

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 PD78F0714

8 位单片微控制器

μ PD78F0714

文档编号. U16928CA2V0UD00 (第 2 版)

发行日期 2009 年 2 月 NS

© NEC Electronics Corporation 2009

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。

I

EEPROM 是 NEC Electronics Corporation 的商标。

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

PC/AT是International Business Machines Corporation的商标。

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

SPARCstation是SPARC International, Inc.的商标。

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

TRON 是 The Realtime Operating system Nucleus 的缩写。

ITRON 是 Industrial TRON 的缩写。

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

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

- 本档所登载的内容有效期截止至 2009 年 2 月，信息先于产品的生产周期发布。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。否则因本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），交通用信号控制设备，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

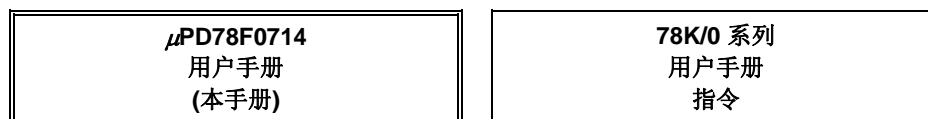
引言

读者 本手册旨在向希望了解 μ PD78F0714 微控制器的功能，设计，开发应用系统及用于这些设备的程序的工程师
目标产品如下所示

μ PD78F0714

目的 本手册旨在提供给用户功能描述的了解，如下面**组织**中所示

Organization μ PD78F0714 微控制器手册分成两个部分: 本手册和指令手册 (与 78K0 微控制器通用).



- 引脚功能
- 内部模块功能
- 中断
- 其片上外围功能
- 电气特性
- CPU 功能
- 指令设置
- 各指令扩展

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

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

规则

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

相关文档

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

相关设备文档

文档名称	文档编号
μPD78F0714 用户手册	本手册
78K/0 系列指令 用户手册	U12326E

<R> 开发工具 (软件)相关文档 (用户手册)

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

<R> 开发工具 (硬件) 相关文档 (用户手册)

文档名称	文档编号
QB-780714 片上仿真器	U17081E
QB-78K0MINI 片上调试仿真器	U17029E
QB-78K0MINI2 具有片上调试功能的仿真器	U18371E

与Flash存储编程相关的文档

文档名称	文档编号
PG-FP4 Flash 存储器编程器用户手册	U15260E

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 半导体设备的质量等级	C11531E
NEC 半导体设备可靠性/质量控制系统	C10983E
半导体设备防静电 ESD 指南	C11892E

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

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

目录

第一章 概述	16
1.1 特征	16
1.2 应用	17
1.3 订购信息	17
1.4 引脚配置（俯视图）	18
1.5 框图	20
1.6 功能概述	21
第二章 引脚功能	23
2.1 引脚功能列表	23
2.2 引脚功能说明	27
2.2.1 P00 到 P03（端口 0）	27
2.2.2 P10 到 P17（端口 1）	27
2.2.3 P20 到 P27（端口 2）	28
2.2.4 P30 到 P33（端口 3）	28
2.2.5 P40 到 P47（端口 4）	29
2.2.6 P50 到 P57（端口 5）	29
2.2.7 P64 到 P67（端口 6）	30
2.2.8 P70 到 P73（端口 7）	30
2.2.9 TW0TO0/RTP10 到 TW0TO5/RTP15	30
2.2.10 AVREF	30
2.2.11 AVSS.....	30
2.2.12 RESET	30
2.2.13 X1 和 X2.....	30
2.2.14 VDD 和 EVDD	30
2.2.15 VSS 和 EVSS.....	31
2.2.16 FLMD0	31
2.3 引脚 I/O 电路以及不使用引脚的推荐连接	32
第三章 CPU 架构	34
3.1 存储器映射	34
3.1.1 内部程序存储空间.....	35
3.1.2 内部数据存储空间.....	36
3.1.3 特殊功能寄存器（SFR）区域.....	36
3.1.4 数据存储器寻址	36
3.2 处理器寄存器	38
3.2.1 控制寄存器	38
3.2.2 通用寄存器	42
3.2.3 特殊功能寄存器（SFR）	43
3.3 指令地址寻址	49
3.3.1 相对寻址.....	49
3.3.2 立即寻址.....	50
3.3.3 表间接寻址	51

3.3.4	寄存器寻址	52
3.4	操作数地址寻址.....	53
3.4.1	隐含寻址.....	53
3.4.2	寄存器寻址	54
3.4.3	直接寻址.....	55
3.4.4	短直接寻址	56
3.4.5	特殊功能寄存器 (SFR) 寻址	57
3.4.6	寄存器间接寻址.....	58
3.4.7	基址寻址.....	59
3.4.8	基址变址寻址	60
3.4.9	堆栈寻址.....	61
第四章	端口功能.....	62
4.1	端口功能	62
4.2	端口配置	64
4.2.1	端口 0	65
4.2.2	端口 1	66
4.2.3	端口 2	70
4.2.4	端口 3	71
4.2.5	端口 4	73
4.2.6	端口 5	74
4.2.7	端口 6	76
4.2.8	端口 7	77
4.3	控制端口功能的寄存器.....	78
4.4	端口功能的操作.....	82
4.4.1	写入 I/O 端口	82
4.4.2	从 I/O 端口中读取	82
4.4.3	I/O 端口上的操作	82
4.5	端口寄存器 n (Pn) 1 位操作指令的注意事项.....	83
第五章	时钟发生器.....	84
5.1	时钟发生器的功能.....	84
5.2	时钟发生器的配置.....	84
5.3	控制时钟发生器的寄存器.....	86
5.4	系统时钟振荡器.....	93
5.4.1	X1 振荡器	93
5.4.2	错误的振荡器连接示例	94
5.4.3	内部振荡器	95
5.4.4	预分频器.....	95
5.5	时钟发生器的操作.....	96
5.6	内部振荡时钟和 X1 输入时钟之间切换需要的时间.....	101
5.7	CPU 时钟转换需要的时间.....	102
5.8	时钟切换流程图和寄存器设置.....	103
5.8.1	从内部振荡时钟切换为 X1 输入时钟	103
5.8.2	从 X1 输入时钟切换为内部振荡时钟	104
5.8.3	寄存器设置	105

第六章	10 位逆变器控制定时器	106
6.1	10 位逆变器控制定时器概述	106
6.2	10 位逆变器控制定时器的功能	106
6.3	10 位逆变器控制定时器的配置	106
6.4	控制 10 位逆变器控制定时器的寄存器	110
6.5	控制 10 位逆变器控制定时器的寄存器	115
第七章	16 位向上 / 向下计数器 ITENC20	122
7.1	16 位向上 / 向下计数器 ITENC20 的功能	122
7.2	16 位向上 / 向下计数器 ITENC20 的配置	124
	(2) 比较寄存器 0 (IT20CM0)	127
	(3) 比较寄存器 1 (IT20CM1)	128
	(4) 捕获 / 比较寄存器 0 (IT20CC0)	129
	(5) 捕获 / 比较寄存器 1 (IT20CC1)	130
7.3	16 位向上 / 向下计数器 ITENC20 的控制寄存器	131
	(1) 定时器单元模式寄存器 (IT20TUM)	132
	(2) 定时器控制寄存器 (IT20TMC)	133
	(3) 捕获 / 比较控制寄存器 (IT20CCR)	134
	(4) 有效沿选择寄存器 (IT20SESA)	135
	(5) 预分频器模式寄存器 (IT20PRM)	136
	(6) 状态寄存器 (IT20STS)	137
	(8) 端口模式寄存器 5 (PM5)	138
7.4	16 位向上 / 向下计数器 ITENC20 的操作	139
	7.4.1 基本操作	139
	7.4.2 通用定时器模式下的操作	140
	7.4.3 UDC 模式下的操作	143
7.5	16 位向上 / 向下计数器 ITENC20 的内部操作	149
	7.5.1 UDC 模式 B 下计数值的清除	149
	7.5.2 比较匹配发生时计数值的清除	150
	7.5.3 传输操作	150
	7.5.4 比较匹配时的中断信号输出	151
	7.5.5 IT20UBD 标志 (IT20STS 寄存器的第 0 位) 操作	151
第八章	16 位定时器 / 事件计数器 00	152
8.1	16 位定时器 / 事件计数器 00 的功能	152
8.2	16 位定时器 / 事件计数器 00 的配置	153
8.3	控制 16 位定时器 / 事件计数器 00 的寄存器	157
8.4	16 位定时器 / 事件计数器 00 的操作	163
	8.4.1 间隔定时器操作	163
	8.4.2 PPG 输出操作	166
	8.4.3 脉冲宽度测量操作	169
	8.4.4 外部事件计数器操作	177
	8.4.5 方波输出操作	180
	8.4.6 单脉冲输出操作	182
8.5	16 位定时器 / 事件计数器 00 的注意事项	187

第九章 9 位定时器 / 事件计数器 50 和 51	190
9.1 8 位定时器 / 事件计数器 50 和 51 的功能.....	190
9.2 8 位定时器 / 事件计数器 50 和 51 的配置.....	192
9.3 控制 8 位定时器 / 事件计数器 50 和 51 的寄存器.....	194
9.4 8 位定时器 / 事件计数器 50 和 51 的操作.....	199
9.4.1 作为间隔定时器工作.....	199
9.4.2 作为外部事件计数器工作.....	201
9.4.3 方波输出操作	202
9.4.4 PWM 输出操作.....	203
9.5 8 位定时器 / 事件计数器 50 和 51 的注意事项	207
第十章 8 位定时器 H0.....	208
10.1 8 位定时器 H0 的功能	208
10.2 8 位定时器 H0 的配置.....	208
10.3 控制 8 位定时器 H0 的寄存器	211
10.4 8 位定时器 H0 的操作	214
10.4.1 作为间隔定时器 / 方波输出操作.....	214
10.4.2 作为 PWM 输出模式操作.....	217
第十一章 看门狗定时器	223
11.1 看门狗定时器的功能	223
11.2 看门狗定时器的配置	225
11.3 控制看门狗定时器的寄存器	226
11.4 看门狗定时器的操作	228
11.4.1 当“内部振荡器不能被停止”由选项字节选择时的看门狗定时器操作	228
11.4.2 当“内部振荡器可以由软件停止”由选项字节选择时的看门狗定时器操作	229
11.4.3 STOP 模式下看门狗定时器的操作（当“内部振荡器可以由软件停止”由选项字节选择时）	230
11.4.4 HALT 模式下看门狗定时器的操作（当“内部振荡器可以由软件停止”由选项字节选择时）	232
第十二章 时钟输出/蜂鸣器输出控制器.....	233
12.1 时钟输出/蜂鸣器输出控制器的功能.....	233
12.2 时钟输出/蜂鸣器输出控制器的配置.....	234
12.3 控制时钟输出/蜂鸣器输出控制器的寄存器.....	234
12.4 时钟输出/蜂鸣器输出控制器的操作.....	237
12.4.1 时钟输出操作	237
12.4.2 作为蜂鸣器输出的操作	237
第十三章 实时输出端口	238
13.1 实时输出端口的功能	238
13.2 实时输出端口的配置	238
13.3 控制实时输出端口的寄存器	243
13.4 实时输出端口的操作	249
13.5 使用实时输出端口.....	259
13.6 实时输出端口的注意事项	260

第十四章	直流逆变器控制功能.....	261
第十五章	A/D 转换器	262
15.1	A/D 转换器的功能	262
15.2	A/D 转换器的配置	263
15.3	在 A/D 转换器中使用的寄存器	265
15.4	输入电压和 A/D 转换结果之间的关系	272
15.5	A/D 转换器的操作	273
15.5.1	A/D 转换器的基本操作	273
15.5.2	触发模式	275
15.5.3	操作模式	276
15.5.4	掉电监控功能	279
15.6	如何阅读 A/D 转换器特性表	283
15.7	A/D 转换器的注意事项	285
第十六章	串行接口 UART0	288
16.1	串行接口 UART0 的功能	288
16.2	串行接口 UART0 的配置	289
16.3	控制串行接口 UART0 的寄存器	292
16.4	串行接口 UART0 的操作	297
16.4.1	操作停止模式	297
16.4.2	异步串行接口 (UART) 模式	298
16.4.3	专用波特率产生器	304
第十七章	串行接口 CSI10	309
17.1	串行接口 CSI10 的功能	309
17.2	串行接口 CSI10 的配置	310
17.3	控制串行接口 CSI10 的寄存器	312
17.4	串行接口 CSI10 的操作	316
17.4.1	操作停止模式	316
17.4.2	3 线串行 I/O 模式	317
第十八章	乘法器/除法器	325
18.1	乘法器/除法器的功能	325
18.2	乘法器/除法器的配置	325
18.3	控制乘法器/除法器的寄存器	330
18.4	乘法器/除法器的操作	331
18.4.1	乘法运算	331
18.4.2	除法运算	333
第十九章	中断功能	335
19.1	中断功能类型	335
19.2	中断源和配置	335
19.3	控制中断功能的寄存器	339
19.4	中断服务操作	347

19.4.1	非屏蔽中断请求响应操作.....	347
19.4.2	可屏蔽中断请求响应.....	349
19.4.3	软件中断请求响应.....	351
19.4.4	多重中断服务.....	352
19.4.5	中断请求保持.....	355
第二十章	待机功能.....	356
20.1	待机功能和配置.....	356
20.1.1	待机功能.....	356
20.1.2	控制待机功能的寄存器.....	358
20.2	待机功能操作.....	360
20.2.1	HALT 模式.....	360
20.2.2	STOP 模式.....	364
第二十一章	复位功能.....	368
21.1	确定复位源的寄存器.....	375
第二十二章	上电清零电路.....	376
22.1	上电清零电路的功能.....	376
22.2	上电清零电路的配置.....	377
22.3	上电清零电路的操作.....	377
22.4	上电清零电路的注意事项.....	378
第二十三章	低电压检测电路.....	380
23.1	低电压检测电路的功能.....	380
23.2	低电压检测电路的配置.....	380
23.3	控制低电压检测电路的寄存器.....	381
23.4	低电压检测电路的操作.....	382
23.5	低电压检测电路的注意事项.....	386
第二十四章	选项字节.....	390
第二十五章	FLASH 存储器.....	391
25.1	内部存储器容量切换寄存器.....	391
25.2	用 Flash 存储器编程器写入.....	392
25.3	编程环境.....	396
25.4	通信模式.....	396
25.5	板上引脚的处理.....	399
25.5.1	FLMD0 引脚.....	399
25.5.2	FLMD1 引脚.....	399
25.5.3	串行接口引脚.....	400
25.5.4	RESET 引脚.....	402
25.5.5	端口引脚.....	402
25.5.6	其它信号引脚.....	402
25.5.7	电源.....	402

25.6	编程方法	403
25.6.1	控制 flash 存储器	403
25.6.2	Flash 存储器编程模式.....	403
25.6.3	选择通信模式.....	404
25.6.4	通信命令.....	405
25.7	通过自写的 Flash 存储器编程	406
25.7.1	用于自编程功能的寄存器.....	407
25.8	引导交换功能	411
25.8.1	引导交换功能概述.....	411
25.8.2	存储器映射和引导区域	412
第二十六章 片上调试功能		413
第二十七章 指令集		414
27.1	在操作列表中使用的约定	414
27.1.1	操作数标识符和描述方法.....	414
27.1.2	操作栏的说明.....	415
27.1.3	操作栏标志的说明.....	415
27.2	运算列表	416
27.3	按寻址类型列出指令	424
第二十八章 电气特性		427
第二十九章 封装图		440
第三十章 等待注意事项		441
30.1	等待注意事项	441
30.2	产生等待的周边硬件	442
30.3	等待发生示例	443
附录 A 开发工具		444
A.1	软件包	448
A.2	语言处理软件	448
A.3	控制软件	449
A.4	Flash 存储器编程工具	450
A.4.1	使用 flash 存储器编程器 PG-FP5, FL-PR5, PG-FP4, FL-PR4 和 PG-FPL 时.....	450
A.4.2	使用带有编程功能的片上调试仿真器 QB-MINI2 时	450
A.5	调试工具（硬件）	451
A.5.1	当使用在线仿真器 QB-780714.....	451
A.5.2	当使用片上调试仿真器 QB-78K0MINI.....	451
A.5.3	使用带有编程功能的片上调试仿真器 QB-MINI2	452
A.6	调试工具（软件）	452
附录 B 寄存器索引		453

B.1	寄存器索引（按寄存器名称的字母排序）	453
B.2	寄存器索引（按寄存器符号的字母排序）	457
附录 C	再版修订记录	461
C.1	此版本的主要修订记录	461

第一章 概述

1.1 特征

- 最短指令执行时间可以从高速 (0.1 μs : @ 20 MHz 以 X1 输入时钟操作) 变化到低速 (8.33 μs : @ 240 kHz 以内部振荡时钟操作)
- 通用寄存器: 8 位 \times 32 个寄存器 (8 位 \times 8 个寄存器 \times 4 组)
- 片上乘法器 / 除法器
 - 16 位 \times 16 位 = 32 位 (乘法)
 - 32 位 \div 16 位 = 32 位, 16 位余数 (除法)
- ROM, RAM 容量

项目 产品代号	程序存储器 (ROM)		数据存储器 (内部高速 RAM)
	$\mu\text{PD78F0714}$	Flash 存储器	32 KB

- 片上单路供电 flash 存储器
- 自编程 (带引导交换功能)
- 片上调试功能
- 片上上电清零 (POC) 电路和低电压检测电路 (LVI)
- 通过使用内部振荡器, 这种 CPU 默认启动可以实现快启动
- 片上看门狗定时器 (可以以内部振荡时钟操作)
- 片上时钟输出 / 蜂鸣器输出控制器 00
- 片上实时输出端口
- I/O 端口: 48
- 定时器: 7 通道
- 串行接口: 2 通道 (UART: 1 通道, CSI: 1 通道)
- 10 位分辨率 A/D 转换器: 8 通道
- 电源电压: $V_{\text{DD}} = 4.0$ 到 5.5 V
- 工作环境温度: $T_{\text{A}} = -40$ 到 $+85^{\circ}\text{C}$

<R>

1.2 应用

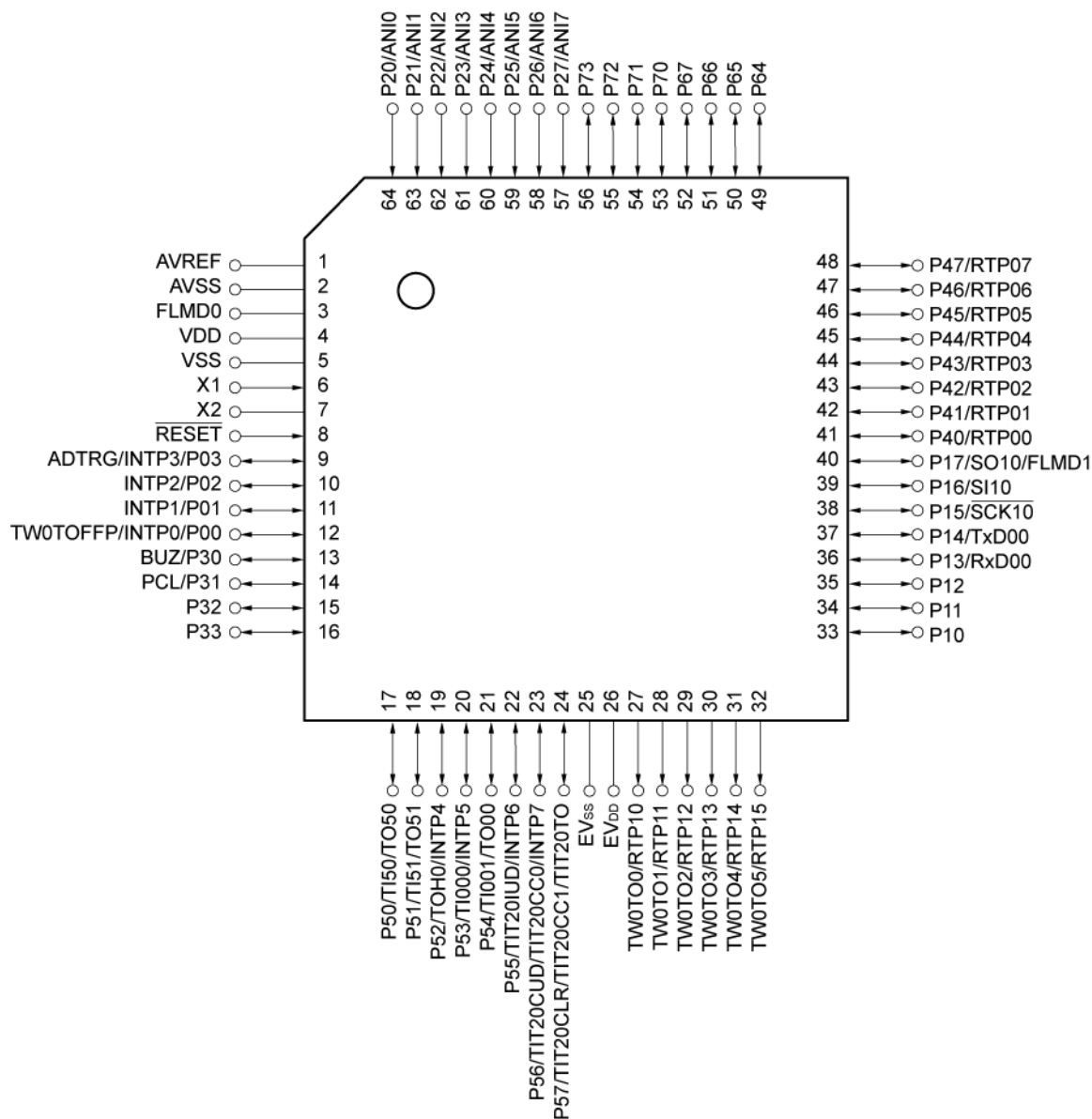
- 家用电器
 - 电冰箱
 - 洗碗机
 - 洗衣机, 干衣机
 - 室外空调单元
 - 微波炉, 电饭锅
- 工业设备
 - 水泵

1.3 订购信息

产品代号	封装
μ PD78F0714GK-9ET	64 引脚塑封 TQFP (精密间距) (12 × 12)

1.4 引脚配置 (俯视图)

- 64 引脚塑封 TQFP (精密间距) (12 × 12)

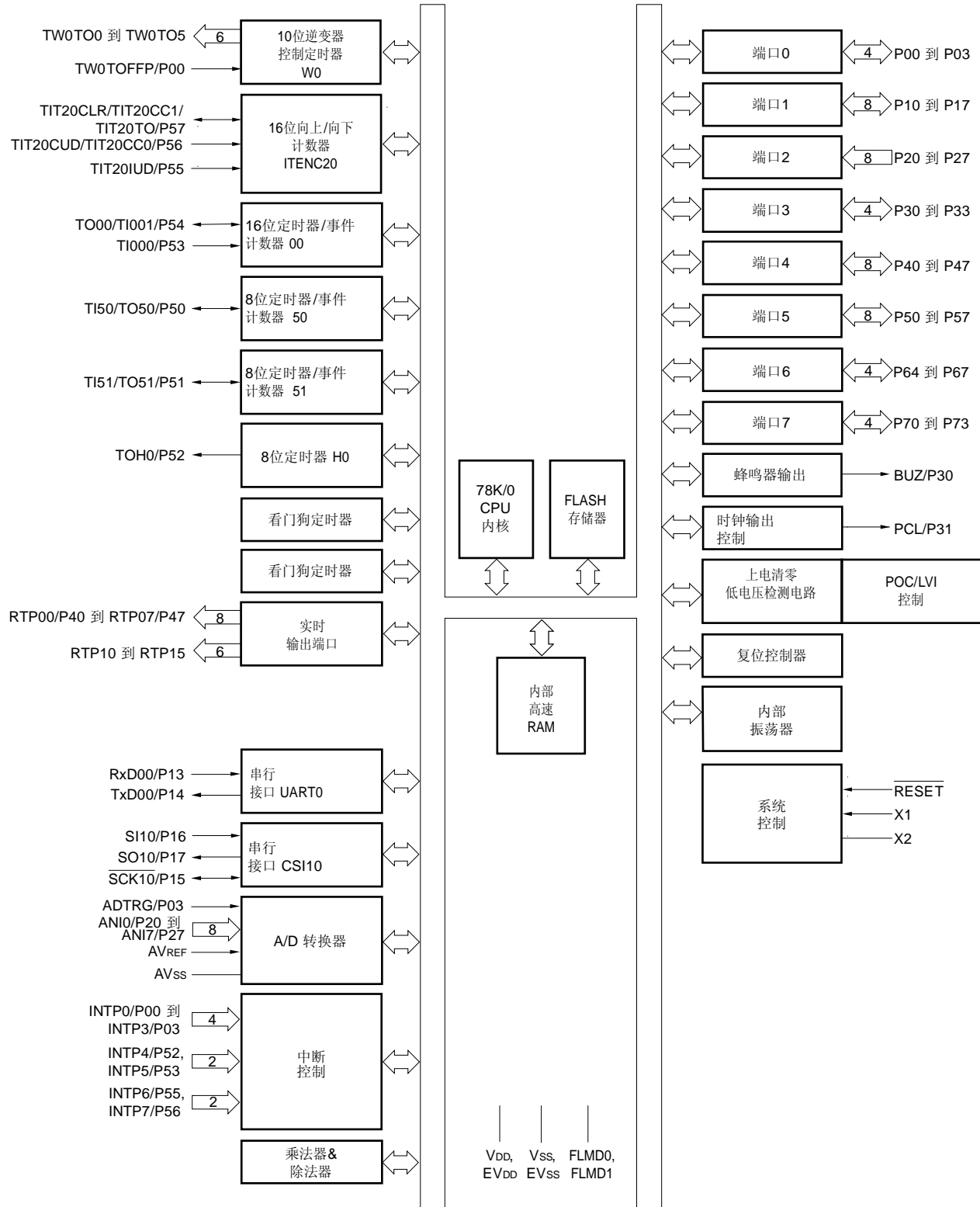


注意事项 连接 AV_{ss} 引脚到 V_{ss}。

引脚标识

ADTRG:	A/D 触发输入	RxD00:	接收数据
ANI0 到 ANI7:	模拟输入	SCK10	串行时钟输入 / 输出
AVREF:	模拟参考电压	SI10:	串行数据输入
AVss:	模拟地	SO10:	串行数据输出
BUZ:	蜂鸣器输出	TI000, TI001:	定时器输入
EVDD:	端口电源	TI50, TI51:	定时器输入
EVss:	端口地	TIT20CLR:	向上 / 向下计数器清除
FLMD0, FLMD1:	Flash 编程模式	TIT20CUD:	向上 / 向下计数器时钟选择
INTP0 到 INTP7:	外部中断输入	TIT20CC0, TIT20CC1:	向上 / 向下计数器捕获输入
P00 到 P03:	端口 0	TIT20IUD:	向上 / 向下计数器时钟
P10 到 P17:	端口 1	TIT20TO:	向上 / 向下计数器输出
P20 到 P27:	端口 2	TO00:	定时器输出
P30 到 P33:	端口 3	TO50, TO51:	定时器输出
P40 到 P47:	端口 4	TOH0:	定时器输出
P50 到 P57:	端口 5	TW0TO0 to TW0TO5:	定时器输出
P64 到 P67:	端口 6	TW0TOFFP:	定时器输出关闭
P70 到 P73:	端口 7	TxD00:	发送数据
PCL:	可编程时钟输出	VDD:	电源
RESET:	复位	Vss:	地
RTP00 到 RTP07:	实时输出端口	X1, X2:	晶体振荡器 (X1 输入时钟)
RTP10 到 RTP15:	实时输出端口		

1.5 框图



1.6 功能概述

项目		μ PD78F0714
内部存储器	Flash 存储器（支持自编程）	32 KB
	高速 RAM	1 KB
存储器空间		64 KB
X1 输入时钟（振荡频率）		陶瓷 / 晶体 / 外部时钟振荡 [20 MHz ($V_{DD} = 4.0$ 到 5.5 V)]
内部振荡时钟（振荡频率）		内部振荡器（240 kHz（典型值））
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 / 0.2 μ s / 0.4 μ s / 0.8 μ s / 1.6 μ s（X1 输入时钟：@ $f_{XP} = 20$ MHz 操作） 8.3 μ s / 16.6 μ s / 33.2 μ s / 66.4 μ s / 132.8 μ s（典型值）（内部振荡时钟：@ $f_R = 240$ kHz（典型值）操作）
指令集		<ul style="list-style-type: none"> • 16 位运算 • 乘 / 除（8 位 \times 8 位，16 位 \div 8 位） • 位操作（置位，复位，检验和布尔运算） • BCD 调整，等等
I/O 端口		总共： 48 CMOS I/O 40 CMOS 输入 8
定时器		<ul style="list-style-type: none"> • 10 位逆变器控制定时器： 1 通道 • 16 为向上 / 向下计数器： 1 通道 • 16-位定时器 / 事件计数器： 1 通道 • 8 位定时器 / 事件计数器： 2 通道 • 8 位定时器： 1 通道 • 看门狗定时器： 1 通道
	定时器输出	11（逆变器控制输出：6）
时钟输出		156.25 kHz, 312.5 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz, 20MHz (X1 输入时钟：20 MHz)
蜂鸣器输出		2.44 kHz, 4.88 kHz, 9.77 kHz, 19.5kHz (X1 输入时钟：20 MHz)
实时输出口		<ul style="list-style-type: none"> • 8 位 \times 1 或 4 位 \times 2 • 6 位 \times 1 或 4 位 \times 2
A/D 转换器		10 位分辨率 \times 8 通道
串行接口		<ul style="list-style-type: none"> • UART 模式： 1 通道 • 3 线串行 I/O 模式： 1 通道
乘法器 / 除法器		<ul style="list-style-type: none"> • 16 位 \times 16 位 = 32 位（乘法） • 32 位 \div 16 位 = 32 位 16 位余数（除法）
向量中断源	内部	非屏蔽：1，可屏蔽：19
	外部	8
复位		<ul style="list-style-type: none"> • 使用 RESET 引脚的复位 • 看门狗定时器引起的内部复位 • 上电清零引起的内部复位 • 低电压检测电路引起的内部复位
电源电压		$V_{DD} = 4.0$ 到 5.5 V
工作环境温度		$T_A = -40$ 到 $+85^\circ\text{C}$
封装		64 引脚塑封 TQFP（精密间距）（12 \times 12）

<R>

定时器的概述如下所示。

		10 位逆变器控制 定时器	16 位向上 / 向下 计数器 ITENC20	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	看门狗定时 器
					TM50	TM51		
操作模式	间隔定时器	1 通道	1 通道	1 通道	1 通道	1 通道	1 通道	-
	外部事件计数 器	-	1 通道	1 通道	1 通道	1 通道	-	-
功能	定时器输出	6 输出	1 输出	1 输出	1 输出	1 输出	1 输出	-
	PPG 输出	-	-	1 输出	-	-	-	-
	PWM 输出	6 输出	1 输出	-	1 输出	1 输出	1 输出	-
	脉冲宽度测量	-	-	2 输出	-	-	-	-
	方波输出	-	1 输出	1 输出	1 输出	1 输出	1 输出	-
	看门狗定时器	-	-	-	-	-	-	1 通道
	中断源	4	4	2	1	1	1	-

第二章 引脚功能

2.1 引脚功能列表

存在三种类型的引脚 I/O 缓存电源：AV_{REF}，EV_{DD} 和 V_{DD}。这些电源和引脚之间的关系如下所示。

表 2-1. 引脚 I/O 缓存电源

电源	对应引脚
AV _{REF}	P20 到 P27
EV _{DD}	P20 到 P27 以外的端口引脚
V _{DD}	端口引脚以外的引脚

(1) 端口引脚 (1 / 2)

引脚名称	I/O	功能	复位后	复用功能
P00	I/O	端口 0。 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片上上拉电阻的使用。	输入	INTP0/TW0TOFFP
P01				INTP1
P02				INTP2
P03				INTP3/ADTRG
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片上上拉电阻的使用。	输入	-
P11				-
P12				-
P13				RxD00
P14				TxD00
P15				SCK10
P16				SI10
P17				SO10/FLMD1
P20 到 P27	输入	端口 2。 8 位输入端口。	输入	ANI0 到 ANI7
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片上上拉电阻的使用。	输入	BUZ
P31				PCL
P32				-
P33				-
P40 到 P47	I/O	端口 4。 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片上上拉电阻的使用。	输入	RTP00 到 RTP07

(1) 端口引脚 (2 / 2)

引脚名称	I/O	功能	复位后	复用功能
P50	I/O	端口 5。 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片上上拉电阻的使用。	输入	TI50/TO50
P51				TI51/TO51
P52				TOH0/INTP4
P53				TI000/INTP5
P54				TI001/TO00
P55				TIT20IUD/INTP6
P56				TIT20CUD /TIT20CC0/INTP7
P57				TIT20CLR /TIT20CC1 /TIT20TO
P64 到 P67	I/O	端口 6。 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片上上拉电阻的使用。	输入	—
P70 到 P73	I/O	端口 7。 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片上上拉电阻的使用。	输入	—

(2) 非端口引脚 (1/2)

引脚名称	I/O	功能	复位后	复用功能
INTP0	输入	可以指定有效沿（上升沿,下降沿或双边沿）的外部中断请求输入	输入	P00/TW0TOFFP
INTP1				P01
INTP2				P02
INTP3				P03/ADTRG
INTP4				P52/TOH0
INTP5				P53/TI000
INTP6				P55/TIT20IUD
INTP7				P56/TIT20CC0 /TIT20CUD
SI10	输入	输入到串行接口的串行数据	输入	P16
SO10	输出	从串行接口输出的串行数据	输入	P17/FLMD1
$\overline{SCK10}$	I/O	串行接口的时钟输入 / 输出	输入	P15
RxD00	输入	输入到异步串行接口的串行数据	输入	P13
TxD00	输出	从异步串行接口输出的串行数据	输入	P14
TW0TOFFP	输入	停止 10 位逆变器控制定时器输出的外部输入	输入	P00/INTP0
TW0TO0- TW0TO5	输出	10 位逆变器控制定时器输出	输出	RTP10-RTP15
TIT20IUD	输入	输入到 16 位向上 / 向下计数器的外部计数时钟	输入	P55/INTP6
TIT20CUD		输入到 16 位向上 / 向下计数器的计数操作切换		P56/TIT20CC0 /INTP7
TIT20CC0		输入到 16 位向上 / 向下计数器的外部捕获触发		P56/TIT20CUD /INTP7
TIT20CC1				P57/TIT20CLR /TIT20TO
TIT20CLR		输入到 16 位向上 / 向下计数器的外部清除		P57/TIT20CC1 /TIT20TO
TIT20TO	输出	16 位向上 / 向下计数器的脉冲信号输出	输入	P57/TIT20CLR /TIT20CC1
TI000	输入	输入到 16 位定时器 / 事件计数器 00 的外部计数时钟 输入到 16 位定时器 / 事件计数器 00 的捕获寄存器（CR000, CR010）的捕获触发	输入	P53/INTP5
TI001		输入到 16 位定时器 / 事件计数器 00 的捕获寄存器（CR000）的捕获触发		P54/TO00
TO00	输出	16 位定时器 / 事件计数器 00 输出	输入	P54/TI001
TI50	输入	输入到 8 位定时器 / 事件计数器 50 的外部计数时钟	输入	P50/TO50
TI51		输入到 8 位定时器 / 事件计数器 51 的外部计数时钟		P51/TO51
TO50	输出	8 位定时器 / 事件计数器 50 输出	输入	P50/TI50
TO51		8 位定时器 / 事件计数器 51 输出		P51/TI51
TOH0		8 位定时器 H0 输出		P52/INTP4

(2) 非端口引脚 (2 / 2)

引脚名称	I/O	功能	复位后	复用功能
PCL	输出	时钟输出 (用于 X1 输入时钟的修整)	输入	P31
BUZ	输出	蜂鸣器输出	输入	P30
RTP00 到 RTP07	输出	实时输出端口 0 输出	输入	P40 到 P47
RTP10 到 RTP15	输出	实时输出端口 1 输出	输出	TW0TO0 到 TW0TO5
ADTRG	输入	A/D 转换器触发输入。	输入	P03/INTP3
ANI0 到 ANI7	输入	A/D 转换器模拟输入。	输入	P20 到 P27
AVREF	输入	A/D 转换器参考电压输入和端口 2 的正电源	-	-
AVSS	-	A/D 转换器地电平。确保与 EVSS 或 VSS 电平相同。	-	-
RESET	输入	系统复位输入	-	-
X1	输入	用于 X1 输入时钟振荡的连接晶振	-	-
X2	-		-	-
VDD	-	正电源 (除端口以外)	-	-
EVDD	-	端口的正电源	-	-
VSS	-	地电平 (除端口以外)	-	-
EVSS	-	端口的地电平	-	-
FLMD0	-	Flash 存储器编程模式设置	-	-
FLMD1	输入		输入	P17/SO10

2.2 引脚功能说明

2.2.1 P00 到 P03（端口 0）

P00 到 P03 用作 4 位 I/O 端口。这些引脚也用作外部中断请求输入、定时器输出停止外部信号和 A/D 转换器触发输入。

以下操作模式可以以 1 位为单位进行指定。

(1) 端口模式

P00 到 P03 用作 4 位 I/O 端口。P00 到 P03 可以通过使用端口模式寄存器 0（PM0）以 1 位为单位设置为输入或输出端口。可以通过上拉电阻选择寄存器 0（PU0）来指定片上上拉电阻的使用。

(2) 控制模式

P00 到 P03 用作外部中断请求输入、定时器输出停止外部信号和 A/D 转换器触发输入。

(a) INTP0 到 INTP3

这些是可以指定为有效沿（上升沿，下降沿，或双边沿）的外部中断请求输入引脚。当有效沿输入时，INTP2 也用作实时输出端口的的外部触发信号输入。

(b) TW0TOFFP

这是用来停止定时器输出（TW0TO0 到 TW0TO5）的外部输入引脚。

(c) ADTRG

这是 A/D 转换器的外部触发信号输入引脚。

2.2.2 P10 到 P17（端口 1）

P10 到 P17 用作 8 位 I/O 端口。这些引脚也用作串行接口数据 I/O、时钟 I/O 和 flash 存储器编程模式设置的引脚。

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 到 P17 可以通过使用端口模式寄存器 1（PM1）以 1 位为单位设置为输入或输出端口。可以通过上拉电阻选择寄存器 1（PU1）来指定片上上拉电阻的使用。

(2) 控制模式

P10 到 P17 用作串行接口数据 I/O 和时钟 I/O 的引脚。

(a) SI10

这是串行接口的串行数据输入引脚。

(b) SO10

这是串行接口的串行数据输出引脚。

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

这是串行接口的串行时钟 I/O 引脚。

(d) RxD00

这是异步串行接口的串行数据输入引脚。

(e) TxD00

这是异步串行接口的串行数据输出引脚。

(f) FLMD1

该引脚设置 flash 存储器的编程模式。

2.2.3 P20 到 P27 (端口 2)

P20 到 P27 用作 8 位输入端口。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 到 P27 用作 8 位输入端口。

<R> **注意事项** 当在端口模式下使用 P20 到 P27 时，要在 $EV_{DD} = AV_{REF}$ 下使用。

(2) 控制模式

P20 到 P27 可用作 A/D 转换器模拟输入引脚 (ANI0 到 ANI7)。将这些引脚用作模拟输入引脚时，请参见 15.6 A/D 转换器的注意事项中的 (5) ANI0/P20 到 ANI7/P27。

2.2.4 P30 到 P33 (端口 3)

P30 到 P33 用作 4 位 I/O 端口。这些引脚也用作时钟输出和蜂鸣器输出引脚。

以下操作模式可以以 1 位为单位进行指定。

(1) 端口模式

P30 到 P33 用作 4 位 I/O 端口。P30 到 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) 来指定片上上拉电阻的使用。

(2) 控制模式

P30 到 P33 用作时钟输出和蜂鸣器输出引脚。

(a) BUZ

这是蜂鸣器输出引脚。

(b) PCL

这是时钟输出引脚。

注意事项 确保在复位后下拉 P31 以防止故障。

备注 μ PD78F0714 的 P31 和 P32 引脚可以在使用片上调试功能时设置片上调试模式。关于详细情况，参见第二十六章 片上调试功能。

2.2.5 P40 到 P47（端口 4）

P40 到 P47 用作 8 位 I/O 端口。这些引脚也用作实时输出端口引脚。

以下操作模式可以被指定。

(1) 端口模式

P40 到 P47 用作 8 位 I/O 端口。P40 到 P47 可以通过使用端口模式寄存器 4（PM4）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 4（PU4）来指定片上上拉电阻的使用。

(2) 控制模式

P40 到 P47 用作实时输出端口（RTP00 到 RTP07）的引脚，它与一个触发同步输出数据。

2.2.6 P50 到 P57（端口 5）

P50 到 P57 用作 8 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。

以下操作模式可以被指定。

(1) 端口模式

P50 到 P57 用作 8 位 I/O 端口。P50 到 P57 可以通过使用端口模式寄存器 5（PM5）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 5（PU5）来指定片上上拉电阻的使用。

(2) 控制模式

P50 到 P57 用作外部中断请求输入以及定时器 I/O 的引脚。

(a) INTP4 到 INTP7

这些是可以指定有效沿（上升沿，下降沿，或双边沿）的外部中断请求输入引脚。

(b) TI50, TI51

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

(c) TO50, TO51

这些是从 8 位定时器 / 事件计数器 50 和 51 输出的引脚。

(d) TI000

这是用于将外部计数时钟输入到 16 位定时器 / 事件计数器 00 中的引脚，也用于将捕获触发信号输入到捕获寄存器（CR00, CR01）中。

(e) TI001

这用于将捕获触发信号输入到 16 位定时器 / 事件计数器 00 的捕获寄存器（CR00）中的引脚。

(f) TO00, TOH0

这些是 16 位定时器 / 事件计数器 00 和 8 位定时器 H0 的定时器输出引脚。

(g) TIT20IUD

这是用于将外部计数时钟输入到 16 位向上 / 向下计数器 ITENC20 中的引脚。

(h) TIT20IUD

这是用于将计数操作切换信号输入到 16 位向上 / 向下计数器 ITENC20 中的引脚。

(i) TIT20CLR

这是用于将清除信号输入到 16 位向上 / 向下计数器 ITENC20 中的引脚。

(j) TIT20CC0, TIT20CC1

这些是用于将外部捕获触发输入到 16 位向上 / 向下计数器 ITENC20 中的引脚。

(k) TIT20TO

这是 16 位向上 / 向下计数器 ITENC20 的输出引脚。

2.2.7 P64 到 P67 (端口 6)

P60 到 P67 用作 4 位 I/O 端口。P64 到 P67 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。

可以通过上拉电阻选择寄存器 6 (PU6) 来对 P64 到 P67 指定片上上拉电阻的使用。

2.2.8 P70 到 P73 (端口 7)

P70 到 P73 用作 4 位 I/O 端口。P70 到 P73 可以通过使用端口模式寄存器 7 (PM7) 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 7 (PU7) 来指定片上上拉电阻的使用。

2.2.9 TW0TO0/RTP10 到 TW0TO5/RTP15

这些是 10 位逆变器控制定时器输出引脚。

并且, 这些引脚也用作实时输出端口引脚。

2.2.10 AV_{REF}

这是 A/D 转换器参考电压输入引脚。

未使用 A/D 转换器时, 直接将该引脚连接在 EV_{DD} 或 V_{DD}[※]上。

注 当端口 2 用作数字端口时, 直接将其连接到 EV_{DD}。

2.2.11 AV_{SS}

这是 A/D 转换器地电平引脚。即使当 A/D 转换器没有被使用时, 通常也将该引脚的电平设置为与 EV_{SS} 引脚或 V_{SS} 引脚相同的电平。

2.2.12 $\overline{\text{RESET}}$

该引脚为有效低电平系统复位输入引脚。

2.2.13 X1 和 X2

这些是用于连接 X1 输入时钟晶振的引脚。

当提供外部时钟时, 输入一个信号到 X1 引脚并输入反相信号到 X2 引脚。

备注 带片上调试功能的产品 (产品代号未定) 的 X1 和 X2 引脚可以在使用片上调试功能时设置片上调试模式。关于详细情况, 参见第二十六章 片上调试功能。

2.2.14 V_{DD} 和 EV_{DD}

V_{DD} 是除端口以外的正电源引脚。

EV_{DD} 是端口的正电源引脚。

2.2.15 Vss 和 EVss

Vss 是除端口以外的地电平引脚。

EVss 是端口的地电平引脚。

2.2.16 FLMD0

该引脚设置 flash 存储器的编程模式。

在 flash 存储器编程模式下，连接 FLMD0 到 flash 存储器编程器，并且在正常操作模式下，连接到 EVss 或 Vss。

2.3 引脚I/O电路以及不使用引脚的推荐连接

表 2-2 显示了引脚 I/O 电路的类型以及不使用引脚的推荐连接。

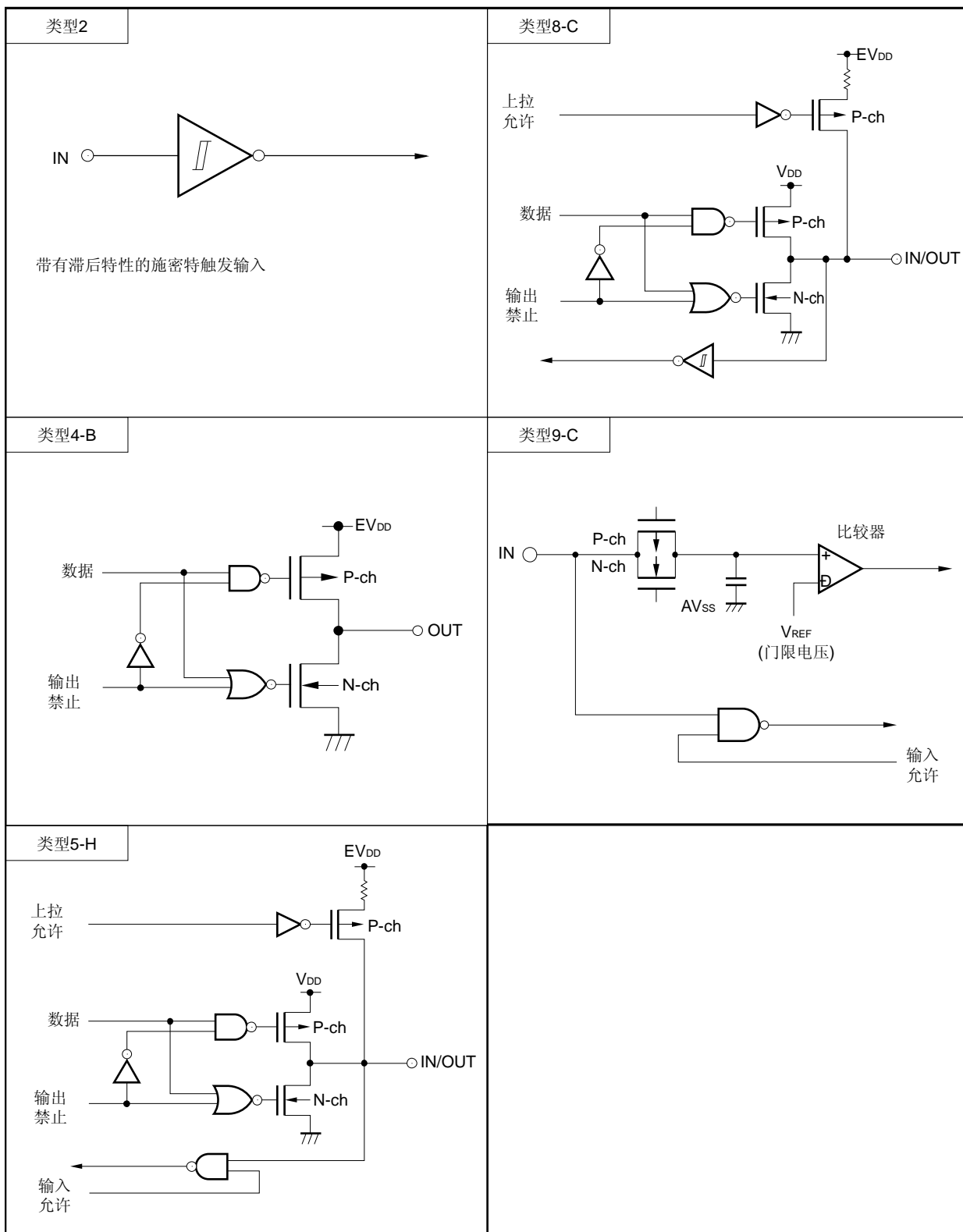
各个类型的 I/O 电路和配置请参见图 2-1。

表 2-2. 引脚 I/O 电路类型

引脚名称	I/O 电路类型	I/O	不使用的引脚的推荐连接
P00/INTP0/TW0TOFFP	8-C	I/O	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 输出： 保持开路。
P01/INTP1			
P02/INTP2			
P03/INTP3/ADTRG			
P10			
P11			
P12			
P13/RxD00			
P14/TxD00			
P15/ SCK10			
P16/SI10	8-C		
P17/SO10/FLMD1	5-H		
P20/ANI0 到 P27/ANI7	9-C	输入	连接到 EV _{DD} 或 EV _{SS} 。
P30/BUZ	5-H	I/O	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 输出： 保持开路。
P31/PCL			
P32			
P33			
P40/RTP00 到 P47/RTP07			
P50/TI50/TO50			
P51/TI51/TO51			
P52/TOH0/INTP4			
P53/TI000/INTP5			
P54/TI001/TO00			
P55/TIT20IUD/INTP6	8-C		
P56/TIT20CUD/TIT20CC0/INTP7			
P57/TIT20CLR/TIT20CC1/TIT20TO			
P64 到 P67			
P70 到 P73			
P64 到 P67			
P70 到 P73			
P64 到 P67			
P70 到 P73			
P64 到 P67			
TW0TO0/RTP10-TW0TO5/RTP15	4-B	输出	保持开路。
RESET	2	输入	-
AV _{REF}	-	-	直接连接到 EV _{DD} 或 V _{DD} [*] 。
AV _{SS}			直接连接到 EV _{SS} 或 V _{SS} 。
FLMD0			连接到 EV _{SS} 或 V _{SS} 。

注 当端口 2 用作数字端口时，直接将其连接到 EV_{DD}。

图 2-1. 引脚 I/O 电路列表



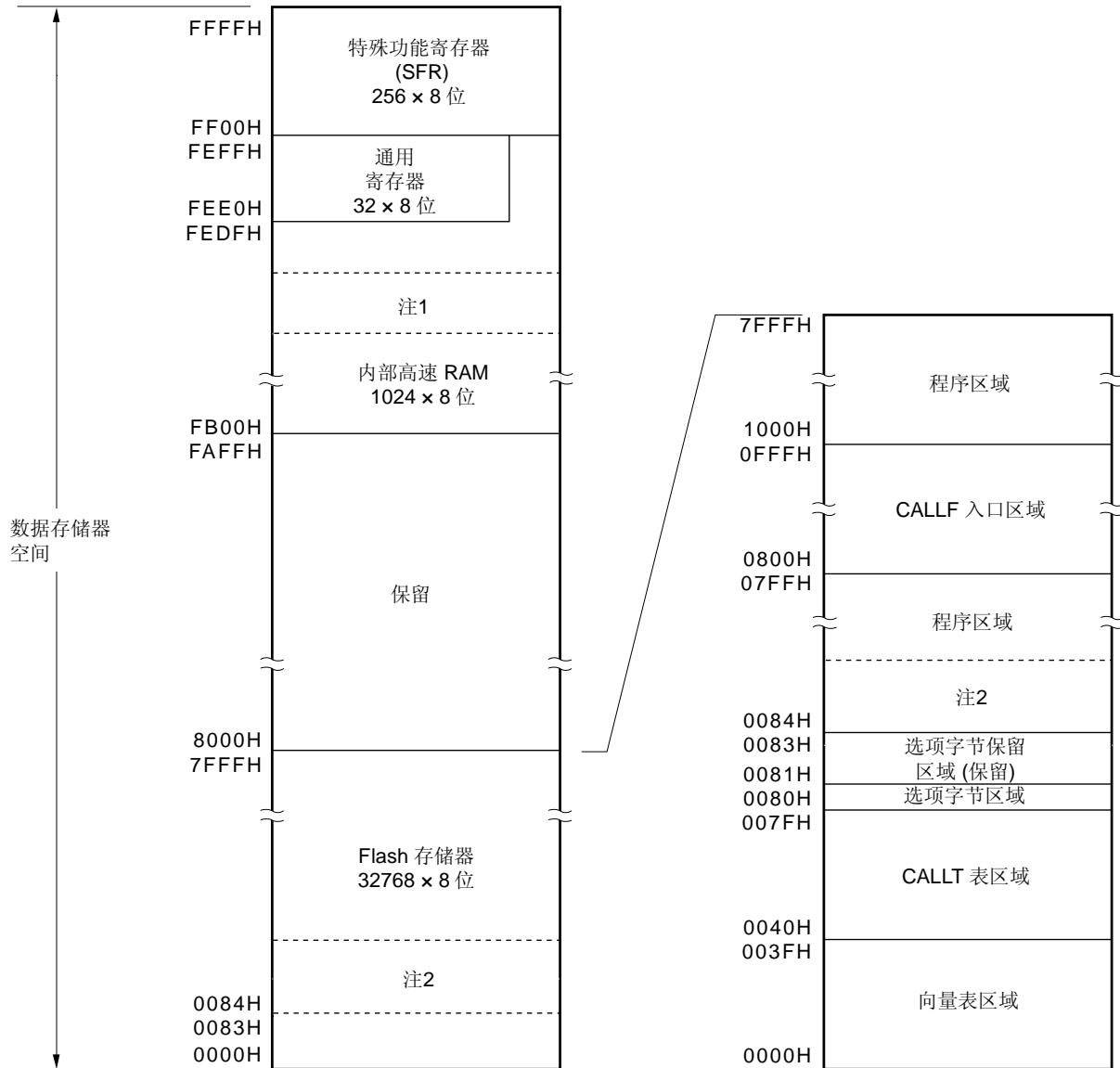
第三章 CPU架构

3.1 存储器映射

μ PD78F0714 产品可以访问一个 64KB 的存储器空间。图 3-1 表示存储器映射。

注意事项 因为内部存储器大小切换寄存器（IMS）的初始值为 CFH，通过初始化设置其为 C8H。

图 3-1. 存储器映射 (μ PD78F0714)



- 注
1. 该区域在片上调试期间占用 9 字节（计划），因为它用作通信期间的用户数据备份区域。
 2. 该区域在片上调试期间不能使用，因为它用作通信命令区域（256 字节到 1 KB）。

3.1.1 内部程序存储空间

内部程序存储空间用于保存程序以及表数据。通常情况下，它使用程序计数器（PC）进行寻址。如下所示， μ PD78F0714 产品中包含有内部 ROM（flash 存储器）。

表 3-1. 内部 ROM 容量

产品代号	内部 ROM	
	结构	容量
μ PD78F0714	Flash 存储器	32768 × 8 位（0000H 到 7FFFH）

内部程序存储空间分为以下各区域。

(1) 向量表区域

64 字节的区域 0000H 到 003FH 被用作向量表区域。在发生复位信号输入或生成各个中断请求之后的跳转指令程序开始地址保存在向量表区域中。

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

表 3-2. 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	0020H	INTCM11
		0022H	INTCC10
0004H	INTLVI	0024H	INTCC11
0006H	INTP0	0026H	— ^注
0008H	INTP1	0028H	INTTM00
000AH	INTP2	002AH	INTTM01
000CH	INTP3	002CH	INTSRE00
000EH	INTP4	002EH	INTSR00
0010H	INTP5	0030H	INTST00
0012H	INTP6	0032H	INTTM50
0014H	INTP7	0034H	INTTM51
0016H	INTTW0UD	0036H	INTTMH0
0018H	INTTW0CM3	0038H	INTCSI10
001AH	INTTW0CM4	003AH	INTDMU
001CH	INTTW0CM5	003CH	INTAD
001EH	INTCM10		

注 不存在对应于向量表地址 0026H 的中断请求。

(2) CALLT 指令表区域

64 字节区域 0040H 到 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

1 个字节区域 0080H 被保留为选项字节区域。关于详细情况，请参见第二十四章 选项字节。

(4) CALLF 指令入口区域

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

3.1.2 内部数据存储空间

μ PD78F0714 产品包含有以下 RAM。

(1) 内部高速 RAM

内部高速 RAM 以 1024×8 位的配置分配到区域 FB00H 到 FEFH。

32 字节区域 FEE0H 到 FEFH 分配给 4 个通用寄存器 bank 中, 每个 bank 中包含有 8 个 8 位寄存器。

该区域不能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。

3.1.3 特殊功能寄存器 (SFR) 区域

片上周边硬件特殊功能寄存器 (SFR) 位于 FF00H 到 FFFFH 的区域中 (参见 3.2.3 特殊功能寄存器 (SFR) 中的表 3-3 特殊功能寄存器列表)。

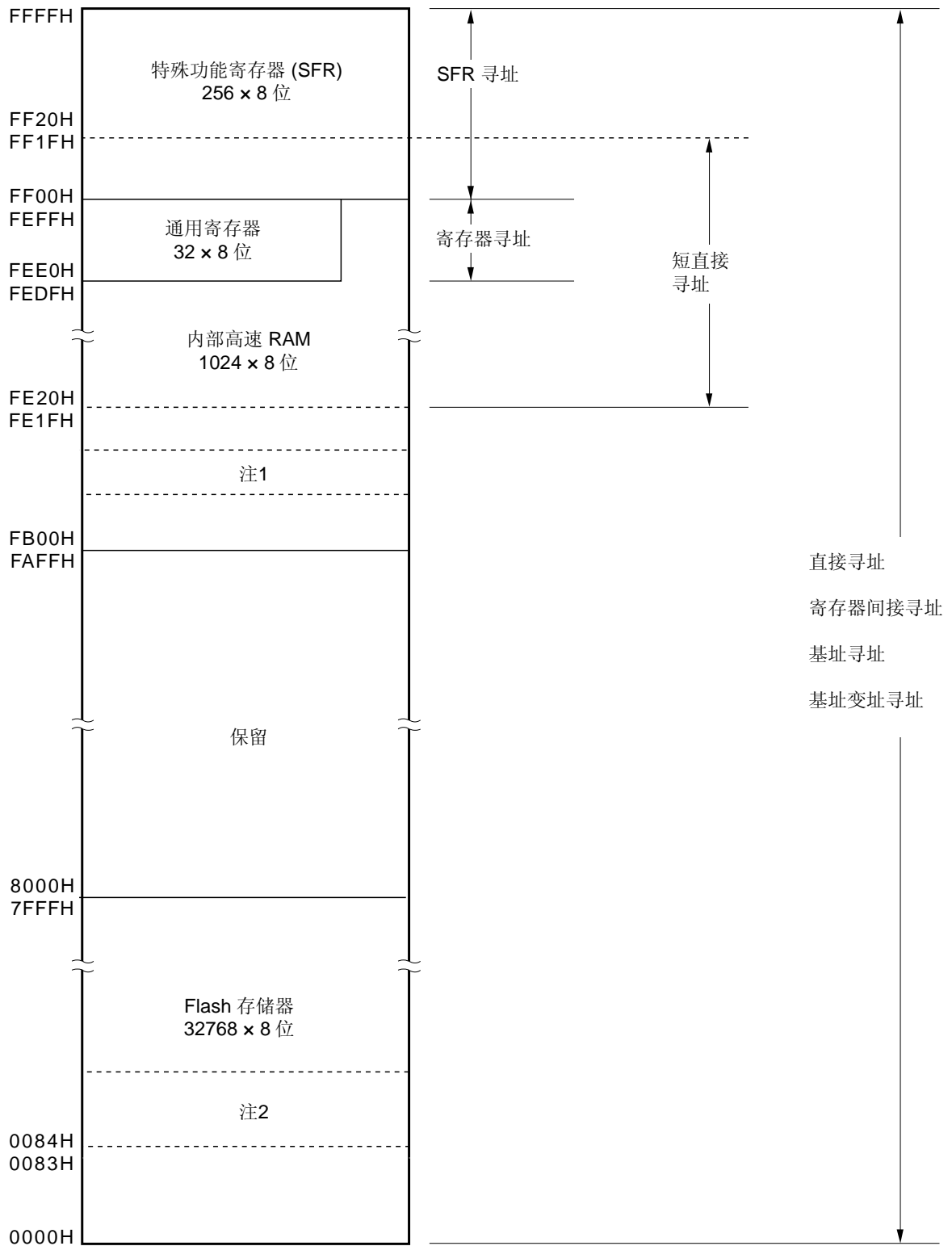
注意事项 不要访问 SFR 没有分配的地址。

3.1.4 数据存储寻址

寻址是指指定下一个要执行的指令的地址, 或寄存器地址, 或者与指令执行相关的存储器的地址的方法。

基于可操作性及其它一些考虑, μ PD78F0714 中与指令执行相关的存储器寻址具有多种模式。特别是对于那些包含有数据存储器的区域, 将可以使用为特殊功能寄存器 (SFR) 以及通用寄存器的功能所设计的特殊的寻址方法。图 3-2 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情况, 请参见 3.4 操作数地址寻址。

图 3-2. 数据存储器与寻址间的对应关系 (μ PD78F0714)



- 注
1. 该区域在片上调试期间占用 9 字节（计划），因为它用作通信期间的用户数据备份区域。
 2. 该区域在片上调试期间不能被使用，因为它用作通信命令区域（256 字节到 1 KB）。

3.2 处理器寄存器

μ PD78F0714 产品包含以下处理器寄存器。

3.2.1 控制寄存器

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

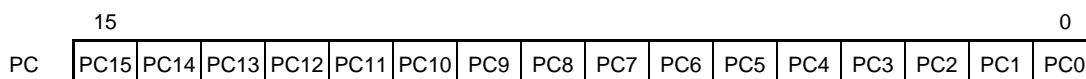
(1) 程序计数器（PC）

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

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

$\overline{\text{RESET}}$ 输入会将地址 0000H 和 0001H 处的复位向量表的值赋值到程序计数器中。

图 3-3. 程序计数器的格式



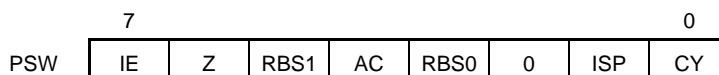
(2) 程序状态字（PSW）

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。

中断请求生成或在 PUSH PSW 指令执行时程序状态字的内容会被自动保存在堆栈中，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。

$\overline{\text{RESET}}$ 输入设置 PSW 为 02H。

图 3-4. 程序状态字的格式



(a) 中断允许标志（IE）

该标志用于控制 CPU 的中断请求响应操作。

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

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

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

(b) 零标志（Z）

当操作结果为零时，该标志被设置为 1。其它情况下则复位为 0。

(c) 寄存器 bank 选择标志 (RBS0 和 RBS1)

这些是 2 位标志，用于选择 4 个寄存器 bank 中的一个 bank。

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

(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。其它情况下则复位为 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) 所指定的等级向量中断请求 (参见 19.3 (3) 优先级指定标志寄存器 (PR0L, PR0H, PR1L, PR1H))。实际中断请求的响应是由中断允许标志 (IE) 来控制的。

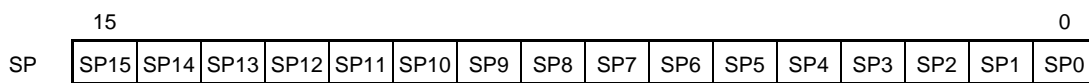
(f) 进位标志 (CY)

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溢出值，并可在位操作指令执行过程中用作位累加器。

(3) 堆栈指针 (SP)

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

图 3-5. 堆栈指针的格式



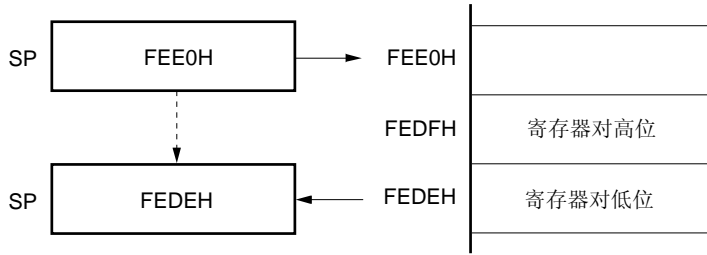
向堆栈存储器进行写 (保存) 操作前 SP 累减，从堆栈存储器进行读取 (恢复) 操作后 SP 累加。

各个堆栈保存 / 恢复数据的操作如图 3-6 和 3-7 所示。

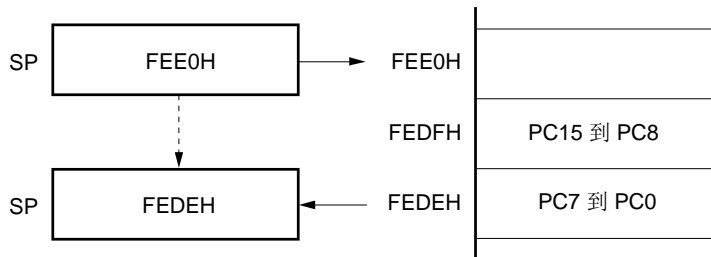
注意事项 由于 $\overline{\text{RESET}}$ 输入会使 SP 内容变为未知，因此在使用堆栈前请务必初始化 SP。

图 3-6. 保存在堆栈存储器中的数据

(a) PUSH rp 指令 (当 SP = FEE0H 时)



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



(c) 中断, BRK 指令 (当 SP = FEE0H 时)

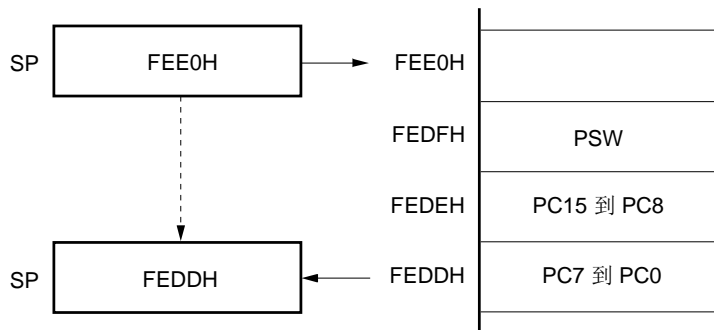
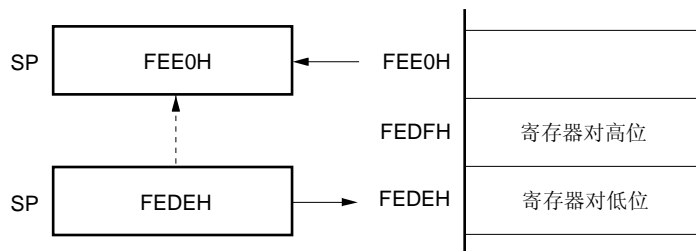
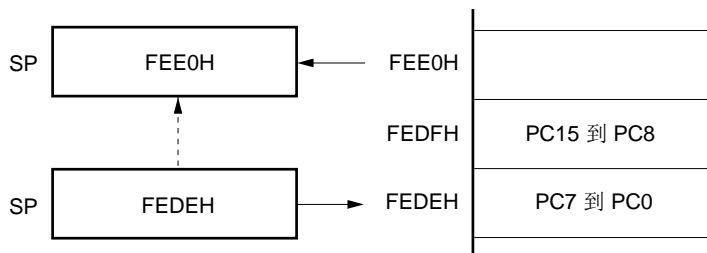


图 3-7. 从堆栈存储器中恢复的数据

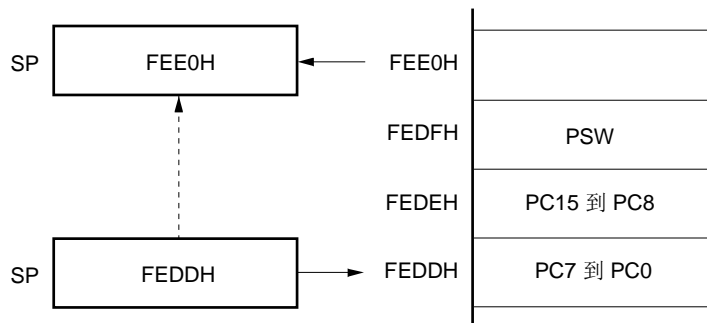
(a) POP rp 指令 (当 SP = FEDEH 时)



(b) RET 指令 (当 SP = FEDEH 时)



(c) RETI, RETB 指令 (当 SP = FEDDH 时)



3.2.2 通用寄存器

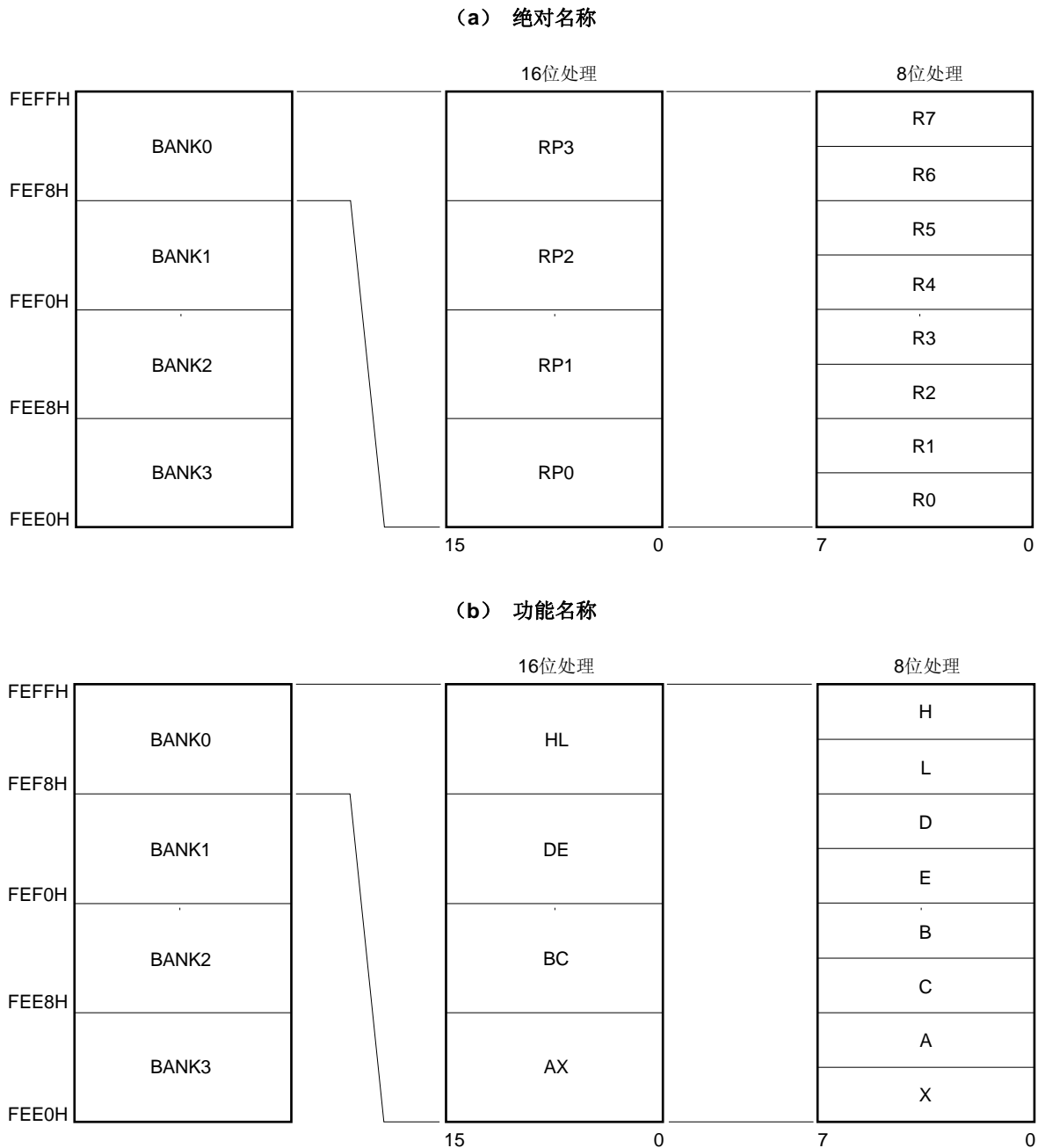
通用寄存器映射在数据存储器的特定地址（FEE0H 到 FEFH）上。通用寄存器由 4 个 bank 组成，每个 bank 由 8 个 8 位寄存器（X，A，C，B，E，D，L 和 H）组成。

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 到 R7 以及 RP0 到 RP3）来进行描述。

用于指令执行的寄存器 bank 通过 CPU 控制指令（SEL RBn）来进行设置。根据 4 寄存器 bank 的结构特点，可以通过在各个 bank 之中切换用于正常处理的寄存器以及用于中断的寄存器来编写高效的程序段。

图 3-8. 通用寄存器的结构



3.2.3 特殊功能寄存器 (SFR)

与通用寄存器不同，每个特殊功能寄存器都有一个特殊功能。

SFR 被分配到 FF00H 到 FFFFH 区域中。

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

每个操作位单元可以按如下进行指令。

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

表 3-3 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 RA18K0 中的保留字，而在 CC78K0 中则通过使用 #pragma sfr 导引来定义为一个 sfr 变量。当使用 RA78K0 或 ID78K0-QB 时，符号不能作为指令操作数写入。
- R/W
表示相应的特殊功能寄存器是被读取还是被写入。
R/W: 允许读取 / 写入
R: 只读
W: 只写
- 可操作位单元
表示可操作位单元 (1, 8, 或 16)。“-”表示操作不能实现的位单元。
- 复位后
表示 RESET 输入时各寄存器的状态。

表 3-3. 特殊功能寄存器列表 (1 / 5)

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	√	√	–	不确定
FF03H	端口寄存器 3	P3		R/W	√	√	–	00H
FF04H	端口寄存器 4	P4		R/W	√	√	–	00H
FF05H	端口寄存器 5	P5		R/W	√	√	–	00H
FF06H	端口寄存器 6	P6		R/W	√	√	–	00H
FF07H	端口寄存器 7	P7		R/W	√	√	–	00H
FF08H	10 位缓存寄存器 0	TW0BF CM0	TW0BF CM0L	R/W	–	√	√	0000H
FF09H			–			–		
FF0AH	10 位缓存寄存器 1	TW0BF CM1	TW0BF CM1L	R/W	–	√	√	0000H
FF0BH			–			–		
FF0CH	10 位缓存寄存器 2	TW0BF CM2	TW0BF CM2L	R/W	–	√	√	0000H
FF0DH			–			–		
FF0EH	10 位缓存寄存器 3	TW0BF CM3	TW0BF CM3L	R/W	–	√	√	00FFH
FF0FH			–			–		
FF10H	16 位向上 / 向下计数器	IT20 UDC	IT20 UDCL	R/W	–	√	√	0000H
FF11H			–			–		
FF12H	16 位比较寄存器 0	IT20 CM0	IT20 CM0L	R/W	–	√	√	0000H
FF13H			–			–		
FF14H	16 位比较寄存器 1	IT20 CM1	IT20 CM1L	R/W	–	√	√	0000H
FF15H			–			–		
FF16H	16 位定时器计数器 00	TM00		R	–	–	√	0000H
FF17H								
FF18H	接收缓冲寄存器 0	RXB00		R	–	√	–	FFH
FF19H	发送移位寄存器 0	TXS00		W	–	√	–	FFH
FF1AH	A/D 转换结果寄存器	ADCR		R	–	–	√	不确定
FF1BH								
FF1FH	串行 I/O 移位寄存器 10	SIO10		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
FF23H	端口模式寄存器 3	PM3		R/W	√	√	–	FFH
FF24H	端口模式寄存器 4	PM4		R/W	√	√	–	FFH
FF25H	端口模式寄存器 5	PM5		R/W	√	√	–	FFH
FF26H	端口模式寄存器 6	PM6		R/W	√	√	–	FFH
FF27H	端口模式寄存器 7	PM7		R/W	√	√	–	FFH

表 3-3. 特殊功能寄存器列表 (2 / 5)

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	
FF28H	直流控制寄存器 00	DCCTL00		R/W	√	√	-	00H
FF2AH	8 位定时器 H 模式寄存器 0	TMHMD0		R/W	√	√	-	00H
FF2CH	8 位定时器计数器 50	TM50		R	-	√	-	00H
FF2DH	8 位定时器比较寄存器 50	CR50		R/W	-	√	-	00H
FF2EH	定时器时钟选择寄存器 50	TCL50		R/W	-	√	-	00H
FF2FH	8 位定时器模式控制寄存器 50	TMC50		R/W	√	√	-	00H
FF30H	上拉电阻选项寄存器 0	PU0		R/W	√	√	-	00H
FF31H	上拉电阻选项寄存器 1	PU1		R/W	√	√	-	00H
FF33H	上拉电阻选项寄存器 3	PU3		R/W	√	√	-	00H
FF34H	上拉电阻选项寄存器 4	PU4		R/W	√	√	-	00H
FF35H	上拉电阻选项寄存器 5	PU5		R/W	√	√	-	00H
FF36H	上拉电阻选项寄存器 6	PU6		R/W	√	√	-	00H
FF37H	上拉电阻选项寄存器 7	PU7		R/W	√	√	-	00H
FF38H	直流控制寄存器 01	DCCTL01		R/W	√	√	-	00H
FF3AH	预分频模式寄存器	IT20PRM		R/W	√	√	-	07H
FF3BH	状态寄存器	IT20STS		R	√	√	-	00H
FF3CH	8 位定时器计数器 51	TM51		R	-	√	-	00H
FF3DH	8 位定时器比较寄存器 51	CR51		R/W	-	√	-	00H
FF3EH	定时器时钟选择寄存器 51	TCL51		R/W	-	√	-	00H
FF3FH	8 位定时器模式控制寄存器 51	TMC51		R/W	√	√	-	00H
FF40H	时钟输出选择寄存器	CKS		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
FF50H	10 位缓存寄存器 4	TW0BF CM4	TW0BF CM4L	R/W	-	√	√	0000H
FF51H			-					
FF52H	10 位缓存寄存器 5	TW0BF CM5	TW0BF CM5L	R/W	-	√	√	0000H
FF53H			-					
FF54H	10 位比较寄存器 0	TW0CM0		R/W	-	-	√	0000H
FF55H								
FF56H	10 位比较寄存器 1	TW0CM1		R/W	-	-	√	0000H
FF57H								
FF58H	10 位比较寄存器 2	TW0CM2		R/W	-	-	√	0000H
FF59H								
FF5AH	10 位比较寄存器 3	TW0CM3		R/W	-	-	√	00FFH
FF5BH								
FF5CH	10 位比较寄存器 4	TW0CM4		R/W	-	-	√	0000H
FF5DH								
FF5EH	10 位比较寄存器 5	TW0CM5		R/W	-	-	√	0000H
FF5FH								

表 3-3. 特殊功能寄存器列表 (3 / 5)

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	
FF60H	余数数据寄存器 0	SDR0	SDR0L	R	-	√	√	00H
FF61H			SDR0H		-	√		00H
FF62H	乘法 / 除法数据寄存器 A0	MDA0L	MDA0LL	R/W	-	√	√	00H
FF63H			MDA0LH		-	√		00H
FF64H		MDA0H	MDA0HL	R/W	-	√	√	00H
FF65H			MDA0HH		-	√		00H
FF66H	乘法 / 除法数据寄存器 B0	MDB0	MDB0L	R/W	-	√	√	00H
FF67H			MDB0H		-	√		00H
FF68H	乘法器 / 除法器控制寄存器 0	DMUC0		R/W	√	√	-	00H
FF6AH	捕获 / 比较控制寄存器 00	CRC00		R/W	√	√	-	00H
FF6BH	16 位定时器输出控制寄存器 00	TOC00		R/W	√	√	-	00H
FF6CH	A/D 转换器模式寄存器	ADM		R/W	√	√	-	00H
FF6DH	模拟输入通道指定寄存器	ADS		R/W	√	√	-	00H
FF6EH	掉电比较模式寄存器	PFM		R/W	√	√	-	00H
FF6FH	掉电比较门限寄存器	PFT		R/W	-	√	-	00H
FF70H	异步串行接口操作模式寄存器 00	ASIM00		R/W	√	√	-	01H
FF71H	波特率发生器控制寄存器 00	BRGC00		R/W	-	√	-	1FH
FF73H	异步串行接口接收错误状态寄存器 00	ASIS00		R	-	√	-	00H
FF78H	低电压检测寄存器	LVIM		R/W	√	√	-	00H [*]
FF7AH	16 位定时器捕获 / 比较寄存器 00	CR00		R/W	-	-	√	0000H
FF7BH								
FF7CH	16 位定时器捕获 / 比较寄存器 01	CR01		R/W	-	-	√	0000H
FF7DH								
FF7EH	16 位定时器模式控制寄存器 00	TMC00		R/W	√	√	-	00H
FF7FH	预分频模式寄存器 00	PRM00		R/W	√	√	-	00H
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	-	√	-	不确定
FF88H	逆变器定时器控制寄存器	TW0C		R/W	√	√	-	00H
FF89H	逆变器定时器模式寄存器	TW0M		R/W	√	√	-	00H
FF8AH	死区时间重新加载寄存器	TW0DTIME		R/W	-	√	-	FFH
FF8BH	A/D 触发选择寄存器	TW0TRGS		R/W	√	√	-	00H
FF8CH	逆变器定时器输出控制寄存器	TW0OC		R/W	√	√	-	00H

注 只有在LVI复位后，该值为83H。

表 3-3. 特殊功能寄存器列表 (4 / 5)

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	
FF90H	16 位定时器捕获 / 比较寄存器 0	IT20C C0	IT20C C0L	R/W	-	√	√	0000H
FF91H						-	-	
FF92H	16 位定时器捕获 / 比较寄存器 1	IT20C C1	IT20C C1L	R/W	-	√	√	0000H
FF93H						-	-	
FF94H	捕获 / 比较控制寄存器	IT20CCR		R/W	√	√	-	00H
FF95H	定时器单元模式寄存器	IT20TUM		R/W	√	√	-	00H
FF96H	定时器控制寄存器	IT20TMC		R/W	√	√	-	00H
FF97H	有效沿选择寄存器	IT20SESA		R/W	√	√	-	00H
FF98H	看门狗定时器模式寄存器	WDTM		R/W	-	√	-	67H
FF99H	看门狗定时器允许寄存器	WDTE		R/W	-	√	-	9AH
FF9EH	8 位定时器 H 比较寄存器 00	CMP00		R/W	-	√	-	00H
FF9FH	8 位定时器 H 比较寄存器 01	CMP01		R/W	-	√	-	00H
FFA0H	内部振荡模式寄存器	RCM		R/W	√	√	-	00H
FFA1H	主时钟模式寄存器	MCM		R/W	√	√	-	00H
FFA2H	主 OSC 控制寄存器	MOC		R/W	√	√	-	00H
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
FFAAH	噪声消除时间选择寄存器	NRC1		R/W	√	√	-	00H
FFACH	复位控制标志寄存器	RESF		R	-	√	-	00H ^{※1}
FFB0H	实时输出缓存寄存器 0L	RTBL00		R/W	√	√	-	00H
FFB2H	实时输出缓存寄存器 0H	RTBH00		R/W	√	√	-	00H
FFB4H	实时输出端口模式寄存器 0	RTPM00		R/W	√	√	-	00H
FFB5H	实时输出端口控制寄存器 0	RTPC00		R/W	√	√	-	00H
FFB8H	实时输出缓存寄存器 1L	RTBL01		R/W	√	√	-	00H
FFBAH	实时输出缓存寄存器 1H	RTBH01		R/W	√	√	-	00H
FFBCH	实时输出端口模式寄存器 1	RTPM01		R/W	√	√	-	00H
FFBDH	实时输出端口控制寄存器 1	RTPC01		R/W	√	√	-	00H
FFC0H	Flash 保护命令寄存器	PFCMD		W	-	√	-	不确定
FFC2H	Flash 状态寄存器	PFS		R/W	√	√	-	00H
FFC4H	Flash 编程模式控制寄存器	FLPMC		R/W	√	√	-	0XH ^{※2}

- 注
1. 该值根据复位源而改变。
 2. 根据操作模式而不同。
 - 用户模式: 08H
 - 板上模式: 0CH

表 3-3. 特殊功能寄存器列表 (5 / 5)

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	
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	√	√		DFH
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	内部存储器大小切换寄存器 ^注	IMS		R/W	-	√	-	CFH
FFFBH	处理器时钟控制寄存器	PCC		R/W	√	√	-	00H
FFFDH	系统等待控制寄存器	VSWC		R/W	√	√	-	00H

注 因为内部存储器大小切换寄存器 (IMS) 的初始值为 CFH, 通过初始化设置其为 C8H。

3.3 指令地址寻址

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

3.3.1 相对寻址

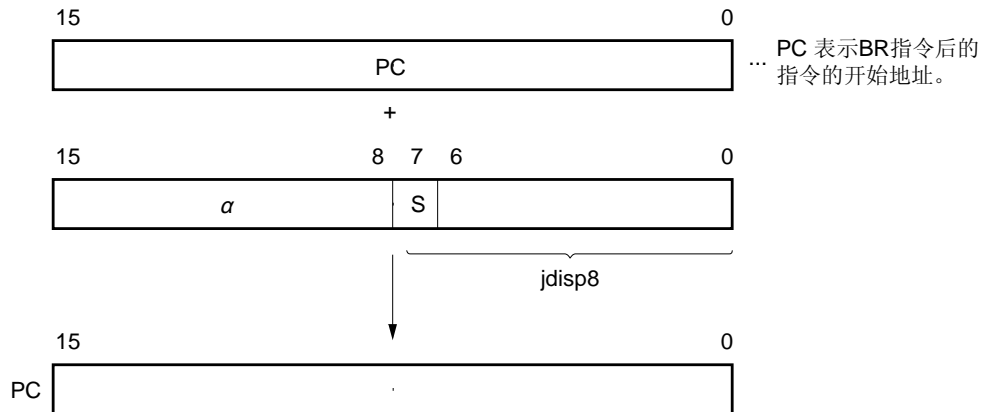
[功能]

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

换句话说，相对寻址是由从下一条指令的开始地址开始，范围在-128 到 +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，**α** 的所有位为 0。
 当 **S = 1** 时，**α** 的所有位为 1。

3.3.2 立即寻址

[功能]

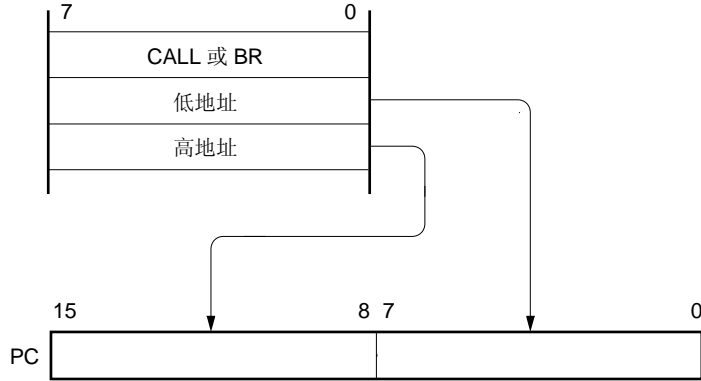
指令字中的立即数据被传送到程序计数器（PC）中并转向该地址。

该功能可以在执行 CALL !addr16 或 BR !addr16 或 CALLF !addr11 指令时实现。

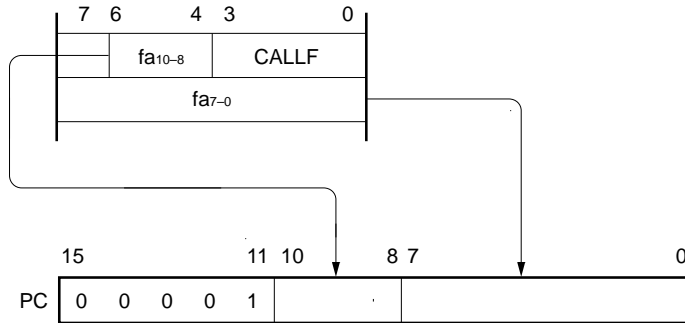
CALL !addr16 和 BR !addr16 指令可以跳转到整个存储器空间中。CALLF !addr11 则被转移到 0800H 到 0FFFH 的区域中。

[图示]

在 CALL !addr16 和 BR !addr16 指令的情况下



在 CALLF !addr11 指令的情况下



<R> 3.3.3 表间接寻址

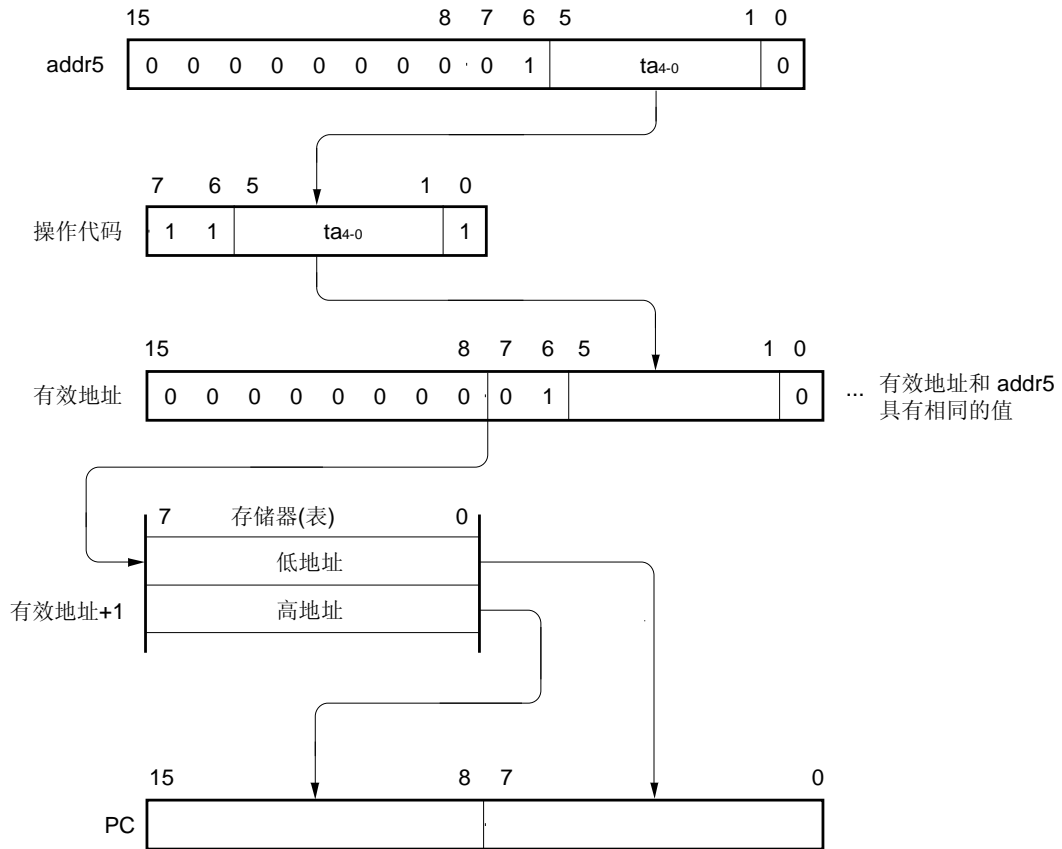
[功能]

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

该功能在执行 CALLT [addr5]指令时实现。

该指令涉及到由 addr5 表示的并且保存在范围从 0040H 到 007FH 的存储器表的地址中，并且允许跳转至整个存储器空间。

[图示]

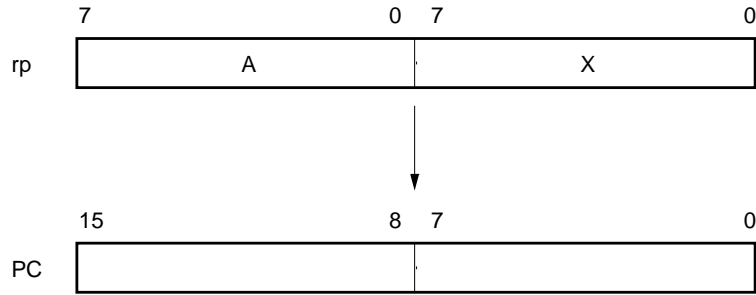


3.3.4 寄存器寻址

[功能]

通过指令字指定的寄存器对 (AX) 内容可以被赋值给程序计数器 (PC) 并且转向该地址。
该功能在执行 BR AX 指令时实现。

[图示]



3.4 操作数地址寻址

可以通过以下方法来指定寄存器和存储器（寻址）用以实现指令执行过程中的操作。

3.4.1 隐含寻址

[功能]

在通用寄存器中用作累加器（A 和 AX）的寄存器会自动（隐含）寻址。

在 μ PD78F0714 指令字中，以下指令使用隐含寻址。

指令	通过隐含寻址指令的寄存器
MULU	A 寄存器用于存放被乘数而 AX 寄存器用于存放运算结果。
DIVUW	AX 寄存器用于存放被除数和商。
ADJBA/ADJBS	A 寄存器用于存放那些要进行十进制校正的数字。
ROR4/ROL4	A 寄存器用于存放进行数字循环的数字数据。

[操作数格式]

由于隐含寻址可以通过指令自动进行，因此不需要特殊的操作数格式。

[说明示例]

在 MULU X 的情况下

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

3.4.2 寄存器寻址

[功能]

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

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

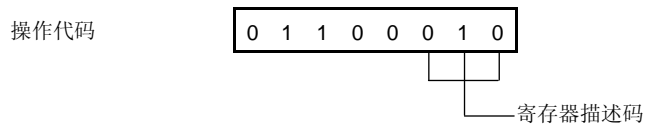
[操作数格式]

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

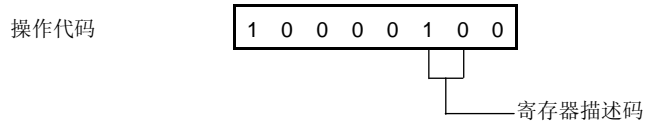
‘r’ 和 ‘rp’ 可以使用绝对名称（R0 到 R7 以及 RP0 到 RP3）进行描述，也可以使用功能名称（X, A, C, B, E, D, L, H, AX, BC, DE 和 HL）来进行描述。

[说明示例]

MOV A, C; 将 C 寄存器选择为 r 时



INCW DE; 将 DE 寄存器对选作 rp 时



3.4.3 直接寻址

[功能]

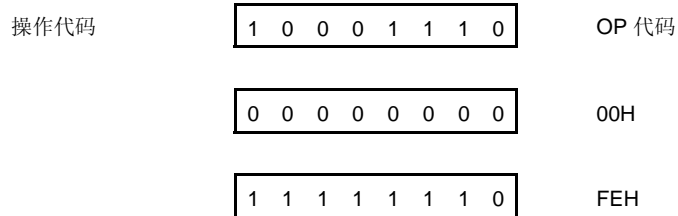
要进行操作的存储器通过变为操作数地址的指令字中的立即数直接进行寻址。

[操作数格式]

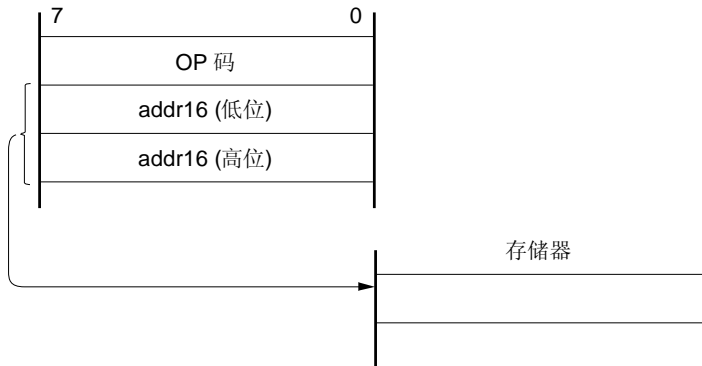
标识符	说明
addr16	标签或 16 位立即数据

[说明示例]

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



[图示]



3.4.4 短直接寻址

[功能]

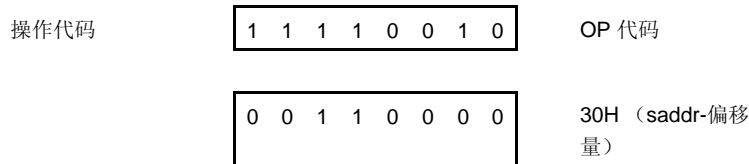
在固定空间中要进行操作的存储空间可以通过指令字中的 8 位数据直接进行访问。
 该寻址方式应用于 256 字节的空间 FE20H 到 FF1FH 中。内部 RAM 和特殊功能寄存器 (SFR) 分别映射在 FE20H 到 FEFFH 以及 FF00H 到 FF1FH 的区域。
 采用短直接寻址的 SFR 区域 (FF00H 到 FF1FH) 是整个 SFR 区域中的一部分。在程序中经常访问的端口和定时器 / 事件计数的比较和捕获寄存器映射在该区域中, 使 SFR 可以使用较少的字节和时钟进行操作。
 当 8 位立即数据在 20H 到 FFH 内时, 有效地址的第 8 位将会清除为 0。当在 00H 到 1FH 内时, 有效地址的第 8 位则将会被设置为 1。参见[图示]。

[操作数格式]

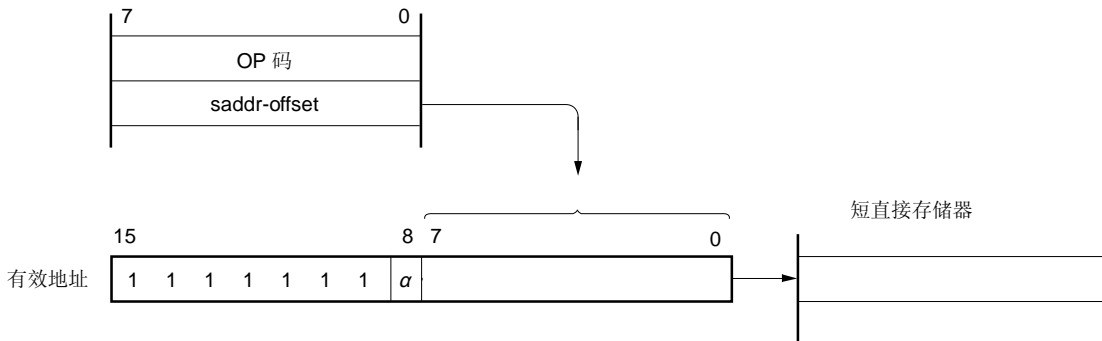
标识符	说明
saddr	表示标签或 FE20H 到 FF1FH 的立即数据
saddrp	表示标签或 FE20H 到 FF1EH 的立即数据 (仅限偶地址)

[说明示例]

MOV 0FE30H, A; 当传输 A 寄存器的值到 saddr (FE30H) 时



[图示]



当 8 位立即数据在 20H 到 FFH 间时, $\alpha = 0$
 当 8 位立即数据在 00H 到 1FH 间时, $\alpha = 1$

3.4.5 特殊功能寄存器 (SFR) 寻址

[功能]

通过指令字中的 8 位立即数据对存储器映射特殊功能寄存器 (SFR) 进行寻址。

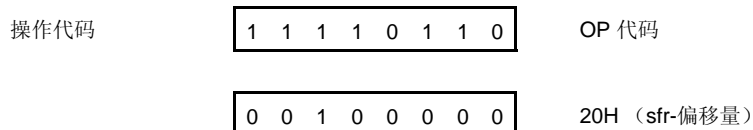
该寻址方式应用于 FF00H 到 FFCFH 以及 FFE0H 到 FFFFH 的 240 个字节的空间内。然而，映射在 FF00H 到 FF1FH 内的 SFR 可以通过短直接寻址方式进行访问。

[操作数格式]

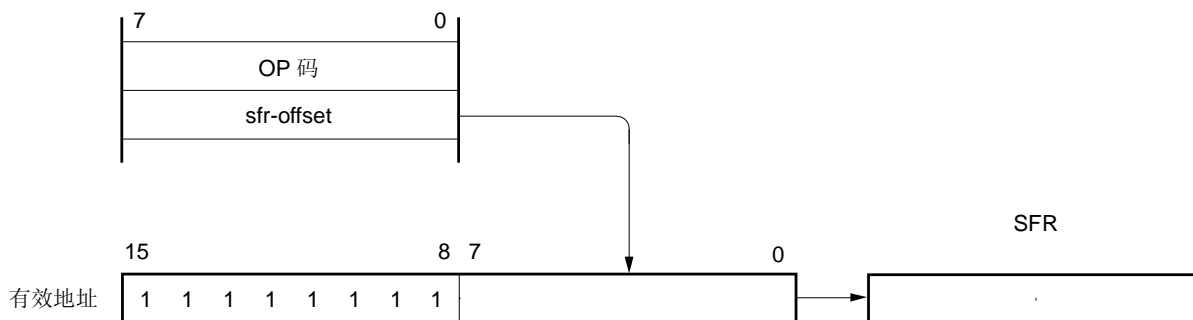
标识符	说明
sfr	特殊功能寄存器名
sfrp	16 位可操作特殊功能寄存器名 (仅限偶地址)

[说明示例]

MOV PM0, A; 当 PM0 (FF20H) 被选作 sfr 时



[图示]



3.4.6 寄存器间接寻址

[功能]

由指令字中的寄存器对指定代码以及寄存器 bank 选择标志（RBS0 和 RBS1）指定的寄存器对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。

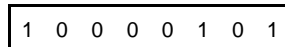
[操作数格式]

标识符	说明
-	[DE], [HL]

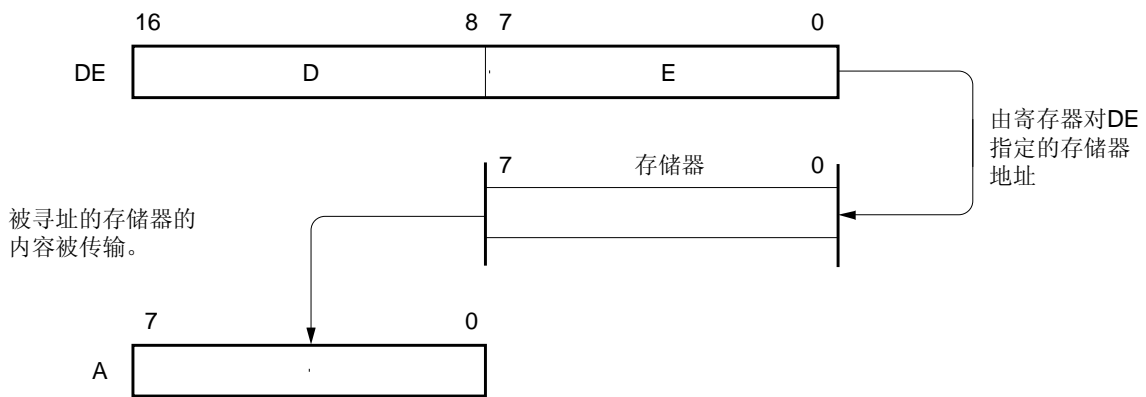
[说明示例]

MOV A, [DE]; 当[DE]被选作寄存器对时

操作代码



[图示]



3.4.7 基址寻址

[功能]

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

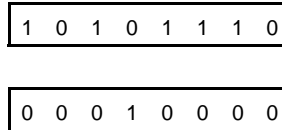
[操作数格式]

标识符	说明
-	[HL + byte]

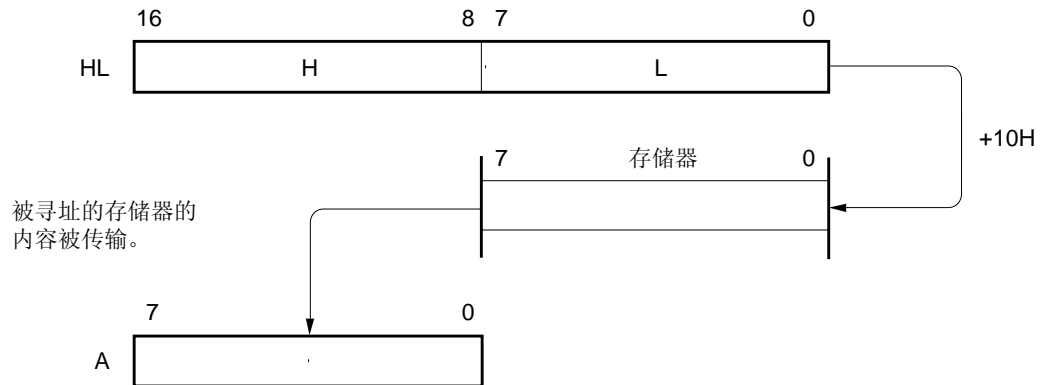
[说明示例]

MOV A, [HL + 10H]: 将字节设置为 10H 时

操作代码



[图示]



3.4.8 基址变址寻址

[功能]

在指令字中指令的 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 bank 中的 HL 寄存器对是由寄存器 bank 选择标志（RBS0 和 RBS1）指定的，并会根据相加结果对寄存器进行寻址。加法操作是通过将 B 或 C 寄存器内容作为正数扩展至 16 位来完成的。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。

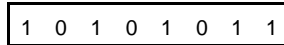
[操作数格式]

标识符	说明
-	[HL + B], [HL + C]

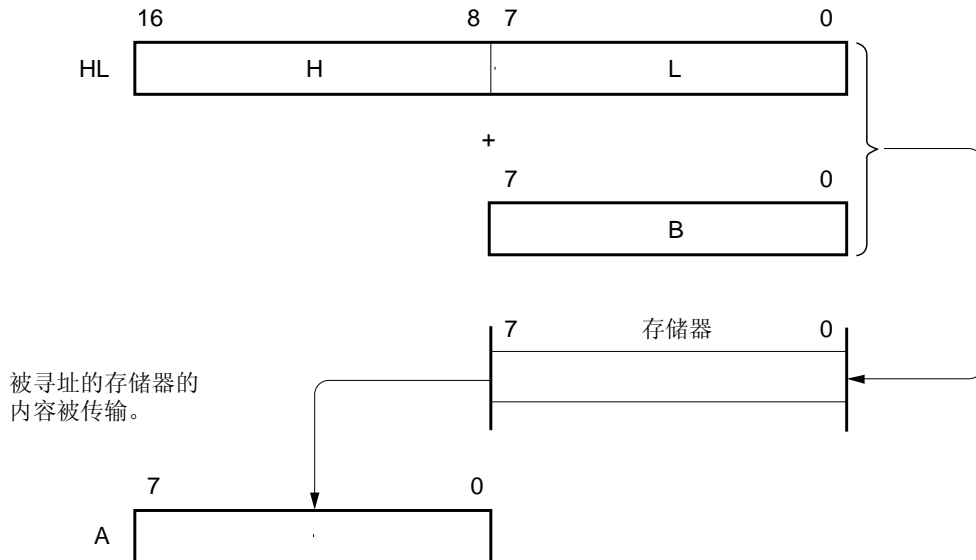
[说明示例]

在 MOV A, [HL + B]（选择 B 寄存器）的情况下

操作代码



[图示]



3.4.9 堆栈寻址

[功能]

堆栈区域通过堆栈指针（SP）的内容进行间接寻址。

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

使用堆栈寻址时，只能访问内部高速 RAM 区域。

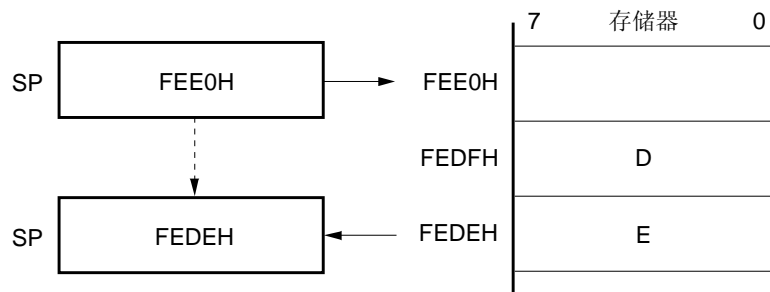
[说明示例]

在 PUSH DE（保存 DE 寄存器）的情况下

操作代码

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

[图示]



第四章 端口功能

4.1 端口功能

共有两种类型的引脚 I/O 缓存电源：AVREF 和 EVDD。这些供电电源和引脚间的关系如下所示。

表 4-1. 引脚 I/O 缓存电源

电源	对应引脚
AVREF	P20 到 P27 ^注
EVDD	P20 到 P27 以外的端口引脚

注 当端口 2 用作数字端口时，连接 AVREF 到 EVDD。

μ PD78F0714 产品提供的端口如图 4-1 中所示，这些端口支持多种控制操作。各个端口的功能如表 4-2 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情况，请参见第二章 引脚功能。

图 4-1. 端口类型

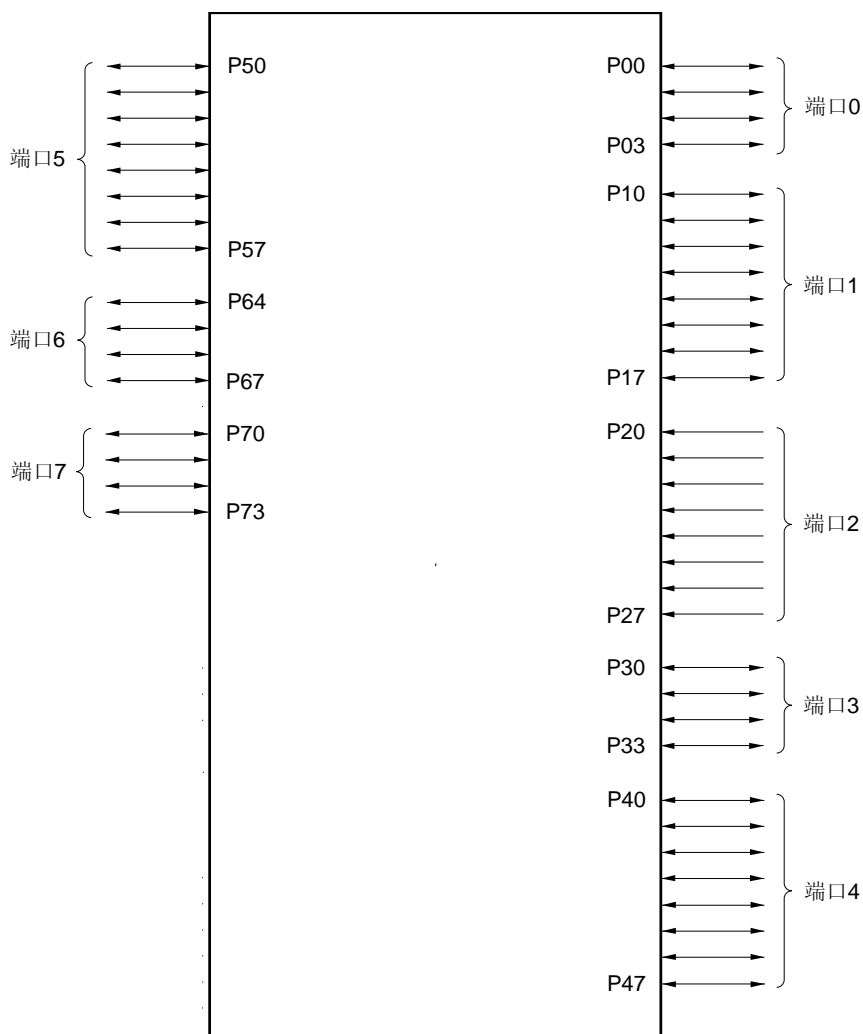


表 4-2. 端口功能

引脚名称	I/O	功能	复位后	复用功能
P00	I/O	端口 0。 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	INTP0/TW0TOFFP
P01				INTP1
P02				INTP2
P03				INTP3/ADTRG
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	-
P11				-
P12				-
P13				RxD00
P14				TxD00
P15				SCK10
P16				SI10
P17				SO10/FLMD1
P20 到 P27	输入	端口 2。 8 位输入端口。	输入	ANI0 到 ANI7
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	BUZ
P31				PCL
P32				-
P33				-
P40 到 P47	I/O	端口 4。 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	RTP00 到 RTP07
P50	I/O	端口 5。 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	TI50/TO50
P51				TI51/TO51
P52				TOH0/INTP4
P53				TI000/INTP5
P54				TI001/TO00
P55				TIT20IUD/INTP6
P56				TIT20CUD /TIT20CC0/INTP7
P57				TIT20CLR /TIT20CC1 /TIT20TO
P64 到 P67	I/O	端口 6。 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	-
P70 到 P73	I/O	端口 7。 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	-

4.2 端口配置

端口由以下硬件组成。

表 4-3. 端口配置

项目	配置
控制寄存器	端口模式寄存器 (PM0, PM1, PM3 到 PM7) 端口寄存器 (P0 到 P7) 上拉电阻选项寄存器 (PU0, PU1, PU3 到 PU7)
端口	总共: 48 (CMOS I/O: 40, CMOS 输入: 8)
上拉电阻	总共: 40 (软件控制: 40)

4.2.1 端口 0

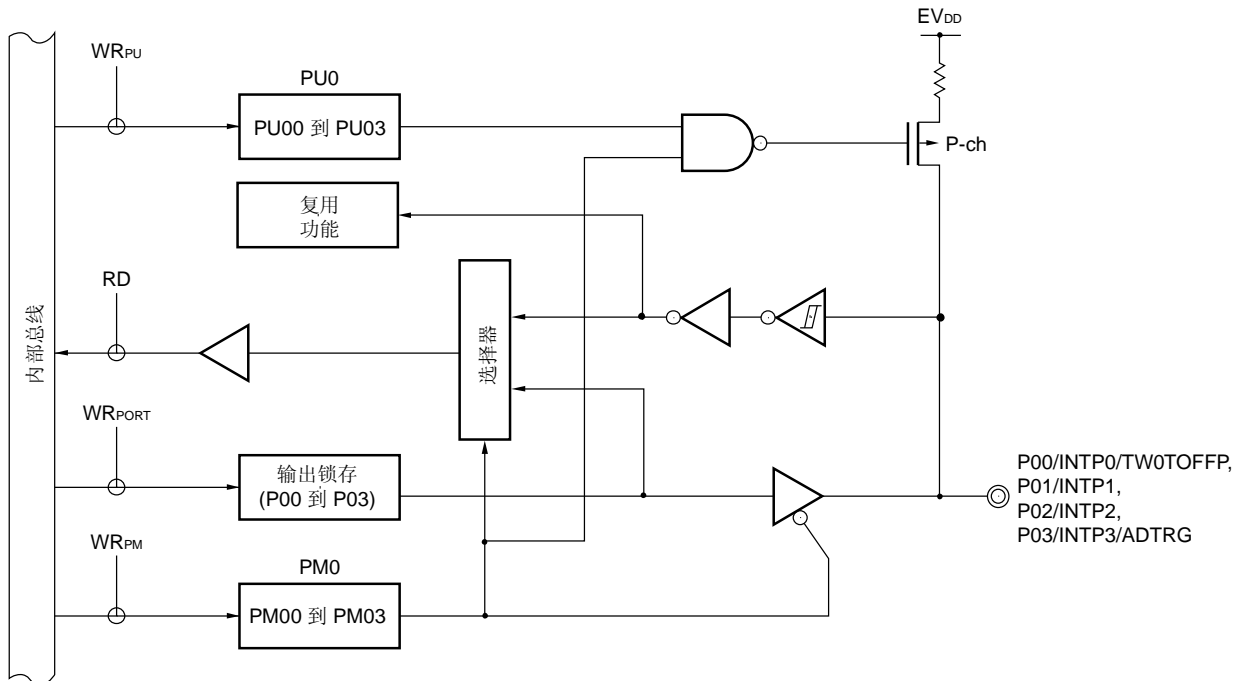
端口 0 是带有输出锁存的 4 位 I/O 端口。端口 0 可以通过使用端口模式寄存器 0 (PM0) 以 1 位为单位设置为输入模式或输出模式。当 P00 到 P03 引脚用作输入端口时，片上上拉电阻可以通过上拉电阻选项寄存器 0 (PU0) 进行指定。

该端口也用于外部中断请求输入、定时器输出停止外部信号和 A/D 转换器触发输入。

RESET 输入设置端口 0 为输入模式。

图 4-2 显示了端口 0 的框图。

图 4-2. P00 到 P03 的框图



PU0: 上拉电阻选项寄存器 0

PM0: 端口模式寄存器 0

RD: 读取信号

WR_{xx}: 写入信号

4.2.2 端口 1

端口 1 是带有输出锁存的 8 位 I/O 端口。端口 1 可以通过使用端口模式寄存器 1 (PM1) 以 1 位为单位设置为输入模式或输出模式。当 P10 到 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项寄存器 1 (PU1) 进行指定。

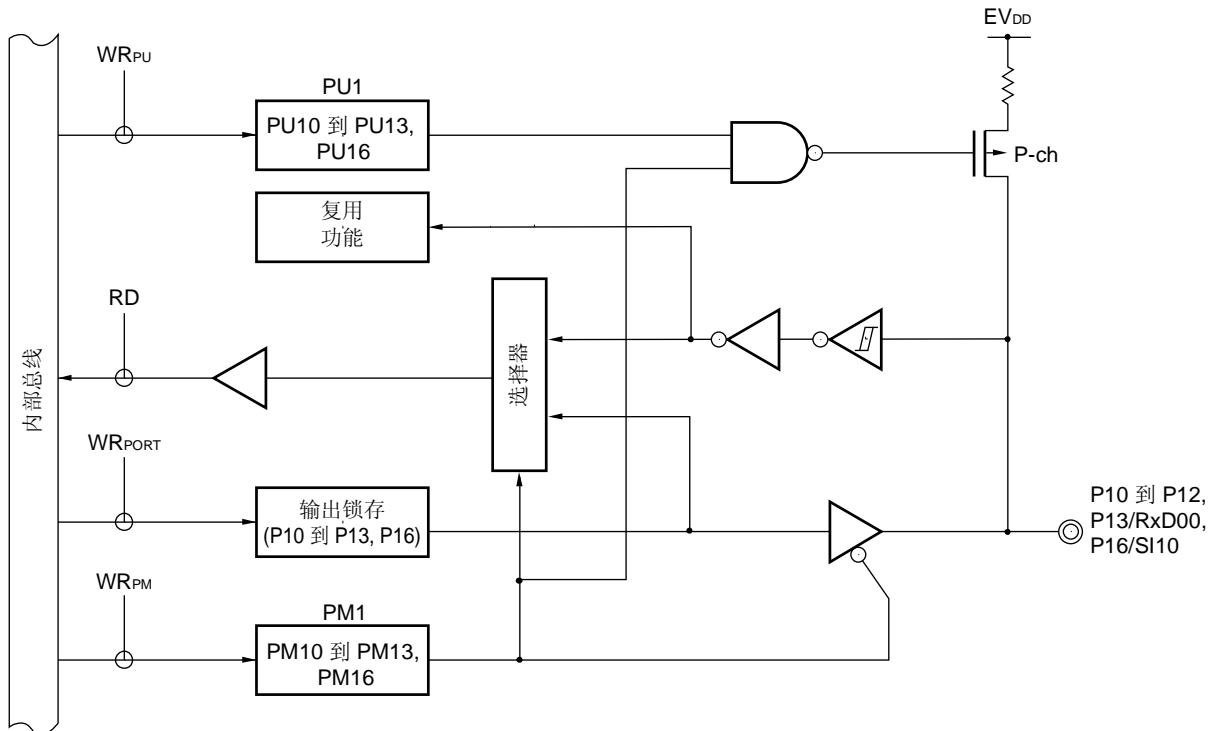
该端口也可用于串行接口数据 I/O、时钟 I/O 和 flash 存储器编程模式设置。

$\overline{\text{RESET}}$ 输入设置端口 1 为输入模式。

图 4-3 到 4-6 显示了端口 1 的框图。

注意事项 当 P15/ $\overline{\text{SCK10}}$ 、P16/SI10 和 P17/SO10 用作通用端口时，不要写串行时钟选择寄存器 10 (CSIC10)。

图 4-3. P10 到 P13 和 P16 的框图



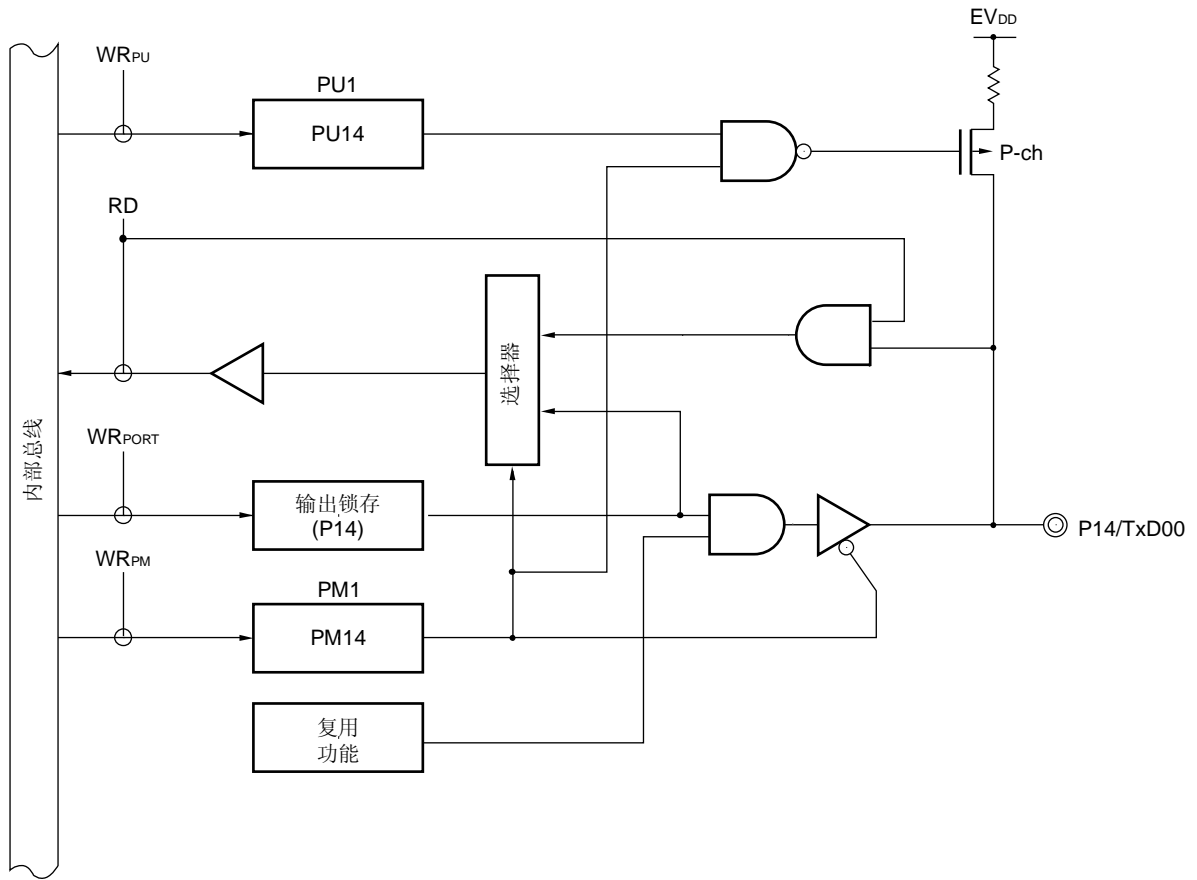
PU1: 上拉电阻选项寄存器 1

PM1: 端口模式寄存器 1

RD: 读取信号

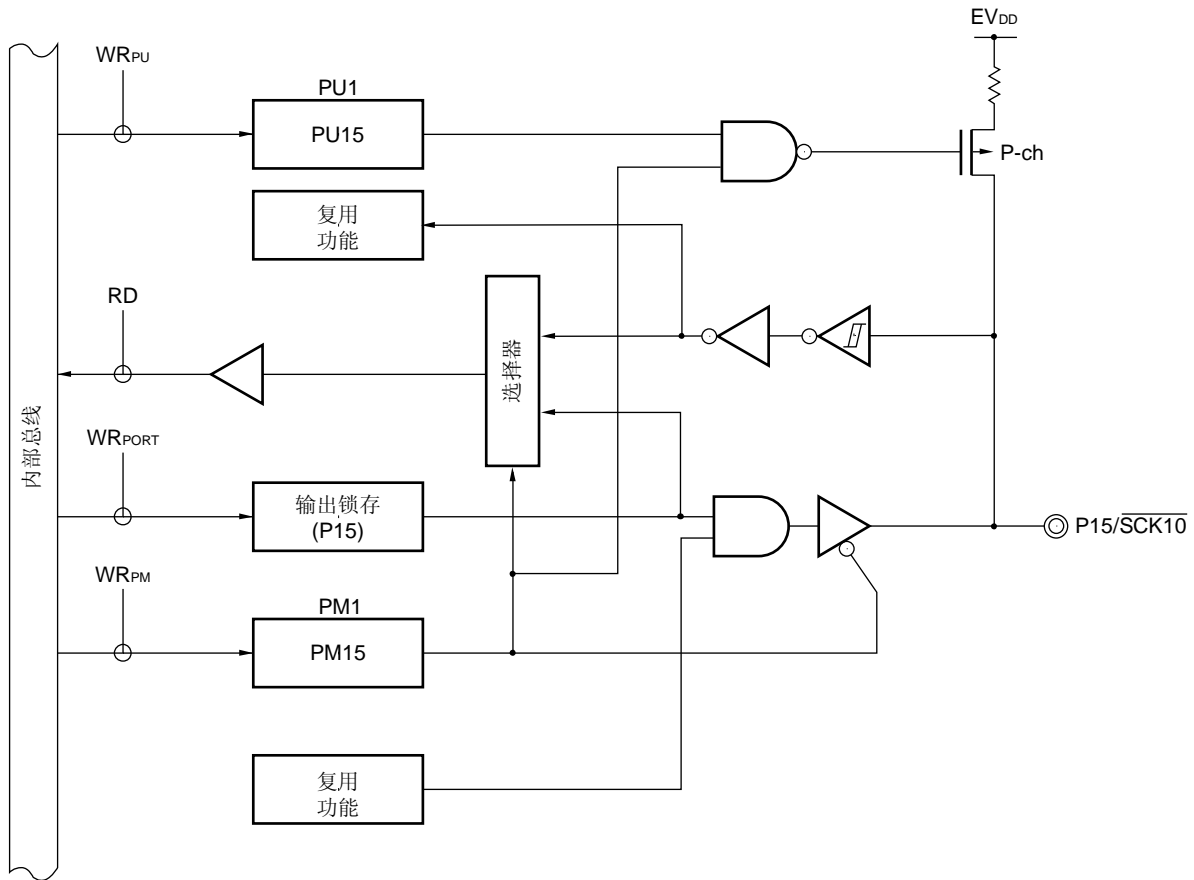
WR_{xx}: 写入信号

图 4-4. P14 的框图



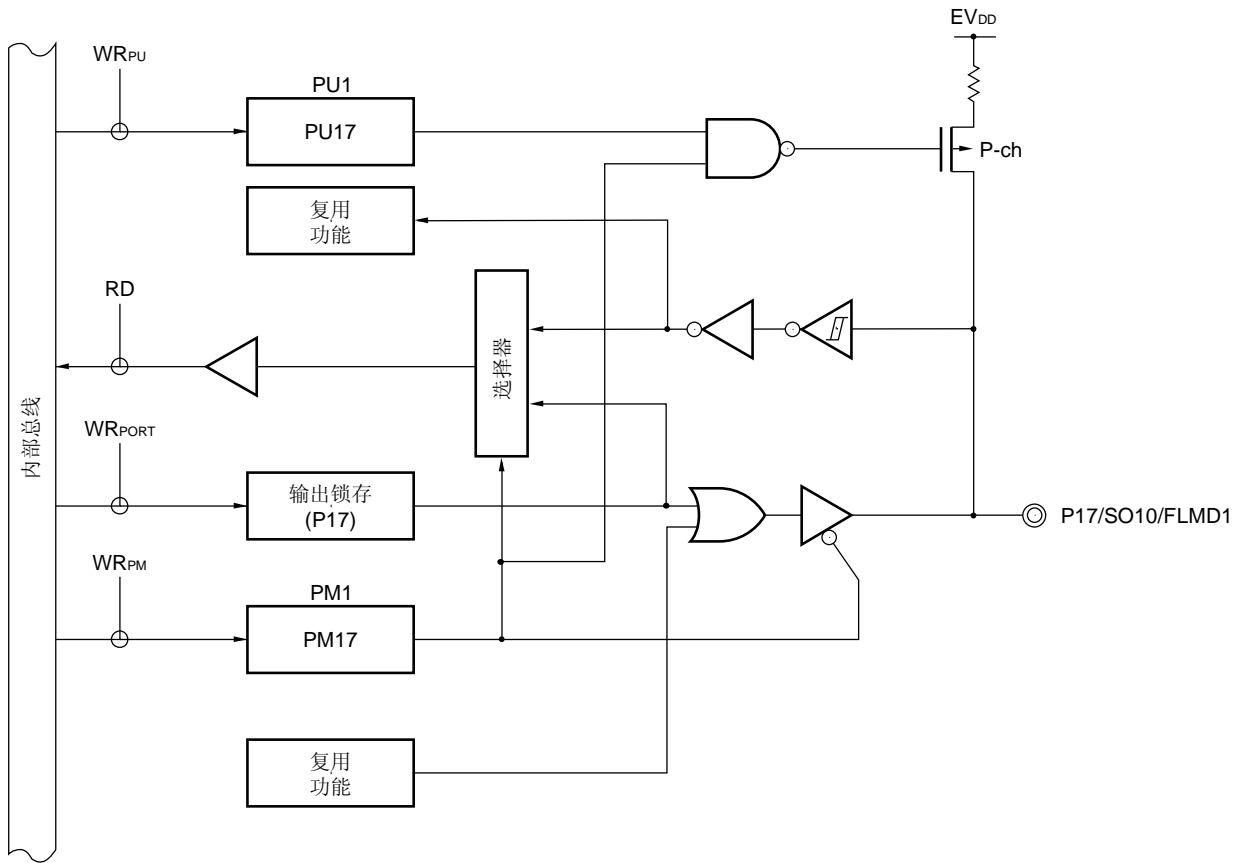
- PU1: 上拉电阻选项寄存器 1
- PM1: 端口模式寄存器 1
- RD: 读取信号
- WR_{xx}: 写入信号

图 4-5. P15 的框图



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

图 4-6. P17 的框图



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

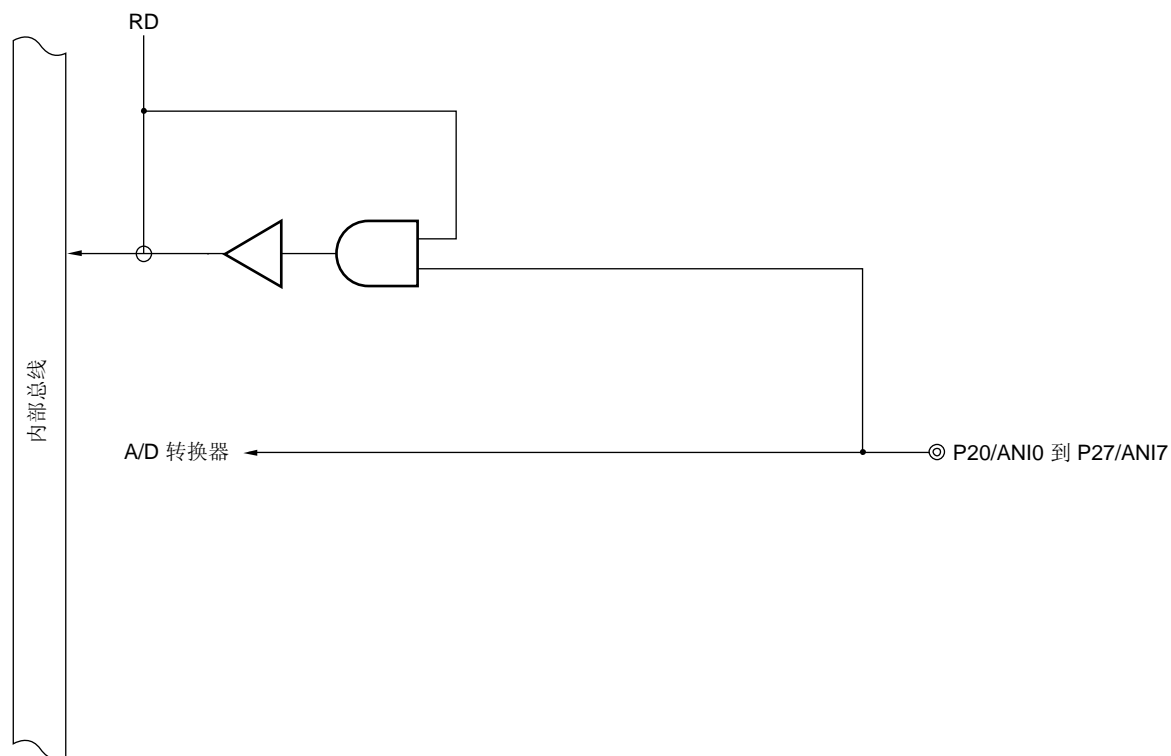
4.2.3 端口 2

端口 2 为 8 位输入端口。

该端口也可用于 A/D 转换器模拟输入。

图 4-7 显示了端口 2 的框图。

图 4-7. P20 到 P27 的框图



RD: 读取信号

<R> 注意事项 当以端口模式使用 P20 到 P27 时，需要 $EV_{DD} = AV_{REF}$ 。

4.2.4 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模式或输出模式。当用作输入端口时，可以通过上拉电阻选项寄存器 3 (PU3) 来指定片上上拉电阻的使用。

该端口也用于蜂鸣器输出和时钟输出。

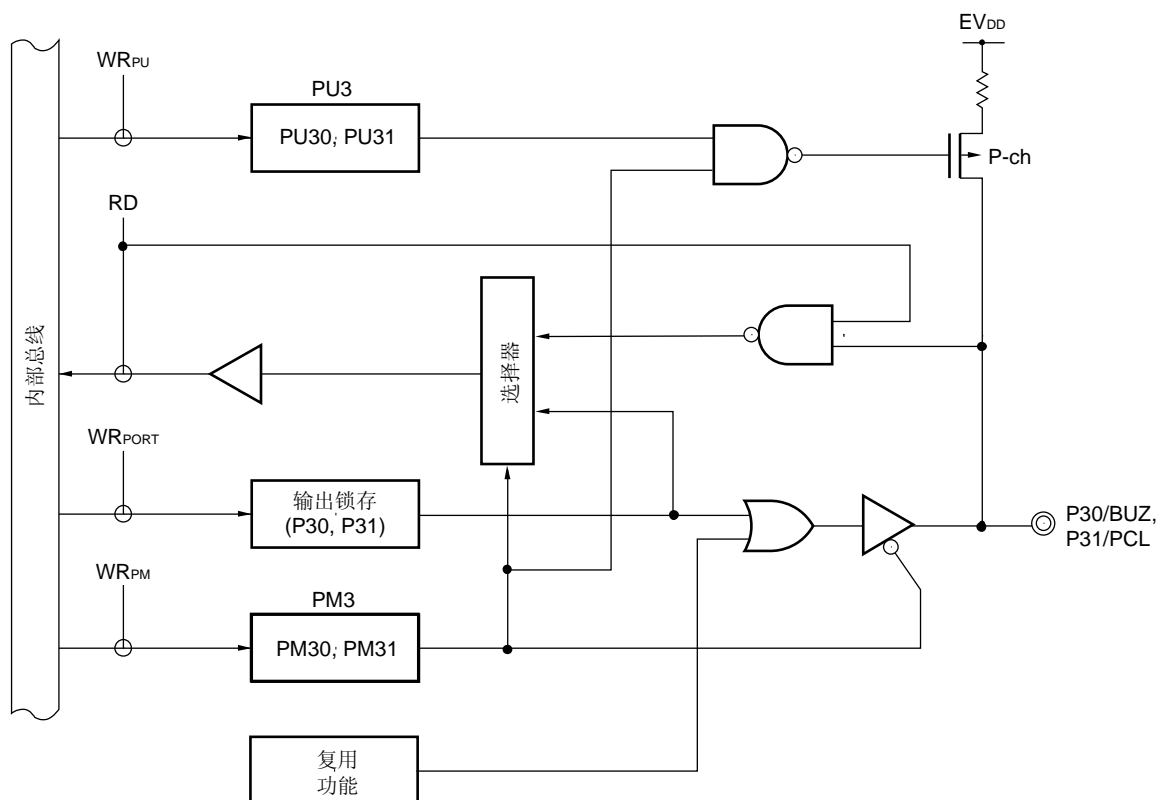
$\overline{\text{RESET}}$ 输入设置端口 3 为输入模式。

图 4-8 和 4-9 显示了端口 3 的框图。

注意事项 确保在复位后下拉 P31 以防止故障。

备注 $\mu\text{PD78F0714}$ 的 P31/INTP2 和 P32/INTP3 引脚可以在使用片上调试功能时设置片上调试模式。关于详细情况，参见第二十六章 片上调试功能。

图 4-8. P30 和 P31 的框图



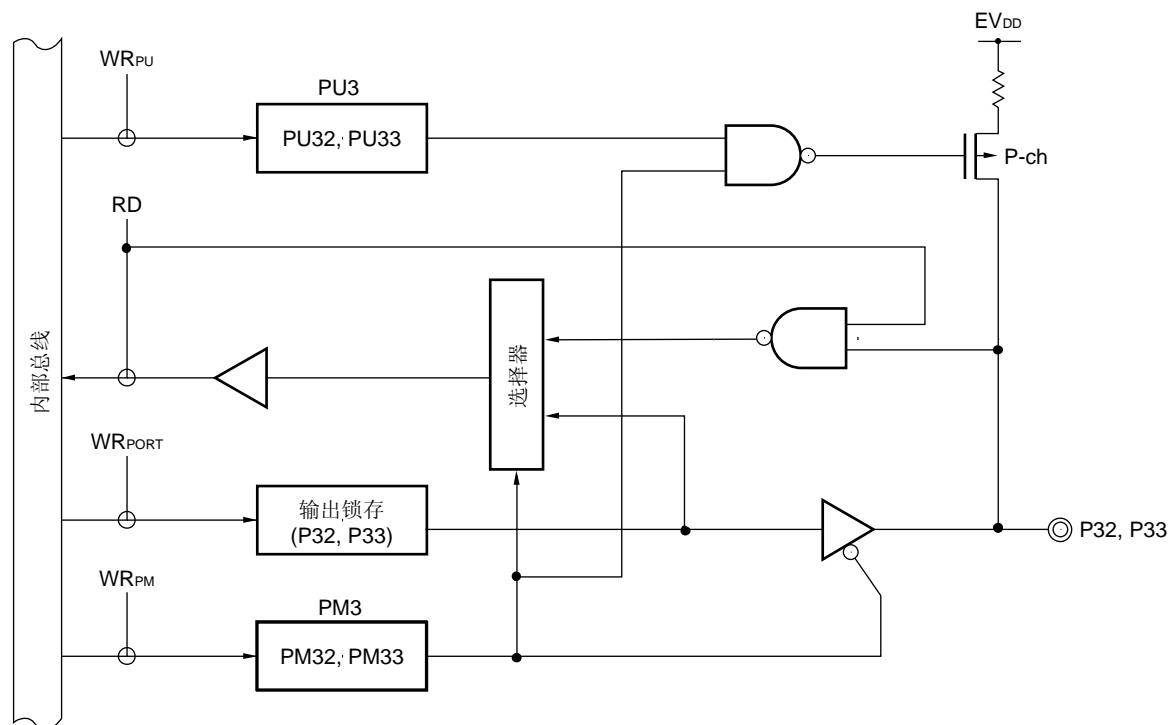
PU3: 上拉电阻选项寄存器 3

PM3: 端口模式寄存器 3

RD: 读取信号

WRxx: 写入信号

图 4-9. P32 和 P33 的框图



PU3: 上拉电阻选项寄存器 3

PM3: 端口模式寄存器 3

RD: 读取信号

WR_{xx}: 写入信号

4.2.5 端口 4

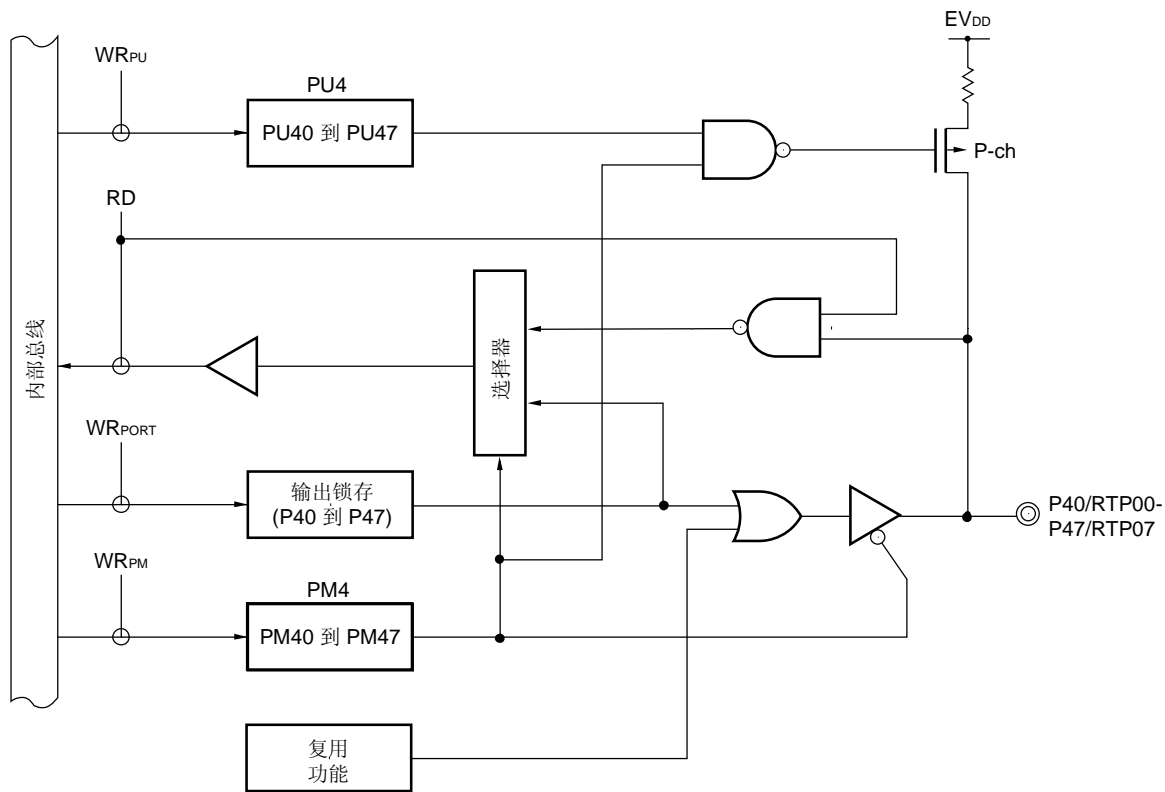
端口 4 是带有输出锁存的 8 位 I/O 端口。端口 4 可以通过使用端口模式寄存器 4 (PM4) 以 1 位为单位设置为输入模式或输出模式。片上上拉电阻的使用可以通过上拉电阻选项寄存器 4 (PU4) 以 1 位为单位来指定。

该端口也用作实时输出端口。

$\overline{\text{RESET}}$ 输入设置端口 4 为输入模式。

图 4-10 显示了端口 4 的框图。

图 4-10. P40 到 P47 的框图



- PU4: 上拉电阻选项寄存器 4
- PM4: 端口模式寄存器 4
- RD: 读取信号
- WR_{xx}: 写入信号

4.2.6 端口 5

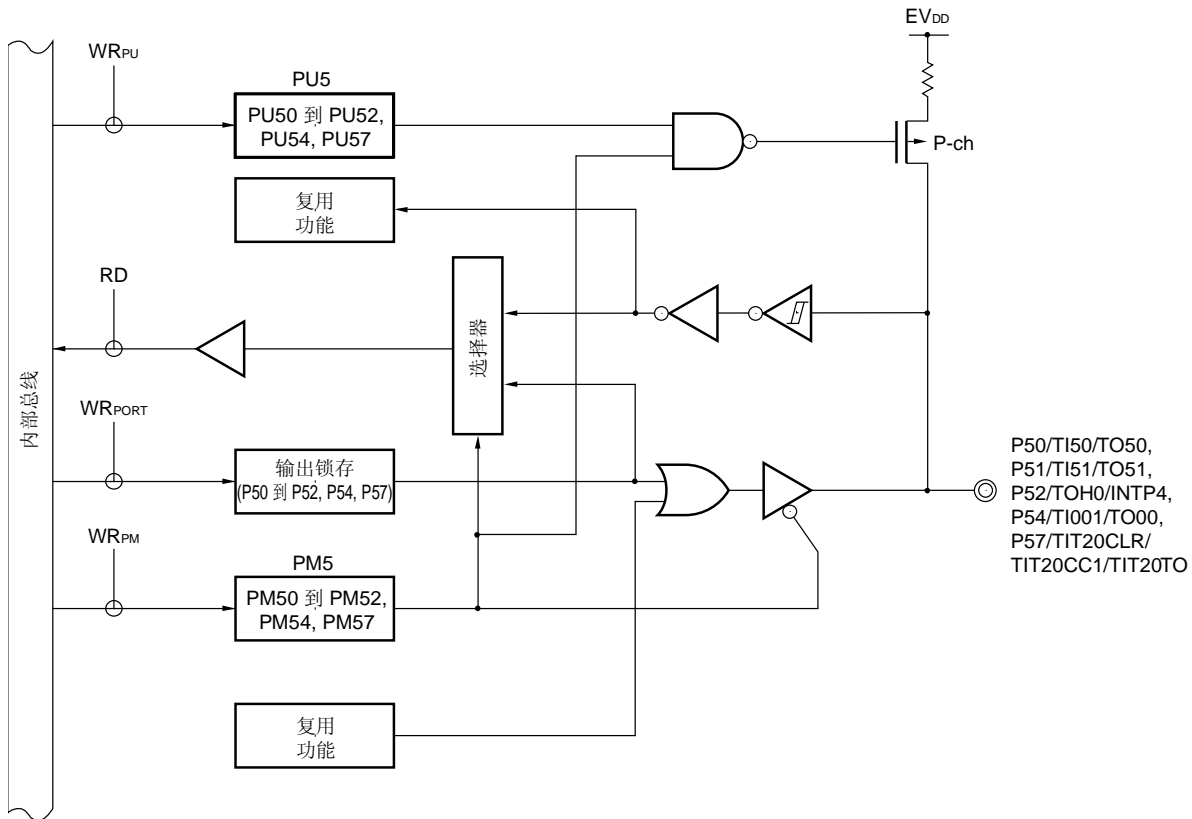
端口 5 是带有输出锁存的 8 位 I/O 端口。端口 5 可以通过使用端口模式寄存器 5 (PM5) 以 1 位为单位设置为输入模式或输出模式。片上上拉电阻的使用可以通过上拉电阻选项寄存器 5 (PU5) 以 1 位为单位来指定。

该端口也可用作外部中断请求输入以及定时器 I/O。

$\overline{\text{RESET}}$ 输入设置端口 5 为输入模式。

图 4-11 和 4-12 显示了端口 5 的框图。

图 4-11. P50 到 P52、P54 和 P57 的框图



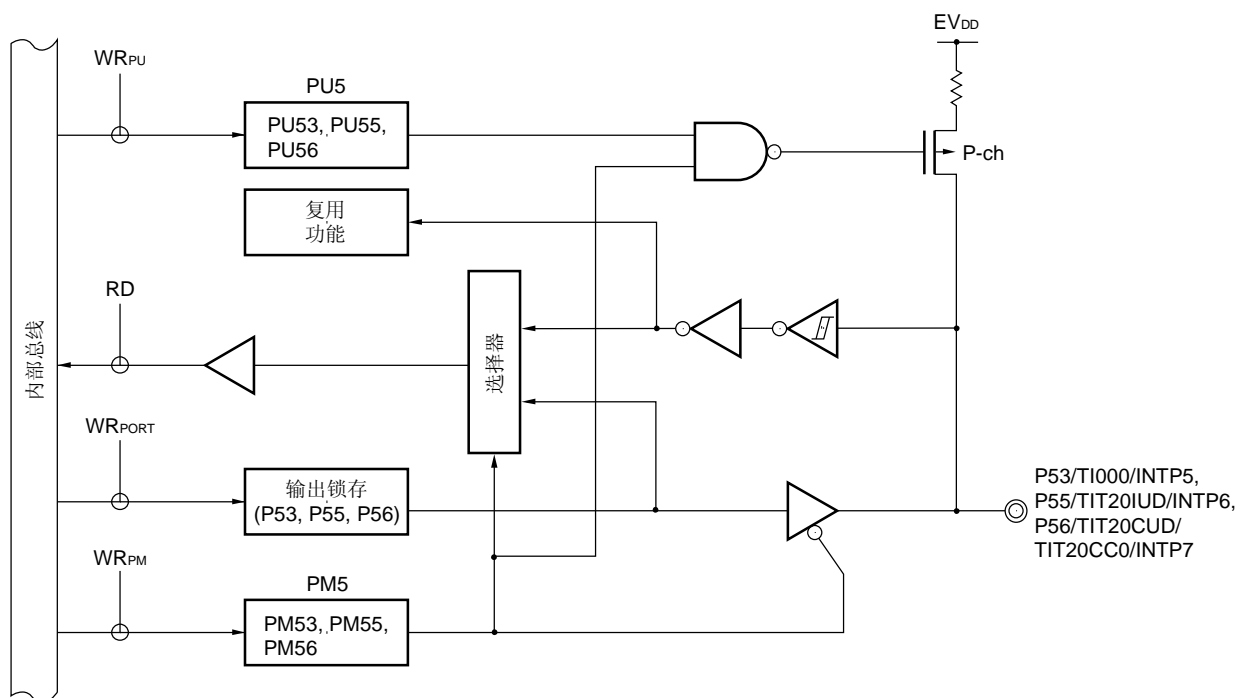
PU5: 上拉电阻选项寄存器 5

PM5: 端口模式寄存器 5

RD: 读取信号

WRxx: 写入信号

图 4-12. P53、P55 和 P56 的框图



PU5: 上拉电阻选项寄存器 5

PM5: 端口模式寄存器 5

RD: 读取信号

WR_{xx}: 写入信号

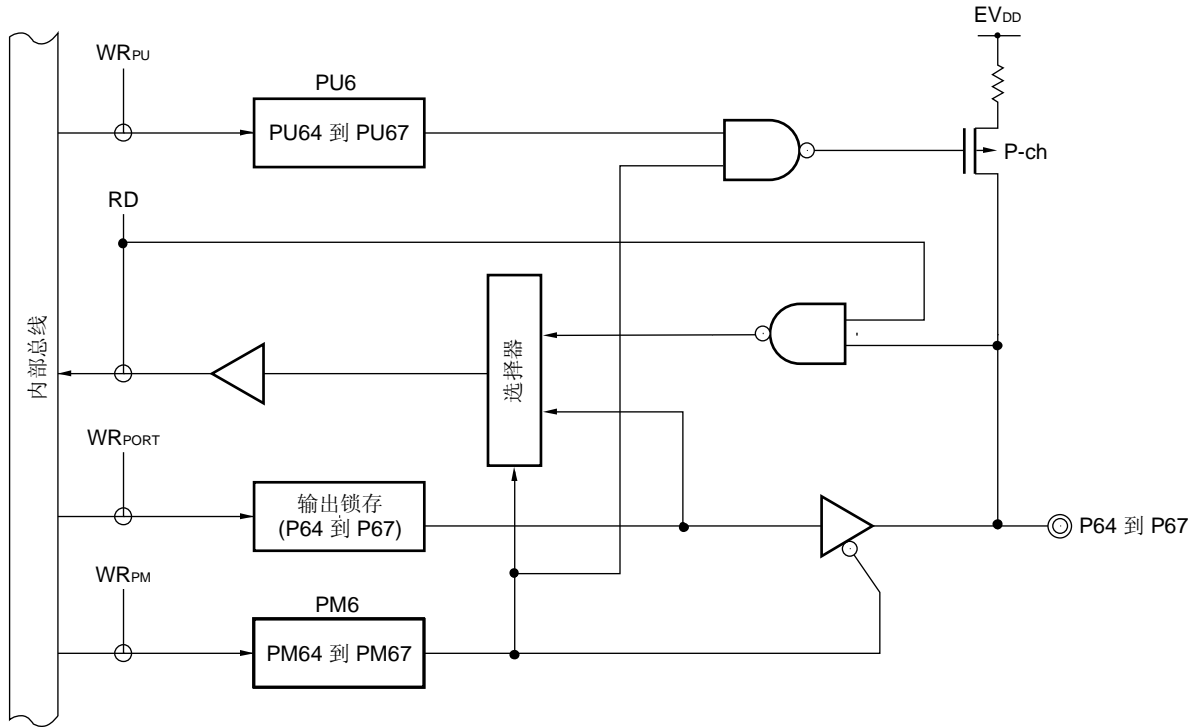
4.2.7 端口 6

端口 6 是带有输出锁存的 4 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。当用作输入端口时, 可以通过上拉电阻选项寄存器 6 (PU6) 来指定片上上拉电阻的使用。

$\overline{\text{RESET}}$ 输入设置端口 6 为输入模式。

图 4-13 显示了端口 6 的框图。

图 4-13. P64 到 P67 的框图



PU6: 上拉电阻选项寄存器 6

PM6: 端口模式寄存器 6

RD: 读取信号

WR_{xx} : 写入信号

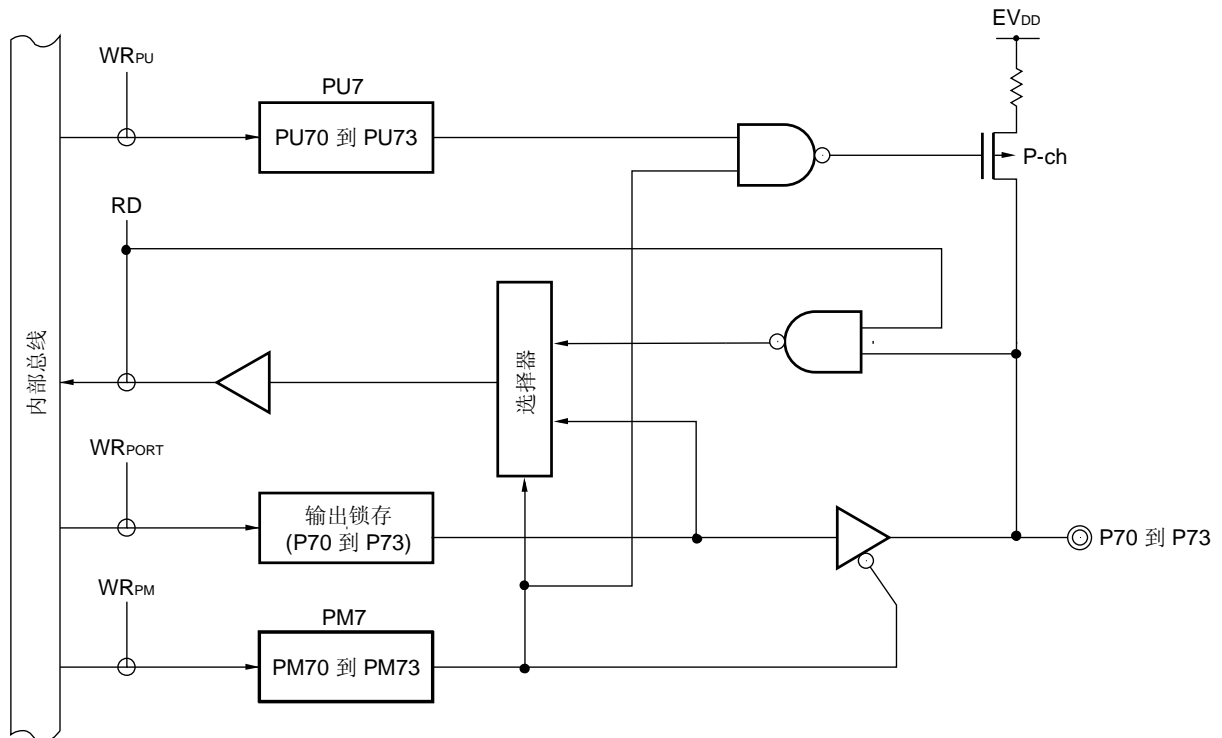
4.2.8 端口 7

端口 7 是带有输出锁存的 4 位 I/O 端口。端口 7 可以通过使用端口模式寄存器 7 (PM7) 以 1 位为单位设置为输入模式或输出模式。当 P70 到 P73 引脚用作输入端口时，片上上拉电阻可以通过上拉电阻选项寄存器 7 (PU7) 进行指定。

RESET 输入设置端口 7 为输入模式。

图 4-14 显示了端口 7 的框图。

图 4-14. P70 到 P73 的框图



PU7: 上拉电阻选项寄存器 7

PM7: 端口模式寄存器 7

RD: 读取信号

WR_{xx}: 写入信号

4.3 控制端口功能的寄存器

端口功能由以下三种类型的寄存器进行控制。

- 端口模式寄存器（PM0, PM1, PM3 到 PM7）
- 端口寄存器（P0 到 P7）
- 上拉电阻选项寄存器（PU0, PU1, PU3 到 PU7）

(1) 端口模式寄存器（PM0, PM1, PM3 到 PM7）

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。

RESET 输入设置这些寄存器为 FFH。

当端口引脚用作复用功能时，应按表 4-4 中所示对端口模式寄存器和输出锁存器进行设置。

图 4-15. 端口模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	1	1	1	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	1	1	1	1	FF26H	FFH	R/W
PM7	1	1	1	1	PM73	PM72	PM71	PM70	FF27H	FFH	R/W

PMmn	PMn 引脚 I/O 模式选择 (m = 0, 1, 3 到 7; n = 0 到 7)
0	输出模式 (输出缓存打开)
1	输入模式 (输出缓存关闭)

表 4-4. 使用复用功能时端口模式寄存器以及输出锁存的设置

引脚名称	复用功能		PM _{xx}	P _{xx}
	功能名称	I/O		
P00	INTP0	输入	1	×
	TW0TOFFP	输入	1	×
P01	INTP1	输入	1	×
P02	INTP2	输入	1	×
P03	INTP3	输入	1	×
	ADTRG	输入	1	×
P13	RxD00	输入	1	×
P14	TxD00	输出	0	1
P15	SCK10	输入	1	×
		输出	0	1
P16	SI10	输入	1	×
P17	SO10	输出	0	0
	FLMD1	输入	1	×
P20-P27	ANI0-ANI7	输入	1	×
P30	BUZ	输出	0	0
P31	PCL	输出	0	0
P40-P47	RTP00-RTP07	输出	0	0
P50	TI50	输入	1	×
	TO50	输出	0	0
P51	TI51	输入	1	×
	TO51	输出	0	0
P52	INTP4	输入	1	×
	TOH0	输出	0	0
P53	INTP5	输入	1	×
	TI000	输入	1	×
P54	TI001	输入	1	×
	TO00	输出	0	0
P55	INTP6	输入	1	×
	TIT20IUD	输入	1	×
P56	INTP7	输入	1	×
	TIT20CUD	输入	1	×
	TIT20CC0	输入	1	×
P57	TIT20CC1	输入	1	×
	TIT20CLR	输入	1	×
	TIT20TO	输出	0	0

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

(2) 端口寄存器 (P0 到 P7)

当数据从端口中输出时，这些寄存器用于写数据，数据是从芯片中输出的。
 如果在输入模式中读取数据，那么将会读取引脚电平。如果在输出模式中读取，则将会读取输出锁存的值。
 这些寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。
 $\overline{\text{RESET}}$ 输入清除这些寄存器为 00H (但是 P2 不确定)。

图 4-16. 端口寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
P0	0	0	0	0	P03	P02	P01	P00	FF00H	00H (输出锁存)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FF01H	00H (输出锁存)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FF02H	不确定	R
P3	0	0	0	0	P33	P32	P31	P30	FF03H	00H (输出锁存)	R/W
P4	P47	P46	P45	P44	P43	P42	P41	P40	FF04H	00H (输出锁存)	R/W
P5	P57	P56	P55	P54	P53	P52	P51	P50	FF05H	00H (输出锁存)	R/W
P6	P67	P66	P65	P64	0	0	0	0	FF06H	00H (输出锁存)	R/W
P7	0	0	0	0	P73	P72	P71	P70	FF07H	00H (输出锁存)	R/W

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

(3) 上拉电阻选项寄存器 (PU0, PU1 和 PU3 到 PU7)

这些寄存器指定 P00 到 P03、P10 到 P17、P30 到 P33、P40 到 P47、P50 到 P57、P64 到 P67 以及 P70 到 P73 的片上上拉电阻是否使用。片上上拉电阻可以以 1 位为单位用于那些设置到引脚输入模式中的位中，且对于这些引脚来说，片上上拉电阻的使用已经指定。不管 PU0，PU1 和 PU3 到 PU7 是何设置，片上上拉电阻都不能被连接到设置到输出模式中的位上以及用作复用功能输出引脚的位上。

这些寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。

RESET 输入清除这些寄存器为 00H。

图 4-17. 上拉电阻选项寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU0	0	0	0	0	PU03	PU02	PU01	PU00	FF30H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W
PU3	0	0	0	0	PU33	PU32	PU31	PU30	FF33H	00H	R/W
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	FF34H	00H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	FF35H	00H	R/W
PU6	PU67	PU66	PU65	PU64	0	0	0	0	FF36H	00H	R/W
PU7	0	0	0	0	PU73	PU72	PU71	PU70	FF37H	00H	R/W

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

4.4 端口功能的操作

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。

注意事项 在 1 位存储器操作指令的情况下，尽管单独一位被修改，但是端口仍以 8 位为单位被访问。因此，在输入和输出引脚混合的端口上，即使对于未修改的位，指定为输入的引脚的输出锁存内容也不确定。

4.4.1 写入 I/O 端口

(1) 输出模式

通过传送指令对输出锁存写入值，而输出锁存的内容则会从引脚中输出。

一旦数据被写入到输出锁存中，它将会被保留直到数据被再次写入到输出锁存中。

输出锁存的数据被复位清除。

(2) 输入模式

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

一旦数据被写入到输出锁存中，它将会被保留直到数据被再次写入到输出锁存中。

4.4.2 从 I/O 端口中读取

(1) 输出模式

通过传送指令读取输出锁存的内容。输出锁存的内容不会发生变化。

(2) 输入模式

通过传送指令读取引脚状态。输出锁存的内容不会发生变化。

4.4.3 I/O 端口上的操作

(1) 输出模式

操作是在输出锁存的内容上进行的，且操作结果会被写入到输出锁存中。输出锁存的内容会从引脚中输出。

一旦数据被写入到输出锁存中，它将会被保留直到数据被再次写入到输出锁存中。

输出锁存的数据被复位清除。

(2) 输入模式

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

<R> 4.5 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的值也可能被写入。

因此，当端口从输入模式切换到输出模式时，推荐重写输出锁存。

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

说明：PMnm 位 = 1 的端口中 Pn 寄存器的写 / 读对象是各自的输出锁存和引脚状态。
 μ PD78F0714 中，1 位操作指令按照以下顺序执行。

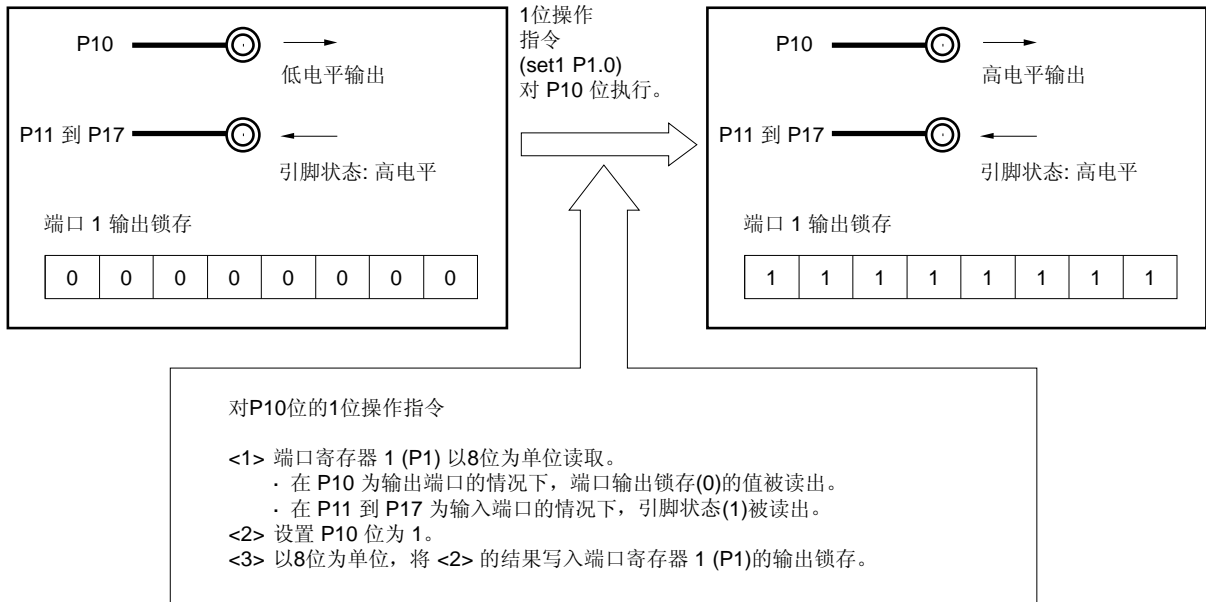
- <1> 以 8 位为单位读取 Pn 寄存器。
- <2> 操作目标位。
- <3> 以 8 位为单位写入 Pn 寄存器。

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

通过步骤 <2>中的操作，此值变为 FFH。

通过步骤 <3>中的操作，FFH 被写到输出锁存中。

图 4-18. 1 位操作指令 (P10)



第五章 时钟发生器

5.1 时钟发生器的功能

时钟发生器用于生成提供给 CPU 以及周边硬件的时钟。

以下两个系统时钟振荡器可用。

- X1 振荡器
X1 振荡器提供 $f_{XP} = 5.0$ 到 20.0 MHz 的时钟。通过执行 STOP 指令或设置主 OSC 控制寄存器 (MOC) 和处理器时钟控制寄存器 (PCC) 可以停止振荡。
- 内部振荡器
内部振荡器用于提供一个 $f_R = 240$ kHz (典型值) 的时钟。当“可以由软件停止”由选项字节设置并且 X1 输入时钟用作 CPU 时钟时, 通过设置内部振荡模式寄存器 (RCM) 可以停止振荡。

- 备注**
1. f_{XP} : X1 输入时钟振荡频率
 2. f_R : 内部振荡时钟频率

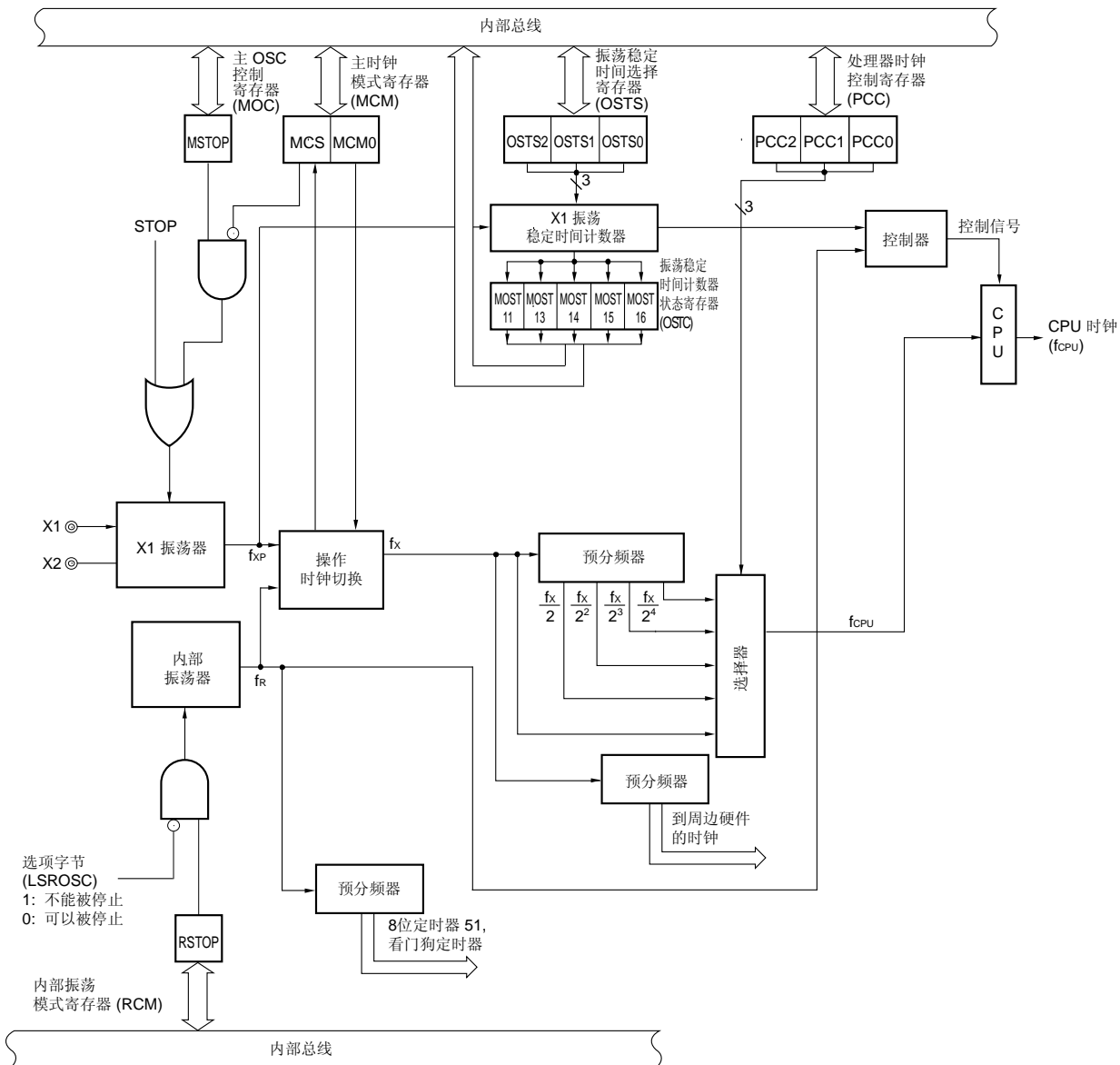
5.2 时钟发生器的配置

时钟发生器由以下硬件组成。

表 5-1. 时钟发生器的配置

项目	配置
控制寄存器	处理器时钟控制寄存器 (PCC) 内部振荡模式寄存器 (RCM) 主时钟模式寄存器 (MCM) 主 OSC 控制寄存器 (MOC) 振荡稳定时间计数器状态寄存器 (OSTC) 振荡稳定时间选择寄存器 (OSTS) 系统等待控制寄存器 (VSWC)
振荡器	X1 振荡器 内部振荡器

图 5-1. 时钟发生器的框图



5.3 控制时钟发生器的寄存器

以下七个寄存器是用于控制时钟发生器的。

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

(1) 处理器时钟控制寄存器（PCC）

PCC 寄存器用来设置 CPU 时钟分频率。

PCC 可以通过 1 位或 8 位存储器操作指令来进行设置。

RESET 输入清除 PCC 为 00H。

图 5-2. 处理器时钟控制寄存器（PCC）的格式

地址：FFF_{BH} 复位后：00H 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		
			MCM0 = 0		MCM0 = 1
0	0	0	f _x	f _R	f _{xP}
0	0	1	f _x /2	f _R /2	f _{xP} /2
0	1	0	f _x /2 ²	— ^注	f _{xP} /2 ²
0	1	1	f _x /2 ³	— ^注	f _{xP} /2 ³
1	0	0	f _x /2 ⁴	— ^注	f _{xP} /2 ⁴
除上面以外			禁止设置		

注 禁止设置。

注意事项 确保将第 3 位到第 7 位清除为 0。

- 备注
1. MCM0: 主时钟模式寄存器（MCM）的第 0 位
 2. f_x: 主系统时钟振荡频率（X1 输入时钟振荡频率或内部振荡时钟频率）
 3. f_R: 内部振荡时钟频率
 4. f_{xP}: X1 输入时钟振荡频率

在 μ PD78F0714 中，最快的指令可以在 2 个 CPU 时钟周期内执行。因此，CPU 时钟（f_{CPU}）和最小指令执行时间之间的关系如表 5-2 所示。

表 5-2. CPU 时钟与最短指令执行时间之间的关系

CPU 时钟 (f _{CPU})	最短指令执行时间: 2/f _{CPU}		
	X1 输入时钟 ^{#1}		内部振荡时钟 ^{#1}
	20 MHz 操作时	16 MHz 操作时	240 kHz (典型值) 操作时
f _x	0.1 μ s	0.125 μ s	8.3 μ (典型值)
f _x /2	0.2 μ s	0.25 μ s	16.6 μ (典型值)
f _x /2 ²	0.4 μ s	0.5 μ s	— ^{#2}
f _x /2 ³	0.8 μ s	1.0 μ s	— ^{#2}
f _x /2 ⁴	1.6 μ s	2.0 μ s	— ^{#2}

- 注
1. 主时钟模式寄存器 (MCM) 用来设置 CPU 时钟 (X1 输入时钟 / 内部振荡时钟) (参见图 5-4)。
 2. 禁止设置。

(2) 内部振荡模式寄存器 (RCM)

该寄存器用于设置内部振荡器的操作模式。

当通过选项字节对内部振荡器设置“可以由软件停止”并且 X1 输入时钟被选作 CPU 时钟时，该寄存器有效。如果通过选项字节对内部振荡器选择“不能被停止”，对该寄存器的设置无效。

RCM 可以通过 1 位或 8 位存储器操作指令来进行设置。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 5-3. 内部振荡模式寄存器 (RCM) 的格式

地址: FFA0H 复位后: 00H R/W

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

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

注意事项 在设置 RSTOP 之前，确保主时钟模式寄存器 (MCM) 第 1 位 (MCS) 为 1。

(3) 主时钟模式寄存器 (MCM)

该寄存器设置 CPU 时钟 (X1 输入时钟 / 内部振荡时钟)。

MCM 可以通过 1 位或 8 位存储器操作指令来进行设置。

RESET 输入清除该寄存器为 00H。

图 5-4. 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	0	MCS	MCM0

MCS	CPU 时钟状态
0	操作于内部振荡时钟
1	操作于 X1 输入时钟

MCM0	CPU 源时钟的选择
0	内部振荡时钟
1	X1 输入时钟

注 第 1 位是只读的。

注意事项 当内部振荡时钟被选作 CPU 的源时钟时, 内部振荡器输出 (fx) 的分频时钟被提供给周边硬件 (fx = 240 kHz (典型值))。

操作于内部振荡时钟的周边硬件的操作并不保证。因此, 当内部振荡时钟被选作 CPU 的源时钟时, 不要使用周边硬件。此外, 在将 CPU 的源时钟由 X1 输入时钟切换为内部振荡时钟之前, 停止周边硬件。然而, 注意当 CPU 操作于内部振荡时钟时, 以下周边硬件可以被使用。

- 看门狗定时器
- 当 $f_R/2^7$ 被选作计数时钟时的 8 位定时器 51
- 将外部时钟选作时钟源的周边硬件
(当 16 位向上 / 向下计数器 ITENC20 或 16 位定时器 / 事件计数器 00 的外部计数时钟被选择时除外)

(4) 主 OSC 控制寄存器 (MOC)

该寄存器用于选择 X1 输入时钟的操作模式。

当 CPU 操作于内部振荡时钟时，该寄存器用来停止 X1 振荡器的操作。因此，该寄存器只有在 CPU 操作于内部振荡时钟时才有效。

MOC 可以通过 1 位或 8 位存储器操作指令来进行设置。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 5-5. 主 OSC 控制寄存器 (MOC) 的格式

地址: FFA2H 复位后: 00H R/W

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

MSTOP	X1 振荡器操作的控制
0	X1 振荡器工作
1	X1 振荡器停止

注意事项 在设置 MSTOP 之前，确保主时钟模式寄存器 (MCM) 第 1 位 (MCS) 为 0。

(5) 振荡稳定时间计数状态寄存器 (OSTC)

这是 X1 输入时钟振荡稳定时间计数器的状态寄存器。如果内部振荡时钟被用作 CPU 时钟，X1 输入时钟振荡稳定时间可以被检查。

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，MSTOP = 1 可以将 OSTC 清除为 00H。

图 5-6. 振荡稳定时间计数状态寄存器 (OSTC) 的格式

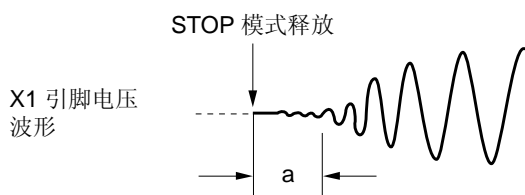
地址: FFA3H 复位后: 00H R

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 _{XP} = 20 MHz	
1	0	0	0	0	2 ¹¹ /f _{XP} 最小值	102.4 μs 最小值
1	1	0	0	0	2 ¹³ /f _{XP} 最小值	409.6 μs 最小值
1	1	1	0	0	2 ¹⁴ /f _{XP} 最小值	819.2 μs 最小值
1	1	1	1	0	2 ¹⁵ /f _{XP} 最小值	1.64 ms 最小值
1	1	1	1	1	2 ¹⁶ /f _{XP} 最小值	3.27 ms 最小值

- 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。
 - 如果在内部振荡时钟被用作 CPU 时钟时先进入 STOP 模式，然后释放，那么将按如下对振荡稳定时间进行设置。
 - OSTC 振荡稳定期望时间 ≤ 由 OSTC 设定的振荡稳定时间

X1 振荡稳定时间计数器用于累计通过 OSTC 所设置的振荡稳定时间。因此，需要注意在 STOP 模式被释放后，只有达到 OSTC 设定的振荡稳定时间时的状态会被赋予 OSTC。
 - 无论 STOP 模式是由 $\overline{\text{RESET}}$ 输入释放还是由中断产生释放，STOP 模式释放时的等待时间都不包含 STOP 模式释放到时钟振荡开始（下面的“a”）的时间。



备注 f_{XP}: X1 输入时钟振荡频率

(6) 振荡稳定时间选择寄存器 (OSTS)

该寄存器用于在 STOP 模式释放时选择 X1 振荡稳定等待时间。

由 OSTS 设置的等待时间只有在 STOP 模式被释放并且 X1 输入时钟被选作 CPU 时钟时才有效。内部振荡时钟被选作 CPU 时钟并且 STOP 模式被释放后，振荡稳定时间必须通过 OSTC 来确认。

OSTS 可以通过 8 位存储器操作指令来进行设置。

$\overline{\text{RESET}}$ 输入设置 OSTS 为 05H。

图 5-7. 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

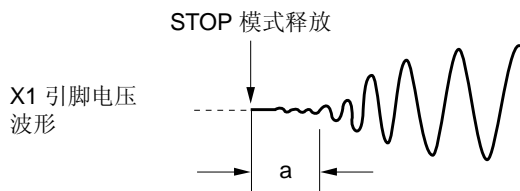
OSTS2	OSTS1	OSTS0	振荡稳定时间选择	
				$f_{XP} = 20 \text{ MHz}$
0	0	1	$2^{11}/f_{XP}$	102.4 μs
0	1	0	$2^{13}/f_{XP}$	409.6 μs
0	1	1	$2^{14}/f_{XP}$	819.2 μs
1	0	0	$2^{15}/f_{XP}$	1.64 ms
1	0	1	$2^{16}/f_{XP}$	3.27 ms
除上面以外			禁止设置	

注意事项 1. 如果在内部振荡时钟被用作 CPU 时钟时先进入 STOP 模式，然后释放，那么将按如下对振荡稳定时间进行设置。

- OSTC 振荡稳定期望时间 \leq 由 OSTS 设定的振荡稳定时间

X1 振荡稳定时间计数器用于累计通过 OSTS 所设置的振荡稳定时间。因此，需要注意在 STOP 模式被释放后，只有达到 OSTS 设定的振荡稳定时间时的状态会被赋予 OSTC。

2. 无论 STOP 模式是由 $\overline{\text{RESET}}$ 输入释放还是由中断产生释放，STOP 模式释放时的等待时间都不包含 STOP 模式释放到时钟振荡开始（下面的“a”）的时间。”



备注 f_{XP} : X1 输入时钟振荡频率

(7) 系统等待控制寄存器 (VSWC)

该寄存器用来控制高速 CPU 和低速周边 I/O 连接时的等待状态。

VSWC 可以通过 1 位或 8 位存储器操作指令来进行设置。

RESET 输入清除该寄存器为 00H。

图 5-8. 系统等待控制寄存器 (VSWC) 的格式

地址: FFFDH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
VSWC	0	0	0	0	0	0	PDW1	0

PDW1	系统时钟数据等待的控制
0	无等待
1	两个等待状态被插入

- 注意事项**
1. 如果最小指令执行时间为 $0.125 \mu\text{s}$ 或更小 ($f_{XP} = 16 \text{ MHz}$ 或更大), 确保插入两个等待状态。
 2. 确保将第 0 位和第 2 到 7 位清除为 0。

5.4 系统时钟振荡器

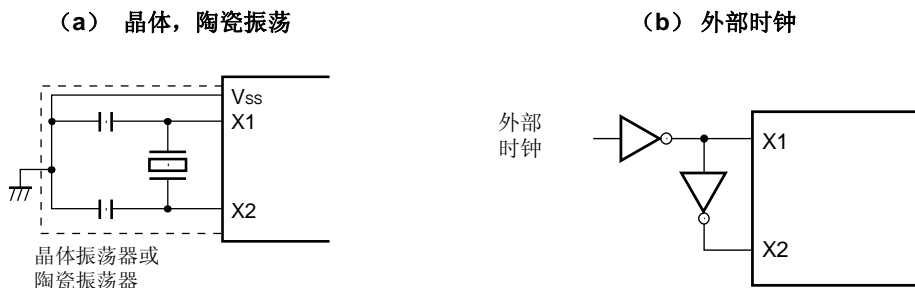
5.4.1 X1 振荡器

X1 振荡器使用连接到 X1 和 X2 引脚的晶体振荡器或陶瓷振荡器（标准：20 MHz）来振荡。

一个外部时钟可以被输入到 X1 振荡器。在这种情况下，输入时钟信号到 X1 引脚并且输入反相信号到 X2 引脚。

图 5-9 显示了 X1 振荡器的外部电路的示例。

图 5-9. X1 振荡器外部电路示例



注意事项 当使用 X1 振荡器时，图 5-9 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其周围布线。
- 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。

5.4.2 错误的振荡器连接示例

图 5-10 显示了错误的谐振器连接方式。

图 5-10. 错误的振荡器连接示例 (1 / 2)

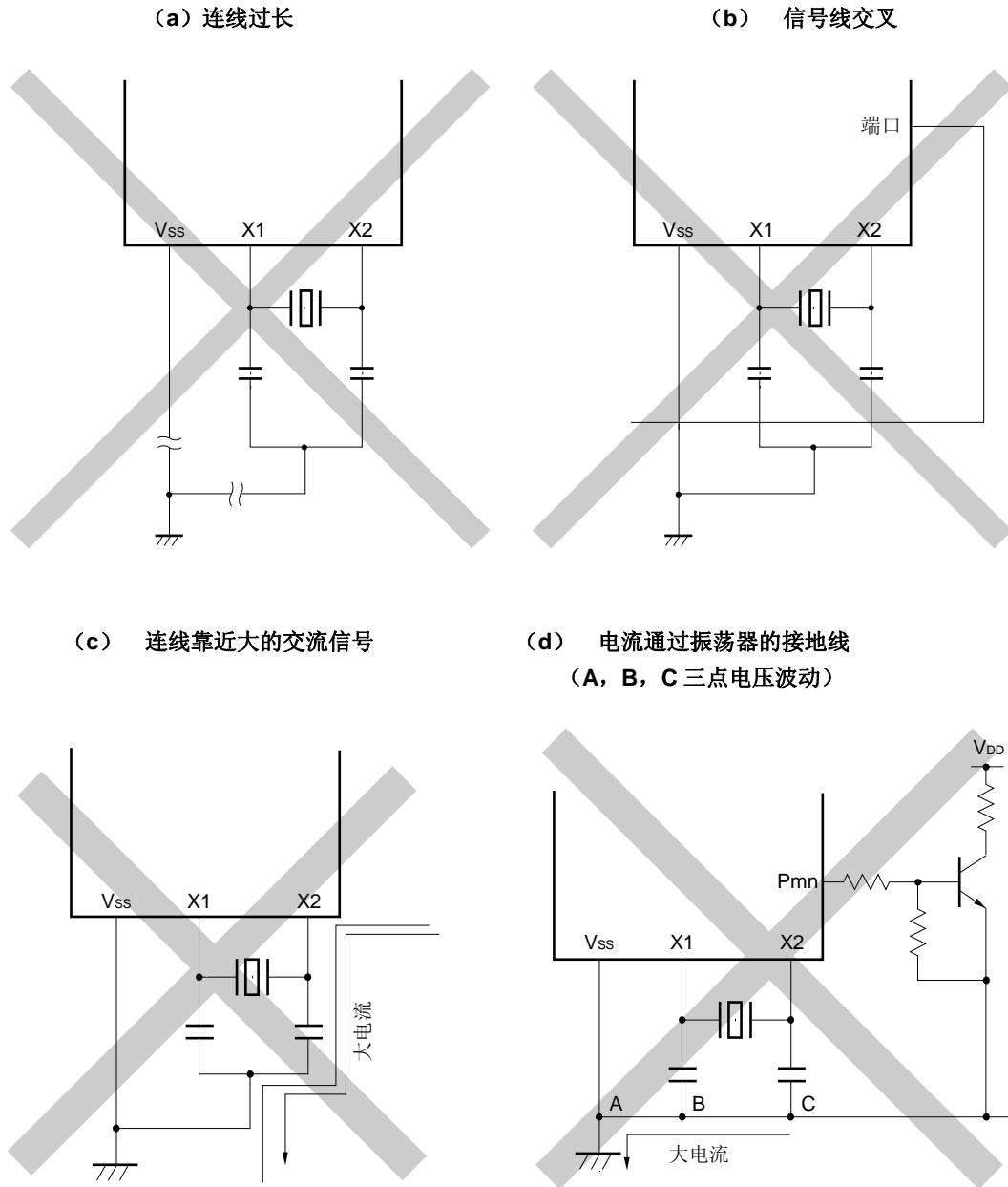
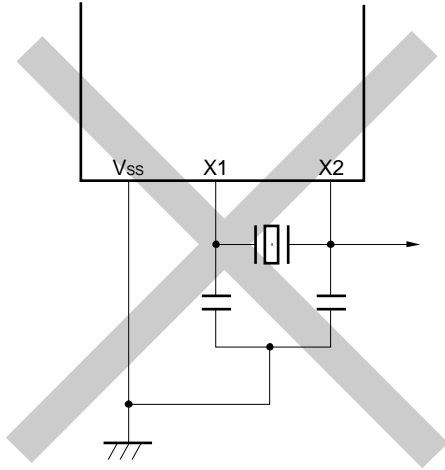


图 5-10. 错误的振荡器连接示例 (2 / 2)

(e) 从振荡器中获取信号



5.4.3 内部振荡器

内部振荡集成于 μ PD78F0714中。

可以通过选项字节来选择“可以由软件停止”或“不能被停止”。内部振荡时钟在 $\overline{\text{RESET}}$ 释放后总是振荡（240 kHz（典型值））。

5.4.4 预分频器

当X1输入时钟被选作CPU的源时钟时，预分频器通过分频X1振荡器的输出来产生各种时钟。

注意事项 当内部振荡时钟被选作CPU的源时钟时，预分频器通过分频内部振荡器的输出（ $f_x = 240 \text{ kHz}$ （典型值））来产生各种时钟。

5.5 时钟发生器的操作

时钟发生器用于生成以下时钟并控制 CPU 的操作模式，例如待机模式。

- X1 输入时钟 f_{XP}
- 内部振荡时钟 f_R
- CPU 时钟 f_{CPU}
- 到周边硬件的时钟

在 μ PD78F0714 产品中，片上内部振荡器在复位释放后开始输出时，CPU 开始工作，因此以下操作被允许。

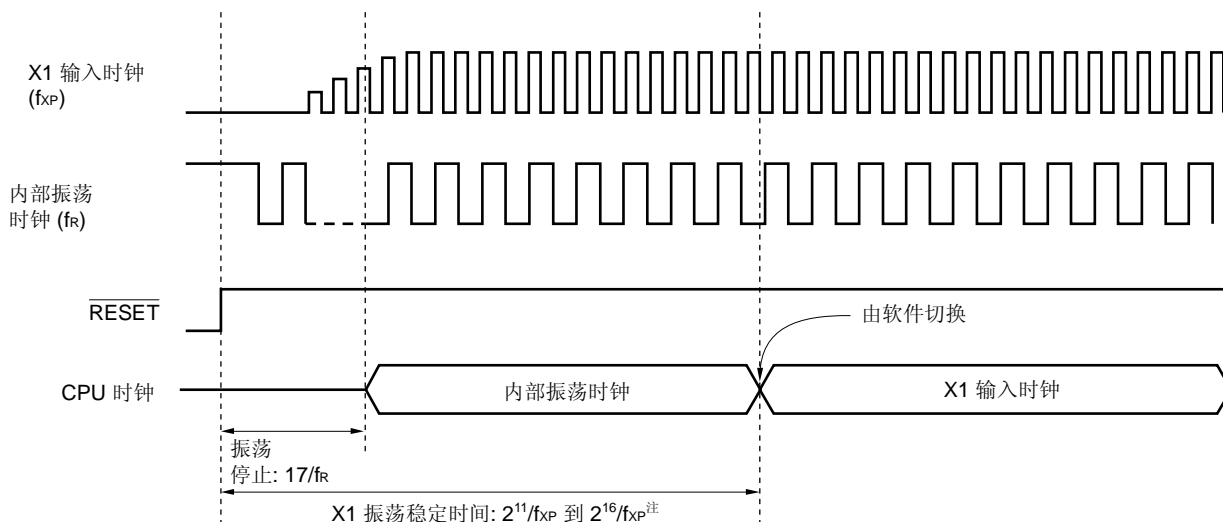
(1) 安全功能的增强

当 X1 输入时钟按默认设置被设置为 CPU 时钟时，如果 X1 输入时钟损坏或错误连接，那么设备将不能运行，因此复位释放后不应进行操作。然而，CPU 的开始时钟是片上内部振荡时钟，因此设备可以在复位释放后通过内部振荡时钟开始操作。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) 性能的提高

由于 CPU 无需等待 X1 输入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。使用内部振荡器的 CPU 默认启动时序图如图 5-11 所示。

图 5-11. 使用内部振荡器的 CPU 默认启动时序图



注 使用振荡稳定时间计数器状态寄存器（OSTC）来检查。

- 当 $\overline{\text{RESET}}$ 信号产生时，主时钟模式寄存器（MCM）的第 0 位被清除为 0 并且内部振荡时钟被设置为 CPU 时钟。然而，在 $\overline{\text{RESET}}$ 释放后内部振荡时钟的 17 个周期过去后，时钟提供给 CPU（或者提供给 CPU 的时钟停止 17 个周期）。在 $\overline{\text{RESET}}$ 期间，X1 输入时钟和内部振荡时钟的振荡被停止。
- $\overline{\text{RESET}}$ 释放后，在 X1 输入时钟振荡稳定时间过去后，使用主时钟模式寄存器（MCM）的第 0 位（MCM0），CPU 时钟可以从内部振荡时钟切换为 X1 输入时钟。此时，在切换 CPU 时钟前使用振荡稳定时间计数器状态寄存器来检查振荡稳定时间。CPU 时钟状态可以使用 MCM 的第 1 位（MCS）来检查。
- 如果 X1 输入时钟被选作 CPU 时钟，当通过选项字节对内部振荡时钟选择“可以由软件停止”时，内部振荡器可以使用内部振荡模式寄存器（RCM）来设置为停止 / 振荡。确保此时的 MCS 为 1。
- 当内部振荡时钟被用作 CPU 时钟时，X1 输入时钟可以使用主 OSC 控制寄存器（MOC）来设置为停止 / 振荡。确保此时的 MCS 为 0。
- X1 输入时钟被用作 CPU 时钟时，当释放 STOP 模式时，由振荡稳定时间寄存器（OSTS）选择的振荡稳定时间（ $2^{11}/f_{XP}$, $2^{13}/f_{XP}$, $2^{14}/f_{XP}$, $2^{15}/f_{XP}$, $2^{16}/f_{XP}$ ）必须要保证。

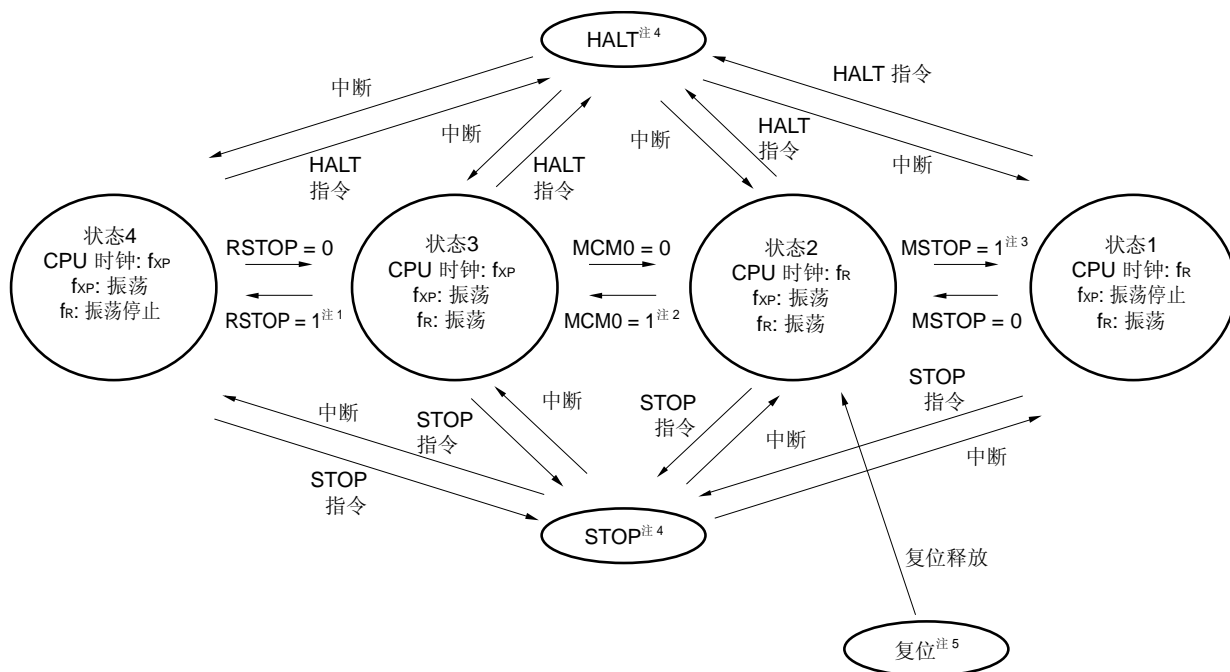
此外，当 $\overline{\text{RESET}}$ 释放并且在内部振荡时钟用作 CPU 时钟期间 STOP 模式被释放，不存在振荡稳定时间等待。

当切换 X1 输入时钟作为 CPU 时钟时，通过使用振荡稳定时间计数器状态寄存器（OSTC）来检查振荡稳定时间。

该产品的状态转换图如图 5-12 所示，每种操作状态中的操作时钟之间的关系以及振荡控制标志和每个时钟的振荡状态之间的关系分别如表 5-3 和 5-4 所示。

图 5-12. 状态转换图 (1 / 2)

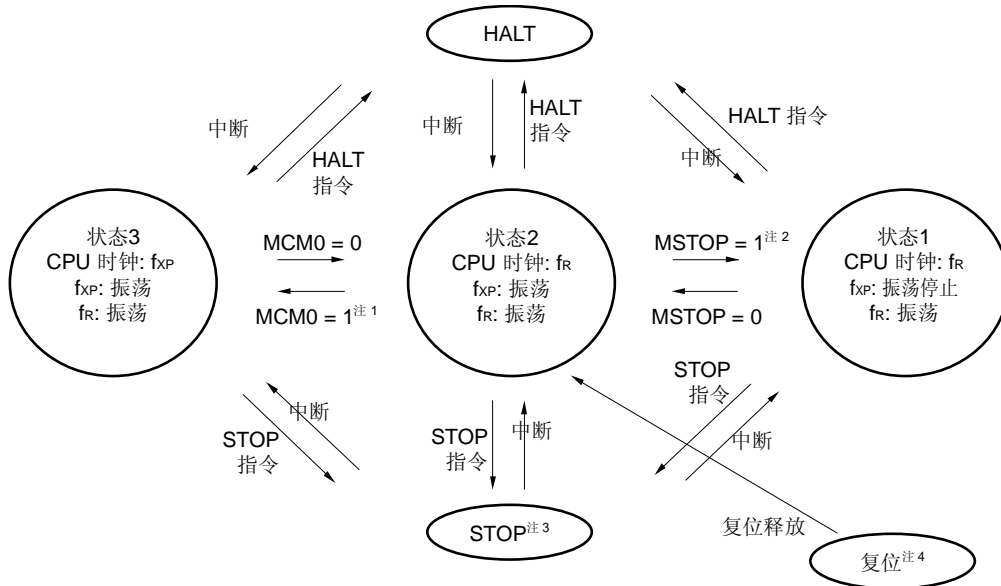
(1) 当“内部振荡器可以由软件停止”被选项字节选择时



- 注
1. 当从状态 3 转移到状态 4 时，确保主时钟模式寄存器 (MCM) 的第 1 位 (MCS) 为 1。
 2. 在复位和 STOP 被释放后，在从状态 2 转移到状态 3 之前，使用振荡稳定时间计数器状态寄存器 (OSTC) 来检查 X1 输入时钟振荡稳定时间状态。
 3. 当从状态 2 转移到状态 1 时，确保 MCS 为 0。
 4. 当“内部振荡器可以由软件停止”由选项字节选定时，无论看门狗定时器的源时钟是什么，在 HALT 和 STOP 模式下，看门狗定时器都将停止操作。然而，如果 RSTOP = 0，即使在 HALT 和 STOP 模式下，内部振荡器的振荡也不会停止。
 5. 所有复位源 ($\overline{\text{RESET}}$ 输入, POC, LVI 和 WDT)

图 5-12. 状态转换图 (2 / 2)

(2) 当“内部振荡器不能被停止”被选项字节选择时



- 注
1. 在复位和 STOP 被释放后，在从状态 2 转移到状态 3 之前，使用振荡稳定时间计数器状态寄存器（OSTC）来检查 X1 输入时钟振荡稳定时间状态。
 2. 当从状态 2 转移到状态 1 时，确保 MCS 为 0。
 3. 如果“内部振荡器不能被停止”由选项字节选择，即使在 STOP 模式下，看门狗定时器也会使用内部振荡时钟操作。内部振荡时钟的分频可以被选作 8 位定时器 51（TM51）的计数源，所以在看门狗定时器溢出前使用 TM51 的中断请求来清除看门狗定时器。如果该处理未被执行，在 STOP 指令执行后，看门狗定时器溢出时将产生一个内部复位信号。
 4. 所有复位源（ $\overline{\text{RESET}}$ 输入，POC，LVI 和 WDT）

表 5-3. 每种操作状态下操作时钟之间的关系

状态 操作模式	X1 振荡器		内部振荡器			释放后的 CPU 时钟	提供给周边的预分频器 时钟	
	MSTOP = 0	MSTOP = 1	注 1	注 2			MCM0 = 0	MCM0 = 1
				RSTOP = 0	RSTOP = 1			
复位	停止		停止			内部振荡时钟	停止	
STOP			振荡	振荡	停止 ^{注4}	注 5	停止	
HALT	振荡	停止 ^{注3}				注 6	内部振荡 时钟	X1

- 注
1. 当通过选项字节对内部振荡器选择“不能被停止”时
 2. 当通过选项字节对内部振荡器选择“可以由软件停止”时
 3. 只有当内部振荡器振荡时。
 4. 只有当 X1 振荡器振荡时。
 5. 在 STOP 指令执行时使用 CPU 时钟操作。
 6. 在 HALT 指令执行时使用 CPU 时钟操作。

注意事项 RSTOP 设置只有当通过选项字节对内部振荡器设置“可以由软件停止”时才有效。

备注 MSTOP: 主 OSC 控制寄存器 (MOC) 的第 7 位
RSTOP: 内部振荡模式寄存器 (RCM) 的第 0 位
MCM0: 主时钟模式寄存器 (MCM) 的第 0 位

表 5-4. 振荡控制标志和时钟振荡状态

		X1 振荡器	内部振荡器
MSTOP = 1	RSTOP = 0	停止	振荡
	RSTOP = 1	禁止设置	
MSTOP = 0	RSTOP = 0	振荡	振荡
	RSTOP = 1		停止

注意事项 RSTOP 设置只有当通过选项字节对内部振荡器设置“可以由软件停止”时才有效。

备注 MSTOP: 主 OSC 控制寄存器 (MOC) 的第 7 位
RSTOP: 内部振荡模式寄存器 (RCM) 的第 0 位

5.6 内部振荡时钟和X1 输入时钟之间切换需要的时间

主时钟模式寄存器（MCM）的第 0 位（MCM0）用来切换内部振荡时钟和 X1 输入时钟。

在实际切换操作中，切换不会在 MCM0 重写后立即发生；在切换 MCM0 后，有几个指令将使用切换前的时钟执行（参见表 5-5）。

MCM 的第 1 位（MCS）用来判断操作是使用内部振荡时钟还是 X1 输入时钟来执行。

要在切换时钟后停止原始时钟，等待表 5-5 中显示的时钟个数。

表 5-5. 内部振荡时钟和 X1 输入时钟之间切换需要的最长时间

PCC			切换需要的时间	
PCC2	PCC1	PCC0	X1→内部振荡时钟	内部振荡时钟 → X1
0	0	0	$f_{XP}/f_R + 1 \text{ clock}$	2 个时钟
0	0	1	$f_{XP}/2f_R + 1 \text{ clock}$	

注意事项 要计算最长时间，设置 $f_R = 120 \text{ kHz}$ 。

- 备注**
1. PCC：处理器时钟控制寄存器
 2. f_{XP} ：X1 输入时钟振荡频率
 3. f_R ：内部振荡时钟频率
 4. 最长时间是切换前的 CPU 时钟的时钟个数。

5.7 CPU时钟转换需要的时间

CPU 时钟可以使用处理器时钟控制寄存器（PCC）的第 0 到 2 位（PCC0 到 PCC2）来转换。

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指令（参见表 5-6）。

表 5-6. CPU 时钟转换需要的最长时间

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

注意事项 当 CPU 操作于内部振荡时钟时，禁止设置以下值。

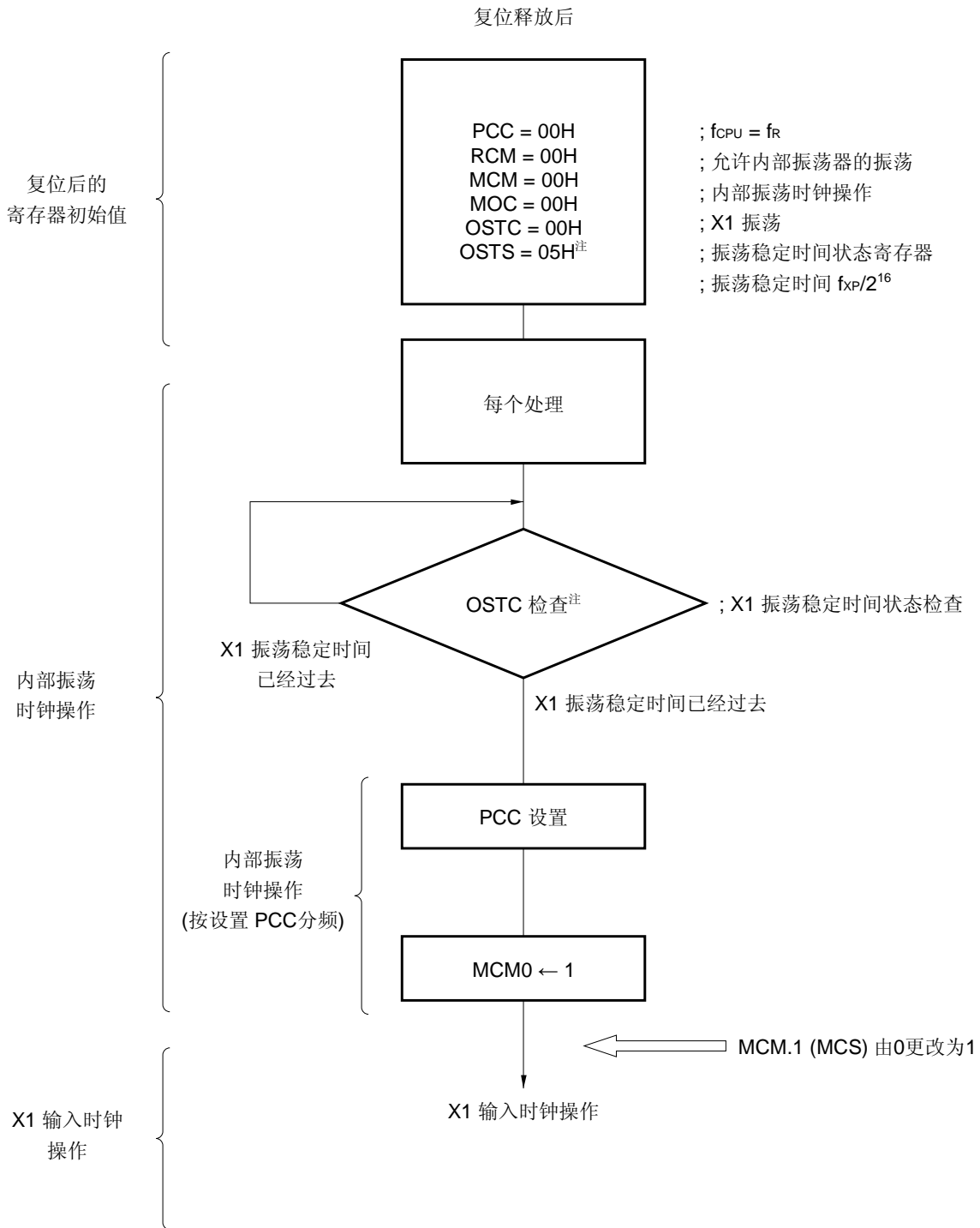
- PCC2, PCC1, PCC0 = 0, 1, 0
- PCC2, PCC1, PCC0 = 0, 1, 1
- PCC2, PCC1, PCC0 = 1, 0, 0

备注 最长时间是转换前的 CPU 时钟的时钟个数。

5.8 时钟切换流程图和寄存器设置

5.8.1 从内部振荡时钟切换为X1 输入时钟

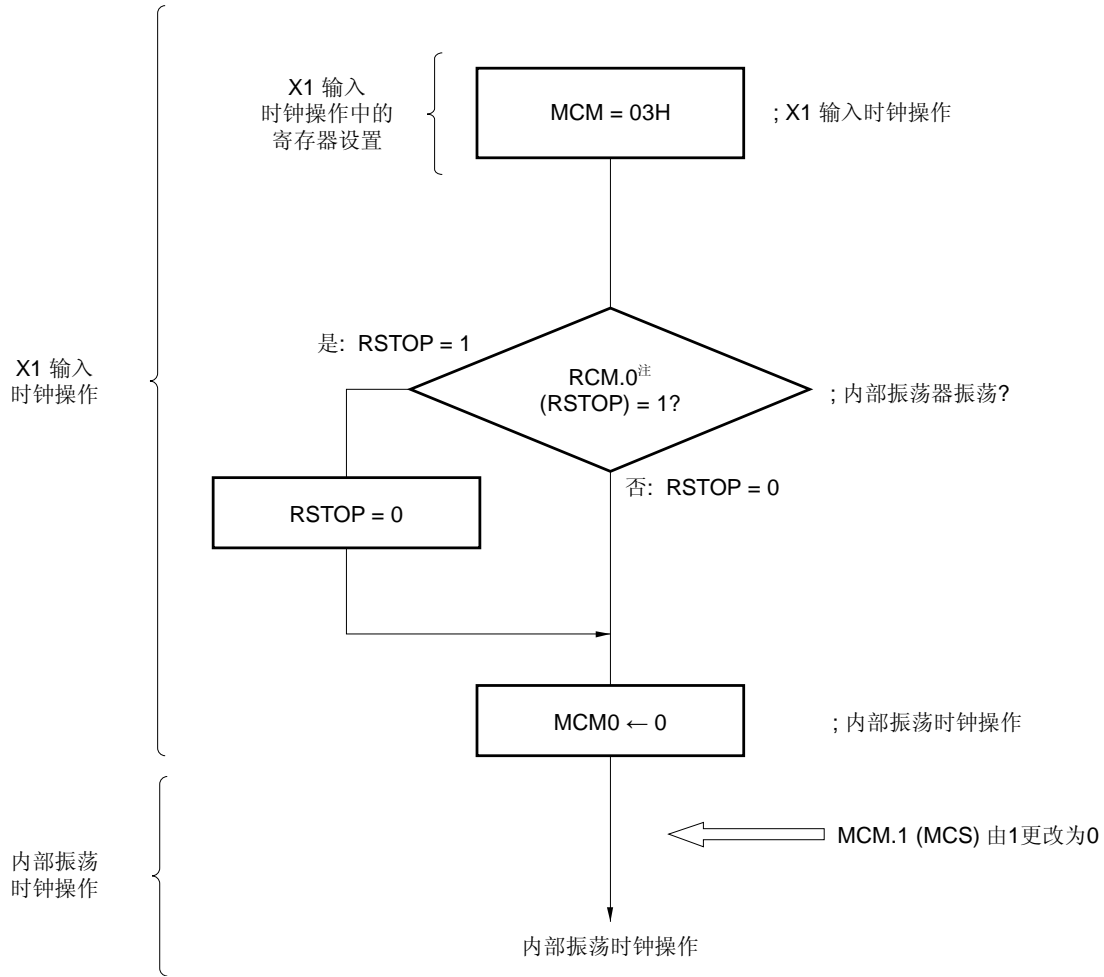
图 5-13 从内部振荡时钟切换为 X1 输入时钟（流程图）



注 在复位释放后，使用 OSTC 寄存器检查 X1 振荡器的振荡稳定等待时间，在振荡稳定时间过去后切换为 X1 输入时钟操作。OSTS 寄存器的设置只有在 X1 输入时钟操作期间 STOP 模式由中断释放后才有效。

5.8.2 从X1输入时钟切换为内部振荡时钟

图 5-14 从 X1 输入时钟切换为内部振荡时钟（流程图）



注 只有当通过选项字节对内部振荡器选择“可以由软件停止”时才需要。

5.8.3 寄存器设置

下表标识每种模式被设置时设置标志和状态标志的状态。

表 5-7. 时钟和寄存器设置

fCPU	模式	设置标志			状态标志
		MCM 寄存器	MOC 寄存器	RCM 寄存器	MCM 寄存器
		MCM0	MSTOP	RSTOP ^{注1}	MCS
X1 输入时钟 ^{注2}	内部振荡器振荡	1	0	0	1
	内部振荡器停止	1	0	1	1
内部振荡时钟	X1 振荡	0	0	0	0
	X1 停止	0	1	0	0

- 注
1. 只有当通过选项字节对内部振荡器选择“可以由软件停止”时才有效。
 2. 在 X1 输入时钟操作期间，不要设置 MSTOP = 1（即使设置 MSTOP = 1，X1 振荡也不会停止）。

第六章 10 位逆变器控制定时器

6.1 10 位逆变器控制定时器概述

10 位逆变器控制定时器使逆变器控制变为可能。它由一个 8 位死区时间产生定时器组成，并且允许非重叠有效电平输出。

6.2 10 位逆变器控制定时器的功能

10 位逆变器控制定时器实现逆变器控制。它集成一个 8 位定时器用于死区时间产生并且可以输出不重叠有效电平的波形。总共六个正相位和负相位被输出。此外，还提供有效电平更改功能和通过外部输入（TW0TOFFP）关闭输出的功能。

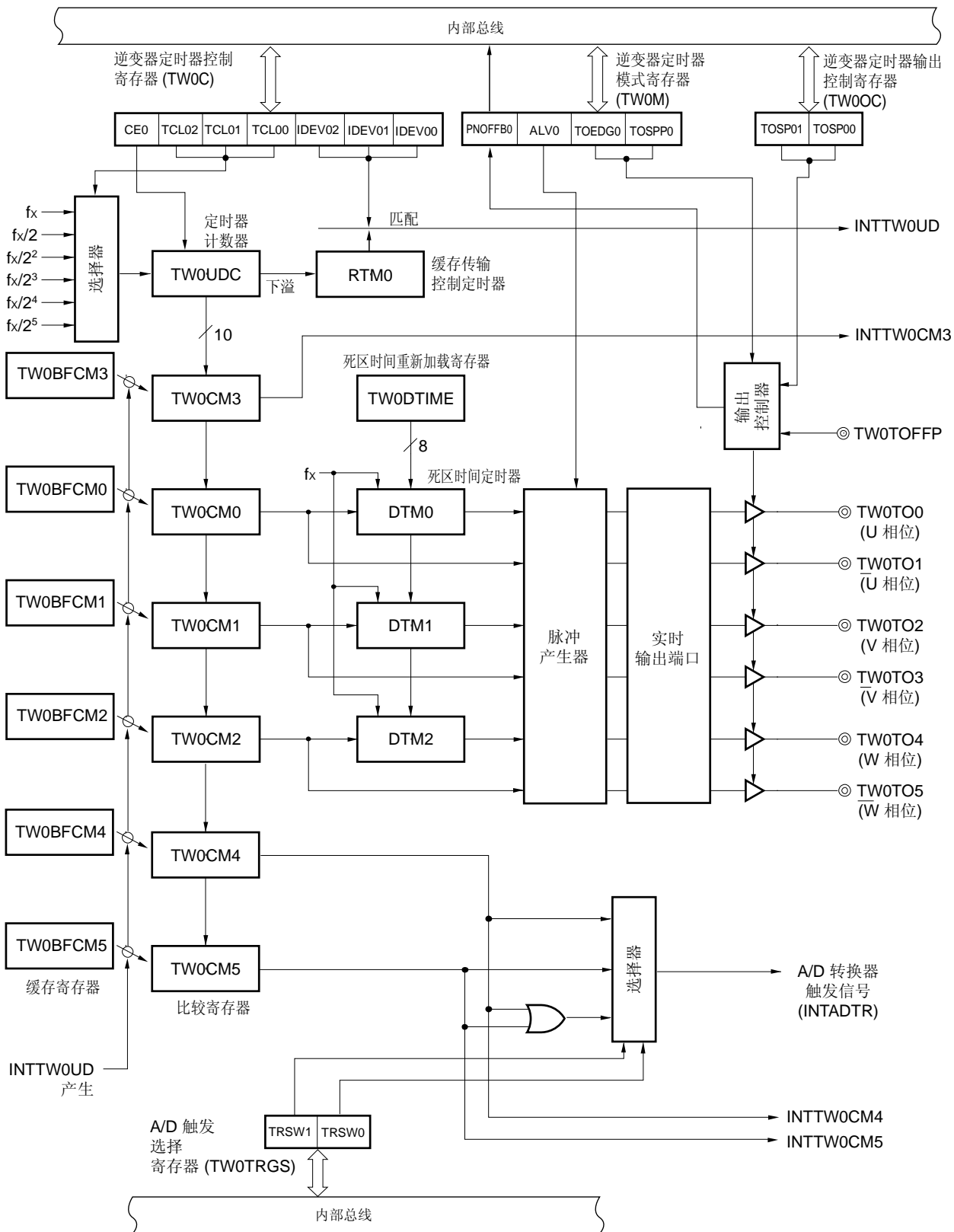
6.3 10 位逆变器控制定时器的配置

10 位逆变器控制定时器的包含以下硬件。

表 6-1. 10 位逆变器控制定时器的配置

项目	功能
定时器计数器	10 位向上 / 向下计数器 × 1 (TW0UDC) 死区时间定时器 × 3 (DTM0, DTM1, DTM2) 缓存传输控制定时器 × 1 (RTM0)
寄存器	10 位比较寄存器 × 6 (TW0CM0 到 TW0CM5) 10 位缓存寄存器 × 6 (TW0BFCM0 到 TW0BFCM5) 死区时间重新加载寄存器 × 1 (TW0DTRG)
定时器输出	6 (TW0TO0, TW0TO1, TW0TO2, TW0TO3, TW0TO4, TW0TO5)
控制寄存器	逆变器定时器控制寄存器 (TW0C) 逆变器定时器模式寄存器 (TW0M) A/D 触发选择寄存器 (TW0TRGS) 逆变器定时器输出控制寄存器 (TW0OC)

图 6-1. 10 位逆变器控制定时器的框图



(1) 10 位向上 / 向下计数器 (TW0UDC)

TW0UDC 是与计数时钟上升沿同步对计数脉冲计数的 10 位向上 / 向下计数器。当定时器启动时，计数脉冲的个数从 0 递增，当预置到比较寄存器 3 (TW0CM3) 中的值与 TW0UDC 计数值匹配时，它切换为向下计数操作。

如果在向下计数期间计数值变为 000H，一个下溢信号产生并且中断请求信号 INTTW0UD 产生。当下溢发生时，它从向下计数操作切换为向上计数操作。INTTW0UD 在每次下溢时正常产生，但是发生的次数要除以逆变器定时器控制寄存器 (TW0C) 的 IDEV00 到 IDEV02 位。

TW0UDC 不能被读取 / 写入。

TW0UDC 的周期由 TM0CM3 控制。

计数时钟可以从六种时钟中选择：fx, fx/2, fx/4, fx/8, fx/16, fx/32。

$\overline{\text{RESET}}$ 输入或清除 TW0C7 的 CE0 位将设置 TW0UDC 为 000H。

(2) 10 位比较寄存器 0 到 2 (TW0CM0 到 TW0CM2)

TW0CM0 到 TW0CM2 是与 TW0UDC 比较的 10 位比较寄存器，如果它们匹配，触发器的内容将会被更改。

TW0CM0 到 TW0CM2 中的每个寄存器与一个缓存寄存器 (TW0BFCM0 到 TW0BFCM2) 一同被提供，所以缓存的内容可以在中断请求信号 INTTW0UD 产生时转移到 TW0CM0 到 TW0CM2。

只有当 TW0UDC 停止时，对 TW0CM0 到 TW0CM2 的写操作才有可能。

要设置输出时序，写入数据到 TW0BFCM0 到 TW0BFCM2。

$\overline{\text{RESET}}$ 输入或清除 TW0C 的 CE0 位将设置这些寄存器为 000H。

(3) 10 位比较寄存器 3 (TW0CM3)

TW0CM3 是控制 TW0UDC 的上限值的 10 位比较寄存器。如果 TW0UDC 的计数值与 TW0CM3 的值或 0 匹配，向上 / 向下计数在下一个时钟时被切换。

TW0CM3 提供一个缓存寄存器 (TW0BFCM3)，在中断请求信号 INTTW0UD 产生，缓存寄存器的内容转移到 TW0CM3。

只有当 TW0UDC 停止时，TW0CM3 才能被写入。

要设置 TW0UDC 的周期，写入数据到 TW0BFCM3。

$\overline{\text{RESET}}$ 输入设置 TW0CM3 为 0FFH。

不要设置 TW0CM3 为 000H。

(4) 10 位比较寄存器 4, 5 (TW0CM4, TW0CM5)

TW0CM4 和 TW0CM5 是与 TW0UDC 比较的 10 位比较寄存器，如果它们匹配，中断请求信号将产生。

TW0CM4 和 TW0CM5 中的一个寄存器由一个缓存寄存器 (TW0BFCM4, TW0BFCM5) 提供，所以缓存的内容可以在中断请求信号 INTTW0UD 产生时转移到 TW0CM4 到 TW0CM5。

只有当 TW0UDC 停止时，对 TW0CM4 和 TW0CM5 的写操作才可能。

要设置输出时序，写入数据到 TW0BFCM4 和 TW0BFCM5。

$\overline{\text{RESET}}$ 输入或清除 TW0C 的 CE0 位将设置这些寄存器为 000H。

(5) 10 位缓存寄存器 0 到 5 (TW0BFCM0 到 TW0BFCM5)

TW0BFCM0 到 TW0BFCM5 是 10 位寄存器。它们在中断请求信号 INTTW0UD 发生时转移数据到对应每个缓存寄存器的比较寄存器 (TW0CM0 到 TW0CM5)。

无论 TW0UDC 计数是停止还是操作, TW0BFCM0 到 TW0BFCM5 都可以被读取 / 写入。

$\overline{\text{RESET}}$ 输入设置 TW0BFCM0 到 TW0BFCM2、TW0BFCM4 和 TW0BFCM5 为 000H, 设置 TW0BFCM3 为 0FFH。

这些寄存器可以以字和字节为单位读取 / 写入。对应少于 8 位的读取 / 写入, TW0BFCM0L 到 TW0BFCM5L 被使用。

(6) 死区时间重新加载寄存器 (TW0DTIME)

TW0DTIME 是设置死区时间的 8 位寄存器, 它对三个死区时间定时器 (DTM0 到 DTM2) 是公用的。

然而, 从 TW0DTIME 向 DTM0、DTM1 和 DTM2 加载数据的时序是独立的。

TW0DTIME 只能在 TW0UDC 计数停止时才能写入。在定时器操作期间, 如果执行重写 TW0DTIME 的指令, 数据不会更改。

$\overline{\text{RESET}}$ 输入设置 TW0DTIME 为 FFH。

如果 TW0DTIME 被设置为 00H, 将会执行死区时间为 $1/f_x$ 的输出。

(7) 死区时间定时器 0 到 2 (DTM0 到 DTM2)

DTM0 到 DTM2 是产生死区时间的 8 向下计数器。

TW0CM0 到 TW0CM2 与 TW0UDC 之间的比较匹配时, 死区时间重新加载寄存器 (TW0DTIME) 的值被重新加载之后, 向下计数才能执行。DTM0 到 DTM2 在从 00H 变为 FFH 时产生下溢信号并停止在 FFH。

计数时钟是 f_x 。

DTM0 到 DTM2 不能读取 / 写入。

$\overline{\text{RESET}}$ 输入或清除 TW0C 的 CE0 位将设置这些寄存器为 FFH。

(8) 缓存传输控制定时器 (RTM0)

RTM0 是 3 位向上计数器。它具有分频中断请求信号 INTTW0UD 的功能。

递增将随着 TW0UDC 下溢信号而执行。当数值与 TW0C 的 IDEV00 到 IDEV02 位中设置的分频数匹配时, INTTW0UD 才会产生。

RTM0 不能被读取 / 写入。

$\overline{\text{RESET}}$ 输入设置 RTM0 为 7H。产生 INTTW0UD 和清除 TW0C 的 CE0 位也将设置 RTM0 为 7H。

6.4 控制 10 位逆变器控制定时器的寄存器

以下四个寄存器控制 10 位逆变器控制定时器。

- 逆变器定时器控制寄存器 (TW0C)
- 逆变器定时器模式寄存器 (TW0M)
- A/D 触发选择寄存器 (TW0TRGS)
- 逆变器定时器输出控制寄存器 (TW0OC)

(1) 逆变器定时器控制寄存器 (TW0C)

TW0C 控制 TW0UDC、死区时间定时器 0 到 2 (DTM0 到 DTM2) 以及缓存传输控制定时器 (RTM0) 的操作, 指定 TW0UDC 的计数时钟并选择比较寄存器传输周期。

TW0C 由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除 TW0C 为 00H。

图 6-2. 逆变器定时器控制寄存器的格式

地址: FF88H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TW0C	CE0	0	TCL02	TCL01	TCL00	IDEV02	IDEV01	IDEV00

CE0	TW0UDC, DTM0 到 DTM2, RTM0 操作控制
0	清除并停止 (TW0TO0 到 TW0TO5 为高阻)
1	计数允许

TCL02	TCL01	TCL00	计数时钟选择	
				fx = 20 MHz 时
0	0	0	fx	20 MHz
0	0	1	fx/2	10 MHz
0	1	0	fx/2 ²	5 MHz
0	1	1	fx/2 ³	2.5 MHz
1	0	0	fx/2 ⁴	1.25 MHz
1	0	1	fx/2 ⁵	625 kHz
除上面以外			禁止设置	

IDEV02	IDEV01	IDEV00	INTTW0UD 产生频率选择
0	0	0	每个 TW0UDC 下溢时发生。
0	0	1	每两个 TW0UDC 下溢时发生。
0	1	0	每三个 TW0UDC 下溢时发生。
0	1	1	每四个 TW0UDC 下溢时发生。
1	0	0	每五个 TW0UDC 下溢时发生。
1	0	1	每六个 TW0UDC 下溢时发生。
1	1	0	每七个 TW0UDC 下溢时发生。
1	1	1	每八个 TW0UDC 下溢时发生。

备注 fx: 系统时钟振荡频率

(2) 逆变器定时器模式寄存器 (TWOM)

TWOM 控制 TW0TO0 到 TW0TO5 输出的操作并指定有效电平，同时设置 TW0TOFFP 的有效沿。

TWOM 由 1 位或 8 位存储器操作指令来设置。

RESET 输入清除 TWOM 为 00H。

图 6-3. 逆变器定时器模式寄存器的格式

地址: FF89H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TWOM	0	0	0	PNOFFB0 ^注	ALV0	TOEDG0	TOSPP0	0

PNOFFB0 ^注	输出到 TW0TO0 到 TW0TO5 的控制状态标志
0	输出禁止状态 (TW0TO0 到 TW0TO5 为高阻)
1	输出允许状态

ALV0	TW0TO0 到 TW0TO5 输出有效电平指定
0	低电平
1	高电平

TOEDG0	TW0TOFFP 有效沿指定
0	下降沿
1	上升沿

TOSPP0	由 TW0TOFFP 的有效沿对 W0TO0 到 TW0TO5 输出停止的控制
0	输出不被停止。
1	输出被停止 (TW0TO0 到 TW0TO5 为高阻)。

注 PNOFFB0 位是只读标志。该位不能由软件置位或复位。

PNOFFB0 位在以下情况下复位。

- 当 TW0UDC 停止时 (CE0 = 0)
- 在 TW0UDC 操作期间 (CE0 = 1)，当输出停止由 TW0TOFFP 和 INTWDT 产生时

注意事项 总是设置 TWOM 的第 0、5 到 7 位为 0。

备注 1. TW0TO0 到 TW0TO5 在以下情况下变为高阻状态。然而，如果设置 CE0 = 1，TW0UDC、DTM0 到 DTM2 和 RTM0 定时器不停止。

- TOSPP0 = 1 期间，有效沿输入到 TW0TOFFP 引脚。

要恢复TW0TO0 到 TW0TO5的输出，执行以下流程。

- <1> 写 0 到 CE0 并停止定时器。
- <2> 在使用到的输出停止功能标志中写入 0。
- <3> 复位寄存器为其默认值。

2. PNOFFB0、ALV0、CE0 和 TW0TO0 到 TW0TO5 关系如下。

PNOFFB0	ALV0	CE0	TW0TO0, TW0TO2, TW0TO4	TW0TO1, TW0TO3, TW0TO5
0	0	0	高阻	高阻
0	1	0	高阻	高阻
0	0/1	1	高阻	高阻
1	0/1	1	PWM 波形输出	PWM 波形输出

(3) A/D 触发选择寄存器 (TW0TRGS)

TW0TRGS 是用来从 INTTW0CM4 和 INTTW0CM5 中选择 A/D 转换器触发信号的寄存器，当比较寄存器 (TW0CM4, TW0CM5) 和定时器计数器 (TW0UDC) 匹配时，INTTW0CM4 和 INTTW0CM5 就会产生。

TW0TRGS 由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入设置 TW0TRGS 为 00H。

图 6-4. A/D 触发选择寄存器的格式

地址: FF8BH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TW0TRGS	0	0	0	0	0	0	TRSW1	TRSW0

TRSW1	TRSW0	A/D 触发的选择
0	0	无输出 (INTADTR 保持低电平)
0	1	INTTW0CM4
1	0	INTTW0CM5
1	1	INTTW0CM4 或 INTTW0CM5

(4) 逆变器定时器输出控制寄存器 (TW0OC)

TW0OC 以相位 (U 相位 / V 相位 / W 相位) 为单位设置定时器输出停止。

TW0OC 由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入设置 TW0OC 为 00H。

图 6-5. 逆变器定时器输出控制寄存器的格式

地址: FF8CH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TW0OC	0	0	0	0	0	0	TOSPW1	TOSPW0

TOSPW1	TOSPW0	PWM 输出的输出控制
0	0	TW0TO0 到 TW0TO5 输出允许
0	1	TW0TO0 和 TW0TO1 输出被禁止 (U 相位关闭)
1	0	TW0TO2 和 TW0TO3 输出被禁止 (V 相位关闭)
1	1	TW0TO4 和 TW0TO5 输出被禁止 (W 相位关闭)

6.5 控制 10 位逆变器控制定时器的寄存器

(1) 设置过程

- (a) 使用逆变器控制寄存器 (TW0C) 的 TCL00 到 TCL02 位设置 TW0UDC 的计数时钟, 使用 IDEV00 到 IDEV02 位设置中断请求信号 INTTW0UD 的发生频率。
- (b) 使用逆变器定时器模式寄存器 (TW0M) 的 ALV0 位设置 TW0TO0 到 TW0TO5 引脚的有效电平。
- (c) 在 10 位比较寄存器 3 (TW0CM3) 中设置第一个 PWM 周期的一半宽度。
- PWM 周期 = TW0CM3 值 $\times 2 \times$ TW0UDC 时钟速率
(TW0UDC 的时钟速率使用 TW0C 来设置)
- (d) 在 10 位缓存寄存器 3 (TW0BFCM3) 中设置第二个 PWM 周期的一半宽度。
- (e) 设置死区时间宽度到死区时间重新加载寄存器 (TW0DTIME)。
- 死区时间宽度 = (TW0DTIME + 1) $\times 1/f_x$
f_x: 内部系统时钟
- (f) 设置在第一个周期中使用的 F/F 置位 / 复位时序到 10 位比较寄存器 0 到 2 (TW0CM0 到 TW0CM2)。
- (g) 设置在第二个周期中使用的 F/F 置位 / 复位时序到 TW0BFCM3。
- (h) 在 TW0C 的 CE0 位置位 (1) 后, TW0UDC、死区时间定时器 0 到 2 (DTM0 到 DTM2) 和缓存传输控制定时器 (RTM0) 会被允许。

注意事项 总是使用位操作指令来设置 CE0 位。

- (i) 设置 TW0UDC 操作过程中用于下一个周期的 F/F 置位 / 复位时序到 TW0BFCM0 到 TW0BFCM5。
- (j) 要停止 TW0UDC 操作, 设置 TW0C 的 CE0 位为 0。

注意事项 在 CE0 位重写时不能同时重写另一位。

(2) 对应于设置值的输出波形宽度

- $PWM_{cycle} = TW0CM3 \times 2 \times T_{TW0}$
- 死区时间宽度 = $T_{DTM} = (TW0DTIME + 1) \times 1/f_x$
- 正相位的有效宽度(TW0TO0, TW0TO2, TW0TO4 引脚)
= $\{(TW0CM3 - TW0CM_{up}) + (TW0CM3 - TW0CM_{down})\} \times T_{TW0} - T_{DTM}$
- 负相位的有效宽度 (TW0TO1, TW0TO3, TW0TO5 引脚)
= $(TW0CM_{down} + TW0CM_{up}) \times T_{TW0} - T_{DTM}$

f_x : 系统时钟振荡频率

T_{TW0} : TW0UDC 计数时钟

$TW0CM_{up}$: TW0UDC 向上计数期间 TW0CM0 到 TW0CM2 的设置值

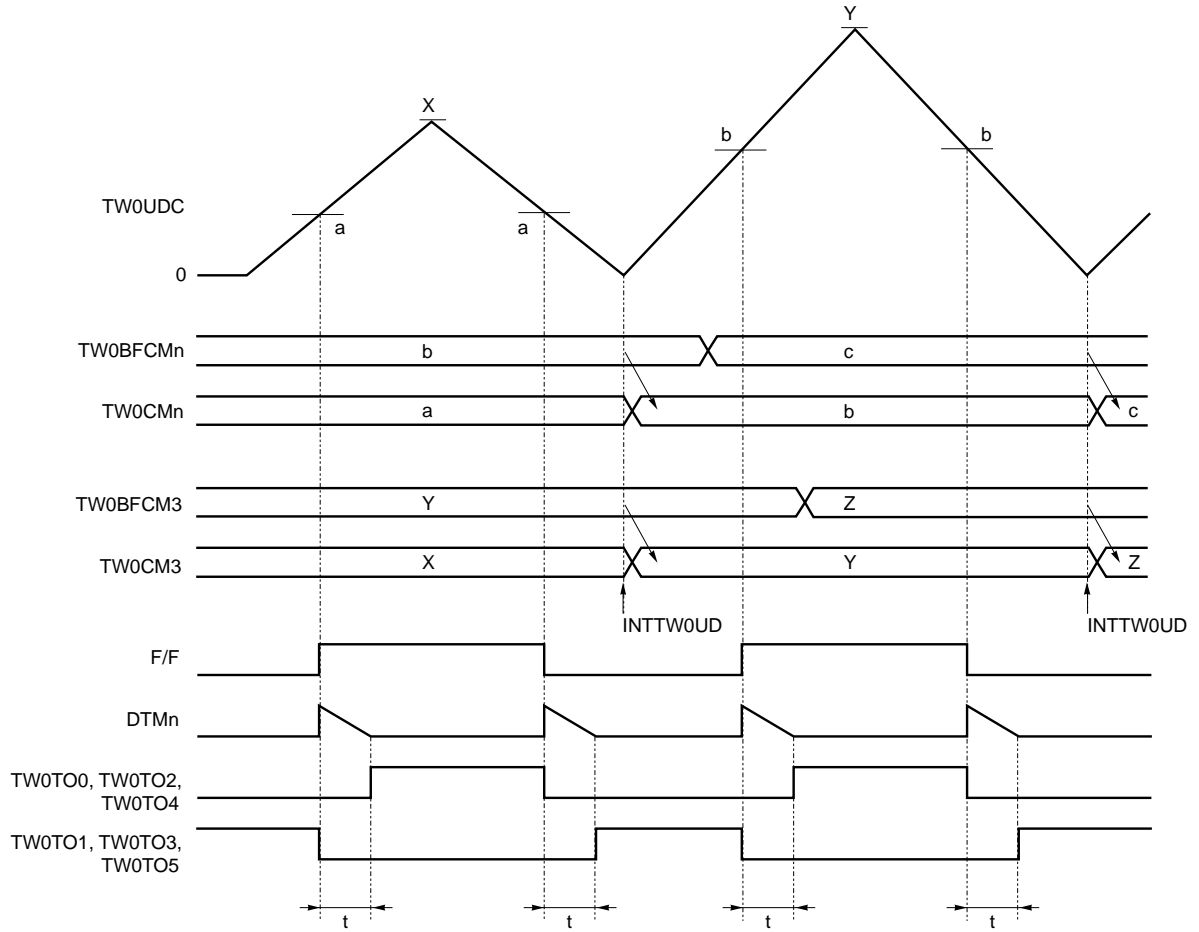
$TW0CM_{down}$: TW0UDC 向下计数期间 TW0CM0 到 TW0CM2 的设置值

注意事项 如果通过以上计算某个值在正相位或负相位的有效宽度变为 0 或负数, TW0TO0 到 TW0TO5 以 0 有效宽度输出固定为无效电平的波形 (参阅图 6-7)。

然而, 如果设置 $TW0CM_n = 0$ 和 $TW0BFCM_n \geq TW0CM3$, 那么 TW0TO0 到 TW0TO5 输出有效电平波形。

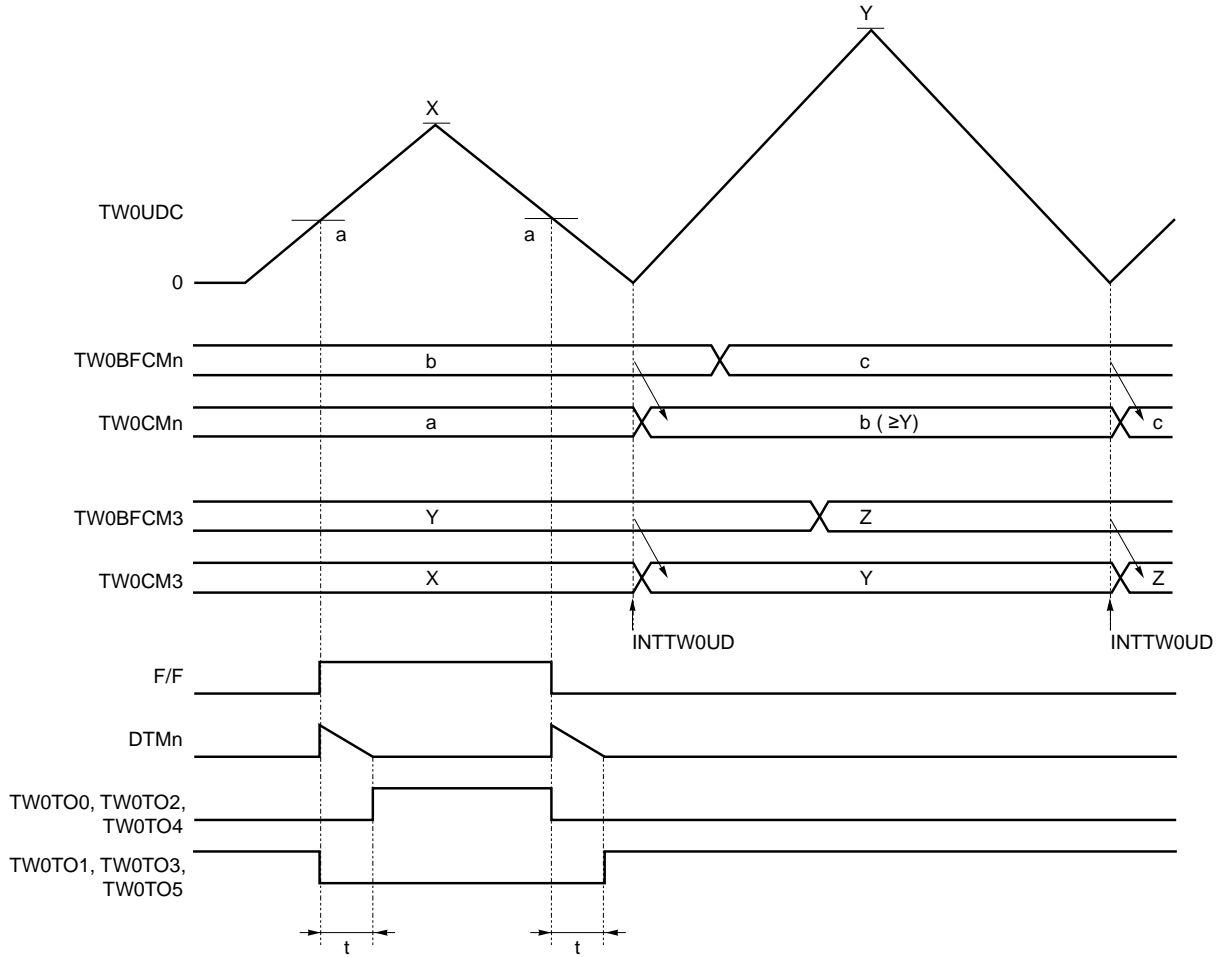
(3) 操作时序

图 6-6. TW0UDC 操作时序 (基本操作)



- 备注
1. $n = 0$ 到 2
 2. t : 死区时间 = $(TW0DIME + 1) \times 1/f_x$
(f_x : 系统时钟振荡频率)
 3. 上图假设高电平有效, 同时会产生未分频的 INTTW0UD。

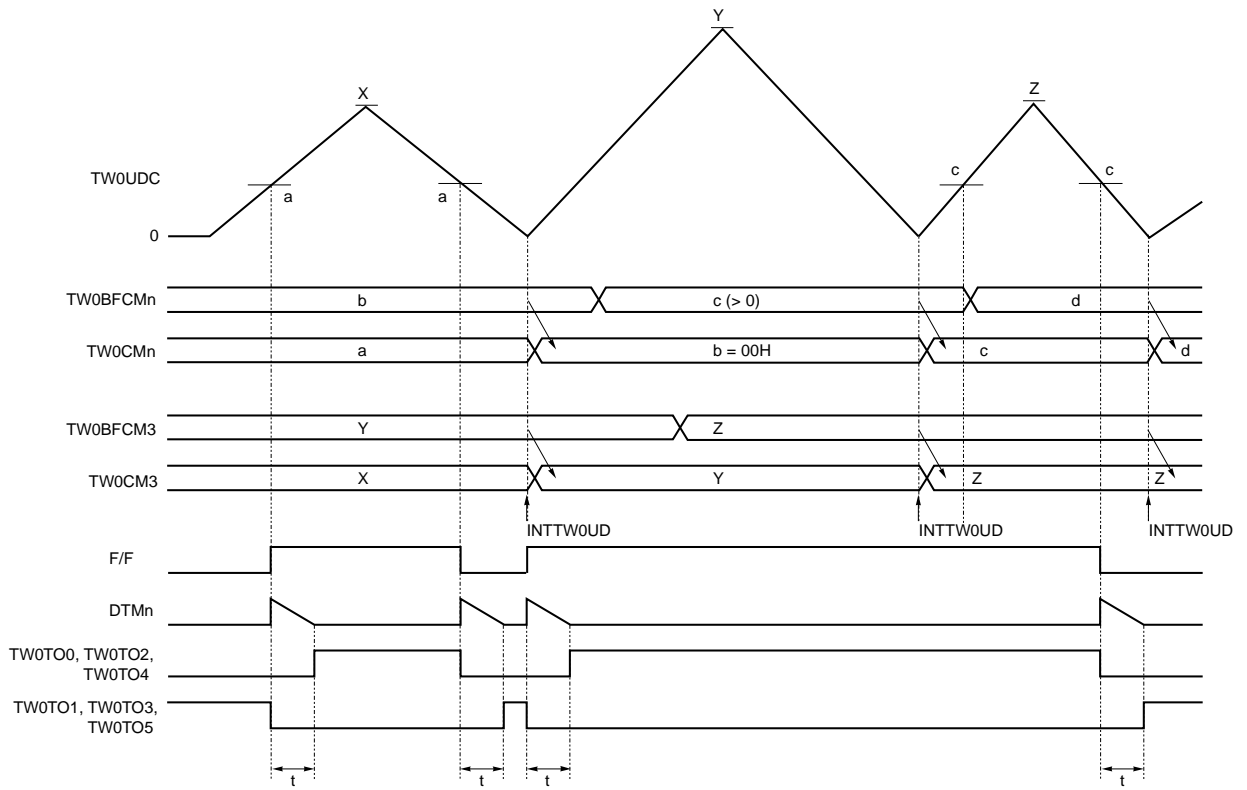
图 6-7. TW0UDC 操作时序(TW0CMn (TW0BFCMn) ≥ TW0CM3 (TW0BFCM3))



- 备注**
1. n = 0 到 2
 2. t: 死区时间 = $(TW0DIME + 1) \times 1/f_x$
(f_x : 系统时钟振荡频率)
 3. 上图假设高电平有效, 同时会产生未分频的 INTTW0UD。

如果大于 TW0CM3 的值设置到 TW0BFCMn 中, 正相位 (TW0TO0, TW0TO2, TW0TO4 引脚) 中的低电平输出和负相位 (TW0TO1, TW0TO3, TW0TO5 引脚) 中的高电平输出会继续。在控制逆变器等过程中输出低电平和
高电平宽度比 PWM 周期长的信号时, 该设置有效。

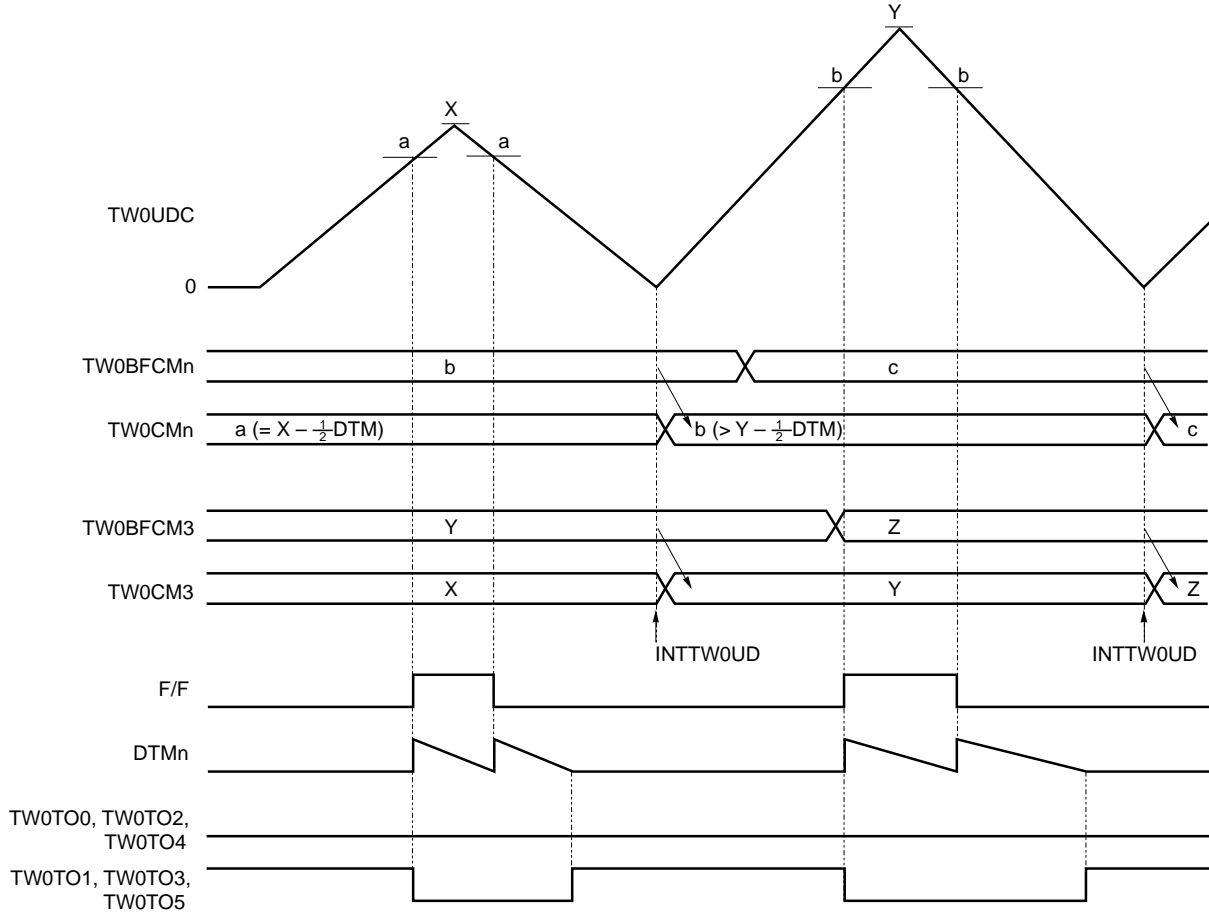
图 6-8. TW0UDC 操作时序(TW0CMn (TW0BFCMn) = 000H)



- 备注
1. $n = 0$ 到 2
 2. t : 死区时间 = $(TW0DIME + 1) \times 1/f_x$
(f_x : 系统时钟振荡频率)
 3. 上图假设高电平有效, 同时会产生未分频的 INTTW0UD。

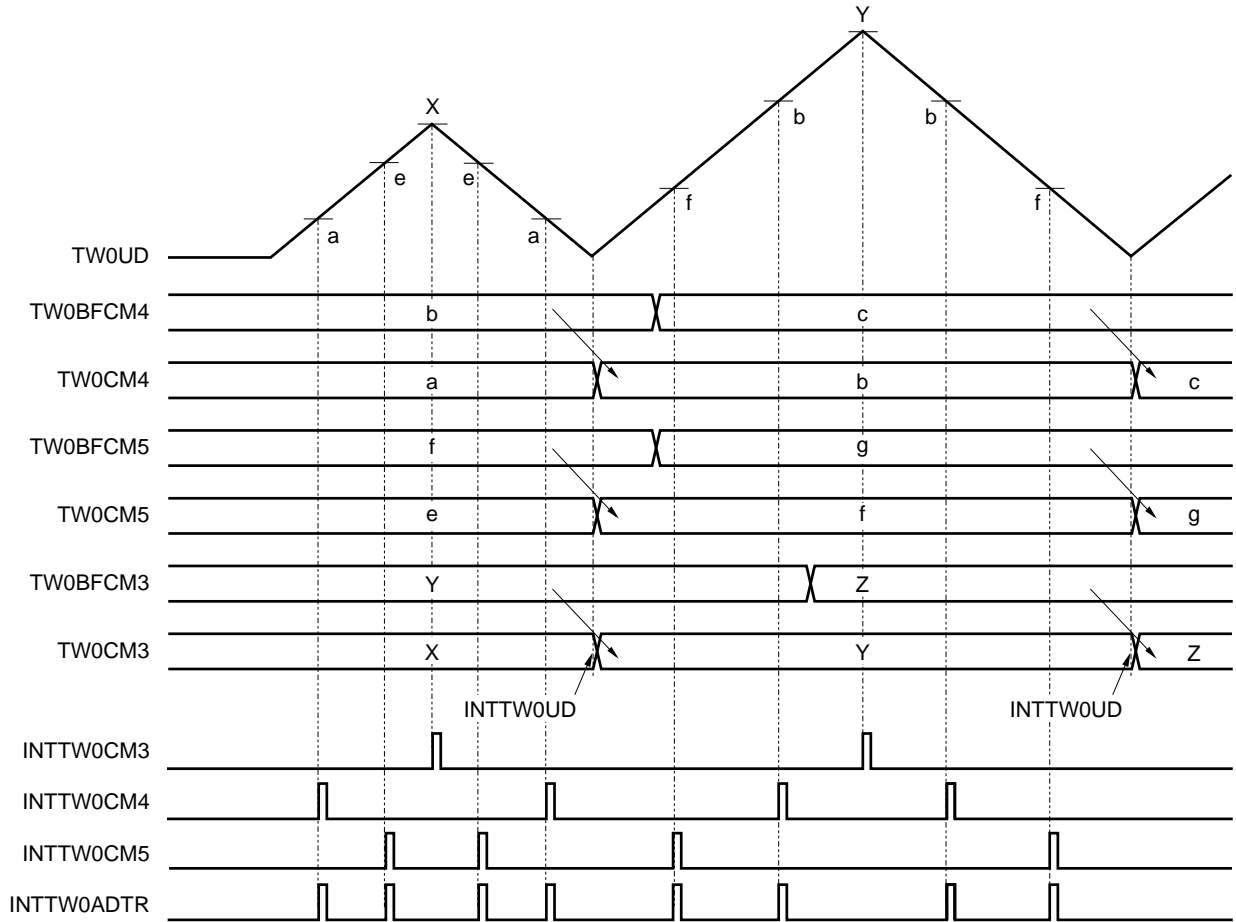
图 6-9. TW0UDC 操作时序

($TW0CMn(TW0BFCMn) = TW0CM3 - 1/2DTM$, $TW0CMn(TW0BFCMn) > TW0CM3 - 1/2DTM$)



- 备注
1. $n = 0$ 到 2
 2. 上图假设高电平有效，同时会产生未分频的INTTW0UD。

图 6-10. TW0UDC 操作时序 (IDEV02 到 IDEV00 = 000B, TW0TRGS = 03H)



第七章 16 位向上 / 向下计数器ITENC20

7.1 16 位向上 / 向下计数器ITENC20 的功能

16 位向上 / 向下计数器 ITENC20 具有以下功能。

- 通用定时器模式
 - 自由运行定时器
 - PWM 输出
- 向上 / 向下计数器模式
 - UDC 模式 A
 - UDC 模式 B
- 16 位 2 相编码器输入向上 / 向下计数器 & 通用定时器 (IT20UDC)： 1 通道
- 比较寄存器： 2
- 捕获 / 比较寄存器： 2
- 中断请求源
 - 捕获 / 比较匹配中断： 2
 - 比较匹配中断请求： 2
- 捕获请求信号： 2 类
 - 使用对应于捕获 / 比较寄存器的 TIT20CC0 和 TIT20CC1 引脚的有效沿作为捕获触发，IT20UDC 的值可以被锁存。
- 计数时钟可以从预分频器的分频时钟中选择（设置计数时钟频率为 10 MHz 或更小）
- 定时器 / 计数器时钟源： 2 类
 - （外部脉冲输入或内部系统时钟分频）
- 2 相编码器输入
 - 2 相外部编码器信号通过外部时钟输入引脚（TIT20IUD， TIT20CUD）用作定时器 / 计数器的计数时钟。计数器模式可以从以下四种模式中选择。
 - 模式 1： 对计数脉冲输入引脚（TIT20IUD）的输入脉冲进行计数。
向上 / 向下通过另一个输入引脚（TIT20CUD）的电平来指定。
 - 模式 2： 使用向上计数脉冲输入引脚和向下计数脉冲输入引脚的各自输入脉冲来向上 / 向下计数。
 - 模式 3： 使用输入到两个引脚的脉冲的相位关系来向上 / 向下计数。
 - 模式 4： 使用输入到两个引脚的脉冲的相位关系来向上 / 向下计数。 分别使用脉冲的上升沿和下降沿来完成计数。
- PWM 输出功能
 - 在通用定时器模式下，16 位分辨率 PWM 可以从 TIT20TO 引脚输出。

• 定时器清除

根据使用的模式，以下定时器清除操作被执行。

(a) 通用定时器模式：定时器清除操作可以在与 IT20CM0 设置值匹配时被执行。

(b) 向上 / 向下计数器模式：定时器清除操作可以在以下四种条件下选择。

(i) 在 IT20UDC 向上计数操作期间，定时器清除在与 IT20CM0 设置值匹配时被执行，并且在 IT20UDC 向下计数操作期间，与 IT20CM1 设置值匹配时被执行

(ii) 定时器清除只通过外部输入执行。

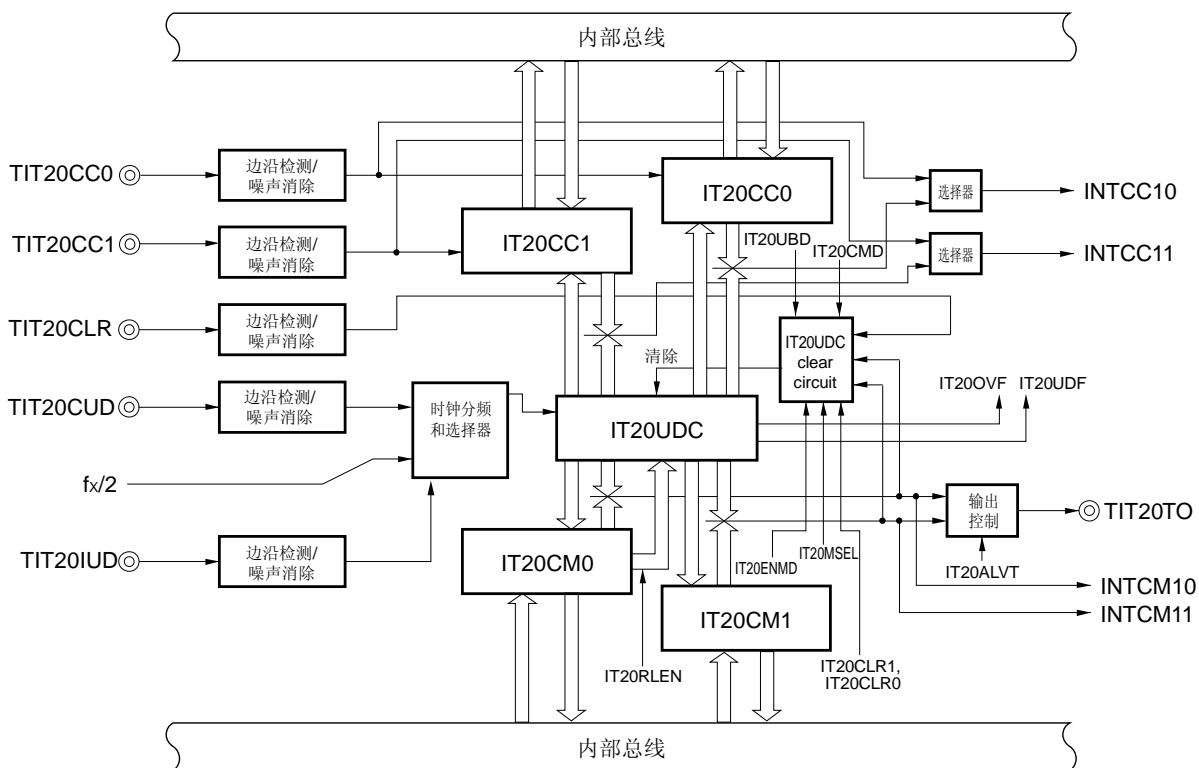
(iii) 定时器清除在 IT20UDC 计数值和 IT20CM0 设置值匹配发生时被执行。

(iv) 定时器清除在外部输入和 IT20UDC 计数值和 IT20CM0 设置值匹配发生时被执行。

• 外部脉冲输出 (TIT20TO) : 1

图 7-1 显示 16 位向上 / 向下计数器 ITENC20 的框图。

图 7-1. 16 位向上 / 向下计数器 ITENC20 的框图



7.2 16 位向上 / 向下计数器 ITENC20 的配置

16 位向上 / 向下计数器 ITENC20 由以下硬件组成。

表 7-1. 16 位向上 / 向下计数器 ITENC20 的配置

项目	配置	产生的中断信号	捕获触发
定时器计数器	16 位向上 / 向下计数器 (IT20UDC)	-	-
寄存器	16 位定时器比较寄存器 0 (IT20CM0)	INTCM10	-
	16 位定时器比较寄存器 1 (IT20CM1)	INTCM11	-
	16 位定时器捕获 / 比较寄存器 0 (IT20CC0)	INTCC10	TIT20CC0
	16 位定时器捕获 / 比较寄存器 1 (IT20CC1)	INTCC11	TIT20CC1
定时器输入	TIT20IUD, TIT20CUD, TIT20CC0, TIT20CC1, TIT20CLR	-	-
定时器输出	TIT20TO	-	-
控制寄存器	定时器单元模式寄存器 (IT20TUM)	-	-
	定时器控制寄存器 (IT20TMC)	-	-
	捕获 / 比较控制寄存器 (IT20CCR)	-	-
	有效沿选择寄存器 (IT20SESA)	-	-
	预分频器模式寄存器 (IT20PRM)	-	-
	状态寄存器 (IT20STS)	-	-
	噪声消除时间选择寄存器 1 (NRC1)	-	-
	端口模式寄存器 5 (PM5)	-	-
端口寄存器 5 (P5)	-	-	

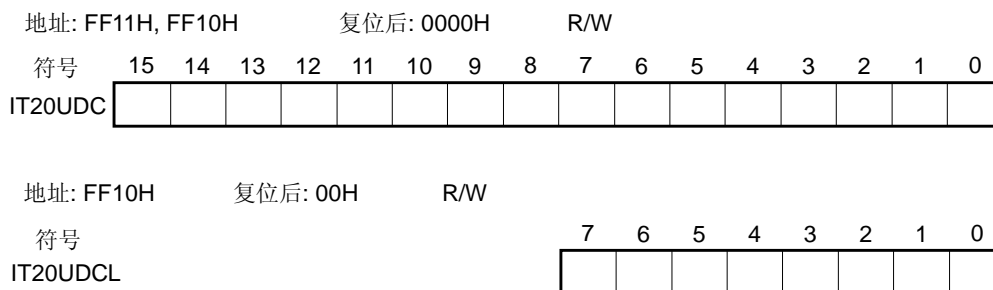
(1) 16 位向上 / 向下计数器 (IT20UDC)

IT20UDC 是 2 相编码器输入向上 / 向下计数器和通用定时器。

它可以由 16 位存储器操作指令读取或写入。

并且, 低 8 位可以由 8 位存储器操作指令以 IT20UDCL 来读取或写入。

$\overline{\text{RESET}}$ 输入设置 IT20UDC 位 0000H。



- 注意事项**
1. 只有当 IT20TMC 寄存器的 IT20CE 位为 0 (计数操作被禁止) 时才允许写入 IT20UDC。
 2. 禁止分别设置 IT20TUM 寄存器的 IT20CMD 位 (通用定时器模式) 和 IT20MSEL 位 (UDC 模式 B) 为 0 和 1。
 3. 禁止连续读取 IT20UDC。如果 IT20UDC 被连续读取, 第二个读取值可能与实际值不同。如果 IT20UDC 必须被读取两次, 确保在第一个和第二个读取操作之间读取另一个寄存器。
 4. 禁止写入相同的值到 IT20UDC、IT20CC0 和 IT20CC1 寄存器以及 IT20STS 寄存器。
允许写入相同的值到 IT20CCR、IT20TUM、IT20TMC、IT20SESA 和 IT20PRM 寄存器以及 IT20CM0 和 IT20CM1 寄存器 (即使在计数操作期间, 写入相同的值也被保证)。

IT20UDC 的启动和停止由定时器控制寄存器 (IT20TMC) 的 IT20CE 位来控制。

IT20UDC 操作由以下模式组成。

(a) 通用定时器模式

在通用定时器模式下, IT20UDC 作为 16 位间隔定时器、自由运行定时器或 PWM 输出来操作。

计数基于软件选择的时钟来执行。

使用预分频器模式寄存器 (IT20PRM) 的 IT20PRM2 到 IT20PRM0 位, 可以从 $f_x/2$ 、 $f_x/4$ 、 $f_x/8$ 、 $f_x/16$ 、 $f_x/32$ 、 $f_x/64$ 或 $f_x/128$ 的预分频器分频时钟中选择计数时钟 (f_x : 内部系统时钟)。

(b) 向上 / 向下计数器模式 (UDC 模式)

在 UDC 模式下, IT20UDC 用作 16 位向上 / 向下计数器, 并且基于 TIT20CUD 和 TIT20IUD 输入信号执行计数。

通过 IT20TUM 寄存器的 IT20MSEL 位, 可以为该模式选择两种操作模式。

(i) UDC 模式 A (当 IT20CMD 位 = 1, IT20MSEL 位 = 0 时)

通过设置 IT20TMC 寄存器的 IT20CLR1 和 IT20CLR0 位, IT20UDC 可以被清除。

(ii) UDC 模式 B (当 IT20CMD 位 = 1, IT20MSEL 位 = 1 时)

在 IT20UDC 向上计数操作期间, IT20UDC 在与 IT20CM0 匹配时被清除。

在 IT20UDC 向下计数操作期间, IT20UDC 在与 IT20CM1 匹配时被清除。

当 IT20TMC 寄存器的 IT20CE 位为 1 时，IT20UDC 在操作模式为通用模式时向上计数，在操作模式为 UDC 模式时向上 / 向下计数。

- 注意事项**
1. TIT20CUD 和 TIT20CC0 是复用引脚。因此，当 TIT20CUD 引脚用于 UDC 模式时，TIT20CC0 引脚的外部捕获功能不能被使用。
 2. TIT20CLR 和 TIT20CC1 是复用引脚。因此，当 TIT20CLR 输入用于 UDC 模式 A 时，TIT20CC1 引脚的外部捕获功能不能被使用。

清除 IT20UDC 的条件如下所示，依赖于操作模式。

表 7-2. 16 位向上 / 向下计数器 (IT20UDC) 的清除条件

操作模式	IT20TUM 寄存器		IT20TMC 寄存器			IT20UDC 清除
	IT20CMD 位	IT20MSEL 位	IT20ENMD 位	IT20CLR1 位	IT20CLR0 位	
通用定时器模式	0	0	0	×	×	清除不执行
			1	×	×	与 IT20CM0 设置值匹配时清除
UDC 模式 A	1	0	×	0	0	只由 TIT20CLR 输入清除
			×	0	1	向上计数操作期间，与 IT20CM0 设置值匹配时清除
			×	1	0	向上计数操作期间，由 TIT20CLR 输入或与 IT20CM0 设置值匹配时清除
			×	1	1	清除不执行
UDC 模式 B	1	1	×	×	×	向上计数操作期间，与 IT20CM0 设置值匹配时清除或者向下计数操作期间，与 IT20CM1 设置值匹配时清除
除上面以外						禁止设置

备注 ×: 表示该位的设置值被忽略。

(2) 比较寄存器 0 (IT20CM0)

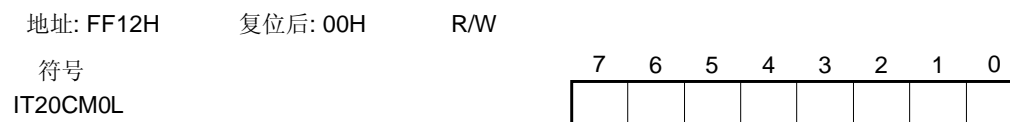
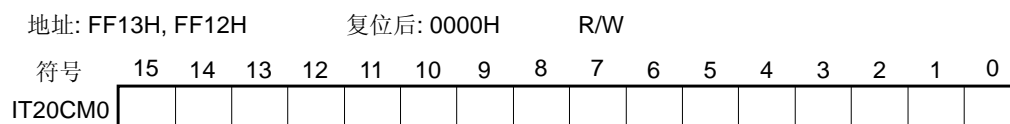
IT20CM0 是总是将其值与 IT20UDC 的值进行比较的 16 位寄存器。当比较寄存器的值与 IT20UDC 的值匹配时，才能产生一个中断信号。各种模式下的中断产生时序描述如下。

- 在通用定时器模式 (IT20TUM 寄存器的 IT20CMD 位 = 0) 和 UDC 模式 A (IT20TUM 寄存器的 IT20MSEL 位 = 0) 下，中断信号 (INTCM10) 总是在匹配发生时产生。
- 在 UDC 模式 B (IT20TUM 寄存器的 IT20MSEL 位 = 1) 下，中断信号 (INTCM10) 只有在向下计数期间的匹配发生时产生。

IT20CM0 可以由 16 位存储器操作指令读取或写入。

并且，低 8 位可以由 8 位存储器操作指令以 IT20CM0L 来读取或写入。

$\overline{\text{RESET}}$ 输入设置 IT20CM0 为 0000H。



注意事项 当 IT20TMC 寄存器的 IT20CE 位为 1 时，禁止覆盖 IT20CM0 寄存器的值。

(3) 比较寄存器 1 (IT20CM1)

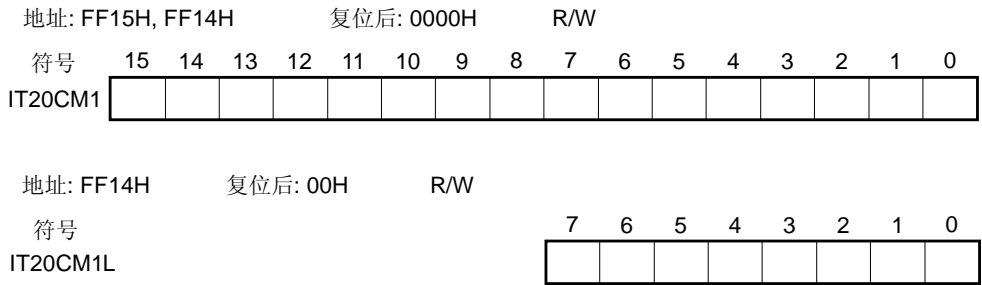
IT20CM1 是总是将其值与 IT20UDC 的值进行比较的 16 位寄存器。当比较寄存器的值与 IT20UDC 的值匹配时，才能产生一个中断信号。各种模式下的中断产生时序如下所示。

- 在通用定时器模式 (IT20TUM 寄存器的 IT20CMD 位 = 0) 和 UDC 模式 A (IT20TUM 寄存器的 IT20MSEL 位 = 0) 下，中断信号 (INTCM11) 总是在匹配发生时产生。
- 在 UDC 模式 B (IT20TUM 寄存器的 IT20MSEL 位 = 1) 下，中断信号 (INTCM11) 只有在向下计数期间的匹配发生时产生。

IT20CM1 可以由 16 位存储器操作指令读取或写入。

并且，低 8 位可以由 8 位存储器操作指令以 IT20CM1L 来读取或写入。

RESET 输入设置 IT20CM1 为 0000H。



注意事项 当 IT20TMC 寄存器的 IT20CE 位为 1 时，禁止覆盖 IT20CM1 寄存器的值。

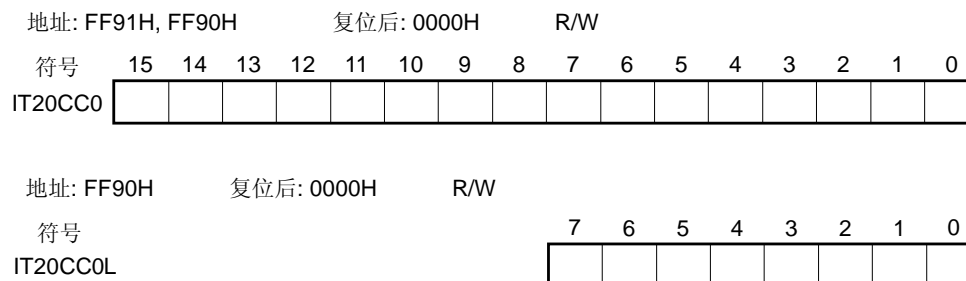
(4) 捕获 / 比较寄存器 0 (IT20CC0)

IT20CC0 是 16 位寄存器。使用捕获 / 比较控制寄存器 (IT20CCR)，它可以被指定为捕获寄存器或比较寄存器。

IT20CC0 可以由 16 位存储器操作指令读取或写入。

并且，低 8 位可以由 8 位存储器操作指令以 IT20CC0L 来读取或写入。

$\overline{\text{RESET}}$ 输入设置 IT20CC0 为 0000H。



- 注意事项**
1. 当用作捕获寄存器 (IT20CCR 寄存器的 IT20CMS0 位 = 0) 时，来自 CPU 的写入访问被禁止。
 2. 当用作比较寄存器 (IT20CCR 寄存器的 IT20CMS0 位 = 1) 并且 IT20TMC 寄存器的 IT20CE 位为 1 时，覆盖 IT20CC0 寄存器的值被禁止。
 3. 当 IT20TMC 寄存器的 IT20CE 位为 0 时，捕获触发被禁止。
 4. 当操作模式从捕获寄存器更改为比较寄存器时，设置一个新的比较值。
 5. 禁止连续读取 IT20CC0。如果 IT20CC0 被连续读取，第二个读取值可能与实际值不同。如果 IT20CC0 必须被读取两次，确保在第一个和第二个读取操作之间读取另一个寄存器。

(a) 当设置为捕获寄存器时

当 IT20CC0 被设置为捕获寄存器时，对应的 TIT20CC0 信号的有效沿被作为捕获触发检测。IT20UDC 与捕获触发同步锁存计数值 (捕获操作)。锁存的值被保持在捕获寄存器中，直到下一个捕获操作。外部中断的有效沿 (上升沿, 下降沿, 双边沿) 由有效沿选择寄存器 (IT20SESA) 来选择。当 IT20CC0 寄存器被指定为捕获寄存器时，中断在 TIT20CC0 信号的有效沿检测到时产生。

注意事项 TIT20CUD 和 TIT20CC0 是复用功能引脚。因此，当 TIT20CUD 引脚用于 UDC 模式时，TIT20CC0 引脚的外部捕获功能不能使用。

(b) 当设置为比较寄存器时

当 IT20CC0 被设置为比较寄存器时，它总是将其值与 IT20UDC 的值进行比较。如果 IT20CC0 的值与 IT20UDC 的值匹配，IT20CC0 将产生一个中断信号 (INTCC10)。

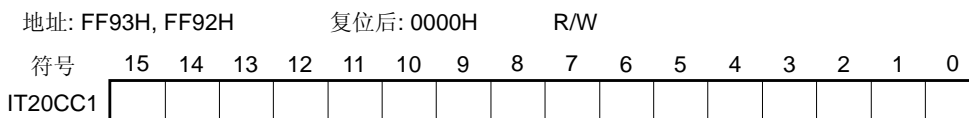
(5) 捕获 / 比较寄存器 1 (IT20CC1)

IT20CC1 是 16 位寄存器。使用捕获 / 比较控制寄存器 (IT20CCR)，它可以被指定为捕获寄存器或比较寄存器。

IT20CC1 可以由 16 位存储器操作指令读取或写入。

并且，低 8 位可以由 8 位存储器操作指令以 IT20CC1L 来读取或写入。

RESET 输入设置 IT20CC1 为 0000H。



- 注意事项
1. 当用作捕获寄存器 (IT20CCR 寄存器的 IT20CMS1 位 = 0) 时，来自 CPU 的写入访问被禁止。
 2. 当用作比较寄存器 (IT20CCR 寄存器的 IT20CMS1 位 = 1) 并且 IT20TMC 寄存器的 IT20CE 位为 1 时，覆盖 IT20CC1 寄存器的值被禁止。
 3. 当 IT20TMC 寄存器的 IT20CE 位为 0 时，捕获触发被禁止。
 4. 当操作模式从捕获寄存器更改为比较寄存器时，设置一个新的比较值。
 5. 禁止连续读取 IT20CC1。如果 IT20CC1 被连续读取，第二个读取值可能与实际值不同。如果 IT20CC1 必须被读取两次，确保在第一个和第二个读取操作之间读取另一个寄存器。

(a) 当设置为捕获寄存器时

当 IT20CC1 被设置为捕获寄存器时，对应的 TIT20CC1 信号的有效沿被作为捕获触发检测。IT20UDC 与捕获触发同步锁存计数值 (捕获操作)。锁存的值被保持在捕获寄存器中，直到下一个捕获操作。外部中断的有效沿 (上升沿，下降沿，双边沿) 由有效沿选择寄存器 (IT20SESA) 来选择。当 IT20CC1 寄存器被指定为捕获寄存器时，中断在 TIT20CC1 信号的有效沿检测时被产生。

注意事项 TIT20CLR 和 TIT20CC1 是复用功能引脚。因此，当 TIT20CLR 引脚用于 UDC 模式 A 时，TIT20CC1 引脚的外部捕获功能不能使用。

(b) 当设置为比较寄存器时

当 IT20CC1 被设置为比较寄存器时，它总是将其值与 IT20UDC 的值进行比较。如果 IT20CC1 的值与 IT20UDC 的值匹配，IT20CC1 将产生一个中断信号 (INTCC11)。

7.3 16 位向上 / 向下计数器 ITENC20 的控制寄存器

16 位向上 / 向下计数器 ITENC20 由以下九个寄存器控制。

- 定时器单元模式寄存器 (IT20TUM)
- 定时器控制寄存器 (IT20TMC)
- 捕获 / 比较控制寄存器 (IT20CCR)
- 有效沿选择寄存器 (IT20SESA)
- 预分频器模式寄存器 (IT20PRM)
- 状态寄存器 (IT20STS)
- 噪声消除时间选择寄存器 1 (NRC1)
- 端口模式寄存器 5 (PM5)
- 端口寄存器 5 (P5)

(1) 定时器单元模式寄存器 (IT20TUM)

IT20TUM 是一个用来指定 IT20UDC 操作模式或控制 PWM 输出引脚操作的 8 位寄存器。

该寄存器可以由 1 位或 8 位存储器操作指令来读取或写入。

RESET 输入设置 IT20TUM 为 00H。

图 7-2. 定时器单元模式寄存器 (IT20TUM) 的格式

地址: FF95H	复位后: 00H	R/W						
符号	<7>	6	5	4	<3>	<2>	1	<0>
IT20TUM	IT20CMD	0	0	0	IT20TOE	IT20ALVT	0	IT20MSEL

IT20CMD	IT20UDC 操作模式指定
0	通用定时器模式 (向上计数)
1	UDC 模式 (向上/向下计数)

IT20TOE	定时器输出 (TIT20TO) 允许的指定
0	定时器输出禁止
1	定时器输出允许
当 IT20CMD 位 = 1 (UDC 模式) 时, 无论 IT20TOE 位的设置为何, 定时器输出都不被执行。此时, 定时器输出是由 IT20ALVT 位设置的电平的反转相位电平。	

IT20ALVT	定时器输出 (TIT20TO) 有效电平的指定
0	有效电平为高电平
1	有效电平为低电平
当 IT20CMD 位 = 1 (UDC 模式) 时, 无论 IT20TOE 位的设置为何, 定时器输出都不被执行。此时, 定时器输出是由 IT20ALVT 位设置的电平的反转相位电平。	

IT20MSEL	UDC 模式 (向上/向下计数) 下操作的指定
0	UDC 模式 A IT20UDC 可以通过设置 IT20TMC 寄存器的 IT20CLR1 和 IT20CLR0 位来清除。
1	UDC 模式 B IT20UDC 在以下情况下被清除。 · IT20UDC 向上计数操作期间与 IT20CM0 匹配时 · IT20UDC 向下计数操作期间与 IT20CM1 匹配时
当 UDC 模式 B 被设置时, IT20TMC 寄存器的 IT20ENMD、IT20CLR1 和 IT20CLR0 位变为无效。	

- 注意事项 1. 在 IT20UDC 操作 (IT20TMC 寄存器的 IT20CE 位 = 1) 期间, 禁止更改 IT20TUM 寄存器的值。
 2. 当 IT20CMD 位 = 0 (通用定时器模式) 时, 禁止设置 IT20MSEL = 1 (UDC 模式 B)。

(2) 定时器控制寄存器 (IT20TMC)

IT20TMC 寄存器用来允许 / 禁止 IT20UDC 操作并设置传输和定时器清除操作。

该寄存器可以由 1 位或 8 位存储器操作指令来读取或写入。

RESET 输入设置 IT20TMC 到 00H。

图 7-3. 定时器控制寄存器 (IT20TMC) 的格式

地址: FF96H	复位后: 00H	R/W						
符号	7	<6>	5	4	3	2	1	0
IT20TMC	0	IT20CE	0	0	IT20RLEN	IT20ENMD	IT20CLR1	IT20CLR0

IT20CE	IT20UDC 操作控制
0	计数操作禁止
1	计数操作允许

IT20RLEN	IT20CM0 到 IT20UDC 传输操作的指定
0	传输操作禁止
1	传输操作允许
<ul style="list-style-type: none"> · 当IT20RLEN = 1时, 设置到IT20CM0的值在IT20UDC下溢发生时被传输到IT20UDC。 · 当IT20TUM寄存器的IT20CMD位 = 0 (通用定时器模式), IT20RLEN位设置变为无效。 · IT20RLEN位只有在UDC模式A (IT20TUM寄存器的IT20CMD位 = 1, IT20MSEL位 = 0)下才有效。在通用定时器模式 (IT20CMD位 = 0) 和UDC模式B (IT20CMD位 = 1, IT20MSEL位 = 1)时, 即使IT20RLEN位被置位(1), 传输也不被执行。 	

IT20ENMD	通用定时器模式下IT20UDC清除操作的控制
0	清除禁止 (自由运行模式) 即使当IT20UDC和IT20CM0值匹配时, 清除也不被执行。
1	清除允许 当IT20UDC和IT20CM0值匹配时, 清除被执行。
当IT20TUM寄存器的IT20CMD位 = 1 (UDC模式)时, IT20ENMD位设置变为无效。	

IT20CLR1	IT20CLR0	IT20UDC 清除源指定
0	0	只由外部输入(TIT20CLR)清除
0	1	IT20UDC计数值和IT20CM0设置值匹配时清除
1	0	由TIT20CLR或IT20UDC计数值和IT20CM0设置值匹配时清除
1	1	不清除
<ul style="list-style-type: none"> · IT20UDC计数值和IT20CM0设置值匹配时清除只有在IT20UDC向上计数操作期间才有效(IT20UDC在T20UDC向下计数操作期间不被清除)。 · 当IT20TUM寄存器的IT20CMD位 = 0 (通用定时器模式)时, IT20CLR1和IT20CLR0位设置无效。 · 当IT20TUM寄存器的IT20CMD位 = 1 (UDC模式B)时, IT20CLR1和IT20CLR0位设置无效。 · 当通过位IT20CLR1和IT20CLR0由TIT20CLR清除被允许时, 无论IT20CE位的值是1还是0, 清除都被执行。 		

注意事项 在 IT20UDC 操作 (IT20CE = 1) 期间, 禁止更改 IT20TMC 寄存器中 IT20CE 位以外的位的值。

(3) 捕获 / 比较控制寄存器 (IT20CCR)

IT20CCR 寄存器指定捕获 / 比较寄存器 (IT20CC0, IT20CC1) 的操作模式。

该寄存器可以由 1 位或 8 位存储器操作指令来读取或写入。

RESET 输入设置 IT20CCR 为 00H。

图 7-4. 捕获 / 比较控制寄存器 (IT20CCR) 的格式

地址: FF94H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	<1>	<0>
IT20CCR	0	0	0	0	0	0	IT20CMS1	IT20CMS0

IT20CMS1	IT20CC1 操作模式指定
0	作为捕获寄存器操作
1	作为比较寄存器操作

IT20CMS0	IT20CC0 操作模式指定
0	作为捕获寄存器操作
1	作为比较寄存器操作

- 注意事项
1. 在 IT20UDC 操作 (IT20CE 位 = 1) 期间, 禁止覆盖 IT20CCR 寄存器。
 2. TIT20CUD 和 TIT20CC0 是复用功能引脚。因此, 当 TIT20CUD 引脚用于 UDC 模式时, TIT20CC0 引脚的外部捕获功能不能使用。
 3. TIT20CLR 和 TIT20CC1 是复用功能引脚。因此, 当 TIT20CLR 引脚用于 UDC 模式 A 时, TIT20CC1 引脚的外部捕获功能不能使用。

(4) 有效沿选择寄存器 (IT20SESA)

IT20SESA 寄存器用来指定从外部引脚 (TIT20CC0, TIT20CC1, TIT20IUD, TIT20CUD, TIT20CLR) 输入的外部中断请求的有效沿。

可以为每个引脚单独指定有效沿 (上升沿, 下降沿, 或双边沿)。

该寄存器可以由 1 位或 8 位存储器操作指令来读取或写入。

RESET 输入设置 IT20SESA 为 00H。

图 7-5. 有效沿选择寄存器 (IT20SESA) 的格式



注意事项 在 IT20UDC 操作 (IT20CE = 1) 期间, 禁止更改 IT20SESA 寄存器位的值。

(5) 预分频器模式寄存器 (IT20PRM)

IT20PRM 寄存器用来执行以下选择。

- 通用定时器模式 (IT20TUM 寄存器的 IT20CMD 位 = 0) 下的计数时钟的选择
- UDC 模式 (IT20CMD = 1) 下的计数操作模式的选择

该寄存器可以由 1 位或 8 位存储器操作指令来读取或写入。

RESET 输入设置 IT20PRM 为 07H。

图 7-6. 预分频器模式寄存器 (IT20PRM) 的格式

地址: FF3AH	复位后: 07H	R/W						
符号	7	6	5	4	3	2	1	0
IT20PRM	0	0	0	0	0	IT20PRM2	IT20PRM1	IT20PRM0

IT20PRM2	IT20PRM1	IT20PRM0	IT20CMD = 0	IT20CMD = 1	
			计数时钟	计数时钟	向上/向下计数
0	0	0	禁止设置	禁止设置 (模式 4) (此时, IT20SESA 寄存器被允许。)	
0	0	1	fx/2		
0	1	0	fx/4		
0	1	1	fx/8		
1	0	0	fx/16	TIT20IUD	模式 1
1	0	1	fx/32		模式 2
1	1	0	fx/64		模式 3
1	1	1	fx/128		模式 4

备注 fx: 内部系统时钟

- 注意事项
1. 在 IT20UDC 操作 (IT20CE = 1) 期间, 禁止重写 IT20PRM 寄存器的值。
 2. 当 IT20TUM 寄存器的 IT20CMD 位 = 1 (UDC 模式) 时, 禁止设置 IT20PRM2 到 IT20PRM0 的值为 000、001、010 和 011。
 3. 当 IT20UDC 处于模式 4 时, 对 TIT20IUD 和 TIT20CUD 引脚的有效沿的指定无效。

(a) 处于通用定时器模式 (IT20TUM 寄存器的 IT20CMD 位 = 0)

计数时钟固定为内部时钟。IT20UDC 的时钟速率由 IT20PRM2 到 IT20PRM0 位指定。

(b) UDC 模式 (IT20TUM 寄存器的 IT20CMD 位 = 1)

UDC 模式下, IT20UDC 计数触发如下所示。

操作模式	IT20UDC 操作
模式 1	当 TIT20CUD = 高电平时, 向下计数 当 TIT20CUD = 低电平时, 向上计数
模式 2	TIT20IUD 输入的有效沿检测时, 向上计数 TIT20CUD 输入的有效沿检测时, 向下计数
模式 3	TIT20IUD 输入的有效沿检测时, 通过 TIT20CUD 输入电平自动判断
模式 4	TIT20IUD 输入的双边沿和 TIT20CUD 输入的双边沿检测时, 自动判断

(6) 状态寄存器 (IT20STS)

IT20STS 寄存器表示 IT20UDC 的操作状态。

该寄存器可以由 1 位或 8 位存储器操作指令来读取。

RESET 输入设置 IT20STS 为 00H。

图 7-7. 状态寄存器 (IT20STS) 的格式

地址: FF3BH	复位后: 00H	R						
符号	7	6	5	4	3	<2>	<1>	<0>
IT20STS	0	0	0	0	0	IT20UDF	IT20OVF	IT20UBD

IT20UDF	IT20UDC 下溢标志
0	无IT20UDC计数下溢
1	IT20UDC计数下溢
从CPU对IT20STS 寄存器的读取访问完成时, IT20UDF位被清除(0)。	

IT20OVF	IT20UDC 溢出标志
0	无IT20UDC计数溢出
1	IT20UDC计数溢出
从CPU对IT20STS 寄存器的读取访问完成时, IT20OVF位被清除(0)。	

IT20UBD	IT20UDC 向上/向下计数操作状态
0	IT20UDC 向上计数处理中
1	IT20UDC 向下计数处理中
IT20UBD位的状态根据模式而不同, 如下所示。 · 当IT20TUM 寄存器的IT20CMD 位 = 0 (通用定时器模式)时, IT20UBD位由硬件固定为0。 · 当IT20TUM 寄存器的IT20CMD 位 = 1 (UDC 模式)时, IT20UBD位表示IT20UDC向上/向下计数状态。	

注意事项 在 IT20UDC 操作 (IT20CE 位 = 1) 期间, 禁止覆盖 IT20STS 寄存器。

(7) 噪声消除时间选择寄存器 1 (NRC1)

NRC1 寄存器选择用于消除 TIT20IUD、TIT20CUD、TIT20CC0、TIT20CC1 或 TIT20CLR 引脚上的数字噪声的采样时钟。如果由 NRC1 寄存器选定的时钟连续 5 次没有在这些引脚上检测到某电平，那么此信号就作为噪声消除。

该寄存器可以由 1 位或 8 位存储器操作指令来读取或写入。

RESET 输入清除该寄存器为 00H。

图 7-8. 噪声消除时间选择寄存器 1 (NRC1) 的格式

地址: FFAAH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
NRC1	0	0	0	0	0	0	NRC11	NRC10

NRC11	NRC10	采样时钟选择
0	0	$f_x/2^3$
0	1	$f_x/2^2$
1	0	$f_x/2$
1	1	f_x

备注 f_x : 内部系统时钟

- 注意事项**
1. 如果输入脉冲持续 4 到 5 个时钟，脉冲是被作为有效沿检测还是作为噪声消除将不确定。所以，脉冲实际上要作为有效沿检测，脉冲电平必须被输入 5 个或更多时钟。
 2. 如果噪声与采样时钟同步产生，通过对输入引脚使用滤波器来消除噪声。
 3. 如果引脚用作正常输入端口引脚，噪声将不被消除。

(8) 端口模式寄存器 5 (PM5)

该寄存器以 1 位为单位设置端口 5 输入 / 输出。

当使用 P57/TIT20CLR/TIT20CC1/TIT20TO 引脚作为定时器输出时，清除 PM57 和 P57 的输出锁存为 0。

当使用 P55/TIT20IUD/INTP6、P56/TIT20CUD/TIT20CC0/INTP7 和 P57/TIT20CLR/TIT20CC1/TIT20TO 引脚作为定时器输入时，设置 PM55、PM56 和 PM57 为 1。此时，P55、P56 和 P57 的输出锁存可能为 0 或 1。

PM5 可以由 1 位或 8 位存储器操作指令来设置。

RESET 输入设置该寄存器为 FFH。

图 7-9. 端口模式寄存器 5 (PM5) 的格式

地址: FF25H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

PM5n	PM5n 硬件 I/O 模式选择 (n = 0 到 7)
0	输出模式 (输出缓存打开)
1	输入模式 (输出缓存关闭)

7.4 16 位向上 / 向下计数器 ITENC20 的操作

7.4.1 基本操作

可以对 16 位向上 / 向下计数器 ITENC20 选择以下两种操作模式。

(1) 通用定时器模式 (IT20TUM 寄存器的 IT20CMD 位 = 0)

在通用定时器模式下, 该计数器或者作为 16 位间隔定时器或者作为 PWM 输出定时器操作 (计数操作只能是向上计数)。

到 IT20UDC 的计数时钟由预分频器模式寄存器 (IT20PRM) 来选择。

(2) 向上 / 向下计数器模式 (UDC 模式) (IT20TUM 寄存器的 IT20CMD 位 = 1)

在 UDC 模式下, 该计数器作为 16 位向上 / 向下计数器操作。

由 IT20PRM 寄存器设置的外部时钟输入 (TIT20IUD, TIT20CUD 引脚) 被用作 IT20UDC 的计数时钟。

根据 IT20UDC 清除条件, UDC 模式被进一步分为两种模式。

• UDC 模式 A (IT20TUM 寄存器的 IT20CMD 位 = 1, IT20MSEL 位 = 0)

使用 IT20TMC 寄存器的 IT20CLR1 和 IT20CLR0 位, IT20UDC 清除源只能选作外部清除输入 (TIT20CLR)、向上计数操作期间 IT20UDC 的计数值和 IT20CM0 的设置值之间的匹配信号或者两个信号的逻辑和 (OR)。在 IT20UDC 下溢发生时, IT20UDC 可以重新加载 IT20CM0 的值。

• UDC 模式 B (IT20TUM 寄存器的 IT20CMD 位 = 1, IT20MSEL 位 = 1)

IT20UDC 的计数值和 IT20CM0 的设置值之间匹配后, IT20UDC 的状态如下所示。

<1> 在向上计数操作的情况下, IT20UDC 被清除 (0000H) 并且产生 INTCM10 中断。

<2> 在向下计数操作的情况下, IT20UDC 的计数值递减 (-1)。

IT20UDC 的计数值和 IT20CM1 的设置值之间匹配后, IT20UDC 的状态如下所示。

<1> 在向上计数操作的情况下, IT20UDC 的计数值递增 (+1)。

<2> 在向下计数操作的情况下, IT20UDC 被清除 (0000H) 并且产生 INTCM11 中断。

7.4.2 通用定时器模式下的操作

在通用定时器模式下，16 位向上 / 向下计数器 ITENC20 可以执行以下操作。

(1) 间隔操作

IT20UDC 和 IT20CM0 总是比较它们的值，并且当匹配发生时，INTCM10 中断会产生。

IT20UDC 按匹配之后的计数时钟被清除 (0000H)。

此外，当又一个计数时钟输入时，IT20UDC 向上计数为 0001H。间隔时间可以按照以下公式计算。

$$\text{间隔时间} = (\text{IT20CM0 值} + 1) \times \text{IT20UDC 计数时钟速率}$$

注意事项 通过设置 IT20TMC 寄存器的 IT20ENMD 位为 1，间隔操作可以实现。

(2) 自由运行操作

IT20UDC 执行一个从 0000H 到 FFFFH 的完全计数操作，并且在 IT20STS 寄存器的 IT20OVF 位被置位 (为 1) 后，IT20UDC 被清除并重新计数。自由运行周期可以按照以下公式计算。

$$\text{自由运行周期} = 65,536 \times \text{IT20UDC 计数时钟速率}$$

注意事项 通过设置 IT20TMC 寄存器的 IT20ENMD 位为 0，自由运行操作可以实现。

(3) 比较功能

IT20UDC 连接两个比较寄存器 (IT20CM0, IT20CM1) 通道和两个捕获 / 比较寄存器 (IT20CC0, IT20CC1) 通道。

当 IT20UDC 的计数值和一个比较寄存器的设置值匹配时，一个匹配中断 (INTCM10, INTCM11, INTCC10^注, INTCC11^注) 会输出。

特别在间隔操作的情况下，IT20UDC 在 INTCM10 中断产生时被清除。

注 当 IT20CC0 和 IT20CC1 设置为比较寄存器模式时，该匹配中断产生。

(4) 捕获功能

IT20UDC 连接两个捕获 / 比较寄存器 (IT20CC0, IT20CC1) 通道。

当 IT20CC0 和 IT20CC1 设置为捕获寄存器模式时, IT20UDC 的值与对应的捕获触发信号同步捕获。

此外, 一个中断请求 (INTCC10, INTCC11) 由 TIT20CC0 和 TIT20CC1 输入信号产生。

表 7-3. 到 16 位捕获寄存器的捕获触发信号

捕获寄存器	捕获触发信号
IT20CC0	TIT20CC0
IT20CC1	TIT20CC1

备注 IT20CC0 和 IT20CC1 是捕获 / 比较寄存器。使用这些寄存器中的哪一个通过捕获 / 比较寄存器 (IT20CCR) 来指定。

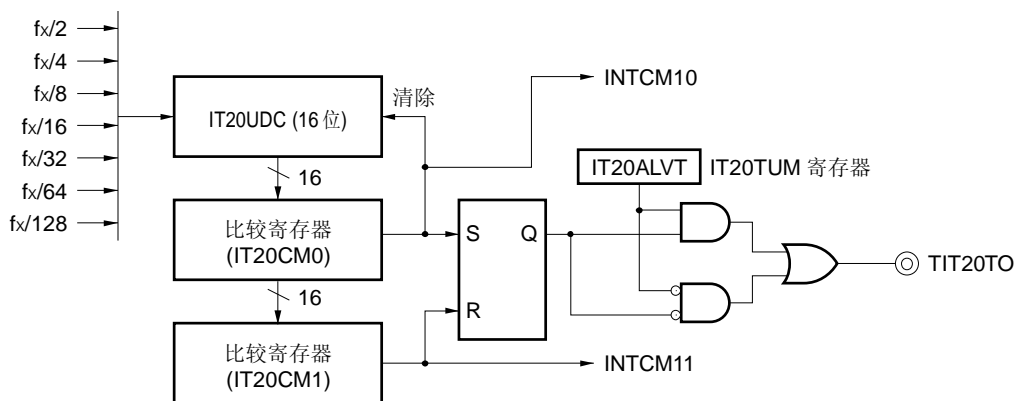
捕获触发的有效沿由有效沿选择寄存器 (IT20SESA) 来指定。如果双边沿被选作捕获触发, 可以测量外部输入脉冲的宽度。如果单边沿被选作捕获触发, 输入脉冲周期被测量。

(5) PWM 输出操作

通过使用定时器单元模式寄存器 (IT20TUM) 设置 IT20UDC 为通用定时器模式 (IT20CMD 位 = 0), 从 TIT20TO 引脚的 PWM 输出操作将会执行。

分辨率为 16 位, 并且计数时钟可以从七个内部时钟 (fx/2, fx/4, fx/8, fx/16, fx/32, fx/64, fx/128) 中选择。

图 7-10. PWM 输出操作的框图



注意事项 确保设置 IT20UDC 的计数时钟为 10 MHz 或更低。

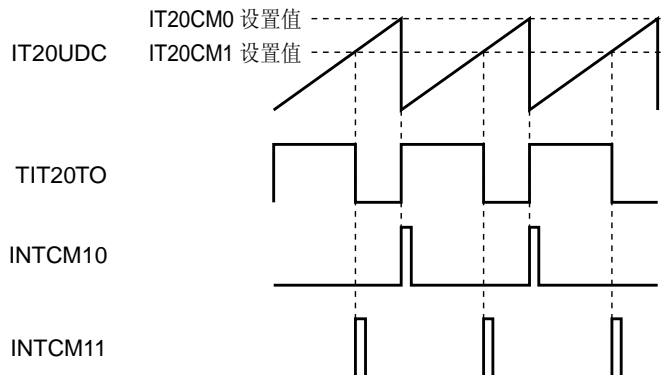
备注 fx: 内部系统时钟

(a) 操作说明

IT20CM0 寄存器是用作设置 PWM 输出周期的比较寄存器。当该寄存器的值与 IT20UDC 的值匹配时，INTCM10 中断就会产生。比较匹配由硬件保存，并且 IT20UDC 在匹配后的下一个计数时钟时被清除。

IT20CM1 是用作设置 PWM 输出占空比的比较寄存器。对 PWM 周期设置需要的占空比。

图 7-11. PWM 信号输出示例（当 IT20ALVT 位 = 0 时）



- 注意事项**
1. 在 IT20UDC 操作（IT20TMC 寄存器的 IT20CE 位 = 1）期间，禁止更改 IT20CM0 和 IT20CM1 寄存器的值。
 2. 在 IT20UDC 操作期间，禁止更改 IT20TUM 寄存器的 IT20ALVT 位的值。
 3. PWM 信号输出从 IT20CE 位被置位（为 1）后的第二个 PWM 周期开始执行。

7.4.3 UDC模式下的操作

(1) UDC 模式下的操作概述

在 UDC 模式 (IT20TUM 寄存器的 IT20CMD 位 = 1) 下, 输入到 IT20UDC 的计数时钟只能为 TIT20IUD 和 TIT20CUD 引脚的外部输入。根据 IT20PRM 寄存器的设置 (总共四种选择), UDC 模式下向上 / 向下计数的判别基于 TIT20IUD 和 TIT20CUD 输入的相位差来确定。

表 7-4. UDC 模式下的计数操作列表

IT20PRM 寄存器			操作模式	IT20UDC 操作
IT20PRM2	IT20PRM1	IT20PRM0		
1	0	0	模式 1	当 TIT20CUD = 高电平时, 向下计数 当 TIT20CUD = 低电平时, 向上计数
1	0	1	模式 2	TIT20IUD 输入的有效沿检测时, 向上计数 TIT20CUD 输入的有效沿检测时, 向下计数
1	1	0	模式 3	TIT20IUD 输入的有效沿检测时, 通过 TIT20CUD 输入电平自动判断
1	1	1	模式 4	TIT20IUD 输入的双边沿和 TIT20CUD 输入的双边沿检测时, 自动判断

根据 IT20UDC 清除条件, UDC 模式被进一步分为两种模式 (在两种模式下, 计数操作只按照 TIT20IUD 和 TIT20CUD 输入执行)。

- UDC 模式 A (IT20TUM 寄存器的 IT20CMD 位 = 1, IT20MSEL 位 = 0)

使用 IT20TMC 寄存器的 IT20CLR1 和 IT20CLR0 位, IT20UDC 清除源只能选作外部清除输入 (TIT20CLR)、向上计数操作期间 IT20UDC 的计数值和 IT20CM0 的设置值之间的匹配信号或者两个信号的逻辑和 (OR)。在 IT20UDC 下溢发生时, IT20UDC 可以传输 IT20CM0 的值。

- UDC 模式 B (IT20TUM 寄存器的 IT20CMD 位 = 1, IT20MSEL 位 = 1)

IT20UDC 的计数值和 IT20CM0 的设置值之间匹配后, IT20UDC 的状态如下所示。

- <1> 在向上计数操作的情况下, IT20UDC 被清除 (0000H) 并且 INTCM10 中断将会产生。
- <2> 在向下计数操作的情况下, IT20UDC 的计数值递减 (-1)。

IT20UDC 的计数值和 IT20CM1 的设置值之间匹配后, IT20UDC 的状态如下所示。

- <1> 在向上计数操作的情况下, IT20UDC 的计数值递增 (+1)。
- <2> 在向下计数操作的情况下, IT20UDC 被清除 (0000H) 并且 INTCM11 中断将会产生。

(2) UDC 模式下的向上 / 向下计数操作

根据 IT20PRM 寄存器的设置，UDC 模式下向上 / 向下计数的判别基于 TIT20IUD 和 TIT20CUD 输入的相位差来确定。

(a) 模式 1 (IT20PRM2 位 = 1, IT20PRM1 位 = 0, IT20PRM0 位 = 0)

在模式 1 下，TIT20IUD 引脚的有效沿检测时，以下计数操作基于 TIT20CUD 引脚的电平来执行。

- 当 TIT20CUD 引脚 = 高电平时，IT20UDC 向下计数操作
- 当 TIT20CUD 引脚 = 低电平时，IT20UDC 向上计数操作

图 7-12. 模式 1 (当上升沿被指定为 TIT20IUD 引脚的有效沿时)

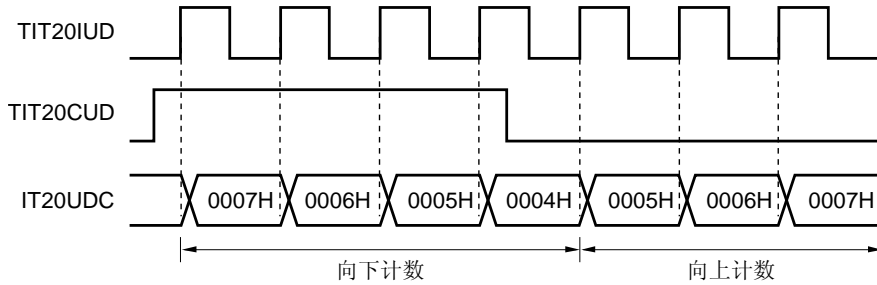
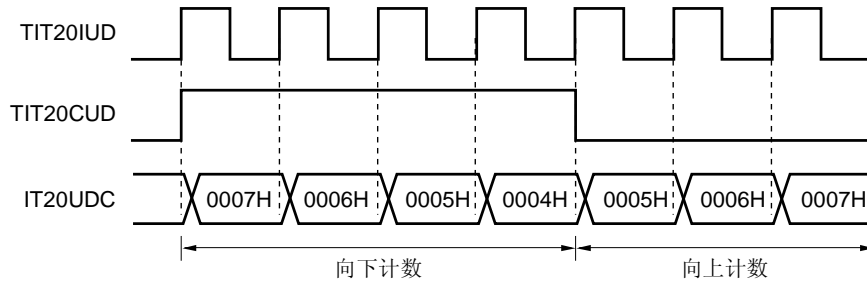


图 7-13. 模式 1 (当上升沿被指定为 TIT20IUD 引脚的有效沿时)：
TIT20IUD, TIT20CUD 引脚边沿同时发生时



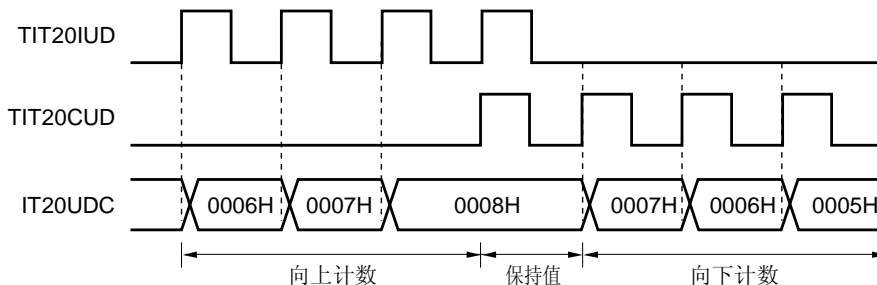
(b) 模式 2 (IT20PRM2 位 = 1, IT20PRM1 位 = 0, IT20PRM0 位 = 1)

模式 2 下的计数操作如下所示。

- 当 TIT20IUD 引脚的有效沿检测时, IT20UDC 向上计数
- 当 TIT20CUD 引脚的有效沿检测时, IT20UDC 向下计数

注意事项 如果计数时钟同时输入到 TIT20IUD 引脚和 TIT20CUD 引脚时, 计数操作不会被执行并且前面的值会被保持下来。

图 7-14. 模式 2 (当上升沿被指定为 TIT20IUD, TIT20CUD 引脚的有效沿时)



(c) 模式 3 (IT20PRM2 = 1, IT20PRM1 = 1, IT20PRM0 = 0)

在模式 3 下, 当相位相差 90 度的两个信号被输入到 TIT20IUD 和 TIT20CUD 引脚时, TIT20CUD 的电平在 TIT20IUD 引脚的有效沿输入时被采样 (参见图 7-15)。

如果在 TIT20IUD 引脚的有效沿输入时采样的 TIT20CUD 引脚电平为低, 当有效沿输入到 TIT20IUD 引脚时, IT20UDC 向下计数。

如果在 TIT20IUD 引脚的有效沿输入时采样的 TIT20CUD 引脚电平为高, 当有效沿输入到 TIT20IUD 引脚时, IT20UDC 向上计数。

图 7-15. 模式 3 (当上升沿被指定为 TIT20IUD 引脚的有效沿时)

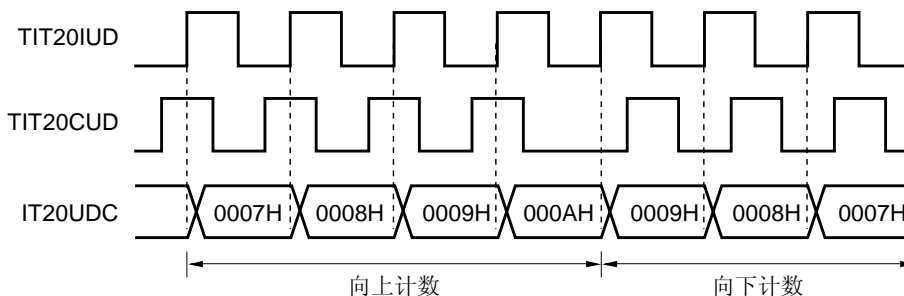
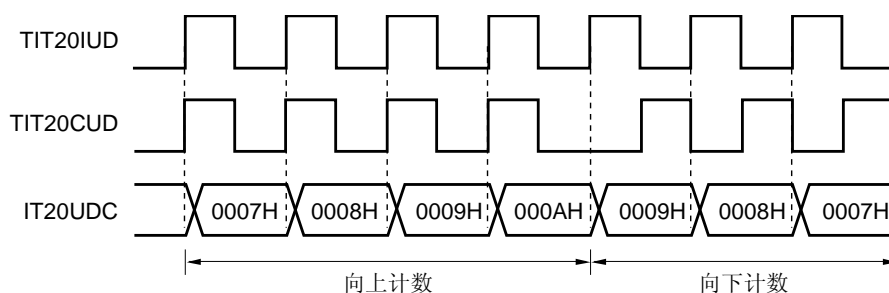


图 7-16. 模式 3 (当上升沿被指定为 TIT20IUD 引脚的有效沿时):
TIT20IUD, TIT20CUD 引脚边沿同时发生时

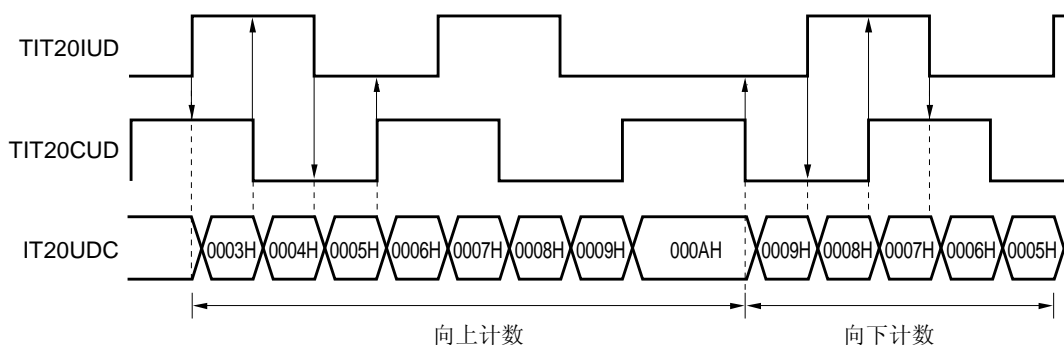


(d) 模式 4 (IT20PRM2 = 1, IT20PRM1 = 1, IT20PRM0 = 1)

在模式 4 下, 当相位相差 90 度的两个信号被输入到 TIT20IUD 和 TIT20CUD 引脚时, 向上 / 向下操作会自动判别并且计数会根据图 7-17 中显示的时序来执行。

在模式 4 下, 计数在输入到 TIT20IUD 和 TIT20CUD 引脚的两个信号的双边沿执行。因此, IT20UDC 每个输入信号的周期内计算四次 (×4 计数)。

图 7-17. 模式 4



- 注意事项
1. 当模式 4 指定为 IT20UDC 的操作模式时, TIT20IUD 和 TIT20CUD 引脚的有效沿指定无效。
 2. 在模式 4 下, 如果 TIT20IUD 引脚边沿和 TIT20CUD 引脚边沿同时输入, IT20UDC 继续执行输入前执行的相同计数操作 (向上或向下)。

(3) UDC 模式 A 下的操作

(a) 间隔操作

按照 IT20UDC 的计数值和 IT20CM0 的设置值匹配后的计数时钟的操作如下所示。

- 在向上计数操作的情况下： IT20UDC 被清除 (0000H) 并且 INTCM10 中断被产生。
- 在向下计数操作的情况下： IT20UDC 计数值递减 (-1) 并且 INTCM10 中断被产生。

备注 间隔操作可以与传输操作组合。

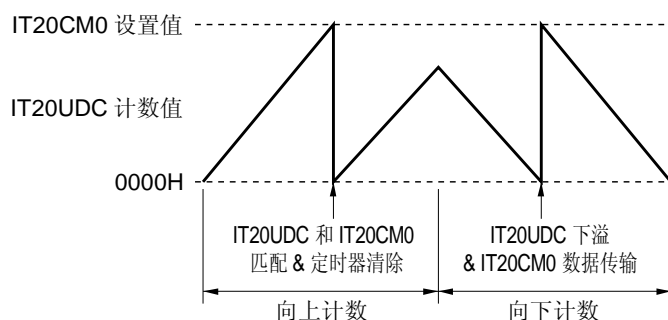
(b) 传输操作

在 IT20UDC 向下计数操作期间，IT20UDC 的计数值变为 0000H 后的下一个计数时钟时的操作如下所示。

- 在向下计数操作的情况下： 保持在 IT20CM0 中的数据会被传输。
- 在向上计数操作的情况下： IT20UDC 的计数值递增 (+1)。

备注 1. 传输允许 / 禁止可以使用 IT20TMC 寄存器的 IT20RLEN 位来设置。
2. 传输操作可以与间隔操作组合。

图 7-18. 结果操作和传输操作组合时 IT20UDC 操作示例



(c) 比较功能

IT20UDC 连接两个比较寄存器 (IT20CM0, IT20CM1) 通道和两个捕获 / 比较寄存器 (IT20CC0, IT20CC1) 通道。

当 IT20UDC 的计数值和一个比较寄存器的设置值匹配时，一个匹配中断 (INTCM10, INTCM11, INTCC10^注, INTCC11^注) 就会输出。

注 当 IT20CC0 和 IT20CC1 被设置为比较寄存器模式时，该匹配中断就会产生。

(d) 捕获功能

IT20UDC 连接两个捕获 / 比较寄存器 (IT20CC0, IT20CC1) 通道。

当 IT20CC0 和 IT20CC1 被设置为捕获寄存器模式时，IT20UDC 的值与对应的捕获触发信号同步捕获。

当 IT20CC0 和 IT20CC1 被设置为捕获寄存器模式时，一个捕获中断 (INTCC10, INTCC11) 会在有效沿检测时产生。

(4) UDC 模式 B 下的操作

(a) 基本操作

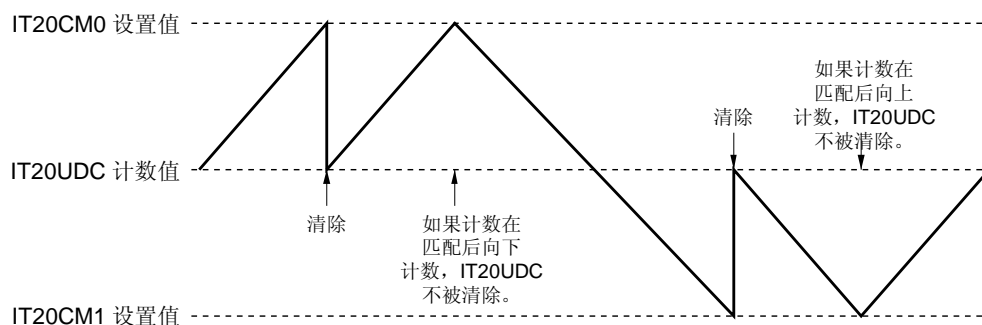
当 IT20UDC 处于 UDC 模式 B 下时，按照 IT20UDC 的计数值和 IT20CM0 的设置值匹配后下一个计数时钟操作如下所示。

- 在向上计数操作的情况下： IT20UDC 被清除 (0000H) 并且 INTCM10 中断被产生。
- 在向下计数操作的情况下： IT20UDC 计数值递减 (-1)。

当 IT20UDC 处于 UDC 模式 B 下时，按照 IT20UDC 的计数值和 IT20CM1 的设置值匹配后下一个计数时钟操作如下所示。

- 在向上计数操作的情况下： IT20UDC 的计数值递增 (+1)。
- 在向下计数操作的情况下： IT20UDC 被清除 (0000H) 并且 INTCM11 中断被产生。

图 7-19. UDC 模式下 IT20UDC 操作示例



(b) 比较功能

IT20UDC 连接两个比较寄存器 (IT20CM0, IT20CM1) 通道和两个捕获 / 比较寄存器 (IT20CC0, IT20CC1) 通道。

当 IT20UDC 的计数值和一个比较寄存器的设置值匹配时，一个匹配中断 (INTCM10 (只在向上计数操作期间), INTCM11 (只在向下计数操作期间), INTCC10^注, INTCC11^注) 就会输出。

注 当 IT20CC0 和 IT20CC1 被设置为比较寄存器模式时，该匹配中断就会产生。

(c) 捕获功能

IT20UDC 连接两个捕获 / 比较寄存器 (IT20CC0, IT20CC1) 通道。

当 IT20CC0 和 IT20CC1 被设置为捕获寄存器模式时，IT20UDC 的值与对应的捕获触发信号同步捕获。

当 IT20CC0 和 IT20CC1 被设置为捕获寄存器模式时，一个捕获中断 (INTCC10, INTCC11) 就会在有效沿检测时产生。

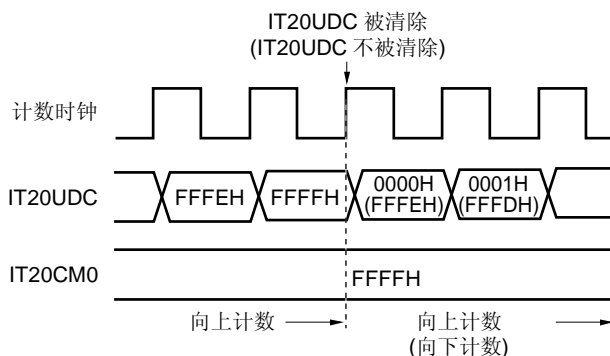
7.5 16 位向上 / 向下计数器 ITENC20 的内部操作

7.5.1 UDC 模式 B 下计数值的清除

当 IT20UDC 处于 UDC 模式 B 下时，计数值清除操作如下所示。

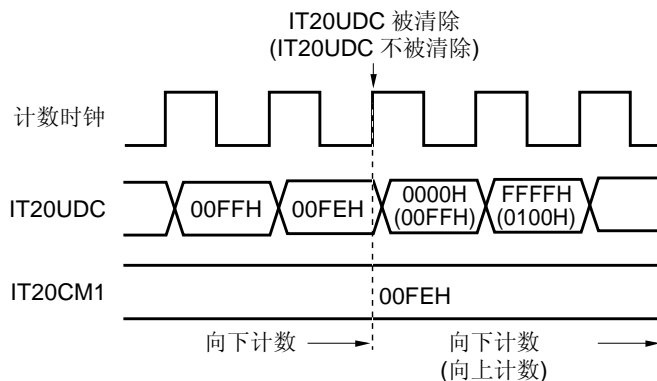
- 在 IT20UDC 向上计数操作的情况下： IT20UDC 在与 IT20CM0 匹配时被清除
- 在 IT20UDC 向下计数操作的情况下： IT20UDC 在与 IT20CM1 匹配时被清除

图 7-20. IT20UDC 向上计数操作期间与 IT20CM0 匹配时的清除操作



- 备注
1. 计数时钟的上升沿被设置为有效沿。
 2. 上图中括号中的项目应用于向下计数操作。

图 7-21. IT20UDC 向下计数操作期间与 IT20CM1 匹配时的清除操作

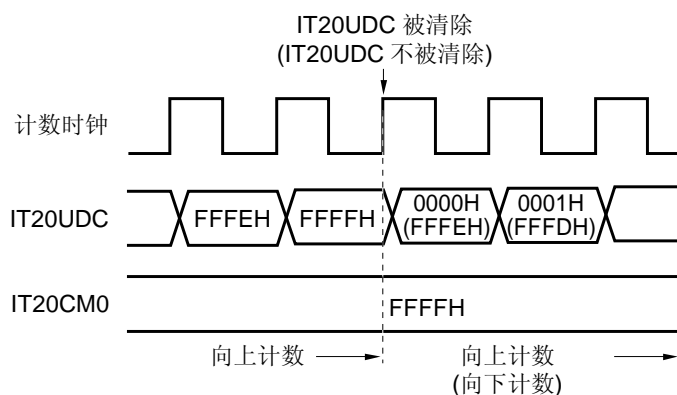


- 备注
1. 计数时钟的上升沿被设置为有效沿。
 2. 上图中括号中的项目应用于向上计数操作。

7.5.2 比较匹配发生时计数值的清除

比较匹配发生时，IT20UDC 清除操作期间的内部操作如下所示。

图 7-22. 比较匹配时的计数值清除操作



注意事项 IT20UDC 的计数值和 IT20CM0 的设置值匹配后下一个计数时钟时的操作如下所示。

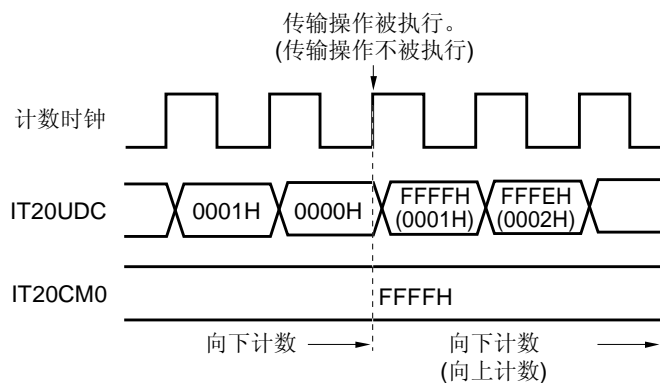
- 在向上计数的情况下：清除操作被执行。
- 在向下计数的情况下：清除操作不被执行。

- 备注**
1. 计数时钟的上升沿被时钟为有效沿。
 2. 上图中括号中的项目应用于向下计数操作。

7.5.3 传输操作

IT20UDC 传输操作期间的内部操作如下所示。

图 7-23. 传输操作期间的内部操作



注意事项 IT20UDC 的计数值变为 0000H 后的计数操作如下所示。

- 在向下计数的情况下：传输操作被执行。
- 在向上计数的情况下：传输操作不被执行。

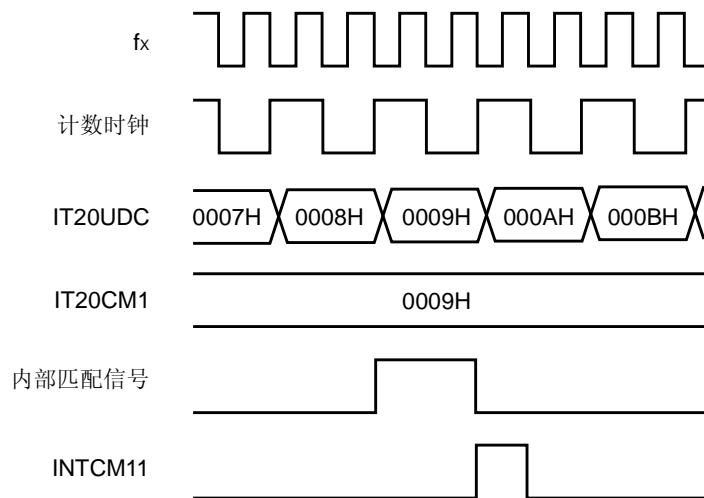
- 备注**
1. 计数时钟的上升沿被时钟为有效沿。
 2. 上图中括号中的项目应用于向上计数操作。

7.5.4 比较匹配时的中断信号输出

当 IT20UDC 的计数值与 IT20CM0、IT20CM1、IT20CC0[※] 或 IT20CC1[※] 寄存器的设置值匹配时，一个中断信号就会输出。中断产生时序如下所示。

注 当 IT20CC0 和 IT20CC1 被设置为比较寄存器模式时。

图 7-24. 比较匹配时的中断输出
(IT20CM1 的操作模式被设置为通用定时器模式并且计数时钟被设置为 $fx/2$)



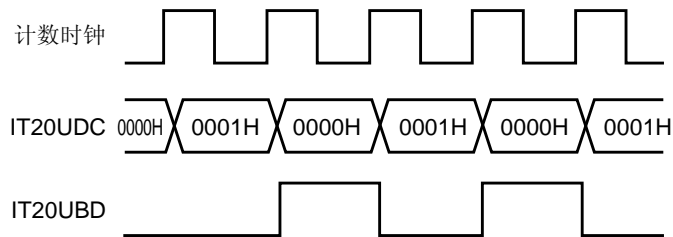
备注 fx: 内部系统时钟

中断信号，例如图 7-24 中所示，在 IT20UDC 的计数值和对应的比较寄存器的设置值匹配后的下一个计数时就会输出。

7.5.5 IT20UBD 标志 (IT20STS 寄存器的第 0 位) 操作

在 UDC 模式 (IT20TUM 寄存器的 IT20CMD 位 = 1) 下，在 IT20UDC 向上 / 向下计数操作期间，IT20UBD 标志在每个内部操作时钟时按照下面更改。

图 7-25. IT20UBD 标志操作



8.1 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。

- 间隔定时器
- PPG 输出
- 脉冲宽度测量
- 外部事件计数器
- 方波输出
- 单脉冲输出

(1) 间隔定时器

16 位定时器 / 事件计数器 00 可以在预定时间间隔生成一个中断请求。

(2) 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自由设置。

(3) 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。

(4) 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。

(5) 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。

(6) 单脉冲输出

16 位定时器 / 事件计数器 00 可以输出一个单脉冲，该单脉冲的输出脉冲宽度可以自由设置。

8.2 16 位定时器 / 事件计数器 00 的配置

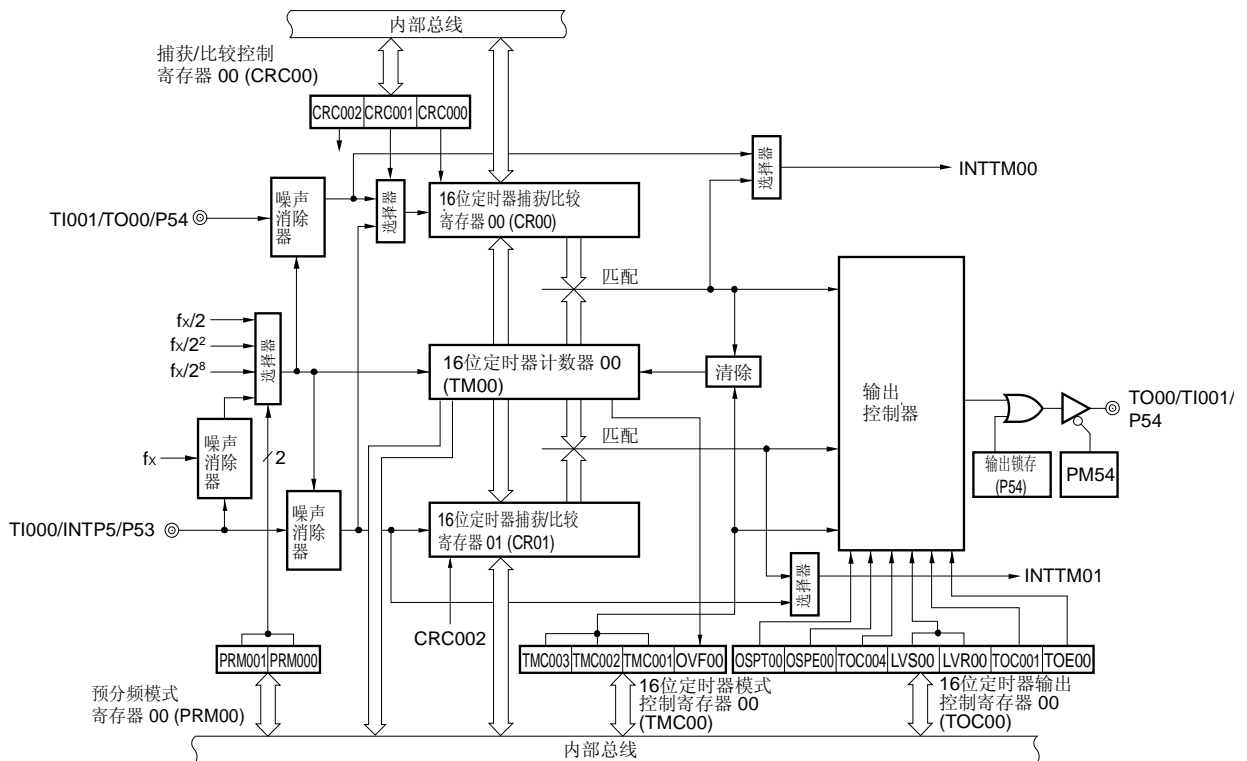
16 位定时器 / 事件计数器 00 由以下硬件组成。

表 8-1. 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 (TM00)
寄存器	16 位定时器捕获 / 比较寄存器。 16 位 (CR00, CR01)
定时器输入	TI000, TI001
定时器输出	TO00, 输出控制器
控制寄存器	16 位定时器模式控制寄存器 00 (TMC00) 16 位定时器捕获 / 比较控制寄存器 00 (CRC00) 16 位定时器输出控制寄存器 00 (TOC00) 预分频模式寄存器 00 (PRM00) 端口模式寄存器 5 (PM5) 端口寄存器 5 (P5)

图 8-1 显示了框图。

表 8-1. 16 位定时器 / 事件计数器 00 的框图

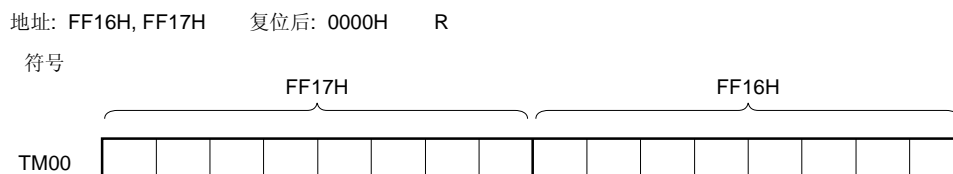


(1) 16 位定时器计数器 00 (TM00)

TM00 是一个 16 位只读寄存器，用于对计数脉冲进行计数。

计数器与输入时钟的上升沿同步累加。

图 8-2. 16 位定时计数器 00 (TM00) 的格式



在以下情况中计数值会被复位为 0000H。

- <1> 在 $\overline{\text{RESET}}$ 输入时
- <2> 如果 TMC003 和 TMC002 被清除
- <3> 如果在有效沿被输入到 TI000 引脚时，TI000 引脚的有效沿在发生清除&开始的模式中被输入
- <4> 如果 TM00 和 CR00 在发生清除&开始的模式中匹配时
- <5> OSPT00 被设置为单脉冲输出模式

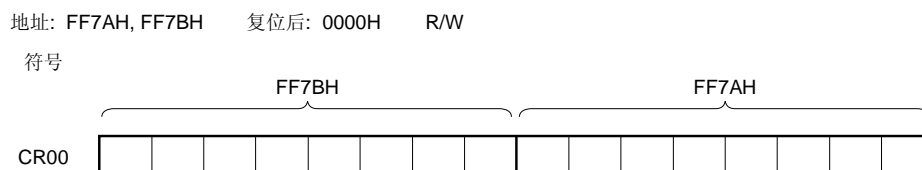
(2) 16 位定时器捕获 / 比较寄存器 00 (CR00)

CR00 是一个具有捕获寄存器和比较寄存器功能的 16 位寄存器。它是用作捕获寄存器还是比较寄存器由捕获 / 比较控制寄存器 00 (CRC00) 的第 0 位 (CRC000) 来设置。

CR00 可以由 16 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除该寄存器为 0000H。

图 8-3. 16 位定时器捕获 / 比较寄存器 00 (CR00) 的格式



- **当 CR00 被用作比较寄存器时**

在 CR00 中设定的值持续地与 16 位定时器计数器 00 (TM00) 的计数值进行比较，如果两者匹配，则将生成一个中断请求信号 (INTTM00)。值会一直被保存直到 CR00 被重写。

- **当 CR00 被用作捕获寄存器时**

可以将 TI000 引脚或 TI001 引脚的有效沿选作捕获触发。使用预分频模式寄存器 00 (PRM00) 来设置 TI000 或 TI001 引脚的有效沿 (参见表 8-2)。

表 8-2. CR00 捕获触发和 TI000 和 TI001 引脚的有效沿

(1) TI000 引脚的有效沿被选作捕获触发 (CRC001 = 1, CRC000 = 1)

CR00 捕获触发	TI000 引脚有效沿		
	ES001	ES000	
下降沿	上升沿	0	1
上升沿	下降沿	0	0
无捕获操作	双边沿	1	1

(2) TI001 引脚的有效沿被选作捕获触发 (CRC001 = 0, CRC000 = 1)

CR00 捕获触发	TI001 引脚有效沿		
	ES101	ES100	
下降沿	下降沿	0	0
上升沿	上升沿	0	1
双边沿	双边沿	1	1

- 备注**
- 禁止设置 ES001, ES000 = 1, 0 和 ES101, ES100 = 1, 0。
 - ES001, ES000: 预分频模式寄存器 00 (PRM00) 的第 5 和 4 位
ES101, ES100: 预分频模式寄存器 00 (PRM00) 的第 7 和 6 位
CRC001, CRC000: 捕获 / 比较控制寄存器 00 (CRC00) 的第 1 和 0 位

- 注意事项**
- 在 TM00 和 CR00 匹配时发生清除&开始的模式中, 在 CR00 中设置 0000H 以外的值。
 - 在自由运行和使用 TI000 引脚有效沿的清除模式中, 如果 CR00 被清除为 0000H, 当紧跟着溢出 (FFFFH) 后 CR00 的值从 0000H 变为 0001H 时, 会产生一个中断请求 (INTTM00)。在 TM00 和 CR00 匹配后, 在 TI000 引脚的有效沿被检测后或者在定时器被单脉冲触发清除后, INTTM00 就会产生。
 - 当 P54 被用作 TI001 引脚的有效沿输入时, 它不能用作定时器输出 (TO00)。此外, 当 P54 被用作 TO00 时, 它不能用作 TI001 引脚的有效沿输入。
 - 当 CR00 被用作捕获寄存器时, 如果寄存器读取时间和捕获触发输入冲突, 读取数据不确定 (捕获数据本身是正确的值)。
如果计数停止输入和捕获触发输入冲突, 捕获的数据不确定。
 - 在 TM00 操作期间不要重写 CR00。

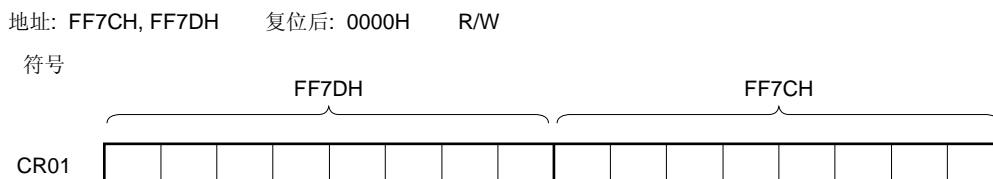
(3) 16 位定时器捕获 / 比较寄存器 01 (CR01)

CR01 是一个具有捕获寄存器和比较寄存器功能的 16 位寄存器。它是用作捕获寄存器还是比较寄存器由捕获 / 比较控制寄存器 00 (CRC00) 的第 2 位 (CRC002) 来设置。

CR01 可以由 16 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除该寄存器为 0000H。

图 8-4. 16 位定时器捕获 / 比较寄存器 01 (CR01) 的格式



- **当 CR01 被用作比较寄存器时**

在 CR01 中设定的值持续地与 16 位定时器计数器 00 (TM00) 的计数值进行比较, 如果两者匹配, 则将生成一个中断请求信号 (INTTM01)。值会一直被保存直到 CR01 被重写。

- **当 CR01 被用作捕获寄存器时**

可以将 TI000 引脚的有效沿选作捕获触发。使用预分频模式寄存器 00 (PRM00) 来设置 TI000 引脚的有效沿 (参见表 8-3)。

表 8-3. CR01 捕获触发和 TI000 引脚的有效沿 (CRC002 = 1)

CR01 捕获触发	TI000 引脚有效沿	
	ES001	ES000
下降沿	0	0
上升沿	0	1
双边沿	1	1

备注

1. 禁止设置 ES001, ES000 = 1, 0。
2. ES001, ES000: 预分频模式寄存器 00 (PRM00) 的第 5 和 4 位
CRC002: 捕获 / 比较控制寄存器 00 (CRC00) 的第 2 位

注意事项 1. 如果 CR01 寄存器被清除为 0000H, 在 TM00 寄存器溢出后, 在 TM00 寄存器和 CR00 寄存器匹配时定时器被清除和启动后, 或者在定时器被 TI000 引脚的有效沿或单脉冲触发清除后, 中断请求 (INTTM01) 就会产生。

2. 当 CR01 被用作捕获寄存器时, 如果寄存器读取时间和捕获触发输入冲突, 读取数据不确定 (捕获数据本身是正确的值)。

如果计数停止输入和捕获触发输入冲突, 捕获的数据不确定。

3. CR01 可以在 TM00 操作期间重写。关于如何重写 CR01 的详细情况, 参见图 8-15 的注意事项 2。

8.3 控制 16 位定时器 / 事件计数器 00 的寄存器

以下六个寄存器用来控制 16 位定时器 / 事件计数器 00。

- 16 位定时器模式控制寄存器 00 (TMC00)
- 捕获 / 比较控制寄存器 00 (CRC00)
- 16 位定时器输出控制寄存器 00 (TOC00)
- 预分频模式寄存器 00 (PRM00)
- 端口模式寄存器 5 (PM5)
- 端口寄存器 5 (P5)

(1) 16 位定时器模式控制寄存器 00 (TMC00)

该寄存器设置 16 位定时器的操作模式，16 位定时器计数器 00 (TM00) 的清除模式和输出时序并检测溢出。

TMC00 可以通过 1 位或 8 位存储器操作指令来进行设置。

$\overline{\text{RESET}}$ 输入清除 TMC00 为 00H。

注意事项 16 位定时器计数器 00 (TM00) 在 TMC002 和 TMC003 分别被设置为 0, 0 以外 (操作停止模式) 的数值时开始操作。将 TMC002 和 TMC003 清除为 0, 0 将会停止操作。

图 8-5. 16 位定时器模式控制寄存器 00 (TMC00) 的格式

地址: FF7EH 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	TMC001	操作模式和清除模式选择	TO00 反转时序选择	中断请求产生
0	0	0	操作停止 (TM00 被清除为 0)	没有变化	不产生
0	0	1			
0	1	0	自由运行模式	TM00 和 CR00 匹配或 TM00 和 CR01 匹配	TM00 和 CR00 匹配或 TM00 和 CR01 匹配时产生
0	1	1		TM00 和 CR00 匹配, TM00 和 CR01 匹配或 TI000 引脚的有效沿	
1	0	0	TI000 引脚有效沿时清除&开始发生	-	
1	0	1			
1	1	0	TM00 和 CR00 匹配时清除&开始发生	TM00 和 CR00 匹配或 TM00 和 CR01 匹配	
1	1	1		TM00 和 CR00 匹配, TM00 和 CR01 匹配或 TI000 引脚的有效沿	

OVF00	16 位定时器计数器 00 (TM00) 溢出检测
0	溢出未检测到
1	溢出检测到

- 注意事项**
1. 在重写 OVF00 标志以外的位之前, 定时器操作必须被停止。
 2. 使用预分频模式寄存器 00 (PRM00) 来设置 TI000/P53 引脚的有效沿。
 3. 如果以下任意模式被选择: TM00 和 CR00 匹配时清除&开始发生的模式, TI000 引脚有效沿时清除&开始发生的模式或者自由运行模式, 那么当 CR00 的值为 FFFFH 并且 TM00 的值由 FFFFH 变为 0000H 时, OVF00 标志被设置为 1。

- 备注**
1. TO00: 16 位定时器 / 事件计数器 00 输出引脚
 2. TI000: 16 位定时器 / 事件计数器 00 输入引脚
 3. TM00: 16 位定时器计数器 00
 4. CR00: 16 位定时器捕获 / 比较寄存器 00
 5. CR01: 16 位定时器捕获 / 比较寄存器 01

(2) 捕获 / 比较控制寄存器 00 (CRC00)

该寄存器控制 16 位定时器捕获 / 比较寄存器 (CR00, CR01) 的操作。

CRC00 可以通过 1 位或 8 位存储器操作指令来进行设置。

$\overline{\text{RESET}}$ 输入清除 CRC00 为 00H。

图 8-6. 捕获 / 比较控制寄存器 00 (CRC00) 的格式

地址: FF6AH 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	CR01 操作模式选择
0	作为比较寄存器进行操作
1	作为捕获寄存器进行操作

CRC001	CR00 捕获触发选择
0	在 TI001 引脚的有效沿上进行捕获
1	通过反相方式在 TI000 引脚的有效沿上进行捕获

CRC000	CR00 操作模式选择
0	作为比较寄存器进行操作
1	作为捕获寄存器进行操作

- 注意事项**
1. 在设置 CRC00 之前, 定时器操作必须停止。
 2. 使用 16 位定时器模式控制寄存器 00 (TMC00), 当选择 TM00 和 CR00 匹配时清除 & 开始发生的模式时, CR00 不应该被指定为捕获寄存器。
 3. 如果上升沿和下降沿都被指定为 TI000 引脚的有效沿, 捕获操作不会执行。
 4. 要确保捕获操作正确执行, 捕获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。

(3) 16 位定时器输出控制寄存器 00 (TOC00)

该寄存器控制 16 位定时器 / 事件计数器 00 输出控制器的操作。它通过软件置位 / 复位定时器输出 F/F, 允许 / 禁止输出反转和 16 位定时器 / 事件计数器 00 定时器输出, 允许 / 禁止单脉冲输出操作并且设置单脉冲输出触发。

TOC00 可以通过 1 位或 8 位存储器操作指令来进行设置。

$\overline{\text{RESET}}$ 输入清除 TOC00 为 00H。

图 8-7. 16 位定时器输出控制寄存器 00 (TOC00) 的格式

地址: FF6BH 复位后: 00H R/W

符号

TOC00

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

OSPT00	通过软件控制的单脉冲输出触发	
0	无单脉冲触发	
1	单脉冲触发	

OSPE00	单脉冲输出操作控制	
0	连续脉冲输出模式	
1	单脉冲输出模式 ^注	

TOC004	使用 CR01 和 TM00 匹配的定时器输出 F/F 控制	
0	禁止反转操作	
1	允许反转操作	

LVS00	LVR00	定时器输出 F/F 状态设置
0	0	没有变化
0	1	定时器输出 F/F 复位 (0)
1	0	定时器输出 F/F 置位 (1)
1	1	禁止设置

TOC001	使用 CR00 和 TM00 匹配的定时器输出 F/F 控制	
0	禁止反转操作	
1	允许反转操作	

TOE00	定时器输出控制	
0	禁止输出 (输出固定为电平 0)	
1	允许输出	

注 单脉冲输出模式只在自由运行模式和 TI000 引脚有效沿时清除&开始发生的模式中正确运行。在 TM00 寄存器和 CR00 寄存器匹配时清除&开始发生的模式中，单脉冲不能输出，因为溢出不会发生。

- 注意事项
1. 在设置 TOC004 以外的位之前，定时器操作必须被停止。
 2. 如果 LVS00 和 LVR00 被读取，0 被读出。
 3. 在数据被设置后，OSPT00 被自动清除，所以 0 被读出。
 4. 在单脉冲输出以外的模式中，不要设置 OSPT00 为 1。
 5. 要连续写入 OSPT00，需要通过预分频模式寄存器 00 (PRM00) 来选择两个或更多计数时钟周期的写入间隔。
 6. 在 TOE00 之前不要设置 LVS00 为 1，并且不要同时设置 LVS00 和 TOE00 为 1。

(4) 预分频模式寄存器 00 (PRM00)

该寄存器用来设置 16 为定时器计数器 00 (TM00) 的计数时钟和 TI000 和 TI001 引脚输入的有效沿。

PRM00 可以通过 1 位或 8 位存储器操作指令来进行设置。

$\overline{\text{RESET}}$ 输入清除 PRM00 为 00H。

图 8-8. 预分频模式寄存器 00 (PRM00) 的格式

地址: FF7FH 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	TI001 引脚有效沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	双边沿

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

PRM001	PRM000	计数时钟选择 ^{注1}
0	0	$f_x/2$ (10 MHz)
0	1	$f_x/2^2$ (5 MHz)
1	0	$f_x/2^8$ (78.125 kHz)
1	1	TI000 引脚有效沿 ^{注2}

- 注**
1. 确保设置计数时钟使其满足以下条件。
 - $V_{DD} = 4.0$ 到 5.5 V: 计数时钟 ≤ 10 MHz
 2. 外部时钟需要一个比内部时钟 (f_x) 长两个周期的脉冲。

- 备注**
1. f_x : X1 输入时钟振荡频率
 2. TI000, TI001: 16 位定时器 / 事件计数器 00 输入引脚
 3. 括号中数字是用于 $f_x = 20$ MHz 的操作。

- 注意事项**
1. 当内部振荡时钟被选作 CPU 的源时钟时，内部振荡器的时钟被分频并提供给计数时钟。如果计数时钟是内部振荡时钟，16 位定时器 / 事件计数器 00 的操作并不保证。当外部时钟被使用并且内部振荡时钟被选作 CPU 的源时钟时，16 位定时器 / 事件计数器 00 的操作也不保证，因为内部振荡时钟作为采样时钟被提供用来消除噪声。
 2. 在停止定时器操作后，总是设置数据到 PRM00。
 3. 如果 TI000 引脚的有效沿将被设置为计数时钟，不要使用 TI000 引脚的有效沿和捕获触发来设置清除 & 开始模式。
 4. 如果在系统刚刚复位后 TI000 或 TI001 引脚为高电平，在上升沿或双边沿被设置为允许 16 位定时器计数器 00 (TM00) 操作的 TI000 引脚或 TI001 引脚的有效沿后，上升沿立即被检测。因此，当上拉 TI000 或 TI001 引脚时需要注意。然而，当在操作被停止一次后重新允许操作时，上升沿不会被检测。
 5. 当 P54 被用作 TI001 引脚的有效沿时，它不能被用作定时器输出 (TO00)，并且当用作 TO00 时，它不能用作 TI001 引脚的有效沿。

(5) 端口模式寄存器 5 (PM5)

该寄存器以 1 位为单位对端口 5 的输入 / 输出进行设置。

为定时器输出使用 P54/TO00/TI001 引脚时，应将 PM54 以及 P54 输出锁存设置为 0。

为定时器输入使用 P54/TO00/TI001 和 P53/TI000/INTP5 引脚时，应将 PM54 和 PM53 设置为 1。此时，P54 和 P53 的输出锁存可能为 0 也可能为 1。

PM5 可以通过 1 位或 8 位存储器操作指令来进行设置。

$\overline{\text{RESET}}$ 输入设置 PM5 为 FFH。

图 8-9. 端口模式寄存器 5 (PM5) 的格式

地址: FF25H 复位后: FFH R/W

符号

	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

PM5n	P5n 引脚 I/O 模式选择 (n = 0 到 7)
0	输出模式 (输出缓存打开)
1	输入模式 (输出缓存关闭)

8.4 16 位定时器 / 事件计数器 00 的操作

8.4.1 间隔定时器操作

按照图 8-10 所示设置 16 位定时器模式控制寄存器 00 (TMC00) 和捕获 / 比较控制寄存器 00 (CRC00) 来使其作为间隔定时器操作。

设置

基本操作设置过程如下所示。

- <1> 设置 CRC00 寄存器 (关于设置值, 参见图 8-10)。
- <2> 设置任意值到 CR00 寄存器。
- <3> 使用 PRM00 寄存器来设置计数时钟。
- <4> 设置 TMC00 寄存器来开始操作 (关于设置值, 参见图 8-10)。

注意事项 CR00 不能在 TM00 操作期间重写。

备注 关于允许 INTTM00 中断的方法, 请参见第十九章 中断功能。

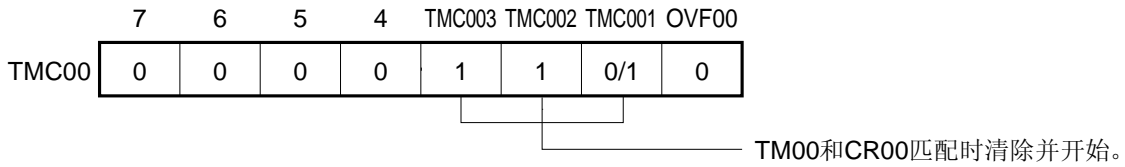
使用 16 位定时器捕获 / 比较寄存器 00 (CR00) 中预置的计数值作为间隔, 中断请求将重复产生。

当 16 位定时器计数器 00 (TM00) 的计数值与 CR00 中的设置值匹配时, TM00 值被清除为 0 的同时计数继续, 并且中断请求信号 (INTTM00) 将会产生。

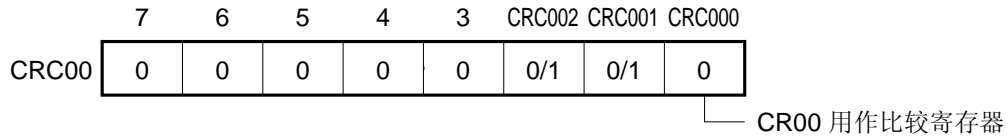
16 位定时器 / 事件计数器 00 的计数时钟可以使用预分频模式寄存器 00 (PRM00) 的第 0 和 1 位 (PRM000, PRM001) 来选择。

图 8-10. 间隔定时器操作的控制寄存器设置

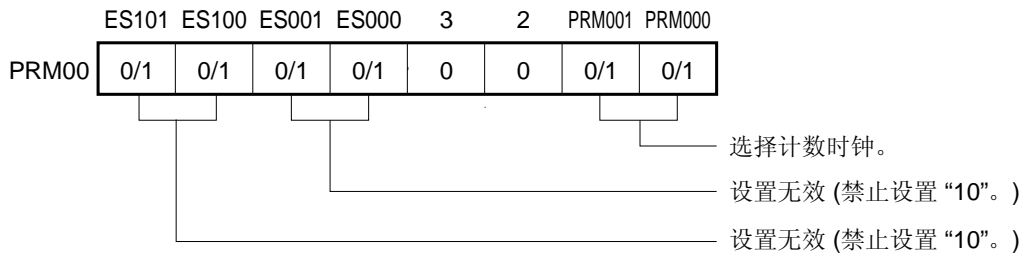
(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕获 / 比较控制寄存器 00 (CRC00)

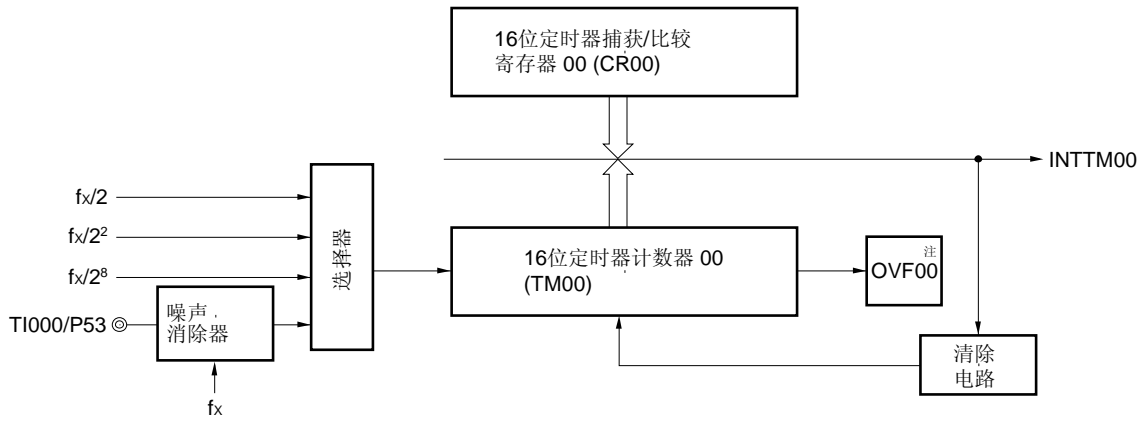


(c) 预分频模式寄存器 00 (PRM00)



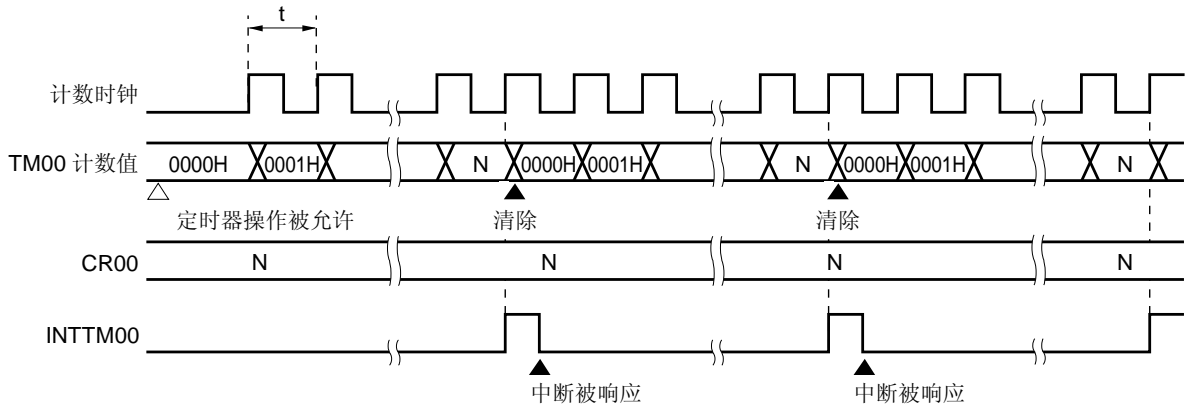
备注 0 / 1: 设置 0 或 1 来使另一个功能与间隔定时器同时使用。
关于详细情况, 参见各自控制寄存器的说明。

图 8-11. 间隔定时器配置图



注 只有当 16 位定时器捕获 / 比较寄存器 00 被设置为 FFFFH 时，OVF00 才被设置为 1。

图 8-12. 间隔定时器操作的时序



备注 间隔时间 = (N + 1) × t
N = 0001H 到 FFFFH

8.4.2 PPG 输出操作

按照图 8-13 所示设置 16 位定时器模式控制寄存器 00 (TMC00) 和捕获 / 比较控制寄存器 00 (CRC00) 来使其作为 PPG (可编程脉冲产生器) 操作。

设置

基本操作设置过程如下所示。

- <1> 设置 CRC00 寄存器 (关于设置值, 参见图 8-13)。
- <2> 设置任意值到 CR00 寄存器作为周期。
- <3> 设置任意值到 CR01 寄存器作为占空比。
- <4> 设置 TOC00 寄存器 (关于设置值, 参见图 8-13)。
- <5> 使用 PRM00 寄存器来设置计数时钟。
- <6> 设置 TMC00 寄存器来开启操作 (关于设置值, 参见图 8-13)。

注意事项 要在操作期间改变占空比 (CR01 寄存器的值) 的值, 参见图 8-15 PPG 输出操作时序中的注意事项 2。

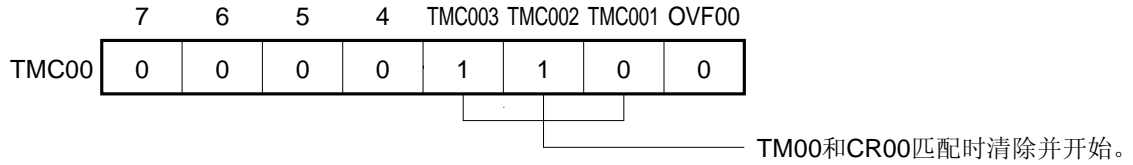
备注

- 1. 关于 TO00 引脚设置的情况, 请参见 8.3 (5) 端口模式寄存器 5 (PM5)。
- 2. 关于允许 INTTM00 中断的方法, 请参见第十九章 中断功能。

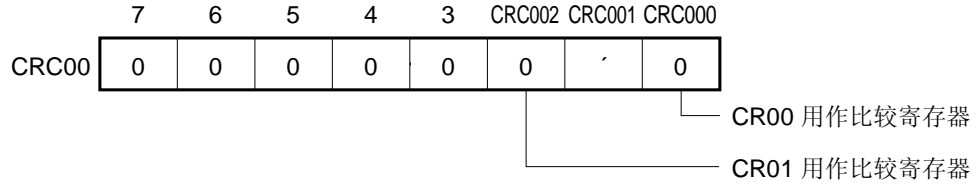
在 PPG 输出模式中, 方波从 TO00 引脚输出, 它的占空比和周期分别对应 16 位定时器捕获 / 比较寄存器 01 (CR01) 和 16 位定时器捕获 / 比较寄存器 00 (CR00) 中预置的计数值。

图 8-13. PPG 输出操作的控制寄存器设置

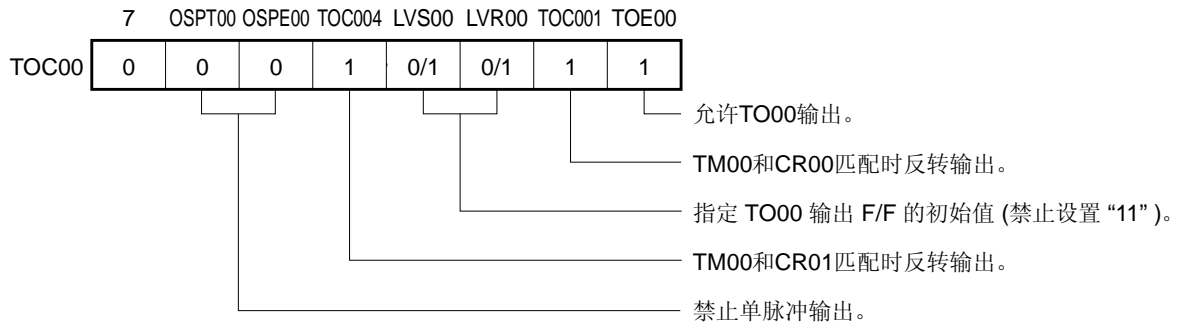
(a) 16 位定时器模式控制寄存器 00 (TMC00)



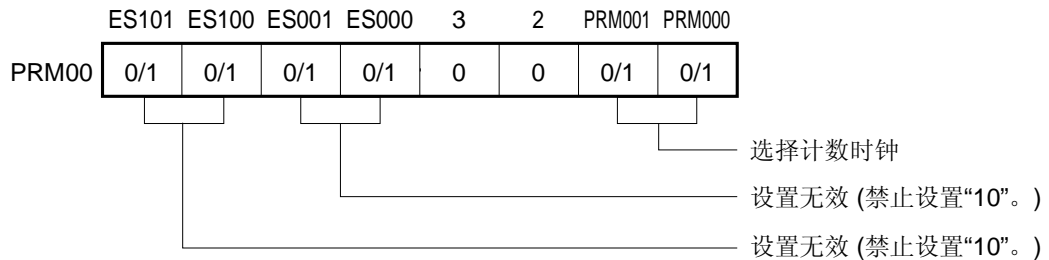
(b) 捕获 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频模式寄存器 00 (PRM00)



注意事项 1. 以下范围内的值才能设置到 CR00 和 CR01 中:

$$0000H \leq CR01 < CR00 \leq FFFFH$$

2. 通过 PPG 输出产生的脉冲的周期 (CR00 设置值 + 1) 具有 (CR01 设置值 + 1) / (CR00 设置值 + 1) 的占空比。

备注 x: 不关注

图 8-14. PPG 输出操作的配置图

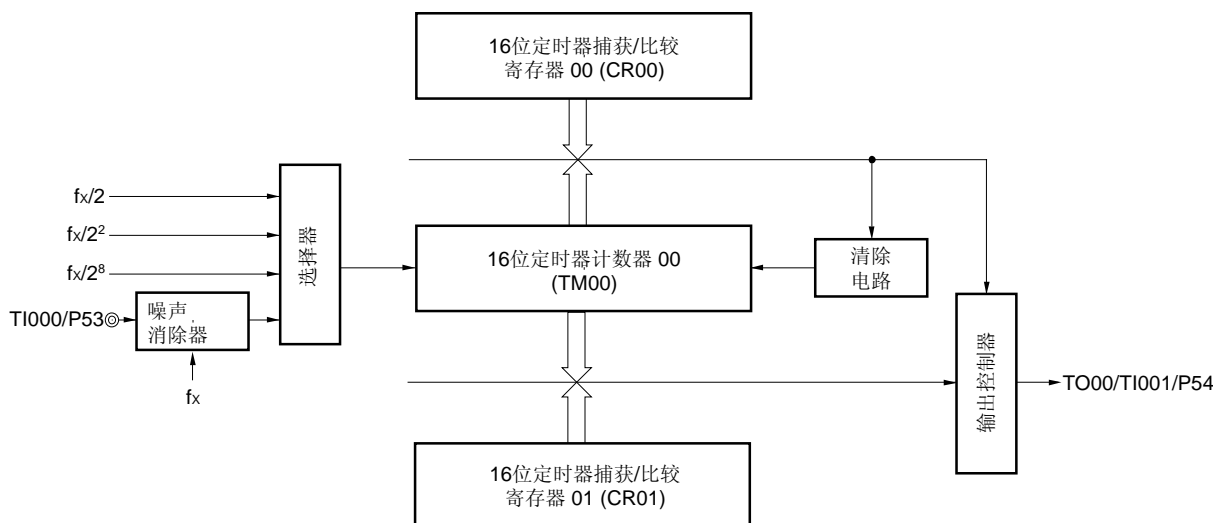
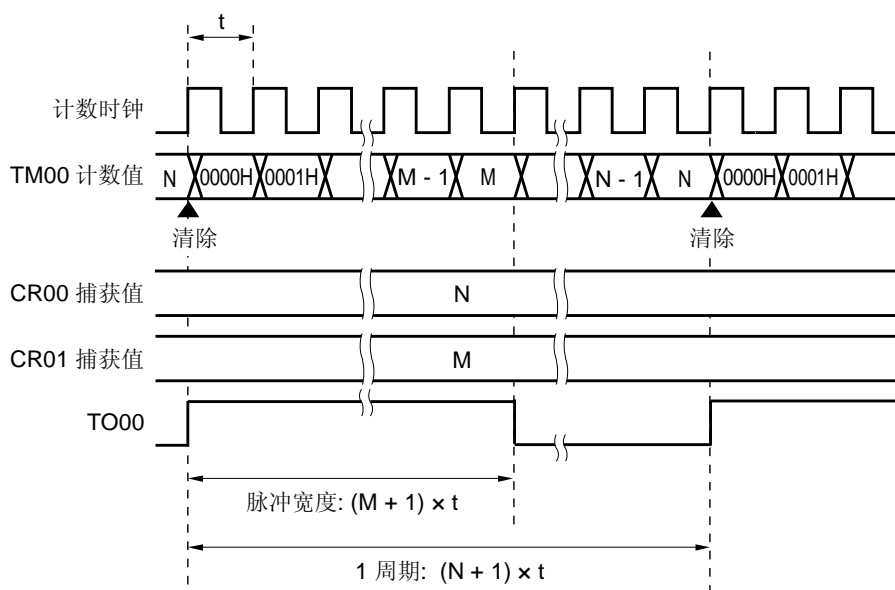


图 8-15. PPG 输出操作时序



- 注意事项
1. CR00 不能在 TM00 操作期间重写。
 2. 在 PPG 输出操作中，使用以下流程来在 TM00 操作期间改变脉冲宽度（重写 CR01）。
 - <1> 禁止通过 TM00 和 CR01 匹配的定时器输出反转操作（TOC004 = 0）
 - <2> 禁止 INTTM01 中断（TMMK01 = 1）
 - <3> 重写 CR01
 - <4> 等待 TM00 计数时钟的一个周期
 - <5> 允许通过 TM00 和 CR01 匹配的定时器输出反转操作（TOC004 = 1）
 - <6> 清除 INTTM01 的中断请求标志（TMIF01 = 0）
 - <7> 允许 INTTM01 中断（TMMK01 = 0）

备注 0000H ≤ M < N ≤ FFFFH

8.4.3 脉冲宽度测量操作

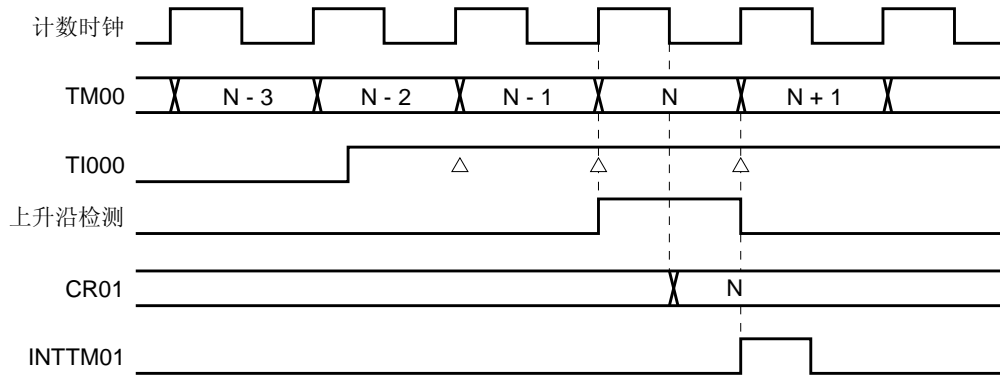
可以使用 16 定时器计数器 00 (TM00) 来测量输入到 TI000 引脚和 TI001 引脚的信号的脉冲宽度。

存在两种测量方法：用使用于自由运行模式下的 TM00 来测量和通过重启与输入到 TI000 引脚的信号的边沿同步的定时器来测量。

当中断发生时，读取捕获寄存器的有效值，检查溢出标志，然后计算需要的脉冲宽度。检查溢出标志后清除。

在信号脉冲宽度以预分频模式寄存器 00 (PRM00) 选择的计数时钟周期采样和 TI000 或 TI001 引脚的有效电平被检测到两次之前，捕获操作不会执行，这样可以消除短脉冲宽度的噪声。

图 8-16. 指定上升沿的 CR01 捕获操作



设置

基本操作设置过程如下所示。

- <1> 设置 CRC00 寄存器（关于设置值，参见图 8-17、8-20、8-22 和 8-24）。
- <2> 使用 PRM00 寄存器来设置计数时钟。
- <3> 设置 TMC00 寄存器来开始操作（关于设置值，参见图 8-17、8-20、8-22 和 8-24）。

注意事项 要使用两个捕获寄存器，设置 TI000 设置 TI001 引脚。

- 备注**
1. 关于 TI000（或 TI001）引脚设置的情况，请参见 8.3（5） 端口模式寄存器 5（PM5）。
 2. 关于允许 INTTM00（或 INTTM01）中断的方法，请参见第十九章 中断功能。

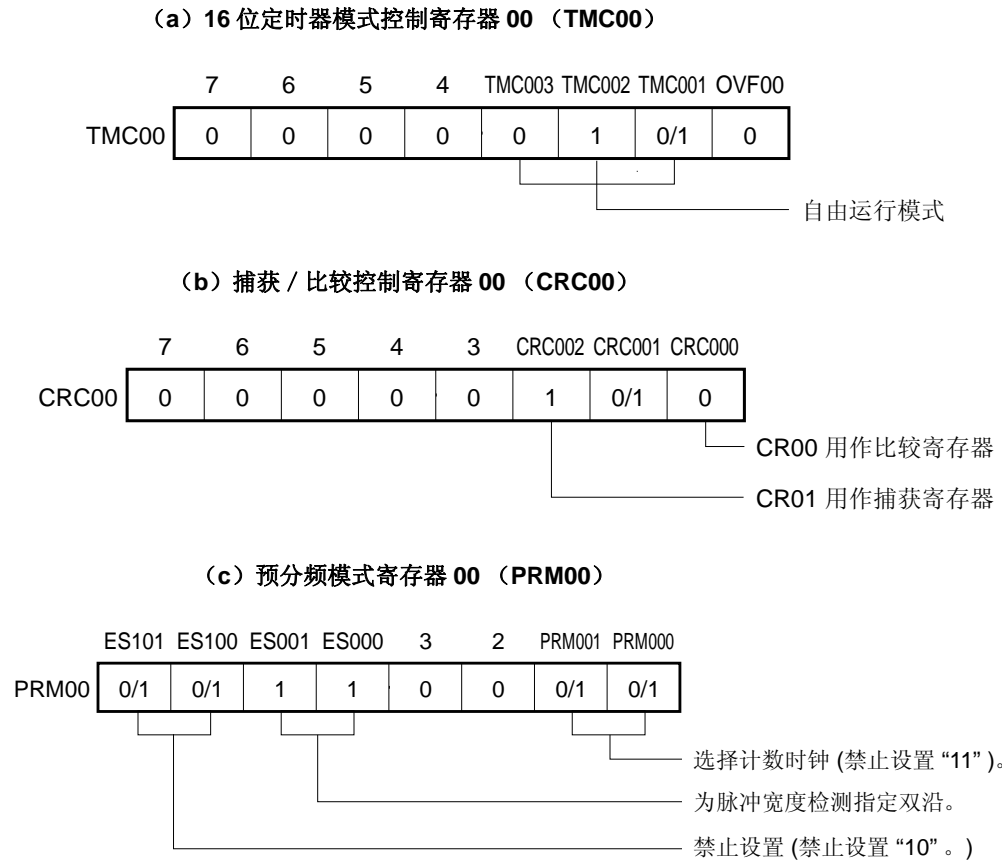
(1) 使用自由运行计数器和一个捕获寄存器的脉冲宽度测量

当 16 位定时器计数器 00 (TM00) 在自由运行模式下操作并且通过预分频模式寄存器 00 (PRM00) 指定的边沿输入到 TI000 引脚时, TM00 的值被放入 16 位定时器捕获 / 比较寄存器 01 (CR01) 并且一个外部中断请求信号 (INTTM01) 将会设置。

通过使用 PRM00 的第 4 和 5 位 (ES000 和 ES001) 来指定 TI000 引脚的上升沿和下降沿。

采样使用 PRM00 选择的计数时钟来执行, 并且捕获操作只有在 TI000 引脚的有效电平被检测到两次时才会执行, 这样可以消除短脉冲宽度的噪声。

图 8-17. 使用自由运行计数器和一个捕获寄存器的脉冲宽度测量控制寄存器设置
(当 TI000 和 CR01 被使用时)



备注 0 / 1: 设置 0 或 1 来使另一个功能与脉冲宽度测量同时使用。关于详细情况, 参见各自控制寄存器的说明。

图 8-18. 使用自由运行计数器的脉冲宽度测量的配置图

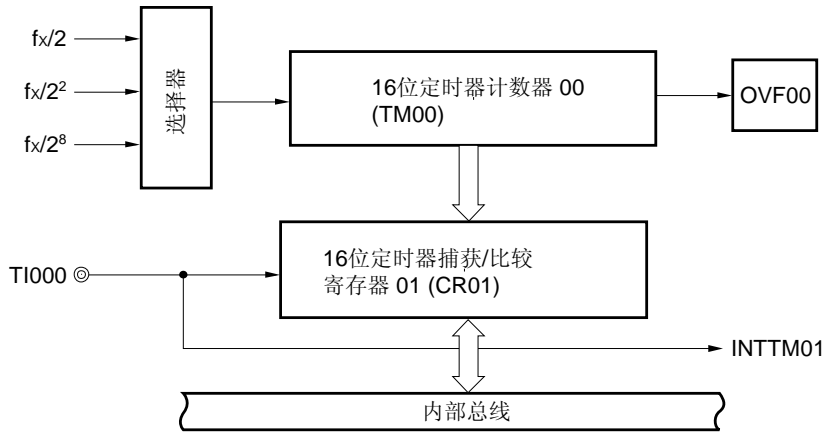
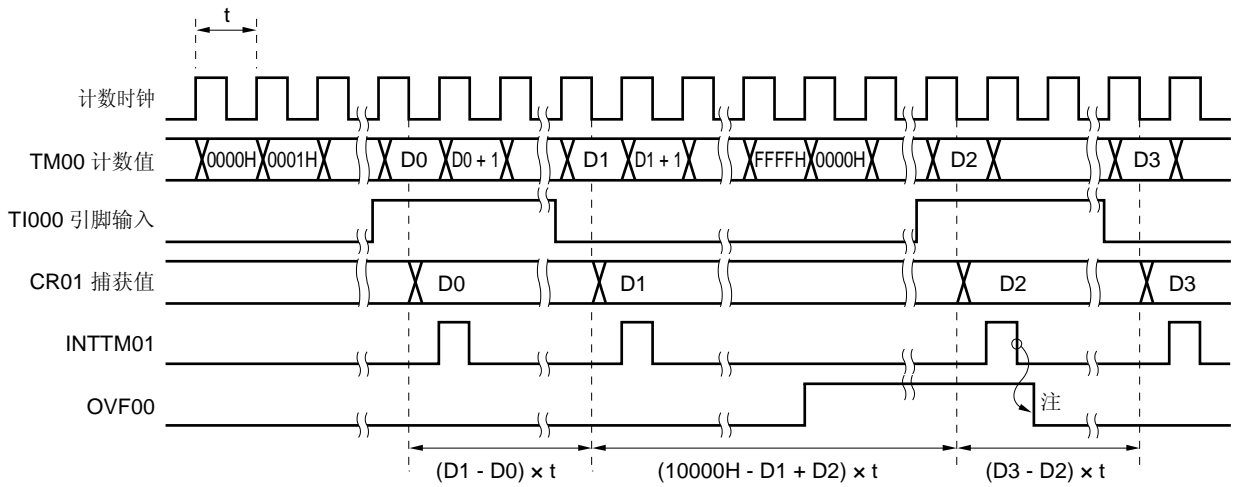


图 8-19. 使用自由运行计数器和一个捕获寄存器的脉冲宽度测量操作的时序（指定双边沿）



注 通过软件清除 OVF00。

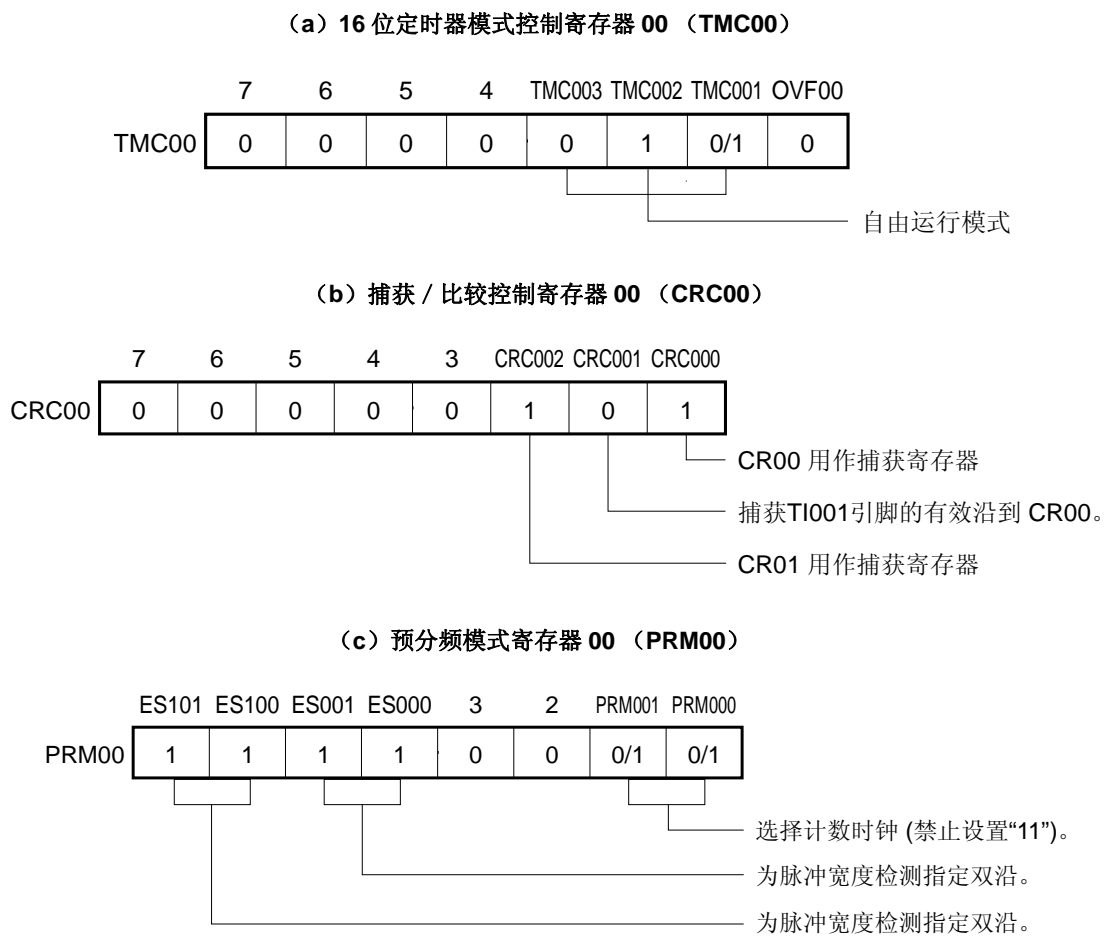
(2) 使用自由运行计数器的两个脉冲宽度的测量

当 16 位定时器计数器 00 (TM00) 在自由运行模式下操作时, 可以同步测量输入到 TI000 引脚和 TI001 引脚的两个信号的脉冲宽度。

当通过预分频模式寄存器 00 (PRM00) 的第 4 和 5 位 (ES000 和 ES001) 指定的边沿输入到 TI000 引脚时, TM00 的值被放入 16 位定时器捕获 / 比较寄存器 01 (CR01) 并且一个中断请求信号 (INTTM01) 将会设置。同时, 当通过预分频模式寄存器 00 (PRM00) 的第 6 和 7 位 (ES100 和 ES101) 指定的边沿输入到 TI001 引脚时, TM00 的值被放入 16 位定时器捕获 / 比较寄存器 00 (CR00) 并且一个中断请求信号 (INTTM00) 被设置。通过使用 PRM00 的第 4 和 5 位 (ES000 和 ES001) 以及第 6 和 7 位 (ES100 和 ES101) 来指定上升沿和下降沿作为 TI000 和 TI001 的边沿。

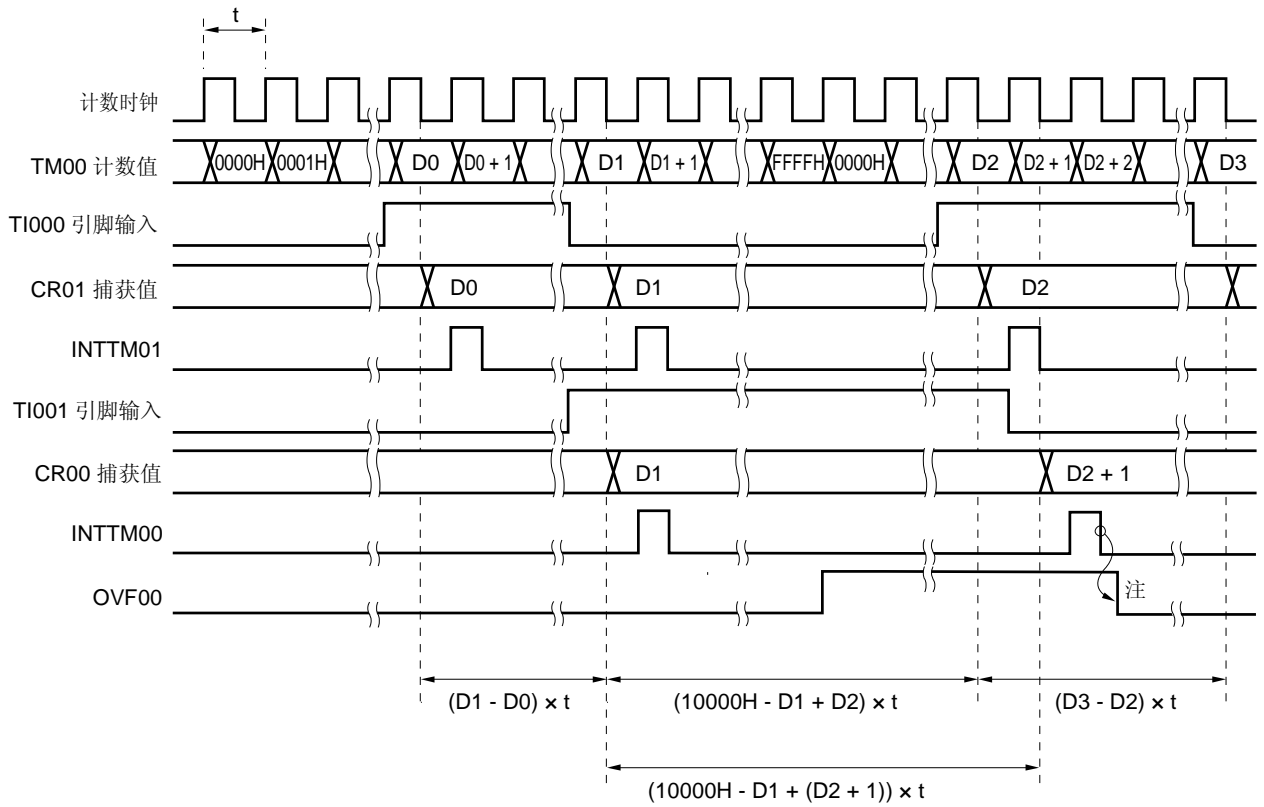
采样使用预分频模式寄存器 00 (PRM00) 选择的计数时钟周期来执行, 并且捕获操作只有在 TI000 或 TI001 引脚的有效电平被检测到两次时才会执行, 这样可以消除短脉冲宽度的噪声。

图 8-20. 使用自由运行计数器的两个脉冲宽度测量的控制寄存器设置



备注 0 / 1: 设置 0 或 1 来使另一个功能与脉冲宽度测量同时使用。关于详细情况, 参见各自控制寄存器的说明。

图 8-21. 使用自由运行计数器的脉冲宽度测量操作的时序（指定双边沿）



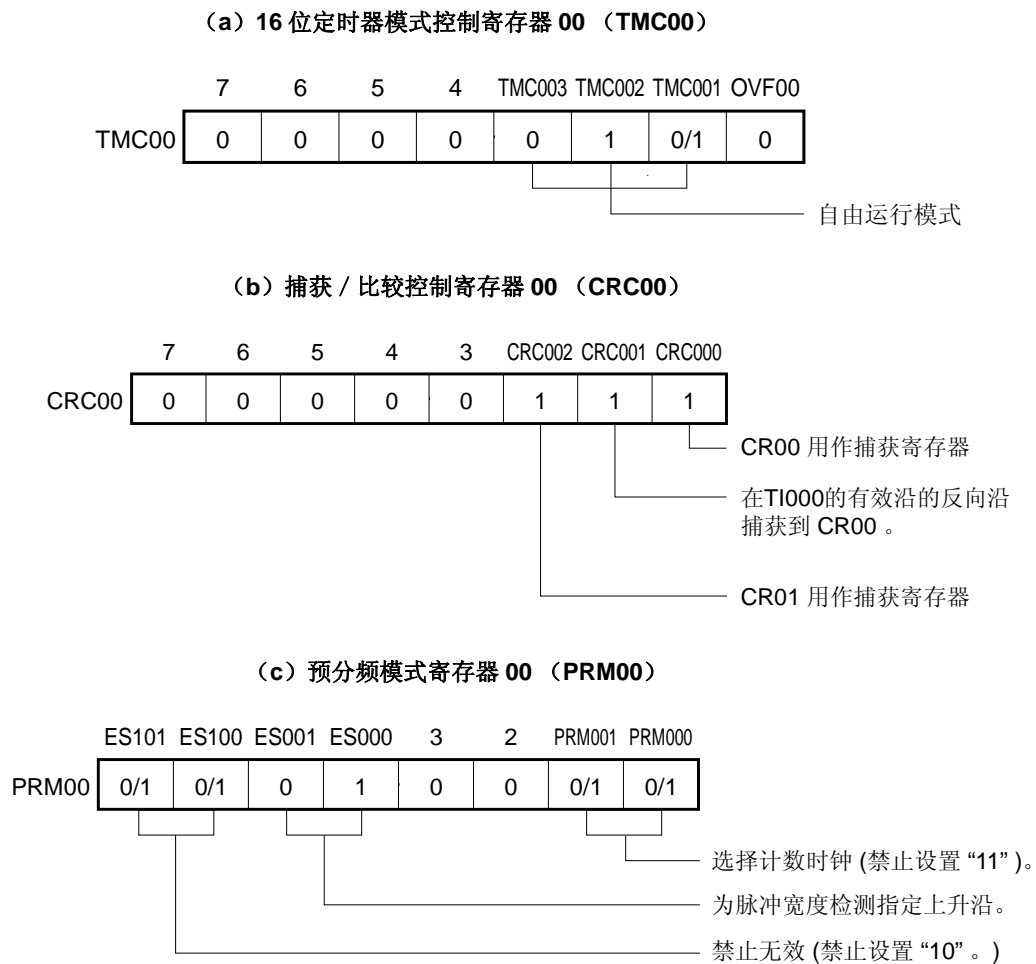
注 通过软件清除 OVF00。

(3) 使用自由运行计数器和两个捕获寄存器的脉冲宽度测量

当 16 位定时器计数器 00 (TM00) 在自由运行模式下操作时, 可以测量输入到 TI000 引脚的信号脉冲宽度。当通过预分频模式寄存器 00 (PRM00) 的第 4 和 5 位 (ES000 和 ES001) 指定的上升沿或下降沿输入到 TI000 引脚时, TM00 的值被放入 16 位定时器捕获 / 比较寄存器 01 (CR01) 并且一个中断请求信号 (INTTM01) 将会设置。

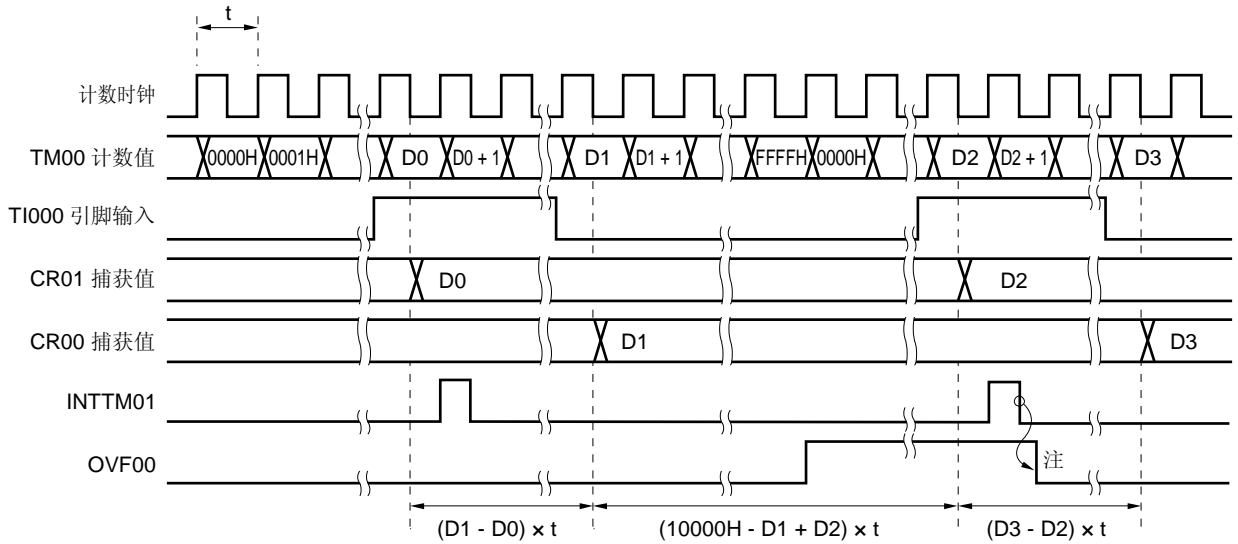
同时, 当捕获操作的反相边沿输入到 CR01 时, TM00 的值被放入 16 位定时器捕获 / 比较寄存器 00 (CR00)。采样使用预分频模式寄存器 00 (PRM00) 选择的计数时钟周期来执行, 并且捕获操作只有在 TI000 引脚的有效电平被检测到两次时才会执行, 这样可以消除短脉冲宽度的噪声。

图 8-22. 使用自由运行计数器和两个捕获寄存器的脉冲宽度测量控制寄存器设置 (指定上升沿)



备注 0 / 1: 设置 0 或 1 来使另一个功能与脉冲宽度测量同时使用。关于详细情况, 参见各自控制寄存器的说明。

图 8-23. 使用自由运行计数器和两个捕获寄存器的脉冲宽度测量操作的时序（指定上升沿）



注 通过软件清除 OVF00。

(4) 通过重启方法的脉冲宽度测量

当 TI000 引脚的有效沿的输入被检测时，16 位定时器计数器 00 (TM00) 的计数值被放入 16 位定时器捕获 / 比较寄存器 01 (CR01)，然后输入到 TI000 引脚的信号的脉冲宽度通过清除 TM00 和重启计数操作来测量。

双边沿中的任意一个——上升或下降——可以使用预分频模式寄存器 00 (PRM00) 的第 4 和 5 位 (ES000 和 ES001) 来选择。

采样使用预分频模式寄存器 00 (PRM00) 选择的计数时钟周期来执行，并且捕获操作只有在 TI000 引脚的有效电平被检测到两次时才会执行，这样可以消除短脉冲宽度的噪声。

图 8-24. 通过重启方法的脉冲宽度测量的控制寄存器设置
(指定上升沿)

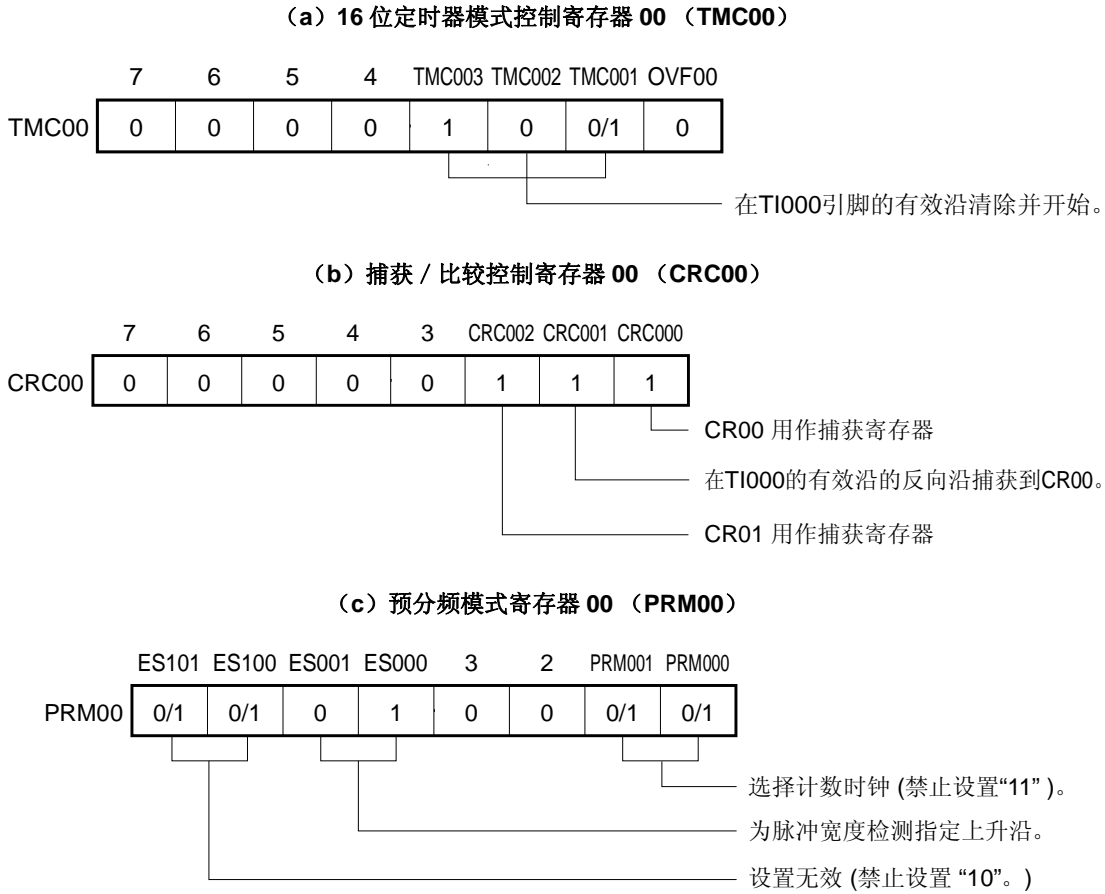
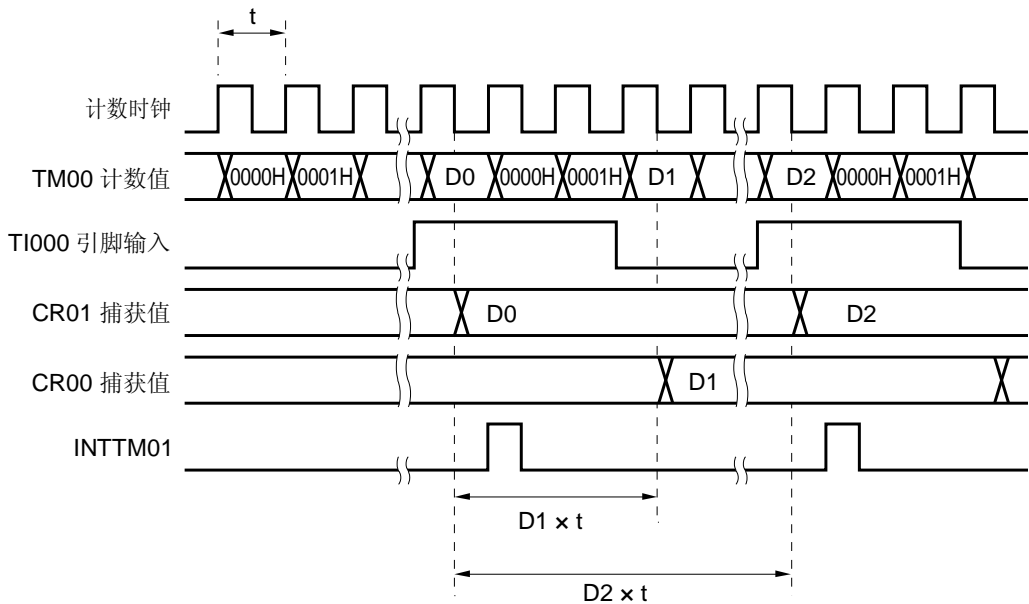


图 8-25. 通过重启方法的脉冲宽度测量操作的时序
(指定上升沿)



8.4.4 外部事件计数器操作

设置

基本操作设置过程如下所示。

- <1> 设置 CRC00 寄存器（关于设置值，参见图 8-26）。
- <2> 使用 PRM00 寄存器来设置计数时钟。
- <3> 设置任意值到 CR00 寄存器（0000H 不能设置）。
- <4> 设置 TMC00 寄存器来开始操作（关于设置值，参见图 8-26）。

- 备注**
- 1. 关于 TI000 引脚设置的详细情况，请参见 8.3 (5) 端口模式寄存器 5 (PM5)。
 - 2. 关于允许 INTTM00 中断的方法，请参见第十九章 中断功能。

外部事件计数器使用 16 位定时器计数器 00 (TM00) 来计数输入到 TI000 引脚的外部时钟脉冲的个数。

每次预分频模式寄存器 00 (PRM00) 指定的有效沿输入时，TM00 增加。

当 TM00 计数值与 16 位定时器捕获 / 比较寄存器 00 (CR00) 的值匹配时，TM00 被清除为 0 并且中断请求信号 (INTTM00) 会产生。

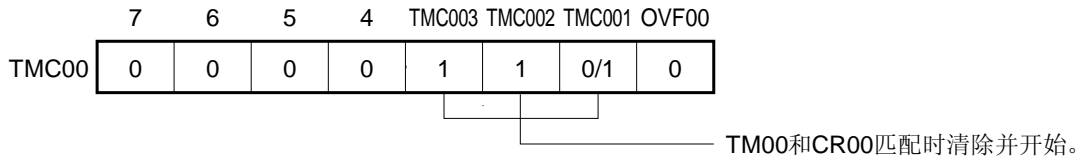
输入 0000H 以外的值到 CR00 (1 位脉冲的计数操作不能执行)。

三种边沿中的任意一种——上升，下降或双边沿——可以使用预分频模式寄存器 00 (PRM00) 的第 4 和 5 位 (ES000 和 ES001) 来选择。

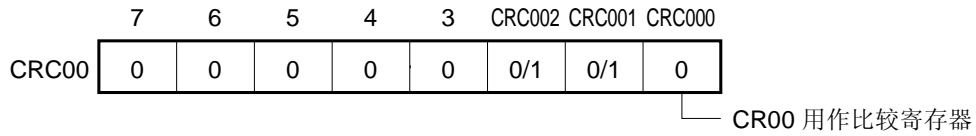
采样使用内部时钟 (fx) 来执行，并且操作只有在 TI000 引脚的有效电平被检测到两次时才会执行，这样可以消除短脉冲宽度的噪声。

图 8-26. 外部事件计数器模式下的控制寄存器设置
(指定上升沿)

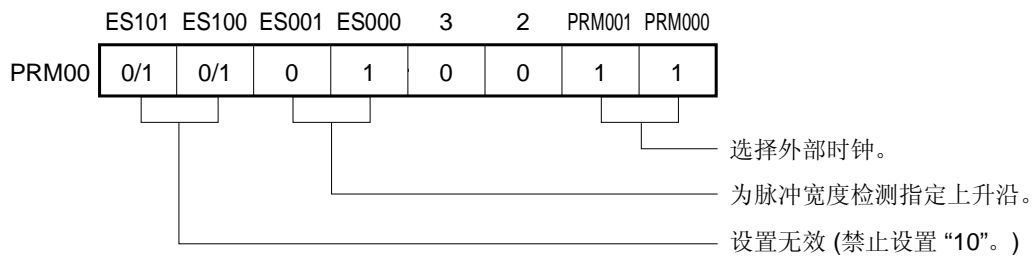
(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕获 / 比较控制寄存器 00 (CRC00)

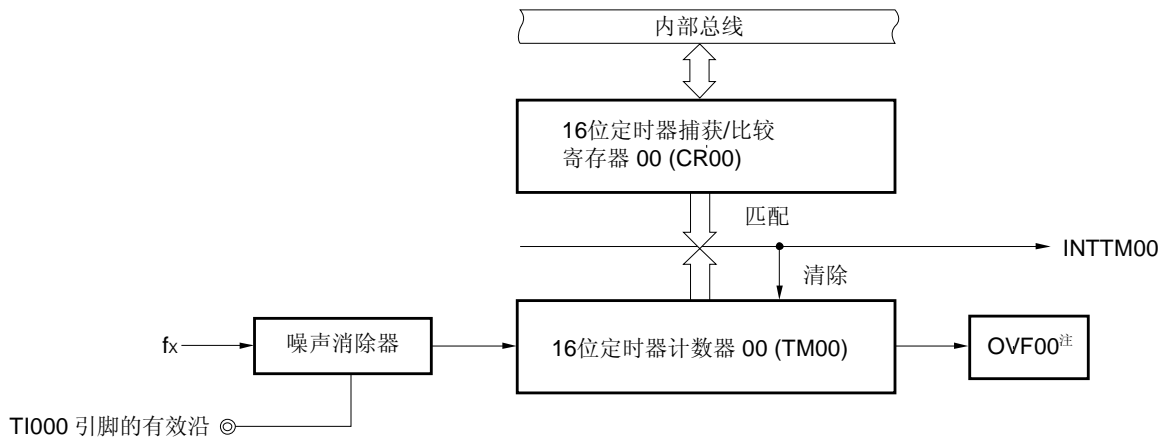


(c) 预分频模式寄存器 00 (PRM00)



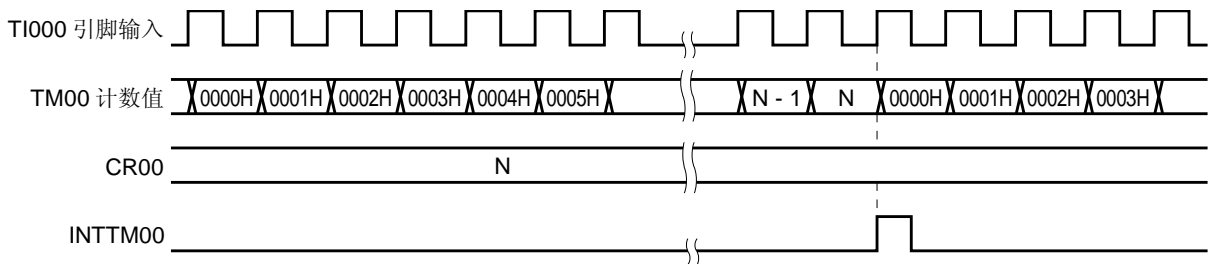
备注 0 / 1: 设置 0 或 1 来使另一个功能与外部事件计数器同时使用。关于详细情况, 参见各自控制寄存器的说明。

图 8-27. 外部事件计数器操作的配置图



注 OVF00 只有当 CR00 被设置为 FFFFH 时才会被设置为 1。

图 8-28. 外部事件计数器操作时序（指定上升沿）



注意事项 当读取外部事件计数器计数值时，TM00 应该会被读取。

8.4.5 方波输出操作

设置

基本操作设置过程如下所示。

- <1> 使用 PRM00 寄存器来设置计数时钟。
- <2> 设置 CRC00 寄存器（关于设置值，参见图 8-29）。
- <3> 设置 TOC00 寄存器（关于设置值，参见图 8-29）。
- <4> 设置任意值到 CR00 寄存器（0000H 不能被设置）。
- <5> 设置 TMC00 寄存器来开始操作（关于设置值，参见图 8-29）。

注意事项 CR00 不能在 TM00 操作期间重写。

- 备注**
1. 关于 TO00 引脚设置的详细情况，请参见 8.3（5） 端口模式寄存器 5（PM5）。
 2. 关于允许 INTTM00 中断的方法，请参见第十九章 中断功能。

任意选定的频率的方波可以以预置到 16 位定时器捕获 / 比较寄存器 00（CR00）的计数值确定的间隔输出。

通过设置 16 位定时器输出控制寄存器 00（TOC00）的第 0 位（TOE00）和第 1 位（TOC001）为 1，TO00 引脚输出状态以预置到 CR00 的计数值 + 1 确定的间隔反转。这样方波就以任意选定的频率输出。

图 8-29. 方波输出模式下的控制寄存器设置（1 / 2）

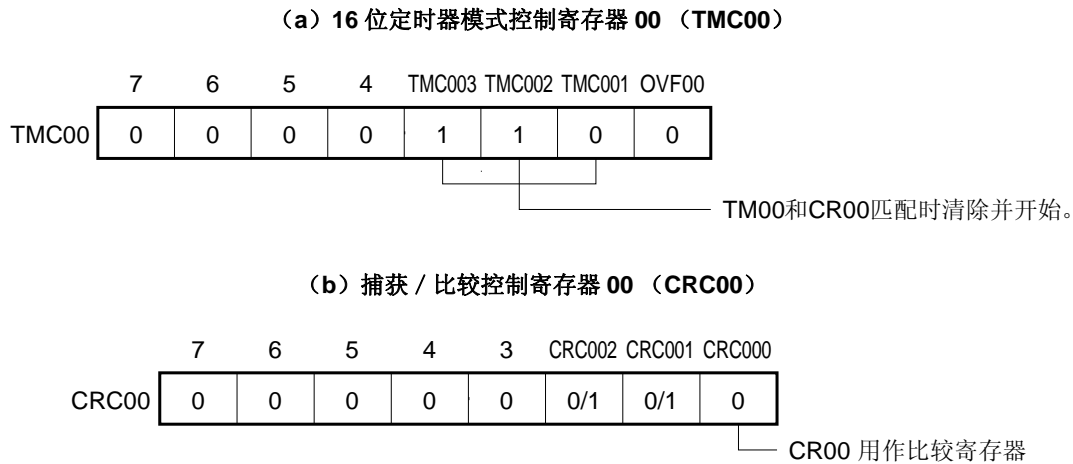
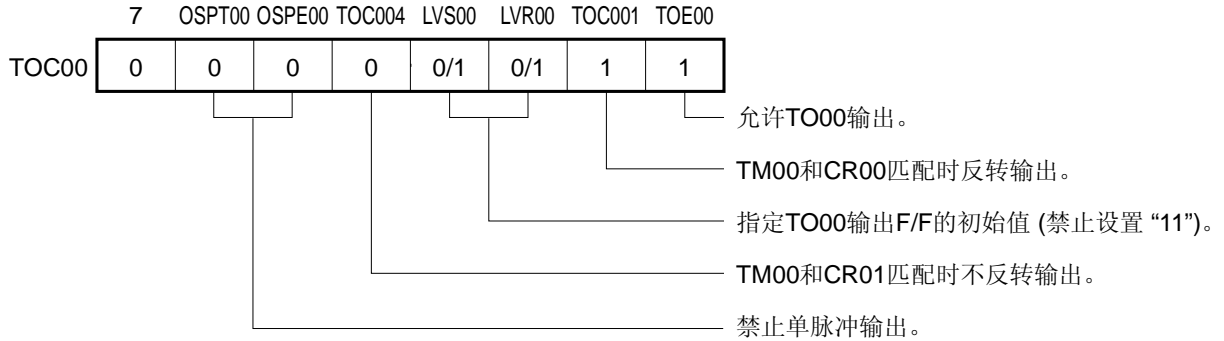
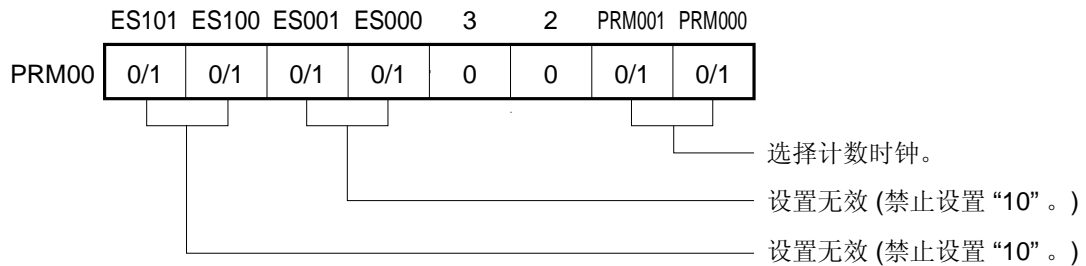


图 8-29. 方波输出模式下的控制寄存器设置 (2 / 2)

(c) 16 位定时器输出控制寄存器 00 (TOC00)

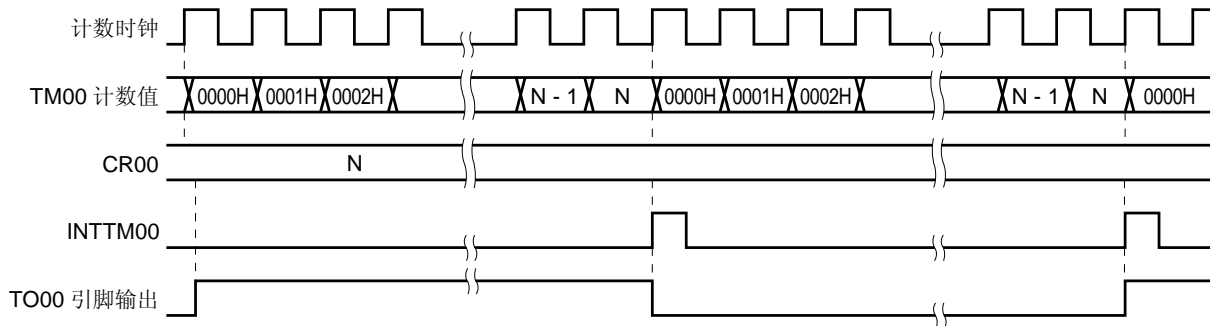


(d) 预分频模式寄存器 00 (PRM00)



备注 0 / 1: 设置 0 或 1 来使另一个功能与方波输出同时使用。关于详细情况, 参见各自控制寄存器的说明。

图 8-30. 方波输出操作时序



8.4.6 单脉冲输出操作

16 位定时器 / 事件计数器 00 可以与软件触发或外部触发 (TI000 引脚输入) 同步输出一个单脉冲。

设置

基本操作设置过程如下所示。

- <1> 使用 PRM00 寄存器来设置计数时钟。
- <2> 设置 CRC00 寄存器 (关于设置值, 参见图 8-31 和 8-33)。
- <3> 设置 TOC00 寄存器 (关于设置值, 参见图 8-31 和 8-33)。
- <4> 设置任意值到 CR00 和 CR01 寄存器 (0000H 不能被设置)。
- <5> 设置 TMC00 寄存器来开始操作 (关于设置值, 参见图 8-31 和 8-33)。

- 备注**
- 1. 关于 TO00 引脚设置的详细情况, 请参见 8.3 (5) 端口模式寄存器 5 (PM5)。
 - 2. 关于允许 INTTM00 (如果需要, INTTM01) 中断的方法, 请参见第十九章 中断功能。

(1) 软件触发的单脉冲输出

通过按照图 8-31 所示设置 16 位定时器模式控制寄存器 00 (TMC00)、捕获 / 比较控制寄存器 00 (CRC00) 和 16 位定时器输出控制寄存器 00 (TOC00), 并且通过软件设置 TOC00 寄存器的第 6 位 (OSPT00) 为 1, 一个单脉冲就可以从 TO00 引脚输出。

通过设置 OSPT00 位为 1, 16 位定时器 / 事件计数器 00 被清除并启动, 并且其输出在达到事先设置到 16 位定时器捕获 / 比较寄存器 01 (CR01) 中的计数值 (N) 时变为有效。然后, 输出在达到事先设置到 16 位定时器捕获 / 比较寄存器 00 (CR00) * 中的计数值 (M) 时变为无效。

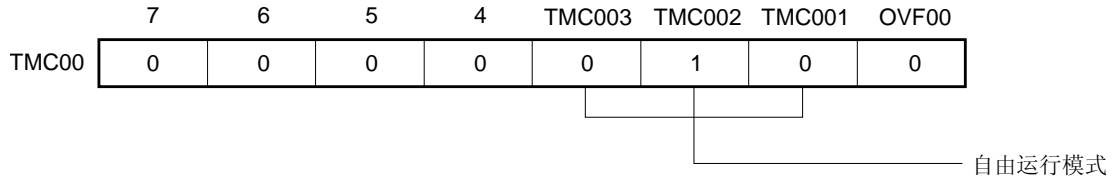
即使在单脉冲被输出后, TM00 寄存器也会继续其操作。要停止 TM00 寄存器, TMC00 寄存器的 TMC003 和 TMC002 位必须清除为 00。

注 这是 $N < M$ 的情况。当 $N > M$ 时, 输出按照 CR00 寄存器变为有效并且按照 CR01 寄存器变为无效。不要设置 N 为 M。

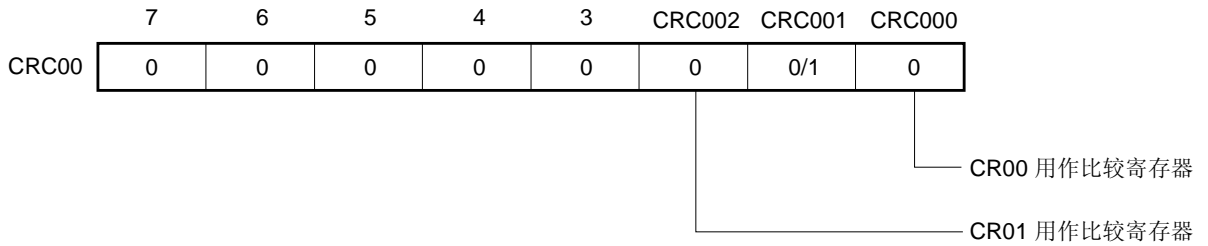
- 注意事项**
- 1. 单脉冲输出期间不要设置 OSPT00 位。要再次输出单脉冲, 则应等待直到当前单脉冲输出完成后。
 - 2. 当使用软件触发的 16 位定时器 / 事件计数器 00 的单脉冲输出时, 不要更改 TI000 引脚或其替换功能端口引脚的电平。
因为外部触发即使在这种情况下也会有效, 在 TI000 引脚或其替换功能端口引脚的电平时定时器被清除并启动, 这样会导致脉冲以不希望的时序输出。

图 8-31. 软件触发单脉冲输出的控制寄存器设置

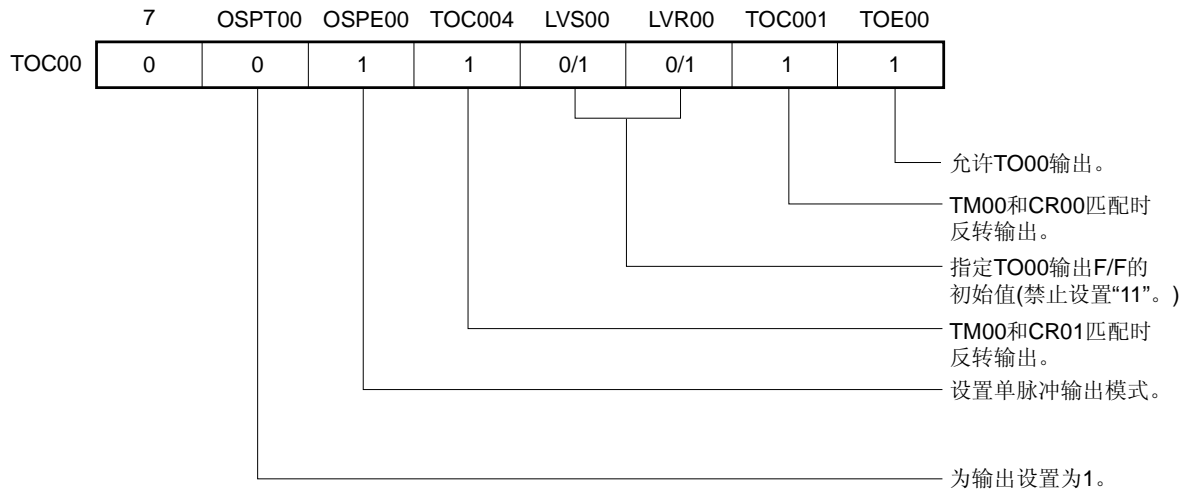
(a) 16 位定时器模式控制寄存器 00 (TMC00)



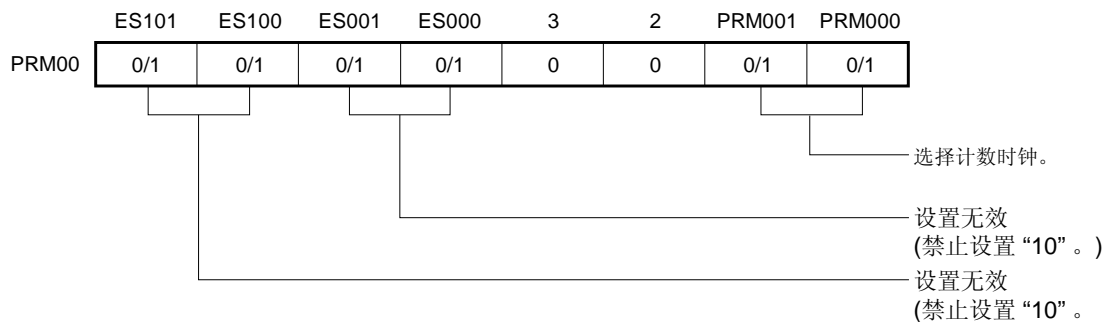
(b) 捕获 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)

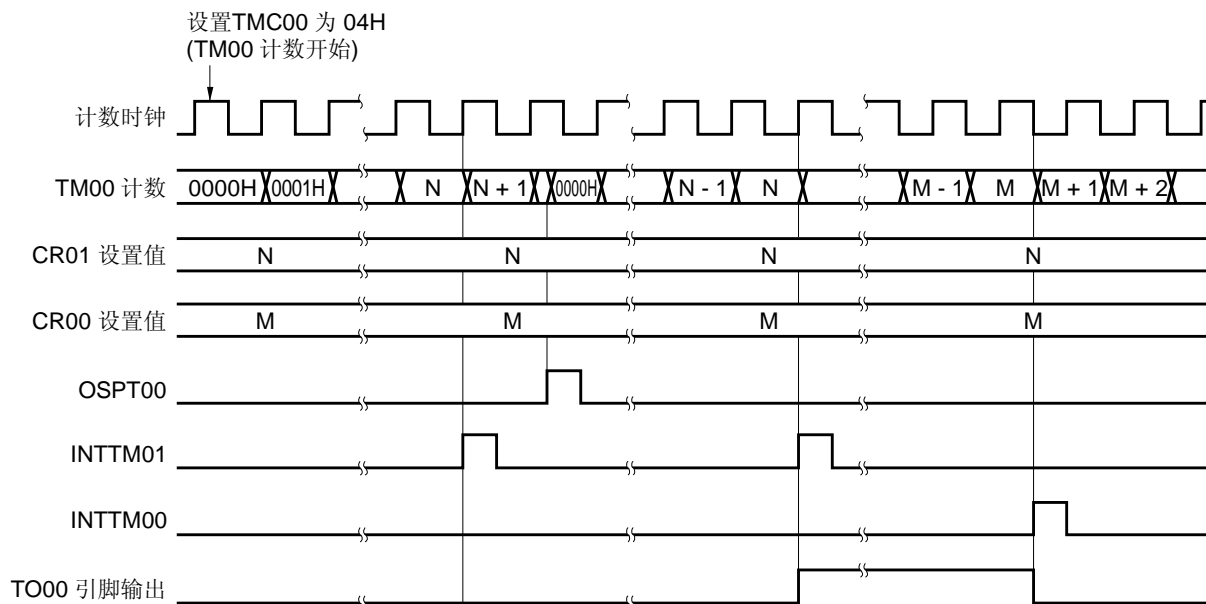


(d) 预分频模式寄存器 00 (PRM00)



注意事项 不要设置 0000H 到 CR00 和 CR01 寄存器。

图 8-32. 软件触发单脉冲输出操作的时序



注意事项 只要 **00**（操作停止模式）以外的值被设置到 **TMC003** 和 **TMC002** 位，16 位定时器计数器 **00** 就开始操作。

备注 $N < M$

(2) 外部触发的单脉冲输出

通过按照图 8-33 所示设置 16 位定时器模式控制寄存器 00（TMC00）、捕获 / 比较控制寄存器 00（CRC00）和 16 位定时器输出控制寄存器 00（TOC00），并且使用 TI000 引脚的有效沿作为外部触发，一个单脉冲可以从 TO00 引脚输出。

TI000 引脚的有效沿通过预分频模式寄存器 00（PRM00）的第 4 位和第 5 位（ES000，ES001）来进行设置。上升沿、下降沿或双边沿都可以指定。

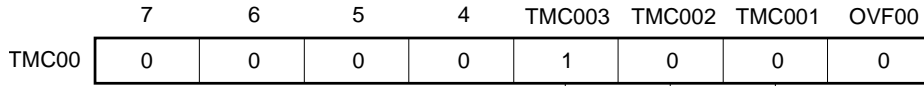
当 TI000 因的有效沿被检测时，16 位定时器 / 事件计数器被清除并启动，并且其输出在达到事先设置到 16 位定时器捕获 / 比较寄存器 01（CR01）中的计数值时变为有效。然后，输出在达到事先设置到 16 位定时器捕获 / 比较寄存器 00（CR00）*中的计数值时变为无效。

注 这是 $N < M$ 的情况。当 $N > M$ 时，输出按照 CR00 寄存器变为有效并且按照 CR01 寄存器变为无效。不要设置 N 为 M。

注意事项 如果在单脉冲输出期间外部触发再次产生，它将被忽略。

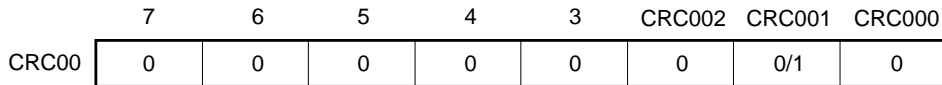
图 8-33. 外部触发单脉冲输出的控制寄存器设置（指定上升沿）

(a) 16 位定时器模式控制寄存器 00 (TMC00)



在 TI000 引脚的有效沿清除并开始。

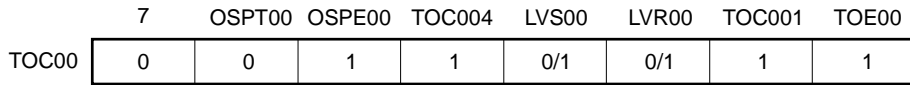
(b) 捕获 / 比较控制寄存器 00 (CRC00)



CR00 用作比较寄存器

CR01 用作比较寄存器

(c) 16 位定时器输出控制寄存器 00 (TOC00)



允许 TO00 输出。

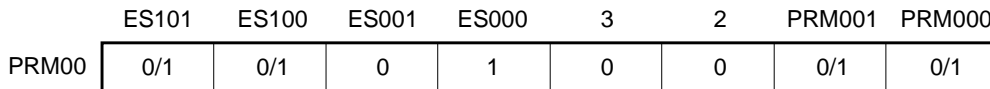
TM00 和 CR00 匹配时反转输出。

指定 TO00 输出 F/F 的初始值。(禁止设置“11”。)

TM00 和 CR01 匹配时反转输出。

设置单脉冲输出模式。

(d) 预分频模式寄存器 00 (PRM00)



选择计数时钟 (禁止设置“11”)。

为脉冲宽度检测指定上升沿。

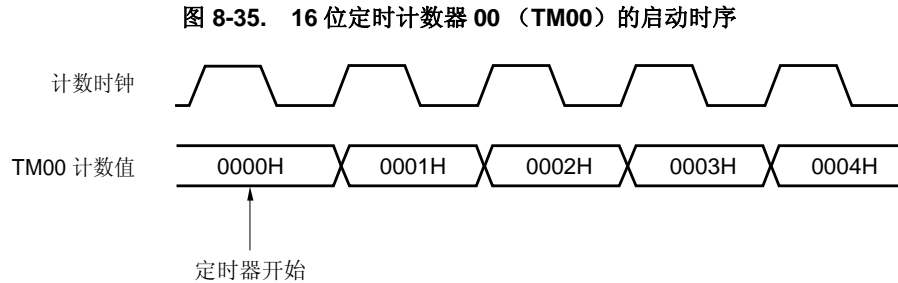
设置无效 (禁止设置“10”。)

注意事项 不要设置 0000H 到 CR00 和 CR01 寄存器。

8.5 16 位定时器 / 事件计数器 00 的注意事项

(1) 定时器启动时的错误

在定时器启动后生成一个匹配信号所需的时间内最多一个时钟的误差可能会产生。这是由于 16 位定时器计数器 00 (TM00) 与计数时钟异步启动。



(2) 16 位定时器捕获 / 比较寄存器 00 设置

在 TM00 和 CR00 匹配时清除&开始发生的模式下，设置 16 位定时器捕获 / 比较寄存器 00 (CR00) 为 0000H 以外的值。这意味着当 16 位定时器 / 事件计数器 00 被用作外部事件计数器时，1 个脉冲的计数操作不能执行。

(3) 捕获寄存器数据保持时序

在 16 位定时器 / 事件计数器 00 停止后，16 位定时器捕获 / 比较寄存器 00 和 01 (CR00 和 CR01) 的值不会保证。

(4) 有效沿设置

在分别清除 16 位定时器模式控制寄存器 (TMC00) 的 2 和 3 位 (TMC002 和 TMC003) 为 0，0 后，设置 TI000 引脚的有效沿，然后停止定时器操作。有效沿通过预分频模式寄存器 00 (PRM00) 的第 4 位和第 5 位 (ES000, ES001) 来进行设置。

(5) 重新触发的单脉冲

(a) 通过软件的单脉冲输出

当一个单脉冲被输出时，不要设置 OSPT00 位为 1。在与 CR00 寄存器匹配时发生的 INTTM00 或者与 CR01 寄存器匹配时发生的 INTTM01 发生之前，不要再次输出单脉冲。

(b) 外部触发的单脉冲输出

如果在单脉冲输出期间外部触发再次产生，它将被忽略。

(c) 单脉冲输出功能

当使用软件触发的 16 位定时器 / 事件计数器 00 的单脉冲输出时，不要更改 TI000 引脚或其替换功能端口引脚的电平。

因为外部触发即使在这种情况下也会有效，在 TI000 引脚或其替换功能端口引脚的电平时定时器会被清除并启动，导致脉冲以不希望的时序输出。

(6) OVF00 标志的操作

<1> OVF00 标志在下列情况下被设置为 1。

当以下任意模式被选择时：TM00 和 CR00 匹配时清除&开始发生的模式，TI000 引脚的有效沿时清除&开始发生的模式或者自由运行模式

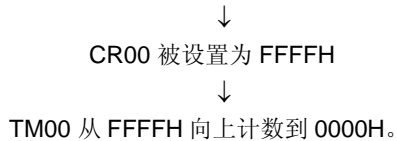
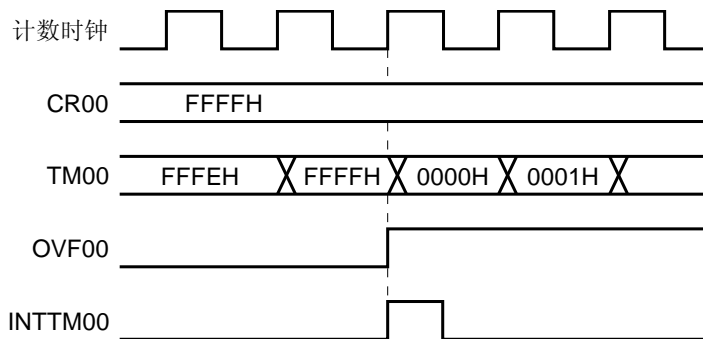


图 8-36. OVF00 标志的操作时序



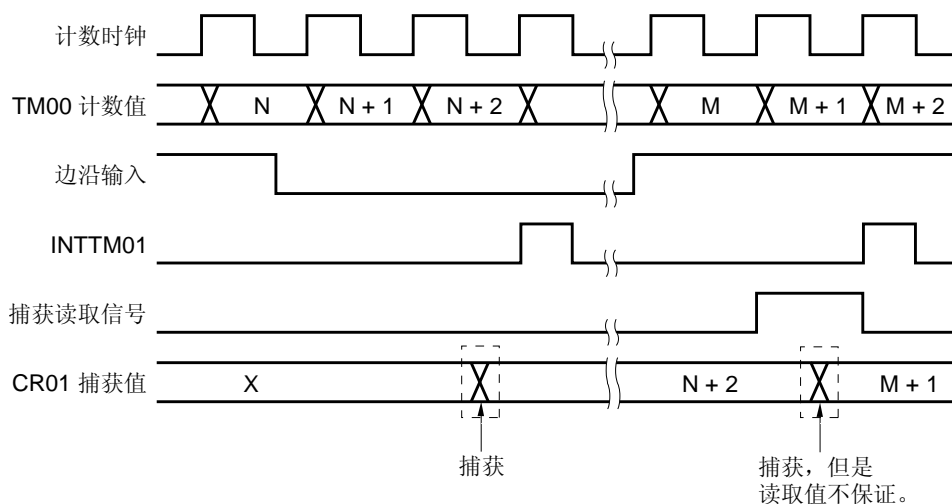
<2> 即使在 TM00 溢出发生之后下一个计数时钟（TM00 变为 0001H 之前）之前 OVF00 标志被清除，OVF00 标志也将被重新设置并且禁止清除。

(7) 冲突操作

16 位定时器捕获 / 比较寄存器（CR00 / CR01）的读取周期和捕获触发输入（CR00 / CR01 用作捕获寄存器）之间的冲突

捕获触发输入具有优先权。从 CR00 / CR01 中读取的数据不确定。

图 8-37. 捕获寄存器数据保持时序



(8) 定时器操作

- <1> 即使 16 位定时器计数器 00 (TM00) 被读取, 其数值也不会被 16 位定时器捕获 / 比较寄存器 01 (CR01) 捕获。
- <2> 不论 CPU 的操作模式, 当定时器停止时, 输入到 TI000/TI001 引脚的信号也不会被响应。
- <3> 单脉冲输出模式只在自由运行模式和 TI000 引脚有效沿时清除&开始发生的模式中正确运行。在 TM00 寄存器和 CR00 寄存器匹配时清除&开始发生的模式中, 单脉冲不可能输出, 因为溢出不会发生。

(9) 捕获操作

- <1> 如果 TI000 引脚的有效沿被指定为计数时钟, 由捕获寄存器指定作为 TI000 触发的捕获操作也没有可能会执行。
- <2> 要确保捕获操作的可靠性, 捕获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。
- <3> 捕获操作在计数时钟的下降沿执行。然而, 在下一个计数时钟的上升沿, 一个中断请求输入 (INTTM00 / INTTM01) 也会产生。

(10) 比较操作

即使捕获触发已经被输入, 捕获操作可能也不会对设置为比较模式的 CR00 / CR01 操作。

(11) 边沿检测

- <1> 如果在系统刚刚复位后 TI000 或 TI001 引脚为高电平, 并且上升沿或双边沿被设置为允许 16 位定时器计数器 00 (TM00) 操作的 TI000 引脚或 TI001 引脚的有效沿, 在操作被允许后, 上升沿立即被检测。因此, 当上拉 TI000 或 TI001 引脚时需要注意。但是, 操作停止一次后, 上升沿不会在重启时被检测。
- <2> 用于移除噪声的采样时钟与 TI000 引脚有效沿用作计数时钟和捕获触发时不同。在前者情况下, 计数时钟为 f_x , 而后者情况下, 计数时钟由预分频模式寄存器 00 (PRM00) 来选择。捕获操作只在有效沿采样到两次时才会执行, 这样可以消除短脉冲宽度的噪声。

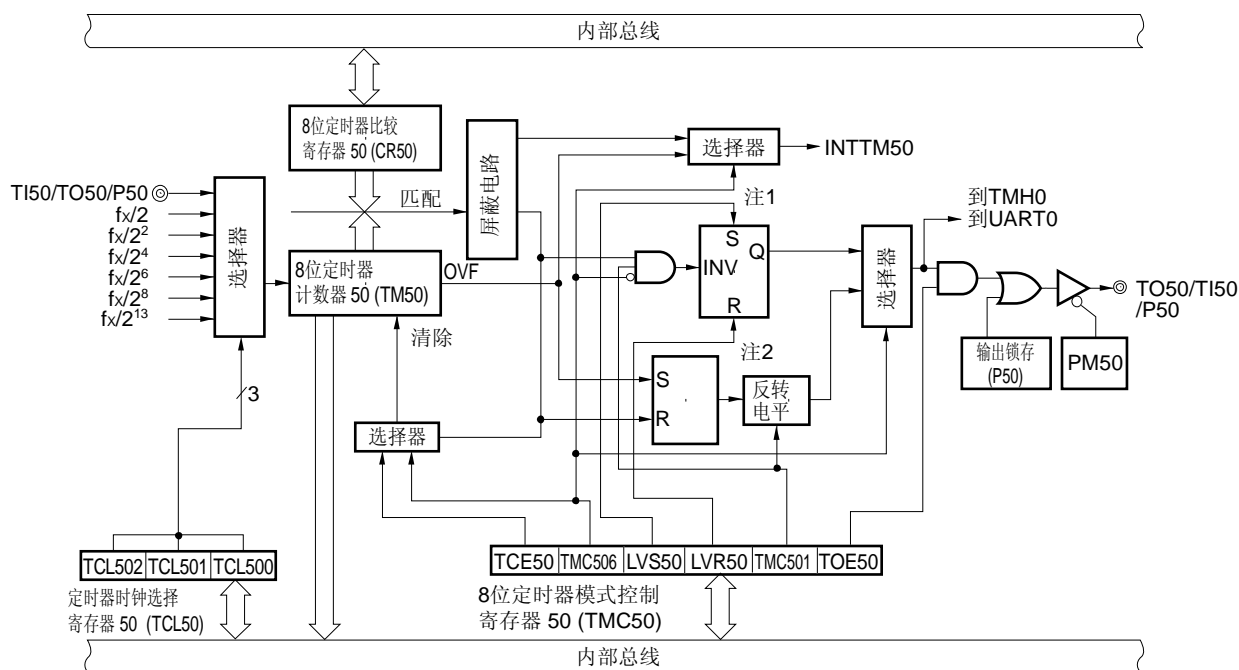
9.1 8 位定时器 / 事件计数器 50 和 51 的功能

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

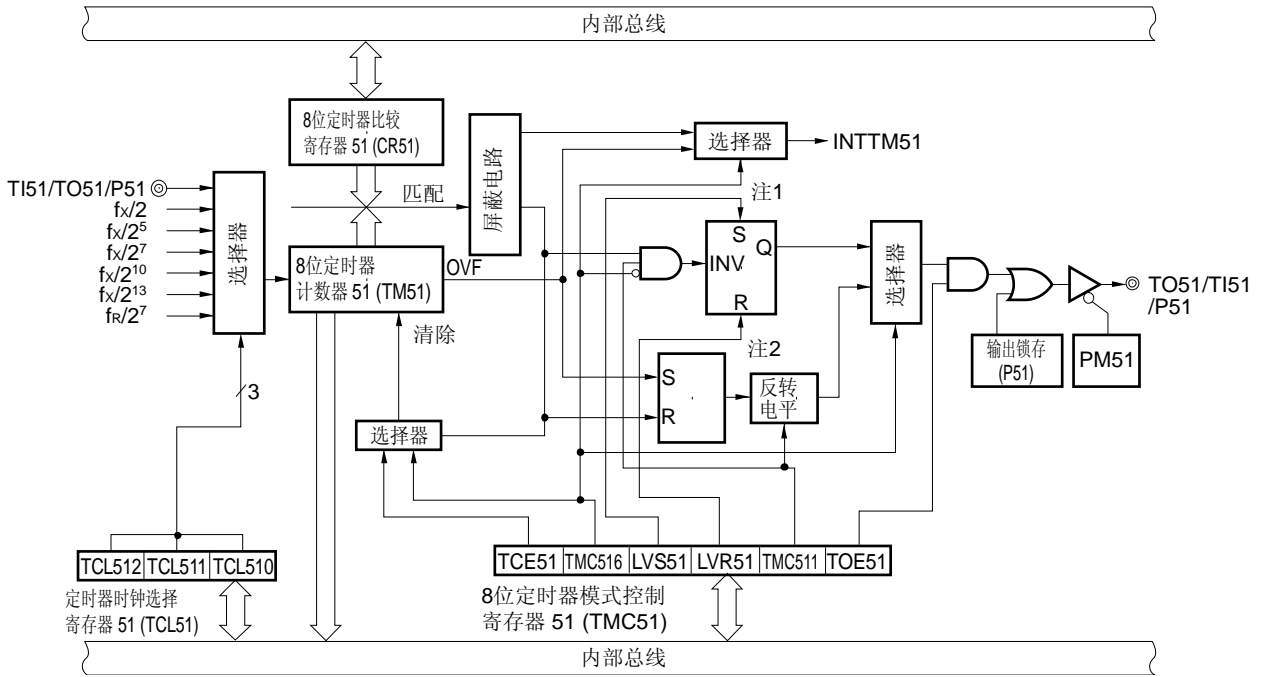
图 9-1 和 9-2 显示了 8 位定时器 / 事件计数器 50 和 51 的框图。

图 9-1. 8 位定时器 / 事件计数器 50 的框图



- 注
1. 定时器输出 F/F
 2. PWM 输出 F/F

图 9-2. 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

9.2 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 由以下硬件组成。

表 9-1. 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 5 (PM5) 端口寄存器 5 (P5)

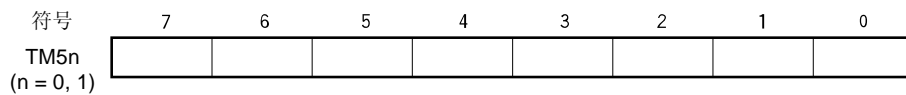
(1) 8 位定时器计数器 5n (TM5n)

TM5n 是一个对计数脉冲计数的 8 位寄存器，并且只读。

计数器与计数时钟的上升沿同步递增。

图 9-3. 8 位定时器计数器 5n (TM5n) 的格式

地址: FF2CH (TM50), FF3CH (TM51) 复位后: 00H R



在以下情况下，计数值被清除为 00H。

- <1> $\overline{\text{RESET}}$ 输入
- <2> 当 TCE5n 被清除时
- <3> 当 TM5n 和 CR5n 在 TM5n 和 CR5n 匹配时清除&开始发生的模式下匹配时

(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 中设置的值总是与 8 位定时器计数器 5n (TM5n) 的计数值比较，如果它们匹配，中断请求 (INTTM5n) 就会产生。

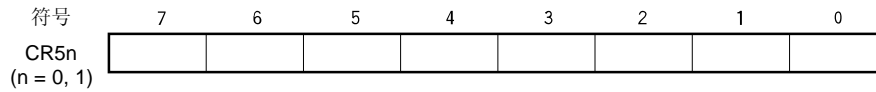
在 PWM 模式下，当 TO5n 引脚由于 TM5n 溢出以及 TM5n 的值和 CR5n 匹配而变为有效时，TO5n 引脚变为无效。

CR5n 的值可以在 00H 到 FFH 之间被设置。

$\overline{\text{RESET}}$ 输入清除 CR5n 为 00H。

图 9-4. 8 位定时器比较寄存器 5n (CR5n) 的格式

地址: FF2DH (CR50), FF3DH (CR51) 复位后: 00H R/W



- 注意事项**
1. 在 TM5n 和 CR5n 匹配时清除 & 开始发生的模式 (TMC5n6 = 0) 下，不要在操作期间向 CR5n 中写入其它值。
 2. 在 PWM 模式下，要使 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

9.3 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 5 (PM5)
- 端口寄存器 5 (P5)

(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 8 位存储器操作指令设置。

RESET 输入清除 TCL5n 为 00H。

备注 n = 0, 1

图 9-5. 定时器时钟选择寄存器 50 (TCL50) 的格式

地址: FF2EH 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
0	0	0	TI50 引脚下降沿
0	0	1	TI50 引脚上升沿
0	1	0	$f_x/2$ (10 MHz)
0	1	1	$f_x/2^2$ (5 MHz)
1	0	0	$f_x/2^4$ (1.25 MHz)
1	0	1	$f_x/2^6$ (312.5 kHz)
1	1	0	$f_x/2^8$ (78.125 kHz)
1	1	1	$f_x/2^{13}$ (2.44 kHz)

注 确保设置计数时钟使其满足以下条件。

- $V_{DD} = 4.0$ 到 5.5 V: 计数时钟 ≤ 10 MHz

注意事项 1. 当内部振荡时钟被选作 CPU 的源时钟时, 内部振荡器的时钟被分频并作为计数时钟提供。如果计数时钟为内部振荡时钟, 8 位定时器 / 事件计数器 50 的操作并不保证。

2. 当向 TCL50 重新写入其它数据时, 事先停止定时器操作。

3. 确保清除第 3 到 7 位为 0。

备注 1. f_x : X1 输入时钟振荡频率

2. 括号中的数字应用于 $f_x = 20$ MHz 的操作。

图 9-6. 时钟选择寄存器 51 (TCL51) 的格式

地址: FF3EH 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 [*]
0	0	0	TI51 引脚下降沿
0	0	1	TI51 引脚上升沿
0	1	0	$f_x/2$ (10 MHz)
0	1	1	$f_x/2^5$ (625 kHz)
1	0	0	$f_x/2^7$ (156.25 kHz)
1	0	1	$f_x/2^{10}$ (19.53 kHz)
1	1	0	$f_x/2^{13}$ (2.44 kHz)
1	1	1	$f_R/2^7$ (1.88 kHz)

注 确保设置计数时钟使其满足以下条件。

- $V_{DD} = 4.0$ 到 5.5 V: 计数时钟 ≤ 10 MHz

- 注意事项
1. 当内部振荡时钟被选作 CPU 的源时钟时, 内部振荡器的时钟被分频并作为计数时钟被提供。如果计数时钟为内部振荡时钟, 8 位定时器 / 事件计数器 51 的操作不被保证。
 2. 当向 TCL51 重新写入其它数据时, 事先停止定时器操作。
 3. 确保清除第 3 到 7 位为 0。

- 备注
1. f_x : X1 输入时钟振荡频率
 f_R : 内部振荡时钟频率
 2. 括号中的数字应用于 $f_x = 20$ MHz 和 $f_R = 240$ kHz (典型值) 的操作。

(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。

RESET 输入清除该寄存器为 00H。

备注 n = 0, 1

图 9-7. 8 位定时器模式控制寄存器 50 (TMC50) 的格式

地址: FF2FH 复位后: 00H R / W

符号	<7>	6	5	4	<3>	<2>	1	<0>
TMC50	TCE50	TMC506	0	0	LVS50	LVR50	TMC501	TOE50
	TMC50 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 操作模式选择							
	0	TM50 和 CR50 匹配时清除&开始发生的模式						
	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 复位 (0)					
	1	0	定时器输出 F/F 置位 (1)					
	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			高有效			
	1	反转操作允许			低有效			
	TOE50 定时器输出控制							
	0	输出禁止 (TM50 输出低电平)						
	1	输出允许						

(关于注意事项和备注, 参阅下页。)

图 9-8. 8 位定时器模式控制寄存器 51 (TMC51) 的格式

地址: FF3FH 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 匹配时清除&开始发生的模式							
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 复位 (0)						
1	0	定时器输出 F/F 置位 (1)						
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				高有效			
1	反转操作允许				低有效			
TOE51	定时器输出控制							
0	输出禁止 (TM51 输出低电平)							
1	输出允许							

- 注意事项**
1. LVS5n 和 LVR5n 的设置 在 PWM 模式以外有效。
 2. 不要同时重写以下位。
 - TMC5n1 和 TOE5n
 - TMC5n6 和 TOE5n
 - TMC5n1 和 TMC5n6
 - TMC5n6 和 LVS5n, LVR5n
 - TOE5n 和 LVS5n, LVR5n
 3. 重新写入 TMC5n6 前停止操作。

- 备注**
1. 在 PWM 模式下, 通过清除 TCE5n 为 0, PWM 输出变为无效。
 2. 如果 LVS5n 和 LVR5n 被读取, 值为 0。
 3. TMC5n6、LVS5n、LVR5n、TMC5n1 和 TOE5n 位的值在 TO5n 引脚上反映出来, 而不管 TCE5n 的值。
 4. n = 0, 1

(3) 端口模式寄存器 5 (PM5)

该寄存器以 1 位为单位设置端口 5 输入 / 输出。

当使用 P50/TO50/TI50 和 P51/TO51/TI51 引脚作为定时器输出时，清除 PM50 和 PM51 以及 P50 和 P51 的输出锁存为 0。

当使用 P50/TO50/TI50 和 P51/TO51/TI51 引脚作为定时器输入时，设置 PM50 和 PM51 为 1。P50 和 P51 的输出锁存这时可能为 0 或 1。

PM5 可以由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入设置该寄存器为 FFH。

图 9-9. 端口模式寄存器 5 (PM5) 的格式

地址: FF25H 复位后: FFH R / W

符号

	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

PM5n	P5n 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 / 事件计数器 50 和 51 的操作

9.4.1 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 值会清零并且中断请求信号 (INTTM5n) 会产生，此时计数继续。

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

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 匹配时清除&开始发生的模式。
(TMC5n = 0000xxx0B x = 不关注)

<2> TCE5n = 1 被设置后，计数操作开始。

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

<4> INTTM5n 按照相同的间隔重复产生。

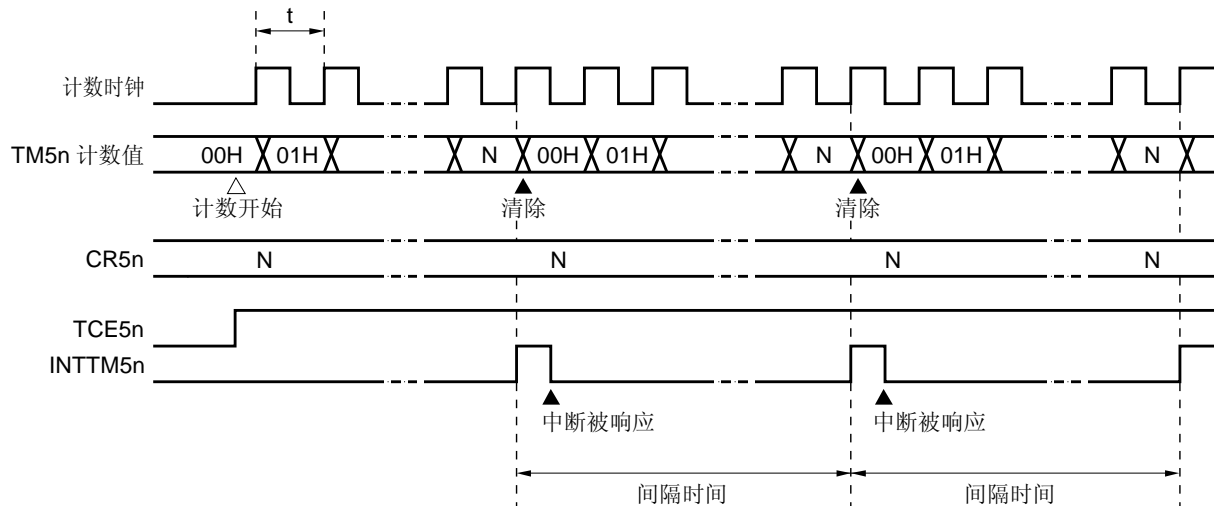
清除 TCE5n 为 0 来停止计数操作。

注意事项 不要在操作期间向 CR5n 写入其它值。

备注 n = 0, 1

图 9-10. 间隔定时器操作时序 (1 / 2)

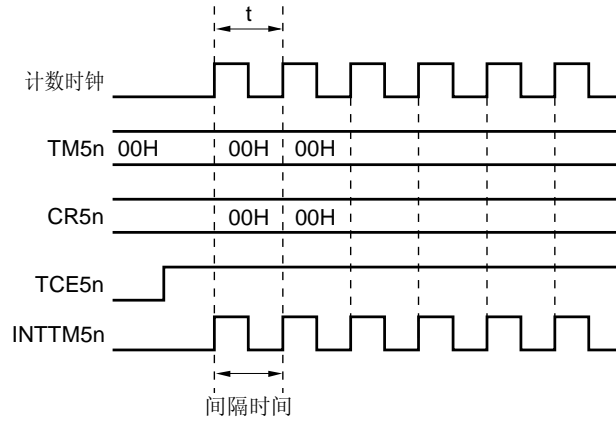
(a) 基本操作



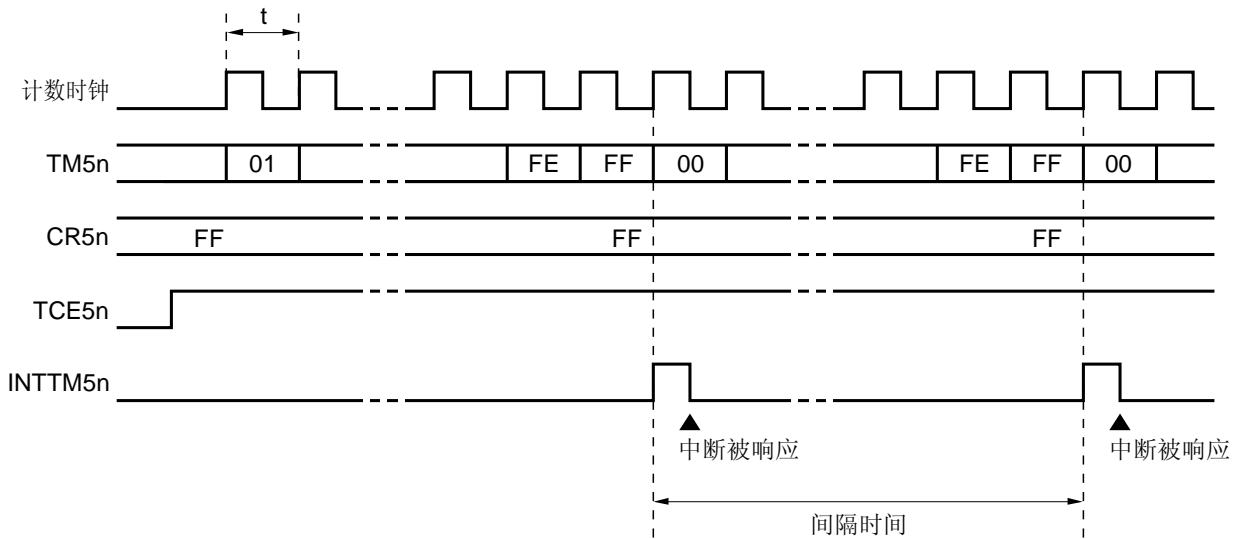
备注 间隔时间 = (N + 1) × t
N = 00H 到 FFH
n = 0, 1

图 9-10. 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

9.4.2 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。上升沿或下降沿都可以选择。

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

只要 TM5n 值与 CR5n 的值匹配, INTTM5n 就被产生。

设置

<1> 设置每个寄存器。

- 设置端口模式寄存器 (PM50 或 PM51) * 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 匹配时清除 & 开始发生的模式, 禁止定时器 F/F 反转操作, 禁止定时器输出。
(TMC5n = 0000x×00B x = 不关注)

<2> 当设置 TCE5n = 1 时, 从 TI5n 引脚输入的脉冲个数就会被计数。

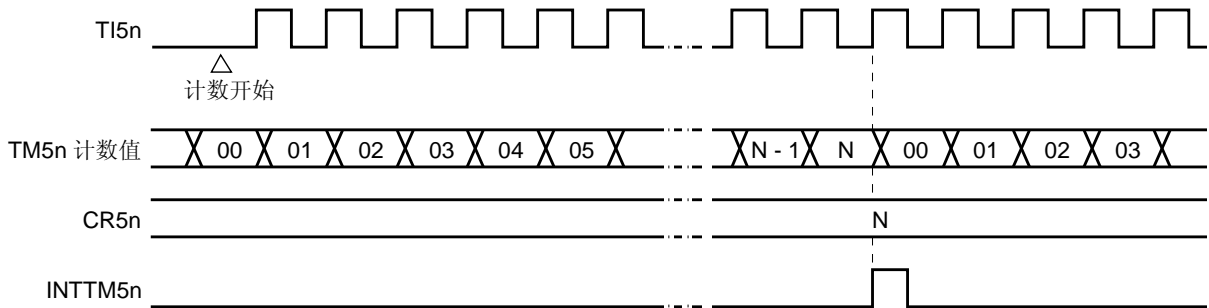
<3> 当 TM5n 和 CR5n 的值匹配时, INTTM5n 就会产生 (TM5n 被清除为 00H)。

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

注 8 位定时器 / 事件计数器 50: PM50

8 位定时器 / 事件计数器 51: PM51

图 9-11. 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

9.4.3 方波输出操作

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

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, TO5n 引脚的输出状态按照预先设置到 CR5n 中的计数值所确定的间隔反转。这样就使能允许任意选定频率的方波输出 (占空比= 50%)。

设置

<1> 设置每个寄存器。

- 清除端口输出锁存 (P50 或 P51) ^注 和端口模式寄存器 (PM50 或 PM51) ^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 匹配时清除&开始发生的模式。

LVS5n	LVR5n	定时器输出 F/F 状态设置
1	0	高电平输出
0	1	低电平输出

定时器输出 F/F 反转允许

定时器输出允许

(TMC5n = 00001011B 或 00000111B)

<2> TCE5n = 1 被设置后, 计数操作开始。

<3> 定时器输出 F/F 由 TM5n 和 CR5n 的匹配而反转。在 INTTM5n 被产生后, TM5n 清除为 00H。

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

频率如下所示。

$$\text{频率} = 1/2t(N + 1)$$

(N: 00H 到 FFH)

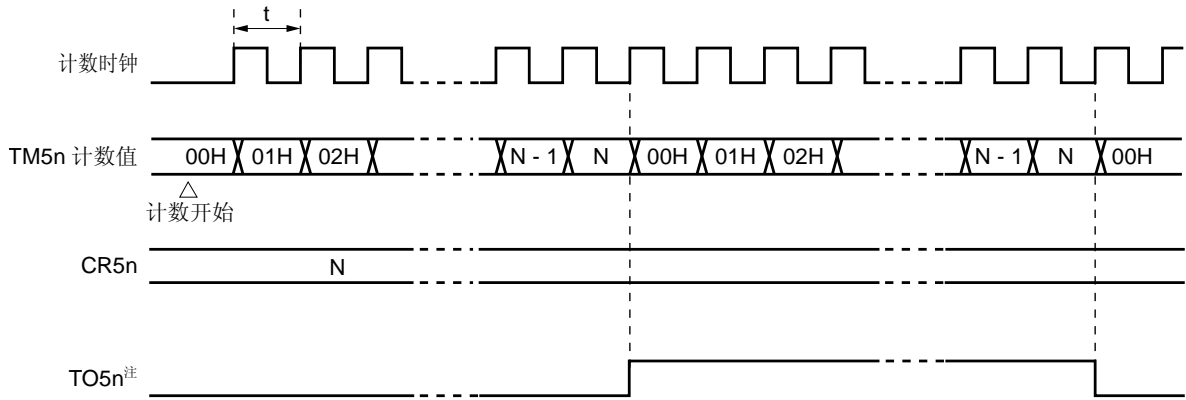
注 8 位定时器 / 事件计数器 50: P50, PM50

8 位定时器 / 事件计数器 51: P51, PM51

注意事项 不要在操作期间向 CR5n 写入其它值。

备注 n = 0, 1

图 9-12. 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。

9.4.4 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。

由设置到 8 位定时器比较寄存器 5n (CR5n) 中的值所确定的占空比的脉冲从 TO5n 输出。

设置 PWM 脉冲的有效电平宽度到 CR5n 中; 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 在 PWM 模式下, 要使 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 $n = 0, 1$

(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 (P50 或 P51)[※] 和端口模式寄存器 (PM50 或 PM51)[※] 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	高有效
1	低有效

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 来停止计数操作。

注 8 位定时器 / 事件计数器 50: P50, PM50
8 位定时器 / 事件计数器 51: P51, PM51

PWM 输出操作

- <1> PWM 输出 (从 TO5n 输出) 在溢出发生前会输出一个无效电平。
- <2> 当溢出发生时, 有效电平会输出。在 CR5n 与 8 位定时器计数器 5n (TM5n) 的计数值匹配前, 有效电平都会输出。
- <3> 在 CR5n 与计数值匹配后, 在溢出再次发生前, 无效电平会输出。
- <4> 操作 <2>和<3>重复执行, 直到计数操作停止。
- <5> 当计数操作由 TCE5n = 0 停止时, PWM 输出变为无效。
关于时序的细节, 见图 9-13 和 9-14。

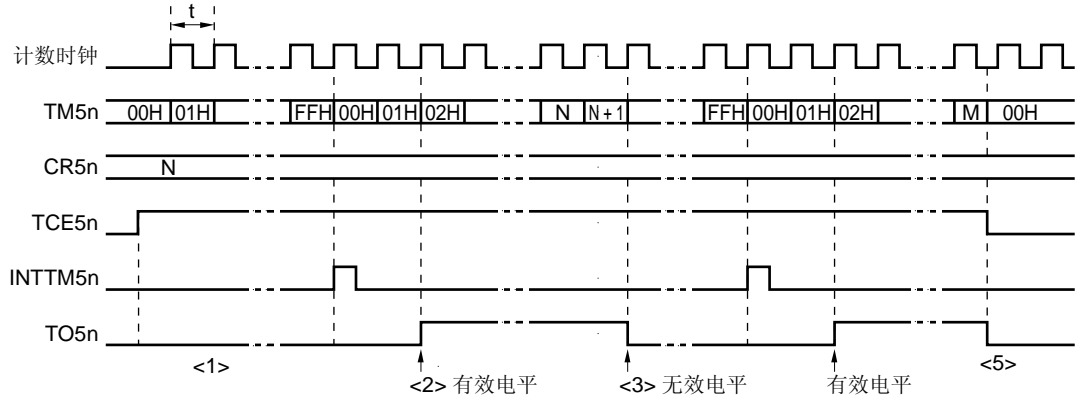
周期、有效电平宽度和占空比如下所示。

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

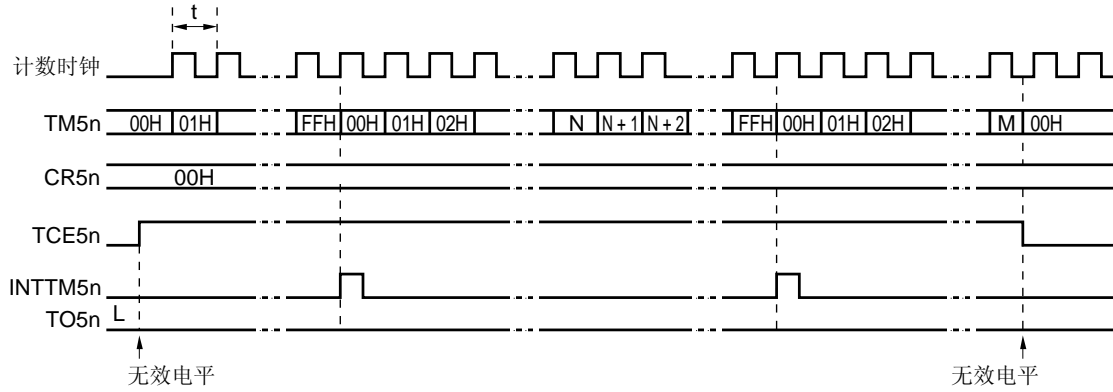
备注 n = 0, 1

图 9-13. PWM 输出操作时序

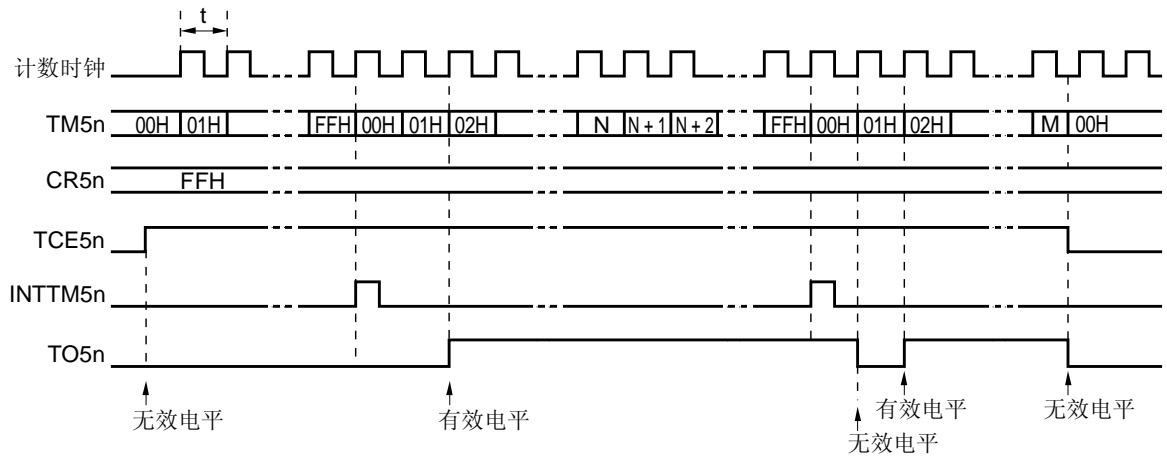
(a) 基本操作 (有效电平 = H)



(b) CR5n = 00H



(c) CR5n = FFH

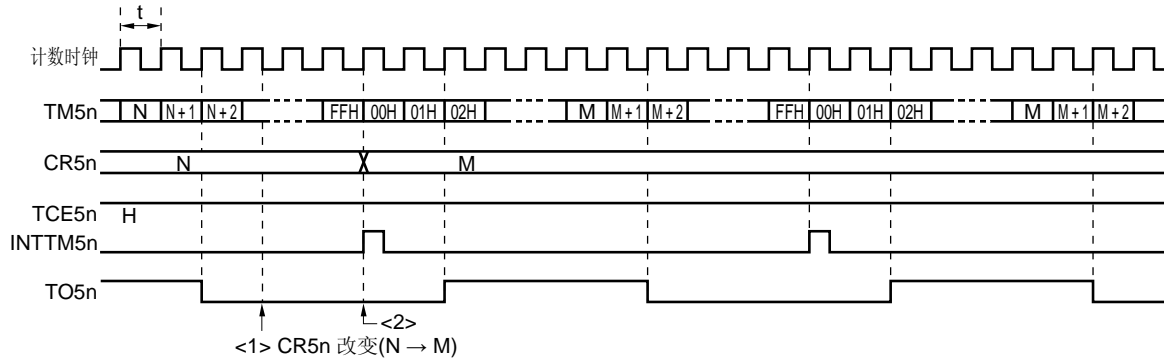


- 备注
- 图 9-13 (a) 中的<1> 到 <3> 和 <5> 对应于 9.4.4 (1) PWM 输出基本操作中 PWM 输出操作 中的 <1>到<3>和<5>。
 - $n = 0, 1$

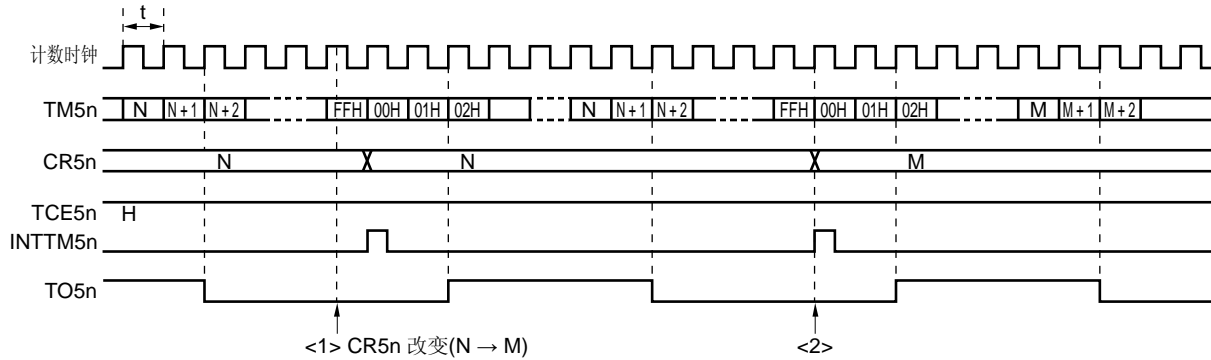
(2) CR5n 改变的操作

图 9-14. 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 → 值在改变后立即产生的溢出时转移到 CR5n。



(b) CR5n 值在 FFH 的时钟上升沿后从 N 变为 M → 值在第二个溢出时转移到 CR5n。



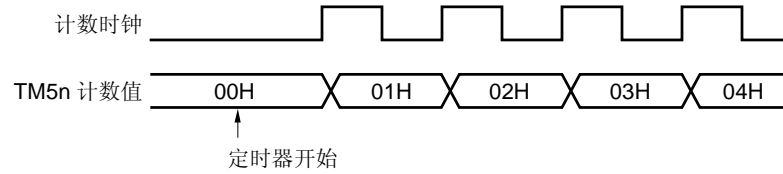
注意事项 当在图 9-14 中的 <1> 和 <2> 之间从 CR5n 读取时, 读取的值与实际值不同 (读取值: M, CR5n 的实际值: N)。

9.5 8 位定时器 / 事件计数器 50 和 51 的注意事项

(1) 定时器开始错误

在定时器开始后产生匹配信号需要的时间内，最多一个时钟的错误可能发生。这是因为 8 位定时器计数器 50 和 51 (TM50, TM51) 与计数时钟异步开始。

图 9-15. 8 位定时器计数器 5n 开始时序



备注 $n = 0, 1$

第十章 8 位定时器H0

10.1 8 位定时器H0 的功能

8 位定时器 H0 有以下功能。

- 间隔定时器
- PWM 输出模式
- 方波输出

10.2 8 位定时器H0 的配置

8 位定时器 H0 由以下硬件组成。

表 10-1. 8 位定时器 H0 的配置

项目	配置
定时器寄存器	8 位定时器计数器 H0
寄存器	8 位定时器 H 比较寄存器 00 (CMP00) 8 位定时器 H 比较寄存器 01 (CMP01)
定时器输出	TOH0
控制寄存器	8 位定时器 H 模式寄存器 0 (TMHMD0) 端口模式寄存器 5 (PM5) 端口寄存器 5 (P5)

图 10-1 表示框图。

(1) 8 位定时器 H 比较寄存器 00 (CMP00)

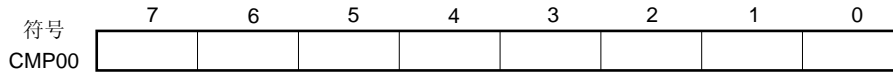
该寄存器可以由 8 位存储器操作指令读取或写入。

如果定时器计数器的值与 CMP00 的值匹配，一个中断请求信号 (INTTMH0) 就会产生。同时，定时器计数器的值被清除。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 10-2. 8 位定时器 H 比较寄存器 00 (CMP00) 的格式

地址: FF9EH 复位后: 00H R/W



注意事项 CMP00 在定时器计数操作期间不能重新写入。

(2) 8 位定时器 H 比较寄存器 01 (CMP01)

该寄存器可以由 8 位存储器操作指令读取或写入。

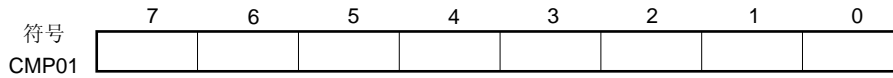
该寄存器只在 PWM 输出模式下使用。

如果定时器计数器的值与 CMP01 的值匹配，定时器计数器的值被清除，但是中断请求信号 (INTTMH0) 不产生。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 10-3. 8 位定时器 H 比较寄存器 01 (CMP01) 的格式

地址: FF9FH 复位后: 00H R/W



CMP01 可以在定时器计数操作期间重新写入。

如果 CMP01 的值在定时器操作期间重写，计数器的值与 CMP01 的值匹配时传输就会执行。如果传输时序和从 CPU 写入 CMP01 冲突，传输不被执行。

注意事项 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，要确保设置 CMP01 (即使设置相同的值到 CMP01，也要确保再次设置)。

10.3 控制 8 位定时器H0 的寄存器

以下三个寄存器用来控制 8 位定时器 H0。

- 8 位定时器 H 模式寄存器 0 (TMHMD0)
- 端口模式寄存器 5 (PM5)
- 端口寄存器 5 (P5)

(1) 8 位定时器 H 模式寄存器 0 (TMHMD0)

该寄存器控制定时器 H 的模式。

该寄存器可以由 1 位活 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 10-4. 8 位定时器 H 模式寄存器 0 (TMHMD0) 的格式

地址: FF2AH 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 (fcnt) 选择 ^{注1}					
0	0	0	f _x /2 (10 MHz)					
0	0	1	f _x /2 ³ (2.5 MHz)					
0	1	0	f _x /2 ⁶ (312.5 kHz)					
0	1	1	f _x /2 ⁸ (78.125 kHz)					
1	0	0	f _x /2 ¹⁰ (19.53 kHz)					
1	0	1	TM50 输出 ^{注2}					
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
1. 确保设置计数时钟使其满足以下条件。
 - V_{DD} = 4.0 到 5.5 V: 计数时钟 ≤ 10 MHz
 2. 当 TM50 输出被选作计数时钟时, 遵守以下条件。
 - PWM 模式 (TMC506 = 1)
 - 设置时钟以使占空比为 50% 并且事先启动 8 位定时器 / 事件计数器 50 的操作。
 - TM50 和 CR50 匹配时进入清除&开始模式 (TMC506 = 0)
 - 允许定时器 F/F 反转操作 (TMC501 = 1) 并且事先启动 8 位定时器 / 事件计数器 50 的操作。
 无论在何种模式下, 不需要使能允许 TO50 引脚作为定时器输出引脚 (TMC50 寄存器的第 0 位 (TOE50) 可能是 0 或 1)。

- 注意事项**
1. 当内部振荡时钟被选作 CPU 的源时钟时，内部振荡器的时钟被分频并作为计数时钟提供。如果计数时钟为内部振荡时钟，8 位定时器 H0 的操作并不保证。
 2. 当 $TMHE0 = 1$ 时，禁止设置 $TMHMD0$ 的其它位。
 3. 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，要确保设置 8 位定时器 H 比较寄存器 01 ($CMP01$) (即使设置相同的值到 $CMP01$ ，也要确保再次设置)。

- 备注**
1. fx: X1 输入时钟振荡频率
 2. 括号中的数字应用于 $fx = 20 \text{ MHz}$ 的操作。
 3. 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

(2) 端口模式寄存器 5 (PM5)

该寄存器以 5 位为单位设置端口 1 输入 / 输出。

当使用 P52/TOH0/INTP4 引脚作为定时器输出时，清除 PM52 和 P52 的输出锁存为 0。

PM5 可以由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入设置该寄存器为 FFH。

图 10-5. 端口模式寄存器 5 (PM5) 的格式

地址: FF25H 复位后: FFH R / W

符号	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50
PM5n	P5n 引脚 I / O 模式选择 (n = 0 到 7)							
0	输出模式 (输出缓存打开)							
1	输入模式 (输出缓存关闭)							

10.4 8 位定时器 H0 的操作

10.4.1 作为间隔定时器 / 方波输出操作

当 8 位定时器计数器 H0 和比较寄存器 00 (CMP00) 的值匹配时, 中断请求信号 (INTTMH0) 会产生并且 8 位定时器计数器 H0 被清除为 00H。

比较寄存器 01 (CMP01) 不用于间隔定时器模式。因为即使 CMP01 寄存器被设置, 8 位定时器计数器 H0 和 CMP01 寄存器的匹配也不被检测, 定时器输出不受影响。

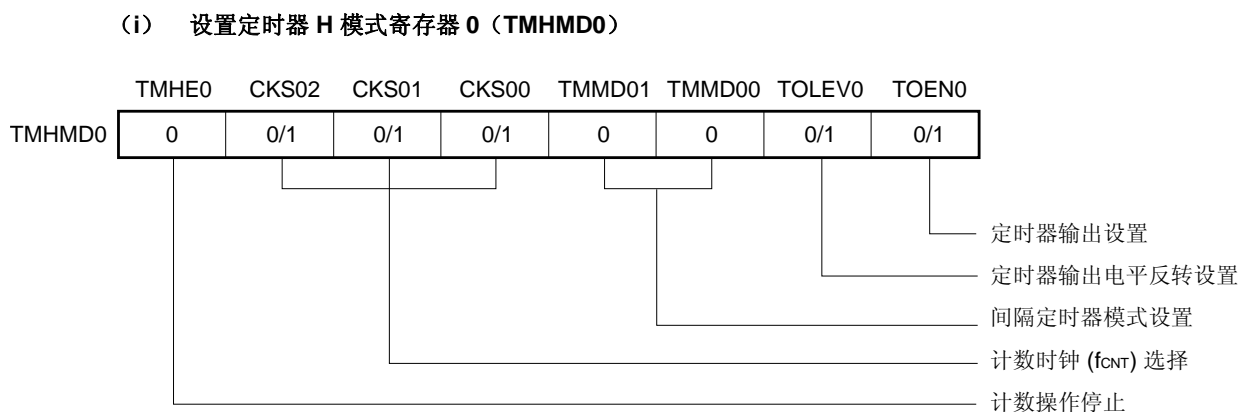
通过设置定时器 H 模式寄存器 0 (TMHMD0) 的第 0 位 (TOEN0) 为 1, 任意频率 (占空比 = 50%) 的方波从 TOH0 被输出。

(1) 用法

以相同的间隔重复产生 INTTMH0 信号。

<1> 设置每个寄存器。

图 10-6. 间隔定时器 / 方波输出操作期间的寄存器设置



(ii) CMP00 寄存器设置

- 比较值 (N)

<2> 当 TMHE0 = 1 时, 计数操作开始。

<3> 当 8 位定时器计数器 H0 和 CMP00 寄存器的值匹配时, INTTMH0 会产生并且 8 位定时器计数器 H0 被清除为 00H。

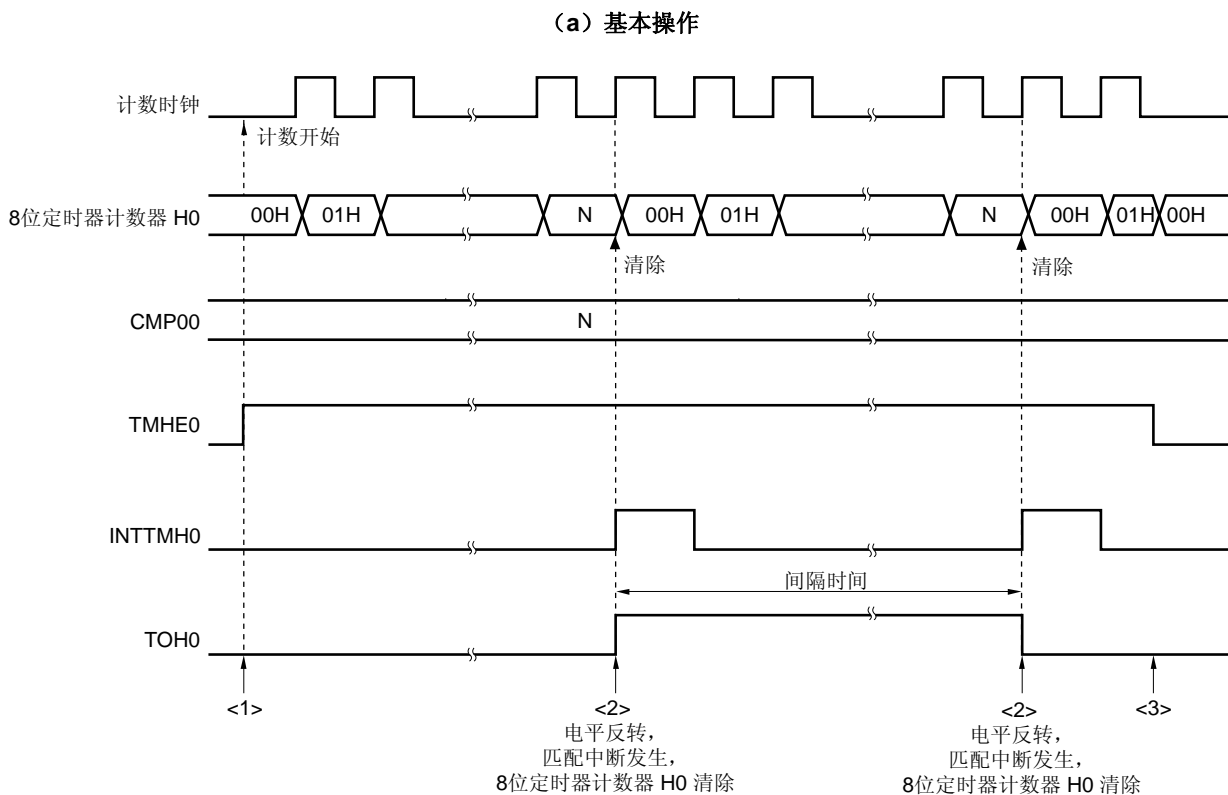
$$\text{间隔时间} = (N + 1) / f_{\text{CNT}}$$

<4> 然后, INTTMH0 信号会以同样间隔产生。要停止计数操作, 清除 TMHE0 为 0。

(2) 时序图

间隔定时器 / 方波输出操作的时序如下所示。

图 10-7. 间隔定时器 / 方波输出操作时序 (1 / 2)

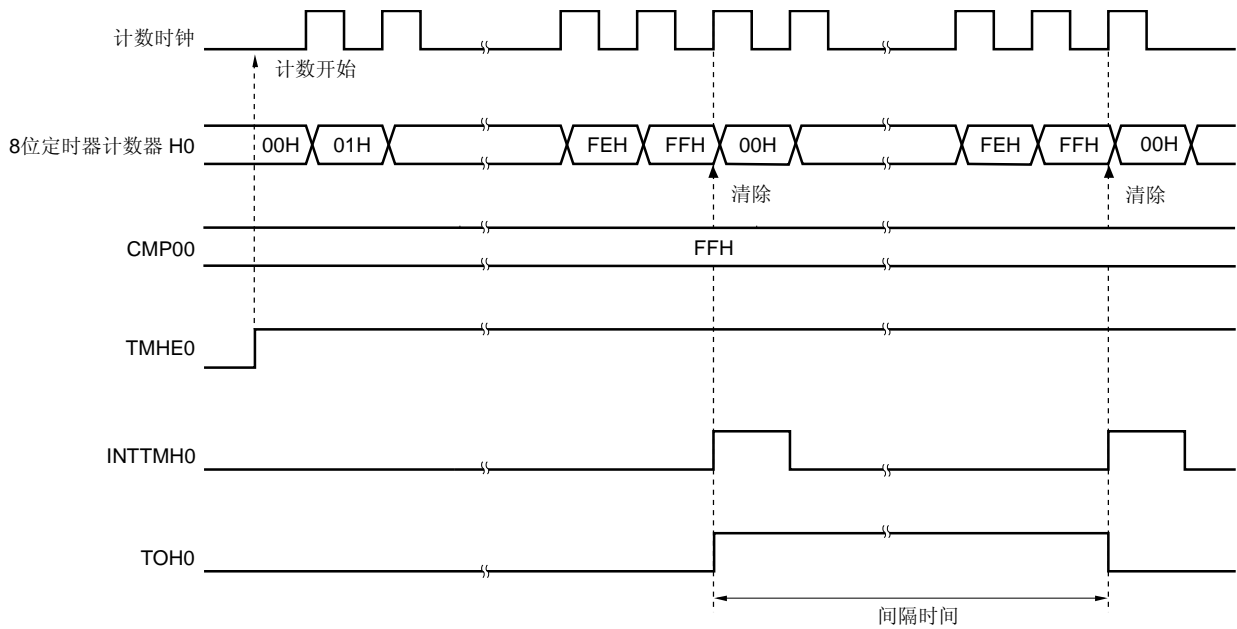


- <1> 通过设置 **TMHE0** 位为 1，计数操作就会使能允许。在操作允许后不超过一个时钟之内，计数时钟就会开始计数。
- <2> 当 8 位定时器计数器 **H0** 的值和 **CMP00** 寄存器匹配时，8 位定时器计数器 **H0** 被清除，**TOH0** 输出电平被反转，并且 **INTTMH0** 信号也会输出。
- <3> 在定时器 **H0** 操作期间，通过清除 **TMHE0** 位为 0，**INTTMH0** 信号和 **TOH0** 输出会变为无效。如果从第一个就存在无效电平，电平会被保持。

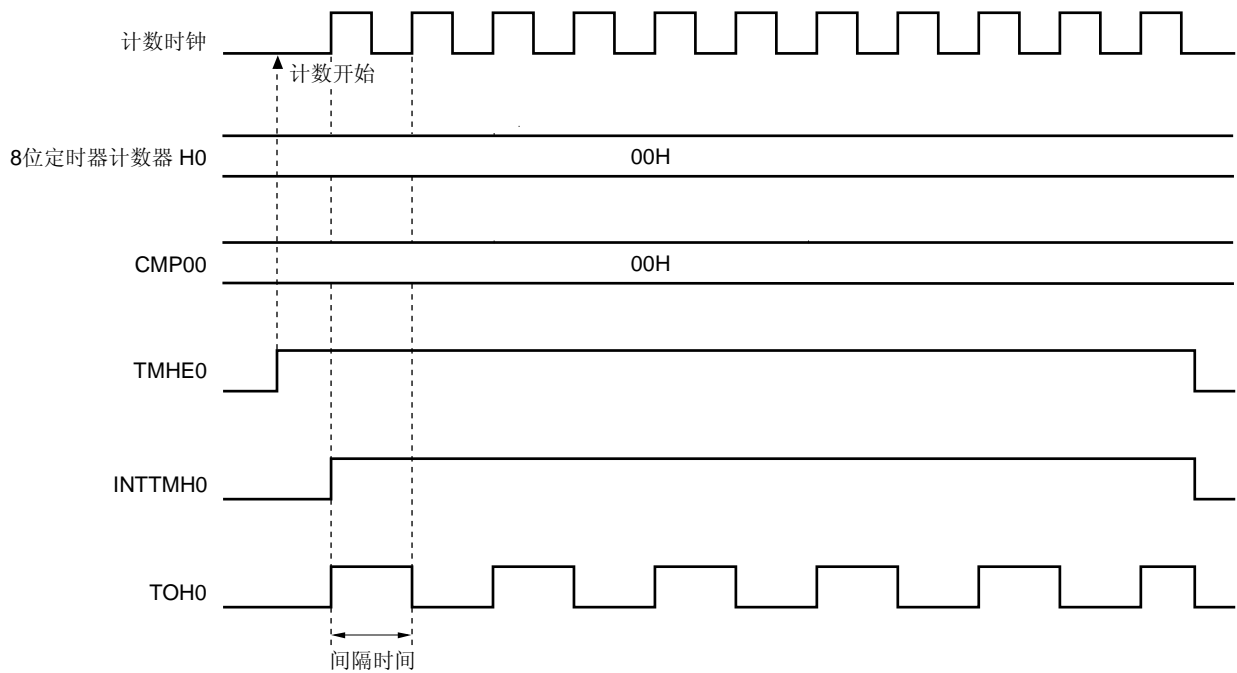
备注 **N = 01H 到 FEH**

图 10-7. 间隔定时器 / 方波输出操作时序 (2 / 2)

(b) 当 CMP00= FFH 时的操作



(c) 当 CMP00 = 00H 时的操作



10.4.2 作为PWM输出模式操作

PWM 输出模式下，任意占空比和任意周期的脉冲都可以输出。

8 位定时器比较寄存器 00 (CMP00) 控制定时器输出 (TOH0) 的周期。在定时器操作期间，禁止重新写入 CMP00 寄存器。

8 位定时器比较寄存器 01 (CMP01) 控制定时器输出 (TOH0) 的占空比。在定时器操作期间，可以重新写入 CMP01 寄存器。

PWM 输出模式下的操作如下所示。

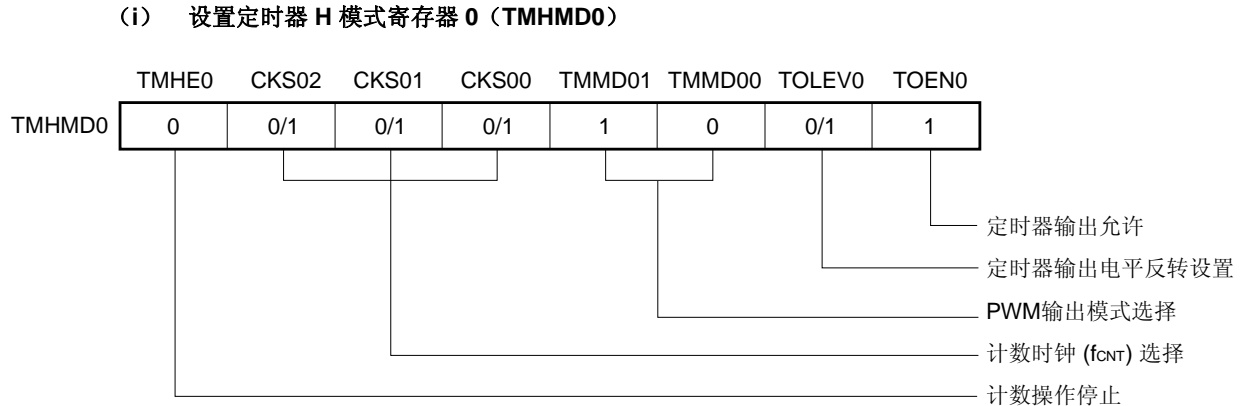
在定时器计数开始后，当 8 位定时器计数器 H0 和 CMP00 寄存器匹配时，TOH0 输出电平变为有效并且 8 位定时器计数器 H0 会清除为 0。当 8 位定时器 H0 和 CMP01 寄存器匹配时，TOH0 输出变为无效。

(1) 用法

PWM 输出模式下，设定的任意占空比和任意周期的脉冲就会输出。

<1> 设置每个寄存器。

图 10-8. PWM 输出模式下的寄存器设置



(ii) 设置 CMP00 寄存器

- 比较值 (N)：周期设置

(iii) 设置 CMP01 寄存器

- 比较值 (M)：占空比设置

备注 $00H \leq \text{CMP01 (M)} < \text{CMP00 (N)} \leq \text{FFH}$

<2> 当 TMHE0 = 1 时，计数操作开始。

<3> CMP00 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 H0 的值和 CMP00 寄存器匹配时，8 位定时器计数器 H0 被清除，中断请求信号 (INTTMH0) 会产生，并且 TOH0 输出变为有效。同时，与 8 位定时器计数器 H0 比较的比较寄存器从 CMP00 寄存器改变为 CMP01 寄存器。

- <4> 当 8 位定时器计数器 H0 和 CMP01 寄存器匹配时，TOH0 输出变为无效并且与 8 位定时器计数器 H0 比较的比较寄存器从 CMP01 寄存器改变为 CMP00 寄存器。这时，8 位定时器计数器 H0 不被清除并且 INTTMH0 信号不产生。
- <5> 通过重复执行过程<3>和<4>，一个具有任意周期的脉冲就会得到。
- <6> 要停止计数操作，设置 $TMHE0 = 0$ 。

如果 CMP00 寄存器的设置值为 N，CMP01 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。

$$\begin{aligned} \text{PWM 脉冲输出周期} &= (N+1)/f_{CNT} \\ \text{占空比} &= \text{有效宽度} : \text{PWM 的总宽度} = (M + 1) : (N + 1) \end{aligned}$$

- 注意事项**
1. 在 PWM 输出模式下，在重新写入寄存器后需要三个操作时钟（使用 TMHMD0 寄存器的 CKS02 到 CKS00 位选择的信号）周期来传输 CMP01 寄存器值。
 2. 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，要确保设置 CMP01 寄存器（即使设置相同的值到 CMP01 寄存器，也要确保再次设置）。

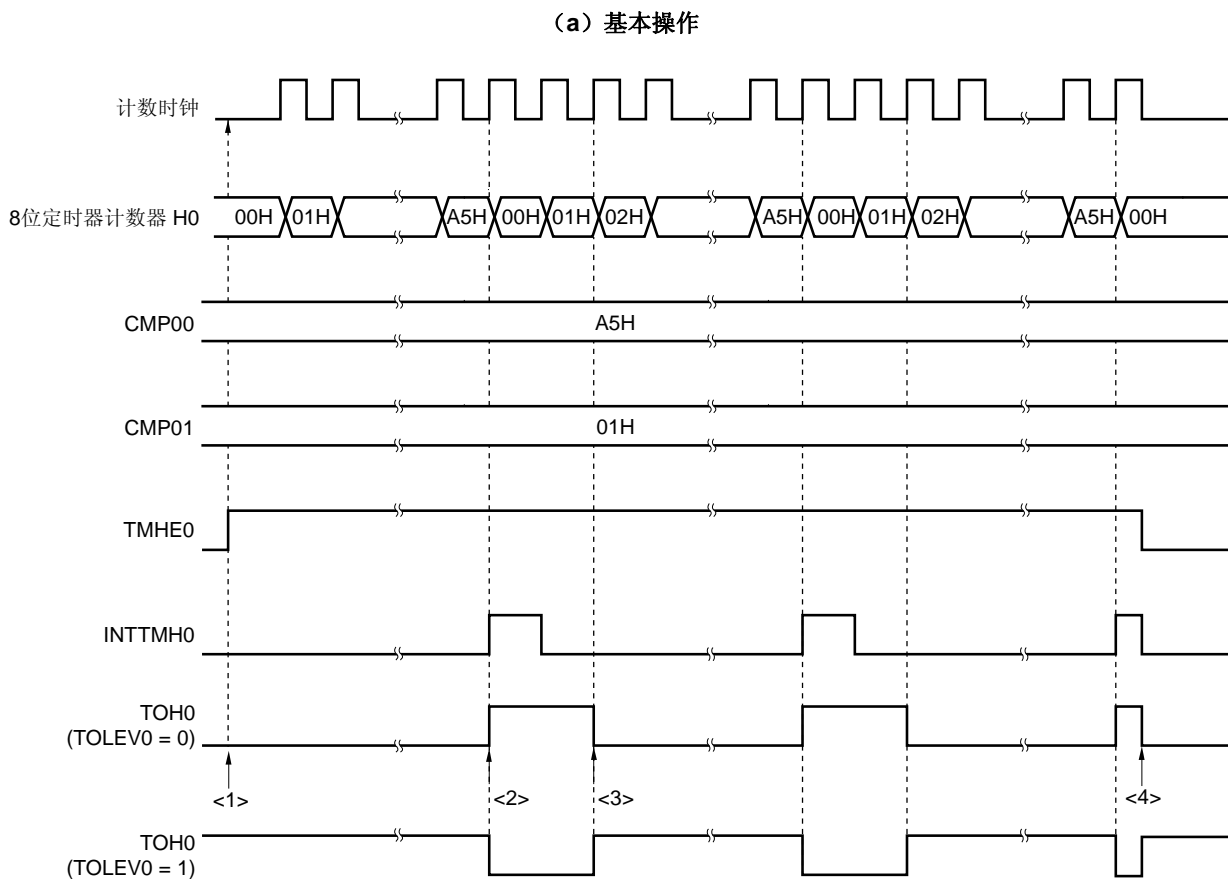
(2) 时序图

PWM 输出模式下的操作时序如下所示。

注意事项 确保 CMP01 寄存器的设置值 (M) 和 CMP00 寄存器的设置值 (N) 在以下范围内。

$$00H \leq \text{CMP01 (M)} < \text{CMP00 (N)} \leq \text{FFH}$$

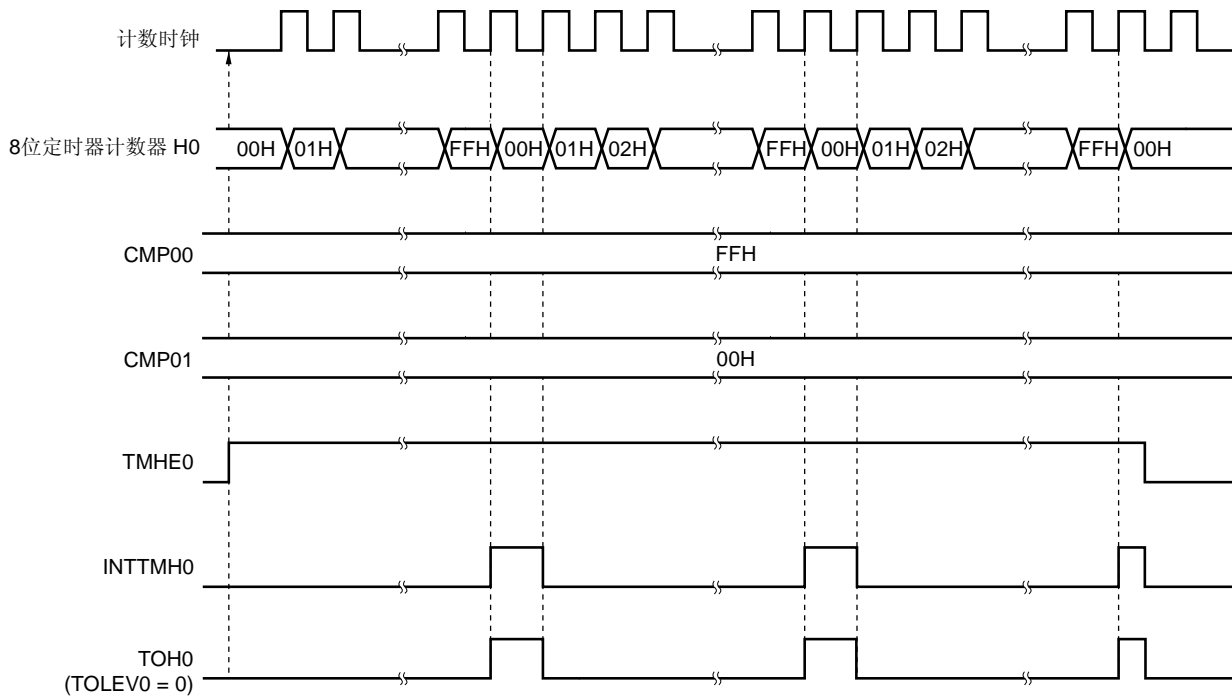
图 10-9. PWM 输出模式下的操作时序 (1 / 4)



- <1> 计数操作通过设置 TMHE0 位为 1 允许。通过屏蔽一个计数时钟来启动 8 位定时器计数器 H0 向上计数。同时，TOH0 输出保持无效（当 TOLEV0 = 0 时）。
- <2> 当 8 位定时器计数器 H0 的值和 CMP00 寄存器匹配时，TOH0 的输出电平反转，8 位定时器计数器 H0 被清除，并且 INTTMH0 信号也会输出。
- <3> 当 8 位定时器计数器 H0 的值和 CMP01 寄存器匹配时，TOH0 输出电平被返回。这时，8 位定时器计数器的值不被清除并且 INTTMH0 信号不输出。
- <4> 在定时器 H0 操作期间清除 TMHE0 位为 0 可以设置 INTTMH0 信号和 TOH0 输出无效。

图 10-9. PWM 输出模式下的操作时序 (2 / 4)

(b) 当 CMP00 = FFH, CMP01 = 00H 时的操作



(c) 当 CMP00 = FFH, CMP01 = FEH 时的操作

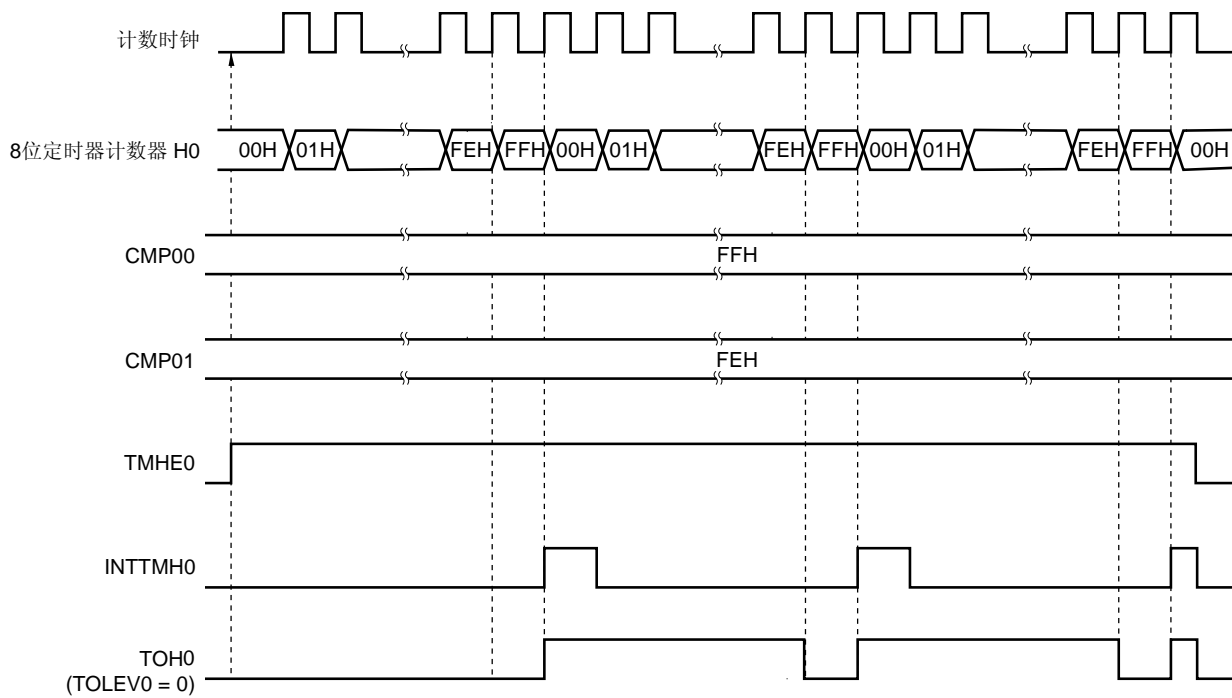


图 10-9. PWM 输出模式下的操作时序 (3 / 4)

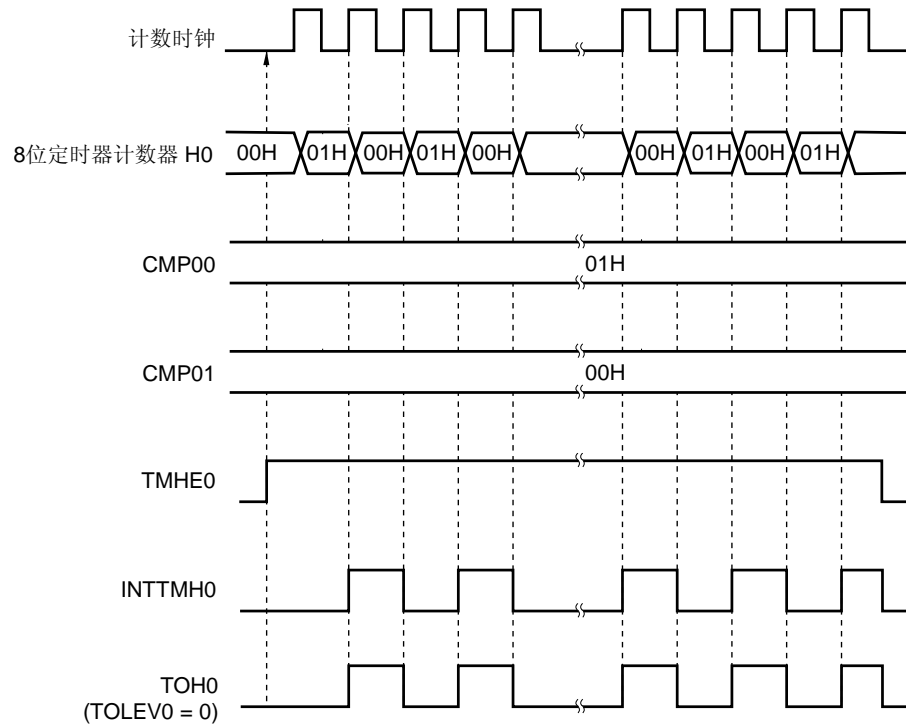
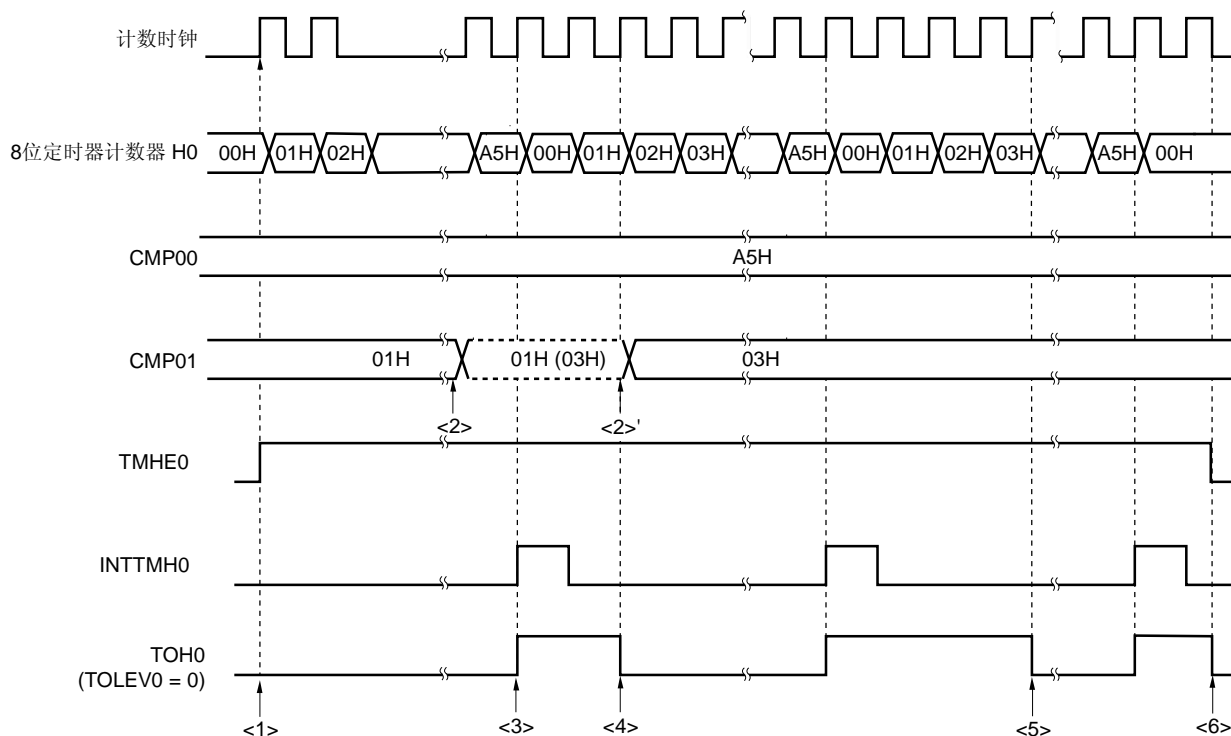
(d) 当 $CMP00 = 01H$, $CMP01 = 00H$ 时的操作

图 10-9. PWM 输出模式下的操作时序 (4 / 4)

(e) 通过改变 CMP01 (CMP01 = 01H → 03H, CMP00 = A5H) 的操作



- <1> 计数操作通过设置 TMHE0 位为 1 允许。通过屏蔽一个计数时钟来启动 8 位定时器计数器 H0 向上计数。同时, TOH0 输出保持无效 (当 TOLEV0 = 0 时)。
- <2> CMP01 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 H0 的值和 CMP00 寄存器匹配时, 8 位定时器计数器 H0 被清除, TOH0 输出电平变为有效, 并且 INTTMH0 信号也会输出。
- <4> 如果 CMP01 寄存器的值改变, 这个值会被锁存并且不会转移到寄存器。当 8 位定时器计数器 H0 的值与 CMP01 寄存器以改变前的值匹配时, 这个值被转移到 CMP01 寄存器并且 CMP01 寄存器的值被更改 (<2>')。
- 然而, 从 CMP01 寄存器的值被更改到这个值被转移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, 改变的值不能转移到寄存器。
- <5> 当 8 位定时器计数器 H0 的值和 CMP01 寄存器匹配时, TOH0 输出电平变为无效。8 位定时器计数器 H0 不清除并且 INTTMH0 信号也不会产生。
- <6> 在定时器 H0 操作期间清除 TMHE0 位为 0 可以设置 INTTMH0 信号和 TOH0 输出无效。

第十一章 看门狗定时器

11.1 看门狗定时器的功能

看门狗定时器被用来检测一个无意的程序循环。如果程序循环被检测，一个内部复位信号被产生。

当由于看门狗定时器的一个复位发生时，复位控制标志寄存器（RESF）的第 4 位（WDTRF）被设置为 1。关于 RESF 的细节，见第二十一章 复位功能。

表 11-1. 看门狗定时器的循环检测时间

循环检测时间	
内部振荡时钟操作期间	X1 输入时钟操作期间
$f_R/2^{11}$ (8.53 ms)	$f_{XP}/2^{13}$ (409.6 μ s)
$f_R/2^{12}$ (17.07 ms)	$f_{XP}/2^{14}$ (819.2 μ s)
$f_R/2^{13}$ (34.13 ms)	$f_{XP}/2^{15}$ (1.64 ms)
$f_R/2^{14}$ (68.27 ms)	$f_{XP}/2^{16}$ (3.28 ms)
$f_R/2^{15}$ (136.53 ms)	$f_{XP}/2^{17}$ (6.55 ms)
$f_R/2^{16}$ (273.07 ms)	$f_{XP}/2^{18}$ (13.11 ms)
$f_R/2^{17}$ (546.13 ms)	$f_{XP}/2^{19}$ (26.21 ms)
$f_R/2^{18}$ (1.09 s)	$f_{XP}/2^{20}$ (52.43 ms)

- 备注**
1. f_R : 内部振荡时钟频率
 2. f_{XP} : X1 输入时钟振荡频率
 3. 括号中的数字应用于 $f_R = 240$ kHz (典型值), $f_{XP} = 20$ MHz 的操作。

看门狗定时器 (WDT) 的操作模式根据内部振荡器的选项字节设置切换, 如表 11-2 所示。

表 11-2. 选项字节设置和看门狗定时器的操作模式

	选项字节	
	内部振荡器不能被停止	内部振荡器可以由软件停止
看门狗定时器时钟源	固定为 f_R ^{#1} .	<ul style="list-style-type: none"> 可由软件选择 (f_{XP}, f_R 或被停止) 当复位被释放时: f_R
复位后的操作	操作以最大间隔 ($f_R/2^{18}$) 开始。	操作以最大间隔 ($f_R/2^{18}$) 开始。
操作模式选择	溢出时间只能被更改一次。	时钟选择 / 溢出时间只能被更改一次。
特征	看门狗定时器不能被停止。	看门狗定时器可以被停止 ^{#2} 。

- 注**
- 只要电源被提供，内部振荡器就不能被停止（除非在复位周期内）。
 - 提供给看门狗定时器的时钟被停止的条件根据看门狗定时器的时钟源而不同。
 - <1> 如果时钟源为 f_{XP} ，提供给看门狗定时器的时钟在以下条件下被停止。
 - 当 f_{XP} 被停止时
 - 在 HALT/STOP 模式下
 - 在振荡稳定时间期间
 - <2> 如果时钟源为 f_R ，提供给看门狗定时器的时钟在以下条件下被停止。
 - 如果 CPU 时钟为 f_{XP} 并且如果 f_R 在 STOP 指令执行之前由软件停止
 - 在 HALT/STOP 模式下

- 备注**
- f_R : 内部振荡时钟频率
 - f_{XP} : X1 输入时钟振荡频率

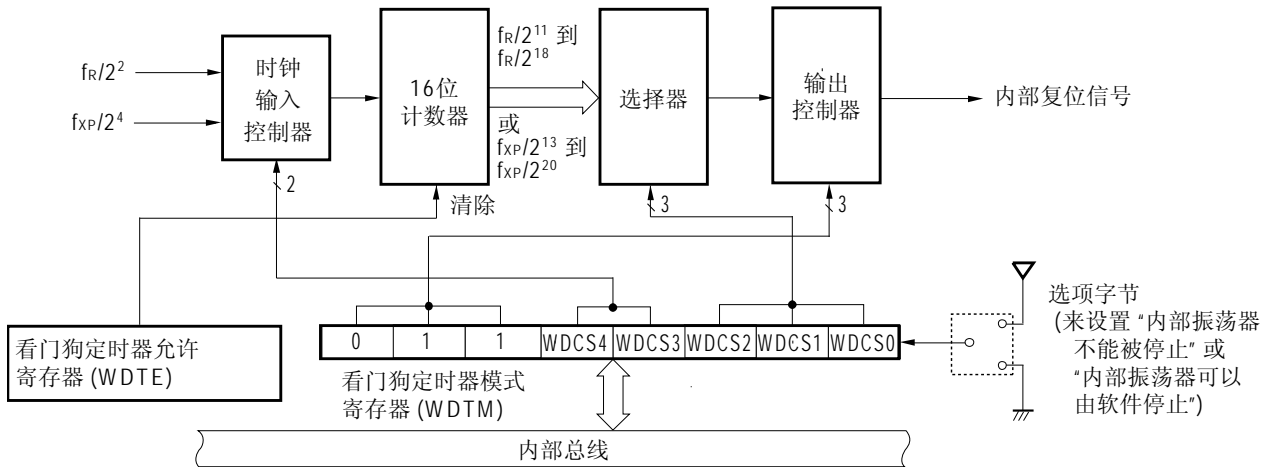
11.2 看门狗定时器的配置

看门狗定时器由以下硬件组成。

表 11-3. 看门狗定时器的配置

项目	配置
控制寄存器	看门狗定时器模式寄存器 (WDTM) 看门狗定时器允许寄存器 (WDTE)

图 11-1. 看门狗定时器的框图



11.3 控制看门狗定时器的寄存器

看门狗定时器由以下两个寄存器控制。

- 看门狗定时器模式寄存器 (WDTM)
- 看门狗定时器允许寄存器 (WDTE)

(1) 看门狗定时器模式寄存器 (WDTM)

该寄存器设置看门狗定时器的溢出时间和操作时钟。

该寄存器可以由 8 位存储器操作指令来设置并且可以多次读取，但是在复位释放后只能写入一次。

$\overline{\text{RESET}}$ 输入设置该寄存器为 67H。

图 11-2. 看门狗定时器模式寄存器 (WDTM) 的格式

地址: FF98H 复位后: 67H R/W

符号	7	6	5	4	3	2	1	0
WDTM	0	1	1	WDCS4	WDCS3	WDCS2	WDCS1	WDCS0

WDCS4 ^{注1}	WDCS3 ^{注1}	操作时钟选择
0	0	内部振荡时钟 (f_R)
0	1	X1 输入时钟 (f_{XP})
1	×	看门狗定时器操作被停止

WDCS2 ^{注2}	WDCS1 ^{注2}	WDCS0 ^{注2}	溢出时间设置	
			内部振荡时钟操作期间	X1 输入时钟操作期间
0	0	0	$f_R/2^{11}$ (8.53 ms)	$f_{XP}/2^{13}$ (409.6 μs)
0	0	1	$f_R/2^{12}$ (17.07 ms)	$f_{XP}/2^{14}$ (819.2 μs)
0	1	0	$f_R/2^{13}$ (34.13 ms)	$f_{XP}/2^{15}$ (1.64 ms)
0	1	1	$f_R/2^{14}$ (68.27 ms)	$f_{XP}/2^{16}$ (3.28 ms)
1	0	0	$f_R/2^{15}$ (136.53 ms)	$f_{XP}/2^{17}$ (6.55 ms)
1	0	1	$f_R/2^{16}$ (273.07 ms)	$f_{XP}/2^{18}$ (13.11 ms)
1	1	0	$f_R/2^{17}$ (546.13 ms)	$f_{XP}/2^{19}$ (26.21 ms)
1	1	1	$f_R/2^{18}$ (1.09 s)	$f_{XP}/2^{20}$ (52.43 ms)

- 注
1. 如果“内部振荡器不能被停止”由选项字节指定，该位不能被设置。无论何值被写入，内部振荡时钟都将被选择。
 2. 复位在最大周期被释放 (WDCS2, 1, 0 = 1, 1, 1)。

- 注意事项**
1. 如果数据被写入 **WDTM**，一个等待周期被产生。关于详细情况，参见第三十章 等待注意事项。
 2. 分别设置第 7、6 和 5 位为 0、1 和 1（当“内部振荡器不能被停止”由选项字节指定时，其它值被忽略）。
 3. 复位释放后，**WDTM** 只能通过 8 位存储器操作指令写入一次。如果试图写入两次，一个内部复位信号被产生。
 4. **WDTM** 不能通过 1 位存储器操作指令来设置。

- 备注**
1. f_R : 内部振荡时钟频率
 2. f_{XP} : X1 输入时钟振荡频率
 3. x: 不关注
 4. 括号中的数字应用于 $f_R = 240 \text{ kHz}$ （典型值）， $f_{XP} = 20 \text{ MHz}$ 的操作。

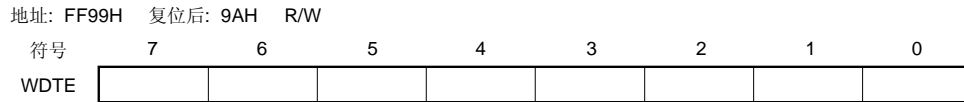
(2) 看门狗定时器允许寄存器 (WDTE)

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 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入设置该寄存器为 **9AH**。

图 11-3. 看门狗定时器允许寄存器 (WDTE) 的格式



- 注意事项**
1. 如果一个 **ACH** 以外的值被写入 **WDTE**，一个内部复位信号被产生。
 2. 如果对 **WDTE** 的一个 1 位存储器操作指令被执行，一个内部复位信号被产生。
 3. 从 **WDTE** 读取的值为 **9AH**（这不同于写入的值 (**ACH**)）。

11.4 看门狗定时器的操作

11.4.1 当“内部振荡器不能被停止”由选项字节选择时的看门狗定时器操作

看门狗定时器的操作时钟固定为内部振荡时钟。

复位释放后，操作以最大周期被启动（看门狗定时器模式寄存器（WDTM）第 2、1 和 0 位（WDCS2，WDCS1，WDCS0）= 1, 1, 1）。看门狗定时器操作不能被停止。

以下显示了复位释放后看门狗定时器的操作。

1. 复位释放后的状态如下所示。
 - 操作时钟：内部振荡时钟
 - 周期： $f_R/2^{18}$ （1.09 秒：以 $f_R = 240$ kHz（典型值）的操作）
 - 计数开始
2. 应该按照下面所示通过 8 位存储器操作指令来设置看门狗定时器（WDTM）^{注 1, 2}。
 - 周期：使用第 2 到 0 位（WDCS2 到 WDCS0）来设置
3. 以上流程执行后，写 ACH 到 WDTM 来清除计数为 0 并允许重新计数。

- 注**
1. 操作时钟（内部振荡时钟）不能被更改。如果任意值被写入 WDTM 的第 3 和 4 位（WDCS3，WDCS4），将被忽略。
 2. 只要 WDTM 被写入，看门狗定时器的计数器就被清除。

注意事项 在该模式下，即使在 STOP 指令执行期间，看门狗定时器的操作也绝对不能被停止。对应 8 位定时器 51（TM51），内部振荡时钟的分频可以被选作计数源，所以在 STOP 指令执行后看门狗定时器溢出之前可以使用 TM51 的中断请求来清除看门狗定时器。如果该处理不被执行，当 STOP 指令执行后看门狗定时器溢出时，一个内部复位信号将被产生。

11.4.2 当“内部振荡器可以由软件停止”由选项字节选择时的看门狗定时器操作

看门狗定时器的操作时钟可以被选择为内部振荡时钟或 X1 输入时钟。

复位释放后，操作以最大周期被启动（看门狗定时器模式寄存器（WDTM）第 2、1 和 0 位（WDCS2, WDCS1, WDCS0） = 1, 1, 1）。

以下显示了复位释放后看门狗定时器的操作。

1. 复位释放后的状态如下所示。
 - 操作时钟：内部振荡时钟（ f_R ）
 - 周期： $f_R/2^{18}$ （1.09 秒：以 $f_R = 240 \text{ kHz}$ （典型值）的操作）
 - 计数开始
2. 应该按照下面所示通过 8 位存储器操作指令来设置看门狗定时器（WDTM）^{注 1, 2, 3}。
 - 操作时钟：使用第 3 和 4 位（WDCS3 和 WDCS4），以下任意一个可以被选择。
 - 内部振荡时钟（ f_R ）
 - X1 输入时钟（ f_{XP} ）
 - 看门狗定时器操作被停止
 - 周期：使用第 2 到 0 位（WDCS2 到 WDCS0）来设置
3. 以上流程执行后，写 ACH 到 WDTE 来清除计数为 0 并允许重新计数。

- 注**
1. 只要 WDTM 被写入，看门狗定时器的计数器就被清除。
 2. 分别设置第 7、6 和 5 位为 0、1、1。不要设置其它值。
 3. 如果通过设置 WDCS4 和 WDCS3 分别为 1 和 x 看门狗定时器被停止，即使以下处理被执行，内部复位信号也不会被产生。
 - WDTM 被写入两次。
 - 对 WDTE 执行 1 位存储器操作指令。
 - ACH 以外的值被写入 WDTE

注意事项 在该模式下，看门狗定时器操作在 HALT/STOP 指令执行期间被停止。在 HALT/STOP 模式被释放后，使用 HALT/STOP 指令执行之前通过 WDTM 设置的看门狗定时器操作时钟的计数再次开始。这时，计数器不被清零而是保持其值。

关于 STOP 模式和 HALT 模式期间看门狗定时器的操作的各种状态，参见 11.4.3 STOP 模式下看门狗定时器的操作和 11.4.4 HALT 模式下看门狗定时器的操作。

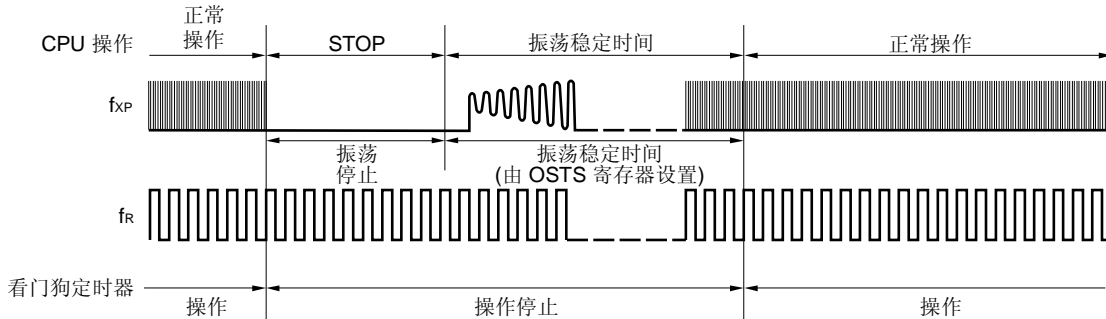
11.4.3 STOP模式下看门狗定时器的操作（当“内部振荡器可以由软件停止”由选项字节选择时）

在 STOP 指令执行期间，看门狗定时器停止计数，无论 X1 输入时钟或内部振荡时钟是否被使用。

(1) STOP 指令执行时，当 CPU 时钟和看门狗定时器操作时钟是 X1 输入时钟 (f_{XP}) 时

当 STOP 指令被执行时，看门狗定时器的操作被停止。STOP 模式被释放后，计数停止将持续由振荡稳定时间选择寄存器 (OSTS) 设置的振荡稳定时间，然后计数将使用操作停止前的操作时钟再次开始计数。这时，计数器不被清零而是保持其值。

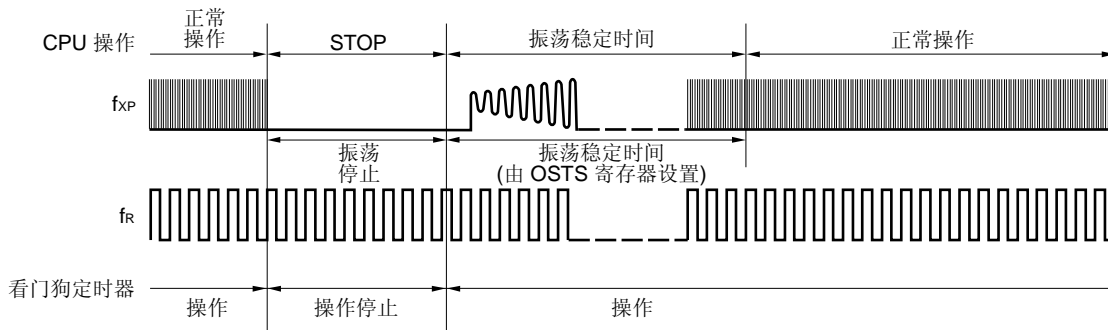
图 11-4. STOP 模式下的操作（CPU 时钟和 WDT 操作时钟：X1 输入时钟）



(2) STOP 指令执行时，当 CPU 时钟是 X1 输入时钟 (f_{XP}) 而看门狗定时器操作时钟是内部振荡时钟 (f_R) 时

当 STOP 指令被执行时，看门狗定时器的操作被停止。STOP 模式被释放后，计数使用操作停止前的操作时钟再次开始计数。这时，计数器不被清零而是保持其值。

图 11-5. STOP 模式下的操作
(CPU 时钟：X1 输入时钟，WDT 操作时钟：内部振荡时钟)



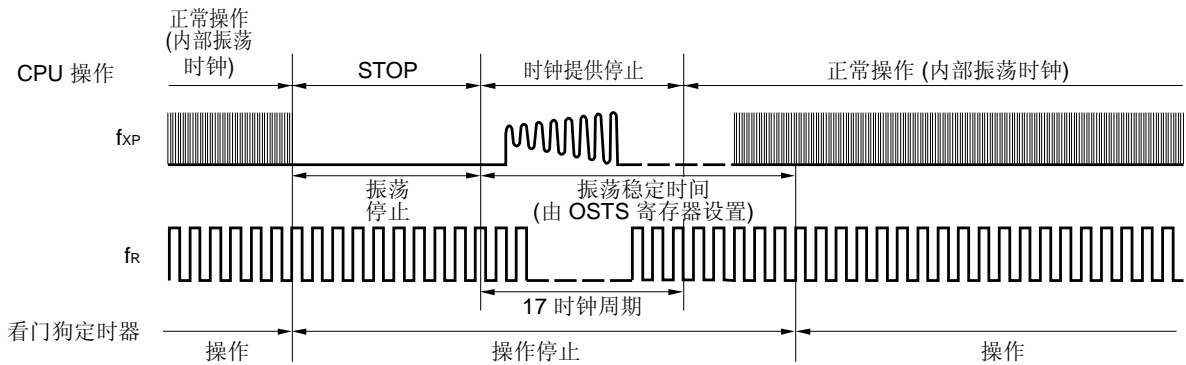
(3) STOP 指令执行时，当 CPU 时钟是内部振荡时钟 (f_R) 而看门狗定时器操作时钟是 X1 输入时钟 (f_{XP}) 时

当 STOP 指令被执行时，看门狗定时器的操作被停止。STOP 模式释放后，在<1> 或 <2>的时序（不管哪个在前）之前计数被停止，然后计数使用操作停止前的操作时钟再次开始计数。这时，计数器不被清零而是保持其值。

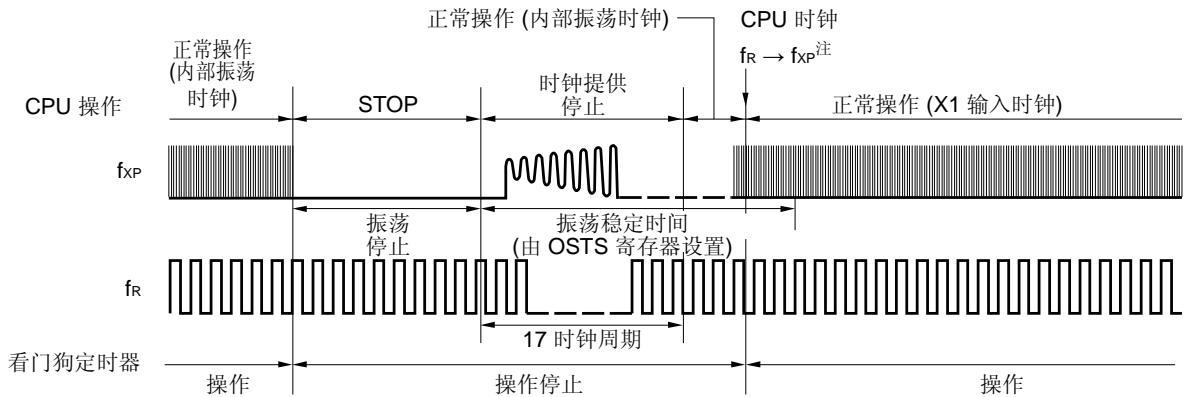
- <1> 由振荡稳定时间选择寄存器 (OSTS) 设置的振荡稳定时间过去。
- <2> CPU 时钟切换为 X1 输入时钟 (f_{XP})。

图 11-6. STOP 模式下的操作
(CPU 时钟: 内部振荡时钟, WDT 操作时钟: X1 输入时钟)

- <1> 由振荡稳定时间选择寄存器 (OSTS) 设置的振荡稳定时间过去后计数再次开始的时序



- <2> CPU 时钟切换为 X1 输入时钟 (f_{XP}) 后计数开始的时序

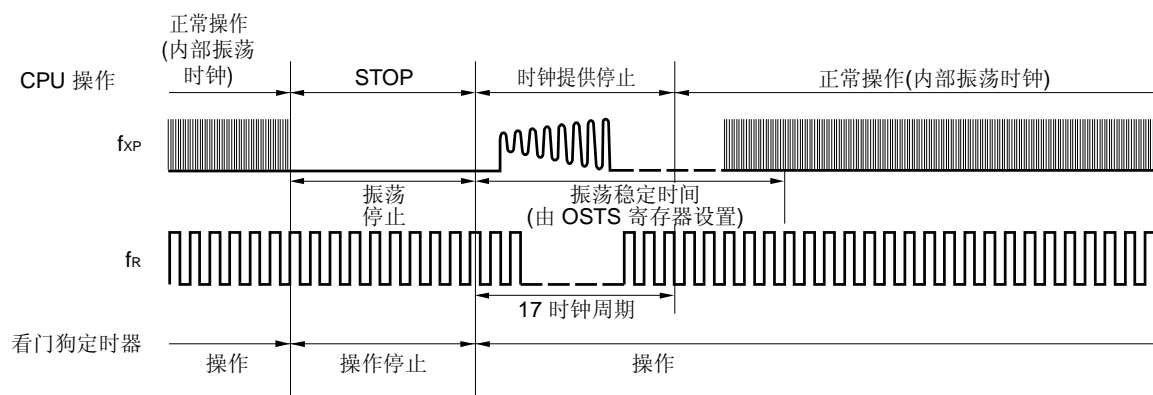


注 使用振荡稳定时间计数器状态寄存器 (OSTC) 来确认 f_{XP} 的振荡稳定时间。

(4) STOP 指令执行期间，当 CPU 时钟和看门狗定时器操作时钟是内部振荡时钟 (f_R) 时

当 STOP 指令被执行时，看门狗定时器的操作被停止。STOP 模式被释放后，计数使用操作停止前的操作时钟再次开始计数。这时，计数器不被清零而是保持其值。

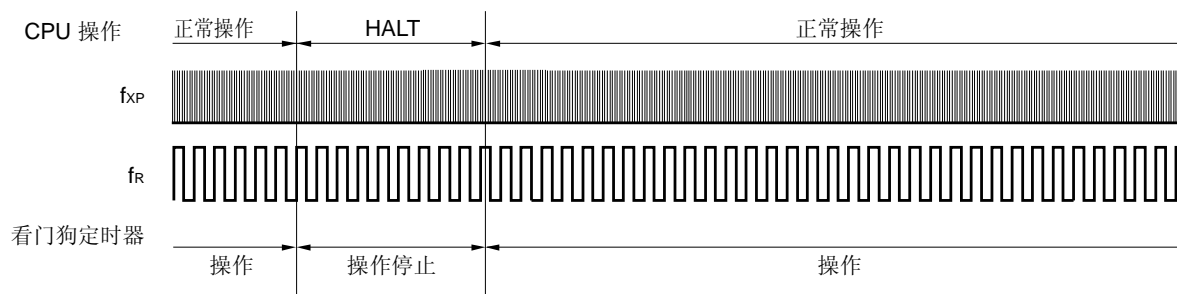
图 11-7. STOP 模式下的操作 (CPU 时钟和 WDT 操作时钟: 内部振荡时钟)



11.4.4 HALT 模式下看门狗定时器的操作 (当“内部振荡器可以由软件停止”由选项字节选择时)

在 HALT 指令执行期间，无论 CPU 时钟是 X1 输入时钟 (f_{XP}) 还是内部振荡时钟 (f_R)，无论看门狗定时器操作时钟是 X1 输入时钟 (f_{XP}) 还是内部振荡时钟 (f_R)，看门狗定时器都停止计数。HALT 模式被释放后，计数使用操作停止前的操作时钟再次开始计数。这时，计数器不被清零而是保持其值。

图 11-8. HALT 模式下的操作



第十二章 时钟输出/蜂鸣器输出控制器

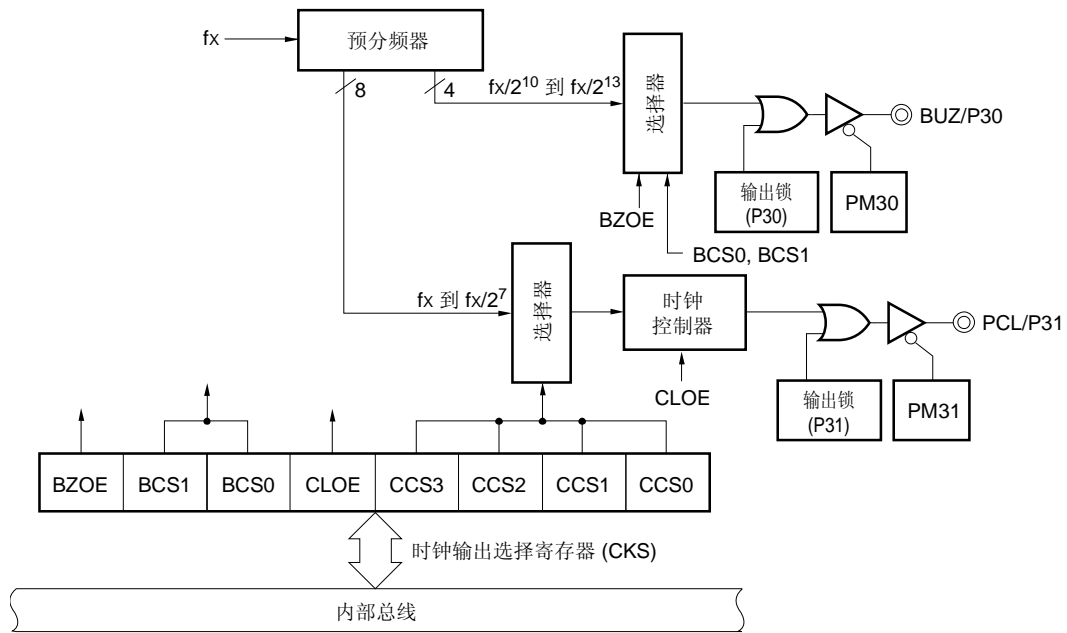
12.1 时钟输出/蜂鸣器输出控制器的功能

时钟输出控制器用于远程控制发送过程中的载波输出和提供给周边 LSI 的时钟输出。时钟选择寄存器 (CKS) 选择的时钟被输出。

此外, 蜂鸣器输出是指 CKS 选择的蜂鸣器频率的方波输出。

图 12-1 表示时钟输出/蜂鸣器输出控制器的框图。

图 12-1. 时钟输出/蜂鸣器输出控制器的框图



12.2 时钟输出/蜂鸣器输出控制器的配置

时钟输出/蜂鸣器输出控制器由以下硬件组成。

表 12-1. 时钟输出/蜂鸣器输出控制器的配置

项目	配置
控制寄存器	时钟输出选择寄存器 (CKS) 端口模式寄存器 3 (PM3) 端口寄存器 3 (P3)

12.3 控制时钟输出/蜂鸣器输出控制器的寄存器

以下三个寄存器被用来控制时钟输出/蜂鸣器输出控制器。

- 时钟输出选择寄存器 (CKS)
- 端口模式寄存器 3 (PM3)
- 端口寄存器 3 (P3)

(1) 时钟输出选择寄存器 (CKS)

该寄存器设置时钟输出 (PCL) 和蜂鸣器频率输出 (BUZ) 的输出允许/禁止以及设置输出时钟。

CKS 通过一个 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除 CKS 为 00H。

图 12-2. 时钟输出选择寄存器 (CKS) 的格式

地址: FF40H 复位后: 00H R/W

符号	<7>	6	5	<4>	3	2	1	0
CKS	BZOE	BCS1	BCS0	CLOE	CCS3	CCS2	CCS1	CCS0

BZOE	BUZ 输出允许/禁止指定
0	时钟分频电路操作被停止。BUZ 固定为低电平。
1	时钟分频电路操作被允许。BUZ 输出被允许。

BCS1	BCS0	BUZ 输出时钟选择
0	0	$f_x/2^{10}$ (19.5 kHz)
0	1	$f_x/2^{11}$ (9.77 kHz)
1	0	$f_x/2^{12}$ (4.88 kHz)
1	1	$f_x/2^{13}$ (2.44 kHz)

CLOE	PCL 输出允许/禁止指定
0	时钟分频电路操作被停止。PCL 固定为低电平。
1	时钟分频电路操作被允许。PCL 输出被允许。

CCS3	CCS2	CCS1	CCS0	PCL 输出时钟选择
0	0	0	0	f_x (20 MHz)
0	0	0	1	$f_x/2$ (10 MHz)
0	0	1	0	$f_x/2^2$ (5 MHz)
0	0	1	1	$f_x/2^3$ (2.5 MHz)
0	1	0	0	$f_x/2^4$ (1.25 MHz)
0	1	0	1	$f_x/2^5$ (625 kHz)
0	1	1	0	$f_x/2^6$ (312.5 kHz)
0	1	1	1	$f_x/2^7$ (156.25 kHz)
除上面以外				禁止设置

- 备注
1. f_x : X1 输入时钟振荡频率
 2. 括号中的数字用于 $f_x = 20$ MHz 的操作。

(2) 端口模式寄存器 3 (PM3)

该寄存器以 1 位为单位设置端口 3 输入/输出。

当使用 P31/PCL 引脚用于时钟输出和使用 P30/BUZ 引脚用于蜂鸣器输出时，清除 PM31、PM30 和 P31、P30 的输出锁存为 0。

PM3 通过一个 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入设置 PM3 为 FFH。

图 12-3. 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 管脚输入/输出模式选择 (n = 0 到 3)
0	输出模式 (输出缓存打开)
1	输入模式 (输出缓存关闭)

12.4 时钟输出/蜂鸣器输出控制器的操作

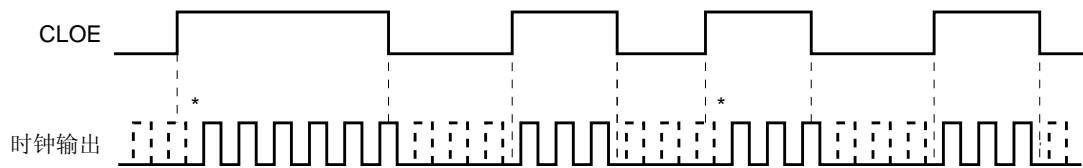
12.4.1 时钟输出操作

时钟脉冲按照以下过程被输出。

- <1> 使用时钟输出选择寄存器 (CKS) 的第 0 到 3 位 (CCS0 到 CCS3) 来选择时钟脉冲输出频率 (时钟脉冲输出处于禁止状态)。
- <2> 设置 CKS 的第 4 位 (CLOE) 为 1 来允许时钟输出。

备注 时钟输出控制器被设计为在时钟输出的输出允许/禁止切换期间不输出窄脉冲。如图 12-4 所示, 确认从时钟的低电平周期 (图中用*标记的) 启动输出。当停止输出时, 保护时钟的高电平后也这样做。

图 12-4. 远程控制输出应用举例



12.4.2 作为蜂鸣器输出的操作

蜂鸣器频率按照以下过程被输出。

- <1> 使用时钟输出选择寄存器 (CKS) 的第 5 和 6 位 (BCS0, BCS1) 来选择蜂鸣器输出频率 (蜂鸣器输出处于禁止状态)。
- <2> 设置 CKS 的第 7 位 (BZOE) 为 1 来允许蜂鸣器输出。

第十三章 实时输出端口

13.1 实时输出端口的功能

先前设置在实时输出缓存寄存器中的数据可以通过硬件与定时器中断或外部中断请求同步传输到输出锁中，然后向外部输出。这叫做实时输出功能。向外部输出数据的引脚叫做实时输出端口。

通过使用实时输出端口，可以输出一个没有抖动的信号。因此，这特别适合任意模式以任意间隔输出的应用（步进电机的开环控制，等等）。

同时，也可以在指定的引脚对输出模式执行 PWM 调制。

μ PD78F0714 片上包含以下 2 通道实时输出端口。可以以 1 位为单位指定实时输出端口。

- 8 位 \times 1 或 4 位 \times 2 ... 实时输出端口 0
- 6 位 \times 1 或 4 位 \times 1 ... 实时输出端口 1

13.2 实时输出端口的配置

实时输出端口包含以下硬件。

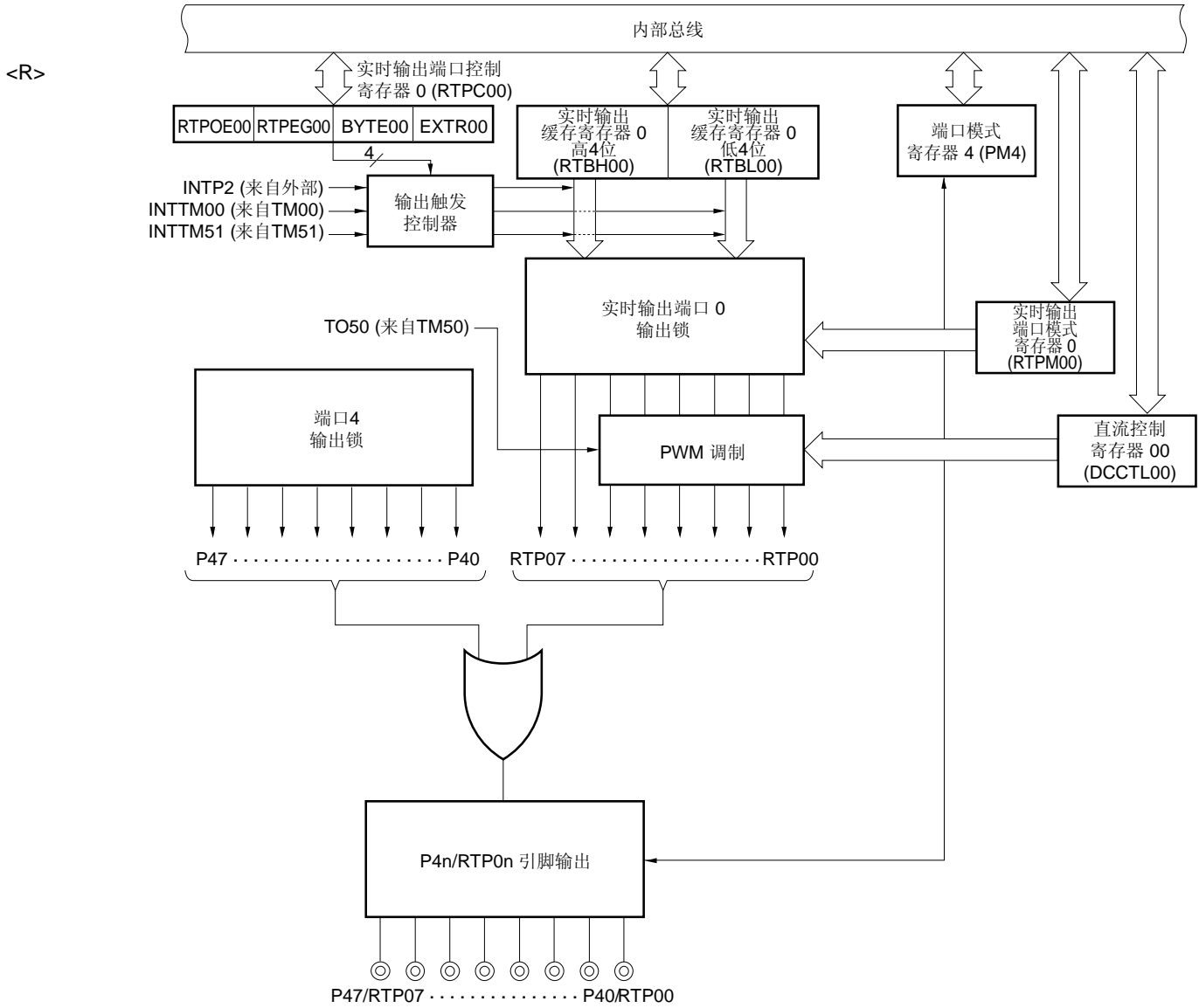
表 13-1. 实时输出端口的配置

项目	配置
寄存器	实时输出缓存寄存器 n (RTBL0n, RTBH0n)
控制寄存器	端口模式寄存器 4 (PM4) 实时输出端口模式寄存器 n (RTPM0n) 实时输出端口控制寄存器 n (RTPC0n) 直流控制寄存器 0n (DCCTL0n)

n = 0, 1.

图 13-1. 实时输出端口的框图 (1/2)

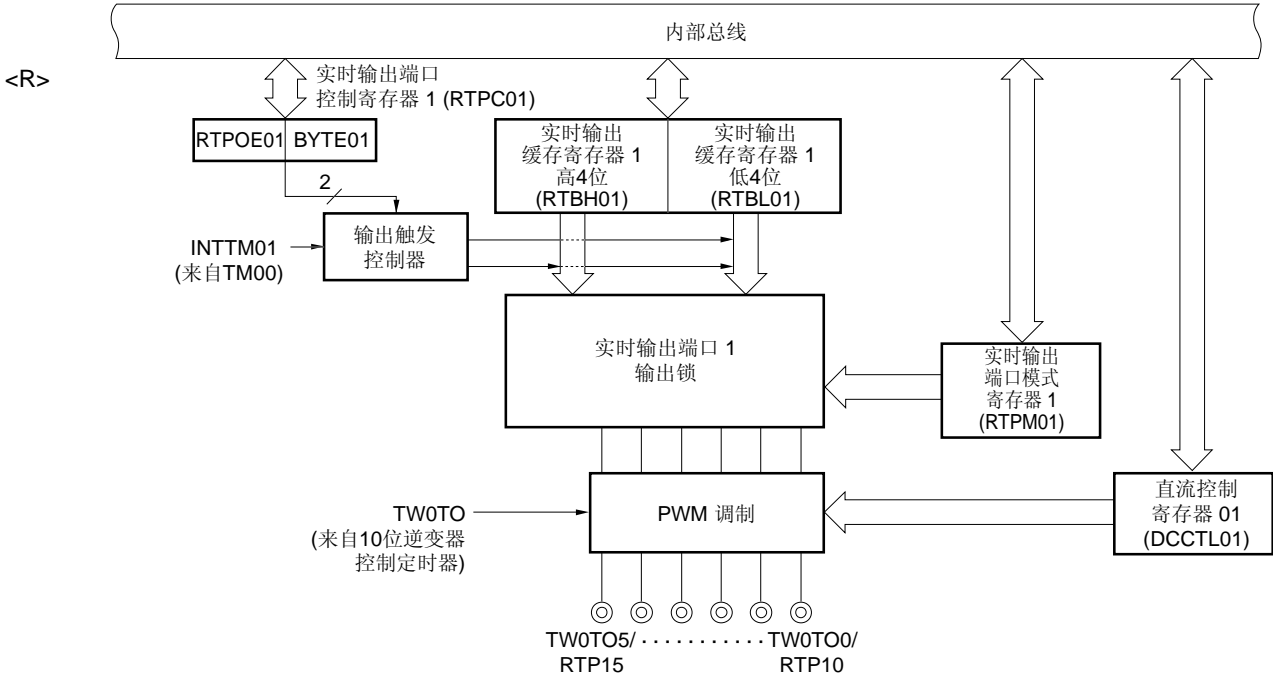
(a) 实时输出端口 0 (8 位 × 1 或 4 位 × 2)



备注 n = 0 到 7

图 13-1. 实时输出端口的框图 (2/2)

(b) 实时输出端口 1 (6 位 × 1 或 4 位 × 1)



备注 n = 0 到 5

(1) 实时输出缓存寄存器 0 (RTBL00, RTBH00)

该寄存器由事先保持输出数据的两个 4 位寄存器组成。

RTBL00 和 RTBH00 的地址单独映射到特殊功能寄存器 (SFR) 区域, 如图 13-2 所示。

当指定 4 位 × 2 通道作为操作模式时, 数据被单独设置到 RTBL00 和 RTBH00。

无论哪一个地址被指定, RTBL00 和 RTBH00 的数据可以一次一起被读出。

当指定 8 位 × 1 通道作为操作模式时, 通过写 8 位数据到 RTBL00 或 RTBH00 来设置到 RTBL00 和 RTBH00。

无论哪一个地址被指定, RTBL00 和 RTBH00 的数据可以一次一起被读出。

图 13-2 表示 RTBL00 和 RTBH00 的配置, 表 13-2 表示 RTBL00 和 RTBH00 修改期间的操作。

图 13-2. 实时输出缓存寄存器 0 的配置

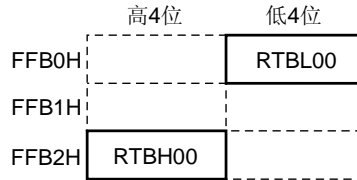


表 13-2. 实时输出缓存寄存器 0 修改期间的操作

<R>

操作模式	要修改的寄存器	读取		写入 ^注	
		高 4 位	低 4 位	高 4 位	低 4 位
4 位 × 2 通道	RTBL00	RTBH00	RTBL00	无效	RTBL00
	RTBH00	RTBH00	RTBL00	RTBH00	无效
8 位 × 1 通道	RTBL00	RTBH00	RTBL00	RTBH00	RTBL00
	RTBH00	RTBH00	RTBL00	RTBH00	RTBL00

注 在实时输出端口中设置数据后, 到实时输出触发被产生时, 输出数据应该被设置在 RTBL00 和 RTBH00 中。

(2) 实时输出缓存寄存器 1 (RTBL01, RTBH01)

该寄存器由事先保持输出数据的两个 4 位寄存器^注组成。

RTBL01 和 RTBH01 的地址单独映射到特殊功能寄存器 (SFR) 区域, 如图 13-3 所示。

当指定 4 位 × 1 通道作为操作模式时, 数据被设置到 RTBL01。

当指定 6 位 × 1 通道作为操作模式时, 通过写 6 位数据到 RTBL01 或 RTBH01, 数据被设置到 RTBL01 和 RTBH01。无论哪一个地址被指定, RTBL01 和 RTBH01 的数据可以一次一起被读出。

图 13-3 表示 RTBL01 和 RTBH01 的配置, 表 13-3 表示 RTBL01 和 RTBH01 修改期间的操作。

注 对于 RTBH01, 只有 4 位中的两位有效。

图 13-3. 实时输出缓存寄存器 1 的配置

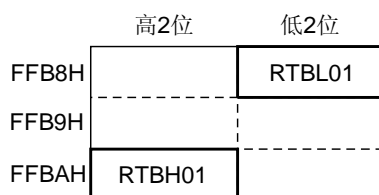


表 13-3. 实时输出缓存寄存器 1 修改期间的操作

<R>	操作模式	要修改的寄存器	读取		写入 ^注	
			高 2 位	低 4 位	高 2 位	低 4 位
	4 位 × 1 通道	RTBL01	无效	RTBL01	无效	RTBL01
	6 位 × 1 通道	RTBL01	RTBH01	RTBL01	RTBH01	RTBL01
		RTBH01	RTBH01	RTBL01	RTBH01	RTBL01

注 在实时输出端口中设置数据后, 到实时输出触发被产生时, 输出数据应该被设置在 RTBL01 和 RTBH01 中。

13.3 控制实时输出端口的寄存器

以下七类寄存器控制实时输出端口。

- 端口模式寄存器 4 (PM4)
- 实时输出端口模式寄存器 0, 1 (RTPM00, RTPM01)
- 实时输出端口控制寄存器 0, 1 (RTPC00, RTPC01)
- 直流控制寄存器 00, 01 (DCCTL00, DCCTL01)

(1) 端口模式寄存器 4 (PM4)

该寄存器设置替换用作实时输出引脚 (RTP00 到 RTP07) 的端口 4 引脚 (P40 到 P47) 的输入/输出模式。要使用端口 4 作为实时输出端口, 端口引脚的输入/输出模式必须被设置为输出模式 (PM4n = 0: n = 0 到 7)。

PM4 由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入设置该寄存器为 FFH。

图 13-4. 端口模式寄存器 4 的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W	
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FF24H	FFH	R/W	
PM4n	P4n 引脚 I/O 模式选择 (n = 0 到 7)											
	0	输出模式 (输出缓存打开)										
	1	输入模式 (输出缓存关闭)										

(2) 实时输出端口模式寄存器 0 (RTPM00)

该寄存器以 1 位为单位设置实时输出端口模式或端口模式。

要被设置的输出为 RTP00 到 RTP07。

RTPM00 由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 13-5. 实时输出端口模式寄存器 0 的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W	
RTPM00	RTPM07	RTPM06	RTPM05	RTPM04	RTPM03	RTPM02	RTPM01	RTPM00	FFB4H	00H	R/W	
RTPM00n	实时输出端口选择 (n = 0 到 7)											
	0	端口模式										
	1	实时输出端口模式										

注意事项 当使用一个端口作为实时输出端口时, 设置端口为输出模式 (通过清除端口模式寄存器 4 (PM4) 的对应位为 0)。

RTPM007 RTPM006 RTPM005 RTPM004 RTPM002 RTPM001 RTPM000 RTPM00 RTPM00n 实时输出端口选择 (n = 0 到 7)

(3) 实时输出端口模式寄存器 1 (RTPM01)

该寄存器用来以 1 位为单位事先设置实时输出端口模式。

要被设置的输出为 RTP10 到 RTP15。

RTPM01 由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 13-6. 实时输出端口模式寄存器 1 的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
RTPM01	0	0	RTPM015	RTPM014	RTPM013	RTPM012	RTPM011	RTPM010	FFBCH	00H	R/W

RTPM01n	实时输出端口选择 (n = 0 到 5)
0	实时输出缓存被禁止。
1	实时输出缓存被允许

<R>

<R>

注意事项 确保设置 RTPM01 的第 6 和 7 位为 0。

备注 当用作实时输出端口时，RTP10 到 RTP15 变为输出。

(4) 实时输出端口控制寄存器 0 (RTPC00)

该寄存器用来设置实时输出端口的操作模式、输出触发和操作允许/禁止。

要被设置的输出为 RTP00 到 RTP07。

实时输出端口的操作模式和输出触发之间的关系如表 13-4 所示。

RTPC00 由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 13-7. 实时输出端口控制寄存器 0 的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
RTPC00	RTPOE00	RTPEG00	BYTE00	EXTR00	0	0	0	0	FFB5H	00H	R/W

RTPOE00	实时输出端口操作控制
0	禁止操作 ^注
1	允许操作

RTPEG00	INTP2 有效沿指定
0	下降沿
1	上升沿

BYTE00	实时输出端口操作模式
0	4 位 × 2 通道
1	8 位 × 1 通道

EXTR00	由 INTP2 控制实时输出
0	INTP2 不用作实时输出触发。
1	INTP2 用作实时输出触发。

注 当 RTPM00n (实时输出端口模式寄存器 0 (RTPM00) 的第 n (n = 0 到 7) 位) 为 1 时, INV00 (直流控制寄存器 00 (DCCTL00) 的第 4 位) 为 0, 并且实时输出操作被禁止 (RTPOE00 = 0), RTP00 到 RTP07 输出 “0”。

表 13-4. 实时输出端口操作模式和输出触发

BYTE00	EXTR00	操作模式	RTBH00 → 端口输出	RTBL00 → 端口输出
0	0	4 位 × 2 通道	INTTM51	INTTM00
0	1		INTTM00	INTP2
1	0	8 位 × 1 通道	INTTM00	
1	1		INTP2	

(5) 实时输出端口控制寄存器 1 (RTPC01)

该寄存器用来设置实时输出端口的操作模式和操作允许或禁止。

要被设置的输出为 RTP10 到 RTP15。

实时输出端口的操作模式和输出触发之间的关系如表 13-5 所示。

RTPC01 由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 13-8. 实时输出端口控制寄存器 1 的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
RTPC01	RTPOE01	0	BYTE01	0	0	0	0	0	FFBDH	00H	R/W

RTPOE01	实时输出端口操作控制
0	禁止操作 ^注
1	允许操作

BYTE01	实时输出端口操作模式
0	4 位 × 1 通道
1	6 位 × 1 通道

注 当 RTPM01n (实时输出端口模式寄存器 1 (RTPM01) 的第 n (n = 0 到 5) 位) 为 1 时, INV01 (直流控制寄存器 00 (DCCTL01) 的第 4 位) 为 0, 并且实时输出操作被禁止 (RTPOE01 = 0), RTP10 到 RTP15 输出“0”。

表 13-5. 实时输出端口操作模式和输出触发

BYTE01	操作模式	RTBH01 → 端口输出	RTBL01 → 端口输出
0	4 位 × 1 通道	–	INTTM01
1	6 位 × 1 通道	INTTM01	

(6) 直流控制寄存器 00 (DCCTL00)

该寄存器用来允许/禁止 PWM 调制和允许/禁止实时输出口的的输出波形的反转。

要被设置的输出为 RTP00 到 RTP07。

DCCTL00 由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 13-9. 直流控制寄存器 00 的格式

地址: FF28H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
DCCTL00	DCEN00	PWMCH00	PWMCL00	INV00	0	0	0	0

DCEN00	输出操作指定
0	RTP 输出
1	PWM 调制的 RTP 输出 ^注

PWMCH00	PWM 调制指定 (RTP00, RTP02, RTP04 输出指定)
0	PWM 调制被禁止
1	PWM 调制被允许

PWMCL00	PWM 调制指定 (RTP01, RTP03, RTP05 输出指定)
0	PWM 调制被禁止
1	PWM 调制被允许

INV00	输出波形指定
0	反转被禁止
1	反转被允许

注 PWM 信号使用 TO50 输出。

备注 1. 要被设置的输出为 RTP00 到 RTP07。
 2. PWMCH00、PWMCL00 和 INV00 设置只有在 DCEN00 = 1 时才有效。

(7) 直流控制寄存器 01 (DCCTL01)

该寄存器用来允许/禁止 PWM 调制和允许/禁止实时输出端口的输出波形的反转。

要被设置的输出为 RTP10 到 RTP15。

DCCTL01 由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 13-10. 直流控制寄存器 01 的格式

地址: FF38H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
DCCTL01	DCEN01	PWMCH01	PWMCL01	INV01	0	0	0	0

DCEN01	输出操作指定
0	逆变器定时器输出 (RTP10 到 RTP15)
1	PWM 调制的 RTP 输出 ^注

PWMCH01	PWM 调制指定 (RTP10, RTP12, RTP14 输出指定)
0	PWM 调制被禁止
1	PWM 调制被允许

PWMCL01	PWM 调制指定 (RTP11, RTP13, RTP15 输出指定)
0	PWM 调制被禁止
1	PWM 调制被允许

INV01	输出波形指定
0	反转被禁止
1	反转被允许

注 PWM 信号使用逆变器定时器输出 (TW0TO0 到 TW0TO5)。

备注

1. 要被设置的输出为 RTP10 到 RTP15。
2. PWMCH01、PWMCL01 和 INV01 设置只有在 DCEN01 = 1 时才有效。

13.4 实时输出端口的操作

(1) 使用 RTP00 到 RTP07 作为实时输出端口 实时输出端口 0

(8 位 × 1 或 4 位 × 2)

当实时输出端口控制寄存器 0 (RTPC00) 的第 7 位 (RTPOE00) 为 1, 实时输出操作被允许时, 实时输出缓存寄存器 0 (RTBH00, RTBL00) 中的数据与选择的传输触发 (由 EXTR00 和 BYTE00 设置) 同步传输到输出锁存中。在传输的数据中, 只有通过设置实时输出端口模式寄存器 0 (RTPM00) 指定为实时输出端口的位的数据从位 RTP00 到 RTP07 输出。通过 RTPM00 指定为端口模式的端口可以用作通用输入/输出端口。

通过设置 EXTR00 和 BYTE00, 操作模式可以选择为 8 位 × 1 或 4 位 × 2。通过设置 INV00, 可以反转输出波形。同时, 通过设置 PWMCL00 和 PWMCH00, 也可以执行输出模式的 PWM 调制。

当 RTPM00n = 1 并且 INV00 = 0 时, 如果实时输出被禁止 (RTPOE00 = 0), 然后 RTP00 到 RTP07 输出 0。

控制寄存器每一位的设置和实时输出之间的关系如表 13-6 所示, 操作时序的示例如图 13-11 所示。

备注	EXTR00:	实时输出端口控制寄存器 0 (RTPC00) 的第 4 位
	BYTE00:	实时输出端口控制寄存器 0 (RTPC00) 的第 5 位
	INV00:	直流控制寄存器 00 (DCCTL00) 的第 4 位
	PWMCL00:	直流控制寄存器 00 (DCCTL00) 的第 5 位
	PWMCH00:	直流控制寄存器 00 (DCCTL00) 的第 6 位
	RTPM00n:	实时输出端口模式寄存器 0 (RTPM00) 的第 n (n = 0 到 7) 位

表 13-6. 控制寄存器每一位的设置与实时输出之间的关系

PM4n	P4n	DCEN00	INV00	PWMCH00/ PWMCL00	RTPOE00	RTPM00n	RTBH00m/ RTBL00m	引脚 P4n 状态										
1	x	x	x	x	x	x	x	输入端口										
0	1	x	x	x	x	x	x	“高”输出										
								“低”输出										
	0	0	0	x	x	0	x	x	“低”输出									
									1	0	x	“低”输出						
						1	0	x	1	0	x	1	“低”输出					
													1	0	x	“高”输出		
						1	0	0	0	0	0	x	x	“低”输出				
														1	0	x	“低”输出	
											1	0	x	1	0	x	1	“低”输出
																		1
		1	0	0	1						0	x	x	“TO50”输出				
														1	0	x	“TO50”输出	
		1	0	x	1	0	x	1	“TO50”输出									
									1	0	x	“高”输出						
		1	1	0	0	0	0	x	x	“高”输出								
										1	0	x	“高”输出					
1	0						x	1	0	x	1	“高”输出						
												1	0	x	“低”输出			
1	0						x	1	0	x	x	“TO50”输出						
												1	0	x	“TO50”输出			
1	0	x	1	0	x	1	“TO50”输出											
							1	0	x	“低”输出								

PM4n: 端口模式寄存器 4 (PM4) 的第 n 位

P4n: 端口 4 (P4) 的第 n 位

DCEN00: 直流控制寄存器 00 (DCCTL00) 的第 7 位

INV00: DCCTL00 的第 4 位

PWMCH00: DCCTL00 的第 6 位

PWMCL00: DCCTL00 的第 5 位

RTPOE00: 实时输出端口控制寄存器 0 (RTPC00) 的第 7 位

RTPM00n: 实时输出端口模式寄存器 0 (RTPM00) 的第 n 位

RTBH00m: 实时输出缓存寄存器 0H (RTBH00) 的第 m 位

RTBL00m: 实时输出缓存寄存器 0L (RTBL00) 的第 m 位

n = 0 到 7

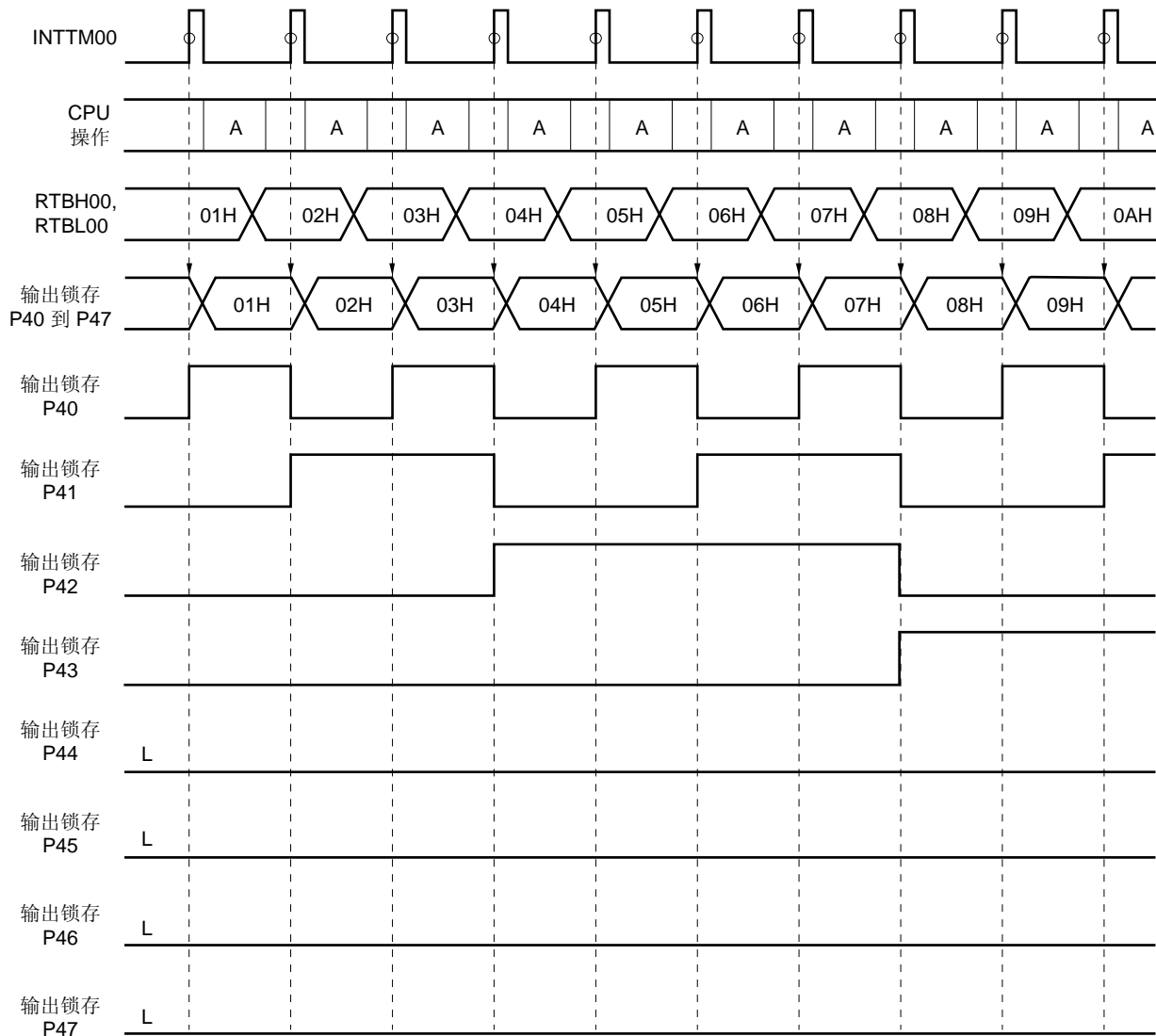
m = 0 到 3

x: 不关注。

图 13-11. 实时输出端口操作时序示例 (8 位 × 1) (1/3)

(a) 8 位 × 1 通道, 反转输出被禁止, 无 PWM 调制

(EXTR00 = 0, BYTE00 = 1, INV00 = 0, PWMCH00 = 0, PWMCL00 = 0)

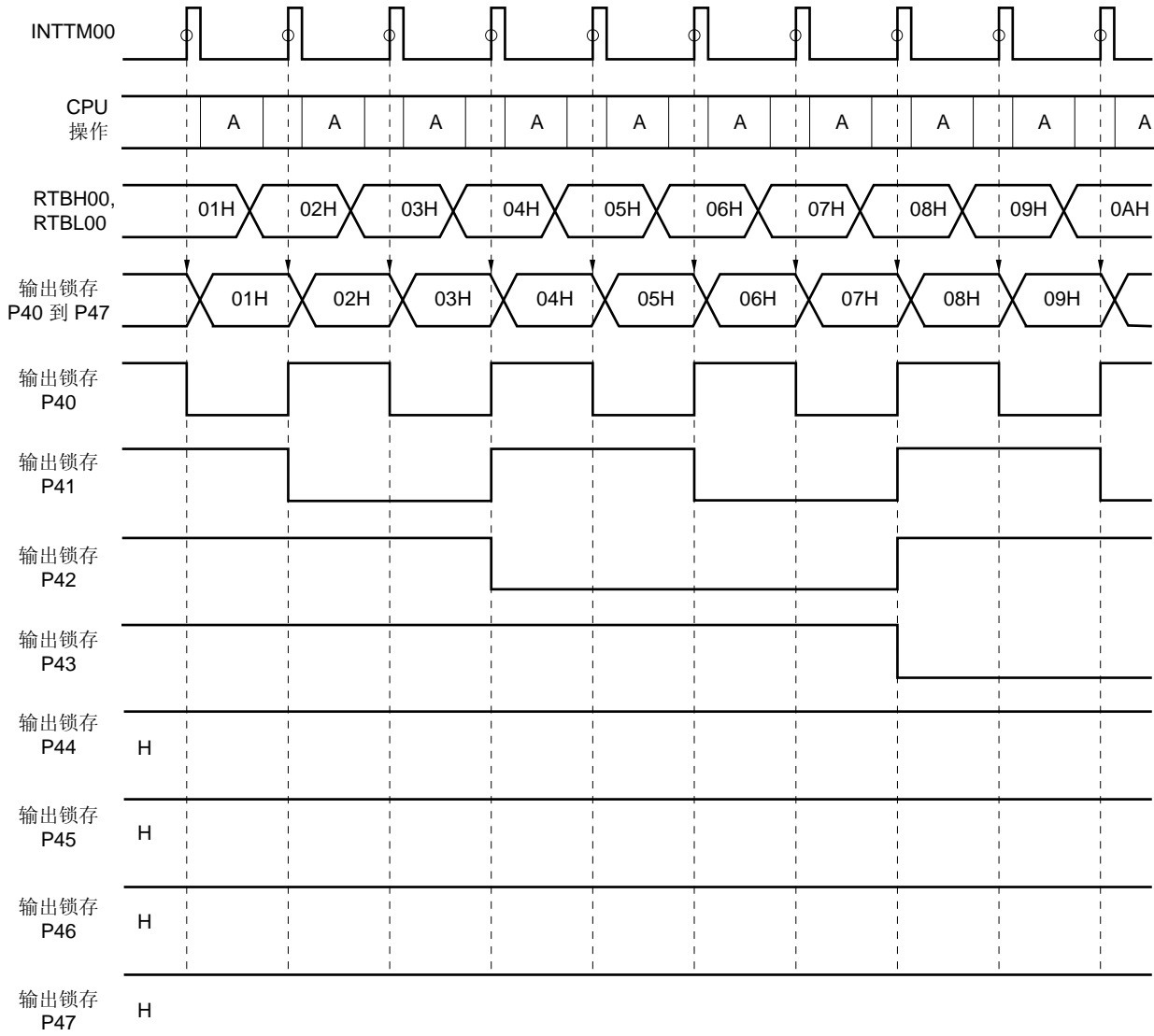


A: INTTM00 软件处理 (RTBH00, RTBL00 写)

图 13-11. 实时输出端口操作时序示例 (8 位 × 1) (2/3)

(b) 8 位 × 1 通道, 反转输出被允许, 无 PWM 调制

(EXTR00 = 0, BYTE00 = 1, INV00 = 1, PWMCH00 = 0, PWMCL00 = 0)

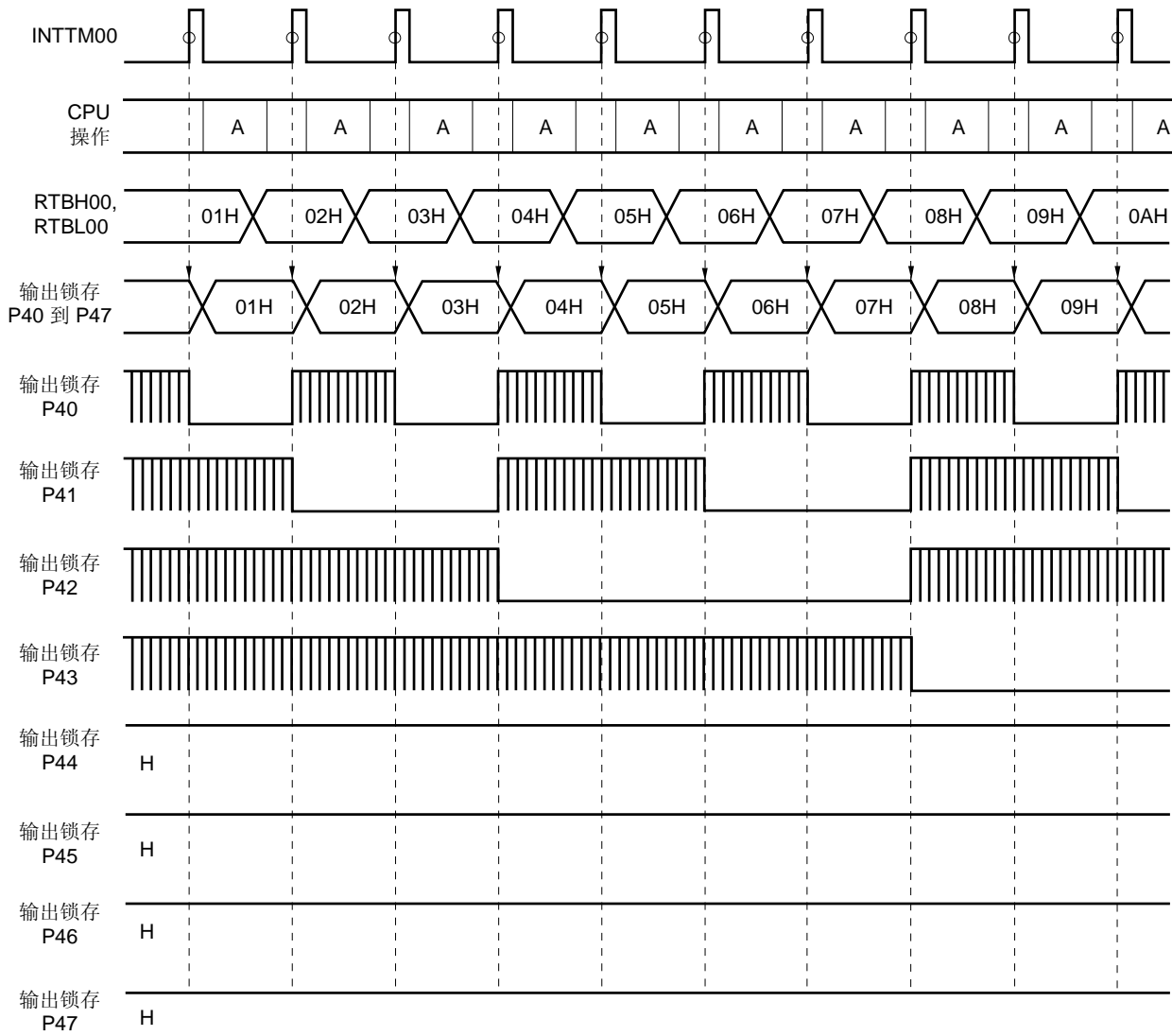


A: INTTM00 软件处理 (RTBH00, RTBL00 写)

图 13-11. 实时输出端口操作时序示例 (8 位 × 1) (3/3)

(c) 8 位 × 1 通道, 反转输出被允许, PWM 调制

(EXTR00 = 0, BYTE00 = 1, INV00 = 1, PWMCH00 = 1, PWMCL00 = 1)



A: INTTM00 软件处理 (RTBH00, RTBL00 写)

(2) 使用 RTP10 到 RTP15 作为实时输出端口 实时输出端口 1**(6 位 × 1 或 4 位 × 1)**

当实时输出端口控制寄存器 1 (RTPC01) 的第 7 位 (RTPOE01) 为 1, 实时输出操作被允许时, 实时输出缓存寄存器 1 (RTBH01, RTBL01) 中的数据与 INTTM01 同步传输到输出锁存中。在传输的数据中, 只有通过设置实时输出端口模式寄存器 1 (RTPM01) 指定为实时输出端口的位的数据从位 RTP10 到 RTP15 输出。当逆变器定时器输出由 DCEN01 指定时, 可以使用 RTP10 到 RTP15 作为逆变器定时器输出。

通过设置 BYTE01, 操作模式可以选择为 6 位 × 1 或 4 位 × 1。

通过设置 INV01, 可以反转输出波形。同时, 通过设置 PWMCL01 和 PWMCH01, 也可以执行输出模式的 PWM 调制。

当 RTPM01n = 1 并且 INV01 = 0 时, 如果实时输出被禁止 (RTPOE01 = 0), 然后 RTP10 到 RTP15 输出 0。

控制寄存器每一位的设置和实时输出之间的关系如表 13-7 所示, 操作时序的示例如图 13-12 所示。

备注	BYTE01:	实时输出端口控制寄存器 1 (RTPC01) 的第 5 位
	DCEN01:	直流控制寄存器 1 (DCCTL1) 的第 7 位
	INV01:	直流控制寄存器 1 (DCCTL1) 的第 4 位
	PWMCL01:	直流控制寄存器 1 (DCCTL1) 的第 5 位
	PWMCH01:	直流控制寄存器 1 (DCCTL1) 的第 6 位
	RTPM01n:	实时输出端口模式寄存器 1 (RTPM01) 的第 n (n = 0 到 5) 位

<R>

表 13-7. 控制寄存器每一位的设置与实时输出之间的关系

CE0	DCEN01	INV01	PWMCH01/ PWMCL01	RTPOE01	RTPM01n	RTBH01m/ RTBL01m	引脚 TW0TOn 状态			
0	x	x	x	x	x	x	高阻			
1	0	x	x	x	x	x	TW0TOn			
				1	0	x	“低”输出			
	1	1	0	0	0	x	x	“低”输出		
					1	0	x	“低”输出		
					1	0	x	“低”输出		
					1	1	x	“高”输出		
				1	0	x	x	TW0T00		
					1	0	x	TW0T00		
					1	0	x	TW0T00		
					1	1	x	“高”输出		
			1	1	1	0	0	x	x	“高”输出
							1	0	x	“高”输出
						1	0	x	x	TW0T00
							1	0	x	TW0T00
							1	0	x	TW0T00
							1	1	x	“低”输出

CE0: 逆变器定时器控制寄存器 (TW0C) 的第 7 位

DCEN01: 直流控制寄存器 01 (DCCTL01) 的第 7 位

INV01: DCCTL01 的第 4 位

PWMCH01: DCCTL01 的第 6 位

PWMCL01: DCCTL01 的第 5 位

RTPOE01: 实时输出端口控制寄存器 1 (RTPC01) 的第 7 位

RTPM01n: 实时输出端口模式寄存器 1 (RTPM01) 的第 n 位

RTBH01m: 实时输出缓存寄存器 1H (RTBH01) 的第 m 位

RTBL01m: 实时输出缓存寄存器 1L (RTBL01) 的第 m 位

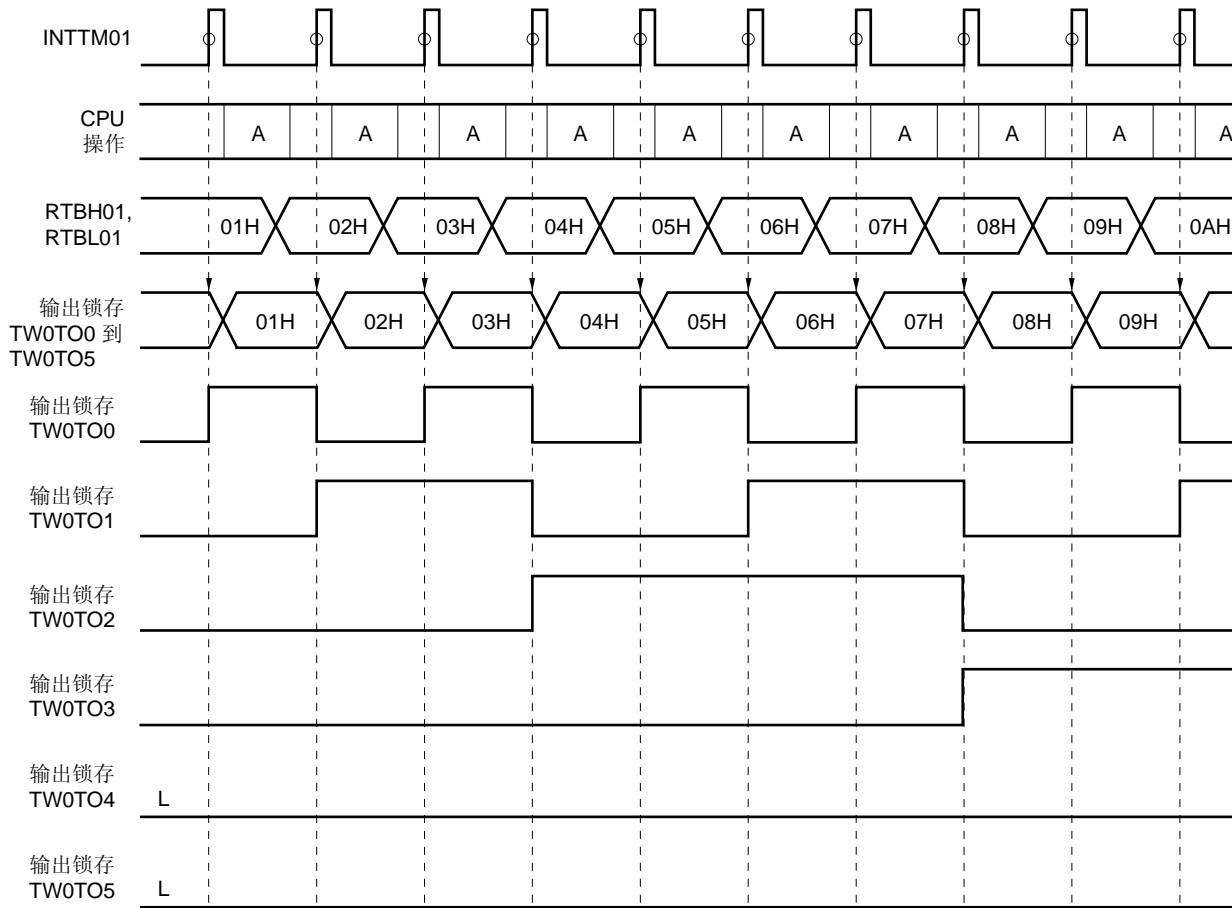
n = 0 到 5

m = 0 到 3

x: 不关注。

图 13-12. 实时输出端口操作时序示例 (6 位 × 1) (1/3)

(a) 6 位 × 1 通道, 反转输出被禁止, 无 PWM 调制
 (BYTE01 = 1, INV01 = 0, PWMCH01 = 0, PWMCL01 = 0)

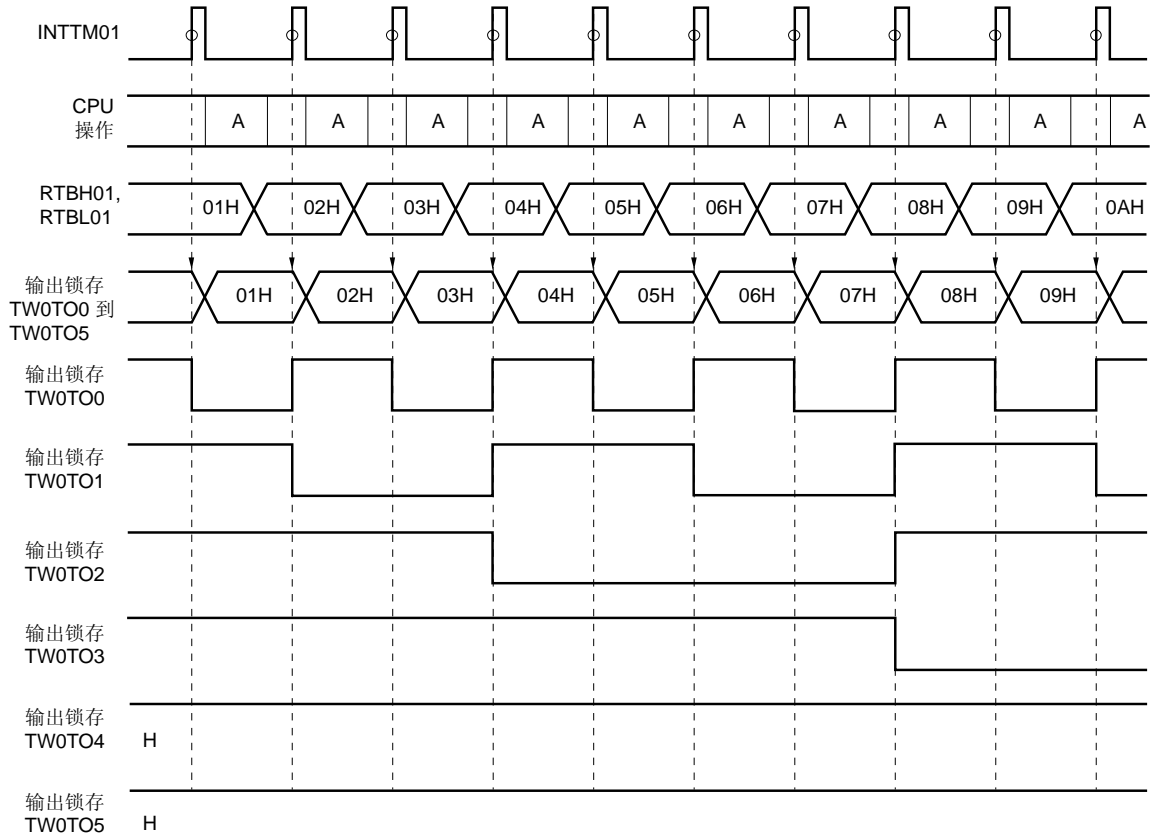


A: INTTM01 软件处理 (RTBH01, RTBL01 写)

图 13-12. 实时输出端口操作时序示例 (6 位 × 1) (2/3)

(b) 6 位 × 1 通道, 反转输出被允许, 无 PWM 调制

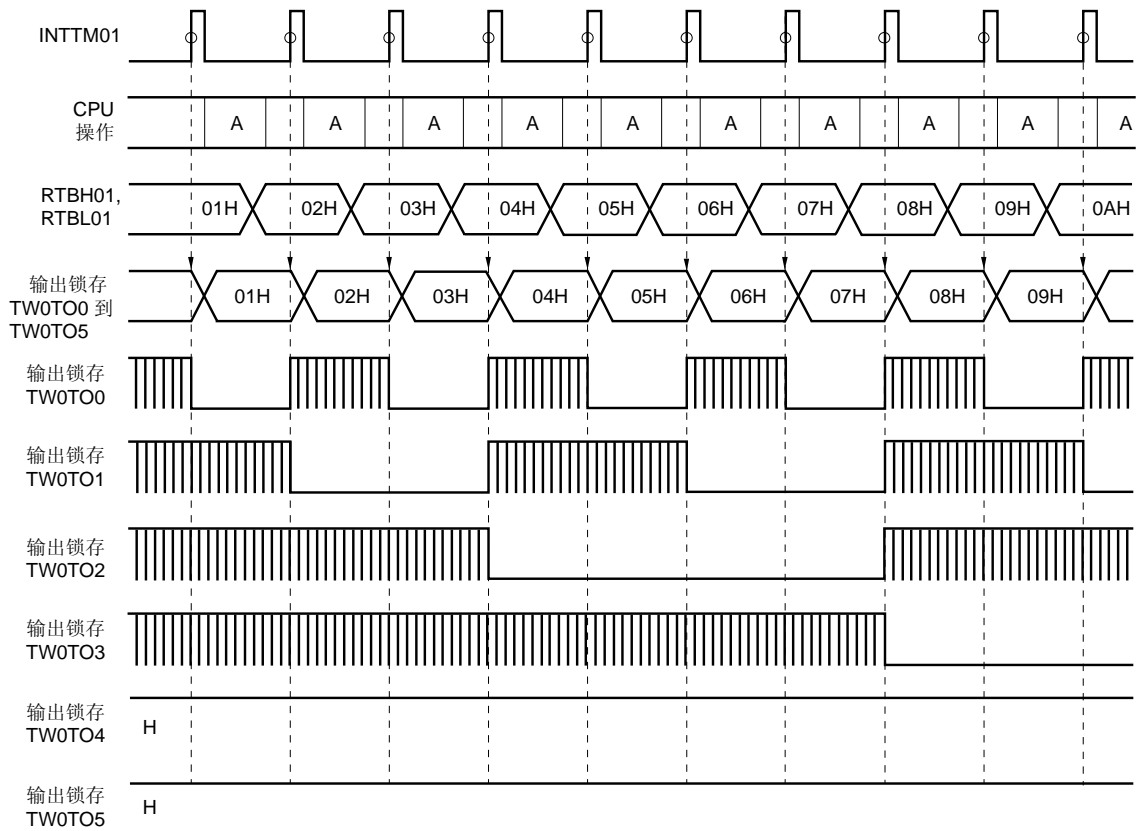
(BYTE01 = 1, INV01 = 1, PWMCH01 = 0, PWMCL01 = 0)



A: INTTM01 软件处理 (RTBH01, RTBL01 写)

图 13-12. 实时输出口操作时序示例 (6 位 × 1) (3/3)

(c) 6 位 × 1 通道, 反转输出被允许, PWM 调制
 (BYTE01 = 1, INV01 = 1, PWMCH01 = 1, PWMCL01 = 1)



A: INTTM01 软件处理 (RTBH01, RTBL01 写)

13.5 使用实时输出端口

当使用实时输出端口时，执行以下步骤。

(1) 禁止实时输出操作。

清除实时输出端口控制寄存器 n (RTPC0n) 的第 7 位 (RTPOE0n) 为 0。

(2) 初始化设置

- 设置初始值到端口输出锁存中 (仅限实时输出端口 0)。
- 以 1 位为单位指定实时输出端口模式。
设置实时输出端口模式寄存器 n (RTPM0n)。
- 选择操作模式 (触发和有效沿)。
设置 RTPC00 的第 4、5 和 6 位 (EXTR00, BYTE00 和 RTPEG00) 或设置 RTPC01 的第 5 位 (BYTE01)。
- 对于实时输出端口 0, 在实时输出缓存寄存器 0 (RTBH00, RTBL00) 中设置与端口输出锁存中相等的初始值。
对于实时输出端口 1, 在实时输出缓存寄存器 1 (RTBH01, RTBL01) 中设置初始值。
- 设置直流控制寄存器 0n (DCCTL0n)。

(3) 允许实时输出操作。

$RTPOE0n = 1$

(4) 设置端口输出锁存为 0 (仅限实时输出端口 0)。

备注 对于实时输出端口 0, 通过实时输出操作输出的值是端口输出锁存和实时输出求或的值 (参见图 13-1 (a))。因此, 当实时输出端口 0 被使用时, 在实时输出操作被允许 ($RTPOE00 = 0 \rightarrow 1$) 后, 端口输出锁存应该被设置为 0, 直到第一个传输触发被产生。

(5) 在选择的传输触发被产生前设置下一个输出到 RTBH0n 和 RTBL0n。

(6) 通过使用对应于所选触发的中断服务程序, 按顺序设置下一个实时输出值到 RTBH0n 和 RTBL0n。

备注 $n = 0, 1$

13.6 实时输出端口的注意事项

- (1) 在执行初始设置之前，通过清除实时输出端口控制寄存器 n (RTPC0n) 的第 7 位 (RTPOE0n) 为 0 来禁止实时输出操作 ($n = 0, 1$)。
- (2) 一旦实时输出操作被禁止 (RTPOE0n = 0)，确保在允许实时输出操作 (RTPOE0n = 0 → 1) 之前，设置与输出锁存相同的初始值到输出缓存寄存器 n (RTBH0n 和 RTBL0n) 中 ($n = 0, 1$)。

第十四章 直流逆变器控制功能

μ PD78F0714 通过 10 位逆变器控制定时器和实时输出端口的组合来实现 3 相 PWM 直流逆变器控制。

参见以下章。

- 第六章 10 位逆变器控制定时器
- 第十三章 实时输出端口

<R>

关于使用直流逆变器控制的应用系统，参阅以下应用笔记。

- μ PD78F0714 无传感器(BEMF) 120° 激励方法的电动机控制(U18051E)
- μ PD78F0714 霍尔器件 120°激励方法的电动机控制(U18774E)

第十五章 A/D转换器

15.1 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，它由 10 位分辨率的最多八个通道（ANI0 到 ANI7）组成。
A/D 转换器有以下两个功能。

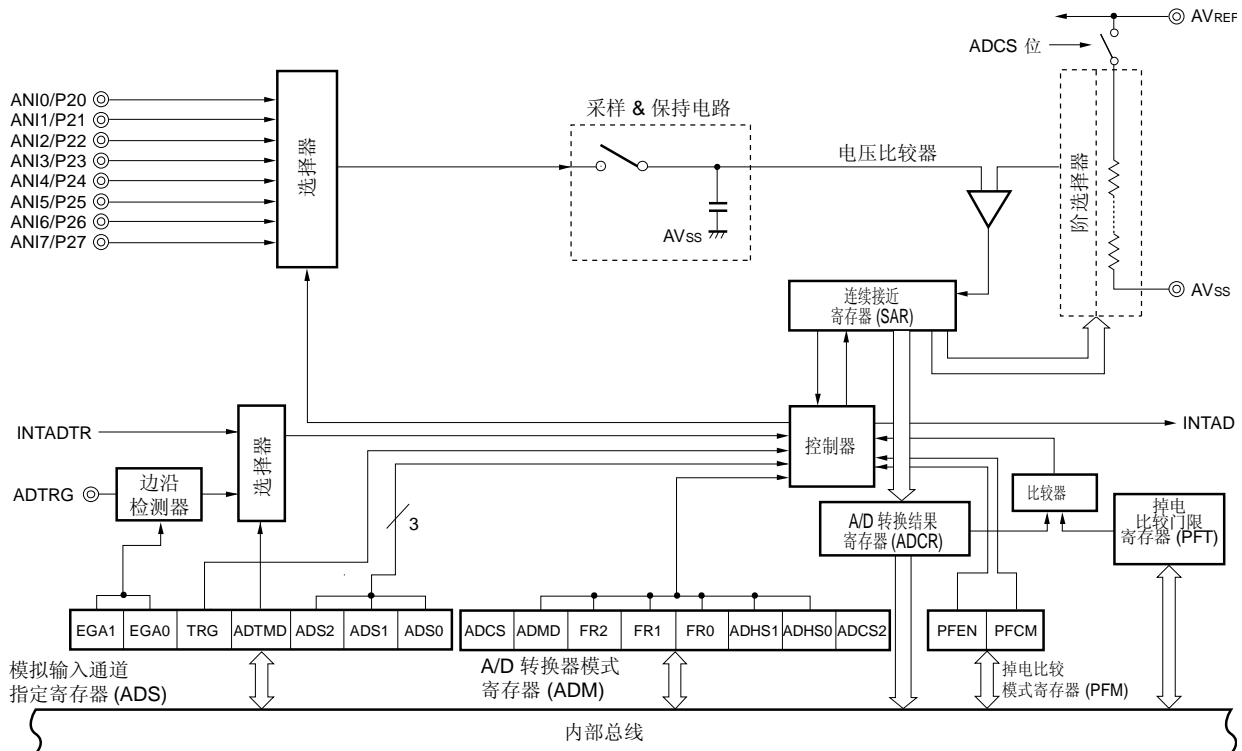
(1) 10 位分辨率 A/D 转换器

10 位分辨率 A/D 转换对从模拟输入 ANI0 到 ANI7 中选择一个通道重复执行。每次一个 A/D 转换结束后，一个中断请求（INTAD）被产生。

(2) 掉电检测功能

该功能用来检测电池中的电压下降。A/D 转换结果（ADCR 寄存器值）和掉电比较门限寄存器（PFT）值比较。只有当比较的条件满足时，INTAD 才会被产生。

图 15-1. A/D 转换器的框图



15.2 A/D转换器的配置

A/D 转换器由以下硬件组成。

表 15-1. 软件中使用的 A/D 转换器的寄存器

项目	配置
寄存器	逐次逼近寄存器 (SAR) A/D 转换结果寄存器 (ADCR) A/D 转换器模式寄存器 (ADM) 模拟输入通道指定寄存器 (ADS) 掉电比较模式寄存器 (PFM) 掉电比较门限寄存器 (PFT)

(1) ANI0 到 ANI7 引脚

这些是 8 通道 A/D 转换器的模拟输入引脚。它们输入要被转换为数字信号的模拟信号。通过模拟输入通道指定寄存器 (ADS) 选择模拟输入引脚以外的引脚可以用作输入端口引脚。

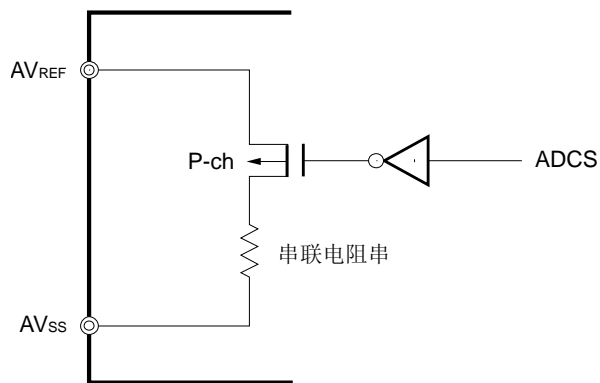
(2) 采样&保持电路

当 A/D 转换被启动时，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信号，并且在 A/D 转换过程中保持采样的模拟输入电压值。

(3) 串联电阻串

串联电阻串被连接到 AV_{REF} 和 AV_{SS} 之间，并且产生一个与模拟输入信号比较的电压。

图 15-2. 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的模拟输入电压和串联电阻串的输出电压。

(5) 逐次逼近寄存器 (SAR)

该寄存器比较采样的模拟电压和串联电阻串的输出电压，并从最高位 (MSB) 转换结果。

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

(6) A/D 转换结果寄存器 (ADCR)

每次 A/D 转换被完成后, A/D 转换结果被从逐次逼近寄存器 (SAR) 加载到该寄存器, 并且 ADCR 寄存器以它的高 10 位 (低 6 位固定为 0) 保持 A/D 转换结果。

(7) 控制器

当 A/D 转换完成时或者当掉电检测功能被使用时, 该控制器比较 A/D 转换结果 (ADCR 寄存器的值) 和掉电比较门限寄存器 (PFT) 的值。只有指定的比较条件满足时才产生中断 INTAD。

(8) AVREF 引脚

这个引脚输入一个模拟电源/参考电压到 A/D 转换器。即使当 A/D 转换器没有被使用时, 总是使该引脚与 V_{DD} 引脚的电平相同。

基于 AVREF 和 AVSS 之间的电压, 输入到 ANI0 到 ANI7 的信号被转换为数字信号。

(9) AVSS 引脚

这是 A/D 转换器的地电平引脚。即使当 A/D 转换器没有被使用时, 总是使这个引脚与 V_{SS} 的电平相同。

(10) 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。

(11) 模拟输入通道指定寄存器 (ADS)

该寄存器被用来指定要转换为数字信号的模拟电压的输入端口。

(12) 掉电比较模式寄存器 (PFM)

该寄存器用来设置掉电监控模式。

(13) 掉电比较门限寄存器 (PFT)

该寄存器用来设置要与 A/D 转换结果寄存器 (ADCR) 的值比较的门限值。

15.3 在A/D转换器中使用的寄存器

A/D 转换器使用以下五个寄存器。

- A/D 转换器模式寄存器 (ADM)
- 模拟输入通道指定寄存器 (ADS)
- A/D 转换结果寄存器 (ADCR)
- 掉电比较模式寄存器 (PFM)
- 掉电比较门限寄存器 (PFT)

(1) A/D 转换器模式寄存器 (ADM)

该寄存器设置要被 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。

RESET 输入清除该寄存器为 00H。

图 15-3. A/D 转换器模式寄存器 (ADM) 的格式

地址: FF6CH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADM	ADCS	ADMD	FR2 ^{注1}	FR1 ^{注1}	FR0 ^{注1}	ADHS1 ^{注1}	ADHS0 ^{注1}	ADCS2

ADCS	A/D 转换操作控制
0	停止转换操作
1	允许转换操作

ADMD	操作模式控制
0	选择模式
1	扫描模式

<R>

FR2	FR1	FR0	ADHS1	ADHS0	A/D 转换时间选择
x	x	x	0	0	禁止设置
0	0	0	0	1	96/fx
0	0	1	0	1	72/fx
0	1	0	0	1	48/fx
0	1	1	0	1	24/fx
1	0	0	0	1	224/fx
1	0	1	0	1	168/fx
1	1	0	0	1	112/fx
1	1	1	0	1	56/fx
0	0	0	1	0	72/fx
0	0	1	1	0	54/fx
0	1	0	1	0	36/fx
0	1	1	1	0	18/fx
1	x	x	1	0	禁止设置
x	x	x	1	1	禁止设置

ADCS2	升压参考电压产生器操作控制 ^{注2}
0	停止参考电压产生器的操作
1	允许参考电压产生器的操作

注 1. 在 FR2 到 FR0、ADHS1 和 ADHS0 的组合中选择 A/D 转换时间。

关于 A/D 转换的详细情况，参见表 15-3。

2. 一个升压电路被集成用来实现低压操作。产生升压参考电压的电路的操作由 ADCS2 控制，并且从操作开始到操作稳定需要 1 μs。因此，在 ADCE 被设置为 1 后 1 μs 或更长时间过去后，ADCS2 被设置为 1，这时的转换结果比第一次的转换结果有更高优先级。

备注 fx: X1 输入时钟振荡频率

(a) 控制升压参考电压产生器

当 ADCS2 位 = 0 时，A/D 转换器的电源下降。在 ADCS2 位被设置为 1 后，转换器需要 1 μs 或更多的建立时间。

因此，从设置 ADCS 位为 1 的第一个结果到 ADCS2 位被设置为 1 至少 1 μs ，A/D 转换的结果才变为有效。

表 15-2. ADCS 和 ADCS2 的设置

ADCS	ADCS2	A/D 转换操作
0	0	停止状态（直流电源消耗通道不存在）
0	1	转换等待模式（只有参考电压产生器消耗电源）
1	0	转换模式（参考电压产生器操作被停止 ^{#1} ）
1	1	转换模式（参考电压产生器操作 ^{#2} ）

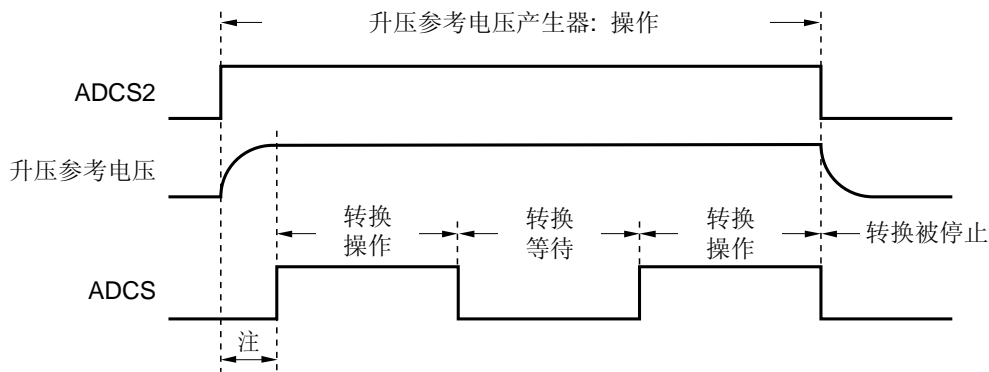
注 1. 如果 ADCS 和 ADCS2 位从 00B 变为 10B，升压参考电压产生器自动打开。如果 ADCS2 位为 0 时 ADCS 位被清除为 0，电压产生器自动关闭。在软件触发模式（ADS.TRG 位 = 0）下，禁止使用第一个 A/D 转换结果。

在硬件触发模式（TRG 位 = 1）下，只有在升压参考电压产生器的振荡稳定时间过去后启动 A/D 转换时才能使用 A/D 转换结果。

2. 如果 ADCS 和 ADCS2 位从 00B 变为 11B，升压参考电压产生器自动打开。如果 ADCS2 位为 1 时 ADCS 位被清除为 0，电压产生器保持打开。在软件触发模式（TRG 位 = 0）下，禁止使用第一个 A/D 转换结果。

在硬件触发模式（TRG 位 = 1）下，只有在升压参考电压产生器的振荡稳定时间过去后启动 A/D 转换时才能使用 A/D 转换结果。

图 15-4. 升压参考电压产生器被使用时的时序图



注 从 ADCS2 位上升到 ADCS 位下降的时间必须为 1 μs 或更长以使参考电压稳定。

注意事项 1. 在向位 FR0 到 FR2、ADHS0 和 ADHS1 重新写入不同数据前，A/D 转换必须被停止。

2. 如果数据被写入 ADM，一个等待周期被产生。关于详细情况，见第三十章 等待注意事项。

<R>

表 15-3. A/D 转换时间

FR2	FR1	FR0	ADHS1	ADHS0	转换时间 (t _{CONV})					
					f _x = 20 MHz	f _x = 16 MHz	f _x = 10 MHz	f _x = 8.38 MHz	f _x = 5 MHz	
×	×	×	0	0	禁止设置	禁止设置	禁止设置	禁止设置	禁止设置	禁止设置
0	0	0	0	1	96/f _x	4.8 μs	6 μs	9.6 μs	11.5 μs	19.2 μs
0	0	1	0	1	72/f _x	3.6 μs [‡]	4.5 μs [‡]	7.2 μs	8.6 μs	14.4 μs
0	1	0	0	1	48/f _x	禁止设置	禁止设置	4.8 μs	5.8 μs	9.6 μs
0	1	1	0	1	24/f _x	禁止设置	禁止设置	禁止设置	禁止设置	4.8 μs
1	0	0	0	1	224/f _x	11.2 μs	14 μs	22.4 μs	26.8 μs	44.8 μs
1	0	1	0	1	168/f _x	8.4 μs	10.5 μs	16.8 μs	20.1 μs	33.6 μs
1	1	0	0	1	112/f _x	5.6 μs	7 μs	11.2 μs	13.4 μs	22.4 μs
1	1	1	0	0	56/f _x	禁止设置	4.5 μs [‡]	5.6 μs	6.7 μs	11.2 μs
0	0	0	1	0	72/f _x	3.6 μs [‡]	禁止设置	7.2 μs	8.6 μs	14.4 μs
0	0	1	1	0	54/f _x	禁止设置	禁止设置	5.4 μs	6.5 μs	10.8 μs
0	1	0	1	0	36/f _x	禁止设置	禁止设置	3.6 μs [‡]	4.3 μs [‡]	7.2 μs
0	1	1	1	0	18/f _x	禁止设置	禁止设置	禁止设置	禁止设置	3.6 μs [‡]
1	×	×	1	0	禁止设置	禁止设置	禁止设置	禁止设置	禁止设置	禁止设置
×	×	×	1	1	禁止设置	禁止设置	禁止设置	禁止设置	禁止设置	禁止设置

注 如果 $3.6 \mu\text{s} \leq t_{\text{CONV}} < 4.8 \mu\text{s}$, 只有在 $\text{AV}_{\text{REF}} \geq 4.5 \text{ V}$ 时才能设置。

备注 f_x: X1 输入时钟振荡频率

(2) 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端口。

ADS 可以通过一个 1 位或 8 位存储器操作指令来设置。

RESET 输入清除该寄存器为 00H。

图 15-5. 模拟输入通道指定寄存器 (ADS) 的格式

地址: FF6DH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADS	EGA1	EGA0	TRG	ADTMD	0	ADS2	ADS1	ADS0
EGA1 ^{注1}	EGA0 ^{注1}	外部触发信号 (ADTRG) 边沿的指定						
0	0	无边沿检测						
0	1	下降沿						
1	0	上升沿						
1	1	双边沿						
TRG	触发模式选择							
0	软件触发模式							
1	硬件触发模式							
ADTMD ^{注2}	硬件触发模式的指定							
0	外部触发 (ADTRG 引脚输入)							
1	定时器触发 (INTADTR 信号被产生)							
ADS2	ADS1	ADS0	模拟输入通道指定					
			选择模式			扫描模式		
0	0	0	ANI0			ANI0		
0	0	1	ANI1			ANI0, ANI1		
0	1	0	ANI2			ANI0 到 ANI2		
0	1	1	ANI3			ANI0 到 ANI3		
1	0	0	ANI4			ANI0 到 ANI4		
1	0	1	ANI5			ANI0 到 ANI5		
1	1	0	ANI6			ANI0 到 ANI6		
1	1	1	ANI7			ANI0 到 ANI7		

- 注**
1. 只有当硬件触发模式 (TRG 位 = 1) 和外部触发模式 (ADTRG 引脚输入: ADTMD 位 = 1) 被选择时, EGA1 和 EGA0 位才有效。
 2. 只有当硬件触发模式 (TRG 位 = 1) 被选择时, ADTMD 位才有效。

- 注意事项**
1. 确认清除 ADS 的第 3 位为 0。
 2. 如果数据被写入 ADS, 一个等待周期被产生。关于详细情况, 见第三十章 等待注意事项。

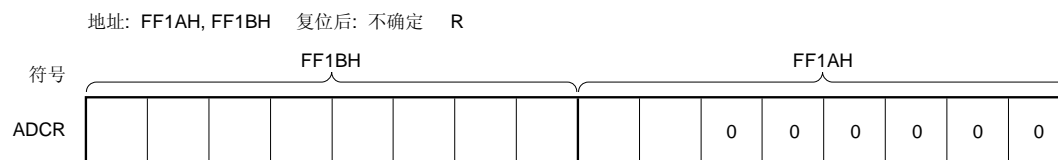
(3) A/D 转换结果寄存器 (ADCR)

该寄存器是一个保存 A/D 转换结果的 16 位寄存器。每次 A/D 转换结束，转换结果从逐次逼近寄存器中加载。低 6 位固定为 0。转换结果按照从 MSB 开始的顺序存储在 ADCR 中。转换结果的高 8 位保存在 FF1BH 中，低 2 位保存在 FF1AH 中。

ADCR 可以通过一个 16 位存储器操作指令来读取。

$\overline{\text{RESET}}$ 输入使 ADCR 不确定。

图 15-6. 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) 和模拟输入通道指定寄存器 (ADS) 时，ADCR 的内容可能变为不确定。在转换完成后并且写入 ADM 和 ADS 前读取转换结果。使用上面以外的时序可能导致一个不正确的转换结果被读取。
 2. 如果数据从 ADCR 被读取，一个等待周期被产生。关于详细情况，见第三十章 等待注意事项。

(4) 掉电比较模式寄存器 (PFM)

掉电比较模式寄存器 (PFM) 用来比较 A/D 转换结果 (ADCR 寄存器的值) 和掉电比较门限寄存器 (PFT) 的值。PFM 可以通过一个 1 位或 8 位存储器操作指令来设置。
 $\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 15-7. 掉电比较模式寄存器 (PFM) 的格式

地址: FF6EH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PFM	PFEN	PFCM	0	0	0	0	0	0

PFEN	掉电比较允许
0	停止掉电比较(用作正常 A/D 转换器)
1	允许掉电比较 (用作掉电检测)

PFCM		掉电比较模式选择
0	ADCR 的高 8 位 \geq PFT	中断请求信号 (INTAD) 产生
	ADCR 的高 8 位 $<$ PFT	无 INTAD 产生
1	ADCR 的高 8 位 \geq PFT	无 INTAD 产生
	ADCR 的高 8 位 $<$ PFT	INTAD 产生

注意事项 如果数据被写入 PFM, 一个等待周期被产生。关于详细情况, 见第三十章 等待注意事项。

(5) 掉电比较门限寄存器 (PFT)

掉电比较门限寄存器 (PFT) 是设置与 A/D 转换结果进行比较的门限值的寄存器。PFT 中的 8 位数据与 10 位 A/D 转换结果的高 8 位 (FF1BH) 进行比较。PFT 可以通过一个 8 位存储器操作指令来设置。
 $\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 15-8. 掉电比较门限寄存器 (PFT) 的格式

地址: FF6FH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PFT	PFT7	PFT6	PFT5	PFT4	PFT3	PFT2	PFT1	PFT0

注意事项 如果数据被写入 PFT, 一个等待周期被产生。关于详细情况, 见第三十章 等待注意事项。

15.4 输入电压和A/D转换结果之间的关系

输入到模拟输入引脚（AN10 到 AN17）的模拟输入电压和理论的 A/D 转换结果（被保存在 A/D 转换结果寄存器（ADCR）中）之间的关系通过以下表达式被表示。

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

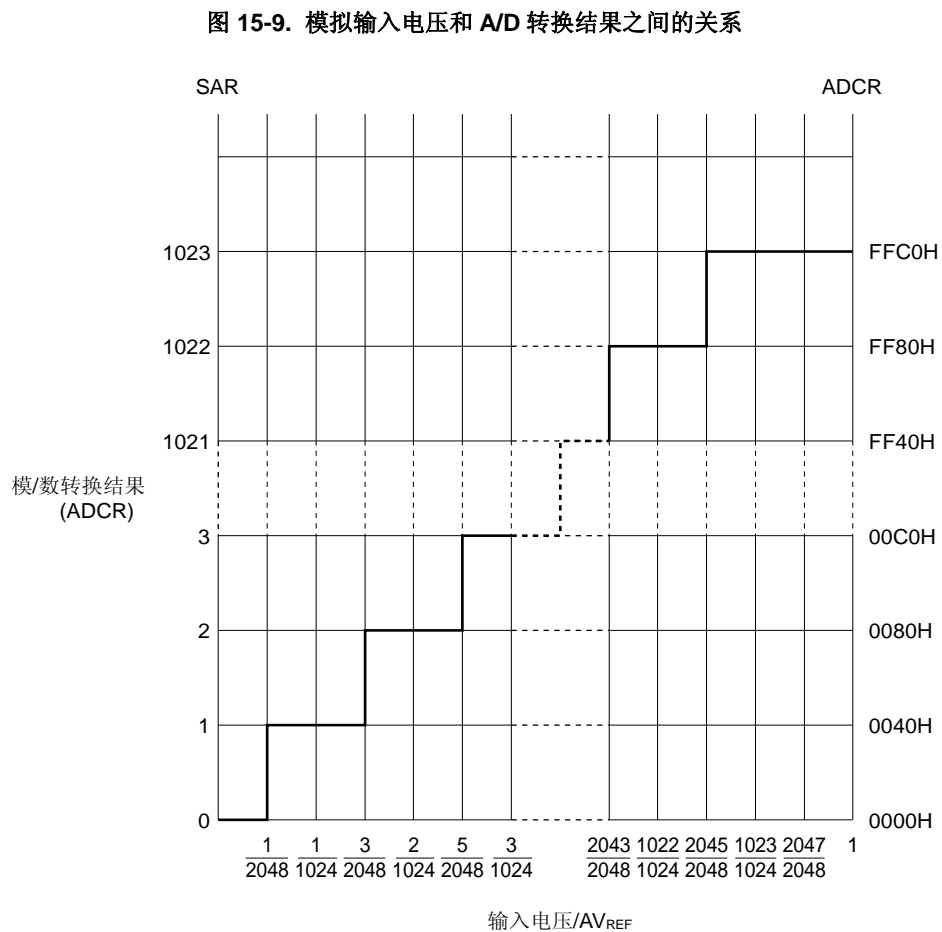
$$ADCR = SAR \times 64$$

或者

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

- 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 15-9 表示模拟输入电压和 A/D 转换结果之间的关系。

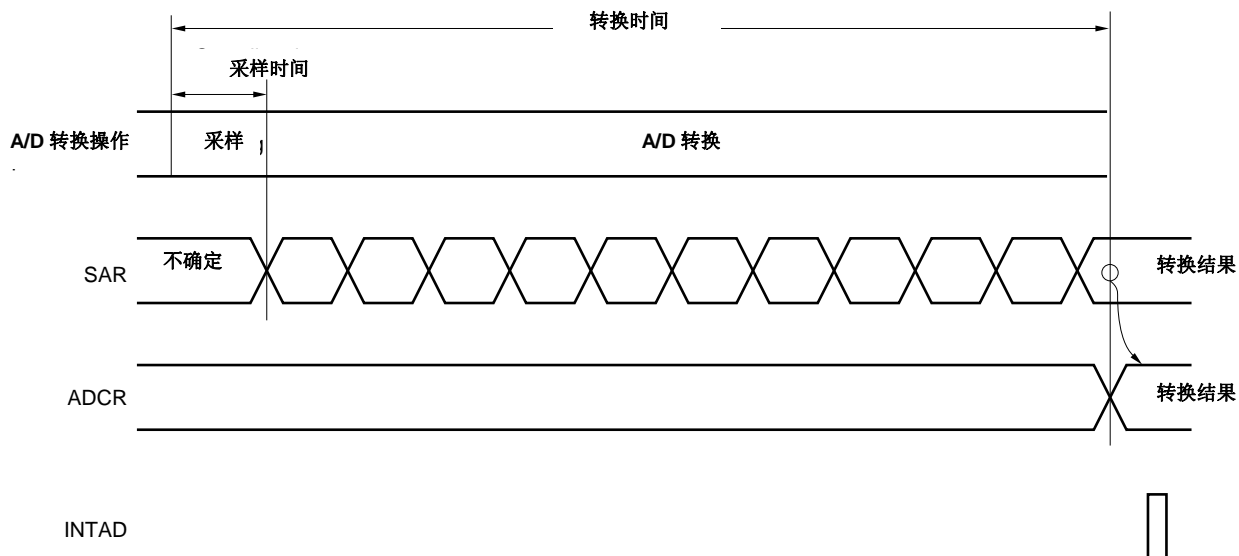


15.5 A/D转换器的操作

15.5.1 A/D转换器的基本操作

- <1> 使用模拟输入通道指定寄存器（ADS）来为 A/D 转换选择一个通道。
使用 A/D 转换器模式寄存器（ADM）的 FR2 到 FR0、ADHS1 和 ADSH0 位来选择转换时间。
- <2> 设置 ADCS2 为 1 并等待 1 μ s 或更长。
- <3> 设置 ADCS 为 1 并开始转换操作。
(<4> 到 <10>是硬件执行的操作。)
- <4> 选择的模拟输入通道的电压输入被采样&保持电路采样。
- <5> 当采样被完成一定时间后，采样&保持电路被置于保持状态，并且输入模拟电压一直被保持到 A/D 转换操作结束。
- <6> 逐次逼近寄存器（SAR）的第 9 位被设置。通过阶选择器，串联电阻串电压阶被设置为 $(1/2) AV_{REF}$ 。
- <7> 通过电压比较器，串联电阻串阶和模拟输入之间的差别被比较。如果模拟输入比 $(1/2) AV_{REF}$ 大，SAR 的 MSB 仍然被设置为 1。如果模拟输入比 $(1/2) AV_{REF}$ 小，MSB 被复位为 0。
- <8> 然后，SAR 的第 8 位自动被设置为 1，并且处理下一个比较。串联电阻串电压阶按照第 9 位的预设值被选择，如下所示。
 - 第 9 位 = 1: $(3/4) AV_{REF}$
 - 第 9 位 = 0: $(1/4) AV_{REF}$
 电压阶和模拟输入电压被比较，并且 SAR 的第 8 位按照下面被修改。
 - 模拟输入电压 \geq 电压阶: 第 8 位 = 1
 - 模拟输入电压 < 电压阶: 第 8 位 = 0
- <9> 按照这种方式，比较被继续，直到 SAR 的第 0 位。
- <10> 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），然后被锁存。
这时，A/D 转换结束中断请求（INTAD）也被产生。
- <11> 重复步骤<4> 到 <10>，直到 ADCS 被清除为 0。
要停止 A/D 转换器，清除 ADCS 为 0。
要从 ADCS2 = 1 的状态重新启动 A/D 转换器，从<3>开始。然而，要从 ADCS2 = 0 的状态重新启动 A/D 转换器，从<2>开始。

图 15-10. A/D 转换器的基本操作



A/D 转换操作被持续执行，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 转换操作过程中向 ADM、模拟输入通道指定寄存器 (ADS)、掉电比较模式寄存器 (PFM) 或掉电比较门限寄存器 (PFT) 中的一个执行写操作，转换操作被初始化，并且如果 ADCS 位被置位 (1)，转换从开始时再次启动。

$\overline{\text{RESET}}$ 输入使 A/D 转换结果寄存器 (ADCR) 不确定。

15.5.2 触发模式

μ PD78F0714 具有以下三种设置 A/D 启动时序的触发模式。这些触发模式由 ADS 寄存器设置。

- 软件触发模式
- 外部触发模式（硬件触发模式）
- 定时器触发模式（硬件触发模式）

(1) 软件触发模式

该模式通过在 ADS.TRG 位为 0 时设置 ADM.ADCS 位为 1 来启动 A/D 转换。

在 A/D 转换完成后，只要 ADCS 位没有被清除为 0，转换就重复执行。

如果在转换期间 ADM、ADS、PFM 或 PFT 寄存器被重新写入，A/D 转换被中止并且从开始时再次启动。

(2) 外部触发模式（硬件触发模式）

该状态下，ADS.TRG 位被设置为 1 并且 ADS.ADTMD 位被清除为 0。该模式通过在 ADCS 位被设置为 1 后检测外部触发（ADTRG）来启动 A/D 转换。

A/D 转换器在 ADCS 位被设置为 1 后等待外部触发（ADTRG）。

输入到引脚的信号的有效沿通过使用 ADS.EGA1 和 ADS.EGA0 位来指定。当指定的有效沿被检测时，A/D 转换被启动。

当 A/D 转换完成时，A/D 转换器再次等待外部触发（ADTRG）。

如果在 A/D 转换期间有效沿被输入到 ADTRG 引脚，A/D 转换继续而不检测触发。

如果在转换期间 ADM、ADS、PFM 或 PFT 寄存器被重新写入，A/D 转换被中止并且 A/D 转换器等待外部触发（ADTRG）。

(3) 定时器触发模式（硬件触发模式）

该模式通过在 ADCS 被设置为 1 并且 TRG 位 = 1 和 ADTMD 位 = 1 后检测定时器触发（INTADTR）来启动 A/D 转换。

A/D 转换器在 ADCS 位被设置为 1 后等待定时器触发（INTADTR）。

当 INTADTR 信号产生时，A/D 转换被启动。

当 A/D 转换完成时，A/D 转换器再次等待定时器触发（INTADTR）。

如果 INTADTR 信号在 A/D 转换期间被产生，A/D 转换被中止并且从开始时再次启动。

如果在转换期间 ADM、ADS、PFM 或 PFT 寄存器被重新写入，A/D 转换被中止并且 A/D 转换器等待定时器触发（INTADTR）。

15.5.3 操作模式

可以使用以下两种操作模式。这些操作模式由 ADM 寄存器设置。

- 选择模式
- 扫描模式

(1) 选择模式

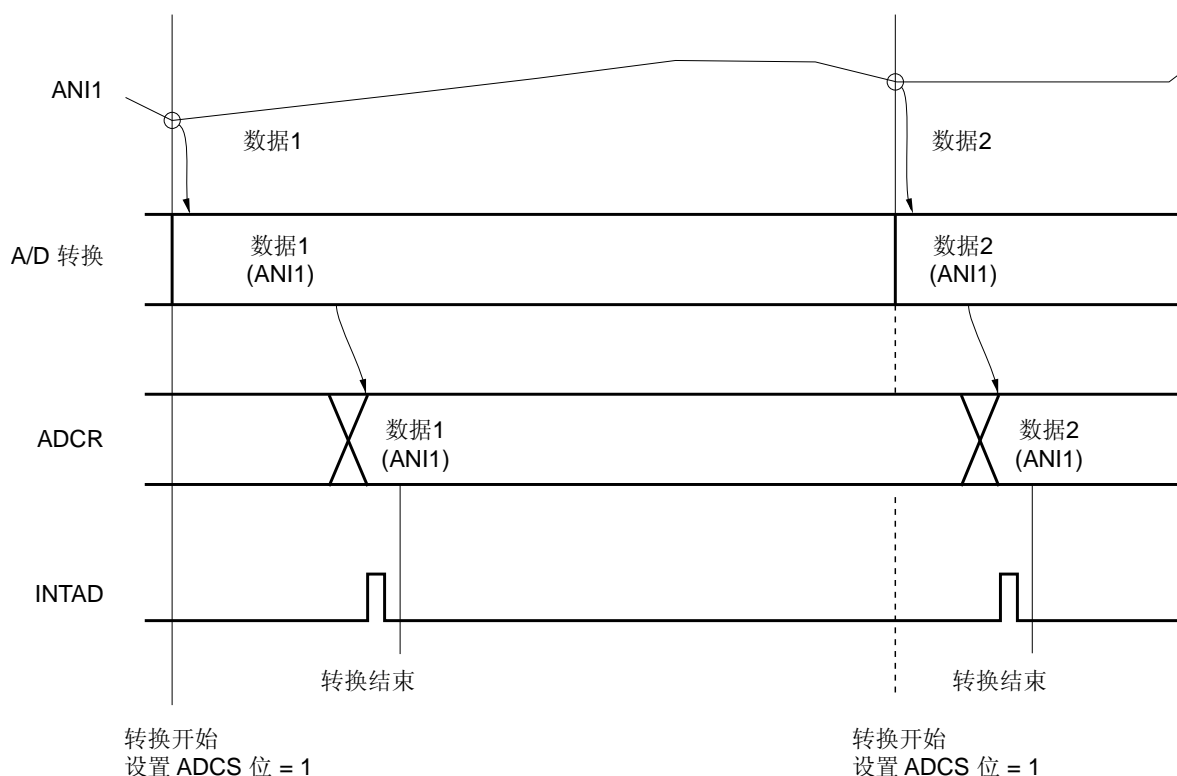
ADM.ADM位 = 0 时，由 ADS 寄存器指定的一个输入模拟信号被转换。当转换完成时，转换的结果被保存到 ADCR 寄存器中。

同时，A/D 转换结束中断请求信号 (INTAD) 被产生。然而，INTAD 信号是否产生依赖于 PFM 和 PFT 寄存器的设置。关于详细情况，参阅 15.5.4 掉电检测功能。

如果在转换期间向 ADM、ADS、PFM 和 PFT 寄存器写入任意值，A/D 转换被中止。在软件触发模式下，A/D 转换从开始时再次启动。在硬件触发模式下，A/D 转换器等待一个触发。

如果在硬件触发模式下触发被检测，A/D 转换被中止并且从开始时再次启动。

图 15-11. 选择模式操作时序示例 (ADS.ADS2 到 ADS.ADS0 位 = 001B)



(2) 扫描模式

在该模式下，ADM.ADM位 = 1 时，由 ADS 寄存器指定的模拟信号和从 AN10 引脚的输入按顺序被选择并转换。当一个模拟输入信号的转换完成时，转换结果被保存到 ADCR 寄存器中，并且同时 A/D 转换结束中断请求信号 (INTAD) 被产生。

所有模拟输入信号的 A/D 转换结果被保存到 ADCR 寄存器中。因此，建议一旦一个模拟输入信号的 A/D 转换完成立即保存 ADCR 寄存器的内容到 RAM 中。

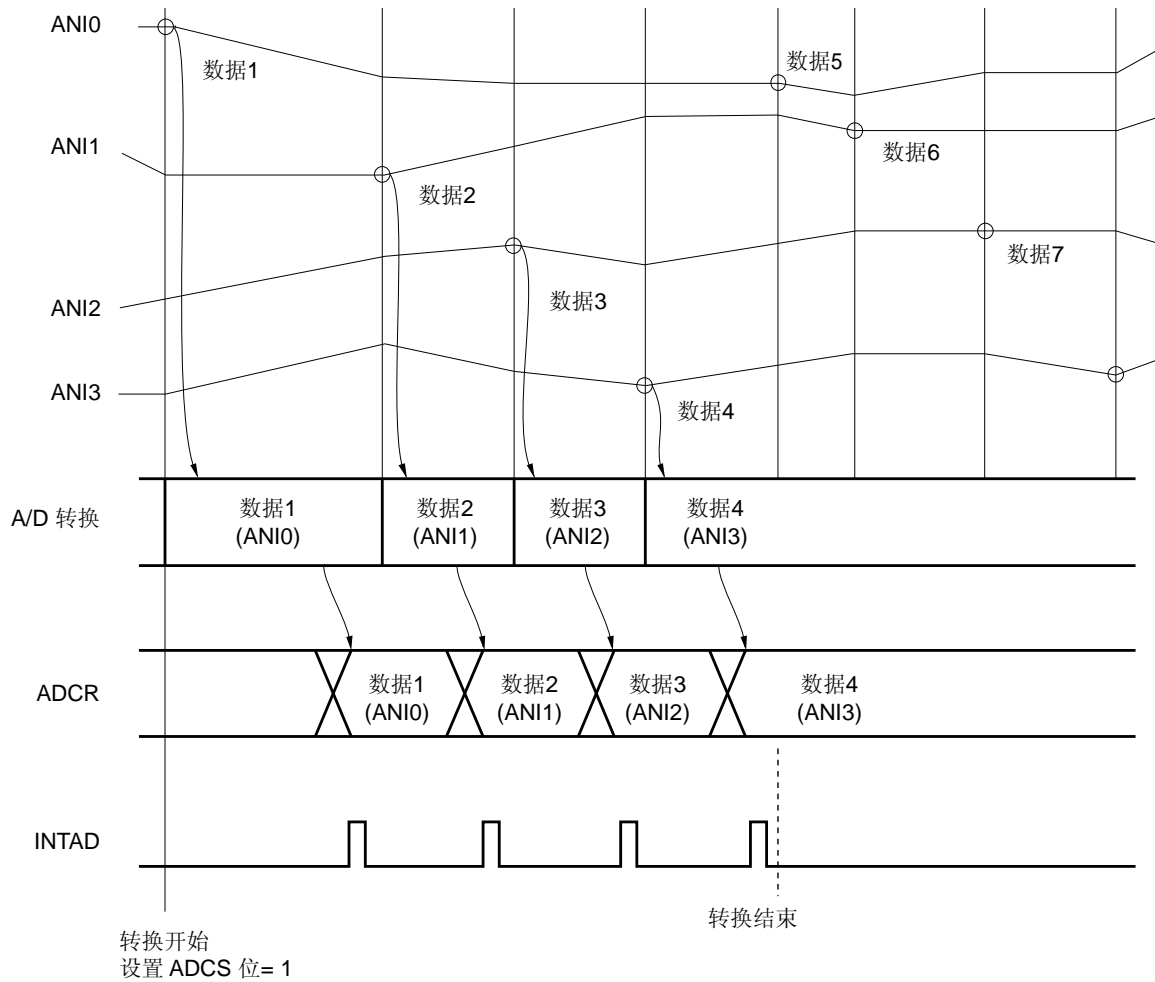
在硬件触发模式 (ADS.TRG 位 = 1) 下，在完成由 ADS 寄存器指定的模拟信号和 AN10 引脚的输入的 A/D 转换后，A/D 转换器等待一个触发。

如果在转换期间向 ADM、ADS、PFM 和 PFT 寄存器写入任意值，A/D 转换被中止。在软件触发模式下，A/D 转换从开始时再次启动。在硬件触发模式下，A/D 转换器等待一个触发。转换从 AN10 引脚再次开始。

如果在硬件触发模式下触发被检测，A/D 转换被中止并且从开始 (AN10 引脚) 时再次启动。

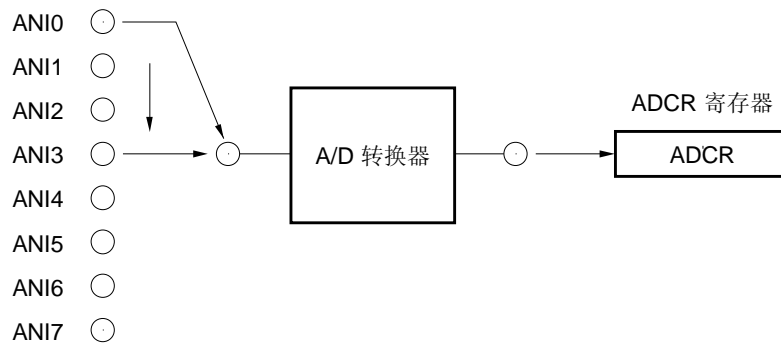
图 15-12. 扫描模式操作时序示例 (ADS.ADS2 到 ADS.ADS0 位 = 011B)

(a) 时序示例



(b) 框图

模拟输入引脚



15.5.4 掉电监控功能

通过掉电比较模式寄存器 (PFM) 的第 7 位 (PFEN) 的设置, 以下两种功能可以被选择。

- 正常 10 位 A/D 转换器 (PFEN = 0)
- 掉电检测功能 (PFEN = 1)

(1) 正常 A/D 转换操作 (PFEN = 0 时)

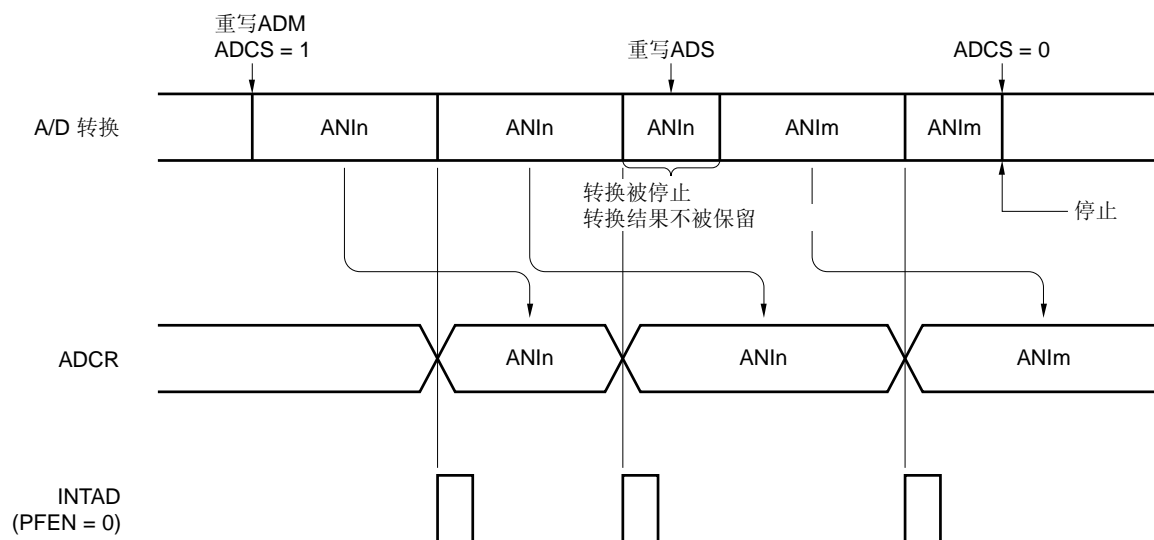
通过设置 A/D 转换器模式寄存器 (ADM) 的第 7 位为 1 和掉电比较模式寄存器 (PFM) 的第 7 位 (PFEN) 为 0, 模拟输入通道指定寄存器 (ADS) 指定的模拟输入引脚上的电压的 A/D 转换操作被启动。

当 A/D 转换完成后, A/D 转换的结果被保存到 A/D 转换结果寄存器 (ADCR) 中, 并且产生一个中断请求信号 (INTAD)。一旦 A/D 转换开始并且当 A/D 转换完成时, 下一个 A/D 转换操作立即被启动。A/D 转换操作重复执行, 直到新的数据被写入 ADS。

如果 ADM、ADS、掉电比较模式寄存器 (PFM) 和掉电比较门限寄存器 (PFT) 在 A/D 转换期间被重新写入, 正在执行的 A/D 转换操作被停止并从开始时重新启动。

如果在 A/D 转换过程中 0 被写入 ADCS, A/D 转换立即被停止。此时, 转换结果不确定。

图 15-13. A/D 转换操作



- 备注
1. n = 0 到 7
 2. m = 0 到 7

(2) 掉电检测功能 (PFEN = 1 时)

通过设置 A/D 转换器模式寄存器 (ADM) 的第 7 位为 1 和掉电比较模式寄存器 (PFM) 的第 7 位 (PFEN) 为 1, 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 转换操作被启动。

当 A/D 转换完成时, A/D 转换的结果被保存到 A/D 转换结果寄存器 (ADCR) 中, 值与掉电比较门限寄存器 (PFT) 进行比较, 并且在由 PFM 的第 6 位 (PFCM) 指定的条件下产生一个中断请求信号 (INTAD)。

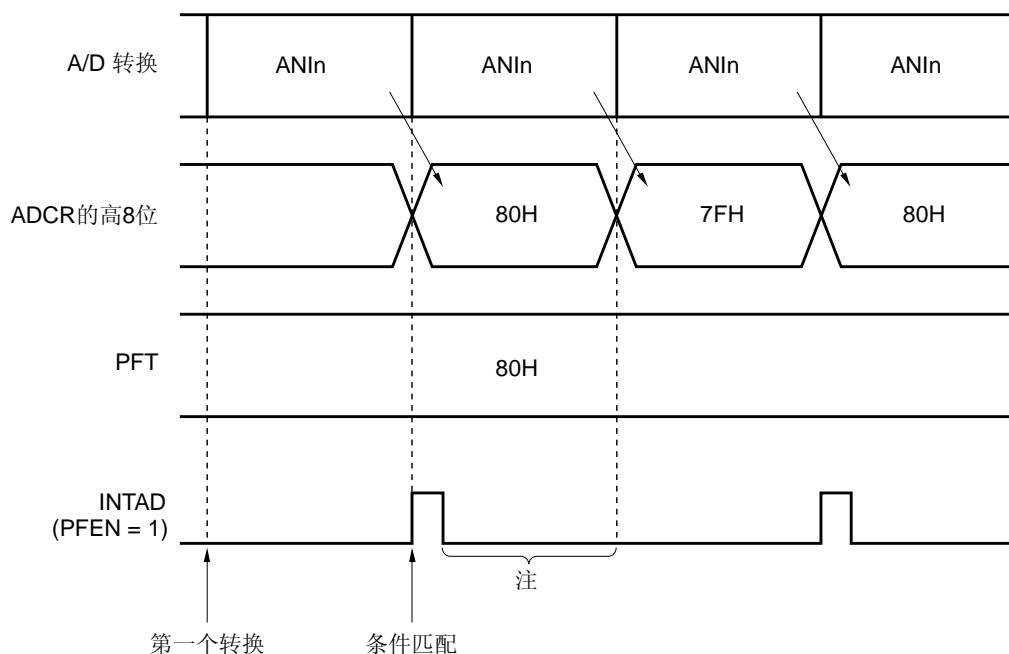
<1> 当 PFEN = 1 并且 PFCM = 0 时

当 A/D 转换结束时, ADCR 的高 8 位和 PFT 值进行比较, 并且只有当 ADCR 的高 8 位 \geq PFT 时 INTAD 产生。

<2> 当 PFEN = 1 并且 PFCM = 1 时

当 A/D 转换结束时, ADCR 的高 8 位和 PFT 值进行比较, 并且只有当 ADCR 的高 8 位 $<$ PFT 时 INTAD 产生。

图 15-14. 掉电检测 (当 PFEN = 1 并且 PFCM = 0 时)



注 如果在 INTAD 被输出后下一个转换结束之前转换结果未被读取, 结果将被下一个转换结果替换。

备注 n = 0 到 7

(3) 设置

设置方法如下所示。

- 当用作正常 A/D 转换操作时
 - <1> 设置 A/D 转换器模式寄存器 (ADCS2) 的第 0 位 (ADCE) 为 1。
 - <2> 使用模拟输入通道指定寄存器 (ADS) 的第 2 到 0 位 (ADS2 到 ADS0) 和 ADM 的第 5 到 1 位 (FR2 到 FR0, ADHS1, ADHS0) 来选择通道和转换时间。
 - <3> 设置 ADM 的第 7 位 (ADCS) 来启动 A/D 转换。
 - <4> 产生一个中断请求信号 (INTAD)。
 - <5> 传送 A/D 转换数据到 A/D 转换结果寄存器 (ADCR) 中。
- <更改通道>
 - <6> 使用 ADS 的第 2 到 0 位 (ADS2 到 ADS0) 来更改通道并启动 A/D 转换。
 - <7> 产生一个中断请求信号 (INTAD)。
 - <8> 传送 A/D 转换数据到 A/D 转换结果寄存器 (ADCR) 中。
- <完成 A/D 转换>
 - <9> 清除 ADCS 为 0。
 - <10> 清除 ADCS2 为 0。

- 注意事项**
1. 确认<1> 到 <3>的周期为 1 μ s 或更长。
 2. 如果<1> 和 <2> 的顺序反转, 没有问题。
 3. <1>可以被省略。然而, 在这种情况下, 不要使用<3> 后的第一个转换结果。
 4. <4> 到 <7>的周期与使用 ADM 的第 5 到 1 位 (FR2 到 FR0, ADHS1, ADHS0) 设置的转换时间不同。<6> 到 <7>的周期是使用 FR2 到 FR0、ADHS1 和 ADHS0 设置的转换时间。

- 当用作掉电检测功能时
 - <1> 设置掉电比较模式寄存器 (PFM) 的第 7 位 (PFEN)。
 - <2> 使用 PFM 的第 6 位 (PFCM) 来设置掉电比较条件。
 - <3> 设置 A/D 转换器模式寄存器 (ADCS2) 的第 0 位 (ADCE) 为 1。
 - <4> 使用模拟输入通道指定寄存器 (ADS) 的第 2 到 0 位 (ADS2 到 ADS0) 和 ADM 的第 5 到 1 位 (FR2 到 FR0, ADHS1, ADHS0) 来选择通道和转换时间。
 - <5> 设置一个门限值到掉电比较门限寄存器 (PFT) 中。
 - <6> 设置 ADM 的第 7 位 (ADCS) 为 1。
 - <7> 传送 A/D 转换数据到 A/D 转换结果寄存器 (ADCR) 中。
 - <8> ADCR 的高 8 位与 PFT 进行比较, 并且如果条件匹配, 产生一个中断请求信号 (INTAD)。
- <更改通道>
 - <9> 使用 ADS 的第 2 到 0 位 (ADS2 到 ADS0) 来更改通道。
 - <10> 传送 A/D 转换数据到 A/D 转换结果寄存器 (ADCR) 中。
 - <11> ADCR 的高 8 位与掉电比较门限寄存器 (PFT) 进行比较, 并且如果条件匹配, 产生一个中断请求信号 (INTAD)。
- <完成 A/D 转换>
 - <12> 清除 ADCS 为 0。
 - <13> 清除 ADCS2 为 0。

- 注意事项**
1. 确认<3>到<6>的周期为 1 μ s 或更长。
 2. 如果<3>、<4>和<5>的顺序改变, 没有问题。
 3. 如果掉电检测功能被使用, <3>必须被省略。
 4. <7>到<11>的周期与使用 ADM 的第 5 到 1 位 (FR2 到 FR0, ADHS1, ADHS0) 设置的转换时间不同。<9>到<11>的周期是使用 FR2 到 FR0、ADHS1 和 ADHS0 设置的转换时间。

备注 无论是选择模式还是扫描模式, 当掉电检测功能被允许时, 总是对所有 A/D 转换结果执行比较操作。

15.6 如何阅读A/D转换器特性表

这里，A/D 转换器的独特术语被解释。

(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被叫做 **1LSB**（最低位）。1LSB 在满幅中占的百分比被表示为 **%FSR**（满幅范围）。

当分辨率为 10 位时，1LSB 如下所示。

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

精度与分辨率无关，而是由全部误差确定。

(2) 全部误差

这表示时间测量值和理论值之间的最大误差值。

零幅度误差、满幅度误差、积分线性误差和差分线性误差的组合表示全部误差。

注意，在特性表中量化误差未被包含在全部误差内。

(3) 量化误差

当模拟值被转换为数字值时，一个 $\pm 1/2\text{LSB}$ 误差会自然地发生。在 A/D 转换器中，一个在 $\pm 1/2\text{LSB}$ 范围内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全部误差、零幅度误差、满幅度误差、积分线性误差和差分线性误差中。

图 15-15. 全部误差

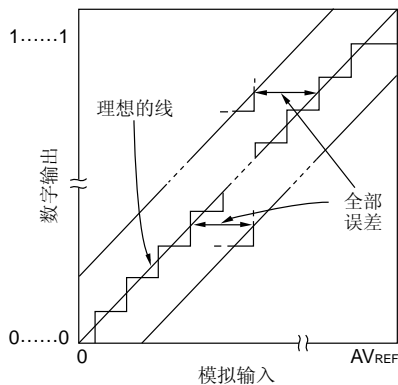
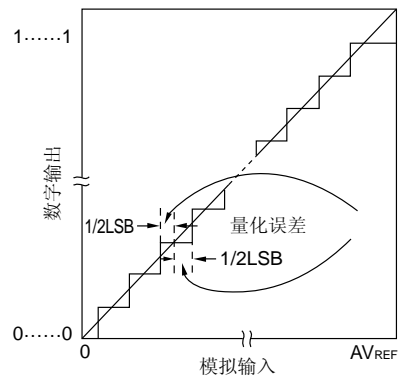


图 15-16. 量化误差



(4) 零幅度误差

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) 满幅度误差

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。它表示当零幅度误差和满幅度误差为 0 时，实际测量值和理想直线之间的最大误差值。

(7) 差分线性误差

当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 15-17. 零幅度误差

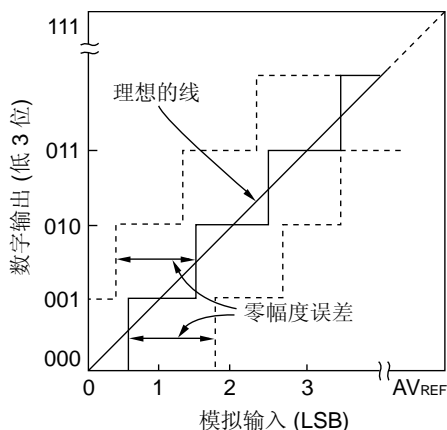


图 15-18. 满幅度误差

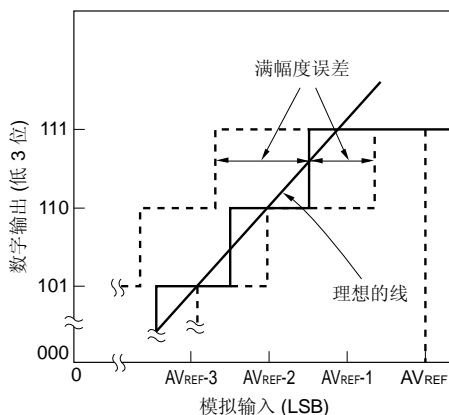


图 15-19. 积分线性误差

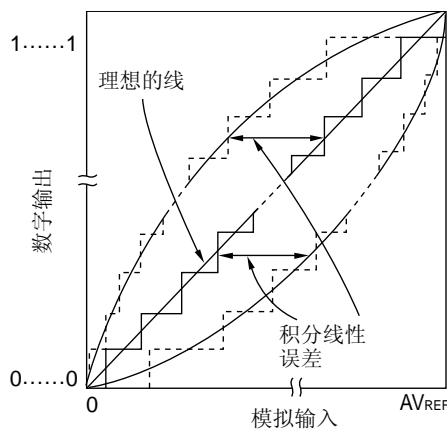
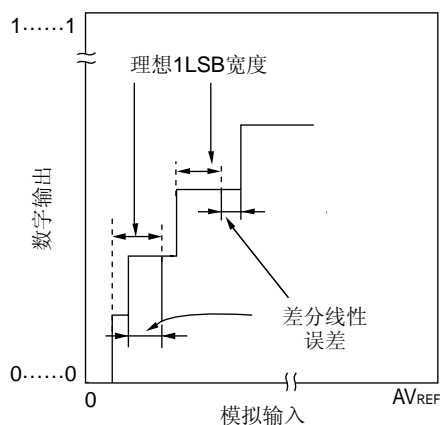


图 15-20. 差分线性误差

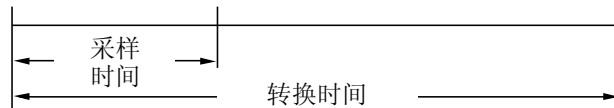


(8) 转换时间

这表示从采样启动到数字输出被获得的时间。在特性表中，采样时间被包含在转换时间中。

(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。



15.7 A/D转换器的注意事项

(1) 在待机模式下的工作电流

A/D 转换器在待机模式下停止工作。此时，通过清除 A/D 转换器模式寄存器（ADM）的第 7 位（ADCS）和为 0，工作电流可以减少（参见图 15-2）。

(2) ANI0 到 ANI7 的输入范围

遵守 ANI0 到 ANI7 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）被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）和读取 ADCR 之间的冲突

ADCR 读取具有优先权。读取操作后，新的转换值被写入 ADCR。

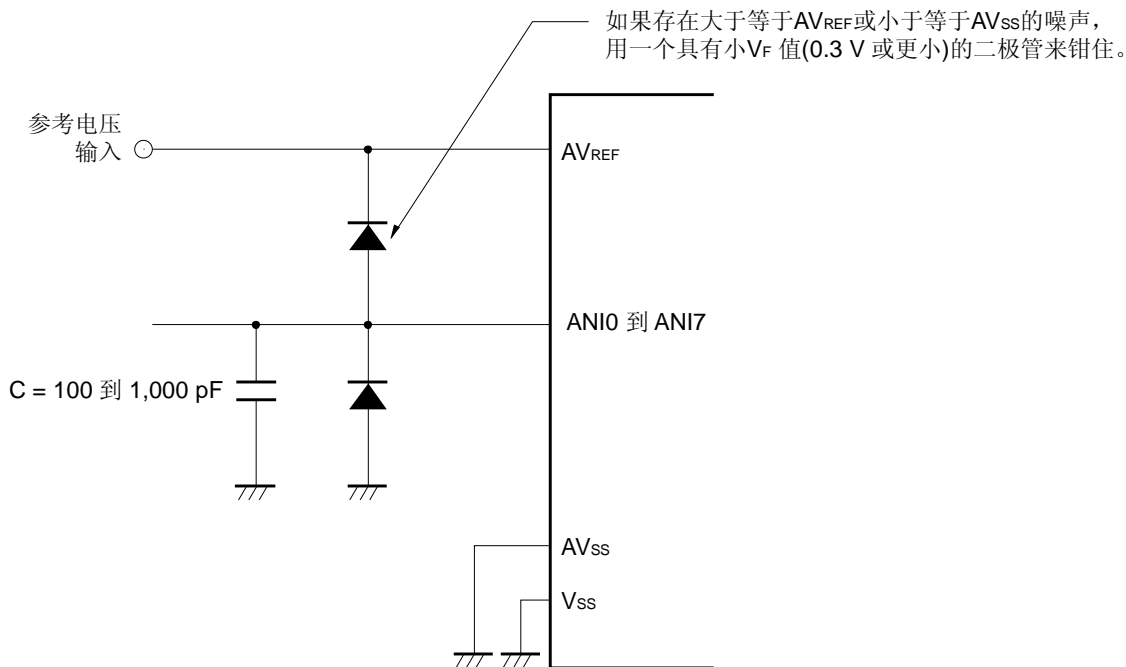
<2> 转换结束时 ADCR 写和 A/D 转换器模式寄存器（ADM）写或模拟输入通道指定寄存器（ADS）之间的冲突

ADM 或 ADS 写具有优先权。ADCR 写入不被执行，转换结束中断信号（INTAD）也不产生。

(4) 噪声对策

要维持 10 位分辨率，必须注意 AV_{REF} 引脚和 ANI0 到 ANI7 上输入的噪声。因为与模拟输入的输出阻抗成比例增加，建议外部连接一个电容来减小噪声，如图 15-21 所示。

图 15-21. 模拟输入引脚连接



(5) ANI0/P20 到 ANI7/P27

- <1> 模拟输入引脚（ANI0 到 ANI7）也被用作输入端口引脚（P20 到 P27）。
当使用 ANI0 到 ANI7 的任意引脚的 A/D 转换被执行时，转换过程中不要访问端口 2；否则，转换分辨率可能会退化。
- <2> 如果一个数字脉冲被应用于当前用作 A/D 转换的引脚的邻近引脚，由于耦合噪声，期望的 A/D 转换值可能不会被获得。因此，不要在处于 A/D 转换的引脚的邻近引脚上应用脉冲。

(6) ANI0 到 ANI7 引脚的输入阻抗

在该 A/D 转换器中，内部采样电容被充电并且采样被执行。
因为在采样期间以外只有漏电流流动并且在采样期间对电容充电的电路也流动，输入阻抗波动并且没有办法。
然而，要执行足够的采样，建议使模拟输入源的输出阻抗为 10 kΩ 或更低，或者连接一个大约 100 pF 的电容器到 ANI0 – ANI7 引脚（参见图 15-21）。

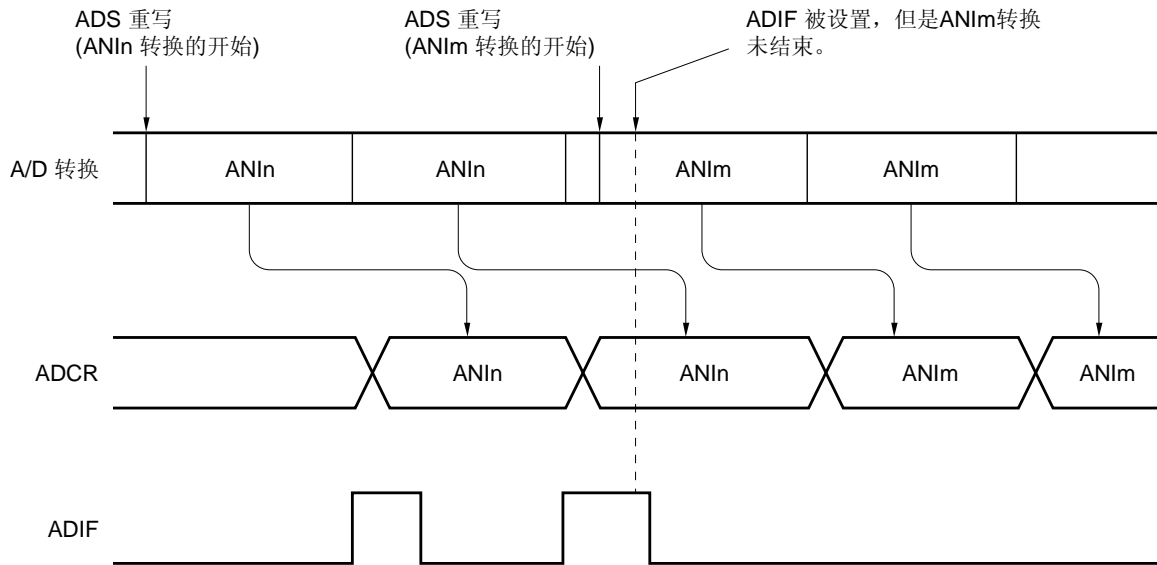
(7) AVREF 引脚输入阻抗

一个几十 kΩ 的串联电阻串被连接到 AVREF 和 AVSS 引脚之间。
因此，如果参考电压源的输出阻抗很高，这将导致 AVREF 和 AVSS 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）

即使模拟输入通道指定寄存器（ADS）被更改，中断请求标志（ADIF）也不会被清除。
因此，如果在 A/D 转换过程中模拟输入引脚被更改，更改前的模拟输入的 A/D 转换结果和 ADIF 可能在 ADS 刚刚被重新写入前被设置。这时需要注意，因为当 ADIF 在 ADS 被重新写入后立即被读取时，尽管更改后的模拟输入的 A/D 转换没有结束，ADIF 也会被设置。
当 A/D 转换被停止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 15-22. A/D 转换结束中断请求产生时序



- 备注
1. n = 0 到 7
 2. m = 0 到 7

(9) 在 A/D 转换刚刚开始后的转换结果

如果在 ADCS2 位被设置为 1 后的 $1\ \mu\text{s}$ 内 ADCS 位被设置为 1 或者如果 ADCS2 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) 读取操作

当对 A/D 转换器模式寄存器 (ADM) 和模拟输入通道指定寄存器 (ADS) 执行写入操作时，ADCR 的内容可能变为不确定。在转换完成后并且写入 ADM 和 ADS 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

模拟输入块的等效电路被表示如下

图 15-23. ANIn 引脚的内部等效电路

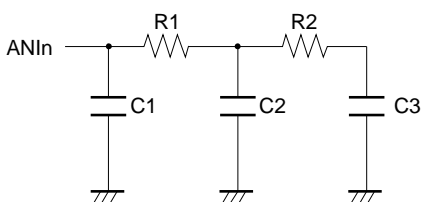


表 15-4. 等效电路的电阻和电容值 (参考值)

AVREF	R1	R2	C1	C2	C3
4.5 V	4 k Ω	2.7 k Ω	8 pF	1.4 pF	.6 pF

- 备注
1. 表 15-4 中显示的电阻和电容值是不被保证的值。
 2. n = 0 到 7

16.1 串行接口UART00 的功能

串行接口 UART00 有以下两种模式。

(1) 操作停止模式

这个模式在串行通信未被执行时使用，并且可以减少电源消耗。

关于详细情况，见 16.4.1 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。

关于详细情况，见 16.4.2 异步串行接口 (UART) 模式和 16.4.3 专用波特率产生器。

- 两个引脚配置
TxD00: 发送数据输出引脚
RxD00: 接收数据输入引脚
- 通信数据的数据长度可以从 7 或 8 位中选择。
- 专用的片上 5 位波特率产生器允许任意波特率被设置。
- 发送和接收可以单独执行。
- 四个操作时钟输入可以选择
- 固定为 LSB 在前通信

注意事项 1. 如果提供给串行接口 UART00 的源时钟未被停止（例如，在 HALT 模式下），正常操作继续。如果提供给串行接口 UART00 的源时钟被停止（例如，在 STOP 模式下），每个寄存器停止工作，并且保持时钟刚刚停止前的值。TxD00 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，操作不被保证。因此，复位电路来使 POWER00 = 0、RXE00 = 0 和 TXE00 = 0。

2. 设置 POWER00 = 1 然后设置 TXE00 = 1（发送）或 RXE00 = 1（接收）来开始通信。

3. TXE00 和 RXE00 与由 BRGC00 设置的基时钟 (f_{XCLK0}) 同步。要再次允许发送或接收，在 TXE00 或 RXE00 已经被清除为 0 后至少两个基时钟周期后设置 TXE00 或 RXE00 为 1。如果 TXE00 或 RXE00 在基时钟的两个周期内被设置，发送电路或接收电路可能没有初始化。

4. 在设置 TXE00 = 1 后至少两个基时钟 (f_{XCLK0}) 周期后设置发送数据到 Txs00。

<R>

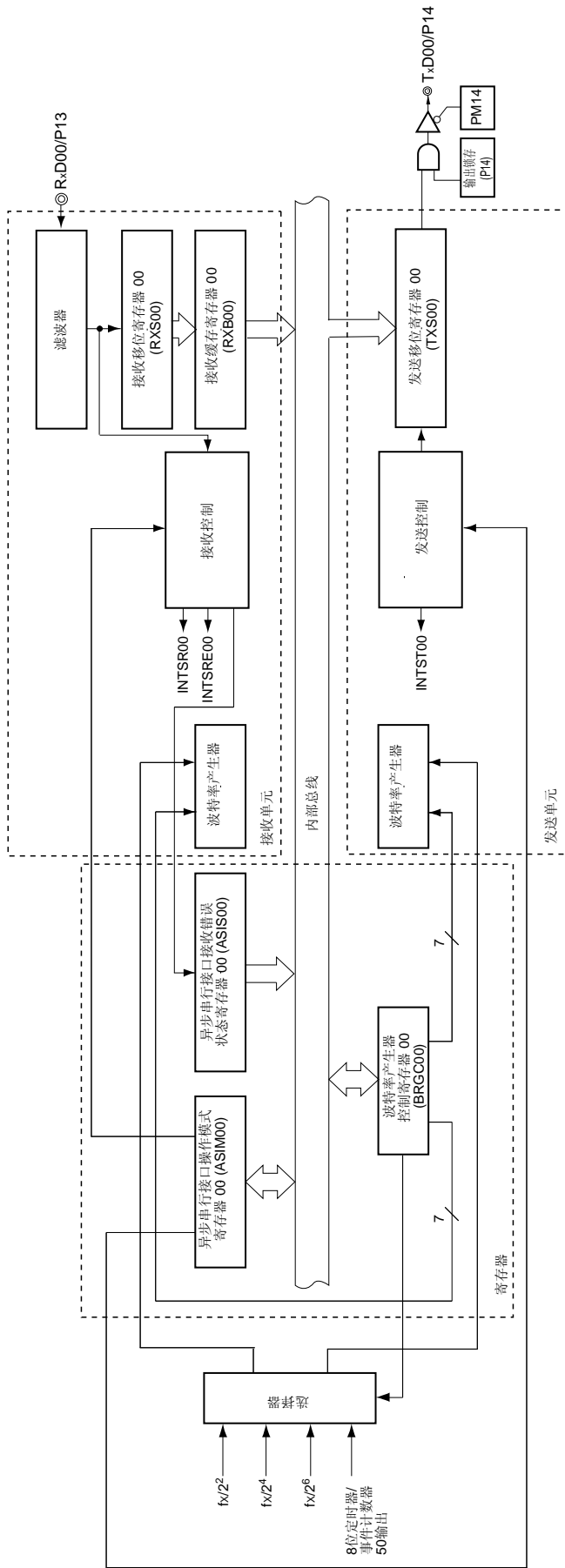
16.2 串行接口UART0 的配置

串行接口 UART0 由以下硬件组成。

表 16-1. 串行接口 UART0 的配置

项目	配置
寄存器	接收缓存寄存器 00 (RXB00) 接收移位寄存器 00 (RXS00) 发送移位寄存器 00 (TXS00)
控制寄存器	异步串行接口操作模式寄存器 00 (ASIM00) 异步串行接口接收错误状态寄存器 00 (ASIS00) 波特率产生器控制寄存器 00 (BRGC00) 端口模式寄存器 1 (PM1) 端口寄存器 1 (P1)

图 16-1. 串行接口 UART0 的框图



(1) 接收缓存寄存器 00 (RXB00)

该 8 位寄存器存储由接收移位寄存器 00 (RXS00) 转换来的并行数据。

每次数据的 1 字节被接收，新接收的数据从接收移位寄存器 00 (RXS00) 转移到该寄存器。

如果数据长度被设置为 7 位，接收数据被转移到 RXB00 的第 0 到 6 位并且 RXB00 的 MSB 总是 0。

如果溢出错误 (OVE00) 发生，接收数据不会被转移到 RXB00。

RXB00 可以由 8 位存储器操作指令读取。数据不能被写入该寄存器。

RESET 输入或 POWER0 = 0 设置该寄存器为 FFH。

(2) 接收移位寄存器 00 (RXS00)

该寄存器将输入到 RxD00 引脚的串行数据转换为并行数据。

RXS00 不能被程序直接操作。

(3) 发送移位寄存器 00 (TXS00)

该寄存器用来设置发送数据。当数据被写入 TXS00 时，发送被启动，并且串行数据从 TxD00 引脚被发送。TXS00 可以由 8 位存储器操作指令来写入。该寄存器不能被读取。

RESET 输入、POWER0 = 0 或 TXE00 = 0 设置该寄存器为 FFH。

<R>

- 注意事项**
1. 在设置 TXE00 = 1 后至少两个基时钟 (f_{CLK0}) 周期后设置发送数据到 TXS00。
 2. 在发送完成中断信号 (INTST00) 产生前，不要向 TXS00 写入下一个发送数据。

16.3 控制串行接口UART00 的寄存器

串行接口 UART00 由以下五个寄存器控制。

- 异步串行接口操作模式寄存器 00 (ASIM00)
- 异步串行接口接收错误状态寄存器 00 (ASIS00)
- 波特率产生器控制寄存器 00 (BRGC00)
- 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 00 (ASIM00)

该 8 位寄存器控制串行接口 UART00 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。

$\overline{\text{RESET}}$ 输入设置该寄存器为 01H。

图 16-2. 异步串行接口操作模式寄存器 00 (ASIM00) 的格式 (1/2)

地址: FF70H 复位后: 01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM00	POWER00	TXE00	RXE00	PS001	PS000	CL00	SL00	1

POWER00	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

TXE00	允许/禁止发送
0	禁止发送（同步复位发送电路）。
1	允许发送

RXE00	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 POWER00 = 0 时, RxD00 引脚的输入固定为高电平。
 2. 异步串行接口接收错误状态寄存器 00 (ASIS00)、发送移位寄存器 00 (TXS00) 和接收缓存寄存器 00 (RXB00) 被复位。

图 16-2. 异步串行接口操作模式寄存器 00 (ASIM00) 的格式 (2/2)

PS001	PS000	发送操作	接收操作
0	0	不输出奇偶位。	无奇偶位接收
0	1	输出 0 奇偶。	以 0 奇偶接收 ^注
1	0	输出奇数奇偶。	按照奇数奇偶判断。
1	1	输出偶数奇偶。	按照偶数奇偶判断。

CL00	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

SL00	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“以 0 奇偶接收”被选择，奇偶位不被判断。因此，异步串行接口接收错误状态寄存器 00 (ASIS00) 的第 2 位 (PE00) 不被设置并且错误中断不发生。

- 注意事项**
1. 开始时，设置 POWER00 为 1 然后设置 TXE00 为 1。要停止发送，清除 TXE00 为 0，并且然后清除 POWER00 为 0。
 2. 开始时，设置 POWER00 为 1 然后设置 RXE00 为 1。要停止接收，清除 RXE00 为 0，并且然后清除 POWER00 为 0。
 3. 当一个高电平输入到 RxD00 引脚时，设置 POWER00 为 1 然后设置 RXE00 为 1。如果当低电平被输入时，POWER00 被设置为 1 并且 RXE00 被设置为 1，接收被启动。
 4. TXE00 和 RXE00 与由 BRGC00 设置的基时钟 (f_{CLK0}) 同步。要再次允许发送或接收，在 TXE00 或 RXE00 已经被清除为 0 后至少两个基时钟周期后设置 TXE00 或 RXE00 为 1。如果 TXE00 或 RXE00 在基时钟的两个周期内被设置，发送电路或接收电路可能没有初始化。
 5. 在设置 TXE00 = 1 后至少两个基时钟 (f_{CLK0}) 周期后设置发送数据到 TXS00。
 6. 在重新写入 PS001、PS000 和 CL00 位前，清除 TXE00 和 RXE00 位为 0。
 7. 当重新写入 SL00 位时，确保 TXE00 = 0。接收总是按照“停止位个数 = 1”来执行，因此不受 SL00 位的设置值影响。
 8. 确保设置第 0 位为 1。

<R>

(2) 异步串行接口接收错误状态寄存器 00 (ASIS00)

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 (PE00, FE00, OVE00)。

该寄存器可以由 8 位存储器操作指令只读。

RESET 输入、ASIM00 的第 7 位 (POWER00) = 0 或者 ASIM00 的第 5 位 (RXE00) = 0 清除该寄存器为 00H。并且该寄存器的读取也将该寄存器清除为 00H。

图 16-3. 异步串行接口接收错误状态寄存器 00 (ASIS00) 的格式

地址: FF73H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIS00	0	0	0	0	0	PE00	FE00	OVE00

PE00	表明奇偶错误的状态标志
0	如果 POWER00 = 0 并且 RXE00 = 0, 或者然后 ASIS00 寄存器被读取
1	如果发送数据的奇偶位与接收完成时的奇偶位不匹配

FE00	表明帧错误的状态标志
0	如果 POWER00 = 0 并且 RXE00 = 0, 或者然后 ASIS00 寄存器被读取
1	如果接收完成时停止位未被检测到

OVE00	表明溢出错误的状态标志
0	如果 POWER00 = 0 并且 RXE00 = 0, 或者然后 ASIS00 寄存器被读取
1	如果接收数据被设置到 RXB00 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. PE00 位的操作依赖于异步串行接口操作模式寄存器 00 (ASIM00) 的 PS001 和 PS000 位的设置值而不同。
 2. 只有接收数据的第一位被作为停止位检查, 而不管停止位的个数。
 3. 如果溢出错误发生, 下一个接收数据不会被写入接收缓冲区寄存器 00 (RXB00), 而是被忽略。
 4. 如果数据从 ASIS00 被读取, 一个等待周期被产生。关于详细情况, 见第三十章 等待注意事项。

(3) 波特率产生器控制寄存器 00 (BRGC00)

该寄存器选择串行接口 UART0 的基时钟和 5 位计数器的分频值。

BRGC00 可以由 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入设置该寄存器为 1FH。

图 16-4. 波特率产生器控制寄存器 00 (BRGC00) 的格式

地址: FF71H 复位后: 1FH R/W

符号	7	6	5	4	3	2	1	0
BRGC00	TPS001	TPS000	0	MDL004	MDL003	MDL002	MDL001	MDL000

TPS001	TPS000	基时钟 (f _{CLK0}) 选择 ^{注1}
0	0	TM50 输出 ^{注2}
0	1	f _x /2 ² (5 MHz)
1	0	f _x /2 ⁴ (1.25 MHz)
1	1	f _x /2 ⁶ (312.5 kHz)

MDL004	MDL003	MDL002	MDL001	MDL000	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 _{CLK0} /8
0	1	0	0	1	9	f _{CLK0} /9
0	1	0	1	0	10	f _{CLK0} /10
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	0	1	1	27	f _{CLK0} /27
1	1	1	0	0	28	f _{CLK0} /28
1	1	1	0	1	29	f _{CLK0} /29
1	1	1	1	0	30	f _{CLK0} /30
1	1	1	1	1	31	f _{CLK0} /31

- 注
1. 确保设置基时钟使其满足以下条件。
 - V_{DD} = 4.0 到 5.5 V: 基时钟 ≤ 10 MHz
 2. 当 TM50 输出被选作计数时钟时, 遵守以下条件。
 - PWM 模式 (TMC506 = 1)

设置时钟使占空比为 50% 并且事先启动 8 位定时器/事件计数器 50 的操作。
 - TM50 和 CR50 匹配时 (TMC506 = 0) 进入清除&开始模式

允许定时器 F/F 反转操作 (TMC501 = 1) 并且事先启动 8 位定时器/事件计数器 50 的操作。

无论处于何种模式, 不需要允许 TO50 引脚作为定时器输出引脚 (TMC 寄存器的第 00 位 (TOE50) 可能是 0 或 1)。

- 注意事项**
1. 当内部振荡时钟被选作 CPU 的源时钟时，内部振荡器的时钟被分频并作为计数时钟被提供。如果基时钟为内部振荡时钟，串行接口 UART00 的操作不被保证。
 2. 当重新写入 MDL004 到 MDL000 位时，确保 ASIM00 寄存器的第 6 位 (TXE00) 和第 5 位 (RXE00) = 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 TPS001 和 TPS000 位选择的基时钟的频率
 2. f_x: X1 输入时钟振荡频率
 3. k: 由 MDL004 到 MDL000 位设置的值 (k = 8, 9, 10, ..., 31)
 4. x: 不关注
 5. 括号中的数字应用于 f_x = 20 MHz 的操作。
 6. 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 P14/TxD00 引脚作为串行接口数据输出时，清除 PM14 为 0 并设置输出锁存 P14 为 1。

当使用 P13/RxD00 引脚作为串行接口数据输入时，设置 PM13 为 1。P13 的输出锁存这时可能是 0 或 1。

PM1 可以由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入设置该寄存器为 FFH。

图 16-5. 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 I/O 模式选择 (n = 0 到 7)							
0	输出模式 (输出缓存打开)							
1	输入模式 (输出缓存关闭)							

16.4 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

16.4.1 操作停止模式

在这种模式下，串行通信不能被执行，因此可以减少电源消耗。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 ASIM00 的第 7、6 和 5 位 (POWER00、TXE00 和 RXE00) 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 00 (ASIM00) 来设置。

ASIM00 可以由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入设置该寄存器为 01H。

地址: FF70H 复位后: 01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM00	POWER00	TXE00	RXE00	PS001	PS000	CL00	SL00	1
POWER00	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} 。							
TXE00	允许/禁止发送							
0	禁止发送 (同步复位发送电路)。							
RXE00	允许/禁止接收							
0	禁止接收 (同步复位接收电路)。							

- 注
1. 当 POWER00 = 0 时，RxD00 引脚的输入固定为高电平。
 2. 异步串行接口接收错误状态寄存器 00 (ASIS00)、发送移位寄存器 00 (TXS00) 和接收缓存寄存器 00 (RXB00) 被复位。

注意事项 在清除 TXE00 和 RXE00 为 0 后清除 POWER00 为 0 来设置操作停止模式。
要开始操作，设置 POWER00 为 1 然后设置 TXE00 或 RXE00 为 1。

备注 要作为通用端口引脚使用 RxD00/P13 和 TxD00/P14 引脚，见 第四章 端口功能。

16.4.2 异步串行接口 (UART) 模式

在这种模式下，1 字节的数据在起始位后被发送/接收，并且全双工操作可以被执行。它包含一个专用的波特率产生器，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 00 (ASIM00)
- 异步串行接口接收错误状态寄存器 00 (ASIS00)
- 波特率产生器控制寄存器 00 (BRGC00)
- 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 BRGC00 寄存器 (参见图 16-4)。
- <2> 设置 ASIM00 寄存器的第 1 到 4 位 (SL00, CL00, PS000 和 PS001) (见 图 16-2)。
- <3> 设置 ASIM00 寄存器的第 7 位 (POWER00) 为 1。
- <4> 设置 ASIM00 寄存器的第 6 位 (TXE00) 为 1。→ 发送允许。
设置 ASIM00 寄存器的第 5 位 (RXE00) 为 1。→ 接收允许。
- <R> <5> 设置<4>后至少两个时钟周期写入数据到 TXS00 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 16-2. 寄存器设置和引脚之间的关系

POWER00	TXE00	RXE00	PM14	P14	PM13	P13	UART00 操作	引脚功能	
								TxD00/ P14	RxD00/P13
0	0	0	x [#]	x [#]	x [#]	x [#]	停止	P14	P13
1	0	1	x [#]	x [#]	1	x	接收	P14	RxD00
	1	0	0	1	x [#]	x [#]	发送	TxD00	P13
	1	1	0	1	1	x	发送/ 接收	TxD00	RxD00

注 可以被设置为端口功能。

备注

x: 不关注

POWER00: 异步串行接口操作模式寄存器 00 (ASIM00) 的第 7 位

TXE00: ASIM00 的第 6 位

RXE00: ASIM00 的第 5 位

PM1x: 端口模式寄存器

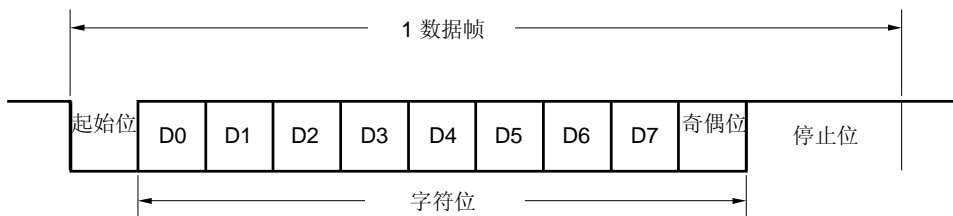
P1x: 端口输出锁存

(2) 通信操作

(a) 正常发送/接收数据的格式和波形举例

图 16-6 和 16-7 表示正常发送/接收数据的格式和波形举例。

图 16-6. 正常 UART 发送/接收数据的格式



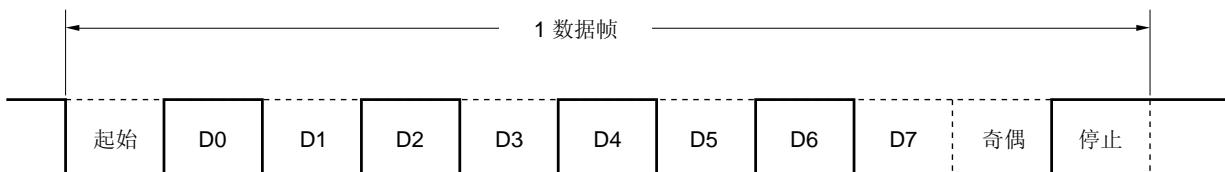
一个数据帧由以下位组成。

- 起始位 ... 1 位
- 符号位 ... 7 或 8 位 (LSB 在前)
- 奇偶位 ... 偶数奇偶, 奇数奇偶, 0 奇偶或无奇偶
- 停止位 ... 1 或 2 位

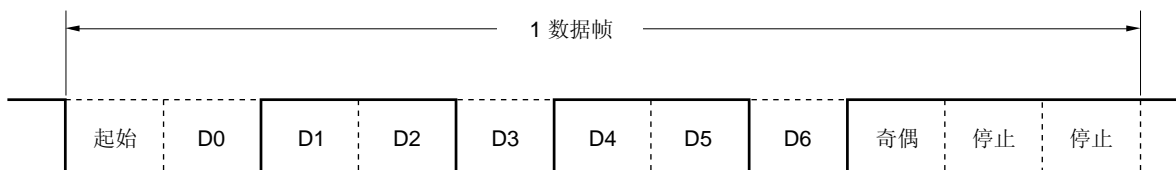
一个数据帧中的符号位长度、奇偶和停止位长度由异步串行接口操作模式寄存器 00 (ASIM00) 来指定。

图 16-7. 正常 UART 发送/接收数据波形举例

1. 数据长度: 8 位, 奇偶: 偶数奇偶, 停止位: 1 位, 通信数据: 55H



2. 数据长度: 7 位, 奇偶: 奇数奇偶, 停止位: 2 位, 通信数据: 36H



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



(b) 奇偶类型和操作

奇偶位用来检测通信数据中的位错误。通常，同样奇偶位类型在发送和接收端被使用。使用偶数奇偶和奇数奇偶，一个 1 位（奇数个数）错误可以被检测。使用 0 奇偶和无奇偶，错误不能被检测。

(i) 偶数奇偶

• 发送

包含奇偶位的发送数据被控制以使为“1”的位个数是偶数。

奇偶位的值如下所示。

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

包含奇偶位的接收数据中为“1”的位的个数被计数。如果它为奇数，一个奇偶错误发生。

(ii) 奇数奇偶

• 发送

与偶数奇偶不同，包含奇偶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

包含奇偶位的接收数据中为“1”的位的个数被计数。如果它为偶数，一个奇偶错误发生。

(iii) 0 奇偶

当数据被发送时，奇偶位被清除为 0，不管发送数据是什么。

当数据被接收时，奇偶位不被检测。因此，不管奇偶位是“0”还是“1”，奇偶错误都不会发生。

(iv) 无奇偶

无奇偶位附加到发送数据上。

当数据被接收时，接收方假设没有奇偶位。因为没有奇偶位，奇偶错误不会发生。

(c) 发送

当异步串行接口操作模式寄存器 00 (ASIM00) 的第 7 位 (POWER00) 被设置为 1 时, TxD00 引脚输出一个高电平。如果 ASIM00 的第 6 位 (TXE00) 被设置为 1, 发送允许。通过在设置 TXE00 = 1 后至少两个基时钟 (f_{CLK0}) 后写入发送数据到发送移位寄存器 00 (TXS00), 发送可以被启动。起始位、奇偶位和停止位自动附加到数据上。

当发送被启动时, 起始位从 TxD00 引脚被输出, 并且剩余数据按照 LSB 在前的顺序接着被输出。当发送完成时, ASIM00 设置的奇偶和停止位被附加到数据上并且发送完成中断请求 (INTST00) 被产生。

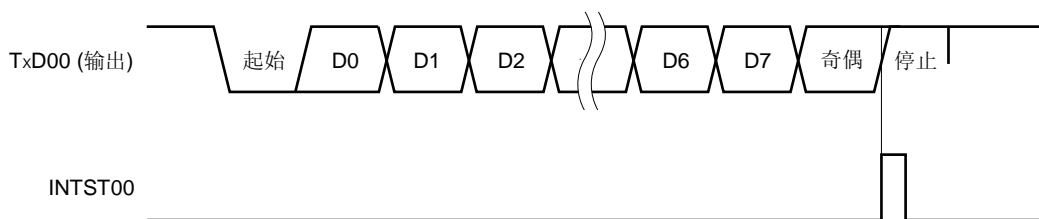
在下一个要被发送的数据被写入 TXS00 之前, 发送被停止。

图 16-8 表示发送完成中断请求 (INTST00) 的时序。只要最后的停止位被输出, 这个中断就发生。

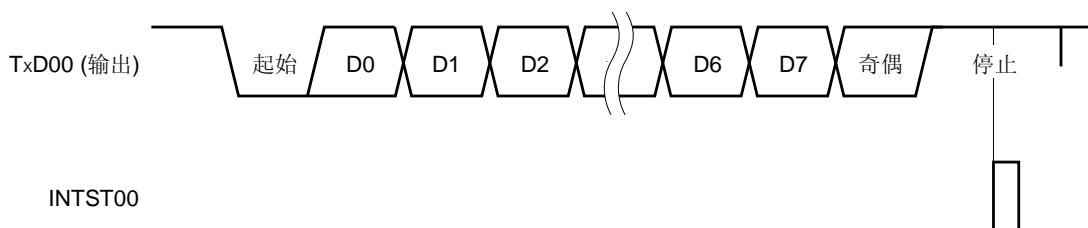
注意事项 在发送数据被写入 TXS00 后, 在发送完成中断信号 (INTST00) 产生之前, 不要写入下一个发送数据。

图 16-8. 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

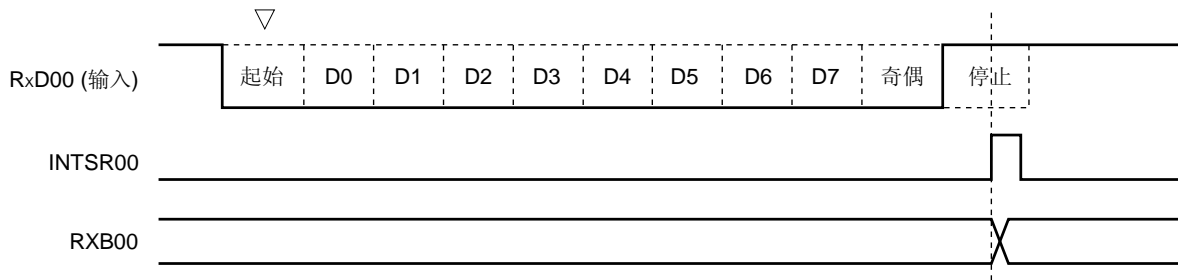
当异步串行接口操作模式寄存器 00 (ASIM00) 的第 7 位 (POWER00) 被设置为 1 然后 ASIM00 的第 5 位 (RXE00) 被设置为 1 时, 接收被允许并且 RxD00 引脚的输入被采样。

当 RxD00 引脚输入的下降沿被检测时, 波特率产生器的 5 位计数器开始计数。当波特率产生器控制寄存器 00 (BRGC00) 的设置值计满时, RxD00 引脚输入再次被采样 (图 16-9 中的 ▽)。如果 RxD00 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 00 (RXS00) 中。当停止位被接收时, 接收完成中断 (INTSR00) 产生并且 RXS00 的数据被写入接收缓冲区寄存器 00 (RXB00)。然而, 如果一个溢出错误 (OVE00) 发生, 接收数据不会被写入 RXB00。

当接收进行时, 即使奇偶错误 (PE00) 发生, 接收会继续执行到停止位的接收位置, 并且一个接收错误中断 (INTSR00) 在接收完成时被产生。

图 16-9. 接收完成中断请求时序



- 注意事项**
1. 即使接收错误发生, 也要确保读取接收缓存寄存器 00 (RXB00)。否则, 当下一个数据被接收时, 一个溢出错误将发生, 并且接收错误状态将持续。
 2. 接收总是按照“停止位数 = 1”来执行。第二个停止位被忽略。
 3. 读取 RXB00 前确保读取异步串行接口接收错误状态寄存器 00 (ASIS00)。

(e) 接收错误

接收期间可能发生三种错误：奇偶错误、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 00 (ASIS00) 的错误标志被设置，一个接收错误中断 (INTSR00) 产生。

接收期间哪一个错误发生可以通过在接收错误中断 (INTSR00) 服务程序中读取 ASIS00 的内容来鉴别 (见图 16-3)。

当 ASIS00 被读取时，ASIS00 的内容被复位为 0。

表 16-3. 接收错误的原因

接收错误	原因
奇偶错误	发送指定的奇偶与接收数据的奇偶不匹配。
帧错误	停止位未被检测到。
溢出错误	下一个数据的接收在数据从接收缓冲区寄存器 00 (RXB00) 读取前完成。

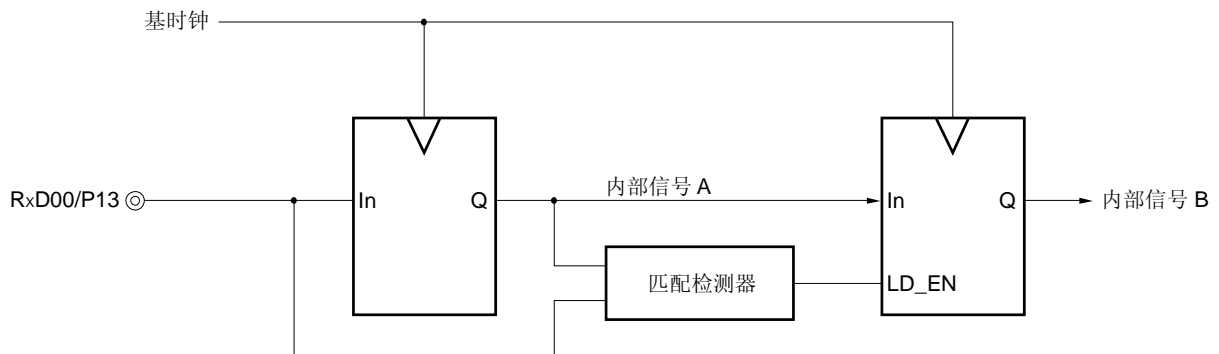
(f) 接收数据的噪声滤波器

RXD00 信号由预处理模块输出的基时钟采样。

如果两个采样值相同，匹配检测的输出更改，并且数据按照输入数据被采样。

因为电路被配置为图 16-10 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 16-10. 噪声滤波器电路



16.4.3 专用波特率产生器

专用波特率产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) 波特率产生器的配置

- 基时钟

当异步串行接口操作模式寄存器 00 (ASIM00) 的第 7 位 (POWER00) 被设置为 1 时，由波特率产生器控制寄存器 00 (BRGC00) 的第 7 和 6 位 (TPS001 和 TPS000) 选择的时钟提供给每个模块。这个时钟叫做基时钟，其频率称为 f_{XCLK0} 。当 $POWER00 = 0$ 时，基时钟固定为低电平。

- 发送计数器

当串行接口操作模式寄存器 00 (ASIM00) 的第 7 位 (POWER00) 或第 0 位 (TXE00) 为 0 时，这个计数器停止操作，清除为 0。

当 $POWER00 = 1$ 并且 $TXE00 = 1$ 时，它开始计数。

当发送的第一个数据被写入发送移位寄存器 00 (TXS00) 时，计数器被清除为 0。

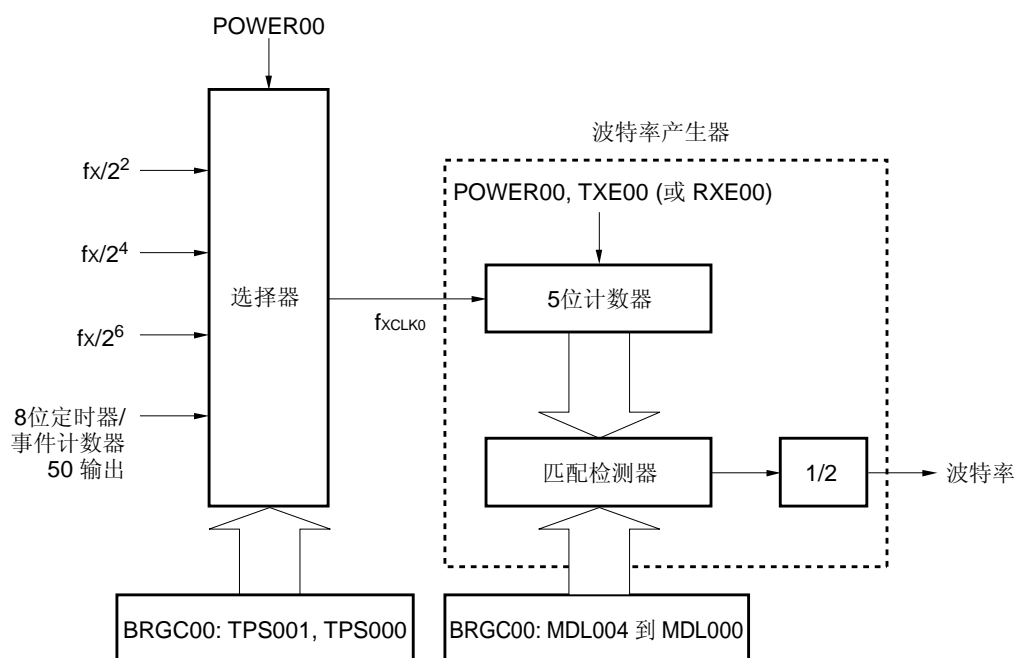
- 接收计数器

当串行接口操作模式寄存器 00 (ASIM00) 的第 7 位 (POWER00) 或第 5 位 (RXE00) 为 0 时，这个计数器停止操作，清除为 0。

当起始位被检测时，它开始计数。

在一帧接收后，下一个起始位检测前，计数器停止操作。

图 16-11. 波特率产生器的配置



备注

- POWER00: 异步串行接口操作模式寄存器 00 (ASIM00) 的第 7 位
- TXE00: ASIM00 的第 6 位
- RXE00: ASIM00 的第 5 位
- BRGC00: 波特率产生器控制寄存器 00

(2) 串行时钟的产生

串行时钟可以通过使用波特率产生器控制寄存器 00 (BRGC00) 来产生。

通过使用 BRGC00 的第 7 和 6 位 (TPS001 和 TPS000) 来选择输入到 5 位计数器的时钟。

BRGC00 的第 4 到 0 位 (MDL004 到 MDL000) 可以用来选择 5 位计数器的分频值。

(a) 波特率

波特率可以由以下表达式计算。

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

f_{XCLK0} : 由 BRGC00 寄存器的 TPS001 和 TPS000 位选择的基时钟的频率

k: 由 BRGC00 寄存器的 MDL004 到 MDL000 位设置的值 (k = 8, 9, 10, ..., 31)

(b) 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, 保持波特率误差在接收端允许的误差范围内。
 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显示的范围。

举例:

基时钟频率 = 2.5 MHz = 2,500,000 Hz
 BRGC00 寄存器的 MDL004 到 MDL000 位的设置值 = 10000B (k = 16)
 目标波特率 = 76,800 bps

$$\begin{aligned} \text{波特率} &= 2.5 \text{ M}/(2 \times 16) \\ &= 2,500,000/(2 \times 16) = 78,125 [\text{bps}] \end{aligned}$$

$$\begin{aligned} \text{误差} &= (78,125/76,800 - 1) \times 100 \\ &= 1.725 [\%] \end{aligned}$$

(3) 设置波特率举例

表 16-4. 波特率产生器的设置数据

波特率 [bps]	fx = 20.0 MHz				fx = 16.0 MHz			
	TPS001, TPS000	k	计算的值	ERR[%]	TPS001, TPS000	k	计算的值	ERR[%]
2400	-	-	-	-	-	-	-	-
4800	-	-	-	-	3	26	4808	0.16
9600	3	16	9766	1.73	3	13	9615	0.16
10400	3	15	10417	0.16	3	12	10417	0.16
19200	3	8	19531	1.73	2	26	19231	0.16
31250	2	20	31250	0	2	16	31250	0
38400	2	16	39063	1.73	2	13	38462	0.16
76800	2	8	78125	1.73	1	26	76923	0.16
115200	1	22	113636	-1.36	1	17	117647	2.12
153600	1	16	156250	1.73	1	13	153846	0.16
230400	1	11	227273	-1.36	-	-	-	-

备注 TPS001, TPS000: 波特率产生器控制寄存器 00 (BRGC00) 的第 7 和 6 位 (基时钟 (fxCLK0) 的设置)

k: 由 BRGC00 的 MDL004 到 MDL000 位设置的值 (k = 8, 9, 10, ..., 31)

fx: X1 输入时钟振荡频率

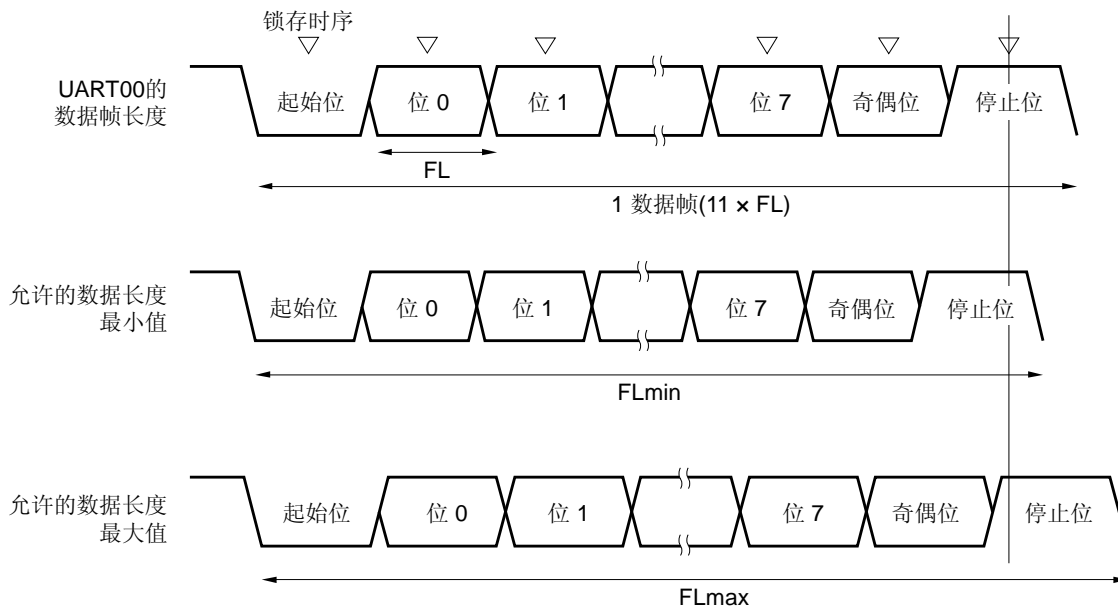
ERR: 波特率误差

(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 16-12. 接收期间允许的波特率范围



如图 16-12 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产生器控制寄存器 00 (BRGC00) 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，数据可以被正确接收。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: BRGC00 的设置值
 FL: 1 位数据长度
 锁存时序的余量: 2 时钟

$$\text{最小允许的数据帧长度: } FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

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，最大允许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 16-5. 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
8	+3.53%	-3.61%
16	+4.14%	-4.19%
24	+4.34%	-4.38%
31	+4.44%	-4.47%

- 备注
1. 接收的允许误差依赖于于一帧的位的个数、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: BRGC00 的设置值

17.1 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 I/O 模式

(1) 操作停止模式

该模式在串行通信未被执行时使用，并且可以减少电源消耗。

关于详细情况，见 **17.4.1 操作停止模式**。

(2) 3 线串行 I/O 模式 (MSB/LSB 在前可选)

该模式用于使用三根线通信 8 位数据：串行时钟线 ($\overline{\text{SCK10}}$) 和两条串行数据线 (SI10 和 SO10)，传输 8 位数据。

在 3 线串行 I/O 模式下，数据通信的处理时间可以被缩短，因为发送和接收可以同时执行。

此外，8 位数据传输时是 MSB 在前还是 LSB 在前可以指定，所以该接口可以连接到任意设备。

3 线串行 I/O 模式用于连接带时钟驱动串行接口的外围 ICs 和显示控制器。

关于详细情况，见 **17.4.2 3 线串行 I/O 模式**。

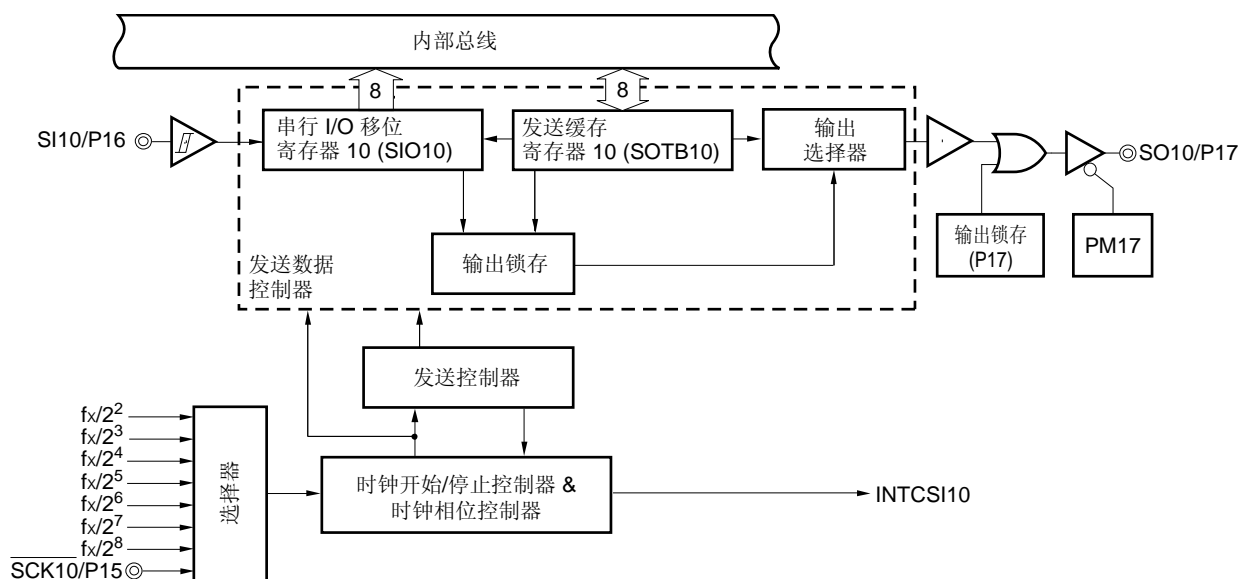
17.2 串行接口CSI10 的配置

串行接口 CSI10 由以下硬件组成。

表 17-1. 串行接口 CSI10 的配置

项目	配置
寄存器	发送缓存寄存器 10 (SOTB10) 串行 I/O 移位寄存器 10 (SIO10) 发送控制器 时钟开始/停止控制器&时钟相位控制器
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 17-1. 串行接口 CSI10 的框图



(1) 发送缓存寄存器 10 (SOTB10)

该寄存器设置发送数据。

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, 发送/接收通过向 SOTB10 写入数据来启动。

写入 SOTB10 的数据由串行 I/O 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。

$\overline{\text{RESET}}$ 输入使该寄存器不确定。

注意事项 当 $\text{CSOT10} = 1$ (串行通信期间) 时, 不要访问 SOTB10。

(2) 串行 I/O 移位寄存器 10 (SIO10)

这是一个将数据从并行数据转换为串行数据或者反过来的 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, 接收被启动。

接收期间, 数据从串行输入引脚 (SI10) 读取到 SIO10。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

注意事项 当 $\text{CSOT10} = 1$ (串行通信期间) 时, 不要访问 SIO10。

17.3 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。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 17-2. 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 I/O 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 (发送禁止)。							
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	通信被停止。							
1	通信进行中。							

- 注
1. 第 0 位是只读位。
 2. 当使用 P15/ $\overline{\text{SCK10}}$ 、P16/SI10 和 P17/SO10/FLMD1 作为通用端口引脚时，参见第四章 端口功能和图 17-3 中的注意事项 3。
 3. CSIM10 的第 0 位 (CSOT10) 和串行 I/O 移位寄存器 10 (SIO10) 被复位。
 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 的输出固定为低电平。当数据从 SIO10 被读取时，接收被启动。
 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。

RESET 输入清除该寄存器为 00H。

图 17-3. 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	数据发送/接收时序的指定	类型
0	0		1
0	1		2
1	0		3
1	1		4

CKS102	CKS101	CKS100	CSI10 串行时钟选择 ^注	模式
0	0	0	$f_x/2^2$ (5 MHz)	主模式
0	0	1	$f_x/2^3$ (2.5 MHz)	主模式
0	1	0	$f_x/2^4$ (1.25 MHz)	主模式
0	1	1	$f_x/2^5$ (625 kHz)	主模式
1	0	0	$f_x/2^6$ (312.5 kHz)	主模式
1	0	1	$f_x/2^7$ (156.25 kHz)	主模式
1	1	0	$f_x/2^8$ (78.13 kHz)	主模式
1	1	1	输入到 SCK10 的外部时钟	从模式

注 确保设置串行时钟使其满足以下条件。

- $V_{DD} = 4.0$ 到 5.5 V: 串行时钟 ≤ 5 MHz

- 注意事项**
1. 当内部振荡时钟被选择作为提供给 CPU 的时钟时，内部振荡器的时钟被分频并作为串行时钟被提供。此时，串行接口 CSI10 的操作不被保证。
 2. 当 CSIE10 = 1（操作允许）时，不要写入 CSIC10。
 3. 要使用 P15/SCK10、P16/SI10 和 P17/SO10/FLMD1 作为通用端口引脚时，清除 CKP10 为 0。
 4. 复位后，数据时钟的相位类型是类型 1。

- 备注**
1. 括号中的数字用于 $f_x = 20$ MHz 的操作。
 2. f_x : X1 输入时钟振荡频率

(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/ $\overline{\text{SCK10}}$ 作为串行接口的时钟输出引脚时，清除 PM15 为 0，并设置 P15 的输出锁存为 1。

当 P17/SO10/FLMD1 用作数据输出引脚时，清除 PM17 和 P17 的输出锁存为 0。

当使用 P15/ $\overline{\text{SCK10}}$ 作为串行接口的时钟输入引脚并且使用 P16/SI10 作为数据输入引脚时，设置 PM15 和 PM16 为 1。这时，P15 和 P16 的输出锁存可能是 0 或 1。

PM1 可以由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入设置该寄存器为 FFH。

图 17-4. 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 I/O 模式选择 (n = 0 到 7)
0	输出模式 (输出缓存打开)
1	输入模式 (输出缓存关闭)

17.4 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 I/O 模式

17.4.1 操作停止模式

在这种模式下，串行通信不执行。因此，电源消耗可以减少。此外，在这种模式下，P15/ $\overline{\text{SCK10}}$ 、P16/SI10 和 P17/SO10/FLMD1 引脚可以被用作普通 I/O 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，清除 CSIM10 的第 7 位 (CSIE10) 为 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。

$\overline{\text{RESET}}$ 输入清除 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 I/O 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. 当使用 P15/ $\overline{\text{SCK10}}$ 、P16/SI10 和 P17/SO10/FLMD1 作为通用端口引脚时，参见第四章 端口功能和图 17-3 中的注意事项 3。
 2. CSIM10 的第 0 位 (CSOT10) 和串行 I/O 移位寄存器 10 (SIO10) 被复位。

17.4.2 3 线串行 I/O 模式

3 线串行 I/O 模式用于连接带时钟驱动串行接口的外围 ICs 和显示控制器。

在该模式下，通信通过使用三根线来执行：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 I/O 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 17-3)。
- <2> 设置 CSIM10 寄存器的第 0、4 和 6 位 (CSOT10、DIR10 和 TRMD10) (见 图 17-2)。
- <3> 设置 CSIM10 寄存器的第 7 位 (CSIE10) 为 1。→ 发送/接收被允许。
- <4> 写入数据到发送缓存寄存器 10 (SOTB10)。→ 数据发送/接收开始。
从串行 I/O 移位寄存器 10 (SIO10) 读取数据。→ 数据接收开始。

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 17-2. 寄存器设置和引脚之间的关系

CSIE10	TRMD10	PM16	P16	PM17	P17	PM15	P15	CSI10 操作	引脚功能		
									SI10/P16	SO10/P17 /FLMD1	SCK10 / P15
0	x	x ^{注1}	x ^{注1}	x ^{注1}	x ^{注1}	x ^{注1}	x ^{注1}	停止	P16	P17 /FLMD1	P15 ^{注2}
1	0	1	x	x ^{注1}	x ^{注1}	1	x	从接收 ^{注3}	SI10	P17 /FLMD1	SCK10 (输入) ^{注3}
1	1	x ^{注1}	x ^{注1}	0	0	1	x	从发送 ^{注3}	P16	SO10	SCK10 (输入) ^{注3}
1	1	1	x	0	0	1	x	从发送/ 接收 ^{注3}	SI10	SO10	SCK10 (输入) ^{注3}
1	0	1	x	x ^{注1}	x ^{注1}	0	1	主接收	SI10	P17 /FLMD1	SCK10 (输出)
1	1	x ^{注1}	x ^{注1}	0	0	0	1	主发送	P16	SO10	SCK10 (输出)
1	1	1	x	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 P15/SCK10 作为端口引脚，清除 CKP10 为 0。
 3. 要使用从模式，设置 CKS102、CKS101 和 CKS100 为 1、1、1。

备注

x: 不关注

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 到 0 位

PM1x: 端口模式寄存器

P1x: 端口输出锁存

(2) 通信操作

在 3 线串行 I/O 模式下，数据以 8 位为单元被发送或接收。数据的每个位按照串行时钟同步发送或接收。

如果串行操作模式寄存器 10 (CSIM10) 的第 6 位 (TRMD10) 是 1，数据可以被发送或接收。当数据被写入发送缓存寄存器 10 (SOTB10) 时，发送/接收被启动。此外，当串行操作模式寄存器 10 (CSIM10) 的第 6 位 (TRMD10) 为 0 时，数据可以被接收。

当数据从串行 I/O 移位寄存器 10 (SIO10) 读取时，接收被启动。

通信被启动后，CSIM10 的第 0 位 (CSOT10) 被设置为 1。当 8 位数据的通信完成时，通信完成中断请求标志 (CSIIF10) 被设置，并且 CSOT10 被清除为 0。然后，下一个通信被允许。

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 17-5. 3 线串行 I/O 模式的时序 (1/2)

(1) 发送/接收时序 (类型 1; TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 0)

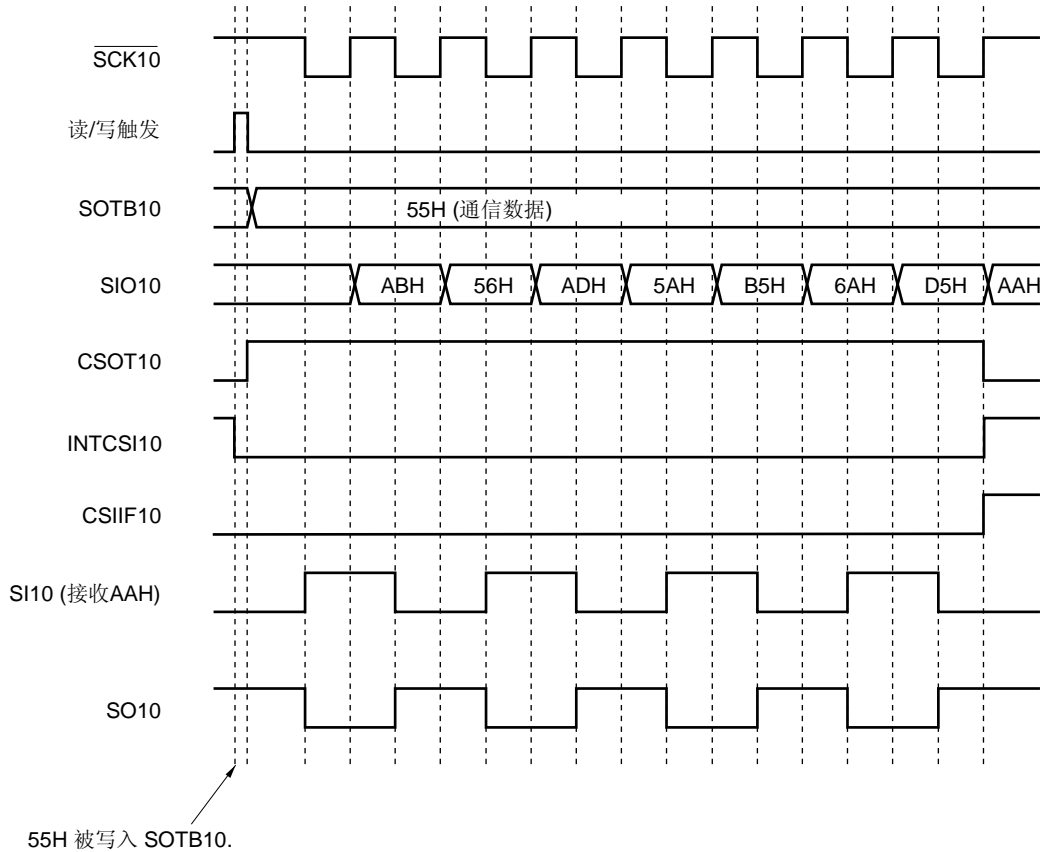


图 17-5. 3 线串行 I/O 模式的时序 (2/2)

(2) 发送/接收时序 (类型 2; TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 1)

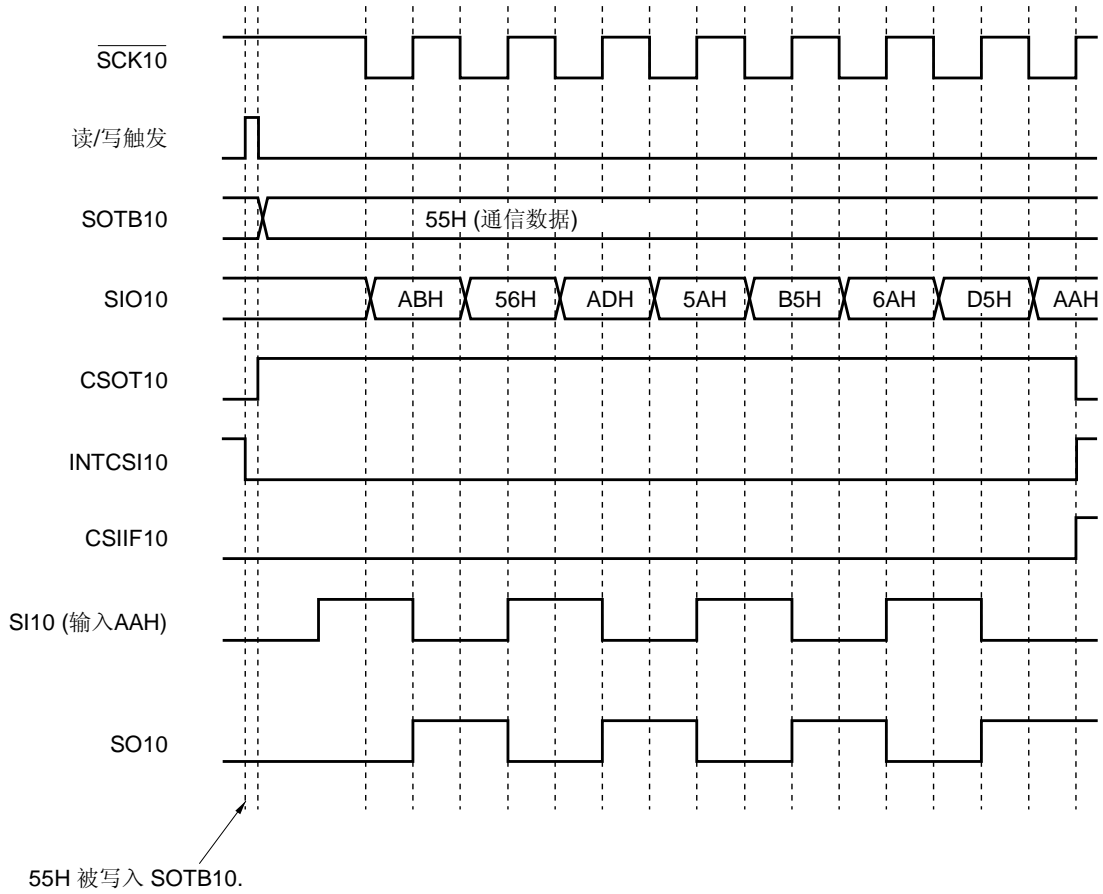
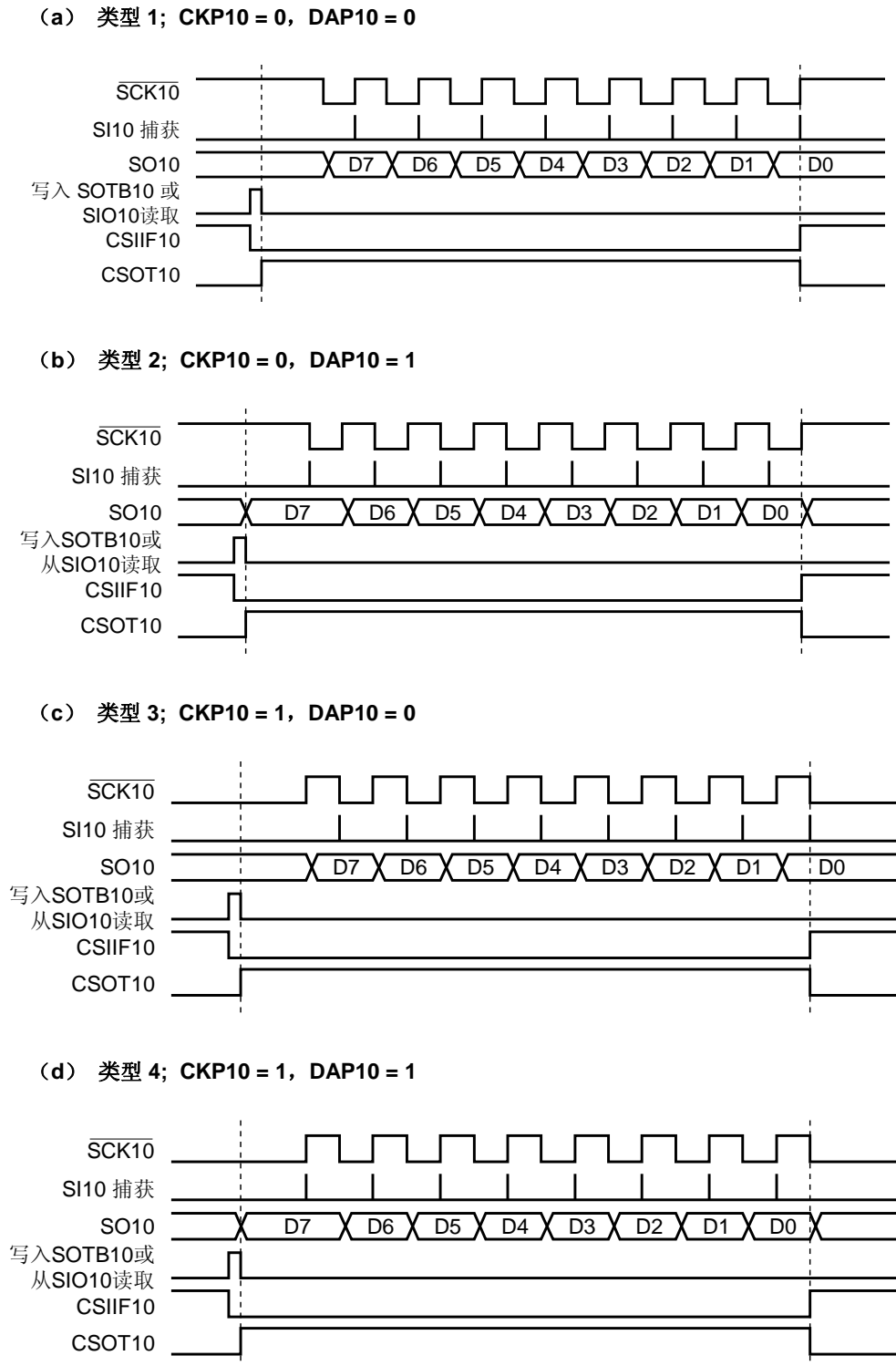


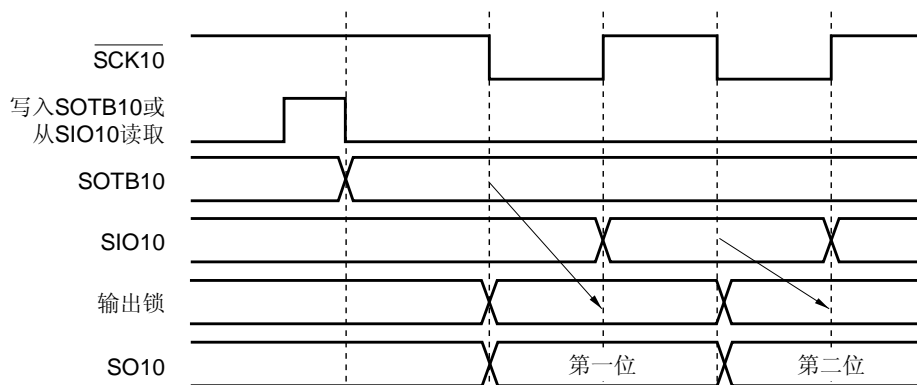
图 17-6. 时钟/数据相位的时序



(3) 输出到 SO10 引脚的时序 (第一位)

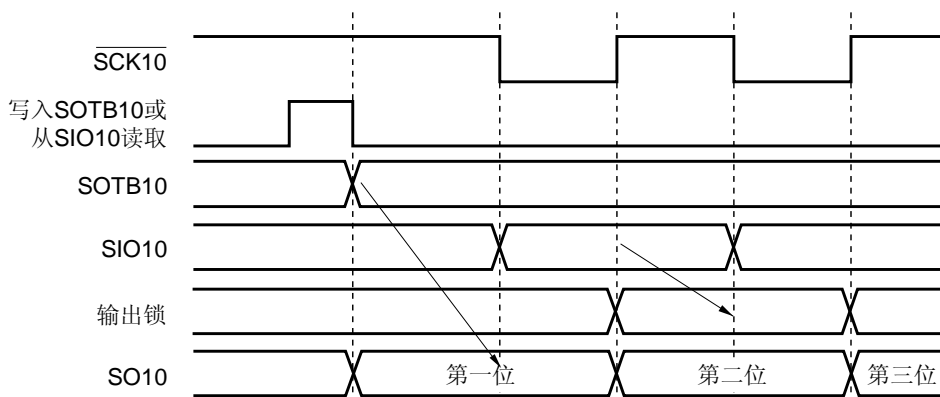
当通信开始时, 发送缓存寄存器 10 (SOTB10) 的值从 SO10 引脚被输出。这时第一位的输出操作如下所示。

图 17-7. 第一位的输出操作

(1) 当 CKP10 = 0, DAP10 = 0 (或 CKP10 = 1, DAP10 = 0) 时

第一位在 $\overline{\text{SCK10}}$ 的下降沿 (或上升沿) 直接由 SOTB10 寄存器锁存到输出锁存, 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 寄存器的值被转移到 SIO10 寄存器, 并且移位第一位。同时, 通过 SI10 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, 并且数据从 SO10 引脚输出。

(2) 当 CKP10 = 0, DAP10 = 1 (或 CKP10 = 1, DAP10 = 1) 时

第一位在 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, 并且通过输出选择器从 SO10 引脚输出。然后, 在 $\overline{\text{SCK10}}$ 的下一个下降沿 (或上升沿), SOTB10 寄存器的值被转移到 SIO10 寄存器, 并且移位第一位。同时, 通过 SI10 引脚接收数据的第一位被保存到 SIO10 寄存器。

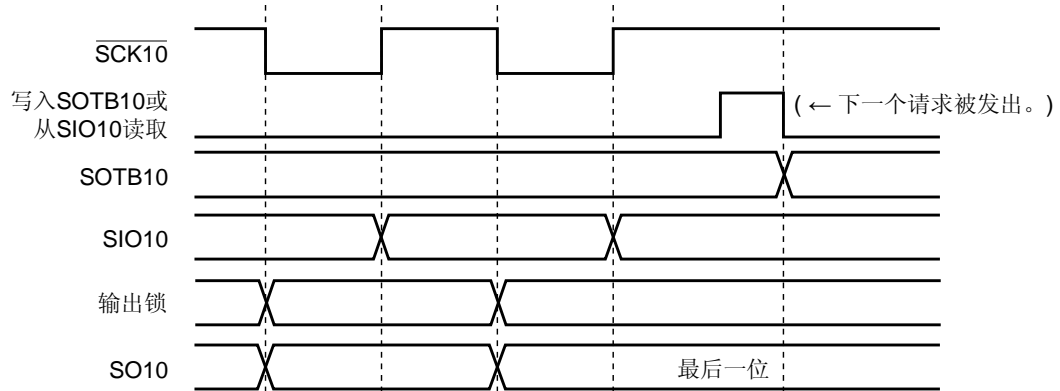
第二位以及后面的位在 $\overline{\text{SCK10}}$ 的后面的上升沿 (或下降沿) 由 SIO10 寄存器锁存到输出锁存, 并且数据从 SO10 引脚输出。

(4) SO10 引脚的输出值 (最后一位)

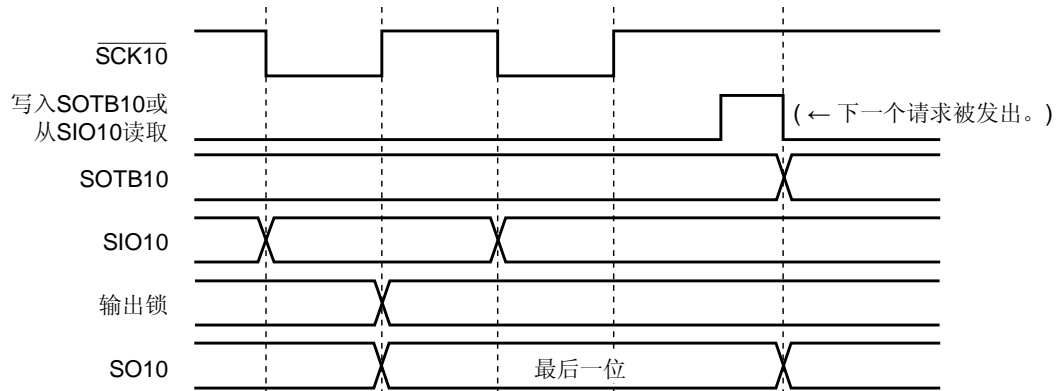
通信完成后, SO10 引脚保持最后一位的输出值。

图 17-8. SO10 引脚的输出值 (最后一位)

(1) 类型 1; 当 CKP10 = 0 并且 DAP10 = 0 (或 CKP10 = 1, DAP10 = 0) 时



(2) 类型 2; 当 CKP10 = 0 并且 DAP10 = 1 (或 CKP10 = 1, DAP10 = 1) 时



(5) SO10 输出

如果串行操作模式寄存器 10 (CSIM10) 的第 7 位 (CSIE10) 被清除为 0, SO10 的输出状态如下所示。

表 17-3. SO10 输出状态

TRMD10	DAP10	DIR10	SO10 输出 ^{注1}
TRMD10 = 0 ^{注2}	-	-	输出低电平 ^{注2}
TRMD10 = 1	DAP10 = 0	-	SO10 锁存的值 (低电平输出)
	DAP10 = 1	DIR10 = 0	SOTB10 的第 7 位的值
		DIR10 = 1	SOTB10 的第 0 位的值

- 注
1. SO10/P17/FLMD1 引脚的实际输出不只由 SO10 输出决定, 还由 PM17 和 P17 决定。
 2. 复位后的状态。

注意事项 如果一个值被写入 TRMD10、DAP10 和 DIR10, SO10 的输出值会改变。

第十八章 乘法器/除法器

18.1 乘法器/除法器的功能

乘法器/除法器具有以下功能。

- 16 位 × 16 位 = 32 位（乘法）
- 32 位 ÷ 16 位 = 32 位，16 位余数（除法）

18.2 乘法器/除法器的配置

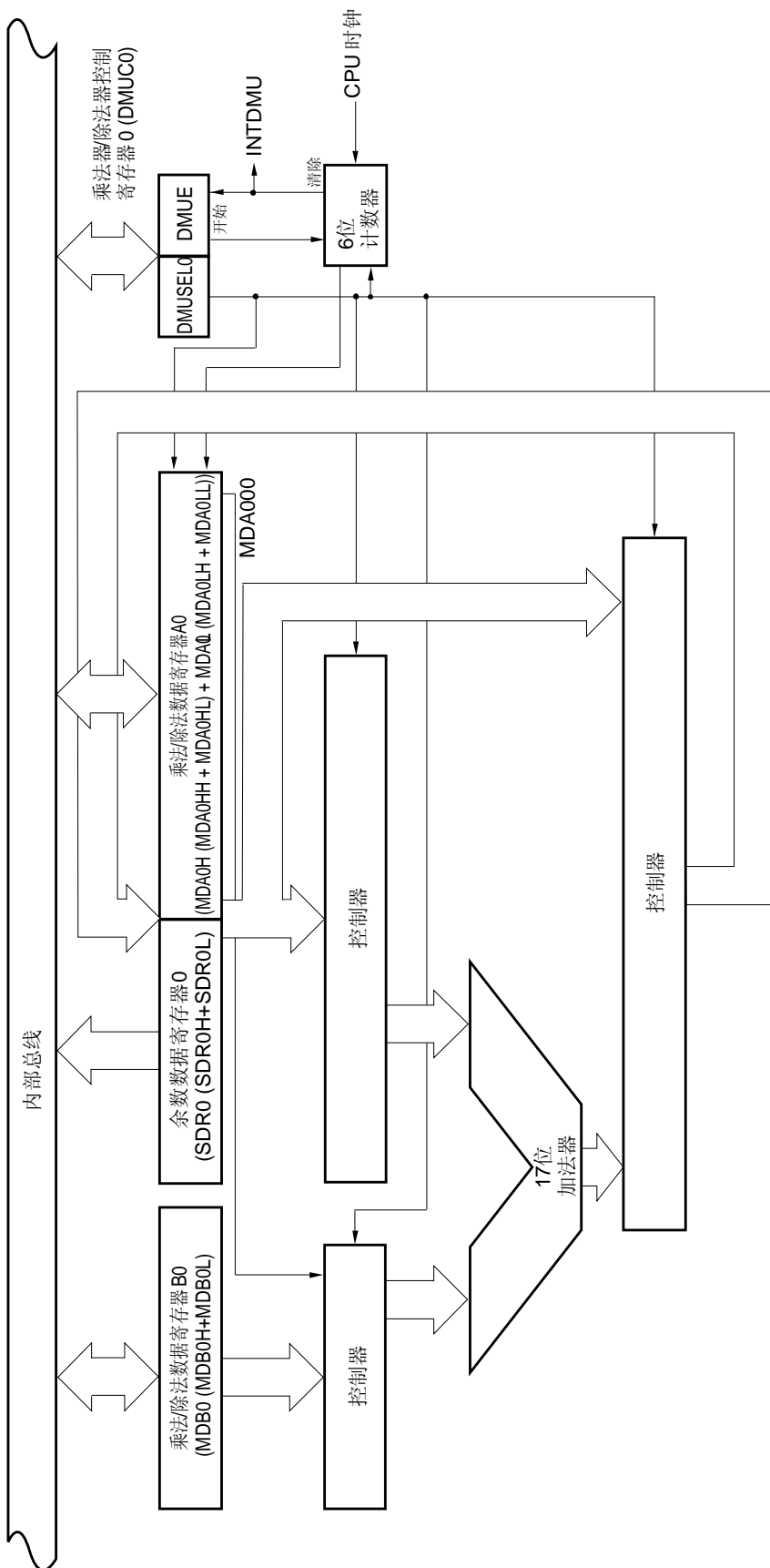
乘法器/除法器由以下硬件组成。

表 18-1. 乘法器/除法器的配置

项目	配置
寄存器	余数数据寄存器 0 (SDR0) 乘法/除法数据寄存器 A0 (MDA0H, MDA0L) 乘法/除法数据寄存器 B0 (MDB0)
控制寄存器	乘法器/除法器控制寄存器 0 (DMUC0)

图 18-1 表示乘法器/除法器的框图。

图 18-1. 乘法器/除法器的框图



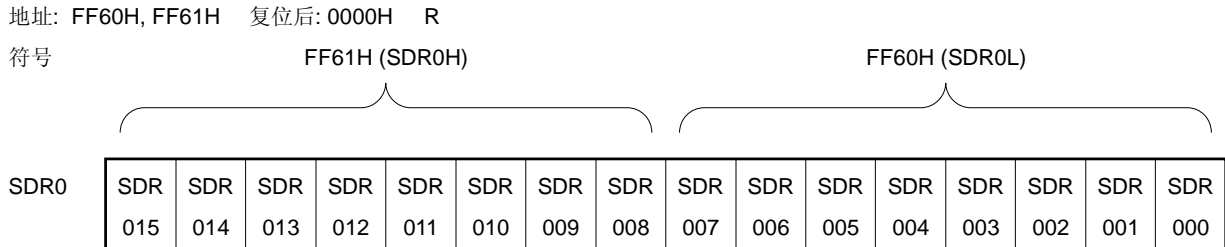
(1) 余数数据寄存器 0 (SDR0)

SDR0 是一个保存余数的 16 位寄存器。在乘法模式下，该寄存器保存 0，在除法模式下，保存运算结果的余数。

该寄存器可以由 8 位或 16 位存储器操作指令读取。

RESET 输入清除该寄存器为 0000H。

图 18-2. 余数数据寄存器 0 (SDR0) 的格式



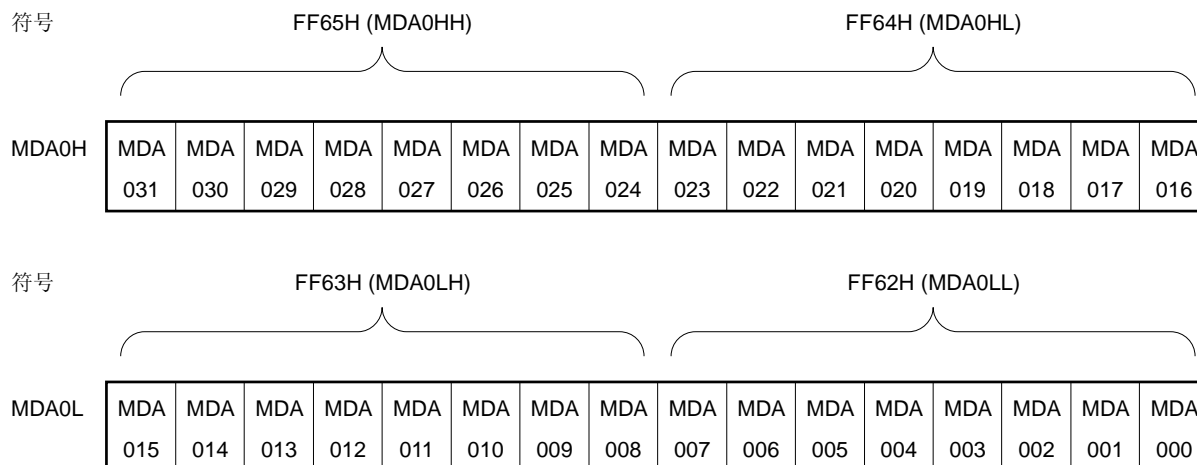
- 注意事项
1. 运算处理期间 (乘法器/除法器控制寄存器 0 (DMUC0) 的第 7 位 (DMUE) 为 1) SDR0 的读取值不被保证。
 2. 当运算开始时 (当 DMUE 被设置为 1 时), SDR0 被复位。

(2) 乘法/除法数据寄存器 A0 (MDA0H, MDA0L)

MDA0 是一个 32 位寄存器，在乘法模式下设置 16 位乘数 A，在除法模式下设置 32 位被除数，并保存运算的 32 位结果（高 16 位：MDA0H，低 16 位：MDA0L）。

图 18-3. 乘法/除法数据寄存器 A0 (MDA0H, MDA0L) 的格式

地址: FF62H, FF63H, FF64H, FF65H 复位后: 0000H, 0000H R/W



- 注意事项
1. 当在乘法模式下运算开始时（乘法器/除法器控制寄存器 0 (DMUC0) 被设置为 81H 时），MDA0H 被清除为 0。
 2. 在运算处理（乘法器/除法器控制寄存器 0 (DMUC0) 的第 7 位 (DMUE) 为 1）期间，不要更改 MDA0 的值。在这种情况下，如果运算被执行，结果将不确定。
 3. 运算处理 (DMUE 为 1 时) 期间，从 MDA0 读取的值不被保证。

18.3 控制乘法器/除法器的寄存器

乘法器/除法器由乘法器/除法器控制寄存器 0 (DMUC0) 来控制。

(1) 乘法器/除法器控制寄存器 0 (DMUC0)

DMUC0 是一个控制乘法器/除法器的 8 位寄存器。

该寄存器可以由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 18-5. 乘法器/除法器控制寄存器 0 (DMUC0) 的格式

地址: FF68H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	0
DMUC0	DMUE	0	0	0	0	0	0	DMUSEL0
DMUE ^注	运算开始/停止							
0	停止运算							
1	开始运算							
DMUSEL0	运算模式(乘法/除法)选择							
0	除法模式							
1	乘法模式							

注 当 DMUE 被设置为 1 时，运算开始。在运算完成时 DMUE 自动清除为 0。

- 注意事项
1. 如果在运算处理（当 DMUE 为 1 时）期间 DMUE 被清除为 0，运算结果不被保证。如果清除指令执行时运算完成，如果中断标志被设置，运算结果被保证。
 2. 在运算处理（当 DMUE 为 1 时）期间，不要更改 DMUSEL0 的值。如果更改，不确定的运算结果将被保存到乘法/除法数据寄存器 A0 (MDA0) 和余数数据寄存器 0 (SDR0) 中。
 3. 如果在运算处理（当 DMUE 为 1 时）期间 DMUE 被清除为 0，运算处理被停止。要再次执行运算，设置乘法/除法数据寄存器 A0 (MDA0)、乘法/除法数据寄存器 B0 (MDB0) 和乘法器/除法器控制寄存器 0 (DMUC0)，并启动运算（通过设置 DMUE 为 1）。

18.4 乘法器/除法器的操作

18.4.1 乘法运算

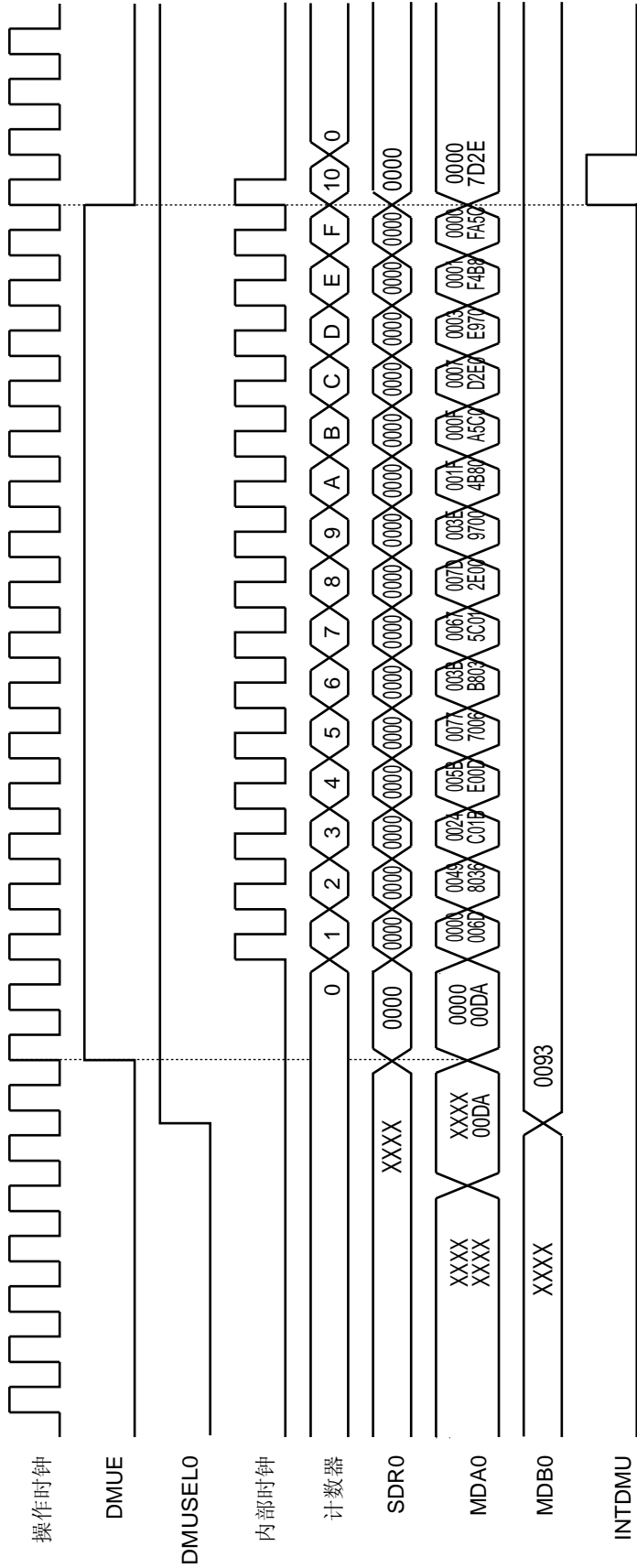
- 初始设置
 1. 设置运算数据到乘法/除法数据寄存器 A0L (MDA0L) 和乘法/除法数据寄存器 B0 (MDB0) 中。
 2. 设置乘法器/除法器控制寄存器 0 (DMUC0) 的第 0 位 (DMUSEL0) 和第 7 位 (DMUE) 为 1。运算将开始。

- 运算期间
 3. 当运算开始后 16 个内部时钟周期发出时, 运算将完成 (在运算期间, 中间数据被保存在 MDA0L 和 MDA0H 寄存器中, 因此这些寄存器的读取值不被保证)。

- 运算结束
 4. 运算结果数据保存在 MDA0L 和 MDA0H 寄存器中。
 5. DMUE 被清除为 0 (运算结束)。
 6. 运算后, 一个中断请求信号 (INTDMU) 被产生。

- 下一个运算
 7. 要执行下一个乘法, 从 **18.4.1 乘法运算** 中的初始设置开始。
 8. 要执行下一个除法, 从 **18.4.2 除法运算** 中的初始设置开始。

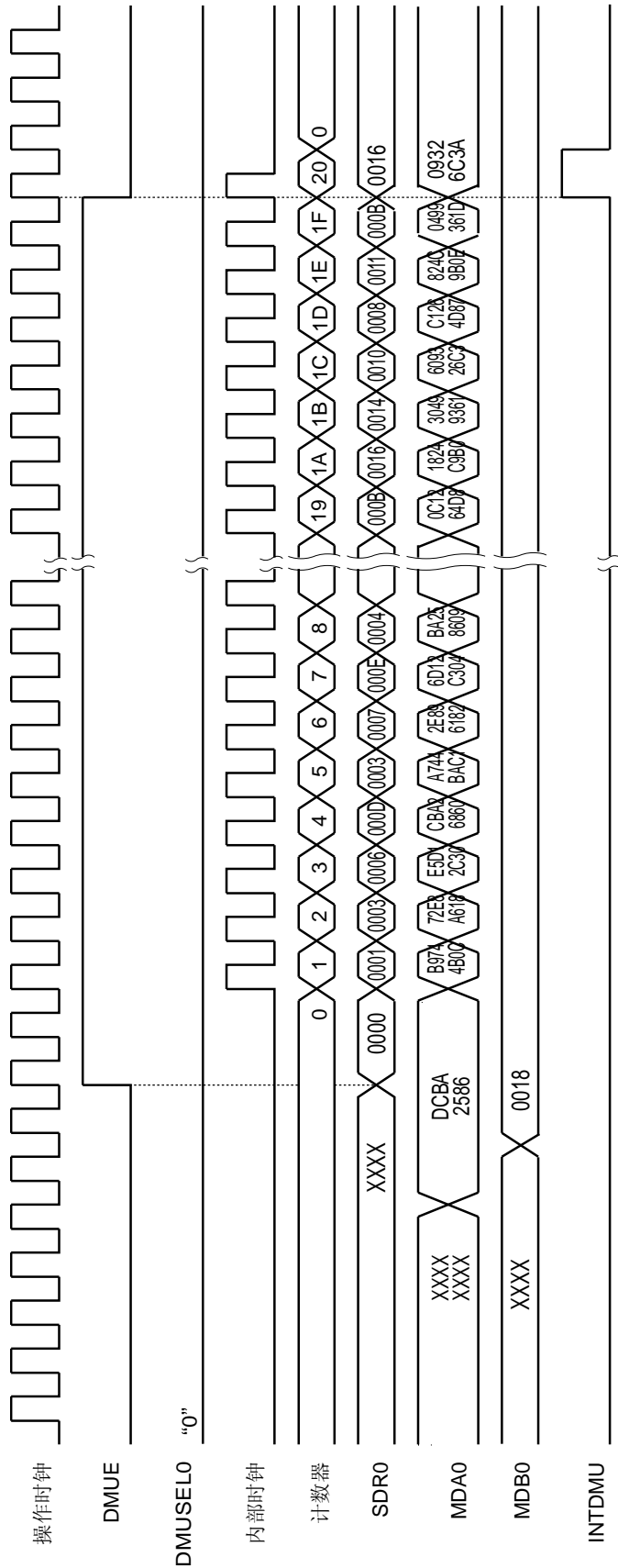
图 18-6. 乘法运算的时序图 (00DAH × 0093H)



18.4.2 除法运算

- 初始设置
 1. 设置运算数据到乘法/除法数据寄存器 A0 (MDA0) 和乘法/除法数据寄存器 B0 (MDB0) 中。
 2. 设置乘法器/除法器控制寄存器 0 (DMUC0) 的第 0 位 (DMUSEL0) 和第 7 位 (DMUE) 为 1。运算将开始。
- 运算期间
 3. 当运算开始后 32 个内部时钟周期发出时，运算将完成（在运算期间，中间数据被保存在 MDA0L 和 MDA0H 寄存器以及余数数据寄存器 0 (SDR0) 中，因此这些寄存器的读取值不被保证）。
- 运算结束
 4. 运算结果数据保存在 MDA0L、MDA0H 和 SDR0 寄存器中。
 5. DMUE 被清除为 0（运算结束）。
 6. 运算后，一个中断请求信号 (INTDMU) 被产生。
- 下一个运算
 7. 要执行下一个乘法，从 **18.4.1 乘法运算** 中的初始设置开始。
 8. 要执行下一个除法，从 **18.4.2 除法运算** 中的初始设置开始。

图 18-7. 除法运算的时序图 (DCBA2586H ÷ 0018H)



第十九章 中断功能

19.1 中断功能类型

以下三种类型的中断功能被使用。

<R>

(1) 非屏蔽中断

即使在中断被禁止时，非屏蔽中断也被响应。它不受优先级控制并且具有比所有其它中断请求更高的优先级。然而，在非屏蔽中断服务期间，中断请求保持未决。非屏蔽中断产生待机释放信号并释放 STOP 模式和 HALT 模式。
 μ PD78F0714 中的唯一一个非屏蔽中断是来自低压检测电路的中断。

(2) 可屏蔽中断

这些中断受屏蔽控制。通过设置优先级指定标志寄存器 (PR0L, PR0H, PR1L, PR1H)，可屏蔽中断可分为高中断优先级组和低中断优先级组。
当生成高优先级中断时，多重中断服务可以被应用于低优先级中断。如果具有相同优先级的两个或更多中断被同时产生，每个中断根据预先确定的优先级来服务（参见表 19-1）。
等待释放信号生成，并且 STOP 以及 HALT 模式也被释放。
8 个外部中断请求和 19 个内部中断请求被规定为可屏蔽中断。

(3) 软件中断

这是通过执行 BRK 指令所生成的向量中断。即使当中断被禁止时，它仍可以响应。软件中断不受制于中断优先级控制。

19.2 中断源和配置

总共 29 个中断源存在于非屏蔽、可屏蔽和软件中断中（参见表 19-1）。

表 19-1. 中断源列表 (1/2)

中断类型	默认优先级 注 1	中断源		内部/外部	向量表地址	基础结构类型注 2
		名称	触发			
<R> 非屏蔽	-	INTLVI	低电压检测注 3	内部	0004H	(A)
可屏蔽	0	INTP0	引脚输入边沿检测	外部	0006H	(B)
	1	INTP1			0008H	
	2	INTP2			000AH	
	3	INTP3			000CH	
	4	INTP4			000EH	
	5	INTP5			0010H	
	6	INTP6			0012H	
	7	INTP7			0014H	
	8	INTTW0UD	TW0UDC 下溢	内部	0016H	(A)
	9	INTTW0CM3	TW0UDC 和 TW0CM3 匹配		0018H	
	10	INTTW0CM4	TW0UDC 和 TW0CM4 匹配		001AH	
	11	INTTW0CM5	TW0UDC 和 TW0CM5 匹配		001CH	
	12	INTCM10	IT20UDC 和 IT20CM0 匹配		001EH	
	13	INTCM11	IT20UDC 和 IT20CM1 匹配		0020H	
	14	INTCC10	IT20UDC 和 IT20CC0 匹配 (指定比较寄存器时) TIT20CC0 引脚有效沿检测 (指定捕获寄存器时)		0022H	
	15	INTCC11	IT20UDC 和 IT20CC1 匹配 (指定比较寄存器时) TIT20CC1 引脚有效沿检测 (指定捕获寄存器时)		0024H	
	-	-	-		0026H注 4	
	16	INTTM00	TM00 和 CR00 匹配 (指定比较寄存器时) TI001 引脚有效沿检测 (指定捕获寄存器时)		0028H	
	17	INTTM01	TM00 和 CR01 匹配 (指定比较寄存器时) TI000 引脚有效沿检测 (指定捕获寄存器时)		002AH	
	18	INTSRE00	UART00 接收错误发生		002CH	
19	INTSR00	UART00 接收结束	002EH			
20	INTST00	UART00 发送结束	0030H			

- 注
1. 默认优先级是两个或更多可屏蔽中断同时产生时应用的优先级。0 是最高优先级，26 是最低优先级。
 2. 基础结构类型 (A) 到 (C) 相应于图 19-1 中的 (A) 到 (C)。
 3. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 被设为 1 时。
 4. 不存在对应于向量表地址 0026H 的中断请求。

表 19-1. 中断源列表 (2/2)

中断类型	默认优先级 ^{注1}	中断源		内部/外部	向量表地址	基础结构类型 ^{注2}
		名称	触发			
可屏蔽	21	INTTM50	TM50 和 CR50 匹配 (指定比较寄存器时)	内部	0032H	(A)
	22	INTTM51	TM51 和 CR51 匹配 (指定比较寄存器时)		0034H	
	23	INTTMH0	TMH0 和 CMP00 匹配 (指定比较寄存器时)		0036H	
	24	INTCSI10	CSI10 通信结束		0038H	
	25	INTDMU	乘/除运算结束		003AH	
	26	INTAD	A/D 转换结束		003CH	
软件	-	BRK	BRK 指令执行	-	003EH	(C)
复位	-	RESET	复位输入	-	0000H	-
		POC	上电清零			
		LVI	低电压检测 ^{注3}			
		WDT	WDT 溢出			

- 注
1. 默认优先级是两个或更多可屏蔽中断同时产生时应用的优先级。0 是最高优先级，26 是最低优先级。
 2. 基础结构类型 (A) 到 (C) 相应于图 19-1 中的 (A) 到 (C)。
 3. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 被设为 1 时。

图 19-1. 中断功能的基础配置 (1/2)

<R>

(A) 内部非屏蔽中断

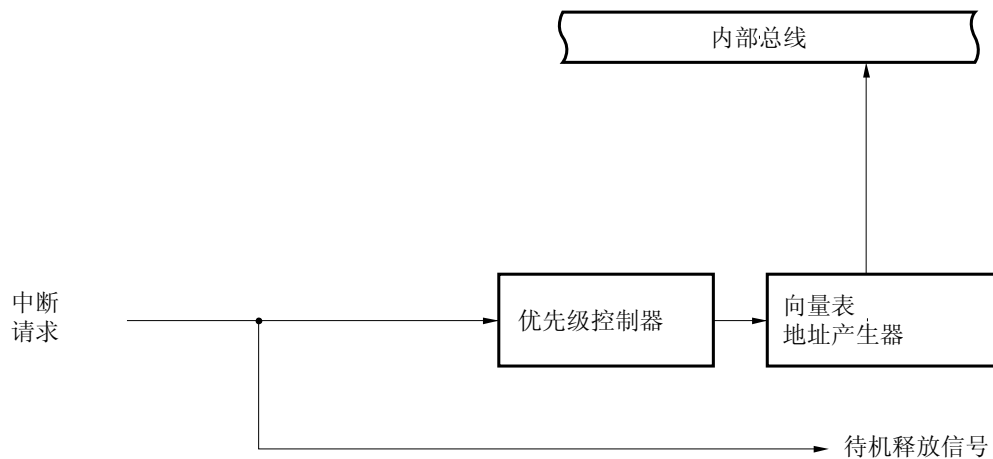
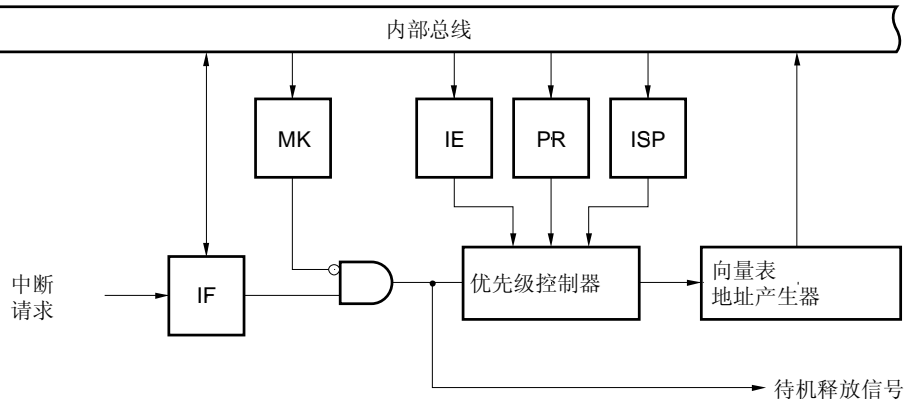
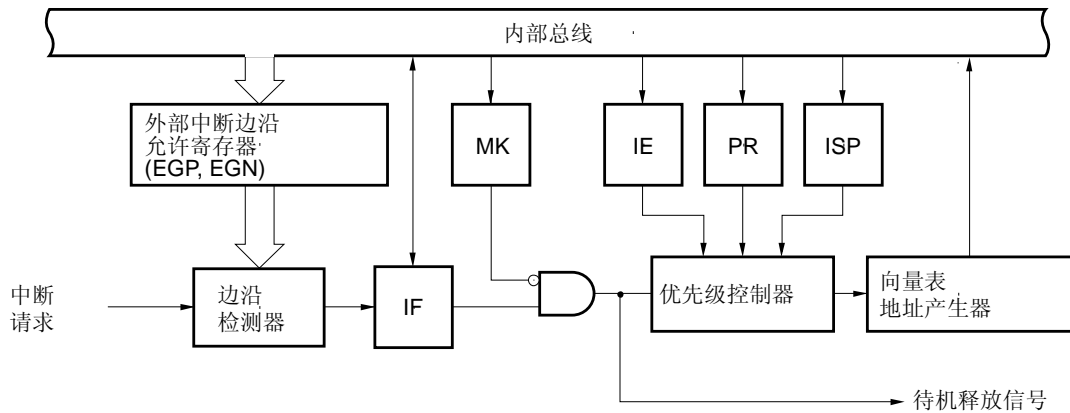


图 19-1. 中断功能的基础配置 (2/2)

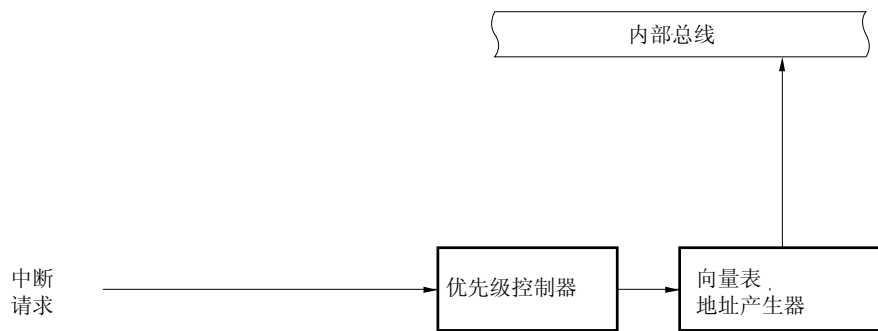
(B) 内部可屏蔽中断



(C) 外部可屏蔽中断 (INTP0 到 INTP7)



(D) 软件中断



- 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

19.3 控制中断功能的寄存器

以下六种类型的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 19-2 显示了相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 19-2. 相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
INTP0	PIF0	IF0L	PMK0	MK0L	PPR0	PR0L
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTP5	PIF5		PMK5		PPR5	
INTP6	PIF6		PMK6		PPR6	
INTP7	PIF7	IF0H	PMK7	MK0H	PPR7	PR0H
INTTW0UD	UDIFW0		UDMKW0		UDPRW0	
INTTW0CM3	CM3IFW0		CM3MKW0		CM3PRW0	
INTTW0CM4	CM4IFW0		CM4MKW0		CM4PRW0	
INTTW0CM5	CM5IFW0		CM5MKW0		CM5PRW0	
INTCM10	CMIF10		CMMK10		CMPR10	
INTCM11	CMIF11		CMMK11		CMPR11	
INTCC10	CCIF10		CCMK10		CCPR10	
INTCC11	CCIF11	IF1L	CCMK11	MK1L	CCPR11	PR1L
INTTM00	TMIF00		TMMK00		TMPR00	
INTTM01	TMIF01		TMMK01		TMPR01	
INTSRE00	SREIF00		SREMK00		SREPR00	
INTSR00	SRIF00		SRMK00		SRPR00	
INTST00	STIF00		STMK00		STPR00	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM51	TMIF51		TMMK51		TMPR51	
INTTMH0	TMIFH0	TMMKH0	TMPRH0			
INTCSI10	CSIF10	CSIMK10	CSIPR10			
INTDMU	DMUIF	DMUMK	DMUPR			
INTAD	ADIF	ADMK	ADPR			

(1) 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H)

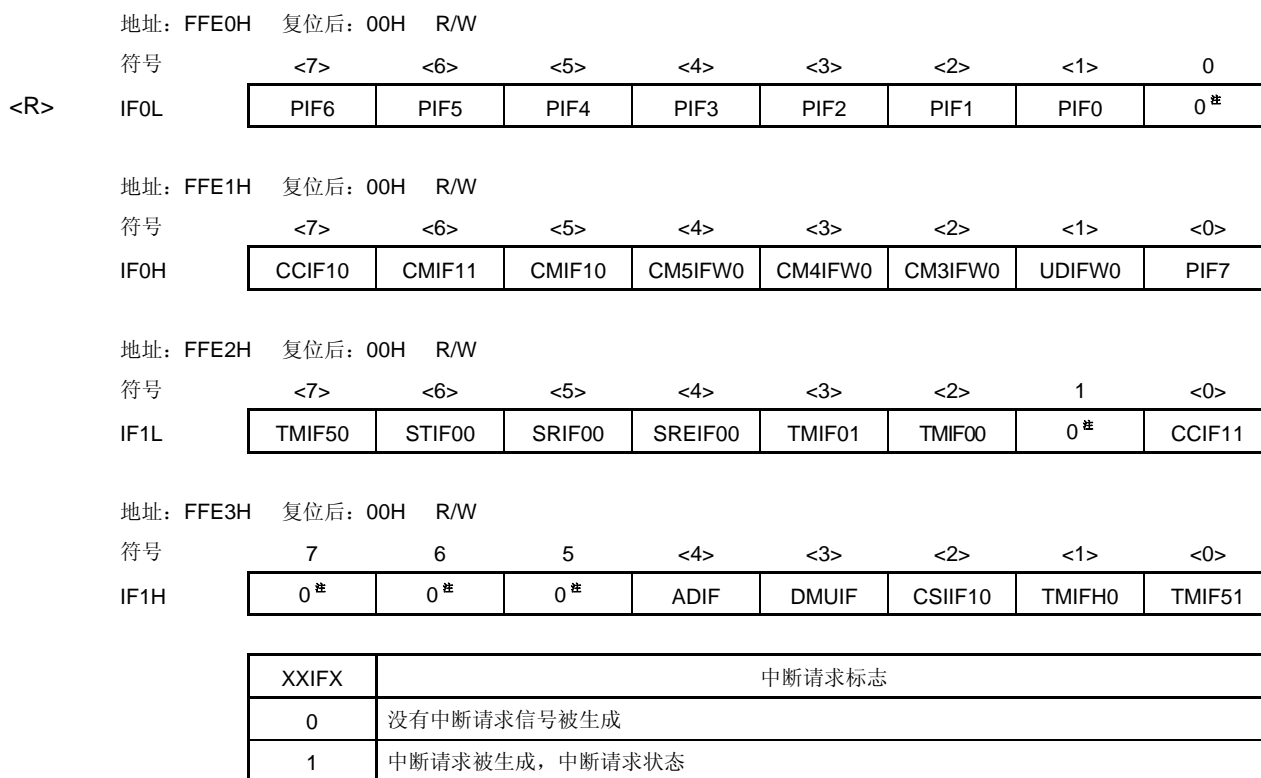
当相应的中断请求被生成或者当执行指令时，中断请求标志将会被设为 1。在中断请求响应或 $\overline{\text{RESET}}$ 输入的基础上，当执行指令时，它们将会被清 0。

当中断被响应时，中断请求标志将会自动清除，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 被组成 16 位寄存器 IF0、IF1 时，它们可以通过 16 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除这些寄存器为 00H。

图 19-2. 中断请求标志寄存器的格式 (IF0L, IF0H, IF1L, IF1H)



注 确保清除 IF0L 的第 0 位、IF1L 的第 1 位和 IF1H 的第 5 到 7 位为 0。

- 注意事项 1. 当在等待释放后操作计时器、串行接口或 A/D 转换器时，应在清除中断请求标志后立即对其进行操作。中断请求标志可以通过噪声来设置。
2. 当操作中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“IF0L.0 = 0;”或“asm(“clr1 IF0L, 0”);”的位操作指令，因为符合的汇编语言必须是 1 位存储器操作指令 (CLR1)。如果使用例如“F0L&=0xfe;”的 8 位存储器操作指令以 C 语言来描述程序，并且程序符合时，它将会变为三种指令的汇编语言。

```

mov a, IF0L
and a, #0FEH
mov IF0L, a
    
```

在这种情况下，即使同一中断请求标志寄存器 (IF0L) 的其它位中的请求标志在“mov a, IF0L”和“mov IF0L, a”之间的时间内被设为 1，标志也将会在“mov IF0L, a”处被清除。因此，在 C 语言中使用 8 位存储器操作指令时必须要注意。

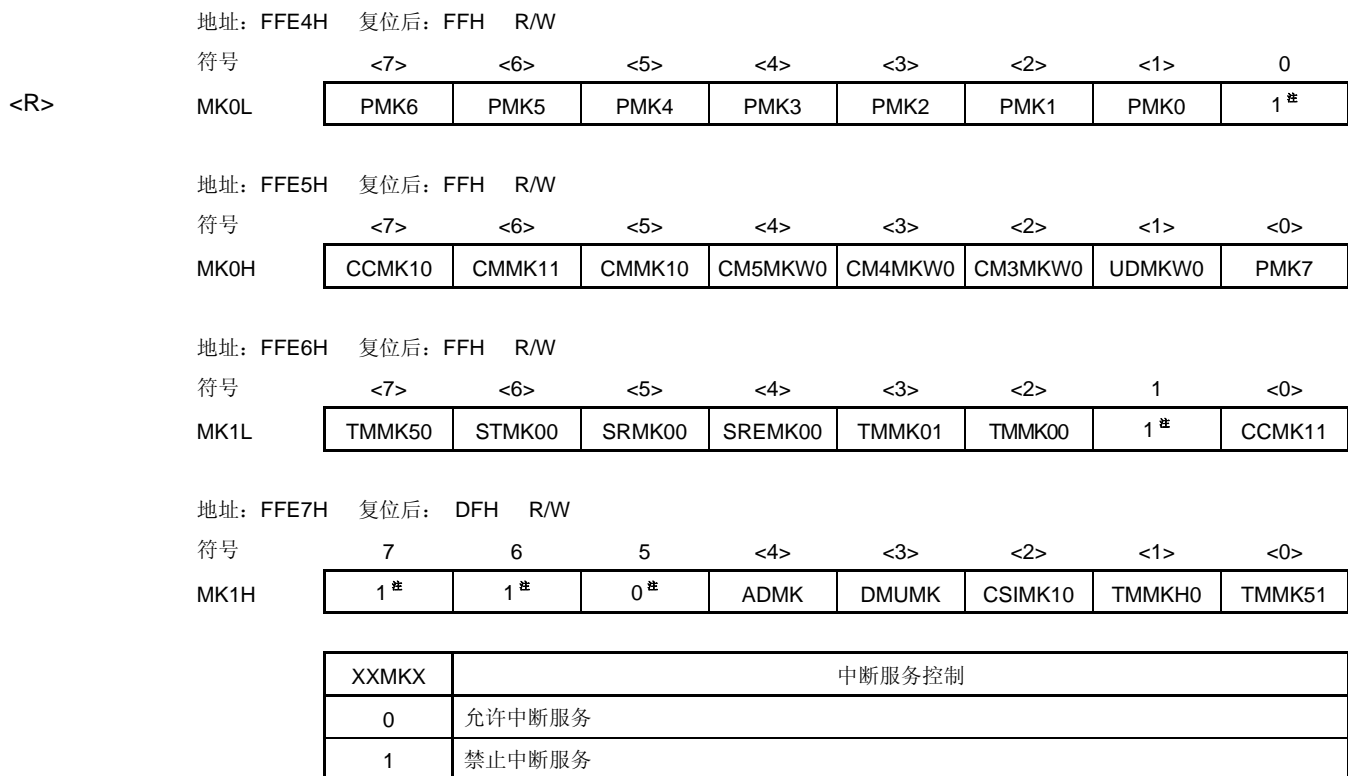
(2) 中断请求标志寄存器 (MK0L, MK0H, MK1L, MK1H)

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 被组成 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。

$\overline{\text{RESET}}$ 输入设置 MK0L、MK0H 和 MK1L 为 FFH 并设置 MK1H 为 DFH。

图 19-3. 中断屏蔽标志寄存器的格式 (MK0L, MK0H, MK1L, MK1H)



注 确保设置 MK0L 的第 0 位、MK1L 的第 1 位和 MK1H 的第 6 到 7 位为 1。
 确保清除 MK1H 的第 5 位为 0。

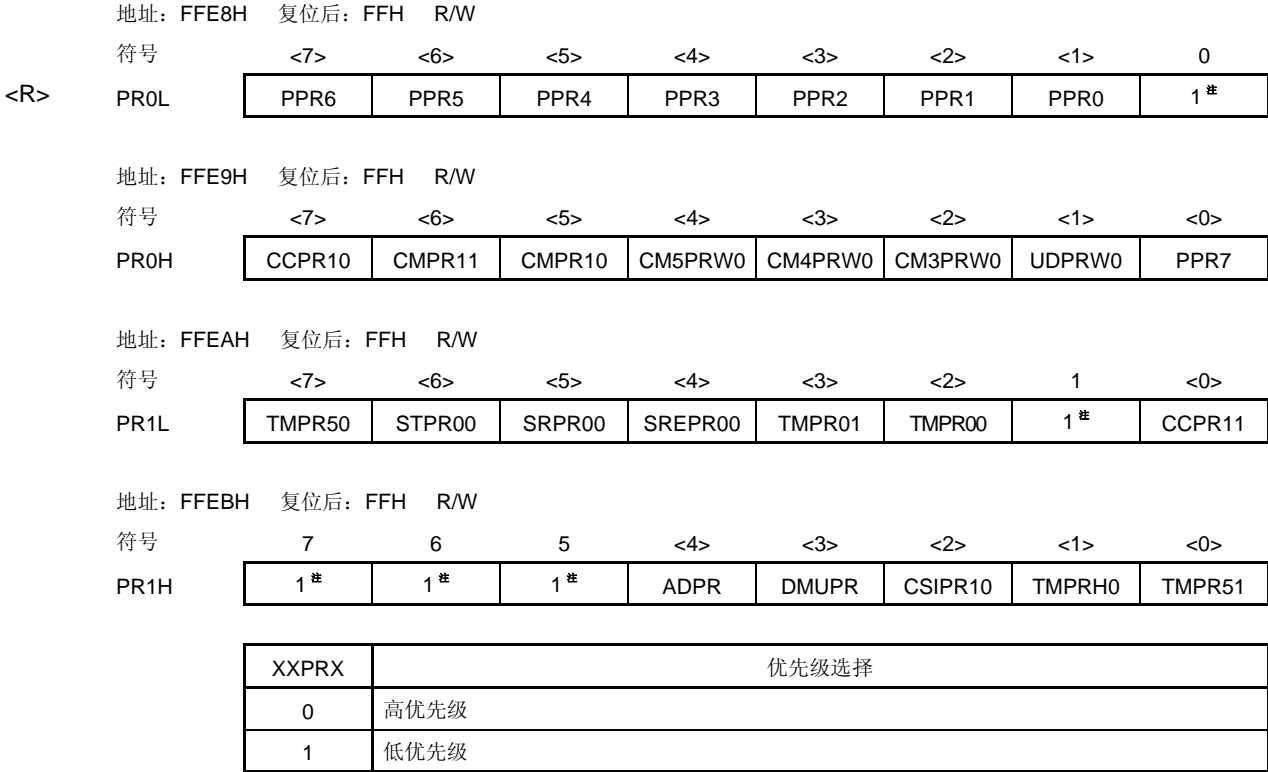
(3) 优先级指定标志寄存器 (PR0L, PR0H, PR1L, PR1H)

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 被组成 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。

RESET 输入设置这些寄存器为 FFH。

图 19-4. 优先指定标志寄存器格式 (PR0L, PR0H, PR1L, PR1H)



注 确保设置 PR0L 的第 0 位、PR1L 的第 1 位和 PR1H 的第 5 到 7 位为 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 INTP7 指定了有效沿。

EGP 和 EGN 可以通过 1 位或 8 位操作指令来设置。

$\overline{\text{RESET}}$ 输入清除这些寄存器为 00H。

图 19-5. 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	EGP7	EGP6	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	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn 引脚有效沿选择 (n=0 到 7)
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 19-3 显示了相应于 EGPn 和 EGNn 的端口。

表 19-3. 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	P00		INTP0	
EGP1	EGN1	P01		INTP1	
EGP2	EGN2	P02		INTP2	
EGP3	EGN3	P03		INTP3	
EGP4	EGN4	P52		INTP4	
EGP5	EGN5	P53		INTP5	
EGP6	EGN6	P55		INTP6	
EGP7	EGN7	P56		INTP7	

注意事项 通过将 EGPn 以及 EGNn 清 0 来选择端口模式，因为当外部中断功能被转换到端口功能时，边沿可以被检测到。

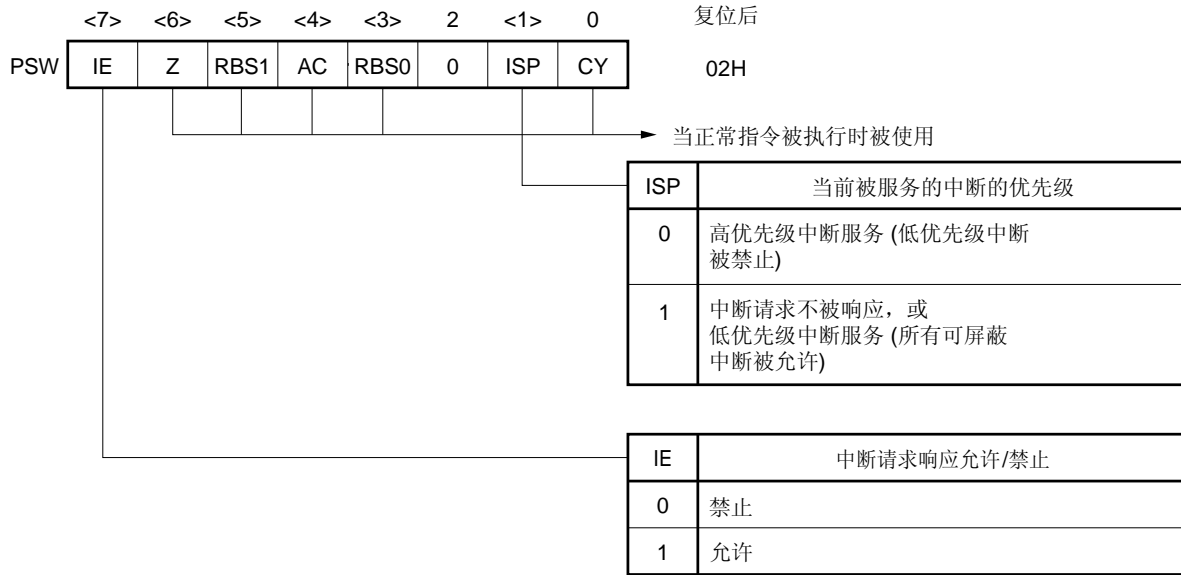
备注 n = 0 到 7

(5) 程序状态字 (PSW)

程序状态字是用于为中断请求保持指令执行结果以及当前状态的寄存器。设置允许/禁止可屏蔽中断的 IE 标志以及控制多重中断服务的 ISP 标志被映射到 PSW。

除了 8 位读/写，该寄存器还可以通过使用位操作指令和专用指令 (EI 和 DI) 来执行操作。当向量中断请求被响应时，如果执行了 BRK 指令，那么 PSW 的内容将会被自动保存到堆栈中，且 IE 标志将会复位为 0。如果可屏蔽中断请求被响应，那么所响应的中断的优先级指定标志中的内容将会被传输到 ISP 标志中。通过 PUSH PSW 指令，PSW 的内容也可以被保存到堆栈中。它们可以通过 RETI, RETB 以及 POP PSW 指令从堆栈中恢复。 $\overline{\text{RESET}}$ 输入设置 PSW 为 02H。

图 19-6. 程序状态字的格式



19.4 中断服务操作

<R>

19.4.1 非屏蔽中断请求响应操作

即使在中断响应被禁止的状态下，非屏蔽中断也会无条件响应。它不受中断优先级控制并且在所有中断中具有最高优先级。

如果非屏蔽中断请求被响应，那么内容将会按照 PSW 的顺序先被保存到堆栈中，PC、IE 标志和 ISP 标志被复位（0），并且向量表的内容被加载到 PC 并跳转。。

这禁止多重中断的响应。

如果在一个非屏蔽中断服务程序期间一个新的非屏蔽中断请求被产生，它将在当前非屏蔽中断服务程序完成（在 RETI 指令执行后）并且一条主程序指令执行后被响应。然而，如果新的非屏蔽中断请求在非屏蔽中断服务程序执行期间产生两次或更多，在非屏蔽中断服务程序完成后只有一个非屏蔽中断请求被响应。

图 19-7 和 19-8 分别表示非屏蔽中断请求的响应时序以及多重非屏蔽中断请求产生时的响应操作。

注意事项 确保使用 RETI 指令来从非屏蔽中断中恢复处理。

图 19-7. 非屏蔽中断请求响应时序

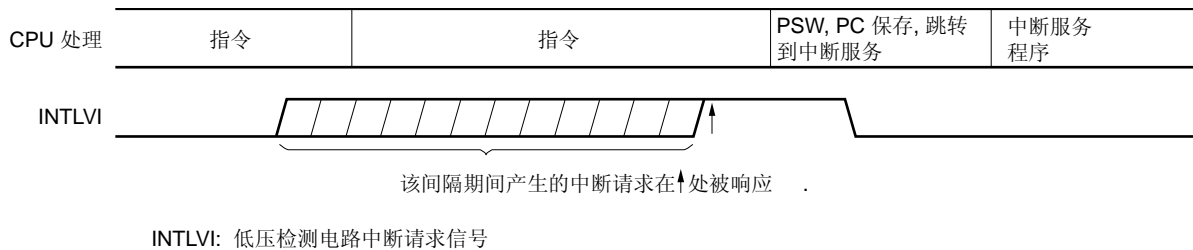
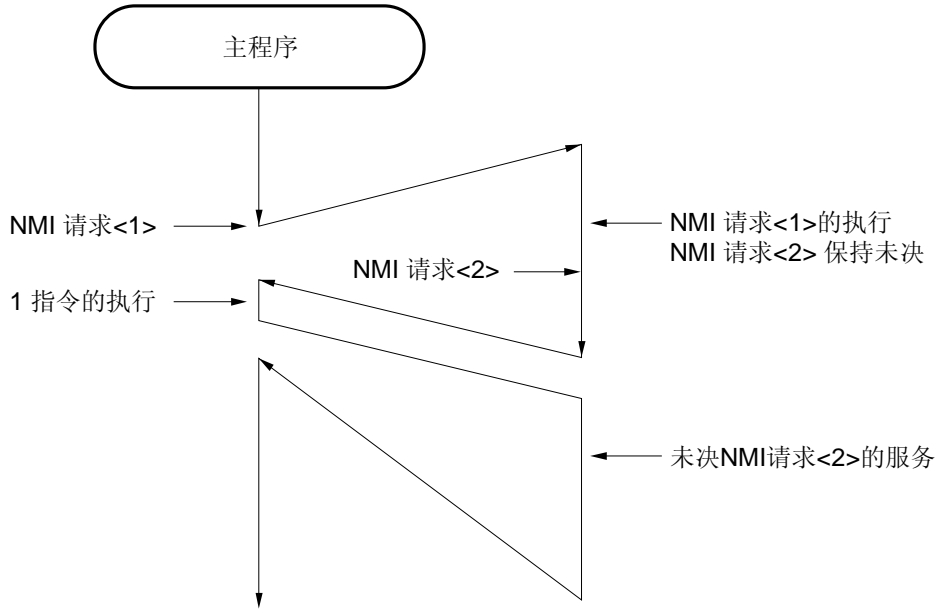
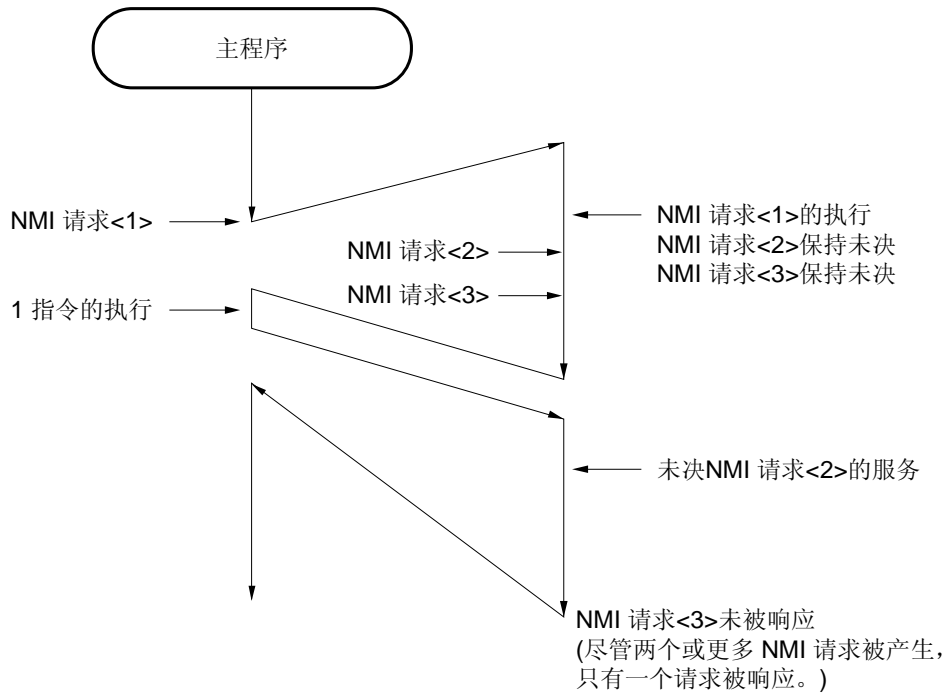


图 19-8. 非屏蔽中断请求响应操作

(a) 如果在非屏蔽中断服务程序执行期间一个非屏蔽中断请求被产生



(b) 如果在非屏蔽中断服务程序执行期间两个非屏蔽中断请求被产生



19.4.2 可屏蔽中断请求响应

当中断请求标志被设为 1 且相应于该中断请求的屏蔽 (MK) 标志被清 0 时, 可屏蔽中断请求将会被响应。如果中断处于中断允许状态 (当 IE 标志被设为 1), 那么向量中断请求将会被响应。然而, 在高优先级中断请求进行期间低优先级中断请求将不会被响应 (当 ISP 标志复位为 0 时)。

此外, 如果 EI 指令在非屏蔽中断服务程序执行期间被执行, 非屏蔽中断请求和可屏蔽中断请求都不被响应。

从可屏蔽中断请求生成到执行中断服务的时间被列于下表 19-4 中。

关于中断请求的响应时序, 请参见图 19-10 以及 19-11。

表 19-4. 从可屏蔽中断请求生成到服务之间的时间

	最短时间	最长时间 [#]
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 指令前生成中断请求, 那么等待时间将会变长。

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

如果同时生成两个或更多的可屏蔽中断, 那么在优先级指定标志中所指定的具有高优先级的请求将会先被响应。

如果两个或更多的中断请求具有同一优先级, 则将会先响应具有最高默认优先级的请求。

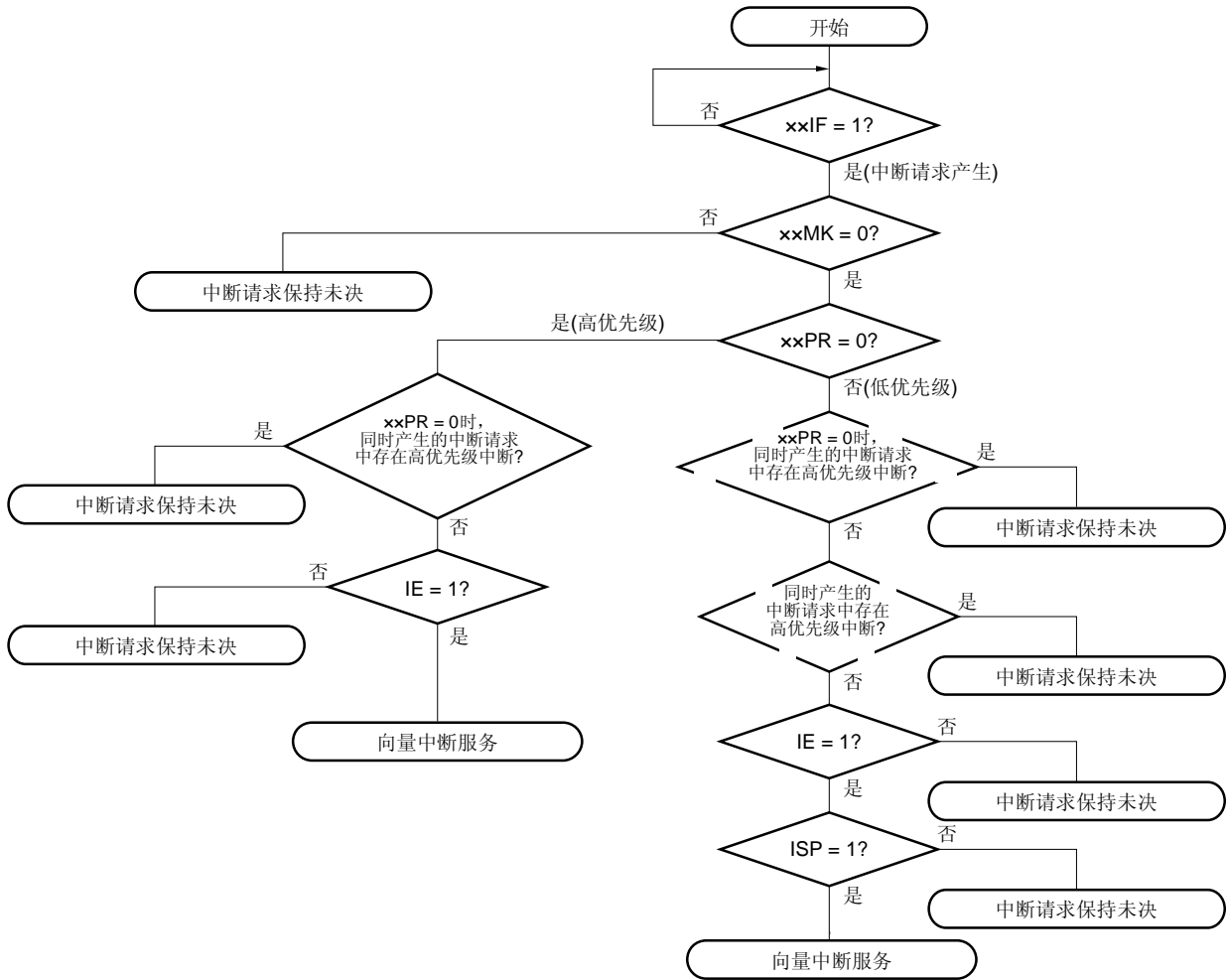
被保持为未决状态的中断请求会在变为可以响应时被响应。

图 19-9 显示了中断请求响应的算法。

如果可屏蔽中断请求被响应, 那么内容将会按照 PSW 的顺序先被保存到堆栈中, 然后被保存到 PC 中, IE 标志也将会复位 (0), 而相应于响应的中断的优先级指定标志中的内容则将会被传输到 ISP 标志中。为每个中断请求所决定的向量表数据被装载到 PC 中并分支。

通过使用 RETI 指令可以实现从中断中恢复。

图 19-9. 中断请求响应处理算法



xxIF: 中断请求标志

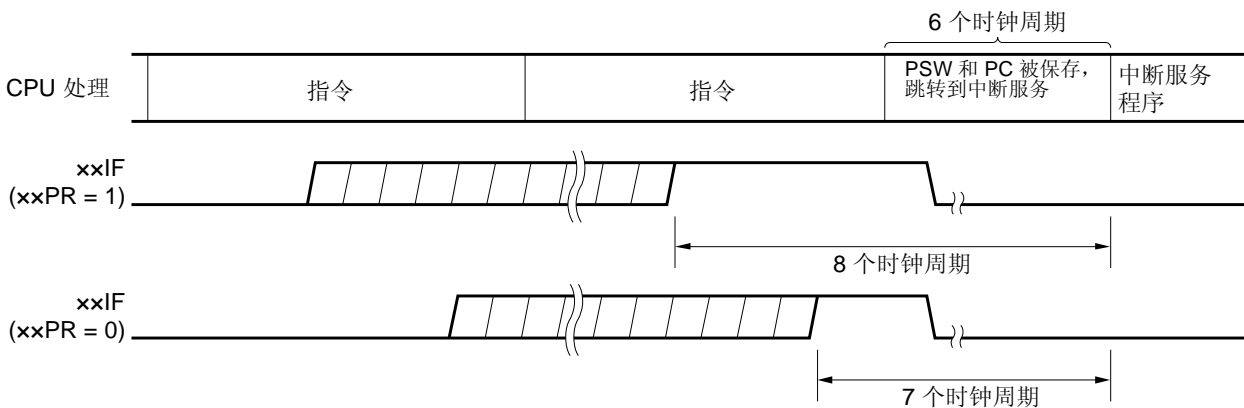
xxMK: 中断屏蔽标志

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)

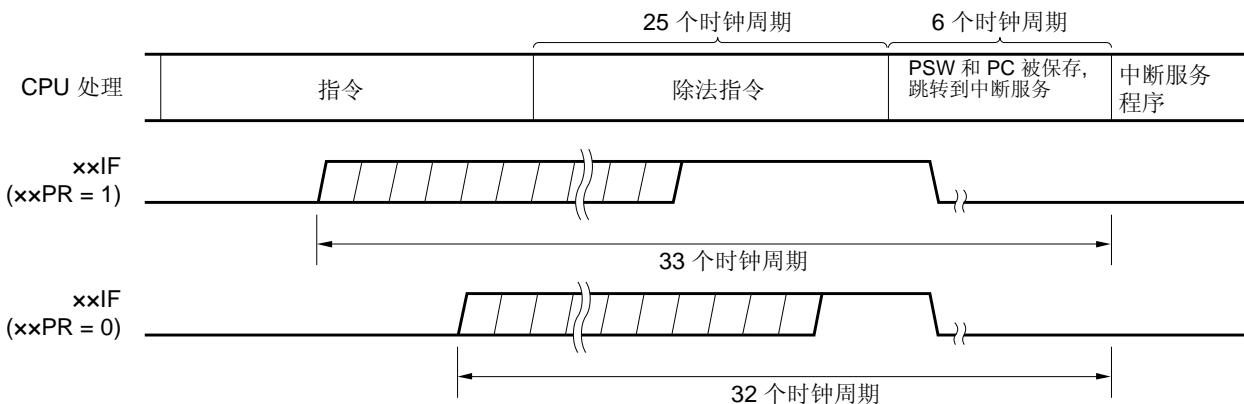
ISP: 表示当前服务中断优先级的标志 (0=高优先级中断服务, 1=无中断请求响应, 或低优先级中断服务)

图 19-10. 中断请求响应时序 (最短时间)



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

图 19-11. 中断请求响应时序 (最长时间)



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

19.4.3 软件中断请求响应

软件中断请求是通过 BRK 指令执行来响应的。软件中断不能被禁止。

如果软件中断被响应，那么内容将会按照程序状态字 (PSW) 的顺序先被保存到堆栈中，然后被保存到程序计数器 (PC) 中，IE 标志也将会复位 (0)，而向量表 (003EH, 003FH) 的内容则被装载到 PC 中并分支。

通过使用 RETB 指令可以实现从软件中断中恢复。

注意事项 不使用 RETI 指令来从软件中断中恢复。

19.4.4 多重中断服务

当另一个中断请求被响应时，多重中断服务将会在执行中断期间发生。

除非中断请求响应允许状态被选择（IE=1），否则多重中断服务将不会发生（非屏蔽中断除外）。同时，当中断请求被响应时，中断请求响应将会被禁止（IE=0）。因此，为了允许多重中断服务，在允许中断响应的中断服务中有必要通过 EI 指令来设置（1）IE 标志。

此外，即使允许中断，多重中断服务也可能不会被允许，这受到中断优先级控制的限制。两种优先级控制的类型是可用的：默认优先级控制以及可编程优先级控制。可编程优先级控制用于多重中断服务。

在中断允许状态中，如果生成了一个与当前进行的中断具有相同优先级或更高优先级的中断请求，那么它将会为多重中断服务来响应。如果在中断服务期间生成了一个中断请求，其优先级低于当前所进行的中断的优先级，那么它不会为多重中断服务来响应。因为中断处于中断禁止状态或具有低优先级，被禁止的中断请求被保持为未决状态。当前中断服务结束时，在执行一个主程序指令执行后未决中断请求将会被响应。

<R> 非屏蔽中断服务期间，不能服务多重中断。

表 19-5 显示了为多重中断服务所允许的中断请求，而图 19-12 则显示了多重中断服务的示例。

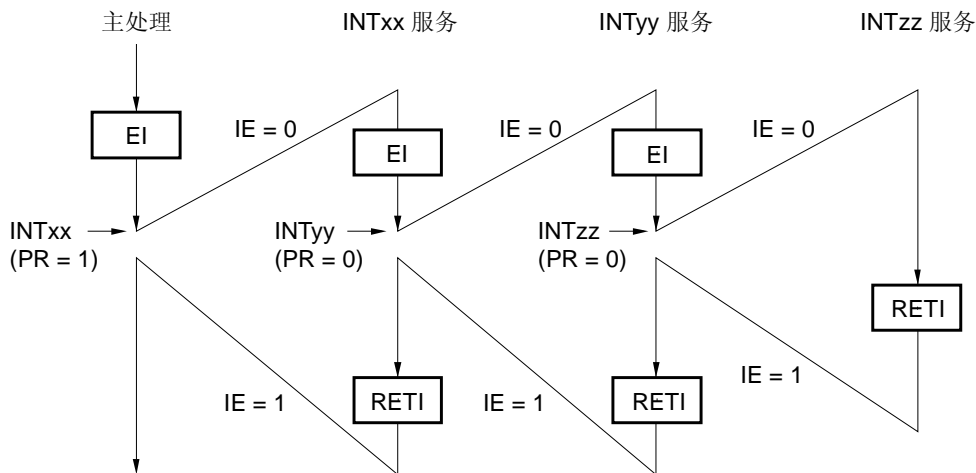
<R> 表 19-5. 中断服务期间为多重中断服务所允许的中断请求之间的关系

多重中断请求		非屏蔽中断请求	可屏蔽中断请求				软件中断请求
			PR = 0		PR = 1		
			IE = 1	IE = 0	IE = 1	IE = 0	
正在进行的 interrupt							
非屏蔽中断请求		×	×	×	×	×	○
可屏蔽中断	ISP = 0	○	○	×	×	×	○
	ISP = 1	○	○	×	○	×	○
软件中断		○	○	×	○	×	○

- 备注**
- : 允许多重中断服务
 - ×: 禁止多重中断服务
 - 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 PR0L、PR0H、PR1L 和 PR1H 中的标志。
 PR = 0: 高优先级
 PR = 1: 低优先级

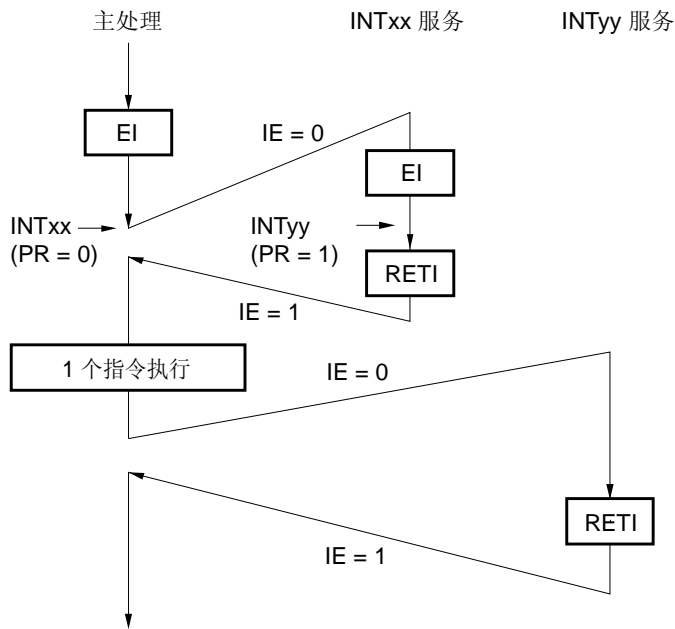
图 19-12. 多重中断服务的示例 (1/2)

例 1. 发生两次多重中断服务



在中断 INTxx、两个中断请求 (INTyy 和 INTzz) 的服务响应期间发生了多重中断服务。在每个中断请求被响应之前, EI 指令必须被发出以允许中断请求响应。

例 2. 由于优先级控制没有发生多重中断服务

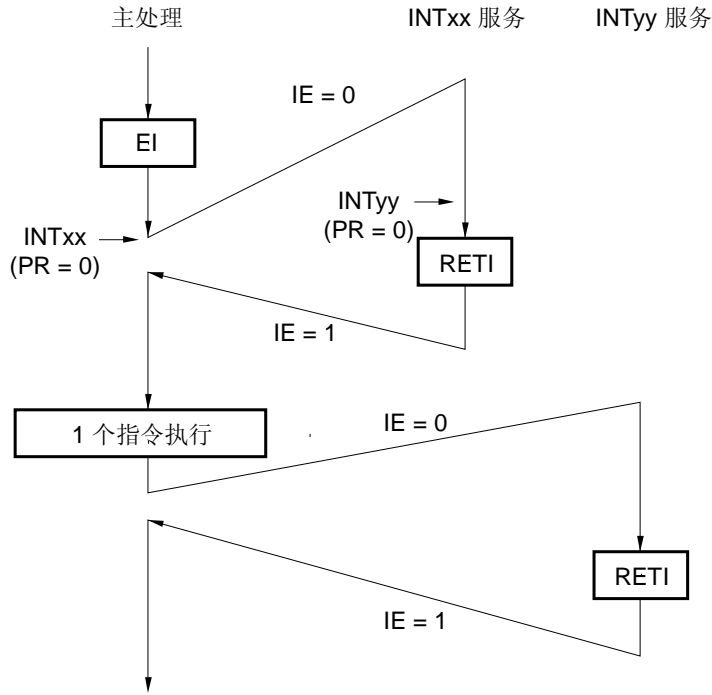


在中断 INTxx 的服务期间发出的中断请求 INTyy 由于其优先级低于 INTxx 的优先级, 因此它将被不会响应, 而多重中断服务也不会发生。INTyy 中断请求被保持为未决状态, 并且在执行一个主程序指令后被响应。

- PR = 0: 高优先级
- PR = 1: 低优先级
- IE = 0: 禁止中断请求响应。

图 19-12. 多重中断服务的示例 (2/2)

例 3. 多重中断服务不会发生，因为中断不被允许



在中断 INTxx 服务期间不允许中断 (EI 指令没有发出)，因此，中断请求 INTyy 不会被响应，而多重中断服务也不会发生。INTyy 中断请求被保持为未决状态，并且在执行一个主程序指令后被响应。

PR = 0: 高优先级

IE = 0: 禁止中断请求响应。

19.4.5 中断请求保持

即使在另一个指令执行时为指令发出中断请求，指令也会在请求响应被保持为未决状态的位置存在，直至下个指令结束。以下列出了这些指令（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 指令不是以下列出的中断请求保持指令。然而，通过执行 BRK 指令引起的软件中断导致 IE 标志被清除为 0。因此，如果在 BRK 指令执行期间一个可屏蔽中断请求被产生，中断请求不被响应。然而，非屏蔽中断请求被响应。

图 19-13 显示了中断请求被保持为未决状态的时序。

图 19-13. 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。

第二十章 待机功能

20.1 待机功能和配置

20.1.1 待机功能

表 20-1. 每种操作状态中操作时钟的关系

状态 操作模式	X1 振荡器		内部振荡器			释放后的 CPU 时钟	提供给周边的预分频器 时钟	
	MSTOP = 0	MSTOP = 1	注 1	注 2			MCM0 = 0	MCM0 = 1
				RSTOP = 0	RSTOP = 1			
复位	停止		停止			内部振荡 时钟	停止	
STOP			振荡	振荡	停止 ^{注 4}	注 5	停止	
HALT	振荡	停止 ^{注 3}				注 6	内部振荡 时钟	X1

- 注
1. 当通过选项字节对内部振荡器选择“不能被停止”时
 2. 当通过选项字节对内部振荡器选择“可以由软件停止”时
 3. 只有当内部振荡器振荡时
 4. 只有当 X1 振荡时
 5. STOP 指令执行时使用 CPU 时钟操作。
 6. HALT 指令执行时使用 CPU 时钟操作。

注意事项 只有当通过选项字节对内部振荡器选择“可以由软件停止”时，RSTOP 设置才有效。

备注

MSTOP: 主 OSC 控制寄存器 (MOC) 的第 7 位
RSTOP: 内部振荡模式寄存器 (RCM) 的第 0 位
MCM0: 主时钟模式寄存器 (MCM) 的第 0 位

待机功能设计用来降低系统工作电流。有以下两种模式可选。

(1) HALT 模式

HALT 指令执行设置了 HALT 模式。在 HALT 模式中，CPU 操作时钟被停止。如果在 HALT 模式被设置之前 X1 振荡器或内部振荡器正在操作，每个时钟的振荡继续。在这种模式中，工作电流不会像在 STOP 模式中那样被减少那么多，但 HALT 模式对于在中断请求生成后立即恢复操作以及执行间隙运行是有效的。

(2) STOP 模式

STOP 指令执行设置了 STOP 模式。在 STOP 模式下，X1 振荡器停止，停止整个系统，因此可以显著减少 CPU 工作电流。

因为该模式可以通过中断请求来清除，因此它允许执行间隙运行。然而，因为 STOP 模式被释放后将会需要一个等待时间来确保振荡稳定时间的安全，因此如果有必要在中断请求生成的情况下立即开始处理，则需选择 HALT 模式。

在以上两种模式中的任意一种模式中，寄存器、标志以及数据存储器中的所有内容在待机状态被设置前都会被保持。输入/输出端口输出锁存以及输出缓存的状态也会被保持。

- 注意事项**
1. 当 CPU 以 X1 输入时钟或内部振荡时钟操作时，STOP 和 HALT 模式可以被使用。然而，当 STOP 指令在内部振荡时钟操作期间被执行时，X1 振荡器停止，但是内部振荡器不会停止。
 2. 当转至 STOP 模式时，应确保在执行 STOP 指令前停止周边硬件操作。
 3. 当使用待机功能时，建议按以下顺序来实现 A/D 转换器的电流减少：先将 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 清 0，用来停止 A/D 转换操作，然后再执行 HALT 或 STOP 指令。
 4. 在 STOP 模式下，如果在 STOP 模式被设置之前内部振荡器正在操作，内部振荡时钟的振荡不能被停止。然而，当内部振荡时钟用作 CPU 时钟时，在 STOP 模式被释放后，CPU 操作被停止 $17/f_r$ (s)。

20.1.2 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。

- 振荡稳定时间计数器状态寄存器（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，参见第五章 时钟发生器。

(1) 振荡稳定时间计数器状态寄存器（OSTC）

这是 X1 输入时钟振荡稳定时间计数器的状态寄存器。如果内部振荡时钟用作 CPU 时钟，X1 输入时钟振荡稳定时间可以被检查。

OSTC 可以通过 1 位或 8 位存储器操作指令来读取。

复位释放（通过 RESET 输入、POC、LVI、WDT 复位）、STOP 指令和 MSTOP（MOC 寄存器的第 7 位）=1 将会把 OSTC 清除为 00H。

图 20-1. 振荡稳定时间计数器状态寄存器（OSTC）的格式

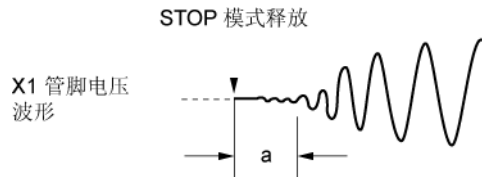
地址：FFA3H 复位后：00H R

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

MOST11	MOST13	MOST14	MOST15	MOST16	振荡稳定时间状态	
					$f_{XP} = 20\text{MHz}$	
1	0	0	0	0	$2^{11}/f_{XP}$ 最小值	102.4 μs 最小
1	1	0	0	0	$2^{13}/f_{XP}$ 最小值	409.6 μs 最小
1	1	1	0	0	$2^{14}/f_{XP}$ 最小值	819.2 μs 最小
1	1	1	1	0	$2^{15}/f_{XP}$ 最小值	1.64 ms 最小
1	1	1	1	1	$2^{16}/f_{XP}$ 最小值	3.27 ms 最小

- 注意事项**
1. 上述时间过去后，位将会从 MOST11 按顺序被设为 1 并且保持为 1。
 2. 如果先进入 STOP 模式，然后在内部振荡时钟被用作 CPU 时钟时释放 STOP 模式，则将会按以下方式设置振荡稳定时间。
 - 期望 OSTC 振荡稳定时间 \leq 由 OSTS 所设置的振荡稳定时间

X1 振荡稳定时间计数器累计只在由 OSTS 所设置的振荡稳定时间内计数。因此，需注意在 STOP 模式被释放后，只有由 OSTS 所设置的振荡稳定时间内的状态才会被设置到 OSTC 中。
 3. STOP 模式释放时的等待时间不包含 STOP 模式释放后时钟振荡开始（下面的“a”）前的时间，无论 STOP 模式是否由 RESET 输入或中断产生来释放。



备注 fx: X1 输入时钟振荡频率

(2) 振荡稳定时间选择寄存器 (OSTS)

当 STOP 模式被释放时, 该寄存器用于选择 X1 振荡稳定等待时间。当 X1 输入时钟选作 CPU 时钟时, 由 OSTS 设置的等待时间只有在 STOP 模式被释放后才有效。当内部振荡时钟被选择时, STOP 模式被释放后, 使用 OSTC 检查振荡稳定时间。

OSTS 可以通过 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入设置 OSTS 为 05H。

图 20-2. 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

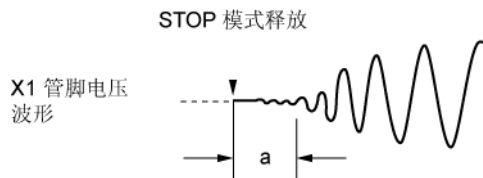
OSTS2	OSTS1	OSTS0	振荡稳定时间选择	
				$f_{XP} = 20\text{MHz}$
0	0	1	$2^{11}/f_{XP}$	102.4 μs
0	1	0	$2^{13}/f_{XP}$	409.6 μs
0	1	1	$2^{14}/f_{XP}$	819.2 μs
1	0	0	$2^{15}/f_{XP}$	1.64 ms
1	0	1	$2^{16}/f_{XP}$	3.27 ms
除上面以外			禁止设置	

注意事项 1. 如果先进入 STOP 模式, 然后在内部振荡时钟被用作 CPU 时钟时释放 STOP 模式, 则将会按以下方式设置振荡稳定时间。

- 期望 OSTC 振荡稳定时间 \leq 由 OSTS 所设置的振荡稳定时间

X1 振荡稳定时间计数器累计只在由 OSTS 所设置的振荡稳定时间内计数。因此, 需注意在 STOP 模式被释放后, 只有由 OSTS 所设置的振荡稳定时间内的状态才会被设置到 OSTC 中。

2. STOP 模式释放时的等待时间不包含 STOP 模式释放后时钟振荡开始 (下面的“a”) 前的时间, 无论 STOP 模式是否由 $\overline{\text{RESET}}$ 输入或中断产生来释放。



备注 fx: X1 输入时钟振荡频率

20.2 待机功能操作

20.2.1 HALT 模式

(1) HALT 模式

HALT 模式通过执行 HALT 指令来设置。不管 CPU 时钟在设置前是否是 X1 输入时钟或内部振荡器时钟，HALT 模式都可以被设置。

以下显示了在 HALT 模式中的运行状态。

表 20-2. 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 在 X1 输入时钟上运行		当 HALT 指令被执行时，此时 CPU 在内部振荡时钟上运行	
		当内部振荡时间继续时	当内部振荡时钟停止时 ^{注1}	当 X1 输入时钟振荡继续时	当 X1 输入时钟振荡停止时
项目					
系统时钟		提供给 CPU 的时钟被停止。			
CPU		操作停止			
端口（锁存）		设置 HALT 模式前的状态被保持			
10 位逆变器控制定时器		可运行		操作不被保证	
16 位向上/向下计数器 ITENC20		可运行		操作不被保证	
16 位定时器/事件计数器 00		可运行		操作不被保证	
8 位定时器/事件计数器 50		可运行		当 TI50 以外的计数时钟被选择时，操作不被保证	
8 位定时器/事件计数器 51		可运行		当 $f_{R/2}$ 或 TI51 以外的计数时钟被选择时，操作不被保证	
8 位定时器 H0		可运行		在 8 位定时器/事件计数器 50 操作期间，当 TM50 输出以外的计数时钟被选择时，操作不被保证	
看门狗计时器	内部振荡器不能被停止 ^{注2}	可运行	-	可运行	
	内部振荡器可以被停止 ^{注2}	操作停止			
时钟输出/蜂鸣器输出控制器		可运行			操作不被保证
实时输出口		可运行		当外部触发（INTP2）以外的触发被使用或当 TI51 以外的时钟被选作 8 位定时器/事件计数器 51 的计数时钟时，操作不被保证	
A/D 转换器		可运行		操作不被保证	
串行接口	UART00	可运行		在 8 位定时器/事件计数器 50 操作期间，当 TM50 输出以外的串行时钟被选择时，操作不被保证	
	CSI10	可运行		当外部 SCK10 以外的串行时钟被选择时，操作不被保证	
乘法器/除法器		可运行		操作不被保证	
上电清零功能		可运行			
低压检测功能		可运行			
外部中断		可运行			

- 注
1. 当通过选项字节对内部振荡器选择“由软件停止”时和内部振荡器由软件停止时（关于选项字节，参见第二十四章 选项字节）。
 2. “内部振荡器不能被停止”或“内部振荡器可以由软件停止”可以通过选项字节来选择。

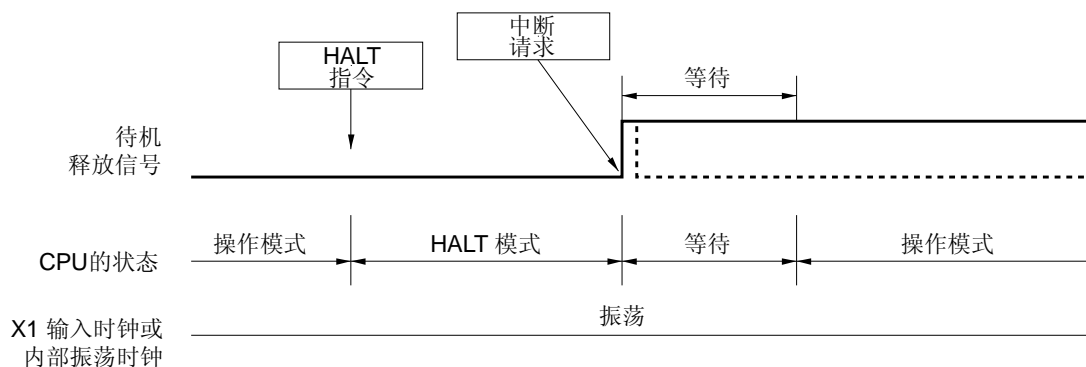
(2) HALT 模式释放

HALT 模式可以通过以下两种来源来释放。

(a) 通过未屏蔽的中断请求释放

当生成未屏蔽的中断请求时，HALT 模式将会被释放。如果允许中断响应，那么向量中断服务将会被执行。如果禁止中断响应，则将会执行下一个地址指令。

图 20-3. 通过生成中断请求来释放 HALT 模式



- 备注**
1. 虚线表明已经释放了待机状态的中断请求被响应时的情况。
 2. 等待时间如下：
 - 当向量中断服务被执行时：8 或 9 个时钟
 - 当向量中断服务没有被执行时：2 或 3 个时钟

<R>

(b) 通过非屏蔽中断请求释放

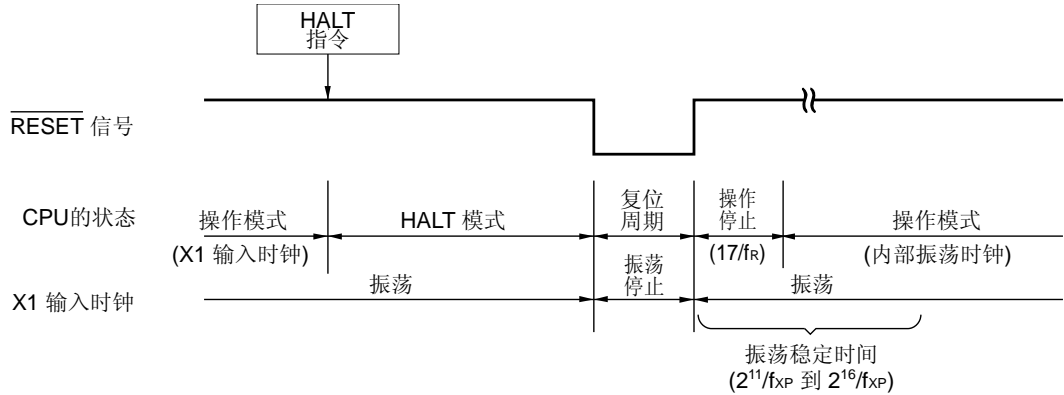
当一个非屏蔽中断请求被产生时，HALT 模式被释放并且不论中断响应被允许或禁止，向量中断服务都被执行。

(c) 通过 $\overline{\text{RESET}}$ 输入释放

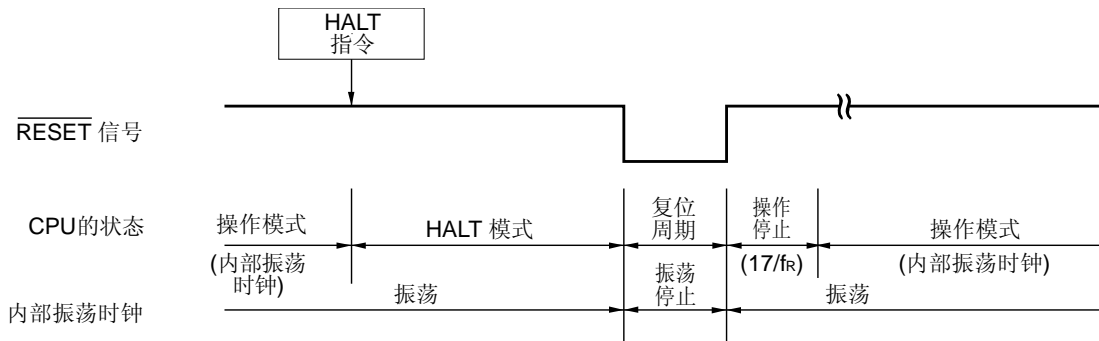
当 $\overline{\text{RESET}}$ 信号被输入时，HALT 模式将会被释放，之后，在使用正常复位操作的情况下，程序将会在分支到复位向量地址后被执行。

图 20-4. 通过 $\overline{\text{RESET}}$ 输入释放 HALT 模式

(1) 当 X1 输入时钟被用作 CPU 时钟时



(2) 当内部振荡时钟被用作 CPU 时钟时



- 备注
1. f_{XP} : X1 输入时钟振荡频率
 2. f_R : 内部振荡时钟频率

表 20-3. HALT 模式中中断请求响应操作

释放源	MK _{xx}	PR _{xx}	IE	ISP	操作
可屏蔽中断请求	0	0	0	×	下一地址指令执行
	0	0	1	×	中断服务执行
	0	1	0	1	下一地址指令执行
	0	1	×	0	
	0	1	1	1	中断服务执行
	1	×	×	×	HALT 模式保持
非屏蔽中断请求	–	–	×	×	中断服务执行
RESET 输入	–	–	×	×	服务处理

×: 不关注

<R>

20.2.2 STOP 模式

(1) STOP 模式设置及运行状态

通过执行 STOP 指令，STOP 模式被设置，并且无论 CPU 时钟在设置之前是 X1 输入时钟还是内部振荡时钟，它都将被设置。

注意事项 因为中断请求信号被用于清除待机模式，因此，如果有一个带有中断请求标志设置以及中断屏蔽标志复位的中断源时，如果中断请求被设置，那么待机将会立即被清除。因此，执行 STOP 指令并且系统返回操作模式之后，其中系统是在使用振荡稳定时间选择寄存器（OSTS）的等待时间设置过去后立即返回操作模式的，STOP 模式将会立即被复位到 HALT 模式。

以下显示了 STOP 模式中的运行状态。

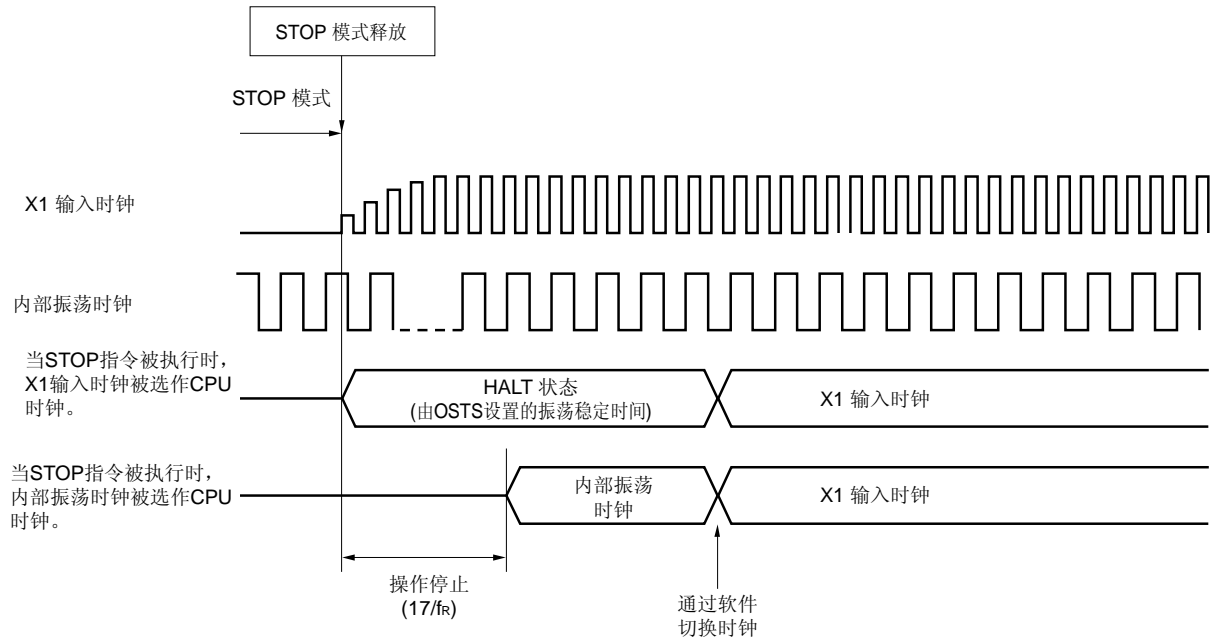
表 20-4. STOP 模式中的运行状态

STOP 模式设置		当 STOP 指令被执行时，此时 CPU 在 X1 输入时钟上运行		当 STOP 指令被执行时，此时 CPU 在内部振荡时钟上运行
		当内部振荡时间继续时	当内部振荡时钟停止时 ^{注1}	
系统时钟		只有 X1 振荡器振荡被停止。提供给 CPU 的时钟被停止。		
CPU		操作停止		
端口（锁存）		设置 STOP 模式前的状态被保持		
10 位逆变器控制定时器		操作停止		
16 位向上/向下计数器 ITENC20		操作停止		
16 位定时器/事件计数器 00		操作停止		
8 位定时器/事件计数器 50		只有当 TI50 被选作计数时钟时可运行		
8 位定时器/事件计数器 51		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		
看门狗定时器	内部振荡器不能被停止 ^{注2}	可运行	-	可运行
	内部振荡器可以被停止 ^{注2}	操作停止		
时钟输出/蜂鸣器输出控制器		操作停止		
实时输出端口		只有当 TM51 操作期间 INTTM51 被选作计数时钟或外部触发（INTP2）被选择时才可运行		
A/D 转换器		操作停止		
串行接口	UART00	只有当 TM50 操作期间 TM50 输出被选作计数时钟时才可运行		
	CSI10	只有当外部 SCKT0 被选作串行时钟时才可运行		
乘法器/除法器		操作停止		
上电清零功能		可运行		
低压检测功能		可运行		
外部中断		可运行		

- 注
1. 当通过选项字节对内部振荡器选择“由软件停止”时和内部振荡器由软件停止时（关于选项字节，参见第二十四章 选项字节）。
 2. “内部振荡器不能被停止”或“内部振荡器可以由软件停止”可以通过选项字节来选择。

(2) STOP 模式释放

图 20-5. STOP 模式被释放时的运行时间



STOP 模式可以通过以下三种源来释放。

(a) 通过未屏蔽中断请求释放

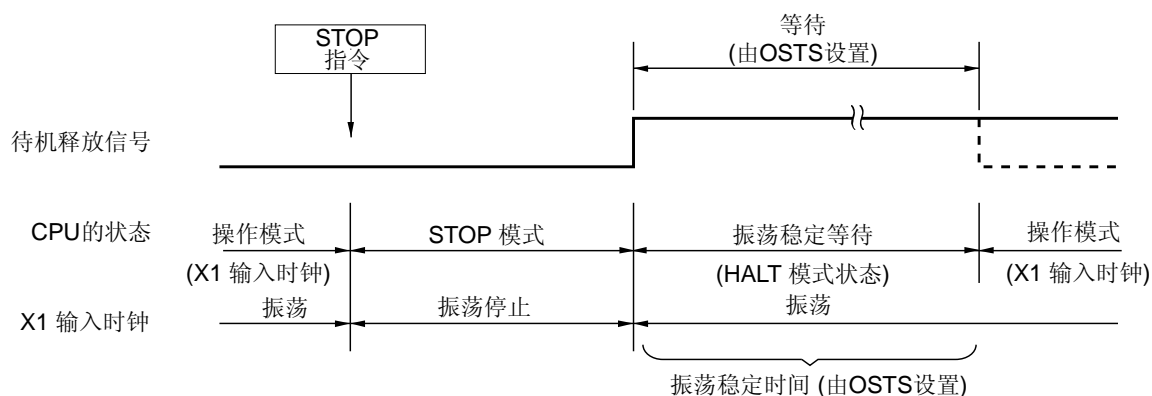
当生成未屏蔽的中断请求时，STOP 模式将会被释放。在振荡稳定时间过去之后，如果允许中断响应，则将会执行向量中断服务。如果禁止中断响应，则将会执行下一个地址指令。

<R> (b) 通过非屏蔽中断请求释放

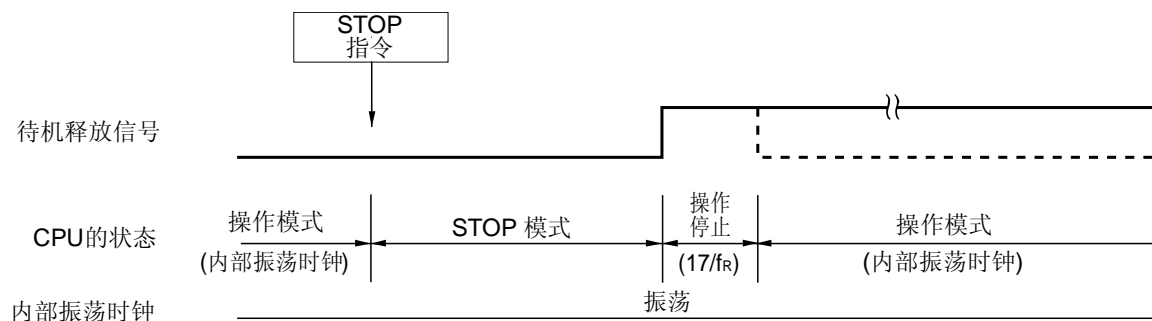
当一个非屏蔽中断请求产生时，STOP 模式被释放，无论中断响应被运行或禁止。并且振荡稳定时间过去后，向量中断服务被执行。

图 20-6. 通过生成中断请求来释放 STOP 模式

(1) 当 X1 输入时钟被用作 CPU 时钟时



(2) 当内部振荡时钟被用作 CPU 时钟时



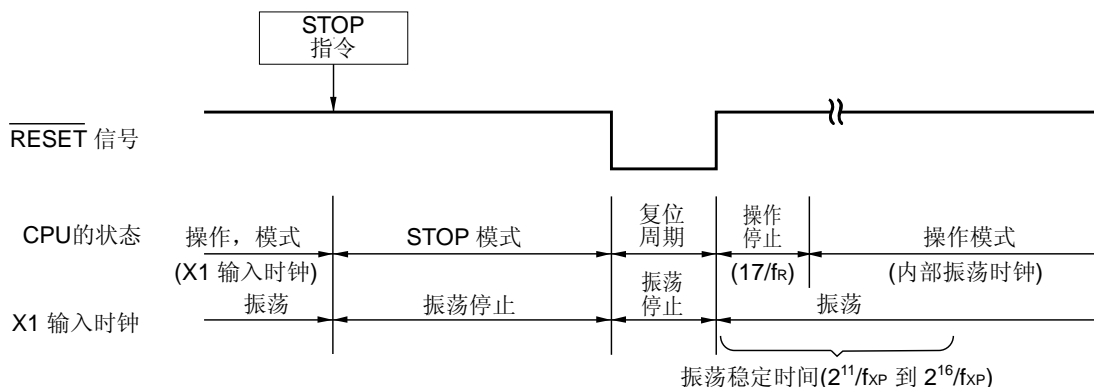
- 备注
1. 虚线表明已经释放了待机状态的中断请求被响应时的情况。
 2. fr: 内部振荡时钟频率

(c) 通过 $\overline{\text{RESET}}$ 输入释放

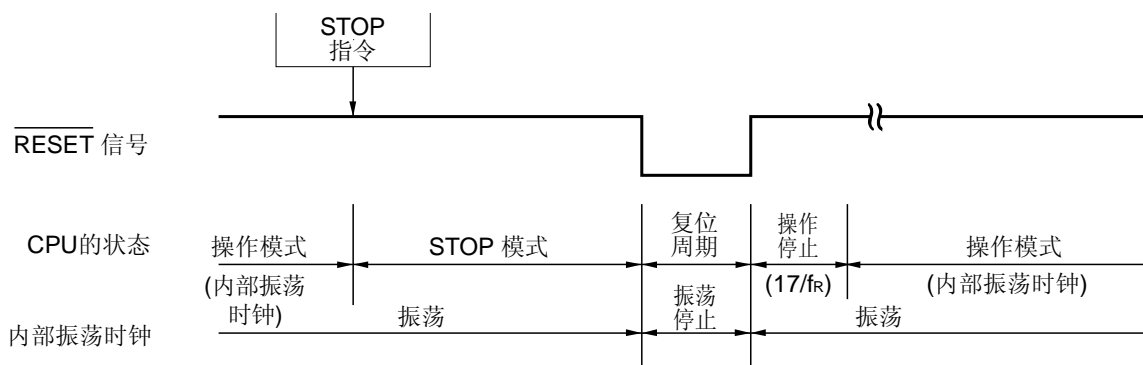
当 $\overline{\text{RESET}}$ 信号被输入时，STOP 模式被释放并且在振荡稳定时间过去后，复位操作被执行。

图 20-7. 通过 $\overline{\text{RESET}}$ 输入释放 STOP 模式

(1) 当 X1 输入时钟被用作 CPU 时钟时



(2) 当内部振荡时钟被用作 CPU 时钟时



- 备注
1. f_{XP} : X1 输入时钟振荡频率
 2. f_R : 内部振荡时钟频率

表 20-5. STOP 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	×	下一地址指令执行
	0	0	1	×	中断服务执行
	0	1	0	1	下一地址指令执行
	0	1	×	0	
	0	1	1	1	中断服务执行
	1	×	×	×	STOP 模式保持
非屏蔽中断请求	-	-	×	×	中断服务执行
$\overline{\text{RESET}}$ 输入	-	-	×	×	服务处理

×: 不关注

<R>

第二十一章 复位功能

以下四种操作可以用于生成复位信号。

- (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) 通过比较低压检测电路 (LVI) 的工作电压与检测电压的内部复位

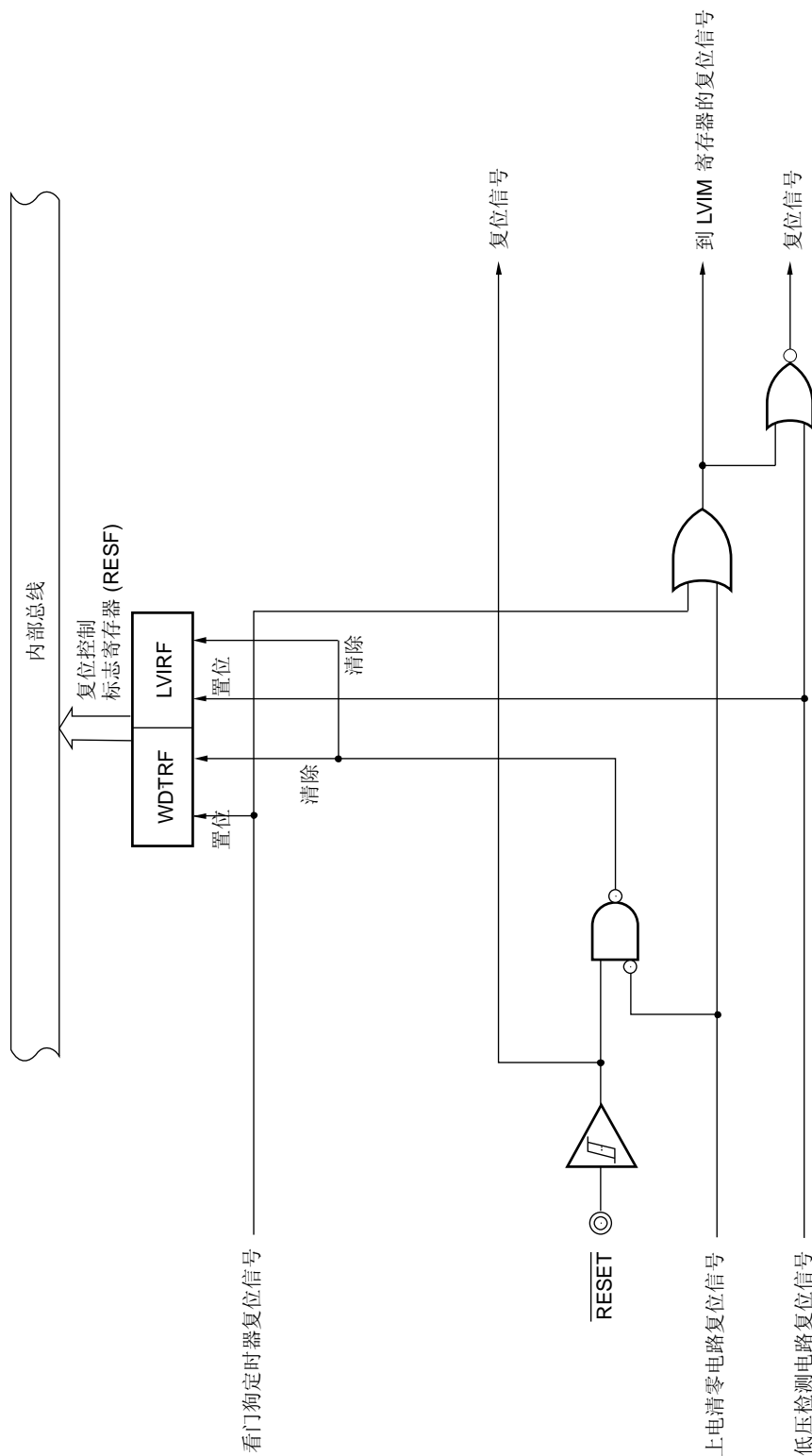
外部和内部复位没有功能差异。两种情况下，当复位信号输入时程序从地址 0000H 和 0001H 处开始执行。

低电平被输入到 $\overline{\text{RESET}}$ 引脚时，看门狗定时器溢出时，或通过 POC 及 LVI 电路电压检测，以及硬件中的每一项被设置为显示在表 21-1 中的状态时，复位被应用。在复位输入期间或在复位释放后的振荡稳定时间内，每个引脚为高阻抗。

当高电平输入到 $\overline{\text{RESET}}$ 引脚时，复位被释放并且在 CPU 时钟操作停止 $17/f_R$ (s) 后程序使用内部振荡时钟开始执行。由看门狗定时器产生的复位在复位后自动被释放，并且在 CPU 时钟操作停止 $17/f_R$ (s) 后程序使用内部振荡时钟开始执行 (参见图 21-2 到 21-4)。复位后，当 $V_{DD} > V_{POC}$ 或 $V_{DD} > V_{LVI}$ 时，通过 POC 和 LVI 电路电源检测生成的复位将会被自动释放，并且在复位处理后，并且在 CPU 时钟操作停止 $17/f_R$ (s) 后程序使用内部振荡时钟开始执行 (参见第二十二章 上电清零电路 以及第二十三章 低压检测电路)。

- 注意事项**
1. 为外部复位将 $10\mu\text{s}$ 或更高的低电平输入到 $\overline{\text{RESET}}$ 引脚中。
 2. 复位输入期间，X1 输入时钟和内部振荡时钟停止振荡。
 3. 当 STOP 模式通过复位被释放时，在 STOP 模式中的内容在复位输入期间将会被保持。然而，端口引脚变为高阻抗。

图 21-1. 复位功能框图



注意事项 LVI 电路内部复位不会复位 LVI 电路。

备注 LVIM: 低压检测寄存器

图 21-2. 通过 RESET 输入的复位时序

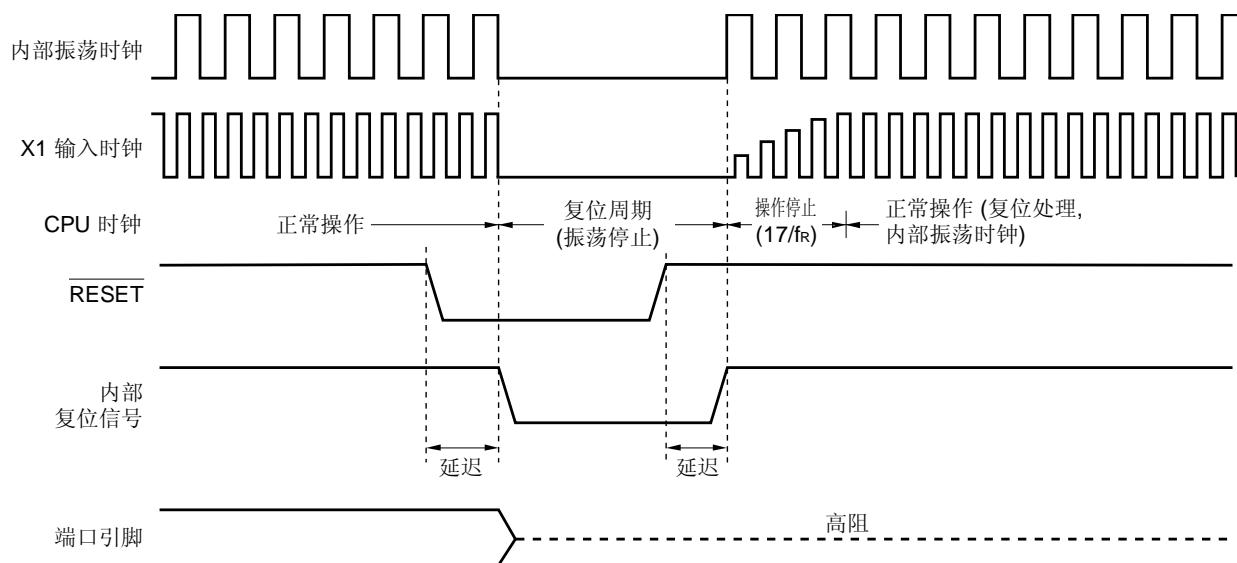
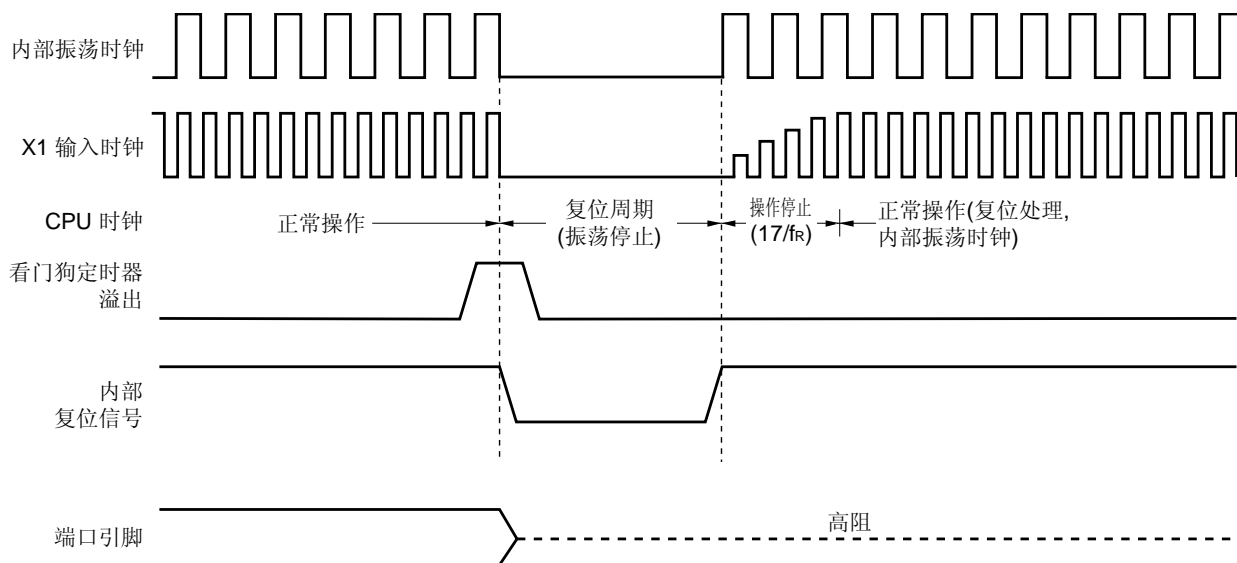
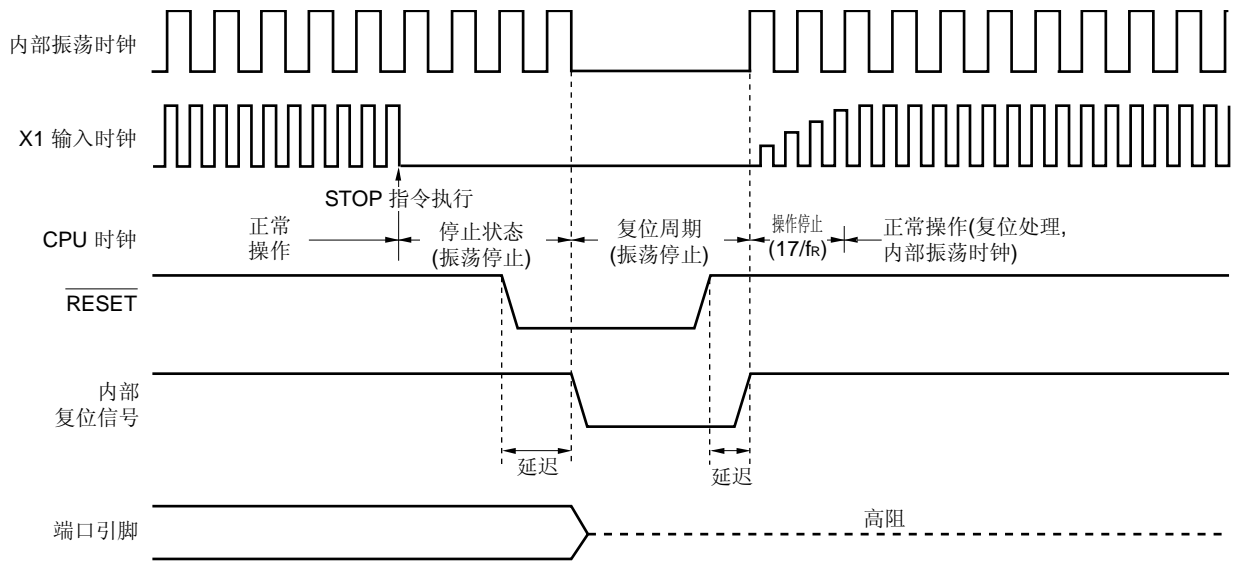


图 21-3. 由于看门狗定时器溢出而复位的时序



注意事项 看门狗定时器内部复位会复位看门狗定时器。

图 21-4. 通过 $\overline{\text{RESET}}$ 输入在 STOP 模式中复位的时序



备注 关于上电清零电路和低压检测电路的复位时间，参见第二十二章 上电清零电路 和 第二十三章 低压检测电路。

表 21-1. 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)		复位矢量表 (0000H, 0001H) 中的内容被设置,
堆栈指针 (SP)		不确定
程序状态字 (PSW)		02H
RAM	数据存储器	不确定 ^{注2}
	通用寄存器	不确定 ^{注2}
端口寄存器 (P0 到 P7) (输出锁存)		00H (只对 P2 不确定)
端口模式寄存器 (PM0, PM1, PM3 到 PM7)		FFH
上拉电阻选择寄存器 (PU0, PU1, PU3 到 PU7)		00H
内部存储器大小切换寄存器 (IMS)		CFH
处理器时钟控制寄存器 (PCC)		00H
内部振荡模式寄存器 (RCM)		00H
主时钟模式寄存器 (MCM)		00H
主 OSC 控制寄存器 (MOC)		00H
振荡稳定时间选择寄存器 (OSTS)		05H
振荡稳定时间计数器状态寄存器 (OSTC)		00H
系统等待控制寄存器 (VSWC)		00H
10 位逆变器控制定时器	比较寄存器 (TW0CM0 到 TW0CM2, TW0CM4, TW0CM5)	000H
	比较寄存器 (TW0CM3)	0FFH
	缓存寄存器 (TW0BFCM0 到 TW0BFCM2, TW0BFCM4, TW0BFCM5)	000H
	缓存寄存器 (TW0BFCM3)	0FFH
	死区时间重新加载寄存器 (TW0DTIME)	FFH
	控制寄存器 (TW0C)	00H
	模式寄存器 (TW0M)	00H
	A/D 触发选择寄存器 (TW0TRGS)	00H
	输出控制寄存器 (TW0OC)	00H
16 位向上/向下计数器 ITENC20	向上/向下计数器 (IT20UDC)	0000H
	比较寄存器 0, 1 (IT20CM0, IT20CM1)	0000H
	捕获/比较寄存器 0, 1 (IT20CC0, IT20CC1)	0000H
	单元模式寄存器 (IT20TUM)	00H
	控制寄存器 (IT20TMC)	00H
	捕获/比较控制寄存器 (IT20CCR)	00H
	有效沿选择寄存器 (IT20SESA)	00H
	预分频模式寄存器 (IT20PRM)	07H
	状态寄存器 (IT20STS)	00H

- 注
1. 在复位输入或振荡稳定时间等待期间内, 只有在硬件状态中 PC 的内容变为不确定的。其他的硬件状态在复位后都保持不变。
 2. 在等待模式中执行复位时, 预先复位状态即使在复位后也将会被保持。

表 21-1. 复位响应后的硬件状态 (2/3)

硬件		复位响应后的状态
16 位定时器/事件计数器 00	定时器计数器 00 (TM00)	0000H
	捕获/比较寄存器 00, 01 (CR00, CR01)	0000H
	模式控制寄存器 00 (TMC00)	00H
	预比例模式寄存器 00 (PRM00)	00H
	捕获/比较控制寄存器 00 (CRC00)	00H
	定时器输出控制寄存器 00 (TOC00)	00H
8 位定时器/事件计数器 50, 51	定时器计数器 50, 51 (TM50, TM51)	00H
	比较寄存器 50, 51 (CR50, CR51)	00H
	定时器时钟选择寄存器 50, 51 (TCL50, TCL51)	00H
	模式控制寄存器 50, 51 (TMC50, TMC51)	00H
8 位定时器 H0	比较寄存器 00, 01 (CMP00, CMP01)	00H
	模式寄存器 (TMHMD0)	00H
时钟输出/蜂鸣器输出控制器	时钟输出选择寄存器 (CKS)	00H
看门狗定时器	模式寄存器 (WDTM)	67H
	允许寄存器 (WDTE)	9AH
实时输出端口	缓存寄存器 (RTBL00, RTBH00, RTBL01, RTBH01)	00H
	模式寄存器 (RTPM00, RTPM01)	00H
	控制寄存器 (RTPC00, RTPC01)	00H
	直流控制寄存器 (DCCTL00, DCCTL01)	00H
模/数转换器	转换结果寄存器 (ADCR)	不确定
	模式寄存器 (ADM)	00H
	模拟输入通道指定寄存器 (ADS)	00H
	掉电比较模式寄存器 (PFM)	00H
	掉电比较门限寄存器 (PFT)	00H
串行接口 UART00	接收缓冲寄存器 00 (RXB00)	FFH
	发送移位寄存器 00 (TXS00)	FFH
	异步串行接口操作模式寄存器 00 (ASIM00)	01H
	波特率产生器控制寄存器 00 (BRGC00)	1FH
串行接口 CSI10	发送缓冲寄存器 10 (SOTB10)	不确定
	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
乘法器/除法器	余数数据寄存器 0 (SDR0)	0000H
	乘法/除法数据寄存器 A0 (MDA0H, MDA0L)	0000H
	乘法/除法数据寄存器 B0 (MDB0)	0000H
	乘法器/除法器控制寄存器 0 (DMUC0)	00H

表 21-1. 复位响应后的硬件状态 (3/3)

硬件		复位响应后的状态
复位功能	复位控制标志寄存器 (RESF)	00H ^{#1}
低压检测	低压检测寄存器 (LVIM)	00H ^{#1}
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 (MK0L, MK0H, MK1L)	FFH
	屏蔽标志寄存器 1H (MK1H)	DFH
	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
Flash 存储器	Flash 保护命令寄存器 (PFCMD)	不确定
	Flash 状态寄存器 (PFS)	00H
	Flash 编程模式控制寄存器 (FLPMC)	0XH ^{#2}

注 1. 这些值根据复位源来变化。

复位源 寄存器	RESET 输入	通过 POC 复位	通过 WDT 复位	通过 LVI 复位
RESF	参见表 21-2.			
LVIM	清除 (00H)	清除 (00H)	清除 (00H)	保持

2. 该值根据操作模式而改变。

- 用户模式: 08H
- 片上模式: 0CH

21.1 确定复位源的寄存器

许多内部复位生成源存在于 μ PD78F0714中。复位控制标志寄存器（RESF）用于保存已经生成复位请求的源。RESF可以通过8位存储器操作指令来读取。

$\overline{\text{RESET}}$ 输入、上电清零（POC）电路所生成的复位以及读取RESF可以将RESF设为00H。

图 21-5. 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）所生成的内部复位请求
0	没有生成内部复位请求，或RESF被清除。
1	生成内部复位请求。

LVIRF	通过低压检测电路（LVI）所生成的内部复位请求
0	没有生成内部复位请求，或RESF被清除。
1	生成内部复位请求。

注 复位后的值根据复位源而变化。

注意事项 不能通过1位存储器操作指令来读取数据。

复位请求生成时RESF的状态显示在表21-2中。

图 21-2. 当复位请求生成时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)

第二十二章 上电清零电路

22.1 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。
- 比较电源电压（ V_{DD} ）和检测电压（ $V_{POC} = 3.5\text{ V} \pm 0.2\text{ V}^{\text{注}}$ ），当 $V_{DD} < V_{POC}$ 时，产生内部复位信号。

注 该值可能在评估后更改。

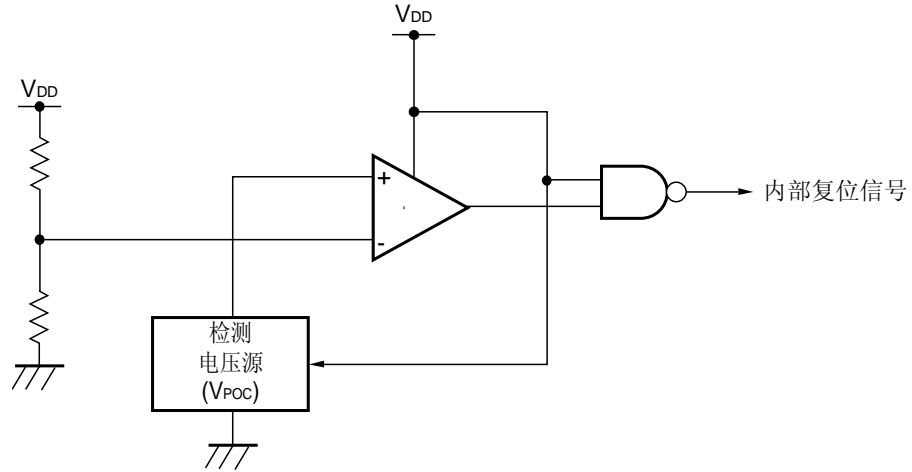
注意事项 如果在 POC 电路中内部复位信号被产生，复位控制标志寄存器（RESF）被清除为 00H。

备注 这个产品包含多个产生内部复位信号的硬件功能。当内部复位信号通过看门狗定时器（WDT）、低电压检测电路（LVI）产生时，一个表明复位源的标志位于复位控制标志寄存器（RESF）中。当内部复位信号被 WDT 或 LVI 产生时，RESF 不会被清除为 00H，并且标志被设置为 1。
关于 RESF 的详细情况，参见 **第二十一章 复位功能**。

22.2 上电清零电路的配置

上电清零电路的框图如图 22-1 所示。

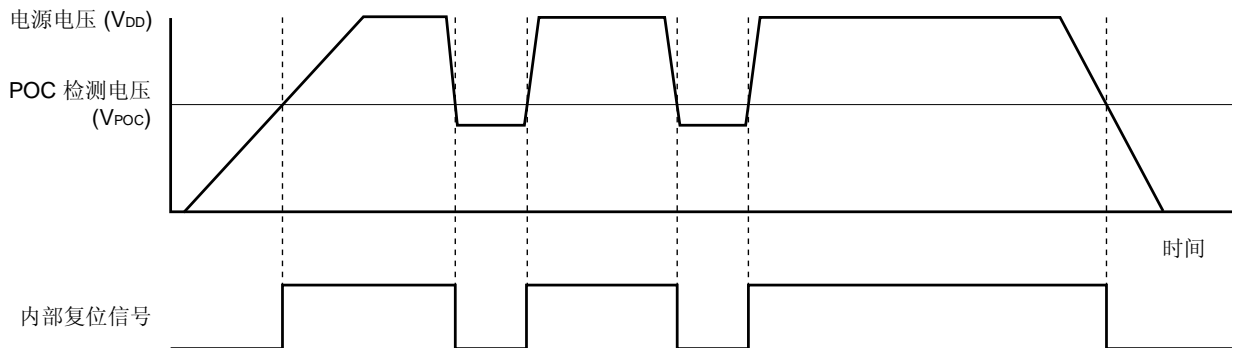
图 22-1. 上电清零电路的框图



22.3 上电清零电路的操作

在上电清零电路中，电源电压 (V_{DD}) 和检测电压 (V_{POC}) 被比较，当 $V_{DD} < V_{POC}$ 时，内部复位信号被产生。

图 22-2. 上电清零电路中内部复位信号产生的时序



22.4 上电清零电路的注意事项

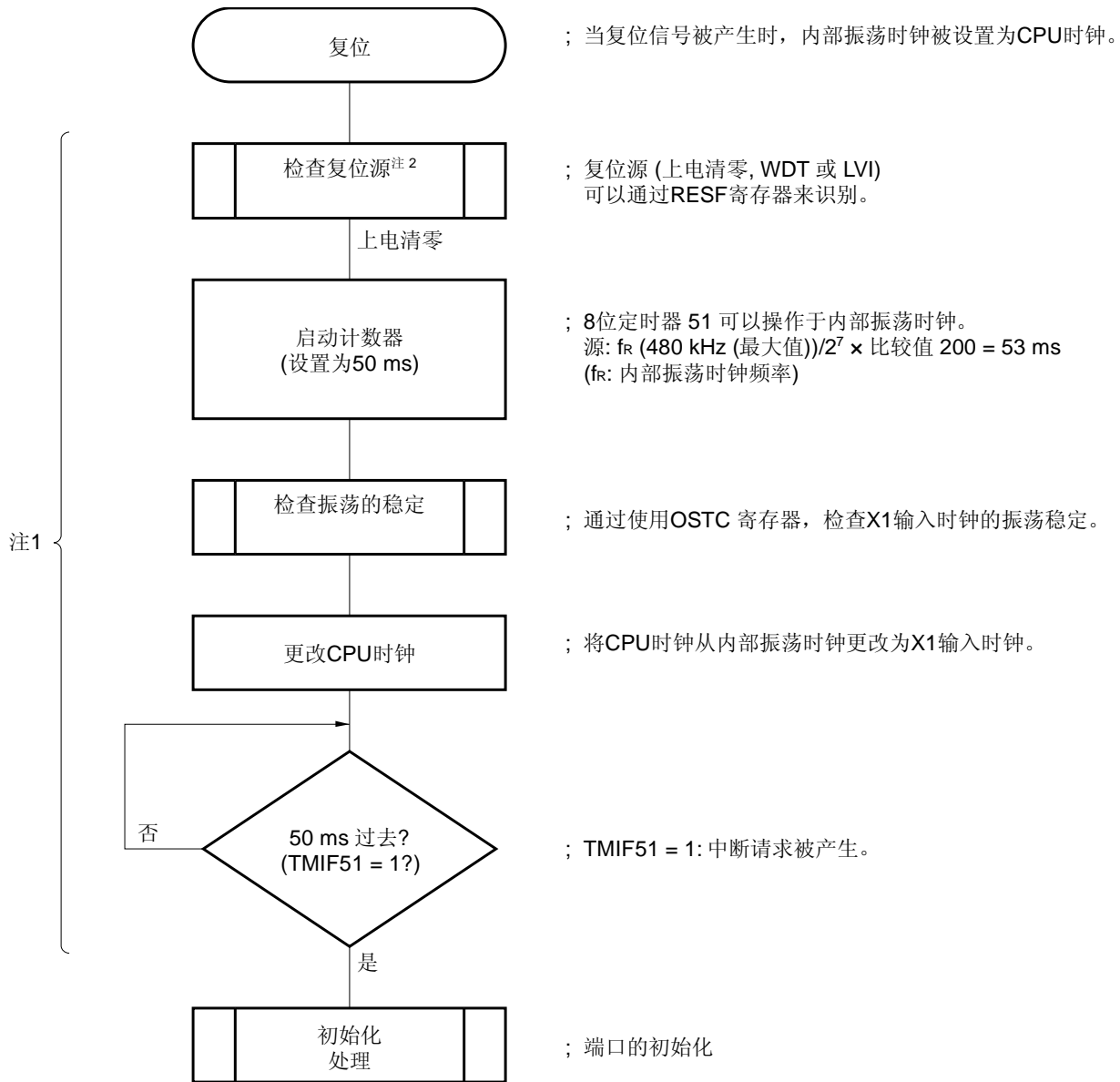
在一个电源电压 (V_{DD}) 以一定的周期在 POC 检测电压 (V_{POC}) 附近波动的系统中, 系统可能会被重复复位并从复位状态被释放。在这种情况下, 从复位释放到微控制器的操作开始的时间可以通过以下措施被任意设置。

<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 22-3. 复位释放后的软件处理举例 (1/2)

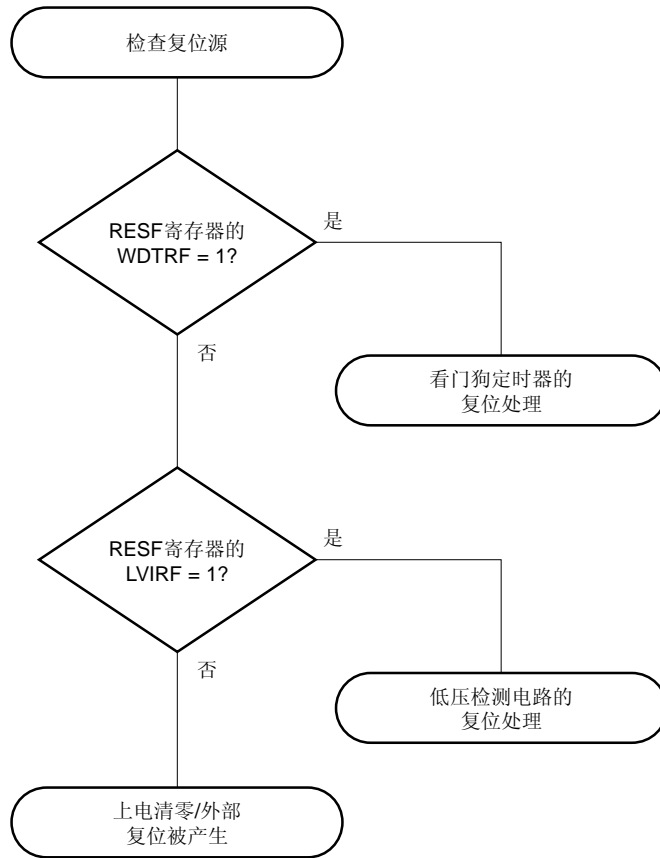
- 如果电源电压在 POC 检测电压附近的波动周期为 50ms 或更少



- 注
1. 如果在这个周期中复位被再次产生, 初始化处理不被启动。
 2. 流程图在下页被表示。

图 22-3. 复位释放后的软件处理举例 (2/2)

• 检查复位源



第二十三章 低电压检测电路

23.1 低电压检测电路的功能

低电压检测电路（LVI）具有以下功能。

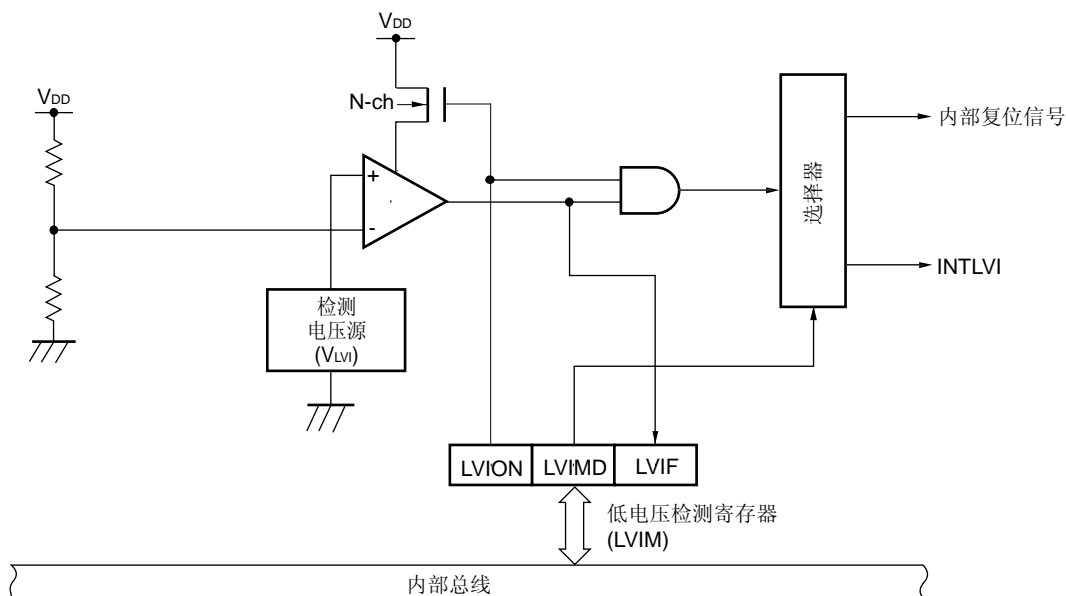
- 比较电源电压（ V_{DD} ）和检测电压（ $V_{LVI} = 4.3\text{ V} \pm 0.2\text{ V}$ ），并且当 $V_{DD} < V_{LVI}$ 时产生非屏蔽中断信号或内部复位信号。
- 中断或复位功能可通过软件选择。
- 在 STOP 模式下可以操作。

当低电压检测电路被用于复位时，如果复位发生，复位控制标志寄存器（RESF）的第 0 位（LVIRF）被设置为 1。关于 RESF 的详细情况，参见 第二十一章 复位功能。

23.2 低电压检测电路的配置

低电压检测电路的框图如下所示。

图 23-1. 低电压检测电路的框图



23.3 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)

(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。

这个寄存器可以通过一个 1 位或 8 位存储器操作指令来设置。

图 23-2. 低电压检测寄存器 (LVIM) 的格式

地址: FF78H 复位后: 00H R/W^{注1}

符号	7	6	5	4	3	2	1	0
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF

LVION ^{注2,3}	允许低电压检测操作
0	禁止操作
1	允许操作

LVIMD ^{注2}	低电压检测工作模式选择
0	当电源电压 (V _{DD}) < 检测电压 (V _{LVI}) 时, 产生中断信号
1	当电源电压 (V _{DD}) < 检测电压 (V _{LVI}) 时, 产生复位信号

LVIF ^{注4}	低电压检测标志
0	电源电压 (V _{DD}) > 检测电压 (V _{LVI}), 或者当操作禁止时
1	电源电压 (V _{DD}) < 检测电压 (V _{LVI})

- 注
1. 第 0 位只读。
 2. 在 LVI 复位以外的复位情况下, LVION 和 LVIMD 被清除为 0。在 LVI 复位的情况下, 这些不会被清除为 0。
 3. 当 LVION 被设置为 1 时, LVI 电路的比较器的操作被启动。使用软件产生一个从 LVION 被设置为 1 到电压在 LVIF 中被确认至少 0.2 ms 的等待。
 4. 当 LVION = 1 并且 LVIMD = 0 时, LVIF 的值作为中断请求信号 INTLVI 被输出。

注意事项 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。

23.4 低电压检测电路的操作

低电压检测电路可以被用作以下两种模式。

- 用作复位
比较电源电压 (V_{DD}) 和检测电压, 并且当 $V_{DD} < V_{LVI}$ 时产生一个内部复位信号。
- 用作中断
比较电源电压 (V_{DD}) 和检测电压, 并且当 $V_{DD} < V_{LVI}$ 时产生一个非屏蔽中断信号 (INTLVI)。

操作按照下面来设置。

<R> (1) 当用作复位时

- 当启动操作时
 - <1> 设置 LVIM 的第 7 位 (LVION) 为 1 (允许 LVI 操作)
 - <2> 使用软件产生一个至少 0.2 ms 的等待。
 - <3> 等待直到通过 LVIM 的第 0 位 (LVIF) 检查到 (电源电压 (V_{DD}) > 检测电压 (V_{LVI}))。
 - <4> 设置 LVIM 的第 1 位 (LVIMD) 为 1 (当电源电压 (V_{DD}) < 检测电压 (V_{LVI}) 时, 产生内部复位信号)。

图 23-3 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<1>到<4>。

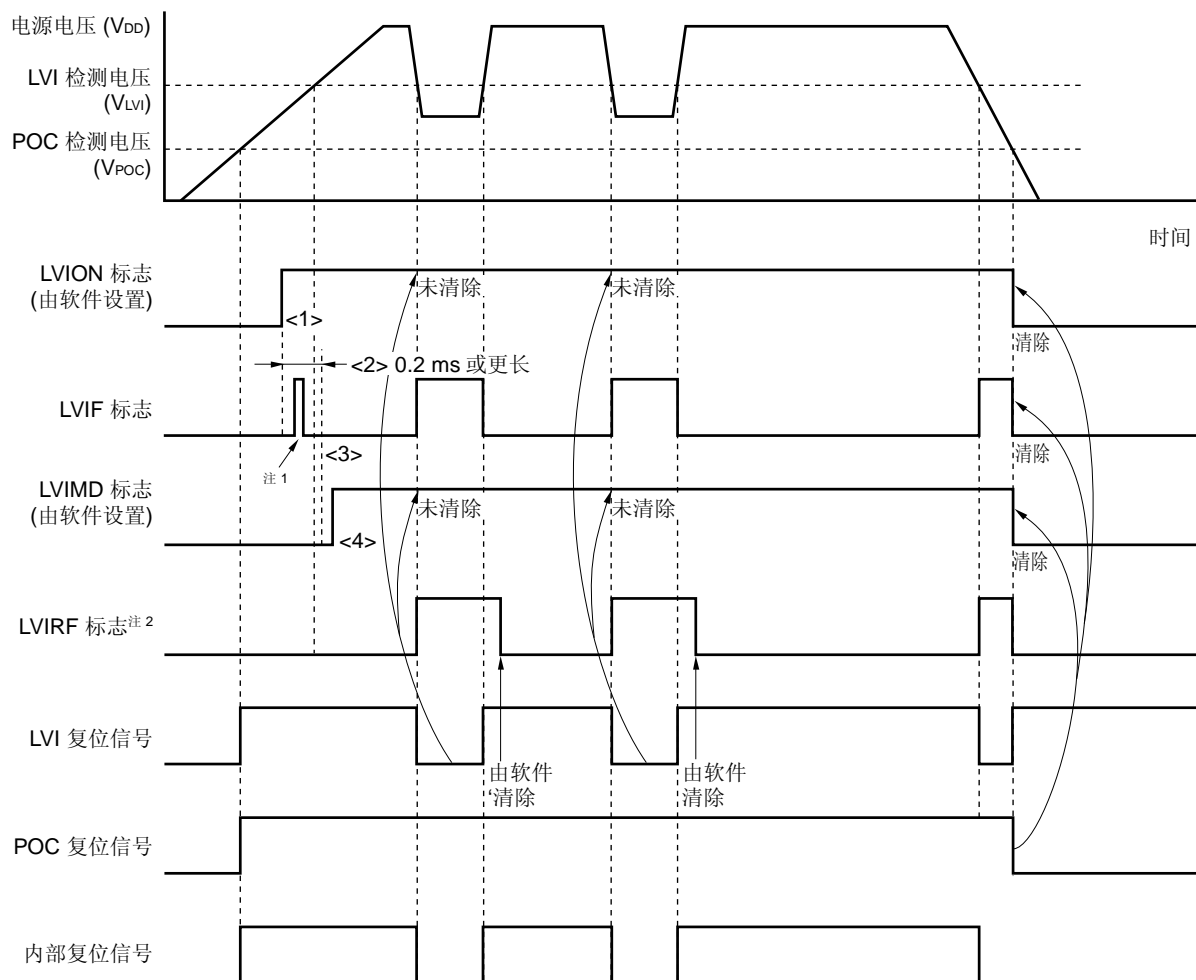
注意事项 1. 如果低电压检测电路用于复位, 一个非屏蔽中断信号 (INTLVI) 被产生。步骤<2> 到 <4>必须被添加到 LVI 中断复位程序中。

2. 当 LVIMD 被设置为 1 时, 如果电源电压 (V_{DD}) > 检测电压 (V_{LVI}), 内部复位信号不会被产生。

- 当停止操作时
下面任一过程必须被执行。
 - 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:
首先清除 LVIMD 为 0, 然后清除 LVION 为 0。

<R>

图 23-3. 低电压检测电路内部复位信号产生时序



- 注
1. LVIF 标志可设置为 1。
 2. LVIRF 是复位控制标志寄存器 (RESF) 的第 0 位。关于 RESF 的详细情况, 参见 第二十一章 复位功能。

备注 上面图 23-3 中的<1>到<4>对应于 23.4 (1) 当用作复位时的“当启动操作时”的描述中的<1>到<4>。

<R> (2) 当用作中断时

- 开始操作之前

事先作为全局变量定义具有以下意义的标志（任意名称）。

0: LVI 操作后，发现“电源电压 (V_{DD}) > 检测电压 (V_{LVI})”。

1: LVI 操作后，未发现“电源电压 (V_{DD}) > 检测电压 (V_{LVI})”。

- 当启动操作时

<1> 设置 LVIM 的第 7 位 (LVION) 为 1（允许 LVI 操作）

<2> 标志判断（0: 执行步骤<4> 到 <6>, 如下所示, 1: 执行步骤<3> 下面显示的后面的步骤。）

<3> 使用软件产生一个至少 0.2 ms 的等待。

<4> 等待直到通过 LVIM 的第 0 位 (LVIF) 检查到（电源电压 (V_{DD}) > 检测电压 (V_{LVI})）。

<5> 清除开始操作前设置的标志。

<6> 执行 EI 指令（当向量中断被使用时）。

- 低电压检测中断信号 (INTLVI) 产生后

<1> 标志判断（0: 执行步骤<4> 和 <5>, 如下所示, 1: 执行步骤<2> 下面显示的后面的步骤。）

<2> 使用软件产生一个至少 0.2 ms 的等待。

<3> 等待直到通过 LVIM 的第 0 位 (LVIF) 检查到（电源电压 (V_{DD}) > 检测电压 (V_{LVI})）。

<4> 执行低电压检测中断产生时要执行的程序。

<5> 返回主程序。

图 23-4 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“当启动操作时”中的 <1>、<3> 和 <4>。

- 当停止操作时

下面任一过程必须被执行。

- 当使用 8 位存储器操作指令时：

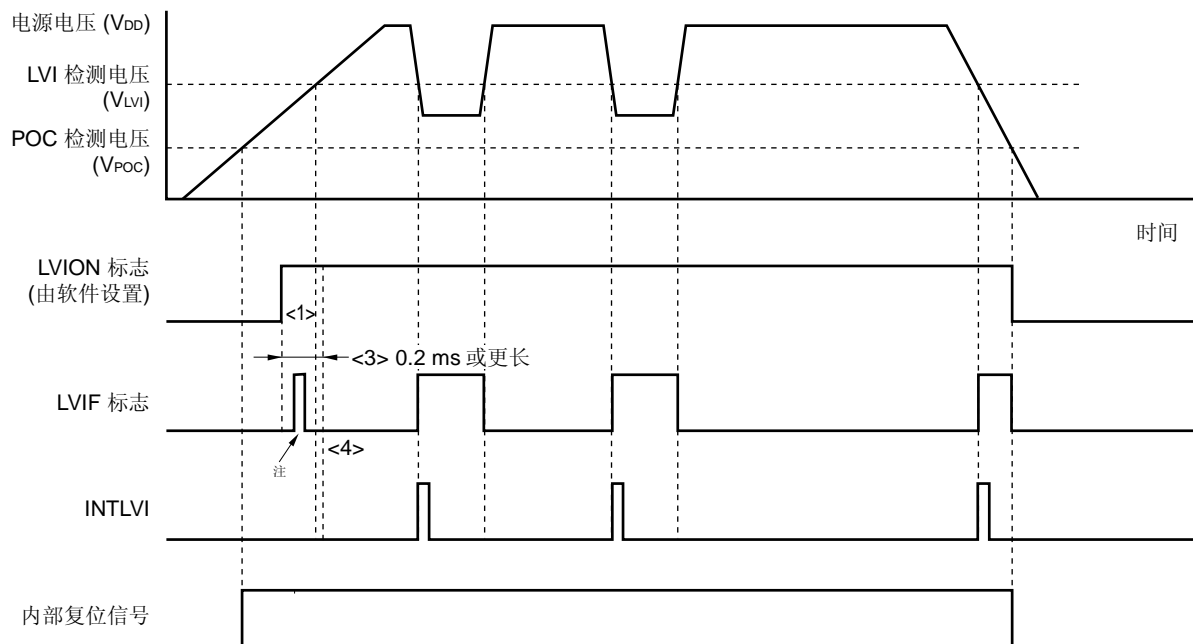
写入 00H 到 LVIM。

- 当使用 1 位存储器操作指令时：

清除 LVION 为 0。

<R>

图 23-4. 低电压检测电路中断信号产生时序



注 LVIF 标志可设置为 1。

备注 上面图 23-4 中的<1>、<3>和<4>对应于 23.4 (2) 当用作中断时的“当启动操作时”的描述中的<1>、<3>和<4>。

23.5 低电压检测电路的注意事项

在一个电源电压 (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如何被使用的操作如下。

(1) 当用作复位时

系统可能会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

中断请求可能被频繁产生。采取下面措施 (2)。

在该系统中, 采取以下措施。

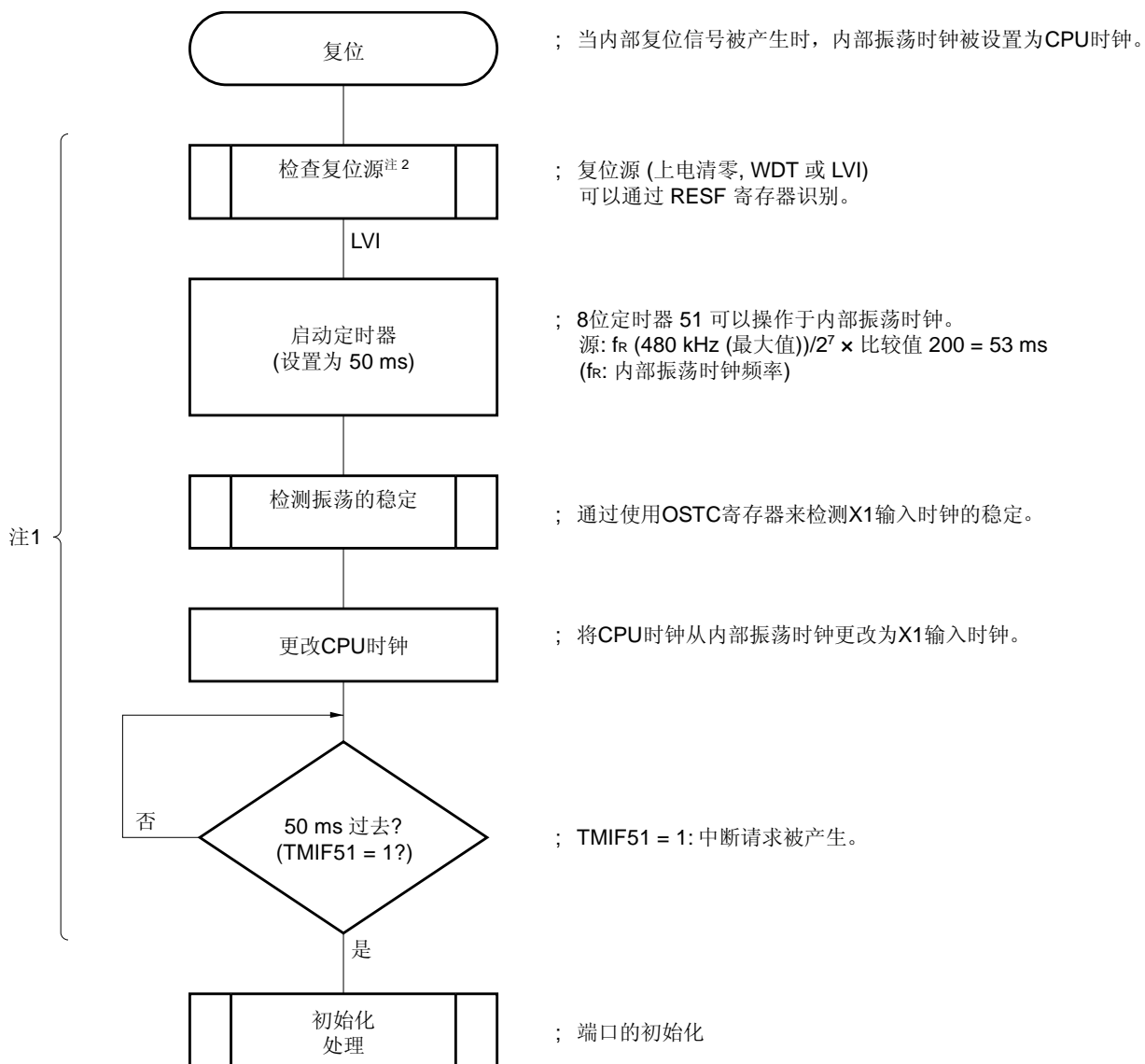
<措施>

(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。

图 23-5. 复位释放后的软件处理举例 (1/2)

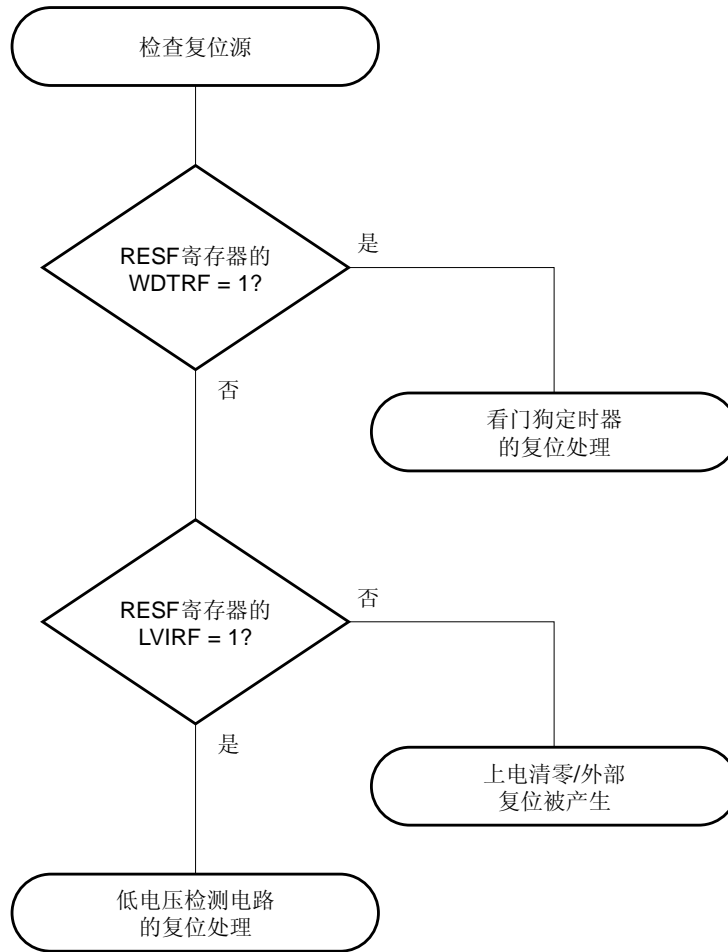
- 如果电源电压在 LVI 检测电压附近的波动周期为 50ms 或更少



- 注
1. 如果在这期间复位再次被产生, 初始化处理不被启动。
 2. 流程图在下页显示。

图 23-5. 复位释放后的软件处理举例（2/2）

• 检查复位源



(2) 当用作中断时

在 LVI 中断复位程序中使用低电压检测寄存器 (LVIM) 的第 0 位 (LVIF) 来检查“电源电压 (V_{DD}) > 检测电压 (V_{LVI})”。

如果电源电压波动周期在 LVI 检测电压附近很长时，等待电源电压波动周期，使用 LVIF 标志检查“电源电压 (V_{DD}) > 检测电压 (V_{LVI})”。

第二十四章 选项字节

μPD78F0714 可以使用选项字节实现停止或运行内部振荡器。

选项字节位于 flash 存储器中的地址 0080H。

当使用 flash 存储器版本产品时，确保在选项字节中将运行/禁止设置为停止内部振荡器。

图 24-1. 选项字节的分配

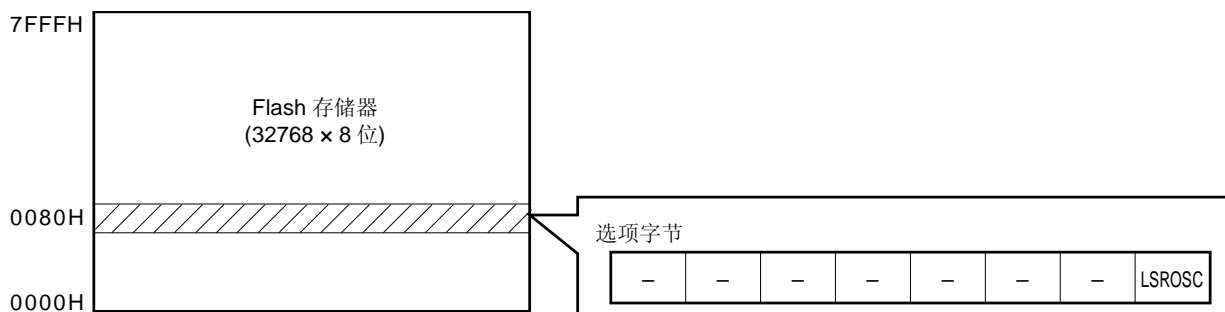
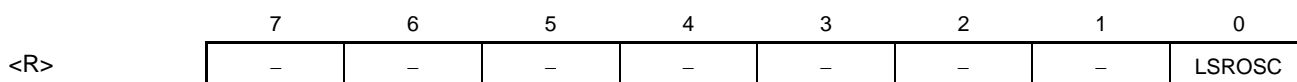


图 24-2. 选项字节的格式

地址: 0080H



LSROSC	内部振荡器操作
0	可以由软件停止
1	不能被停止

- 注意事项**
1. 要使用引导交换功能，确保保存引导簇 1 中的选项数据（关于引导交换功能，参见 25.8 引导交换功能）。
 2. 确保清除第 1 到 7 位为 0。

备注 设置选项字节的软件编码示例如下所示。

```
OPT      CSEG      AT 0080H
OPTION:  DB        01H      ; 设置到选项字节（内部振荡器不能被停止）
```


第二十五章 FLASH存储器

μ PD78F0714 包含 flash 存储器，当它安装在电路板上时，程序可以被写入、擦除和覆盖。

25.1 内部存储器容量切换寄存器

内部存储器容量通过使用内部存储器容量切换寄存器来设置。

IMS 由 8 位存储器操作指令设置。

RESET 输入设置 IMS 为 CFH。

注意事项 因为存储器容量切换寄存器（IMS）的初始值为 CFH，通过初始化设置 IMS 为 C8H。

图 25-1. 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

符号	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0
	RAM2	RAM1	RAM0	内部高速 RAM 容量选择				
	1	1	0	1024 字节				
	除上面以外			禁止设置				
	ROM3	ROM2	ROM1	ROM0	内部 ROM 容量选择			
	1	0	0	0	32 KB			
	除上面以外				禁止设置			

25.2 用Flash存储器编程器写入

通过使用专用的 flash 存储器编程器，数据可以在板上或板外被写入 flash 存储器。

(1) 板上编程

在 μ PD78F0714 被安装到目标系统中后，flash 存储器的内容可以被重新写入。连接专用 flash 存储器编程器的连接器必须安装到目标系统中。

(2) 板外编程

在 μ PD78F0714 被安装到目标系统中前，使用一个专用的程序适配器（FA 系列），数据可以被写入 flash 存储器。

备注 FA 系列是 Naito Densai Machida Mfg. Co., Ltd 的一个产品。

表 25-1. μ PD78F0714 和专用 Flash 存储器编程器之间的连线

专用 Flash 存储器编程器的引脚配置			使用 CSI10		使用 CSI10 + HS		使用 UART00	
信号名称	输入/输出	引脚功能	引脚名称	引脚编号	引脚名称	引脚编号	引脚名称	引脚编号
SI/RxD	输入	接收信号	SO10/P17	40	SO10/P17	40	TxD00/P14	37
SO/TxD	输出	发送信号	SI10/P16	39	SI10/P16	39	RxD00/P13	36
SCK	输出	发送时钟	$\overline{\text{SCK10}}$ /P15	38	$\overline{\text{SCK10}}$ /P15	38	不需要	不需要
CLK	输出	到 μ PD78F0714 的时钟	X1	6	X1	6	X1	6
			X2 [※]	7	X2 [※]	7	X2 [※]	7
/RESET	输出	复位信号	$\overline{\text{RESET}}$	8	$\overline{\text{RESET}}$	8	$\overline{\text{RESET}}$	8
FLMD0	输出	模式信号	FLMD0	3	FLMD0	3	FLMD0	3
FLMD1	输出	模式信号	FLMD1/SO10/P17	40	FLMD1/SO10/P17	40	FLMD1/SO10/P17	40
H/S	输入	握手信号	-	-	P64	49	-	不需要
V _{DD}	输入/输出	V _{DD} 电压产生	V _{DD}	4	V _{DD}	4	V _{DD}	4
			EV _{DD}	26	EV _{DD}	26	EV _{DD}	26
			AV _{REF}	1	AV _{REF}	1	AV _{REF}	1
GND	-	地	V _{SS}	5	V _{SS}	5	V _{SS}	5
			EV _{SS}	25	EV _{SS}	25	EV _{SS}	25
			AV _{SS}	2	AV _{SS}	2	AV _{SS}	2

注 当使用 flash 存储器编程器以外的时钟时，连接编程器的 CLK 到 X1，并且连接其反相信号到 X2。

使用适配器对 flash 存储器写入时的建议连接示例如下所示。

图 25-2. 3 线串行 I/O (CSI10) 模式下 Flash 存储器写入适配器连线示例

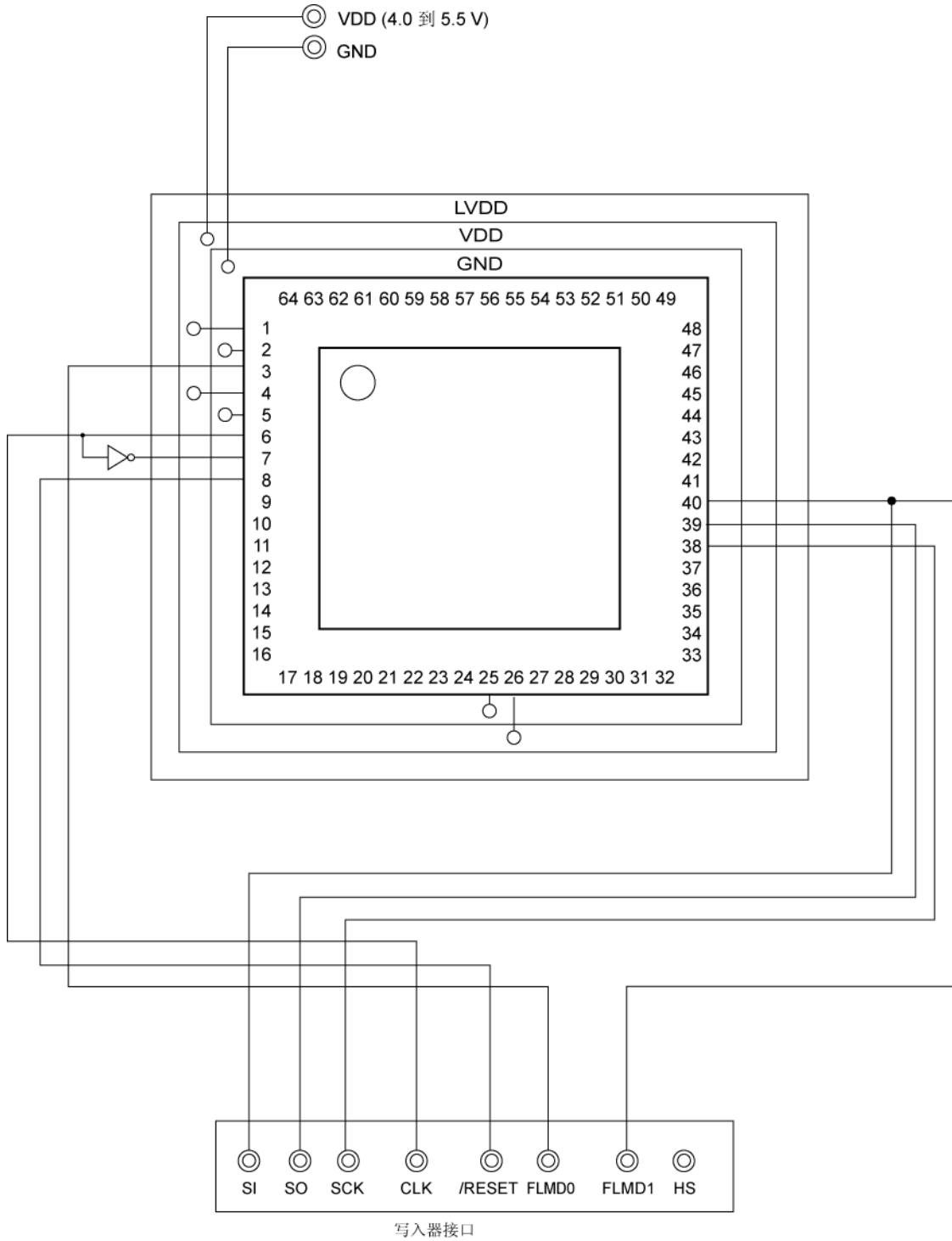


图 25-3. 3 线串行 I/O (CSI10 + HS) 模式下 Flash 存储器写入适配器连线示例

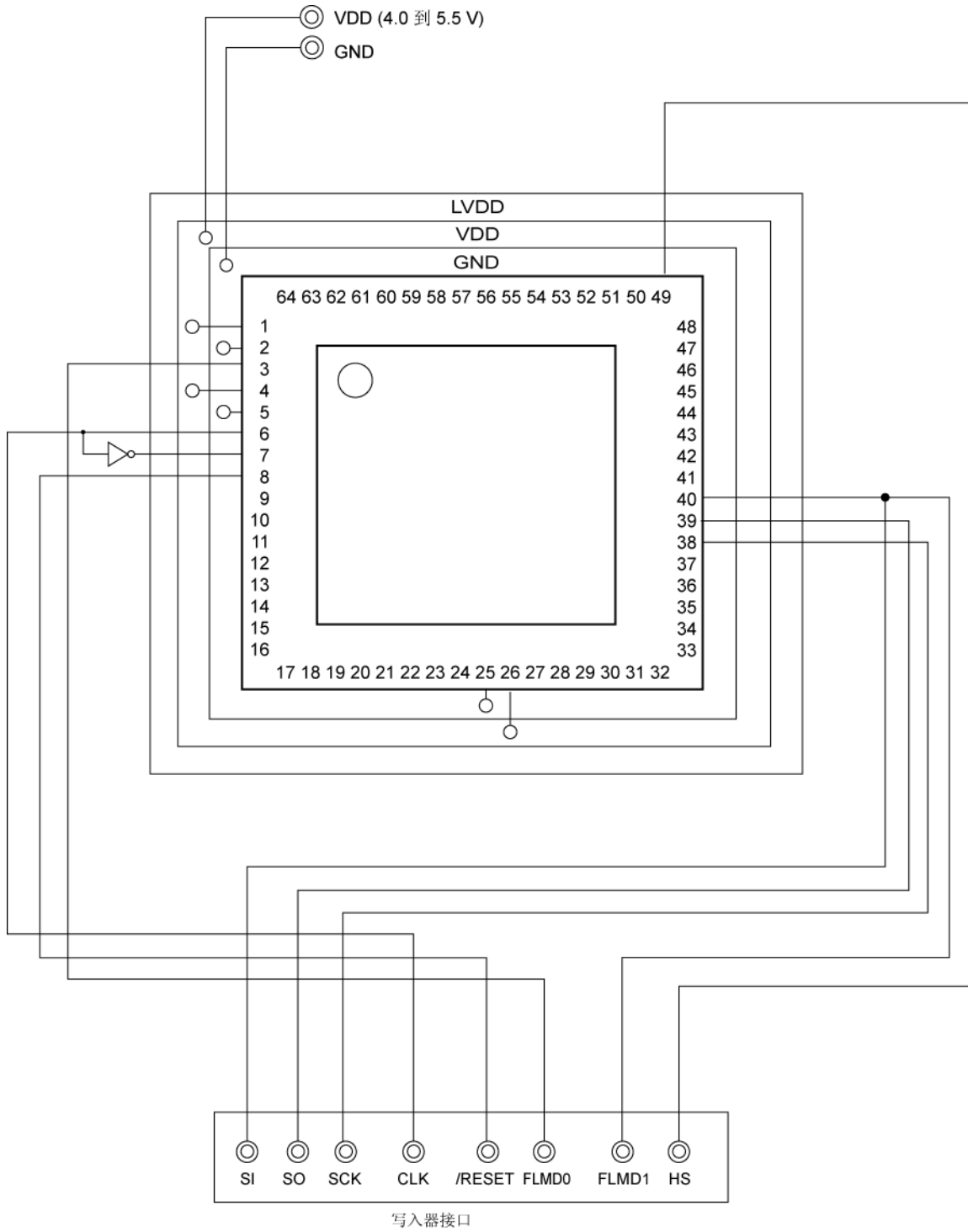
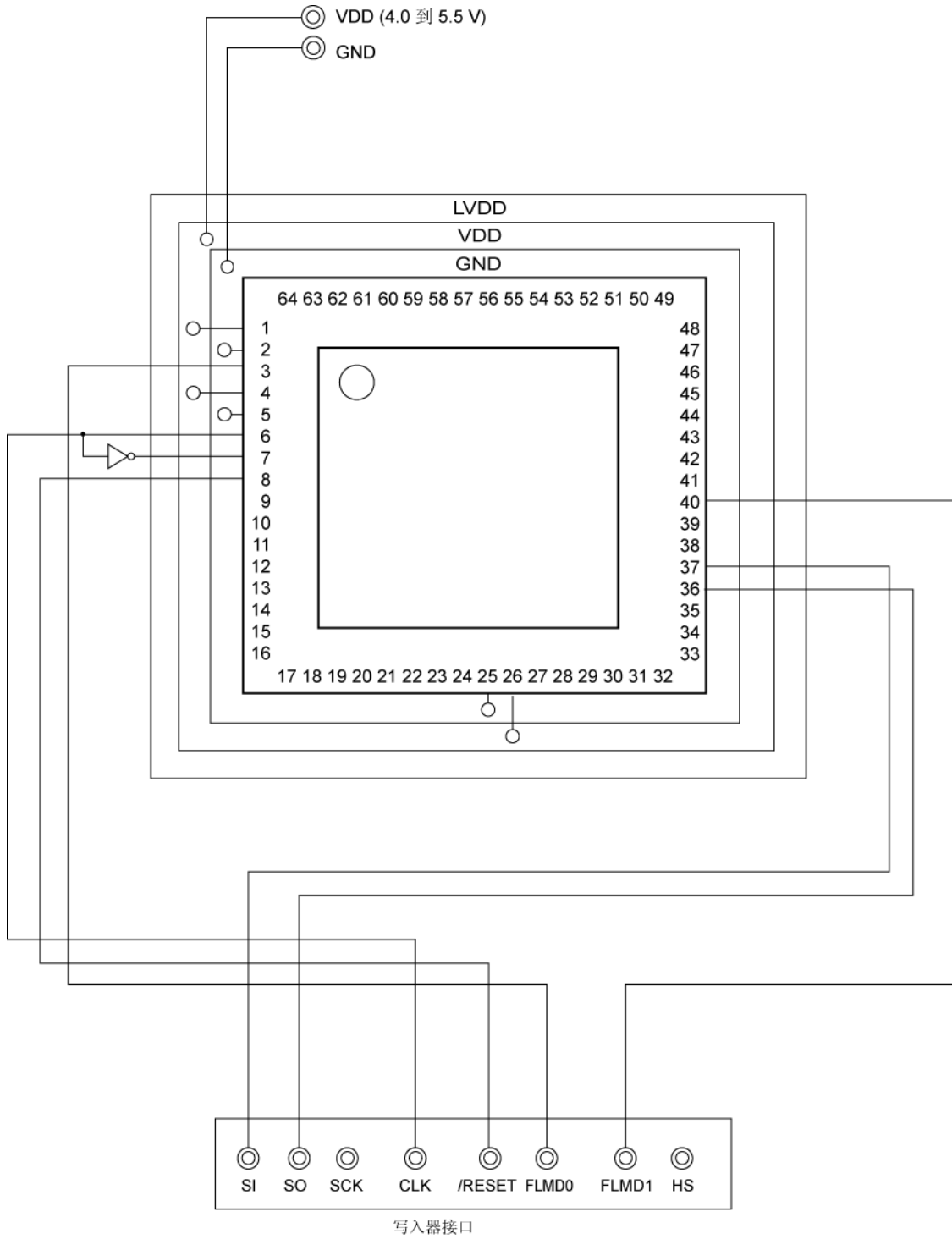


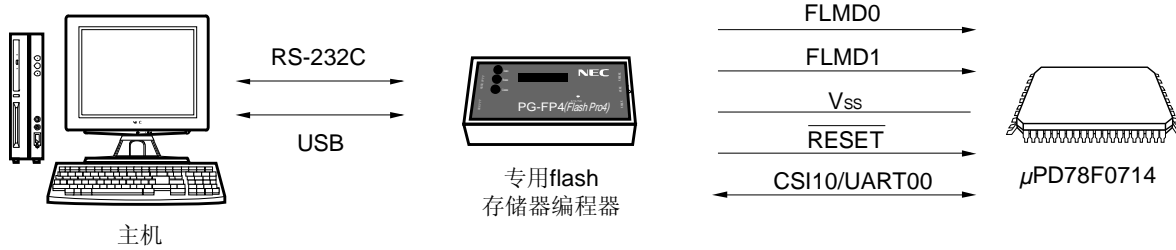
图 25-4. UART (UART0) 模式下 Flash 存储器写入适配器连线示例



25.3 编程环境

写入程序到 μ PD78F0714 的 flash 存储器所需要的环境如下所示。

图 25-5. 写入程序到 Flash 存储器的环境



控制专用 flash 存储器编程器的主机是必需的。

CSI10 或 UART00 被用于专用 flash 存储器编程器和 μ PD78F0714 之间的接口，例如写入和擦除。要在板外写入 flash 存储器，一个专用的程序适配器（FA 系列）是必需的。

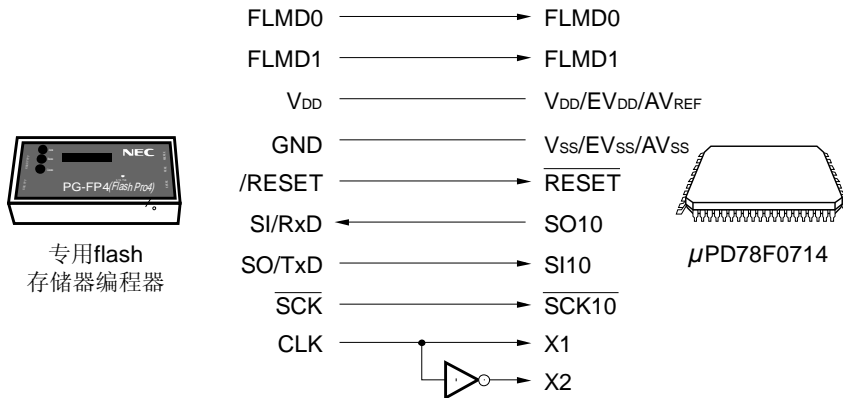
25.4 通信模式

专用 flash 存储器编程器和 μ PD78F0714 之间的通信由使用 μ PD78F0714 的 CSI10 或 UART00 来建立。

(1) CSI10

传输速率：200 kHz 到 2 MHz

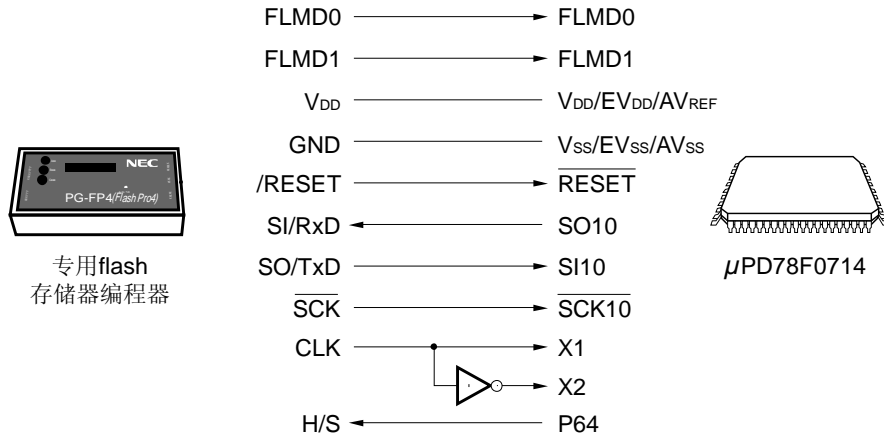
图 25-6. 与专用 Flash 存储器编程器的通信（CSI10）



(2) 支持握手的 CSI 通信模式

传输速率：200 kHz 到 2 MHz

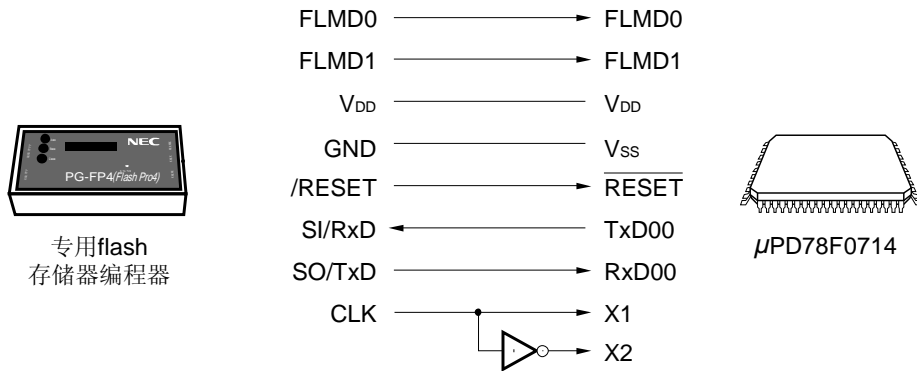
图 25-7. 与专用 Flash 存储器编程器的通信 (CSI10 + HS)



(3) UART00

传输速率：4800 到 76800 bps

图 25-8. 与专用 Flash 存储器编程器的通信 (UART00)



如果 Flashpro IV 用作专用 flash 存储器编程器，Flashpro IV 为 μ PD78F0714 产生以下信号。关于详细情况，参阅 Flashpro IV 手册。

表 25-2. 引脚连接

Flashpro IV			μ PD78F0714	连接	
信号名称	输入/输出	引脚功能	引脚名称	CSI10	UART00
FLMD0	输出	模式信号	FLMD0	◎	◎
FLMD1	输出	模式信号	FLMD1	○	○
V _{DD}	输入/输出	V _{DD} 电压产生	V _{DD} , EV _{DD} , AV _{REF}	◎	◎
GND	-	地	V _{SS} , EV _{SS} , AV _{SS}	◎	◎
CLK	输出	输出到 μ PD78F0714 的时钟	X1, X2 [#]	○	○
/RESET	输出	复位信号	$\overline{\text{RESET}}$	◎	◎
SI/RxD	输入	接收信号	SO10/TxD00	◎	◎
SO/TxD	输出	发送信号	SI10/RxD00	◎	◎
SCK	输出	发送时钟	$\overline{\text{SCK10}}$	◎	×
H/S	输入	握手信号	P64	△	×

注 当使用 flash 存储器编程器以外的时钟时，连接编程器的 CLK 到 X1，并且连接其反相信号到 X2。

备注 ◎: 确保连接该引脚。
 ○: 如果信号由目标板上产生，则该引脚无须被连接。
 ×: 该引脚无须被连接。
 △: 握手模式

25.5 板上引脚的处理

要在板上写入 flash 存储器，连接专用 flash 存储器编程器的连接器必须在目标系统中被提供。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 flash 存储器编程模式被设置时，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，引脚必须按照下面处理。

25.5.1 FLMD0 引脚

正常操作模式下，0V 输入到 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。下面表示 FLMD0 引脚连接的一个示例。

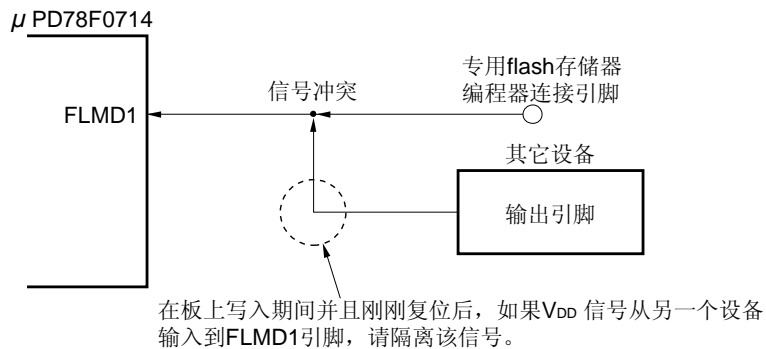
图 25-9. FLMD0 引脚连接示例



25.5.2 FLMD1 引脚

当 0 V 被输入到 FLMD0 引脚时，FLMD1 引脚无用。当 V_{DD} 提供给 FLMD0 引脚时，进入 flash 存储器编程模式，所以 FLMD1 必须输入与 V_{SS} 相同的电压。FLMD1 引脚连接示例如下所示。

图 25-10. FLMD1 引脚连接示例



25.5.3 串行接口引脚

用于串行接口的引脚列表如下。

表 25-3. 用于串行接口的引脚

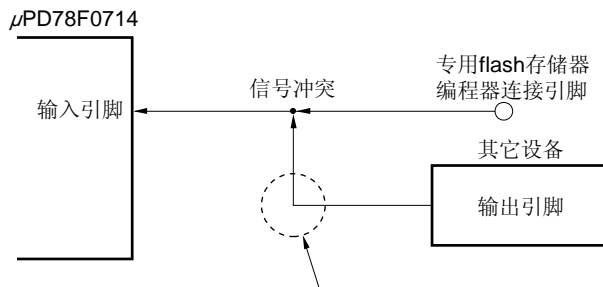
串行接口	使用的引脚
CSI10	SO10, SI10, SCK10
CSI10 + HS	SO10, SI10, SCK10, P64
UART00	TxD00, RxD00

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，小心信号不能冲突或者连接到的其它设备不能出故障。

(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），信号冲突会发生。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 25-11. 信号冲突（串行接口的输入引脚）

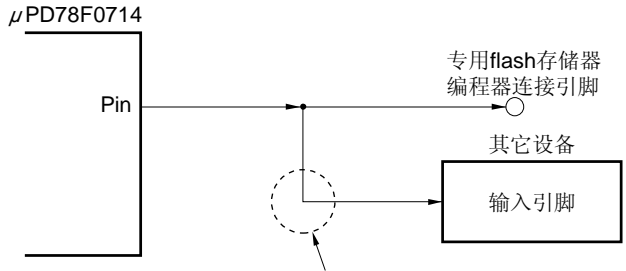


在 flash 存储器编程模式下，设备输出的信号与专用 flash 存储器编程器发送的信号冲突。因此，隔离其它设备的信号。

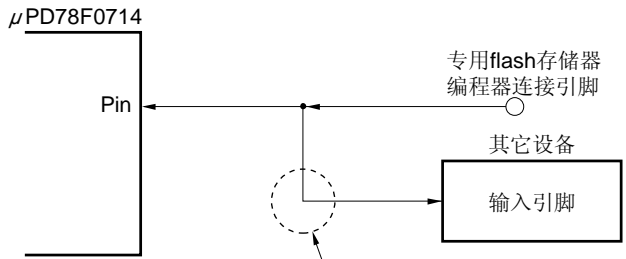
(2) 其它设备的故障

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，隔离于其它设备的连接。

图 25-12.其它设备的故障



如果 μ PD78F0714在flash存储器编程模式下输出的信号影响其它设备，请隔离其它设备的信号。



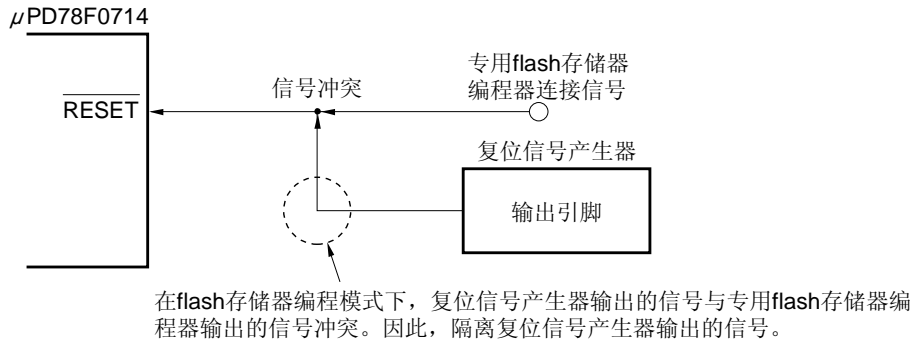
如果专用flash存储器编程器在flash存储器编程模式下输出的信号影响其它设备，请隔离其它设备的信号。

25.5.4 $\overline{\text{RESET}}$ 引脚

如果专用 flash 存储器编程器的复位信号被连接到板上复位信号产生器的 $\overline{\text{RESET}}$ 引脚，信号冲突将发生。要防止该冲突，隔离与复位信号产生器的连接。

当 flash 存储器编程模式被设置时，如果复位信号从用户系统被输入，flash 存储器将不会被正确编程。不要输入专用 flash 存储器编程器的复位信号以外的信号。

图 25-13. 信号冲突 ($\overline{\text{RESET}}$ 引脚)



25.5.5 端口引脚

当 flash 存储器编程模式被设置时，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。如果连接到端口的的外部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} 。

25.5.6 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。

然而，要从编程器输入操作时钟，连接编程器外的时钟到 X1，并且连接其反相信号到 X2。

25.5.7 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 V_{SS} 。

要使用板上电源，按照正常工作模式连接。

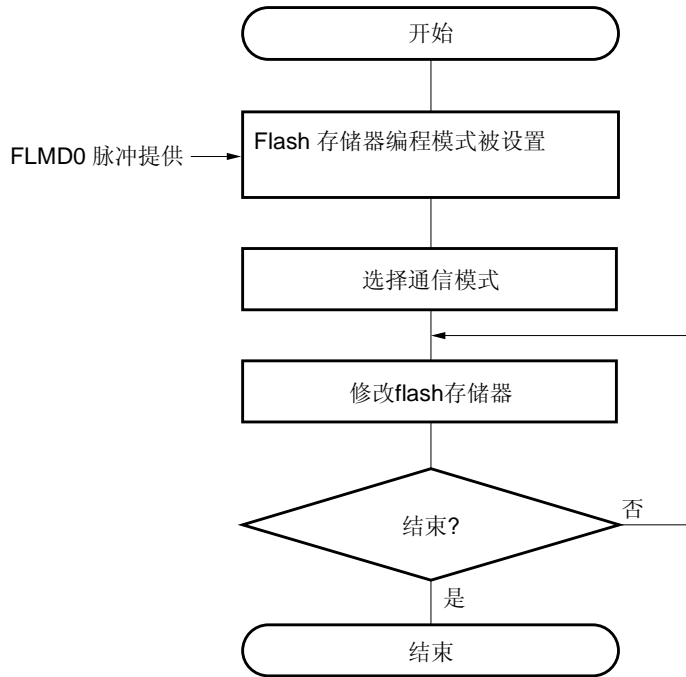
按照正常工作模式提供其它电源 (EV_{DD} , EV_{SS} , AV_{REF} 和 AV_{SS})。

25.6 编程方法

25.6.1 控制flash存储器

下面的图表示操作 flash 存储器的过程。

图 25-14. Flash 存储器操作过程



25.6.2 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 PD78F0714 为 flash 存储器编程模式。要设置该模式，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 25-15. Flash 存储器编程模式

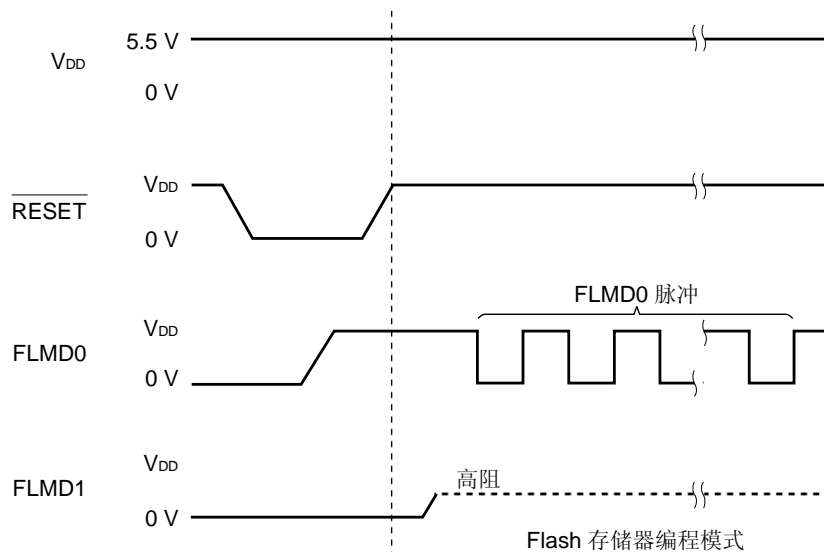


表 25-4. FLMD0 和 FLMD1 引脚的操作模式的关系

FLMD0	FLMD1	工作模式
0	×	正常工作模式
V _{DD}	0	Flash 存储器编程模式
V _{DD}	V _{DD}	禁止设置

25.6.3 选择通信模式

在 μ PD78F0714 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（最多 11 个脉冲）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 25-5. 通信模式

通信模式	标准设置 ^{#1}					使用的引脚	FLMD0 的脉冲数
	端口	速度	目标上	频率	多速率		
<R> UART (UART00)	UART-ch0	9600, 19200, 31250, 38400, 76800, 153600 ^{#2} bps ^{#3}	可选	5 M 到 20 MHz ^{#4}	1.0	TxD00, RxD00	0
3 线串行 I/O (CSI10)	SIO-ch0	200 k to 2 MHz ^{#4}				SO10, SI10, SCK10	8
可握手的 3 线串行 I/O (CSI10 + HS)	SIO-H/S	200 k to 2 MHz ^{#4}				SO10, SI10, SCK10, P64	11

注 1. Flashpro IV 上标志设置的选择项目。

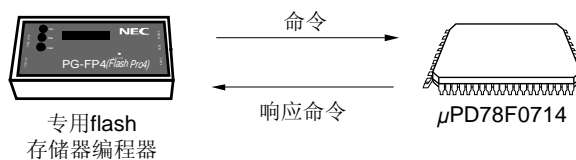
- <R>
2. 如果周边硬件时钟频率为 2.5 MHz 或更小，这不能被选择。
 3. 因为波特率误差以外因素，例如信号波形回转，也会影响 UART 通信，要彻底评估回转和波特率误差。
 4. 可能的设置范围因电压不同而不同。关于详细情况，参阅电气特性一章。

注意事项 当选取 UART00 时，接收时钟是在收到 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出的。

25.6.4 通信命令

μ PD78F0714 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 PD78F0714 的信号被叫做命令，并且从 μ PD78F0714 发送到专用 flash 存储器编程器的命令被叫做响应命令。

图 25-16. 通信命令



μ PD78F0714 的 flash 存储器控制命令在下面的表中被列出。所有的命令从编程器发出，并且 μ PD78F0714 执行对应各自命令的处理。

表 25-6. Flash 存储器控制命令

分类	命令名	功能
校验	批校验命令	将整个存储器的内容与输入数据进行比较
擦除	批擦除命令	擦除整个存储器的内容
空白检查	批空白检查命令	检查整个存储器的擦除状态
数据写入	高速写命令	通过指定写地址和要写入的字节数来写入数据，并执行校验检查。
	连线写命令	紧跟刚刚执行的高速写命令的地址后写入数据，并执行校验检查。
系统设置，控制	状态读命令	获取操作状态
	振荡频率设置命令	设置振荡频率
	擦除时间设置命令	为批擦除设置擦除时间
	写时间设置命令	为写数据设置写时间
	波特率设置命令	设置使用 UART 时的波特率
	Silicon 签名命令	读取 silicon 签名信息
	复位命令	从每个状态逃逸

μ PD78F0714 对专用 flash 存储器编程器发出的命令返回一个响应。 μ PD78F0714 发出的响应命令在下面被列出。

表 25-7. 响应命令

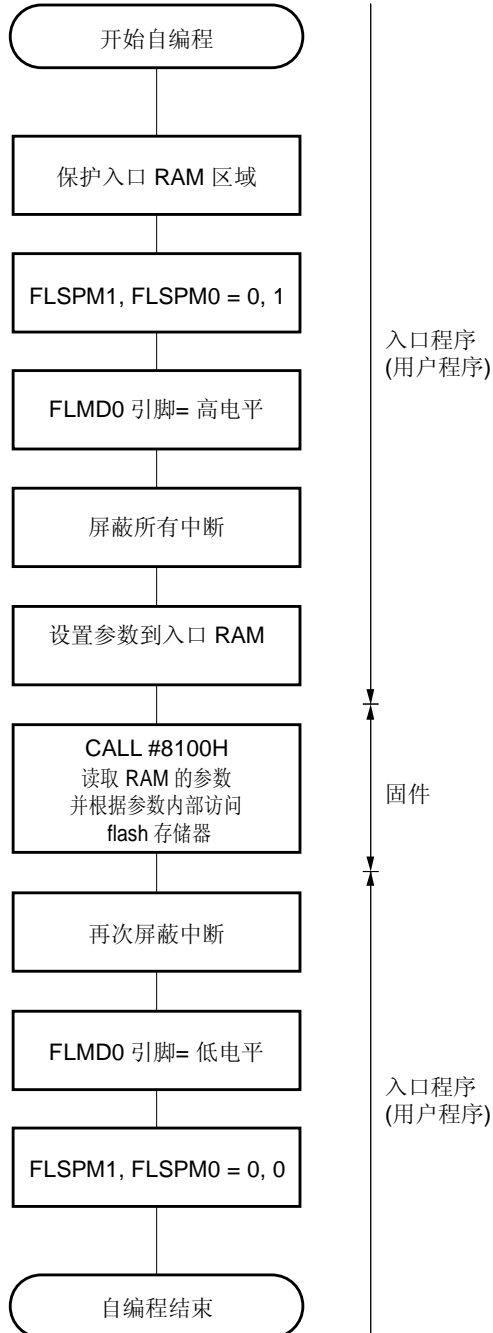
命令名	功能
ACK	响应命令/数据。
NAK	响应非法命令/数据。

25.7 通过自写的Flash存储器编程

μ PD78F0714 支持通过用户程序来重写 flash 存储器的自编程功能，所以程序可以现场更新。
编程模式通过 flash 编程模式控制寄存器（FLPMC）的第 0 和 1 位（FLSPM0 和 FLSPM1）来选择。
自编程的流程如下所示。

<R> 备注 关于自编程功能的详细情况，参阅将出版的单独文档（文档名： μ PD78F0711，78F0712，78F0714 Flash 存储器自编程用户手册（U18886E））。

图 25-17. 自编程流程



25.7.1 用于自编程功能的寄存器

以下三个寄存器用于自编程功能。

- Flash 编程模式控制寄存器 (FLPMC)
- Flash 保护命令寄存器 (PFCMD)
- Flash 状态寄存器 (PFS)

(1) Flash 编程模式控制寄存器 (FLPMC)

该寄存器用来允许或禁止 flash 存储器的写入或擦除并设置自编程期间的操作模式。

FLPMC 只能以指定的顺序 (参见 25.7.1 (2) Flash 保护命令寄存器) 写入, 所以应用系统不会因为由噪声或程序挂起引起的故障而无意停止。

FLPMC 可以由 1 位或 8 位存储器操作指令来设置。

RESET 输入设置该寄存器为 0xH^注。

注 根据操作模式而不同。

- 用户模式: 08H
- 板上模式: 0CH

图 25-18. Flash 编程模式控制寄存器 (FLPMC) 的格式

地址: FFC4H 复位后: 0xH^{注1} R/W^{注2}

符号	7	6	5	4	3	2	1	0
FLPMC	0	0	0	0	FWEDIS	FWEPR	FLSPM1	FLSPM0

FWEDIS	flash存储器写入/擦除的控制
0	写入/擦除允许 ^{注3}
1	写入/擦除禁止

FWEPR	FLMD0 引脚的状态
0	低电平
1	高电平 ^{注3}

FLSPM1 ^{注4}	FLSPM0 ^{注4}	自编程期间的操作模式选择
0	0	正常模式 flash 存储器的指令可以从所有地址获取。
0	1	自编程模式 A1 固件可以被调用 (CALL #8100H)。
1	1	自编程模式 A2 指令从固件 ROM 中获取。 该模式在固件中被设置, 并且不能由用户设置。
1	0	禁止设置

- 注
- 根据操作模式而不同。
 - 用户模式: 08H
 - 板上模式: 0CH
 - 第 2 位 (FWEPR) 只读。
 - 对于实际的写入/擦除, FLMPD0 引脚必须为高电平 (FWEPR = 1) 并且 FWEDIS = 0。

FWEDIS	FWEPR	Flash 存储器写入/擦除的允许或禁止
0	1	写入/擦除允许
除上面以外		写入/擦除禁止

- 用户 ROM (flash 存储器) 或固件 ROM 可以通过 FLSPM1 和 FLSPM0 来选择, 并且由模式引脚设置的应用系统操作模式或自编程模式可以被选择。

- 注意事项
- 确保保持 FWEDIS 为 0, 直到 flash 存储器的写入或擦除完成。
 - 在正常模式下, 确保 FWEDIS = 1。
 - 在执行跳转到内部 RAM 后修改 FLSPM1 和 FLSPM0。Flash 存储器的地址由 FLSPM1 = 0 时 CPU 的地址信号或 FLSPM1 = 1 时写入的固件设置值来指定。在板上模式下, FLSPM1 和 FLSPM0 的指定被忽略。

(2) Flash 保护命令寄存器 (PFCMD)

如果应用系统因为由噪声或程序挂起引起的故障而无意停止，写入 flash 编程模式控制寄存器 (FLPMC) 的操作可能对系统有严重影响。PFCMD 用来保护 FLPMC 被写入，所以应用系统不会被无意停止。

只有当写入操作按照以下指定顺序执行时，写 FLPMC 才被允许。

- <1> 写指定值到 PFCMD (PFCMD = A5H)
- <2> 写入要设置到 FLPMC 的值 (该步骤的写入无效)
- <3> 写入要设置到 FLPMC 的值反转值 (该步骤的写入无效)
- <4> 写入要设置到 FLPMC 的值 (该步骤的写入有效)

这将重写寄存器的值，所以寄存器不能被非法写入。

非法保存操作的发生可以通过 flash 状态寄存器 (PFS) 的第 0 位 (FPRERR) 来检查。

每次 FLPMC 的值被更改，A5H 必须被写入 PFCMD。

PFCMD 由 8 位存储器操作指令设置。

$\overline{\text{RESET}}$ 输入使该寄存器不确定。

图 25-19. Flash 保护命令寄存器 (PFCMD) 的格式

地址: FFC0H 复位后: 不确定W

符号	7	6	5	4	3	2	1	0
PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

(3) Flash 状态寄存器 (PFS)

如果数据没有按照正确顺序写入被保护 (写 flash 保护命令寄存器 (PFCMD)) 的 flash 编程模式控制寄存器 (FLPMC)，FLPMC 不会被写入并且一个保护错误发生。如果这种情况发生，PFS 的第 0 位 (FPRERR) 被设置为 1。

该位是一个积累的标志。检查 FPRERR 后，通过对其写入 0 来清除。

PFS 可以由 1 位或 8 位存储器操作指令来设置。

$\overline{\text{RESET}}$ 输入清除该寄存器为 00H。

图 25-20. Flash 状态寄存器 (PFS) 的格式

地址: FFC2H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PFS	0	0	0	0	0	0	0	FPRERR

FPRERR 标志的操作条件如下所示。

<设置条件>

- 当最近对周边寄存器执行的保存指令操作不是写入指定值 (A5H) 到 PFCMD 时, 如果 PFCMD 被写入
- 如果<1>后的第一个保存指令操作是用于 FLPMC 以外的周边寄存器
- 如果<2>后的第一个保存指令操作是用于 FLPMC 以外的周边寄存器
- 如果要设置到 FLPMC 的值的反转值以外的值通过<2>后的第一个保存指令被写入
- 如果<3>后的第一个保存指令操作是用于 FLPMC 以外的周边寄存器
- 如果要设置到 FLPMC 的值 (<2>中写入的值) 以外的值通过<3>后的第一个保存指令被写入

备注 上面三角括号中的数字对应于 (2) Flash 保护命令寄存器 (PFCMD) 中的数字。

<复位条件>

- 如果 0 被写入 FPRERR 标志
- 如果 $\overline{\text{RESET}}$ 被输入

<指定顺序说明示例>

要将 05H 写入 FLPMC

```
MOV  PFCMD, #0A5H    ;写 A5H 到 PFCMD.
MOV  FLPMC, #05H     ;写 05H 到 FLPMC.
MOV  FLPMC, #0FAH    ;写 0FAH (05H 的反转值) 到 FLPMC.
MOV  FLPMC, #05H     ;写 05H 到 FLPMC.
```

25.8 引导交换功能

μ PD78F0714 具有引导交换功能。

如果在引导区域被自编程重写时一个暂时的电源故障由于某种原因发生并且引导区域中的程序丢失，引导交换功能可以在上电、复位和启动重新应用后正确执行程序。

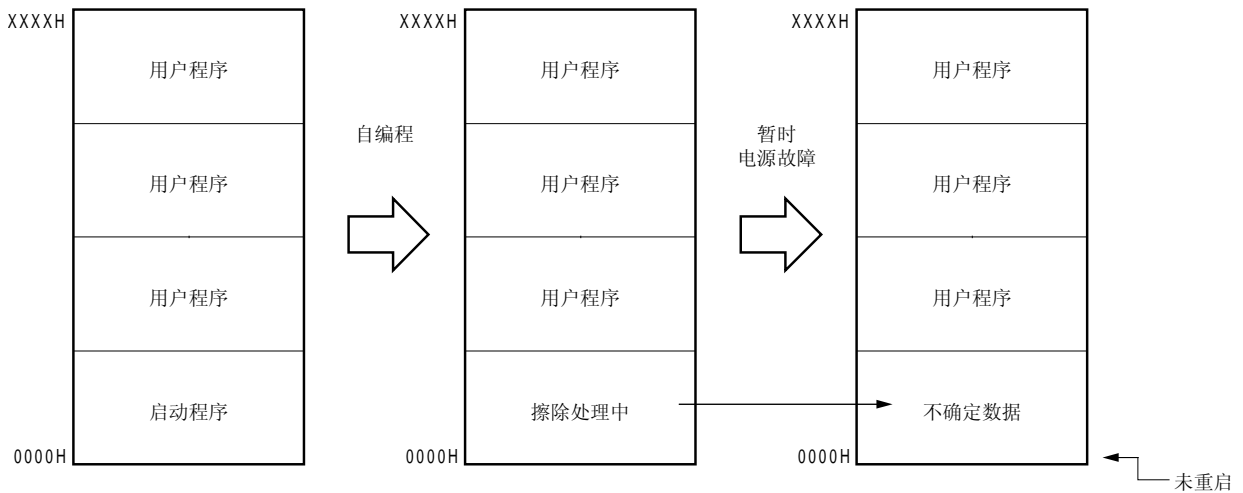
25.8.1 引导交换功能概述

在通过自编程擦除引导程序区域之前，向要交换的块中写入一个新的引导程序并设置引导标志^注。如果一个暂时的电源故障发生，当系统下次复位和启动时，地址被交换。因此，上面要被交换的区域用作引导区域，并且程序被正确执行。图 25-21 表示引导交换功能的图片。

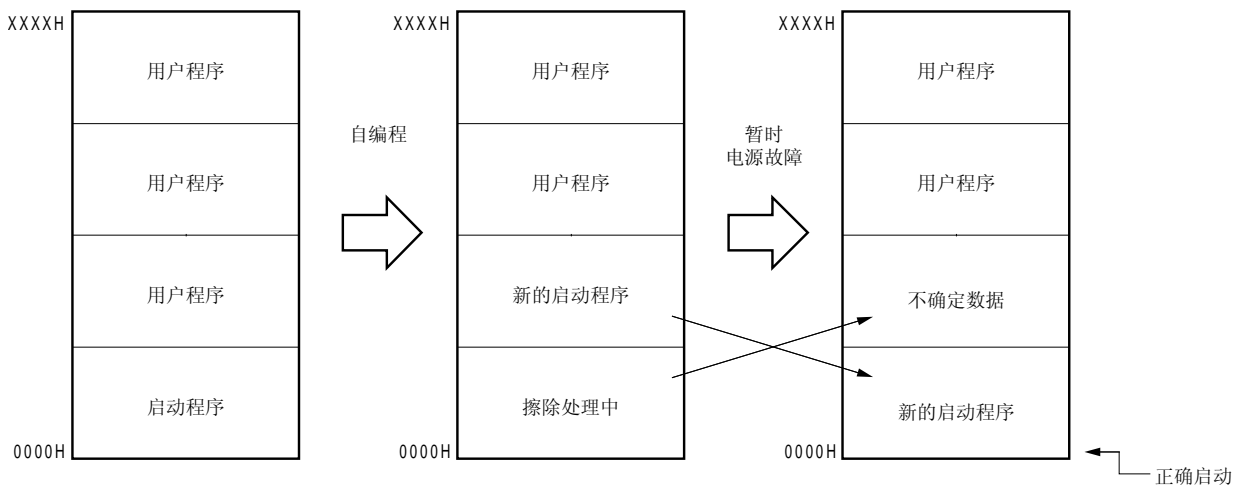
注 引导标志位于 μ PD78F0714 的 flash 存储器控制固件中。

图 25-21. 引导交换功能的图片

(1) 如果引导交换不被支持



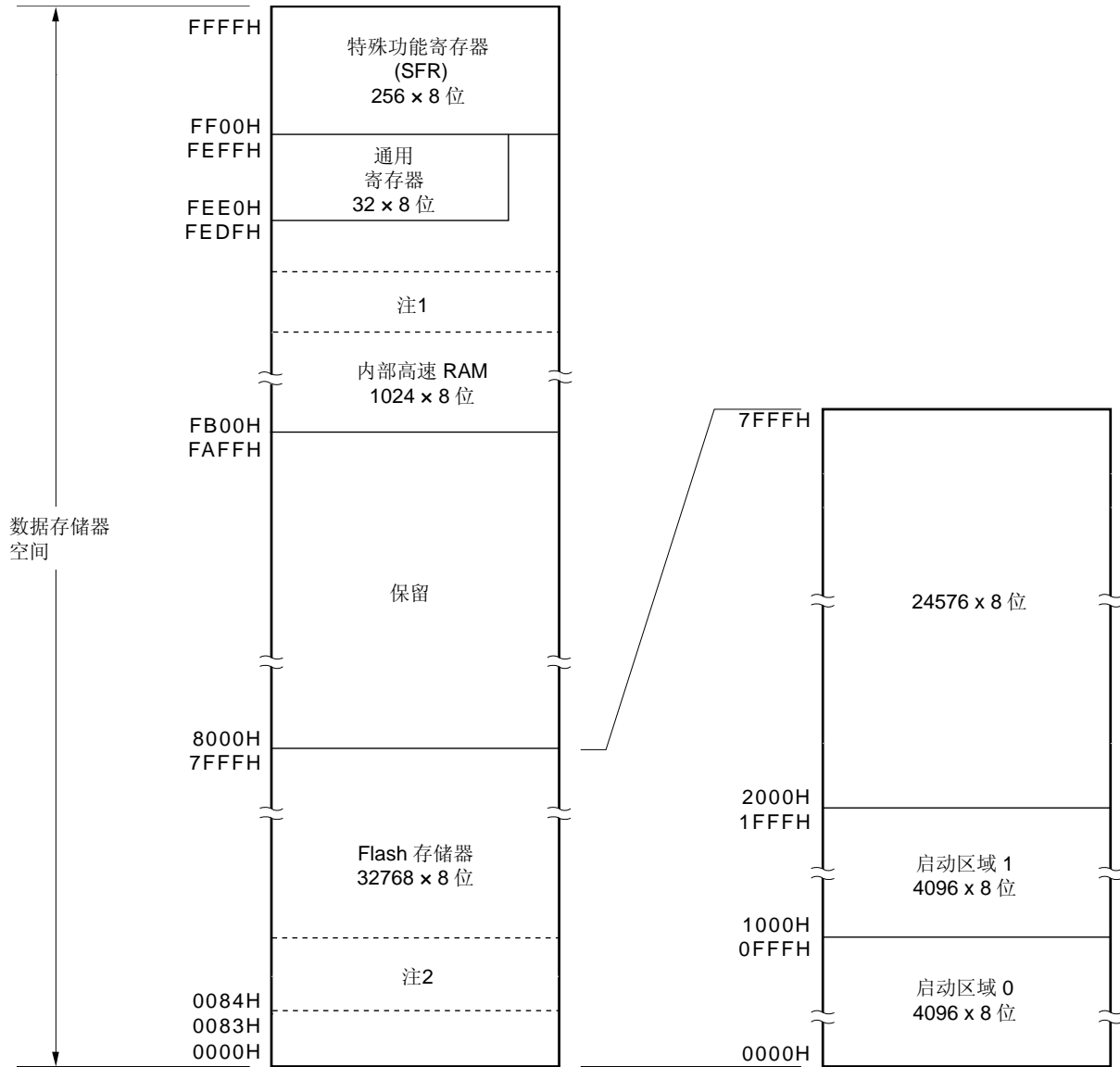
(2) 如果引导交换被支持



25.8.2 存储器映射和引导区域

图 25-22 表示存储器映射和引导区域。 μ PD78F0714 的引导程序区域以 4 KB 为单位。当引导交换被执行时，图中的引导区域 0 和引导区域 1 被交换。

图 25-22. 存储器映射和引导区域



- 注
1. 9 字节产品（预定计划）的该区域可以在片上调试期间被使用，因为它用作通信期间的用户程序的备份区域。
 2. 产品的该区域不能在片上调试期间被使用，因为它用作通信命令区域（256 字节到 1 KB）。

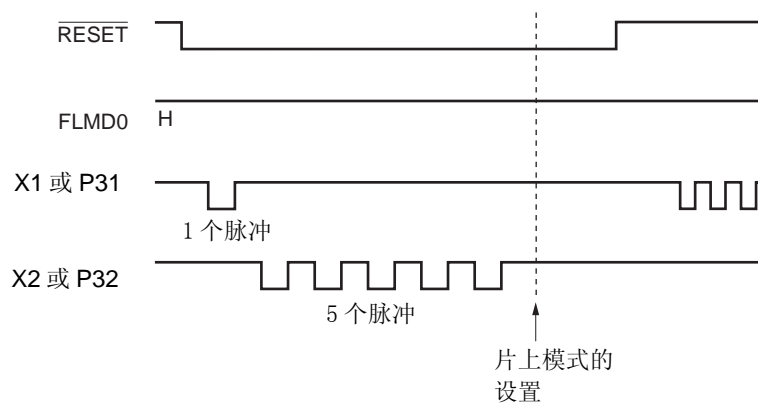
第二十六章 片上调试功能

<R> μ PD78F0714 使用 V_{DD} 、FLMD0、 $\overline{\text{RESET}}$ 、X1（或 P31）、X2（或 P32）和 V_{SS} 引脚通过片上调试仿真器（QB-78K0MINI 或 QB-MINI2）连接到主机以实现片上调试。是使用 X1 和 P31 还是 X2 和 P32 可以选择。

- 注意事项
1. 复位后确保下拉 P31 以防止故障。
 2. 当使用 P31 用作片上调试功能时，不建议将 P31 用作片上调试之外的功能。

<R> 备注 关于片上调试功能的详细情况，参阅 QB-78K0MINI 用户手册（U17029E）或 QB-MINI2 用户手册（U18371E）。

图 26-1. 设置片上调试功能的时序图



第二十七章 指令集

本章列表 μ PD78F0714 微控制器指令集中的指令。关于每种操作和操作码的详细情况，参阅单独的文档 **78K/0 系列指令用户手册 (U12326E)**。

27.1 在操作列表中使用的约定

27.1.1 操作数标识符和描述方法

操作数连同指令操作数标识符（关于详细情况，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**[]**是关键字并且本身被描述。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签。当使用标签时，确认描述**#**、**!**、**\$**和**[]**符号。

关于操作数寄存器标识符 **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）可以被用于描述。

表 27-1. 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，只能是偶地址） ^注
saddr	FE20H 到 FF1FH 立即数或标签
saddrp	FE20H 到 FF1EH 立即数或标签（只能是偶地址）
addr16	0000H 到 FFFFH 立即数或标签 （对于 16 位数据转移指令只能是偶地址）
addr11	0800H 到 0FFFH 立即数或标签
addr5	0040H 到 007EH 立即数或标签（只能是偶地址）
word	16 位立即数或标签
byte	8 位立即数或标签
bit	3 位立即数或标签
RBn	RB0 到 RB3

注 地址从 FFD0H 到 FFDFH 无法被这些操作数访问。

备注 关于特殊功能寄存器符号，见表 **3-3 特殊功能寄存器列表**。

27.1.2 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 bank 选择标志
IE:	中断请求允许标志
NMIS:	不可屏蔽中断服务标志
():	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 位立即数或标签
addr11:	立即数或标签
addr5:	立即数或标签 (仅限偶地址)
jdisp8:	有符号 8 位数据 (位移值)

27.1.3 操作栏标志的说明

(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:	上次保存的值被恢复

27.2 运算列表

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. 当内部高速 RAM 区域被访问时，或者对于一个没有数据访问的指令
 2. 当一个区域除去内部高速 RAM 区域被访问时
 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. 该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. 当内部高速 RAM 区域被访问时，或者对于一个没有数据访问的指令
 2. 当一个区域除去内部高速 RAM 区域被访问时
 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. 该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. 当内部高速 RAM 区域被访问时，或者对于一个没有数据访问的指令
 2. 当一个区域除去内部高速 RAM 区域被访问时
 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. 该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, !addr16	3	8	9	$A \leftarrow A \vee (\text{addr16})$	x
		A, [HL]	1	4	5	$A \leftarrow A \vee (\text{HL})$	x
		A, [HL + byte]	2	8	9	$A \leftarrow A \vee (\text{HL} + \text{byte})$	x
		A, [HL + B]	2	8	9	$A \leftarrow A \vee (\text{HL} + B)$	x
	A, [HL + C]	2	8	9	$A \leftarrow A \vee (\text{HL} + C)$	x	
	XOR	A, #byte	2	4	–	$A \leftarrow A \nabla \text{byte}$	x
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \nabla \text{byte}$	x
		A, r ^{注 3}	2	4	–	$A \leftarrow A \nabla r$	x
		r, A	2	4	–	$r \leftarrow r \nabla A$	x
		A, saddr	2	4	5	$A \leftarrow A \nabla (\text{saddr})$	x
		A, !addr16	3	8	9	$A \leftarrow A \nabla (\text{addr16})$	x
		A, [HL]	1	4	5	$A \leftarrow A \nabla (\text{HL})$	x
		A, [HL + byte]	2	8	9	$A \leftarrow A \nabla (\text{HL} + \text{byte})$	x
		A, [HL + B]	2	8	9	$A \leftarrow A \nabla (\text{HL} + B)$	x
	A, [HL + C]	2	8	9	$A \leftarrow A \nabla (\text{HL} + C)$	x	
	CMP	A, #byte	2	4	–	$A - \text{byte}$	x x x
		saddr, #byte	3	6	8	$(\text{saddr}) - \text{byte}$	x x x
		A, r ^{注 3}	2	4	–	$A - r$	x x x
		r, A	2	4	–	$r - A$	x x x
		A, saddr	2	4	5	$A - (\text{saddr})$	x x x
		A, !addr16	3	8	9	$A - (\text{addr16})$	x x x
		A, [HL]	1	4	5	$A - (\text{HL})$	x x x
A, [HL + byte]		2	8	9	$A - (\text{HL} + \text{byte})$	x x x	
A, [HL + B]		2	8	9	$A - (\text{HL} + B)$	x x x	
A, [HL + C]	2	8	9	$A - (\text{HL} + C)$	x x x		

- 注
1. 当内部高速 RAM 区域被访问时，或者对于一个没有数据访问的指令
 2. 当一个区域除去内部高速 RAM 区域被访问时
 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. 该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			
递增/ 递减	INC	r	1	2	-	r ← r + 1	x	x	
		saddr	2	4	6	(saddr) ← (saddr) + 1	x	x	
	DEC	r	1	2	-	r ← r - 1	x	x	
		saddr	2	4	6	(saddr) ← (saddr) - 1	x	x	
	INCW	rp	1	4	-	rp ← rp + 1			
	DECW	rp	1	4	-	rp ← rp - 1			
旋转	ROR	A, 1	1	2	-	(CY, A7 ← A0, A _{m-1} ← A _m) × 1 次			x
	ROL	A, 1	1	2	-	(CY, A0 ← A7, A _{m+1} ← A _m) × 1 次			x
	RORC	A, 1	1	2	-	(CY ← A0, A7 ← CY, A _{m-1} ← A _m) × 1 次			x
	ROLC	A, 1	1	2	-	(CY ← A7, A0 ← CY, A _{m+1} ← A _m) × 1 次			x
	ROR4	[HL]	2	10	12	A ₃₋₀ ← (HL) ₃₋₀ , (HL) ₇₋₄ ← A ₃₋₀ , (HL) ₃₋₀ ← (HL) ₇₋₄			
	ROL4	[HL]	2	10	12	A ₃₋₀ ← (HL) ₇₋₄ , (HL) ₃₋₀ ← A ₃₋₀ , (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. 当内部高速 RAM 区域被访问时, 或者对于一个没有数据访问的指令
2. 当一个区域除去内部高速 RAM 区域被访问时

- 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. 该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	$CY \leftarrow CY \wedge (\text{saddr.bit})$			x
		CY, sfr.bit	3	-	7	$CY \leftarrow CY \wedge \text{sfr.bit}$			x
		CY, A.bit	2	4	-	$CY \leftarrow CY \wedge A.\text{bit}$			x
		CY, PSW.bit	3	-	7	$CY \leftarrow CY \wedge \text{PSW.bit}$			x
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \wedge (\text{HL}).\text{bit}$			x
	OR1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \vee (\text{saddr.bit})$			x
		CY, sfr.bit	3	-	7	$CY \leftarrow CY \vee \text{sfr.bit}$			x
		CY, A.bit	2	4	-	$CY \leftarrow CY \vee A.\text{bit}$			x
		CY, PSW.bit	3	-	7	$CY \leftarrow CY \vee \text{PSW.bit}$			x
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \vee (\text{HL}).\text{bit}$			x
	XOR1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \oplus (\text{saddr.bit})$			x
		CY, sfr.bit	3	-	7	$CY \leftarrow CY \oplus \text{sfr.bit}$			x
		CY, A.bit	2	4	-	$CY \leftarrow CY \oplus A.\text{bit}$			x
		CY, PSW.bit	3	-	7	$CY \leftarrow CY \oplus \text{PSW.bit}$			x
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \oplus (\text{HL}).\text{bit}$			x
	SET1	saddr.bit	2	4	6	$(\text{saddr.bit}) \leftarrow 1$			
		sfr.bit	3	-	8	$\text{sfr.bit} \leftarrow 1$			
		A.bit	2	4	-	$A.\text{bit} \leftarrow 1$			
		PSW.bit	2	-	6	$\text{PSW.bit} \leftarrow 1$	x	x	x
		[HL].bit	2	6	8	$(\text{HL}).\text{bit} \leftarrow 1$			
	CLR1	saddr.bit	2	4	6	$(\text{saddr.bit}) \leftarrow 0$			
		sfr.bit	3	-	8	$\text{sfr.bit} \leftarrow 0$			
		A.bit	2	4	-	$A.\text{bit} \leftarrow 0$			
		PSW.bit	2	-	6	$\text{PSW.bit} \leftarrow 0$	x	x	x
		[HL].bit	2	6	8	$(\text{HL}).\text{bit} \leftarrow 0$			
	SET1	CY	1	2	-	$CY \leftarrow 1$			1
	CLR1	CY	1	2	-	$CY \leftarrow 0$			0
	NOT1	CY	1	2	-	$CY \leftarrow \overline{CY}$			x

- 注
1. 当内部高速 RAM 区域被访问时，或者对于一个没有数据访问的指令
 2. 当一个区域除去内部高速 RAM 区域被访问时

- 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. 该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
<R>	CALL	!addr16	3	7	-	(SP - 1) ← (PC + 3) _H , (SP - 2) ← (PC + 3) _L , PC ← addr16, SP ← SP - 2			
	CALLF	!addr11	2	5	-	(SP - 1) ← (PC + 2) _H , (SP - 2) ← (PC + 2) _L , PC ₁₅₋₁₁ ← 00001, PC ₁₀₋₀ ← addr11, SP ← SP - 2			
	CALLT	[addr5]	1	6	-	(SP - 1) ← (PC + 1) _H , (SP - 2) ← (PC + 1) _L , PC _H ← (addr5 + 1), PC _L ← (addr5), SP ← SP - 2			
	BRK		1	6	-	(SP - 1) ← PSW, (SP - 2) ← (PC + 1) _H , (SP - 3) ← (PC + 1) _L , PC _H ← (003FH), PC _L ← (003EH), SP ← SP - 3, IE ← 0			
	RET		1	6	-	PC _H ← (SP + 1), PC _L ← (SP), SP ← SP + 2			
	RETI		1	6	-	PC _H ← (SP + 1), PC _L ← (SP), PSW ← (SP + 2), SP ← SP + 3	R	R	R
	RETB		1	6	-	PC _H ← (SP + 1), PC _L ← (SP), PSW ← (SP + 2), SP ← SP + 3	R	R	R
堆栈操作	PUSH	PSW	1	2	-	(SP - 1) ← PSW, SP ← SP - 1			
		rp	1	4	-	(SP - 1) ← rp _H , (SP - 2) ← rp _L , SP ← SP - 2			
	POP	PSW	1	2	-	PSW ← (SP), SP ← SP + 1	R	R	R
		rp	1	4	-	rp _H ← (SP + 1), rp _L ← (SP), SP ← SP + 2			
	MOVW	SP, #word	4	-	10	SP ← word			
		SP, AX	2	-	8	SP ← AX			
AX, SP		2	-	8	AX ← SP				
无条件跳转	BR	!addr16	3	6	-	PC ← addr16			
		\$addr16	2	6	-	PC ← PC + 2 + jdisp8			
		AX	2	8	-	PC _H ← A, PC _L ← X			
条件跳转	BC	\$addr16	2	6	-	PC ← PC + 2 + jdisp8 如果 CY = 1			
	BNC	\$addr16	2	6	-	PC ← PC + 2 + jdisp8 如果 CY = 0			
	BZ	\$addr16	2	6	-	PC ← PC + 2 + jdisp8 如果 Z = 1			
	BNZ	\$addr16	2	6	-	PC ← PC + 2 + jdisp8 如果 Z = 0			

- 注
1. 当内部高速 RAM 区域被访问时, 或者对于一个没有数据访问的指令
 2. 当一个区域除去内部高速 RAM 区域被访问时

- 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. 该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.bit) = 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 (允许中断)	
	DI		2	–	6	IE ← 0 (禁止中断)	
	HALT		2	6	–	设置 HALT 模式	
	STOP		2	6	–	设置 STOP 模式	

- 注**
1. 当内部高速 RAM 区域被访问时，或者对于一个没有数据访问的指令
 2. 当一个区域除去内部高速 RAM 区域被访问时

- 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. 该时钟周期提供给内部 ROM 程序。

27.3 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 时

(3) 位操作指令

MOV1、AND1、OR1、XOR1、SET1、CLR1、NOT1、BT、BF、BTCLR

第二操作数 第一操作数	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	\$addr16	无
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) 调用指令/跳转指令

CALL、CALLF、CALLT、BR、BC、BNC、BZ、BNZ、BT、BF、BTCLR、DBNZ

第二操作数 第一操作数	AX	!addr16	!addr11	[addr5]	\$addr16
基本指令	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
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

第二十八章 电气特性

<R> 注意事项 μ PD78F0714 具有片上调试功能。片上调试功能被使用后不要大量使用该产品，因为使用片上调试功能后稳定性得不到保证，考虑到 flash 存储器重写次数的问题。日电电子不接受关于这项产品的投诉。

最大绝对额定值 ($T_A = 25^\circ\text{C}$) (1 / 2)

参数	符号	条件	额定值	单位	
电源电压	V_{DD}		-0.3 到 +6.5	V	
	EV_{DD}		-0.3 到 +6.5	V	
	V_{SS}		-0.3 到 +0.3	V	
	EV_{SS}		-0.3 到 +0.3	V	
	AV_{REF}		-0.3 到 $V_{DD} + 0.3$ ^注	V	
	AV_{SS}		-0.3 到 +0.3	V	
输入电压	V_i	P00 到 P03, P10 到 P17, P20 到 P27, P30 到 P33, P40 到 P47, P50 到 P57, P64 到 P67, P70 到 P73, X1, X2, RESET	-0.3 到 $V_{DD} + 0.3$ ^注	V	
输出电压	V_o		-0.3 到 $V_{DD} + 0.3$ ^注	V	
模拟输入电压	V_{AN}		$AV_{SS} - 0.3$ 到 $AV_{REF} + 0.3$ ^注 和 -0.3 到 $V_{DD} + 0.3$ ^注	V	
输出电流, 高	I_{OH}	每个引脚	-10	mA	
		所有引脚总和 -60 mA	P00 到 P03, P30 到 P33, P50 到 P57	-30	mA
		P10 到 P17, P40 到 P47, P64 到 P67, P70 到 P73, TW0TO0 到 TW0TO5	-30	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°C) (2 / 2)

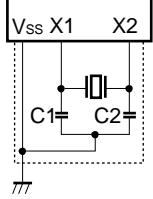
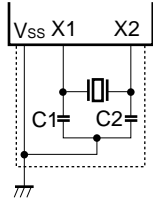
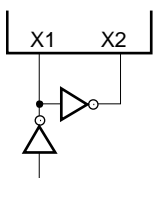
参数	符号	条件	额定值	单位	
输出电流, 低	I _{OL}	每个引脚	P00 到 P03, P10 到 P17, P30 到 P33, P40 到 P47, P64 到 P67, P70 到 P73	20	mA
			P50 到 P57, TW0TO0 到 TW0TO5	30	mA
		所有引脚总和 280 mA	P10 到 P17, P30 到 P33	30	mA
			P10 到 P17, P40 到 P47, P64 到 P67, P70 到 P73	50	mA
			TW0TO0 到 TW0TO5	100	mA
工作环境温度	T _A	在正常操作模式中	-40 到 +85	°C	
		在 flash 存储器编程模式	-10 到 +85		
存储温度	T _{stg}		-40 到 +125	°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

($T_A = -40$ 到 $+85^\circ\text{C}$, $4.0\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

谐振器	推荐电路	参数	条件	最小值	典型值	最大值	单位
陶瓷谐振器		振荡器频率 (f_{XP}) ^注	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	5.0		20	MHz
晶体谐振器		振荡器频率 (f_{XP}) ^注	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	5.0		20	MHz
外部时钟		X1 输入频率 (f_{XP}) ^注	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	5.0		20	MHz
		X1 输入高 / 低电平宽度 (t_{XPH} , t_{XPL})	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	24		100	ns

注 只显示振荡器特性。关于指令执行时间请参照 **AC 特性**。

注意事项 1. 当使用 X1 振荡器时，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接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 CPU 通过内部振荡时钟被启动，因此需要使用振荡稳定时间计数器状态寄存器 (OSTC) 来检查 X1 输入时钟振荡稳定时间。充分评估要使用的谐振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 对于谐振器的选择和振荡器常量，要求顾客要么自己对振荡进行评估要么通过谐振器制造商来进行评估。

内部振荡器特性

($T_A = -40$ 到 $+85^\circ\text{C}$, $3.3\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $3.3\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

谐振器	参数	条件	最小值	典型值	最大值	单位
内部振荡器	振荡频率 (f_R)		120	240	480	kHz

DC 特性 (1/2)

(TA = -40 到 +85°C, 4.0 V ≤ VDD = EVDD ≤ 5.5 V, 4.0 V ≤ AVREF ≤ VDD, VSS = EVSS = AVSS = 0 V)

参数	符号	条件	最小值	典型值	最大值	单位	
输出电流, 高	IOH	每个引脚	4.0 V ≤ VDD ≤ 5.5 V		-5	mA	
		P00 到 P03, P30 到 P33, P50 到 P57 的所有引脚	4.0 V ≤ VDD ≤ 5.5 V		-25	mA	
		P10 到 P17, P40 到 P47, P64 到 P67, P70 到 P73, TW0TO0 到 TW0TO5 的所有引脚	4.0 V ≤ VDD ≤ 5.5 V		-25	mA	
输出电流, 低	IOL	P00 到 P03, P10 到 P17, P30 到 P33, P40 到 P47, P64 到 P67, P70 到 P73 的每个引脚	4.0 V ≤ VDD ≤ 5.5 V		10	mA	
		P50 到 P57, TW0TO0 到 TW0TO5 的每个引脚	4.0 V ≤ VDD ≤ 5.5 V		15	mA	
		P10 到 P17, P30 到 P33 的所有引脚	4.0 V ≤ VDD ≤ 5.5 V		15	mA	
		P10 到 P17, P40 到 P47, P64 到 P67, P70 到 P73 的所有引脚	4.0 V ≤ VDD ≤ 5.5 V		25	mA	
		TW0TO0 到 TW0TO5 的所有引脚	4.0 V ≤ VDD ≤ 5.5 V		70	mA	
		P50 到 P57 的所有引脚	4.0 V ≤ VDD ≤ 5.5 V		70	mA	
输入电压, 高	VIH1	P14, P17, P30 到 P33, P40 到 P47, P64 到 P67, P70 到 P73 的所有引脚	4.0 V ≤ VDD ≤ 5.5 V	0.7VDD	VDD	V	
	VIH2	P00 到 P03, P10-P13, P15, P16, P50 到 P57, RESET	4.0 V ≤ VDD ≤ 5.5 V	0.8VDD	VDD	V	
	VIH3	P20 到 P27 ^注	4.0 V ≤ VDD ≤ 5.5 V	0.7AVREF	AVREF	V	
	VIH4	X1, X2	4.0 V ≤ VDD ≤ 5.5 V	VDD - 0.5	VDD	V	
输入电压, 低	VIL1	P14, P17, P30 到 P33, P40 到 P47, P64 到 P67, P70 到 P73 的所有引脚	4.0 V ≤ VDD ≤ 5.5 V	0	0.3VDD	V	
	VIL2	P00 到 P03, P10 到 P13, P15, P16, P50 到 P57, RESET	4.0 V ≤ VDD ≤ 5.5 V	0	0.2VDD	V	
	VIL3	P20 到 P27 ^注	4.0 V ≤ VDD ≤ 5.5 V	0	0.3AVREF	V	
	VIL4	X1, X2	4.0 V ≤ VDD ≤ 5.5 V	0	0.4	V	
<R>	输出电压, 高	VOH	P00 到 P03, P10 到 P17, P30 到 P33, P40 到 P47, P64 到 P67, P70 到 P73	4.0 V ≤ VDD ≤ 5.5 V, IOH = -5 mA	VDD - 1.0	V	
P50 到 P57, TW0TO0 到 TW0TO5			4.0 V ≤ VDD ≤ 5.5 V, IOH = -1 mA	VDD - 1.0	V		
IOH = -100 μA			4.0 V ≤ VDD ≤ 5.5 V	VDD - 0.5	V		
<R>	输出电压, 低	VOL	P50 到 P57, TW0TO0 到 TW0TO5	4.0 V ≤ VDD ≤ 5.5 V, IOL = 15 mA	0.4	2.0	V
<R>			P00 到 P03, P10 到 P17, P30 到 P33, P40 到 P47, P64 到 P67, P70 到 P73	4.0 V ≤ VDD ≤ 5.5 V, IOL = 10 mA		1.5	V
<R>			总共 IOL = 20 mA				
		IOL = 400 μA	4.0 V ≤ VDD ≤ 5.5 V		0.5	V	

注 当用作数字输入端口, 设置 AVREF = EVDD。

备注 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

DC 特性 (2 / 2)

(TA = -40 到 +85°C, 4.0 V ≤ VDD = EVDD ≤ 5.5 V, 4.0 V ≤ AVREF ≤ VDD, VSS = EVSS = AVSS = 0 V)

参数	符号	条件		最小值	典型值	最大值	单位	
输入泄漏电流, 高	I _{LIH1}	V _I = V _{DD}	P00 到 P03, P10 到 P17, P30 到 P33, P40 到 P47, P64 到 P67, P70 到 P73, RESET			3	μA	
		V _I = AV _{REF}	P20 到 P27			3	μA	
	I _{LIH2}	V _I = V _{DD}	X1, X2 ^{注1}			20	μA	
输入泄漏电流, 低	I _{LIL1}	V _I = 0 V	P00 到 P03, P10 到 P17, P20 到 P27, P30 到 P33, P40 到 P47, P50 到 P57, P64 到 P67, P70 到 P73, RESET			-3	μA	
	I _{LIL2}		X1, X2 ^{注1}			-20	μA	
输出泄漏电流, 高	I _{LOH}	V _O = V _{DD}				3	μA	
输出泄漏电流, 低	I _{LOL}	V _O = 0 V				-3	μA	
上拉电阻值	R _L	V _I = 0 V		10	30	100	kΩ	
FLMDO 电源电压	F _{lmd}	在正常操作模式中		0		0.2V _{DD}	V	
供电电流 ^{注2}	I _{DD1}	X1 晶体振荡器工作 ^{注3}	f _{XP} = 20 MHz V _{DD} = 5.0 V ±10% ^{注4}	当 A/D 转换器停止		18	36	mA
				当 A/D 转换器工作 ^{注5}		20	40	mA
	I _{DD2}	X1 晶体振荡器 HALT 模式	f _{XP} = 20 MHz V _{DD} = 5.0 V ±10%	当周边功能停止		3.5	7.0	mA
				当周边功能工作			15	mA
I _{DD3}	内部振荡器时钟操作模式 ^{注6}	V _{DD} = 5.0 V ±10%				3.0	6.0	mA
I _{DD4}	STOP 模式	V _{DD} = 5.0 V ±10%	内部振荡器 OFF			3.5	35.5	μA
			内部振荡器 ON			17.5	63.5	μA

<R>

- 注
1. 当 X1 的反向电平输入 X2 时
 2. 全部电流通过内部电源 (V_{DD}) 包括外部工作电流 (然而, 通过端口上拉电阻的电流不包括在内)。
 3. I_{DD1} 包括外部工作电流。
 4. 当 PCC = 00H。
 5. 包括通过 AV_{REF} 引脚的电流。
 6. 当 X1 振荡停止。

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。

AC 特性

(1) 基本操作

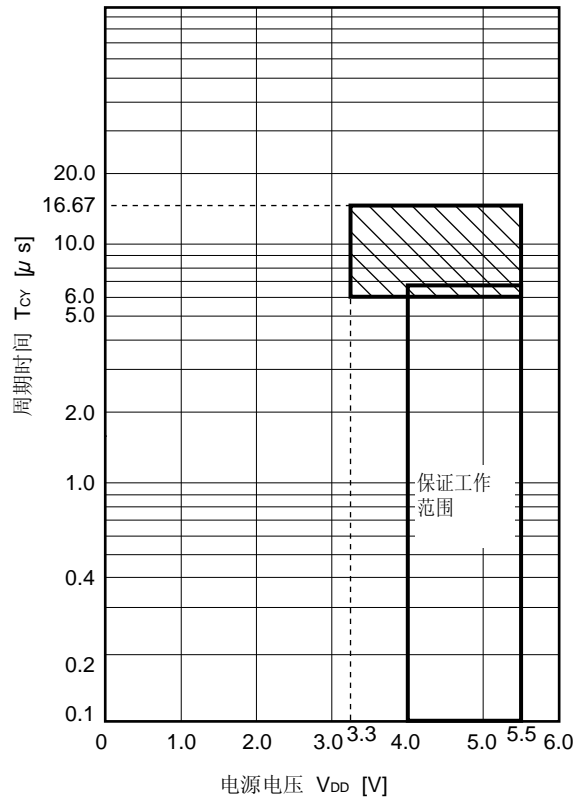
($T_A = -40$ 到 $+85^\circ\text{C}$, $4.0\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件		最小值	典型值	最大值	单位
<R> 指令周期 (最小指令执行时间)	T_{CY}	主系统时钟工作	X1 输入时钟	0.1		6.4	μs
			内部振荡时钟	$3.3\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	4.17	8.33	16.67
TI000, TI001 输入高电平宽度, 低电平宽度	t_{TIH0} , t_{TIL0}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		$2/f_{sam1} + 0.1$ ^{#1}			μs
TI50, TI51 输入频率	f_{TI5}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$				10	MHz
TI50, TI51 输入高电平宽度, 低电平宽度	t_{TIH5} , t_{TIL5}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		50			ns
TIT20IUD, TIT20CUD, TIT20CC0, TIT20CC1, TIT20CLR 输入高电平宽度, 低电平宽度	t_{WUDH} , t_{WUDL}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		$2/f_{sam2} + 0.1$ ^{#2}			μs
中断输入高电平宽度, 低电平宽度	t_{INTH} , t_{INTL}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		1			μs
<R> ADTRG 输入高电平宽度, 低电平宽度	t_{ADTRGH} , t_{ADTRGL}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		1			μs
RESET 低电平宽度	t_{RSL}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		10			μs

- 注
1. 可以使用预分频模式寄存器 00 (PRM00) 的第 0 位和第 1 位 (PRM000, PRM001) 来选择 $f_{sam} = f_{XP}/2$, $f_{XP}/4$, 或 $f_{XP}/256$ 。当将 TI000 有效沿选作计数时钟 $f_{sam1} = f_{XP}$ 时需注意。
 2. $f_{sam2} = f_{XP}/2^3$, $f_{XP}/2^2$, $f_{XP}/2$, 或 f_{XP} 的选择可能使用噪声消减时间选择寄存器 1 (NCR1) 的第 0 和 1 位 (NRC10, NRC11)。

<R>

T_{CY} vs. V_{DD} (主系统时钟操作)



备注 阴影部分表示的值仅适用于选择内部振荡器时钟。

(2) 串行接口

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $4.0\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

(a) UART 模式 (UART0, 专用波特率发生器输出)

参数	符号	条件	最小值	典型值	最大值	单位
传输速率					312.5	kbps

(b) 3 线串行 I/O 模式 (主模式, $\overline{\text{SCK10}}$...内部时钟输出)

参数	符号	条件	最小值	典型值	最大值	单位
$\overline{\text{SCK10}}$ 周期时间	t_{KCY1}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	200			ns
$\overline{\text{SCK10}}$ 高 / 低电平宽度	t_{KH1} , t_{KL1}		$t_{\text{KCY1}}/2 - 10$			ns
SI10 设置时间 (到 $\overline{\text{SCK10}}\uparrow$)	t_{SIK1}		30			ns
SI10 保持时间 (从 $\overline{\text{SCK10}}\uparrow$)	t_{KSI1}		30			ns
从 $\overline{\text{SCK10}}\downarrow$ 至 SO10 输出的延迟时间	t_{KSO1}	$C = 100\text{ pF}^{\#}$			30	ns

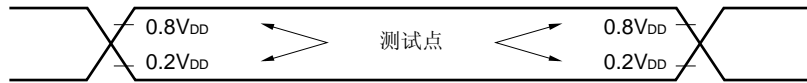
注 C 是 $\overline{\text{SCK10}}$ 和 SO10 输出线路的负载电容。

(c) 3 线串行 I/O 模式 (从模式, $\overline{\text{SCK10}}$...外部时钟输入)

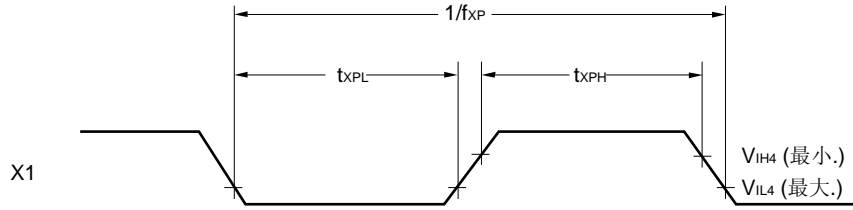
参数	符号	条件	最小值	典型值	最大值	单位
$\overline{\text{SCK10}}$ 周期时间	t_{KCY2}		400			ns
$\overline{\text{SCK10}}$ 高 / 低电平宽度	t_{KH2} , t_{KL2}		$t_{\text{KCY2}}/2$			ns
SI10 设置时间 (到 $\overline{\text{SCK10}}\uparrow$)	t_{SIK2}		80			ns
SI10 保持时间 (从 $\overline{\text{SCK10}}\uparrow$)	t_{KSI2}		50			ns
从 $\overline{\text{SCK10}}\downarrow$ 至 SO10 输出的延迟时间	t_{KSO2}	$C = 100\text{ pF}^{\#}$			120	ns

注 C 是 SO10 输出线路的负载电容。

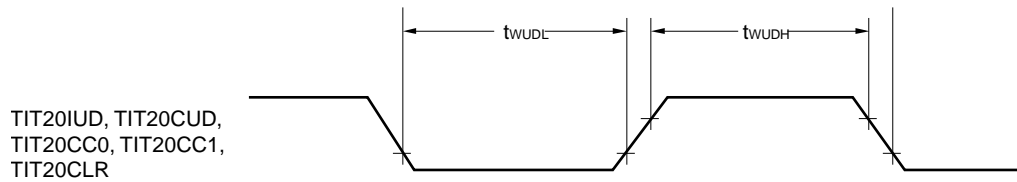
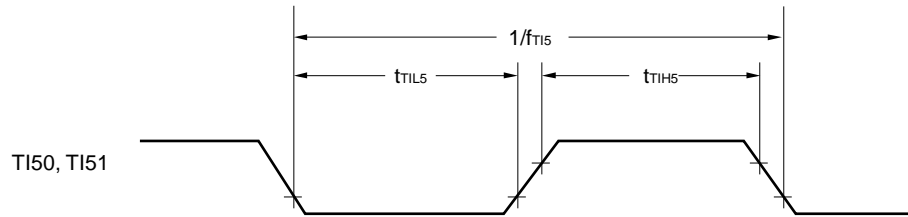
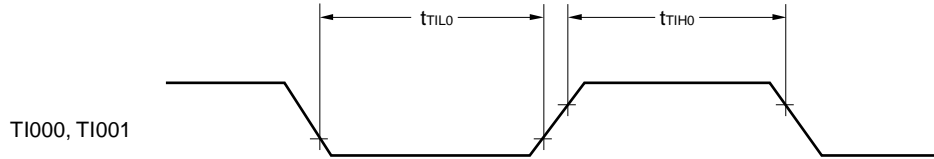
AC 时序测试点 (X1 输入除外)



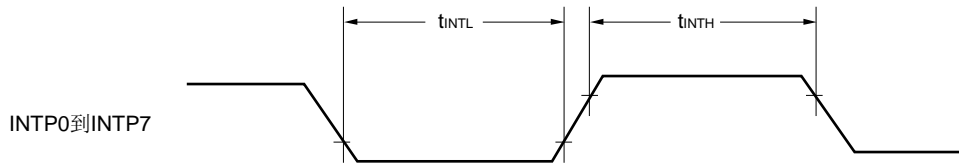
时钟时序



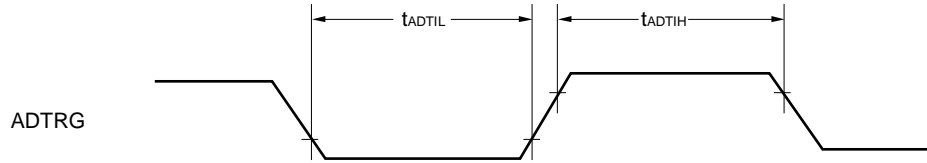
TI 的时序



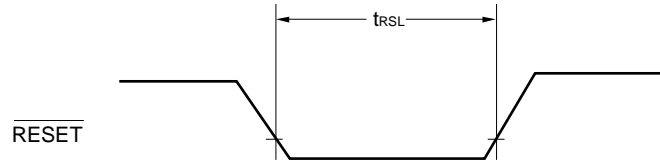
中断请求输入时序



<R> A/D 触发输入的时序

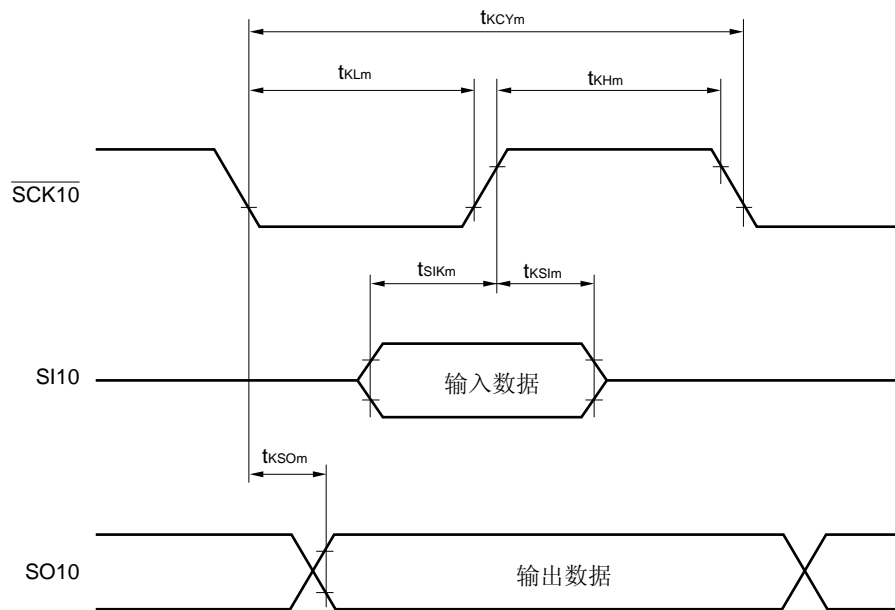


$\overline{\text{RESET}}$ 输入时序



串行传输时序

3 线串行 I/O 模式:



备注 m = 1, 2

A/D 转换器特性

($T_A = -40$ 到 $+85^\circ\text{C}$, $4.0\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
分辨率			10	10	10	位
总误差 ^{注 1, 2}		$4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$		± 0.2	± 0.4	%FSR
转换时间	t_{CONV}	$4.5\text{ V} \leq AV_{REF} \leq V_{DD}$	3.6		100	μs
		$4.0\text{ V} \leq AV_{REF} \leq V_{DD}$	4.8		100	μs
零幅度误差 ^{注 1, 2}		$4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$			± 0.4	%FSR
满幅度误差 ^{注 1, 2}		$4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$			± 0.4	%FSR
积分非线性误差 ^{注 1}		$4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$			± 2.5	LSB
微分非线性误差 ^{注 1}		$4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$			± 1.5	LSB
模拟输入电压	V_{IAN}		AV_{SS}		AV_{REF}	V

<R>

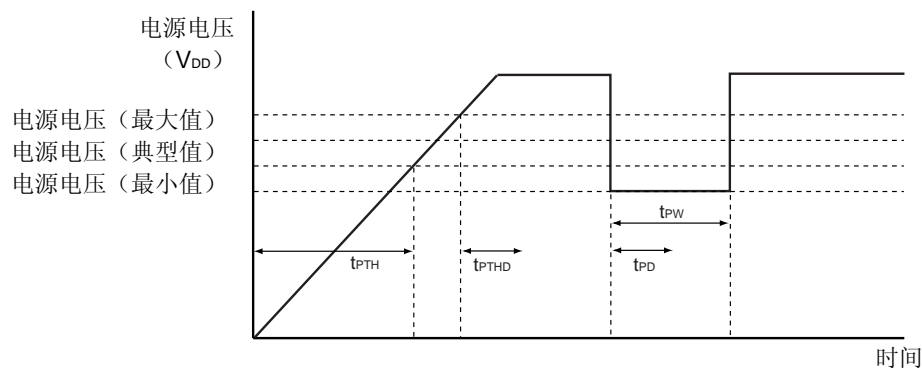
- 注 1. 不包括量化误差 ($\pm 1/2$ LSB)。
2. 该值按满幅度值的比率 (%FSR) 来表示。

POC 电路特性 ($T_A = -40$ 到 $+85^\circ\text{C}$)

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V_{POC}		3.3	3.5	3.7	V
电源上升时间	t_{PTH}	$V_{DD}: 0\text{ V} \rightarrow 3.3\text{ V}$	0.002			ms
响应延迟时间 1 ^注	t_{PTHd}	当电源上升, 达到检测电压 (最大值)			3.0	ms
响应延迟时间 2 ^注	t_{PD}	当 V_{DD} 下降			1.0	ms
最小脉冲宽度	t_{PW}		0.2			ms

注 从电压检测到复位释放要求的时间。

POC 电路时序

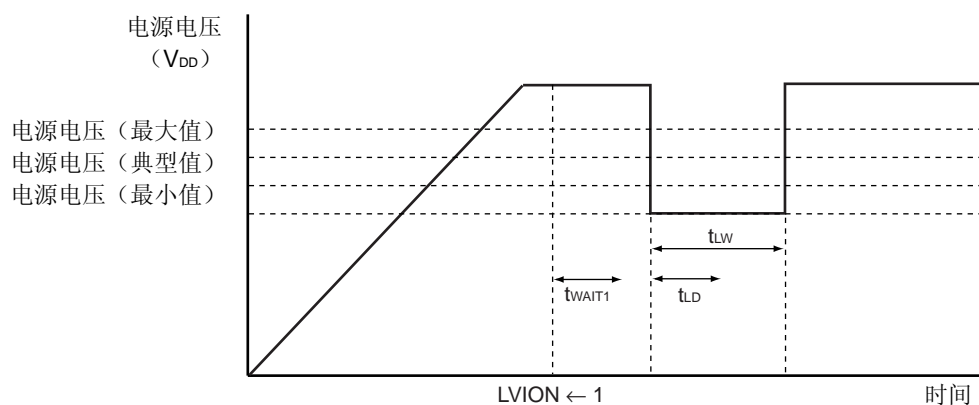


LVI 电路特性 ($T_A = -40$ 到 $+85^\circ\text{C}$)

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V_{LVI}		4.1	4.3	4.5	V
响应时间 ^{注1}	t_{LD}			0.2	2.0	ms
最小脉冲宽度	t_{LW}		0.2			ms
工作稳定等待时间 ^{注2}	t_{LWAIT1}			0.1	0.2	ms

- 注
1. 从电压检测到中断输出或内部复位输出的要求时间。
 2. 设置 LVION 为 1 到工作稳定要求的时间。

LVI 电路时序

数据寄存器 STOP 模式低电源电压数据保持特性 ($T_A = -40$ 到 $+85^\circ\text{C}$)

<R>

参数	符号	条件	最小值	典型值	最大值	单位
数据保持电源电压	V_{DDDR}		2.0		5.5	V
释放信号设置时间	t_{SREL}		0			μs

Flash 存储器编程特性

(TA = +10 到+85°C, 4.0 V ≤ VDD ≤ 5.5 V, 4.0 V ≤ AVREF ≤ VDD, VSS = 0 V)

<R>

(1) 基本特性

参数	符号	条件	最小值	典型值	最大值	单位
VDD 电源电流	IDD	f _x = 20 MHz VDD = 5.5 V			42	mA
分步擦除时间	T _{erass}			10		ms
擦除时间 ^{#1}	芯片单元	T _{eraca}		0.01	2.55	s
	部分单元	T _{erasa}		0.01	2.55	s
写入时间	T _{wrwa}			50	500	μs
每个芯片的重写次数	C _{erwr}	1 次擦除 + 1 次写入 = 1 次重写 ^{#2}			100	次

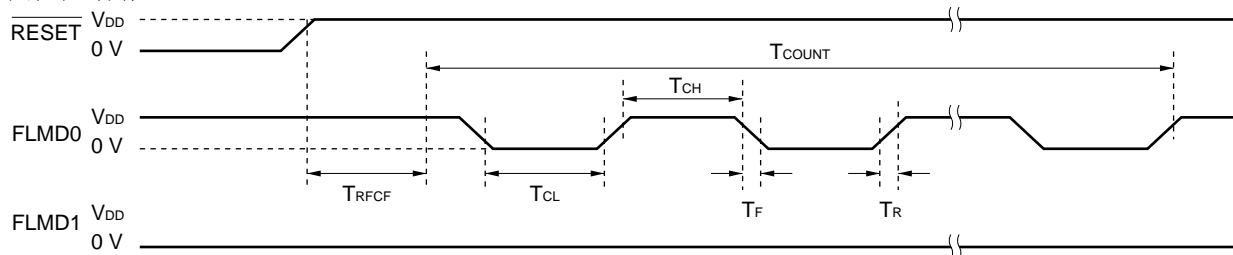
注 1. 擦除验证时间（回写时间）不包括。

2. 如果部分（2KB）写入 512 按字单元操作后擦除，重写操作次数是 1。通过一次擦除而向相同地址进行一次以上写操作是禁止的。

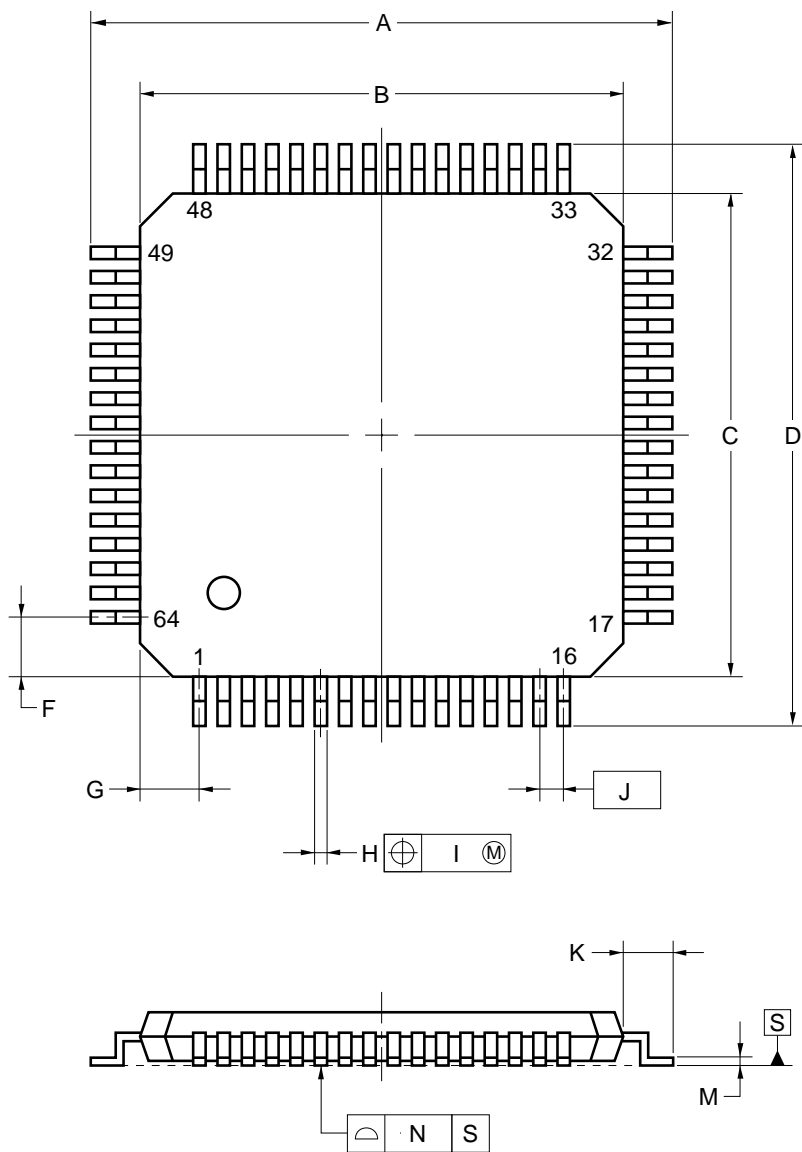
(2) 串行写入工作特性

参数	符号	条件	最小值	典型值	最大值	单位
从 RESET↑到 FLMD0 计数开始的时间	T _{RFCF}		$2^{19}/f_x + \alpha$			μs
计数执行时间	T _{COUNT}				10	ms
FLMD0 计数器高 / 低电平宽度	T _{CH} /T _{CL}		10		100	μs

串行写入操作



64引脚塑封 (12x12)



引脚末端的细节

项目	毫米
A	14.0±0.2
B	12.0±0.2
C	12.0±0.2
D	14.0±0.2
F	1.125
G	1.125
H	0.32 ^{+0.06} _{-0.10}
I	0.13
J	0.65 (T.P.)
K	1.0±0.2
L	0.5
M	0.17 ^{+0.03} _{-0.07}
N	0.10
P	1.0
Q	0.1±0.05
R	3° ^{+4°} _{-3°}
S	1.1±0.1
T	0.25
U	0.6±0.15

P64GK-65-9ET-3

注

在最大用料条件下，每个引脚的中心线位于其理想位置的0.13 mm (T.P.)以内。

30.1 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周边硬件进行通信的周边总线。

因为 CPU 总线的时钟与周边总线时钟异步，因此如果访问 CPU 与访问周边硬件冲突，那么不期望的非法数据将会被传输。

因此，当访问可能导致冲突的周边硬件时，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是一直等待。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 30-1）。执行实时处理时必须注意这点。

30.2 产生等待的周边硬件

表 30-1 表示了当从 CPU 访问产生等待请求的寄存器和等待的 CPU 时钟个数（当 VSWC = 0）。

表 30-1. 产生等待的寄存器以及 CPU 等待时钟的个数（当 VSWC = 0）

周边硬件	寄存器	访问	等待时钟的个数
看门狗定时器	WDTM	写入	3 个时钟（固定）
串行接口 UART00	ASIS00	读取	1 个时钟（固定）
A/D 转换器	ADM	写入	1 到 5 个时钟 ^注 （当 ADM.5 标志 = “1”）
	ADS	写入	1 到 9 个时钟 ^注 （当 ADM.5 标志 = “0”）
	PFM	写入	
	PFT	写入	
	ADCR	读取	1 到 5 个时钟 ^注 （当 ADM.5 标志 = “1”） 1 到 9 个时钟 ^注 （当 ADM.5 标志 = “0”）
<计算等待时钟的最大个数> $\{ (1/f_{\text{MACRO}}) \times 2 / (1/f_{\text{CPU}}) \} + 1$ *小数点后的结果乘以 (1/f _{CPU}) 如果小于 t _{CPUL} 则舍去，如果超过 t _{CPUL} 则向上进位。 f _{MACRO} : 宏工作频率 （当 ADM 的第 5 位 (FR2) = “1” : f _x /2, 当 ADM 的第 5 位 (FR2) = “0” : f _x /2 ²) f _{CPU} : CPU 时钟频率 t _{CPUL} : CPU 时钟的低电平宽度			

注 如果由上述表达式计算的等待时钟个数是 1，CPU 不产生等待周期。

备注 时钟为 CPU 时钟 (f_{CPU})。

30.3 等待发生示例

<1>看门狗定时器

<执行 MOV WDTM, A>

执行时钟个数: 8

(数据写入不产生等待的寄存器 (MOV sfr, A) 占用 5 个时钟.)

<执行 MOV WDTM, #byte >

执行时钟个数: 10

(数据写入不产生等待的寄存器 (MOV sfr, #byte) 占用 7 个时钟.)

<2>串行接口 UART00

<执行 MOV A, ASIS00>

执行时钟个数: 6

(数据从不产生等待的寄存器读出 (MOV A, sfr) 占用 5 个时钟.)

<3>A/D 转换器

表 30-2.当发生等待时等待时钟个数和执行时钟个数 (A/D 转换器)

<当执行 MOV ADM, A; MOV ADS, A; 或 MOV A, ADCR>

- 当 $f_x = 10 \text{ MHz}$, $t_{\text{CPUL}} = 50 \text{ ns}$

寄存器第 5 位 (FR2) 的值	f_{CPU}	等待时钟的个数	执行时钟个数
0	f_x	9 个时钟	14 个时钟
	$f_x/2$	5 个时钟	10 个时钟
	$f_x/2^2$	3 个时钟	8 个时钟
	$f_x/2^3$	2 个时钟	7 个时钟
	$f_x/2^4$	0 个时钟 (1 个时钟 ^注)	5 个时钟 (6 个时钟 ^注)
1	f_x	5 个时钟	10 个时钟
	$f_x/2$	3 个时钟	8 个时钟
	$f_x/2^2$	2 个时钟	7 个时钟
	$f_x/2^3$	0 个时钟 (1 个时钟 ^注)	5 个时钟 (6 个时钟 ^注)
	$f_x/2^4$	0 个时钟 (1 个时钟 ^注)	5 个时钟 (6 个时钟 ^注)

注 执行 MOV A, ADCR

备注 时钟为 CPU 时钟 (f_{CPU})。

f_x : X1 输入时钟频率

t_{CPUL} : CPU 时钟的低电平宽度

在使用 μ PD78F0714 的系统开发中可使用如下开发工具。

图 A-1 所示为开发工具的配置。

- **支持 PC98-NX 系列**

除非特别说明，IBM PC/AT™ 兼容机支持的产品与 PC98-NX 系列计算机是兼容的。当使用 PC98-NX 系列计算机时，参阅 IBM PC/AT 兼容机的使用说明。

- **Windows**

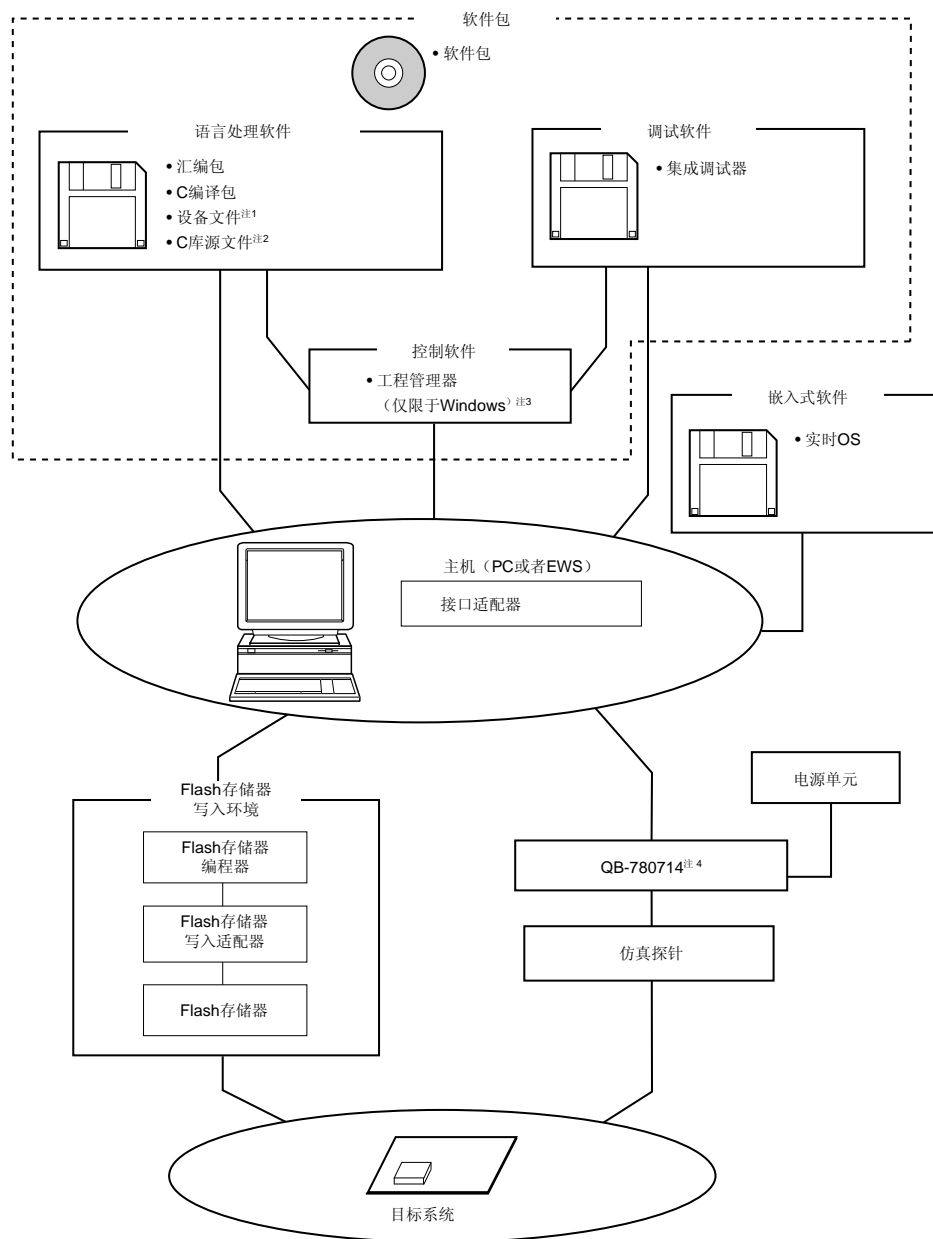
除非特别说明，“Windows 指以下几种操作系统。

- Windows 3.1
- Windows 95, Windows 98, Windows 2000, Windows XP
- Windows NT™ 版 4.0

注意事项 对于 μ PD78F0714 的开发工具，联系日电电子销售代表。

图 A-1. 开发工具的组成 (1/3)

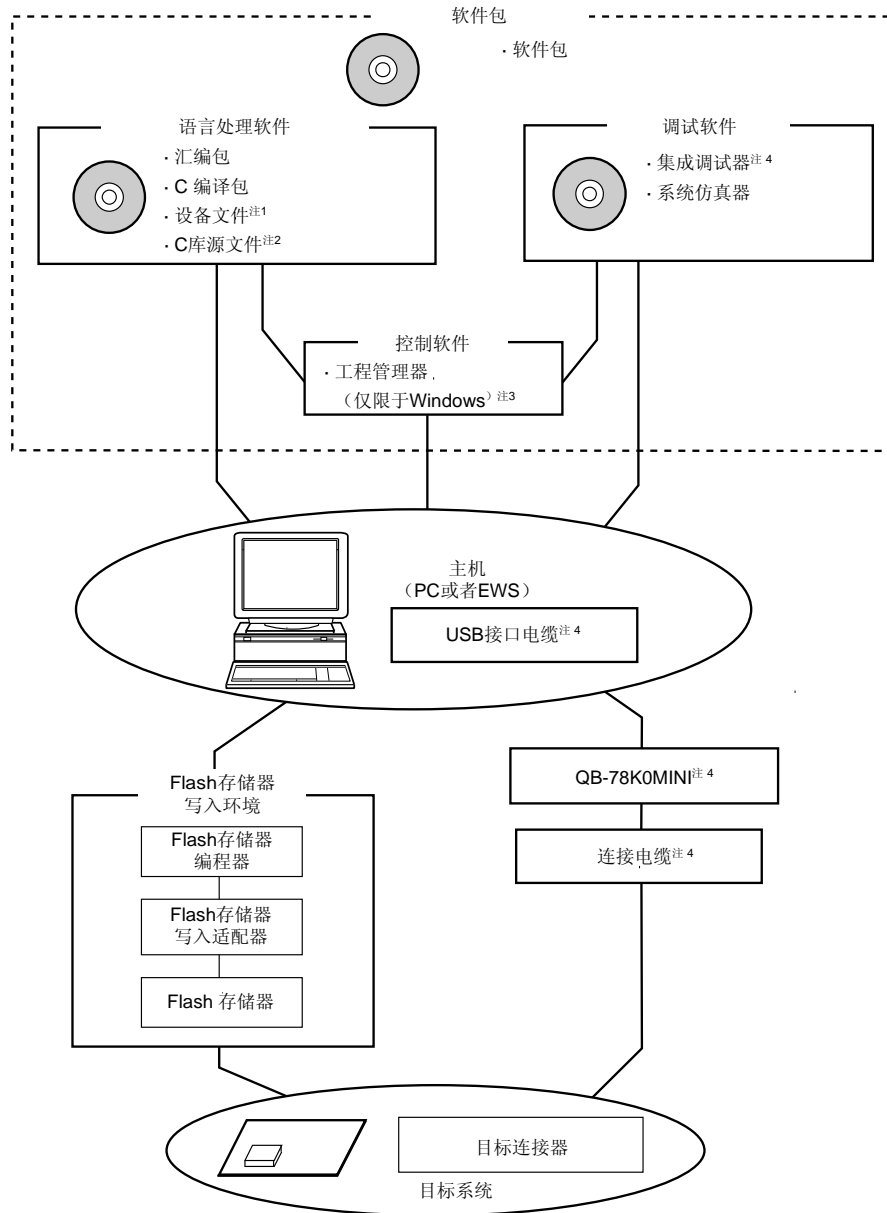
(1) 当使用在线仿真器 QB-780714 时



- 注
1. 从开发工具的下载站点中为 μ PD78F0714 (DF780714) 下载设备文件 (<http://www.necel.com/micro/ods/eng/index.html>)。
 2. C 语言库源文件不包含在软件包中。
 3. 项目管理器 PM+ 包含在汇编程序包中。
PM+ 仅用于 Windows。
 4. 在线仿真器 QB-780714 提供集成调试器 ID78K0-QB, flash 存储器编程器 PG-FPL (或 QB-MINI2), 电源单元和 USB 接口电缆。所有其他产品都需单独购买。从 MINICUBE2 的下载站点中下载用于操作 QB-MINI2 的软件 (<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>)。

图 A-1. 开发工具的组成 (2/3)

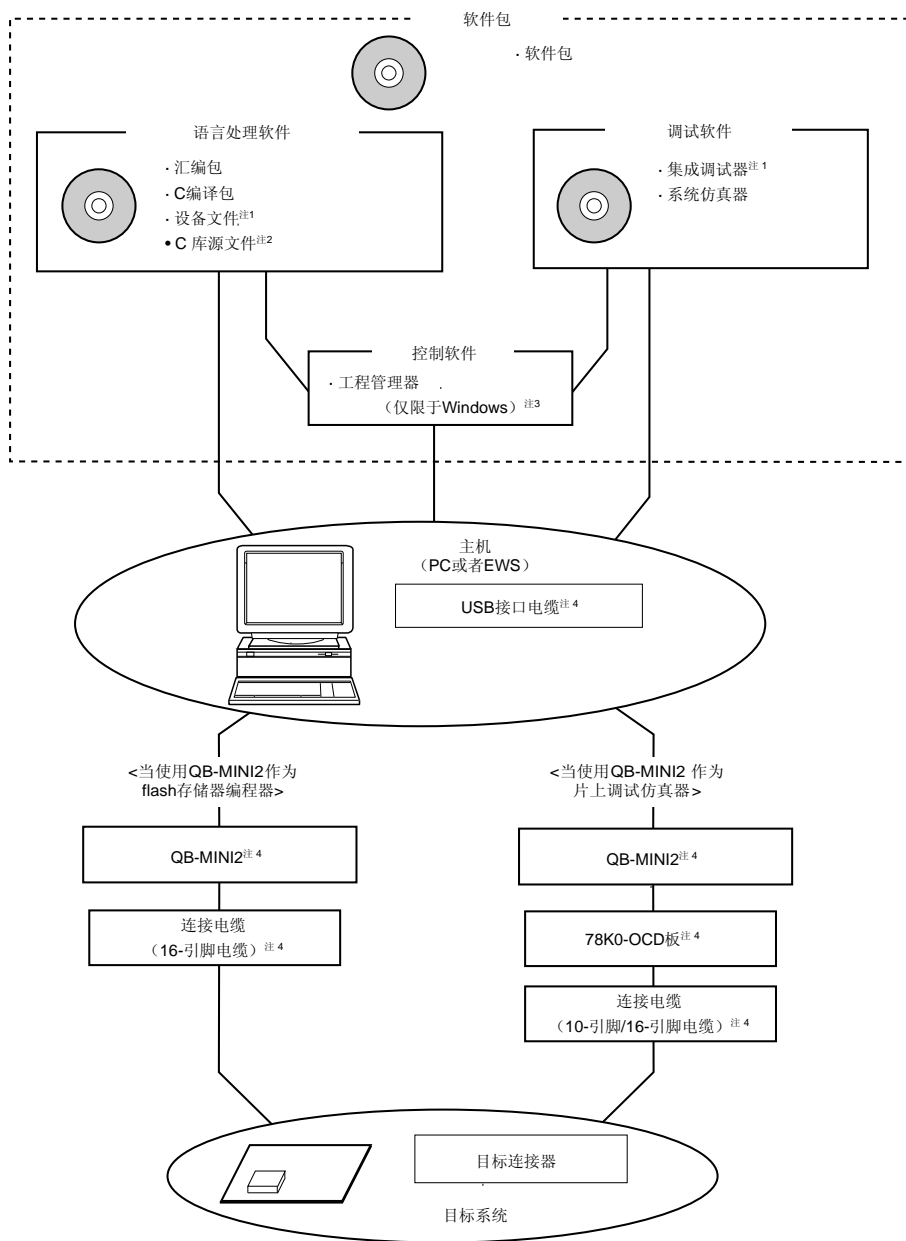
(2) 当使用片上调试仿真器 QB-78K0MINI 时



- 注
1. 从开发工具的下载站点中为 μ PD78F0714 (DF780714) 下载设备文件 ([http : //www.necel.com/micro/ods/eng/index.html](http://www.necel.com/micro/ods/eng/index.html))。
 2. C 语言库源文件不包含在软件包中。
 3. 项目管理器 PM+包含在汇编程序包中。
PM+ 仅用于 Windows。
 4. QB-78K0MINI 提供集成调试器 ID78K0-QB, USB 接口电缆和连接电缆。所有其他产品都需单独购买。

图 A-1. 开发工具的组成 (3 / 3)

(3) 当使用带有编程功能的片上调试仿真器 QB-MINI2 时



- 注
1. 从开发工具的下載站点中为 μ PD78F0714 (DF780714) 下载设备文件和集成调试器 (ID78K0-QB) (<http://www.necel.com/micro/ods/eng/index.html>)。
 2. C 语言库源文件不包含在软件包中。
 3. 项目管理器 PM+ 包含在汇编程序包中。
PM+ 仅用于 Windows。
 4. 片上调试仿真器 QB-MINI2 提供一个 USB 接口电缆, 连接电缆 (10 引脚和 16 引脚电缆) 以及 78K0-OCD 板。所有其他产品都需单独购买。从 MINICUBE2 的下载站点中下载用于操作 QB-MINI2 的软件 (<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>)。

A.1 软件包

SP78K0 78K/0 系列软件包	该软件包包含了适用于 78K/0 系列的开发工具（软件）。
	产品型号： μ SxxxxSP78K0

备注 产品型号中的xxxx随主机和使用的 OS 而变化。

μ SxxxxSP78K0

xxxx	主机	OS	存储介质
AB17	PC-9800 系列,	Windows (日语版)	CD-ROM
BB17	IBM PC/AT 兼容机	Windows (英语版)	

A.2 语言处理软件

RA78K0 汇编程序包	<p>该汇编器将以助记符来写的程序转换为微控制器可执行的目标代码。该汇编器还带有自动创建符号表的功能以及优先跳转指令的功能。该汇编器应该同设备文件（DF780714）（需另外购买）结合使用。</p> <p><在 PC 环境中使用 RA78K0 时的预防措施></p> <p>该汇编程序包是基于 DOS 的应用程序。然而，它也可以通过使用项目管理器（包含于汇编程序包）在 Windows 中进行使用。</p> <p>产品型号：μSxxxxRA78K0</p>
CC78K0 C 语言编译程序包	<p>该编译器将以 C 语言所写的程序转换为微控制器可执行的目标代码。该汇编器应该同汇编程序包以及设备文件（均需另外购买）结合使用。</p> <p><在 PC 环境中使用 CC78K0 时需要提前注意的地方></p> <p>该 C 语言编译程序包是基于 DOS 的应用程序。然而，它也可以通过使用项目管理器（包含于汇编程序包）在 Windows 中进行使用。</p> <p>产品型号：μSxxxxCC78K0</p>
DF780714 ^{註1} 设备文件	<p>该文件包含设备特有的信息。该设备文件应该同工具（RA78K0, CC78K0 以及 ID78K0-QB）（均需另外购买）结合使用。</p> <p>相应的 OS 和主机随使用工具而变化。</p> <p>产品型号：μSxxxxDF780714</p>
CC78K0-L ^{註2} C 语言库源文件	<p>该源文件用于配置 C 语言编译程序包中所包含的对象库。根据用户说明书，该文件需与 C 语言编译程序包中所包含的对象库匹配。</p> <p>产品型号：μSxxxxCC78K0-L</p>

- 注**
- DF780714 可以和 RA78K0, CC78K0, 以及 ID78K0-QB 一样来进行使用。从开发工具下载站点下载 DF780714 (<http://www.necel.com/micro/ods/eng/index.html>)。
 - CC78K0-L 不包含在软件包（SP78K0）中。

备注 产品型号中的xxxx随主机和使用的 OS 而变化。

μSxxxxRA78K0

μSxxxxCC78K0

μSxxxxCC78K0-L

xxxx	主机	OS	存储介质
AB17	PC-9800 系列, IBM PC/AT 兼容机	Windows (日语版)	CD-ROM
BB17		Windows (英语版)	
3P17	HP9000 系列 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4) Solaris™ (Rel. 2.5.1)	

μSxxxxDF780714

xxxx	主机	OS	存储介质
AB13	PC-9800 系列, IBM PC/AT 兼容机	Windows (日语版)	3.5 英寸 2HD FD
BB13		Windows (英语版)	

A.3 控制软件

PM+ 项目管理器	<p>这是一个控制软件，可使用户在 Windows 环境下有效地进行程序开发。用户程序开发中进行的任何操作，如启动编辑器、构造程序和启动调试器，都可以由项目管理器来执行。</p> <p><注意事项> 项目管理器包含于汇编程序包 (RA78K0) 中。 项目管理器仅在 Windows 下使用。</p>
--------------	---

A.4 Flash存储器编程工具

A.4.1 使用flash存储器编程器PG-FP5, FL-PR5, PG-FP4, FL-PR4 和PG-FPL时

FL-PR4, PG-FP4, PL-PR5, PG-FP5 Flash 存储器编程器	Flash 存储器编程器专用于带有片上 Flash 存储器的微控制器。
PG-FPL Flash 存储器编程器	Flash 存储器编程器专用于带有片上 Flash 存储器的微控制器。 包括选择仿真器 QB-780714.
FA-64GK-9ET-A Flash 存储器编程器适配器	Flash 存储器编程器适配器用于连接 Flash 存储器编程器。 • FA-64GK-9ET-A: 64 引脚塑封 TQFP (GK-9ET 类型)

- 备注**
1. FL-PR5, FL-PR4, FA-64GK-9ET-A 是 Naito Densei Machida Mfg.Co., Ltd 的产品。
电话: +81-42-750-4172 Naito Densei Machida Mfg.Co., Ltd.
 2. 使用 flash 存储器编程适配器的最新版本。

A.4.2 使用带有编程功能的片上调试仿真器QB-MINI2 时

QB-MINI2 带有编程功能的片上调试仿真器	这是一个简易 flash 存储器编程器, 专用于带有片上 flash 存储器的微控制器。当使用 μ PD78F0714 开发应用系统时, 作为片上仿真器能够调试硬件和软件。当使用它作为 flash 存储器编程器时, 该仿真器应与连接电缆 (16 引脚电缆) 以及用于连接主机的 USB 接口电缆结合使用。
目标连接器规格	16 引脚通用连接器 (2.54 mm 脚距)

- 备注**
1. QB-MINI2 提供有一个 USB 接口电缆, 连接电缆 (10 引脚电缆和 16 引脚电缆) 以及 78K0-OCD 板。连接电缆 (10 引脚电缆) 和 78 K0-OCD 板仅用于使用片上调试功能。
 2. 从 MINICUBE2 的下载站点中下载用于操作 QB-MINI2 的软件 ([http : //www.necel.com/micro/en/development/asia/minicube2/minicube2.html](http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html)) 。

A.5 调试工具（硬件）

A.5.1 当使用在线仿真器QB-780714

QB-780714 ^注 在线仿真器	当使用 μ PD78F0714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）。仿真器应当与电源单位和仿真探头配合使用。USB用于连接仿真器和主机。
QB-144-CA-01 检测引脚适配器	该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-64GK-EA-01T 交换适配器	该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-EA-01T: 64引脚塑封TQFP（GK-9ET 类型）
QB-64GK-YS-01T 空间适配器	如果需要，该适配器用于调整目标系统与在线仿真器间的高度。 • QB-64GK-YS-01T: 64引脚塑封TQFP（GK-9ET 类型）
QB-64GK-YQ-01T YQ 连接器	该连接器用于连接目标连接器和交换适配器。 • QB-64GK-YQ-01T: 64引脚塑封TQFP（GK-9ET 类型）
QB-64GK-HQ-01T 安装适配器	这个适配器用于使用套接字将目标设备挂接到目标设备。 • QB-64GK-HQ-01T: 64引脚塑封TQFP（GK-9ET 类型）带片上调试功能
QB-64GK-NQ-01T 目标连接器	这个连接器用于挂接在线仿真器到目标系统中 • QB-64GK-NQ-01T: 64引脚塑封TQFP（GK-9ET 类型）

注 QB-780714 提供电源单位，USB 接口电缆和 flash 存储器编程器 PG-FPL。它也提供集成调试器 ID78K0-QB 作为控制软件。

备注 封装内容随产品型号而变化。

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-780714-ZZZ	QB-780714	未包含			
QB-780714-T30MC		QB-80-EP-01T	QB-64GK-EA-01T	QB-64GK-YQ-01T	QB-64GK-NQ-01T

A.5.2 当使用片上调试仿真器QB-78K0MINI

QB-78K0MINI ^注 片上调试仿真器	当使用 μ PD78F0714产品进行应用系统开发时，片上调试仿真器用来调试硬件和软件。它支持集成调试器（ID78K0-QB）随QB-78K0MINI提供。仿真器通过连接电缆和USB接口电缆与主机进行连接
目标连接器规格	10引脚通用连接器（2.54 mm 脚距）

注 QB-78K0MINI 提供一条 USB 接口电缆和一条连接电缆。它也提供集成调试器 ID78K0-QB 作为控制软件。

A.5.3 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	当使用 μ PD78F0714 产品进行应用系统开发时，片上调试仿真器用来调试硬件和软件。这是一个简易 flash 存储器编程器，专用于带有片上 flash 存储器的微控制器。当使用它作为片上调试仿真器时，该仿真器应与连接电缆（10 引脚电缆或 16 引脚电缆）以及用于连接主机的 USB 接口电缆和 78K0-OCD 板结合使用。
目标连接器规格	10 引脚通用目的连接器（2.54 mm 间距）或 16 引脚通用目的连接器（2.54 mm 间距）

- 备注**
1. QB-MINI2 提供有一个 USB 接口电缆，连接电缆（10 引脚电缆和 16 引脚电缆）以及 78K0-OCD 板。连接电缆（10 引脚电缆）和 78 K0-OCD 板仅用于使用片上调试功能。
 2. 从 MINICUBE2 的下载站点中下载用于操作 QB-MINI2 的软件（[http : //www.necel.com/micro/en/development/asia/minicube2/minicube2.html](http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html)）。

A.6 调试工具（软件）

ID78K0-QB 集成调试器	该调试器支持 78K/0 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，并且能够通过使用结合了源程序，分别显示以及带有跟踪结果的存储器显示的集成窗口功能来显示源程序的追踪结果。它应该同设备文件（单独销售）一起使用。
	产品型号： μ SxxxxID78K0-QB

备注 产品型号中的xxxx随主机和使用的 OS 而变化。

μ SxxxxID78K0-QB

xxxx	主机	OS	存储介质
AB17	PC-9800 系列，	Windows（日语版）	CD-ROM
BB17	IBM PC/AT 兼容机	Windows（英语版）	

附录B 寄存器索引

B.1 寄存器索引（按寄存器名称的字母排序）

[A]

A/D 转换结果寄存器 (ADCR)	270
A/D 转换器模式寄存器 (ADM)	266
A/D 触发选择寄存器 (TW0TRGS)	114
模拟输入通道指定寄存器 (ADS)	269
异步串行接口操作模式寄存器 00 (ASIM00)	292
异步串行接口接收异常错误状态寄存器 00 (ASIS00)	294

[B]

波特率产生器控制寄存器 00 (BRGC00)	295
缓存传输控制定时器 (RTM0)	109

[C]

捕获 / 比较控制寄存器 (IT20CCR)	134
捕获 / 比较控制寄存器 00 (CRC00)	159
时钟输出选择寄存器 (CKS)	234

[D]

直流控制寄存器 0 (DCCTL0)	247
直流控制寄存器 1 (DCCTL1)	248
死区时间重新加载寄存器 (TW0DTIME)	109
死区时间定时器 0 (DTM0)	109
死区时间定时器 1 (DTM1)	109
死区时间定时器 2 (DTM2)	109

[E]

8 位定时器比较寄存器 50 (CR50)	193
8 位定时器比较寄存器 51 (CR51)	193
8 位定时器计数器 50 (TM50)	192
8 位定时器计数器 51 (TM51)	192
8 位定时器 H 比较寄存器 00 (CMP00)	210
8 位定时器 H 比较寄存器 01 (CMP01)	210
8 位定时器 H 模式寄存器 0 (TMHMD0)	211
8 位定时器模式控制寄存器 50 (TMC50)	196
8 位定时器模式控制寄存器 51 (TMC51)	196
外部中断下降沿允许寄存器 (EGN)	345
外部中断上升沿允许寄存器 (EGP)	345

[F]

Flash 编程模式控制寄存器 (FLPMC)	407
Flash 保护命令寄存器 (PFCMD)	409
Flash 状态寄存器 (PFS)	409

[I]

内部存储器大小切换寄存器 (IMS)	391
内部振荡器模式寄存器 (RCM)	87
中断屏蔽标志寄存器 0H (MK0H)	343
中断屏蔽标志寄存器 0L (MK0L)	343
中断屏蔽标志寄存器 1H (MK1H)	343
中断屏蔽标志寄存器 1L (MK1L)	343
中断请求标志寄存器 0H (IF0H)	341
中断请求标志寄存器 0L (IF0L)	341
中断请求标志寄存器 1H (IF1H)	341
中断请求标志寄存器 1L (IF1L)	341
逆变器定时器控制寄存器 (TW0C)	110
逆变器定时器模式寄存器 (TW0M)	112
逆变器定时器输出控制寄存器 (TW0OC)	114

[L]

低电压检测寄存器 (LVIM)	381
-----------------	-----

[M]

主时钟模式寄存器 (MCM)	88
主 OSC 控制寄存器 (MOC)	89
乘法 / 除法数据寄存器 A0 (MDA0H,MDA0L)	328
乘法 / 除法数据寄存器 B0 (MDB0)	329
乘法器 / 除法器控制寄存器 0 (DMUC0)	330

[N]

噪声消除时间选择寄存器 1 (NRC1)	138
----------------------	-----

[O]

振荡器稳定时间计数器状态寄存器 (OSTC)	90, 358
振荡器稳定时间选择寄存器 (OSTS)	91, 359

[P]

端口模式寄存器 0 (PM0)	78
端口模式寄存器 1 (PM1)	78
端口模式寄存器 3 (PM3)	78
端口模式寄存器 4 (PM4)	78
端口模式寄存器 5 (PM5)	78
端口模式寄存器 6 (PM6)	78
端口模式寄存器 7 (PM7)	78
端口寄存器 0 (P0)	80
端口寄存器 1 (P1)	80
端口寄存器 2 (P2)	80
端口寄存器 3 (P3)	80
端口寄存器 4 (P4)	80
端口寄存器 5 (P5)	80
端口寄存器 6 (P6)	80
端口寄存器 7 (P7)	80

掉电比较模式寄存器 (PFM)	271
掉电比较门限寄存器 (PFT)	271
预分频器模式寄存器 (IT20PRM)	136
预分频器模式寄存器 00 (PRM00)	161
优先级指定标志寄存器 0H (PR0H)	344
优先级指定标志寄存器 0L (PR0L)	344
优先级指定标志寄存器 1H (PR1H)	344
优先级指定标志寄存器 1L (PR1L)	344
处理器时钟控制寄存器 (PCC)	86
上拉电阻选项寄存器 0 (PU0)	81
上拉电阻选项寄存器 1 (PU1)	81
上拉电阻选项寄存器 3 (PU3)	81
上拉电阻选项寄存器 4 (PU4)	81
上拉电阻选项寄存器 5 (PU5)	81
上拉电阻选项寄存器 6 (PU6)	81
上拉电阻选项寄存器 7 (PU7)	81

[R]

实时输出缓存寄存器 0H (RTBH00)	241
实时输出缓存寄存器 0L (RTBL00)	241
实时输出缓存寄存器 1H (RTBH01)	242
实时输出缓存寄存器 1L (RTBL01)	242
实时输出端口控制寄存器 0 (RTPC00)	245
实时输出端口控制寄存器 1 (RTPC01)	246
实时输出端口模式寄存器 0 (RTPM00)	243
实时输出端口模式寄存器 1 (RTPM01)	244
接收缓存寄存器 00 (RXB00)	291
余数数据寄存器 0 (SDR0)	327
复位控制标志寄存器 (RESF)	375

[S]

串行时钟选择寄存器 10 (CSIC10)	313
串行 I/O 移位寄存器 10 (SIO10)	311
串行操作模式寄存器 10 (CSIM10)	312
状态寄存器 (IT20STS)	137
系统等待控制寄存器 (VSWC)	92
16 位捕获 / 比较寄存器 0 (IT20CC0)	129
16 位捕获 / 比较寄存器 1 (IT20CC1)	130
16 位比较寄存器 0 (IT20CM0)	127
16 位比较寄存器 1 (IT20CM1)	128
16 位定时器捕获 / 比较寄存器 00 (CR00)	154
16 位定时器捕获 / 比较寄存器 01 (CR01)	156
16 位定时器计数器 00 (TM00)	154
16 位定时器模式控制寄存器 00 (TMC00)	157
16 位定时器输出控制寄存器 00 (TOC00)	159
16 位向上 / 向下计数器 (IT20UDC)	125

[T]

10 位缓存寄存器 0 (TW0BFCM0)	109
10 位缓存寄存器 1 (TW0BFCM1)	109
10 位缓存寄存器 2 (TW0BFCM2)	109
10 位缓存寄存器 3 (TW0BFCM3)	109
10 位缓存寄存器 4 (TW0BFCM4)	109
10 位缓存寄存器 5 (TW0BFCM5)	109
10 位比较寄存器 0 (TW0CM0)	108
10 位比较寄存器 1 (TW0CM1)	108
10 位比较寄存器 2 (TW0CM2)	108
10 位比较寄存器 3 (TW0CM3)	108
10 位比较寄存器 4 (TW0CM4)	108
10 位比较寄存器 5 (TW0CM5)	108
10 位向上 / 向下计数器 (TW0UDC)	108
定时器时钟选择寄存器 50 (TCL50)	194
定时器时钟选择寄存器 51 (TCL51)	194
定时器控制寄存器 (IT20TMC)	133
定时器单位模式寄存器 (IT20TUM)	132
发送缓存寄存器 10 (SOTB10)	311
发送移位寄存器 00 (TXS00)	291

[V]

有效沿选择寄存器 (IT20SESA)	135
---------------------------	-----

[W]

看门狗定时器允许寄存器 (WDTE)	227
看门狗定时器模式寄存器 (WDTM)	226

B.2 寄存器索引（按寄存器符号的字母排序）

[A]

ADCR:	A/D 转换结果寄存器.....	270
ADM:	A/D 转换器模式寄存器.....	266
ADS:	模拟输入通道指定寄存器.....	269
ASIM00:	异步串行接口操作模式寄存器 00.....	292
ASIS00:	异步串行接口接收异常错误状态寄存器 00.....	294

[B]

BRGC00:	波特率产生器控制寄存器 00.....	295
---------	---------------------	-----

[C]

CKS:	时钟输出选择寄存器.....	234
CMP00:	8 位定时器 H 比较寄存器 00.....	210
CMP01:	8 位定时器 H 比较寄存器 01.....	210
CR00:	16 位定时器捕获 / 比较寄存器 00.....	154
CR01:	16 位定时器捕获 / 比较寄存器 01.....	156
CR50:	8 位定时器比较寄存器 50.....	193
CR51:	8 位定时器比较寄存器 51.....	193
CRC00:	捕获 / 比较控制寄存器 00.....	159
CSIC10:	串行时钟选择寄存器 10.....	313
CSIM10:	串行操作模式寄存器 10.....	312

[D]

DCCTL0:	直流控制寄存器 0.....	247
DCCTL1:	直流控制寄存器 1.....	248
DMUC0:	乘法器 / 除法器控制寄存器 0.....	330
DTM0:	死区时间定时器 0.....	109
DTM1:	死区时间定时器 1.....	109
DTM2:	死区时间定时器 2.....	109

[E]

EGN:	外部中断下降沿允许寄存器.....	345
EGP:	外部中断上升沿允许寄存器.....	345

[F]

FLPMC:	Flash 编程模式控制寄存器.....	407
--------	----------------------	-----

[I]

IF0H:	中断请求标志寄存器 0H.....	341
IF0L:	中断请求标志寄存器 0L.....	341
IF1H:	中断请求标志寄存器 1H.....	341
IF1L:	中断请求标志寄存器 1L.....	341
IMS:	内部存储器大小切换寄存器.....	391
IT20CC0:	16 位捕获 / 比较寄存器 0.....	129
IT20CC1:	16 位捕获 / 比较寄存器 1.....	130

IT20CCR:	捕获 / 比较控制寄存器.....	134
IT20CM0:	16 位比较寄存器 0.....	127
IT20CM1:	16 位比较寄存器 1.....	128
IT20PRM:	预分频器模式寄存器.....	136
IT20SESA:	有效沿选择寄存器.....	135
IT20STS:	状态寄存器.....	137
IT20TMC:	定时器控制寄存器.....	133
IT20TUM:	定时器单位模式寄存器.....	132
IT20UDC:	16 位向上 / 向下计数器.....	125

[L]

LVIM:	低电压检测寄存器.....	381
-------	---------------	-----

[M]

MCM:	主时钟模式寄存器.....	88
MDA0H:	乘法 / 除法数据寄存器 A0.....	328
MDA0L:	乘法 / 除法数据寄存器 A0.....	328
MDB0:	乘法 / 除法数据寄存器 B0.....	329
MK0H:	中断屏蔽标志寄存器 0H.....	343
MK0L:	中断屏蔽标志寄存器 0L.....	343
MK1H:	中断屏蔽标志寄存器 1H.....	343
MK1L:	中断屏蔽标志寄存器 1L.....	343
MOC:	主 OSC 控制寄存器.....	89

[N]

NRC1:	噪声消除时间选择寄存器 1.....	138
-------	--------------------	-----

[O]

OSTC:	振荡器稳定时间计数器状态寄存器.....	90, 358
OSTS:	振荡器稳定时间选择寄存器.....	91, 359

[P]

P0:	端口寄存器 0.....	80
P1:	端口寄存器 1.....	80
P2:	端口寄存器 2.....	80
P3:	端口寄存器 3.....	80
P4:	端口寄存器 4.....	80
P5:	端口寄存器 5.....	80
P6:	端口寄存器 6.....	80
P7:	端口寄存器 7.....	80
PCC:	处理器时钟控制寄存器.....	86
PFCMD:	Flash 保护命令寄存器.....	409
PFM:	掉电比较模式寄存器.....	271
PFS:	Flash 状态寄存器.....	409
PFT:	掉电比较门限寄存器.....	271
PM0:	端口模式寄存器 0.....	78
PM1:	端口模式寄存器 1.....	78
PM3:	端口模式寄存器 3.....	78
PM4:	端口模式寄存器 4.....	78

PM5:	端口模式寄存器 5	78
PM6:	端口模式寄存器 6	78
PM7:	端口模式寄存器 7	78
PR0H:	优先级指定标志寄存器 0H	344
PR0L:	优先级指定标志寄存器 0L	344
PR1H:	优先级指定标志寄存器 1H	344
PR1L:	优先级指定标志寄存器 1L	344
PRM00:	预分频器模式寄存器 00	161
PU0:	上拉电阻选项寄存器 0	81
PU1:	上拉电阻选项寄存器 1	81
PU3:	上拉电阻选项寄存器 3	81
PU4:	上拉电阻选项寄存器 4	81
PU5:	上拉电阻选项寄存器 5	81
PU6:	上拉电阻选项寄存器 6	81
PU7:	上拉电阻选项寄存器 7	81
[R]		
RCM:	内部振荡器模式寄存器	87
RESF:	复位控制标志寄存器	375
RTBH00:	实时输出缓冲寄存器 0H	241
RTBH01:	实时输出缓冲寄存器 1H	242
RTBL00:	实时输出缓冲寄存器 0L	241
RTBL01:	实时输出缓冲寄存器 1L	242
RTM0:	缓存传输控制定时器	109
RTPC00:	实时输出端口控制寄存器 0	245
RTPC01:	实时输出端口控制寄存器 1	246
RTPM00:	实时输出端口模式寄存器 0	243
RTPM01:	实时输出端口模式寄存器 1	244
RXB00:	接收缓存寄存器 00	291
[S]		
SDR0:	余数数据寄存器 0	327
SIO10:	串行 I/O 移位寄存器 10	311
SOTB10:	发送缓存寄存器 10	311
[T]		
TCL50:	定时器时钟选择寄存器 50	194
TCL51:	定时器时钟选择寄存器 51	194
TM00:	16 位定时器计数器 00	154
TM50:	8-定时器计数器 50	192
TM51:	8 位定时器计数器 51	192
TMC00:	16 位定时器模式控制寄存器 00	157
TMC50:	8 位定时器模式控制寄存器 50	196
TMC51:	8 位定时器模式控制寄存器 51	196
TMHMD0:	8 位定时器 H 模式寄存器 0	211
TOC00:	16 位定时器输出控制寄存器 00	159
TW0BFCM0:	10 位缓存寄存器 0	109
TW0BFCM1:	10 位缓存寄存器 1	109

TW0BFCM2:	10 位缓存寄存器 2	109
TW0BFCM3:	10 位缓存寄存器 3	109
TW0BFCM4:	10 位缓存寄存器 4	109
TW0BFCM5:	10 位缓存寄存器 5	109
TW0C:	逆变器定时器控制寄存器	110
TW0CM0:	10 位比较寄存器 0	108
TW0CM1:	10 位比较寄存器 1	108
TW0CM2:	10 位比较寄存器 2	108
TW0CM3:	10 位比较寄存器 3	108
TW0CM4:	10 位比较寄存器 4	108
TW0CM5:	10 位比较寄存器 5	108
TW0DTIME:	死区时间重新加载寄存器	109
TW0M:	逆变器定时器模式寄存器	112
TW0OC:	逆变器定时器控制寄存器	114
TW0TRGS:	A/D 触发选择寄存器	114
TW0UDC:	10 位向上 / 向下计数器	108
TXS00:	发送移位寄存器 00	291
[V]		
VSWC:	系统等待控制寄存器	92
[W]		
WDTE:	看门狗定时器允许寄存器	227
WDTM:	看门狗定时器模式寄存器	226

C.1 此版本的主要修订记录

(1/3)

页码	内容	类别
介绍		
p.7	修改开发工具（软件）（用户手册）相关的文档	(e)
p.7	修改开发工具（硬件）（用户手册）相关的文档	(e)
第一章 综述		
p.17	修改 1.2 应用	(c)
p.21	修改 1.6 功能概述中的“中断向量源”	(d)
第二章 引脚功能		
p.28	增加 2.2.3 (1) 端口模式的注意事项	(c)
第四章 CPU 架构		
p.51	修改 3.3.3 间接寻址列表	(c)
第五章 端口功能		
p.70	增加 4.2.3 端口 2 的注意事项	(c)
p.83	增加 4.5 端口寄存器 n (Pn) 1 位操作指令的注意事项	(c)
第六章 10 位逆变器控制定时器		
p.107	修改图 6-1 10-位逆变器控制定时器框图	(a)
第十三章 实时输出端口		
pp.239, 240	修改图 13-1 实时输出端口框图	(a)
p.241	删除表 13-2 实时输出缓冲寄存器 0 操作期间的操作中的注 1	(c)
p.242	删除表 13-3 实时输出缓冲寄存器 1 操作期间的操作中的注 1	(c)
p.244	修改图 13-6 实时输出端口模式寄存器 1 的格式	(c)
p.255	修改图 13-7 控制寄存器每位设置与实时输出之间的关系	(a)
第十四章 直流逆变器控制功能		
p.261	增加内容	(c)
第十五章 A/D 转换器		
p.266	修改图 15-3 A/D 转换器模式寄存器 (ADM) 的格式	(b)
p.268	增加表 15-3 A/D 转换时间	(b)
第十六章 串行接口 UART0		
p.288	增加 16.1 (2) 异步串行接口 (UART) 模式的注意事项	(c)
p.291	增加 16.2 (3) 发送移位寄存器 00 (TXS00) 的注意事项	(c)
p.293	修改图 16-2 异步串行接口操作模式寄存器 (ASIM00) 的格式	(c)
p.298	修改 16.4.2 (1) 使用的寄存器中设置操作的处理	(c)

备注 上表中的“类别”区别如下的版本。

(a)：错误纠正，(b)：指定的添加/修改，(c)：内容或注的添加/修改，(d)：封装，部件号或管理部分的添加/修改，(e)：相关文档的添加/修改

页码	内容	类别
第十九章 中断功能		
p.335	增加 19.1 (1) 非屏蔽中断	(b)
p.336	修改表 19-1 中断源列表 (1/2)	(b)
p.337	图 19-1 中断功能的基本配置 (1/2) 增加 (A) 内部非屏蔽中断	(b)
p.342	修改图 19-2 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H) 的格式	(b)
p.343	修改图 19-3 中断屏蔽标志寄存器 (MK0L, MK0H, MK1L, MK1H) 的格式	(b)
p.344	修改图 19-4 优先级指定标志寄存器 (PR0L, PR0H, PR1L, PR1H) 的格式	(b)
p.347	增加 19.4.1 非屏蔽中断请求响应操作	(b)
p.352	增加 19.4.4 多重中断服务的内容	(b)
p.352	修改表 19-5 在中断服务期间多重中断请求的关系	(b)
第二十章 待机功能		
p.361	增加 (b) 非屏蔽中断的释放到 (2) HALT 模式释放	(b)
p.363	修改表 20-3 HALT 模式中断请求的响应操作	(b)
p.366	增加 (b) 非屏蔽中断的释放到 (2) STOP 模式释放	(b)
p.367	修改表 20-5 STOP 模式中断请求的响应操作	(b)
第二十三章 低电压检测电路		
p.382	修改 23.4 (1) 当作为复位使用	(b)
p.383	修改图 23-3 低电压检测电路内部复位信号产生的时序	(b)
p.384	修改 23.4 (2) 当作为中断使用	(b)
p.385	修改图 23-4 低电压检测电路中信号产生的时序	(b)
第二十四章 选项字节		
p.390	修改图 24-2 选项字节的格式	(b)
第二十五章 FLASH 存储器		
p.404	修改表 25-5 通信模式并添加注	(c)
p.406	修改 25.7 Flash 存储器自写入编程中的备注	(e)
第二十六章 片上调试功能		
p.413	修改内容和备注	(e)
第二十七章 指令集		
p.422	修改 27.2 操作列表中的 CALLT	(c)

备注 上表中的“类别”区别如下的版本。

(a)：错误纠正，(b)：指定的添加/修改，(c)：内容或注的添加/修改，(d)：封装，部件号或管理部分的添加/修改，(e)：相关文档的添加/修改

页码	内容	类别
第二十八章 电气特性		
p.427	修改 注意事项	(c)
p.430	修改在 直流特性 (1/2) 中的 输出电压, 高和输出电压, 低	(b)
p.431	修改在 直流特性 (2/2) 中的 供电电流	(b)
p.432	在 交流特性 中修改 指令周期 并添加 ADTRG 输入高电平宽度, 低电平宽度	(b)
pp.433, 436	修改 交流特性 中 Tcy 对 VDD (主系统时钟工作) 并添加 A/D 触发输入时序	(b)
p.437	修改 A/D 转换器特性 中的 转换时间	(b)
p.438	修改 数据存储器停止模式低电源电压数据保持特性	(b)
p.439	修改 Flash 存储器编程特性 中的 (1) 基本特性	(b)
附录 A 开发工具		
全部	章节修订	(c)
附录 C 修订记录		
p.461	增加章节	(c)

备注 上表中的“类别”区别如下的版本。

(a)：错误纠正， (b)：指定的添加/修改， (c)：内容或注的添加/修改， (d)：封装，部件号或管理部分的添加/修改， (e)：相关文档的添加/修改

详细信息请联系：

中国区

MCU 技术支持热线：

电话：+86-400-700-0606 (普通话)

服务时间：9:00-12:00，13:00-17:00（不含法定节假日）

网址：

<http://www.cn.necel.com/>（中文）

<http://www.necel.com/>（英文）

[北京]

日电电子（中国）有限公司

中国北京市海淀区知春路 27 号

量子芯座 7，8，9，15 层

电话：（+86）10-8235-1155

传真：（+86）10-8235-7679

[深圳]

日电电子（中国）有限公司深圳分公司

深圳市福田区益田路卓越时代广场大厦 39 楼

3901，3902，3909 室

电话：（+86）755-8282-9800

传真：（+86）755-8282-9899

[上海]

日电电子（中国）有限公司上海分公司

中国上海市浦东新区银城中路 200 号

中银大厦 2409-2412 和 2509-2510 室

电话：（+86）21-5888-5400

传真：（+86）21-5888-5230

[香港]

香港日电电子有限公司

香港九龙旺角太子道西 193 号新世纪广场

第 2 座 16 楼 1601-1613 室

电话：（+852）2886-9318

传真：（+852）2886-9022

2886-9044

上海恩益禧电子国际贸易有限公司

中国上海市浦东新区银城中路 200 号

中银大厦 2511-2512 室

电话：（+86）21-5888-5400

传真：（+86）21-5888-5230

[成都]

日电电子（中国）有限公司成都分公司

成都市二环路南三段 15 号天华大厦 7 楼 703 室

电话：(+86)28-8512-5224

传真：(+86)28-8512-5334

[长春]

日电电子（中国）有限公司长春分公司

吉林省长春市朝阳区

西安大路 727 号中银大厦 A 座 1609 室

电话：(+86)431-8859-7533 / 8859-8533

传真：(+86)431-8680-2944

[大连]

日电电子（中国）有限公司长春分公司

大连市中山路 88 号天安国际大厦 2701 室

电话：(+86)411-8230-8815 / 8230-8825

传真：(+86)411-8230-8835