

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.



用户手册初稿

μ PD78F9510, 78F9511, 789512

8 位单片微控制器

μ PD78F9510

μ PD78F9511

μ PD78F9512

文档编号. U18372CA1V0UD00 (第一版)
发行日期 2009 年 1 月 NS CP(K)

© NEC Electronics Corporation 2007
日本印刷

[备注]

CMOS 设备注意事项

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。

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。

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。

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。

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。

电源关闭状态下的输入信号

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

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

PC/AT是International Business Machines Corporation的商标。

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

SPARCstation是SPARC International, Inc.的商标。

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

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

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

- 本档所登载的内容有效期截止至 2009 年 01 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。

引言

目标读者 本手册适用于那些希望理解 μ PD78F9510, 78F9511, 78F9512 产品功能，并设计开发相关应用系统和程序的用户工程师。

目的 本手册用于帮助用户了解如下**组织**中描述的功能。

组织 μ PD78F9510, 78F9511, 78F9512 有两本手册：本手册和指令手册(78K/0S 系列通用)。

μ PD78F9510, 78F9511,
78F9512
用户手册

- 引脚功能
- 内部模块功能
- 中断
- 其他内置外设功能
- 电气特性

78K/0S 系列
指令用户手册

- CPU 功能
- 指令集
- 指令描述

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

要掌握 μ PD78F9510, 78F9511, 78F9512 的全部功能和用法

→ 请按**目录**顺序阅读本手册。

◇ 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。

◇ 如何获悉某已知名称的寄存器详细功能

→ 请参阅 **附录 C 寄存器索引**。

◇ 如何获悉 78K/0S 系列指令的详细信息

→ 请参阅 **78K/0S 系列指令用户手册 (U11047E)**。

◇ 如何获悉 μ PD78F9510, 78F9511, 78F9512 的电气特性

→ 请参阅 **第十七章 电气特性 (目标)**。

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

相关文档 本手册中提到的相关文档可能包括有初稿版本。但是, 初稿版本没有特别注明。

设备相关文档

文档名称	文档编号
μPD78F9510, 78F9511, 78F9512 用户手册	本手册
78K0S78K0S 系列指令用户手册	U11047E

开发软件工具相关文档 (用户手册)

文档名称	文档编号	
RA78K0S 汇编包	操作篇	U16656E
	语言篇	U14877E
	结构化汇编语言篇	U11623E
CC78K0S C 编译器	操作篇	U16654E
	语言篇	U14872E
ID78K0S-NS Ver. 2.52 集成调试器	操作篇	U16584E
ID78K0S-QB Ver. 2.81 集成调试器	操作篇	U17287E
PM+ Ver.5.20		U16934E

开发硬件工具相关文档 (用户手册)

文档名称	文档编号
IE-78K0S-NS 在线仿真器	U13549E
IE-78K0S-NS-A 在线仿真器	U15207E
QB-78K0SKX1MINI 在线仿真器	U17272E

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

Flash存储器编程的相关文档

文档名称	文档编号
PG-FP4 Flash 存储器编程器用户手册	U15260E
PG-FPL2 Flash 存储器编程器用户手册	U17307E

其它相关文档

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任何声明条件下修改。读者开发设计时，应该使用每个文档的最新版本。

目录

第一章 概述	14
1.1 特征.....	14
1.2 订购信息	15
1.3 引脚配置（俯视图）	16
1.4 78K0S/Kx1+ 产品系列	17
1.5 框图.....	18
1.6 功能概要	19
第二章 引脚功能	20
2.1 引脚功能列表.....	20
2.2 引脚功能	22
2.2.1 P20 ~ P23（端口 2）	22
2.2.2 P32 和 P34（端口 3）	23
2.2.3 P40 ~ P47（端口 4）	23
2.2.4 RESET	23
2.2.5 X1 和 X2.....	23
2.2.6 VDD	23
2.2.7 VSS.....	23
2.3 引脚 I/O 电路和未使用引脚的连接	24
第三章 CPU 架构	26
3.1 存储器空间	26
3.1.1 内部程序存储器空间.....	29
3.1.2 内部数据存储器空间.....	30
3.1.3 特殊功能寄存器（SFR）区域.....	30
3.1.4 数据存储器寻址.....	30
3.2 处理器寄存器	33
3.2.1 控制寄存器	33
3.2.2 通用寄存器	36
3.2.3 特殊功能寄存器（SFR）	37
3.3 指令地址寻址	41
3.3.1 相对寻址.....	41
3.3.2 立即寻址.....	42
3.3.3 表间接寻址	42
3.3.4 寄存器寻址	43
3.4 操作数地址寻址.....	44
3.4.1 直接寻址.....	44
3.4.2 短直接寻址	45
3.4.3 特殊功能寄存器（SFR）寻址.....	46
3.4.4 寄存器寻址	47
3.4.5 寄存器间接寻址	48

3.4.6	基址寻址	49
3.4.7	堆栈寻址	50
第四章	端口功能	51
4.1	端口的功能	51
4.2	端口配置	52
4.2.1	端口 2	52
4.2.2	端口 3	56
4.2.3	端口 4	57
4.3	控制端口功能的寄存器	58
4.4	端口功能的操作	62
4.4.1	写入 I/O 端口	62
4.4.2	从 I/O 端口中读取	62
4.4.3	I/O 端口上的操作	62
第五章	时钟发生器	63
5.1	时钟发生器的功能	63
5.1.1	系统时钟振荡器	63
5.1.2	用于间隔时间产生的时钟振荡器	63
5.2	时钟发生器的配置	64
5.3	控制时钟发生器的寄存器	66
5.4	系统时钟振荡器	69
5.4.1	高速内部振荡器	69
5.4.2	晶体 / 陶瓷谐振器	69
5.4.3	外部时钟输入电路	71
5.4.4	预分频器	71
5.5	CPU 时钟发生器的操作	72
5.6	为周边硬件提供时钟的时钟发生器的操作	78
第六章	16 位定时器 / 事件计数器 00	80
6.1	16 位定时器 / 事件计数器 00 的功能	80
6.2	16 位定时器 / 事件计数器 00 的配置	81
6.3	控制 16 位定时器 / 事件计数器 00 的寄存器	85
6.4	16 位定时器 / 事件计数器 00 的操作	91
6.4.1	间隔定时器操作	91
6.4.2	外部事件计数器操作	93
6.4.3	脉冲宽度测量操作	96
6.4.4	方波输出操作	104
6.4.5	PPG 输出操作	106
6.4.6	单脉冲输出操作	109
6.5	16 位定时器 / 事件计数器 00 相关的注意事项	114
第七章	8 位定时器 H1	121
7.1	8 位定时器 H1 的功能	121

7.2	8 位定时器 H1 的配置	121
7.3	控制 8 位定时器 H1 的寄存器.....	124
7.4	8 位定时器 H1 的操作	126
7.4.1	作为间隔定时器 / 方波输出工作	126
7.4.2	作为 PWM 输出模式操作	129
第八章	看门狗定时器.....	135
8.1	看门狗定时器的功能	135
8.2	看门狗定时器的配置	137
8.3	控制看门狗定时器的寄存器	138
8.4	看门狗定时器的操作	140
8.4.1	当“低速内部振荡器可由软件停止”由选项字节选择时的看门狗定时器操作	140
8.4.2	当“低速内部振荡器可由软件停止”由选项字节选择时的看门狗定时器操作	142
8.4.3	STOP 模式下看门狗定时器的操作（当“低速内部振荡器可由软件停止”由选项字节选择时）	144
8.4.4	HALT 模式下看门狗定时器的操作（当“低速内部振荡器可由软件停止”由选项字节选择时）	145
第九章	中断功能	146
9.1	中断功能类型	146
9.2	中断源和配置	146
9.3	中断功能控制寄存器	148
9.4	中断服务操作	151
9.4.1	可屏蔽中断请求响应操作.....	151
9.4.2	多重中断服务.....	153
9.4.3	中断请求未决.....	155
第十章	待机功能	156
10.1	待机功能和配置.....	156
10.1.1	待机功能.....	156
10.1.2	待机期间使用的检测器	158
10.2	待机功能操作	159
10.2.1	HALT 模式.....	159
10.2.2	STOP 模式	162
第十一章	复位功能.....	166
11.1	确定复位源的寄存器	173
第十二章	上电清零电路	174
12.1	上电清零电路的功能	174
12.2	上电清零电路的配置	175
12.3	上电清零电路的操作	175
12.4	上电清零电路的注意事项	176
第十三章	低电压检测电路.....	178
13.1	低电压检测电路的功能	178

13.2 低电压检测电路的配置.....	178
13.3 控制低电压检测电路的寄存器.....	179
13.4 低电压检测电路的操作.....	181
13.5 低电压检测电路的注意事项	185
第十四章 选项字节.....	188
14.1 选项字节的功能.....	188
14.2 选项字节的格式.....	189
14.3 当 RESET 引脚用作输入端口引脚 (P34) 时的注意事项.....	190
第十五章 FLASH 存储器	191
15.1 特征.....	191
15.2 存储器配置.....	192
15.3 功能概要	192
15.4 用 Flash 存储器编程器写入.....	193
15.5 编程环境.....	194
15.6 板上引脚的处理.....	196
15.6.1 X1 和 X2 引脚	196
15.6.2 RESET 引脚	197
15.6.3 端口引脚.....	198
15.6.4 电源供电	198
15.7 板上和板外 Flash 存储器编程	199
15.7.1 Flash 存储器编程模式.....	199
15.7.2 通信命令	199
15.7.3 安全设置.....	200
15.8 通过自写的 Flash 存储器编程	201
15.8.1 自编程概要	201
15.8.2 自编程功能注意事项	204
15.8.3 用于自编程功能的寄存器	204
15.8.4 切换正常模式到自编程模式举例	211
15.8.5 切换自编程模式到正常模式举例	214
15.8.6 自编程模式下 block 擦除操作示例	217
15.8.7 自编程模式下 block 空白检查操作示例	220
15.8.8 自编程模式下字节写操作示例.....	223
15.8.9 自编程模式下内部校验操作示例	226
15.8.10 自编程模式下命令执行时间最小化时操作举例.....	230
15.8.11 自编程模式下中断禁止时间最小化操作举例	237
第十六章 指令集概述	248
16.1 操作.....	248
16.1.1 操作数标识符和描述方法	248
16.1.2 “操作栏”的说明	249
16.1.3 “标志”栏的说明	249
16.2 操作列表.....	250
16.3 按寻址类型列出指令	255

第十七章 电气特性 (目标)	258
第十八章 封装图	269
第十九章 推荐焊接条件	270
附录 A 开发工具	271
A.1 软件包	276
A.2 语言处理软件	276
A.3 控制软件	277
A.4 Flash 存储器写入工具	277
A.5 调试工具 (硬件)	278
A.5.1 当使用在线仿真器 QB-78K0SKX1	278
A.5.2 当使用在线仿真器 QB-MINI2	278
A.5.3 当使用在线仿真器 IE-78K0S-NS or IE-78K0S-NS-A	279
A.5.4 当使用在线仿真器 QB-78K0SKX1MINI	279
A.6 调试工具 (软件)	280
附录 B 目标系统设计的注意事项	281
附录 C 寄存器索引	284
C.1 寄存器索引 (寄存器名称)	284
C.2 寄存器索引 (符号)	286
附录 D 注意事项列表	288

第一章 概述

1.1 特征

○ 78K0S CPU 核

○ ROM 和 RAM 容量

产品代号 \ 项目	程序存储器 (Flash 存储器)	存储器 (内部高速 RAM)
μ PD78F9510	1 KB	128 字节
μ PD78F9511	2 KB	
μ PD78F9512	4 KB	

○ 最小指令执行时间: 0.2 μ s (以 10 MHz@4.0 ~ 5.5 V 操作)

○ 时钟

• 高速系统时钟 ... 从以下两个源中选择

- 陶瓷 / 晶体谐振器: 1 ~ 10 MHz

- 高速内部振荡器: 8 MHz \pm 3% (-10 ~ +70°C), 8 MHz \pm 5% (-40 ~ +85°C)

• 低速内部振荡器 240 kHz (典型值) ... 看门狗定时器, 定时器时钟间歇操作

○ I/O 端口: 14 (CMOS I/O: 13, CMOS 输入: 1)

○ 定时器: 3 通道

• 16 位定时器 / 事件计数器: 1 通道 ... 定时器输出 \times 1, 捕获输入 \times 2

• 8 位定时器: 1 通道 ... PWM 输出 \times 1

• 看门狗定时器: 1 通道 ... 可以低速内部振荡时钟操作

○ 片上上电清零 (POC) 电路 (当电压下降到 2.1 V \pm 0.1 V 或更低时, 一个复位被自动产生)

○ 片上低电压检测 (LVI) 电路 (当达到检测电压时, 一个中断 / 复位 (可选) 被自动产生)

• 检测电压: 可从 2.35 和 4.3 V 之间的十个电平中选择

○ 单路供电 flash 存储器

• Flash 自编程允许

• 软件保护功能: 防止外部复制 (无 flash 读取命令)

• 通过专用 flash 存储器编程器写入需要的时间: 大约 3 秒 (4 KB)

* 支持大量产品线上的写入操作

○ 安全功能

• 独立于 CPU 的时钟操作的看门狗定时器

... 如果系统时钟停止, 挂起可以被检测

• 可由 LVI 检测的电源电压

... 在电源电压下降到操作电压以下之前, 适当的处理可以被执行

• 装备选项字节功能

... 在硬件中设置重要系统操作

○ 支持汇编和 C 语言

○ 增强的开发环境

• 支持全功能仿真器 (IECUBE), 简化仿真器 (MINICUBE2) 和仿真器

○ 电源电压: $V_{DD} = 2.0 \sim 5.5$ V

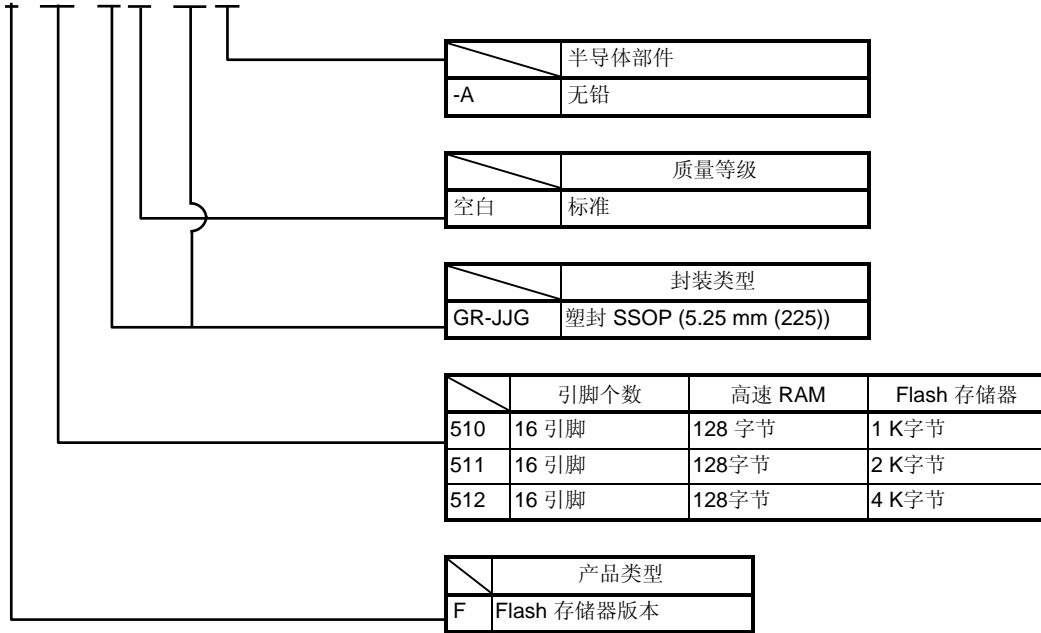
* 在 2.2 ~ 5.5 V 的电压范围内使用该产品, 因为上电清零电路 (POC) 的检测电压 (V_{POC}) 为 2.1 V \pm 0.1 V。

○ 工作温度范围: $T_A = -40 \sim +85^\circ\text{C}$

1.2 订购信息

产品代号

μ PD78F9 xxx - xx (x) - xxx -A



[产品代号列表]

μ PD78F9510GR-JJG-A^註

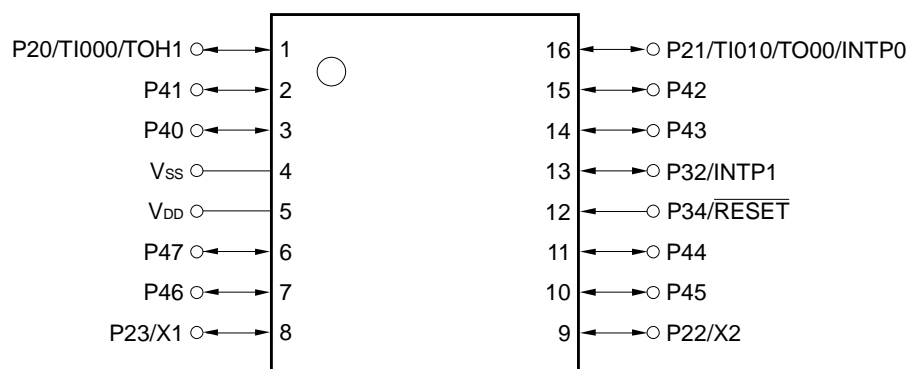
μ PD78F9511GR-JJG-A^註

μ PD78F9512GR-JJG-A^註

注 开发中

1.3 引脚配置（俯视图）

16 引脚塑封 SSOP



INTP0, INTP1: 外部中断输入

P20 ~ P23: 端口 2

P32, P34: 端口 3

P40 ~ P47: 端口 4

 $\overline{\text{RESET}}$: 复位

TI000, TI010: 定时器输入

TO00, TOH1: 定时器输出

VDD: 电源

Vss: 地

X1, X2: 晶体振荡器 (X1 输入时钟)

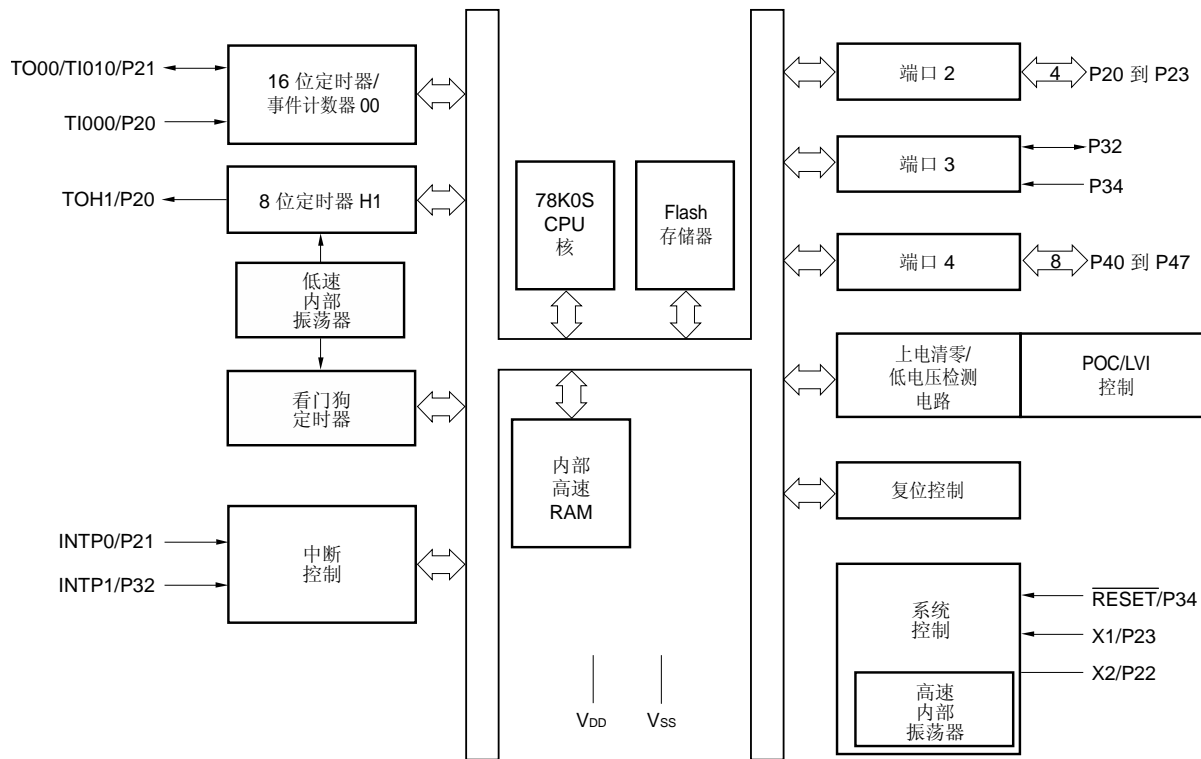
1.4 78K0S/Kx1+ 产品系列

下表显示 78K0S/Kx1+ 的产品系列。

产品代号		78K0S/KU1+	78K0S/KY1+	78K0S/KA1+	78K0S/KB1+
项目		10 引脚	16 引脚	20 引脚	30 引脚
内部存储器	Flash 存储器	1 KB, 2 KB, 4 KB		2 KB	4 KB
	RAM	128 字节		128 字节	256 字节
电源电压		$V_{DD} = 2.0 \sim 5.5 \text{ V}^{\#1}$			
最小指令执行时间		0.20 μs (10 MHz, $V_{DD} = 4.0 \sim 5.5 \text{ V}$) 0.33 μs (6 MHz, $V_{DD} = 3.0 \sim 5.5 \text{ V}$) 0.40 μs (5 MHz, $V_{DD} = 2.7 \sim 5.5 \text{ V}$) 1.0 μs (2 MHz, $V_{DD} = 2.0 \sim 5.5 \text{ V}$)			
系统时钟 (振荡频率)		高速内部振荡 (8 MHz (典型值)) 晶体 / 陶瓷振荡 (1 ~ 10 MHz) ^{#4} 外部时钟输入振荡 (1 ~ 10 MHz)			
TMH1 和 WDT 的时钟 (振荡频率)		低速内部振荡 (240 kHz (典型值))			
端口	CMOS I/O	7	13	15	24
	CMOS 输入	1	1	1	1
	CMOS 输出	-	-	1	1
定时器	16 位 (TM0)	1 通道 ^{#4}			
	8 位 (TMH)	1 通道			
	8 位 (TM8)	-		1 通道	
	WDT	1 通道			
串行接口		-	支持 LIN-Bus 的 UART: 1 通道		
A/D 转换器 ^{#2}		10 位: 4 通道 (2.7 ~ 5.5 V) ^{#2}			
乘法器 (8 位 × 8 位)		-			提供
中断	外部	2		4	
	内部	5 ^{#3}		9	
复位	RESET 引脚	提供			
	POC	2.1 V ±0.1 V			
	LVI	提供 (可由软件选择)			
	WDT	提供			
工作温度范围		标准产品: -40 ~ +85°C	标准产品, (A) 等级产品: -40 ~ +85°C (A2) 等级产品: -40 ~ +125°C		

- 注 1. 在 2.2 ~ 5.5 V 的电压范围内使用该产品, 因为上电清零电路 (POC) 的检测电压 (V_{POC}) 为 2.1 V ±0.1 V。
2. 对 78K0S/KU1+ (开发中) 和 78K0S/KY1+ 分别提供不含 A/D 转换器 ($\mu\text{PD78F95xx}$) 的产品。该产品不含 A/D 转换器。
3. 不含 A/D 转换器的产品为 4
4. 不支持 78K0S/KU1+ (开发中) 中的不含 A/D 转换器 ($\mu\text{PD78F950x}$) 的产品。

1.5 框图



1.6 功能概要

项目		μ PD78F9510	μ PD78F9511	μ PD78F9512
内部存储器	Flash 存储器	1 KB	2 KB	4 KB
	高速 RAM	128 字节		
存储器空间		64 KB		
X1 输入时钟 (振荡频率)		晶体 / 陶瓷 / 外部时钟输入: 10 MHz ($V_{DD} = 2.0 \sim 5.5$ V)		
内部振荡时钟	高速 (振荡频率)	内部振荡: 8 MHz (典型值)		
	低速 (用于 TMH1 和 WDT)	内部振荡: 240 kHz (典型值)		
通用寄存器		8 位 \times 8 个寄存器		
指令执行时间		0.2 μ s / 0.4 μ s / 0.8 μ s / 1.6 μ s / 3.2 μ s (X1 输入时钟: $f_x = 10$ MHz)		
I/O 端口		总数: 14 引脚 CMOS I/O: 13 引脚 CMOS 输入: 1 引脚		
定时器		<ul style="list-style-type: none"> • 16 位定时器 / 事件计数器: 1 通道 • 8 位定时器 (定时器 H1): 1 通道 • 看门狗定时器: 1 通道 		
	定时器输出	2 引脚 (PWM: 1 引脚)		
向量中断源	外部	2		
	内部	4		
复位		<ul style="list-style-type: none"> • RESET 引脚产生的复位 • 看门狗定时器产生的内部复位 • 上电清零电路产生的内部复位 • 低电压检测电路产生的内部复位 		
电源电压		$V_{DD} = 2.0 \sim 5.5$ V ^注		
工作温度范围		-40 \sim +85°C		
封装		16 引脚塑封 SSOP		

注 在 2.2 \sim 5.5 V 的电压范围内使用该产品, 因为上电清零电路 (POC) 的检测电压 (V_{POC}) 为 2.1 V \pm 0.1 V。

第二章 引脚功能

2.1 引脚功能列表

(1) 端口引脚

引脚名称	I/O	功能		复位后	复用功能引脚
P20	I/O	端口 2。 4 位 I/O 端口。 可以按位设置为输入或输出模式。 片上上拉电阻可以通过软件连接。		输入	TI000/TOH1
P21					TI010/TO00/ INTP0
P22 ^注					X2 ^注
P23 ^注					X1 ^注
P32	I/O	端口 3	可以按位设置为输入或输出模式。 片上上拉电阻可以通过软件连接。	输入	INTP1
P34 ^注					
P40 ~ P47	I/O	端口 4。 8 位 I/O 端口。 可以按位设置为输入或输出模式。 片上上拉电阻可以通过软件连接。		输入	—

注 关于引脚功能的设置方法，参见第十四章 选项字节。

注意事项 P22/X2 和 P23/X1 引脚在复位期间下拉。

(2) 非端口引脚

引脚名称	I/O	功能	复位后	复用功能引脚
INTP0	输入	可以为外部中断输入指定有效沿（上升沿，下降沿，或双边沿）	输入	P21 /TI010/ TO00
INTP1				P32
TI000	输入	外部计数时钟输入到 16 位定时器 / 事件计数器 00。 捕获触发信号输入到 16 位定时器 / 事件计数器 00 的捕获寄存器 (CR000 和 CR010) 中	输入	P20 /TOH1
TI010		捕获触发信号输入到 16 位定时器 / 事件计数器 00 的捕获寄存器 (CR000) 中		P21/TO00/ INTP0
TO00	输出	16 位定时器 / 事件计数器 00 输出	输入	P21/TI010/ INTP0
TOH1	输出	8 位定时器 H1 输出	输入	P20/TI000
RESET [‡]	输入	系统复位输入	输入	P34 [‡]
X1 [‡]	输入	用于系统时钟振荡的晶体 / 陶瓷振荡器的连接 外部时钟输入	-	P23 [‡]
X2 [‡]	-	用于系统时钟振荡的晶体 / 陶瓷振荡器的连接	-	P22 [‡]
V _{DD}	-	正电源	-	-
V _{SS}	-	地电平	-	-

注 关于引脚功能的设置方法，参见第十四章 选项字节。

注意事项 P22/X2 和 P23/X1 引脚在复位期间下拉。

2.2 引脚功能

2.2.1 P20 ~ P23 (端口 2)

P20 ~ P23 组成 4 位 I/O 端口。除用作 I/O 端口引脚以外，这些引脚也具有输入 / 输出定时器信号和输入外部中断请求信号的功能。

P22 和 P23 也分别用作 X2 和 X1。关于引脚功能的设置方法，参见第十四章 选项字节。

这些引脚可以以 1 位为单位设置为以下操作模式。

(1) 端口模式

P20 ~ P23 用作 4 位 I/O 端口。通过使用端口模式寄存器 2 (PM2)，该端口的每一位可以被设置为输入或输出模式。此外，通过使用上拉电阻选项寄存器 2 (PU2)，片上上拉电阻可以被连接到端口。

(2) 控制模式

P20 ~ P23 用于输入 / 输出定时器信号和输入外部中断请求信号。

(a) TI000

该引脚将外部计数时钟输入到 16 位定时器 / 事件计数器 00 中，或者将捕获触发信号输入到 16 位定时器 / 事件计数器 00 的捕获寄存器 (CR000 和 CR010) 中。

(b) TI010

该引脚将捕获触发信号输入到 16 位定时器 / 事件计数器 00 的捕获寄存器 (CR000) 中。

(c) TO00

该引脚从 16 位定时器 / 事件计数器 00 输出一个信号。

(d) TOH1

该引脚从 8 位定时器 H1 输出一个信号。

(e) INTP0

这是可以指定有效沿 (上升沿, 下降沿, 或双边沿) 的外部中断请求输入引脚。

注意事项 P22/X2 和 P23/X1 引脚在复位期间下拉。

2.2.2 P32 和 P34 (端口 3)

P32 是 1 位 I/O 端口。除用作 I/O 端口引脚以外，该引脚也用于输入外部中断请求信号。

P34 是 1 位输入端口。该引脚也用作 RESET 引脚，并且当 $\overline{\text{P34}}$ 上电时，这是复位功能。

关于引脚功能的设置方法，参见第十四章 选项字节。

当 P34 用作输入端口引脚时，连接上拉电阻。

P32 和 P34 可以按位设置为以下操作模式。

(1) 端口模式

P32 用作 1 位 I/O 端口。通过使用端口模式寄存器 3 (PM3)，该引脚可以被设置为输入或输出模式。此外，通过使用上拉电阻选项寄存器 3 (PU3)，片上上拉电阻可以被连接到端口。

P34 仅用作 1 位输入端口。

(2) 控制模式

P32 用作可以指定有效沿（上升沿，下降沿，或双边沿）的外部中断请求输入引脚 (INTP1)。

2.2.3 P40 ~ P47 (端口 4)

P40 ~ P47 组成 8 位 I/O 端口。通过使用端口模式寄存器 4 (PM4)，该端口的每一位可以被设置为输入或输出模式。此外，通过使用上拉电阻选项寄存器 4 (PU4)，片上上拉电阻可以被连接到端口。

2.2.4 $\overline{\text{RESET}}$

该引脚输入一个低有效的系统复位信号。当上电时，无论选项字节的设置是什么，这都是复位功能。

2.2.5 X1 和 X2

这些引脚连接一个振荡器来振荡 X1 输入时钟。

X1 和 X2 也分别用作 P23 和 P22。关于引脚功能的设置方法，参见第十四章 选项字节。

提供一个外部时钟到 X1。

注意事项 P22/X2 和 P23/X1 引脚在复位期间下拉。

2.2.6 V_{DD}

这是正电源引脚。

2.2.7 V_{SS}

这是地引脚。

确保连接 V_{SS} 到稳定的地 (= 0 V)。

2.3 引脚I/O电路和未使用引脚的连接

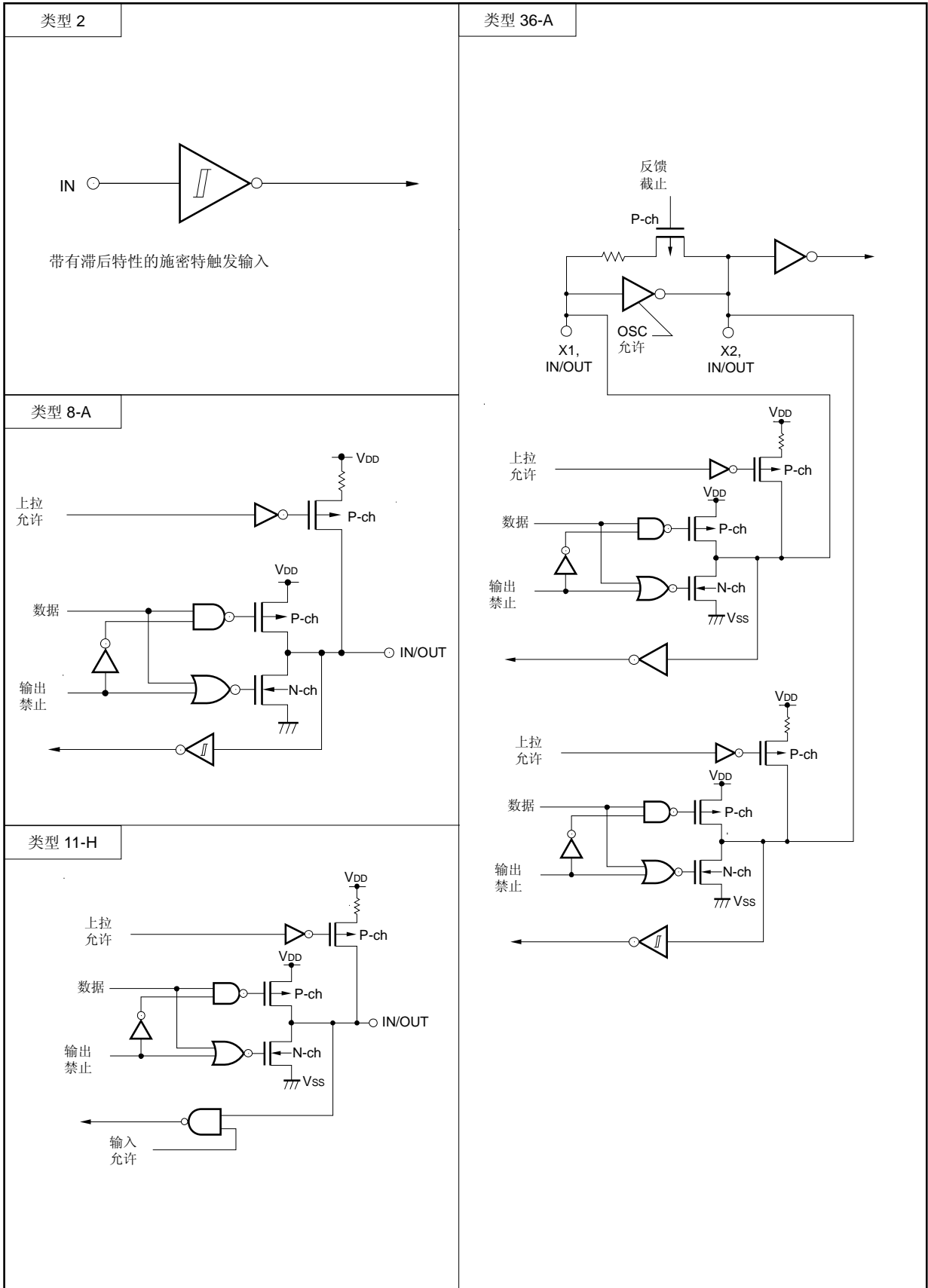
表 2-1 表示每个引脚的 I/O 电路类型和未使用引脚的连接。

关于各个类型的 I/O 电路和配置，参见图 2-1。

表 2-1. 引脚 I/O 电路类型和未使用引脚的连接

引脚名称	I/O 电路类型	I/O	未使用引脚的推荐连接
P20/TI000/TOH1	11-H	I/O	输入： 通过电阻单独连接到 V_{DD} 或 V_{SS} 。
P21/TI010/TO00/INTP0			输出： 保持开路。
P22/X2	36-A		输入： 通过电阻单独连接到 V_{SS} 。
P23/X1			输出： 保持开路。
P32/INTP1	8-A		输入： 通过电阻单独连接到 V_{DD} 或 V_{SS} 。
		输出： 保持开路。	
P34/ $\overline{\text{RESET}}$	2	输入	通过电阻连接到 V_{DD} 。
P40 ~ P47	8-A	I/O	输入： 通过电阻单独连接到 V_{DD} 或 V_{SS} 。
			输出： 保持开路。

图 2-1. 引脚 I/O 电路

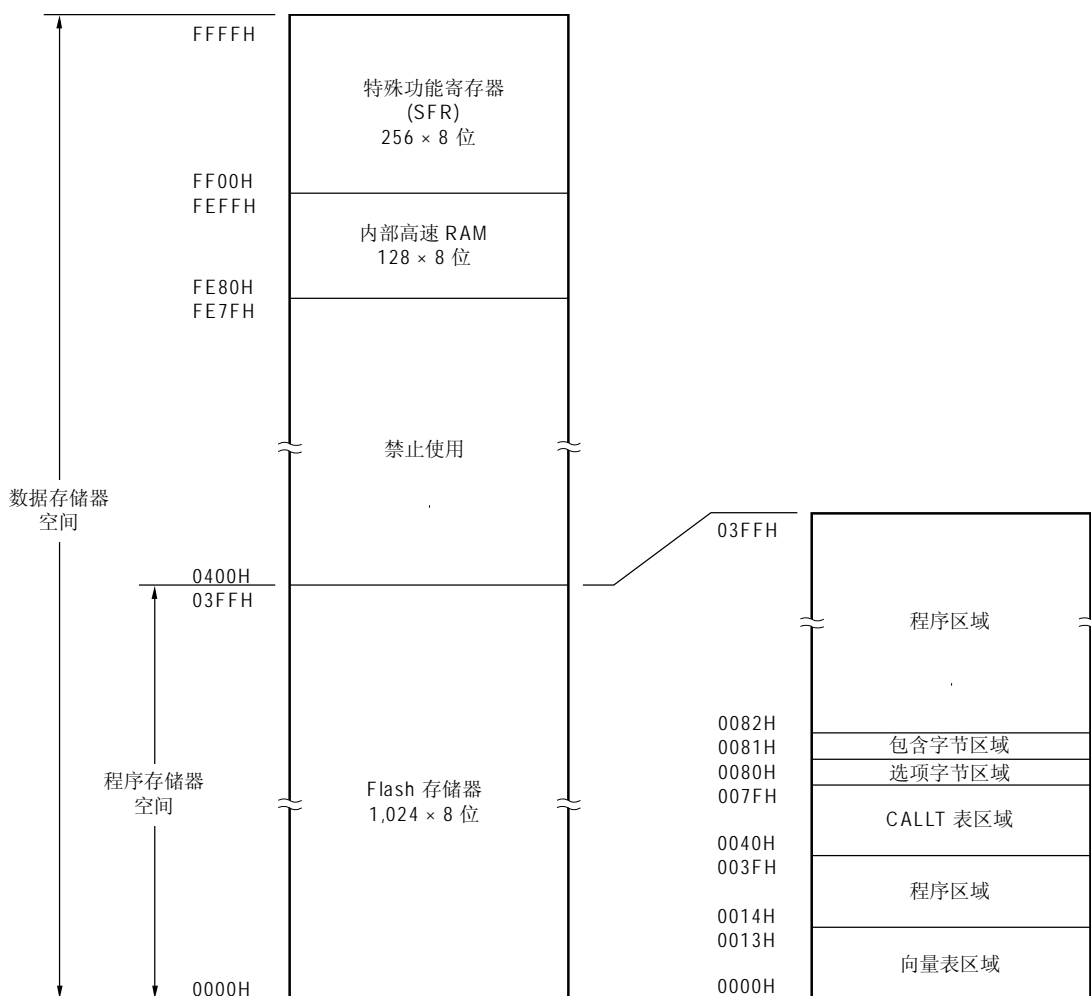


第三章 CPU架构

3.1 存储器空间

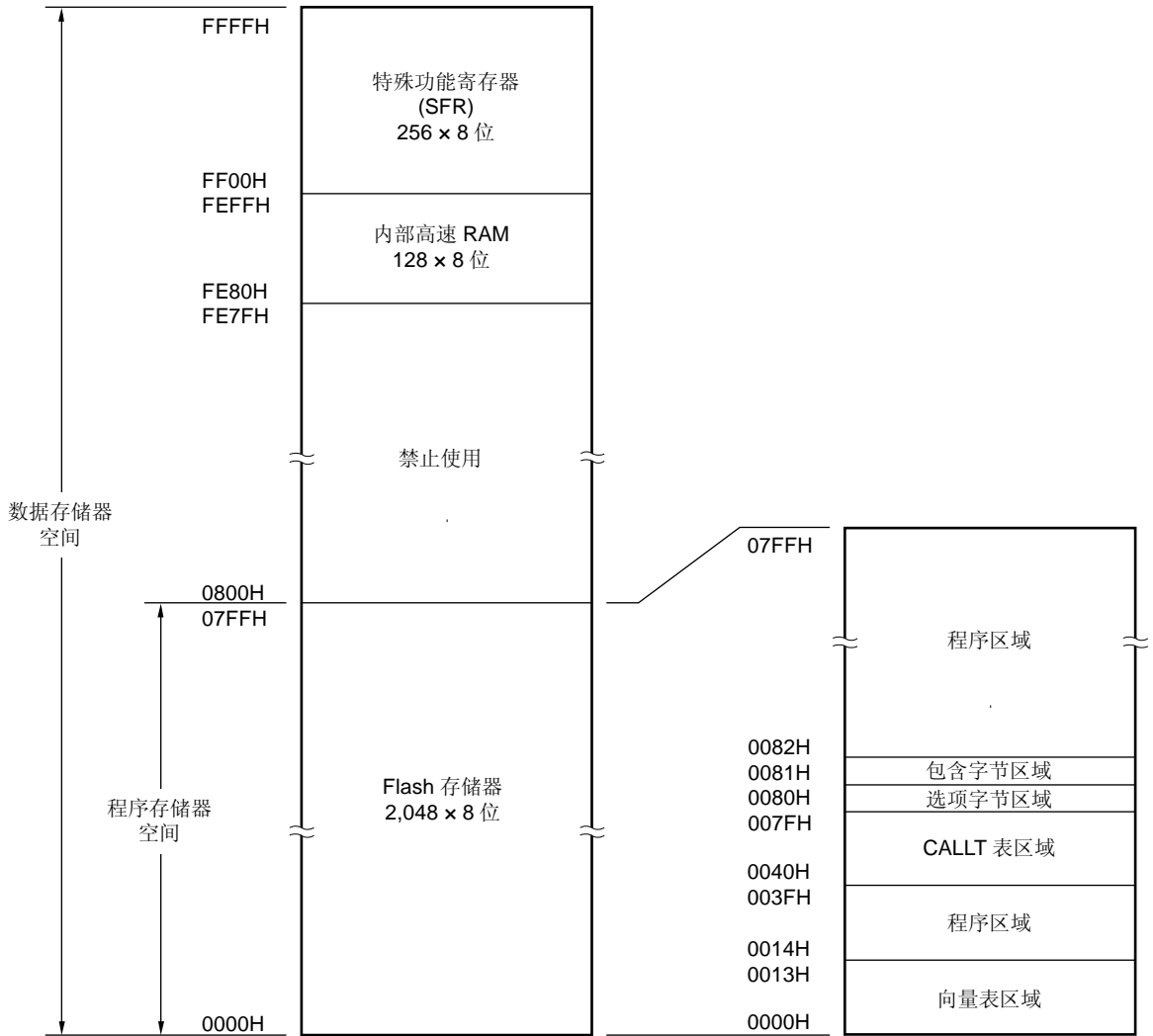
μPD78F9510、78F9511、78F9512 可以访问 64 KB 的存储器空间。图 3-1 到 3-3 显示了存储器映射图。

图 3-1. 存储器映射 (μPD78F9510)



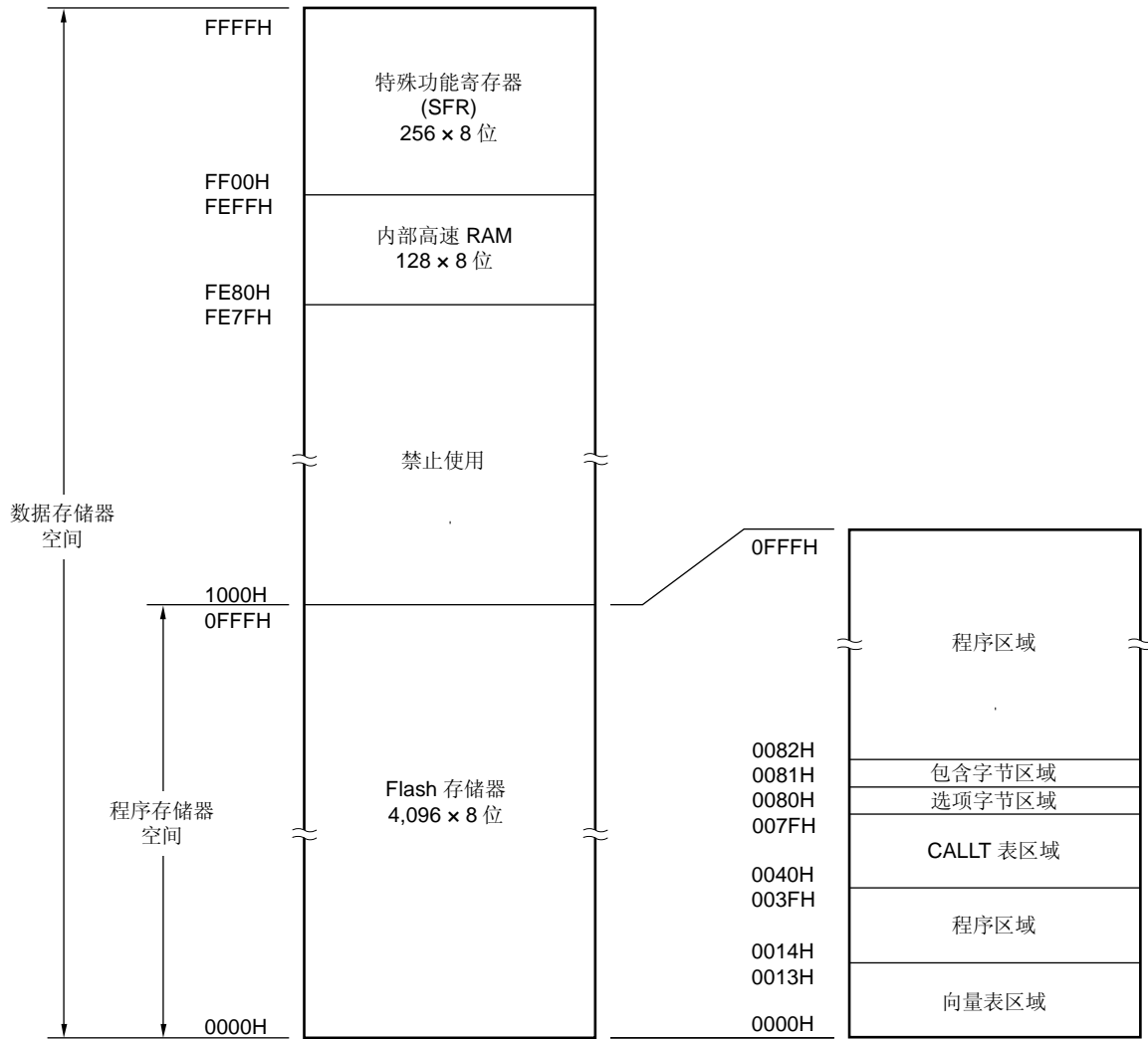
备注 选项字节和保护字节每个为 1 字节。

图 3-2. 存储器映射 (μ PD78F9511)



备注 选项字节和保护字节每个为 1 字节。

图 3-3. 存储器映射 (μ PD78F9512)



备注 选项字节和保护字节每个为 1 字节。

3.1.1 内部程序存储器空间

内部程序存储器空间用于保存程序以及表格数据。该空间通常使用程序计数器（PC）进行寻址。 μ PD78F9510、78F9511、78F9512 提供包含以下容量的内部 ROM（或 flash 存储器）。

表 3-1. 内部 ROM 容量

产品代号	内部 ROM	
	结构	容量
μ PD78F9510	Flash 存储器	1,024 × 8 位
μ PD78F9511		2,048 × 8 位
μ PD78F9512		4,096 × 8 位

以下区域被分配到内部程序存储器空间。

(1) 向量表区域

地址为 0000H 到 0013H 的 20 字节的区域被保留为向量表区域。该区域保存由 RESET 或中断请求产生的跳转时使用的开始地址。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	中断请求	向量表地址	中断请求
0000H	复位	000CH	INTTMH1
0006H	INTLVI	000EH	INTTM000
0008H	INTP0	0010H	INTTM010
000AH	INTP1		

(2) CALLT 指令表区域

1 字节调用指令（CALLT）的子程序入口地址可以保存在地址为 0040H 到 007FH 的 64 字节区域中。

(3) 选项字节区域

选项字节区域是地址为 0080H 的 1 字节区域。关于详细情况，参阅第十四章 选项字节。

(4) 保护字节区域

保护字节区域是地址为 0081H 的 1 字节区域。关于详细情况，参阅第十五章 Flash 存储器。

3.1.2 内部数据存储器空间

μ PD78F9510、78F9511、78F9512 提供 128 字节的内部高速 RAM。

内部高速 RAM 也可以用作栈存储器。

3.1.3 特殊功能寄存器 (SFR) 区域

片上周边硬件的特殊功能寄存器 (SFR) 被分配到地址为 FF00H 到 FFFFH 的区域 (参见表 3-3)。

3.1.4 数据存储器寻址

μ PD78F9510、78F9511、78F9512 提供大范围的寻址模式来使存储器操作尽量高效。包含数据存储器的区域 (FE80H 到 FFFFH) 和特殊功能寄存器 (SFR) 区域可以根据各自的功能使用独特的寻址模式来访问。图 3-4 到 3-6 显示了数据存储器寻址。

图 3-4. 数据存储器寻址 (μ PD78F9510)

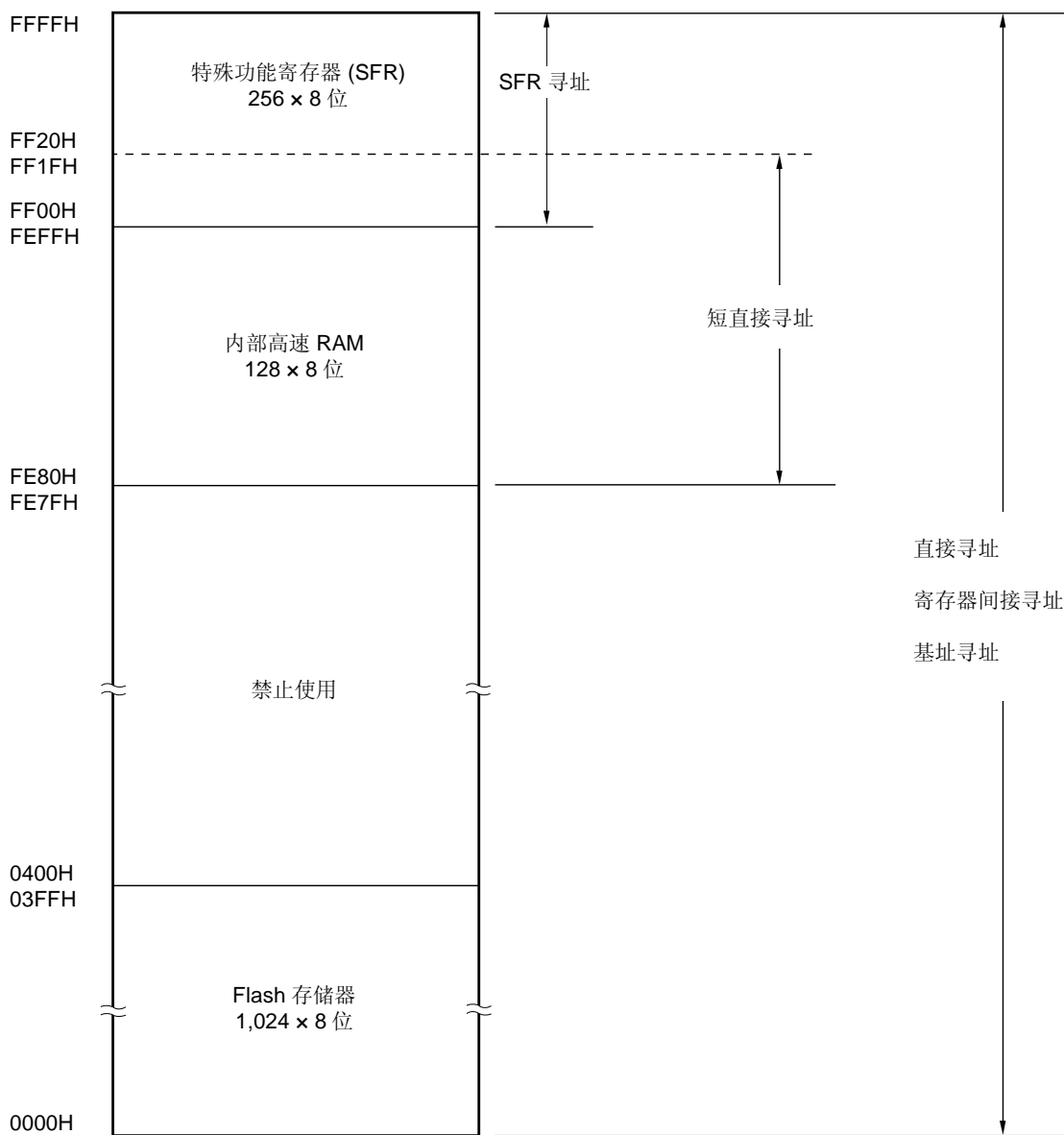


图 3-5. 数据存储地址 (μPD78F9511)

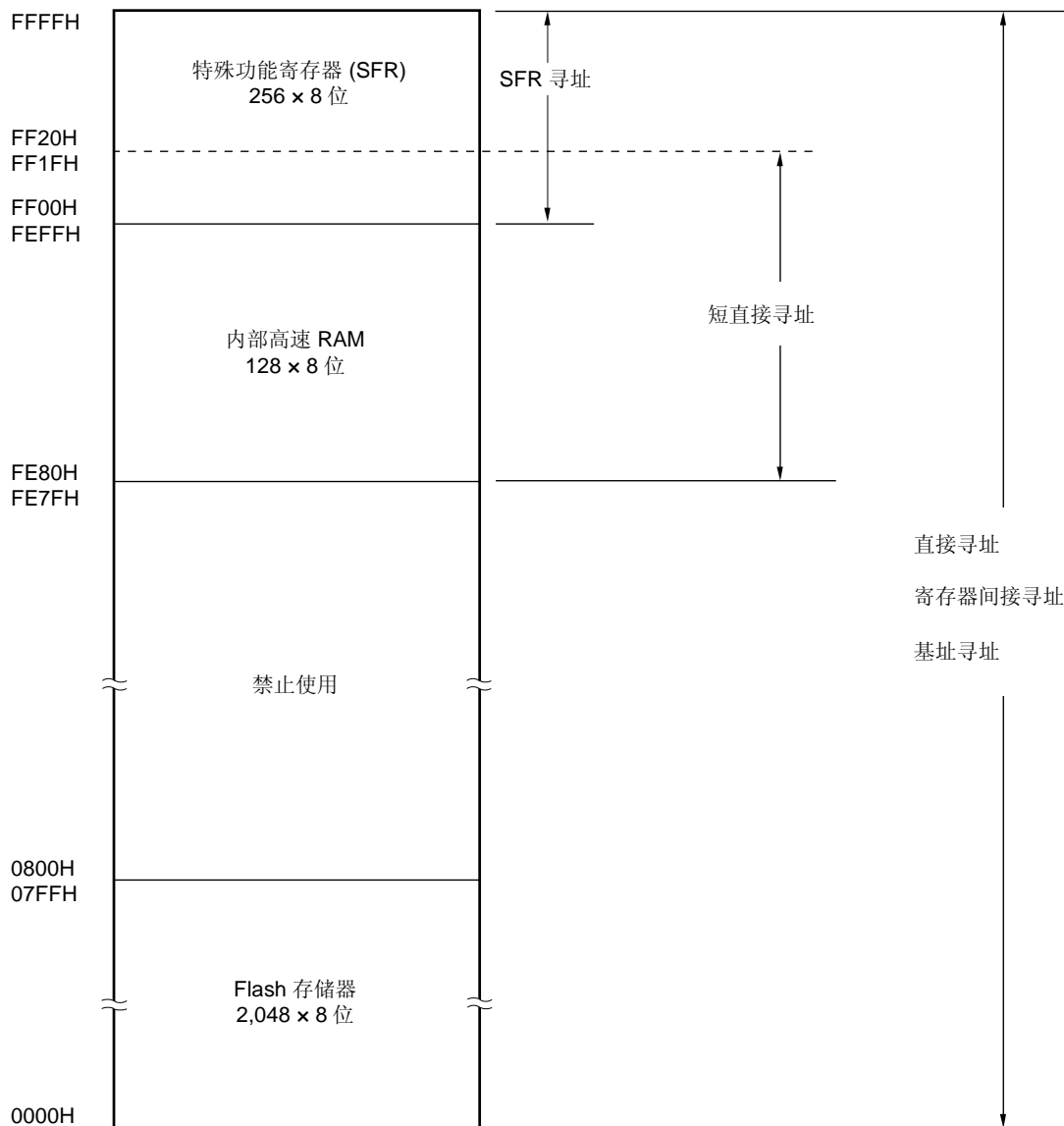
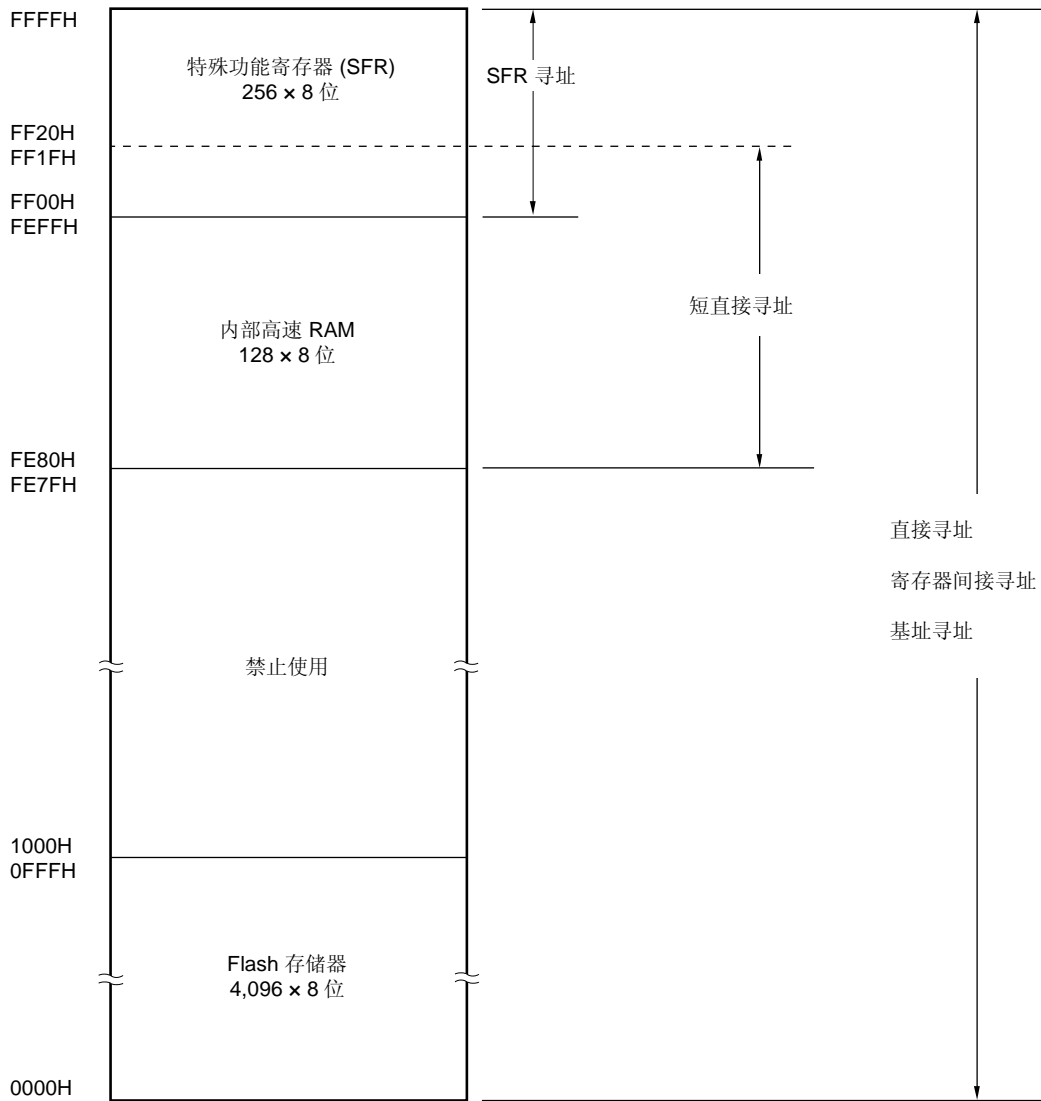


图 3-6. 数据存储器寻址 (μ PD78F9512)



3.2 处理器寄存器

μ PD78F9510、78F9511、78F9512 提供以下处理器寄存器。

3.2.1 控制寄存器

控制寄存器具有控制程序顺序状态以及堆栈存储器的特殊功能。控制寄存器包含一个程序计数器，一个程序状态字以及一个堆栈指针。

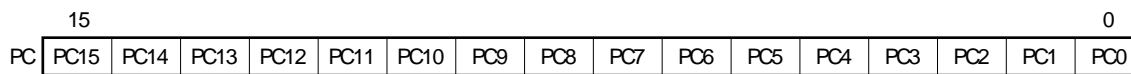
(1) 程序计数器 (PC)

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，则要对立即数或寄存器内容进行设置。

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

图 3-7. 程序计数器的配置



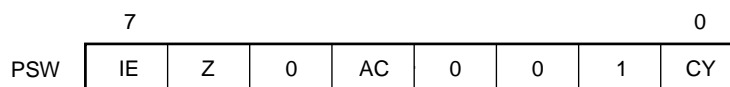
(2) 程序状态字 (PSW)

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置位 / 复位。

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

复位信号生成将 PSW 设为 02H。

图 3-8. 程序状态字的配置



(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 $IE = 0$ 时，中断禁止 (DI) 状态被设置。所有中断请求被禁止。

当 $IE = 1$ 时，中断允许 (EI) 状态被设置。对各种中断源，中断请求响应使用中断屏蔽标志来控制。

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

(b) 零标志 (Z)

当操作结果为零时，该标志被设置为 1。在所有其它情况下，它被复位为 0。

(c) 辅助进位标志 (AC)

如果操作结果中有一个来自第 3 位的进位或在第 3 位上有错位，那么该标志将会被设为 1。在所有其它情况下，它被复位为 0。

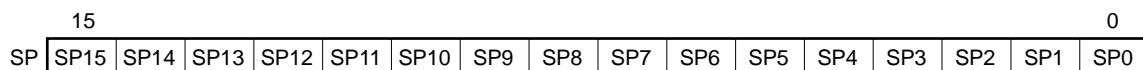
(d) 进位标志 (CY)

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

(3) 堆栈指针 (SP)

这是一个 16 位寄存器，用于保存存储器堆栈区域的开始地址。只有内部高速 RAM 区域可以被设置为堆栈区域（内部高速 RAM 以外的区域不能被设置为堆栈区域）。

图 3-9. 堆栈指针的配置



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

各个堆栈保存 / 恢复数据的操作如图 3-10 和 3-11 所示。

- 注意事项**
1. 由于复位信号生成会使用 SP 内容变为未定义，因此在使用堆栈存储器前请务必初始化 SP。
 2. 堆栈指针只能被设置为高速 RAM 区域，并且只有低 10 位可以实际被设置。
因此，如果堆栈指针被指定为 0FF00H，它将被转换为高速 RAM 区域中的 0FB00H，因为 0FF00H 处于 SFR 区域中而不在高速 RAM 区域中。
当数值被实际推入堆栈时，0FB00H 减去 1 变为 0FAFFH，但是由于该值不在高速 RAM 区域中，它将被转换为 0FEFFH，这与 0FF00H 被设置为堆栈指针时的值相同。

图 3-10. 保存在堆栈存储器中的数据

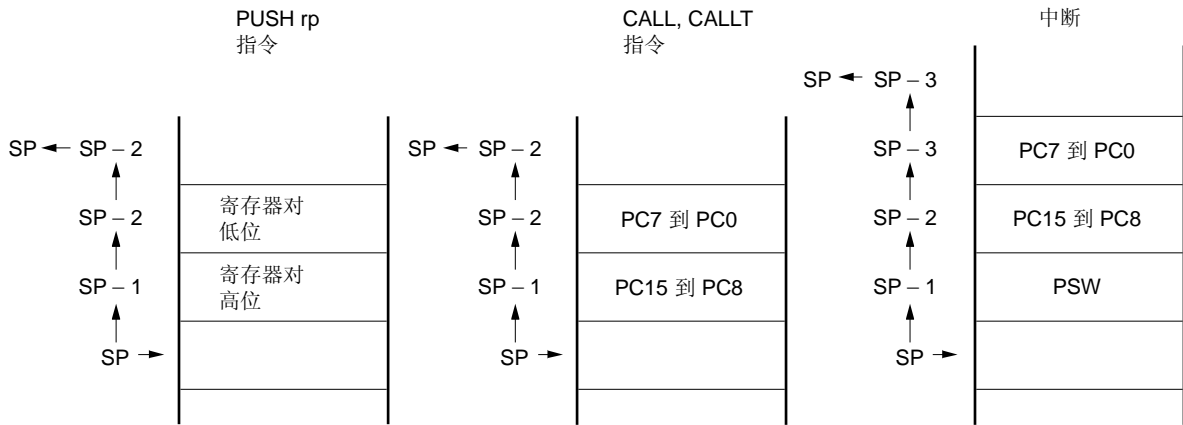
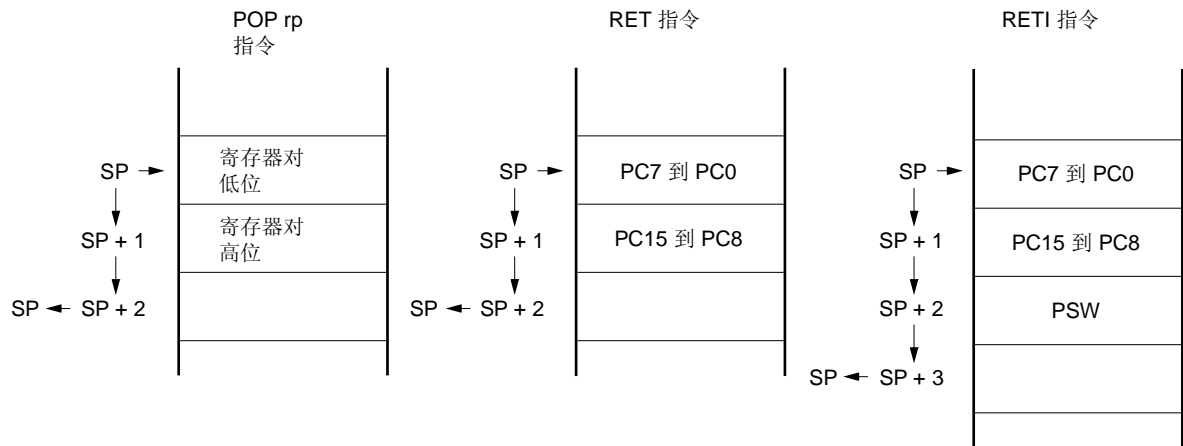


图 3-11. 从堆栈存储器中恢复的数据



3.2.2 通用寄存器

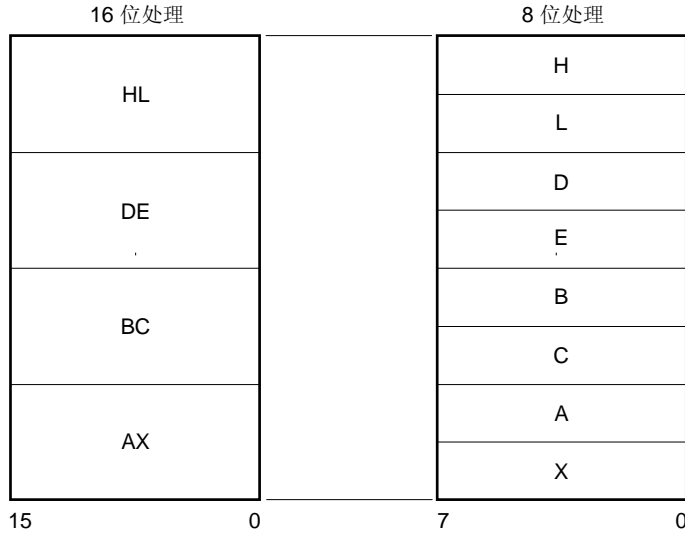
通用寄存器由八个 8 位寄存器 (X, A, C, B, E, D, L 和 H) 组成。

除每个寄存器都可以用作一个 8 位寄存器以外, 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。

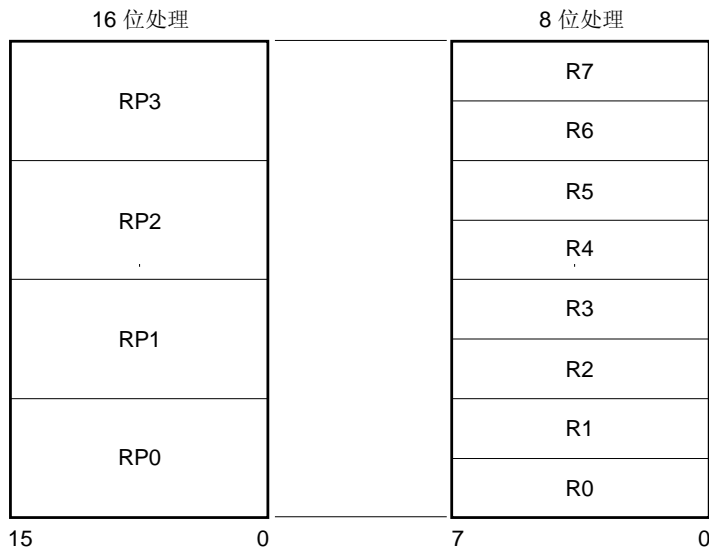
这些寄存器可以使用功能名称 (X, A, C, B, E, D, L, H, AX, BC, DE 和 HL) 和绝对名称 (R0 到 R7 以及 RP0 到 RP3) 来进行描述。

图 3-12. 通用寄存器的配置

(a) 功能名称



(b) 绝对名称



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。

特殊功能寄存器被分配到 FF00H 到 FFFFH 的 256 字节区域中。

通过使用运算、传送以及位操作指令，特殊功能寄存器可以像通用寄存器那样进行操作。根据特殊功能寄存器的类型，可操作位单元不同（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

描述 1 位操作指令操作数 (`sfr.bit`) 的汇编器保留的符号。该操作也可以通过使用一个地址和位来指定。

- 8 位操作

描述 8 位操作指令操作数 (`sfr`) 的汇编器保留的符号。该操作也可以通过使用一个地址来指令。

- 16 位操作

描述 16 位操作指令操作数的汇编器保留的符号。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。该表格中符号的含义如下所示。

- 符号

表示执行的特殊功能寄存器的地址。它是 RA18K0 中的保留字，而在 CC78K0S 中则通过使用 `#pragma sfr` 导引来定义为一个 `sfr` 变量。因此，如果汇编器或集成调试器被使用，这些符号可以用作指令操作数。

- R/W

表示特殊功能寄存器是被读取还是被写入。

R/W: 读取 / 写入

R: 只读

W: 只写

- 同时操作的位数

表示特殊功能寄存器被修改的位单元（1，8 和 16）。

- 复位后

表示复位输入时特殊功能寄存器的状态。

表 3-3. 特殊功能寄存器 (1 / 3)

地址	符号	特殊功能寄存器 (SFR) 的名称								R/W	同时操作的位数			复位后	参考页
		7	6	5	4	3	2	1	0		1	8	16		
FF00H, FF01H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF02H	P2	0	0	0	0	P23	P22	P21	P20	R/W 注 1	√	√	-	00H	60
FF03H	P3	0	0	0	P34	0	P32	0	0		√	√	-	00H	60
FF04H	P4	P47	P46	P45	P44	P43	P42	P41	P40		√	√	-	00H	60
FF05H 到 FF0DH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF0EH	CMP01	-	-	-	-	-	-	-	-	R/W	-	√	-	00H	123
FF0FH	CMP11	-	-	-	-	-	-	-	-		-	√	-	00H	123
FF10H, FF11H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF12H	TM00	-	-	-	-	-	-	-	-	R	-	-	√ 注 2	0000H	82
FF13H		-	-	-	-	-	-	-	-		-	-	-	√ 注 2	0000H
FF14H	CR000	-	-	-	-	-	-	-	-	R/W	-	-	√ 注 2	0000H	82
FF15H		-	-	-	-	-	-	-	-		-	-	-	√ 注 2	0000H
FF16H	CR010	-	-	-	-	-	-	-	-	R/W	-	-	√ 注 2	0000H	84
FF17H		-	-	-	-	-	-	-	-		-	-	-	√ 注 2	0000H
FF18H 到 FF21H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF22H	PM2	1	1	1	1	PM23	PM22	PM21	PM20	R/W	√	√	-	FFH	59
FF23H	PM3	1	1	1	1	1	PM32	1	1		√	√	-	FFH	59
FF24H	PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40		√	√	-	FFH	59
FF25H 到 FF31H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF32H	PU2	0	0	0	0	PU23	PU22	PU21	PU20	R/W	√	√	-	00H	61
FF33H	PU3	0	0	0	0	0	PU32	0	0		√	√	-	00H	61
FF34H	PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40		√	√	-	00H	61
FF35H 到 FF47H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF48H	WDTM	0	1	1	WDC S4	WDC S3	WDC S2	WDC S1	WDC S0	R/W	-	√	-	67H	138
FF49H	WDTE	-	-	-	-	-	-	-	-		-	√	-	9AH	139

- 注 1. P34 是仅输入端口。
 2. 只有通过短直接寻址, 16 位访问才可能。

表 3-3. 特殊功能寄存器 (2 / 3)

地址	符号	特殊功能寄存器 (SFR) 的名称								R/W	同时操作的位数			复位后	参考页
		7	6	5	4	3	2	1	0		1	8	16		
FF50H	LVIM	<LVI ON>	0	0	0	0	0	<LVI MD>	<LVI F>	R/W	√	√	-	00H ^{#1}	179
FF51H	LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0		-	√	-	00H ^{#1}	180
FF52H, FF53H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF54H	RESF	0	0	0	WDT RF	0	0	0	LVIRF	R	-	√	-	00H ^{#2}	173
FF55H 到 FF57H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF58H	LSRCM	0	0	0	0	0	0	0	<LSR STOP>	R/W	√	√	-	00H	67
FF59H 到 FF5FH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF60H	TMC00	0	0	0	0	TMC 003	TMC 002	TMC 001	<OVF 00>	R/W	√	√	-	00H	86
FF61H	PRM00	ES 110	ES 100	ES 010	ES 000	0	0	PRM 001	PRM 000		√	√	-	00H	89
FF62H	CRC00	0	0	0	0	0	CRC 002	CRC 001	CRC 000		√	√	-	00H	87
FF63H	TOC00	0	<OSPT 00>	<OSP E 00>	TOC 004	<LVS 00>	<LVR 00>	TOC 001	<TOE 00>		√	√	-	00H	88
FF64H 到 FF6FH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF70H	TMHMD1	<TMH E1>	CKS1 2	CKS1 1	CKS1 0	TMM D11	TMM D10	<TOLE V1>	<TOE N1>	R/W	√	√	-	00H	125
FF71H 到 FF9FH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FFA0H	PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0	W	-	√	-	不确定	206
FFA1H	PFS	0	0	0	0	0	WEP RERR	VCER R	FPRE RR	R/W	√	√	-	00H	207
FFA2H	FLPMC	0	PRSE LF4	PRSE LF3	PRSE LF2	PRSE LF1	PRSE LF0	0	FLSP M		-	√	-	不确定	205
FFA3H	FLCMD	0	0	0	0	0	FLCM D2	FLCM D1	FLCM D0		√	√	-	00H	208
FFA4H	FLAPL	FLA P7	FLA P6	FLA P5	FLA P4	FLA P3	FLA P2	FLA P1	FLA P0		√	√	-	不确定	209

注 1. 只在 LVI 产生的复位后保持。

2. 该值根据复位源而改变。

备注 对于三角括号 (<>) 中的位名称, 它是 RA78K0S 中的保留字, 而在 CC78K0S 中则通过使用 #pragma sfr 导引来定义为一个 sfr 变量。

表 3-3. 特殊功能寄存器 (3 / 3)

地址	符号	特殊功能寄存器 (SFR) 的名称								R/W	同时操作的位数			复位后	参考页
		7	6	5	4	3	2	1	0		1	8	16		
FFA5H	FLAPH	0	0	0	0	FLA P11	FLA P10	FLA P9	FLA P8	R/W	√	√	-	不确定	209
FFA6H	FLAPH C	0	0	0	0	FLAP C11	FLAP C10	FLAP C9	FLAP C8		√	√	-	00H	209
FFA7H	FLAPLC	FLAP C7	FLAP C6	FLAP C5	FLAP C4	FLAP C3	FLAP C2	FLAP C1	FLAP C0		√	√	-	00H	209
FFA8H	FLW	FLW7	FLW6	FLW5	FLW4	FLW3	FLW2	FLW1	FLW0		-	√	-	00H	210
FFA9H 到 FFDFH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FFE0H	IF0	0	<TMIF 010>	<TMIF 000>	<TMIF H1>	<PIF 1>	<PIF 0>	<LVI IF>	0	R/W	√	√	-	00H	149
FFE1H 到 FFE3H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FFE4H	MK0	1	<TMM K010>	<TMM K000>	<TMM KH1>	<PMK 1>	<PMK 0>	<LVIM K>	1	R/W	√	√	-	FFH	150
FFE5H 到 FFEBH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FFECH	INTM0	0	0	ES11	ES10	ES01	ES00	0	0	R/W	-	√	-	00H	150
FFEDH 到 FFF2H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FFF3H	PPCC	0	0	0	0	0	0	PPCC 1	PPCC 0	R/W	√	√	-	02H	66
FFF4H	OSTS	0	0	0	0	0	0	OSTS 1	OSTS 0		-	√	-	不确定	68
FFF5H 到 FFFAH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FFFBH	PCC	0	0	0	0	0	0	PCC1	0	R/W	√	√	-	02H	66

备注 对于三角括号 (<>) 中的位名称, 它是 RA78K0S 中的保留字, 而在 CC78K0S 中则通过使用 #pragma sfr 导引来定义为一个 sfr 变量。

3.3 指令地址寻址

指令地址通过程序计数器（PC）的内容来确定。PC 的内容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）。当执行跳转指令时，跳转目的地址信息会被设置到 PC 中并且通过以下寻址方法跳转（关于指令的详细情况，参阅 **78K/0S 系列指令用户手册（U11047E）**）。

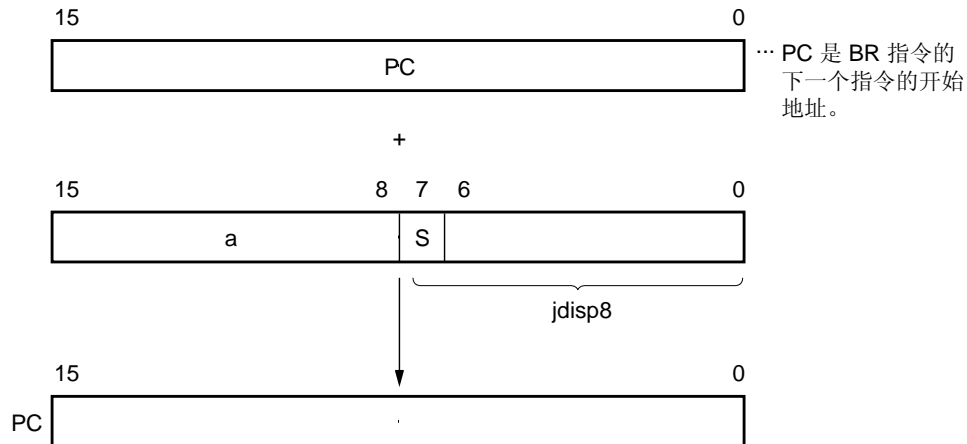
3.3.1 相对寻址

[功能]

将指令代码的 8 位立即数（替换值：**jdisp8**）与以下指令中的开始地址相加后所得的值传送到程序计数器（PC）中并转向相加结果指向的地址。替换值被当作带符号位的二进制补码（-128 到 +127），而第 7 位则变为一个符号位。也就是说，在相对寻址中，跳转的范围在以下指令的开始地址的-128 和 +127 之间。

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

[图示]



当 $S = 0$ 时，**a** 表明所有位为“0”。
当 $S = 1$ 时，**a** 表明所有位为“1”。

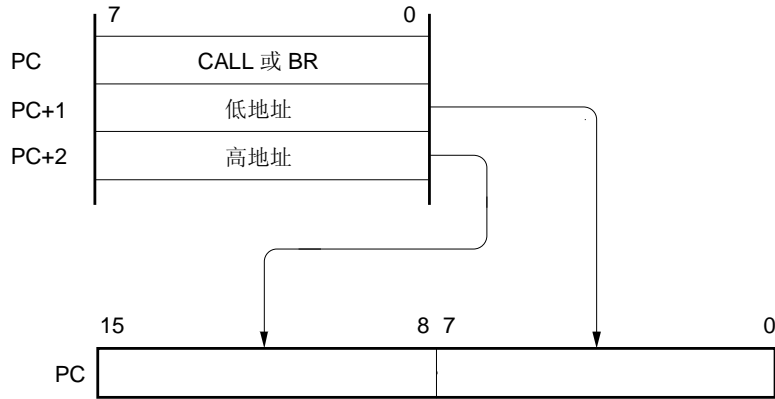
3.3.2 立即寻址

[功能]

指令字中的立即数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 和 BR !addr16 指令时实现。
 CALL !addr16 和 BR !addr16 指令可以转移到整个存储器空间中。

[图示]

使用 CALL !addr16 和 BR !addr16 指令时

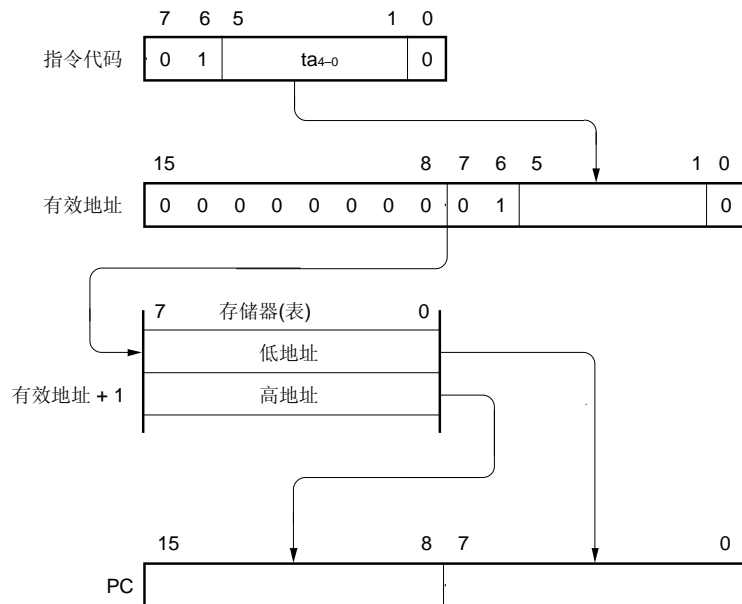


3.3.3 表间接寻址

[功能]

通过指令码的立即数的 1 到 5 位寻址的特定位置的表内容（跳转目的地址）被赋值给程序计数器（PC）并转向该地址。
 表间接寻址在执行 CALLT [addr5]指令时实现。根据保存在存储器表 40H 到 7FH 中的地址，该指令可以转移到整个存储器空间中。

[图示]

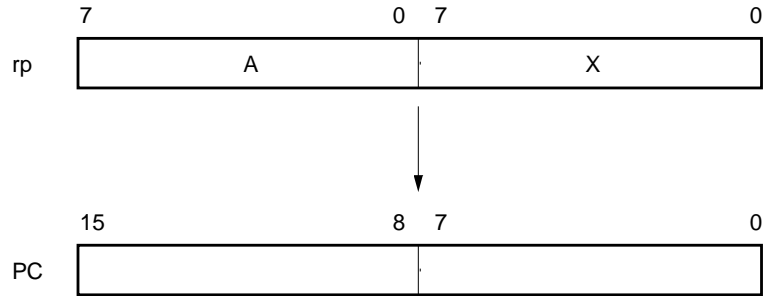


3.3.4 寄存器寻址

[功能]

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（寻址）来指定寄存器和存储器用以实现指令执行过程中的操作。

3.4.1 直接寻址

[功能]

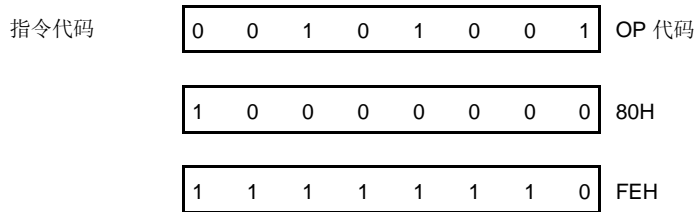
指令字中立即数表示的存储器被直接寻址。

[操作数格式]

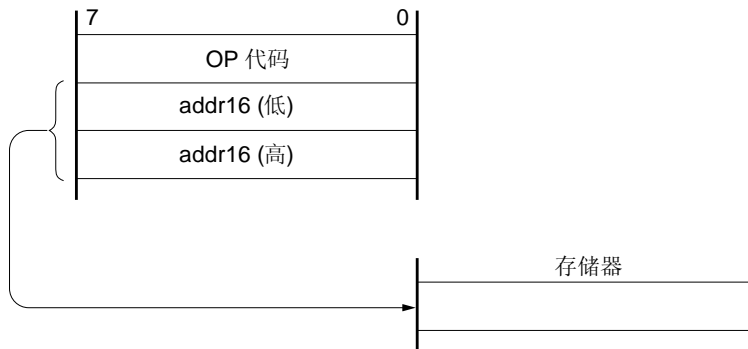
标识符	说明
addr16	标签或 16 位立即数

[说明示例]

MOV A, !FE80H; 当设置 !addr16 为 FE80H 时



[图示]



3.4.2 短直接寻址

[功能]

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

该寻址应用的固定空间为 FE80H 到 FF1FH (FE80H 到 FEFFH (内部高速 RAM) + FF00H 到 FF1FH (特殊功能寄存器)) 的 160 字节空间。

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。在该区域中, 在程序中经常访问的端口和定时器计数的比较寄存器被映射, 并且这些 SFR 可以使用少量的字节和时钟进行操作。

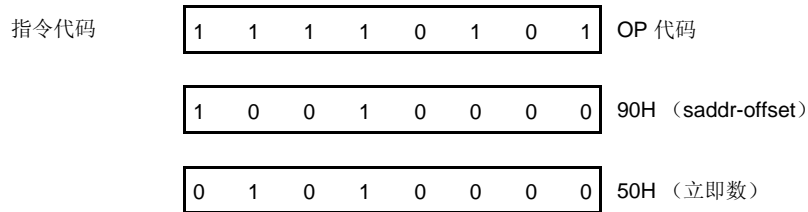
当 8 位立即数在 80H 到 FFH 内时, 有效地址的第 8 位将会被清除为 0。当 8 位立即数在 00H 到 1FH 内时, 有效地址的第 8 位则将会被设置为 1。参见以下的[图示]。

标识符	说明
saddr	标签或 FE80H 到 FF1FH 的立即数
saddrp	标签或 FE80H 到 FF1FH 的立即数 (仅限偶地址)

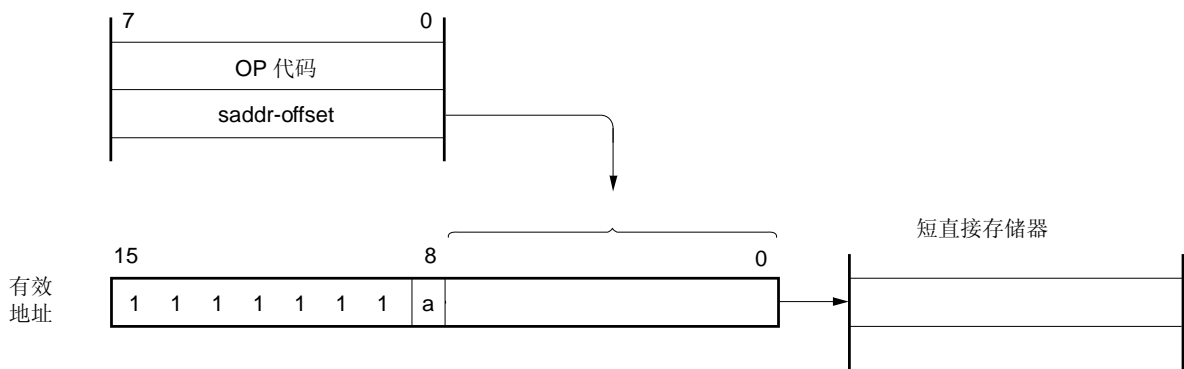
[说明示例]

EQU DATA1 0FE90H ; DATA1 表示 saddr 区域的 FE90H,

MOV DATA1, #50H ; 当设置立即数为 50H 时



[图示]



当 8 位立即数为 20H 到 FFH 时, a = 0。
 当 8 位立即数为 00H 到 1FH 时, a = 1。

3.4.3 特殊功能寄存器 (SFR) 寻址

[功能]

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

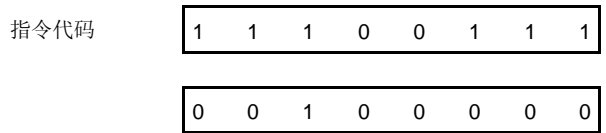
该寻址方式应用于 256 字节的空间 FF00H 到 FFFFH 中。然而，映射在 FF00H 到 FF1FH 内的 SFR 可以通过短直接寻址方式进行访问。

[操作数格式]

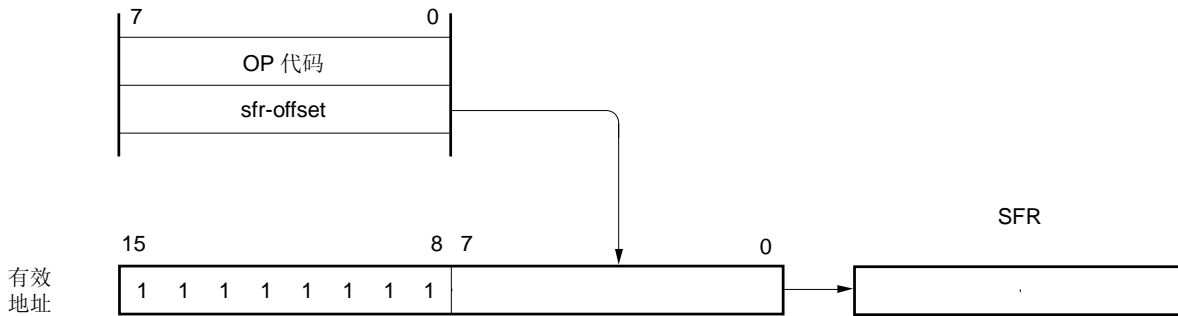
标识符	说明
sfr	特殊功能寄存器名

[说明示例]

MOV PM0, A; 当为 sfr 选择 PM0 时



[图示]



3.4.4 寄存器寻址

[功能]

一个通用寄存器作为操作数被访问。

将被访问的通用寄存器使用指令代码中的寄存器指定代码和功能名称来指定。

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，指令代码中将会有 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; 当为 r 选择 C 寄存器时

指令代码

0 0 0 0 1 0 1 0

0 0 1 0 0 1 0 1

寄存器描述码

INCW DE; 当为 rp 选择 DE 寄存器时

指令代码

1 0 0 0 1 0 0 0

寄存器描述码

3.4.5 寄存器间接寻址

[功能]

使用指定为操作数的寄存器对的内容，存储器被寻址。将被访问的寄存器对使用指令代码中的寄存器对指定代码来指定。所有存储器空间都可以实现该寻址。

[操作数格式]

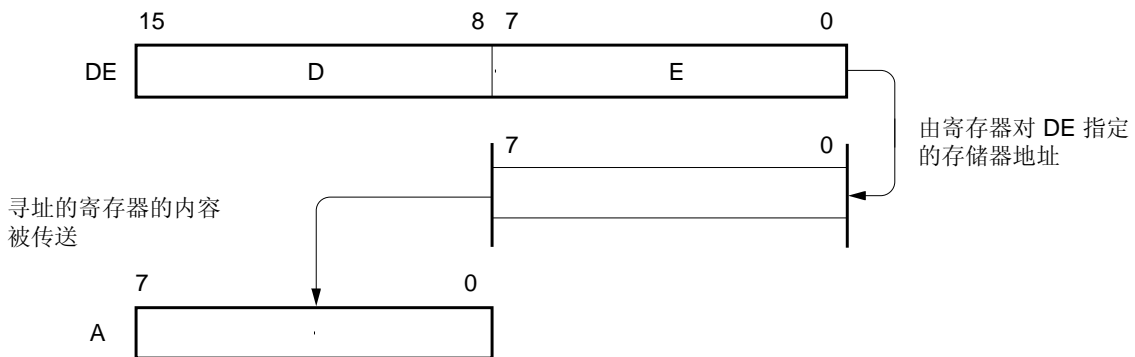
标识符	说明
-	[DE], [HL]

[说明示例]

MOV A, [DE]; 当选择寄存器对 [DE] 时



[图示]



3.4.6 基址寻址

[功能]

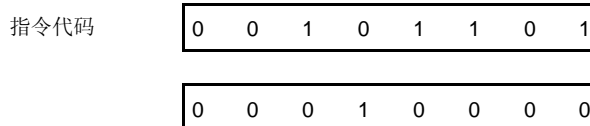
8 位立即数被加到基地址寄存器的内容中，即 HL 寄存器对，并且相加的和被用来寻址存储器。加法操作是通过将偏移数据扩展会 16 位的正数来完成的。第 16 位的进位忽略不计。所有存储器空间都可以实现该寻址。

[操作数格式]

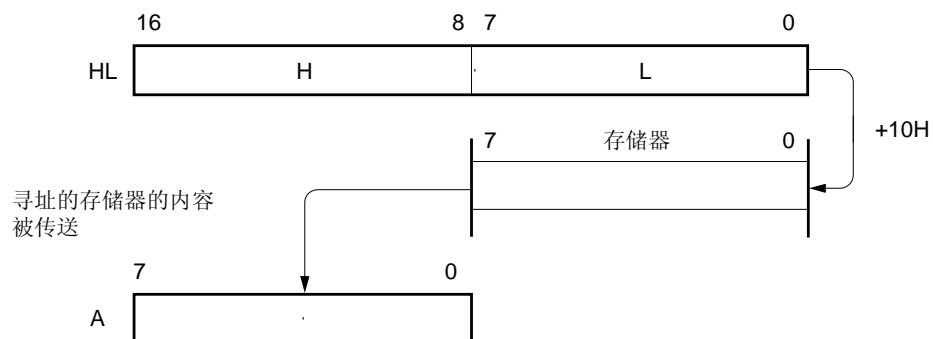
标识符	说明
-	[HL+byte]

[说明示例]

MOV A, [HL+10H]; 当设置字节为 10H 时



[图示]



3.4.7 堆栈寻址

[功能]

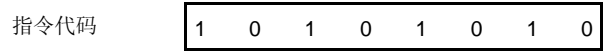
堆栈区域通过堆栈指针（SP）的内容进行间接寻址。

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。

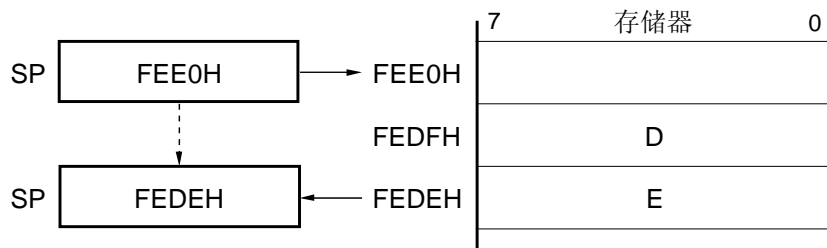
堆栈寻址只能用于访问内部高速 RAM 区域。

[说明示例]

在 PUSH DE 的情况下



[图示]



第四章 端口功能

4.1 端口的功能

μ PD78F9510、78F9511、78F9512 具有如图 4-1 中所示的端口，这些端口支持多种控制操作。表 4-1 表示每个端口的功能。

除了用作数字 I/O 端口外，每个端口还具有复用功能。关于详细情况，参阅第二章 引脚功能。

图 4-1. 端口功能

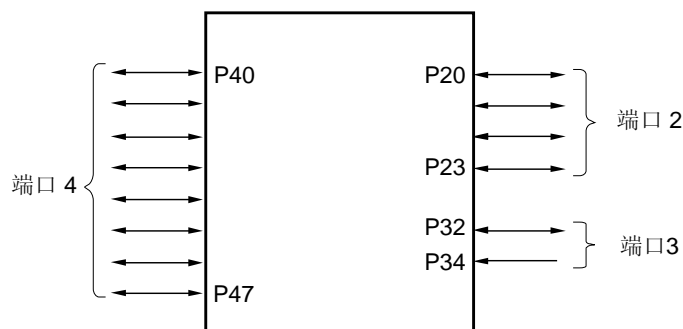


表 4-1. 端口功能

引脚名称	I/O	功能		复位后	复用功能引脚
P20	I/O	端口 2 4 位 I/O 端口。 可以按位指定为输入或输出模式。 片上上拉电阻可以通过软件连接。		输入	TI000/TOH1
P21					TI010/TO00/ INTP0
P22 ^注					X2 ^注
P23 ^注					X1 ^注
P32	I/O	端口 3	可以按位指定为输入或输出模式。 片上上拉电阻可以通过软件连接。	输入	INTP1
P34 ^注					
P40 ~ P47	I/O	端口 4 8 位 I/O 端口。 可以以 1 位为单位指定为输入或输出模式。 片上上拉电阻可以通过软件连接。		输入	-

注 关于引脚功能的设置方法，参见第十四章 选项字节。

注意事项 P22/X2 和 P23/X1 引脚在复位期间下拉。

- 备注
1. 当高速内部振荡被选作系统时钟时，P22 和 P23 可以被分配。
 2. 当外部时钟输入被选作系统时钟时，P22 可以被分配。

4.2 端口配置

端口由以下硬件单元组成。

表 4-2. 端口的配置

项目	配置
控制寄存器	端口模式寄存器 (PM2 ~ PM4) 端口寄存器 (P2 ~ P4) 上拉电阻选项寄存器 (PU2 ~ PU4)
端口	总数: 14 (CMOS I/O: 13, CMOS 输入: 1)
上拉电阻	总数: 13

4.2.1 端口 2

端口 2 是带有输出锁存的 4 位 I/O 端口。该端口的每一位可以通过使用端口模式寄存器 2 (PM2) 设置为输入或输出模式。当 P20 ~ P23 引脚用作输入端口时, 片上上拉电阻可以通过上拉电阻选择寄存器 2 (PU2) 以 1 位为单位进行连接。

P22 和 P23 引脚也用作系统时钟振荡器的 X2 和 X1 引脚。因此, P22 和 P23 引脚的功能根据选择的系统时钟振荡器而不同。以下三种系统时钟振荡器可以被使用。

(1) 高速内部振荡器

P22 和 P23 引脚可以用作 I/O 端口引脚。

(2) 晶体 / 陶瓷振荡器

P22 和 P23 引脚不能用作 I/O 端口引脚, 因为它们用作 X2 和 X1 引脚。

(3) 外部时钟输入

P22 引脚可以用作 I/O 端口引脚。

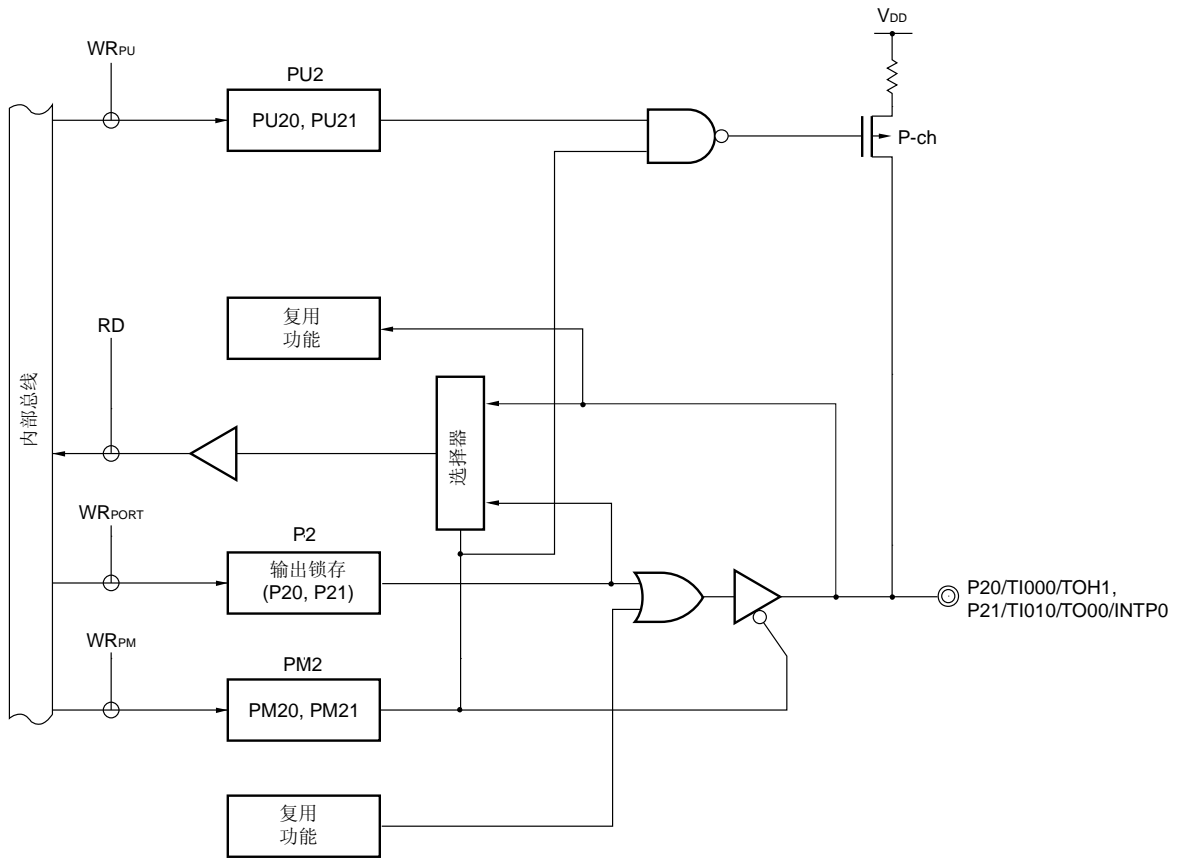
P23 引脚用作输入外部时钟的 X1 引脚, 因此它不能用作 I/O 端口引脚。

系统时钟振荡通过选项字节来选择。关于详细情况, 参阅第十四章 选项字节。

复位信号生成将端口 2 设置为输入模式。

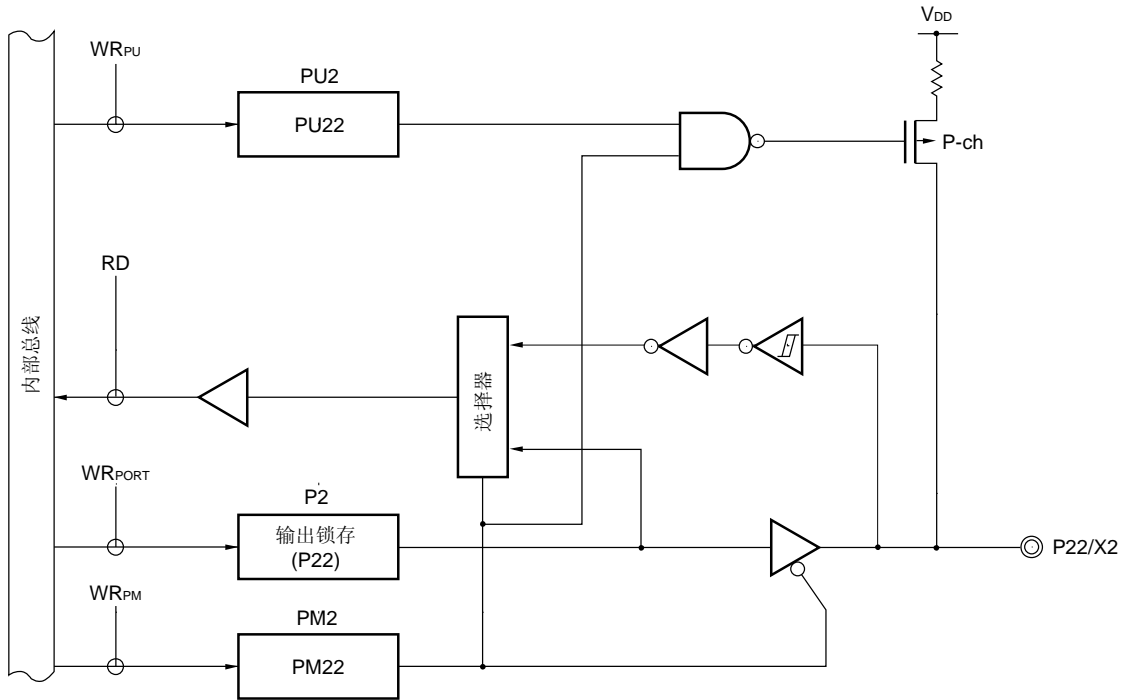
图 4-2 ~ 4-4 显示了端口 2 的框图。

图 4-2. P20 和 P21 的框图



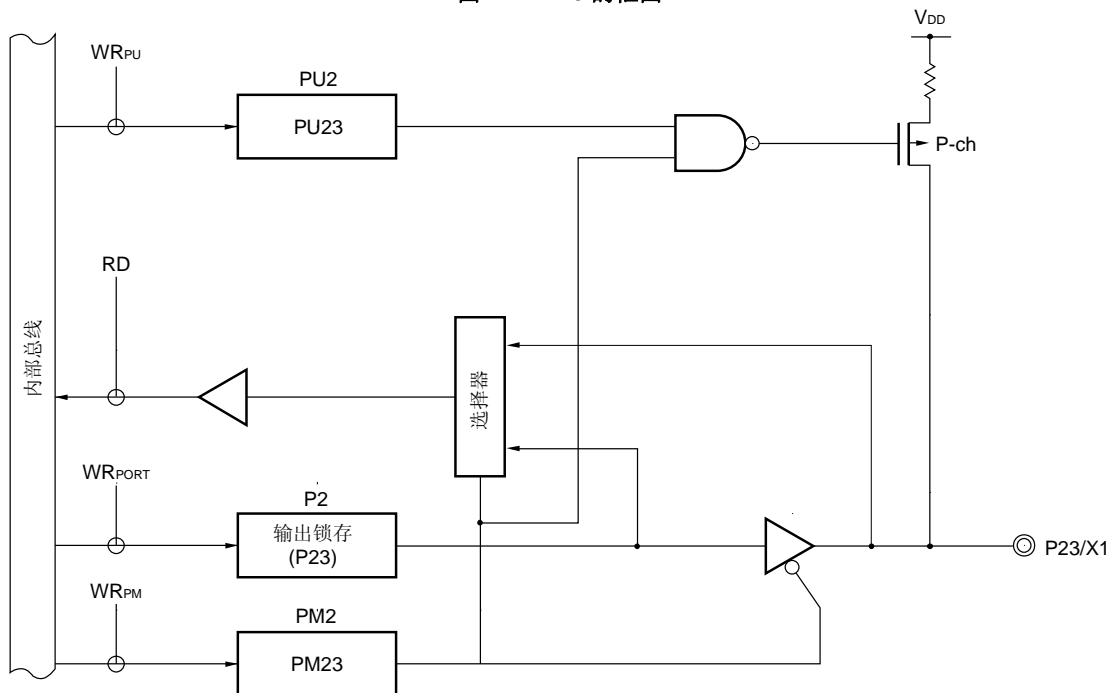
- P2: 端口寄存器 2
- PU2: 上拉电阻选项寄存器 2
- PM2: 端口模式寄存器 2
- RD: 读取信号
- WR_{xx}: 写入信号

图 4-3. P22 的框图



- P2: 端口寄存器 2
- PU2: 上拉电阻选项寄存器 2
- PM2: 端口模式寄存器 2
- RD: 读取信号
- WR_{xx}: 写入信号

图 4-4. P23 的框图



- P2: 端口寄存器 2
- PU2: 上拉电阻选项寄存器 2
- PM2: 端口模式寄存器 2
- RD: 读取信号
- WR_{xx}: 写入信号

4.2.2 端口 3

P32 引脚是带有输出锁存的 1 位 I/O 端口。该引脚可以通过使用端口模式寄存器 3 (PM3) 设置为输入或输出模式。当该引脚用作输入端口时，片上上拉电阻可以通过上拉电阻选择寄存器 3 (PU3) 以 1 位为单位进行连接。该引脚也可用于外部中断请求输入。

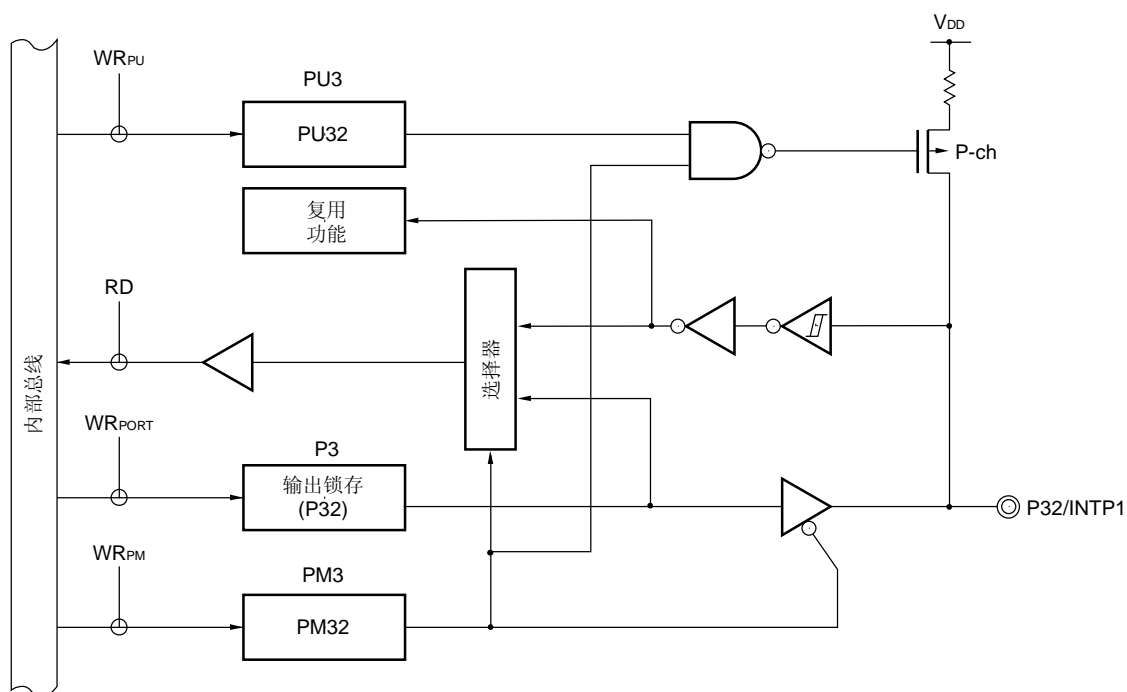
复位信号生成将端口 3 设置为输入模式。

P34 引脚为 1 位输入端口。该引脚也用作 $\overline{\text{RESET}}$ 引脚，并且当上电时，它是复位功能。关于引脚功能的设置方法，参见第十四章 选项字节。

当 P34 用作输入端口引脚时，连接上拉电阻。

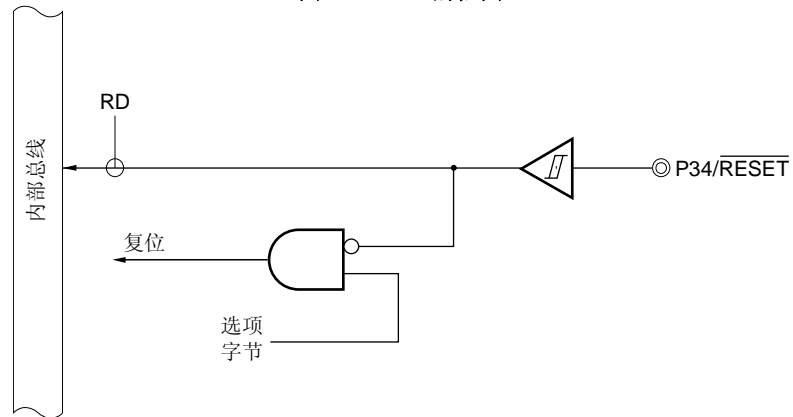
图 4-5 和 4-6 显示了端口 3 的框图。

图 4-5. P32 的框图



- P3: 端口寄存器 3
- 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

图 4-6. P34 的框图



RD: 读取信号

注意事项 因为 P34 引脚复用作 $\overline{\text{RESET}}$ 引脚，如果它用作输入端口引脚，输入一个外部复位信号到 $\overline{\text{RESET}}$ 引脚的功能不能被使用。端口的功能通过选项字节来选择。关于详细情况，参阅第十四章 选项字节。

同时，由于选项字节在复位释放后被引用，如果在引用之前低电平被输入 $\overline{\text{RESET}}$ 引脚，复位状态不会被释放。当它用作输入端口引脚时，连接上拉电阻。

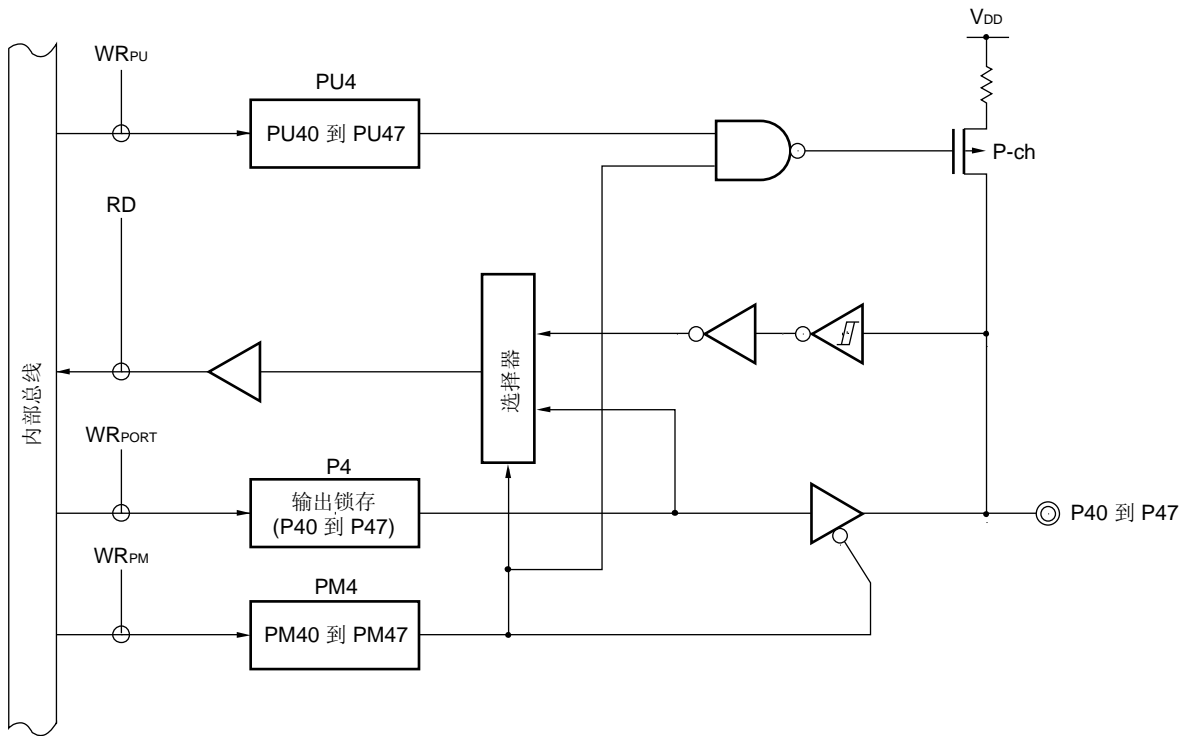
4.2.3 端口 4

端口 4 是带有输出锁存的 8 位 I/O 端口。该端口的每一位可以通过使用端口模式寄存器 4 (PM4) 设置为输入或输出模式。当 P40 ~ P47 引脚用作输入端口时，片上上拉电阻可以通过上拉电阻选择寄存器 4 (PU4) 以 1 位为单位进行连接。

复位信号生成将端口 4 设置为输入模式。

图 4-7 显示了端口 4 的框图。

图 4-7. P40 ~ P47 的框图



- P4: 端口寄存器 4
- 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

4.3 控制端口功能的寄存器

端口由以下四种类型的寄存器进行控制。

- 端口模式寄存器 (PM2 ~ PM4)
- 端口寄存器 (P2 ~ P4)
- 上拉电阻选项寄存器 (PU2~PU4)

(1) 端口模式寄存器 (PM2 ~ PM4)

这些寄存器用来以 1 位为单位设置对应的端口为输入或输出模式。

每个端口模式寄存器可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将这些寄存器设置为 FFH。

当端口引脚用作复用功能时，应按表 4-3 中所示对端口模式寄存器和输出锁存进行设置。

注意事项 因为 P21 和 P32 也用作外部中断引脚，如果这些引脚被设置为输出模式并且其输出电平被更改，对应的中断请求标志被设置。要在输出模式下使用端口引脚，事先设置对应的中断屏蔽标志为 1。

图 4-8. 端口模式寄存器的格式

地址: FF22H, 复位后: FFH, R/W

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

地址: FF23H, 复位后: FFH, R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	PM32	1	1

地址: FF24H, 复位后: FFH, R/W

符号	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

PMmn	Pmn 引脚 I/O 模式选择 (m = 2~4; n = 0~7)
0	输出模式 (输出缓存打开)
1	输入模式 (输出缓存关闭)

(2) 端口寄存器 (P2 ~ P4)

这些寄存器用来将要从对应端口引脚输出的数据写入连接到芯片的外部设备中。

当一个端口寄存器被读取时，在输入模式下引脚电平被读取，而在输出模式下端口的输出锁存值被读取。

P20 ~ P23、P32 和 P40 ~ P47 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将这些寄存器设置为 00H。

图 4-9. 端口寄存器的格式

地址: FF02H, 复位后: 00H (输出锁存) R/W

符号	7	6	5	4	3	2	1	0
P2	0	0	0	0	P23	P22	P21	P20

地址: FF03H, 复位后: 00H^注 (输出锁存) R/W^注

符号	7	6	5	4	3	2	1	0
P3	0	0	0	P34	0	P32	0	0

地址: FF04H, 复位后: 00H (输出锁存) R/W

符号	7	6	5	4	3	2	1	0
P4	P47	P46	P45	P44	P43	P42	P41	P40

Pmn	m = 2 ~ 4; n = 0 ~ 7	
	输出数据的控制 (输出模式下)	输入数据读取 (输入模式下)
0	输出 0	输入低电平
1	输出 1	输入高电平

注 由于 P34 只能读取，其复位值不确定。

(3) 上拉电阻选项寄存器 (PU2~PU4)

这些寄存器用来指定片上上拉电阻是否连接到 P20 ~ P23、P32 和 P40 ~ P47。通过设置 PU2~PU4，一个片上上拉电阻可以被连接到对应于 PU2 ~ PU4 的位的端口引脚。

PU2 ~ PU4 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将这些寄存器设置为 00H。

图 4-11. 上拉电阻选项寄存器的格式

地址: FF32H, 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PU2	0	0	0	0	PU23	PU22	PU21	PU20

地址: FF33H, 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PU3	0	0	0	0	0	PU32	0	0

地址: FF34H, 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40

PU _{mn}	P _m n 的片上上拉电阻连接的选择 (m = 2~4; n = 0~7)
0	不连接片上上拉电阻
1	连接片上上拉电阻

4.4 端口功能的操作

端口的功能根据 I/O 模式的设置而不同，如下所示。

注意事项 尽管 1 位存储器操作指令修改 1 位，但是它以 8 位为单位访问端口。因此，在输入和输出混合的端口中，输入模式下引脚的输出锁存的内容不确定，即使它不是指令修改的对象。

4.4.1 写入 I/O 端口

(1) 输出模式下

一个值可以通过传送指令写入输出锁存。此外，输出锁存的内容从引脚输出。一旦数据被写入到输出锁存中，它将会被保留直到新的数据被写入到输出锁存中。

当复位信号被产生时，清除输出锁存中的数据。

(2) 输入模式下

一个值可以通过传送指令写入输出锁存。然而，因为输出缓存被关闭，引脚状态保持不确定。

一旦数据被写入到输出锁存中，它将会被保留直到新的数据被写入到输出锁存中。

当复位信号被产生时，清除输出锁存中的数据。

4.4.2 从 I/O 端口中读取

(1) 输出模式下

输出锁存的内容可以通过传送指令读取。输出锁存的内容保持不确定。

(2) 输入模式下

引脚状态可以通过传送指令读取。输出锁存的内容保持不确定。

4.4.3 I/O 端口上的操作

(1) 输出模式下

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，它将会被保留直到新的数据被写入到输出锁存中。

复位信号清除输出锁存中的数据。

(2) 输入模式下

引脚电平被读取且操作会在其内容上进行。操作结果被写入输出锁存。然而，引脚状态保持不确定，因为输出缓存被关闭，。

当复位信号被产生时，清除输出锁存中的数据。

第五章 时钟发生器

5.1 时钟发生器的功能

时钟发生器包含产生提供给 CPU 和周边硬件的时钟（系统时钟）的电路和产生提供给看门狗定时器和 8 位定时器 H1（TMH1）的时钟（间隔时间产生时钟）的电路。

5.1.1 系统时钟振荡器

以下三类系统时钟振荡器被使用。

- 高速内部振荡器
该电路内部振荡一个 8 MHz（典型值）的时钟。其振荡可以由 STOP 指令的执行停止。
如果高速内部振荡器被选择提供给系统时钟，X1 和 X2 引脚可以被用作 I/O 端口引脚。
- 晶体 / 陶瓷谐振器
该电路使用通过 X1 和 X2 引脚连接的晶体 / 陶瓷振荡器来振荡一个时钟。它可以振荡 1MHz 到 10 MHz 的时钟。
该电路的振荡可以由 STOP 指令的执行停止。
- 外部时钟输入电路
该电路从连接到 X1 引脚的外部 IC 提供时钟。1MHz 到 10 MHz 的时钟可以被提供。内部时钟提供可以由 STOP 指令的执行停止。
如果外部时钟输入被选作系统时钟，X2 引脚可以被用作 I/O 端口引脚。

系统时钟源使用选项字节来选择。关于详细情况，参阅第十四章 选项字节。

当使用 X1 和 X2 引脚作为 I/O 端口引脚时，关于详细情况，参阅第四章 端口功能。

5.1.2 用于间隔时间产生的时钟振荡器

以下电路用作间隔时间产生的时钟振荡器。

- 低速内部振荡器
该电路用于振荡一个 240 kHz（典型值）的时钟。当通过选项字节指定为其振荡可由软件停止时，振荡可以通过使用低速内部振荡模式寄存器（LSRCM）来停止。

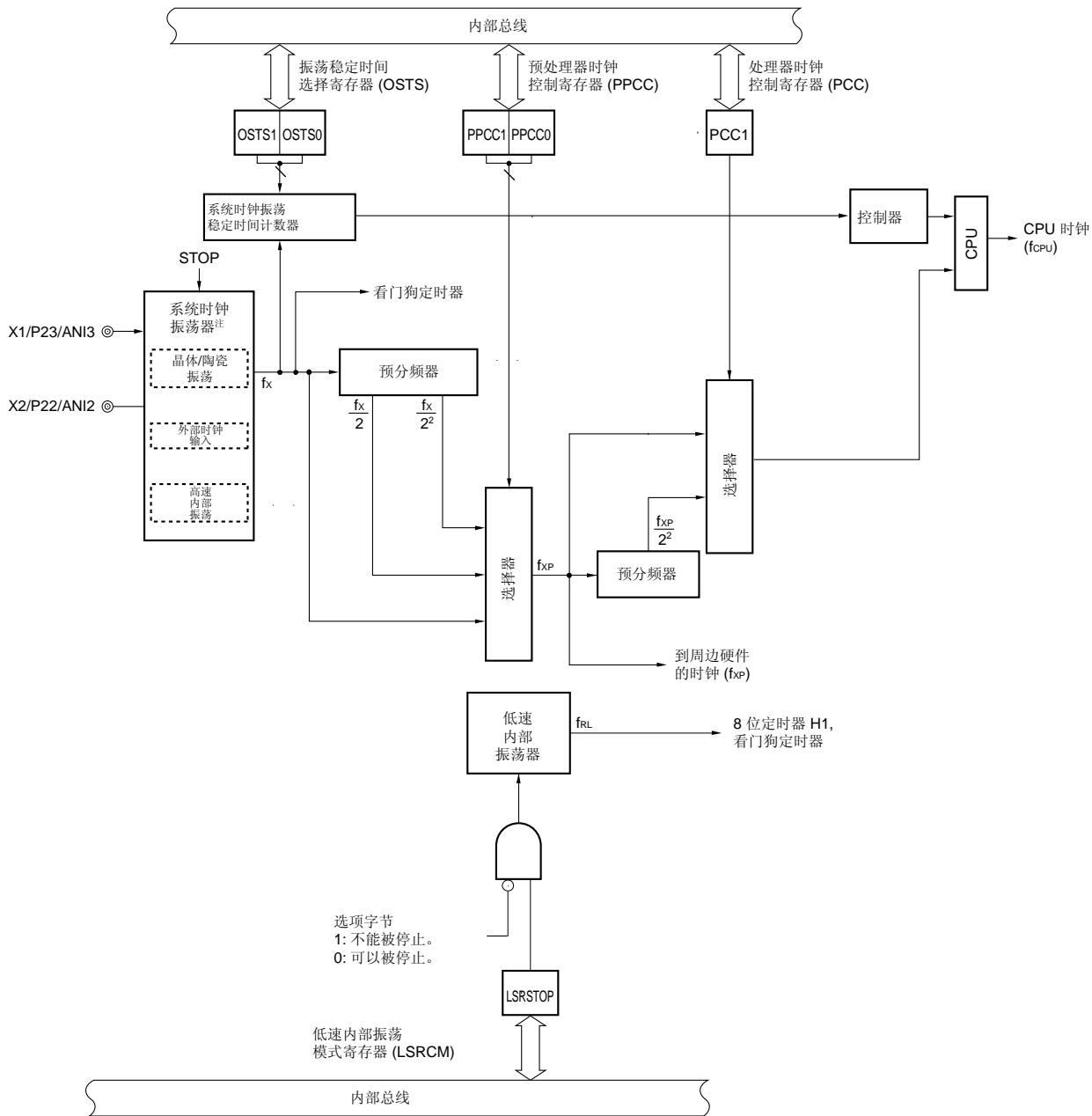
5.2 时钟发生器的配置

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) 预处理器时钟控制寄存器 (PPCC) 低速内部振荡模式寄存器 (LSRCM) 振荡稳定时间选择寄存器 (OSTS)
振荡器	晶体 / 陶瓷振荡器 高速内部振荡器 外部时钟输入电路 低速内部振荡器

图 5-1. 时钟发生器的框图



注 通过使用选项字节来选择高速内部振荡器、晶体 / 陶瓷振荡器或外部时钟输入电路作为系统时钟源。

5.3 控制时钟发生器的寄存器

时钟发生器由以下四个寄存器控制。

- 处理器时钟控制寄存器（PCC）
- 预处理器时钟控制寄存器（PPCC）
- 低速内部振荡模式寄存器（LSRCM）
- 振荡稳定时间选择寄存器（OSTS）

（1）处理器时钟控制寄存器（PCC）和预处理器时钟控制寄存器（PPCC）

这些寄存器用来指定系统时钟的分频比率。

PCC 和 PPCC 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将 PCC 和 PPCC 设为 02H。

图 5-2. 处理器时钟控制寄存器（PCC）的格式

地址：FFF_BH，复位后：02H，R/W

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

图 5-3. 预处理器时钟控制寄存器（PPCC）的格式

地址：FFF₃H，复位后：02H，R/W

符号	7	6	5	4	3	2	1	0
PPCC	0	0	0	0	0	0	PPCC1	PPCC0

PPCC1	PPCC0	PCC1	CPU 时钟 (f_{CPU}) 的选择
0	0	0	f_x
0	1	0	$f_x/2^{\text{注}1}$
0	0	1	$f_x/2^2$
1	0	0	$f_x/2^2^{\text{注}2}$
0	1	1	$f_x/2^3^{\text{注}1}$
1	0	1	$f_x/2^4^{\text{注}2}$
除上面以外			禁止设置

- 注
1. 如果 PPCC = 01H，提供给周边硬件的时钟 (f_{XP}) 为 $f_x/2$ 。
 2. 如果 PPCC = 02H，提供给周边硬件的时钟 (f_{XP}) 为 $f_x/2^2$ 。

μ PD78F9510、78F9511、78F9512 的最快指令在两个 CPU 时钟周期内执行。因此，表 5-2 显示了 CPU 时钟 (f_{CPU}) 与最短指令执行时间之间的关系。

表 5-2. CPU 时钟与最短指令执行时间之间的关系

CPU 时钟 (f_{CPU}) ^注	最短指令执行时间: $2/f_{CPU}$	
	高速内部振荡时钟 (8.0 MHz (典型值))	晶体 / 陶瓷振荡时钟 或外部时钟输入 (10.0 MHz)
f_x	0.25 μ s	0.2 μ s
$f_x/2$	0.5 μ s	0.4 μ s
$f_x/2^2$	1.0 μ s	0.8 μ s
$f_x/2^3$	2.0 μ s	1.6 μ s
$f_x/2^4$	4.0 μ s	3.2 μ s

注 CPU 时钟 (高速内部振荡时钟、晶体 / 陶瓷振荡器时钟或外部时钟输入) 通过选项字节来选择。

(2) 低速内部振荡模式寄存器 (LSRCM)

该寄存器用于选择低速内部振荡器 (240 kHz (典型值)) 的操作模式。

当通过选项字节指定为低速内部振荡器可由软件停止时，该寄存器有效。如果通过选项字节指定为低速内部振荡器不能由软件停止，该寄存器的设置无效并且低速内部振荡器继续振荡。此外，WDT 的源时钟固定为低速内部振荡器。关于详细情况，参阅第八章 看门狗定时器。

LSRCM 可以通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将 LSRCM 设为 00H。

图 5-4. 低速内部振荡模式寄存器 (LSRCM) 的格式

地址: FF58H, 复位后: 00H, R/W

符号	7	6	5	4	3	2	1	<0>
LSRCM	0	0	0	0	0	0	0	LSRSTOP

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

(3) 振荡稳定时间选择寄存器 (OSTS)

该寄存器用于在 STOP 模式被释放时选择从振荡器提供的时钟的振荡稳定时间。只有当晶体 / 陶瓷振荡时钟被选作系统时钟并且 STOP 模式被释放后，由 OSTS 设置的等待时间才有效。如果高速内部振荡器或外部时钟输入被选作为系统时钟源，无等待时间。

通过选项字节选择系统时钟振荡器和上电或复位释放后的振荡稳定时间。关于详细情况，参阅第十四章 选项字节。

OSTS 通过使用 8 位存储器操作指令来设置。

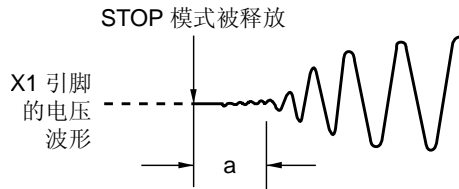
图 5-5. 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFF4H, 复位后: 不确定, R/W

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

OSTS1	OSTS0	振荡稳定时间的选择
0	0	$2^{10}/f_x$ (102.4 μ s)
0	1	$2^{12}/f_x$ (409.6 μ s)
1	0	$2^{15}/f_x$ (3.27 ms)
1	1	$2^{17}/f_x$ (13.1 ms)

- 注意事项**
1. 要设置并释放 STOP 模式，按照下面设置振荡稳定时间。
谐振器的期望振荡稳定时间 \leq 由 OSTS 设置的振荡稳定时间
 2. 无论 STOP 模式是由复位信号产生释放还是由中断产生释放，STOP 模式释放后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。”



3. 通过选项字节选择上电或复位释放后的振荡稳定时间。关于详细情况，参阅第十四章 选项字节。

- 备注**
1. () : $f_x = 10$ MHz
 2. 通过检查将使用的谐振器的特性来确定谐振器的振荡稳定时间。

5.4 系统时钟振荡器

以下三类系统时钟振荡器可用。

- 高速内部振荡器： 内部振荡一个 8 MHz（典型值）的时钟。
- 晶体 / 陶瓷谐振器： 振荡 1MHz 到 10 MHz 的时钟。
- 外部时钟输入电路： 提供一个 1MHz 到 10 MHz 的时钟到 X1 引脚。

5.4.1 高速内部振荡器

μ PD78F9510、78F9511、78F9512 包含一个高速内部振荡器（8 MHz（典型值））。

如果高速内部振荡通过选项字节被选作系统时钟，X1 和 X2 引脚可以被用作 I/O 端口引脚。

关于选项字节的详细情况，参阅第十四章 选项字节。关于 I/O 端口的详细情况，参阅第四章 端口功能。

5.4.2 晶体 / 陶瓷谐振器

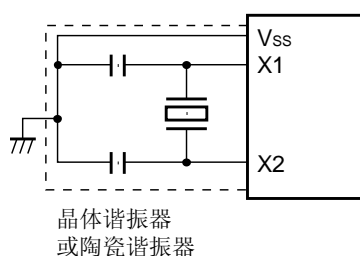
晶体 / 陶瓷振荡器使用连接到 X1 和 X2 引脚之间的晶体或陶瓷谐振器来振荡。

如果晶体 / 陶瓷振荡器通过选项字节被选作系统时钟，X1 和 X2 引脚被用作晶体或陶瓷谐振器连接引脚。

关于选项字节的详细情况，参阅第十四章 选项字节。关于 I/O 端口的详细情况，参阅第四章 端口功能。

图 5-6 显示了晶体 / 陶瓷振荡器的外部电路。

图 5-6. 晶体 / 陶瓷振荡器的外部电路



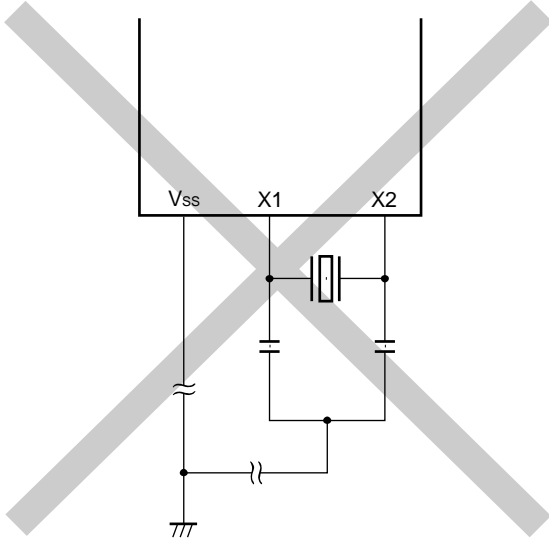
注意事项 当使用晶体 / 陶瓷振荡器时，图 5-6 中被虚线包围的部分的配线应按照如下布线方法布线，以防止走线电容产生不利影响。

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

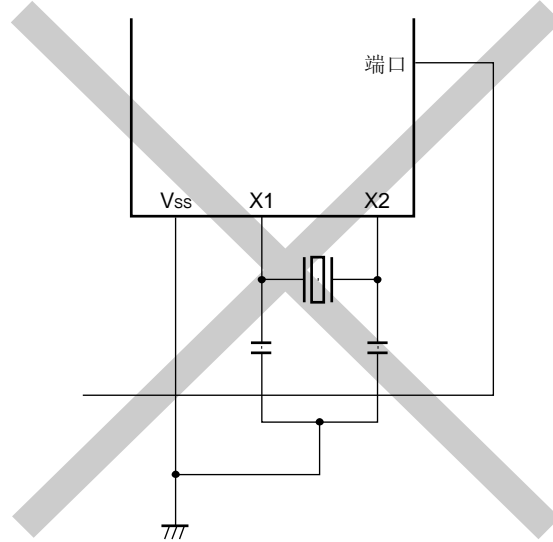
图 5-7 显示了错误的谐振器连接方式。

图 5-7. 错误的谐振器连接示例 (1 / 2)

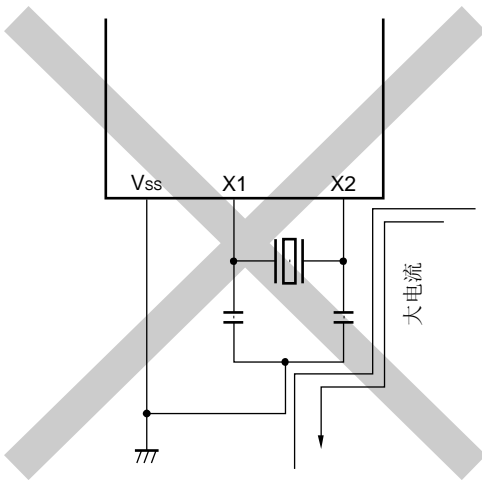
(a) 连接电路的连线过长



(b) 信号线交叉



(c) 连线附近电流过大



(d) 大电流通过振荡器的接地线
(A, B, C 三点电压波动)

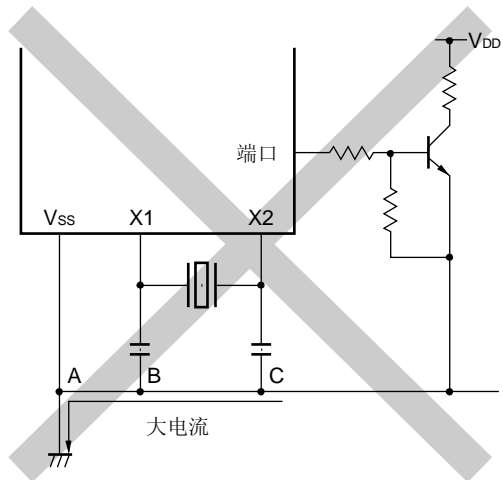
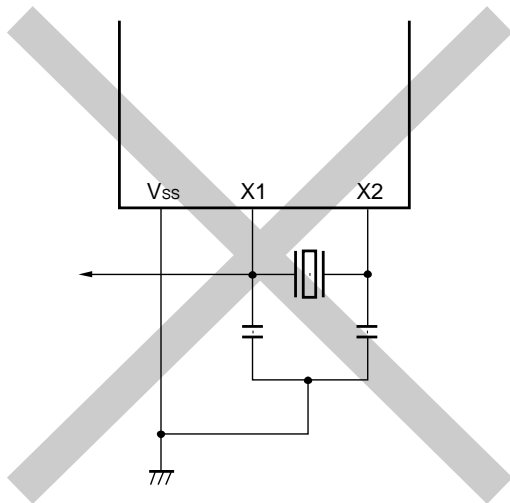


图 5-7. 错误的谐振器连接示例 (2 / 2)

(e) 从振荡器中获取信号



5.4.3 外部时钟输入电路

该电路从连接到 X1 引脚的外部 IC 提供时钟。

如果外部时钟输入通过选项字节被选作系统时钟源，X2 引脚可以被用作 I/O 端口引脚。

关于选项字节的详细情况，参阅第十四章 选项字节。关于 I/O 端口的详细情况，参阅第四章 端口功能。

5.4.4 预分频器

预分频器分频系统时钟振荡器输出的时钟 (f_x) 来产生提供给周边硬件的时钟 (f_{xp})。它也分频周边硬件的时钟来产生提供给 CPU 的时钟。

备注 通过选项字节选择的振荡器的时钟输出（高速内部振荡器、晶体 / 陶瓷振荡器或外部时钟输入电路）被分频。关于选项字节的详细情况，参阅第十四章 选项字节。

5.5 CPU时钟发生器的操作

从以下三类振荡器中的一类振荡的系统时钟 (fx) 中，一个时钟 (fcPU) 被提供给 CPU。

- 高速内部振荡器：内部振荡一个 8 MHz（典型值）的时钟。
- 晶体 / 陶瓷谐振器：振荡 1MHz 到 10 MHz 的时钟。
- 外部时钟输入电路：提供一个 1MHz 到 10 MHz 的时钟到 X1 引脚。

系统时钟振荡器由选项字节来选择。关于选项字节的详细情况，参阅第十四章 选项字节。

(1) 高速内部振荡器

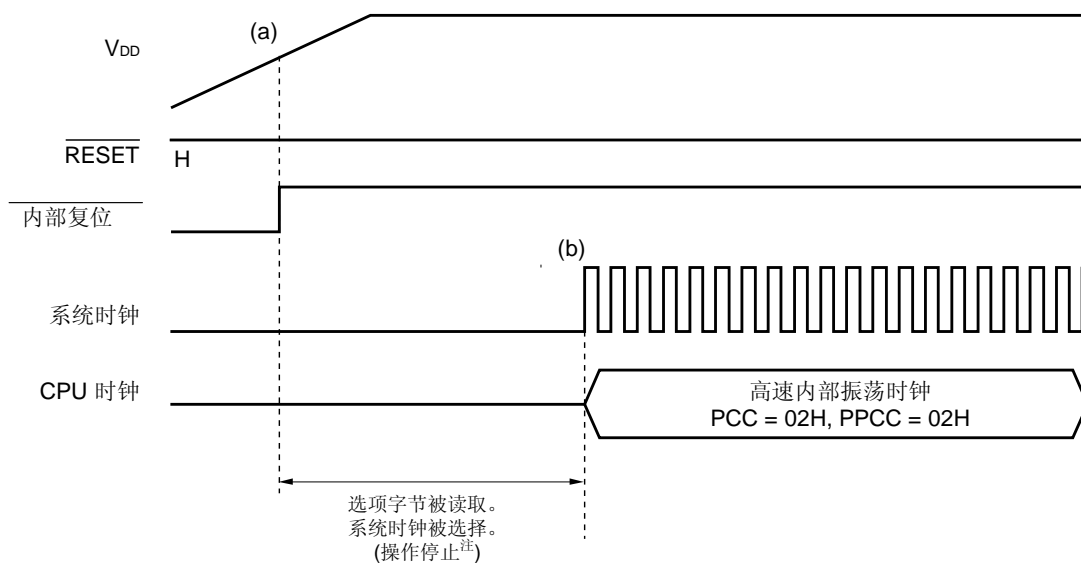
当高速内部振荡由选项字节选择时，可能发生以下情况。

- 开始时间的缩短
如果高速内部振荡器被选作振荡器，CPU 不必等待系统时钟的振荡稳定时间就可以开始。因此，开始时间被缩短。
- 扩展性的改进
如果高速内部振荡器被选作振荡器，X1 和 X2 引脚可以被用作 I/O 端口引脚。关于详细情况，参阅第四章 端口功能。

图 5-8 和 5-9 表示高速内部振荡的默认开始的时序图和状态转换图。

备注 当高速内部振荡被使用时，时钟精度为 $\pm 5\%$ 。

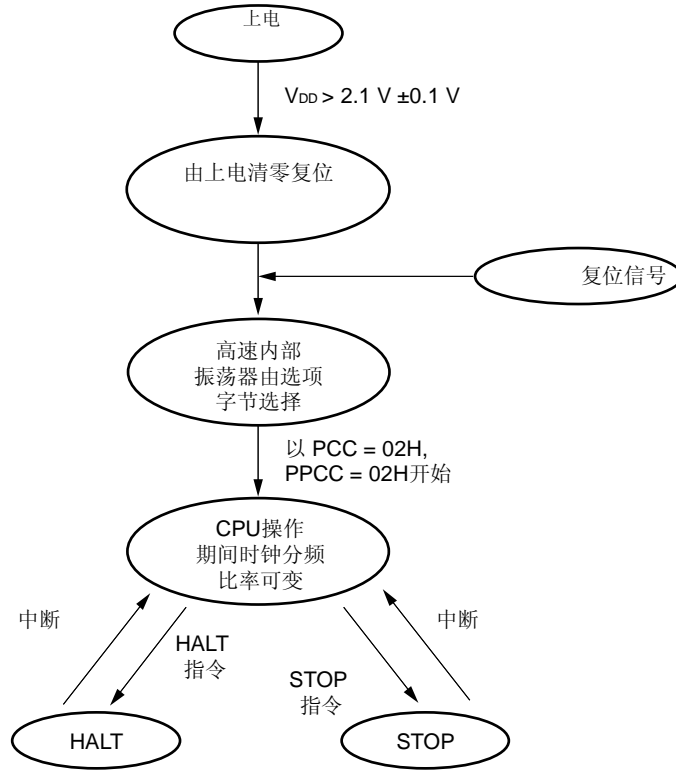
图 5-8. 高速内部振荡的默认开始的时序图



注 操作停止时间为 277 μ s（最小值），544 μ s（典型值）和 1.075 ms（最大值）。

- (a) 上电时内部复位信号由上电清零功能产生，选项字节在复位后被引用，并且系统时钟被选择。
- (b) 选项字节被引用并且系统时钟被选择。然后，高速内部振荡时钟作为系统时钟操作。

图 5-9. 高速内部振荡的默认开始的状态转换



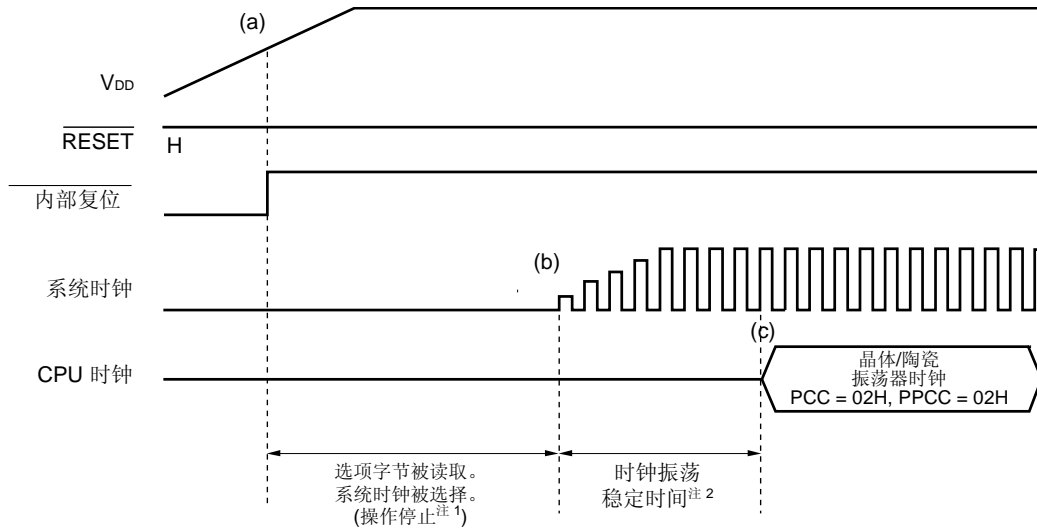
备注 PCC: 处理器时钟控制寄存器
PPCC: 预处理器时钟控制寄存器

(2) 晶体 / 陶瓷谐振器

如果晶体 / 陶瓷振荡通过选项字节被选择，1 MHz 到 10 MHz 的时钟频率被选择并且处理的精度被提高，因为相对于高速内部振荡（8 MHz（典型值））来说频率偏移减小。

图 5-10 和 5-11 表示晶体 / 陶瓷振荡器的默认开始的时序图和状态转换图。

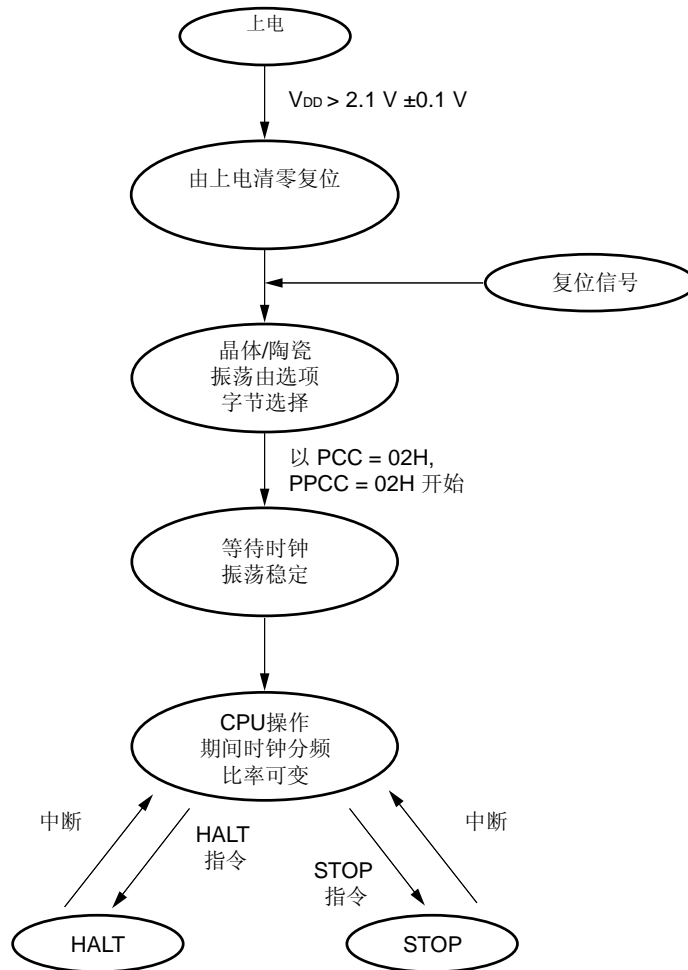
图 5-10. 晶体 / 陶瓷振荡器的默认开始的时序图



- 注**
1. 操作停止时间为 276 μ s（最小值），544 μ s（典型值）和 1.074 ms（最大值）。
 2. 默认开始的振荡稳定时间由选项字节选择。关于详细情况，参阅第十四章 选项字节。STOP 模式被释放后的振荡稳定时间由振荡稳定时间选择寄存器（OSTS）来选择。

- (a) 上电时内部复位信号由上电清零功能产生，选项字节在复位后被引用，并且系统时钟被选择。
- (b) 高速内部振荡时钟被产生后，选项字节被引用并且系统时钟被选择。在这种情况下，晶体 / 陶瓷振荡器时钟被选作系统时钟。
- (c) 如果系统时钟为晶体 / 陶瓷振荡器时钟，在时钟振荡稳定后，它作为 CPU 时钟操作。等待时间由选项字节来选择。关于详细情况，参阅第十四章 选项字节。

图 5-11. 晶体 / 陶瓷振荡器的默认开始的状态转换



备注 PCC: 处理器时钟控制寄存器
PPCC: 预处理器时钟控制寄存器

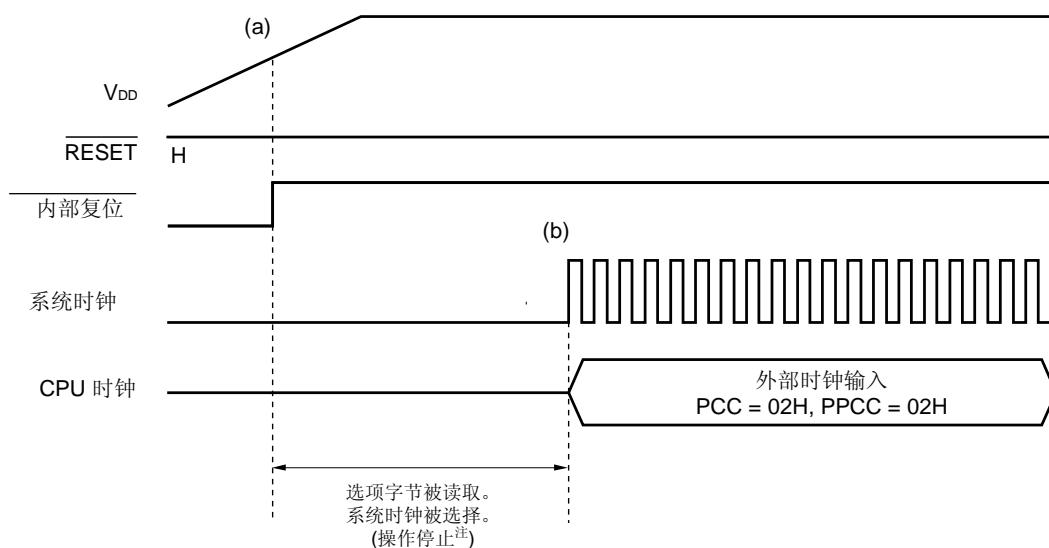
(3) 外部时钟输入电路

如果外部时钟输入由选项字节选择，可能发生以下情况。

- 高速操作
相对于高速内部振荡（8 MHz（典型值））来说处理的精度被提高，因为 1 MHz 到 10 MHz 的振荡频率可以被选择并且具有低频率偏移的外部时钟可以被提供。
- 扩展性的改进
如果外部时钟输入电路被选作振荡器，X2 引脚可以被用作 I/O 端口引脚。关于详细情况，参阅第四章 端口功能。

图 5-12 和 5-13 表示外部时钟输入的默认开始的时序图和状态转换图。

图 5-12. 外部时钟输入的默认开始的时序图

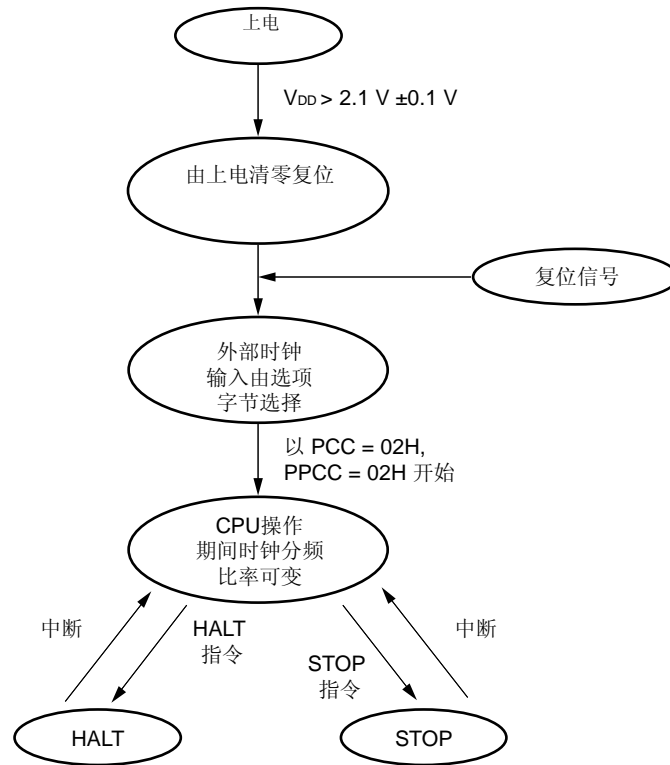


注 操作停止时间为 277 μ s（最小值），544 μ s（典型值）和 1.075 ms（最大值）。

(a) 上电时内部复位信号由上电清零功能产生，选项字节在复位后被引用，并且系统时钟被选择。

(b) 选项字节被引用并且系统时钟被选择。然后，外部时钟作为系统时钟操作。

图 5-13. 外部时钟输入的默认开始的状态转换



备注 PCC: 处理器时钟控制寄存器
 PPCC: 预处理器时钟控制寄存器

5.6 为周边硬件提供时钟的时钟发生器的操作

以下两类时钟被提供给周边硬件。

- 到周边硬件的时钟 (f_{xP})
- 低速内部振荡时钟 (f_{RL})

(1) 到周边硬件的时钟

通过分频系统时钟 (f_x)，到周边硬件的时钟被提供。分频比率由预处理器时钟控制寄存器 (PPCC) 来选择。三种频率可选。“f_x”，“f_x/2”和“f_x/2²”。表 5-3 列出了提供个周边硬件的时钟。

表 5-3. 到周边硬件的时钟

PPCC1	PPCC0	到周边硬件的时钟 (f _{xP}) 的选择
0	0	f _x
0	1	f _x /2
1	0	f _x /2 ²
1	1	禁止设置

(2) 低速内部振荡时钟

用于间隔时间产生的时钟振荡器的低速内部振荡器总是在复位释放后启动，并且以 240 kHz（典型值）振荡。可以通过选项字节指定低速内部振荡器是否可由软件停止。如果指定为低速内部振荡器可由软件停止，振荡可以使用低速内部振荡模式寄存器 (LSRCM) 来启动或停止。如果指定为不能由软件停止，WDT 的时钟源固定为低速内部振荡时钟 (f_{RL})。

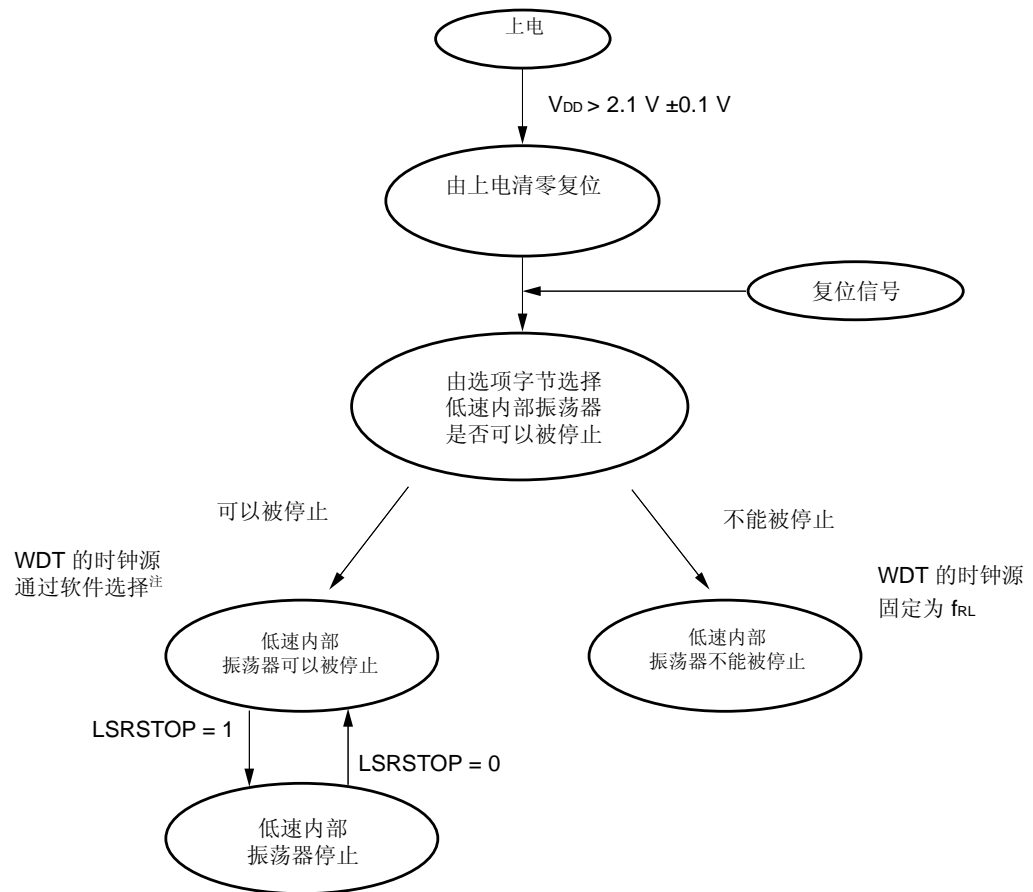
低速内部振荡器独立于 CPU 时钟。因此，如果它被用作 WDT 的源时钟，即使 CPU 时钟被停止，挂起也可以被检测。如果低速内部振荡器被用作 8 位定时器 H1 的计数时钟源，即使在待机状态下，8 位定时器 H1 也可以操作。

表 5-4 表示低速内部振荡器被选作 WDT 和 8 位定时器 H1 的源时钟时的操作状态。图 5-14 表示低速内部振荡器的状态转换

表 5-4. 低速内部振荡器的操作状态

选项字节设置		CPU 状态	WDT 状态	TMH1 状态
可由软件停止	LSRSTOP = 1	操作模式	停止	停止
	LSRSTOP = 0		操作	操作
	LSRSTOP = 1	待机	停止	停止
	LSRSTOP = 0		停止	操作
不能被停止		操作模式	操作	
		待机		

表 5-14. 低速内部振荡器的状态转换



注 看门狗定时器（WDT）的时钟源从 f_x 或 f_{RL} 中选择，或者它可能被停止。关于详细情况，参阅第八章 看门狗定时器。

6.1 16 位定时器 / 事件计数器 00 的功能

16 位定时器 / 事件计数器 00 具有以下功能。

(1) 间隔定时器

16 位定时器 / 事件计数器 00 可以在预定时间间隔生成一个中断请求。

- 计数个数：2 到 65536

(2) 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，脉冲具有一个或更多高电平 / 低电平有效电平脉冲宽度。

- 有效电平脉冲宽度： $2/f_{XP}$ 或更多

(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。

- 有效电平脉冲宽度： $2/f_{XP}$ 或更多

(4) 方波输出

16 位定时器 / 事件计数器 00 可以输出一个任意选中频率的方波。

- 周期： $(2 \text{ 到 } 65536) \times 2 \times \text{计数时钟周期}$

(5) PPG 输出

16 位定时器 / 事件计数器 00 可以输出一个任意周期和脉冲宽度的方波。

- $1 < \text{脉冲宽度} < \text{周期} \leq 65536$

(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任意设置。

6.2 16 位定时器 / 事件计数器 00 的配置

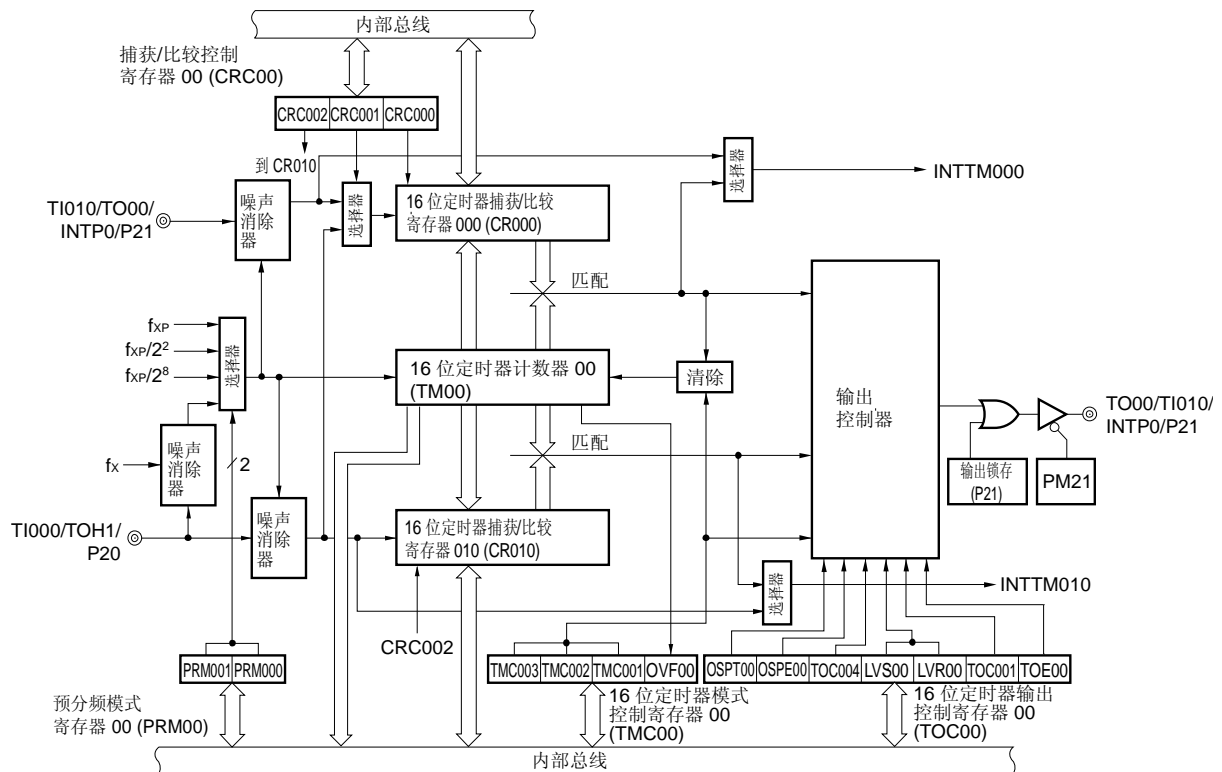
16 位定时器 / 事件计数器 00 由以下硬件组成。

表 6-1. 16 位定时器 / 事件计数器 00 的配置

项目	配置
定时器计数器	16 位定时器计数器 00 (TM00)
寄存器	16 位定时器捕获 / 比较寄存器 000, 010 (CR000, CR010)
定时器输入	TI000, TI010
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 2 (PM2) 端口寄存器 2 (P2)

图 6-1 显示了这些计数器的框图。

图 6-1. 16 位定时器 / 事件计数器 00 的框图



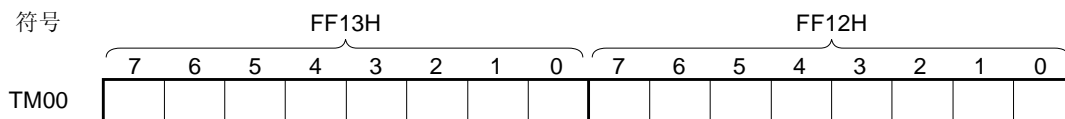
(1) 16 位定时器计数器 00 (TM00)

TM00 是一个 16 位只读寄存器，用于对计数脉冲进行计数。

计数器与计数时钟的上升沿同步累加。如果在操作过程中先读取计数值，临时停止计数时钟的输入，则将会读取该点的计数值。

图 6-2. 16 位定时计数器 00 (TM00) 的格式

地址: FF12H, FF13H 复位后: 0000H R



在以下情况中计数值会被复位为 0000H。

- <1> 如果复位信号被产生
- <2> 如果 TMC003 和 TMC002 被清除
- <3> 在 TI000 的有效沿输入时清除&开始发生的模式下，如果 TI000 引脚的有效沿被输入
- <4> 在 TM00 和 CR000 匹配时清除&开始发生的模式下，如果 TM00 和 CR000 匹配
- <5> 在单脉冲输出模式下，如果 OSPT00 被设置为 1

注意事项 1. 即使 TM00 被读取，也不能通过 CR010 来捕获。

2. 在定时器计数期间，如果 TM00 被参考，定时器计数将在参考处理期间停止，并且在参考处理完成后定时器计数重新开始。

因此，如果参考 TM00 的处理被执行，定时器计数中将出现一个错误。

(2) 16 位定时器捕获 / 比较寄存器 000 (CR000)

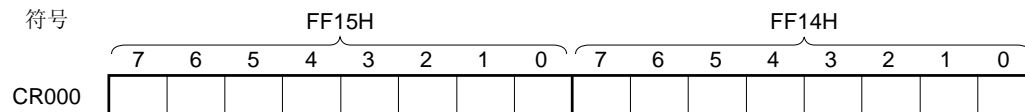
CR000 是一个具有捕获寄存器和比较寄存器功能的 16 位寄存器。它是用作捕获寄存器还是比较寄存器由捕获 / 比较控制寄存器 00 (CRC00) 的第 0 位 (CRC000) 来设置。

CR000 由 16 位存储器操作指令来设置。

复位信号生成将该 CR000 设置为 0000H。

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

地址: FF14H, FF15H 复位后: 0000H R/W



- **当 CR000 被用作比较寄存器时**

在 CR000 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 (INTTM000)。它也可以用作保持间隔时间的寄存器，然后 TM00 被设置为间隔定时器操作。

- 当 CR000 被用作捕获寄存器时

可以将 TI000 引脚或 TI010 引脚的有效沿选作捕获触发。TI000 或 TI010 的有效沿的设置通过预分频模式寄存器 00 (PRM00) 来执行 (参见表 6-2)。

表 6-2. CR000 捕获触发和 TI000 和 TI010 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)

CR000 捕获触发	TI000 引脚有效沿	ES010	ES000
		下降沿	上升沿
上升沿	下降沿	0	0
无捕获操作	双边沿	1	1

(2) TI010 引脚的有效沿被选作捕获触发 (CRC001 = 0, CRC000 = 1)

CR000 捕获触发	TI010 引脚有效沿	ES110	ES100
		下降沿	下降沿
上升沿	上升沿	0	1
双边沿	双边沿	1	1

备注

- 禁止设置 ES010, ES000 = 1, 0 和 ES110, ES100 = 1, 0。
- ES010, ES000: 预分频模式寄存器 00 (PRM00) 的第 5 和 4 位
ES110, 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 与 CR000 匹配时清除 & 开始发生的模式下, 设置 CR000 为 0000H 以外的值。这意味着当该寄存器被用作外部事件计数器时, 1 脉冲计数操作不能被执行。然而, 在自由运行和 TI000 引脚有效沿输入时清除 & 开始发生的模式下, 如果 CR000 被清除为 0000H, 当紧跟着溢出 (FFFFH) 后 CR000 的值从 0000H 变为 0001H 时, 一个中断请求 (INTTM000) 被产生。
- 如果 CR000 的新值小于 16 位定时器计数器 0 (TM00) 的值, TM00 继续计数, 溢出, 然后从 0 开始再次计数。因此, 如果 CR000 的新值小于旧的值, 在 CR000 被更改后, 定时器必须被复位来重新开始。
- 16 位定时器 / 事件计数器 00 停止后 CR000 的值将不能被保证。
- 如果捕获触发已经被输入, 捕获操作可能不会对设置为比较模式的 CR000 执行。
- 当 P21 被用作有效沿的输入引脚 (TI010) 时, 它不能用作定时器输出 (TO00)。当使用 P21 作为定时器输出引脚 (TO00) 时, 它不能用作有效沿的输入引脚 (TI010)。
- 如果当 CR000 用作捕获寄存器时寄存器读取周期和捕获触发的输入冲突, 捕获触发输入具有优先权并且读取的数据不确定。同时, 如果定时器计数停止和捕获触发输入冲突, 捕获触发不确定。
- 更改 CR000 设置可能导致故障。要更改设置, 参考 6.5 16 位定时器 / 事件计数器 00 相关的注意事项 (17) 定时器操作期间更改比较寄存器。

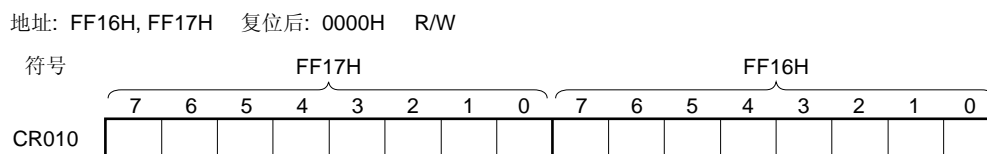
(3) 16 位定时器捕获 / 比较寄存器 010 (CR010)

CR010 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。

CR010 由 16 位存储器操作指令来设置。

复位信号生成将该 CR010 设置为 0000H。

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



- 当 CR010 被用作比较寄存器时

在 CR010 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 (INTTM010)。

- 当 CR010 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 的有效沿 (参见表 6-3)。

表 6-3. CR010 捕获触发和 TI000 引脚的有效沿 (CRC002 = 1)

CR010 捕获触发	TI000 引脚有效沿		
		ES010	ES000
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

1. 禁止设置 ES010, ES000 = 1, 0。
2. ES010, 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. 在自由运行和 TI000 引脚有效沿输入时清除 & 开始发生的模式下, 如果 CR010 被清除为 0000H, 当紧接着溢出 (FFFFH) 后 CR010 的值从 0000H 变为 0001H 时, 一个中断请求 (INTTM010) 被产生。
2. 如果 CR010 的新值小于 16 位定时器计数器 00 (TM00) 的值, TM00 继续计数, 溢出, 然后从 0 开始再次计数。因此, 如果 CR010 的新值小于旧的值, 在 CR010 被更改后, 定时器必须被复位来重新开始。
3. 16 位定时器 / 事件计数器 00 停止后 CR010 的值将不能被保证。
4. 如果捕获触发已经被输入, 捕获操作可能不会对设置为比较模式的 CR010 执行。
5. 如果当 CR010 用作捕获寄存器时寄存器读取周期和捕获触发的输入冲突, 捕获触发输入具有优先权并且读取的数据不确定。同时, 如果定时器计数停止和捕获触发的输入冲突, 捕获数据不确定。

注意事项 6. 在 **TM00** 操作期间更改 **CR010** 设置可能导致故障。要更改设置，参考 **6.5 16 位定时器 / 事件计数器 00** 相关的**注意事项 (17)** 定时器操作期间更改比较寄存器。

6.3 控制 16 位定时器 / 事件计数器 00 的寄存器

以下七类寄存器用来控制 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 2 (PM2)
- 端口寄存器 2 (P2)

(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 通过 1 位或 8 位存储器操作指令来进行设置。复位信号生成将该 TMC00 的值设置为 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) 的值时开始操作。将 TMC002 和 TMC003 设置为 0, 0 以停止操作。

图 6-5. 16 位定时器模式控制寄存器 00 (TMC00) 的格式

地址: FF60H 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 CR000 匹配或 TM00 和 CR010 匹配	< 当作为比较寄存器操作时 > TM00 和 CR000 匹配或 TM00 和 CR010 匹配时产生 < 当作为捕获寄存器操作时 > TI000 引脚和 TI010 引脚有效沿时产生
0	1	1		TM00 和 CR000 匹配, TM00 和 CR010 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 CR000 匹配时清除&开始发生	TM00 和 CR000 匹配或 TM00 和 CR010 匹配	
1	1	1		TM00 和 CR000 匹配, TM00 和 CR010 匹配或 TI000 引脚的有效沿	

OVF00	16 位定时器计数器 00 (TM00) 的溢出检测
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. 如果定时器被停止, 即使信号被输入到 TI000/TI010 引脚, 定时器计数和定时器中断也不会发生。
 3. 除了当 TI000 引脚的有效沿被选作计数时钟时, 在设置 STOP 模式或系统时钟停止模式之前停止计数器操作; 否则, 当系统时钟开始时定时器可能发生故障。
 4. 在停止定时器操作后, 使用预分频模式寄存器 00 (PRM00) 的第 4 和 5 位来设置 TI000 引脚的有效沿。
 5. 如果 TM00 和 CR000 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被选择, 当 CR000 的值为 FFFFH 并且 TM00 的值由 FFFFH 变为 0000H 时, OVF00 标志被设置为 1。
 6. 如果在 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 标志将被重新设置并且禁止清除。
 7. 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 (INTTM0n0) 发生。

备注	TM00:	16 位定时器计数器 00
	CR000:	16 位定时器捕获 / 比较寄存器 000
	CR010:	16 位定时器捕获 / 比较寄存器 010

(2) 捕获 / 比较控制寄存器 00 (CRC00)

该寄存器控制 16 位捕获 / 比较寄存器 (CR000, CR010) 的操作。

CRC00 通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该 CRC00 的值设置为 00H。

图 6-6. 捕获 / 比较控制寄存器 00 (CRC00) 的格式

地址: FF62H 复位后: 00H R/W

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

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

CRC001	CR000 捕获触发选择
0	在 TI010 引脚的有效沿上进行捕获
1	通过反相在 TI000 引脚的有效沿上进行捕获 ^注

CRC000	CR000 操作模式选择
0	作为比较寄存器进行操作
1	作为捕获寄存器进行操作

注 当 CRC001 位为 1 时, 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。

- 注意事项
1. 在设置 CRC00 之前, 定时器操作必须被停止。
 2. 使用 16 位定时器模式控制寄存器 00 (TMC00) 选择 TM00 和 CR00 匹配时清除 & 开始发生的模式时, CR000 不应该被指定为捕获寄存器。
 3. 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 (参阅图 6-17)。

(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, 允许 / 禁止 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 通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该 TOC00 的值设置为 00H。

图 6-7. 16 位定时器输出控制寄存器 00 (TOC00) 的格式

地址: FF63H 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	无单脉冲输出触发							
1	单脉冲输出触发							
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*							
TOC004	使用 CR010 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 CR000 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 CR000 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 OSPT00 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. 当 OSPT00 被连续设置为 1 时, 需要通过预分频模式寄存器 00 (PRM00) 选择的两个或更多个计数时钟周期的写入间隔。

注意事项 6. 当 TOE00 为 0 时，使用 8 位存储器操作指令同时设置 TOE00、LVS00 和 LVR00。当 TOE00 为 1 时，使用 1 位存储器操作指令设置 LVS00 和 LVR00。

(4) 预分频模式寄存器 00 (PRM00)

该寄存器被用来设置 16 为定时器计数器 00 (TM00) 的计数时钟和 TI000、TI010 引脚输入的有效沿。

PRM00 通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该 PRM00 的值设置为 00H。

图 6-8. 预分频模式寄存器 00 (PRM00) 的格式

地址: FF61H 复位后: 00H R/W

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

ES110	ES100	TI010 引脚有效沿检测
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	双边沿

ES010	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
0	0	f_{XP} (10 MHz)
0	1	$f_{XP}/2^2$ (2.5 MHz)
1	0	$f_{XP}/2^8$ (39.06 kHz)
1	1	TI000 引脚有效沿 ^注

备注 1. f_{XP} : 提供给周边硬件的时钟的振荡频率

2. (): $f_{XP} = 10 \text{ MHz}$

注 外部时钟需要一个比内部计数时钟 (f_{XP}) 的两个周期更长的脉冲。

注意事项 1. 在停止定时器操作后，总是设置数据到 PRM00。

2. 如果 TI000 引脚的有效沿将被设置为计数时钟，不要在 TI000 引脚的有效沿时设置清除 / 开始模式和捕获触发。

注意事项 3. 在以下情况下，注意 TI0n0 引脚的有效沿被检测。

<1> 系统复位后，如果高电平被输入 TI0n0 引脚，16 位定时器计数器 00 (TM00) 的操作被允许。

→ 如果上升沿和双边沿都被指定为 TI0n0 引脚的有效沿，在 TM00 操作被允许后，一个上升沿立即被检测。

<2> 如果在 TI0n0 引脚为高电平过程中 TM00 操作被停止，在低电平被输入到 TI0n0 引脚后，TM00 操作被允许。

→ 如果下降沿和双边沿都被指定为 TI0n0 引脚的有效沿，在 TM00 操作被允许后，一个下降沿立即被检测。

<3> 如果在 TI0n0 引脚为低电平过程中 TM00 操作被停止，在高电平被输入到 TI0n0 引脚后，TM00 操作被允许。

→ 如果上升沿和双边沿都被指定为 TI0n0 引脚的有效沿，在 TM00 操作被允许后，一个上升沿立即被检测。

4. 用于移除噪声的采样时钟与 TI000 的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_{XP}，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被采样并且有效电平被检测到两次使才会执行，这样可以消除短脉冲宽度的噪声。

5. 当 P21 被用作有效沿的输入引脚 (TI010) 时，它不能用作定时器输出 (TO00)。当 P21 被用作定时器输出引脚 (TO00) 时，它不能用作有效沿的输入引脚 (TI010)。

备注 n = 0, 1

(5) 端口模式寄存器 2 (PM2)

当为定时器输出使用 P21/TO00/TI010/INTP0 引脚时，应将 PM21 以及 P21 输出锁存清除。

当使用 P20/TI000/TOH1 和 P21/TO00/TI010/INTP0 引脚作为定时器输入时，设置 PM20 和 PM21 为 1。

此时，P20 和 P21 的输出锁存可以为 0 或 1。

PM2 通过 1 位或 8 位存储器操作指令来进行设置。

复位信号生成将该 PM2 的值设置为 FFH。

图 6-9. 端口模式寄存器 2 (PM2) 的格式

地址: FF22H 复位后: FFH R/W

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

PM2n	P2n 引脚 I/O 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)

6.4 16 位定时器 / 事件计数器 00 的操作

6.4.1 间隔定时器操作

按照图 6-11 的显示设置 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) 来允许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 6-10)。
- <2> 设置任意值到 CR000 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 6-10)。

注意事项 在 TM00 操作期间更改 CR000 设置可能导致故障。要更改设置, 参考 6.5 16 位定时器 / 事件计数器 00 相关的注意事项 (17) 定时器操作期间更改比较寄存器。

备注 关于允许 INTTM000 中断的方法, 请参见第九章 中断功能。

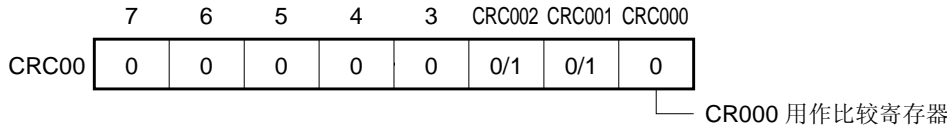
使用 16 位定时器捕获 / 比较寄存器 000 (CR000) 中预置的计数值作为间隔, 中断请求将重复产生。

当 16 位定时器计数器 00 (TM00) 的计数值与 CR000 中的设置值匹配时, TM00 值被清除为 0 同时计数继续, 并且中断请求信号 (INTTM000) 被产生。

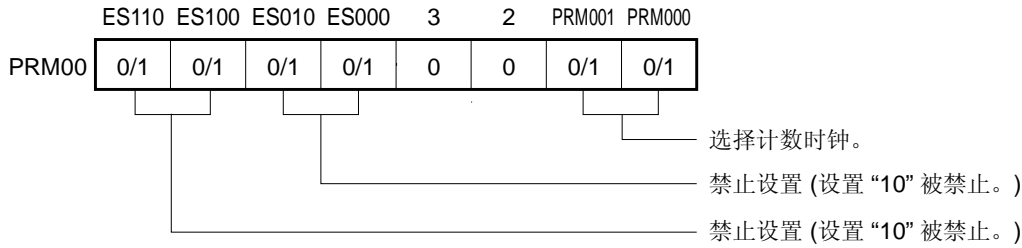
16 位定时器 / 事件计数器的计数时钟可以使用预分频模式寄存器 00 (PRM00) 的第 0 和 1 位 (PRM000, PRM001) 来选择。

图 6-10. 间隔定时器操作的控制寄存器设置

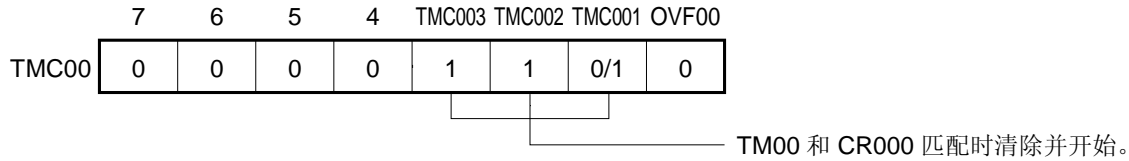
(a) 16 位定时器模式控制寄存器 00 (CRC00)



(b) 预分频模式寄存器 00 (PRM00)

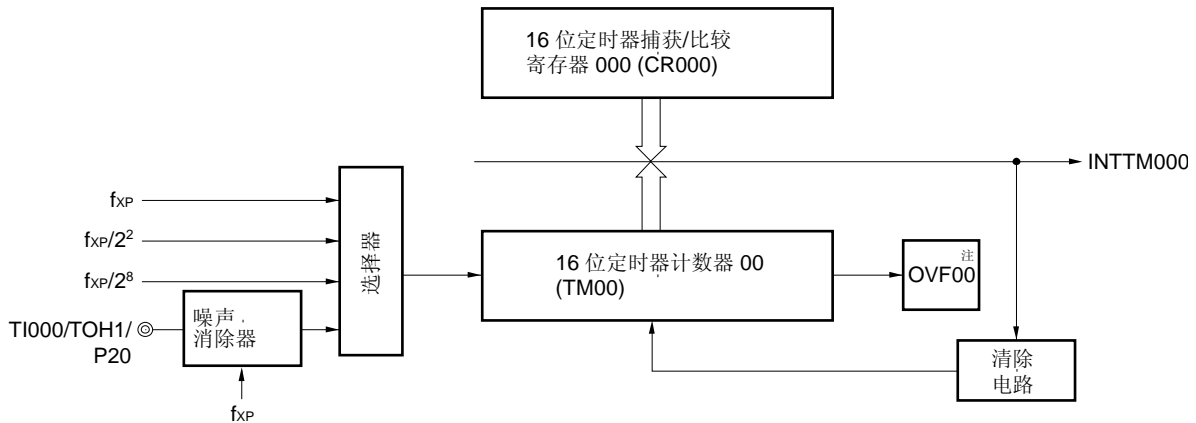


(c) 16 位定时器模式控制寄存器 00 (TMC00)



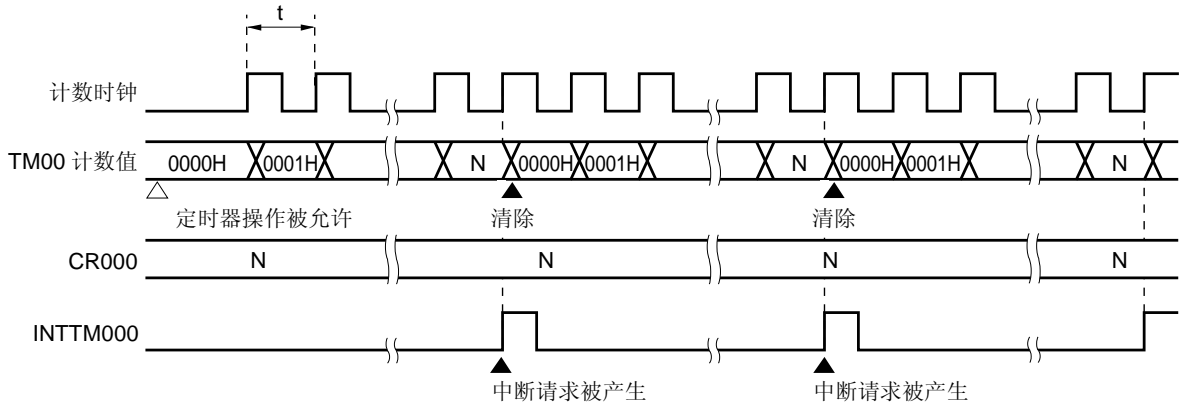
备注 0/1: 设置 0 或 1 来允许另一个功能与间隔定时器同时使用。关于详细情况, 参见各自控制寄存器的说明。

图 6-11. 间隔定时器配置图



注 只有当 16 位定时器捕获 / 比较寄存器 000 被设置为 FFFFH 时, OVF00 才被设置为 1。

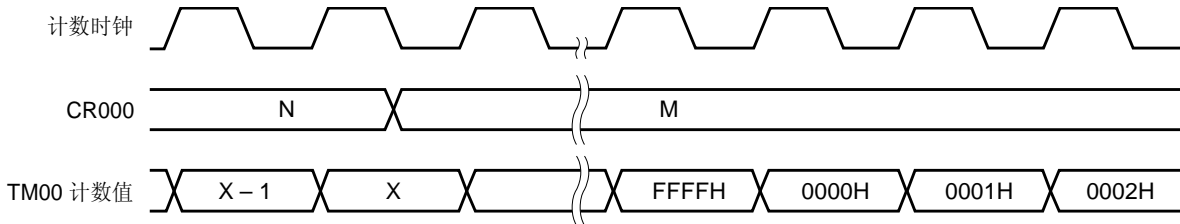
图 6-12. 间隔定时器操作的时序



备注 间隔时间 = $(N + 1) \times t$
 $N = 0001H$ 到 $FFFFH$ (可设置范围)

当比较寄存器在定时器计数操作期间被更改时，如果 16 位定时器捕获 / 比较寄存器 000 (CR000) 被更改后的值比 16 位定时器计数器 00 (TM00) 的值小，TM00 继续计数，溢出并且从 0 重新开始。因此，如果 CR000 更改后的值 (M) 比更改前的值 (N) 小，需要在更改 CR000 后重新开始定时器。

图 6-13. 定时器计数操作期间比较寄存器更改后的时序 (N → M: N > M)



备注 $N > X > M$

6.4.2 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 6-14)。
- <2> 使用 PRM00 寄存器来设置计数时钟。
- <3> 设置任意值到 CR000 寄存器 (0000H 不能被设置)。
- <4> 设置 TMC00 寄存器来开始操作 (关于设置值, 参见图 6-14)。

备注 1. 关于 TI000 引脚设置的详细情况, 请参见 6.3 (5) 端口模式寄存器 2 (PM2)。
 2. 关于允许 INTTM000 中断的方法, 请参见第九章 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) 指定的有效沿被输入时, TM00 增加。

当 TM00 计数值与 16 位定时器捕获 / 比较寄存器 000 (CR000) 的值匹配时, TM00 被清除为 0 并且中断请求信号 (INTTM000) 被产生。

输入 0000H 以外的值到 CR000。(一个脉冲的计数操作被执行。)

上升沿、下降沿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 ES010) 来选择。

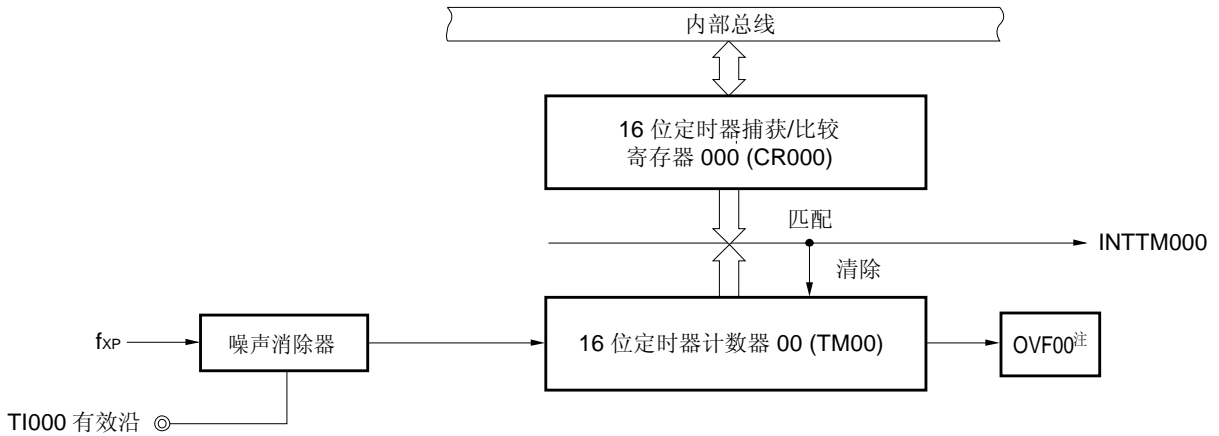
因为只有当使用内部时钟 (fxP) 采样后 TI000 有效沿被检测到两次时, 操作才能执行, 这样可以消除短脉冲的噪声。

图 6-14. 外部事件计数器模式下的控制寄存器设置 (指定上升沿)



备注 0/1: 设置 0 或 1 来允许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。

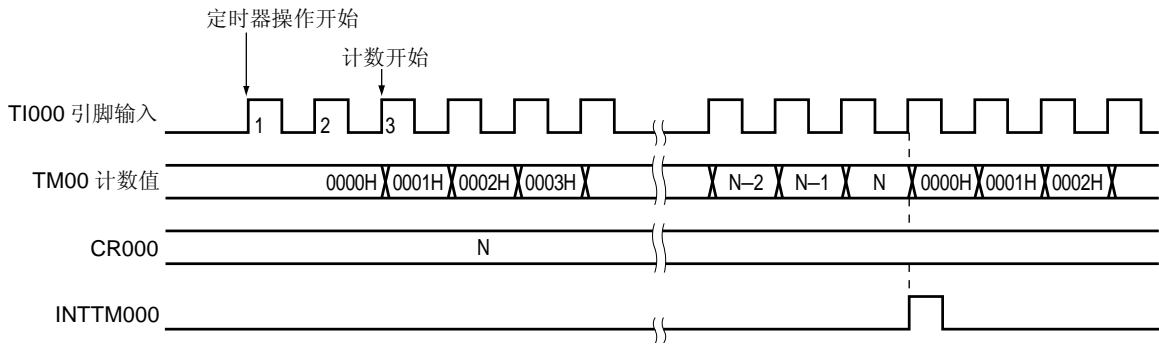
图 6-15. 外部事件计数器的配置图



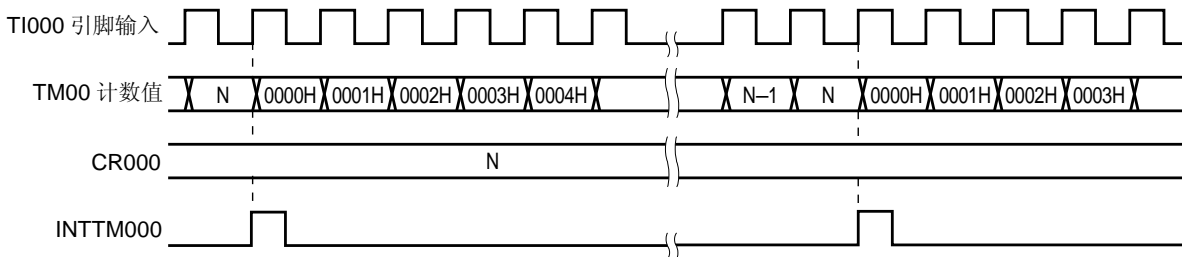
注 只有当 16 位定时器捕获 / 比较寄存器 000 被设置为 FFFFH 时，OVF00 才被设置为 1。

图 6-16. 外部事件计数器操作时序（指定上升沿）

(1) 操作开始后 INTTM000 的产生时序
有效沿被检测两次后计数开始。



(2) INTTM000 被产生两次后 INTTM000 的产生时序



注意事项 当读取外部事件计数器计数值时，TM00 应该被读取。

6.4.3 脉冲宽度测量操作

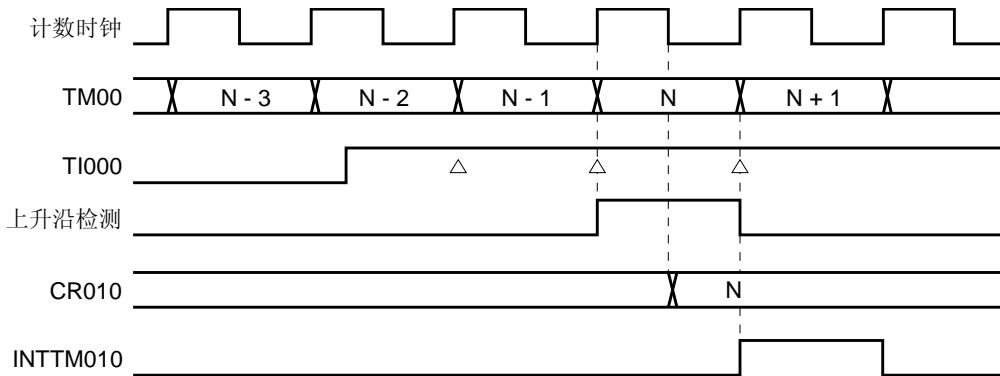
可以使用 16 位定时器计数器 00 (TM00) 来测量输入到 TI000 引脚和 TI010 引脚的信号的脉冲宽度。

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 引脚的信号的有效沿同步的定时器来测量。

当中断发生时，通过读取捕获寄存器的有效值，需要的脉冲宽度可以计算。

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 TI010 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（参见图 6-17）。

图 6-17. 指定上升沿的 CR010 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 6-18、6-21、6-23 和 6-25）。
- <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 6-18、6-21、6-23 和 6-25）。

注意事项 要使用两个捕获寄存器，设置 TI000 设置 TI010 引脚。

- 备注**
1. 关于 TI000（或 TI010）引脚设置的详细情况，请参见 6.3（5） 端口模式寄存器 2（PM2）。
 2. 关于允许 INTTM000（或 INTTM010）中断的方法，请参见第九章 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

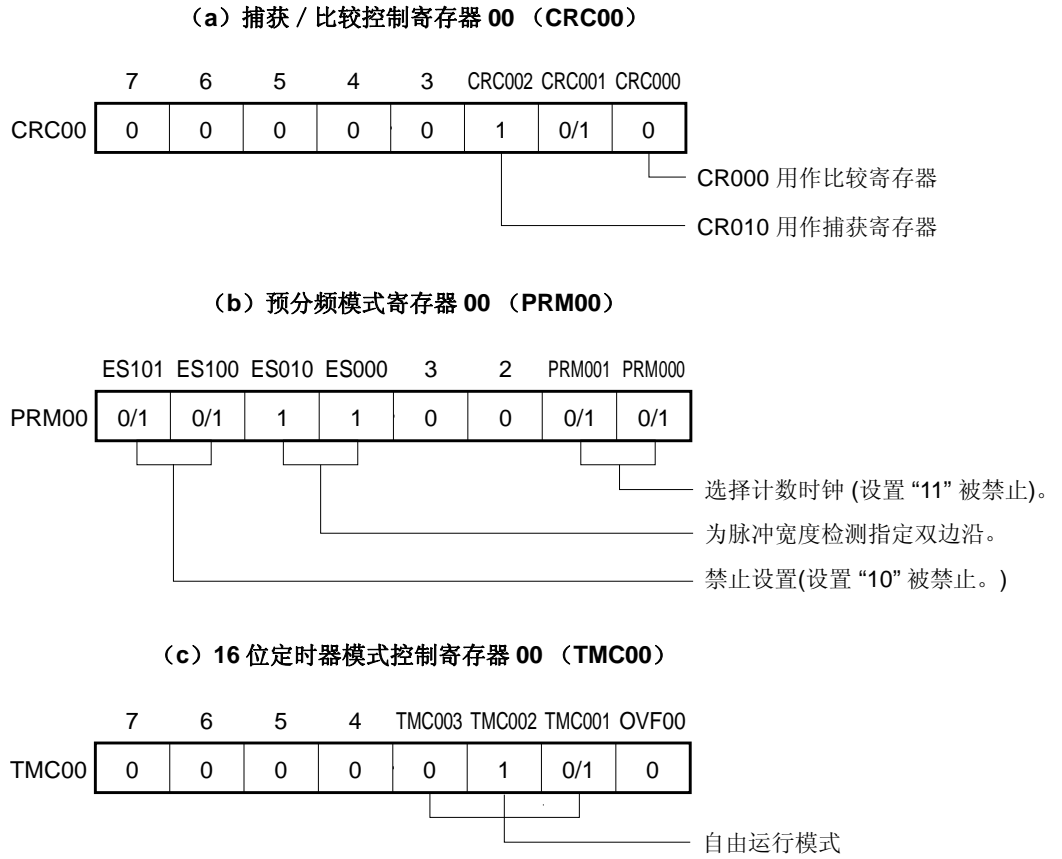
通过使用 PRM00 的第 4 和 5 位（ES000 和 ES010）来指定双边沿和下降沿作为 TI000 引脚有效沿。

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 PRM00 指定的有效沿被输入时，TM00 的值被放入 16 位定时器捕获 / 比较寄存器 010 (CR010) 并且一个外部中断请求信号 (INTTM010) 被设置。

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。

注意事项 该操作示例中可测量的脉冲宽度达到定时器计数器的 1 个周期。

图 6-18. 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 CR010 被使用时)



备注 0/1: 设置 0 或 1 来允许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 6-19. 使用自由运行计数器的脉冲宽度测量的配置图

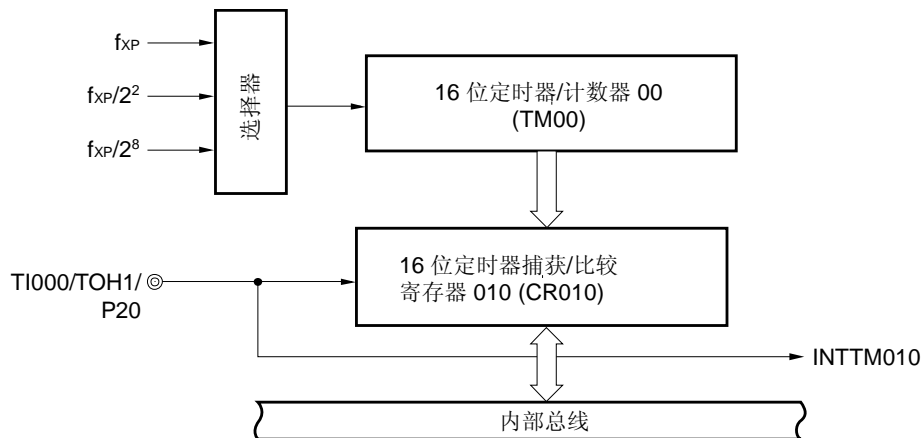
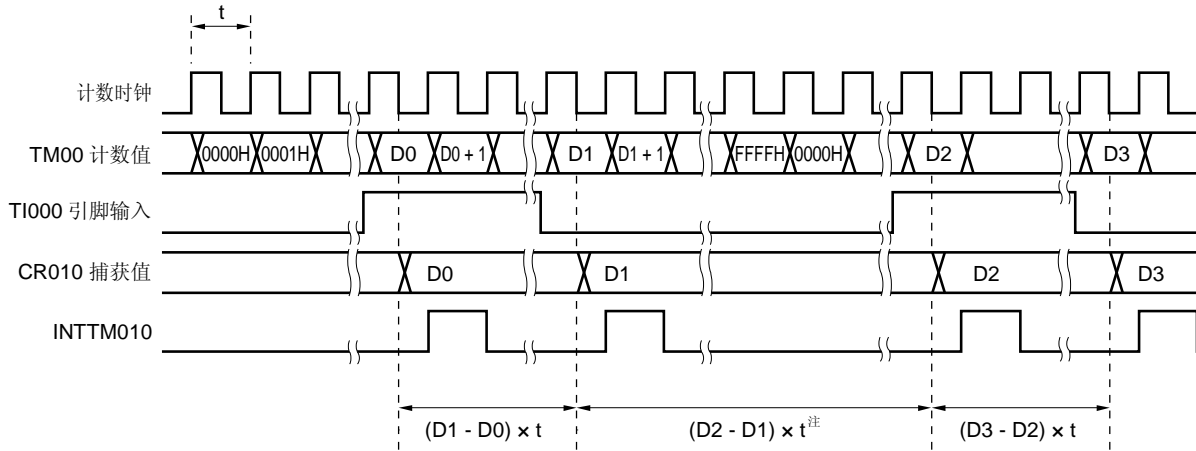


图 6-20. 使用自由运行计数器和一个捕获寄存器的脉冲宽度测量操作的时序图（指定双边沿）



注 进位标志被设置为 1。忽略该设置。

(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，可以同时测量输入到 TI000 引脚和 TI010 引脚的两个信号的脉冲宽度。

通过使用 PRM00 的第 4 和 5 位（ES000 和 ES010）以及第 6 和 7 位（ES100 和 ES110）来指定上升沿和下降沿作为 TI000 和 TI010 的有效沿。

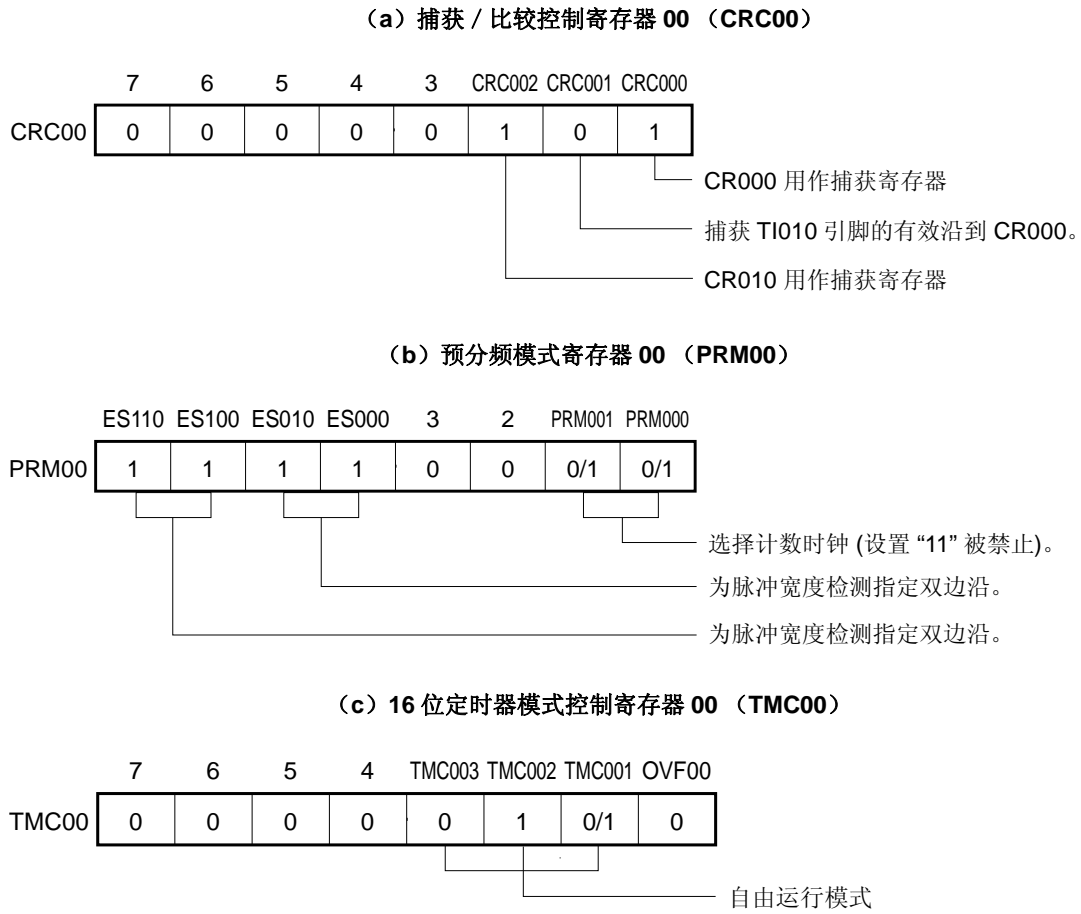
当通过预分频模式寄存器 00（PRM00）的第 4 和 5 位（ES000 和 ES010）指定的边沿输入到 TI000 引脚时，TM00 的值被放入 16 位定时器捕获 / 比较寄存器 010（CR010）并且一个中断请求信号（INTTM010）被设置。

同时，当通过预分频模式寄存器 00（PRM00）的第 6 和 7 位（ES100 和 ES110）指定的边沿输入到 TI010 引脚时，TM00 的值被放入 16 位定时器捕获 / 比较寄存器 000（CR000）并且一个中断请求信号（INTTM000）被设置。

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 TI010 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。

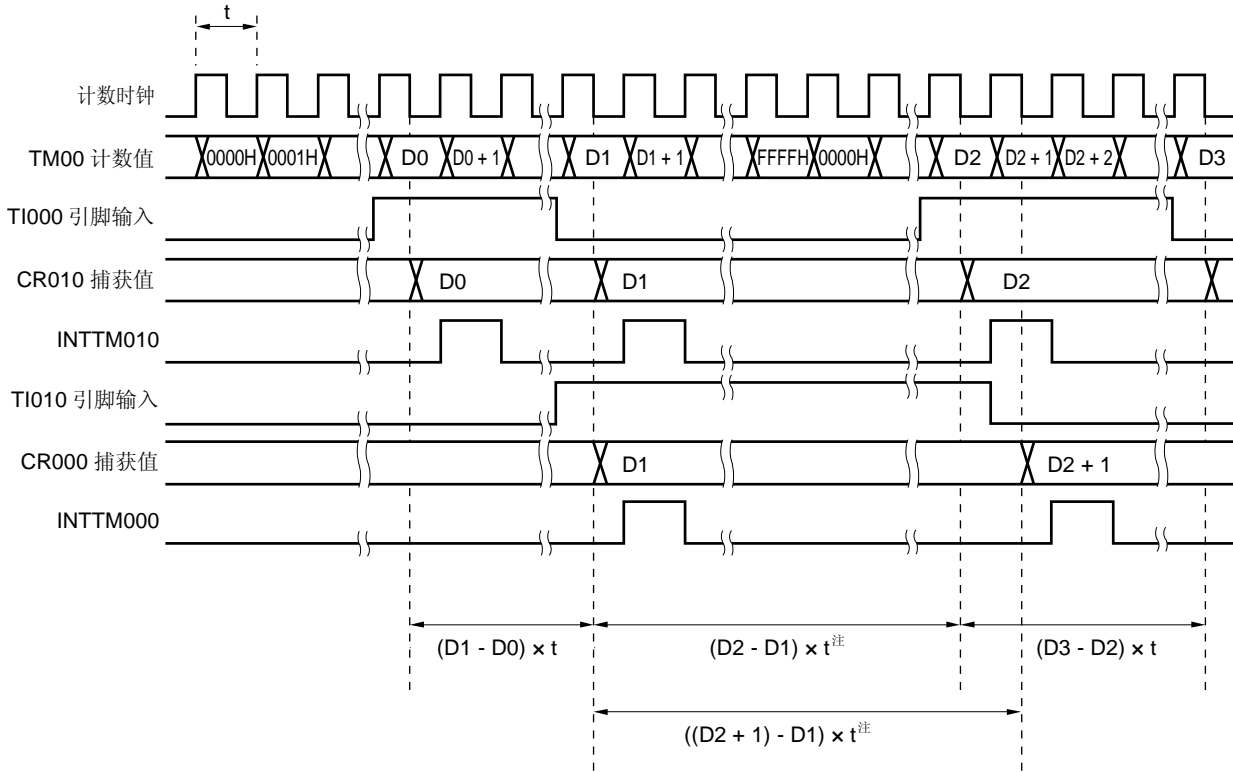
注意事项 该操作示例中可测量的脉冲宽度达到定时器计数器的 1 个周期。

图 6-21. 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 来允许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 6-22. 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图
(指定双边沿)



注 进位标志被设置为 1。忽略该设置。

(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。

通过使用 PRM00 的第 4 和 5 位 (ES000 和 ES010) 来指定双边沿和下降沿作为 TI000 引脚有效沿。

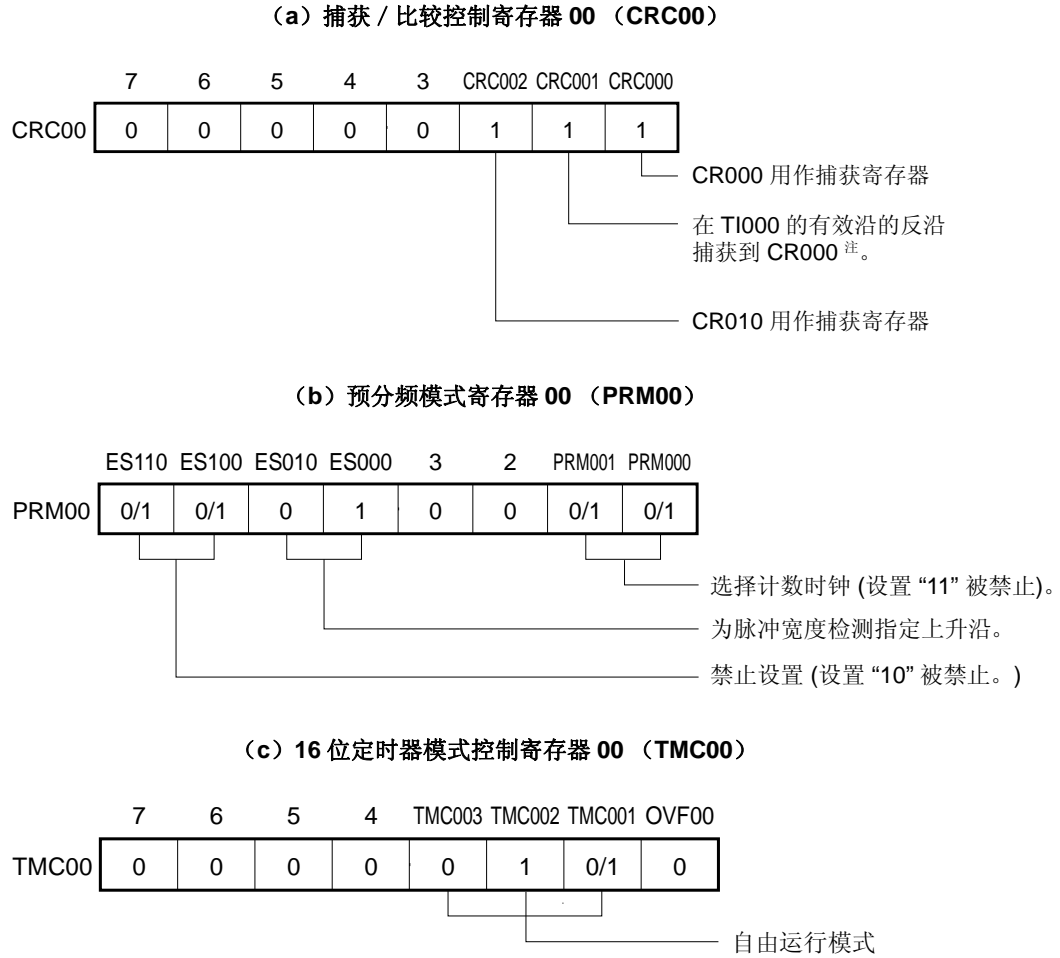
当通过预分频模式寄存器 00 (PRM00) 的第 4 和 5 位 (ES000 和 ES010) 指定的上升沿或下降沿输入到 TI000 引脚时, TM00 的值被放入 16 位定时器捕获 / 比较寄存器 010 (CR010) 并且一个中断请求信号 (INTTM010) 被设置。

同时, 当捕获操作的反向边沿输入到 CR010 时, TM00 的值被放入 16 位定时器捕获 / 比较寄存器 000 (CR000)。

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。

注意事项 该操作示例中可测量的脉冲宽度达到定时器计数器的 1 个周期。

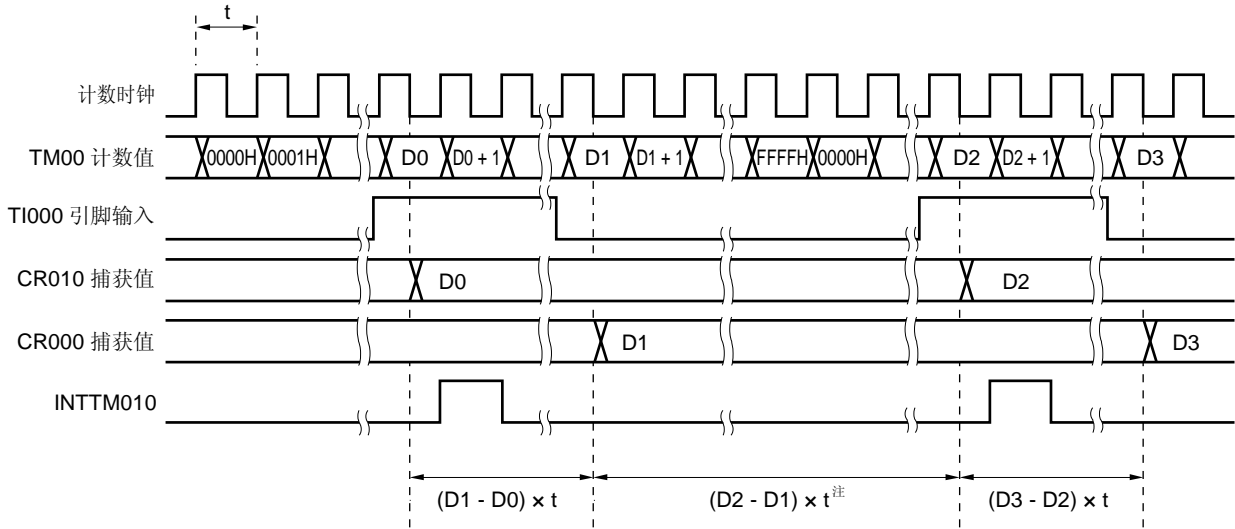
图 6-23. 使用自由运行计数器和两个捕获寄存器的脉冲宽度测量控制寄存器设置
(指定上升沿)



注 如果 TI000 的有效沿被指定为双边沿, 16 位定时器捕获 / 比较寄存器 000 (CR000) 不能执行捕获操作。当 CRC001 位的值为 1 时, 当 TI010 引脚的有效沿被检测时, TM00 的计数值不会被不会到 CR000 寄存器中, 但是从 TI010 引脚的输入可以被用作外部中断源, 因为 INTTM000 在那是被产生。

备注 0/1: 设置 0 或 1 来允许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 6-24. 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图
(指定双边沿)



注 进位标志被设置为 1。忽略该设置。

(4) 通过重启方法的脉冲宽度测量

通过使用 PRM00 的第 4 和 5 位 (ES000 和 ES010) 来指定双边沿和下降沿作为 TI000 引脚的有效沿。

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 010 (CR010), 然后输入到 TI000 引脚的信号的脉冲宽度通过清除 TM00 和重启计数被测量。

通过使用预分频模式寄存器 00 (PRM00) 的第 4 和 5 位 (ES000 和 ES010), 边沿指定可以从两种类型, 上升沿或下降沿, 中选择。

采样使用预分频模式寄存器 00 (PRM00) 选择的间隔来执行, 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。

注意事项 该操作示例中可测量的脉冲宽度达到定时器计数器的 1 个周期。

图 6-25. 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) (1 / 2)

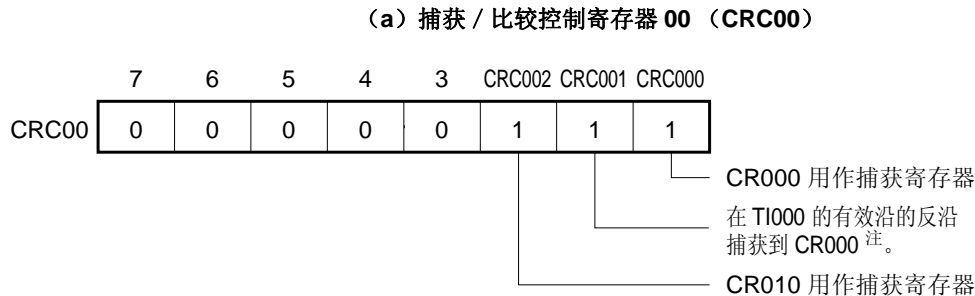
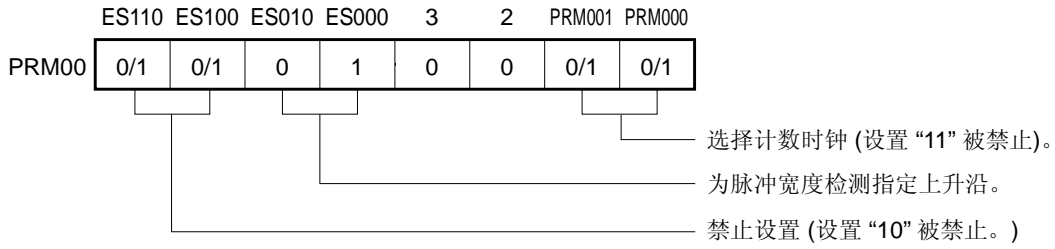
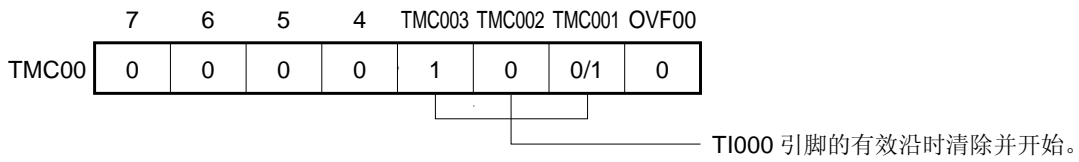


图 6-25. 通过重启方法的脉冲宽度测量的控制寄存器设置
(指定上升沿) (2 / 2)

(b) 预分频模式寄存器 00 (PRM00)

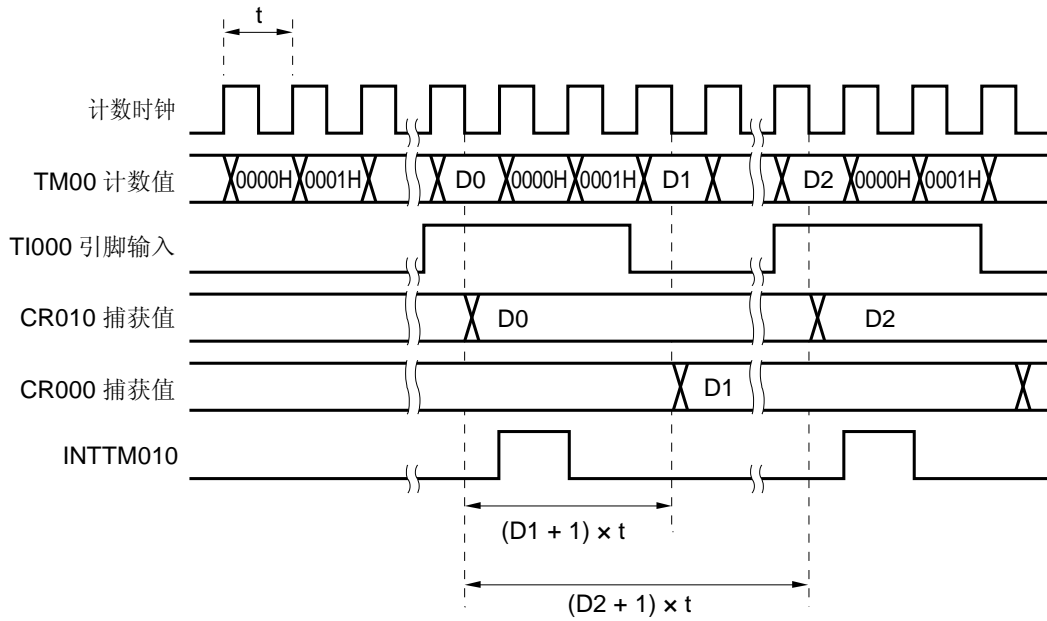


(c) 16 位定时器模式控制寄存器 00 (TMC00)



注 如果 TI000 的有效沿被指定为双边沿, 16 位定时器捕获 / 比较寄存器 000 (CR000) 不能执行捕获操作。

图 6-26. 通过重启方法的脉冲宽度测量操作的时序
(指定上升沿)



6.4.4 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 6-27)。
- <3> 设置 TOC00 寄存器 (关于设置值, 参见图 6-27)。
- <4> 设置任意值到 CR000 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 6-27)。

注意事项 在 TM00 操作期间更改 CR000 设置可能导致故障。要更改设置, 参考 6.5 16 位定时器 / 事件计数器 00 相关的注意事项 (17) 定时器操作期间更改比较寄存器。

- 备注**
- 1. 关于 TO00 引脚设置的详细情况, 请参见 6.3 (5) 端口模式寄存器 2 (PM2)。
 - 2. 关于允许 INTTM000 中断的方法, 请参见第九章 中断功能。

任意频率的方波可以以预置到 16 位定时器捕获 / 比较寄存器 000 (CR000) 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 CR000 的计数值 + 1 确定的间隔反转。这样就运行任意频率的方波被输出。

图 6-27. 方波输出模式下的控制寄存器设置 (1 / 2)

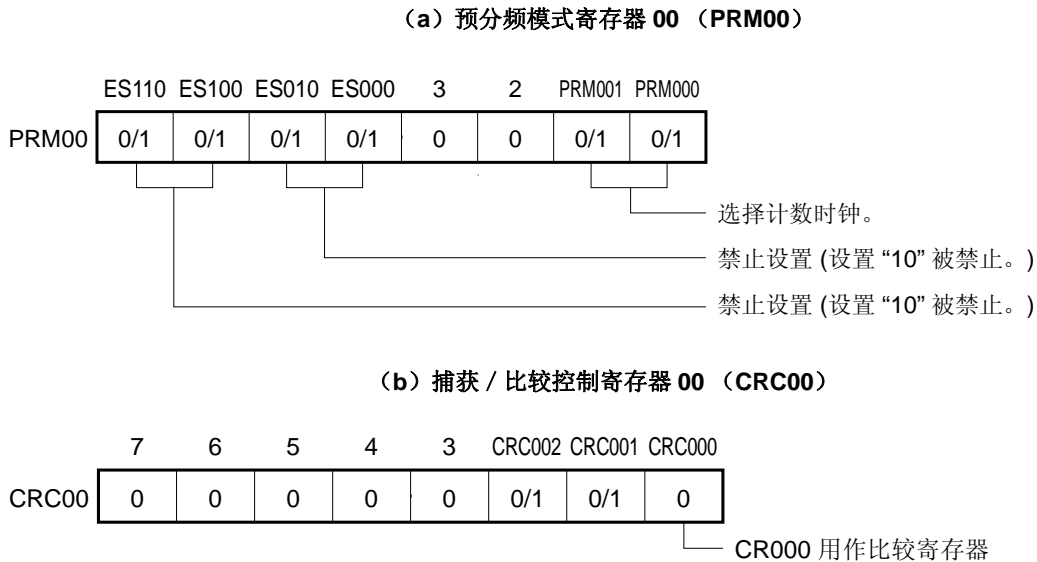
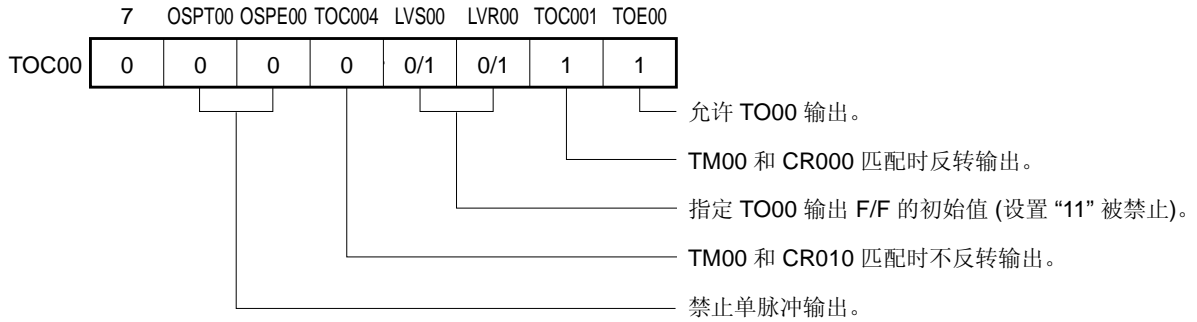
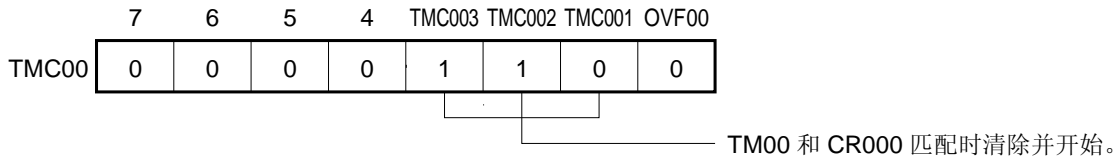


图 6-27. 方波输出模式下的控制寄存器设置 (2 / 2)

(c) 16 位定时器输出控制寄存器 00 (TOC00)

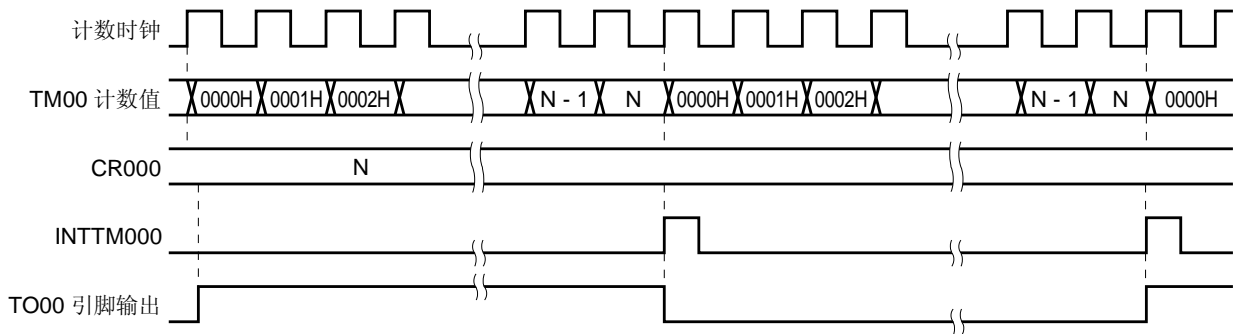


(d) 16 位定时器模式控制寄存器 00 (TMC00)



备注 0/1: 设置 0 或 1 来允许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 6-28. 方波输出操作时序



6.4.5 PPG 输出操作

按照图 6-30 的显示设置 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) 来允许作为 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 6-29)。
- <2> 设置任意值到 CR000 寄存器作为周期。
- <3> 设置任意值到 CR010 寄存器作为占空比。
- <4> 设置 TOC00 寄存器 (关于设置值, 参见图 6-29)。
- <5> 使用 PRM00 寄存器来设置计数时钟。
- <6> 设置 TMC00 寄存器来开始操作 (关于设置值, 参见图 6-29)。

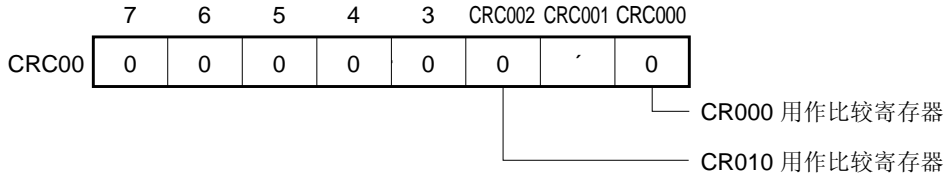
注意事项 在 TM00 操作期间更改 CRC0n0 设置可能导致故障。要更改设置, 参考 6.5 16 位定时器 / 事件计数器 00 相关的注意事项 (17) 定时器操作期间更改比较寄存器。

- 备注**
1. 关于 TO00 引脚设置的详细情况, 请参见 6.3 (5) 端口模式寄存器 2 (PM2)。
 2. 关于允许 INTTM000 中断的方法, 请参见第九章 中断功能。
 3. $n = 0$ 或 1

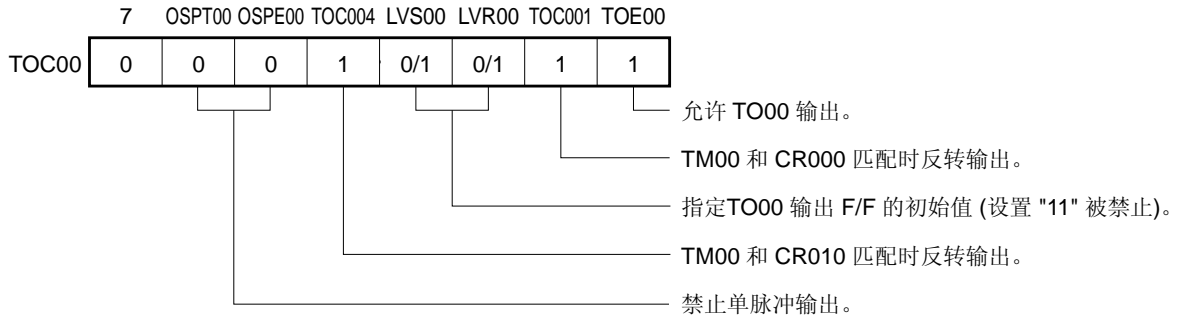
在 PPG 输出模式中, 方波从 TO00 引脚输出, 它的周期和占空比分别对应 16 位定时器捕获 / 比较寄存器 010 (CR010) 和 16 位定时器捕获 / 比较寄存器 000 (CR000) 中预置的计数值。

图 6-29. PPG 输出操作的控制寄存器设置

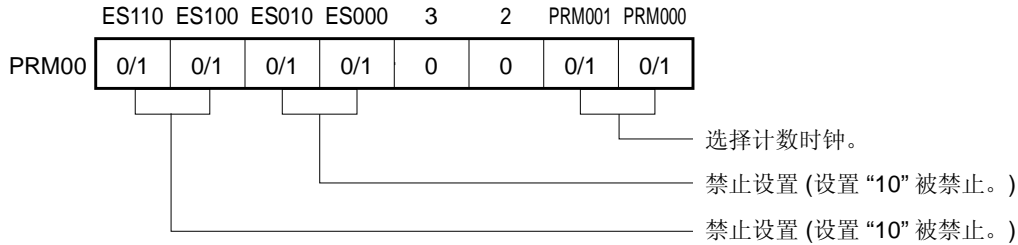
(a) 捕获 / 比较控制寄存器 00 (CRC00)



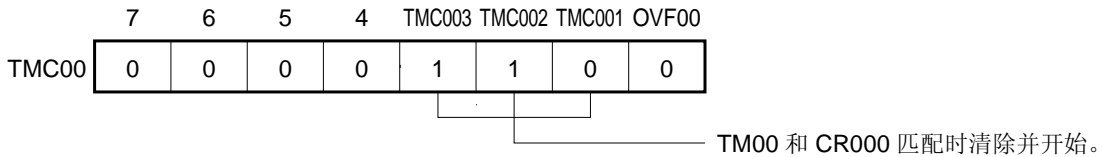
(b) 16 位定时器输出控制寄存器 00 (TOC00)



(c) 预分频模式寄存器 00 (PRM00)



(d) 16 位定时器模式控制寄存器 00 (TMC00)



注意事项 1. 以下范围内的值应该被设置到 CR000 和 CR010 中:

$$0000H < CR010 < CR000 \leq FFFFH$$

2. 通过 PPG 输出产生的脉冲的周期 (CR000 设置值 + 1) 具有 (CR010 设置值 + 1) / (CR000 设置值 + 1) 的占空比。

备注 x: 不关注

图 6-30. PPG 输出操作的配置图

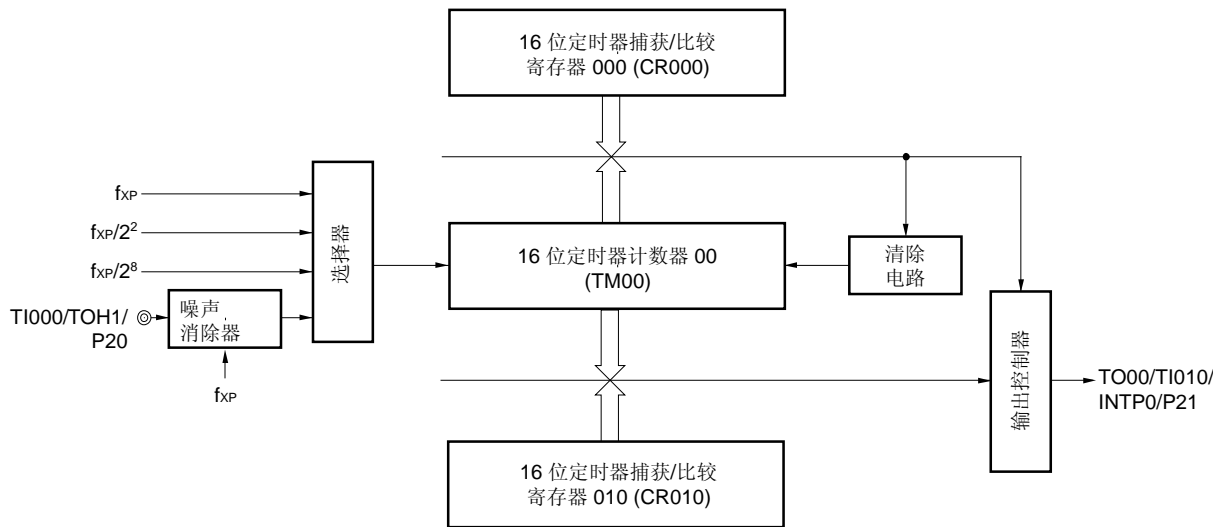
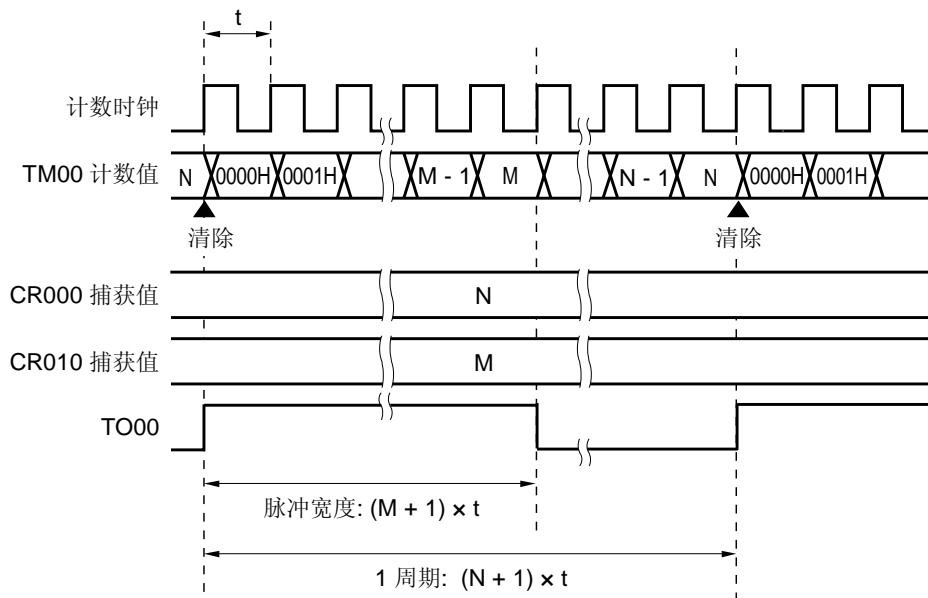


图 6-31. PPG 输出操作时序



备注 0000H < M < N ≤ FFFFH

6.4.6 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 6-32 和 6-34)。
- <3> 设置 TOC00 寄存器 (关于设置值, 参见图 6-32 和 6-34)。
- <4> 设置任意值到 CR000 和 CR010 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 6-32 和 6-34)。

- 备注**
1. 关于 TO00 引脚设置的详细情况, 请参见 6.3 (5) 端口模式寄存器 2 (PM2)。
 2. 关于允许 INTTM000 (如果需要, INTTM010) 中断的方法, 请参见第九章 中断功能。

(1) 软件触发的单脉冲输出

通过按照图 6-33 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), 并且通过软件设置 TOC00 寄存器的第 6 位 (OSPT00) 为 1, 一个单脉冲可以从 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 010 (CR010) 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 000 (CR000) * 中的计数值 (M) 时变为无效。

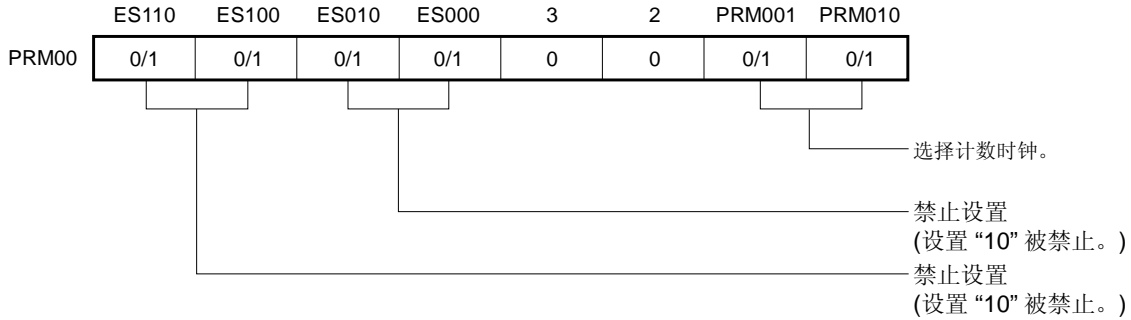
即使在单脉冲被输出后, TM00 寄存器也会继续其操作。要停止 TM00 寄存器, TMC00 寄存器的 TMC003 和 TMC002 位必须被清除为 00。

注 这是 $N < M$ 的情况。当 $N > M$ 时, 输出按照 CR000 寄存器变为有效并且按照 CR010 寄存器变为无效。不要设置 N 为 M。

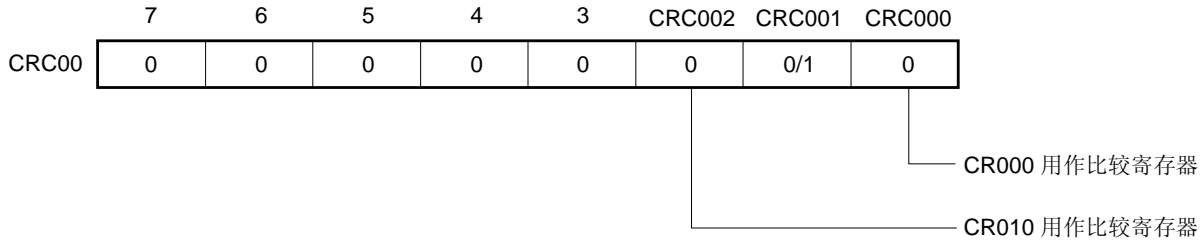
- 注意事项**
1. 单脉冲输出期间不要再次设置 OSPT00 位为 1。要再次输出单脉冲, 则应等待直到当前单脉冲输出完成后。
 2. 当使用软件触发的 16 位定时器 / 事件计数器 00 的单脉冲输出时, 不要更改 TI000 引脚或其替换功能端口引脚的电平。
因为外部触发在这种情况下有效, 在 TI000 引脚或其替换功能端口引脚的电平时定时器被清除并启动, 导致脉冲以不确定的时序输出。

图 6-32. 软件触发单脉冲输出的控制寄存器设置

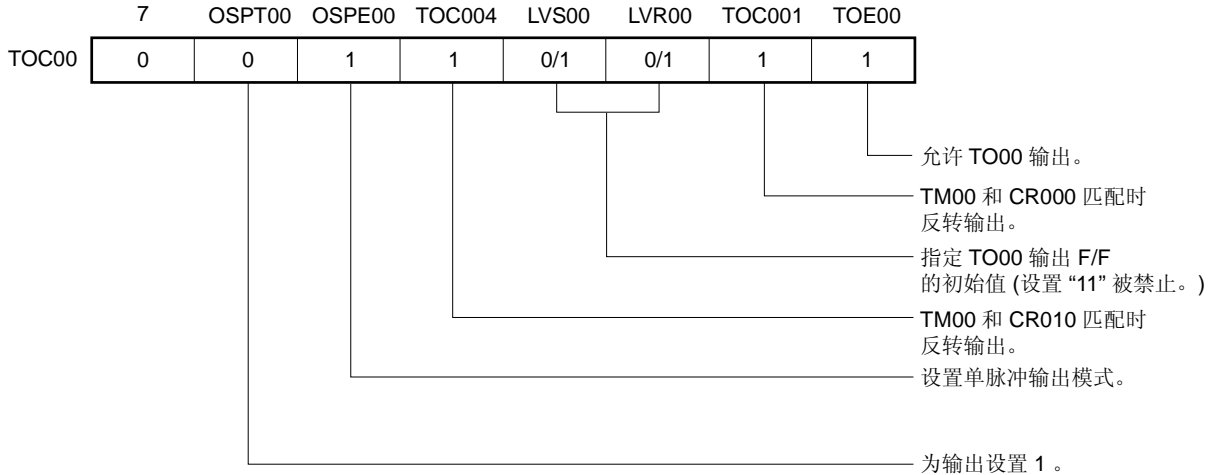
(a) 预分频模式寄存器 00 (PRM00)



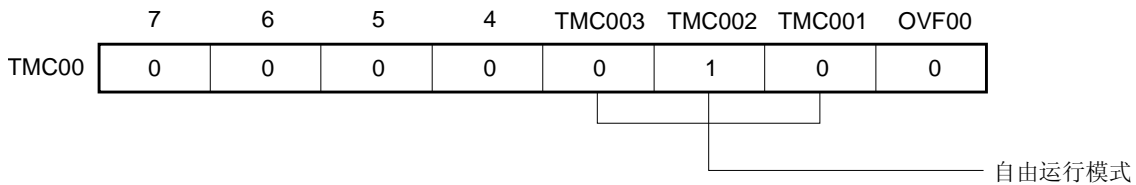
(b) 捕获 / 比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)

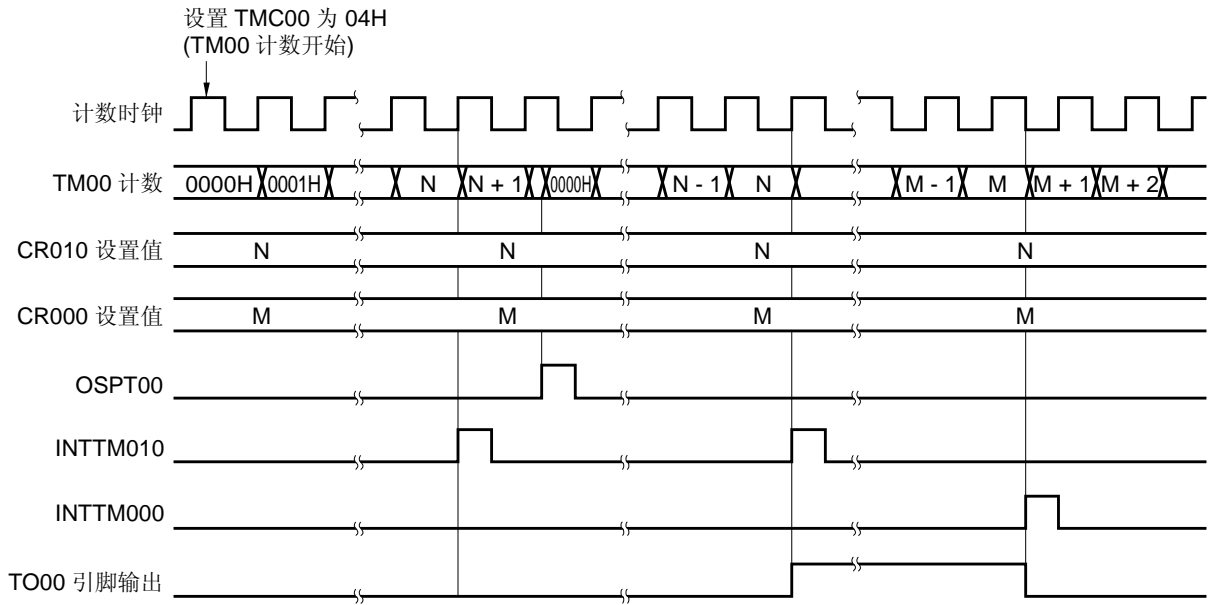


(d) 16 位定时器模式控制寄存器 00 (TMC00)



注意事项 不要设置 0000H 到 CR000 和 CR010 寄存器。

图 6-33. 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 6-35 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, ES010) 来进行设置。上升沿、下降沿或双边沿可以被指定。

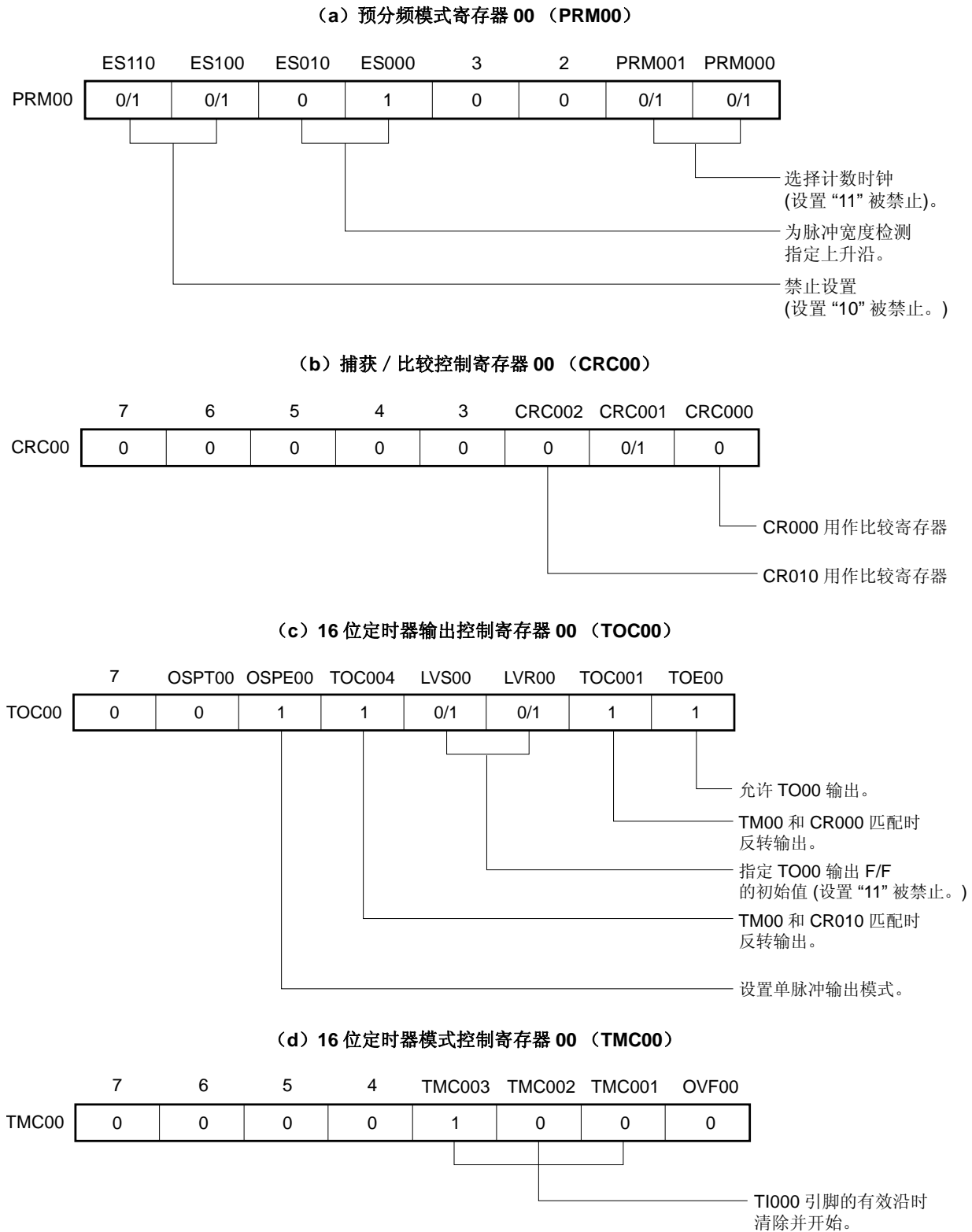
当 TI000 因的有效沿被检测时, 16 位定时器 / 事件计数器被清除并启动, 并且其输出在达到事先设置到 16 位定时器捕获 / 比较寄存器 010 (CR010) 中的计数值时变为有效。然后, 输出在达到事先设置到 16 位定时器捕获 / 比较寄存器 000 (CR000) 中的计数值时变为无效。

注 这是 $N < M$ 的情况。当 $N > M$ 时, 输出按照 CR000 寄存器变为有效并且按照 CR010 寄存器变为无效。不要设置 N 为 M 。

注意事项 单脉冲输出期间不要再次输入外部触发。

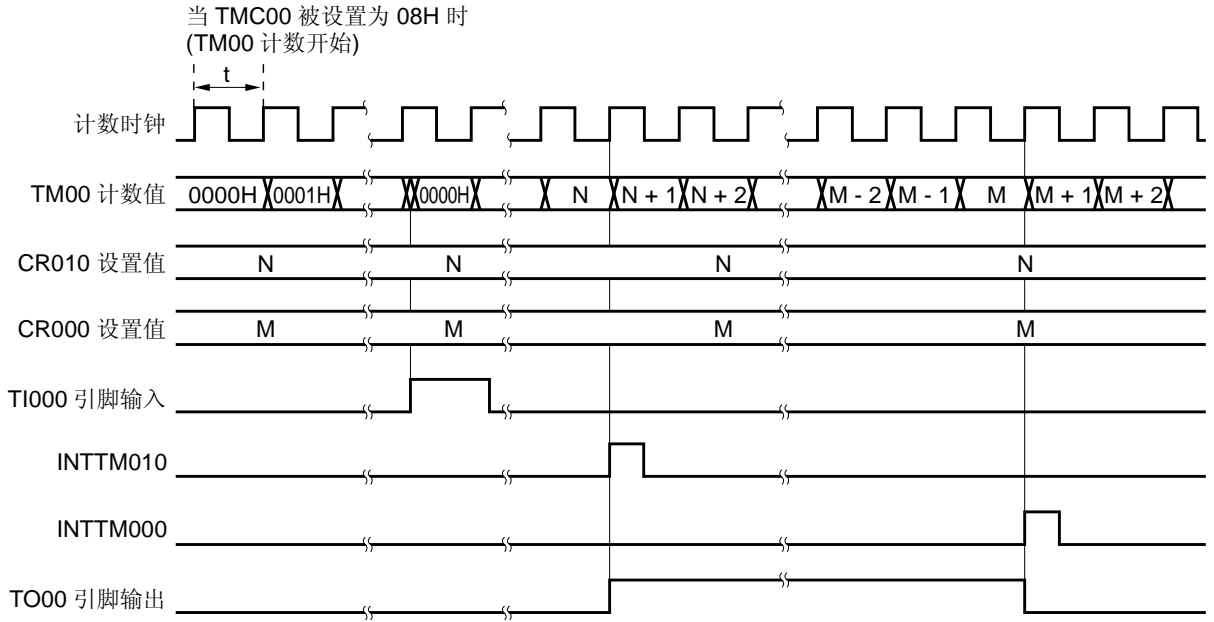
要再次输出单脉冲, 则应等待直到当前单脉冲输出完成后。

图 6-34. 外部触发单脉冲输出的控制寄存器设置 (指定上升沿)



注意事项 不要设置 0000H 到 CR000 和 CR010 寄存器。

图 6-35. 外部触发单脉冲输出操作的时序（指定上升沿）



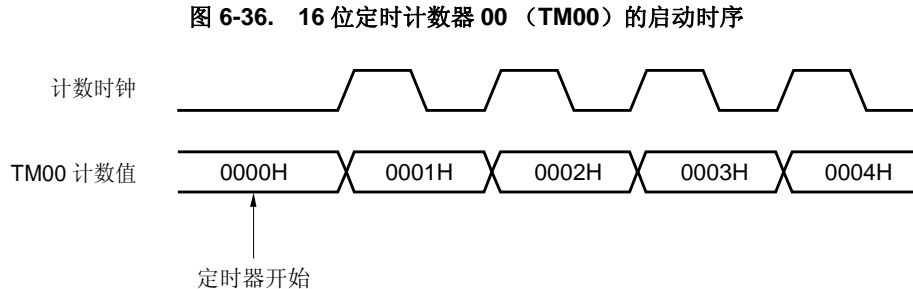
注意事项 只要 00（操作停止模式）以外的值被设置到 TMC002 和 TMC003 位，16 位定时器计数器 00 就开始操作。

备注 N < M

6.5 16 位定时器 / 事件计数器 00 相关的注意事项

(1) 定时器启动时的错误

在定时器启动后生成匹配信号所需的时间内至少一个时钟可能会发生错误。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 位定时器计数器 00 (TM00) 操作

- <1> 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) 的值时开始操作。将 TMC002 和 TMC003 设置为 0, 0 以停止操作。
- <2> 如果 TM00 被读取, 值不会被 16 位定时器捕获 / 比较寄存器 010 (CR010) 捕获。
- <3> 在定时器计数期间, 如果 TM00 被参考, 定时器计数将在参考处理期间停止, 并且在参考处理完成后定时器计数重新开始。
因此, 如果参考 TM00 的处理被执行, 定时器计数中将出现一个错误。
- <4> 如果定时器被停止, 即使信号被输入到 TI000/TI010 引脚, 定时器计数和定时器中断也不会发生。

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

- <1> 在 TM00 和 CR000 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 000 (CR000) 为 0000H 以外的值。这意味着当该寄存器被用作外部事件计数器时, 1 脉冲计数操作不能被执行。
- <2> 当 TM00 和 CR000 匹配时清除 & 开始发生的模式被选择时, CR000 不应该被指定为捕获寄存器。
- <3> 在自由运行和 TI000 引脚有效沿输入时清除 & 开始发生的模式下, 如果 CR0n0 被清除为 0000H, 当紧跟着溢出 (FFFFH) 后 CR0n0 的值从 0000H 变为 0001H 时, 一个中断请求 (INTTM0n0) 被产生。
- <4> 如果 CR0n0 的新值小于 TM00 的值, TM00 继续计数, 溢出, 然后从 0 开始再次计数。因此, 如果 CR0n0 的新值小于旧的值, 在 CR0n0 被更改后, 定时器必须被复位来重新开始。

(4) 捕获寄存器数据保持

在 16 位定时器 / 事件计数器 00 被停止后，16 位定时器捕获 / 比较寄存器 0n0 (CR0n0) 的值不被保证。

备注 n = 0, 1

(5) 16 位定时器模式控制寄存器 00 (TMC00) 的设置

在重写 OVF00 标志以外的位之前，定时器操作必须被停止。

(6) 捕获 / 比较控制寄存器 00 (CRC00) 的设置

在设置 CRC00 之前，定时器操作必须被停止。

(7) 16 位定时器输出控制寄存器 00 (TOC00) 的设置

<1> 在设置 OSPT00 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> 当 OSPT00 被连续设置为 1 时，需要通过预分频模式寄存器 00 (PRM00) 选择的两个或更多个计数时钟周期的写入间隔。

(8) 预分频模式寄存器 00 (PRM00) 的设置

在停止定时器操作后，总是设置数据到 PRM00。

(9) 有效沿设置

在停止定时器操作后，使用预分频模式寄存器 00 (PRM00) 的第 4 和 5 位 (ES000 and ES010) 来设置 TI000 引脚的有效沿。

(10) 单脉冲输出

单脉冲输出只有在自由运行模式或 TI000 引脚有效沿时清除&开始发生的模式中才能正常进行。因为在 TM00 寄存器和 CR000 寄存器匹配时清除&开始发生的模式中溢出不会发生，所以单脉冲不能输出。

(11) 通过软件的单脉冲输出

<1> 单脉冲输出期间不要再次设置 OSPT00 位为 1。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。

因为外部触发在这种情况下有效，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，导致脉冲以不确定的时序输出。

<3> 不要设置 16 位定时器捕获 / 比较寄存器 000 和 010 (CR000 和 CR010) 为 0000H。

(12) 外部触发的单脉冲输出

<1> 单脉冲输出期间不要再次输入外部触发。
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> 不要设置 16 位定时器捕获 / 比较寄存器 000 和 010 (CR000 和 CR010) 为 0000H。

(13) OVF00 标志的操作

<1> OVF00 标志在下列情况下被设置为 1。

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被选择。

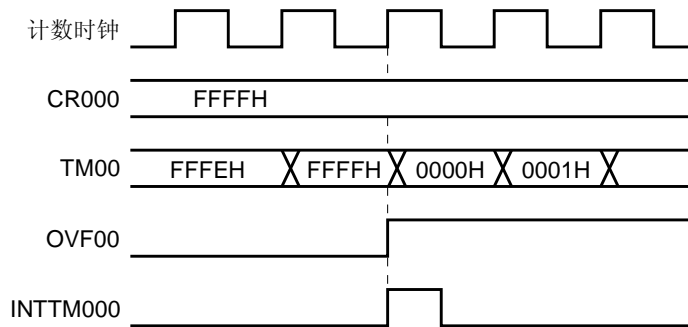
↓

CR000 被设置为 FFFFH。

↓

当 TM00 从 FFFFH 向上计数到 0000H 时。

图 6-37. OVF00 标志的操作时序

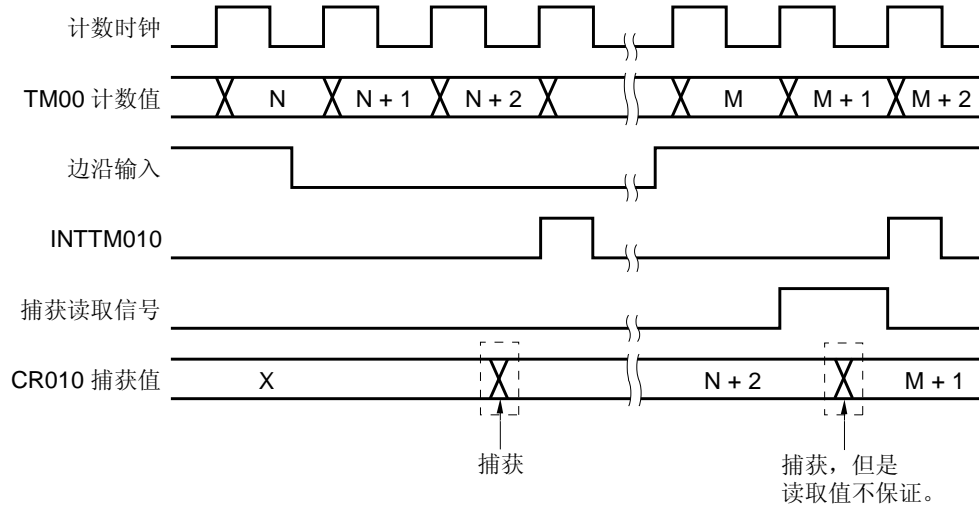


<2> 如果在 TM00 溢出发生之后下一个计数时钟 (TM00 变为 0001H 之前) 之前 OVF00 标志被清除, OVF00 标志将被重新设置并且禁止清除。

(14) 冲突操作

如果当 CR000 / CR010 用作捕获寄存器时寄存器读取周期和捕获触发的输入冲突, 捕获触发输入具有优先权并且读取的数据不确定。同时, 如果定时器计数停止和捕获触发输入冲突, 捕获数据不确定。

图 6-38. 捕获寄存器数据保持时序



(15) 捕获操作

- <1> 如果 TI000 引脚的有效沿将被设置为计数时钟，不要在 TI000 引脚的有效沿时设置清除 / 开始模式和捕获触发。
- <2> 当 CRC001 位为 1 时，如果上升沿和下降沿都被指定为 TI000 引脚的有效沿，CR000 寄存器中的捕获操作不会执行。
- <3> 当 CRC001 位的值为 1 时，当 TI010 引脚的有效沿被检测时，TM00 的计数值不会被写入到 CR000 寄存器中，但是从 TI010 引脚的输入可以被用作外部中断源，因为 INTTM000 在那时被产生。
- <4> 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) 所选择的计数时钟的两个周期更长的脉冲。
- <5> 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 (INTTM0n0) 发生。
- <6> 要使用两个捕获寄存器，设置 TI000 设置 TI010 引脚。

备注 n = 0, 1

(16) 比较操作

如果捕获触发已经被输入，捕获操作可能不会对设置为比较模式的 CR0n0 执行。

备注 n = 0, 1

(17) 定时器操作期间更改比较寄存器

<1> 对于 16 位定时器捕获 / 比较寄存器 0n0 (CR0n0) 用作比较寄存器, 在定时器计数期间 16 位定时器计数器 00 (TM00) 和 16 位定时器捕获 / 比较寄存器 0n0 (CR0n0) 匹配时序附近更改 CR0n0 时, 更改时序可能与匹配的时序冲突, 所以在这种情况下操作不被保证。要在定时器计数期间更改 CR0n0, INTTM000 中断服务程序执行以下操作。

<更改周期 (CR000) >

1. 禁止 TM00 和 CR000 匹配时的定时器输出反转操作 (TOC001 = 0)。
2. 禁止 INTTM000 中断 (TMMK000 = 1)。
3. 重写 CR000。
4. 等待 TM00 计数时钟的一个周期。
5. 允许 TM00 和 CR000 匹配时的定时器输出反转操作 (TOC001 = 1)。
6. 清除 INTTM000 的中断请求标志 (TMIF000 = 0)。
7. 允许 INTTM000 中断 (TMMK000 = 0)。

<更改占空比 (CR010) >

1. 禁止 TM00 和 CR010 匹配时的定时器输出反转操作 (TOC004 = 0)。
2. 禁止 INTTM000 中断 (TMMK000 = 1)。
3. 重写 CR010。
4. 等待 TM00 计数时钟的一个周期。
5. 允许 TM00 和 CR010 匹配时的定时器输出反转操作 (TOC004 = 1)。
6. 清除 INTTM000 的中断请求标志 (TMIF000 = 0)。
7. 允许 INTTM000 中断 (TMMK000 = 0)。

当中断和定时器输出反正被禁止时 (上面的 1 到 4), 定时器继续计数。如果要设置到 CR0n0 中的值很小, TM00 的值可能超过 CR0n0。因此, 考虑 INTTM000 中断产生后定时器时钟和 CPU 的时间差来设置数值。

备注 n = 0 或 1

<2> 如果在定时器计数期间 CR010 被更改而没有执行上面的处理<1>, CR010 中的值可能被重写两次或更多, 导致每次重写时 TO00 引脚的输出电平反转。

(18) 边沿检测

<1> 在以下情况下，注意 TI0n0 引脚的有效沿被检测。

(a) 系统复位后，如果高电平被输入 TI0n0 引脚，16 位定时器计数器 00 (TM00) 的操作被允许。

→ 如果上升沿和双边沿都被指定为 TI0n0 引脚的有效沿，在 TM00 操作被允许后，一个上升沿立即被检测。

(b) 如果在 TI0n0 引脚为高电平过程中 TM00 操作被停止，在低电平被输入到 TI0n0 引脚后，TM00 操作被允许。

→ 如果下降沿和双边沿都被指定为 TI0n0 引脚的有效沿，在 TM00 操作被允许后，一个下降沿立即被检测。

(c) 当在 TI0n0 引脚为低电平过程中 TM00 操作被停止时，在高电平被输入到 TI0n0 引脚后，TM00 操作被允许。

→ 如果上升沿和双边沿都被指定为 TI0n0 引脚的有效沿，在 TM00 操作被允许后，一个上升沿立即被检测。

备注 n = 0, 1

<2> 用于移除噪声的采样时钟与 TI000 的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_{XP} ，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被采样并且有效电平被检测到两次使才会执行，这样可以消除短脉冲宽度的噪声。

(19) 外部事件计数器

<1> 计数开始的时序在两个有效沿检测后。

<2> 当读取外部事件计数器计数值时，TM00 应该被读取。

(20) PPG 输出

<1> 以下范围内的值应该被设置到 CR000 和 CR010 中：

$$0000H \leq CR010 < CR000 \leq FFFFH$$

<2> 通过 PPG 输出产生的脉冲的周期 (CR000 设置值 + 1) 具有 (CR010 设置值 + 1) / (CR000 设置值 + 1) 的占空比。

(21) STOP 模式或系统时钟停止模式设置

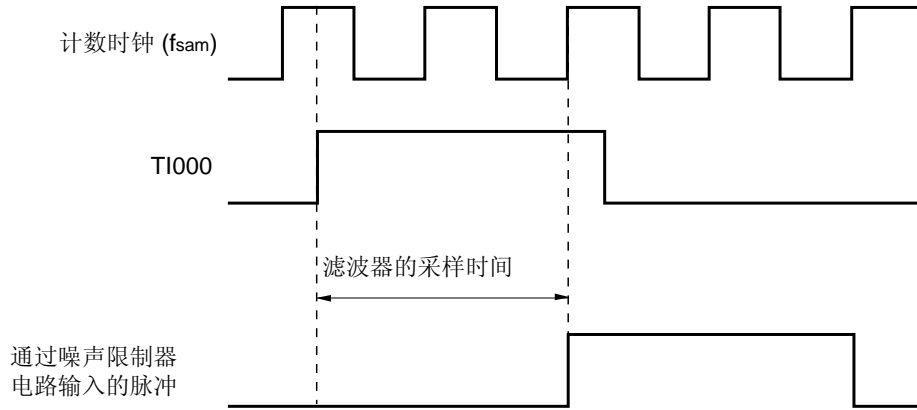
除了当 TI000 引脚的有效沿被选作计数时钟时，在设置 STOP 模式或系统时钟停止模式之前停止计数器操作；否则，当系统时钟开始时定时器可能发生故障。

(22) P21/TI010/TO00 引脚

当 P21 被用作有效沿的输入引脚 (TI010) 时，它不能用作定时器输出 (TO00)。当 P21 被用作定时器输出引脚 (TO00) 时，它不能用作有效沿的输入引脚 (TI010)。

(23) 外部时钟限制

- <1> 当使用 T1000 引脚的输入脉冲作为计数时钟（外部触发）时，确保输入满足 AC 特性的脉冲宽度。关于 AC 特性，参阅第十七章 电气规范（目标）。
- <2> 当一个外部波形被输入 16 位定时器 / 事件计数器 00 时，它由噪声限制器电路采样，因此在设备内变为有效时，一个错误发生。



备注 计数时钟 (f_{sam}) 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。

第七章 8 位定时器H1

7.1 8 位定时器H1 的功能

8 位定时器 H1 有以下功能。

- 间隔定时器
- PWM 输出模式
- 方波输出

7.2 8 位定时器H1 的配置

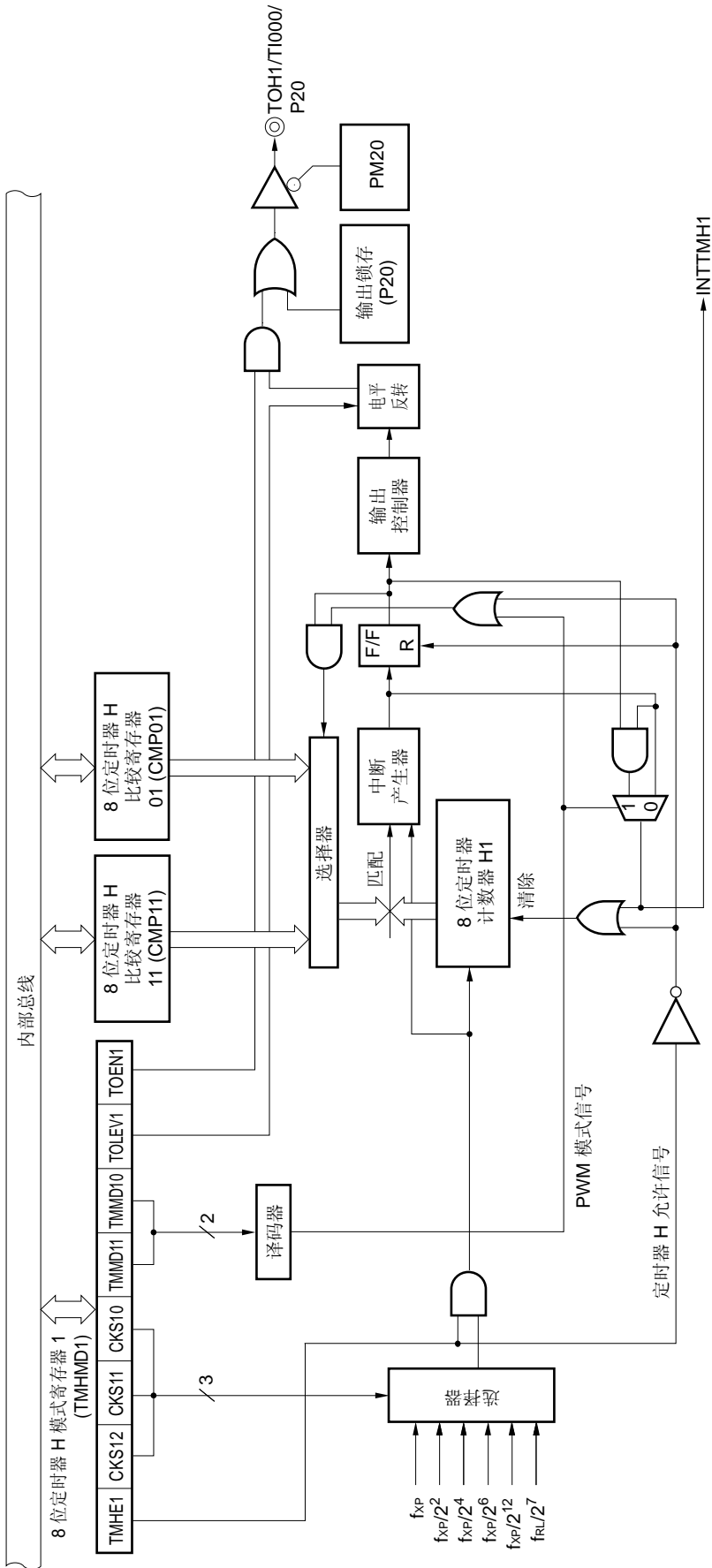
8 位定时器 H1 由以下硬件组成。

表 7-1. 8 位定时器 H1 的配置

项目	配置
定时器寄存器	8 位定时器计数器 H1
寄存器	8 位定时器 H 比较寄存器 01 (CMP01) 8 位定时器 H 比较寄存器 11 (CMP11)
定时器输出	TOH1
控制寄存器	8 位定时器 H 模式寄存器 1 (TMHMD1) 端口模式寄存器 2 (PM2) 端口寄存器 2 (P2)

图 7-1 表示了框图。

图 7-1. 8 位定时器 H1 的框图

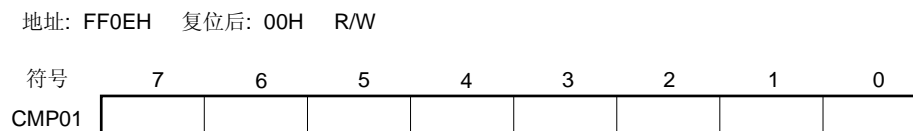


(1) 8 位定时器 H 比较寄存器 01 (CMP01)

该寄存器可以由 8 位存储器操作指令读取或写入。

复位信号清除该寄存器为 00H。

图 7-2. 8 位定时器 H 比较寄存器 01 (CMP01) 的格式



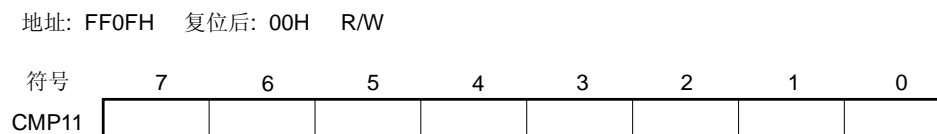
注意事项 CMP01 在定时器计数操作期间不能被重新写入。

(2) 8 位定时器 H 比较寄存器 11 (CMP11)

该寄存器可以由 8 位存储器操作指令读取或写入。

复位信号清除该寄存器为 00H。

图 7-3. 8 位定时器 H 比较寄存器 11 (CMP11) 的格式



CMP11 可以在定时器计数操作期间被重新写入。

如果 CMP11 的值在定时器操作期间被重写，重写后的比较值在计数值与重写前的比较值匹配时起作用。如果计数值和比较值匹配的时序与 CPU 写 CMP11 的时序冲突，重写后的比较值在下一个计数值与重写前的比较值匹配时起作用。

注意事项 在 PWM 输出模式下，在定时器计数操作停止 (TMHE1 = 0) 后当定时器计数操作开始 (TMHE1 = 1) 时，确保设置 CMP11 (即使设置相同的值到 CMP11，也要确保再次设置)。

7.3 控制 8 位定时器 H1 的寄存器

以下四个寄存器用来控制 8 位定时器 H1。

- 8 位定时器 H 模式寄存器 1 (TMHMD1)
- 端口模式寄存器 2 (PM2)
- 端口寄存器 2 (P2)

(1) 8 位定时器 H 模式寄存器 1 (TMHMD1)

该寄存器控制定时器 H 的模式。

该寄存器可以由 1 位或 8 位存储器操作指令来设置。

复位信号清除该寄存器为 00H。

图 7-4. 8 位定时器 H 模式寄存器 1 (TMHMD1) 的格式

地址: FF70H 复位后: 00H R/W

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

TMHE1	定时器操作允许
0	停止定时器计数操作 (计数器被清除为 0)
1	允许定时器计数操作 (计数操作由输入时钟启动)

CKS12	CKS11	CKS10	计数时钟 (f _{CNT}) 的选择
0	0	0	f _{XP} (10 MHz)
0	0	1	f _{XP} /2 ² (2.5 MHz)
0	1	0	f _{XP} /2 ⁴ (625 kHz)
0	1	1	f _{XP} /2 ⁶ (156.25 kHz)
1	0	0	f _{XP} /2 ¹² (2.44 kHz)
1	0	1	f _{RL} /2 ⁷ (1.88 kHz (典型值))
除上面以外			禁止设置

TMMD11	TMMD10	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

TOLEV1	定时器输出电平控制 (默认模式下)
0	低电平
1	高电平

TOEN1	定时器输出控制
0	禁止输出
1	允许输出

- 注意事项**
1. 当 **TMHE1 = 1** 时, 禁止设置 **TMHMD1** 寄存器的其它位。
 2. 在 **PWM** 输出模式下, 在定时器计数操作停止 (**TMHE1 = 0**) 后当定时器计数操作开始 (**TMHE1 = 1**) 时, 确保设置 **8 位定时器 H 比较寄存器 11 (CMP11)** (即使设置相同的值到 **CMP11**, 也要确保再次设置)。

- 备注**
1. f_{XP}: 到周边硬件的时钟的振荡频率
 2. f_{RL}: 低速内部振荡时钟的振荡频率
 3. 括号中的数字应用于 f_{XP} = 10 MHz, f_{RL} = 240 kHz (典型值) 的操作。

7.4 8 位定时器 H1 的操作

7.4.1 作为间隔定时器 / 方波输出工作

当 8 位定时器计数器 H1 和比较寄存器 01 (CMP01) 的值匹配时, 中断请求信号 (INTTMH1) 被产生并且 8 位定时器计数器 H1 被清除为 00H。

比较寄存器 11 (CMP11) 不用于间隔定时器模式。因为即使 CMP11 寄存器被设置, 8 位定时器计数器 H1 和 CMP11 寄存器的匹配不能被检测, 定时器输出不受影响。

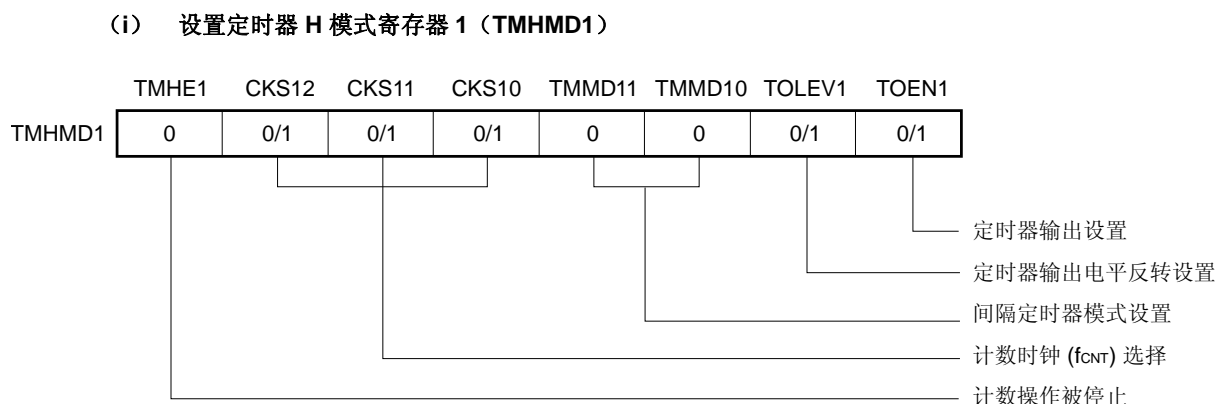
通过设置定时器 H 模式寄存器 1 (TMHMD1) 的第 0 位 (TOEN1) 为 1, 任意频率 (占空比 = 50%) 的方波从 TOH1 被输出。

(1) 用法

以相同的间隔重复产生 INTTMH1 信号。

<1> 设置每个寄存器。

图 7-5. 间隔定时器 / 方波输出操作期间的寄存器设置



(ii) CMP01 寄存器设置

- 比较值 (N)

<2> 当 TMHE1 = 1 时, 计数操作开始。

<3> 当 8 位定时器计数器 H1 和 CMP01 寄存器的值匹配时, INTTMH1 被产生并且 8 位定时器计数器 H1 被清除为 00H。

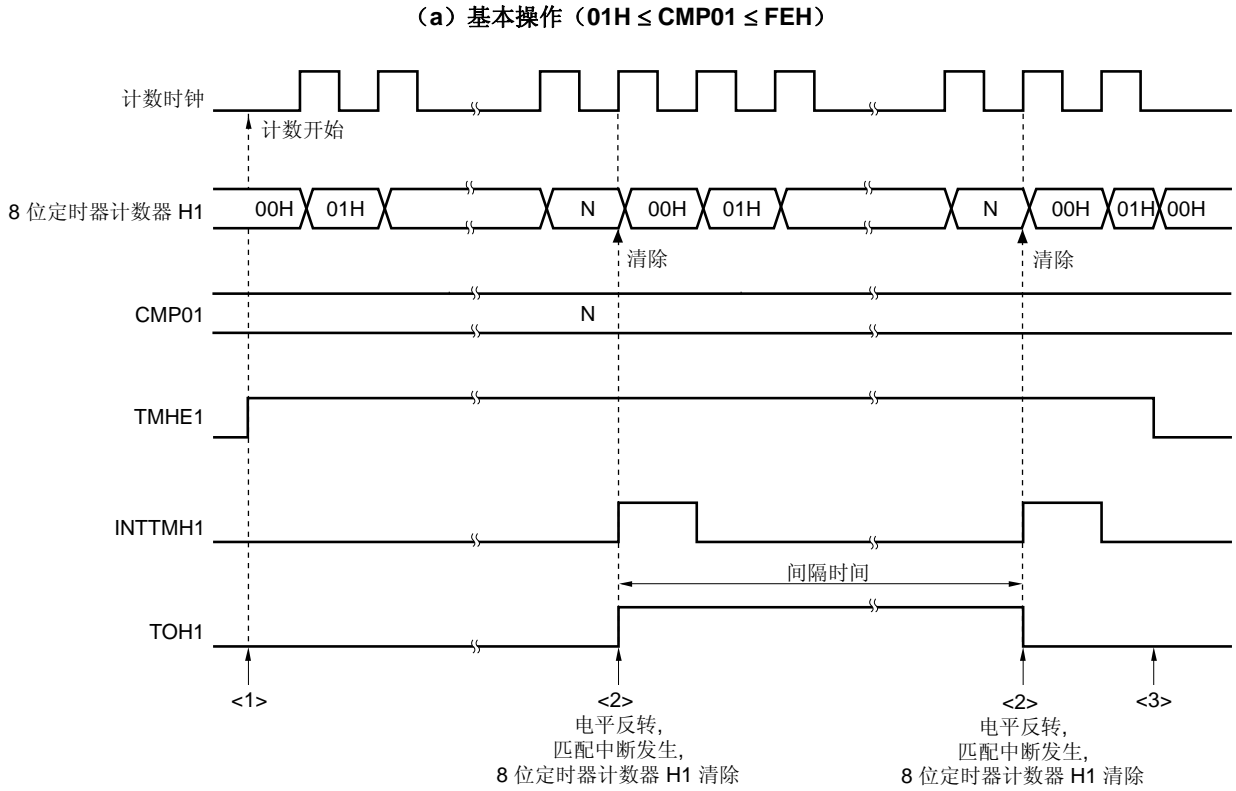
$$\text{间隔时间} = (N + 1) / f_{\text{CNT}}$$

<4> 然后, INTTMH1 信号以同样间隔被产生。要停止计数操作, 清除 TMHE1 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 7-6. 间隔定时器 / 方波输出操作时序 (1 / 2)

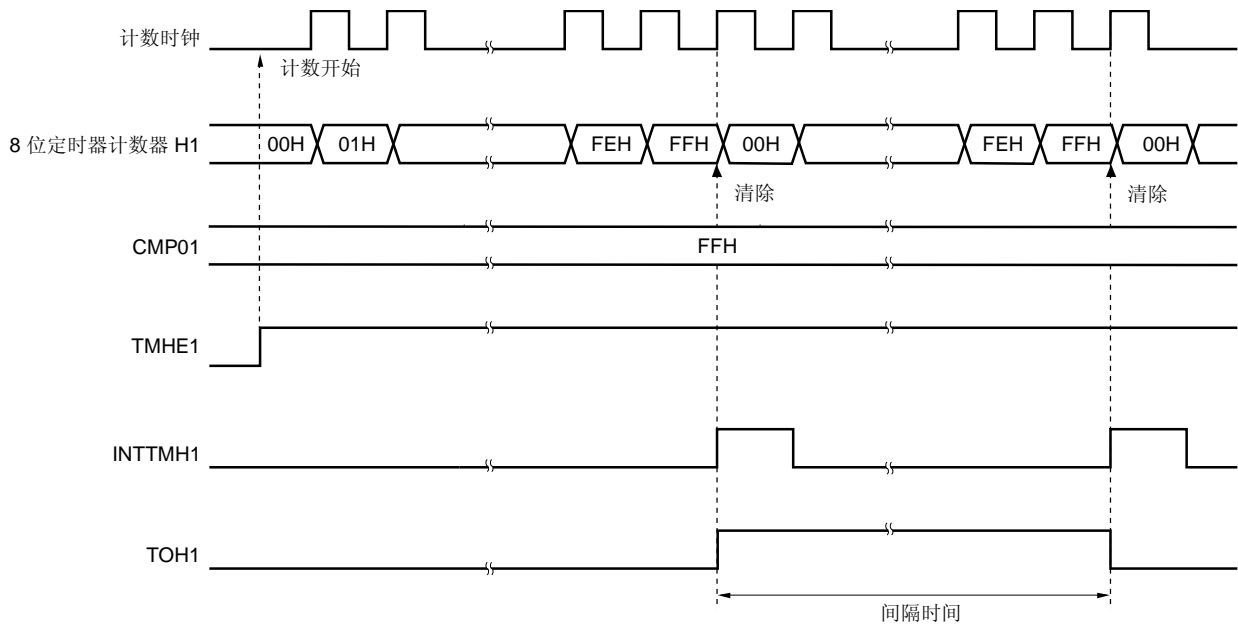


- <1> 通过设置 TMHE1 位为 1，计数操作被允许。在操作被允许后不超过一个时钟内，计数时钟开始计数。
- <2> 当 8 位定时器计数器 H1 的值和 CMP01 寄存器匹配时，8 位定时器计数器 H1 被清除，TOH1 输出电平被反转，并且 INTTMH1 信号被输出。
- <3> 在定时器 H1 操作期间，通过清除 TMHE1 位为 0，INTTMH1 信号和 TOH1 输出变为无效。如果从第一个就存在无效电平，电平被保持。

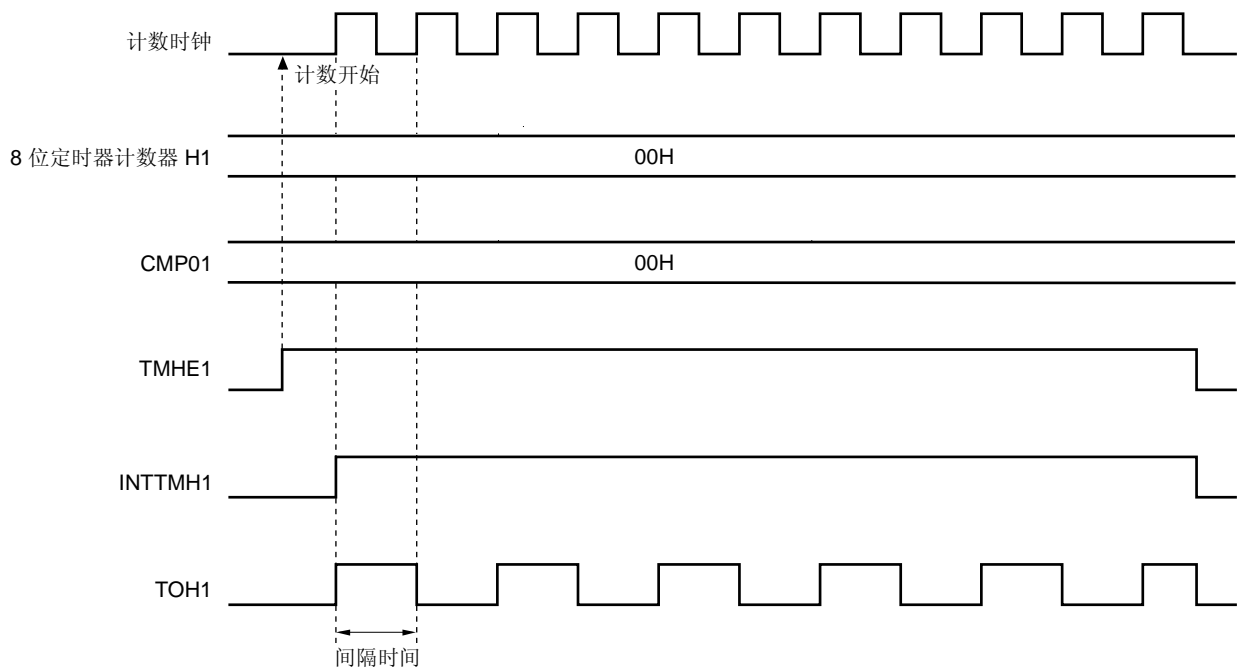
备注 $01H \leq N \leq FEH$

图 7-6. 间隔定时器 / 方波输出操作时序 (2 / 2)

(b) 当 CMP01 = FFH 时的操作



(c) 当 CMP01 = 00H 时的操作



7.4.2 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可以被输出。

8 位定时器比较寄存器 01 (CMP01) 控制定时器输出 (TOH1) 的周期。在定时器操作期间，禁止重新写入 CMP01 寄存器。

8 位定时器比较寄存器 11 (CMP11) 控制定时器输出 (TOH1) 的占空比。在定时器操作期间，可以重新写入 CMP11 寄存器。

PWM 模式下的操作如下所示。

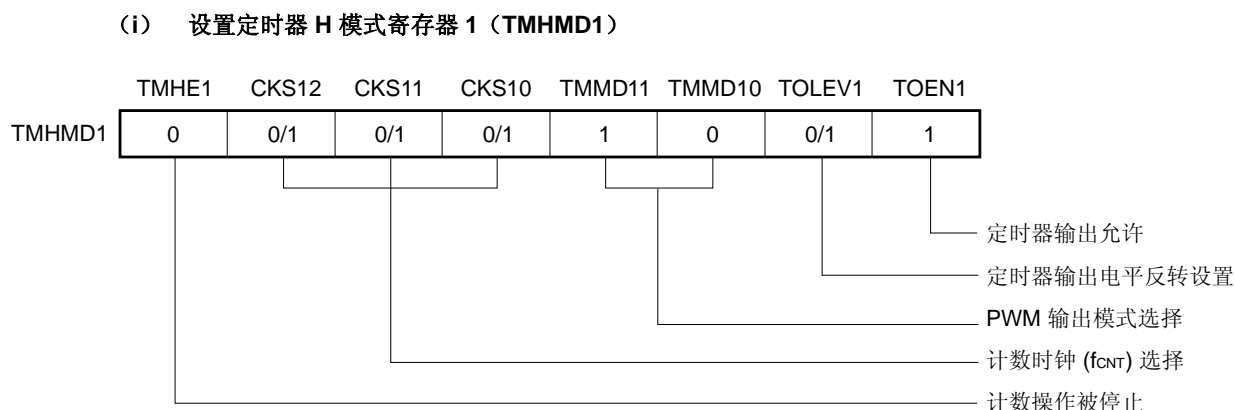
在定时器计数开始后，当 8 位定时器计数器 H1 和 CMP01 寄存器匹配时，TOH1 输出电平变为有效并且 8 位定时器计数器 H1 被清除为 0。当 8 位定时器 H1 和 CMP11 寄存器匹配时，TOH1 输出变为无效。

(1) 用法

PWM 输出模式下，可以设置的任意占空比和任意周期的脉冲被输出。

<1> 设置每个寄存器。

图 7-7. PWM 输出模式下的寄存器设置



(ii) 设置 CMP01 寄存器

- 比较值 (N)：周期设置

(iii) 设置 CMP11 寄存器

- 比较值 (M)：占空比设置

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

<2> 当 TMHE1 = 1 时，计数操作开始。

<3> CMP01 寄存器是计数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 H1 的值和 CMP01 寄存器匹配时，8 位定时器计数器 H1 被清除，中断请求信号 (INTTMH1) 被产生，并且 TOH1 输出变为有效。同时，与 8 位定时器计数器 H1 比较的比较寄存器从 CMP01 寄存器改变为 CMP11 寄存器。

- <4> 当 8 位定时器计数器 H1 和 CMP11 寄存器匹配时，TOH1 输出变为无效并且与 8 位定时器计数器 H1 比较的比较寄存器从 CMP11 寄存器改变为 CMP01 寄存器。这时，8 位定时器计数器 H1 不被清除并且 INTTMH1 信号不被产生。
- <5> 通过重复执行过程<3>和<4>，一个具有任意占空比的脉冲被得到。
- <6> 要停止计数操作，设置 TMHE1 = 0。

如果 CMP01 寄存器的设置值为 N，CMP11 寄存器的设置值为 M，并且计数时钟频率为 f_{CNT} ，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，CMP11 寄存器的设置值可以在定时器计数操作期间改变。然而，在重新写入 CMP11 寄存器的值后需要三个操作时钟（使用 TMHMD1 寄存器的 CKS12 到 CKS10 位选择的信号）或更多周期来传输 CMP11 寄存器值。
 2. 在定时器计数操作停止（TMHE1 = 0）后当定时器计数操作开始（TMHE1 = 1）时，确保设置 CMP11 寄存器（即使设置相同的值到 CMP11 寄存器，也要确保再次设置）。

(2) 时序图

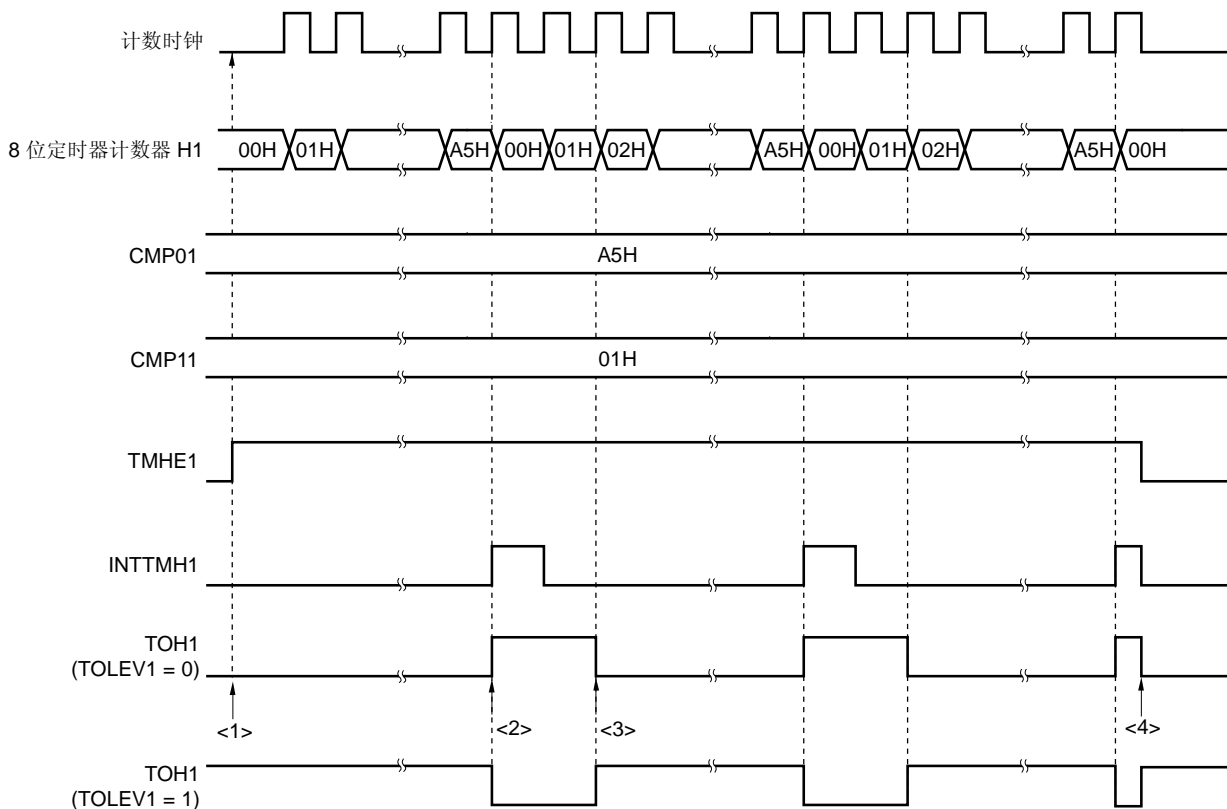
PWM 输出模式下的操作时序如下所示。

注意事项 确保 CMP11 寄存器的设置值 (M) 和 CMP01 寄存器的设置值 (N) 在以下范围内。

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

图 7-8. PWM 输出模式下的操作时序 (1 / 4)

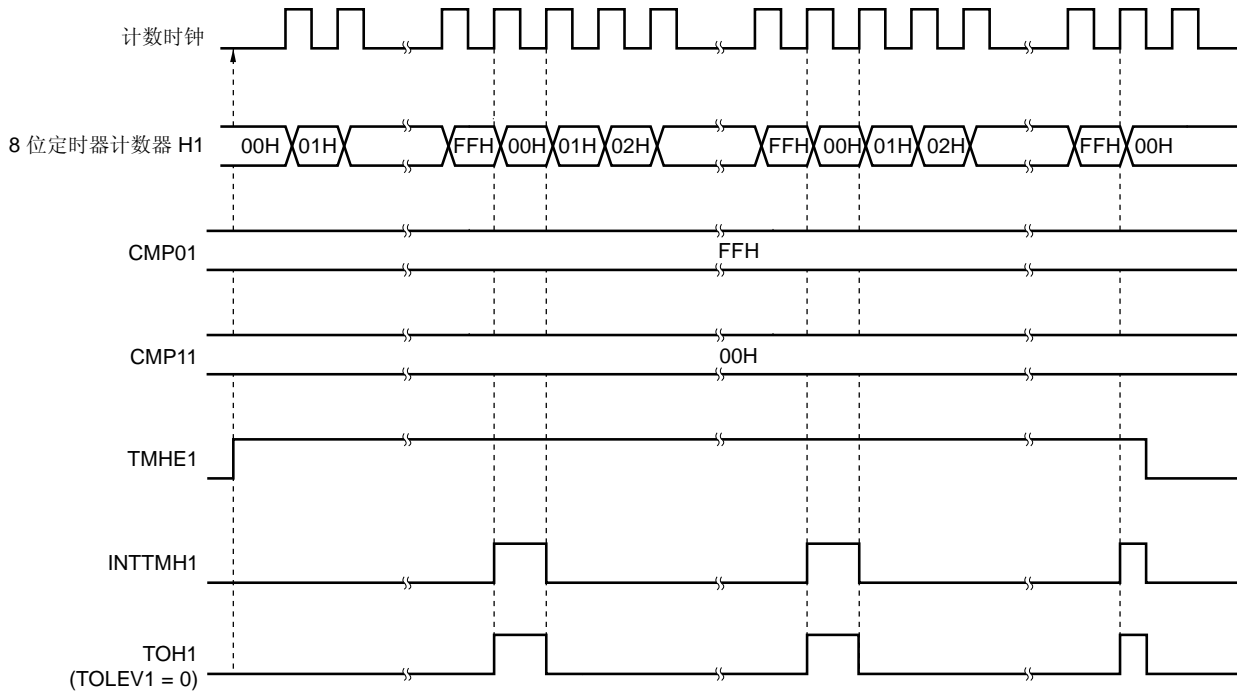
(a) 基本操作 (00H < CMP11 < CMP01 < FFH)



- <1> 计数操作通过设置 **TMHE1** 位为 1 来允许。通过屏蔽一个计数时钟来启动 8 位定时器计数器 H1 向上计数。同时, **TOH1** 输出保持无效 (当 **TOLEV1 = 0** 时)。
- <2> 当 8 位定时器计数器 H1 的值和 **CMP01** 寄存器匹配时, **TOH1** 输出电平被反转, 8 位定时器计数器 H1 被清除, **INTTMH1** 信号被输出。
- <3> 当 8 位定时器计数器 H1 的值和 **CMP11** 寄存器匹配时, **TOH1** 输出电平被返回。这时, 8 位定时器计数器 H1 不被清除并且 **INTTMH1** 信号不被输出。
- <4> 在定时器 H1 操作期间清除 **TMHE1** 位为 0 可以设置 **INTTMH1** 信号和 **TOH1** 输出无效。

图 7-8. PWM 输出模式下的操作时序 (2 / 4)

(b) 当 $CMP01 = FFH$, $CMP11 = 00H$ 时的操作



(c) 当 $CMP01 = FFH$, $CMP11 = FEH$ 时的操作

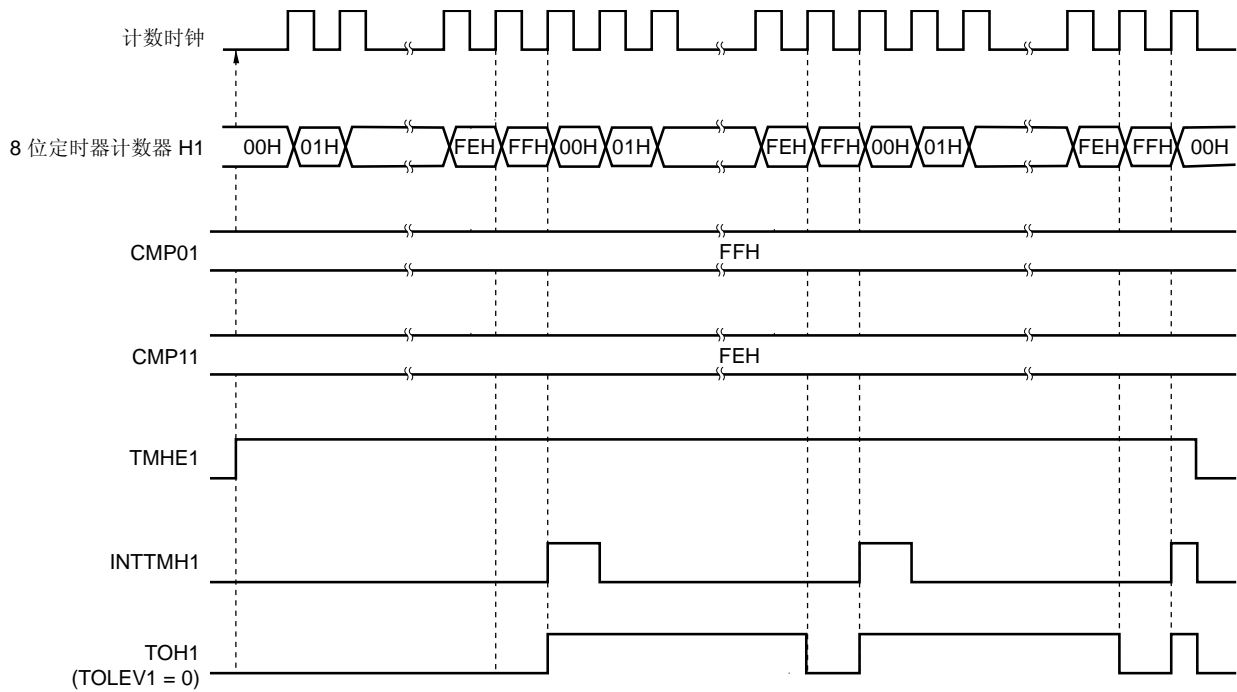


图 7-8. PWM 输出模式下的操作时序 (3 / 4)

(d) 当 $CMP01 = 01H$, $CMP11 = 00H$ 时的操作

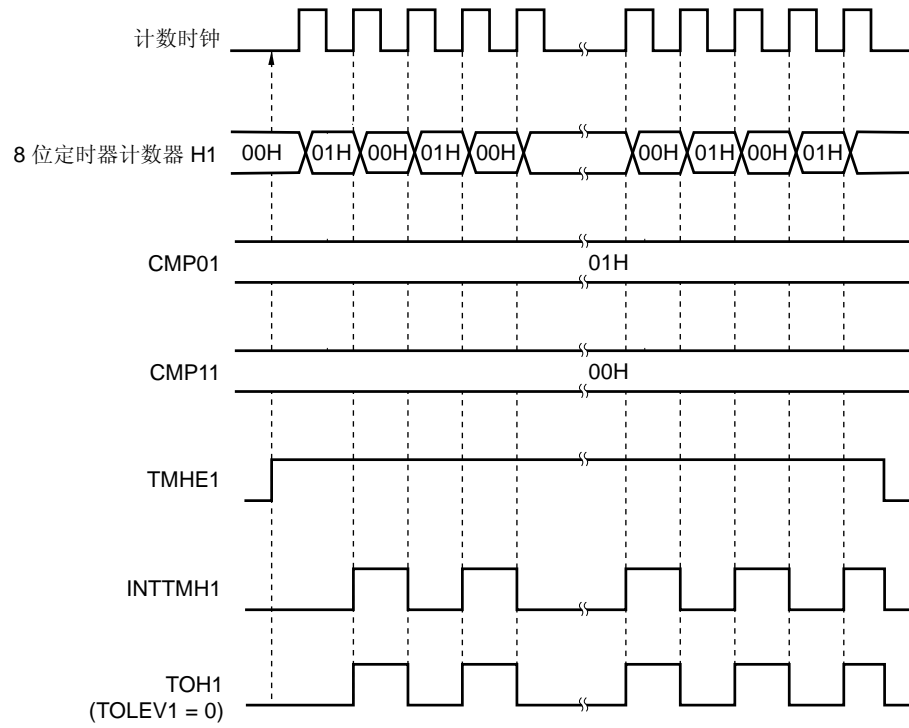
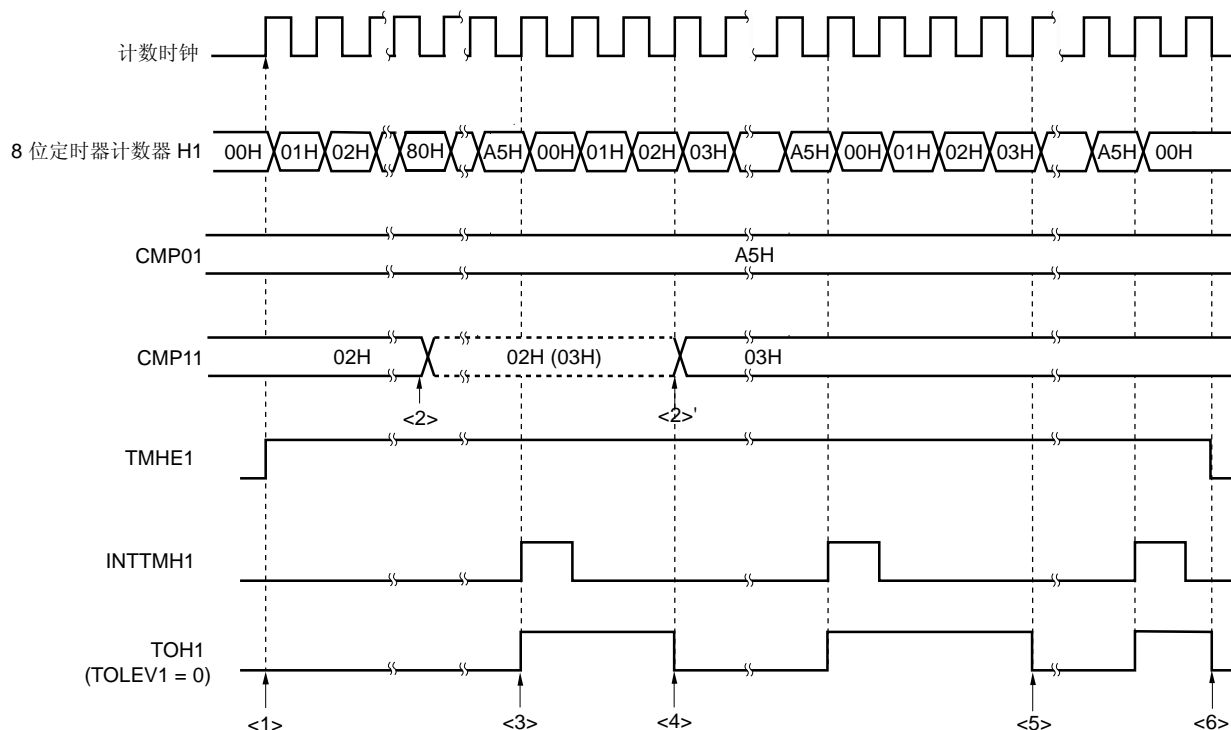


图 7-8. PWM 输出模式下的操作时序 (4 / 4)

(e) 改变 CMP11 (CMP11 = 02H → 03H, CMP01 = A5H) 的操作



- <1> 计数操作通过设置 TMHE1 位为 1 来允许。通过屏蔽一个计数时钟来启动 8 位定时器计数器 H1 向上计数。同时, TOH1 输出保持无效 (当 TOLEV1 = 0 时)。
 - <2> CMP11 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 H1 的值和 CMP01 寄存器匹配时, 8 位定时器计数器 H1 被清除, TOH1 输出变为有效, 并且 INTTMH1 信号被输出。
 - <4> 如果 CMP11 寄存器的值被改变, 这个值被锁存并且不会转移到寄存器。当 8 位定时器计数器 H1 的值与 CMP11 寄存器以改变前的值匹配时, 这个值被转移到 CMP11 寄存器并且 CMP11 寄存器的值被更改 (<2>')。
- 然而, 从 CMP11 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, 改变的值不能被转移到寄存器。
- <5> 当 8 位定时器计数器 H1 的值和 CMP11 寄存器匹配时, TOH1 输出电平变为无效。8 位定时器计数器 H1 不被清除并且 INTTMH1 信号不被产生。
 - <6> 在定时器 H1 操作期间清除 TMHE1 位为 0 可以设置 INTTMH1 信号和 TOH1 输出无效。

第八章 看门狗定时器

8.1 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。如果一个程序循环被检测，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，见第十一章 复位功能。

表 8-1. 看门狗定时器的循环检测时间

循环检测时间	
低速内部振荡时钟操作期间	系统时钟操作期间
$2^{11}/f_{RL}$ (4.27 ms)	$2^{13}/f_x$ (819.2 μ s)
$2^{12}/f_{RL}$ (8.53 ms)	$2^{14}/f_x$ (1.64 ms)
$2^{13}/f_{RL}$ (17.07 ms)	$2^{15}/f_x$ (3.28 ms)
$2^{14}/f_{RL}$ (34.13 ms)	$2^{16}/f_x$ (6.55 ms)
$2^{15}/f_{RL}$ (68.27 ms)	$2^{17}/f_x$ (13.11 ms)
$2^{16}/f_{RL}$ (136.53 ms)	$2^{18}/f_x$ (26.21 ms)
$2^{17}/f_{RL}$ (273.07 ms)	$2^{19}/f_x$ (52.43 ms)
$2^{18}/f_{RL}$ (546.13 ms)	$2^{20}/f_x$ (104.86 ms)

- 备注
1. f_{RL} : 低速内部振荡时钟的振荡频率
 2. f_x : 系统时钟的振荡频率
 3. 括号中的数字应用于 $f_{RL} = 480$ kHz（最大值）， $f_x = 10$ MHz 的操作。

看门狗定时器（WDT）的操作模式根据片上低速内部振荡器的选项字节设置切换，如表 8-2 所示。

表 8-2. 选项字节设置和看门狗定时器的操作模式

	选项字节设置	
	低速内部振荡器不能被停止	低速内部振荡器可由软件停止
看门狗定时器时钟源	固定为 f_{RL} ¹ 。	<ul style="list-style-type: none"> 可由软件选择 (f_x, f_{RL} 或被停止) 当复位被释放时: f_{RL}
复位后的操作	操作以最大间隔 ($2^{18}/f_{RL}$) 开始。	操作以最大间隔 ($2^{18}/f_{RL}$) 开始。
操作模式选择	间隔只能被更改一次。	时钟选择 / 间隔只能被更改一次。
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, 低速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。
 - 如果时钟源为 f_x , 提供给看门狗定时器的时钟在以下条件下被停止。
 - 当 f_x 被停止时
 - 在 HALT / STOP 模式下
 - 在振荡稳定时间期间
 - 如果时钟源为 f_{RL} , 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_x 并且如果 f_{RL} 在 STOP 指令执行之前由软件停止
 - 在 HALT / STOP 模式下

- 备注**
- f_{RL} : 低速内部振荡时钟的振荡频率
 - f_x : 系统时钟的振荡频率

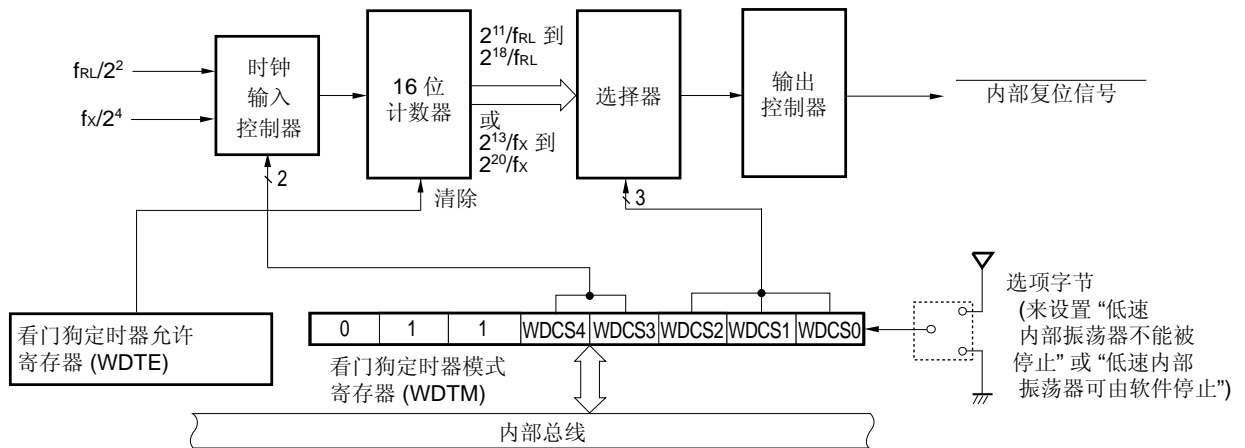
8.2 看门狗定时器的配置

看门狗定时器由以下硬件组成。

表 8-3. 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 8-1. 看门狗定时器的框图



- 备注**
1. f_{RL} ：低速内部振荡时钟的振荡频率
 2. f_x ：系统时钟的振荡频率

8.3 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。

复位信号设置该寄存器为 67H。

图 8-2. 看门狗定时器模式寄存器 (WDTM) 的格式

地址: FF48H 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	低速内部振荡时钟 (f _{RL})
0	1	系统时钟 (f _x)
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	
			低速内部振荡时钟操作期间	系统时钟操作期间
0	0	0	2 ¹¹ /f _{RL} (4.27 ms)	2 ¹³ /f _x (819.2 μs)
0	0	1	2 ¹² /f _{RL} (8.53 ms)	2 ¹⁴ /f _x (1.64 ms)
0	1	0	2 ¹³ /f _{RL} (17.07 ms)	2 ¹⁵ /f _x (3.28 ms)
0	1	1	2 ¹⁴ /f _{RL} (34.13 ms)	2 ¹⁶ /f _x (6.55 ms)
1	0	0	2 ¹⁵ /f _{RL} (68.27 ms)	2 ¹⁷ /f _x (13.11 ms)
1	0	1	2 ¹⁶ /f _{RL} (136.53 ms)	2 ¹⁸ /f _x (26.21 ms)
1	1	0	2 ¹⁷ /f _{RL} (273.07 ms)	2 ¹⁹ /f _x (52.43 ms)
1	1	1	2 ¹⁸ /f _{RL} (546.13 ms)	2 ²⁰ /f _x (104.86 ms)

- 注
1. 如果“低速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，低速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. 分别设置第 7、6 和 5 位为 0、1 和 1。不要设置其它位。

注意事项 2. 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。然而，第一次写入时，如果“1”和“x”被分别设置到 WDCS4 和 WDCS3 并且看门狗定时器被停止，即使以下操作被执行，内部复位信号也不会发生。

- 第二次写入 WDTM
 - 对 WDTE 执行 1 位存储器操作指令。
 - “ACH”以外的值被写入 WDTE
3. WDTM 不能通过 1 位存储器操作指令来设置。
 4. 当通过自写入使用 flash 存储器自编程时，为看门狗定时器设置溢出时间以保证足够的溢出时间（示例 1 字节写入：200 μ s 最小值，1-block 检测：10 ms 最小值）。

- 备注**
1. f_{RL}: 低速内部振荡时钟的振荡频率
 2. f_x: 系统时钟的振荡频率
 3. x: 不关注
 4. 括号中的数字应用于 f_{RL} = 480 kHz（最大值），f_x = 10 MHz 的操作。

(2) 看门狗定时器允许寄存器 (WDTE)

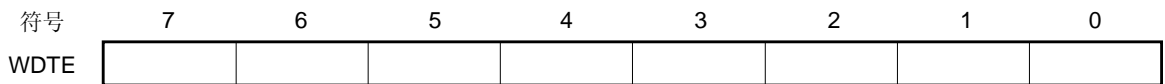
向 WDTE 写入 ACH 将清除看门狗定时器计数器并且再次启动计数。

该寄存器可以由 8 位存储器操作指令来设置。

复位信号设置该寄存器为 9AH。

图 8-3. 看门狗定时器允许寄存器 (WDTE) 的格式

地址: FF49H 复位后: 9AH R/W



- 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)）。

8.4 看门狗定时器的操作

8.4.1 当“低速内部振荡器可由软件停止”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固定为低速内部振荡时钟。

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。

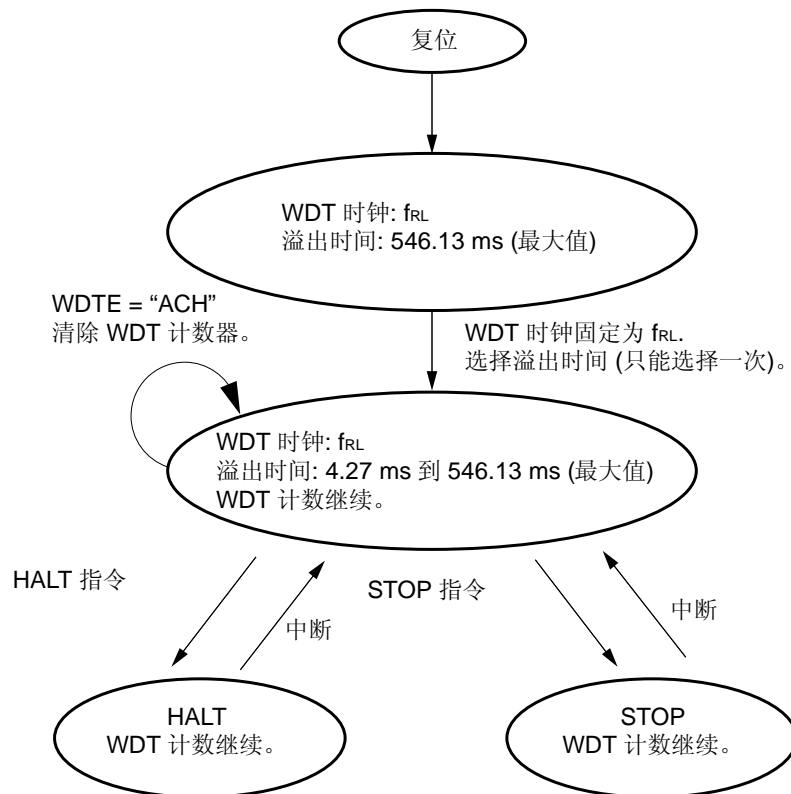
1. 复位释放后的状态如下所示。
 - 操作时钟：低速内部振荡时钟
 - 周期： $2^{18}/f_{RL}$ （546.13 ms：以 $f_{RL} = 480$ kHz（最大值）操作）
 - 计数开始
2. 应该按照下面所示通过 8 位存储器操作指令来设置看门狗定时器（WDTM）^{注 1, 2}。
 - 周期：使用第 2 到 0 位（WDCS2 到 WDCS0）来设置
3. 以上流程执行后，写 ACH 到 WDTE 来清除计数为 0 并允许重新计数。

- 注**
1. 操作时钟（低速内部振荡时钟）不能被停止。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不能被停止。对应 8 位定时器 H1（TMH1），低速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 TMH1 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。

状态转换图如下所示。

图 8-4. 当“低速内部振荡器不能被停止”由选项字节选择时的状态转换图



8.4.2 当“低速内部振荡器可由软件停止”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低速内部振荡时钟或系统时钟。

复位释放后，操作以低速内部振荡时钟的最大周期被启动（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：低速内部振荡时钟
 - 周期： $2^{18}/f_{RL}$ （546.13 ms：以 $f_{RL} = 480$ kHz（最大值）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。
 - 低速内部振荡时钟
 - 系统时钟（fx）
 - 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。

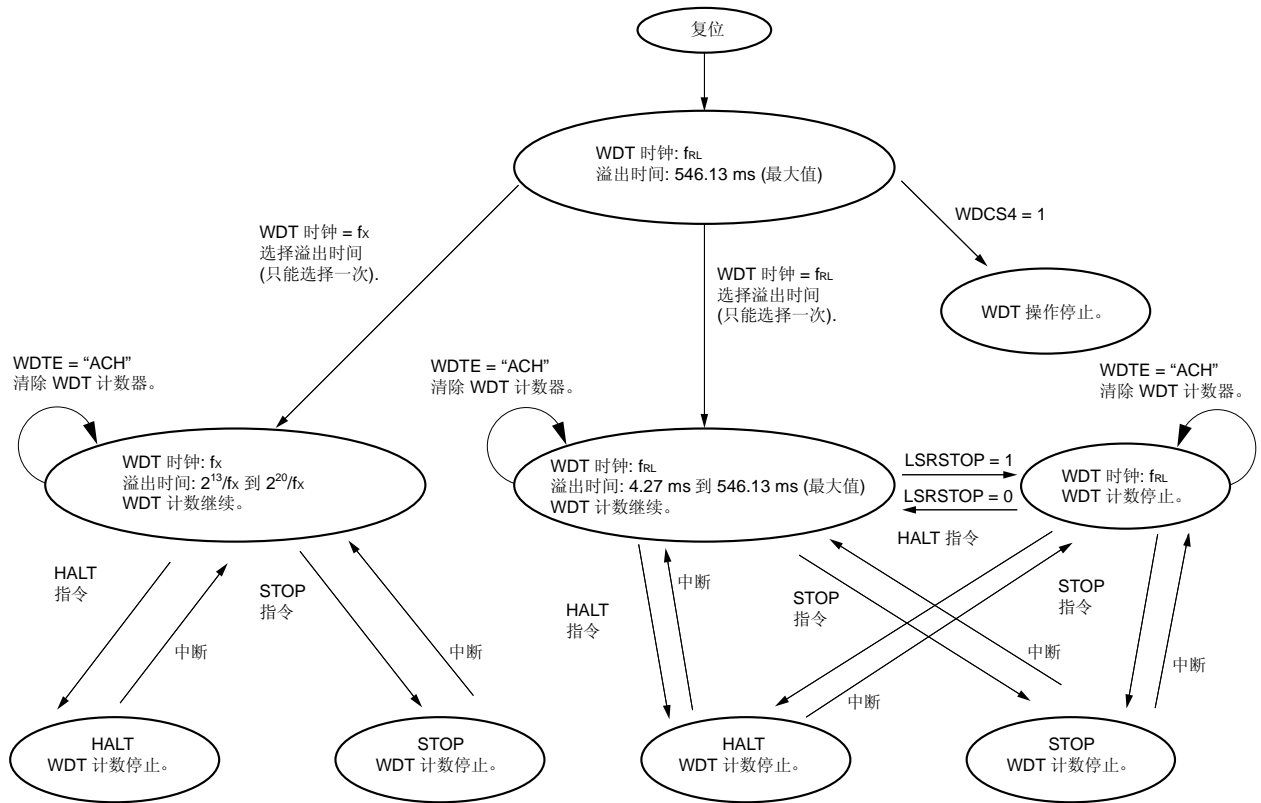
- 注**
1. 只要 WDTM 被写入，看门狗定时器的计数器就被清除。
 2. 分别设置第 7、6 和 5 位为 0、1、1。不要设置其它值。
 3. 第一次写入时，如果通过设置 WDCS4 和 WDCS3 分别为 1 和 x 看门狗定时器被停止，即使以下处理被执行，内部复位信号也不会被产生。
 - WDTM 被写入两次。
 - 对 WDTE 执行 1 位存储器操作指令。
 - ACH 以外的值被写入 WDTE

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。

关于每种状态下 STOP 模式和 HALT 模式期间看门狗定时器的操作，参见 8.4.3 STOP 模式下看门狗定时器的操作和 8.4.4 HALT 模式下看门狗定时器的操作。

状态转换图如下所示。

图 8-5. 当“低速内部振荡器可由软件停止”由选项字节选择时的状态转换图



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

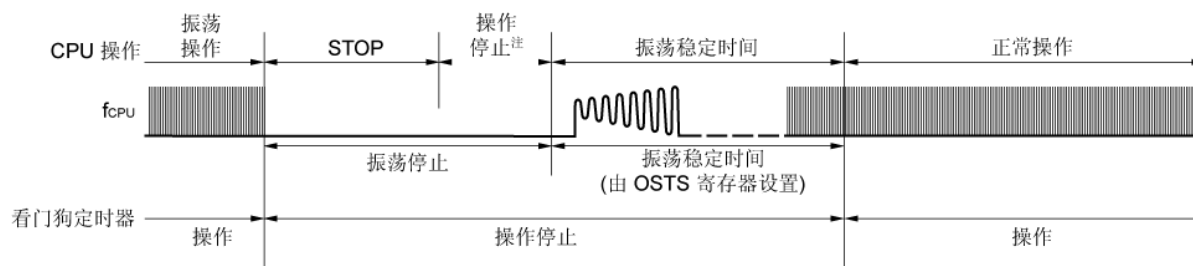
在 STOP 指令执行期间，看门狗定时器停止计数，无论系统时钟或低速内部振荡时钟是否被使用。

(1) STOP 指令执行时，当看门狗定时器操作时钟是系统时钟（fx）时

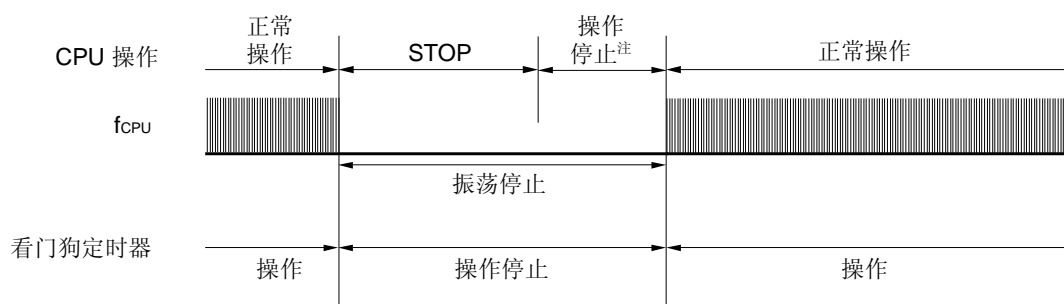
当 STOP 指令被执行时，看门狗定时器的操作被停止。STOP 模式被释放后，操作停止 $34\ \mu\text{s}$ （典型值）（在晶体 / 陶瓷振荡的情况下，操作停止后等待由振荡稳定时间选择寄存器（OSTS）设置的振荡稳定时间），然后计数将使用操作停止前的操作时钟再次开始计数。这时，计数器不被清零而是保持其值。

图 8-6. STOP 模式下的操作
（WDT 操作时钟：到周边硬件的时钟）

<1> CPU 时钟：晶体 / 陶瓷振荡时钟



<2> CPU 时钟：低速内部振荡时钟或外部时钟输入

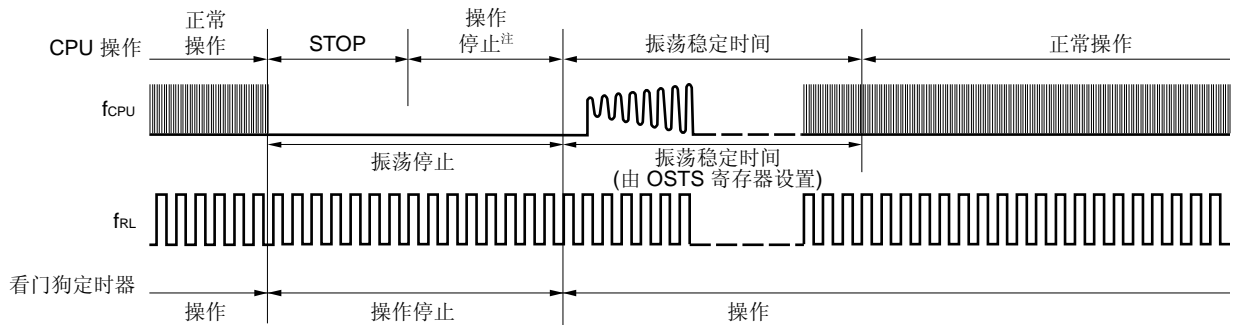
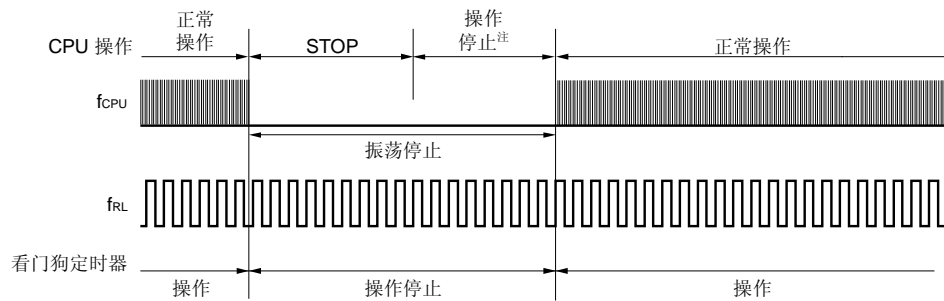


注 操作停止时间为 $17\ \mu\text{s}$ （最小值）， $34\ \mu\text{s}$ （典型值）和 $67\ \mu\text{s}$ （最大值）。

(2) STOP 指令执行时，当看门狗定时器操作时钟是低速内部振荡时钟（f_{RL}）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，操作停止 34 μs （典型值），然后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 8-7. STOP 模式下的操作
(WDT 操作时钟：低速内部振荡时钟)

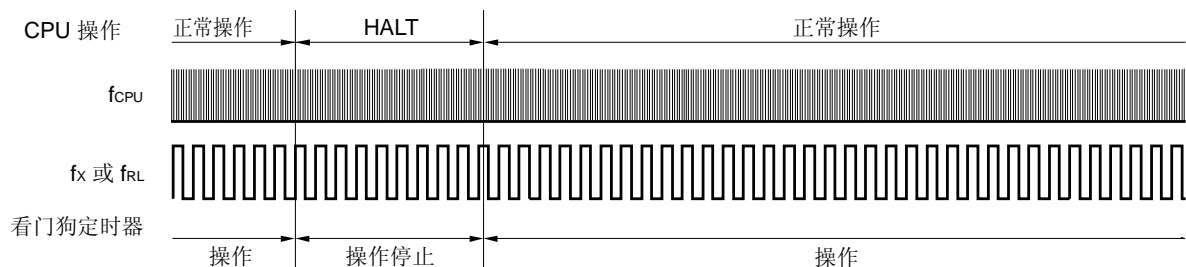
<1> CPU 时钟：晶体 / 陶瓷振荡时钟**<2> CPU 时钟：低速内部振荡时钟或外部时钟输入**

注 操作停止时间为 17 μs （最小值），34 μs （典型值）和 67 μs （最大值）。

8.4.4 HALT 模式下看门狗定时器的操作（当“低速内部振荡器可由软件停止”由选项字节选择时）

在 HALT 指令执行期间，看门狗定时器停止计数，无论看门狗定时器的操作时钟是系统时钟（f_x）还是低速内部振荡时钟（f_{RL}）。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 8-8. HALT 模式下的操作



第九章 中断功能

9.1 中断功能类型

存在两种类型的中断：可屏蔽中断和复位。

- **可屏蔽中断**

这些中断受屏蔽控制。当一个中断请求发生时，待机释放信号发生，并且如果中可以被响应，对应于向量表地址写入的地址的程序被执行（向量中断服务程序）。当几个中断请求被同时产生时，处理按照向量中断服务程序的优先级顺序执行。关于优先级顺序的详细情况，参见表 9-1。

存在可屏蔽中断的四个内部源和两个外部源。

- **复位**

CPU 和 SFR 通过复位信号返回到它们的初始状态。复位信号发生的原因如表 9-1 所示。

当一个复位信号发生时，程序从写入到地址 0000H 和 0001H 中的地址处的程序开始执行

9.2 中断源和配置

总共 6 个可屏蔽中断源和四个复位源（参见表 9-1）。

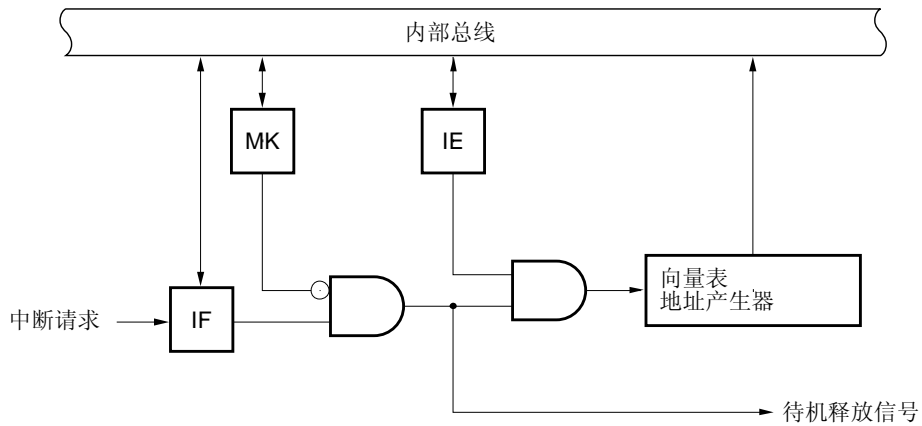
表 9-1. 中断源

中断类型	优先级 ^{注1}	中断源		内部 / 外部	矢量表地址	基本结构类型 ^{注2}
		名称	触发			
可屏蔽	1	INTLVI	低电压检测 ^{注3}	内部	0006H	(A)
	2	INTP0	引脚输入边沿检测	外部	0008H	(B)
	3	INTP1			000AH	
	4	INTTMH1	TMH1 和 CMP01 匹配（当比较寄存器被指定时）	内部	000CH	(A)
	5	INTTM000	TM00 和 CR000 匹配（当比较寄存器被指定时），TI010 引脚的有效沿检测（当捕获寄存器被指定时）		000EH	
	6	INTTM010	TM00 和 CR010 匹配（当比较寄存器被指定时），TI000 引脚的有效沿检测（当捕获寄存器被指定时）		0010H	
复位	-	RESET	复位输入	-	0000H	-
		POC	上电清零			
		LVI	低电压检测 ^{注4}			
		WDT	WDT 溢出			

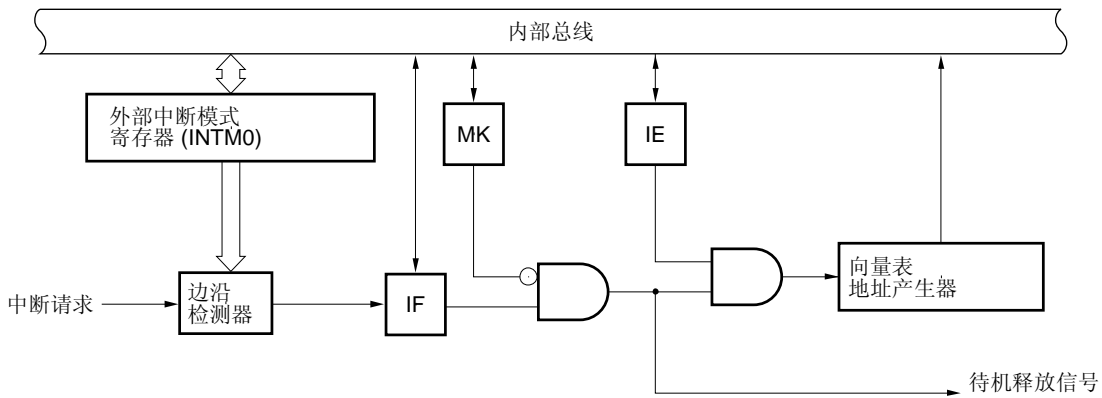
- 注
1. 优先级时当几个可屏蔽中断请求同时产生时的向量中断服务程序的优先级顺序。1 是最高优先级，6 是最低优先级。
 2. 基本配置类型 (A) 和 (B) 对应于图 9-1 中的 (A) 和 (B)。
 3. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) = 0 被选择时。
 4. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) = 1 被选择时。

图 9-1. 中断功能的基本配置

(A) 内部可屏蔽中断



(B) 外部可屏蔽中断



IF: 中断请求标志

IE: 允许中断标志

MK: 中断屏蔽标志

9.3 中断功能控制寄存器

中断功能由以下四类寄存器控制。

- 中断请求标志寄存器 0 (IF0)
- 中断屏蔽标志寄存器 0 (MK0)
- 外部中断模式寄存器 0 (INTM0)
- 程序状态字 (PSW)

表 9-2 列出了中断请求、对应的中断请求标志和中断屏蔽标志。

表 9-2. 中断请求信号和对应的标志

中断请求信号	中断请求标志	中断屏蔽标志
INTLVI	LVIIIF	LVIMK
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTTMH1	TMIFH1	TMMKH1
INTTM000	TMIF000	TMMK000
INTTM010	TMIF010	TMMK010

(1) 中断请求标志寄存器 0 (IF0)

当对应的中断请求被产生或者当执行指令时，中断请求标志将会被设为 1。当中断请求被响应或者当复位信号被输入时，通过执行一个指令，它被清除为 0。

IF0 通过 1 位或 8 位存储器操作指令来设置。

复位信号产生将 IF0 清除为 00H。

图 9-2. 中断请求标志寄存器 0 (IF0) 的格式

地址: FFE0H 复位后: 00H R/W

符号	7	<6>	<5>	<4>	<3>	<2>	<1>	0
IF0	0	TMIF010	TMIF000	TMIFH1	PIF1	PIF0	LVIIIF	0

xxIFx	中断请求标志
0	没有中断请求信号被产生。
1	一个中断请求信号被产生；中断请求状态。

注意事项 因为 P21 和 P32 具有外部中断输入的复用功能，当通过指定端口功能的输出模式更改输出电平时，一个中断请求标志被设置。因此，在使用输出模式前，中断屏蔽标志应该被设置为 1。

(2) 中断屏蔽标志寄存器 0 (MK0)

中断屏蔽标志用于允许和禁止对应的可屏蔽中断。

MK0 通过 1 位或 8 位存储器操作指令来设置。

复位信号产生将 MK0 设置为 FFH。

图 9-3. 中断屏蔽标志寄存器 0 (MK0) 的格式

地址: FFE4H 复位后: FFH R/W

符号	7	<6>	<5>	<4>	<3>	<2>	<1>	0
MK0	1	TMMK010	TMMK000	TMMKH1	PMK1	PMK0	LVIMK	1

××MK×	中断服务控制
0	允许中断服务
1	禁止中断服务

注意事项 因为 P21 和 P32 具有外部中断输入的复用功能，当通过指定端口功能的输出模式更改输出电平时，一个中断请求标志被设置。因此，在使用输出模式前，中断屏蔽标志应该被设置为 1。

(3) 外部中断模式寄存器 0 (INTM0)

该寄存器用来设置 INTP0 和 INTP1 的有效沿。

INTM0 通过 8 位存储器操作指令来设置。

复位信号产生将 INTM0 清除为 00H。

图 9-4. 外部中断模式寄存器 0 (INTM0) 的格式

地址: FFECH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
INTM0	0	0	ES11	ES10	ES01	ES00	0	0

ES11	ES10	INTP1 有效沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升沿及下降沿

ES01	ES00	INTP0 有效沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升沿及下降沿

注意事项 1. 确保清除第 0、1、6 和 7 位为 0。

注意事项 2. 在设置 INTM0 寄存器之前，确保设置对应的中断屏蔽标志 ($\times\times\text{MK}\times = 1$) 来禁止中断。在设置 INTM0 寄存器之后，清除中断请求标志 ($\times\times\text{IF}\times = 0$)，然后清除中断屏蔽标志 ($\times\times\text{MK}\times = 0$)，这将允许中断。

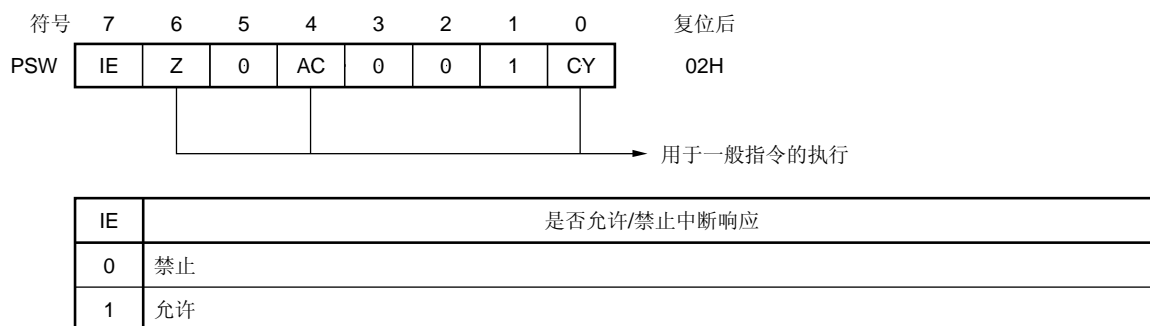
(4) 程序状态字 (PSW)

程序状态字用来为中断请求保持指令执行结果以及当前状态。用来允许和禁止可屏蔽中断的 IE 标志被映射到 PSW。

PSW 可以以 8 位为单位读取和写入，也可使用位操作指令和专用指令 (EI 和 DI) 来执行操作。当一个向量中断被响应时，PSW 被自动保存到堆栈中并且 IE 标志被复位为 0。

复位信号产生将 PSW 设置为 02H。

图 9-5. 程序状态字 (PSW) 配置



9.4 中断服务操作

9.4.1 可屏蔽中断请求响应操作

当中断请求标志被设为 1 且对应的中断屏蔽标志被清除为 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 时)，那么请求将作为向量中断被响应。

在可屏蔽中断请求被产生之后开始向量中断复位需要的时间如表 9-3 所示。

关于中断请求响应时序，参见图 9-7 和 9-8。

表 9-3. 从可屏蔽中断请求产生到服务之间的时间

最短时间	最长时间 ^注
9 个时钟	19 个时钟

注 当中断请求在紧接 BT 和 BF 指令之前被产生时，等待时间最长。

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

当两个或更多可屏蔽中断请求被同时产生时，它们从具有最高优先级的中断请求开始被响应。

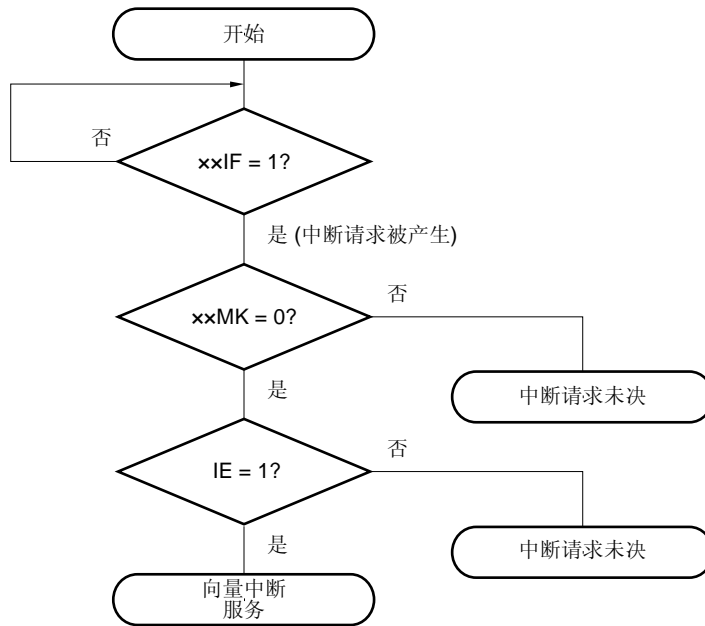
当一个未决中断可以被响应的状态被设置时，它被响应。

图 9-6 显示了中断请求响应的算法。

当可屏蔽中断请求被响应时，PSW 和 PC 的内容将会按照顺序被保存到堆栈中，IE 标志被复位为 0，而每个中断请求对应的向量表中的数据被加载到 PC 并执行跳转。

要从中断服务程序返回，请使用 RETI 指令。

图 9-6. 中断请求响应处理算法

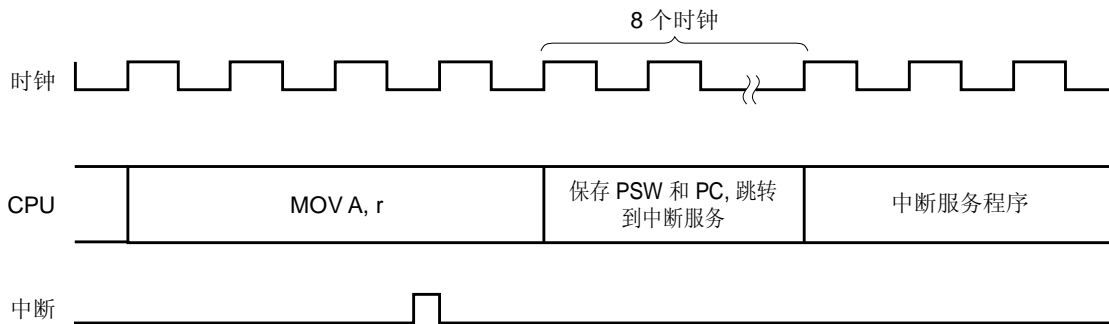


xxIF: 中断请求标志

xxMK: 中断屏蔽标志

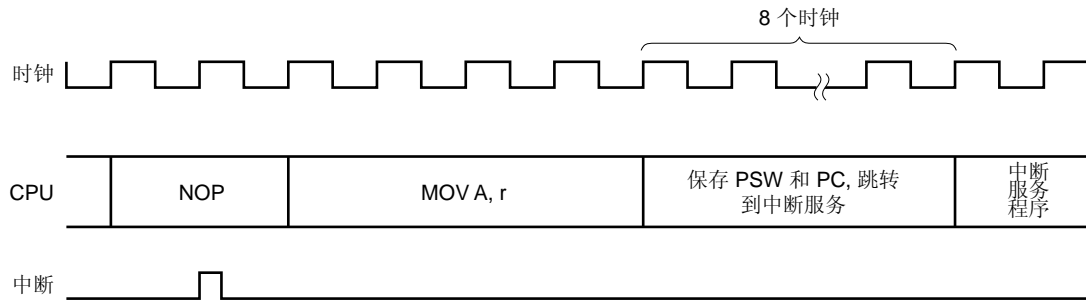
IE: 控制可屏蔽中断请求响应的标志 (1=允许, 0=禁止)

图 9-7. 中断请求响应时序 (MOV A, r 的示例)



如果中断请求标志 (xxIF) 在正在执行的指令时钟 n ($n = 4$ 到 10) 变为 $n - 1$ 之前被设置, 中断在正在执行的指令完成后被响应。图 9-7 显示了 8 位数据传送指令 MOV A, r 的中断请求响应时序的示例。因为指令执行 4 个时钟, 如果中断在取指开始后 3 个时钟发生, 中断响应处理将在 MOV A, r 指令被执行后执行。

图 9-8. 中断请求响应时序（当中断请求标志在指令执行期间的最后一个时钟时被设置）



如果中断请求标志 ($\times\times$ IF) 在指令的最后一个时钟被设置，中断响应处理在下一个指令被执行后开始。

图 9-8 显示了中断请求标志在 NOP（2 时钟指令）的第二个时钟时被设置的中断请求响应时序的示例。在这种情况下，MOV A, r 指令在 NOP 指令后执行，然后中断响应处理被执行。

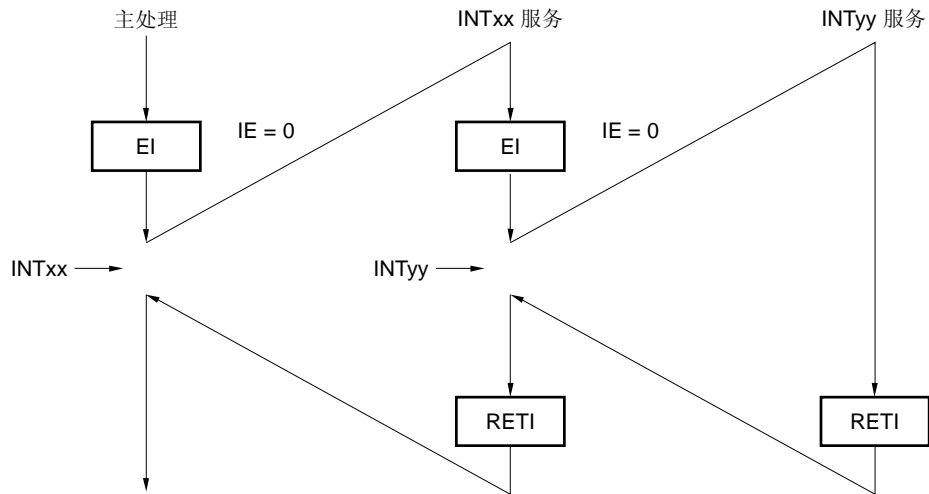
注意事项 在中断请求标志寄存器 0 (IF0) 或中断屏蔽标志寄存器 0 (MK0) 被访问期间，中断请求将保持未决。

9.4.2 多重中断服务

要执行多重中断服务，即一个中断被服务期间另一个中断被响应，中断屏蔽功能必须被用来屏蔽具有低优先级的中断。

图 9-9. 多重中断的示例 (1 / 2)

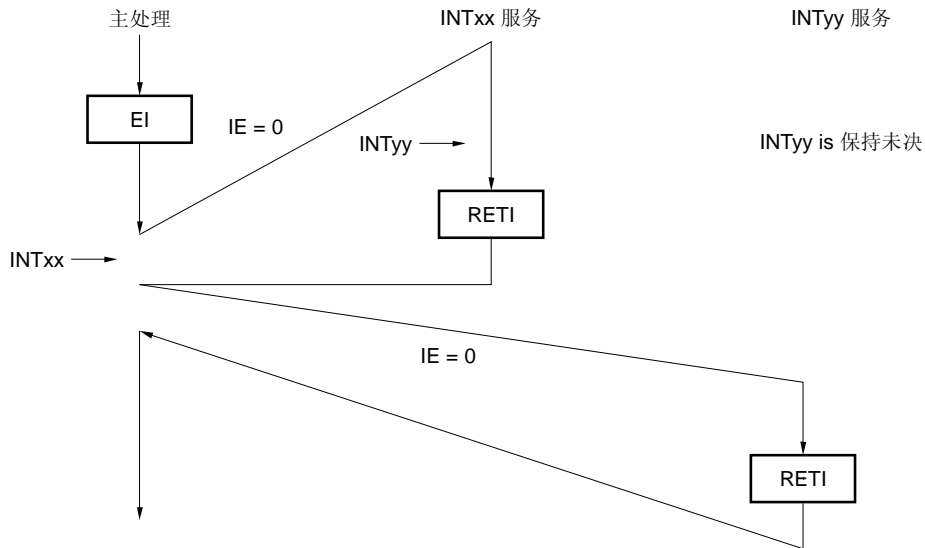
例 1. 多重中断被响应



在中断 INTxx 服务期间，中断请求 INTyy 被响应，并且多重中断被产生。在每个中断请求响应之前，EI 指令被发出，中断屏蔽被释放，并且中断请求响应允许状态被设置。

注意事项 对于低优先级的中，多重中断也可以被响应。

例 2. 由于中断未被允许多重中断未被产生



因为在中断 INTxx 服务期间不允许中断 (EI 指令没有发出)，中断请求 INTyy 不会被响应，并且多重中断服务也不会被产生。INTyy 请求保持未决并且在 INTxx 服务之后响应被执行。

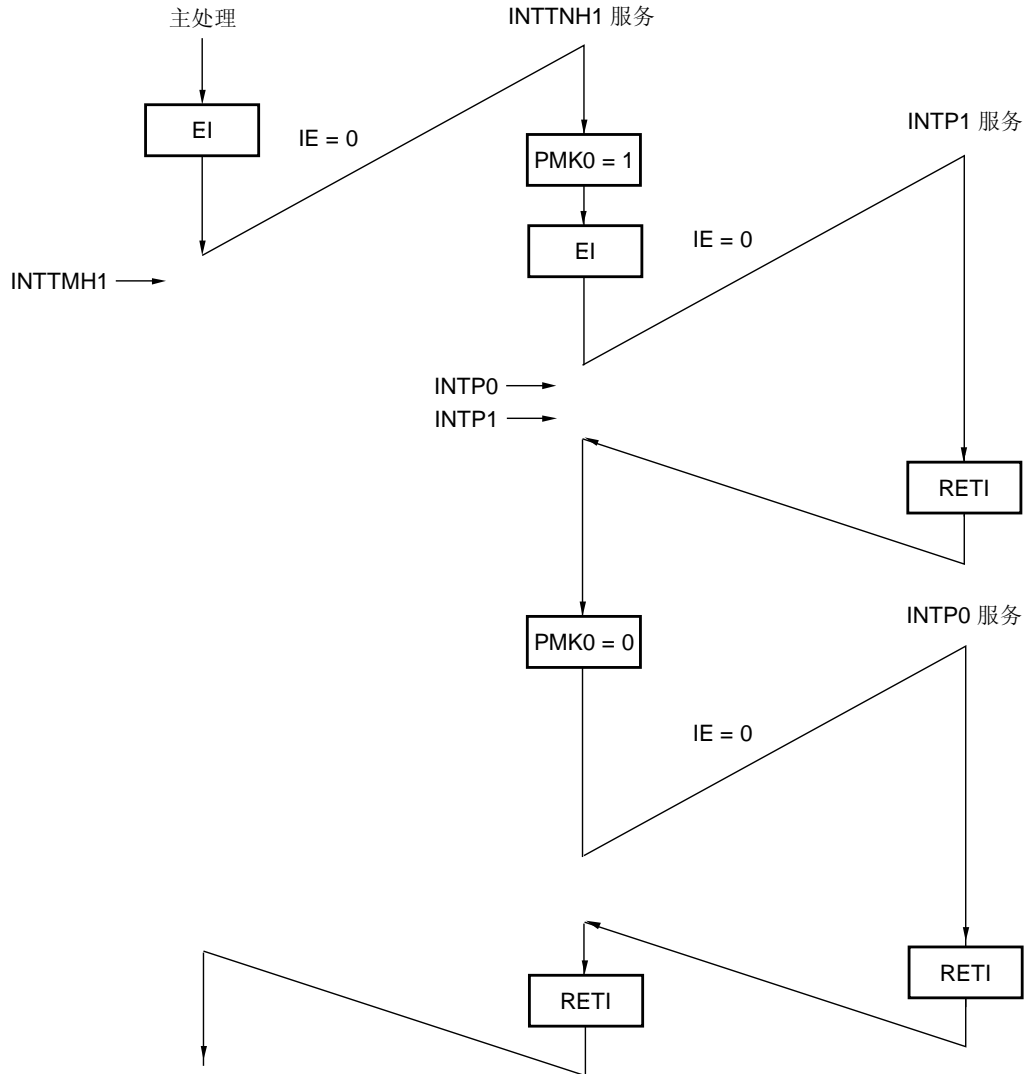
IE = 0: 禁止中断请求响应。

图 9-9. 多重中断的示例 (2 / 2)

例 3. 优先级由多重中断控制

对 INTP0、INTP1 和 INTTMH1 设置向量中断允许状态。

(中断优先级 INTP0 > INTP1 > INTTMH1 (参阅表 9-1))



在中断 INTTMH1 服务中，服务被执行以便 INTP1 中断具有优先权，因为 INTP0 中断首先被屏蔽。随后，一旦 INTP0 的中断屏蔽被释放，INTP0 处理通过多重中断被执行。

IE = 0: 禁止中断请求响应。

9.4.3 中断请求未决

即使中断请求（可屏蔽中断和外部中断）在指令执行期间被产生，一些指令可能保持未决的指令请求的响应，直到下一个指令执行完成。以下表示这样的指令（中断请求未决指令）。

- 中断请求标志寄存器 0 (IF0) 的操作指令
- 中断屏蔽标志寄存器 0 (MK0) 的操作指令

第十章 待机功能

10.1 待机功能和配置

10.1.1 待机功能

表 10-1. 每种操作状态中操作时钟的关系

操作模式 \ 状态	低速内部振荡器		系统时钟	提供给周边硬件的时钟	
	注 1	注 2			
		LSRSTOP = 0			LSRSTOP = 1
复位	停止		停止	停止	
STOP	振荡	振荡 ^{注3}	停止		
HALT					振荡

- 注
1. 当通过选项字节对低速内部振荡器选择“不能被停止”时。
 2. 当低速内部振荡器“可由软件停止”被选择时，低速内部振荡器的振荡可由 LSRSTOP 停止。
 3. 如果看门狗定时器的操作时钟是低速内部振荡时钟，看门狗定时器被停止。

注意事项 只有当通过选项字节对低速内部振荡器选择“可由软件停止”时，LSRSTOP 设置才有效。

备注 LSRSTOP: 低速内部振荡模式寄存器 (LSRCM) 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。系统时钟振荡器的振荡继续。如果低速内部振荡器在 HALT 模式被设置之前正在操作，低速内部振荡器的时钟的振荡继续（参阅表 10-1。低速内部振荡时钟的振荡（它是否可由软件停止）由选项字节设置）。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产生后立即恢复操作以及执行经常间隙操作是有效的。

(2) STOP 模式

STOP 指令执行设置了 STOP 模式。在 STOP 模式下，系统时钟振荡器停止，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操作。然而，当操作停止时间^注在 STOP 模式被释放后产生时，如果处理必须立即由中断请求启动，请选择 HALT 模式（因为当晶体 / 陶瓷振荡被使用时，稳定振荡的额外等待时间过去）。

注 操作停止时间为 17 μs （最小值）、34 μs （典型值）和 67 μs （最大值）。

在以上两种模式的任一种模式中，寄存器、标志以及数据存储器中的所有内容在等待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. 当转至 STOP 模式时，应确保在执行 STOP 指令前停止周边硬件操作（工作于低速内部振荡时钟的周边硬件除外）。
 2. 在 STOP 模式下，如果在 STOP 模式被设置之前低速内部振荡器正在操作，低速内部振荡时钟的振荡不能被停止（参阅表 10-1）。

10.1.2 待机期间使用的检测器

待机模式被释放后的振荡稳定时间由振荡稳定时间选择寄存器（OSTS）控制。

备注 关于开始、停止或选择时钟的寄存器，见第五章 时钟产生器。

(1) 振荡稳定时间选择寄存器（OSTS）

当 STOP 模式被释放时，该寄存器用于选择从振荡器提供的时钟的振荡稳定时间。只有当晶体 / 陶瓷振荡时钟被选作系统时钟时并且 STOP 模式释放后，由 OST S 设置的等待时间才有效。如果高速内部振荡或外部时钟输入被选作系统时钟源，不需要等待时间。

系统时钟振荡器和上电或复位释放后的振荡稳定时间由选项字节来选择。关于详细情况，参阅第十四章 选项字节。

OSTS 通过 8 位存储器操作指令来设置。

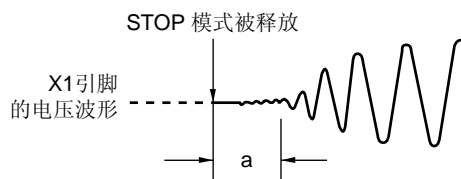
图 10-1. 振荡稳定时间选择寄存器（OSTS）的格式

地址：FFF4H，复位后：不确定，R/W

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

OSTS1	OSTS0	振荡稳定时间的选择
0	0	$2^{10}/f_x$ (102.4 μ s)
0	1	$2^{12}/f_x$ (409.6 μ s)
1	0	$2^{15}/f_x$ (3.27 ms)
1	1	$2^{17}/f_x$ (13.1 ms)

- 注意事项**
- 要设置并且然后释放 STOP 模式，请按照下面设置振荡稳定时间。
期望的谐振器的振荡稳定时间 \leq 由 OST S 设置的振荡稳定时间
 - STOP 模式释放后的等待时间不包含从 STOP 模式释放到时钟振荡开始（下图中的“a”）的时间，无论 STOP 模式是否由复位信号产生还是中断产生来释放。



- 上电或复位释放后的振荡稳定时间由选项字节来选择。关于详细情况，参阅第十四章 选项字节。

- 备注**
- () : $f_x = 10$ MHz
 - 通过检查要使用的谐振器的特性来确定谐振器的振荡稳定时间。

10.2 待机功能操作

10.2.1 HALT 模式

(1) HALT 模式

HALT 模式通过执行 HALT 指令来设置。

以下显示了在 HALT 模式中的操作状态。

注意事项 因为中断请求信号被用于清除等待模式，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清除的中断源时，如果中断请求被设置，那么等待将会立即被清除。

表 10-2. HALT 模式中的操作状态

HALT 模式的设置		低速内部振荡器不能被停止 ^注 。	低速内部振荡器可以被停止 ^注 。	
			当低速内部振荡继续时	当低速内部振荡停止时
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）		保持 HALT 模式被设置前的状态。		
16 位定时器 / 事件计数器 00		可操作		
8 位定时器 H1	设置计数时钟为 f_{XP} 到 $f_{XP}/2^{12}$	可操作		
	设置计数时钟为 $f_{RL}/2^7$	可操作	可操作	操作停止。
看门狗计时器	“系统时钟”选择操作时钟	禁止设置。	操作停止。	
	“低速内部振荡时钟”选择操作时钟	可操作 (操作继续)	操作停止。	
上电清零电路		总是操作。		
低压检测电路		可操作		
外部中断		可操作		

注 通过选项字节对低速内部振荡器选择“不能被停止”或“由软件停止”（关于选项字节，参见第十四章 选项字节）。

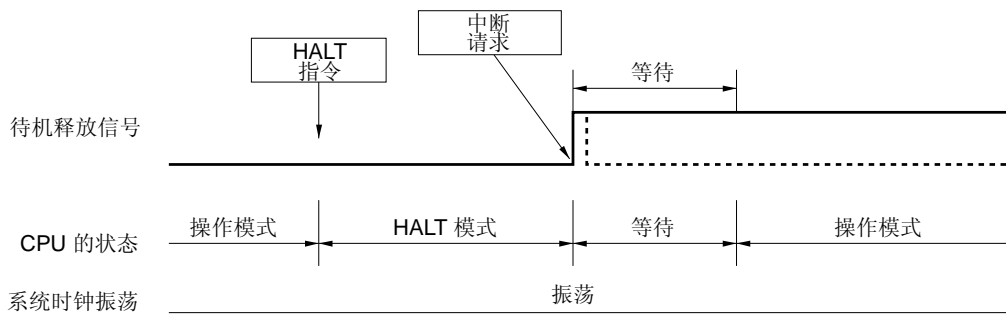
(2) HALT 模式释放

HALT 模式可以通过以下两种源来释放。

(a) 通过未屏蔽中断请求释放

当产生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 10-2. 通过产生中断请求来释放 HALT 模式



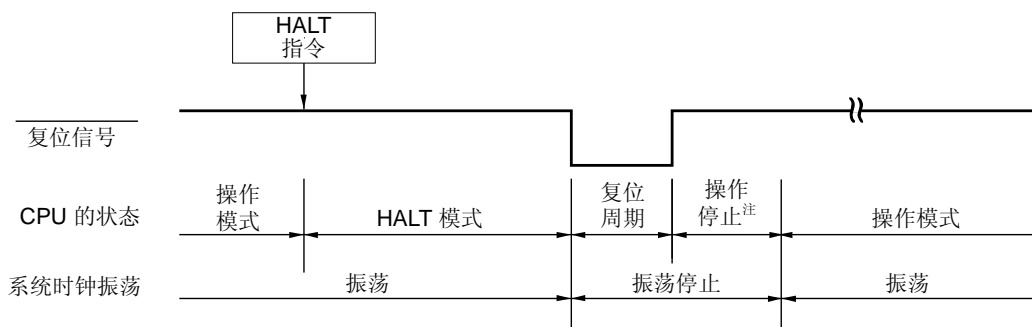
- 备注**
1. 虚线表明已经释放了等待状态的中断请求被响应时的情况。
 2. 等待时间如下：
 - 当向量中断服务被执行时：11 到 13 个时钟
 - 当向量中断服务没有被执行时：3 到 5 个时钟

(b) 通过复位信号产生来释放

当复位信号被输入时，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。

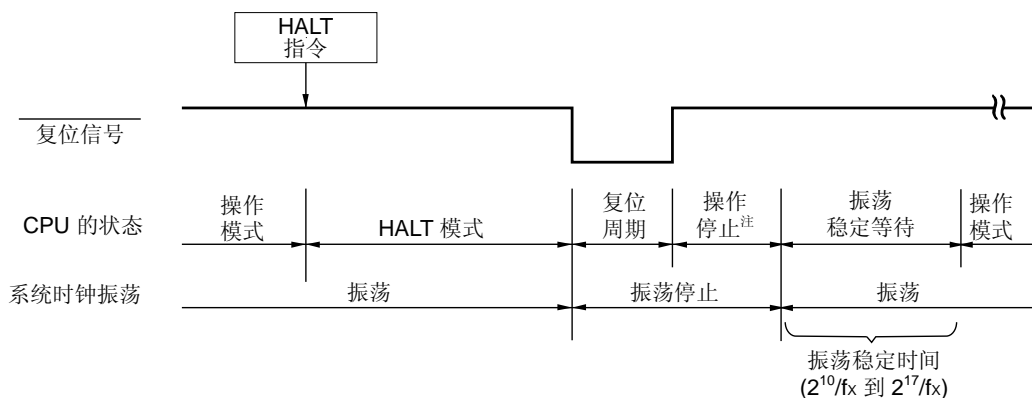
图 10-3. 通过产生复位信号来释放 HALT 模式

(1) 当 CPU 时钟为高速内部振荡时钟或外部输入时钟时



注 操作被停止（277 μs （最小值）、544 μs （典型值）和 1.075 ms（最大值）），因为选项字节被引用。

(2) 当 CPU 时钟为晶体 / 陶瓷振荡时钟时



注 操作被停止（276 μs （最小值）、544 μs （典型值）和 1.074 ms（最大值）），因为选项字节被引用。

备注 fx: 系统时钟振荡频率

表 10-3. HALT 模式中中断请求响应操作

释放源	MK _{xx}	IE	操作
可屏蔽中断请求	0	0	下一地址指令执行
	0	1	中断服务执行
	1	×	HALT 模式保持
复位信号产生	-	×	复位处理

×: 不关注

10.2.2 STOP 模式

(1) STOP 模式设置及操作状态

STOP 模式通过执行 STOP 指令来设置。

注意事项 因为中断请求信号被用于清除等待模式，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，那么等待将会立即被清除。因此，在 STOP 模式下，在 STOP 指令被执行并且操作被停止 34 μs （典型值）后，正常操作模式被恢复（当晶体 / 陶瓷振荡被使用时，在由振荡稳定时间选择寄存器（OSTS）设置的稳定振荡的额外等待时间过去后）。

以下显示了 STOP 模式中的操作状态。

表 10-4. STOP 模式中的操作状态

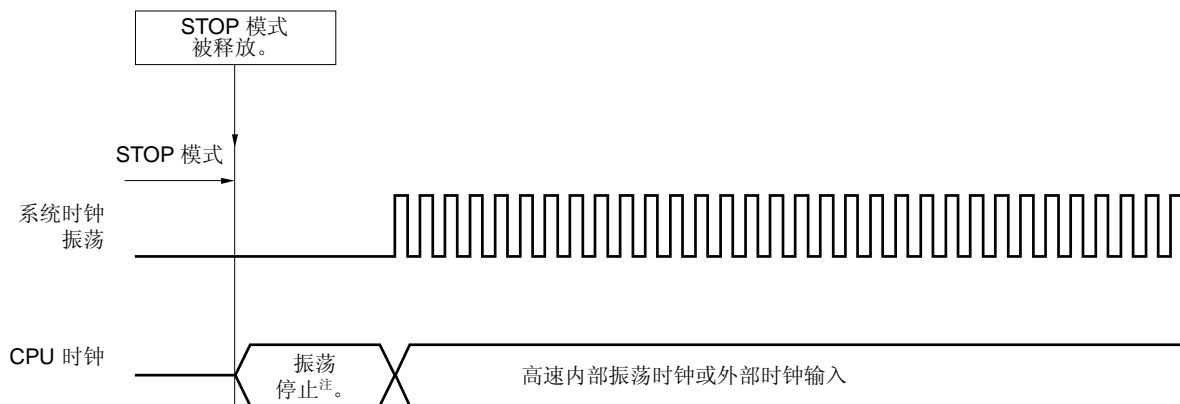
STOP 模式的设置		低速内部振荡器不能被停止 ^注 。	低速内部振荡器可以被停止 ^注 。	
			当低速内部振荡继续时	当低速内部振荡停止时
项目				
系统时钟		振荡停止。		
CPU		操作停止。		
端口（锁存）		保持 STOP 模式被设置前的状态。		
16 位定时器 / 事件计数器 00		操作停止。		
8 位定时器 H1	设置计数时钟为 $f_{XP}/2^{12}$	操作停止。		
	设置计数时钟为 $f_{RL}/2^7$	可操作	可操作	操作停止。
看门狗计时器	“系统时钟”选择操作时钟	禁止设置。	操作停止。	
	“低速内部振荡时钟”选择操作时钟	可操作 (操作继续)	操作停止。	
上电清零电路		总是操作。		
低压检测电路		可操作		
外部中断		可操作		

注 通过选项字节对低速内部振荡器选择不能被停止或由软件停止（关于选项字节，参见第十四章 选项字节）。

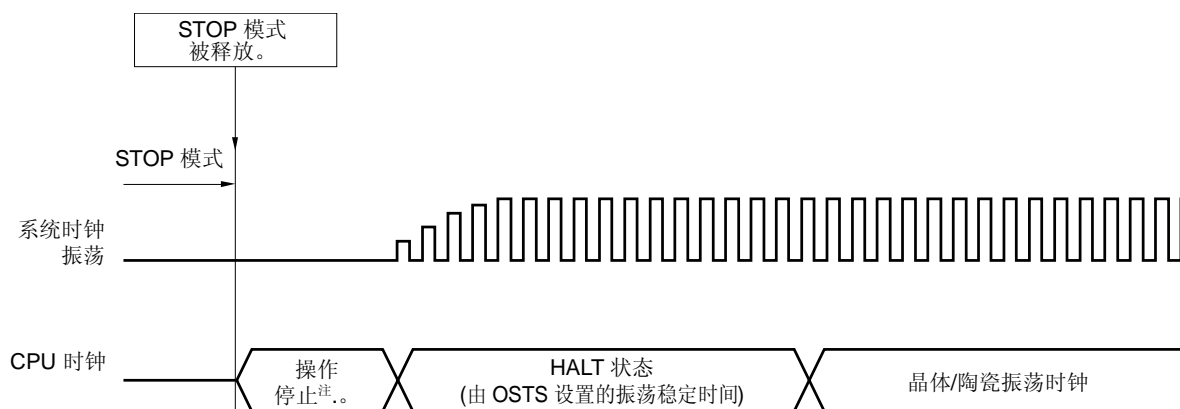
(2) STOP 模式释放

图 10-4. STOP 模式被释放时的操作时序

<1> 如果高速内部振荡时钟或外部输入时钟被选作要被提供的系统时钟



<2> 如果晶体 / 陶瓷振荡时钟被选作要被提供的系统时钟



注 操作停止时间为 17 μs (最小值)、34 μs (典型值) 和 67 μs (最大值)。

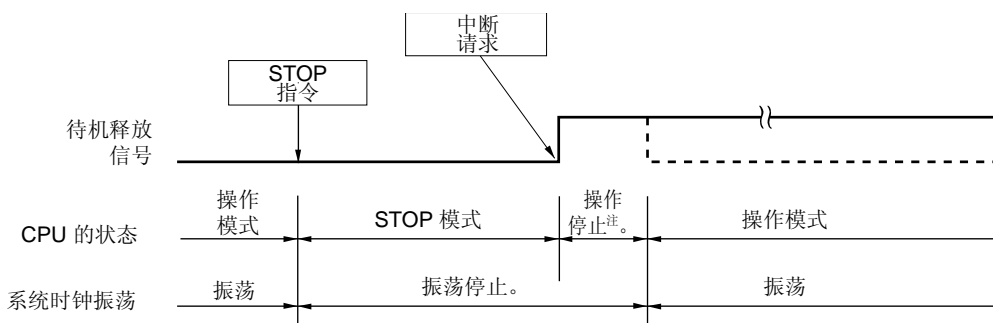
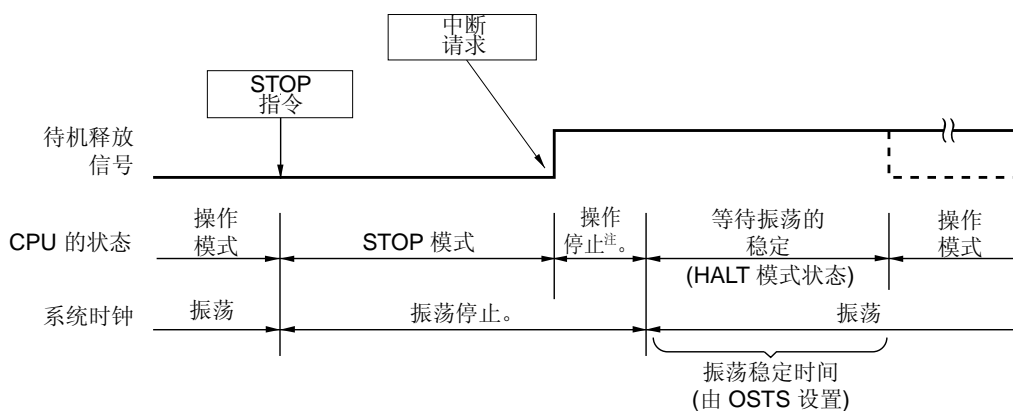
STOP 模式可以通过以下两种源来释放。

(a) 通过未屏蔽中断请求释放

当一个未屏蔽中断请求（8 位定时器 H1^注，低电压检测电路，外部中断请求）被产生时，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。

注 只有当设置计数时钟为 $f_{RU}/2^7$ 时

图 10-5. 通过产生中断请求来释放 STOP 模式

(1) 如果 CPU 时钟为高速内部振荡时钟或外部输入时钟**(2) 如果 CPU 时钟为晶体 / 陶瓷振荡时钟**

注 操作停止时间为 17 μs (最小值)、34 μs (典型值) 和 67 μs (最大值)。

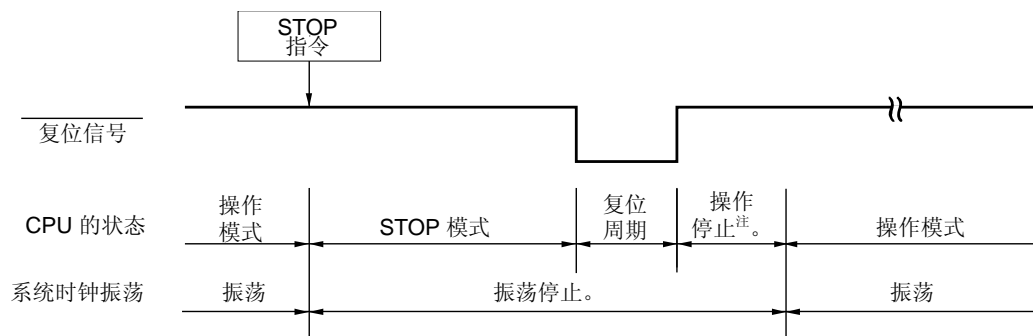
备注 虚线表明已经释放了等待状态的中断请求被响应时的情况。

(b) 通过复位信号产生来释放

当复位信号被输入时，STOP 模式被释放并且在振荡稳定事件过去后，复位操作被执行。

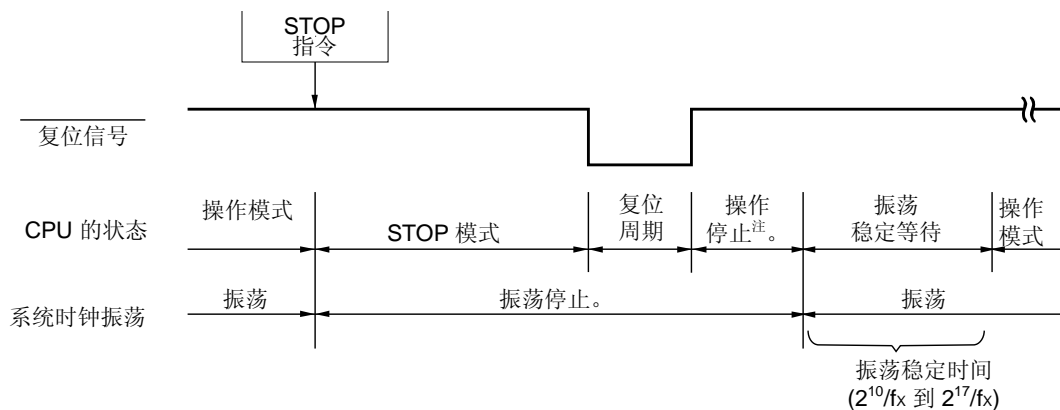
图 10-6. 通过产生复位信号来释放 STOP 模式

(1) 如果 CPU 时钟为高速内部振荡时钟或外部输入时钟



注 操作被停止（277 μs （最小值）、544 μs （典型值）和 1.075 ms（最大值）），因为选项字节被引用。

(2) 如果 CPU 时钟为晶体 / 陶瓷振荡时钟



注 操作被停止（276 μs （最小值）、544 μs （典型值）和 1.074 ms（最大值）），因为选项字节被引用。

备注 fx: 系统时钟振荡频率

表 10-5. STOP 模式下中断请求响应操作

释放源	MK _{xx}	IE	操作
可屏蔽中断请求	0	0	下一地址指令执行
	0	1	中断服务执行
	1	x	STOP 模式保持
复位信号产生	-	x	服务处理

x: 不关注

第十一章 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) 通过看门狗定时器所生成的内部复位
- (3) 通过比较电源电压和上电清零 (POC) 电路的检测电压的内部复位
- (4) 通过比较电源电压和低电源检测电路 (LVI) 的检测电压的内部复位

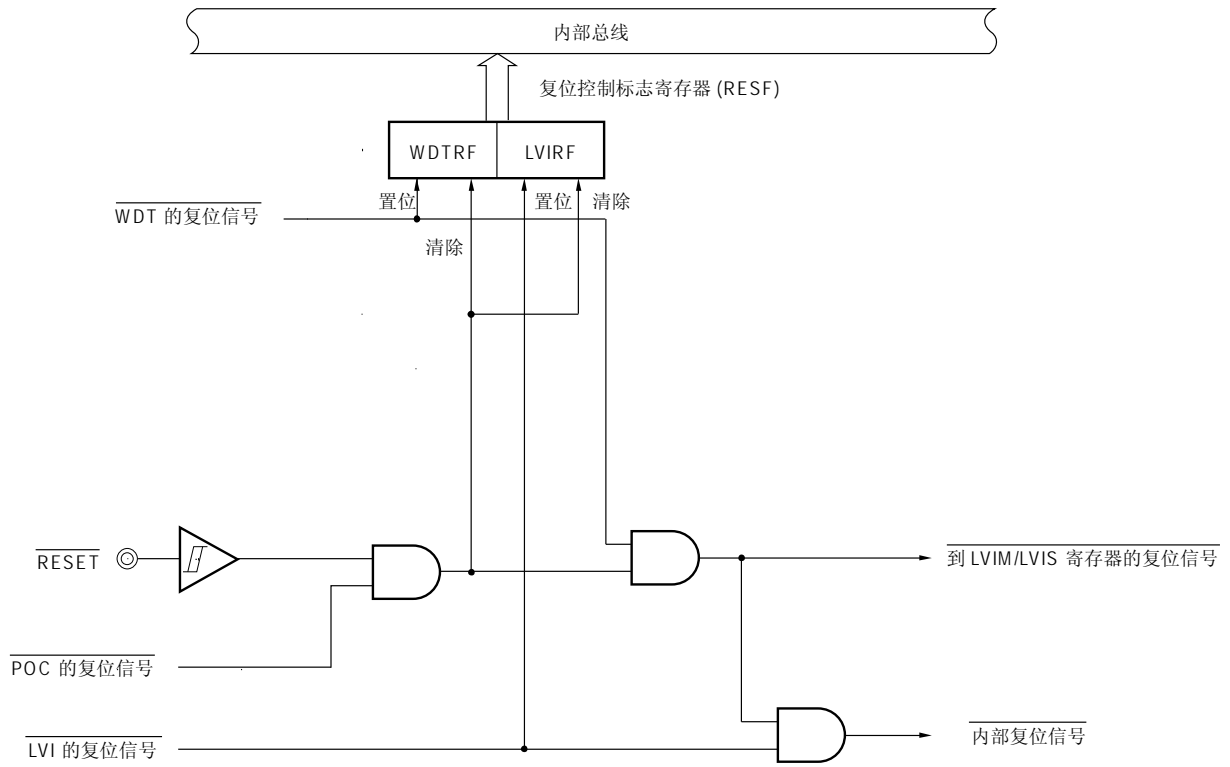
外部和内部复位没有功能差异。两种情况下, 当复位信号产生时程序从写入地址 0000H 和 0001H 的地址处的程序开始执行。

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 14-1 中的状态时, 复位被应用。在复位信号产生期间或在复位释放后的振荡稳定时间内, 每个端口为高阻抗。

当一个低电平被输入到 $\overline{\text{RESET}}$ 引脚时, 复位发生, 并且当一个高电平被输入到 $\overline{\text{RESET}}$ 引脚时, 复位被释放并且 CPU 在引用选项字节后开始程序执行 (如果晶体 / 陶瓷振荡被选择, 在选项字节被引用并且时钟振荡稳定时间过去后)。由看门狗定时器源产生的复位在复位后自动被释放, 并且 CPU 在引用选项字节后开始程序执行 (如果晶体 / 陶瓷振荡被选择, 在选项字节被引用并且时钟振荡稳定时间过去后)。(参见图 11-2 到 11-4)。由 POC 和 LVI 电路上电检测产生的复位在复位后当 $V_{DD} > V_{POC}$ 或 $V_{DD} > V_{LVI}$ 时自动被释放, 并且 CPU 在引用选项字节后开始程序执行 (如果晶体 / 陶瓷振荡被选择, 在选项字节被引用并且时钟振荡稳定时间过去后)。(参见第十二章 上电清零电路 以及第十三章 低电压检测电路)。

- 注意事项**
1. 为外部复位将 $2\mu\text{s}$ 或更高的低电平输入到 $\overline{\text{RESET}}$ 引脚中。
 2. 复位信号产生期间, 系统时钟和低速内部振荡时钟停止振荡。
 3. 当 $\overline{\text{RESET}}$ 引脚用作输入端口引脚 (P34) 时, 如果在由 POC 电路产生的复位被释放后并且选项字节被引用之前低电平被输入到 $\overline{\text{RESET}}$ 引脚, $\mu\text{PD78F9510}$ 、 $78F9511$ 、 $78F9512$ 将被再次复位。复位状态被保持直到高电平被输入到 $\overline{\text{RESET}}$ 引脚。

图 11-1. 复位功能框图

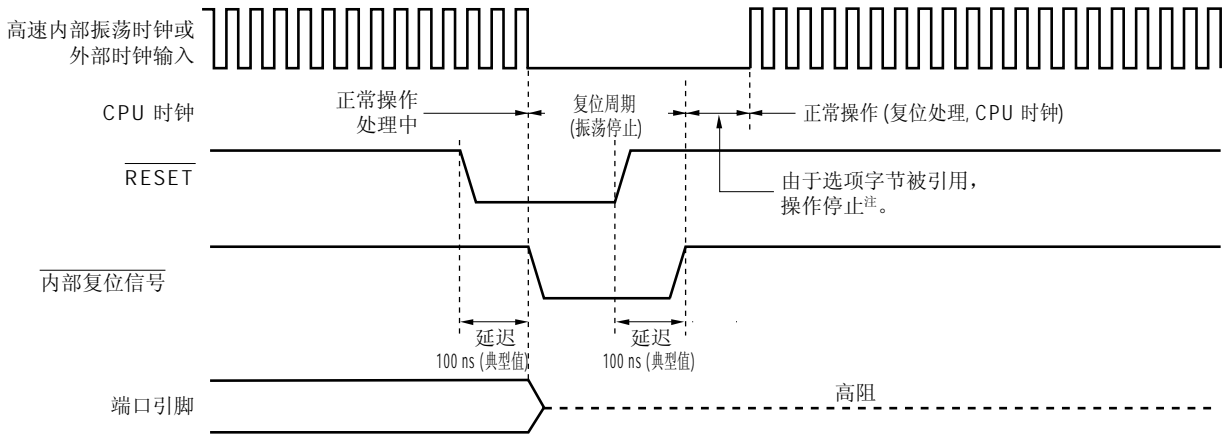


注意事项 LVI 电路不会被由 LVI 电路产生的内部复位信号复位。

- 备注**
1. LVIM: 低电压检测寄存器
 2. LVIS: 低电压检测电平选择寄存器

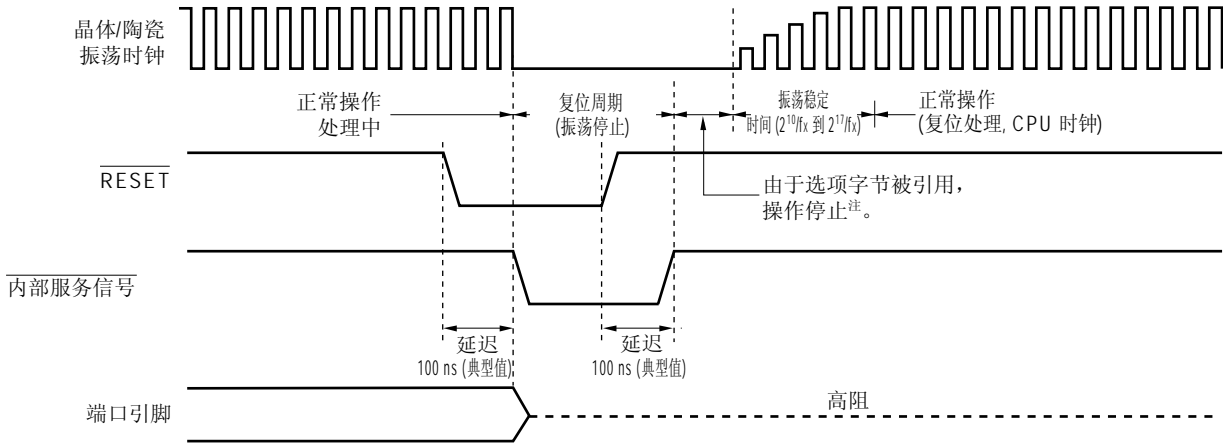
图 11-2. 通过 RESET 输入的复位时序

<1> 使用高速内部振荡时钟或外部时钟输入



注 操作停止时间为 277 μs (最小值)、544 μs (典型值) 和 1.075 ms (最大值)。

<2> 使用晶体 / 陶瓷振荡时钟

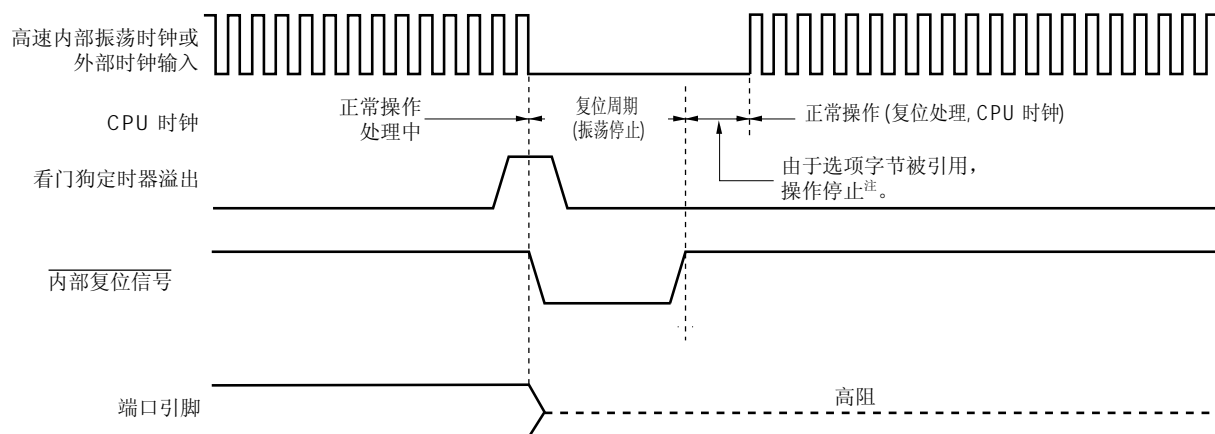


注 操作停止时间为 276 μs (最小值)、544 μs (典型值) 和 1.074 ms (最大值)。

备注 f_x : 系统时钟振荡频率

图 11-3. 由看门狗定时器溢出而复位的时序

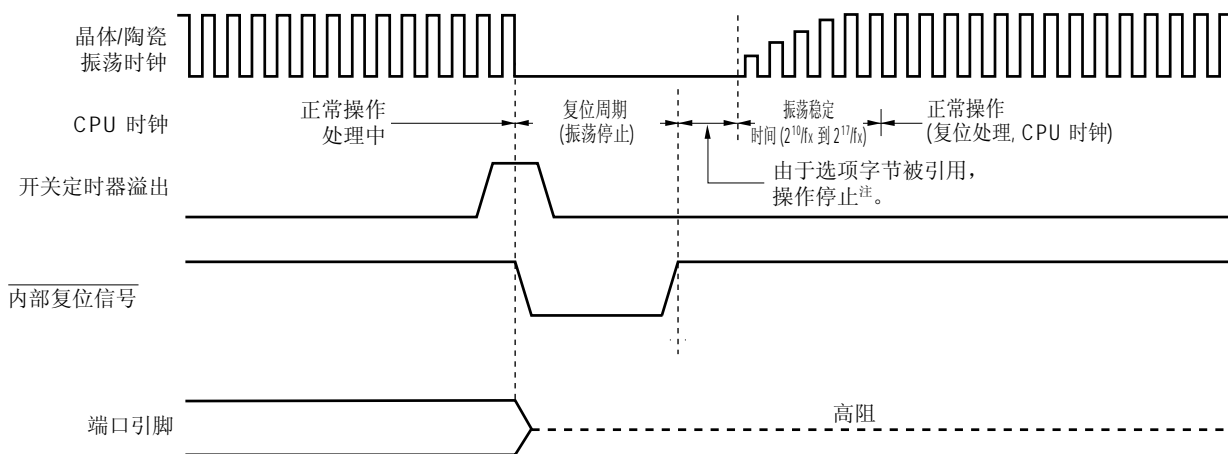
<1> 使用高速内部振荡时钟或外部时钟输入



注 操作停止时间为 277 μs (最小值)、544 μs (典型值) 和 1.075 ms (最大值)。

注意事项 在看门狗定时器的内部复位情况下，看门狗定时器也会被复位。

<2> 使用晶体 / 陶瓷振荡时钟



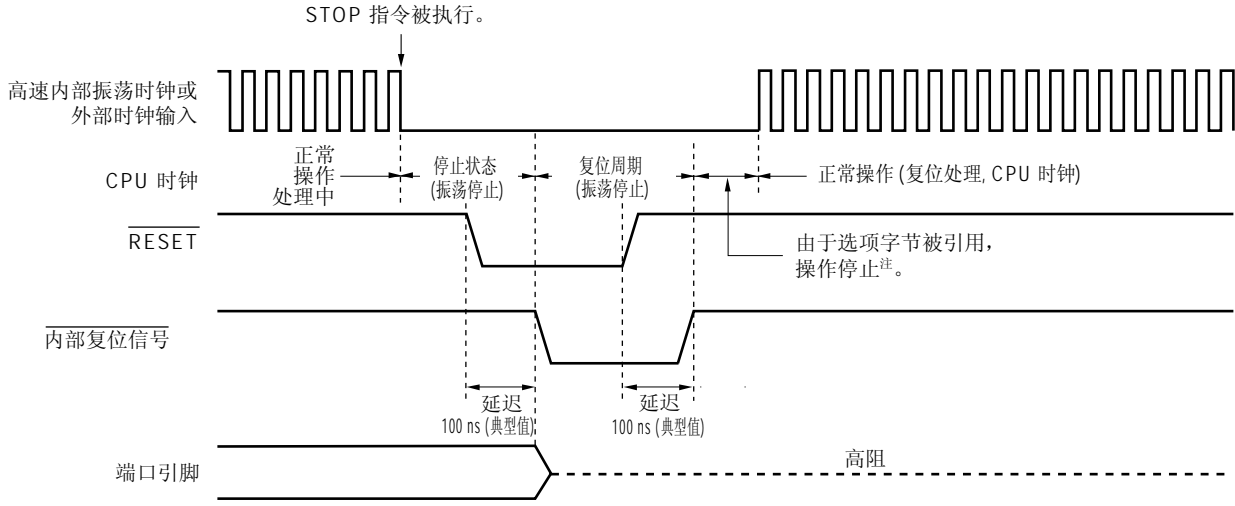
注 操作停止时间为 276 μs (最小值)、544 μs (典型值) 和 1.074 ms (最大值)。

注意事项 在看门狗定时器的内部复位情况下，看门狗定时器也会被复位。

备注 fx: 系统时钟振荡频率

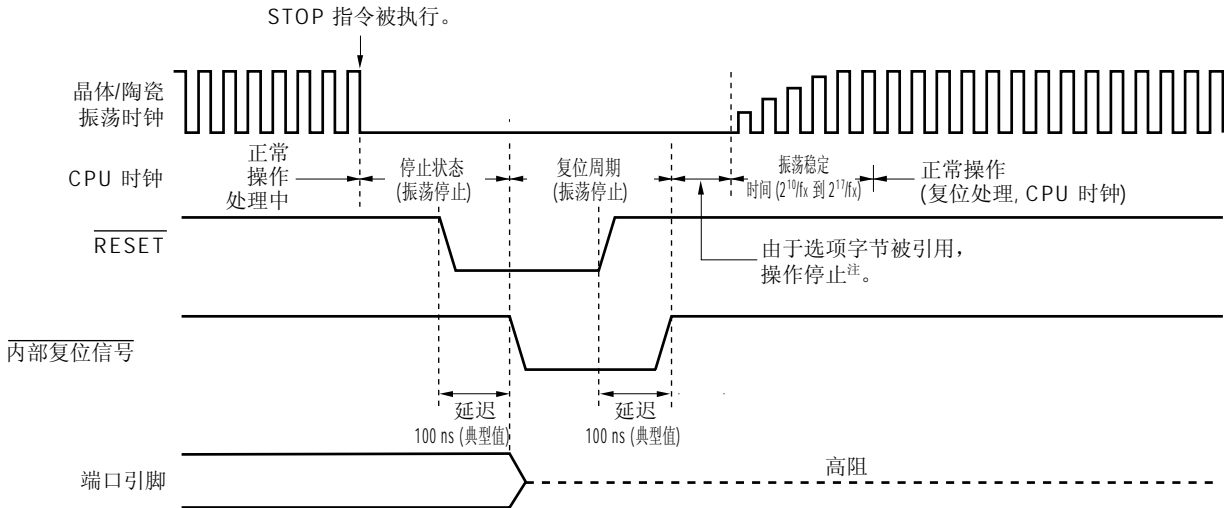
图 11-4. STOP 模式下通过 RESET 输入复位的时序

<1> 使用高速内部振荡时钟或外部时钟输入



注 操作停止时间为 277 μs (最小值)、544 μs (典型值) 和 1.075 ms (最大值)。

<2> 使用晶体 / 陶瓷振荡时钟



注 操作停止时间为 276 μs (最小值)、544 μs (典型值) 和 1.074 ms (最大值)。

备注 1. 关于上电清零电路和低电压检测电路的复位时序, 参见第十二章 上电清零电路 和 第十三章 低电压检测电路。

2. fx: 系统时钟振荡频率

表 11-1. 复位应答后的硬件状态 (1 / 2)

硬件		复位后的状态
程序计数器 (PC) ^{注1}		复位矢量表 (0000H 和 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 (P2 到 P4) (输出锁存)		00H
端口模式寄存器 (PM2 到 PM4)		FFH
上拉电阻选项寄存器 (PU2 到 PU4)		00H
处理器时钟控制寄存器 (PCC)		02H
预处理器时钟控制寄存器 (PPCC)		02H
低速内部振荡模式寄存器 (LSRCM)		00H
振荡稳定时间选择寄存器 (OSTS)		不确定
16 位定时器 00	定时器计数器 00 (TM00)	0000H
	捕获 / 比较寄存器 000, 010 (CR000, CR010)	0000H
	模式控制寄存器 00 (TMC00)	00H
	预分频模式寄存器 00 (PRM00)	00H
	捕获 / 比较控制寄存器 00 (CRC00)	00H
	定时器输出控制寄存器 00 (TOC00)	00H
8 位定时器 H1	比较寄存器 (CMP01, CMP11)	00H
	模式寄存器 1 (TMHMD1)	00H
看门狗定时器	模式寄存器 (WDTM)	67H
	允许寄存器 (WDTE)	9AH

- 注
1. 在复位信号产生期间和振荡稳定时间经过期间, 只有 PC 的内容不确定。其它硬件单元的状态保持不变。
 2. 在待机模式下, 复位后的状态被保持。

表 11-1. 复位应答后的硬件状态 (2 / 2)

硬件		复位后的状态
复位功能	复位控制标志寄存器 (RESF)	00H ^注
低电压检测	低电压检测寄存器 (LVIM)	00H ^注
	低电压检测电平选择寄存器 (LVIS)	00H ^注
中断	请求标志寄存器 (IFO)	00H
	屏蔽标志寄存器 (MKO)	FFH
	外部中断模式寄存器 (INTMO)	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)	不确定
	Flash 编程命令寄存器 (FLCMD)	00H
	Flash 地址指针 L (FLAPL)	不确定
	Flash 地址指针 H (FLAPH)	
	Flash 地址指针 H 比较寄存器 (FLAPHC)	00H
	Flash 地址指针 L 比较寄存器 (FLAPLC)	00H
	Flash 写缓存寄存器 (FLW)	00H

注 这些值根据复位源而变化，如下所示。

复位源		RESET 输入	通过 POC 复位	通过 WDT 复位	通过 LVI 复位
RESF	WDTRF	清除 (0)	清除 (0)	置位 (1)	保持
	LVIRF			保持	置位 (1)
LVIM		清除 (00H)	清除 (00H)	清除 (00H)	保持
LVIS					

11.1 确定复位源的寄存器

许多内部复位生成源存在于 μ PD78F9510、78F9511、78F9512 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。

RESET 输入、上电清零（POC）电路所产生的复位信号以及读取 RESF 可以将 RESF 设为 00H。

图 11-5. 复位控制标志寄存器（RESF）的格式

地址： FF54H 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	通过低电压检测电路（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 11-2 中。

图 11-2. 当复位请求生成时 RESF 的状态

	复位源	RESET 输入	通过 POC 复位	通过 WDT 复位	通过 LVI 复位
标志					
WDTRF		清除 (0)	清除 (0)	置位 (1)	保持
LVIRF				保持	置位 (1)

第十二章 上电清零电路

12.1 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（ $V_{POC} = 2.1\text{ V} \pm 1.1\text{ V}$ ），当 $V_{DD} < V_{POC}$ 时，产生内部复位信号。
- 比较电源电压（ V_{DD} ）和检测电压（ $V_{POC} = 2.1\text{ V} \pm 1.1\text{ V}$ ），当 $V_{DD} \geq V_{POC}$ 时，释放内部复位信号。

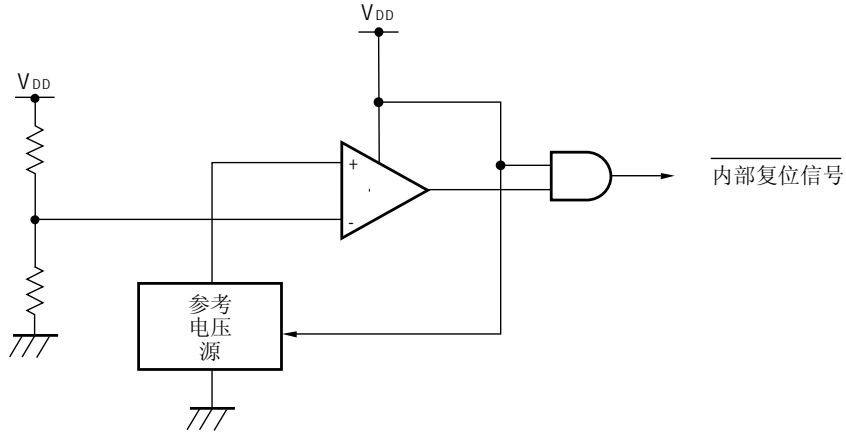
注意事项 1. 如果在 POC 电路中内部复位信号被产生，复位控制标志寄存器（RESF）被清除为 00H。
2. 因为 POC 电路的检测电压（ V_{POC} ）范围为 $2.1\text{ V} \pm 0.1\text{ V}$ ，请使用 2.2 到 5.5 V 范围内的电压。

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，见 第十一章 复位功能。

12.2 上电清零电路的配置

上电清零电路的框图如图 12-1 所示。

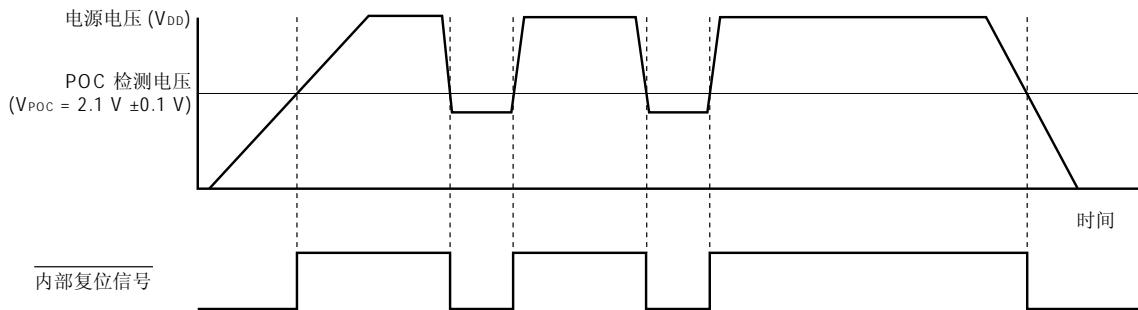
图 12-1. 上电清零电路的框图



12.3 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 ($V_{POC} = 2.1\text{ V} \pm 0.1\text{ V}$) 被比较，当 $V_{DD} < V_{POC}$ 时，内部复位信号被产生，而当 $V_{DD} \geq V_{POC}$ 时，内部复位被释放。

图 12-2. 上电清零电路中内部复位信号产生的时序



12.4 上电清零电路的注意事项

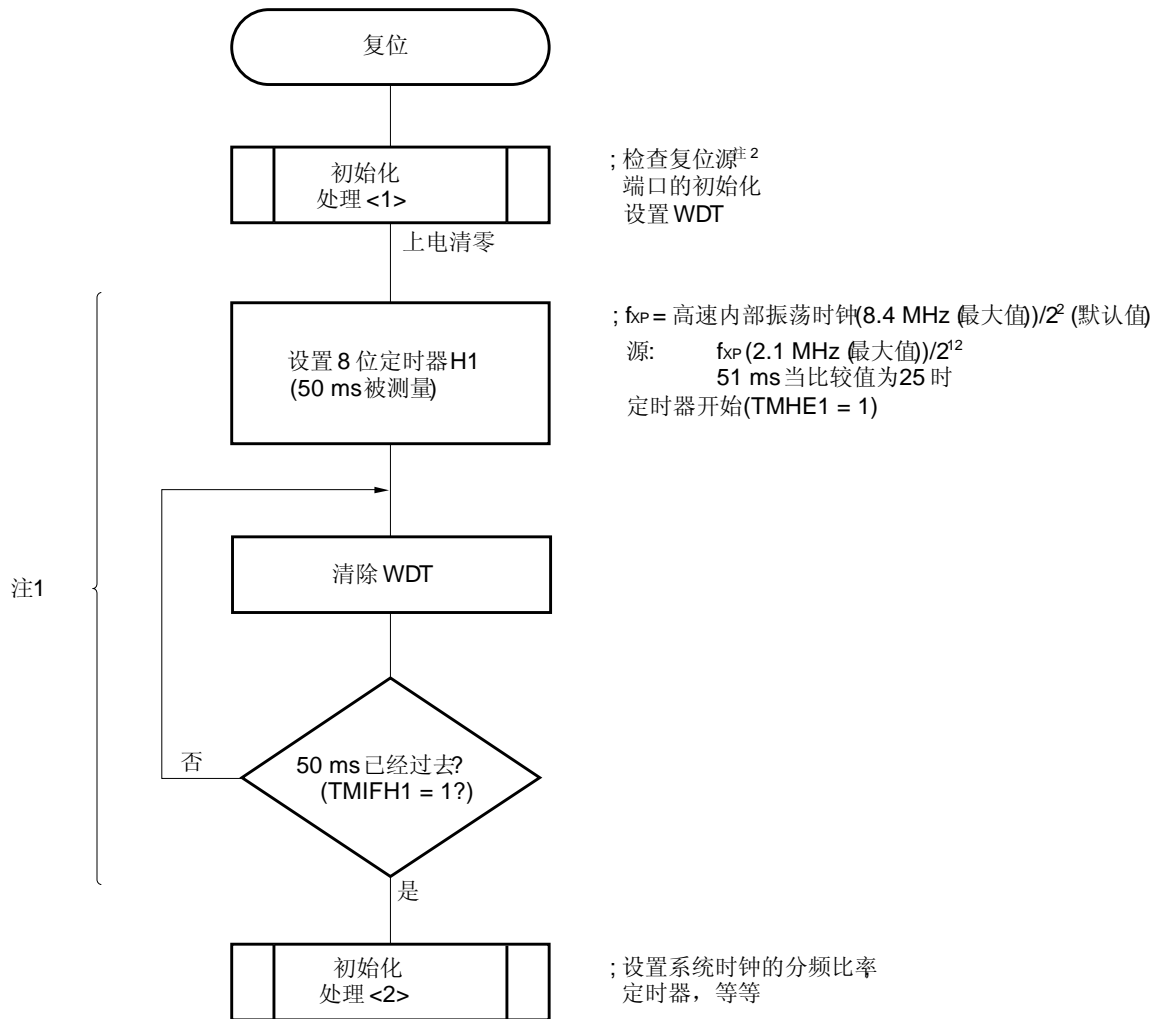
在一个电源电压 (V_{DD}) 以一定的周期在 POC 检测电压 (V_{POC}) 附近波动的系统中, 系统可能会被重复复位并从复位状态被释放。在这种情况下, 从复位释放到微控制器的操作开始的时间可以通过以下措施被任意设置。

<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 12-3. 复位释放后的软件处理举例 (1 / 2)

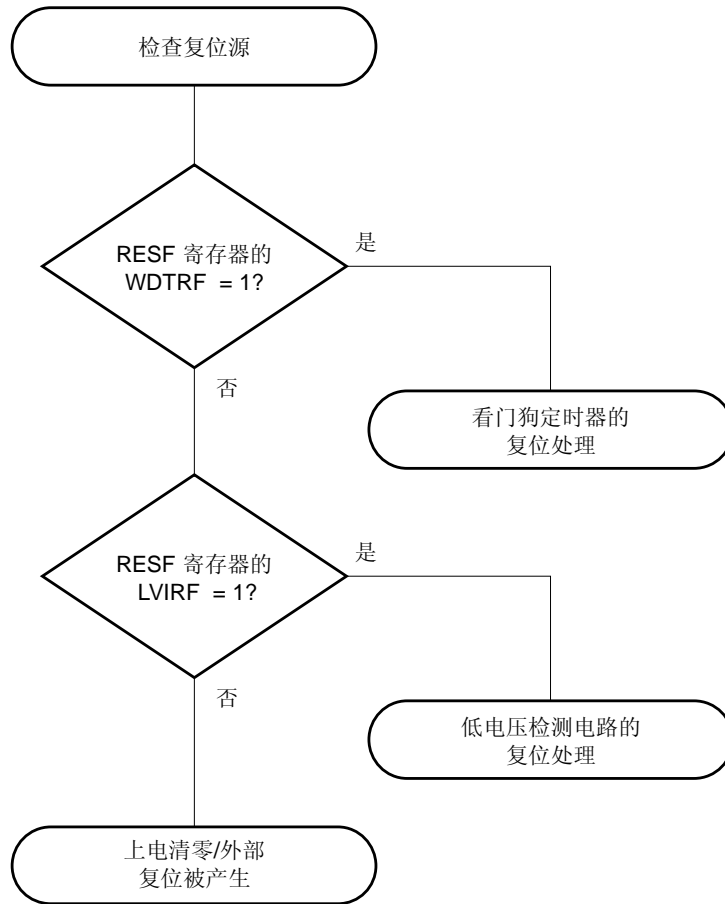
- 如果电源电压在 POC 检测电压附近的波动周期为 50ms 或更少



- 注
1. 如果在这个周期中复位被再次产生, 初始化处理<2>不被启动。
 2. 流程图在下页被表示。

图 12-3. 复位释放后的软件处理举例 (2 / 2)

• 检查复位源



第十三章 低电压检测电路

13.1 低电压检测电路的功能

低电压检测电路（LVI）有以下功能。

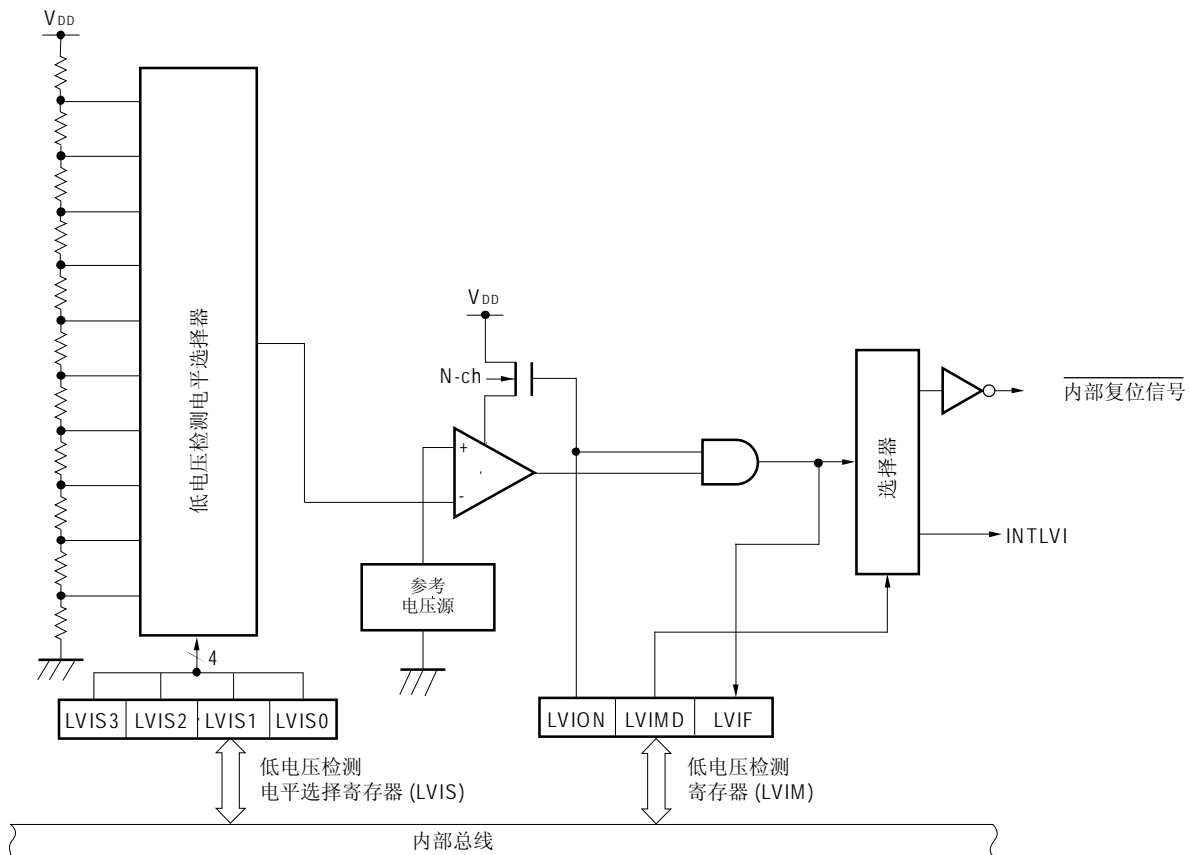
- 比较电源电压（ V_{DD} ）和检测电压（ V_{LVI} ），并且当 $V_{DD} < V_{LVI}$ 时产生一个内部中断信号或内部复位信号。
- 电源电压的检测电平（十个电平）可以通过软件来更改。
- 中断或复位功能可通过软件选择。
- 在 STOP 模式下可以操作。

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 第十一章 复位功能。

13.2 低电压检测电路的配置

低电压检测电路的框图如图 13-1 所示。

图 13-1. 低电压检测电路的框图



13.3 控制低电压检测电路的寄存器

低电压检测电路由以下寄存器控制。

- 低电压检测寄存器 (LVIM)
- 低电压检测电平选择寄存器 (LVIS)

(1) 低电压检测寄存器 (LVIM)

该寄存器设置低电压检测和工作模式。

该寄存器可以通过一个 1 位或 8 位存储器操作指令来设置。

复位信号产生清除该寄存器为 00H^{注1}。

图 13-2. 低电压检测寄存器 (LVIM) 的格式

地址: FF50H 复位后: 00H^{注1} R/W^{注2}

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

LVION ^{注3}	允许低电压检测操作
0	禁止操作
1	允许操作

LVIMD	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. 对于 LVI 产生的复位, LVIM 的值不会被初始化。
 2. 第 0 位是只读位。
 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
1. 要停止 LVI, 遵循以下过程之一。
 - 当使用 8 位操作指令时: 写入 00H 到 LVIM。
 - 当使用 1 位存储器操作指令时: 清除 LVION 为 0。
 2. 确保设置第 2 到 6 位为 0。

(2) 低电压检测电平选择寄存器 (LVIS)

该寄存器选择低电压检测电平。

该寄存器通过一个 8 位存储器操作指令来设置。

复位信号产生清除该寄存器为 00H^注。

图 13-3. 低电压检测电平选择寄存器 (LVIS) 的格式

地址: FF51H, 复位后: 00H^注 R/W

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

LVIS3	LVIS2	LVIS1	LVIS0	检测电平
0	0	0	0	V _{LV10} (4.3 V ±0.2 V)
0	0	0	1	V _{LV11} (4.1 V ±0.2 V)
0	0	1	0	V _{LV12} (3.9 V ±0.2 V)
0	0	1	1	V _{LV13} (3.7 V ±0.2 V)
0	1	0	0	V _{LV14} (3.5 V ±0.2 V)
0	1	0	1	V _{LV15} (3.3 V ±0.15 V)
0	1	1	0	V _{LV16} (3.1 V ±0.15 V)
0	1	1	1	V _{LV17} (2.85 V ±0.15 V)
1	0	0	0	V _{LV18} (2.6 V ±0.1 V)
1	0	0	1	V _{LV19} (2.35 V ±0.1 V)
除上面以外				禁止设置

注 对于 LVI 产生的复位, LVIS 的值不会被初始化。

注意事项 1. 第 4 到 7 位必须被设置为 0。

2. 如果在 LVI 操作期间上面以外的值被写入, 在写入的时刻值变为不确定, 因此在写入之前确保停止 LVI (LVIM 寄存器的第 7 位 (LVION) = 0)。

13.4 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 (V_{LVI})，当 $V_{DD} < V_{LVI}$ 时产生内部复位信号，当 $V_{DD} \geq V_{LVI}$ 时释放内部复位。
- 用作中断
比较电源电压 (V_{DD}) 和检测电压，并且当 $V_{DD} < V_{LVI}$ 时产生一个内部中断信号 ($INTLVI$)。

操作按照下面来设置。

(1) 当用作复位时

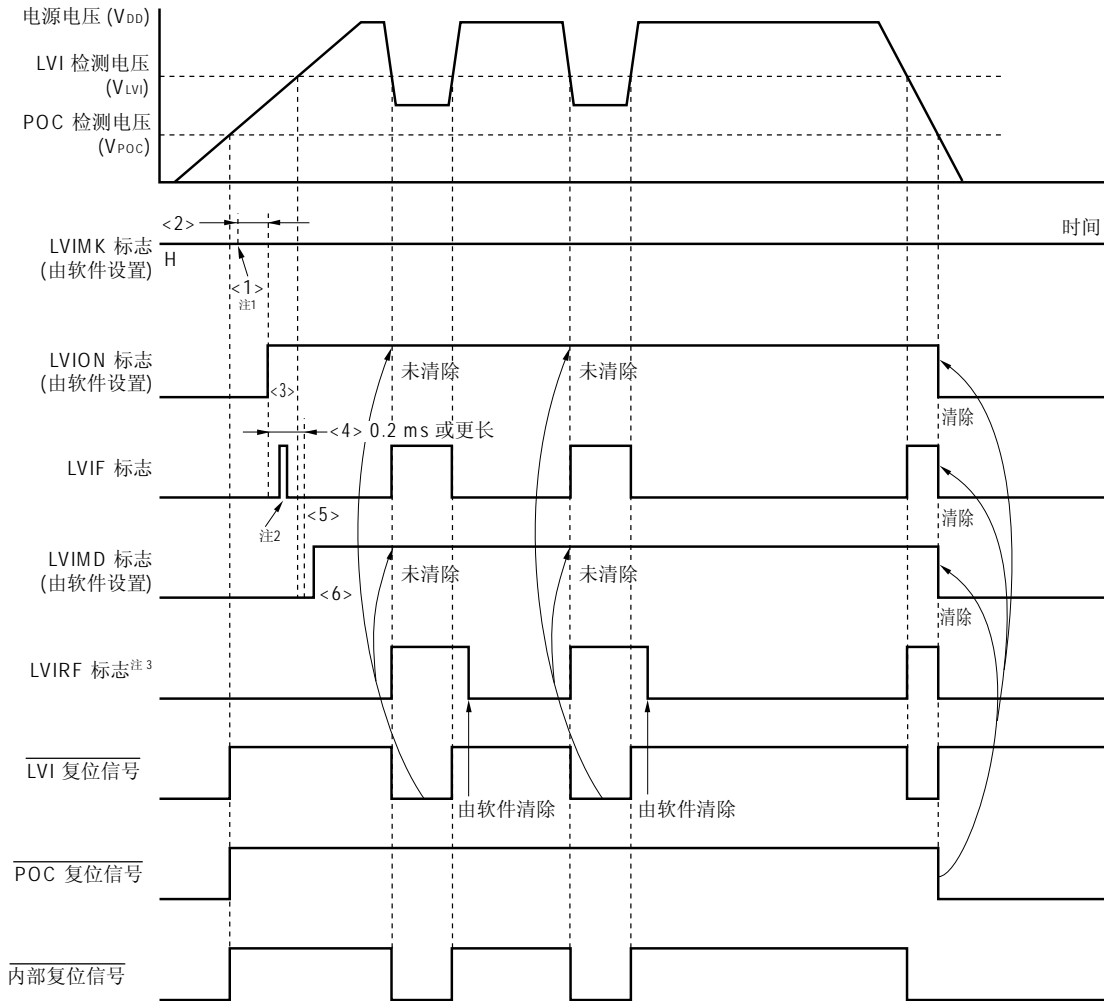
- 当启动操作时
 - <1> 屏蔽 LVI 中断 ($LVIMK = 1$)。
 - <2> 使用低电压检测电平选择寄存器 ($LVIS$) 的第 3 到 0 位 ($LVIS3$ 到 $LVIS0$) 来设置检测电压。
 - <3> 设置 $LVIM$ 的第 7 位 ($LVION$) 为 1 (允许 LVI 操作)
 - <4> 使用软件产生一个至少 0.2 ms 的等待。
 - <5> 等待直到通过 $LVIM$ 的第 0 位 ($LVIF$) 确认“电源电压 (V_{DD}) \geq 检测电压 (V_{LVI})”。
 - <6> 设置 $LVIM$ 的第 1 位 ($LVIMD$) 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 13-4 表示低电压检测电路的内部复位信号产生的时序。该图中的号码<1>到<6>对应上面的<1>到<6>。

- 注意事项**
1. <1>必须总被执行。当 $LVIMK = 0$ 时，一个中断可能会在<3>中的处理后立即发生。
 2. 当 $LVIMD$ 被设置为 1 时，如果电源电压 (V_{DD}) \geq 检测电压 (V_{LVI})，一个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：按顺序清除 $LVIMD$ 为 0 和 $LVION$ 为 0。

图 13-4. 低电压检测电路内部复位信号产生时序



- 注
1. LVIMK 标志被复位信号设置为“1”。
 2. LVIF 标志可设置为 1。
 3. LVIRF 是复位控制标志寄存器 (RESF) 的第 0 位。关于 RESF 的详细情况, 参见 第十一章 复位功能。

备注 上面图 13-4 中的<1>到<6>对应于 13.4 (1) 当用作复位时的“当启动操作时”的描述中的<1>到<6>。

(2) 当用作中断时

• 当启动操作时

- <1> 屏蔽 LVI 中断 (LVIMK = 1)。
- <2> 使用低电压检测电平选择寄存器 (LVIS) 的第 3 到 0 位 (LVIS3 到 LVIS0) 来设置检测电压。
- <3> 设置 LVIM 的第 7 位 (LVION) 为 1 (允许 LVI 操作)
- <4> 使用软件产生一个至少 0.2 ms 的等待。
- <5> 等待直到通过 LVIM 的第 0 位 (LVIF) 确认 “电源电压 (V_{DD}) ≥ 检测电压 (V_{LVI}) ”。
- <6> 清除 LVI 的中断请求标志 (LVIIF) 为 0。
- <7> 释放 LVI 的中断屏蔽标志 (LVIMK)。
- <8> 执行 EI 指令 (当向量中断被使用时)。

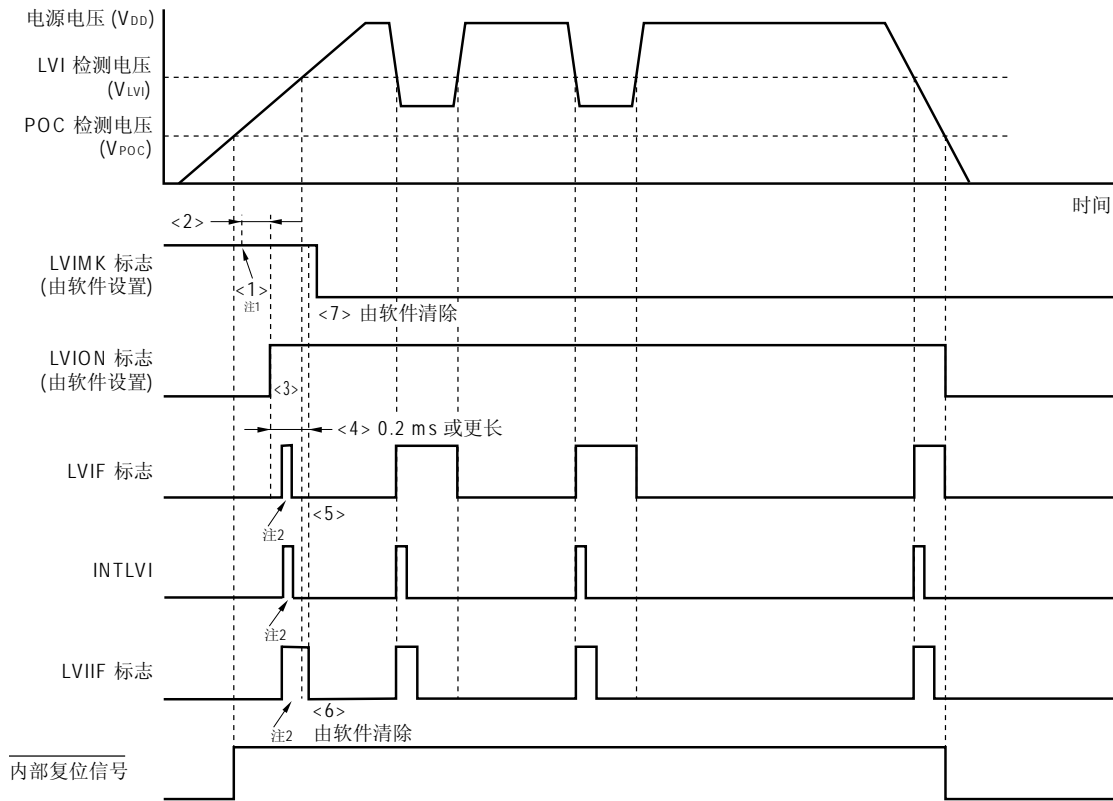
图 13-5 表示低电压检测电路的中断信号产生的时序。该图中的号码<1> 到 <7> 对应上面的<1>到<7>。

• 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。

图 13-5. 低电压检测电路中断信号产生时序



- 注
1. LVIMK 标志被复位信号设置为“1”。
 2. 一个中断请求信号 (INTLVI) 可能被产生, 并且 LVIF 和 LVIIF 标志可能被设置为 1。

备注 上面图 13-5 中的<1>到<7>对应于 13.4 (2) 当用作中断时的“当启动操作时”的描述中的<1>到<7>。

13.5 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) 的 (b)。

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 (见图 13-6)。

(2) 当用作中断时

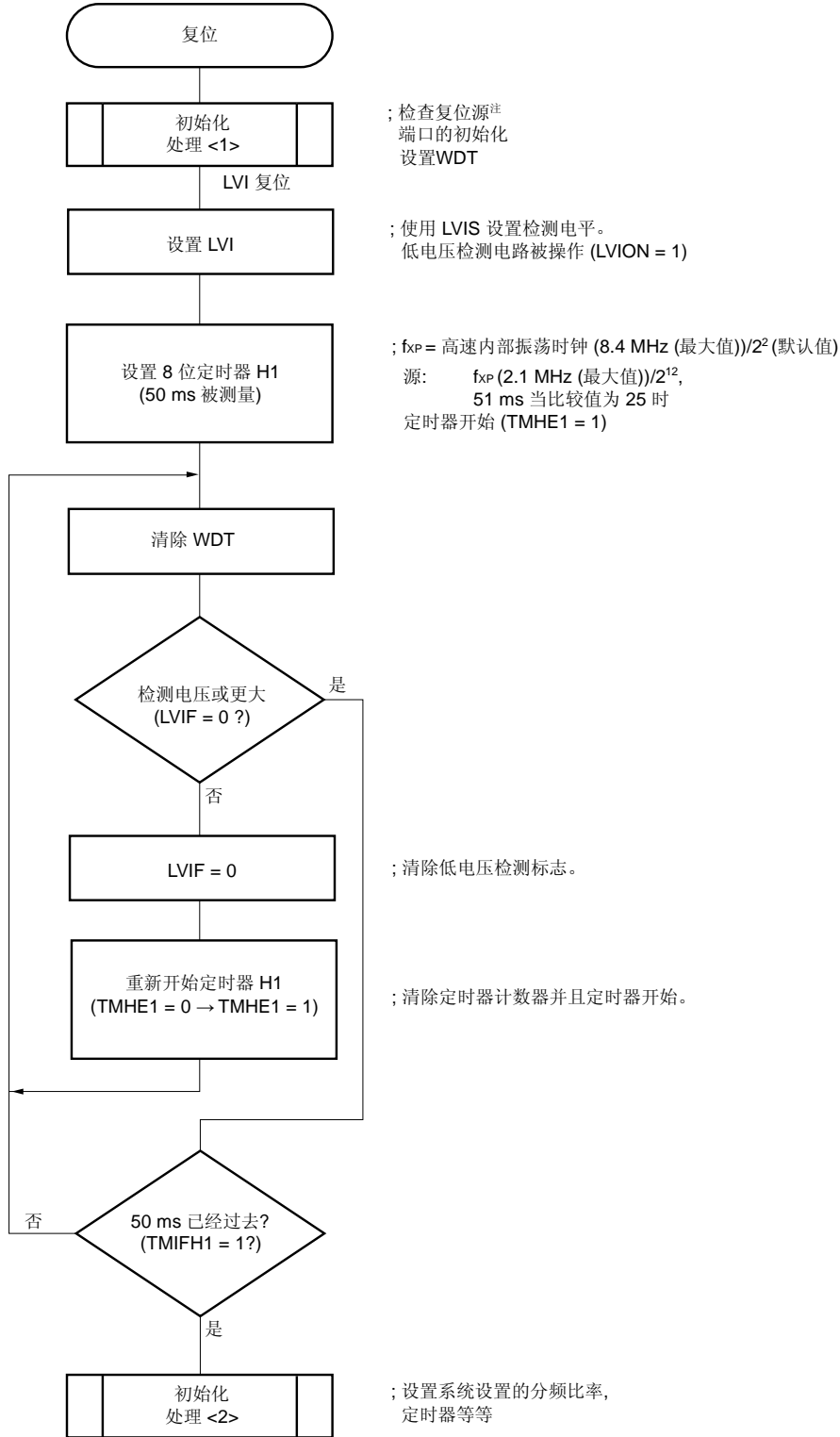
(a) 为低电压检测执行处理^注。在 LVI 中断复位程序中使用低电压检测寄存器 (LVIM) 的第 0 位 (LVIF) 来检查“电源电压 (V_{DD}) \geq 检测电压 (V_{LVI})”。清除中断请求标志寄存器 0 (IF0) 的第 1 位 (LVIIF) 为 0。

(b) 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}) \geq 检测电压 (V_{LVI})”。

注 对于低电压检测处理, CPU 时钟速度被切换为低速等等。

图 13-6. 复位释放后的软件处理举例 (1 / 2)

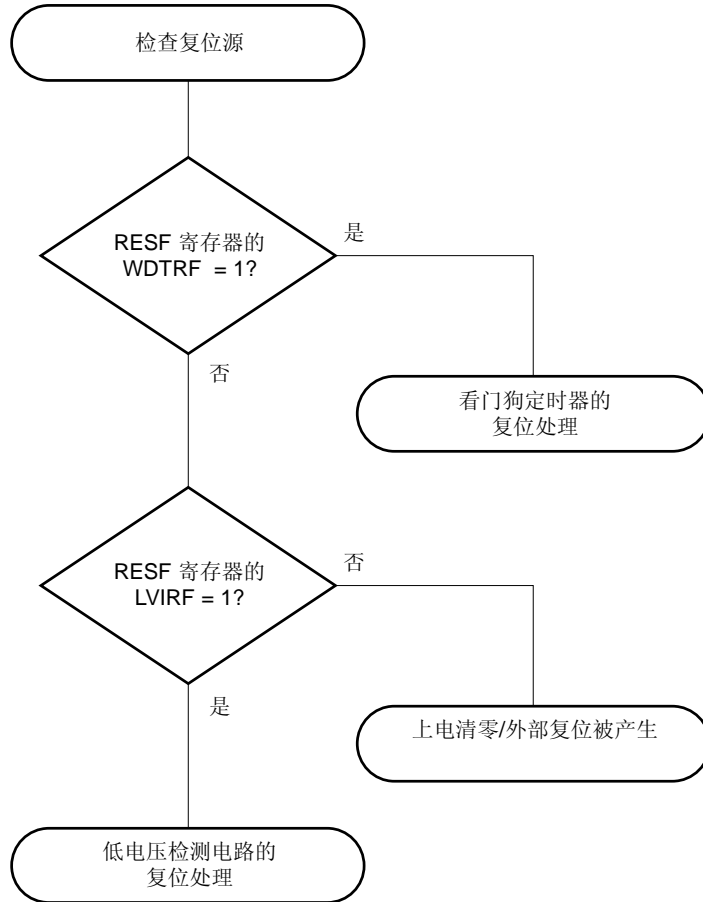
- 如果电源电压在 LVI 检测电压附近的波动周期为 50ms 或更少



注 流程图在下页显示。

图 13-6. 复位释放后的软件处理举例（2 / 2）

• 检查复位源



第十四章 选项字节

14.1 选项字节的功能

μ PD78F9510、78F9511、78F9512 的 flash 存储器的地址 0080H 是选项字节区域。当电源被提供或当复位后开始时，选项字节被自动引用，并且指定功能的设置被执行。当使用该产品时，确保通过使用选项字节设置以下功能。

(1) 系统时钟源的选择

- 高速内部振荡时钟
- 晶体 / 陶瓷振荡时钟
- 外部时钟输入

(2) 低速内部振荡时钟的振荡

- 不能被停止。
- 可由软件停止。

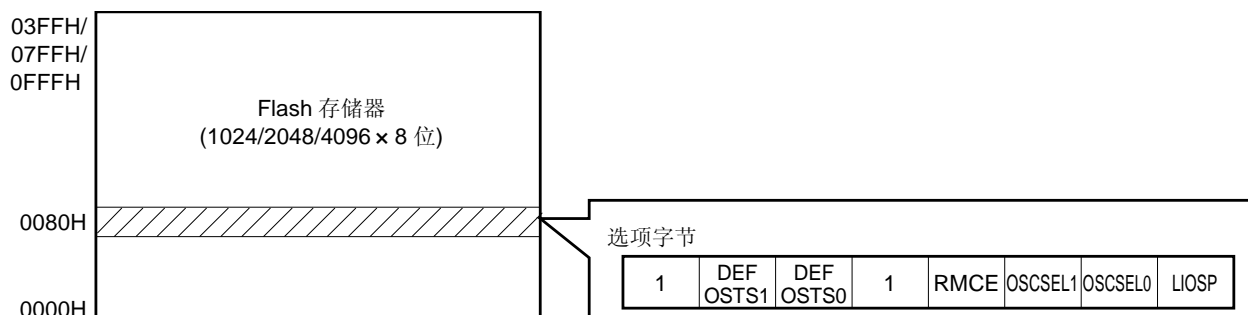
(3) $\overline{\text{RESET}}$ 引脚的控制

- 用作 $\overline{\text{RESET}}$ 引脚
- $\overline{\text{RESET}}$ 引脚用作输入端口引脚 (P34) (参阅 14.3 当 $\overline{\text{RESET}}$ 引脚用作输入端口引脚 (P34) 时的注意事项)。

(4) 上电或复位释放后的振荡稳定时间

- $2^{10}/f_x$
- $2^{12}/f_x$
- $2^{15}/f_x$
- $2^{17}/f_x$

图 14-1. 选项字节的位置



14.2 选项字节的格式

选项字节的格式如下所示。

图 14-2. 选项字节的格式 (1 / 2)

地址: 0080H

	7	6	5	4	3	2	1	0
	1	DEFOSTS1	DEFOSTS0	1	RMCE	OSCSEL1	OSCSEL0	LIACP

DEFOSTS1	DEFOSTS0	上电或复位释放后的振荡稳定时间
0	0	$2^{10}/f_x$ (102.4 s)
0	1	$2^{12}/f_x$ (409.6 s)
1	0	$2^{15}/f_x$ (3.27 ms)
1	1	$2^{17}/f_x$ (13.1 ms)

注意事项 只有当晶体 / 陶瓷振荡时钟被选作系统时钟源时, 该选项的设置才有效。如果高速内部振荡时钟或外部时钟输入被选作系统时钟源, 无等待时间。

RMCE	RESET 引脚的控制
1	RESET 引脚用作其本身。
0	RESET 引脚用作输入端口引脚 (P34)。

注意事项 因为选项字节在复位释放后被引用, 如果在选项字节被引用之前一个低电平被输入到 RESET 引脚, 复位状态将不会被释放。

同时, 当设置 RMCE 为 0 时, 请连接上拉电阻。

OSCSEL1	OSCSEL0	系统时钟源的选择
0	0	晶体 / 陶瓷振荡时钟
0	1	外部时钟输入
1	x	高速内部振荡时钟

注意事项 因为 X1 和 X2 引脚也用作 P23 和 P22 引脚, X1 和 X2 引脚被使用的情况根据选择的系统时钟源而改变。

(1) 晶体 / 陶瓷振荡时钟被选择

X1 和 X2 引脚不能被用作 I/O 端口引脚, 因为它们被用作时钟输入引脚。

(2) 外部时钟输入被选择

因为 X1 引脚被用作外部时钟输入引脚, P23 不能被用作 I/O 端口引脚。

(3) 高速内部振荡时钟被选择

P23 和 P22 引脚可以被用作 I/O 端口引脚。

备注 x: 不关注

图 14-2. 选项字节的格式 (2 / 2)

LIOCP	低速内部振荡
1	不能被停止 (即使 1 被写入到 LSRSTOP 位, 振荡也不会停止)
0	可由软件停止 (当 1 被写入到 LSRSTOP 位时, 振荡停止)

- 注意事项**
1. 如果选择低速内部振荡器不能被停止, 看门狗定时器 (WDT) 的计数时钟固定为低速内部振荡时钟。
 2. 如果选择低速内部振荡器可由软件停止, WDT 的计数时钟的提供在 HALT / STOP 模式下被停止, 无论低速内部振荡模式寄存器 (LSRCM) 的第 0 位 (LSRSTOP) 的设置如何。类似的, 当低速内部振荡时钟以外的时钟被选作 WDT 的计数时钟时, 时钟提供也被停止。
在低速内部振荡器操作 (LSRSTOP = 0) 期间, 计数在 STOP 模式下, 时钟也可以被提供给 8 位定时器 H1。

- 备注**
1. () : $f_x = 10 \text{ MHz}$
 2. 关于谐振器的振荡稳定时间, 参阅要使用的谐振器的特性。
 3. 设置选项字节的软件代码示例如下所示。

```
OPB CSEG AT 0080H
DB 10010001B
```

 - ; 设置到选项字节
 - ; 低速内部振荡器不能被停止。
 - ; 系统时钟为晶体或陶瓷谐振器。
 - ; $\overline{\text{RESET}}$ 引脚被用作输入端口引脚 (P34)。
 - ; 最小振荡稳定时间 ($2^{10}/f_x$)
 4. 关于选项字节被引用的时序, 参见第十一章 复位功能。

14.3 当 $\overline{\text{RESET}}$ 引脚用作输入端口引脚 (P34) 时的注意事项

对于一个已经写入的设备, 其已经通过选项字节设置为“ $\overline{\text{RESET}}$ 已经被用作输入端口引脚 (P34)”, 当重新擦除 / 写入 (使用专用 flash 存储器编程器通过板上编程) 该设备时, 了解以下事项。

在向目标系统提供电源之前, 连接专用 flash 存储器编程器并打开其电源。

如果电源事先被提供给目标系统, 将不能切换为 flash 存储器编程模式。

第十五章 FLASH存储器

15.1 特征

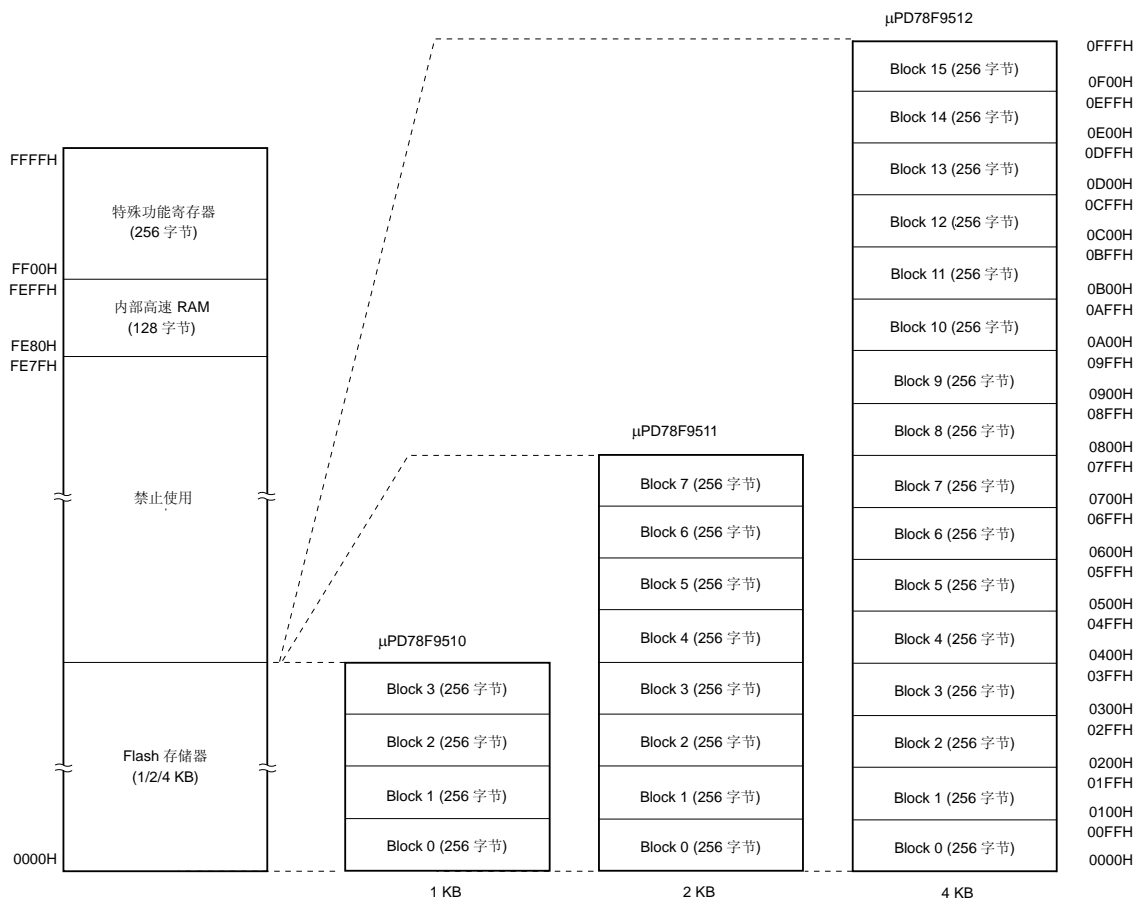
μ PD78F9510、78F9511、78F9512 的内部 flash 存储器具有以下特征。

- 甚至在没有独立的专用的电源供电的情况下擦 / 写
- 容量：1 / 2 / 4 KB
 - 擦除单元：1 block（256 字节）
 - 写单元：1 block（在板上 / 板外编程时），1 字节（在自编程时）
- 重写方法
 - 通过专用的 flash 存储器编程器重写（板上 / 板外编程）
 - 通过用户程序（自编程）重写 flash 存储器
- 支持在板上 / 板外编程时通过安全功能重写 flash 存储器
- 支持在自编程时通过保护字节以 block 为单元的安全功能。

15.2 存储器配置

1 / 2 / 3 KB 内部 flash 存储器部分被分成 8 / 16block，能以 block 为单位编程 / 擦除。通过使用专用的 flash 存储器编程器，所有的 blocks 也能被一次擦除，。

图 15-1. Flash 存储器映射



15.3 功能概要

μPD78F9510、78F9511、78F9512 的内部 flash 存储器能用专用的 flash 存储器编程器的重写函数重写，无论 μPD78F9510、78F9511、78F9512 是否已经安装于目标系统上（板上 / 板外编程）。

假定目标系统生产 / 出货后程序被改变，此时提供用户程序重写程序功能（自编程）。

参见表 15-1 了解 flash 存储器写控制函数。

另外，支持禁止重新写入内部 flash 存储器的用户程序的安全功能，所以程序不能被未经授权的人更改。

参见 15.7.3 安全设置 了解安全功能详细情况。

表 15-1. 重写方法

重写方法	功能概要	操作模式
(1) 板上编程	在设备安装于目标系统后 Flash 存储器能被重写，通过使用专用的 flash 存储器编程器。	Flash 存储器编程模式
(2) 板外编程	在设备安装于目标系统之前 Flash 存储器能被重写，通过使用专用的 flash 存储器编程器和专用的编程适配板（FA 系列）。	
(2) 自编程	Flash 存储器能被执行用户程序重写，该程序事先经板上 / 板外编程方法写入 Flash 存储器。	自编程模式

备注 1: FA 系列是 Naito Densei Machida Mfg. Co., Ltd 的一个产品。

2. 参见下面的部分了解 flash 存储器写控制功能详细情况。

- 15.7 板上和板外 Flash 存储器编程
- 15.8 通过自写的 Flash 存储器编程

15.4 用Flash存储器编程器写入

专用 flash 存储器编程器的下面两字节能用于写数据到 μ PD78F9510、78F9511、78F9512 的内部 flash 存储器。

- FlashPro4 (PG-FP4, FL-PR4)
- PG-FPL2

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 PD78F9510、78F9511、78F9512 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和测试点必须安装到目标系统中。仅当用晶体 / 陶瓷谐振器安装时写入数据要求测试点（参见图 15-6 了解测试点安装）。

(2) 板外编程

在 μ PD78F8024 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 FL-PR4 和 FA 系列是 Naito Densei Machida Mfg. Co., Ltd 的产品。

15.5 编程环境

写入程序到 flash 存储器所需要的环境如下所示。

图 15-2. 写入程序到 Flash 存储器的环境 (FlashPro4)

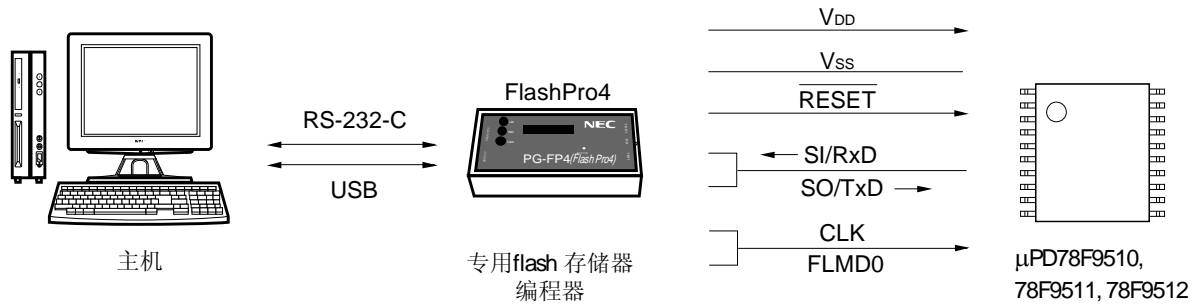
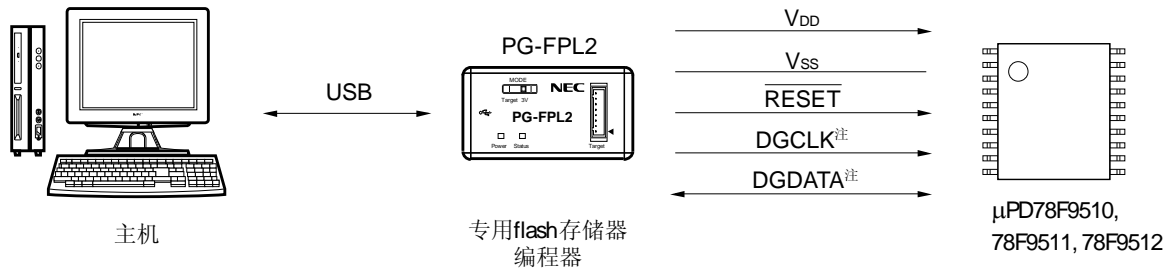


图 15-3. 写入程序到 Flash 存储器的环境 (PG-FPL2)



注 DGCLK 是一个通信时钟，而 DGDATA 是一个通信数据的发送 / 接收信号。

控制专用 flash 存储器编程器的主机是必需的。使用 PG-FP4 或 FL-PR4 时，在从主机下载程序后仅用专用的 flash 存储器编程器即可重写数据。

UART 在连接专用 flash 存储器编程器和 μPD78F9510、78F9511、78F9512 时用来处理写入和擦除操作。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。

用于开发工具的最新编程器固件、GUI 和参数文件从下载网站下载 (<http://www.necel.com/micro/ods/eng/index.html>)。

表 15-2. μ PD78F9510、78F9511、78F9512 和 FlashPro4 间的连线

FlashPro4 连接引脚			μ PD78F9510、78F9511、78F9512 连接引脚	
引脚名称	输入 / 输出	引脚功能	引脚名称	引脚号
CLK [‡]	输出	到 μ PD78F9510、78F9511、78F9512 的时钟	X1/P23	8
FLMD0 [‡]	输出	板上模式信号		
SI / RxD [‡]	输入	接收信号	X2/P22	9
SO / TxD [‡]	输出	接收信号 / 板上模式信号		
/RESET	输出	复位信号	RESET/P34	12
V _{DD}	-	V _{DD} 电压产生 / 电压监视	V _{DD}	5
GND	-	地	V _{DD}	4

注 μ PD78F9510、78F9511、78F9512、CLK 和 FLMD0 信号连接到 X1 引脚，SI/RxD 和 SO/TxD 信号连接到 X2 信号；因此这些信号需要直接连接。

图 15-4. FlashPro4 的连线图

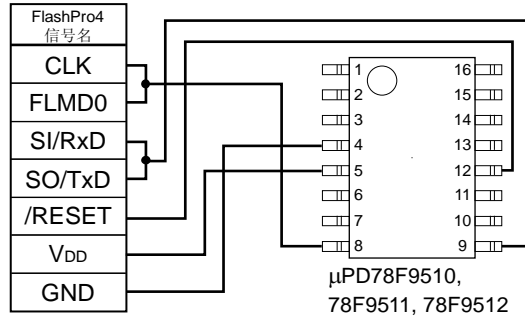
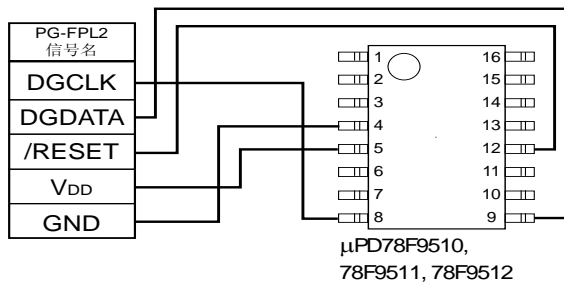


表 15-3. μ PD78F9510、78F9511、78F9512 和 PG-FPL2 间的连线

PG-FPL2 连接引脚			μ PD78F9510、78F9511、78F9512 连接引脚	
引脚名称	输入 / 输出	引脚功能	引脚名称	引脚号
DGCLK	输出	到 μ PD78F9510、78F9511、78F9512 的时钟	X1/P23	8
DGDATA	输入 / 输出	发送 / 接收信号，板上模式信号	X2/P22	9
/RESET	输出	复位信号	RESET/P34	12
V _{DD}	-	V _{DD} 电压产生	V _{DD}	5
GND	-	地	V _{SS}	4

图 15-5. PG-FPL2 的连线图



15.6 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。首先提供选择正常操作模式或者板上 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。

自编程模式下引脚状态和 HALT 模式下的相同。

15.6.1 X1 和 X2 引脚

X1 和 X2 引脚用于 flash 存储器编程的串行接口。因此，如果 X1 和 X2 引脚连接到一个外部设备，产生信号冲突。为了防止信号冲突，用外部设备隔离连接。

同样的，当一个电容连接到 X1 和 X2 引脚，通信时的波形改变，因此通信可能被禁止，取决于电容容量。确保 flash 编程期间隔离与电容的连接。

当用谐振器安装执行板上写入时隔离谐振器困难，而晶体或陶瓷谐振器选作系统时钟时，执行以下处理（1）和（2）。

- （1）在设备和谐振器间安装尽可能小的测试点，并通过测试点连接编程器。保持连线尽可能短（参见图 15-6 和表 15-4）。
- （2）设置通信时钟的晶振频率，用于专用 flash 存储器编程器的 GUI 软件写操作。研究所用谐振器的串 / 并谐振和反谐振，并设置震荡频率超出谐振频率 $\pm 10\%$ （参见图 15-7 和表 15-5）。

图 15-6. 安装测试点示例

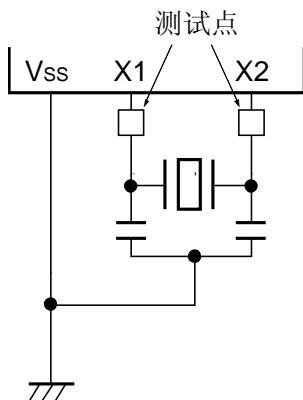


表 15-4.用到的时钟和测试点安装

用到的时钟		测试点安装
高速内部振荡时钟		不要求
外部时钟		
晶体 / 陶瓷振荡时钟	谐振器安装前	要求
	谐振器安装后	

图 15-7. PG-FP4 GUI 软件设置示例

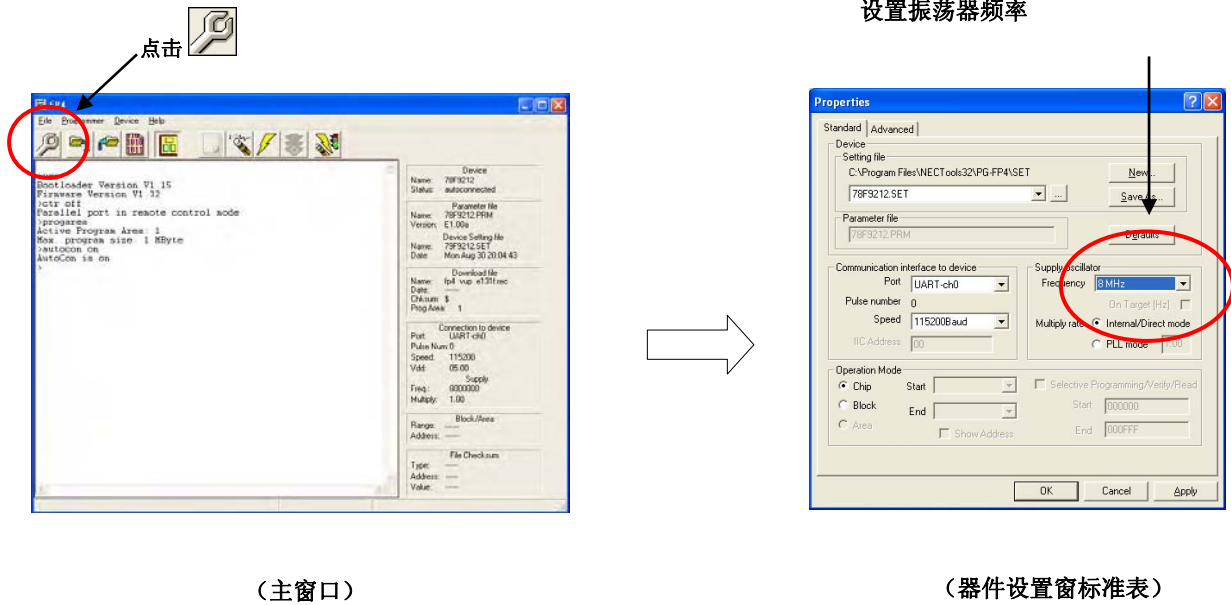


表 15-5. 振荡频率和 PG-FP4 GUI 软件设置值示例

振荡频率	PG-FP4 GUI 软件设置值示例 (通信频率)
$1 \text{ MHz} \leq f_x < 4 \text{ MHz}$	8 MHz
$4 \text{ MHz} \leq f_x < 8 \text{ MHz}$	9 MHz
$8 \text{ MHz} \leq f_x < 9 \text{ MHz}$	10 MHz
$9 \text{ MHz} \leq f_x \leq 10 \text{ MHz}$	8 MHz

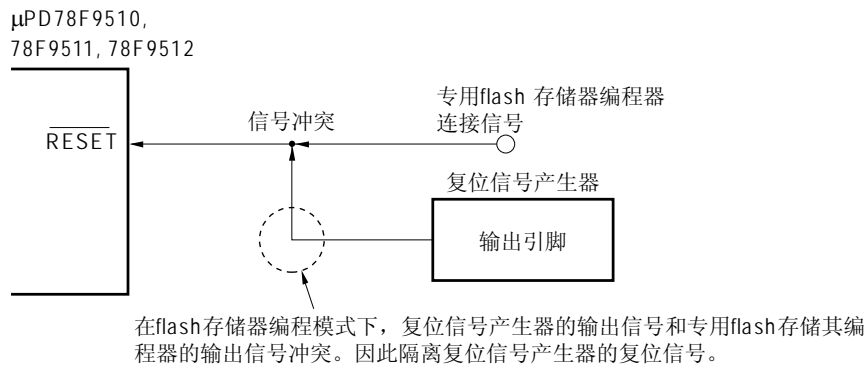
注意事项 以上是推荐值。依赖于使用环境，这些值可能改变，因此充分评估后设置。

15.6.2 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 **RESET** 引脚，信号冲突将发生。要防止这个冲突，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 15-8. 信号冲突 (RESET 引脚)



15.6.3 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}。

自编程模式下引脚状态和 HALT 模式下的相同。

15.6.4 电源供电

连接 V_{DD} 引脚到 flash 存储器编程器的 V_{DD}，和 V_{SS} 引脚到 flash 存储器编程器的 V_{SS}。

15.7 板上和板外Flash存储器编程

15.7.1 Flash存储器编程模式

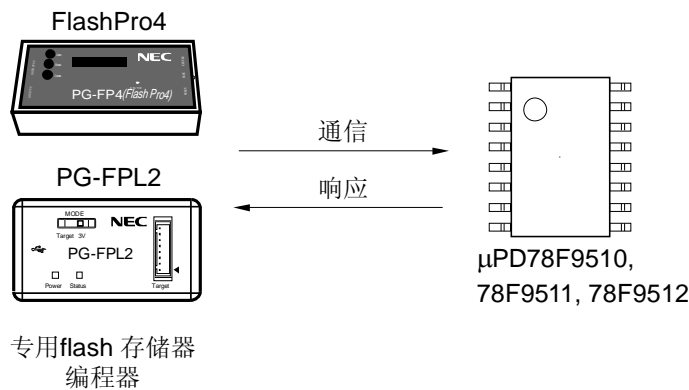
要使用专用 flash 存储器编程器来重新写入 flash 存储器的内容，设置 μ PD78F9510、78F9511、78F9512 为 flash 存储器编程模式。当 μ PD78F9510、78F9511、78F9512 连接到 flash 存储器编程器和一个通信命令发送到微控制器时，微控制器设置为 flash 存储器编程器模式。

当在板上写入 flash 存储器时，通过使用跳线来更改模式。

15.7.2 通信命令

专用 flash 存储器编程器通过命令控制 μ PD78F9510、78F9511、78F9512。从 flash 存储器编程器发送到 μ PD78F9510、78F9511、78F9512 的信号被叫做通信命令，并且从 μ PD78F9510、78F9511、78F9512 发送到专用 flash 存储器编程器的命令被叫做响应。

图 15-9. 通信命令



通信命令列于下表中。所有的通信命令从 flash 存储器编程器发出，并且 μ PD78F9510、78F9511、78F9512 执行对应各自通信命令的处理。

表 15-6. 通信命令

分类	通信命令名称	功能
擦除	批擦除（芯片擦除）命令	擦除整个存储器内容
	Block 擦除命令	擦除指定 block 存储器内容。
写入	写命令	写入指定地址范围和执行内容校验检查。
校验和	校验和命令	读指定地址范围的校验和并和写入数据比较
空白检查	空白检查命令	确认整个存储器的擦除状态。
安全	安全设置命令	禁止批擦除（片擦除）命令，block 擦除命令和写命令来阻止第三方操作。

μ PD78F9510、78F9511、78F9512 对专用 flash 存储器编程器发出的通信命令返回一个响应。 μ PD78F9510、78F9511、78F9512 发出的响应名在下面被列出。

表 15-7. 响应名

命令名	功能
ACK	响应命令 / 数据。
NAK	响应非法命令 / 数据。

15.7.3 安全设置

下面表示的操作可以使用安全设置命令来禁止。

- 批擦除（芯片擦除）被禁止

对 flash 存储器整个 block 的 block 擦除和批擦除（芯片擦除）命令执行是禁止的。一旦批擦除（芯片擦除）命令执行被禁止，所有的禁止设置再也不能被取消。

注意事项 在对批擦除的安全设置被设置后，擦除不能被执行。此外，即使写入命令被执行，与已经写入 flash 存储器的数据不同的数据不能被写入，因为擦除命令无效。

- Block 擦除被禁止

Flash 存储器的 block 擦除命令被禁止执行。该禁止设置能用批擦除（芯片擦除）命令取消。

- 写禁止

Flash 存储器的整个 block 的写和 block 擦除命令被禁止执行。该禁止设置能用批擦除（芯片擦除）命令取消。

备注 当编程模式被设置时，安全设置有效。

当 flash 存储器被发货时，批擦除（芯片擦除）命令、block 擦除命令、写入命令被默认设置为允许。上面的安全设置仅在片上 / 片外编程时有可能。每种安全设置可以被组合使用。

表 15-8 表示当 μ PD78F9510、78F9511、78F9512 安全功能被允许时擦除和写入命令之间的关系。

表 15-8. 安全功能允许时命令间关系

安全 \ 命令	批擦除（芯片擦除）命令	Block 擦除命令	写命令
当批擦除（芯片擦除）安全操作允许时	禁止	禁止	允许 [*]
当 block 擦除安全操作允许时	允许		允许
当写安全操作允许时			禁止

注 因为擦除命令被禁止，不同于已经写入 flash 存储器的数据不能被写入。

表 15-9 显示了每种编程模式下的安全设置和操作间的关系。

表 15-9 每种编程模式下安全设置和操作间的关系

安全设置 \ 编程模式	板上 / 板外编程		自编程	
	安全设置	安全操作	安全设置	安全操作
批擦除 (芯片擦除)	可能	有效 ^{注1}	不可能	无效 ^{注2}
块擦除				
写入				

- 注
1. 每个命令执行被安全设置禁止。
 2. 自编程命令可能执行，不管安全设置。

15.8 通过自写的Flash存储器编程

μPD78F9510、78F9511、78F9512 支持自编程功能，通过用户程序重写 flash 存储器，让升级编程成为可能。

注意事项 自编程处理必须包含在执行自写前的程序中。

- 备注**
1. 使用自编程，参见从 15.8.4 往前的示例。
 2. 使用 μPD78F9510、78F9511、78F9512 的内部 flash 存储器作为外部 EEPROM 存数据，参见 **78K0S/Kx1+ EEPROM 效仿应用笔记 (U17379E)**。

15.8.1 自编程概要

为执行自编程，切换用户程序（正常模式）的正常操作模式到自编程模式。对 flash 存储器的写入 / 擦除处理（事先已设置到寄存器中）通过在自编程模式期间执行 HALT 指令执行。当处理完成，HALT 状态被自动释放。

为切换到自编程模式，执行一个指定寄存器的指定顺序。详细情况参见 15.8.4 切换正常模式到自编程模式举例。

备注 自编程写数据能以 MOV 指令引用。

表 15-10. 自编程模式

模式	用户程序执行	用 HALT 指令执行 Flash 存储器写入 / 擦除
正常模式	允许	-
自编程模式	允许 ^注	允许

注 在自编程模式期间可屏蔽中断服务被禁止。

图 15-10 显示自编程框图，图 15-11 显示自编程状态转移图，表 15-11 列出控制自编程命令。

图 15-10. 自编程框图

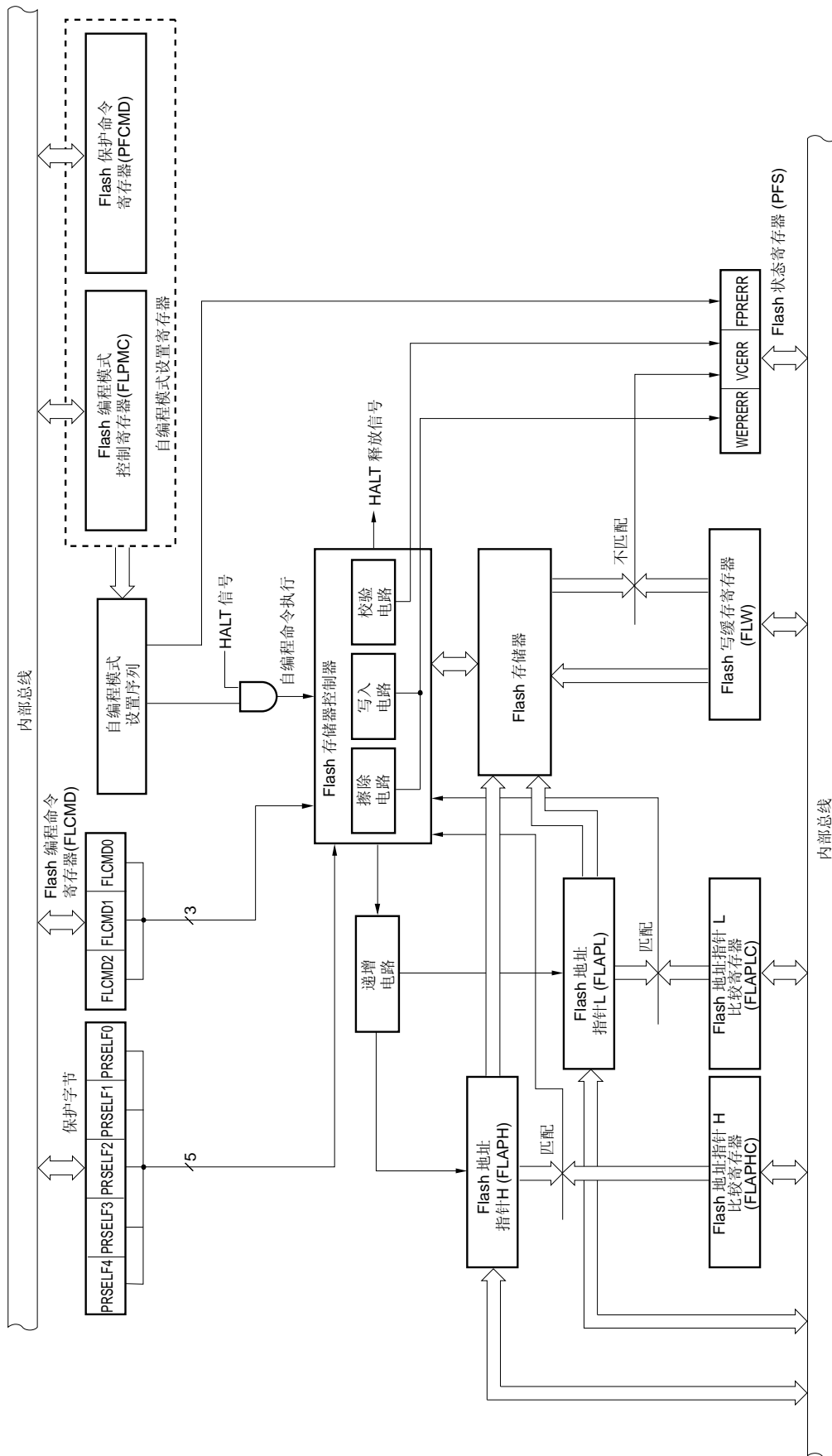


图 15-11. 自编程状态转移图

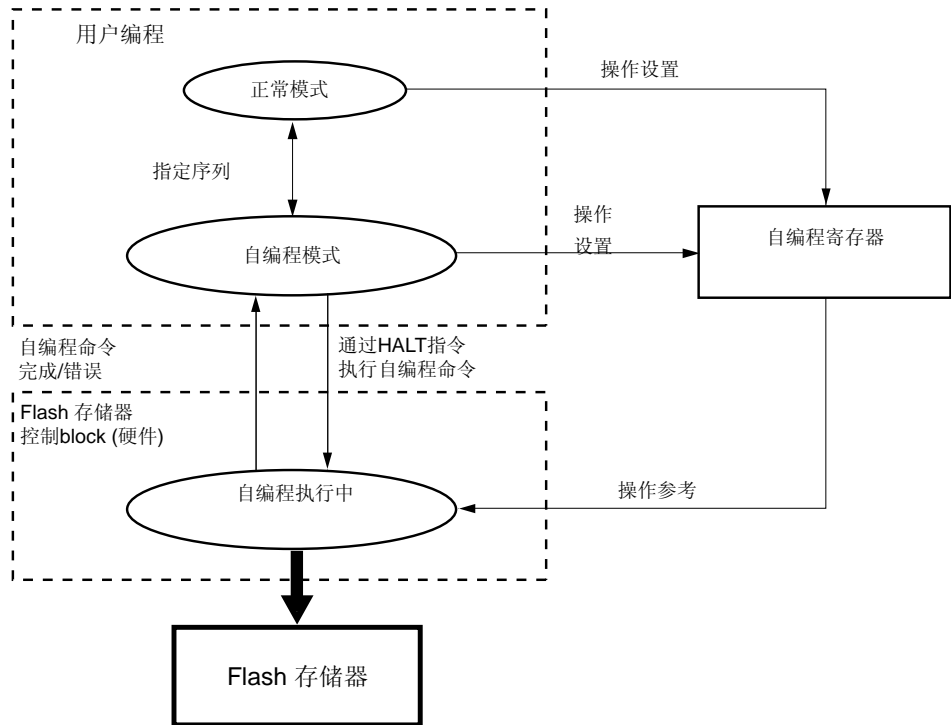


表 15-11. 自编程控制命令

命令名	功能	从 HALT 指令执行到命令执行结束时间
内部校验 1	该命令用于检查数据是否已被正确写入 flash 存储器。它用来检查数据是否已经写入一整 block。	内部校验 1block (内部校验命令执行一次) : 6.8 ms
内部校验 2	该命令用于检查数据是否已被正确写入 flash 存储器。它用来检查数据是否已经写入相同 block 的多个地址。	内部 1 字节校验: 27 μ s
块擦除	该命令用来擦除一个指定 block。在执行前指定 block 数。	8.5 ms
Block 空白检查	该命令用来检查指定 block 的数据是否已经被擦除。指定 block 数目, 然后执行该命令。	480 μ s
字节写	该命令用来写 1 字节数据到 flash 存储器中指定地址。指定写入地址和写入数据, 然后执行该命令。	150 μ s

备注 指定相同 block 内一个地址来执行内部校验 1 命令, 但是如果写入相同 block 内的多地址推荐内部校验 2。

15.8.2 自编程功能注意事项

- 当自编程命令执行时没有指令能被执行。因此，事先清除和重启看门狗定时器以使自编程期间看门狗定时器不溢出。参见表 15-11 了解自编程执行时间。
- 自编程期间发生的中断在自编程模式结束后可被确认。为避免该操作，当用指定顺序切换正常模式到自编程模式时禁止中断服务（设置 MK0 为 FFH，并执行 DI 指令）。
- 当自编程命令执行时不使用 RAM。
- 如果 flash 存储器被写或擦除时供电电压降低或输入复位信号，不能保证写入 / 擦除。
- 在 block 擦除期间空白数据值设为 FFH。
- 预先设置 CPU 时钟在自编程时为 1MHz 或更高。
- 执行指定顺序来设置自编程模式后马上执行自编程 NOP 和 HALT 命令，然后执行自编程。此时，HALT 指令在 10 μ s（最大值）+ 2 CPU 时钟（fCPU）后自动释放。
- 如果振荡器时钟或外部时钟选作系统时钟，在执行指定顺序设置自编程模式后马上执行 NOP 和 HALT 指令，在释放 HALT 状态后等待 8 μ s，然后执行自编程。
- 用 1 位存储器操作指令检查 FPRERR。
- 自编程模式下引脚状态和 HALT 模式下的相同。
- 因为在自编程模式下通过板上 / 板外编程设置安全功能被禁止，自编程命令可被执行不管安全功能的设置。为禁止自编程期间写入或擦除处理，设置保护字节。
- 确保在执行自编程命令前清除 flash 地址指针 H（FLAPH）的第 4 到 7 位和 flash 地址指针 H 比较寄存器（FLAPHC）为 0。如果设置这些位为 1 来执行自编程，设备可能故障。
- 在设置到自编程模式和正常模式前马上清除 FLCMD 寄存器值为 00H。

15.8.3 用于自编程功能的寄存器

以下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）
- Flash 编程命令寄存器（FLCMD）
- Flash 地址指针 H 和 L（FLAPH 和 FLAPL）
- Flash 地址指针 H 比较寄存器和 flash 地址指针 L 比较寄存器（FLAPHC 和 FLAPLC）。
- Flash 写缓存寄存器（FLW）

μ PD78F9510、78F9511、78F9512 在 flash 存储器的 0081H 字节处有一区域称作保护字节。

(1) Flash 编程模式控制寄存器 (FLPMC)

自编程模式下当数据写入 flash 存储器时该寄存器用来设置操作模式，并读保护字节的设置值。

数据仅用指定顺序写入 FLPMC (参见 15.8.3 (2) Flash 保护命令寄存器 (PFCMD))，因此应用系统不被噪声或程序挂起引起的故障停止。

该寄存器由 8 位存储器操作指令设置。

复位信号产生使该寄存器内容不确定。

图 15-12. Flash 编程模式控制寄存器 (FLPMC) 格式

地址: FFA2H 复位后: 不确定^{#1} R/W^{#2}

符号	7	6	5	4	3	2	1	0
FLPMC	0	PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	0	FLSPM

FLSPM	自编程模式时操作模式选择
0	正常模式 这是正常操作状态。执行HALT指令设置待机状态。
1	自编程模式 自编程命令能通过执行指定顺序来改变正常模式时的模式来执行。 设置一个命令，地址和写入的数据，然后执行HALT指令来执行自编程。

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	保护字节的设置值读到这些位中。
---------	---------	---------	---------	---------	-----------------

- 注
1. 复位释放时第 0 位 (FLSPM) 清零。复位释放后保护字节的设置值被读取到第 2 到 6 位 (PRSELF0 到 PRSELF4)。
 2. 第 2 到 6 位 (PRSELF0 到 PRSELF4) 只读。

- 注意事项
1. 设置自编程模式情况下的注意事项，参见 15.8.2 自编程功能注意事项。
 2. 预先设置 CPU 时钟在自编程时为 1MHz 或更高。
 3. 执行指定顺序来设置自编程模式后马上执行自编程 NOP 和 HALT 命令，然后执行自编程。此时，HALT 指令在 10 μs (最大值) + 2 CPU 时钟 (fCPU) 后自动释放。
 4. 如果振荡器时钟或外部时钟选作系统时钟，在执行指定顺序设置自编程模式后马上执行 NOP 和 HALT 指令，在释放 HALT 状态后等待 8μs，然后执行自编程。
 5. 在设置到自编程模式和正常模式前马上清除 FLCMD 寄存器值为 00H。

(2) Flash 保护命令寄存器 (PFCMD)

如果应用系统由于噪声或程序挂起导致的故障而停止，写 flash 编程模式控制寄存器 (FLPMC) 的操作将对系统有严重影响。PFCMD 用来保护 FLPMC 不被写，因此应用系统不会停止。

仅当以下指定顺序的写操作执行时写 FLPMC 被允许。

- <1>写指定值到 PFCMD (A5H)
- <2>写设置值到 FLPMC 的第 0 位 (FLSPM) (在这步写无效)
- <3>写该值的相反值到 FLPMC 的第 0 位 (FLSPM) (在这步写无效)
- <4>写设置值到 FLPMC 的第 0 位 (FLSPM) (在这步写有效)

注意事项 自编程模式下中断服务不能执行。禁止中断服务 (通过在 MK0=FFH 时执行 DI 指令) 在执行指定顺序设置自编程模式前和执行指定顺序改变模式到正常模式后。

这重写寄存器值，因此寄存器不能被非法写。

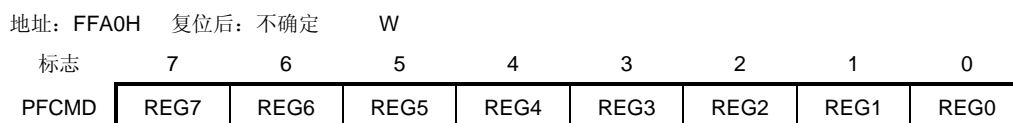
非法写操作发生能被 flash 状态寄存器 (PFS) 的第 0 位 (FPRERR) 检测到。用 1 位存储器操作指令检查 FPRERR。

每次 FLPMC 值改变时 A5H 必须写到 PFCMD。

PFCMD 由 8 位存储器操作指令设置。

复位信号产生导致 PFCMD 不确定。

图 15-13. Flash 保护命令寄存器 (PFCMD) 格式



(3) Flash 状态寄存器 (PFS)

如果数据未按正确顺序写到被保护的 flash 编程模式控制寄存器 (FLPMC) (写 flash 保护命令寄存器 (PFCMD))，FLPMC 不写并且产生一个保护错误。如果它发生，PFS 的第 0 位 (FPRERR) 置为 1。

当 FPRERR 为 1 时，可通过写 0 来清除它。

自编程时发生的错误能反应在 PFS 的第 1 位 (VCERR) 和第 2 位 (WEPRERR)。VCERR 或 WEPRERR 能通过写 0 清除。

为检查操作是否正确执行 PFS 寄存器的所有标志必须预清除到 0。

PFS 由 1 位或 8 位存储器操作指令设置。

复位信号产生清除 PFS 到 00H。

注意事项 用 1 位存储器操作指令检查 FPRERR。

图 15-14. Flash 状态寄存器 (PFS) 格式

地址: FFA1H 复位后: 00H R/W

标志	7	6	5	4	3	2	1	0
PFS	0	0	0	0	0	WEPRERR	VCERR	FPRERR

1. FPRERR 标志操作环境

<置位条件>

- 如果当操作于外设寄存器的存储指令执行时写 PFCMD，不会写指定值 (A5H) 到 FLPMC。
- 如果第一条存储指令操作于<1>对外设寄存器而不是 FLPMC 后
- 如果第一条存储指令操作于<2>对外设寄存器而不是 FLPMC 后
- 如果<2>后一个不是设置到 FLPMC 的值的相反值的数被第一条存储指令写
- 如果第一条存储指令操作于<3>对外设寄存器而不是 FLPMC 后
- 如果<3>后一个不是设置到 FLPMC (<2>中写的值) 的值被第一条存储指令写

备注 上面三角内的数字对应 (2) Flash 保护命令寄存器 (PFCMD)。

<复位条件>

- 如果 0 写入 FPRERR 标志
- 如果复位信号产生

2. VCERR 标志操作环境

<置位条件>

- 擦除校验错误
- 内部写校验错误

如果 VCERR 设置，意味着标志寄存器没有被正确擦除或写入。再次以指定过程擦除或写入存储器。

备注 如果发生擦除或写入保护错误，VCERR 标志也被设置。

<复位条件>

- 如果 0 写入 VCERR 标志
- 当复位信号产生

3. WEPRERR 标志操作环境

<置位条件>

- 如果被保护字保护不被擦除或写入的指定区域被 flash 地址指针 H (FLAPH) 指定，一个命令执行到该区域。
- 如果 1 写到没有被擦除的一位 (该位为 0)。

<复位条件>

- 如果 0 写入 WEPRERR 标志
- 当复位信号产生

(4) Flash 编程命令寄存器 (FLCMD)

该寄存器在自编程模式下用来指定 flash 存储器是否被擦除, 写入, 或校验。

该寄存器由 1 位或 8 位存储器操作指令设置。

复位信号产生清除该寄存器到 00H。

图 15-15. Flash 编程命令寄存器 (FLCMD) 格式

地址: FFA3H 复位后: 00H R/W

标志	7	6	5	4	3	2	1	0
FLCMD	0	0	0	0	0	FLCMD2	FLCMD1	FLCMD0

FLCMD2	FLCMD1	FLCMD0	命令名	功能
0	0	1	内部校验1	该命令用于检查数据是否已被正确写入flash存储器。它用来检查数据是否已经写入一整block。如果错误发生, flash状态寄存器 (PFS) 的第1位 (VCERR) 或第2位 (WEPRERR) 设置为1。
0	1	0	内部校验2	该命令用于检查数据是否已被正确写入flash存储器。它用来检查数据是否已经写入相同block的多个地址。如果错误发生, flash状态寄存器 (PFS) 的第1位 (VCERR) 或第2位 (WEPRERR) 设置为1。
0	1	1	块擦除	该命令用来擦除指定block。它用于板上模式和自编程模式。
1	0	0	Block 空白检查	该命令用来检查指定block的数据是否已经被擦除。
1	0	1	字节写	该命令用来写1字节数据到flash存储器中指定地址。指定写入地址和写入数据, 然后执行该命令。 如果1写入未被清除的一位 (该位数据为0), 则flash状态寄存器 (PFS) 的第2位 (WEPRERR) 变为1。
除上面以外 ^注			禁止设置	

注 任何上述以外的命令执行, 执行将被立刻终止, 并且 flash 状态寄存器 (PFS) 的第 1 位或 2 (WEPRERR 或 VCERR) 被置为 1。

(5) Flash 地址指针 H 和 L (FLAPH 和 FLAPL)

当自编程模式下擦除，写入或校验存储器时这些寄存器用于指定 flash 存储器的开始地址。

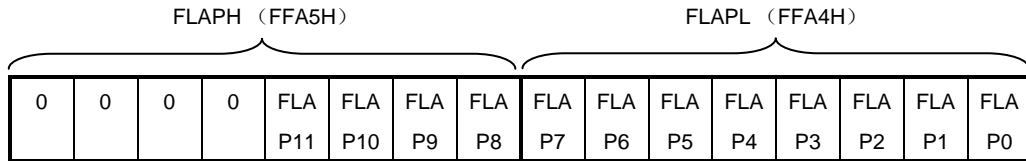
FLAPH 和 FLAPL 由计数器组成，并且当不执行编程命令时他们递增到匹配 FLAPHC 和 FLAPLC 的值。因此当执行编程命令时，再次设置值。

这些寄存器由 1 位或 8 位存储器操作指令设置。

复位信号产生导致这些寄存器不确定。

图 15-16. Flash 地址指针 H / L (FLAPH / FLAPL) 格式

地址: FFA4H, FFA5H 复位后: 00H R/W



注意事项 确保在执行自编程命令前清除 flash 地址指针 H (FLAPH) 的第 4 到 7 位和 flash 地址指针 H 比较寄存器 (FLAPHC) 为 0。当执行自编程命令时这些位的值为 1。

(6) Flash 地址指针 H 比较寄存器和 flash 地址指针 L 比较寄存器 (FLAPHC 和 FLAPLC)

这些寄存器在自编程模式下 flash 存储器校验时用来指定内部顺序操作的地址范围。

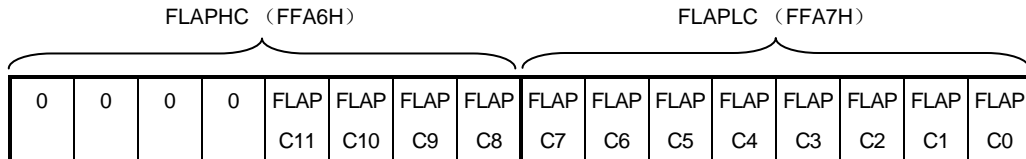
设置 FLAPHC 为 FLAPH 相同的值。设置校验执行到 FLAPLC 的范围的最后地址。

这些寄存器由 1 位或 8 位存储器操作指令设置。

复位信号产生清除这些寄存器到 00H。

图 15-17. Flash 地址指针 H / L 比较寄存器 (FLAPHC / FLAPLC) 格式

地址: FFA6H, FFA7H 复位后: 00H R/W



注意事项 1. 在执行自编程命令前确保清除 FLAPH 和 FLAPHC 的第 4 到 7 位为 0。当执行自编程命令时这些位的值为 1。

2. 设置被 block 擦除，校验，或空白检查的 block 数（和 FLAPH 值相同）到 FLAPHC。
3. 当执行 block 擦除时清除 FLAPLC 到 00H，并且当执行空白检查时设置该寄存器为 FFH。

(7) Flash 写缓存寄存器 (FLW)

该寄存器用来存储写入 flash 存储器的数据。

该寄存器由 8 位存储器操作指令设置。

复位信号产生清除这些寄存器到 00H。

图 15-18. Flash 写缓存寄存器 (FLW) 格式

地址: FFA8H 复位后: 00H R/W

标志	7	6	5	4	3	2	1	0
FLW	FLW7	FLW6	FLW5	FLW4	FLW3	FLW2	FLW1	FLW0

(8) 保护字节

保护字节用来指定禁止写入或擦除的区域。指定区域仅当自编程模式下有效。因为保护区域的自编程无效，确保写到保护区的数据。

图 15-19. 保护字节格式 (1 / 2)

地址: 0081H

7	6	5	4	3	2	1	0
1	PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	1	1

• μ PD78F9510

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	Status
0	1	1	1	0	Blocks 3到0被保护。
0	1	1	1	1	Blocks 1和0被保护。 Blocks 2 和 3可被写入或擦除。
1	1	1	1	1	所有blocks能被写入或擦除。
除上面以外					禁止设置

• μ PD78F9511

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	Status
0	1	1	0	0	Blocks 7到0被保护。
0	1	1	0	1	Blocks 5到0被保护。 Blocks 6 和 7可被写入或擦除。
0	1	1	1	0	Blocks 3到0被保护。 Blocks 4 到 7可被写入或擦除。
0	1	1	1	1	Blocks 1和0被保护。 Blocks 2 到 7可被写入或擦除。
1	1	1	1	1	所有blocks能被写入或擦除。
除上面以外					禁止设置

图 15-19. 保护字节格式 (2 / 2)

• μ PD78F9512

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	Status
0	1	0	0	0	Blocks 15到0被保护。
0	1	0	0	1	Blocks 13到0被保护。 Blocks 14 和 15可被写入或擦除。
0	1	0	1	0	Blocks 11到0被保护。 Blocks 12 到 15可被写入或擦除。
0	1	0	1	1	Blocks 9到0被保护。 Blocks 10 到 15可被写入或擦除。
0	1	1	0	0	Blocks 7到0被保护。 Blocks 8 到 15可被写入或擦除。
0	1	1	0	1	Blocks 5到0被保护。 Blocks 6 到 15可被写入或擦除。
0	1	1	1	0	Blocks 3到0被保护。 Blocks 4 到 15可被写入或擦除。
0	1	1	1	1	Blocks 1和0被保护。 Blocks 2 到 15可被写入或擦除。
1	1	1	1	1	所有blocks能被写入或擦除。
除上面以外					禁止设置

15.8.4 切换正常模式到自编程模式举例

在执行自编程前操作模式必须从正常模式切换到自编程模式。

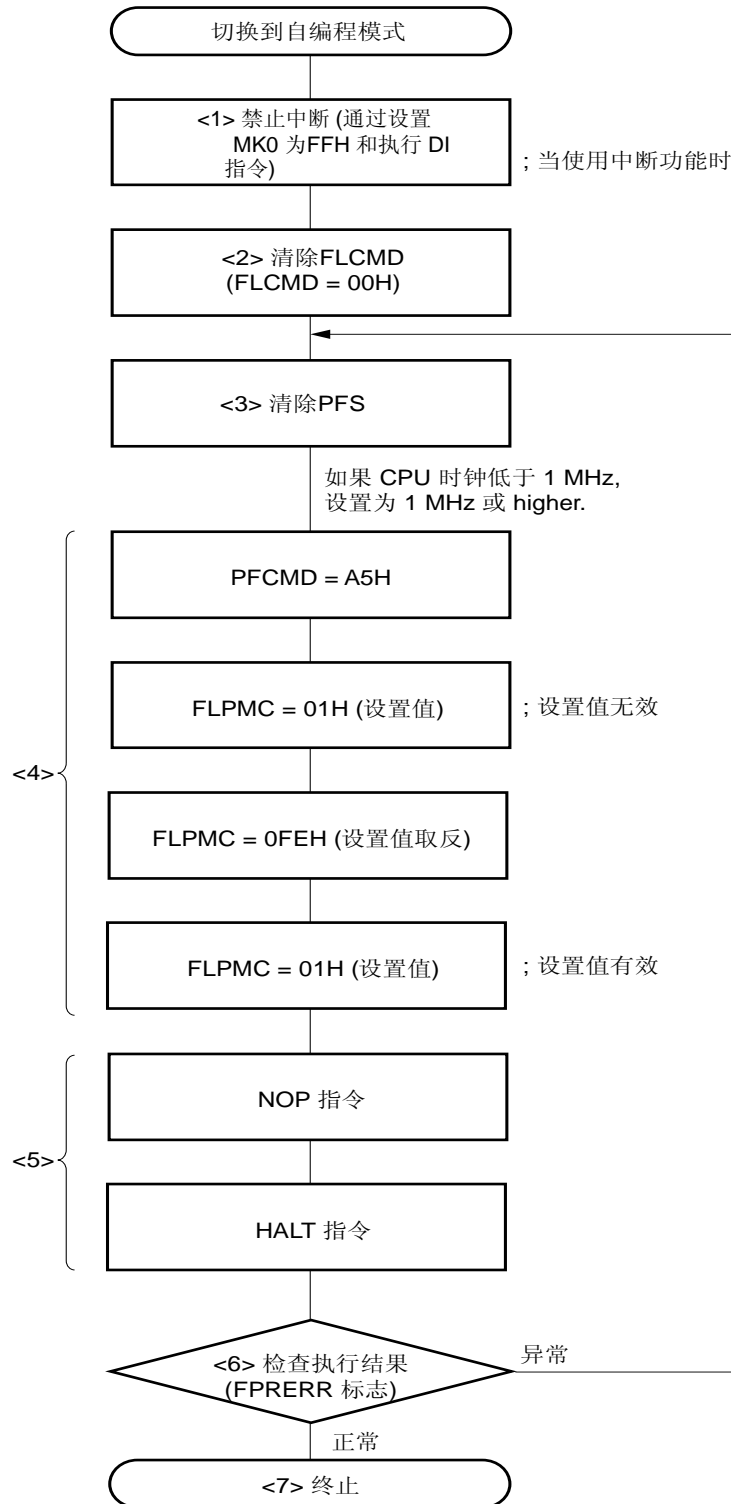
切换到自编程模式的举例说明如下。

- <1> 如果中断功能被使用则禁止中断（通过设置中断屏蔽标志寄存器（MK0）到 FFH 和执行 DI 指令）。
- <2> 清除 FLCMD（FLCMD = 00H）。
- <3> 清除 flash 状态寄存器（PFS）。
- <4> 用指定顺序设置自编程模式。
 - 写指定值（A5H）到 PFCMD。
 - 写 01H 到 FLPMC（该步写无效）。
 - 写 0FEH（01H 的相反值）到 FLPMC（该步写无效）。
 - 写 01H 到 FLPMC（该步写有效）。
- <5> 执行 NOP 指令和 HALT 指令。
- <6> 用 PFS 的第 0 位（FPRERR）检查指定顺序的执行结果。
异常→ <3>，正常→ <7>
- <7> 模式切换完成。

注 如果 CPU 时钟低于 1MHz，设置它到 1MHz 或更高。

注意事项 确保用在数据没有擦除或写入的地址处的用户程序执行上述的一系列操作。

图 15-20. 切换自编程模式示例



注意事项 确保用在数据没有擦除或写入的地址处的用户程序执行上述的一系列操作。

备注 图 15-20 的<1> 到 <7> 对应 15.8.4 的<1> 到 <7> (前页)。

切换模式到自编程模式的程序示例如下。

```

;-----
;START
;-----
        MOV     MK0,#11111111B    ;屏蔽所有中断
        MOV     FLCMD,#00H        ;清除 FLCMD 寄存器

        DI

ModeOnLoop:                                ;配置设置使 CPU 时钟 ≥ 1 MHz
        MOV     PFS,#00H          ;清除 flash 状态寄存器
        MOV     PFCMD,#0A5H       ;PFCMD 寄存器控制
        MOV     FLPMC,#01H        ;FLPMC 寄存器控制（设置值）
        MOV     FLPMC,#0FEH       ;FLPMC 寄存器控制（相反设置值）
        MOV     FLPMC,#01H        ;用 FLPMC 寄存器设置自编程模式
                                       ;控制（设置值）

        NOP
        HALT
        BT      PFS.0,$ModeOLoop  ;检查写指定寄存器完成
                                       ;当发生错误时重复相同的处理

;-----
;结束
;-----

```

15.8.5 切换自编程模式到正常模式举例

在执行自编程后操作模式必须从自编程模式切换到正常模式。

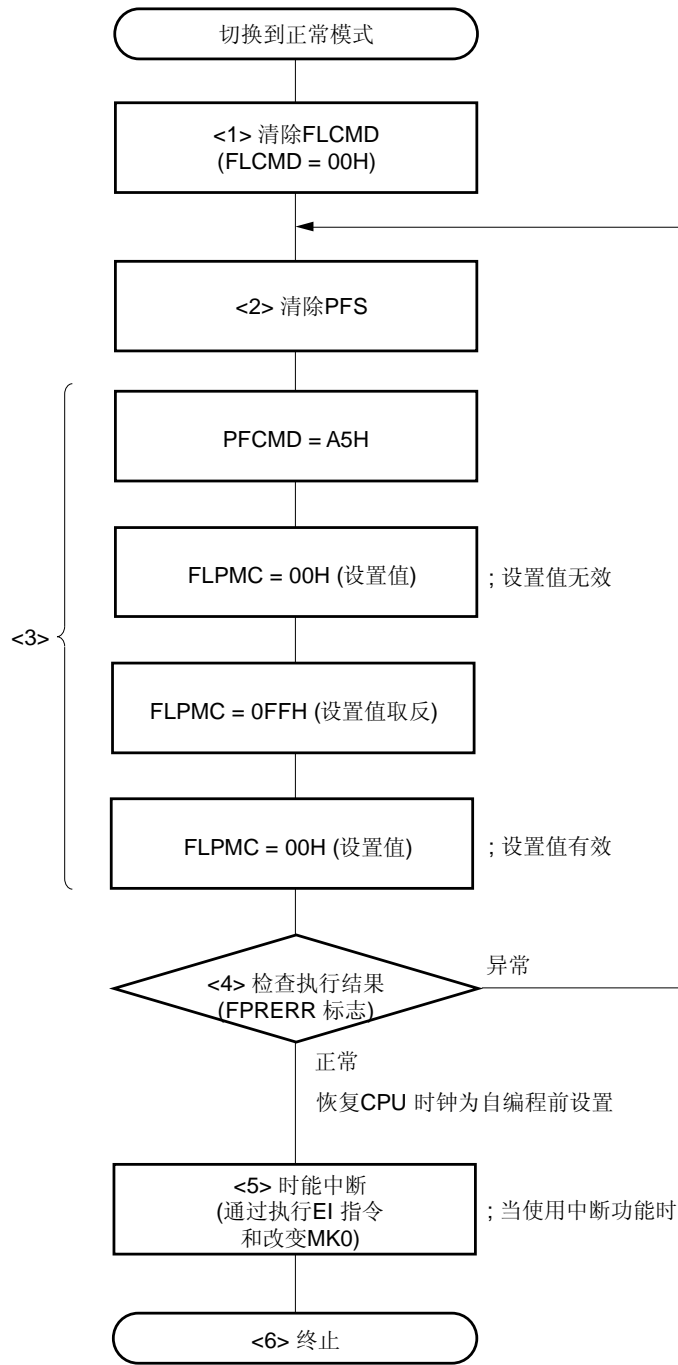
切换到正常模式的举例说明如下。

- <1> 清除 FLCMD (FLCMD = 00H)。
- <2> 清除 flash 状态寄存器 (PFS)。
- <3> 用指定顺序设置正常模式。
 - 写指定值 (A5H) 到 PFCMD。
 - 写 00H 到 FLPMC (该步写无效)。
 - 写 0FFH (00H 的相反值) 到 FLPMC (该步写无效)。
 - 写 00H 到 FLPMC (该步写有效)。
- <4> 用 PFS[#] 的第 0 位 (FPRERR) 检查指定顺序的执行结果。
异常 → <2>, 正常 → <5>
- <5> 允许中断服务 (通过执行 EI 指令和改变 MK0) 到恢复初始状态。
- <6> 模式切换完成

注 恢复 CPU 时钟到自编程前指定顺序正常完成后的设置。

注意事项 确保用在数据没有擦除或写入的地址处的用户程序执行上述的一系列操作。

图 15-21. 切换正常模式示例



注意事项 确保用在数据没有擦除或写入的地址处的用户程序执行上述的一系列操作。

备注 图 15-21 的<1> 到 <6> 对应 15.8.5 的<1> 到 <6>（前页）。

切换模式到正常模式的程序示例如下。

```

;-----
;开始
;-----

        MOV     FLCMD,#00H      ;清除 FLCMD 寄存器

ModeOffLoop:
        MOV     PFS,#00H       ;清除 flash 状态寄存器
        MOV     PFCMD,#0A5H    ;PFCMD 寄存器控制
        MOV     FLPMC,#00H     ;FLPMC 寄存器控制（设置值）
        MOV     FLPMC,#0FFH    ;FLPMC 寄存器控制（相反设置值）
        MOV     FLPMC,#00H     ;通过 FLPMC 寄存器控制设置正常模式（设置值）

        BT     PFS.0,$ModeOffLoop ;检查写指定寄存器完成
                                     ;当发生错误时重复相同的处理
                                     ;恢复 CPU 时钟到自编程前
                                     ;指定顺序正常完成后的设置

        MOV     MK0,#INT_MK0   ;恢复中断屏蔽标志

        EI

;-----
;结束
;-----

```

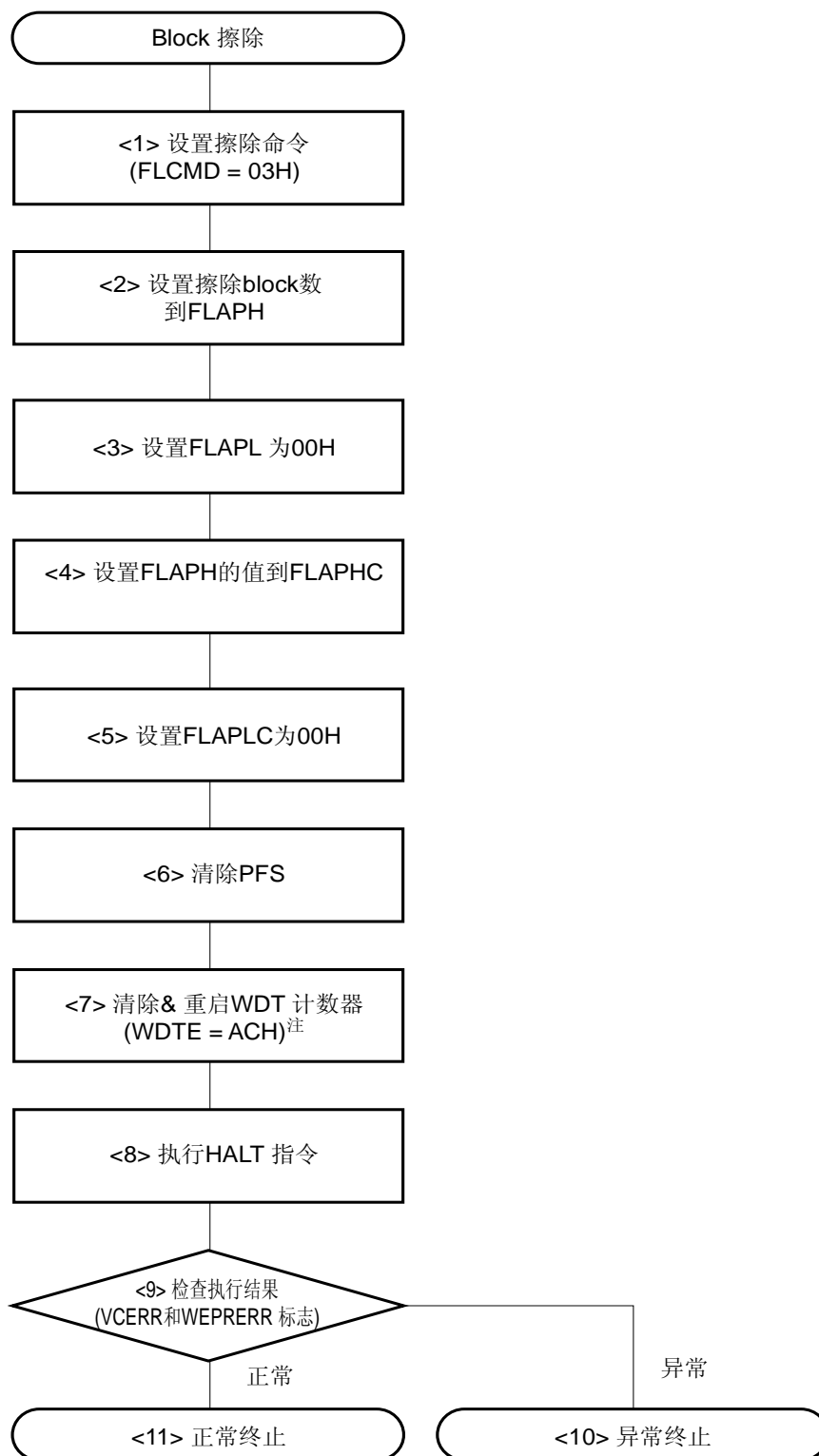
15.8.6 自编程模式下block擦除操作示例

自编程模式下 block 擦除操作举例说明如下。

- <1> 设置 03H（块擦除）到 flash 编程命令寄存器（FLCMD）。
- <2> 设置擦除 block 数到 flash 地址指针 H（FLAPH）。
- <3> 设置 flash 地址指针 L（FLAPL）为 00H。
- <4> 写与 FLAPH 相同的值到 flash 地址指针 H 比较寄存器（FLAPHC）。
- <5> 设置 flash 地址指针 L 比较寄存器（FLAPLC）到 00H。
- <6> 清除 flash 状态寄存器（PFS）。
- <7> 写 ACH 到看门狗定时器允许寄存器（WDTE）（清除和重启看门狗定时器计数器）[※]。
- <8> 执行 HALT 指令然后开始自编程。（如果自编程执行 HALT 指令后马上执行一条指令）
- <9> 用 PFS 的第 1 位（VCERR）和第 2 位（WEPRERR）检查是否产生了自编程错误。
 - 异常 → <10>
 - 正常 → <11>
- <10> Block 擦除处理被异常终止。
- <11> Block 擦除处理被异常终止。

注 当不使用看门狗定时器时不需要该设置。

图 15-22. 自编程模式下块擦除操作示例



注 当不使用看门狗定时器时不需要该设置。

备注 图 15-22 的<1> 到 <11> 对应 15.8.6 的<1> 到 <11> (前页)。

自编程模式下执行 block 擦除的程序举例如下。

```

;-----
;开始
;-----

FlashBlockErase:
    MOV    FLCMD,#03H    ;设置 flash 控制命令 (block 擦除)
    MOV    FLAPH,#07H    ;设置擦除的 block 数 (block 7 在此指定)
    MOV    FLAPL,#00H    ;固定 FLAPL 为 “00H”
    MOV    FLAPHC,#07H   ;设置擦除 block 比较数 (FLAPH 相同值)
    MOV    FLAPLC,#00H   ;固定 FLAPLC 为 “00H”

    MOV    PFS,#00H      ;清除 flash 状态寄存器
    MOV    WDTE,#0ACH    ;清除&重启 WDT
    HALT                                     ;自编程开始

    MOV    A,PFS
    MOV    CmdStatus,A    ;执行结果存储在变量中
                                ; (CmdStatus = 0:正常终止, 除了 0: 异常
                                ; 终止

;-----
;结束
;-----

```

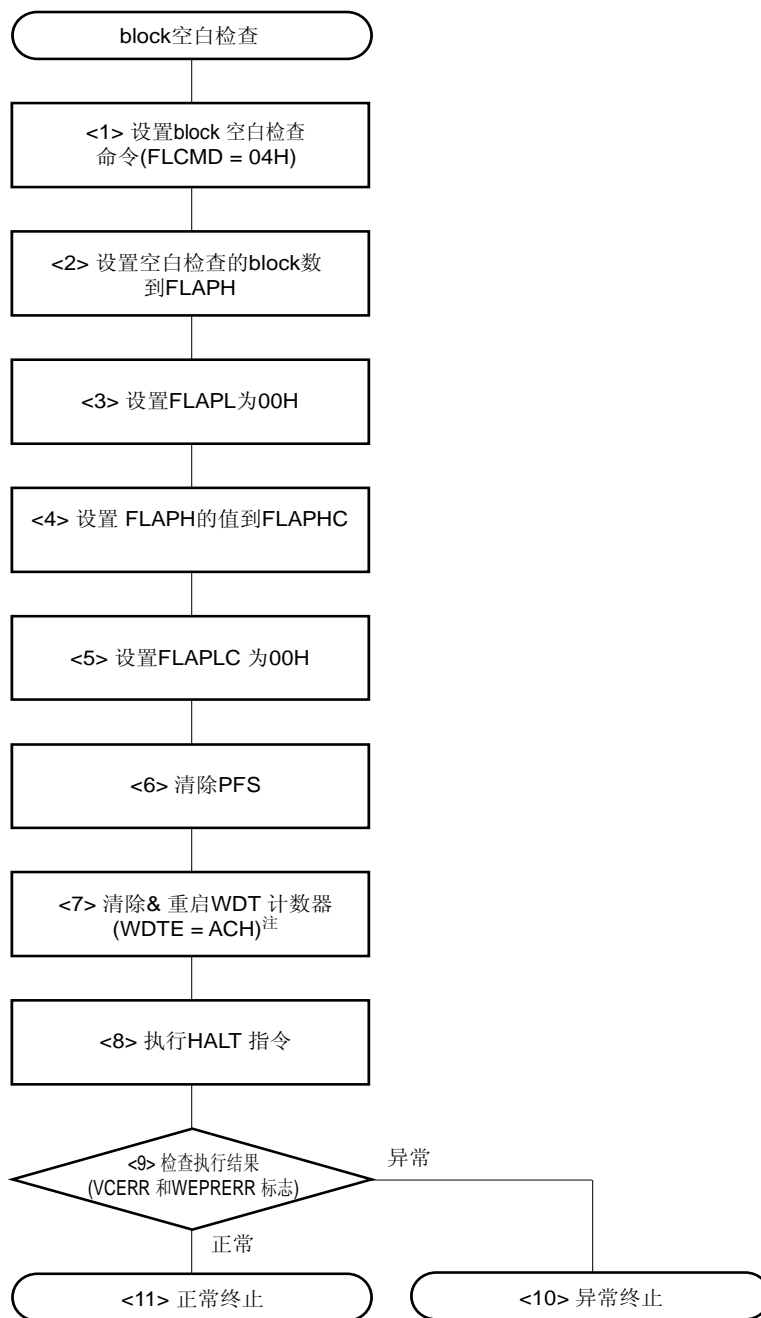
15.8.7 自编程模式下block空白检查操作示例

自编程模式下 block 空白检查操作举例说明如下。

- <1> 设置 04H (block 空白检查) 到 flash 编程命令寄存器 (FLCMD)。
- <2> 设置执行空白检查的 block 数到 flash 地址指针 H (FLAPH)。
- <3> 设置 flash 地址指针 L (FLAPL) 到 00H。
- <4> 写与 FLAPH 相同的值到 flash 地址指针 H 比较寄存器 (FLAPHC)。
- <5> 设置 flash 地址指针 L 比较寄存器 (FLAPLC) 到 FFH。
- <6> 清除 flash 状态寄存器 (PFS)。
- <7> 写 ACH 到看门狗定时器允许寄存器 (WDTE) (清除和重启看门狗定时器计数器) ^注。
- <8> 执行 HALT 指令然后开始自编程。(如果自编程执行 HALT 指令后马上执行一条指令)
- <9> 用 PFS 的第 1 位 (VCERR) 和第 2 位 (WEPRERR) 检查是否产生了自编程错误。
异常 → <10>
正常 → <11>
- <10> Block 空白检查被异常终止。
- <11> Block 空白检查被正常终止。

注 当不使用看门狗定时器时不需要该设置。

图 15-23. 自编程模式下 block 空白检查操作示例



注 当不使用看门狗定时器时不需要该设置。

备注 图 15-23 的<1> 到 <11> 对应 15.8.7 的<1> 到 <11> (前页)。

自编程模式下执行 block 空白检查的程序举例如下。

```

;-----
;开始
;-----

FlashBlockBlankCheck:
    MOV     FLCMD,#04H      ;设置 flash 控制命令 (block 空白检查)
    MOV     FLAPH,#07H     ;设置空白检查的 block 数 (block7
                          ;此时指定)
    MOV     FLAPL,#00H     ;固定 FLAPL 为 “00H”
    MOV     FLAPHC,#07H    ;设置空白检查 block 比较数 (和
                          ;FLAPH 的值相同)
    MOV     FLAPLC,#0FFH   ;固定 FLAPLC 到 “FFH”

    MOV     PFS,#00H       ;清除 flash 状态寄存器
    MOV     WDTE,#0ACH     ;清除&重启 WDT
    HALT                    ;自编程开始
    MOV     A,PFS
    MOV     CmdStatus,A    ;执行结果存储在变量中
                          ; (CmdStatus = 0:正常终止, 除了 0: 异常
                          ;终止)

;-----
;结束
;-----

```


15.8.8 自编程模式下字节写操作示例

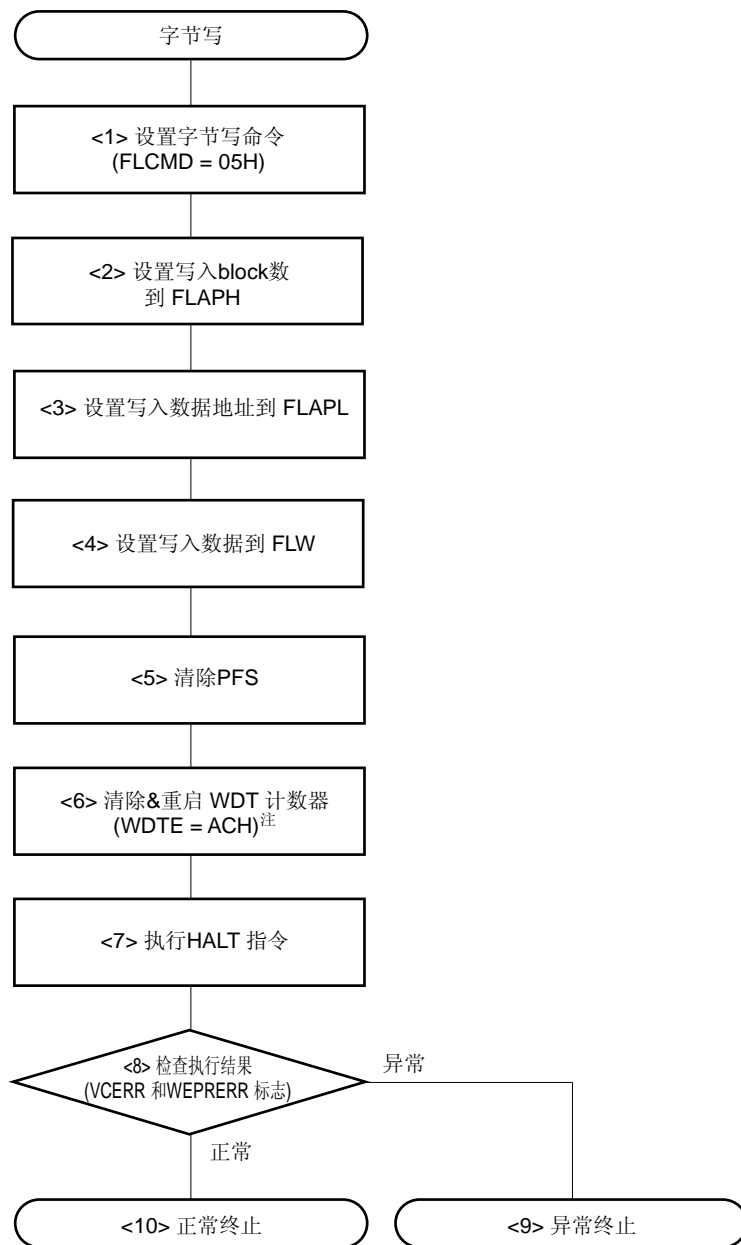
自编程模式下字节写操作举例说明如下。

- <1> 设置 05H (字节写) 到 flash 编程命令寄存器 (FLCMD)。
- <2> 设置将写数的 block 数到 flash 地址指针 H (FLAPH)。
- <3> 设置要写数据的地址到 flash 地址指针 L (FLAPL)。
- <4> 设置要写数据到 flash 写缓存寄存器 (FLW)。
- <5> 清除 flash 状态寄存器 (PFS)。
- <6> 写 ACH 到看门狗定时器允许寄存器 (WDTE) (清除和重启看门狗定时器计数器)^注。
- <7> 执行 HALT 指令然后开始自编程。(如果自编程执行 HALT 指令后马上执行一条指令)
- <8> 用 PFS 的第 1 位 (VCERR) 和第 2 位 (WEPRERR) 检查是否产生了自编程错误。
 - 异常 → <9>
 - 正常 → <10>
- <9> 字节写处理被异常终止。
- <10> 字节写处理被正常终止。

注 当不使用看门狗定时器时不需要该设置。

注意事项 如果写导致失败, 立即擦除 block 并再写。

图 15-24. 自编程模式下字节写操作示例



注 当不使用看门狗定时器时不需要该设置。

备注 图 15-24 的<1> 到 <10> 对应 15.8.8 的<1> 到 <10>（前页）。

自编程模式下执行字节写的程序举例如下。

```

;-----
;开始
;-----
FlashWrite:
    MOV    FLCMD,#05H    ;设置 flash 控制命令（字节写）
    MOV    FLAPH,#07H    ;设置写数据地址
                        ; FLAPH（block7 在此指定）
    MOV    FLAPL,#20H    ;设置被写数的地址
                        ; FLAPL（地址 20H 在此指定）
    MOV    FLW,#10H      ;设置被写数据（10H 在此指定）

    MOV    PFS,#00H      ;清除 flash 状态寄存器
    MOV    WDTE,#0ACH    ;清除&重启 WDT
    HALT
    MOV    A,PFS
    MOV    CmdStatus,A   ;执行结果存储在变量中
                        ;（CmdStatus = 0:异常终止，除了 0: 异常
                        ;终止）

;-----
;结束
;-----

```

15.8.9 自编程模式下内部校验操作示例

自编程模式下内部校验操作举例说明如下。

• 内部校验 1

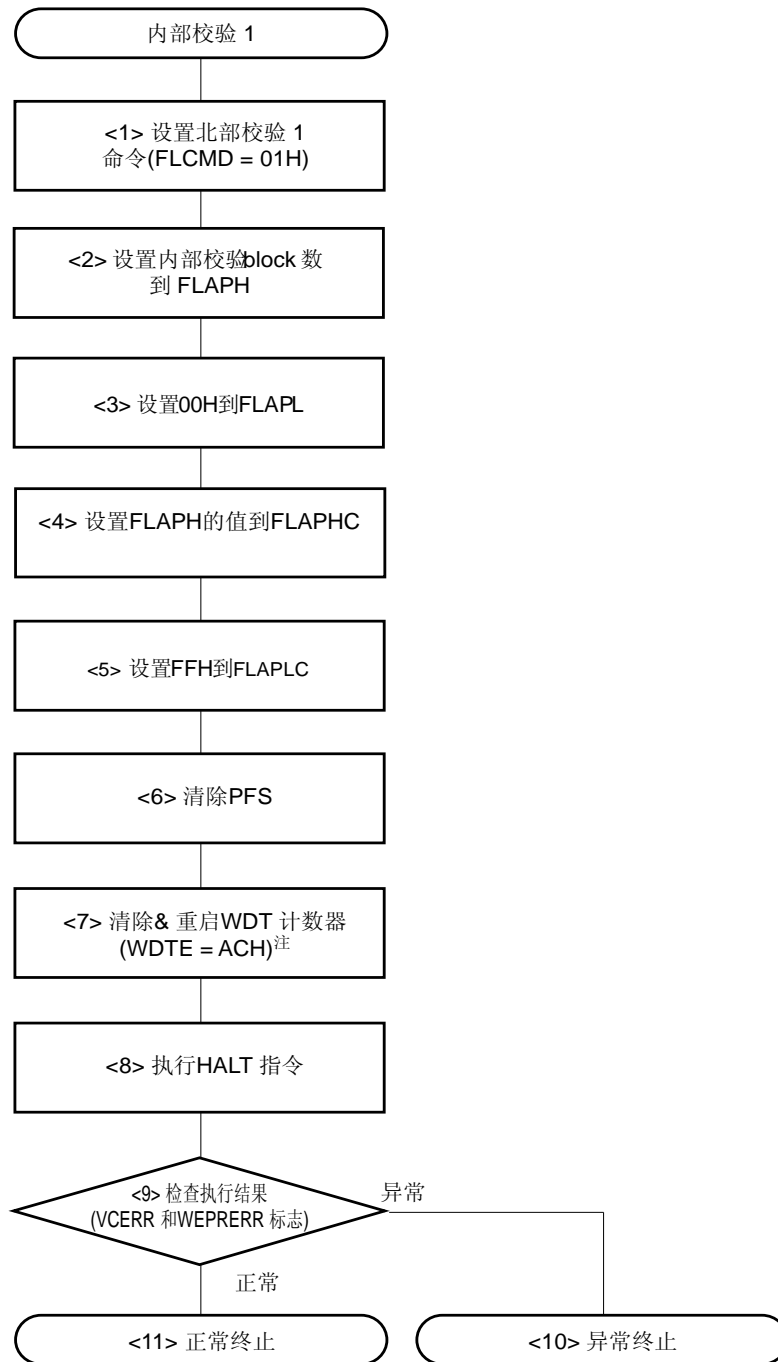
- <1> 设置 01H (内部校验 1) 到 flash 编程命令寄存器 (FLCMD)。
- <2> 设置执行内部校验的 block 数到 flash 地址指针 H (FLAPH)。
- <3> 设置 00H 到 flash 地址指针 L (FLAPL)。
- <4> 写与 FLAPH 相同的值到 flash 地址指针 H 比较寄存器 (FLAPHC)。
- <5> 设置 FFH 到 flash 地址指针 L 比较寄存器 (FLAPLC)。
- <6> 清除 flash 状态寄存器 (PFS)。
- <7> 写 ACH 到看门狗定时器允许寄存器 (WDTE) (清除和重启看门狗定时器计数器) ^注。
- <8> 执行 HALT 指令然后开始自编程。(如果自编程执行 HALT 指令后马上执行一条指令)
- <9> 用 PFS 的第 1 位 (VCERR) 和第 2 位 (WEPRERR) 检查是否产生了自编程错误。
异常 → <10>
正常 → <11>
- <10> 内部校验处理被异常终止。
- <11> 内部校验处理被正常终止。

• 内部校验 2

- <1> 设置 02H (内部校验 2) 到 flash 编程命令寄存器 (FLCMD)。
- <2> 设置执行内部校验的 block 数到 flash 地址指针 H (FLAPH)。
- <3> 设置校验开始地址到 flash 地址指针 L (FLAPL)。
- <4> 写与 FLAPH 相同的值到 flash 地址指针 H 比较寄存器 (FLAPHC)。
- <5> 设置校验结束地址到 flash 地址指针 L 比较寄存器 (FLAPLC)。
- <6> 清除 flash 状态寄存器 (PFS)。
- <7> 写 ACH 到看门狗定时器允许寄存器 (WDTE) (清除和重启看门狗定时器计数器) ^注。
- <8> 执行 HALT 指令然后开始自编程。(如果自编程执行 HALT 指令后马上执行一条指令)
- <9> 用 PFS 的第 1 位 (VCERR) 和第 2 位 (WEPRERR) 检查是否产生了自编程错误。
异常 → <10>
正常 → <11>
- <10> 内部校验处理被异常终止。
- <11> 内部校验处理被正常终止。

注 当不使用看门狗定时器时不需要该设置。

图 15-25. 自编程模式下内部校验 1 操作示例



注 当不使用看门狗定时器时不需要该设置。

备注 图 15-25 的<1> 到 <11> 对应 15.8.9 的内部校验 1<1> 到 <11> (前页)。

图 15-26. 自编程模式下内部校验 2 操作示例



注 当不使用看门狗定时器时不需要该设置。

备注 图 15-26 的<1> 到 <11> 对应 15.8.9 的内部校验 2<1> 到 <11>（前页）。

自编程模式下内部校验 1 和 2 编程举例如下。

- 内部校验 1

```

;-----
;开始
;-----
FlashVerify:
    MOV    FLCMD,#01H    ;设置 flash 控制命令（内部校验 1）
    MOV    FLAPH,#07H    ;设置执行内部校验的 block 数
                        ;到 FLAPH（例：; block7 在此指定）
    MOV    FLAPL,#00H    ;设置 00H
    MOV    FLAPHC,#07H
    MOV    FLAPLC,#FFH    ;设置 FFH

    MOV    PFS,#00H      ;清除 flash 状态寄存器
    MOV    WDTE,#0ACH    ;清除&重启 WDT
    HALT
    MOV    A,PFS
    MOV    CmdStatus,A    ;执行结果存储在变量中
                        ;（CmdStatus = 0:异常终止，除了 0: 异常
                        ;终止）

;-----
;结束
;-----

```

- 内部校验 2

```

;-----
;开始
;-----
FlashVerify:
    MOV    FLCMD,#02H    ;设置 flash 控制命令（内部校验 2）
    MOV    FLAPH,#07H    ;设置执行内部校验的 block 数
                        ;到 FLAPH（例： Block 7 is specified here）
    MOV    FLAPL,#00H    ;设置 FLAPL 为校验开始地址（例：地址
                        ;00H 被指定）
    MOV    FLAPHC,#07H
    MOV    FLAPLC,#20H    ;设置 FLAPLC 为校验结束地址（例：地址
                        ;20H 被指定）

    MOV    PFS,#00H      ;清除 flash 状态寄存器
    MOV    WDTE,#0ACH    ;清除&重启 WDT
    HALT
    MOV    A,PFS
    MOV    CmdStatus,A    ;执行结果存储在变量中
                        ;（CmdStatus = 0:异常终止，除了 0: 异常
                        ;终止）

;-----
;结束
;-----

```

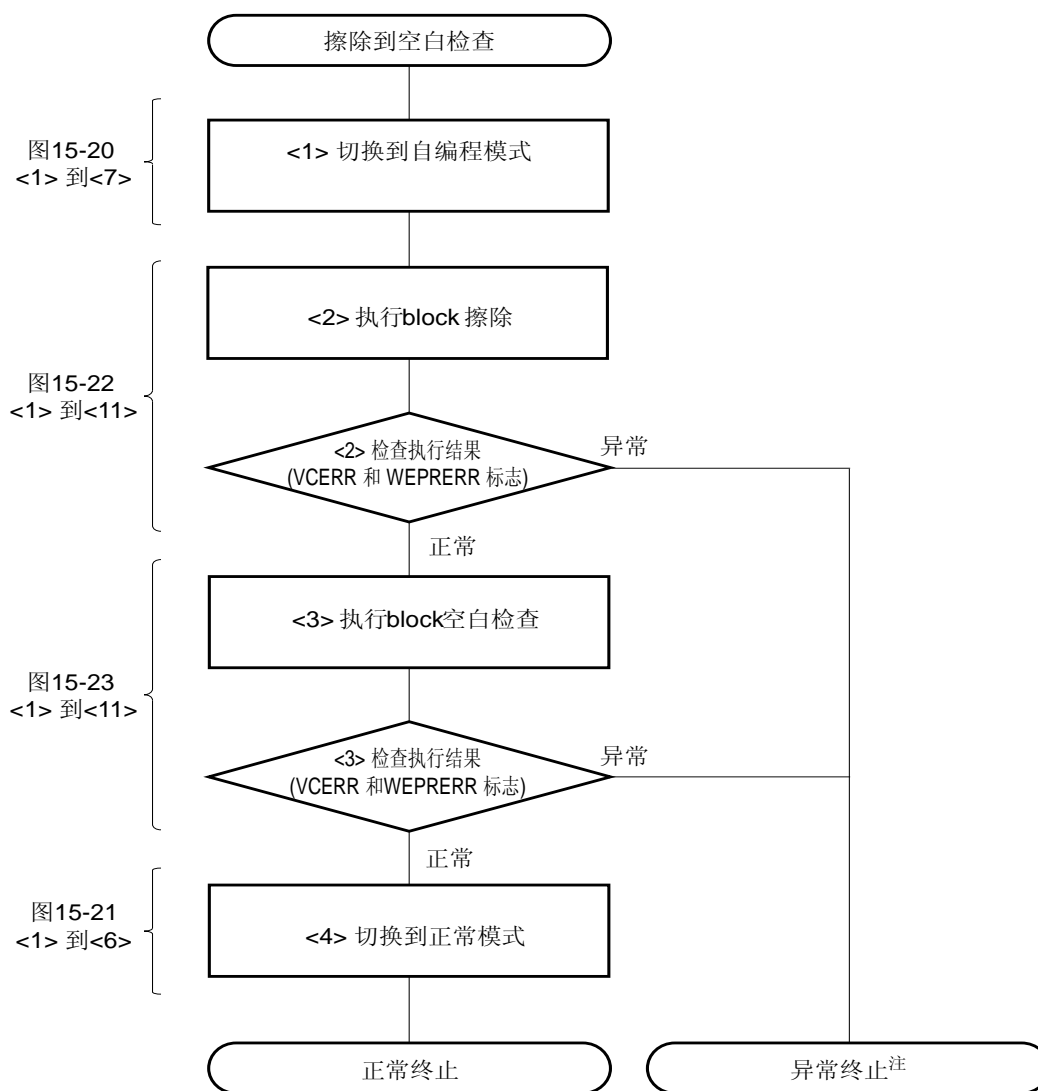
15.8.10 自编程模式下命令执行时间最小化时操作举例

自编程模式下命令执行时间最小化时操作举例说明如下。

(1) 擦除到空白检查

- <1> 从正常模式切换到自编程模式（15.8.4 中的<1> 到 <7>）
- <2> 执行 block 擦除→错误检查（15.8.6 中的<1> 到 <11>）
- <3> 执行 block 空白检查→错误检查（15.8.7 中的<1> 到 <11>）
- <4> 从自编程模式切换到正常模式（15.8.5 中的<1> 到 <6>）

图 15-27.命令执行时间最小化操作示例
(从擦除到空白检查)



注 为了返回正常处理切换到正常模式

备注 图 15-27 的<1> 到 <4> 对应 15.8.10 (1) 的<1> 到 <4> (上面)。

自编程模式下命令执行时间（从擦除到空白检查）最小化程序举例如下。

```

;-----
;开始
;-----

        MOV     MK0,#11111111B    ;屏蔽所有中断
        MOV     FLCMD,#00H        ;清除 FLCMD 寄存器

        DI

ModeOnLoop:
        MOV     PFS,#00H          ;清除 flash 状态寄存器
        MOV     PFCMD,#0A5H      ;PFCMD 寄存器控制
        MOV     FLPMC,#01H       ;FLPMC 寄存器控制（设置值）
        MOV     FLPMC,#0FEH      ;FLPMC 寄存器控制（相反设置值）
        MOV     FLPMC,#01H       ;用 FLPMC 控制寄存器设置自编程模式（设置
                                   ;值）

        NOP
        HALT
        BT      PFS.0,$ModeOnLoop; 检查写指定寄存器完成
                                   ;当发生错误时重复相同的处理

FlashBlockErase:
        MOV     FLCMD,#03H        ;设置 flash 控制命令（block 擦除）
        MOV     FLAPH,#07H        ;设置擦除的 block 数（block1
                                   ;此时指定）
        MOV     FLAPL,#00H        ;固定 FLAPL 为“00H”
        MOV     FLAPHC,#07H       ;设置擦除 block 比较值（和
                                   ;FLAPH 相同）
        MOV     FLAPLC,#00H       ;固定为“00H”

        MOV     WDTE,#0ACH        ;清除&重启 WDT
        HALT                       ;自编程开始
        MOV     A,PFS
        CMP     A,#00H
        BNZ     $StatusError      ;检查擦除错误
                                   ;当错误发生时执行异常中断处理

FlashBlockBlankCheck:
        MOV     FLCMD,#04H        ;设置 flash 控制命令（block 空白检查）
        MOV     FLAPH,#07H        ;设置空白检查的 block 数（block7
                                   ;此时指定）
        MOV     FLAPL,#00H        ;固定 FLAPL 为“00H”

```

```

MOV     FLAPHC,#07H      ;设置空白检查 block 比较数（和                ; FLAPH 相同）
MOV     FLAPLC,#0FFH    ;固定 FLAPLC 到“FFH”
MOV     WDTE,#0ACH      ;清除&重启 WDT
HALT                    ;自编程开始
MOV     A,PFS
CMP     A,#00H
BNZ     $StatusError    ;检查空白检查错误
                                ;执行异常中断处理当错误                ;发生。

MOV     FLCMD,#00H      ;清除 FLCMD 寄存器
ModeOffLoop:
MOV     PFS,#00H        ;清除 flash 状态寄存器
MOV     PFCMD,#0A5H     ;PFCMD 寄存器控制
MOV     FLPMC,#00H      ;FLPMC 寄存器控制（设置值）
MOV     FLPMC,#0FFH     ;FLPMC 寄存器控制（相反设置值）
MOV     FLPMC,#00H      ;通过 FLPMC 寄存器控制设置正常模式（设置值）

BT      PFS.0,$ModeOffLoop ;检查写指定寄存器完成
                                ;当发生错误时重复相同的处理
                                ;恢复 CPU 时钟到自编程前
                                ;指定顺序正常完成后的设置
                                ;
MOV     MK0,#INT_MK0    ;恢复中断屏蔽标志

EI

BR      状态异常

;-----
;结束（异常结束处理）；执行
切换到正常模式来返回正常处理
;-----
StatusError:

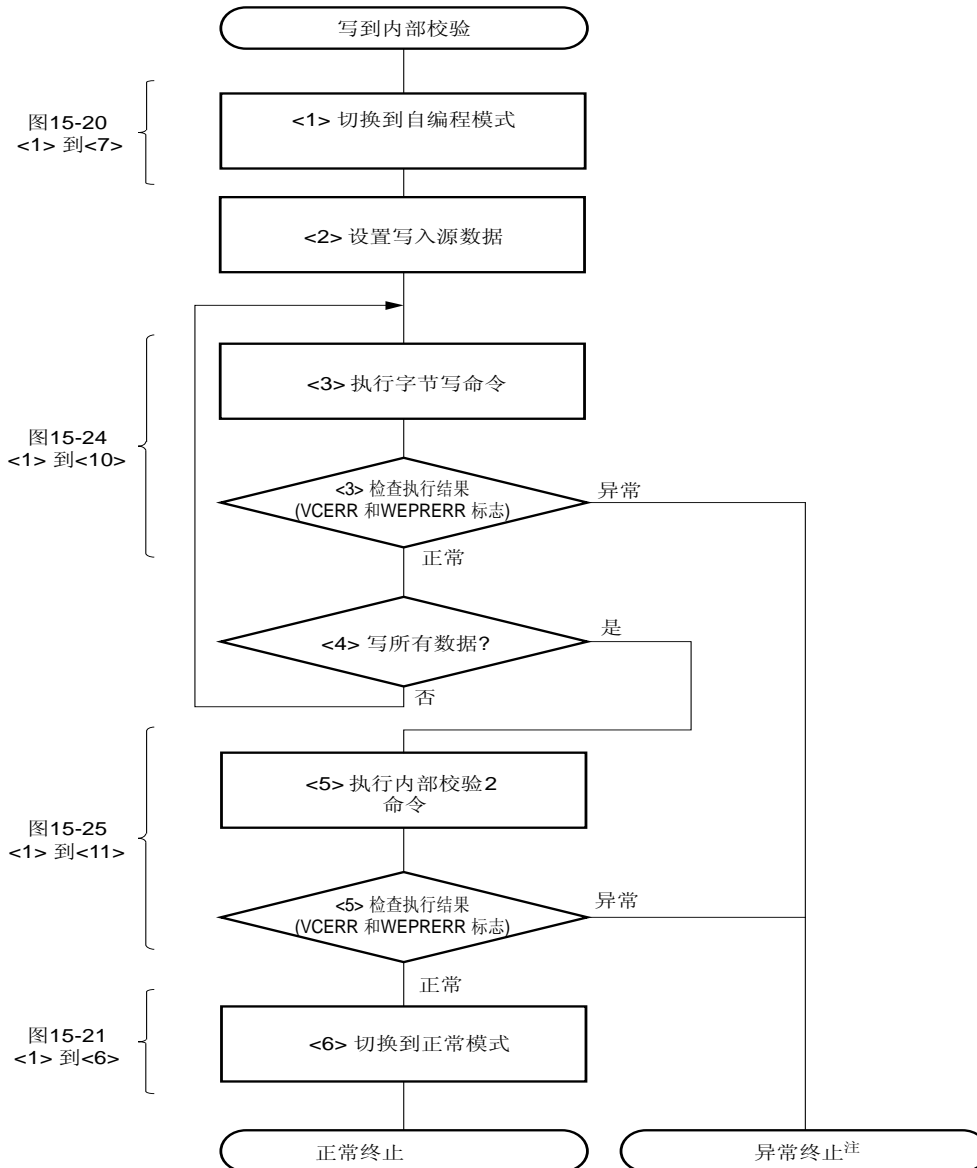
;-----
;结束（正常终止处理）
;-----
StatusNormal:

```

(2) 写入到内部校验

- <1> 从正常模式切换到自编程模式 (15.8.4 中的<1> 到 <7>)
- <2> 指定写入源数据
- <3> 执行字节写→错误检查 (15.8.8 中的<1> 到 <10>)
- <4> <3>重复直到写所有数据。
- <5> 执行内部校验 2→错误检查 (15.8.9 中的<1> 到 <11>)
- <6> 从自编程模式切换到正常模式 (15.8.5 中的<1> 到 <6>)

图 15-28.命令执行时间最小化操作示例
(从写入到内部校验)



注 为了返回正常处理切换到正常模式

备注 图 15-28 的<1> 到 <6> 对应 15.8.10 (2) 的<1> 到 <6> (上面)。

自编程模式下命令执行时间（从写到内部校验）最小化程序举例如下。

```

;-----
;开始
;-----
        MOV     MK0,#11111111B    ;屏蔽所有中断
        MOV     FLCMD,#00H        ;清除 FLCMD 寄存器
        DI

ModeOnLoop:                                ;配置设置使 CPU 时钟≥ 1 MHz
        MOV     PFS,#00H          ;清除 flash 状态寄存器
        MOV     PFCMD,#0A5H       ;PFCMD 寄存器控制
        MOV     FLPMC,#01H        ;FLPMC 寄存器控制（设置值）
        MOV     FLPMC,#0FEH       ;FLPMC 寄存器控制（相反设置值）
        MOV     FLPMC,#01H        ;用 FLPMC 寄存器设置自编程模式
                                        ;（设置值）

        NOP
        HALT
        BT     PFS.0,$ModeOnLoop; 检查写指定寄存器完成
                                        ;当发生错误时重复相同的处理

FlashWrite:
        MOVW    HL,#DataAdrTop    ;设置写入数据的地址
        MOVW    DE,#WriteAdr      ;设置写入数据的地址
FlashWriteLoop:
        MOV     FLCMD,#05H        ;设置 flash 控制命令（字节写）
        MOV     A,D
        MOV     FLAPH,A           ;设置写入数据的地址
        MOV     A,E
        MOV     FLAPL,A           ;设置写入数据的地址
        MOV     A,[HL]
        MOV     FLW,A             ;设置写入数据

        MOV     WDTE,#0ACH        ;清除&重启 WDT
        HALT                        ;自编程开始
        MOV     A,PFS
        CMP     A,#00H
        BNZ     $StatusError       ;检查写入错误
                                        ;执行异常终止处理当错误
                                        ;发生。

        INCW    HL                ;设置写入数据的地址+ 1
        MOVW    AX,HL
        CMPW    AX,#DataAdrBtm    ;执行内部校验处理
        BNC     $FlashVerify       ;如果所有数据写完成
    
```

```

INCW    DE                ; 设置写入数据的地址+ 1
BR      FlashWriteLoop

FlashVerify:
MOVW    HL,#WriteAdr     ; 设置校验地址

MOV     FLCMD,#02H       ; 设置 flash 控制命令（内部校验 2）
MOV     A,H
MOV     FLAPH,A         ; 设置校验开始地址
MOV     A,L
MOV     FLAPL,A         ; 设置校验开始地址
MOV     A,D
MOV     FLAPHC,A        ; 设置校验结束地址
MOV     A,E
MOV     FLAPLC,A        ; 设置校验结束地址

MOV     WDTE,#0ACH      ; 清除&重启 WDT
HALT    ; 自编程开始

MOV     A,PFS
CMP     A,#00H
BNZ    $StatusError    ; 检查内部校验错误
                        ; 执行异常终止处理当错误
                        ; 发生。

MOV     FLCMD,#00H      ; 清除 FLCMD 寄存器

ModeOffLoop:
MOV     PFS,#00H        ; 清除 flash 状态寄存器
MOV     PFCMD,#0A5H     ; PFCMD 寄存器控制
MOV     FLPMC,#00H      ; FLPMC 寄存器控制（设置值）
MOV     FLPMC,#0FFH     ; FLPMC 寄存器控制（相反设置值）
MOV     FLPMC,#00H      ; 通过 FLPMC 寄存器控制设置正常模式（设置值）

BT     PFS.0,$ModeOffLoop ; 检查写指定寄存器完成
                        ; 当发生错误时重复相同的处理
                        ; 恢复 CPU 时钟到自编程前
                        ; 指定顺序正常完成后的设置
                        ;
MOV     MK0,#INT_MK0    ; 恢复中断屏蔽标志

EI

BR      StatusNormal

```

```

;-----
;结束（异常终止处理）；执行
;切换到正常模式来返回正常操作
;-----

```

StatusError:

```

;-----
;结束 (正常终止处理)
;-----
StatusNormal:

;-----
;写入数据
;-----
DataAdrTop:
    DB    XXH
    DB    XXH
    DB    XXH
    DB    XXH

    :
    :

    DB    XXH
DataAdrBtm:
;-----

```

备注 内部校验 2 用于以上程序示例中。用内部校验 1 来校验一整 block。

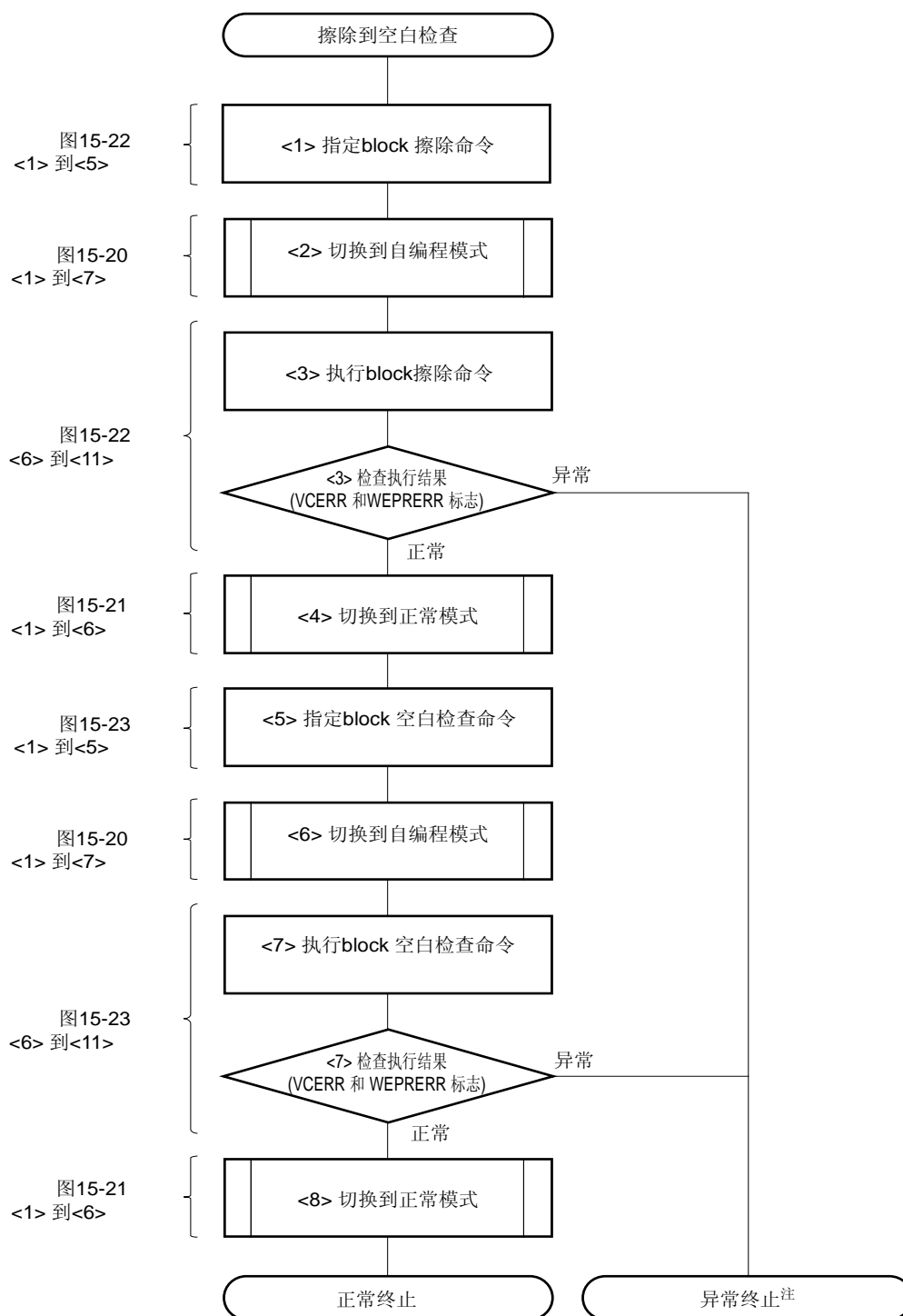
15.8.11 自编程模式下中断禁止时间最小化操作举例

自编程模式下中断禁止时间最小化操作举例说明如下。

(1) 擦除到空白检查

- <1> Block 擦除命令指定（15.8.6 中<1> 到 <5>）
- <2> 从正常模式切换到自编程模式（15.8.4 中的<1> 到 <7>）
- <3> 执行 block 擦除命令→错误检查（15.8.6 中的<6> 到 <11>）
- <4> 从自编程模式切换到正常模式（15.8.5 中的<1> 到 <6>）
- <5> Block 空白检查命令指定（15.8.7 中<1> 到 <5>）
- <6> 从正常模式切换到自编程模式（15.8.4 中的<1> 到 <7>）
- <7> 执行 block 空白检查命令→错误检查（15.8.7 中的<6> 到 <11>）
- <8> 从自编程模式切换到正常模式（15.8.5 中的<1> 到 <6>）

图 15-29. 中断禁止时间最小化操作示例
(从擦除到空白检查)



注 为了返回正常处理切换到正常模式

备注 图 15-29 的<1> 到 <8> 对应 15.8.11 (1) 的<1> 到 <8> (前页)。

自编程模式下中断禁止时间（从擦除到空白检查）最小化程序举例如下。

```

;-----
;开始
;-----
FlashBlockErase:
    ; 设置擦除命令
    MOV     FLCMD,#03H      ; 设置 flash 控制命令（block 擦除）
    MOV     FLAPH,#07H     ; 设置擦除 block 数（block 7 此处指定）
    MOV     FLAPL,#00H     ; 固定 FLAPL 为 “00H”
    MOV     FLAPHC,#07H    ; 设置擦除 block 比较数（和 FLAPH 的值相同）
    MOV     FLAPLC,#00H    ; 固定 FLAPLC 为 “00H”

    CALL    IModeOn        ; 切换到自编程模式

    ; 擦除命令执行
    MOV     PFS,#00H      ; 清除 flash 状态寄存器
    MOV     WDTE,#0ACH    ; 清除&重启 WDT
    HALT                    ; 自编程开始

    MOV     A,PFS
    CMP     A,#00H
    BNZ     $StatusError   ; 检查擦除错误
                                ; 执行异常终止处理当错误
                                ; 发生。

    CALL    IModeOff       ; 切换到正藏模式

    ; 设置空白检查命令
    MOV     FLCMD,#04H    ; 设置 flash 控制命令（block 空白检查）
    MOV     FLAPH,#07H    ; 设置空白检查 block 数（block7 此处指定）
    MOV     FLAPL,#00H    ; 固定 FLAPL 为i°00Hi±
    MOV     FLAPHC,#07H   ; 设置空白检查 block 比较数（相同于
                                ; FLAPH 的值）
    MOV     FLAPLC,#0FFH  ; 固定 FLAPLC 到i°FFHi±

    CALL    IModeOn        ; 切换到自编程模式

    ; 空白检查命令执行
    MOV     PFS,#00H      ; 清除 flash 状态寄存器
    MOV     WDTE,#0ACH    ; 清除&重启 WDT
    HALT                    ; 自编程开始

    MOV     A,PFS
    CMP     A,#00H
    BNZ     $StatusError   ; 检查空白检查错误
                                ; 当错误产生时执行异常终止处理

```

```

CALL    !ModeOff      ;切换到正常模式

BR      StatusNormal

;-----
;结束（异常终止处理）；执行处理
为返回正常处理切换到正常模式
;-----
StatusError:

;-----
;结束（正常中断处理）
;-----
StatusNormal:

;-----
;切换到自编程模式处理
;-----
ModeOn:
MOV     MK0,#11111111B ;屏蔽所有中断
MOV     FLCMD,#00H     ;清除 FLCMD 寄存器

DI

ModeOnLoop:
MOV     PFS,#00H       ;清除 flash 状态寄存器
MOV     PFCMD,#0A5H    ;PFCMD 寄存器控制
MOV     FLPMC,#01H     ;FLPMC 寄存器控制（设置值）
MOV     FLPMC,#0FEH    ;FLPMC 寄存器控制（相反设置值）
MOV     FLPMC,#01H     ;通过 FLPMC 寄存器控制设置自编程模式（设置
                        ;值）

NOP
HALT
BT      PFS.0,$ModeOnLoop ;检查写指定寄存器完成
                        ;当发生错误时重复相同的处理

RET

```

```

;-----
; 切换到正常模式处理
;-----
ModeOffLoop:
    MOV     FLCMD,#00H      ;清除 FLCMD 寄存器
    MOV     PFS,#00H       ;清除 flash 状态寄存器
    MOV     PFCMD,#0A5H    ;PFCMD 寄存器控制
    MOV     FLPMC,#00H     ;FLPMC 寄存器控制（设置值）
    MOV     FLPMC,#0FFH    ;FLPMC 寄存器控制（相反设置值）
    MOV     FLPMC,#00H     ;通过 FLPMC 寄存器控制设置正常模式（设置值）

    BT     PFS.0,$ModeOffLoop ;检查写指定寄存器完成
                                ;当发生错误时重复相同的处理
                                ;恢复 CPU 时钟到自编程前
                                ;指定顺序正常完成后的设置
                                ;
    MOV     MK0,#INT_MK0    ;恢复中断屏蔽标志

    EI

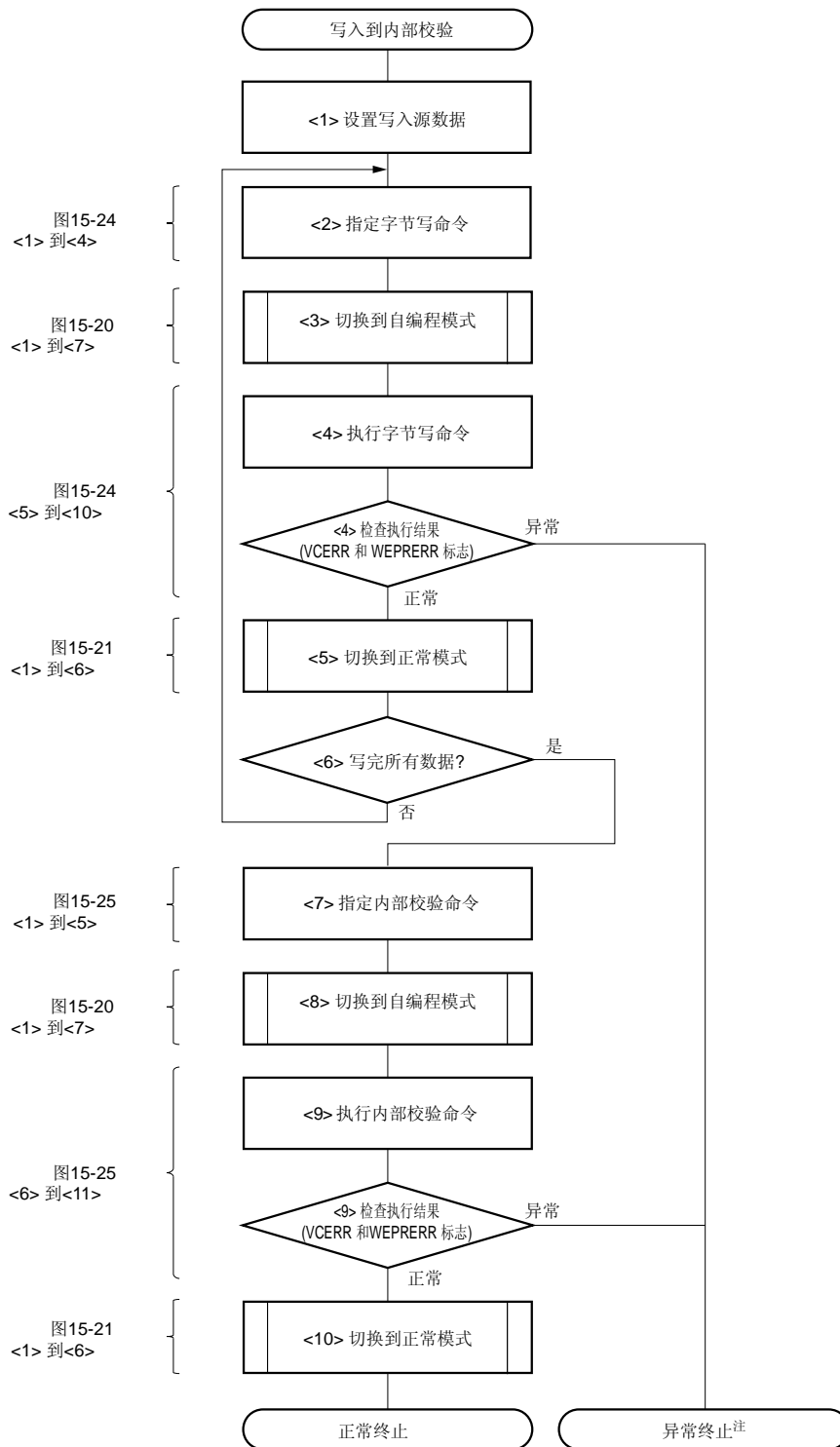
    RET

```

(2) 写入到内部校验

- <1> 指定写入源数据
- <2> 字节写命令指定（**15.8.8** 中<1> 到 <4>）
- <3> 从正常模式切换到自编程模式（**15.8.4** 中的<1> 到 <7>）
- <4> 执行字节写→错误检查（**15.8.8** 中的<5> 到 <10>）
- <5> 从自编程模式切换到正常模式（**15.8.5** 中的<1> 到 <6>）
- <6> <2> 到 <5>重复直到写所有数据。
- <7> 指定内部校验命令（**15.8.9** 中的<1> 到 <5>）
- <8> 从正常模式切换到自编程模式（**15.8.4** 中的<1> 到 <7>）
- <9> 执行内部校验命令→错误检查（**15.8.9** 中的<6> 到 <11>）
- <10> 从自编程模式切换到正常模式（**15.8.5** 中的<1> 到 <6>）

图 15-30. 中断禁止时间最小化操作示例
(从写入到内部校验)



注 为了返回正常处理切换到正常模式

备注 图 15-30 的<1> 到 <10> 对应 15.8.11 (2) 的<1> 到 <10> (前页)。

自编程模式下中断禁止时间（从写到内部校验）最小化程序举例如下。

```

;-----
;开始
;-----
;设置写命令
FlashWrite:
    MOVW    HL,#DataAdrTop    ;设置写入数据的地址
    MOVW    DE,#WriteAdr      ;设置写入数据的地址

FlashWriteLoop:
    MOV     FLCMD,#05H        ;设置 flash 控制命令（字节写）
    MOV     A,D
    MOV     FLAPH,A          ;设置写入数据的地址
    MOV     A,E
    MOV     FLAPL,A          ;设置写入数据的地址
    MOV     A,[HL]
    MOV     FLW,A            ;设置写入数据

    CALL    !ModeOn          ;切换到自编程模式

;写命令执行
    MOV     PFS,#00H          ;清除 flash 状态寄存器
    MOV     WDTE,#0ACH        ;清除&重启 WDT
    HALT                                ;自编程开始
    MOV     A,PFS
    CMP     A,#00H
    BNZ     $StatusError      ;检查写错误
                                ;执行异常终止处理当错误
                                ;发生。

    CALL    !ModeOff          ;切换到正常模式

    MOV     MK0,#INT_MK0      ;恢复中断屏蔽标志

EI

;写所有数据判断
    INCW    HL                ;写入数据的地址+ 1
    MOVW    AX,HL
    CMPW    AX,#DataAdrBtm    ;执行内部校验处理
    BNC     $FlashVerify      ;如果所有数据的写完场

    INCW    DE                ;写入数据的地址+1
    BR     FlashWriteLoop

;设置内部校验命令

```

FlashVerify:

```

MOVW    HL,#WriteAdr    ;设置校验地址

MOV     FLCMD,#02H      ;设置 flash 控制命令（内部校验 2）
MOV     A,H
MOV     FLAPH,A         ;设置校开始验地址
MOV     A,L
MOV     FLAPL,A         ;设置校验开始地址
MOV     A,D
MOV     FLAPHC,A        ;设置校验结束地址
MOV     A,E
MOV     FLAPLC,A        ;设置校验结束地址

CALL    IModeOn         ;切换到自编程模式

;内部校验命令执行
MOV     PFS,#00H        ;清除 flash 状态寄存器
MOV     WDTE,#0ACH      ;清除&重启 WDT
HALT
MOV     A,PFS
CMP     A,#00H
BNZ     $StatusError    ;检查内部校验错误
                        ;执行异常终止处理当错误发生

CALL    IModeOff        ;切换到正常模式

BR     StatusNormal

```

```

;-----
;结束（异常终止处理）；执行处理
为返回正常处理切换到正常模式
;-----

```

StatusError:

```

;-----
;结束（正常中断处理）
;-----

```

StatusNormal:

```

;-----
;切换到自编程模式处理
;-----
ModeOn:
    MOV     MK0,#11111111B   ;屏蔽所有中断
    MOV     FLCMD,#00H      ;清除 FLCMD 寄存器

    DI

ModeOnLoop:
                                ;配置设置使 CPU 时钟≥ 1 MHz
    MOV     PFS,#00H        ;清除 flash 状态寄存器
    MOV     PFCMD,#0A5H     ;PFCMD 寄存器控制
    MOV     FLPMC,#01H      ;FLPMC 寄存器控制（设置值）
    MOV     FLPMC,#0FEH     ;FLPMC 寄存器控制（相反设置值）
    MOV     FLPMC,#01H      ;通过 FLPMC 寄存器控制设置自编程模式（设置
                                ;值）

    NOP
    HALT
    BT      PFS.0,$ModeOnLoop ;检查写指定寄存器完成
                                ;当发生错误时重复相同的处理

    RET

;-----
;切换到正常模式处理
;-----
ModeOffLoop:
    MOV     FLCMD,#00H      ;清除 FLCMD 寄存器
    MOV     PFS,#00H        ;清除 flash 状态寄存器
    MOV     PFCMD,#0A5H     ;PFCMD 寄存器控制
    MOV     FLPMC,#00H      ;FLPMC 寄存器控制（设置值）
    MOV     FLPMC,#0FFH     ;FLPMC 寄存器控制（相反设置值）
    MOV     FLPMC,#00H      ;通过 FLPMC 寄存器控制设置正常模式（设置值）

    BT      PFS.0,$ModeOffloop ;检查写指定寄存器完成
                                ;当发生错误时重复相同的处理
                                ;恢复 CPU 时钟到自编程前
                                ;指定顺序正常完成后的设置
                                ;sequence
    MOV     MK0,#INT_MK0    ;恢复中断屏蔽标志

    EI

    RET

```



```

;-----
; 写入数据
;-----
DataAdrTop:
    DB      XXH
    DB      XXH
    DB      XXH
    DB      XXH

    :
    :

    DB      XXH
DataAdrBtm:
;-----

```

备注 内部校验 2 用于以上程序示例中。用内部校验 1 来校验一整 block。

第十六章 指令集概述

本章列出 μ PD78F9510、78F9511、78F9512 的指令集。关于每条指令的操作和机器语言（指令代码），参考 78K/0S 系列指令用户手册（U11047E）。

16.1 操作

16.1.1 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 16-1. 操作数标识符和描述方法

标识符	描述方法
r rp sfr	X (R0)、A (R1)、C (R2)、B (R3)、E (R4)、D (R5)、L (R6)、H (R7) AX (RP0)、BC (RP1)、DE (RP2)、HL (RP3) 特殊功能寄存器符号
saddr saddrp	FE20H 到 FF1FH 立即数或标签 FE20H 到 FF1FH 立即数或标签（只能是偶数地址）
addr16 addr5	0000H 到 FFFFH 立即数或标签（只能是 16 位数据传输指令的偶数地址） 0040H 到 007FH 立即数或标签（只能是偶数地址）
word byte bit	16 位立即数或标签 8 位立即数或标签 3 位立即数或标签

备注 关于特殊功能寄存器的符号，参见表 3-3 特殊功能寄存器。

16.1.2 “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
IE:	中断请求允许标志
():	由括号中的地址或寄存器内容表示的存储器内容
×H、×L:	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
jdisp8:	有符号 8 位数据 (位移值)

16.1.3 “标志”栏的说明

(空白):	不变
0:	清除为 0
1:	置位为 1
×:	根据结果置位/清除
R:	上次保存的值被恢复

16.2 操作列表

助记符	操作数	字节	时钟	操作	标志
					Z AC CY
MOV	r, #byte	3	6	r ← byte	
	saddr, #byte	3	6	(saddr) ← byte	
	sfr, #byte	3	6	sfr ← byte	
	A, r ^{注 1}	2	4	A ← r	
	r, A ^{注 1}	2	4	r ← A	
	A, saddr	2	4	A ← (saddr)	
	saddr, A	2	4	(saddr) ← A	
	A, sfr	2	4	A ← sfr	
	sfr, A	2	4	sfr ← A	
	A, !addr16	3	8	A ← (addr16)	
	!addr16, A	3	8	(addr16) ← A	
	PSW, #byte	3	6	PSW ← byte	× × ×
	A, PSW	2	4	A ← PSW	
	PSW, A	2	4	PSW ← A	× × ×
	A, [DE]	1	6	A ← (DE)	
	[DE], A	1	6	(DE) ← A	
	A, [HL]	1	6	A ← (HL)	
	[HL], A	1	6	(HL) ← A	
	A, [HL + byte]	2	6	A ← (HL + byte)	
	[HL + byte], A	2	6	(HL + byte) ← A	
XCH	A, X	1	4	A ↔ X	
	A, r ^{注 2}	2	6	A ↔ r	
	A, saddr	2	6	A ↔ (saddr)	
	A, sfr	2	6	A ↔ sfr	
	A, [DE]	1	8	A ↔ (DE)	
	A, [HL]	1	8	A ↔ (HL)	
	A, [HL, byte]	2	8	A ↔ (HL + byte)	

- 注 1. 除去 r = A
2. 除去 r = A、X

备注 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。

助记符	操作数	字节	时钟	操作	标志		
					Z	AC	CY
MOVW	rp, #word	3	6	$rp \leftarrow \text{word}$			
	AX, saddrp	2	6	$AX \leftarrow (\text{saddrp})$			
	saddrp, AX	2	8	$(\text{saddrp}) \leftarrow AX$			
	AX, rp 注	1	4	$AX \leftarrow rp$			
	rp, AX 注	1	4	$rp \leftarrow AX$			
XCHW	AX, rp 注	1	8	$AX \leftrightarrow rp$			
ADD	A, #byte	2	4	$A, CY \leftarrow A + \text{byte}$	x	x	x
	saddr, #byte	3	6	$(\text{saddr}), CY \leftarrow (\text{saddr}) + \text{byte}$	x	x	x
	A, r	2	4	$A, CY \leftarrow A + r$	x	x	x
	A, saddr	2	4	$A, CY \leftarrow A + (\text{saddr})$	x	x	x
	A, !addr16	3	8	$A, CY \leftarrow A + (\text{addr16})$	x	x	x
	A, [HL]	1	6	$A, CY \leftarrow A + (\text{HL})$	x	x	x
	A, [HL + byte]	2	6	$A, CY \leftarrow A + (\text{HL} + \text{byte})$	x	x	x
ADDC	A, #byte	2	4	$A, CY \leftarrow A + \text{byte} + CY$	x	x	x
	saddr, #byte	3	6	$(\text{saddr}), CY \leftarrow (\text{saddr}) + \text{byte} + CY$	x	x	x
	A, r	2	4	$A, CY \leftarrow A + r + CY$	x	x	x
	A, saddr	2	4	$A, CY \leftarrow A + (\text{saddr}) + CY$	x	x	x
	A, !addr16	3	8	$A, CY \leftarrow A + (\text{addr16}) + CY$	x	x	x
	A, [HL]	1	6	$A, CY \leftarrow A + (\text{HL}) + CY$	x	x	x
	A, [HL + byte]	2	6	$A, CY \leftarrow A + (\text{HL} + \text{byte}) + CY$	x	x	x
SUB	A, #byte	2	4	$A, CY \leftarrow A - \text{byte}$	x	x	x
	saddr, #byte	3	6	$(\text{saddr}), CY \leftarrow (\text{saddr}) - \text{byte}$	x	x	x
	A, r	2	4	$A, CY \leftarrow A - r$	x	x	x
	A, saddr	2	4	$A, CY \leftarrow A - (\text{saddr})$	x	x	x
	A, !addr16	3	8	$A, CY \leftarrow A - (\text{addr16})$	x	x	x
	A, [HL]	1	6	$A, CY \leftarrow A - (\text{HL})$	x	x	x
	A, [HL + byte]	2	6	$A, CY \leftarrow A - (\text{HL} + \text{byte})$	x	x	x

注 只有当 rp = BC, DE 或 HL

备注 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。

助记符	操作数	字节	时钟	操作	标志		
					Z	AC	CY
SUBC	A, #byte	2	4	$A, CY \leftarrow A - \text{byte} - CY$	×	×	×
	saddr, #byte	3	6	$(saddr), CY \leftarrow (saddr) - \text{byte} - CY$	×	×	×
	A, r	2	4	$A, CY \leftarrow A - r - CY$	×	×	×
	A, saddr	2	4	$A, CY \leftarrow A - (saddr) - CY$	×	×	×
	A, !addr16	3	8	$A, CY \leftarrow A - (\text{addr16}) - CY$	×	×	×
	A, [HL]	1	6	$A, CY \leftarrow A - (\text{HL}) - CY$	×	×	×
	A, [HL + byte]	2	6	$A, CY \leftarrow A - (\text{HL} + \text{byte}) - CY$	×	×	×
AND	A, #byte	2	4	$A \leftarrow A \wedge \text{byte}$	×		
	saddr, #byte	3	6	$(saddr) \leftarrow (saddr) \wedge \text{byte}$	×		
	A, r	2	4	$A \leftarrow A \wedge r$	×		
	A, saddr	2	4	$A \leftarrow A \wedge (saddr)$	×		
	A, !addr16	3	8	$A \leftarrow A \wedge (\text{addr16})$	×		
	A, [HL]	1	6	$A \leftarrow A \wedge (\text{HL})$	×		
	A, [HL + byte]	2	6	$A \leftarrow A \wedge (\text{HL} + \text{byte})$	×		
OR	A, #byte	2	4	$A \leftarrow A \vee \text{byte}$	×		
	saddr, #byte	3	6	$(saddr) \leftarrow (saddr) \vee \text{byte}$	×		
	A, r	2	4	$A \leftarrow A \vee r$	×		
	A, saddr	2	4	$A \leftarrow A \vee (saddr)$	×		
	A, !addr16	3	8	$A \leftarrow A \vee (\text{addr16})$	×		
	A, [HL]	1	6	$A \leftarrow A \vee (\text{HL})$	×		
	A, [HL + byte]	2	6	$A \leftarrow A \vee (\text{HL} + \text{byte})$	×		
XOR	A, #byte	2	4	$A \leftarrow A \oplus \text{byte}$	×		
	saddr, #byte	3	6	$(saddr) \leftarrow (saddr) \oplus \text{byte}$	×		
	A, r	2	4	$A \leftarrow A \oplus r$	×		
	A, saddr	2	4	$A \leftarrow A \oplus (saddr)$	×		
	A, !addr16	3	8	$A \leftarrow A \oplus (\text{addr16})$	×		
	A, [HL]	1	6	$A \leftarrow A \oplus (\text{HL})$	×		
	A, [HL + byte]	2	6	$A \leftarrow A \oplus (\text{HL} + \text{byte})$	×		

备注 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。

助记符	操作数	字节	时钟	操作	标志		
					Z	AC	CY
CMP	A, #byte	2	4	A - byte	x	x	x
	saddr, #byte	3	6	(saddr) - byte	x	x	x
	A, r	2	4	A - r	x	x	x
	A, saddr	2	4	A - (saddr)	x	x	x
	A, !addr16	3	8	A - (addr16)	x	x	x
	A, [HL]	1	6	A - (HL)	x	x	x
	A, [HL + byte]	2	6	A - (HL + byte)	x	x	x
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
INC	r	2	4	r ← r + 1	x	x	
	saddr	2	4	(saddr) ← (saddr) + 1	x	x	
DEC	r	2	4	r ← r - 1	x	x	
	saddr	2	4	(saddr) ← (saddr) - 1	x	x	
INCW	rp	1	4	rp ← rp + 1			
DECW	rp	1	4	rp ← rp - 1			
ROR	A, 1	1	2	(CY, A ₇ ← A ₀ , A _{m-1} ← A _m) × 1			x
ROL	A, 1	1	2	(CY, A ₀ ← A ₇ , A _{m+1} ← A _m) × 1			x
RORC	A, 1	1	2	(CY ← A ₀ , A ₇ ← CY, A _{m-1} ← A _m) × 1			x
ROLC	A, 1	1	2	(CY ← A ₇ , A ₀ ← CY, A _{m+1} ← A _m) × 1			x
SET1	saddr.bit	3	6	(saddr.bit) ← 1			
	sfr.bit	3	6	sfr.bit ← 1			
	A.bit	2	4	A.bit ← 1			
	PSW.bit	3	6	PSW.bit ← 1	x	x	x
	[HL].bit	2	10	(HL).bit ← 1			
CLR1	saddr.bit	3	6	(saddr.bit) ← 0			
	sfr.bit	3	6	sfr.bit ← 0			
	A.bit	2	4	A.bit ← 0			
	PSW.bit	3	6	PSW.bit ← 0	x	x	x
	[HL].bit	2	10	(HL).bit ← 0			
SET1	CY	1	2	CY ← 1			1
CLR1	CY	1	2	CY ← 0			0
NOT1	CY	1	2	CY ← \overline{CY}			x

备注 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。

助记符	操作数	字节	时钟	操作	标志		
					Z	AC	CY
CALL	!addr16	3	6	$(SP - 1) \leftarrow (PC + 3)_H, (SP - 2) \leftarrow (PC + 3)_L,$ $PC \leftarrow \text{addr16}, SP \leftarrow SP - 2$			
CALLT	[addr5]	1	8	$(SP - 1) \leftarrow (PC + 1)_H, (SP - 2) \leftarrow (PC + 1)_L,$ $PC_H \leftarrow (00000000, \text{addr5} + 1),$ $PC_L \leftarrow (00000000, \text{addr5}), SP \leftarrow SP - 2$			
RET		1	6	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP), SP \leftarrow SP + 2$			
RETI		1	8	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $PSW \leftarrow (SP + 2), SP \leftarrow SP + 3, NMIS \leftarrow 0$	R	R	R
PUSH	PSW	1	2	$(SP - 1) \leftarrow PSW, SP \leftarrow SP - 1$			
	rp	1	4	$(SP - 1) \leftarrow rp_H, (SP - 2) \leftarrow rp_L, SP \leftarrow SP - 2$			
POP	PSW	1	4	$PSW \leftarrow (SP), SP \leftarrow SP + 1$	R	R	R
	rp	1	6	$rp_H \leftarrow (SP + 1), rp_L \leftarrow (SP), SP \leftarrow SP + 2$			
MOVW	SP, AX	2	8	$SP \leftarrow AX$			
	AX, SP	2	6	$AX \leftarrow SP$			
BR	!addr16	3	6	$PC \leftarrow \text{addr16}$			
	\$addr16	2	6	$PC \leftarrow PC + 2 + \text{jdisp8}$			
	AX	1	6	$PC_H \leftarrow A, PC_L \leftarrow X$			
BC	\$saddr16	2	6	$PC \leftarrow PC + 2 + \text{jdisp8}$ 如果 $CY = 1$			
BNC	\$saddr16	2	6	$PC \leftarrow PC + 2 + \text{jdisp8}$ 如果 $CY = 0$			
BZ	\$saddr16	2	6	$PC \leftarrow PC + 2 + \text{jdisp8}$ 如果 $Z = 1$			
BNZ	\$saddr16	2	6	$PC \leftarrow PC + 2 + \text{jdisp8}$ 如果 $Z = 0$			
BT	saddr.bit, \$addr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ 如果 $(\text{saddr.bit}) = 1$			
	sfr.bit, \$addr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ 如果 $\text{sfr.bit} = 1$			
	A.bit, \$addr16	3	8	$PC \leftarrow PC + 3 + \text{jdisp8}$ 如果 $A.bit = 1$			
	PSW.bit, \$addr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ 如果 $PSW.bit = 1$			
BF	saddr.bit, \$addr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ 如果 $(\text{saddr.bit}) = 0$			
	sfr.bit, \$addr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ 如果 $\text{sfr.bit} = 0$			
	A.bit, \$addr16	3	8	$PC \leftarrow PC + 3 + \text{jdisp8}$ 如果 $A.bit = 0$			
	PSW.bit, \$addr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ 如果 $PSW.bit = 0$			
DBNZ	B, \$addr16	2	6	$B \leftarrow B - 1,$ 然后 $PC \leftarrow PC + 2 + \text{jdisp8}$ 如果 $B \neq 0$			
	C, \$addr16	2	6	$C \leftarrow C - 1,$ 然后 $PC \leftarrow PC + 2 + \text{jdisp8}$ 如果 $C \neq 0$			
	saddr, \$addr16	3	8	$(\text{saddr}) \leftarrow (\text{saddr}) - 1,$ 然后 $PC \leftarrow PC + 3 + \text{jdisp8}$ 如果 $(\text{saddr}) \neq 0$			
NOP		1	2	不操作			
EI		3	6	$IE \leftarrow 1$ (允许中断)			
DI		3	6	$IE \leftarrow 0$ (禁止中断)			
HALT		1	2	设置 HALT 模式			
STOP		1	2	设置 STOP 模式			

备注 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。

16.3 按寻址类型列出指令

(1) 8 位指令

MOV、XCH、ADD、ADDC、SUB、SUBC、AND、OR、XOR、CMP、INC、DEC、ROR、ROL、RORC、
 ROLC、PUSH、POP、DBNZ

第二操作数 第一操作数	#byte	A	r	sfr	saddr	laddr16	PSW	[DE]	[HL]	[HL + byte]	\$addr16	1	无
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV [#] XCH [#]	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											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											
[HL + byte]		MOV											

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 [#]	saddrp	SP	无
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	
rp	MOVW	MOVW [#]				INCW DECW PUSH POP
saddrp		MOVW				
sp		MOVW				

注 只有当 rp = BC, DE 或 HL

(3) 位操作指令

SET1、CLR1、NOT1、BT、BF

第二操作数 / 第一操作数	\$addr16	无
A.bit	BT BF	SET1 CLR1
sfr.bit	BT BF	SET1 CLR1
saddr.bit	BT BF	SET1 CLR1
PSW.bit	BT BF	SET1 CLR1
[HL].bit		SET1 CLR1
CY		SET1 CLR1 NOT1

(4) 调用指令 / 跳转指令

CALL、CALLT、BR、BC、BNC、BZ、BNZ、DBNZ

第二操作数 第一操作数	AX	!addr16	[addr5]	\$addr16
基本指令	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
组合指令				DBNZ

(5) 其它指令

RET、RETI、NOP、EI、DI、HALT、STOP

第十七章 电气特性（目标）

注意事项 这些特性显示了产品的目标值，目标值在设备评估后可能会发生变化。工作电压范围可能也会改变。

最大绝对额定值 ($T_A = 25^\circ\text{C}$)

参数	符号	条件	额定值	单位
电源电压	V_{DD}		$-0.3 \sim +6.5$	V
	V_{SS}		$-0.3 \sim +0.3$	V
输入电压	V_I	P20~P23, P32, P34, P40~P47	$-0.3 \sim V_{DD} + 0.3^{\#}$	V
输出电压	V_O		$-0.3 \sim V_{DD} + 0.3^{\#}$	V
输出电流, 高	I_{OH}	每个引脚	-10.0	mA
		P20~P23, P32, P34, P40~P47 的所有引脚	-44.0	mA
输出电流, 低	I_{OL}	每个引脚	20.0	mA
		P20~P23, P32, P34, P40~P47 的所有引脚	44.0	mA
工作环境温度	T_A	在正常工作模式中	$-40 \sim +85$	$^\circ\text{C}$
		Flash 存储器编程期间		$^\circ\text{C}$
存储温度	T_{stg}	Flash 存储器空白状态	$-65 \sim +150$	$^\circ\text{C}$
		Flash 存储器编程已经完成	$-40 \sim +125$	$^\circ\text{C}$

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。

X1 振荡器特性 ($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}$ ^{注1}, $V_{SS} = 0 \text{ V}$)

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_x) 注2		1		10.0	MHz
晶体谐振器		振荡器频率 (f_x) 注2		1		10.0	MHz
外部时钟		X1 输入频率 (f_x) 注2	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	1		10.0	MHz
			$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	1		5.0	
		X1 输入高-/低-电平宽度 (t_{XH} , t_{XL})	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0.045		0.5	s
			$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	0.09		0.5	

- 注 1. 使用这个产品电压范围 2.2~5.5V 因为上电清零 (POC) 电路的检测电压 (V_{POC}) 是 $2.1 \text{ V} \pm 0.1 \text{ V}$ 。
2. 只显示振荡器特性。关于指令执行时间请参阅 **AC 特性**。

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

- 连接线越短越好。
- 连接线不应与其他信号线交叉。
- 流经的电流变化较大的信号线不要在振荡器周围布线。
- 要保持振荡器电容器的接地点电压与 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。

备注 对于谐振器的选择和振荡器常量，要求用户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。

高速内部振荡器特性 ($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}^{\text{注}1}$, $V_{SS} = 0 \text{ V}$)

谐振器	参数	条件	最小值.	典型值.	最大值.	单位
高速内部振荡器	振荡器频率 ($f_x = 8 \text{ MHz}^{\text{注}2}$)	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			± 3	%
		$T_A = -10 \sim +70^\circ\text{C}$				
		$T_A = -40 \sim +85^\circ\text{C}$			± 5	%
	振荡器频率 (f_x) ^{注2}	$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	5.5			MHz

- 注 1. 使用这个产品电压范围 2.2~5.5V 因为上电清零 (POC) 电路的检测电压 (V_{POC}) 是 $2.1 \text{ V} \pm 0.1 \text{ V}$ 。
 2. 只显示振荡器特性。关于指令执行时间请参照 **AC 特性**。

低速内部振荡器特性 ($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}^{\text{注}}$, $V_{SS} = 0 \text{ V}$)

谐振器	参数	条件	最小值.	典型值.	最大值.	单位
低速内部振荡器	振荡器频率 (f_{RL})		120	240	480	kHz

注 使用这个产品电压范围 2.2~5.5V 因为上电清零 (POC) 电路的检测电压 (V_{POC}) 是 $2.1 \text{ V} \pm 0.1 \text{ V}$ 。

DC 特性 ($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}^\ddagger$, $V_{SS} = 0 \text{ V}$) (1/2)

参数	符号	条件		最小值.	典型值.	最大值.	单位
输出电流, 高	I _{OH}	每个引脚	$2.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			-5	mA
		所有引脚总和	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			-25	mA
			$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$			-15	mA
输出电流, 低	I _{OL}	每个引脚	$2.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			10	mA
		所有引脚总和	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			30	mA
			$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$			15	mA
输入电压, 高	V _{IH1}	外部时钟模式下 P23 和 P20, P21 之外的其它引脚		0.8V _{DD}		V _{DD}	V
	V _{IH2}	非外部时钟模式下 P23 和 P20, P21		0.7V _{DD}		V _{DD}	V
输入电压, 低	V _{IL1}	外部时钟模式下 P23 和 P20, P21 之外的其它引脚		0		0.2V _{DD}	V
	V _{IL2}	非外部时钟模式下 P23 和 P20, P21		0		0.3V _{DD}	V
输出电压, 高	V _{OH}	所有引脚总和 I _{OH} = -15 mA	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$ I _{OH} = -5 mA	V _{DD} - 1.0			V
		I _{OH} = -100 A	$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	V _{DD} - .5			V
输出电压, 低	V _{OL}	所有输出引脚之和 I _{OL} = 30 mA	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$ I _{OL} = 10 mA			1.3	V
		$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$ I _{OL} = 400 A				0.4	V
输入泄漏电流, 高	I _{LIH}	V _I = V _{DD}	X1 之外的引脚			1	A
输入泄漏电流, 低	I _{LIL}	V _I = 0 V	X1 之外的引脚			-1	A
输出峰值电流, 高	I _{LOH}	V _O = V _{DD}	X2 之外的引脚			1	A
输出峰值电流, 低	I _{LOL}	V _O = 0 V	X2 之外的引脚			-1	A
上拉电阻值	R _{PU}	V _I = 0 V		10	30	100	kΩ
下拉电阻值	R _{PD}	P22, P23, 复位状态		10	30	100	kΩ

注 使用这个产品电压范围 2.2~5.5V 因为上电清零 (POC) 电路的检测电压 (V_{POC}) 是 2.1 V ±0.1 V。

备注 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

DC 特性 ($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}^{\#1}$, $V_{SS} = 0 \text{ V}$) (2/2)

参数	符号	条件		最小值	典型值	最大值	单位	
供电电流 ^{注2}	I _{DD1} ^{注3}	晶体振荡器, 外部时钟输入振荡操作模式 ^{注6}	f _{XP} = 10 MHz V _{DD} = 5.0 V ±10% ^{注4}		6.1	12.2	mA	
			f _{XP} = 6 MHz V _{DD} = 5.0 V ±10% ^{注4}		5.5	11.0		
			f _X = 5 MHz V _{DD} = 3.0 V ±10% ^{注5}		3.0	6.0		
	I _{DD2}	晶体振荡器, 外部时钟输入 HALT 模式 ^{注6}	f _{XP} = 10 MHz V _{DD} = 5.0 V ±10% ^{注4}	当周边功能停止		1.7	3.8	mA
				当周边功能工作			6.7	
			f _{XP} = 6 MHz V _{DD} = 5.0 V ±10% ^{注4}	当周边功能停止		1.3	3.0	mA
				当周边功能工作			6.0	
			f _X = 5 MHz V _{DD} = 3.0 V ±10% ^{注5}	当周边功能停止		0.48	1	mA
				当周边功能工作			2.1	
	I _{DD3} ^{注3}	高速内部振荡工作模式 ^{注7}	f _{XP} = 8 MHz V _{DD} = 5.0 V ±10% ^{注4}			5.0	10.0	mA
	I _{DD4}	高速内部振荡 HALT 模式 ^{注7}	f _{XP} = 8 MHz V _{DD} = 5.0 V ±10% ^{注4}	当周边功能停止		1.4	3.2	
				当周边功能工作			5.9	
I _{DD5}	STOP 模式	V _{DD} = 5.0 V ±10%	当低速内部振荡停止		3.5	20.0	A	
			当低速内部振荡工作		17.5	32.0		
		V _{DD} = 3.0 V ±10%	当低速内部振荡停止		3.5	15.5	A	
			当低速内部振荡工作		11.0	26.0		

- 注
1. 使用这个产品电压范围 2.2~5.5V 因为上电清零 (POC) 电路的检测电压 (V_{POC}) 是 2.1 V ±0.1 V。
 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. 这些电流包括周边工作电流。
 4. 当处理器时钟控制寄存器 (PCC) 被设为 00H。
 5. 当处理器时钟控制寄存器 (PCC) 被设为 02H。
 6. 当晶体振荡器时钟, 外部时钟输入通过选项字节选作系统时钟源。
 7. 当高速内部振荡时钟通过选项字节选作系统时钟源。

AC 特性

基本操作 ($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}^{\#1}$, $V_{SS} = 0 \text{ V}$)

参数	符号	条件	最小值.	典型值.	最大值.	单位	
周期时间 (最小指令执行时间)	T _{CY}	晶体振荡时钟, 外部时钟输入	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0.2		16	s
			$3.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	0.33		16	s
			$2.7 \text{ V} \leq V_{DD} < 3.0 \text{ V}$	0.4		16	s
			$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	1		16	s
	高速内部振荡时钟	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0.23		4.22	s	
		$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	0.47		4.22	s	
$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$		0.95		4.22	s		
TI000 输入高电平宽度, 低电平宽度	t _{TIH} , t _{TIL}	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$2/f_{\text{sam}} + 0.1^{\#2}$			s	
		$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	$2/f_{\text{sam}} + 0.2^{\#2}$			s	
中断输入高电平宽度, 低电平宽度	t _{INTH} , t _{INTL}		1			s	
RESET 输入低电平宽度	t _{RSL}		2			s	

注 1. 使用这个产品电压范围 2.2~5.5V 因为上电清零 (POC) 电路的检测电压 (V_{POC}) 是 $2.1 \text{ V} \pm 0.1 \text{ V}$ 。

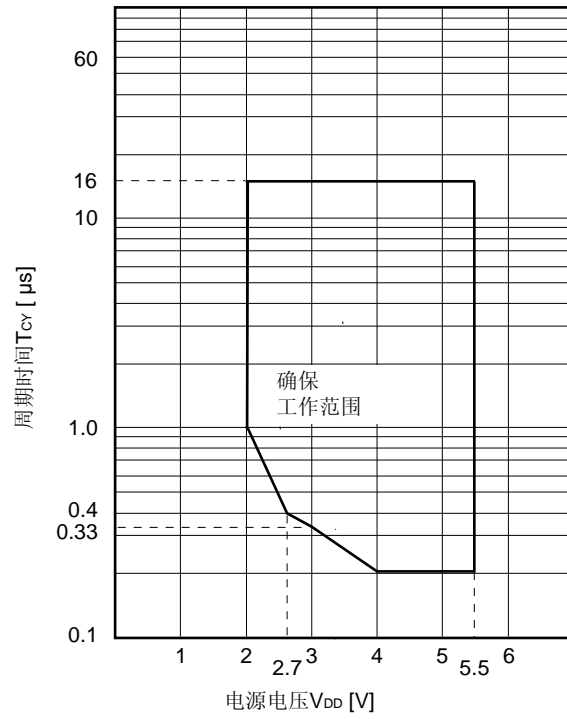
2. 可以使用预分频模式寄存器 00 (PRM00) 的 0 位和第 1 位 (PRM000, PRM001) 来选择 $f_{\text{sam}} = f_{\text{XP}}$, $f_{\text{XP}}/4$, 或 $f_{\text{XP}}/256$ 。将 TI000 有效沿选作计数时钟时需注意 $f_{\text{sam}} = f_{\text{XP}}$ 。

CPU 时钟频率, 周边时钟频率

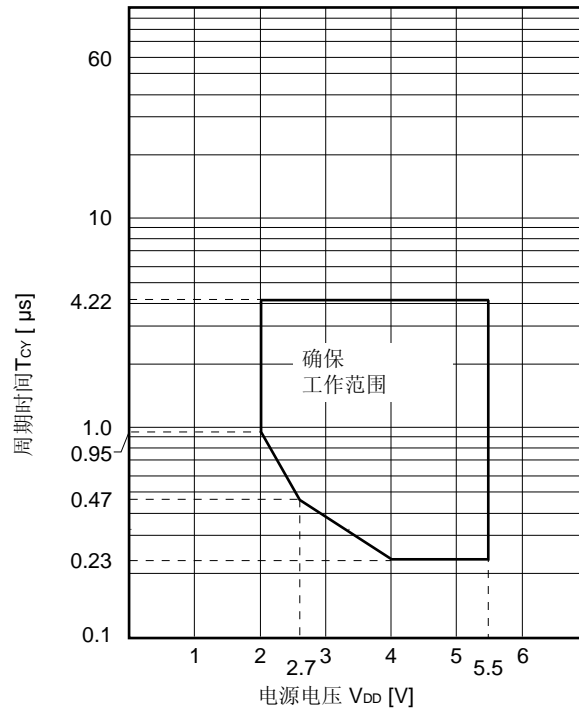
参数	条件	CPU 时钟 (f_{CPU})	周边时钟 (f_{XP})
陶瓷谐振器 晶体谐振器 外部时钟	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$125 \text{ kHz} \leq f_{\text{CPU}} \leq 10 \text{ MHz}$	$125 \text{ kHz} \leq f_{\text{XP}} \leq 10 \text{ MHz}$
	$3.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	$125 \text{ kHz} \leq f_{\text{CPU}} \leq 6 \text{ MHz}$	
	$2.7 \text{ V} \leq V_{DD} < 3.0 \text{ V}$	$125 \text{ kHz} \leq f_{\text{CPU}} \leq 5 \text{ MHz}$	
	$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}^{\#}$	$125 \text{ kHz} \leq f_{\text{CPU}} \leq 2 \text{ MHz}$	$125 \text{ kHz} \leq f_{\text{XP}} \leq 5 \text{ MHz}$
高速内部振荡器	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	500 kHz (典型值) $\leq f_{\text{CPU}} \leq 8 \text{ MHz}$ (典型值)	500 kHz (典型值) $\leq f_{\text{XP}} \leq 8 \text{ MHz}$ (典型值)
	$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	500 kHz (典型值) $\leq f_{\text{CPU}} \leq 4 \text{ MHz}$ (典型值)	
	$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}^{\#}$	500 kHz (典型值) $\leq f_{\text{CPU}} \leq 2 \text{ MHz}$ (典型值)	500 kHz (典型值) $\leq f_{\text{XP}} \leq 4 \text{ MHz}$ (典型值)

注 使用这个产品电压范围 2.2~5.5V 因为上电清零 (POC) 电路的检测电压 (V_{POC}) 是 $2.1 \text{ V} \pm 0.1 \text{ V}$ 。

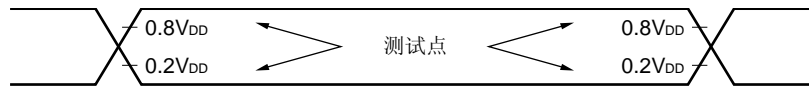
T_{CY} vs. V_{DD} (晶体/陶瓷振荡时钟, 外部时钟输入)



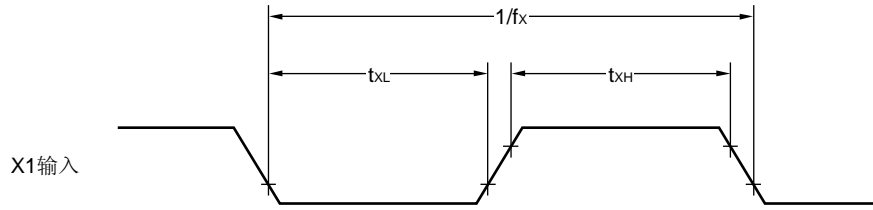
T_{CY} vs. V_{DD} (高速内部振荡器时钟)



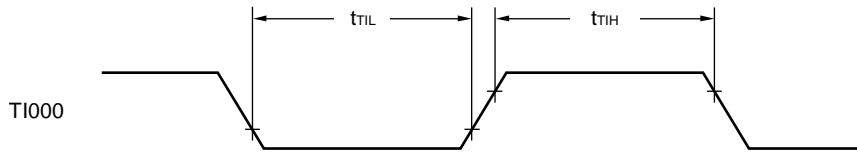
AC 时序测试点 (排除 X1 输入)



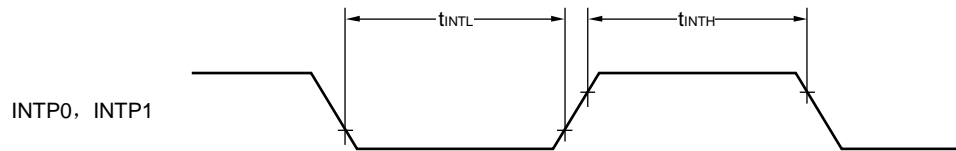
时钟时序



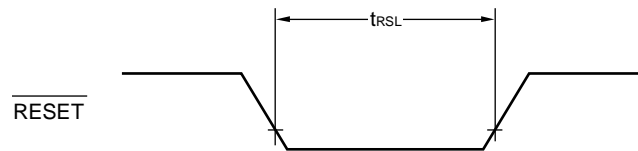
TI000 时序



中断输入时序



$\overline{\text{RESET}}$ 输入时序



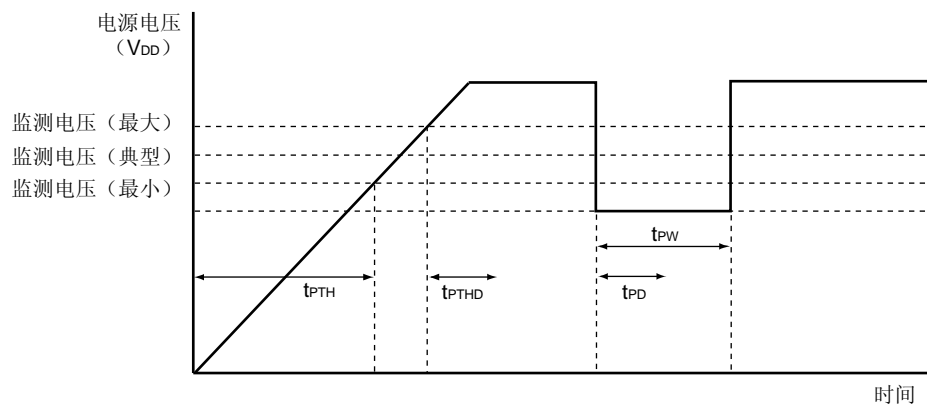
POC 电路特性 ($T_A = -40 \sim +85^\circ\text{C}$)

参数	符号	条件	最小值.	典型值.	最大值.	单位
检测电压	V_{POC}		2.0	2.1	2.2	V
电源上升时间	t_{PTH}	$V_{DD}: 0\text{ V} \rightarrow 2.1\text{ V}$	1.5			s
响应延迟时间 1 ^{注1}	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2}	t_{PD}	当电源电压下降			1.0	ms
最小脉冲宽度	t_{PW}		0.2			ms

注 1. 从电压检测到内部复位释放要求的时间。

2. 从电压检测到内部复位信号产生要求的时间。

POC 电路时序



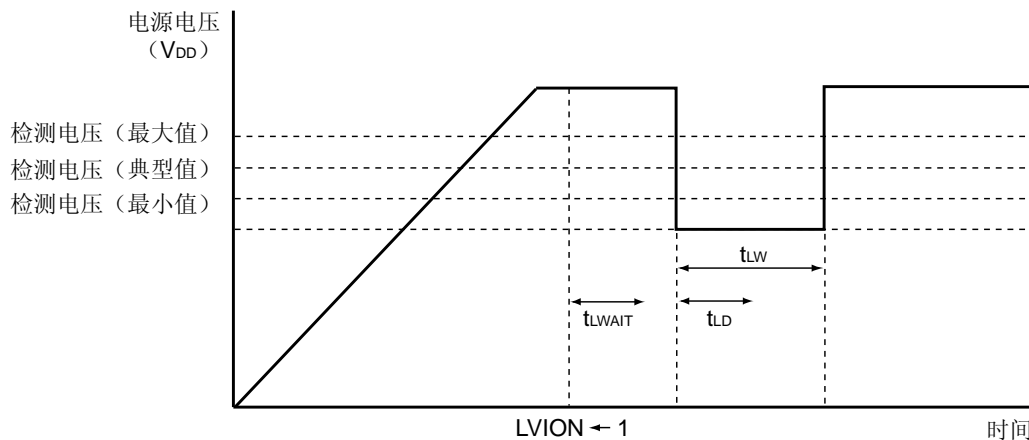
LVI 电路特性 ($T_A = -40 \sim +85^\circ\text{C}$)

参数	符号	条件	最小值.	典型值.	最大值.	单位
检测电压	V_{LV10}		4.1	4.3	4.5	V
	V_{LV11}		3.9	4.1	4.3	V
	V_{LV12}		3.7	3.9	4.1	V
	V_{LV13}		3.5	3.7	3.9	V
	V_{LV14}		3.3	3.5	3.7	V
	V_{LV15}		3.15	3.3	3.45	V
	V_{LV16}		2.95	3.1	3.25	V
	V_{LV17}		2.7	2.85	3.0	V
	V_{LV18}		2.5	2.6	2.7	V
	V_{LV19}		2.25	2.35	2.45	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_{LWAIT}			0.1	0.2	ms

- 注 1. 从电压检测到中断输出或内部复位信号产生的要求时间。
 2. 设置 LVION 为 1 到工作稳定要求的时间。

- 备注 1. $V_{LV10} > V_{LV11} > V_{LV12} > V_{LV13} > V_{LV14} > V_{LV15} > V_{LV16} > V_{LV17} > V_{LV18} > V_{LV19}$
 2. $V_{POC} < V_{LV1m}$ ($m = 0 \sim 9$)

LVI 电路时序

数据寄存器 STOP 模式低电源电压数据保持特性 ($T_A = -40 \sim +85^\circ\text{C}$)

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 ($T_A = -40 \sim +85^\circ\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$)

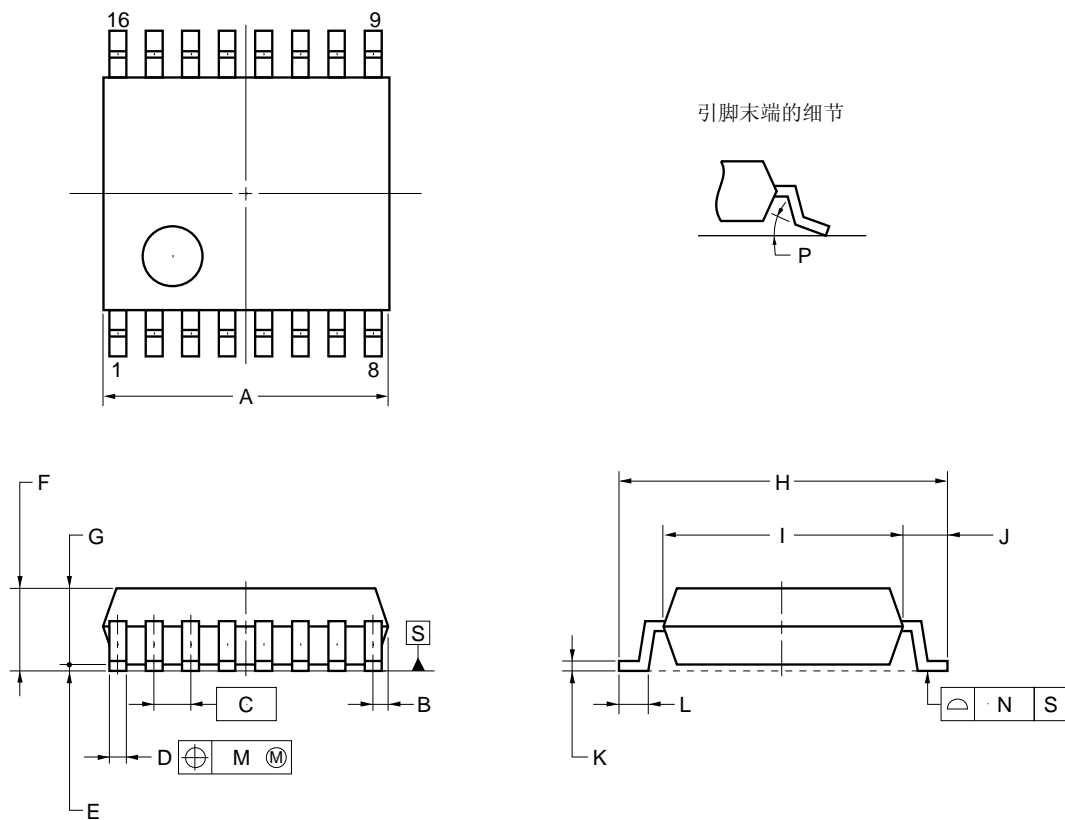
参数	符号	条件	最小值	典型值	最大值	单位
供电电流	I_{DD}	$V_{DD} = 5.5\text{ V}$			7.0	mA
擦除计数 ^注 (每块)	N_{ERASE}	$T_A = -40 \sim +85^\circ\text{C}$	1000			次
片擦除时间	T_{CERASE}	$T_A = -10 \sim +85^\circ\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0.8	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		1.0	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		1.2	s
		$T_A = -10 \sim +85^\circ\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		4.8	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		5.2	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		6.1	s
		$T_A = -40 \sim +85^\circ\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		1.6	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		1.8	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		2.0	s
		$T_A = -40 \sim +85^\circ\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		9.1	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		10.1	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		12.3	s
块擦除时间	T_{BERASE}	$T_A = -10 \sim +85^\circ\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0.4	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		0.5	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		0.6	s
		$T_A = -10 \sim +85^\circ\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		2.6	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		2.8	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		3.3	s
		$T_A = -40 \sim +85^\circ\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0.9	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		1.0	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		1.1	s
		$T_A = -40 \sim +85^\circ\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		4.9	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		5.4	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		6.6	s
字节写入时间	T_{WRITE}	$T_A = -40 \sim +85^\circ\text{C}$, $N_{ERASE} \leq 1000$			150	μs
内部校验	T_{VERIFY}	每 1 块			6.8	ms
		每 1 字节			27	μs
空白检查	T_{BLKCHK}	每 1 块			480	μs
保持年数		$T_A = 85^\circ\text{C}$ ^{注2} , $N_{ERASE} \leq 1000$	10			年

注 1. 擦除时间因擦除次数 (N_{ERASE}) 而不同参考芯片擦除次数和块擦除次数参数。
2. 当工作和不工作的平均温度是 85°C 。

备注 当产品在发货后第一次被写入时，“擦除 → 写入”和“只写入”都被认为一次重写。

第十八章 封装图

16 引脚塑封SSOP (5.72 mm (225))



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0.10mm(T.P.)以内。

项目	毫米
A	5.2±0.3
B	0.475 最大值
C	0.65 (T.P.)
D	0.22±0.08
E	0.125±0.075
F	1.565±0.235
G	1.44
H	6.2±0.3
I	4.4±0.2
J	0.9±0.2
K	0.17 ^{+0.08} _{-0.07}
L	0.5±0.2
M	0.10
N	0.10
P	5°±5°

P16GM-65-225B-5

第十九章 推荐焊接条件

这些产品应该在以下推荐条件下进行焊接和安装。
技术信息请参见以下网站。

半导体设备安装手册 (<http://www.necel.com/pkg/en/mount/index.html>)

- 注意事项 1.** 产品代号最后带-A 的产品是无铅产品。
2. 对于与以下所推荐不同的焊接方法和条件，请与日电电子的销售代表联系。

表 19-1. 表面安装类型焊接条件

16 引脚塑封 SSOP (无铅产品)

μ PD78F9510GR-JJG-A^{註1}, 78F9511GR-JJG-A^{註1}, 78F9512GR-JJG-A^{註1}

焊接方法	焊接条件	推荐条件符号
红外回流	封装峰值温度: 260°C, 时间: 最大 60 秒 (在 220°C 或者更高), 次数: 3 次或更少, 暴露限制: 7 天 ^{註2} (之后在 125°C 预烘 20 小时 到 72 小时)	IR60-207-3
波形焊接	关于详细情况, 请与日电电子的销售代表联系	-
局部加热	引脚温度: 最大 350°C, 时间: 最大 3 秒 (每行引脚)	-

- 注**
1. 处于开发中
 2. 打开干燥的包装后, 存储温度低于 25°C, 相对湿度 65% 条件下, 或减少存储时间。

注意事项 不要使用不同的焊接方法 (局部加热除外)。

附录 A 开发工具

下列开发工具用于使用 μ PD78F9510、78F9511、78F9512 的系统的开发。图 A-1 表示了开发工具。

- 与 PC98-NX 系列兼容

IBM PC/AT™ 支持的产品和兼容的可以用于 PC98-NX 系列。当使用 PC98-NX 系列计算机时，参阅 IBM PC/AT 兼容机的使用说明。

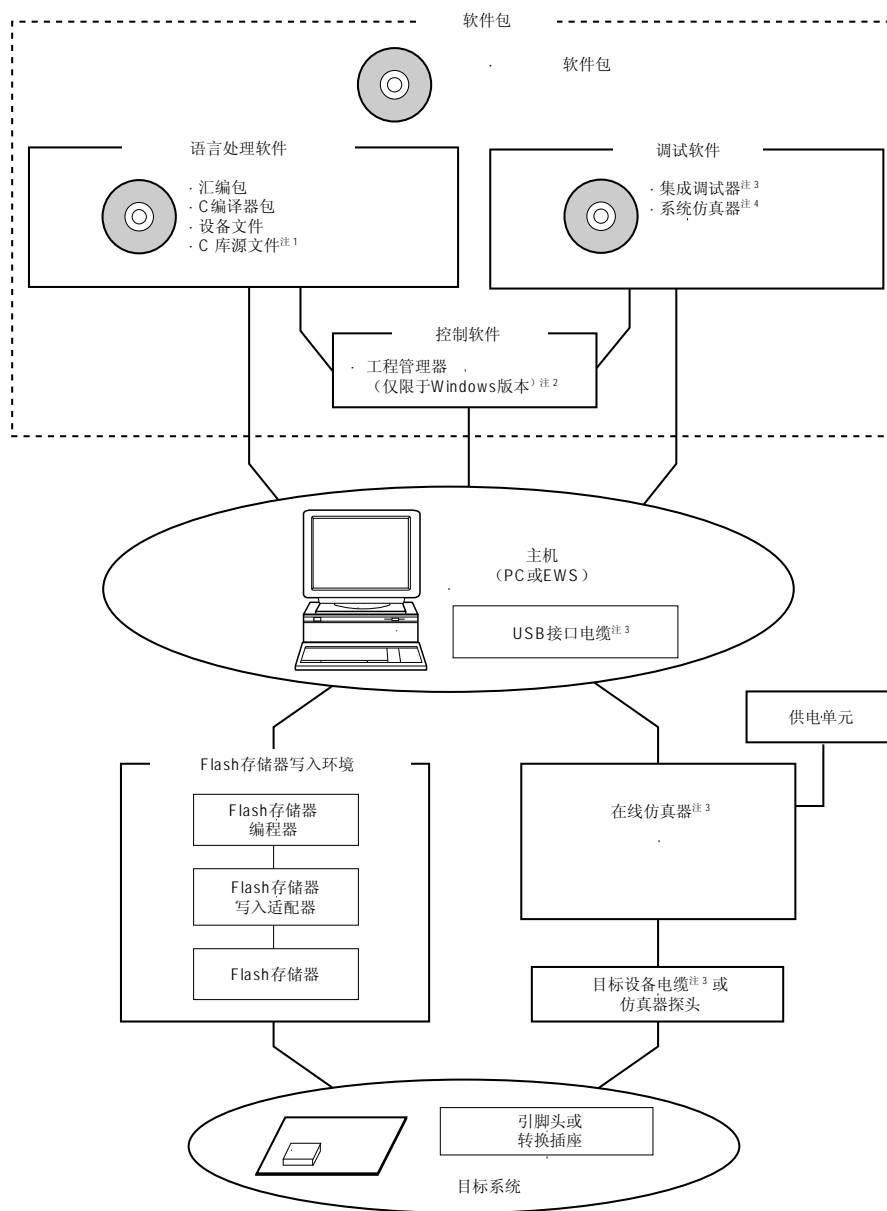
- Windows™

除另有说明外，“Windows”是指如下操作系统。

- Windows 98
- Windows NT™ 版 4.0
- Windows 2000
- Windows XP™

图 A-1. 开发工具 (1 / 4)

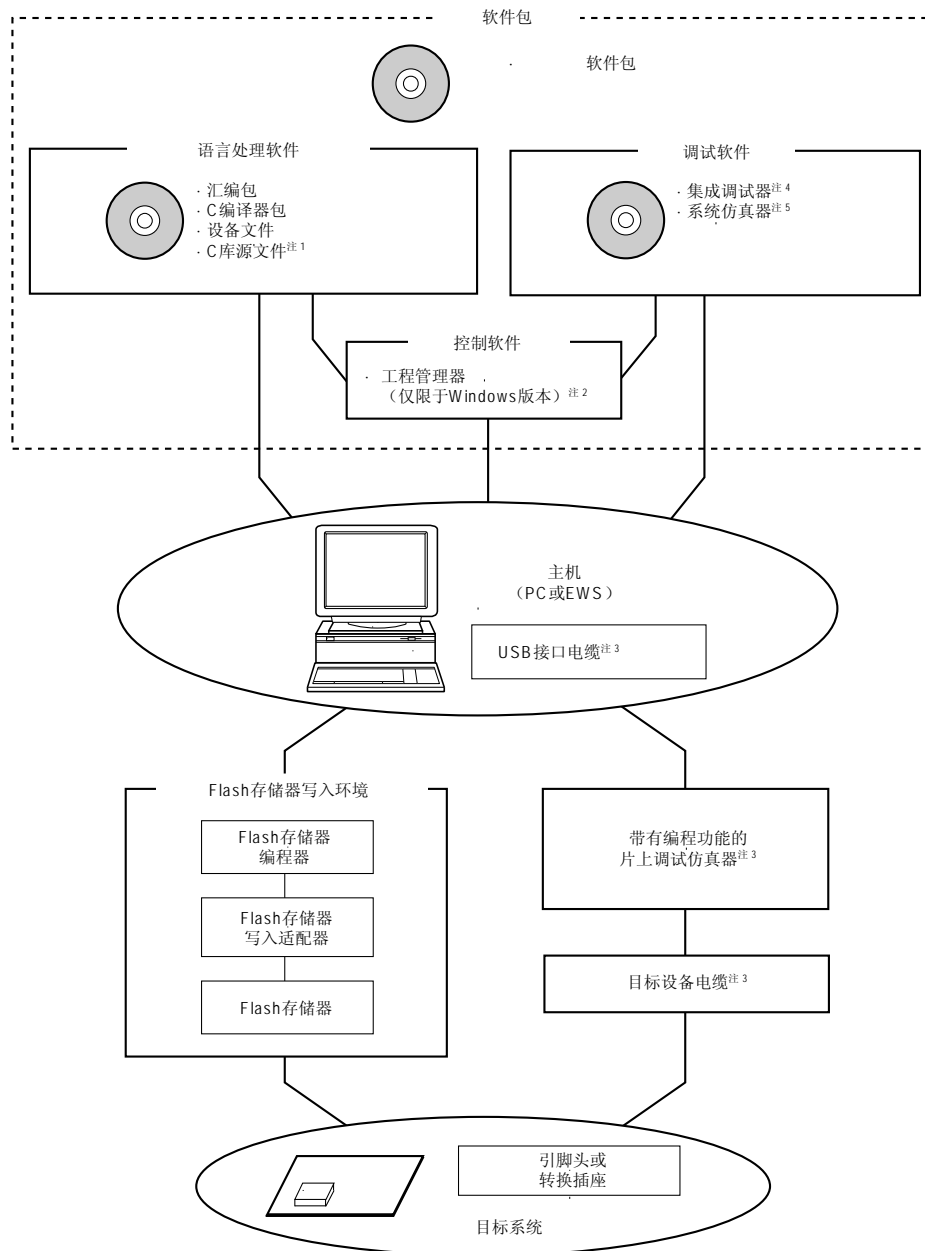
(1) 当使用在线仿真器 QB-78K0SKX1



- 注
1. C 语言库源文件不包含在软件包中。
 2. 项目管理器 PM+ 包含在汇编器包中。
PM+ 仅用于 Windows 环境。
 3. 随集成调试器 ID78K0S-QB 提供的在线仿真器 QB-78K0SKX1，带有编程功能的片上调试仿真器 QB-MINI2，一根 USB 接口电缆，一个供电单元和一个目标设备电缆。其它产品可选。
 4. 计划使用 SM+ 作为 78K0S/Kx1+ 的系统仿真器。

图 A-1. 开发工具 (2 / 4)

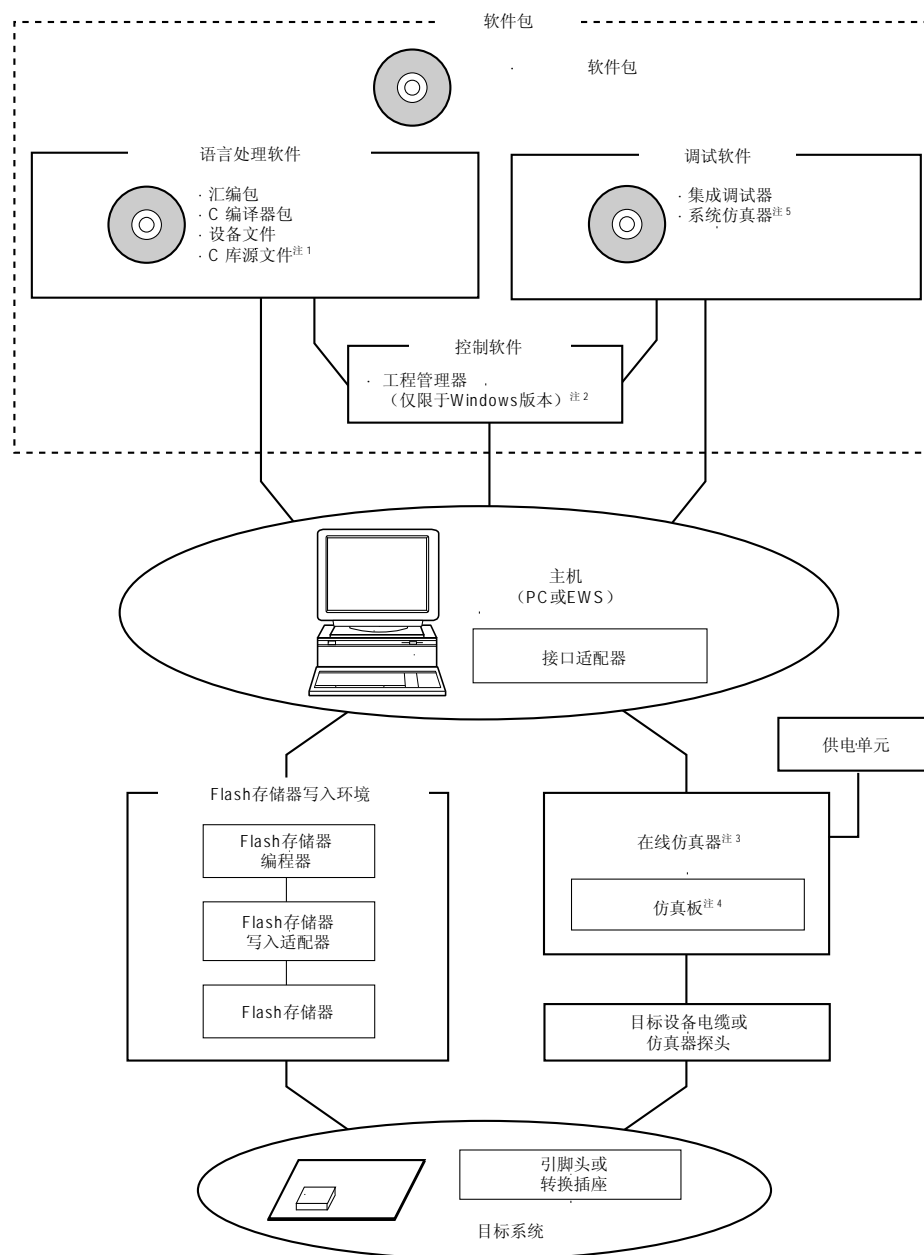
(2) 使用带有编程功能的片上调试仿真器 QB-MINI2



- 注
1. C语言库源文件不包含在软件包中。
 2. 项目管理器 PM+包含在汇编器包中。
PM+仅用于 Windows 环境。
 3. 带有编程功能的片上调试仿真器 QB-MINI2 带有一根 USB 接口电缆和一根目标设备电缆。
 4. 集成调试器 ID78K0S-QB 不包括在 QB-MINI2 中。
集成调试器 ID78K0S-QB 可以从下列网站获取。
<http://www.necel.com/micro/ods/eng/>
 5. 计划使用 SM+作为 78K0S/Kx1+的系统仿真器。

图 A-1. 开发工具 (3 / 4)

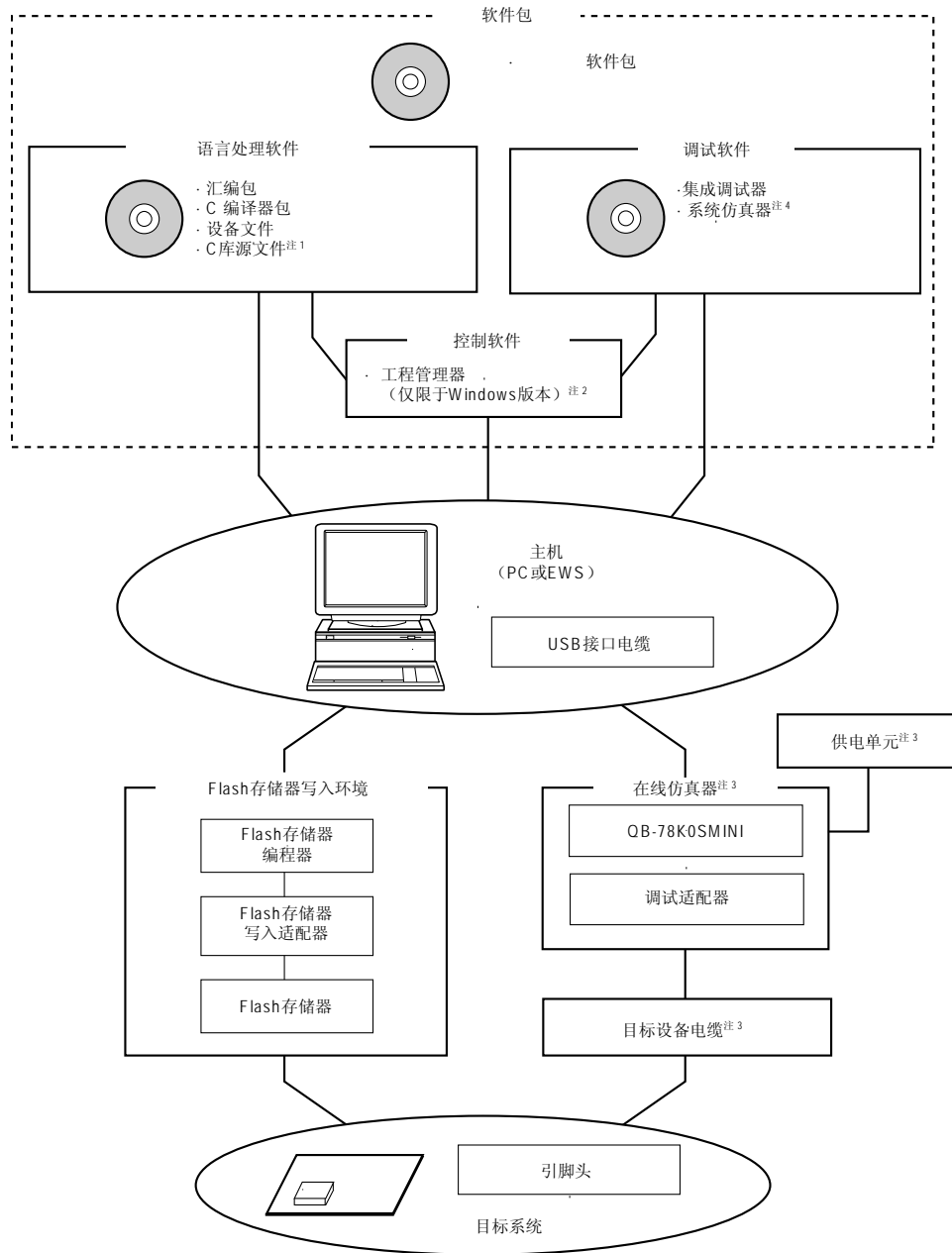
(3) 当使用在线仿真器 IE-78K0S-NS 或 IE-78K0S-NS-A



- 注
1. C 语言库源文件不包含在软件包中。
 2. 项目管理器 PM+包含在汇编器包中。
PM+仅用于 Windows 环境。
 3. 除在线仿真器 IE-78K0S-NS 和 IE-78K0S-NS-A 之外的所有产品是可选的。
 4. 在线仿真器 IE-789234-NS-EM1 提供目标设备电缆。
 5. 计划使用 SM+作为 78K0S/Kx1+的系统仿真器。

图 A-1. 开发工具 (4 / 4)

(4) 当使用在线仿真器 QB-78K0SKX1MINI



- 注
1. C 语言库源文件不包含在软件包中。
 2. 项目管理器 PM+ 包含在汇编器包中。
PM+ 仅用于 Windows 环境。
 3. 在线仿真器 QB-78K0SKX1MINI 随集成调试器 ID78K0S-QB、flash 存储器编程器 PG-FPL2、一个电源单元和一根目标设备电缆一起提供。其它产品可选。
 4. 计划使用 78K0S/Kx1+ 的 SM+ 作为系统仿真器。

A.1 软件包

SP78K0S 软件包	它包含了 78K0S 系列开发需要的软件工具。 包括如下工具。 RA78K0S, CC78K0S, ID78K0S-NS, 78K0S 的 SM+ ^{※1} , SM78K0S ^{※2} 和其它设备文件
	产品代号: μ SxxxxSP78K0S

- 注
1. 支持 78K0S 的 SM+不包含在 SP78K0S 版本。2.00 或更早。
 2. The SM78K0S 不支持 78K0S/Kx1+。
 3. The DF789234 不包含在 SP78K0S 版本 2.00 或更早。DF789234 可以从如下网站获得。
<http://www.necel.com/micro/ods/eng/>

备注 xxxxx 产品代号中的随使用的操作系统而变化。

μ SxxxxSP78K0S

xxxx	主机	OS	存储介质
AB17	PC-9800 系列,	日语 Windows	CD-ROM
BB17	IBM PC/AT 兼容机	英语 Windows	

A.2 语言处理软件

RA78K0S 汇编器包	将助记符表示的程序转换为微处理器可以执行的目标代码的程序。 此外, 也提供产生符号列表和优化跳转指令的自动函数。同设备文件 (DF789234) 一同使用 (单独销售)。 <PC 环境中使用的注意事项> 汇编器包是基于 DOS 的应用程序但可以通过使用 PM 在 Windows 环境下使用 (包含在汇编器包中)。 产品代号: μ SxxxxRA78K0S
CC78K0S C 语言库程序包	将 C 代码表示的程序转换为微处理器可以执行的目标代码的程序。 同汇编器包 (RA78K0S) 和设备文件 (DF789234) 一同使用 (两者都单独销售)。 <PC 环境中使用的注意事项> C 语言库程序包是基于 DOS 的应用程序但可以通过使用 PM 在 Windows 环境下使用 (包含在汇编器包中)。 产品代号: μ SxxxxCC78K0S
DF789234 ^{※1} 设备文件	该文件包含设备特有的信息。 同其它工具 (RA78K0S, CC78K0S, ID78K0S-NS, ID78K0S-QB 或支持 78K0S 的 SM+) 一同使用 (所有的都独立销售)。 产品代号: μ SxxxxDF789234
CC78K0S-L ^{※2} C 语言库源文件	包含在 C 语言编译器包中的目标库组成的函数源文件 需要根据用户的指定来改变 C 语言库程序包中包含的目标库。 由于这是源文件, 因此它的工作环境不依赖于任何特定的操作系统。 产品代号: μ SxxxxCC78K0S-L

- 注 1. DF789234 是个公共文件可用于 RA78K0S、CC78K0S、ID78K0S-NS、ID78K0S-QB 和支持 78K0S 的 SM+。
2. CC78K0S-L 不包含在软件包 (SP78K0S) 中。

备注 产品代号中的xxxx随主机和使用的操作系统而变化。

μSxxxxRA78K0S
μSxxxxCC78K0S
μSxxxxCC78K0S-L

xxxx	主机	OS	提供介质
AB17	PC-9800 系列, IBM PC/AT 兼容机	日语 Windows	CD-ROM
BB17		英语 Windows	
3P17	HP9000 系列 700™	HP-UX™ (Rel.10.10)	
3K17	SPARCstation™	SunOS™ (Rel.4.1.4), Solaris™ (Rel.2.5.1)	

μSxxxxDF789234

xxxx	主机	OS	提供介质
AB13	PC-9800 系列, IBM PC/AT 兼容机	日语 Windows	3.5" 2HD FD
BB13		英语 Windows	

A.3 控制软件

PM+ 项目管理器	这是使用户程序可以在 Windows 环境下进行有效地开发的控制软件。通过该软件，一系列用户程序的开发操作，包括启动编辑器，编译和启动调试器，都可以在 PM+ 上执行。 <注意事项> PM+ 包含于汇编器包 (RA78K0S) 中。它仅在 Windows 环境下使用。
--------------	---

A.4 Flash存储器写入工具

FlashPro4 (FL-PR4, PG-FP4) 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 2 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。
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 flash 存储器的微控制器。当使用所有的 flash 微控制器 (包括 78K0S/Kx1+) 开发应用系统时，它作为片上仿真器能够调试硬件和软件。
FA-78F9212GR-JJG-MX Flash 存储器写入适配器	Flash 存储器写入适配器与 flash 存储器编程器一同使用。

备注 FL-PR4 和 FA-78F9212GR-JJG-MX 是 Naito Densei Machida Mfg 的产品。Co., Ltd.
关于更多的信息，联系：Naito Densei Machida Mfg.Co., Ltd. (电话 +81-42-750-4172)

A.5 调试工具（硬件）

A.5.1 当使用在线仿真器QB-78K0SKX1

QB-78K0SKX1 在线仿真器	当使用 78K0S/Kx1+ 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（ID78K0S-QB）。它被连接到 AC 适配器，目标设备电缆和通过一根 USB 电缆连接到主机。
QB-50-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-64 GR-EA-01T 交换适配器	该交换适配器用于实现从在线仿真器到目标连接器的引脚转换。
QB-16GR-NQ-01T 目标连接器	该目标连接器用于安装在目标系统上。
目标系统中引脚头的规范	0.635 mm × 0.635 mm（高度：6 mm）

备注 QB-78K0SKX1 随 AC 适配器、一根 USB 电缆、目标设备电缆、集成调试器 ID78K0S-QB 和带有编程功能的片上调试仿真器 QB-MINI2 一同提供。

QB-78K0SKX1 同时提供一个仿真探头，一个交换适配器和一个目标设备连接器依赖于订购编号。

A.5.2 当使用在线仿真器QB-MINI2

QB-MINI2 带有编程功能的片上调试仿真器	当使用所有的 flash 微控制器（包括 78K0S/Kx1+）开发应用系统时，它作为片上仿真器能够调试硬件和软件。这是一个简易 flash 存储器编程器，专用于带有 flash 存储器的微控制器。
目标系统中引脚头的规范	16 引脚通用连接器（2.54 mm 间距）

备注 QB-MINI2 提供一条 USB 接口电缆和一条目标设备电缆。此外，集成调试器（包括 ID78K0S-QB）可以作为控制软件使用。集成调试器可以从如下网站获得。

<http://www.necel.com/micro/ods/eng/>

A.5.3 当使用在线仿真器IE-78K0S-NS or IE-78K0S-NS-A

IE-78K0S-NS 在线仿真器	用于使用 78K0S 系列的应用系统的软件和硬件调试的在线仿真器。支持集成调试器 (ID78K0S-NS)。与 AC 适配器、仿真探头和连接到主机的接口适配器组合使用。
IE-78K0S-NS-A 在线仿真器	该在线仿真器涵盖了 IE-78K0S-NS 的所有功能并增强了调试功能，例如增强跟踪功能和定时器功能。
IE-70000-MC-PS-B AC 适配器	适配器提供 100 到 240V AC 输出电源
IE-70000-CD-IF-A PC 卡接口	当使用笔记本类型的 PC 机作为主机时（支持 PCMCIA 座的）需要 PC 卡和接口电缆。
IE-70000 PCI--IF-A 接口适配器	当使用带有 PCI 总线的个人电脑作为主机时需要该适配器。
IE-789234-NS-EM1 仿真板	用于仿真与设备有关的周边硬件的仿真板。 同在线仿真器一同使用。提供一条目标设备电缆。
目标系统中引脚头的规范	0.635 mm × 0.635 mm （高度：6 mm）

A.5.4 当使用在线仿真器QB-78K0SKX1MINI

QB-78K0SKX1MINI 在线仿真器	用于使用 78K0S/Kx1+系列的应用系统的软件和硬件调试的在线仿真器。支持集成调试器 (ID78K0S- QB)。与 AC 适配器、目标设备电缆和连接到主机的接口电缆组合使用。
目标系统中引脚头的规范	0.635 mm × 0.635 mm （高度：6 mm）

A.6 调试工具（软件）

ID78K0S-NS （支持在线仿真器 IE-78K0S-NS/ IE-78K0S-NS-A） 集成调试器	该调试器支持 78K/0S 系列的在线仿真器。ID78K0S-NS 是基于 Windows 的软件。调试器拥有支持 C 语言的增强调试功能。通过使用集成了源程序、反汇编显示和带有跟踪结果的存储器显示的窗口集成功能，跟踪结果可以同源程序对应显示。 同设备文件（DF789234）一同使用（单独销售）。 产品代号：μS××××ID78K0S-NS
ID78K0S-QB （支持在线仿真器 QB-78K0SKX1MINI） 集成调试器	该调试器支持 78K0S/Kx1+系列的在线仿真器。ID78K0S- QB 是基于 Windows 的软件。可能提供支持 C 语言、源程序、反汇编显示和存储器显示的调试功能。同设备文件（DF789234）一同使用（单独销售）。 随在线仿真器 QB-78K0SKX1MINI 提供。 订购编号：μS××××ID78K0S-QB（不用于销售）
支持 78K0S 的 SM+ 系统仿真器	这是一个 78K/0S 系列的系统仿真器。支持 78K0/Kx2 的 SM+ 是基于 Windows 的软件。当在主机上仿真目标系统的操作时，这个仿真器可以执行 C 源代码级或汇编级的调试。通过使用支持 78K0S 的 SM+，应用程序的逻辑和操作可以独立于硬件系统进行验证。因此，可以增强开发的效率并提高软件的质量。 同设备文件（DF789234）一同使用（单独销售）。 产品代号：μS××××SM789234-B
DF789234 ^注 设备文件	这是包含设备特定信息的文件。 同其它工具（RA78K0S，CC78K0S，ID78K0S-NS，ID78K0S-QB 或支持 78K0S 的 SM+）一同使用（所有的都独立销售）。 产品代号：μS××××DF789234

注 DF789234 是个公共文件可用于 RA78K0S、CC78K0S、ID78K0S-NS、ID78K0S-QB 和支持 78K0S 的 SM+。

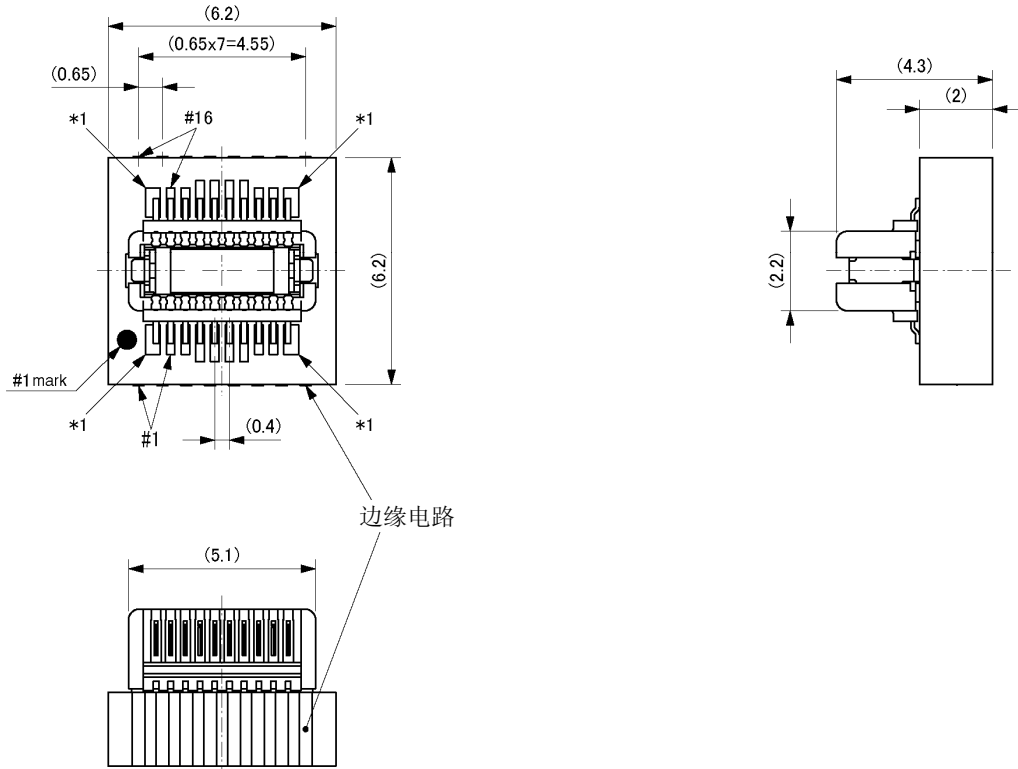
附录B 目标系统设计的注意事项

开发工具的封装图如下所示。

考虑目标系统上禁止安装组件的区域以及使用 QB-78K0SKX1 时对组件安装高度有限制的区域。

图 B-1. 目标系统连接器 (QB-16GR-NQ-01T)

(a) 封装图



(b) 焊盘类型

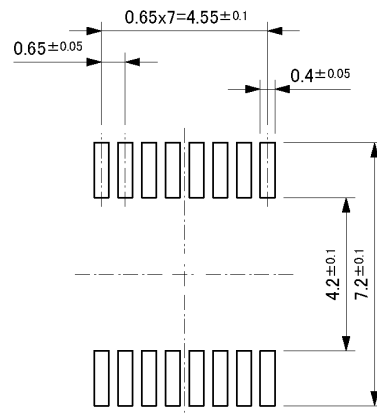


图 B-2. 交换适配器 (QB-16GR-EA-01T)

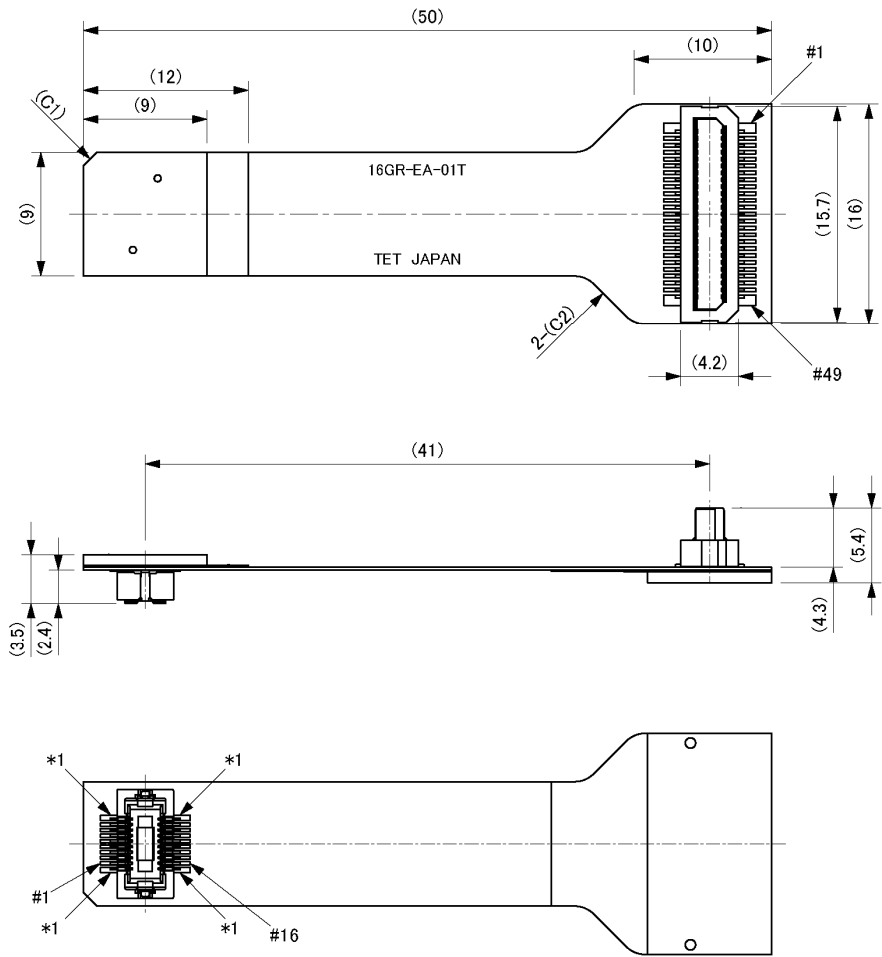
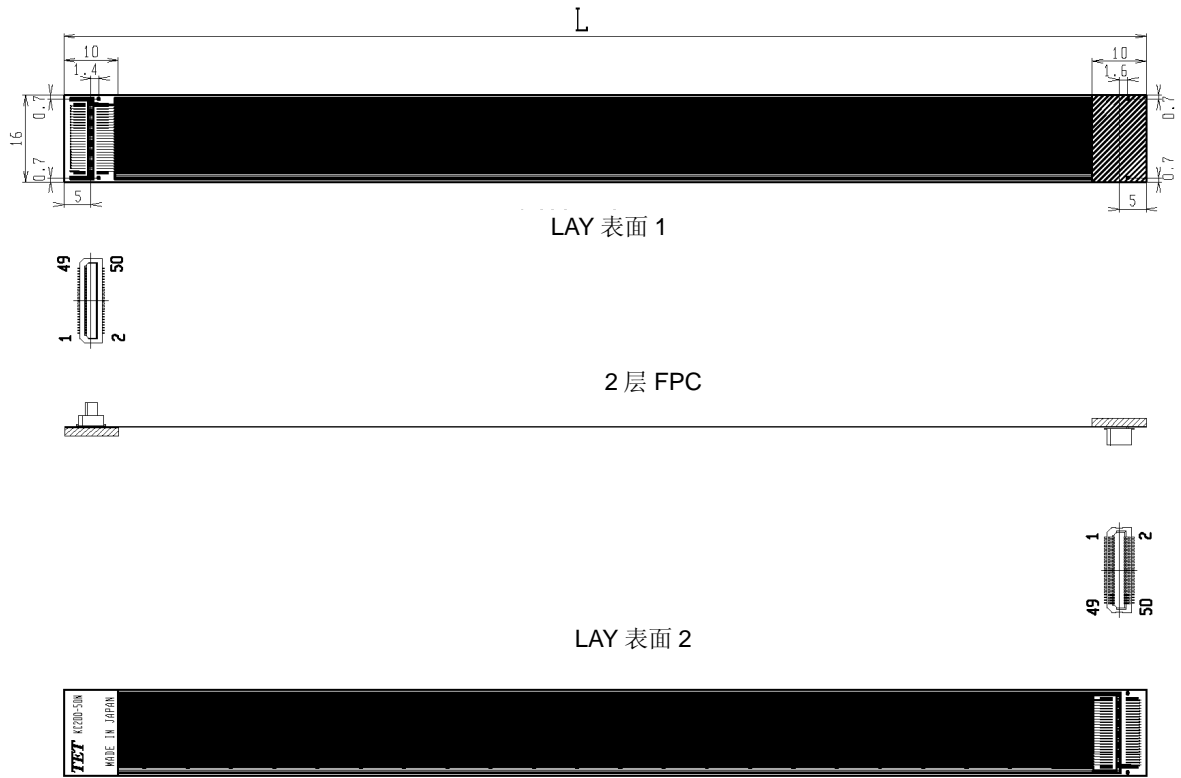


图 B-3. 仿真探头 (QB-50-EP-01T)



C.1 寄存器索引（寄存器名称）

8 位定时器 H 比较寄存器 01（CMP01） ... 123
8 位定时器 H 比较寄存器 11（CMP11） ... 123
8 位定时器 H 模式寄存器 1（TMHMD1） ... 124
16 位定时器捕获 / 比较寄存器 000（CR000） ... 82
16 位定时器捕获 / 比较寄存器 010（CR010） ... 84
16 位定时器计数器 00（TM00） ... 82
16 位定时器模式控制寄存器 00（TMC00） ... 85
16 位定时器输出控制寄存器 00（TOC00） ... 88

[C]

捕获 / 比较控制寄存器 00（CRC00） ... 87

[E]

外部中断模式寄存器 0（INTM0） ... 150

[F]

Flash 地址指针 H（FLAPH） ... 209
Flash 地址指针 L（FLAPL） ... 209
Flash 地址指针 H 比较寄存器（FLAPHC） ... 209
Flash 地址指针 L 比较寄存器（FLAPLC） ... 209
Flash 编程命令寄存器（FLCMD） ... 208
Flash 编程模式控制寄存器（FLPMC） ... 205
Flash 保护命令寄存器（PFCMD） ... 206
Flash 状态寄存器（PFS） ... 206
Flash 写缓存寄存器（FLW） ... 210

[I]

中断屏蔽标志寄存器 0（MK0） ... 150
中断请求标志寄存器 0（IF0） ... 149

[L]

低速内部振荡模式寄存器（LSRCM） ... 67
低电压检测寄存器（LVIM） ... 179
低电压检测电平选择寄存器（LVIS） ... 180

[O]

振荡稳定时间选择寄存器（OSTS） ... 68, 158

[P]

- 端口模式寄存器 2 (PM2) ... 59, 90
- 端口模式寄存器 3 (PM3) ... 59
- 端口模式寄存器 4 (PM4) ... 59
- 端口寄存器 2 (P2) ... 60
- 端口寄存器 3 (P3) ... 60
- 端口寄存器 4 (P4) ... 60
- 预处理器时钟控制寄存器 (PPCC) ... 66
- 预分频模式寄存器 00 (PRM00) ... 89
- 处理器时钟控制寄存器 (PCC) ... 66
- 上拉电阻选项寄存器 2 (PU2) ... 61
- 上拉电阻选项寄存器 3 (PU3) ... 61
- 上拉电阻选项寄存器 r 4 (PU4) ... 61

[R]

- 复位控制标志寄存器 (RESF) ... 173

[W]

- 看门狗定时器允许寄存器 (WDTE) ... 139
- 看门狗定时器模式寄存器 (WDTM) ... 138

C.2 寄存器索引（符号）

[C]

CMP01: 8 位定时器 H 比较寄存器 01 ... 123
CMP11: 8 位定时器 H 比较寄存器 11 ... 123
CR000: 16 位定时器捕获 / 比较寄存器 000 ... 82
CR010: 16 位定时器捕获 / 比较寄存器 010 ... 84
CRC00: 捕获 / 比较控制寄存器 00 ... 87

[F]

FLAPH: Flash 地址指针 H ... 209
FLAPHC: Flash 地址指针 H 比较寄存器 ... 209
FLAPL: Flash 地址指针 L ... 209
FLAPLC: Flash 地址指针 L 比较寄存器 ... 209
FLCMD: Flash 编程命令寄存器 ... 208
FLPMC: Flash 编程模式控制寄存器 ... 205
FLW: Flash 写缓存寄存器 ... 210

[I]

IF0: 中断请求标志寄存器 0 ... 149
INTM0: 外部中断模式寄存器 0 ... 150

[L]

LSRCM: 低速内部振荡模式寄存器 ... 67
LVIM: 低电压检测寄存器 ... 179
LVIS: 低电压检测电平选择寄存器 ... 180

[M]

MK0: 中断屏蔽标志寄存器 0 ... 150

[O]

OSTS: 振荡稳定时间选择寄存器 ... 68, 158

[P]

P2:	端口寄存器 2 ... 60
P3:	端口寄存器 3 ... 60
P4:	端口寄存器 4 ... 60
PCC:	处理器时钟控制寄存器 ... 66
PFCMD:	Flash 保护命令寄存器 ... 206
PFS:	Flash 状态寄存器 ... 206
PM2:	端口模式寄存器 2 ... 59, 90
PM3:	端口模式寄存器 3 ... 59
PM4:	端口模式寄存器 4 ... 59
PPCC:	预处理器时钟控制寄存器 ... 66
PRM00:	预分频模式寄存器 00 ... 89
PU2:	上拉电阻选项寄存器 2 ... 61
PU3:	上拉电阻选项寄存器 3 ... 61
PU4:	上拉电阻选项寄存器 4 ... 61

[R]

RESF:	复位控制标志寄存器 ... 173
-------	-------------------

[T]

TM00:	16 位定时器计数器 00 ... 82
TMC00:	16 位定时器模式控制寄存器 00 ... 85
TMHMD1:	8 位定时器 H 模式寄存器 1 ... 124
TOC00:	16 位定时器输出控制寄存器 00 ... 88

[W]

WDTE:	看门狗定时器允许寄存器 ... 139
WDTM:	看门狗定时器模式寄存器 ... 138

附录 D 注意事项列表

该附录列出了本文档中描述的注意事项。
表中的“分类（硬件 / 软件）”如下所示。

硬件： 微控制器内部 / 外部硬件的注意事项
软件： 软件的注意事项，例如寄存器设置或程序

(1 / 12)

章节	分类	功能	功能详述	注意事项	页码
第二章	硬件	引脚功能	P22/X2, P23/X1	P22/X2 和 P23/X1 引脚在复位期间下拉。	pp. 20, 21, 22, 23 <input type="checkbox"/>
第三章	软件	存储器空间	SP: 堆栈指针	由于复位信号生成会使用SP内容变为未定义，因此在使用堆栈存储器前请务必初始化SP。	p. 34 <input type="checkbox"/>
				堆栈指针只能被设置为高速RAM区域，并且只有低10位可以实际被设置。因此，如果堆栈指针被指定为0FF00H，它将被转换为高速RAM区域中的0FB00H，因为0FF00H处于SFR区域中而不在高速RAM区域中。当数值被实际推入堆栈时，0FB00H减去1变为0FAFFH，但是由于该值不在高速RAM区域中，它将被转换为0FEFFH，这与0FF00H被设置为堆栈指针时的值相同。	p. 34 <input type="checkbox"/>
第四章	硬件	端口功能	P22/X2, P23/X1	P22/X2 和 P23/X1 引脚在复位期间下拉。	p. 51 <input type="checkbox"/>
			P34	因为P34引脚复用作RESET引脚，如果它用作输入端口引脚，输入一个外部复位信号到RESET引脚的功能不能被使用。端口的功能通过选项字节来选择。关于详细情况，参阅第十四章 选项字节。 同时，由于选项字节在复位释放后被引用，如果在引用之前低电平被输入RESET引脚，复位状态不会被释放。当它用作输入端口引脚时，连接上拉电阻。	p. 57 <input type="checkbox"/>
			P21, P32	因为P21和P32也用作外部中断引脚，如果这些引脚被设置为输出模式并且其输出电平被更改，对应的中断请求标志被设置。要在输出模式下使用端口引脚，事先设置对应的中断屏蔽标志为1。	p. 59 <input type="checkbox"/>
			-	尽管1位存储器操作指令修改1位，但是它以8位为单位访问端口。因此，在输入和输出混合的端口中，输入模式下引脚的输出锁存的内容不确定，即使它不是指令修改的对象。	p. 62 <input type="checkbox"/>
第五章	软件	主时钟	OSTS: 振荡稳定时间选择寄存器	要设置并释放STOP模式，按照下面设置振荡稳定时间。 谐振器的期望振荡稳定时间 ≤ 由OSTS设置的振荡稳定时间	p. 68 <input type="checkbox"/>

章节	分类	功能	功能详述	注意事项	页码
第五章	软件	主时钟	OSTS: 振荡稳定时间选择寄存器	无论STOP模式是由复位信号产生释放还是由中断产生释放, STOP模式释放后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。	p. 68 <input type="checkbox"/>
				通过选项字节选择上电或复位释放后的振荡稳定时间。关于详细情况, 参阅第十四章 选项字节。	p. 68 <input type="checkbox"/>
	硬件	晶体 / 陶瓷振荡器	-	当使用晶体 / 陶瓷振荡器时, 图 5-6中被虚线包围的部分的配线应按照如下布线方法布线, 以防止走线电容产生不利影响。 <ul style="list-style-type: none"> • 连接线越短越好。 • 连接线不应与其他信号线交叉。流经的电流变化较大的信号线不要在振荡器周围布线 • 要保持振荡器电容器的接地点电压与Vss相同。不要将电容的地信号接入大电流地。 • 不要从振荡器获取信号。 	p. 69 <input type="checkbox"/>
第六章	硬件	16位定时器 / 事件计数器00	TM00: 16位定时器计数器 00	即使TM00被读取, 也不能通过CR010来捕获。	pp. 82, <input type="checkbox"/> 114
				在定时器计数期间, 如果TM00被参考, 定时器计数将在参考处理期间停止, 并且在参考处理完成后定时器计数重新开始。因此, 如果参考TM00的处理被执行, 定时器计数中将出现一个错误。	pp. 82, <input type="checkbox"/> 114
	软件	16位定时器捕获 / 比较寄存器 000	CR000: 16位定时器捕获 / 比较寄存器 000	在TM00与CR000匹配时清除&开始发生的模式下, 设置CR000为0000H以外的值。这意味着当该寄存器被用作外部事件计数器时, 1脉冲计数操作不能被执行。	pp. 83, <input type="checkbox"/> 114
				然而, 在自由运行和TI000引脚有效沿输入时清除&开始发生的模式下, 如果CR000被清除为0000H, 当紧跟着溢出(FFFFH)后CR000的值从0000H变为0001H时, 一个中断请求(INTTM00)被产生。	pp. 83, <input type="checkbox"/> 114
				如果CR000的新值小于16位定时器计数器0(TM00)的值, TM00继续计数, 溢出, 然后从0开始再次计数。因此, 如果CR000的新值小于旧的值, 在CR000被更改后, 定时器必须被复位来重新开始。	pp. 83, <input type="checkbox"/> 114
				16位定时器 / 事件计数器00停止后CR000的值将不能被保证。	pp. 83, <input type="checkbox"/> 115
				如果捕获触发已经被输入, 捕获操作可能不会对设置为比较模式的CR000执行。	pp. 83, <input type="checkbox"/> 117
				当P21被用作有效沿的输入引脚(TI010)时, 它不能用作定时器输出(TO00)。当使用P21作为定时器输出引脚(TO00)时, 它不能用作有效沿的输入引脚(TI010)。	pp. 83, <input type="checkbox"/> 90, 119
				如果当CR000用作捕获寄存器时寄存器读取周期和捕获触发的输入冲突, 捕获触发输入具有优先权并且读取的数据不确定。同时, 如果定时器计数停止和捕获触发输入冲突, 捕获触发不确定。	pp. 83, <input type="checkbox"/> 116
软件			更改CR000设置可能导致故障。要更改设置, 参考 6.5 16位定时器 / 事件计数器00相关的注意事项 (17) 定时器操作期间更改比较寄存器。	p. 83 <input type="checkbox"/>	

章节	分类	功能	功能详述	注意事项	页码
第六章	软件	16位定时器 / 事件计数器00	CR010: 16位定时器捕获 / 比较寄存器010	在自由运行和TI000 引脚有效沿输入时清除&开始发生的模式下, 如果CR010 被清除为0000H, 当紧跟着溢出 (FFFFH) 后CR010 的值从0000H 变为 0001H时, 一个中断请求 (INTTM010) 被产生。	pp. 84, 114 <input type="checkbox"/>
				如果CR010 的新值小于16位定时器计数器00 (TM00) 的值, TM00继续计数, 溢出, 然后从0开始再次计数。因此, 如果CR010 的新值小于旧的值, 在CR010被更改后, 定时器必须被复位来重新开始。	pp. 84, 114 <input type="checkbox"/>
				16位定时器 / 事件计数器00停止后CR010 的值将不能被保证。	pp. 84, 115 <input type="checkbox"/>
				如果捕获触发已经被输入, 捕获操作可能不会对设置为比较模式的CR010执行。	pp. 84, 117 <input type="checkbox"/>
				如果当CR010用作捕获寄存器时寄存器读取周期和捕获触发的输入冲突, 捕获触发输入具有优先权并且读取的数据不确定。同时, 如果定时器计数停止和捕获触发的输入冲突, 捕获数据不确定。	pp. 84, 116 <input type="checkbox"/>
				在TM00操作期间更改CR010 设置可能导致故障。要更改设置, 参考 6.5 16位定时器 / 事件计数器00相关的注意事项 (17) 定时器操作期间更改比较寄存器。	p. 85 <input type="checkbox"/>
	硬件	16位定时器模式控制寄存器00	TMC00: 16位定时器模式控制寄存器00	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) 的值时开始操作。将TMC002和TMC003设置为0, 0以停止操作。	pp. 85, 114 <input type="checkbox"/>
				在重写OVF00 标志以外的位之前, 定时器操作必须被停止。	pp. 86, 115 <input type="checkbox"/>
				如果定时器被停止, 即使信号被输入到TI000/TI010引脚, 定时器计数和定时器中断也不会发生。	pp. 86, 114 <input type="checkbox"/>
				除了当TI000 引脚的有效沿被选作计数时钟时, 在设置STOP模式或系统时钟停止模式之前停止计数器操作; 否则, 当系统时钟开始时定时器可能发生故障。	pp. 86, 119 <input type="checkbox"/>
				在停止定时器操作后, 使用预分频模式寄存器00 (PRM00) 的第4和5位来设置 TI000引脚的有效沿。	pp. 86, 115 <input type="checkbox"/>
				如果TM00 和 CR000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被选择, 当CR000 的值为FFFFH 并且TM00 的值由FFFFH 变为0000H时, OVF00 标志被设置为1。	p. 86 <input type="checkbox"/>
	软件	16位定时器模式控制寄存器00	TMC00: 16位定时器模式控制寄存器00	如果在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标志将被重新设置并且禁止清除。	pp. 86, 116 <input type="checkbox"/>
				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 (INTTM0n0) 发生。	pp. 86, 117 <input type="checkbox"/>
				在设置CRC00之前, 定时器操作必须被停止。	pp. 87, 115 <input type="checkbox"/>
				使用16位定时器模式控制寄存器00 (TMC00) 选择TM00 和 CR000匹配时清除&开始发生的模式时, CR000 不应该被指定为捕获寄存器。	pp. 87, 114 <input type="checkbox"/>
				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 (参阅图6-17)。	pp. 87, 117 <input type="checkbox"/>
				硬件	16位定时器模式控制寄存器00
使用16位定时器模式控制寄存器00 (TMC00) 选择TM00 和 CR000匹配时清除&开始发生的模式时, CR000 不应该被指定为捕获寄存器。	pp. 87, 114 <input type="checkbox"/>				

章节	分类	功能	功能详述	注意事项	页码
第六章	软件	16位定时器 / 事件计数器00	TOC00: 16位定时器输出控制寄存器00	在设置OSPT00以外的位之前, 定时器操作必须被停止。	pp. 88, 115
				如果 LVS00 和 LVR00 被读取, 0被读出。	pp. 88, 115
				在数据被设置后, OSPT00 被自动清除, 所以0被读出。	pp. 88, 115
				在单脉冲输出以外的模式中, 不要设置OSPT00 为1。	pp. 88, 115
	硬件		当OSPT00 被连续设置为1时, 需要通过预分频模式寄存器00 (PRM00) 选择的两个或更多个计数时钟周期的写入间隔。	pp. 88, 115	
			当TOE00 为0时, 使用8位存储器操作指令同时设置TOE00、LVS00和LVR00。当TOE00 为1时, 使用1位存储器操作指令设置LVS00和LVR00。	p. 89	
	软件		PRM00: 预分频模式寄存器00	在停止定时器操作后, 总是设置数据到PRM00 。	pp. 89, 115
				如果TI000 引脚的有效沿将被设置为计数时钟, 不要在TI000 引脚的有效沿时设置清除 / 开始模式和捕获触发。	pp. 89, 117
				在以下情况下, 注意TI0n0 引脚的有效沿被检测。 <1> 系统复位后, 如果高电平被输入TI0n0 引脚, 16位定时器计数器00 (TM00) 的操作被允许。 →如果上升沿和双边沿都被指定为TI0n0 引脚的有效沿, 在TM00 操作被允许后, 一个上升沿立即被检测。 <2> 如果在TI0n0 引脚为高电平过程中TM00 操作被停止, 在低电平被输入到TI0n0 引脚后, TM00 操作被允许。 →如果下降沿和双边沿都被指定为TI0n0 引脚的有效沿, 在TM00 操作被允许后, 一个下降沿立即被检测。 <3> 如果在TI0n0 引脚为低电平过程中TM00 操作被停止, 在高电平被输入到TI0n0 引脚后, TM00 操作被允许。 →如果上升沿和双边沿都被指定为TI0n0 引脚的有效沿, 在TM00 操作被允许后, 一个上升沿立即被检测。	pp. 90, 119
				用于移除噪声的采样时钟与TI000的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fxp, 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被采样并且有效电平被检测到两次使才会执行, 这样可以消除短脉冲宽度的噪声。	pp. 90, 119
				当P21 被用作有效沿的输入引脚 (TI010) 时, 它不能用作定时器输出 (TO00)。当P21 被用作定时器输出引脚 (TO00) 时, 它不能用作有效沿的输入引脚 (TI010) 。	pp. 90, 119
				当读取外部事件计数器计数值时, TM00应该被读取。	pp. 95, 119
	硬件		间隔定时器	在TM00操作期间更改CR000 设置可能导致故障。要更改设置, 参考 6.5 16位定时器 / 事件计数器00相关的注意事项 (17) 定时器操作期间更改比较寄存器。	p. 91
			外部事件计数器	当读取外部事件计数器计数值时, TM00应该被读取。	pp. 95, 119

章节	分类	功能	功能详述	注意事项	页码
第六章	软件	16位定时器 / 事件计数器00	脉冲宽度测量	要使用两个捕获寄存器，设置TI000 设置 TI010引脚。	pp. 96, 117 <input type="checkbox"/>
				该操作示例中可测量的脉冲宽度达到定时器计数器的1个周期。	pp. 96, 98, 100, 102 <input type="checkbox"/>
			方波输出	在TM00操作期间更改CR000 设置可能导致故障。要更改设置，参考 6.5 16位定时器 / 事件计数器00相关的注意事项 (17) 定时器操作期间更改比较寄存器。	p. 104 <input type="checkbox"/>
			PPG 输出	在TM00操作期间更改CRC0n0 设置可能导致故障。要更改设置，参考 6.5 16位定时器 / 事件计数器00相关的注意事项 (17) 定时器操作期间更改比较寄存器。	p. 106 <input type="checkbox"/>
				以下范围内的值应该被设置到CR000 和 CR010中： 0000H < CR010 < CR000 ≤ FFFFH	pp. 107, 119 <input type="checkbox"/>
			PPG 输出	通过PPG输出产生的脉冲的周期 (CR000设置值 + 1) 具有 (CR010 设置值 + 1) / (CR000 设置值 + 1) 的占空比。	pp. 107, 119 <input type="checkbox"/>
			单脉冲输出：软件触发	单脉冲输出期间不要再次设置OSPT00位为1。要再次输出单脉冲，则应等待直到当前单脉冲输出完成后。	pp. 109, 115 <input type="checkbox"/>
				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。因为外部触发在这种情况下有效，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，导致脉冲以不确定的时序输出。	pp. 109, 115 <input type="checkbox"/>
				不要设置0000H到CR000 和 CR010寄存器。	pp. 110, 116 <input type="checkbox"/>
	硬件	单脉冲输出：外部触发	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。	pp. 111, 114 <input type="checkbox"/>	
			单脉冲输出期间不要再次输入外部触发。要再次输出单脉冲，则应等待直到当前单脉冲输出完成后。	pp. 111, 116 <input type="checkbox"/>	
			不要设置0000H到CR000 和 CR010寄存器。	pp. 112, 116 <input type="checkbox"/>	
	软件	单脉冲输出：外部触发	只要00 (操作停止模式) 以外的值被设置到TMC002和TMC003位，16位定时器计数器00就开始操作。	pp. 113, 114 <input type="checkbox"/>	
			定时器开始错误	在定时器启动后生成匹配信号所需的时间内至少一个时钟可能会发生错误。这是由于16位定时器计数器00 (TM00) 与计数时钟异步启动。	p. 114 <input type="checkbox"/>
			单脉冲输出	单脉冲输出只有在自由运行模式或TI000引脚有效沿时清除&开始发生的模式中才能正常进行。因为在TM00 寄存器和CR000寄存器匹配时清除&开始发生的模式中溢出不会发生，所以单脉冲不能输出。	p. 115 <input type="checkbox"/>
软件	捕获操作	当CRC001 位为1时，如果上升沿和下降沿都被指定为TI000 引脚的有效沿，CR000 寄存器中的捕获操作不会执行。	p. 117 <input type="checkbox"/>		
		当CRC001 位的值为1时，当TI010 引脚的有效沿被检测时，TM00 的计数值不会被不会到CR000 寄存器中，但是从TI010 引脚的输入可以被用作外部中断源，因为INTTM000 在那时被产生。	p. 117 <input type="checkbox"/>		

章节	分类	功能	功能详述	注意事项	页码
第六章	软件	16位定时器 / 事件计数器00	定时器操作期间更改比较寄存器	对于16位定时器捕获 / 比较寄存器0n0 (CR0n0) 用作比较寄存器, 在定时器计数期间16位定时器计数器00 (TM00) 和16位定时器捕获 / 比较寄存器0n0 (CR0n0) 匹配时序附近更改CR0n0 时, 更改时序可能与匹配的时序冲突, 所以在这种情况下操作不被保证。要在定时器计数期间更改CR0n0, INTTM000 中断服务程序执行以下操作。	p. 118 <input type="checkbox"/>
				如果在定时器计数期间CR010 被更改而没有执行上面的处理<1>, CR010 中的值可能被重写两次或更多, 导致每次重写时TO00 引脚的输出电平反转。	p. 118 <input type="checkbox"/>
			外部事件计数器	计数开始的时序在两个有效沿检测后。	p. 119 <input type="checkbox"/>
第七章	软件	8位定时器 H1	CMP01: 8位定时器 H比较寄存器 01	CMP01在定时器计数操作期间不能被重新写入。	p. 123 <input type="checkbox"/>
			CMP11: 8位定时器 H比较寄存器 11	在PWM输出模式下, 在定时器计数操作停止 (TMHE1 = 0) 后当定时器计数操作开始 (TMHE1 = 1) 时, 确保设置CMP11 (即使设置相同的值到CMP11, 也要确保再次设置)。	p. 123 <input type="checkbox"/>
			TMHMD1: 8位定时器 H模式寄存器1	当TMHE1 = 1时, 禁止设置TMHMD1寄存器的其它位。	p. 125 <input type="checkbox"/>
				在PWM输出模式下, 在定时器计数操作停止 (TMHE1 = 0) 后当定时器计数操作开始 (TMHE1 = 1) 时, 确保设置8位定时器H比较寄存器11 (CMP11) (即使设置相同的值到CMP11, 也要确保再次设置)。	p. 125 <input type="checkbox"/>
	硬件	PWM 输出	在PWM输出模式下, CMP11 寄存器的设置值可以在定时器计数操作期间改变。然而, 在重新写入CMP11 寄存器的值后需要三个操作时钟 (使用TMHMD1 寄存器的CKS12 到 CKS10位选择的信号) 或更多周期来传输CMP11 寄存器值。	p. 130 <input type="checkbox"/>	
			在定时器计数操作停止 (TMHE1 = 0) 后当定时器计数操作开始 (TMHE1 = 1) 时, 确保设置CMP11寄存器 (即使设置相同的值到CMP11寄存器, 也要确保再次设置)。	p. 130 <input type="checkbox"/>	
第八章	软件	看门狗定时器	WDTM: 看门狗定时器模式寄存器	分别设置第7、6和5位为0、1和1。不要设置其它值。	p. 138 <input type="checkbox"/>
				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。然而, 第一次写入时, 如果“1”和“x”被分别设置到WDCS4 和 WDCS3并且看门狗定时器被停止, 即使以下操作被执行, 内部复位信号也不会发生。 <ul style="list-style-type: none"> 第二次写入WDTM 对WDTE执行1位存储器操作指令。 “ACH”以外的值被写入WDTE 	p. 139 <input type="checkbox"/>
				WDTM 不能通过1位存储器操作指令来设置。	p. 139 <input type="checkbox"/>
				当通过自写入使用flash存储器自编程时, 为看门狗定时器设置溢出时间以保证足够的溢出时间 (示例 1字节写入: 200 μ s 最小值, 1-block 检测: 10 ms 最小值)。	p. 139 <input type="checkbox"/>

章节	分类	功能	功能详述	注意事项	页码
第八章	软件	看门狗定时器	WDTE: 看门狗定时器允许寄存器	如果一个ACH以外的值被写入WDTE, 一个内部复位信号被产生。	p. 139 <input type="checkbox"/>
				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。	p. 139 <input type="checkbox"/>
				从WDTE读取的值为9AH (这不同于写入的值 (ACH))。	p. 139 <input type="checkbox"/>
	硬件	当“低速内部振荡器可由软件停止”由选项字节选择时	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不能被停止。对应8位定时器H1 (TMH1), 低速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TMH1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。	p. 140 <input type="checkbox"/>	
当“低速内部振荡器可由软件停止”由选项字节选择时		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。	p. 142 <input type="checkbox"/>		
第九章	软件	中断功能	IF0: 中断请求标志寄存器, MK0: 中断屏蔽标志寄存器	因为P21和P32具有外部中断输入的复用功能, 当通过指定端口功能的输出模式更改输出电平时, 一个中断请求标志被设置。因此, 在使用输出模式前, 中断屏蔽标志应该被设置为1。	pp. 149, 150 <input type="checkbox"/>
			INTM0: 外部中断模式寄存器 0	确保清除第0、1、6和7位为0。 在设置INTM0寄存器之前, 确保设置对应的中断屏蔽标志 (××MK× = 1) 来禁止中断。在设置INTM0寄存器之后, 清除中断请求标志 (××IF× = 0), 然后清除中断屏蔽标志 (××MK× = 0), 这将允许中断。	p. 150 <input type="checkbox"/> p. 151 <input type="checkbox"/>
			中断请求被保持未决	在中断请求标志寄存器0 (IF0) 或中断屏蔽标志寄存器0 (MK0) 被访问期间, 中断请求将保持未决。	p. 153 <input type="checkbox"/>
			中断请求未决	对于低优先级的中断, 多重中断也可以被响应。	p. 154 <input type="checkbox"/>

章节	分类	功能	功能详述	注意事项	页码
第十章	软件 硬件 软件 硬件 软件 硬件 软件	待机功能	-	只有当通过选项字节对低速内部振荡器选择“可由软件停止”时，LSRSTOP 设置才有效。	p. 156 <input type="checkbox"/>
			STOP 模式	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（工作于低速内部振荡时钟的周边硬件除外）。	p. 157 <input type="checkbox"/>
				在STOP模式下，如果在STOP模式被设置之前低速内部振荡器正在操作，低速内部振荡时钟的振荡不能被停止（参阅表10-1）。	p. 157 <input type="checkbox"/>
			OSTS：振荡稳定时间选择寄存器	要设置并且然后释放STOP模式，请按照下面设置振荡稳定时间。 期望的谐振器的振荡稳定时间 ≤ 由OSTS设置的振荡稳定时间	p. 158 <input type="checkbox"/>
				STOP模式释放后的等待时间不包含从STOP模式释放到时钟振荡开始（下图中的“a”）的时间，无论STOP模式是否由复位信号产生还是中断产生来释放。	p. 158 <input type="checkbox"/>
				上电或复位释放后的振荡稳定时间由选项字节来选择。关于详细情况，参阅第十四章 选项字节。	p. 158 <input type="checkbox"/>
			HALT 模式设置和操作状态	因为中断请求信号被用于清除等待模式，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清除的中断源时，如果中断请求被设置，那么等待将会立即被清除。	p. 159 <input type="checkbox"/>
	STOP模式设置和操作状态	因为中断请求信号被用于清除等待模式，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，那么等待将会立即被清除。因此，在STOP模式下，在STOP指令被执行并且操作被停止34 μs（典型值）后，正常操作模式被恢复（当晶体 / 陶瓷振荡被使用时，在由振荡稳定时间选择寄存器（OSTS）设置的稳定振荡的额外等待时间过去后）。	p. 162 <input type="checkbox"/>		
第十一章	硬件 硬件 硬件 硬件 软件	复位功能	-	为外部复位将2 μs或更高的低电平输入到RESET引脚中。	p. 166 <input type="checkbox"/>
				复位信号产生期间，系统时钟和低速内部振荡时钟停止振荡。	p. 166 <input type="checkbox"/>
				当RESET引脚用作输入端口引脚（P34）时，如果在由POC电路产生的复位被释放后并且选项字节被引用之前低电平被输入到RESET引脚，μPD78F9510、78F9511、78F9512将被再次复位。复位状态被保持直到高电平被输入到RESET引脚。	p. 166 <input type="checkbox"/>
				LVI电路不会被由LVI电路产生的内部复位信号复位。	p. 167 <input type="checkbox"/>
			由看门狗定时器溢出而复位的时序	在看门狗定时器的内部复位情况下，看门狗定时器也会被复位。	p. 169 <input type="checkbox"/>
			RESF：复位控制标志寄存器	不能通过1位存储器操作指令来读取数据。	p. 173 <input type="checkbox"/>

章节	分类	功能	功能详述	注意事项	页码
第十三章	软件	上电清零电路	上电清零电路的功能	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。	p. 174 <input type="checkbox"/>
	因为POC电路的检测电压（V _{POC} ）范围为 2.1 V ±0.1 V，请使用2.2 到 5.5 V范围内的电压。			p. 174 <input type="checkbox"/>	
	软件		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。	p. 176 <input type="checkbox"/>
第十三章	软件	低电压检测电路	LVIM: 低电压检测寄存器	要停止LVI，遵循以下过程之一。	p. 179 <input type="checkbox"/>
				<ul style="list-style-type: none"> 当使用8位操作指令时：写入 00H 到 LVIM。 当使用1位存储器操作指令时：清除LVION为0。 	
				确保设置第2到6位为0。	p. 179 <input type="checkbox"/>
			LVIS: 低电压检测电平选择寄存器	第4到7位必须被设置为0。	p. 180 <input type="checkbox"/>
			当用作复位时	如果在LVI操作期间上面以外的值被写入，在写入的时刻值变为不确定，因此在写入之前确保停止LVI（LVIM寄存器的第7位（LVION）= 0）。	p. 180 <input type="checkbox"/>
	<1>必须总被执行。当LVIMK = 0时，一个中断可能会在<3>中的处理后立即发生。	p. 181 <input type="checkbox"/>			
	当LVIMD被设置为1时，如果电源电压（V _{DD} ）≥ 检测电压（V _{LVI} ），一个内部复位信号不会被产生。	p. 181 <input type="checkbox"/>			
	低电压检测电路的注意事项	<p>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。</p> <p><1>当用作复位时</p> <p>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。</p> <p><2>当用作中断时</p> <p>中断请求可能被频繁产生。采取下面措施（2）的（b）。</p>	p. 185 <input type="checkbox"/>		
第十四章	硬件	选项字节	上电或复位释放后的振荡稳定时间	只有当晶体 / 陶瓷振荡时钟被选作系统时钟源时，该选项的设置才有效。如果高速内部振荡时钟或外部时钟输入被选作系统时钟源，无等待时间。	p. 189 <input type="checkbox"/>
			RESET 引脚的控制	因为选项字节在复位释放后被引用，如果在选项字节被引用之前一个低电平被输入到RESET引脚，复位状态将不会被释放。同时，当设置RMCE为0时，请连接上拉电阻。	p. 189 <input type="checkbox"/>

章节	分类	功能	功能详述	注意事项	页码
第十四章	硬件	选项字节	系统时钟源的选择	<p>因为X1和X2引脚也用作P23和P22引脚，X1和X2引脚被使用的情况根据选择的系统时钟源而改变。</p> <p>(1) 晶体 / 陶瓷振荡时钟被选择 X1 和 X2 引脚不能被用作I/O 端口引脚，因为它们被用作时钟输入引脚。</p> <p>(2) 外部时钟输入被选择 因为X1引脚被用作外部时钟输入引脚，P23不能被用作I/O端口引脚。</p> <p>(3) 高速内部振荡时钟被选择 P23 和 P22 引脚可以被用作 I/O端口引脚。</p>	p. 189 <input type="checkbox"/>
			低速内部振荡	<p>如果选择低速内部振荡器不能被停止，看门狗定时器 (WDT) 的计数时钟固定为低速内部振荡时钟。</p> <p>如果选择低速内部振荡器可由软件停止，WDT的计数时钟的提供在HALT / STOP 模式下被停止，无论低速内部振荡模式寄存器 (LSRCM) 的第0位 (LSRSTOP) 的设置如何。类似的，当低速内部振荡时钟以外的时钟被选作WDT的计数时钟时，时钟提供也被停止。在低速内部振荡器操作 (LSRSTOP = 0) 期间，计数在STOP 模式下，时钟也可以被提供给8位定时器H1。</p>	p. 190 <input type="checkbox"/>
			当RESET引脚用作输入端口引脚 (P34) 时的注意事项	<p>对于一个已经写入的设备，其已经通过选项字节设置为“RESET已经被用作输入端口引脚 (P34)”，当重新擦除 / 写入 (使用专用flash存储器编程器通过板上编程) 该设备时，了解以下事项。在向目标系统提供电源之前，连接专用flash存储器编程器并打开其电源。如果电源事先被提供给目标系统，将不能切换为flash存储器编程模式。</p>	p. 190 <input type="checkbox"/>
			Flash 存储器	<p>PG-FP4 GUI 软件设置值示例</p> <p>安全设置</p> <p>自编程功能</p>	<p>以上是推荐值。依赖于使用环境，这些值可能改变，因此充分评估后设置。</p> <p>在对批擦除的安全设置被设置后，擦除不能被执行。此外，即使写入命令被执行，与已经写入flash存储器的数据不同的数据不能被写入，因为擦除命令无效。</p> <p>自编程处理必须包含在执行自写前的程序中。</p> <p>当自编程命令执行时没有指令能被执行。因此，事先清除和重启看门狗定时计数器以使自编程期间看门狗定时器不溢出。参见表15-11了解自编程执行时间。</p> <p>自编程期间发生的中断在自编程模式结束后可被确认。为避免该操作，当用指定顺序切换正常模式到自编程模式时禁止中断服务 (设置MK0为FFH，并执行DI指令)。</p> <p>当自编程命令执行时不使用RAM。</p> <p>如果flash存储器被写或擦除时供电电压降低或输入复位信号，不能保证写入 / 擦除。</p>
第十五章	软件	Flash 存储器	安全设置	<p>在对批擦除的安全设置被设置后，擦除不能被执行。此外，即使写入命令被执行，与已经写入flash存储器的数据不同的数据不能被写入，因为擦除命令无效。</p>	p. 200 <input type="checkbox"/>
			自编程功能	<p>自编程处理必须包含在执行自写前的程序中。</p> <p>当自编程命令执行时没有指令能被执行。因此，事先清除和重启看门狗定时计数器以使自编程期间看门狗定时器不溢出。参见表15-11了解自编程执行时间。</p> <p>自编程期间发生的中断在自编程模式结束后可被确认。为避免该操作，当用指定顺序切换正常模式到自编程模式时禁止中断服务 (设置MK0为FFH，并执行DI指令)。</p> <p>当自编程命令执行时不使用RAM。</p> <p>如果flash存储器被写或擦除时供电电压降低或输入复位信号，不能保证写入 / 擦除。</p>	p. 201 <input type="checkbox"/>
			自编程功能	<p>当自编程命令执行时没有指令能被执行。因此，事先清除和重启看门狗定时计数器以使自编程期间看门狗定时器不溢出。参见表15-11了解自编程执行时间。</p>	p. 204 <input type="checkbox"/>
			自编程功能	<p>自编程期间发生的中断在自编程模式结束后可被确认。为避免该操作，当用指定顺序切换正常模式到自编程模式时禁止中断服务 (设置MK0为FFH，并执行DI指令)。</p>	p. 204 <input type="checkbox"/>
			自编程功能	<p>当自编程命令执行时不使用RAM。</p> <p>如果flash存储器被写或擦除时供电电压降低或输入复位信号，不能保证写入 / 擦除。</p>	p. 204 <input type="checkbox"/>

章节	分类	功能	功能详述	注意事项	页码
第十五章	软件	Flash 存储器	自编程功能	在block擦除期间空白数据值设为FFH。	p. 204 <input type="checkbox"/>
				预先设置CPU时钟在自编程时为1MHz或更高。	p. 204 <input type="checkbox"/>
				执行指定顺序来设置自编程模式后马上执行自编程NOP和HALT命令，然后执行自编程。此时，HALT指令在10 μ s（最大值）+ 2 CPU 时钟（fCPU）后自动释放。	p. 204 <input type="checkbox"/>
				如果振荡器时钟或外部时钟选作系统时钟，在执行指定顺序设置自编程模式后马上执行NOP和HALT指令，在释放HALT状态后等待8 μ s，然后执行自编程。	p. 204 <input type="checkbox"/>
				用1位存储器操作指令检查FPRERR。	p. 204 <input type="checkbox"/>
				自编程模式下引脚状态和HALT模式下的相同。	p. 204 <input type="checkbox"/>
				因为在自编程模式下通过板上 / 板外编程设置安全功能被禁止，自编程命令可被执行，不管安全功能的设置。为禁止自编程期间写入或擦除处理，设置保护字节。	p. 204 <input type="checkbox"/>
				确保在执行自编程命令前清除flash地址指针H（FLAPH）的第4到7位和flash地址指针H比较寄存器（FLAPHC）为0。如果设置这些位为1来执行自编程，设备可能故障。	p. 204 <input type="checkbox"/>
			在设置到自编程模式和正常模式前马上清除FLCMD寄存器值为00H。	p. 204 <input type="checkbox"/>	
			FLPMC: Flash 编程模式控制寄存器	设置自编程模式情况下的注意事项，参见15.8.2 自编程功能注意事项。	p. 205 <input type="checkbox"/>
				预先设置CPU时钟在自编程时为1MHz或更高。	p. 205 <input type="checkbox"/>
				执行指定顺序来设置自编程模式后马上执行自编程NOP和HALT命令，然后执行自编程。此时，HALT指令在10 μ s（最大值）+ 2 CPU 时钟（fCPU）后自动释放。	p. 205 <input type="checkbox"/>
				如果振荡器时钟或外部时钟选作系统时钟，在执行指定顺序设置自编程模式后马上执行NOP和HALT指令，在释放HALT状态后等待8 μ s，然后执行自编程。	p. 205 <input type="checkbox"/>
			在设置到自编程模式和正常模式前马上清除FLCMD寄存器值为00H。	p. 205 <input type="checkbox"/>	
			PFCMD: Flash 保护命令寄存器	自编程模式下中断服务不能执行。禁止中断服务（通过在MK0=FFH时执行DI指令）在执行指定顺序设置自编程模式前和执行指定顺序改变模式到正常模式后。	p. 206 <input type="checkbox"/>
			PFS: Flash 状态寄存器	用1位存储器操作指令检查FPRERR。	p. 206 <input type="checkbox"/>
			FLAPH, FLAPL: Flash 地址指针 H和L	确保在执行自编程命令前清除flash地址指针H（FLAPH）的第4到7位和flash地址指针H比较寄存器（FLAPHC）为0。当执行自编程命令时这些位的值为1。	p. 209 <input type="checkbox"/>

章节	分类	功能	功能详述	注意事项	页码
第十五章	软件	Flash 存储器	FLAPHC, FLAPLC: Flash 地址指针 H / L 比较寄存器	在执行自编程命令前确保清除FLAPH和FLAPHC的第4到7位为0。当执行自编程命令时这些位的值为1。	p. 209 <input type="checkbox"/>
				设置被block擦除, 校验, 或空白检查的block数 (和FLAPH值相同) 到FLAPHC。	p. 209 <input type="checkbox"/>
				当执行block擦除时清除FLAPLC到00H, 并且当执行空白检查时设置该寄存器为FFH。	p. 209 <input type="checkbox"/>
			切换到自编程模式	确保用在数据没有擦除或写入的地址处的用户程序执行上述的一系列操作。	pp.211, <input type="checkbox"/> 212, 214, 215
			切换到正常模式		
			字节写	如果写导致失败, 立即擦除block并再写。	p. 223 <input type="checkbox"/>
第十七章	硬件	电气规范	-	这些特性显示了产品的目标值, 目标值在设备评估后可能会发生变化。工作电压范围可能也会改变。	p. 258 <input type="checkbox"/>
			最大绝对额定值	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。	p. 258 <input type="checkbox"/>
			X1 振荡器特性	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。 <ul style="list-style-type: none"> • 连接线越短越好。 • 连接线不应与其他信号线交叉。 • 流经的电流变化较大的信号线不要在振荡器周围布线。 • 要保持振荡器电容器的接地点电压与Vss相同。 • 不要将电容的地信号接入大电流地。 • 不要从振荡器获取信号。 	p. 259 <input type="checkbox"/>
第十九章	硬件	推荐焊接条件	无铅产品	产品代号最后带-A的产品是无铅产品。	p. 270 <input type="checkbox"/>
			-	对于与以下所推荐不同的焊接方法和条件, 请与日电电子的销售代表联系。	p. 270 <input type="checkbox"/>
			-	不要使用不同的焊接方法 (局部加热除外)。	p. 270 <input type="checkbox"/>

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