引脚的有效沿进入的清零&启动模式。

(b) 捕捉 / 比较控制寄存器 00 (CRC00)

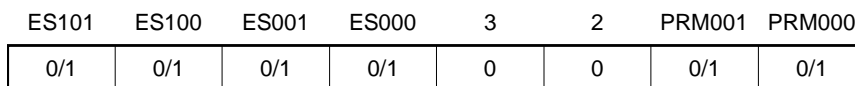


1: CR000 用作捕捉寄存器
0: TI010 引脚用作CR000的捕捉触发。
1: TI000 引脚的反相用作CR000的捕捉触发。
1: CR010 用作捕捉寄存器

(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
(设置 TI000 的有效沿被禁止)
00: 下降沿检测
01: 上升沿检测
10: 禁止设置
11: 双沿检测
(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: 双沿检测

图 7-51. 脉冲宽度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 获得。

(f) 16 位捕捉 / 比较寄存器 000 (CR000)

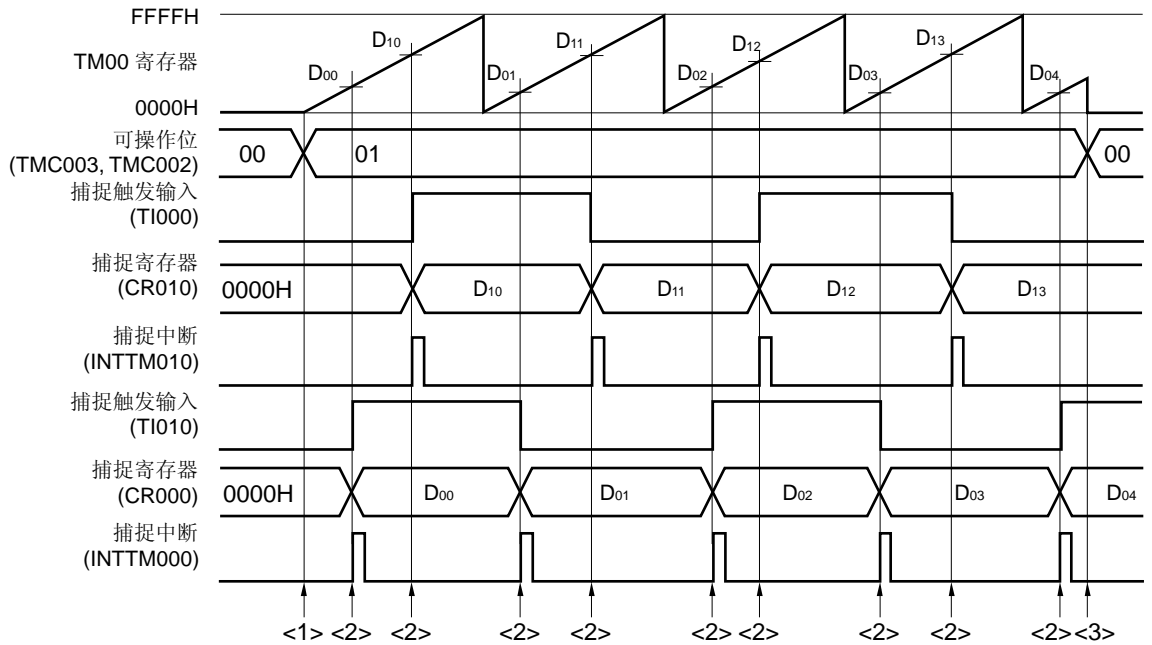
该寄存器用作捕捉寄存器。TI000 或 TI010 引脚可被选作捕捉触发。当检测到捕捉触发的指定沿时，TM00 的计数值将被保存在 CR000 中。

(g) 16 位捕捉 / 比较寄存器 010 (CR010)

该寄存器用作捕捉寄存器。输入到 TI000 引脚的信号被用作捕捉触发信号。当检测到捕捉触发时，TM00 的计数值将被保存在 CR010 中。

图 7-52. 脉冲宽度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零&启动模式示例

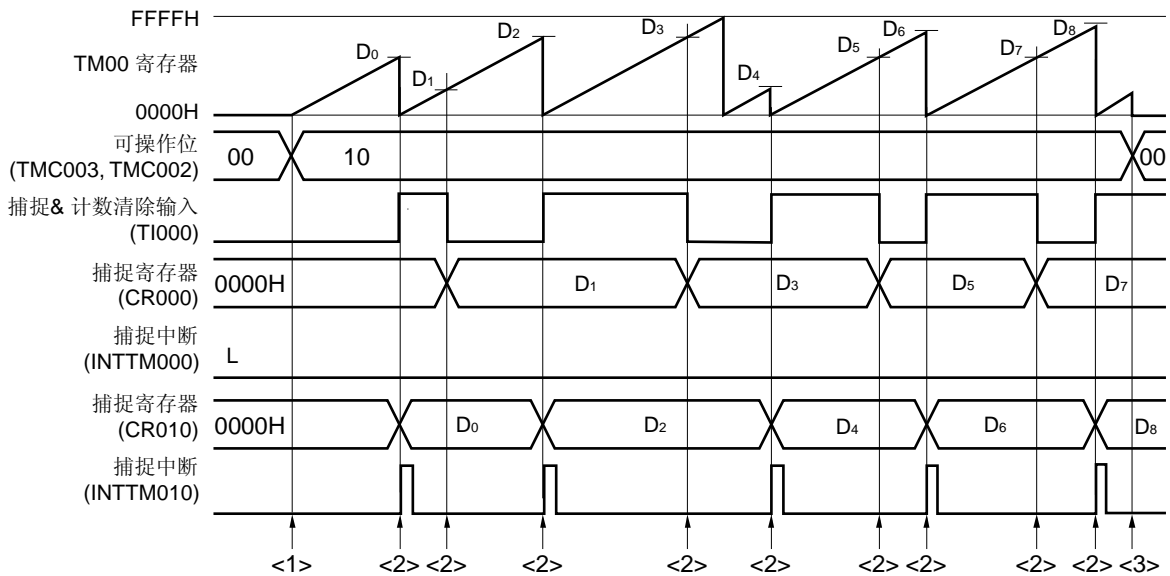
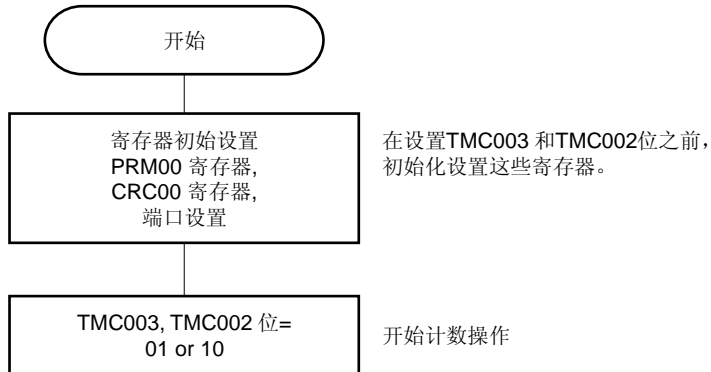
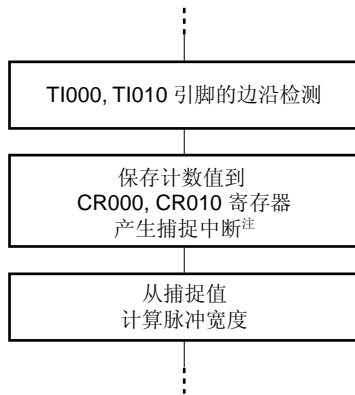


图 7-52. 脉冲宽度测量的软件处理示例 (2 / 2)

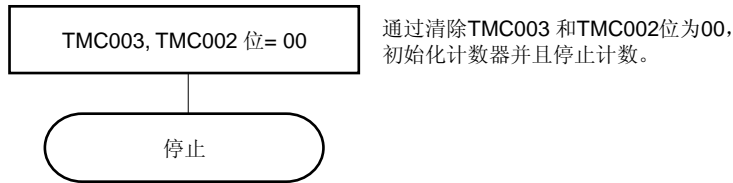
<1> 计数操作开始流程



<2> 捕捉触发输入流程



<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, 将不能产生捕捉中断信号 (INTTM000)。

7.5 TM00 的特殊用途

7.5.1 TM00 操作期间重新写入CR010

一般来说，在 μ PD78F8024 和 μ PD78F8025 产品中当 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新写入。

然而，如果 CR010 用作 PPG 输出并且更改了占空比系数，即使 TM00 正在操作，可以按下列过程修改 CR010 的值。（当改变 CR010 的值为小于当前值时，在它的值匹配 TM00 的值后立即重新写入它。当改变 CR010 的值为大于当前值时，在 CR000 的值匹配 TM00 的值后立即重新写入它。如果 CR010 的值在 CR010 和 TM00 匹配或 CR000 和 TM00 匹配前被重新写入，可能执行无法预料的操作。）

改变 CR010 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，禁止定时器输出反转。
- <3> 更改 CR010 的值。
- <4> 等待 TM00 一个计数时钟周期。
- <5> 当 TM00 的值与 CR010 的值（TOC004 = 1）匹配时，允许定时器输出反转。
- <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，参见第十六章 中断功能。

7.5.2 设置LVS00 和 LVR00

(1) LVS00 和 LVR00 的用途

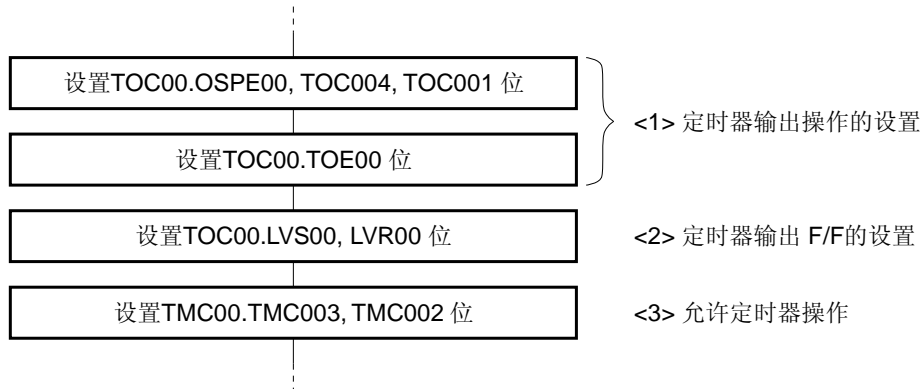
LVS00 和 LVR00 用于设置 TO00 输出的默认值，也用于在没有允许定时器操作（TMC003 和 TMC002 = 00）的情况下反转定时器输出。在不需要软件控制时将 LVS00 和 LVR00 清除为 00（默认值：低电平输出）。

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

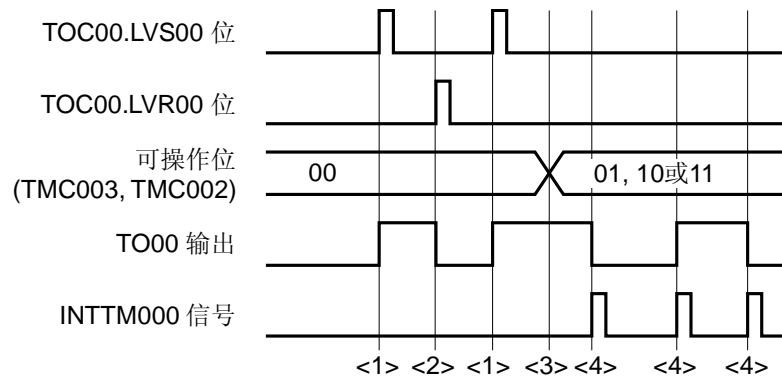
使用以下步骤来设置 LVS00 和 LVR00。

图 7-53. 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 7-54. LVR00 和 LVS00 的时序示例



- <1> 当 LVS00 和 LVR00 = 10 时 TO00 输出高电平。
- <2> 当 LVS00 和 LVR00 = 01 时 TO00 输出低电平（即使 LVS00 和 LVR00 被清除为 00，引脚输出也会从高水平开始保持不变）。
- <3> 当 TMC003 和 TMC002 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 从高水平处开始输出。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> 每次产生中断信号（INTTM00）时，TO00 的输出电平反转。

7.6 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 7-3 显示了对于各个通道的限制。

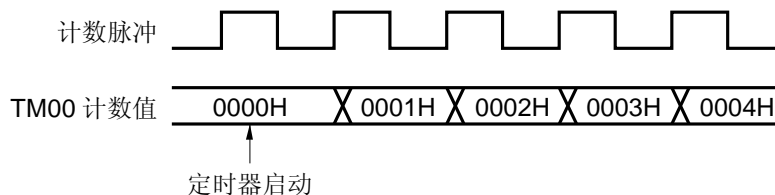
表 7-3. 16 位定时器 / 事件计数器 00 各个通道的限制

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 引脚有效沿输入进入的清零&启动模式	当检测 TI010 引脚的有效沿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 ≤ CR010 < CR000 ≤ FFFFH
用作单脉冲输出	禁止对 CR000 和 CR010 设置相同的值。
用作脉冲宽度测量	禁止使用定时器输出 (TO00) (TOC00 = 00H)。

(2) 定时器启动时的错误

在定时器启动后，产生一个匹配信号可能发生近一个时钟周期的误差。这是由于启动 TM00 计数与计数脉冲不同步引起的。

图 7-55. TM00 计数的开始时序



(3) CR000 和 CR010 的设置 (TM00 与 CR000 匹配时进入的清零&启动模式)

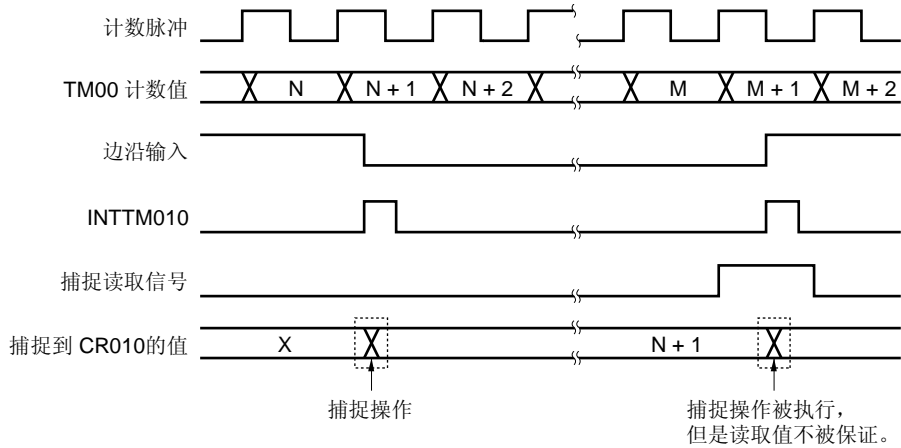
向 CR000 和 CR010 中设定一个不同于 0000H 的值 (当 TM00 用作外部事件计数器时 TM00 不能对单脉冲进行计数)。

(4) 通过捕捉寄存器保存数据的时序

(a) 如果在读取 CR000 / CR010 时，有效沿被输入到 TI000 / TI010 引脚中并检测到 TI000 引脚的反相沿时，CR010 将执行捕捉操作，但 CR000 / CR010 的读取值将不能保证。此时，当检测到 TI000 / TI010 引脚的有效沿时将产生中断信号 (INTTM000 / INTTM010) (在检测到 TI000 引脚的反相边沿时将不会产生中断信号)。

当计数值由于检测到 TI000 / TI010 引脚的有效沿而被捕捉时，则将在产生 INTTM000 / INTTM010 后读取 CR000 / CR010 的值。

图 7-56. 通过捕捉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产生。在必须在当前有效电平输出后输入下一个触发信号。

(7) OVF00 标志的操作

(a) 设置 OVF00 标志 (1)

在以下情况中 TM00 溢出，OVF00 标志会置为 1。

选择在 TM00 与 CR000 匹配的情况下进入的清零&启动模式。

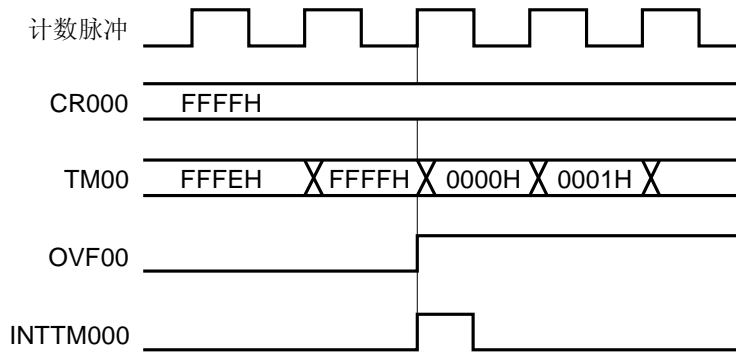
↓

将 CR000 设为 FFFFH。

↓

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

图 7-57. 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

在自由运行定时器模式或通过 TI000 引脚有效沿进入的清零&启动模式下，单脉冲输出能正常操作。单脉冲不能在 TM00 与 CR000 匹配的情况下进入的清零&启动模式中输出。

(9) 捕捉操作**(a) 当 TI000 的有效沿被指定为计数时钟时**

当 TI000 的有效沿作为计数时钟时，指定 TI000 为触发的捕捉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捉值的脉冲宽度

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

(c) 中断信号的产生

捕捉操作在计数时钟的上升沿处进行，但中断信号（INTTM000 和 INTTM010）则在下一个计数时钟的上升沿产生（参见图 7-7）。

(d) 当 CRC001（捕捉 / 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

在 TI000 引脚输入信号相反的相位下将 TM00 寄存器的计数值捕捉至 CR000 寄存器时，捕捉到计数值后将不会产生中断请求信号（INTTM000）。如果在该操作过程中在 TI010 引脚上检测到有效沿，那么将不会执行捕捉操作，但会产生一个 INTTM000 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，一旦操作停止后又允许操作，则不能检测到上升沿。

(b) 消除噪声的采样时钟

根据 TI000 有效沿是用作计数时钟还是捕捉时钟，用于消除噪声的采样时钟会不同。在前面一种情况中，采样时钟固定为 f_{PRS} 。在后面一种情况中，由 PRM00 选择的计数时钟用于采样。

当输入到 TI000 引脚中的信号被采样且连续检测到两次有效电平时，认为检测到有效沿。因此，可以消除具有短脉冲宽度的噪声（参见图 7-7）。

(11) 定时器操作

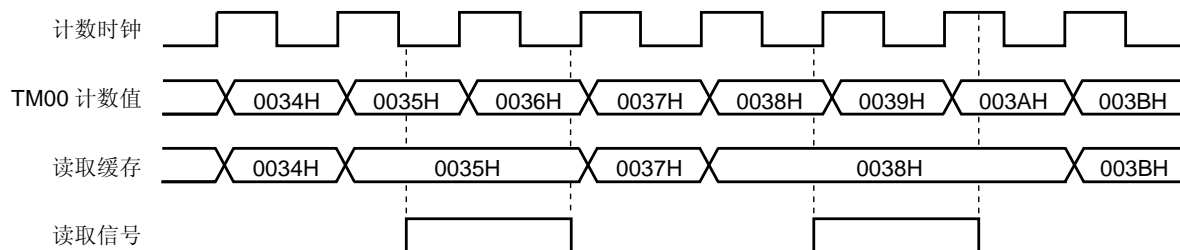
当定时器停止时，不管 CPU 为何种操作模式，不能响应 TI000 / TI010 引脚的输入信号。

备注 f_{PRS} : 外围硬件时钟频率

(12) 读取 16 位定时器计数器 00 (TM00)

可以在没有停止实际计数器时读取 TM00，因为它被读取时捕捉到缓存的计数值是固定的。然而，当计数器向上计数前被读取时，缓存可能没有被更新，因为缓存在计数器向上计数时被更新。

图 7-58. 16 位定时器计数器 00 (TM00) 读取时序



第八章 8 位定时器 / 事件计数器 50 和 51

8.1 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 有以下功能。

- 间隔定时器
- 外部事件计数器
- 方波输出
- PWM 输出

注意事项 8 位定时器 / 事件计数器 50 和 51 和 8 位定时器 H0 和 H1 的输出 (TO50、TO51、TOH0 和 TOH1) 在内部分别被连接到恒定电流驱动的 PWM3、PWM0、PWM1 和 PWM2 引脚。要输入与恒定电流驱动的 PWM0 到 PWM3 引脚相同频率的 PWM 信号，按照下面设置每个计数器的计数时钟。

定时器	计数时钟选择	PWM 频率	
		$f_{PRS} = 4 \text{ MHz}$	$f_{PRS} = 8 \text{ MHz}$
TM50	$f_{PRS}/2^6$	244 Hz	488 Hz
TM51			
TMH0			
TMH1			

8.2 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 包含以下硬件。

表 8-1. 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 1 (PM1) 或者端口模式寄存器 3 (PM3) 端口寄存器 1 (P1) 或者端口寄存器 3 (P3)

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

图 8-1. 8 位定时器 / 事件计数器 50 的框图

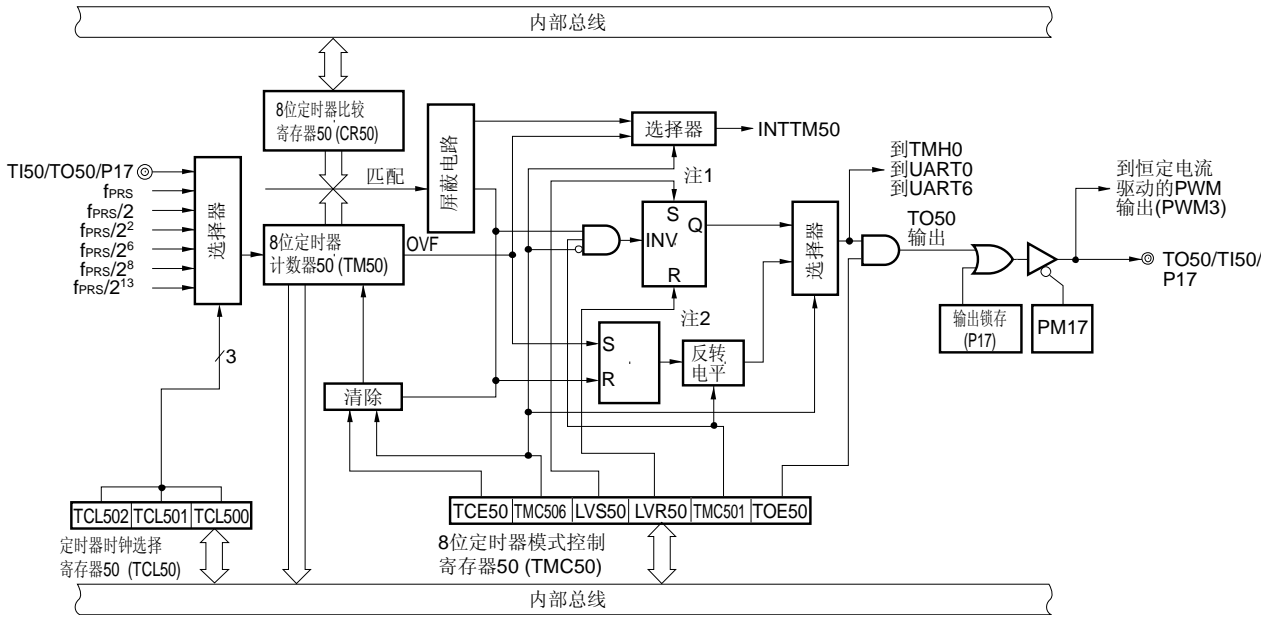
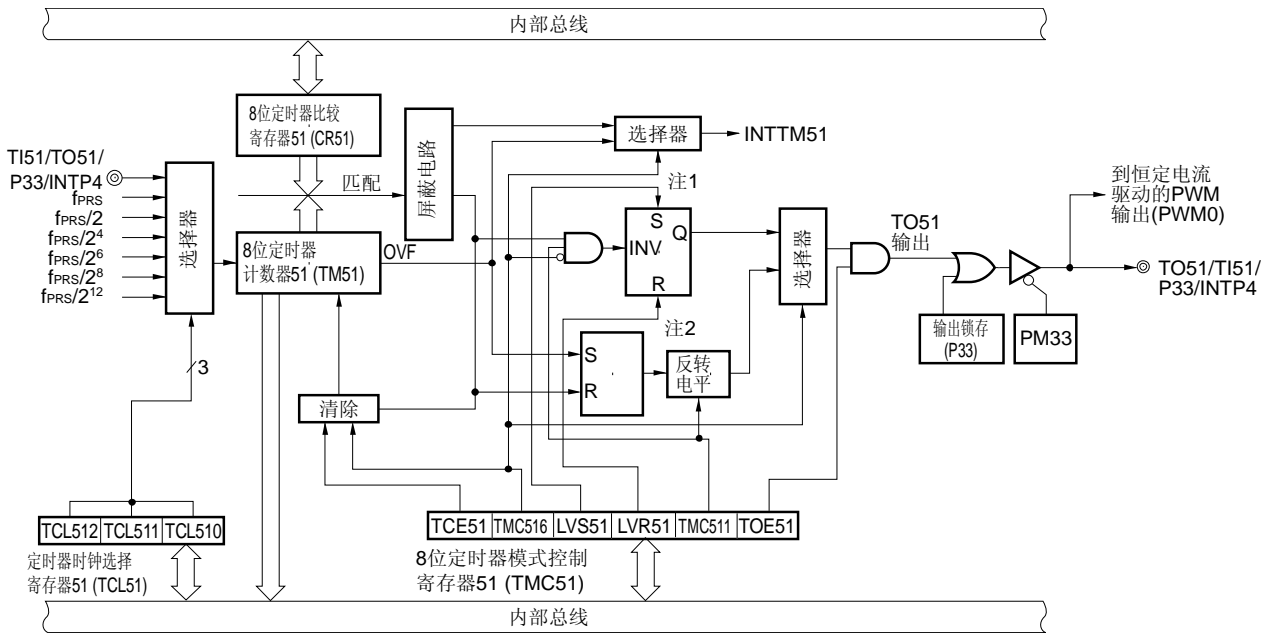


图 8-2. 8 位定时器 / 事件计数器 51 的框图



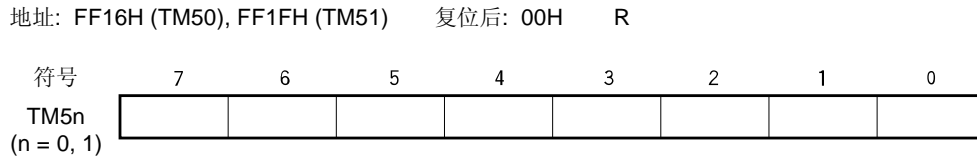
- 注
1. 定时器输出 F/F
 2. PWM 输出 F/F

(1) 8 位定时器计数器 5n (TM5n)

TM5n 是一个对计数脉冲计数的 8 位只读寄存器。

计数器按照计数时钟的上升沿同步递增。

图 8-3. 8 位定时器计数器 5n (TM5n) 的格式



在以下情况下，计数值被清除为 00H。

- <1> 复位信号产生
- <2> 当 TCE5n 被清除时
- <3> 在 TM5n 和 CR5n 匹配时清零&启动的模式下，TM5n 和 CR5n 匹配时

(2) 8 位定时器比较寄存器 5n (CR5n)

CR5n 可以由 8 位存储器操作指令读取和写入。

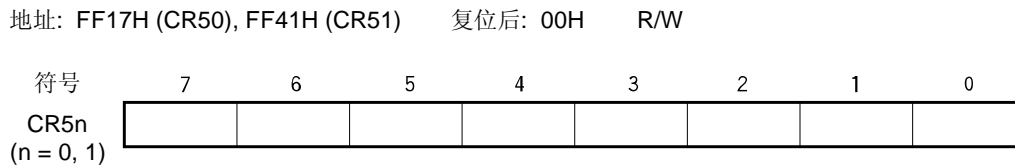
除非在 PWM 模式下，CR5n 中设置的值与 8 位定时器计数器 5n (TM5n) 的计数值比较，如果匹配，将产生中断请求 (INTTM5n)。

在 PWM 模式下，当 TM5n 和 CR5n 的值匹配时，TO5n 输出无效电平，但是不会产生中断。

可以在 00H 到 FFH 范围内设置 CR5n 的值。

复位信号的产生将清除 CR5n 为 00H。

图 8-4. 8 位定时器比较寄存器 5n (CR5n) 的格式



- 注意事项**
1. 在 TM5n 和 CR5n 匹配 (TMC5n6 = 0) 时清零&启动模式下，不要在操作期间向 CR5n 中写入其它值。
 2. 在 PWM 模式下，使 CR5n 的重新写入周期为 3 个计数时钟 (由 TCL5n 选择的时钟) 或更多。

备注 n = 0, 1

8.3 控制 8 位定时器 / 事件计数器 50 和 51 的寄存器

以下四个寄存器用来控制 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 1 (PM1) 或者端口模式寄存器 3 (PM3)
- 端口寄存器 1 (P1) 或者端口寄存器 3 (P3)

(1) 定时器时钟选择寄存器 5n (TCL5n)

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

TCL5n 可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生将清除 TCL5n 为 00H。

备注 n = 0, 1

图 8-5. 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 ^{注1}				
			f _{PRS} = 4 MHz	f _{PRS} = 8 MHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz	
0	0	0	TI50 引脚下降沿 ^{注2}				
0	0	1	TI50 引脚上升沿 ^{注2}				
0	1	0	f _{PRS} ^{注3}	4 MHz	8 MHz	10 MHz	20 MHz ^{注4}
0	1	1	f _{PRS} /2	2 MHz	4 MHz	5 MHz	10 MHz
1	0	0	f _{PRS} /2 ²	1 MHz	2 MHz	2.5 MHz	5 MHz
1	0	1	f _{PRS} /2 ⁶	62.5 kHz	125 kHz	156.25 kHz	312.5 kHz
1	1	0	f _{PRS} /2 ⁸	15.63 kHz	31.25 kHz	39.06 kHz	78.13 kHz
1	1	1	f _{PRS} /2 ¹³	0.49 kHz	0.98 kHz	1.22 kHz	2.44 kHz

- 注**
- 如果外围硬件时钟 (f_{PRS}) 工作于高速系统时钟 (f_{xH}) (XSEL = 1), 根据供电电压改变 f_{PRS} 工作频率。
 - V_{DD} = 2.7 到 5.5 V: f_{PRS} ≤ 20 MHz
 - V_{DD} = 1.8 到 2.7 V: f_{PRS} ≤ 5 MHz
 - 在 STOP 模式下, 不要使用从 TI50 引脚输入的外部时钟启动定时器操作。
 - 如果外围硬件时钟 (f_{PRS}) 工作于内部高速振荡时钟 (f_{RH}) (XSEL = 0), 当 1.8 V ≤ V_{DD} < 2.7 V 时, 禁止设置 TCL502、TCL501、TCL500 = 0、1、0 (计数时钟: f_{PRS})。
 - 只有在 4.0 V ≤ V_{DD} ≤ 5.5 V 的情况下可以设置。

- 注意事项**
- 当向 TCL50 重新写入其它数据时, 先停止定时器操作。
 - 确保清除第 3 到 7 位为 “0”。
 - 8 位定时器 / 事件计数器 50 和 51 和 8 位定时器 H0 和 H1 的输出 (TO50、TO51、TOH0 和 TOH1) 在内部分别被连接到恒定电流驱动的 PWM3、PWM0、PWM1 和 PWM2 引脚。要输入与恒定电流驱动的 PWM0 到 PWM3 引脚相同频率的 PWM 信号, 按照下面设置每个计数器的计数时钟。

定时器	计数时钟选择	PWM 频率	
		f _{PRS} = 4 MHz	f _{PRS} = 8 MHz
TM50	f _{PRS} /2 ⁶	244 Hz	488 Hz
TM51			
TMH0			
TMH1			

备注 f_{PRS}: 外围硬件时钟频率

图 8-6. 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 ^{注1}				
			f _{PRS} = 4 MHz	f _{PRS} = 8 MHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz	
0	0	0	TI51 引脚下降沿 ^{注2}				
0	0	1	TI51 引脚上升沿 ^{注2}				
0	1	0	f _{PRS} ^{注3}	4 MHz	8 MHz	10 MHz	20 MHz ^{注4}
0	1	1	f _{PRS} /2	2 MHz	4 MHz	5 MHz	10 MHz
1	0	0	f _{PRS} /2 ⁴	250 kHz	500 kHz	625 kHz	1.25 MHz
1	0	1	f _{PRS} /2 ⁶	62.5 kHz	125 kHz	156.25 kHz	312.5 kHz
1	1	0	f _{PRS} /2 ⁸	15.63 kHz	31.25 kHz	39.06 kHz	78.13 kHz
1	1	1	f _{PRS} /2 ¹²	0.98 kHz	1.95 kHz	2.44 kHz	4.88 kHz

- 注
- 如果外围硬件时钟 (f_{PRS}) 工作于高速系统时钟 (f_{XH}) (XSEL = 1), 根据供电电压改变 f_{PRS} 工作频率。
 - V_{DD} = 2.7 到 5.5 V: f_{PRS} ≤ 20 MHz
 - V_{DD} = 1.8 到 2.7 V: f_{PRS} ≤ 5 MHz
 - 在 STOP 模式下, 不要使用从 TI51 引脚输入的外部时钟启动定时器操作。
 - 如果外围硬件时钟 (f_{PRS}) 工作于内部高速振荡时钟 (f_{RH}) (XSEL = 0), 当 1.8 V ≤ V_{DD} < 2.7 V 时, 禁止设置 TCL512、TCL511、TCL510 = 0、1、0 (计数时钟: f_{PRS})。
 - 只有在 4.0 V ≤ V_{DD} ≤ 5.5 V 的情况下可以设置。

- 注意事项
- 当向 TCL51 重新写入其它数据时, 先停止定时器操作。
 - 确保清除第 3 到 7 位为 “0”。
 - 8 位定时器 / 事件计数器 50 和 51 和 8 位定时器 H0 和 H1 的输出 (TO50、TO51、TOH0 和 TOH1) 在内部分别被连接到恒定电流驱动的 PWM3、PWM0、PWM1 和 PWM2 引脚。要输入与恒定电流驱动的 PWM0 到 PWM3 引脚相同频率的 PWM 信号, 按照下面设置每个计数器的计数时钟。

定时器	计数时钟选择	PWM 频率	
		f _{PRS} = 4 MHz	f _{PRS} = 8 MHz
TM50	f _{PRS} /2	244 Hz	488 Hz
TM51			
TMH0			
TMH1			

备注 f_{PRS}: 外围硬件时钟频率

(2) 8 位定时器模式控制寄存器 5n (TMC5n)

TMC5n 执行以下五种类型的设置。

- <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 (触发器) 状态设置
- <4> 定时器 F/F 控制或 PWM (自由运行) 模式下的有效电平选择
- <5> 定时器输出控制

TMC5n 可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生将清除该寄存器为 00H。

备注 n = 0, 1

图 8-7. 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[※]

	<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, 计数操作禁止 (计数器停止)						
	1	计数操作开始						
	TMC506		TM50 操作模式选择					
	0	TM50 和 CR50 匹配时清零&启动的模式						
	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 清零 (0) (TO50 的默认值: 低电平)					
	1	0	定时器输出 F/F 置位 (1) (TO50 的默认值: 高电平)					
	1	1	禁止设置					
	TMC501		在其它模式下 (TMC506 = 0)			在 PWM 模式下 (TMC506 = 1)		
			定时器 F/F 控制			有效电平选择		
	0	反转操作禁止			高电平有效			
	1	反转操作允许			低电平有效			
	TOE50		定时器输出控制					
	0	输出禁止 (TO50 输出低电平)						
	1	输出允许						

注 第 2 和 3 位只能写入。

(注意事项和备注见下页。)

图 8-8. 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, 计数操作禁止 (计数器停止)							
1	计数操作开始							
TMC516	TM51 操作模式选择							
0	TM51 和 CR51 匹配时产生清零&启动的模式							
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 清零 (0) (TO51 的默认值: 低电平)						
1	0	定时器输出 F/F 置位 (1) (TO51 的默认值: 高电平)						
1	1	禁止设置						
TMC511	在其它模式下 (TMC516 = 0)				在 PWM 模式下 (TMC516 = 1)			
	定时器 F/F 控制				有效电平选择			
0	反转操作禁止				高电平有效			
1	反转操作允许				低电平有效			
TOE51	定时器输出控制							
0	输出禁止 (TO51 输出低电平)							
1	输出允许							

注 第 2 和 3 位只能写入。

- 注意事项
- 在 PWM 模式以外的模式下, LVS5n 和 LVR5n 的设置有效。
 - 按照下面的顺序执行 <1> 到 <4>, 而不是同时执行。
 - <1> 设置 TMC5n1, TMC5n6: 操作模式设置
 - <2> 设置 TOE5n 以允许输出: 定时器输出允许
 - <3> 设置 LVS5n, LVR5n (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 的其它位。
 - 实际 TO50 / TI50 / P17 和 TO51 / TI51 / P33 / INTP4 引脚的输出根据 PM17 和 P17 以及 PM33 和 P33, 以及 TO5n 的输出确定。

- 备注
- 在 PWM 模式下, 清除 TCE5n 为 0, PWM 输出变为无效。
 - 如果读取 LVS5n 和 LVR5n, 值为 0。
 - TMC5n6、LVS5n、LVR5n、TMC5n1 和 TOE5n 位的值从 TO5n 的输出反映出来, 而与 TCE5n 的值无关。
 - n = 0, 1

(3) 端口模式寄存器 1 和 3 (PM1, PM3)

该寄存器以 1 位为单位设置端口 1 和 3 的输入 / 输出。

当使用 P17 / TO50 / TI50 和 P33 / TO51 / TI51 / INTP4 引脚作为定时器输出时, 清除 PM17 和 PM33 以及 P17 和 P33 的锁存器为 0。

当使用 P17 / TO50 / TI50 和 P33 / TO51 / TI51 / INTP4 引脚作为定时器输入时, 设置 PM17 和 PM33 为 1。P17 和 P33 的锁存器这时可能为 0 或 1。

PM1 和 PM3 可以由 1 位或 8 位存储器操作指令来设置。

复位信号设置该寄存器为 FFH。

图 8-9. 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 复位后: FFH R / W

符号	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10
PM1n	P1n 引脚 I / O 模式选择 (n = 0 到 7)							
0	输出模式 (输出缓冲器开)							
1	输入模式 (输出缓冲器关)							

图 8-10. 端口模式寄存器 3 (PM3) 的格式

地址: FF23H 复位后: FFH R / W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30
PM3n	P3n 引脚 I / O 模式选择 (n = 0 到 3)							
0	输出模式 (输出缓冲器开)							
1	输入模式 (输出缓冲器关)							

8.4 8 位定时器 / 事件计数器 50 和 51 的操作

8.4.1 作为间隔定时器工作

8 位定时器 / 事件计数器 5n 作为间隔定时器工作，按照预先设置到 8 位定时器比较寄存器 5n (CR5n) 中的计数值的间隔重复产生中断请求。

当 8 位定时器计数器 5n (TM5n) 与设置到 CR5n 中的值匹配时，TM5n 的值清零并且产生中断请求信号 (INTTM5n)，继续计数。

TM5n 的计数时钟由定时器时钟选择寄存器 5n (TCL5n) 的第 0 到 2 位 (TCL5n0 到 TCL5n2) 选择。

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 和 CR5n 匹配时产生清零&启动的模式。
(TMC5n = 0000xxx0B x = 无关)

<2> 设置 TCE5n = 1 后，开始计数操作。

<3> 如果 TM5n 和 CR5n 的值匹配，产生 INTTM5n (TM5n 清除为 00H)。

<4> INTTM5n 按照相同的间隔重复产生。

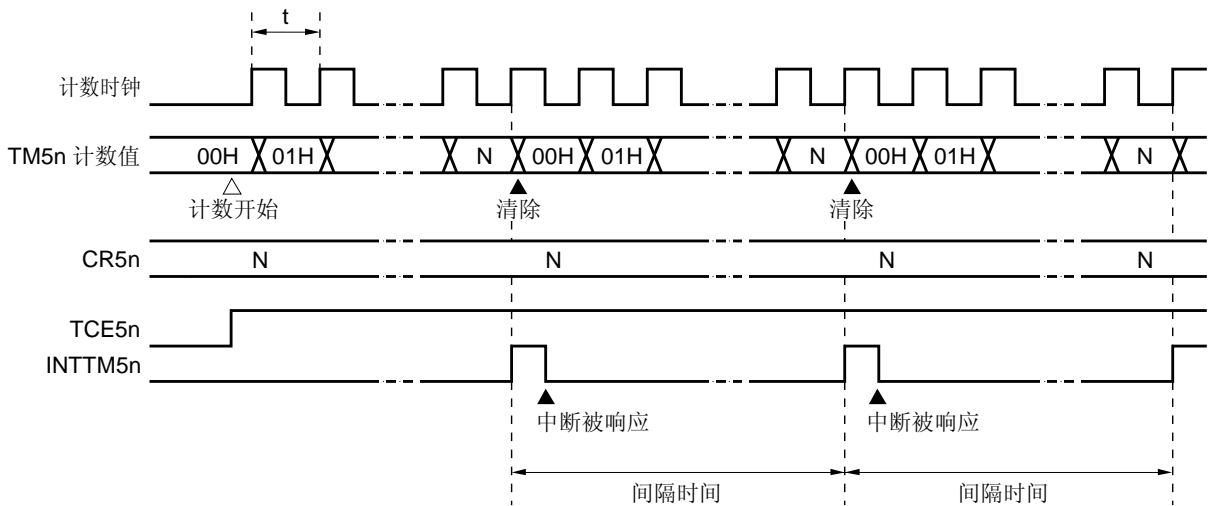
设置 TCE5n 为 0 以停止计数操作。

注意事项 不要在操作期间向 CR5n 写入其它值。

备注 关于如何允许 INTTM5n 信号中断，参见第十六章 中断功能。

图 8-11. 间隔定时器操作时序 (1 / 2)

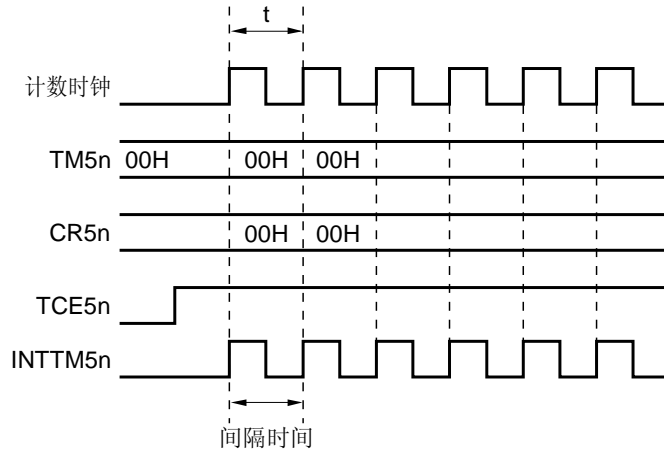
(a) 基本操作



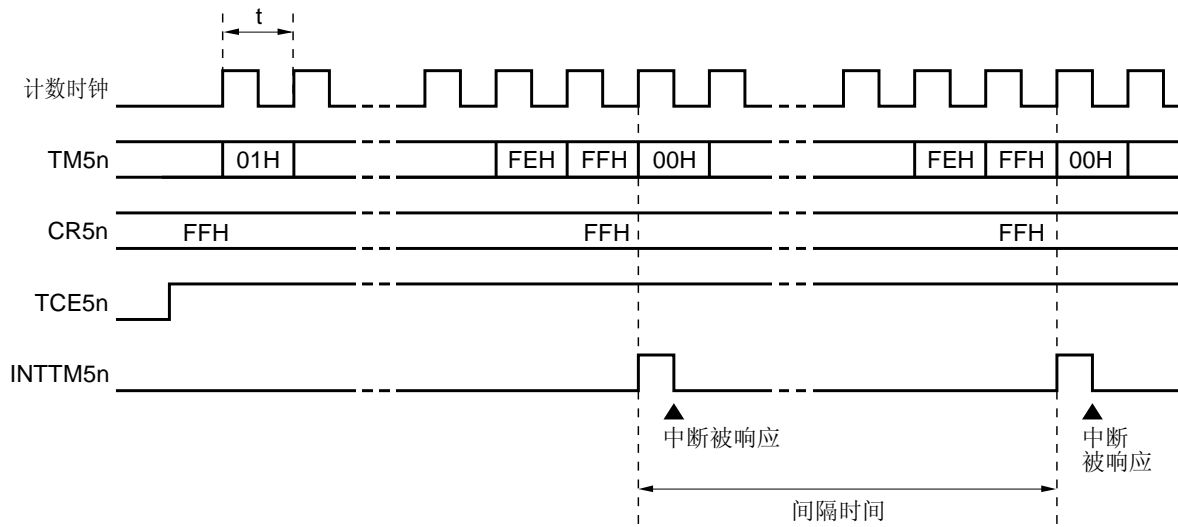
备注 间隔时间 = $(N + 1) \times t$
 $N = 01H$ 到 FFH
 $n = 0, 1$

图 8-11. 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

8.4.2 作为外部事件计数器工作

外部事件计数器通过 8 位定时器计数器 5n (TM5n) 来计数由 TI5n 引脚输入的外部时钟脉冲的个数。

TM5n 每次在定时器时钟选择寄存器 5n (TCL5n) 指定的有效沿输入时递增。可以选择上升沿或下降沿。

当 TM5n 计数值与 8 位定时器比较寄存器 5n (CR5n) 的值匹配时, TM5n 被清零并且产生中断请求信号 (INTTM5n)。

只要 TM5n 值与 CR5n 的值匹配, 就产生 INTTM5n。

设置

<1> 设置每个寄存器。

- 设置端口模式寄存器 (PM17 或 PM33) ^注 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 和 CR5n 匹配时清零&启动的模式,
禁止定时器 F/F 反转操作, 禁止定时器输出。
(TMC5n = 00000000B)

<2> 当设置 TCE5n = 1 时, 对 TI5n 引脚输入的脉冲个数计数。

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

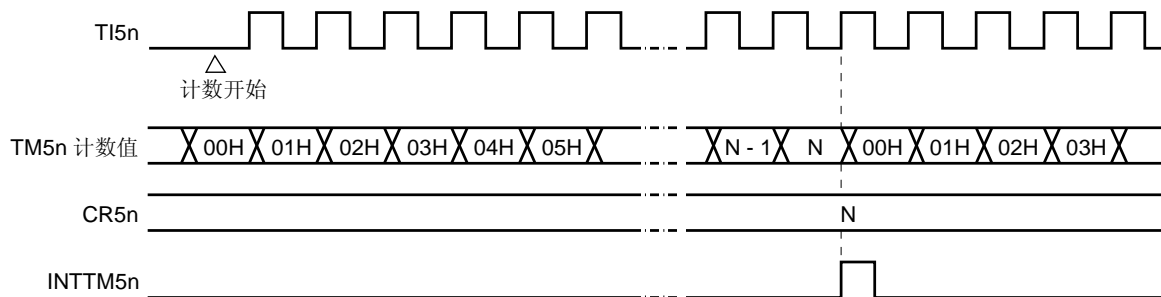
<4> 这些设置完成后, 在每次 TM5n 和 CR5n 的值匹配时产生 INTTM5n。

注 8 位定时器 / 事件计数器 50: PM17

8 位定时器 / 事件计数器 51: PM33

备注 关于如何允许 INTTM5n 信号中断, 参见第十六章 中断功能。

图 8-12. 外部事件计数器操作时序 (指定为上升沿)



备注 N = 00H 到 FFH

n = 0, 1

8.4.3 方波输出操作

按照预先设置到 8 位定时器比较寄存器 5n (CR5n) 中的值所确定的间隔输出任选频率的方波。

通过设置 8 位定时器模式控制寄存器 5n (TMC5n) 的第 0 位 (TOE5n) 为 1, 按照预先设置到 CR5n 中的计数值所确定的间隔反转 TO5n 的输出状态。因此可以产生任意选择频率的方波输出 (占空比= 50%)。

设置

<1> 设置每个寄存器。

- 清除端口输出锁存器 (P17 或 P33) ^注 和端口模式寄存器 (PM17 或 PM33) ^注 为 0。
- TCL5n: 选择计数时钟。
- CR5n: 比较值
- TMC5n: 停止计数操作, 选择 TM5n 和 CR5n 匹配时产生清零&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 清零 (0) (TO5n 输出的默认值: 低电平)
1	0	定时器输出 F/F 置位 (1) (TO5n 输出的默认值: 高电平)

定时器输出允许

(TMC5n = 00001011B 或 00000111B)

<2> 设置 TCE5n = 1 后, 开始计数操作。

<3> TM5n 和 CR5n 匹配时反转定时器输出 F/F。在产生 INTTM5n 后, TM5n 被清除为 00H。

<4> 这些设置完成后, 以相同的间隔反转定时器输出 F/F 并且从 TO5n 输出方波。

频率如下所示。

- 频率 = $1/2t(N + 1)$
(N: 00H 到 FFH)

注 8 位定时器 / 事件计数器 50: P17, PM17

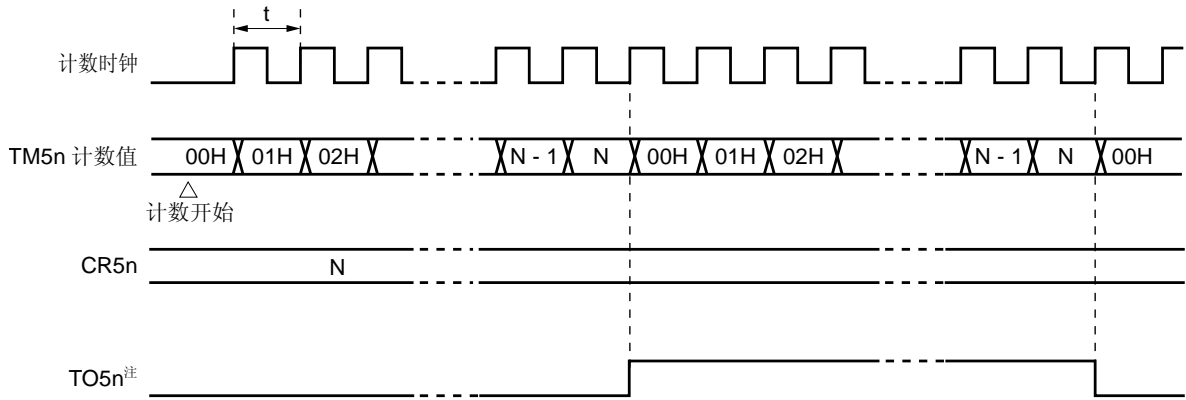
8 位定时器 / 事件计数器 51: P33, PM33

注意事项 不要在操作期间向 CR5n 写入其它值。

备注 1. 关于如何允许 INTTM5n 信号中断, 参见第十六章 中断功能。

2. n = 0, 1

图 8-13. 方波输出操作时序



注 TO5n 输出的初始值可以由 8 位定时器模式控制寄存器 5n (TMC5n) 的第 2 和 3 位 (LVR5n, LVS5n) 来设置。

8.4.4 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) 设置为 1 时, 8 位定时器 / 事件计数器 5n 作为 PWM 输出工作。

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

设置 PWM 脉冲的有效电平宽度为 CR5n; 有效电平可以由 TMC5n 的第 1 位 (TMC5n1) 选择。

计数时钟可以由定时器时钟选择寄存器 5n (TCL5n) 的第 0 到 2 位 (TCL5n0 到 TCL5n2) 来选择。

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

- 注意事项**
1. 在 PWM 模式下, 使 CR5n 的重新写入周期为 3 个计数时钟 (由 TCL5n 选择的时钟) 或更多。
 2. 8 位定时器 / 事件计数器 50 和 51 和 8 位定时器 H0 和 H1 的输出 (TO50、TO51、TOH0 和 TOH1) 在内部分别被连接到恒定电流驱动的 PWM3、PWM0、PWM1 和 PWM2 引脚。要输入与恒定电流驱动的 PWM0 到 PWM3 引脚相同频率的 PWM 信号, 按照下面设置每个计数器的计数时钟。

定时器	计数时钟选择	PWM 频率	
		f _{PRS} = 4 MHz	f _{PRS} = 8 MHz
TM50	f _{PRS} /2 ⁶	244 Hz	488 Hz
TM51			
TMH0			
TMH1			

备注 n = 0, 1

(1) PWM 输出基本操作

设置

<1> 设置每个寄存器。

- 清除端口输出锁存器 (P17 或 P33)[※] 和端口模式寄存器 (PM17 或 PM33)[※] 为 0。
- TCL5n: 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 不变。

TMC5n1	有效电平选择
0	高电平有效
1	低电平有效

定时器输出允许

(TMC5n = 01000001B 或 01000011B)

<2> 当 TCE5n = 1 时, 计数操作开始。
清除 TCE5n 为 0 以停止计数操作。

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

PWM 输出操作

- <1> 在溢出发生前, PWM 输出 (TO5n 输出) 一直输出无效电平。
<2> 当溢出发生时, 输出有效电平。在 CR5n 与 8 位定时器计数器 5n (TM5n) 的计数值匹配前, 输出有效电平。
<3> 在 CR5n 与计数值匹配后, 在溢出再次发生前, 输出无效电平。
<4> 重复执行操作 <2>和<3>, 直到计数操作停止。
<5> 当计数操作被 TCE5n = 0 停止时, PWM 输出变为无效。

关于时序的详细信息, 参见图 8-14 和 8-15。

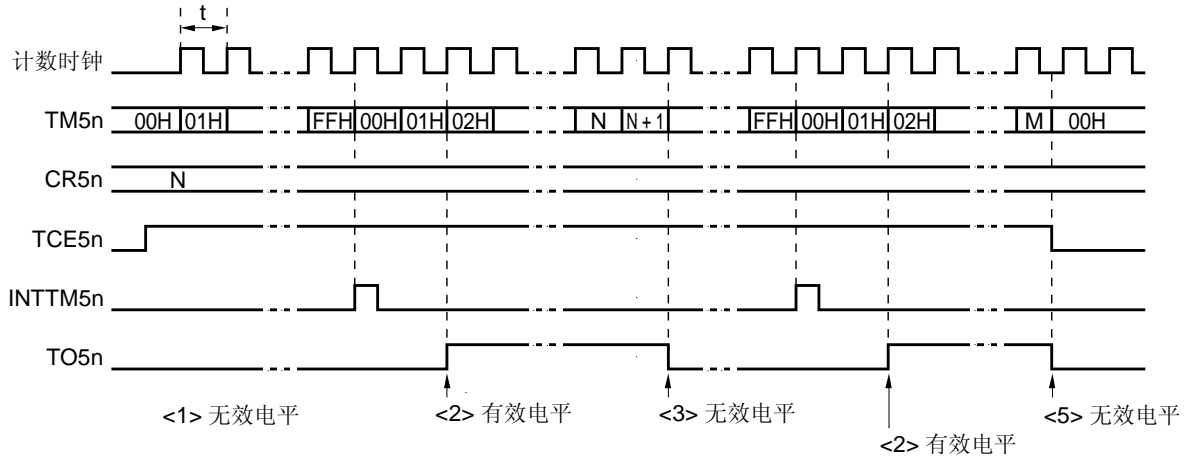
周期、有效电平宽度和占空比如下所示。

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

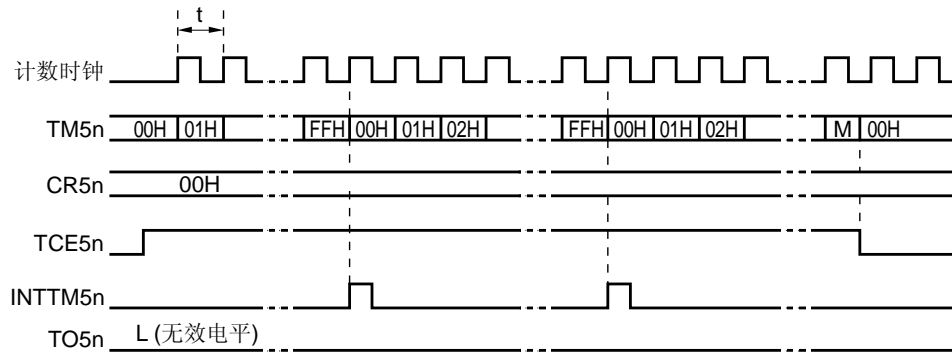
备注 n = 0, 1

图 8-14. PWM 输出操作时序

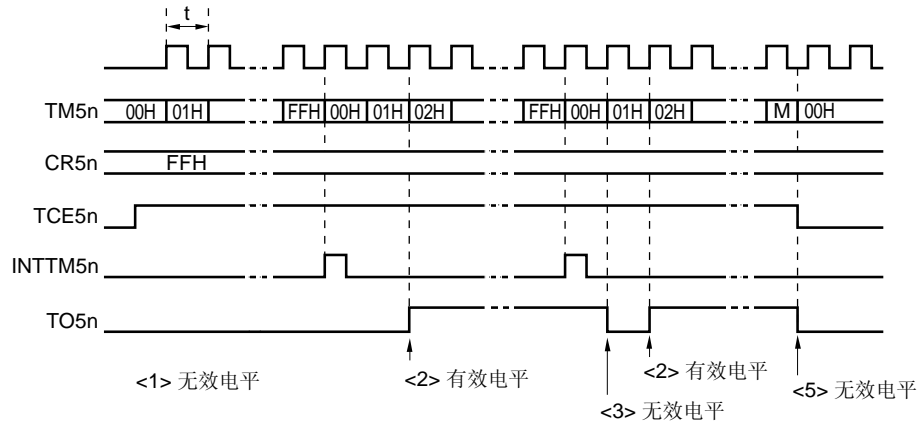
(a) 基本操作 (有效电平 = H)



(b) CR5n = 00H



(c) CR5n = FFH

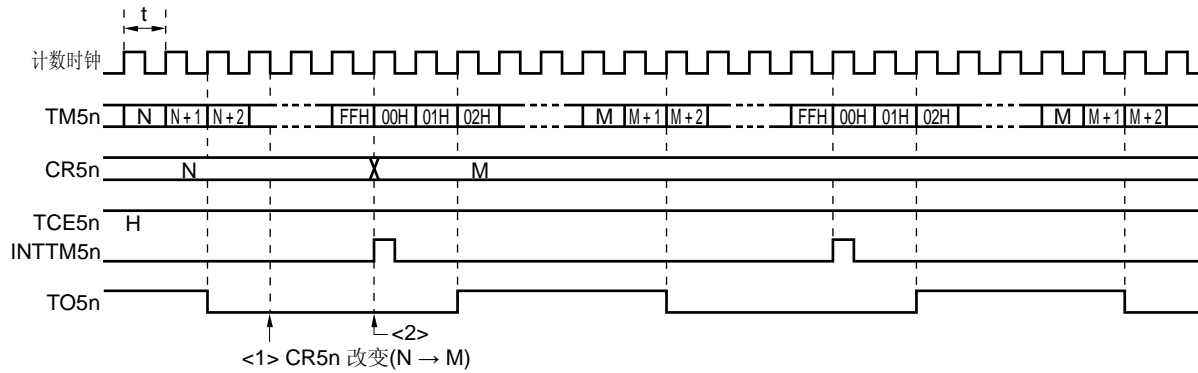


- 备注
- 图 8-14 (a) 中的<1> 到 <3> 和 <5> 对应于 8.4.4 (1) PWM 输出基本操作中 PWM 输出操作 中的 <1>到<3>和<5>。
 - $n = 0, 1$

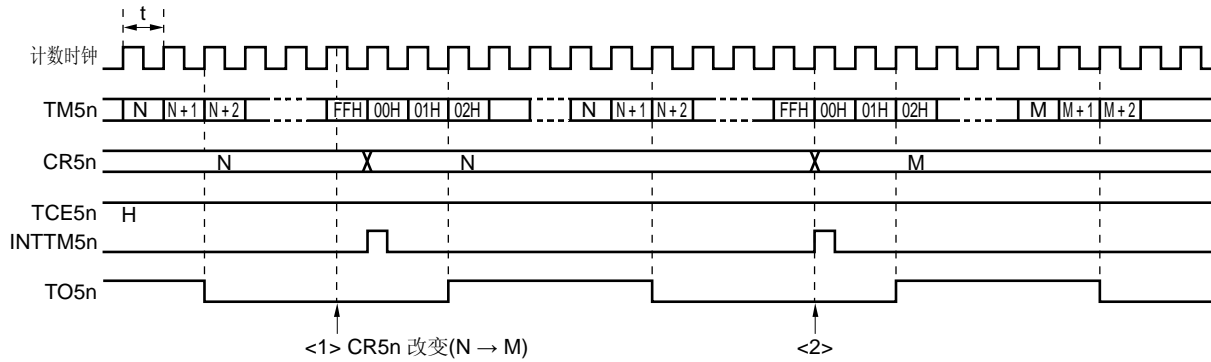
(2) CR5n 改变的操作

图 8-15. CR5n 改变的操作时序

(a) CR5n 值在 FFH 的时钟上升沿前从 N 变为 M → 在改变后产生溢出时该值立即被传送到 CR5n。



(b) CR5n 值在 FFH 的时钟上升沿后从 N 变为 M → 在第二个溢出时该值被传送到 CR5n。



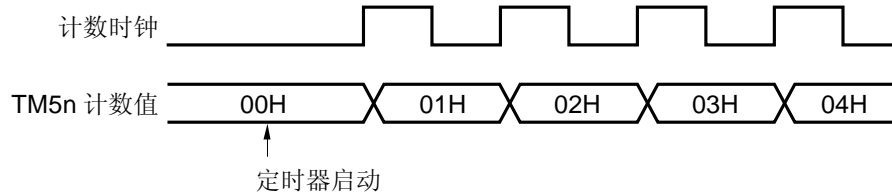
注意事项 在图 8-15 中的<1> 和 <2>之间读取 CR5n，读取的值与实际值不同（读取值：M，CR5n 的实际值：N）。

8.5 8 位定时器 / 事件计数器 50 和 51 的注意事项

(1) 定时器启动错误

在定时器启动后到产生匹配信号需要的时间，最多有一个时钟的误差。这是由于 8 位定时器计数器 50 和 51 (TM50, TM51) 与计数时钟之间异步启动引起。

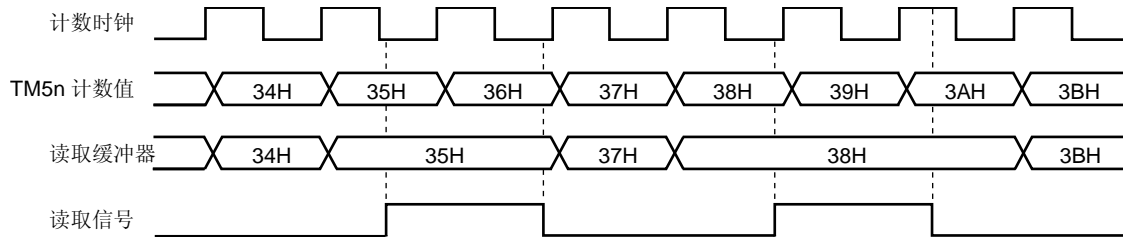
图 8-16. 8 位定时器计数器 5n (TM5n) 的启动时序



(2) 8 位定时器计数器 5n (TM5n) 的读取

可以在没有停止实际计数器时读取 TM5n，因为读取时捕捉到的缓冲器的计数值是固定的。然而，在计数器刚刚向上计数前读取时，缓冲器可能没有被更新，因为在计数器向上计数时更新缓冲器。

图 8-17. 8 位定时器计数器 5n (TM5n) 的读取时序



备注 n = 0, 1

第九章 8 位定时器H0 和H1

9.1 8 位定时器H0 和H1 的功能

8 位定时器 H0 和 H1 有以下功能。

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

注意事项 8 位定时器 / 事件计数器 50 和 51 和 8 位定时器 H0 和 H1 的输出（TO50、TO51、TOH0 和 TOH1）在内部分别被连接到恒定电流驱动的 PWM3、PWM0、PWM1 和 PWM2 引脚。要输入与恒定电流驱动的 PWM0 到 PWM3 引脚相同频率的 PWM 信号，按照下面设置每个计数器的计数时钟。

定时器	计数时钟选择	PWM 频率	
		$f_{PRS} = 4 \text{ MHz}$	$f_{PRS} = 8 \text{ MHz}$
TM50	$f_{PRS}/2^6$	244 Hz	488 Hz
TM51			
TMH0			
TMH1			

9.2 8 位定时器H0 和H1 的配置

8 位定时器 H0 和 H1 包含以下硬件。

表 9-1. 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	TOHn, 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 1 (PM1) 端口寄存器 1 (P1)

注 只有 8 位定时器 H1

备注 n = 0, 1

图 9-1 和 9-2 表示框图。

图 9-1. 8 位定时器 H0 的框图

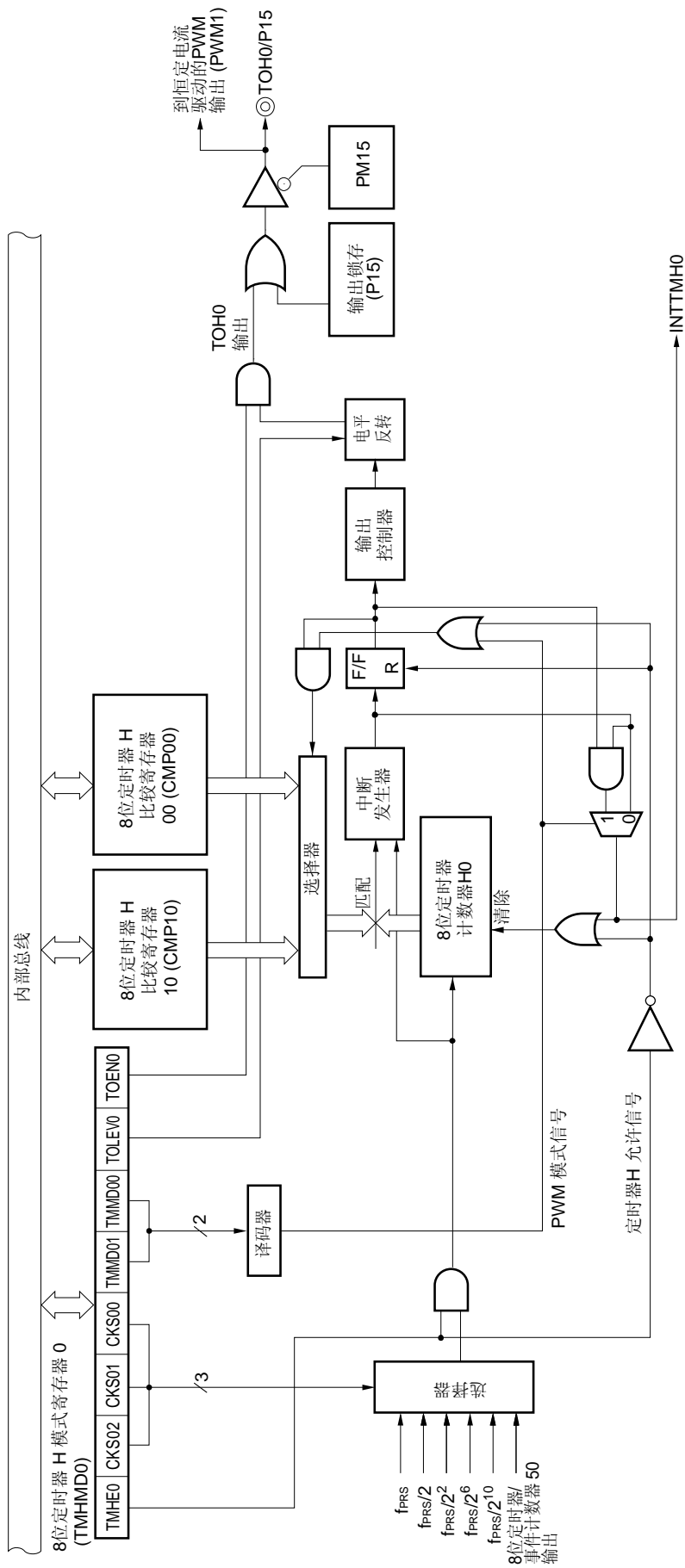
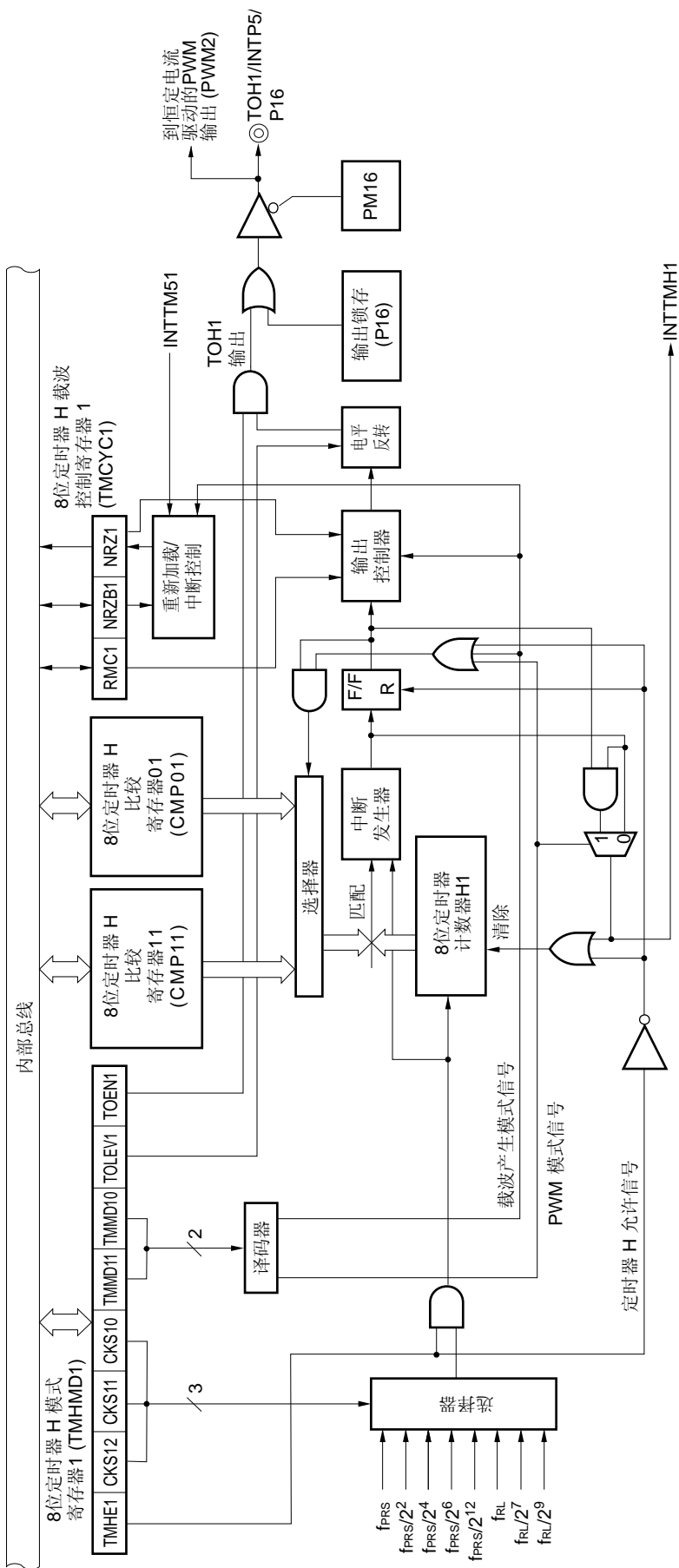


图 9-2. 8 位定时器 H1 的框图



(1) 8 位定时器 H 比较寄存器 0n (CMP0n)

该寄存器可以由 8 位存储器操作指令读取或写入。该寄存器用于所有定时器操作模式。

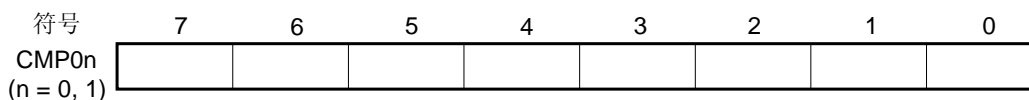
该寄存器总是比较设置到 CMP0n 的值和 8 位定时器计数器 Hn 的值，并且当两个值匹配时，产生中断请求信号 (INTTMHn) 并反转 TOHn 的输出电平。

当定时器停止 (TMHEn = 0) 时重新写入 CMP0n 的值。

复位信号的产生清除该寄存器为 00H。

图 9-3. 8 位定时器 H 比较寄存器 0n (CMP0n) 的格式

地址: FF18H (CMP00), FF1AH (CMP01) 复位后: 00H R/W



注意事项 在定时器计数操作期间不能重新写入 CMP0n。可以在定时器计数操作期间刷新 CMP0n（写入相同的值）。

(2) 8 位定时器 H 比较寄存器 1n (CMP1n)

该寄存器可以由 8 位存储器操作指令读取或写入。该寄存器用于 PWM 输出模式和载波发生器模式。

在 PWM 模式下，该寄存器比较设置到 CMP1n 的值和 8 位定时器计数器 Hn 的值，并且当两个值匹配时，反转 TOHn 的输出电平。不产生中断请求信号。

在载波发生器模式下，CMP1n 寄存器比较设置到 CMP1n 的值和 8 位定时器计数器 Hn 的值，并且当两个值匹配时，产生中断请求信号 (INTTMHn)。这时，计数值被清除。

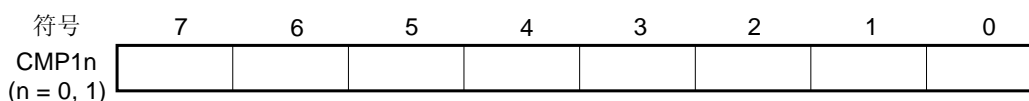
可以在定时器计数操作期间刷新（写入相同的值）和重新写入 CMP1n。

如果在定时器工作期间重新写入 CMP1n，新的值被锁存，并且当定时器的计数值与 CMP1n 的旧值匹配时传送到 CMP1n，然后 CMP1n 的值改变为新值。如果计数值和 CMP1n 值的匹配与写入 CMP1n 的操作冲突，CMP1n 的值不变。

复位信号的产生清除该寄存器为 00H。

图 9-4. 8 位定时器 H 比较寄存器 1n (CMP1n) 的格式

地址: FF19H (CMP10), FF1BH (CMP11) 复位后: 00H R/W



注意事项 在 PWM 输出模式和载波发生器模式下，在定时器计数操作停止（TMHE_n = 0）后当定时器计数操作开始（TMHE_n = 1）时，确保设置 CMP1_n（即使设置相同的值到 CMP1_n，也要确保再次设置）。

备注 n = 0, 1

9.3 控制 8 位定时器 H0 和 H1 的寄存器

以下四个寄存器用来控制 8 位定时器 H0 和 H1。

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

注 只有 8 位定时器 H1

(1) 8 位定时器 H 模式寄存器 n（TMHMD_n）

该寄存器控制定时器 H 的模式。

该寄存器可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生清除该寄存器为 00H。

备注 n = 0, 1

图 9-5. 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	定时器操作允许							
0	停止定时器计数操作(计数器被清除为 0)							
1	允许定时器计数操作(计数操作由输入时钟启动)							
CKS02	CKS01	CKS00	计数时钟选择 ^{*1}					
				$f_{PRS} =$ 4 MHz	$f_{PRS} =$ 8 MHz	$f_{PRS} =$ 10 MHz	$f_{PRS} =$ 20 MHz	
0	0	0	f_{PRS}^{*2}	4 MHz	8 MHz	10 MHz	20 MHz ^{*3}	
0	0	1	$f_{PRS}/2$	2 MHz	4 MHz	5 MHz	10 MHz	
0	1	0	$f_{PRS}/2^2$	1 MHz	2 MHz	2.5 MHz	5 MHz	
0	1	1	$f_{PRS}/2^6$	62.5 kHz	125 kHz	156.25 kHz	312.5 kHz	
1	0	0	$f_{PRS}/2^{10}$	3.91 kHz	7.81 kHz	9.77 kHz	19.54 kHz	
1	0	1	TM50 输出 ^{*4}					
除上面以外			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						
除上面以外		禁止设置						
TOLEV0	定时器输出电平控制(在默认模式下)							
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
- 如果外围硬件时钟 (f_{PRS}) 工作于高速系统时钟 (f_{XH}) ($XSEL = 1$)，根据供电电压改变 f_{PRS} 工作频率。
 - $V_{DD} = 2.7$ 到 5.5 V: $f_{PRS} \leq 20$ MHz
 - $V_{DD} = 1.8$ 到 2.7 V: $f_{PRS} \leq 5$ MHz
 - 如果外围硬件时钟 (f_{PRS}) 工作于内部高速振荡时钟 (f_{RH}) ($XSEL = 0$)，当 1.8 V \leq V_{DD} $<$ 2.7 V 时，禁止设置 $CKS02 = CKS01 = CKS00 = 0$ (计数时钟: f_{PRS})。
 - 只有在 4.0 V \leq V_{DD} \leq 5.5 V 的情况下可以设置。

- 注 4. 当选择 TM50 输出作为计数时钟时，注意以下要点。
- TM50 和 CR50 匹配 (TMC506 = 0) 后计数时钟被清除和启动的模式
首先启动 8 位定时器 / 事件计数器 50 的操作，然后允许定时器 F/F 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，使占空比 = 50%。
在任何模式下都不必允许 (TOE50 = 1) TO50 输出。

- 注意事项 1. 当 TMHE0 = 1 时，禁止设置 TMHMD0 的其它位。然而，能够刷新 TMHMD0 (写入相同的值)。
2. 在 PWM 输出模式下，在定时器计数操作停止 (TMHE0 = 0) 后当定时器计数操作开始 (TMHE0 = 1) 时，确保设置 8 位定时器 H 比较寄存器 10 (CMP10) (即使为 CMP10 设置相同的值，也要确保再次设置)。
3. 根据 PM15 和 P15，以及 TOH0 输出决定实际 TOH0 / P15 引脚的输出。
4. 8 位定时器 / 事件计数器 50 和 51 和 8 位定时器 H0 和 H1 的输出 (TO50、TO51、TOH0 和 TOH1) 在内部分别被连接到恒定电流驱动的 PWM3、PWM0、PWM1 和 PWM2 引脚。要输入与恒定电流驱动的 PWM0 到 PWM3 引脚相同频率的 PWM 信号，按照下面设置每个计数器的计数时钟。

定时器	计数时钟选择	PWM 频率	
		f _{PRS} = 4 MHz	f _{PRS} = 8 MHz
TM50	f _{PRS} /2 ⁶	244 Hz	488 Hz
TM51			
TMH0			
TMH1			

- 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
3. TMC501: TMC50 的第 1 位

图 9-6. 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	定时器操作允许
0	停止定时器计数操作(计数器被清除为 0)
1	允许定时器计数操作(计数操作由输入时钟启动)

CKS12	CKS11	CKS10	计数时钟选择 ^{注 1}				
			$f_{PRS} = 4 \text{ MHz}$	$f_{PRS} = 8 \text{ MHz}$	$f_{PRS} = 10 \text{ MHz}$	$f_{PRS} = 20 \text{ MHz}$	
0	0	0	$f_{PRS}^{注 2}$	4 MHz	8 MHz	10 MHz	20 MHz ^{注 3}
0	0	1	$f_{PRS}/2^2$	1 MHz	2 MHz	2.5 MHz	5 MHz
0	1	0	$f_{PRS}/2^4$	250 kHz	500 kHz	625 kHz	1.25 MHz
0	1	1	$f_{PRS}/2^6$	62.5 kHz	125 kHz	156.25 kHz	312.5 kHz
1	0	0	$f_{PRS}/2^{12}$	0.98 kHz	1.95 kHz	2.44 kHz	4.88 kHz
1	0	1	$f_{RL}/2^7$	1.88 kHz (典型)			
1	1	0	$f_{RL}/2^9$	0.47 kHz (典型)			
1	1	1	f_{RL}	240 kHz (典型)			

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(在默认模式下)
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

- 注
- 如果外围硬件时钟 (f_{PRS}) 工作于高速系统时钟 (f_{XH}) ($XSEL = 1$)，根据供电电压改变 f_{PRS} 工作频率。
 - $V_{DD} = 2.7$ 到 5.5 V : $f_{PRS} \leq 20 \text{ MHz}$
 - $V_{DD} = 1.8$ 到 2.7 V : $f_{PRS} \leq 5 \text{ MHz}$
 - 如果外围硬件时钟 (f_{PRS}) 工作于内部高速振荡时钟 (f_{RH}) ($XSEL = 0$)，当 $1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$ 时，禁止设置 $CKS12 = CKS11 = CKS10 = 0$ (计数时钟: f_{PRS})。
 - 只有在 $4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$ 的情况下可以设置。

- 注意事项**
1. 当 $TMHE1 = 1$ 时，禁止设置 $TMHMD1$ 的其它位。然而，能够刷新 $TMHMD1$ （写入相同的值）。
 2. 在 PWM 输出模式和载波发生器模式下，在定时器计数操作停止（ $TMHE1 = 0$ ）后当定时器计数操作开始（ $TMHE1 = 1$ ）时，确保设置 8 位定时器 H 比较寄存器（ $CMP11$ ）（即使为 $CMP11$ 设置相同的值，也要确保再次设置）。
 3. 当使用载波发生器模式时，设置使 $TMH1$ 的计数时钟频率变为 多于 $TM51$ 的计数时钟频率的 6 倍。
 4. 实际 $TOH1$ / $INTP5$ / $P16$ 引脚输出由 $PM16$ 和 $P16$ ，以及 $TOH1$ 输出决定。
 5. 8 位定时器 / 事件计数器 50 和 51 和 8 位定时器 H0 和 H1 的输出（ $TO50$ 、 $TO51$ 、 $TOH0$ 和 $TOH1$ ）在内部分别被连接到恒定电流驱动的 $PWM3$ 、 $PWM0$ 、 $PWM1$ 和 $PWM2$ 引脚。要输入与恒定电流驱动的 $PWM0$ 到 $PWM3$ 引脚相同频率的 PWM 信号，按照下面设置每个计数器的计数时钟。

定时器	计数时钟选择	PWM 频率	
		$f_{PRS} = 4 \text{ MHz}$	$f_{PRS} = 8 \text{ MHz}$
TM50	$f_{PRS}/2^6$	244 Hz	488 Hz
TM51			
TMH0			
TMH1			

- 备注**
1. f_{PRS} : 外围硬件时钟频率
 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)

该寄存器控制 8 位定时器 H1 的远程控制输出和载波脉冲输出状态。

该寄存器可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生清除该寄存器为 00H。

图 9-7. 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	远程控制输出
0	0	低电平输出
0	1	在INTTM51 信号输入的上升沿, 高电平输出
1	0	低电平输出
1	1	在INTTM51 信号的上升沿, 载波脉冲输出

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 位只读。

注意事项 当 TMHE1 = 1 时, 不要重新写入 RMC1。然而, 能够刷新 TMCYC1 (写入相同的值)。

(3) 端口模式寄存器 1 (PM1)

该寄存器以 1 位为单位设置端口 1 输入 / 输出。

当使用 P15 / TOH0 和 P16 / TOH1 / INTP5 引脚作为定时器输出时, 清除 PM15 和 PM16 以及 P15 和 P16 的输出锁存器为 0。

PM1 可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生设置该寄存器为 FFH。

图 9-8. 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 复位后: FFH R / W

符号	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n 引脚 I / O 模式选择 (n = 0 到 7)
0	输出模式 (输出缓冲器开)
1	输入模式 (输出缓冲器关)

9.4 8 位定时器 H0 和 H1 的操作

9.4.1 作为间隔定时器 / 方波输出工作

当 8 位定时器计数器 H1 和比较寄存器 0n (CMP0n) 的值匹配时, 产生中断请求信号 (INTTMHn) 并且 8 位定时器计数器 Hn 被清除为 00H。

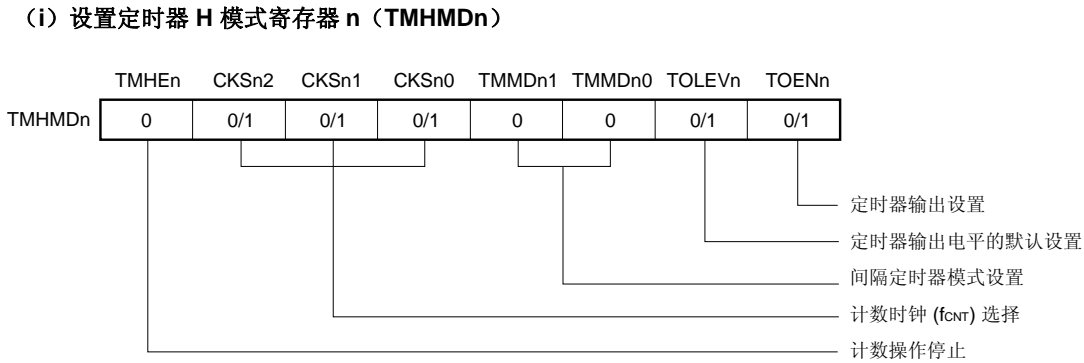
比较寄存器 1n (CMP1n) 不用于间隔定时器模式。因为即使设置 CMP1n 寄存器, 也不能检测 8 位定时器计数器 Hn 和 CMP1n 寄存器的匹配, 定时器输出不受影响。

通过设置定时器 H 模式寄存器 n (TMHMDn) 的第 0 位 (TOENn) 为 1, 从 TOHn 输出任意频率 (占空比 = 50%) 的方波。

设置

<1> 设置每个寄存器。

图 9-9. 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 被设置为比较值, 间隔时间如下所示。

- 间隔时间 = (N + 1)/fcNT

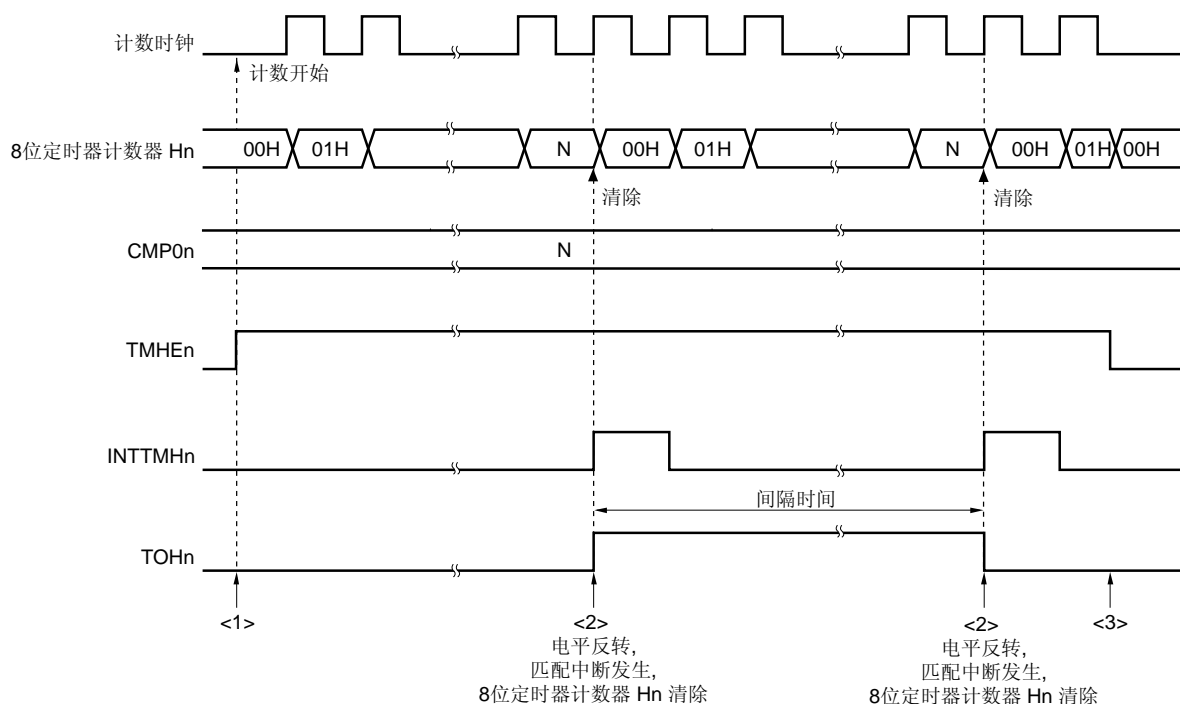
<2> 当 TMHEn = 1 时, 计数操作开始。

<3> 当 8 位定时器计数器 Hn 和 CMP0n 寄存器的值匹配时, 产生 INTTMHn 并且 8 位定时器计数器 Hn 被清除为 00H。

<4> 然后, 以同样间隔产生 INTTMHn 信号。要停止计数操作, 清除 TMHEn 为 0。

- 备注
1. 关于输出引脚的设置, 参见 9.3 (3) 端口模式寄存器 1 (PM1)。
 2. 关于如何允许 INTTMHn 信号中断, 参见 第十六章 中断功能。
 3. n = 0, 1

图 9-10. 间隔定时器 / 方波输出操作时序 (1 / 2)

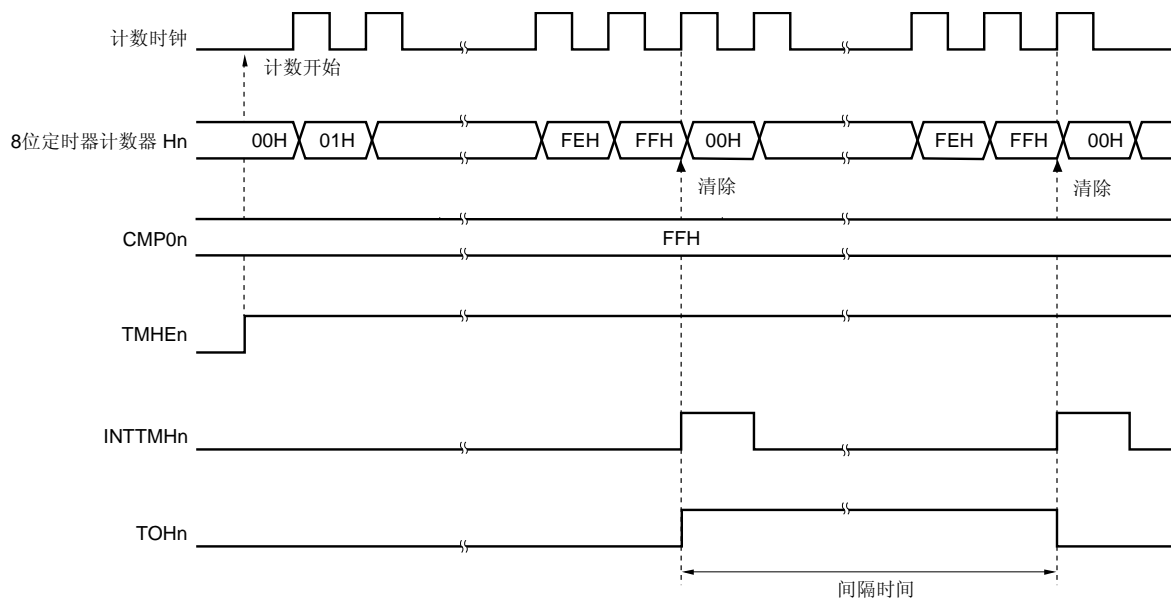
(a) 基本操作 (当 $01H \leq CMP0n \leq FEH$ 时的操作)

- <1> 通过设置 $TMHEn$ 位为 1, 允许计数操作。在允许操作后不超过一个时钟内, 计数时钟开始计数。
- <2> 当 8 位定时器计数器 Hn 的值与 $CMP0n$ 寄存器的值匹配时, 清除定时器计数器的值, 并且反转 $TOHn$ 输出电平。此外, 在计数时钟的上升沿输出 $INTTMHn$ 信号。
- <3> 如果在定时器 H 操作期间清除 $TMHEn$ 位, $INTTMHn$ 信号和 $TOHn$ 输出被设置为默认电平。如果在 $TMHEn$ 位被清除为 0 前它们已经处于默认电平, 则该电平被保持。

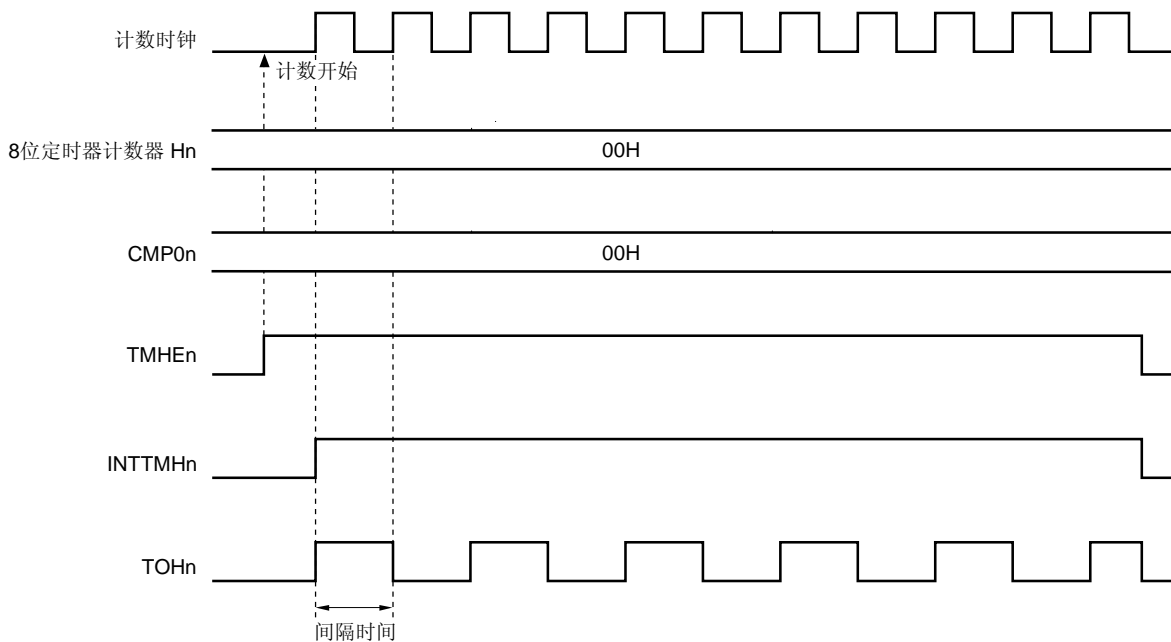
备注 $n = 0, 1$
 $01H \leq N \leq FEH$

图 9-10. 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

9.4.2 作为PWM输出工作

PWM 输出模式下，能够输出任意占空比和任意周期的脉冲。

8 位定时器比较寄存器 0n (CMP0n) 控制定时器输出 (TOHn) 的周期。在定时器操作期间，禁止重新写入 CMP0n 寄存器。

8 位定时器比较寄存器 1n (CMP1n) 控制定时器输出 (TOHn) 的占空比。在定时器操作期间，可以重新写入 CMP1n 寄存器。

PWM 模式下的操作如下所示。

在定时器计数开始后，当 8 位定时器计数器 Hn 和 CMP0n 寄存器匹配时，PWM 输出 (TOHn 输出) 输出一个有效电平并且 8 位定时器计数器 Hn 被清除为 0。当 8 位定时器计数器 Hn 和 CMP1n 寄存器匹配时，PWM 输出 (TOHn 输出) 输出一个无效电平。

注意事项 8 位定时器 / 事件计数器 50 和 51 和 8 位定时器 H0 和 H1 的输出 (TO50、TO51、TOH0 和 TOH1) 在内部分别被连接到恒定电流驱动的 PWM3、PWM0、PWM1 和 PWM2 引脚。要输入与恒定电流驱动的 PWM0 到 PWM3 引脚相同频率的 PWM 信号，按照下面设置每个计数器的计数时钟。

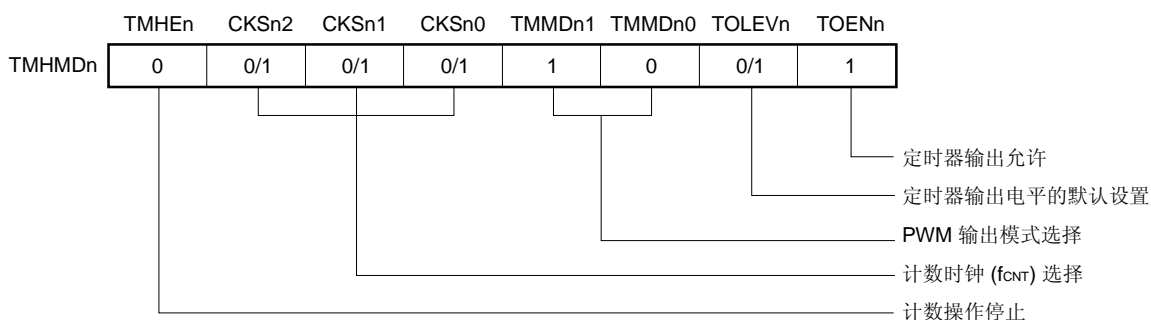
定时器	计数时钟选择	PWM 频率	
		$f_{PRS} = 4 \text{ MHz}$	$f_{PRS} = 8 \text{ MHz}$
TM50	$f_{PRS}/2^6$	244 Hz	488 Hz
TM51			
TMH0			
TMH1			

设置

<1> 设置每个寄存器。

图 9-11. PWM 输出模式下的寄存器设置

(i) 设置定时器 H 模式寄存器 n (TMHMDn)



(ii) 设置 CMP0n 寄存器

- 比较值 (N)：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. $00H \leq \text{CMP1n} (M) < \text{CMP0n} (N) \leq FFH$

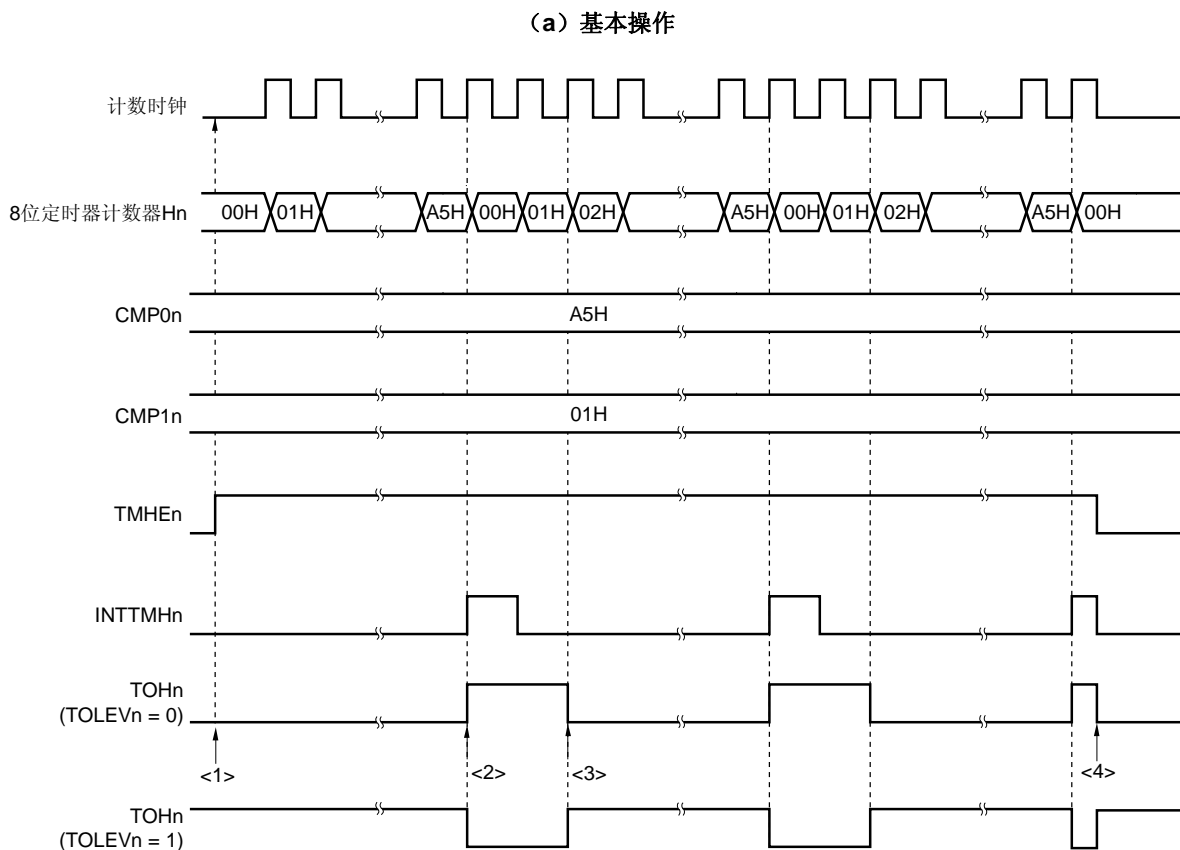
- <2> 当 $TMHE_n = 1$ 时，计数操作开始。
- <3> $CMP0_n$ 寄存器是计数器操作允许后首先被比较的比较寄存器。当 8 位定时器计数器 H_n 的值和 $CMP0_n$ 寄存器匹配时，8 位定时器计数器 H_n 被清除，产生中断请求信号 ($INTTMH_n$)，并且输出一个有效电平。同时，与 8 位定时器计数器 H_n 比较的比较寄存器从 $CMP0_n$ 寄存器改变为 $CMP1_n$ 寄存器。
- <4> 当 8 位定时器计数器 H_n 和 $CMP1_n$ 寄存器匹配时，输出一个无效电平并且将与 8 位定时器计数器 H_n 比较的比较寄存器从 $CMP1_n$ 寄存器改变为 $CMP0_n$ 寄存器。这时，不清除 8 位定时器计数器 H_n 并且不产生 $INTTMH_n$ 信号。
- <5> 通过重复执行过程<3>和<4>，得到一个具有任意占空比的脉冲。
- <6> 要停止计数操作，设置 $TMHE_n = 0$ 。
如果 $CMP0_n$ 寄存器的设置值为 N ， $CMP1_n$ 寄存器的设置值为 M ，并且计数时钟频率为 f_{CNT} ，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. 在定时器计数器工作期间，可以更改 $CMP1_n$ 寄存器的设置值。然而，从 $CMP1_n$ 寄存器的值被更改到这个值被传送到寄存器需要三个工作时钟（通过 $TMHMD_n$ 寄存器的 $CKSn_2$ 到 $CKSn_0$ 位选择的信号）。
 2. 在定时器计数操作停止 ($TMHE_n = 0$) 后，当定时器计数操作开始 ($TMHE_n = 1$) 时，确保设置 $CMP1_n$ 寄存器（即使为 $CMP1_n$ 寄存器设置相同的值，也要确保再次设置）。
 3. 确保 $CMP1_n$ 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. 关于输出引脚的设置，参见 9.3 (3) 端口模式寄存器 1 (PM1)。
 2. 关于如何允许 $INTTMH_n$ 信号中断，参见 第十六章 中断功能。
 3. $n = 0, 1$

图 9-12. PWM 输出模式下的操作时序 (1 / 4)

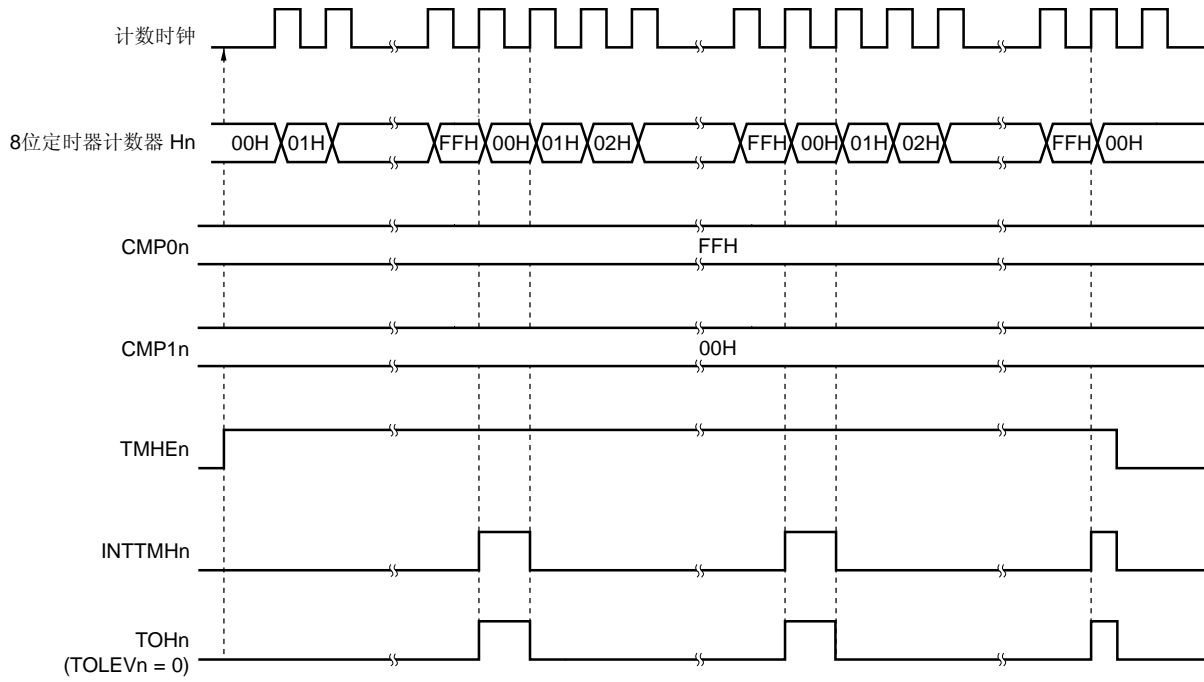


- <1> 通过设置 **TMHEn** 位为 1 来允许计数操作。通过屏蔽一个计数时钟来启动 8 位定时器计数器 **Hn** 向上计数。这时，PWM 输出一个无效电平。
- <2> 当 8 位定时器计数器 **Hn** 的值和 **CMP1n** 寄存器匹配时，输出一个有效电平。这时，8 位定时器计数器 **Hn** 被清除并且输出 **INTTMHn** 信号。
- <3> 当 8 位定时器计数器 **Hn** 的值和 **CMP0n** 寄存器匹配时，输出一个无效电平。这时，8 位计数器值不清除并且不输出 **INTTMHn** 信号。
- <4> 定时器 **Hn** 操作期间清除 **TMHEn** 位为 0 可设置 **INTTMHn** 信号为默认值，并且 PWM 输出无效电平。

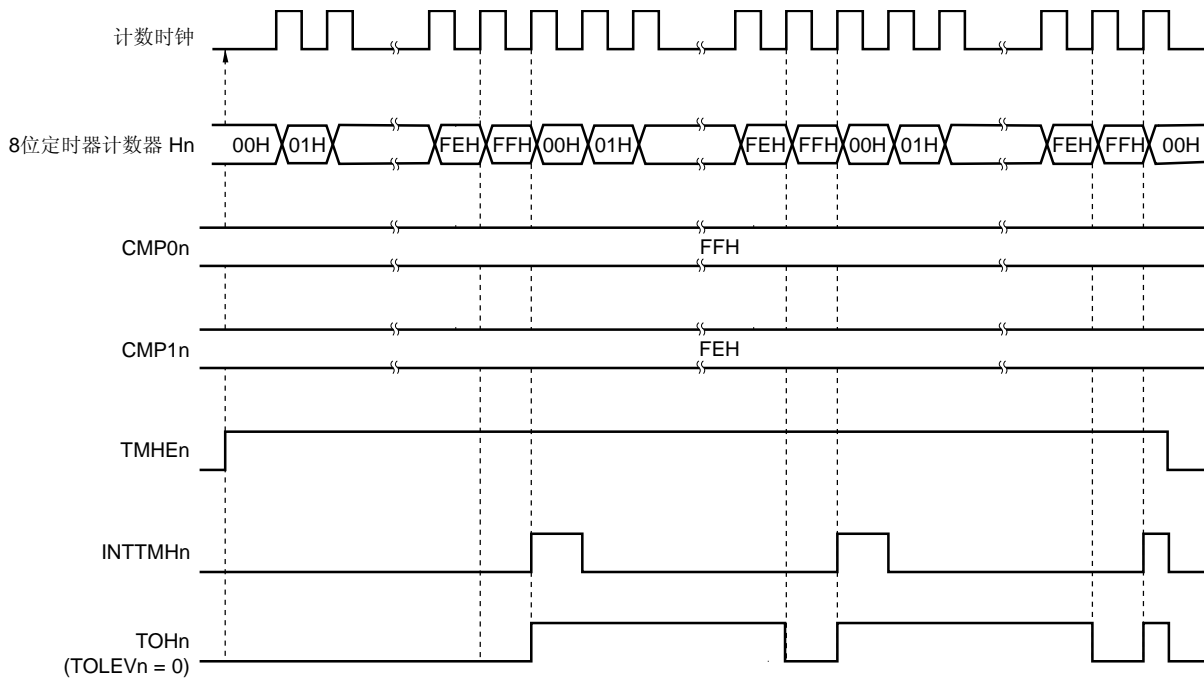
备注 $n = 0, 1$

图 9-12. 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 9-12. PWM 输出模式下的操作时序 (3 / 4)

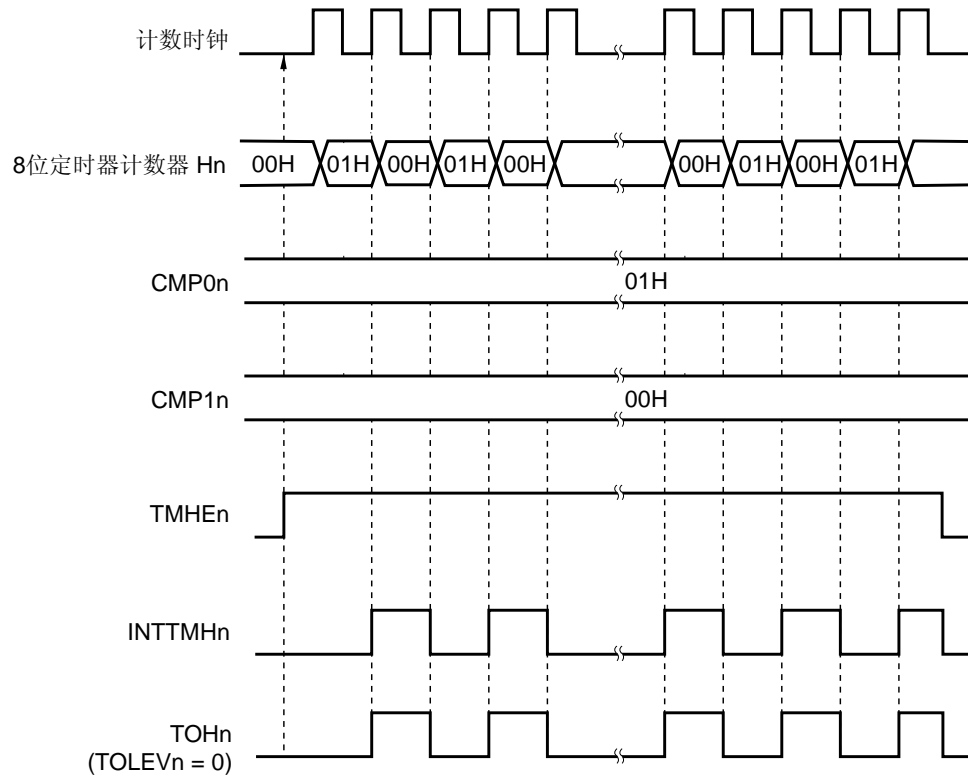
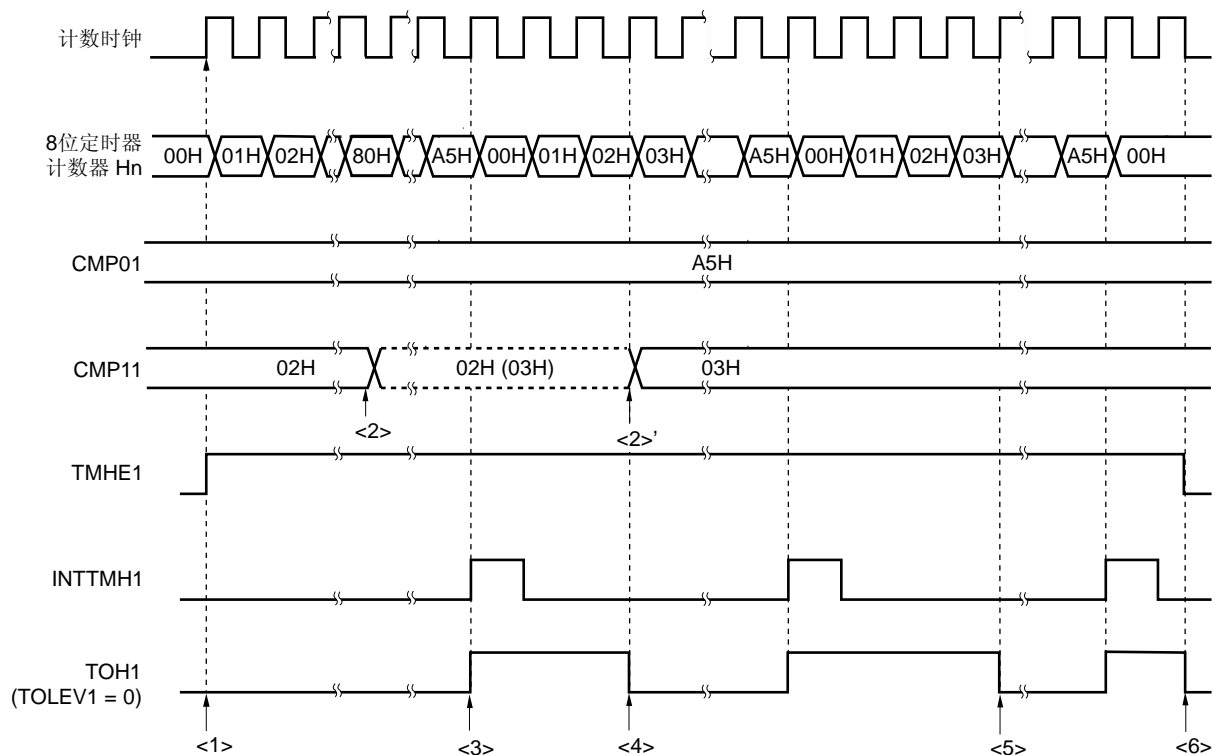
(d) 当 $CMP0n = 01H$, $CMP1n = 00H$ 时的操作备注 $n = 0, 1$

图 9-12. PWM 输出模式下的操作时序 (4 / 4)

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



- <1> 通过设置 $TMHE_n = 1$ 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 H_n 向上计数。这时, PWM 输出无效电平。
- <2> $CMP1_n$ 寄存器的值可以在定时器计数器操作期间改变。这个操作与计数时钟异步。
- <3> 当 8 位定时器计数器 H_n 的值和 $CMP0_n$ 寄存器匹配时, 清除 8 位定时器计数器 H_n 的值, 输出有效电平, 并且输出 $INTTMH_n$ 信号。
- <4> 如果 $CMP1_n$ 寄存器的值被改变, 这个值被锁存并且不会传送到寄存器。当 8 位定时器计数器 H_n 的值与 $CMP1_n$ 寄存器改变前的值匹配时, 这个值被传送到 $CMP1_n$ 寄存器, 并且更改 $CMP1_n$ 寄存器的值 (<2>')。
- 然而, 从改变 $CMP1_n$ 寄存器的值到这个值被传送到寄存器需要三个计数时钟或更多。如果在三个计数时钟内产生匹配信号, 改变的值不能被传送到寄存器。
- <5> 当 8 位定时器计数器 H_n 的值和改变后的 $CMP1_n$ 寄存器匹配时, 输出无效电平。不清除 8 位定时器计数器 H_n , 并且不产生 $INTTMH_n$ 信号。
- <6> 定时器 H_n 操作期间清除 $TMHE_n$ 位为 0 将设置 $INTTMH_n$ 信号为默认值, 并且 PWM 输出无效电平。

备注 $n = 0, 1$

9.4.3 载波发生器操作（只有 8 位定时器 H1）

在载波发生器模式下，8 位定时器 H1 被用来产生一个红外远程控制器的载波信号，并且 8 位定时器 / 事件计数器 51 被用来产生一个红外远程控制信号（时间计数）。

由 8 位定时器 H1 产生的载波时钟按照由 8 位定时器 / 事件计数器 51 设置的周期输出。

在载波发生器模式下，8 位定时器 H1 载波脉冲的输出由 8 位定时器 / 事件计数器 51 控制，并且从 TOH1 输出。

(1) 载波产生

在载波发生器模式下，8 位定时器 H 比较寄存器 01（CMP01）产生低电平宽度载波脉冲波形，8 位定时器 H 比较寄存器 11（CMP11）产生高电平宽度载波脉冲波形。

在 8 位定时器 H1 操作期间，可以重新写入 CMP11 寄存器，而禁止修改 CMP01 寄存器。

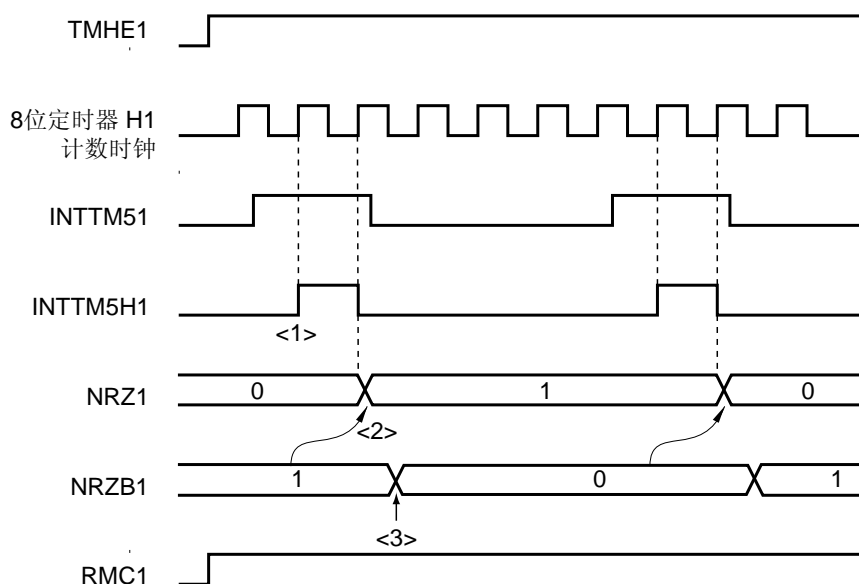
(2) 载波输出控制

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 位控制。与输出的关系如下所示。

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

要在计数操作期间控制载波脉冲输出，TMCYC1 寄存器的 NRZ1 和 NRZB1 位有主和从位配置。NRZ1 位只读，但是 NRZB1 可以被读写。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，NRZB1 位的值被传送到 NRZ1 位。从 NRZB1 位到 NRZ1 位的传送时序如下所示。

图 9-13. 传送时序



- <1> INTTM51 信号与 8 位定时器 H1 的计数时钟同步，并且作为 INTTM5H1 信号输出。
- <2> NRZB1 位的值在 INTTM5H1 信号的上升沿后的第二个时钟时被传送到 NRZ1 位。
- <3> 在已经由 INTTM5H1 中断启动的中断服务程序中，或者通过轮询中断请求标志已经执行检查时序后，写入下一个值到 NRZB1 位。写入下次计数的数据到 CR51 寄存器。

- 注意事项**
1. 在 NRZB1 位被重新写入后第二个时钟之前不要再次重新写入 NRZB1 位，否则不能保证从 NRZB1 位到 NRZ1 位的传送。
 2. 当 8 位定时器 / 事件计数器 51 用于载波发生器模式下时，在时序<1>产生一个中断。当 8 位定时器 / 事件计数器 51 被用于载波发生器模式之外时，中断产生时序不同。

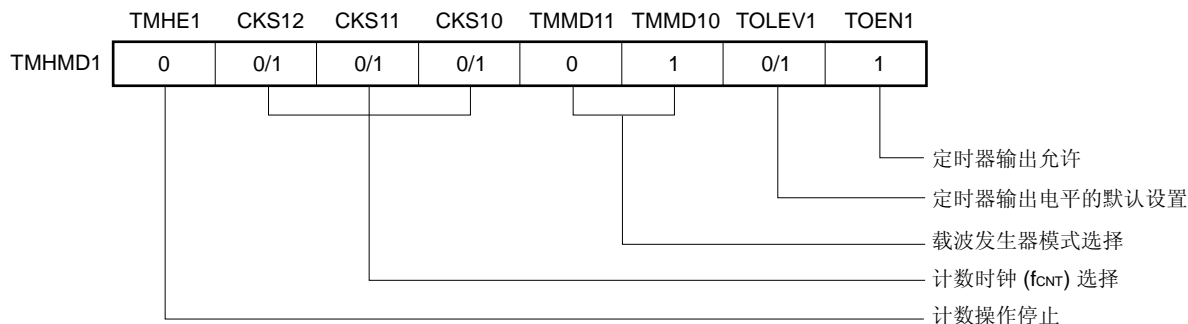
备注 INTTM5H1 是一个内部信号，不是一个中断源。

设置

<1> 设置每个寄存器。

图 9-14. 载波发生器模式下的寄存器设置

(i) 设置 8 位定时器 H 模式寄存器 1 (TMHMD1)



(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 ... 远程控制输出允许位
- NRZB1 = 0 / 1 ... 载波输出允许位

(v) TCL51 和 TMC51 寄存器设置

- 参见 8.3 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 被设置为 1 时, 8 位定时器 / 事件计数器 51 开始计数。

<4> 在允许计数操作后, 参与比较的第一个比较寄存器是 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, 产生 INTTMH1 信号, 清除 8 位定时器计数器 H1。同时, 要与 8 位定时器计数器 H1 比较的比较寄存器从 CMP01 寄存器切换为 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, 产生 INTTMH1 信号, 清除 8 位定时器计数器 H1。同时, 要与 8 位定时器计数器 H1 比较的比较寄存器从 CMP11 寄存器切换为 CMP01 寄存器。

<6> 通过重复执行过程<4>和<5>, 产生一个载波时钟。

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

<8> 在已经由 INTTM5H1 中断启动的中断服务程序中, 或者在通过轮询中断请求标志已经执行检查时序后, 写入下一个值到 NRZB1 位。写入下次计数的数据到 CR51 寄存器。

<9> 当 NRZ1 位为高电平时, 一个载波时钟通过 TOH1 输出。

<10> 通过执行上面的过程，得到一个任意的载波时钟。要停止计数操作，清除 TMHE1 为 0。

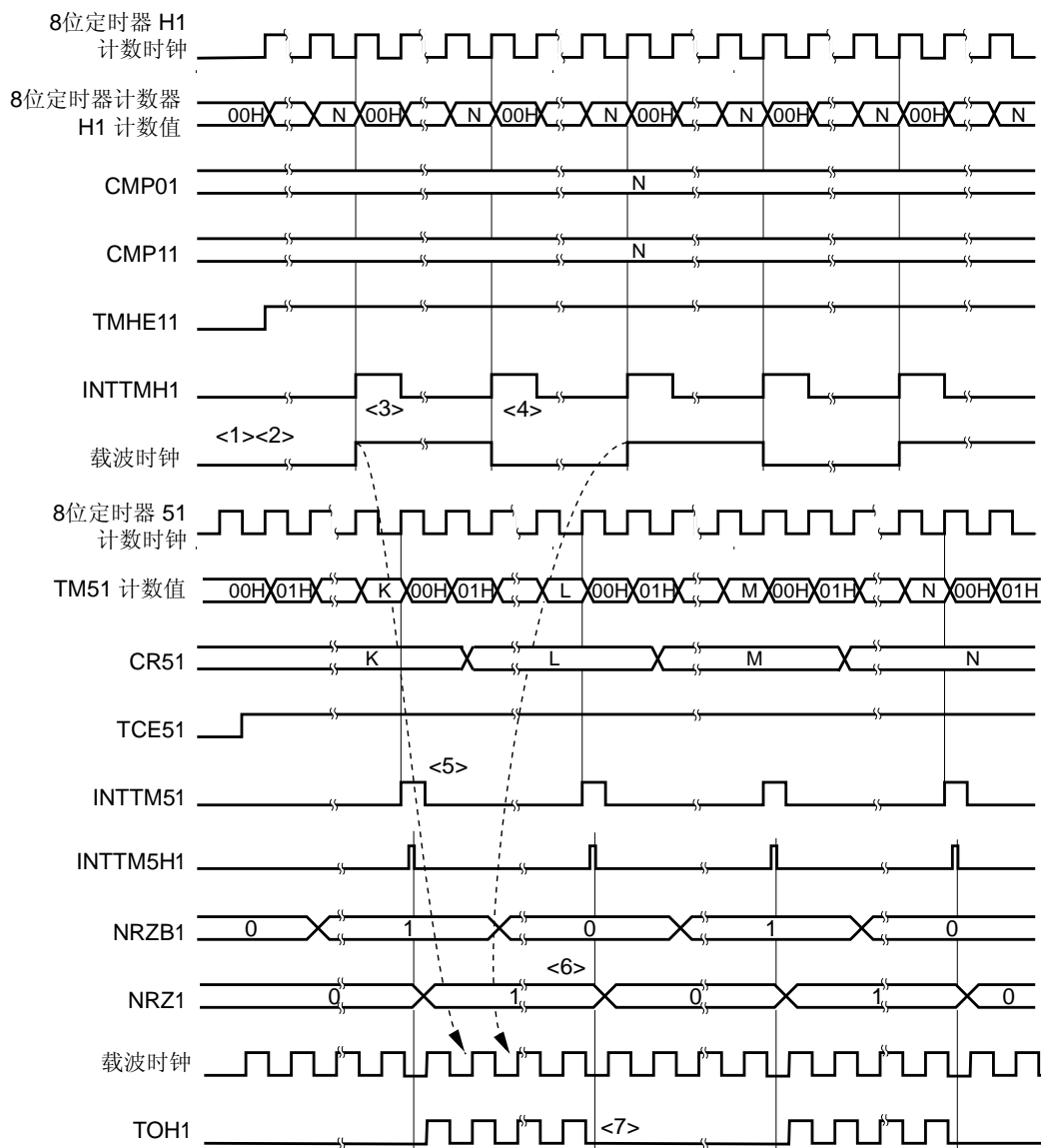
如果 CMP01 寄存器的设置值为 N，CMP11 寄存器的设置值为 M，并且计数时钟频率为 f_{CNT} ，载波时钟输出周期和占空比如下所示。

- 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. 在定时器计数操作停止 (TMHE1 = 0) 后，当定时器计数操作开始 (TMHE1 = 1) 时，确保设置 CMP11 寄存器（即使为 CMP11 寄存器设置相同的值，也要确保再次设置）。
 2. 设置使 TMH1 的计数时钟频率为大于 TM51 的计数时钟频率的 6 倍。
 3. 设置 CMP01 和 CMP11 寄存器的值在 01H 到 FFH 范围内。
 4. 在定时器计数器工作期间，可以更改 CMP11 寄存器的设置值。然而，从改变 CMP11 寄存器的值到这个值被传送到寄存器需要三个工作时钟（通过 TMHMD1 寄存器的 CKS12 到 CKS10 位选择的信号）。
 5. 在计数操作开始前，确保设置 RMC1 位。

- 备注**
1. 关于输出引脚的设置，参见 9.3 (3) 端口模式寄存器 1 (PM1)。
 2. 关于如何允许 INTTMH1 信号中断，参见 第十六章 中断功能。

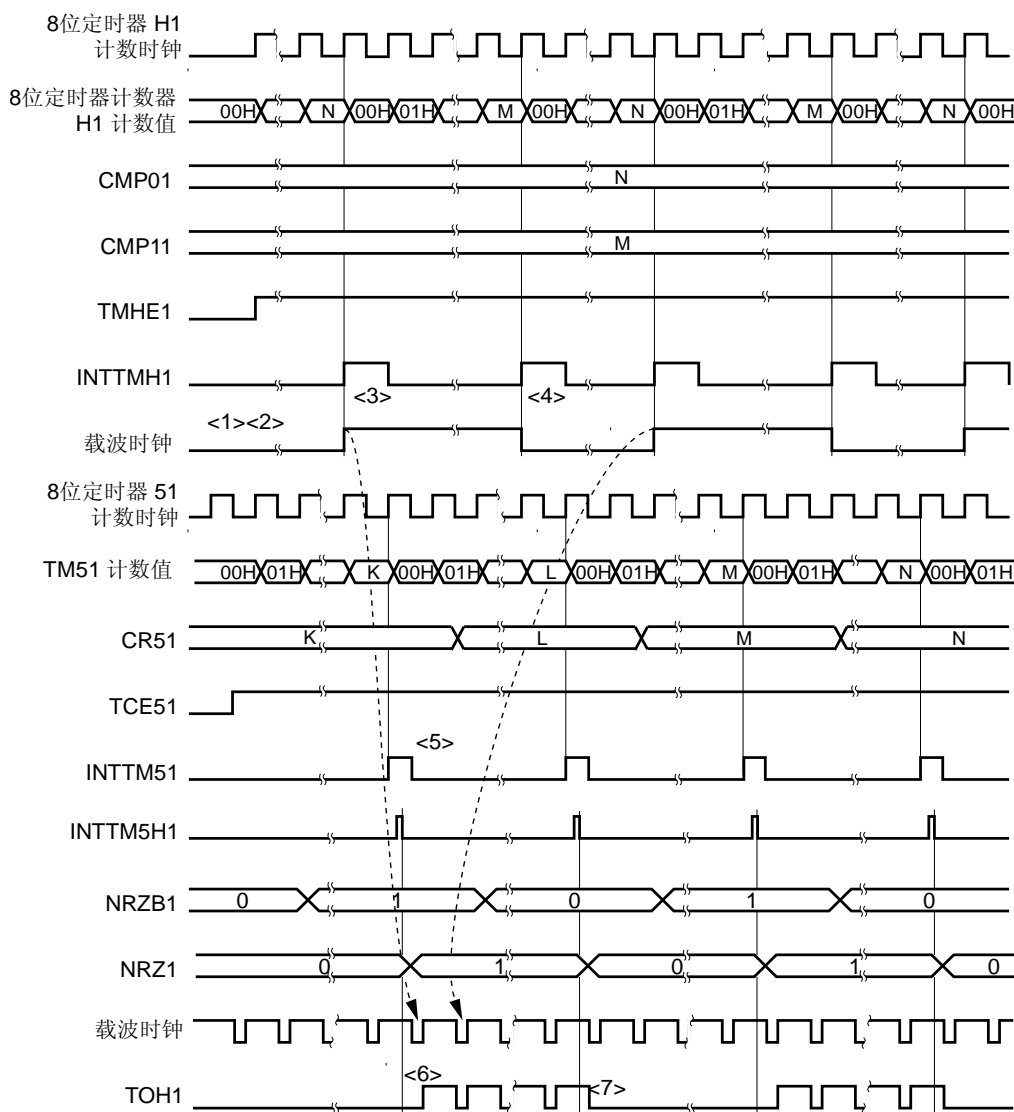
图 9-15. 载波发生器模式下的操作时序 (1 / 3)

(a) 当 $CMP01 = N$, $CMP11 = N$ 时的操作

- <1> 当 $TMHE1 = 0$ 和 $TCE51 = 0$ 时, 停止 8 位定时器计数器 H1 操作。
- <2> 当设置 $TMHE1 = 1$ 时, 8 位定时器计数器 H1 开始计数操作。这时, 载波时钟仍然为默认值。
- <3> 当 8 位定时器计数器 H1 的值与 $CMP01$ 寄存器的值匹配时, 产生第一个 $INTTMH1$ 信号, 反转载波时钟信号, 并且要与 8 位定时器计数器 H1 比较的比较寄存器从 $CMP01$ 寄存器切换为 $CMP11$ 寄存器。8 位定时器计数器 H1 被清除为 00H。
- <4> 当 8 位定时器计数器 H1 的值与 $CMP11$ 寄存器的值匹配时, 产生 $INTTMH1$ 信号, 反转载波时钟信号, 并且要与 8 位定时器计数器 H1 比较的比较寄存器从 $CMP11$ 寄存器切换为 $CMP01$ 寄存器。8 位定时器计数器 H1 清除为 00H。通过重复执行过程<3> 和 <4>, 产生一个具有固定占空比 50%的载波时钟。
- <5> 当产生 $INTTM51$ 信号时, 它与 8 位定时器 H1 的计数时钟同步, 并且作为 $INTTM5H1$ 信号输出。
- <6> $INTTM5H1$ 信号变为 $NRZB1$ 位的数据传送信号, 并且 $NRZB1$ 位的值被传送到 $NRZ1$ 位。
- <7> 当 $NRZ1 = 0$ 被设置时, $TOH1$ 输出变为低电平。

备注 $INTTM5H1$ 是一个内部信号, 不是一个中断源。

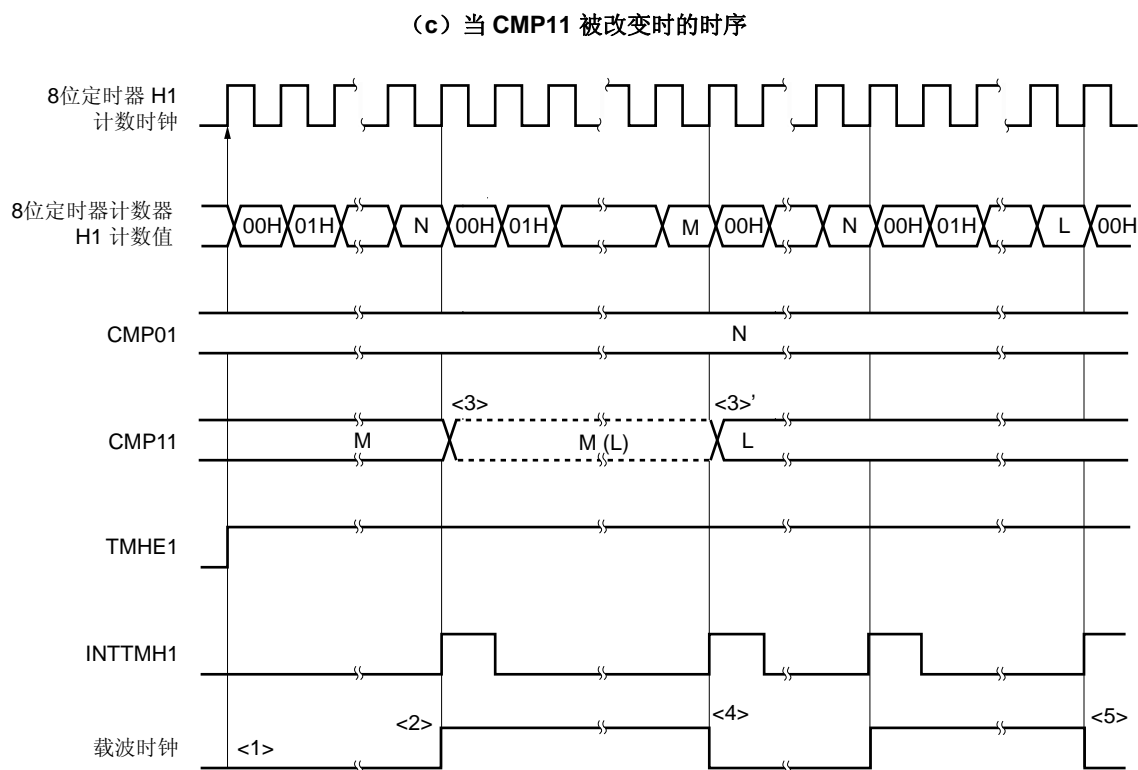
图 9-15. 载波发生器模式下的操作时序 (2 / 3)

(b) 当 $CMP01 = N$, $CMP11 = M$ 时的时序

- <1> 当 $TMHE1 = 0$ 和 $TCE51 = 0$ 时, 停止 8 位定时器计数器 H1 操作。
- <2> 当设置 $TMHE1 = 1$ 时, 8 位定时器计数器 H1 开始计数操作。这时, 载波时钟仍然为默认值。
- <3> 当 8 位定时器计数器 H1 的值与 $CMP01$ 寄存器的值匹配时, 产生第一个 $INTTMH1$ 信号, 载波时钟信号被反转, 并且要与 8 位定时器计数器 H1 比较的比较寄存器从 $CMP01$ 寄存器切换为 $CMP11$ 寄存器。8 位定时器计数器 H1 被清除为 00H。
- <4> 当 8 位定时器计数器 H1 的值与 $CMP11$ 寄存器的值匹配时, 产生 $INTTMH1$ 信号, 反转载波时钟信号, 并且要与 8 位定时器计数器 H1 比较的比较寄存器从 $CMP11$ 寄存器切换为 $CMP01$ 寄存器。8 位定时器计数器 H1 被清除为 00H。通过重复执行过程<3> 和 <4>, 产生一个具有固定占空比 50% 的载波时钟。
- <5> 当产生 $INTTM51$ 信号时, 它与 8 位定时器 H1 的计数时钟同步, 并且作为 $INTTM5H1$ 信号输出。
- <6> 如果 $NRZ1$ 被设置为 1, 在载波时钟的第一个上升沿输出一个载波信号。
- <7> 当 $NRZ1 = 0$ 时, $TOH1$ 输出保持在高电平, 并且在载波时钟为高电平时不会改变为低电平 (从 <6> 和 <7>, 不能保证载波时钟波形的高电平宽度)。

备注 $INTTM5H1$ 是一个内部信号, 不是一个中断源。

图 9-15. 载波发生器模式下的操作时序 (3 / 3)



- <1> 当设置 $TMHE1 = 0$ 时, 停止 8 位定时器计数器 H1 操作。这时, 载波时钟仍然为默认值。
- <2> 当 8 位定时器计数器 H1 的值与 CMP01 寄存器的值匹配时, 输出 INTTMH1 信号, 反转载波信号, 并且定时器计数器 H1 被清除为 00H。同时, 要与 8 位定时器计数器 H1 的值比较的比较寄存器从 CMP01 寄存器改变为 CMP11 寄存器。
- <3> CMP11 寄存器与计数时钟异步, 并且它的值可以在 8 位定时器 H1 工作期间改变。寄存器的值要改变的新值 (L) 被锁存。当 8 位定时器计数器 H1 的值与改变前的 CMP11 寄存器的值 (M) 匹配时, CMP11 寄存器被更改 (<3>')。
- 然而, 从 CMP11 寄存器的值被更改到这个值被传送到寄存器需要三个计数时钟或更多。如果在三个计数时钟内产生匹配信号, 改变的值不能被传送到寄存器。
- <4> 当 8 位定时器计数器 H1 的值与改变前的 CMP11 寄存器的值 (M) 匹配时, 输出 INTTMH1 信号, 反转载波信号, 并且定时器计数器被清除为 00H。同时, 要与 8 位定时器计数器 H1 的值比较的比较寄存器从 CMP11 寄存器改变为 CMP01 寄存器。
- <5> 8 位定时器计数器 H1 的计数值和 CMP11 寄存器的值再次匹配的时序由改变后的值 (L) 表示。

第十章 看门狗定时器

10.1 看门狗定时器的功能

看门狗定时器以内部低速振荡时钟工作。

看门狗定时器用于检测意外的程序循环。如果检测到这种程序循环，将产生一个内部复位信号。

在以下情况下将检测程序循环。

- 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
- 如果向 WDTE 写入 “ACH” 以外的数据
- 如果在窗口关闭周期内向 WDTE 写入
- 如果从非 IMS 寄存器设置的区域中取指 (当 CPU 挂起时的无效检查的检测)
- 如果 CPU 通过执行读 / 写指令访问非 IMS 寄存器设置的区域 (FB00H 到 FFCFH 和 FFE0H 到 FFFFH 除外) (CPU 程序循环期间异常访问的检测)

当由于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) 被设置为 1。关于 RESF 的详细信息，参见 第十八章 复位功能。

10.2 看门狗定时器的配置

看门狗定时器包含以下硬件。

表 10-1. 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)

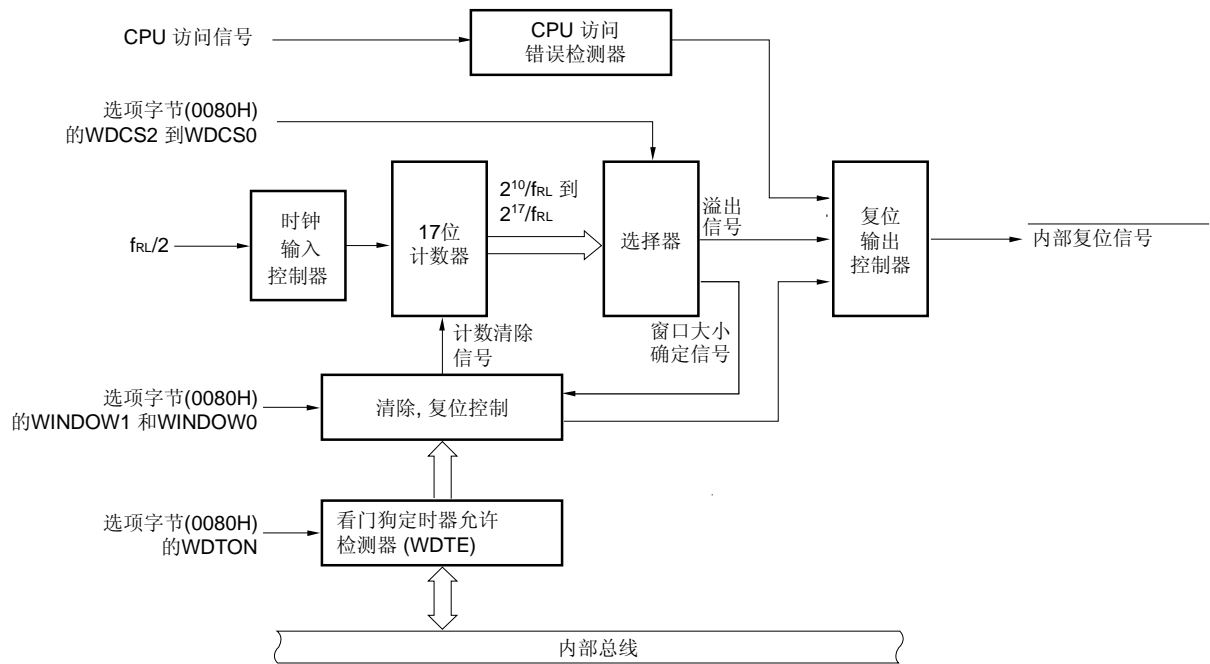
计数器如何被控制、溢出时间和窗口打开周期由选项字节设置。

表 10-2. 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和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到1位 (WDCS2 到 WDCS0)

备注 关于选项字节，参见第二十一章 选项字节。

图 10-1. 看门狗定时器的框图



10.3 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）控制。

(1) 看门狗定时器允许寄存器（WDTE）

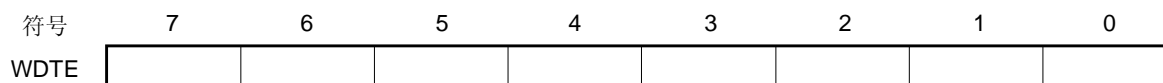
向 WDTE 写入 ACH 将清除看门狗定时器计数器，并且再次启动计数。

该寄存器可以由 8 位存储器操作指令设置。

复位信号的产生设置该寄存器为 9AH 或 1AH^注。

图 10-2. 看门狗定时器允许寄存器（WDTE）的格式

地址: FF99H 复位后: 9AH/1AH^注 R/W



注 WDTE 的复位值根据选项字节（0080H）的 WDTON 设置值而不同。要操作看门狗定时器，设置 WDTON 为 1。

WDTON 设置值	WDTE 复位值
0（看门狗定时器计数操作禁止）	1AH
1（看门狗定时器计数操作允许）	9AH

- 注意事项**
1. 如果向 WDTE 写入一个 ACH 以外的值，将产生一个内部复位信号。然而，如果看门狗定时器的源时钟被停止，当看门狗定时器的源时钟重新工作时，将产生一个内部复位信号。
 2. 如果对 WDTE 执行一个 1 位存储器操作指令，将产生一个内部复位信号。然而，如果看门狗定时器的源时钟被停止，当到看门狗定时器的源时钟重新工作时，将产生一个内部复位信号。
 3. 从 WDTE 读取的值为 9AH/1AH（不同于写入的值（ACH））。

10.4 看门狗定时器的操作

10.4.1 看门狗定时器的控制操作

1. 当使用看门狗定时器时，通过选项字节（0080H）指定它的操作。
 - 通过设置选项字节（0080H）的第 4 位（WDTON）为 1 允许看门狗定时器的计数操作（复位释放后，计数器开始工作）（关于详细信息，参见第二十一章）。

WDTON	看门狗定时器计数器 / 非法访问检测的操作控制
0	计数器操作禁止（复位后计数停止），非法访问检测操作禁止
1	计数器操作允许（复位后计数开始），非法访问检测操作允许

- 通过使用选项字节（0080H）的第 3 到 1 位（WDCS2 到 WDCS0）设置溢出时间（关于详细信息，参见 10.4.2 和第二十一章）。
 - 通过使用选项字节（0080H）的第 6 和 5 位（WINDOW1 和 WINDOW0）设置窗口打开周期（关于详细信息，参见 10.4.3 和第二十一章）。
2. 复位释放后，看门狗定时器开始计数。
 3. 通过在看门狗定时器开始计数后和选项字节设置的溢出时间前，向 WDTE 写入“ACH”，清除看门狗定时器并且再次开始计数。
 4. 然后，在窗口打开周期内，第二次或复位释放后写入 WDTE。如果在窗口关闭周期内写入 WDTE，将产生一个内部复位。
 5. 如果溢出时间期满而没有向 WDTE 写入“ACH”，将产生一个内部复位信号。
在以下情况下将产生一个内部复位信号。
 - 如果对看门狗定时器允许寄存器（WDTE）执行 1 位操作指令
 - 如果向 WDTE 写入“ACH”以外的数据
 - 如果从非 IMS 寄存器设置的区域中取指（CPU 程序循环期间的无效检查的检测）
 - 如果 CPU 通过执行读 / 写指令访问非 IMS 寄存器设置的区域（FB00H 到 FFCFH 和 FFE0 到 FFFFH 除外）（CPU 程序循环期间异常访问的检测）

- 注意事项**
1. 复位释放后，第一次写入 WDTE 将清除看门狗定时器，如果在溢出时间之前写入，看门狗定时器将再次开始计数，而与写入的时序无关。
 2. 如果通过向 WDTE 写入“ACH”清除看门狗定时器，实际的溢出时间可能与选项字节设置的溢出时间不同，最多相差 $2/f_{RL}$ 秒。
 3. 看门狗定时器在计数值溢出（FFFFH）前立即被清除。

注意事项 4. 看门狗定时器在 HALT 和 STOP 模式下的操作根据选项字节的第 0 位 (LSROSC) 的设置值而不同, 如下所示。

	LSROSC = 0 (内部低速振荡器可以由软件停止)	LSROSC = 1 (内部低速振荡器不能被停止)
在 HALT 模式下	看门狗定时器操作停止。	看门狗定时器操作继续。
在 STOP 模式下		

如果 LSROSC = 0, 看门狗定时器在 HALT 或 STOP 模式释放后继续计数。这时, 不清零计数器, 而是从停止时的计数值开始计数。

当 LSROSC = 0 时, 如果通过设置 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, 看门狗定时器停止操作。这时, 不清零计数器。

5. 在 flash 存储器自编程和 EEPROM 仿真过程中, 看门狗定时器继续操作。处理过程中, 延迟中断响应时间。设置溢出时间和窗口大小时要将这个延迟考虑在内。

10.4.2 设置看门狗定时器溢出时间

通过使用选项字节 (0080H) 的第 3 到 1 位 (WDCS2 到 WDCS0) 设置看门狗定时器的溢出时间。

如果溢出发生, 将产生一个内部复位信号。通过在溢出时间前, 在窗口打开周期内向 WDTE 写入 “ACH”, 将清除当前的计数, 并且看门狗定时器再次开始计数。

以下溢出时间被设置。

表 10-3. 看门狗定时器的溢出时间设置

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

注意事项 1. 禁止同时设置 WDCS2 = WDCS1 = WDCS0 = 0 和 WINDOW1 = WINDOW0 = 0。

2. 在 flash 存储器自编程和 EEPROM 仿真过程中, 看门狗定时器将继续操作。处理过程中, 延迟中断响应时间。设置溢出时间和窗口大小时要将这个延迟考虑在内。

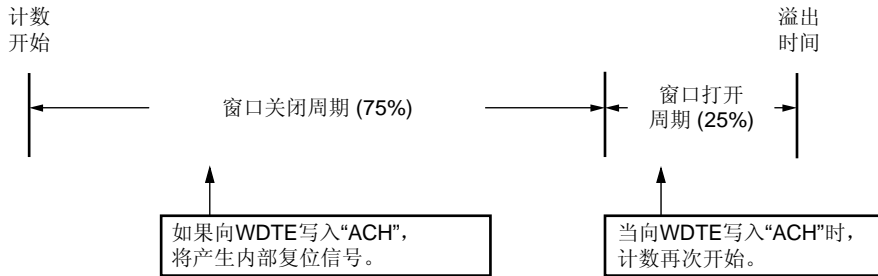
备注 1. f_{RL} : 内部低速振荡时钟频率
2. (): $f_{RL} = 264 \text{ kHz}$ (最大)

10.4.3 设置看门狗定时器的窗口打开周期

通过使用选项字节（0080H）的第 6 和 5 位（WINDOW1，WINDOW0）设置看门狗定时器的窗口打开周期。窗口概述如下所示。

- 如果在窗口打开期间向 WDTE 写入“ACH”，将清除看门狗定时器，并且再次开始计数。
- 在窗口关闭期间，如果向 WDTE 写入“ACH”，将检测到一个异常，并且产生一个内部复位信号。

举例：如果窗口打开周期是 25%



注意事项 复位释放后第一次写 WDTE，将清除看门狗定时器，如果在溢出时间之前写入，看门狗定时器将再次开始计数，而与写入的时序无关。

窗口打开周期的设置如下所示。

表 10-4. 设置看门狗定时器的窗口打开周期

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%

- 注意事项**
1. 禁止同时设置 $WDCS2 = WDCS1 = WDCS0 = 0$ 和 $WINDOW1 = WINDOW0 = 0$ 。
 2. 在 flash 存储器自编程和 EEPROM 仿真过程中，看门狗定时器将继续操作。处理过程中，延迟中断响应时间。设置溢出时间和窗口大小时要将这个延迟考虑在内。

备注 如果溢出时间被设置为 $2^{11}/f_{RL}$ ，窗口关闭时间和打开时间如下所示。

(当 $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ 时)

	窗口打开周期的设置			
	25%	50%	75%	100%
窗口关闭时间	0 到 7.11 ms	0 到 4.74 ms	0 到 2.37 ms	无
窗口打开时间	7.11 到 7.76 ms	4.74 到 7.76 ms	2.37 到 7.76 ms	0 到 7.76 ms

<当窗口打开周期为 25%时>

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

第十一章 A/D转换器

11.1 A/D转换器的功能

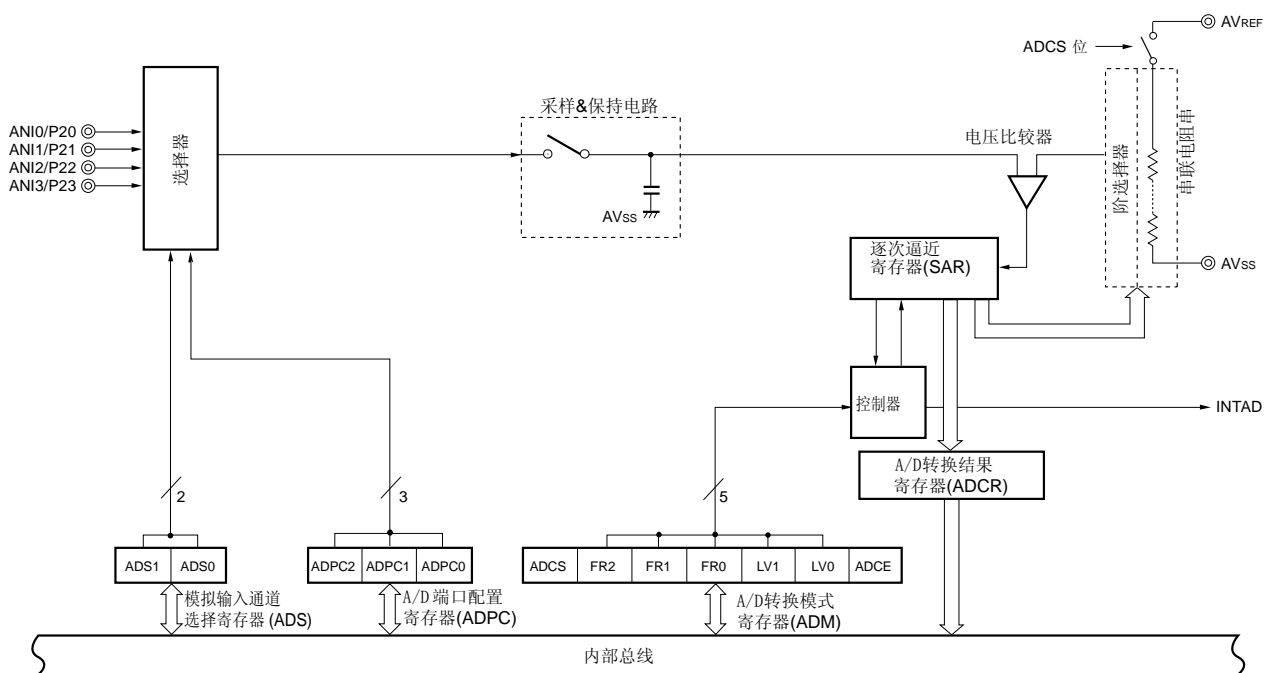
A/D转换器转换模拟输入信号为数字值，4个通道（ANI0至ANI3）10位分辨率。

A/D转换器有以下功能。

• 10位分辨率A/D转换器

重复执行 ANI0 至 ANI3 中选择一个模拟输入通道的 10 位分辨率 A/D 转换。每次一个 A/D 转换结束后，将产生一个中断请求（INTAD）。

图 11-1. A/D 转换器的框图



11.2 A/D转换器的配置

A/D 转换器包含以下硬件。

(1) ANI0 至 ANI3 引脚

这些是 4 通道 A/D 转换器的模拟输入引脚。输入将要转换为数字信号的模拟信号。没有选择为模拟输入的引脚可以用作输入 / 输出引脚。

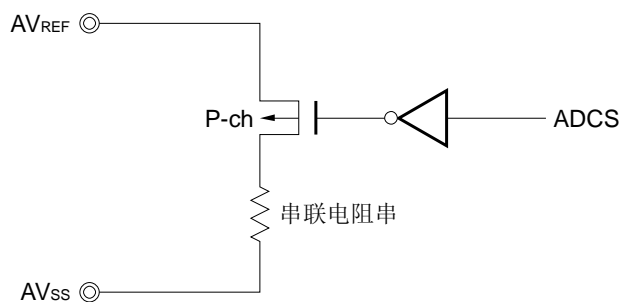
(2) 采样&保持电路

当启动 A/D 转换时，采样&保持电路采样通过选择器选择的模拟输入引脚的输入电压，并且在 A/D 转换过程中保持采样电压值。

(3) 串联电阻串

串联电阻串被连接到 AV_{REF} 和 AV_{SS} 之间，并且产生一个与采样电压值比较的电压。

图 11-2. 串联电阻串的电路配置



(4) 电压比较器

电压比较器比较采样电压值和串联电阻串的输出电压。

(5) 逐次逼近寄存器 (SAR)

该寄存器从最高位 (MSB) 开始转换电压比较器的比较结果。

当电压值的最低位 (LSB) 被转换为数字值时 (A/D 转换结束)，将 SAR 寄存器的内容传送到 A/D 转换结果寄存器 (ADCR)。

(6) 10 位 A/D 转换结果寄存器 (ADCR)

每次完成 A/D 转换后，A/D 转换结果从逐次逼近寄存器加载到该寄存器，并且 ADCR 寄存器使用它的高 10 位 (低 6 位固定为 0) 保存 A/D 转换结果。

(7) 8 位 A/D 转换结果寄存器 (ADCRH)

每次完成 A/D 转换后, A/D 转换结果从逐次逼近寄存器加载到该寄存器, 并且 ADCR 寄存器使用它的高 8 位保存 A/D 转换结果。

注意事项 当从 ADCR 和 ADCRH 读取数据时, 将产生一个等待周期。当停止外围硬件时钟 (fPRS) 时, 不要从 ADCR 和 ADCRH 读取数据。关于详细信息, 参见第二十七章 等待注意事项。

(8) 控制器

该电路控制要转换为数字信号的输入模拟信号的转换时间, 并且启动和停止转换操作。当 A/D 转换完成后, 该控制器产生 INTAD 信号。

(9) AVREF 引脚

该引脚输入一个模拟电源 / 参考电压到 A/D 转换器。当端口 2 用作数字端口时, 使该引脚与 VDD 引脚的电平相同。

根据 AVREF 和 AVSS 之间的电压信号输入到 ANI0 至 ANI3, 并转换为数字信号。

(10) AVSS 引脚

这是 A/D 转换器的地电平引脚。即使没有使用 A/D 转换器时, 该引脚也应与 VSS 的电平相同。

(11) A/D 转换器模式寄存器 (ADM)

该寄存器被用来设置要转换的模拟输入信号的转换时间, 并且启动或停止转换操作。

(12) A/D 端口配置寄存器 (ADPC)

该寄存器将 ANI0 / P20 至 ANI3 / P23 引脚切换为 A/D 转换器的模拟输入或端口的数字输入 / 输出。

(13) 模拟输入通道指定寄存器 (ADS)

该寄存器用于指定要转换为数字信号的模拟电压的输入端口。

(14) 端口模式寄存器 2 (PM2)

该寄存器切换 ANI0 / P20 至 ANI3 / P23 引脚为输入或输出。

11.3 在A/D转换器中使用的寄存器

A/D 转换器使用以下 6 个寄存器。

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

(1) A/D 转换器模式寄存器 (ADM)

该寄存器设置 A/D 转换的模拟输入的转换时间，并且启动 / 停止转换。

ADM 可以通过一个 1 位或 8 位存储器操作指令来设置。

复位信号设置该寄存器为 00H。

图 11-3. A/D 转换器模式寄存器 (ADM) 的格式

地址: FF28H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	FR2 ^{注1}	FR1 ^{注1}	FR0 ^{注1}	LV1 ^{注1}	LV0 ^{注1}	ADCE

ADCS	模/数转换操作控制
0	停止转换操作
1	允许转换操作

ADCE	比较器操作控制 ^{注2}
0	停止比较器操作
1	允许比较器操作

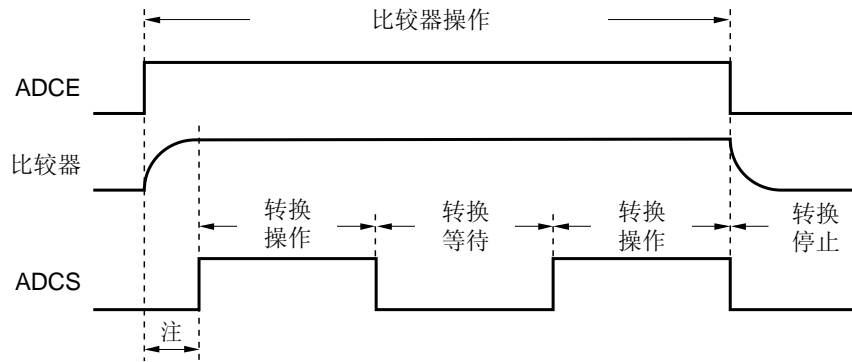
- 注
1. 关于 FR2 至 FR0、LV1、LV0 和 A/D 转换的详细信息，参见表 11-2 A/D 转换时间选择。
 2. ADCS 和 ADCE 控制比较器的操作，并且从操作开始到操作稳定需要 1 μ S。因此，在 ADCE 被设置为 1 后 1 μ S 或更长时间过后，再设置 ADCS 为 1，这时的转换结果比第一次的转换结果具有更高优先级。否则，忽略第一次的转换数据。

表 11-1. ADCS 和 ADCE 的设置

ADCS	ADCE	A/D 转换操作
0	0	停止状态 (不存在直流功耗)
0	1	转换等待模式 (比较器工作，只有比较器产生功耗)
1	0	转换模式 (比较器工作停止 ^注)
1	1	转换模式 (比较器工作 ^注)

注 忽略第一个转换数据。

图 11-4. 使用比较器时的时序图



注 从 ADCE 位的上升到 ADCS 位的下降的时间必须为 1 μ s 或更长，以稳定内部电路。

- 注意事项
1. 在将 FR0 至 FR2、LV1 和 LV0 位修改成其它值之前，必须停止 A/D 转换。
 2. 如果向 ADM 写入数据，将产生一个等待周期。当停止外围硬件时钟 (fPRS) 时，不要向 ADM 写入数据。关于详细信息，参见第二十七章 等待注意事项。

表 11-2. A/D 转换时间选择

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

A/D转换模式寄存器 (ADM)					转换时间选择					转换时钟 (f_{AD})
FR2	FR1	FR0	LV1	LV0		$f_{PRS} = 4\text{ MHz}$	$f_{PRS} = 8\text{ MHz}$	$f_{PRS} = 10\text{ MHz}$	$f_{PRS} = 20\text{ MHz}$	
0	0	0	0	0	$264/f_{PRS}$	$66.0\ \mu\text{s}$	$33.0\ \mu\text{s}$	$26.4\ \mu\text{s}$	$13.2\ \mu\text{s}$	$f_{PRS}/12$
0	0	1	0	0	$176/f_{PRS}$	$44.0\ \mu\text{s}$	$22.0\ \mu\text{s}$	$17.6\ \mu\text{s}$	$8.8\ \mu\text{s}^{\#}$	$f_{PRS}/8$
0	1	0	0	0	$132/f_{PRS}$	$33.0\ \mu\text{s}$	$16.5\ \mu\text{s}$	$13.2\ \mu\text{s}$	$6.6\ \mu\text{s}^{\#}$	$f_{PRS}/6$
0	1	1	0	0	$88/f_{PRS}$	$22.0\ \mu\text{s}$	$11.0\ \mu\text{s}^{\#}$	$8.8\ \mu\text{s}^{\#}$	禁止设置	$f_{PRS}/4$
1	0	0	0	0	$66/f_{PRS}$	$16.5\ \mu\text{s}$	$8.25\ \mu\text{s}^{\#}$	$6.6\ \mu\text{s}^{\#}$		$f_{PRS}/3$
1	0	1	0	0	$44/f_{PRS}$	$11.0\ \mu\text{s}^{\#}$	禁止设置	禁止设置		$f_{PRS}/2$
其它情况					禁止设置					

注 只有当 $4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$ 时, 才可以设置。

(2) $2.3\text{ V} \leq AV_{REF} < 5.5\text{ V}$ ($LV0 = 1$)

A/D转换模式寄存器 (ADM)					转换时间选择				转换时钟 (f_{AD})	
FR2	FR1	FR0	LV1	LV0		$f_{PRS} = 4\text{ MHz}$	$f_{PRS} = 8\text{ MHz}$	$f_{PRS} = 10\text{ MHz}^{\#2}$		$f_{PRS} = 20\text{ MHz}^{\#2}$
0	0	0	0	1	$480/f_{PRS}$	禁止设置	$60.0\ \mu\text{s}$	$48.0\ \mu\text{s}^{\#2}$	$24.0\ \mu\text{s}^{\#2}$	$f_{PRS}/12$
0	0	1	0	1	$320/f_{PRS}$		$40.0\ \mu\text{s}$	$32.0\ \mu\text{s}^{\#2}$	$16.0\ \mu\text{s}^{\#1}$	$f_{PRS}/8$
0	1	0	0	1	$240/f_{PRS}$	$60.0\ \mu\text{s}$	$30.0\ \mu\text{s}$	$24.0\ \mu\text{s}^{\#2}$	$12.0\ \mu\text{s}^{\#1}$	$f_{PRS}/6$
0	1	1	0	1	$160/f_{PRS}$	$40.0\ \mu\text{s}$	$20.0\ \mu\text{s}^{\#1}$	$16.0\ \mu\text{s}^{\#1}$	禁止设置	$f_{PRS}/4$
1	0	0	0	1	$120/f_{PRS}$	$30.0\ \mu\text{s}$	$15.0\ \mu\text{s}^{\#1}$	$12.0\ \mu\text{s}^{\#1}$		$f_{PRS}/3$
1	0	1	0	1	$80/f_{PRS}$	$20.0\ \mu\text{s}^{\#1}$	禁止设置	禁止设置		$f_{PRS}/2$
其它情况					禁止设置					

注 1. 只有当 $4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$ 时, 才可以设置。
 2. 只有当 $2.7\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$ 时, 才可以设置。

注意事项 1. 按照以下条件设置转换时间。

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

- $4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$: $f_{AD} = 0.33$ 至 3.6 MHz
- $2.7\text{ V} \leq AV_{REF} < 4.0\text{ V}$: $f_{AD} = 0.33$ 至 1.8 MHz

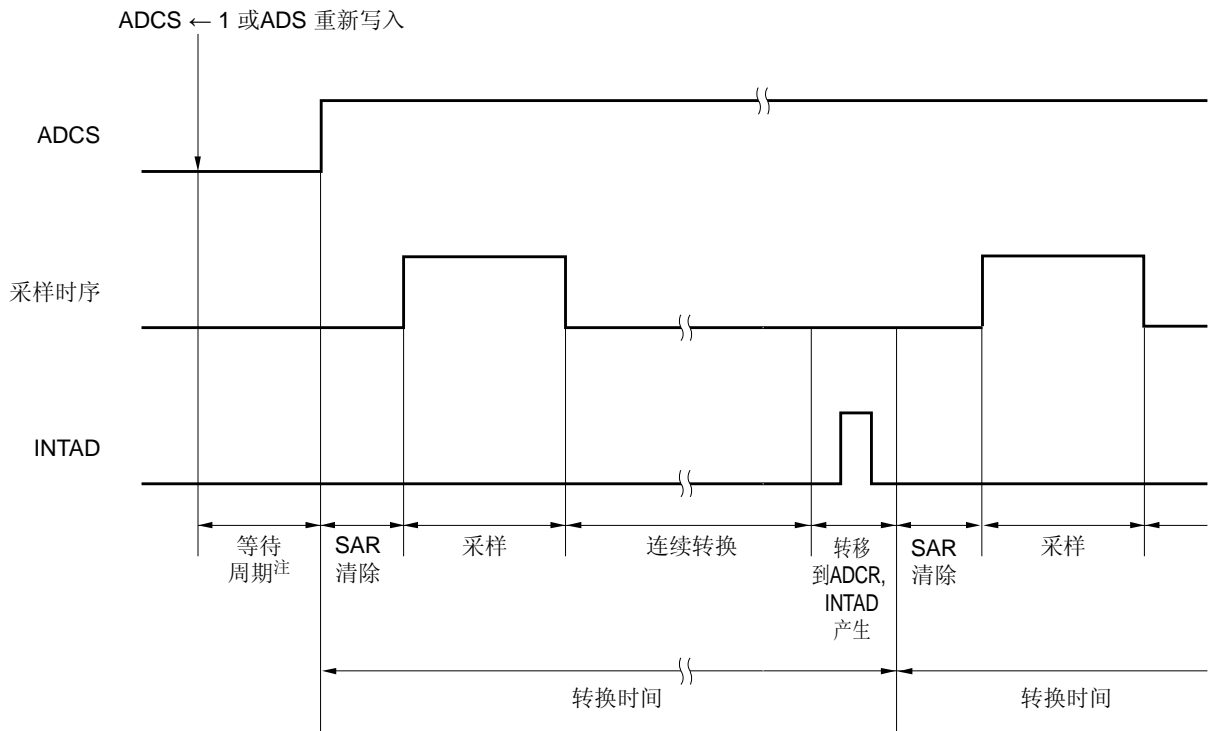
(2) $2.3\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$ ($LV0 = 1$)

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

2. 在修改 FR2 至 FR0、LV1 和 LV0 位为其它值时, 必须停止 A/D 转换 ($ADCS = 0$)。
3. 以上转换时间不包含时钟频率误差。选择转换时间时, 要考虑时钟频率误差。

备注 f_{PRS} : 外围硬件时钟频率

图 11-5. A/D 转换器采样和 A/D 转换时序



注 关于等待周期的详细信息，参见 第二十七章 等待注意事项。

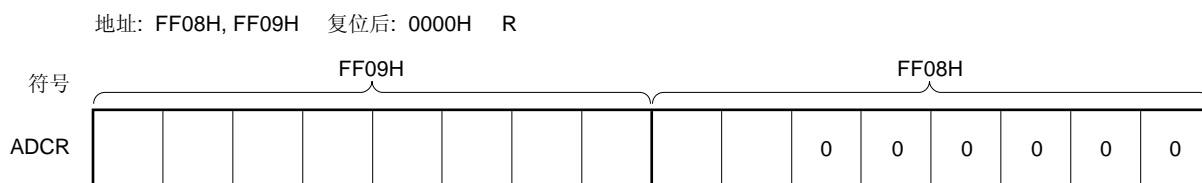
(2) 10 位 A/D 转换结果寄存器 (ADCR)

该寄存器是一个保存 A/D 转换结果的 16 位寄存器。低 6 位固定为 0。每次 A/D 转换结束，从逐次逼近寄存器加载转换结果。转换结果的高 8 位被保存到 FF09H 中，低 2 位被保存到 FF08H 的低 2 位中。

ADCR 可以通过一个 16 位存储器操作指令来读取。

复位信号设置该寄存器为 0000H。

图 11-6. 10 位 A/D 转换结果寄存器 (ADCR) 的格式



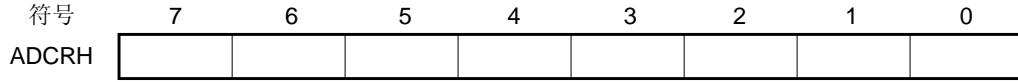
- 注意事项**
1. 当写入 A/D 转换器模式寄存器 (ADM)、模拟输入通道指定寄存器 (ADS) 和 A/D 端口配置寄存器 (ADPC) 时，ADCR 的内容可能变为不确定。在转换完成后，并且写入 ADM、ADS 和 ADPC 前读取转换结果。使用上面以外的时序，可能会读取到不正确的转换结果。
 2. 如果从 ADCR 读取数据，将产生一个等待周期。当外围硬件时钟 (fPRS) 被停止时，不要从 ADCR 读取数据。关于详细信息，参见 第二十七章 等待注意事项。

(3) 8 位 A/D 转换结果寄存器 (ADCRH)

该寄存器是一个保存 A/D 转换结果的 8 位寄存器。保存 10 位分辨率结果的高 8 位。
ADCRH 可以通过一个 8 位存储器操作指令来读取。
复位信号设置该寄存器为 00H。

图 11-7. 8 位 A/D 转换结果寄存器 (ADCRH) 的格式

地址: FF09H 复位后: 00H R



- 注意事项**
1. 当写入 A/D 转换器模式寄存器 (ADM)、模拟输入通道指定寄存器 (ADS) 和 A/D 端口配置寄存器 (ADPC) 时, ADCRH 的内容可能变为不确定。在转换完成后, 并且写入 ADM、ADS 和 ADPC 前读取转换结果。使用上面以外的时序, 可能会读取到不正确的转换结果。
 2. 如果从 ADCRH 读取数据, 将产生一个等待周期。当停止外围硬件时钟 (fPRS) 时, 不要从 ADCRH 读取数据。关于详细信息, 参见 第二十七章 等待注意事项。

(4) 模拟输入通道指定寄存器 (ADS)

该寄存器指定要 A/D 转换的模拟电压的输入通道。

ADS 可以通过一个 1 位或 8 位存储器操作指令来设置。

复位信号设置该寄存器为 00H。

图 11-8. 模拟输入通道指定寄存器 (ADS) 的格式

地址: FF29H 复位后: 00H R/W

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

ADS1	ADS0	模拟输入通道指定
0	0	ANI0
0	1	ANI1
1	0	ANI2
1	1	ANI3

- 注意事项
1. 确保清除第 2 至 7 位为 0。
 - 2 通过使用端口模式寄存器 2 (PM2) 来设置要用作 A/D 转换的通道为输入模式。
 3. 如果向 ADS 写入数据, 将产生一个等待周期。当停止外围硬件时钟 (fPRS) 时, 不要向 ADS 写入数据。关于详细信息, 参见第二十七章 等待注意事项。

(5) A/D 端口配置寄存器 (ADPC)

该寄存器切换 ANI0 / P20 至 ANI3 / P23 引脚为 A/D 转换器的模拟输入或端口的数字输入 / 输出。

ADPC 可以通过一个 1 位或 8 位存储器操作指令来设置。

复位信号设置该寄存器为 00H。

图 11-9. A/D 端口配置寄存器 (ADPC) 的格式

地址: FF2FH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADPC	0	0	0	0	0	ADPC2	ADPC1	ADPC0

ADPC2	ADPC1	ADPC0	数字输入/输出 (D)/模拟输入 (A)切换			
			P23/ANI3	P22/ANI2	P21/ANI1	P20/ANI0
0	0	0	A	A	A	A
0	0	1	A	A	A	D
0	1	0	A	A	D	D
0	1	1	A	D	D	D
1	0	0	D	D	D	D
除上面以外			禁止设置			

- 注意事项
1. 确保清除第 3 至 7 位为 0。
 2. 通过使用端口模式寄存器 2 (PM2) 来设置要用作 A/D 转换的通道为输入模式。
 3. 如果向 ADPC 写入数据, 将产生一个等待周期。当停止外围硬件时钟 (fPRS) 时, 不要向 ADPC 写入数据。关于详细信息, 参见 第二十七章 等待注意事项。

(6) 端口模式寄存器 2 (PM2)

当使用 ANI0 / P20 至 ANI3 / P23 引脚作为模拟输入端口时，设置 PM20 至 PM23 为 1。P20 至 P23 的输出锁存器这时可能为 0 或 1。

如果设置 PM20 至 PM23 为 0，它们不能被用作模拟输入端口引脚。

PM2 可以通过一个 1 位或 8 位存储器操作指令来设置。

复位信号设置该寄存器为 FFH。

图 11-10. 端口模式寄存器 2 (PM2) 格式

地址: FF22H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM2	1	1	1	1	PM23	PM22	PM21	PM20

PM2n	P2n 引脚输入/输出模式选择 (n = 0 到 3)
0	输出模式(输出缓冲器开)
1	输入模式(输出缓冲器关)

ANI0/P20 至 ANI3/P23 引脚依赖于 ADPC、ADS 和 PM2 的设置，如下所示。

表 11-3. 设置 ANI0 / P20 至 ANI3 / P23 引脚的功能

ADPC	PM2	ADS	ANI0/P20 至 ANI3/P23 引脚
模拟输入选择	输入模式	选择ANI.	模拟输入 (要被转换)
		不选择 ANI.	模拟输入 (不被转换)
	输出模式	选择ANI.	禁止设置
		不选择 ANI.	
数字输入 / 输出选择	输入模式	-	数字输入
	输出模式	-	数字输出

11.4 A/D转换器操作

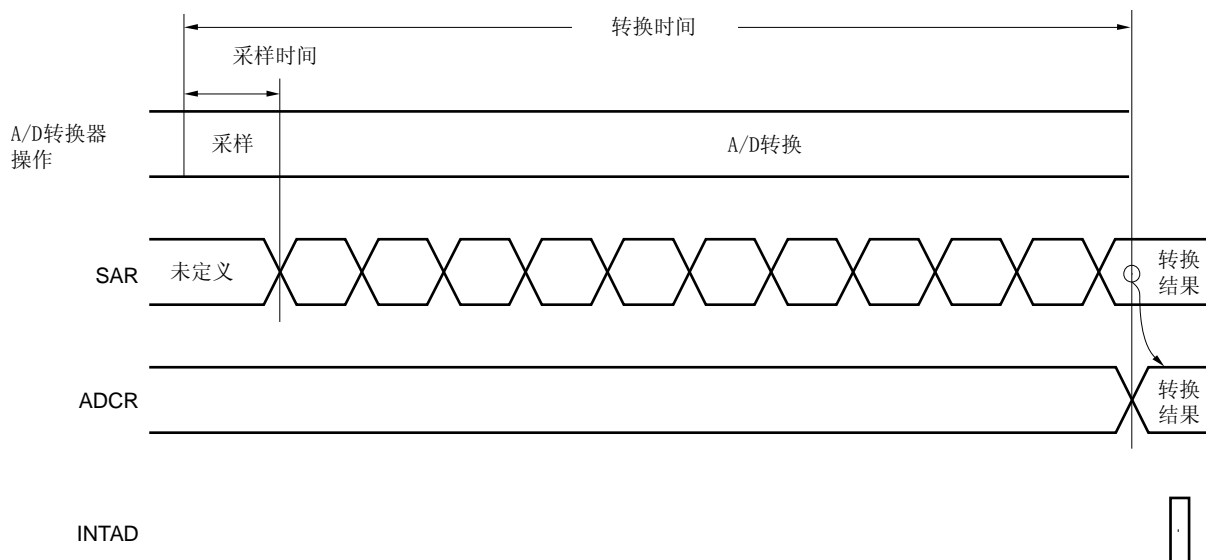
11.4.1 A/D转换器的基本操作

- <1> 设置 A/D 转换器模式寄存器 (ADM) 的第 0 位 (ADCE) 为 1 来启动比较器的操作。
- <2> 通过使用 A/D 端口配置寄存器 (ADPC) 来为 A/D 转换设置模拟输入通道, 并且通过使用端口模式寄存器 2 (PM2) 设置为输入模式。
- <3> 通过使用 ADM 的第 5 至 1 位 (FR2 至 FR0、LV1 和 LV0) 设置 A/D 转换时间。
- <4> 使用模拟输入通道指定寄存器 (ADS) 为 A/D 转换选择一个通道。
- <5> 通过设置 ADM 的第 7 位 (ADCS) 为 1 启动转换操作。
(<6> 至 <12> 是硬件执行的操作。
- <6> 采样&保持电路采样所选择的模拟输入通道的电压。
- <7> 当采样完成一定时间后, 采样&保持电路被置于保持状态, 并且采样的电压一直保持到 A/D 转换操作结束。
- <8> 设置逐次逼近寄存器 (SAR) 的第 9 位。通过分接选择器, 串联电阻串分接电压设置为 $(1/2) AV_{REF}$ 。
- <9> 通过电压比较器, 比较串联电阻串阶和采样电压之间的差别。如果模拟输入比 $(1/2) AV_{REF}$ 大, SAR 的 MSB 保持设置为 1。如果模拟输入比 $(1/2) AV_{REF}$ 小, MSB 被复位为 0。
- <10> 然后, SAR 的第 8 位自动被设置为 1, 并且处理下一个比较。按照第 9 位的预设值选择串联电阻串电压阶, 如下所示。
 - 第 9 位 = 1: $(3/4) AV_{REF}$
 - 第 9 位 = 0: $(1/4) AV_{REF}$
 比较分接电压和采样电压, 并且按照下面修改 SAR 的第 8 位。
 - 模拟输入电压 \geq 分接电压: 第 8 位 = 1
 - 模拟输入电压 $<$ 分接电压: 第 8 位 = 0
- <11> 按照这种方式, 继续比较, 直到 SAR 的第 0 位。
- <12> 一旦 10 位比较完成, 有效的结果值仍然在 SAR 中, 同时结果值被传送到 A/D 转换结果寄存器 (ADCR, ADCRH), 并锁存。
这时, 产生 A/D 转换结束中断请求 (INTAD)。
- <13> 重复步骤<6> 至 <12>, 直到 ADCS 被清除为 0。
要停止 A/D 转换器, 清除 ADCS 为 0。
要从 ADCE = 1 的状态重新启动 A/D 转换器, 从<5>开始。当 ADCE = 0 时, 要再次启动 A/D 转换器, 设置 ADCE 为 1, 等待 $1 \mu s$ 或更长时间, 启动<5>。要更改 A/D 转换的通道, 从<4>开始。

注意事项 确保<1> 至 <5>的周期为 $1 \mu s$ 或更长。

- 备注** 两种类型的 A/D 转换结果寄存器可以使用。
- ADCR (16 位): 保存 10 位 A/D 转换值
 - ADCRH (8 位): 保存 8 位 A/D 转换值

图 11-11. A/D 转换器的基本操作



持续执行 A/D 转换操作，直到 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 通过软件被复位 (0)。

如果在 A/D 转换操作过程中执行一个向模拟输入通道指定寄存器 (ADS) 的写操作，将初始化转换操作，并且如果 ADCS 位被置位 (1)，转换操作重新开始。

复位信号设置 A/D 转换结果寄存器 (ADCR, ADCRH) 为 0000H 或 00H。

11.4.2 输入电压和转换结果

输入到模拟输入引脚（ANI0 至 ANI3）的模拟输入电压和理论的 A/D 转换结果（被保存在 10 位 A/D 转换结果寄存器（ADCR）中）之间的关系通过以下表达式表示。

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

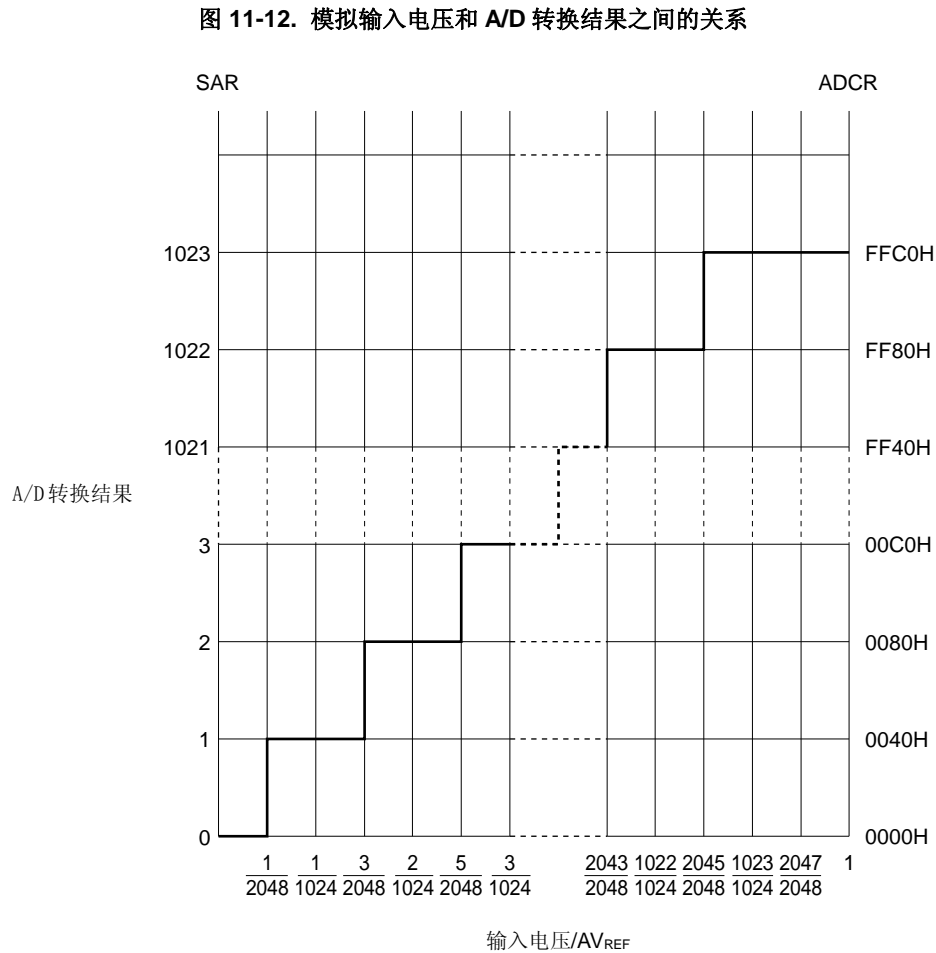
$$ADCR = SAR \times 64$$

或者

$$\left(\frac{ADCR}{64} - 0.5 \right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5 \right) \times \frac{AV_{REF}}{1024}$$

- 其中，INT(): 返回括号中的值的整数部分的函数
- V_{AIN} : 模拟输入电压
- AV_{REF} : AV_{REF} 引脚电压
- ADCR: A/D 转换结果寄存器（ADCR）的值
- SAR: 逐次逼近寄存器

图 11-12 表示模拟输入电压和 A/D 转换结果之间的关系。



11.4.3 A/D转换器工作模式

A/D 转换器的工作模式是选择模式。通过模拟输入通道指定寄存器 (ADS)，从 ANI0 至 ANI3 中选择一个模拟输入通道，并且执行 A/D 转换。

(1) A/D 转换操作

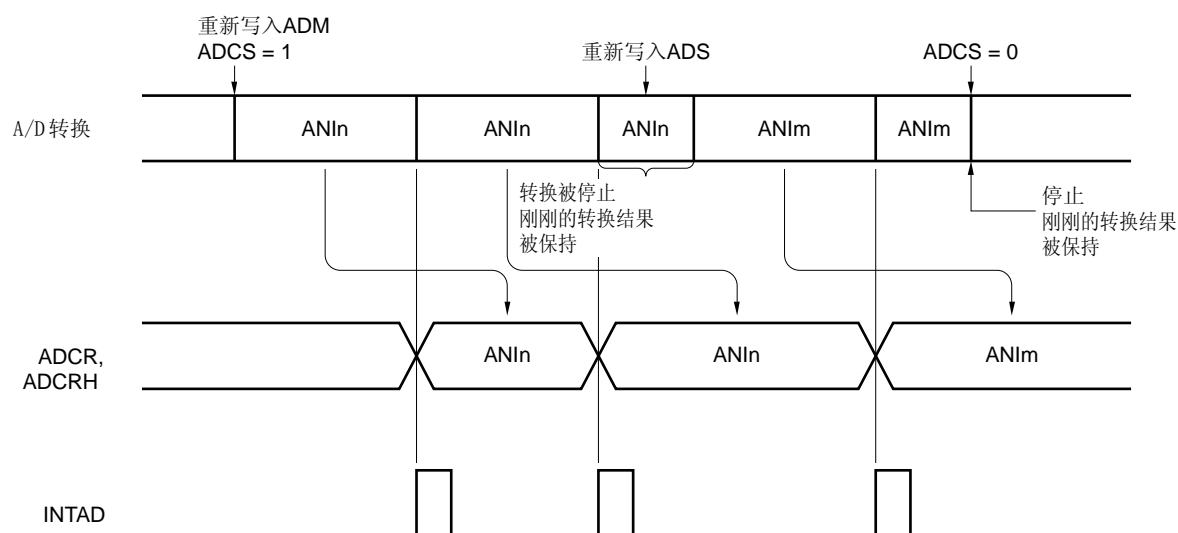
通过设置 A/D 转换器模式寄存器 (ADM) 的第 7 位为 1，启动模拟输入通道指定寄存器 (ADS) 指定的模拟输入引脚上的电压的 A/D 转换操作。

当 A/D 转换被完成后，A/D 转换的结果保存到 A/D 转换结果寄存器 (ADCR) 中，并且产生一个中断请求信号 (INTAD)。当一个 A/D 转换完成后，立即启动下一个 A/D 转换操作。

如果在 A/D 转换过程中向 ADS 写入 0，将停止正在执行的 A/D 转换操作，并且从开始处重新启动。

如果在 A/D 转换过程中向 ADCS 写入 0，将立即停止 A/D 转换。这时，保持之前的转换结果。

图 11-13. A/D 转换操作



- 备注
1. n = 0 至 3
 2. m = 0 至 3

设置方法如下所示。

- <1> 设置 A/D 转换器模式寄存器 (ADM) 的第 0 位 (ADCE) 为 1。
 - <2> 通过使用 A/D 端口配置寄存器 (ADPC) 的第 2 至 0 位 (ADPC2 至 ADPC0) 和端口模式寄存器 2 (PM2) 的第 3 至 0 位 (PM23 至 PM20) 设置在模拟输入模式中要使用的通道。
 - <3> 通过使用 ADM 的第 5 至 0 位 (FR2 至 FR0、LV1 和 LV0) 选择转换时间。
 - <4> 通过使用模拟输入通道指定寄存器 (ADS) 的第 1 和 0 位 (ADS1 和 ADS0) 选择一个要使用的通道。
 - <5> 设置 ADM 的第 7 位 (ADCS) 启动 A/D 转换。
 - <6> 当 A/D 转换被完成后, 将产生一个中断请求信号 (INTAD)。
 - <7> 将 A/D 转换数据传送到 A/D 转换结果寄存器 (ADCR, ADCRH) 中。
- <更改通道>
- <8> 使用 ADS 的第 1 和 0 位 (ADS1 和 ADS0) 更改通道并启动 A/D 转换。
 - <9> 当 A/D 转换被完成后, 将产生一个中断请求信号 (INTAD)。
 - <10> 将 A/D 转换数据传送到 A/D 转换结果寄存器 (ADCR, ADCRH) 中。
- <完成 A/D 转换>
- <11> 清除 ADCS 为 0。
 - <12> 清除 ADCE 为 0。

- 注意事项**
1. 确保<1>至<5>的周期为 1 μ s 或更长。
 2. <1>可在<2>和<4>之间完成。
 3. 可以省略<1>。然而, 在这种情况下, 忽略<5>后面第一个转换的数据。
 4. <6>至<9>的周期与使用 ADM 的第 5 至 1 位 (FR2 至 FR0, LV1, LV0) 设置的转换时间不同。<8>至<9>的周期是使用 FR2 至 FR0、LV1 和 LV0 设置的转换时间。

11.5 如何阅读A/D转换器特性表

这里，解释 A/D 转换器的专业术语。

(1) 分辨率

这是可以被识别的最小模拟输入电压。数字输出的每位表示的模拟输入电压的百分比叫做 1LSB（最低位）。1LSB 在满幅中占的百分比表示为%FSR（满幅范围）。

当分辨率为 10 位时，1LSB 如下所示。

$$1\text{LSB} = 1/2^{10} = 1/1024 \\ = 0.098\%\text{FSR}$$

精度与分辨率无关，而是由总误差确定。

(2) 总误差

这表示实际测量值和理论值之间的最大误差值。

零度误差、满度误差、积分线性误差和微分线性误差的组合表示总误差。

注意，在特性表中量化误差未包含在总误差内。

(3) 量化误差

当模拟值被转换为数字值时，通常会产生 $\pm 1/2\text{LSB}$ 的误差。在 A/D 转换器中，相差 $\pm 1/2\text{LSB}$ 的模拟输入电压被转换为同样的数字码，所以量化误差无法避免。

注意，在特性表中，量化误差未被包含在总误差、零度误差、满度误差、积分线性误差和微分线性误差中。

图 11-14. 总误差

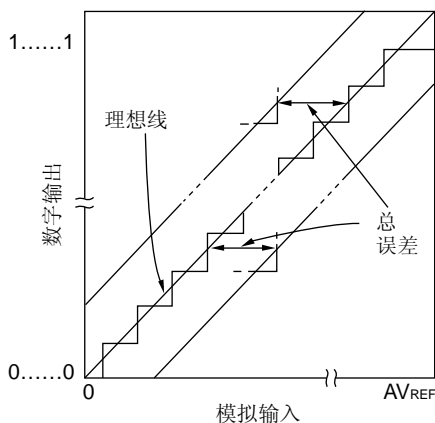
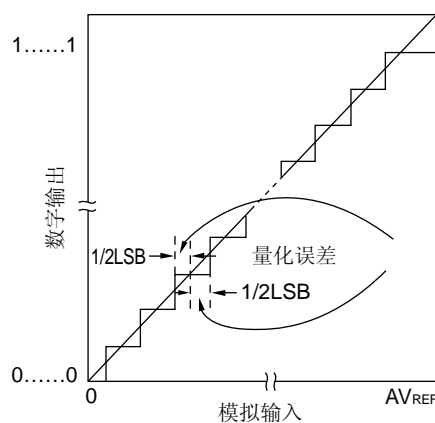


图 11-15. 量化误差



(4) 零度误差

这表示当数字输出从 0.....000 更改为 0.....001 时模拟输入电压的实际测量值和理论值之间的差别（ $1/2\text{LSB}$ ）。

如果实际测量值比理论值大，它表示当数字输出从 0..... 001 更改为 0..... 010 时模拟输入电压的实际测量值和理论值之间的差别（ $3/2\text{LSB}$ ）。

(5) 满度误差

这表示当数字输出从 1.....110 更改为 1.....111 时模拟输入电压的实际测量值和理论值之间的差别（满幅 - 3/2LSB）。

(6) 积分线性误差

这表示转换特性偏离理想线性关系的程度。它表示当零度误差和满度误差为 0 时，实际测量值和理想直线之间的最大误差值。

(7) 微分线性误差

当代码输出的理想宽度为 1LSB 时，这表示实际测量值和理想值之间的差别。

图 11-16. 零度误差

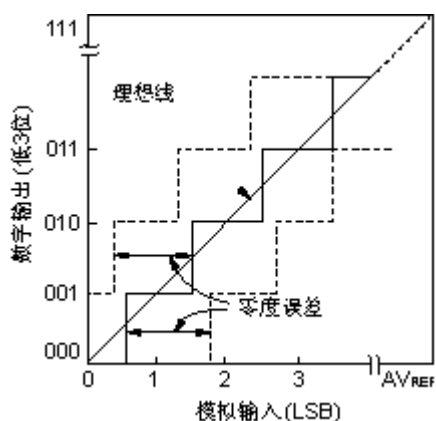


图 11-17. 满度误差

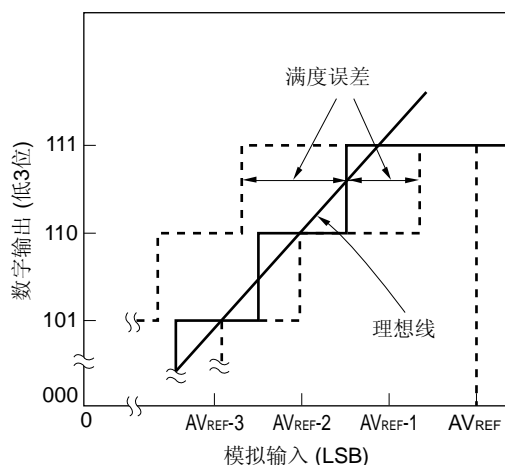


图 11-18. 积分线性误差

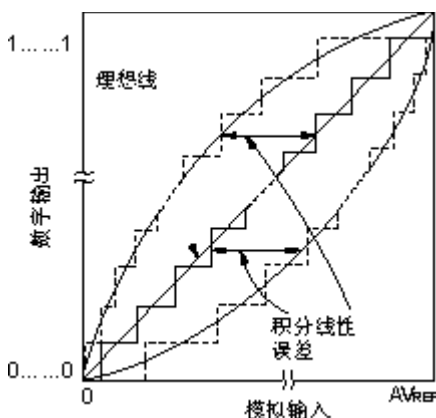
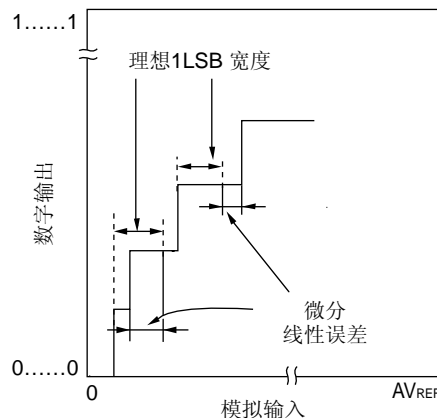


图 11-19. 微分线性误差

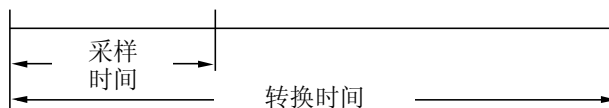


(8) 转换时间

这表示从采样启动到数字输出所获得的时间。在特性表中，转换时间中包含了采样时间。

(9) 采样时间

这是为要被采样&保持电路采样的模拟电压打开模拟开关的时间。



11.6 A/D转换器的注意事项

(1) 在 STOP 模式下的工作电流

A/D 转换器在 STOP 模式下停止工作。这时，通过清除 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 和第 0 位 (ADCE) 为 0，可以减少工作电流。

要从待机状态重新启动，清除中断请求标志寄存器 1L (IF1L) 的第 0 位 (ADIF) 为 0 并启动操作。

(2) ANI0 至 ANI3 的输入范围

遵守 ANI0 至 ANI3 输入电压的额定范围。如果 AV_{REF} 或更高的电压或者 AV_{SS} 或更低的电压（即使在绝对最大额定范围内）输入到模拟输入通道，那个通道的转换值变为不确定。此外，其它通道的转换值也可能受影响。

(3) 冲突操作

<1> 在转换结束时通过指令写入 A/D 转换结果寄存器 (ADCR, ADCRH) 和读取 ADCR 或 ADCRH 之间的冲突 ADCR 或 ADCRH 读操作优先级高。读取操作后，新的转换值被写入 ADCR 或 ADCRH。

<2> 在转换结束时写入 ADCR 或 ADCRH、写入 A/D 转换器模式寄存器 (ADM)、模拟输入通道指定寄存器 (ADS) 或写入 A/D 端口配置寄存器 (ADPC) 之间的冲突 ADM、ADS 或 ADPC 写入优先级高。不执行 ADCR 或 ADCRH 的写入，也不产生转换结束中断信号 (INTAD)。

(4) 噪声对策

要维持 10 位分辨率，必须注意 AV_{REF} 引脚和 ANI0 至 ANI3 上输入的噪声。

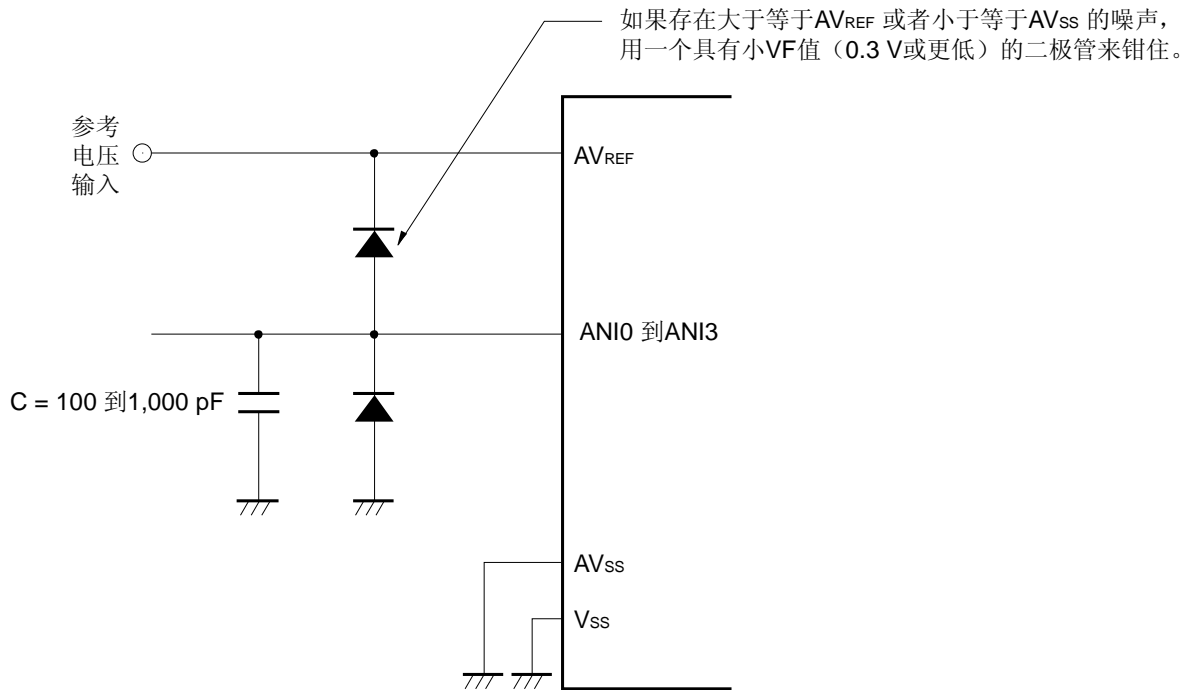
<1> 将一个具有低等效电阻和优质频率响应的电容连接到电源。

<2> 模拟输入源的输出阻抗越大，干扰越大。要减少噪声，建议如图 11-20 连接外部电容，。

<3> 在转换过程中不要切换这些引脚和其它引脚。

<4> 如果在转换启动后立即设置 HALT 模式，可以提高精度。

图 11-20. 模拟输入引脚连接



(5) AN10/P20 至 AN13/P23

- <1> 模拟输入引脚（AN10 至 AN13）也被用作输入端口引脚（P20 至 P23）。
当执行 AN10 至 AN13 中任意引脚的 A/D 转换时转换过程中不要访问 P20 至 P23；否则，转换分辨率可能会降低。建议从距离 AV_{REF} 最远的 AN10/P20 开始选择引脚用作 P20 至 P23。
- <2> 如果当前用作 A/D 转换的引脚的邻近引脚有数字脉冲，由于耦合噪声，可能不会获得期望的 A/D 转换值。因此，不要在处于 A/D 转换的引脚的邻近引脚上使用脉冲。

(6) AN10 至 AN13 引脚的输入阻抗

采样时间内，A/D 转换器对一个采样电容充电。

因此，当采样没有处理时只有漏电流流动，并且采样过程中有对电容充电的电流。从而，输入阻抗根据采样是否处理和其它状态而波动。

要确保采样有效，建议保持模拟输入源的输出阻抗在 10 k Ω 以内，并连接一个大约 100 pF 的电容到 AN10 至 AN13 引脚（见图 11-20）。

(7) AV_{REF} 引脚输入阻抗

一个几十 k Ω 的串联电阻串连接到 AV_{REF} 和 AV_{SS} 引脚之间。

因此，如果参考电压源的输出阻抗很高，这将导致 AV_{REF} 和 AV_{SS} 引脚之间的一个串联电阻串，导致一个大的参考电压误差。

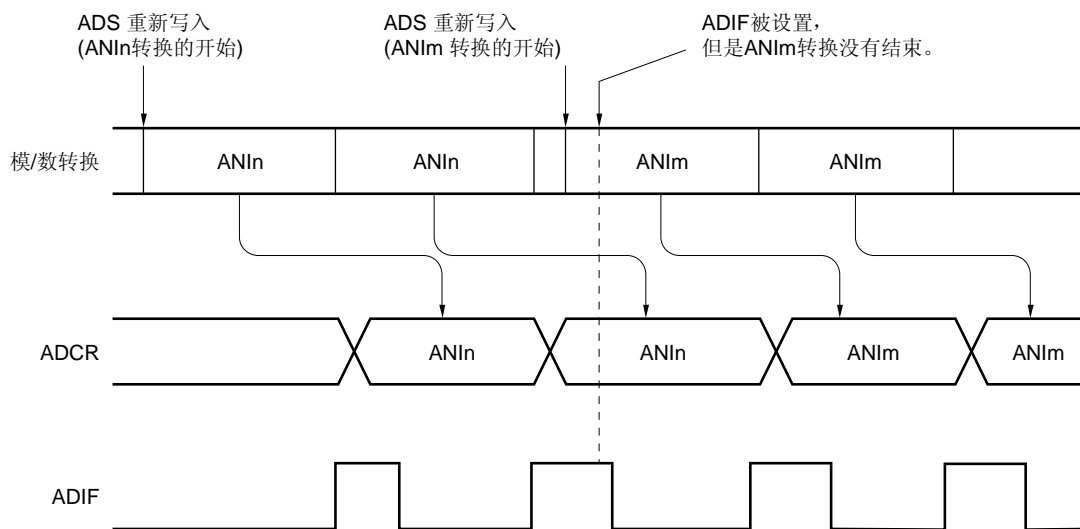
(8) 中断请求标志 (ADIF)

即使更改模拟输入通道指定寄存器 (ADS)，也不会清除中断请求标志 (ADIF)。

因此，如果在 A/D 转换过程中更改模拟输入引脚，可能在刚刚重写 ADS 前，设置了更改前的模拟输入的 A/D 转换结果和 ADIF。这时需要注意，因为在 ADS 被重新写入后立即读取 ADIF 时，尽管更改后的模拟输入的 A/D 转换没有结束，也会设置 ADIF。

当停止 A/D 转换并且然后重新开始时，在 A/D 转换操作重新开始前清除 ADIF。

图 11-21. A/D 转换结束中断请求产生时序



- 备注
1. n = 0 至 3
 2. m = 0 至 3

(9) 在 A/D 转换刚刚开始后的转换结果

如果在 ADCE 位被设置为 1 后的 1 μ s 内 ADCS 位被设置为 1 或者如果 ADCE 位 = 0 时 ADCS 位被设置为 1，在 A/D 转换开始后的第一个 A/D 转换值可能不会落入额定范围内。采取措施，比如检测 A/D 转换结束中断请求 (INTAD) 并清除第一个转换结果。

(10) A/D 转换结果寄存器 (ADCR, ADCRH) 读取操作

当对于 A/D 转换器模式寄存器 (ADM)、模拟输入通道指定寄存器 (ADS) 和 A/D 端口配置寄存器 (ADPC) 的一个写入操作被执行时，ADCR 和 ADCRH 的内容可能变为不确定。在转换完成后并且写入 ADM、ADS 和 ADPC 前读取转换结果。使用上面以外的时序可能导致错误的转换结果被读取。

(11) 内部等效电路

模拟输入模块的等效电路表示如下。

图 11-22. ANIn 引脚的内部等效电路

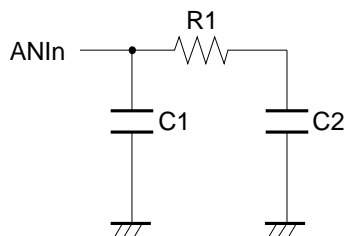


表 11-4. 等效电路的电阻和电容值 (参考值)

AV_{REF}	R1	C1	C2
$4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$	8.1 k Ω	8 pF	5 pF
$2.7\text{ V} \leq AV_{REF} < 4.0\text{ V}$	31 k Ω	8 pF	5 pF
$2.3\text{ V} \leq AV_{REF} < 2.7\text{ V}$	381 k Ω	8 pF	5 pF

- 备注
1. 表 11-4 中显示的电阻和电容值是不被保证的值。
 2. n = 0 至 3

12.1 串行接口UART0 的功能

串行接口 UART0 有以下两种模式。

(1) 操作停止模式

未执行串行通信时使用这个模式，可以降低功耗。

关于详细信息，参见 12.4.1 操作停止模式。

(2) 异步串行接口 (UART) 模式

这个模式的功能概述如下。

关于详细信息，参见 12.4.2 异步串行接口 (UART) 模式和 12.4.3 专用波特率发生器。

- 最大传输速率：625 kbps
- 两个引脚配置 TxD0：发送数据输出引脚
RxD0：接收数据输入引脚
- 通信数据的数据长度可选择 7 位或 8 位。
- 专用的片上 5 位波特率发生器允许设置任意波特率。
- 发送和接收可以独立执行（全双工操作）。
- 固定为 LSB 在前通信

注意事项 1. 如果未停止串行接口 UART0 的时钟（例如，在 HALT 模式下），则继续正常操作。如果停止提供给串行接口 UART0 的时钟（例如，在 STOP 模式下），则每个寄存器停止工作，并且保持时钟刚刚停止前的值。TxD0 引脚也保持时钟刚刚停止前的值并输出。然而，时钟重新开始后，不能保证操作的可靠性。因此，使 POWER0 = 0、RXE0 = 0 和 TXE0 = 0，以复位电路。

2. 设置 POWER0 = 1，然后设置 TXE0 = 1（发送）或 RXE0 = 1（接收）以开始通信。

3. TXE0 和 RXE0 与由 BRGC0 设置的基本时钟（f_{XCLK0}）同步。要再次允许发送或接收，在 TXE0 或 RXE0 已经被清除为 0 后至少两个基本时钟周期后，再设置 TXE0 或 RXE0 为 1。如果在基本时钟的两个周期内设置 TXE0 或 RXE0，发送电路或接收电路可能没有被初始化。

4. 在设置 TXE0 = 1 后至少一个基本时钟（f_{XCLK0}）周期后，将待发送的数据设置到 TXS0。

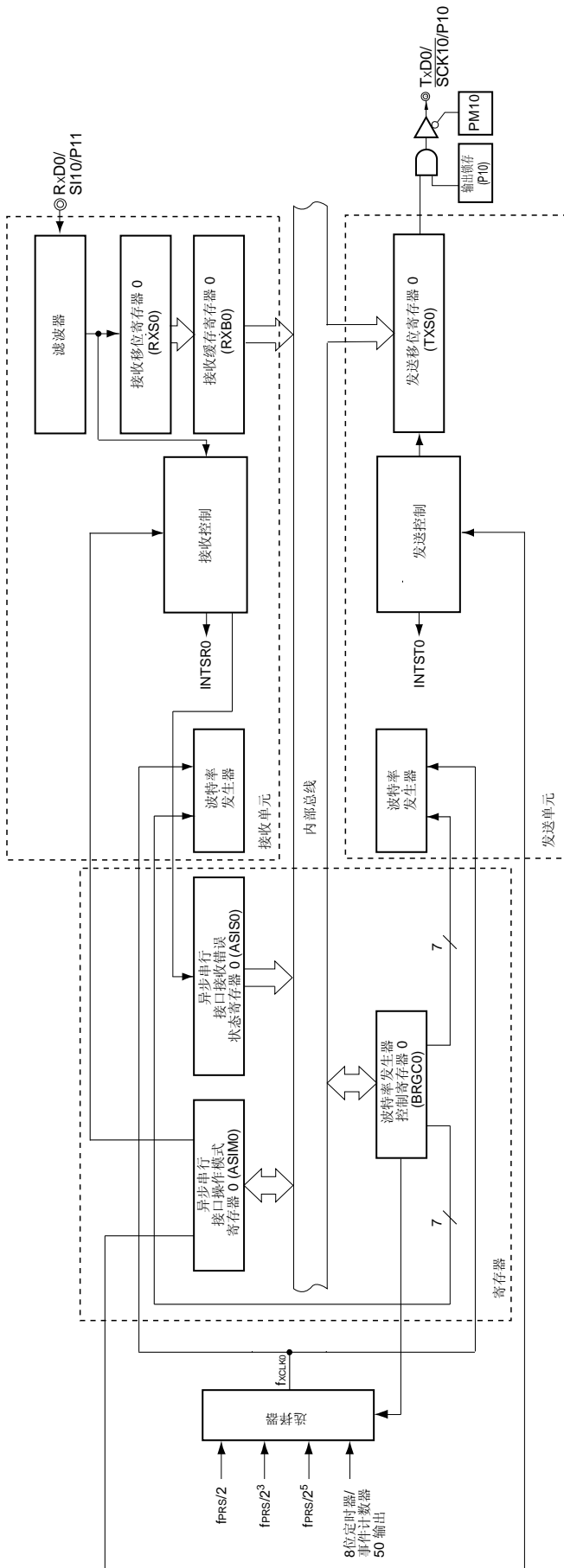
12.2 串行接口UART0 的配置

串行接口 UART0 包含以下硬件。

表 12-1. 串行接口 UART0 的配置

项目	配置
寄存器	接收缓冲寄存器 0 (RXB0) 接收移位寄存器 0 (RXS0) 发送移位寄存器 0 (TXS0)
控制寄存器	异步串行接口操作模式寄存器 0 (ASIM0) 异步串行接口接收错误状态寄存器 0 (ASIS0) 波特率发生器控制寄存器 0 (BRGC0) 端口模式寄存器 1 (PM1) 端口寄存器 1 (P1)

图 12-1. 串行接口 UART0 的框图



(1) 接收缓冲寄存器 0 (RXB0)

该 8 位寄存器存储由接收移位寄存器 0 (RXS0) 转换的并行数据。

每次接收 1 字节数据，新接收的数据从接收移位寄存器 0 (RXS0) 传送到该寄存器。

如果设置数据长度为 7 位，接收的数据被传送到 RXB0 的第 0 至 6 位，并且 RXB0 的 MSB 总是 0。

如果发生溢出错误 (OVE0)，接收的数据不会被传送到 RXB0。

RXB0 可以由 8 位存储器操作指令读取。不能向该寄存器写入数据。

复位信号的产生和 POWER0 = 0 设置该寄存器为 FFH。

(2) 接收移位寄存器 0 (RXS0)

该寄存器将输入到 RxD0 引脚的串行数据转换为并行数据。

RXS0 不能由程序直接操作。

(3) 发送移位寄存器 0 (TXS0)

该寄存器用于设置发送数据。当数据被写入到 TXS0 时，启动发送，并且从 TxD0 引脚发送串行数据。

TXS0 可以由 8 位存储器操作指令来写入。不能读取该寄存器。

复位信号的产生、POWER0 = 0 和 TXE0 = 0 设置该寄存器为 FFH。

- 注意事项**
1. 在设置 TXE0 = 1 后至少一个基本时钟 (f_{xCLK0}) 周期后，再将待发送的数据设置到 TXS0。
 2. 在产生发送完成中断信号 (INTST0) 前，不要向 TXS0 写入下一个待发送的数据。

12.3 控制串行接口UART0 的寄存器

串行接口 UART0 由以下五个寄存器控制。

- 异步串行接口操作模式寄存器 0 (ASIM0)
- 异步串行接口接收错误状态寄存器 0 (ASIS0)
- 波特率发生器控制寄存器 0 (BRGC0)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

(1) 异步串行接口操作模式寄存器 0 (ASIM0)

该 8 位寄存器控制串行接口 UART0 的串行通信操作。

该寄存器可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生设置该寄存器为 01H。

图 12-2. 异步串行接口操作模式寄存器 0 (ASIM0) 的格式 (1 / 2)

地址: FF70H 复位后: 01H R / W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	允许 / 禁止内部操作时钟的操作
0 ^{#1}	禁止内部操作时钟的操作（固定时钟为低电平）并且异步复位内部电路 ^{#2} 。
1	允许内部操作时钟的操作

TXE0	允许 / 禁止发送
0	禁止发送（同步复位发送电路）。
1	允许发送

RXE0	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. 当 POWER0 = 0 时，RxD0 引脚的输入恒为高电平。
 2. 异步串行接口接收错误状态寄存器 0 (ASIS0)、发送移位寄存器 0 (TXS0) 和接收缓冲寄存器 0 (RXB0) 被复位。

图 12-2. 异步串行接口操作模式寄存器 0 (ASIM0) 的格式 (2 / 2)

PS01	PS00	发送操作	接收操作
0	0	不输出校验位。	无校验位接收
0	1	输出零校验。	以零校验接收 [#]
1	0	输出奇校验。	按照奇校验判断。
1	1	输出偶校验。	按照偶校验判断。

CL0	指定发送 / 奇数数据的字符长度
0	数据的字符长度 = 7 位
1	数据的字符长度 = 8 位

SL0	指定发送数据的停止位个数
0	停止位个数 = 1
1	停止位个数 = 2

注 如果选择“以零校验接收”，不判断校验位。因此，不设置异步串行接口接收错误状态寄存器 0 (ASIS0) 的第 2 位 (PE0)，且不会发生错误中断。

- 注意事项**
1. 要开始发送，设置 POWER0 为 1 然后设置 TXE0 为 1。要停止发送，清除 TXE0 为 0，并且然后清除 POWER0 为 0。
 2. 要开始接收，设置 POWER0 为 1 然后设置 RXE0 为 1。要停止接收，清除 RXE0 为 0，并且然后清除 POWER0 为 0。
 3. 当一个高电平输入到 RxD0 引脚时，设置 POWER0 为 1，然后设置 RXE0 为 1。如果输入低电平时，设置 POWER0 为 1，并且设置 RXE0 为 1，启动接收。
 4. TXE0 和 RXE0 与由 BRGC0 设置的基本时钟 (f_{XCLK0}) 同步。要再次允许发送或接收，在 TXE0 或 RXE0 已经被清除为 0 后至少两个基本时钟周期后，再设置 TXE0 或 RXE0 为 1。如果在基本时钟的两个周期内设置 TXE0 或 RXE0，发送电路或接收电路可能没有被初始化。
 5. 在设置 TXE0 = 1 后至少一个基本时钟 (f_{XCLK0}) 周期后将发送数据设置到 TXS0。
 6. 在重新写入 PS01、PS00 和 CL0 位前，清除 TXE0 和 RXE0 位为 0。
 7. 当重新写入 SL0 位时，确保 TXE0 = 0。接收总是按照“停止位个数 = 1”来执行，因此不受 SL0 位的设置值影响。
 8. 确保设置第 0 位为 1。

(2) 异步串行接口接收错误状态寄存器 0 (ASIS0)

该寄存器表明串行接口 UART0 接收完成时的错误状态。它包含三个错误标志位 (PE0, FE0, OVE0)。

该寄存器可以由 8 位存储器操作指令只读。

复位信号的产生或清除 ASIS0 的第 7 位 (POWER0) 或第 5 位 (RXE0) 将清除该寄存器为 00H。当读取该寄存器时, 读出的值为 00H。如果接收错误发生, 读取 ASIS0, 然后读取接收缓冲寄存器 0 (RXB0) 清除错误标志。

图 12-3. 异步串行接口接收错误状态寄存器 0 (ASIS0) 的格式

地址: FF73H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	表示校验错误的状态标志
0	如果 POWER0 = 0 或者 RXE0 = 0, 或者然后 ASIS0 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

FE0	表示帧错误的状态标志
0	如果 POWER0 = 0 或者 RXE0 = 0, 或者如果读取 ASIS0 寄存器
1	如果接收完成时未检测到停止位

OVE0	表示溢出错误的状态标志
0	如果 POWER0 = 0 或者 RXE0 = 0, 或者如果读取 ASIS0 寄存器
1	如果接收数据被设置到 RXB0 寄存器并且下一个接收操作在数据被读取前完成

- 注意事项**
1. PE0 位的操作依赖于异步串行接口操作模式寄存器 0 (ASIM0) 的 PS01 和 PS00 位的设置值而不同。
 2. 对于接收数据的停止位, 不管停止位的个数是多少, 只检测第一个停止位。
 3. 如果溢出错误发生, 下一个接收的数据不会被写入接收缓冲寄存器 0 (RXB0), 而是被忽略。
 4. 如果从 ASIS0 读取数据, 将产生一个等待周期。当停止外围硬件时钟 (fPRS) 时, 不要从 ASIS0 读取数据。关于详细信息, 参见第二十七章 等待注意事项。

(3) 波特率发生器控制寄存器 0 (BRGC0)

该寄存器选择串行接口 UART0 的基本时钟和 5 位计数器的分频值。

BRGC0 可以由 8 位存储器操作指令来设置。

复位信号的产生设置该寄存器为 1FH。

图 12-4. 波特率发生器控制寄存器 0 (BRGC0) 的格式

地址: FF71H 复位后: 1FH R / W

符号	7	6	5	4	3	2	1	0
BRGC0	TPS01	TPS00	0	MDL04	MDL03	MDL02	MDL01	MDL00

TPS01	TPS00	基本时钟 (f _{XCLK0}) 选择 ^{注1}				
		f _{PRS} = 4 MHz	f _{PRS} = 8 MHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz	
0	0	TM50 输出 ^{注2}				
0	1	f _{PRS} /2	2 MHz	4 MHz	5 MHz	10 MHz
1	0	f _{PRS} /23	500 kHz	1 MHz	1.25 MHz	2.5 MHz
1	1	f _{PRS} /25	125 kHz	250 kHz	312.5 kHz	625 kHz

MDL04	MDL03	MDL02	MDL01	MDL00	k	5 位计数器输出时钟的选择
0	0	×	×	×	×	禁止设置
0	1	0	0	0	8	f _{XCLK0} /8
0	1	0	0	1	9	f _{XCLK0} /9
0	1	0	1	0	10	f _{XCLK0} /10
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	0	1	0	26	f _{XCLK0} /26
1	1	0	1	1	27	f _{XCLK0} /27
1	1	1	0	0	28	f _{XCLK0} /28
1	1	1	0	1	29	f _{XCLK0} /29
1	1	1	1	0	30	f _{XCLK0} /30
1	1	1	1	1	31	f _{XCLK0} /31

注 1. 如果外围硬件时钟 (f_{PRS}) 工作于高速系统时钟 (f_{XH}) (XSEL = 1), f_{PRS} 工作频率根据供电电压改变。

- V_{DD} = 2.7 至 5.5 V: f_{PRS} ≤ 20 MHz
- V_{DD} = 1.8 至 2.7 V: f_{PRS} ≤ 5 MHz

- 注** 2. 当选择 TM50 输出作为基本时钟时，注意以下要点。
- TM50 和 CR50 匹配时计数时钟被清除并开始的模式 (TMC506 = 0)
开始 8 位定时器 / 事件计数器 50 的操作，然后允许定时器 F/F 反转操作 (TMC501 = 1)。
 - PWM 模式 (TMC506 = 1)
开始 8 位定时器 / 事件计数器 50 的操作，然后设置计数时钟使占空比 = 50%。
在任何模式下都不必允许 (TOE50 = 1) TO50 输出。

- 注意事项**
1. 当重新写入 MDL04 至 MDL00 位时，确保 ASIM0 寄存器的第 6 位 (TXE0) 和第 5 位 (RXE0) = 0。
 2. 当重新写入 TPS01 至 TPS00 位时，确保 ASIM0 寄存器的第 7 位 (POWER0) = 0。
 3. 波特率值是 5 位计数器的输出时钟的 2 分频。

- 备注**
1. f_{CLK0}: 由 TPS01 和 TPS00 位选择的基本时钟的频率
 2. f_{PRS}: 外围硬件时钟频率
 3. k: 由 MDL04 至 MDL00 位设置的值 (k = 8, 9, 10, ..., 31)
 4. x: 无关
 5. 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

(4) 端口模式寄存器 1 (PM1)

该寄存器以 1 位为单位设置端口 1 输入 / 输出。

当使用 P10 / TxD0 / SCK10 引脚作为串行接口数据输出时，清除 PM10 为 0 并设置输出锁存器 P10 为 1。

当使用 P11 / RxD0 / SI10 引脚作为串行接口数据输入时，设置 PM11 为 1。P11 的输出锁存器这时可能是 0 或 1。

PM1 可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生设置该寄存器为 FFH。

图 12-5. 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10
PM1n	P1n 引脚 输入 / 输出模式选择 (n = 0 至 7)							
0	输出模式 (输出缓冲器开)							
1	输入模式 (输出缓冲器关)							

12.4 串行接口UART0 的操作

串行接口 UART0 有以下两种模式。

- 操作停止模式
- 异步串行接口 (UART) 模式

12.4.1 操作停止模式

在这种模式下，不能执行串行通信，因此可以减少功耗。此外，在这种模式下引脚可以用作普通端口引脚。要设置操作停止模式，清除 ASIM0 的第 7、6 和 5 位 (POWER0、TXE0 和 RXE0) 为 0。

(1) 使用的寄存器

操作停止模式由异步串行接口操作模式寄存器 0 (ASIM0) 来设置。

ASIM0 可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生设置该寄存器为 01H。

地址: FF70H 复位后: 01H R / W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1
POWER0	允许 / 禁止内部操作时钟的操作							
0 ^{#1}	禁止内部操作时钟的操作 (固定时钟为低电平) 并且异步复位内部电路 ^{#2} 。							
TXE0	允许 / 禁止发送							
0	禁止发送 (同步复位发送电路)。							
RXE0	允许 / 禁止接收							
0	禁止接收 (同步复位接收电路)。							

- 注
1. 当 POWER0 = 0 时，RxD0 引脚的输入恒为高电平。
 2. 异步串行接口接收错误状态寄存器 0 (ASIS0)、发送移位寄存器 0 (TXS0) 和接收缓存寄存器 0 (RXB0) 被复位。

注意事项 在清除 TXE0 和 RXE0 为 0 后清除 POWER0 为 0 以设置操作停止模式。
要开始通信，设置 POWER0 为 1 然后设置 TXE0 或 RXE0 为 1。

备注 要作为通用端口引脚使用 RxD0 / SI10 / P11 和 TxD0 / SCK10 / P10 引脚，参见 第五章 端口功能。

12.4.2 异步串行接口 (UART) 模式

在这种模式下，在起始位后发送 / 接收 1 字节数据，并且可以执行全双工操作。它包含一个专用的波特率发生器，因此通信可以在很宽的波特率范围内执行。

(1) 使用的寄存器

- 异步串行接口操作模式寄存器 0 (ASIM0)
- 异步串行接口接收错误状态寄存器 0 (ASIS0)
- 波特率发生器控制寄存器 0 (BRGC0)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

在 UART 模式下设置操作的基本过程如下所示。

- <1> 设置 BRGC0 寄存器 (见 图 12-4)。
- <2> 设置 ASIM0 寄存器的第 1 至 4 位 (SL0, CL0, PS00 和 PS01) (见 图 12-2)。
- <3> 设置 ASIM0 寄存器的第 7 位 (POWER0) 为 1。
- <4> 设置 ASIM0 寄存器的第 6 位 (TXE0) 为 1。→ 发送允许。
设置 ASIM0 寄存器的第 5 位 (RXE0) 为 1。→ 接收允许。
- <5> 写入数据到 TXS0 寄存器。→ 数据发送开始。

注意事项 当设置端口模式寄存器和端口寄存器时，注意与通信另一端的关系。

寄存器设置和引脚之间的关系如下所示。

表 12-2. 寄存器设置和引脚之间的关系

POWER0	TXE0	RXE0	PM10	P10	PM11	P11	UART0 操作	引脚功能	
								TxD0 / SCK10 / P10	RxD0 / SI10 / P11
0	0	0	x ^注	x ^注	x ^注	x ^注	停止	SCK10 / P10	SI10 / P11
1	0	1	x ^注	x ^注	1	x	接收	SCK10 / P10	RxD0
	1	0	0	1	x ^注	x ^注	发送	TxD0	SI10 / P11
	1	1	0	1	1	x	发送 / 接收	TxD0	RxD0

注 可以被设置为端口功能或串行接口 CSI10。

备注

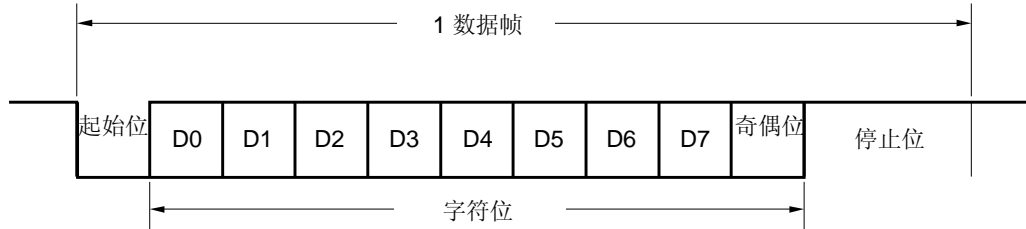
- x: 无关
- POWER0: 异步串行接口操作模式寄存器 0 (ASIM0) 的第 7 位
- TXE0: ASIM0 的第 6 位
- RXE0: ASIM0 的第 5 位
- PM1x: 端口模式寄存器
- P1x: 端口输出锁存器

(2) 通信操作

(a) 正常发送 / 接收数据的格式和波形举例

图 12-6 和 12-7 表示正常发送 / 接收数据的格式和波形举例。

图 12-6. 正常 UART 发送 / 接收数据的格式



一个数据帧由以下位组成。

- 起始位 ... 1 位
- 符号位 ... 7 或 8 位 (LSB 在前)
- 校验位 ... 偶校验, 奇校验, 0 校验或无校验
- 停止位 ... 1 或 2 位

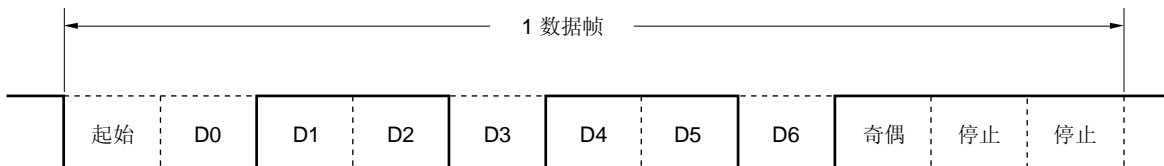
一个数据帧中的符号位长度、校验和停止位长度由异步串行接口操作模式寄存器 0 (ASIM0) 指定。

图 12-7. 正常 UART 发送 / 接收数据波形举例

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



2. 数据长度: 7 位, 校验: 奇校验, 停止位: 2 位, 通信数据: 36H



3. 数据长度: 8 位, 校验: 无, 停止位: 1 位, 通信数据: 87H



(b) 校验类型和操作

校验位用于检测通信数据中的位错误。通常，在发送和接收端使用同样类型的校验位。使用偶校验和奇校验，可以检测一个 1 位（奇数个数）错误。使用零校验和无校验，不能检测到错误。

(i) 偶校验

• 发送

包含校验位的发送数据被控制以使为“1”的位个数是偶数。

校验位的值如下所示。

如果发送数据含有奇数个“1”位： 1

如果发送数据含有偶数个“1”位： 0

• 接收

包含校验位的接收数据中为“1”的位的个数被计数。如果它为奇数，则发生一个校验错误。

(ii) 奇校验

• 发送

与偶校验不同，包含校验位的发送数据被控制以使为“1”的位个数是奇数。

如果发送数据含有奇数个“1”位： 0

如果发送数据含有偶数个“1”位： 1

• 接收

包含校验位的接收数据中为“1”的位的个数被计数。如果它为偶数，则发生一个校验错误。

(iii) 0 校验

当发送数据时，校验位被清除为 0，无论发送的数据是什么。

当接收数据时，不检测校验位。因此，无论校验位是“0”还是“1”，校验错误都不会发生。

(iv) 无校验

发送数据中无校验位。

当接收数据时，按照无校验位执行接收操作。因为没有校验位，不会发生校验错误。

(c) 发送

如果异步串行接口操作模式寄存器 0 (ASIM0) 的第 7 位 (POWER0) 被设置为 1, 并且 ASIM0 的第 6 位 (TXE0) 被设置为 1, 发送允许。通过写入发送数据到发送移位寄存器 0 (TXS0), 可以启动发送。起始位、校验位和停止位自动附加到数据上。

当启动发送时, 从 TxD0 引脚输出起始位, 并且按照 LSB 在前的顺序接着输出剩余数据。当发送完成时, ASIM0 设置的校验和停止位被附加到数据上, 并且产生发送完成中断请求 (INTST0)。

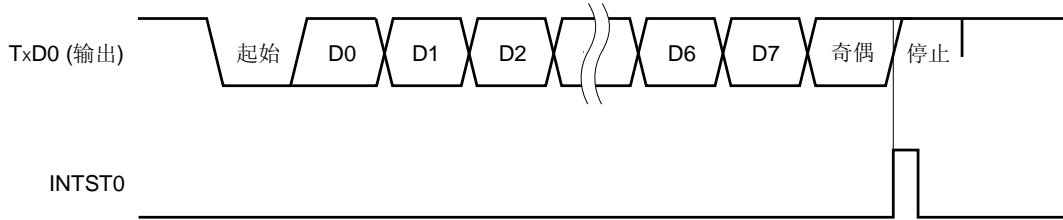
在下一个将要发送的数据被写入到 TXS0 之前, 停止发送。

图 12-8 表示发送完成中断请求 (INTST0) 的时序。只要输出最后的停止位, 就发生这个中断。

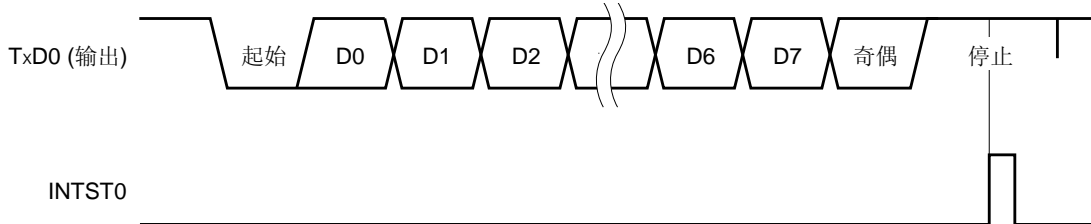
注意事项 在发送数据被写入 TXS0 后, 在产生发送完成中断信号 (INTST0) 之前, 不要写入下一个发送数据。

图 12-8. 发送完成中断请求时序

1. 停止位长度: 1



2. 停止位长度: 2



(d) 接收

当异步串行接口操作模式寄存器 0 (ASIM0) 的第 7 位 (POWER0) 被设置为 1, 然后 ASIM0 的第 5 位 (RXE0) 被设置为 1 时, 允许接收并且采样 RxD0 引脚的输入。

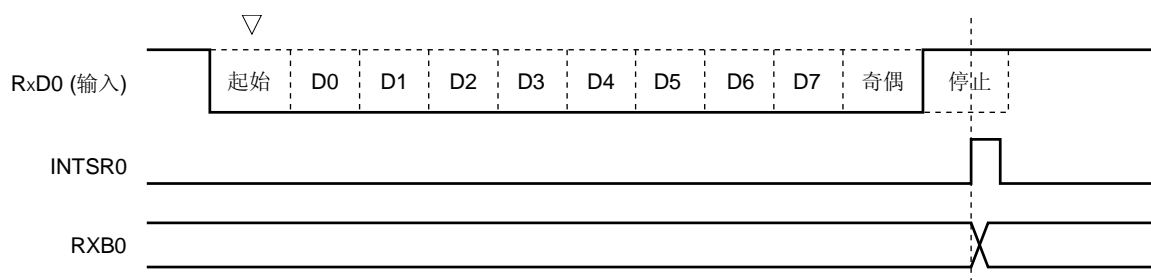
当检测到 RxD0 引脚输入的下降沿时, 波特率发生器的 5 位计数器开始计数。当计数值到达波特率发生器控制寄存器 0 (BRGC0) 的设置值时, 再次采样 RxD0 引脚的输入 (图 12-9 中的 ▽)。如果 RxD0 引脚这时是低电平, 它被识别为起始位。

当检测到起始位时, 接收开始, 并且串行数据按照设置的波特率存储到接收移位寄存器 0 (RXS0) 中。当接收到停止位时, 将产生接收完成中断 (INTSR0), 并且 RXS0 中的数据被写入到接收缓冲寄存器 0 (RXB0)。然而, 如果发生一个溢出错误 (OVE0), 接收的数据不会被写入到 RXB0。

当正在接收操作时, 即使校验错误 (PE0) 发生, 也将继续接收操作直到接收到停止位, 并且在接收完成时产生一个接收错误中断 (INTSR0)。

INTSR0 在接收完成并且接收错误的情况下发生。

图 12-9. 接收完成中断请求时序



注意事项 1. 如果接收错误发生, 读取异步串行接口接收错误状态寄存器 0 (ASIS0), 然后读取接收缓冲寄存器 0 (RXB0) 清除错误标志。

否则, 当接收下一个数据时, 将发生一个溢出错误, 并且保持接收错误状态。

2. 接收总是按照“停止位个数 = 1”来执行。忽略第二个停止位。

(e) 接收错误

接收期间可能发生三种错误：校验错误、帧错误或溢出错误。如果由于接收数据设置了异步串行接口接收错误状态寄存器 0 (ASIS0) 的错误标志，将产生一个接收错误中断 (INTSR0)。

可以通过在接收错误中断 (INTSR0) 服务程序中读取 ASIS0 的内容来鉴别接收期间哪一个错误发生 (见 图 12-3)。

当读取 ASIS0 时，ASIS0 的内容被清除为 0。

表 12-3. 接收错误的原因

接收错误	原因
校验错误	发送指定的校验与接收数据的校验不匹配。
帧错误	未检测到停止位。
溢出错误	下一个数据的接收在数据从接收缓冲寄存器 0 (RXB0) 读取前完成。

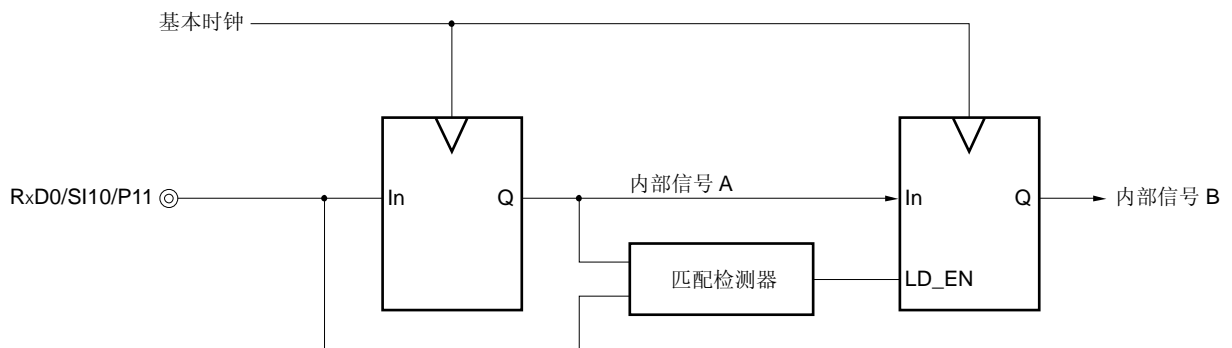
(f) 接收数据的噪声滤波器

由预分频模块输出的基本时钟采样 RXD0 信号。

如果两个采样值相同，匹配检测的输出更改，并且作为输入采样数据。

因为电路的配置如图 12-10 所示，接收操作的内部处理从外部信号状态被延迟两个时钟周期。

图 12-10. 噪声滤波器电路



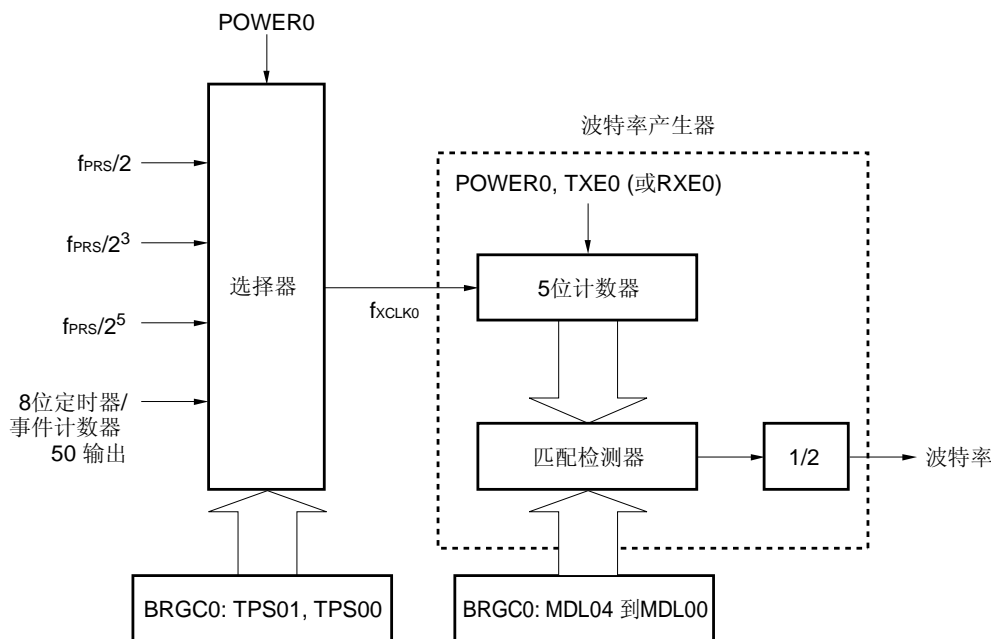
12.4.3 专用波特率发生器

专用波特率发生器由一个源时钟选择器和一个 5 位可编程计数器组成，并且为 UART0 的发送 / 接收产生串行时钟。单独的 5 位计数器提供给发送和接收。

(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 0 (ASIM0) 的第 7 位 (POWER0) 被设置为 1 时，由波特率发生器控制寄存器 0 (BRGC0) 的第 7 和 6 位 (TPS01 和 TPS00) 选择的时钟提供给每个模块。这个时钟叫做基本时钟，其频率称为 f_{CLK0} 。当 $POWER0 = 0$ 时，基本时钟固定为低电平。
- 发送计数器
当串行接口操作模式寄存器 0 (ASIM0) 的第 7 位 (POWER0) 或第 0 位 (TXE0) 为 0 时，这个计数器停止操作，清除为 0。
当 $POWER0 = 1$ 并且 $TXE0 = 1$ 时，它开始计数。
当发送的第一个数据被写入到发送移位寄存器 0 (TXS0) 时，计数器被清除为 0。
- 接收计数器
当串行接口操作模式寄存器 0 (ASIM0) 的第 7 位 (POWER0) 或第 5 位 (RXE0) 为 0 时，这个计数器停止操作，清除为 0。
当检测到起始位时，开始计数。
在接收到一帧数据后，检测到下一个起始位前，计数器停止操作。

图 12-11. 波特率发生器的配置



备注 POWER0: 异步串行接口操作模式寄存器 0 (ASIM0) 的第 7 位
 TXE0: ASIM0 的第 6 位
 RXE0: ASIM0 的第 5 位
 BRGC0: 波特率发生器控制寄存器 0

(2) 串行时钟的产生

通过使用波特率发生器控制寄存器 0 (BRGC0) 指定要产生的串行时钟。

通过使用 BRGC0 的第 7 和 6 位 (TPS01 和 TPS00) 选择输入到 5 位计数器的时钟。

BRGC0 的第 4 至 0 位 (MDL04 至 MDL00) 可以用于选择 5 位计数器的分频值 (fxCLK0/8 至 fxCLK0/31)。

12.4.4 波特率的计算

(1) 波特率计算表达式

波特率可以由以下表达式计算。

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

f_{XCLK0}: 由 BRGC0 寄存器的 TPS01 和 TPS00 位选择的基本时钟的频率

k: 由 BRGC0 寄存器的 MDL04 至 MDL00 位设置的值 (k = 8, 9, 10, ..., 31)

表 12-4. TPS01 和 TPS00 的设置值

TPS01	TPS00	基本时钟 (fxCLK0) 选择 ^{#1}				
		f _{PRS} = 4 MHz	f _{PRS} = 8 MHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz	
0	0	TM50 输出 ^{#2}				
0	1	f _{PRS} /2	5 MHz	4 MHz	5 MHz	10 MHz
1	0	f _{PRS} /23	500 kHz	1 MHz	1.25 MHz	2.5 MHz
1	1	f _{PRS} /25	125 kHz	250 kHz	312.5 kHz	625 kHz

- 注
- 如果外围硬件时钟 (f_{PRS}) 工作于高速系统时钟 (f_{XH}) (XSEL = 1), f_{PRS} 工作频率根据供电电压改变。
 - V_{DD} = 2.7 至 5.5 V: f_{PRS} ≤ 20 MHz
 - V_{DD} = 1.8 至 2.7 V: f_{PRS} ≤ 5 MHz
 - 当选择 TM50 输出作为基本时钟时, 注意以下要点。
 - TM50 和 CR50 匹配时清零&启动模式 (TMC506 = 0)
开始 8 位定时器 / 事件计数器 50 的操作, 然后允许定时器 F/F 反转操作 (TMC501 = 1)。
 - PWM 模式 (TMC506 = 1)
开始 8 位定时器 / 事件计数器 50 的操作, 然后设置计数时钟使占空比 = 50%。
在任何模式下都不必允许 (TOE50 = 1) TO50 输出。

(2) 波特率误差

波特率误差可以由以下表达式计算。

$$\bullet \text{ 误差 (\%)} = \left[\frac{\text{实际波特率 (带有误差的波特率)}}{\text{期望的波特率 (正确的波特率)}} - 1 \right] \times 100 [\%]$$

- 注意事项
- 在发送期间, 确保波特率误差处于接收端的允许误差范围之内。
 - 确保接收期间的波特率误差满足 (4) 接收期间允许的波特率范围中显示的范围。

举例： 基本时钟频率 = 2.5 MHz = 2,500,000 Hz
 BRGC0 寄存器的 MDL04 至 MDL00 位的设置值 = 10000B (k = 16)
 目标波特率 = 76,800 bps

$$\begin{aligned} \text{波特率} &= 2.5 \text{ M} / (2 \times 16) \\ &= 2,500,000 / (2 \times 16) = 78,125 \text{ [bps]} \end{aligned}$$

$$\begin{aligned} \text{误差} &= (78,125 / 76,800 - 1) \times 100 \\ &= 1.725 \text{ [%]} \end{aligned}$$

(3) 设置波特率举例

表 12-5. 波特率发生器的设置数据

波特率 [bps]	f _{PRS} = 2.0 MHz				f _{PRS} = 5.0 MHz				f _{PRS} = 10.0 MHz				f _{PRS} = 20.0 MHz			
	TPS01, TPS00	k	计算的 值	ERR [%]	TPS01, TPS00	k	计算的 值	ERR [%]	TPS01, TPS00	k	计算的 值	ERR [%]	TPS01, TPS00	k	计算的 值	ERR [%]
4800	2H	26	4808	0.16	3H	16	4883	1.73	-	-	-	-	-	-	-	-
9600	2H	13	9615	0.16	3H	8	9766	1.73	3H	16	9766	1.73	-	-	-	-
10400	2H	12	10417	0.16	2H	30	10417	0.16	3H	15	10417	0.16	3H	30	10417	0.16
19200	1H	26	19231	0.16	2H	16	19531	1.73	3H	8	19531	1.73	3H	16	19531	1.73
24000	1H	21	23810	-0.79	2H	13	24038	0.16	2H	26	24038	0.16	3H	13	24038	0.16
31250	1H	16	31250	0	2H	10	31250	0	2H	20	31250	0	3H	10	31250	0
33600	1H	15	33333	-0.79	2H	9	34722	3.34	2H	19	32895	-2.1	3H	9	34722	3.34
38400	1H	13	38462	0.16	2H	8	39063	1.73	2H	16	39063	1.73	3H	8	39063	1.73
56000	1H	9	55556	-0.79	1H	22	56818	1.46	2H	11	56818	1.46	2H	22	56818	1.46
62500	1H	8	62500	0	1H	20	62500	0	2H	10	62500	0	2H	20	62500	0
76800	-	-	-	-	1H	16	78125	1.73	2H	8	78125	1.73	2H	16	78125	1.73
115200	-	-	-	-	1H	11	113636	-1.36	1H	22	113636	-1.36	2H	11	113636	-1.36
153600	-	-	-	-	1H	8	156250	1.73	1H	16	156250	1.73	2H	8	156250	1.73
312500	-	-	-	-	-	-	-	-	1H	8	312500	0	1H	16	312500	0
625000	-	-	-	-	-	-	-	-	-	-	-	-	1H	8	625000	0

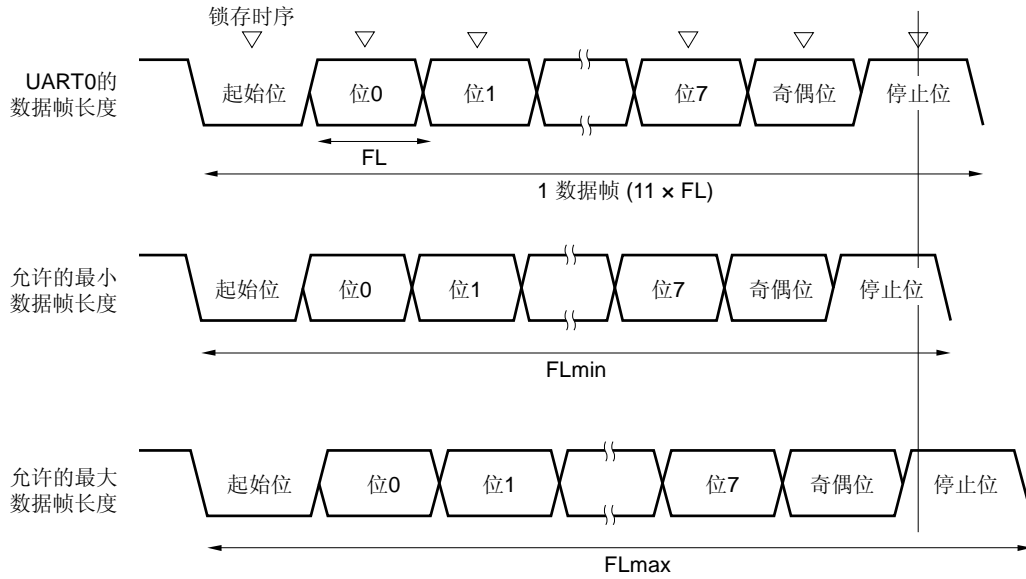
备注 TPS01, TPS00: 波特率发生器控制寄存器 0 (BRGC0) 的第 7 和 6 位 (基本时钟 (f_{CLK0}) 的设置)
 k: 由 BRGC0 的 MDL04 至 MDL00 位设置的值 (k = 8, 9, 10, ..., 31)
 f_{PRS}: 外围硬件时钟频率
 ERR: 波特率误差

(4) 接收期间允许的波特率范围

接收期间与发送端的允许误差如下所示。

注意事项 通过使用下面的计算表达式，确保接收期间的波特率误差在允许的误差范围内。

图 12-12. 接收期间允许的波特率范围



如图 12-12 所示，在起始位被检测后，接收数据的锁存时序由通过波特率发生器控制寄存器 0 (BRGC0) 设置的计数器决定。如果最后一个数据（停止位）满足这个锁存时序，可以正确接收数据。假设 11 位数据被接收，理论值可以按照下面计算。

$$FL = (\text{Brate})^{-1}$$

- Brate: UART0 的波特率
- k: BRGC0 的设置值
- FL: 1 位数据长度
- 锁存时序的余量: 2 时钟

$$\text{允许的最小数据帧长度: } FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

因此，发送端的最大可接收波特率如下所示。

$$BR_{\max} = (FL_{\min}/11)^{-1} = \frac{22k}{21k+2} \text{ Brate}$$

类似的，允许的最大数据帧长度可以按照下面计算。

$$\frac{10}{11} \times FL_{\max} = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FL_{\max} = \frac{21k-2}{20k} FL \times 11$$

因此，发送端的最小可接收波特率如下所示。

$$BR_{\min} = (FL_{\max}/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

UART0 和发送端之间允许的波特率误差可以从上面的最小和最大波特率表达式计算出来，如下所示。

表 12-6. 允许的最大 / 最小波特率误差

分频比率 (k)	允许的最大波特率误差	允许的最小波特率误差
8	+3.53%	-3.61%
16	+4.14%	-4.19%
24	+4.34%	-4.38%
31	+4.44%	-4.47%

- 备注**
1. 接收的允许误差与每帧的位数、输入时钟频率和分频比率 (k)。输入时钟频率越高并且分频 (k) 比率越大，允许的误差越大。
 2. k: BRGC0 的设置值

13.1 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

这个模式在未执行串行通信时使用，并且可以减少功耗。

关于详细信息，参见 13.4.1 操作停止模式。

(2) 异步串行接口 (UART) 模式

该模式支持 LIN (本地互连网络) 总线。该模式的功能概述如下。

关于详细信息，参见 13.4.2 异步串行接口 (UART) 模式和 13.4.3 专用波特率发生器。

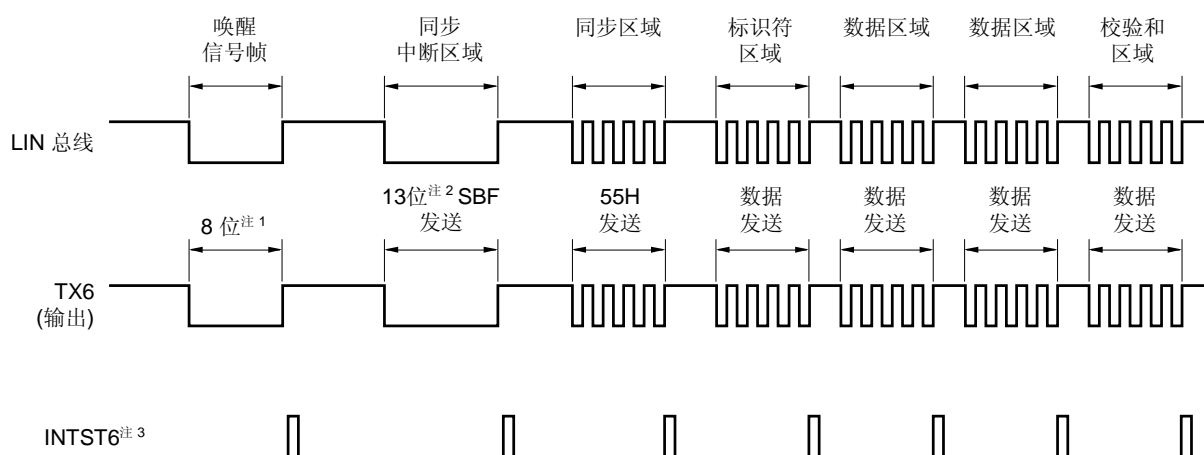
- 最大传输速率：625 kbps
- 两个引脚配置 TxD6：发送数据输出引脚
RxD6：接收数据输入引脚
- 通信数据的数据长度可选择 7 位或 8 位。
- 专用的内部 8 位波特率发生器允许设置任意大小波特率。
- 可以独立执行发送和接收 (全双工操作)。
- MSB 或 LSB 在前通信可以选择
- 反转发送操作
- 同步中断区域发送长度为 13~20 位
- 可规定多于 11 位的同步中断区域接收 (提供 SBF 接收标志)

- 注意事项**
1. TxD6 输出反转功能只反转发送端，而不反转接收端。要使用该功能，接收端必须准备好反转数据的接收。
 2. 如果未停止串行接口 UART6 的时钟 (例如，在 HALT 模式下)，则继续正常操作。如果提供给串行接口 UART6 的时钟被停止 (例如，在 STOP 模式下)，则寄存器停止工作，并且保持时钟刚刚停止前的值。TxD6 引脚也保持时钟刚刚停止前的值并输出。然而，时钟重新开始后，不保证操作的可靠性。因此，使 POWER6 = 0、RXE6 = 0 和 TXE6 = 0，以复位电路。
 3. 设置 POWER6 = 1，然后设置 TXE6 = 1 (发送) 或 RXE6 = 1 (接收)，启动通信。
 4. TXE6 和 RXE6 与由 CKSR6 设置的基本时钟 (fxCLK6) 同步。要再次允许发送或接收，在 TXE6 或 RXE6 已经被清除为 0 后的至少两个基本时钟周期后，再设置 TXE6 或 RXE6 为 1。如果在基本时钟的两个周期内设置 TXE6 或 RXE6，发送电路或接收电路可能没有被初始化。
 5. 在设置 TXE6 = 1 后至少一个基本时钟 (fxCLK6) 周期后，再将待发送的数据设置到 TXB6。
 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模块工作时钟。然而，这不影响通信的结果，因为接收端在检测到起始位时会初始化时序。如果接口用于 LIN 通信操作，不要使用连续发送功能。

备注 LIN 代表本地互连网络，它是一个有助于降低汽车网络成本的低速（1 至 20kbps）串行通信协议。LIN 通信是单主设备通信，最多 15 个从设备可以连接到一个主设备。LIN 从设备用于控制开关、制动器和传感器，并且它们通过 LIN 网络连接到 LIN 主设备。一般，LIN 主设备被连接到一个网络，例如 CAN（控制器局域网）。此外，LIN 使用单线方式并且通过符合 ISO9141 的收发器连接到节点。在 LIN 协议中，主设备发送一个带有波特率信息的帧，从设备接收帧并修正波特率误差。因此，当从设备的波特率误差为 $\pm 15\%$ 或更小时，可以进行通信。

图 13-1 和 13-2 概述了 LIN 的发送和接收操作。

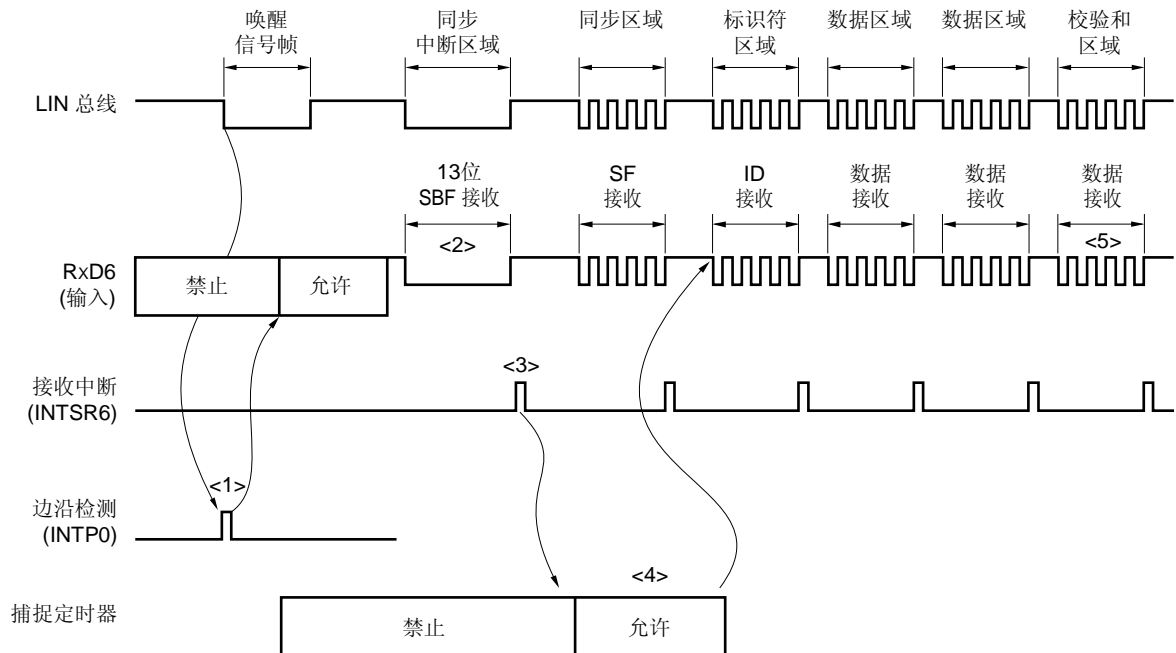
图 13-1. LIN 发送操作



- 注**
1. 在 8 位模式下，可以通过发送 80H 代替唤醒信号帧。
 2. 可由硬件输出同步中断区域。输出宽度是由异步串行接口控制寄存器 6 (ASICL6) 的第 4 至 2 位 (SBL62 至 SBL60) 设置 (参见 13.4.2 (2) (h) SBF 发送)。
 3. 在每个发送完成时输出 INTST6 信号。当发送 SBF 时，也输出 INTST6 信号。

备注 每个区域之间的间隔由软件控制。

图 13-2. LIN 接收操作



接收处理如下所示。

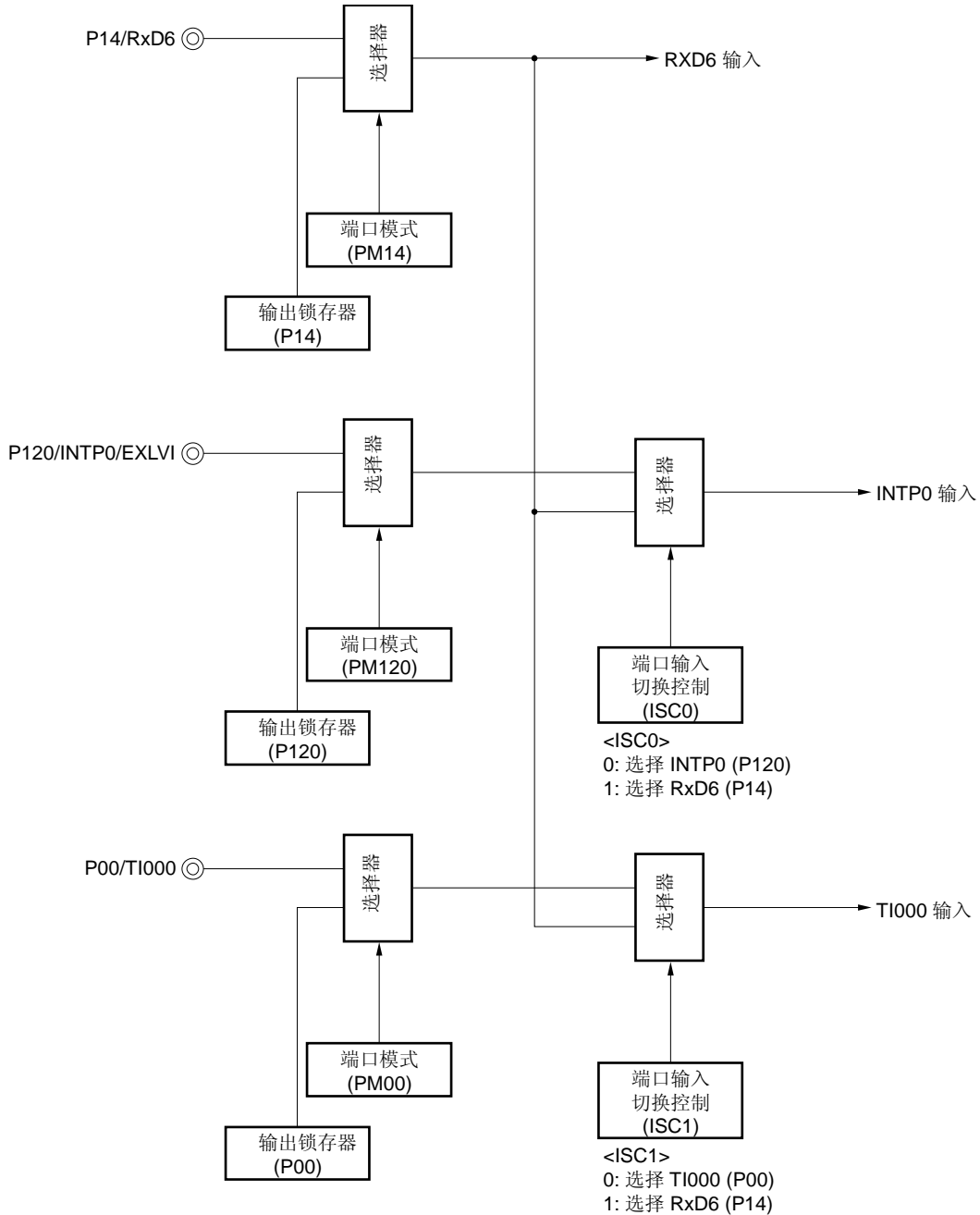
- <1> 在引脚的边沿检测唤醒信号，允许 UART6，并设置 SBF 接收模式。
- <2> 继续接收直到检测到 STOP 位。当检测到 11 位或更多的低电平数据的 SBF 时，认为 SBF 接收已经正确完成，并且输出一个中断信号。如果检测到少于 11 位的低电平数据的 SBF 时，认为 SBF 接收错误发生。不输出中断信号，并且恢复 SBF 接收模式。
- <3> 如果 SBF 接收已经正确完成，输出中断信号。通过 SBF 接收结束中断服务程序启动 16 位定时器 / 事件计数器 00，并测量同步区域的位间隔（脉冲宽度）（参见 7.4.8 脉冲宽度测量操作）。禁止检测 OVE6、PE6 和 FE6 错误，并且不执行 UART 通信的错误检测处理以及移位寄存器和 RXB6 之间的数据传送。移位寄存器保持复位时的值 FFH。
- <4> 从同步区域的位间隔计算波特率误差，SF 接收后，禁止 UART6 并且重新设置波特率发生器控制寄存器 6 (BRGC6)。
- <5> 通过软件辨别校验和区域。同时在校验和区域接收后，通过软件执行处理来初始化 UART6 并再次设置 SBF 接收模式。

图 13-3 表示 LIN 接收操作的端口配置。

通过检测外部中断 (INTPO) 的边沿，接收从 LIN 主设备发送的唤醒信号。从 LIN 主设备发送的同步区域的长度可以使用 16 位定时器 / 事件计数器 00 的外部事件捕捉操作来测量，并计算波特率误差。

接收端口输入 (RxD6) 的输入源可以通过端口输入切换控制 (ISC0 / ISC1) 被输入到外部中断 (INTPO) 和 16 位定时器 / 事件计数器 00，而不用外部连接 RxD6 和 INTPO / TI000。

图 13-3. LIN 接收操作的端口配置



备注 ISC0, ISC1: 输入切换控制寄存器 (ISC) 的第 0 和 1 位 (见 图 13-11)

用于 LIN 通信操作的外围功能如下所示。

<使用的外围功能>

- 外部中断 (INTP0) ; 唤醒信号检测
使用: 检测唤醒信号边沿并检测通信的开始。
- 16 位定时器 / 事件计数器 00 (TI000) ; 波特率误差检测
使用: 通过检测同步区域 (SF) 长度来检测波特率误差 (在捕捉模式下测量 TI000 输入边沿间隔) 并按照位的个数分隔。
- 串行接口 UART6

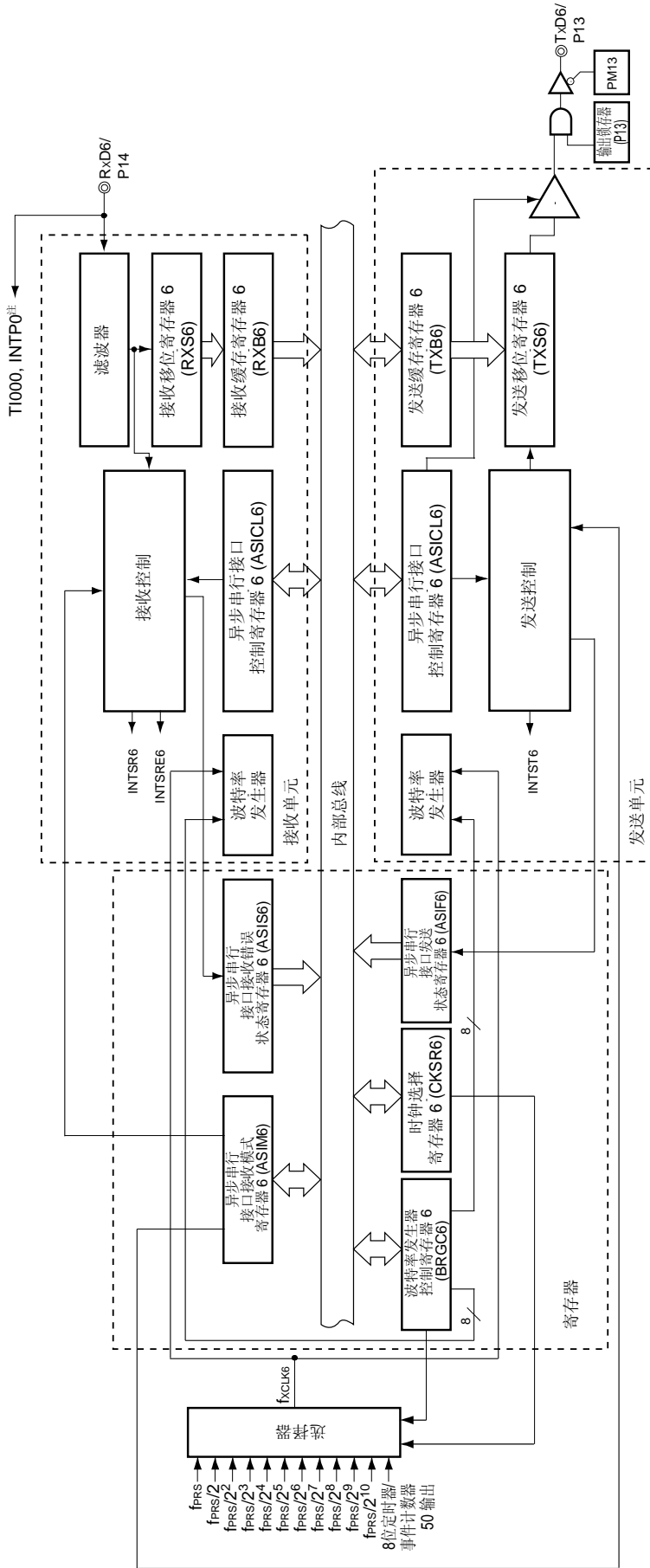
13.2 串行接口UART6 的配置

串行接口 UART6 包含以下硬件。

表 13-1. 串行接口 UART6 的配置

项目	配置
寄存器	接收缓存寄存器 6 (RXB6) 接收移位寄存器 6 (RXS6) 发送缓存寄存器 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) 输入切换控制寄存器 (ISC) 端口模式寄存器 1 (PM1) 端口寄存器 1 (P1)

图 13-4. 串行接口 UART6 的框图



注 通过输入切换控制寄存器 (ISC) 可选。

(1) 接收缓存寄存器 6 (RXB6)

该 8 位寄存器存储由接收移位寄存器 6 (RXS6) 转换的并行数据。

每接收 1 字节的数据，新接收的数据从 RXS6 传送到该寄存器。如果设置数据长度为 7 位，数据传送如下所示。

- 在 LSB 在前的接收中，接收数据被传送到 RXB6 的第 0 至 6 位，RXB6 的 MSB 总是 0。
- 在 MSB 在前的接收中，接收数据被传送到 RXB6 的第 1 至 7 位，RXB6 的 LSB 总是 0。

如果溢出错误 (OVE6) 发生，接收数据不会被传送到 RXB6。

RXB6 可以由 8 位存储器操作指令读取。不能向该寄存器写入数据。

复位信号的产生设置该寄存器为 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为并行数据。

RXS6 不能被程序直接操作。

(3) 发送缓存寄存器 6 (TXB6)

该缓存寄存器用来设置发送数据。当数据被写入到 TXB6 时，启动发送。

该寄存器可以由 8 位存储器操作指令读取或写入。

复位信号的产生设置该寄存器为 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，不要向 TXB6 写入数据。
 2. 在通信操作期间（当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 和 6 位 (POWER6, TXE6) 为 1 或者 ASIM6 的第 7 和 5 位 (POWER6, RXE6) 为 1 时），不要通过软件刷新（向其写入相同数据）TXB6。
 3. 在设置 TXE6 = 1 后至少一个基本时钟 (f_{CLK6}) 周期后，再将待发送的数据设置到 TXB6。

(4) 发送移位寄存器 6 (TXS6)

该寄存器以串行数据从 TxD6 引脚发送由 TXB6 传送来的数据。对于第一次发送，将数据写入到 TXB6 后立即从 TXB6 传送，或者对于连续发送，一帧发送完成后 INTST6 发生前立即传送。从 TXB6 发送数据，并且在基本时钟的下降沿从 TxD6 引脚发送。

TXS6 不能由程序直接操作。

13.3 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)
- 输入切换控制寄存器 (ISC)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

(1) 异步串行接口操作模式寄存器 6 (ASIM6)

该 8 位寄存器控制串行接口 UART6 的串行通信操作。

该寄存器可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生设置该寄存器为 01H。

备注 在通信操作期间 (ASIM6 的第 7 和 6 位 (POWER6, TXE6) = 1 或者 ASIM6 的第 7 和 5 位 (POWER6, RXE6) = 1), ASIM6 可以通过软件被刷新 (向其写入同样的值)。

图 13-5. 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 (固定时钟为低电平) 并且异步复位内部电路 ^{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. 在发送期间, 当 POWER6 = 0 时, TxD6 引脚的输出固定为高电平 (如果 TXDLV6=0)。并且, 从 RxD6 引脚输入的电平固定为高电平。
 2. 异步串行接口接收错误状态寄存器 6 (ASIS6)、异步串行接口发送状态寄存器 6 (ASIF6)、异步串行接口控制寄存器 6 (ASICL6) 的第 7 位 (SBRF6) 和第 6 位 (SBRT6) 以及接收缓存寄存器 6 (RXB6) 被复位。

图 13-5. 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。	无校验位接收
0	1	输出零校验。	按零校验接收 [#]
1	0	输出奇数校验。	按照奇数校验判断。
1	1	输出偶数校验。	按照偶数校验判断。

CL6	指定发送数据的字符长度
0	数据的字符长度 = 7 位
1	数据的字符长度 = 8 位

SL6	指定发送数据的停止位个数
0	停止位个数 = 1
1	停止位个数 = 2

ISRM6	允许 / 禁止错误情况下接收完成中断的发生
0	在错误情况下产生“INTSRE6”（这时，不产生 INTSR6）。
1	在错误情况下产生“INTSR6”（这时，不产生 INTSRE6）。

注 如果选择“按零校验接收”，不判断校验位。因此，不要设置异步串行接口接收错误状态寄存器 6 (ASIS6) 的第 2 位 (PE6)，这样也不会产生出错中断。

- 注意事项**
1. 要开始发送，设置 POWER6 为 1 然后设置 TXE6 为 1。要停止发送，清除 TXE6 为 0，并且然后清除 POWER6 为 0。
 2. 要开始接收，设置 POWER6 为 1 然后设置 RXE6 为 1。要停止接收，清除 RXE6 为 0，并且然后清除 POWER6 为 0。
 3. 当一个高电平输入到 RxD6 引脚时，设置 POWER6 为 1 然后设置 RXE6 为 1。如果输入低电平时，设置 POWER6 为 1，并且设置 RXE6 为 1，启动接收。
 4. TXE6 和 RXE6 与由 CKSR6 设置的基本时钟 (fxCLK6) 同步。要再次允许发送或接收，在 TXE6 或 RXE6 已经被清除为 0 后至少两个基本时钟周期后，再设置 TXE6 或 RXE6 为 1。如果在基本时钟的两个周期内设置 TXE6 或 RXE6，发送电路或接收电路可能没有初始化。
 5. 在设置 TXE6 = 1 后至少一个基本时钟 (fxCLK6) 周期后，再将待发送的数据设置到 TXB6。
 6. 在重新写入 PS61、PS60 和 CL6 位前，清除 TXE6 和 RXE6 位为 0。
 7. 当用于 LIN 通信操作时，PS61 和 PS60 位恒为 0。
 8. 在重新写入 SL6 位前，清除 TXE6 为 0。接收总是按照“停止位个数 = 1”来执行，因此不受 SL6 位的设置值影响。
 9. 当重新写入 ISRM6 位时，确保 RXE6 = 0。

(2) 异步串行接口接收错误状态寄存器 6 (ASIS6)

该寄存器表明串行接口 UART6 接收完成时的错误状态。它包含三个错误标志位 (PE6, FE6, OVE6)。

该寄存器可以由 8 位存储器操作指令只读。

复位信号的产生或清除 ASIS6 的第 7 位 (POWER6) 或第 5 位 (RXE6) 将清除该寄存器为 00H。当该寄存器被读取时, 读出 00H。如果发生接收错误, 读取 ASIS6, 然后读取接收缓存寄存器 6 (RXB6) 将清除错误标志。

图 13-6. 异步串行接口接收错误状态寄存器 6 (ASIS6) 的格式

地址: FF53H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIS6	0	0	0	0	0	PE6	FE6	OVE6

PE6	表示校验错误的状态标志
0	如果 POWER6 = 0 或者 RXE6 = 0, 或者如果读取 ASIS6 寄存器
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	表示帧错误的状态标志
0	如果 POWER6 = 0 或者 RXE6 = 0, 或者如果读取 ASIS6 寄存器
1	如果接收完成时未检测到停止位

OVE6	表示溢出错误的状态标志
0	如果 POWER6 = 0 或者 RXE6 = 0, 或者如果读取 ASIS6 寄存器
1	如果接收的数据被设置到 RXB6 寄存器并且下一个接收操作在数据被读取前完成

- 注意事项
1. PE6 位的操作依赖于异步串行接口操作模式寄存器 6 (ASIM6) 的 PS61 和 PS60 位的设置值而不同。
 2. 对于接收数据的停止位, 不管停止位的个数是多少, 只将接收数据的第一位作为停止位检测。
 3. 如果溢出错误发生, 下一个接收数据不会被写入接收缓存寄存器 6 (RXB6), 而是被忽略。
 4. 如果从 ASIS6 读取数据, 将产生一个等待周期。当停止外围硬件时钟 (fPRS) 时, 不要从 ASIS6 读取数据。关于详细信息, 参见第二十七章 等待注意事项。

(3) 异步串行接口发送状态寄存器 6 (ASIF6)

该寄存器表明串行接口 UART6 的发送的状态。它包含两个状态标志位 (TXBF6 和 TXSF6)。

当数据从 TXB6 寄存器被传送到 TXS6 寄存器后, 通过向 TXB6 寄存器写入下一个数据, 可以继续发送, 即使在中断期间也不受影响。

该寄存器可以由 8 位存储器操作指令执行只读操作。

复位信号产生或清除 ASIM6 的第 7 位 (POWER6) 或第 6 位 (TXE6) 将清除该寄存器为 00H。

图 13-7. 异步串行接口发送状态寄存器 6 (ASIF6) 的格式

地址: FF55H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIF6	0	0	0	0	0	0	TXBF6	TXSF6

TXBF6	发送缓存数据标志
0	如果 POWER6 = 0 或 TXE6 = 0, 或者如果数据被传送到发送移位寄存器 6 (TXS6)
1	如果数据被写入到发送缓存寄存器 6 (TXB6) (如果在 TXB6 中存在数据)

TXSF6	发送移位寄存器数据标志
0	如果 POWER6 = 0 或 TXE6 = 0, 或者如果在完成发送后下一个没有从发送缓存寄存器 6 (TXB6) 传送
1	如果数据从发送缓存寄存器 6 (TXB6) 中传送 (如果数据发送正在进行)

- 注意事项**
1. 要连续发送数据, 写入第一个发送数据 (第一个字节) 到 TXB6 寄存器。确保检查 TXBF6 标志为“0”。如果正确, 将下一个发送数据 (第二个字节) 写入到 TXB6 寄存器。如果当 TXBF6 标志为“1”时数据被写入 TXB6 寄存器, 将不能保证发送数据的正确性。
 2. 要在连续发送完成时初始化发送单元, 必须在发送完成中断产生后必须检查 TXSF6 标志位为“0”, 然后执行初始化。如果当 TXSF6 标志为“1”时执行初始化, 将不能保证发送数据的正确性。

(4) 时钟选择寄存器 6 (CKSR6)

该寄存器选择串行接口 UART6 的基本时钟。CKSR6 可以由 8 位存储器操作指令设置。复位信号的产生清除该寄存器为 00H。

备注 在通信操作期间 (ASIM6 的第 7 和 6 位 (POWER6, TXE6) = 1 或者 ASIM6 的第 7 和 5 位 (POWER6, RXE6) = 1), 可以通过软件刷新 CKSR6 (向其写入同样的值)。

图 13-8. 时钟选择寄存器 6 (CKSR6) 的格式

地址: FF56H 复位后: 00H R / W

符号	7	6	5	4	3	2	1	0
CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60

TPS63	TPS62	TPS61	TPS60	基本时钟 (f _{CLK6}) 选择 ^{注1}				
				f _{PRS} = 4 MHz	f _{PRS} = 8 MHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz	
0	0	0	0	f _{PRS} ^{注2}	4 MHz	8 MHz	10 MHz	20 MHz ^{注3}
0	0	0	1	f _{PRS} /2	2 MHz	4 MHz	5 MHz	10 MHz
0	0	1	0	f _{PRS} /2 ²	1 MHz	2 MHz	2.5 MHz	5 MHz
0	0	1	1	f _{PRS} /2 ³	500 kHz	1 MHz	1.25 MHz	2.5 MHz
0	1	0	0	f _{PRS} /2 ⁴	250 kHz	500 kHz	625 kHz	1.25 MHz
0	1	0	1	f _{PRS} /2 ⁵	125 kHz	250 kHz	312.5 kHz	625 kHz
0	1	1	0	f _{PRS} /2 ⁶	62.5 kHz	125 kHz	156.25 kHz	312.5 kHz
0	1	1	1	f _{PRS} /2 ⁷	31.25 kHz	62.5 kHz	78.13 kHz	156.25 kHz
1	0	0	0	f _{PRS} /2 ⁸	15.625 kHz	31.25 kHz	39.06 kHz	78.13 kHz
1	0	0	1	f _{PRS} /2 ⁹	7.813 kHz	15.625 kHz	19.53 kHz	39.06 kHz
1	0	1	0	f _{PRS} /2 ¹⁰	3.906 kHz	7.813 kHz	9.77 kHz	19.53 kHz
1	0	1	1	TM50 输出 ^{注4}				
除上面以外				禁止设置				

- 注**
- 如果外围硬件时钟 (f_{PRS}) 工作于高速系统时钟 (f_{XH}) (XSEL = 1), f_{PRS} 工作频率根据供电电压改变。
 - V_{DD} = 2.7 至 5.5 V : f_{PRS} ≤ 20 MHz
 - V_{DD} = 1.8 至 2.7 V : f_{PRS} ≤ 5 MHz
 - 如果外围硬件时钟 (f_{PRS}) 工作于内部高速振荡时钟 (f_{RH}) (XSEL = 0), 当 1.8 V ≤ V_{DD} < 2.7 V 时, TPS63 = TPS62 = TPS61 = TPS60 = 0 的设置 (计数时钟: f_{PRS}) 被禁止。
 - 只有当 4.0 V ≤ V_{DD} ≤ 5.5 V 时才可以设置。
 - 当选择 TM50 输出作为基本时钟时, 注意以下要点。
 - TM50 和 CR50 匹配时清零&启动模式 (TMC506 = 0)
开始 8 位定时器 / 事件计数器 50 的操作, 然后允许定时器 F/F 反转操作 (TMC501 = 1) .
 - PWM 模式 (TMC506 = 1)
开始 8 位定时器 / 事件计数器 50 的操作, 然后设置计数时钟使占空比 = 50%。
在任何模式下都不必允许 (TOE50 = 1) TO50 输出。

注意事项 当重新写入 TPS63 至 TPS60 时, 确保 POWER6 = 0。

- 备注**
- f_{PRS}: 外围硬件时钟频率
 - 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

(5) 波特率发生器控制寄存器 6 (BRGC6)

该寄存器设置串行接口 UART6 的 8 位计数器的分频值。

BRGC6 可以由 8 位存储器操作指令来设置。

复位信号的产生设置该寄存器为 FFH。

备注 在通信操作期间 (ASIM6 的第 7 和 6 位 (POWER6, TXE6) = 1 或者 ASIM6 的第 7 和 5 位 (POWER6, RXE6) = 1)，可以通过软件刷新 BRGC6 (向其写入同样的值)。

图 13-9. 波特率发生器控制寄存器 6 (BRGC6) 的格式

地址: FF57H 复位后: FFH R / W

符号	7	6	5	4	3	2	1	0
BRGC6	MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	k	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	$f_{CLK6}/4$
0	0	0	0	0	1	0	1	5	$f_{CLK6}/5$
0	0	0	0	0	1	1	0	6	$f_{CLK6}/6$
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	$f_{CLK6}/252$
1	1	1	1	1	1	0	1	253	$f_{CLK6}/253$
1	1	1	1	1	1	1	0	254	$f_{CLK6}/254$
1	1	1	1	1	1	1	1	255	$f_{CLK6}/255$

- 注意事项**
1. 当重新写入 MDL67 至 MDL60 位时，确保 ASIM6 寄存器的第 6 位 (TXE6) 和第 5 位 (RXE6) = 0。
 2. 波特率是 8 位计数器的输出时钟的 1/2。

- 备注**
1. f_{CLK6} : 由 CKSR6 寄存器的 TPS63 至 TPS60 位选择的基本时钟的频率
 2. k: 由 MDL67 至 MDL60 位设置的值 (k = 4, 5, 6, ..., 255)
 3. ×: 无关

(6) 异步串行接口控制寄存器 6 (ASICL6)

该寄存器控制串行接口 UART6 的串行通信操作。

ASICL6 可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生设置该寄存器为 16H。

注意事项 在通信操作期间 (ASIM6 的第 7 和 6 位 (POWER6, TXE6) = 1 或者 ASIM6 的第 7 和 5 位 (POWER6, RXE6) = 1), 可以通过软件刷新 ASICL6 (向其写入同样的值)。然而, 在 SBF 接收 (SBRT6 = 1) 或 SBF 发送 (自从 SBTT6 被置位 (1) 直到 INTST6 发生) 期间, 不要通过刷新操作设置 SBRT6 和 SBTT6 都为 1, 因为这样可能触发 SBF 接收或 SBF 发送。

图 13-10. 异步串行接口控制寄存器 6 (ASICL6) 的格式 (1 / 2)

地址: FF58H 复位后: 16H R/W*

符号	<7>	<6>	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6

SBRF6	SBF 接收状态标志
0	如果 POWER6 = 0 并且 RXE6 = 0 或者如果 SBF 接收已经正确完成
1	SBF 接收正在进行

SBRT6	SBF 接收触发
0	-
1	SBF 接收触发

SBTT6	SBF 发送触发
0	-
1	SBF 发送触发

注 第 7 位只读。

图 13-10. 异步串行接口控制寄存器 6 (ASICL6) 的格式 (2 / 2)

SBL62	SBL61	SBL60	SBF 发送输出宽度控制
1	0	1	SBF 以 13 位长度输出。
1	1	0	SBF 以 14 位长度输出。
1	1	1	SBF 以 15 位长度输出。
0	0	0	SBF 以 16 位长度输出。
0	0	1	SBF 以 17 位长度输出。
0	1	0	SBF 以 18 位长度输出。
0	1	1	SBF 以 19 位长度输出。
1	0	0	SBF 以 20 位长度输出。

DIR6	第一位指定
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**
1. 在 SBF 接收错误的情况下，返回到 SBF 接收模式。保持 SBRF6 标志的状态 (1)。
 2. 设置 SBRT6 位前，确保 ASIM6 的第 7 位 (POWER6) 和第 5 位 (RXE6) = 1。设置 SBRT6 位为 1 后，在 SBF 接收完成前 (在中断请求信号被产生前) 不要清除为 0。
 3. SBRT6 位的读出值总是 0，在 SBF 接收正确完成后，SBRT6 被自动清除为 0。
 4. 设置 SBTT6 位为 1 前，确保 ASIM6 的第 7 位 (POWER6) 和第 6 位 (TXE6) = 1。设置 SBTT6 位为 1 后，在 SBF 接收完成前 (在中断请求信号被产生前) 不要清除为 0。
 5. SBTT6 位的读出值总是 0，在 SBF 发送结束时，SBTT6 被自动清除为 0。
 6. 接收期间不要设置 SBRT6 位为 1，并且发送期间不要设置 SBTT6 位为 1。
 7. 重新写入 DIR6 和 TXDLV6 位之前，清除 TXE6 和 RXE6 位为 0。
 8. 如果 TXDLV6 被设置为 1 (反转 TxD6 的输出)，无论 POWER6 和 TXE6 设置为何值，TxD6/P13 引脚都不能作为通用端口使用。如果 TxD6/P13 作为通用端口使用，需设置 TXDLV6 位为 0 (TxD6 通常输出)。

(7) 输入切换控制寄存器 (ISC)

输入切换控制寄存器 (ISC) 在 LIN (本地互连网络) 接收期间用来接收主设备发送的状态信号。
 当设置 ISC0 和 ISC1 为 1 时, 从 P14/RxD6 引脚输入的信号被选择为 INTP0 和 TI000 的输入源。
 该寄存器可以由 1 位或 8 位存储器操作指令来设置。
 复位信号的产生清除该寄存器为 00H。

图 13-11. 输入切换控制寄存器 (ISC) 的格式

地址: FF4FH 复位后: 00H R/W

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

ISC1	TI000 输入源选择
0	TI000 (P00)
1	RxD6 (P14)

ISC0	INTP0 输入源选择
0	INTP0 (P120)
1	RxD6 (P14)

(8) 端口模式寄存器 1 (PM1)

该寄存器以 1 位为单位设置端口 1 输入 / 输出。
 当使用 P13 / TxD6 引脚作为串行接口数据输出时, 清除 PM13 为 0 并设置输出锁存器 P13 为 1。
 当使用 P14 / RxD6 引脚作为串行接口数据输入时, 设置 PM14 为 1。P14 的输出锁存器这时可能是 0 或 1。
 PM1 可以由 1 位或 8 位存储器操作指令来设置。
 复位信号的产生设置该寄存器为 FFH。

图 13-12. 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n 引脚 输入 / 输出模式选择 (n = 0 至 7)
0	输出模式 (输出缓冲器开)
1	输入模式 (输出缓冲器关)

13.4 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

13.4.1 操作停止模式

在这种模式下，串行通信不能被执行；因此，可以降低功耗。此外，在这种模式下引脚可以用作普通端口引脚。要设置操作停止模式，清除 ASIM6 的第 7、6 和 5 位 (POWER6、TXE6 和 RXE6) 为 0。

(1) 使用的寄存器

操作停止模式由异步串行接口操作模式寄存器 6 (ASIM6) 来设置。

ASIM6 可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生设置该寄存器为 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 (固定时钟为低电平) 并且异步复位内部电路 ^{#2} 。						
	TXE6	允许 / 禁止发送						
	0	禁止发送 (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. 在发送期间，当 POWER6 = 0 时，TxD6 引脚的输出固定为高电平 (如果 TXDLV6=0)。并且，从 RxD6 引脚输入的电平固定为高电平。
 2. 异步串行接口接收错误状态寄存器 6 (ASIS6)、异步串行接口发送状态寄存器 6 (ASIF6)、异步串行接口控制寄存器 6 (ASICL6) 的第 7 位 (SBRF6) 和第 6 位 (SBRT6) 以及接收缓存寄存器 6 (RXB6) 被复位。

注意事项 在清除 TXE6 和 RXE6 为 0 后，清除 POWER6 为 0，以停止操作。
要开始通信，设置 POWER6 为 1，然后设置 TXE6 或 RXE6 为 1。

备注 要作为通用端口引脚使用 RxD6 / P14 和 TxD6 / P13 引脚，参见 第五章 端口功能。

13.4.2 异步串行接口 (UART) 模式

在这种模式下，在起始位后发送 / 接收 1 字节的数据，并且可以执行全双工操作。它包含一个专用的波特率发生器，因此可以在很宽的波特率范围内执行通信。

(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)
- 输入切换控制寄存器 (ISC)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

在 UART 模式下设置操作的基本过程如下所示。

- <1> 设置 CKSR6 寄存器 (见 图 13-8)。
- <2> 设置 BRGC6 寄存器 (见 图 13-9)。
- <3> 设置 ASIM6 寄存器的第 0 至 4 位 (ISR6, SL6, CL6, PS60, PS61) (见 图 13-5)。
- <4> 设置 ASICL6 寄存器的第 0 和 1 位 (TXDLV6, DIR6) (见 图 13-10)。
- <5> 设置 ASIM6 寄存器的第 7 位 (POWER6) 为 1。
- <6> 设置 ASIM6 寄存器的第 6 位 (TXE6) 为 1。→ 发送允许。
设置 ASIM6 寄存器的第 5 位 (RXE6) 为 1。→ 接收允许。
- <7> 写入数据到发送缓存寄存器 6 (TXB6)。→ 数据发送开始。

注意事项 当设置端口模式寄存器和端口寄存器时，注意与通信另一方的关系。

寄存器设置和引脚之间的关系如下所示。

表 13-2. 寄存器设置和引脚之间的关系

POWER6	TXE6	RXE6	PM13	P13	PM14	P14	UART6 操作	引脚功能	
								TxD6 / P13	RxD6 / P14
0	0	0	×	×	×	×	停止	P13	P14
1	0	1	×	×	1	×	接收	P13	RxD6
	1	0	0	1	×	×	发送	TxD6	P14
	1	1	0	1	1	×	发送 / 接收	TxD6	RxD6

注 可以被设置为端口功能。

备注

- ×: 无关
- 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
- PM1×: 端口模式寄存器
- P1×: 端口输出锁存器

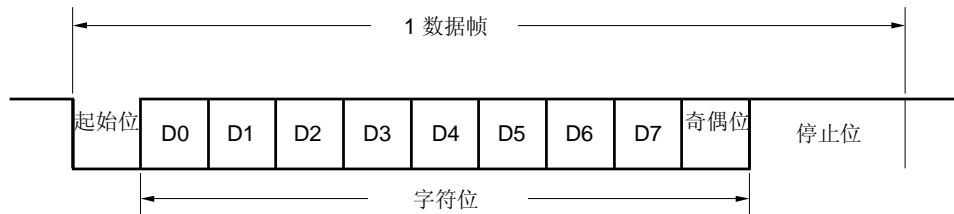
(2) 通信操作

(a) 正常发送 / 接收数据的格式和波形举例

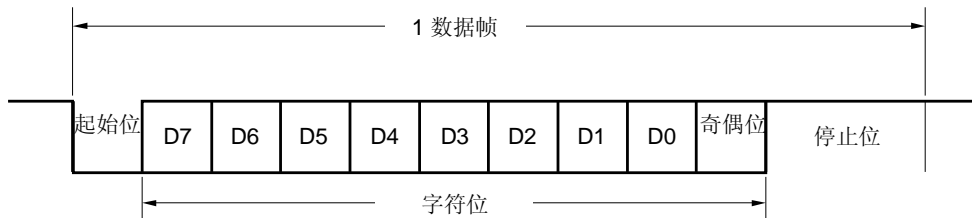
图 13-13 和 13-14 表示正常发送 / 接收数据的格式和波形举例。

图 13-13. 正常 UART 发送 / 接收数据的格式

1. LSB 在前发送 / 接收



2. MSB 在前发送 / 接收



一个数据帧由以下位组成。

- 起始位 ... 1 位
- 符号位 ... 7 或 8 位
- 校验位 ... 偶数校验, 奇数校验, 0 校验或无校验
- 停止位 ... 1 或 2 位

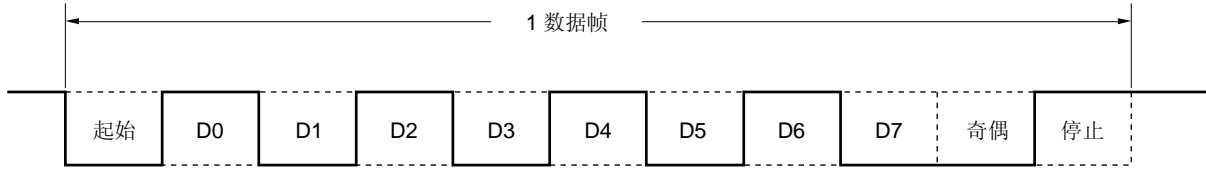
由异步串行接口操作模式寄存器 6 (ASIM6) 指定一个数据帧中的符号位长度、校验和停止位长度。

由异步串行接口控制寄存器 6 (ASICL6) 第 1 位 (DIR6) 指定是以 LSB 在前还是 MSB 在前执行数据通信。

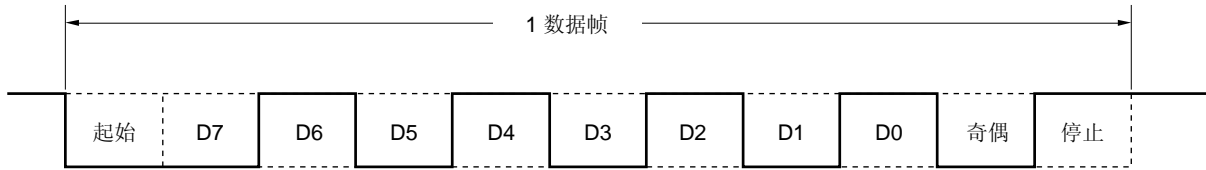
由 ASICL6 的第 0 位 (TXDLV6) 指定 Tx D6 引脚输出正常数据还是反转数据。

图 13-14. 正常 UART 发送 / 接收数据波形举例

1. 数据长度：8 位，LSB 在前，校验：偶数校验，停止位：1 位，通信数据：55H



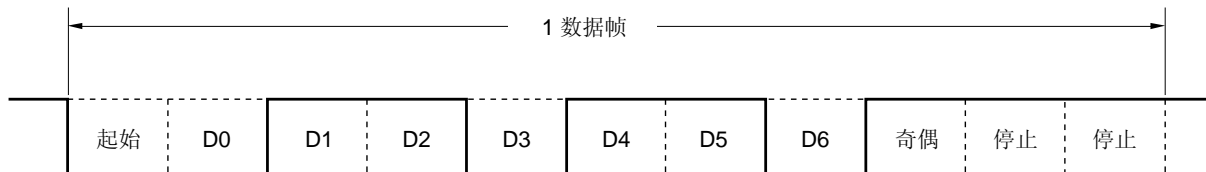
2. 数据长度：8 位，MSB 在前，校验：偶数校验，停止位：1 位，通信数据：55H



3. 数据长度：8 位，MSB 在前，校验：偶数校验，停止位：1 位，通信数据：55H，Tx/D6 引脚反转输出



4. 数据长度：7 位，LSB 在前，校验：奇数校验，停止位：2 位，通信数据：36H



5. 数据长度：8 位，LSB 在前，校验：无，停止位：1 位，通信数据：87H



(b) 校验类型和操作

校验位用来检测通信数据中的位错误。通常，在发送和接收端使用同样类型的校验位。使用偶数校验和奇数校验，可以检测 1 位（奇数个数）错误。使用零校验和无校验，不能检测错误。

注意事项 当设备用于 LIN 通信操作时，固定 PS61 和 PS60 位为 0。

(i) 偶数校验

• 发送

包含校验位的发送数据被控制以使为“1”的位个数是偶数。

校验位的值如下所示。

如果发送数据含有奇数个“1”位： 1

如果发送数据含有偶数个“1”位： 0

• 接收

包含校验位的接收数据中为“1”的位的个数被计数。如果它为奇数，则一个校验错误发生。

(ii) 奇数校验

• 发送

与偶数校验不同，包含校验位的发送数据被控制以使为“1”的位个数是奇数。

如果发送数据含有奇数个“1”位： 0

如果发送数据含有偶数个“1”位： 1

• 接收

包含校验位的接收数据中为“1”的位的个数被计数。如果它为偶数，则发生校验错误。

(iii) 零校验

当发送数据时，校验位被清除为 0，不管发送的数据是什么。

当接收数据时，不检测校验位。因此，不管校验位是“0”还是“1”，校验错误都不会发生。

(iv) 无校验

无校验位附加到发送数据上。

当接收数据时，假设没有校验位执行接收操作。因为没有校验位，不会发生校验错误。

(c) 正常发送

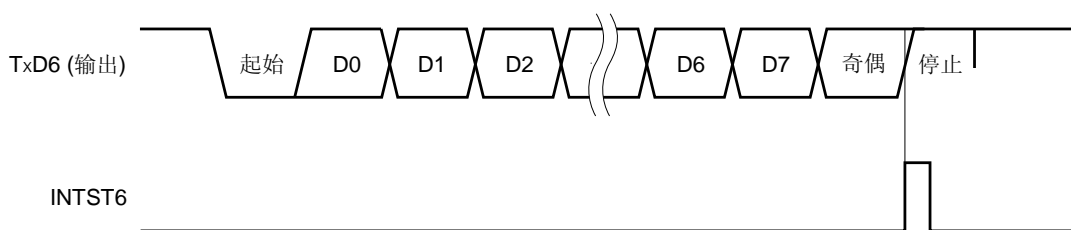
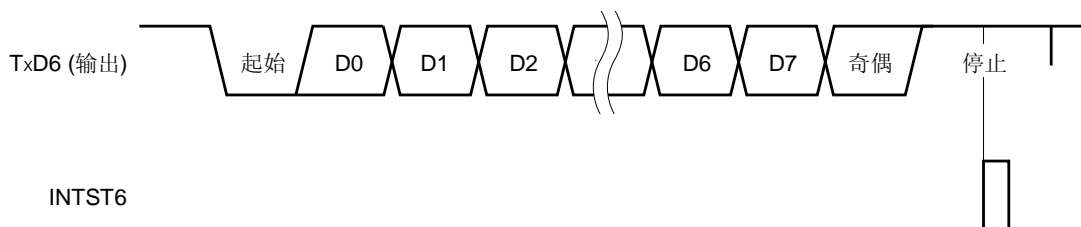
当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 被设置为 1, 然后 ASIM6 的第 6 位 (TXE6) 被设置为 1 时, 允许发送。通过写入发送数据到发送缓存寄存器 6 (TXB6), 可以启动发送。起始位、校验位和停止位自动附加到数据上。

当启动发送时, TXB6 中的数据被传送到发送移位寄存器 6 (TXS6)。然后, 按顺序从 TXS6 输出到 TxD6 引脚发送数据。当发送完成时, ASIM6 设置的校验和停止位被附加到数据上, 并且产生发送完成中断请求 (INTST6)。

在下一个要发送的数据被写入 TXB6 之前, 停止发送。

图 13-15 表示发送完成中断请求 (INTST6) 的时序。只要输出最后的停止位, 就发生这个中断。

图 13-15. 正常发送完成中断请求时序

1. 停止位长度: 1**2. 停止位长度: 2**

(d) 连续发送

只要发送移位寄存器 6 (TXS6) 已经开始移位操作, 下一个发送数据就可以被写入发送缓存寄存器 6 (TXB6)。因此, 即使正在处理一个数据帧发送后的 INTST6 中断, 也可以连续发送数据, 并且可以实现一个高效的通信速率。此外, 当发送完成中断发生时, 通过读取异步串行接口操作模式寄存器 6 (ASIM6) 的第 0 位 (TXSF6), 可以高效地两次 (2 字节) 写入 TXB6 寄存器, 而不用等待一个数据帧的发送时间。要连续发送数据, 确保参考 ASIF6 寄存器来检查发送状态和 TXB6 寄存器是否可以被写入, 然后写入数据。

- 注意事项**
1. 连续发送期间, ASIF6 寄存器的 TXBF6 和 TXSF6 标志会从“10”更改为“11”和更改为“01”。因此, 要检查状态, 不要使用 TXBF6 和 TXSF6 标志的组合来判断。当执行连续发送时, 只读取 TXBF6 标志。
 2. 当设备用于 LIN 通信操作时, 不能使用连续发送功能。在将待发送的数据写入到发送缓存寄存器 6 (TXB6) 之前, 确保异步串行接口发送状态寄存器 6 (ASIF6) 为 00H。

TXBF6	写入 TXB6 寄存器
0	写入允许
1	写入禁止

注意事项 要连续发送数据, 将第一个待发送的数据 (第一个字节) 写入到 TXB6 寄存器。确保检查 TXBF6 标志为“0”。如果满足, 写入下一个待发送的数据 (第二个字节) 到 TXB6 寄存器。如果当 TXBF6 标志为“1”时将数据写入到 TXB6 寄存器, 不能保证发送数据的正确性。

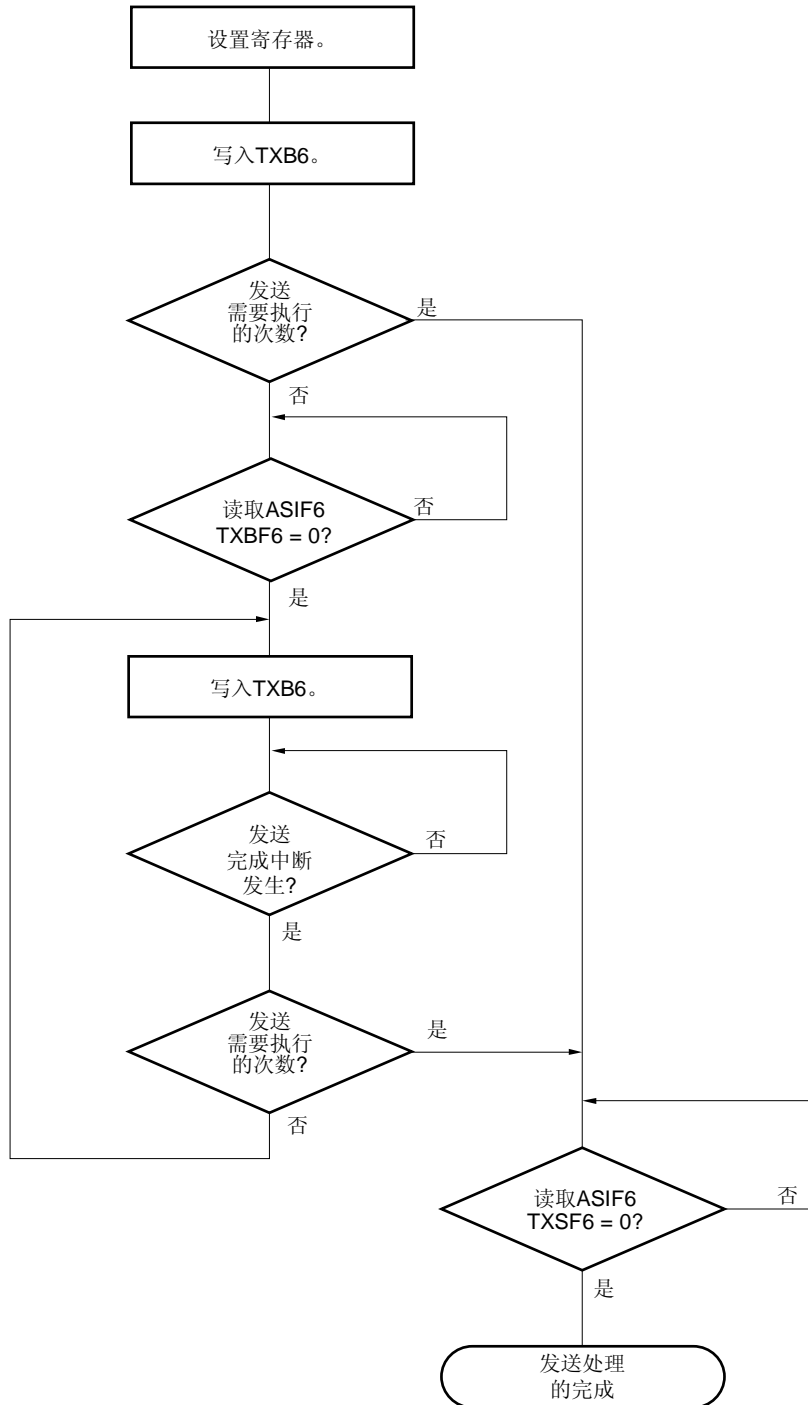
通信状态可以使用 TXSF6 标志来检查。

TXSF6	发送状态
0	发送完成。
1	发送进行中。

- 注意事项**
1. 要在连续发送完成时初始化发送单元, 确保检查在发送完成中断产生后 TXSF6 标志位为“0”, 然后执行初始化。如果当 TXSF6 标志为“1”时执行初始化, 不保证发送数据的正确性。
 2. 连续发送期间, 一个数据帧发送后, 下一个发送可能在 INTST6 中断服务程序执行前完成。对此, 通过开发一个可以计数发送数据个数的程序并通过参考 TXSF6 标志, 可以实现检测。

图 13-16 表示连续发送处理流程的举例。

图 13-16. 连续发送处理流程举例

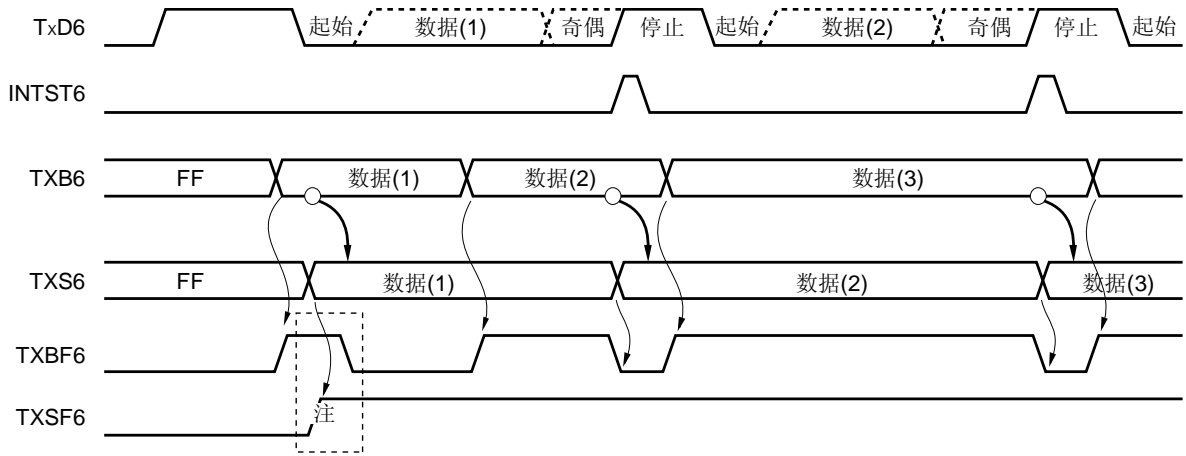


备注

- TXB6: 发送缓存寄存器 6
- ASIF6: 异步串行接口操作模式寄存器 6
- TXBF6: ASIF6 的第 1 位 (发送缓存数据标志)
- TXSF6: ASIF6 的第 0 位 (发送移位寄存器数据标志)

图 13-17 表示开始连续发送的时序，图 13-18 表示结束连续发送的时序。

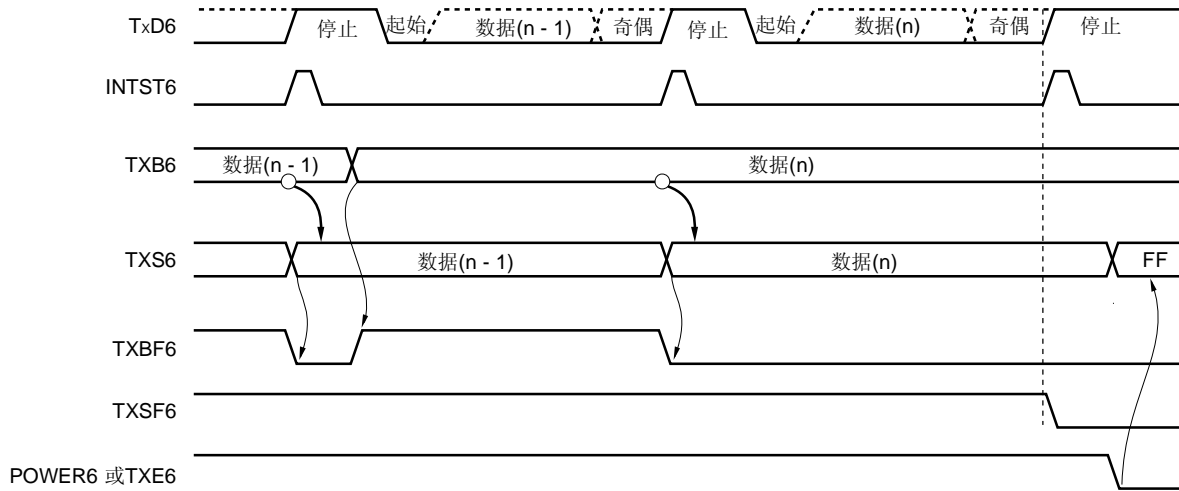
图 13-17. 开始连续发送的时序



注 当 ASIF6 被读取时，存在一段时间 TXBF6 和 TXSF6 = 1, 1。因此，只使用 TXBF6 位来判断是否允许写入。

- 备注**
- TxD6: TxD6 引脚（输出）
 - INTST6: 中断请求信号
 - TXB6: 发送缓存寄存器 6
 - TXS6: 发送移位寄存器 6
 - ASIF6: 异步串行接口操作模式寄存器 6
 - TXBF6: ASIF6 的第 1 位
 - TXSF6: ASIF6 的第 0 位

图 13-18. 结束连续发送的时序



- 备注**
- TxD6: TxD6 引脚（输出）
 - INTST6: 中断请求信号
 - TXB6: 发送缓存寄存器 6
 - TXS6: 发送移位寄存器 6
 - ASIF6: 异步串行接口操作模式寄存器 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 7（ASIM6）的第 6 位
 - TXE6: 异步串行接口操作模式寄存器 6（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) 被设置为 1, 然后 ASIM6 的第 5 位 (RXE6) 被设置为 1 时, 允许接收并且对 RxD6 引脚的输入进行采样。

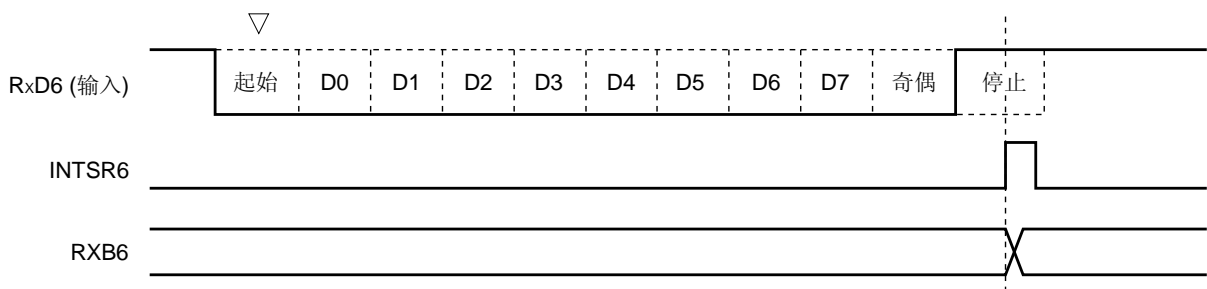
当检测到 RxD6 引脚输入的下降沿时, 波特率发生器的 8 位计数器开始计数。当计数值达到波特率发生器控制寄存器 6 (BRGC6) 的设置值时, 再次采样 RxD6 引脚的输入 (图 13-19 中的 ▽)。如果 RxD6 引脚这时是低电平, 它被识别为起始位。

当起始位被检测时, 接收开始, 并且串行数据按照设置的波特率存储到接收移位寄存器 6 (RXS6) 中。当停止位被接收时, 将产生接收完成中断 (INTSR6), 并且 RXS6 的数据被写入到接收缓存寄存器 6 (RXB6)。

然而, 如果一个溢出错误 (OVE6) 发生, 接收数据不会被写入到 RXB6。

当接收进行时, 即使校验错误 (PE6) 发生, 将继续执行接收操作直到接收到停止位, 并且在接收完成时, 产生接收错误中断 (INTSR6 / INTSRE6)。

图 13-19. 接收完成中断请求时序



- 注意事项**
1. 如果接收错误发生, 读取 **ASIS6**, 然后读取 **RXB6** 清除错误标志。否则, 当接收下一个数据时, 一个溢出错误将发生, 并且保持接收错误状态。
 2. 接收总是按照“停止位个数 = 1”来执行。第二个停止位被忽略。
 3. 读取 **RXB6** 前确保读取异步串行接口接收错误状态寄存器 6 (**ASIS6**)。

(f) 接收错误

接收期间可能发生三种错误：校验错误、帧错误或溢出错误。如果由于数据接收异步串行接口接收错误状态寄存器 6 (ASIS6) 的错误标志标志位被置位，则产生接收错误中断请求 (INTSR6 / INTSRE6)。

通过在接收错误中断 (INTSR6 / INTSRE6) 服务程序中读取 ASIS6 的内容，可确定接收期间出现的是何种错误 (见 图 13-6)。

当读取 ASIS6 时，ASIS6 的内容被清除为 0。

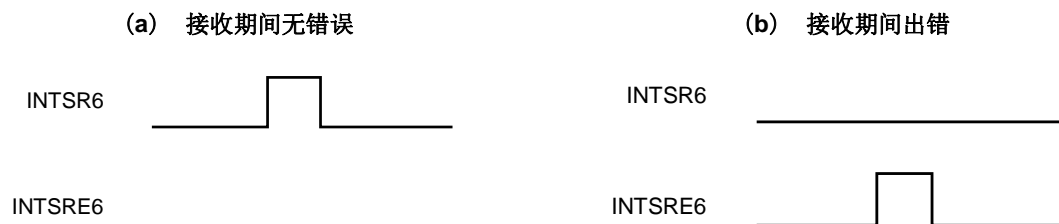
表 13-3. 接收错误的原因

接收错误	原因
校验错误	发送指定的校验与接收数据的校验不匹配。
帧错误	未检测到停止位。
溢出错误	下一个数据的接收在数据从接收缓存寄存器 6 (RXB6) 读取前完成。

通过清除异步程序接口操作模式寄存器 6 (ASIM6) 的第 0 位 (ISRM6) 为 0，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)

图 13-20. 接收错误中断

1. 如果 ISRM6 被清除为 0 (接收完成中断 (INTSR6) 和错误中断 (INTSRE6) 被分开)



2. 如果 ISRM6 被设置为 1 (错误中断包含在 INTSR6 中)



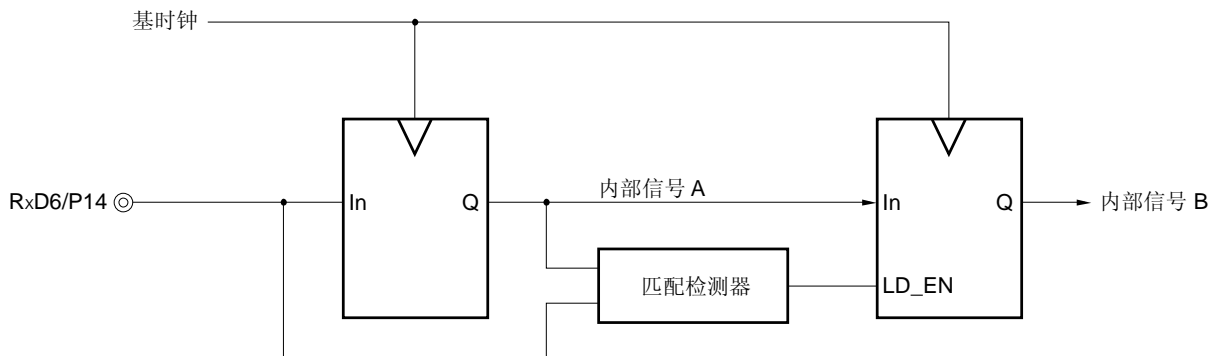
(g) 接收数据的噪声滤波器

RxD6 信号由预分频模块输出的基本时钟采样。

如果两个采样值相同，匹配检测的输出发生改变，并把采样的数据作为输入数据。

因为电路的配置如图 13-21 所示，接收操作的内部处理从外部信号状态开始延迟两个时钟才执行。

图 13-21. 噪声滤波器电路

**(h) SBF 发送**

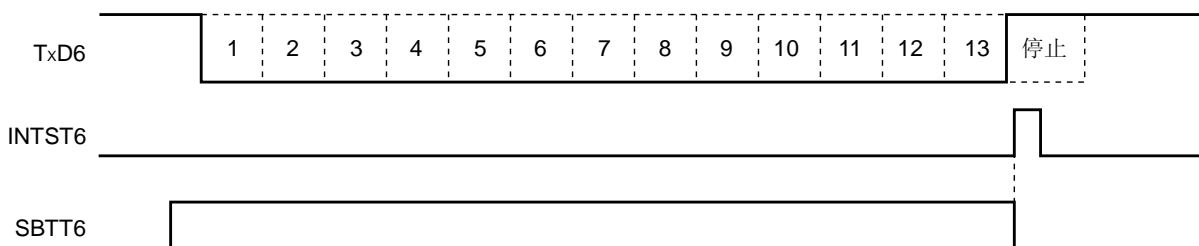
当设备用于 LIN 通信操作时，SBF（同步中断区域）发送控制功能被用于发送。关于 LIN 的发送操作，见图 13-1 LIN 发送操作。

当异步串行接口模式寄存器 6（ASIM6）的第 7 位（POWER6）设置为 1 时，TxD6 引脚输出高电平。然后，当 ASIM6 的第 6 位（TXE6）设置为 1 时，进入发送允许状态，并且通过设置异步串行接口控制寄存器 6（ASICL6）的第 5 位（SBTT6）位 1 来启动 SBF 发送。

然后，13 至 20 位（由 ASICL6 的第 4 至 2 位（SBL62 至 SBL60）设置）输出低电平。SBF 发送完成后，将产生发送完成中断请求（INTST6），并且 SBTT6 被自动清除。然后，恢复正常发送模式。

直到下一个发送的数据被写入发送缓存寄存器 6（TXB6），或者直到 SBTT6 被设置为 1 时，才停止 SBF 发送。

图 13-22. SBF 发送



备注 TxD6: TxD6 引脚（输出）
 INTST6: 发送完成中断请求
 SBTT6: 异步串行接口控制寄存器 6（ASICL6）的第 5 位

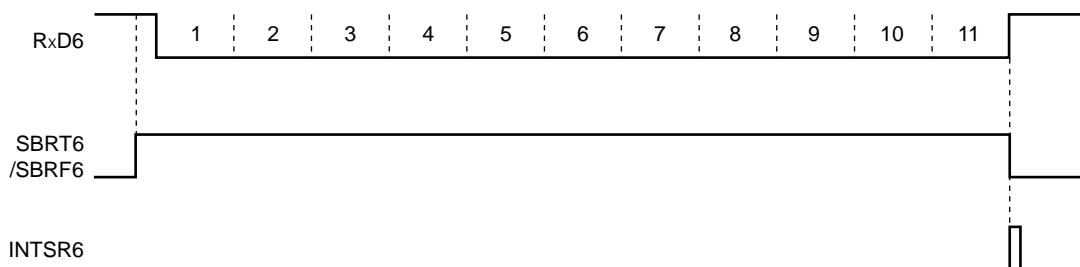
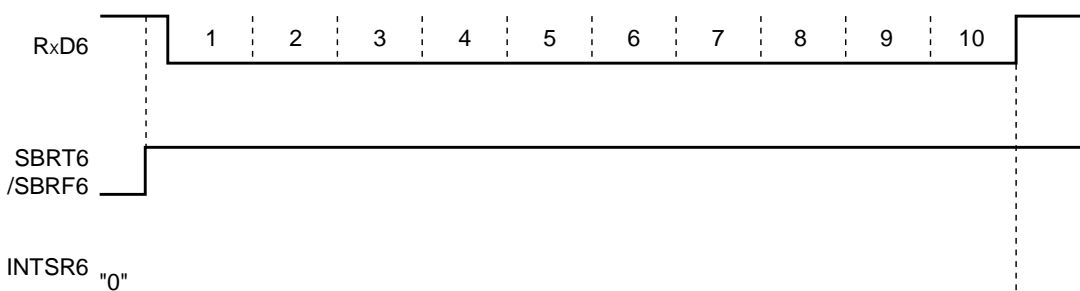
(i) SBF 接收

当设备用于 LIN 通信操作时，SBF（同步中断区域）接收控制功能被用于接收。关于 LIN 的接收操作，见图 13-2 LIN 接收操作。

当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 被设置为 1，然后 ASIM6 的第 5 位 (RXE6) 被设置为 1 时，允许接收。当异步串行接口控制寄存器 6 (ASICL6) 的第 6 位 (SBRT6) 被设置为 1 时，允许 SBF 接收。在 SBF 接收允许状态下，采样 RxD6 引脚，并且按照与正常接收允许状态相同的方式检测起始位。

当检测到起始位时，开始接收，并且串行数据按照设置的波特率存储到接收移位寄存器 6 (RXS6) 中。当接收到停止位时，并且如果 SBF 的宽度大于或等于 11 位时，像正常处理一样产生接收完成中断请求 (INTSR6)。这时，SBRF6 和 SBRT6 位被自动清除，SBF 接收结束。禁止检测 OVE6、PE6 和 FE6（异步串行接口接收错误状态寄存器 6 (ASIS6) 的第 0 至 2 位）错误，不进行 UART 通信的错误检测。此外，不执行接收移位寄存器 6 (RXS6) 和接收缓存寄存器 6 (RXB6) 之间的数据传送，并且保持复位值 FFH。如果 SBF 的宽度为 10 位或更少，由于停止位接收后的错误处理，不会产生中断，并且恢复 SBF 接收模式。在这种情况下，不清除 SBRF6 和 SBRT6 位。

图 13-23. SBF 接收

1. 正常 SBF 接收（多于 10.5 位的宽度的停止位被检测）**2. SBF 接收错误（10.5 位或更少的宽度的停止位被检测）**

备注 RxD6: RxD6 引脚（输入）
 SBRT6: 异步串行接口控制寄存器 6 (ASICL6) 的第 6 位
 SBRF6: ASICL6 的第 7 位
 INTSR6: 接收完成中断请求

13.4.3 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，并且为 UART6 的发送 / 接收产生串行时钟。单独的 8 位计数器提供给发送和接收。

(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) 被设置为 1 时，由时钟选择寄存器 6 (CKSR6) 的第 3 至 0 位 (TPS63 至 TPS60) 选择的时钟提供给每个模块。这个时钟叫做基本时钟，其频率称为 f_{CLK6} 。当 POWER6 = 0 时，基本时钟固定为低电平。

- 发送计数器

当串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 或第 6 位 (TXE6) 为 0 时，这个计数器停止操作，清除为 0。

当 POWER6 = 1 并且 TXE6 = 1 时，它开始计数。

当发送的第一个数据被写入发送缓存寄存器 6 (TXB6) 时，计数器被清除为 0。

如果连续发送数据，当一帧数据发送完成时，计数器再次被清除为 0。如果后面没有数据被发送，不清除计数为 0，并且在 POWER6 或 TXE6 被清除为 0 之前继续计数。

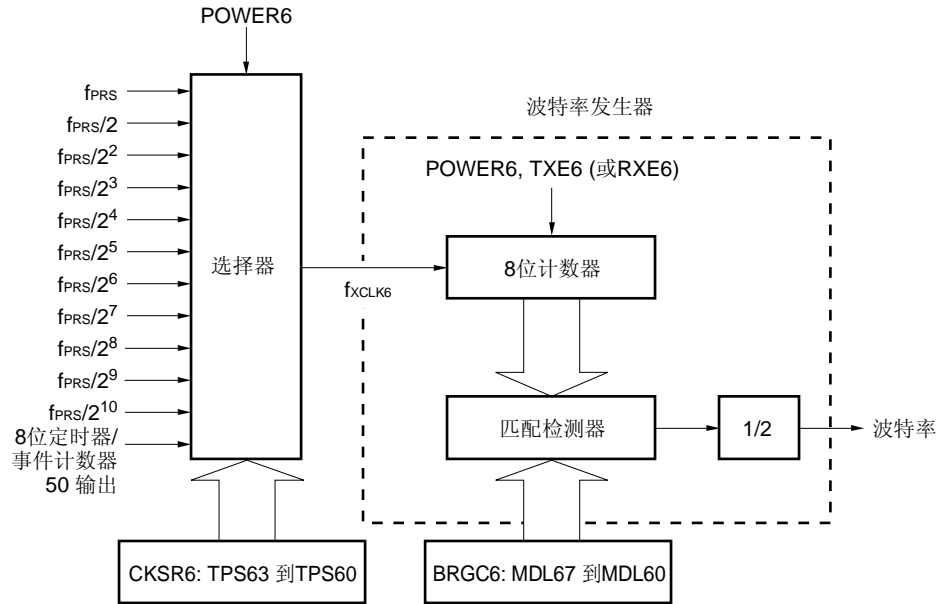
- 接收计数器

当串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 或第 5 位 (RXE6) 为 0 时，这个计数器停止操作，清除为 0。

当检测到起始位时，开始计数。

在一帧接收后，下一个起始位检测前，计数器停止操作。

图 13-24. 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

可以使用时钟选择寄存器 6 ($CKSR6$) 和波特率发生器控制寄存器 6 ($BRGC6$) 指定要产生的串行时钟。可以由 $CKSR6$ 的第 3 至 0 位 (TPS63 至 TPS60) 设置输入到 8 位计数器的时钟, 可以由 $BRGC6$ 的第 7 至 0 位 (MDL67 至 MDL60) 设置 8 位计数器的分频值 ($f_{XCLK6}/4$ 至 $f_{XCLK6}/255$)。

13.4.4 波特率的计算

(1) 波特率计算表达式

波特率可以由以下表达式计算。

$$\bullet \text{ 波特率} = \frac{f_{\text{CLK6}}}{2 \times k} [\text{bps}]$$

f_{CLK6} : 由 CKSR6 寄存器的 TPS63 至 TPS60 位选择的基本时钟的频率

k : 由 BRGC6 寄存器的 MDL67 至 MDL60 位设置的值 ($k = 4, 5, 6, \dots, 255$)

表 13-4. TPS63 至 TPS60 的设置值

TPS63	TPS62	TPS61	TPS60	基本时钟 (f_{CLK6}) 选择 ^{注1}				
				$f_{\text{PRS}} =$ 4 MHz	$f_{\text{PRS}} =$ 8 MHz	$f_{\text{PRS}} =$ 10 MHz	$f_{\text{PRS}} =$ 20 MHz	
0	0	0	0	$f_{\text{PRS}}^{\text{注2}}$	4 MHz	8 MHz	10 MHz	20 MHz ^{注3}
0	0	0	1	$f_{\text{PRS}}/2$	2 MHz	4 MHz	5 MHz	10 MHz
0	0	1	0	$f_{\text{PRS}}/2^2$	1 MHz	2 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{\text{PRS}}/2^3$	500 kHz	1 MHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{\text{PRS}}/2^4$	250 kHz	500 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{\text{PRS}}/2^5$	125 kHz	250 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{\text{PRS}}/2^6$	62.5 kHz	125 kHz	156.25 kHz	312.5 kHz
0	1	1	1	$f_{\text{PRS}}/2^7$	31.25 kHz	62.5 kHz	78.13 kHz	156.25 kHz
1	0	0	0	$f_{\text{PRS}}/2^8$	15.625 kHz	31.25 kHz	39.06 kHz	78.13 kHz
1	0	0	1	$f_{\text{PRS}}/2^9$	7.813 kHz	15.625 kHz	19.53 kHz	39.06 kHz
1	0	1	0	$f_{\text{PRS}}/2^{10}$	3.906 kHz	7.813 kHz	9.77 kHz	19.53 kHz
1	0	1	1	TM50 输出 ^{注4}				
除上面以外				禁止设置				

- 注
- 如果外围硬件时钟 (f_{PRS}) 工作于高速系统时钟 (f_{XH}) ($\text{XSEL} = 1$)， f_{PRS} 工作频率根据供电电压改变。
 - $V_{\text{DD}} = 2.7$ 至 5.5 V : $f_{\text{PRS}} \leq 20$ MHz
 - $V_{\text{DD}} = 1.8$ 至 2.7 V : $f_{\text{PRS}} \leq 5$ MHz
 - 如果外围硬件时钟 (f_{PRS}) 工作于内部高速振荡时钟 (f_{RH}) ($\text{XSEL} = 0$)，当 $1.8 \text{ V} \leq V_{\text{DD}} < 2.7 \text{ V}$ 时，禁止 TPS63 = TPS62 = TPS61 = TPS60 = 0 的设置 (计数时钟: f_{PRS})。
 - 只有在 $4.0 \text{ V} \leq V_{\text{DD}} \leq 5.5 \text{ V}$ 的情况下才可以设置。
 - 当选择 TM50 输出作为基本时钟时，注意以下要点。
 - TM50 和 CR50 匹配时清零&启动模式 ($\text{TMC506} = 0$)
开始 8 位定时器 / 事件计数器 50 的操作，然后允许定时器 F/F 反转操作 ($\text{TMC501} = 1$)。
 - PWM 模式 ($\text{TMC506} = 1$)
开始 8 位定时器 / 事件计数器 50 的操作，然后设置计数时钟使占空比 = 50%。
在任何模式下都不必允许 ($\text{TOE50} = 1$) TO50 输出。

(2) 波特率误差

波特率误差可以由以下表达式计算。

$$\bullet \text{ 误差 (\%)} = \left[\frac{\text{实际波特率 (带有误差的波特率)}}{\text{期望的波特率 (正确的波特率)}} - 1 \right] \times 100 [\%]$$

- 注意事项**
1. 在发送期间，确保波特率误差在接收端的允许误差范围之内。
 2. 确保接收期间的波特率误差满足 (4) 接收期间允许的波特率范围中描述的范围。

举例： 基本时钟频率 = 10 MHz = 10,000,000 Hz
BRGC6 寄存器的 MDL67 至 MDL60 位的设置值 = 00100001B (k = 33)
目标波特率 = 153600 bps

$$\begin{aligned} \text{波特率} &= 10 \text{ M} / (2 \times 33) \\ &= 10000000 / (2 \times 33) = 151,515 [\text{bps}] \end{aligned}$$

$$\begin{aligned} \text{误差} &= (151515/153600 - 1) \times 100 \\ &= -1.357 [\%] \end{aligned}$$

(3) 设置波特率举例

表 13-5. 波特率发生器的设置数据

波特率 [bps]	f _{PRS} = 2.0 MHz				f _{PRS} = 5.0 MHz				f _{PRS} = 10.0 MHz				f _{PRS} = 20.0 MHz			
	TPS63- TPS60	k	计算的 值	ERR [%]	TPS63- TPS60	k	计算的 值	ERR [%]	TPS63- TPS60	k	计算的 值	ERR [%]	TPS63- TPS60	k	计算的 值	ERR [%]
300	8H	13	301	0.16	7H	65	301	0.16	8H	65	301	0.16	9H	65	301	0.16
600	7H	13	601	0.16	6H	65	601	0.16	7H	65	601	0.16	8H	65	601	0.16
1200	6H	13	1202	0.16	5H	65	1202	0.16	6H	65	1202	0.16	7H	65	1202	0.16
2400	5H	13	2404	0.16	4H	65	2404	0.16	5H	65	2404	0.16	6H	65	2404	0.16
4800	4H	13	4808	0.16	3H	65	4808	0.16	4H	65	4808	0.16	5H	65	4808	0.16
9600	3H	13	9615	0.16	2H	65	9615	0.16	3H	65	9615	0.16	4H	65	9615	0.16
19200	2H	13	19231	0.16	1H	65	19231	0.16	2H	65	19231	0.16	3H	65	19231	0.16
24000	1H	21	23810	-0.79	3H	13	24038	0.16	4H	13	24038	0.16	5H	13	24038	0.16
31250	1H	16	31250	0	4H	5	31250	0	4H	15	31250	0	5H	10	31250	0
38400	1H	13	38462	0.16	0H	65	38462	0.16	1H	65	38462	0.16	2H	65	38462	0.16
48000	0H	21	47619	-0.79	2H	13	48077	0.16	3H	13	48077	0.16	4H	13	48077	0.16
76800	0H	13	76923	0.16	0H	33	75758	-1.36	0H	65	76923	0.16	1H	65	76923	0.16
115200	0H	9	111111	-3.55	1H	11	113636	-1.36	0H	43	116279	0.94	0H	87	114943	-0.22
153600	-	-	-	-	1H	8	156250	1.73	0H	33	151515	-1.36	1H	33	151515	-1.36
312500	-	-	-	-	0H	8	312500	0	1H	8	312500	0	2H	8	312500	0
625000	-	-	-	-	0H	4	625000	0	1H	4	625000	0	2H	4	625000	0

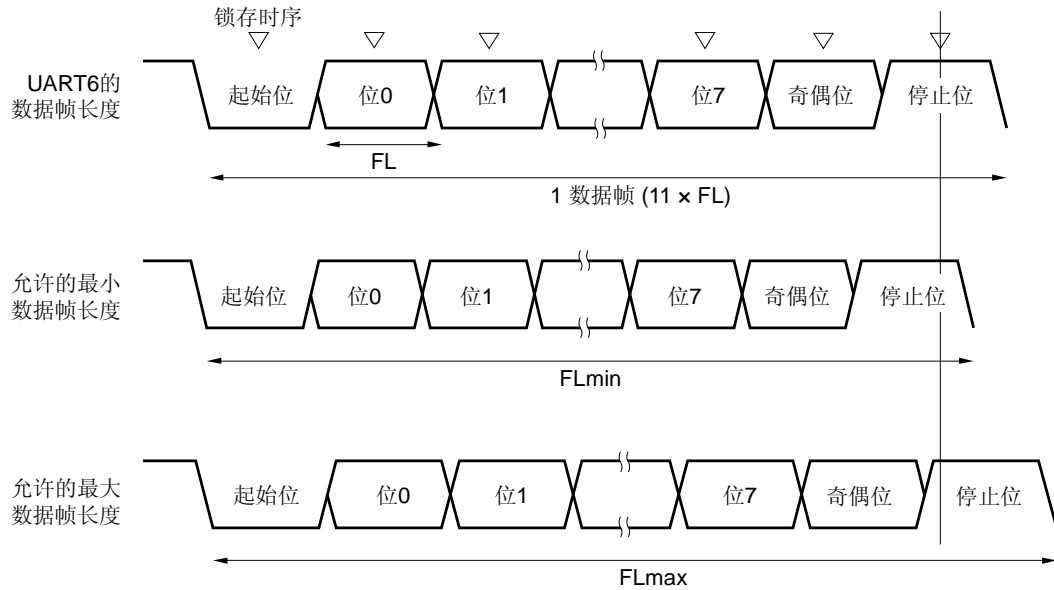
备注 TPS63 至 TPS60: 时钟选择寄存器 6 (CKSR6) 的第 3 至 0 位 (基本时钟 (f_{XCLK6}) 的设置)
k: 由波特率发生器控制寄存器 6 (BRGC6) 的 MDL67 至 MDL60 位设置的值 (k = 4, 5, 6, ..., 255)
f_{PRS}: 外围硬件时钟频率
ERR: 波特率误差

(4) 接收期间允许的波特率范围

接收期间与发送端的允许误差如下所示。

注意事项 通过使用下面的计算表达式，确保接收期间的波特率误差在允许的误差范围内。

图 13-25. 接收期间允许的波特率范围



如图 13-25 所示，在起始位被检测后，接收数据的锁存时序由通过波特率发生器控制寄存器 6 (BRGC6) 设置的计数器决定。如果最后一个数据 (停止位) 满足这个锁存时序，可以正确接收数据。

假设 11 位数据被接收，理论值可以按照下面计算。

$$FL = (\text{Brate})^{-1}$$

Brate: UART6 的波特率

k: BRGC6 的设置值

FL: 1 位数据长度

锁存时序的余量: 2 时钟

$$\text{允许的最小数据帧长度: } FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

因此，发送端的最大可接收波特率如下所示。

$$BR_{\max} = (FL_{\min}/11)^{-1} = \frac{22k}{21k+2} \text{ Brate}$$

类似的，允许的最大数据帧长度可以按照下面计算。

$$\frac{10}{11} \times FL_{\max} = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FL_{\max} = \frac{21k-2}{20k} FL \times 11$$

因此，发送端的最小可接收波特率如下所示。

$$BR_{\min} = (FL_{\max}/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

UART6 和发送端之间允许的波特率误差可以从上面的最小和最大波特率表达式计算出来，如下所示。

表 13-6. 最大 / 最小允许波特率误差

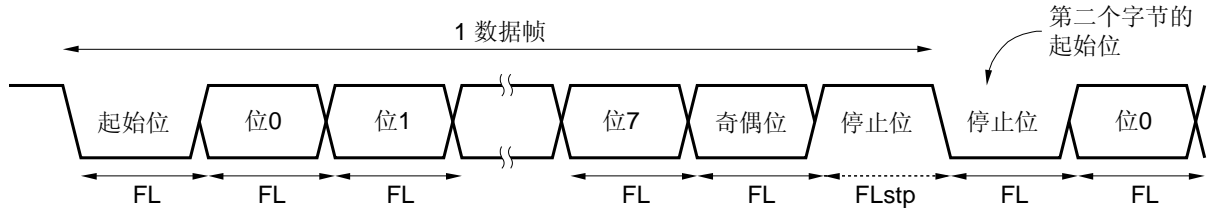
分频比率 (k)	最大允许波特率误差	最小允许波特率误差
4	+2.33%	-2.44%
8	+3.53%	-3.61%
20	+4.26%	-4.31%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.73%

- 备注**
1. 接收的允许误差依赖于帧的位的个数、输入时钟频率和分频比率 (k)。输入时钟频率越高并且分频 (k) 比率越大，允许的误差越大。
 2. k: BRGC6 的设置值

(5) 连续发送期间的数据帧长度

当数据被连续发送时，从停止位到下一个起始位的数据帧长度从正常值延长了两个基本时钟周期。然而，因为当起始位被检测时，接收端的时序被初始化，通信的结果不受影响。

图 13-26. 连续发送期间的数据帧长度



1 位数据长度为 FL，停止位长度为 FLstp 并且基本时钟频率为 f_{CLK6} ，满足以下表达式。

$$\text{FLstp} = \text{FL} + 2/f_{\text{CLK6}}$$

因此，连续发送期间的数据帧长度为：

$$\text{数据帧长度} = 11 \times \text{FL} + 2/f_{\text{CLK6}}$$

14.1 串行接口CSI10 的功能

串行接口 CSI10 有以下两种模式。

- 操作停止模式
- 3 线串行输入 / 输出模式

(1) 操作停止模式

在未执行串行通信时使用该模式，可以降低功耗。

关于详细信息，见 **14.4.1 操作停止模式**。

(2) 3 线串行输入 / 输出模式 (可选 MSB / LSB)

该模式使用三条线（一条串行时钟线（SCK10）和两条串行数据线（SI10 和 SO10））进行 8 位数据的通信：。

在 3 线串行输入 / 输出模式下，因为可以同时执行发送和接收，可以缩短数据通信的处理时间。

此外，8 位数据传输时可以指定 MSB 或者 LSB，所以该接口可以连接到任意设备。

3 线串行输入 / 输出模式用于连接带时钟驱动串行接口的外围芯片和显示控制器。

关于详细信息，见 **14.4.2 3 线串行输入 / 输出模式**。

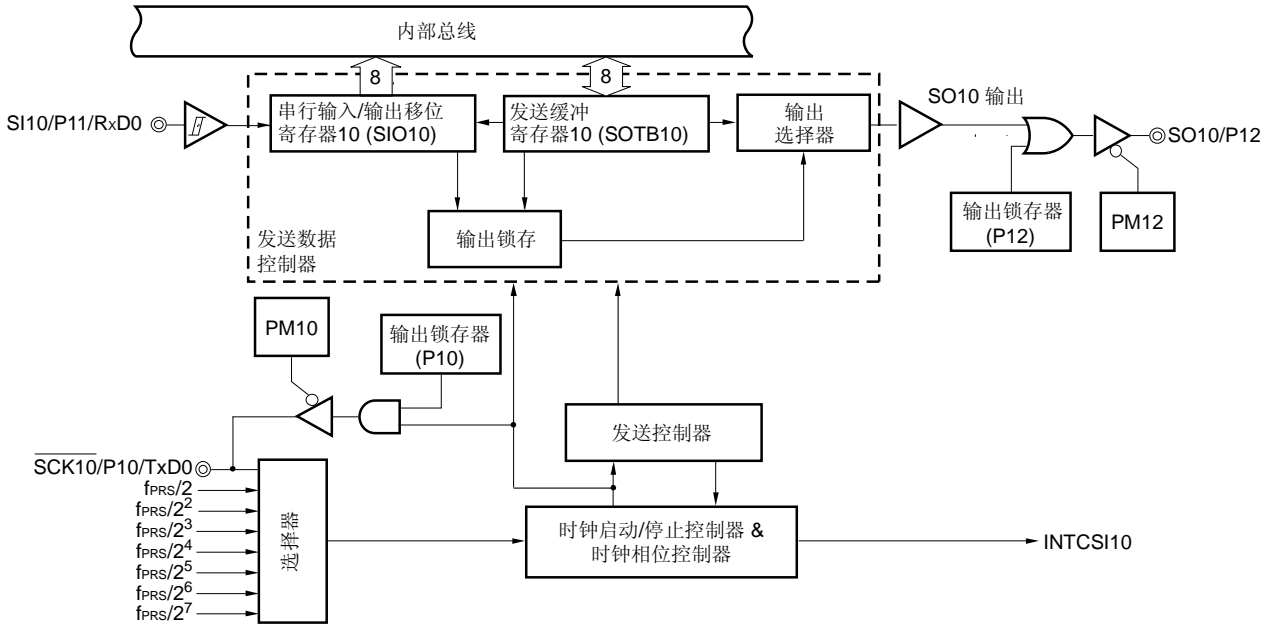
14.2 串行接口CSI10 的配置

串行接口 CSI10 包含以下硬件。

表 14-1. 串行接口 CSI10 的配置

项目	配置
控制器	发送控制器 时钟启动 / 停止控制器&时钟相位控制器
寄存器	发送缓冲寄存器 10 (SOTB10) 串行输入 / 输出移位寄存器 10 (SIO10)
控制寄存器	串行操作模式寄存器 10 (CSIM10) 串行时钟选择寄存器 10 (CSIC10) 端口模式寄存器 1 (PM1) 端口寄存器 1 (P1)

图 14-1. 串行接口 CSI10 的框图



(1) 发送缓冲寄存器 10 (SOTB10)

该寄存器用于设置发送数据。

当串行操作模式寄存器 10 (CSIM10) 的第 7 位 (CSIE10) 和第 6 位 (TRMD10) 为 1 时, 通过向 SOTB10 写入数据, 启动发送 / 接收。

写入 SOTB10 的数据由串行输入 / 输出移位寄存器 10 从并行数据转换为串行数据, 并输出到串行输出引脚 (SO10)。

SOTB10 可以由 8 位存储器操作指令写入或读取。

复位信号的产生设置该寄存器为 00H。

注意事项 当 CSOT10 = 1 (串行通信期间) 时, 不要访问 SOTB10。

(2) 串行输入 / 输出移位寄存器 10 (SIO10)

这是将数据从并行数据转换为串行数据或者反向操作的 8 位寄存器。

该寄存器可以由 8 位存储器操作指令来读取。

如果串行操作模式寄存器 10 (CSIM10) 的第 6 位 (TRMD10) 为 0, 通过从 SIO10 读取数据, 启动接收。

接收期间, 数据从串行输入引脚 (SI10) 读取到 SIO10。

复位信号的产生设置该寄存器为 00H。

注意事项 当 CSOT10 = 1 (串行通信期间) 时, 不要访问 SIO10。

14.3 控制串行接口CSI10的寄存器

串行接口 CSI10 由以下四个寄存器控制。

- 串行操作模式寄存器 10 (CSIM10)
- 串行时钟选择寄存器 10 (CSIC10)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

(1) 串行操作模式寄存器 10 (CSIM10)

CSIM10 用于选择操作模式以及允许或禁止操作。

CSIM10 可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生设置该寄存器为 00H。

图 14-2. 串行操作模式寄存器 10 (CSIM10) 的格式

地址: FF80H 复位后: 00H R / W^{注1}

符号	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10
CSIE10	3 线串行输入 / 输出模式下的操作控制							
0	禁止操作 ^{注2} 并且异步复位内部电路 ^{注3} 。							
1	允许操作							
TRMD10 ^{注4}	发送 / 接收模式控制							
0 ^{注5}	接收模式 (禁止发送)。							
1	发送 / 接收模式							
DIR10 ^{注6}	第一位指定							
0	MSB							
1	LSB							
CSOT10	通信状态标志							
0	通信停止。							
1	通信进行中。							

- 注
1. 第 0 位是只读位。
 2. 要设置 P10 / $\overline{\text{SCK10}}$ / TxD0 和 P12 / SO10 为通用端口, 设置 CSIM10 为默认状态 (00H)。
 3. CSIM10 的第 0 位 (CSOT10) 和串行输入 / 输出移位寄存器 10 (SIO10) 复位。
 4. 当 CSOT10 = 1 (串行通信期间) 时, 不要重新写入 TRMD10。
 5. 当 TRMD10 为 0 时, SO10 的输出 (见图 14-1) 固定为低电平。当从 SIO10 读取数据时, 启动接收。
 6. 当 CSOT10 = 1 (串行通信期间) 时, 不要重新写入 DIR10。

注意事项 确保清除第 5 位和第 3 至 1 位为 0。

(2) 串行时钟选择寄存器 10 (CSIC10)

该寄存器指定数据发送 / 接收的时序并设置串行时钟。

CSIC10 可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生设置该寄存器为 00H。

图 14-3. 串行时钟选择寄存器 10 (CSIC10) 的格式

地址: FF81H 复位后: 00H R / W

符号	7	6	5	4	3	2	1	0
CSIC10	0	0	0	CKP10	DAP10	CKS102	CKS101	CKS100

CKP10	DAP10	指定数据发送 / 接收时序的	类型
0	0		1
0	1		2
1	0		3
1	1		4

CKS102	CKS101	CKS100	CSI10 串行时钟选择 ^{1,2}				模式	
			f _{PRS} = 4 MHz	f _{PRS} = 8 MHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz		
0	0	0	f _{PRS} /2	2 MHz	4 MHz	5 MHz	禁止设置	主模式
0	0	1	f _{PRS} /2 ²	1 MHz	2 MHz	2.5 MHz	5 MHz	
0	1	0	f _{PRS} /2 ³	500 kHz	1 MHz	1.25 MHz	2.5 MHz	
0	1	1	f _{PRS} /2 ⁴	250 kHz	500 kHz	625 kHz	1.25 MHz	
1	0	0	f _{PRS} /2 ⁵	125 kHz	250 kHz	312.5 kHz	625 kHz	
1	0	1	f _{PRS} /2 ⁶	62.5 kHz	125 kHz	156.25 kHz	312.5 kHz	
1	1	0	f _{PRS} /2 ⁷	31.25 kHz	62.5 kHz	78.13 kHz	156.25 kHz	
1	1	1	输入到 SCK10 的外部时钟 ³				从模式	

注 1. 如果外围硬件时钟 (f_{PRS}) 工作于高速系统时钟 (f_{XH}) (XSEL = 1), f_{PRS} 工作频率根据供电电压改变。

- V_{DD} = 2.7 至 5.5 V: f_{PRS} ≤ 20 MHz
- V_{DD} = 1.8 至 2.7 V: f_{PRS} ≤ 5 MHz

注 2. 按如下条件设置串行时钟。

- $V_{DD} = 4.0$ 至 5.5 V: 串行时钟 ≤ 6.25 MHz
- $V_{DD} = 2.7$ 至 4.0 V: 串行时钟 ≤ 4 MHz
- $V_{DD} = 1.8$ 至 2.7 V: 串行时钟 ≤ 2 MHz

3. 在 STOP 模式下, 不要启动从 $\overline{SCK10}$ 引脚输入的外部时钟的串行通信。

注意事项 1. 当 $CSIE10 = 1$ (操作允许) 时, 不要写入 $CSIC10$ 。

2. 要使用 $P10 / \overline{SCK10} / TxD0$ 和 $P12 / SO10$ 作为通用端口, 设置 $CSIC10$ 为默认状态 (00H) 。

3. 复位后, 数据时钟的相位类型为类型 1。

备注 f_{PRS} : 外围硬件时钟频率

(3) 端口模式寄存器 1 (PM1)

该寄存器按位设置端口 1 为输入 / 输出。

当使用 P10 / $\overline{\text{SCK10}}$ 作为串行接口的时钟输出引脚时，将 PM10 清 0，并设置 P10 的输出锁存器为 1。

当使用 P12 / SO10 作为串行接口的数据输出引脚时，将 PM12 和 P12 的输出锁存器清 0。

当使用 P10 / $\overline{\text{SCK10}}$ / TxD0 作为串行接口的时钟输入引脚并且使用 P11 / SI10 / RxD0 作为数据输入引脚时，设置 PM10 和 PM11 为 1。这时，P10 和 P11 的输出锁存器可能是 0 或 1。

PM1 可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生设置该寄存器为 FFH。

图 14-4. 端口模式寄存器 1 (PM1) 的格式

地址 FF21H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n 引脚输入/输出模式选择 (n = 0 到 7)
0	输出模式 (输出缓冲器开)
1	输入模式 (输出缓冲器关)

14.4 串行接口CSI10 的操作

串行接口 CSI10 可以用于以下两种模式。

- 操作停止模式
- 3 线串行输入 / 输出模式

14.4.1 操作停止模式

在这种模式下，不执行串行通信。因此，可以降低功耗。此外，在这种模式下，P10 / $\overline{\text{SCK10}}$ / TxD0、P11 / SI10 / RxD0 和 P12 / SO10 引脚可以用作普通输入 / 输出端口引脚。

(1) 使用的寄存器

操作停止模式由串行操作模式寄存器 10 (CSIM10) 来设置。

要设置操作停止模式，将 CSIM10 的第 7 位 (CSIE10) 清 0。

(a) 串行操作模式寄存器 10 (CSIM10)

CSIM10 可以由 1 位或 8 位存储器操作指令来设置。

复位信号的产生设置 CSIM10 为 00H。

地址: FF80H 复位后: 00H R / W

符号	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10
	CSIE10	3 线串行输入 / 输出模式下的操作控制						
	0	禁止操作 ^{注1} 并且异步复位内部电路 ^{注2} 。						

- 注
1. 要使用 P10 / $\overline{\text{SCK10}}$ / TxD0 和 P12 / SO10 作为通用端口，设置 CSIM10 为默认状态 (00H)。
 2. 复位 CSIM10 的第 0 位 (CSOT10) 和串行输入 / 输出移位寄存器 10 (SIO10)。

14.4.2 3 线串行输入 / 输出模式

3 线串行输入 / 输出模式用于连接带时钟驱动串行接口的外围芯片和显示控制器。

该模式中使用三根线来进行通信：串行时钟线 ($\overline{\text{SCK10}}$)、串行输出 (SO10) 和串行输入 (SI10) 线。

(1) 使用的寄存器

- 串行操作模式寄存器 10 (CSIM10)
- 串行时钟选择寄存器 10 (CSIC10)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

在 3 线串行输入 / 输出模式下设置操作的基本过程如下所示。

- <1> 设置 CSIC10 寄存器（见 图 14-3）。
- <2> 设置 CSIM10 寄存器的第 4 和 6 位（DIR10 和 TRMD10）（见 图 14-2）。
- <3> 设置 CSIM10 寄存器的第 7 位（CSIE10）为 1。→ 允许发送 / 接收。
- <4> 向发送缓冲寄存器 10（SOTB10）写入数据。→ 开始发送 / 接收数据。
从串行输入 / 输出移位寄存器 10（SIO10）读取数据。→ 开始接收数据。

注意事项 当设置端口模式寄存器和端口寄存器时，要考虑与通信另一端的关系。

寄存器设置和引脚之间的关系如下所示。

表 14-2. 寄存器设置和引脚之间的关系

CSIE10	TRMD10	PM11	P11	PM12	P12	PM10	P10	CSI10 操作	引脚功能		
									SI10 / RxD0 / P11	SO10 / P12	SCK10 / TxD0 / P10
0	0	× ^{注1}	× ^{注1}	× ^{注1}	× ^{注1}	× ^{注1}	× ^{注1}	停止	RxD0 / P11	P12 ^{注2}	TxD0 / P10 ^{注3}
1	0	1	×	× ^{注1}	× ^{注1}	1	×	从接收 ^{注4}	SI10	P12 ^{注2}	SCK10 (输入) ^{注4}
1	1	× ^{注1}	× ^{注1}	0	0	1	×	从发送 ^{注4}	RxD0 / P11	SO10	SCK10 (输入) ^{注4}
1	1	1	×	0	0	1	×	从发送 / 接收 ^{注4}	SI10	SO10	SCK10 (输入) ^{注4}
1	0	1	×	× ^{注1}	× ^{注1}	0	1	主接收	SI10	P12 ^{注2}	SCK10 (输出)
1	1	× ^{注1}	× ^{注1}	0	0	0	1	主发送	RxD0 / P11	SO10	SCK10 (输出)
1	1	1	×	0	0	0	1	主发送 / 接收	SI10	SO10	SCK10 (输出)

- 注**
1. 可以设置为端口功能。
 2. 要使用 P12/SO10 作为通用端口，设置串行时钟选择寄存器 10（CSIC10）为默认状态值（00H）。
 3. 要使用 P10 / SCK10 / TxD0 作为端口引脚，将 CKP10 清 0。
 4. 要使用从模式，设置 CKS102、CKS101 和 CKS100 分别为 1、1、1。

备注

×: 不必考虑

CSIE10: 串行操作模式寄存器 10（CSIM10）的第 7 位

TRMD10: CSIM10 的第 6 位

CKP10: 串行时钟选择寄存器 10（CSIC10）的第 4 位

CKS102, CKS101, CKS100: CSIC10 的第 2 至 0 位

PM1×: 端口模式寄存器

P1×: 端口输出锁存器

(2) 通信操作

在 3 线串行输入 / 输出模式下，以 8 位为单元发送或接收数据。按照串行时钟同步发送或接收数据的每个位。

如果串行操作模式寄存器 10 (CSIM10) 的第 6 位 (TRMD10) 为 1，可以发送或接收数据。当数据写入发送缓冲寄存器 10 (SOTB10) 时，启动发送 / 接收。此外，当串行操作模式寄存器 10 (CSIM10) 的第 6 位 (TRMD10) 为 0 时，可以接收数据。

当从串行输入 / 输出移位寄存器 10 (SIO10) 读取数据时，启动接收。

启动通信后，CSIM10 的第 0 位 (CSOT10) 置 1。当 8 位数据的通信完成时，设置通信完成中断请求标志 (CSIIF10)，并且将 CSOT10 清 0。然后，允许下一次通信。

注意事项 当 CSOT10 = 1 (串行通信期间) 时，不要访问控制寄存器和数据寄存器。

图 14-5. 3 线串行输入 / 输出模式的时序 (1 / 2)

(a) 发送 / 接收时序 (类型 1: TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 0)

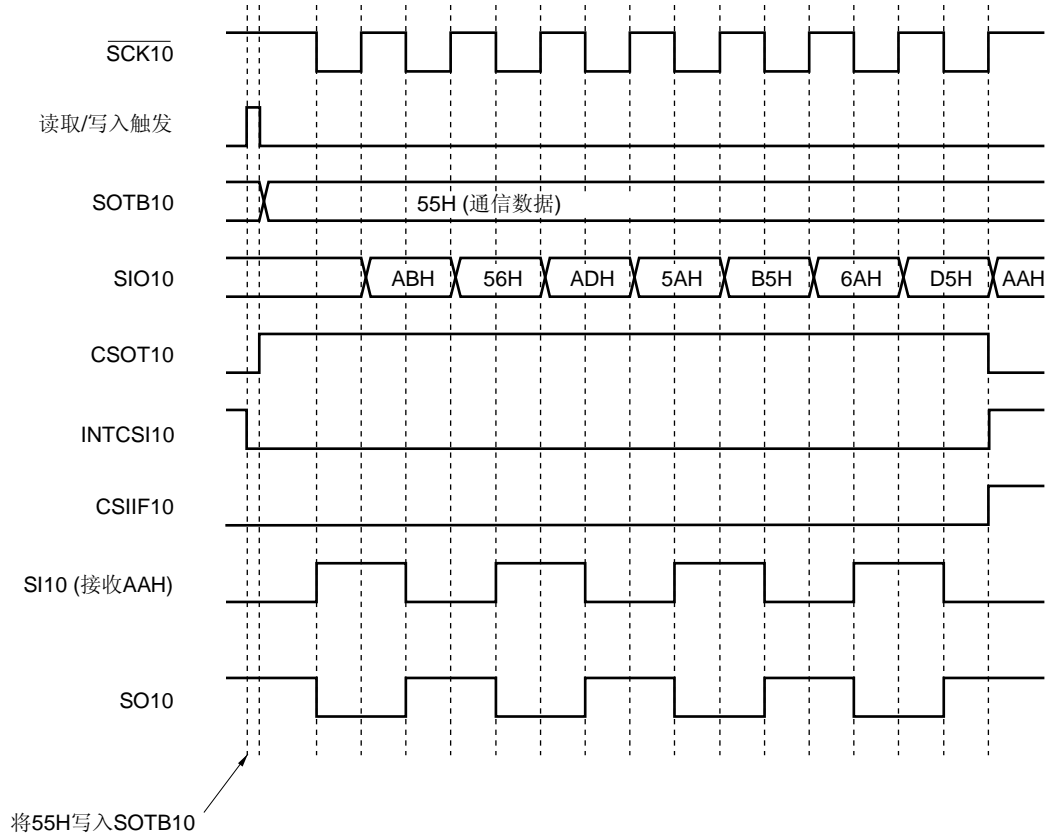


图 14-5. 3 线串行输入 / 输出模式的时序 (2 / 2)

(b) 发送 / 接收时序 (类型 2: TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 1)

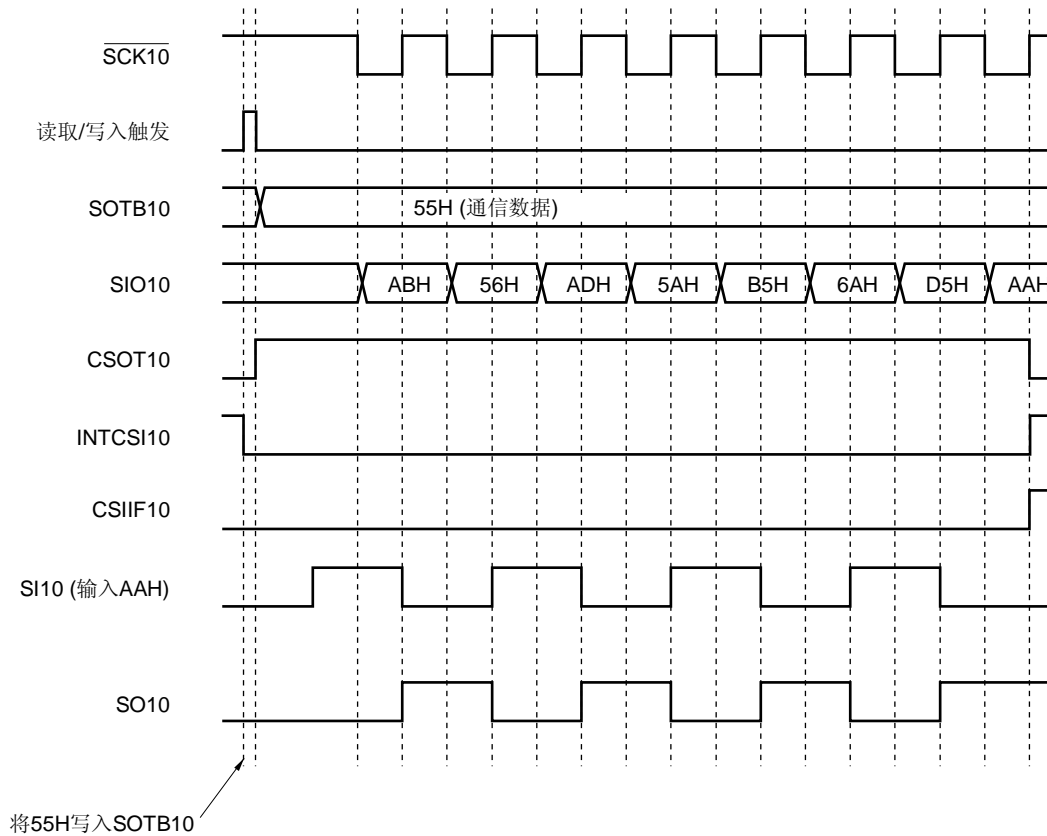
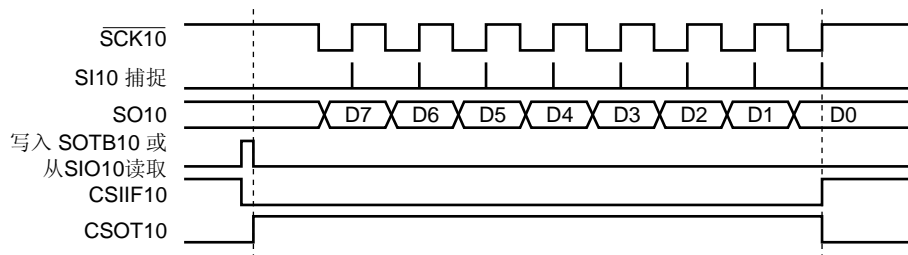
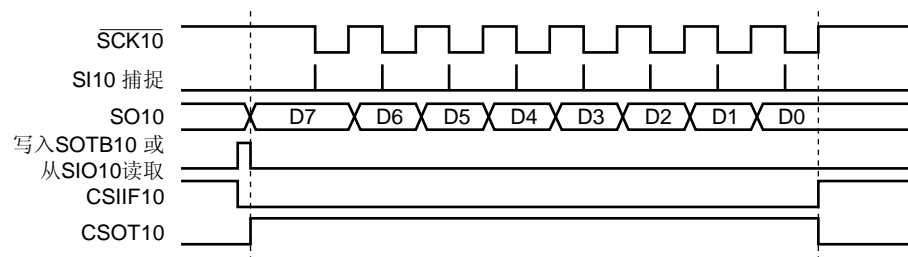


图 14-6. 时钟 / 数据相位的时序

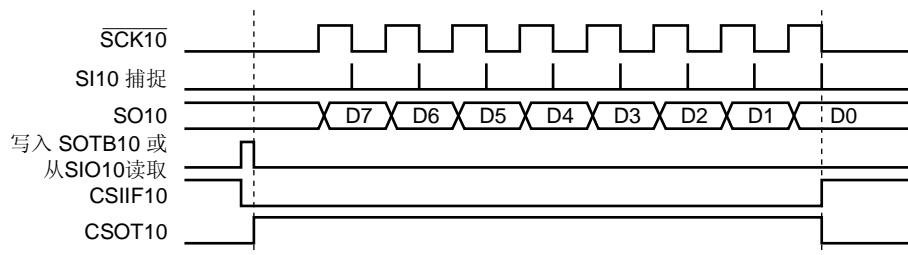
(a) 类型 1: CKP10 = 0, DAP10 = 0, DIR10 = 0



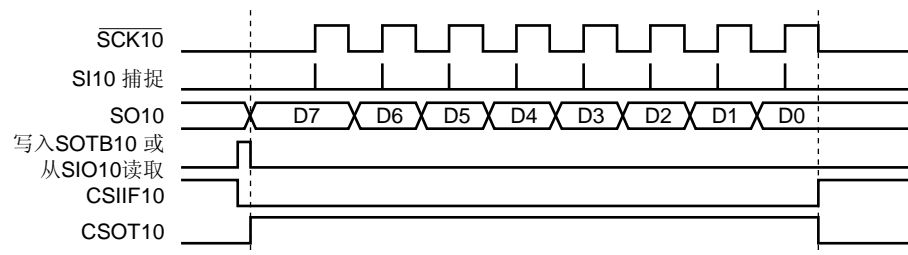
(b) 类型 2: CKP10 = 0, DAP10 = 1, DIR10 = 0



(c) 类型 3: CKP10 = 1, DAP10 = 0, DIR10 = 0



(d) 类型 4: CKP10 = 1, DAP10 = 1, DIR10 = 0

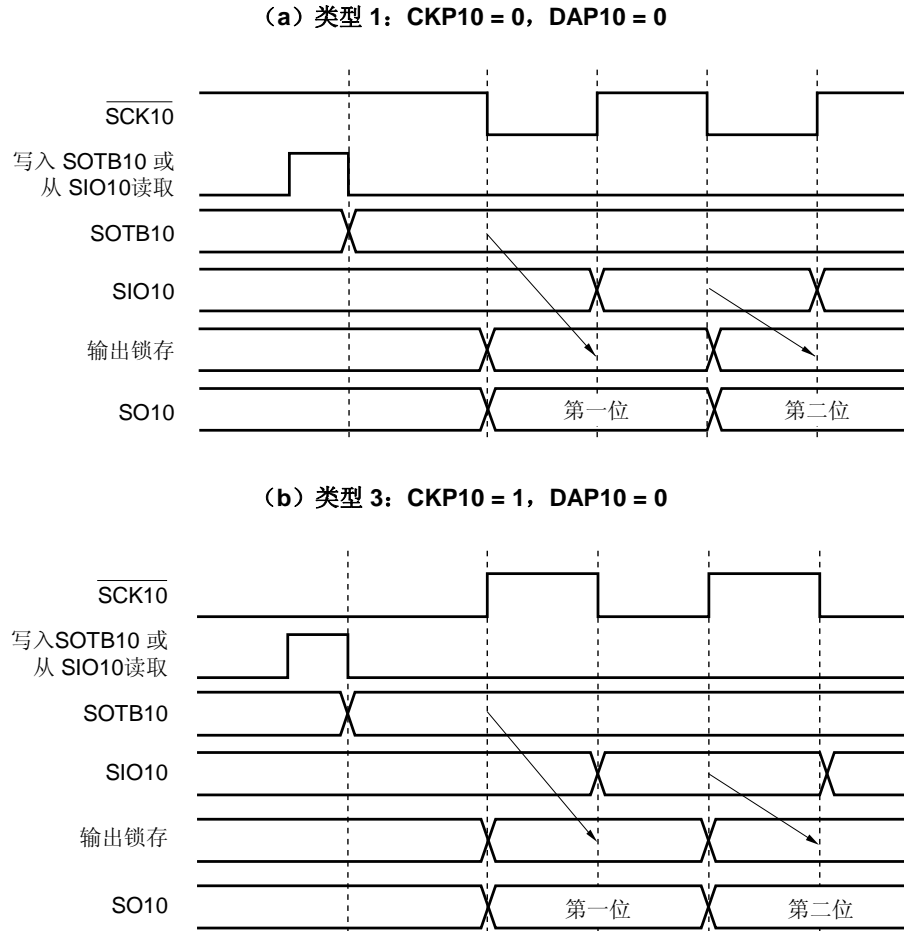


备注 上图所示为采用 MSB 数据发送方式的通信操作。

(3) 输出到 SO10 引脚的时序 (第一位)

当通信开始时, 发送缓冲寄存器 10 (SOTB10) 的值从 SO10 引脚输出。此时第一位的输出操作如下所示。

图 14-7. 第一位的输出操作 (1 / 2)

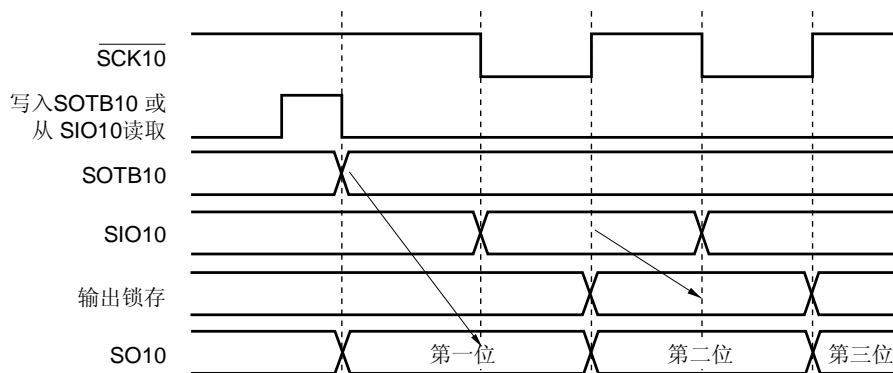


在 $\overline{\text{SCK10}}$ 的下降沿 (或上升沿) 直接由 SOTB10 寄存器将第一位锁存到输出锁存器, 并且通过输出选择器从 SO10 引脚输出。然后, 在 $\overline{\text{SCK10}}$ 的下一个上升沿 (或下降沿), SOTB10 寄存器的值传送到 SIO10 寄存器, 并且移位第一位。同时, 通过 SIO10 引脚接收数据的第一位被保存到 SIO10 寄存器。

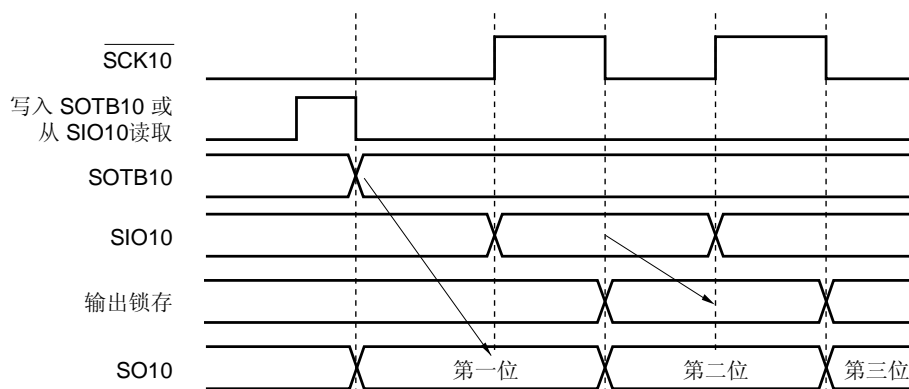
第二位以及后面的位在 $\overline{\text{SCK10}}$ 的后面的下降沿 (或上升沿) 由 SIO10 寄存器锁存到输出锁存器, 并且从 SO10 引脚输出数据。

图 14-7. 第一位的输出操作 (2 / 2)

(c) 类型 2: CKP10 = 0, DAP10 = 1



(d) 类型 4: CKP10 = 1, DAP10 = 1



在 SOTB10 寄存器写信号或 SIO10 寄存器读信号的下降沿直接由 SOTB10 寄存器锁存第一位，并且通过输出选择器从 SO10 引脚输出。然后，在 SCK10 的下一个下降沿（或上升沿），SOTB10 寄存器的值被传送到 SIO10 寄存器，并且移位第一位。同时，通过 SIO10 引脚接收的第一位数据被保存到 SIO10 寄存器。第二位以及后面的位在 SCK10 的后面的上升沿（或下降沿）由 SIO10 寄存器锁存到输出锁存器，并且从 SO10 引脚输出数据。

(4) SO10 引脚的输出值 (最后一位)

通信完成后, SO10 引脚保持最后一位的输出值。

图 14-8. SO10 引脚的输出值 (最后一位) (1 / 2)

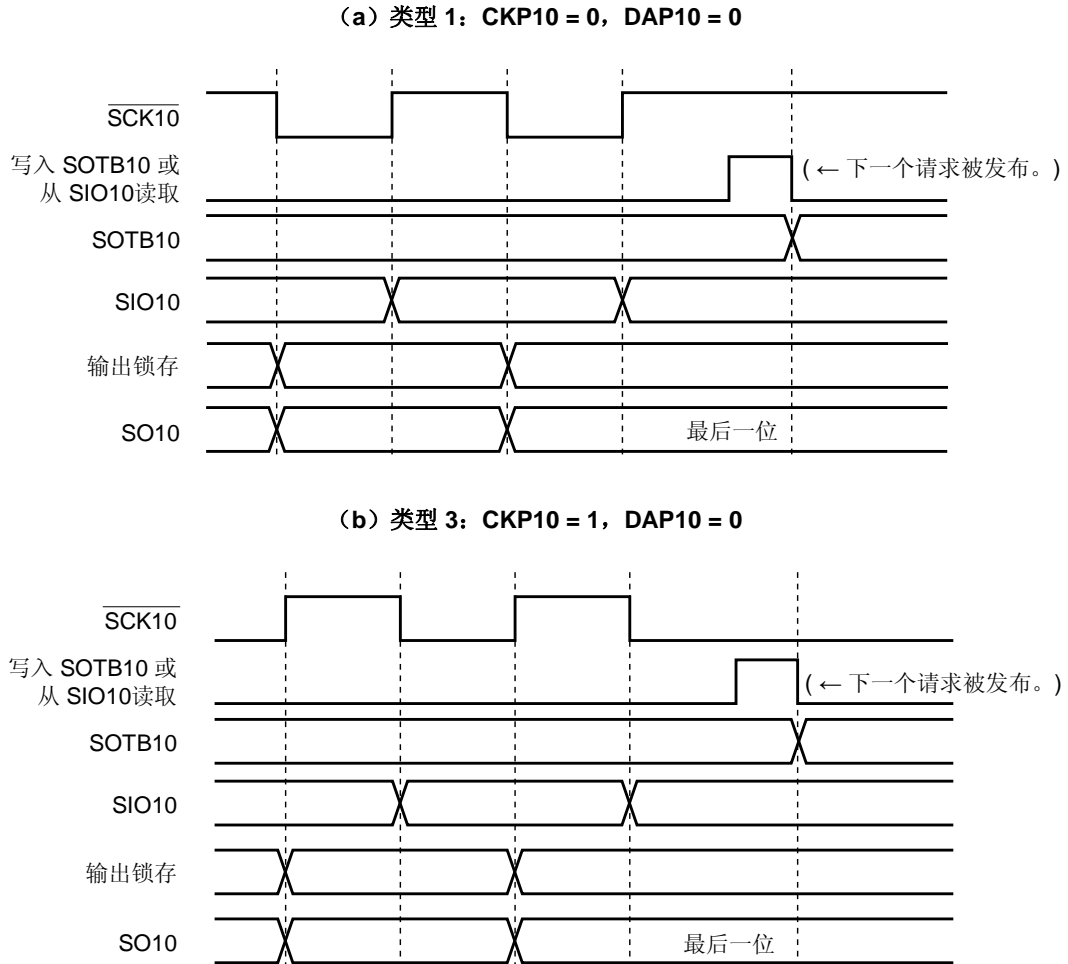
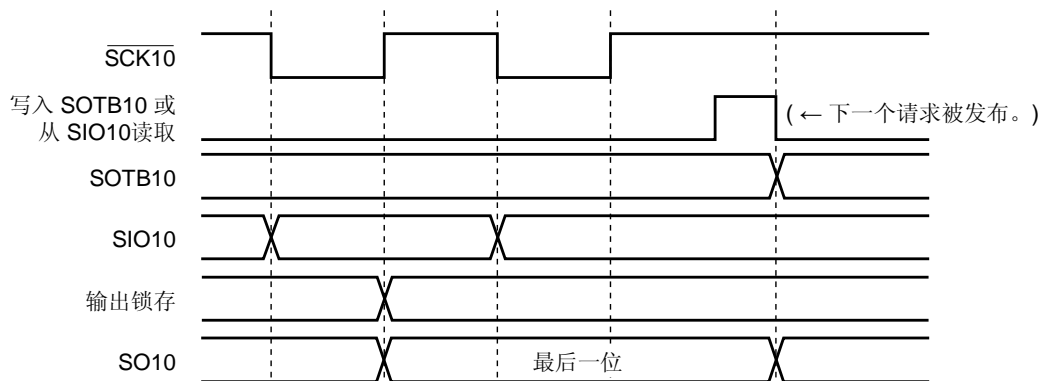
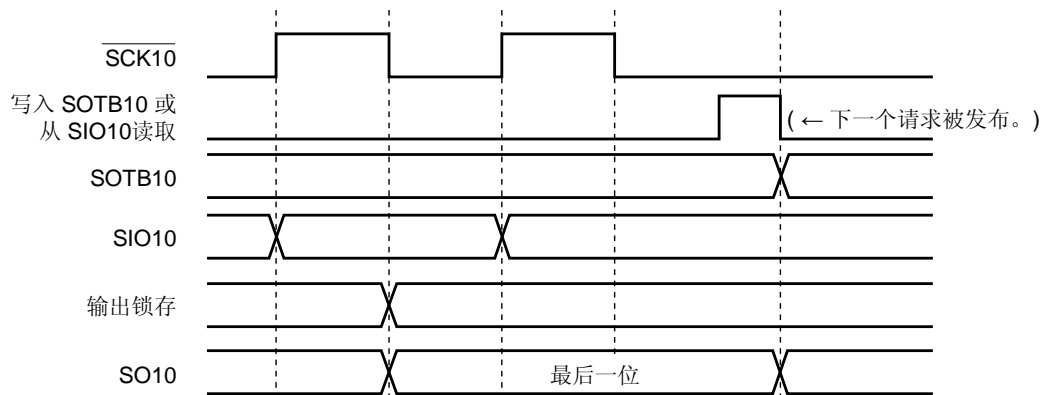


图 14-8. SO10 引脚的输出值（最后一位）（2 / 2）

(c) 类型 2: CKP10 = 0, DAP10 = 1



(d) 类型 4: CKP10 = 1, DAP10 = 1



(5) SO10 输出 (见图 14-1)

根据 CSIE10, TRMD10, DAP10 和 DIR10 的设置值, SO10 的输出状态如下所示。

表 14-3. SO10 输出状态

CSIE10	TRMD10	DAP10	DIR10	SO10 输出 ^{注1}
CSIE10 = 0 ^{注2}	TRMD10 = 0 ^{注2,3}	-	-	低电平输出 ^{注2}
		DAP10 = 0	-	低电平输出
	TRMD10 = 1	DAP10 = 1	DIR10 = 0	SOTB10 的位 7 的值
			DIR10 = 1	SOTB10 的位 0 的值
CSIE10 = 1	TRMD10 = 0 ^{注3}	-	-	低电平输出
	TRMD10 = 1	-	-	发送数据 ^{注4}

- 注
1. SO10 / P12 引脚的实际输出由 PM12 和 P12 确定, SO10 的输出也是如此。
 2. 复位后的状态。
 3. 要使用 SO10/P12 引脚作为通用端口, 设置串行时钟选择寄存器 10 (CSIC10) 为默认状态的值 (00H)。
 4. 发送完成后, SO10 引脚保持最后发送的数据位的输出值。

注意事项 如果对 CSIE10、TRMD10、DAP10 和 DIR10 进行写操作, 会改变 SO10 的输出值。

15.1 串行接口IIC0 的功能

串行接口 IIC0 具有以下两种模式。

(1) 操作停止模式

该模式在没有执行串行传输时使用。以降低功耗。

(2) I²C 总线模式（支持多个主设备）

该模式通过两根线执行多个设备的 8 位数据传输：串行时钟线（SCL0）和串行数据总线（SDA0）。

该模式符合 I²C 总线格式，且主设备可以通过串行数据总线为从设备产生“起始条件”、“地址”、“传输方向指定”、“数据”和“停止条件”。从设备通过硬件自动检测接收的状态和数据。该功能可以简化控制 I²C 总线的应用程序部分。

由于 SCL0 和 SDA0 引脚是漏极开路输出，IIC0 总线的串行时钟线和串行数据总线需要上拉电阻。

图 15-1 显示了串行接口 IIC0 的框图。

图 15-1. 串行接口 IIC0 的框图

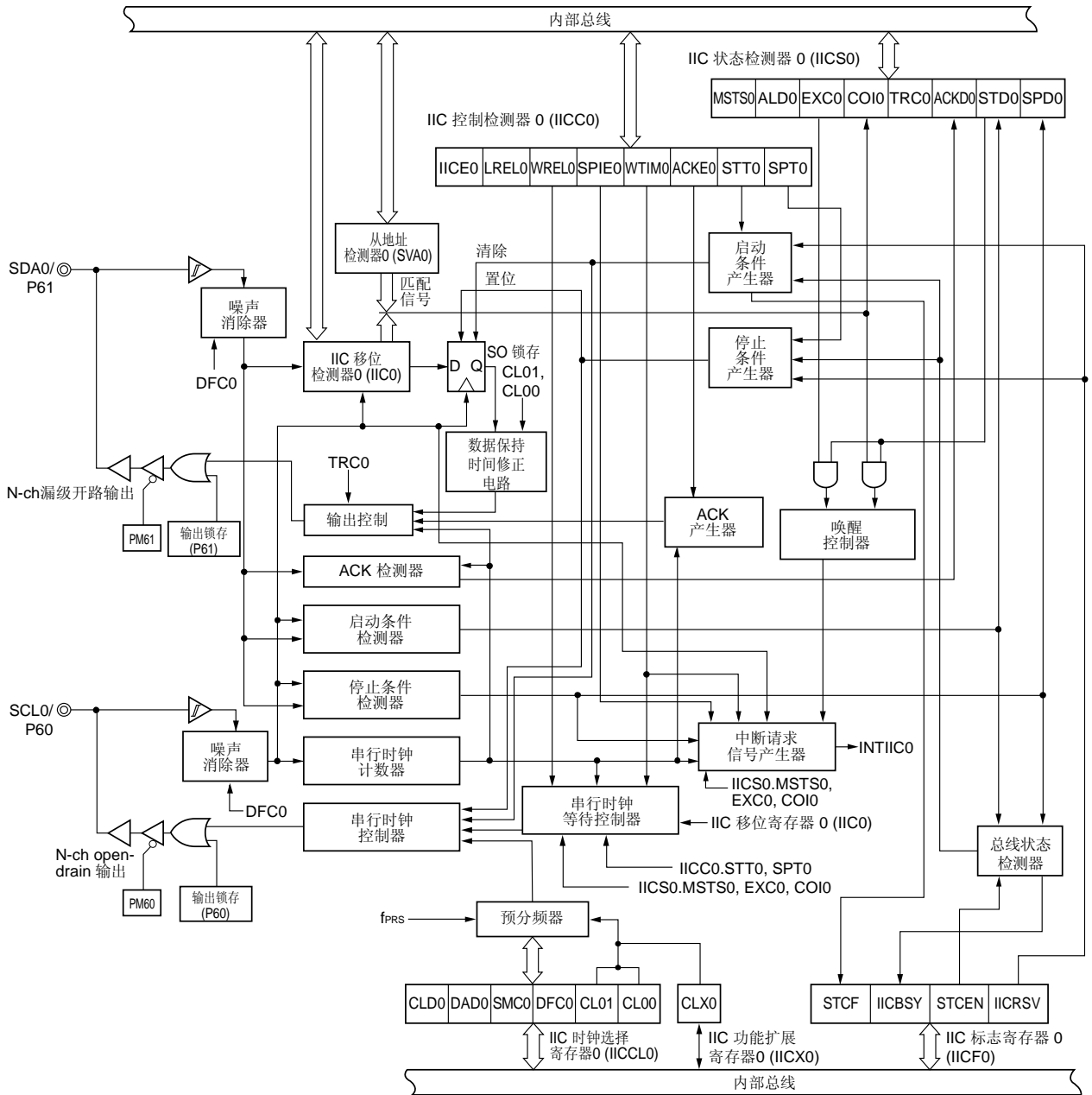
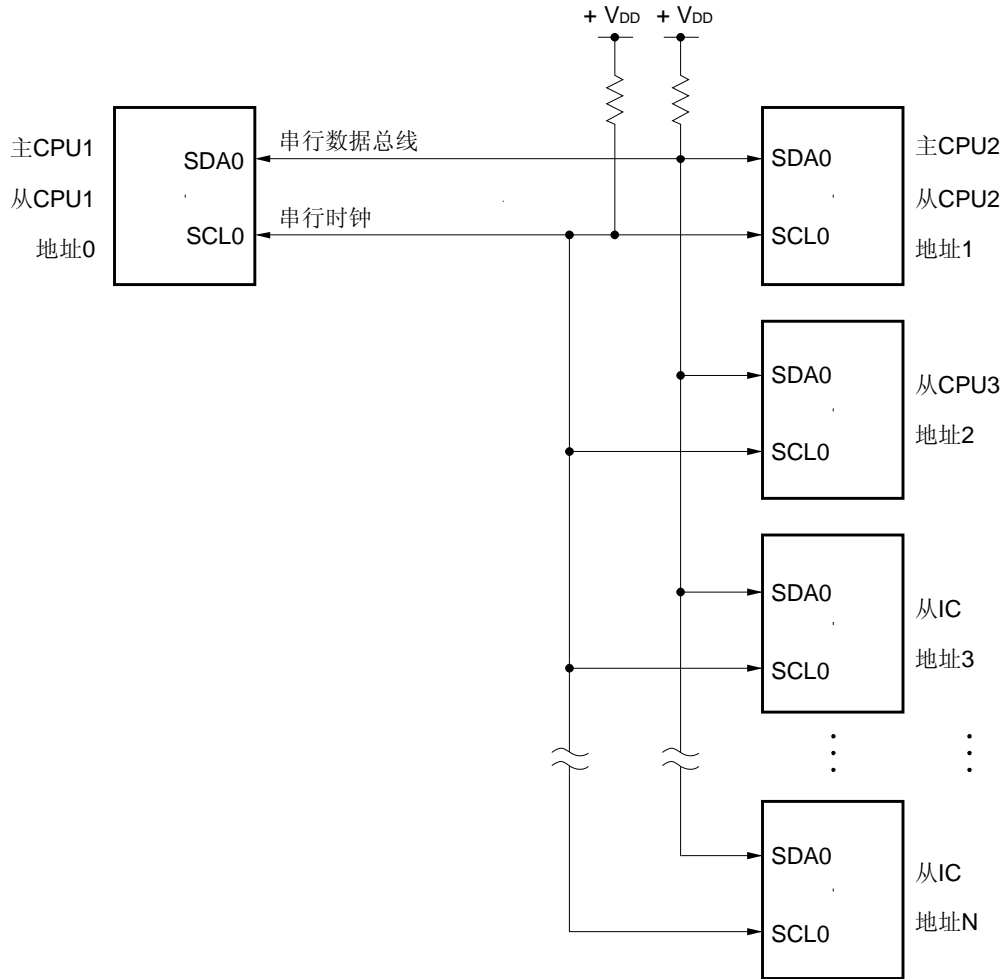


图 15-2 显示了串行总线配置示例

图 15-2. 使用 I²C 总线的串行总线配置示例



15.2 串行接口 IIC0 的配置

串行接口 IIC0 包括以下硬件。

表 15-1. 串行接口 IIC0 的配置

项目	配置
寄存器	IIC 移位寄存器 0 (IIC0) 从设备地址寄存器 0 (SVA0)
控制寄存器	IIC 控制寄存器 (IICC0) IIC 状态寄存器 0 (IICS0) IIC 标志寄存器 0 (IICF0) IIC 时钟选择寄存器 0 (IICCL0) IIC 功能扩展寄存器 0 (IICX0) 端口模式寄存器 6 (PM6) 端口寄存器 6 (P6)

(1) IIC 移位寄存器 0 (IIC0)

IIC0 用于通过串行时钟将 8 位串行数据转换为 8 位并行数据，反之亦然。IIC0 既可以用于发送也可以用于接收。

实际的传输和接收操作可以通过对 IIC0 进行写操作和读操作来控制。

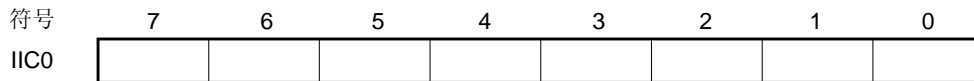
在等待周期中，通过将数据写入 IIC0 来取消等待状态和开始数据传输。

IIC0 通过 8 位存储器操作指令来设置。

复位信号产生将 IIC0 设置为 00H。

图 15-3. IIC 移位寄存器 0 (IIC0) 的格式

地址: FFA5H 复位后: 00H R/W



注意事项 1. 在数据传输期间不要将数据写入 IIC0。

2. 只能在等待周期对 IIC0 进行写或读。禁止在非等待状态的其它通信状态中访问 IIC0。然而，当设备用作主设备时，IIC0 则只能在通信触发位 (STT0) 设置为 1 后进行一次写操作。

(2) 从设备地址寄存器 0 (SVA0)

该寄存器用于在从模式时保存本地地址。

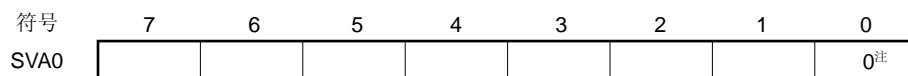
SVA0 通过 8 位存储器操作指令来设置。

然而，当 STD0=1 时（当检测到起始条件时），禁止对该寄存器进行重写操作。

复位信号产生将 SVA0 清为 00H。

图 15-4. 从属地址寄存器 0 (SVA0) 的格式

地址: FFA7H 复位后: 00H R/W



注 第 0 位固定为 0。

(3) SO 锁存

SO 锁存用于保持 SDA0 引脚的输出电平。

(4) 唤醒控制器

当由该寄存器接收的地址与在从设备地址寄存器 0 (SVA0) 中设置的地址值相匹配或者当接收到扩展代码时，该电路将产生中断请求 (INTIIC0)。

(5) 预分频器

用于选择所使用的采样时钟。

(6) 串行时钟计数器

该计数器对在发送 / 接收操作过程中输出或输入的串行时钟进行计数，它也用于检验 8 位数据是否已发送或接收。

(7) 中断请求信号产生器

该电路控制了中断请求信号 (INTIIC0) 的产生。

一个 I²C 中断请求可以通过以下两种触发来产生。

- 串行时钟的第 8 或第 9 个时钟的下降沿 (由 WTIM0 位来设置)
- 当检测到停止条件时所产生的中断请求 (由 SPIE0 位来设置)

备注 WTIM0 位: IIC 控制寄存器 0 (IICC0) 的第 3 位
SPIE0 位: IIC 控制寄存器 0 (IICC0) 的第 4 位

(8) 串行时钟控制器

在主模式中，该电路通过 SCL0 引脚从采样时钟产生时钟输出。

(9) 串行时钟等待控制器

该电路控制等待时间。

(10) ACK 产生器，停止条件检测器，起始条件检测器以及 ACK 检测器

这些电路产生并检测每种状态。

(11) 数据保持时间校正电路

该电路根据串行时钟下降沿产生数据保持时间。

(12) 起始条件产生器

当 STT0 位被设置为 1 时，该电路产生一个起始条件信号。

然而，在通信保留禁止状态中 (IICRSV 位=1)，当总线没有被释放 (IICBSY 位=1) 时，起始条件请求将被忽略，且 STCF 位被设置为 1。

(13) 停止条件产生器

当 SPT0 位设置为 1 时，该电路将产生一个停止条件。

(14) 总线状态检测器

该电路检测是否通过检测起始条件和停止条件释放总线。

然而，由于总线状态不能在以下操作后立即被检测，因此初始状态将通过 **STCEN** 位来设置。

备注	STT0 位：	IIC 控制寄存器 0 (IICC0) 的第 1 位
	SPT0 位：	IIC 控制寄存器 0 (IICC0) 的第 0 位
	IICRSV 位：	IIC 标志寄存器 0 (IICF0) 的第 0 位
	IICBSY 位：	IIC 标志寄存器 0 (IICF0) 的第 6 位
	STCF 位：	IIC 标志寄存器 0 (IICF0) 的第 7 位
	STCEN 位：	IIC 标志寄存器 0 (IICF0) 的第 1 位

15.3 控制串行接口 IIC0 的寄存器

串行接口 IIC0 通过以下七个寄存器来控制。

- IIC 控制寄存器 0 (IICC0)
- IIC 标志寄存器 0 (IICF0)
- IIC 状态寄存器 0 (IICS0)
- IIC 时钟选择寄存器 0 (IICCL0)
- IIC 功能扩展寄存器 0 (IICX0)
- 端口模式寄存器 6 (PM6)
- 端口寄存器 6 (P6)

(1) IIC 控制寄存器 0 (IICC0)

该寄存器用于允许 / 停止 I²C 的操作，设置等待时间并设置其它 I²C 的操作。

IICC0 通过 1 位或 8 位存储器操作指令来设置。然而，当 IICE0 位=0 时或在等待周期时则将设置 SPIE0、WTIM0 以及 ACKE0 位。当 IICE0 位从“0”设置为“1”时，这些位可以同时进行设置。

复位信号产生将 IICC0 设置为 00H。

图 15-5. IIC 控制寄存器 0 (IICC0) 的格式 (1 / 4)

地址: FFA6H 复位后: 00H R / W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICC0	IICE0	LRELO	WRELO	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	允许 IIC 操作
0	停止操作。复位 IIC 状态寄存器 0 (IICS0) ^{注1} 。停止内部操作。
1	允许操作。
当 SCL0 和 SDA0 线处于高电平时，确保置位 (1)。	
清除条件 (IICE0=0)	置位条件 (IICE0 = 1)
<ul style="list-style-type: none"> 通过指令清除 复位 	<ul style="list-style-type: none"> 通过指令来设置

LRELO ^{注2}	退出通信
0	正常操作
1	退出当前通信并进入待机模式。执行后该设置自动清 0。 它在已经接收到本地无关扩展代码的情况下使用。 SCL0 和 SDA0 线被设置为高阻抗。 以下 IIC 控制寄存器 0 (IICC0) 以及 IIC 状态寄存器 0 (IICS0) 的标志被清 0。 • STT0 • SPT0 • MST0 • EXC0 • COI0 • TRC0 • ACKD0 • STD0
退出通信后的待机模式将保持有效，直到遇到下一个通信起始条件时。 <ul style="list-style-type: none"> 在检测到停止条件后，主模式模式下重启。 在起始条件之后，产生地址匹配或接收到扩展代码。 	
清除条件 (LRELO = 0)	置位条件 (LRELO = 1)
<ul style="list-style-type: none"> 执行后自动被清除 复位 	<ul style="list-style-type: none"> 通过指令来设置

WRELO ^{注2}	等待取消
0	不取消等待
1	取消等待。在等待被取消后该设置被自动清除。
在发送状态 (TRC0=1) 下，在第九个时钟脉冲的等待周期中 WRELO 被设置 (取消等待) 时，SDA0 线将进入高阻状态 (TRC0=0)。	
清除条件 (WRELO = 0)	置位条件 (WRELO = 1)
<ul style="list-style-type: none"> 执行后自动清除 复位 	<ul style="list-style-type: none"> 通过指令来设置

- 注 1. IICS0 寄存器、IICF0 寄存器的 STCF0 和 IICBSY 位以及 IICCL0 寄存器的 CLD0 和 DAD0 位被复位。
 2. 当 IICE0=0 时，该标志的信号无效。

注意事项 在 I²C 允许操作后 (IICE0=1)，当 SCL0 线处于高电平且 SDA0 线处于低电平时，立即检测到起始条件。在允许 I²C 操作 (IICE0=1) 后，立即使用 1 位存储器操作指令设置 LRELO (1)。

图 15-5. IIC 控制寄存器 0 (IICC0) 的格式 (2 / 4)

SPIE0 ^{注1}	当停止条件被检测到时允许 / 禁止中断请求的产生	
0	禁止	
1	允许	
清除条件 (SPIE0 = 0)		置位条件 (SPIE0 = 1)
<ul style="list-style-type: none"> • 通过指令清除 • 复位 		<ul style="list-style-type: none"> • 通过指令来设置

WTIMO ^{注1}	等待和中断请求产生的控制	
0	在第八个时钟的下降沿产生中断请求。 主模式：在输出八个时钟后，时钟输出被设置为低电平并且设置等待。 从模式：在输入八个时钟后，时钟设置为低电平且为主设备设置等待。	
1	在第九个时钟的下降沿产生中断请求。 主模式：在输出九个时钟后，时钟输出被设置为低电平并且设置等待。 从模式：在输入九个时钟后，时钟被设置为低电平且为主设备设置等待。	
地址传送期间，在第九个时钟的下降沿产生中断，与该位的设置无关。当完成地址传输时，该位的设置将有效。当处于主模式时，在地址传输期间将在第九个时钟的下降沿插入一个等待。对于已经接收了一个本地地址的从设备来说，在发出响应 (ACK) 后将在第九个时钟的下降沿插入一个等待。然而，当从设备接收了一个扩展代码时，则将在第八个时钟的下降沿插入一个等待。		
清除条件 (WTIMO = 0)		置位条件 (WTIMO = 1)
<ul style="list-style-type: none"> • 通过指令清除 • 复位 		<ul style="list-style-type: none"> • 通过指令来设置

ACKE0 ^{注1, 2}	响应控制	
0	禁止响应。	
1	允许响应。在第九个时钟期间，SDA0 线被设置为低电平。	
清除条件 (ACKE0 = 0)		置位条件 (ACKE0 = 1)
<ul style="list-style-type: none"> • 通过指令清除 • 复位 		<ul style="list-style-type: none"> • 通过指令来设置

- 注
1. 当 IICE0=0 时，该标志的信号是无效的。
 2. 在地址传输期间，如果代码不是扩展代码，则该设置值无效。
当设备用作从设备且地址匹配时，不管设置何值都会产生响应。

图 15-5. IIC 控制寄存器 0 (IICC0) 的格式 (3 / 4)

STT0 ^注	起始条件触发	
0	不产生起始条件。	
1	当总线被释放时 (在 STOP 模式中): 产生一个起始条件 (对于以主设备开始来说)。当 SCL0 线是高电平时, SDA0 线从高电平转换到低电平产生起始条件。接着, 在额定的时间后, SCL0 将变为低电平 (等待状态)。 当第三方通信时: <ul style="list-style-type: none"> • 当允许通信保留功能时 (IICRSV=0) 作为起始条件预约标志。当设置为 1 时, 在总线被释放后将自动产生一个起始条件。 • 当禁止通信保留功能时 (IICRSV=1) STCF 被设置为 1 且设置 (1) 到 STT0 的信息被清除。不产生起始条件。 在等待状态中 (当设备为主设备时): 在释放等待后产生一个复位条件。	
关于设置时间的注意事项 <ul style="list-style-type: none"> • 对于主接收: 传输期间不能被设置为 1。当 ACKE0 被清 0 且从设备被通知进行最后接收时, 只有在等待周期可以设置为 1。 • 对于主传输: 响应期间起始条件不能被正常产生。在输出第九个时钟后的等待时间内被设置为 1。 • 不能与 SPT0 同时设置为 1。 • 禁止在 STT0 清 0 前将其设置为 1, 并且再对其进行设置。 		
清除条件 (STT0 = 0)		置位条件 (STT0 = 1)
<ul style="list-style-type: none"> • 禁止在通信保留时通过将 SST0 设置为 1 来清除。 • 通过在仲裁中的丢失清除 • 在产生起始条件后通过主设备进行清除 • 通过 LREL0 = 1 (退出通信) 进行清除 • 当 IICE0=0 (停止操作) 时 • 复位 		<ul style="list-style-type: none"> • 通过指令来设置

注 当 IICE0=0 时, 该标志的信号是无效的。

- 备注
1. 在数据设置后, 当第 1 位 (STT0) 被读取时, 它将变为 0。
 2. IICRSV: 标志寄存器 (IICF0) 的第 0 位
 STCF: 标志寄存器 (IICF0) 的第 7 位

图 15-5. IIC 控制寄存器 0 (IICC0) 的格式 (4 / 4)

SPT0	停止条件触发	
0	不产生停止条件。	
1	产生停止条件（主设备传输终结）。 在 SDA0 进入低电平后，将 SCL0 线设置为高电平或等待其进入高电平。接着，在额定的时间后，SDA0 线将从低电平变成高电平，产生停止条件。	
<p>关于设置时间的注意事项</p> <ul style="list-style-type: none"> 对于主接收：在传输期间不能被设置为 1。 当 ACKE0 被清 0 且从设备通知结束接收时，只有在等待周期可以被设置为 1。 对于主发送：在响应期间，不能正常产生停止条件。因此，在输出第九个时钟后的等待时间内对其进行设置。 不能与 STT0 同时设置为 1。 只有在主模式时 SPT0 可以被设置为 1^注。 当 WTIMO 被清 0 时，如果 SPT0 在输出八个时钟后的等待时间内被设置为 1，则需注意在第九个时钟的高电平时间内将会产生一个停止条件。WTIMO 应该在输出八个时钟后的等待时间内从 0 变为 1，而 SPT0 则应该在输出第九个时钟后的等待时间内设置为 1。 禁止在 SPT0 清 0 前重复设置为 1。 		
清除条件 (SPT0 = 0)		置位条件 (SPT0 = 1)
<ul style="list-style-type: none"> 由仲裁失败清除 在检测到停止条件后自动清除 通过 LREL0 = 1 (退出通信) 进行清除 当 IICE0=0 (停止操作) 时 复位 		<ul style="list-style-type: none"> 通过指令来设置

注 只在主模式中将 SPT0 设置为 1。然而，在转换到允许操作状态后检测到第一个停止条件前必须将 SPT0 置为 1。关于详细信息，参加 15.5.15 其它注意事项。

注意事项 当 IIC 状态寄存器 0 (IICS0) 的第 3 位 (TRC0) 设置为 1 时，在第九个时钟期间 WREL0 设置为 1，取消等待，在这之后，TRC0 被清除且 SDA0 线将被设置为高阻抗。

备注 当第 0 位 (SPT0) 设置后再读取时，它将变为 0。

(2) IIC 状态寄存器 0 (IICS0)

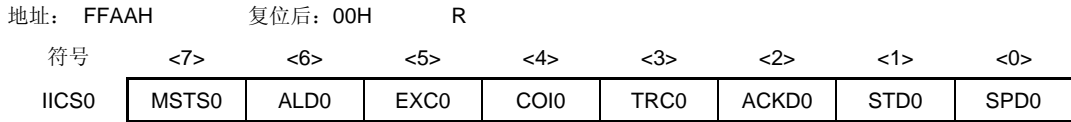
该寄存器显示了 IIC 的状态。

只有当 STT0 = 1 时且在等待时间内，IICS0 会通过 1 位或 8 位存储器操作指令来读取。

复位信号产生将 IICS0 设置为 00H。

注意事项 如果数据从 IICS0 被读取，产生一个等待周期。当外围硬件时钟 (fPRS) 停止时，不要从 IICS0 读取数据。关于详细信息，参见第二十七章 等待注意事项。

图 15-6. IIC 状态寄存器 0 (IICS0) 的格式 (1 / 3)



MSTS0	主设备状态	
0	从设备状态或通信等待状态	
1	主设备通信状态	
清除条件 (MSTS0 = 0)		置位条件 (MSTS0 = 1)
<ul style="list-style-type: none"> 当检测到一个停止条件时 当 ALD0 = 1 (仲裁失败) 时 通过 LREL0 = 1 (退出通信) 进行清除 当 IICE0 从 1 变为 0 (停止操作) 时 复位 		<ul style="list-style-type: none"> 当产生一个起始条件时

ALD0	仲裁失败的检测	
0	该状态表示没有仲裁或仲裁结果是“win”。	
1	该状态显示仲裁结果是“loss”。MSTS0 被清除。	
清除条件 (ALD0 = 0)		置位条件 (ALD0 = 1)
<ul style="list-style-type: none"> 在读取 IICS0 后自动清除^注 当 IICE0 从 1 变为 0 (停止操作) 时 复位 		<ul style="list-style-type: none"> 当仲裁结果是“失败”时

EXC0	扩展代码接收检测	
0	没有接收到扩展代码	
1	接收到扩展代码	
清除条件 (EXC0 = 0)		置位条件 (EXC0 = 1)
<ul style="list-style-type: none"> 当检测到一个起始条件时 当检测到一个停止条件时 通过 LREL0 = 1 (退出通信) 进行清除 当 IICE0 从 1 变为 0 (停止操作) 时 复位 		<ul style="list-style-type: none"> 当接收到的地址数据的高四位是“0000”或“1111” (在第八个时钟的上升沿进行设置) 时。

注 当 1 位存储器操作指令未对 IICS0 进行操作，该寄存器也会被清除。因此，当使用 ALD0 位时，将在其它位的数据之前先读取该位的数据。

备注 LREL0: IIC 控制寄存器 0 (IICC0) 的第 6 位
IICE0: IIC 控制寄存器 0 (IICC0) 的第 7 位

图 15-6. IIC 状态寄存器 0 (IICS0) 的格式 (2 / 3)

COI0	匹配地址的检测	
0	地址不匹配。	
1	地址匹配。	
清除条件 (COI0 = 0)		置位条件 (COI0 = 1)
<ul style="list-style-type: none"> 当检测到一个起始条件时 当检测到一个停止条件时 通过 LRELO = 1 (退出通信) 进行清除 当 IICE0 从 1 变为 0 (停止操作) 时 复位 		<ul style="list-style-type: none"> 当接收到的地址匹配本地地址 (从设备地址寄存器 0 (SVA0)) 时 (在第八个时钟的上升沿设置)。

TRC0	发送 / 接收状态的检测	
0	接收状态 (不同于发送状态)。SDA0 线被设置为高阻抗。	
1	发送状态。允许 SO0 锁存器中的值输出到 SDA0 线 (第一个字节的第九个时钟下降沿的有效开始)。	
清除条件 (TRC0 = 0)		置位条件 (TRC0 = 1)
<p><主设备和从设备></p> <ul style="list-style-type: none"> 当检测到一个停止条件时 通过 LRELO = 1 (退出通信) 进行清除 当 IICE0 从 1 变为 0 (停止操作) 时 通过 WRELO = 1^注 (等待取消) 进行清除 当 ALD0 从 0 变为 1 (仲裁失败) 时 复位 <p><主设备></p> <ul style="list-style-type: none"> 当 “1” 被输出到第一个字节的 LSB (传输方向指定位) <p><从设备></p> <ul style="list-style-type: none"> 当检测到一个起始条件时 当 “0” 被输入到第一个字节的 LSB (传输方向指定位) <p><当不用于通信时></p>		<p><主设备></p> <ul style="list-style-type: none"> 当产生一个起始条件时 当 “0” 被输出到第一个字节的 LSB (传输方向指示位) <p><从设备></p> <ul style="list-style-type: none"> 当 “1” 被输入到第一个字节的 LSB (传输方向指示位)

注 当 IIC 状态寄存器 0 (IICS0) 的第 3 位 (TRC0) 为 1 时, 如果通过在第九个时钟将 IIC 控制寄存器 0 (IICC0) 的第 5 位 (WRELO) 设置为 1 来取消等待状态, 那么 TRC0 将被清零, 且 SDA0 线将进入高阻状态。

备注 LRELO: IIC 控制寄存器 0 (IICC0) 的第 6 位
IICE0: IIC 控制寄存器 0 (IICC0) 的第 7 位

图 15-6. IIC 状态寄存器 0 (IICS0) 的格式 (3 / 3)

ACKD0	响应 ($\overline{\text{ACK}}$) 的检测	
0	没有检测到 $\overline{\text{ACK}}$ 。	
1	检测到 $\overline{\text{ACK}}$ 。	
清除条件 (ACKD0 = 0)		置位条件 (ACKD0 = 1)
<ul style="list-style-type: none"> 当检测到一个停止条件时 在下一个字节的第一个时钟的上升沿 通过 LREL0 = 1 (退出通信) 进行清除 当 IICE0 从 1 变为 0 (停止操作) 时 复位 		<ul style="list-style-type: none"> SDA0 线在 SCL0 的第九个时钟的上升沿被设置为低电平后

STD0	起始条件的检测	
0	没有检测到起始条件。	
1	检测到起始条件。这表明地址传输阶段有效。	
清除条件 (STD0 = 0)		置位条件 (STD0 = 1)
<ul style="list-style-type: none"> 当检测到一个停止条件时 在地址传输后的下一个字节的第一个时钟的上升沿处 通过 LREL0 = 1 (退出通信) 进行清除 当 IICE0 从 1 变为 0 (停止操作) 时 复位 		<ul style="list-style-type: none"> 当检测到一个起始条件时

SPD0	停止条件的检测	
0	没有检测到停止条件。	
1	检测到停止条件。主设备通信结束，释放主线。	
清除条件 (SPD0 = 0)		置位条件 (SPD0 = 1)
<ul style="list-style-type: none"> 在设置该位并检测到起始条件后的地址传输字节的第一个时钟的上升沿处 当 IICE0 从 1 变为 0 (停止操作) 时 复位 		<ul style="list-style-type: none"> 当检测到停止条件时

备注 LREL0: IIC 控制寄存器 0 (IICC0) 的第 6 位
 IICE0: IIC 控制寄存器 0 (IICC0) 的第 7 位

(3) IIC 标志寄存器 0 (IICF0)

该寄存器设置 I²C 的操作模式并显示 I²C 总线的状态。

IICF0 通过 1 位或 8 位存储器操作指令来设置。然而，STCF 和 IICBSY 位为只读。

IICRSV 位可以用于允许 / 禁止通信保留功能 (参见 15.5.14 通信保留)。

STCEN 可以用于设置 IICBSY 位的初始值 (参见 15.5.15 其它注意事项)。

只有当 I²C 的操作禁止 (IIC 控制寄存器 0 (IICC0) 的第 7 位 (IICE0) = 0) 时，才可以对 IICRSV 和 STCEN 进行写操作。当允许操作时，可以读取 IICF0 寄存器。

复位信号产生将 IICF0 设置为 00H。

图 15-7. IIC 标志寄存器 0 (IICF0) 的格式

地址: FFABH 复位后: 00H R/W^注

符号	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	STT0 清除标志
0	产生自动条件
1	起始条件产生不成功: 清除 STT0 标志
清除条件 (STCF = 0)	置位条件 (STCF = 1)
· 由 STT0 = 1 清零 ^L · 当 IICE0 = 0 (操作停止) · 复位	· 起始条件产生不成功且 STT0 通信预约禁止时 (IICRSV = 1).

IICBSY	I ² C 总线状态标志
0	总线释放状态 (当 STCEN = 1 时的通信初始状态)
1	总线通信状态 (当 STCEN = 0 时通信初始状态)
清除条件 (IICBSY = 0)	置位条件 (IICBSY = 1)
· 停止条件检测 · 当 IICE0 = 0 (操作停止) · 复位	· 起始条件检测 · 当 STCEN = 0, IICE0 的设置

STCEN	初始启动允许触发
0	操作允许后 (IICE0 = 1), 检测到起始条件时立即允许开始条件的产生。 ^L
1	操作允许后 (IICE0 = 1), 允许起始条件产生而不需要停止条件检测。
清除条件 (STCEN = 0)	置位条件 (STCEN = 1)
· 起始条件检测 · 复位	· 由指令设置

IICRSV	通信预约功能禁止位
0	允许通信预约
1	禁止通信预约
清除条件 (IICRSV = 0)	置位条件 (IICRSV = 1)
· 由指令清除 · 复位	· 由指令设置

注 第 6 和 7 位为只读。

- 注意事项**
1. 只有当操作停止 (IICE0=0) 时才能对 STCEN 进行写入操作。
 2. 当产生第一个起始条件 (STT0=1) 时, 由于当 STCEN=1 时, 不管实际总线处于何种状态, 总线释放状态 (IICBSY=0) 都能被识别, 因此需确保没有第三方通信, 以避免通信被破坏。
 3. 只有当操作停止 (IICE0=0) 时才能对 IICRSV 进行写入操作。

备注 STT0: IIC 控制寄存器 0 (IICC0) 的第 1 位
IICE0: IIC 控制寄存器 0 (IICC0) 的第 7 位

(4) IIC 时钟选择寄存器 0 (IICCL0)

该寄存器用于为 I²C 总线设置传输时钟。

IICCL0 通过 1 位或 8 位存储器操作指令来设置。然而，CLD0 和 DAD0 位为只读。SMC0、CL01 和 CL00 结合 IIC 功能扩展寄存器 0 (IICX0) 的第 0 位 (CLX0) 来进行设置 (参见 15.3 (6) IIC 传输时钟设置方法)。

当 IIC 控制寄存器 0 (IICC0) 的第 7 位 (IICE0) 为 0 时设置 IICCL0。

复位信号产生将 IICCL0 设置为 00H。

图 15-8. IIC 时钟选择寄存器 0 (IICCL0) 的格式

地址: FFA8H 复位后: 00H R / W^注

符号	7	6	<5>	<4>	<3>	<2>	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	CL01	CL00

CLD0	SCL0 引脚电平的检测 (仅当 IICE0=1 时有效)	
0	在低电平处检测到 SCL0 引脚。	
1	在高电平处检测到 SCL0 引脚。	
清除条件 (CLD0 = 0)		置位条件 (CLD0 = 1)
<ul style="list-style-type: none"> • 当 SCL0 引脚在低电平时 • 当 IICE0=0 (停止操作) 时 • 复位 		<ul style="list-style-type: none"> • 当 SCL0 引脚在高电平时

DAD0	SDA0 引脚电平的检测 (仅当 IICE0=1 时有效)	
0	在低电平处检测到 SDA0 引脚。	
1	在高电平处检测到 SDA0 引脚。	
清除条件 (DAD0 = 0)		置位条件 (DAD0 = 1)
<ul style="list-style-type: none"> • 当 SDA0 引脚在低电平时 • 当 IICE0=0 (停止操作) 时 • 复位 		<ul style="list-style-type: none"> • 当 SDA0 引脚在高电平时

SMC0	操作模式转换
0	在标准模式中操作。
1	在高速模式中操作。

DFC0	数字滤波器操作控制
0	关闭数字滤波器。
1	打开数字滤波器。
数字滤波器只能在高速模式中使用。 在高速模式中, 不论 DFC0 位是置位 (1) 还是清除 (0), 传输时钟不会变化。 数字滤波器在高速模式中用于清除噪声。	

注 第 4 和 5 位为只读。

备注 IICE0: IIC 控制寄存器 0 (IICC0) 的第 7 位

(5) IIC 功能扩展寄存器 0 (IICX0)

该寄存器设置了 I²C 的功能扩展。

IICX0 通过 1 位或 8 位存储器操作指令来设置。结合根据 IIC 时钟选择寄存器 0 (IICCL0) 的第 3、1 和 0 (SMC0, CL01, 和 CL00) 位来设置 CLX0 (参见 15.3 (6) IIC 传输时钟设置方法)。

当 IIC 控制寄存器 0 (IICCC0) 的第 7 位 (IICE0) 为 1 时对 IICX0 进行设置。

复位信号产生将 IICX0 设置为 00H。

图 15-9. IIC 功能扩展寄存器 0 (IICX0) 的格式

地址: FFA9H	复位后: 00H	R / W						
符号	7	6	5	4	3	2	1	<0>
IICX0	0	0	0	0	0	0	0	CLX0

(6) IIC 传输时钟设置方法

使用以下表达式来计算 IIC 传输时钟频率 (fscl)。

$$f_{SCL} = 1/(m \times T + t_r + t_f)$$

m = 12, 24, 44, 86 (见 表 15-2 选择时钟设置)

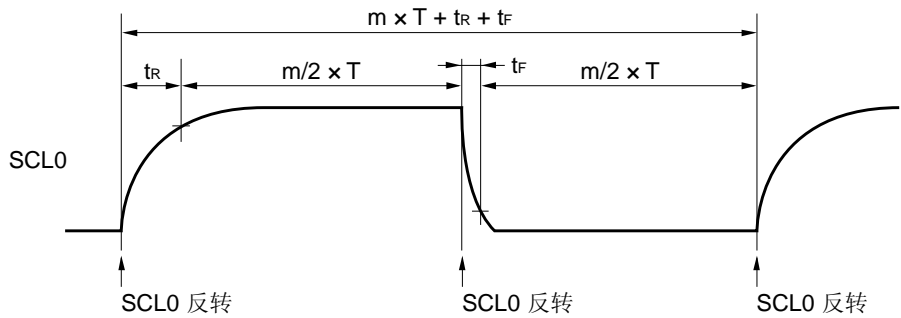
T: 1/fw

t_r: SCL0 上沿时间

t_f: SCL0 下沿时间

例如, 当 fw = fPRS/2 = 4.19 MHz, m = 86, t_r = 200 ns 以及 t_f = 50 ns 时使用以下表达式计算的 IIC 传输时钟频率 (fscl)。

$$f_{SCL} = 1/(86 \times 238.7 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 48.1 \text{ kHz}$$



使用 IIC 时钟选择寄存器 0 (IICCL0) 的第 3、1 和 0 位 (SMC0、CL01 和 CL00) 与 IIC 功能扩展寄存器 0 (IICX0) 的第 0 位 (CLX0) 的组合来对选择时钟进行设置。

表 15-2. 选择时钟设置

IICX0 位 0	IICCL0			选择时钟 (fw) ^{注 1,2}	传输时钟 (fw/m)	可设置的选择时钟 (fw) 范围	操作模式
	位 3	位 1	位 0				
CLX0	SMC0	CL01	CL00				
0	0	0	0	f _{PRS} /2	fw/44	2.00 到 4.19 MHz	正常模式 (SMC0 位 = 0)
0	0	0	1	f _{PRS} /2	fw/86	4.19 到 8.38 MHz	
0	0	1	0	f _{PRS} /4	fw/86		
0	1	0	×	f _{PRS} /2	fw/24	4.00 到 8.38 MHz	高速模式 (SMC0 位 = 1)
0	1	1	0	f _{PRS} /4	fw/24		
1	1	0	×	f _{PRS} /2	fw/12	4.00 到 4.19 MHz	
1	1	1	0	f _{PRS} /4	fw/12		
除上面以外				禁止设置			

- 注 1. 如果外围硬件时钟 (f_{PRS}) 工作于高速系统时钟 (f_{XH}) (XSEL = 1), f_{PRS} 工作频率根据供电电压改变。
- V_{DD} = 2.7 到 5.5 V: f_{PRS} ≤ 20 MHz
 - V_{DD} = 1.8 到 2.7 V: f_{PRS} ≤ 5 MHz
2. 如果外围硬件时钟 (f_{PRS}) 工作于内部高速振荡时钟 (f_{XH}) (XSEL = 0), 按照下面设置 CLX0、SMC0、CL01 和 CL00。

IICX0 位 0	IICCL0			选择时钟 (fw) ^{注 1,2}	传输时钟 (fw/m)	可设置的选择时钟 (fw) 范围	操作模式
	位 3	位 1	位 0				
CLX0	SMC0	CL01	CL00				
0	0	0	0	f _{PRS} /2	fw/44	3.8 MHz 到 4.2 MHz	正常模式 (SMC0 位 = 0)
0	1	0	×	f _{PRS} /2	fw/24		高速模式 (SMC0 位 = 1)

注意事项 在允许操作 (通过将 IIC 控制寄存器 0 (IICC0) 的第 7 位 (IICE0) 设置为 1) 前, 通过使用 CLX0, SMC0, CL01 以及 CL00 来确定 I²C 的传输时钟频率。要改变传输时钟频率, 需将 IICE0 清 0。

- 备注 1. ×: 无关
2. f_{PRS}: 外围硬件时钟频率

(7) 端口模式寄存器 6 (PM6)

该寄存器以 1 位为单位设置端口 6 的输入 / 输出。

当将 P60 / SCL0 引脚用作时钟输入 / 输出并且将 P61 / SDA0 引脚用作串行数据输入 / 输出时，将 PM60 和 PM61 以及 P60 和 P61 的输出锁存器清 0。

在设置输出模式前将 IICE0 (IIC 控制寄存器 0 (IICC0) 的第 7 位) 设置为 1，因为当 IICE0 为 0 时，P60 / SCL0 和 P61 / SDA0 引脚将输出一个低电平 (恒定)。

PM6 通过 1 位或 8 位存储器操作指令来设置。

复位信号产生将 PM6 设置为 FFH。

图 15-10. 端口模式寄存器 6 (PM6) 的格式

地址: FF26H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PM6n	P6n 引脚输入/输出模式选择 (n = 0, 1)
0	输出模式 (输出缓冲器开)
1	输入模式 (输出缓冲器关)

15.4 I²C总线模式功能

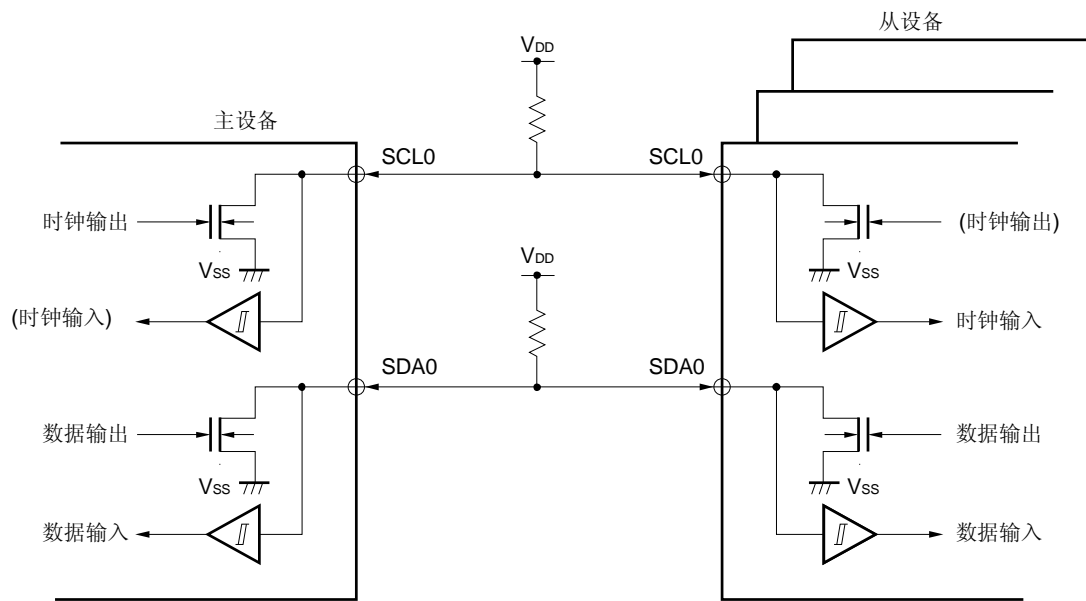
15.4.1 引脚配置

串行时钟引脚（SCL0）和串行数据引脚（SDA0）由以下内容组成。

- (1) SCL0.... 该引脚用于串行时钟输入和输出。
该引脚是主设备和从设备的 N-ch 漏极开路输出。输入为 Schmitt 输入。
- (2) SDA0.... 该引脚用于串行数据输入和输出。
该引脚既是主设备和从设备的 N-ch 漏极开路输出。输入为 Schmitt 输入。

由于串行时钟线和串行数据总线的输出均为 N-ch 漏极开路输出，因此需要一个外部上拉电阻。

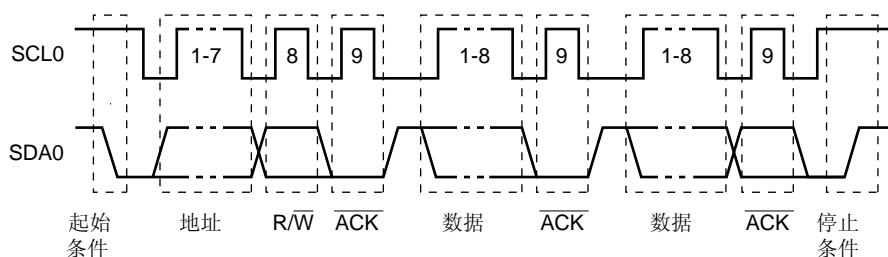
图 15-11. 引脚配置图



15.5 I²C总线定义和控制方法

以下的章节描述了 I²C 总线串行数据通信格式以及由 I²C 总线所使用的信号。图 15-12 显示了通过 I²C 总线串行数据总线输出的“起始条件”、“地址”、“数据”以及“停止条件”的传输时序。

图 15-12. I²C 总线串行数据传输时间



主设备产生起始条件，从设备地址以及停止条件。

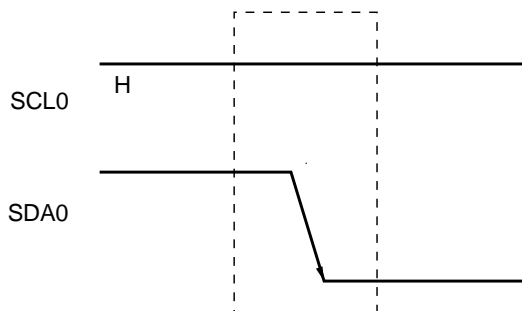
可以通过主设备或从设备来产生响应 (\overline{ACK}) (通常，由接收 8 位数据的设备输出)。

串行时钟 (SCL0) 通过主设备连续输出。然而，在从设备中，SCL0 的低电平周期可以扩展且可以插入一个等待。

15.5.1 起始条件

当 SCL0 引脚处于高电平且 SDA0 引脚从高电平变为低电平时，满足起始条件。当开始串行传输时，SCL0 引脚和 SDA0 引脚的起始条件是主设备产生并发送给从设备的信号。当设备用作从设备时，可以检测到起始条件。

图 15-13. 起始条件



检测到停止条件 (SPD0: IIC 状态寄存器 0 (IICS0) 中的第 0 位 = 1) 后，当 IIC 控制寄存器 0 (IICS0) 的第 1 位 (STT0) 被设置为 1 时，将输出一个起始条件。当检测到起始条件时，IICS0 的第 1 位 (STD0) 将被设置为 1。

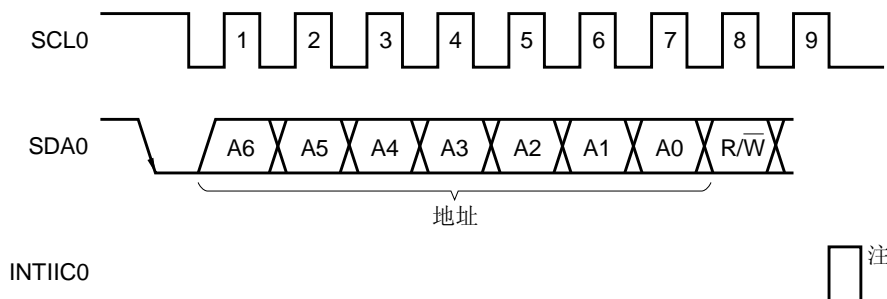
15.5.2 地址

起始条件后的数据的 7 位定义为地址。

地址是一个 7 位数据段，用于选择一个通过总线连接到主设备的从设备。因此，每个通过总线连接的从设备必须拥有唯一的地址。

从设备包括用于检测起始条件，并检查 7 位地址数据是否与保存在从设备地址寄存器 0 (SVA0) 中的数据值相匹配的硬件。如果地址数据匹配 SVA0 值，那么将检测到该从设备，且从设备将与主设备进行通信，直到主设备产生一个起始条件或停止条件。

图 15-14. 地址



注 如果在从设备操作过程中接收到不同于本地地址或扩展代码的数据，将不会产生 INTIIC0。

从设备地址和第 8 位（指定传输方向，如 15.5.3 传输方向指定所述）一起被写入 IIC 移位寄存器 0 (IIC0) 中，然后输出。接收到的地址被写入 IIC0 中。

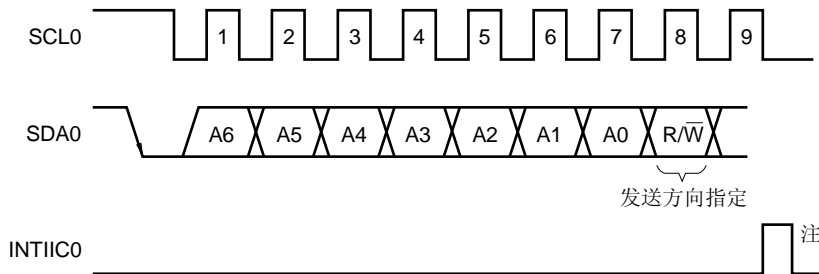
从设备地址被分配到 IIC0 中的高 7 位。

15.5.3 传输方向指定

除了 7 位地址数据外，主设备还会发送 1 位数据用于指定传输方向。

当该传输方向指定位的值为“0”时，表示主设备正在将数据传输到从设备中。当传输方向指定位的值为“1”时，表示主设备正在从从设备中接收数据。

图 15-15. 传输方向指定



注 如果在从设备操作过程中接收到不同于本地地址或扩展代码的数据，将不会产生 INTIIC0。

15.5.4 响应 ($\overline{\text{ACK}}$)

$\overline{\text{ACK}}$ 信号用于在传输方或接收方检查串行数据的状态。

每次接收到 8 位数据时，接收方都会返回 $\overline{\text{ACK}}$ 信号。

发送方通常在传输 8 位数据后接收 $\overline{\text{ACK}}$ 信号。当从接收方返回 $\overline{\text{ACK}}$ 信号时，认为已经正确接收并继续处理。是否被检测到 $\overline{\text{ACK}}$ 信号，可以通过使用 IIC 状态寄存器 0 (IICS0) 的第 2 位 (ACKD0) 来检查。

当主设备接收到最后的数据项时，它不会返回 $\overline{\text{ACK}}$ 信号，相反，它会产生一个停止条件。如果在接收数据后从设备没有返回 $\overline{\text{ACK}}$ 信号，那么主设备将输出一个停止条件或重启条件并停止传输。如果没有返回 $\overline{\text{ACK}}$ 信号，可能是由于以下的原因造成的。

- <1> 没有正常完成接收。
- <2> 接收到最后一个数据项。
- <3> 由地址指定的接收方不存在。

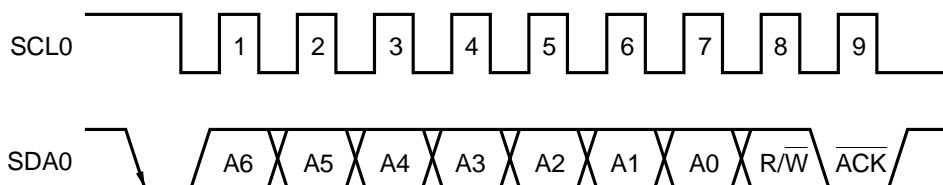
要产生 $\overline{\text{ACK}}$ 信号，接收方需在第九个时钟将 SDA0 变为低电平（表示正常接收）。

通过将 IIC 控制寄存器 0 (IICC0) 的第 2 位 (ACKE0) 设置为 1 允许自动产生 $\overline{\text{ACK}}$ 信号。由 7 位地址信息后的第八位的数据设置 IICS0 寄存器的第 3 位 (TRC0)。通常，将 ACEK0 设置为 1 用于接收 (TRC0=0)。

如果在接收过程 (TRC0 = 0) 中，从设备不再接收数据或者不再需要下一个数据项，那么从设备必须通过将 ACEK0 清 0，以通知主设备它将不再接收任何数据。

在接收过程 (TRC0 = 0) 中，当主设备不需要下一个数据项时，它必须将 ACEK0 清 0，这样就不再产生 $\overline{\text{ACK}}$ 信号。通过这种方式，主设备在传输时通知从设备它不再需要任务数据 (将停止发送)。

图 15-16. $\overline{\text{ACK}}$



当接收到本地地址时，不管 ACEK0 的值是什么，都将自动产生 $\overline{\text{ACK}}$ 信号。当接收到非本地地址时，将不会产生 $\overline{\text{ACK}}$ 信号 (NACK)。

当接收到扩展代码时，如果 ACEK0 事先设置为 1，则产生 $\overline{\text{ACK}}$ 信号。

依赖于如下等待时序的设置，接收数据时产生 $\overline{\text{ACK}}$ 的方法是不同的。

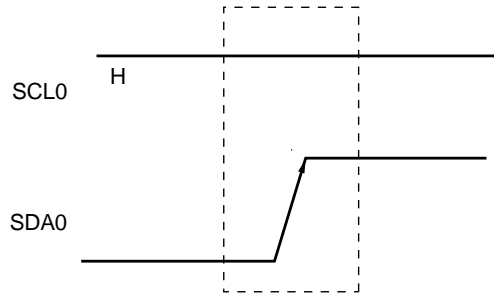
- 当选择第 8 时钟等待状态 (IICC0 寄存器的第 3 位 (WTIM0) = 0) 时：
通过在释放等待状态前将 ACEK0 设置为 1， $\overline{\text{ACK}}$ 信号会在 SCL0 引脚的第八个时钟的下降沿产生。
- 当选择第 9 时钟等待状态 (IICC0 寄存器的第 3 位 (WTIM0) = 1) 时：
通过事先将 ACEK0 设置为 1 来产生 $\overline{\text{ACK}}$ 信号。

15.5.5 停止条件

当 SCL0 引脚处于高电平时，将 SDA0 引脚从低电平变为高电平将会产生一个停止条件。

停止条件是完成串行传输后，主设备产生并发送给从设备的信号。当设备被用作从设备时，可以检测到停止条件。

图 15-17. 停止条件



当 IIC 控制寄存器 0 (IICC0) 的第 0 位 (SPT0) 设置为 1 时，将产生一个停止条件。当检测到停止条件时，IIC 状态寄存器 0 (IICS0) 的第 0 位 (SPD0) 将被设置为 1，并且当 IICC0 的第 4 位 (SPIE0) 设置为 1 时将产生 INTIIC0 信号。

15.5.6 等待

等待用于通知通信方设备（主设备或从设备）正在准备发送或接收数据（也就是处于等待状态）。

将 SCL0 引脚设置为低电平用于通知通信方处于等待状态。当为主设备和从设备取消等待状态时，可以开始下一个数据传输。

图 15-18. 等待 (1 / 2)

(1) 当主设备输出第九个时钟后等待而从设备输出第八个时钟后等待时
(主设备传输，从设备接收，且 ACKE0 = 1)

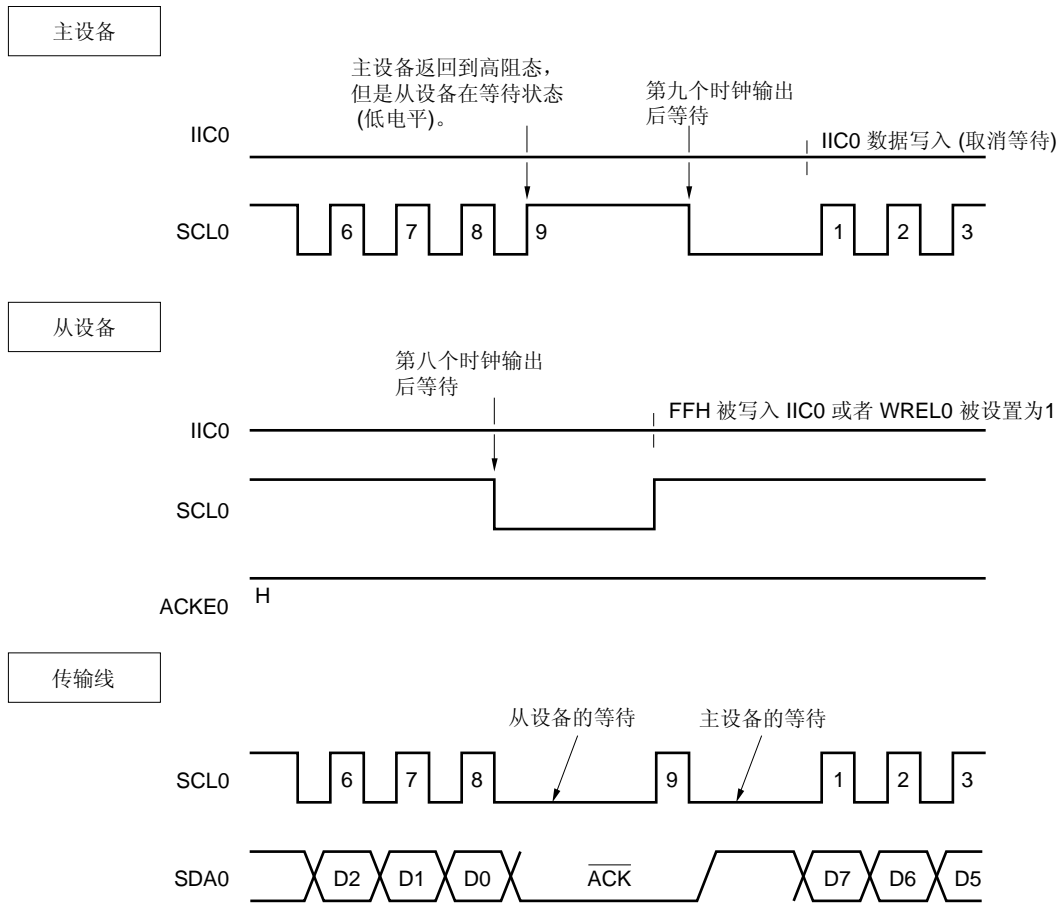
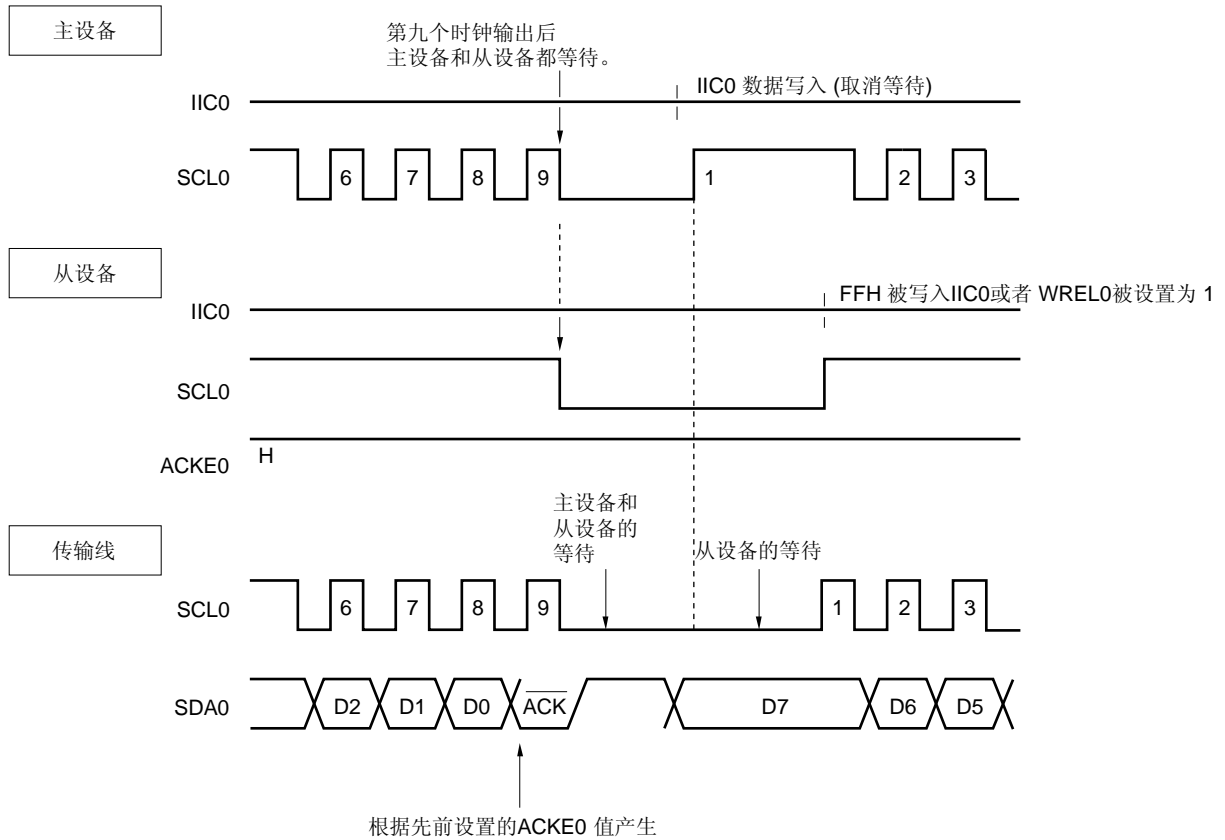


图 15-18. 等待 (2 / 2)

(2) 当主设备和从设备都输出九个时钟后等待
(主设备传输, 从设备接收, 且 **ACKE0 = 1**)



备注 **ACKE0**: IIC 控制寄存器 0 (IICC0) 的第 2 位
 WRELO: IIC 控制寄存器 0 (IICC0) 的第 5 位

根据 IIC 控制寄存器 0 (IICC0) 的第 3 位 (WTIM0) 的设置可能自动产生一个等待。
通常情况下, 当 IICC0 的第 5 位 (WRELO) 设置为 1 或者当 FFH 写入 IIC 移位寄存器 0 (IIC0) 时, 接收方将取消等待状态, 而当数据写入 IIC0 时, 发送方将取消等待状态。

- 主设备也可以通过以下的任一方法取消等待状态。
- 通过将 IICC0 的第 1 位 (STT0) 设置为 1
 - 通过将 IICC0 的第 0 位 (SPT0) 设置为 1

15.5.7 取消等待

I²C 通常通过以下处理取消等待状态。

- 将数据写入 IIC 移位寄存器 0 (IIC0)
- 对 IIC 控制寄存器 0 (IICC0) 的第 5 位 (WRELO) 进行设置 (取消等待)
- 对 IIC0 寄存器的第 1 位 (STT0) 进行设置 (产生起始条件)^注
- 对 IIC0 寄存器的第 0 位 (SPT0) 进行设置 (产生停止条件)^注

注 只有主设备

当执行以上等待取消处理时, IIC 将取消等待状态且恢复通信。

要取消等待状态并且发送数据 (包括地址), 则应将数据写入 IIC0。

要在取消等待状态后接收数据或完成数据发送, 则应将 IIC0 控制寄存器 0 (IICC0) 的第 5 位 (WRELO) 设置为 1。

要在取消等待状态后产生一个重启条件, 则应将 IICC0 的第 1 位 (STT0) 设置为 1。

要在取消等待状态后产生一个停止条件, 则应将 IICC0 的第 0 位 (SPT0) 设置为 1。

每个等待状态只执行一次取消处理。

例如, 如果通过将 WRELO 设置为 1 来取消等待状态后数据写入 IIC0, 那么一个不正确的值可能被输出到 SDA0, 因为改变 SDA0 线的时序与对 IIC0 进行写操作的时序发生冲突。

除了上述情况之外, 当终止通信时, 如果 IICE0 被清 0, 将停止通信, 因此, 取消等待状态。

如果 I²C 总线由于噪声被死锁, 将通过设置 IICC0 中的第 6 位 (LRELO) 来从通信中保存处理过程, 因此, 等待状态被取消。

15.5.8 中断请求 (INTIIC0) 产生时间和等待控制

IIC 控制寄存器 0 (IICC0) 的第 3 位 (WTIM0) 的设置决定了 INTIIC0 产生的时序以及相应的等待控制, 如表 15-3 所示。

表 15-3. INTIIC0 产生时序和等待控制

WTIM0	从设备操作期间			主设备操作期间		
	地址	数据接收	数据传输	地址	数据接收	数据传输
0	9 ^{注1,2}	8 ^{注2}	8 ^{注2}	9	8	8
1	9 ^{注1,2}	9 ^{注2}	9 ^{注2}	9	9	9

注 1. 只有当与设置到从设备地址寄存器 0 (SVA0) 中的地址匹配时, 将在第九个时钟的下降沿产生从设备的 INTIIC0 信号和等待周期。

这种情况下, 不论 IICC0 的第 2 位 (ACKE0) 的值是什么, 都将产生 \overline{ACK} 信号。对于已经接收到一个扩展码的从设备来说, 将在第八个时钟的下降沿产生 INTIIC0 信号。

然而, 如果在重启后, 地址仍不匹配, 则在第九个时钟的下降沿将产生 INTIIC0 信号, 但不会产生等待。

2. 如果接收到的地址与从设备地址寄存器 0 (SVA0) 的内容不匹配, 且没有接收到扩展码, 那么 INTIIC0 信号和等待都不会产生。

备注 表中的数字表示串行时钟的时钟信号的数量。中断请求和等待控制都会与这些时钟信号的下降沿同步。

(1) 在地址发送 / 接收期间

- 从设备操作：不论 WTIMO 位是什么，都将根据上述注 1 和注 2 中所描述的条件确定中断及等待时序。
- 主设备操作：不论 WTIMO 位是什么，都会在第 9 个时钟的下降沿产生中断及等待时序。

(2) 在数据接收期间

- 主 / 从设备操作：由 WTIMO 位确定中断及等待时序。

(3) 数据发送期间

- 主 / 从设备操作：由根据 WTIMO 位确定中断及等待时序。

(4) 等待取消方法

四种等待取消方法如下所示。

- 将数据写入 IIC 移位寄存器 0 (IIC0)
- 对 IIC 控制寄存器 0 (IICCO) 的第 5 位 (WRELO) 进行设置 (取消等待)
- 对 IIC0 寄存器的第 1 位 (STT0) 进行设置 (产生起始条件)^注
- 对 IIC0 寄存器的第 0 位 (SPT0) 进行设置 (产生停止条件)^注

注 只有主设备

当已经选择了一个第 8 时钟等待 (WTIMO=0) 时，必须在等待取消前确定有无 $\overline{\text{ACK}}$ 的产生。

(5) 停止条件检测

当检测到停止条件时，将产生 INTIIC0 (仅当 SPIE0=1 时) 信号。

15.5.9 地址匹配检测方法

在 I²C 总线模式中，主设备可以通过传输相应的从设备地址来选择一个特定的从设备。

可以通过硬件自动检测地址匹配。当一个本地地址被设置到从设备地址寄存器 0 (SVA0) 中，并且设置在 SVA0 中的地址与主设备发送的从设备地址匹配时，或者当接收到一个扩展码时，都将产生一个中断请求 (INTIIC0)。

15.5.10 错误检测

在 I²C 总线模式中，通过传输设备的 IIC 移位寄存器 0 (IIC0) 获得数据发送期间串行数据总线 (SDA0) 的状态，这样，发送前的 IIC0 数据可以与已经发送的 IIC0 数据进行比较，以实现发送错误的检测。比较后的数值不匹配时，可以判断为发生了发送错误。

15.5.11 扩展码

(1) 当接收地址的高 4 位是“0000”或“1111”时，代表接收到扩展码的扩展码接收标志（EXC0）将被设置为 1，并且在第八个时钟的下降沿发出中断请求（INTIIC0）。而保存在从设备地址寄存器 0（SVA0）中的本地地址不会受到影响。

(2) 如果通过一个 10 位地址传输设置 SVA0 为“11110xx0”且“11110xx0”由主设备传送，结果如下。需注意在第八个时钟的下降沿将产生 INTIIC0 信号。

- 数据匹配的高四位： EXC0 = 1
- 数据匹配中的七位： COI0 = 1

备注 EXC0: IIC 状态寄存器 0（IICS0）的第 5 位
 COI0: IIC 状态寄存器 0（IICS0）的第 4 位

(3) 由于发生中断请求后的处理过程根据扩展码后数据的不同而不同，因此这类处理由软件执行。如果在从设备运行时接收到扩展码，那么即使从设备的地址不匹配，从设备也会参与通信。例如，在接收到扩展码后，如果你不想将目标设备当作从设备来操作，则应将 IIC 控制寄存器 0（IICC0）的第 6 位（LREL0）设置为 1，为下一次通信操作设置好等待模式。

表 15-4. 主扩展码位定义

从设备地址	R / W 位	描述
0 0 0 0 0 0 0	0	通常调用地址
1 1 1 1 0 X X	0	10 位从设备地址指定（用于地址验证）
1 1 1 1 0 X X	1	10 位从地址指定（用于地址匹配后的读命令发布）

备注 对于上述以外的扩展码，请参见由 NXP 出版的 I²C 总线定义。

15.5.12 仲裁

当多个主设备同时产生一个起始条件时（当在 **STD0** 被设置为 1 前 **STT0** 被设置为 1 时），调整时钟数目直至数据改变后执行主设备间的通信。这种操作被称为仲裁。

当多个主设备中的一个主设备在仲裁中失败时，将通过仲裁失败发生时序设置 **IIC** 状态寄存器 0 (**IICS0**) 中的仲裁失败标志 (**ALD0**) 为 1，而 **SCL0** 和 **SDA0** 线都被设置为高阻，释放总线。

基于下一次中断请求时序（第八个或第九个时钟，当检测到停止条件等）以及通过软件设置 **ALD0=1** 检测仲裁失败。

关于中断请求时序的详细信息，参见 **15.5.17 I²C 中断请求 (INTIIC0)** 发生的时序。

备注 **STD0**: **IIC** 状态寄存器 0 (**IICS0**) 的第 1 位
STT0: **IIC** 控制寄存器 0 (**IICC0**) 的第 1 位

图 15-19. 仲裁时序举例

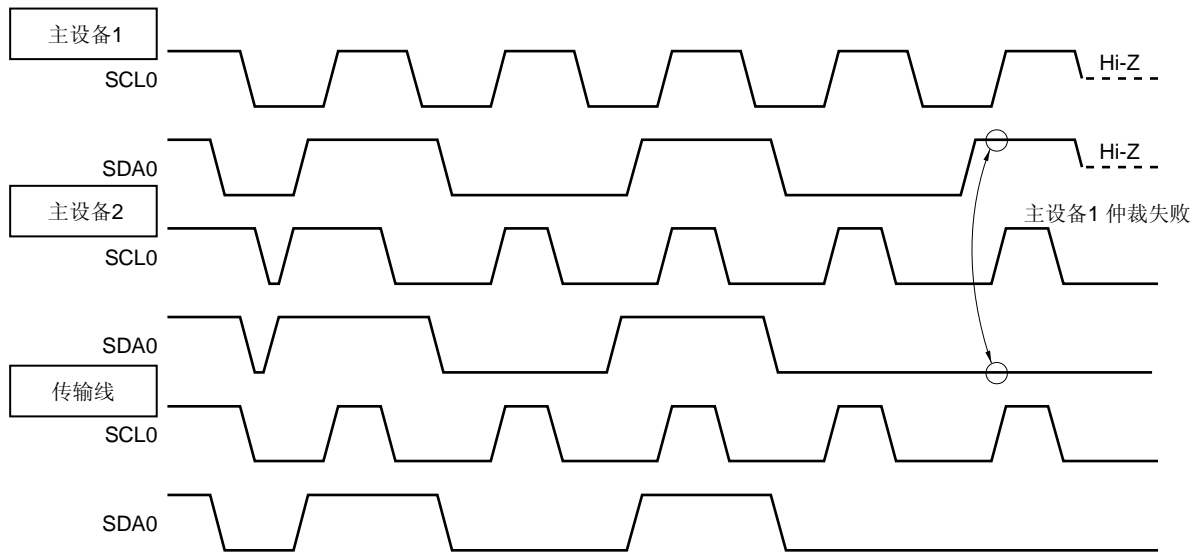


表 15-5. 仲裁期间的状态及中断请求的产生时序

仲裁期间的状态	中断请求产生时序
地址传输期间	在字节传输后的第八个或第九个时钟的下降沿 ^{注1}
地址传输后读取 / 写入数据	
扩展码传输期间	
扩展码传输后读取 / 写入数据	
数据传输期间	
数据传输后 ACK 信号传输周期内	
数据传输期间检测到重启条件时	
数据传输期间检测到停止条件时	当产生停止条件时 (当 SPIE0=1) ^{注2}
数据处于低电平时, 此时正试图产生一个重启条件	在字节传输后的第八个或第九个时钟的下降沿 ^{注1}
检测到停止条件时, 此时正试图产生一个重启条件	当产生停止条件时 (当 SPIE0=1) ^{注2}
数据处于低电平时, 此时正试图产生一个停止条件	在字节传输后的第八个或第九个时钟的下降沿 ^{注1}
SCL0 处于低电平时, 此时正试图产生一个重启条件	

- 注**
1. 当 WTIMO (IIC 控制寄存器 0 (IICC0) 的第 3 位) =1 时, 将在第九个时钟的下降沿产生中断请求。当 WTIMO=0, 并且接收到扩展码的从设备地址时, 将在第八个时钟的下降沿产生中断请求。
 2. 当有可能发生仲裁时, 应设置 SPIE0=1, 进行主设备操作。

备注 SPIE0: IIC 控制寄存器 0 (IICC0) 的第 4 位

15.5.13 唤醒功能

当已经接收到一个本地地址和一个扩展码时, I²C 总线的从设备唤醒功能用于产生一个中断请求信号 (INTIIC0)。

当地址不匹配时, 该功能通过防止不必要的 INTIIC0 信号的产生以使处理更为有效。

当检测到一个起始条件时, 将设置唤醒待机模式。由于仲裁失败可能更改主设备为从设备, 此时发送地址信息, 该唤醒待机模式有效。

然而, 当检测到一个停止条件时, 将设置 IIC 控制寄存器 0 (IICC0) 的第 4 位 (SPIE0), 而与唤醒功能无关, 这也决定了中断请求是否被允许。

15.5.14 通信预约

(1) 当允许通信预约功能时 (IIC 标志寄存器 0 (IICF0) 的第 0 位 (IICRSV) =0)

为了启动当前没有使用总线的主设备通信，可以使用通信预约功能，当总线被释放时允许发送一个起始条件信号。在两种模式下不使用总线。

- 仲裁的结果不是主设备和从设备操作时
- 接收到扩展码并且禁止从设备操作 (当 IIC 控制寄存器 0 (IICC0) 的第 6 位 (LREL0) 被设置为 1 时，不返回 $\overline{\text{ACK}}$ 信号，总线被释放) 时。

没有使用总线时 (检测到一个停止条件后)，如果 IICC0 的第 1 位 (STT0) 被设置为 1，则自动产生一个起始条件，并设置等待状态。

在 IICC0 的第 4 位 (SPIE0) 被设置为 1 后，如果一个地址写入到 IIC 移位寄存器 0 (IIC0) 中，并且通过释放总线的中断请求信号 (INTIIC0) 检测到该地址 (检测到停止条件)，则作为主设备自动开始通信。在检测到停止条件前写入到 I²C 的数据是无效的。

当 STT0 被设置为 1 时，操作模式 (如起始条件或者通信预约) 将会根据总线状态来确定。

- 如果总线已经被释放.....产生起始条件
- 如果总线没有被释放 (等待模式)通信预约

在 STT0 设置为 1，并且当等待时序过去后，通过使用 MSTS0 (IIC 状态寄存器 0 (IICS0) 的第 7 位) 来检查通信预约是否运行。

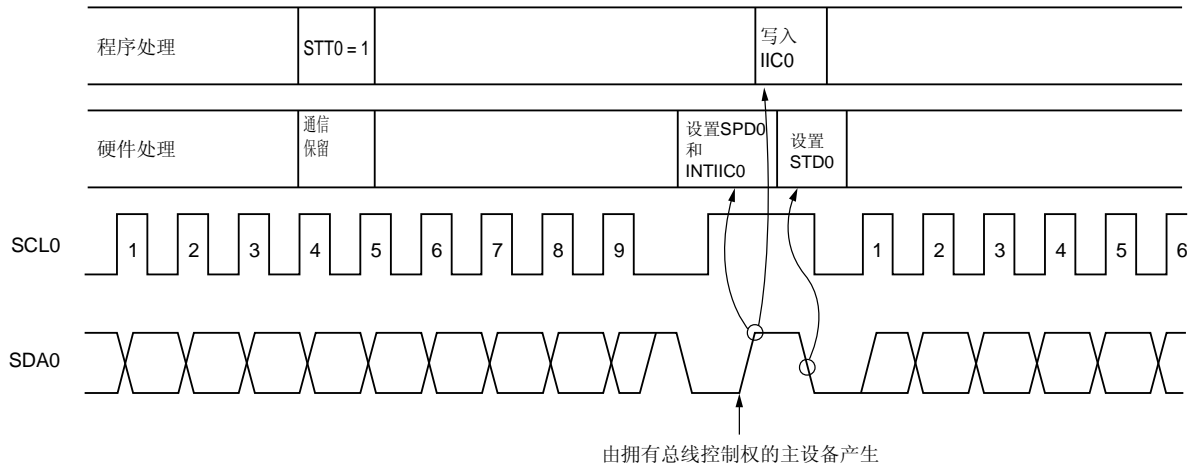
通过软件设置的等待周期被列于表 15-6 中。

表 15-6. 等待周期

CLX0	SMC0	CL01	CL00	等待周期
0	0	0	0	46 个时钟
0	0	0	1	86 个时钟
0	0	1	0	172 个时钟
0	1	0	0	30 个时钟
0	1	0	1	
0	1	1	0	60 个时钟
1	1	0	0	18 个时钟
1	1	0	1	
1	1	1	0	36 个时钟

图 15-20 显示了通信预约时序。

图 15-20. 通信预约时序



备注 IIC0: IIC 移位寄存器 0
 STT0: IIC 控制寄存器 0 (IICC0) 的第 1 位
 STD0: IIC 状态寄存器 0 (IICS0) 的第 1 位
 SPD0: IIC 状态寄存器 0 (IICS0) 的第 0 位

通过以下时序接收通信预约。在 IIC 状态寄存器 0 (IICS0) 的第 1 位 (STD0) 设置为 1 后, 在检测到停止条件前, 可以通过设置 IIC 控制寄存器 0 (IICC0) 的第 1 位 (STT0) 来完成通信预约。

图 15-21. 接收通信预约的时序

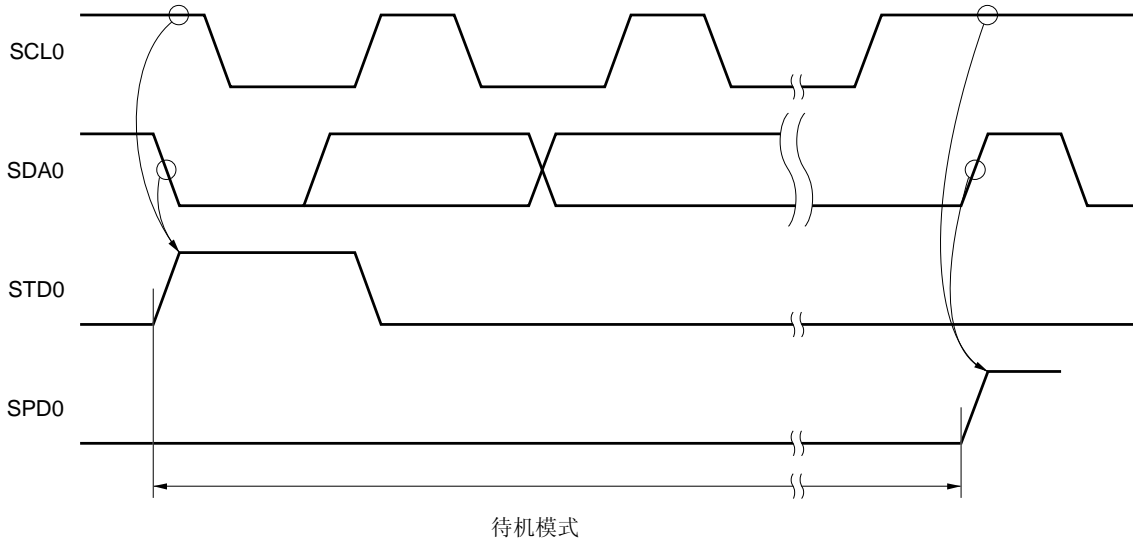
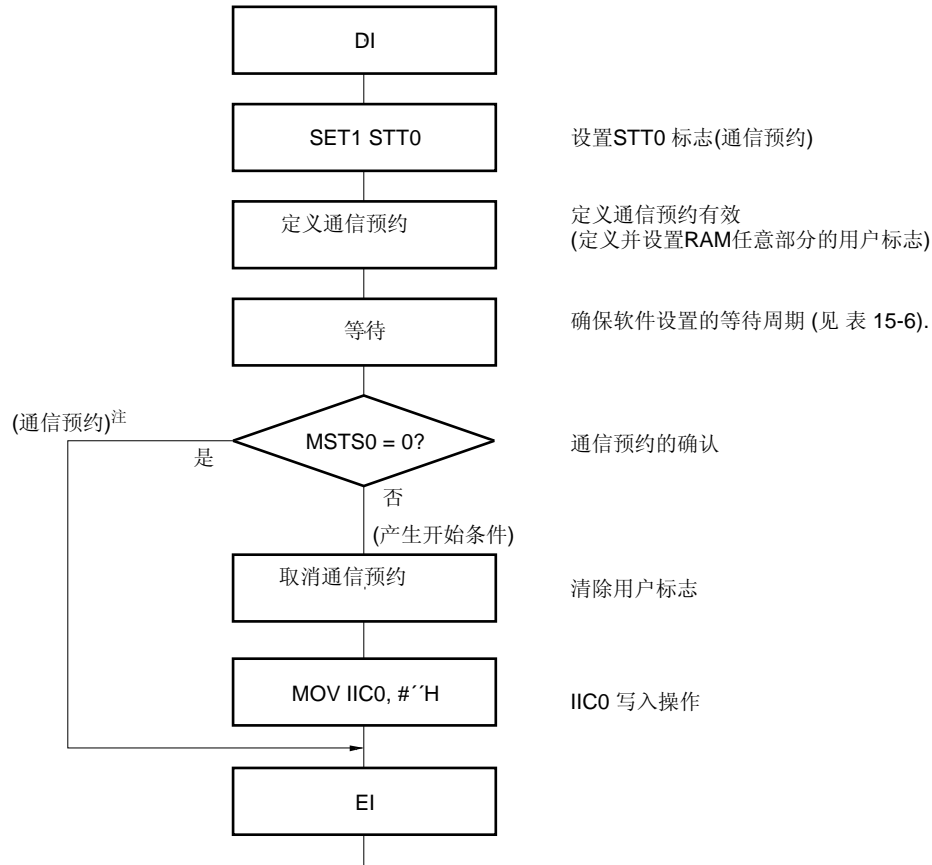


图 15-22 显示了通信预约协议。

图 15-22. 通信预约协议



注 当发生停止条件中断请求时，通信预约操作将对 IIC 移位寄存器 0 (IIC0) 执行写操作。

备注 STT0: IIC 控制寄存器 0 (IICC0) 的第 1 位
 MSTS0: IIC 状态寄存器 0 (IICS0) 的第 7 位
 IIC0: IIC 移位寄存器 0

(2) 当禁止通信预约功能时 (IIC 标志寄存器 0 (IICF0) 的第 0 位 (IICRSV) =1)

在总线通信期间，总线未使用时将 IIC 控制寄存器 0 (IICC0) 的第 1 位 (STT0) 设置为 1 时，那么该请求会被拒绝并且不会产生起始条件。以下两种状态包含在没有使用总线的状态。

- 仲裁的结果不是主设备和从设备操作时
- 接收到扩展码并且禁止从设备操作 (不返回 ACK 信号，当 IIC 控制寄存器 0 (IICC0) 的第 6 位 (LREL0) 设置为 1 时释放总线)。

检查 STCF (IICF0 的第 7 位)，以确定是否产生起始条件，请求是否被拒绝。在设置 STT0=1 后，需要表 15-7 中显示的时间直到 STCF 被设置为 1。因此，需要通过软件确保时间的安全。

表 15-7. 等待周期

CL01	CL00	等待周期
0	0	6 个时钟
0	1	6 个时钟
1	0	12 个时钟

15.5.15 其它注意事项

(1) 当 STCEN (IIC 标志寄存器 0 (IICF0) 的第 1 位) = 0 时

在允许 I²C 操作后 (IICE0=1)，不论实际的总线状态如何，总线通信状态 (IICBSY (IICF0 的第 6 位) =1) 都将立即被识别。当从一个没有检测到停止条件的模式变为主设备通信模式时，将先产生一个停止条件以释放总线，然后再执行主设备通信。

当使用多主设备模式时，在没有释放总线时 (当没有检测到停止条件时) 是不能执行主设备通信的。

使用以下顺序来产生一个停止条件。

- <1> 设置 IIC 时钟选择寄存器 0 (IICCL0)。
- <2> 将 IIC 控制寄存器 0 (IICC0) 的第 7 位 (IICE0) 设置为 1。
- <3> 将 IICC0 的第 0 位 (SPT0) 设置为 1。

(2) 当 STCEN = 1 时

在允许 I²C 操作 (IICE0=1) 后，不论实际的总线状态如何，都将立即识别总线释放状态 (IICBSY=0)。产生第一个起始条件 (STT0 (IIC 控制寄存器 0 (IICC0) 的第 1 位) =1)，有必要确认总线已经释放以避免不干扰到其他通信。

(3) 如果其他 I²C 通信已经在进行中

如果允许 I²C 操作，并且当 SDA0 引脚为低电平和 SCL0 引脚为高电平时，设备已经参与了正在进行的通信，那么 I²C 的硬件模块会认为 SDA0 引脚已经变为低电平 (检测到一个起始条件)。如果此时总线上的值能够被识别为一个扩展码，将返回一个 ACK 信号，但这会妨碍其他 I²C 通信。为了避免这种情况，应按以下顺序来开始 I²C 通信。

- <1> 当检测到停止条件时，将 IICC0 的第 4 位 (SPIE0) 清 0 以禁止中断请求信号 (INTIIC0) 的产生。
- <2> 将 IICC0 的第 7 位 (IICE0) 设置为 1 以允许 I²C 的操作。
- <3> 等待起始条件的检测。
- <4> 在 ACK 信号被返回 (IICE0 被设置为 1 后的 4 至 80 个时钟) 前，将 IICC0 的第 6 位 (LREL0) 设置为 1，用来强行禁止检测。

(4) 在允许操作 (IICE0=1) 前，通过使用 SMC0、CL01、CL00 (IICL0 的第 3、1 和 0 位) 以及 CLX0 (IICX0 的第 0 位) 来确定传输时钟频率。要改变传输时钟频率，则需清零 IICE0。

- (5) 在 STT0 和 SPT0 (IICC0 的第 1 和 0 位) 设置后和被清 0 前, 禁止对它们再一次设置。
- (6) 当预约发送时, 应将 SPIE0 (IICL0 的第 4 位) 设置为 1, 这样, 当检测到停止条件时就会产生中断请求。产生中断请求后当通信数据写入 IIC0 时传输开始。当检测到停止条件时, 除非产生中断, 否则设备都会停止在等待状态中, 因为当通信开始时没有产生中断请求。然而, 当通过软件检测 MSTSO (IICS0 的第 7 位) 时, 则没有必要将 SPIE0 设置为 1。

15.5.16 通信操作

以下显示了三种带有流程图的操作流程。

(1) 在单主系统中的主设备操作

在单主系统中, 将 μ PD78F8024 和 78F8025 用作主设备时的流程图如下所示。

该流程图总体上被分为初始设置和通信过程。在启动时执行初始设置。如果需要与从设备通信, 则应先准备通信, 然后执行通信处理。

(2) 多主控制系统中的主设备操作

在 I²C 总线多主控制系统中, 当总线参与通信时, 通过 I²C 总线规范不能判断总线是否被释放或使用。此时, 当数据或时钟保持一定时间 (1 帧) 的高电平时, 在总线释放状态时 μ PD78F8024 和 78F8025 参与通信。

该流程图总体上被分为初始设置、通信等待以及通信处理。这里省略了当 μ PD78F8024 和 78F8025 在仲裁中失败或被指定为从设备时的处理, 仅显示了作为主设备时的处理。在开始参与通信时执行初始设置。然后, 主设备等待通信请求或从设备等待详细指定。实际通信在通信处理中执行, 并且它支持与从设备的发送 / 接收以及其他主设备的仲裁。

(3) 从设备操作

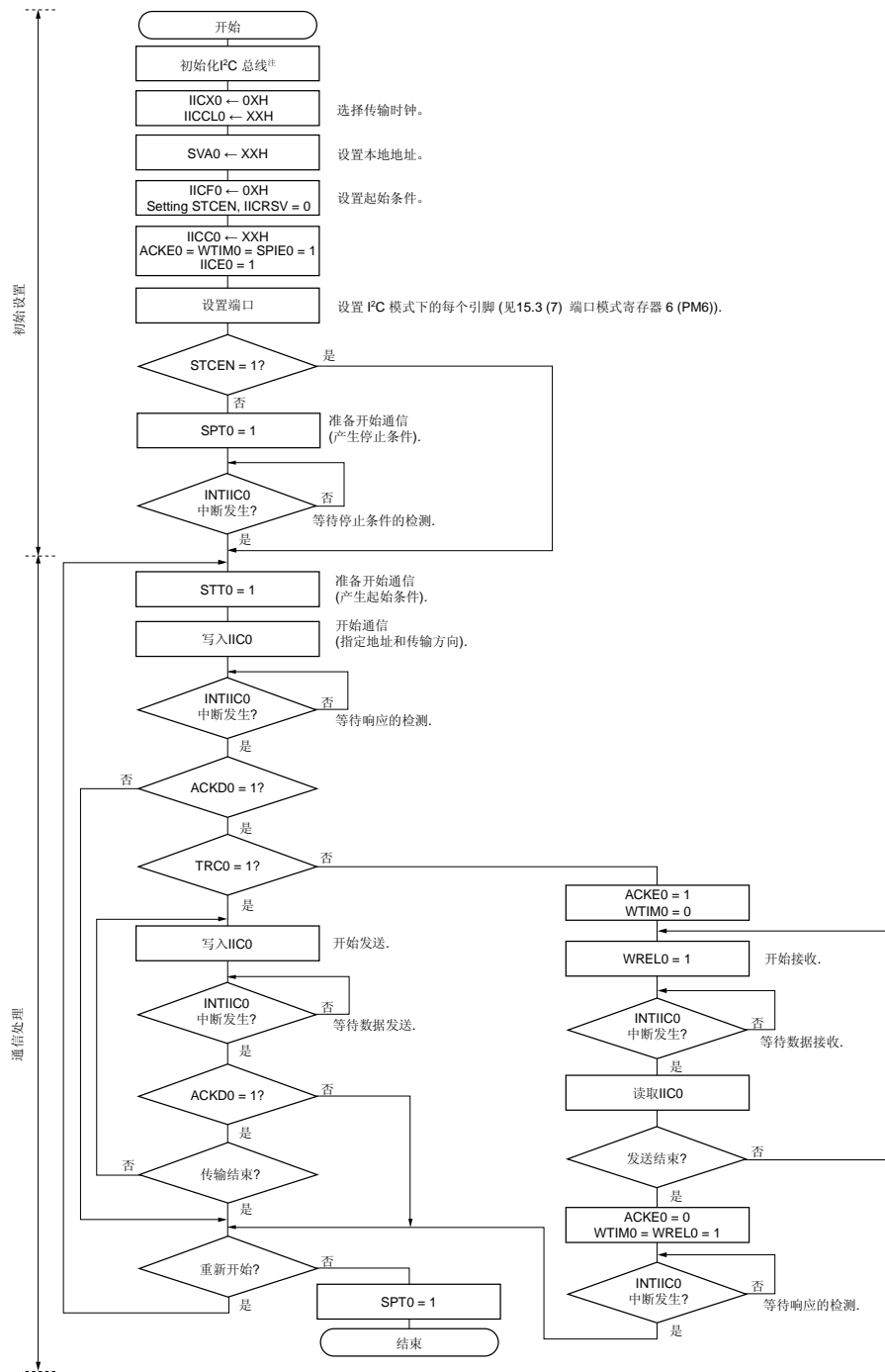
μ PD78F8024 和 78F8025 用作 I²C 总线从设备时的例子如下所示。

当用作从设备时, 将通过中断开始操作。在启动时执行初始设置, 然后等待 INTIIC0 中断出现 (通信等待)。当发生 INTIIC0 中断时, 可以判断通信状态, 并将其结果当作标志传递到主处理中。

通过检查标志可以执行必需的通信处理。

(1) 单主设备系统中的主设备操作

图 15-23. 单主设备系统中的主设备操作

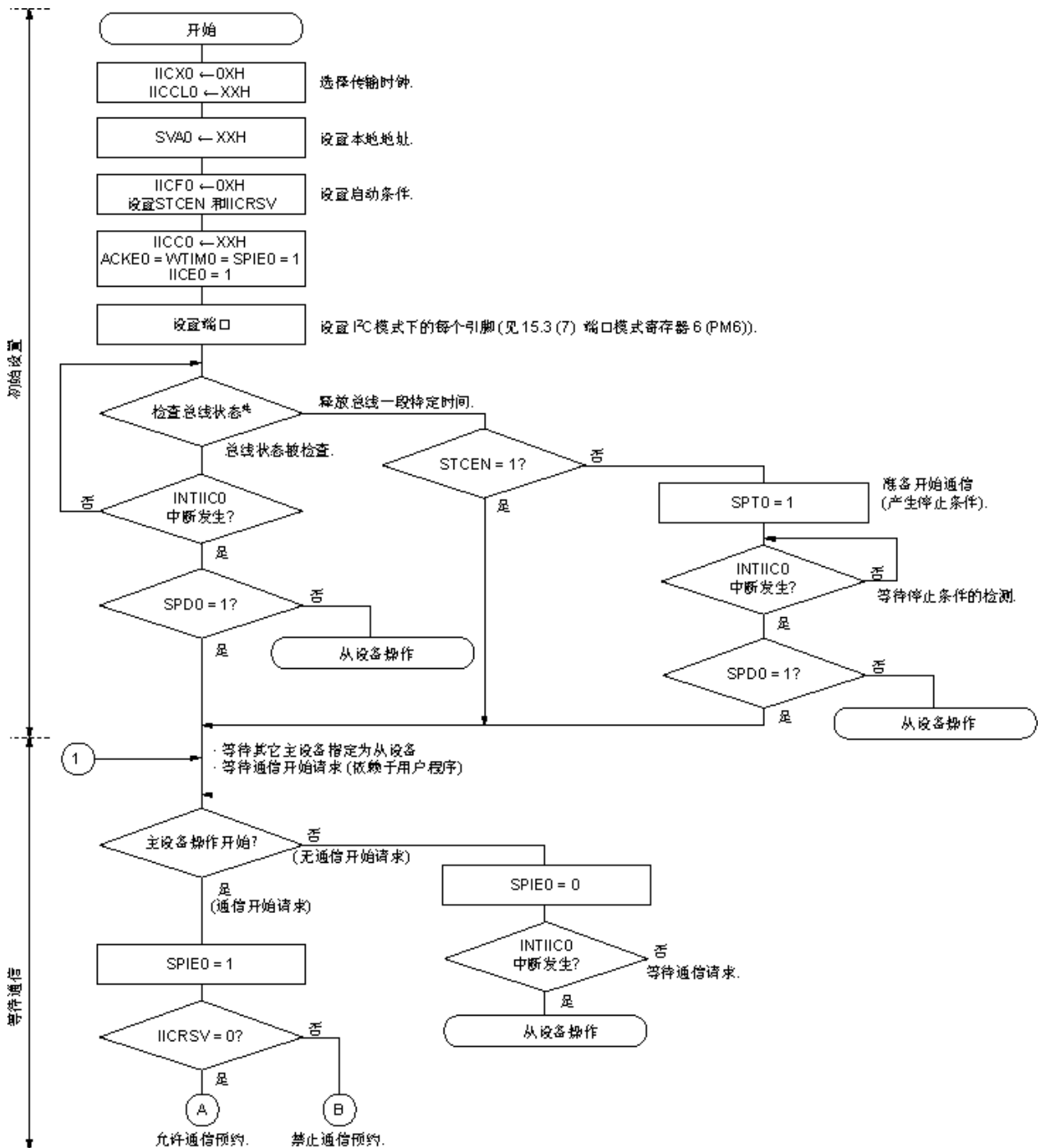


注 按照正在通信的产品规范释放（SCL0 和 SDA0 引脚=高电平）I²C 总线。例如，如果 EEPROM 输出一个低电平到 SDA0 引脚，则将在输出端口模式中设置 SCL0 引脚，并从输出端口中输出一个时钟脉冲直到 SDA0 引脚一直处于高电平。

备注 至于传输和接收格式则应遵守正在通信的产品规范。

(2) 多主控制系统中的主设备操作

图 15-24. 多主控制系统中的主设备操作 (1 / 3)



注 确定总线在一个特定的时间内被释放 (CLD0 位=1, DAD0 位=1) (例如, 在 1 帧的时间内)。如果 SDA0 引脚一直处于低电平, 则应按照正在进行通信的产品规范来决定是否释放 I²C 总线 (SCL0 和 SDA0 引脚=高电平)。

图 15-24. 多主控制系统中的主设备操作 (2 / 3)

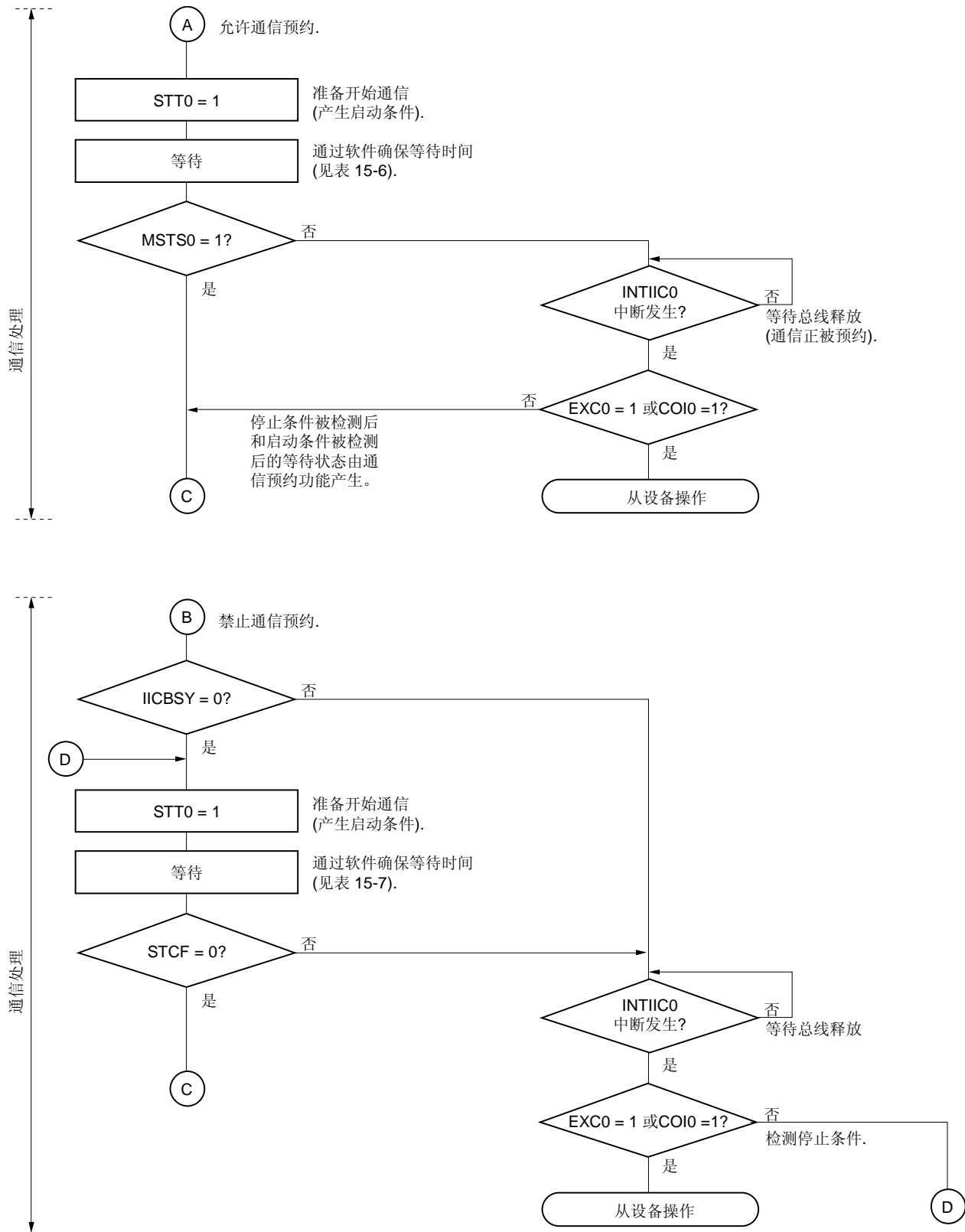
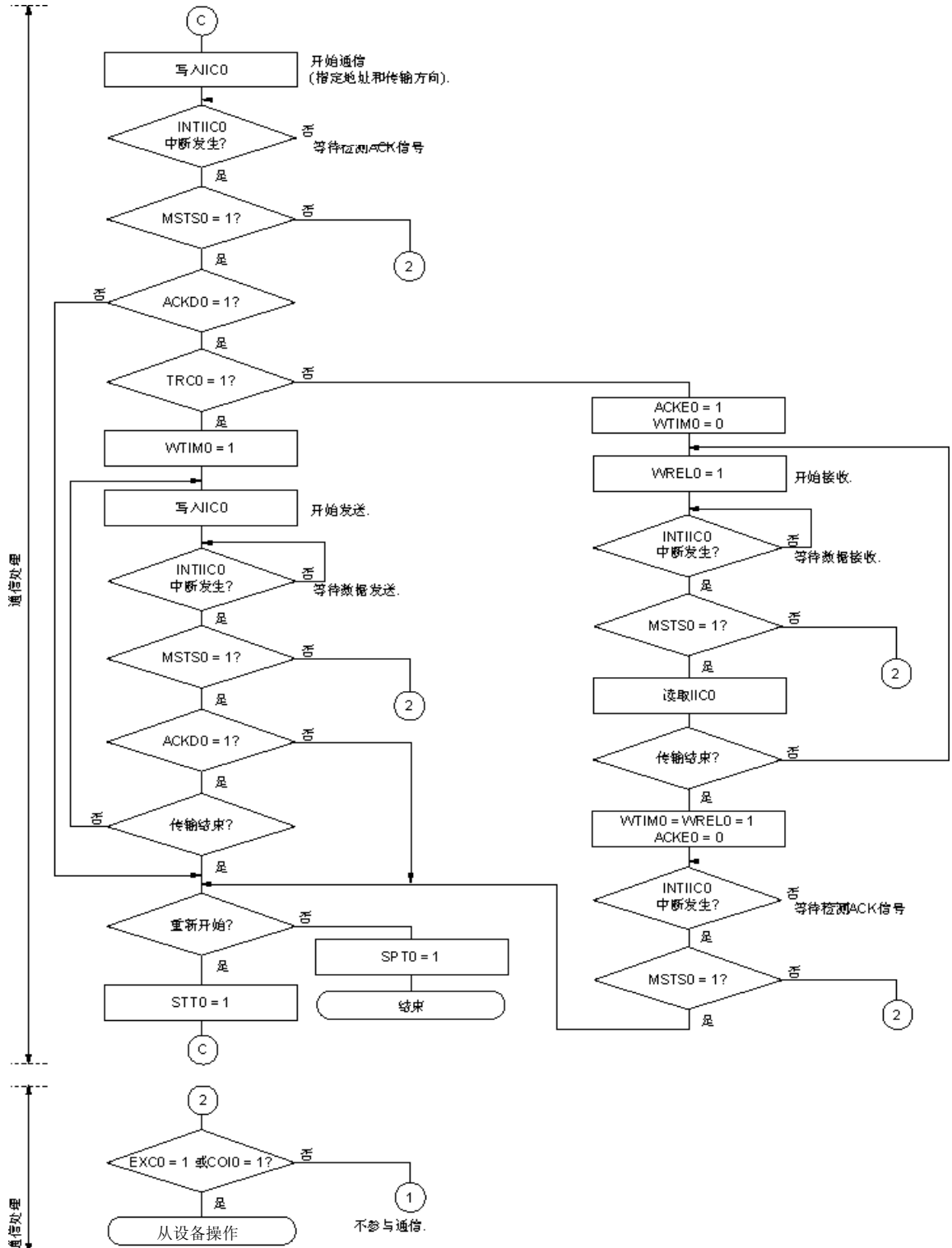


图 15-24. 多主控制系统中的主设备操作 (3 / 3)



备注

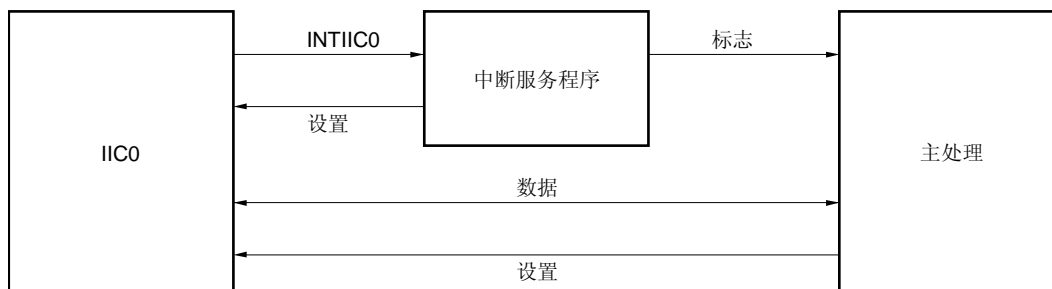
1. 至于发送和接收格式则应遵守正在通信的产品规范。
2. 要在多主控制系统中将设备用作主设备，则应在每次中断 INTIIC0 后读取 MSTS0 位来检查仲裁结果。
3. 要在多主控制系统中将设备用作从设备，则应在每次中断 INTIIC0 发生时通过使用 IICS0 和 IICF0 寄存器来检查状态，并确定要执行的处理。

(3) 从设备操作

从设备操作的处理过程如下。

基本上，从设备操作是事件驱动的操作。因此，必须通过 INTIIC0 中断（必需在实质上改变操作状态的处理，例如在通信期间对停止条件的检测）来处理。

在以下说明中，将假设数据通信不支持扩展码。同时，也将假设 INTIIC0 中断服务只执行状态转换处理，而实际数据通信将通过主处理来执行。



因此，数据通信处理将通过以下三种标志来执行，并且将它们传递到主处理中而不是 INTIIC0。

<1> 通信模式标志

该标志说明了以下两种通信状态。

- 清除模式： 不执行数据通信时的状态
- 通信模式： 执行数据通信时的状态（从有效地址检测到停止条件检测，不从主设备中检测 $\overline{\text{ACK}}$ ，地址不匹配）

<2> 准备标志

该标志表示允许数据通信。它的功能与普通数据通信的 INTIIC0 中断相同。该标志通过中断服务来设置，并通过主处理来清除。当通信开始时将通过中断服务来清除该标志。然而，当第一个数据被发送时，准备标志将不能通过中断服务来设置。因此，在不清除标志的情况下发送第一个数据（对于下一个数据来说，地址匹配被解释为请求）。

<3> 通信方向标志

该标志显示了通信的方向。它的值与 TRC0 相同。

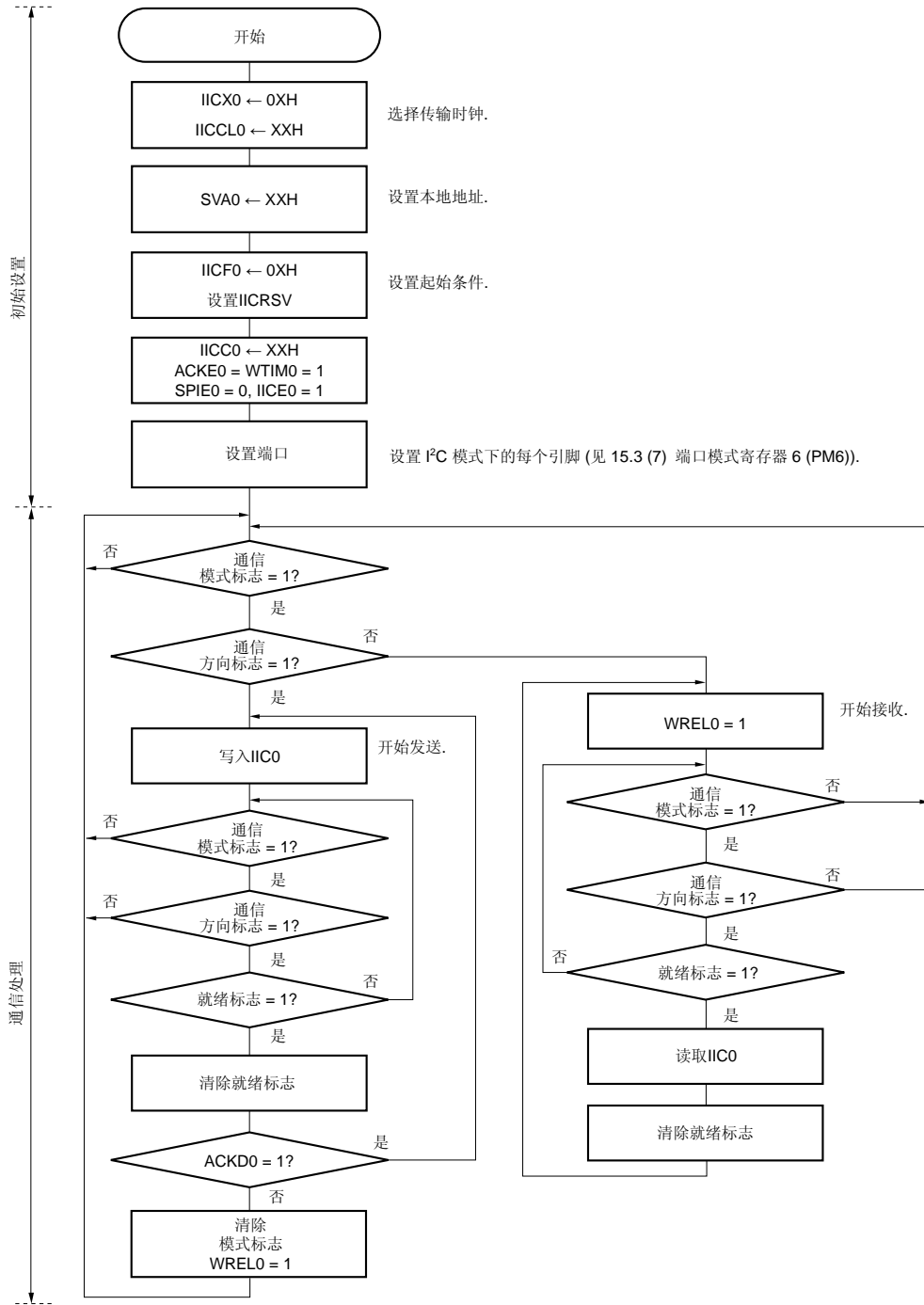
以下说明了从设备操作的主程序处理。

启动串行接口 IIC0 并且一直等待至允许通信。当允许通信时，将通过使用通信模式标志和就绪标志来执行通信（停止条件和起始条件的处理可以通过中断来完成。此时，可以通过使用标志来检查状态）。

传输操作被重复直至主设备不再返回 $\overline{\text{ACK}}$ 。如果 $\overline{\text{ACK}}$ 不再从主设备中返回，则通信完成。

对于接收来说，将会接收到一定的数据量。当通信完成时， $\overline{\text{ACK}}$ 不会作为下一个数据返回。之后，主设备将产生一个停止条件或重启条件。然后退出此时所产生的通信状态。

图 15-25. 从设备操作流程图 (1)



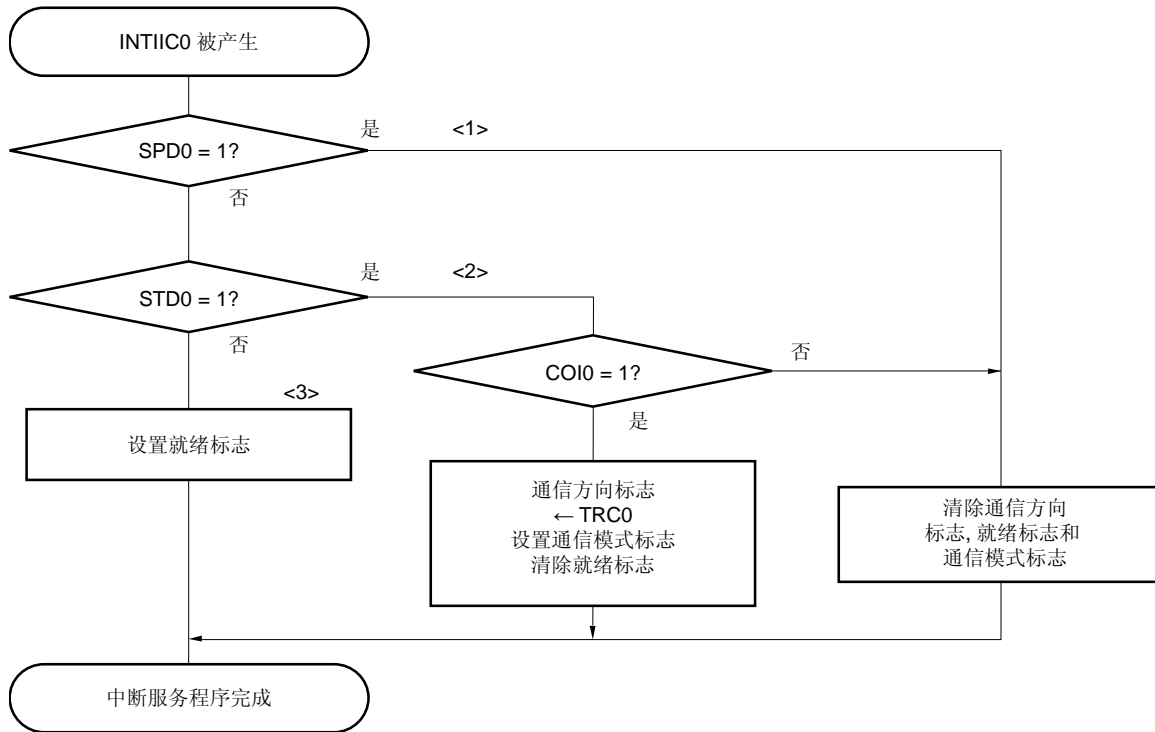
备注 至于传输和接收格式则应遵守正在通信的产品规范。

以下说明了带有 INTIIC0 中断的从设备的处理程序示例（处理中假设不使用扩展码）。INTIIC0 中断检查状态，并执行以下操作。

- <1> 如果发出停止条件，将会停止通信。
- <2> 如果发出起始条件，将会检查地址，如果地址不匹配则结束通信。如果地址匹配，则将设置通信模式并取消等待，而处理也将从中断中返回（就绪标志清零）。
- <3> 对于数据发送 / 接收来说，只会对就绪标志进行设置。处理会通过保持为等待状态的 I²C 总线从中断中返回。

备注 上述的<1>到<3>相应于图 15-26 从设备操作流程图中（2）中的<1>到<3>。

图 15-26. 从设备操作流程图中（2）



15.5.17 I²C中断请求 (INTIIC0) 发生的时序

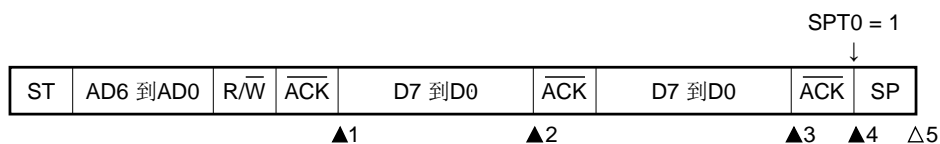
以下显示了当产生 INTIIC0 信号时传输或接收数据的时序、及 INTIIC0 信号产生时 IICS0 寄存器的值。

备注	ST:	起始条件
	AD6 到 AD0:	地址
	R / \overline{W} :	传输方向指示
	\overline{ACK} :	响应
	D7 到 D0:	数据
	SP:	停止条件

(1) 主设备操作

(a) 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止 (发送 / 接收)

(i) 当 **WTIM0 = 0** 时



- ▲1: IICS0 = 1000x110B
- ▲2: IICS0 = 1000x000B
- ▲3: IICS0 = 1000x000B (将 WTIM0 设置为 1) ^注
- ▲4: IICS0 = 1000xx00B (将 SPT0 设置为 1)
- Δ5: IICS0 = 00000001B

注 设置 WTIM0 为 1 并改变产生 INTIIC0 中断请求信号的时序，可以产生一个停止条件。

备注 ▲: 总是产生的
 Δ: 仅当 SPIE0=1 时产生
 x: 无关

(ii) 当 **WTIM0 = 1** 时

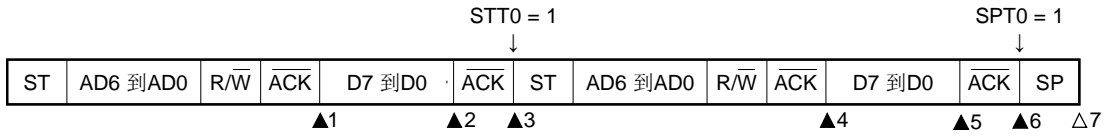


- ▲1: IICS0 = 1000x110B
- ▲2: IICS0 = 1000x100B
- ▲3: IICS0 = 1000xx00B (将 SPT0 设置为 1)
- Δ4: IICS0 = 00000001B

备注 ▲: 总是产生的
 Δ: 仅当 SPIE0=1 时产生
 x: 无关

(b) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止 (重启)

(i) 当 **WTIM0 = 0** 时

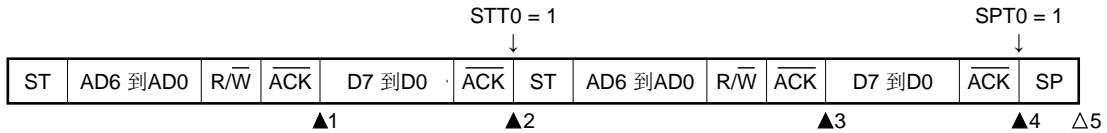


- ▲1: IICS0 = 1000x10B
- ▲2: IICS0 = 1000x000B (将 **WTIM0** 设置为 1) ^{注1}
- ▲3: IICS0 = 1000xx00B (将 **WTIM0** 清 0 ^{注2}, 将 **STT0** 设置为 1)
- ▲4: IICS0 = 1000x110B
- ▲5: IICS0 = 1000x000B (将 **WTIM0** 设置为 1) ^{注3}
- ▲6: IICS0 = 1000xx00B (将 **SPT0** 设置为 1)
- △7: IICS0 = 00000001B

- 注
1. 要产生一个起始条件, 应将 **WTIM0** 设置为 1 并改变产生 **INTIIC0** 中断请求信号的时序。
 2. 将 **WTIM0** 清 0 以恢复原始设置。
 3. 为了产生一个停止条件, 应将 **WTIM0** 设置为 1 并改变产生 **INTIIC0** 中断请求信号的时序。

- 备注
- ▲: 总是产生的
 - △: 仅当 **SPIE0=1** 时产生
 - x: 无关

(ii) 当 **WTIM0 = 1** 时

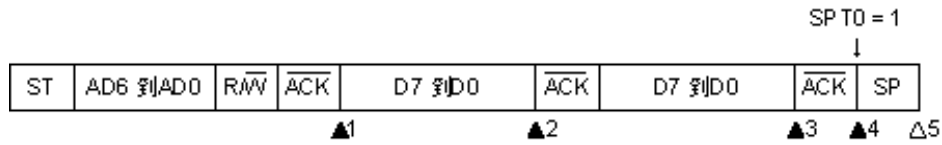


- ▲1: IICS0 = 1000x110B
- ▲2: IICS0 = 1000xx00B (将 **STT0** 设置为 1 时)
- ▲3: IICS0 = 1000x110B
- ▲4: IICS0 = 1000xx00B (将 **SPT0** 设置为 1)
- △5: IICS0 = 00000001B

- 备注
- ▲: 总是产生的
 - △: 仅当 **SPIE0=1** 时产生
 - x: 无关

(c) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止 (扩展码发送)

(i) 当 $WTIMO = 0$ 时



- ▲1: IIC0 = 1010x110B
- ▲2: IIC0 = 1010x000B
- ▲3: IIC0 = 1010x000B (将 $WTIMO$ 设置为 1) 注
- ▲4: IIC0 = 1010xx00B (将 $SPT0$ 设置为 1)
- Δ5: IIC0 = 00000001B

注 将 $WTIMO$ 设置为 1 并改变产生 $INTIIC0$ 中断请求信号的时序, 可产生一个停止条件。

备注 ▲: 总是产生的
 Δ: 仅当 $SPIE0=1$ 时产生
 x: 无关

(ii) 当 $WTIMO = 1$ 时



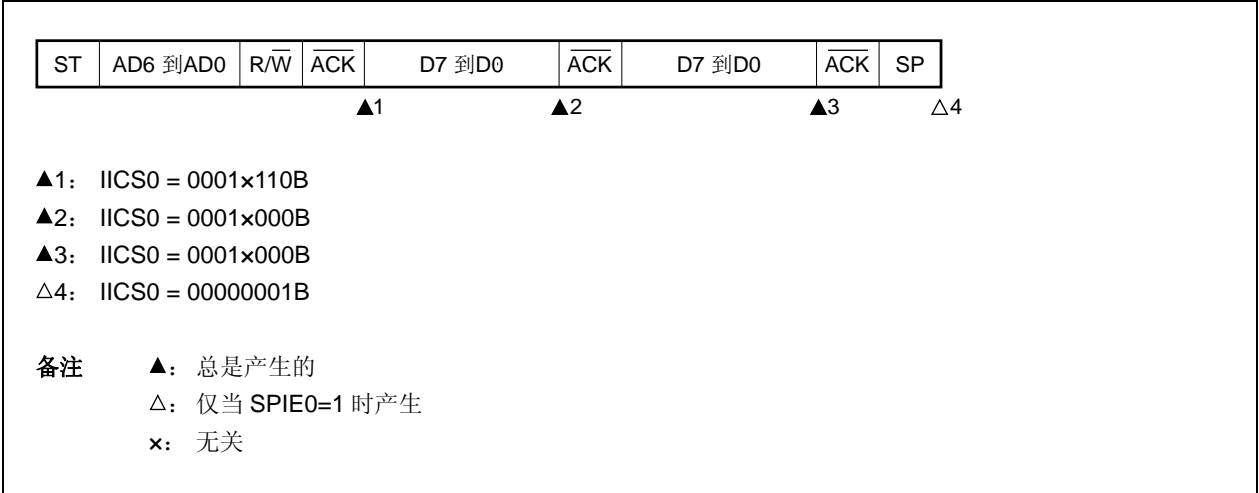
- ▲1: IIC0 = 1010x110B
- ▲2: IIC0 = 1010x100B
- ▲3: IIC0 = 1010xx00B (将 $SPT0$ 设置为 1)
- Δ4: IIC0 = 00001001B

备注 ▲: 总是产生的
 Δ: 仅当 $SPIE0=1$ 时产生
 x: 无关

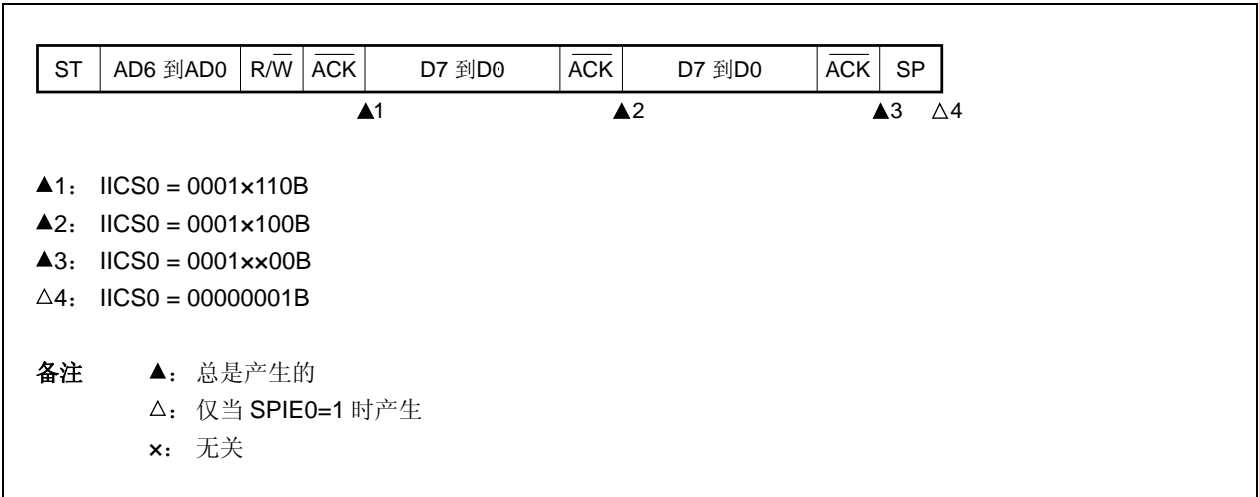
(2) 从设备操作 (从设备地址数据接收)

(a) 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止

(i) 当 **WTIM0 = 0** 时

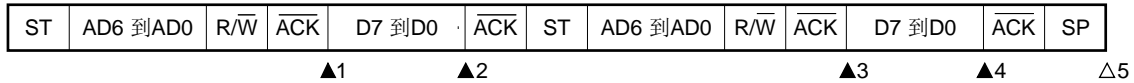


(ii) 当 **WTIM0 = 1** 时



(b) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

(i) 当 $WTIMO = 0$ 时 (重启后, 与 $SVA0$ 匹配)



▲1: IICSO = 0001x110B

▲2: IICSO = 0001x000B

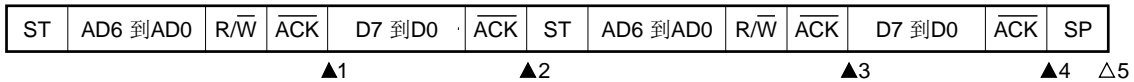
▲3: IICSO = 0001x110B

▲4: IICSO = 0001x000B

△5: IICSO = 00000001B

备注 ▲: 总是产生的
 △: 仅当 $SPIE0=1$ 时产生
 x: 无关

(ii) 当 $WTIMO = 1$ 时 (重启后, 与 $SVA0$ 匹配)



▲1: IICSO = 0001x110B

▲2: IICSO = 0001xx00B

▲3: IICSO = 0001x110B

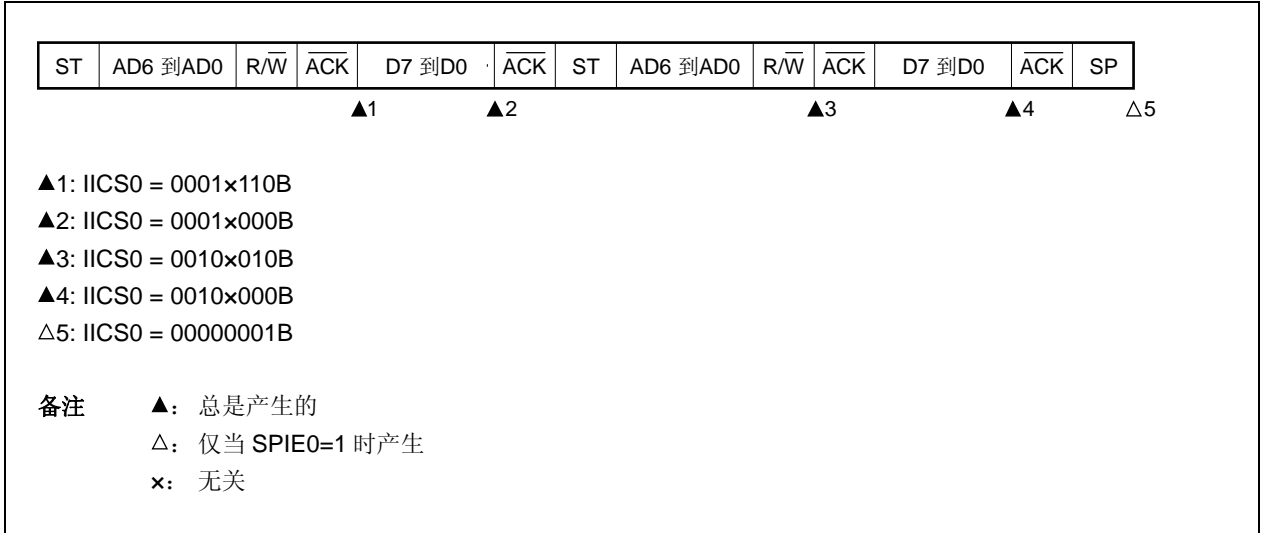
▲4: IICSO = 0001xx00B

△5: IICSO = 00000001B

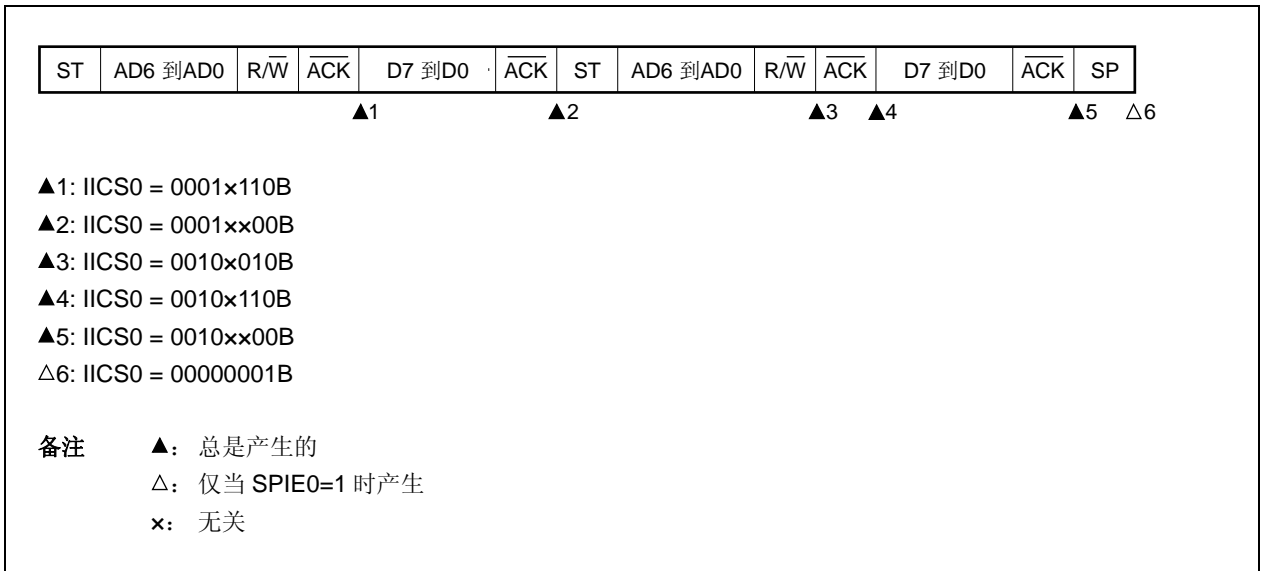
备注 ▲: 总是产生的
 △: 仅当 $SPIE0=1$ 时产生
 x: 无关

(c) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 代码 ~ 数据 ~ 停止

(i) 当 $WTIMO = 0$ 时 (重启后, 与地址不匹配 (=扩展码))

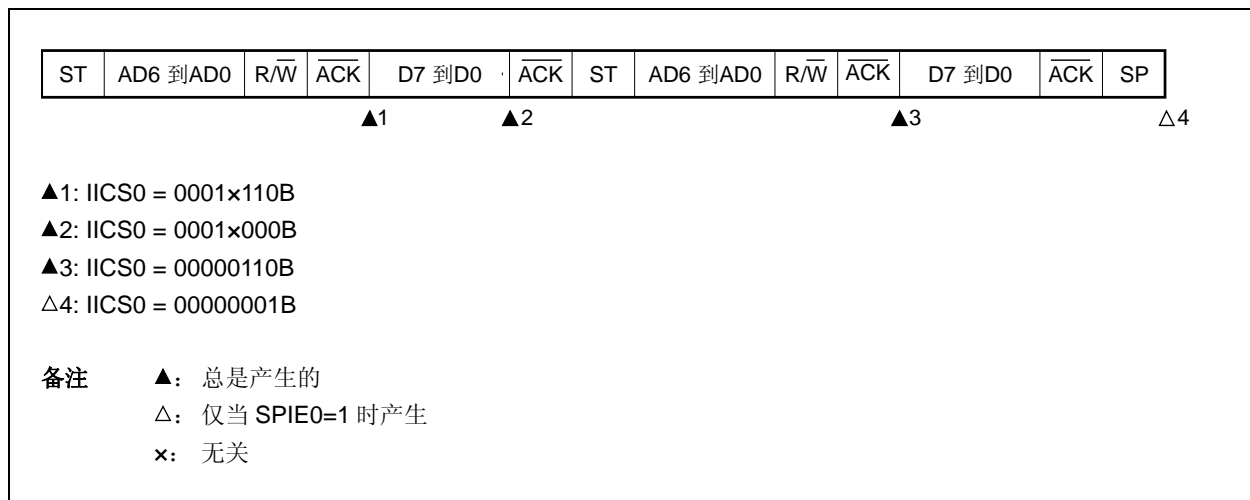


(ii) 当 $WTIMO = 1$ 时 (重启后, 与地址不匹配 (=扩展码))

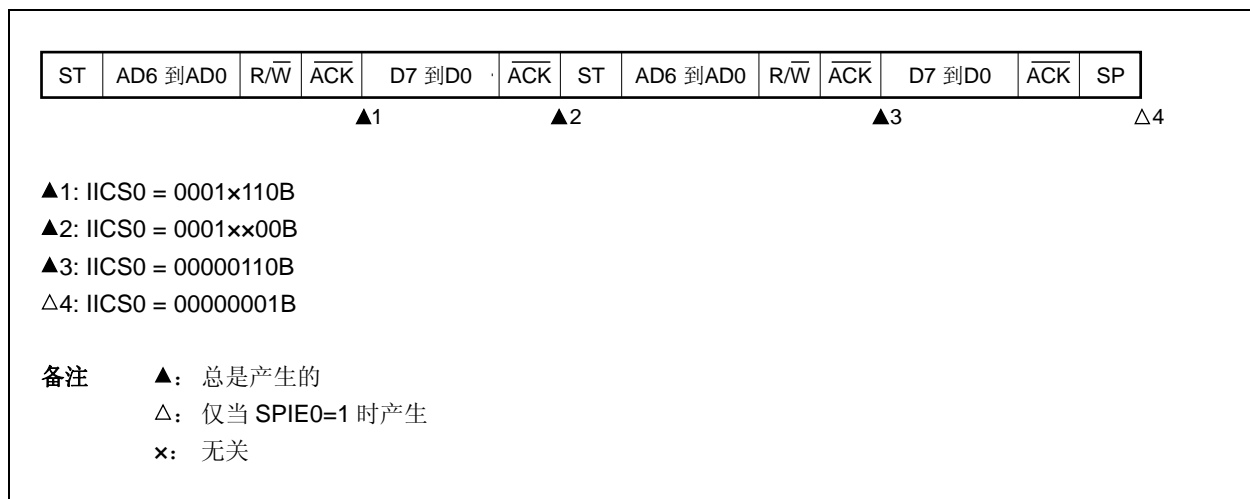


(d) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

(i) 当 **WTIMO = 0** 时 (重启后, 与地址不匹配 (=非扩展码))



(ii) 当 **WTIMO = 1** 时 (重启后, 与地址不匹配 (=非扩展码))

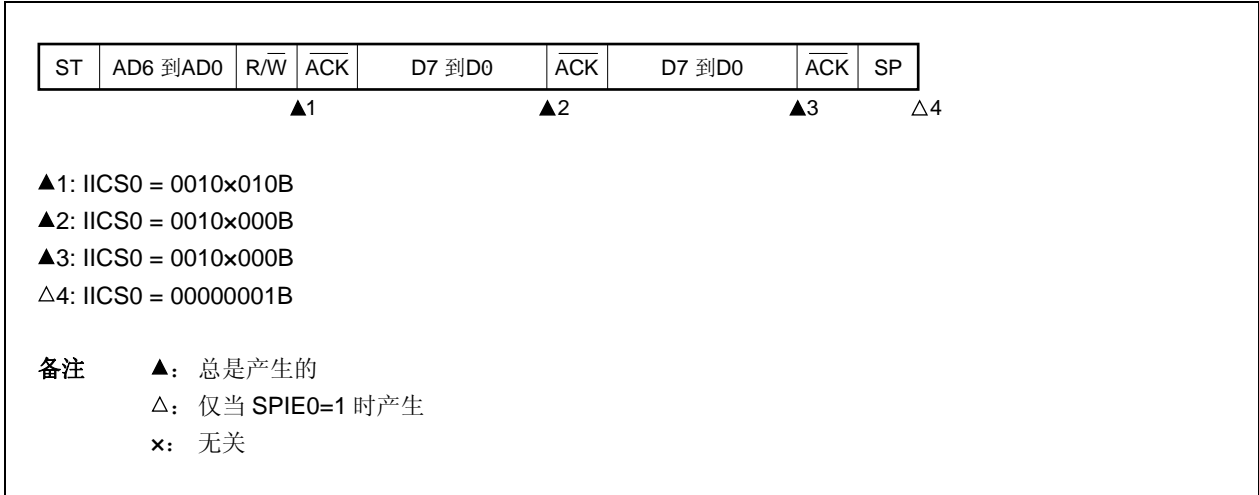


(3) 从设备操作（接收扩展码时）

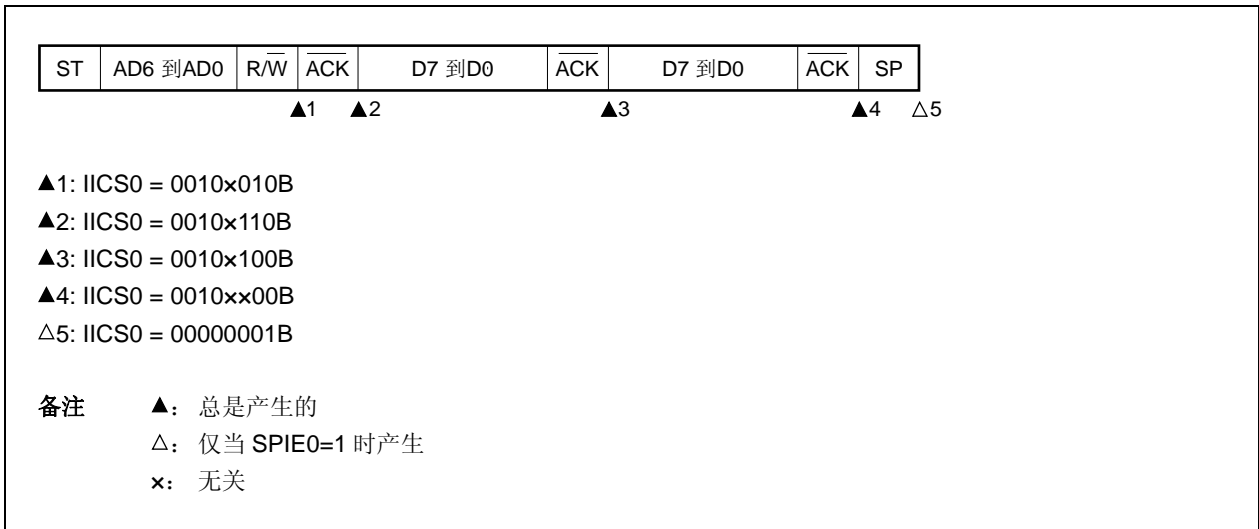
当设备接收到扩展码时，它始终参与通信。

(a) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止

(i) 当 **WTIMO = 0** 时

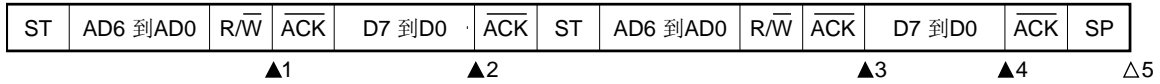


(ii) 当 **WTIMO = 1** 时



(b) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

(i) 当 **WTIM0 = 0** 时 (重启后, 与 **SVA0** 匹配)



▲1: IICS0 = 0010x010B

▲2: IICS0 = 0010x000B

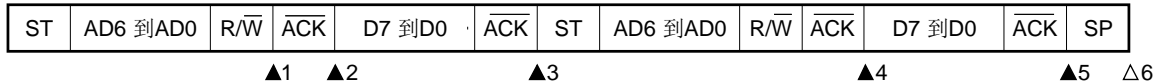
▲3: IICS0 = 0001x110B

▲4: IICS0 = 0001x000B

△5: IICS0 = 00000001B

备注 ▲: 总是产生的
 △: 仅当 SPIE0=1 时产生
 x: 无关

(ii) 当 **WTIM0 = 1** 时 (重启后, 与 **SVA0** 匹配)



▲1: IICS0 = 0010x010B

▲2: IICS0 = 0010x110B

▲3: IICS0 = 0010xx00B

▲4: IICS0 = 0001x110B

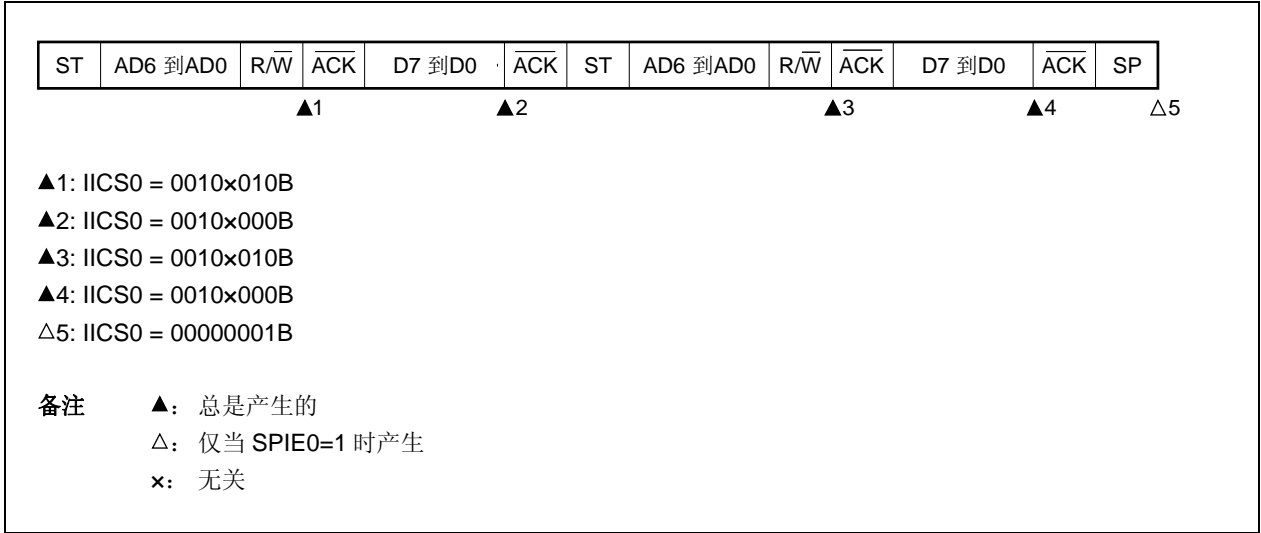
▲5: IICS0 = 0001xx00B

△6: IICS0 = 00000001B

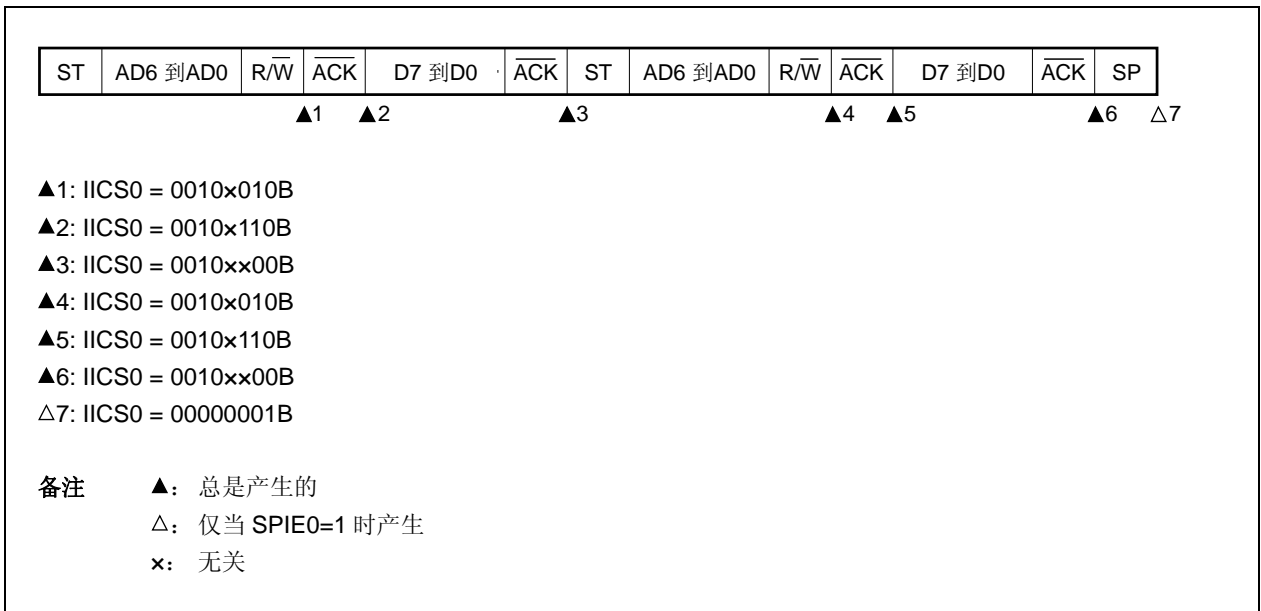
备注 ▲: 总是产生的
 △: 仅当 SPIE0=1 时产生
 x: 无关

(c) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 代码 ~ 数据 ~ 停止

(i) 当 **WTIMO = 0** 时 (重启后, 接收扩展码)

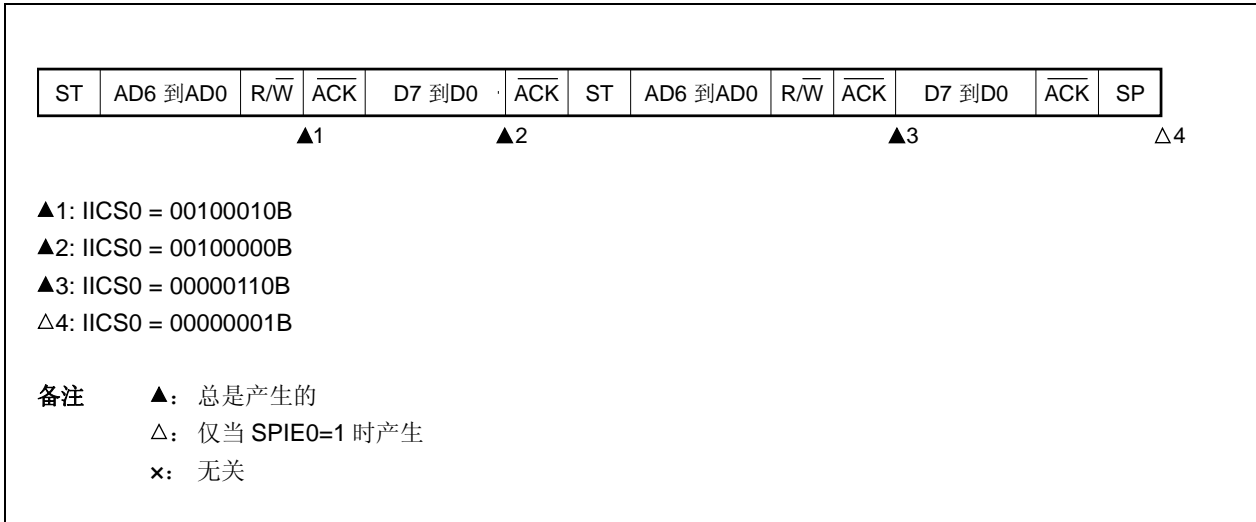


(ii) 当 **WTIMO = 1** 时 (重启后, 接收扩展码)

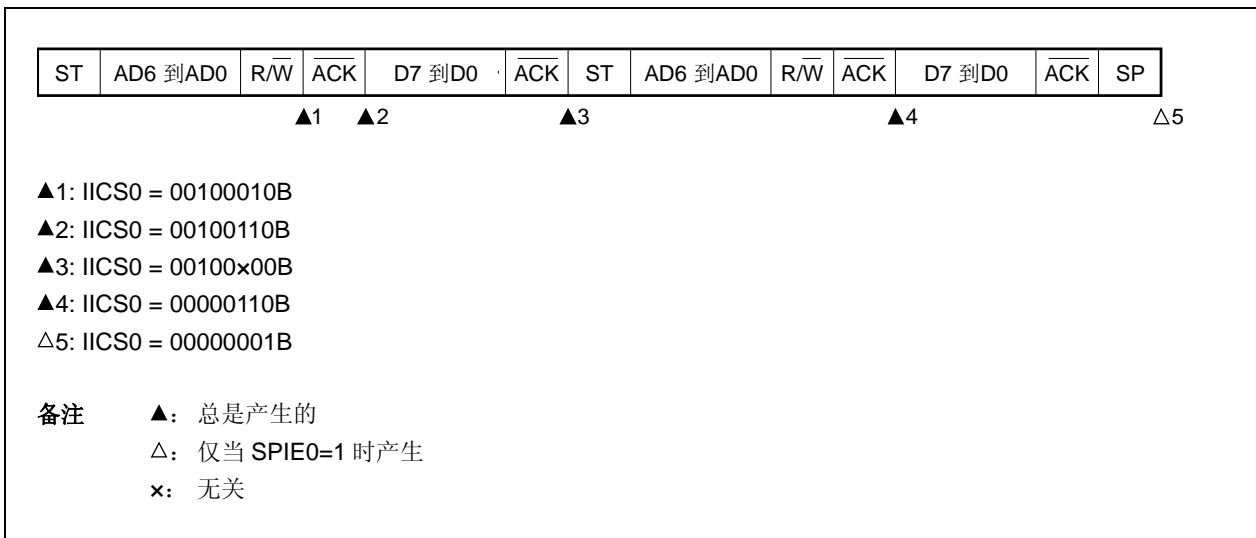


(d) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

(i) 当 **WTIMO = 0** 时 (重启后, 地址不匹配 (=非扩展码))

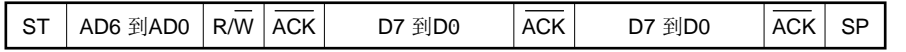


(ii) 当 **WTIMO = 1** 时 (重启后, 地址不匹配 (=非扩展码))



(4) 没有通信的操作

(a) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止



△1

△1: IICS0 = 00000001B

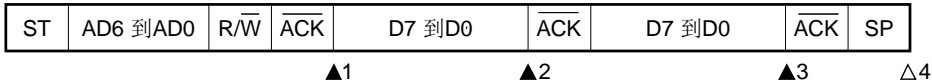
备注 △: 仅当 SPIE0=1 时产生

(5) 仲裁失败操作 (仲裁失败后作为从设备进行操作)

当设备在多主控制系统中用作主设备时, 每次发生中断请求信号 INTIIC0 时读取 MSTSO 位以检查仲裁结果。

(a) 当在从设备地址数据传输期间发生仲裁失败

(i) 当 WTIMO = 0 时



▲1: IICS0 = 0101x110B

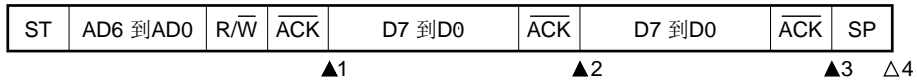
▲2: IICS0 = 0001x000B

▲3: IICS0 = 0001x000B

△4: IICS0 = 00000001B

备注 ▲: 总是产生的
 △: 仅当 SPIE0=1 时产生
 x: 无关

(ii) 当 **WTIMO = 1** 时



▲1: IICS0 = 0101x110B

▲2: IICS0 = 0001x100B

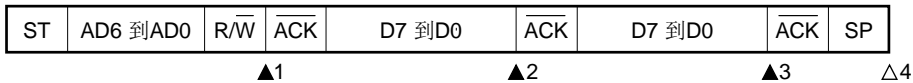
▲3: IICS0 = 0001xx00B

△4: IICS0 = 00000001B

备注 ▲: 总是产生的
 △: 仅当 SPIE0=1 时产生
 x: 无关

(b) 当在扩展码的传输期间发生仲裁失败时

(i) 当 **WTIMO = 0** 时



▲1: IICS0 = 0110x010B

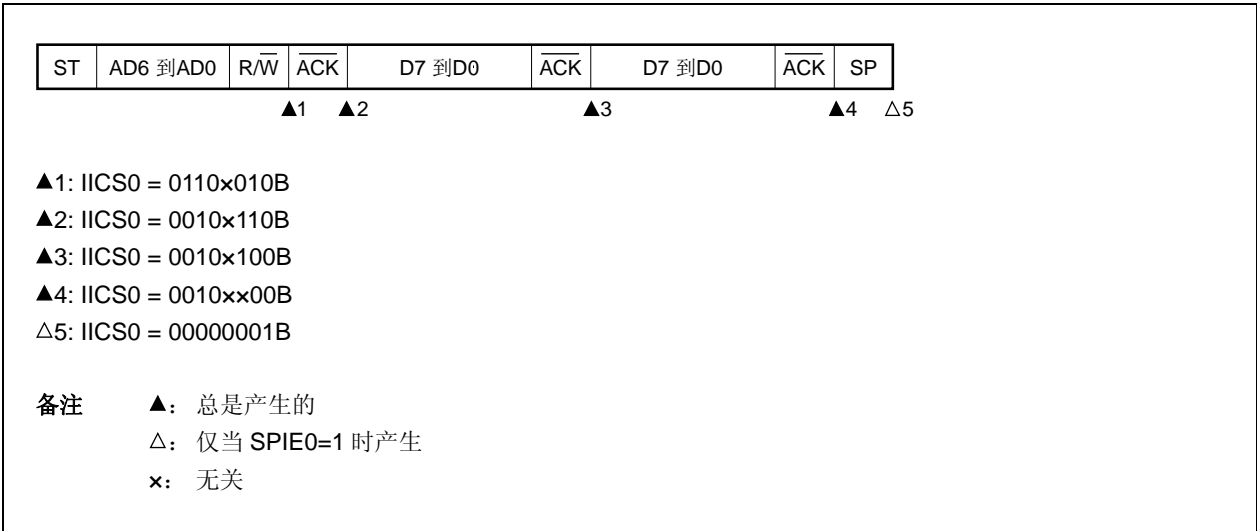
▲2: IICS0 = 0010x000B

▲3: IICS0 = 0010x000B

△4: IICS0 = 00000001B

备注 ▲: 总是产生的
 △: 仅当 SPIE0=1 时产生
 x: 无关

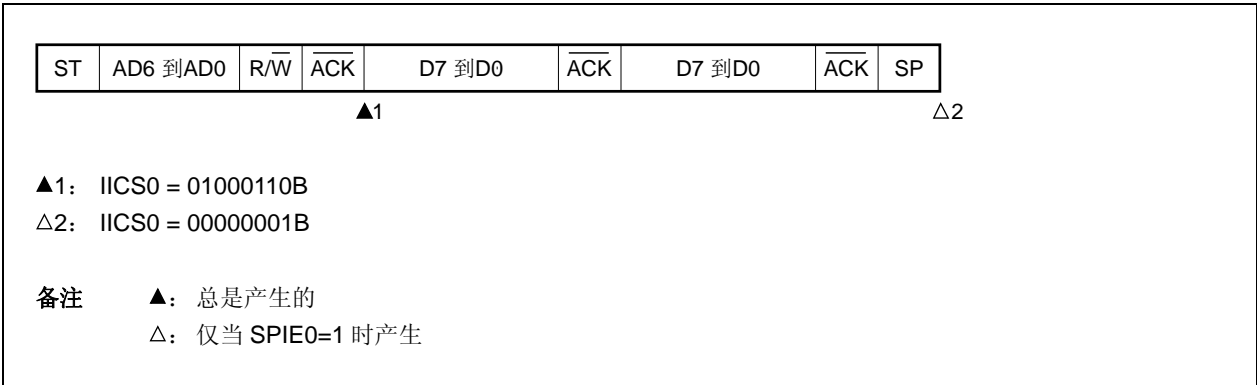
(ii) 当 **WTIMO = 1** 时



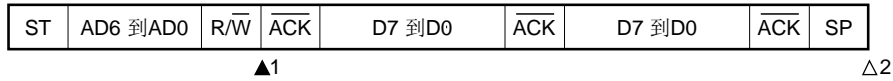
(6) 仲裁失败发生时的操作（仲裁失败后没有通信）

当设备在多主控制系统中被用作主设备时，每次发生中断请求信号 INTIIC0 时读取 MSTS0 位以检查仲裁结果。

(a) 从设备地址数据的传输期间发生仲裁失败（当 **WTIMO=1** 时）



(b) 当在扩展码的传输期间发生仲裁失败时



▲1: IIC0 = 0110x010B

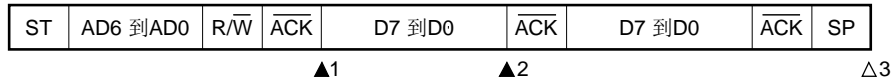
通过软件来设置 LREL0 = 1

△2: IIC0 = 00000001B

备注 ▲: 总是产生的
 △: 仅当 SPIE0=1 时产生
 x: 无关

(c) 当在数据传输期间发生仲裁失败时

(i) 当 WTIMO = 0 时



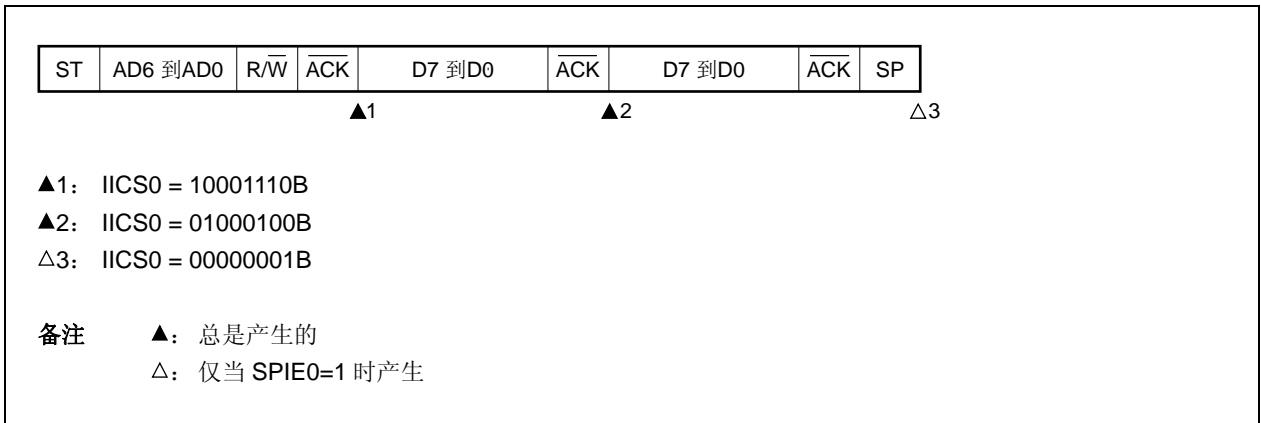
▲1: IIC0 = 10001110B

▲2: IIC0 = 01000000B

△3: IIC0 = 00000001B

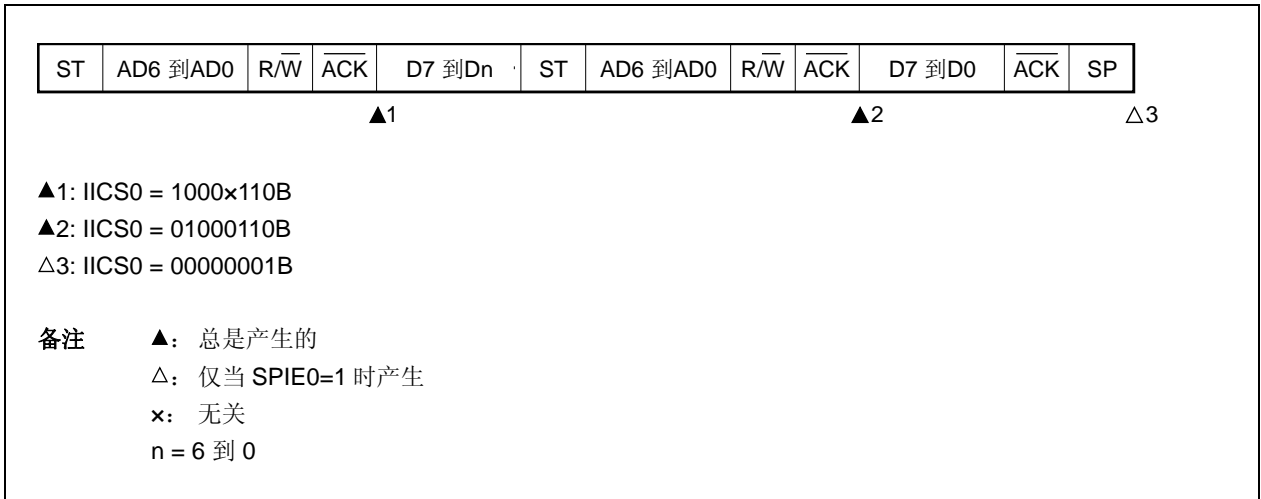
备注 ▲: 总是产生的
 △: 仅当 SPIE0=1 时产生

(ii) 当 **WTIMO = 1** 时

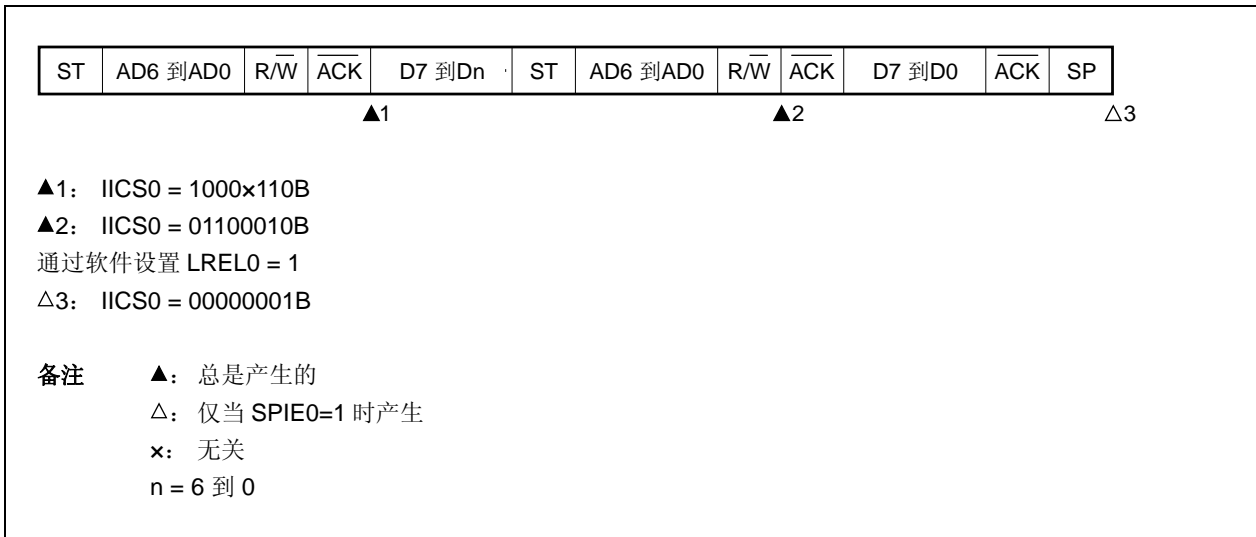


(d) 当在数据传输期间由于重启条件导致仲裁失败时

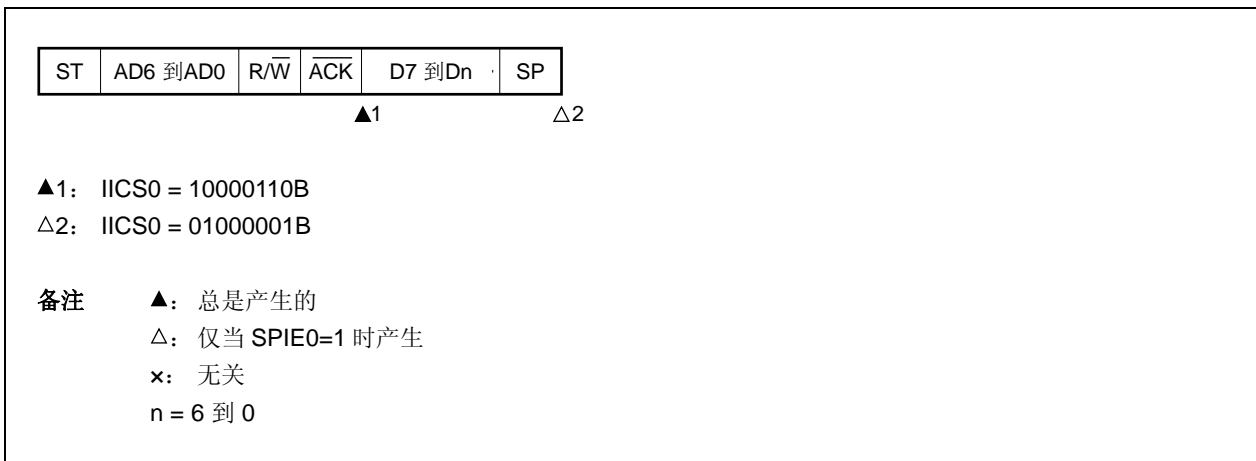
(i) 非扩展码 (例: 与 **SVA0** 不匹配)



(ii) 扩展码

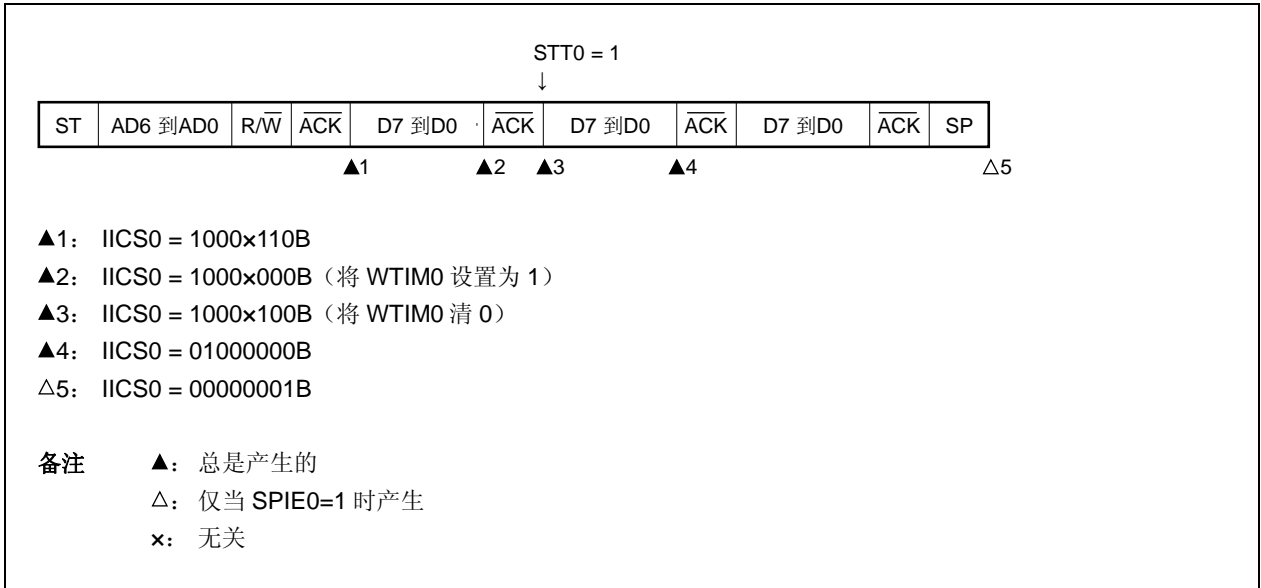


(e) 当在数据传输期间由于停止条件导致仲裁失败时

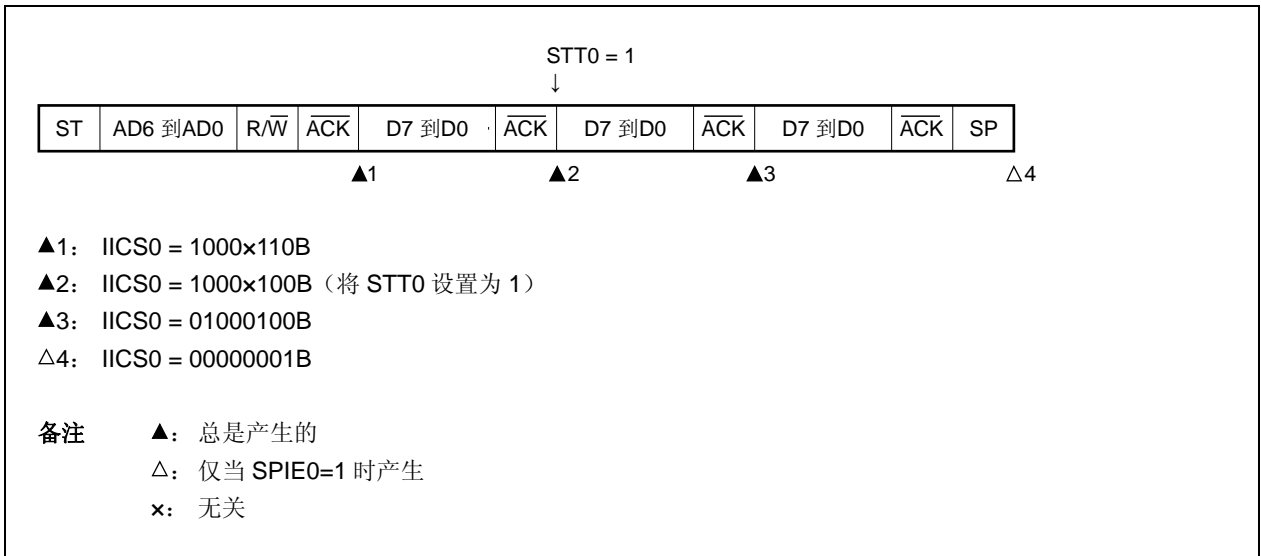


(f) 在试图产生一个重启条件时由于低电平数据导致仲裁失败时

(i) 当 $WTIM0 = 0$ 时

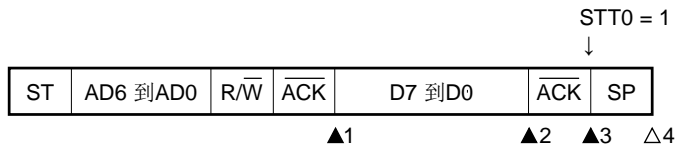


(ii) 当 $WTIM0 = 1$ 时



(g) 当试图产生一个重启条件时由于停止条件导致仲裁失败时

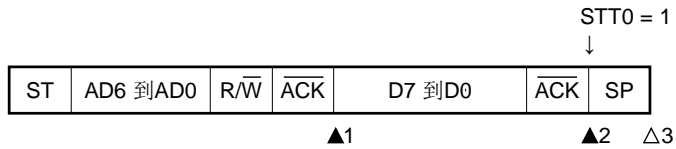
(i) 当 $WTIM0 = 0$ 时



- ▲1: IICS0 = 1000x110B
- ▲2: IICS0 = 1000x000B (将 $WTIM0$ 设置为 1)
- ▲3: IICS0 = 1000xx00B (将 $STT0$ 设置为 1 时)
- Δ4: IICS0 = 01000001B

备注 ▲: 总是产生的
 Δ: 仅当 $SPIE0=1$ 时产生
 x: 无关

(ii) 当 $WTIM0 = 1$ 时

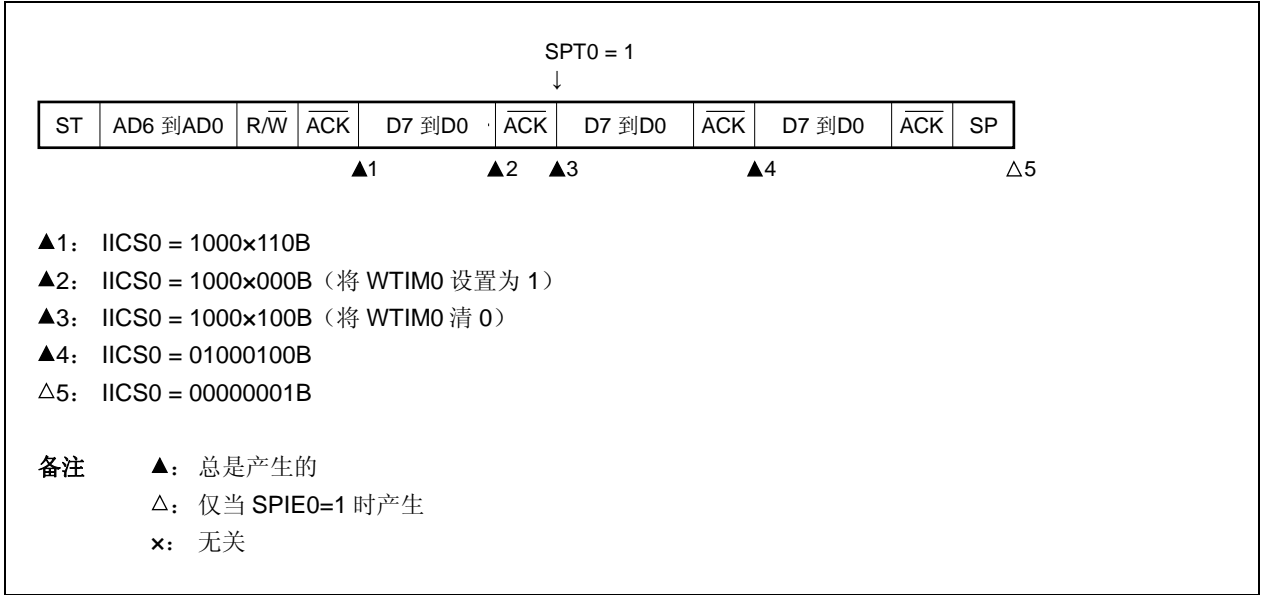


- ▲1: IICS0 = 1000x110B
- ▲2: IICS0 = 1000xx00B (将 $STT0$ 设置为 1 时)
- Δ3: IICS0 = 01000001B

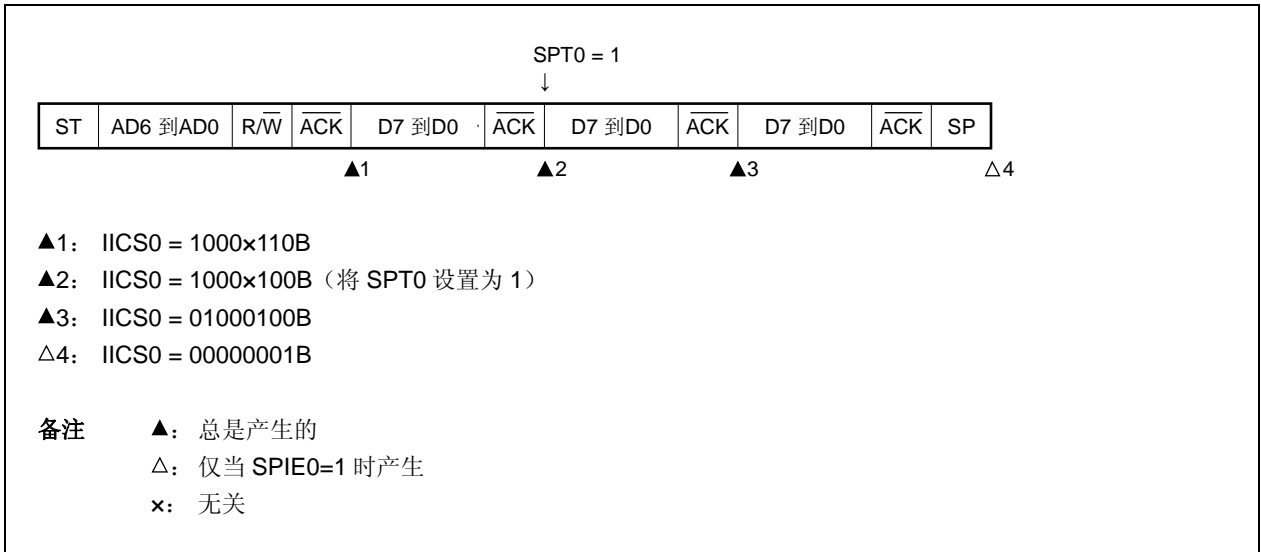
备注 ▲: 总是产生的
 Δ: 仅当 $SPIE0=1$ 时产生
 x: 无关

(h) 当试图产生一个停止条件时由于低电平数据导致仲裁失败时

(i) 当 **WTIMO = 0** 时



(ii) 当 **WTIMO = 1** 时



15.6 时序图

当使用 I²C 总线模式时，主设备会通过串行总线来输出一个地址以从多个从设备中选择一个作为它的通信方。

输出从设备地址后，主设备将会传输指定数据传输方向的 TRC0 位（IIC 状态寄存器 0（IICS0）的第 3 位），然后开始与从设备进行串行通信。

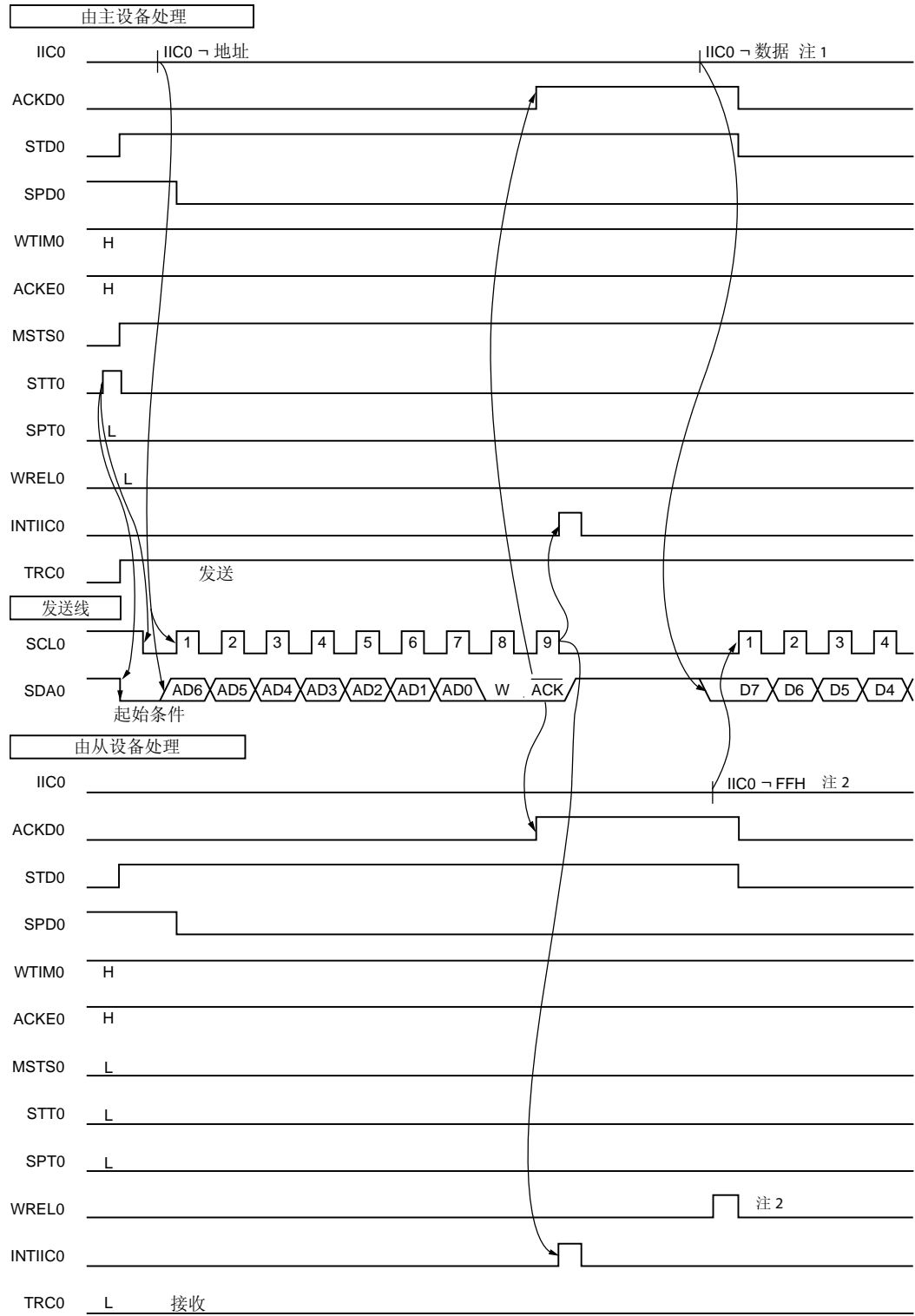
图 15-27 和 15-28 显示了数据通信的时序图。

IIC 移位寄存器 0（IIC0）的移位操作与串行时钟（SCL0）的下降沿同步。传输数据被传输到 SO0 锁存器中并通过 SDA0 引脚被输出（MSB 先输出）

在 SCL0 的上升沿将 SDA0 引脚输入的数据捕捉到 IIC0 中。

图 15-27. 主设备到从设备的通信示例
(当主设备和从设备选择第 9 时钟等待时) (1 / 3)

(1) 起始条件 ~ 地址

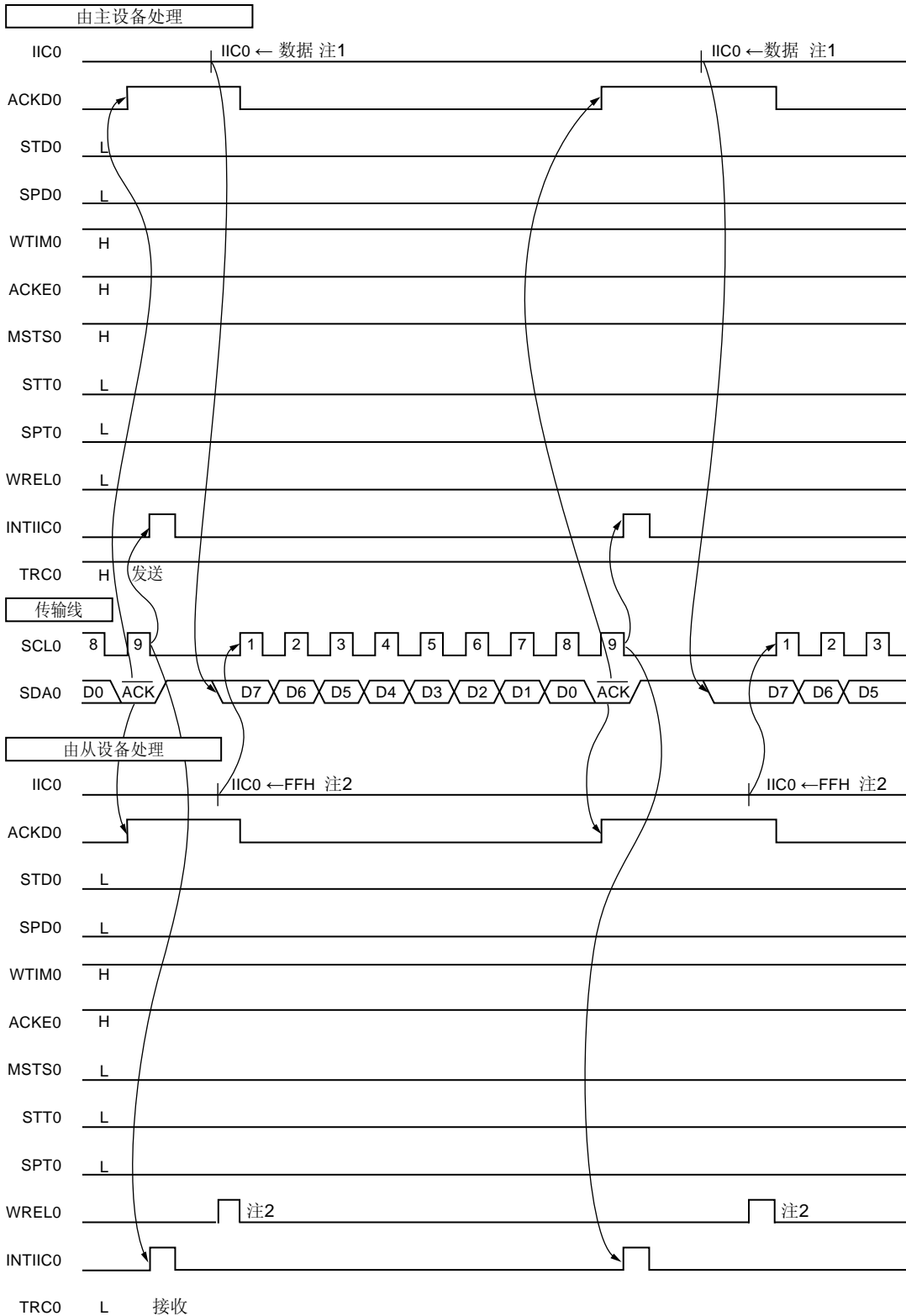


注 1. 要取消主发送期间的状态等待, 需向 IIC0 写入数据, 不要设置 WRELO。

2. 要取消从等待, 需向 IIC0 写入 “FFH” 或设置 WRELO。

图 15-27. 主设备到从设备的通信示例
(当主设备和从设备选择第 9 时钟等待时) (2 / 3)

(2) 数据

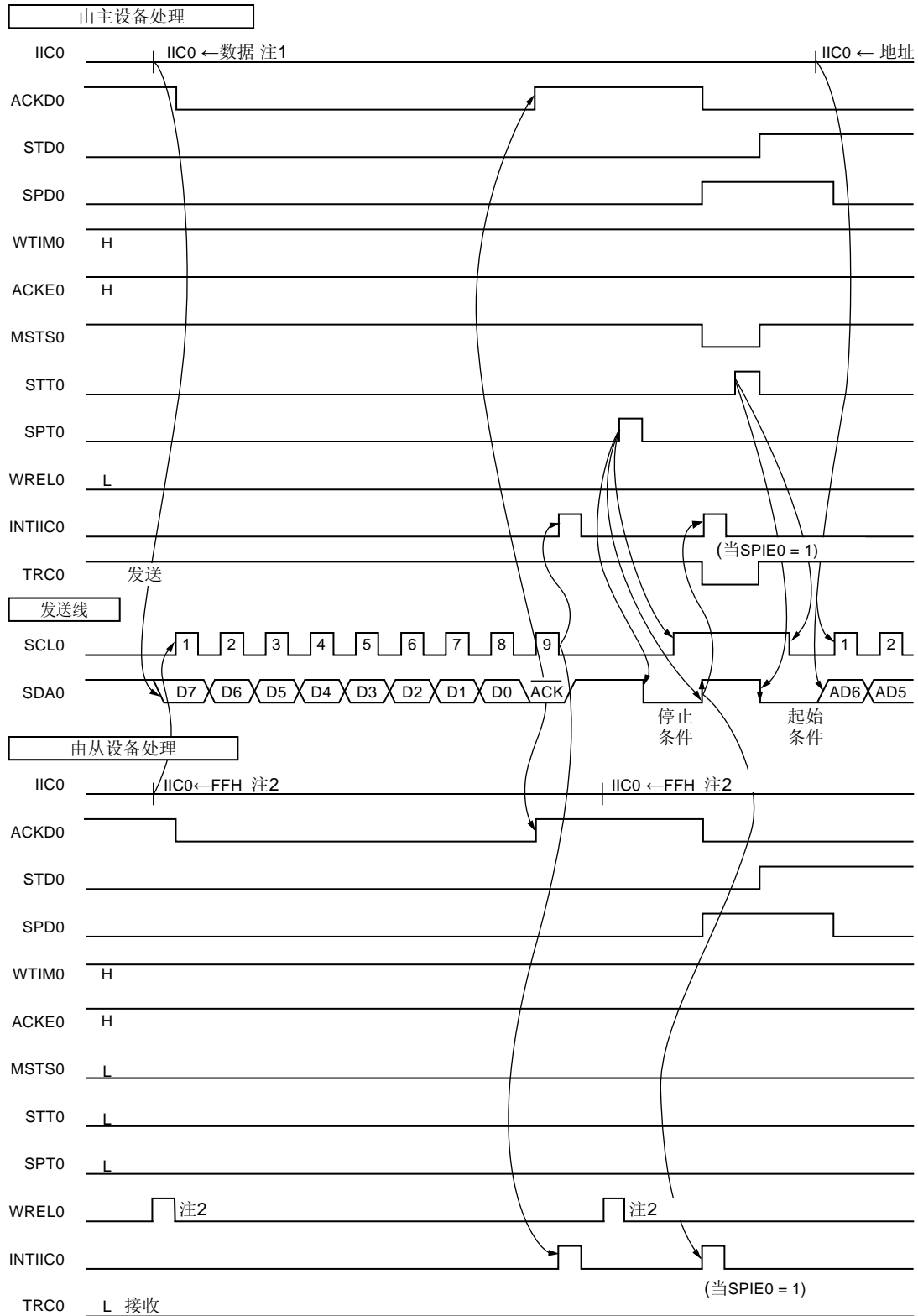


注 1. 要取消主发送期间的状态等待，需向 IIC0 写入数据，不要设置 WRELO。

2. 要取消从等待，需向 IIC0 写入“FFH”或设置 WRELO。

图 15-27. 主设备到从设备的通信示例
(当主设备和从设备选择第 9 时钟等待时) (3 / 3)

(3) 停止条件

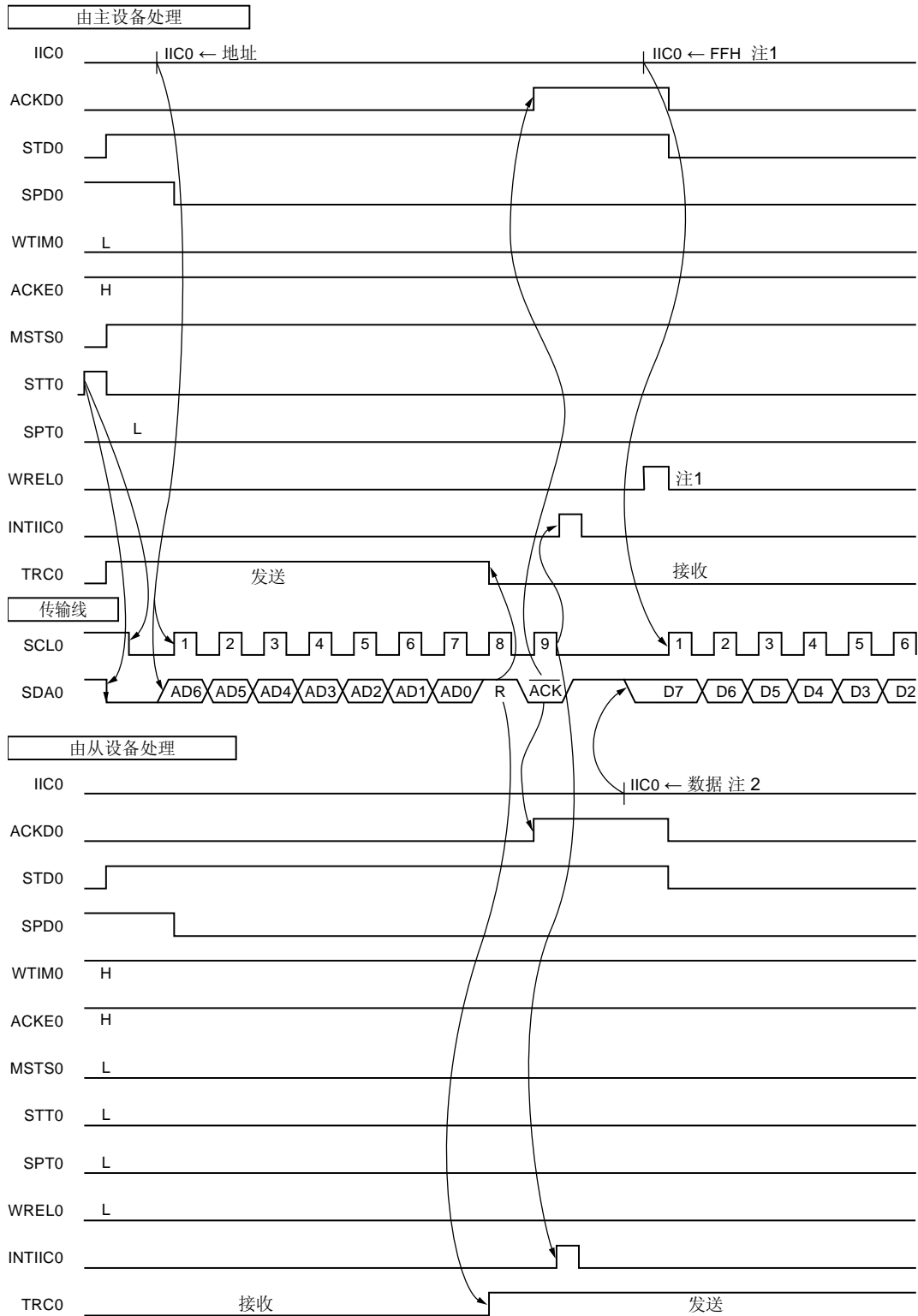


注 1. 要取消主发送期间的状态等待，需向 IIC0 写入数据，不要设置 WRELO。

2. 要取消从等待，需向 IIC0 写入“FFH”或设置 WRELO。

图 15-28. 从设备到主设备的通信示例
(当主设备选择 8 时钟等待, 而从设备选择第 9 时钟等待时) (1 / 3)

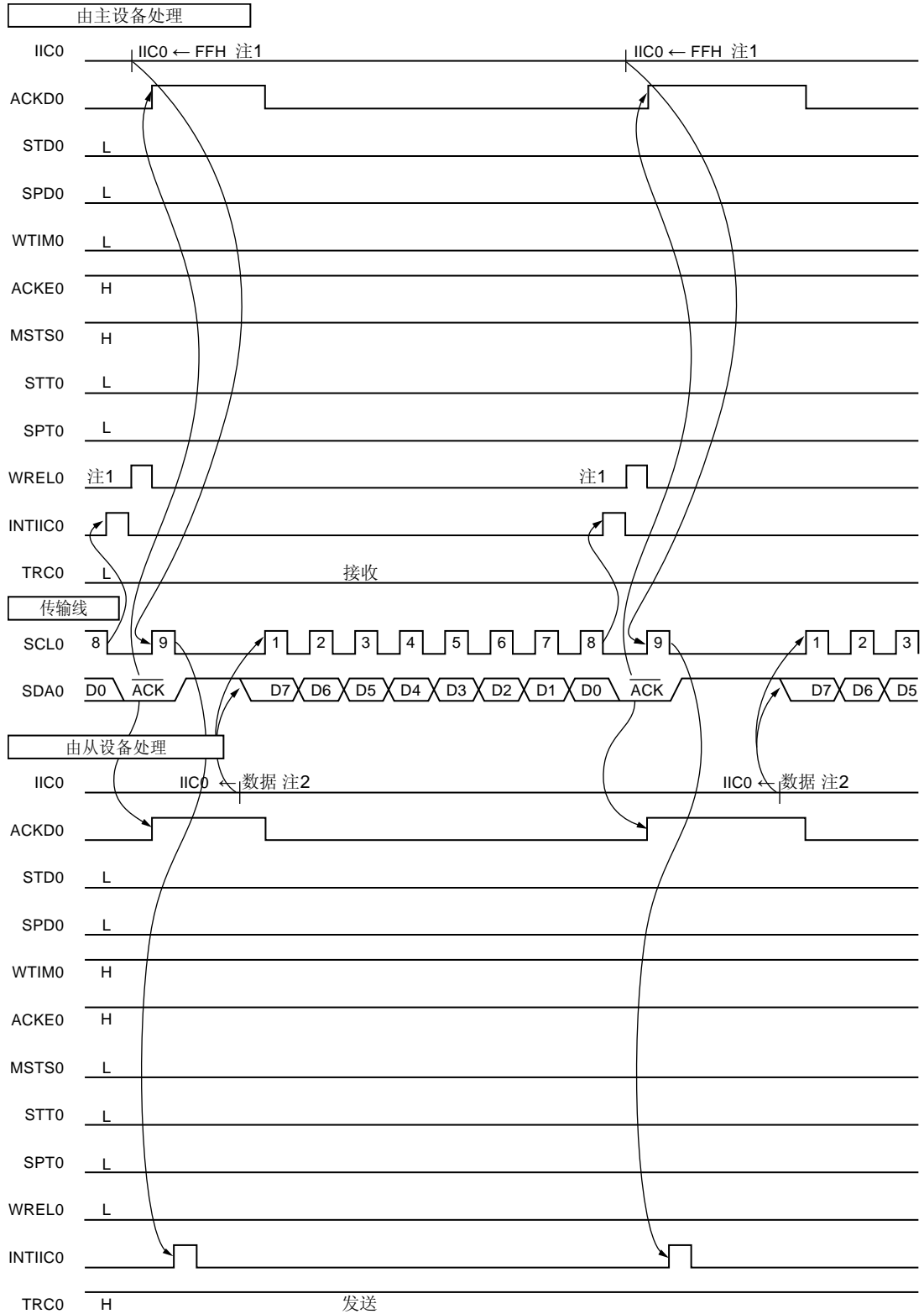
(1) 起始条件 ~ 地址



- 注
1. 要取消主等待, 需向 IIC0 写入“FFH”或设置 WRELO。
 2. 要取消从发送期间的等待状态, 需向 IIC0 写入数据, 不要设置 WRELO。

图 15-28. 从设备到主设备的通信示例
(当主设备选择第 8 时钟等待, 而从设备选择第 9 时钟等待时) (2 / 3)

(2) 数据

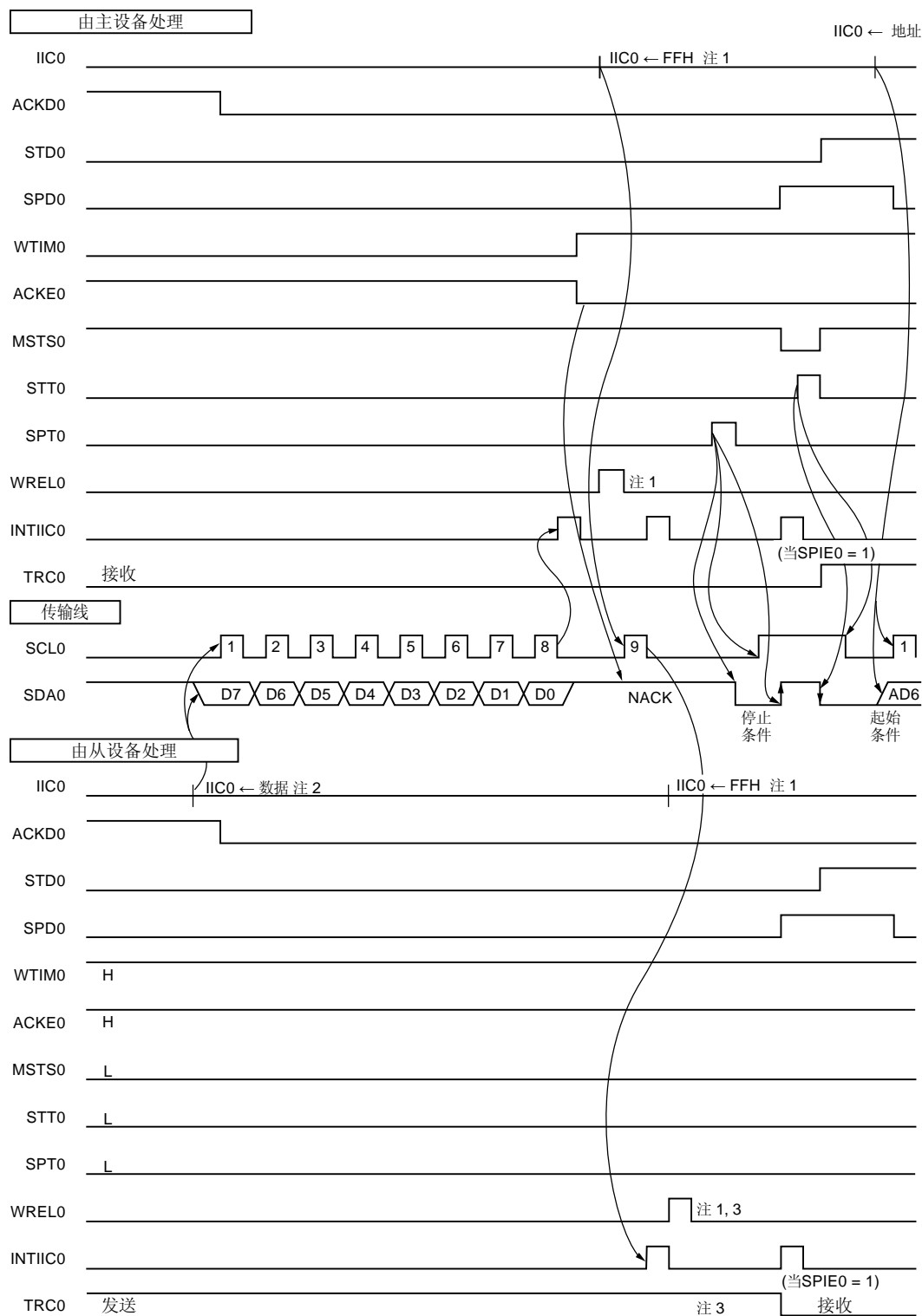


- 注 1. 要取消主等待, 需向 IIC0 写入“FFH”或设置 WRELO。
 注 2. 要取消从发送期间的等待状态, 需向 IIC0 写入数据, 不要设置 WRELO。

图 15-28. 从设备到主设备的通信示例

(当主设备选择第 8 时钟及第 9 时钟等待，而从设备选择第 9 时钟等待时) (3 / 3)

(3) 停止条件



- 注
1. 要取消等待，需向 IIC0 写入“FFH”或设置 WRELO。
 2. 要取消从发送期间的等待状态，需向 IIC0 写入数据，不要设置 WRELO。
 3. 如果通过设置 WRELO 取消从发送期间的等待状态，TRC0 将被清零。

第十六章 中断功能

16.1 中断功能类型

可以使用以下两种类型的中断功能。

(1) 可屏蔽中断

这类中断可进行屏蔽控制。通过设置优先级指定标志寄存器 (PROL, PROH, PR1L, PR1H) 将可屏蔽中断分为高中断优先级组和低中断优先级组。

当产生高优先级中断时, 中断嵌套可以应用于低优先级中断。如果同时产生两个或更多的具有同一优先级的中断请求, 则依照中断服务的优先级来处理。关于优先级次序, 请参照 **表 16-1**。

产生待机释放信号, 并且释放 STOP 以及 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: 6, 内部: 14

(2) 软件中断

这是通过执行 BRK 指令所产生的中断。即使当禁止中断时仍可以响应此类中断。软件中断不受中断优先级控制。

注意事项 P30 / INTP1 引脚在封装内连接到恒电流驱动部分的 SH 引脚。因此, 可以通过使用 CPU 部分的中断功能检查 SH 引脚的输出。

16.2 中断源和配置

μ PD78F8024 和 78F8025 一共有 21 个中断源, 其中包括可屏蔽中断和软件中断。此外最多还具有四个复位源 (参照 **表 16-1**)。

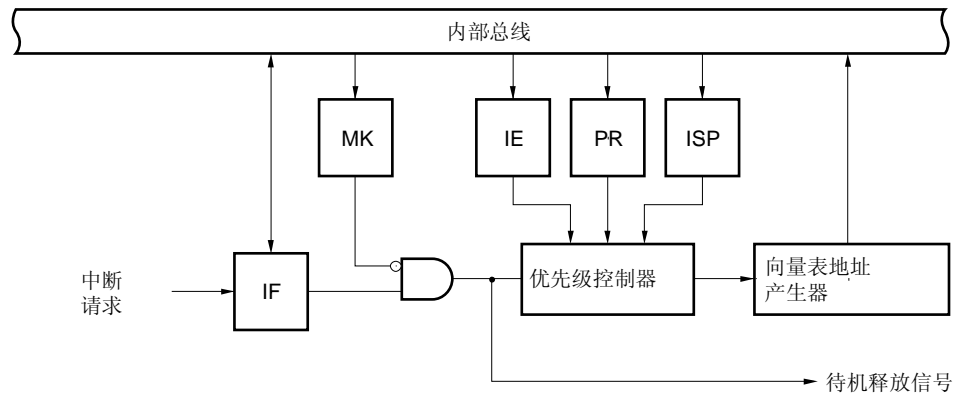
表 16-1. 中断源列表

中断类型	默认优先级 注1	中断源		内部 / 外部	向量表地址	基本结构类型注2
		名称	触发			
可屏蔽	0	INTLVI	低电压检测注3	内部	0004H	(A)
	1	INTP0	引脚输入脉冲边沿检测	外部	0006H	(B)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTSRE6	UART6 接收错误产生	内部	0012H	(A)
	8	INTSR6	UART6 接收结束		0014H	
	9	INTST6	UART6 传输结束		0016H	
	10	INTCSI10 / INTST0	CSI10 通信结束 / UART0 传输结束		0018H	
	11	INTTMH1	TMH1 和 CMP01 匹配 (指定为比较寄存器时)		001AH	
	12	INTTMH0	TMH0 和 CMP00 匹配 (指定为比较寄存器时)		001CH	
	13	INTTM50	TM50 和 CR50 匹配 (指定为比较寄存器时)		001EH	
	14	INTTM000	TM00 和 CR000 匹配 (指定为比较寄存器时) TI010 引脚有效沿检测 (指定为捕捉寄存器时)		0020H	
	15	INTTM010	TM00 和 CR010 匹配 (指定为比较寄存器时) TI000 引脚有效沿检测 (指定为捕捉寄存器时)		0022H	
	16	INTAD	A/D 转换结束		0024H	
	17	INTSR0	UART0 接收结束或接收错误产生		0026H	
	18	INTTM51注4	TM51 和 CR51 匹配 (指定为比较寄存器时)		002AH	
19	INTIIC0	IIC0 通信结束	0034H			
软件	-	BRK	BRK 指令执行	-	003EH	(D)
复位	-	RESET	RESET 引脚输入	-	0000H	-
		POC	上电清零			
		LVI	低电压检测注5			
		WDT	WDT 溢出			

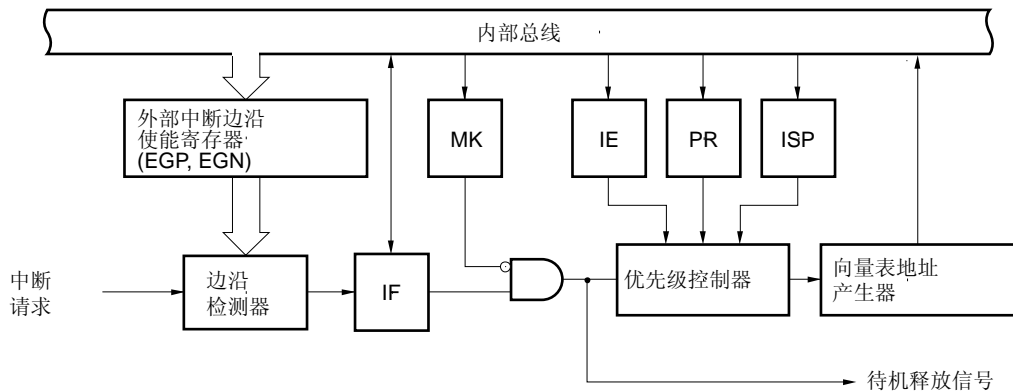
- 注
1. 如果同时产生两个或更多的可屏蔽中断，则采用默认优先级决定处理中断的顺序。0 表示最高优先级，而 19 则表示最低优先级。
 2. 基本结构类型 (A) 到 (C) 对应于图 16-1 中的 (A) 到 (C)。
 3. 当低压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 清 0 时。
 4. 当在载波发生模式下使用 8 位定时器 / 事件计数器 51 时，INTTM5H1 信号产生时产生中断 (参见图 9-13 传输时序)。
 5. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 设为 1 时。

图 16-1. 中断功能的基本结构 (1 / 2)

(A) 内部可屏蔽中断



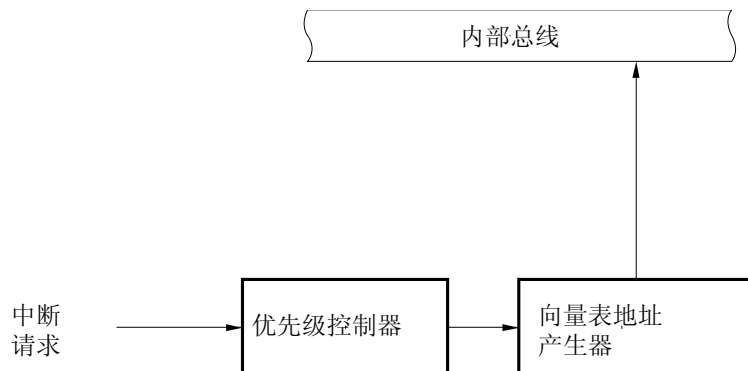
(B) 外部可屏蔽中断 (INTP0 到 INTP5)



- IF: 中断请求标志
- IE: 允许中断标志
- ISP: 使用中的优先级标志
- MK: 中断屏蔽标志
- PR: 优先级指定标志

图 16-1. 中断功能的基本结构 (2 / 2)

(C) 软件中断



- IF: 中断请求标志
- IE: 中断允许标志
- ISP: 使用中的优先级标志
- MK: 中断屏蔽标志
- PR: 优先级指定标志

16.3 控制中断功能的寄存器

以下六个寄存器被用于控制中断功能。

- 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H)
- 中断屏蔽标志寄存器 (MK0L, MK0H, MK1L, MK1H)
- 优先级指定标志寄存器 (PR0L, PR0H, PR1L, PR1H)
- 外部中断上升沿允许寄存器 (EGP)
- 外部中断下降沿允许寄存器 (EGN)
- 程序状态字 (PSW)

表 16-2 所示为对应于中断请求源的中断请求标志, 中断屏蔽标志以及优先级指定标志的列表。

表 16-2. 对应于中断请求源的标志

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						
INTP4	PIF4		PMK4						
INTP5	PIF5		PMK5						
INTSRE6	SREIF6		SREMK6		SREPR6				
INTSR6	SRIF6	IF0H	SRMK6	MK0H	SRPR6	PR0H			
INTST6	STIF6		STMK6		STPR6				
INTCSI10	CSIIF10 注 1		DUALIF0 注 1		CSIMK10 注 2		DUALMK0 注 2	CSIPR10 注 3	DUALPR0 注 3
INTST0	STIF0 注 1				STMK0 注 2			STPR0 注 3	
INTTMH1	TMIFH1				TMMKH1			TMPRH1	
INTTMH0	TMIFH0				TMMKH0			TMPRH0	
INTTM50	TMIF50				TMMK50			TMPR50	
INTTM000	TMIF000				TMMK000			TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010				
INTAD	ADIF	IF1L	ADMK	MK1L	ADPR	PR1L			
INTSR0	SRIF0		SRMK0		SRPR0				
INTTM51 ^{注 4}	TMIF51		TMMK51		TMPR51				
INTIIC0	IICIF0	IF1H	IICMK0	MK1H	IICPR0	PR1H			

- 注
1. 如果产生中断源 INTCSI10 或 INTST0，IF0H 的第 2 位置（1）。
 2. MK0H 的第 2 位支持中断源 INTCSI10 和 INTST0。
 3. PR0H 的第 2 位支持中断源 INTCSI10 和 INTST0。
 4. 当在载波发生模式下使用 8 位定时器 / 事件计数器 51 时，产生 INTTM5H1 信号时产生中断（参见图 9-13 传输时序）。

(1) 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H)

当产生相应的中断请求或者当执行指令时，设置中断请求标志为 1。在中断请求响应或复位信号产生而执行指令时，它们将清 0。

当响应中断时，将自动清除中断请求标志，然后进入中断程序。

IF0L, IF0H, IF1L, 以及 IF1H 可以通过 1 位或 8 位存储器操作指令来设置。当 IF0L 和 IF0H, IF1L 和 IF1H 组成 16 位寄存器 IF0, IF1 时，可以通过 16 位存储器操作指令来设置。

复位信号的产生将这些寄存器设置为 00H。

图 16-2. 中断请求标志寄存器的格式 (IF0L, IF0H, IF1L, IF1H)

地址: FFE0H 复位后: 00H R / W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	SREIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIF

地址: FFE1H 复位后: 00H R / W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	TMIF010	TMIF000	TMIF50	TMIFH0	TMIFH1	DUALIF0 CSIIF10 STIF0	STIF6	SRIF6

地址: FFE2H 复位后: 00H R / W

符号	7	6	5	4	<3>	2	<1>	<0>
IF1L	0	0	0	0	TMIF51	0	SRIF0	ADIF

地址: FFE3H 复位后: 00H R / W

符号	7	6	5	4	3	2	1	<0>
IF1H	0	0	0	0	0	0	0	IICIF0

XXIFX	中断请求标志
0	没有产生中断请求信号
1	产生中断请求，中断请求状态

- 注意事项
1. 确保将 IF1L 的第 2、4 到 7 位和 IF1H 的第 1 到 7 位清 0。
 2. 在待机释放后操作定时器，串行接口或 A/D 转换器时，应在清除中断请求标志后应再执行一次操作。噪声可能会设置中断请求标志。

注意事项 3. 当操作中断请求标志寄存器的标志时，应使用 1 位存储器操作指令（CLR1）。当以 C 语言描述时，应使用例如“F0L. 0=0”或“_asm（“clr1 IF0L, 0”）；”的位操作指令，因为编译的汇编语言必须是 1 位操作指令（CLR1）。

如果使用例如“F0L&=0xfe；” 8 位存储器操作指令的 C 语言来描述程序，编译时，将编译为三条汇编语言指令。

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

在这种情况下，在“mov a, IF0L”和“mov IF0L, a”之间的时间内，即使同一中断请求标志寄存器（IF0L）的其它位中的请求标志被设为 1，该标志也将在“mov IF0L, a”处被清除。因此，在 C 语言中必须谨慎使用 8 位存储器操作指令。

(2) 中断请求标志寄存器 (MK0L, MK0H, MK1L, MK1H)

中断屏蔽标志用于允许 / 禁止相应的可屏蔽中断服务。

MK0L, MK0H, MK1L, 以及 MK1H 可以通过 1 位或 8 位存储器操作指令来设置。当 MK0L 和 MK0H, MK1L 和 MK1H 组成 16 位寄存器 MK0 和 MK1 时, 它们可以通过 16 位存储器操作指令来设置。

复位信号的产生将这些寄存器设置为 FFH。

图 16-3. 中断屏蔽标志寄存器的格式 (MK0L, MK0H, MK1L, MK1H)

地址: FFE4H 复位后: FFH R / W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	SREMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

地址: FFE5H 复位后: FFH R / W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	TMMK010	TMMK000	TMMK50	TMMKH0	TMMKH1	DUALMK0 CSIMK0 STMK0	STMK6	SRMK6

地址: FFE6H 复位后: FFH R / W

符号	7	6	5	4	<3>	2	<1>	<0>
MK1L	1	1	1	1	TMMK51	1	SRMK0	ADMK

地址: FFE7H 复位后: FFH R / W

符号	7	6	5	4	3	2	1	<0>
MK1H	1	1	1	1	1	1	1	IICMK0

XXMKX	中断服务控制						
0	允许中断服务						
1	禁止中断服务						

注意事项 确保将 MK1L 的第 2、4 到 7 位和 MK1H 的第 1 到 7 位设置为 1。

(3) 优先级指定标志寄存器 (PR0L, PR0H, PR1L, PR1H)

优先级指定标志寄存器用于设置相应的可屏蔽中断优先级。

PR0L, PR0H, PR1L, 以及 PR1H 可以通过 1 位或 8 位存储器操作指令来设置。当 PR0L 和 PR0H, PR1L 和 PR1H 组成 16 位寄存器 PR0 和 PR1 时, 它们可以通过 16 位存储器操作指令来设置。

复位信号的产生将这些寄存器设置为 FFH。

图 16-4. 优先指定标志寄存器格式 (PR0L, PR0H, PR1L, PR1H)

地址: FFE8H 复位后: FFH R / W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0L	SREPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	LVIPR

地址: FFE9H 复位后: FFH R / W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0H	TMPR010	TMPR000	TMPR50	TMPRH0	TMPRH1	DUALPR0 CSIPR10 STPR0	STPR6	SRPR6

地址: FFEAH 复位后: FFH R / W

符号	7	6	5	4	<3>	2	<1>	<0>
PR1L	1	1	1	1	TMPR51	1	SRPR0	ADPR

地址: FFE BH 复位后: FFH R / W

符号	7	6	5	4	3	2	1	<0>
PR1H	1	1	1	1	1	1	1	IICPR0

XXPRX	优先级选择
0	高优先级
1	低优先级

注意事项 确保将 PR1L 的第 2、4 到 7 位和 PR1H 的第 1 到 7 位设置为 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)

这些寄存器为 INTP0 到 INTP5 指定有效沿。

EGP 和 EGN 可以通过 1 位或 8 位操作指令来设置。

复位信号的产生将这些寄存器设置为 00H。

图 16-5. 外部中断上升沿允许寄存器 (EGP) 的格式
以及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	上升沿及下降沿

表 16-3 所示为对应于 EGPn 和 EGNn 的端口。

表 16-3. 相应于 EGPn 和 EGNn 的端口

检测允许寄存器		边沿检测端口	中断请求信号
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P30	INTP1
EGP2	EGN2	P31	INTP2
EGP3	EGN3	P32	INTP3
EGP4	EGN4	P33	INTP4
EGP5	EGN5	P16	INTP5

注意事项 从外部中断功能切换到端口功能时，可能检测到边沿。因此通过将 EGPn 以及 EGNn 清 0 可选择端口模式，

备注 n = 0 到 5

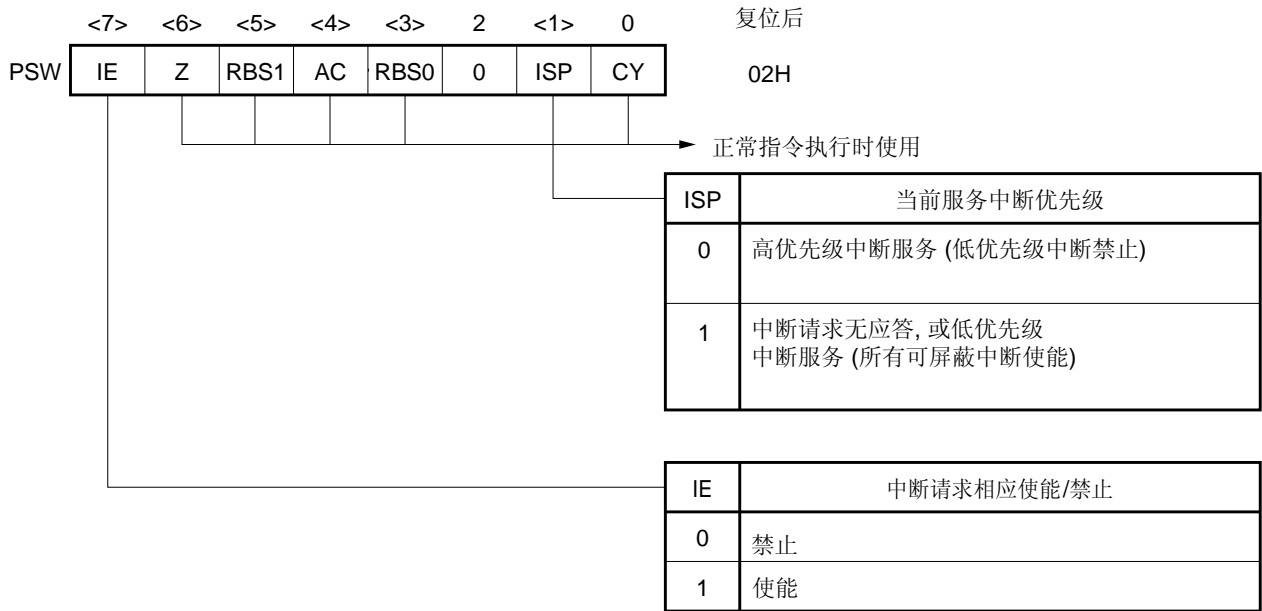
(5) 程序状态字 (PSW)

程序状态字用于保持指令执行结果以及中断请求的当前状态，其中包括设置允许 / 禁止可屏蔽中断的 IE 标志以及控制中断嵌套的 ISP 标志。

除了 8 位读 / 写，还可以通过使用位操作指令和专用指令 (EI 和 DI) 对该寄存器执行操作。当响应中断请求时，如果执行了 BRK 指令，则将 PSW 的内容自动保存到堆栈中，且将 IE 标志复位为 0。如果响应可屏蔽中断请求，则所响应的中断的优先级指定标志中的内容传输到 ISP 标志中。通过 PUSH PSW 指令，也可以将 PSW 的内容保存到堆栈中。可以通过 RETI, RETB 以及 POP PSW 指令从堆栈中恢复。

复位信号的产生将 PSW 设置为 02H。

图 16-6. 程序状态字的格式



16.4 中断服务操作

16.4.1 可屏蔽中断响应

当中断请求标志设为 1 且对应于该中断请求的屏蔽 (MK) 标志被清 0 时, 将响应可屏蔽中断。如果中断处于中断允许状态 (当 IE 标志被设为 1), 则响应中断请求。然而, 在处理高优先级中断请求期间不会响应低优先级中断请求 (当 ISP 标志复位为 0 时)。

从可屏蔽中断请求的产生到执行中断服务的时间如下表 16-4 所示。

关于中断请求的响应时间, 请参照图 6-8 以及 16-9。

表 16-4. 从可屏蔽中断的产生到服务的时间

	最短时间	最长时间 ^注
当 $\times\times PR = 0$	7 个时钟	32 个时钟
当 $\times\times PR = 1$	8 个时钟	33 个时钟

注 如果在 RET 指令前产生中断请求, 那么等待时间将变长。

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

如果同时产生两个或更多的可屏蔽中断, 那么将首先响应在优先级指定标志中所指定的具有高优先级的请求。如果两个或更多的中断请求具有同一优先级, 则将先响应具有默认优先级最高的请求。

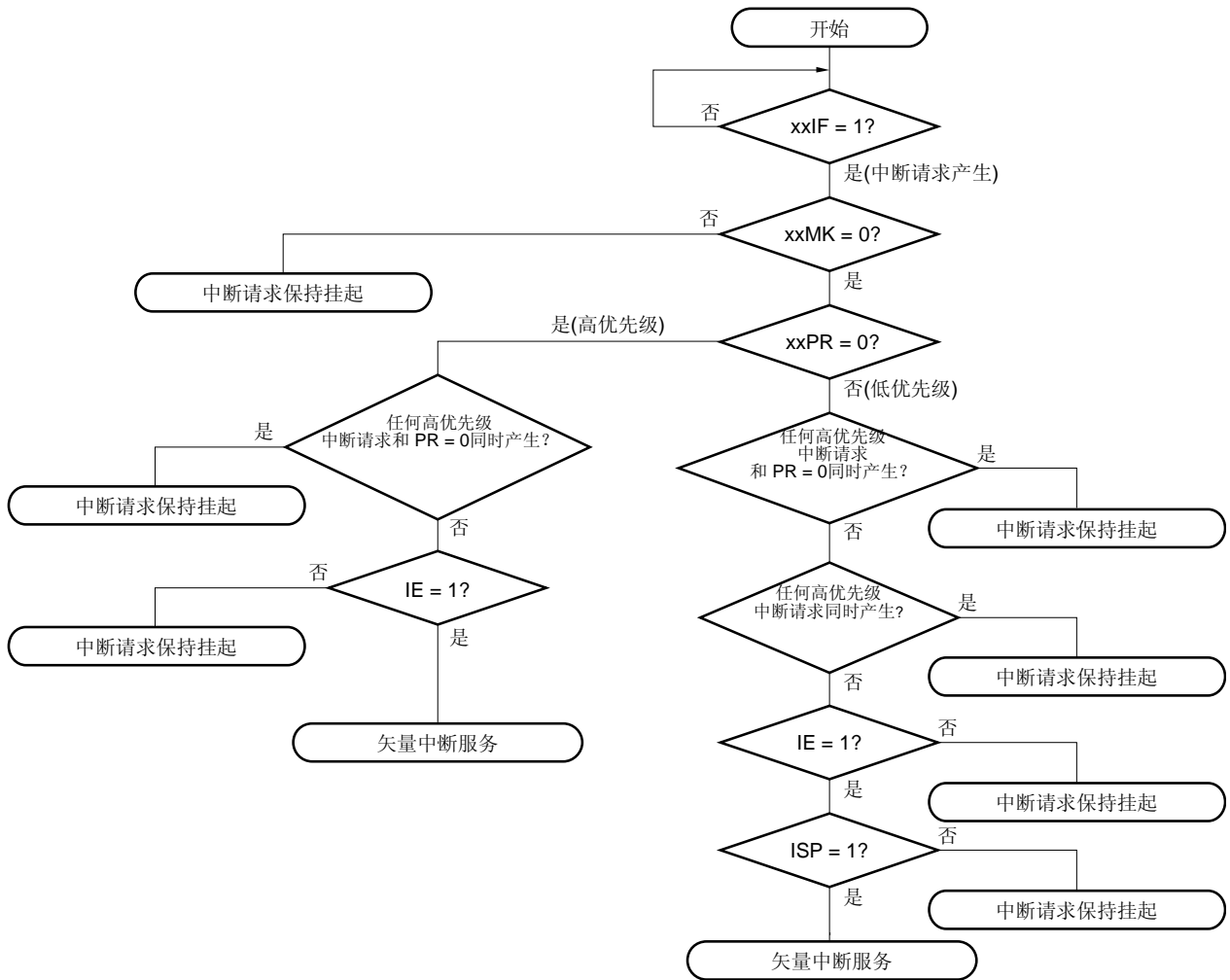
当允许响应中断请求时, 响应处于等待状态的中断请求。

图 16-7 所示为响应中断请求的算法。

如果响应可屏蔽中断请求, 则按照先保存 PSW 然后保存 PC 的顺序入栈, 也将复位 IE 标志 (0), 而对应于被响应中断的优先级指定标志中的内容则被传输到 ISP 标志中。将每个中断请求的向量表数据装载到 PC 中并转移。

通过使用 RETI 指令可以从中断返回。

图 16-7. 中断请求响应处理算法



xxIF: 中断请求标志

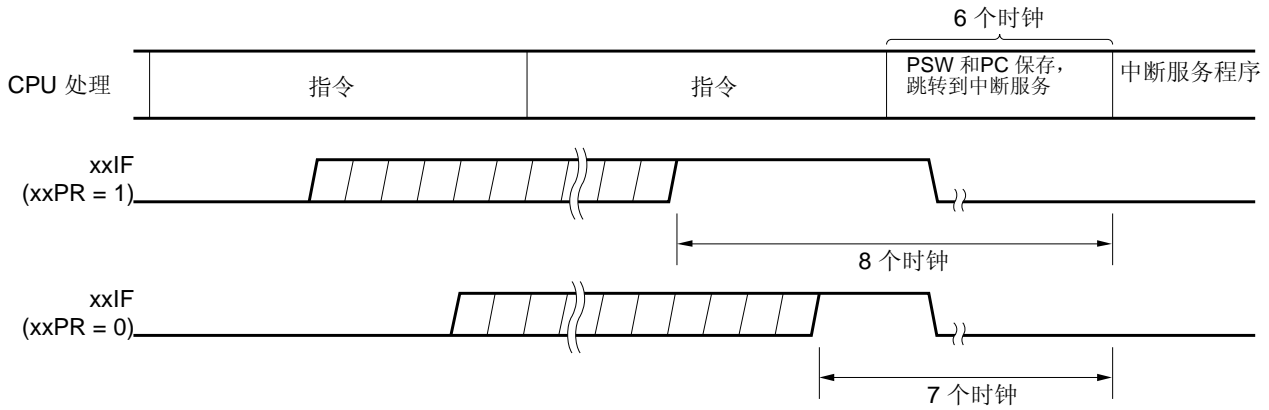
xxMK: 中断屏蔽标志

xxPR: 优先级指定标志

IE: 控制可屏蔽中断请求控制响应的标志 (1=允许, 0=禁止)

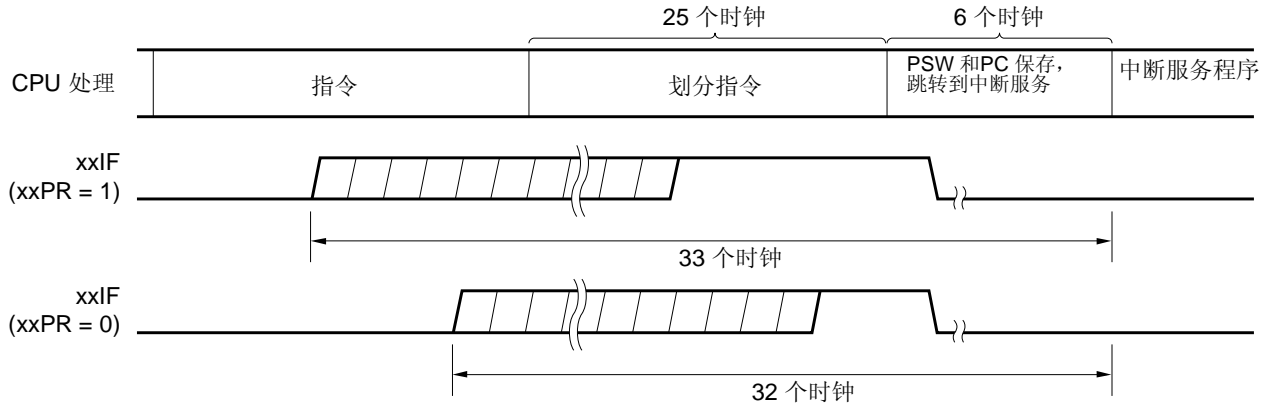
ISP: 表示当前服务中断优先级的标志 (0=高优先级中断服务, 1=无中断请求响应, 或低优先级中断服务)

图 16-8. 中断请求响应时间（最短时间）



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

图 16-9. 中断请求响应时间（最长时间）



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

16.4.2 软件中断请求响应

软件中断响应通过 BRK 指令执行来响应。不能禁止软件中断。

如果响应软件中断，则按照先保存程序状态字（PSW）然后保存程序计数器（PC）的顺序入栈，也将复位 IE 标志（0），而向量表（003EH, 003FH）的内容则装载到 PC 中并转移。

通过使用 RETB 指令可以从软件中断返回。

注意事项 不要使用 RETI 指令从软件中断返回。

16.4.3 中断嵌套

在执行中断服务程序时，又响应了其他中断，这时就产生了中断嵌套。

除非选择允许中断请求响应状态（IE = 1），否则不会产生中断嵌套。在响应中断请求时，禁止响应其它中断请求（IE = 0）。因此，如果要允许中断嵌套，必须在中断服务期间执行 EI 指令，将 IE 标志置 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，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。

在中断允许状态中，如果产生的中断请求的优先级等于或高于正在服务的中断优先级，则响应该中断请求，从而产生中断嵌套。如果产生的中断请求的优先级低于正在服务的中断优先级，则不响应该中断请求。由于禁止中断或中断请求的优先级较低，这些不被允许的中断请求处于等待状态。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 16-5 所示为允许中断嵌套的中断请求之间的关系，图 16-10 为中断嵌套示例。

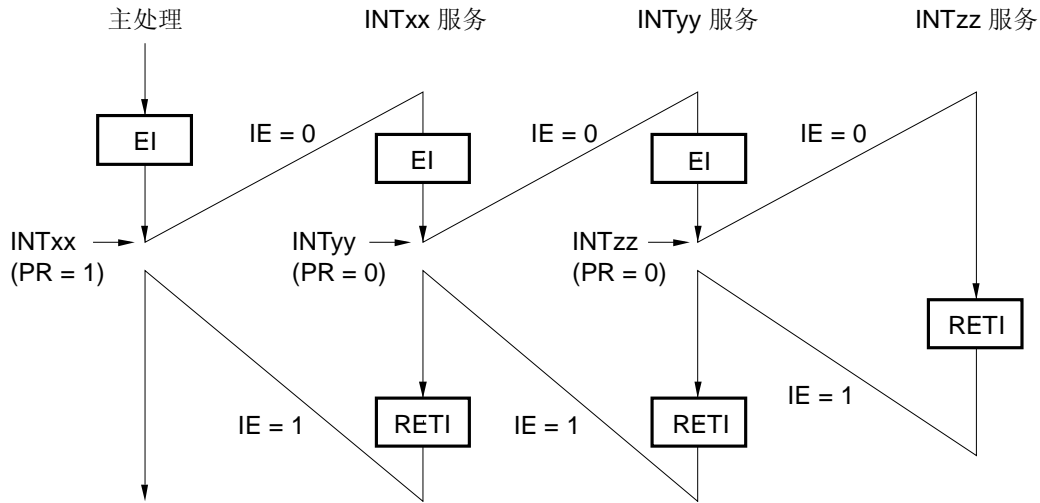
表 16-5. 中断服务期间允许进行中断嵌套的中断请求之间的关系

中断嵌套		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	
正在进行的	中断					
可屏蔽中断	ISP = 0	○	×	×	×	○
	ISP = 1	○	×	○	×	○
软件中断		○	×	○	×	○

- 备注**
- : 允许中断嵌套
 - ×: 禁止中断嵌套
 - ISP 和 IE 是包含在 PSW 中的标志。
 ISP = 0: 高优先级中断被服务。
 ISP = 1: 无中断请求响应，或低优先级中断被服务。
 IE = 0: 禁止中断请求响应。
 IE = 1: 允许中断请求响应。
 - PR 是 P0L, P0H, P1L 和 P1H 中的标志。
 PR = 0: 高优先级
 PR = 1: 低优先级

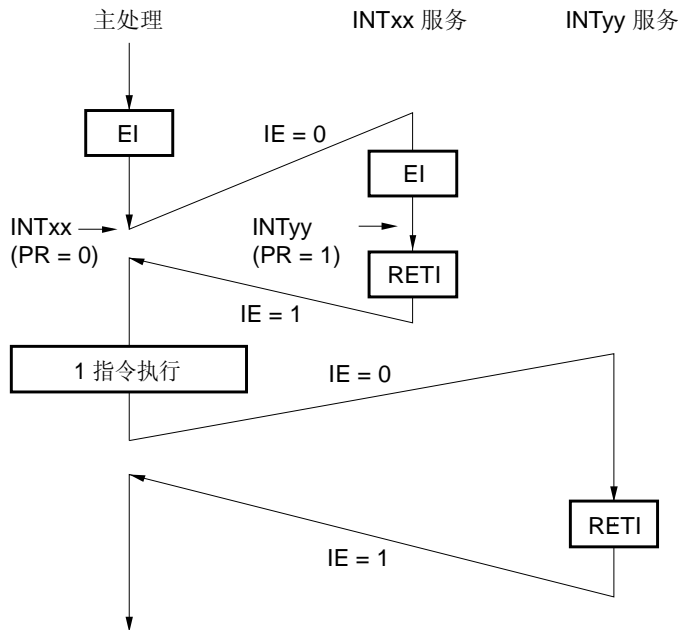
图 16-10. 中断嵌套的示例 (1 / 2)

例 1. 发生两次中断嵌套



在中断 INTxx 中响应了两个中断请求 INTyy 和 INTzz，此时发生中断嵌套。在每个中断请求响应之前，必须执行 EI 指令才能允许中断请求响应。

例 2. 由于优先级控制没有发生中断嵌套

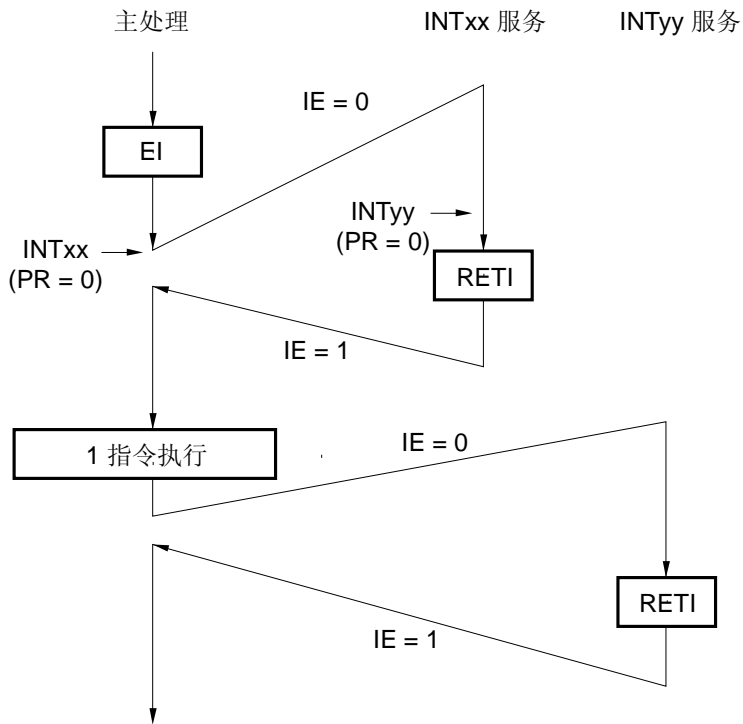


在中断 INTxx 的服务期间发出的中断请求 INTyy 由于其优先级低于 INTxx 的优先级，因此将不会响应它，也不会发生中断嵌套。INTyy 中断请求保持为挂起状态，并且在执行一个主程序指令后响应。

- PR = 0: 高优先级
- PR = 1: 低优先级
- IE = 0: 禁止中断请求响应。

图 16-10. 中断嵌套的示例 (2 / 2)

例 3. 由于不允许中断没有产生中断嵌套，



在中断 INT_{xx} 服务期间不允许中断 (EI 指令没有发出)，因此，不会响应中断请求 INT_{yy} ，也不会发生中断嵌套。 INT_{yy} 中断请求保持为挂起状态，并且在执行一个主处理指令后响应。

PR = 0: 高优先级

IE = 0: 禁止中断请求响应。

16.4.4 中断请求保持

在某些指令执行期间，即使发出中断请求，响应请求也要保持在挂起状态，直至下个指令结束。以下列出了这类指令（中断请求保存指令）。

- MOV PSW, #byte
- MOV PSW, A
- 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
- 对 IF0L, IF0H, IF1L, IF1H, MK0L, MK0H, MK1L, MK1H, PR0L, PR0H, PR1L 以及 PR1H 寄存器的操作指令。

注意事项 BRK 指令不属于以上列出的中断请求保持指令之一。但是，通过执行 BRK 指令激活的软件中断会清除 IE 标志。因此，即使在 BRK 指令的执行过程中产生可屏蔽中断请求，也不会响应中断请求。

图 16-11 所示为中断请求处于等待状态的时序。

图 16-11. 中断请求保持



- 备注**
1. 指令 N: 中断请求保持指令
 2. 指令 M: 不同于中断请求保持指令的指令
 3. xxPR（优先级）值不会影响xxIF（中断请求）的操作。

第十七章 待机功能

17.1 待机功能和配置

17.1.1 待机功能

待机功能设计用来降低系统工作电流。有以下两种模式。

(1) HALT 模式

通过执行 HALT 指令设置 HALT 模式。在 HALT 模式下，CPU 操作时钟停止。如果在设置 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处于工作状态，则设置后每个时钟的振荡都将会继续进行。在这种模式下，工作电流不如在 STOP 模式下减少得多，但 HALT 模式对于在中断请求产生后立即恢复操作以及执行频繁发生的间歇性操作非常有效。

(2) STOP 模式

通过执行 STOP 指令设置 STOP 模式。在 STOP 模式下，将停止高速系统时钟振荡器和内置高速振荡器，整个系统将停止操作，因此可以大大减少 CPU 工作电流。

可以通过中断请求释放该模式，因此可以执行间歇性的操作。但是，当选择 X1 时钟作为系统时钟时，在释放 STOP 模式需要一个等待时间以确保振荡稳定，因此，如果产生中断请求后需要立即执行处理，则应选择 HALT 模式。

对于以上两种模式中的任一种模式，寄存器、标志以及数据存储器中的所有内容都将保持设置待机模式前的状态。也将保持输入 / 输出端口的输出锁存器以及输出缓冲器的状态。

- 注意事项**
1. 当切换到 STOP 模式时，应确保在执行 STOP 指令前停止使用主系统时钟的外围硬件操作。
 2. 当使用待机功能时，建议按以下顺序来减少 A/D 转换器的消耗电流：先将 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 和第 0 位 (ADCE) 清 0，用来停止 A/D 转换操作，然后再执行 STOP 指令。

17.1.2 控制待机功能的寄存器

待机功能由以下两种寄存器来控制。

- 振荡稳定时间计数状态寄存器 (OSTC)
- 振荡稳定时间选择寄存器 (OSTS)

备注 关于开始、停止或选择时钟的寄存器，参见第六章 时钟发生器。

(1) 振荡稳定时间计数状态寄存器 (OSTC)

这是 X1 时钟振荡稳定时间计数器状态寄存器。当 X1 时钟振荡器与作为 CPU 时钟的内部高速振荡器时钟一起启动时，可以检测 X1 时钟振荡器稳定时间。

OSTC 可以通过 1 位或 8 位存储器操作指令来读取。

当复位释放时（通过 RESET 输入、POC、LVI、WDT 复位），STOP 指令和 MSTOP（MOC 寄存器的第 7 位）=1 将清除 OSTC 为 00H。

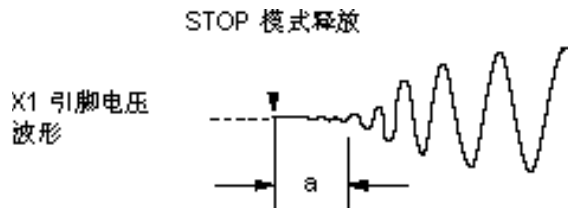
图 17-1. 振荡稳定时间计数状态寄存器 (OSTC) 的格式

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	
					fx = 10 MHz	fx = 20 MHz
1	0	0	0	0	$2^{11}/f_x$ 最小	204.8 μ s 最小
1	1	0	0	0	$2^{13}/f_x$ 最小	819.2 μ s 最小
1	1	1	0	0	$2^{14}/f_x$ 最小	1.64 ms 最小
1	1	1	1	0	$2^{15}/f_x$ 最小	3.27 ms 最小
1	1	1	1	1	$2^{16}/f_x$ 最小	6.55 ms 最小

- 注意事项**
- 上述时间过后，将从 MOST11 位开始按顺序将各位设置为 1，并且保持。
 - 振荡稳定时间计数器累计由 OSTC 所设置的振荡稳定时间。如果先进入 STOP 模式，然后在内置高速振荡器时钟用作 CPU 时钟时释放 STOP 模式，按以下方式设置振荡稳定时间。
 - 期望的 OSTC 振荡稳定时间 \leq 由 OSTC 所设置的振荡稳定时间
 因此，需注意，只有在释放 STOP 模式后，由 OSTC 所设置的振荡稳定时间的状态才会设置到 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（下面的“a”）。



备注 fx: X1 时钟振荡频率

(2) 振荡稳定时间选择寄存器 (OSTS)

当释放 STOP 模式时, 该寄存器用于选择 X1 时钟振荡稳定等待时间。

当 X1 时钟设置为 CPU 时钟时, 由 OSTS 设置释放 STOP 模式后所需等待的时间。

当内置高速振荡时钟设置为 CPU 时钟时, 使用 OSTC 来确定在 STOP 模式释放后所期望的振荡稳定时间已经过去。可以使用 OSTC 检测是否已经达到设置的振荡稳定时间。

OSTS 可以通过 8 位存储器操作指令来设置。

复位信号的产生设置 OSTS 为 05H。

图 17-2. 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFA4H 复位后: 05H R/W

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

OSTS2	OSTS1	OSTS0	振荡稳定时间选择		
			$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$	
0	0	1	$2^{11}/f_x$	204.8 μs	102.4 μs
0	1	0	$2^{13}/f_x$	819.2 μs	409.6 μs
0	1	1	$2^{14}/f_x$	1.64 ms	819.2 μs
1	0	0	$2^{15}/f_x$	3.27 ms	1.64 ms
1	0	1	$2^{16}/f_x$	6.55 ms	3.27 ms
其它			禁止设置		

- 注意事项**
- 如果在 X1 时钟用作 CPU 时钟时设置 STOP 模式, 则应在执行 STOP 指令前设置 OSTS。
 - 在 X1 时钟振荡时间内不要改变 OSTS 寄存器的值。
 - 振荡稳定时间计数器达到由 OSTS 所设置的振荡稳定时间。如果先进入 STOP 模式, 然后在内置高速振荡时钟用作 CPU 时钟时释放 STOP 模式, 按以下方式设置振荡稳定时间。
 - 期望的 OSTC 振荡稳定时间 \leq 由 OSTS 所设置的振荡稳定时间
 因此, 需注意, 在释放 STOP 模式后, 只有由 OSTS 所设置的振荡稳定时间的状态才会设置到 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 (下面的“a”)。



备注 f_x : X1 时钟振荡频率

17.2 待机功能操作

17.2.1 HALT 模式

(1) HALT 模式

通过执行 HALT 指令设置 HALT 模式。不管在设置前 CPU 时钟是高速系统时钟还是内部高速振荡器时钟，都可以设置 HALT 模式。

以下所示为 HALT 模式中的运行状态。

表 17-1. HALT 模式中的运行状态

HALT 模式设置		当 HALT 指令被执行时，此时 CPU 以主系统时钟运行		
		当 CPU 以内置高速振荡时钟 (f _{RH}) 运行时	当 CPU 以 X1 时钟 (f _X) 运行时	当 CPU 以外部主系统时钟 (f _{EXCLK}) 运行时
项目				
系统时钟		提供给 CPU 的时钟被停止		
主系统时钟	f _{RH}	继续运行（不能停止）	保持设置 HALT 模式前的状态	
	f _X	设置 HALT 模式前的状态被保持	继续运行（不能停止）	设置 HALT 模式前的状态被保持
	f _{EXCLK}	通过外部时钟输入来运行或停止		继续运行（不能停止）
	f _{RL}	设置 HALT 模式前的状态被保持		
CPU		停止运行		
Flash 存储器				
RAM		设置 HALT 模式前的状态被保持		
端口（锁存器）				
16 位定时器/事件计数器 00		可运行		
8 位定时器/事件计数器	50			
	51			
8 位定时器	H0			
	H1			
看门狗定时器		可运行。当通过选项字节设置“内部低速振荡器可被软件停止”时，提供给看门狗定时器的时钟被停止。		
A/D		可运行		
串行接口	UART0			
	UART6			
	CSI10			
	IIC0			
上电清零功能				
低压检测功能				
外部中断				

备注

f_{RH}: 内置高速振荡时钟

f_X: X1 时钟

f_{EXCLK}: 外部主系统时钟

f_{RL}: 内置低速振荡时钟

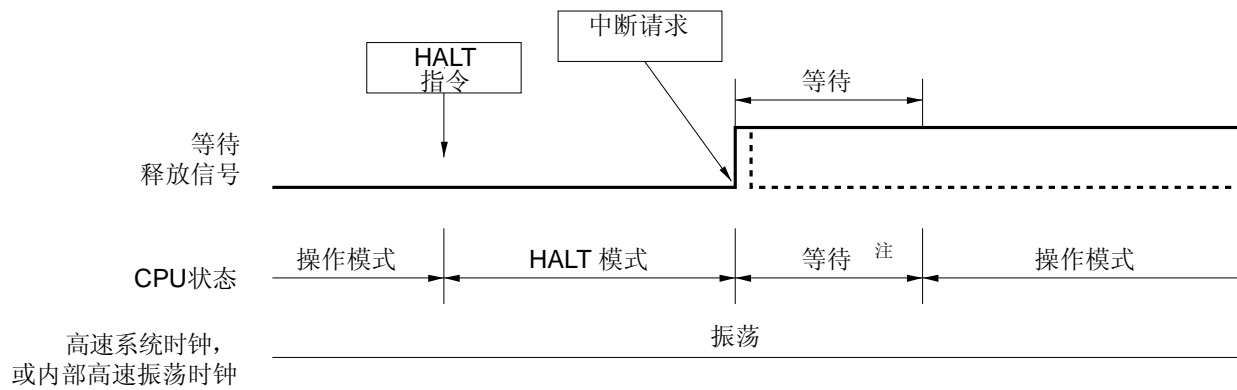
(2) 释放 HALT 模式

可以通过以下两种中断源释放 HALT 模式。

(a) 通过非屏蔽中断请求释放

当产生非屏蔽中断请求时，将释放 HALT 模式。如果允许中断响应，则执行中断服务。如果禁止中断响应，则执行下一个地址指令。

图 17-3. 通过产生中断请求来释放 HALT 模式



注 等待时间如下：

- 当执行中断服务时：11 或 12 个时钟
- 当没有执行中断服务时：4 或 5 个时钟

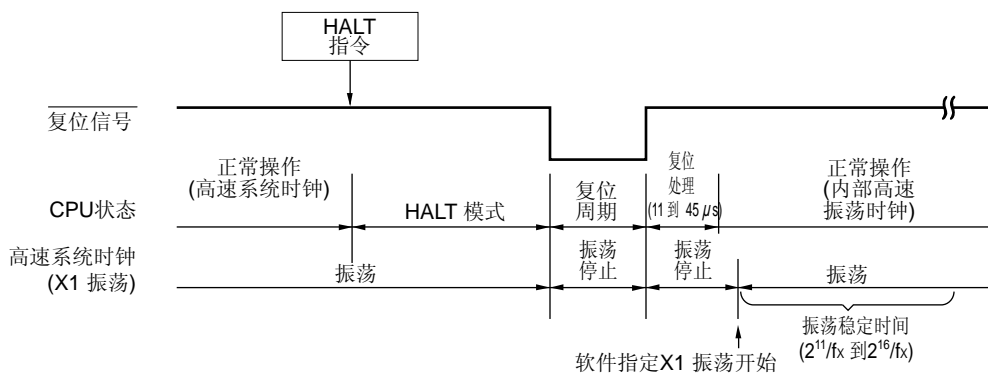
备注 虚线表明已经释放了待机模式的中断请求的响应情况。

(b) 通过复位信号的产生释放

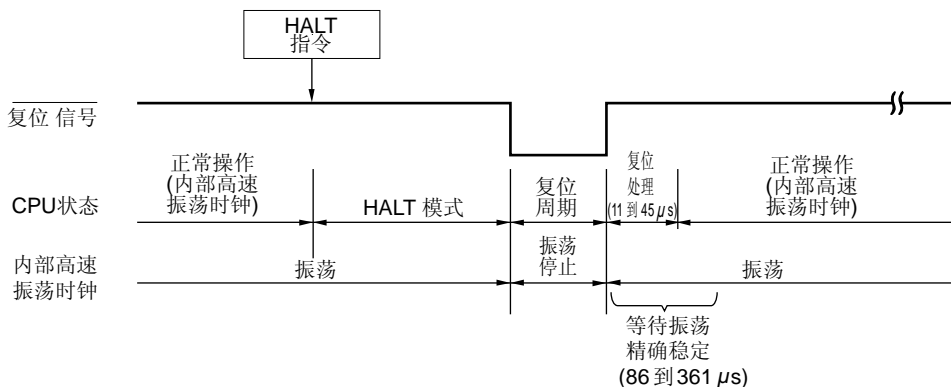
当产生复位信号时，将释放 HALT 模式，之后，与使用正常复位操作的情况相同，程序在跳转到复位向量地址后开始执行。

图 17-4. 通过复位释放 HALT 模式

(1) 当高速系统时钟设置为 CPU 时钟时



(2) 当内置高速振荡时钟设置为 CPU 时钟时



备注 fx: X1 时钟振荡频率

表 17-2. HALT 模式下中断请求响应操作

释放源	MK _{xx}	PR _{xx}	IE	ISP	操作
可屏蔽中断请求	0	0	0	×	执行下一地址指令
	0	0	1	×	执行中断服务
	0	1	0	1	执行下一地址指令
	0	1	×	0	
	0	1	1	1	执行中断服务
	1	×	×	×	保持 HALT 模式
复位	-	-	×	×	复位处理

×: 无关

17.2.2 STOP 模式

(1) STOP 模式设置及运行状态

通过执行 STOP 指令设置 STOP 模式。不管在设置前 CPU 时钟是高速系统时钟，还是内部高速振荡器时钟，都可以设置 STOP 模式。

注意事项 因为中断请求信号被用于释放待机模式，因此，如果有一个中断源，在中断屏蔽标志复位的情况下设置了中断请求标志，则立即释放待机模式。因此，执行 STOP 指令后，立即将 STOP 模式复位到 HALT 模式，并且在经历了 OSTS 设置的等待时间后，系统返回操作模式。

以下所示为 STOP 模式中的运行状态。

表 17-3. STOP 模式中的运行状态

STOP 模式设置		当 STOP 指令被执行时, 此时 CPU 以主系统时钟运行		
		当 CPU 以内置高速振荡时钟 (f _{RH}) 运行时	当 CPU 以 X1 时钟 (f _X) 运行时	当 CPU 以外部主系统时钟 (f _{EXCLK}) 运行时
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}		保持设置 STOP 模式前的状态		
CPU		停止运行		
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		停止运行		
8 位定时器/事件计数器	50 ^注	只有当 TI50 被选作计数时钟时可运行		
	51 ^注	只有当 TI51 被选作计数时钟时可运行		
8 位定时器	H0	在 8 为定时器/时间计数器 50 操作时仅当 TM50 输出被选作计数时钟时可运行		
	H1	仅当 f _{RL} , f _{RU} /2 ⁷ , f _{RL} /2 ⁹ 被选作计数时钟时运行		
看门狗计时器		可运行。当通过选项字节设置“内部低速振荡器可被软件停止”时, 提供给看门狗定时器的时钟停止。		
A/D		停止运行		
串行接口	UART0	在 8 位定时器/时间计数器 50 操作时, 仅当 TM50 的输出被选作串行时钟时可运行		
	UART6			
	CSI10 ^注	只有当外部时钟被选作串行时钟时可运行		
	IIC0 ^注	停止运行		
上电清零功能		可运行		
低压检测功能				
外部中断				

注 在停止模式下, 不要启动这些以外围硬件引脚输入的外部时钟工作的功能运行。

备注 f_{RH}: 内置高速振荡时钟
 f_X: X1 时钟
 f_{EXCLK}: 外部主系统时钟
 f_{RL}: 内置低速振荡时钟

- 注意事项
1. 为了使用在 STOP 模式下停止运行的外围硬件, 以及在 STOP 模式下停止了时钟振荡的外围硬件, 在释放 STOP 模式后, 应重新启动外围硬件。
 2. 即使在选项字节中选择“内部低速振荡器可被软件停止”, 在 STOP 模式下, 内部低速振荡器时钟将继续以 STOP 模式设置前的状态运行。为了停止 STOP 模式下的内部低速振荡器的振荡, 需使用软件停止, 然后再执行 STOP 指令。

注意事项 3. 当 CPU 以高速时钟 (X1 振荡) 运行时, 为了在释放 STOP 模式后缩短振荡稳定时间, 应按照以下顺序在执行 STOP 指令前将 CPU 时钟转变为内部高速振荡时钟。

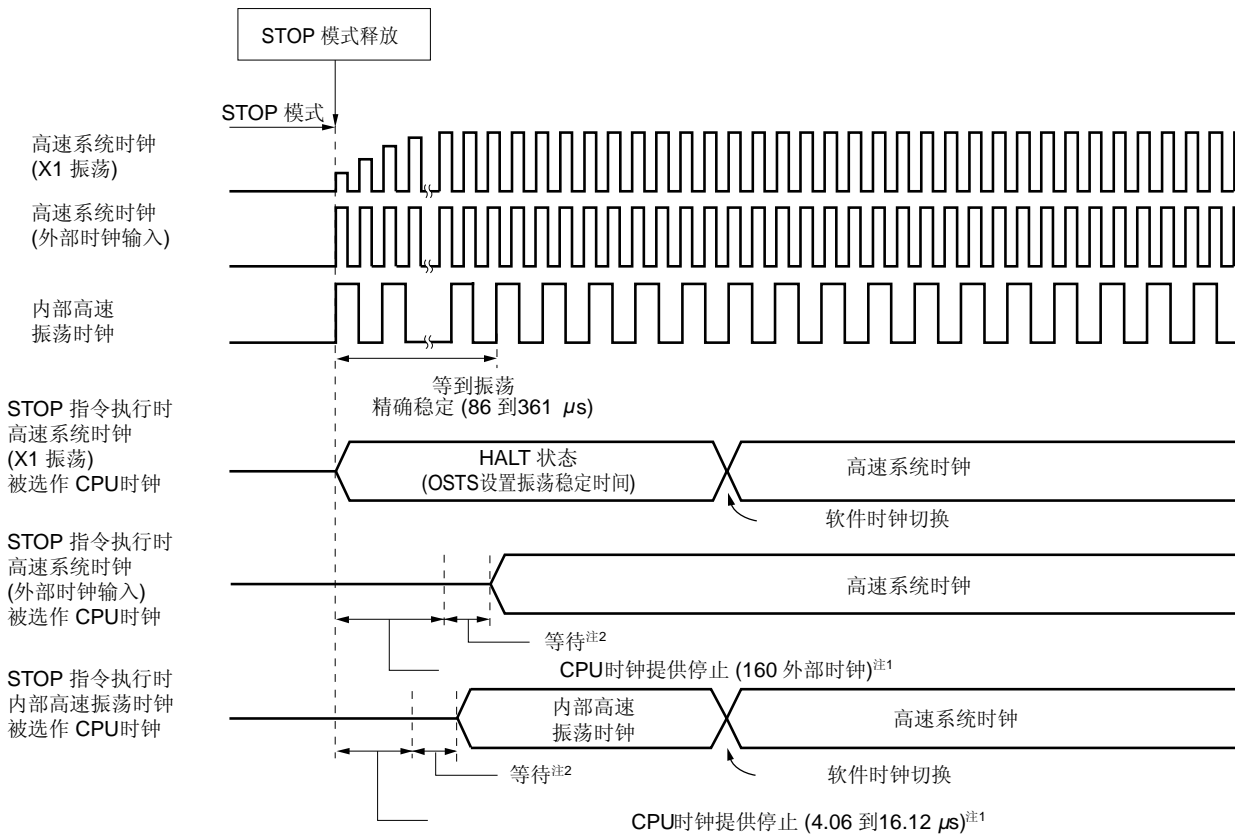
<1>设置 RSTOP 为 0 (开始内部高速振荡器振荡) → <2> 设置 MCM0 为 0 (切换 CPU 从 X1 振荡到内部高速振荡) → <3>检查 MCS 为 0 (检查 CPU 时钟) → <4>检查 RSTS 为 1 (检查内部高速振荡操作) → <5>执行 STOP 指令

释放 STOP 模式后, 在将 CPU 时钟从内置高速振荡时钟变为高速系统时钟 (X1 振荡) 前, 将使用振荡稳定时间计数状态寄存器 (OSTC) 来检查振荡稳定时间。

4. 如果在 AMPH = 1 时执行 STOP 指令, 在 STOP 模式被释放后, 当内部高速振荡器时钟选作 CPU 时钟时, 提供给 CPU 的时钟将停止 4.06 到 16.12 μs , 或者, 当内部高速系统时钟 (外部时钟输入) 选作 CPU 时钟时, 则为 160 个外部时钟周期。
5. 确保内部高速振荡器稳定操作之后 (RSTS = 1), 执行 STOP 指令。

(2) 释放 STOP 模式

图 17-5. STOP 模式释放时的操作时序
(产生非屏蔽中断请求时)



注 1. 当 AMPH = 1

2. 等待时间如下:

- 当执行中断服务时: 17 或 18 个时钟
- 当没有执行中断服务时: 11 或 12 个时钟

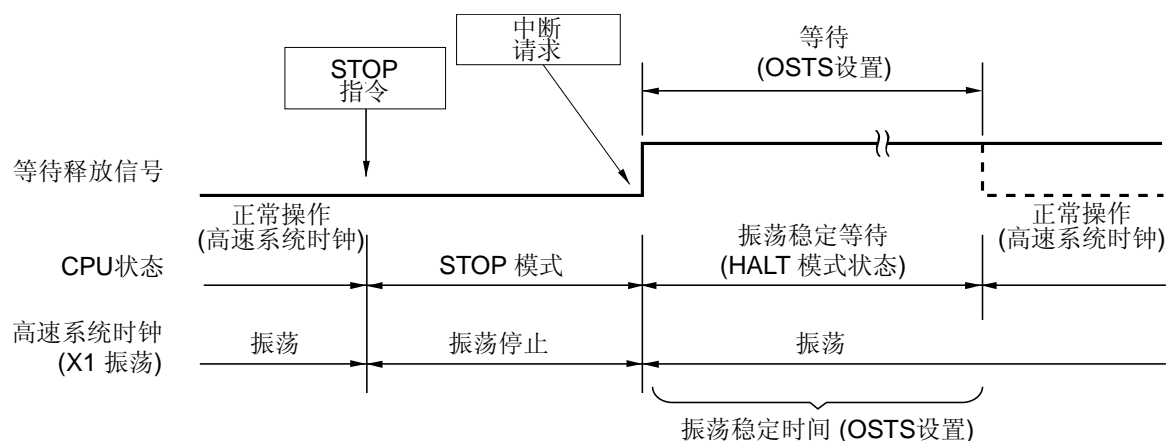
STOP 模式可以通过以下两种源来释放。

(a) 通过非屏蔽中断请求释放

当产生非屏蔽中断请求时，将释放 STOP 模式。在振荡稳定时间过去之后，如果允许中断响应，则将会执行中断服务。如果禁止中断响应，则执行下一个地址指令。

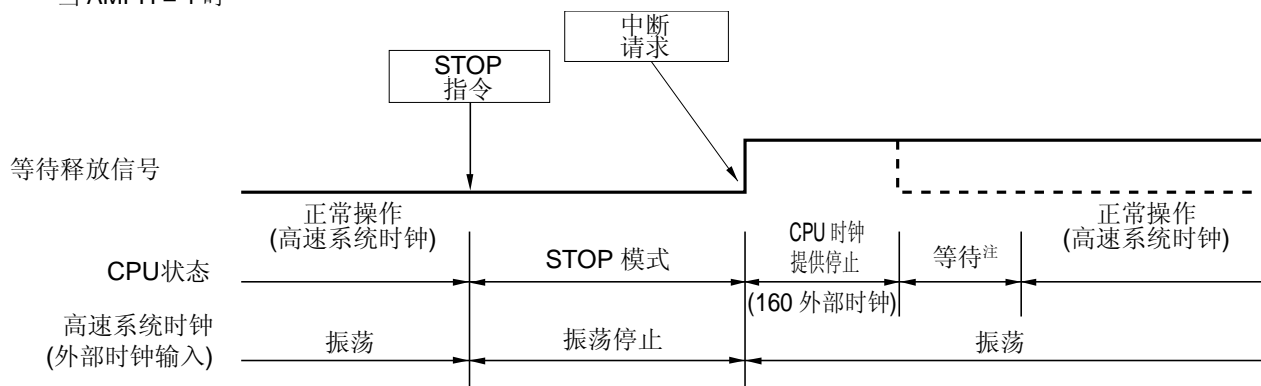
图 17-6. 通过产生中断请求来释放 STOP 模式

(1) 当高速系统时钟 (X1 振荡) 设置为 CPU 时钟时



(2) 当高速系统时钟 (外部时钟输入) 设置为 CPU 时钟时(1/2)

当 AMPH = 1 时



注 等待时间如下：

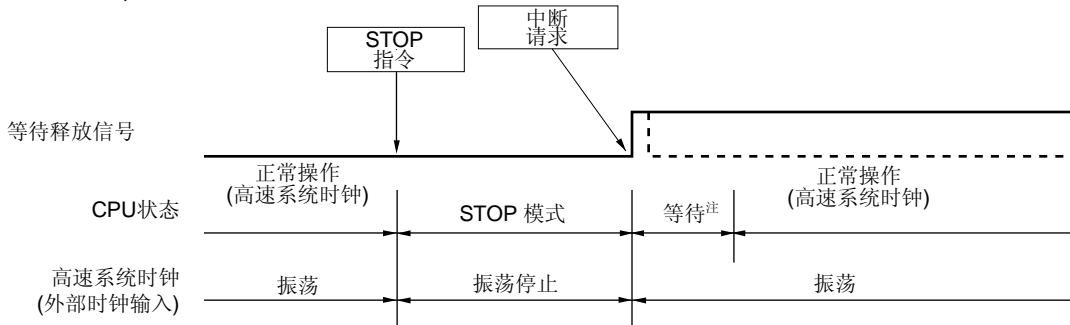
- 当执行中断服务时：17 或 18 个时钟
- 当没有执行中断服务时：11 或 12 个时钟

备注 虚线表示已经释放了待机模式的中断请求的响应情况。

图 17-6. 通过中断请求产生释放的 STOP 模式 (2/2)

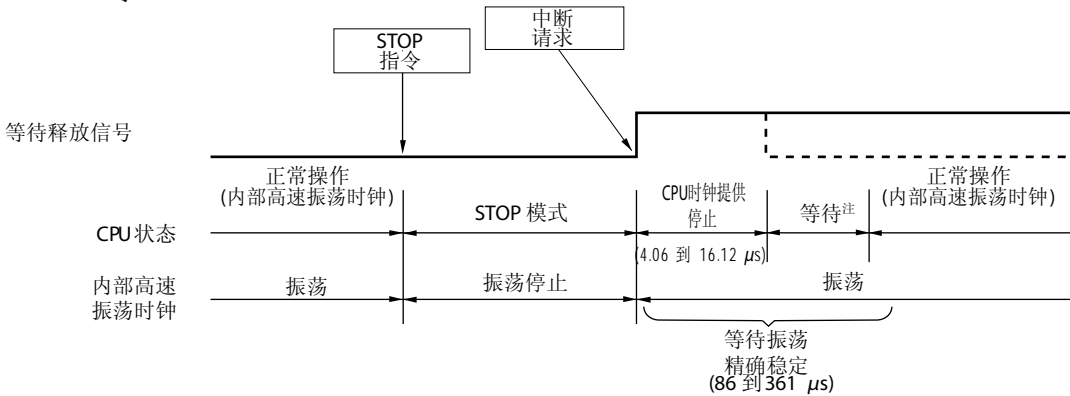
(2) 当高速系统时钟 (外部时钟输入) 设置为 CPU 时钟时(2/2)

当 AMPH = 0 时

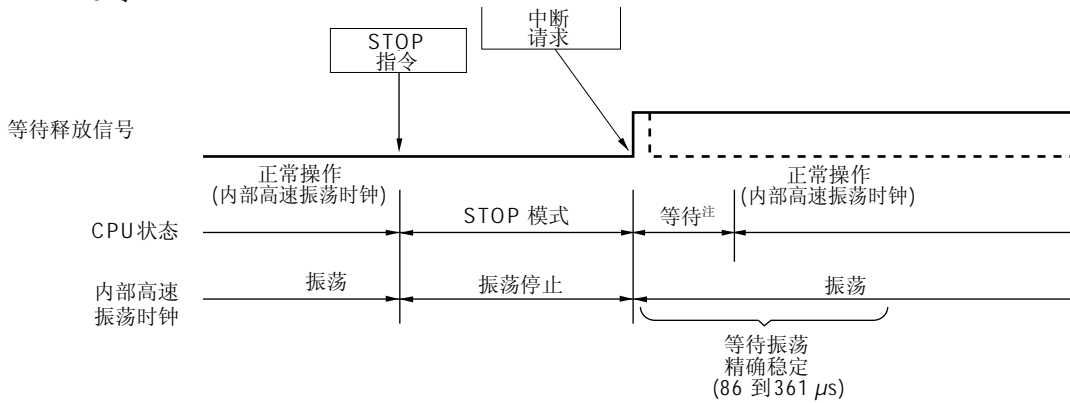


(3) 当内置高速振荡时钟设置为 CPU 时钟时

当 AMPH = 1 时



当 AMPH = 0 时



注 等待时间如下:

- 当执行中断服务时: 17 或 18 个时钟
- 当没有执行中断服务时: 11 或 12 个时钟

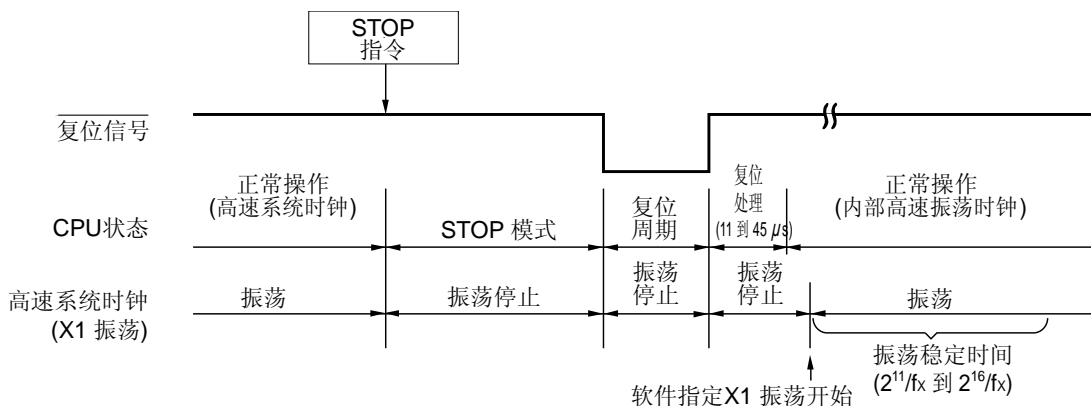
备注 虚线表示已经释放了待机模式的中断请求的响应情况。

(b) 通过复位信号的产生来释放

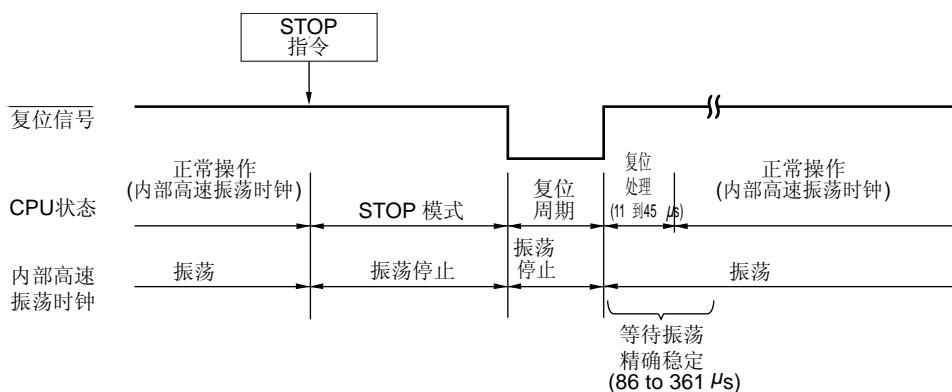
当产生复位信号时，释放 STOP 模式，然后如同使用正常复位操作的情况一样，程序跳转到复位向量地址后开始执行。

图 17-7. 通过复位释放的 STOP 模式

(1) 当高速系统时钟设置为 CPU 时钟时



(2) 当内置高速振荡时钟设置为 CPU 时钟时



备注 fx: X1 时钟振荡频率

表 17-4. STOP 模式下中断请求响应操作

释放源	MKxx	PRxx	IE	ISP	操作
可屏蔽中断请求	0	0	0	×	执行下一条地址指令
	0	0	1	×	执行中断服务程序
	0	1	0	1	执行下一条地址指令
	0	1	×	0	执行下一条地址指令
	0	1	1	1	执行中断服务
	1	×	×	×	保持 STOP 模式
复位	-	-	×	×	复位处理

×: 无关

第十八章 复位功能

以下四种操作可以用于产生复位信号。

- (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的内部复位
- (3) 通过比较上电清零 (POC) 电路的工作电压与检测电压的内部复位
- (4) 通过比较低电源检测器 (LVI) 的工作电压与检测电压的内部复位

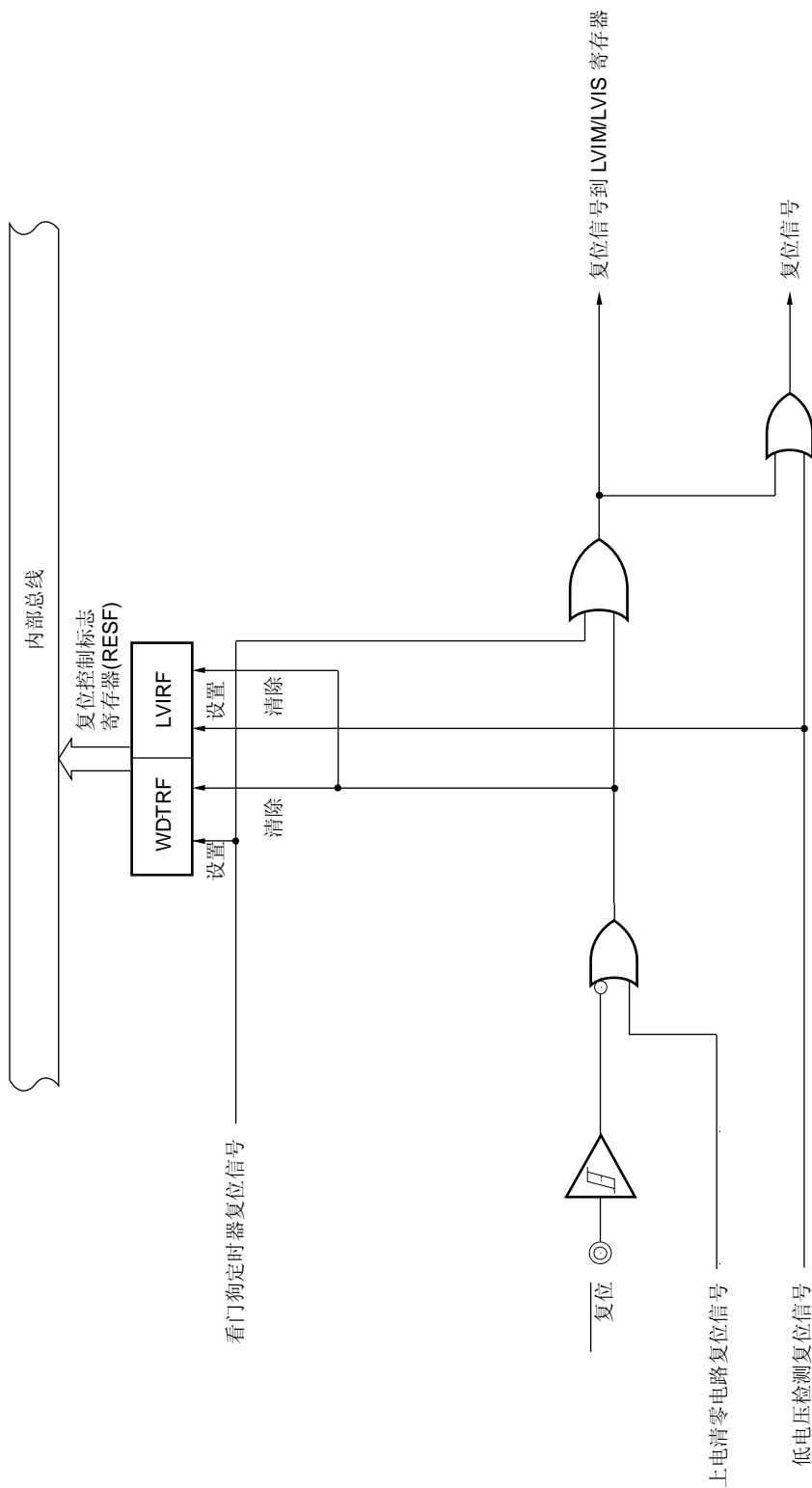
外部和内部复位没有功能差异。两种情况下，当产生复位信号时程序均从地址 0000H 和 0001H 处开始执行。

当低电平输入到 $\overline{\text{RESET}}$ 引脚时，看门狗定时器溢出时，或通过 POC 及 LVI 电路执行电压检测都可以引起复位，每项硬件的设置状态如表 18-1 及 18-2 所示。在产生复位信号期间或在复位释放后的振荡稳定时间内，每个引脚均为高阻态。

当低电平输入到 $\overline{\text{RESET}}$ 引脚时，设备复位。在复位处理后，当高电平输入到 $\overline{\text{RESET}}$ 引脚时，设备从复位状态中释放，且复位处理后以内部高速振荡时钟开始执行程序。将自动释放由看门狗定时器所产生的复位，并且在复位处理后，使用内部高速振荡时钟（参照图 18-2 到 18-4）开始执行程序。复位后，当 $V_{DD} \geq V_{POC}$ 或 $V_{DD} \geq V_{LVI}$ 时，将自动释放通过 POC 和 LVI 电源检测电路产生的复位，并且在复位处理后，使用内部高速振荡时钟（参照第十九章 上电清零电路 以及第二十章 低电压检测电路）开始执行程序。

- 注意事项**
1. 对于外部复位，输入到 $\overline{\text{RESET}}$ 引脚的低电平至少应为 $10\mu\text{s}$ 。
 2. 在复位输入期间，X1 时钟，内部高速振荡时钟以及内部低速振荡时钟将停止振荡。外部主系统时钟输入将变为无效。
 3. 当通过复位释放 STOP 模式时，在复位输入期间将保持 STOP 模式下的内容。但是，端口引脚变为高阻态。

图 18-1. 复位功能结构图



注意事项 LVI 电路内部复位不会复位 LVI 电路。

- 备注**
1. LVIM: 低压检测寄存器
 2. LVIS: 低压检测电平选择寄存器

图 18-2. 通过 RESET 输入的复位时序

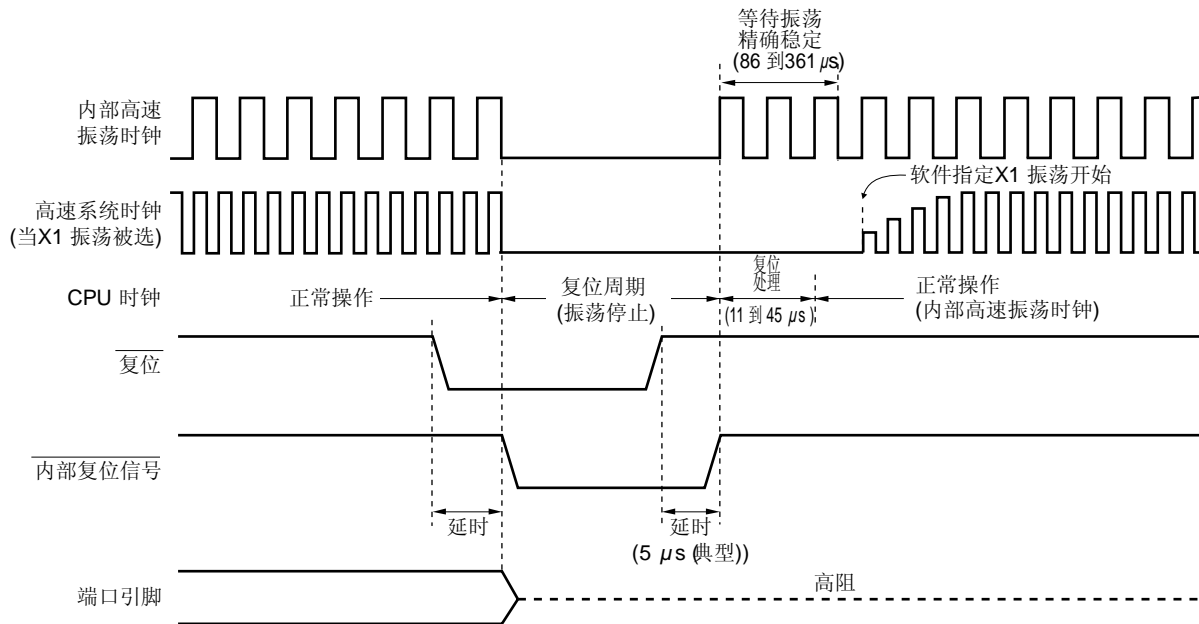
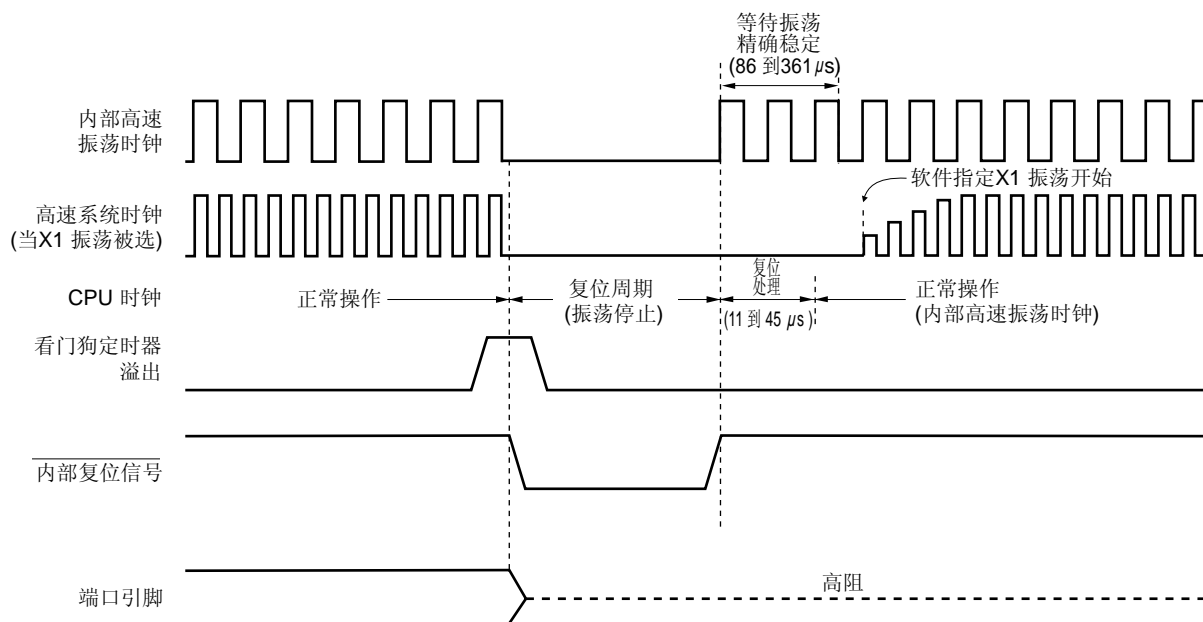
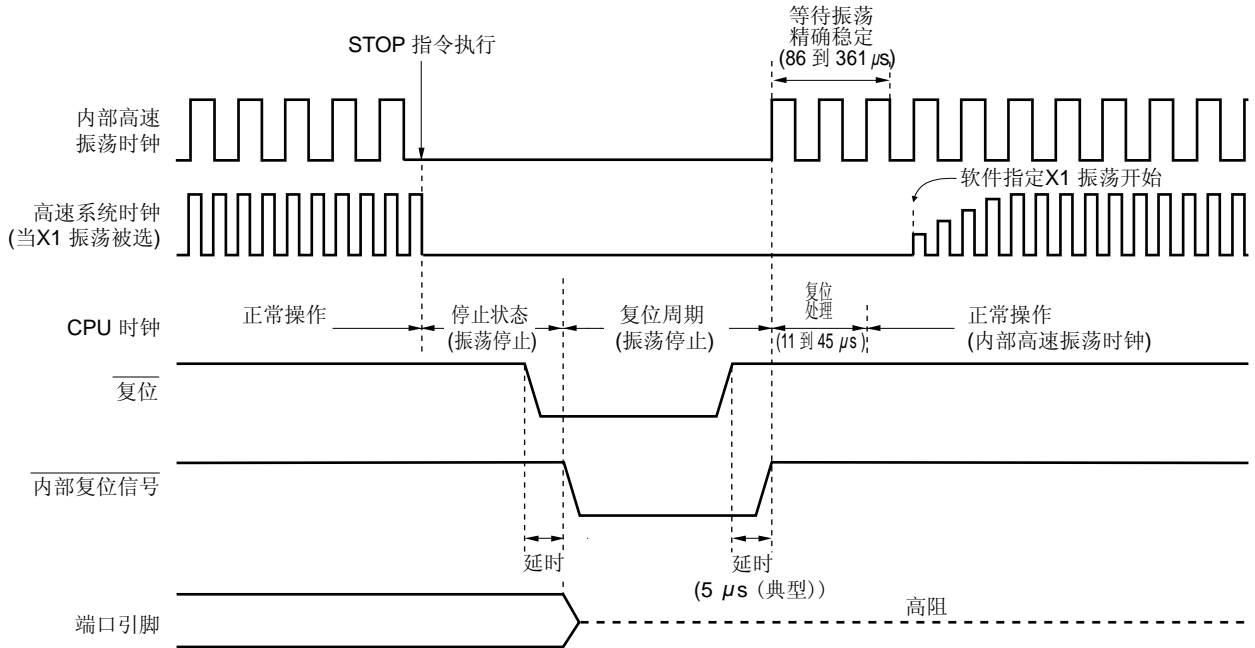


图 18-3. 由于看门狗定时器溢出而复位的时序



注意事项 看门狗定时器的内部复位会复位看门狗定时器。

图 18-4. 通过 $\overline{\text{RESET}}$ 输入在 STOP 模式下复位的时序



备注 关于上电清零电路和低电压检测电路的复位时序，请参照 **第十九章 上电清零电路** 和 **第二十章 低电压检测电路**。

表 18-1. 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}	停止运行	
	f _X	停止运行（引脚为输入 / 输出端口模式）	
	f _{EXCLK}	时钟输入无效（引脚为输入 / 输出端口模式）	
f _{RL}		停止运行	
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			
A/D 转换器			
串行接口	UART0		
	UART6		
	CSI10		
	IIC0		
上电清零功能			可运行
低压检测功能			停止运行
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 18-2. 复位响应后的硬件状态 (1 / 3)

硬件		复位响应后的状态 ^{注1}
程序计数器 (PC)		设置为复位向量表 (0000H, 0001H) 中的内容,
堆栈指针 (SP)		未定义
程序状态字 (PSW)		02H
RAM	数据存储器	未定义 ^{注2}
	通用寄存器	未定义 ^{注2}
端口寄存器 (P0 至 P3, P6 到 P12) (输出锁存器)		00H
端口模式寄存器 (PM0 到 PM3, PM6, PM12)		FFH
上拉电阻选择寄存器 (PU0, PU1, PU3, PU12)		00H
内部存储器容量切换寄存器 (IMS)		CFH ^{注3}
时钟运行模式选择寄存器 (OSCCTL)		00H
处理器时钟控制寄存器 (PCC)		01H
内部振荡模式寄存器 (RCM)		80H
主 OSC 控制寄存器 (MOC)		80H
主时钟模式寄存器 (MCM)		00H
振荡稳定时间计数器状态寄存器 (OSTC)		00H
振荡稳定时间选择寄存器 (OSTS)		05H
16 位定时器 / 事件计数器 00	定时器计数器 00 (TM00)	0000H
	捕捉 / 比较寄存器 000, 010 (CR000, CR010)	0000H
	模式控制寄存器 00(TMC00)	00H
	预分频模式寄存器 00(PRM00)	00H
	捕捉 / 比较控制寄存器 00 (CRC00)	00H
	定时器输出控制寄存器 00 (TOC00)	00H
8 位定时器 / 事件计数器 50, 51	定时器计数器 50, 51 (TM50, TM51)	00H
	比较寄存器 50, 51 (CR50, CR51)	00H
	定时器时钟选择寄存器 50, 51 (TCL50, TCL51)	00H
	模式控制寄存器 50, 51 (TMC50, TMC51)	00H

- 注
1. 在产生复位信号或振荡稳定时间等待期间内, 硬件状态只有 PC 的内容不确定。复位后其他的硬件状态都保持不变。
 2. 在待机模式下执行复位时, 即使在复位后也将保持预先复位状态。
 3. 复位释放后, 内部存储器容量切换寄存器 (IMS) 的初始值为 CFH, 与内部存储器容量无关。因此, 按照如下所示为每种产品设置相应的值。

产品	IMS
μPD78F8024	42H
μPD78F8025	C8H

<R>

表 18-2. 复位响应后的硬件状态 (2 / 3)

硬件		复位响应后的状态 ^{注1}
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}
A/D 转换器	10 位 A/D 转换结果寄存器 (ADCR)	0000H
	8 位 A/D 转换结果寄存器 (ADCRH)	00H
	模式寄存器 (ADM)	00H
	模拟输入通道选择寄存器 (ADS)	00H
	A/D 端口配置寄存器 (ADPC)	00H
串行接口 UART0	接收缓冲寄存器 0 (RXB0)	FFH
	发送移位寄存器 0 (TXS0)	FFH
	异步串行接口操作模式寄存器 0 (ASIM0)	01H
	异步串行接口接收错误状态寄存器 0 (ASIS0)	00H
	波特率发生器控制寄存器 0 (BRGC0)	1FH
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
	输入切换控制寄存器 (ISC)	00H
串行接口 CSI10	发送缓冲寄存器 10 (SOTB10)	00H
	串行输入 / 输出移位寄存器 10 (SIO10)	00H
	串行操作模式寄存器 10 (CSIM10)	00H
	串行时钟选择寄存器 10 (CSIC10)	00H

- 注
1. 在复位信号产生或振荡稳定时间等待期间内，硬件状态中只有 PC 的内容不确定。其他的硬件状态在复位后都保持不变。
 2. 仅 8 位定时器 H1
 3. 由选项字节的设置决定 WDTE 的复位值。

表 18-2. 复位应答后的硬件状态 (3 / 3)

硬件		复位响应后的状态 ^{注1}
串行接口 IIC0	位移寄存器 0 (IIC0)	00H
	控制寄存器 0 (IICC0)	00H
	从地址寄存器 0 (SVA0)	00H
	时钟选择寄存器 0 (IICCL0)	00H
	功能扩展寄存器 0 (IICX0)	00H
	状态寄存器 0 (IICS0)	00H
	标志寄存器 0 (IICF00)	00H
复位功能	复位控制标志寄存器 (RESF)	00H ^{注2}
低压检测	低压检测寄存器 (LVIM)	00H ^{注2}
	低压检测电平选择寄存器 (LVIS)	00H ^{注2}
中断	请求标志寄存器 0L, 0H, 1L, 1H (IF0L, IF0H, IF1L, IF1H)	00H
	屏蔽标志寄存器 0L, 0H, 1L, 1H (MK0L, MK0H, MK1L, MK1H)	FFH
	优先级指定标志寄存器 0L, 0H, 1L, 1H (PR0L, PR0H, PR1L, PR1H)	FFH
	外部中断上升允许寄存器 (EGP)	00H
	外部中断下降允许寄存器 (EGN)	00H

- 注
1. 在复位信号产生或振荡稳定时间等待期间内，硬件状态中只有 PC 的内容不确定。其他的硬件状态在复位后都保持不变。
 2. 这些值根据复位源的不同而变化。

复位源		RESET 输入	通过 POC 复位	通过 WDT 复位	通过 LVI 复位
RESF	WDTRF 标志	清除 (0)	清除 (0)	置位 (1)	保持
	LVIRF 标志			保持	置位 (1)
LVIM		清除 (00H)	清除 (00H)	清除 (00H)	保持
LVIS					

18.1 确定复位源的寄存器

在 μ PD78F8024 和 78F8025 中有许多内部复位源。复位控制标志寄存器 (RESF) 用于保存已经产生复位请求的复位源。

可以通过 8 位存储器操作指令读取 RESF。

$\overline{\text{RESET}}$ 输入，通过上电清零 (POC) 电路所产生的复位，以及读取 RESF 可以将 RESF 设为 00H。

图 18-5. 复位控制标志寄存器 (RESF) 的格式

地址: FFACH 复位后: 00H^注 R

符号	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LVIRF

WDTRF	通过看门狗定时器 (WDT) 所产生的内部复位请求
0	没有产生内部复位请求，或 RESF 被清除。
1	产生内部复位请求。

LVIRF	通过低电压检测电路 (LVI) 所产生的内部复位请求
0	没有产生内部复位请求，或 RESF 被清除。
1	产生内部复位请求。

注 复位后的值根据复位源的不同而不同。

注意事项 不能通过 1 位存储器操作指令来读取数据。

复位请求产生时 RESF 的状态如表 18-3 中所示。

图 18-3. 当复位请求产生时 RESF 的状态

标志 \ 复位源	$\overline{\text{RESET}}$ 输入	通过 POC 复位	通过 WDT 复位	通过 LVI 复位
WDTRF	清除 (0)	清除 (0)	置位 (1)	保持
LVIRF			保持	置位 (1)

第十九章 上电清零电路

19.1 上电清零电路的功能

上电清零电路（POC）有以下功能。

- 上电时产生内部复位信号。
在 1.59V POC 模式（选项字节：POCMODE = 0），当电源电压（V_{DD}）超过 $1.59\text{ V} \pm 1.5\text{ V}^{\text{‰}}$ 时，释放复位信号。
在 2.7V/1.59V POC 模式（选项字节：POCMODE = 1），当电源电压（V_{DD}）超过 $2.7\text{ V} \pm 0.2\text{ V}^{\text{‰}}$ 时，释放复位信号。
- 比较电源电压（V_{DD}）和检测电压（V_{POC} = 1.59 V ±0.15 V），当 V_{DD} < V_{POC} 时，产生内部复位信号。

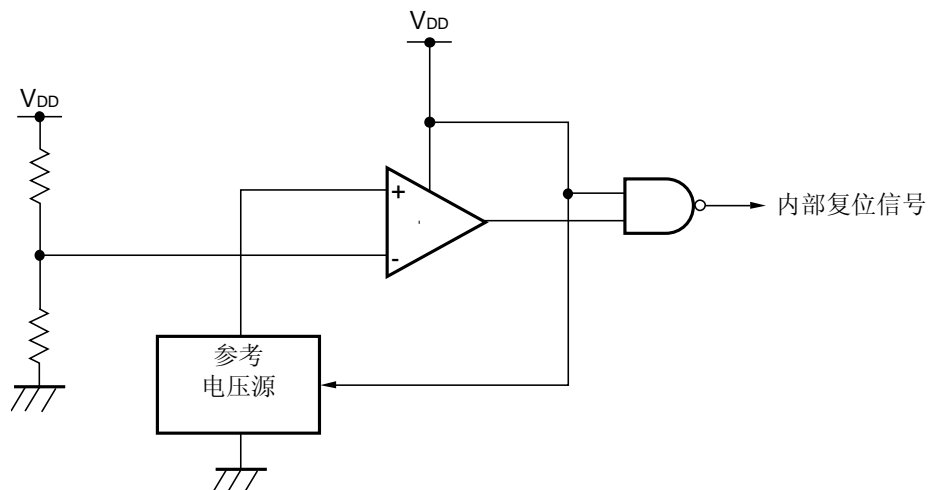
注意事项 如果在 POC 电路中产生内部复位信号，复位控制标志寄存器（RESF）清除为 00H。

备注 该产品包含多个产生内部复位信号的硬件功能。当通过看门狗定时器（WDT）、低电压检测电路（LVI）产生内部复位信号时，复位控制标志寄存器（RESF）中的标志表明复位源。当内部复位信号由 WDT 或 LVI 产生时，RESF 不会清为 00H，标志位设置为 1。
关于 RESF 的详细信息，参见 第十八章 复位功能。

19.2 上电清零电路的配置

上电清零电路的框图如图 19-1 所示。

图 19-1. 上电清零电路的框图



19.3 上电清零电路的操作

(1) 在 1.59V POC 模式（选项字节：POCMODE = 0）

- 上电时产生内部复位信号。当电源电压 (V_{DD}) 超过检测电压 ($V_{POC} = 1.59\text{ V} \pm 1.5\text{ V}$) 时，释放复位信号。
- 比较电源电压 (V_{DD}) 和检测电压 ($V_{POC} = 1.59\text{ V} \pm 1.5\text{ V}$)。当 $V_{DD} < V_{POC}$ 时，产生内部复位信号。当 $V_{DD} \geq V_{POC}$ 时复位释放。

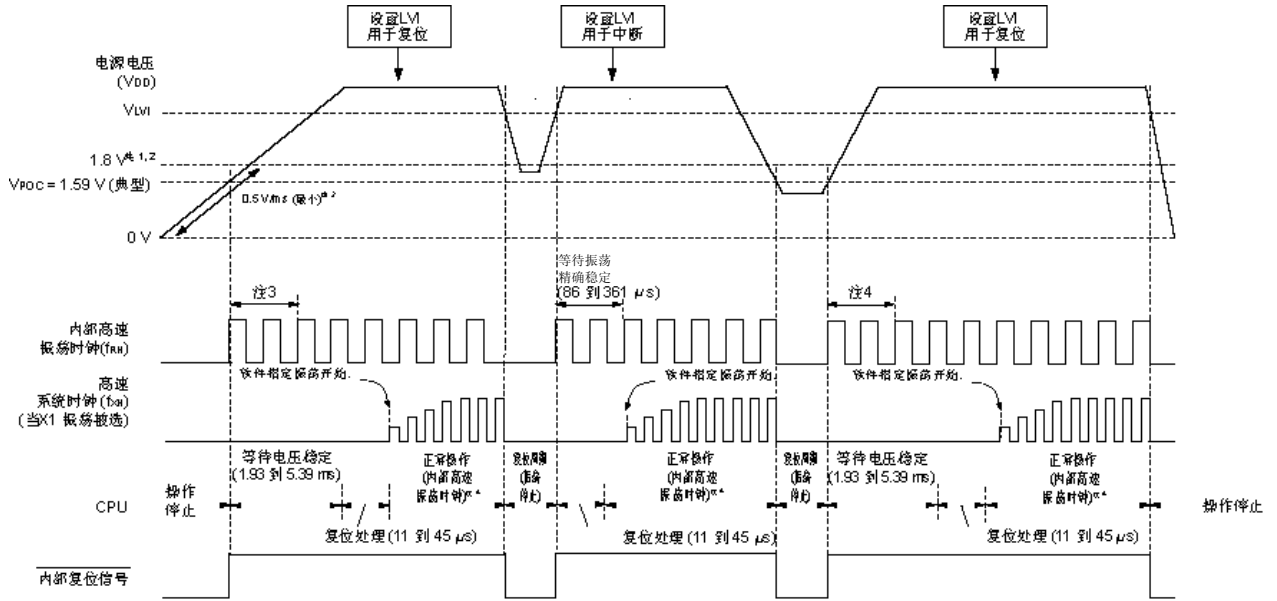
(2) 在 2.7V/1.59V POC 模式（选项字节：POCMODE = 1）

- 上电时产生内部复位信号。当电源电压 (V_{DD}) 超过检测电压 ($V_{DDPOC} = 2.7\text{ V} \pm 2\text{ V}$) 时，释放复位信号。
- 比较电源电压 (V_{DD}) 和检测电压 ($V_{POC} = 1.59\text{ V} \pm 1.5\text{ V}$)。当 $V_{DD} < V_{POC}$ 时，产生内部复位信号。当 $V_{DD} \geq V_{DDPOC}$ 时复位释放。

通过上电清零电路和低电压检测电路的内部复位信号的产生时序表示如下。

图 19-2. 通过上电清零电路和低电压检测电路产生内部复位信号的时序 (1 / 2)

(1) 在 1.59V POC 模式 (选项字节: POCMODE = 0)



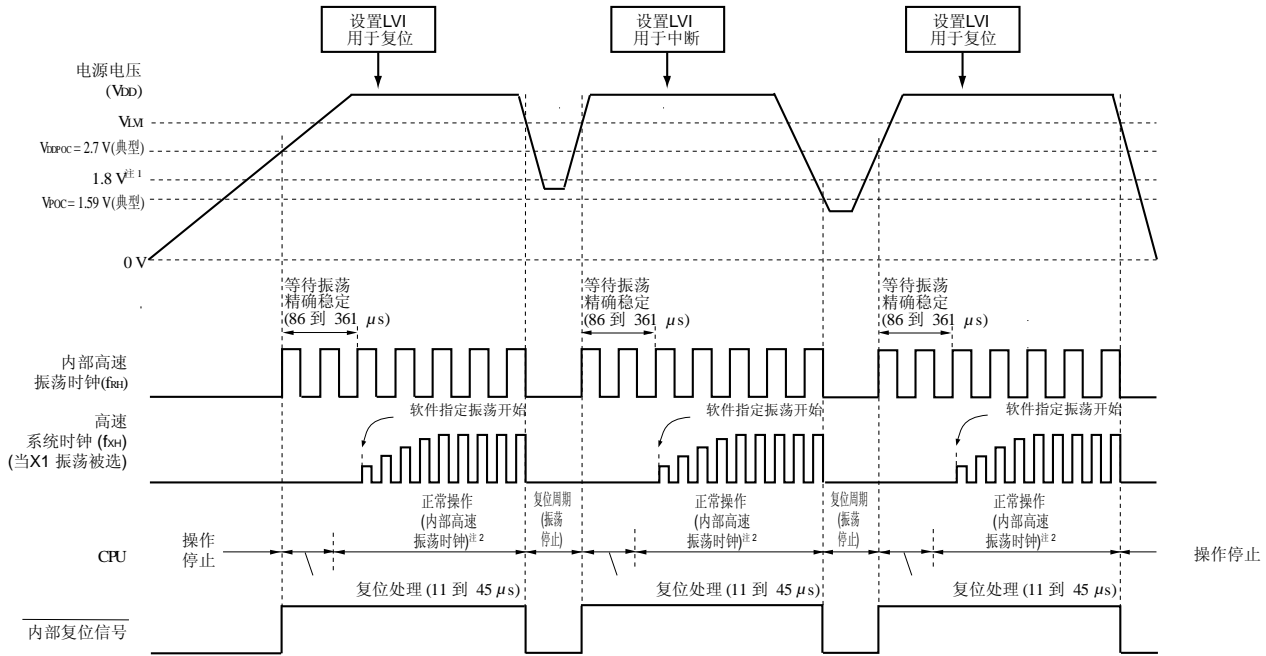
- 注
1. 保证操作范围是 $1.8 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$ 。在电源电压下降时, 要将电压范围低于工作保证范围时将状态设置为复位状态, 可以使用低电压检测电路的复位功能或者将低电平输入到 **RESET** 引脚。
 2. 如果上电时电压以低于 0.5 V/ms (最小) 的速率上升到 1.8V , 在上电后且电压到达 1.8V 前, 将低电平输入到 **RESET** 引脚, 或者使用选项字节 ($\text{POCMODE} = 1$) 设置 $2.7 \text{ V}/1.59 \text{ V}$ POC 模式。
 3. 内部电压稳定时间包含内部高速振荡时钟的振荡精确稳定时间。
 4. 可以选择内部高速振荡时钟或者高速系统时钟作为 CPU 时钟。要使用 X1 时钟, 使用 OSTC 寄存器来确认是否经历了振荡稳定时间。

注意事项 在复位状态释放后, 通过软件设置低电压检测电路 (参见 第二十章 低电压检测电路)。

备注 VLVI: LVI 检测电压
VPOC: POC 检测电压

图 19-2. 通过上电清零电路和低电压检测电路产生内部复位信号的时序 (2 / 2)

(2) 在 2.7V/1.59V POC 模式 (选项字节: POCMODE = 1)



- 注
1. 保证操作范围是 $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ 。在电源电压下降时，要将电压范围低于工作保证范围时将状态设置为复位状态，可以使用低电压检测电路的复位功能或者将低电平输入到 **RESET** 引脚。
 2. 可以选择内部高速振荡时钟或者高速系统时钟作为 CPU 时钟。要使用 X1 时钟，使用 OSTC 寄存器来确认是否经历了振荡稳定时间。

- 注意事项
1. 在释放复位状态后，通过软件设置低电压检测电路（参见第 20 章 低电压检测电路）。
 2. 电源电压达到 1.59V（典型）后，需要 1.93 到 5.39ms 的电压振荡稳定时间。如果上电电压在 1.93ms 内从 1.59V（典型）上升到 2.7V（典型），在复位处理前自动产生 0 到 5.39ms 的电源上电振荡稳定时间。

备注

V_{LVI}: LVI 检测电压
 V_{POC}: POC 检测电压

19.4 上电清零电路的注意事项

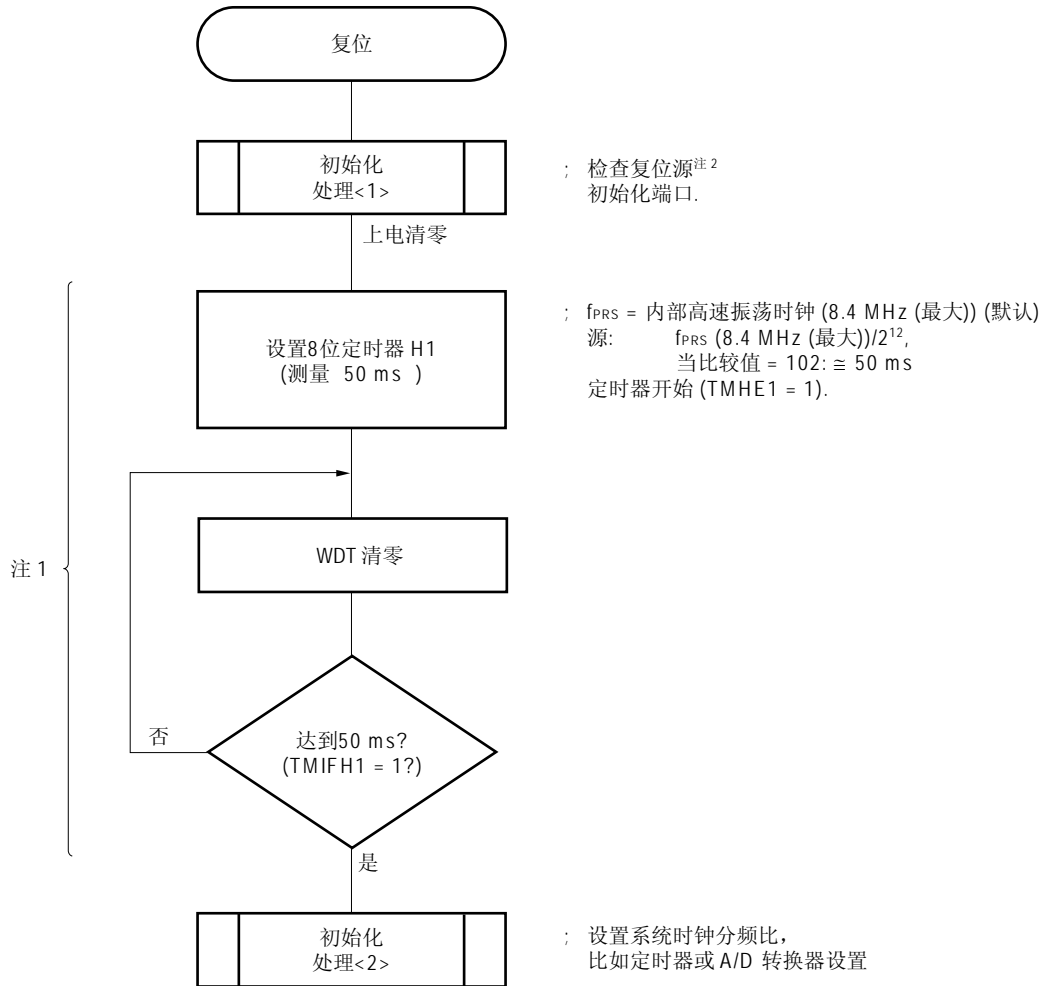
在一个电源电压 (V_{DD}) 以一定的周期在 POC 检测电压 (V_{Poc}) 附近波动的系统中, 系统可能会被重复复位并从复位状态被释放。在这种情况下, 可采用以下方法任意设置从复位释放到微控制器的启动所需要的时间。

<措施>

在释放复位信号后, 通过使用定时器的软件计数器的方法等待每个系统的电源电压波动周期, 然后初始化端口。

图 19-3. 复位释放后的软件处理举例 (1 / 2)

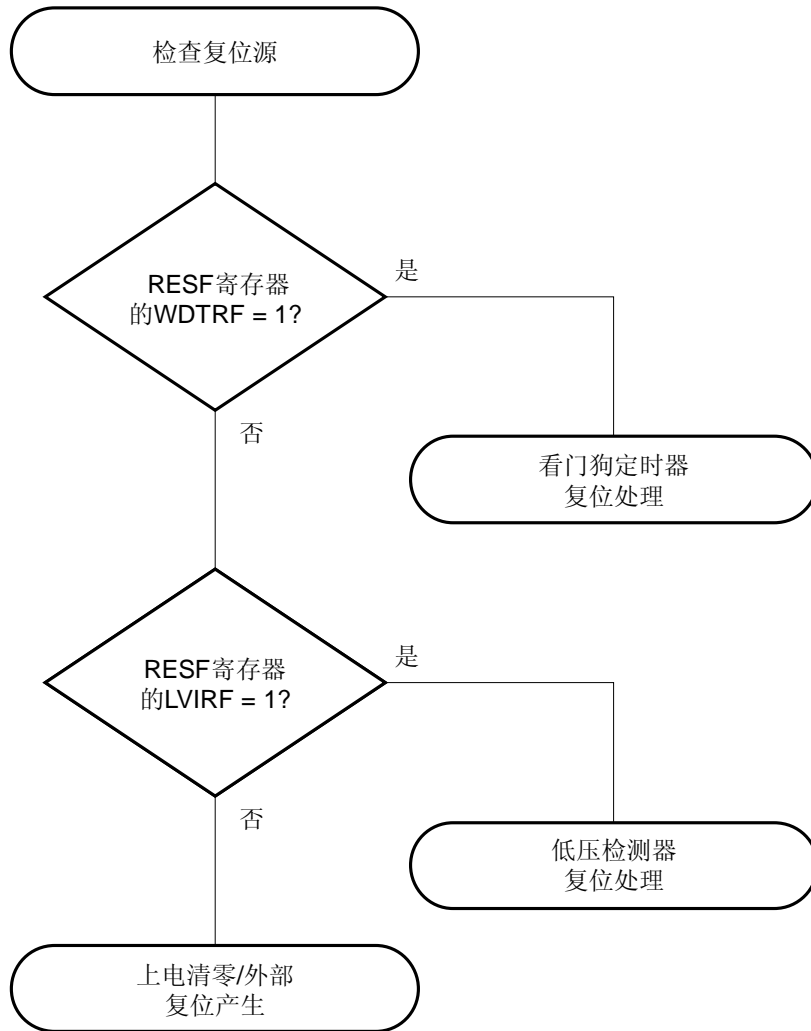
- 如果电源电压在 POC 检测电压附近的波动周期为 50ms 或更少



- 注 1. 如果在这个周期中再次产生复位, 不启动初始化处理<2>。
 注 2. 流程图在下页。

图 19-3. 复位释放后的软件处理举例（2 / 2）

• 检查复位源



第二十章 低电压检测电路

20.1 低电压检测电路的功能

低电压检测电路有以下功能。

- LVI 电路比较电源电压 (V_{DD}) 和检测电压 (V_{LVI}) 或者比较从外部输入引脚 ($EXLVI$) 输入的电压和检测电压 ($V_{EXLVI} = 1.21\text{ V}$ (典型): 定值), 并产生内部复位或内部中断信号。
- 可以通过软件来选择电源电压 (V_{DD}) 或从外部输入引脚 ($EXLVI$) 输入的电压。
- 可通过软件选择复位或中断功能。
- 可以通过软件来更改电源电压的检测电平 (16 个电平)。
- 在 STOP 模式下可以操作。

根据软件的选择, 按照下表产生复位和中断信号。

电源电压 (V_{DD}) 的电平检测选择 (LVISEL = 0)		从外部输入引脚 ($EXLVI$) 输入的电压的电平检测选择 (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, 产生一个内部复位信号, 当 $V_{DD} \geq V_{LVI}$ 时释放。	当 V_{DD} 下降到低于 V_{LVI} ($V_{DD} < V_{LVI}$) 或 V_{DD} 达到 V_{LVI} 或更高 ($V_{DD} \geq V_{LVI}$) 时, 产生一个内部中断信号。	当 $EXLVI < V_{EXLVI}$ 时, 产生一个内部复位信号, 当 $EXLVI \geq V_{EXLVI}$ 时, 释放复位信号。	当 $EXLVI$ 下降到低于 V_{EXLVI} ($EXLVI < V_{EXLVI}$) 或 $EXLVI$ 达到 V_{EXLVI} 或更高 ($EXLVI \geq V_{EXLVI}$) 时, 产生一个内部中断信号。

备注 LVISEL: 低电压检测寄存器 (LVIM) 的第 2 位
LVIMD: LVIM 的第 1 位

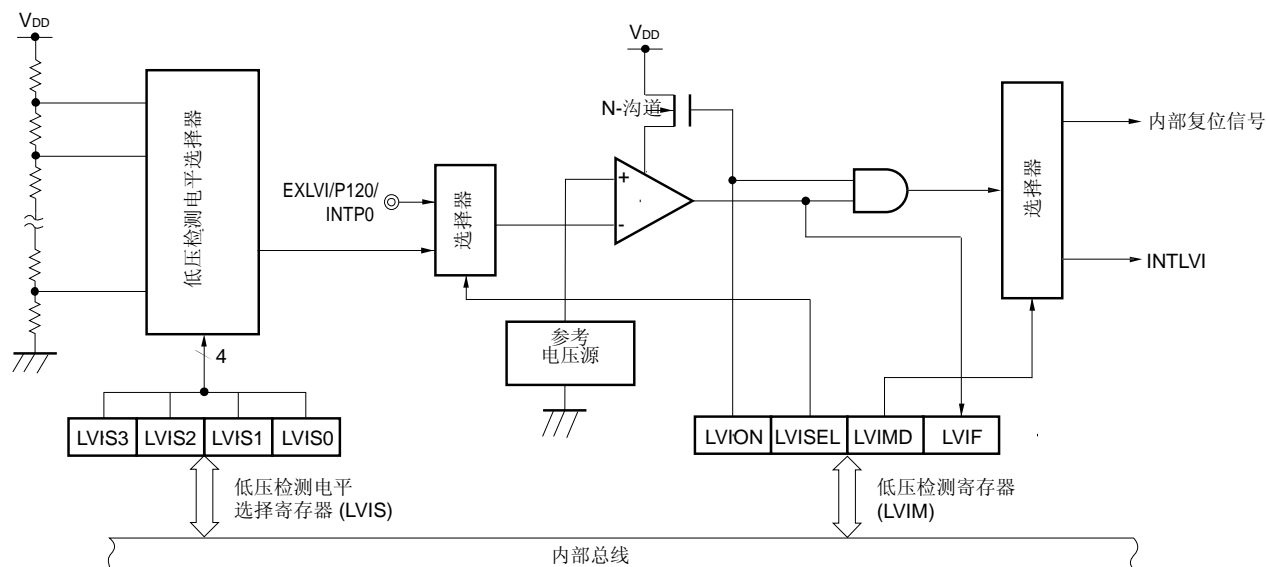
当低电压检测电路在工作时, 可以通过读取低电压检测标志 (LVIF: LVIM 的第 0 位) 来检查, 电源电压或从外部输入引脚 ($EXLVI$) 输入的电压比检测电平是高还是低。

当低电压检测电路用于复位时, 如果发生复位, 复位控制标志寄存器 (RESF) 的第 0 位 (LVIRF) 设置为 1。关于 RESF 的详细信息, 参见 **第十八章 复位功能**。

20.2 低电压检测电路的配置

低电压检测电路的框图如图 20-1 所示。

图 20-1. 低电压检测电路的框图



20.3 控制低电压检测电路的寄存器

低电压检测电路由以下寄存器控制。

- 低电压检测寄存器 (LVIM)
- 低电压检测电平选择寄存器 (LVIS)
- 端口模式寄存器 12 (PM12)

(1) 低电压检测寄存器 (LVIM)

这个寄存器设置低电压检测和工作模式。

可以通过一个 1 位或 8 位存储器操作指令设置这个寄存器。

除 LVI 复位之外的复位信号清除该寄存器为 00H。

图 20-2. 低电压检测寄存器 (LVIM) 的格式

地址: FFBEH 复位后: 00H^{注1} R/W^{注2}

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

LVION ^{注3, 4}	允许低电压检测操作
0	禁止操作
1	允许操作

LVISEL ^{注3}	电压检测选择
0	检测电源电压 (V _{DD}) 的电平
1	检测从外部输入引脚 (EXLVI) 输入的电压的电平

LVIMD ^{注3}	低电压检测工作模式 (中断 / 复位) 选择
0	<ul style="list-style-type: none"> LVISEL = 0: 当电源电压 (V_{DD}) 下降到低于检测电压 (V_{LVI}) (V_{DD} < V_{LVI}) 或 V_{DD} 达到 V_{LVI} 或更高 (V_{DD} ≥ V_{LVI}) 时, 产生一个内部中断信号。 LVISEL = 1: 当从外部输入引脚 (EXLVI) 输入的电压下降到低于检测电压 (V_{EXLVI}) (V_{DD} < V_{LVI}) 或 EXLVI 达到 V_{EXLVI} 或更高 (EXLVI ≥ V_{EXLVI}) 时, 产生一个中断信号。
1	<ul style="list-style-type: none"> LVISEL = 0: 当电源电压 (V_{DD}) < 检测电压 (V_{LVI}) 时, 产生一个内部复位信号, 当 V_{DD} ≥ V_{LVI} 时释放复位信号。 LVISEL = 1: 当从外部输入引脚 (EXLVI) 输入的电压 < 检测电压 (V_{LVI}) 时, 产生一个内部复位信号, 当 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: 电源电压 (V_{DD}) ≥ 检测电压 (V_{LVI}), 或者当操作被禁止时 LVISEL = 1: 从外部输入引脚 (EXLVI) 输入的电压 ≥ 检测电压 (V_{EXLVI}), 或者当操作被禁止时
1	<ul style="list-style-type: none"> LVISEL = 0: 电源电压 (V_{DD}) < 检测电压 (V_{LVI}) LVISEL = 1: 从外部输入引脚 (EXLVI) 输入的电压 < 检测电压 (V_{EXLVI})

- 注
1. 该位在产生除 LVI 复位以外的复位时清除为 00H。
 2. 第 0 位只读。
 3. 在 LVI 复位以外的复位情况下, LVION、LVIMD 和 LVISEL 清除为 0。在 LVI 复位的情况下, 这些不会清除为 0。
 4. 当 LVION 设置为 1 时, 启动 LVI 电路的比较器的操作。通过软件来等待从 LVION 置为 1 直到操作稳定的稳定时间 (10μs (最小))。操作稳定后, 从进入低于 LVI 检测电压状态直到 LVIF 设置为 1, 需要 200μs (最小) 的外部输入 (最小脉冲宽度: 200μs (最小))。

- 注意事项
1. 要停止 LVI, 遵循以下过程之一。
 - 当使用 8 位存储器操作指令时: 写入 00H 到 LVIM。
 - 当使用 1 位存储器操作指令时: 清除 LVION 为 0。
 2. 从外部输入引脚 (EXLVI) 输入的电压必须满足 EXLVI < V_{DD}。
 3. 当使用 LVI 作为一个中断 (LVIMD = 0) 时, 如果在低于 LVI 检测电压状态, LVION 清零, 将产生 INTLVI 信号且 LVIF 变为 1。
 4. 当使用 LVI 作为一个复位信号 (LVIMD = 1) 时, 在允许 LVI 操作 (LVION = 1) 后, 不要写 LVIM 寄存器。

(2) 低电压检测电平选择寄存器 (LVIS)

这个寄存器选择低电压检测电平。

可以通过一个 1 位或 8 位存储器操作指令设置这个寄存器。

除 LVI 复位以外的复位信号的产生将清除该寄存器为 00H。

图 20-3. 低电压检测电平选择寄存器 (LVIS) 的格式

地址: FFBFH 复位后: 00H^注 R/W

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

LVIS3	LVIS2	LVIS1	LVIS0	检测电平
0	0	0	0	V _{LV10} (4.24 V ±0.1 V)
0	0	0	1	V _{LV11} (4.09 V ±0.1 V)
0	0	1	0	V _{LV12} (3.93 V ±0.1 V)
0	0	1	1	V _{LV13} (3.78 V ±0.1 V)
0	1	0	0	V _{LV14} (3.62 V ±0.1 V)
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} (2.08 V ±0.1 V)
1	1	1	1	V _{LV115} (1.93 V ±0.1 V)

注 LVIS 的值在 LVI 复位时不复位，而是保持原值。其它复位时清除为 00H。

- 注意事项
1. 确认清除第 4 到 7 位为“0”。
 2. 在 LVI 操作时不要改变 LVIS 的值。
 3. 当检测从外部输入引脚 (EXLVI) 输入的电压时，检测电压 (V_{EXLVI} = 1.21 V (典型)) 是固定的。因此，不需要设置 LVIS。

(3) 端口模式寄存器 12 (PM12)

当使用 P120 / EXLVI / INTP0 引脚作为外部低电压检测电压输入时，设置 PM120 为 1。这时，P120 的输出锁存器可能是 0 或 1。

可以通过一个 1 位或 8 位存储器操作指令设置 PM12。

复位信号的产生设置 PM12 为 FFH。

图 20-4. 端口模式寄存器 12 (PM12) 的格式

地址: FF2CH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM12	1	1	1	PM124	PM123	PM122	PM121	PM120

PM12n	P12n 引脚输入 / 输出模式选择 (n = 0~2)
0	输出模式 (输出缓冲器开)
1	输入模式 (输出缓冲器关)

20.4 低电压检测电路的操作

低电压检测电路可以用作以下两种模式。

(1) 用作复位 (LVIMD = 1)

- 如果 LVISEL = 0，比较电源电压 (V_{DD}) 和检测电压 (V_{LVI})，当 V_{DD} < V_{LVI} 时产生内部复位信号，当 V_{DD} ≥ V_{LVI} 时释放内部复位信号。
- 如果 LVISEL = 1，比较外部输入引脚 (EXLVI) 输入的电压和检测电压 (V_{EXLVI} = 1.21 V (典型))，当 EXLVI < V_{EXLVI} 时产生内部复位信号，当 EXLVI ≥ V_{EXLVI} 时释放内部复位。

(2) 用作中断 (LVIMD = 0)

- 如果 LVISEL = 0，比较电源电压 (V_{DD}) 和检测电压 (V_{LVI})。当 V_{DD} 下降到低于 V_{LVI} (V_{DD} < V_{LVI}) 或 V_{DD} 达到 V_{LVI} 或更高 (V_{DD} ≥ V_{LVI}) 时，产生一个中断信号 (INTLVI)。
- 如果 LVISEL = 1，比较外部输入引脚 (EXLVI) 输入的电压和检测电压 (V_{EXLVI} = 1.21 V (典型))。当 EXLVI 下降低于 V_{EXLVI} (EXLVI < V_{EXLVI}) 时或 EXLVI 达到 V_{EXLVI} 或更高 (EXLVI ≥ V_{EXLVI}) 时，产生中断信号 (INTLVI)。

当低电压检测电路在工作时，电源电压或从外部输入引脚 (EXLVI) 输入的电压比检测电平是高还是低，可以通过读取低电压检测标志 (LVIF: LVIM 的第 0 位) 来检查。

备注 LVIMD: 低电压检测寄存器 (LVIM) 的第 1 位
LVISEL: LVIM 的第 2 位

20.4.1 当用作复位时

(1) 当检测电源电压 (V_{DD}) 的电平时

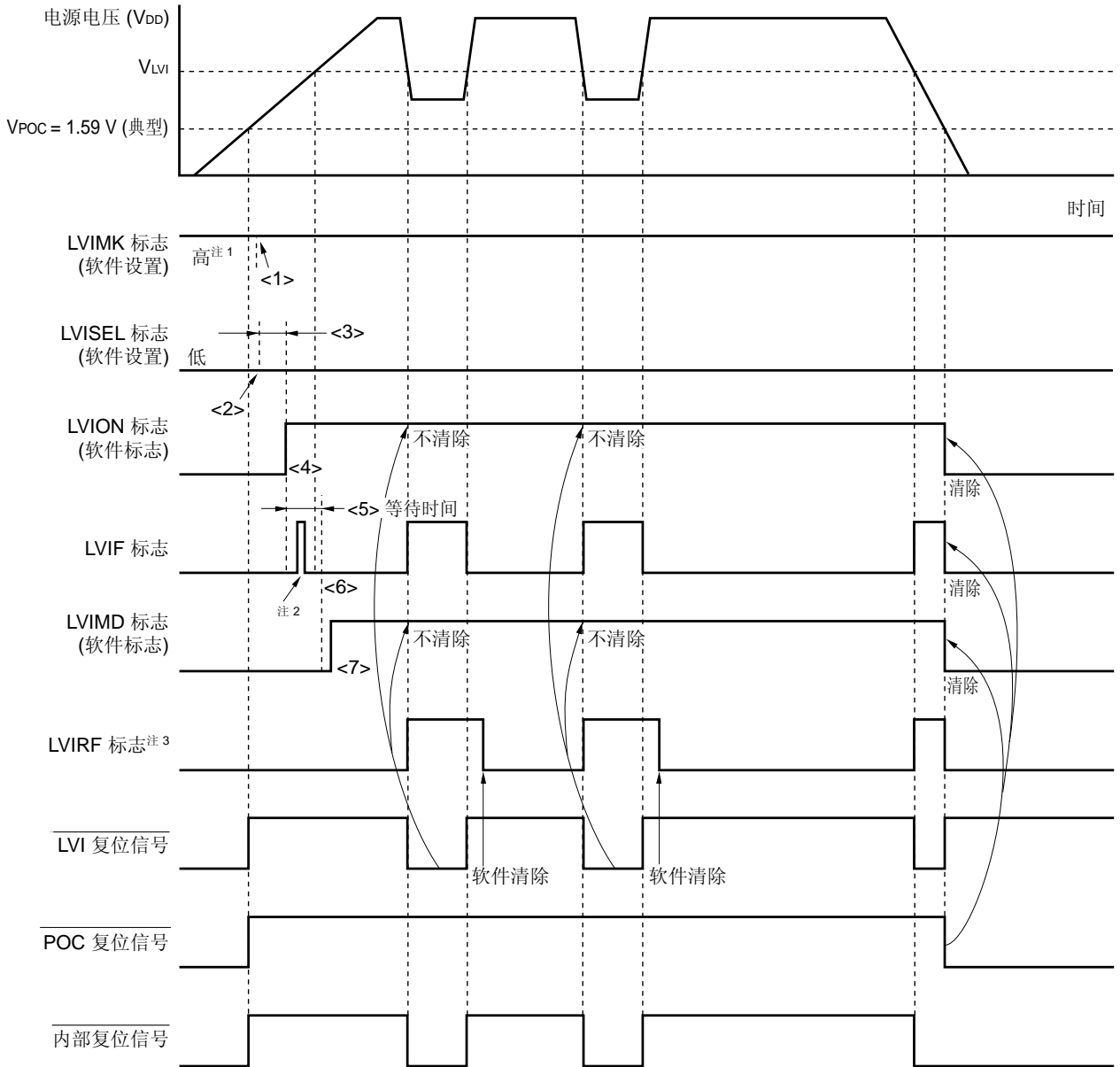
- 当启动操作时
 - <1> 在停止的状态下 (LVION = 0) 检测 LVI 的操作。当 LVION = 1 时, 不要执行如下<2> 至 <8> 的设置。
 - <2> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <3> 清除低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 为 0 (检测电源电压 (V_{DD}) 的电平) (默认值)。
 - <4> 使用低电压检测电平选择寄存器 (LVIS) 的第 3 到 0 位 (LVIS3 到 LVIS0) 来设置检测电压。
 - <5> 设置 LVIM 的第 7 位 (LVION) 为 1 (允许 LVI 操作)
 - <6> 通过软件等待操作稳定时间 (10 μ s (最小))。
 - <7> 等待直到通过 LVIM 的第 0 位 (LVIF) 检查到 (电源电压 (V_{DD}) \geq 检测电压 (V_{LVI}))。
 - <8> 设置 LVIM 的第 1 位 (LVIMD) 为 1 (当检测电平时, 产生复位)。

图 20-5 表示内部复位信号被低电压检测电路产生的时序。这个时序图中的号码对应上面的<2>到<8>。

- 注意事项**
1. 必须执行<1>和<2>。当 LVIMK = 0 时, 可能在<5>中的处理后立即产生中断。
 2. 当 LVIMD 设置为 1 时, 如果电源电压 (V_{DD}) \geq 检测电压 (V_{LVI}), 不会产生内部复位信号。

图 20-5. 低压检测器的内部复位信号产生时序（检测电源电压（V_{DD}））（1 / 2）

(1) 1.59V POC 模式（选项字节：POCMODE = 0）

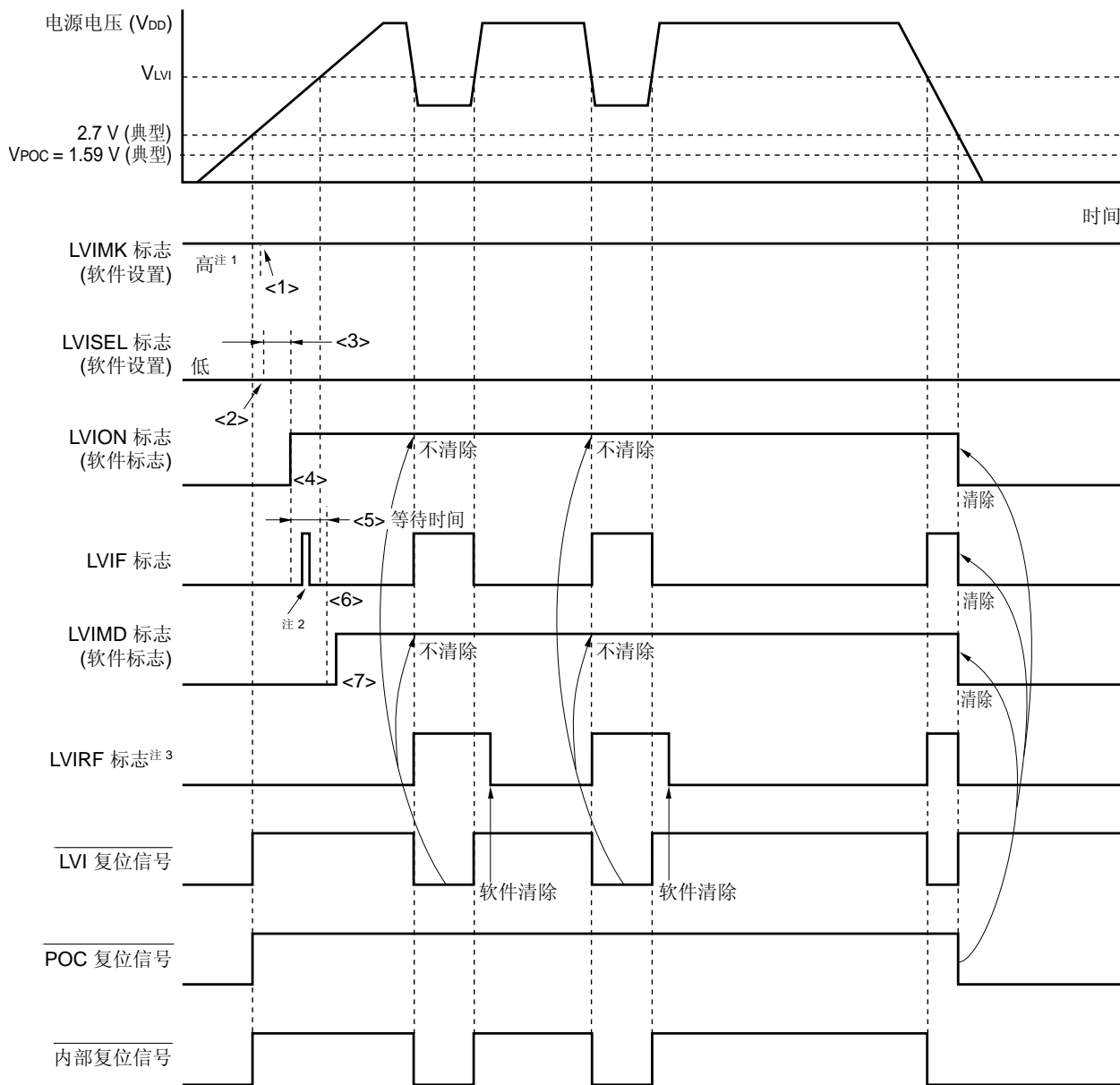


- 注
1. LVIMK 标志被复位信号设置为“1”。
 2. LVIF 标志可设置为 1。
 3. LVIRF 是复位控制标志寄存器（RESF）的第 0 位。关于 RESF 的详细信息，参见第十八章 复位功能。

备注 上面图 20-5 中的<2>到<8>对应于 20.4.1 (1) 当检测电源电压（V_{DD}）的电平时中的“当启动操作时”中描述的<2>到<8>。

图 20-5. 低压检测器的内部复位信号产生时序 (检测电源电压 (V_{DD})) (2 / 2)

(2) 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1)



- 注
1. LVIMK 标志被复位信号设置为“1”。
 2. LVIF 标志可设置为 1。
 3. LVIRF 是复位控制标志寄存器 (RESF) 的第 0 位。关于 RESF 的详细信息, 参见第十八章 复位功能。

备注 上面图 20-5 中的<2>到<8>对应于 20.4.1 (1) 当检测电源电压 (V_{DD}) 的电平中的“当启动操作时”的描述的<2>到<8>。

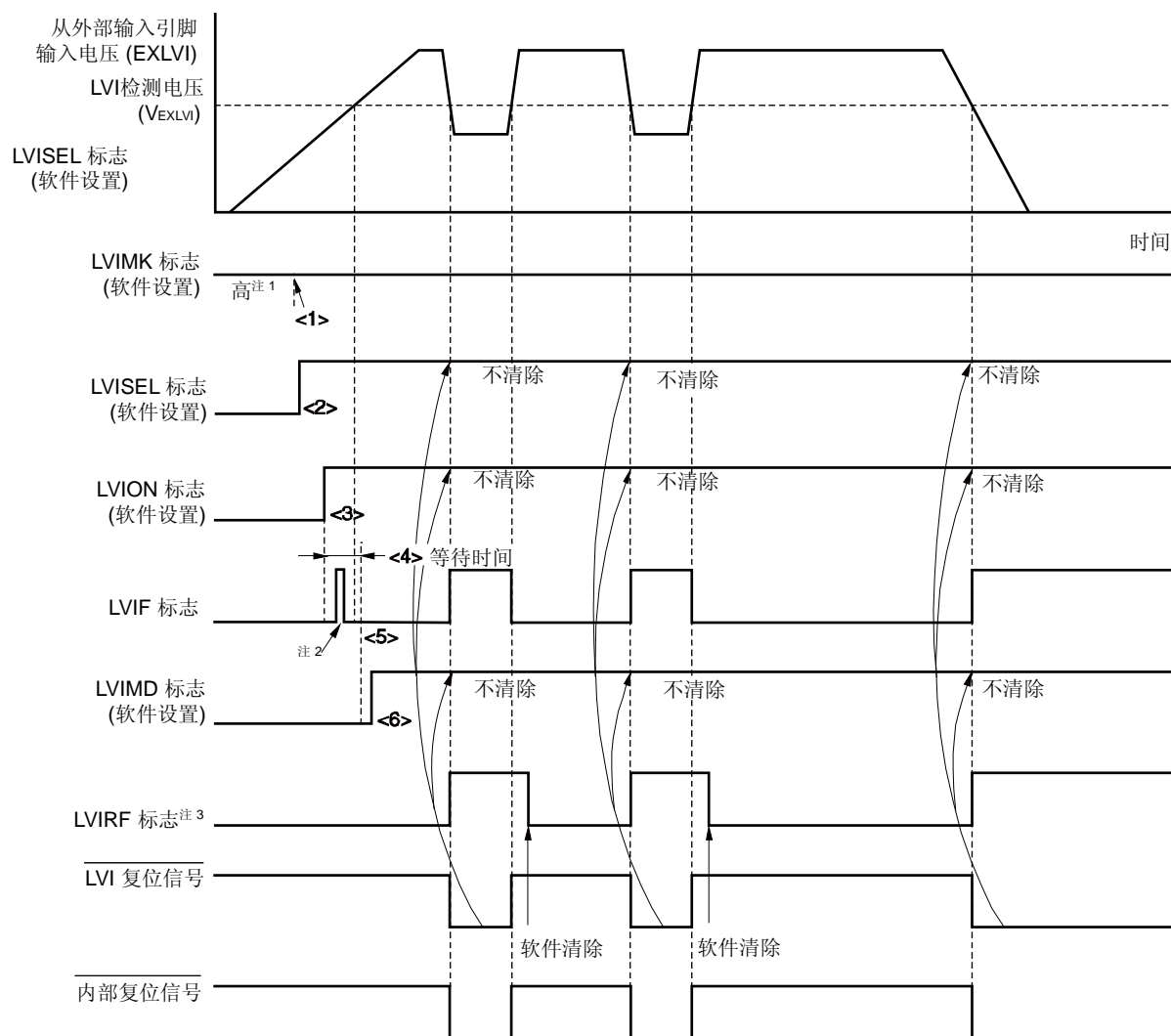
(2) 当检测从外部输入引脚 (EXLVI) 输入的电压的电平时

- 当启动操作时
 - <1> 在停止的状态下 (LVION = 0) 检测 LVI 的操作。当 LVION = 1 时, 不要执行如下<2> 至 <7> 的设置。
 - <2> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <3> 设置低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 为 1 (检测从外部输入引脚 (EXLVI) 输入的电压的电平)
 - <4> 设置 LVIM 的第 7 位 (LVION) 为 1 (允许 LVI 操作)
 - <5> 通过软件等待操作稳定时间 (10 μ s (最小))。
 - <6> 等待直到通过 LVIM 的第 0 位 (LVIF) 检查到 (从外部输入引脚 (EXLVI) 输入的电压 \geq 检测电压 (V_{EXLVI} = 1.21 V (典型)))。
 - <7> 设置 LVIM 的第 1 位 (LVIMD) 为 1 (当电平被检测时, 产生复位)。

图 20-6 表示内部复位信号被低电压检测电路产生的时序。这个时序图中的号码对应上面的<2>到<7>。

- 注意事项**
1. 必须执行<1>和<2>。当 LVIMK = 0 时, 可能在<4>中的处理后立即产生中断。
 2. 当 LVIMD 设置为 1 时, 如果从外部输入引脚 (EXLVI) 输入的电压 \geq 检测电压 (V_{EXLVI} = 1.21 V (典型)), 不会产生内部复位信号。
 3. 从外部输入引脚 (EXLVI) 输入的电压必须满足 EXLVI < V_{DD}。

图 20-6. 低压检测器的内部复位信号产生时序
(检测从外部输入引脚输入电压的电平 (EXLVI))



- 注
1. LVIMK 标志被复位信号设置为“1”。
 2. LVIF 标志可设置为 1。
 3. LVIRF 标志是复位控制标志寄存器 (RESF) 的第 0 位。关于 RESF 的详细信息，参见 第十八章 复位功能。

备注 上面图 20-6 中的 <2>到<7>对应于 20.4.1 (2) 当检测从外部输入引脚 (EXLVI) 输入的电压的电平时中的“当启动操作时”中描述的<2>到<7>。

20.4.2 当用作中断时

(1) 当检测电源电压 (V_{DD}) 的电平时

- 当启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <2> 清除低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 为 0 (检测电源电压 (V_{DD}) 的电平) (默认值)。
 - <3> 使用低电压检测电平选择寄存器 (LVIS) 的第 3 到 0 位 (LVIS3 到 LVIS0) 来设置检测电压。
 - <4> 清除 LVIM 的第 1 位 (LVIMD) 为 0 (当检测电平时, 产生中断信号) (默认值)。
 - <5> 设置 LVIM 的第 7 位 (LVION) 为 1 (允许 LVI 操作)
 - <6> 通过软件等待操作稳定时间 (10 μ s (最小))。
 - <7> 使用 LVIM 的第 0 位 (LVIF), 当检测 V_{DD} 的下降沿时, 确认 “电源电压 (V_{DD}) \geq 检测电压 (V_{LVI})”, 或者当检测 V_{DD} 的上升沿时, 确认 “电源电压 (V_{DD}) < 检测电压 (V_{LVI})”。
 - <8> 清除 LVI 的中断请求标志 (LVIIF) 为 0。
 - <9> 释放 LVI 的中断屏蔽标志 (LVIMK)。
 - <10> 执行 EI 指令 (当使用向量中断时)。

图 20-7 表示低电压检测电路产生中断信号的时序。这个时序图中的号码对应上面的<1>到<9>。

- 当停止操作时

必须执行下面任一过程。

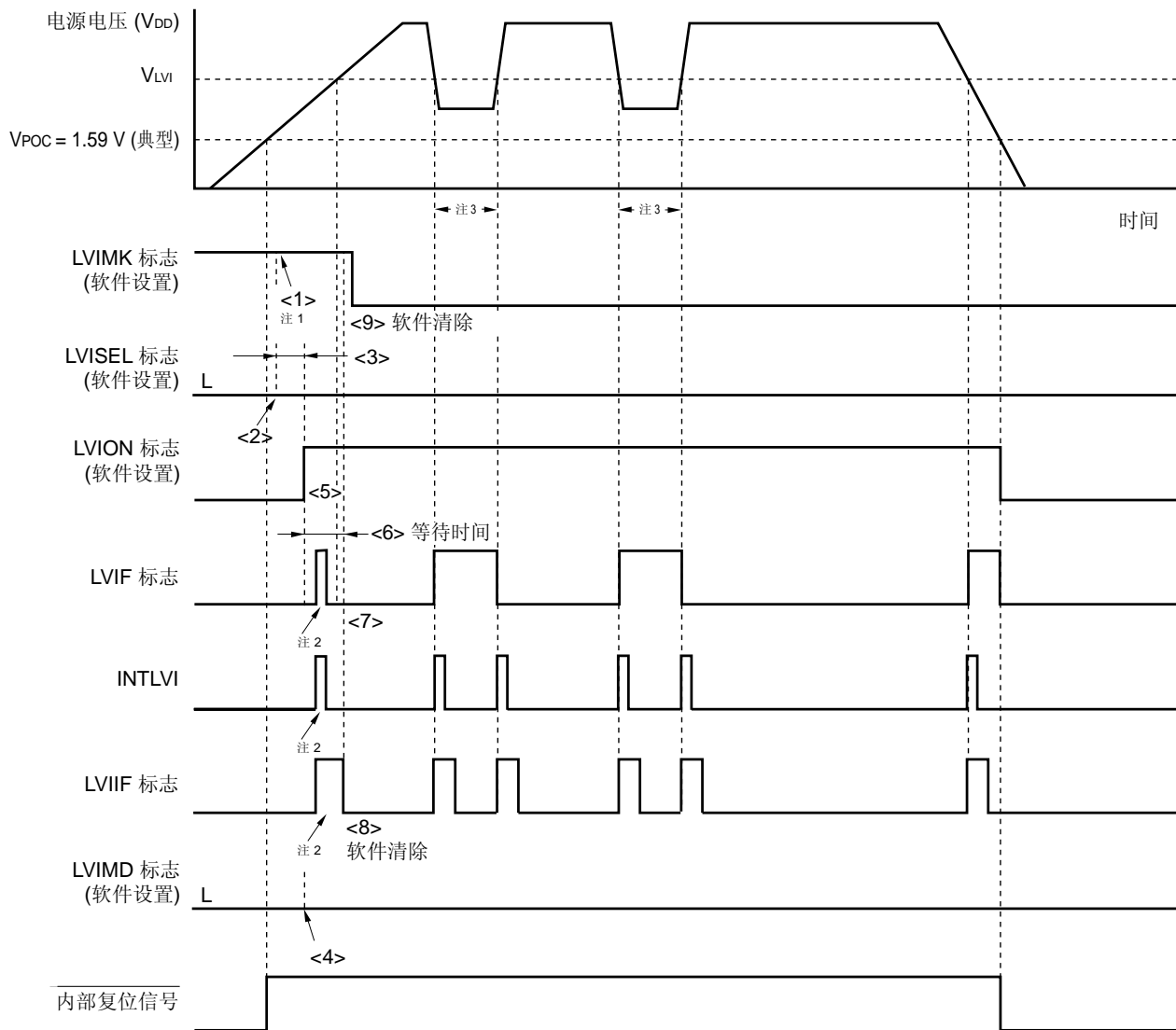
 - 当使用 8 位存储器操作指令时:

写入 00H 到 LVIM。
 - 当使用 1 位存储器操作指令时:

清除 LVION 为 0。

图 20-7. 低压检测器的中断信号产生时序
(检测电源电压 (V_{DD})) (1 / 2)

(1) 1.59V POC 模式 (选项字节: POCMODE = 0)

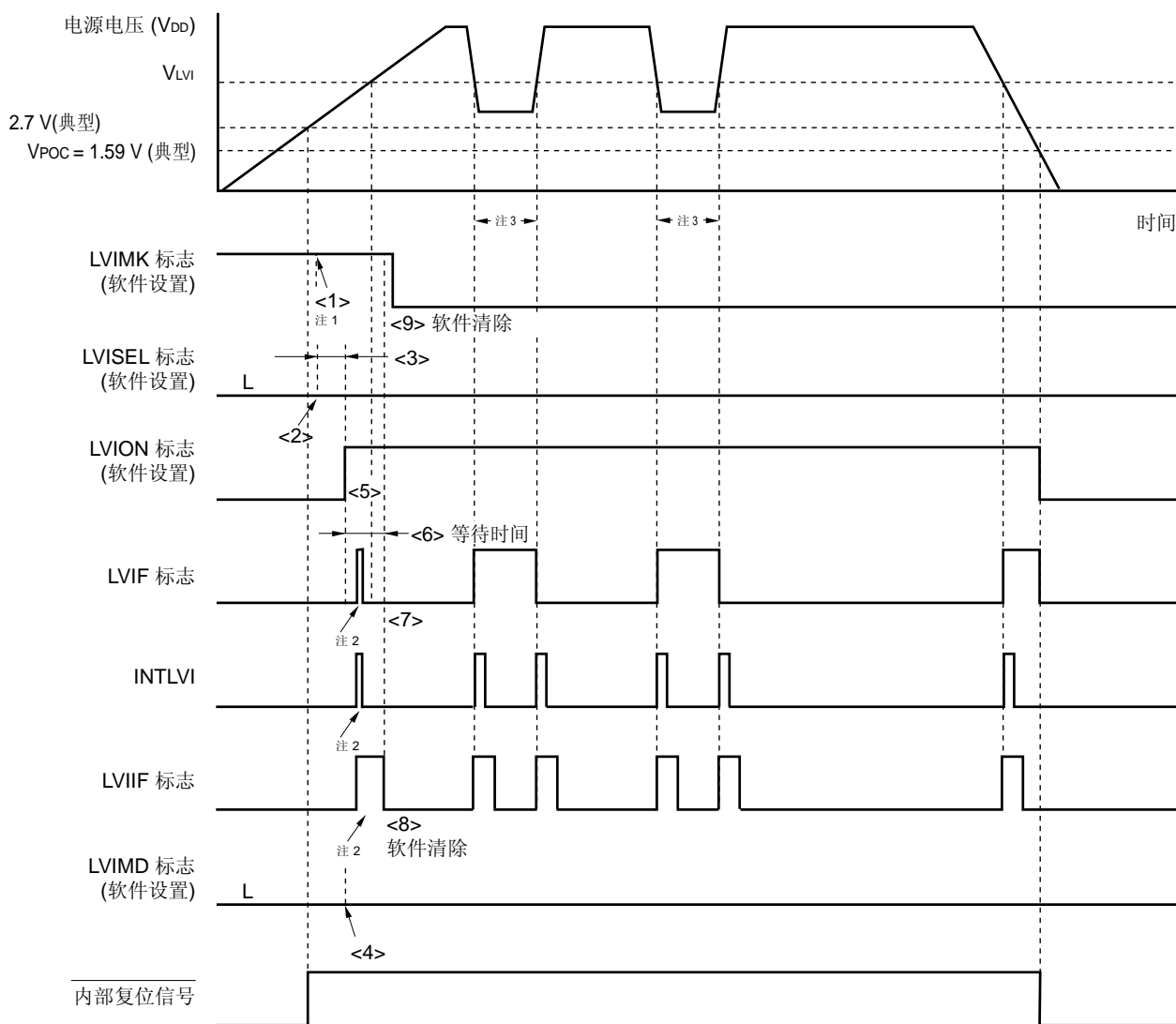


- 注
1. 复位信号将 LVIMK 标志设置为“1”。
 2. 中断请求标志寄存器的 LVIIF 标志和 LVIF 标志可能置位 (1)。
 3. 如果在低于 LVI 检测电压状态 LVION 清零, 将产生 INTLVI 信号且 LVIIF 变为 1。

备注 上面图 20-7 中的<1>到<9>对应于 20.4.2 (1) 当检测电源电压 (V_{DD}) 的电平时中的“当启动操作时”中描述的<1>到<9>。

图 20-7. 低压检测器的中断信号产生时序
(检测电源电压 (V_{DD})) (2 / 2)

(2) 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1)



- 注
1. LVIMK 标志被复位信号设置为“1”。
 2. 中断请求标志寄存器的 LVIIIF 标志和 LVIF 标志可能置位 (1)。
 3. 如果在低于 LVI 检测电压状态 LVION 清零, 将产生 INTLVI 信号且 LVIIIF 变为 1。

备注 上面图 20-7 中的<1>到<9>对应于 20.4.2 (1) 当检测电源电压 (V_{DD}) 的电平时中的“当启动操作时”中描述的<1>到<9>。

(2) 当检测从外部输入引脚 (EXLVI) 输入的电压的电平时

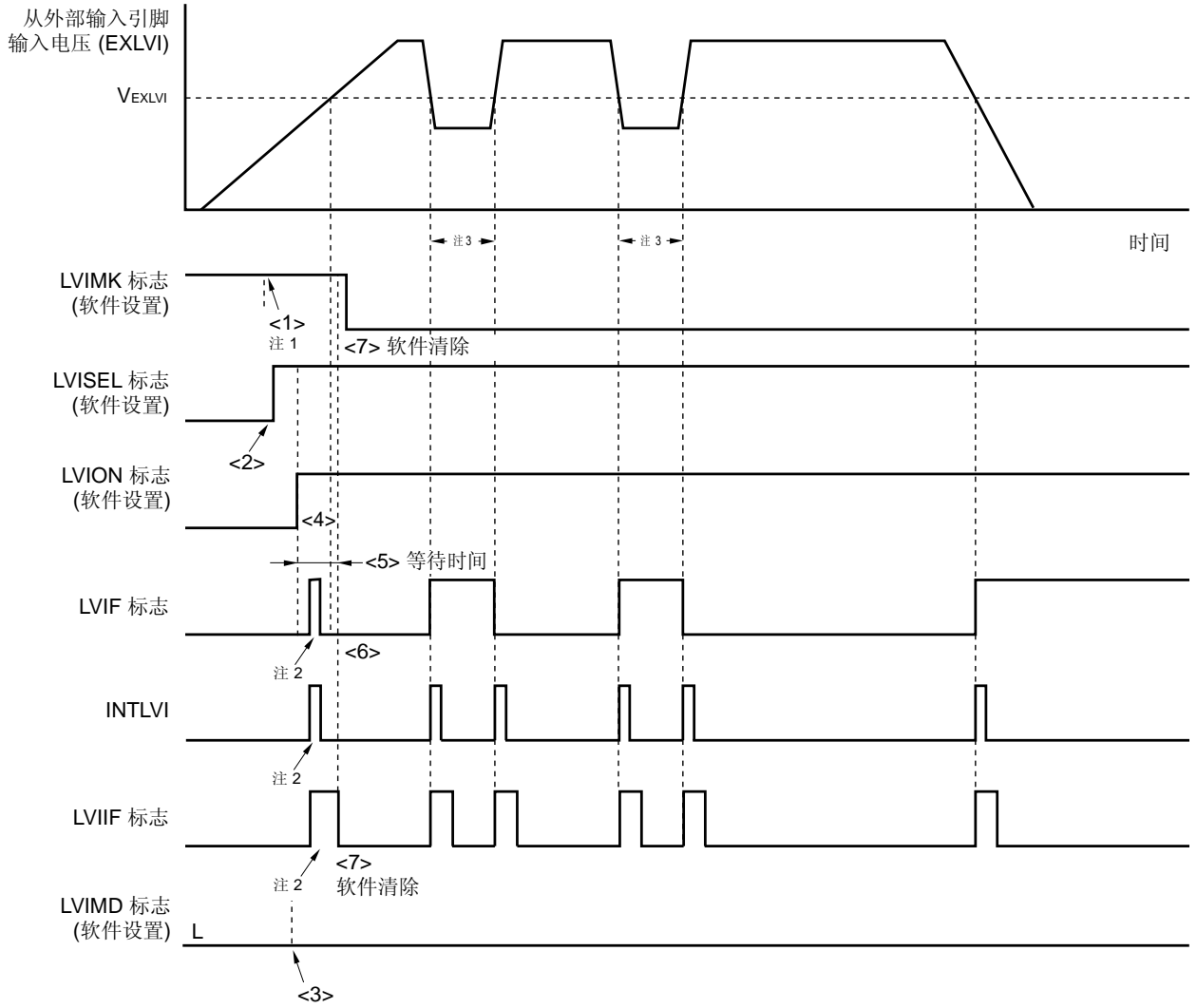
- 当启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <2> 设置低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 为 1 (检测从外部输入引脚 (EXLVI) 输入的电压的电平)
 - <3> 清除 LVIM 的第 1 位 (LVIMD) 为 0 (当电平被检测时, 产生中断信号) (默认值)。
 - <4> 设置 LVIM 的第 7 位 (LVION) 为 1 (允许 LVI 操作)
 - <5> 通过软件等待操作稳定时间 (10 μ s (最小))。
 - <6> 当检测 EXLVI 的下降沿时, 确认“从外部输入引脚 (EXLVI) 输入的电压 \geq 检测电压 ($V_{EXLVI} = 1.21\text{ V}$ (典型))”或者当检测 EXLVI 的上升沿时, 确认“从外部输入引脚 (EXLVI) 输入的电压 (EXLVI) $<$ 检测电压 ($V_{EXLVI} = 1.21\text{ V}$ (典型))”。
 - <7> 清除 LVI 的中断请求标志 (LVIIF) 为 0。
 - <8> 释放 LVI 的中断屏蔽标志 (LVIMK)。
 - <9> 执行 EI 指令 (当使用向量中断时)。

图 20-8 表示中断信号被低电压检测电路产生的时序。这个时序图中的号码对应上面的<1>到<8>。

注意事项 从外部输入引脚 (EXLVI) 输入的电压必须满足 $EXLVI < V_{DD}$ 。

- 当停止操作时
 - 必须执行下面任意一个过程。
 - 当使用 8 位存储器操作指令时:
 - 写入 00H 到 LVIM。
 - 当使用 1 位存储器操作指令时:
 - 清除 LVION 为 0。

图 20-8. 低压检测器的中断信号产生时序
(检测从外部输入引脚输入电压的电平 (EXLVI))



- 注
1. LVIMK 标志被复位信号设置为“1”。
 2. 中断请求标志寄存器的 LVIIF 标志和 LVIF 标志可能置位 (1)。
 3. 如果在低于 LVI 检测电压状态 LVION 被清零, 将产生 INTLVI 信号且 LVIIF 变为 1。

备注 上面图 20-8 中的 <1>到<8>对应于 20.4.2 (2) 当检测从外部输入引脚的输入电压的电平 (EXLVI) 时的“当启动操作时”中描述的<1>到<8>。

20.5 低电压检测电路的注意事项

20.5.1 当供电电压在LVI检测电压附近波动时

在电源电压 (V_{DD}) 按一定的周期在 LVI 检测电压 (V_{LVI}) 附近波动时, 根据低电压检测电路的使用方法, 操作如下。

(1) 当用作复位时

系统可能被重复复位并从复位状态被释放。

此时, 从复位释放到微控制器操作开始的时间可以通过下面措施 (1) 任意设置。

(2) 当用作中断时

可能频繁产生中断请求。采取下面措施 (2) 的 (b)。

<措施>

(1) 当用作复位时

在释放复位信号后, 通过使用定时器的软件计数器的方法等待每个系统的电源电压波动周期, 然后初始化端口。

(2) 当用作中断时

(a) 在 LVI 中断的服务程序中, 通过使用低电压检测寄存器 (LVIM) 的第 0 位 (LVIF), 当检测 V_{DD} 的下降沿时, 确认“电源电压 (V_{DD}) \geq 检测电压 (V_{LVI})”, 或者当检测 V_{DD} 的上升沿时, 确认“电源电压 (V_{DD}) $<$ 检测电压 (V_{LVI})”。清除中断请求标志寄存器 0L (IFOL) 的第 0 位 (LVIIF) 为 0。

(b) 对于电源电压在 LVI 检测电压附近波动周期长的系统中, 等待电源电压波动周期经历过后, 执行上述 (a) 中提到的过程。

备注 如果低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 被设置为“1”, 上面词语的意义更改如下。

- 电源电压 (V_{DD}) → 从外部输入引脚 (EXLVI) 输入的电压
- 检测电压 (V_{LVI}) → 检测电压 ($V_{EXLVI} = 1.21 V$)

20.5.2 使用LVI作为复位信号时的注意事项

当使用 LVI 用作复位信号 (LVIMD = 1) 时, 根据外围硬件的设置情况, LVI 的使用方法和设置过程将受到限制。根据外围硬件的设置情况, 按照如下所示方法设置 LVI。

如果执行了这些设置以外的设置, LVI 可能不会正常操作。

(1) 当不使用 16 位定时器/事件计数器 00 时, 或者当设置捕捉/比较控制寄存器 00 (CRC00) 为 “00H (复位值)” 时使用 16 位定时器/事件计数器 00 时。

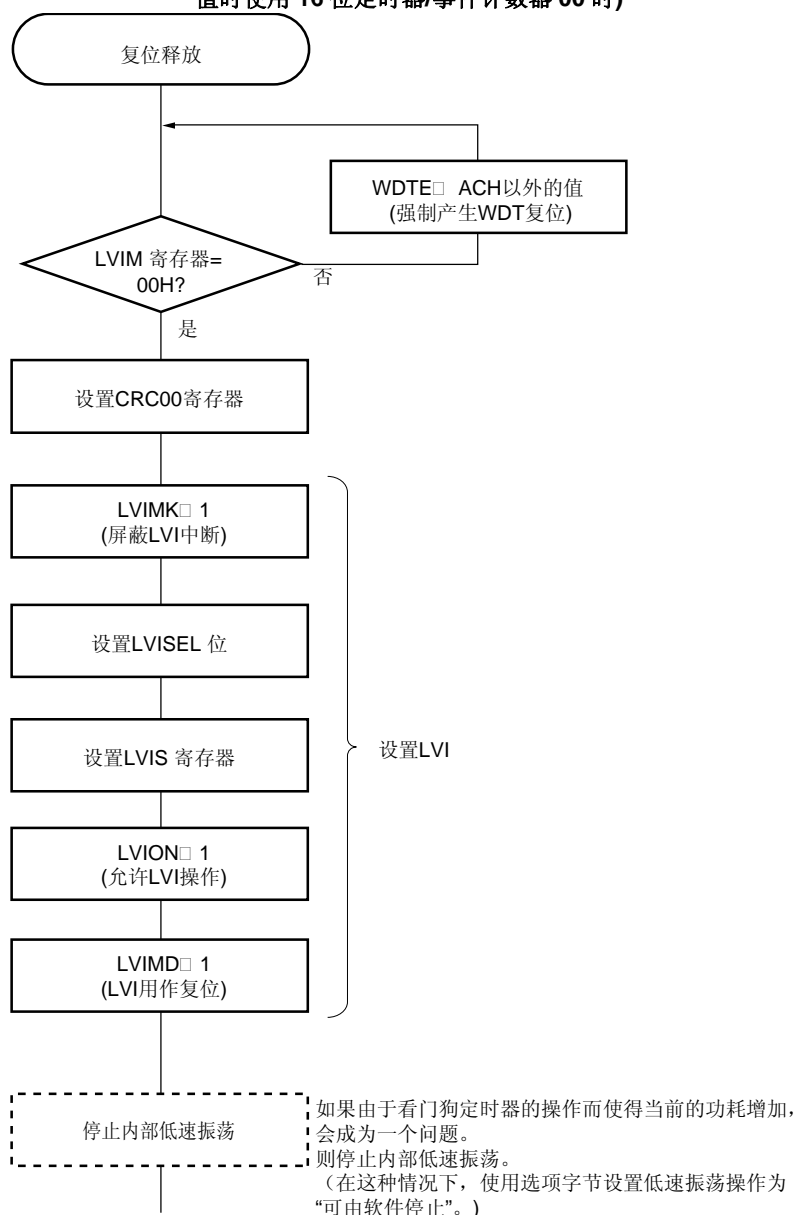
→ 不要写 CRC00 寄存器。

(2) 当设置捕捉/比较控制寄存器 00 (CRC00) 为 “00H (复位值)” 以外的数值时使用 16 位定时器/事件计数器 00 时。

→ 确保复位释放后使用看门狗定时器产生一个复位信号, 并且在 LVI 操作已经停止后设置 CRC00 寄存器和 LVI。(参见如下设置过程的流程图)

设置 LVI 后, 不要设置 LVI 和 CRC00 寄存器。

图 20-9. CRC00 寄存器和 LVI 的设置过程 (当设置捕捉/比较控制寄存器 00 (CRC00) 为 “00H (复位值)” 以外的数值时使用 16 位定时器/事件计数器 00 时)



第二十一章 选项字节

21.1 选项字节的功能

μ PD78F8024 和 78F8025 的 flash 存储器地址 0080H 到 0084H 是选项字节区域。在启动电源或从复位状态重新启动时，自动引用选项字节，并且设置指定的功能。当使用产品时，确认通过使用选项字节设置以下功能。

要在自编程过程中使用启动交换功能，0080H 到 0084H 被 1080H 到 1084H 替换。因此，预先将 0080H 到 0084H 和 1080H 到 1084H 设置为同样的值。

注意事项 确保 0082H, 0083H 和 0084H 设置为 00H（当使用启动交换操作时，设置 0082H / 1082H, 0083H / 1083H 和 0084H / 1084H）。

(1) 0080H / 1080H

- 内部低速振荡器操作
 - 可以由软件停止。
 - 无法停止。
- 看门狗定时器溢出时间设置
- 看门狗定时器计数器操作
 - 允许计数器操作
 - 禁止计数器操作
- 看门狗定时器窗打开周期设置

注意事项 在启动交换操作过程中，将 1080H 设置为与 0080H 相同的值，因为 0080H 与 1080H 交换。

(2) 0081H / 1081H

- 选择 POC 模式
 - 在 2.7V/1.59V POC 模式操作期间（POCMODE = 1）
设备上电时处于复位状态并且直到电源电压达到 2.7V（典型）。当电压超过 2.7V（典型）时，从复位状态释放。之后，不会在 2.7V 时检测到 POC，而是在 1.59V（典型）被检测到。
如果上电之后电源电压以小于 0.5V/ms（最小）速度提升到 1.8V，建议使用 2.7V/1.59V POC 模式。
 - 在 1.59V POC 模式操作期间（POCMODE = 0）
设备上电时处于复位状态并且直到电源电压达到 1.59V（典型）。当电压超过 1.59V（典型）时，从复位状态释放。之后，在 1.59V（典型）时检测到 POC，与上电过程相同。

注意事项 POCMODE 只能使用专用 flash 存储器编程器写入。它不能在自编程或者自编程期间的启动交换操作期间设置。但是，因为在启动交换操作中，0081H 和 1081H 的值互换，因此为 1081H 和 0081H 设置相同的值。

21.2 字节选项的格式

字节选项的格式表示如下。

图 21-1. 字节选项的格式 (1 / 2)

地址: 0080H / 1080H^注

7	6	5	4	3	2	1	0
0	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	LSROSC
WINDOW1	WINDOW0	看门狗定时器窗口打开周期					
0	0	25%					
0	1	50%					
1	0	75%					
1	1	100%					
WDTON	看门狗定时器计数器控制 / 非法访问检测操作						
0	禁止计数器操作 (复位后计数停止), 禁止非法访问检测操作						
1	允许计数器操作 (复位后计数开始), 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	$2^{10}/f_{RL}$ (3.88 ms)				
0	0	1	$2^{11}/f_{RL}$ (7.76 ms)				
0	1	0	$2^{12}/f_{RL}$ (15.52 ms)				
0	1	1	$2^{13}/f_{RL}$ (31.03 ms)				
1	0	0	$2^{14}/f_{RL}$ (62.06 ms)				
1	0	1	$2^{15}/f_{RL}$ (124.12 ms)				
1	1	0	$2^{16}/f_{RL}$ (248.24 ms)				
1	1	1	$2^{17}/f_{RL}$ (496.48 ms)				
LSROSC	内部低速振荡器操作						
0	可以由软件停止 (当向 RCM 寄存器的第 1 位 (LSRSTOP) 写 1 时停止)						
1	无法停止 (即使向 LSRSTOP 写 1 也无法停止)						

注 在启动交换操作过程中, 将 1080H 设置为与 0080H 相同的值, 因为 0080H 与 1080H 交换。

注意事项 1. 禁止同时设置 $WDCS2 = WDCS1 = WDCS0 = 0$ 和 $WINDOW1 = WINDOW0 = 0$ 。

2. 当在 $1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$ 时使用看门狗定时器时, 禁止设置 $WINDOW1 = WINDOW0 = 0$ 。

3. 看门狗定时器在自编程和 flash 存储器的 EEPROM 仿真期间继续操作。在处理期间, 中断响应时间延迟。设置溢出时间和窗口大小时应考虑到这种延时。

4. 如果 $LSROSC = 0$ (振荡器可以由软件停止), 在 HALT 和 STOP 模式下, 无论内部振荡控制器模式寄存器 (RCM) 的第 1 位 (LSRSTOP) 的设置如何, 都不能向看门狗定时器提供计数器时钟。当 8 位定时器 H1 工作于内部低速振荡器时钟, 即使处于 HALT / STOP 模式计数器时钟也会提供给 8 位定时器 H1。

5. 确保清除第 7 位。

备注 1. f_{RL} : 内部低速振荡器时钟频率

2. (): $f_{RL} = 264\text{ kHz}$ (最大)

图 21-1. 选项字节的格式 (2 / 2)

地址: 0081H / 1081^{注1,2}

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

POCMODE	POC 模式选择
0	1.59 V POC 模式
1	2.7 V/1.59 V POC 模式

- 注**
1. POCMODE 只能使用专用 flash 存储器编程器写入。它不能在自编程或者自编程的启动交换操作期间设置。然而，因为在启动交换操作中，0081H 和 1081H 的值互换，因此为 1081H 和 0081H 设置相同的值。
 2. 为改变 POC 模式设置，flash 存储器批擦除（片擦除）之后再次设置 0081H 的值。在指定块存储器擦除之后无法改变设置。

注意事项 确保清零第 7 位。

地址: 0082H / 1082H, 0083H / 1083H, 0084H / 1084H^注

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

- 注** 确保设置 0082H, 0083H 和 0084H 为 00H, 因为这些地址是保留区域。同时也要设置 1082H, 1083H 和 1084H 为 00H 因为当使用启动交换操作时 0082H, 0083H 和 0084H 被替换为 1082H, 1083H 和 1084H。

下面是软件设置选项字节的例子描述。

```

OPT    CSEG  AT 0080H
OPTION: DB    30H    ; 允许看门狗定时器操作（非法访问检测操作），
                   ; 看门狗定时器窗口打开周期：50%，
                   ; 看门狗定时器溢出时间：210/fRL，
                   ; 内部低速振荡器可以由软件停止。
                   DB    00H    ; 1.59 V POC 模式
                   DB    00H    ; 保留区域
                   DB    00H    ; 保留区域
                   DB    00H    ; 保留区域

```

备注 参考复位过程中的选项字节设置。关于复位过程时序，参见 第十八章 复位功能。

第二十二章 FLASH存储器

μPD78F8024 和 78F8025 包含 flash 存储器，当它安装在电路板上时，可以写入、擦除和覆盖程序。

22.1 内部存储器容量切换寄存器

内部存储器容量可以由内部存储器容量切换寄存器（IMS）选择。

IMS 由 8 位存储器操作指令设置。

复位信号将 IMS 设为 CFH。

注意事项 确保复位释放后每个产品的设置值如表 22-1 所示。

图 22-1. 内部存储器容量切换寄存器（IMS）的格式

地址: FFF0H 复位后: CFH R / W

标志	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	1	0	512 字节				
	1	1	0	1024 字节				
	除以上情况			禁止设置				
	ROM3	ROM2	ROM1	ROM0	内部 ROM 容量选择			
	0	0	1	0	8 KB			
	1	0	0	0	32 KB			
	除以上情况				禁止设置			

表 22-1. 内部存储器容量切换寄存器设置

产品	IMS 设置
μPD78F8024	42H
μPD78F8025	C8H

22.2 用Flash存储器编程器写入

通过使用专用的 flash 存储器编程器，可以在板或脱板写入 flash 存储器数据。

(1) 板上编程

在 μ PD78F8024 或者 78F8025 安装到目标系统中后，可以重新写入 flash 存储器的内容。必须目标系统中安装连接专用 flash 存储器编程器的连接器。

(2) 板外编程

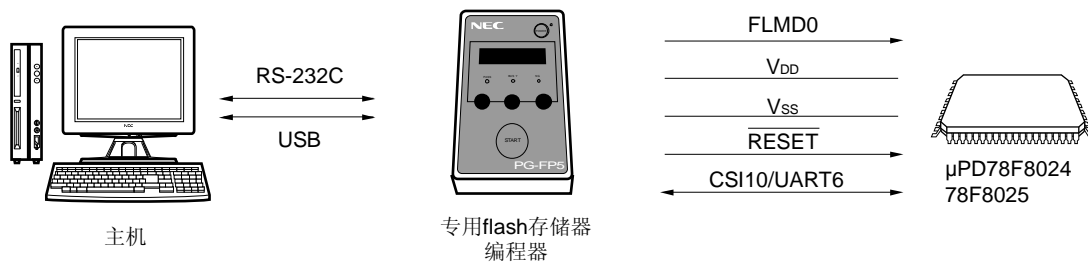
在 μ PD78F8024 或者 78F8025 安装到目标系统中前，通过使用专用的程序适配器（FA 系列），将数据写入 flash 存储器。

备注 FA 系列是 Naito Densai Machida Mfg. Co., Ltd 的产品。

22.3 编程环境

写入程序到 μ PD78F8024 和 78F8025 的 flash 存储器所需要的环境如下所示。

图 22-2. 写入程序到 Flash 存储器的环境



需要一个控制专用 flash 存储器编程器的主机。

CSI10 或者 UART6 用于专用 flash 存储器编程器和 μ PD78F8024 或者 78F8025 之间的接口，用作写入、擦除等操作。要在板外写入 flash 存储器，必需使用专用的程序适配器（FA 系列）。

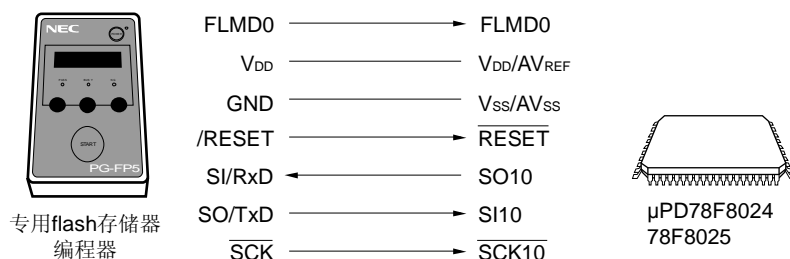
22.4 通信模式

建立专用 flash 存储器编程器和 μ PD78F8024 或者 78F8025 之间通信，可以使用 CSI10 或 UART6 串行通信接口。

(1) CSI10

传输速率：2.4 kHz 到 2.5 MHz

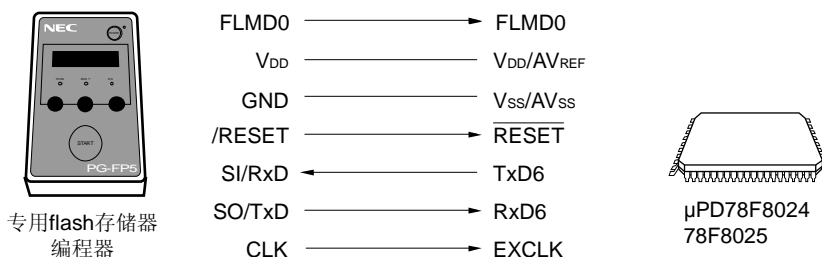
图 22-3. 与专用 Flash 存储器编程器的通信 (CSI10)



(2) UART6

传输速率：115200 bps

图 22-4. 与专用 Flash 存储器编程器的通信 (UART6)



专用 flash 存储器编程器为 μPD78F8024 或者 78F8025 产生如下信号。详细信息参考 PG-FP5、FL-PR5、PG-FP4 或 FL-PR4 的用户手册。

表 22-2. 引脚连接

专用 Flash 存储器编程器			μPD78F8024, 78F8025	连接	
信号名称	输入 / 输出	引脚功能	引脚名称	CSI10	UART6
FLMD0	输出	模式信号	FLMD0	○	○
V _{DD}	输入 / 输出	V _{DD} 电压产生 / 电源监视	V _{DD} , AV _{REF}	○	○
GND	-	地	V _{SS} , AV _{SS}	○	○
CLK	输出	时钟输出给 μPD78F8024 或者 78F8025	EXCLK / X2 / P122	× ^註	○
/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	发送时钟	SCK10	○	×

注 当使用 CSI10 时，只能使用内部高速振荡器时钟 (f_{RH})。

- 备注
- ： 确保连接这个引脚。
 - ： 如果信号由目标板上产生，则无需连接这个引脚。
 - ×： 无需连接这个引脚。

当使用专用 flash 存储器编程器（FA 系列）时，对于未使用的引脚，采用表 2-2 引脚输入 / 输出电路类型所列出的未使用引脚的推荐连接，或者表 22-3 当 Flash 存储器编程适配器连接时的未使用引脚处理（要求）中描述的连接。

表 22-3 当 Flash 存储器编程适配器连接时的未使用引脚处理（要求）

引脚名称	引脚处理
P00, P01	通过一个电阻连接到 $V_{SS}^{\#1}$
P10, P11	通过一个电阻连接到 $V_{SS}^{\#2}$
P14	通过一个电阻连接到 $V_{SS}^{\#3}$
P16, P17	通过一个电阻连接到 $V_{SS}^{\#1}$
P30 到 P33	
P60, P61	通过一个电阻连接到 V_{SS} 或者直接连接到 V_{SS} .
P120	通过一个电阻连接到 $V_{SS}^{\#1}$

- 注
1. 在设计中，这些引脚可能未使用电阻直接连接到 V_{SS} ，因此在 Flash 存储器编程期间，适配器板上 flash 存储器不能切换到正常操作模式。
 2. 当通过 CSI10 与专用 flash 存储器编程器串行通信时，将这些引脚连接到编程器。
 3. 当通过 UART6 与专用 flash 存储器编程器串行通信时，将这些引脚连接到编程器。

22.5 板上引脚的操作

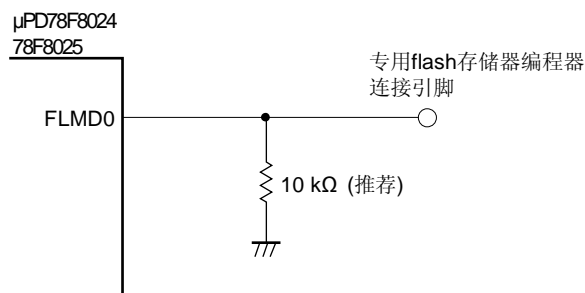
要在板上写入 flash 存储器，在目标系统中必须提供连接专用 flash 存储器编程器的连接器。首先提供选择正常工作模式或者板上 flash 存储器编程模式的功能。

当设置为 flash 存储器编程模式时，所有在编程 flash 存储器中没有使用的引脚与刚刚复位后的状态相同。因此，如果外部设备在复位后没有立即识别引脚状态，引脚必须按照下面的描述来处理。

22.5.1 FLMD0 引脚

正常操作模式下，输入 0V 到 FLMD0 引脚上。在 flash 存储器编程模式下， V_{DD} 写入电压提供给 FLMD0 引脚。如下是 FLMD0 引脚连接的一个示例。

图 22-5. FLMD0 引脚连接示例



22.5.2 串行接口引脚

用于串行接口的引脚如下所示。

表 22-4. 用于串行接口的引脚

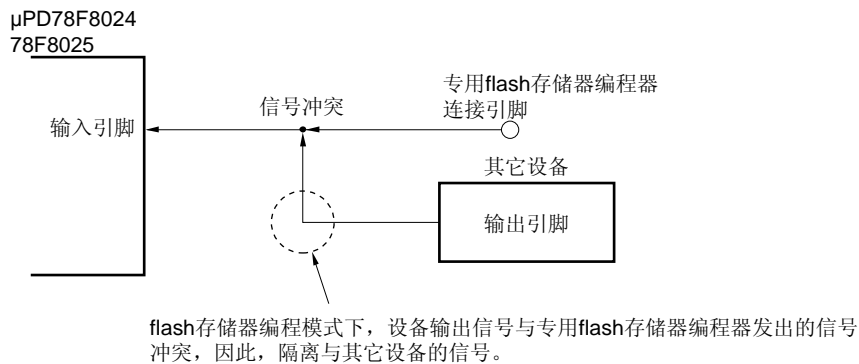
串行接口	使用的引脚
CSI10	SO10, SI10, SCK10
UART6	TxD6, RxD6

为了将专用 flash 存储器编程器连接到连接板上其它设备的串行接口引脚，小心不要产生信号冲突或者引起其它的连接设备故障。

(1) 信号冲突

如果专用 flash 存储器编程器（输出）连接到连接其它设备（输出）的串行接口引脚（输入），会发生信号冲突。为了避免这种冲突，或者隔离与其它设备的连接，或者使其它设备处于输出高阻状态。

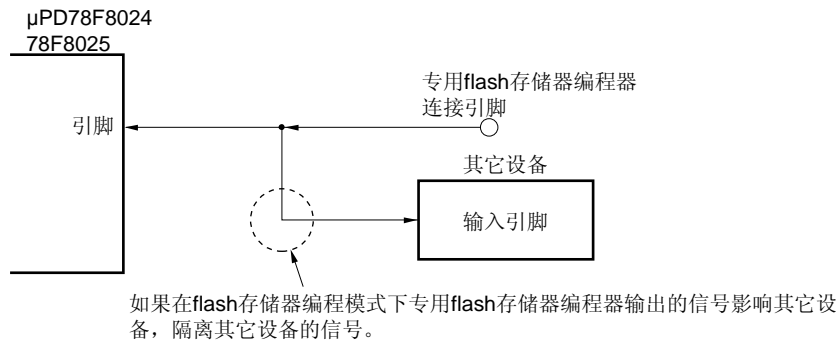
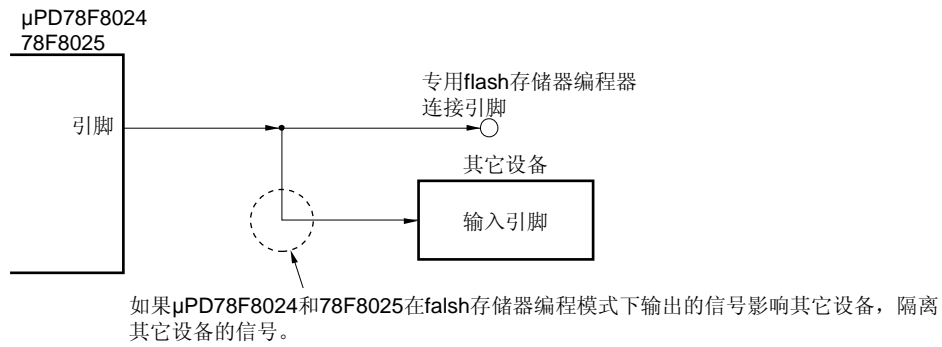
图 22-6. 信号冲突（串行接口的输入引脚）



(2) 其它设备的故障

如果专用 flash 存储器编程器（输出或输入）连接到连接其它设备（输入）的串行接口引脚（输入或输出），信号可能输出到其它设备，引起设备的故障。为了避免这种故障，隔绝与其它设备的连接。

图 22-7.其它设备的故障

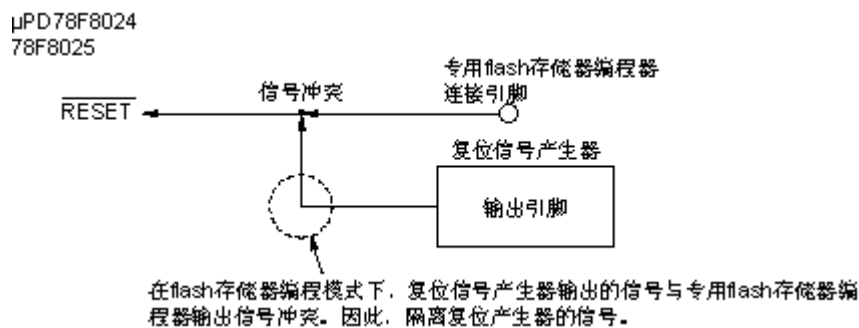


22.5.3 RESET引脚

如果专用 flash 存储器编程器的复位信号连接到板上复位信号产生器的 $\overline{\text{RESET}}$ 引脚，发生信号冲突。要防止这个冲突，隔离与复位信号产生器的连接。

当设置 flash 存储器编程模式时，如果从用户系统输入复位信号，将无法对 flash 存储器正确编程。不要输入专用 flash 存储器编程器的复位信号以外的信号。

图 22-8. 信号冲突 ($\overline{\text{RESET}}$ 引脚)



22.5.4 端口引脚

当设置 flash 存储器编程模式时，所有在编程 flash 存储器中没有使用的引脚与刚刚复位后的状态相同。如果连接到端口的设备在复位后没有立即识别端口状态，端口引脚必须通过一个电阻连接到 V_{DD} 或 V_{SS}。

22.5.5 REGC 引脚

与正常工作时的方式相同，通过一个电容（0.47 到 1 μ F：建议）连接 REGC 引脚到 GND。

22.5.6 其它信号引脚

当使用板上时钟时，按照正常工作的模式中的状态连接 X1 和 X2。

要从专用 flash 存储器编程器输入操作时钟，将编程器的 CLK 连接到 EXCLK / X2 / P122。

- 注意事项**
1. 当使用 CSI10 时，只能使用内部高速振荡器时钟（f_{RH}）
 2. 当使用 UART6 时，只能使用 X1 时钟（f_X）或外部系统主时钟（f_{EXCLK}）。

22.5.7 电源

要使用 flash 存储器编程器的电源输出，连接 V_{DD} 引脚到 flash 存储器编程器的 V_{DD}，并且连接 V_{SS} 引脚到 flash 存储器编程器的 GND。

要使用板上电源，按照正常工作模式连接。

即使使用板上电源时，确保连接 V_{DD} 和 V_{SS} 引脚到 flash 存储器编程器的 V_{DD} 和 GND 来使用 flash 存储器编程器使用的电源监视功能。

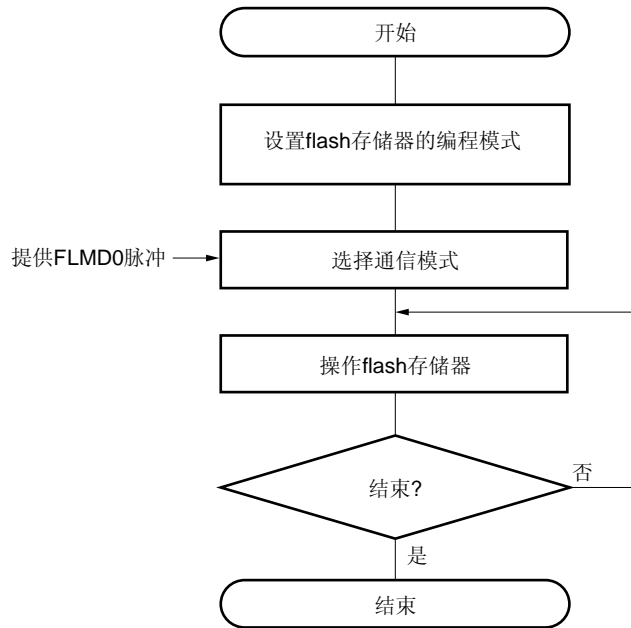
按照正常工作模式提供其它电源（AV_{REF} 和 AV_{SS}）。

22.6 编程方法

22.6.1 控制flash存储器

下面的图表示操作 flash 存储器的过程。

图 22-9. Flash 存储器操作过程



22.6.2 Flash存储器编程模式

要使用专用 flash 存储器编程器来重新写入 flash 存储器的内容，设置 μ PD78F8024 和 78F8025 为 flash 存储器编程模式。要设置这个模式，设置 FLMD0 引脚为 V_{DD} ，并且清除复位信号。

当在板上写入 flash 存储器时，通过使用跳线来更改模式。

图 22-10. Flash 存储器编程模式

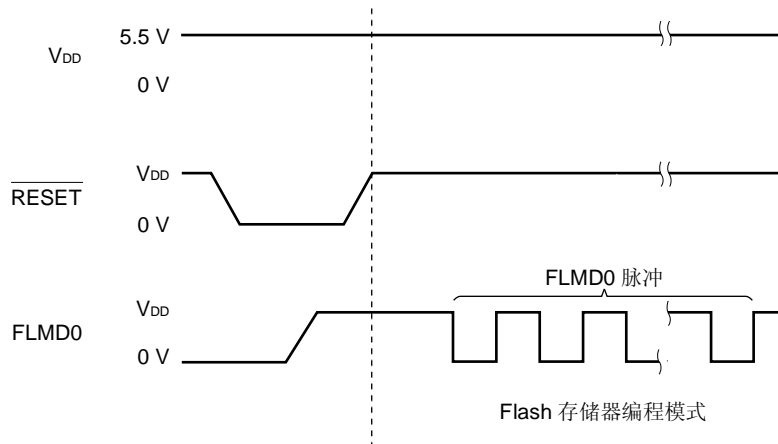


表 22-5. FLMD0 引脚和复位后工作模式之间的关系

FLMD0	工作模式
0	正常工作模式
V _{DD}	Flash 存储器编程模式

22.6.3 选择通信模式

在 μ PD78F8024 和 78F8025 中，通信模式是由进入专用 flash 存储器编程模式之后由 FLMD0 引脚上输入的脉冲选择的。专用 flash 存储器编程产生这些 FLMD0 脉冲。

下表表明了脉冲数与通信模式之间的关系。

表 22-6. 通信模式

通信模式	标准设置 ^{注1}				使用的引脚	外围时钟	FLMD0 的脉冲数
	端口	速度	频率	多速率			
UART (UART6)	UART-Ext-Osc	115200 bps ^{注3}	2 至 20 MHz ^{注2}	1.0	TxD6, RxD6	f _X	0
	UART-Ext-FPnCK (n = 4: PG-FP4, FL-PR4, n = 5: PG-FP5, FL-PR5)					f _{EXCLK}	3
3 线串行输入 / 输出 (CSI10)	CSI-内部-OSC	2.4 kHz 到 2.5 MHz	–		SO10, SI10, SCK10	f _{RH}	8

- 注
1. 在 flash 存储器编程器的 GUI 上的标准设置的选择项目。
 2. 可能的设置范围因电压不同而不同。详情参考电气特性一章。
 3. 因为波特率误差以外因素，例如信号波形回转，也会影响 UART 通信，要彻底评估回转和波特率误差。

注意事项 当选取 UART6 时，在收到 FLMD0 脉冲之后从专用 flash 存储器编程器发送的复位命令基础上，计算接收时钟。

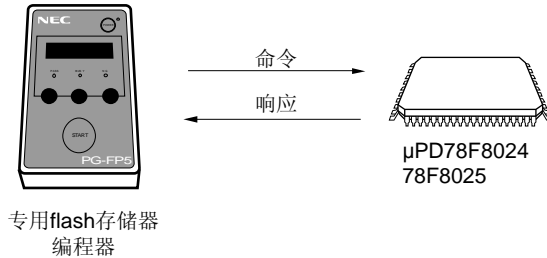
备注

f_X: X1 时钟
f_{EXCLK}: 外部主系统时钟
f_{RH}: 内部高速振荡器时钟

22.6.4 通信命令

μ PD78F8024 和 78F8025 通过使用命令与专用 flash 存储器编程器通信。从专用 flash 存储器编程器发送到 μ PD78F8024 或者 78F8025 的信号叫做命令，并且从 μ PD78F8024 或者 78F8025 发送到专用 flash 存储器编程器的信号叫做响应。

图 22-11. 通信命令



在下面的表中列出 μ PD78F8024 和 78F8025 的 flash 存储器控制命令。所有的命令从编程器发出，并且 μ PD78F8024 或者 78F8025 执行对应各自命令的处理。

表 22-7. Flash 存储器控制命令

分类	命令名	功能
校验	Verify	比较 flash 存储器的指定区域的内容和从编程器发送的数据。
擦除	Chip Erase	擦除整个 flash 存储器
	Block Erase	擦除 flash 存储器中的指定区域。
空白检查	Block Blank Check	检查 flash 存储器的指定块是否被正确擦除。
写入	Programming	写入数据到 flash 存储器的指定区域。
获取信息	Status	获取当前操作状态（状态数据）。
	Silicon Signature	获取 μ PD78F8024 或者 78F8025 信息（例如产品编号和 flash 存储器配置）。
	Version Get	获取 μ PD78F8024 或者 78F8025 的版本和固件版本。
	Checksum	获取指定区域的校验和数据。
安全	Security Set	设置安全信息。
其它	Reset	用作检测通信的同步状态。
	Oscillating Frequency Set	设置振荡器频率。

μ PD78F8024 或者 78F8025 对专用 flash 存储器编程器发出的命令返回一个响应信号。下面列出 μ PD78F8024 或者 78F8025 发出的响应名称。

表 22-8. 响应名

响应信号名称	功能
ACK	响应命令 / 数据。
NAK	响应非法命令 / 数据。

22.7 安全设置

μ PD78F8024 和 78F8025 支持禁止重新写入内部 flash 存储器的用户程序的安全功能，所以程序不能被未经授权的人更改。

下面表示的操作可以使用安全设置命令来执行。当编程模式被设置时，安全设置有效。

- 禁止批擦除（芯片擦除）

在板 / 脱板编程过程中，对 flash 存储器的块擦除和对整个块的批擦除（芯片擦除）命令的执行被该禁止设置。一旦禁止批擦除（芯片擦除）命令的执行，所有禁止设置（包含批擦除（芯片擦除）的禁止）不能再次取消。

注意事项 在设置批擦除的安全设置后，不能执行擦除。此外，即使执行写入命令，也不能写入与已经写入 flash 存储器的数据不同的数据，因为擦除命令无效。

- 禁止块擦除

在板 / 脱板编程过程中，执行禁止对 flash 存储器中的指定块的块擦除命令。然而，通过自编程的方式，可以擦除块。

- 禁止写入

在板 / 脱板编程过程中，执行禁止对 flash 存储器中整个块的写入和块擦除命令。然而，通过自编程的方式，可以写入块。

- 禁止重写启动区域 0

对 flash 存储器中的启动区域 0（0000H 到 0FFFH）的批擦除（芯片擦除）命令、块擦除命令和写入命令的执行被这个设置禁止。

注意事项 如果对启动区域 0 应用了安全设置，对于设备，将不能执行启动区域 0 的重写和批擦除（芯片擦除）。

当 flash 存储器出厂时，批擦除（芯片擦除）命令、块擦除命令、写入命令和重新写入启动区域 0 的默认设置为使能。可以通过在板 / 脱板编程和自编程设置安全。可以组合使用各种安全设置。

所有安全设置被批擦除（芯片擦除）命令清除。

当 μ PD78F8024 和 78F8025 的安全功能使能时擦除和写入命令之间的关系如表 22-9 所示。

表 22-9. 允许安全设置功能和命令之间的关系

(1) 在板 / 脱板编程过程中

有效的安全	执行的命令		
	批擦除 (芯片擦除)	块擦除	写入
批擦除 (芯片擦除) 的禁止	不能批擦除	不能擦除块。	可以执行 ^注 。
块擦除的禁止	可以批擦除。		可以执行。
写入的禁止			不能执行。
重新写入启动区域 0 的禁止	不能批擦除	不能擦除启动区域 0。	不能写入启动区域 0。

注 确保没有写入数据到写入区域。因为在禁止批擦除 (芯片擦除) 后不能擦除数据, 如果数据没有擦除, 不要写入数据。

(2) 在自编程过程中

有效的安全	执行的命令	
	块擦除	写入
批擦除 (芯片擦除) 的禁止	可以擦除块。	可以执行。
块擦除的禁止		
写入的禁止		
重新写入启动区域 0 的禁止	不能擦除启动区域 0。	不能写入启动区域 0。

表 22-10 表明如何在各种编程模式下进行安全设置。

表 22-10. 在各种编程模式下的安全设置

(1) 在板 / 脱板编程

安全	安全设置	如何禁止安全设置
批擦除 (芯片擦除) 的禁止	通过专用 flash 存储器编程器的 GUI 来设置, 等等。	设置后不能禁止。
块擦除的禁止		执行批擦除 (芯片擦除) 命令
写入的禁止		
重新写入启动区域 0 的禁止		设置后不能禁止。

(2) 自编程

安全	安全设置	如何禁止安全设置
批擦除 (芯片擦除) 的禁止	通过使用信息库来设置。	设置后不能禁止。
块擦除的禁止		在在板 / 脱板编程过程中执行批擦除 (芯片擦除) 命令 (在自编程过程中, 不能被使禁止)
写入的禁止		
重新写入启动区域 0 的禁止		设置后不能禁止。

22.8 Flash存储器的自编程

μ PD78F8024 和 78F8025 支持通过用户程序将程序写入 flash 存储器的自编程功能。因为这个功能允许用户程序通过使用自编程库来重新写入 flash 存储器，可以用它来现场更新程序。

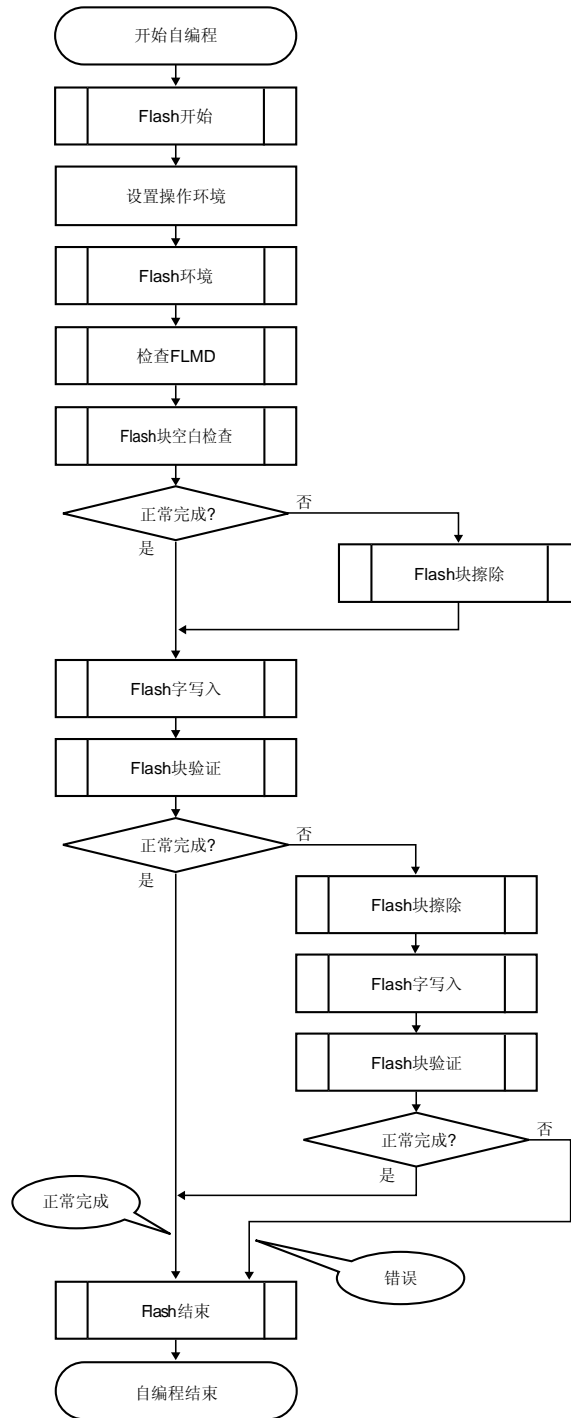
如果在自编程过程中一个中断发生，自编程可以暂时停止，并且中断服务程序被执行。为了执行中断服务程序，自编程模式终止后恢复到正常模式，并且执行 EI 指令。在自编程模式稍后恢复后，自编程可以继续。

备注 关于自编程功能和自编程库的详细信息，参阅 **78K0 微控制器自编程库 Type01 用户手册 (U18274E)**。

- 注意事项**
1. 内部高速振荡器在自编程期间开始振荡，与 RSTOP 标志（内部振荡器模式寄存器（RCM）的第 0 位）的设置无关。即使执行 STOP 指令也不能终止内部高速振荡器的振荡。
 2. 在自编程期间在 FLMD0 引脚上输入高电平。
 3. 确保在自编程之前执行 DI 指令。
自编程功能检查中断请求标志（IF0L, IF0H, IF1L 和 IF1H）。如果产生中断请求，自编程终止。
 4. 即使在 DI 状态自编程仍然可以没有屏蔽的中断请求终止。为避免这种情况，通过中断屏蔽标志寄存器（MK0L, MK0H, MK1L 和 MK1H）来屏蔽中断。

下图表示通过使用自编程库重新写入 flash 存储器的流程。

图 22-12. 自编程的流程（重新写入 Flash 存储器）



备注 关于自编程库的详细信息，参阅 **78K0 微控制器自编程库 Type01 用户手册 (U18274E)**。

下表表明自编程库的处理时间和中断响应时间。

表 22-11. 自编程库的处理时间 (1 / 3)

(1) 当使用内部高速振荡器时钟并且 RAM 入口不在短直接寻址范围内

库名称		处理时间 (μs)			
		C 编译器正常模型		C 编译器 / 汇编器的静态模型	
		最小	最大	最小	最大
自编程开始库		4.0	4.5	4.0	4.5
初始化库		1105.9	1106.6	1105.9	1106.6
模式检查库		905.7	906.1	904.9	905.3
块空白检查库		12776.1	12778.3	12770.9	12772.6
块擦除库		26050.4	349971.3	26045.3	349965.6
字写入库		$1180.1 + 203 \times w$	$1184.3 + 2241 \times w$	$1172.9 + 203 \times w$	$1176.3 + 2241 \times w$
块验证库		25337.9	25340.2	25332.8	25334.5
自编程结束库		4.0	4.5	4.0	4.5
获取信息库	选项值: 03H	1072.9	1075.2	1067.5	1069.1
	选项值: 04H	1060.2	1062.6	1054.8	1056.6
	选项值: 05H	1023.8	1028.2	1018.3	1022.1
设置信息库		70265.9	759995.0	70264.9	759994.0
EEPROM 写入库		$1316.8 + 347 \times w$	$1320.9 + 2385 \times w$	$1309.0 + 347 \times w$	$1312.4 + 2385 \times w$

(2) 当使用内部高速振荡器时钟并且 RAM 入口在短直接寻址范围内

库名称		处理时间 (μs)			
		C 编译器正常模型		C 编译器 / 汇编器的静态模型	
		最小	最大	最小	最大
自编程开始库		4.0	4.5	4.0	4.5
初始化库		449.5	450.2	449.5	450.2
模式检查库		249.3	249.7	248.6	248.9
块空白检查库		12119.7	12121.9	12114.6	12116.3
块擦除库		25344.7	349266.4	25339.6	349260.8
字写入库		$445.8 + 203 \times w$	$449.9 + 2241 \times w$	$438.5 + 203 \times w$	$441.9 + 2241 \times w$
块验证库		24682.7	24684.9	24677.6	24679.3
自编程结束库		4.0	4.5	4.0	4.5
获取信息库	选项值: 03H	417.6	419.8	412.1	413.8
	选项值: 04H	405.0	407.4	399.5	401.3
	选项值: 05H	367.4	371.8	361.9	365.8
设置信息库		69569.3	759297.3	69568.3	759296.2
EEPROM 写入库		$795.1 + 347 \times w$	$799.3 + 2385 \times w$	$787.4 + 347 \times w$	$790.8 + 2385 \times w$

- 备注**
1. 以上处理时间是针对写入起始地址位于内部高速 RAM 且处于内部高速振荡器稳定操作期间 (RSTS=1)。
 2. RSTS: 内部振荡器模式寄存器 (RCM) 的第 7 位
 3. w: 写入数据的字个数 (1 字 = 4 字节)

表 22-11. 自编程库的处理时间 (2 / 3)

(3) 当使用高速系统时钟 (X1 振荡器或外部时钟输入) 并且 RAM 入口不在短直接寻址范围内

库名称		处理时间 (μs)			
		C 编译器正常模型		C 编译器 / 汇编器的静态模型	
		最小	最大	最小	最大
自编程开始库		34/f _{CPU}			
初始化库		55/f _{CPU} + 594			
模式检查库		36/f _{CPU} + 495		30/f _{CPU} + 495	
块空白检查库		179/f _{CPU} + 6429		136/f _{CPU} + 6429	
块擦除库		179/f _{CPU} + 19713	179/f _{CPU} + 268079	136/f _{CPU} + 19713	136/f _{CPU} + 268079
字写入库		333/f _{CPU} + 647 + 136 × w	333/f _{CPU} + 647 + 1647 × w	272/f _{CPU} + 647 + 136 × w	272/f _{CPU} + 647 + 1647 × w
块验证库		179/f _{CPU} + 13284		136/f _{CPU} + 13284	
自编程结束库		34/f _{CPU}			
获取信息库	选项值: 03H	180/f _{CPU} + 581		134f _{CPU} + 581	
	选项值: 04H	190/f _{CPU} + 574		144/f _{CPU} + 574	
	选项值: 05H	350/f _{CPU} + 535		304/f _{CPU} + 535	
设置信息库			80/f _{CPU} + 572934	72/f _{CPU} + 43181	72/f _{CPU} + 572934
EEPROM 写入库			333/f _{CPU} + 729 + 1722 × w	268/f _{CPU} + 729 + 209 × w	268/f _{CPU} + 729 + 1722 × w

- 备注**
1. 以上处理时间是针对写入起始地址位于内部高速 RAM 且处于内部高速振荡器稳定操作期间 (RSTS=1)。
 2. RSTS: 内部振荡器模式寄存器 (RCM) 的第 7 位
 3. f_{CPU}: CPU 运行时钟频率
 4. w: 写入数据的字个数 (1 字= 4 字节)

表 22-11. 自编程库的处理时间 (3 / 3)

(4) 当使用高速系统时钟 (X1 振荡器或外部时钟输入) 并且 RAM 入口在短直接寻址范围内

库名称		处理时间 (μs)			
		C 编译器正常模型		C 编译器 / 汇编器的静态模型	
		最小	最大	最小	最大
自编程开始库		$34/f_{\text{CPU}}$			
初始化库		$55/f_{\text{CPU}} + 272$			
模式检查库		$36/f_{\text{CPU}} + 173$		$30/f_{\text{CPU}} + 173$	
块空白检查库		$179/f_{\text{CPU}} + 6108$		$136/f_{\text{CPU}} + 6108$	
块擦除库		$179/f_{\text{CPU}} + 19371$	$179/f_{\text{CPU}} + 267738$	$136/f_{\text{CPU}} + 19371$	$136/f_{\text{CPU}} + 267738$
字写入库		$333/f_{\text{CPU}} + 247 + 136 \times w$	$333/f_{\text{CPU}} + 247 + 1647 \times w$	$272/f_{\text{CPU}} + 247 + 136 \times w$	$272/f_{\text{CPU}} + 247 + 1647 \times w$
块验证库		$179/f_{\text{CPU}} + 12964$		$136/f_{\text{CPU}} + 12964$	
自编程结束库		$34/f_{\text{CPU}}$			
获取信息库	选项值: 03H	$180/f_{\text{CPU}} + 261$		$134/f_{\text{CPU}} + 261$	
	选项值: 04H	$190/f_{\text{CPU}} + 254$		$144/f_{\text{CPU}} + 254$	
	选项值: 05H	$350/f_{\text{CPU}} + 213$		$304/f_{\text{CPU}} + 213$	
设置信息库		$80/f_{\text{CPU}} + 42839$	$80/f_{\text{CPU}} + 572592$	$72/f_{\text{CPU}} + 42839$	$72/f_{\text{CPU}} + 572592$
EEPROM 写入库		$333/f_{\text{CPU}} + 516 + 209 \times w$	$333/f_{\text{CPU}} + 516 + 1722 \times w$	$268/f_{\text{CPU}} + 516 + 209 \times w$	$268/f_{\text{CPU}} + 516 + 1722 \times w$

- 备注**
1. 以上处理时间是针对写入起始地址位于内部高速 RAM 且处于内部高速振荡器稳定操作期间 (RSTS=1)。
 2. RSTS: 内部振荡器模式寄存器 (RCM) 的第 7 位
 3. f_{CPU} : CPU 运行时钟频率
 4. w : 写入数据的字个数 (1 字=4 字节)

表 22-12. 自编程库的中断响应时间 (1 / 2)

(1) 当使用内部高速振荡器时钟

库名称	中断响应时间 (μs (最大))			
	C 编译器正常模型		C 编译器 / 汇编器的静态模型	
	RAM 入口不在短直接寻址范围内	RAM 入口在短直接寻址范围内	RAM 入口不在短直接寻址范围内	RAM 入口在短直接寻址范围内
块空白检查库	1100.9	431.9	1095.3	426.3
块擦除库	1452.9	783.9	1447.3	778.3
字写入库	1247.2	579.2	1239.2	571.2
块验证库	1125.9	455.9	1120.3	450.3
设置信息库	906.9	312.0	905.8	311.0
EEPROM 写入库	1215.2	547.2	1213.9	545.9

- 备注**
1. 以上处理时间是处于内部高速振荡器稳定操作期间 (RSTS=1)。
 2. RSTS: 内部振荡器模式寄存器 (RCM) 的第 7 位

(2) 当使用高速系统时钟 (C 编译器正常模型)

库名称	中断响应时间 (μs (最大))			
	RSTOP = 0, RSTS = 1		RSTOP = 1	
	RAM 入口不在短直接寻址范围内	RAM 入口在短直接寻址范围内	RAM 入口不在短直接寻址范围内	RAM 入口在短直接寻址范围内
块空白检查库	$179/f_{\text{CPU}} + 567$	$179/f_{\text{CPU}} + 246$	$179/f_{\text{CPU}} + 1708$	$179/f_{\text{CPU}} + 569$
块擦除库	$179/f_{\text{CPU}} + 780$	$179/f_{\text{CPU}} + 459$	$179/f_{\text{CPU}} + 1921$	$179/f_{\text{CPU}} + 782$
字写入库	$333/f_{\text{CPU}} + 763$	$333/f_{\text{CPU}} + 443$	$333/f_{\text{CPU}} + 1871$	$333/f_{\text{CPU}} + 767$
块验证库	$179/f_{\text{CPU}} + 580$	$179/f_{\text{CPU}} + 259$	$179/f_{\text{CPU}} + 1721$	$179/f_{\text{CPU}} + 582$
设置信息库	$80/f_{\text{CPU}} + 456$	$80/f_{\text{CPU}} + 200$	$80/f_{\text{CPU}} + 1598$	$80/f_{\text{CPU}} + 459$
EEPROM 写入库	$29/f_{\text{CPU}} + 767$	$29/f_{\text{CPU}} + 447$	$29/f_{\text{CPU}} + 767$	$29/f_{\text{CPU}} + 447$
	$333/f_{\text{CPU}} + 696$	$333/f_{\text{CPU}} + 376$	$333/f_{\text{CPU}} + 1838$	$333/f_{\text{CPU}} + 700$

注 EEPROM 写入库中断响应时间如果长于最大值则是最大值，取决于 f_{CPU} 的值。

- 备注**
1. f_{CPU} : CPU 运行时钟频率
 2. RSTOP: 内部振荡器模式寄存器 (RCM) 的第 0 位
 3. RSTS: 内部振荡器模式寄存器 (RCM) 的第 7 位

表 22-12. 自编程库的中断响应时间 (2 / 2)

(3) 当使用高速系统时钟 (C 编译器 / 汇编器的静态模型)

库名称	中断响应时间 (μs (最大))			
	RSTOP = 0, RSTS = 1		RSTOP = 1	
	RAM 入口不在短直接寻址范围内	RAM 入口在短直接寻址范围内	RAM 入口不在短直接寻址范围内	RAM 入口在短直接寻址范围内
块空白检查库	$136/f_{\text{CPU}} + 567$	$136/f_{\text{CPU}} + 246$	$136/f_{\text{CPU}} + 1708$	$136/f_{\text{CPU}} + 569$
块擦除库	$136/f_{\text{CPU}} + 780$	$136/f_{\text{CPU}} + 459$	$136/f_{\text{CPU}} + 1921$	$136/f_{\text{CPU}} + 782$
字写入库	$272/f_{\text{CPU}} + 763$	$272/f_{\text{CPU}} + 443$	$272/f_{\text{CPU}} + 1871$	$272/f_{\text{CPU}} + 767$
块验证库	$136/f_{\text{CPU}} + 580$	$136/f_{\text{CPU}} + 259$	$136/f_{\text{CPU}} + 1721$	$136/f_{\text{CPU}} + 582$
设置信息库	$72/f_{\text{CPU}} + 456$	$72/f_{\text{CPU}} + 200$	$72/f_{\text{CPU}} + 1598$	$72/f_{\text{CPU}} + 459$
EEPROM 写入库	$19/f_{\text{CPU}} + 767$	$19/f_{\text{CPU}} + 447$	$19/f_{\text{CPU}} + 767$	$19/f_{\text{CPU}} + 447$
	$268/f_{\text{CPU}} + 696$	$268/f_{\text{CPU}} + 376$	$268/f_{\text{CPU}} + 1838$	$268/f_{\text{CPU}} + 700$

注 EEPROM 写入库中断响应时间如果长于最大值则是最大值，取决于 f_{CPU} 的值。

- 备注
1. f_{CPU} : CPU 运行时钟频率
 2. RSTOP: 内部振荡器模式寄存器 (RCM) 的第 0 位
 3. RSTS: 内部振荡器模式寄存器 (RCM) 的第 7 位

22.8.1 启动交换功能

如果由于暂时电源故障或其它原因导致的重新写入启动区域失败，启动区域中的数据可能丢失并且程序无法通过复位重新启动。

启动交换功能被用来避免这个问题

在通过自编程擦除启动区域 0 之前，这是一个启动程序区域，事先写入一个新的启动程序到启动区域 1。当程序被正确写入启动区域 1 时，通过使用 μ PD78F8024 和 78F8025 的固件的设置信息功能来交换启动区域 1 和启动区域 0，所以启动区域 1 被用作启动区域。然后，擦除或写入原始启动程序区域，启动区域 0。

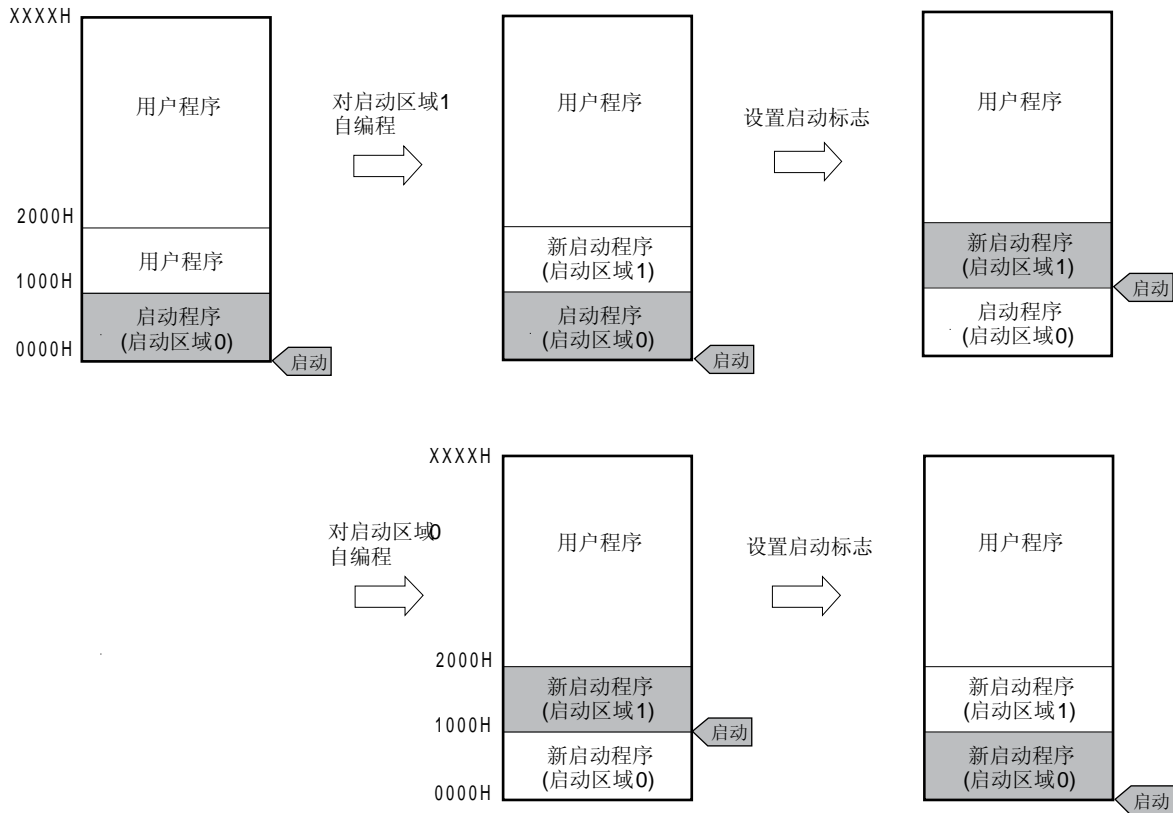
结果，即使在启动编程区域被重新写入时电源故障发生，程序也会正确执行，因为它从当前程序被复位和启动时要被交换的启动区域 1 启动。

如果程序被正确写入启动区域 0，通过使用 μ PD78F8024 和 78F8025 的固件的设置信息功能来恢复原来的启动区域。

- 注** 启动区域是 4 KB 大小区域并且启动区域 0 和 1 被启动交换功能交换。
 启动区域 0 (0000H 到 0FFFH)：原始启动程序区域
 启动区域 1 (1000H 到 1FFFH)：由启动交换功能确定的区域

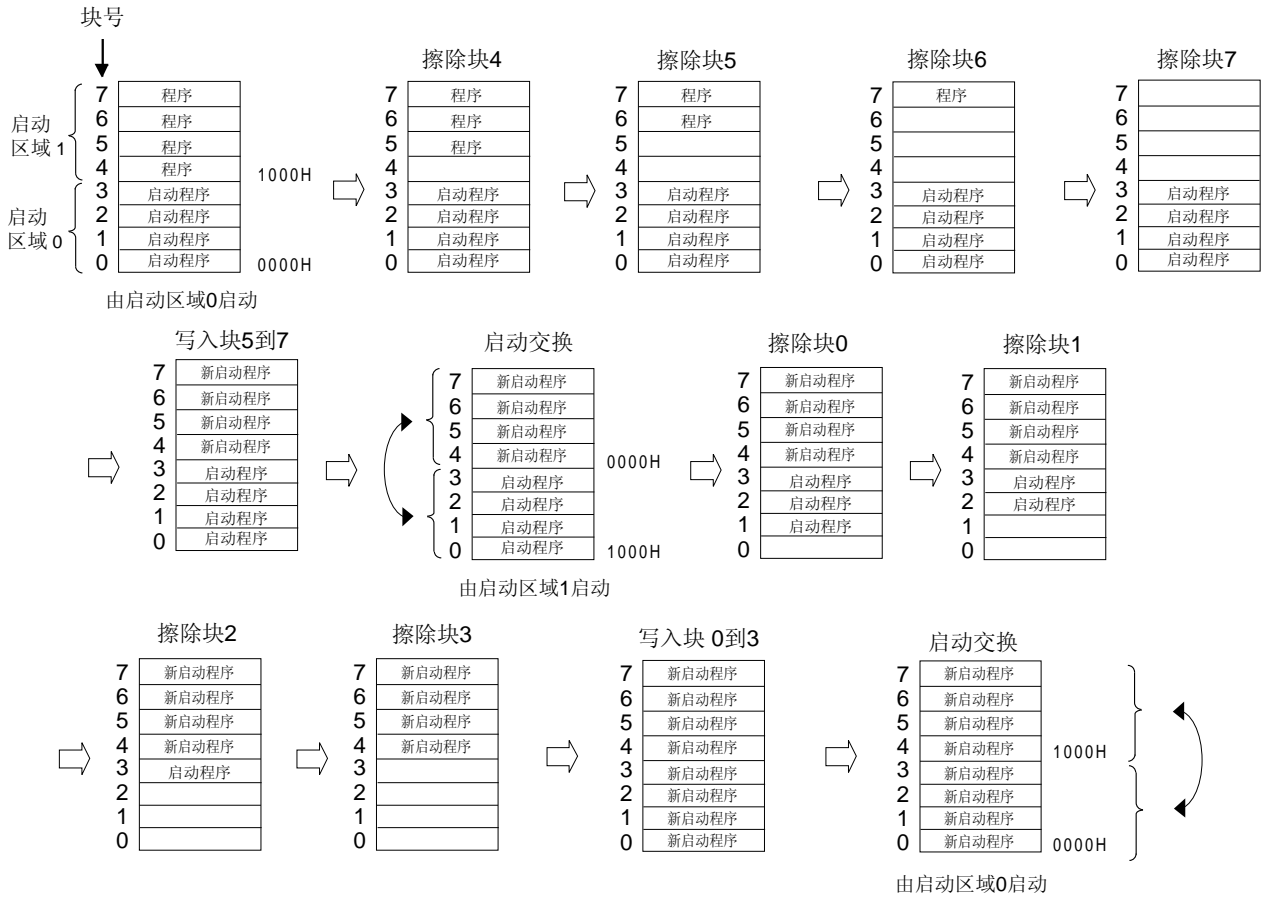
注意事项 当执行启动交换时，不要对专用 flash 存储器编程器使用 E.P.V 命令。

图 22-13. 启动交换功能



备注 启动标志设置后，当产生复位时启动区域 1 变为 0000H 到 0FFFH。

图 22-14.执行启动交换示例



第二十三章 指令集

本章给出 μ PD78F8024 和 78F8025 微控制器指令集中的指令。关于每种指令操作和操作码的详细信息，参阅单独的文档 **78K/0 系列指令用户手册 (U12326E)**。

23.1 在操作列表中使用的约定

23.1.1 操作数标识符和描述方法

操作数连同指令操作数标识符（关于详细信息，参阅汇编器规范）的描述方法在每个指令的“操作数”栏被描述。当存在两个或更多描述方法时，选择它们中的一个。大写字母和符号：**#**、**!**、**\$**和**[]**是关键字，必须按照其原样书写。每个符号有以下意义。

- **#**: 立即数规格
- **!**: 绝对地址规格
- **\$**: 相对地址规格
- **[]**: 间接地址规格

在立即数的情况下，描述一个适当的数值或标号。当使用标号时，注意必须加上**#**、**!**、**\$**和**[]**符号。

关于操作数寄存器标识符，可以使用 **r** 和 **rp** 任一功能名（**X**、**A**、**C** 等等）或者绝对名（下面表中括号中的名字，**R0**、**R1**、**R2** 等等）描述。

表 23-1. 操作数标识符和描述方法

标识符	描述方法
r	X (R0)、A (R1)、C (R2)、B (R3)、E (R4)、D (R5)、L (R6)、H (R7)
rp	AX (RP0)、BC (RP1)、DE (RP2)、HL (RP3)
sfr	特殊功能寄存器符号 ^注
sfrp	特殊功能寄存器符号（16 位可操作寄存器，只能是偶数地址） ^注
saddr	FE20H 到 FF1FH 立即数或标号
saddrp	FE20H 到 FF1FH 立即数或标号（只能是偶数地址）
addr16	0000H 到 FFFFH 立即数或标号 （对于 16 位数据转移指令只能是偶数地址）
addr11	0800H 到 0FFFH 立即数或标号
addr5	0040H 到 007FH 立即数或标号（只能是偶数地址）
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 不能使用这些操作数访问地址 FFD0H 到 FFD7H。

备注 关于特殊功能寄存器符号，见表 **4-6 特殊功能寄存器列表**。

23.1.2 操作栏的说明

- A: A 寄存器; 8 位累加器
- X: X 寄存器
- B: B 寄存器
- C: C 寄存器
- D: D 寄存器
- E: E 寄存器
- H: H 寄存器
- L: L 寄存器
- AX: AX 寄存器对; 16 位累加器
- BC: BC 寄存器对
- DE: DE 寄存器对
- HL: HL 寄存器对
- PC: 程序计数器
- SP: 堆栈指针
- PSW: 程序状态字
- CY: 进位标志
- AC: 辅助进位标志
- Z: 零标志
- RBS: 寄存器组选择标志
- IE: 中断请求使能标志
- (): 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
- ∧: 逻辑乘 (AND)
- ∨: 逻辑加 (OR)
- ⋈: 归一的逻辑加 (归一 OR)
- : 反转数据
- addr16: 16 位立即数或标号
- jdisp8: 有符号 8 位数据 (位移值)

23.1.3 标志栏的说明

- (空白): 不影响
- 0: 清除为 0
- 1: 置位为 1
- x: 根据结果置位/清除
- R: 恢复上次保存的值

23.2 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	AC	CY		
8 位数据转移	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, !addr16		3	8	9	A ← (addr16)				
		!addr16, 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	A, r	注 3	1	2	–	A ↔ r				
		A, saddr		2	4	6	A ↔ (saddr)				
		A, sfr		2	–	6	A ↔ sfr				
		A, !addr16		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. 当访问内部高速 RAM 区域，或者针对没有数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时
 3. 除去“r = A”

- 备注
1. 一个指令时钟周期是一个由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（fCPU）周期。
 2. 这个时钟周期提供给内部 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 位数据转移	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) + CY	x	x	x
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	x	x	x
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	x	x	x
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	x	x	x
A, [HL + C]		2	8	9	A, CY ← A + (HL + C) + CY	x	x	x	

- 注
1. 当访问内部高速 RAM 区域，或者针对没有数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时
 3. 只有当 rp = BC, DE 或 HL
 4. 除去 “r = A”

- 备注
1. 一个指令时钟周期是一个由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (fcpu) 周期。
 2. 这个时钟周期提供给内部 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. 当访问内部高速 RAM 区域，或者针对没有数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时
 3. 除去“r = A”

- 备注
1. 一个指令时钟周期是一个由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（f_{cpu}）周期。
 2. 这个时钟周期提供给内部 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, laddr16	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, laddr16	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, laddr16	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. 当访问内部高速 RAM 区域，或者针对没有数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时
 3. 除去“r = A”

- 备注
1. 一个指令时钟周期是一个由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（f_{cpu}）周期。
 2. 这个时钟周期提供给内部 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			
自增/自减	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			
旋转	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	-	加完之后十进制调整累加器	x	x	x
	ADJBS		2	4	-	减完之后十进制调整累加器	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
[HL].bit, CY	2	6	8	(HL).bit ← CY					

- 注 1. 当访问内部高速 RAM 区域，或者针对没有数据访问的指令
2. 当访问内部高速 RAM 以外的区域时

- 备注 1. 一个指令时钟周期是一个由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (fCPU) 周期。
2. 这个时钟周期提供给内部 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	$CY \leftarrow CY \wedge (\text{saddr.bit})$	×
		CY, sfr.bit	3	–	7	$CY \leftarrow CY \wedge \text{sfr.bit}$	×
		CY, A.bit	2	4	–	$CY \leftarrow CY \wedge A.\text{bit}$	×
		CY, PSW.bit	3	–	7	$CY \leftarrow CY \wedge \text{PSW.bit}$	×
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \wedge (\text{HL}).\text{bit}$	×
	OR1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \vee (\text{saddr.bit})$	×
		CY, sfr.bit	3	–	7	$CY \leftarrow CY \vee \text{sfr.bit}$	×
		CY, A.bit	2	4	–	$CY \leftarrow CY \vee A.\text{bit}$	×
		CY, PSW.bit	3	–	7	$CY \leftarrow CY \vee \text{PSW.bit}$	×
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \vee (\text{HL}).\text{bit}$	×
	XOR1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \oplus (\text{saddr.bit})$	×
		CY, sfr.bit	3	–	7	$CY \leftarrow CY \oplus \text{sfr.bit}$	×
		CY, A.bit	2	4	–	$CY \leftarrow CY \oplus A.\text{bit}$	×
		CY, PSW.bit	3	–	7	$CY \leftarrow CY \oplus \text{PSW.bit}$	×
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \oplus (\text{HL}).\text{bit}$	×
	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	$\text{PSW.bit} \leftarrow 1$	× × ×
		[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	$\text{PSW.bit} \leftarrow 0$	× × ×
		[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	–	$CY \leftarrow \neg CY$	×

- 注
1. 当访问内部高速 RAM 区域，或者针对没有数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时

- 备注
1. 一个指令时钟周期是一个由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（f_{cpu}）周期。
 2. 这个时钟周期提供给内部 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	laddr16	3	7	-	$(SP - 1) \leftarrow (PC + 3)_H, (SP - 2) \leftarrow (PC + 3)_L,$ $PC \leftarrow \text{addr16}, SP \leftarrow SP - 2$			
	CALLF	laddr11	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	$AX \leftarrow SP$				
无条件跳转	BR	laddr16	3	6	-	$PC \leftarrow \text{addr16}$			
		\$addr16	2	6	-	$PC \leftarrow PC + 2 + \text{jdisp8}$			
		AX	2	8	-	$PC_H \leftarrow A, PC_L \leftarrow X$			
条件跳转	BC	\$addr16	2	6	-	$PC \leftarrow PC + 2 + \text{jdisp8}$ 如果 $CY = 1$			
	BNC	\$addr16	2	6	-	$PC \leftarrow PC + 2 + \text{jdisp8}$ 如果 $CY = 0$			
	BZ	\$addr16	2	6	-	$PC \leftarrow PC + 2 + \text{jdisp8}$ 如果 $Z = 1$			
	BNZ	\$addr16	2	6	-	$PC \leftarrow PC + 2 + \text{jdisp8}$ 如果 $Z = 0$			

- 注
1. 当访问内部高速 RAM 区域，或者针对没有数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时

- 备注
1. 一个指令时钟周期是一个由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（f_{CPU}）周期。
 2. 这个时钟周期提供给内部 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	
条件跳转	BT	saddr.bit, \$addr16	3	8	9	PC ← PC + 3 + jdisp8 如果 (saddr.如果) = 1		
		sfr.bit, \$addr16	4	–	11	PC ← PC + 4 + jdisp8 如果 sfr.bit = 1		
		A.bit, \$addr16	3	8	–	PC ← PC + 3 + jdisp8 如果 A.bit = 1		
		PSW.bit, \$addr16	3	–	9	PC ← PC + 3 + jdisp8 如果 PSW.bit = 1		
		[HL].bit, \$addr16	3	10	11	PC ← PC + 3 + jdisp8 如果 (HL).bit = 1		
	BF	saddr.bit, \$addr16	4	10	11	PC ← PC + 4 + jdisp8 如果 (saddr.bit) = 0		
		sfr.bit, \$addr16	4	–	11	PC ← PC + 4 + jdisp8 如果 sfr.bit = 0		
		A.bit, \$addr16	3	8	–	PC ← PC + 3 + jdisp8 如果 A.bit = 0		
		PSW.bit, \$addr16	4	–	11	PC ← PC + 4 + jdisp8 如果 PSW.bit = 0		
		[HL].bit, \$addr16	3	10	11	PC ← PC + 3 + jdisp8 如果 (HL).bit = 0		
	BTCLR	saddr.bit, \$addr16	4	10	12	PC ← PC + 4 + jdisp8 如果 (saddr.bit) = 1 然后复位 (saddr.bit)		
		sfr.bit, \$addr16	4	–	12	PC ← PC + 4 + jdisp8 如果 sfr.bit = 1 然后复位 sfr.bit		
		A.bit, \$addr16	3	8	–	PC ← PC + 3 + jdisp8 如果 A.bit = 1 然后复位 A.bit		
		PSW.bit, \$addr16	4	–	12	PC ← PC + 4 + jdisp8 如果 PSW.bit = 1 然后复位 PSW.bit	× × ×	
		[HL].bit, \$addr16	3	10	12	PC ← PC + 3 + jdisp8 如果 (HL).bit = 1 然后复位 (HL).bit		
	DBNZ	B, \$addr16	2	6	–	B ← B – 1, 然后 PC ← PC + 2 + jdisp8 如果 B ≠ 0		
		C, \$addr16	2	6	–	C ← C – 1, 然后 PC ← PC + 2 + jdisp8 如果 C ≠ 0		
		saddr, \$addr16	3	8	10	(saddr) ← (saddr) – 1, 然后 PC ← PC + 3 + jdisp8 如果 (saddr) ≠ 0		
	CPU 控制	SEL	Rbn	2	4	–	RBS1, 0 ← n	
		NOP		1	2	–	不操作	
EI			2	–	6	IE ← 1 (使能中断)		
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. 当访问内部高速 RAM 区域, 或者针对没有数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时

- 备注
1. 一个指令时钟周期是一个由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{cpu}) 周期。
 2. 这个时钟周期提供给内部 ROM 程序。

23.3 按寻址类型列出指令

(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	laddr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	\$addr16	1	无
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B、C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
laddr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

注 除去“r = A”

(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 ^註	sfrp	saddrp	laddr16	SP	无
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW ^註						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
laddr16		MOVW						
SP	MOVW	MOVW						

注 只有当 rp = BC、DE、HL

(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
组合指令					BT BF BTCLR DBNZ

(5) 其它指令

ADJBA、ADJBS、BRK、RET、RETI、RETB、SEL、NOP、EI、DI、HALT、STOP

第二十四章 电气特性

24.1 CPU 部分

最大绝对额定值 (T_A = 25°C) (1/2)

参数	符号	条件		额定值	单位
电源电压	V _{DD}			-0.5 至 +6.5	V
	V _{SS}			-0.5 至 +0.3	V
	AV _{REF}			-0.5 至 V _{DD} + 0.3 [‡]	V
	AV _{SS}			-0.5 至 +0.3	V
REGC 引脚输入电压	V _{I_{REGC}}			-0.5 至 +3.6 和 -0.5 至 V _{DD}	V
输入电压	V _{I1}	P00, P01, P10 至 P17, P20 至 P23, P30 至 P33, P120 至 P122, X1, X2, RESET, FLMD0		-0.3 至 V _{DD} + 0.3 [‡]	V
	V _{I2}	P60, P61 (N-ch 漏极开路)		-0.3 至 +6.5	V
输出电压	V _O			-0.3 至 V _{DD} + 0.3 [‡]	V
模拟输入电压	V _{AN}	ANI0 至 ANI3		-0.3 至 AV _{REF} + 0.3 [‡] 和 -0.3 至 V _{DD} + 0.3 [‡]	V
输出电流, 高	I _{OH}	每个引脚	P00, P01, P10 至 P17, P30 至 P33, P120	-10	mA
		所有引脚总和 -80 mA	P00, P01, P120	-25	mA
			P10 至 P17, P30 至 P33	-55	mA
		每个引脚	P20 至 P23	-0.5	mA
		所有引脚总和		-2	mA
		每个引脚	P121, P122	-1	mA
		所有引脚总和		-2	mA

注 必须为 6.5 V 或更低。

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。

最大绝对额定值 ($T_A = 25^\circ\text{C}$) (2/2)

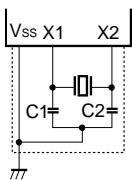
参数	符号	条件		额定值	单位
输出电流, 低	I _{OL}	每个引脚	P00, P01, P10 至 P17, P30 至 P33, P60, P61, P120	30	mA
		所有引脚总和 200 mA	P00, P01, P120	60	mA
			P10 至 P17, P30 至 P33, P60, P61	140	mA
		每个引脚	P20 to P23	1	mA
		所有引脚总和		4	mA
		每个引脚	P121, P122	4	mA
		所有引脚总和		8	mA
工作环境温度	T _A			-40 至 +85	°C
存储温度	T _{stg}			-65 至 +150	°C

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

($T_A = -40$ 至 $+85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ^{註1}, $V_{SS} = AV_{SS} = 0\text{ V}$)

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 晶体振荡器		X1 时钟振荡频率 (f_x) ^{註2}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	1.0 ^{註3}		20.0	MHz
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	1.0		5.0	

- 注
1. 未使用恒流驱动器时: $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
使用恒流驱动器时: $4.5\text{ V} \leq V_{DD} = CV_{DD} \leq 5.5\text{ V}$
 2. 只显示振荡器特性。关于指令执行时间请参照 **AC 特性**。
 3. 在线编程过程中使用 UART6 时为最小值 2.0 MHz。

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 V_{SS} 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. 由于复位释放后通过内部高速振荡时钟启动 CPU，因此用户需要使用振荡稳定时间计数状态寄存器 (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)。

备注 对于振荡器的选择和振荡器常数，需要客户自己对振荡器进行评估，或者通过振荡器制造商来进行评估。

内部振荡器特性

($T_A = -40$ 至 $+85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ^{注1}, $V_{SS} = AV_{SS} = 0\text{ V}$)

振荡器	参数	条件	最小	典型值	最大	单位	
8 MHz 内部振荡器	内部高速振荡时钟频率 (f_{RH}) ^{注2}	RSTS = 1	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	7.6	8.0	8.4	MHz
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	7.6	8.0	10.4	MHz
		RSTS = 0		2.48	5.6	9.86	MHz
240 kHz 内部振荡器	内部低速振荡时钟频率 (f_{RL})	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		216	240	264	kHz
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		192	240	264	kHz

- 注 1. 未使用恒流驱动器时: $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
 使用恒流驱动器时: $4.5\text{ V} \leq V_{DD} = CV_{DD} \leq 5.5\text{ V}$
2. 只显示振荡器特性。关于指令执行时间请参照 **AC 特性**。

备注 RSTS: 内部振荡模式寄存器 (RCM) 的第 7 位

DC 特性 (1/4)

(TA = -40 至 +85°C, 1.8 V ≤ VDD ≤ 5.5 V^{注1}, AVREF ≤ VDD, VSS = AVSS = 0 V)

参数	符号	条件	最小.	典型值.	最大.	单位
输出电流, 高 ^{注2}	IOH1	P00, P01, P10 至 P17, P30 至 P33, P120 的每个引脚	4.0 V ≤ VDD ≤ 5.5 V		-3.0	mA
			2.7 V ≤ VDD < 4.0 V		-2.5	mA
			1.8 V ≤ VDD < 2.7 V		-1.0	mA
		P00, P01, P120 的全部 ^{注4}	4.0 V ≤ VDD ≤ 5.5 V		-12.9	mA
			2.7 V ≤ VDD < 4.0 V		-10	mA
			1.8 V ≤ VDD < 2.7 V		-4.3	mA
		P10 至 P17, P30 至 P33 的全部 ^{注4}	4.0 V ≤ VDD ≤ 5.5 V		-30.0	mA
			2.7 V ≤ VDD < 4.0 V		-19.0	mA
			1.8 V ≤ VDD < 2.7 V		-10.0	mA
	所有引脚 ^{注4}	4.0 V ≤ VDD ≤ 5.5 V		-50.0	mA	
		2.7 V ≤ VDD < 4.0 V		-29.0	mA	
		1.8 V ≤ VDD < 2.7 V		-15.0	mA	
	IOH2	P20 至 P23 的每个引脚	AVREF = VDD		-0.1	mA
P121, P122 的每个引脚				-0.1	mA	
输出电流, 低 ^{注3}	IOL1	P00, P01, P10 至 P17, P30 至 P33, P120 的每个引脚	4.0 V ≤ VDD ≤ 5.5 V		8.5	mA
			2.7 V ≤ VDD < 4.0 V		5.0	mA
			1.8 V ≤ VDD < 2.7 V		2.0	mA
		P60, P61 的每个引脚	4.0 V ≤ VDD ≤ 5.5 V		15.0	mA
			2.7 V ≤ VDD < 4.0 V		5.0	mA
			1.8 V ≤ VDD < 2.7 V		2.0	mA
		P00, P01, P120 的全部 ^{注4}	4.0 V ≤ VDD ≤ 5.5 V		20.0	mA
			2.7 V ≤ VDD < 4.0 V		15.0	mA
			1.8 V ≤ VDD < 2.7 V		8.6	mA
		P10 至 P17, P30 至 P33, P60, P61 的全部 ^{注4}	4.0 V ≤ VDD ≤ 5.5 V		37.0	mA
			2.7 V ≤ VDD < 4.0 V		35.0	mA
			1.8 V ≤ VDD < 2.7 V		20.0	mA
		所有引脚 ^{注4}	4.0 V ≤ VDD ≤ 5.5 V		65.0	mA
			2.7 V ≤ VDD < 4.0 V		50.0	mA
			1.8 V ≤ VDD < 2.7 V		28.6	mA
	IOL2	P20 至 P23 的每个引脚	AVREF = VDD		0.4	mA
		P121, P122 的每个引脚			0.4	mA

- 注 1. 未使用恒流驱动器时: $1.8 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$
 使用恒流驱动器时: $4.5 \text{ V} \leq V_{DD} = CV_{DD} \leq 5.5 \text{ V}$
2. 保证设备操作的电流, 即使电流从 VDD 流向一个输出引脚。
3. 保证设备操作的电流, 即使电流从一个输出引脚流向 GND。
4. 占空比为 70% 的 (输出电流时间为 $0.7 \times t$, 没有输出时间为 $0.3 \times t$, t 为描述时间) 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。
- IOH 的占空比为 n% 处: 引脚的总共输出电流 = $(I_{OH} \times 0.7) / (n \times 0.01)$
 <例> 占空比为 50% 处, IOH = 20.0 mA
 引脚的总共输出电流 = $(20.0 \times 0.7) / (50 \times 0.01) = 28.0 \text{ mA}$
- 然而, 允许流入一个引脚的电流不会根据占空比而改变。比最大绝对额定值更高的电流不能流入一个引脚。

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 (2/4)

($T_A = -40$ 至 $+85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}^{\ddagger}$, $AV_{REF} \leq V_{DD}$, $V_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小.	典型值.	最大.	单位
输入电压, 高	V_{IH1}	P12, P13, P15, P121, P122, EXCLK	$0.7V_{DD}$		V_{DD}	V
	V_{IH2}	P00, P01, P10, P11, P14, P16, P17, P30 至 P33, P120, RESET	$0.8V_{DD}$		V_{DD}	V
	V_{IH3}	P20 至 P23	$AV_{REF} = V_{DD}$		AV_{REF}	V
	V_{IH4}	P60, P61			6.0	V
输入电压, 低	V_{IL1}	P12, P13, P15, P60, P61, P121, P122, EXCLK	0		$0.3V_{DD}$	V
	V_{IL2}	P00, P01, P10, P11, P14, P16, P17, P30 至 P33, P120, RESET	0		$0.2V_{DD}$	V
	V_{IL3}	P20 至 P23	$AV_{REF} = V_{DD}$		$0.3AV_{REF}$	V
输出电压, 高	V_{OH1}	P00, P01, P10 至 P17, P30 至 P33, P120	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OH1} = -3.0\text{ mA}$		$V_{DD} - 0.7$	V
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $I_{OH1} = -2.5\text{ mA}$		$V_{DD} - 0.5$	V
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$, $I_{OH1} = -1.0\text{ mA}$		$V_{DD} - 0.5$	V
	V_{OH2}	P20 至 P23	$AV_{REF} = V_{DD}$, $I_{OH2} = -100\ \mu\text{A}$		$V_{DD} - 0.5$	V
			P121, P122	$I_{OH2} = -100\ \mu\text{A}$		$V_{DD} - 0.5$
输出电压, 低	V_{OL1}	P00, P01, P10 至 P17, P30 至 P33, P120	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 8.5\text{ mA}$		0.7	V
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $I_{OL1} = 5.0\text{ mA}$		0.7	V
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$, $I_{OL1} = 2.0\text{ mA}$		0.5	V
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$, $I_{OL1} = 0.5\text{ mA}$		0.4	V
	V_{OL2}	P20 至 P23	$AV_{REF} = V_{DD}$, $I_{OL2} = 0.4\text{ mA}$		0.4	V
			P121, P122	$I_{OL2} = 0.4\text{ mA}$		0.4
	V_{OL3}	P60, P61	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 15.0\text{ mA}$		2.0	V
			$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 5.0\text{ mA}$		0.4	V
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $I_{OL1} = 5.0\text{ mA}$		0.6	V
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $I_{OL1} = 3.0\text{ mA}$		0.4	V
$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$, $I_{OL1} = 2.0\text{ mA}$				0.4	V	

注 未使用恒流驱动器时: $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
 使用恒流驱动器时: $4.5\text{ V} \leq V_{DD} = CV_{DD} \leq 5.5\text{ V}$

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 (3/4)

($T_A = -40$ 至 $+85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}^{\ddagger}$, $AV_{REF} \leq V_{DD}$, $V_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小.	典型值.	最大.	单位	
输入漏电流, 高	I _{LIH1}	P00, P01, P10 至 P17, P30 至 P33, P60, P61, P120, FLMD0, RESET	V _I = V _{DD}		1	μA	
	I _{LIH2}	P20 至 P23	V _I = AV _{REF} = V _{DD}		1	μA	
	I _{LIH3}	P121, P122 (X1, X2)	V _I = V _{DD}		1	μA	
			I/O 端口模式				
		OSC 模式		20	μA		
输入漏电流, 低	I _{LIL1}	P00, P01, P10 至 P17, P30 至 P33, P60, P61, P120, FLMD0, RESET	V _I = V _{SS}		-1	μA	
	I _{LIL2}	P20 至 P23	V _I = V _{SS} , AV _{REF} = V _{DD}		-1	μA	
	I _{LIL3}	P121, P122 (X1, X2)	V _I = V _{SS}		-1	μA	
			I/O 端口模式				
		OSC 模式		-20	μA		
上拉电阻	R _U	V _I = V _{SS}		10	20	100	kΩ
FLMD0 电源电压	V _{IL}	在正常工作模式中		0		0.2V _{DD}	V
	V _{IH}	在自编程模式中		0.8V _{DD}		V _{DD}	V

注 未使用恒流驱动器时: $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
 使用恒流驱动器时: $4.5\text{ V} \leq V_{DD} = CV_{DD} \leq 5.5\text{ V}$

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 (4/4)

(TA = -40 至 +85°C, 1.8 V ≤ VDD ≤ 5.5 V^{‡1}, AVREF ≤ VDD, VSS = AVSS = 0 V)

参数	符号	条件		最小.	典型值.	最大.	单位	
电源电流 ^{‡2}	IDD1	操作模式	fXH = 20 MHz, ^{‡3} VDD = 5.0 V	方波输入		3.2	5.5	mA
				振荡器连接		4.5	6.9	
			fXH = 10 MHz, ^{‡3, 4} VDD = 5.0 V	方波输入		1.6	2.8	mA
				振荡器连接		2.3	3.9	
			fXH = 10 MHz, ^{‡3, 4} VDD = 3.0 V	方波输入		1.5	2.7	mA
				振荡器连接		2.2	3.2	
			fXH = 5 MHz, ^{‡3, 4} VDD = 3.0 V	方波输入		0.9	1.6	mA
				振荡器连接		1.3	2.0	
	fXH = 5 MHz, ^{‡3, 4} VDD = 2.0 V	方波输入		0.7	1.4	mA		
		振荡器连接		1.0	1.6			
			fRH = 8 MHz, VDD = 5.0 V ^{‡5}		1.4	2.5	mA	
	IDD2	HALT 模式	fXH = 20 MHz, ^{‡3} VDD = 5.0 V	方波输入		0.8	2.6	mA
				振荡器连接		2.0	4.4	
			fXH = 10 MHz, ^{‡3, 4} VDD = 5.0 V	方波输入		0.4	1.3	mA
振荡器连接					1.0	2.4		
fXH = 5 MHz, ^{‡3, 4} VDD = 3.0 V			方波输入		0.2	0.65	mA	
			振荡器连接		0.5	1.1		
		fRH = 8 MHz, VDD = 5.0 V ^{‡5}		0.4	1.2	mA		
IDD3 ^{‡6}	STOP 模式				1	20	μA	
		TA = -40 至 +70°C			1	10	μA	
A/D 转换器工作电流	IADC ^{‡7}	2.3 V ≤ AVREF ≤ VDD, ADCE = 1			0.86	1.9	mA	
看门狗定时器工作电流	IWD1 ^{‡8}	240 kHz 内部低速振荡时钟操作过程中			5	10	μA	
LVI 工作电流	ILVI ^{‡9}				9	18	μA	

- 备注
1. fXH: 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. fRH: 内部高速振荡时钟频率

(注释在下一页)

- 注
1. 未使用恒流驱动器时: $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
使用恒流驱动器时: $4.5\text{ V} \leq V_{DD} = CV_{DD} \leq 5.5\text{ V}$
 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, 其中包括外围硬件工作电流以及输入漏电电流。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 8 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 A/D 转换器, 看门狗定时器以及 LVI 电路的电流。
 4. 当 AMPH (时钟操作模式选择寄存器 (OSCCTL) 的 0 位) = 0 时。
 5. 不包括 X1 振荡器以及 240 kHz 内部振荡器的工作电流, 也不包括流入 A/D 转换器, 看门狗定时器以及 LVI 电路的电流。
 6. 不包括 240 kHz 内部振荡器的工作电流, 也不包括流入 A/D 转换器, 看门狗定时器以及 LVI 电路的电流。
 7. 只流入 A/D 转换器 (AV_{REF}) 的电流。当 A/D 转换器在操作模式或 HALT 模式中进行操作时, CPU 部分的电流值等于 I_{DD1} 或 I_{DD2} 与 I_{ADC} 的和。 $\mu\text{PD78F8024}$ 或者 78F8025 的电流值是 CPU 部分的电流值与恒流驱动器部分的电流值的总和。关于恒流驱动器模块的电流值的详细信息, 请参考 24.2 带有可选降压或升压转换器的恒流驱动器模块部分。
 8. 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)。看门狗定时器工作时 CPU 部分的电流值等于 I_{DD1} , I_{DD2} 或 I_{DD3} 与 I_{WDT} 的总和。 $\mu\text{PD78F8024}$ 或者 78F8025 的电流值是 CPU 部分的电流值与恒流驱动器部分的电流值的和。关于恒流驱动器模块的电流值的详细信息, 请参考 24.2 带有可选降压或升压转换器的恒流驱动器模块部分。
 9. 仅流入 LVI 电路的电流。LVI 电路工作时 CPU 部分的电流值等于 I_{DD1} , I_{DD2} 或 I_{DD3} 与 I_{LVI} 的和。 $\mu\text{PD78F8024}$ 或者 78F8025 的电流值是 CPU 部分的电流值与恒流驱动器部分的电流值的总和。关于恒流驱动器模块的电流值的详细信息, 请参考 24.2 带有可选降压或升压转换器的恒流驱动器模块部分。

AC 特性

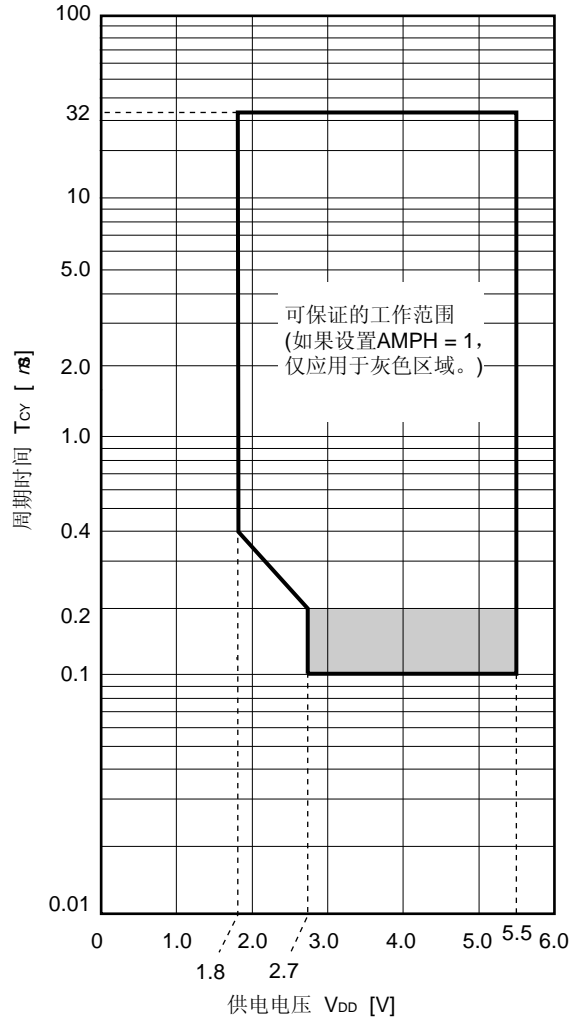
(1) 基本操作

 $(T_A = -40 \text{ 至 } +85^\circ\text{C}, 1.8 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}^{\#1}, AV_{REF} \leq V_{DD}, V_{SS} = AV_{SS} = 0 \text{ V})$

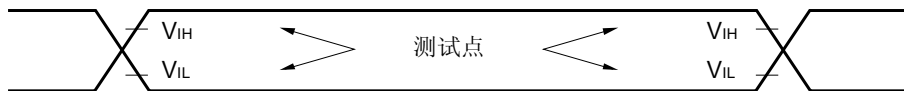
参数	符号	条件	最小.	典型值.	最大.	单位	
指令周期 (最小指令执行时间)	TCY	主系统时钟 (fXP) 操作	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0.1		32	μs
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	$0.4^{\#2}$		32	μs
外围硬件时钟频率	fPRS	fPRS = fXH (XSEL = 1)	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			20	MHz
			$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}^{\#3}$			20	MHz
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$			5	MHz
		fPRS = fRH (XSEL = 0)	$2.7 \text{ V} \leq V_{DD} < 5.5 \text{ V}$	7.6		8.4	MHz
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}^{\#4}$	7.6		10.4	MHz
外部主系统时钟频率	fEXCLK	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$1.0^{\#5}$		20.0	MHz	
		$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	1.0		5.0	MHz	
外部主系统时钟输入高电平宽度, 低电平宽度	tEXCLKH.	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	24			ns	
	tEXCLKL	$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	96			ns	
TI000, TI010 输入高电平宽度, 低电平宽度	tTIH0, tTIL0	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$2/f_{sam} + 0.1^{\#6}$			μs	
		$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	$2/f_{sam} + 0.2^{\#6}$			μs	
		$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	$2/f_{sam} + 0.5^{\#6}$			μs	
TI50, TI51 输入频率	fTI5	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			10	MHz	
		$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}$			10	MHz	
		$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$			5	MHz	
TI50, TI51 输入高电平宽度, 低 电平宽度	tTIH5, tTIL5	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	50			ns	
		$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	50			ns	
		$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	100			ns	
中断输入高电平宽度, 低电平宽度	tINTH, tINTL		1			μs	
RESET 低电平宽度	tRSL		10			μs	

- 注
1. 未使用恒流驱动器时: $1.8 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$
使用恒流驱动器时: $4.5 \text{ V} \leq V_{DD} = CV_{DD} \leq 5.5 \text{ V}$
 2. 使用 8 MHz 内部振荡器操作时为 $0.38 \mu\text{s}$ 。
 3. 主系统时钟频率的特性。由外围功能设置的分频时钟为 $f_{XH}/2$ (10 MHz) 或更低。
 4. 主系统时钟频率特性。由外围功能设置的分频时钟为 $f_{RH}/2$ (10 MHz) 或更低。
 5. 在线编程过程中使用 UART6 时为最小值 2.0 MHz。
 6. 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 $f_{sam} = f_{PRS}$ 。

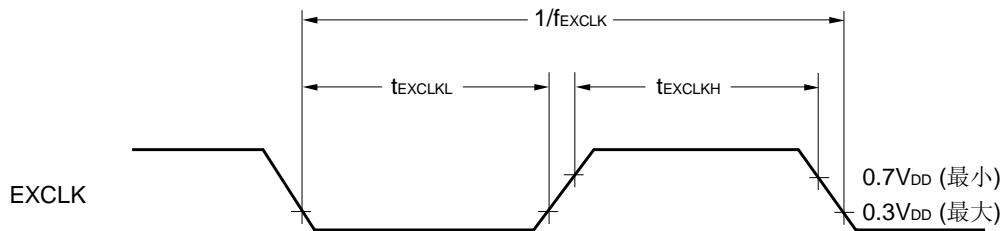
T_{CY} vs. V_{DD} (主系统时钟操作)



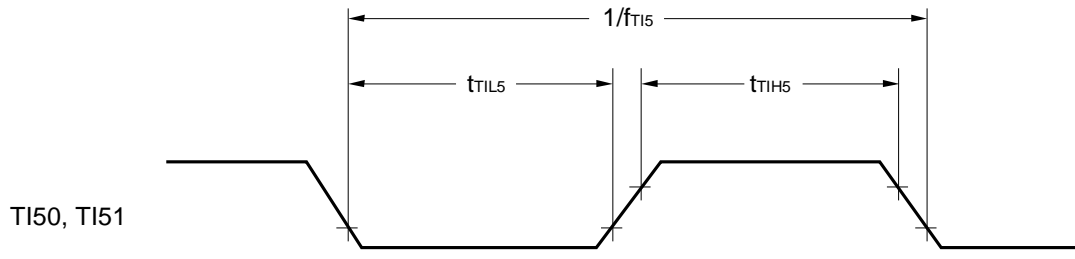
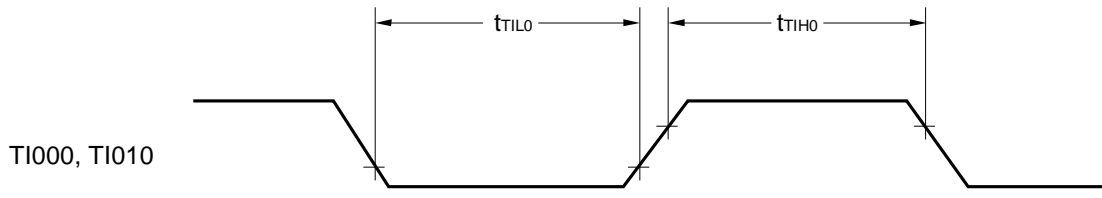
AC 时序测试点 (不包括外部主系统时钟)



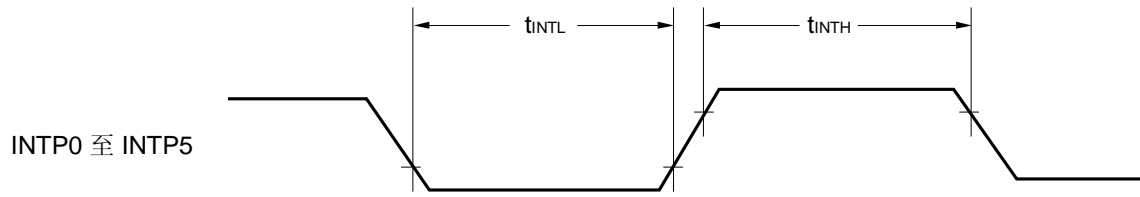
外部主系统时钟的时序



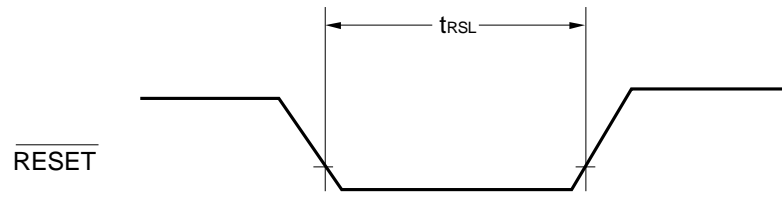
TI 的时序



中断请求输入时序



$\overline{\text{RESET}}$ 输入时序



(2) 串行接口

($T_A = -40$ 至 $+85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}^{\ddagger}$, $AV_{REF} \leq V_{DD}$, $V_{SS} = AV_{SS} = 0\text{ V}$)

注 未使用恒流驱动器时: $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$
 使用恒流驱动器时: $4.5\text{ V} \leq V_{DD} = CV_{DD} \leq 5.5\text{ V}$

(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					625	kbps

(b) UART0 (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					625	kbps

(c) IIC0

参数	符号	条件	标准模式		高速模式		单位
			最小.	最大.	最小.	最大.	
SCL0 时钟频率	f _{SCL}		0	100	0	400	kHz
重启条件的设置时间	t _{SU} : STA		4.7	–	0.6	–	μs
保持时间 ^{‡1}	t _{HD} : STA		4.0	–	0.6	–	μs
SCL0 = “L” 时的保持时间	t _{LOW}	内部时钟操作	4.7	–	1.3	–	μs
SCL0 = “H” 时的保持时间	t _{HIGH}		4.0	–	0.6	–	μs
数据设置时间 (接收)	t _{SU} : DAT		250	–	100	–	ns
数据保持时间 (发送) ^{‡2}	t _{HD} : DAT	选择 $f_w = f_{XH}/2^N$ ^{‡3}	0	3.45	0	0.9 ^{‡4}	μs
		选择 $f_w = f_{RH}/2^N$ ^{‡3}	0	3.45	0	1.00 ^{‡5}	
停止条件的设置时间	t _{SU} : STO		4.0	–	0.6	–	μs
总线空闲时间	t _{BUF}		4.7	–	1.3	–	μs

- 注
1. 检测到启动/重启条件时, 在保持时间过后生成第一个时钟脉冲。
 2. 在正常传输过程中会产生 t_{HD}: DAT 的最大值 (MAX.), 并且会在 ACK (响应) 时序中插入一个等待状态。
 3. f_w 表示由 IICCL 和 IICX0 寄存器所选择的 IIC0 传输时钟。
 4. 选择 f_w ≥ 4.4 MHz 时
 5. 选择 f_w < 4.4 MHz 时

(d) CSI10 (主模式, SCK10... 内部时钟输出)

参数	符号	条件	最小.	典型值.	最大.	单位
SCK10 周期时间	t_{KCY1}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	160			ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	250			ns
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	500			ns
SCK10 高/低电平宽度	t_{KH1} , t_{KL1}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$t_{KCY1}/2 - 15^{\#1}$			ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	$t_{KCY1}/2 - 25^{\#1}$			ns
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$t_{KCY1}/2 - 50^{\#1}$			ns
SI10 设置时间 (至 SCK10 \uparrow)	t_{SIK1}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	55			ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	80			ns
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	170			ns
SI10 操作时间 (从 SCK10 \uparrow)	t_{KSI1}		30			ns
从 SCK10 \downarrow 至 SO10 输出的延迟时间	t_{KSO1}	$C = 50\text{ pF}^{\#2}$			40	ns

- 注 1. 采用高速系统时钟 (f_{XH}) 时的值
2. C 是 SCK10 和 SO10 输出线路的负载电容。

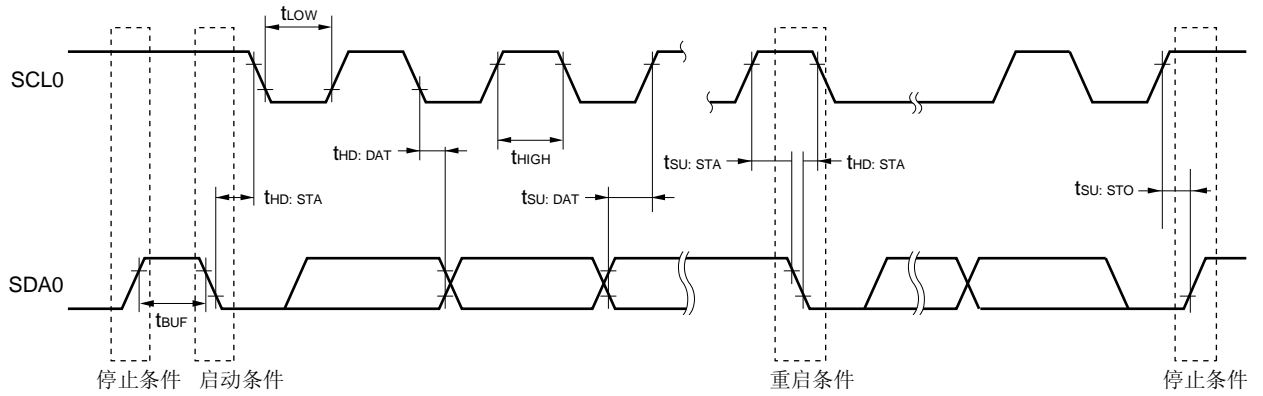
(e) CSI10 (从模式, SCK10... 外部时钟输入)

参数	符号	条件	最小.	典型值.	最大.	单位
SCK10 周期时间	t_{KCY2}		400			ns
SCK10 高/低电平宽度	t_{KH2} , t_{KL2}		$t_{KCY2}/2$			ns
SI10 设置时间 (到 SCK10 \uparrow)	t_{SIK2}		80			ns
SI10 保持时间 (从 SCK10 \uparrow)	t_{KSI2}		50			ns
从 SCK10 \downarrow 至 SO10 输出的延迟时间	t_{KSO2}	$C = 50\text{ pF}^{\#}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		120	ns
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		120	ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		165	ns

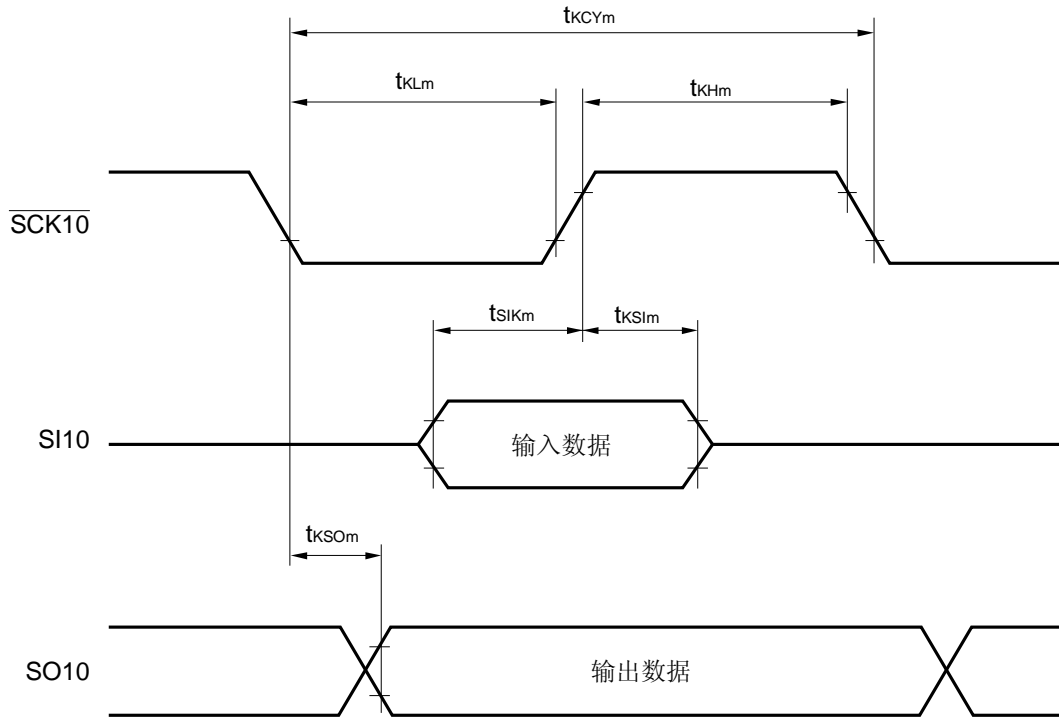
- 注 C 是 SO10 输出线路的负载电容。

串行传输时序

IIC0:



CSI10:



备注 $m = 1, 2$

A/D 转换器特性

($T_A = -40$ 至 $+85^\circ\text{C}$, $2.3\text{ V} \leq AV_{REF} \leq V_{DD} \leq 5.5\text{ V}$ ^{注1}, $V_{SS} = AV_{SS} = 0\text{ V}$)

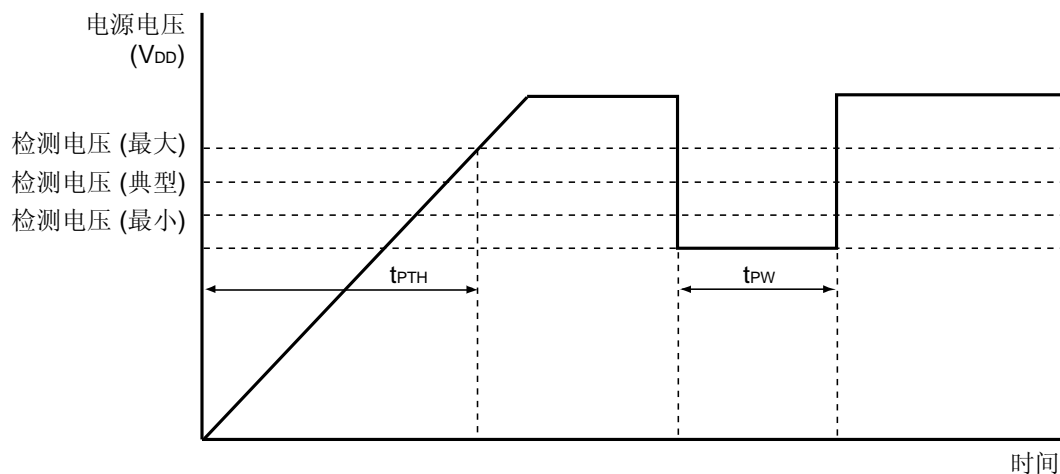
参数	符号	条件	最小.	典型值.	最大.	单位
分辨率	RES				10	位
总误差 ^{注2, 3}	AINL	$4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$			± 0.4	%FSR
		$2.7\text{ V} \leq AV_{REF} < 4.0\text{ V}$			± 0.6	%FSR
		$2.3\text{ V} \leq AV_{REF} < 2.7\text{ V}$			± 1.2	%FSR
转换时间	tCONV	$4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$	6.1		66.6	μs
		$2.7\text{ V} \leq AV_{REF} < 4.0\text{ V}$	12.2		66.6	μs
		$2.3\text{ V} \leq AV_{REF} < 2.7\text{ V}$	27		66.6	μs
零度误差 ^{注2, 3}	EzS	$4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$			± 0.4	%FSR
		$2.7\text{ V} \leq AV_{REF} < 4.0\text{ V}$			± 0.6	%FSR
		$2.3\text{ V} \leq AV_{REF} < 2.7\text{ V}$			± 0.6	%FSR
满度误差 ^{注2, 3}	EFS	$4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$			± 0.4	%FSR
		$2.7\text{ V} \leq AV_{REF} < 4.0\text{ V}$			± 0.6	%FSR
		$2.3\text{ V} \leq AV_{REF} < 2.7\text{ V}$			± 0.6	%FSR
积分非线性误差 ^{注2}	ILE	$4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$			± 2.5	LSB
		$2.7\text{ V} \leq AV_{REF} < 4.0\text{ V}$			± 4.5	LSB
		$2.3\text{ V} \leq AV_{REF} < 2.7\text{ V}$			± 6.5	LSB
微分非线性误差 ^{注2}	DLE	$4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$			± 1.5	LSB
		$2.7\text{ V} \leq AV_{REF} < 4.0\text{ V}$			± 2.0	LSB
		$2.3\text{ V} \leq AV_{REF} < 2.7\text{ V}$			± 2.0	LSB
模拟输入电压	VAIN		AVSS		AVREF	V

- 注 1. 未使用恒流驱动器时: $2.3\text{ V} \leq AV_{REF} \leq V_{DD} \leq 5.5\text{ V}$
 使用恒流驱动器时: $2.3\text{ V} \leq AV_{REF} \leq V_{DD} \leq 5.5\text{ V}$, $4.5\text{ V} \leq V_{DD} = CV_{DD} \leq 5.5\text{ V}$
2. 不包括量化误差 ($\pm 1/2$ LSB)。
3. 该值按满度值的比率 (%FSR) 来表示。

1.59 V POC 电路特性 ($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{SS} = 0\text{ V}$)

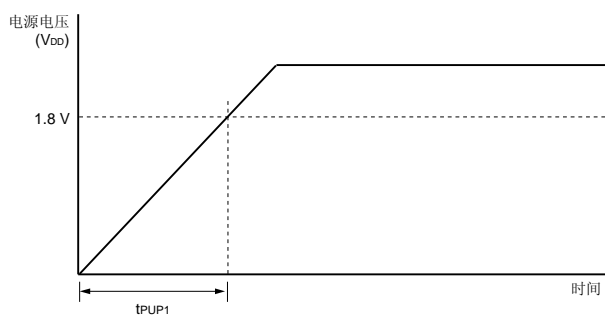
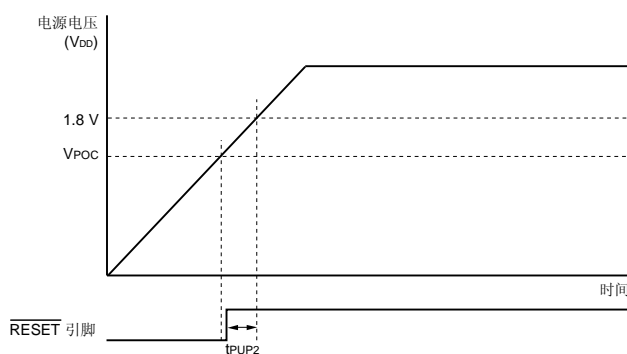
参数	符号	条件	最小.	典型值.	最大.	单位
检测电压	V_{POC}		1.44	1.59	1.74	V
电源电压上升斜度	t_{PTH}	$V_{DD}: 0\text{ V} \rightarrow V_{POC}$ 的变化斜度	0.5			V/ms
最小脉冲宽度	t_{PW}		200			μs

POC 电路时序

电源电压上升时间 ($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{SS} = 0\text{ V}$)

参数	符号	条件	最小.	典型值.	最大.	单位
上升至 1.8 V 的最大时间 (V_{DD} (MIN.)) ($V_{DD}: 0\text{ V} \rightarrow 1.8\text{ V}$)	t_{PUP1}	POCMODE (选项字节) = 0, 未使用 $\overline{\text{RESET}}$ 输入时			3.6	ms
上升至 1.8 V 的最大时间 (V_{DD} (MIN.)) (释放 $\overline{\text{RESET}}$ 输入 $\rightarrow V_{DD}: 1.8\text{ V}$)	t_{PUP2}	POCMODE (选项字节) = 0, 未使用 $\overline{\text{RESET}}$ 输入时			1.9	ms

电源电压上升时间的时序

• 未使用 $\overline{\text{RESET}}$ 引脚输入时• 使用 $\overline{\text{RESET}}$ 引脚输入时

2.7 V POC 电路特性 ($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{SS} = 0\text{ V}$)

参数	符号	条件	最小	典型值	最大	单位
电源电压应用的检测电压	V_{DDPOC}	POCMODE (选项字节) = 1	2.50	2.70	2.90	V

备注 根据 POCMODE (选项字节) 的设置, POC 电路的操作如下所述。

选项字节设置	POC 模式	操作
POCMODE = 0	1.59 V 模式操作	电源打开后复位状态将会保持直到达到 $V_{POC} = 1.59\text{ V}$ (TYP.), 当超过 V_{POC} 时将会释放复位。在这之后, 将会在与电源打开时相同的 V_{POC} 处执行 POC 检测。 当 POCMODE 为 0 时, 电源电压必须在 t_{PUP1} 或 t_{PUP2} 时上升。
POCMODE = 1	2.7 V/1.59 V 模式操作	电源打开后会保持复位状态, 直到达到 $V_{DDPOC} = 2.7\text{ V}$ (TYP.), 当超过 V_{DDPOC} 时复位将会释放。在这之后, 将会在 $V_{POC} = 1.59\text{ V}$ (TYP.) 时执行 POC 检测, 而不是 $V_{DDPOC} = 1.59\text{ V}$ 时。 当电压上升比 t_{PTH} 缓和时, 从电源打开到电压到 1.8V 时推荐使用 2.7 V/1.59 V POC 模式。

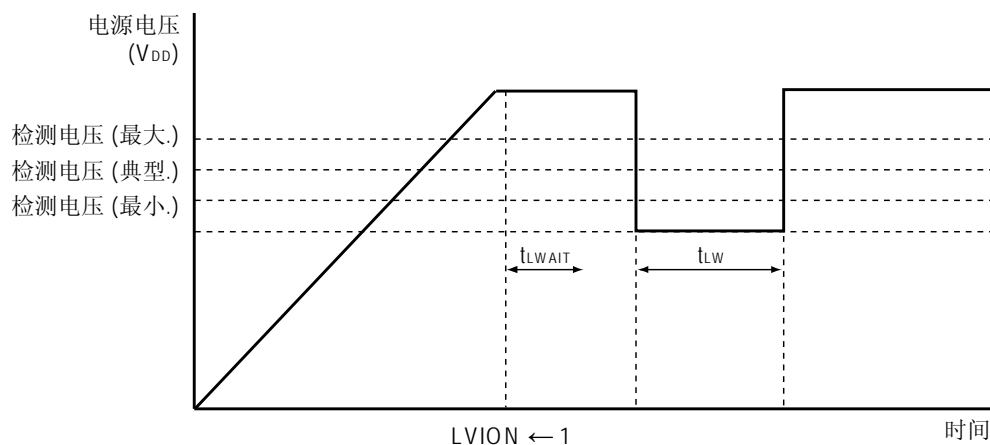
LVI 电路特性 ($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{POC} \leq V_{DD} \leq 5.5\text{ V}$, $AV_{REF} \leq V_{DD}$, $V_{SS} = 0\text{ V}$)

参数	符号	条件	最小.	典型值.	最大.	单位	
检测电压	电源电压电平	V_{LV10}		4.14	4.24	4.34	V
		V_{LV11}		3.99	4.09	4.19	V
		V_{LV12}		3.83	3.93	4.03	V
		V_{LV13}		3.68	3.78	3.88	V
		V_{LV14}		3.52	3.62	3.72	V
		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_{LV10}		2.60	2.70	2.80	V
		V_{LV11}		2.45	2.55	2.65	V
		V_{LV12}		2.29	2.39	2.49	V
		V_{LV13}		2.14	2.24	2.34	V
		V_{LV14}		1.98	2.08	2.18	V
		V_{LV15}		1.83	1.93	2.03	V
	外部输入引脚 ^{注1}	$EXLVI < V_{DD}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	1.11	1.21	1.31	V	
最小脉冲宽度	t_{LW}		200			μs	
工作稳定等待时间 ^{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 $V_{LV1(n-1)} > V_{LV1n}$: $n = 1$ 至 15

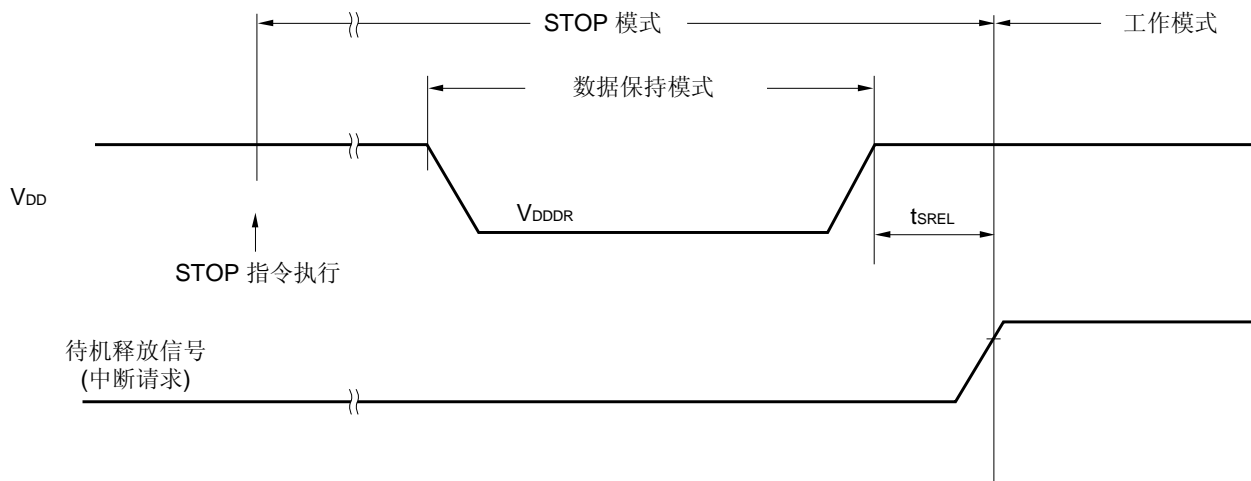
LVI 电路时序



数据寄存器 STOP 模式低电源电压数据保持特性 (TA = -40 至 +85°C)

参数	符号	条件	最小.	典型值.	最大.	单位
数据保持电源电压	V _{DDDR}		1.44 [#]		5.5	V

注 该值根据 POC 检测电压而变化。当电压下降时，在 POC 复位有效前，一直保持数据，但是当 POC 复位有效时，不能保持数据。



Flash 编程特性

($T_A = -40$ 至 $+85^\circ\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $AV_{REF} \leq V_{DD}$, $V_{SS} = AV_{SS} = 0\text{ V}$)

• 基本特性

参数		符号	条件		最小.	典型值.	最大.	单位		
V _{DD} 电源电流		I _{DD}	f _{XP} = 10 MHz (TYP.), 20 MHz (MAX.)			4.5	11.0	mA		
擦除时间 ^{注1, 2}	所有数据块	T _{eraca}				20	200	ms		
	数据块单元	T _{erasa}				20	200	ms		
写入时间 (以 8 位为单位) ^{注1}		T _{wrwa}				10	100	μs		
每个芯片的重写次数		C _{erwr}	1 次擦除 + 擦除后 1 次写入 = 1 次重写 ^{注3}	• 使用 flash 寄存器程序器时, 且使用由 NEC Electronics 提供的库 ^{注4} 时	• 用于程序更新	保持时间: 15 年	1000		次	
				• 使用由 NEC Electronics 提供的 EEPROM 模拟库 ^{注5}	• 且可重写 ROM 容量为 4KB 时	• 用于数据更新	保持时间: 5 年	10000		次
				上述条件以外的条件 ^{注6}		保持时间: 10 年	100		次	

- 注
1. flash 寄存器的特性。关于专用 flash 编程器 PG-FP4 或 PG-FP5 在自编程过程中使用并重写时间时的特性, 请参见表 22-11 和 22-12。
 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写入时, “擦除 → 写入” 和 “只写入” 都认为一次重写。
 4. 不包括由 78K0/Kx2 Flash 存储器自编程用户手册 (文档编号: U17516E) 指定的样本库。
 5. 不包括由 78K0/Kx2 EEPROM 模拟应用笔记 (文档编号: U17517E) 指定的样本程序。
 6. 当使用由 78K0/Kx2 Flash 存储器自编程用户手册 (文档编号: U17516E) 指定的样本库以及由 78K0/Kx2 EEPROM 模拟应用笔记 (文档编号: U17517E) 指定的样本程序时包含这类条件。

- 备注
1. f_{XP}: 主系统时钟振荡频率
 2. 关于串行写入操作特性, 请参照 78K0/Kx2 Flash 存储器编程 (编程器) 应用笔记 (U17739E)。

24.2 带有可选降压或升压转换器的恒流驱动器部分

最大绝对额定值 ($T_A = 25^\circ\text{C}$)

参数	符号	条件	额定值	单位
电源电压	CV_{DD}		-0.3 至 +6.0	V
输入电压	V_{IN}		-0.3 至 +42	V
电流检测电压	V_{SENSA}	SENSA0 至 SENSA3	升压模式时 -0.3 至 +42	V
			降压模式时 $\text{V}_{\text{IN}} - 5$ 至 V_{IN}	V
	V_{SENSB}	SENSB0 至 SENSB3	-0.3 至 CV_{DD}	V
栅极驱动输出电压	V_{DRVOUT}		-0.3 至 CV_{DD}	V
逻辑输入电压	V_i	EN, PWMn ($n = 0$ 至 3)	-0.3 至 CV_{DD}	V
栅极驱动峰值电流 (用来驱动外部电源 MOSFET)	$\text{I}_{\text{DRVPEAK}}$	$f_{\text{CHOP}} = 1 \text{ MHz}$, 脉冲宽度 = 10 ns	700	mA
存储温度	T_{ST}		-55 至 +150	$^\circ\text{C}$
结点温度	T_J		150	$^\circ\text{C}$

推荐工作条件 ($T_A = 25^\circ\text{C}$)

参数	符号	条件	最小.	典型值.	最大.	单位
电源电压	CV_{DD}		4.5	5.0	5.5	V
输入电压	V_{IN}	降压模式时 ($\text{V}_{\text{IN}} > \text{V}_{\text{OUT}}$)	9		38	V
		升压模式时 ($\text{V}_{\text{IN}} < \text{V}_{\text{OUT}}$)	9		28	V
输出电压	V_{OUT}	升压模式时 ($\text{V}_{\text{IN}} < \text{V}_{\text{OUT}}$)			37	V
PWMn ($n = 0$ 至 3) 输入 频率	f_{PWM}	PWM 输入的比率 = 50 %			500	Hz
PWM 输入的比率 ^注	DP_{PWM}		0		100	%
工作温度	T_{OP}		-40		85	$^\circ\text{C}$
结点温度	T_J		-40		125	$^\circ\text{C}$
PWM 等待时间	t_{WAIT}	EN 引脚上升到 PWM 输入的等待时间	100			s
栅极驱动器平均输出电流	I_{DRV}	$\text{C}_{\text{LOAD}} = 1000 \text{ pF}$		60		mA

注 建议不要使用 PWMn 输出的 0% 或 100%附近的占空比。

注意事项 确保按以下顺序开关电源。

- 打开电源: $\text{CV}_{\text{DD}} \rightarrow \text{V}_{\text{IN}}$
- 关闭电源: $\text{V}_{\text{IN}} \rightarrow \text{CV}_{\text{DD}}$

备注 当以串连形式连接的负载为 LED 时, 推荐的最多的负载个数为 7 个。

电气特性 (1/2)

(如果没有指定其他内容, 那么“降压操作过程中, $V_{IN} = 30\text{ V}$, $C_{VDD} = 5\text{ V}$, $T_A = 25^\circ\text{C}$, 且没有外部电源 MOSFET”)

• 常规

参数	符号	条件		最小.	典型值.	最大.	单位
工作消耗电流	I _{OPCVDD}	C _{VDD}	EN = 高电平, PWMn = 高电平		3.3	7	mA
	I _{OPVIN}	V _{IN}				250	A
待机消耗电流	I _{STBY1}	C _{VDD}	EN = 低电平, MODE = 低电平 (升压模式时)			10	A
	I _{STBY2}		EN = 低电平, MODE = 高电平 (降压模式时)			60	A

• 栅极驱动器转换部分

参数	符号	条件		最小.	典型值.	最大.	单位
驱动器打开时的电阻	R _{ON (SOURCE)}	I _{SOURCE} = 100 mA			7	12	Ω
	R _{ON (SINK)}	I _{SINK} = 100 mA			7	12	Ω
栅极电压上升/下降时间	t _r , t _f	C _{LOAD} = 1000 pF			20	50	ns

• 保护部分

参数	符号	条件		最小.	典型值.	最大.	单位
UVLO 工作电压	V _{LUVLO}	低阈值			3.3		V
	V _{HYSUVLO}	电磁宽			0.3		V
SH 高电平输出电压	V _{SHH}	I _{OUT} = 10 mA		0.8C _{VDD}		C _{VDD}	V
SH 低电平输出电压	V _{SHL}	I _{OUT} = -10 mA		0		0.2C _{VDD}	V
SENSB 过电流保护操作阈值电压	V _{TSENSB}	模式 = 低电平 (升压模式时)		0.315	0.35	0.385	V
过压保护电压	V _{OVP}	模式 = 低电平 (升压模式时)		38	39.5	41	V

• 逻辑控制部分

参数	符号	条件		最小.	典型值.	最大.	单位
高电平输入电压	V _{IH}	带有内部下拉的 EN 引脚 (200 kΩ)		0.7C _{VDD}		C _{VDD}	V
低电平输入电压	V _{IL}			0		0.3C _{VDD}	V
下拉电阻	R _{PD}			100	200	300	kΩ

- 注意事项 1. 当工作温度超过 150°C (TYP.) 时热保护电路开始工作。此时, 从 SH 引脚输出高电平。
2. 在升压模式中, 不会减小 8 μs 或小于 8 μs 时间宽度的浪涌电流。

电气特性 (2/2)

(如果没有指定其他内容, 那么“降压操作过程中, $V_{IN} = 30\text{ V}$, $CV_{DD} = 5\text{ V}$, $T_A = 25^\circ\text{C}$, 且没有外部电源 MOSFET”)

• 参考电压部分

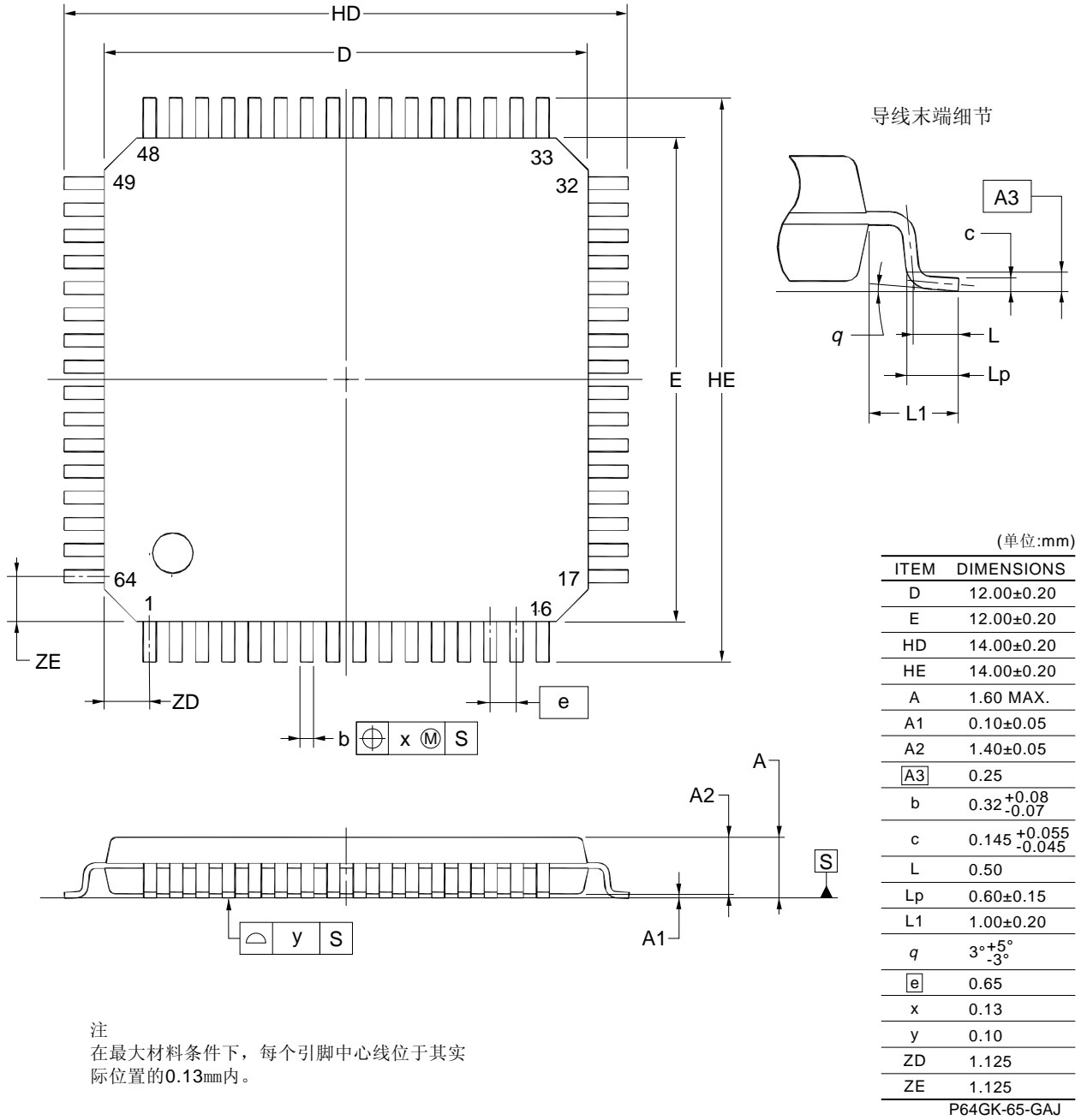
参数	符号	条件	最小.	典型值.	最大.	单位
参考电压	V_{REF}	CCAn 引脚和 CCBn 引脚短路	0.102	0.115	0.128	V
数字软启动时间	t_{SO}	EN = 高电平, PWMn = 低电平 → 高电平	32		128	μA

• 电流敏感放大器 (高压侧)

参数	符号	条件	MIN.	TYP.	MAX.	单位
输出电压	V_{TCSENS}	模式 = 高电平 (降压模式时), V_{IN} 至 $V_{OUT} = 0.115\text{ V}$, 通过 SENSBn 引脚检测	0.105	0.115	0.125	V

- 注意事项
1. 当工作温度超过 150°C (TYP.) 时热保护电路开始工作。此时, 从 SH 引脚输出高电平。
 2. 在升压模式中, 不会减小 $8\ \mu\text{s}$ 或小于 $8\ \mu\text{s}$ 时间宽度的浪涌电流。

64引脚塑封 LQFP(12x12)



第二十六章 推荐的焊接条件

这些产品应该在以下推荐条件下进行焊接和安装。

对于与以下所推荐不同的焊接方法和条件，请与 NEC Electronics 的销售代表联系。
技术信息请参见以下网站。

半导体设备安装手册 (<http://www.necel.com/pkg/en/mount/index.html>)。

表 26-1. 表面安装类型焊接条件

焊接方法	焊接条件	推荐条件符号
红外再流焊	封装峰值温度：260°C，时间：最大 60 分钟（大于或等于 220°C），此数：小于等于 3 次，暴露限值：7 天 ^注 （之后，在 125°C 下预烘干 10 至 72 小时）	IR60-107-3
局部加热	引脚温度：最高 350°C，时间：最长 3 秒（每个引脚列）	-

注 烘干之后，在允许的储藏期间，以 25°C 或低于 25°C 和 65% RH 或低于 65% RH 的条件储藏。

第二十七章 等待注意事项

27.1 等待注意事项

本产品具有两种内部系统总线。

一种是 CPU 总线，另一种是用于与低速外围硬件进行通信的外围总线。

因为 CPU 总线的时钟与外围总线时钟异步，因此如果访问 CPU 与访问外围硬件冲突，那么将会传输不期望的非法数据。

因此，当访问可能导致冲突的外围硬件时，CPU 将重复执行处理，直到传输正确数据为止。

因此，CPU 不会开始下一条指令处理而一直等待。如果这种情况发生，那么一条指令的执行时钟数将会增加等待时钟的个数（关于等待时钟的个数，参见表 27-1）。执行实时处理时必须注意这点。

27.2 产生等待的外围硬件

表 27-1 列出了通过 CPU 访问时会发出等待请求的寄存器。

表 27-1. 产生等待的寄存器以及 CPU 等待时钟的个数

外围硬件	寄存器	访问	等待时钟的个数
串行接口 UART0	ASIS0	读取	1 个时钟 (固定)
串行接口 UART6	ASIS6	读取	1 个时钟 (固定)
串行接口 IIC0	IICS0	读取	1 个时钟 (固定)
A/D 转换器	ADM	写入	1 至 5 个时钟 (选择 $f_{AD} = f_{PRS}/2$ 时)
	ADS	写入	1 至 7 个时钟 (选择 $f_{AD} = f_{PRS}/3$ 时)
	ADPC	写入	1 至 9 个时钟 (选择 $f_{AD} = f_{PRS}/4$ 时)
	ADCR	读取	2 至 13 个时钟 (选择 $f_{AD} = f_{PRS}/6$ 时) 2 至 17 个时钟 (选择 $f_{AD} = f_{PRS}/8$ 时) 2 至 25 个时钟 (选择 $f_{AD} = f_{PRS}/12$ 时)
<p>以上的时钟个数是为 f_{CPU} 和 f_{PRS} 选择同一源时钟时的个数。在以下条件下，可以通过以下表达式来进行计算等待时钟的个数。</p> <p><计算等待时钟的个数></p> <ul style="list-style-type: none"> 等待时钟的个数 = $\frac{2 f_{CPU}}{f_{AD}} + 1$ <p>* 如果等待时钟的个数 ≤ 0.5，那么小数部分将会被省略，而如果等待时钟的个数 > 0.5，那么将会对小数部分取整。</p> <p>f_{AD}: A/D 转换时钟频率 ($f_{PRS}/2$ 至 $f_{PRS}/12$)</p> <p>f_{CPU}: CPU 时钟频率</p> <p>f_{PRS}: 外围硬件时钟频率</p> <p>f_{XP}: 主系统时钟频率</p> <p><等待时钟的最大/最小个数的条件></p> <ul style="list-style-type: none"> 时钟的最多个数: CPU (f_{XP}) 的最高速度, A/D 转换时钟 ($f_{PRS}/12$) 的最低速度 时钟的最少数: CPU ($f_{XP}/16$) 的最小速度, A/D 转换时钟 ($f_{PRS}/2$) 的最高速度 			

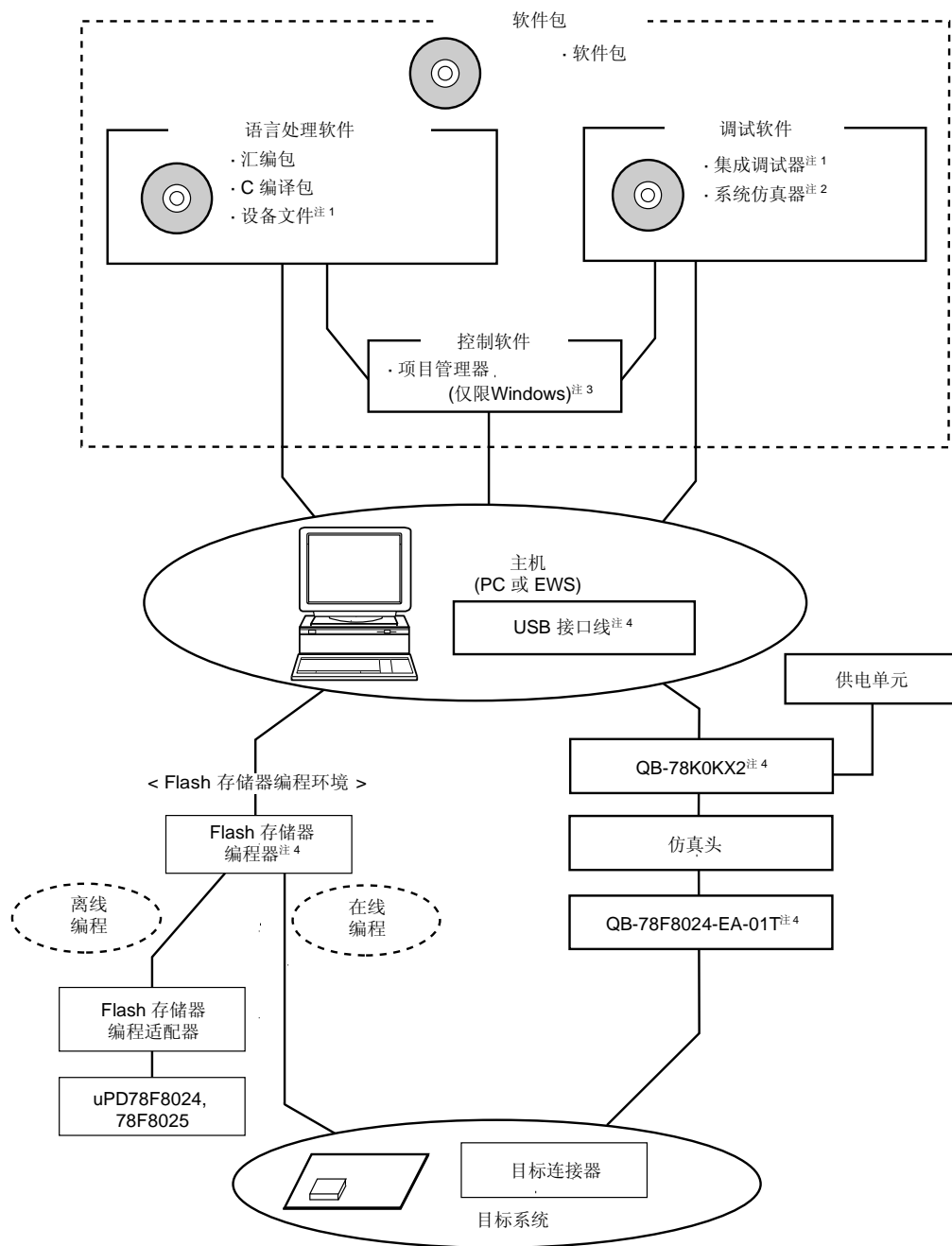
备注 时钟为 CPU 时钟 (f_{CPU})。

附录 A 开发工具

在使用 μ PD78F8024 和 μ PD78F8025 的系统开发中可使用如下开发工具。

图 A-1 所示为开发工具的组成。

图 A-1. 开发工具的组成



- 注
1. 从下载站点下载 μ PD78F8024 和 78F8025 (DF788024) 的设备文件及配合开发工具使用的集成调试器 ID78K0-QB。
(<http://www.necel.com/micro/en/ods/index.html>)。
 2. 软件包中包含了 SM+ for 78K0 (指令仿真版本)。不包含 SM+ for 78K0/Kx2 (指令 + 外围仿真版本)。
 3. 汇编程序包中包含了项目管理器 PM+。
PM+ 仅用于 WindowsTM。
 4. QB-78K0KX2 提供了集成调试器 ID78K0-QB, 一根 USB 接口线缆, 带有编程功能的片上调试仿真器 QB-MINI2。其它产品都需单独购买。
调试 μ PD78F8024 和 78F8025 软件, 需要 QB-78F8024-EA-01T (单独购买)。

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 微控制器的开发工具（软件）。
------------------------	--------------------------------

A.2 语言处理软件

RA78K0 ^{注1} 汇编程序包	<p>该汇编器将以助记符编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（DF788024）（需另外购买）结合使用。</p> <p><在 PC 环境中使用 RA78K0 时需要提前注意的地方></p> <p>该汇编程序包是基于 DOS 的应用程序。然而，它也可以通过使用基于 Windows 的项目管理器（PM+），在 Windows 中使用。PM+ 包含在汇编程序包中。</p>
CC78K0 ^{注1} C 语言编译程序包	<p>该编译器将以 C 语言编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 时需要提前注意的地方></p> <p>该汇编程序包是基于 DOS 的应用程序。然而，它也可以通过基于 Windows 的项目管理器（PM+）在 Windows 中使用。PM+ 包含在汇编程序包中。</p>
DF788024 ^{注2} 设备文件	<p>该文件包含设备特有的信息。</p> <p>该设备文件应该同工具（RA78K0， CC78K0， SM+ for 78K0/Kx2， ID78K0-QB 以及系统仿真器）（均需另外购买）结合使用。</p> <p>相应的 OS 和主机根据使用工具的不同而不同。</p>

- 注
1. 如果 RA78K0 和 CC78K0 的版本是 Ver.4.00 或者更新的版本，不同版本的 RA78K0 和 CC78K0 可以安装在同一台机器上。
 2. DF788024 可以和 RA78K0， CC78K0， SM+ for 78K0/Kx2， 以及 ID78K0-QB 一样来进行使用。从开发工具的下载站点中下载 DF788024（<http://www.necel.com/micro/en/ods/index.html>）。

A.3 Flash存储器编程工具

A.3.1 使用flash存储器编程器PG-FP5, FL-PR5, PG-FP4, 和 FL-PR4 时

PG-FP5, FL-PR5, PG-FP4 ^注 , FL-PR4 Flash 存储器编程器	用于带片上 Flash 存储器的微控制器的 Flash 存储器编程器。
FA-64GK-GAJ-B Flash 存储器编程适配器	用于连接 Flash 存储器编程器的 Flash 存储器编程适配器。

注 逐步淘汰

- 备注**
1. FL-PR5, FL-PR4, 和 FA-64GK-GAJ-B 是 Naito Densei Machida Mfg.Co., Ltd 的产品。
(<http://www.ndk-m.co.jp/>, TEL: +81-42-750-4172)。
 2. 使用 flash 存储器编程适配器的最新版本。

A.3.2 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	用于带有片上 flash 存储器的微控制器的简易 flash 存储器编程器。该仿真器应与连接线缆（16 引脚线缆）以及用于连接主机的 USB 接口线缆结合使用。
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 提供有一根 USB 接口线缆, MINI2 连接线缆（10 引脚和 16 引脚线缆）以及 78K0-OCD 板。编程时不使用连接线缆（10 引脚线缆）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/en/ods/index.html>)。

A.4 调试工具（硬件）

QB-78K0KX2 在线仿真器	当使用 μ PD78F8024 和 78F8025 产品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，通过 USB 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-78F8024-EA-01T 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。
QB-64GK-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。
QB-64GK-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。
QB-64GK-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。
QB-64GK-NQ-01T 目标连接器	该目标连接器用于安装在目标系统上。

备注 1. QB-78K0KX2 提供了集成调试器 ID78K0-QB，一根 USB 接口线缆，带有编程功能的片上调试仿真器 QB-MINI2。其它产品需单独购买。

调试 μ PD78F8024 和 78F8025 软件，需使用 QB-78F8024-EA-01T（单独购买）。

2. 封装内容随产品型号而变化，如下所示。

封装内容 产品型号	在线仿真器	仿真探头	交换适配器	YQ 连接器	目标连接器
QB-78K0KX2-T64GK	QB-78K0KX2	QB-80-EP-01T	QB-64GK-EA-04T ^注	QB-64GK-YQ-01T	QB-64GK-NQ-01T
QB-78F8024-EA-01T	-	-	QB-78F8024-EA-01T ^注	-	-

注：调试 μ PD78F8024 及 μ PD78F8025 的软件，使用交换适配器时选择 QB-78F8024-EA-01T 而不是 QB-64GK-EA-04T。

A.5 调试工具（软件）

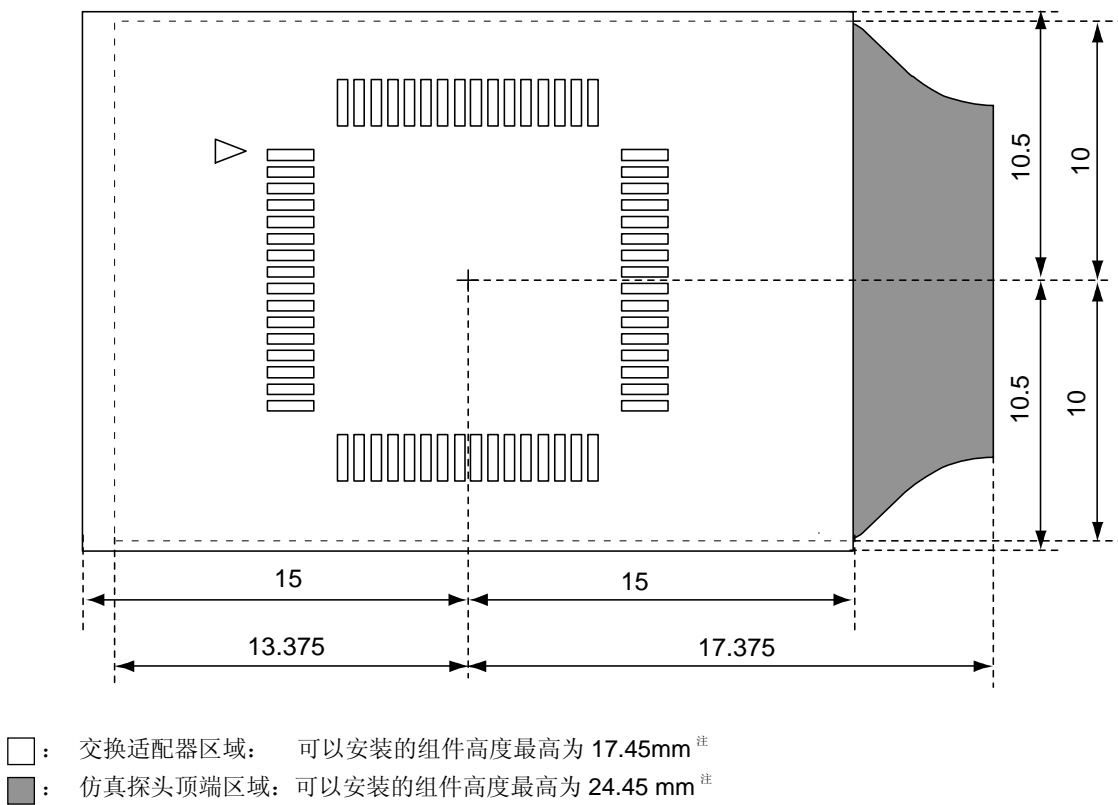
<p>ID78K0-QB 集成调试器</p>	<p>该调试器支持 78K0 系列微控制器的在线仿真器。ID78K0-QB 是基于 Windows 的软件。它改善了 C 语言兼容的调试功能，并且，通过使用结合了带有跟踪结果的源程序显示，反汇编显示以及存储器显示的集成窗口功能，能够以源程序显示追踪结果。它应该同设备文件（DF788024）一起使用。</p>
<p>SM+ for 78K0 SM+ for 78K0/Kx2 系统仿真器</p>	<p>系统仿真器是基于 Windows 的软件。</p> <p>在主机上模拟目标系统的操作时，该仿真器用于执行 C 源程序级或汇编程序级的调试操作。</p> <p>使用系统仿真器可以在不依赖硬件的基础上进行应用逻辑测试和性能测试，提高了开发效率和软件质量。</p> <p>系统仿真器应该同设备文件（DF788024）结合使用。</p> <p>下列两种系统仿真器可以支持 μPD78F8024 和 78F8025。</p> <ul style="list-style-type: none"> • SM+ for 78K0（指令仿真版本） 能够仿真 CPU。包含在软件包中。 • SM+ for 78K0/Kx2（指令 + 外围仿真版本） 能够仿真 CPU 和外围硬件（端口，定时器，串口等）。与软件包分开售卖。

注 从下载站点为开发工具下载 ID78K0-QB
（<http://www.necel.com/micro/en/ods/index.html>）。

附录B 目标系统设置时的注意事项

本章节说明了在使用 QB-78K0KX2 时目标系统上禁止安装组件的区域以及对组件安装高度有限的区域。

图 B-1. 对于 64 引脚 GK 封装



注 通过使用空间适配器可以调节高度（每个增加 2.4 mm）。

附录 C 寄存器索引

C.1 寄存器索引 (按字母排列寄存器名称)

[A]

A/D 转换模式寄存器 (ADM).....	252
A/D 端口配置寄存器 (ADPC).....	95, 259
模拟输入通道指定寄存器 (ADS).....	258
异步串行接口控制寄存器 6 (ASICL6).....	306
异步串行接口工作模式寄存器 0 (ASIM0).....	276
异步串行接口工作模式寄存器 6 (ASIM6).....	300
异步串行接口接收错误状态寄存器 0 (ASIS0).....	278
异步串行接口接收错误状态寄存器 6 (ASIS6).....	302
异步串行接口发送状态寄存器 6 (ASIF6).....	303

[B]

波特率发生控制寄存器 0 (BRGC0).....	279
波特率发生控制寄存器 6 (BRGC6).....	305

[C]

捕获/比较控制寄存器 00 (CRC00).....	138
时钟工作模式选择寄存器 (OSCCTL).....	103
时钟选择寄存器 6 (CKSR6).....	303

[E]

8 位 A/D 转换结果寄存器 (ADCRH).....	257
8 位定时器比较寄存器 50 (CR50).....	200
8 位定时器比较寄存器 51 (CR51).....	200
8 位定时器计数器 50 (TM50).....	200
8 位定时器计数器 51 (TM51).....	200
8 位定时器 H 载波控制寄存器 1 (TMCYC1).....	224
8 位定时器 H 比较寄存器 00 (CMP00).....	219
8 位定时器 H 比较寄存器 01 (CMP01).....	219
8 位定时器 H 比较寄存器 10 (CMP10).....	219
8 位定时器 H 比较寄存器 11 (CMP11).....	219
8 位定时器 H 模式寄存器 0 (TMHMD0).....	220
8 位定时器 H 模式寄存器 1 (TMHMD1).....	220
8 位定时器模式控制寄存器 50 (TMC50).....	204
8 位定时器模式控制寄存器 51 (TMC51).....	204
外部中断下降沿使能寄存器 (EGN).....	428
外部中断上升沿使能寄存器 (EGP).....	428

[I]

IIC 时钟选择寄存器 0 (IICCL0).....	363
IIC 控制寄存器 0 (IICCO).....	354
IIC 标志寄存器 0 (IICF0).....	361
IIC 功能扩展寄存器 0 (IICX0).....	364

IIC 移位寄存器 0 (IIC0)	351
IIC 状态寄存器 0 (IICS0)	359
输入切换控制寄存器 (ISC)	308
内部存储器容量切换寄存器 (IMS)	485
内部振荡模式寄存器 (RCM)	106
中断屏蔽标志寄存器 0H (MK0H)	426
中断屏蔽标志寄存器 0L (MK0L)	426
中断屏蔽标志寄存器 1H (MK1H)	426
中断屏蔽标志寄存器 1L (MK1L)	426
中断请求标志寄存器 0H (IF0H)	424
中断请求标志寄存器 0L (IF0L)	424
中断请求标志寄存器 1H (IF1H)	424
中断请求标志寄存器 1L (IF1L)	424
[L]	
低电压检测电平选择寄存器 (LVIS)	468
低电压检测寄存器 (LVIM)	466
[M]	
主时钟模式寄存器 (MCM)	108
主 OSC 控制寄存器 (MOC)	107
[O]	
振荡稳定时间计数状态寄存器 (OSTC)	109, 438
振荡稳定时间选择寄存器 (OSTS)	110, 439
[P]	
端口模式寄存器 0 (PM0)	92, 143
端口模式寄存器 1 (PM1)	92, 206, 225, 280, 308, 336
端口模式寄存器 2 (PM2)	92, 260
端口模式寄存器 3 (PM3)	92, 206
端口模式寄存器 6 (PM6)	92, 366
端口模式寄存器 12 (PM12)	92, 469
端口寄存器 0 (P0)	93
端口寄存器 1 (P1)	93
端口寄存器 2 (P2)	93
端口寄存器 3 (P3)	93
端口寄存器 6 (P6)	93
端口寄存器 12 (P12)	93
预分频模式寄存器 00 (PRM00)	141
优先级指定标志寄存器 0H (PR0H)	427
优先级指定标志寄存器 0L (PR0L)	427
优先级指定标志寄存器 1H (PR1H)	427
优先级指定标志寄存器 1L (PR1L)	427
预处理时钟控制寄存器 (PCC)	105
上拉电阻选择寄存器 0 (PU0)	94
上拉电阻选择寄存器 1 (PU1)	94
上拉电阻选择寄存器 3 (PU3)	94

上拉电阻选择寄存器 12 (PU12).....	94
[R]	
接收缓存寄存器 0 (RXB0)	275
接收缓存寄存器 6 (RXB6)	299
接收移位寄存器 0 (RXS0)	275
接收移位寄存器 6 (RXS6)	299
复位控制标志寄存器 (RESF).....	458
[S]	
串行时钟选择寄存器 10 (CSIC10).....	334
串行 I/O 移位寄存器 10 (SIO10).....	332
串行工作模式寄存器 10 (CSIM10).....	333
16 位定时器捕获/比较寄存器 000 (CR000).....	132
16 位定时器捕获/比较寄存器 010 (CR010).....	132
16 位定时器计数器 00 (TM00).....	132
16 位定时器模式控制寄存器 00 (TMC00).....	136
16 位定时器输出控制寄存器 00 (TOC00).....	139
从地址寄存器 0 (SVA0)	351
[T]	
定时器时钟选择寄存器 50 (TCL50)	201
定时器时钟选择寄存器 51 (TCL51)	201
10 位 A/D 转换结果寄存器 (ADCR)	256
发送缓存寄存器 10 (SOTB10).....	332
发送缓存寄存器 6 (TXB6).....	299
发送移位寄存器 0 (TXS0).....	275
发送移位寄存器 6 (TXS6).....	299
[W]	
看门狗定时器使能寄存器 (WDTE).....	244

C.2 寄存器索引 (按字母排列寄存器符号)

[A]

ADCR: 10 位 A/D 转换结果寄存器	256
ADCRH: 8 位 A/D 转换结果寄存器	257
ADM: A/D 转换模式寄存器	252
ADPC: A/D 端口配置寄存器	95, 259
ADS: 模拟输入通道指定寄存器	258
ASICL6: 异步串行接口控制寄存器 6	306
ASIF6: 异步串行接口发送状态寄存器 6	303
ASIM0: 异步串行接口工作模式寄存器 0	276
ASIM6: 异步串行接口工作模式寄存器 6	300
ASIS0: 异步串行接口接收错误状态寄存器 0	278
ASIS6: 异步串行接口接收错误状态寄存器 6	302

[B]

BRGC0: 波特率发生控制寄存器 0	279
BRGC6: 波特率发生控制寄存器 6	305

[C]

CKSR6: 时钟选择寄存器 6	303
CMP00: 8 位定时器 H 比较寄存器 00	219
CMP01: 8 位定时器 H 比较寄存器 01	219
CMP10: 8 位定时器 H 比较寄存器 10	219
CMP11: 8 位定时器 H 比较寄存器 11	219
CR000: 16 位定时器捕获/比较寄存器 000	132
CR010: 16 位定时器捕获/比较寄存器 010	132
CR50: 8 位定时器比较寄存器 50	200
CR51: 8 位定时器比较寄存器 51	200
CRC00: 捕获/比较控制寄存器 00	138
CSIC10: 串行时钟选择寄存器 10	334
CSIM10: 串行工作模式寄存器 10	333

[E]

EGN: 外部中断下降沿使能寄存器	428
EGP: 外部中断上升沿使能寄存器	428

[I]

IF0H: 中断请求标志寄存器 0H	424
IF0L: 中断请求标志寄存器 0L	424
IF1H: 中断请求标志寄存器 1H	424
IF1L: 中断请求标志寄存器 1L	424
IIC0: IIC 移位寄存器 0	351
IICC0: IIC 控制寄存器 0	354

IICCL0: IIC 时钟选择寄存器 0	363
IICF0: IIC 标志寄存器 0	361
IICS0: IIC 状态寄存器 0	359
IICX0: IIC 功能扩展寄存器 0	364
IMS: 内部存储器容量切换寄存器寄存器	485
ISC: 输入切换控制寄存器寄存器	308
[L]	
LVIM: 低电压检测寄存器	466
LVIS: 低电压检测电平选择寄存器	468
[M]	
MCM: 主时钟模式寄存器	108
MK0H: 中断屏蔽标志寄存器 0H	426
MK0L: 中断屏蔽标志寄存器 0L	426
MK1H: 中断屏蔽标志寄存器 1H	426
MK1L: 中断屏蔽标志寄存器 1L	426
MOC: 主 OSC 控制寄存器	107
[O]	
OSCCTL: 时钟操作模式选择寄存器	103
OSTC: 振荡稳定时间计数状态寄存器	109, 438
OSTS: 振荡稳定时间选择寄存器	110, 439
[P]	
P0: 端口寄存器 0	93
P1: 端口寄存器 1	93
P2: 端口寄存器 2	93
P3: 端口寄存器 3	93
P6: 端口寄存器 6	93
P12: 端口寄存器 12	93
PCC: 预处理时钟控制寄存器	105
PM0: 端口模式寄存器 0	92, 143
PM1: 端口模式寄存器 1	92, 206, 225, 280, 308, 336
PM2: 端口模式寄存器 2	92, 260
PM3: 端口模式寄存器 3	92, 206
PM6: 端口模式寄存器 6	92, 366
PM12: 端口模式寄存器 12	92, 469
PROH: 优先级指定标志寄存器 0H	427
PROL: 优先级指定标志寄存器 0L	427
PR1H: 优先级指定标志寄存器 1H	427
PR1L: 优先级指定标志寄存器 1L	427
PRM00: 预分频模式寄存器 00	141
PU0: 上拉电阻选择寄存器 0	94
PU1: 上拉电阻选择寄存器 1	94
PU3: 上拉电阻选择寄存器 3	94
PU12: 上拉电阻选择寄存器 12	94

[R]

RCM: 内部振荡模式寄存器	106
RESF: 复位控制标志寄存器	458
RXB0: 接收缓存寄存器 0	275
RXB6: 接收缓存寄存器 6	299
RXS0: 接收移位寄存器 0	275
RXS6: 接收移位寄存器 6	299

[S]

SIO10: 串行 I/O 移位寄存器 10	332
SOTB10: 发送缓存寄存器 10	332
SVA0: 从地址寄存器 0	351

[T]

TCL50: 定时器时钟选择寄存器 50	201
TCL51: 定时器时钟选择寄存器 51	201
TM00: 16 位定时器计数器 00	132
TM50: 8 位定时器计数器 50	200
TM51: 8 位定时器计数器 51	200
TMC00: 16 位定时器模式控制寄存器 00	136
TMC50: 8 位定时器模式控制寄存器 50	204
TMC51: 8 位定时器模式控制寄存器 51	204
TMCYC1: 8 位定时器 H 载波模式寄存器 1	224
TMHMD0: 8 位定时器 H 模式寄存器 0	220
TMHMD1: 8 位定时器 H 模式寄存器 1	220
TOC00: 16 位定时器输出控制寄存器 00	139
TXB6: 发送缓存寄存器 6	299
TXS0: 发送移位寄存器 0	275
TXS6: 发送移位寄存器 6	299

[W]

WDTE: 看门狗定时器使能寄存器	244
-------------------------	-----

附录 D 修改历史

D.1 本版本的主要修改内容

(1/2)

页	描述
全篇	增加 μ PD78F8025 增加 SM+ for 78K0
第一章 概述	
p. 22	修改 1.7 功能概述 中的主要系统时钟（振荡频率）
第四章 CPU 结构	
p. 52	修改 4.2.1 (2) 程序状态字 (PSW) 中的描述
第五章 端口功能	
p. 88	增加 图 5-14 P60 和 P61 的框图中的注意事项
第六章 时钟发生器	
p. 104	增加 图 6-2 时钟操作模式选择寄存器 (OSCCTL) 的格式中的注意事项 2
第七章 16 位定时器/事件计数器 00	
p. 142	在图 7-9 预分频模式寄存器 00 (PRM00) 的格式中修改注 1 和增加注 3
第八章 8 位定时器/事件计数器 50 和 51	
pp. 202, 203	在图 8-5. 定时器时钟选择寄存器 50 (TCL50) 的格式和图 8-6. 时钟选择寄存器 51 (TCL51) 的格式中修改注 1 和增加注 4
第九章 8 位定时器 H0 和 H1	
pp. 221, 223	在图 9-5. 8 位定时器 H 模式寄存器 0 (TMHMD0) 的格式和图 9-6. 8 位定时器 H 模式寄存器 1 (TMHMD1) 的格式中修改注 1 和增加注 3
第十章 看门狗定时器	
p. 242	修改 10.1 看门狗定时器的功能中的描述
p. 245	修改 10.4.1 看门狗定时器的控制操作中的描述
p. 248	修改 10.4.3 设置看门狗定时器的窗口打开周期中的备注
第十一章 A/D 转换器	
p. 254	修改表 11-2. A/D 转换时间选择中的注意事项 1
第十二章 串行接口 UART0	
p. 279	修改图 12-4. 波特率发生器控制寄存器 0 (BRGC0) 的格式中的注 1
p. 289	修改表 12-4. TPS01 和 TPS00 的设置值中的注 1
第十三章 串行接口 UART6	
p. 300	修改图 13-5. 异步串行接口操作模式寄存器 6 (ASIM6) 的格式 (1/2) 中的注 1
p. 304	在图 13-8. 时钟选择寄存器 6 (CKSR6) 的格式中修改注 1 和增加注 3
p. 309	修改 13.4.1 (1) 使用的寄存器中的注 1
p. 325	在表 13-4. TPS63 到 TPS60 的设置值中修改注 1 和增加注 3
第十四章 串行接口 CSI10	
p. 334	修改图 14-3. 串行时钟选择寄存器 10 (CSIC10) 的格式中的注 1
p. 338	在表 14-2. 寄存器设置和引脚之间的关系中修改并增加注 2
p. 347	修改 14.4.2 (5) SO10 输出
第十五章 串行接口 IIC0	
p. 365	修改表 15-2. 选择时钟设置中的注 1

页	描述
第十八章 复位功能	
p. 451	修改 图 18-1 复位功能框图
第十九章 上电清零电路	
p. 460	修改 图 19-1 上电清零电路的框图
第二十章 低压检测电路	
遍历	操作稳定时间 (10 μ s (最大)) \rightarrow 操作稳定时间 (10 μ s (最小))
p. 467	修改图 20-2. 低电压检测寄存器 (LVIM) 的格式中的注 4
第二十一章 选项字节	
p. 482	修改 21.1 (2) 0081H/1081H 中的注意事项
p. 484	修改 图 21-1 选项字节的格式 (2/2) 中的“地址: 0081H/1081H”中的注 1
第二十二章 FLASH 存储器	
p. 485	修改图 22-1. 内部存储器容量切换寄存器 (IMS) 的格式
第二十四章 电气特性	
p. 521	修改 X1 振荡器特性
p. 528	修改 AC 特性中的 (1) 基本操作中的如下项目 <ul style="list-style-type: none"> 指令周期 (最小指令执行时间) 外围硬件的时钟频率 外部主系统时钟频率 外部主系统时钟输入高电平宽度, 低电平宽度 增加 AC 特性中的 (1) 基本操作中的注 3
p. 529	修改 Tcy vs. V _{DD} (主系统时钟操作) 中的图
p. 531	修改 AC 特性中的 (1) 串行接口中的如下项目 <ul style="list-style-type: none"> (c) IIC0 中的数据保持时间 (发送)
p. 537	修改 LVI 电路特性中的操作稳定等待时间
p. 539	修改 Flash 存储器编程特性中的每个芯片的重写次数
附录 A 开发工具	
p. 547	全部章节
附录 D 修改历史	
p. 562	增加 D.2 前一个版本的修改历史

<R> D.2 前一个版本的修改历史

这是前一个版本的修改历史。文中的章节代表每一个版本的章节。

(1/2)

版本	描述	章节
第二版	工作环境温度 $-20 \sim +85^{\circ}\text{C} \rightarrow -40 \sim +85^{\circ}\text{C}$	遍历
	修改 1.3 订购信息	第一章 概述
	修改 3.1 带有可选降压或升压转换器的恒流驱动器	第三章 带有可选降压或升压转换器的恒流驱动器
	修改图 3-2. 降压模式的框图	
	修改备注 1	
	增加 5.2.2 端口 1 中的注意事项 2	第五章 端口功能
	修改图 7-41 PPG 输出操作的寄存器设置示例中的 (f) 16 位捕捉 / 比较寄存器 000 (CR000) 的内容	第七章 16 位定时器/事件计数器 00
	表 11-2. A/D 转换时间选择 • 修改 (2) $2.3\text{ V} \leq AV_{\text{REF}} < 5.5\text{ V}$ ($LV0 = 1$) • 修改 注意事项 1 中的 (2) $2.3\text{ V} \leq AV_{\text{REF}} < 5.5\text{ V}$ ($LV0 = 1$)	第十一章 A/D 转换器
	在图 13-10. 异步串行接口控制寄存器 6 (ASICL6) 的格式 (2 / 2) 中增加注意事项 8	第十三章 串行接口 UART6
	修改表 15-4. 主扩展码位定义并增加备注	第十五章 串行接口 IIC0
	修改图 15-27. 从主设备到从设备的通信示例和图 15-28. 从从设备到主设备的通信示例	
	修改图 17-3. 通过生成中断请求来释放 HALT 模式	第十七章 待机功能
	在表 17-3. STOP 模式中的运行状态中增加注意事项 5	
	修改图 17-5. STOP 模式被释放时的操作时序 (产生非屏蔽中断请求时) 中的注 2	
	修改图 17-6. 通过生成中断请求来释放 STOP 模式中的注	
	在图 20-2 低压检测寄存器的格式 (LVIM) 中增加注意事项 4	第二十章 低压检测电路
	20.4.1 (1) 当检测电源电压 (VDD) 的电平时和当检测从外部输入引脚 (EXLVI) 输入的电压的电平时 • 增加 <1> • 修改注意事项 1	
	20.5.1 当供电电压在 LVI 检测电压附近波动时 • 修改 <措施> (2) (b) • 删除旧版本中的图 20-9 复位释放后的软件处理示例	
	增加 20.5.2 使用 LVI 作为复位信号时的注意事项	
	修改图 22-1. 内部存储器容量切换寄存器 (IMS) 的格式	第二十二章 FLASH 存储器
	修改 22.7 安全设置中的注意事项	
	目标值 \rightarrow 保证值	第二十四章 电气特性

版本	描述	章节
第二版	封装图修改	第二十五章 封装图
	在表 26-1. 表面安装类型焊接条件中增加红外回流焊的焊接条件	第二十六章 推荐的焊接条件
	QB-78K0xxx → QB-78K0KX2, QB-64GK-EA-04T → QB-78F8024-EA-01T, FA-64GK-9ET-A → FA-64GK-GAJ-B	附录 A 开发工具
	修改 图 A-1 开发工具配置中的注 4	
	A.5 调试工具（硬件） • 修改 备注 1, 2 • 增加注	
增加章节	附录 D 修改历史	

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
四川省成都市二环路南三段 15 号
天华大厦 608 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