

To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
 - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
 - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
 - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.



用户手册

μ PD78F9500, 78F9501, 78F9502

8 位单片微控制器

μ PD78F9500

μ PD78F9501

μ PD78F9502

文档编号. U18681CA2V0UD00 (第二版)
发行日期 2009 年 1 月 NS

© 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。

- 本文档所登载的内容有效期截止至 2009 年 1 月，信息先于产品的生产周期发布。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。否则因本文件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 PD78F9500, 78F9501, 78F9502 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 PD78F9500, 78F9501, 78F9502: 有两本手册: 本手册和指令手册(78K/0S 系列通用)。

μ PD78F9500, 78F9501,
78F9502
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 PD78F9500, 78F9501, 78F9502 全部功能
 - 请按**目录**顺序阅读本手册 标识 <R> 表示主要修改点.可以在 PDF 文件中通过查找“<R>”来轻松查看更新处, 并制定它的寻找区域。
- ◇ 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 PD78F9500, 78F9501, 78F9502 的电气特性
 - 请参阅 **第十七章 电气特性**。

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 xxxx**B**
 十进制 ... xxxx
 十六进制 ... xxxx**H**

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
μPD78F9500, 78F9501, 78F9502 用户手册	本手册
78K0S 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-QB 集成调试器 V2.81	操作篇	U17287E
PM+ V5.20		

开发硬件工具相关文档 (用户手册)

文档名称	文档编号
QB-78K0SKX1 在线仿真器	U18219E
QB-MINI2 带有编程功能的片上调试仿真器	U18371E

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-FP5 Flash 存储器编程器用户手册	U18865E

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	13
1.1	特点	13
1.2	订货信息	14
1.3	引脚配置 (俯视图).....	15
1.4	78K0S/Kx1+ 产品线.....	16
1.5	框图	17
1.6	各产品功能列表	18
第二章	引脚功能	19
2.1	引脚功能列表.....	19
2.2	引脚功能	20
2.2.1	P20 至 P23 (端口 2).....	20
2.2.2	P32 和 P34 (端口 3).....	21
2.2.3	P40 和 P43 (端口 4).....	21
2.2.4	RESET	21
2.2.5	VDD	21
2.2.6	VSS	21
2.3	引脚 I/O 电路及不使用的引脚连接	22
第三章	CPU 结构	23
3.1	存储空间	23
3.1.1	内部程序存储空间	26
3.1.2	内部数据存储空间	27
3.1.3	特殊功能寄存器(SFR)存储区	27
3.1.4	数据存储寄存器寻址	27
3.2	处理器寄存器.....	30
3.2.1	控制寄存器.....	30
3.2.2	通用寄存器.....	33
3.2.3	特殊功能寄存器 (SFR).....	34
3.3	指令地址寻址.....	38
3.3.1	相对寻址	38
3.3.2	立即寻址	39
3.3.3	表间接寻址.....	39
3.3.4	寄存器寻址.....	40
3.4	操作数地址寻址	41
3.4.1	直接寻址	41
3.4.2	短直接寻址.....	42
3.4.3	特殊功能寄存器 (SFR)寻址.....	43
3.4.4	寄存器寻址.....	44
3.4.5	寄存器间接寻址	45
3.4.6	基址寻址	46

3.4.7	堆栈寻址.....	47
第四章	端口功能.....	48
4.1	端口功能.....	48
4.2	端口配置.....	49
4.2.1	端口 2.....	49
4.2.2	端口 3.....	53
4.2.3	端口 4.....	54
4.3	控制端口功能寄存器.....	55
4.4	端口功能操作.....	59
4.4.1	写入 I/O 端口.....	59
4.4.2	从 I/O 端口读出.....	59
4.4.3	I/O 端口的操作.....	59
第五章	时钟发生器.....	60
5.1	时钟发生器的功能.....	60
5.1.1	系统时钟振荡器.....	60
5.1.2	用作时间间隔发生器的时钟振荡器.....	60
5.2	时钟发生器的格式.....	61
5.3	时钟发生器控制寄存器.....	63
5.4	系统时钟振荡器.....	65
5.4.1	高速内部振荡.....	65
5.4.2	外部时钟输入电路.....	65
5.4.3	预分频器.....	65
5.5	CPU 时钟发生器的操作.....	66
5.6	向外围设备提供时钟的时钟发生器的操作.....	70
第六章	8 位定时器 H1.....	72
6.1	8 位定时器 H1 功能.....	72
6.2	8 位定时器 H1 的构造.....	72
6.3	8 位定时器 H1 的控制寄存器.....	75
6.4	8 位定时器 H1 操作.....	77
6.4.1	间隔定时/方波输出模式下的操作.....	77
6.4.2	PWM 输出模式下的操作.....	81
第七章	看门狗定时器.....	87
7.1	看门狗定时器的功能.....	87
7.2	看门狗配置.....	89
7.3	看门狗定时器控制寄存器.....	90
7.4	看门狗定时器的工作.....	92
7.4.1	当通过选项字节选择“低速内部振荡器无法停止”时看门狗定时器的工作.....	92
7.4.2	当通过选项字节选择“低速内部振荡器可被停止”时看门狗定时器的工作.....	94
7.4.3	STOP 模式下看门狗定时器的工作（当通过选项字节选择“低速内部振荡器可被停止”时）.....	96
7.4.4	HALT 模式下看门狗定时器的工作（当通过选项字节选择“低速内部振荡器可被停止”时）.....	97

第八章 中断功能	98
8.1 中断功能类型.....	98
8.2 中断源及其构造.....	98
8.3 中断功能控制寄存器.....	100
8.4 中断服务工作原理.....	102
8.4.1 可屏蔽中断请求的识别原理.....	102
8.4.2 多重中断服务.....	104
8.4.3 中断请求挂起.....	106
第九章 待机功能	107
9.1 待机功能及其配置.....	107
9.1.1 待机功能.....	107
9.2 待机功能的工作原理.....	109
9.2.1 HALT 模式.....	109
9.2.2 STOP 模式.....	111
第十章 复位功能	114
10.1 复位源寄存器.....	118
第十一章 上电清零电路	119
11.1 上电清零电路的功能.....	119
11.2 上电清零电路的构造.....	120
11.3 上电清零电路的工作原理.....	120
11.4 使用上电复位电路的注意事项.....	121
第十二章 低电压检测器	123
12.1 低电压检测器的功能.....	123
12.2 低电压检测器的构造.....	123
12.3 低电压检测器控制寄存器.....	124
12.4 低电压检测器的工作.....	126
12.5 使用低电压检测器的注意事项.....	130
第十三章 选项字节	133
13.1 选项字节的功能.....	133
13.2 选项字节的格式.....	134
13.3 当 RESET 引脚用作单向输入端口 (P34) 时的注意事项.....	135
第十四章 Flash 存储器	136
14.1 特性.....	136
14.2 内存配置.....	137
14.3 功能概述.....	137
14.4 通过 Flash 存储器编程器写入.....	138
14.5 编程环境.....	139
14.6 板上引脚的处理.....	140

14.6.1	EXCLK 引脚	140
14.6.2	RESET 引脚	141
14.6.3	端口引脚	141
14.6.4	电源	141
14.7	板上和离板 Flash 存储器编程	142
14.7.1	Flash 存储器编程模式	142
14.7.2	通信命令	142
14.7.3	安全设置	143
14.8	自编程模式下的 Flash 存储器编程	144
14.8.1	自编程概述	144
14.8.2	自编程功能的注意事项	147
14.8.3	用于自编程功能的寄存器	147
14.8.4	正常模式切换到自编程模式的实例	154
14.8.5	自编程模式切换到正常模式的实例	157
14.8.6	自编程模式下块擦除操作的实例	160
14.8.7	自编程模式下空白块检测操作的实例	163
14.8.8	自编程模式下字节写入操作的实例	166
14.8.9	自编程模式下内部验证操作的实例	169
14.8.10	自编程模式下应最小化命令执行时间的操作实例	173
14.8.11	自编程模式下应最短中断-禁用时间时的操作实例	179
第十五章	片上调试功能	190
15.1	将 QB-MINI2 连接到 PD78F9500、78F9501、78F9502	190
15.1.1	INTP1 引脚的连接	191
15.1.2	EXCLK 和 P22 引脚的连接	192
15.2	用户资源安全	193
第十六章	指令集综述	194
16.1	操作	194
16.1.1	操作数识别符和描述方法	194
16.1.2	“运算” 栏的描述	195
16.1.3	“标志” 栏的描述	195
16.2	操作清单	196
16.3	按地址类型所列指令	201
第十七章	电气特性	204
第十八章	封装图	214
第十九章	建议焊接条件	215
附录 A	开发工具	216
A.1	软件包	219

A.2	编程语言处理软件	219
A.3	控制软件.....	220
A.4	闪存写入工具.....	220
A.5	调试工具 (硬件)	221
A.5.1	当使用在线仿真器 QB-78K0SKX1 时	221
A.5.2	当使用内置调试仿真器 QB-MINI2 时.....	221
A.6	调试工具 (软件)	222
附录 B	目标系统设计注意事项.....	223
附录 C	寄存器索引	225
C.1	寄存器索引(寄存器名称).....	225
C.2	寄存器索引(符号).....	226
附录 D	注意事项列表	228
附录 E	修订记录.....	235
E.1	本版本主要修订之处.....	235

第一章 概述

1.1 特点

○ 78K0S CPU 内核

○ ROM 和 RAM 容量

目 产品编号	程序存储器 (闪存)	存储器 (内部高速 RAM)
μPD78F9500	1 KB	128 个字节
μPD78F9501	2 KB	
μPD78F9502	4 KB	

○ 最小指令执行时间: 0.2 μs (10 MHz 时钟频率, 工作电压在 4.0 至 5.5 V 之间)

○ 时钟

• 高速系统时钟 ... 从以下两种时钟源中选择

- 外部时钟: 2 至 10 MHz
- 高速内部振荡器: 8 MHz ±2% (-10 至 +85°C), 8 MHz ±5% (-40 至 +85°C)

• 低速内部振荡器 240 kHz (典型值.) ... 看门狗定时器, 间断工作的定时器时钟

○ I/O 端口: 8 (CMOS I/O: 7, CMOS 输入端口: 1)

○ 定时器: 2 通道

- 8 位定时器: 1 通道 ... PWM 输出 × 1
- 看门狗定时器: 1 通道 ... 可运行于内部低速振荡时钟之下

○ 内置上电清零电路(POC) (当电压掉至 2.1 V ±0.1V 或以下时, 会自动产生复位信号)

○ 内置低电压检测电路(LVI) (当供电电压值达到低电压检测电压值时, 该电路便产生中断或复位(可选)信号)

- 检测电压值: 2.35 至 4.3 V 之间可选, 共十个检测电压等级可选

○ 单电源供电闪存

- 闪存自编程功能允许
- 软件保护功能: 禁止外部拷贝 (无闪存读命令)
- 由专用闪存编程器写闪存所需要的时间: 大约 3 秒 (4 KB)
- * 支持量产 Flash 编程

○ 安全功能

- 看门狗定时器有独立于 CPU 的时钟支持运行
... 即便系统时钟停止工作, 仍能检测到程序异常
- 由 LVI 进行供电电压掉电检测
... 在供电电压掉至工作电压之下之前, 系统会执行适当的处理程序
- 配备选项字节功能
... 在硬件中设置的重要系统运行设置

○ 支持汇编和 C 两种编程语言

○ 增强型开发环境

- 支持全功能仿真器 (IECUBE), 简易仿真器 (MINICUBE2), 及软件仿真器^註

注 用于 μPD78F9500, 78F9501, 78F9502 的软件仿真器正在开发中

○ 供电电压: $V_{DD} = 2.0$ 至 5.5 V

* 该产品应在 2.2 至 5.5 V 电压范围内使用, 因为上电清零电路(POC)的检测电压(V_{Poc})为 2.1 V ± 0.1 V。

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

1.2 订货信息

产品号

μPD78F9 xxx - xx (x) - xxx -A



[产品号列表]

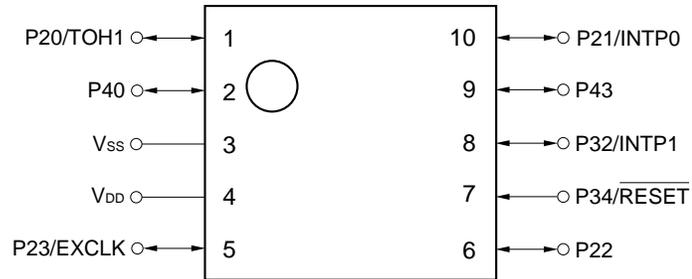
$\mu\text{PD78F9500MA-CAC-A}$

$\mu\text{PD78F9501MA-CAC-A}$

$\mu\text{PD78F9502MA-CAC-A}$

1.3 引脚配置 (俯视图)

10 引脚 塑料 SSOP



INTP0, INTP1: 外部中断输入

P20 至 P23: 端口 2

P30, P34: 端口 3

P40, P43: 端口 4

 $\overline{\text{RESET}}$: 复位

TOH1: 定时器输出

V_{DD}: 电源V_{ss}: 地EXCLK: 外部时钟输入
(主系统时钟)

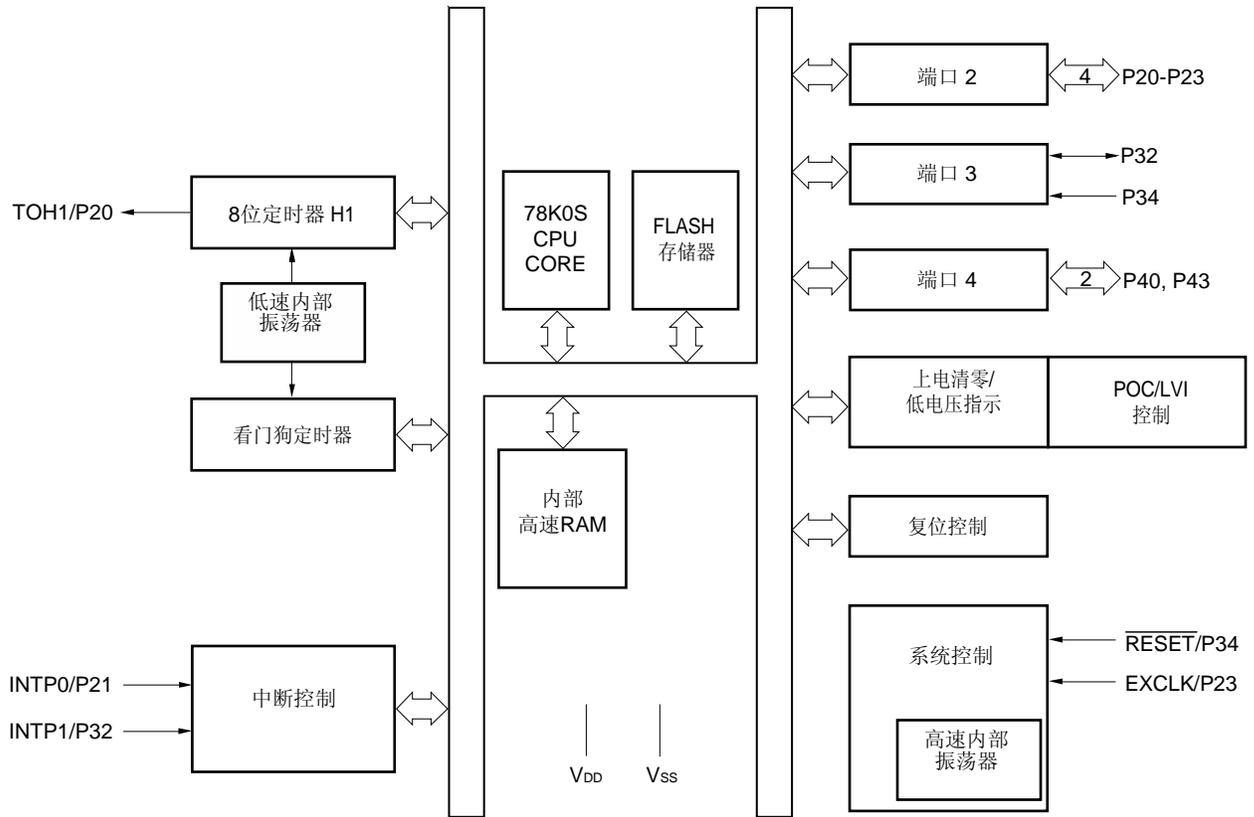
1.4 78K0S/Kx1+ 产品线

下表所列为 78K0S/Kx1+ 产品线。

产品号		78K0S/KU1+	78K0S/KY1+	78K0S/KA1+	78K0S/KB1+
条目					
引脚数目		10 引脚	16 引脚	20 引脚	30/32 引脚
内部存储器	闪存	1 KB, 2 KB, 4 KB		2 KB	4 KB, 8 KB
	RAM	128 字节		128 字节	256 字节
供电电压		$V_{DD} = 2.0$ 至 5.5 V ^{注 1}			
最小指令执行时间		$0.20 \mu\text{s}$ (10 MHz, $V_{DD} = 4.0$ 至 5.5 V) $0.33 \mu\text{s}$ (6 MHz, $V_{DD} = 3.0$ 至 5.5 V) $0.40 \mu\text{s}$ (5 MHz, $V_{DD} = 2.7$ 至 5.5 V) $1.0 \mu\text{s}$ (2 MHz, $V_{DD} = 2.0$ 至 5.5 V)			
系统时钟 (振荡频率)		高速内部振荡器 (8 MHz (典型值)) 晶体/陶瓷 振荡器 (2 至 10 MHz) ^{注 2} 外部时钟振荡器 (2 至 10 MHz)			
用于 TMH1 和 WDT 的时钟 (振荡频率)		低速内部振荡器 (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 ^{注 3}			
	8 位 (TMH)	1 通道			
	8 位 (TM8)	-	-	1 通道	
	WDT	1 通道			
串行接口		-	-	支持 LIN 总线的 UART: 1 通道	
A/D 转换器 ^{注 4}		10 位: 4 通道 (2.7 至 5.5 V) ^{注 4}			
乘法器 (8 位 × 8 位)		-	-	-	提供
中断	内部中断	5 ^{注 5}		9	
	外部中断	2		4	
复位	RESET 引脚	提供			
	POC	2.1 V (典型值.)			
	LVI	提供 (通过软件选择)			
	WDT	提供			
工作温度范围		标准产品: -40 至 +85°C	标准产品, (A) 级产品: -40 至 +85°C (A2) 级产品: -40 至 +125°C		

- 注 s**
1. 这些产品应在如下电压范围内使用, 因为上电清零电路(POC)的检测电压(V_{Poc})为供电电源范围。
标准产品, (A) 级产品: 2.2 至 5.5 V, (A2) 级产品: 2.26 至 5.5 V
 2. 该产品 ($\mu\text{PD78F950x}$) 不支持 晶体/陶瓷 振荡器
 3. 78K0S/KU1+中不带有 A/D 转换器 ($\mu\text{PD78F950x}$)的产品没有 16 位定时器(TM0)。
 4. 分别为 78K0S/KU1+和 78K0S/KY1+提供这种不含 A/D 转换器 ($\mu\text{PD78F95xx}$)产品。该产品没有 A/D 转换器。
 5. 在 78K0S/KU1+中, 这种不带 A/D 转换器的产品有 2 个系数, 而在 78K0S/KY1+, 这种不带 A/D 转换器的产品有 4 个系数。

1.5 框图



1.6 各产品功能列表

条目		μ PD78F9500	μ PD78F9501	μ PD78F9502
内部存储器	闪存	1 KB	2 KB	4 KB
	高速 RAM	128 个字节		
存储空间		64 KB		
时钟	高速系统时钟	外部时钟输入: 10 MHz ($V_{DD} = 2.7$ 至 5.5 V)		
		内部高速振荡时钟		
	内部低速振荡时钟		240 kHz (典型值.)	
通用寄存器		8 位 \times 8 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 (高速系统时钟: $f_x = 10$ MHz)		
I/O 端口		总数 I: 8 引脚 CMOS I/O: 7 引脚 CMOS 输入: 1 引脚		
定时器		<ul style="list-style-type: none"> • 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)		
中断矢量源	外部中断	2		
	内部中断	2		
复位		<ul style="list-style-type: none"> • 由 RESET 引脚复位 • 由看门狗定时器进行内部复位 • 由上电清零电路进行内部复位 • 由低电压检测电路进行内部复位 		
供电电压		$V_{DD} = 2.0$ 至 5.5 V ^{**}		
工作温度范围		$T_A = -40$ 至 $+85^\circ\text{C}$		
封装类型		10-引脚 塑料 SSOP		

注 该产品应在 2.2 至 5.5 V 电压范围内使用, 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（功能状态）	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 通过设置软件，可以连接一个内置上拉电阻		输入端口	TOH1
P21					INTP0
P22					—
P23 [‡]					EXCLK [‡]
P32	I/O	端口 3 通过设置软件，可以连接一个内置上拉电阻	可以按位设置为输入或输出模式	输入端口	INTP1
P34 [‡]			仅用于输入功能模式		
P40, P43	I/O	端口 4 2 位 I/O 端口 可以按位设置为输入或输出模式 通过设置软件，可以连接一个内置上拉电阻		输入端口	—

注 需要了解更多有关引脚功能设置方法的信息，请参阅 第 13 章 选项字节。

注意事项 复位期间，P22 引脚和 P23/EXCLK 引脚电平被强制下拉。而 P34/RESET 引脚电平在复位期间，由复位功能引脚/上电清零电路强制上拉。

(2) 非端口引脚

引脚名称	I/O	功能		复位后（功能状态）	复用功能引脚
INTP0	输入	外部中断输入端，可指定有效边沿触发（上升沿，下降沿，或者同时指定为上升沿和下降沿）		输入端口	P21
INTP1					P32
TOH1	输出	8 位定时器 H1 输出端		输入端口	P20
RESET [‡]	输入	系统复位输入端		输入端口	P34 [‡]
EXCLK [‡]	输入	用于主系统时钟的外部时钟输入端		输入端口	P23 [‡]
V _{DD}	—	正供电电源		—	—
V _{SS}	—	地（电位）		—	—

注 需要了解更多有关引脚功能设置方法的信息，请参阅 第十三章 选项字节。

注意事项 复位期间，P22 引脚和 P23/EXCLK 引脚电平被强制下拉。而 P34/RESET 引脚电平在复位期间，由复位功能引脚/上电清零电路强制上拉。

2.2 引脚功能

2.2.1 P20 至 P23 (端口 2)

P20 至 P23 组成 4 位 I/O 端口。除了用作 I/O 端口引脚功能之外，这些引脚还可用于输出定时器信号，输入外部中断请求信号，以及输入主系统时钟的外部时钟信号。

P23 也用作 EXCLK。需要了解更多有关引脚功能设置方法的信息，请参阅 **第十三章 选项字节**。
这些引脚可以按位设置为如下操作模式。

(1) 端口模式

P20 至 P23 用作为 4 位 I/O 端口。该端口的每一位均可通过端口模式寄存器 2 (PM2) 设置为输入或输出模式，此外，可以通过上拉电阻选择寄存器 2 (PU2)，将内置上拉电阻连接到该端口。

(2) 端口控制模式

P20 至 P23 的功能是输出一定时器信号，以及输入一个外部中断请求信号。

(a) TOH1

该引脚输出一个 8 位定时器 H1 信号。

(b) INTP0

该引脚为一外部中断请求信号输入端，可以指定为相应的有效边沿触发（上升沿，下降沿，或者同时指定为上升沿和下降沿）

(c) EXCLK

该引脚为一输入引脚，为主系统时钟的外部时钟输入端。

注意事项 复位期间，P22 引脚和 P23/EXCLK 引脚电平被强制下拉。

2.2.2 P32 和 P34 (端口 3)

P32 为 1 位 I/O 端口。除了用作 I/O 端口引脚功能之外, 该引脚还用于输入外部中断请求信号。

P34 为 1 位仅输入端口。该引脚也用作 $\overline{\text{RESET}}$ 引脚, 电源上电时, 该引脚用作复位功能引脚。

需要了解更多有关引脚功能设置方法的信息, 请参阅 **第 13 章 选项字节**。

当 P34 用作输入端口引脚时, 要接上拉电阻。

P32 和 P34 引脚可以按位设置为如下操作模式。

(1) 端口模式

P32 用作 1 位 I/O 端口。该端口可以通过端口模式寄存器 3 (PM3) 设置为输入模式或输出模式, 此外, 还可以通过使用上拉电阻选择寄存器 3 (PU3), 将内置上拉电阻连接到该端口。

P34 用作 1 位只输入端口。

(2) 端口控制模式

P32 用作外部中断请求信号输入引脚(INTP1), 可以指定为相应的有效边沿触发(上升沿, 下降沿, 或者同时指定为上升沿和下降沿)。

注意事项 而 $\overline{\text{P34/RESET}}$ 引脚电平在复位期间, 由复位功能引脚/上电清零电路强制上拉。

2.2.3 P40 和 P43 (端口 4)

P40 和 P43 构成一 4 位 I/O 端口。该端口的每一位均可通过端口模式寄存器 4 (PM4) 设置为输入模式或输出模式, 此外, 还可以通过使用上拉电阻选择寄存器 4 (PU4), 接入内置上拉电阻。

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

该引脚输入一低有效系统复位信号。当电源上电时, 不论选项字节如何设置, 该引脚均用作复位功能引脚。

注意事项 而 $\overline{\text{P34/RESET}}$ 引脚电平在复位期间, 由复位功能引脚/上电清零电路强制上拉。

2.2.5 V_{DD}

该引脚为正供电电源输入引脚。

2.2.6 V_{SS}

为接地引脚。

请确保将 V_{SS} 连接至稳定的 GND (= 0 V)。

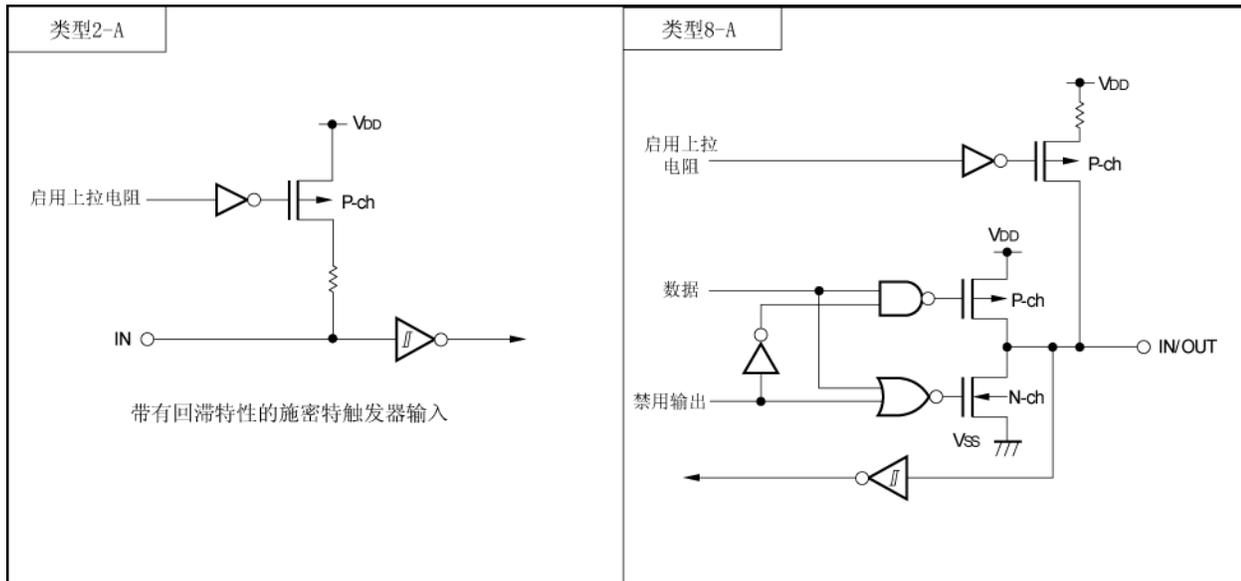
2.3 引脚 I/O 电路及不使用的引脚连接

表 2-1 列出了各引脚的 I/O 电路类型及不使用引脚的建议连接方式。
需要了解有关每个 I/O 电路类型结构，请参见 图 2-1。

表 2-1. 引脚 I/O 电路类型及不使用的引脚连接

引脚名称	I/O 电路类型	I/O	不使用的引脚建议连接方式
P20/TOH1	8-A	I/O	输入： 分别经电阻连接到 VDD 端或 VSS 端。 输出： 悬空。
P21/INTP0			
P22			
P23/EXCLK			
P32/INTP1			
P34/RESET	2-A	输入	在选项字节中将 ENPU34 设置为“1”，并将该引脚悬空。
P40 及 P43	8-A	I/O	输入： 分别经电阻连接到 VDD 端或 VSS 端。 输出： 悬空。

图 2-1. 引脚 I/O 电路

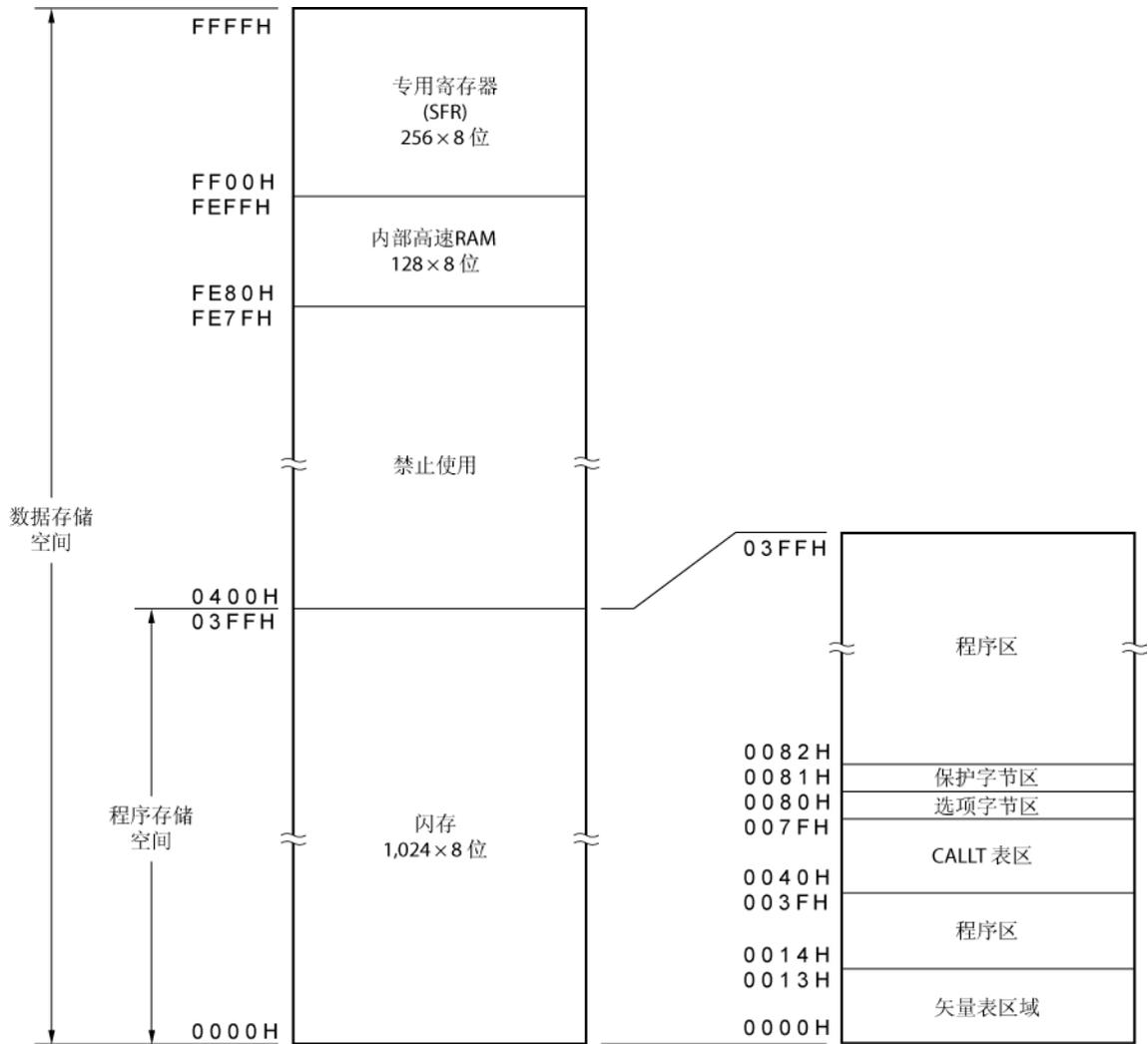


第三章 CPU结构

3.1 存储空间

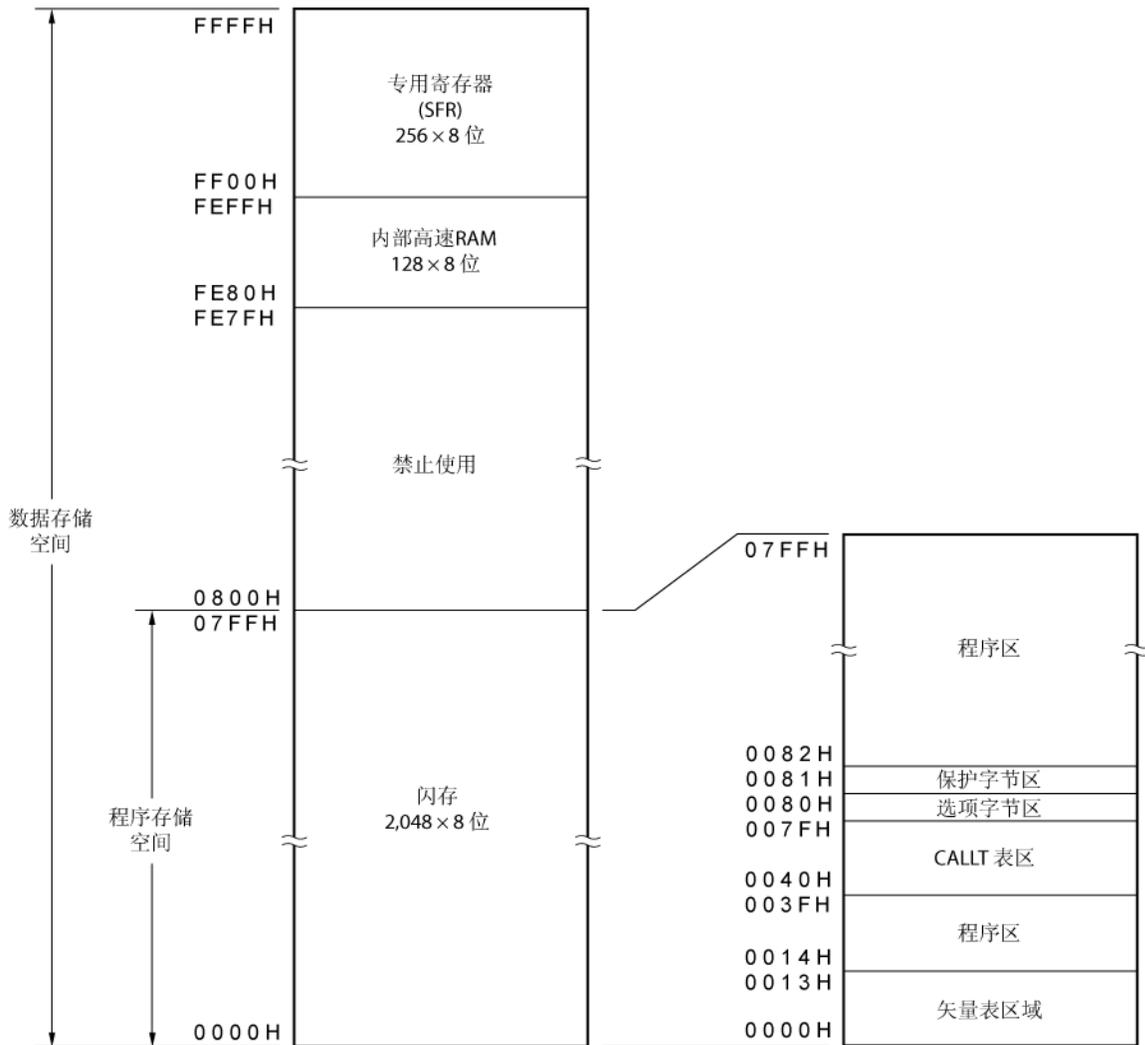
μ PD78F9500, 78F9501, 78F9502 可以访问 64 KB 存储空间。图 3-1 至 3-3 为存储器映射图。

图 3-1. 存储器映射 (μ PD78F9500)



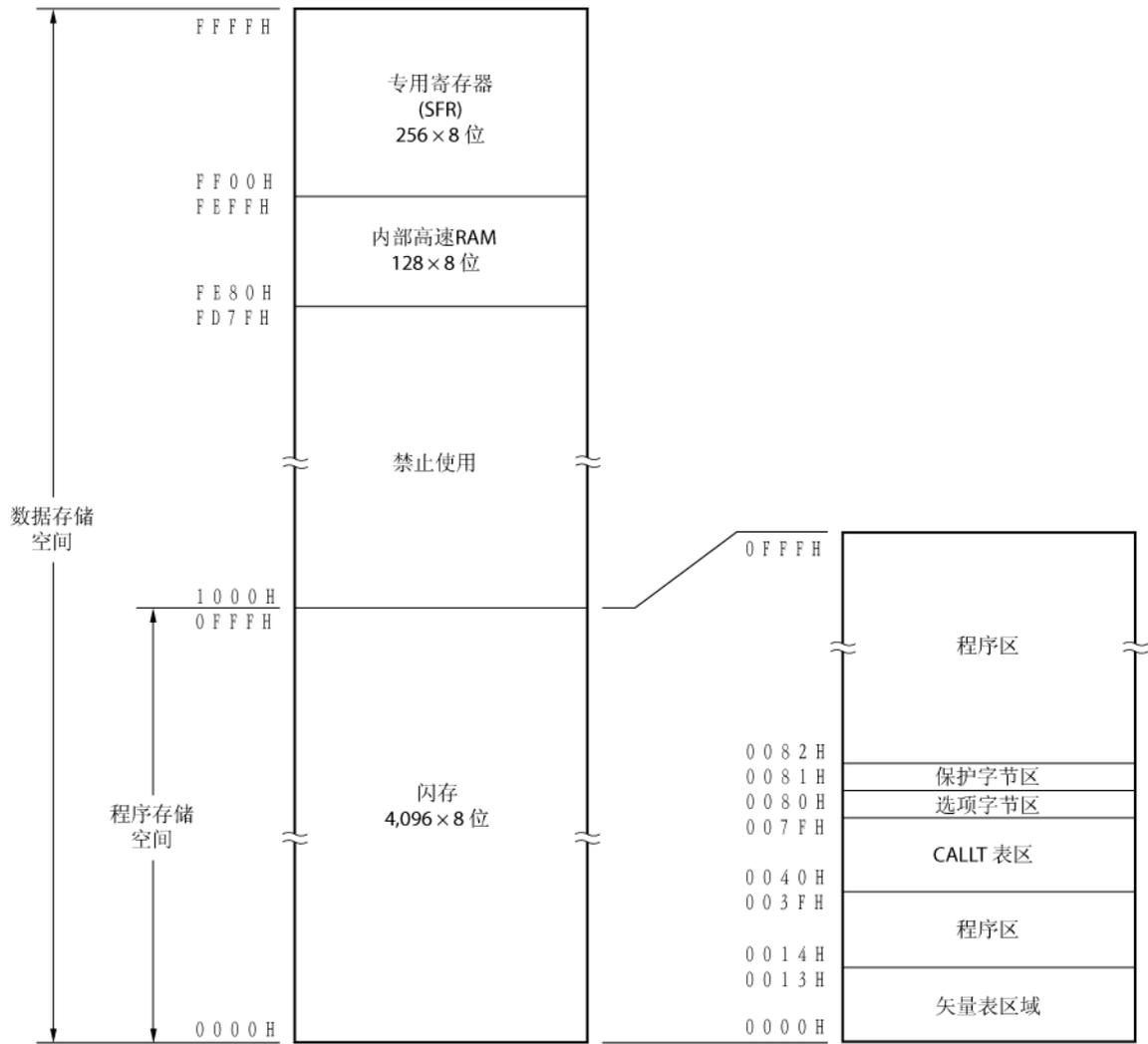
备注 选项字节和保护字节均为一个字节长度。

图 3-2. 存储器映射 (μ PD78F9501)



备注 选项字节和保护字节均为一个字节长度。

图 3-3. 存储器映射 (μ PD78F9502)



备注 选项字节和保护字节均为一个字节长度。

3.1.1 内部程序存储空间

内部程序存储空间存储程序和表数据，通常由程序计数器(PC)进行寻址。

μ PD78F9500，78F9501，78F9502 配置了如下存储容量的内部 ROM (或闪存)。

表 3-1. 内部 ROM 容量

产品编号	内部 ROM	
	结构	容量
μ PD78F9500	闪存	1,024 × 8 位
μ PD78F9501		2,048 × 8 位
μ PD78F9502		4,096 × 8 位

以下存储区域分配至内部程序存储空间。

(1) 矢量表存储区域

从地址 0000H 至 0013H 这 20 个字节存储区域保留用作矢量表存储区域。该存储区存储分支（或跳转）程序的起始地址（由 RESET 或中断请求而引起的）。对于 16 地址来说，低 8 位地址存储在偶地址中，而高 8 位地址存储在奇地址中。

表 3-2. 矢量表

矢量表地址	中断请求
0000H	Reset
0006H	INTLVI
0008H	INTPO
000AH	INTP1
000CH	INTTMH1

(2) CALLT 指令表存储区

该 1 字节调用指令(CALLT)子程序入口地址可以存储在 0040H 至 007FH 地址范围的 64 字节存储区中。

(3) 选项字节存储区

该选项字节为一个地址为 0080H 的 1 字节存储区。需要了解详细情况，请参阅 **第十三章 选项字节**。

(4) 保护字节存储区

该保护字节为一个地址为 0081H 的 1 字节存储区。需要了解详细情况，请参阅 **第十四章 闪存**。

3.1.2 内部数据存储空间

μ PD78F9500, 78F9501, 78F9502 分别配置了 128 字节的内部高速 RAM。

该内部高速 RAM 也可用作 堆栈存储区。

3.1.3 特殊功能寄存器(SFR)存储区

内置外围硬件的特殊功能寄存器 (SFRs) 分配的地址空间为 FF00H 至 FFFFH (见表 3-3)。

3.1.4 数据存储器寻址

μ PD78F9500, 78F9501, 78F9502 提供大范围寻址模式以尽可能提高存储器操作效率。FE80H 至 FEFFH 存储区域包含数据存储器区和特殊功能寄存器(SFR)区, 这些存储区可以由相应的专门寻址方式进行访问。图 3-4 至 3-6 说明了数据存储器寻址情况。

图 3-4. 数据存储器寻址 (μ PD78F9500)

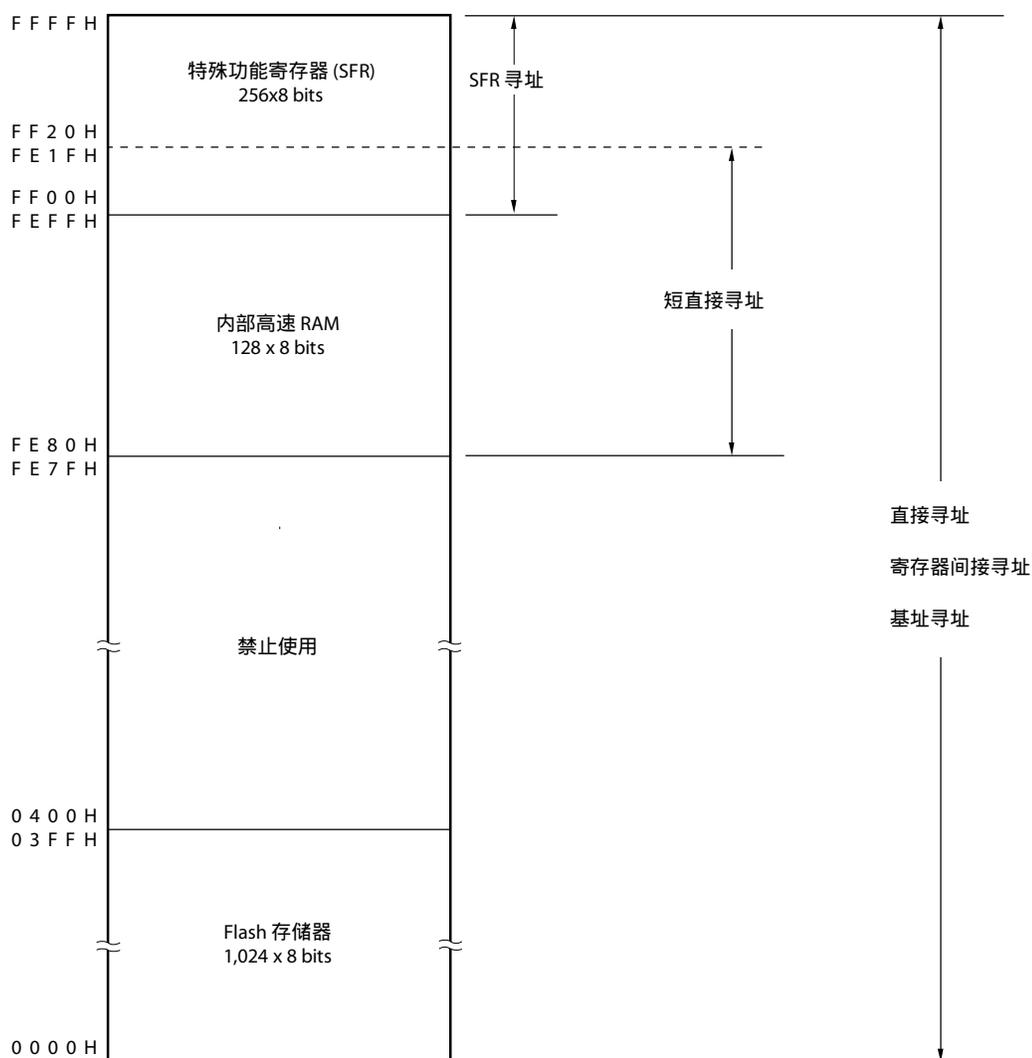


图 3-5. 数据存储地址 (μ PD78F9501)

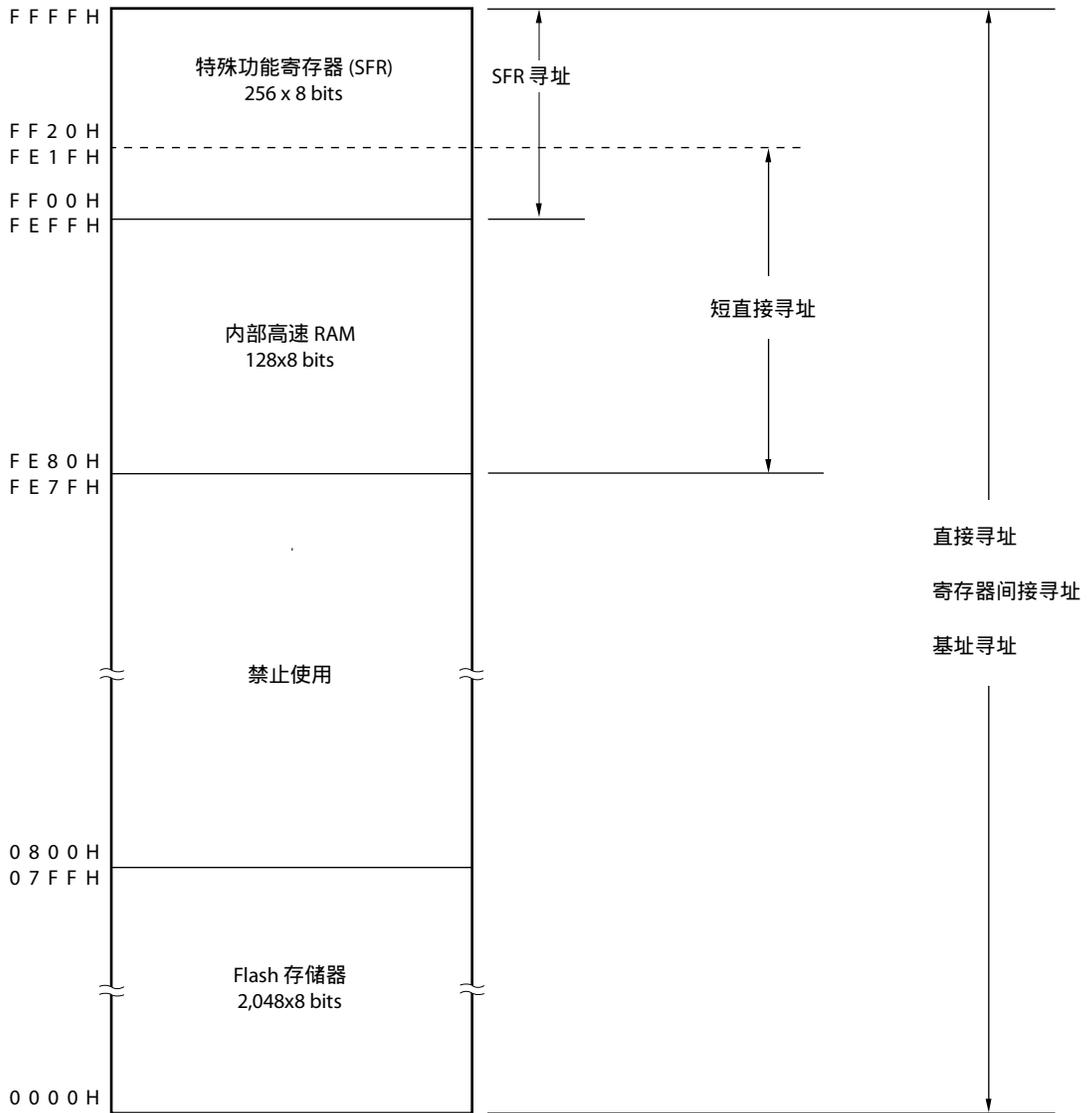
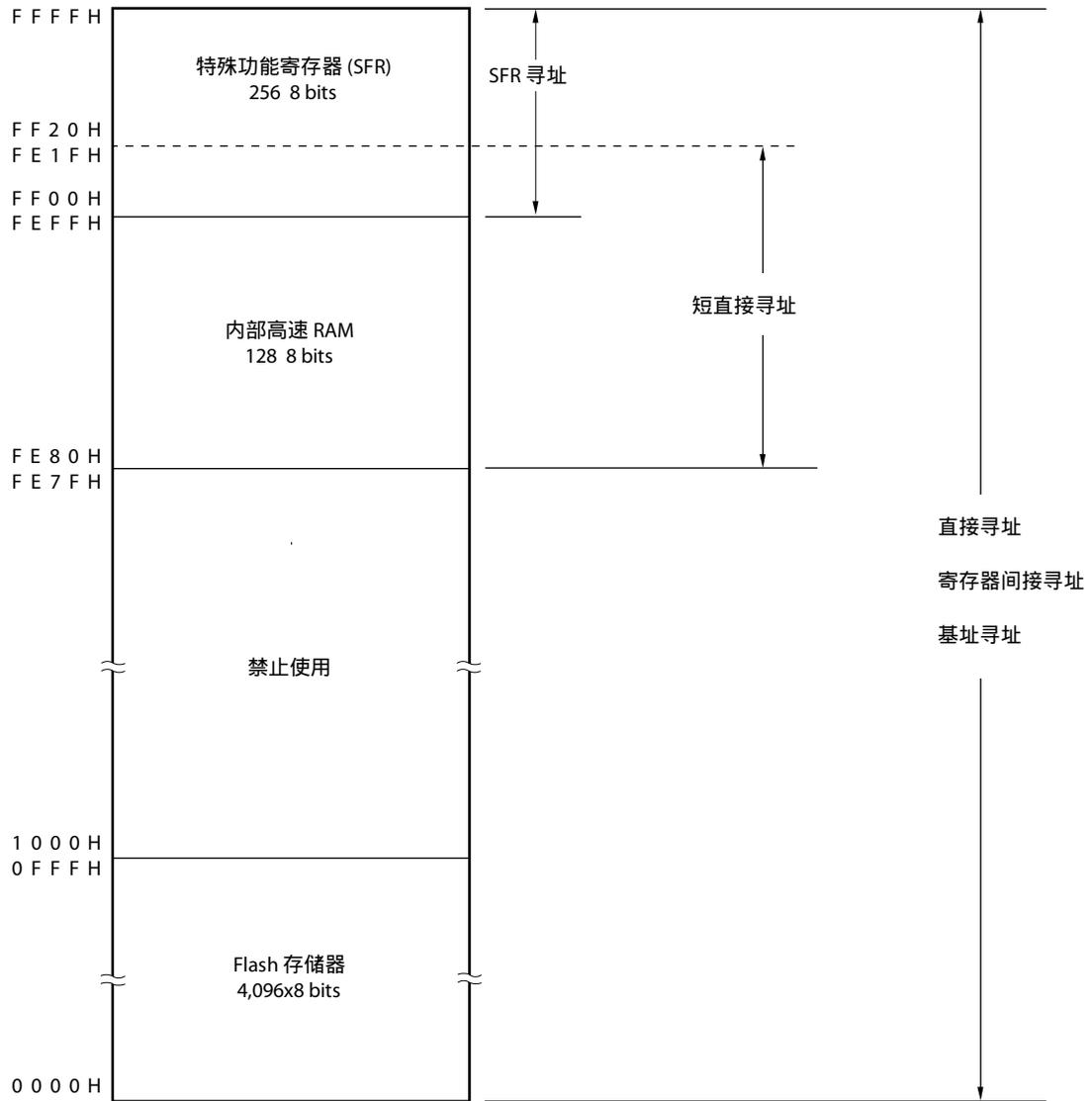


图 3-6. 数据存储器寻址 (μ PD78F9502)



3.2 处理器寄存器

μ PD78F9500, 78F9501, 78F9502 包含如下几种内置处理器寄存器。

3.2.1 控制寄存器

控制寄存器具有控制程序执行的顺序状态及堆栈空间的特殊功能，包括一个程序计数器，一个程序状态字及一个堆栈指针。

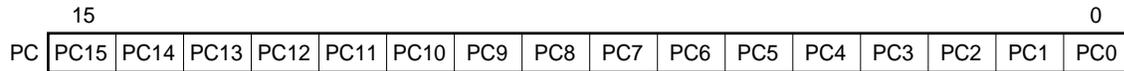
(1) 程序计数器(PC)

程序计数器是一个 16 位的寄存器，用于存放下一条即将执行的指令的地址。

在正常情况下，根据获取指令的字节数，程序计数器 PC 的值会自动累加。当执行分支指令时，则设置立即数或寄存器内容。

RESET 输入将复位向量表的地址为 0000H 和 0001H 中的值赋给程序计数器。

图 3-7. 程序计数器结构



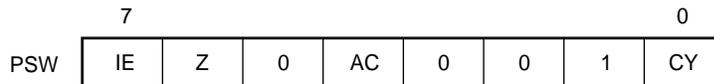
(2) 程序状态字(PSW)

程序状态字 (PSW) 是一个 8 位寄存器，由各种标志位组成，通过指令执行对其进行设置或清零。

根据中断请求的产生或 PUSH PSW 指令执行，程序状态字的内容自动入栈；通过执行 RETI 和 POP PSW 指令，程序状态字的值自动恢复。

RESET 输入将程序状态字的内容设置为 02H。

图 3-8. 程序状态字格式



(a) 中断允许标志位(IE)

该标志用于控制 CPU 响应中断请求操作。

当 IE 为 0 时，表示不允许中断（DI），即禁止所有中断。

当 IE 为 1 时，表示允许中断（EI），通过各种中断源的中断屏蔽标志来控制响应中断请求。

当执行 DI 指令或中断请求得到响应时，该标志复位（0）；当执行 EI 指令时，该标志设置为 1。

(b) 零标志位 (Z)

当操作结果为 0 时，该标志置 1，其它情况清 0。

(c) 半进位标志位 (AC)

如果操作结果中有一个来自第 3 位的进位或在第 3 位上有借位，则该标志置 1。其他情况该标志清 0。

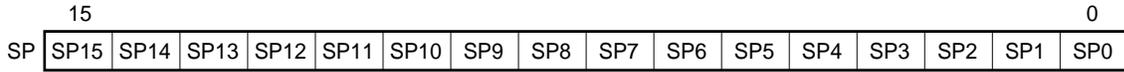
(d) 进位标志位 (CY)

该标志存储的是在执行加减指令时出现的上溢或下溢。它也存储循环指令执行中的转移值，还可以在位操作指令执行中作为位累加器使用。

(3) 堆栈指针 (SP)

这是一个 16 位的寄存器，用来存放内存堆栈区的起始地址。只有内部高速 RAM 区域才能被设置为堆栈区(除了内部高速 RAM 区域以外，其它区域不能被设置为堆栈区)。

图 3-9. 堆栈指针格式



在向堆栈写 (存) 内容时，堆栈指针 SP 累减，而从堆栈中读出 (恢复) 内容时，堆栈指针 SP 累加。
堆栈的数据存储/恢复操作过程如图 3-10 和 3-11 所示。

- 注意事项**
1. 由于 RESET 后 SP 值不确定，所以在使用堆栈前必须先对 SP 初始化。
 2. 堆栈指针只能设置到高速 RAM 区，而且实际上只有其低 10 位可以设置为堆栈指针。
0FF00H 在特殊功能寄存器 SFR 区，而不在高速 RAM 区，因此将其转换为高速 RAM 区的 0FB00H。
当数据真正入栈后，0FB00H 减 1 至 0FAFFH，但数据并不位于高速 RAM 区，将 0FB00H 转换为 0FEFFH，与堆栈指针设置为 0FF00H 时的值相同。

图 3-10. 将数据存入到堆栈区

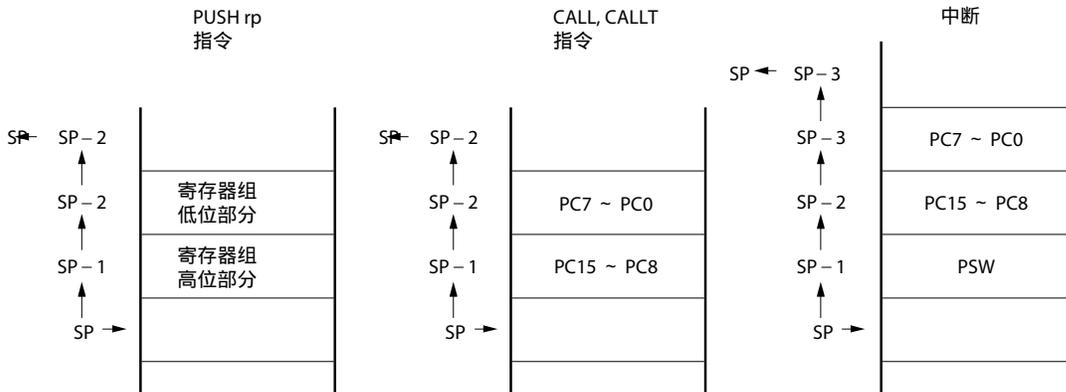
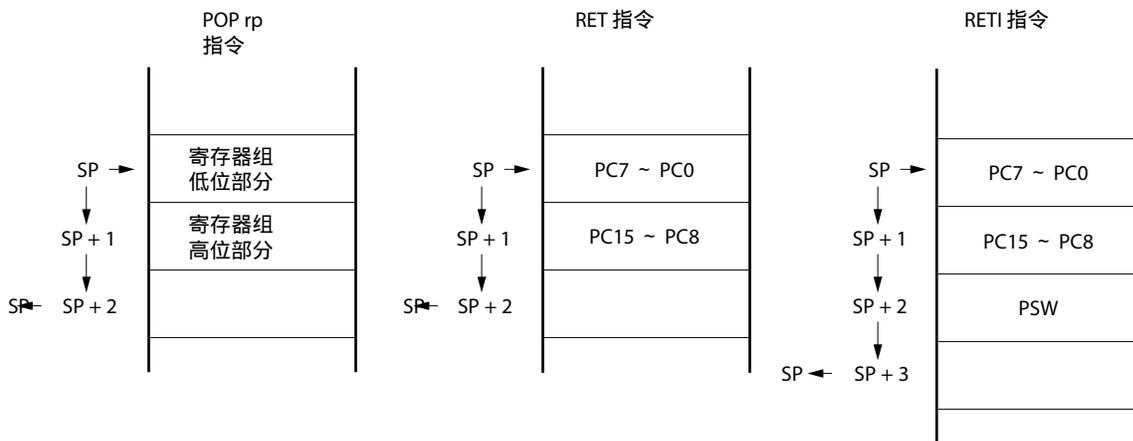


图 3-11. 从堆栈区读出数据



3.2.2 通用寄存器

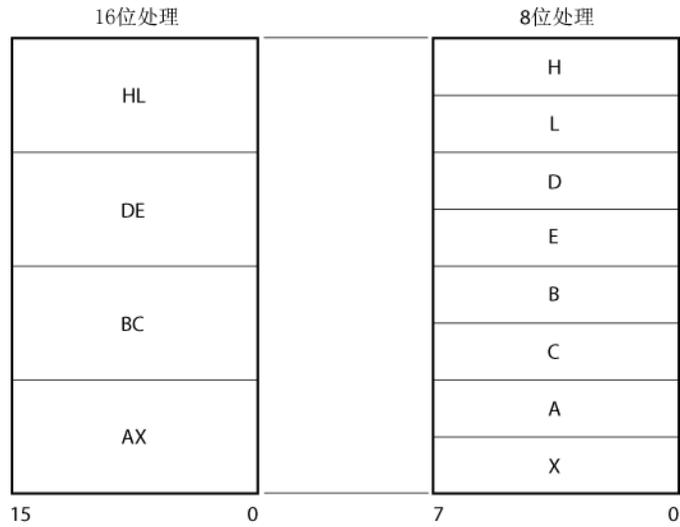
通用寄存器有8个8位寄存器(X、A、C、B、E、D、L和H)组成。

此外每个寄存器可作为一个8位寄存器使用，两个成对的8位寄存器可作为一个16位寄存器(AX、BC、DE和HL)使用。

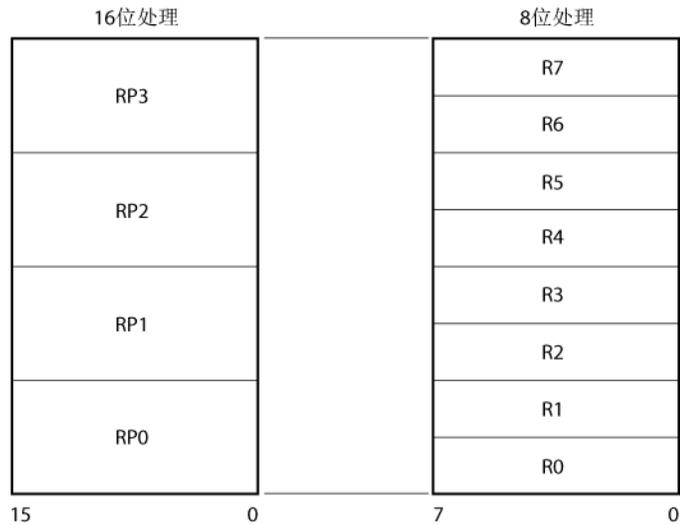
描述通用寄存器可以使用功能名称(X、A、C、B、E、D、L、H、AX、BC、DE和HL)和绝对名称(R0到R7，RP0到RP3)。

图 3-12. 通用寄存器结构

(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特定的功能。

从 FF00H 到 FFFFH 总共 256 字节的区域分配给特殊功能寄存器。

特殊功能寄存器可像通用寄存器那样用操作指令、转移指令以及位操作指令进行操作。根据特殊功能寄存器的类型不同，可操作的位单元也不同，可以是1位、8位和16位。

每种操作位单元描述如下。

- 1 位操作
1 位操作指令的操作数(sfr.bit)被描述为汇编程序的保留符号。该操作也可由一个地址和位来定义。
- 8 位操作
8 位操作指令的操作数(sfr)被描述为汇编程序的保留符号。该操作也可由一个地址来定义。
- 16 位操作
16 位操作指令的操作数(sfrp) 被描述为汇编程序的保留符号。当指定地址时，表示为一个偶地址。

表 3-3 为特殊功能寄存器列表。表中术语的含义如下：

- 符号
符号表示特殊功能寄存器的地址，在 RA78K0S 中是保留字。并用 CC78K0S 中的 #pragma sfr 指令定义为 sfr 变量。因此，如果使用汇编程序或集成调试器，这些符号用作指令操作数。
- R/W
表示特殊功能寄存器可读或可写。
R/W：读/写
R：只读
W：只写
- 可同时操作的位数
表示可特殊功能寄存器中可操作的位单元 (1、8、或 16)。
- 复位后的状态
表示 RESET 输入后每个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)名称								R/W	可同时操作的位数			复位后状态	参考页码
		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	√	√	-	00H	57
FF03H	P3	0	0	0	P34	0	P32	0	0		√	√	-	00H	57
FF04H	P4	0	0	0	0	P43	0	0	P40		√	√	-	00H	57
FF05H 至 FF0DH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF0EH	CMP01	-	-	-	-	-	-	-	-	R/W	-	√	-	00H	74
FF0FH	CMP11	-	-	-	-	-	-	-	-		-	√	-	00H	74
FF10H 至 FF21H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF22H	PM2	1	1	1	1	PM23	PM22	PM21	PM20	R/W	√	√	-	FFH	56
FF23H	PM3	1	1	1	1	1	PM32	1	1		√	√	-	FFH	56
FF24H	PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40		√	√	-	FFH	56
FF25H 至 FF31H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF32H	PU2	0	0	0	0	PU23	PU22	PU21	PU20	R/W	√	√	-	00H	58
FF33H	PU3	0	0	0	PU34	0	PU32	0	0		√	√	-	00H	58
FF34H	PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40		√	√	-	00H	58
FF35H 至 FF47H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF48H	WDTM	0	1	1	WDC S4	WDC S3	WDC S2	WDC S1	WDC S0	R/W	-	√	-	67H	90
FF49H	WDTE	-	-	-	-	-	-	-	-		-	√	-	9AH	91

注 只有 P34 为只输入端口。

表 3-3. 特殊功能寄存器 (2/3)

地址	符号	特殊功能寄存器(SFR)名称								R/W	可同时操作的位数			复位后状态	参考页码
		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 ^{注1}	124
FF51H	LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0		-	√	-	00H ^{注1}	125
FF52H, FF53H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF54H	RESF	0	0	0	WDT RF	0	0	0	LVIRF	R	-	√	-	00H ^{注2}	118
FF55H 至 FF57H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF58H	LSRCM	0	0	0	0	0	0	0	<LSR STOP>	R/W	√	√	-	00H	64
FF59H 至 FF6FH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FF70H	TMHMD1	<TMH E1>	CKS1 2	CKS1 1	CKS1 0	TMM D11	TMM D10	<TOLE V1>	<TOE N1>	R/W	√	√	-	00H	75
FF71H 至 FF9FH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FFA0H	PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0	W	-	√	-	未定义	148
FFA1H	PFS	0	0	0	0	0	WEP RERR	VCER R	FPRE RR	R/W	√	√	-	00H	149
FFA2H	FLPMC	0	PRSE LF4	PRSE LF3	PRSE LF2	PRSE LF1	PRSE LF0	0	FLSP M		-	√	-	未定义	147
FFA3H	FLCMD	0	0	0	0	0	FLCM D2	FLCM D1	FLCM D0		√	√	-	00H	151
FFA4H	FLAPL	FLA P7	FLA P6	FLA P5	FLA P4	FLA P3	FLA P2	FLA P1	FLA P0		√	√	-	未定义	152

注 s 1. 只有由 LVI 引起的复位保留该状态。
2. 根据复位源的不同而变化。

备注 对尖括号(<>)内的位名称来说,其在 RA78K0S 中定义为一保留字,而且使用 CC78K0S 中的 #pragma sfr 指令定义为 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)名称								R/W	可同时操作的位数			复位后状态	参考页码
		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	√	√	-	未定义	152
FFA6H	FLAPHC	0	0	0	0	FLAP C11	FLAP C10	FLAP C9	FLAP C8		√	√	-	00H	152
FFA7H	FLAPLC	FLAP C7	FLAP C6	FLAP C5	FLAP C4	FLAP C3	FLAP C2	FLAP C1	FLAP C0		√	√	-	00H	152
FFA8H	FLW	FLW7	FLW6	FLW5	FLW4	FLW3	FLW2	FLW1	FLW0		-	√	-	00H	153
FFA9H 至 FFDFH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FFE0H	IF0	0	0	0	<TMIF H1>	<PIF 1>	<PIF 0>	<LVI IF>	0	R/W	√	√	-	00H	100
FFE1H 至 FFE3H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FFE4H	MK0	1	1	1	<TMM KH1>	<PMK 1>	<PMK 0>	<LVIM K>	1	R/W	√	√	-	FFH	101
FFE5H 至 FFEBH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FFECH	INTM0	0	0	ES11	ES10	ES01	ES00	0	0	R/W	-	√	-	00H	101
FFEDH 至 FFF2H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FFF3H	PPCC	0	0	0	0	0	0	PPCC 1	PPCC 0	R/W	√	√	-	02H	63
FFF4H 至 FFFAH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FFFBH	PCC	0	0	0	0	0	0	PCC1	0	R/W	√	√	-	02H	63

备注 对尖括号(<>)内的位名称来说，其在 RA78K0S 中定义为一保留字，CC78K0S 中使用#pragma sfr 指令定义为 sfr 变量。

3.3 指令地址寻址

一条指令的地址是由程序计数器(PC)决定的。根据执行指令时所获取的下一条指令字节数，程序计数器(PC)的内容自动增加（每个字节加1）。在执行转移指令时，将程序计数器（PC）的内容设置为转移目的地址,并按以下寻址方式确定地址。（要了解每条指令的详细信息，请参阅**78K/0S**系列指令用户手册(U11047E)）。

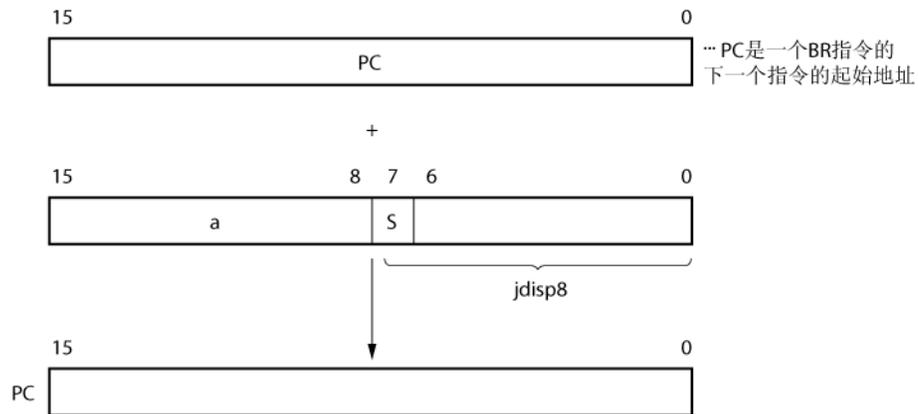
3.3.1 相对寻址

[功能]

将一条指令的8位立即数(偏移量: jdisp8)与下一条指令的起始地址相加，结果赋给程序计数器（PC），然后转向相加结果指向的地址。偏移量为带符号位的补码(-128至+127)，其中第7位是符号位。换句话说，在相对寻址中，相对转移值是从下一条指令的起始地址开始，范围在-128与+127之间。

当执行“BR \$addr16”指令或条件转移指令时，将实现上述功能。

[图示]



S=0时，a表明所有的位是“0”。
S=1时，a表明所有的位是“1”。

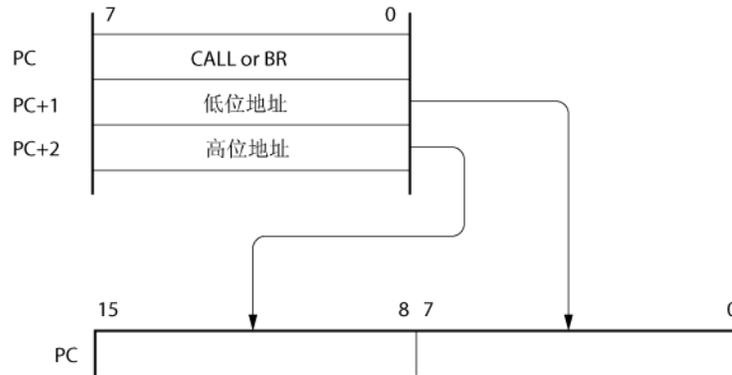
3.3.2 立即寻址

[功能]

将指令中的立即数赋给程序计数器 (PC)，然后转向该地址。
 在执行“CALL !addr16”指令、“BR !addr16”指令时，实现此功能。
 CALL !addr16 和 BR !addr16 指令的转移地址范围是所有内存空间。

[图示]

CALL !addr16 和 BR !addr16 指令

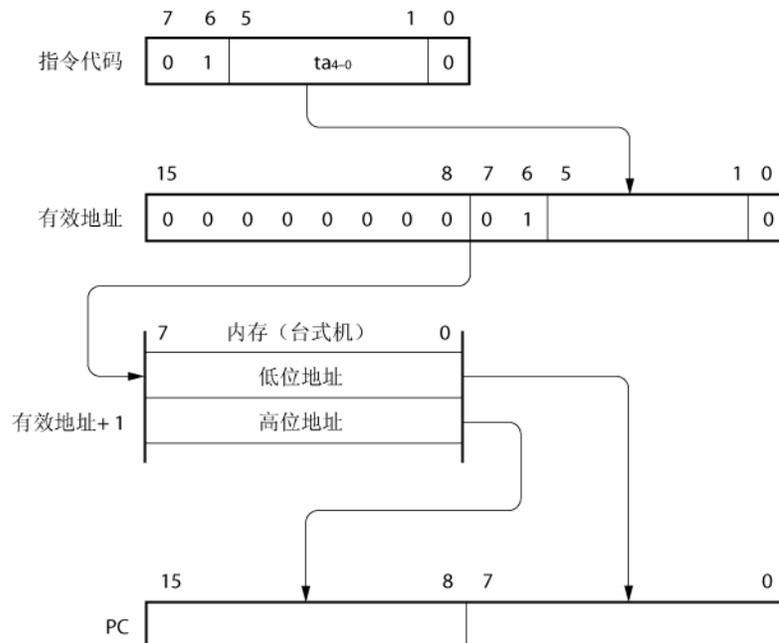


3.3.3 表间接寻址

[功能]

通过指令码低5位的立即数（从第1位到第5位），访问特定存储单元的表的内容（转移目的地址），并将表的内容赋给程序计数器(PC)，然后转向该地址。
 在执行 CALLT [addr5]指令时，进行表间接寻址。该指令访问的地址范围是内存表中 40H 至 7FH 之间所存储的地址，转移地址范围可以是整个存储器空间。

[图示]

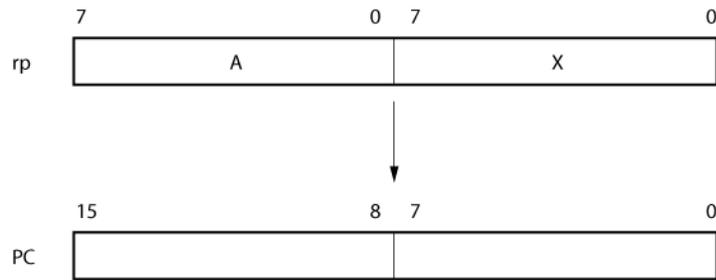


3.3.4 寄存器寻址

[功能]

将一条指令字的寄存器对(Ax)的内容赋值给程序计数器(PC)，然后转向该地址。
在执行“BR Ax”指令时，实现此功能。

[图示]



3.4 操作数地址寻址

以下方法(寻址)用来规定指令执行期间寄存器寻址和存储器寻址所进行的操作。

3.4.1 直接寻址

[功能]

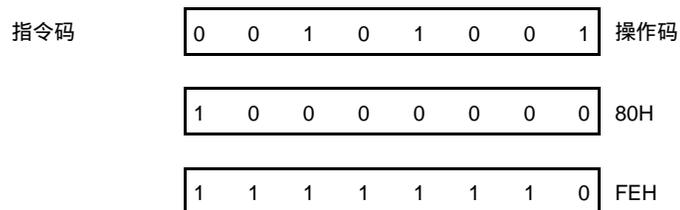
直接寻址方式根据指令字中的立即数直接寻址。

[操作数格式]

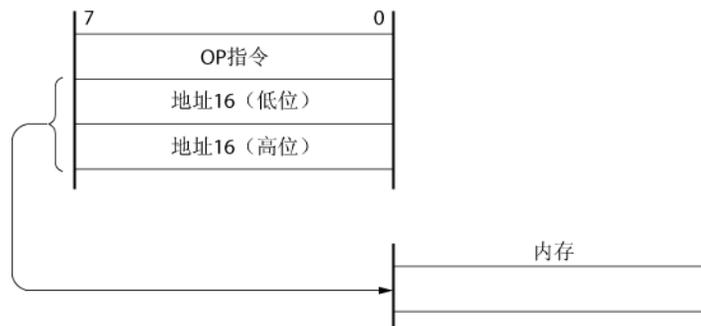
标识符	描述
addr16	标号或 16 位立即数

[举例]

MOV A, !FE80H ; 将!addr16 设置为 FE80H



[图示]



3.4.2 短直接寻址

[功能]

用指令字中 8 位立即数直接对内存的固定操作区域寻址。

该方式的寻址范围是 FE80H 到 FF1FH 总共 160 字节的固定内存区域(FE80H 至 FEFFH (内部高速 RAM) + FF00H 至 FF1FH (特殊功能寄存器))。

采用短直接寻址方式的特殊功能寄存器 (SFR) 区域(FF00H 至 FF1FH)是整个特殊功能寄存器SFR 区域的一部分。程序中经常访问的端口、用作计时器计数器的比较寄存器都被映射到该区域。这些特殊功能寄存器 (SFR) 可以用很少的字节数和时钟数进行操作。

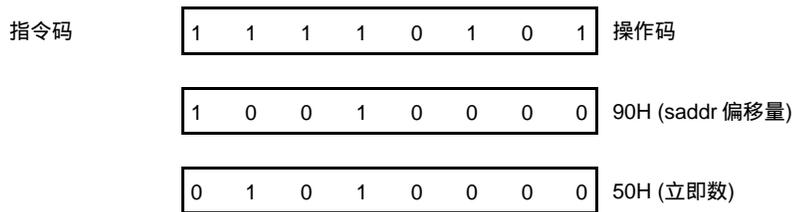
如果8 位立即数是在80H 和 FFH 之间, 则将一个有效地址的第8 位清除为0; 如果8 位立即数是在00H 与 1FH 之间, 则一个有效地址的第 8 位设置为 1。参见下面 [图示]。

[操作数格式]

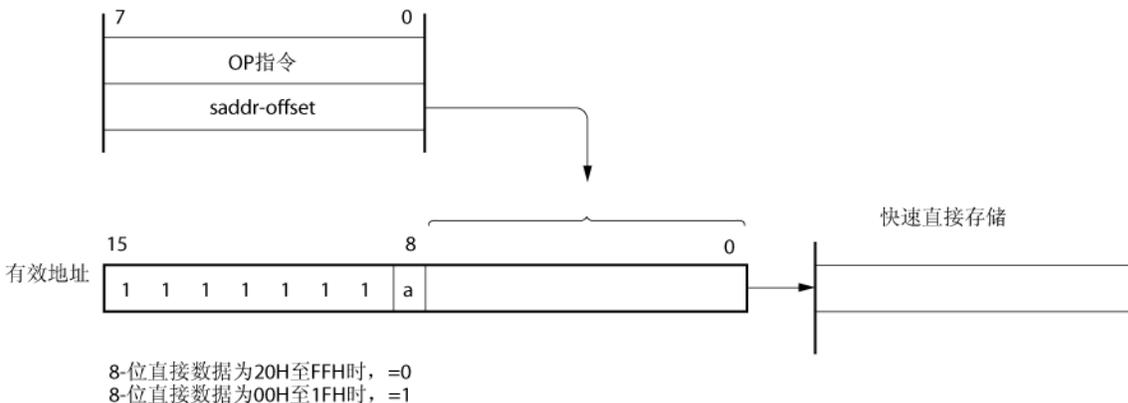
标识符	描述
saddr	标号或从 FE80H 到 FF1FH 的立即数
saddrp	标号或从 FE80H 到 FF1FH 的立即数 (仅使用偶地址)

[举例]

EQU DATA1 0FE90H ; DATA1 为 saddr 区的 FE90H ,
MOV DATA1 , #50H ; 设置立即数为 50H



[图示]



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-映射的特殊功能寄存器(SFR)区域进行寻址。
寻址区间为FF00H 至 FFFFH共256字节。而映射在FF00H 到 FF1FH 区间的特殊功能寄存器则采用短直接寻址方式。

[操作数格式]

标识符	描述
sfr	特殊功能寄存器名

[举例]

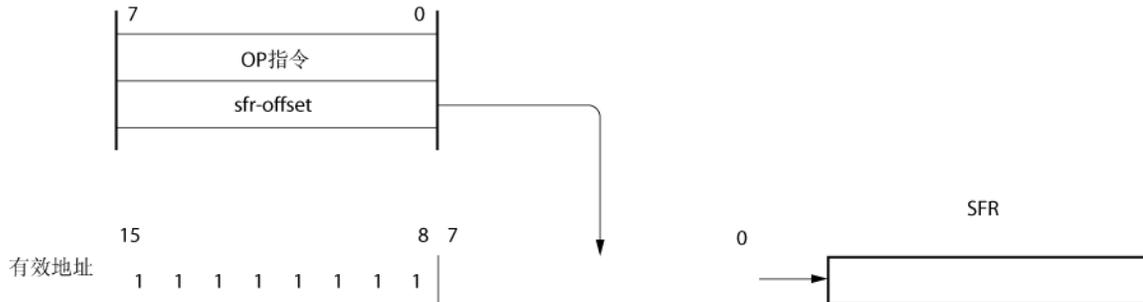
MOV PM0 , A ; 当 sfr 选择为 PM0

指令码

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

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

[图示]



3.4.4 寄存器寻址

[功能]

通用寄存器作为操作数访问。

通过指令码中的寄存器指定码和功能名称对通用寄存器进行访问。

当执行具有下述操作数格式的指令时，执行寄存器寻址操作。当指定 8 位寄存器时，用指令码中的 3 位指定一个 8 位寄存器。

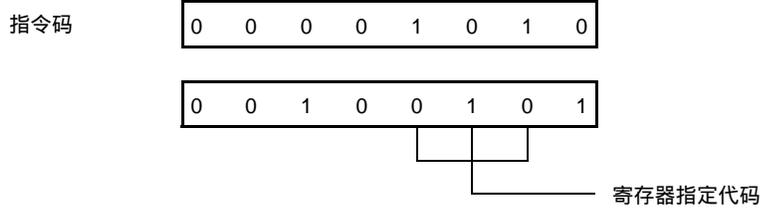
[操作数格式]

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

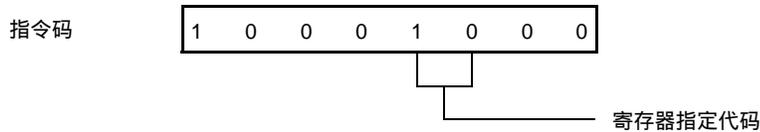
'r' 和 'rp'既可以用绝对名称(R0 至 R7 以及 RP0 至 RP3)描述，也可以用功能名称(X、A、C、B、E、D、L、H、AX、BC、DE、以及 HL)描述。

[举例]

MOV A, C; 选择寄存器 C 作为 r



INCW DE; 选择寄存器对 DE 作为 rp



3.4.5 寄存器间接寻址

[功能]

根据把寄存器对的内容作为操作数来进行寻址。由指令码中的寄存器对标识码来指定要访问的寄存器对。这种方式的寻址范围是整个存储空间。

[操作数格式]

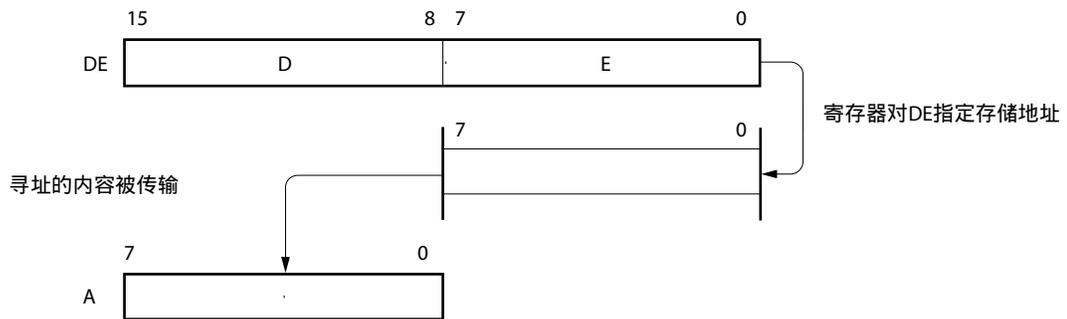
标识符	描述
-	[DE], [HL]

[举例]

MOV A, [DE]; 选择寄存器对 [DE]

指令码

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

[图示]

3.4.6 基址寻址

[功能]

将8位立即数加到基址寄存器对HL中,根据相加结果去寻址。将偏移量扩展为16位正数,完成加法操作,进位忽略不计。该寻址方式可对整个内存空间进行。

[操作数格式]

标识符	描述
-	[HL+byte]

[举例]

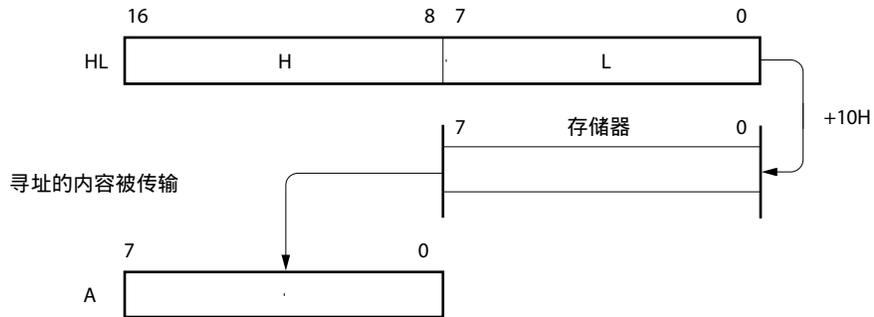
MOV A, [HL+10H]; byte 的值为 10H

指令码

0 0 1 0 1 1 0 1

0 0 0 1 0 0 0 0

[图示]



3.4.7 堆栈寻址

[功能]

根据堆栈指针(SP)的内容对堆栈区域进行间接寻址。

当执行 PUSH, POP, 子程序调用和 RETURN 指令时, 或者根据中断请求产生对寄存器进行保存或恢复时, 将自动采用这种寻址方式。

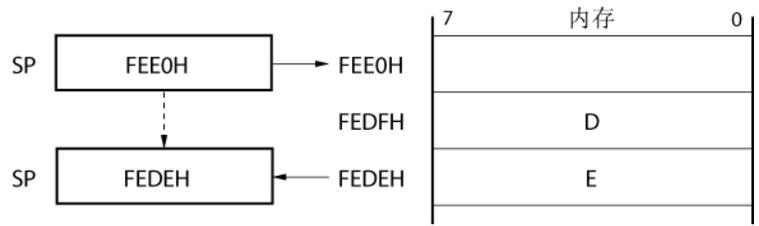
该方式仅对内部高速 RAM 区域进行寻址。

[举例]

以 PUSH DE 指令为例



[图示]



第四章 端口功能

4.1 端口功能

μ PD78F9500, 78F9501, 78F9502 产品中提供的端口如图 4-1 所示, 这些端口可用于多种控制操作。每个端口的功能如表 4-1 所示。

除了作为数字 I/O 端口外, 这些端口还具有复用功能。需要了解这些端口详细的复用功能, 请参阅第二章 引脚功能。

图 4-1. 端口功能

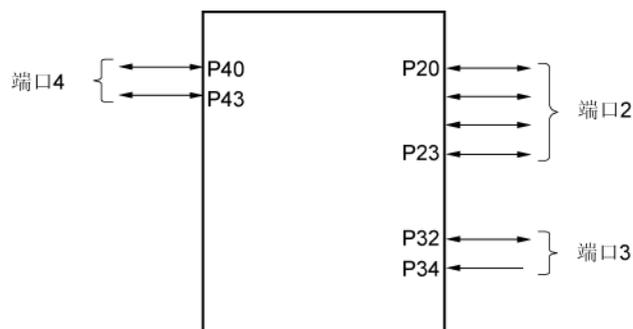


表 4-1. 端口功能

引脚名称	I/O	功能		复位后的功能状态	复用功能引脚
P20	I/O	端口 2 4 位 I/O 端口 可以设置为 1 位输入或输出模式 可以通过设置软件连接内置上拉电阻		输入	TOH1
P21					INTP0
P22					-
P23 [‡]					EXCLK [‡]
P32	I/O	端口 3 可以通过设置软件 连接内置上拉电阻.	可以设置为 1 位输入或输出模式	输入	INTP1
P34 [‡]	输入		仅用于输入	输入	$\overline{\text{RESET}}^{\#}$
P40 和 P43	I/O	端口 4 2 位 I/O 端口 可以设置为 1 位输入或输出模式 可以通过设置软件定义连接内置上拉电阻		输入	-

注 需要了解更多的关于引脚功能设置方法, 请参阅 第十三章 选项字节。

注意事项 复位期间, P22 引脚和 P23/EXCLK 引脚电平被强制下拉。而 P34/ $\overline{\text{RESET}}$ 引脚电平在复位期间, 由复位功能引脚/上电清零电路强制上拉。

备注 当高速内部振荡器选作系统时钟时, 可以配置 P23 引脚。

4.2 端口配置

端口由以下硬件组成

表 4-2. 端口配置

项目	配置
控制寄存器	端口模式寄存器(PM2 至 PM4) 端口寄存器(P2 至 P4) 上拉电阻选择寄存器(PU2 至 PU4)
端口	总数: 8 (CMOS I/O: 7, CMOS 输入: 1)
上拉电阻	总数: 7

4.2.1 端口 2

端口2是具有输出锁存的4位I/O端口。通过使用端口模式寄存器2 (PM2)，可将端口2的每一位设置为位输入或输出模式或1位输出模式。如果P20至P23引脚作为输入端口使用时，可由上拉电阻选择寄存器2 (PU2)定义内置上拉电阻的使用。

该端口也可用于定时器 I/O，以及外部中断请求输入。

引脚 P23 也可用作系统时钟振荡器的 EXCLK 引脚。因此，根据所选择的系统时钟振荡器的不同，EXCLK 引脚功能也有所不同。可以使用以下两种系统时钟振荡器。

(1) 高速内部振荡器

引脚 P23 可以用作 I/O 端口引脚。

(2) 外部时钟输入

引脚 P23 用作 EXCLK 引脚输入外部时钟，所以不能用作 I/O 端口引脚。

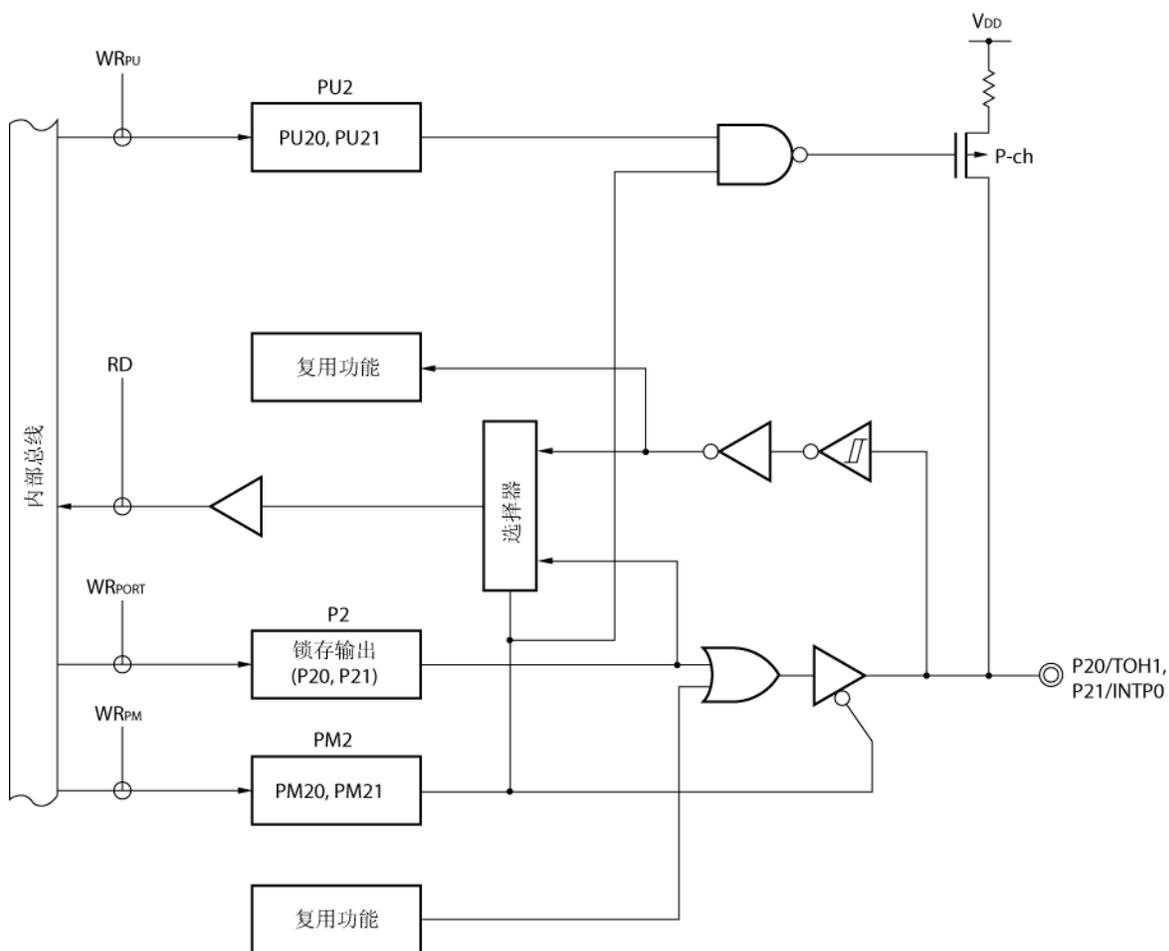
在选项字节定义选择系统时钟振荡器。需要了解有关更详细的情况，请参阅第十三章选项字节。

Reset 信号设置端口 2 为输入模式。

图 4-2 至 4-4 显示了端口 2 的结构框图。

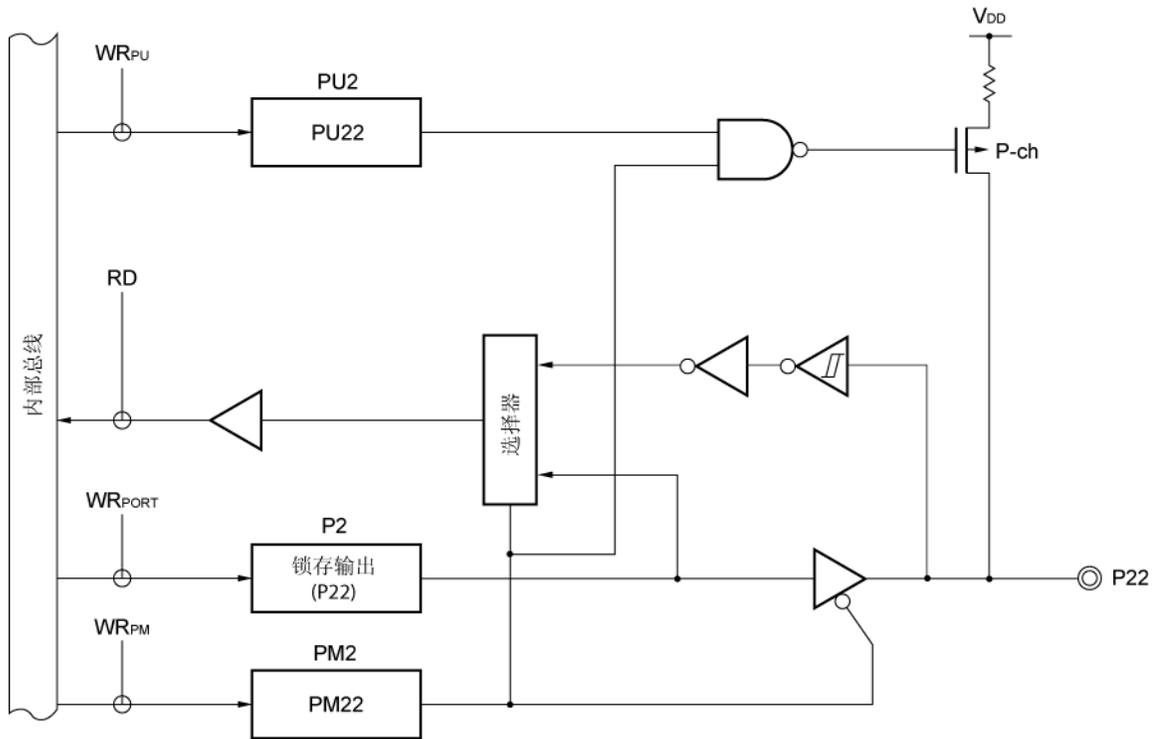
<R>

图 4-2. 端口 P20 和端口 P21 的结构框图



- P2: 端口寄存器 2
 PU2: 上拉电阻选择寄存器 2
 PM2: 端口模式寄存器 2
 RD: 读信号
 WR_{xx}: 写信号

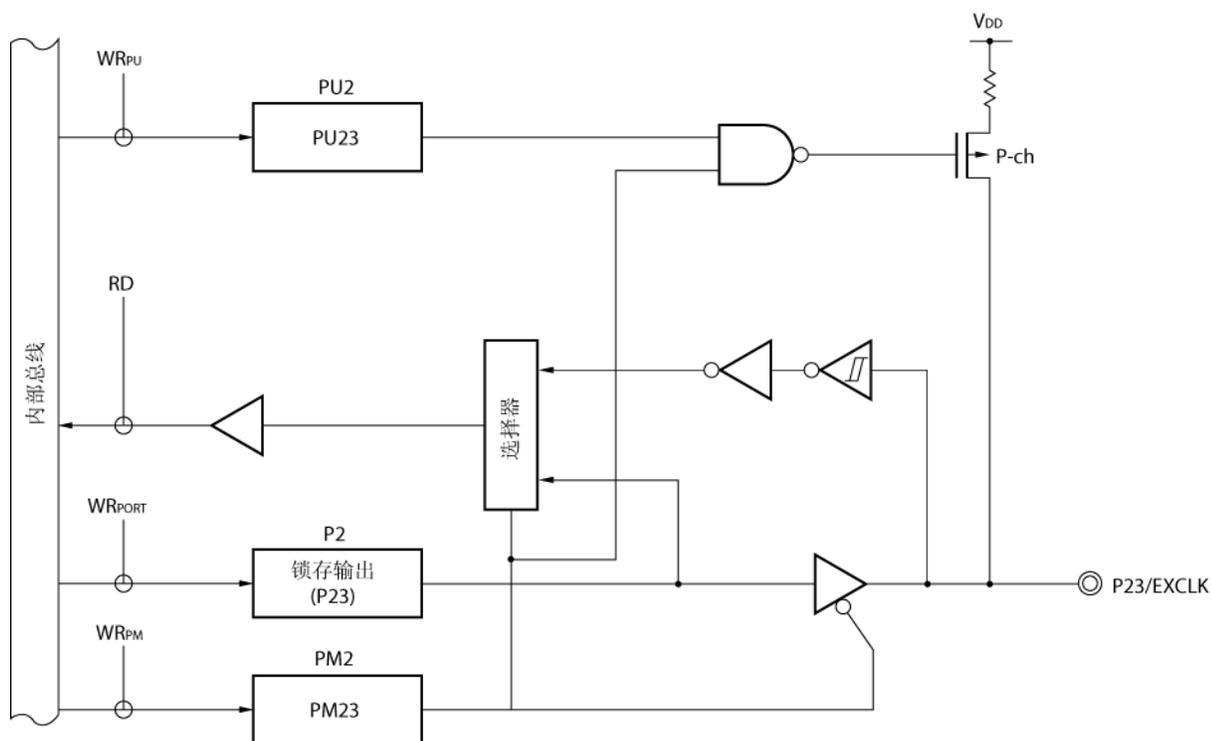
图 4-3. 端口 P22 结构框图



- P2: 端口寄存器 2
- PU2: 上拉电阻选择寄存器 2
- PM2: 端口模式寄存器 2
- RD: 读信号
- WR_{xx}: 写信号

<R>

图 4-4. 端口 P23 结构框图



- P2: 端口寄存器 2
 PU2: 上拉电阻选择寄存器 2
 PM2: 端口模式寄存器 2
 RD: 读信号
 WR_{xx}: 写信号

4.2.2 端口 3

P32 引脚是具有输出锁存的 1 位 I/O 端口。通过使用端口模式寄存器 3 (PM3)，可将该引脚设置为输入或输出模式。如果作为输入端口使用时，可由上拉电阻选择寄存器 3 (PU3)定义内置上拉电阻的使用。该引脚也可用作外部中断请求的输入端。

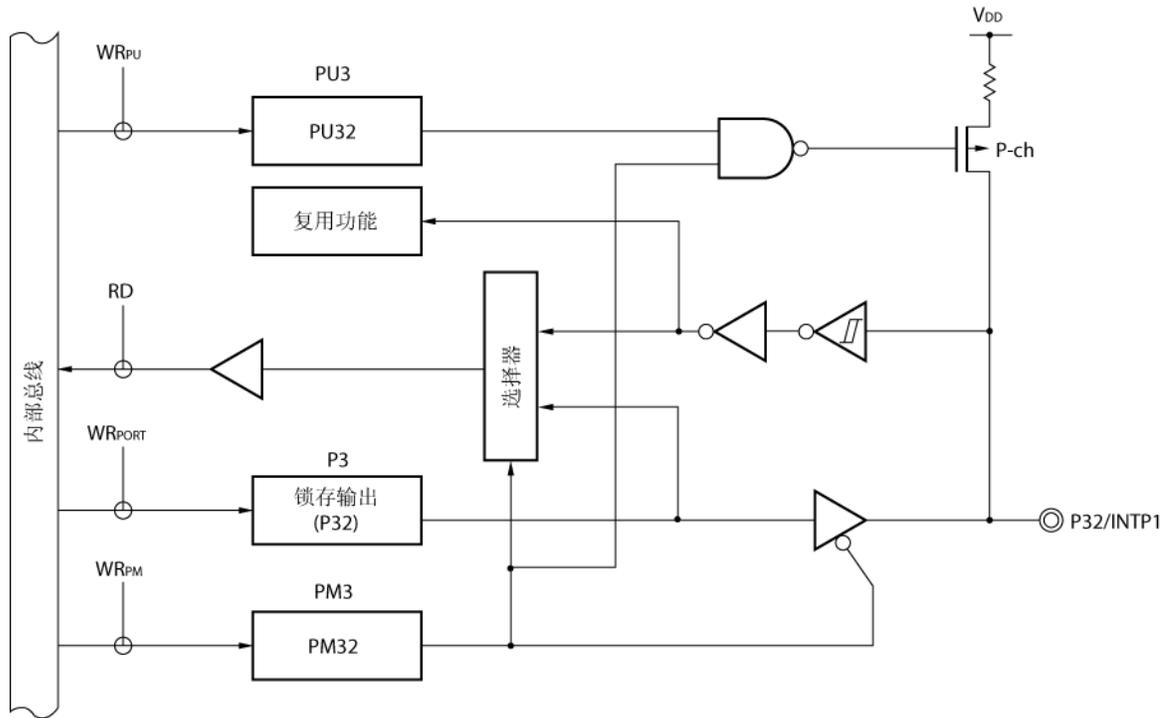
Reset 信号将引脚设置为输入模式。

P34 引脚为 1 位只输入端口。该引脚也可用作 $\overline{\text{RESET}}$ 引脚，系统上电时，该引脚为复位功能引脚。需要了解更多的关于引脚功能设置方法，请参阅第十三章 选项字节。

在 P32 和 P34 用作输入端口引脚时，要接上拉电阻。

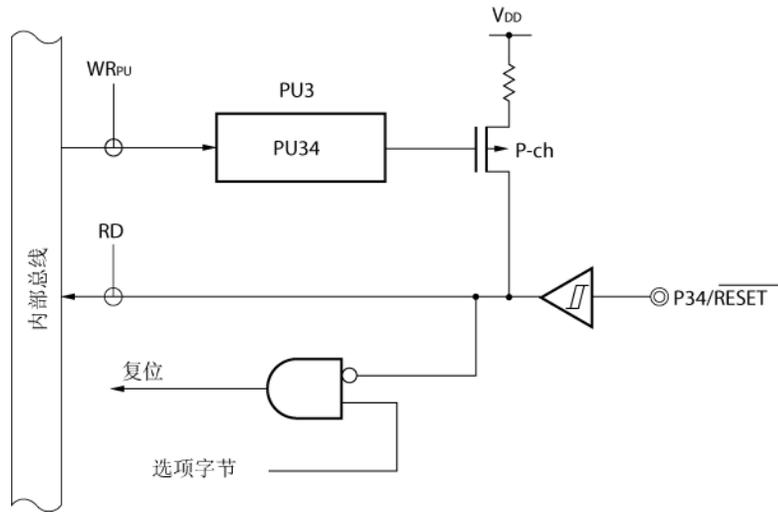
图 4-5 和 4-6 显示了端口 3 的结构框图。

图 4-5. 端口 P32 结构框图



- P3: 端口寄存器 3
 PU3: 上拉电阻选择寄存器 3
 PM3: 端口模式寄存器 3
 RD: 读信号
 WRxx: 写信号

图 4-6. 端口 P34 结构框图



RD: 读信号

注意事项 由于 P34 引脚可作 $\overline{\text{RESET}}$ 功能引脚使用，因此，如果该引脚作输入端口引脚用，就不能用作向 $\overline{\text{RESET}}$ 引脚输入外部复位信号。该引脚的端口功能是由选项字节选择设置的。需要了解有关更详细的情况，请参阅第十三章 选项字节。

而且，由于复位释放以后才引用选项字节，因此如果在引用之前向 $\overline{\text{RESET}}$ 引脚输入了低电平信号，就无法释放复位状态。当该引脚用作输入端口时，需连接上拉电阻。

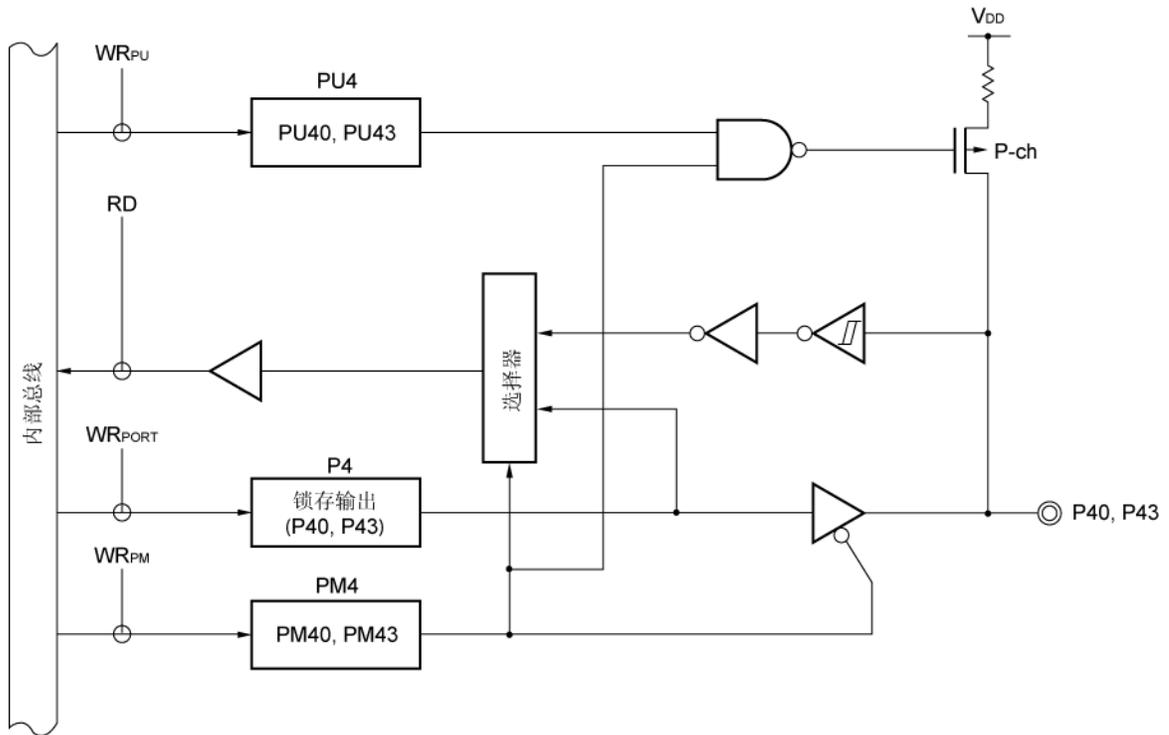
4.2.3 端口 4

端口 4 是具有输出锁存的 2 位 I/O 端口。通过使用端口模式寄存器 4 (PM4)，可将该端口的每一位 设置为输入或输出模式。如果 P40 至 P43 引脚作为输入端口使用时，可通过使用上拉电阻选择寄存器 4 (PU4) 定义按位连接内置上拉电阻。

Reset 信号将端口 4 设置为输入模式。

图 4-7 显示了端口 4 的结构框图。

图 4-7. 端口 P40 和端口 P43 结构框图



- P4: 端口寄存器 4
 PU4: 上拉电阻选择寄存器 4
 PM4: 端口模式寄存器 4
 RD: 读信号
 WR_{xx}: 写信号

4.3 控制端口功能寄存器

端口功能由以下三种寄存器控制

- 端口模式寄存器(PM2 至 PM4)
- 端口寄存器(P2 至 P4)
- 上拉电阻选择寄存器(PU2 至 PU4)

(1) 端口模式寄存器(PM2 至 PM4)

这类寄存器定义端口的 1 位输入或输出模式。

可由 1 位或 8 位存储器操作指令设置各端口模式寄存器。

RESET 输入将这些寄存器的内容置为 FFH。

如果某一端口引脚使用复用功能，可按表 4-3 所示设置端口模式寄存器和输出锁存。

注意事项 由于 P21 和 P32 也作外部中断输入引脚使用，所以，如果这些引脚各位设置为输出模式并且改变了其输出电平，则相应的中断请求标志置 1。因此，如果使用该端口引脚的输出模式，要事先将相应的中断屏蔽位置为 1。

图 4-8. 端口模式寄存器格式

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	1	1	1	1	PM43	1	1	PM40

PMmn	Pmn 引脚的 I/O 模式选择 (m = 2 至 4; n = 0 至 3)
0	输出模式(输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

(2) 端口寄存器(P2 至 P4)

这些寄存器用于写入由相应端口引脚输出到外部设备的数据。

如果读端口寄存器，则在输入模式下，读取的是引脚电平。而在输出模式下读取的是端口输出锁存的值。

P20 至 P23, P32, P40 以及 P43 可由 1 位或 8 位存储器操作指令进行设置。

Reset 信号将这些寄存器设置为 00H。

图 4-9. 端口寄存器格式

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	0	0	0	0	P43	0	0	P40

Pmn	m = 2 至 4; n = 0 至 4	
	数据输出控制 (输出模式下)	读数据输入 (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 是只读的, 所以其复位后值不确定.

(3) 上拉电阻选择寄存器(PU2 至 PU4)

这类寄存器规定是否使用 P20 至 P23, P40 以及 P43 等引脚的内置上拉电阻。通过对寄存器 PU2 至 PU4 进行设置, 可以将端口引脚按 PU2 至 PU4 相应位设置使用内置上拉电阻。

PU2 至 PU4 可由 1 位或 8 位存储器操作指令进行设置。

Reset 信号将这些寄存器设置为 00H。

图 4-10. 上拉电阻选择寄存器的格式

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	PU34	0	PU32	0	0

地址: FF34H, 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PU4	0	0	0	0	PU43	0	0	PU40

PU _m n	P _m n (m = 2 至 4; n = 0 至 4)的内置上拉电阻连接选择
0	不连接内置上拉电阻
1	连接内置上拉电阻

4.4 端口功能操作

根据端口 I/O 模式设置的不同，端口操作是不同的，具体如下所示。

注意事项 尽管 1 位存储器操作指令按位操作，但实际上是以 8 位形式访问端口。因此，如果一个端口既有输入引脚又有输出引脚，即便不对其进行位操作，端口定义为输入的引脚的输出锁存内容是不确定的。

4.4.1 写入 I/O 端口

(1) 输出模式

使用传送指令对输出锁存进行写操作，这样输出锁存的内容就从引脚输出。一旦数据写入输出锁存，则它将一直保存到有新数据写入输出锁存为止。

通过复位将输出锁存的内容清零。

(2) 输入模式

使用传送指令对输出锁存进行写操作，不过一旦输出缓冲关闭，引脚状态将不再改变。

一旦数据写入输出锁存，它将一直保存到新数据写入。

通过复位将输出锁存的内容清零。

4.4.2 从 I/O 端口读出

(1) 输出模式

使用传送指令读取输出锁存的内容。输出锁存的内容不会改变。

(2) 输入模式

使用传送指令读取引脚状态。输出锁存的内容不改变。

4.4.3 I/O 端口的操作

(1) 输出模式

对输出锁存的内容执行一个操作，结果写入输出锁存。输出锁存的内容从引脚输出。

一旦数据写入输出锁存，它将一直保存到新数据写入。

通过复位将输出锁存的内容清零。

(2) 输入模式

读取引脚电平，对获取内容执行操作，则将引脚内容写入输出锁存，但是如果输出缓冲已关闭，则引脚状态将不再改变。

通过复位将输出锁存的内容清零。

第五章 时钟发生器

5.1 时钟发生器的功能

时钟发生器包括两个时钟发生电路，其中一个电路产生的时钟信号(系统时钟)用于 CPU 和外围硬件，而另外一个电路产生的时钟信号(时间间隔发生时钟)用于看门狗定时器和 8 位定时器 H1 (TMH1)。

5.1.1 系统时钟振荡器

使用以下三种系统时钟振荡器。

- 高速内部振荡

该电路内部产生 8 MHz (典型值)时钟信号。可以通过执行 STOP 指令停止其振荡。

如果选用高速内部振荡用作系统时钟，EXCLK 引脚可以用作 I/O 端口引脚。

- 外部时钟输入电路

该电路由外部 IC 向 EXCLK 引脚输入时钟信号。可以通过执行 STOP 指令停止内部时钟。

可以通过使用选项字节来选择系统时钟源。需要了解更详细情况，请参阅 **第十三章 选项字节**。

当将 EXCLK 引脚用作 I/O 端口引脚时。需要了解详细情况，请参阅 **第四章 端口功能**。

5.1.2 用作时间间隔发生器的时钟振荡器

以下电路用作时间间隔发生器的时钟振荡器。

- 低速内部振荡器

该电路产生一 240 kHz (典型值)的振荡时钟信号。当低速内部振荡器由可以通过软件停止其振荡的选项字节指定时，该振荡器可以使用低速内部振荡模式寄存器(LSRCM) 来停止其振荡。

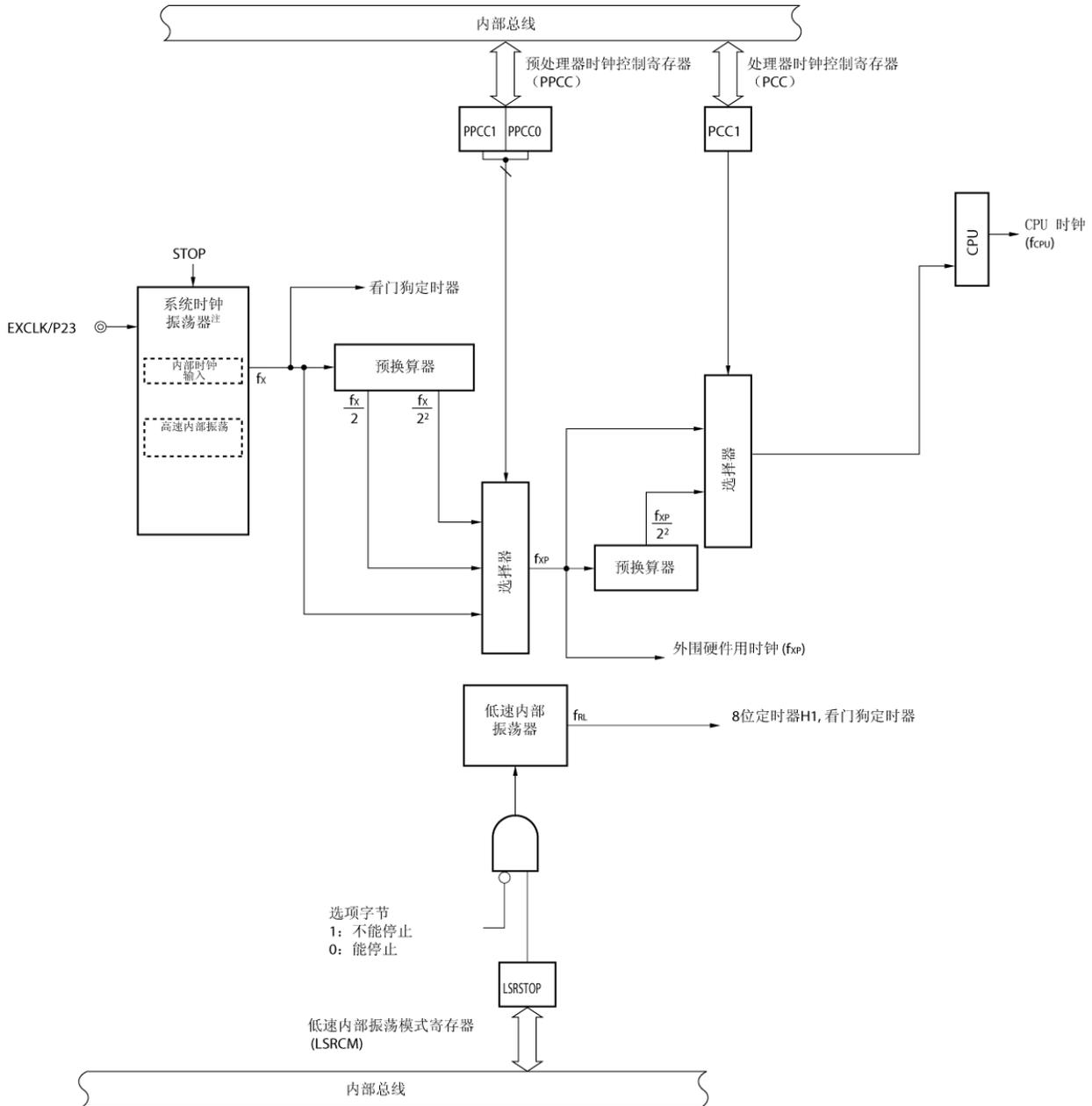
5.2 时钟发生器的格式

时钟发生器由以下硬件组成。

表 5-1. 时钟发生器的格式

项目	格式
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(LSRM)
振荡器	高速内部振荡 外部时钟输入电路 低速内部振荡器

图 5-1. 时钟发生器功能框图



注 通过使用选项字节选择高速内部振荡或外部时钟输入电路作为系统时钟源。

5.3 时钟发生器控制寄存器

时钟发生器由以下三种寄存器进行控制。

- 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)

(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)

这些寄存器用于指定系统时钟的分频比。

由 1 位或 8 位存储器操作指令设置 PCC 和 PPCC。

Reset 信号设置 PCC 及 PPCC 为 02H。

图 5-2. 处理器时钟控制寄存器格式(PCC)

地址: FFFBH, 复位: 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格式(PPCC)

地址: FFF3H, 复位: 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 时钟选择 (f_{CPU}) ^{注1}
0	0	0	f_x
0	1	0	$f_x/2$ ^{注2}
0	0	1	$f_x/2^2$
1	0	0	$f_x/2^2$ ^{注3}
0	1	1	$f_x/2^3$ ^{注2}
1	0	1	$f_x/2^4$ ^{注3}
其它			禁止设置

- 注
1. CPU 时钟的设定范围根据所使用的供电电压的不同而有所不同。务必请参阅第十七章 电气特性 中 AC 特点中所描的 CPU 时钟及外围时钟频率等相关内容。
 2. 如果 PPCC = 01H, 提供给外围硬件设备的时钟(f_{XP})信号为 $f_x/2$ 。
 3. 如果 PPCC = 02H, 提供给外围硬件设备的时钟(f_{XP})信号为 $f_x/2^2$ 。

在 μ PD78F9500, 78F9501, 78F9502 中, 速度最快的指令执行时间是 2 个 CPU 时钟。CPU 时钟(f_{CPU})与指令最短执行时间的关系如表 5-2 所示。

表 5-2. CPU 时钟与最小指令执行时间之间的关系

CPU 时钟 (f_{CPU}) ^注	最小指令执行时间: $2/f_{CPU}$	
	高速内部振荡时钟 (8.0 MHz (典型值))	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 使用选项字节选择 CPU 时钟(高速内部振荡时钟或外部时钟输入)。

(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。

当低速内部振荡器由选项字节指定可以通过软件停止其振荡时, 对该寄存器的设置是有效的.如果低速内部振荡器由选项字节指定不可以通过软件停止其振荡, 对该寄存器的设置是无效的, 同时低速内部振荡器将持续振荡。而且, WDT 的时钟源固定选为低速内部振荡器。需要了解更详细情况, 请参阅 [第七章 看门狗定时器](#)。

LSRCM 由 1 位或 8 位存储器操作指令设置。

Reset 信号设置 LSRM 为 00H。

图 5-4. 低速内部振荡模式寄存器(LSRCM)格式

地址: FF58H, 复位: 00H, R/W

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

LSRSTOP	低速内部振荡器的起振/停止
0	低速内部振荡器振荡
1	低速内部振荡器停止

5.4 系统时钟振荡器

有如下两种系统时钟可供选用。

- 高速内部振荡： 内部振荡时钟 8 MHz (典型值)。
- 外部时钟输入电路： 向 EXCLK 引脚提供 2 MHz 至 10 MHz 的时钟信号。

5.4.1 高速内部振荡

μ PD78F9500, 78F9501, 78F9502 提供一个 高速内部振荡 (8 MHz (典型值))。

如果通过选项字节将高速内部振荡器选作时钟源, 则 EXCLK 引脚可以用作 I/O 端口引脚。

有关选项字节的详细情况, 请参阅 **第十三章 选项字节**。有关 I/O 端口的详细情况, 请参阅 **第四章 端口功能**。

5.4.2 外部时钟输入电路

该电路向 EXCLK 引脚提供一外部 IC 时钟信号。

5.4.3 预分频器

预分频器将由系统时钟振荡器输出的时钟信号(f_x)进行分频, 生成外围硬件设备所需的分频时钟(f_{xp})。同时预分频器还将外围硬件设备的时钟信号(f_{xp})进行分频产生所需要的时钟信号。

备注 由选项字节(高速内部振荡, 或外部时钟输入电路)选定的振荡器的输出时钟被分频。有关选项字节的详细情况, 请参阅 **第十三章 选项字节**。

5.5 CPU时钟发生器的操作

由系统时钟(fx)向提供的时钟信号(fcpu)可由以下三种类型的振荡器振荡中任何一种振荡产生。

- 高速内部振荡: 内部振荡时钟 8 MHz (典型值)
 - 外部时钟输入电路: 向 EXCLK 引脚提供 2 MHz 至 10 MHz 的时钟信号。
- 系统时钟振荡器由选项字节进行选定。有关选项字节的详细情况, 请参阅 **第十三章 选项字节**。

(1) 高速内部振荡

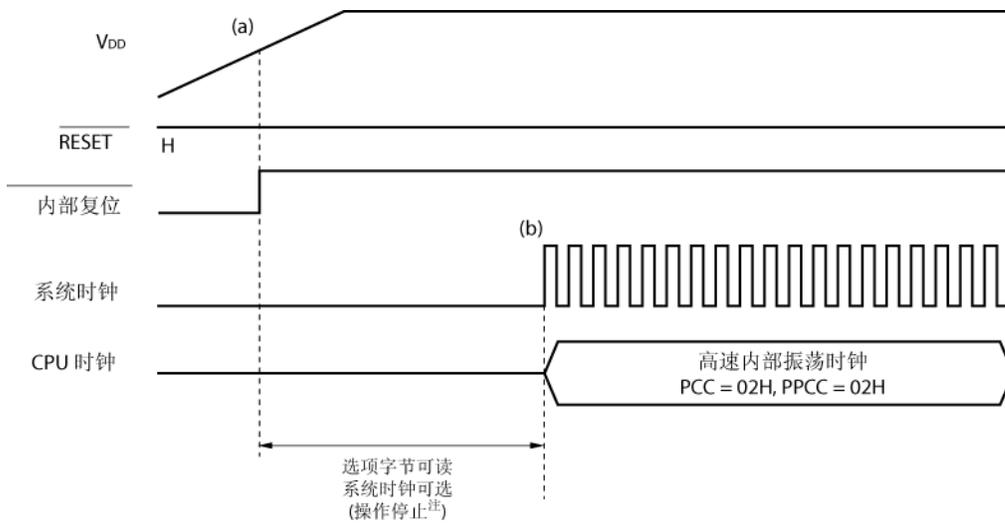
当由选项字节选定了高速内部振荡器时, 下属性能改进便成为可能。

- 启动时间的缩短
如果选择高速内部振荡器作为振荡器, 则 CPU 可不必等待系统时钟的稳定时间而直接启动。因此, 可以缩短启动时间。
- 提高扩展性能
如果选择高速内部振荡器作为振荡器, 则 EXCLK 引脚可以用作 I/O 端口引脚。需要了解更详细情况, 请参阅 **第四章 端口功能**。

图 5-5 和 5-6 显示了高速内部振荡器下缺省启动的时序图及状态转换图。

备注 使用高速内部振荡器时, 时钟精度为 $\pm 5\%$ 。

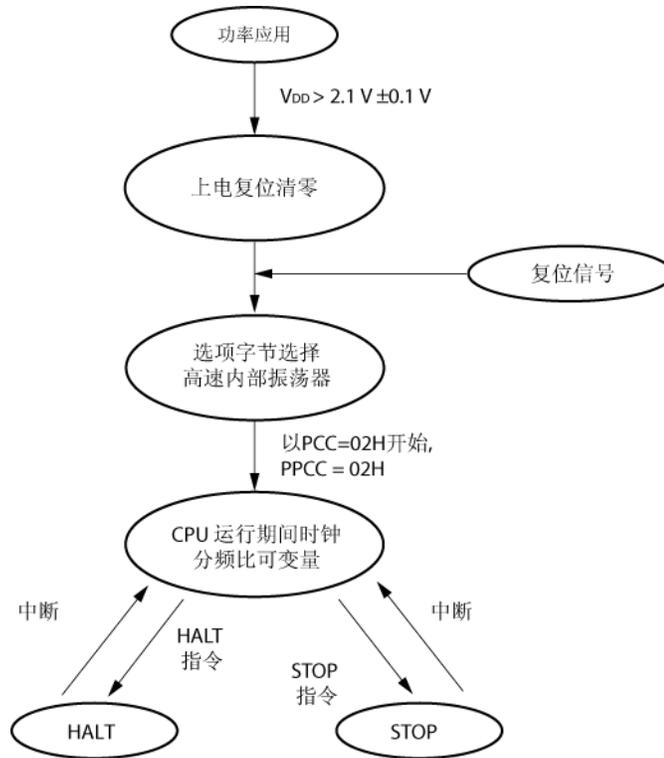
图 5-5. 高速内部振荡器下缺省启动的时序图



注 操作停止时间为 277 μs (MIN), 544 μs (典型值), 及 1.075 ms (MAX.)。

- (a) 上电时，由上电清零电路产生一内部复位信号，复位后，引用选项字节的内容并选择系统时钟。
- (b) 引用选项字节的内容并选择系统时钟之后，高速内部振荡器便作为系统时钟进行工作。

图 5-6. 高速内部振荡器下缺省启动的状态转移图



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) 外部时钟输入电路

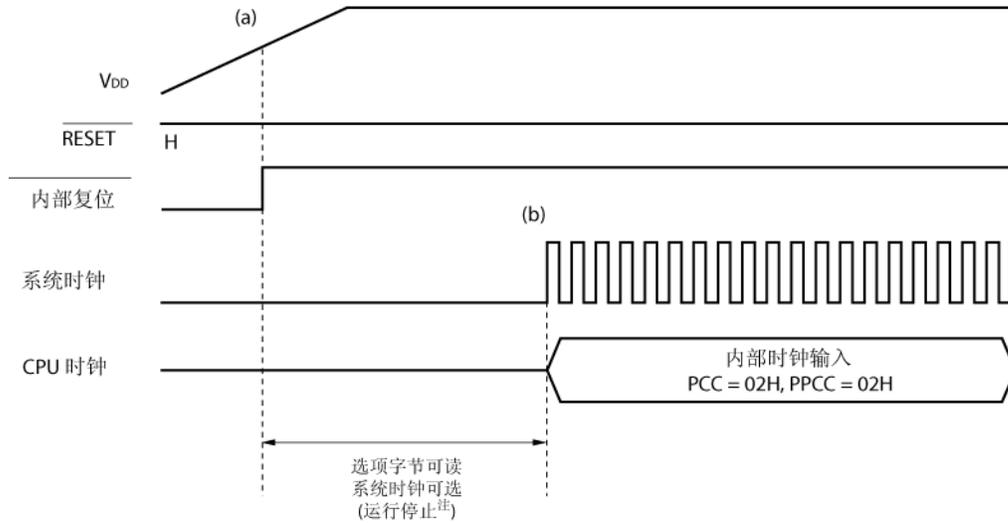
当由选项字节选定了外部时钟输入时，可以进行以下操作。

- 高速操作

相比高速内部振荡器(8 MHz (典型值))下，处理精度提高了。因为可以选择 2 MHz 至 10 MHz 的振荡频率，同时还可以提供频率波动很小的外部时钟信号。

图 5-7 和 5-8 显示了外部时钟输入下缺省启动的时序图及状态转换图。

图 5-7. 由外部时钟输入缺省启动的时序图

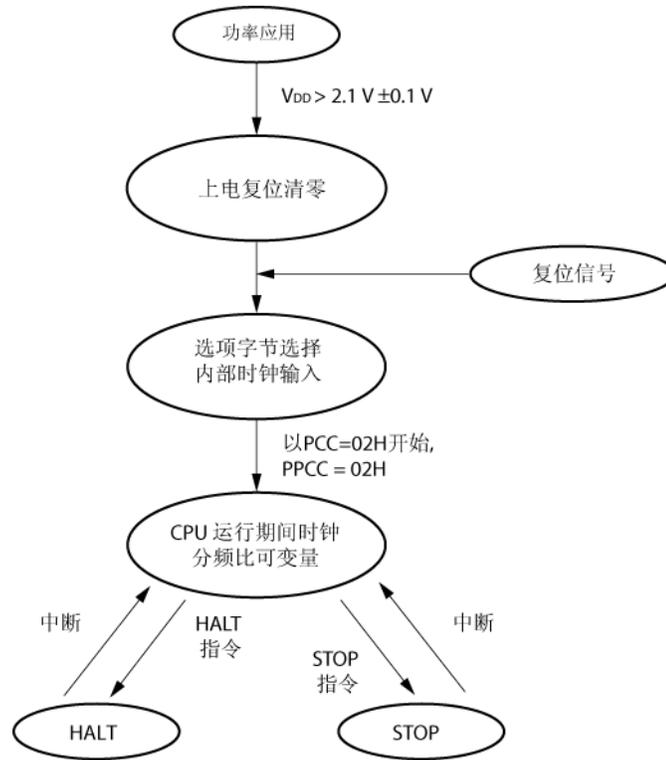


注 操作停止时间为 277 μ s (MIN.), 544 μ s (典型值), and 1.075 ms (MAX.)。

(a) 上电时，由上电清零电路产生一内部复位信号，复位后，引用选项字节的内容并选择系统时钟。

(b) 引用选项字节的内容并选择系统时钟之后，外部时钟输入便作为系统时钟进行工作。

图 5-8. 由外部时钟输入缺省启动的状态转移图



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 向外围设备提供时钟的时钟发生器的操作

有以下两种类型的时钟外围硬件设备使用。

- 向外围硬件设备输出的分频时钟 (fxp)
- 低速内部振荡时钟 (fRL)

(1) 向外围硬件设备输出的分频时钟

通过对系统时钟(fx)分频向外围硬件设备提供时钟。分频比通过预处理器时钟控制寄存器(PPCC)来进行选择。有三种可选频率：“fx”，“fx/2”，及“fx/2²”。表 5-3 列出了可供外围设备使用的时钟频率。

表 5-3. 外围硬件设备输出的分频时钟

PPCC1	PPCC0	外围硬件设备输出的分频时钟 (fxp)的选择
0	0	fx
0	1	fx/2
1	0	fx/2 ²
1	1	禁止设置

(2) 低速内部振荡时钟

低速内部振荡器作为用于间隔时间发生器的时钟振荡器，总是在复位释放之后便会启动，并且以 240 kHz (典型值)的频率振荡。

它可以由选项字节进行指定该低速内部振荡器是否可由软件停止。如果指定可以由软件停止其振荡，则可以使用低速内部振荡模式寄存器(LSRCM)来启动或停止其振荡。如果指定不可以由软件停止其振荡，则 WDT 的时钟源固定为低速内部振荡时钟 (fRL)。

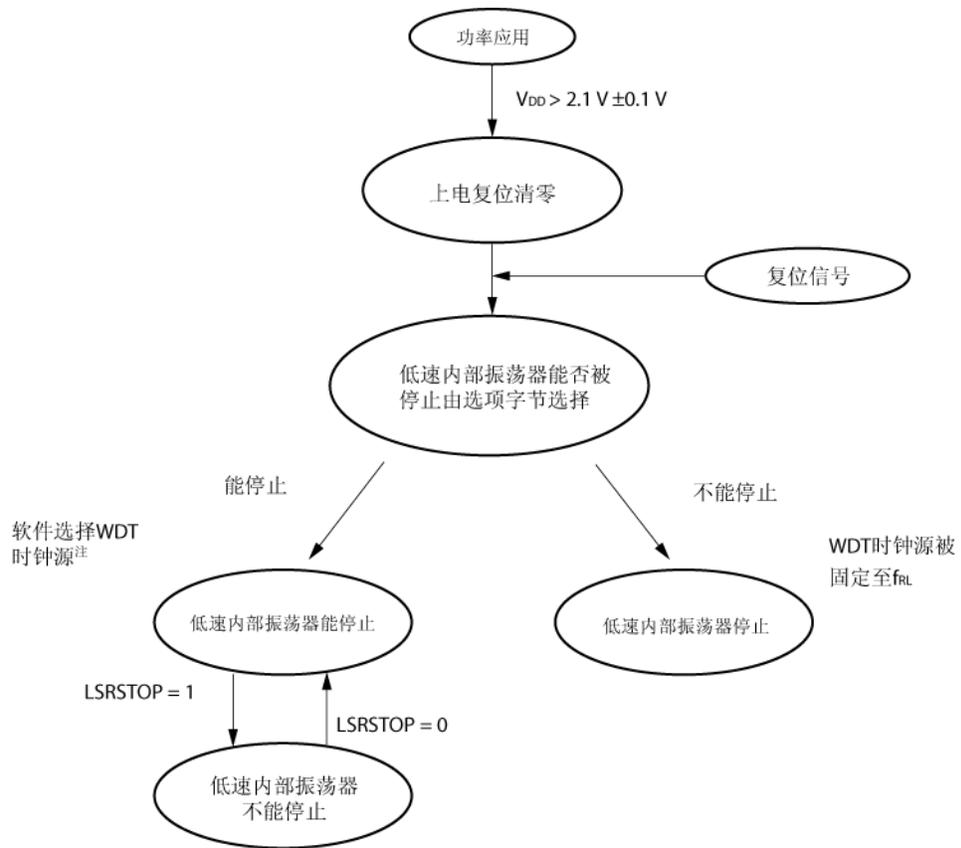
低速内部振荡器独立于 CPU 时钟。因此，如果将其用作 WDT 的时钟源，即便是 CPU 时钟停止工作，仍然可以检测到挂起程序。如果低速内部振荡器用作 8 位定时器 H1 的计数时钟源，则即便是待机状态，8 位定时器 H1 仍然可以工作。

表 5-4 显示了低速内部振荡器选作 WDT 时钟源和 8 位定时器 H1 的计数时钟源时的操作状态。图 5-9 显示了低速内部振荡器的状态转移情况。

表 5-4. 低速内部振荡器的状态操作

选项字节的设置		CPU 状态	WDT 状态	TMH1 状态
可由软件停止振荡	LSRSTOP = 1	操作模式	停止	停止
	LSRSTOP = 0		运行	运行
	LSRSTOP = 1	待机	停止	停止
	LSRSTOP = 0		停止	运行
不能停止		操作模式	运行	
		待机		

图 5-9. 低速内部振荡器的状态转移图



注 看门狗定时器(WDT)的时钟源从 f_x 或 f_{RL} 中进行选择提供，否则看门狗定时器会停止工作。需要了解更详细情况，请参阅 **第七章 看门狗定时器**。

第六章 8 位定时器H1

6.1 8 位定时器H1 功能

8 位定时器 H1 具有如下功能

- 间隔定时
- PWM 输出模式
- 方波输出

6.2 8 位定时器H1 的构造

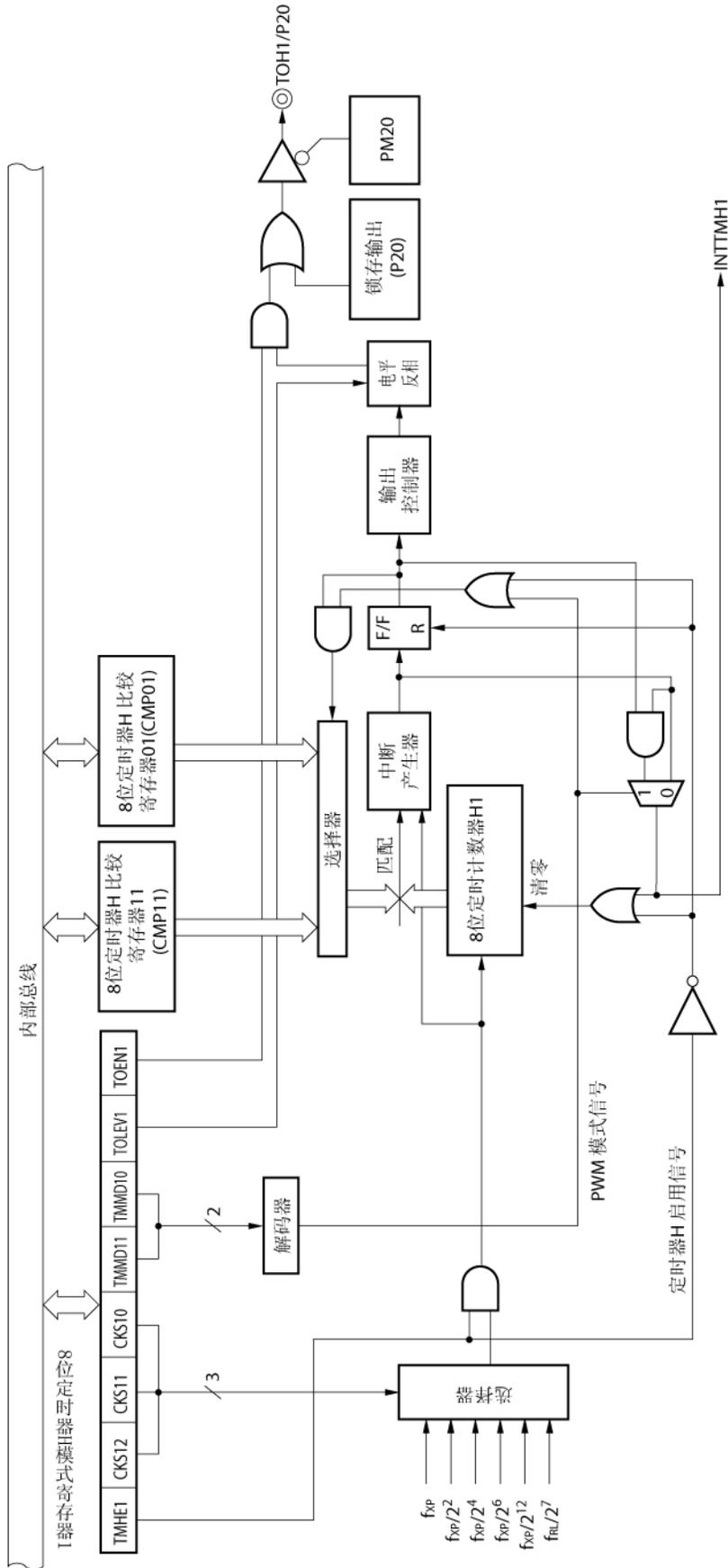
8 位定时器 H1 包括如下硬件

表 6-1. 8 位定时器 H1 的构造

名称	构造
定时寄存器	8 位定时计数器 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)

表 6-1 为功能模块表

图 6-1. 8 位定时器 H1 功能模块图



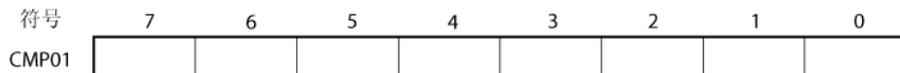
(1) 8 位定时器 H 比较寄存器 01 (CMP01)

输入 8 位存储器操作指令可实现该寄存器的读写操作。

输入复位信号可将该寄存器清零。

图 6-2. 8 位定时器 H 比较寄存器 01 的格式 (CMP01)

地址: FF0EH 复位后: 00H R/W



注意事项 在定时器计数期间，**CMP01** 无法进行重写操作。

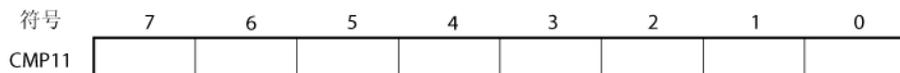
(2) 8 位定时器 H 比较寄存器 11 (CMP11)

输入 8 位存储器操作指令可实现该寄存器的读写操作。

复位信号可将该寄存器清零。

图 6-3. 8 位定时器 H 比较寄存器 01 的格式 (CMP11)

地址: FF0FH 复位后: 00H R/W



在定时器计数期间，可对 **CMP11** 进行重写。

如果在定时器计数期间对 **CMP11** 进行重写，**CMP11** 新值将在当前计数值与 **CMP11** 旧值匹配后替代旧值。如果在当前计数值与 **CMP11** 旧值匹配时，与 CPU 对 **CMP11** 写入时序发生冲突，则新 **CMP11** 值将于下一计数值和 **CPM11** 旧值匹配后生效。

注意事项 **PWM** 输出模式下，停止定时器计数后 (**TMHE1 = 0**)，再次开始定时器计数操作前 (**TMHE1 = 1**) 请确保先设置 **CMP11**，即使新值与旧值相同，也得再次设置。

6.3 8 位定时器 H1 的控制寄存器

以下三个寄存器用以控制 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。

输入 8 位存储器操作指令可设置该寄存器。

复位信号可将该寄存器清零。

图 6-4. 8 位定时器 H 模式寄存器 1 的格式 (TMHMD1)

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	启用定时器操作							
0	停止定时器计数操作 (计数器清零)							
1	禁用定时器计数操作 (以输入时钟开始计数操作)							
CKS12	CKS11	CKS10	计数时钟 (f _{CNT}) 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 (TYP.))					
上述除外			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						
上述除外		禁止设置						
TOLEV1	定时器输出电平控制 (默认模式)							
0	低电平							
1	高电平							
TOEN1	定时器输出控制							
0	禁用输出							
1	启用输出							

- 注意事项**
1. 当 **TMHE1 = 1**, 禁止对 **TMHMD1** 寄存器的其它位进行设置。
 2. **PWM 输出模式**下, 停止定时器计数后 (**TMHE1 = 0**), 再次开始定时器计数操作前 (**TMHE1 = 1**) 请确保先设置 **8 位定时器 H 比较寄存器 CMP11**, 即使新值与旧值相同, 也得再次设置。

- 备注**
1. f_{XP}: 外围硬件时钟的振荡频率。
 2. f_{RL}: 低速内部时钟的振荡频率。
 3. 括号内的值适用于 f_{XP} = 10 MHz, f_{RL} = 240 kHz (典型值)。

(2) 端口模式寄存器 2 (PM2)

当使用 P20/TOH1 引脚作为定时器输出时，PM20 清零，P20 上的输出锁存值也清零。

1 位或 8 位存储器操作指令可设置 PM2 寄存器。

复位信号可将该寄存器设置为 FFH。

图 6-5. 端口模式寄存器 2 格式 (PM2)

地址: FF22H 复位后: FFH R/W

编号	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 to 3)
0	输出模式 (输出缓存开启)
1	输入模式 (输出缓存关闭)

6.4 8 位定时器H1 操作**6.4.1 间隔定时/方波输出模式下的操作**

当 8 位定时计数器 H1 和比较寄存器 01 (CMP01) 匹配时，将生成中断请求信号 (INTTMH1)，此时 8 位定时计数器被清零。

间隔定时模式下不使用比较寄存器 11 (CMP11)。由于该模式下不对 8 位定时寄存器 H1 和 CMP11 寄存器的匹配情况进行检测，因此即使设置了 CMP11 寄存器，定时器输出也不受影响。

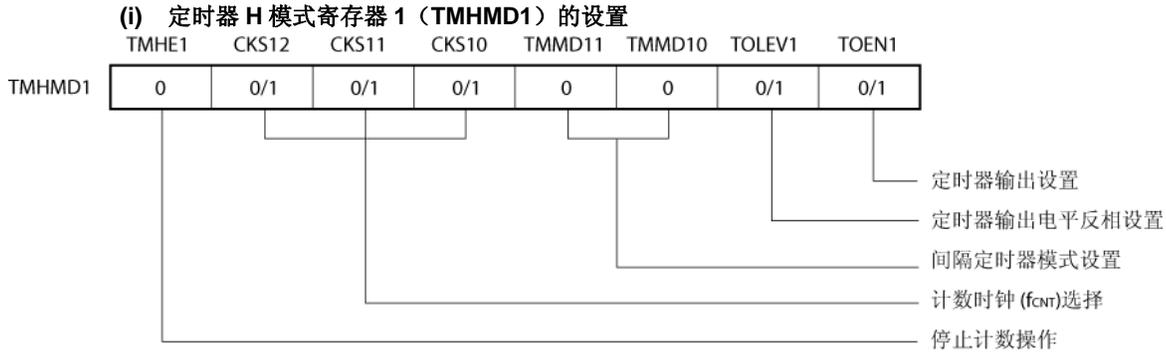
通过将定时器 H 模式寄存器 (TMHMD1) 的第 0 位 (TOEN1) 置 1，可从 TOH1 端口输出任意频率的方波 (占空比为 50%)。

(1) 使用实例

重复产生相同间隔的 INTTMH1 信号。

<1> 设置各寄存器。

图 6-6. 间隔定时/方波输出操作时的寄存器设置

**(ii) CMP01 寄存器设置**

- 比较参考值 (N)

<2> 当 TMHE1 = 1, 计数操作开始。

<3> 当 8 位定时计数器 H1 和 CMP01 寄存器匹配时, INTTMH1 信号产生, 8 位定时计数器 H1 被清零。

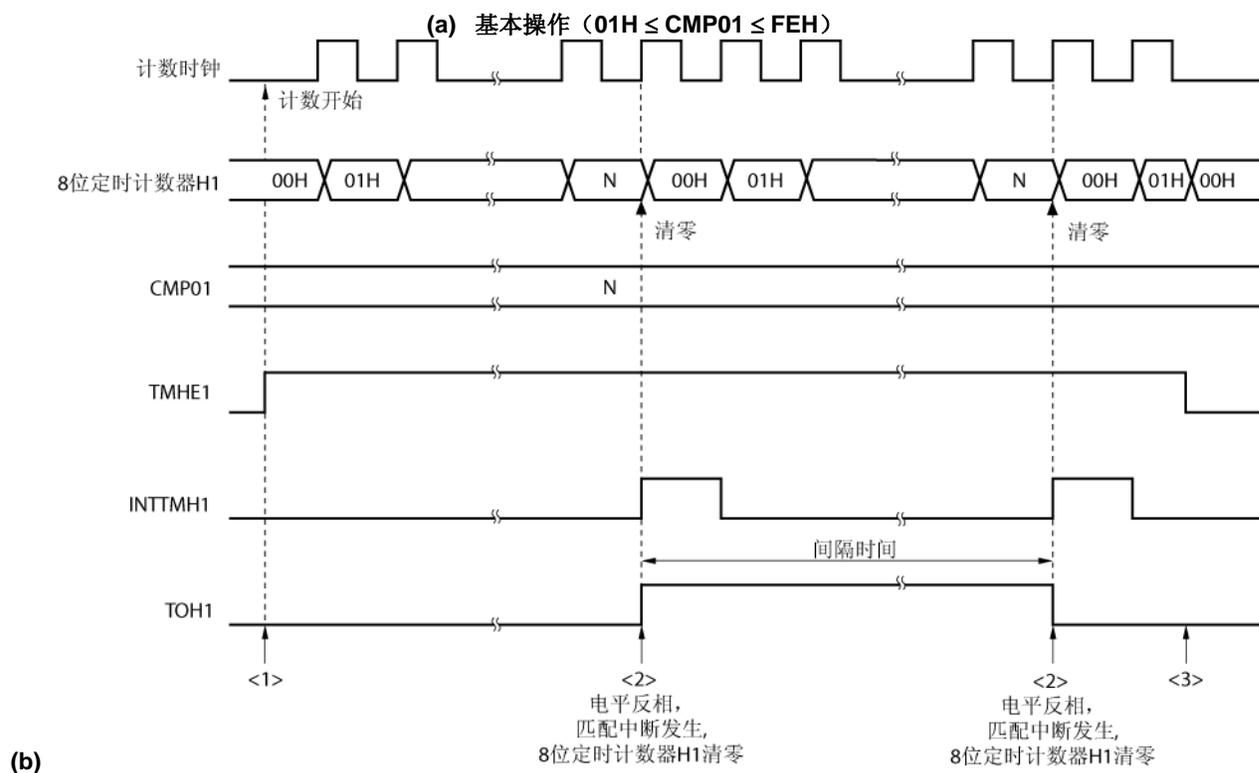
$$\text{间隔时间} = (N + 1)/f_{\text{CNT}}$$

<4> 间隔时间到后, 再次产生 INTTMH1 信号。将 TMHE1 清零可停止计数操作。

(2) 时序图

间隔定时/方波输出操作的时序如下图所示。

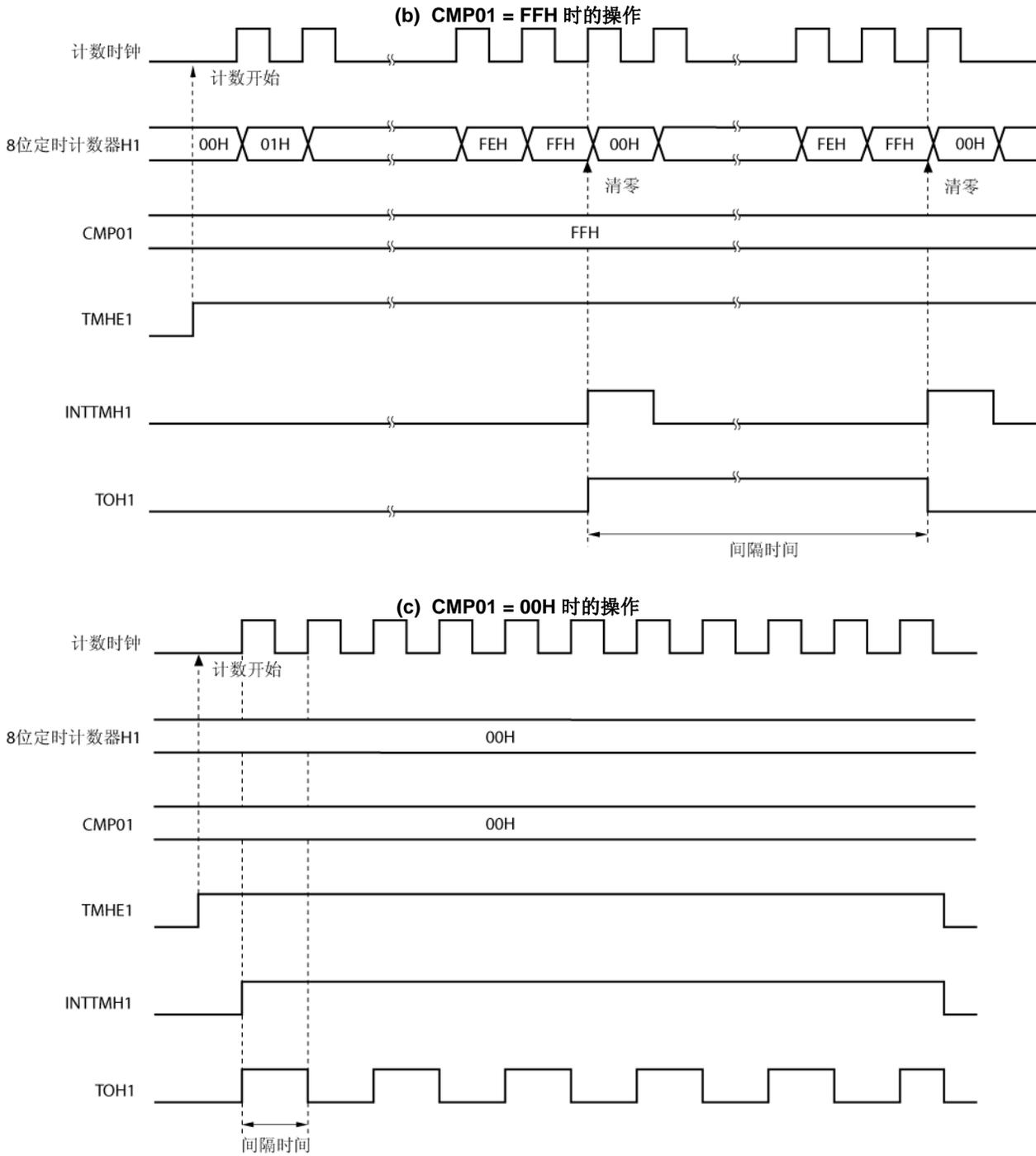
图 6-7. 间隔定时/方波输出操作的时序 (1/2)



- <1> 通过将 TMHE1 位置 1，开启计数操作。计数操作开始后不超过 1 个时钟周期，计数时钟开始计数。
- <2> 当 8 位定时计数器 H1 和 CMP01 寄存器数值匹配时，8 位定时计数器 H1 清零，TOH1 输出电平翻转，同时输出 INTTMH1 信号。
- <3> 计数器 H1 工作期间，通过将 TMHE1 位清 0，使 INTTMH1 信号和 TOH1 输出无效。如果从一开始就无效，则输出电平将保持不变。

备注 $01H \leq N \leq FEH$

图 6-7. 间隔定时/方波输出操作时序 (2/2)



6.4.2 PWM 输出模式下的操作

PWM 输出模式下，可输出任意周期任意占空比的脉冲。

8 位定时器比较寄存器 01 (CMP01) 控制定时器输出信号 (TOH1) 的周期。禁止在定时器工作期间重写 CMP01 寄存器。

8 位定时比较寄存器 11 (CMP11) 控制定时器输出信号 (TOH1) 的占空比。禁止在定时器工作期间重写 CMP11 寄存器。

PWM 输出模式下的操作如下所示。

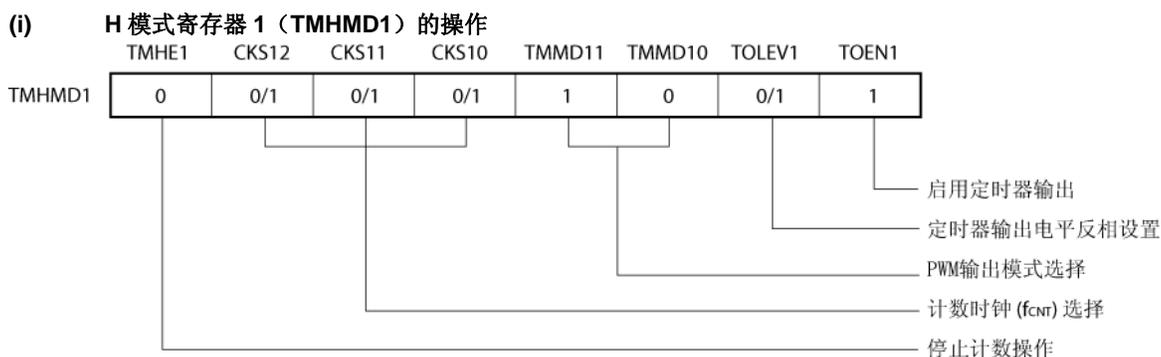
当 8 位定时计数器 H1 和 CMP01 寄存器数值匹配时，TOH1 开始输出，同时 8 位定时计数器被清 0。当 8 位定时计数器 H1 和 CMP11 寄存器匹配时，TOH1 输出无效。

(1) 使用实例

PWM 输出模式下，可输出任意周期任意占空比的脉冲。

<1> 设置各寄存器。

图 6-8. PWM 输出模式下的寄存器设置



(ii) CMP01 寄存器的设置

- 比较参考值 (N)：设置周期

(iii) CMP11 寄存器的设置

- 比较参考值 (N)：设置占空比

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> 计数操作开始后，CMP01 寄存器将首先进行比较。当 8 位定时计数器 H1 和 CMP01 寄存器数值匹配时，8 位定时计数器清零，生成中断请求信号 (INTTMH1)，TOH1 输出有效。同时，CMP11 取代 CMP01 与 8 位定时计数器 H1 进行比较。

- <4> 当 8 位定时计数器 H1 和 CMP11 寄存器数值匹配时, TOH1 输出无效, 同时 CMP11 取代 CMP01 与 8 位定时计数器 H1 进行比较。此时, 8 位定时计数器不清零, 且不产生 INTTMH1 信号。
- <5> 过程 <3> 和 <4> 重复执行, 输出任意占空比的脉冲信号。
- <6> 通过将 TMHE1 清零, 停止计数操作。

假设 CMP01 寄存器的设置值为 N, CMP11 寄存器的值为 M, 计数时钟频率为 f_{CNT} , 则 PWM 脉冲的输出周期和占空比如下。

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

- 注意事项**
1. PWM 输出模式下, 可在定时器计数期间重新设置 CMP11 寄存器的值。但是, 新值写入后需要三个或三个以上工作时钟周期 (使用 TMHMD1 寄存器的 CKS12 到 CKS10 位来选择) 才能起作用。
 2. 停止定时器计数后 (TMHE1 = 0), 再次开始定时器计数操作前 (TMHE1 = 1) 请确保先设置 CMP11, 即使新值与旧值相同, 也必须再次设置。

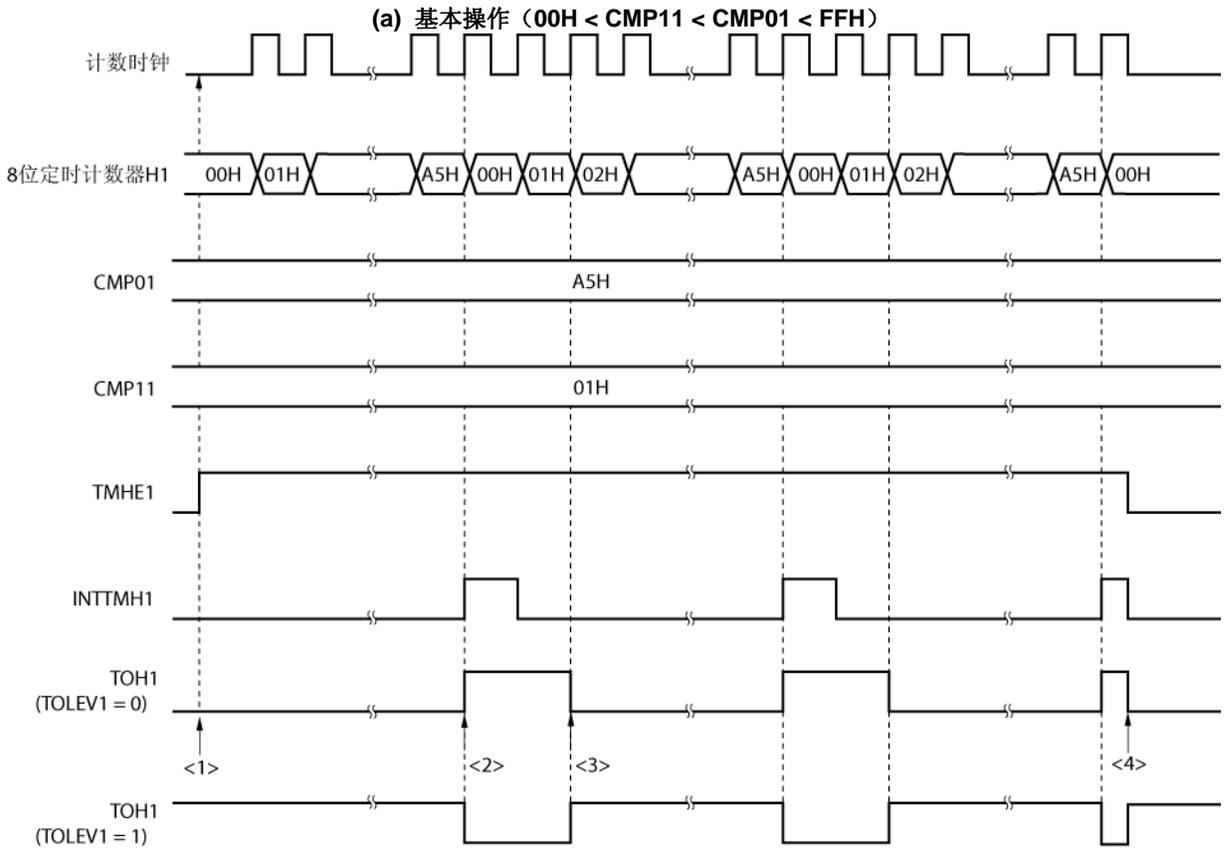
(2) 时序图

PWM 输出模式下的工作时序如下图所示。

注意事项 请确定 CMP11 寄存器设置值 (M) 和 CMP01 寄存器设置值 (N) 位于如下范围内。

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

图 6-9. PWM 输出模式下的操作时序 (1/4)



- <1> 通过将 **TMHE1** 位置 1，开始计数操作。计数操作开始一个时钟周期后，8 位定时计数器开始计数。此时，**TOH1** 保持无效（当 **TOLEV1 = 0** 时）。
- <2> 当 8 位定时计数器 **H1** 和 **CMP01** 寄存器数值匹配时，**TOH1** 输出电平翻转，8 位定时计数器 **H1** 清零，同时输出 **INTTMH1** 中断信号。
- <3> 当 8 位定时计数器 **H1** 和 **CMP11** 寄存器数值匹配时，**TOH1** 输出电平再次翻转。此时，不对 8 位定时计数器进行清零操作，且不输出 **INTTMH1** 中断信号。
- <4> 在定时计数器工作时，将 **TMHE1** 位清零，**INTTMH1** 信号和 **TOH1** 输出无效。

图 6-9. PWM 输出模式下的工作时序 (2/4)

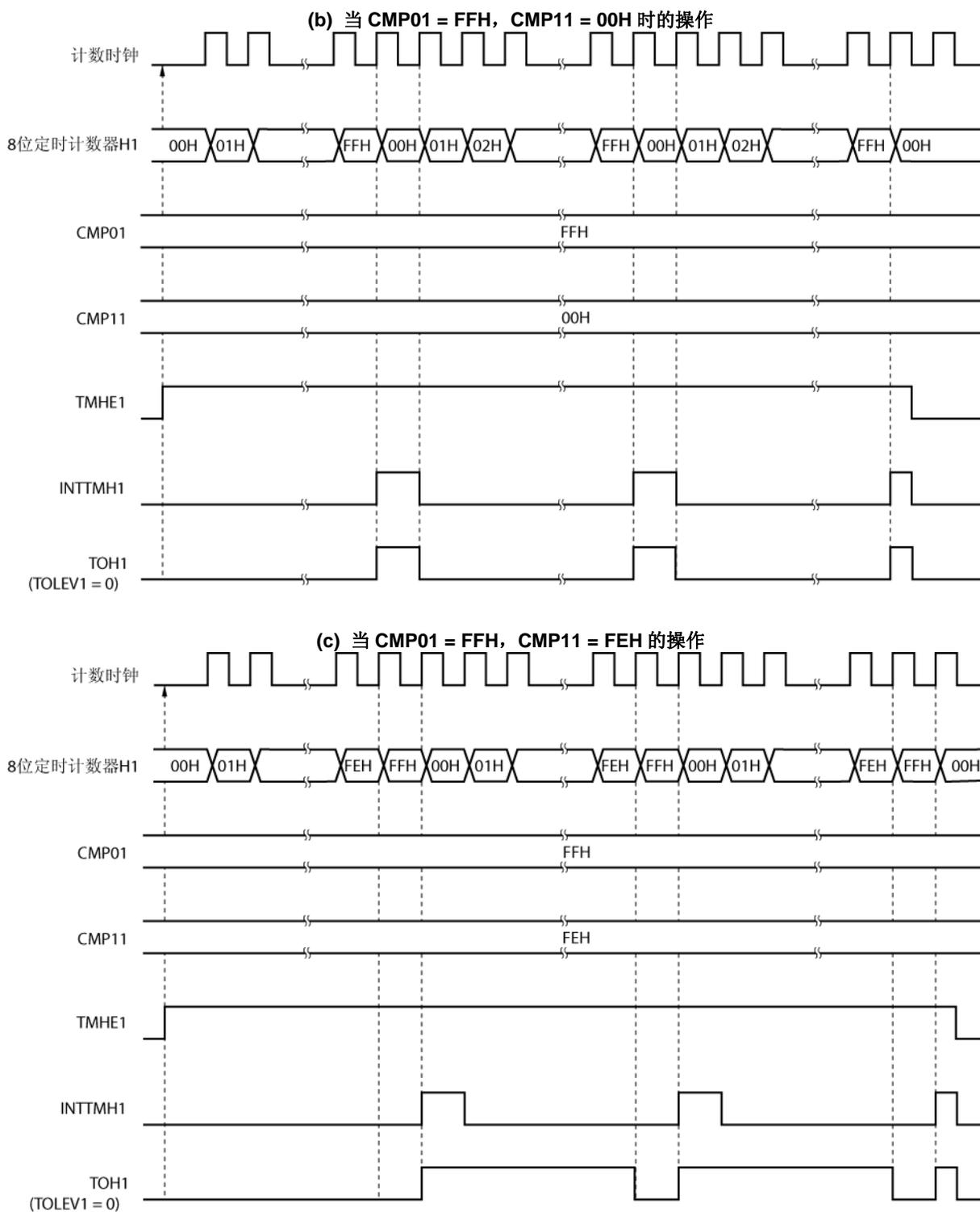


图 6-9. PWM 输出模式下的工作时序 (3/4)

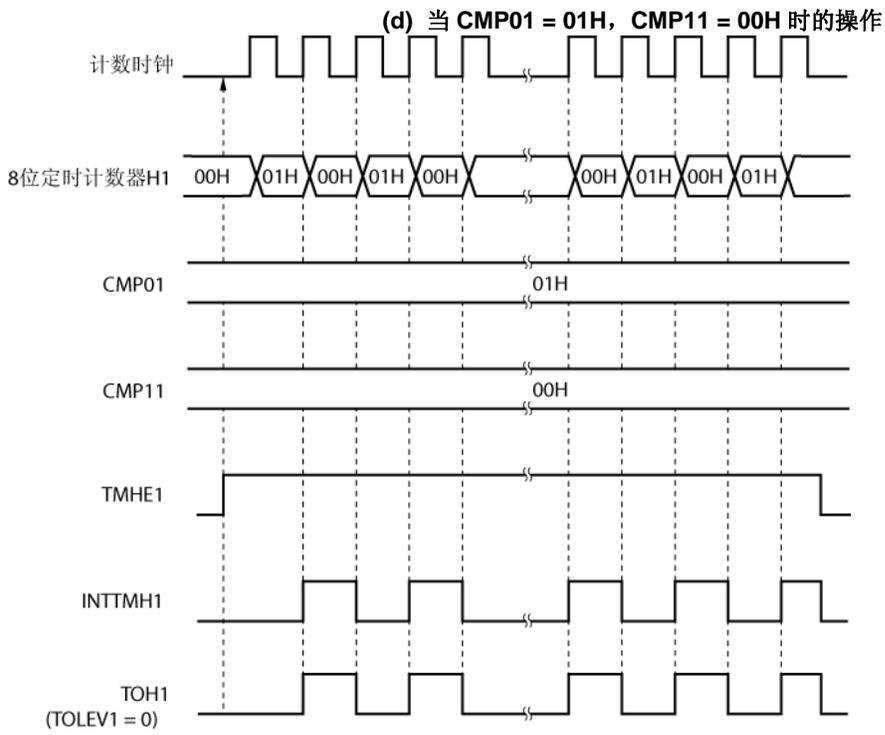
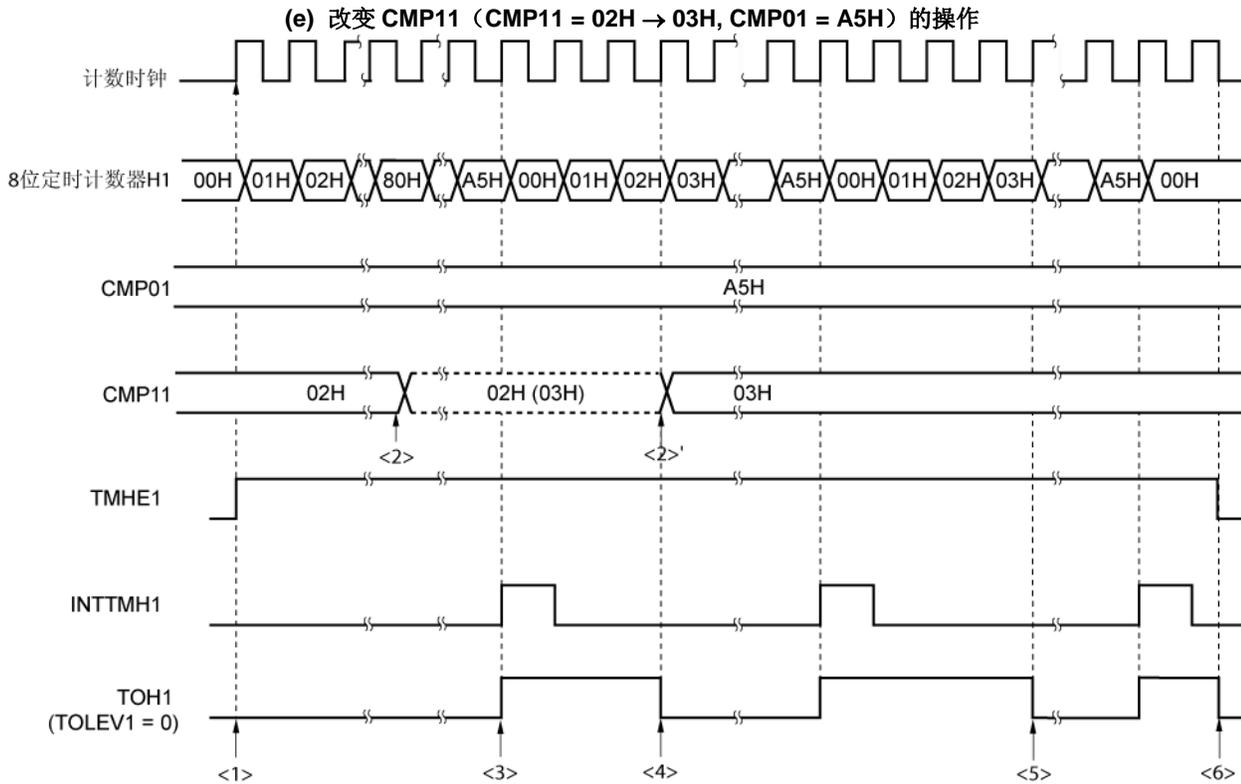


图 6-9. PWM 输出模式下的工作时序 (4/4)



- <1> 通过将 **TMHE1** 位置 1, 开始计数操作。计数操作开始一个时钟周期后, 8 位定时计数器开始计数。此时, **TOH1** 保持无效 (当 **TOLEV1 = 0** 时)。
- <2> 在计数器工作期间, 可以重新设置 **CMP11** 寄存器的值。该操作与计数时钟同步。
- <3> 当 8 位定时计数器 **H1** 和 **CMP01** 寄存器数值匹配时, **TOH1** 输出电平有效, 8 位定时计数器 **H1** 被清零, 同时输出 **INTTMH1** 信号。
- <4> 如果重新设置 **CMP11** 寄存器的值, 则新值将首先被锁存, 而不是立即传送到寄存器。当 8 位定时计数器 **H1** 和旧的 **CMP11** 寄存器值匹配时, 新值进入 **CMP11** 寄存器, 完成设置。 (<2>')。但是, 从设置 **CMP11** 到新值进入硬件寄存器需要三个或三个以上时钟周期。如果匹配发生在三个时钟周期以内, 则新值将无法写入寄存器。
- <5> 当 8 位定时计数器 **H1** 和改变后的 **CMP11** 寄存器数值匹配时, **TOH1** 输出电平变为无效。此时, 不对 8 位定时计数器进行清零, 且不输出 **INTTMH1** 信号。
- <6> 在定时计数器 **H1** 工作期间, 将 **TMHE1** 位清零, **INTTMH1** 信号和 **TOH1** 输出无效。

第七章 看门狗定时器

7.1 看门狗定时器的功能

看门狗定时器用于检测错误程序循环。一旦发现错误程序循环，将产生内部复位信号。

当看门狗定时器引发复位时，复位控制标识寄存器（RESF）的第四位（WDTRF）将被置1。RESF 详见第10章复位功能。

表 7-1. 看门狗定时器环路检测时间

环路检测时间	
低速内部振荡时钟工作时	系统时钟工作时
$2^{11}/f_{RL}$ (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} : 低速内部振荡时钟振荡频率
 2. f_x : 系统时钟振荡频率
 3. 括号内的值适用于 $f_{RL} = 480$ kHz (最大值) 且 $f_x = 10$ MHz

通过设置片上低速内部振荡器的选项字节（如表 7-2 所示）可改变看门狗定时器（WDT）的工作模式。

表 7-2. 选项字节的设置与看门狗定时器的工作模式

	选项字节的设置	
	低速内部振荡器无法停止	低速内部振荡器可通过软件停止
看门狗定时器时钟源	固定为 f_{RL} ^{#1} .	<ul style="list-style-type: none"> • 可通过软件选择 (f_x, f_{RL} 或停止) • 当按下并释放复位时: f_{RL}
复位后的操作	经过最大间隔时间 ($2^{18}/f_{RL}$) 后开始工作	经过最大间隔时间 ($2^{18}/f_{RL}$) 后开始工作
工作模式选择	间隔时间大小只能更改一次	时钟选择/间隔只能更改一次
特点	看门狗定时器无法停止	看门狗定时器可以停止 ^{#2} .

- 注**
1. 只要电源处于供电状态，低速内部振荡器将不能停止（除非在复位周期内）。
 2. 看门狗定时器工作时钟可被停止需满足的条件因看门狗定时器时钟源的不同而不同。
 - <1> 如果时钟源为 f_x ，则满足以下条件时看门狗定时器的工作时钟可被停止。
 - 当 f_x 停止时
 - 在 HALT/STOP 模式下
 - 振荡稳定时间内
 - <2> 如果时钟源 f_{RL} ，则满足以下条件时看门狗定时器的工作时钟可被停止。
 - 如果 CPU 时钟为 f_x ，且 f_{RL} 在 STOP 指令执行之前已被程序停止
 - 在 HALT/STOP 模式下

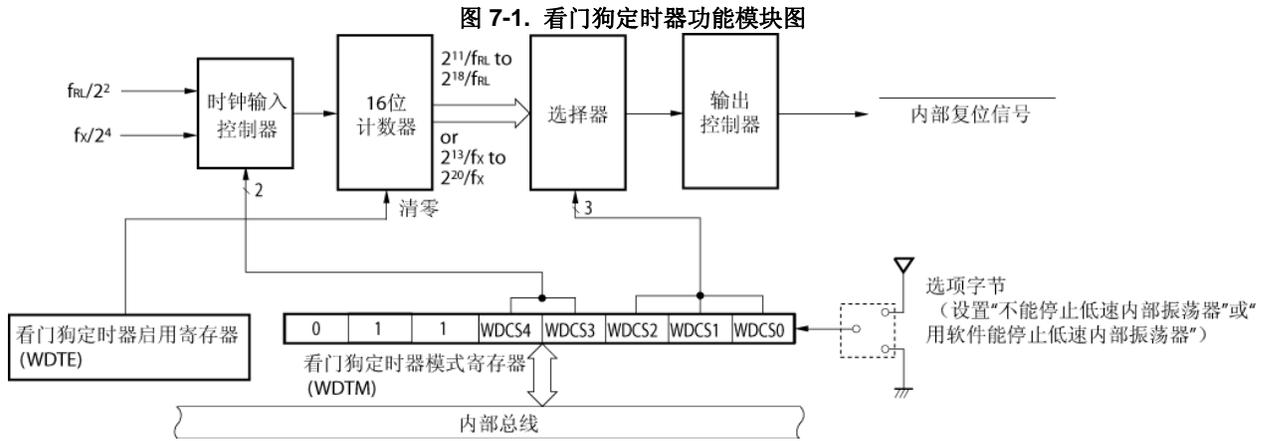
- 备注**
1. f_{RL} : 低速内部振荡时钟振荡频率
 2. f_x : 系统时钟振荡频率

7.2 看门狗配置

看门狗定时器包含如下硬件。

表 7-3. 看门狗定时器配置

名称	构造
控制寄存器	看门狗定时器模式寄存器 (WDTM) 看门狗定时器使能寄存器 (WDTE)



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

7.3 看门狗定时器控制寄存器

看门狗定时器受如下寄存器控制。

- 看门狗定时器模式寄存器 (WDTM)
- 看门狗使能寄存器 (WDTE)

(1) 看门狗定时器模式寄存器 (WDTM)

该寄存器用于设置看门狗定时器的溢出时间和工作时钟。

可通过 8 位存储器操作指令操作该寄存器，且可以进行多次读操作，但只能在复位释放后进行一次写操作。复位信号产生后，寄存器将被设置为 67H。

图 7-2. 看门狗定时器模式寄存器的格式 (WDTM)

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}	工作时钟选择
0	0	低速内部振荡时钟 (f _{RL})
0	1	系统时钟 (f _x)
1	×	看门狗定时器已停止工作

WDCS2 ^{注2}	WDCS1 ^{注2}	WDCS0 ^{注2}	溢出时间设置	
			低速内部振荡时钟工作期间内	系统时钟工作期间内
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“低速内部振荡器无法停止”是通过选项字节设置的，则无法进行工作时钟选择。因为无论写入什么值，都将选择低速内部振荡时钟。
 2. 复位在最大周期后释放 (WDCS2, 1, 0 = 1, 1, 1)

注意事项 1. 将第 7, 6, 5 位分别设置为 0, 1 和 1。不要设置为其它值。

注意事项 2. 复位释放后，WDTM 仅能被 8 位存储器操作指令写入一次。如果试图第二次写入，将产生内部复位信号。但是，如果 WDCS4 和 WDCS3 相应被设置为“1”和“x”，且看门狗定时器已被停止，则即使执行以下操作也不会产生内部复位信号。

- 第二次写 WDTM
 - 发送 1 位存储器操作指令给 WDTE
 - 发送一个非“ACH”的值给 WDTE
3. 不能通过 1 位存储器操作指令设置 WDTM。
 4. 当使用 flash 存储器程序时，请为看门狗定时器设置溢出时间以确保溢出时间不会太短（例 1 个字节的写入：最少 200 μ s，1 个模块的删除：最少 10 ms）。

- 备注**
1. f_{RL}: 低速内部振荡时钟振荡频率
 2. f_x: 系统时钟振荡频率
 3. x: 无作用
 4. 括号内的数字适用于 f_{RL} = 480 kHz（最大值），f_x = 10 MHz 时。

(2) 看门狗定时器使能寄存器 (WDTE)

写 ACH 到 WDTE 可将看门狗定时器计数值清零，并重新开始计数。

可通过 8 位存储器操作指令设置该寄存器。

复位信号将把该寄存器设置为 9AH。

图 7-3. 看门狗定时器使能寄存器的格式 (WDTE)

地址: FF49H 复位后: 9AH R/W

符号	7	6	5	4	3	2	1	0
WDTE								

- 注意事项**
1. 如果将一个非 ACH 的值写入 WDTE，将产生一个内部复位信号。
 2. 如果通过 1 位存储器操作指令操作 WDTE，将产生一个内部复位信号。
 3. 从 WDTE 读回的值为 9AH（该值不同于写入值 ACH）。

7.4 看门狗定时器的工作

7.4.1 当通过选项字节选择“低速内部振荡器无法停止”时看门狗定时器的工作

看门狗定时器的工作时钟固定为低速内部振荡时钟。

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

以下为复位释放后，看门狗定时器的操作。

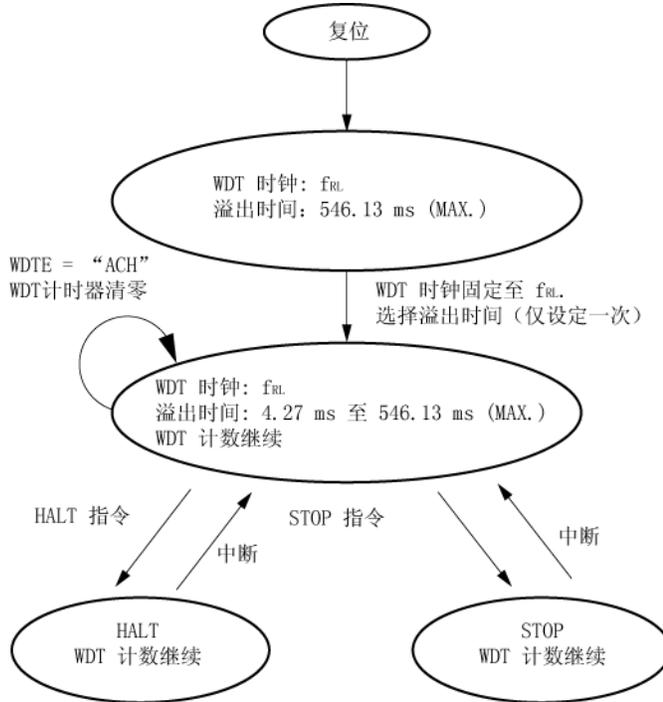
1. 复位释放后的状态如下所示。
 - 工作时钟：低速内部振荡时钟
 - 周期： $2^{18}/f_{RL}$ （546.13 ms：操作频率 $f_{RL} = 480$ kHz（最大值））
 - 计数开始
2. 以下内容应通过 8 位存储器操作指令^{#1,2}设置到看门狗定时器模式寄存器（WDTM）中。
 - 周期：通过第 2 至 0 位（WDCS2 至 WDCS0）来设置
3. 在以上步骤执行之后，向 WDTE 写入 ACH，将计数值清零，使能重新计数。

- 注**
1. 不能改变工作时钟（低速内部振荡时钟）。向 WDTM 的第 3 和 4 位（WDCS3，WDCS4）写入的任何值都将被忽略。
 2. 只要 WDTM 一被写入，则看门狗计数器的计数值将被清零。

注意 该模式下，即使在 STOP 指令执行时，也无法停止看门狗定时器的工作。对于 8 位定时器 H1，可选择低速内部振荡时钟作为其计数源，因此 STOP 指令执行后，看门狗定时器溢出前，需通过 TMH1 中断请求清除看门狗定时器的计数值。如果在 STOP 指令执行后，看门狗定时器溢出前，不进行计数值的清除工作，则将产生内部复位信号。

状态变换图表如下所示

图 7-4. 当通过选项字节选择“低速内部振荡器无法停止”时的状态转换图



7.4.2 当通过选项字节选择“低速内部振荡器可被停止”时看门狗定时器的工作

看门狗定时器的工作时钟可以是低速内部振荡时钟或系统时钟。

复位释放后，经过低速内部振荡时钟的最大周期（看门狗定时器模式寄存器（WDTM）的第2, 1, 0位（WDCS2, WDCS1, WDCS0）=1,1,1）后工作开始进行。

以下为复位释放后，看门狗定时器的操作。

1. 复位释放后的状态如下所示。
 - 工作时钟：低速内部振荡时钟
 - 周期： $2^{18}/f_{RL}$ （546.13 ms： $f_{RL} = 480$ kHz（最大值）的工作周期）
 - 计数开始
2. 以下内容应通过8位存储器操作指令^{#1,2,3}设置到看门狗定时器模式寄存器（WDTM）中。
 - 工作时钟：通过设置第3和4位（WDCS3和WDCS4）选择以下任何一种时钟
低速内部振荡时钟（ f_{RL} ）
系统时钟（ f_x ）
看门狗定时器操作停止
 - 周期：通过第2至0位（WDCS2至WDCS0）来设置
3. 在以上步骤执行之后，向WDTE写入ACH，将计数值清零，使能重新计数。

- 注
1. 只要WDTM一被写入，则看门狗计数器的计数值将被清零。
 2. 把第7, 6和5位分别设置为0, 1, 1。不要设置为其它值。
 3. 第一次写入时，如果通过将WDCS4置1, WDCS3置x来停止看门狗定时器的工作，则即使出现以下情况也不会产生内部复位信号。
 - 第二次写WDTM。
 - 发送1位存储器操作指令给WDTE。
 - 发送一个非“ACH”的值给WDTE。

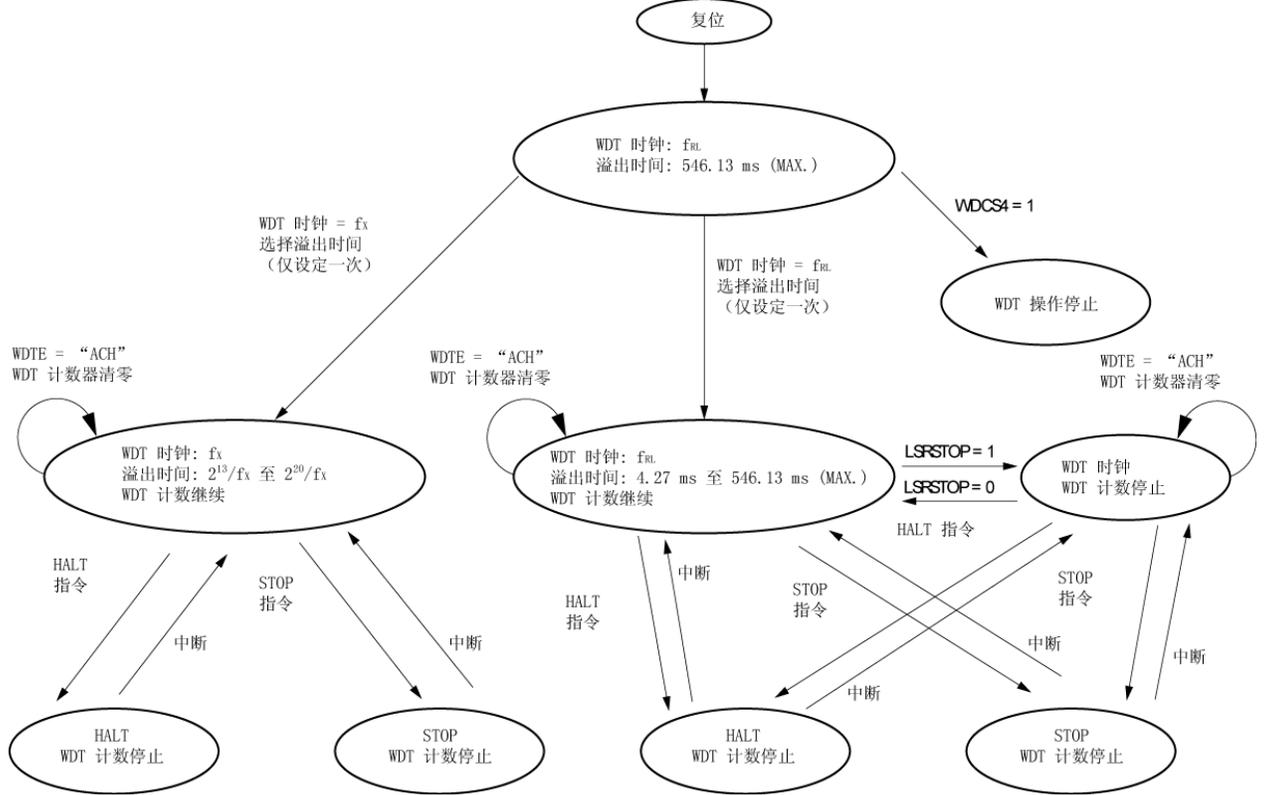
注意 该模式下，看门狗定时器在HALT/STOP指令执行期间停止工作。HALT/STOP模式指令释放后，计数再次启动，使用执行HALT/STOP指令前由WDTM设置的看门狗定时器的工作时钟。此时，计数值不会清零，而是保持原值不变。

STOP模式和HALT模式各种状态下看门狗定时器的工作情况，详见7.4.3 STOP模式下看门狗定时器的工作，

7.4.4 HALT模式下看门狗定时器的工作。

状态转换图如下所示。

图 7-5. 当通过选项字节选择“低速内部振荡器可被停止”时的状态转换图



7.4.3 STOP模式下看门狗定时器的工作（当通过选项字节选择“低速内部振荡器可被停止”时）

无论是采用系统时钟还是低速内部振荡时钟，在 STOP 指令执行期间，看门狗定时器都将停止计数。

(1) 当看门狗定时器工作时钟为系统时钟（fx）且 STOP 指令执行时

STOP 指令执行后，看门狗定时器停止工作。STOP 模式释放后，停止工作 34 μs （典型值），然后再次开始计数，并以看门狗定时器的工作时钟作为计数时钟，直至工作停止。此时，计数值不会被清零，而是保持原值不变。

图 7-6. STOP 模式下的工作（WDT 工作时钟：外围电路工作时钟）

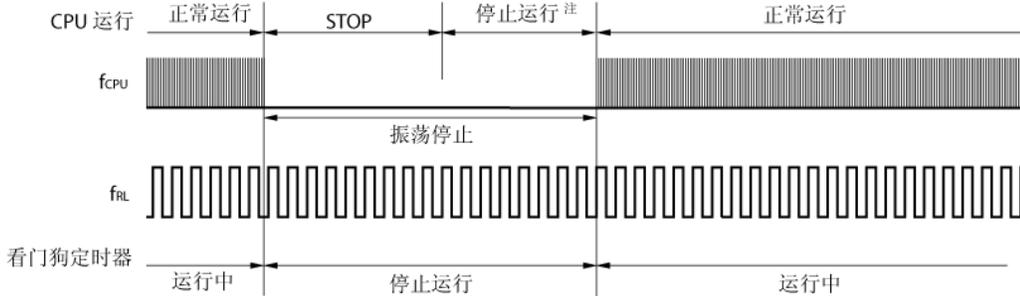


注 操作停止时长为 17 μs （最小值），34 μs （典型值），67 μs （最大值）

(2) 当看门狗定时器工作时钟为低速内部振荡时钟（fRL）且 STOP 指令执行时

STOP 指令执行后，看门狗定时器停止工作。STOP 模式释放后，停止工作 34 μs （典型值），然后再次开始计数，并以看门狗定时器的工作时钟作为计数时钟，直至工作停止。此时，计数值不会被清零，而是保持原值不变。

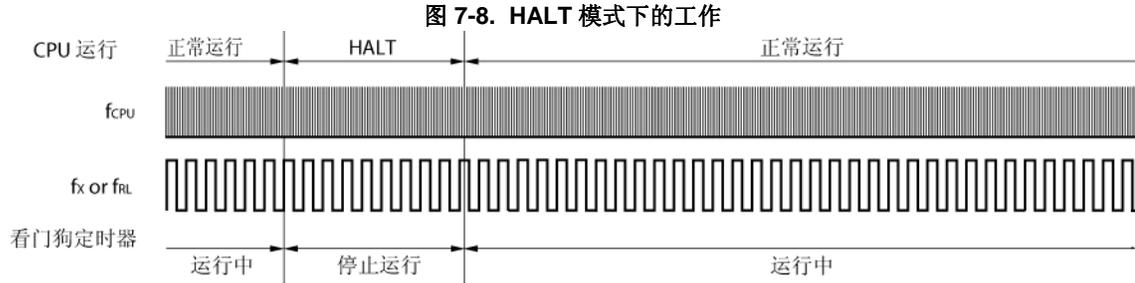
图 7-7. STOP 模式下的工作（WDT 工作时钟：低速内部振荡时钟）



注 操作停止时长为 17 μs （最小值），34 μs （典型值），67 μs （最大值）。

7.4.4 HALT模式下看门狗定时器的工作（当通过选项字节选择“低速内部振荡器可被停止”时）

无论是采用系统时钟（fx）还是低速内部振荡时钟（fRL），在 HALT 指令执行期间，看门狗定时器都将停止计数。HALT 模式指令释放后，计数器将再次启动，直至工作停止。此时计数值不会清零，而是保持原值不变。



第八章 中断功能

8.1 中断功能类型

有两类中断：可屏蔽中断和复位中断。

- **可屏蔽中断**

该中断受屏蔽控制。当中断请求产生，产生释放待机状态信号，且某个中断可被识别时，将按照矢量地址表中指定的地址开始执行程序（矢量中断服务）。同时产生多个中断请求时，按矢量中断服务中的优先级次序决定中断处理的先后顺序。关于优先级次序的说明，详见表 8-1。

有两类内部中断源和两类外部可屏蔽中断源。

- **复位中断**

通过复位信号，CPU 和 SFR 将恢复到其初始状态。复位信号产生的原因如表 8-1 所示。

当复位信号产生后，程序从 0000H 和 0001H 中指定的地址处开始执行。

8.2 中断源及其构造

共有 4 个可屏蔽中断源，以及 4 个复位中断源（见表 8-1）。

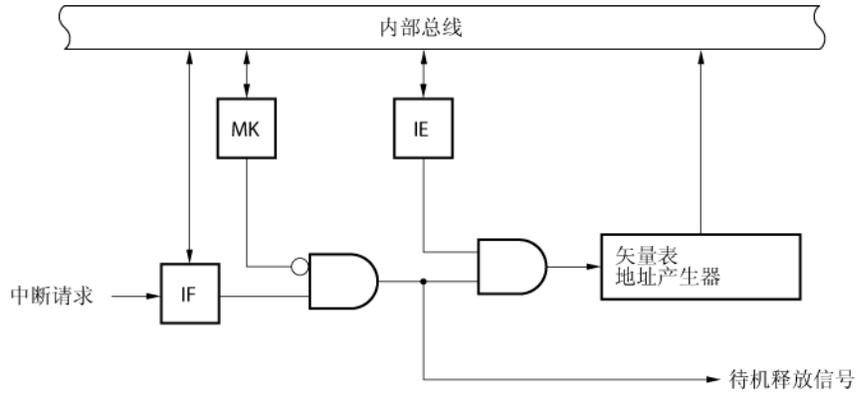
表 8-1. 中断源

中断类型	优先级 ^{注1}	中断源		内部/外部	矢量表地址	基本构造类型 ^{注2}
		名称	触发			
可屏蔽中断	1	INTLVI	低压检测 ^{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	内部	000CH	(A)
复位中断	-	RESET	复位输入	-	0000H	-
		POC	供电切断			
		LVI	低压检测 ^{注4}			
		WDT	WDT 溢出			

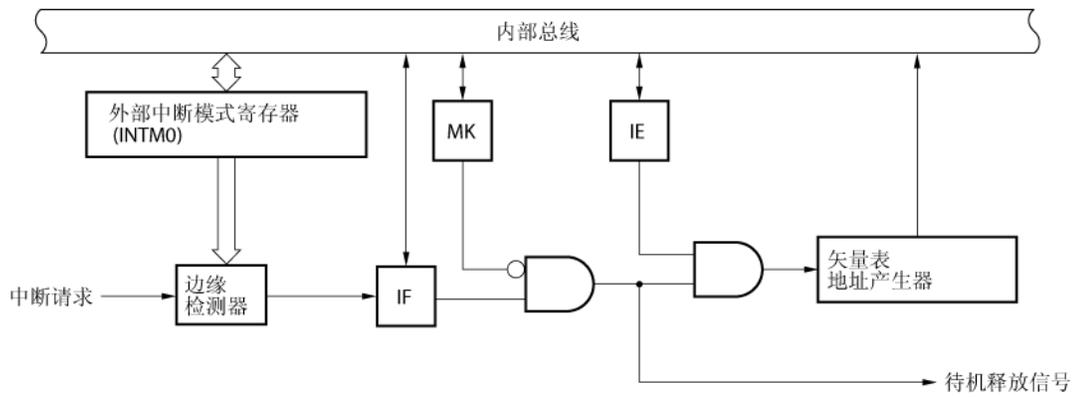
- 注
1. 当同时产生多个可屏蔽中断请求时，矢量中断服务的中断优先级将决定中断执行的先后次序，其中 1 优先级最高，4 最低。
 2. 基本构造类型 (A) 和 (B) 对应于图 8-1 中的 (A) 和 (B)。
 3. 当低压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 设为 0 时。
 4. 当低压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 设为 1 时。

图 8-1. 中断功能的基本构造

(A) 内部可屏蔽中断



(B) 外部可屏蔽中断



IF: 中断请求标志
 IE: 中断使能标志
 MK: 中断屏蔽标志

8.3 中断功能控制寄存器

中断功能受如下四类寄存器控制。

- 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 8-2 列出了中断请求及其相应的中断请求标志、中断屏蔽标志。

表 8-2. 中断请求信号及其相应标志

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

(1) 中断请求标志寄存器 0 (IF0)

收到中断请求或中断指令执行时，相应的中断请求标志将被置 1。当中断请求被响应或复位信号产生时，通过指令执行将标志清 0。

可通过 1 位或 8 位存储器操作指令设置 IF0。

复位信号将 IF0 清为 00H。

图 8-2. 中断请求标志寄存器的格式 (IF0)

地址: FFE0H 复位之后: 00H R/W

编号	7	6	5	<4>	<3>	<2>	<1>	0
IF0	0	0	0	TMIFH1	PIF1	PIF0	LVIIIF	0

xxIFx	中断请求标志
0	无中断请求信号
1	产生一个中断请求; 中断请求状态

注意事项 因为 P21 和 P32 还有作为外部中断输入的功能，所以通过指定端口功能的输出模式来改变输出电平时，中断请求标志置 1。因此，使用输出模式前应把中断屏蔽标志置 1。

(2) 中断屏蔽标志寄存器 0 (MK0)

中断屏蔽标志用于使能和关闭相应的可屏蔽中断。

可通过 1 位或 8 位存储器操作指令设置 MK0。

复位信号将 MK0 设置为 FFH。

图 8-3. 中断屏蔽标志寄存器 0 的格式 (MK0)

地址: FFE4H 复位后: FFH R/W

编号	7	6	5	<4>	<3>	<2>	<1>	0
MK0	1	1	1	TMMKH1	PMK1	PMK0	LVIMK	1

××MK×	中断服务控制
0	使能中断服务
1	关闭中断服务

注意事项 因为 P21 和 P32 还有作为外部中断输入的功能，所以通过指定端口功能的输出模式来改变输出电平时，中断请求标志将被置 1。因此，使用输出模式前应把中断屏蔽标志置 1。

(3) 外部中断模式寄存器 0 (INTM0)

该寄存器用以设置 INTP0 和 INTP1 的有效沿。

可通过 8 位存储器操作指令设置 INTM0。

复位信号将 INTM0 清零。

图 8-4. 外部中断模式寄存器 0 的格式 (INTM0)

地址: FFECH 复位后: 00H R/W

编号	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	下降沿有效
0	1	上升沿有效
1	0	禁止设置
1	1	上升和下降沿均有效

ES01	ES00	INTP 有效沿选择
0	0	下降沿有效
0	1	上升沿有效
1	0	禁止设置
1	1	上升和下降沿均有效

注意事项 1. 请确保将第 0, 1, 6, 7 位清零。

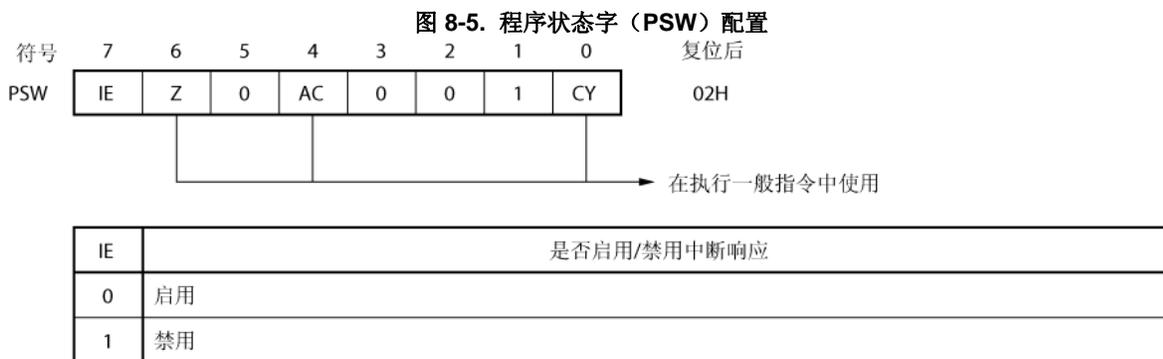
注意事项 2. 在设置 INTMO 寄存器前，请确保先设置相应中断屏蔽标志 ($\times\text{MK}\times = 1$)，关闭中断。设置好 INTMO 寄存器后，清除中断请求标志 ($\times\text{IF}\times = 0$) 然后再清除中断屏蔽标志 ($\times\text{MK}\times = 0$)，开启中断服务。

(4) 程序状态字 (PSW)

程序状态字用来保存指令执行结果和中断请求的当前状态。IE 标志用以使能和关闭可屏蔽中断，被映射到 PSW 内。

可通过 8 位操作指令、位操作指令以及专用指令 (EI 和 DI) 对 PSW 进行读写操作。当某个矢量中断被响应后，PSW 被自动保存到堆栈中，同时 IE 标识被清零。

复位信号将把 PSW 置为 02H。



8.4 中断服务工作原理

8.4.1 可屏蔽中断请求的识别原理

当中断请求标志被置 1 且相应中断屏蔽标志被清 0 时，可屏蔽中断请求能被响应。如果中断使能状态字有效 (当 IE 标识被置 1)，该请求作为矢量中断可以被响应。

可屏蔽中断请求产生后，开启矢量中断服务所需的时间如表 8-3 所示。

中断请求识别的时序参见图 8-7 和 8-8。

表 8-3. 从中断请求产生到开始中断服务所需的时间

最小时间	最大时间 ^注
9 个时钟周期	19 个时钟周期

注 如果中断请求紧靠在 BT 和 BF 指令之前产生，则此时等待时间最长。

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

当同时产生两个或两个以上中断时，将首先识别优先级最高的中断请求。

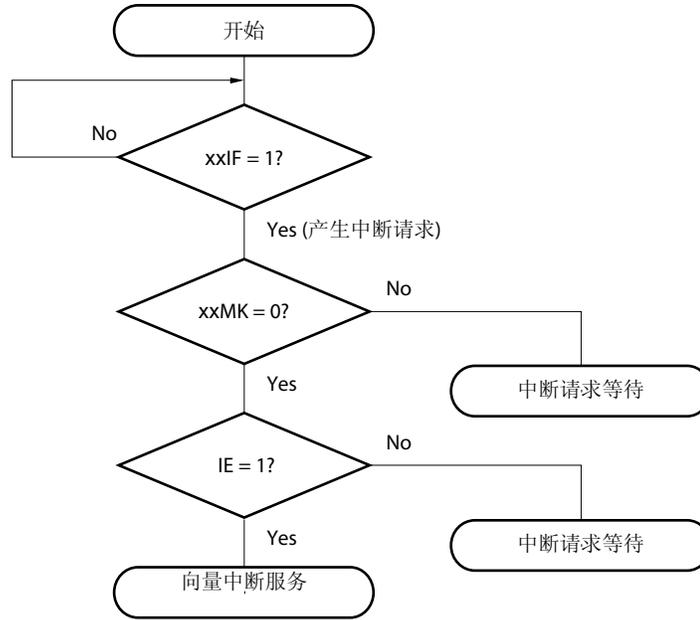
当允许响应中断请求时，处于挂起状态中的请求被响应。

图 8-6 显示了中断请求识别的算法。

当可屏蔽中断请求被识别后，PSW 和 PC 的内容依次保存到堆栈中，IE 标识清零，各中断请求在矢量表中的数据决定哪个加载到 PC，并执行跳转。

要退出中断服务，请使用 RETI 指令。

图 8-6. 中断请求识别过程的算法

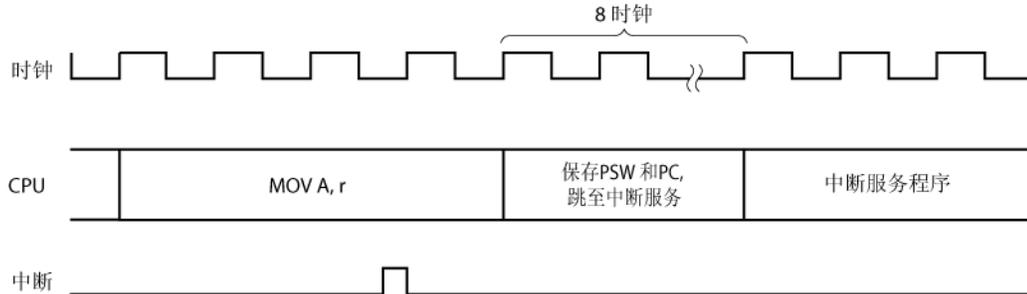


xxIF: 中断请求标志

xxMK: 中断屏蔽标志

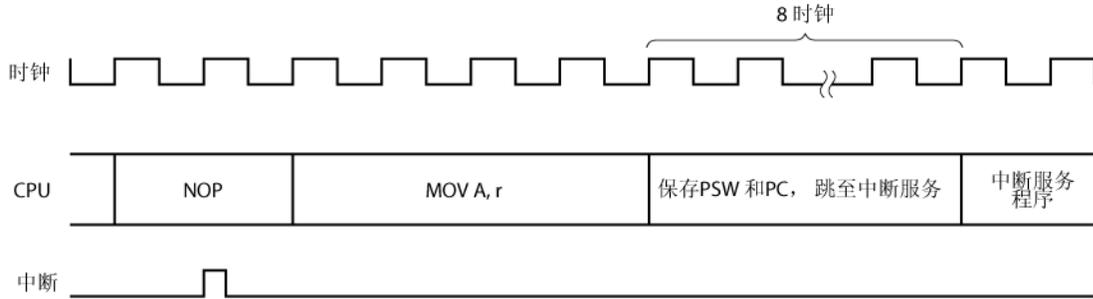
IE: 控制可屏蔽中断请求识别的标志（1 为使能，0 为关闭）

图 8-7. 中断请求识别时序（MOV A, r 的示例）



如果中断请求标志（xxIF）在指令时钟 n （ $n = 4$ 至 10 ）变为 $n-1$ 之前被置 1，则该中断要等到当前指令执行完成后才能被识别。例图 8-7 为 8 位数据转移指令 MOV A, r 的中断请求识别时序。该指令执行需要四个时钟周期，如果中断请求发生在第三个时钟周期处，则中断识别处理将在 MOV A, r 指令执行完之后进行。

图 8-8. 中断请求识别时序（当中断请求标志在指令执行的最后一个时钟周期时置 1）



如果中断请求标志（ $\times\times IF$ ）在指令的最后一个时钟周期内被置 1，则中断响应处理过程将在下条指令执行完之后进行。

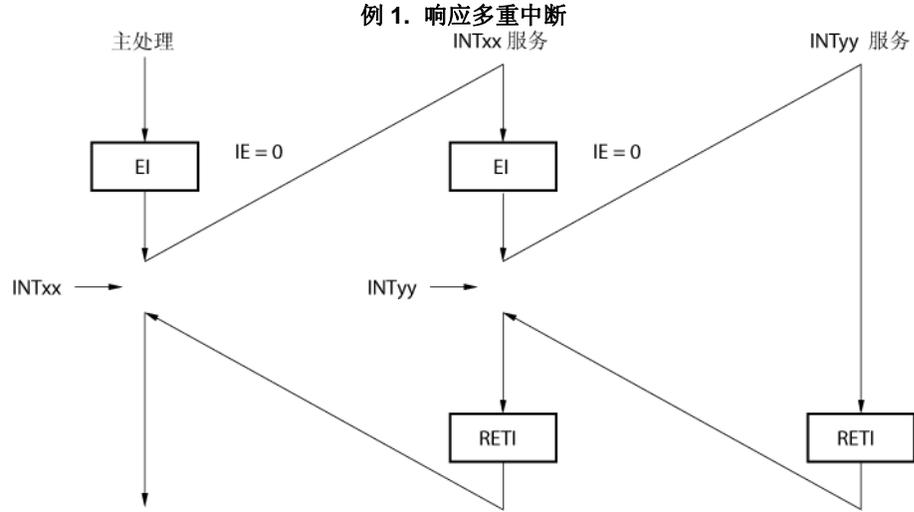
例图为中断请求标志在 NOP 指令（2 时钟指令）第二个时钟周期处被置 1 时的中断请求响应时序。该例中，MOV A, r 指令在 NOP 指令之后执行，然后再进行中断响应处理。

注意事项 在读写中断请求标志寄存器 0（IF0）或中断屏蔽标志寄存器 0（MK0）时，中断请求将被挂起。

8.4.2 多重中断服务

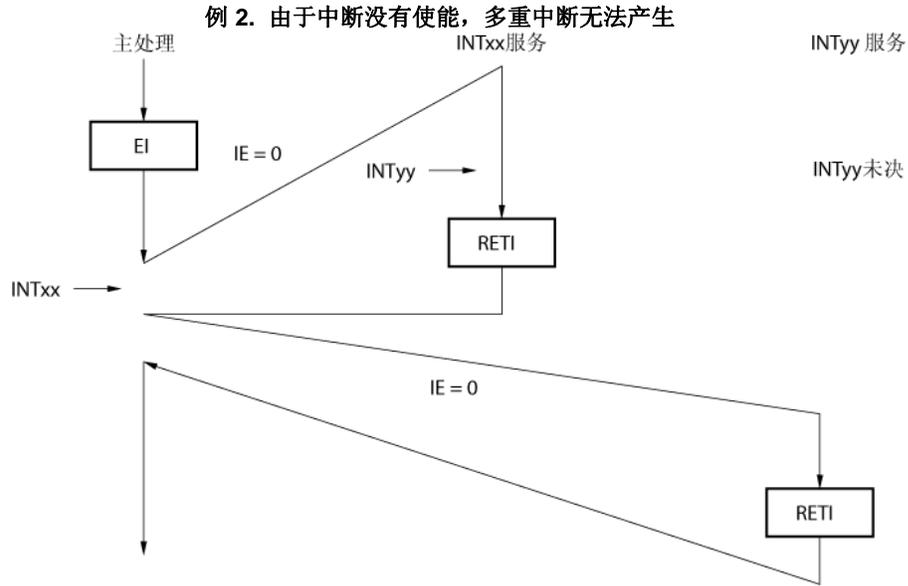
多重中断服务下，一个中断服务执行时同时要响应另一个中断。为了实现多重中断服务功能，必须使用中断屏蔽功能屏蔽掉高优先级中断，以使低优先级中断能够被识别。

图 8-9. 多重中断示例 (1/2)



在中断 INT_{xx} 服务程序中，中断请求 INT_{yy} 被响应，且产生多重中断。每次识别中断请求之前，将发出 EI 指令，释放中断屏蔽，且将中断请求识别使能状态位置 1。

注意事项 优先级最低的中断，多重中断可以被响应。



因为在中断 INT_{xx} 的服务程序中中断未使能 (EI 指令没发出)，因此中断请求 INT_{yy} 不响应，多重中断也不会产生。 INT_{yy} 请求挂起， INT_{xx} 服务程序执行完后，再响应 INT_{yy} 请求。

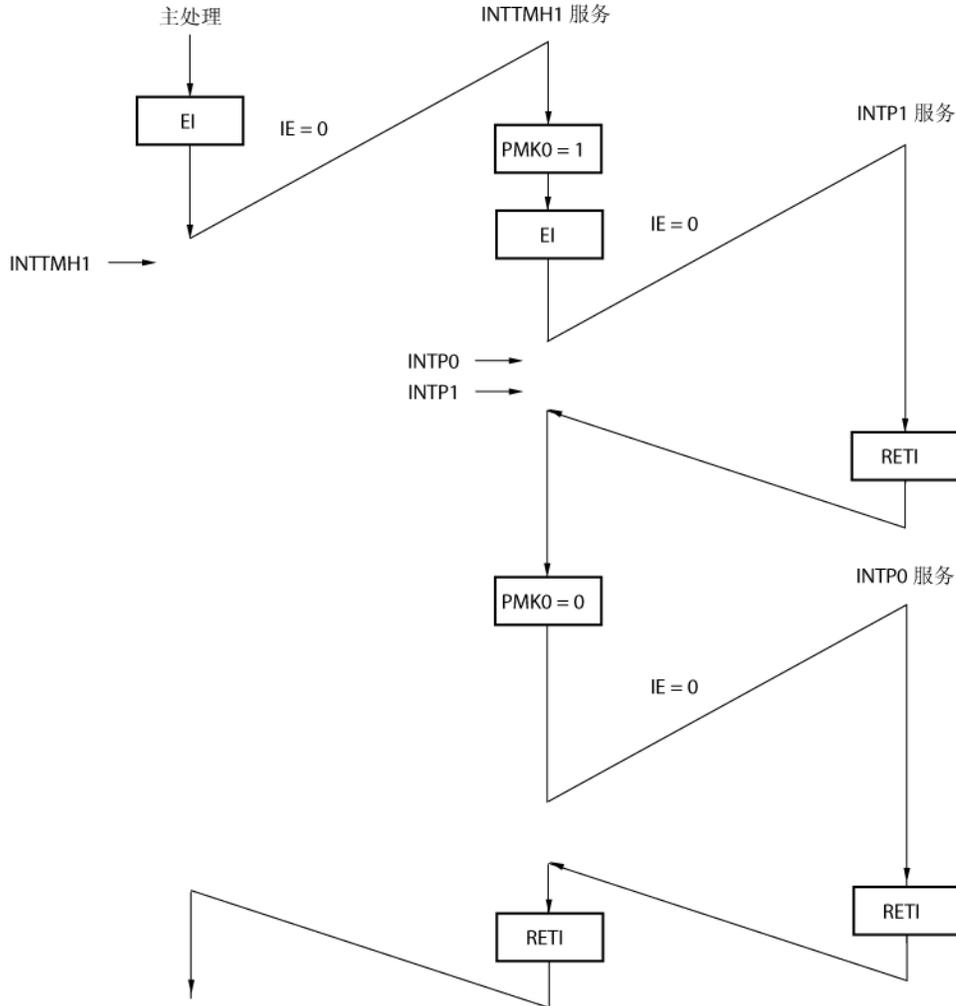
IE = 0: 禁止响应中断请求。

图 8-9. 多重中断示例 (2/2)

例 3. 优先级受多重中断控制

INTP0, INTP1 和 INTTMH1 设置矢量中断使能状态。

(中断优先级 INTP0 > INTP1 > INTTMH1 (参见表 8-1))



在 INTTMH1 中断服务中, 由于 INTP0 中断最初被屏蔽, 因此 INTP1 中断获得了优先执行权。

之后, 一旦 INTP0 中断屏蔽被释放, 将通过多重中断功能开始执行 INTP0 中断处理。

IE = 0: 禁止响应中断请求。

8.4.3 中断请求挂起

执行某些指令时可能挂起响应, 即使执行期间产生了中断请求 (可屏蔽中断和外部中断), 应等到下一指令执行完毕后再响应中断。以下为这类指令 (中断请求挂起指令)。

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

第九章 待机功能

9.1 待机功能及其配置

9.1.1 待机功能

表 9-1. 各工作状态下，工作时钟之间的关系

工作模式 \ 状态	低速内部振荡器			系统时钟	外围硬件工作时钟
	注 1	注 2			
		LSRSTOP = 0	LSRSTOP = 1		
复位	停止			停止	停止
STOP	工作	工作 ^{注 3}	停止	停止	停止
HALT				工作	工作

- 注
1. 当通过选项字节选择“低速内部振荡器不能被停止”时
 2. 当选择“低速内部振荡器可被程序停止”时，可通过 LSRSTOP 停止低速内部振荡器的工作。
 3. 如果以低速内部振荡器时钟作为看门狗定时器的时钟，则看门狗定时器将停止工作。

注意事项 只有当选择字节被设置为“低速内部振荡器可被程序停止”时，LSRSTOP 设置才会有效。

备注 LSRSTOP：低速内部振荡模式寄存器（LSRCM）第 0 位。

待机功能旨在减小系统的工作电流。以下为两种可用的模式。

(1) HALT 模式

执行 HALT 指令，将开启 HALT 模式，CPU 工作时钟停止。系统时钟振荡器继续振荡。如果在进入 HALT 模式之前低速内部振荡器已经开始工作，则低速内部振荡器将继续振荡（参见表 9-1. 低速内部振荡器时钟的工作情况（“不能被停止”或“可被软件停止”）可通过选项字节来选择）。该模式下，工作电流的节省量不如 STOP 模式多，但对于收到中断请求便需立即开始工作的场合，以及工作频繁断续的场合，使用 HALT 模式更有效。

(2) STOP 模式

执行 STOP 指令将开启 STOP 模式。该模式下，系统时钟停止振荡，整个系统工作停止，因此减少了 CPU 工作电流。

因为该模式可被中断请求释放，因此适合于间歇式的工作场合。但是，STOP 模式释放后，会产生操作停止时间^注，如果需要通过中断请求立即开始处理工作，则请选择 HALT 模式。

注 工作停止时长为 17 μs （最小值），34 μs （典型值），67 μs （最大值）。

任意一种模式下，寄存器、标志、数据存储器的所有内容都会保持为进入待机模式之前的内容。I/O 端口输出锁存和输出缓存状态也都保持不变。

- 注意事项**
1. 当转换到 STOP 模式时，请确保在执行 STOP 指令前，先停止外围硬件的工作（除了工作时钟为低速内部振荡时钟的外围硬件）。
 2. 如果在进入 STOP 模式前低速内部振荡器就已经开始工作了，则 STOP 模式中无法停止低速内部振荡器工作。（参加表 9-1）

9.2 待机功能的工作原理

9.2.1 HALT 模式

(1) HALT 模式

执行 HALT 指令选定 HALT 模式。

HALT 模式下的工作状态如下表所示。

注意事项 由于中断请求信号可用于释放待机模式，因此如果某个中断源的中断请求标志置 1，且中断屏蔽标志清 0，则立即退出待机模式。

表 9-2. HALT 模式下的工作状态

名称		HALT 模式的设置	低速内部振荡器不能被停止 ^注	
			低速内部振荡器继续	低速内部振荡器可被停止 ^注
系统时钟			CPU 时钟被关闭	
CPU			停止工作	
端口（锁存）			选择 HALT 模式前的状态	
8 位定时器 H1	将计数时钟设置为 f_{XP} 到 $f_{XP}/2^{12}$ 间的一个值		可工作	
	将计数时钟设置为 $f_{RL}/2^7$		可工作	可工作 停止工作
看门狗定时器	“系统时钟”被选作工作时钟		禁止设置	停止工作
	“低速内部振荡器时钟”选作工作时钟		工作（继续工作）	停止工作
上电复位电路			一直工作	
低压检测器			可工作	
外部中断			可工作	

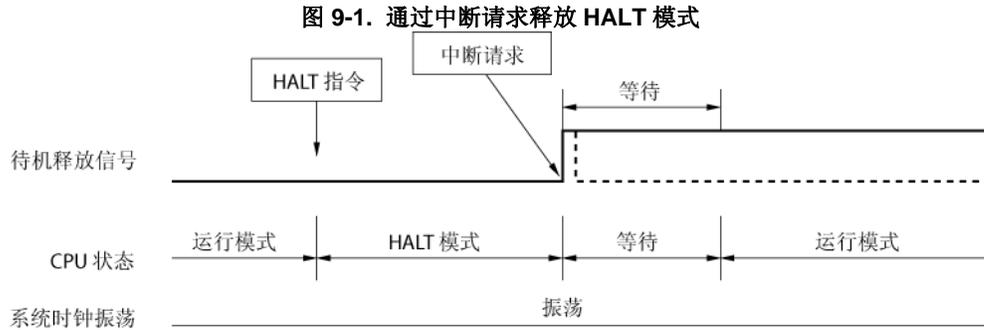
注 低速内部振荡器“不能被停止”或“能被软件停止”都是通过选项字节来选择的（选项字节参见第 13 章 选项字节）。

(2) HALT 模式释放

HALT 模式可通过以下两种途径释放。

(a) 通过未屏蔽的中断请求来释放

当产生一个未屏蔽中断请求时，HALT 模式释放。如果允许响应中断，则执行矢量中断服务。如果禁止响应中断，则将执行下一地址处的指令。



- 备注**
1. 虚线表示了释放待机模式的中断请求被响应的情况。
 2. 等待时间如下：
 - 当执行矢量中断服务时：11 至 13 个时钟周期
 - 当未执行矢量中断服务时：3 至 5 个时钟周期

(b) 通过复位信号来释放

当输入信号为复位信号时，退出 HALT 模式，和正常复位情况一样，程序跳转到复位矢量指定的地址处后，执行程序。



注 参见选项字节，工作停止时间为 277 μs （最小值），544 μs （典型值），1.075 ms（最大值）

表 9-3. HALT 模式下，对应于中断请求的操作

释放源	MK _{xx}	IE	操作
可屏蔽中断请求	0	0	执行下一地址指令
	0	1	执行中断服务程序
	1	×	保持 HALT 模式
复位信号	-	×	进行复位处理

×: 0 或 1 都行

9.2.2 STOP 模式

(1) STOP 模式设置及其工作状态

通过执行 STOP 指令设置 STOP 模式。

注意事项 由于中断请求信号可用于释放待机模式，因此如果某个中断源的中断请求标志被置 1，且中断屏蔽标志被清 0，则立即退出待机模式。这样，在 STOP 模式下，STOP 指令执行后，恢复正常工作模式，然后工作停止 34 μ s（典型值）。

STOP 模式下的工作状态如下表所示

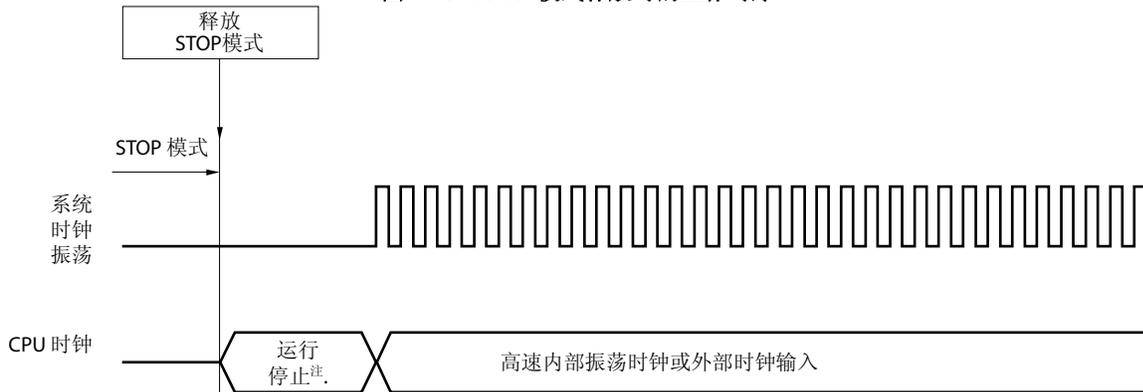
表 9-4. STOP 模式下的工作状态

名称		STOP 模式的设置	低速内部振荡器 不能被停止 ^注	低速内部振荡器可被停止 ^注	
				低速内部振荡继续	低速内部振荡停止
系统时钟			振荡停止		
CPU			停止工作		
端口（锁存）			选择 STOP 模式前的状态		
8 位定时器 H1	将计数时钟设置为 f_{XP} 到 $f_{XP}/2^{12}$ 间的一个值		停止工作		
	将计数时钟设置为 $f_{RL}/2^7$		可工作	可工作	停止工作
看门狗定时器	“系统时钟”被选作工作时钟		禁止设置		停止工作
	“低速内部振荡器时钟”被选作工作时钟		工作（继续工作）		停止工作
上电复位电路			一直工作		
低压检测器			可工作		
外部中断			可工作		

注 低速内部振荡器“不能被停止”或“能被软件停止”都是通过选项字节来选择的（选项字节参见第 13 章 选项字节）。

(2) STOP 模式的释放

图 9-3. STOP 模式释放时的工作时序



注 工作停止时长为 17 μs (最小值, 34 μs (典型值), 67 μs (最大值))。

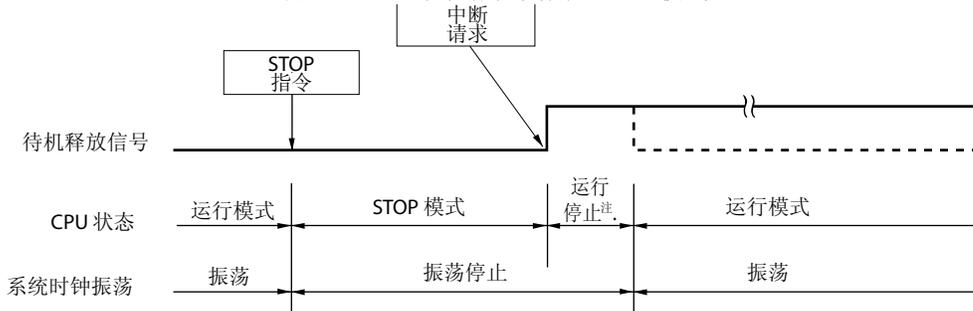
STOP 模式可通过以下两种途径释放。

(a) 通过未屏蔽的中断请求来释放

当一个未屏蔽中断请求 (8 位定时器 H1^注, 低电压检测, 外部中断请求) 产生时, STOP 模式被释放。振荡稳定后, 如果允许响应中断, 将执行矢量中断服务程序。如果禁止响应中断, 则执行下一地址处的指令。

注 仅当计数时钟设置为 $f_{RL}/2^7$ 时

图 9-4. 通过中断请求来释放 STOP 模式

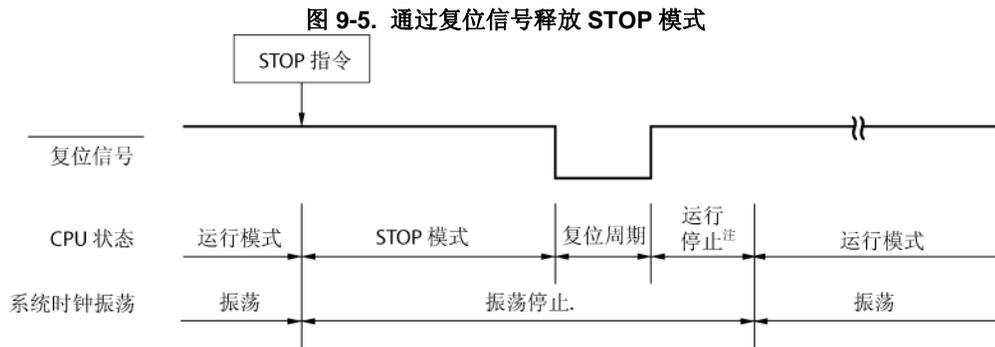


注 工作停止时长为 17 μs (最小值, 34 μs (典型值), 67 μs (最大值))。

备注 虚线表示释放待机模式的中断请求被响应的情况。

(b) 通过复位信号来释放

当复位信号产生时，STOP 模式释放，复位操作在振荡时钟稳定后开始执行。



注 参见选项字节，工作停止时间为 277 μs （最小值），544 μs （典型值），1.075 ms（最大值）

表 9-5. STOP 模式下，对应于中断请求的操作

释放源	MK _{xx}	IE	操作
可屏蔽中断请求	0	0	执行下一地址指令
	0	1	执行中断服务程序
	1	×	保持 STOP 模式
复位信号	-	×	复位处理

×: 0 或 1 都行

第十章 复位功能

共有以下四种复位信号。

- (1) 通过 $\overline{\text{RESET}}$ 引脚输入的外部复位信号。
- (2) 看门狗定时器溢出产生的内部复位。
- (3) 上电复位电路（POC）中供电电压与检测电压比较产生的内部复位。
- (4) 低电压检测器中（LVI）供电电压与检测电压比较产生的内部复位。

外部复位与内部复位在功能上没有区别。两种情况下，当复位信号产生时，程序均从地址 0000H 和 0001H 处开始执行。

当有低电平输入到 $\overline{\text{RESET}}$ 引脚中时，看门狗定时器溢出时，POC 和 LVI 电路检测出低压时，或者某任意硬件被设置为表 10-1 状态时，均会产生复位。复位信号产生期间以及复位释放后的振荡稳定期内，除了 P130（P130 为低电平输出）外，其它引脚均为高阻。

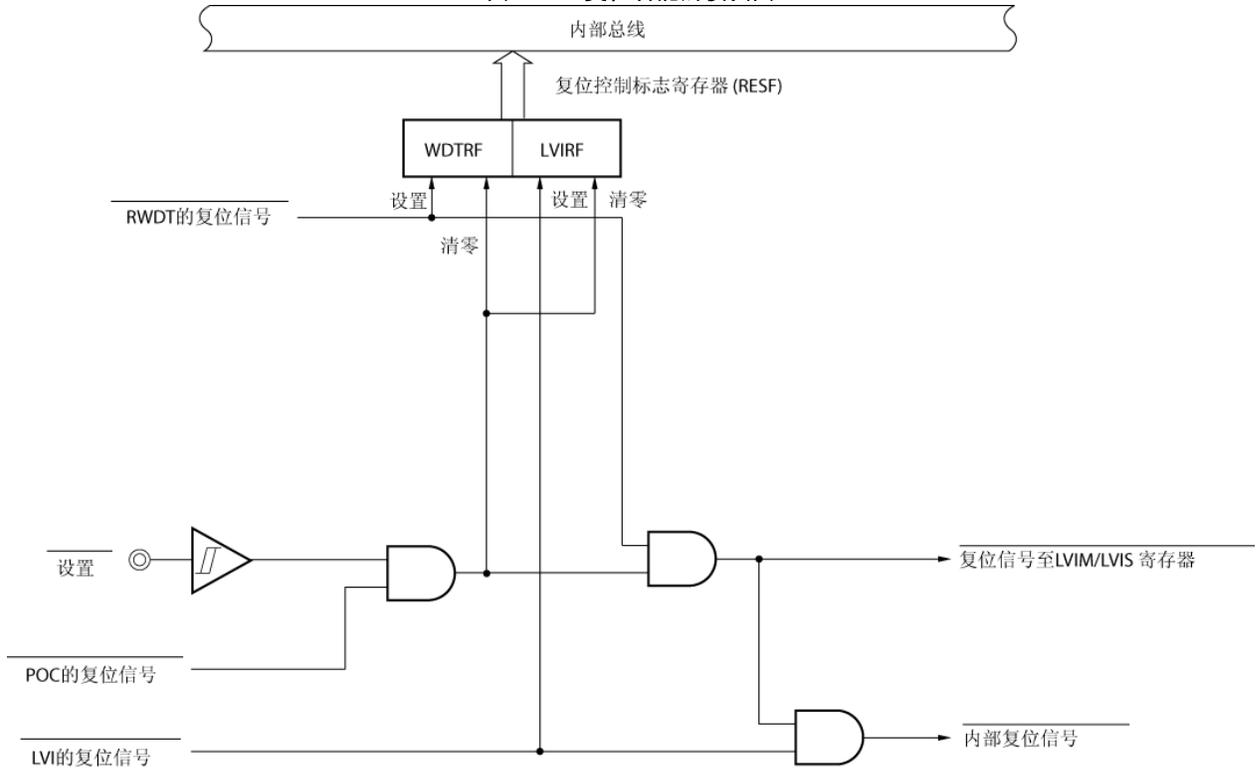
当有低电平输入到 $\overline{\text{RESET}}$ 引脚时，复位发生。接着当输入高电平时，复位释放，CPU 参考选项字节后开始执行程序。由看门狗定时器产生的复位在复位发生后自动释放，CPU 参考选项字节后开始执行程序（见图 10-2 至 10-4）。由 POC 和 LVI 电源检测电路产生的复位，在复位发生后且 $V_{DD} > V_{POC}$ （或 $V_{DD} > V_{LVI}$ ）时自动释放，CPU 参考选项字节后开始执行程序。（见第 11 章 上电复位电路 以及 第 12 章 低电压检测器）。

注意事项 1. 对于外部复位，输入到 $\overline{\text{RESET}}$ 引脚上的低电平至少要保持 2 μs 。

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。

<R> 3. 当 $\overline{\text{RESET}}$ 引脚作为只输入端口引脚（P34）使用时，如果有低电平输入到 $\overline{\text{RESET}}$ 引脚中，且 POC 电路、LVI 电路或看门狗定时器释放复位后， $\mu\text{PD78F9500}$ ，78F9501，78F9502 被复位。复位状态将保持到有高电平输入到 $\overline{\text{RESET}}$ 引脚时为止。

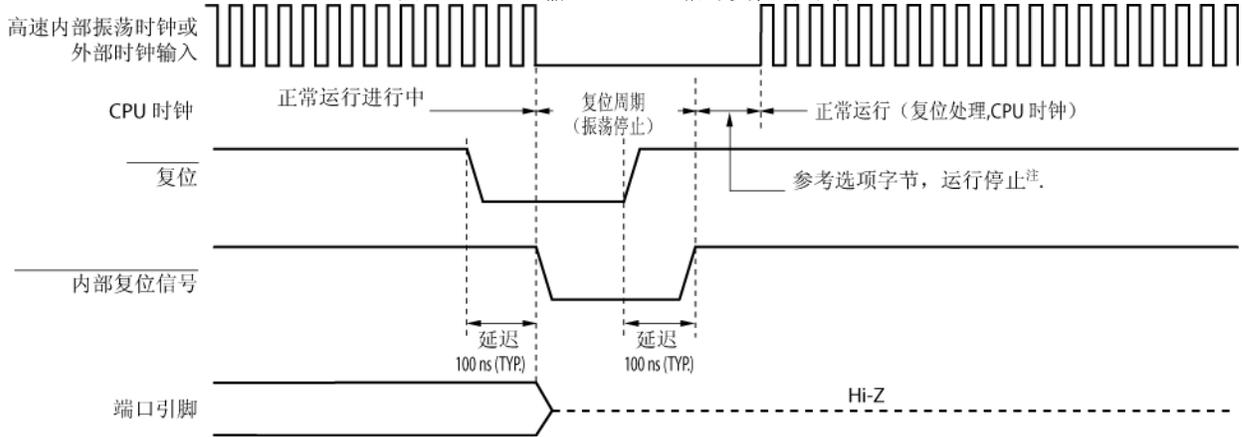
图 10-1. 复位功能的模块图



注意事项 LVI 不能被 LVI 电路的内部复位信号复位。

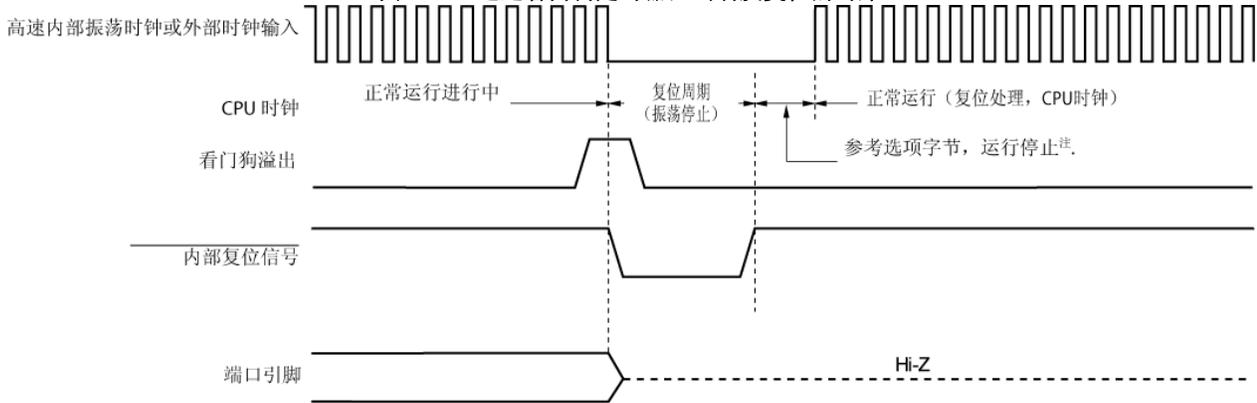
- 备注**
1. LVIM: 低压检测寄存器
 2. LVIS: 低压检测电平选择寄存器

图 10-2. 通过输入 RESET 信号复位的时序



注 工作停止时长为 277 μs (最小值), 544 μs (典型值) 和 1.075 ms (最大值)。

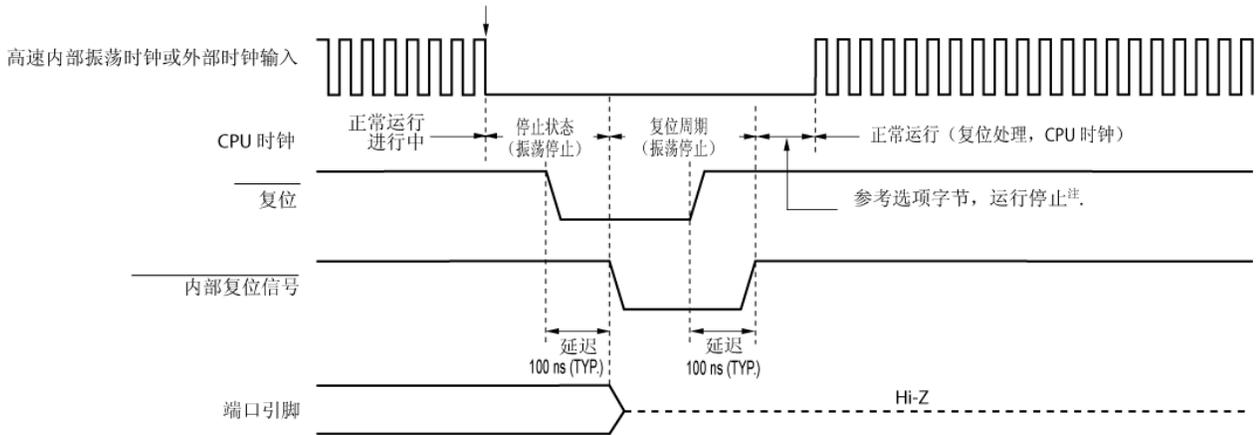
图 10-3. 通过看门狗定时器溢出引发复位的时序



注 工作停止时长为 277 μ s (最小值), 544 μ s (典型值) 和 1.075 ms (最大值)。

注意事项 看门狗定时器内部复位的情况下, 看门狗定时器也复位。

图 10-4. STOP 模式下, 通过输入 RESET 信号复位的时序



注 工作停止时长为 277 μ s (最小值), 544 μ s (典型值) 和 1.075 ms (最大值)。

注意事项 上电复位电路和低电压检测器的复位时序请参见第 11 章 上电复位电路和第 12 章 低电压检测器

表 10-1. 复位识别后的硬件状态

硬件		复位后的状态
程序计数器 (PC) ^{注 1}		设置复位矢量表的内容 (0000H 和 0001H)
堆栈指针 (SP)		不确定
程序状态字 (PSW)		02H
RAM	数据内存	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
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
	使能寄存器 (WDTE)	9AH
复位功能	复位控制标志寄存器 (RESF)	00H ^{注 3}
低电压检测器	低电压检测寄存器 (LVIM)	00H ^{注 3}
	低电压检测电平选择寄存器 (LVIS)	00H ^{注 3}
中断	中断请求标志寄存器 (IF0)	00H
	中断屏蔽标志寄存器 (MK0)	FFH
	外部中断模式寄存器 (INTM0)	00H
Flash 存储器	Flash 保护指令寄存器 (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 编程模式控制寄存器 (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 1. 其它硬件单元的状态将保持不变。复位信号产生且振荡器振荡稳定时，除了 PC 外，其它寄存器都已被定义。
2. 待机模式下，复位后的状态将被保持。

注 3. 数值的变化由复位源决定，如下所示。

复位源		RESET 输入	由 POC 复位	由 WDT 复位	由 LVI 复位
RESF	WDTRF	清零 (0)	清零 (0)	置 1 (1)	保持
	LVIRF			保持	置 1 (1)
LVIM		清零 (00H)	清零 (00H)	清零 (00H)	保持
LVIS					

10.1 复位源寄存器

片上包含多种内部中断源，如 μ PD78F9500, 78F9501, 78F9502。复位控制标志寄存器 (RESF) 用于保存产生复位请求的中断源的种类。

RESF 可被 8 位存储器操作指令读取。

RESET 输入，上电复位 (POC) 电路产生复位信号，读取 RESF，都会将 RESF 清零。

图 10-5. 复位控制标志寄存器的格式 (RESF)

地址: FF54H 复位后: 00H^注 R

编号	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LVIRF

WDTRF	内部复位请求来自看门狗定时器 (WDT)
0	未产生内部复位请求，或 RESF 被清零
1	产生内部复位请求

LVIRF	内部复位请求来自低电压检测器 (LVI)
0	未产生内部复位请求，或 RESF 被清零
1	内部复位请求产生

注 复位后的数值因复位源的不同而不同。

注意事项 请不要用 1 位操作指令来读取数据。

当产生复位请求时 RESF 的状态如表 10-2 所示。

表 10-2. 产生复位请求时 RESF 的状态

复位源	RESET 输入	由 POC 复位	由 WDT 复位	由 LVI 复位
WDTRF	清零 (0)	清零 (0)	置 1 (1)	保持
LVIRF			保持	置 1 (1)

11.1 上电清零电路的功能

上电清零电路（POC）具有如下功能。

- 上电时产生内部复位信号
- 比较供电电压（ V_{DD} ）与检测电压（ $V_{POC} = 2.1\text{ V} \pm 0.1\text{ V}$ ），当 $V_{DD} < V_{POC}$ 时产生内部复位信号。
- 比较供电电压（ V_{DD} ）与检测电压（ $V_{POC} = 2.1\text{ V} \pm 0.1\text{ V}$ ），当 $V_{DD} \geq V_{POC}$ 时释放内部复位信号。

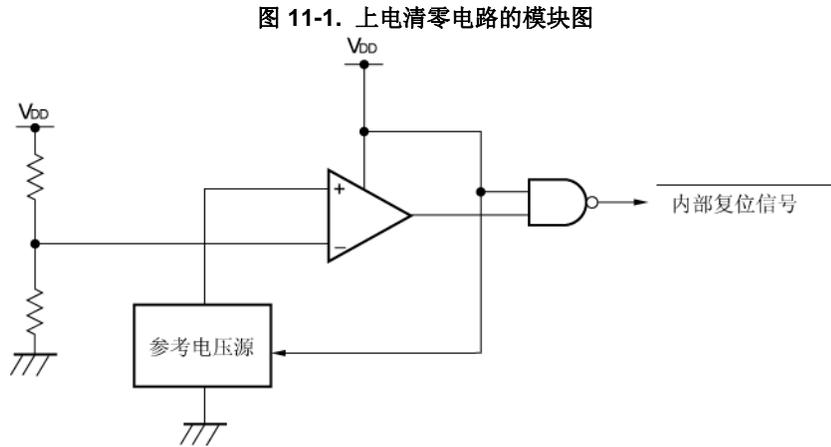
注意事项

1. 如果在 POC 电路产生内部复位信号，则复位控制标志寄存器（RESF）清零。
2. POC 电路的检测电压（ V_{POC} ）位于 $2.1\text{ V} \pm 0.1\text{ V}$ 之间，因此应采用 2.2 到 5.5 V 的电压供电。

备注 该产品含有多种可产生内部复位信号硬件功能。说明复位源的标志位于复位控制标志寄存器（RESF）中。内部复位信号产生时，该标识用以区分复位源是看门狗定时器（WDT）或低压检测电路还是其它。当复位信号由 WDT 或 LVI 产生时，RESF 不被清零而是被置 1。
RESF 详见第 10 章 复位功能。

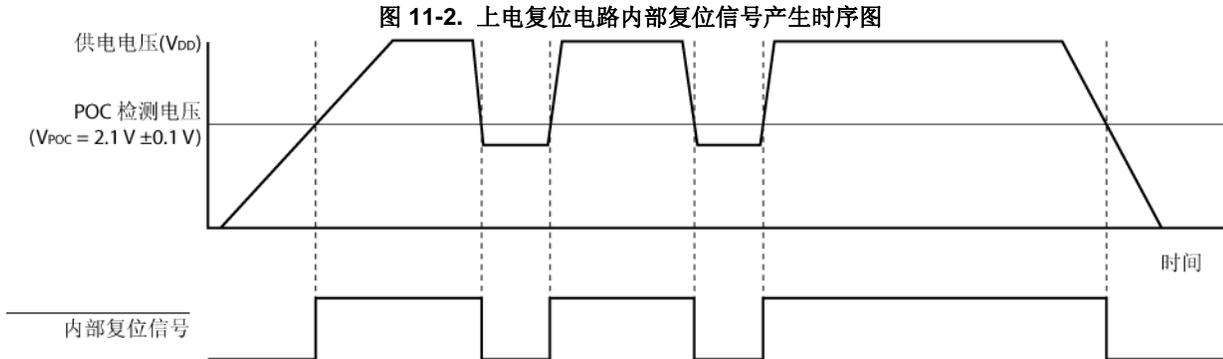
11.2 上电清零电路的构造

上电清零电路的模块图如图 11-1 所示。



11.3 上电清零电路的工作原理

比较上电清零电路的供电电压 (V_{DD}) 与检测电压 ($V_{POC} = 2.1\text{ V} \pm 0.1\text{ V}$)，当 $V_{DD} < V_{POC}$ 时产生内部复位信号。当 $V_{DD} \geq V_{POC}$ 时释放内部复位信号。



11.4 使用上电复位电路的注意事项

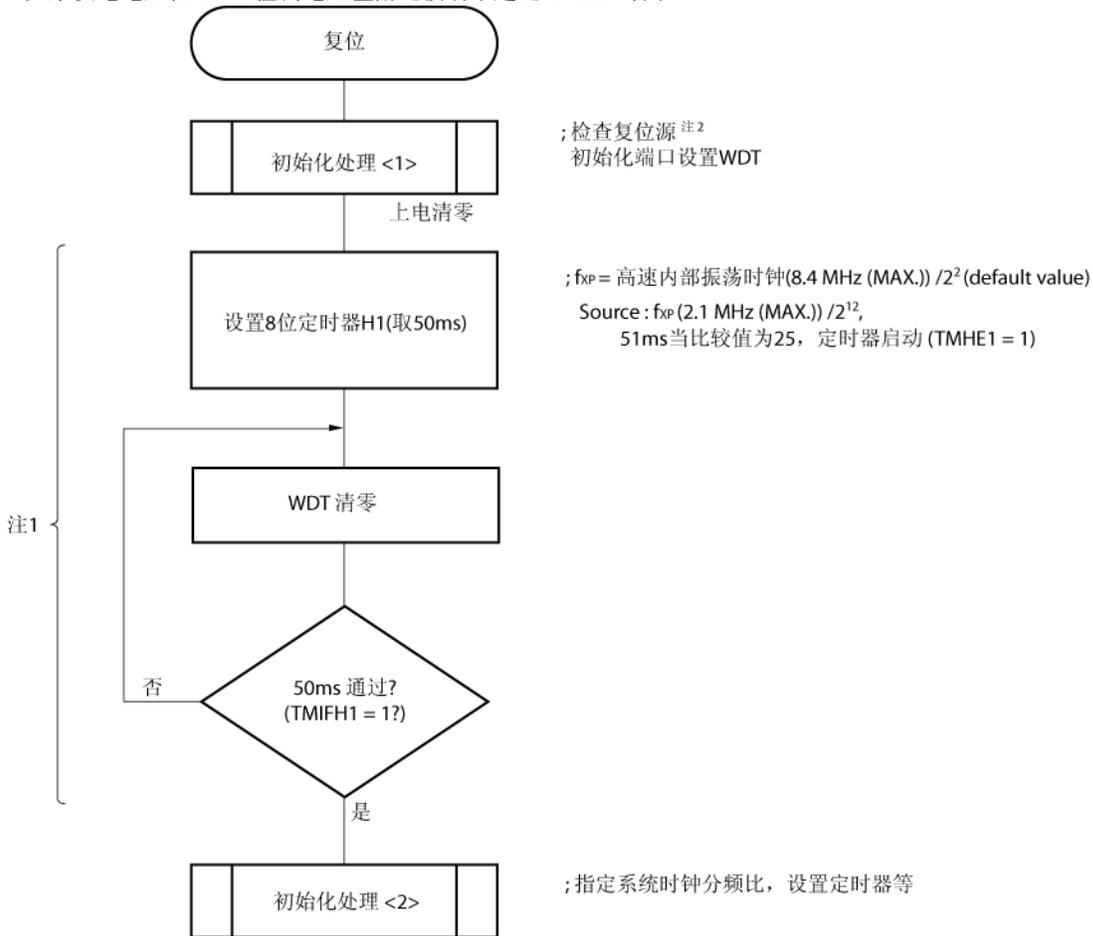
当系统供电电压 (V_{DD}) 在 POC 检测电压值 (V_{POC}) 附近波动一定时长时, 系统可能会反复复位和释放。此时, 可通过以下措施任意设置从复位释放到微控制器开始工作间的时长。

<措施>

释放复位信号后, 通过软件计数器定时等待, 等各系统的供电电压稳定后, 进行端口的初始化。

图 11-3. 复位释放后的软件处理示例 (1/2)

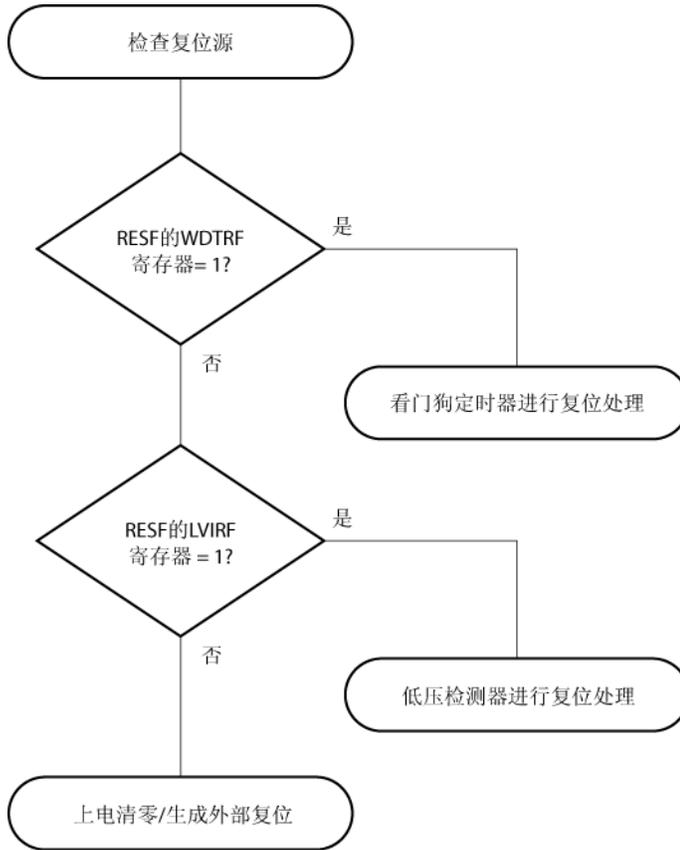
- 如果供电电压在 POC 检测电压值附近波动不超过 50 ms 时间。



- 注
1. 如果复位在此期间产生, 则初始化步骤<2>将不执行。
 2. 流程图见下一页。

图 11-3. 复位释放后的软件处理示例 (2/2)

• 查找复位源



12.1 低电压检测器的功能

低电压检测器 (LVI) 具有如下功能。

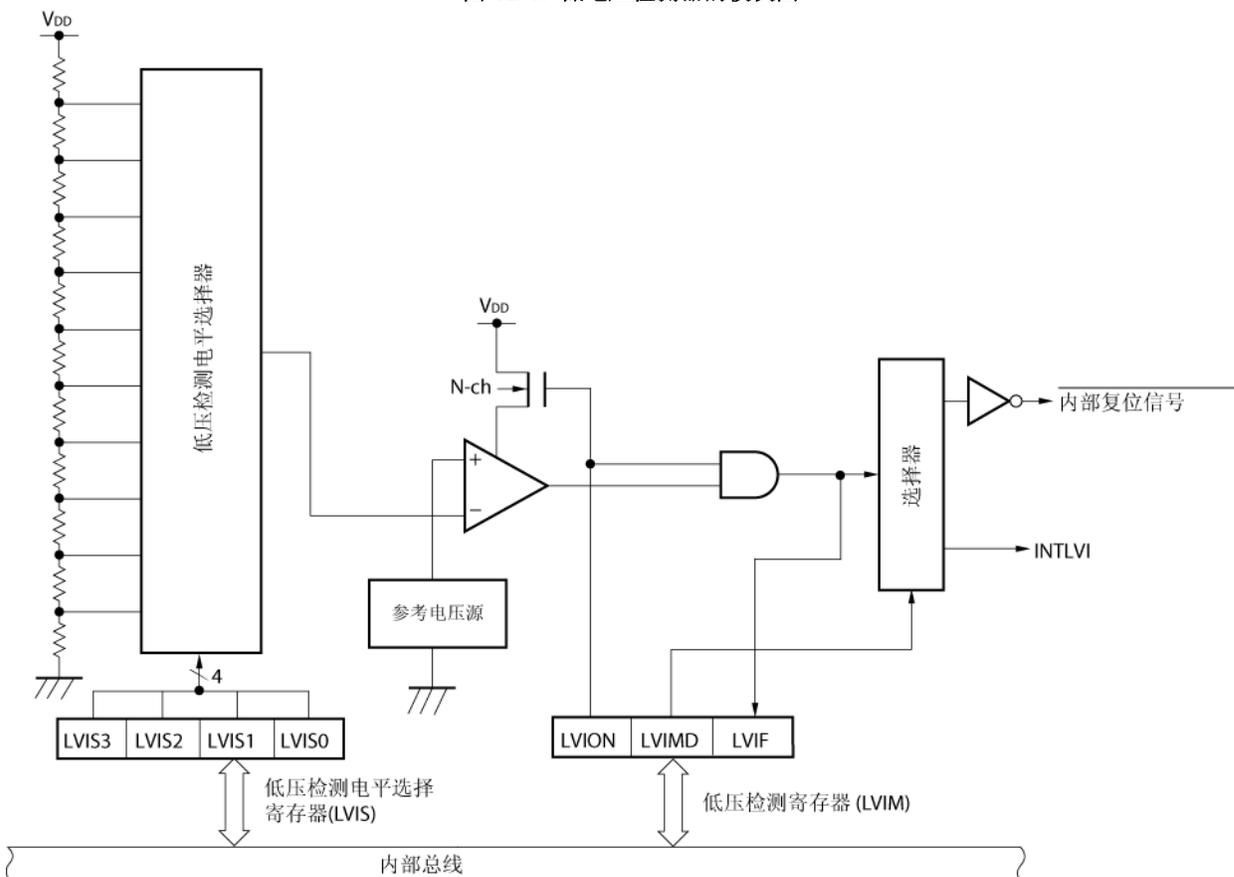
- 比较供电电压 (V_{DD}) 和检测电压 (V_{LVI})，当 $V_{DD} < V_{LVI}$ 时产生内部中断信号或内部复位信号。
- 可通过软件选择供电电压的检测电平 (共十种)。
- 可通过软件选择中断或复位功能。
- 在 STOP 模式下可工作。

低电压检测器可产生复位信号。当复位发生时，复位控制标识寄存器 (RESF) 的第 0 位 (LVIRF) 将被置 1。关于 RESF 详见第 10 章 复位功能。

12.2 低电压检测器的构造

低电压检测器的模块图如图 12-1 所示。

图 12-1. 低电压检测器的模块图



12.3 低电压检测器控制寄存器

低电压检测器受如下寄存器控制。

- 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)

该寄存器可设置低电压检测以及工作模式。

可通过 1 位或 8 位存储器操作指令操作该寄存器。

复位信号将把寄存器置为 00H^{注1}。

图 12-2. 低电压检测寄存器的格式 (LVIM)

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}	低电压检测器工作使能
0	工作使能关闭
1	工作使能

LVIMD	低电压检测器工作模式选择
0	当供电电压 (V _{DD}) < 检测电压 (V _{LVI}) 时产生中断信号
1	供电电压 (V _{DD}) < 检测电压 (V _{LVI}) 时产生内部复位信号

LVIF ^{注4}	低压检测标志
0	供电电压 (V _{DD}) ≥ 检测电压 (V _{LVI}) 或工作使能关闭时
1	供电电压 (V _{DD}) < 检测电压 (V _{LVI}) 时

- 注
1. LVI 引发的复位, LVIM 值不会被初始化。
 2. 第 0 位为只读位。
 3. 当 LVION 被置 1 时, LVI 电路中的比较器开始工作。LVION 置 1 时, 可用软件启动等待, 直到 LVIF 确定电压。该等待时间至少为 0.2 ms。
 4. 当 LVION = 1 且 LVIMD = 0 时, LVIF 的值将作为中断信号 INTLVI 输出。

注意事项 1. 请按如下任意一种操作方式来关闭 LVI。

- 采用 8 位存储器操作指令时: 向 LVIM 写入 00H。
- 采用 1 位存储器操作指令: 将 LVION 清零。

2. 请确保第 2 至 6 位被清零。

(2) 低电压检测电平选择寄存器 (LVIS)

该寄存器可用于选择低电压检测电平。

可通过 8 位存储器操作指令设置该寄存器。

复位信号将把该寄存器置为 00H[#]。

图 12-3. 低电压检测电平选择寄存器的格式 (LVIS)

地址: FF51H, 复位后: 00H[#] R/W

编号	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)
其它				禁止设置

注 由 LVI 产生的复位, LVIS 的值将不被初始化。

注意事项 1. 第 4 至 7 位必须设置为 0。

2. 如果在 LVI 工作期间向 LVIS 写入上表以外的其它值, 在写入时会变得不确定。因此在写 LVIS 之前, 请确保先停止 LVI (将 LVIM 寄存器第 7 位 (LVION) 清零)。

12.4 低电压检测器的工作

低电压检测器具有以下两种工作模式。

- 复位模式
比较供电电压 (V_{DD}) 与检测电压 (V_{LVI})，当 $V_{DD} < V_{LVI}$ 时，产生内部复位信号，而当 $V_{DD} \geq V_{LVI}$ 时，释放内部复位。
- 中断模式
比较供电电压 (V_{DD}) 与检测电压 (V_{LVI})，当 $V_{DD} < V_{LVI}$ 时，产生中断信号 (INTLVI)。

工作设置如下。

(1) 采用复位模式时

- 开始工作时
 - <1> 屏蔽 LVI 中断 ($LVIMK = 1$)。
 - <2> 通过设置低电压检测电平选择寄存器 (LVIS) 的第 3 至 0 位，设置检测电压。
 - <3> 将 LVIM 的第 7 位 (LVION) 置 1 (使能 LVI 工作)。
 - <4> 使用软件开启等待 (至少 0.2 ms)
 - <5> 等待“供电电压 (V_{DD}) \geq 检测电压 (V_{LVI})”，即判读 LVIM 的第 0 位 (LVIF)。
 - <6> 将 LVIM 的第 1 位 (LVIMD) 置 1 (当供电电压 (V_{DD}) $<$ 检测电压 (V_{LVI}) 时产生内部复位信号)。

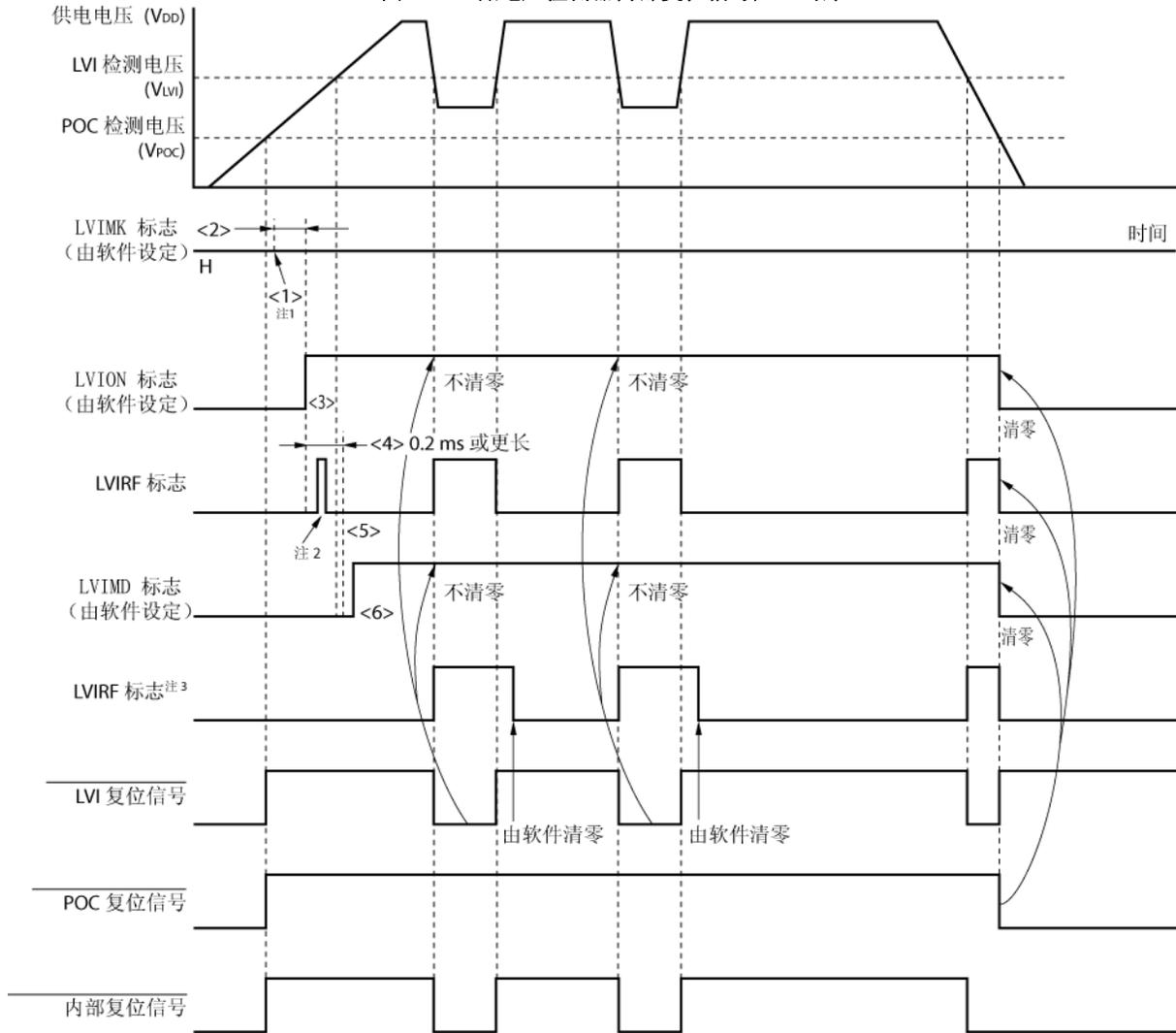
图 12-4 为产生低电压检测器内部复位信号的时序图。图中标号<1>至<6>分别对应于以上<1>至<6>步。

注意事项

1. 步骤<1>必须保持。因为如果当 $LVIMK = 0$ 时，步骤<3>完成后可能会立即产生中断。
2. $LVIMD$ 被设置为 1 后，供电电压 (V_{DD}) \geq 检测电压 (V_{LVI}) 时不会产生内部复位信号。

- 停止工作时，需执行以下任何一种操作。
 - 使用 8 位存储器操作指令时：向 LVIM 写入 00H。
 - 使用 1 位存储器操作指令时：将 LVIMD 和 LVION 清零。

图 12-4. 低电压检测器内部复位信号产生时序



- 注
1. 复位信号产生后，LVIMK 标志被置 1。
 2. LVIF 标志可被置 1。
 3. LVIRF 为复位控制标志寄存器 (RESF) 的第 0 位。关于 RESF 详见第 10 章 复位功能。

备注 图 12-4 中<1>至<6>分别对应于“12.4 (1) 采用复位模式”下开始工作时的第<1>至<6>步。

(2) 采用中断模式时

• 开始工作时

- <1> 屏蔽 LVI 中断 (LVIMK = 1)。
- <2> 通过设置低电压检测电平选择寄存器 (LVIS) 的第 3 至 0 位, 设置检测电压。
- <3> 将 LVIM 的第 7 位 (LVION) 置 1 (使能 LVI 工作)。
- <4> 使用软件开启等待 (至少 0.2 ms)。
- <5> 等待“供电电压 (V_{DD}) \geq 检测电压 (V_{LVI})”, 即判读 LVIM 的第 0 位 (LVIF)。
- <6> 将 LVI 的中断请求标志 (LVIIF) 清零。
- <7> 释放 LVI 的中断屏蔽标志 (LVIMK)。
- <8> 执行 EI 指令 (当使用矢量中断时)。

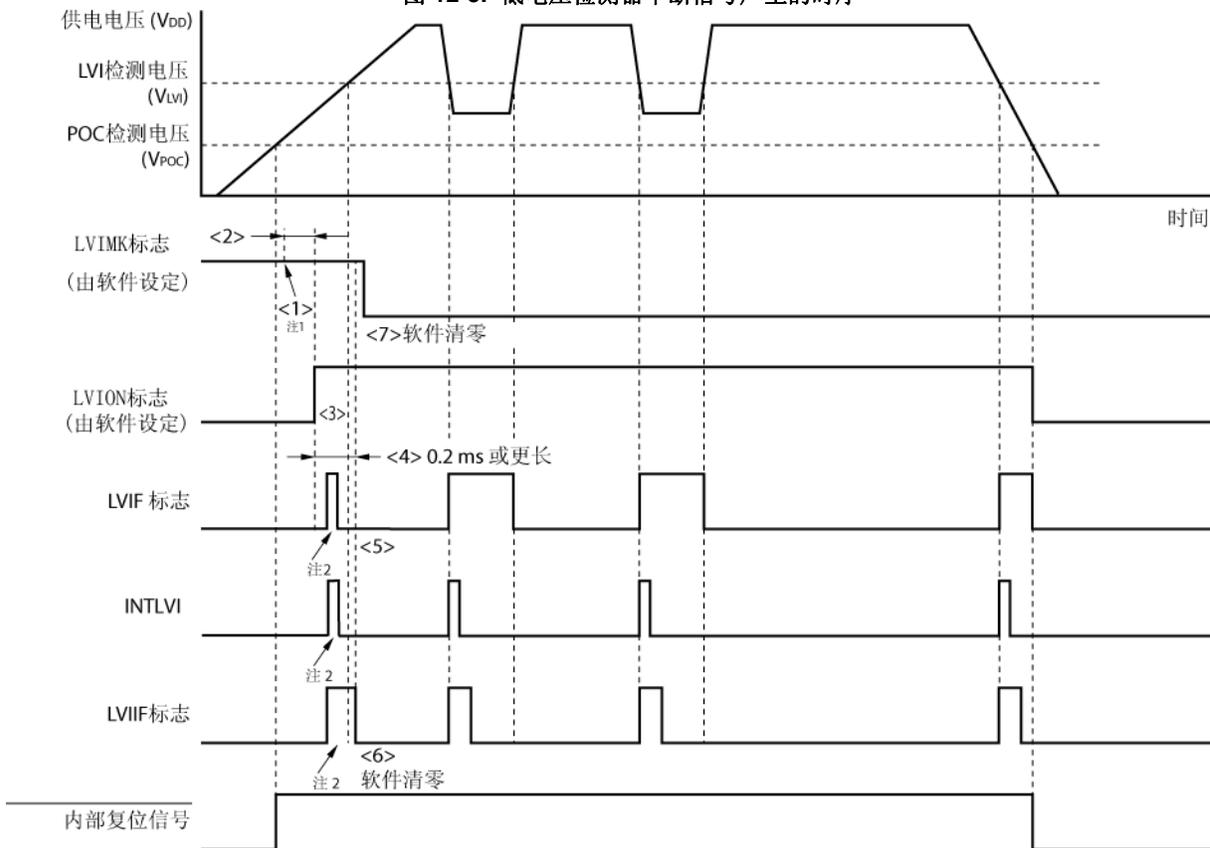
图 12-5 为产生低电压检测器中断信号时序图。图中标号<1>至<6>分别对应于以上<1>至<7>步。

• 停止工作时

需执行以下任意一种操作。

- 使用 8 位存储器操作指令时: 向 LVIM 写入 00H。
- 使用 1 位存储器操作指令时: 将 LVION 清零。

图 12-5. 低电压检测器中断信号产生的时序



- 注 1. 复位信号产生后，LVIMK 标志置 1。
 2. 可能会产生中断请求信号（INTLVI），LVIF 和 LVIIF 标志可能会被置 1。

备注 图 12-5 中<1>至<7>分别对应于“12.4 (2) 采用中断模式”下开始工作时的第<1>至<7>步。

12.5 使用低电压检测器的注意事项

当系统供电电压 (V_{DD}) 在 LVI 检测电压值 (V_{Lvi}) 附近波动一定时间时, 需根据低电压检测器模式的不同来采取不同措施。

<1> 采用复位模式时

系统可能会反复复位和释放。

此时, 可通过下述操作 (1) 任意设置从复位释放到微控制器开始工作期间的等待时长。

<2> 采用中断模式时

中断请求可能会频繁产生。执行下述操作 (2) 的第 (b) 步。

此时, 选用如下操作。

<操作>

(1) 采用复位模式时

复位信号释放后, 通过软件定时器, 定时等待各系统供电电压结束波动达到稳定, 然后初始化端口 (参见图 12-6)。

(2) 采用中断模式时

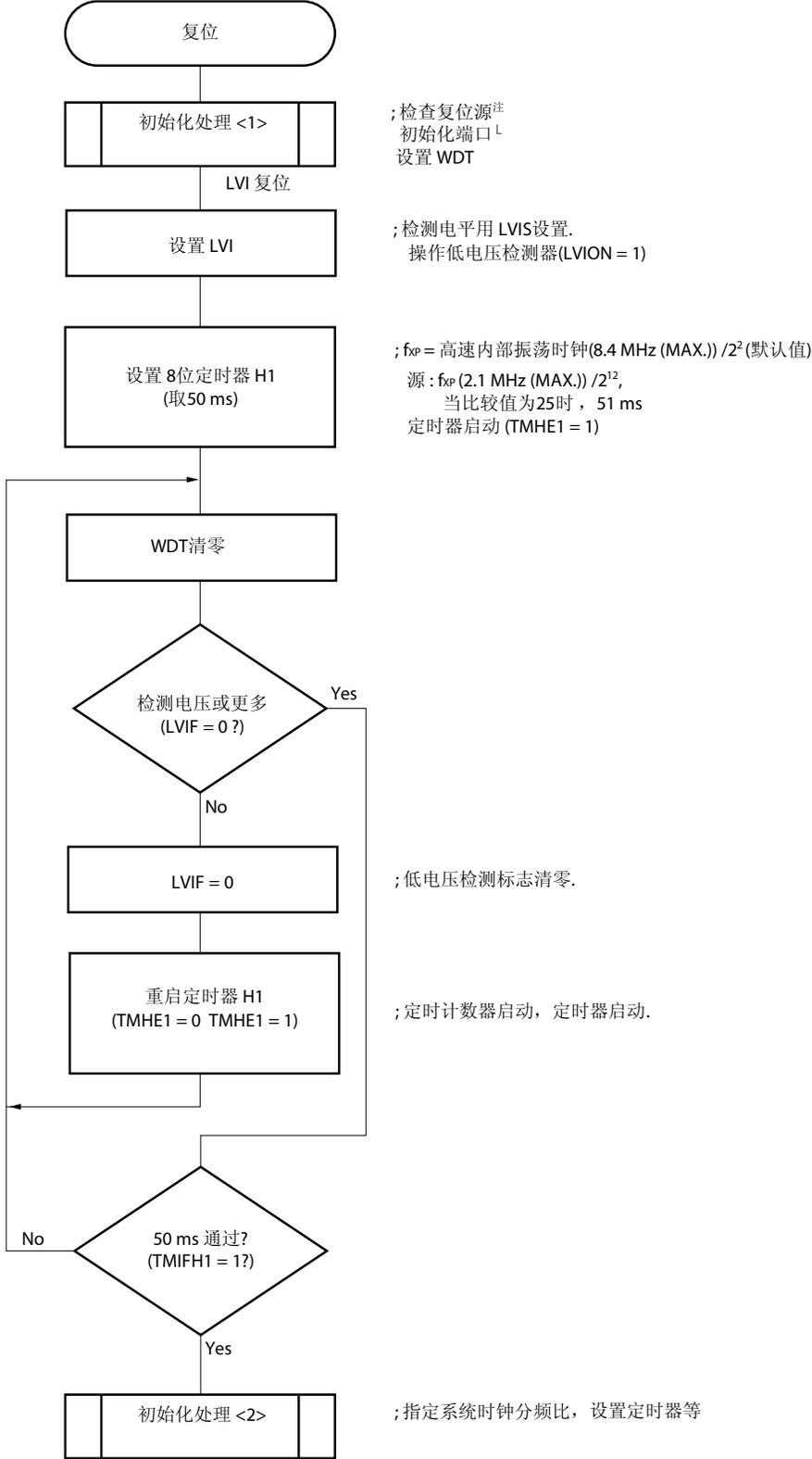
(a) 执行低压检测处理^注。在 LVI 中断服务程序中检测低电压检测寄存器 (LVIIF) 第 0 位 (LVIF) 是否为“供电电压 (V_{DD}) \geq 检测电压 (V_{Lvi})”, 并将中断请求标志寄存器 (IF0) 第 1 位 (LVIIF) 清零。

(b) 当系统供电电压在 LVI 检测电压值附近波动时间过长时, 等待波动稳定, 检测 LVIF 标志是否为“供电电压 (V_{DD}) \geq 检测电压 (V_{Lvi})”, 并将 LVIIF 标志清零。

注 低压检测时, CPU 时钟速率将被切换为低速。

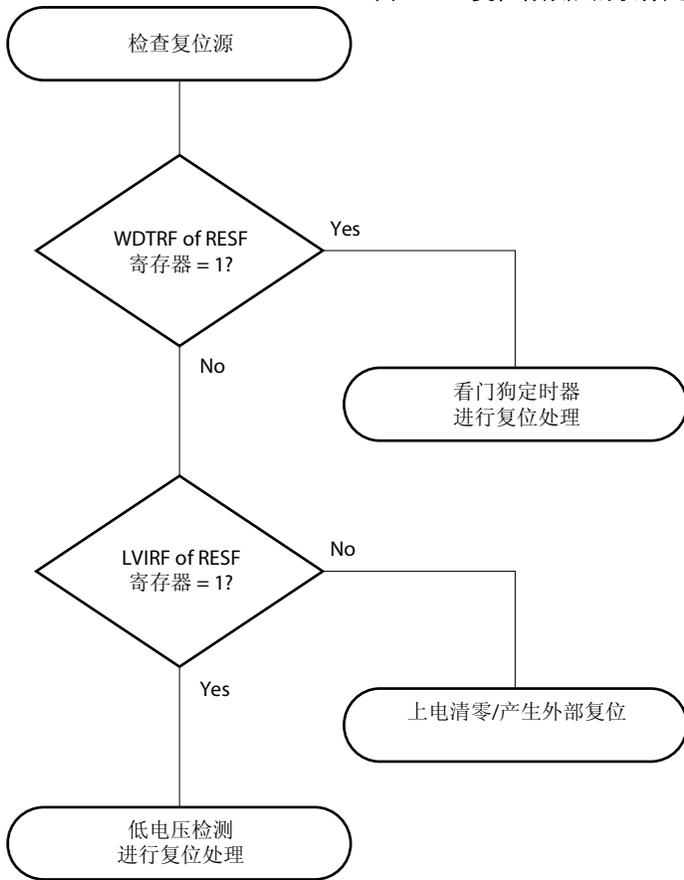
图 12-6. 复位释放后的软件处理处理示例 (1/2)

- 如果供电电压在 LVI 检测电压附近波动不超过 50ms 时间。



注 流程图参见下一页。

图 12-6. 复位释放后的软件处理处理示例 (2/2)



- 查找复位源

13.1 选项字节的功能

μ PD78F9500, 78F9501, 78F9502 的选项字节位于 FLASH 存储器的 0080H 地址处。上电或复位后，将自动引用选项字节，执行具体功能的相关设置。使用该产品时，请务必通过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
- 外部时钟输入

(2) 低速内部振荡时钟的振荡

- 不能被停止
- 可被软件停止

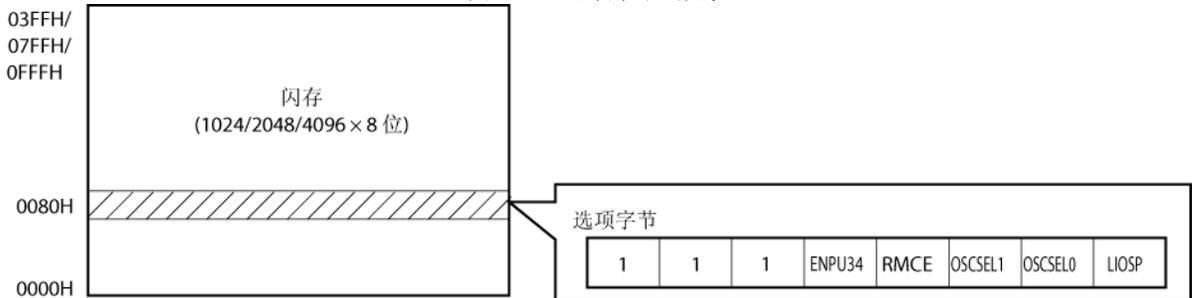
(3) $\overline{\text{RESET}}$ 引脚的控制

- 用作 $\overline{\text{RESET}}$ 引脚
- $\overline{\text{RESET}}$ 引脚被用作单向输入端口 (P34) 时 (参见 13.3 当 $\overline{\text{RESET}}$ 引脚被用作单向输入端口 (P34) 时的注意事项)。
- 选择 $\overline{\text{RESET}}$ 引脚上的自带上拉电阻，或者将 $\overline{\text{RESET}}$ 引脚设置为开路。

(4) 电源上电或复位释放时的振荡稳定建立时长

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

图 13-1. 选项字节的定位



13.2 选项字节的格式

选项字节的格式如下所示。

图 13-2. 选项字节的格式 (1/2)

地址: 0080H

7	6	5	4	3	2	1	0
1	1	1	ENPU34	RMCE	OSCSEL1	OSCSEL0	LIOCP

ENPU34	RESET 引脚片上上拉电阻的选择
1	选择 RESET 引脚片上上拉电阻。
0	不选择 RESET 引脚片上上拉电阻。

备注 当用作 RESET 引脚时, 可通过将 ENPU34 置 1 实现引脚开路。

RMCE	RESET 引脚的控制
1	RESET 引脚用作复位输入
0	RESET 引脚用作输入端口 (P34)

注意事项 由于复位释放后才能引用选项字节, 所以如果在引用前向 RESET 引脚输入低电平, 则复位状态将无法释放。

当用作单向输入端口 (P34) 时, 可通过 PU3 上的 PU34 设置片上上拉电阻。

OSCSEL1	OSCSEL0	系统时钟源的选择
0	0	禁止设置
0	1	外部时钟输入
1	×	高速内部振荡时钟

注意事项 因为 EXCLK 引脚还可作为 P23 引脚使用, 因此根据系统时钟源的不同, EXCLK 引脚的用途也将不同。

(1) 选择外部时钟输入

由于 P23 引脚被用作外部时钟输入端口, 因此不能再被用作 I/O 端口。

(2) 选择高速内部振荡时钟

P23 引脚可被用作 I/O 端口。

备注 ×: 1 或 0

图 13-2. 选项字节的格式 (2/2)

LIOCP	低速内部振荡器
1	不能被停止 (即使 LSRSTOP 位置 1, 振荡也不会停止)
0	可被软件停止 (LSRSTOP 位置 1 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”，则看门狗定时器 (WDT) 的计数时钟只能采用低速内部振荡时钟。
 2. 如果选择“低速内部振荡器可被软件停止”，那么无论低速内部振荡模式寄存器 (LSRCM) 是否被置 1，只要进入 HALT/STOP 模式，WDT 计数时钟都将被关闭。同样，采用低速内部振荡器以外的时钟作为 WDT 的计数时钟，只要进入 HALT/STOP 模式后也将被关闭。
低速内部振荡器工作时 (LSRSTOP = 0)，即使在 STOP 模式下，时钟也可提供给 8 位定时器 H1。

- 备注**
1. (): $f_x = 10 \text{ MHz}$
 2. 关于振荡器的振荡稳定建立时长，请参阅所选振荡器的特性指标。
 3. 软件编程设置选项字节的示例如下所示。

```

OPB CSEG AT 0080H
DB 10010001B      ;数据设置到选项字节
                  ;低速内部振荡器不能被停止
                  ;RESET引脚用作单向输入端口 (P34)
                  ;最小振荡稳定时长 ( $2^{10}/f_x$ )

```

4. 关于引用选项字节时的操作时序，见第 10 章复位功能。

13.3 当RESET引脚用作单向输入端口 (P34) 时的注意事项

使用专用 FLASH 存储器编程器对设备进行二次擦写时，请注意如下几点。其中被写设备已通过选项字节设置为“RESET 引脚作为单向输入端口 (P34) 使用”。

向目标系统供电前，请先连接并开启专用 FLASH 存储器编程器。

如果在连接编程器前先对目标系统上电，则将无法切换到 FLASH 存储器编程模式。

第十四章 Flash存储器

14.1 特性

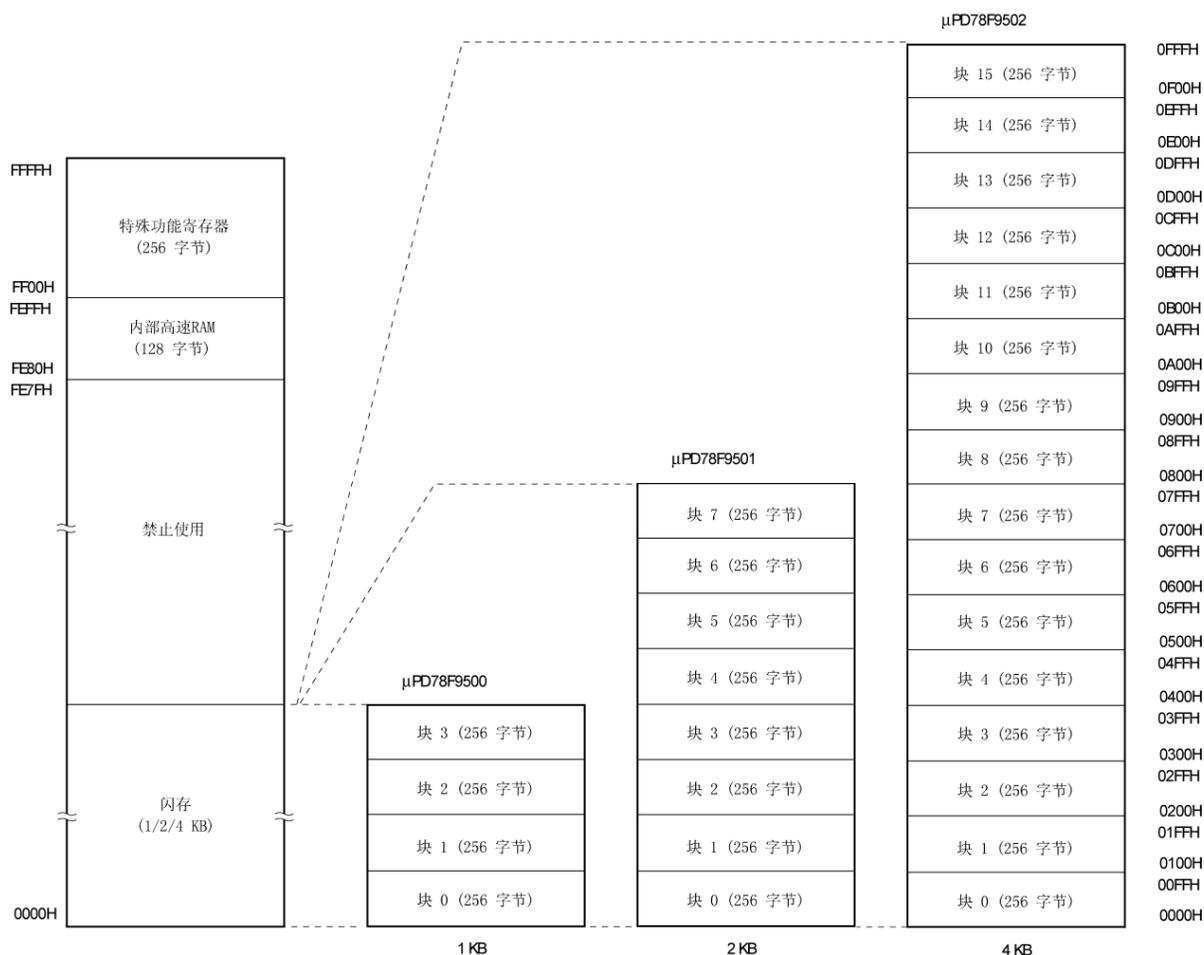
μPD78F9500、78F9501、78F9502 的内部 Flash 存储器有以下特性。

- 擦除/写入而不必准备单独的专用电源
- 容量： 1/2/4 KB
 - 擦除单元： 1 块（256 字节）
 - 写入单元： 1 块（在板上/离板编程时），1 字节（自编程时）
- 重写方法
 - 通过与专用 Flash 存储器编程器通信重写（板上/离板编程）
 - 由用户程序重写 Flash 存储器（自编程）
- 在板上/离板编程时通过安全功能支持 Flash 存储器重写
- 通过保护字节，在自编程时支持以块为单元的安全功能

14.2 内存配置

1/2/4 KB 内部 Flash 存储器区分成 4/8/16 块且可在块单元中编程/擦除。通过专用 Flash 存储器编程器可一次性擦除所有块。

图 14-1. Flash 存储器映射



14.3 功能概述

μPD78F9500、78F9501、78F9502 的内部 Flash 存储器可以使用专用 Flash 存储器编程器写入，而不管 μPD78F9500、78F9501、78F9502 是否已经安装在目标系统上（板上/离板编程）。

提供以用户程序重写程序的功能（自编程），当假定在目标系统生产/出货之后要改变程序，则重写程序对应用程序来说极为理想。

有关 Flash 存储器写入控制功能的详情，请参看表 14-1。

此外，还支持禁止未经授权将用户程序写入内部 Flash 存储器的安全功能。

有关安全功能的更多内容，请参看 14.7.3 安全设置。

表 14-1. 重写方法

重写方法	功能概述	工作模式
板上编程	可以在装置安装在目标系统之后使用专用 Flash 存储器编程器重写 Flash 存储器。	Flash 存储器编程模式
离板编程	可以在装置安装在目标系统之前使用专用 Flash 存储器编程器和专用程序适配器板重写 Flash 存储器（FA 系列）。	
自编程	可以执行通过板上/离板编程预先写入 Flash 存储器的用户程序来重写 Flash 存储器。	自编程模式

- 备注
1. FA 系列是 Naito Densei Machida Mfg. 有限公司的产品。
 2. 有关 Flash 存储器写入控制功能的详细内容，请参看以下章节。
 - 14.7 板上和离板 Flash 存储器编程
 - 14.8 自编程的 Flash 存储器编程

14.4 通过Flash 存储器编程器写入

以下两类专用 Flash 存储器编程器可以用于将数据写入 μ PD78F9500、78F9501、78F9502 的内部 Flash 存储器。

- FlashPro4 (PG-FP4、FL-PR4)
- FlashPro5 (PG-FP5、FL-PR5)

数据可以使用专用 Flash 存储器编程器板上或离板写入 Flash 存储器。

(1) 板上编程

μ PD78F9500、78F9501、78F9502 安装在目标系统之后可以重写 Flash 存储器的内容。连接专用 Flash 存储器编程器的连接器必须安装在目标系统上。

(2) 离板编程

μ PD78F9500、78F9501、78F9502 安装在目标系统上之前，可以用专用程序适配器将数据写入 Flash 存储器（FA 系列）。

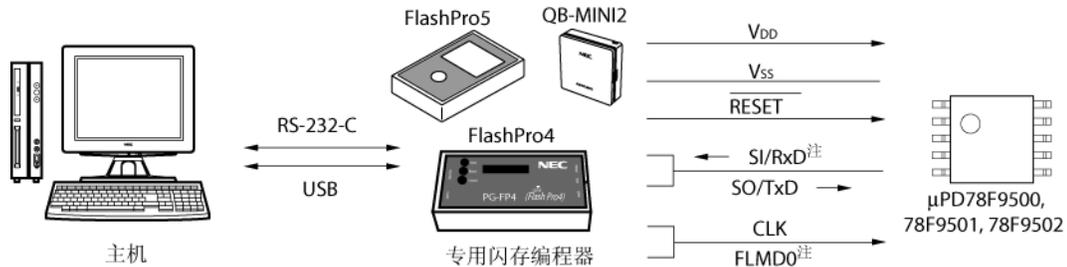
- <R> 备注 FL-PR4、FL-PR5 和 FA 系列是 Naito Densei Machida Mfg. 有限公司的产品。

14.5 编程环境

将程序写入 Flash 存储器的编程环境如下所示。

<R>

图 14-2. 将程序写入 Flash 存储器的环境 (FlashPro4/FlashPro5/QB-MINI2)



<R>

注 当使用 FlashPro5 和 QB-MINI2 时，无需连接该信号线。

需要使用主机控制专用 Flash 存储器编程器。当使用 PG-FP4、FL-PR4、PG-FP5 或者 FL-PR5 时，仅可以在从主机下载程序之后使用专用 Flash 存储器编程器写入数据。

当在专用 Flash 存储器编程器与 μPD78F9500、78F9501、78F9502 之间进行连接时，UART 用于诸如写入和擦除的操作。要离板写入 Flash 存储器，则需要专用程序适配器 (FA 系列)。

从开发工具的下载网站下载最新编程器固件、GUI 和参数文件，网址 (<http://www.necel.com/micro/ods/eng/index.html>)。

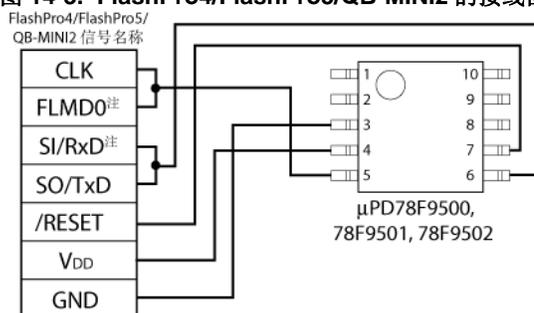
<R> 表 14-2. 在 μ PD78F9500、78F9501、78F9502 和 FlashPro4/FlashPro5/QB-MINI2 接线之间

FlashPro4/FlashPro5/QB-MINI2 连接引脚			μ PD78F9500、78F9501、78F9502 连接引脚	
引脚名称	I/O	引脚功能	引脚名称	引脚编号
CLK ^{注1}	输出	μ PD78F9500、78F9501、78F9502 的时钟	EXCLK/P23	5
FLMD0 ^{注1,2}	输出	板上模式信号		
SI/RxD ^{注1,2}	输入	接收信号	P22	6
SO/TxD ^{注1}	输出	接收信号/板上模式信号		
/RESET	输出	复位信号	RESET/P34	7
V _{DD}	-	V _{DD} 电压产生/电压监视器	V _{DD}	4
GND	-	接地	V _{SS}	3

注 1. 在 μ PD78F9500、78F9501、78F9502 中，CLK 和 FLMD0 信号连接 EXCLK 引脚；因此，需要直接连接这些信号。

<R> 2. 当使用 FlashPro5 和 QB-MINI2 时，不必连接该信号。

<R> 图 14-3. FlashPro4/FlashPro5/QB-MINI2 的接线图



注 当使用 FlashPro5 和 QB-MINI2 时，不必连接信号。

14.6 板上引脚的处理

要板上写入 Flash 存储器，则必须在目标系统上提供连接专用 Flash 存储器编程器的连接器。首先提供选择板上正常工作模式或者 Flash 存储器编程模式的功能。

当设置 Flash 存储器编程模式时，不用于编程 Flash 存储器的所有引脚处于复位后的相同状态。因此，如果复位后外部装置不能识别这一状态，则引脚必须进行如下处理。

自编程模式下的引脚状态与 HALT 模式下相同。

14.6.1 EXCLK 引脚

EXCLK 引脚用作 Flash 存储器编程的串行接口。因此，如果 EXCLK 引脚连接到外部装置，则发生信号冲突。要防止信号冲突，则需使连接与外部装置隔离。

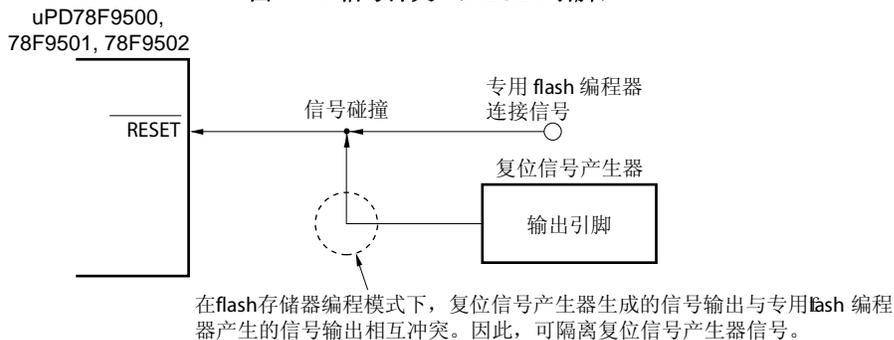
当将电容器连接到 EXCLK 引脚时，通信期间的波形改变。因此电容器的容量可能导致无法通信。当进行 Flash 存储器编程时，确保断开电容器的连接。

14.6.2 $\overline{\text{RESET}}$ 引脚

如果专用 Flash 存储器编程器的复位信号与已连接到板上复位信号发生器的 $\overline{\text{RESET}}$ 相连，就会发生信号冲突。要防止这一冲突，则需隔离复位信号发生器。

Flash 存储器编程模式时，如果从用户系统输入复位信号，则将不能正确编程 Flash 存储器。禁用输入专用 Flash 存储器编程器的复位信号之外的任何信号。

图 14-4. 信号冲突 ($\overline{\text{RESET}}$ 引脚)



14.6.3 端口引脚

当设置 Flash 存储器编程模式时，所有不用于 Flash 存储器编程的引脚进入与复位后的相同状态。如果连接到这一端口的的外部装置不能在复位后立即识别这个状态，则端口引脚必须经过一个电阻连接到 V_{DD} 或者 V_{SS} 。

自编程模式下的引脚状态与 HALT 模式下相同。

14.6.4 电源

将 V_{DD} 引脚连接到 Flash 存储器编程器的 V_{DD} 并将 V_{SS} 引脚连接到 Flash 存储器编程器的 V_{SS} 。

14.7 板上和离板Flash 存储器编程

14.7.1 Flash 存储器编程模式

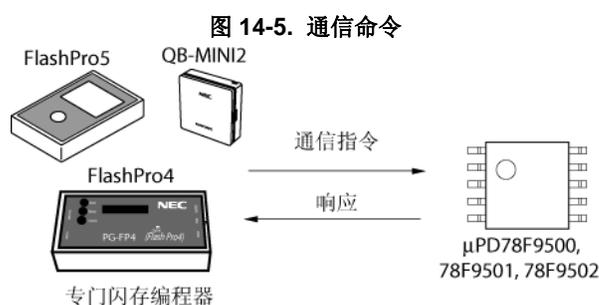
要使用专用 Flash 存储器编程器重写 Flash 存储器的内容，则需在 Flash 存储器编程模式下设置 μ PD78F9500、78F9501、78F9502。当 μ PD78F9500、78F9501、78F9502 连接到 Flash 存储器编程器且通信命令传送到微控制器，微控制器设置处于 Flash 存储器编程模式。

当板上写入 Flash 存储器时，使用跳线改变这一模式。

14.7.2 通信命令

专用 Flash 存储器编程器使用命令控制 μ PD78F9500、78F9501、78F9502。从 Flash 存储器编程器发送到 μ PD78F9500、78F9501、78F9502 的信号称作通信命令，且从 μ PD78F9500、78F9501、78F9502 发送到专用 Flash 存储器编程器的命令称作响应。

<R>



通信命令列于下表中。所有通信命令从编程器发出，且 μ PD78F9500、78F9501、78F9502 进行相应通信命令的处理。

表 14-3. 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	擦除整个内存的内容。
	块擦除命令	擦除指定块的内容。
写入	写入命令	写入指定地址范围并执行内容的验证检查。
检查和	检验和命令	读取指定地址范围的校验和与写入的数据进行比较。
空白检测	空白检测命令	确认整个内存的擦除状态。
安全	安全设置命令	禁用批擦除（芯片擦除）命令、块擦除命令和写入命令以防止第三方操作。

μ PD78F9500、78F9501、78F9502 返回对专用 Flash 存储器编程器的通信命令的响应。从 μ PD78F9500、78F9501、78F9502 发送的响应名称在下表列出。

表 14-4. 响应名称

命令名称	功能
ACK	对命令/数据.的应答
NAK	对无效命令/数据.的应答

14.7.3 安全设置

可以使用安全设置命令禁用以下所示操作。

- 禁用批擦除（芯片擦除）

禁止执行 Flash 存储器中整个块的块擦除和批擦除（芯片擦除）命令。一旦禁止执行批擦除（芯片擦除）命令，所有禁用设置再也不能取消。

注意事项 设置批擦除安全设置之后，不可以对设备进行擦除。此外，即使执行写入命令，不可以写入已写入 Flash 存储器之外的数据，因为已禁用擦除命令。

- 禁用块擦除

禁止执行 Flash 存储器中的块擦除命令。此禁用设置可以通过使用批擦除（芯片擦除）命令取消。

- 禁用写入

禁止对 Flash 存储器中的整个块执行写入和块擦除命令。此禁用设置可以通过使用批擦除（芯片擦除）命令取消。

备注 当下次设置编程模式时，安全设置有效。

当 Flash 存储器通过默认设置下，启用批擦除（芯片擦除）、块擦除和写入命令有效。以上安全设置仅可用于板上/离板编程。每个安全设置可以组合使用。

表 14-5 给出了当 μ PD78F9500、78F9501、78F9502 安全功能启用后，擦除和写入命令之间的关系。

表 14-5. 启用安全功能时命令之间的关系

安全 \ 命令	批擦除（芯片擦除）命令	块擦除命令	写入命令
当批擦除（芯片擦除）安全操作启用时	禁用	禁用	有效 ^注
当块擦除安全操作启用时	有效		有效
当写入安全操作 启用时			禁用

注 因为禁用擦除命令，不能写入已写入 Flash 存储器之外的数据。

表 14-6 给出了每个编程模式下安全设置和操作之间的关系。

表 14-6. 每个编程模式下安全设置和操作之间的关系

安全设置 \ 编程模式	板上/离板编程		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）	允许	有效 ^{#1}	不允许	无效 ^{#2}
块擦除				
写入				

- 注
1. 通过安全设置禁止执行任何指令执行。
 2. 不管安全设置如何，自编程命令都能够执行。

14.8 自编程模式下的Flash 存储器编程

μ PD78F9500、78F9501、78F9502 支持自编程功能，可以用于经用户程序重写 Flash 存储器，故它具有在线升级程序的功能。

注意事项 进行自编程之前，自编程处理程序必须先被包含在程序中。

- 备注**
1. 如果要使用自编程，请参看 14.8.4 之后提到的使用实例。
 2. 要将 μ PD78F9500、78F9501、78F9502 的内部 Flash 存储器用作外部 EEPROM 以便存储数据，请参看 **78K0S/Kx1+ EEPROM 模拟应用笔记 (U17379E)**。

14.8.1 自编程概述

要进行自编程，需将模式从用户程序的正常操作（正常模式）切换到自编程模式。在自编程模式期间，先设置特定寄存器，通过 HALT 指令执行擦写处理程序。处理过程完成之后自动释放 HALT 状态。

要切换到自编程模式，请执行特定寄存器的特定序列。有关详细内容，请参看 **14.8.4 正常模式切换到自编程模式的实例**。

备注 自编程写入的数据通过 MOV 指令引用。

表 14-7. 自编程模式

模式	用户程序执行	通过 HALT 指令执行 Flash 存储器的写入/擦除
正常模式	启用	-
自编程模式	启用 [#]	启用

注 自编程模式期间禁用可屏蔽中断服务。

图 14-6 给出了自编程的方块图。图 14-7 给出了自编程状态转换图。表 14-8 列出了控制自编程的命令。

图 14-7. 自编程状态转换图

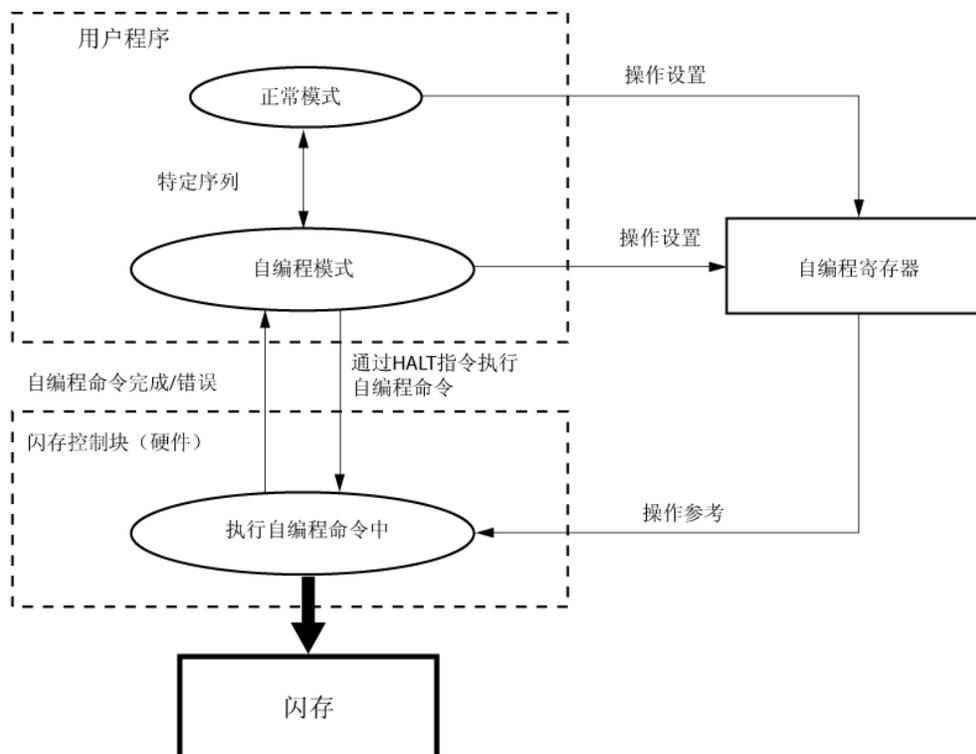


表 14-8. 自编程控制命令

命令名称	功能	从 HALT 指令执行到命令执行结束的时间
内部校验 1	此命令用于检查数据是否已经正确写入 Flash 存储器。其用于检查数据是否已写入整个块。	1 块内部校验（内部校验命令执行一次）： 6.8 ms
内部校验 2	此命令用于检查数据是否正确写入 Flash 存储器。其用于检查数据是否已写入同一块。	1 字节内部校验： 27 μs
块擦除 ^注	此命令用于擦除指定块。执行前先指定块序号。	8.5 ms
块空白检测	此命令用于检查是否已擦除指定块中的数据。指定块序号 接着执行此命令。	480 μs
字节写入	此命令用于将 1-字节数据写入 Flash 存储器的指定地址。指定写入地址和写入数据，接着执行此命令。	150 μs

<R> 注 设置重试的次数使其大于块擦除时间除以一次擦除的时间（8.5 ms）的值，与 Flash 存储器块擦除所需时间（最大值）一致。

备注 可以执行通过指定相同块中的地址进行命令内部校验 1，而如果数据写入同一块的两个或更多地址，则推荐内部校验 2。

14.8.2 自编程功能的注意事项

- 执行自编程命令时禁止执行其他指令。因此，看门狗定时计数器提前清零和重启，以防自编程期间看门狗定时器溢出。有关执行自编程所需时间的详细情况，请参看表 14-8。
- 自编程期间发生的中断可以在自编程模式结束后才能响应。为了避免此操作，在模式从正常模式切换到自编程模式之前通过特定顺序，禁用中断服务（通过将 MK0 设为 FFH 并执行 DI 指令）。
- 执行自编程命令时不使用 RAM。
- 如果电源电压降低或者输入复位信号时写入或擦除 Flash 存储器，则不保证写入/擦除顺利完成。
- 块擦除期间空白数据设置的值是 FFH。
- 自编程期前设置 CPU 时钟使其至少为 1 MHz。
- 执行特定顺序到自编程模式后，立即执行 NOP 和 HALT 指令，接着执行自编程。此时， $10 \mu\text{s}(\text{MAX.}) + 2 \text{ CPU 时钟}(\text{f}_{\text{CPU}})$ 之后自动释放 HALT 指令。
- 如果选择振荡器的时钟或外部时钟作为系统时钟，则在进行自编程模式的特定顺序设置之后立即执行 NOP 和 HALT 指令，释放 HALT 状态之后等待 $8 \mu\text{s}$ ，然后执行自编程。
- 使用 1-位内存操作指令检查 FPRERR。
- 自编程模式下的引脚状态与 HALT 模式下相同。
- 由于在自编程模式下禁用经由板上/离板编程的安全功能设置，因此不管安全功能如何设置，均可以执行自编程命令。自编程期间为了禁用写入或者擦除过程，请设置保护字节。
- 执行自编程命令之前，确保 Flash 存储器地址指针 H (FLAPH) 的位 4 到 7 和 Flash 存储器地址指针 H 比较寄存器 (FLAPHC) 清零。如果当执行自编程命令时这些位值为 1，则可能装置不能正常运行。
- 设置自编程模式和正常工作模式之前，将 FLCMD 寄存器的值清零。

14.8.3 用于自编程功能的寄存器

以下寄存器用于自编程功能。

- Flash 存储器编程模式控制寄存器 (FLPMC)
- Flash 存储器保护命令寄存器 (PFCMD)
- Flash 存储器状态寄存器 (PFS)
- Flash 存储器编程命令寄存器 (FLCMD)
- Flash 存储器地址指针 H 和 L (FLAPH 和 FLAPL)
- Flash 存储器地址指针 H 比较寄存器和 Flash 存储器地址指针 L 比较寄存器 (FLAPHC 和 FLAPLC)
- Flash 存储器写入缓冲寄存器 (FLW)

$\mu\text{PD78F9500}$ 、 78F9501 、 78F9502 在 Flash 存储器地址 0081H 具有称作保护字节的一个区域。

(1) Flash 存储器编程模式控制寄存器 (FLPMC)

当自编程模式下，此寄存器用于设置数据写入 Flash 存储器时工作模式，并读取保护字节的设定值。

数据仅在特定顺序下可以写入 FLPMC（请参看 14.8.3 (2) Flash 存储器保护命令寄存器 (PFCMD)），以便应用程序系统不会由于噪声干扰或程序挂起的故障而停止。

此寄存器通过 8-位内存操作指令设置。
复位信号产生使此寄存器的内容不确定。

图 14-8. Flash 存储器编程模式控制寄存器 (FLPMC) 的格式

地址: FFA2H 复位后: 不确定^{注1} R/W^{注2}

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

FLSPM	自编程模式期间工作模式的选择
0	正常模式 这是正常工作状态。执行HALT指令设置待机状态。
1	自编程模式 自编程命令可以在正常模式时通过执行特定顺序语句改变模式。 设置要写入的命令、地址和数据，接着执行HALT指令以进行自编程。

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	读取这些位的保护字节设置值。
---------	---------	---------	---------	---------	----------------

- 注 1. 复位释放时位 0 (FLSPM) 清零。复位后保护字节的设置值可从位 2 到 6 读取 (PRSELF0 到 PRSELF4)
2. 位 2 到 6 (PRSELF0 到 PRSELF4) 是只读的。

- 注意事项 1. 在设置自编程模式时的注意事项，请参看 14.8.2 自编程功能注意事项。
2. 设置 CPU 时钟，以使其在自编程期间至少为 1 MHz。
3. 进行自编程模式的特定顺序设置之后，立即执行 NOP 和 HALT 指令，接着进行自编程。此时，HALT 指令在 10 μ s (MAX.) + 2 CPU 时钟 (f_{CPU}) 之后自动释放。
4. 如果选择振荡器的时钟或外部时钟作为系统时钟，则在进行自编程模式的特定顺序设置之后立即执行 NOP 和 HALT 指令，释放 HALT 状态之后等待 8 μ s 且接着进行自编程。
5. 设置自编程模式和正常工作模式之前，清零 FLCMD 寄存器的值。

(2) Flash 存储器保护命令寄存器 (PFCMD)

如果应用程序系统由于由于噪音干扰或程序暂停的故障而意外停止，对 Flash 存储器编程模式控制寄存器 (FLPMC) 进行写入操作可能对系统有严重影响。PFCMD 用于保护 FLPMC 的写入，以使应用程序系统不意外而停止。

必须按以下特定顺序对 FLPMC 执行写入操作。

- <1> 将特定值写入 PFCMD (A5H)
- <2> 将要设置的值写入 FLPMC 的位 0 (FLSPM) (此步骤中写入无效)
- <3> 将该值的反码写入 FLPMC 的位 0 (FLSPM) (此步骤中写入无效)
- <4> 将该值写入 FLPMC 的位 0 (FLSPM) (此步骤中写入有效)

注意事项 自编程模式下不能执行中断服务。进行自编程模式的特定顺序设置之前和将此模式变为正常模式的特定顺序之后，禁用中断服务（当 MK0 = FFH 时通过执行 DI 指令）。

这样重复将值写入寄存器，可使寄存器不能被非法写入。

通过 Flash 存储器状态寄存器 (PFS) 的位 0 (FPRERR) 可以验证是否出现非法的写入操作。使用 1-位内存操作指令检查 FPRERR。

每当 FLPMC 的值改变时 A5H 必须写入 PFCMD。

可以通过 8-位内存操作指令设置 PFCMD。

复位信号产生使 PFCMD 不确定。

图 14-9. Flash 存储器保护命令寄存器 (PFCMD) 的格式

地址: FFA0H	复位后: 不确定	W								
符号	7	6	5	4	3	2	1	0		
PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0		

(3) Flash 存储器状态寄存器 (PFS)

如果数据不按正确顺序（写入 Flash 存储器保护命令寄存器 (PFCMD)）写入受保护的 Flash 存储器编程模式控制寄存器 (FLPMC)，则 FLPMC 不能写入且发生保护错误，PFS (FPRERR) 的位 0 设为 1。

当 FPRERR 为 1 时，通过写入 0 可以将其清零。

自编程期间可能出现的错误反映在 PFS 的位 1 (VCERR) 和位 2 (WEPRERR)。VCERR 或者 WEPRERR 可以通过写 0 来清零。

所有 PFS 寄存器的标志必须预清零以检查操作是否正确执行。

PFS 可以通过 1-位或者 8-位内存操作指令设置。

复位信号产生使 PFS 清零。

注意事项 使用 1-位内存操作指令检查 FPRERR。

图 14-10. Flash 存储器状态寄存器 (PFS) 的格式

地址:	FFA1H	复位后:	00H	R/W				
符号	7	6	5	4	3	2	1	0
PFS	0	0	0	0	0	WEPRERR	VCERR	FPRERR

1. FPRERR 标志的工作条件

<设置条件>

- 刚执行的对周边寄存器的写入命令没有给 PFCMD 写入特定值 (PFCMD = A5H) 时, 对 PFCMD 进行写入操作。
- 如果在<1>之后第一存储指令操作处于外围寄存器上而不是 FLPMC
- 如果在<2>之后第一存储指令操作处于外围寄存器上而不是 FLPMC
- 在<2>之后, 如果第一条存储指令所写的值不同于 FLPMC 值的反码
- 在<3>之后, 如果第一条存储指令操作的是外部存储器, 而不是 FLPMC
- 在<3>之后, 如果第一条存储指令所写的值不同于 FLPMC 值 (在<2>中所写)

备注 以上尖括弧中的号码对应于 (2) Flash 存储器保护命令寄存器 (PFCMD)。

<复位条件>

- 如果 0 写入 FPRERR 标志
- 如果产生复位信号

2. VCERR 标志的工作条件

<置位条件>

- 擦除验证错误
- 内部写入验证错误

如果 VCERR 置 1, 意味着没有正确擦除或写入 Flash 存储器。按指定程序再次擦除或者写入内存。

备注 当发生擦除或写入保护错误时, VCERR 标记也会被置 1。

<复位条件>

- 当 0 写入 VCERR 标志时
- 当产生复位信号时

3. WEPRERR 标志的工作条件

<设置条件>

- 用保护字节指定一个区域防止该区域因 FLASH 地址指针 H (FLAPH) 和执行相关命令而引起的数据擦除或写入。
- 如果 1 写入 没有擦除的一位 (数据为 0 的一位)。

<复位条件>

- 当 0 写入 WEPRERR 标志时
- 当产生复位信号时

(4) Flash 存储器编程命令寄存器 (FLCMD)

此寄存器用于指定在自编程模式下是否擦除、写入或校验 Flash 存储器。

使用 1-位或者 8-位内存操作指令设置此寄存器。

复位信号产生将此寄存器清零。

图 14-11. Flash 存储器编程命令寄存器 (FLCMD) 的格式

地址: FFA3H 复位后: 00H R/W

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

FLCMD2	FLCMD1	FLCMD0	命令名称	功能
0	0	1	内部校验 1	此命令用于检查数据是否正确写入 Flash 存储器。其用于检查数据是否已写入整个块。如果出现错误, 则 Flash 存储器状态寄存器 (PFS) 的位 1 (VCERR) 或者位 2 (WEPRERR) 置为 1。
0	1	0	内部校验 2	此命令用于检查数据是否正确写入 Flash 存储器。其用于检查数据已写入相同块。如果出现错误, 则 Flash 存储器状态寄存器 (PFS) 的位 1 (VCERR) 或者位 2 (WEPRERR) 置为 1。
0	1	1	块擦除	此命令用于擦除指定块。其用于板上模式和自编程模式。
1	0	0	块空白检测	此命令用于检查指定块是否已擦除。
1	0	1	字节写入	此命令用于将 1-字节数据写入 Flash 存储器的指定地址。指定写入地址和写入数据, 接着执行此命令。 如果将 1 写入没有擦除的一位 (数据为 0 的一位), 则 Flash 存储器状态寄存器 (PFS) 的位 2 (WEPRERR) 置为 1。
除以上数据 ^注			禁用设置	

注 如果执行这些命令意外的任何命令, 则可以立即终止命令执行, 且 Flash 存储器状态寄存器 (PFS) 的位 1 或者 2 (WEPRERR 或者 VCERR) 被置为 1。

(5) Flash 存储器地址指针 H 和 L (FLAPH 和 FLAPL)

当自编程模式下擦除、写入或者验证内存时，这些寄存器用于指定 Flash 存储器的开始地址。

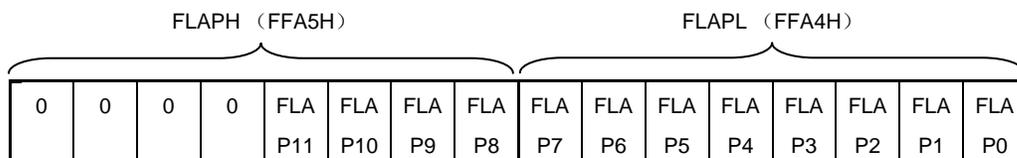
FLAPH 和 FLAPL 由计数器组成，且当不执行编程命令时其递增到这些值与 FLAPHC 和 FLAPLC 的值匹配。因此，当执行编程命令时，再次设置此值。

通过 1-位或者 8-位内存操作指令设置这些寄存器。

复位信号产生使寄存器 不确定。

图 14-12. Flash 存储器地址指针 H/L (FLAPH/FLAPL) 的格式

地址: FFA4H, FFA5H 复位后: 00H R/W



注意事项 执行自编程命令之前确保 **FLAPH** 和 **FLAPHC** 的位 4 到 7 清零。如果当执行自编程命令时这些位的值是 1，则可能装置不能正常运行。

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

这些寄存器用于指定当自编程模式下验证 Flash 存储器时内部序列操作的地址范围。

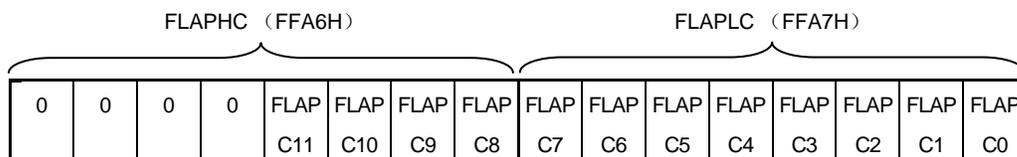
将 FLAPHC 设为与 FLAPH 相同的值。将 FLASH 存储器校验范围的最末端地址赋给 FLAPLC。

通过 1-位或者 8-位内存操作指令设置这些寄存器。

复位信号产生使这些寄存器清零。

图 14-13. Flash 存储器地址指针 H/L 比较寄存器 (FLAPHC/FLAPLC) 的格式

地址: FFA6H, FFA7H 复位后: 00H R/W



- 注意事项**
1. 执行自编程命令之前确保 **FLAPH** 和 **FLAPHC** 的位 4 到 7 清零。如果当执行自编程命令时这些位的值是 1，则可能装置不能正常运行。
 2. 设置对 **FLAPHC** 进行块擦除、验证或者空白检测的块序号（与 **FLAPH** 相同的值）。
 3. 当块擦除时 **FLAPLC** 清零，且当空白检测时此寄存器设为 **FFH**。

(7) Flash 存储器 写入缓冲寄存器 (FLW)

此寄存器用于存储要写入 Flash 存储器的数据。

通过 8-位内存操作指令设置此寄存器。

复位信号产生使该寄存器清零。

图 14-14. Flash 存储器 写入缓冲寄存器 (FLW) 的格式

地址: FFA8H 复位后: 00H R/W

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

(8) 保护字节

此保护字节用于指定要保护免于写入或者擦的区域。仅在自编程模式下指定区域有效。因为保护区的自编程无效，所以保证了写入保护区的数据。

图 14-15. 保护字节的格式 (1/2)

地址: 0081H

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

• μ PD78F9500

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	状态
0	1	1	1	0	保护块 3到0。
0	1	1	1	1	保护块 1和0。 块 2和3可以写入或者擦除。
1	1	1	1	1	所有块可以 写入或者擦除。
除以上数据					禁用设置

• μ PD78F9501

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	状态
0	1	1	0	0	保护块 7到0。
0	1	1	0	1	保护块 5到0。 块 6和7可以写入或者擦除。
0	1	1	1	0	保护块 3到0。 块 4到7可以写入或者擦除。
0	1	1	1	1	保护块 1和0。 块 2到7可以写入或者擦除。
1	1	1	1	1	所有块可以 写入或者擦除。
除以上数据					禁用设置

图 14-15. 保护字节的格式 (2/2)

• μ PD78F9502

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	状态
0	1	0	0	0	保护块 15到0。
0	1	0	0	1	保护块 13到0。 块 14和15可以写入或者擦除。
0	1	0	1	0	保护块 11到0。 块 12到15可以写入或者擦除。
0	1	0	1	1	保护块 9到0。 块 10到15可以写入或者擦除。
0	1	1	0	0	保护块 7到0。 块 8到15可以写入或者擦除。
0	1	1	0	1	保护块 5到0。 块 6到15可以写入或者擦除。
0	1	1	1	0	保护块 3到0。 块 4到15可以写入或者擦除。
0	1	1	1	1	保护块 1和0。 块 2到15可以写入或者擦除。
1	1	1	1	1	所有 块可以写入或者擦除。
除以上数据					禁用设置

14.8.4 正常模式切换到自编程模式的实例

进行自编程之前工作模式必须从正常模式切换到自编程模式。

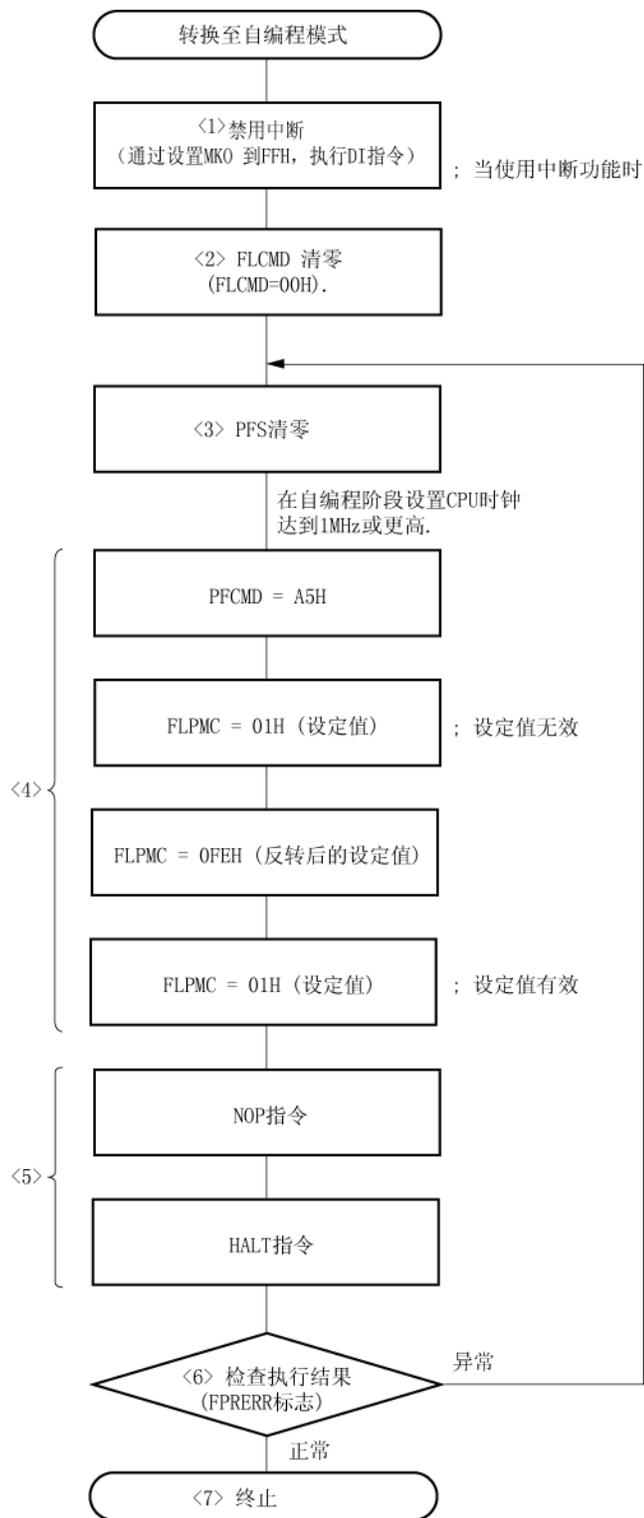
下文说明了切换自编程模式的实例。

- <1> 如果使用中断 功能则禁用中断（通过将中断屏蔽标志寄存器（MK0）设为 FFH 并执行 DI 指令）。
- <2> FLCMD 清零（FLCMD=00H）。
- <3> Flash 存储器状态寄存器（PFS）清零。
- <4> 使用特定顺序设置自编程模式。^注
 - 将特定值（A5H）写入 PFCMD。
 - 将 01H 写入 FLPMC（此步骤写入无效）。
 - 将 0FEH（01H 的反码）写入 FLPMC（此步骤写入无效）。
 - 将 01H 写入 FLPMC（此步骤写入有效）。
- <5> 执行 NOP 指令和 HALT 指令。
- <6> 使用 PFS 的位 0（FPRERR）检查特定顺序的执行结果。
异常→ <3>, 正常→ <7>
- <7> 模式切换完成。

注 自编程期间设置 CPU 时钟，以使其为 1 MHz 或者更大值。

注意事项 在执行上述操作时，必须保证用户程序所使用的地址没有数据擦除或写入操作发生。

图 14-16. 切换到自编程模式的实例



注意事项 在执行上述操作时，必须保证用户程序所使用的地址没有数据擦除或写入操作发生。

备注 图 14-16 中的<1>到 <7>对应于 14.8.4 的<1>到<7>（上一页）。

将模式切换到自编程模式的程序实例如下所示。

```
; -----  
; START  
; -----  
      MOV     MK0,#11111111B      ; 屏蔽所有中断  
      MOV     FLCMD,#00H         ; FLCMD 寄存器清零  
  
      DI  
  
ModeOnLoop:                               ; 配置设置以使 CPU 时钟≥ 1 MHz  
      MOV     PFS,#00H           ; Flash 存储器状态寄存器清零  
      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 (对设置值取反)  
      MOV     FLPMC,#01H         ; 通过 FLPMC 寄存器设置自编程模式  
                                   ; 控制 (设置值)  
  
      NOP  
      HALT  
      BT     PFS.0,$ ModeOnLoop  ; 检查写入指定寄存器的完成情况  
                                   ; 发生错误时重复相同过程。  
  
; -----  
; END  
; -----
```

14.8.5 自编程模式切换到正常模式的实例

进行自编程之工作模式必须从自编程模式返回到正常模式后。

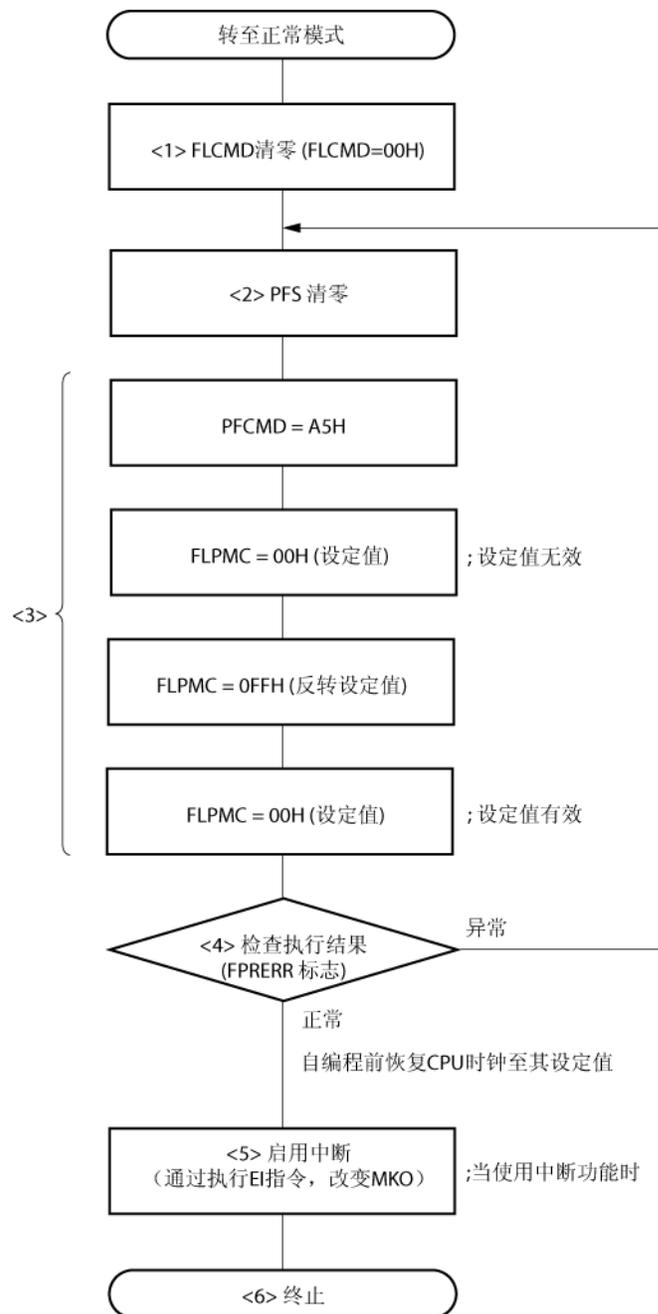
下文说明了切换到正常模式的实例。

- <1> FLCMD 清零 (FLCMD=00H)。
- <2> Flash 存储器状态寄存器 (PFS) 清零。
- <3> 使用特定顺序设置正常模式。
 - 特定值 (A5H) 写入 PFCMD
 - 00H 写入 FLPMC (此步骤写入无效)
 - 0FFH (00H 的反码) 写入 FLPMC (此步骤写入无效)
 - 将 00H 写入 FLPMC (此步骤写入有效)
- <4> 使用 PFS 的位 0 (FPRERR) 检查特定顺序的执行结果。
异常→ <2>, 正常→ <5>
- <5> 启用中断服务 (通过执行 EI 指令和改变 MK0) 以恢复初始状态。
- <6> 模式切换完成。

注 在指定顺序正常完成后, 恢复自编程之前的 CPU 时钟设置。

注意事项 在执行上述操作时, 必须保证用户程序所使用的地址没有数据擦除或写入操作发生。

图 14-17. 切换到正常模式的实例



注意事项 在执行上述操作时，必须保证用户程序所使用的地址没有数据擦除或写入操作发生。

备注 图 14-17 中的<1>到<6>对应于 14.8.5 中的<1>到<6>（上一页）。

将模式切换到正常模式的程序实例如下所示。

```

; -----
; START
; -----

        MOV     FLCMD,#00H      ; FLCMD 寄存器清零
ModeOffLoop:
        MOV     PFS,#00H       ; Flash 存储器状态寄存器清零
        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 (设置值取反)
        MOV     FLPMC,#00H     ; 经由 FLPMC 寄存器控制设置正常模式 (设置值)

        BT     PFS.0,$ModeOffLoop ; 检查写入特定寄存器的完成情况
                                           ; 发生错误时重复相同的处理
                                           ; 恢复自编程之前的 CPU 时钟设置
                                           ;

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

        EI

; -----
; END
; -----

```

14.8.6 自编程模式下块擦除操作的实例

下文说明了自编程模式下块擦除操作的实例。

- <1> 将 03H（块擦除）设置到 Flash 存储器程序命令寄存器（FLCMD）。
- <2> 将要擦除的块序号设置到 Flash 存储器地址指针 H（FLAPH）。
- <3> 将 Flash 存储器地址指针 L（FLAPL）设为 00H。
- <4> 将与 FLAPH 相同的值写入 Flash 存储器地址指针 H 比较寄存器（FLAPHC）。
- <5> 将 Flash 存储器地址指针 L 比较寄存器（FLAPLC）设为 00H。
- <6> Flash 存储器状态寄存器（PFS）清零。
- <7> 将 ACH 写入看门狗定时器启用寄存器（WDTE）（看门狗定时计数器清零并重启）[※]。
- <8> 执行 HALT 指令接着开始自编程。（如果已经执行自编程，则在 HALT 指令之后立即执行一指令。）
- <9> 使用 PFS 的位 1（VCERR）和位 2（WEPRERR）检查自编程错误是否已经发生。
异常 → <10>
正常 → <11>
- <10> 块擦除过程异常终止。
- <11> 块擦除过程正常终止。

注 当不使用看门狗定时器时，不需要此设置。

图 14-18. 自编程模式下块擦除操作的实例



注 当不使用看门狗定时器时，不需要此设置。

备注 图 14-18 中的<1>到<11>对应于 14.8.6 的<1>到<11>（上一页）。

以下给出了自编程模式下进行块擦除的程序实例。

```
; -----  
; START  
; -----  
  
FlashBlockErase:  
    MOV     FLCMD,#03H      ; 设置 Flash 存储器控制命令（块擦除）  
    MOV     FLAPH,#07H     ; 设置要擦除的块序号（此处指定块 7）  
    MOV     FLAPL,#00H     ; 将 FLAPL 固定为“00H”  
    MOV     FLAPHC,#07H    ; 设置擦除块比较号码（与 FLAPH 的值相同）  
    MOV     FLAPLC,#00H    ; 将 FLAPLC 固定为“00H”  
  
    MOV     PFS,#00H       ; Flash 存储器状态寄存器清零  
    MOV     WDTE,#0ACH     ; 清零&重启 WDT  
    HALT                    ; 开始自编程  
    MOV     A,PFS  
    MOV     CmdStatus,A    ; 执行结果存储在变量中  
                                ; （CmdStatus = 0: 正常终止, 0 以外: 异常  
                                ; 终止）  
  
; -----  
; END  
; -----
```

14.8.7 自编程模式下空白块检测操作的实例

下文说明了自编程模式下空白块检测操作的实例。

- <1> 将 04H（空白块检测）设置到 Flash 存储器程序命令寄存器（FLCMD）。
- <2> 将进行空白检测的块序号设为 Flash 存储器地址指针 H（FLAPH）。
- <3> 将 Flash 存储器地址指针 L（FLAPL）设为 00H。
- <4> 将与 FLAPH 相同的值写入 Flash 存储器地址指针 H 比较寄存器（FLAPHC）。
- <5> 将 Flash 存储器地址指针 L 比较寄存器（FLAPLC）设为 00H。
- <6> Flash 存储器状态寄存器（PFS）清零。
- <7> 将 ACH 写入看门狗定时器启用寄存器（WDTE）（看门狗定时计数器清零并重启）[※]。
- <8> 执行 HALT 指令接着开始自编程。（如果已经执行自编程，则在 HALT 指令之后立即执行一指令。）
- <9> 使用 PFS 的位 1（VCERR）和位 2（WEPRERR）检查自编程错误是否已经发生。
 - 异常 → <10>
 - 正常 → <11>
- <10> 块空白检测异常终止。
- <11> 块空白检测正常终止。

注 当不使用看门狗定时器时，不需要此设置。

图 14-19. 自编程模式下块空白检测操作的实例



注 当不使用看门狗定时器时不需要此设置不需要此设置。

备注 图 14-19 中的<1>到<11>对应于 14.8.7 的<1>到<11>（上一页）。

下文给出了自编程模式下进行空白块检测的程序实例。

```

; -----
; START
; -----

FlashBlockBlankCheck:
    MOV    FLCMD,#04H    ; 设置 Flash 存储器控制命令（空白块检测）
    MOV    FLAPH,#07H    ; 设置空白检测的块序号（此处指定块 7
                        ; ）
    MOV    FLAPL,#00H    ; 将 FLAPL 固定为“00H”
    MOV    FLAPHC,#07H   ; 设置 空白检测块比较号码（与 FLAPH
                        ; 值相同）
    MOV    FLAPLC,#0FFH  ; 将 FLAPLC 固定为“FFH”

    MOV    PFS,#00H      ; Flash 存储器状态寄存器清零
    MOV    WDTE,#0ACH    ; 清零&重启 WDT
    HALT                    ; 开始自编程
    MOV    A,PFS
    MOV    CmdStatus,A    ; 执行结果存储在变量中
                        ; （CmdStatus = 0: 正常终止, 0 以外: 异常
                        ; 终止）

; -----
; END
; -----

```

14.8.8 自编程模式下字节写入操作的实例

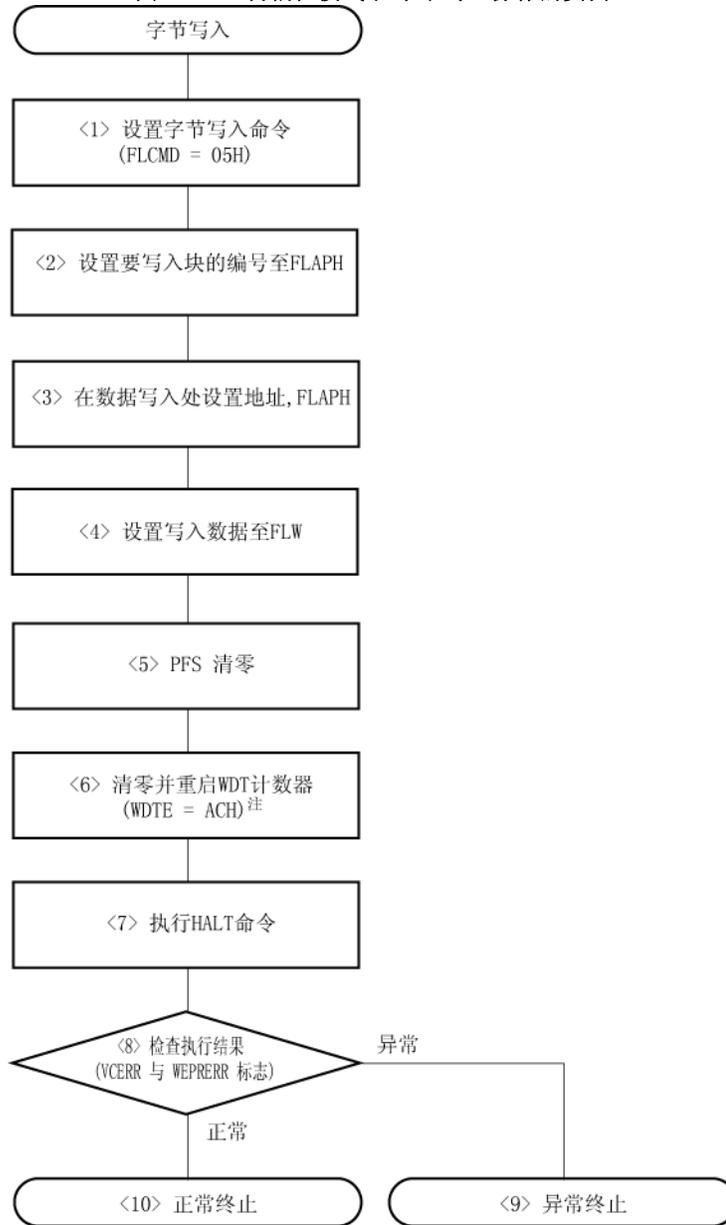
下文说明了自编程模式下字节写入操作的实例。

- <1> 将 05H（字节写入）设置到 Flash 存储器程序命令寄存器（FLCMD）。
- <2> 将要写入数据的块序号设为 Flash 存储器地址指针 H（FLAPH）。
- <3> 将要写入数据的地址设为 Flash 存储器地址指针 L（FLAPL）。
- <4> 将要写入的数据设置到 Flash 存储器 写入缓冲寄存器（FLW）。
- <5> Flash 存储器状态寄存器（PFS）清零。
- <6> 将 ACH 写入看门狗定时器启用寄存器（WDTE）（看门狗定时计数器清零并重启）^注。
- <7> 执行 HALT 指令接着开始自编程。（如果已经执行自编程，则在 HALT 指令之后立即执行一指令。）
- <8> 使用 PFS 的位 1（VCERR）和位 2（WEPRERR）检查自编程错误是否已经发生。
 - 异常 → <9>
 - 正常 → <10>
- <9> 字节写入过程异常终止。
- <10> 字节写入过程正常终止。

注 当不使用看门狗定时器时，不需要此设置。

注意事项 如果写入结果失败，将该块擦除一次并再次写入。

图 14-20. 自编程模式下 字节写入操作的实例



注 当不使用看门狗定时器时，不需要此设置。

备注 图 14-20 中的<1>到<10>对应于 14.8.8 中的<1>到<10>（上一页）。

下文给出了自编程模式下进行字节写入的程序实例。

```
; -----  
; START  
; -----  
FlashWrite:  
    MOV    FLCMD,#05H    ; 设置 Flash 存储器控制命令（字节写入）  
    MOV    FLAPH,#07H    ; 通过 FLAPH 设置要输入数据的地址  
                        ; （此处指定块 7）  
    MOV    FLAPL,#20H    ; 通过 FLAPL 设置要输入数据的地址  
                        ; （此处指定地址 20H）  
    MOV    FLW,#10H      ; 设置要写入的数据（此处指定 10H）  
  
    MOV    PFS,#00H      ; Flash 存储器状态寄存器清零  
    MOV    WDTE,#0ACH    ; 清零&重启 WDT  
    HALT                               ; 开始自编程  
    MOV    A,PFS  
    MOV    CmdStatus,A    ; 执行结果存储在变量中  
                        ; （CmdStatus = 0: 正常终止, 0 以外: 异常  
                        ; 终止）  
  
; -----  
; END  
; -----
```

14.8.9 自编程模式下内部验证操作的实例

下文说明了自编程模式下内部验证操作的实例。

• 内部校验 1

- <1> 将 01H（内部校验 1）设置到 Flash 存储器程序命令寄存器（FLCMD）。
- <2> 将进行内部验证的块序号设为 Flash 存储器地址指针 H（FLAPH）。
- <3> 将 Flash 存储器地址指针 L（FLAPL）设为 00H。
- <4> 将与 FLAPH 相同的值写入 Flash 存储器地址指针 H 比较寄存器（FLAPHC）。
- <5> 将 Flash 存储器地址指针 L 比较寄存器（FLAPLC）设为 00H。
- <6> Flash 存储器状态寄存器（PFS）清零。
- <7> 将 ACH 写入看门狗定时器启用寄存器（WDTE）（看门狗定时计数器清零并重启）^注。
- <8> 执行 HALT 指令接着开始自编程。（如果已经执行自编程，则在 HALT 指令之后立即执行一指令。）
- <9> 使用 PFS 的位 1（VCERR）和位 2（WEPRERR）检查自编程错误是否已经发生。
 - 异常 → <10>
 - 正常 → <11>
- <10> 内部验证过程异常终止。
- <11> 内部验证过程正常终止。

• 内部校验 2

- <1> 将 02H（内部校验 2）设置到 Flash 存储器程序命令寄存器（FLCMD）。
- <2> 将进行内部验证的块序号设为 Flash 存储器地址指针 H（FLAPH）。
- <3> 将 Flash 存储器地址指针 L（FLAPL）设为开始地址。
- <4> 将与 FLAPH 相同的值写入 Flash 存储器地址指针 H 比较寄存器（FLAPHC）。
- <5> 将 Flash 存储器地址指针 L 比较寄存器（FLAPLC）设为结束地址。
- <6> Flash 存储器状态寄存器（PFS）清零。
- <7> 将 ACH 写入看门狗定时器启用寄存器（WDTE）（看门狗定时计数器清零并重启）^注。
- <8> 执行 HALT 指令接着开始自编程。（如果已经执行自编程，则在 HALT 指令之后立即执行一指令。）
- <9> 使用 PFS 的位 1（VCERR）和位 2（WEPRERR）检查自编程错误是否已经发生。
 - 异常 → <10>
 - 正常 → <11>
- <10> 内部验证过程异常终止。
- <11> 内部验证过程正常终止。

注 当不使用看门狗定时器时，不需要此设置。

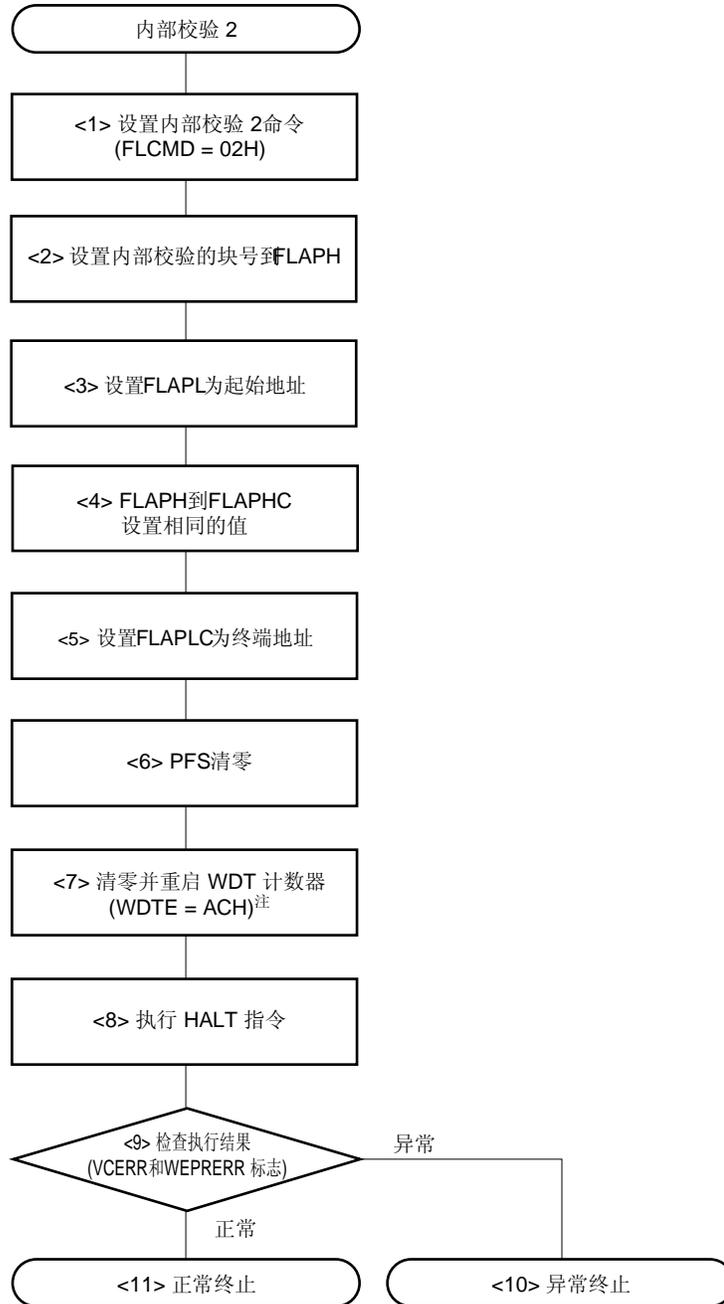
图 14-21. 自编程模式下内部验证操作的实例



注 当不使用看门狗定时器时，不需要此设置。

备注 图 14-21 中的<1>到<11>对应于 14.8.9 中的内部校验 1 <1>到<11>（上一页）。

图 14-22. 自编程模式下内部验证操作的实例



注 当不使用看门狗定时器时，不需要此设置。

备注 图 14-22 中的<1>到<11> 对应于 14.8.9 中的内部校验 2 <1>到<11> （前一页）。

下文给出了自编程模式下进行内部验证的程序实例。

• 内部校验 1

```

; -----
; START
; -----
FlashVerify:
    MOV     FLCMD,#01H    ; 设置 Flash 存储器控制命令（内部校验 1）
    MOV     FLAPH,#07H    ; 将内部验证的块序号
                        ; 设为 FLAPH（实例：此处指定块 7）
    MOV     FLAPL,#00H    ; 将 FLAPL 设为 00H
    MOV     FLAPHC,#07H
    MOV     FLAPLC,#FFH   ; 将 FLAPLC 设为 FFH

    MOV     PFS,#00H      ; Flash 存储器状态寄存器清零
    MOV     WDTE,#0ACH    ; 清零&重启 WDT
    HALT                                ; 开始自编程
    MOV     A,PFS
    MOV     CmdStatus,A   ; 执行结果存储在变量中
                        ; （CmdStatus = 0：正常终止, 0 以外：异常
                        ; 终止）

; -----
; END
; -----

```

• 内部校验 2

```

; -----
; START
; -----
FlashVerify:
    MOV     FLCMD,#02H    ; 设置 Flash 存储器控制命令（内部校验 2）
    MOV     FLAPH,#07H    ; 将内部验证的块序号
                        ; 设为 FLAPH（实例：此处指定块 7）
    MOV     FLAPL,#00H    ; 将 FLAPL 设为校验开始地址（实例：此处指定地址
                        ; 00H）
    MOV     FLAPHC,#07H
    MOV     FLAPLC,#20H   ; 设置 FLAPLC 设为校验结束地址（实例：此处指定地址
                        ; 20H）

    MOV     PFS,#00H      ; Flash 存储器状态寄存器清零
    MOV     WDTE,#0ACH    ; 清零&重启 WDT
    HALT                                ; 开始自编程
    MOV     A,PFS
    MOV     CmdStatus,A   ; 执行结果存储在变量中
                        ; （CmdStatus = 0：正常终止, 0 以外：异常
                        ; 终止）

; -----
; END
; -----

```

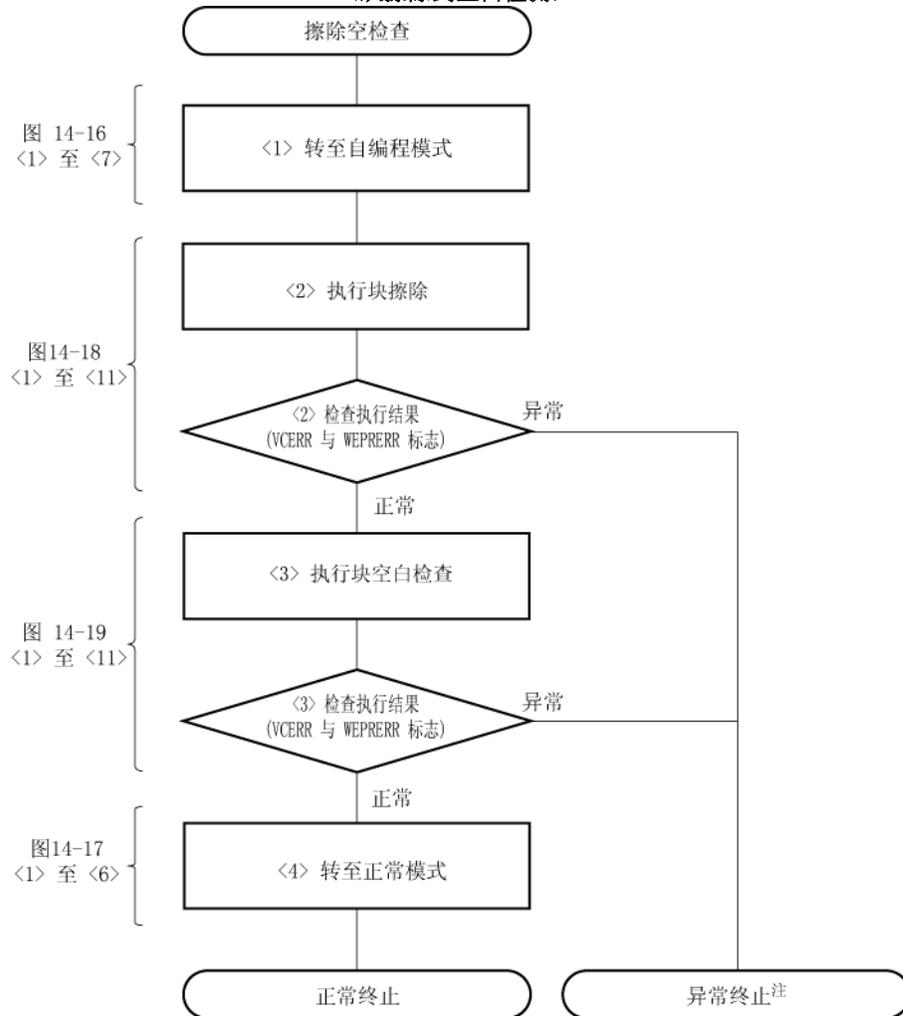
14.8.10 自编程模式下应最小化命令执行时间的操作实例

下文说明了自编程模式下应最小化命令执行时间时的操作实例。

(1) 擦除空白检测

- <1> 模式从正常模式切换到自编程模式（14.8.4 中的<1>到<7>）
- <2> 执行块擦除→错误检查（14.8.6 中的<1>到<11>）
- <3> 执行块空白检测→错误检查（14.8.7 中的<1>到<11>）
- <4> 模式从自编程模式切换到正常模式（14.8.5 中的<1>到<6>）

图 14-23. 最短命令执行时间时的操作实例
(从擦除到空白检测)



注 进行过程切换到正常模式以返回到正常过程。

备注 图 14-23 中的<1>到<4>对应于上文 14.8.10 (1) 中的<1>到<4>。

下文给出了自编程模式下 应最小化命令执行时间（从擦除到空白检测）时的程序实例。

```

; -----
; START
; -----
      MOV     MK0,#11111111B   ; 屏蔽所有中断
      MOV     FLCMD,#00H      ; FLCMD 寄存器清零

      DI

ModeOnLoop:
      ; 配置设置以使 CPU 时钟 ≥ 1 MHz
      MOV     PFS,#00H        ; Flash 存储器状态寄存器清零
      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（设置值取反）
      MOV     FLPMC,#01H      ; 通过 FLPMC 寄存器设置自编程模式控制（设置
                               ; 值）

      NOP
      HALT
      BT     PFS.0,$ModeOnLoop ; 检查写入特定寄存器的完成情况
                               ; 发生错误时重复相同过程。

FlashBlockErase:
      MOV     FLCMD,#03H      ; 设置 Flash 存储器控制命令（块擦除）
      MOV     FLAPH,#07H      ; 设置要擦除的块序号（此处指定块 7
                               ; ）
      MOV     FLAPL,#00H      ; 将 FLAPL 固定为“00H”
      MOV     FLAPHC,#07H     ; 设置擦除块比较号码（与 FLAPH
                               ; 值相同）
      MOV     FLAPLC,#00H     ; 将 FLAPLC 固定为“00H”

      MOV     WDTE,#0ACH      ; 清零&重启 WDT
      HALT                    ; 开始自编程
      MOV     A,PFS
      CMP     A,#00H
      BNZ     $StatusError    ; 检查擦除错误
                               ; 当出现错误时执行异常终止
                               ; 过程。

FlashBlockBlankCheck:
      MOV     FLCMD,#04H      ; 设置 Flash 存储器控制命令（空白块检测）
      MOV     FLAPH,#07H      ; 设置空白检测的块序号（此处指定块 7
                               ; ）
      MOV     FLAPL,#00H      ; 将 FLAPL 固定为“00H”
      MOV     FLAPHC,#07H     ; 设置空白检测块比较号码（与 FLAPH
                               ; 值相同）

```

```

MOV     FLAPLC,#0FFH    ; 将 FLAPLC 固定为“FFH”
MOV     WDTE,#0ACH     ; 清零&重启 WDT
HALT
MOV     A,PFS
CMP     A,#00H
BNZ     $StatusError   ; 检查空白检测错误
                        ; 当出现错误时执行异常终止
                        ; 过程。

MOV     FLCMD,#00H     ; FLCMD 寄存器清零
ModeOffLoop:
MOV     PFS,#00H       ; Flash 存储器状态寄存器清零
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（设置值反转）
MOV     FLPMC,#00H    ; 经由 FLPMC 寄存器控制设置正常模式（设置值）

BT PFS.0,$ModeOffLoop ; 检查写入特定寄存器的完成情况
                        ; 发生错误时重复相同过程。
                        ; 正确执行特定顺序之后，
                        ; 恢复自编程开始前的 CPU 时钟设置

MOV     MK0,#INT_MK0   ; 恢复中断屏蔽标志

EI

BR     StatusNormal

; -----
; END（异常终止过程）；进行过程切换到
; 正常模式以返回到正常过程
; -----
StatusError:

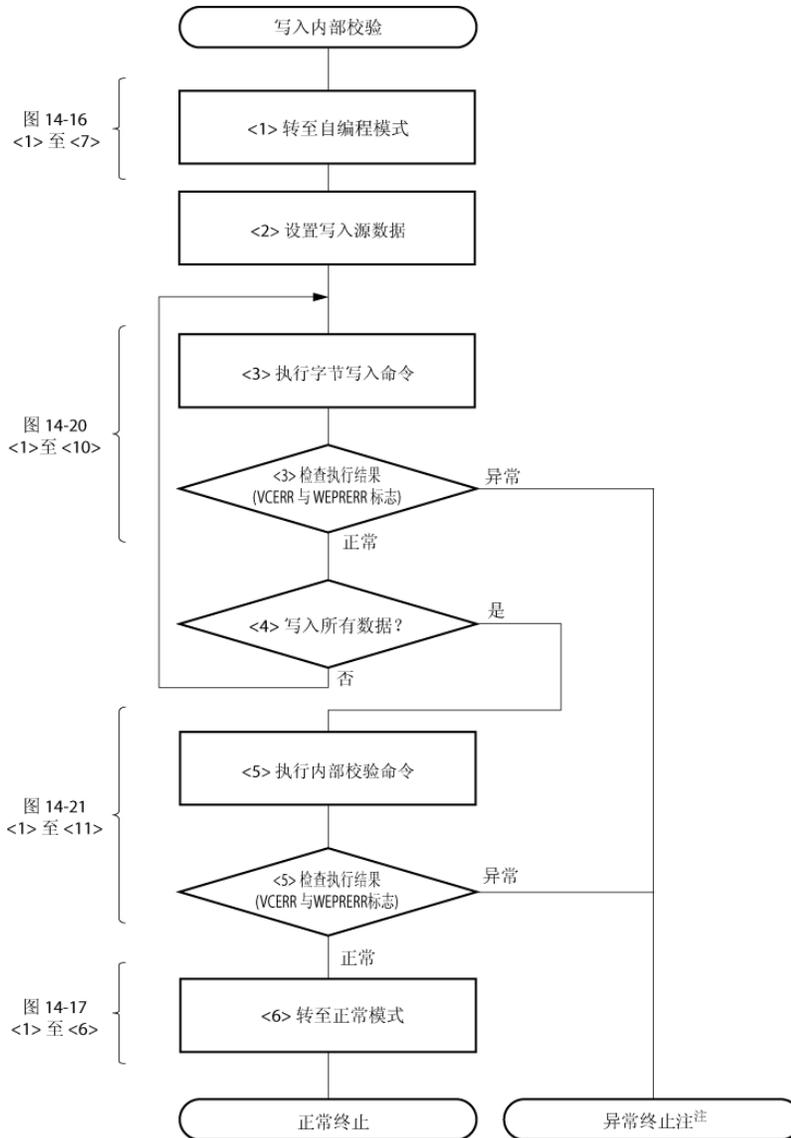
; -----
; END（正常终止过程）
; -----
StatusNormal:

```

(2) 写入内部验证

- <1> 模式从正常模式切换到自编程模式（14.8.4 中的<1>到<7>）
- <2> 写入源数据确定
- <3> 进行字节写入→错误检查（14.8.8 中的<1>到<10>）
- <4> 重复<3>直到所有数据写入
- <5> 进行内部验证→错误检查（14.8.9 中的<1>到<11>）
- <6> 模式从自编程模式切换到正常模式（14.8.5 中的<1>到<6>）

图 14-24. 应最小化命令执行时间时的操作实例
(从写入到内部验证)



注 进行过程切换到正常模式以返回到正常过程。

备注 图 14-24 中的<1>到<6>对应于上文 14.8.10 (2) 中的<1>到<6>。

下文给出了自编程模式下应最小化命令 执行时间时（从写入到内部验证）的程序实例。

```

; -----
; START
; -----
      MOV     MK0,#11111111B   ; 屏蔽所有中断
      MOV     FLCMD,#00H      ; FLCMD 寄存器清零
      DI

ModeOnLoop:
      ; 配置设置以使 CPU 时钟≥ 1 MHz
      MOV     PFS,#00H        ; Flash 存储器状态寄存器清零
      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（设置值反转）
      MOV     FLPMC,#01H      ; 通过 FLPMC 寄存器设置自编程模式控制
      ; （设置值）

      NOP
      HALT
      BT PFS.0,$ModeOnLoop   ; 检查写入特定寄存器的完成情况
      ; 发生错误时重复相同过程。

FlashWrite:
      MOVW    HL,#DataAdrTop  ; 设置定位要写入的数据的地址
      MOVW    DE,#WriteAdr    ; 设置要写入的数据的地址
FlashWriteLoop:
      MOV     FLCMD,#05H      ; 设置 Flash 存储器控制命令（字节写）
      MOV     A,D
      MOV     FLAPH,A         ; 设置要写入的数据的地址
      MOV     A,E
      MOV     FLAPL,A         ; 设置要写入的数据的地址
      MOV     A,[HL]
      MOV     FLW,A           ; 设置要写入的数据

      MOV     WDTE,#0ACH      ; 清零&重启 WDT
      HALT                    ; 开始自编程
      MOV     A,PFS
      CMP     A,#00H
      BNZ     $StatusError    ; 检查写入错误
      ; 当出现错误时执行异常终止
      ; 过程。

      INCW    HL               ; 定位要写入数据的地址 + 1
      MOVW    AX,HL
      CMPW    AX,#DataAdrBtm  ; 进行内部校验过程
      BNC     $FlashVerify    ; 如果完成所有数据的写入

```

```

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

```

FlashVerify:

```

MOVW HL,#WriteAdr ; 设置校验地址

MOV FLCMD,#02H ; 设置 Flash 存储器控制命令（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 ; 清零&重启 WDT
HALT ; 开始自编程
MOV A,PFS
CMP A,#00H
BNZ $StatusError ; 检查内部校验错误
; 当出现错误时执行异常终止
; 过程。

MOV FLCMD,#00H ; FLCMD 寄存器清零
ModeOffLoop:
MOV PFS,#00H ; Flash 存储器状态寄存器清零
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（设置值反转）
MOV FLPMC,#00H ; 经由 FLPMC 寄存器控制设置正常模式（设置值）

BT PFS.0,$ModeOffLoop ; 检查写入特定寄存器的完成情况
; 发生错误时重复相同过程。
; 正确执行特定顺序之后，
; 恢复自编程开始前的 CPU 时钟设置

MOV MK0,#INT_MK0 ; 恢复中断屏蔽标志

EI

BR StatusNormal

```

```

; -----
; END（异常终止过程）：进行过程切换到
; 正常模式以返回到正常过程
; -----

```

StatusError:

```

; -----
; END (正常终止过程)
; -----
StatusNormal:

; -----
; 要写入的数据
; -----
DataAdrTop:
    DB    XXH
    DB    XXH
    DB    XXH
    DB    XXH

    :
    :

    DB    XXH
DataAdrBtm:
; -----

```

备注 在上面的程序例子中使用了内部校验 2。验证一个完整的块，使用内部校验 1。

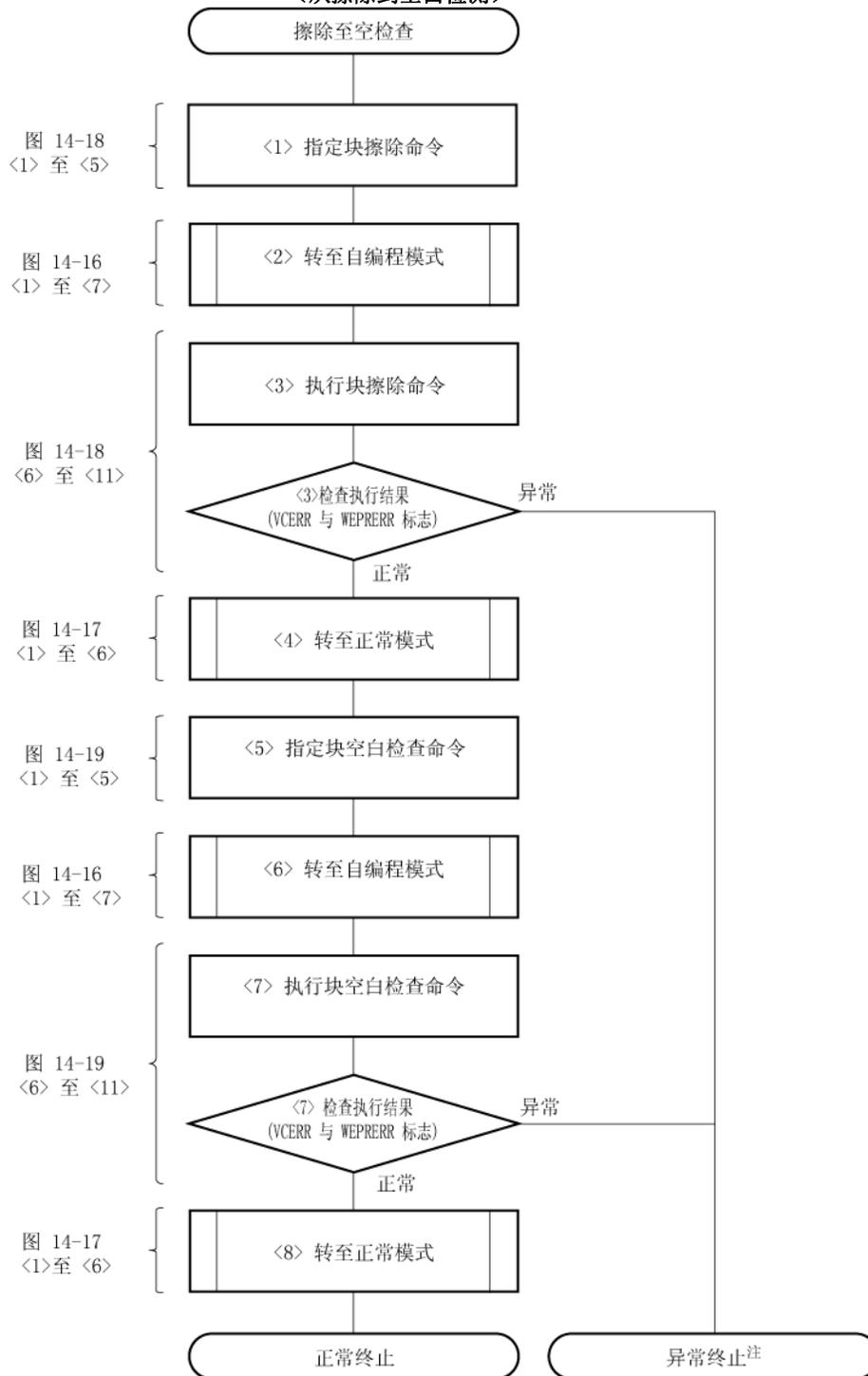
14.8.11 自编程模式下应最短中断-禁用时间时的操作实例

下文说明了自编程模式下应最短中断-禁用时间时的操作实例。

(1) 擦除空白检测

- <1> 块擦除命令说明 (14.8.6 中的<1>到<5>)
- <2> 模式从正常模式切换到自编程模式 (14.8.4 中的<1>到<7>)
- <3> 执行块擦除命令 → 错误检查 (14.8.6 中的<6>到<11>)
- <4> 模式从自编程模式切换到正常模式 (14.8.5 中的<1>到<6>)
- <5> 块空白检测命令说明 (14.8.7 中的<1>到<5>)
- <6> 模式从正常模式切换到自编程模式 (14.8.4 中的<1>到<7>)
- <7> 执行块空白检测命令 → 错误检查 (14.8.7 中的<6>到<11>)
- <8> 模式从自编程模式切换到正常模式 (14.8.5 中的<1>到<6>)

图 14-25. 最短中断-禁用时间时的操作实例
(从擦除到空白检测)



注 进行过程切换到正常模式以返回到正常过程。

备注 图 14-25 中的<1>到<8>对应于 14.8.11 (1) 中的<1>到<8> (上一页)。

下文给出了自编程模式下应最短中断-禁用时间时（从擦除到空白检测）的程序实例。

```

; -----
; START
; -----
FlashBlockErase:
    ; 设置 擦除命令
    MOV     FLCMD,#03H      ; 设置 Flash 存储器控制命令（块擦除）
    MOV     FLAPH,#07H     ; 设置要擦除的块序号（此处指定块 7）
    MOV     FLAPL,#00H     ; 将 FLAPL 固定为“00H”
    MOV     FLAPHC,#07H   ; 设置擦除块比较号码（与 FLAPH 的值相同）
    MOV     FLAPLC,#00H   ; 将 FLAPLC 固定为“00H”

    CALL    IModeOn       ; 切换到自编程模式

    ; 执行擦除命令
    MOV     PFS,#00H      ; Flash 存储器状态寄存器清零
    MOV     WDTE,#0ACH    ; 清零&重启 WDT
    HALT                    ; 开始自编程
    MOV     A,PFS
    CMP     A,#00H
    BNZ     $StatusError  ; 检查擦除错误
                                ; 当出现错误时执行异常终止
                                ; 过程。

    CALL    IModeOff     ; 切换到正常模式

    ; 设置空白检测命令
    MOV     FLCMD,#04H    ; 设置 Flash 存储器控制命令（块空白检测）
    MOV     FLAPH,#07H    ; 设置空白检测的块序号（此处指定块 7）
    MOV     FLAPL,#00H    ; 将 FLAPL 固定为“00H”
    MOV     FLAPHC,#07H   ; 设置空白检测块比较号码（与 FLAPH
                                ; 值相同）
    MOV     FLAPLC,#0FFH  ; 将 FLAPLC 固定为“FFH”

    CALL    IModeOn       ; 切换到自编程模式

    ; 执行空白检测命令
    MOV     PFS,#00H      ; Flash 存储器状态寄存器清零
    MOV     WDTE,#0ACH    ; 清零&重启 WDT
    HALT                    ; 开始自编程
    MOV     A,PFS
    CMP     A,#00H
    BNZ     $StatusError  ; 检查空白检测错误
                                ; 当出现错误时执行异常终止过程

```

```

CALL    !ModeOff    ; 切换到正常模式

        BR          StatusNormal

; -----
; END (异常终止过程); 进行过程切换到
        正常模式以返回到正常过程
; -----
StatusError:

; -----
; END (正常终止过程)
; -----
StatusNormal:

; -----
; 过程切换到自编程模式
; -----
ModeOn:
        MOV        MK0,#11111111B    ; 屏蔽所有中断
        MOV        FLCMD,#00H        ; FLCMD 寄存器清零

        DI

ModeOnLoop:
; 配置设置以使 CPU 时钟≥ 1 MHz
        MOV        PFS,#00H          ; Flash 存储器状态寄存器清零
        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 (设置值反转)
        MOV        FLPMC,#01H        ; 经由 FLPMC 寄存器控制设置自编程模式 (设置
; 值)

        NOP
        HALT
        BT PFS.0,$ModeOnLoop        ; 检查写入特定寄存器的完成情况
; 发生错误时重复相同过程。

        RET

; -----
; 过程切换到正常模式
; -----
ModeOffLoop:
        MOV        FLCMD,#00H        ; FLCMD 寄存器清零
        MOV        PFS,#00H          ; Flash 存储器状态寄存器清零
        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 (设置值反转)
MOV    FLPMC,#00H    ; 经由 FLPMC 寄存器控制设置正常模式 (设置值)

BT PFS.0,$ModeOffLoop ; 检查写入特定寄存器的完成情况
                        ; 发生错误时重复相同过程。
                        ; 正确执行特定顺序之后,
                        ; 自编程之前将 CPU 时钟恢复其设置

MOV    MK0,#INT_MK0  ; 恢复中断屏蔽标志

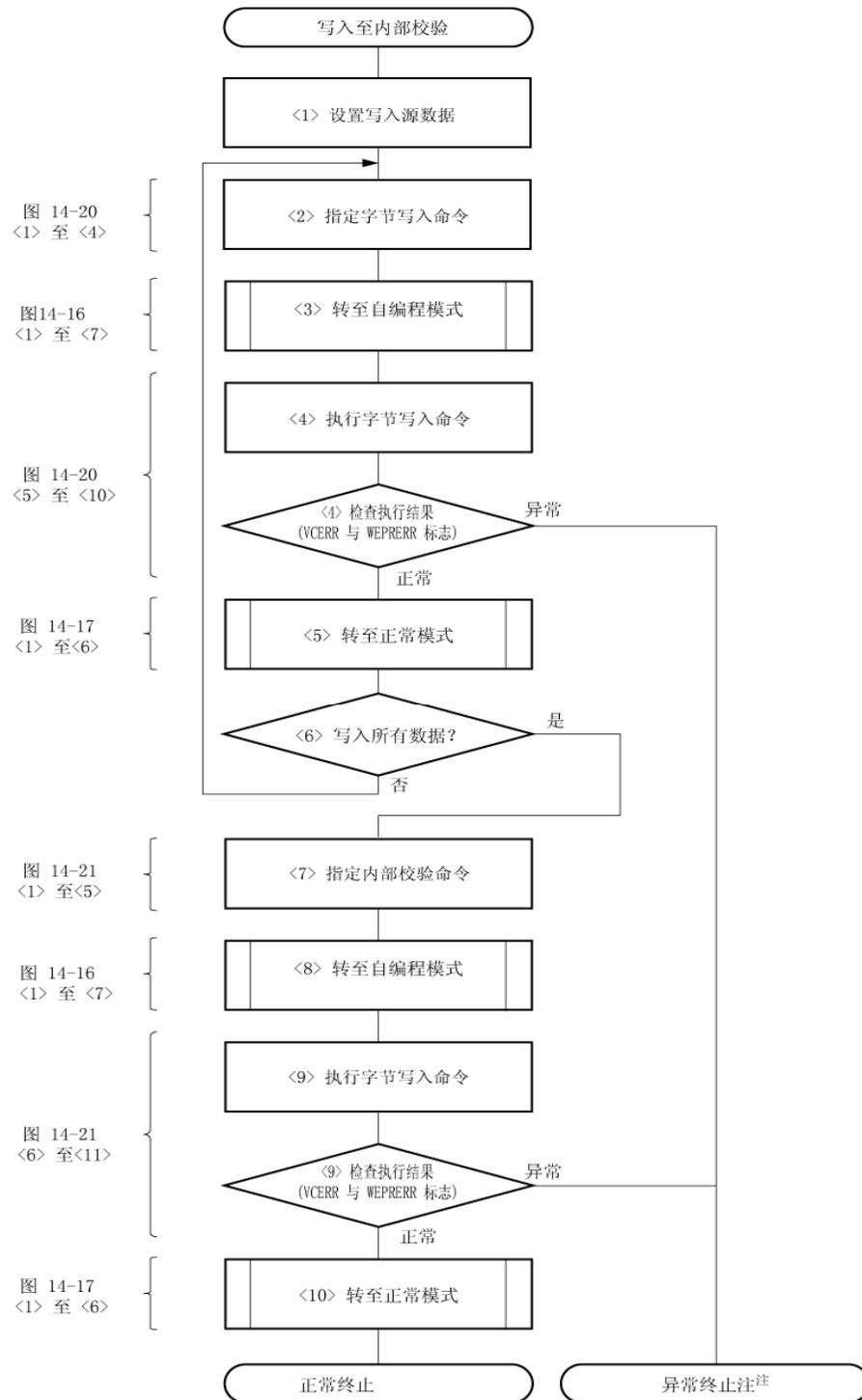
EI

RET
```

(2) 写入内部验证

- <1> 写入源数据的说明
- <2> 字节写入命令的说明 (14.8.8 中的<1>到<4>)
- <3> 模式从正常模式切换到自编程模式 (14.8.4 中的<1>到<7>)
- <4> 进行字节写入命令 → 错误检查 (14.8.8 中的<5>到<10>)
- <5> 模式从自编程模式切换到正常模式 (14.8.5 中的<1>到<6>)
- <6> 重复<2>到<5>直到写入所有数据
- <7> 指定内部校验命令 (14.8.9 中的<1>到<5>)
- <8> 模式从正常模式切换到自编程模式 (14.8.4 中的<1>到<7>)
- <9> 进行内部校验命令 → 错误检查 (14.8.9 中的<6>到<11>)
- <10> 模式从自编程模式切换到正常模式 (14.8.5 中的<1>到<6>)

图 14-26. 最短中断-禁用时间时的操作实例
(从写入到内部验证)



注 进行过程切换到正常模式以返回到正常过程。

备注 图 14-26 中的<1>到<10>对应于 14.8.11 (2) 中的<1>到<10> (上一页)。

下文给出了自编程模式下最短中断-禁用时间（从写入到内部校验）时的程序实例。

```

; -----
; START
; -----
; 设置写入命令
FlashWrite:
    MOVW    HL,#DataAdrTop    ; 设置定位要写入的数据的地址
    MOVW    DE,#WriteAdr      ; 设置要写入的数据的地址

FlashWriteLoop:
    MOV     FLCMD,#05H        ; 设置 Flash 存储器控制命令（字节写入）
    MOV     A,D
    MOV     FLAPH,A           ; 设置要写入的数据的地址
    MOV     A,E
    MOV     FLAPL,A           ; 设置要写入的数据的地址
    MOV     A,[HL]
    MOV     FLW,A             ; 设置要写入的数据

    CALL    !ModeOn           ; 切换到自编程模式

; 执行写入命令
    MOV     PFS,#00H          ; Flash 存储器状态寄存器清零
    MOV     WDTE,#0ACH        ; 清零&重启 WDT
    HALT                                ; 开始自编程
    MOV     A,PFS
    CMP     A,#00H
    BNZ     $StatusError      ; 检查写入错误
                                ; 当出现错误时执行异常终止
                                ; 过程。

    CALL    !ModeOff          ; 切换到正常模式

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

EI

; 判断写入所有数据
    INCW    HL                 ; 定位要写入数据的地址 + 1
    MOVW    AX,HL
    CMPW    AX,#DataAdrBtm    ; 进行内部验证过程
    BNC     $FlashVerify      ; 如果完成所有数据的写入

    INCW    DE                 ; 要写入数据的地址 + 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 存储器控制命令（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

CALL    !ModeOn         ; 切换到自编程模式

; 进行内部校验命令
MOV     PFS,#00H        ; Flash 存储器状态寄存器清零
MOV     WDTE,#0ACH      ; 清零&重启 WDT
HALT
MOV     A,PFS
CMP     A,#00H
BNZ     $StatusError    ; 检查内部校验错误
; 当出现错误时执行异常终止过程

CALL    !ModeOff        ; 切换到正常模式

BR     StatusNormal

```

```

; -----
; END（异常终止过程）；进行过程切换到
; 正常模式以返回到正常过程
; -----

```

StatusError:

```

; -----
; END（正常终止过程）
; -----

```

StatusNormal:

```

; -----
; 过程切换到自编程模式
; -----

```

ModeOn:

```

MOV     MK0,#11111111B  ; 屏蔽所有中断
MOV     FLCMD,#00H      ; FLCMD 寄存器清零

```

DI

```

ModeOnLoop:
MOV     PFS,#00H      ; Flash 存储器状态寄存器清零
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 (设置值反转)
MOV     FLPMC,#01H    ; 经由 FLPMC 寄存器控制设置自编程模式 (设置
; 值)

```

NOP

HALT

```

BT PFS.0,$ModeOnLoop ; 检查写入特定寄存器的完成情况
; 发生错误时重复相同过程。

```

RET

; -----

; 过程切换到正常模式

; -----

ModeOffLoop:

```

MOV     FLCMD,#00H    ; FLCMD 寄存器清零
MOV     PFS,#00H      ; Flash 存储器状态寄存器清零
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 (设置值反转)
MOV     FLPMC,#00H    ; 经由 FLPMC 寄存器控制设置正常模式 (设置值)

```

```

BT PFS.0,$ModeOffLoop ; 检查写入特定寄存器的完成情况
; 发生错误时重复相同过程。
; 正确执行特定顺序之后,
; 自编程之前将 CPU 时钟恢复其设置

```

```

MOV     MK0,#INT_MK0  ; 恢复中断屏蔽标志

```

EI

RET

; -----

; 要写入的数据

; -----

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 实例。验证一个完整的块，使用内部校验 1。

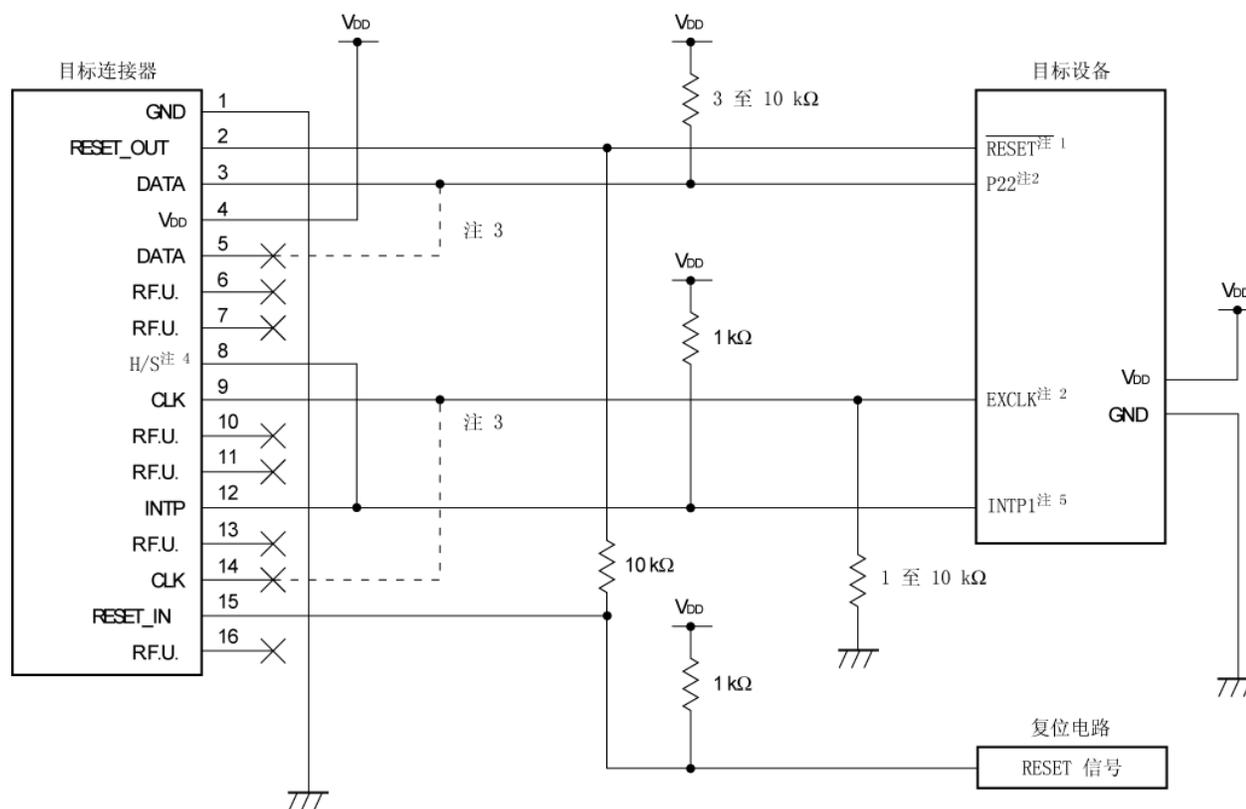
第十五章 片上调试功能

15.1 将QB-MINI2 连接到 μ PD78F9500、78F9501、78F9502

μ PD78F9500、78F9501、78F9502 使用 $\overline{\text{RESET}}$ 、EXCLK、P22、INTP1、V_{DD} 和 GND 引脚通过片上调试 仿真器(QB-MINI2)与主机进行通信。

注意事项 禁止对指定大规模生产的产品使用片上调试功能，因为使用这一功能时可能超过闪存器的保证写入次数，且因此不能保证产品的可靠性。对于使用片上调试功能时发生的问题，NEC Electronics 不承担任何责任。

图15-1. 建议的电路连接



注意事项 电路连接实例中所描述的常量仅为参考值。如果您针对大规模生产进行闪存编程，需要充分估计是否满足目标装置的规格。

- 注
1. $\overline{\text{RESET}}$ 引脚用于在调试器启动时下载监控程序或进行强制复位。因此，不能使用另外用作 $\overline{\text{RESET}}$ 引脚的引脚。有关复位引脚连接的详细信息，请参阅 **QB-MINI2 用户手册 (U18371E)**。
 2. 这是在 EXCLK 和 P22 引脚不用于目标系统时的引脚连接。当使用 EXCLK 和 P22 引脚时，请参阅 **15.1.2 EXCLK 和 P22 引脚的连接**。
 3. 连接虚线部分不会出现任何问题。
 4. 连接这个引脚以增强调试期间运行与中断之间时间测量的准确性。如果这一引脚断开，也可以进行调试，但会出现几毫秒单位的测量误差。

- 注 5. INTP1 引脚用于调试期间 QB-MINI2 与目标装置之间的通信。因此，当通过 QB-MINI2 进行调试时，不能使用 INTP1 引脚及其备用功能引脚。如需 INTP1 引脚连接的详细信息，请参阅 15.1.1 INTP1 引脚连接。

通信的引脚取决于是否写入监控程序。(请参阅表 15-1)在写入监控程序之后，EXCLK 和 P22 引脚可以用作 I/O 端口引脚或振荡引脚。

表 15-1.与 QB-MINI2 通信的引脚

写入监控程序之前	写入监控程序之后
EXCLK, P22, $\overline{\text{RESET}}$, INTP1, V_{DD} , V_{SS}	$\overline{\text{RESET}}$, INTP1, V_{DD} , V_{SS}

15.1.1 INTP1 引脚的连接

INTP1 引脚仅用于调试期间 QB-MINI2 与目标装置之间的通信。下文给出了在这些情况下适合于相关情况的设计电路。

- (1) INTP1 引脚不用于目标系统(如图 15-1 中所示。建议电路连接)
→ 请参阅图 15-2。
- (2) QB-MINI2 仅用于编程而不用于调试
→ 请参阅图 15-3。
- (3) QB-MINI2 用于调试且 INTP1 引脚的调试仅通过实机进行。
→ 请参阅图 15-4。

图15-2. INTP1 引脚不用于目标系统情况下的电路连接

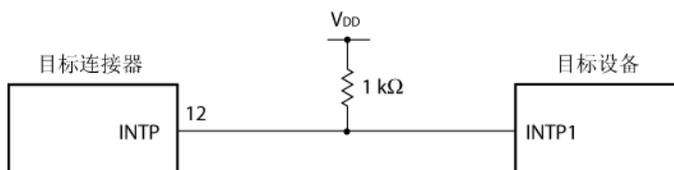
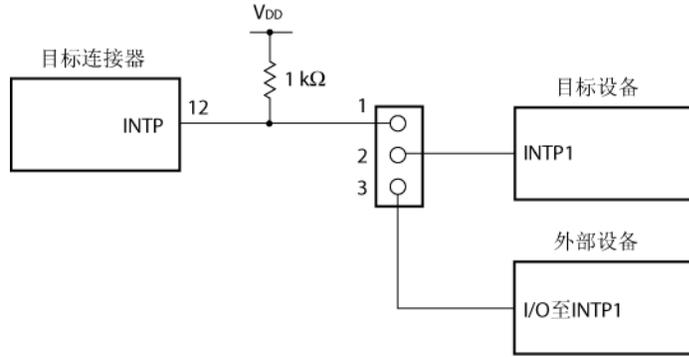


图15-3. QB-MINI2仅用于编程情况下的电路连接



图15-4. QB-MINI2 用于调试且INTP1 引脚的调试仅通过实机进行情况下的电路连接



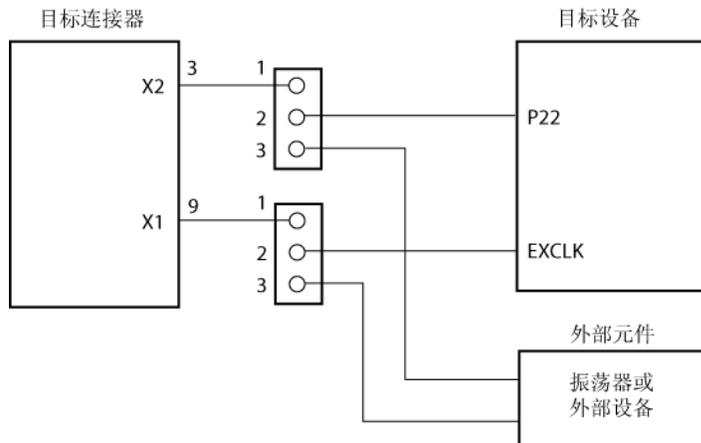
- * 跳线设置
- | | |
|--------------------|--------|
| 当用连接的 QB-MINI2 调试时 | 1-2 短路 |
| 其它情况 | 2-3 短路 |

注意事项 如果通过实机运行进行调试而不使用 QB-MINI2，则使用 QB-编程器写入用户程序。由调试器下载的程序包括监控程序，且如果监控程序不经过 QB-MINI2 控制则会出现程序故障。

15.1.2 EXCLK和P22 引脚的连接

当调试器第一次启动时且当通过 QB- 编程器编程时，使用 EXCLK 和 P22 引脚(下载监控程序时)。

图15-5. EXCLK和P22引脚用于目标系统情况下的电路连接



- * 跳线设置
- | | |
|---|--------|
| 当调试器第一次启动时且当通过 QB- 编程器编程时，使用 EXCLK 和 P22 引脚(下载监控程序时)。 | 1-2 短路 |
| 其他情况: | 2-3 短路 |

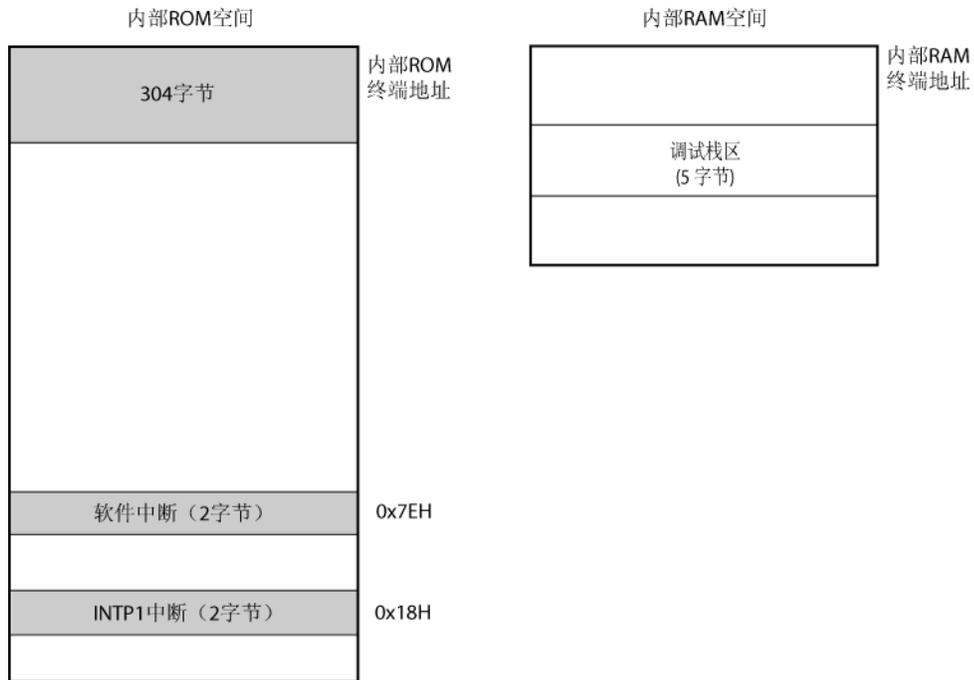
15.2 用户资源安全

用户必须准备以下设定来进行 QB-MINI2 与目标系统之间的通信并进行每项调试功能。如需设定的详细信息，请参阅 QB-MINI2 用户手册 (U18371E)。

□ 存储空间的安全

图 15-6 的阴影部分是为放入调试监控程序预留的区域，因此用户程序不能放在这些位置。

图15-6. 为调试监控程序分配的存储空间



□ 通信串口的安全

不要改变由调试监控程序执行，用于 QB-MINI2 与目标装置之间通信的 INTP1 引脚的寄存器设置。

第十六章 指令集综述

本章列出了 μ PD78F9500、78F9501、78F9502 的指令集。如需每个指令的操作和机器语言(指令码)的详细信息, 请参阅 **78K/0S 系列指令用户手册 (U11047E)**。

16.1 操作

16.1.1 操作数识别符和描述方法

根据指令操作数识别符的描述方法, 操作数在每个指令的“操作数”栏描述(有关详细信息, 请参考汇编程序规范)。当存在两种或两种以上描述方法时, 选择其中一种。大写字母和符号 #、!、\$ 及 [] 是关键字, 并对其进行描述。每个符号有以下含义。

- #: 立即数据指定
- !: 绝对地址指定
- \$: 相对地址指定
- []: 间接地址指定

在立即数据情况下, 描述适当的数值或标注。当使用标注时, 确保描述 #、!、\$ 和 [] 符号。

对于操作数寄存器识别符, r 和 rp, 功能名称 (X、A、C 等)或绝对名称(下括弧内的名称, R0、R1、R2 等) 可用于描述说明。

表 16-1. 操作数识别符和描述方法

识别符	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) 专用功能寄存器符号
saddr saddrp	FE20H 到 FF1FH 立即数据或标注 FE20H 到 FF1FH 立即数据或标注 (仅偶地址)
addr16 addr5	0000H 到 FFFFH 立即数据或标注 (即使仅有 16-位数据转移指令) 0040H 到 007FH 立即数据或标注 (仅偶地址)
word byte bit	16-位立即数据或标注 8-位立即数据或标注 3-位立即数据或标注

备注 特殊功能寄存器符号, 请参看表 3-3 特殊功能寄存器。

16.1.2 “运算”栏的描述

A:	A 寄存器; 8-位累加器
X:	X 寄存器
B:	B 寄存器
C:	C 寄存器
D:	D 寄存器
E:	E 寄存器
H:	H 寄存器
L:	L 寄存器
AX:	AX 寄存器对; 16-位累加器
BC:	BC 寄存器对
DE:	DE 寄存器对
HL:	HL 寄存器对
PC:	程序计数器
SP:	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:	中断请求使能标志
():	括号中的地址或寄存器所指的存储器单元内容
×H, ×L:	16-位寄存器的高 8 位和第 8 位
∧:	逻辑与 (AND)
∨:	逻辑或 (OR)
⊕:	逻辑异或 (异 OR)
—:	数据取反
addr16:	16-位立即数或标号
jdisp8:	带符号的 8-位数据(偏移值)

16.1.3 “标志”栏的描述

(空白):	不变
0:	清零
1:	设为 1
×:	根据结果设定/清零
R:	存储以前保存的值

16.2 操作清单

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	x x x
	A, PSW	2	4	A ← PSW	
	PSW, A	2	4	PSW ← A	x x x
	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. r = A 除外。
 2. r = A、X 除外。

备注 一个指令时钟周期为处理器时钟控制寄存器(PCC)所选择的一个 CPU 时钟周期(fCPU)。

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 仅当 rp = BC、DE 或 HL 时。

备注 一个指令时钟周期为处理器时钟控制寄存器(PCC)所选择的一个 CPU 时钟周期(fCPU)。

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	$(\text{saddr}), CY \leftarrow (\text{saddr}) - \text{byte} - CY$	×	×	×
	A, r	2	4	$A, CY \leftarrow A - r - CY$	×	×	×
	A, saddr	2	4	$A, CY \leftarrow A - (\text{saddr}) - CY$	×	×	×
	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	$(\text{saddr}) \leftarrow (\text{saddr}) \wedge \text{byte}$	×		
	A, r	2	4	$A \leftarrow A \wedge r$	×		
	A, saddr	2	4	$A \leftarrow A \wedge (\text{saddr})$	×		
	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	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	×		
	A, r	2	4	$A \leftarrow A \vee r$	×		
	A, saddr	2	4	$A \leftarrow A \vee (\text{saddr})$	×		
	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	$A \leftarrow A \vee \text{byte}$	×		
	saddr, #byte	3	6	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	×		
	A, r	2	4	$A \leftarrow A \vee r$	×		
	A, saddr	2	4	$A \leftarrow A \vee (\text{saddr})$	×		
	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	$A \leftarrow A \vee (\text{HL} + \text{byte})$	×		

备注 一个指令时钟周期为处理器时钟控制寄存器(PCC)所选择的一个 CPU 时钟周期(fCPU)。

助记符	操作数	字节	时钟	操作	标志		
					Z	AC	CY
CMP	A, #byte	2	4	A – byte	×	×	×
	saddr, #byte	3	6	(saddr) – byte	×	×	×
	A, r	2	4	A – r	×	×	×
	A, saddr	2	4	A – (saddr)	×	×	×
	A, !addr16	3	8	A – (addr16)	×	×	×
	A, [HL]	1	6	A – (HL)	×	×	×
	A, [HL + byte]	2	6	A – (HL + byte)	×	×	×
ADDW	AX, #word	3	6	AX, CY ← AX + word	×	×	×
SUBW	AX, #word	3	6	AX, CY ← AX – word	×	×	×
CMPW	AX, #word	3	6	AX – word	×	×	×
INC	r	2	4	r ← r + 1	×	×	
	saddr	2	4	(saddr) ← (saddr) + 1	×	×	
DEC	r	2	4	r ← r – 1	×	×	
	saddr	2	4	(saddr) ← (saddr) – 1	×	×	
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			×
ROL	A, 1	1	2	(CY, A ₀ ← A ₇ , A _{m+1} ← A _m) × 1			×
RORC	A, 1	1	2	(CY ← A ₀ , A ₇ ← CY, A _{m-1} ← A _m) × 1			×
ROLC	A, 1	1	2	(CY ← A ₇ , A ₀ ← CY, A _{m+1} ← A _m) × 1			×
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	×	×	×
	[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	×	×	×
	[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}			×

备注 一个指令时钟周期为处理器时钟控制寄存器(PCC)所选择的一个 CPU 时钟周期(fCPU)。

助记符	操作数	字节	时钟	运算	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, \$saddr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ 如果 (saddr.bit) = 1			
	sfr.bit, \$saddr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ 如果 sfr.bit = 1			
	A.bit, \$saddr16	3	8	$PC \leftarrow PC + 3 + \text{jdisp8}$ 如果 A.bit = 1			
	PSW.bit, \$saddr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ 如果 PSW.bit = 1			
BF	saddr.bit, \$saddr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ 如果 (saddr.bit) = 0			
	sfr.bit, \$saddr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ 如果 sfr.bit = 0			
	A.bit, \$saddr16	3	8	$PC \leftarrow PC + 3 + \text{jdisp8}$ 如果 A.bit = 0			
	PSW.bit, \$saddr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ 如果 PSW.bit = 0			
DBNZ	B, \$saddr16	2	6	$B \leftarrow B - 1$, 如果 $B \neq 0$ 那么 $PC \leftarrow PC + 2 + \text{jdisp8}$			
	C, \$saddr16	2	6	$C \leftarrow C - 1$, 如果 $C \neq 0$ 那么 $PC \leftarrow PC + 2 + \text{jdisp8}$			
	saddr, \$saddr16	3	8	$(\text{saddr}) \leftarrow (\text{saddr}) - 1$, 如果 $(\text{saddr}) \neq 0$, 那么 $PC \leftarrow PC + 3 + \text{jdisp8}$			
NOP		1	2	无操作			
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 一个指令时钟周期为处理器时钟控制寄存器(PCC)所选择的一个 CPU 时钟周期(fCPU)。

16.3 按地址类型所列指令

(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 2 操作数 第 1 操作数	#byte	A	r	sfr	saddr	laddr16	PSW	[DE]	[HL]	[HL + byte]	\$addr16	1	None
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 r = A 除外。

(2) 16-位指令

MOVW、XCHW、ADDW、SUBW、CMPW、PUSH、POP、INCW、DECW

第 1 操作数 \ 第 2 操作数	#word	AX	rp [#]	saddrp	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	
rp	MOVW	MOVW [#]				INCW DECW PUSH POP
saddrp		MOVW				
sp		MOVW				

注 仅当 rp = BC、DE 或 HL 时。

(3) 位操作指令

SET1、CLR1、NOT1、BT、BF

第 1 操作数 \ 第 2 操作数	\$addr16	None
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) Call 指令/转移 指令

CALL、CALLT、BR、BC、BNC、BZ、BNZ、DBNZ

第 1 操作数 \ 第 2 操作数	AX	!addr16	[addr5]	\$addr16
基本 指令	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
复合指令				DBNZ

(5) 其他指令

RET、RETI、NOP、EI、DI、HALT、STOP

第十七章 电气特性

最大绝对额定值 ($T_A = 25^\circ\text{C}$)

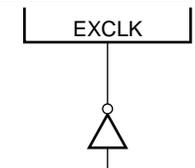
参数	符号	条件	额定值	单位
电源电压	V_{DD}		-0.3 到+6.5	V
	V_{SS}		-0.3 到+0.3	V
输入电压	V_I	P20 到 P23、P32、P34、P40、P43	-0.3 到 $V_{DD} + 0.3$ ^注	V
输出电压	V_O		-0.3 到 $V_{DD} + 0.3$ ^注	V
模拟输入电压	V_{AN}		-0.3 到 $V_{DD} + 0.3$ ^注	V
输出电流, 高	I_{OH}	每个引脚	-10.0	mA
		P20 到 P23、P32、P40、P43 的总和	-44.0	mA
输出电流, 低	I_{OL}	每个引脚	20.0	mA
		P20 到 P23、P32、P40、P43 的总和	44.0	mA
运行环境温度	T_A	在正常工作模式下	-40 到+85	°C
		闪存编程期间		
存储温度	T_{stg}	闪存空白状态	-65 到+150	°C
		已进行闪存编程	-40 到+125	°C

注 必须是 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，最大绝对额定值是产品受到物理损坏的边界额定值，因此，必须在确保不超过最大绝对额定值的条件下使用产品。

备注 除非另行指定，备用功能引脚的特性与端口引脚功能相同。

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

谐振器	推荐电路	参数	条件	最小值	典型值	最大值	单位
外部时钟		外部主系统时钟频率 (f_{EXCLK}) ^{#2}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.0		10.0	MHz
			$2.0\text{ V} \leq V_{DD} < 2.7\text{ V}$	2.0		5.0	
		外部主系统时钟输入高-/低-电平宽度 (t_{EXCLKH} , t_{EXCLKL})	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0.045		0.25	s
			$2.0\text{ V} \leq V_{DD} < 2.7\text{ V}$	0.09		0.25	

- 注 1. 请在 2.2 到 5.5 V 电压范围内使用本产品，因为上电清零电路(POC)的检测电压(V_{POC})是 $2.1\text{ V} \pm 0.1\text{ V}$ 。
2. 仅给出了振荡器特性。有关指令执行时间的详细信息，请参看 **AC 特性**。

备注 有关谐振器选择和振荡器常量，用户需要自己评估振荡器或要求制造商评估测试。

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

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	振荡频率 (f_x) ^{#2}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$T_A = -10$ 到 $+85^\circ\text{C}$			± 2	%
			$T_A = -40$ 到 $+85^\circ\text{C}$			± 5	%
		$2.0\text{ V} \leq V_{DD} < 2.7\text{ V}$	5.5				MHz

- 注 1. 请在 2.2 到 5.5 V 电压范围内使用本产品，因为上电清零电路(POC)的检测电压(V_{POC})是 $2.1\text{ V} \pm 0.1\text{ V}$ 。
2. 仅给出了振荡器特性。有关指令执行时间的详细信息，请参看 **AC 特性**。

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

谐振器	参数	条件	MIN.	TYP.	MAX.	单位
低速内部振荡器	振荡频率 (f_{RL})		120	240	480	kHz

- 注 请在 2.2 到 5.5 V 电压范围内使用本产品，因为上电清零电路(POC)的检测电压(V_{POC})是 $2.1\text{ V} \pm 0.1\text{ V}$ 。

DC 特性 ($T_A = -40$ 到 $+85^\circ\text{C}$, $V_{DD} = 2.0$ 到 5.5 V ^注, $V_{SS} = 0\text{ V}$) (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
		所有引脚的总和	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-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
		所有引脚的总和	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			30	mA
			$2.0\text{ V} \leq V_{DD} < 4.0\text{ V}$			15	mA
<R> 输入电压, 高	V_{IH1}			$0.8V_{DD}$		V_{DD}	V
<R> 输入电压, 低	V_{IL1}			0		$0.2V_{DD}$	V
输出电压, 高	V_{OH}	输出引脚的总和 $I_{OH} = -15\text{ mA}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ $I_{OH} = -5\text{ mA}$	$V_{DD} - 1.0$			V
		$I_{OH} = -100\ \mu\text{A}$	$2.0\text{ V} \leq V_{DD} < 4.0\text{ V}$	$V_{DD} - 0.5$			V
输出电压, 低	V_{OL}	输出引脚的总和 $I_{OL} = 30\text{ mA}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ $I_{OL} = 10\text{ mA}$			1.3	V
		$2.0\text{ V} \leq V_{DD} < 4.0\text{ V}$ $I_{OL} = 400\ \mu\text{A}$				0.4	V
输入漏电流, 高	I_{LH}	$V_i = V_{DD}$	EXCLK 以外的引脚			1	μA
输入漏电流, 低	I_{LIL}	$V_i = 0\text{ V}$	EXCLK 以外的引脚			-1	μA
输出漏电流, 高	I_{LOH}	$V_o = V_{DD}$	EXCLK 以外的引脚			1	μA
输出漏电流, 低	I_{LOL}	$V_o = 0\text{ V}$	EXCLK 以外的引脚			-1	μA
上拉电阻值	R_{PU}	$V_i = 0\text{ V}$		10	30	100	k Ω
		$V_i = 0\text{ V}$ (P34, 复位状态)		10	30	100	k Ω

注 请在 2.2 到 5.5 V 电压范围内使用本产品, 因为上电清零电路(POC)的检测电压(V_{POC})是 $2.1\text{ V} \pm 0.1\text{ V}$ 。

备注 除非另行指定, 备用功能引脚的特性与端口引脚功能相同。

DC 特性 ($T_A = -40$ 到 $+85^\circ\text{C}$, $V_{DD} = 2.0$ 到 5.5 V ^{注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	
电源电流 ^{注2}	I _{DD1} ^{注3}	外部时钟输入振荡工作模式 ^{注6}	f _X = 10 MHz V _{DD} = 5.0 V ±10% ^{注4}		6.1	12.2	mA	
			f _X = 6 MHz V _{DD} = 5.0 V ±10% ^{注4}		5.5	11.0	mA	
			f _X = 5 MHz V _{DD} = 3.0 V ±10% ^{注5}		3.0	6.0	mA	
	I _{DD2}	外部时钟输入 HALT 模式 ^{注6}	f _X = 10 MHz V _{DD} = 5.0 V ±10% ^{注4}	停止外围功能时		1.7	3.8	mA
				运行外围功能时			6.7	
			f _X = 6 MHz V _{DD} = 5.0 V ±10% ^{注4}	停止外围功能时		1.3	3.0	mA
				运行外围功能时			6.0	
			f _X = 5 MHz V _{DD} = 3.0 V ±10% ^{注5}	停止外围功能时		0.48	1	mA
				运行外围功能时			2.1	
	I _{DD3} ^{注3}	高速内部振荡工作模式 ^{注7}	f _X = 8 MHz V _{DD} = 5.0 V ±10% ^{注4}		5.0	10.0	mA	
	I _{DD4}	高速内部振荡工作模式 HALT 模式 ^{注7}	f _X = 8 MHz V _{DD} = 5.0 V ±10% ^{注4}	停止外围功能时		1.4	3.2	mA
				运行外围功能时			5.9	
I _{DD5}	STOP 模式	V _{DD} = 5.0 V ±10%	停止低速内部振荡工作模式时		3.5	20.0	μA	
			运行低速内部振荡工作模式时		17.5	32.0		
		V _{DD} = 3.0 V ±10%	停止低速内部振荡工作模式时		3.5	15.5	μA	
			运行低速内部振荡工作模式时		11.0	26.0		

- 注
- 请在 2.2 到 5.5 V 电压范围内使用本产品，因为上电清零电路(POC)的检测电压(V_{POC})是 2.1 V ±0.1 V。
 - 总电流流经内部电源(V_{DD})。然而，不包括流经端口上拉电阻的电流。
 - I_{DD1} 和 I_{DD3} 包括外围工作电流。
 - 当处理器时钟控制寄存器(PCC)设定为 00H 时。
 - 当处理器时钟控制寄存器(PCC)设定为 02H 时。
 - 当使用选项字节选择外部时钟输入作为系统时钟源时。
 - 当使用选项字节选择高速内部振荡工作模式时钟作为系统时钟源时。

AC 特性

基本工作情况($T_A = -40$ 到 $+85^\circ\text{C}$, $V_{DD} = 2.0$ 到 $5.5\text{ V}^{\#}$, $V_{SS} = 0\text{ V}$)

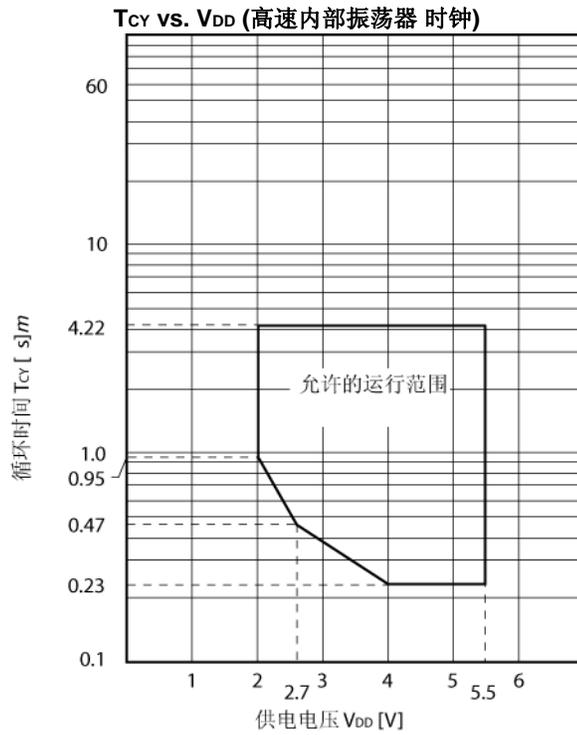
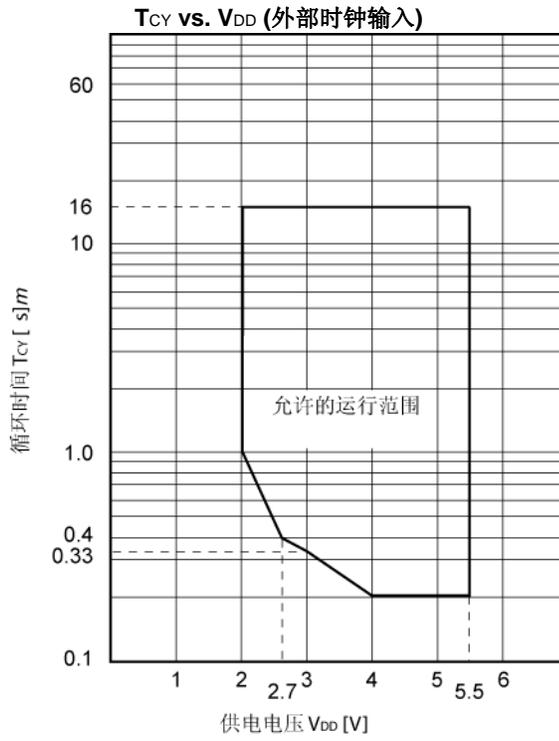
参数	符号	条件	最小值	典型值	最大值	单位	
周期 (最小指令执行时间)	T_{CY}	外部时钟输入	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0.2		16	μs
			$3.0\text{ V} \leq V_{DD} < 4.0\text{ V}$	0.33		16	μs
			$2.7\text{ V} \leq V_{DD} < 3.0\text{ V}$	0.4		16	μs
			$2.0\text{ V} \leq V_{DD} < 2.7\text{ V}$	1		16	μs
		高速内部振荡工作模式时钟	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0.23		4.22	μs
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	0.47		4.22	μs
$2.0\text{ V} \leq V_{DD} < 2.7\text{ V}$	0.95			4.22	μs		
中断输入高电平宽度, 低电平宽度	t_{INTH} , t_{INTL}		1			μs	
$\overline{\text{RESET}}$ 输入低电平宽度	t_{RSL}		2			μs	

注 请在 2.2 到 5.5 V 电压范围内使用本产品, 因为上电清零电路(POC)的检测电压(V_{POC})是 $2.1\text{ V} \pm 0.1\text{ V}$ 。

CPU 时钟频率、外围时钟频率

参数	条件	CPU 时钟(f_{CPU})	外围时钟(f_{XP})
外部时钟	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$125\text{ kHz} \leq f_{CPU} \leq 10\text{ MHz}$	$500\text{ kHz} \leq f_{XP} \leq 10\text{ MHz}$
	$3.0\text{ V} \leq V_{DD} < 4.0\text{ V}$	$125\text{ kHz} \leq f_{CPU} \leq 6\text{ MHz}$	
	$2.7\text{ V} \leq V_{DD} < 3.0\text{ V}$	$125\text{ kHz} \leq f_{CPU} \leq 5\text{ MHz}$	
	$2.0\text{ V} \leq V_{DD} < 2.7\text{ V}^{\#}$	$125\text{ kHz} \leq f_{CPU} \leq 2\text{ MHz}$	$500\text{ kHz} \leq f_{XP} \leq 5\text{ MHz}$
高速内部振荡器	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$500\text{ kHz (TYP.)} \leq f_{CPU} \leq 8\text{ MHz (TYP.)}$	$2\text{ MHz (TYP.)} \leq f_{XP} \leq 8\text{ MHz (TYP.)}$
	$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	$500\text{ kHz (TYP.)} \leq f_{CPU} \leq 4\text{ MHz (TYP.)}$	
	$2.0\text{ V} \leq V_{DD} < 2.7\text{ V}^{\#}$	$500\text{ kHz (TYP.)} \leq f_{CPU} \leq 2\text{ MHz (TYP.)}$	$2\text{ MHz (TYP.)} \leq f_{XP} \leq 4\text{ MHz (TYP.)}$

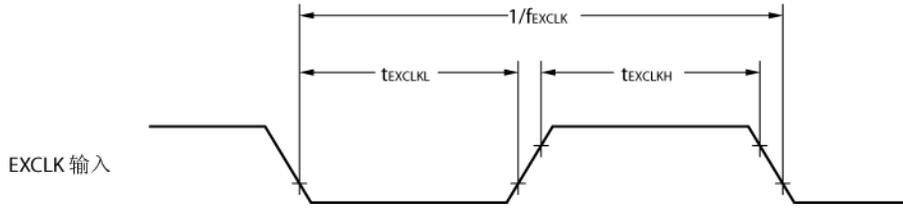
注 请在 2.2 到 5.5 V 电压范围内使用本产品, 因为上电清零电路(POC)的检测电压(V_{POC})是 $2.1\text{ V} \pm 0.1\text{ V}$ 。



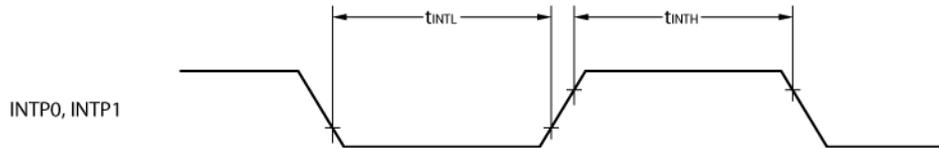
AC 时序测试点 (不包括 EXCLK 输入)



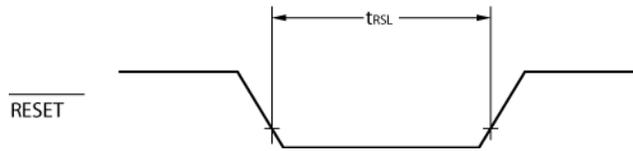
时钟时序



中断输入时序



RESET 输入时序

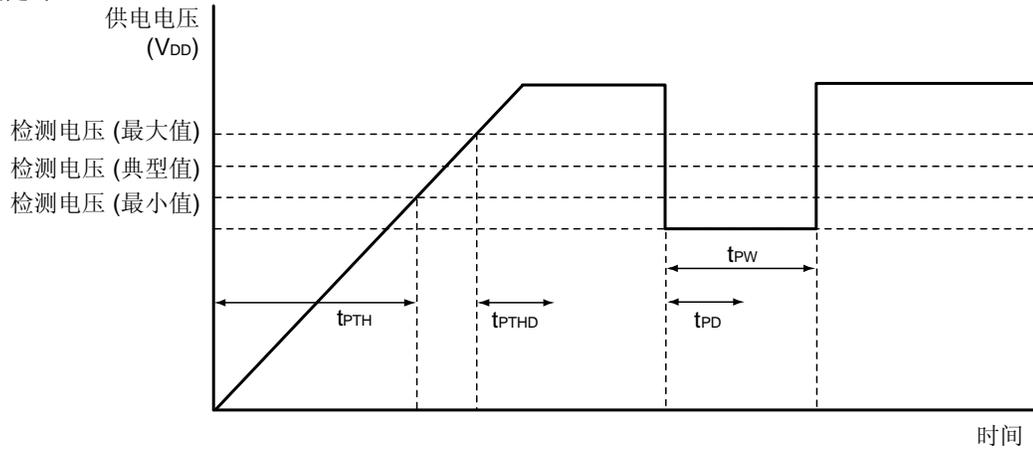


POC 电路特性 ($T_A = -40$ 到 $+85^\circ\text{C}$)

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V_{POC}		2.0	2.1	2.2	V
电源电压上升时间	t_{PTH}	$V_{\text{DD}}: 0\text{ V} \rightarrow 2.1\text{ V}$	1.5			μs
响应延迟时间 1 ^{注1}	t_{PTHD}	电源电压上升时, 在达到检测电压(MAX.)之后			3.0	ms
响应延迟时间 2 ^{注2}	t_{PD}	电源电压下降时			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所需时间。
 2. 从电压检测到内部复位信号产生所需时间。

POC 电路定时



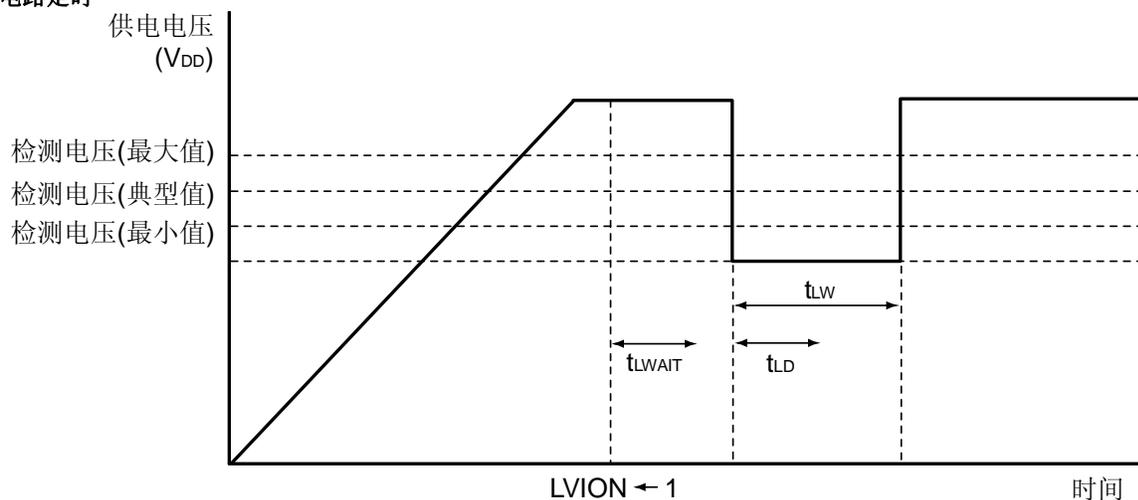
LVI 电路特性 ($T_A = -40$ 到 $+85^\circ\text{C}$)

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V_{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
最短脉冲宽度	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所需时间。
 2. 从将 LVION 设为 1 到工作稳定所需时间。

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

LVI 电路定时

数据存储器 STOP 模式低电源电压数据保持特性 ($T_A = -40$ 到 $+85^\circ\text{C}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
数据保留电源电压	V_{DDDR}		2.0		5.5	V
释放信号设定时间	t_{SREL}		0			μs

闪存编程特性 ($T_A = -40$ 到 $+85^\circ\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$)

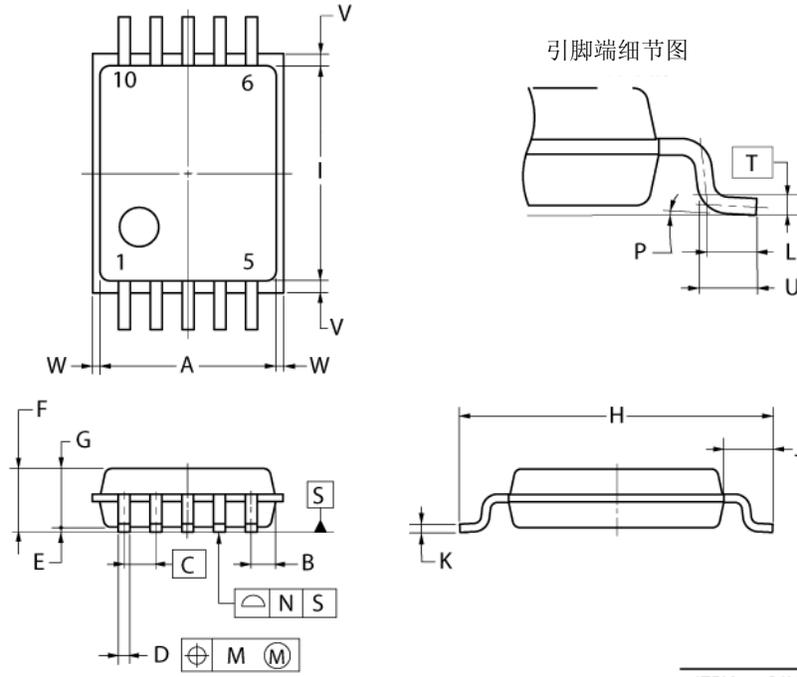
参数	符号	条件		MIN.	TYP.	MAX.	单位
电源电流	I_{DD}	$V_{DD} = 5.5\text{ V}$				7.0	mA
擦除计数 ^{‡1} (每 1 块)	N_{ERASE}	$T_A = -40$ 到 $+85^\circ\text{C}$		1000			Times
芯片擦除时间	T_{CERASE}	$T_A = -10$ 到 $+85^\circ\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			0.8	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$			1.0	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$			1.2	s
		$T_A = -10$ 到 $+85^\circ\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			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
		$T_A = -40$ 到 $+85^\circ\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			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
		$T_A = -40$ 到 $+85^\circ\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			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}	$T_A = -10$ 到 $+85^\circ\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			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
		$T_A = -10$ 到 $+85^\circ\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			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
		$T_A = -40$ 到 $+85^\circ\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			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
		$T_A = -40$ 到 $+85^\circ\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			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}	$T_A = -40$ 到 $+85^\circ\text{C}$, $N_{ERASE} \leq 1000$				150	μs
内部校验	T_{VERIFY}	每 1 块				6.8	ms
		每 1 字节				27	μs
空白检测	T_{BLKCHK}	每 1 块				480	μs
保留年		$T_A = 85^\circ\text{C}$ ^{‡2} , $N_{ERASE} \leq 1000$		10			年

- 注 1. 取决于擦除计数(N_{ERASE})，擦除时间有所不同。请参看芯片擦除时间和 Block 擦除时间参数。
2. 当工作和不工作时平均温度为 85°C 时。

备注 在产品出货后初次写入时，“擦除 → 写入”和“只写”都认为是一次重写。

第十八章 封装图

10 引脚塑封 SSOP 封装(5.72mm(225))



注
在最大使用材料情况下，每条引线的中心线位于其实际位置(T.P.)的0.13mm内

(UNI T:mm)	
ITEM	DIMENSIONS
A	3.60 ±0.10
B	0.50
C	0.65 (T.P.)
D	0.24 ±0.08
E	0.10 ±0.05
F	1.45 MAX .
G	1.20 ±0.10
H	6.40 ±0.20
I	4.40 ±0.10
J	1.00 ±0.20
K	0.17 ^{+0.08} _{-0.07}
L	0.50
M	0.13
N	0.10
P	3° ^{+5°} _{-3°}
T	0.25 (T.P.)
U	0.60 ±0.15
V	0.25 MAX .
W	0.15 MAX .

P10MA-65-C AC

第十九章 建议焊接条件

这些产品应在以下建议条件下焊接和安装。

有关技术信息，请参看以下网站。

半导体装置安装手册 (<http://www.necel.com/pkg/en/mount/index.html>)

- 注意事项**
1. 产品的器件编号尾缀带有 **-A**，是无铅产品。
 2. 如需以下建议接方法和条件之外的焊接方法和条件，请联系日电电子 销售代表。

表 19-1. 表面安装类型焊接条件

• **10-引脚塑料 SSOP (无铅产品)**

μ PD78F9500MA-CAC-A, 78F9501MA-CAC-A, 78F9502MA-CAC-A

焊接方法	焊接条件	建议条件符号
红外回流	包装最高温度: 260°C, 时间: 60 秒 max. (220°C 或更高), 计数: 3 次或 3 次以下, 暴晒期限: 7 天 ^注 (此后, 在 125°C 下预烘干 10 到 72 小时)	IR60-107-3
波峰焊	如需详细信息, 请联系 NEC Electronics 销售代表。	-
局部加热	引脚温度: 350°C max., 时间: 3 秒 max. (每排引脚)	-

注 打开干包装之后, 在允许的保存期内将其保存在 25°C 以下和 65% RH 或更低湿度的环境下。

注意事项 禁止同时使用不同焊接方法(局部加热除外)。

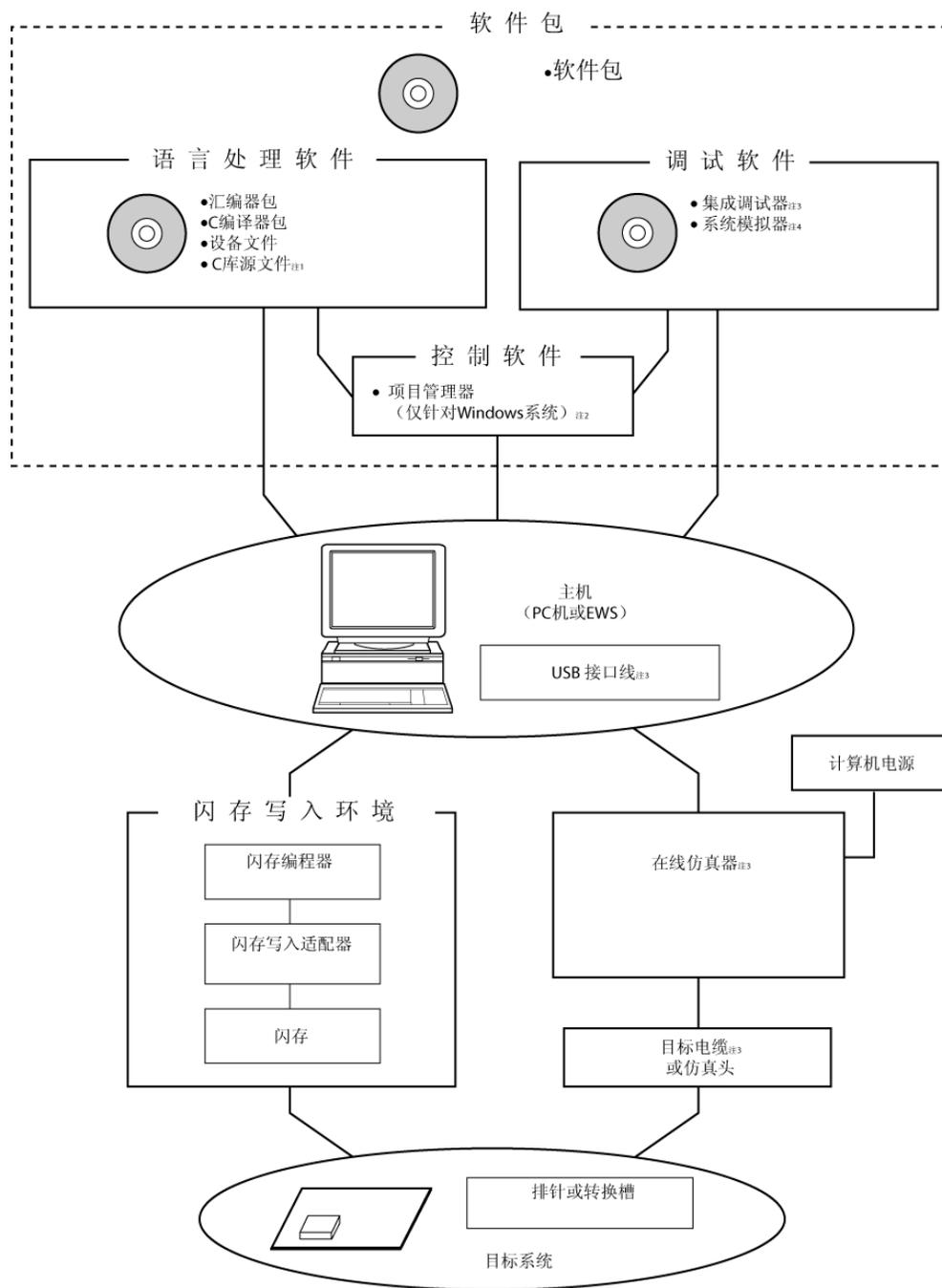
附录A 开发工具

以下开发工具可用于 μ PD78F9500, 78F9501, 78F9502 的应用系统开发。开发工具如图 A-1 所示。

- 与 PC98-NX 系列计算机兼容
除非特别说明, IBM PC/AT™ 兼容机支持的产品与 PC98-NX 系列计算机是兼容的。当使用 PC98-NX 系列计算机时, 请参阅 IBM PC/AT 及其兼容机的使用说明。

图 A-1. 开发工具 (1/2)

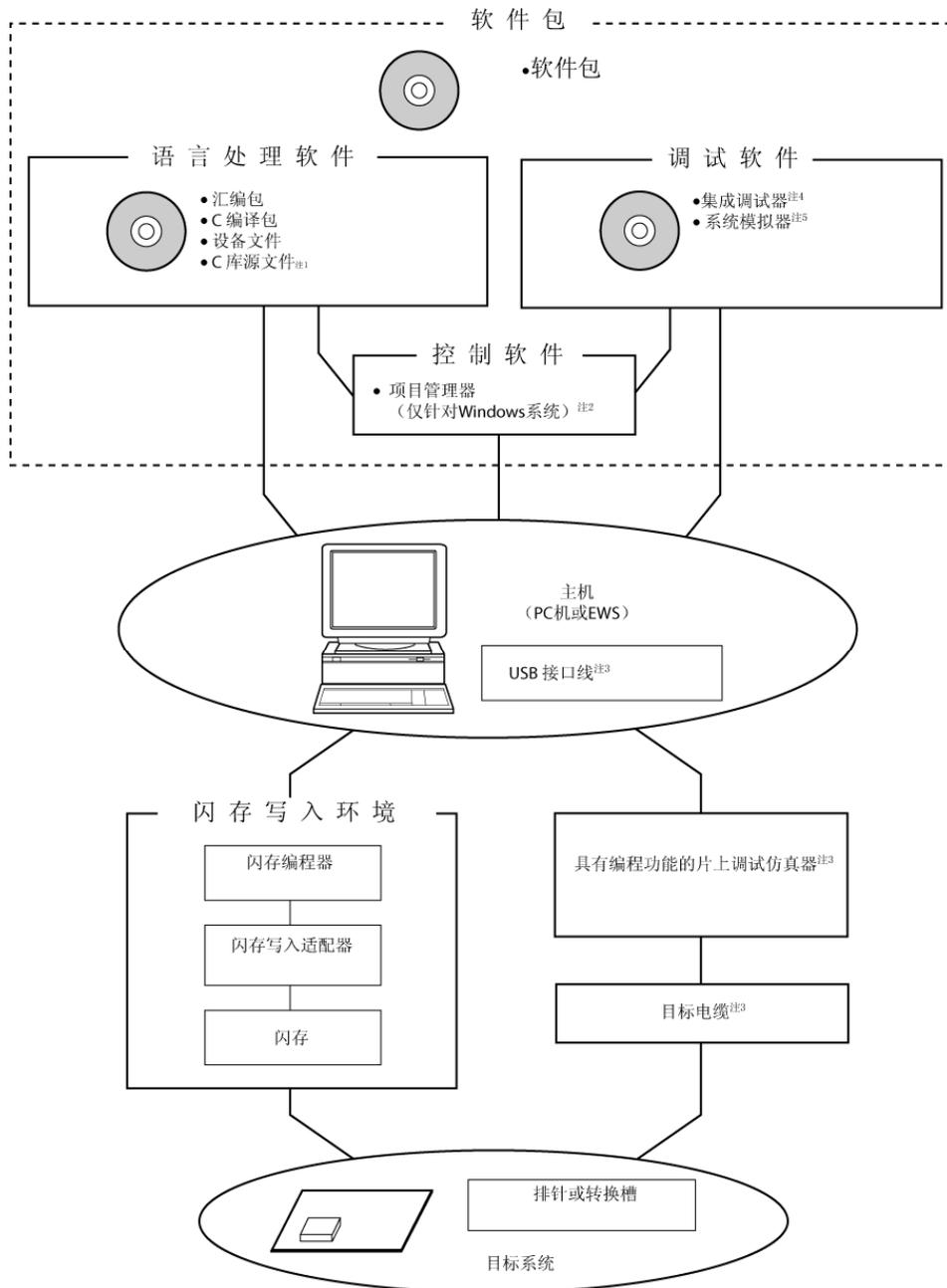
(1) 当使用在线仿真器 QB-78K0SKX1 时



- 注
1. 该软件包不包含 C 语言源程序库。
 2. 汇编程序包中包含有项目管理器 PM+。
PM+只能在 Windows™ 操作系统环境下使用。
 3. 在线仿真器 QB-78K0SKX1 随集成调试器 ID78K0S-QB，带有编程功能 QB-MINI2 的内置调试仿真器，一根 USB 接口电缆，一个电源及一根目标电缆等一起供货，其他配件为选购件。
 4. 用于 μ PD78F9500, 78F9501, 78F9502 的系统模拟器正在开发中。

图 A-1. 开发工具 (2/2)

(2) 当使用带编程功能 QB-MINI2 的内置调试仿真器时



- 注
1. 该软件包不包含 C 语言源程序库。
 2. 汇编程序包中包含有项目管理器 PM+。
PM+只能在 Windows™ 操作系统环境下使用。
 3. 提供带编程功能 QB-MINI2 的内置调试仿真器，并配有一根 USB 接口电缆和一根目标电缆。
 4. QB-MINI2 中不包含集成调试器 ID78K0S-QB。
集成调试器 ID78K0S-QB 可从如下网站获取。
<http://www.necel.com/micro/ods/eng/>
 5. 用于 μ PD78F9500, 78F9501, 78F9502 的系统模拟器正在开发中。

A.1 软件包

SP78K0S 软件包	该软件包包含了 78K0S 系列应用系统开发所需要的一些工具软件。 包含如下一些工具。 RA78K0S, CC78K0S, ID78K0S-NS, 用于 78K0S/Kx1+的 SM+ ^{注1} , SM78K0S ^{注2} , 及设备文件 ^{注3}
	产品号: $\mu S_{xxxx}SP78K0S$

- 注
1. SP78K0S 2.00 版或早期版本软件包不包含用于 78K0S/Kx1+的 SM+。
 2. SM78K0S 不支持 78K0S/Kx1+。
 3. SP78K0S 2.00 版或早期版本软件包不包含 DF789234。

备注 产品号中的xxxx随使用的 OS 的不同而有所不同。

$\mu S_{xxxx}SP78K0S$

xxxx	主机	OS	存储介质
AB17	PC-9800 系列, IBM PC/AT 及其兼容机	Windows (日文版)	CD-ROM
BB17		Windows (英文版)	

A.2 编程语言处理软件

RA78K0S 汇编程序包	将由助记符编写的程序转换成微控制器能够执行的目标代码。 此外, 还提供符号表自动生成功能及最优分支转移指令, 结合设备文件(DF789234) (单独销售)一起使用。 <在 PC 环境下使用时应该注意的注意事项> 该汇编程序包是基于 DOS 应用程序的, 但是通过使用 Windows 的 PM+(包含在该软件包中), 也可以在 Windows 环境下使用该软件包。 产品号: $\mu S_{xxxx}RA78K0S$
CC78K0S C 语言库程序包	将由 C 语言编写的程序转换成微控制器能够执行的目标代码。 该 C 语言库程序包要结合汇编软件包(RA78K0S)及设备文件(DF789234)一起使用 (这两种程序包均独立销售)。 <在 PC 环境下使用时应该注意的注意事项> 该 C 语言编译器软件包是基于 DOS 应用程序的, 但是通过使用 Windows 的 PM+(包含在该软件包中), 也可以在 Windows 环境下使用该软件包。 产品号: $\mu S_{xxxx}CC78K0S$
DF789234 ^{注1} 设备文件	该文件包含设备特定信息。 该设备文件应当结合其他一些工具(RA78K0S, CC78K0S, ID78K0S-QB, 或用于 78K0S/Kx1+的 SM+)一起使用。 (所有这些软件工具包均独立销售)。 产品号: $\mu S_{xxxx}DF789234$
CC78K0S-L ^{注2} C 语言库源程序文件	组成目标库的函数源程序文件包含在 C 编译器软件包中。 可以根据用户的特定需求对编译器软件包中的目标库进行必要的更改。 因为这是一个源程序文件, 其工作环境不依赖于任何特定的操作系统。 产品号: $\mu S_{xxxx}CC78K0S-L$

- 注 1. DF789234 为一公共文件，可以用于 RA78K0S，CC78K0S，ID78K0S-QB 以及用于 78K0S/Kx1+的 SM+。
2. 该软件包(SP78K0S)不包含 CC78K0S-L。

备注 产品号中的xxxx随主机和使用的 OS 的不同而有所不同。

μ SxxxxRA78K0S

μ SxxxxCC78K0S

μ SxxxxCC78K0S-L

xxxx	主机	OS	存储介质
AB17	PC-9800 系列, IBM PC/AT 及其兼容机	Windows (日文版)	CD-ROM
BB17		Windows (英文版)	
3P17	HP9000 系列 700™	HP-UX™ (Rel.10.10)	
3K17	SPARC 站™	SunOS™ (Rel.4.1.4) Solaris™ (Rel.2.5.1)	

μ SxxxxDF789234

xxxx	主机	OS	存储介质
AB13	PC-9800 系列, IBM PC/AT 及其兼容机	Windows (日文版)	3.5" 2HD FD
BB13		Windows (英文版)	

A.3 控制软件

PM+ 项目管理器	<p>这是一款为在 Windows 环境下有效地开发用户程序而设计的控制软件，使用该软件，用户程序开发所进行的一系列操作，如启动编辑器、构建程序和启动调试器，都可以由项目管理器 PM+ 执行。</p> <p><注意事项> PM+ 包含在汇编程序软件包 (RA78K0S) 中，能在 Windows 环境中使用。</p>
--------------	--

A.4 闪存写入工具

<R> FlashPro4 (FL-PR4, PG-FP4) FlashPro5 (FL-PR5, PG-FP5) 闪存编程器	闪存编程器专门为具有内置闪存的微控制器设计使用
QB-MINI2 带有编程功能的内置调试仿真器	这是一款专门为具有内置闪存的微控制器设计使用的闪存编程器。同时，在使用所有带闪存的微控制器(包括 78K0S/Kx1+)进行应用程序系统开发时，该带有编程功能的内置调试仿真器也可用作内置调试仿真器用于调试硬件或软件。
FA-78F9202MA-CAC-MX 闪存写入适配器	闪存写入适配器。要与闪存编程器一起使用。

备注 FL-PR4，FL-PR5，及 FA-78F9202MA-CAC-MX 为 Naito Densei Machida Mfg. Co., Ltd 产品。
要了解有关该产品更多信息，请联系：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 时

QB-78K0SKX1 在线仿真器	在使用 78K0S/Kx1+产品开发应用系统时，该在线仿真器用于调试硬件和软件。它支持集成调试器(ID78K0S-QB)。该仿真器应当与 AC 适配器、标系统电缆、以及通过 USB 接口电缆连接的主机一起使用。
QB-50-EP-01T 仿真探针	这是一款灵活型仿真探针，用于连接在线仿真器与目标系统。
QB-10MA-EA-01T 交换适配器	该交换适配器用于执行由仿真器引脚类型到目标连接器引脚类型的转换。
QB-10MA-NQ-01T 目标系统连接器	该目标连接器用于安装目标系统。
目标系统上的排针规格	0.635 mm × 0.635 mm (高度: 6 mm)

A.5.2 当使用内置调试仿真器QB-MINI2 时

QB-MINI2 带有编程功能的内置调试仿真器	在应用所有带闪存的微控制器(包括 78K0S/Kx1+)进行应用程序系统开发时，使用该带有编程功能的内置调试仿真器进行调试硬件或软件。该带有编程功能的内置调试仿真器也可专门用作闪存编程器，专门用于具有内置闪存的微控制器的应用系统开发。
目标系统上的排针规格	16 引脚通用连接器 (排针脚距 2.54 mm)

A.6 调试工具 (软件)

ID78K0S-QB (支持 QB-78K0SKX1 及 QB-MINI2) 集成调试器 (配件)	该调试器支持用于 78K0S/Kx1+系列的在线仿真器。ID78K0S-QB 是一款基于 Windows 的软件。由于提供了支持 C 语言的调试功能, 使得源程序编写, 分开显示, 以及存储器显示等成为可能。该集成调试器应当与设备文件(DF789234) (单独销售)一起使用。该集成调试器提供在线仿真器 QB-78K0SKX1。
用于 78K0S/Kx1+ ^{注1} 的 SM+系统模拟器	这是一款用于 78K/0S 系列的系统模拟器。用于 78K0S/Kx1+ 的 SM+是一款基于 Windows 的软件。在模拟主机上目标系统操作的同时, 该模拟器可以对 C 源程序或汇编文件进行调试。通过使用用于 78K0S/Kx1+ 的 SM+, 可以在不依赖硬件开发的基础上进行应用程序逻辑测试及应用程序性能测试, 从而提高的开发效率和软件质量。 该 SM+应当结合设备文件(DF789234) (单独销售)一起使用。
DF789234 ^{注2} 设备文件	这是一个含有设备特定信息的文件。 该设备文件应当结合其他一些工具(RA78K0S, CC78K0S, ID78K0S-QB, 或用于 78K0S/Kx1+的 SM+)一起使用。(所有这些软件工具包均独立销售)。
	产品号: μ SxxxxDF789234

- 注
1. 用于 μ PD78F9500, 78F9501, 78F9502 的系统模拟器正在开发中。
 2. DF789234 为一公共文件, 可以用于 RA78K0S, CC78K0S, ID78K0S-QB 以及用于 78K0S/Kx1+的 SM+。

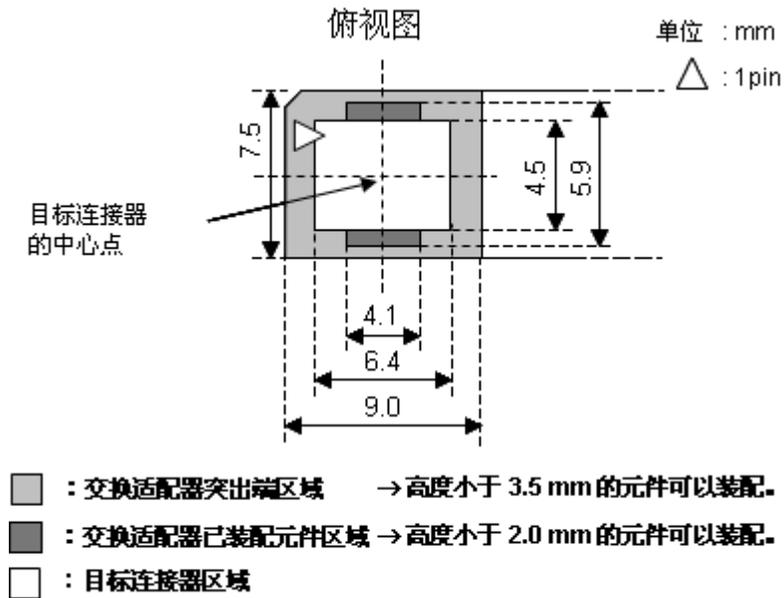
附录B 目标系统设计注意事项

本章说明使用在线仿真器 QB-78K0SKX1 时，目标系统上哪些区域禁止安装元件，哪些区域安装元件有高度限制。

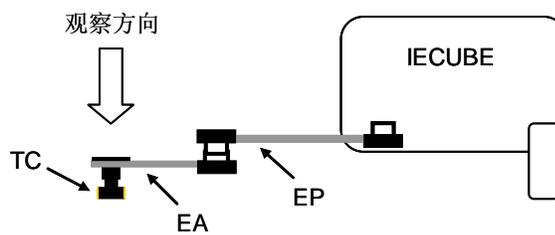
关于目标连接器、交换适配器、和仿真探头的封装图，请参见如下网址。

http://www.necel.com/micro/en/development/asia/iecube/outline_QB.html

图 B-1. 当使用 78K0S/Kx1+仿真探针(用于 10 引脚 MA 封装)时

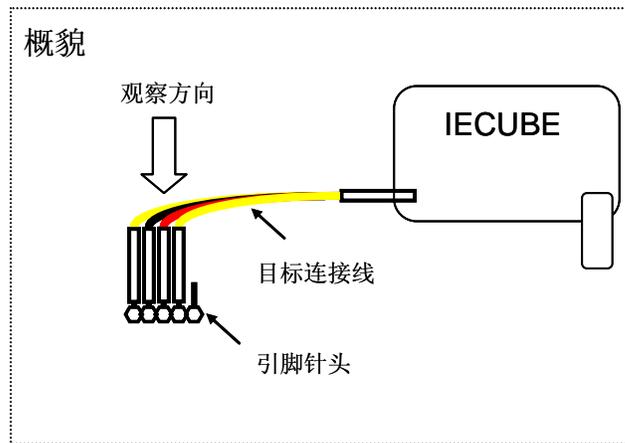
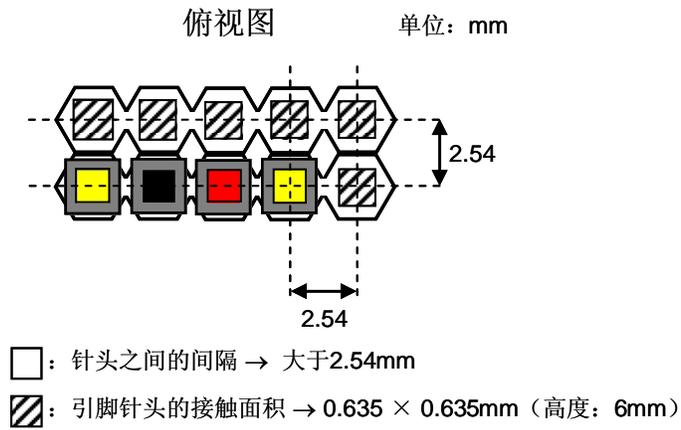


概貌



注 EP: 仿真探头
EA: 交换适配器
TC: 目标连接器

图 B-2. 当使用 78K0S/Kx1+ 目标电缆 (单线)时



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) ... 74
8 位定时器 H 比较寄存器 11 (CMP11) ... 74
8 位定时器 H 模式寄存器 1 (TMHMD1) ... 75

[E]

外部中断模式寄存器 0 (INTM0) ... 101

[F]

闪存地址指针 H 比较寄存器 (FLAPHC)... 152
闪存地址指针 L 比较寄存器(FLAPLC) ... 152
闪存地址指针 H (FLAPH) ... 152
闪存地址指针 L (FLAPL) ... 152
闪存编程指令寄存器(FLCMD) ... 151
闪存编程模式控制寄存器(FLPMC) ... 147
闪存保护指令寄存器(PFCMD) ... 148
闪存状态寄存器(PFS) ... 149
闪存写入缓冲寄存器(FLW) ... 153

[I]

中断屏蔽标志寄存器 0 (MK0) ... 101
中断请求标志寄存器 I 0 (IF0) ... 100

[L]

低速内部振荡模式寄存器(LSRCM) ... 64
低电压检测寄存器(LVIM) ... 124
低电压检测电平选择寄存器(LVIS) ... 125

[P]

端口模式寄存器 2 (PM2) ... 56, 77
端口模式寄存器 3 (PM3) ... 56
端口模式寄存器 4 (PM4) ... 56
端口寄存器 2 (P2) ... 57
端口寄存器 3 (P3) ... 57
端口寄存器 4 (P4) ... 57
预处理器时钟控制寄存器 (PPCC) ... 63
处理器时钟控制寄存器 (PCC) ... 63
上拉电阻选项寄存器 2 (PU2) ... 58
上拉电阻选项寄存器 3 (PU3) ... 58
上拉电阻选项寄存器 4 (PU4) ... 58

[R]

复位控制标志寄存器 (RESF) ... 118

[W]

看门狗定时器启用寄存器 (WDTE) ... 91

看门狗定时器模式寄存器(WDTM) ... 90

C.2 寄存器索引(符号)**[C]**

CMP01: 8 位定时器 H 比较寄存器 01 ... 74

CMP11: 8 位定时器 H 比较寄存器 11 ... 74

[F]

FLAPH: 闪存 地址指针 H ... 152

FLAPHC: 闪存地址指针 H 比较寄存器... 152

FLAPL: 闪存地址指针 L ... 152

FLAPLC: 闪存地址指针 L 比较寄存器... 152

FLCMD: 闪存编程指令寄存器... 151

FLPMC: 闪存编程模式控制寄存器... 147

FLW: 闪存写入缓冲寄存器... 153

[I]

IF0: 中断请求标志寄存器 0 ... 100

INTM0: 外部中断模式寄存器 0 ... 101

[L]

LSRCM: 低速内部振荡模式寄存器... 64

LVIM: 低电压检测寄存器... 124

LVIS: 低电压检测电平选择寄存器... 125

[M]

MK0: 中断屏蔽标志寄存器 0 ... 101

[P]

P2: 端口寄存器 2 ... 57

P3: 端口寄存器 3 ... 57

P4: 端口寄存器 4 ... 57

PCC: 处理器时钟控制寄存器 ... 63

PFCMD: 闪存保护指令寄存器... 148

PFS: 闪存状态寄存器... 149

PM2: 端口模式寄存器 2 ... 56, 77

PM3: 端口模式寄存器 3 ... 56

PM4: 端口模式寄存器 4 ... 56

PPCC: 预处理器时钟控制寄存器... 63

PU2: 上拉电阻选项寄存器 2 ... 58

PU3: 上拉电阻选项寄存器 3 ... 58

PU4: 上拉电阻选项寄存器 4 ... 58

[R]

RESF: 复位控制标志寄存器 ... 118

[T]

TMHMD1: 8 位定时器 H 模式寄存器 1 ... 75

[W]

WDTE: 看门狗定时器启用寄存器... 91

WDTM: 看门狗定时器模式寄存器... 90

附录 D 注意事项列表

本附录列出了本文档中的注意事项。

表中“类别”（硬件/软件）定义如下：

硬件：微控制器内部/外部硬件的注意事项

软件：寄存器设置或程序等软件的注意事项

(1/7)

章节	类别	功能	功能详情	注意事项	页码	
第2章	硬件	引脚功能	P22, P23/EXCLK, P34/RESET	复位期间，P22 引脚和 P23/EXCLK 引脚电平被强制下拉。而 P34/RESET 引脚电平在复位期间，由复位功能引脚/上电清零电路强制上位。	pp. 19-21 <input type="checkbox"/>	
第3章	软件	存储空间	SP: 堆栈指针	由于RESET 输入并未定义SP 内容，所以在使用堆栈前必须先对SP 初始化。	p. 32 <input type="checkbox"/>	
				堆栈指针只能设置到高速RAM区，而且实际上只有其低10位可以设置为堆栈指针。 0FF00H 在特殊功能寄存器SFR区，而不在高速RAM区，因此将其转换为位于高速RAM区的0FB00H。 当数据真正入栈后，0FB00H 减1至0FAFFH，但数据并不位于高速RAM区，因此要将0FB00H转换为0FEFFH，即与堆栈指针设置为0FF00H 时的值相同。	p. 32 <input type="checkbox"/>	
第4章	硬件	端口功能	P22, P23/EXCLK, P34/RESET	复位期间，P22 引脚和 P23/EXCLK 引脚电平被强制下拉。而 P34/RESET 引脚电平在复位期间，由复位功能引脚/上电清零电路强制上位。	p. 48 <input type="checkbox"/>	
				P34	由于P34引脚可作 RESET 功能引脚使用，因此，如果该引脚作输入端口引脚用，就不能用作向RESET 引脚输入外部复位信号。该引脚的端口功能是由选择字节选择设置的。需要了解有关更详细的情况，请参阅 第十三章 选择字节。 而且，由于选择字节是在复位释放以后才引用的，因此如果在引用之前向 RESET 引脚输入了低电平信号，就没法释放复位状态。当该引脚用作输入端口时，需连接上拉电阻。	p. 54 <input type="checkbox"/>
				P21, P32	由于P21和P32也作外部中断输入引脚使用，所以，如果这些引脚各位设置为输出模式并且改变了其输出电平，则相应的中断请求标志置1。因此，如果使用该端口引脚的输出模式，要事先将相应的中断屏蔽位置为1。	p. 56 <input type="checkbox"/>
				-	尽管1位存储器操作指令按位操作，但是仍然可以访问8位端口。因此，如果一个端口既有输入引脚又有输出引脚，即便对其进行位操作，则对于端口的所有定义为输入的引脚的输出锁存内容是未定义的。	p. 59 <input type="checkbox"/>
第6章	软件	8位定时器 H1	CMP01: 8位定时器H比较寄存器01	在定时器计数期间，CMP01无法进行重置操作。	p. 74 <input type="checkbox"/>	

章节	分类	功能	详细功能	注意事项	页码	
第6章	软件	8位定时器H1	CMP11: 8位定时器H比较寄存器11	PWM输出模式下, 停止定时器计数后 (TMHE1 = 0), 再次开始定时器计数操作前 (TMHE1 = 1) 请确保先设置CMP11, 即使新值与旧值相同, 也得再次设置。	p. 74 <input type="checkbox"/>	
			TMHMD1: 8位定时器H模式寄存器1	当TMHE1 = 1, 禁止对TMHMD1 寄存器的其它位进行设置。 PWM输出模式下, 停止定时器计数后 (TMHE1 = 0), 再次开始定时器计数操作前 (TMHE1 = 1)请确保先设置8位定时器H比较寄存器CMP11, 即使新值与旧值相同, 也得再次设置。	p. 76 <input type="checkbox"/> p. 76 <input type="checkbox"/>	
	硬件		PWM 输出	PWM输出模式下, 可在定时器计数期间重新设置CMP11寄存器的值。但是, 新值写入后需要三个或三个以上工作时钟周期 (通过使用TMHMD1寄存器的CKS12到CKS10位来选择) 才能起作用。	p. 82 <input type="checkbox"/>	
				停止定时器计数后 (TMHE1 = 0), 再次开始定时器计数操作前 (TMHE1 = 1) 请确保先设置CMP11, 即使新值与旧值相同, 也得再次设置。	p. 82 <input type="checkbox"/>	
	软件			请确定CMP11寄存器设置值 (M) 和CMP01寄存器设置值 (N) 位于如下范围内。 $00H \leq CMP11 (M) < CMP01 (N) \leq FFH$	p. 82 <input type="checkbox"/>	
	第7章	软件	看门狗定时器	WDTM: 看门狗定时器模式寄存器	将第7, 6, 5位分别设置为0, 1和1。不要设置为其它值。 重置键释放后, WDTM仅能被8位存储器操作指令写入一次。如果试图第二次写入, 将产生内部重置信号。但是, 如果WDCS4和WDCS3相应被设置为“1”和“x”, 且看门狗定时器已被停止, 则即使执行以下操作也不会产生内部重置信号。 • 第二次写WDTM • 发送1位存储器操作指令给WDTE • 发送一个非“ACH”的值给WDTE	p. 90 <input type="checkbox"/> p. 91 <input type="checkbox"/>
				不能通过1位存储器操作指令设置WDTM。	p. 91 <input type="checkbox"/>	
				当使用flash存储器程序时, 请为看门狗定时器设置溢出时间以确保溢出时间不会太短 (例 1个字节的写入: 最少200 μ s, 1个模块的删除: 最少10 ms)。	p. 91 <input type="checkbox"/>	
				WDTE: 看门狗定时器使能寄存器	如果将一个非ACH的值写入WDTE, 将产生一个内部重置信号。 如果通过1位存储器操作指令操作WDTE, 将产生一个内部重置信号。	p. 91 <input type="checkbox"/> p. 91 <input type="checkbox"/>
				从WDTE读回的值为9AH (该值不同于写入值ACH)。	p. 91 <input type="checkbox"/>	
硬件			当通过选项字节选择“低速内部振荡器无法停止”时	该模式下, 即使在STOP指令执行时, 也无法停止看门狗定时器的工作。对于8位定时器H1, 可选择低速内部振荡时钟作为其计数源, 因此STOP指令执行后, 看门狗定时器溢出前, 需通过TMH1终端请求清除看门狗定时器的计数值。如果在STOP指令执行后, 看门狗定时器溢出前, 不进行计数值的清除工作, 则将产生内部重置信号。	p. 92 <input type="checkbox"/>	
			当通过选项字节选择“低速内部振荡器可被停止”时	该模式下, 看门狗定时器在HALT/STOP指令执行期间停止工作。HALT/STOP模式指令释放后, HALT/STOP指令被WDTM执行之前, 计数再次启动, 并以看门狗定时器的工作时钟作为计数时钟。此时, 计数值不会被清零, 而是保持原值不变。	p. 94 <input type="checkbox"/>	

章节	类别	功能	功能详情	注意事项	页码
第8章	软件	中断功能	IF0: 中断请求标识寄存器0, MK0: 中断屏蔽标识寄存器0	因为P21和P32还有作为外部中断输入的功能, 所有通过指定端口功能的输出模式来改变输出电平时, 中断请求标识将被置1。因此, 使用输出模式前应把中断屏蔽标识置1。	pp. 100, 101 <input type="checkbox"/>
			INTM0: 外部中断模式寄存器0	请确保将第0, 1, 6, 7位清零。 在设置INTM0寄存器前, 请确保先设置相应中断屏蔽标识 ($\times MK \times = 1$), 关闭中断。设置好INTM0寄存器后, 请清除中断请求标识 ($\times \times IF \times = 0$) 然后再清除中断屏蔽标识 ($\times \times MK \times = 0$), 开启中断服务。	p. 101 <input type="checkbox"/> p. 102 <input type="checkbox"/>
			中断请求被挂起	在读写中断请求标识寄存器0 (IF0) 或中断屏蔽标识寄存器0 (MK0) 时, 中断请求将被挂起。	p. 104 <input type="checkbox"/>
			中断请求挂起	多中断可以被识别, 即使是其中优先级最低的中断也不例外。	p. 105 <input type="checkbox"/>
第9章	软件	等待功能	-	只有当选择字节被设置为“低速内部振荡器可被程序停止”时, LSRSTOP设置才会有效。	p. 107 <input type="checkbox"/>
			硬件	STOP 模式	当转换到STOP模式时, 请确保在执行STOP指令前, 先停止外围硬件的工作 (除了工作时钟为低速内部振荡器时钟的外围硬件)。 如果在进入STOP模式前低速内部振荡器就已经开始工作了, 则STOP模式中无法让低速内部振荡器停止工作。(参加表9-1)
	软件		HALT 模式设置及其工作状态	由于中断请求信号可用来释放等待模式, 因此如果某个中断源的中断请求标识被置1, 且中断屏蔽标识被清0, 则等待模式 (如果正处在该模式下) 标识将立即被清零。	p. 109 <input type="checkbox"/>
			STOP模式设置及其工作状态	由于中断请求信号可用来释放等待模式, 因此如果某个中断源的中断请求标识被置1, 且中断屏蔽标识被清0, 则等待模式 (如果正处在该模式下) 标识将立即被清零。这样, 在STOP模式下, STOP指令执行后, 恢复正常工作模式, 然后工作停止34 μs (典型值)。	p. 111 <input type="checkbox"/>
第10章	硬件	复位功能	-	对于外部重置, 输入到RESET引脚上的低电平至少要保持2 μs 。 重置信号产生期间, 系统时钟和低速内部振荡器时钟停止振荡。 当RESET引脚作为单向输入端口引脚 (P34) 使用时, 如果有低电平输入到RESET引脚中, 且重置资源已被POC电路、LVI电路或看门狗寄存器释放, $\mu PD78F9500$, $78F9501$, $78F9502$ 将被重置。重置状态将保持到有高电平输入到RESET引脚时为止。 LVI不能被LVI电路的内部重置信号重置。	p. 114 <input type="checkbox"/> p. 114 <input type="checkbox"/> p. 114 <input type="checkbox"/> p. 115 <input type="checkbox"/>
			通过看门狗定时器溢出引发重置的时序	看门狗定时器还可被看门狗定时器的内部重置信号重置。	p. 116 <input type="checkbox"/>
			RESF: 重置控制标识寄存器	请不要用1位操作指令来读取数据。	p. 118 <input type="checkbox"/>

章节	类别	功能	功能详情	注意事项	页码
第11章	软件	上电清零电路	上电清零电路功能	如果内部重置信号产生于POC电路中，则重置控制标识寄存器（RESF）被清零。	p. 119 <input type="checkbox"/>
	POC电路的检测电压（V _{POC} ）位于2.1 V ±0.1 V之间，因此应采用2.2到5.5 V的电压供电。			p. 119 <input type="checkbox"/>	
	使用上电清零电路的注意事项			当系统供电电压（V _{DD} ）在POC检测电压值（V _{POC} ）附近波动一定时长时，系统可能会反复重置和释放。此时，可通过以下措施任意设置从重置释放到微控制器开始工作间的时长。	p. 121 <input type="checkbox"/>
第12章	软件	低电压检测器	LVIM: 低压检测寄存器	请按如下任意一种操作方式来关闭LVI。	p. 124 <input type="checkbox"/>
				<ul style="list-style-type: none"> 采用8位存储器操作指令时：向LVIM写入00H。 采用1位存储器操作指令：将LVION清零。 	
				请确保第2至6位被清零。	p. 124 <input type="checkbox"/>
			LVIS: 低压检测电平选择寄存器	第4至7位必须设置为0。	p. 125 <input type="checkbox"/>
				如果在LVI工作期间向LVIS写入上表以外的其它值，该值将会立即变为未定义。因此在写LVIS之前，请确保先关闭LVI（将LVIM寄存器第7位（LVION）清零）。	p. 125 <input type="checkbox"/>
	采用重置模式时	<1> 步骤<1>必须保持。因为如果当LVIMK = 0时，步骤<3>完成后可能会立即产生中断。	p. 126 <input type="checkbox"/>		
		LVIMD被设置为1后，供电电压（V _{DD} ）≥ 检测电压（V _{LVI} ）时不会产生内部重置信号。	p. 126 <input type="checkbox"/>		
	使用低压检测器的注意事项	当系统供电电压（V _{DD} ）在LVI检测电压值（V _{LVI} ）附近波动一定时间时，需根据低压检测器模式的不同来采取不同措施。	p. 130 <input type="checkbox"/>		
		<1> 采用重置模式时 系统可能会反复重置和释放。 此时，可通过下述操作（1）任意设置从重置释放到微控制器开始工作期间的等待时长。			
		<2> 采用中断模式时 中断请求可能会频繁产生。执行下述操作（2）的第（b）步。			
第13章	硬件	选项字节	RESET引脚的控制	由于重置释放后需引用选项字节，所以如果在引用前向RESET引脚输入低电平，则重置状态将无法释放。 当用作单向输入端口（P34）时，可通过PU3上的PU34设置片上上拉电阻的值。	p. 134 <input type="checkbox"/>
			系统时钟源的选择	因为EXCLK引脚还可作为P23引脚使用，因此根据系统时钟源的不同，EXCLK引脚的用途也将不同。 (1) 选择外部时钟输入 由于P23引脚被用作外部时钟输入端口，因此不能再被用作I/O端口。 (2) 选择高速内部振荡时钟 P23引脚可被用作I/O端口。	p. 134 <input type="checkbox"/>

章节	类别	功能	功能详情	注意事项	页码
第13章	硬件	选项字节	低速内部振荡器	如果选择“低速内部振荡器不能被停止”，则看门狗定时器（WDT）的计数时钟只能采用低速内部振荡时钟。	p. 135 <input type="checkbox"/>
				如果选择“低速内部振荡器可被软件停止”，那么无论低速内部振荡模式寄存器（LSRCM）是否被置1，只要进入HALT/STOP模式，WDT计数时钟都将被关闭。同样，采用低速内部振荡器以外的时钟作为WDT的计数时钟，只要进入HALT/STOP模式后也将被关闭。 低速内部振荡器工作时（LSRSTOP = 0），可作为8位定时器H1的计数时钟，即使在STOP模式下也不会被关闭。	p. 135 <input type="checkbox"/>
			当RESET引脚用作单向输入端口（P34）时的注意事项	使用专用FLASH存储器编程器对设备进行二次擦写时，请注意如下几点。其中被写设备已通过选项字节设置为“RESET引脚作为单向输入端口（P34）使用”。向目标系统供电前，请先连接并开启专用FLASH存储器编程器。 如果在连接编程器前先对目标系统上电，则目标系统将无法切换到FLASH存储器编程模式。	p. 135 <input type="checkbox"/>
第14章	软件	闪存	安全设置	设置批擦除安全设置之后，不可以对期间进行擦除。此外，即使执行写入命令，不可以写入已写入闪存之外的数据，因为已禁用擦除命令	p. 143 <input type="checkbox"/>
			自编程功能	自编程过程必须在进行自编程之前包括在程序中。	p. 144 <input type="checkbox"/>
		执行自编程命令时禁止执行其他指令。因此，看门狗定时计数器提前清零和重启，以防自编程期间看门狗定时器溢出。有关执行自编程所花时间的详细情况，请参看表 14-8。	p. 147 <input type="checkbox"/>		
		自编程期间发生的中断可以在自编程模式结束后确认。为了避免此操作，在通过特定顺序将模式从正常模式切换到自编程模式之前，禁用中断服务（通过将MKO 设为FFH并执行DI指令）。	p. 147 <input type="checkbox"/>		
		执行自编程命令时不适用RAM。	p. 147 <input type="checkbox"/>		
		如果电源电压降低或者输入复位信号时，写入或擦除闪存，则不保证写入/擦除顺利完成。	p. 147 <input type="checkbox"/>		
		块擦除期间空白数据设置的值是FFH。	p. 147 <input type="checkbox"/>		
		自编程期间设置CPU时钟，以使其为1 MHz或者更大。	p. 147 <input type="checkbox"/>		
		进行自编程模式的特定顺序设置之后立即执行NOP和HALT指令，接着执行自编程。此时，10 μs(MAX.) + 2 CPU时钟(fcpu)之后自动释放HALT指令。	p. 147 <input type="checkbox"/>		
		如果选择振荡器的时钟或外部时钟作为系统时钟，则在进行自编程模式的特定顺序设置之后立即执行NOP和HALT指令，释放HALT状态之后等待 8 μs且接着进行自编程。	p. 147 <input type="checkbox"/>		
使用1-位内存操作指令检查FPRERR。	p. 147 <input type="checkbox"/>				
自编程模式下的引脚状态与HALT模式下相同。	p. 147 <input type="checkbox"/>				

章节	类别	功能	功能详情	注意事项	页码
第14章	软件	闪存	自编程功能	由于在自编程模式下禁用经由板上/离板编程的安全功能设置，因此不管安全功能如何设置，均可以执行自编程命令。自编程期间为了禁用写入或者擦除过程，请设置保护位。	p. 147 <input type="checkbox"/>
				执行自编程命令之前确保闪存地址指针H (FLAPH) 和闪存地址指针 H 比较寄存器 (FLAPHC) 的位4到7 清零。如果当执行自编程命令时这些位的值是1，则可能装置不能正常运行。	p. 147 <input type="checkbox"/>
				设置自编程模式和正常工作模式之前FLCMD寄存器的值立即清零。	p. 147 <input type="checkbox"/>
			FLPMC: 闪存编程模式控制寄存器	在设置自编程模式时的注意事项，请参看14.8.2 自编程功能注意事项。	p. 148 <input type="checkbox"/>
				设置CPU时钟，以使其在自编程期间为1 MHz或者更大值。	p. 148 <input type="checkbox"/>
				进行自编程模式的特定顺序设置之后，立即执行NOP和HALT指令，接着进行自编程。此时，在10 μ s (MAX.) + 2 CPU时钟 (f _{cpu}) 之后自动执行HALT指令。	p. 148 <input type="checkbox"/>
				如果选择振荡器的时钟或外部时钟作为系统时钟，则在进行自编程模式的特定顺序设置之后立即执行NOP和HALT指令，释放HALT状态之后等待 8 μ s且接着进行自编程。	p. 148 <input type="checkbox"/>
			PFCMD: 闪存保护命令寄存器	自编程模式下不能执行中断服务。进行自编程模式的特定顺序设置之前和将此模式变为正常模式的特定顺序之后，禁用中断服务（当MK0和MK1 = FFH时通过执行DI指令）。	p. 149 <input type="checkbox"/>
				PFS: 闪存状态寄存器	使用1-位内存操作指令检查FPRERR。
			FLAPH, FLAPL: 闪存地址指针 H和L	执行自编程命令之前确保FLAPH和FLAPHC的位4 到7清零。如果当执行自编程命令时这些位的值是1，则可能装置不能正常运行。	p. 152 <input type="checkbox"/>
			FLAPHC, FLAPLC: 闪存地址指针 H/L 比较寄存器	执行自编程命令之前确保FLAPH和FLAPHC的位4到7清零。如果当执行自编程命令时这些位的值是1，则可能装置不能正常运行。	p. 152 <input type="checkbox"/>
				设置对FLAPHC进行块擦除、验证或者空白检查的块号码（与FLAPH相同的值）。	p. 152 <input type="checkbox"/>
				当块擦除时FLAPLC清零，且当空白检查时此寄存器设为FFH。	p. 152 <input type="checkbox"/>

章节	类别	功能	功能详情	注意事项	页码
第14章	软件	闪存	切换到自编程模式	确保在未擦除或写入数据的地址处执行0使用用户程序的上除系列操作。	pp. 154, <input type="checkbox"/> 155, 157, 158
			切换到正常模式		
			Byte write	如果写入结果失败, 将该块擦除一次并再次写入。	p. 166 <input type="checkbox"/>
第15章	硬件	片上调试功能	将QB-MINI2连接到PD78F9500、78F9501、78F9502	禁止对指定大规模生产的产品使用片上调试功能, 因为使用这一功能时可能超过闪存可允许的承保写入次数, 且因此不能保证产品的可靠性。对于使用片上调试功能时发生的问题, NEC Electronics不承担任何责任。	p. 190 <input type="checkbox"/>
				电路连接实例中所描述的常量仅为参考值。如果您针对大规模生产进行闪存编程, 需要充分估计是否满足目标装置的规格。	p. 190 <input type="checkbox"/>
			QB-MINI2 用于调试且INTP1 引脚的调试仅通过实机进行的情况	如果通过实机运行进行调试而不使用QB-MINI2, 则使用QB-编程器写入用户程序。由调试器下载的程序包括监控程序, 且如果监控程序不经过QB-MINI2控制则会出现程序故障。	p. 192 <input type="checkbox"/>
第17章	硬件	电气特性	最大绝对额定值	如果最大绝对额定值即使瞬间超过任何参数, 产品质量也会受到影响。也就是, 最大绝对额定值是产品受到物理损坏的边界额定值, 因此, 必须在确保不超过最大绝对额定值的条件下使用产品。	p. 204 <input type="checkbox"/>
第19章	硬件	建议焊接条件	无铅产品	零件号码一端带产品-A 的产品是无铅产品。	p. 215 <input type="checkbox"/>
			-	如需以下建议接方法和条件之外的焊接方法和条件, 请联系NEC Electronics 销售代表。	p. 215 <input type="checkbox"/>
				禁止同时使用不同焊接方法(局部加热除外)。	p. 215 <input type="checkbox"/>

E.1 本版本主要修订之处

页码	描述
p. 50	修改了 20 页和 21 页中图 4-2 结构框图
p. 52	修改了 23 页中图 4-4 结构框图
p. 114	第 10 章 复位功能 • 修改了注意事项 3
p. 138	14.4 通过闪存编程器写入 • 添加了 FlashPro5 至专门闪存编程器 • 删除了专门闪存编程器的 PG-FPL2 • 修改了备注
pp. 139, 140	14.5 编程环境 • 修改了图 14-2 程序写入闪存环境(FlashPro4/FlashPro5/ QB-MINI2) 并添加了注 • 修改了表 14-2 PD78F9500, 78F9501, 78F9502 和 FlashPro4/FlashPro5/ QB-MINI2 的接线图 并添加了注 2 • 修改了图 14-3 FlashPro4/FlashPro5/QB-MINI2 接线图 • 删除了专门闪存编程器 的 PG-FPL2
p. 142	修改了 图 14-5 通信命令
p. 146	在表 14-8 自编程控制命令中添加了注
p. 190	添加了第 15 章 片上调试功能
pp. 205, 206	第 17 章 电气特性 • 修改了高速内部振荡器特点 • 修改了 输入电压, 高(V_{IH1})和输入电压, 低 (V_{IL1})的 DC 特点
p. 220	A.4 闪存写入工具 • 添加了 FlashPro5 • 删除了 PG-FPL2
p. 223	添加了 附录 B 有关目标系统设计的注意事项
p. 235	添加了 附录 E 修订记录

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