

致尊敬的顾客

关于产品目录等资料中的旧公司名称

NEC电子公司与株式会社瑞萨科技于2010年4月1日进行业务整合（合并），整合后的新公司暨“瑞萨电子公司”继承两家公司的所有业务。因此，本资料中虽还保留有旧公司名称等标识，但是并不妨碍本资料的有效性，敬请谅解。

瑞萨电子公司网址：<http://www.renesas.com>

2010年4月1日
瑞萨电子公司

【发行】瑞萨电子公司（<http://www.renesas.com>）

【业务咨询】<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

78K0S/KB1+

8 位单片微控制器

μPD78F9232	μPD78F9234
μPD78F9232(A)	μPD78F9234(A)
μPD78F9232(A2)	μPD78F9234(A2)

文档编号	U17446CA5V0UD00 (第 5 版)
发行日起	2010 年 8 月

[备注]

CMOS 设备注意事项

① 输入引脚处的电压波形

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

② 未使用的输入引脚的处理

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

③ ESD防护措施

如果MOS设备周围有强电场，将会击穿氧化栅极，从而影响设备的运行。因此必须采取措施，尽可能防止静电产生。一旦有静电，必须立即释放。对于环境必须有适当的控制。如果空气干燥，应当使用增湿器。建议避免使用容易产生静电的绝缘体。半导体设备的存放和运输必须使用抗静电容器、抗静电屏蔽袋或导电材料容器。所有的测试和测量工具包括工作台和工作面必须良好接地。操作员应当佩戴静电消除手带以保证良好接地。不能用手直接接触半导体设备。对于装配有半导体设备的PW板也应采取类似的静电防范措施。

④ 初始化之前的状态

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

⑤ 电源开关顺序

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

⑥ 电源关闭状态下的输入信号

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

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

PC/AT是International Business Machines Corporation的商标。

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

SPARCstation是SPARC International, Inc.的商标。

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

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

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

- 本文档所登载的内容有效期截止至 2010 年 08 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。

M5 02.11-1

引言

目标读者 本手册适用于那些希望理解 78K0S/KB1+产品功能，并设计开发相关应用系统和程序的用户工程师。
主要产品如下。

- 78K0S/KB1+: μ PD78F9232, 78F9234, 78F9232(A), 78F9234(A), 78F9232(A2), 78F9234(A2)

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 78K0S/KB1+ 有两本手册：本手册和指令手册(78K/0S 系列通用)。

78K0S/KB1+ 用户手册	78K/0S 系列 指令用户手册
• 引脚功能 • 内部模块功能 • 中断 • 其他内置外设功能 • 电气特性	• 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。

- ◇ 如何理解 78K0S/KB1+ 的整体功能
 - 按照 **目录** 顺序阅读本手册。标记<R>表示主要修改点。通过在 PDF 文件的“查找”区域中指定“<R>”，可以方便地找到修改点。
- ◇ 如何阅读寄存器格式
 - 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 78K0S/KB1+ 的电气特性
 - 请参阅 **第二十二章**和**第二十三章 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 xxxxB
		十进制 ... xxxx
		十六进制 ... xxxxH

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

设备相关文档

文档名称	文档编号
78K0S/KB1+ 用户手册	本手册
78K/0S 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 Ver.2.00	操作篇
	语言篇
	结构化汇编语言篇
CC78K0S C 编译器 Ver.2.00	操作篇
	语言篇
SM+ 系统仿真器	操作篇
	用户开发接口
ID78K0S-QB 集成调试器 Ver.3.00	操作篇
PM+ Ver.6.30	U18416E

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

Flash存储器编程的相关文档

文档名称	文档编号
PG-FP5 Flash 存储器编程器用户手册	U18865E
QB-Programmer 编程软件	操作篇
	U18527E

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。

其它相关文档

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

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

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

目录

第一章 概述.....	14
1.1 特征.....	14
1.2 订购信息	15
1.3 引脚配置（俯视图）	16
1.4 78K0S/Kx1+ 产品系列.....	18
1.5 框图.....	19
1.6 功能概述	20
第二章 引脚功能	21
2.1 引脚功能列表	21
2.2 引脚功能	23
2.2.1 P00~P03（端口 0）	23
2.2.2 P20~P23（端口 2）	23
2.2.3 P30~P34（端口 3）	23
2.2.4 P40~P47（端口 4）	24
2.2.5 P120~P123（端口 12）	24
2.2.6 P130（端口 13）	24
2.2.7 RESET	24
2.2.8 X1 和 X2	25
2.2.9 AV_{REF}	25
2.2.10 AV_{SS}	25
2.2.11 V_{DD}	25
2.2.12 V_{SS}	25
2.3 引脚 I/O 电路和未使用引脚的连接方式.....	26
第三章 CPU 架构.....	28
3.1 存储器空间	28
3.1.1 内部程序存储空间.....	30
3.1.2 内部数据存储器空间	30
3.1.3 特殊功能寄存器（SFR）区域.....	31
3.1.4 数据存储器寻址	31
3.2 处理器寄存器.....	33
3.2.1 控制寄存器	33
3.2.2 通用寄存器	36
3.2.3 特殊功能寄存器（SFRs）	37
3.3 指令地址寻址	42
3.3.1 相对寻址	42
3.3.2 立即寻址	43
3.3.3 表间接寻址	43
3.3.4 寄存器寻址	44
3.4 操作数地址寻址.....	45
3.4.1 直接寻址	45
3.4.2 短直接寻址	46
3.4.3 特殊功能寄存器（SFR）寻址.....	47
3.4.4 寄存器寻址	48

3.4.5 寄存器间接寻址	49
3.4.6 基址寻址	50
3.4.7 堆栈寻址	51
第四章 端口功能	52
4.1 端口功能	52
4.2 端口配置	54
4.2.1 端口 0	55
4.2.2 端口 2	56
4.2.3 端口 3	57
4.2.4 端口 4	60
4.2.5 端口 12	63
4.2.6 端口 13	65
4.3 控制端口功能的寄存器	66
4.4 端口功能操作	71
4.4.1 写入 I/O 端口	71
4.4.2 从 I/O 端口读入	71
4.4.3 I/O 端口的操作	71
第五章 时钟发生器	72
5.1 时钟发生器的功能	72
5.1.1 系统时钟振荡器	72
5.1.2 产生间隔时间的时钟振荡器	72
5.2 时钟发生器的配置	73
5.3 时钟发生器的控制寄存器	75
5.4 系统时钟振荡器	78
5.4.1 高速内部振荡器	78
5.4.2 晶体/陶瓷振荡器	78
5.4.3 外部时钟输入电路	80
5.4.4 预分频器	80
5.5 CPU 时钟发生器的操作	81
5.6 外围硬件时钟发生器的操作	86
第六章 16 位定时器/事件计数器 00	88
6.1 16 位定时器/事件计数器 00 的功能	88
6.2 16 位定时器/事件计数器 00 的配置	89
6.3 16 位定时器/事件计数器 00 的控制寄存器。	93
6.4 16 位定时器/事件计数器 00 的操作	99
6.4.1 时间间隔定时器操作	99
6.4.2 外部事件计数器操作	101
6.4.3 脉冲宽度测量操作	104
6.4.4 方波输出操作	112
6.4.5 PPG 输出操作	114
6.4.6 单脉冲输出操作	117
6.5 有关 16 位定时器/事件计数器 00 的注意事项	122
第七章 8 位定时器 80	129
7.1 8 位定时器 80 的功能	129

7.2	8 位定时器 80 的配置	130
7.3	控制 8 位定时器 80 的寄存器	132
7.4	8 位定时器 80 的操作	133
7.4.1	作为间隔定时器的操作	133
7.5	8 位定时器 80 的注意事项	135
第八章 8 位定时器 H1		136
8.1	8 位定时器 H1 的功能	136
8.2	8 位定时器 H1 的配置	136
8.3	控制 8 位定时器 H1 的寄存器	139
8.4	8 位定时器 H1 的操作	142
8.4.1	作为间隔定时器/方波输出的操作	142
8.4.2	用作 PWM 输出模式的操作	145
第九章 看门狗定时器		151
9.1	看门狗定时器的功能	151
9.2	看门狗定时器的配置	153
9.3	控制看门狗定时器的寄存器	154
9.4	看门狗定时器的操作	156
9.4.1	当选项字节选择“低速内部振荡器不能停止”时，看门狗定时器的操作	156
9.4.2	当选项字节选择“低速内部振荡器可以由软件停止”时，看门狗定时器的操作	158
9.4.3	STOP 模式下看门狗定时器的操作（当选项字节选择为“低速内部振荡器可以由软件停止”时）	160
9.4.4	HALT 模式下看门狗定时器的操作（当选项字节选择“低速内部振荡器可以由软件停止”时）	162
第十章 A/D 转换器		163
10.1	A/D 转换器的功能	163
10.2	A/D 转换器的配置	165
10.3	A/D 转换器中使用的寄存器	167
10.4	A/D 转换器的操作	172
10.4.1	A/D 转换器的基本操作	172
10.4.2	输入电压和转换结果	174
10.4.3	A/D 转换器操作模式	175
10.5	如何阅读 A/D 转换器特征表	177
10.6	A/D 转换器的注意事项	179
第十一章 串行接口 UART6		182
11.1	串行接口 UART6 的功能	182
11.2	串行接口 UART6 的配置	186
11.3	串行接口 UART6 的控制寄存器	189
11.4	串行接口 UART6 的操作	198
11.4.1	操作停止模式	198
11.4.2	异步串行接口（UART）模式	199
11.4.3	专用波特率发生器	213
第十二章 乘法器		220
12.1	乘法器功能	220
12.2	乘法器配置	220
12.3	乘法器控制寄存器	222

12.4 乘法器操作	223
第十三章 中断功能	225
13.1 中断功能类型	225
13.2 中断源和配置	225
13.3 中断功能的控制寄存器	228
13.4 中断服务操作	233
13.4.1 可屏蔽的中断请求的响应操作	233
13.4.2 中断嵌套	235
13.4.3 中断请求等待	237
第十四章 待机功能	238
14.1 待机功能及配置	238
14.1.1 待机功能	238
14.1.2 待机期间用到的寄存器	240
14.2 待机功能的操作	241
14.2.1 HALT 模式	241
14.2.2 STOP 模式	244
第十五章 复位功能	248
15.1 确认复位源的寄存器	255
第十六章 上电清零电路	256
16.1 上电清零电路的功能	256
16.2 上电清零电路的配置	257
16.3 上电清零电路的操作	257
16.4 上电清零电路的使用注意事项	258
第十七章 低电压检测器	260
17.1 低电压检测器的功能	260
17.2 低电压检测器的配置	260
17.3 低电压检测器的控制寄存器	261
17.4 低电压检测器的操作	263
17.5 低电压检测器的注意事项	266
第十八章 选项字节	269
18.1 选项字节的功能	269
18.2 选项字节的格式	270
18.3 RESET 引脚用作只能输入的端口引脚(P34)时的注意事项	271
第十九章 FLASH 存储器	272
19.1 特点	272
19.2 存储器配置	273
19.3 功能概要	274
19.4 使用 FLASH 编程器的写入	274
19.5 编程环境	275
19.6 On-Board 引脚连接	277

19.6.1 X1 和 X2 引脚	277
19.6.2 RESET 引脚	278
19.6.3 端口引脚	279
19.6.4 供电	279
19.7 On-Board 和 Off-Board Flash 存储器编程	280
19.7.1 Flash 存储器编程模式	280
19.7.2 通讯命令	280
19.7.3 安全设定	281
19.8 通过自编程方式对 FLASH 进行编程	282
19.8.1 自编程概述	282
19.8.2 自编程注意事项	285
19.8.3 自编程功能所使用的寄存器	285
19.8.4 从正常模式切换到自编程模式的示例	293
19.8.5 从自编程模式切换到正常模式的示例	296
19.8.6 自编程模式下块擦除操作示例	299
19.8.7 自编程模式下空白块的校验操作示例	302
19.8.8 自编程模式下字节写入操作示例	305
19.8.9 自编程模式下内部校验操作示例	308
19.8.10 自编程模式下最短命令执行时间的操作示例	312
19.8.11 自编程模式下最短中断禁止时间的操作过程	319
第二十章 片上调试功能	330
20.1 连接 QB-MINI2 和 78K0S/KB1+	330
20.1.1 INTP3 引脚的连接	331
20.1.2 X1 和 X2 引脚的连接	332
20.2 用户资源的保留	333
第二十一章 指令集概述	334
21.1 操作	334
21.1.1 操作数标识符和描述方法	334
21.1.2 “操作”栏的描述	335
21.1.3 “标志”栏的描述	335
21.2 操作列表	336
21.3 按寻址类型的指令列表	340
第二十二章 电气特性（标准产品，（A）等级产品）	344
第二十三章 电气特性（目标值）（（A2）等级产品）	356
第二十四章 封装图	370
第二十五章 推荐焊接条件	373
附录 A 开发工具	375
A.1 软件包	378
A.2 语言处理软件	378
A.3 Flash 存储器写入工具	379

A.3.1 当使用 flash 存储器编程器 PG-FP5 和 FL-PR5 时	379
A.3.2 当使用带编程功能的片上调试仿真器 QB-MINI2 时	379
A.4 调试工具（硬件）	379
A.4.1 当使用在线仿真器 QB-78K0SKX1 时	379
A.4.2 当使用带编程功能的片上调试仿真器 QB-MINI2 时	380
A.5 调试工具（软件）	381
附录 B 目标系统设计注意事项	382
附录 C 寄存器索引	384
C.1 寄存器索引(寄存器名称)	384
C.2 寄存器索引 (符号)	387
附录 D 注意事项列表	390
附录 E 修订历史	410
E.1 该版本的主要修订信息	410
E.2 以往版本的修订历史	411

第一章 概述

<R> 1.1 特征

O 78K0S CPU 核

0 ROM 和 RAM 的容量

项目 产品型号	程序存储器（Flash 存储器）	存储器（内部高速 RAM）
μPD78F9232	4 KB	256 字节
μPD78F9234	8 KB	

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

○ 时钟

- 高速系统时钟... 从如下三个时钟源中选择
 - 陶瓷/晶体 振荡器
 - 2 ~ 10 MHz (标准产品, (A) 等级产品)
 - 2 ~ 8 MHz ((A2) 等级产品)
 - 外部时钟
 - 2 ~ 10 MHz (标准产品, (A) 等级产品)
 - 2 ~ 8 MHz ((A2) 等级产品)
 - 高速内部振荡器
 - 8 MHz $\pm 3\%$ ($-10 \sim +80^{\circ}\text{C}$),
 - 8 MHz $\pm 5\%$ (标准产品, (A) 等级产品) $-40 \sim +85^{\circ}\text{C}$,
 - (A2) 等级产品 $-40 \sim +125^{\circ}\text{C}$)

- 低速内部振荡 (240 kHz (典型值)) ... 看门狗定时器, 该定时器运行于间隔中断模式

O I/O 端口: 26 (CMOS I/O: 24, CMOS 输入: 1, CMOS 输出: 1)

0 定时器: 4 通道

- 16 位定时器/事件计数器: 1 通道...定时器输出 $\times 1$, 捕获输入 $\times 2$
- 8 位定时器: 2 通道...PWM 输出 $\times 1$
- 看门狗定时器: 1 通道...运行于低速内部振荡时钟

0 串行接口: UART (支持 LIN (本地互连网络) 总线) 1 通道

0 片上乘法器: $8 \text{ 位} \times 8 \text{ 位} = 16 \text{ 位}$

○ 10 位分辨率 A/D 转换器：4 通道

○片上内置上电清零（POC）电路（当电压降到 2.1V（典型值）以下时，自动产生复位）

○ 片上内置低电压检测 (LVI) 电路 (当达到检测电压时, 产生中断/复位 (可选))

- 检测电压在 2.35 和 4.3 V 之间有 10 个可选电平

○ 单电源供电的 Flash 存储器

- 支持 **Flash** 自编程
- 软件保护功能 避免外部拷贝（没有 **Flash** 读取命令）
- 通过专用 **Flash** 存储器编程器写入时所需时间： 约为 3 秒钟（4KB）
- * 支持大批量生产线上的 **Flash** 编程

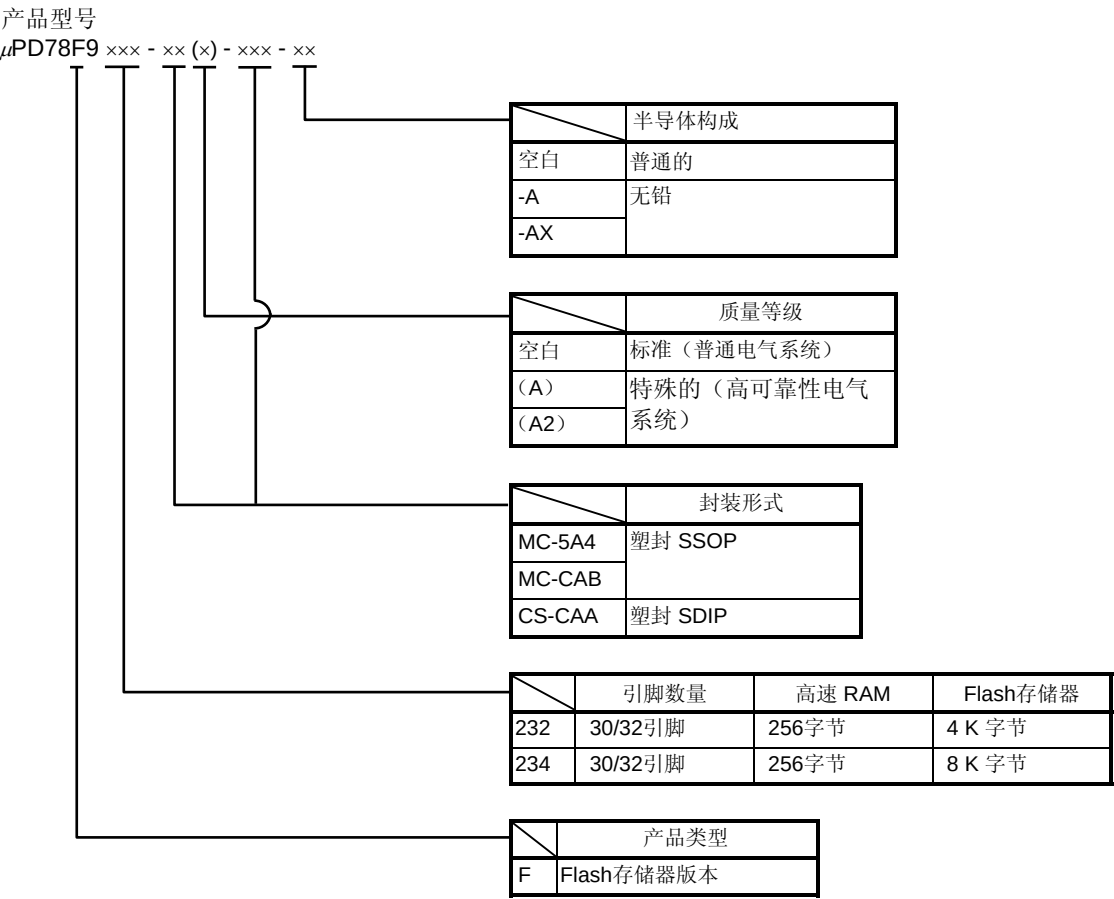
○ 安全功能

- 看门狗定时器的操作时钟和 **CPU** 时钟相互独立
 - ... 即使系统时钟停止也能检测到系统挂起
- 通过 **LVI** 可以检测供电电压降
 - ... 在电源电压降到低于操作电压之前执行适当的处理
- 内置选项字节功能
 - ... 重要的系统操作设置在硬件中设定

○ 支持汇编语言和 C 语言

- 增强的开发环境
 - 支持全功能仿真器（IECUBE），简易仿真器（MINICUBE2）和软件仿真器
- 供电电压： $V_{DD} = 2.0 \sim 5.5 \text{ V}^{\#}$
 - * 在如下电压范围内使用这些产品，因为 POC 电路检测电压（ V_{POC} ）决定供电电压范围。
标准产品，（A）等级产品： $2.2 \sim 5.5 \text{ V}$ （（A2）等级产品） $2.26 \sim 5.5 \text{ V}$
- 工作环境温度范围：
 - 标准产品，（A）等级产品： $T_A = -40 \sim +85^{\circ}\text{C}$
 - （A2）等级产品： $T_A = -40 \sim +125^{\circ}\text{C}$

1.2 订购信息



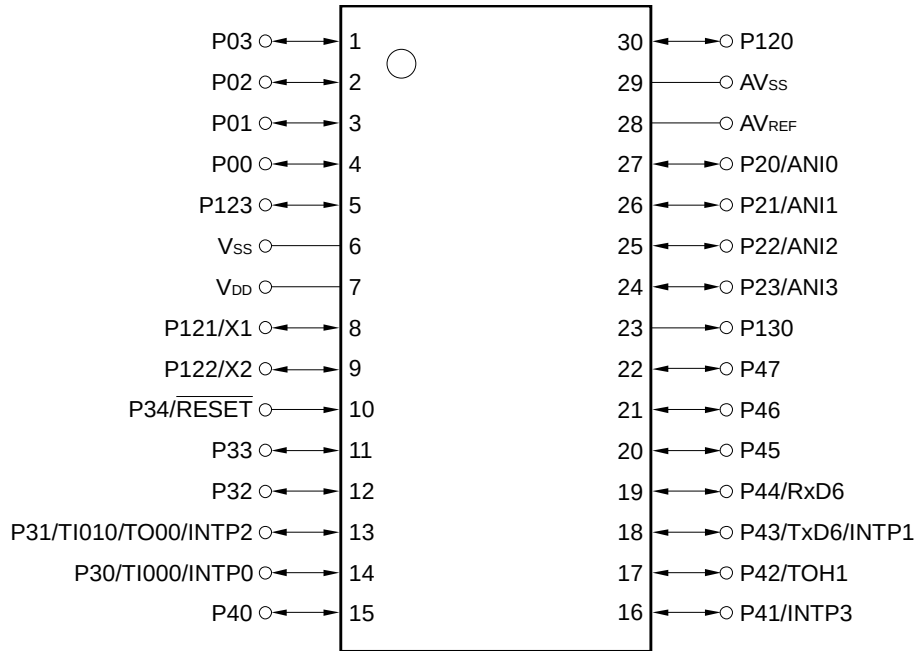
想要了解设备和推荐应用在质量等级方面的规格说明，请参阅NEC Electronics Corporation 出版的“NEC 半导体设备质量等级”（文档编号 C11531E）。

[产品型号列表]

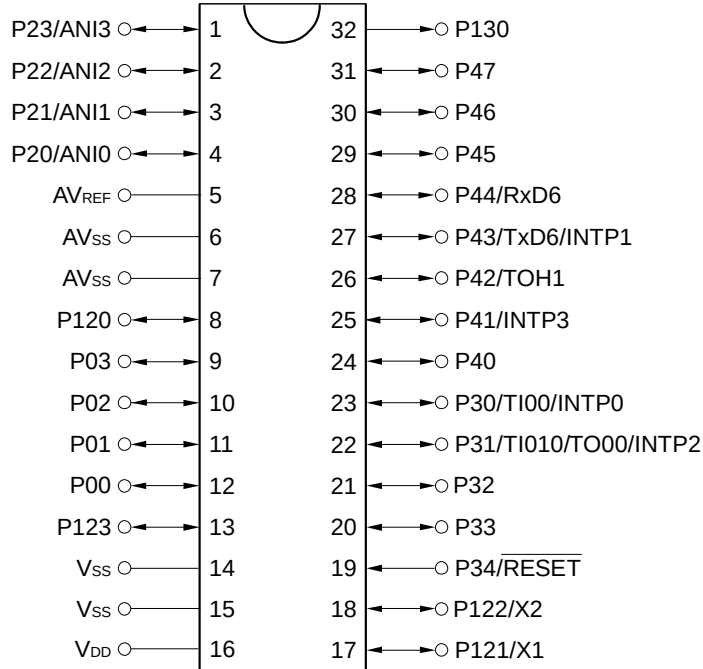
- | | |
|------------------------------------|------------------------------------|
| $\mu\text{PD78F9232MC-5A4-A}$ | $\mu\text{PD78F9234MC-5A4-A}$ |
| $\mu\text{PD78F9232MC(A)-5A4-A}$ | $\mu\text{PD78F9234MC(A)-5A4-A}$ |
| $\mu\text{PD78F9232MC(A2)-5A4-A}$ | $\mu\text{PD78F9234MC(A2)-5A4-A}$ |
| $\mu\text{PD78F9232MC(A)-CAB-AX}$ | $\mu\text{PD78F9234MC(A)-CAB-AX}$ |
| $\mu\text{PD78F9232MC(A2)-CAB-AX}$ | $\mu\text{PD78F9234MC(A2)-CAB-AX}$ |
| $\mu\text{PD78F9232CS-CAA-A}$ | $\mu\text{PD78F9234CS-CAA-A}$ |

1.3 引脚配置（俯视图）

• 30 脚塑封 SSOP



<R> • 32 脚塑封 SDIP



注意事项 连接 AV_{SS} 引脚到 V_{SS}。

引脚名称

<R>	ANIO~ANI3:	模拟输入	P130:	端口 13
	AVREF:	模拟供电电源	RESET:	复位
		模拟参考电压	RxD6:	接收数据
		P20 ~ P23 的供电电源	TI000, TI010:	定时器输入
	AVSS:	模拟电源地	TO00, TOH1:	定时器输出
	INTP0~INTP3:	外部中断输入	TxD6:	发送数据
	P00~P03:	端口 0	VDD:	供电电源
	P20~P23:	端口 2	VSS ^注	地
	P30~P34:	端口 3	X1, X2:	晶体振荡器 (X1 输入时钟)
	P40~P47:	端口 4		
	P121~P123:	端口 12		

1.4 78K0S/Kx1+ 产品系列

下表列出了 78K0S/Kx1+ 系列产品。

<R>

产品型号		78K0S/KU1+	78K0S/KY1+	78K0S/KA1+	78K0S/KB1+
项目		10 脚	16 脚	20 脚	30/32 脚
内部存储器	Flash 存储器	1 KB, 2 KB, 4 KB		2 KB	4 KB 8KB
	RAM	128 字节		128 字节	256 字节
供电电压		V _{DD} = 2.0 ~ 5.5 V ^{#1}			
最小指令执行时间		0.20 μs (10 MHz, V _{DD} = 4.0~5.5 V) 0.33 μs (6 MHz, V _{DD} = 3.0~5.5 V) 0.40 μs (5 MHz, V _{DD} = 2.7~5.5 V) 1.0 μs (2 MHz, V _{DD} = 2.0~5.5 V)			
系统时钟 (振荡频率)		高速内部振荡 (8 MHz (典型值)) 晶体/陶瓷振荡 (2~10 MHz) ^{注 2} 外部时钟输入振荡 (2~10 MHz)			
仅供 TMH1 和 WDT 的时钟 (振荡频率)		低速内部振荡 (240 kHz (典型值))			
端口	CMOS I/O	7	13	15	24
	CMOS 输入	1	1	1	1
	CMOS 输出	—	—	1	1
定时器	16 位 (TM0)	1 通道 ^{注 3}			
	8 位 (TMH)	1 通道			
	8 位 (TM8)	—		1 通道	
	看门狗定时器	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 (2.7~5.5V) ^{注 4}			
乘法器 (8 位 × 8 位)		—			提供
中断	内部	5 ^{注 5}		9	
	外部	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)			
	看门狗定时器	提供			
工作环境温度		标准产品: -40 ~ +85°C	标准产品, (A) 等级产品: -40 ~ +85°C (A2) 等级产品: -40 ~ +125°C		

注 1 在如下电压范围内使用这些产品, 因为 POC 电路检测电压 (V_{POC}) 决定供电电压范围。

标准产品, (A) 等级产品: 2.2 ~ 5.5 V ((A2) 等级产品) 2.26 ~ 5.5 V

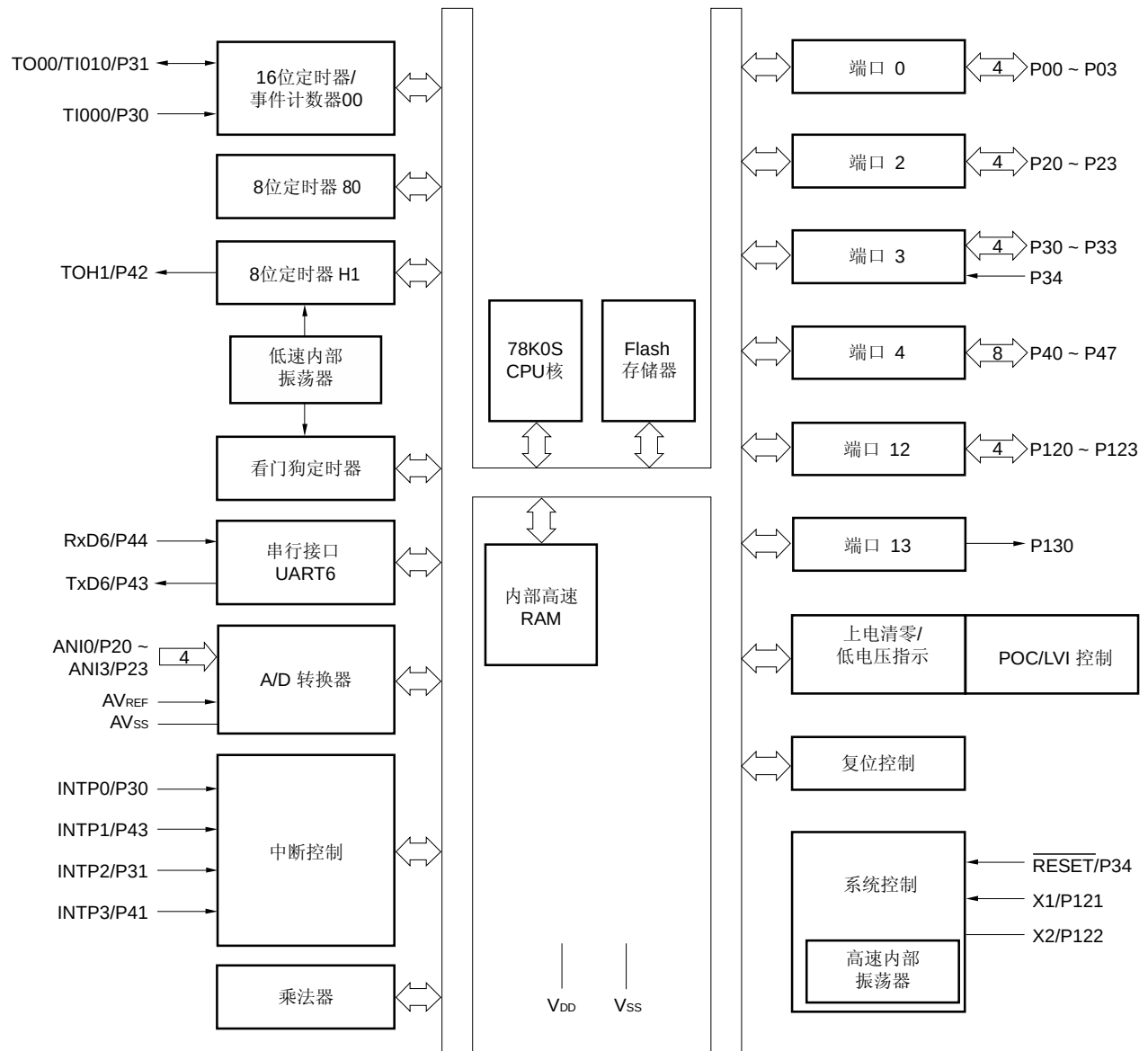
2 $\mu\text{PD78F950x}$ 持晶体/陶瓷振荡器。

3 78K0S/KU1+ 系列不具有 A/D 转换器的产品 ($\mu\text{PD78F950x}$) 不支持 TM0。

4 78K0S/KU1+ 和 78K0S/KY1+ 系列中除了产品 ($\mu\text{PD78F950x}$) 不具有 A/D 转换器外, 其他产品具有 A/D 转换器。

5 78K0S/KU1+ 和 78K0S/KY1+ 中不具有 A/D 转换器的产品分别有 2 个和 4 个内部中断源。

1.5 框图



1.6 功能概述

项目		μ PD78F9232	μ PD78F9234
内部存储器	Flash 存储器	4 KB	8 KB
	高速 RAM	256 字节	256 字节
存储器空间		64 KB	
X1 输入时钟（振荡频率）		晶体/陶瓷/外部时钟输入： 10 MHz（ $V_{DD} = 2.0 \sim 5.5 \text{ V}$ ）	
内部振荡时钟	高速（振荡频率）	内部振荡：8 MHz（典型值）	
	低速（TMH1 和 WDT）	内部振荡：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 μs /0.4 μs /0.8 μs /1.6 μs /3.2 μs （X1 输入时钟： $f_x = 10 \text{ MHz}$ ）	
乘法器		8 位 $\times$ 8 位 = 16 位	
I/O 端口		总共：26 引脚 CMOS I/O：24 引脚 CMOS 输入：1 引脚 CMOS 输出：1 引脚	
定时器		• 16 位定时器/事件计数器：1 通道 • 8 位定时器（定时器 H1）：1 通道 • 8 位定时器（定时器 80）：1 通道 • 看门狗定时器：1 通道	
	定时器输出	2 引脚（PWM：1 引脚）	
A/D 转换器		10 位分辨率 $\times$ 4 通道	
串行接口		支持 LIN 总线的 UART 模式：1 通道	
向量中断源	外部	4	
	内部	9	
复位		• 使用 RESET 引脚复位 • 使用看门狗计时器进行内部复位 • 使用 POC 进行内部复位 • 使用低电压检测器进行内部复位	
供电电压		$V_{DD} = 2.0 \sim 5.5 \text{ V}^{\text{注}}$	
工作环境温度		标准产品，（A）等级产品： $-40 \sim +85^\circ\text{C}$ （A2）等级产品： $-40 \sim +125^\circ\text{C}$	
封装		• 30 引脚塑封 SSOP • 32 引脚塑封 SDIP	

<R>

注 因为上电清零（POC）的检测电压（ V_{POC} ）决定了供电电压范围，所以在如下电压范围内使用本产品：
 标准产品，（A）等级产品：2.2 ~ 5.5 V，（A2）等级产品：2.26 ~ 5.5 V

第二章 引脚功能

2.1 引脚功能列表

(1) 端口引脚

引脚名称	I/O	功能		复位后	复用功能引脚
P00～P03	I/O	端口 0。 4 位 I/O 端口。 可以按位设定输入/输出模式。 可以通过软件设置连接内部上拉电阻。		输入	—
P20～P23	I/O	端口 2。 4 位 I/O 端口。 可以按位设定输入/输出模式。 可以通过软件设置连接内部上拉电阻。		输入	ANI0 ～ ANI3
P30	I/O	端口 3	可以按位设定输入/输出模式。 可以通过软件设置连接内部上拉电阻。	输入	TI000/INTP0
P31					TI010/TO00/ INTP2
P32					—
P33					—
P34 [※]		输入	只能用作输入	输入	RESET [※]
P40	I/O	端口 4。 8 位 I/O 端口。 可以按位设定输入或输出模式。 可以通过软件设置连接内部上拉电阻。		输入	—
P41					INTP3
P42					TOH1
P43					TxD6/INTP1
P44					RxD6
P45					—
P46					
P47					
P120	I/O	端口 12。 4 位 I/O 端口。 可以按位设定输入或输出模式。 只有 P120 和 P123 可以通过软件设置连接内部上拉电阻。		输入	—
P121 [※]					X1 [※]
P122 [※]					X2 [※]
P123					—
P130	输出	端口 13。 1 位端口只能用作输出。		输出	—

注 引脚功能的设置方法，请参考 第十八章 选项字节。

注意事项 在复位期间，P121/X1 和 P122/X2 引脚处于下拉状态。

(2) 非端口引脚

引脚名称	I/O	功能	复位后	复用功能引脚
INTP0	输入	指定外部中断输入的有效沿（上升沿、下降沿或双沿）	输入	P30/TI000
INTP1				P43/TxD6
INTP2				P31/TI010/TO00
INTP3				P41
RxD6	输入	异步串行接口的串行数据输入	输入	P44
TxD6	输出	异步串行接口的串行数据输出	输入	P43/INTP1
TI000	输入	输入到 16 位定时器/事件计数器 00 的外部计数时钟。 输入到 16 位定时器/事件计数器 00 的捕捉寄存器（CR000 和 CR010）的捕捉触发	输入	P30/INTP0
TI010		输入到 16 位定时器/事件计数器 00 的捕捉寄存器（CR000）的捕捉触发		P31/TO00/INTP2
TO00	输出	16 位定时器/事件计数器 00 输出	输入	P31/TI010/INTP2
TOH1	输出	8 位定时器 H1 输出	输入	P42
ANI0 ~ ANI3	输入	A/D 转换器的模拟输入	输入	P20~P23
AVREF	—	A/D 转换器参考电压输入，以及 P20 ~ P23 和 A/D 转换器的正向供电电压	—	—
AVSS	—	A/D 转换器的地，与 Vss 电位相同	—	—
RESET ^注	输入	系统复位输入	—	P34 ^注
X1 ^注	输入	系统时钟振荡的晶体/陶瓷振荡器的连接。 外部时钟输入	输入	P121 ^注
X2 ^注	—	系统时钟振荡的晶体/陶瓷振荡器的连接。	输入	P122 ^注
VDD	—	电源电压	—	—
VSS	—	地	—	—

注 引脚功能的设置方法，请参考 第十八章 选项字节。

注意事项 在复位期间，P121/X1 和 P122/X2 引脚处于下拉状态。

2.2 引脚功能

2.2.1 P00~P03（端口 0）

端口 0 是个 4 位的 I/O 端口。可通过端口模式寄存器 0（PM0）按位设置为输入或输出模式。可通过上拉电阻选择寄存器 0（PU0）设定内部上拉电阻的使用。

2.2.2 P20~P23（端口 2）

端口 2 是个 4 位的 I/O 端口。除了作为 I/O 端口引脚使用外，这些引脚还有向 A/D 转换器输入模拟信号的功能。这些引脚能以 1 位为单位设置成以下的工作模式。

（1） 端口模式

P20~P23 是 4 位 I/O 端口。可通过端口模式寄存器 2（PM2）按位设定为输入或输出模式。另外，可通过上拉电阻选择寄存器 2（PU2）设定内部上拉电阻的使用。

（2） 控制模式

P20~P23 也有作为 A/D 转换器的模拟输入引脚（ANI0 至 ANI3）的功能。当作为模拟输入引脚使用时，请参考 10.6 A/D 转换器注意事项（5） ANI0/P20 ~ ANI3/P23。

2.2.3 P30 ~ P34（端口 3）

P30 ~ P33 组成一个 4 位的 I/O 端口，端口 3。除了作为 I/O 端口引脚使用外，P30 和 P31 引脚还能作为定时器信号的输入/输出引脚使用，也能输入外部中断请求信号。

P34 是一个 1 位仅作输入的端口。该引脚也可以用作 $\overline{\text{RESET}}$ 引脚，并且当电源打开上电时，这是复位功能。

引脚功能的设置方法，请参考 第十八章 选项字节。当 P34 作为输入引脚使用时，通过外部电阻上拉 P34 引脚。

P30 ~ P34 能以 1 位为单位设定为以下的工作模式。

（1） 端口模式

P30 ~ P33 是个 4 位的 I/O 端口。可通过端口模式寄存器 3（PM3）按位设定为输入或输出模式。另外，可通过上拉电阻选择寄存器 3（PU3）设置内部上拉电阻的使用。

P34 是一个仅作输入的端口。

（2） 控制模式

P30 和 P31 引脚能作为定时器信号的输入/输出引脚，也可用于外部中断请求信号的输入。

（a） INTP0 和 INTP2

这是外部中断请求输入引脚，可指定有效沿（上升沿、下降沿，或兼有上升沿和下降沿）。

（b） TI000

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

（c） TI010

该引脚将捕捉触发信号输入到 16 位定时器/计数器 00 的捕捉寄存器（CR000）。

（d） TO00

该引脚从 16 位定时器/计数器 00 输出信号。

2.2.4 P40~P47（端口 4）

P40~P47 是一个 8 位的 I/O 端口。除了作为 I/O 端口引脚使用外，这些引脚还具有输出定时器信号，输入外部中断请求信号和输入/输出串行接口的数据的功能。

这些引脚能以 1 位为单位设置成以下的工作模式。

（1） 端口模式

P40~P47 是一个 8 位的 I/O 端口。端口的每一位都可以通过设置端口模式寄存器 4（PM4）设定为输入或者输出模式。另外，可通过上拉电阻选择寄存器 4（PU4）设定内部上拉电阻的使用。

（2） 控制模式

P41~P44 具有输出内部定时器的信号，输入外部中断请求信号和输入/输出串行接口的数据的功能。

（a） INTP1 和 INTP3

这是外部中断请求输入引脚，可指定有效沿（上升沿、下降沿，或兼有上升沿和下降沿）。

（b） TOH1

8 位定时器 H1 的输出引脚。

（c） TxD6

这个引脚从异步串行接口输出串行数据。

（d） RxD6

这个引脚输入串行数据到异步串行接口。

2.2.5 P120~P123（端口 12）

P120~P123 是一个 4 位的 I/O 端口。

端口的每一位都可以通过设置端口模式寄存器 12（PM12）设定为输入或者输出模式。可通过上拉电阻选择寄存器 12（PU12）设定 P123 使用内部上拉电阻。

P121 和 P122 也能复用为 X1 和 X2 引脚。对复用功能的详细描述，请参考 第十八章 选项字节。

注意事项 在复位期间，P121/X1 和 P122/X2 引脚处于下拉状态。

2.2.6 P130（端口 13）

这是一个只能用作输出的端口。

2.2.7 $\overline{\text{RESET}}$

该引脚输入低电平有效的系统复位信号。上电时，此复位功能不论选项字节如何设置。

2.2.8 X1 和 X2

这些引脚连接振荡器到 X1 输入时钟振荡。

X1 和 X2 也能复用为 P121 和 P122 引脚，对复用功能的详细描述，请参考 第十八章 选项字节。

从 X1 提供外部时钟的输入。

注意事项 在复位期间，P121/X1 和 P122/X2 引脚处于下拉状态。

2.2.9 AV_{REF}

A/D 转换器参考电压输入引脚，也是 P20 ~ P23 和 A/D 转换器的正向供电电压。不使用 A/D 转换器时，该引脚连接到 V_{DD} 。

2.2.10 AV_{SS}

A/D 转换器地电平引脚。即使不使用 A/D 转换器时，该引脚的电平应该始终和 V_{SS} 相等。

2.2.11 V_{DD}

这是电源电压引脚。

2.2.12 V_{SS}

这是地引脚。

<R>

2.3 引脚I/O电路和未使用引脚的连接方式

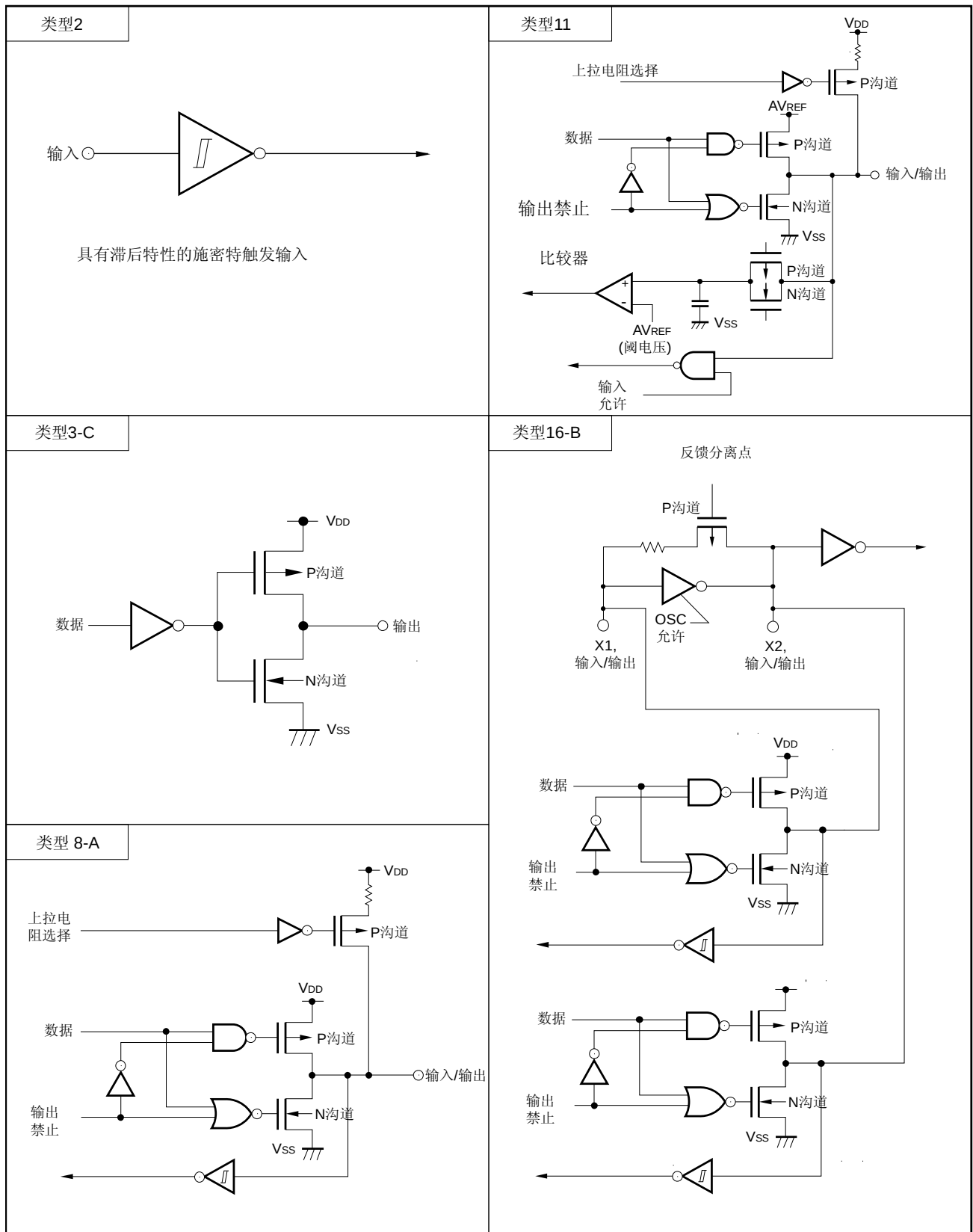
每个引脚的 I/O 电路类型和未使用引脚的连接方式如表 2-1 所示。

每种类型的 I/O 电路的结构如图 2-1 所示。

表 2-1. 引脚 I/O 电路的类型和未使用引脚的连接方式

引脚名称	I/O 电路类型	I/O	未使用引脚的推荐连接方式
P00~P03	8-A	I/O	输入： 通过电阻单独连接到 AV _{REF} 或者 V _{SS} 。 输出： 悬空。
P20/ANI0~P23/ANI3	11		输入： 通过电阻单独连接到 AV _{REF} 或者 V _{SS} 。 输出： 悬空。
P30/TI000/INTP0	8-A		输入： 通过电阻单独连接到 AV _{REF} 或者 V _{SS} 。 输出： 悬空。
P31/TI010/TO00/INTP2			
P32 和 P33			
P34/RESET	2	输入	通过电阻连接到 V _{DD}
P40	8-A	I/O	输入： 通过电阻单独连接到 AV _{REF} 或者 V _{SS} 。 输出： 悬空。
P41/INTP3			
P42/TOH1			
P43/TxD6/INTP1			
P44/RxD6			
P45~P47			
P120			
P121/X1	16-B	I/O	输入： 通过电阻单独连接到 V _{SS} 输出： 悬空
P122/X2			
P123	8-A		输入： 通过电阻单独连接到 V _{DD} 或 V _{SS} 输出： 悬空
P130	3-C	输出	悬空。
AV _{REF}	—	输入	直接连接到 V _{DD} 。
AV _{SS}	—	—	直接连接到 V _{SS} 。

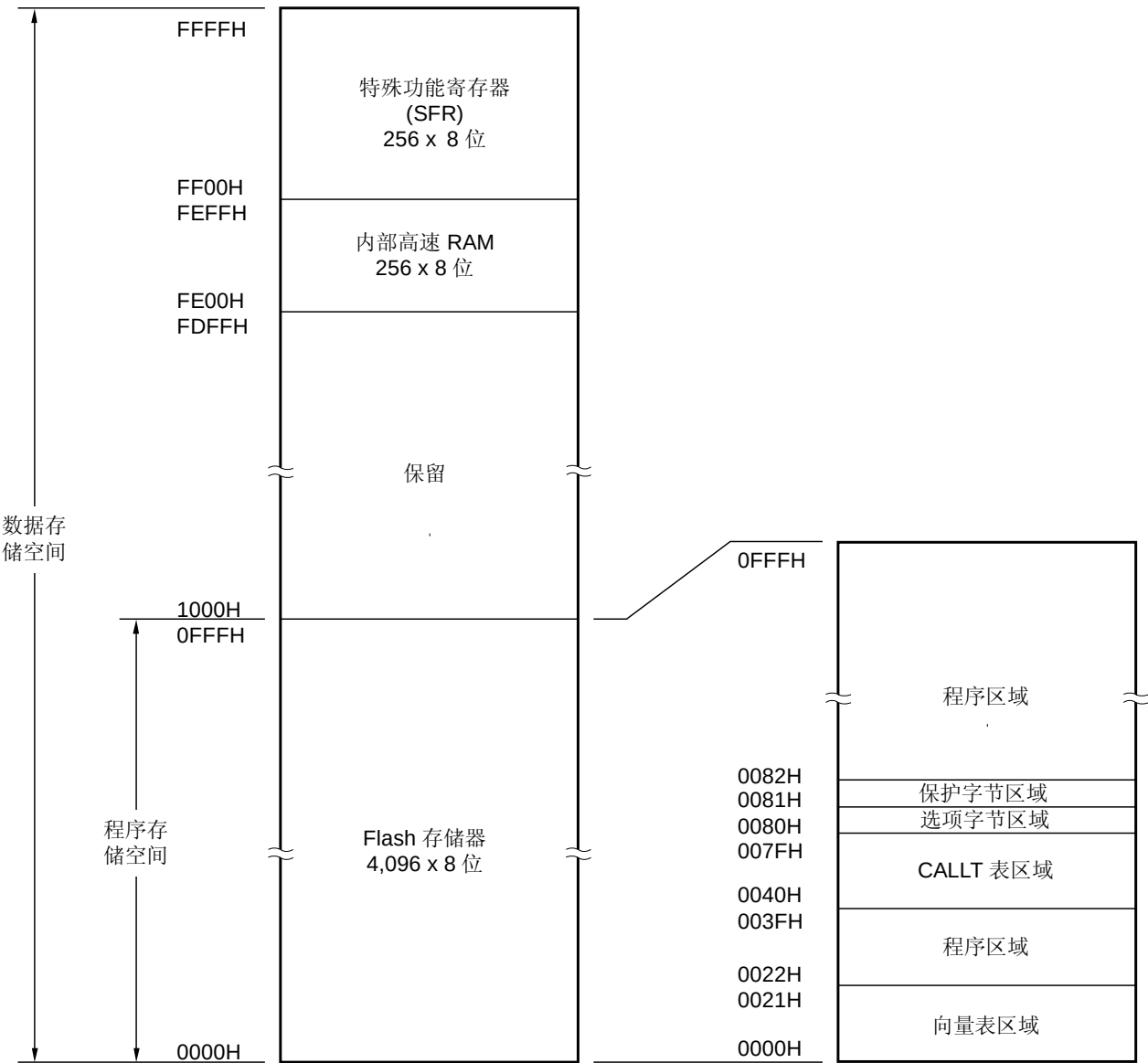
图 2-1. 引脚 I/O 电路



3.1 存储器空间

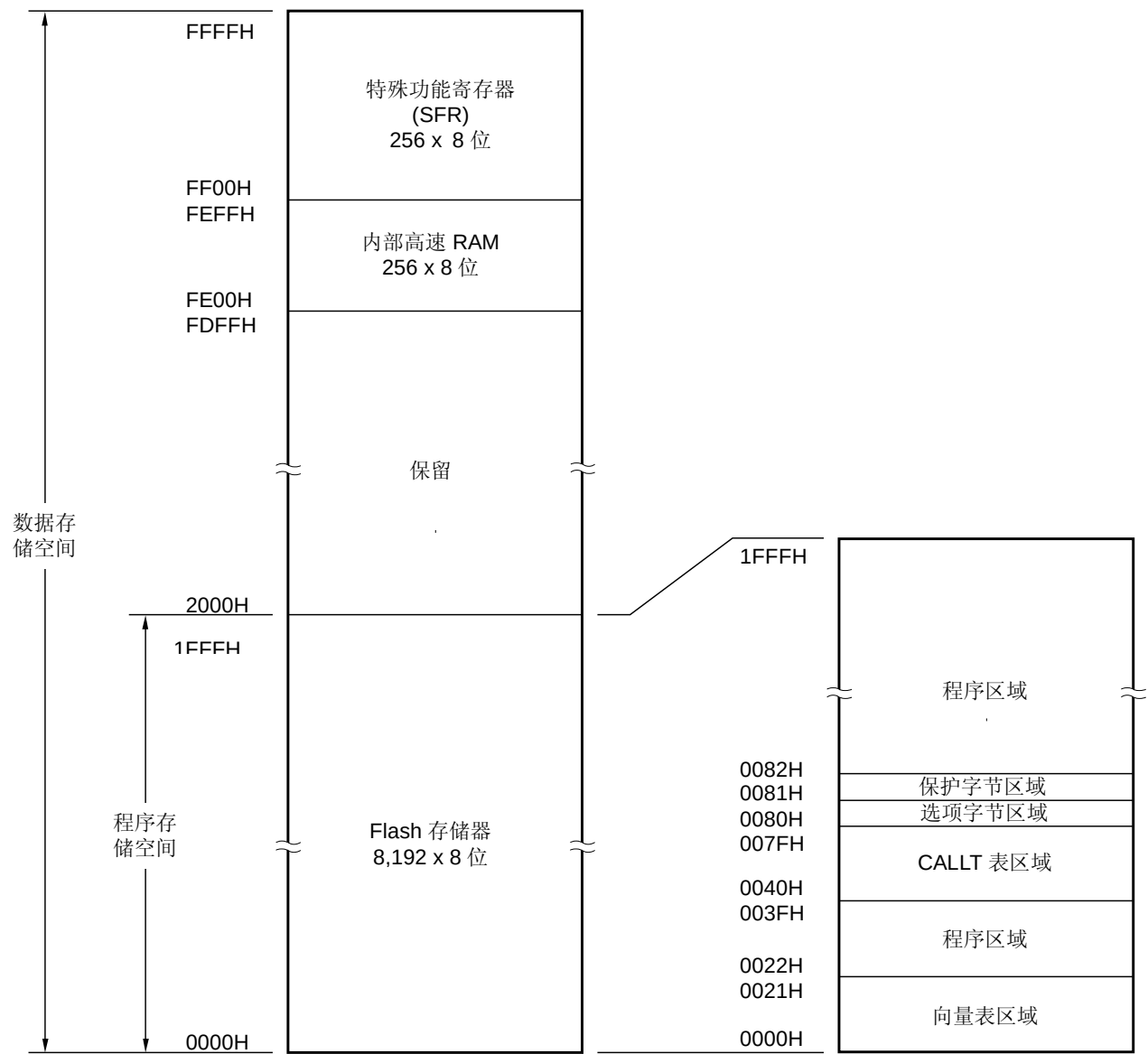
78K0S/KB1+可以访问大小为 64KB 的存储空间。图 3-1 和 3-2 显示了存储空间映射图。

图 3-1. 存储空间映射图 (μPD78F9232)



备注 选项字节和保护字节各为 1 字节。

图 3-2. 存储空间映射图 (μPD78F9234)



备注 选项字节和保护字节各为 1 字节。

3.1.1 内部程序存储空间

内部程序存储空间用于存储程序和表格数据。该空间一般通过程序计数器（PC）来寻址。

78K0S/KB1+产品提供了以下容量的内部 ROM（或 FLASH 存储器）

表 3-1. 内部 ROM 容量

产品型号	内部 ROM	
	结构	容量
μ PD78F9232	Flash 存储器	4,096 $\times$ 8 位
μ PD78F9234		8,192 $\times$ 8 位

内部程序存储空间主要分为以下几个区域。

（1）向量表区域

从地址0000H 到 0021H总共34字节保留作为向量表区域。在向量表中存放的是根据复位信号输入或产生的中断请求进行转移的程序起始地址。对于一个16位地址，低8位存放在偶地址，高8位存放在奇地址。

表 3-2. 向量表

向量表地址	中断请求	向量表地址	中断请求
0000H	Reset	0012H	INTAD
0006H	INTLVI	0016H	INTP2
0008H	INTP0	0018H	INTP3
000AH	INP1	001AH	INTTM80
000CH	INTTMH1	001CH	INTSRE6
000EH	INTTM000	001EH	INTSR6
0010H	INTTM010	0020H	INTST6

注意事项 向量表地址 0014H 不对应任何中断源。

（2）CALLT 指令表区域

0040H 至 007FH 共 64 字节区域，可存放 1 字节调用指令（CALLT）的子程序入口地址。

（3）选项字节区域

选项字节区域是地址 0080H 的 1 字节区域。详情请参见 第十八章 选项字节。

（4）保护字节区域

保护字节区域是地址 0081H 的 1 字节区域。详情请参见 第十九章 FLASH 存储器。

3.1.2 内部数据存储器空间

μ PD78F9232 和 μ PD78F9234 可以提供 256 字节内部高速 RAM。

内部高速 RAM 也可以被用作堆栈空间。

3.1.3 特殊功能寄存器（SFR）区域

片内外围硬件的特殊功能寄存器（SFRs）被分配在 FF00H ~ FFFFH 区域内（参见 表 3-3）。

3.1.4 数据存储寻址

78K0S/KB1+ 提供宽范围的寻址方式使存储器操作更有效率。包括数据存储器和特殊功能寄存器（SFR）的存储区间（FE00H 到 FFFFH）可以通过与其功能一致的唯一寻址模式进行访问。图 3-3 和 3-4 举例说明数据存储空间寻址。

图 3-3. 数据存储寻址（μPD78F9232）

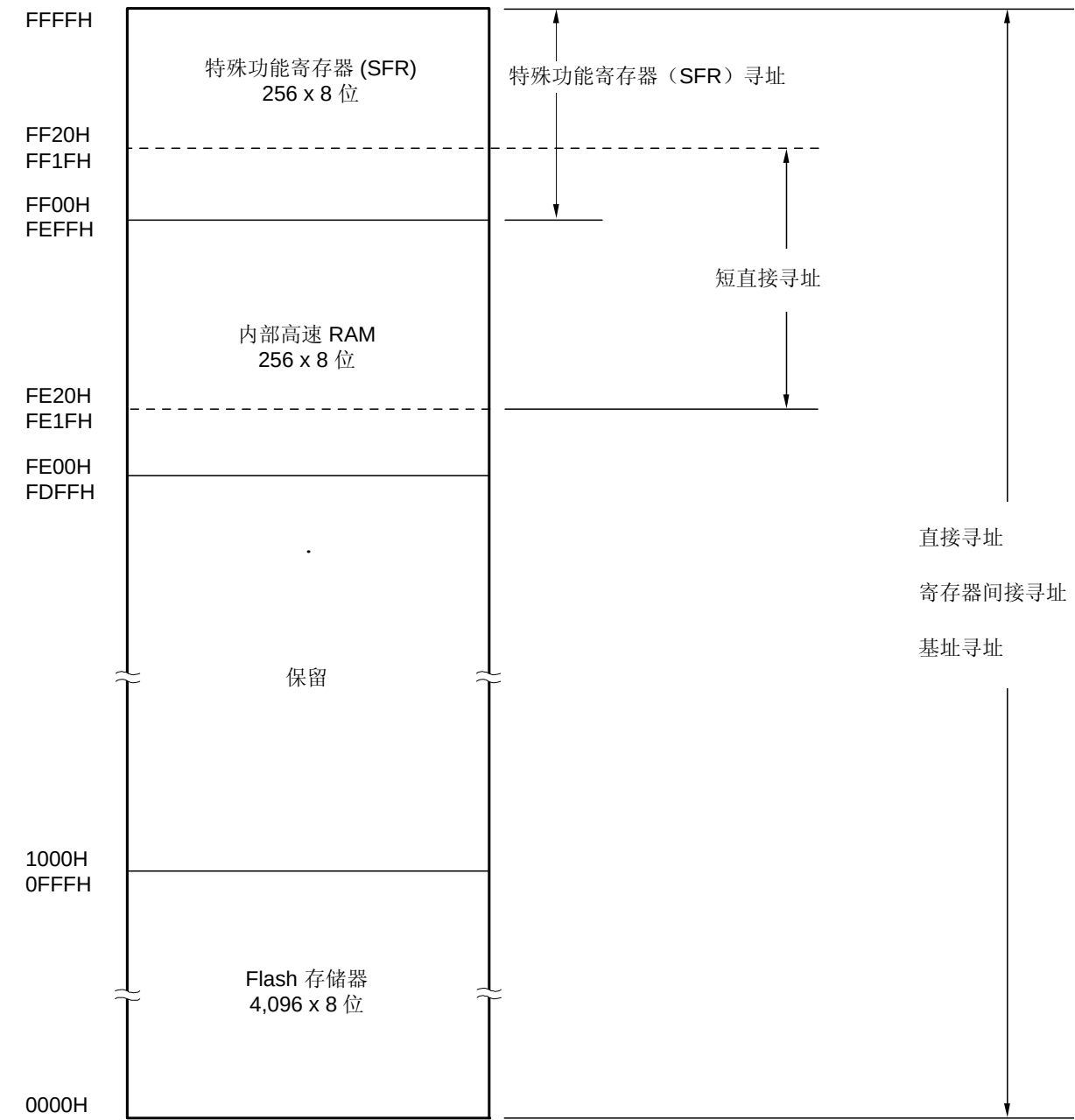
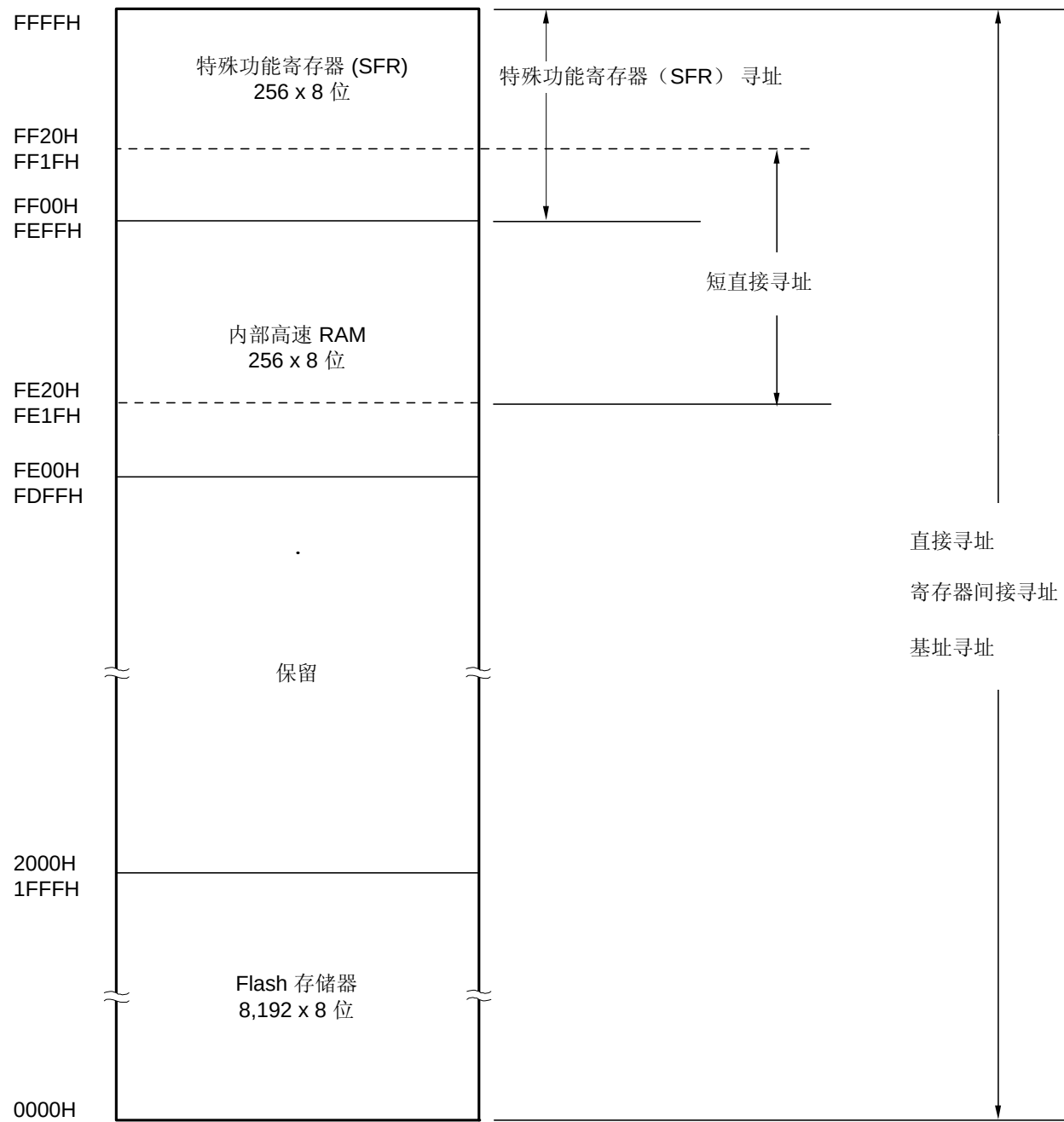


图 3-4. 数据存储器寻址（ μ PD78F9234）



3.2 处理器寄存器

78K0S/KB1+包含以下几种片内处理器寄存器。

3.2.1 控制寄存器

该控制寄存器具有特殊功能，用来控制程序执行顺序的状态和堆栈空间。控制寄存器包括程序计数器(PC)、程序状态字(PSW)和堆栈指针(SP)。

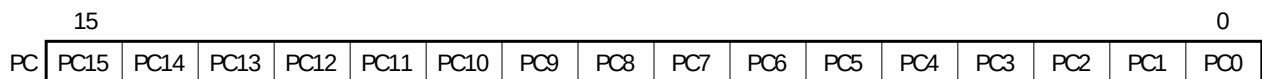
(1) 程序计数器 (PC)

程序计数器是一个 16 位寄存器，用于存放下一条即将执行的程序的地址信息。

在正常操作中，根据获取的指令字节数，程序计数器 (PC) 的值会自动累加。当执行分支跳转指令时，则设置为立即数或寄存器的值。

当产生复位信号时，把复位向量表地址为 0000H 和 0001H 的值赋给程序计数器。

图 3-5. 程序计数器结构



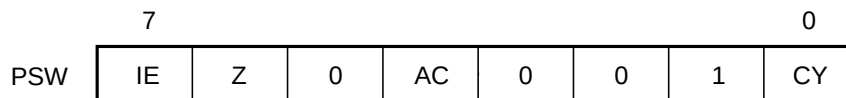
(2) 程序状态字 (PSW)

程序状态字是一个 8 位寄存器，由各种标志位组成，通过指令执行对其进行置位/复位。

根据中断请求的产生或 PUSH PSW 指令执行，程序状态字自动入栈；通过执行 RETI 和 POP PSW 指令，程序状态字自动出栈。

复位信号输入将程序状态字的值设定为 02H。

图 3-6. 程序状态字结构



(a) 中断允许标志 (IE)

该标志用于控制 CPU 响应中断请求的操作。

当 IE = 0，设置为中断禁止 (DI) 状态。所有的中断请求都被禁止。

当 IE = 1，设置为中断允许 (EI) 状态。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，该标志复位为 0；当执行 EI 指令时，该标志置为 1。

(b) 0 标志 (Z)

当操作结果为 0 时，该标志置 1，其它情况下置 0。

(c) 辅助进位标志 (AC)

如果操作结果在第 3 位产生进位或借位，该标志置 1，其它情况下置 0。

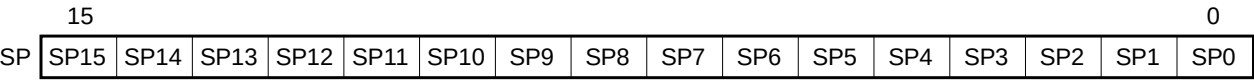
(d) 进位标志 (CY)

该标志存储的是执行加/减指令时产生的上溢或下溢。 它也存储移位指令执行中的移出值，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 位寄存器，用来存放内部堆栈区域的起始地址。只有内部高速 RAM 区才能被设置为堆栈区（除内部高速 RAM 区域外，其他区域不能用作堆栈区域）。

图 3-7. 堆栈指针结构



在向堆栈写（存入）内容时，堆栈指针累减，而从堆栈中读出（恢复）内容时，堆栈指针累加。
每个堆栈的数据存储/恢复操作过程如图 3-8 和 3-9 所示。

- 注意事项
- 1. 由于复位信号的产生会使 SP 的内容不确定，所以在使用堆栈前必须先对 SP 初始化。
 - 2. 堆栈指针只能设置在高速 RAM 区域，并且只有低 10 位可以真正设置。
因此，如果堆栈指针被指定到 0FF00H，它要转换成高速 RAM 区域的 0FB00H，因为 0FF00H 在 SFR 区域内，并且不是高速 RAM 区域。
当值被真正压进栈时，0FB00H 减 1 变成 0FAFFH，但是因为该值不在高速 RAM 区域，它被转换成 0FEFFH，与当堆栈指针设置到 0FF00H 时的值相同。

图 3-8. 将数据存入堆栈

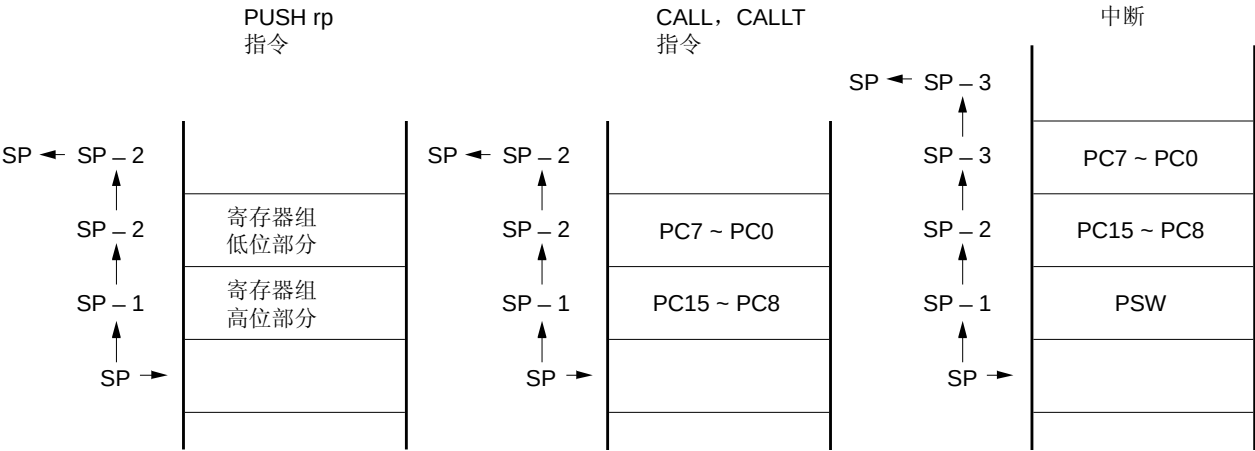
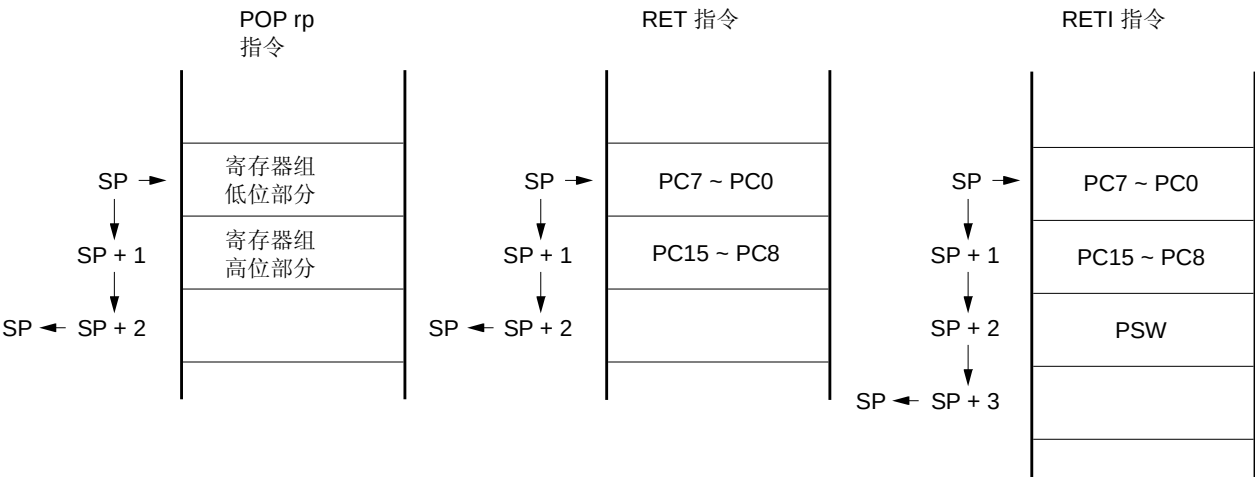


图 3-9. 从堆栈恢复数据



3.2.2 通用寄存器

通用寄存器包含了 8 个 8 位寄存器（X, A, C, B, E, D, L 和 H）。

此外，每个寄存器都可以当作一个 8 位寄存器使用，两个成对的 8 位寄存器可以当作一个 16 位寄存器使用（AX, BC, DE 和 HL）。

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

图 3-10. 通用寄存器结构（1/2）

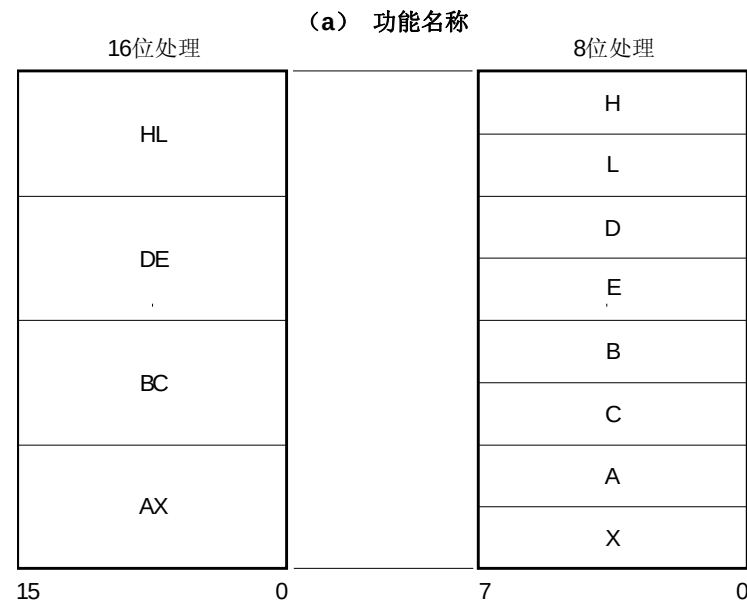
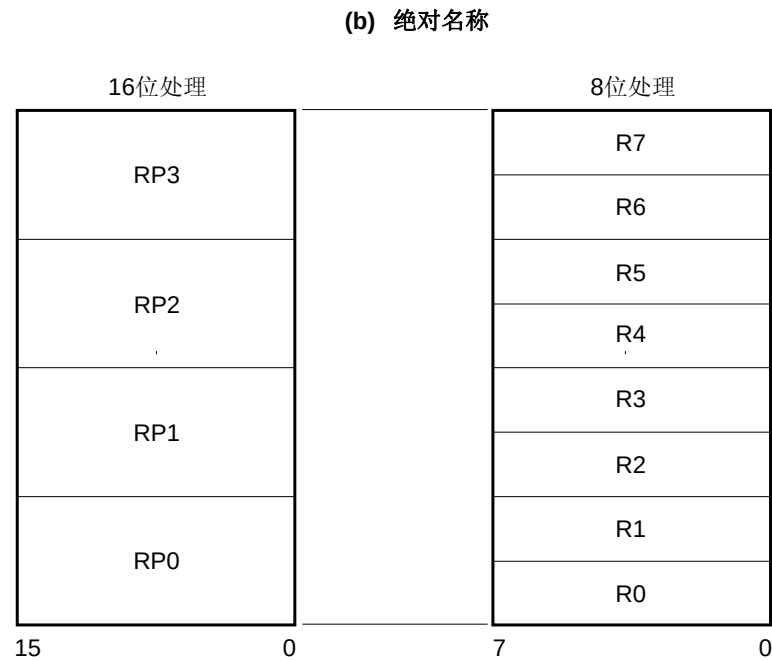


图 3-10. 通用寄存器结构（2/2）



3.2.3 特殊功能寄存器（SFRs）

与通用寄存器不同，每一个特殊功能寄存器都有其特定的功能。

特殊功能寄存器分配在 FF00H 到 FFFFH 的 256 字节。

特殊功能寄存器可以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，可操作的单元也不同（1 位、8 位和 16 位）。

每个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。该操作也可以由地址和位来指定。
- 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。该操作也可以由一个地址来指定。
- 16 位操作
16 位操作指令的操作数被描述为汇编程序的保留符号。当指定地址时，必须使用偶地址。

表 3-3 列出了特殊功能寄存器列表。表中术语符号的含义如下：

- 符号
表示特殊功能寄存器的地址，在 RA78K0S 中定义为保留字，并在 CC78K0S 中通过 #pragma sfr 伪指令来定义为 sfr 变量。于是，如果使用汇编器或集成调试器时，这些符号可以被用作指令操作数。
- 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
- 可同时位操作的数量
特殊功能寄存器可以位操作的单位（1，8，和 16）。
- 复位后
表示复位信号产生后，特殊功能寄存器的状态。

表 3-3. 特殊功能寄存器(1/4)

地址	符号	位号								R/W	可同时操作的位单元数目			复位后	参考页
		7	6	5	4	3	2	1	0		1	8	16		
FF00H	P0	0	0	0	0	P03	P02	P01	P00	R/W 注 1	√	√	–	00H	68
FF01H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF02H	P2	0	0	0	0	P23	P22	P21	P20	R/W 注 1	√	√	–	00H	68
FF03H	P3	0	0	0	P34	P33	P32	P31	P30		√	√	–	00H	68
FF04H	P4	P47	P46	P45	P44	P43	P42	P41	P40		√	√	–	00H	68
FF05H to FF0BH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF0CH	P12	0	0	0	0	P123	P122	P121	P120	R/W 注 1	√	√	–	00H	68
FF0DH	P13	0	0	0	0	0	0	0	P130		√	√	–	00H	68
FF0EH	CMP01	–	–	–	–	–	–	–	–	R/W	–	√	–	00H	138
FF0FH	CMP11	–	–	–	–	–	–	–	–		–	√	–	00H	138
FF10H	MUL 0L	MUL LO	–	–	–	–	–	–	–	R	–	√	√	不确定	220
FF11H	MUL 0H		–	–	–	–	–	–	–		–	√	√		220
FF12H	TM00	–	–	–	–	–	–	–	–		–	–	√注 2	0000H	90
FF13H		–	–	–	–	–	–	–	–	–	–	–	√注 2		
FF14H	CR000	–	–	–	–	–	–	–	–	R/W	–	–	√注 2	0000H	90
FF15H		–	–	–	–	–	–	–	–		–	–	√注 2		
FF16H	CR010	–	–	–	–	–	–	–	–		–	–	√注 2	0000H	92
FF17H		–	–	–	–	–	–	–	–	–	–	–	√注 2		
FF18H	ADCR	–	–	–	–	–	–	–	–	R	–	–	√注 2	不确定	170
FF19H		0	0	0	0	0	0	–	–		–	–	√注 2		
FF1AH	ADCRH	–	–	–	–	–	–	–	–		–	√	–		171
FF1BH to FF21H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF20H	PM0	1	1	1	1	PM03	PM02	PM01	PM00	R/W	√	√	–	FFH	67
FF21H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF22H	PM2	1	1	1	1	PM23	PM22	PM21	PM20	R/W	√	√	–	FFH	67, 171
FF23H	PM3	1	1	1	1	PM33	PM32	PM31	PM30		√	√	–	FFH	67, 98
FF24H	PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40		√	√	–	FFH	67, 141, 197
FF25H to FF2BH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF2CH	PM12	1	1	1	1	PM123	PM122	PM121	PM120	R/W	√	√	–	FFH	67
FF2DH to FF2FH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF30H	PU0	0	0	0	0	PU03	PU02	PU01	PU00	R/W	√	√	–	00H	70
FF31H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF32H	PU2	0	0	0	0	PU23	PU22	PU21	PU20	R/W	√	√	–	00H	70
FF33H	PU3	0	0	0	0	PU33	PU32	PU31	PU30		√	√	–	00H	70

- 注
1. P34 只能作为输入端口。
 2. 仅能通过短直接寻址方式进行 16 位访问。

<R>

表 3-3. 特殊功能寄存器(2/4)

地址	符号	位号								R/W	可同时操作的位单元数目			复位后	参考页
		7	6	5	4	3	2	1	0		1	8	16		
FF34H	PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	R/W	√	√	—	00H	70
FF35H to FF3BH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF3CH	PU12	0	0	0	0	PU123	0	0	PU120	R/W	√	√	—	00H	70
FF3DH to FF47H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF48H	WDTM	0	1	1	WDCS 4	WDCS 3	WDCS 2	WDCS 1	WDCS 0	R/W	—	√	—	67H	154
FF49H	WDTE	—	—	—	—	—	—	—	—		—	√	—	9AH	155
FF50H	LVIM	<LVI ON>	0	0	0	0	0	<LVI MD>	<LVI F>		√	√	—	00H 注 1	261
FF51H	LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0		—	√	—	00H 注 1	262
FF52H, FF53H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF54H	RESF	0	0	0	WDT RF	0	0	0	LVIRF	R	—	√	—	00H 注 2	255
FF55H to FF57H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF58H	LSRCM	0	0	0	0	0	0	0	<LSR STOP>	R/W	√	√	—	00H	76
FF59H to FF5FH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF60H	TMC00	0	0	0	0	TMC 003	TMC 002	TMC 001	<OVF 00>	R/W	√	√	—	00H	93
FF61H	PRM00	ES110	ES100	ES010	ES000	0	0	PRM 001	PRM 000		√	√	—	00H	97
FF62H	CRC00	0	0	0	0	0	CRC 002	CRC 001	CRC 000		√	√	—	00H	95
FF63H	TOC00	0	<OSPT 00>	<OSPE 00>	TOC 004	<LVS 00>	<LVR 00>	TOC 001	<TOE 00>		√	√	—	00H	96
FF64H to FF6FH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF70H	TMHMD 1	<TMH E1>	CKS12	CKS11	CKS10	TMMD 11	TMMD 10	<TOLE V1>	<TOEN 1>	R/W	√	√	—	00H	139
FF71H to FF7FH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF80H	ADM	<ADCS>	0	FR2	FR1	FR0	0	0	<ADCE>	R/W	√	√	—	00H	168
FF81H	ADS	0	0	0	0	0	0	ADS1	ADS0		√	√	—	00H	170
FF82H, FF83H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

注 1. 只有在 LVI 引起的复位后才能保持。
2. 随复位原因的不同而有所差异。

备注 对于用括号(<>)表示的位名称, 该位名称在 RA78K0 中定义为保留字节, 并定义为 sfr 变量, 在 CC78K0 中通过 #pragma sfr 指令来使用。

<R>

表 3-3. 特殊功能寄存器(3/4)

地址	符号	位号								R/W	可同时操作的位单元数目			复位后	参考页
		7	6	5	4	3	2	1	0		1	8	16		
FF84H	PMC2	0	0	0	0	PMC23	PMC22	PMC21	PMC20	R/W	√	√	—	00H	69, 171
FF85H to FF8BH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF8CH	ISC	0	0	0	0	0	0	ISC1	ISC0	R/W	√	√	—	00H	197
FF8DH to FF8FH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF90H	ASIM6	<POWER6>	<TXE6>	<RXE6>	PS61	PS60	CL6	SL6	ISRM6	R/W	√	√	—	01H	189
FF91H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FF92H	RXB6	—	—	—	—	—	—	—	—	R	—	√	—	FFH	188
FF93H	ASIS6	0	0	0	0	0	PE6	FE6	OVE6		—	√	—	00H	191
FF94H	TXB6	—	—	—	—	—	—	—	—	R/W	—	√	—	FFH	188
FF95H	ASIF6	0	0	0	0	0	0	TXBF6	TXSF6	R	—	√	—	00H	192
FF96H	CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60	R/W	—	√	—	00H	193
FF97H	BRGC6	MLD67	MLD66	MLD65	MLD64	MLD63	MLD62	MLD61	MLD60		—	√	—	FFH	194
FF98H	ASICL6	<SBRF6>	<SBRT6>	SBT6	SBL62	SBL61	SBL60	DIR6	TXDLV6		√	√	—	16H	195
FF99H to FF9FH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FFA0H	PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0	W	—	√	—	不确定	287
FFA1H	PFS	0	0	0	0	0	WEPRERR	VCERR	FPRERR	R/W	√	√	—	00H	287
FFA2H	FLPMC	0	PRSEL F4	PRSEL F3	PRSEL F2	PRSEL F1	PRSEL F0	0	FLSPM		—	√	—	不确定	286
FFA3H	FLCMD	0	0	0	0	0	FLCMD 2	FLCMD 1	FLCMD 0		√	√	—	00H	289
FFA4H	FLAPL	FLAP7	FLAP6	FLAP5	FLAP4	FLAP3	FLAP2	FLAP1	FLAP0		√	√	—	不确定	290
FFA5H	FLAPH	0	0	0	FLA P12	FLA P11	FLA P10	FLA P9	FLA P8		√	√	—	290	290
FFA6H	FLAPHC	0	0	0	FLAP C12	FLAP C11	FLAP C10	FLAP C9	FLAP C8		√	√	—	00H	290
FFA7H	FLAPLC	FLAP C7	FLAP C6	FLAP C5	FLAP C4	FLAP C3	FLAP C2	FLAP C1	FLAP C0		√	√	—	290	290
FFA8H	FLW	FLW7	FLW6	FLW5	FLW4	FLW3	FLW2	FLW1	FLW0		—	√	—	291	291
FFA9H to FFCBH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FFCCH	TMC80	<TCE80>	0	0	0	0	TCL 801	TCL 800	0	R/W	√	√	—	00H	132
FFCDH	CR80	—	—	—	—	—	—	—	—	W	—	√	—	不确定	131
FFCEH	TM80	—	—	—	—	—	—	—	—	R	—	√	—	00H	131
FFCFH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

备注 对于用括号(<>)表示的位名称, 该位名称在 RA78K0 中定义为保留字节, 并定义为 sfr 变量, 在 CC78K0 中通过 #pragma sfr 指令来使用。

<R>

表 3-3. 特殊功能寄存器(4/4)

地址	符号	位号								R/W	可同时操作的位单元数目			复位后	参考页
		7	6	5	4	3	2	1	0		1	8	16		
FFD0H	MRA0	—	—	—	—	—	—	—	—	W	—	√	—	不确定	220
FFD1H	MRB0	—	—	—	—	—	—	—	—		—	√	—		220
FFD2H	MULC0	0	0	0	0	0	0	0	<MULST0>	R/W	√	√	—	00H	222
FFD3H to FFD7H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FFE0H	IF0	<ADIF>	<TMIF010>	<TMIF000>	<TMIFH1>	<PIF1>	<PIF0>	<LVIF>	0	R/W	√	√	—	00H	229
FFE1H	IF1	0	<STIF6>	<SRIF6>	<SREIF6>	<TMIF80>	<PIF3>	<PIF2>	0		√	√	—	00H	229
FFE2H, FFE3H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FFE4H	MK0	<ADMK>	<TMMK010>	<TMMK000>	<TMMKH1>	<PMK1>	<PMK0>	<LVIMK>	1	R/W	√	√	—	FFH	230
FFE5H	MK1	1	<STMK6>	<SRMK6>	<SREMK6>	<TMMK80>	<PMK3>	<PMK2>	1		√	√	—	FFH	230
FFE6H to FFE7H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FFE8H	INTM0	ES21	ES20	ES11	ES10	ES01	ES00	0	0	R/W	—	√	—	00H	231
FFEDH	INTM1	0	0	0	0	0	0	ES31	ES30		—	√	—	00H	232
FFEEH to FFF2H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FFF3H	PPCC	0	0	0	0	0	0	PPCC1	PPCC0	R/W	√	√	—	02H	75
FFF4H	OSTS	0	0	0	0	0	0	OSTS1	OSTS0		—	√	—	不确定注	77, 240
FFF5H to FFFAH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FFFBH	PCC	0	0	0	0	0	0	PCC1	0	R/W	√	√	—	02H	75
FFFBH to FFFFH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

注 复位释放后振荡器稳定时间通过选项字节来选择。详细内容，请参考第十八章 选项字节。

备注 对于用括号(<>)表示的位名称，该位名称在 RA78K0 中定义为保留字节，并定义为 sfr 变量，在 CC78K0 中通过 #pragma sfr 指令来使用。

3.3 指令地址寻址

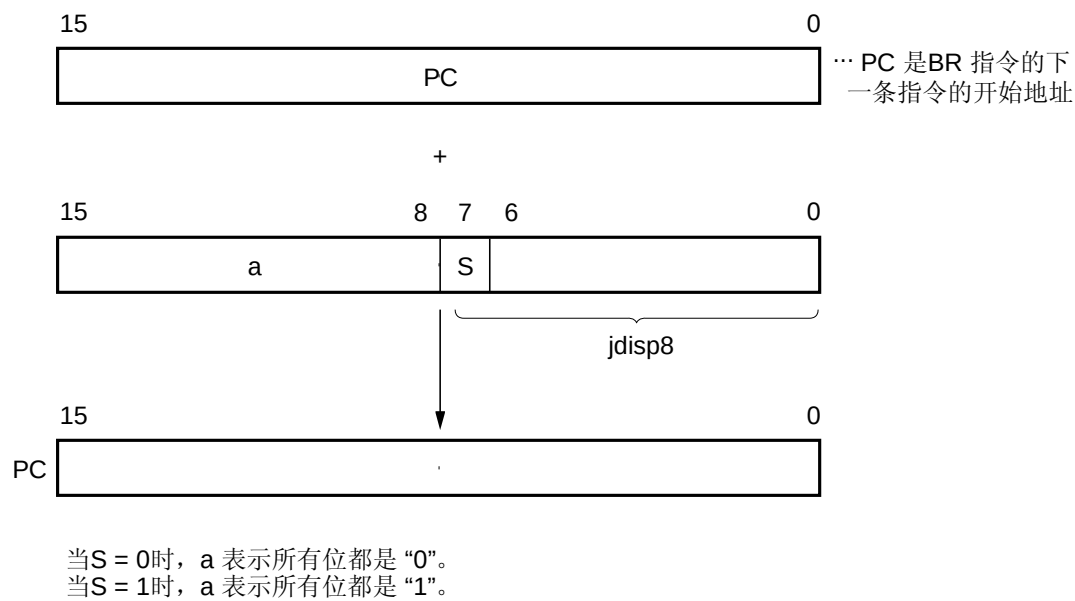
一条指令的地址是由程序计数器（PC）的内容决定的。根据执行指令时所获取的下一条指令的字节数，程序计数器（PC）的内容自动增加（每字节+1）。在执行分支指令时，将程序计数器（PC）的内容设置为分支的目的地址，并按以下方式确定地址。（想要了解每条指令的详细信息，请参考 **78K0S 系列指令用户手册（U11047E）**）。

3.3.1 相对寻址

[功能]

将一条指令代码的 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。这个偏移量是带符号位的补码（-128 ~ +127），其中第 7 位是符号位。换句话说，在相对寻址中，分支跳转的范围是从下一条指令起始地址的-128 到+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]



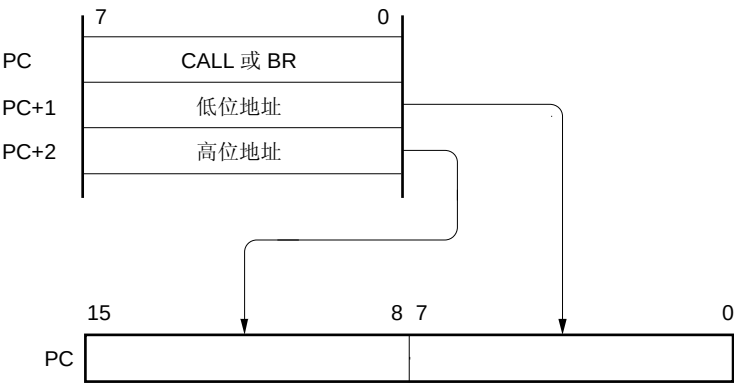
3.3.2 立即寻址

[功能]

将指令字中的立即数赋给程序计数器（PC），然后跳转到该地址。
在执行 CALL !addr16 指令和 BR !addr16 指令时，实现此功能。”
CALL !addr16 和 BR !addr16 指令可以跳转的地址范围是所有内存空间。”

[图示]

CALL !addr16 指令和 BR !addr16 指令

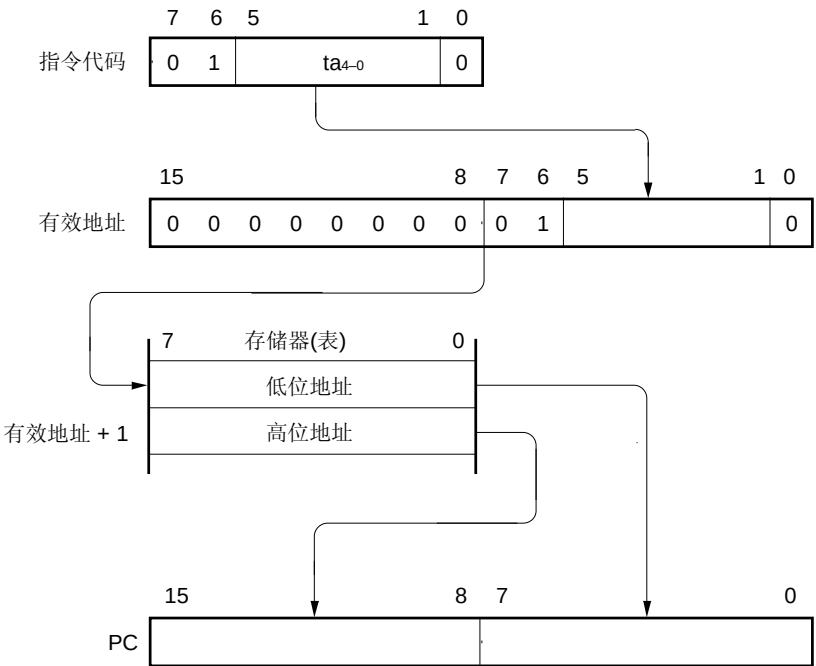


3.3.3 表间接寻址

[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的 40H 至 7FH 之间所存储的地址，跳转地址范围可以是整个存储器空间。

[图示]

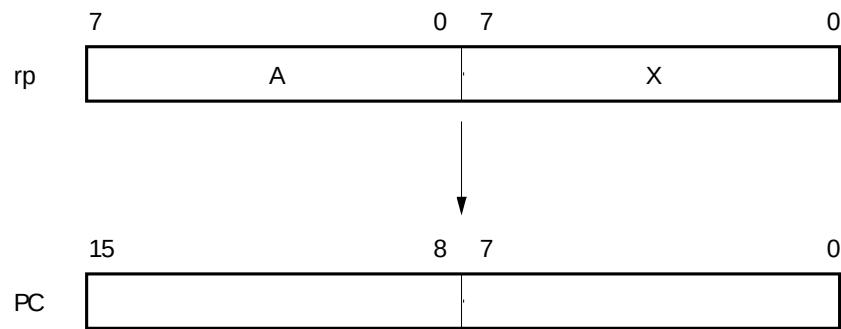


3.3.4 寄存器寻址

[功能]

将一条指令的寄存器对（AX）的内容赋值给程序计数器（PC），然后跳转到该地址。
当执行 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）用来指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]

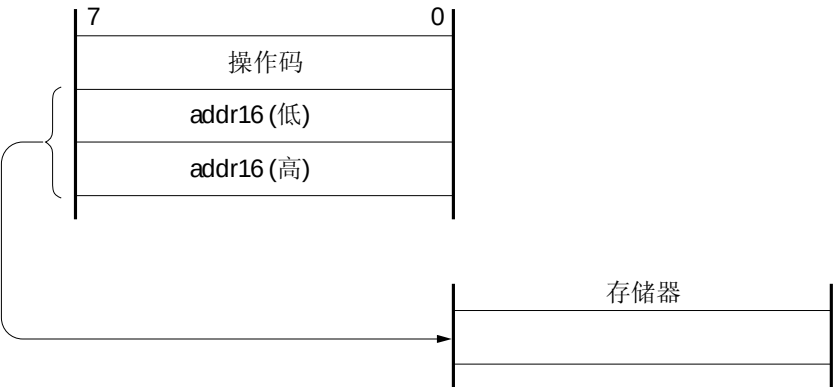
标识符	描述
addr16	标号或 16 位立即数

[描述举例]

MOV A, !0FE80H; 将!addr16 设置为 FE80H

指令码	<table><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr></table>	0	0	1	0	1	0	0	1	操作码
0	0	1	0	1	0	0	1			
	<table><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	1	0	0	0	0	0	0	0	80H
1	0	0	0	0	0	0	0			
	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	1	1	1	1	1	1	1	0	FEH
1	1	1	1	1	1	1	0			

[图示]



3.4.2 短直接寻址

[功能]

用指令中的 8 位立即数直接对内存中的固定操作区域寻址。

该方式的寻址范围，是 FE20H 到 FF1FH（FE20H ~ FEFFH（内部高速 RAM）+ FF00H ~ FF1FH（特殊功能寄存器））这 256 字节的固定空间。

采取短直接寻址的特殊功能寄存器（SFR）区域（FF00H 到 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和定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 20H 到 FFH 之间时，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 到 1FH 之间，则把一个有效地址的第 8 位设置为 1。参见下面的[图示]。

[操作数格式]

标识符	描述
saddr	标号或 FE20H 至 FF1FH 的立即数
saddrp	标号或 FE20H 至 FF1FH 的立即数（仅使用偶地址）

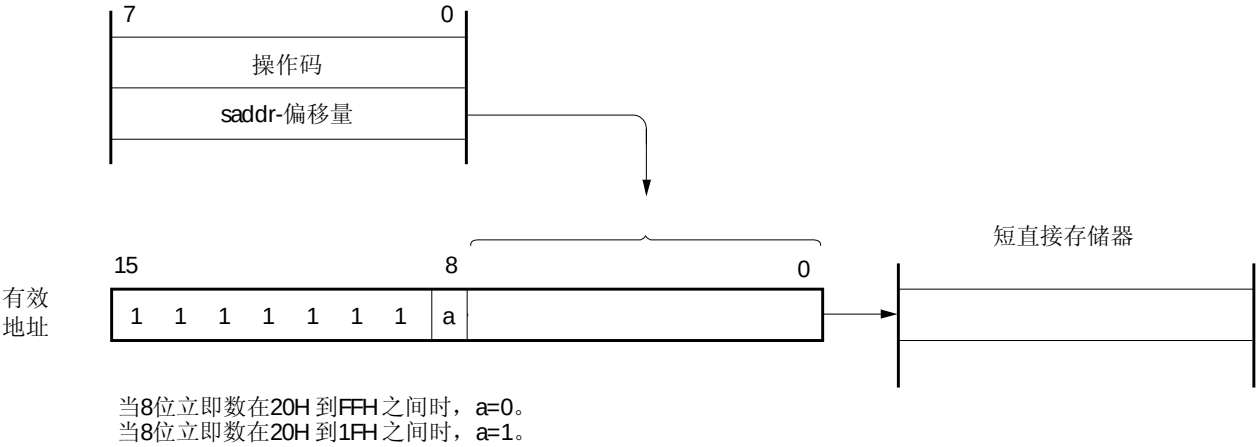
[描述举例]

EQU DATA1 0FE90H ； DATA1 指定为 FE90H 的 saddr 区域；

MOV DATA1, #50H ； 把立即数 50H 赋给 DATA1 单元；

指令码	1 1 1 1 0 1 0 1	操作码
	1 0 0 1 0 0 0 0	90H（saddr-偏移量）
	0 1 0 1 0 0 0 0	50H（立即数）

[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 位立即数对内存中的特殊功能寄存器（SFR）进行寻址。
这种寻址方式适用于 FF00H 到 FFFFH 共 256 字节的空间。但是，访问 SFR 映射在 FF00H 到 FF1FH 的区域需要使用短直接寻址方式。

[操作数格式]

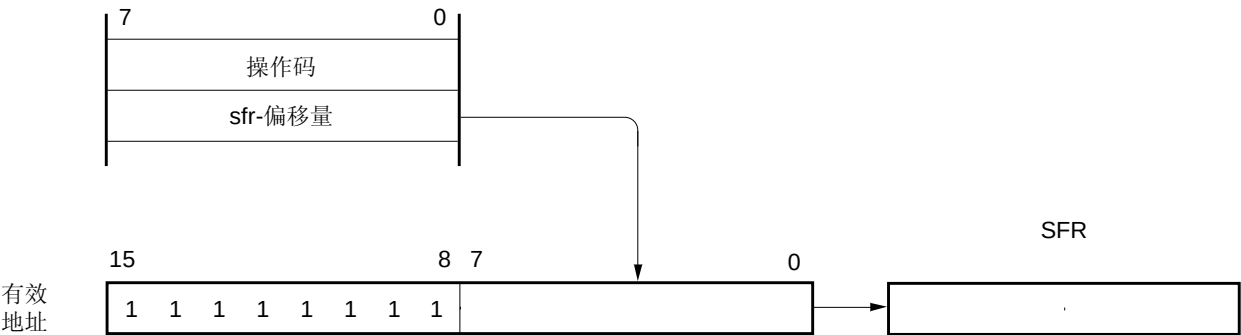
标识符	描述
sfr	特殊功能寄存器名

[描述举例]

MOV PM0, A; 当选择 PM0 为 sfr 时

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]

通用寄存器可以作为操作数被访问。
由指令代码中的寄存器标识码和功能名称来指定需要访问的通用寄存器。
当具有下列操作数格式的指令执行时，采用寄存器寻址方式。如果使用 8 位寄存器，则指令代码中有 3 位用来表示一个 8 位寄存器。

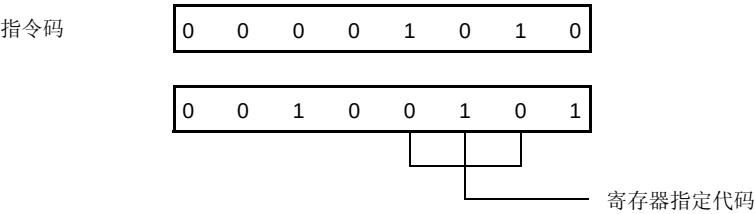
[操作数格式]

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

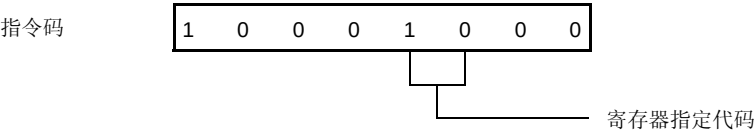
'r' 和 'rp' 可用绝对名称（R0 ~ R7 和 RP0 ~ RP3）或者功能名称（X, A, C, B, E, D, L, H, AX, BC, DE, 和 HL）来描述。

[描述举例]

MOV A, C: 选择寄存器 C 作为 r



INCW DE: 选择寄存器组 DE 作为 rp



3.4.5 寄存器间接寻址

[功能]

通过把寄存器对中的内容作为操作数来访问存储器。由指令中的寄存器组标识码来指定要访问的寄存器对。这种寻址方式可以在整个内存空间进行。

[操作数格式]

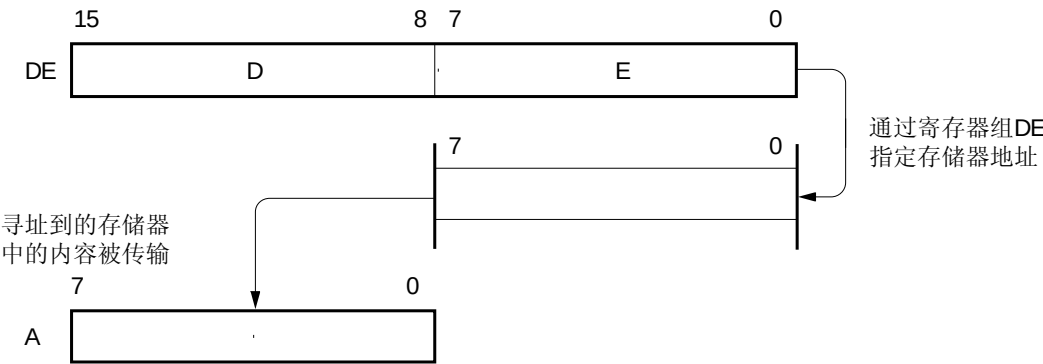
标识符	描述
-	[DE], [HL]

[描述举例]

MOV A, [DE]; 选择寄存器组[DE]时

指令码	0 0 1 0 1 0 1 1
-----	-----------------

[图示]



3.4.6 基址寻址

[功能]

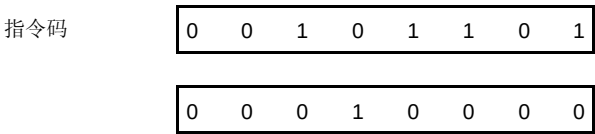
将 8 位立即数加到寄存器对 HL 中，HL 寄存器组作为基地址寄存器，根据相加的和去寻址。通过将偏移量扩展为 16 位正数，完成加法操作。进位忽略不计。这种寻址方式可以在整个内存空间进行。

[操作数格式]

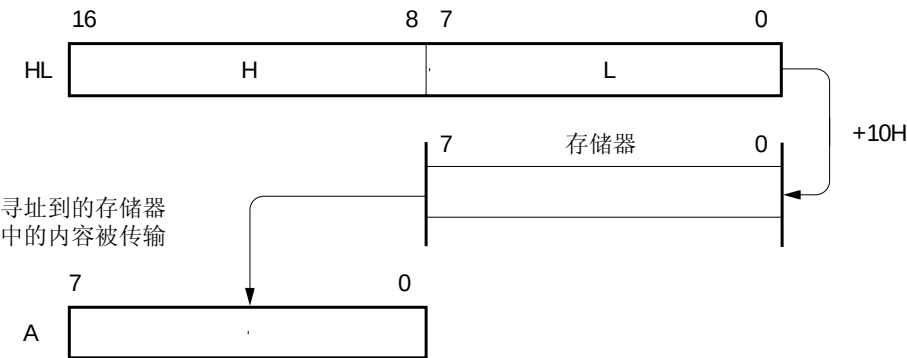
标识符	描述
—	[HL+byte]

[描述举例]

MOV A, [HL+10H]；设定字节的值为 10H



[图示]



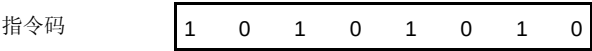
3.4.7 堆栈寻址

[功能]

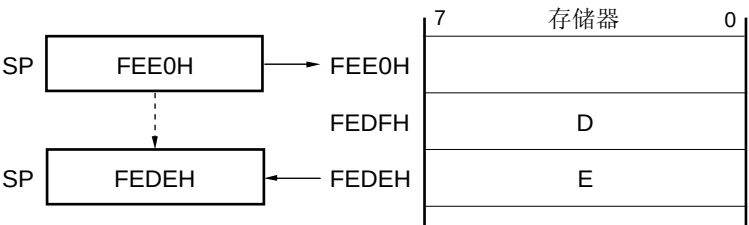
根据堆栈指针（SP）的内容对堆栈区域进行间接寻址。
当执行 **PUSH**、**POP**、子程序调用和 **RETURN** 指令时，或者根据中断请求产生对寄存器进行保存/恢复时，将自动使用这种寻址方式。
该方式仅能对内部高速 **RAM** 区域进行寻址。

[描述举例]

以 **PUSH DE** 指令为例



[图示]



第四章 端口功能

4.1 端口功能

78K0S/KB1+ 产品中提供的端口如图 4-1 所示，可以用于不同的控制操作。每个端口的功能如表 4-1 所示。除了作为数字 I/O 端口外，这些端口都具有复用功能。需要了解详细的复用功能，请参阅第二章 引脚功能。

图 4-1 端口功能

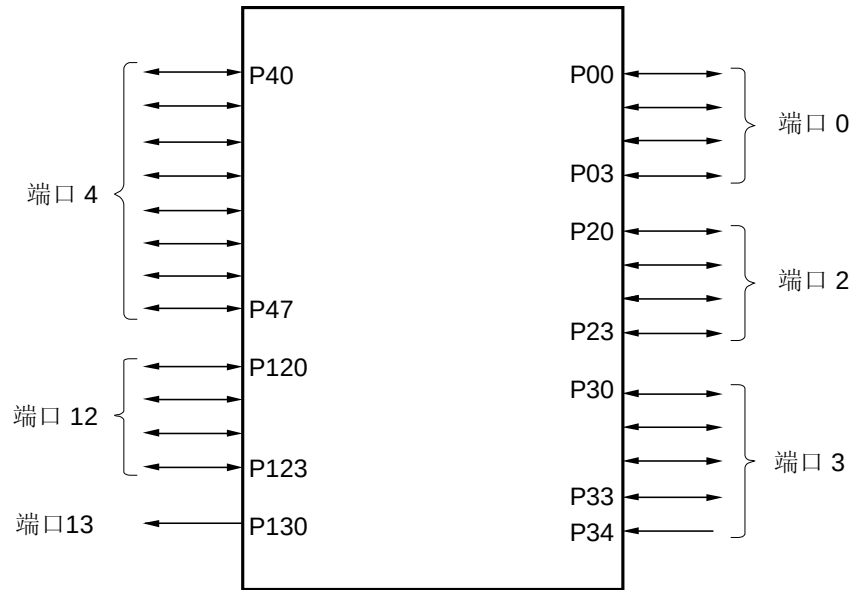


表 4-1 端口功能

引脚名称	I/O	功能		复位后	复用功能引脚
P00～P03	I/O	端口 0。 4 位 I/O 端口。 可以按位设定为输入或输出模式。 可以通过软件设置内部上拉电阻的连接。		输入	—
P20～P23	I/O	端口 2。 4 位 I/O 端口。 可以按位设定为输入或输出模式。 可以通过软件设置内部上拉电阻的连接。		输入	ANI0 ～ ANI3
P30	I/O	端口 3	可以按位设定为输入或输出模式。 可以通过软件设置内部上拉电阻的连接。	输入	TI000/INTP0
P31					TI010/TO00/ INTP2
P32					—
P33					—
P34 [※]	输入		只能用作输入	输入	RESET [※]
P40	I/O	端口 4。 8 位 I/O 端口。 可以按位设定为输入或输出模式。 可以通过软件设置内部上拉电阻的连接。		输入	—
P41					INTP3
P42					TOH1
P43					TxD6/INTP1
P44					RxD6
P45					—
P46					—
P47					—
P120	I/O	端口 12。 4 位 I/O 端口。 可以按位设定为输入或输出模式。 只有 P120 和 P123 可以通过软件设置连接内部上拉电阻。		输入	—
P121 [※]					X1 [※]
P122 [※]					X2 [※]
P123					—
P130	输出	端口 13 1 位输出端口		输出	—

注 关于复用功能的设置，请参考 第十八章 选项字节。

注意事项 在复位期间，P121/X1 和 P122/X2 引脚为下拉状态。

备注

1. 当系统时钟采用高速内部振荡器时，P121 和 P122 引脚可以分配它用。
2. 当系统时钟采用外部时钟信号输入时，P122 引脚可以分配它用。

4.2 端口配置

端口包括下列硬件单元

表 4-2. 端口配置

项目	配置
控制寄存器	端口模式寄存器 (PM0, PM2, PM3, PM4, PM12) 端口寄存器 (P0, P2, P3, P4, P12, P13) 端口模式控制寄存器 2 (PMC2) 上拉电阻选择寄存器 (PU0, PU2, PU3, PU4, PU12)
端口	总共: 26 (CMOS I/O: 24, CMOS 输入: 1, CMOS 输出: 1)
上拉电阻	总共: 22

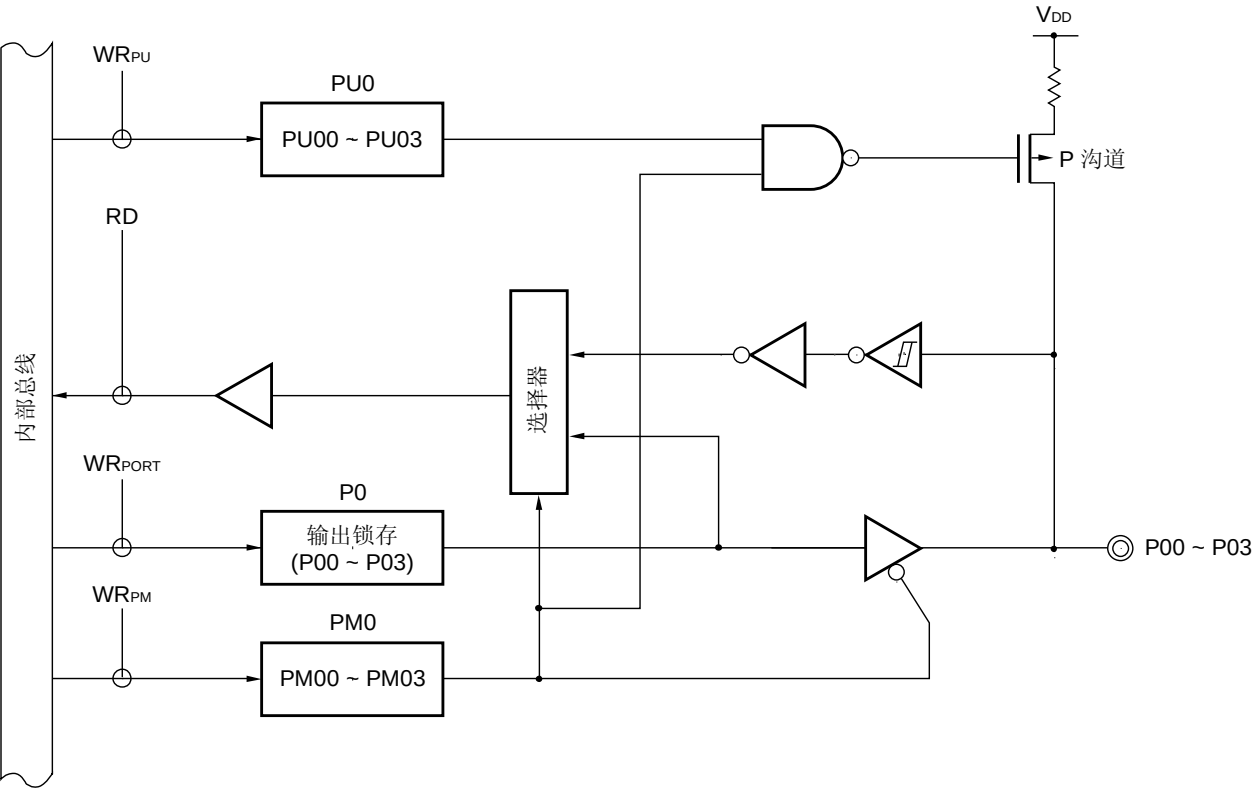
4.2.1 端口 0

端口 0 是具有输出锁存的 4 位 I/O 端口。端口的每一位都可以通过端口模式寄存器 0（PM0）设定为输入或输出模式。当用作输入端口时，可通过上拉电阻选择寄存器 0（PU0）设置内部上拉电阻的使用。

复位后端口 0 为输入模式。

图 4-2 是端口 0 的框图。

图 4-2. P00 至 P03 的框图



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

4.2.2 端口 2

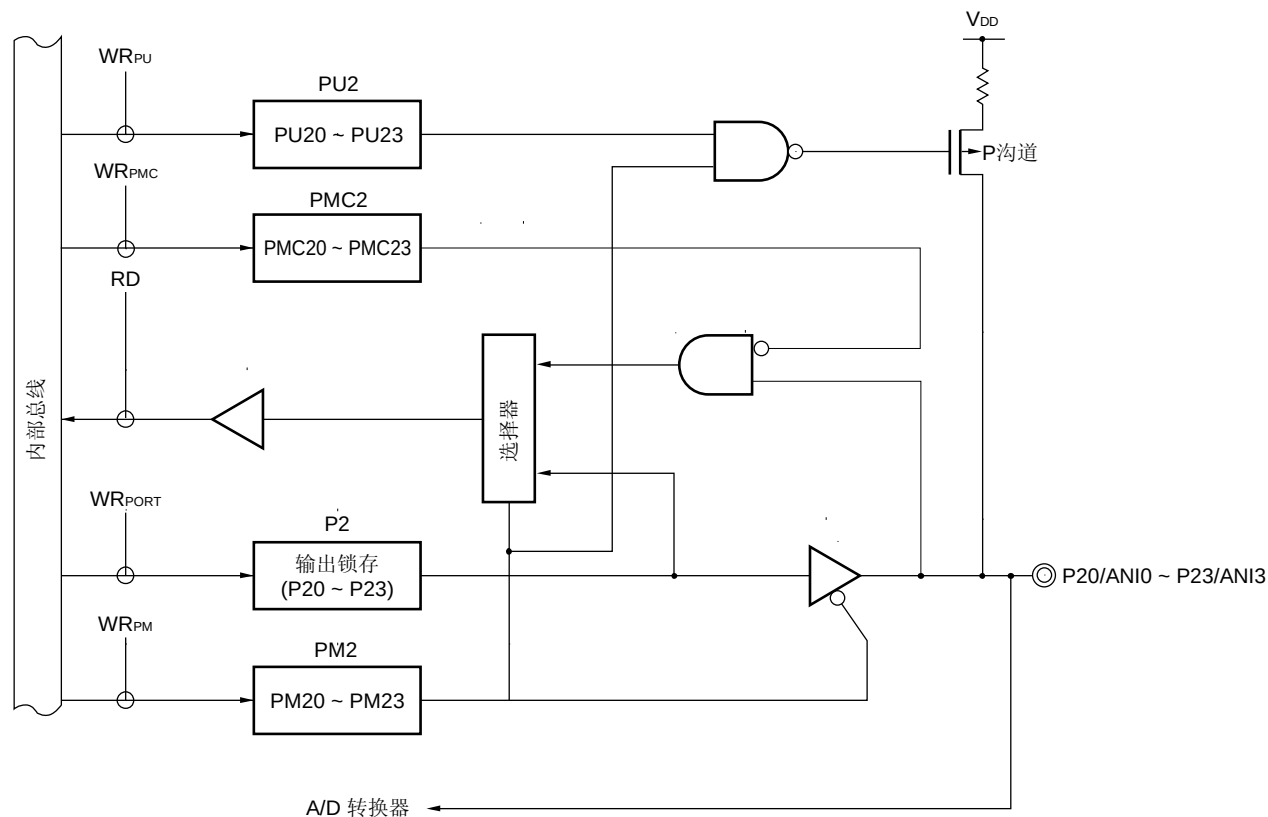
端口 2 是具有输出锁存的 4 位 I/O 端口。端口的每一位都可以通过端口模式寄存器 2 (PM2) 设定为输入或输出模式。当用作输入端口时，可通过上拉电阻选择寄存器 2 (PU2) 设置内部上拉电阻的使用。

该端口也可以作为内部 A/D 转换器的模拟输入引脚使用。

复位后端口 2 为输入模式。

图 4-2 是端口 2 的框图。

图 4-3. P20 到 P23 的框图



- 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
- PMC2: 端口模式控制寄存器 2
- RD: 读信号
- WR_{xx}: 写信号

4.2.3 端口 3

引脚 P30 ~ P33 组成了一个带输出锁存的 4 位 I/O 端口。端口的每一位都可以通过端口模式寄存器 3（PM3）设定为输入或输出模式。当 P30 ~ P33 用作输入端口时，可通过上拉电阻选择寄存器 3（PU3）设置内部上拉电阻的使用。

这个端口还能作为定时器信号的输入/输出引脚使用，也能作为输入外部中断请求信号引脚使用。

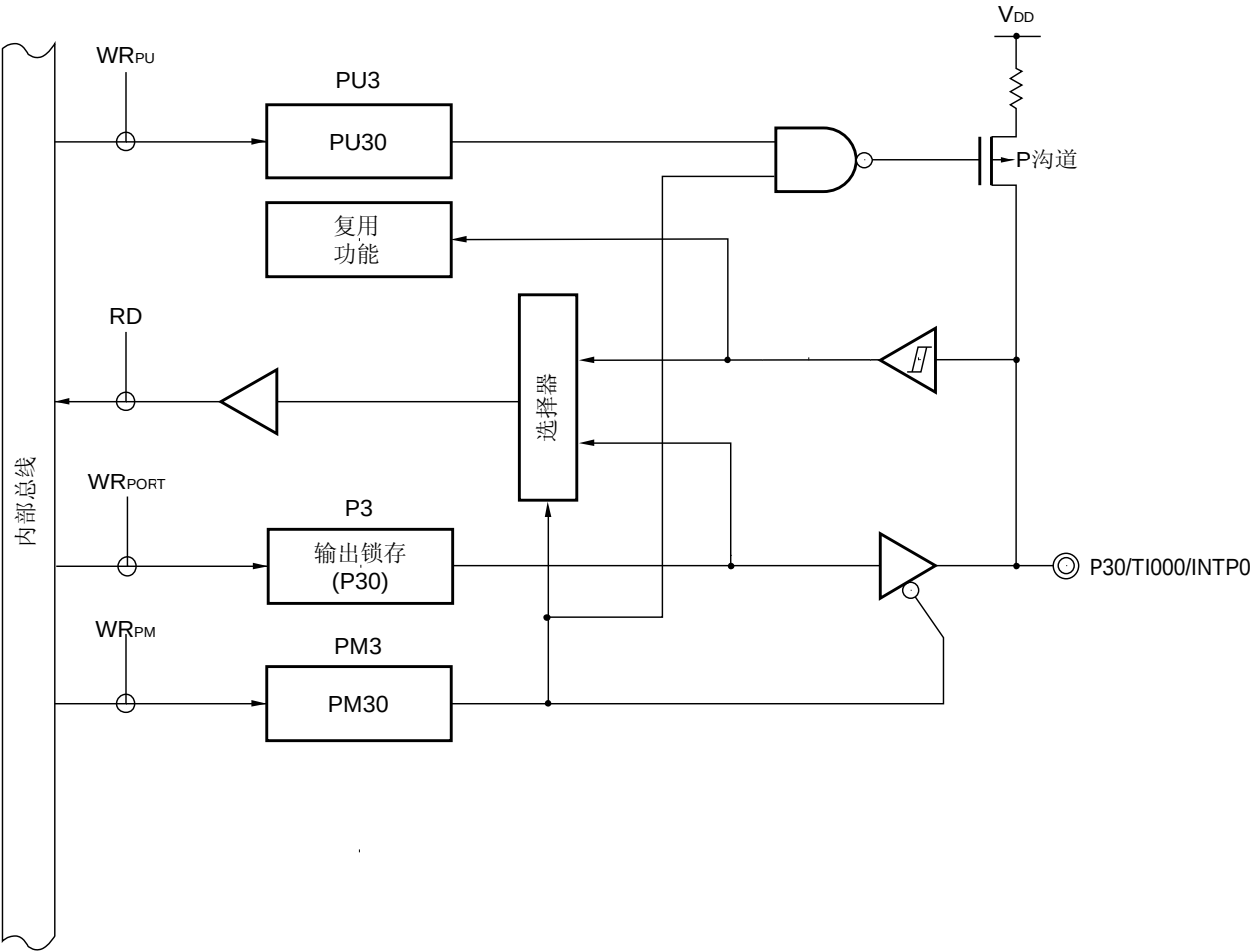
复位后端口 3 为输入模式。

P34 是一个 1 位只能用作输入的端口。该引脚也可以用作 $\overline{\text{RESET}}$ 引脚，并且当电源打开上电时，是复位功能。

引脚复用功能的设置方法，请参考 第十八章 选项字节。P34 作为输入引脚使用时，连接外部上拉电阻。

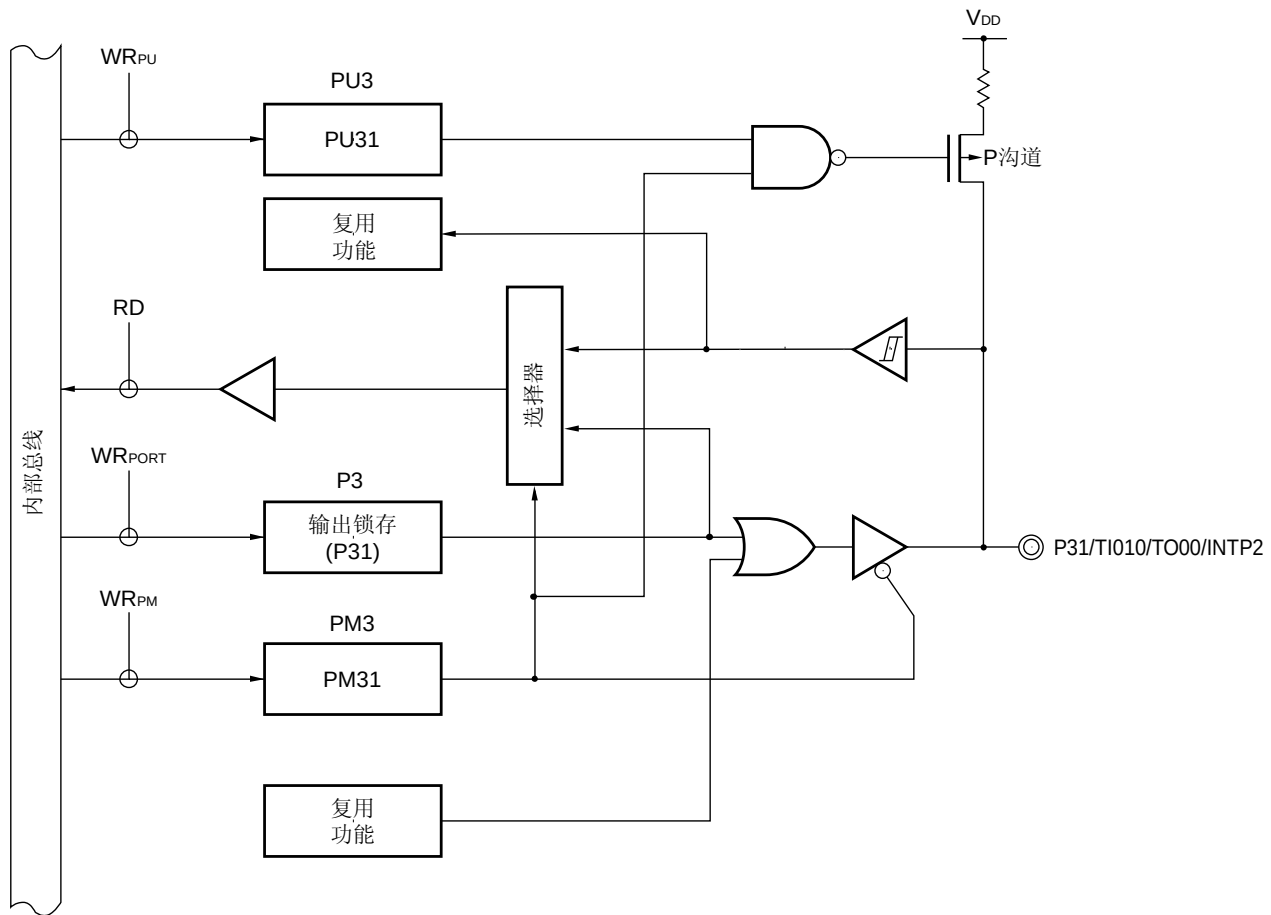
图 4-4 到 4-7 是端口 3 的框图。

图 4-4 P30 的框图



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

图 4-5 P31 的框图



P3: 端口寄存器 3
PU3: 上拉电阻选择寄存器 3
PM3: 端口模式寄存器 3
RD: 读信号
 WR_{xx} : 写信号

图 4-6 P32 和 P33 的框图

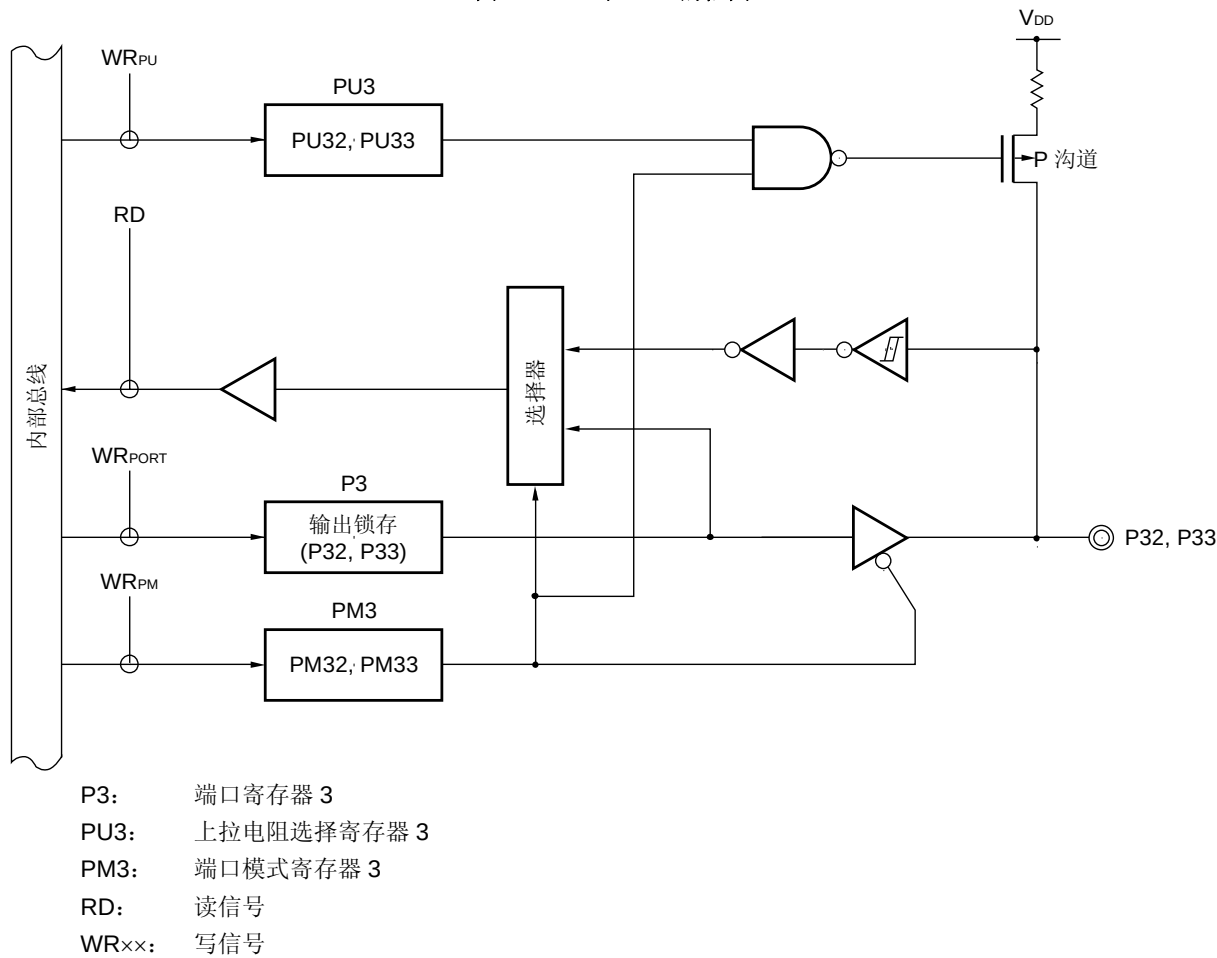
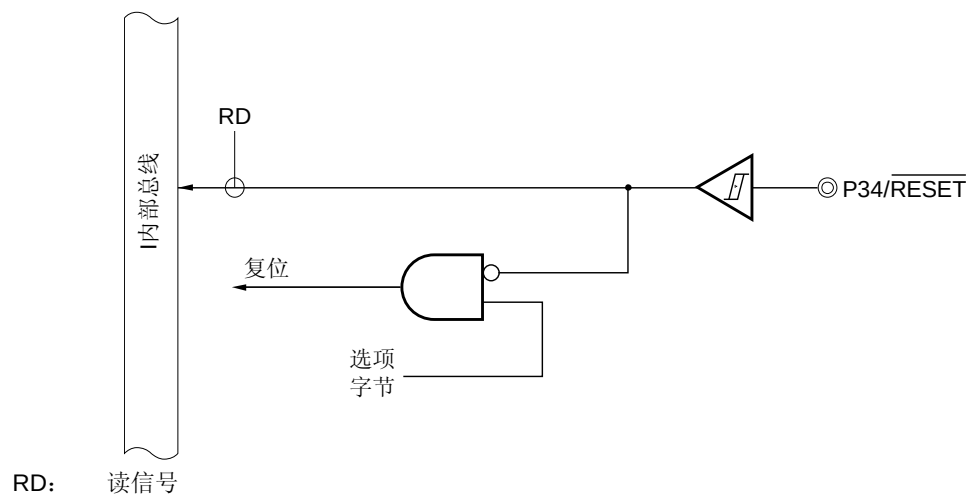


图 4-7 P34 的框图



注意事项 因为引脚 P34 可以作为 RESET 引脚使用，如果当它作为输入端口引脚使用的时候，外部输入一个复位信号也不能实现复位功能。这个端口的功能通过选项字节来选择。详情请参见 第十八章 选项字节。

同样，因为在复位释放之后会参考选项字节，如果在参考选项字节前，有一个低电平输入到复位引脚，那么复位状态不会被释放。当其作为输入端口引脚时，应连接上拉电阻。

4.2.4 端口 4

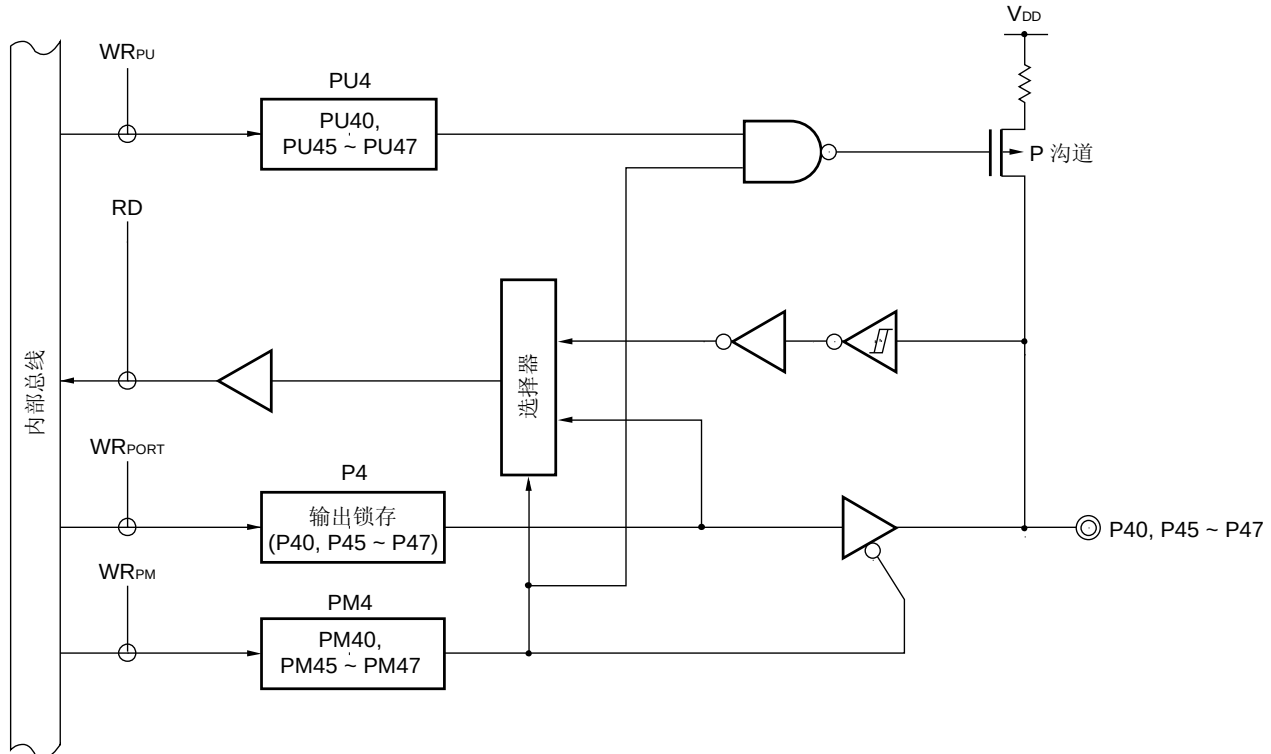
端口 4 是具有输出锁存的 8 位 I/O 端口。端口的每一位都可以通过设置端口模式寄存器 4（PM4）设定为输入或者输出模式。当 P40~P47 作为输入端口时，可以通过上拉电阻选择寄存器 4（PU4）来设置上拉电阻的使用。

P41~P44 可以复用为外部中断请求输入，串行接口数据 I/O，以及时钟输出。

复位后端口 4 为输入模式。

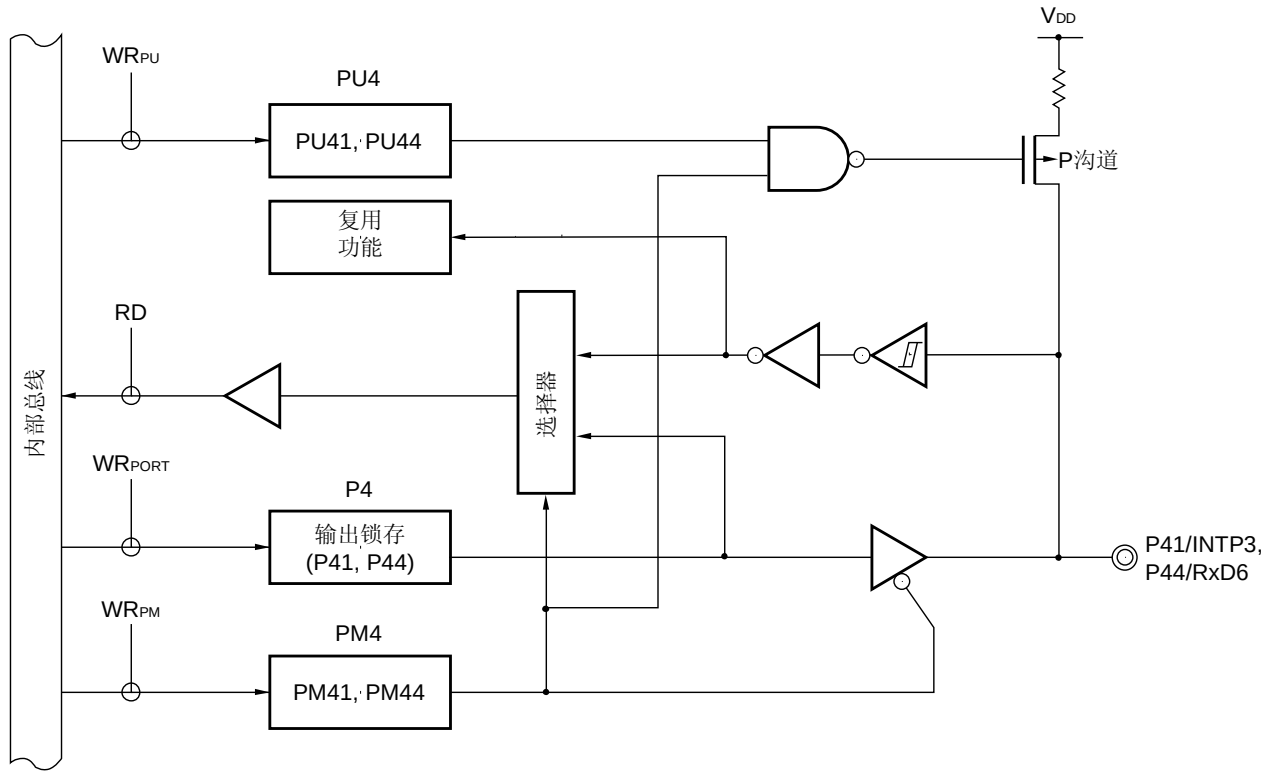
图 4-8 到 4-11 是端口 4 的框图。

图 4-8 P40 和 P45 ~ P47 的框图



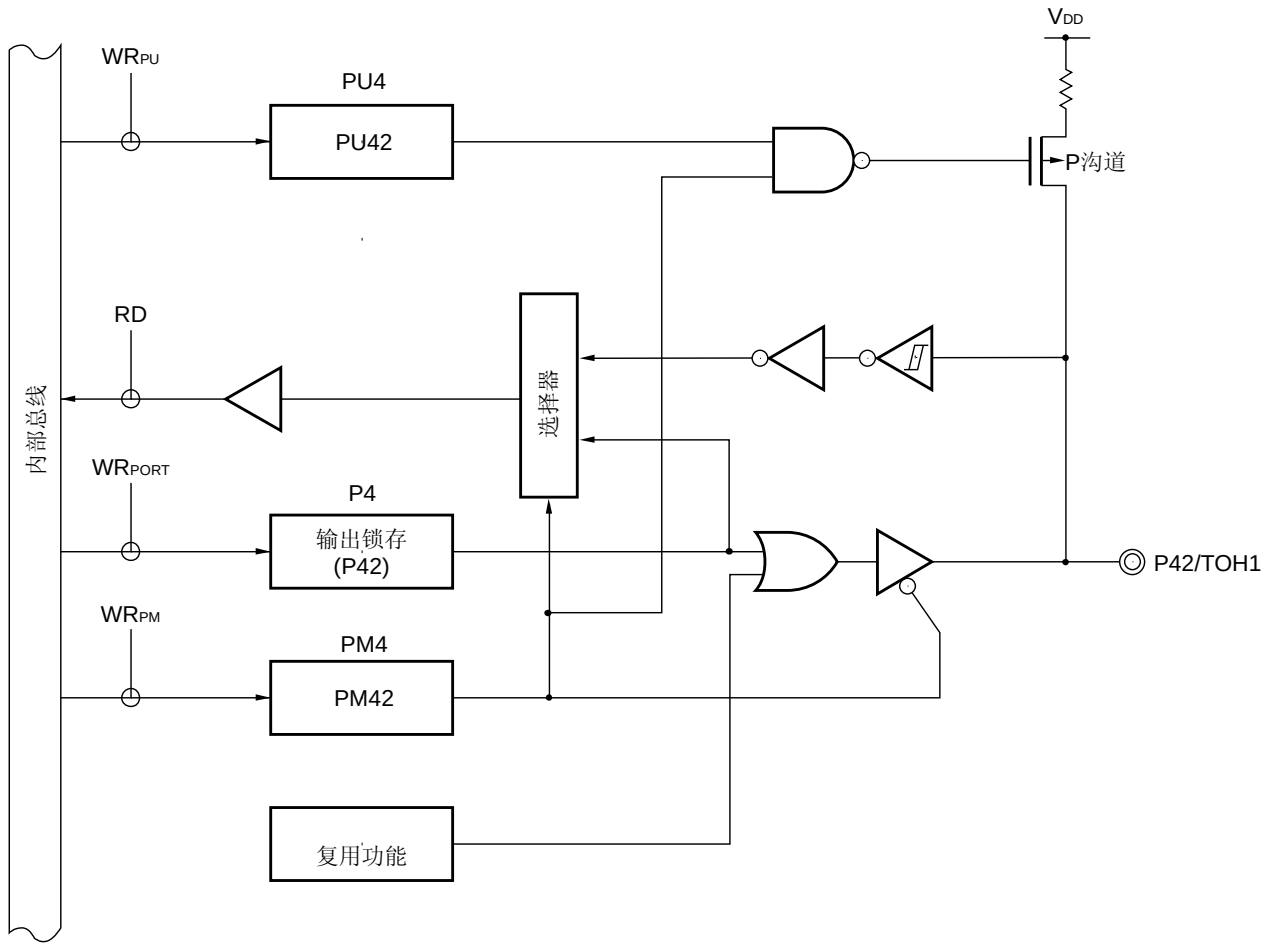
- P4: 端口寄存器 4
 PU4: 上拉电阻选择寄存器 4
 PM4: 端口模式寄存器 4
 RD: 读信号
 WR_{xx}: 写信号

图 4-9 P41 和 P44 的框图



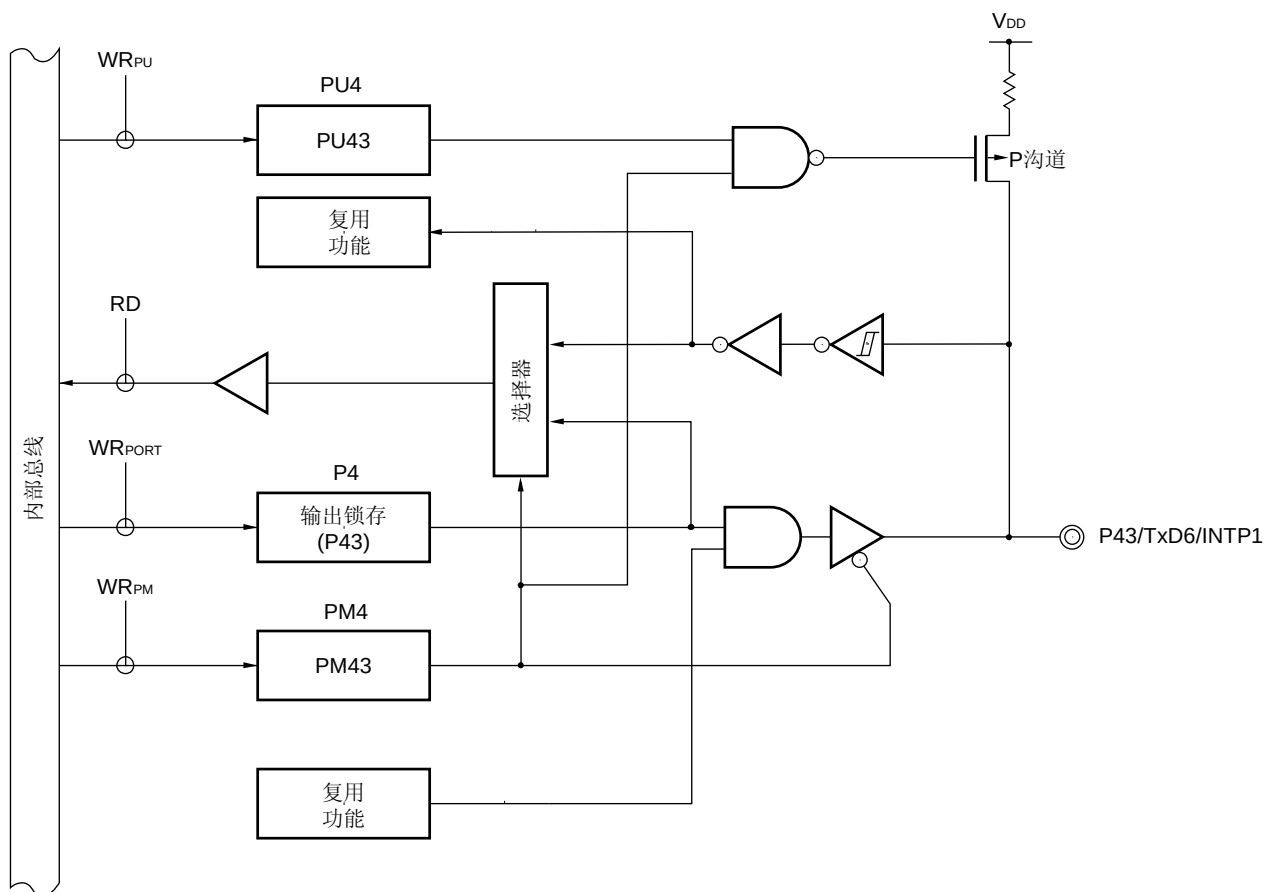
P4: 端口寄存器 4
PU4: 上拉电阻选择寄存器 4
PM4: 端口模式寄存器 4
RD: 读信号
WR_{xx}: 写信号

图 4-10 P42 的框图



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

图 4-11 P43 的框图



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

4.2.5 端口 12

端口 12 是具有输出锁存的 4 位 I/O 端口。端口的每一位都可以通过设置端口模式寄存器 12 (PM12) 设定为输入或者输出模式。当 P120 和 P123 被当作输入端口使用时，可以通过上拉电阻选择寄存器 12 (PU12) 来设置上拉电阻的使用。

引脚 P121 和 P122 也可以被当作系统时钟振荡器的 X1 和 X2 引脚使用。P121 和 P122 引脚的功能取决于系统时钟振荡器的选择。可以使用下列三种系统时钟振荡器。

(1) 高速内部振荡器

P121 和 P122 引脚能够被用作 I/O 端口引脚。

(2) 晶体/陶瓷振荡器

P121 和 P122 引脚作为 X1 和 X2 引脚使用时，不能作为 I/O 端口引脚使用。

(3) 外部时钟输入

P121 引脚作为输入外部时钟的 X1 引脚时，不能作为 I/O 端口引脚使用。

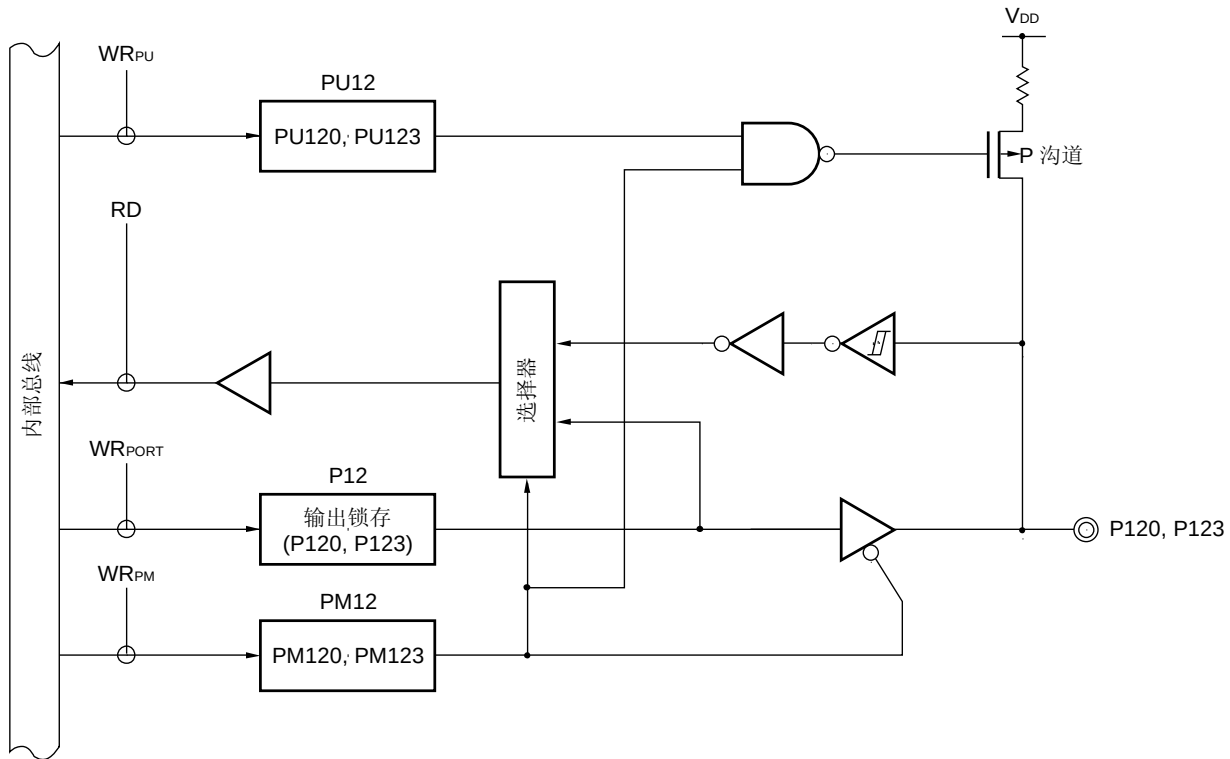
P122 引脚可以作为 I/O 端口引脚使用。

系统时钟源由选项字节选择。详情请参见 第十八章 选项字节。

复位后端口 12 为输入模式。

图 4-12 和 4-13 是端口 12 的框图。

图 4-12 P120 和 P123 的框图



PU12: 上拉电阻选择寄存器 12

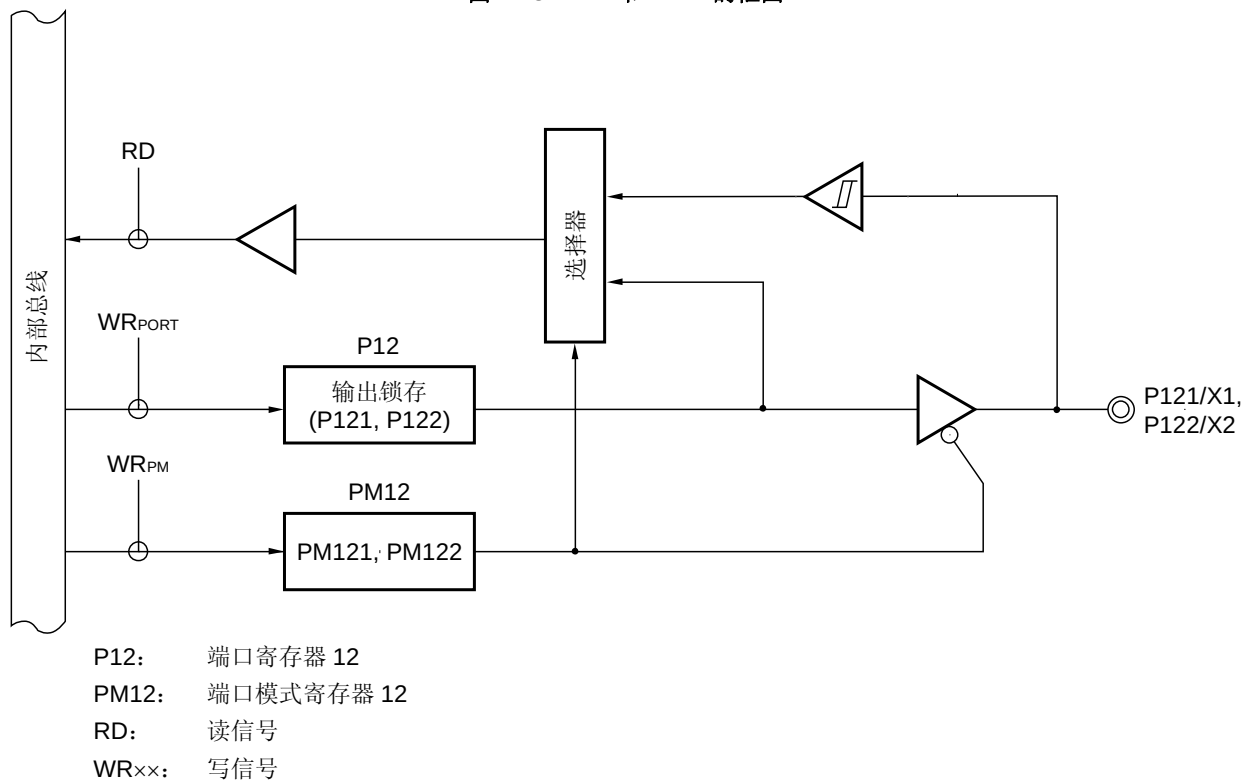
P12: 端口寄存器 12

PM12: 端口模式寄存器 12

RD: 读信号

WR $\times\times$: 写信号

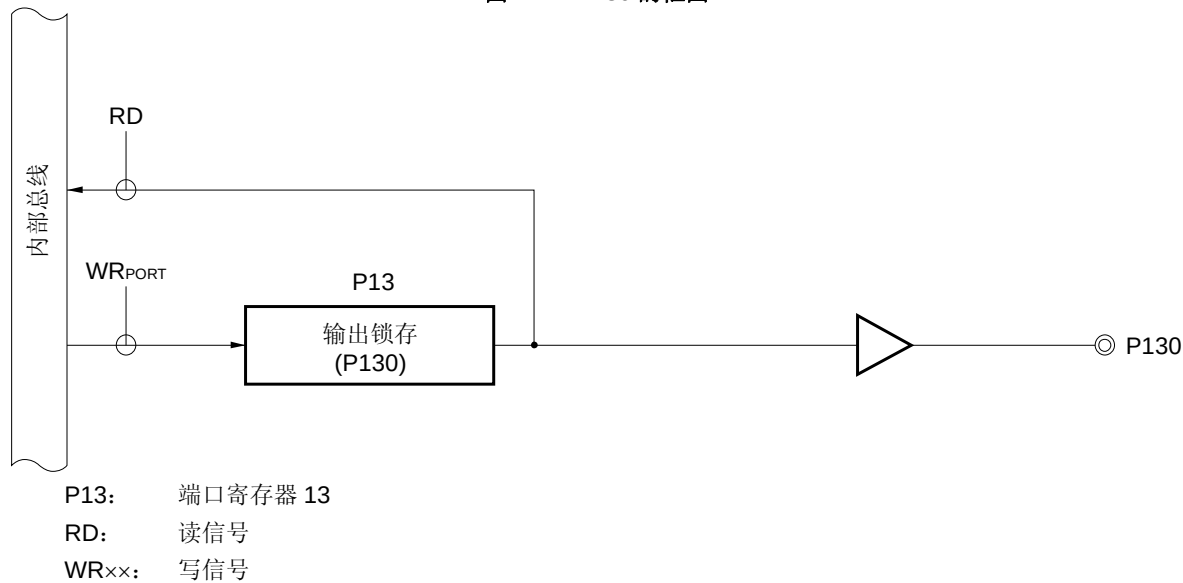
图 4-13 P121 和 P122 的框图



4.2.6 端口 13

这是一个 1 位输入端口。
图 4-14 是端口 13 的框图。

图 4-14 P130 的框图



备注 当输入复位信号时，P130 引脚会输出低电平。如果 P130 引脚在复位释放后立即输出一个高电平，则这个 P130 引脚的输出信号可以作为一个虚拟的 CPU 复位信号。

4.3 控制端口功能的寄存器

端口由下列四种寄存器控制。

- 端口模式寄存器 (PM0, PM2, PM3, PM4, PM12)
- 端口寄存器 (P0, P2, P3, P4, P12, P13)
- 端口模式控制寄存器 2 (PMC2)
- 上拉电阻选择寄存器 (PU0, PU2, PU3, PU4, PU12)

(1) 端口模式寄存器 (PM0, PM2, PM3, PM4, PM12)

这类寄存器逐位定义端口的输入或者输出模式。
每个端口模式寄存器可以由 1 位或者 8 位的操作指令设置。
复位后这些寄存器的值为 FFH。
如果端口引脚使用复用功能，可如表 4-3 所示，设置端口模式寄存器和输出锁存。

注意事项 因为 P30, P31 和 P43 引脚也被用作外部中断引脚，这些引脚的任何一个被设置成输出模式，并且它的输出电平被改变，则相应的中断请求标志被置位。因此使用这个端口引脚为输出模式时，于是，预先设置相应的中断屏蔽标志为 1。

图 4-15 端口模式寄存器的格式

地址: FF20H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM0	1	1	1	1	PM03	PM02	PM01	PM00

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	PM33	PM32	PM31	PM30

地址: FF24H 复位后: FFH R/W

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

地址: FF2CH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM12	1	1	1	1	PM123	PM122	PM121	PM120

PMmn	Pmn 引脚 I/O 模式选择 (m = 0, 2, 3, 4, 或 12; n = 0 ~ 7)
0	输出模式 (输出缓冲打开)
1	输入模式 (输出缓冲关闭)

(2) 端口寄存器 (P0, P2, P3, P4, P12, P13)

这些寄存器用于设置从相应端口引脚向连接在芯片上的外部器件输出的数据。

当在读取端口寄存器时，在输入模式下将读取引脚的电平，在输出模式下将读取端口输出锁存器的值。

P00~P03, P20~P23, P30~P34, P40 ~ P47, P120~P123, 和 P130 可由 1 位或者 8 位存储器操作指令设置。

复位后这些寄存器的值为 00H。

图 4-16 端口寄存器的格式

地址: FF00H 复位后: 00H (输出锁存) 读/写

符号	7	6	5	4	3	2	1	0
P0	0	0	0	0	P03	P02	P01	P00

地址: FF02H 复位后: 00H (输出锁存) 读/写

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) 读/写^注

符号	7	6	5	4	3	2	1	0
P3	0	0	0	P34	P33	P32	P31	P30

地址: FF04H 复位后: 00H (输出锁存) 读/写

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

地址: FF0CH 复位后: 00H (输出锁存) 读/写

符号	7	6	5	4	3	2	1	0
P12	0	0	0	0	P123	P122	P121	P120

地址: FF0DH 复位后: 00H (输出锁存) 读/写

符号	7	6	5	4	3	2	1	0
P13	0	0	0	0	0	0	0	P130

Pmn	m = 0, 2, 3, 4, 12, 或 13; n = 0~7							
	输出数据的控制 (在输出模式下)				输入数据的读取 (在输入模式下)			
0	输出 0				输入低电平			
1	输出 1				输入高电平			

注 因为 P34 是只读的，所以它复位后的值不确定

(3) 端口模式控制寄存器 2 (PMC2)

这个寄存器指定端口模式或者 A/D 转换模式。
PMC2 寄存器的每一位对应于端口 2 的每一个引脚，并且可以逐位指定。
PMC2 可以由 1 位或者 8 位的存储器操作指令设置。
复位后 PMC2 的值为 00H。

图 4-17 端口模式控制寄存器 2 的格式

地址：FF84H 复位后：00H 读/写

符号	7	6	5	4	3	2	1	0
PMC2	0	0	0	0	PMC23	PMC22	PMC21	PMC20

PMC2n	操作模式说明 (n = 0 ~ 3)
0	端口模式
1	A/D转换模式

注意事项 当 PMC20 ~PMC23 设置为 1 时，P20/ANI0~P23/ANI3 引脚不能用做端口引脚。将引脚设置为 A/D 转换器模式时，确保上拉电阻选择寄存器（PU20~PU23）置为 0。

表 4-3 使用复用功能时，端口模式寄存器，端口寄存器（输出锁存）和端口模式控制寄存器的设置

引脚名称	复用功能引脚		PMxx	Pxx	PMC2n (n = 0 ~ 3)
	名称	I/O			
P20~P23	ANI0 ~ ANI3	输入	1	×	1
P30	TI000	输入	1	×	—
	INTP0	输入	1	×	—
P31	TO00	输出	0	0	—
	TI010	输入	1	×	—
	INTP2	输入	1	×	—
P41	INTP3	输入	1	×	—
P42	TOH1	输出	0	0	—
P43	TxD6	输出	0	1	—
	INTP1	输入	1	×	—
P44	RxD6	输入	1	×	—

备注 ×: 无需考虑
 PMxx: 端口模式寄存器，
 Pxx: 端口寄存器（端口具有输出锁存）
 PMC2x: 端口模式控制寄存器

(4) 上拉电阻选择寄存器 (PU0, PU2, PU3, PU4, PU12)

这些寄存器指定是否使用连接到 P00~P03, P20~P23, P30~P33, P40~P47, P120 和 P123 引脚的内置上拉电阻。通过设置 PU0, PU2, PU3, PU4 或者 PU12, 片上的上拉电阻可以连接到 PU0, PU2, PU3, PU4 或者 PU12 各位对应的端口引脚。

PU0, PU2, PU3, PU4 和 PU12 可以由 1 位或者 8 位存储器操作指令设置。
复位后这些寄存器的值为 00H。

图 4-18 上拉电阻寄存器的格式

地址: FF30H 复位后: 00H 读/写

符号	7	6	5	4	3	2	1	0
PU0	0	0	0	0	PU03	PU02	PU01	PU00

地址: FF32H 复位后: 00H 读/写

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 读/写

符号	7	6	5	4	3	2	1	0
PU3	0	0	0	0	PU33	PU32	PU31	PU30

地址: FF34H 复位后: 00H 读/写

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

地址: FF3CH 复位后: 00H 读/写

符号	7	6	5	4	3	2	1	0
PU12	0	0	0	0	PU123	0	0	PU120

PUmn	Pmn的片上上拉电阻寄存器的连接选择 (m = 0, 2, 3, 4, 或 12; n = 0 ~ 7)
0	不连接片上上拉电阻
1	连接片上上拉电阻

4.4 端口功能操作

输入模式和输出模式的端口操作是不同的，如下所示。

注意事项 虽然 1 位存储器操作指令只对 1 位进行操作，但实际是以 8 位形式访问端口。因此，如果一个端口既有输入又有输出引脚，则对于端口中定义为输入模式的引脚，其输出锁寄存器中的内容是不确定的。

4.4.1 写入I/O端口

(1) 在输出模式下

通过传送指令可以把数据写入到输出锁寄存器中。并且，输出锁寄存器中的数据通过引脚输出。一旦数据写入输出锁寄存器，它将一直保存，直到有新数据写入。

复位信号的产生，会清空输出锁寄存器中的数据。

(2) 在输入模式下

通过传送指令可以把数据写入到输出锁寄存器中。因为输出缓冲已经关闭，引脚状态保持不变。

一旦数据写入输出锁寄存器，它将一直保存，直到有新数据写入。

复位后清空输出锁寄存器中的数据。

4.4.2 从I/O端口读入

(1) 在输出模式下

通过传送指令可以读取输出锁寄存器中的数据。输出锁寄存器中的数据保持不变

(2) 在输入模式下

通过传送指令可以读取引脚状态。输出锁寄存器中的数据保持不变

4.4.3 I/O端口的操作

(1) 在输出模式下

对输出锁寄存器的内容执行一个操作。结果被写入输出锁寄存器。输出锁寄存器的内容从引脚输出。

一旦数据写入输出锁寄存器，它将一直保存，直到有新数据写入。

复位后清空输出锁寄存器中的数据。

(2) 在输入模式下

读取引脚电平，对获取的内容执行一个操作。操作结果被写入输出锁寄存器。但是，引脚状态保持不变，因为输出缓冲已经关闭，

复位后清空输出锁寄存器中的数据。

第五章 时钟发生器

5.1 时钟发生器的功能

时钟发生器包括两部分电路，一个电路产生时钟（系统时钟）供给 CPU 和外围硬件，另一个电路产生时钟（间隔时间发生时钟）供给看门狗定时器和 8 位定时器 H1（TMH1）。

5.1.1 系统时钟振荡器

可以使用下列三种系统时钟振荡器。

- 高速内部振荡器

内置的高速振荡器可以产生 8MHz（典型值）的时钟。通过执行 STOP 指令使其振荡停止。

如果使用内置高速振荡器作为系统时钟，则 X1 和 X2 引脚可以作为 I/O 端口引脚使用。

- 晶体/陶瓷振荡器

晶体/陶瓷振荡器产生的时钟信号，通过 X1 和 X2 引脚输入单片机。输入振荡频率的范围为 2~10MHz。通过执行 STOP 指令使其振荡停止。

- 外部时钟输入电路

该电路通过 X1 引脚从外部 IC 输入时钟信号。输入振荡频率的范围为 2~10MHz。内部时钟信号的提供，可以通过执行 STOP 指令来停止。

当外部时钟输入作为系统时钟信号时，X2 引脚可以作为 I/O 引脚使用。

通过设置选项字节可以指定系统时钟源。详情请参见 第十八章 选项字节。

如 X1 和 X2 引脚作为 I/O 端口引脚使用时，详细请参见 第四章 端口功能。

5.1.2 产生间隔时间的时钟振荡器

以下的电路用于产生间隔时间的时钟振荡器。

- 低速内部振荡器

低速内部振荡器产生一个 240KHz（典型值）的时钟信号。设置低速内部振荡模式寄存器（LSRCM），可以使该振荡器停止工作。当设置选项字节时可以通过软件停止振荡器工作。

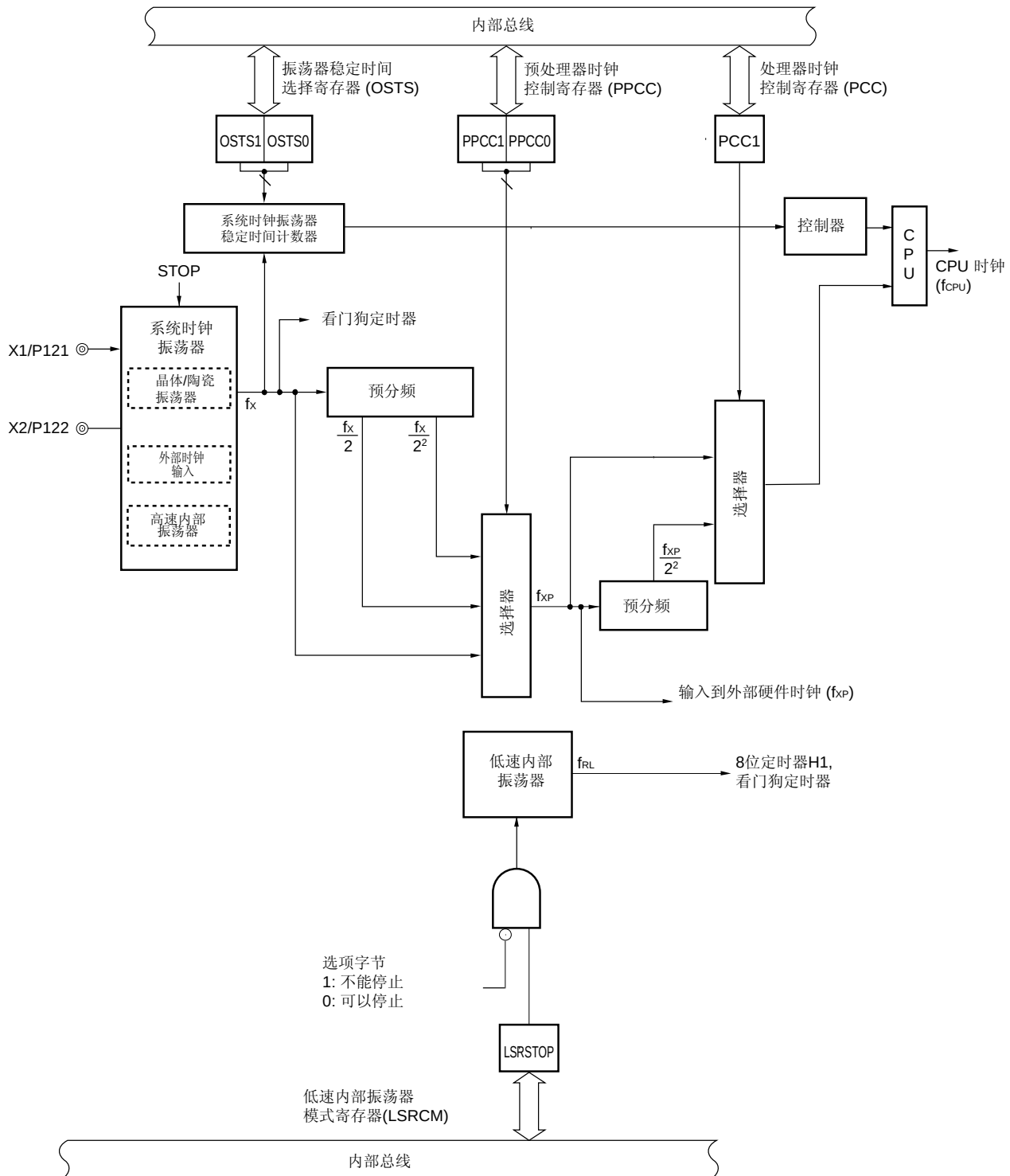
5.2 时钟发生器的配置

时钟发生器由下列硬件组成。

表 5-1. 时钟发生器的配置

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

图 5-1 时钟发生器的框图



注 通过选项字节选择高速内部振荡器、晶体/陶瓷振荡器或外部时钟输入作为系统时钟源。

5.3 时钟发生器的控制寄存器

下列四个寄存器用来控制时钟发生器。

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

(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）

这两个寄存器用来指定系统时钟的分频比。
PCC 和 PPCC 可以由 1 位或 8 位的存储器操作指令操作。
复位后 PCC 和 PPCC 的初始值为 02H。

图 5-2 处理器时钟控制寄存器 PCC 的格式

地址: FFFBH 复位后: 02H 读/写

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

图 5-3 预处理器时钟控制寄存器 PPCC 的格式

地址：FFF3H 复位后： 02H 读/写

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

PPCC1	PPCC0	PCC1	选择 CPU 时钟 (f _{CPU})
0	0	0	f _x
0	1	0	f _x /2 ^{注1}
0	0	1	f _x /2 ²
1	0	0	f _x /2 ^{2注2}
0	1	1	f _x /2 ^{3注1}
1	0	1	f _x /2 ^{4注2}
其他状态			禁止设置

注 1. 当 PPCC=01H 时，供给外围设备的时钟(f_{xp})为 f_x/2。
2. 当 PPCC=02H 时，供给外围设备的时钟(f_{xp})为 f_x/2²。

78K0S/KB1+最快的指令可以在两个 CPU 时钟周期内完成。因此，CPU 时钟和指令最短执行时间的关系如下表 5-2 所示。

表 5-2 CPU 时钟和指令最短执行时间的关系

CPU 时钟 (f _{CPU}) ^注	指令最短执行时间: 2/f _{CPU}	
	高速内部振荡时钟 (8.0 MHz (典型值))	晶体/陶瓷振荡时钟或外部时钟输入 (10.0 MHz)
f _x	0.25 μs	0.2 μs
f _x /2	0.5 μs	0.4 μs
f _x /2 ²	1.0 μs	0.8 μs
f _x /2 ³	2.0 μs	1.6 μs
f _x /2 ⁴	4.0 μs	3.2 μs

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

(2) 低速内部振荡模式寄存器(LSRCM)

这个寄存器用来选择低速内部振荡器（240KHz（典型值））的操作模式。

通过选项字节设置低速内部振荡器有效时，可以由软件使低速振荡器停止工作。如果选项字节设置低速内部振荡器不能由软件来停止，该寄存器的设置无效，低速内部振荡器继续工作。并且，WDT 的时钟源总是由低速内部振荡器提供。详情请参见 第九章 看门狗定时器

低速内部振荡模式寄存器（LSRCM）可以由 1 位或 8 位存储器操作指令操作。

复位信号的产生，会将低速内部振荡模式寄存器（LSRCM）的值清为 00H。

图 5-4 低速内部振荡模式寄存器(LSRCM)的格式

地址：FF58H 复位后：00H 读/写

符号	7	6	5	4	3	2	1	<0>
LSRCM	0	0	0	0	0	0	0	LSRSTOP
LSRSTOP	低速内部振荡器振荡/停止状态							
0	低速内部振荡器工作							
1	低速内部振荡器停止							

(3) 振荡稳定时间选择寄存器 OSTS

该寄存器用于选择 STOP 模式释放时振荡器提供的时钟振荡稳定时间。仅当晶体/陶瓷振荡器时钟被选为系统时钟源，并且在 STOP 模式释放后，通过 OSTS 设置的等待时间是有效的。如果是高速内部振荡器或外部时钟输入作为系统时钟时，不需要等待稳定时间。

上电后或复位模式释放后的系统时钟振荡器和振荡稳定时间，由选项字节选择。详情请参见 第十八章 选项字节。OSTS 可以由 8 位存储器操作指令操作。

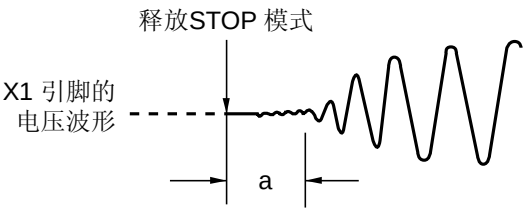
图 5-5 振荡稳定时间选择寄存器（OSTS）的格式

地址：FFF4H 复位后：不确定 R/W

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

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

- 注意事项
1. 设置和释放 STOP 模式时，按照下面的方法设置振荡稳定时间。
OSTS 设置的振荡稳定时间应该大于等于预期的振荡器振荡稳定时间。
 2. 无论 STOP 模式是被复位信号唤醒还是被中断唤醒，STOP 模式释放后的等待时间不包括从释放 STOP 模式到晶振开始振荡的时间（下图中的“a”部分）。



3. 上电或者复位后的振荡稳定时间可以通过选项字节来选择。详情请参见 第十八章 选项字节。

- 备注
1. ()： $f_x = 10\text{ MHz}$
 2. 通过查看所使用振荡器的特性说明，决定振荡器的振荡稳定时间。

5.4 系统时钟振荡器

可以使用下列三种系统时钟振荡器。

- | | |
|------------|------------------------|
| • 高速内部振荡器 | 内置振荡产生 8MHz（典型值）的时钟 |
| • 晶体/陶瓷振荡器 | 振荡时钟 2~10MHz |
| • 外部时钟输入电路 | 由 X1 引脚输入 2~10MHz 时钟信号 |

5.4.1 高速内部振荡器

78K0S/KB1+ 包含一个高速内部振荡器（8 MHz（典型值））。

如果选项字节设置高速内部振荡器作为系统时钟源，则 X1 和 X2 引脚可以作为 I/O 端口引脚使用。

关于选项字节的详情，请参见 第十八章 选项字节。关于 I/O 端口的详细情况，请参见 第四章 端口功能。

5.4.2 晶体/陶瓷振荡器

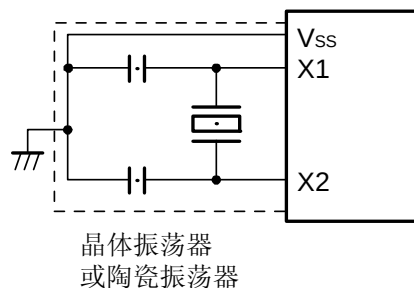
晶体/陶瓷振荡器通过与 X1 和 X2 引脚相连的晶体或陶瓷振荡器产生振荡时钟信号。

如果通过选项字节设置晶体/陶瓷振荡器作为系统时钟源，则 X1 和 X2 引脚是晶体或陶瓷振荡器的连接引脚。

关于选项字节的详情，请参见 第十八章 选项字节。关于 I/O 端口的详细情况，请参见 第四章 端口功能。

图 5-6 显示了晶体/陶瓷振荡器的外部电路。

图 5-6 晶体/陶瓷振荡器的外部电路



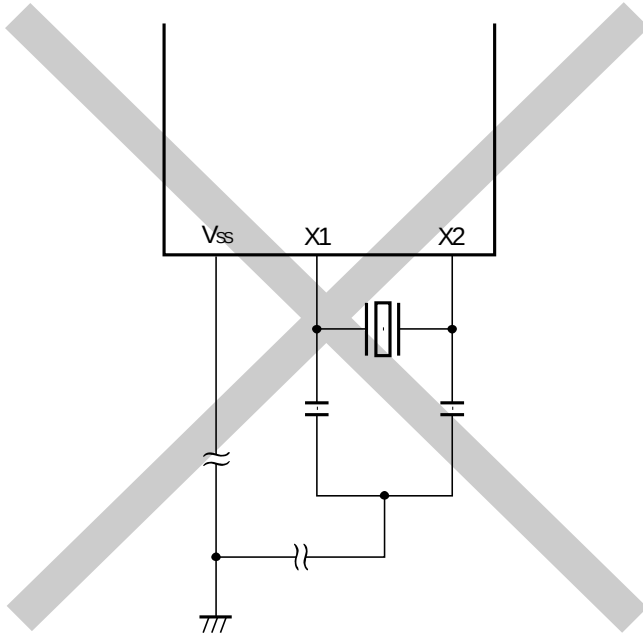
注意事项 当使用晶体/陶瓷振荡器时，布线应该按照图 5-6 中虚线内的方式连接，这样可以减小线间电容的负面影响。

- 保证连线要尽量的短。
- 不要使其它的信号线与振荡器导线互相交叉。不要在通过大电流的信号线旁边进行振荡器的布线。
- 保持振荡器电容的地和 Vss 同电位。不要让电容使用的地通过大电流。
- 不要从振荡器上取信号

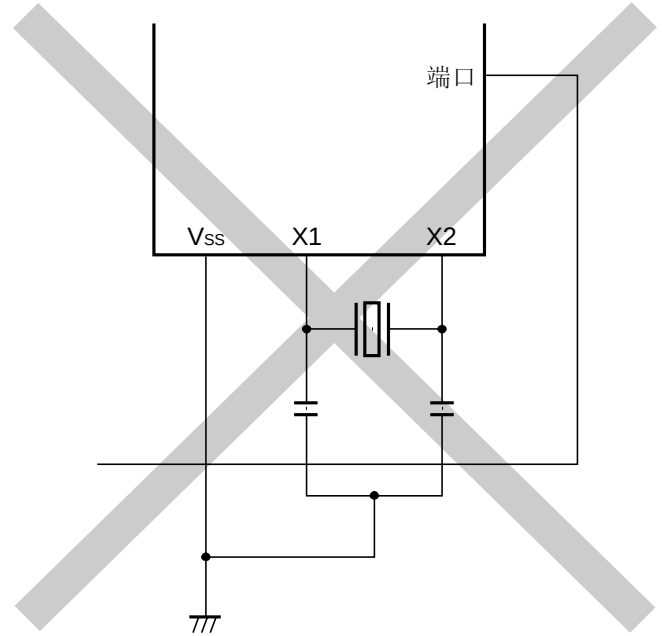
图 5-7 显示了振荡器错误连接方式的示例。

图 5-7 振荡器错误连接方式的示例 (1/2)

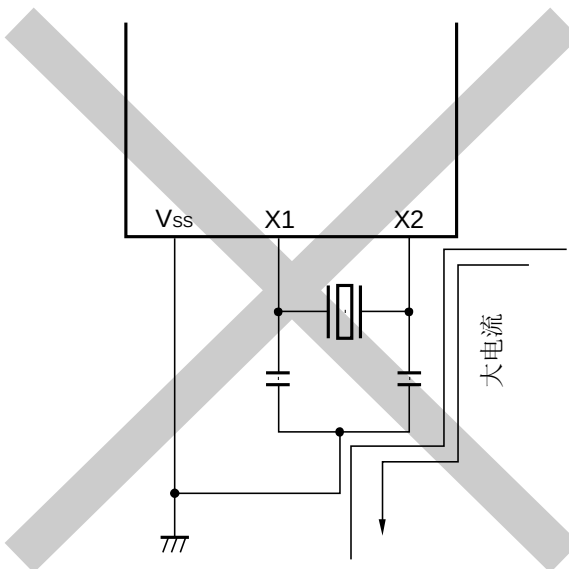
(a) 连接线太长



(b) 信号线交叉



(c) 附近有大电流波动



(d) 大电流通过振荡器的地线
(A, B 和 C 点的电位波动)

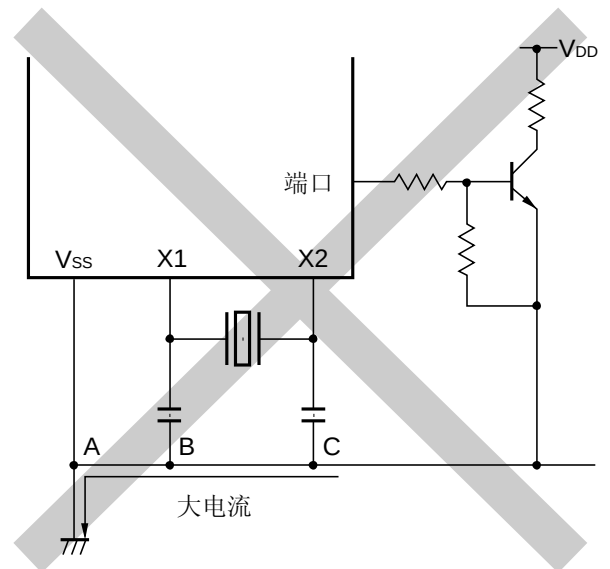
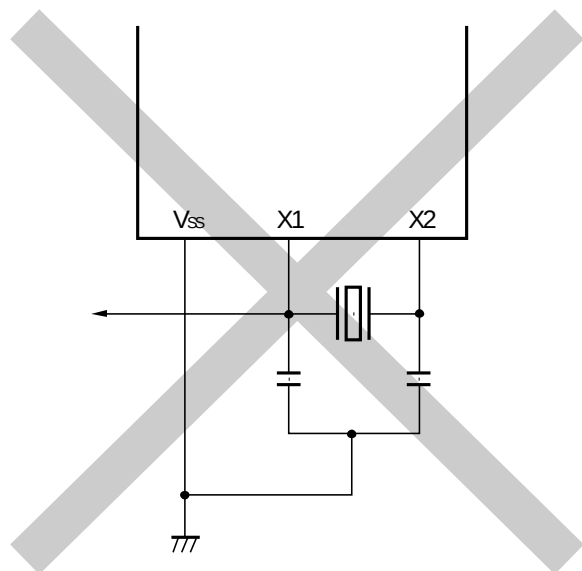


图 5-7 振荡器错误连接方式的示例 (2/2)

(e) 从振荡器取信号



5.4.3 外部时钟输入电路

该电路通过 X1 引脚从外部 IC 输入时钟信号。

如果通过选项字节选择外部时钟输入信号作为系统的时钟源，则 X2 引脚可以作为 I/O 端口引脚使用。

关于选项字节的详情，请参见 **第十八章 选项字节**。关于 I/O 端口的详细情况，请参见 **第四章 端口功能**。

5.4.4 预分频器

预分频器将系统时钟振荡器的时钟 (f_x) 输出分频后，产生一个时钟 (f_{xp}) 提供给外围硬件。也能将外围硬件的时钟 (f_{xp}) 分频后作为 CPU 时钟。

备注 通过对选项字节选择的系统时钟源（高速内部振荡器，晶体/陶瓷振荡器或外部时钟输入电路）进行分频，得到时钟输出。关于选项字节的详情，请参见 **第十八章 选项字节**。

5.5 CPU时钟发生器的操作

CPU 时钟信号(f_{CPU})来自于下列三种振荡器中的其中的一种产生的系统时钟。

- 高速内部振荡器 内置振荡产生 8MHz（典型值）的时钟
- 晶体/陶瓷振荡器 振荡时钟 2~10MHz
- 外部时钟输入电路 由 X1 引脚输入 2~10MHz 时钟信号

系统时钟源由选项字节选择。关于选项字节的详情，请参见**第十八章 选项字节**。

(1) 高速内部振荡器

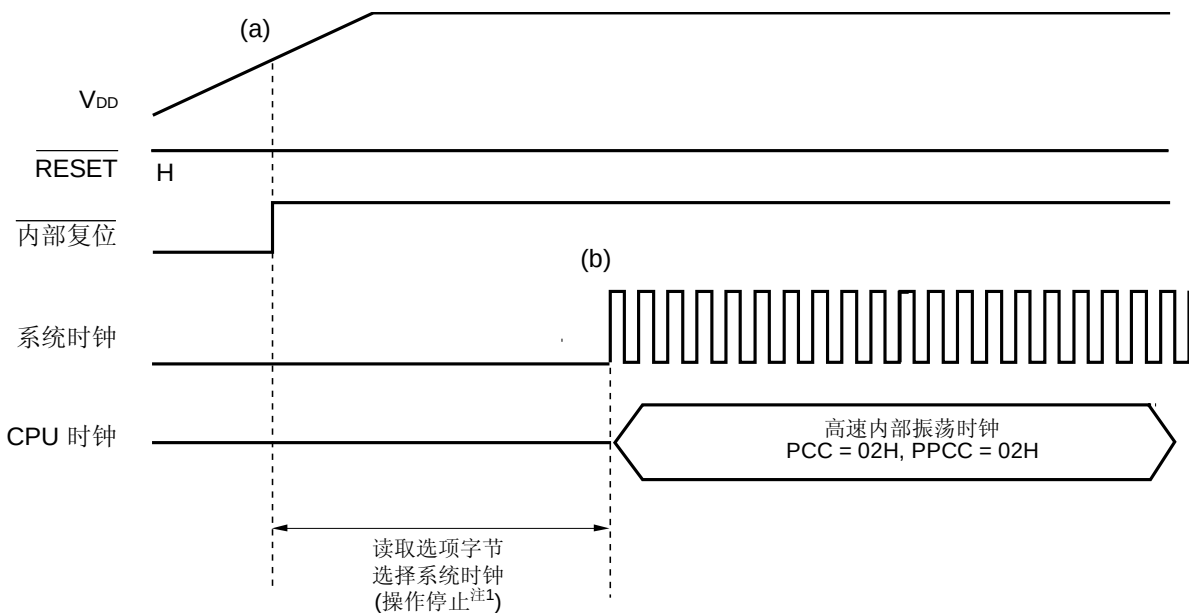
当通过选项字节选择高速内部振荡器时，可能有如下优点。

- 缩短启动时间
如果选择高速内部振荡器作为振荡器，则 CPU 不需要等待系统时钟的振荡稳定时间，就可以开始工作。因此，缩短了启动时间。
- 扩展性提高
如果选择高速内部振荡器作为振荡器，则 X1 和 X2 引脚就可以作为 I/O 端口引脚使用。详细情况，请参见 **第四章 端口功能**。

默认由高速内部振荡器启动时的时序图和状态转换框图如图 5-8，5-9 所示。

备注 当使用高速内部振荡器时，时钟精确度为 $\pm 5\%$

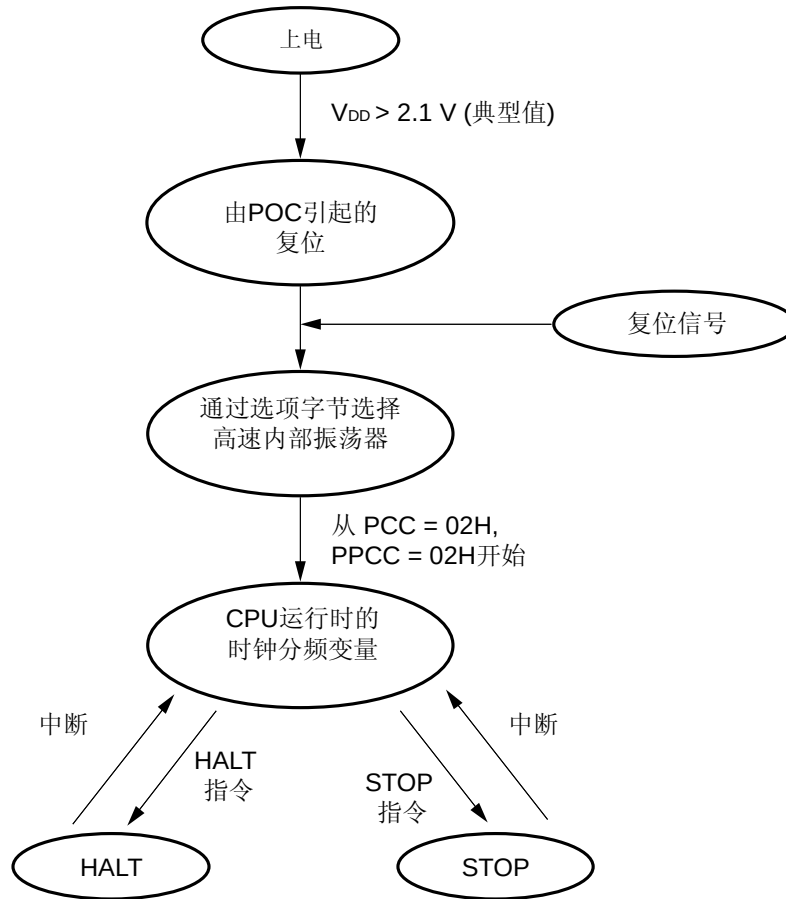
图 5-8 默认由高速内部振荡器启动的时序图



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

- (a) 系统上电时的内部复位信号由上电清零功能产生，复位后参考选项字节来选择系统时钟。
- (b) 根据选项字节选择系统时钟。然后，高速内部振荡器作为系统时钟使用。

图 5-9 默认由高速内部振荡器启动的状态转换框图



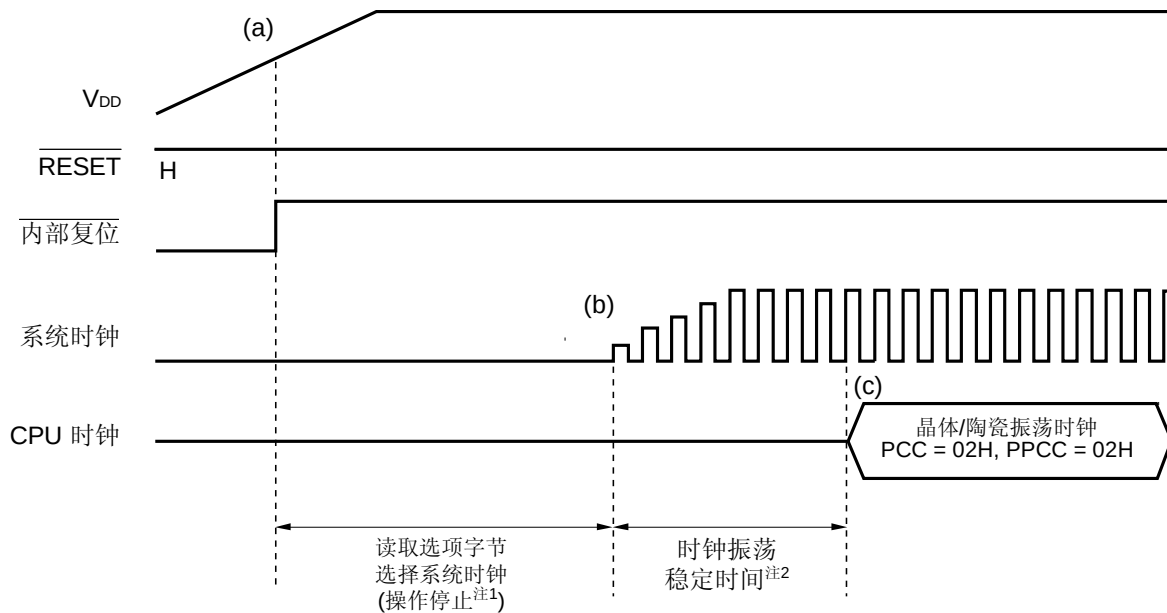
备注 PCC 预处理器时钟控制寄存器
 PPCC 预处理器时钟控制寄存器

(2) 晶体/陶瓷振荡器

如果通过选项字节选择晶体/陶瓷振荡器，晶体/陶瓷振荡器的时钟频率可在 $2\sim 10\text{MHz}$ 之间选择，与高速内部振荡器（ 8MHz （典型值））相比，因为频率偏差更小，所以处理精度更高。

默认由晶体/陶瓷振荡器启动的时序图和状态转换框图如图 5-10，5-11 所示

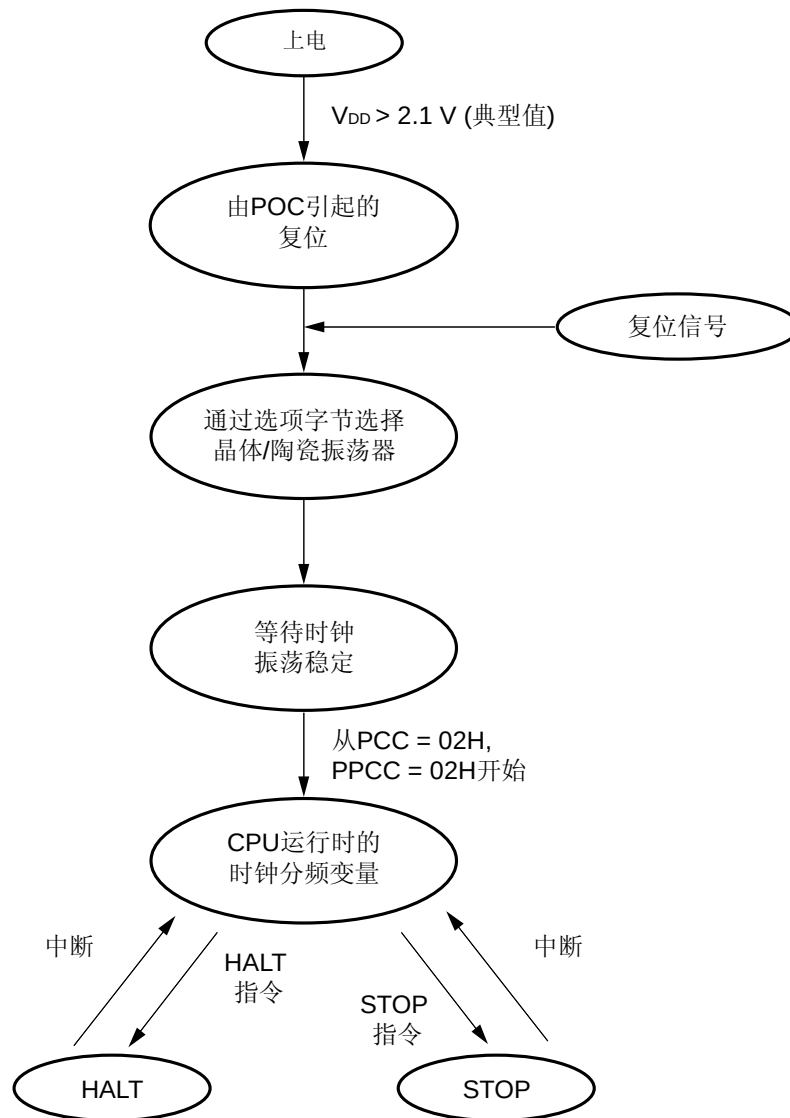
图 5-10 默认由晶体/陶瓷振荡器启动时的时序图



- 注
1. 操作停止时间为：276 μ s（最小值），544 μ s（典型值），1.075ms（最大值）。
 2. 默认启动时的时钟振荡稳定时间由选项字节设定。详情请参见 **第十八章 选项字节**。通过时钟振荡稳定时间寄存器（OSTS）选择复位模式释放后的时钟振荡稳定时间。

- (a) 系统上电时的内部复位信号由上电清零功能产生，复位后参考选项字节来选择系统时钟。
- (b) 在高速内部振荡时钟产生之后，根据选项字节选择系统时钟。在这种情况下，选择晶体/陶瓷振荡器为系统时钟源
- (c) 如果系统时钟是晶体/陶瓷振荡器，在时钟振荡稳定之后，它作为 CPU 时钟开始工作。等待时间由选项字节选择。详情请参见 **第十八章 选项字节**。

图 5-11 默认由晶体/陶瓷振荡器启动时的状态转换框图



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

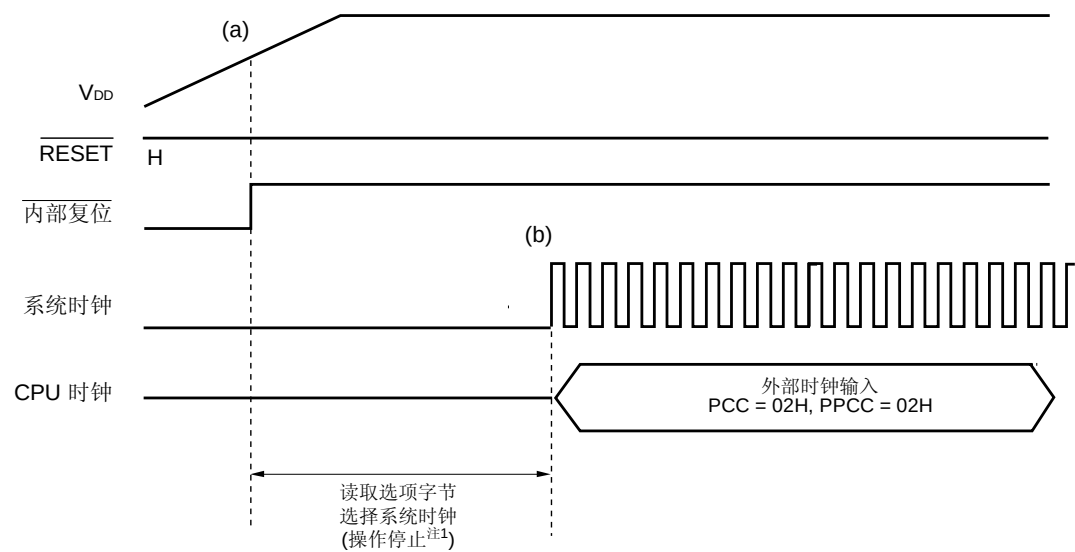
(3) 外部时钟输入电路

当通过选项字节选择外部时钟输入时，可能有如下优点。

- 高速操作
因为外部时钟频率可在 2~10MHz 之间选择，并且外部时钟的频率偏差较小，所以同高速内部振荡器（8MHz（典型值））相比，外部时钟输入具有较高的精确度。
- 扩展性提高
如果选择外部时钟输入作为振荡器，则 X2 引脚就可以作为 I/O 端口引脚使用。详细情况，请参见第四章 端口功能。

默认由外部时钟输入启动时的时序图和状态转换框图如图 5-12，5-13 所示。

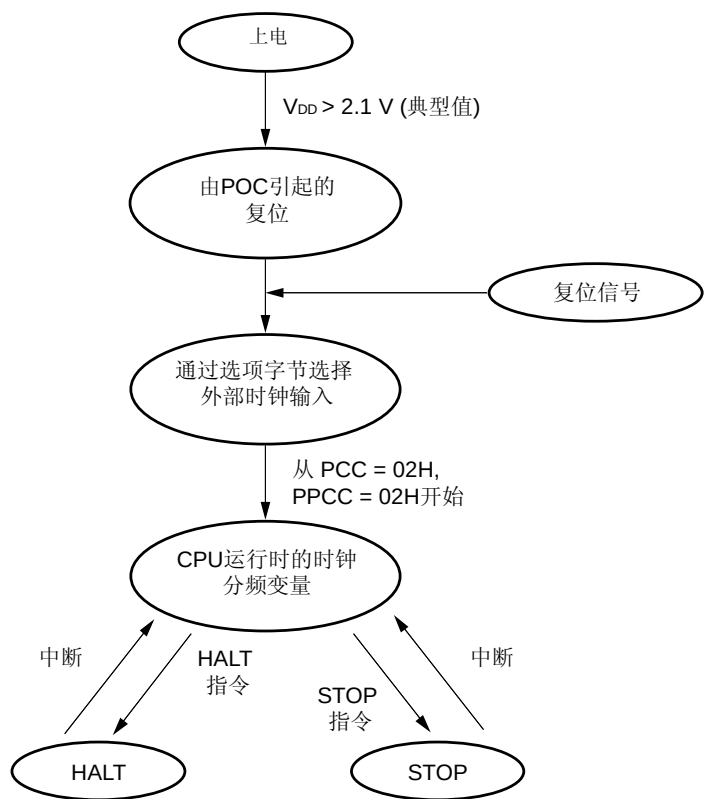
图 5-12 默认由外部时钟输入启动时的时序图



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

- (a) 系统上电时的内部复位信号由上电清零功能产生，复位后参考选项字节，选择系统时钟。
(b) 根据选项字节，选择系统时钟。然后外部时钟输入作为系统时钟工作。

图 5-13 默认由外部时钟输入启动时的状态转换框图



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 外围硬件时钟发生器的操作

下列两种时钟信号提供给外围硬件使用：

- 外围硬件时钟信号（f_{XP}）
- 低速内部振荡时钟（f_{RL}）

(1) 外围硬件时钟信号

系统时钟（f_x）分频，得到外围硬件的时钟。分频比由预处理器时钟控制寄存器（PPCC）选择。可以选择如下三种时钟频率：“f_x”，“f_x/2”，和“f_x/2²”。表 5-3 列举了外围硬件时钟提供情况。

表 5-3 外围硬件时钟信号

PPCC1	PPCC0	外围硬件时钟信号的选择(f _{XP})
0	0	f _x
0	1	f _x /2
1	0	f _x /2 ²
1	1	禁止设置

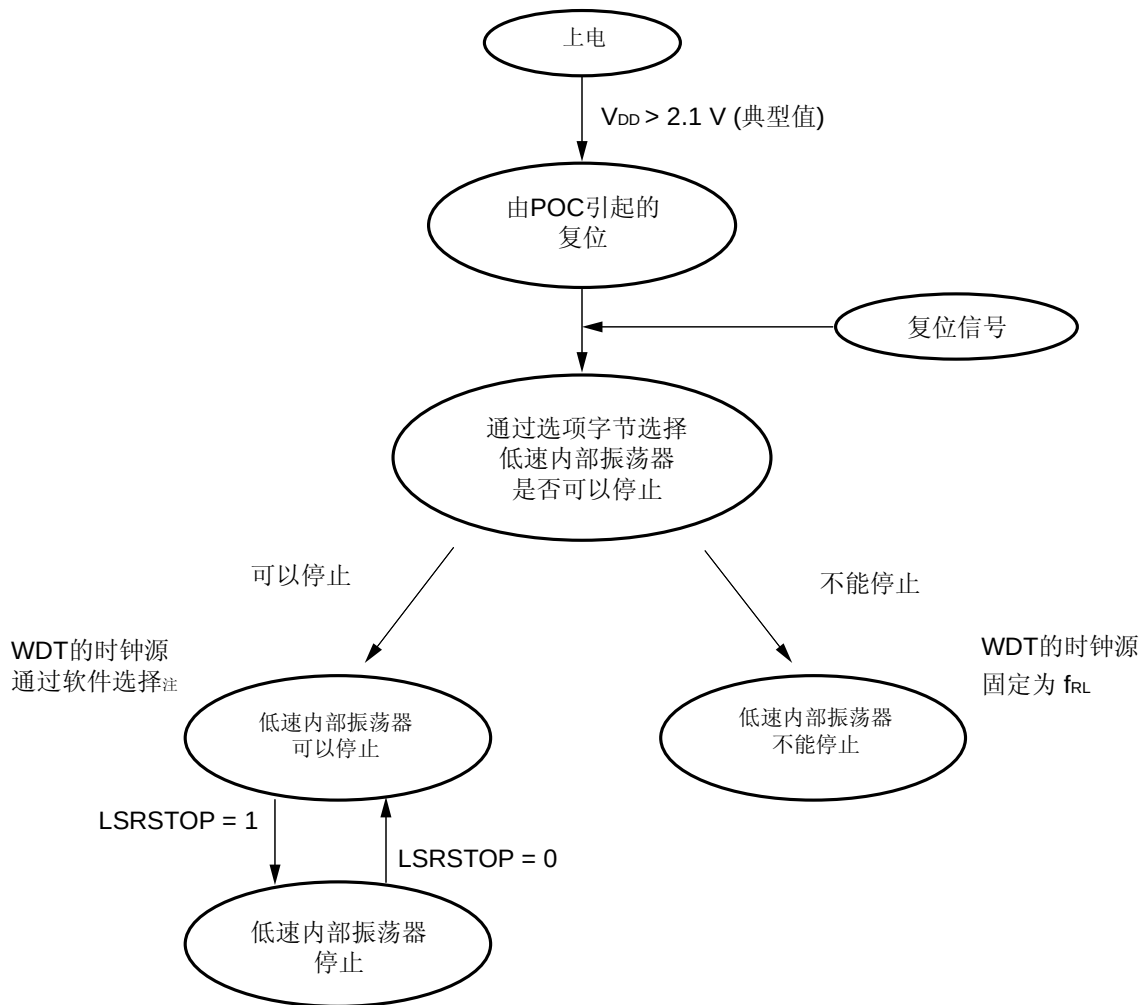
(2) 低速内部振荡时钟

在复位释放后时钟振荡器的低速内置振荡器开始工作，用于产生间隔时间，振荡频率为 240 kHz（典型值）。
可以通过选项字节指定低速内置振荡器是否可以由软件停止。如果设置为可以通过软件停止低速内部振荡器的工作，则可以通过低速内部振荡模式控制寄存器（LSRCM）控制振荡器的启动或停止。如果指定为不能通过软件停止低速内部振荡器，则 WDT 的时钟信号源只能锁定为低速内部振荡器时钟（f_{RL}）。
低速内部振荡器时钟独立于 CPU 的时钟。因此，如果用它来作为 WDT 的时钟源，即使 CPU 时钟停止，也可检测到系统挂起。如果把它用作为 8 位定时器 H1 的时钟源，则 8 位定时器 H1 可以在待机状态下工作。
表 5-4 显示了当低速内部振荡器作为 WDT 和 8 位定时器 H1 时钟源时，低速内部振荡器的工作状态。图 5-14 显示了低速内部振荡器的状态转换。

表 5-4 低速内部振荡器的工作状态

选项字节设置		CPU状态	WDT状态	TMH1状态
可以通过软件停止	LSRSTOP = 1	工作模式	停止	停止
	LSRSTOP=0		工作	工作
	LSRSTOP=1	待机	停止	停止
	LSRSTOP=0		停止	工作
不能停止工作		工作模式	工作	
		待机		

图 5-14 低速内部振荡器的状态转换



注 看门狗定时器(WDT)的时钟源可以为 f_x 或 f_{RL} ，或者它可能停止。详细情况，请参见 第九章 看门狗定时器。

6.1 16 位定时器/事件计数器 00 的功能

16 位定时器/事件计数器 00 有如下功能。

(1) 间隔定时器

16 位定时器/事件计数器 00 按照预置的时间间隔产生中断请求。

- 计数值 2 ~ 65536

(2) 外部事件计数器

16 位定时器/事件计数器 00 可以测量外部输入的有效脉冲个数。

- 有效脉冲宽度为： $2/f_{XP}$ 或更大

(3) 脉冲宽度测量

16 位定时器/事件计数器 00 可以测量外部输入信号的脉冲宽度。

- 有效脉冲宽度为： $2/f_{XP}$ 或更大

(4) 方波输出

16 位定时器/事件计数器 00 可以输出任意频率的方波。

- 周期： $(2 \sim 65536) \times 2 \times$ 计数时钟周期

(5) PPG 输出

16 位定时器/事件计数器 00 可以输出任意周期和脉冲宽度的方波。

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

(6) 单脉冲输出

16 位定时器/事件计数器 00 可以输出任意脉冲宽度的单脉冲。

6.2 16 位定时器/事件计数器 00 的配置

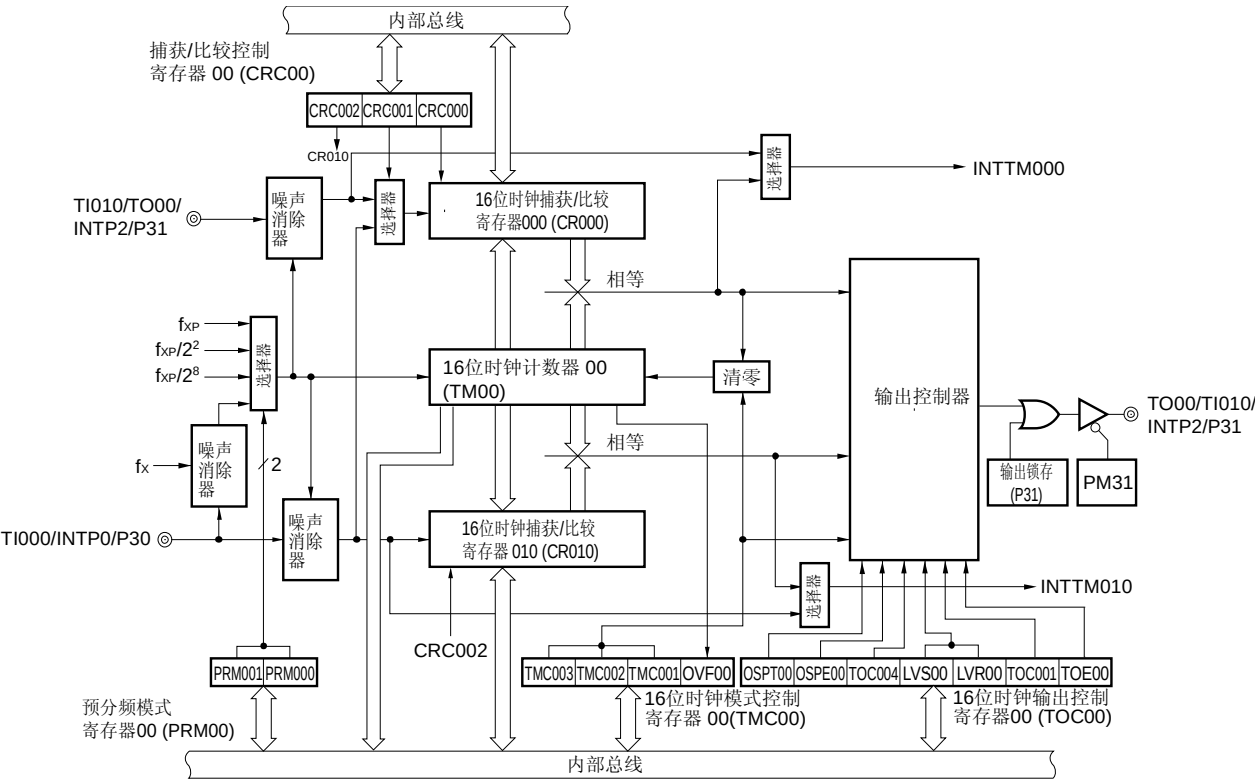
16 位定时器/事件计数器 00 由如下硬件组成。

表 6-1 16 位定时器/事件计数器 00 的配置

项目	配置
定时器计数器	16 位定时器计数器 00 (TM00)
寄存器	16 位定时器捕获/比较寄存器 000, 010 (CR000, CR010)
定时器输入	TI000, TI010
定时器输出	TO00, 输出控制器
控制寄存器	16 位定时器模式控制寄存器 00 (TMC00) 捕获/比较控制寄存器 00 (CRC00) 16 位定时器输出控制寄存器 00 (TOC00) 预分频模式寄存器 00 (PRM00) 端口模式寄存器 3 (PM3) 端口寄存器 3 (P3)

图 6-1 是这些计数器的框图。

图 6-1 16 位定时器/事件计数器 00 框图



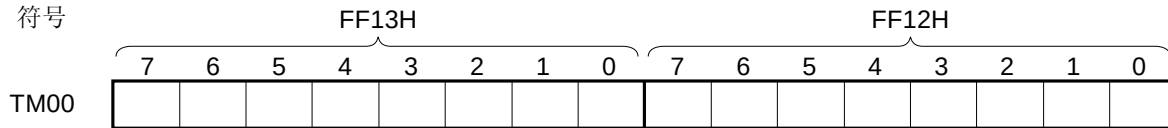
(1) 16 位定时器计数器 00 (TM00)

TM00 是 16 位只读寄存器，用于脉冲计数。

计数器随计数时钟的上升沿同步增加。如果在运行过程中读取计数器的值，那么计数时钟的输入将被暂停，并读取该时刻的计数值。

图 6-2. 16 位定时器计数器 00 (TM00) 的格式

地址: FF12H, FF13H 复位后: 0000H 只读



如下情况中计数值被重置为 0000H。

- <1> 产生复位信号
- <2> 如果 TMC003 和 TMC002 清零
- <3> 在清零&启动模式中，如果输入 TI000 有效沿时，TI000 输入的有效沿会将计数值清零。
- <4> 在清零&启动模式中，如果 TM00 和 CR000 相等时，会将计数值清零。
- <5> 如果 OSPT00 在单脉冲输出模式下被置 1

注意事项 1. 读取 TM00 时，CR010 无法捕获其值。

2. 当读取 TM00 时，不会发生计数丢失，因为计数时钟暂时停止工作，在读取后重新恢复。

(2) 16 位定时器捕获/比较寄存器 000 (CR000)

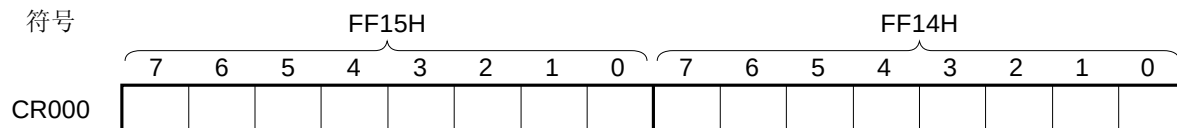
CR000 是 16 位寄存器，具有捕获寄存器和比较寄存器的功能。是作为比较寄存器使用，还是作为捕获寄存器使用，由捕获/比较控制寄存器 00 (CR00) 的第 0 位 (CRC000) 来设置。

CR000 由 16 位存储器操作指令设置。

复位信号的产生，会将 CR000 清零为 0000H。

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

地址: FF14H, FF15H 复位后: 0000H 读/写



- **当 CR000 作为比较寄存器使用**

CR000 的设定值与 16 位定时器/计数器 00 (TM00) 的计数值不断比较，如果两值相等，则产生中断请求 (INTTM000)。当设置 TM00 作为间隔定时器时，该寄存器也可以用来保存间隔时间。

- **当 CR000 作为捕获寄存器使用**

可选择 TI000 或 TI010 引脚的有效沿作为捕获触发信号。通过预分频模式寄存器 00 (PRM00) 设置 TI000 或 TI010 的有效沿。(参见表 6-2)

表 6-2 CR000 捕获触发和 TI000 与 TI010 的有效沿

(1) 选择 TI000 引脚有效沿作为捕获触发 (CRC001 = 1, CRC000 = 1)

CR000 捕获触发	TI000 引脚有效沿	ES010 ES000	
		ES010	ES000
下降沿	上升沿	0	1
上升沿	下降沿	0	0
无捕获操作	上升和下降沿均为有效沿	1	1

(2) 选择 TI010 引脚有效沿作为捕获触发 (CRC001 = 0, CRC000 = 1)

CR000 捕获触发	TI010 引脚有效沿	ES110 ES100	
		ES110	ES100
下降沿	下降沿	0	0
上升沿	上升沿	0	1
上升和下降沿均为有效沿	上升和下降沿均为有效沿	1	1

- 备注**
1. 禁止设置 ES010, ES000 = 1, 0 和 ES110, ES100 = 1, 0。
 2. ES010, ES000: 预分频模式寄存器 00 (PRM00) 的第 5 位和第 4 位
ES110, ES100: 预分频模式寄存器 00 (PRM00) 的第 7 位和第 6 位
CRC001, CRC000: 捕获/比较控制寄存器 00 (CRC00) 的第 1 位和第 0 位。

- 注意事项**
1. 当 TM00 和 CR000 相等而进入清零&启动模式时, CR000 的值不要设置为 0000H。也就是说当该寄存器作为外部事件计数器时, 不能执行 1 个脉冲的计数操作。然而, 在自由运行模式和由 TI000 引脚有效沿进入的清零&启动模式中, 如果 CR000 清为 0000H, 则 TM00 溢出后 (FFFFH), CR000 从 0000H 变为 0001H 时, 产生一个中断请求 (INTTM000)。
 2. 如果 CR000 的新值小于 16 位定时器计数器 0 (TM00) 的值, TM00 继续计数, 溢出, 然后从 0 开始重新计数。因此, 如果 CR000 的新值小于原来的值, 在 CR000 的值改变后, 定时器必须复位重新启动。
 3. 当 16 位定时器/事件计数器 00 停止工作后, CR000 的值是无法保证的。
 4. 当 CR000 设为比较模式, 即使有捕获触发信号输入, 也不执行捕获操作。
 5. 当 P31 用做 TI010 的有效沿输入引脚时, 不能用作定时器输出引脚 (TO00)。同样, 当 P31 被用于定时器输出引脚 TO00 时, 也不能用于 TI010 的有效沿输入引脚。
 6. 当 CR000 作为捕获寄存器使用, 如果同时发生寄存器的读操作和输入捕获触发操作, 则捕获触发输入有优先权, 而读出的数据为不确定值。同样, 如果定时器停止计数与捕获触发输入信号发生冲突, 捕获触发为不确定状态。
 7. 如果在 TM00 操作中改变 CR000 的设置, 可能引起错误。要改变设置, 请参考 6.5 16 位定时器/事件计数器 00 的相关注意事项 (17) 在定时器操作过程中改变比较寄存器。

(3) 16 位捕获/比较寄存器 010 (CR010)

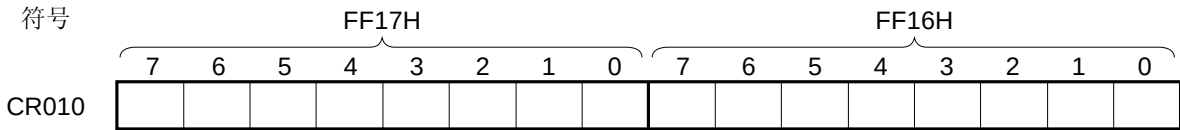
CR010 是可以作为捕获寄存器和比较寄存器使用的 16 位寄存器。是作为比较寄存器使用，还是作为捕获寄存器使用，由捕获/比较控制寄存器 00 (CR00) 的第 0 位 (CRC002) 来设置。

CR010 由 16 位操作指令设置。

复位信号的产生，会将 CR010 清为 0000H。

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

地址: FF16H, FF17H 复位后: 0000H 读/写



• 当 CR010 用于比较寄存器

CR010 的设定值与 16 位定时器/计数器 00 (TM00) 的计数值不断比较，如果两值相等，则产生中断请求 (INTTM010)。

• 当 CR010 用于捕获寄存器

可选择 TI000 引脚的有效沿作为捕获触发信号。TI000 的有效沿通过预分频模式寄存器 00 (PRM00) 来设置 (参考 表 6-3)。

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

CR010 捕获触发	TI000 引脚有效沿	ES010 ES000	
		ES010	ES000
下降沿	下降沿	0	0
上升沿	上升沿	0	1
上升和下降沿均为有效沿	上升和下降沿均为有效沿	1	1

备注 1. 禁止设置 ES010、ES000 = 1, 0。
 2. ES010, ES000: 预分频模式寄存器 00 (PRM00) 的第 5 位和第 4 位
 CRC002: 捕获/比较控制寄存器 00 (CRC00) 的第 2 位

- 注意事项 1. 在自由运行模式和由 TI000 引脚有效沿进入的清零&启动模式中，如果 CR010 清为 0000H，则 TM00 溢出后 (FFFFH)，CR010 从 0000H 变为 0001H 时，产生一个中断请求 (INTTM010)。
 2. 如果 CR010 新值小于 16 位定时器计数器 0 (TM00) 的值，TM00 继续计数，溢出，然后从 0 开始重新计数。因此，如果 CR010 的新值小于原来的值，在 CR010 的值改变后，定时器必须复位重新启动。
 3. 当 16 位定时器/事件计数器 00 停止工作后，CR010 的值是无法保证的。
 4. 当 CR010 设为比较模式，即使有捕获触发信号输入，也不执行捕获操作。

5. 当 CR010 作为捕获寄存器使用，如果同时发生寄存器的读操作和输入捕获触发操作，则捕获触发输入有优先权，而读出的数据为不确定值。同样，如果定时器停止计数与捕获触发输入发生冲突，捕获触发为不确定状态。
6. 如果在 TM00 操作中改变 CR010 的设置，可能引起错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 00 的相关注意事项（17）在定时器操作过程中改变比较寄存器。

6.3 16 位定时器/事件计数器 00 的控制寄存器。

使用如下六个类型的寄存器来控制 16 位定时器/事件计数器 00。

- 16 位定时器模式控制寄存器 00（TMC00）
- 捕获/比较控制寄存器 00（CRC00）
- 16 位定时器输出控制寄存器 00（TOC00）
- 预分频模式寄存器 00（PRM00）
- 端口模式寄存器 3（PM3）
- 端口寄存器 3（P3）

（1）16 位定时器模式控制寄存器 00（TMC00）

该寄存器用于设置 16 位定时器的操作模式，16 位定时器计数器 00（TM00）清零模式，输出时序和溢出检测。

TMC00 由 1 位或 8 位存储器操作指令设置。

复位信号的产生，会将 TMC00 值清为 00H。

注意事项 当 TMC002 和 TMC003（操作停止模式）的值不为 0 时，16 位定时器 TM00 开始工作，当 TMC002 和 TMC003 位的值设置为 0 时 16 位定时器 TM00 停止工作。

图 6-5. 16 位定时器模式控制寄存器 00 (TMC00) 的格式

地址: FF60H 复位后: 00H 读/写

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

TMC003	TMC002	TMC001	操作模式和清零模式选择	TO00 反转时序选择	中断请求发生
0	0	0	操作停止 (TM00 清零)	无改变	无中断产生
0	0	1			
0	1	0	自由运行模式	当 TM00 和 CR000 相等时, 或 TM00 和 CR010 相等时反转	<当作为比较寄存器时> 当 TM00 和 CR000 相等或
0	1	1		当 TM00 和 CR000 相等时, 或 TM00 与 CR010 相等时, 或 TI000 引脚有效沿时反转	TM00 和 CR010 相等时产生中 断请求 <当作为捕获寄存器时>
1	0	0	在 TI000 引脚有效沿时清零& 启动	—	在 TI000 引脚和 TI010 引脚有 效沿产生中断请求
1	0	1			
1	1	0	当 TM00 和 CR000 相等时清零 &启动	当 TM00 和 CR000 相等, 或 TM00 和 CR010 相等时反转	
1	1	1		当 TM00 和 CR000 相等时, 或 TM00 与 CR010 相等时, 或 TI000 引脚有效沿时反转	

OVF00	16 位定时器计数器 00 (TM00) 的溢出检测
0	未检测到溢出
1	检测到有溢出

- 注意事项**
1. 向 16 位定时器模式控制寄存器 TMC00 的各位写入数据前, 定时器必须停止工作, OVF00 标志位除外。
 2. 如果定时器停止, 即使有信号输入 TI000/TI010 引脚, 不会进行时钟计数, 也不会产生定时器中断。
 3. 除 TI000 引脚有效沿作为计数时钟以外, 在设置 STOP 模式或系统时钟停止模式之前, 停止定时器工作; 否则在系统时钟启动时, 定时器将发生故障。
 4. 当定时器停止工作后, 才能通过预分频模式寄存器 00 (PRM00) 的第 4 位和第 5 位来设置 TI000 引脚有效沿。
 5. 如果在 TM00 和 CR00 相等进入清零&启动模式时, 或因 TI000 有效沿进入清零&启动模式时, 或者定时器工作在自由工作模式时, 当 CR000 的值为 FFFFH, TM00 的值从 FFFFH 变为 0000H 时, OVF00 标志位置 1。
 6. 即便在 TM00 溢出之后, 而在下一个计数时钟计数之前 (TM00 变为 0001H 之前), 清除 OVF00 标志, OVF00 标志还会被重新置 1, 清除操作无效。
 7. 在计数时钟的下降沿时执行捕获操作, 在下个计数时钟的上升沿, 会产生一个中断请求输入 (INTTM0n0)。

备注

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

(2) 捕获/比较控制寄存器 00 (CRC00)

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

CRC00 由 1 位或 8 位存储器操作指令设置。

复位信号的产生, 会将 CRC00 值清为 00H。

图 6-6. 捕获/比较控制寄存器 00 (CRC00) 的格式

地址: FF62H 复位后: 00H 读/写

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

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

CRC001	CR000 捕获触发信号选择
0	TI010 引脚有效沿触发捕获
1	当 TI000 引脚有效沿反向 ^① 时触发捕获

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

注 当 CRC001 位的值为 1, 如果 TI000 的上升沿和下降沿均被选作有效触发沿, 则捕获寄存器不工作。

- 注意事项
1. 设置 CRC00 之前, 定时器操作必须停止。
 2. 当通过 16 位定时器模式控制寄存器 TMC00 选择 TM00 和 CR000 相等进入清零&启动模式时, CR000 就不能被指定为捕获寄存器。
 3. 为了确保捕获操作的可靠性, 要求被捕获的信号宽度要大于 2 个计数时钟 (由预分频器模式寄存器 00 (PRM00) 选择) (参见图 6-17)。

(3) 16 位定时器输出控制寄存器 00 (TOC00)

该寄存器用于控制 16 位定时器/事件计数器输出的操作。它可以通过软件设置定时器的输出 F/F 置位/复位，输出反相允许/禁止，16 位定时器/事件计数器 00 的定时器输出允许/禁止，单脉冲输出操作允许/禁止，以及设置单脉冲输出触发。

TOC00 由 1 位或 8 位存储器操作指令设置。

复位信号的产生，会将 TOC00 值为 00H。

图 6-7. 16 位定时器输出控制寄存器 00 (TOC00) 的格式

地址: FF63H 复位后: 00H 读/写

符号	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
OSPT00		由软件控制的单脉冲输出触发						
0		无单脉冲输出触发						
1		触发单脉冲输出						
OSPE00		单脉冲输出操作控制						
0		连续脉冲输出模式						
1		单脉冲输出模式 ^注						
TOC004		CR010 与 TM00 相等作为定时器输出 F/F 控制						
0		禁止反相操作						
1		允许反相操作						
LVS00		LVR00	定时器输出 F/F 状态设置					
0		0	无改变					
0		1	定时器输出 F/F 复位（0）					
1		0	定时器输出 F/F 置位（1）					
1		1	禁止设置					
TOC001		CR000 与 TM00 相等作为定时器输出 F/F 控制位						
0		禁止反相操作						
1		允许反相操作						
TOE00		定时器输出控制						
0		禁止输出（输出固定为低电平）						
1		允许输出						

注 只有在自由运行模式和 TI000 引脚输入有效沿进入清零&开始模式下，单脉冲输出模式才能够正常操作。在 TM00 和 CR000 相等时进入的清零&开始模式下，由于不会发生溢出，所以单脉冲输出也不可能发生。

- 注意事项**
- 除了 OSPT00 以外，对其它寄存器设置之前，定时器必须停止工作。
 - 当读取 LVS00 和 LVR00 时，读出的值为 0。
 - 数据设置之后，OSPT00 自动清零，读取该位时，读出的值为 0。
 - 除非在单脉冲输出模式下，OSPT00 不可以设置为 1。
 - 当连续设置 OSPT00 为 1 时，需要用预分频模式寄存器 00 (PRM00) 选择两个或更多计数时钟周期的写入间隔。
 - 当 TOE00 为 0 时，设置 TOE00, LVS00, 和 LVR00, 要用 8 位存储器操作指令同时设置。当 TOE00 为 1 时，LVS00, LVR00 能够用 1 位存储器操作指令设置。

(4) 预分频模式寄存器 00 (PRM00)

该寄存器用于设置 16 位定时器计数器 00 (TM00) 的计数时钟和 TI000, TI010 引脚输入的有效沿。
PRM00 由 1 位或 8 位存储器操作指令设置。
复位信号的产生, 会将 PRM00 值清为 00H。

图 6-8. 预分频模式寄存器 00 (PRM00) 的格式

地址: FF61H 复位后: 00H 读/写

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

ES110	ES100	TI010 引脚有效沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	下降沿和上升沿均为有效沿

ES010	ES000	TI000 引脚有效沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	下降沿和上升沿均为有效沿

PRM001	PRM000	计数时钟选择
0	0	f_{XP} (10 MHz)
0	1	$f_{XP}/2^2$ (2.5 MHz)
1	0	$f_{XP}/2^8$ (39.06 kHz)
1	1	TI000 引脚有效沿 ^注

备注 1. f_{XP} : 提供给外围设备的时钟振荡频率
 2. () : $f_{XP} = 10\text{ MHz}$

注 外部时钟要求脉冲的长度大于两个内部计数时钟 (f_{XP}) 周期。

- 注意事项
1. 定时器停止工作后，才可以向 **PRM00** 写入数据。
 2. 如果指定 **TI000** 引脚的有效沿作为计数时钟，就不要设置在 **TI000** 引脚的有效沿时进入清零/启动模式和捕获触发器。
 3. 在如下情况时，要注意 **TI0n0** 引脚有效沿的检测。
 - <1> 系统复位后，如果 **TI0n0** 引脚立即输入高电平，那么允许 16 位定时器计数器 00 (**TM00**) 操作。
 - 如果上升沿或双沿被指定为 **TI0n0** 引脚的有效沿，**TM00** 操作允许后，马上会检测到一个上升沿。
 - <2> 如果当 **TI0n0** 引脚是高电平的时候，停止 **TM00** 操作，则 **TI0n0** 引脚输入低电平后，允许 **TM00** 操作。
 - 如果下降沿或双沿被指定为 **TI0n0** 引脚的有效沿，**TM00** 操作允许后，马上会检测到一个下降沿。
 - <3> 当 **TI0n0** 为低电平，**TM00** 停止工作时，则 **TI0n0** 引脚输入高电平后，允许 **TM00** 操作。
 - 如果上升沿或双沿被指定为 **TI0n0** 引脚的有效沿，**TM00** 操作允许后，马上会检测到一个上升沿。
 4. 当 **TI000** 有效沿用作计数时钟和用作捕获触发时，用来消除噪音的采样时钟是不同的。在前一种情况中，计数时钟为 f_{XP} 。而在后一种情况中，由预分频模式寄存器 00 (**PRM00**) 选择计数时钟。直到采样到有效沿，并且连续两次检测到有效电平，捕获操作才会完成，因此可以消除具有窄脉冲宽度的噪声。
 5. 当 **P31** 用做 **TI010** 的有效沿输入引脚时，不能用作定时器输出引脚 (**TO00**)。同样，当 **P31** 被用于定时器输出引脚 **TO00** 时，也不能用于 **TI010** 的有效沿输入引脚。

备注 $n = 0, 1$

(5) 端口模式寄存器 3 (**PM3**)

该寄存器可以逐位设置端口 3 的输入输出模式

当 **P31/TO00/TI010/INTP2** 引脚作为定时器输出使用时，设置 **PM31** 为 0，**P31** 的输出锁存器为 0。

当将 **P30/TI000/INTP0** 和 **P31/TO00/TI010/INTP2** 引脚作为定时器输入时，设置 **PM30** 和 **PM31** 为 1。

此时，**P30** 和 **P31** 的输出锁存可能为 0 或 1。

PM3 由 1 位或 8 位存储器操作指令设置。

复位信号的产生，会将 **PM3** 置为 **FFH**。

图 6-9. 端口模式寄存器 3 (**PM3**) 的格式

地址: **FF23H** 复位后: **FFH** 读/写

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30

PM3n	P3n 引脚的输入/输出模式选择 ($n = 0 \sim 3$)
0	输出模式 (输出锁存打开)
1	输入模式 (输出锁存关闭)

6.4 16 位定时器/事件计数器 00 的操作

6.4.1 时间间隔定时器操作

通过设置 16 位定时器模式控制寄存器 00 (TMC00) 和捕获/比较控制寄存器 00 (CRC00)，定时器可作为时间间隔定时器使用，如图 6-10 所示。

设置

基本操作的设置步骤如下：

- <1> 设置 CRC00 寄存器（设置值参考图 6-10）。
- <2> 设置 CR000 寄存器的值。
- <3> 通过 PRM00 寄存器设置计数时钟。
- <4> 设置 TMC00 寄存器，启动操作（设置值参考图 6-10）。

注意事项 如果在 TM00 操作中改变 CR000 的设置，可能引起错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 00 的相关注意事项（17）在定时器操作过程中改变比较寄存器。

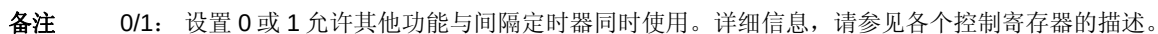
备注 如何允许 INTTM000 中断，请参见第十二章 中断功能。

以 16 位定时器捕获/比较寄存器 000 (CR000) 中预先设定的值作为间隔，重复产生中断请求。

当 16 位定时器 TM00 的计数值和 CR000 的设定值相等时，TM00 中的值清零，然后继续计数，同时产生中断请求信号 (INTTM000)。

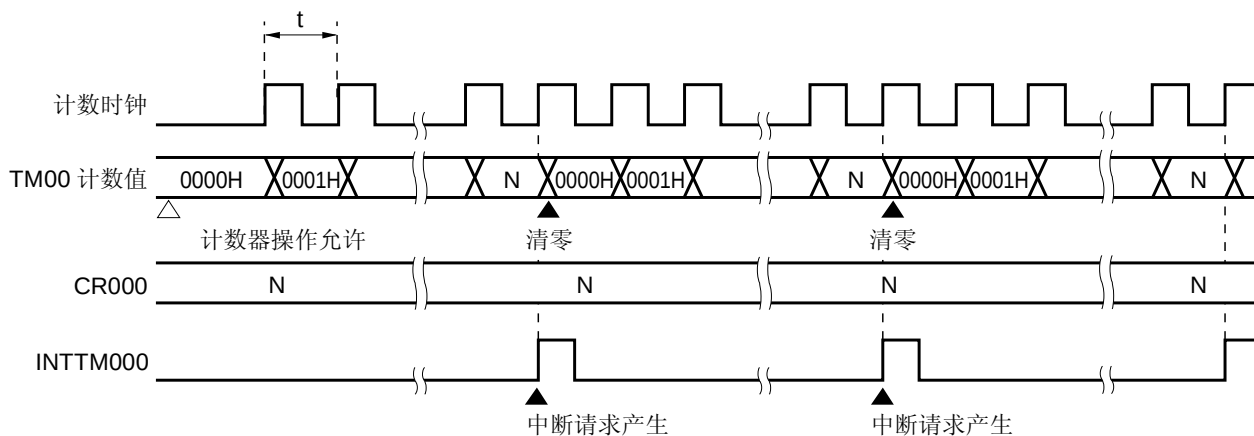
16 位定时器/计数器的计数时钟可以通过预分频模式寄存器 PRM00 的第 0 位和第 1 位 (PRM000, PRM001) 设置。

(a) 捕获/比较控制寄存器 00 (CRC00)



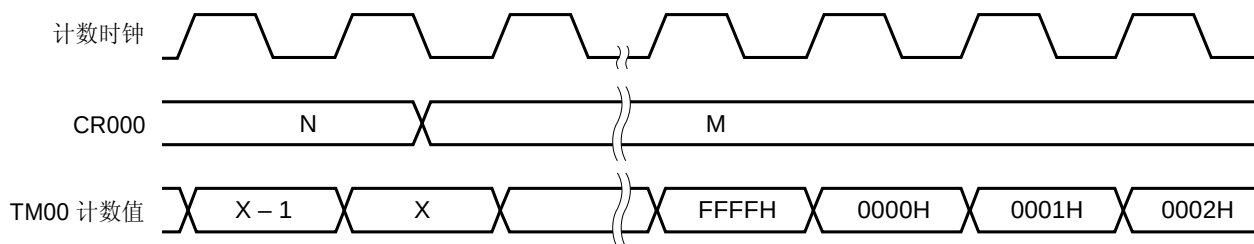
注 只有当 16 位定时器捕获/比较寄存器 000 (CR000) 设置为 FFFFH 时, OVF00 标志位置 1。

图 6-12. 间隔定时器操作的时序



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

当定时器计数过程中比较寄存器被改变时，如果 16 位定时器捕获/比较寄存器 CR000 改变后的新值小于 16 位定时器 TM00 的值，则 TM00 继续计数，溢出然后从 0 开始计数。因此，如果 CR000 改变后的值 (M) 小于改变前的值 (N)，那么在 CR000 改变之后，需要重新启动定时器。

图 6-13. 在定时器计数操作过程中比较寄存器改变之后的时序 ($N \rightarrow M$: $N > M$)

备注 $N > X > M$

6.4.2 外部事件计数器操作

设置

基本操作的设置步骤如下：

- <1> 设置 CRC00 寄存器（设置值参考图 6-14）。
- <2> 通过 PRM00 寄存器设置计数时钟。
- <3> 设置 CR000 寄存器的值（不能设置成 0000H）。
- <4> 设置 TMC00 寄存器，启动操作（设置值参考图 6-14）。

备注 1. 对于 TI000 引脚的设置，请参见 6.3 (5) 端口模式寄存器 3 (PM3)。
 2. 如何允许 INTTM000 中断，参见第十三章 中断功能。

外部事件计数器使用 16 位定时器计数器 00 (TM00) 对 TI000 引脚输入的外部时钟脉冲进行计数。

每当预分频模式寄存器 00 (PRM00) 指定的有效沿输入时, TM00 递增。

当 TM00 的计数值与 16 位定时器捕获/比较寄存器 000 (CR000) 的值相等时, TM00 清零, 并且产生中断请求信号 (INTTM000)。

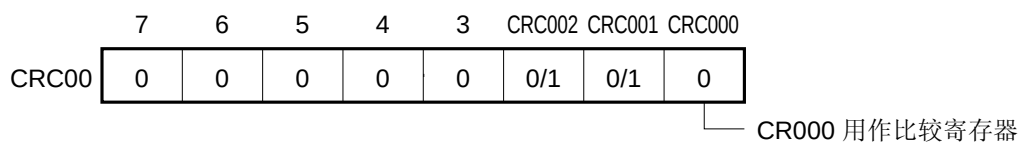
CR000 不能赋值为 0000H。(单个脉冲的一次计数操作不能完成。)

通过设置预分频模式寄存器 PRM00 的第 4, 5 位 (ES000, ES010) 来选择上升沿, 下降沿或上升沿和下降沿均为有效沿。

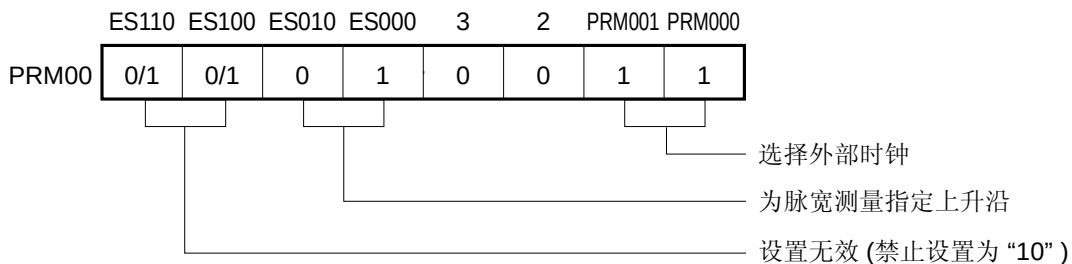
因为只有当使用内部时钟 (f_{XP}) 采样检测到两次 TI000 引脚有效沿时, 操作才能进行, 所以可以消除具有窄脉冲宽度的噪声。

图 6-14. 外部事件计数器模式控制寄存器的配置 (指定上升沿)

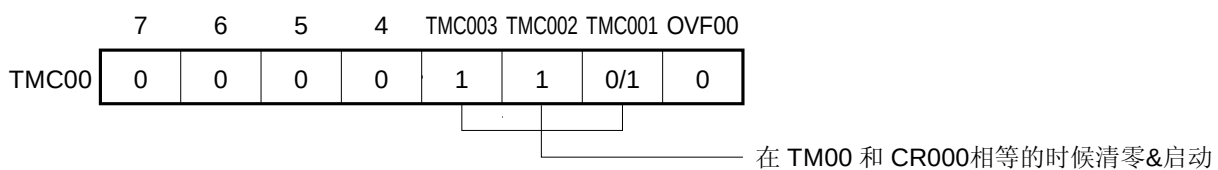
(a) 捕获/比较控制寄存器 00 (CRC00)



(b) 预分频模式寄存器 00 (PRM00)

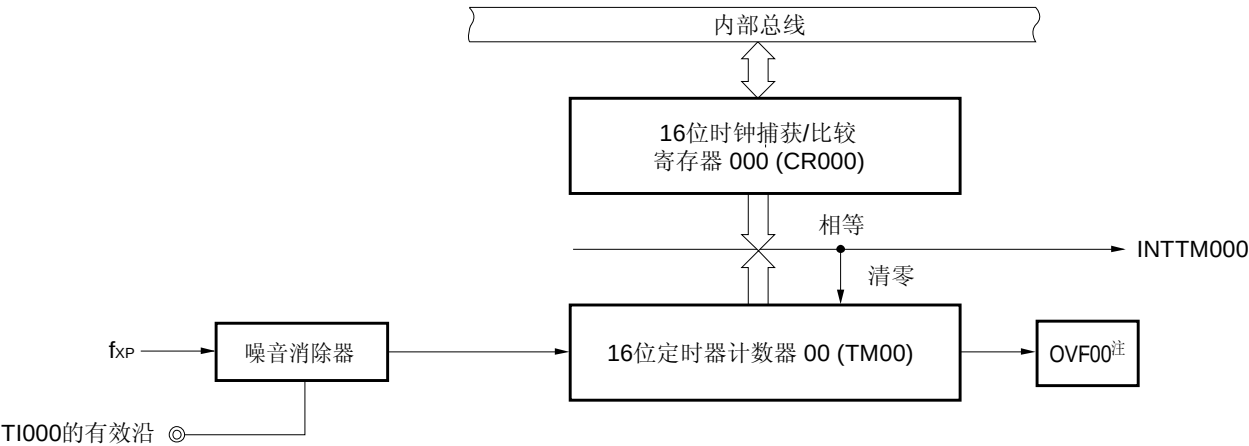


(c) 16 位定时器模式控制寄存器 00 (TMC00)



备注 0/1: 设置 0 或 1 允许其他功能与外部时间计数器同时使用。详细信息, 请参见各个控制寄存器的描述。

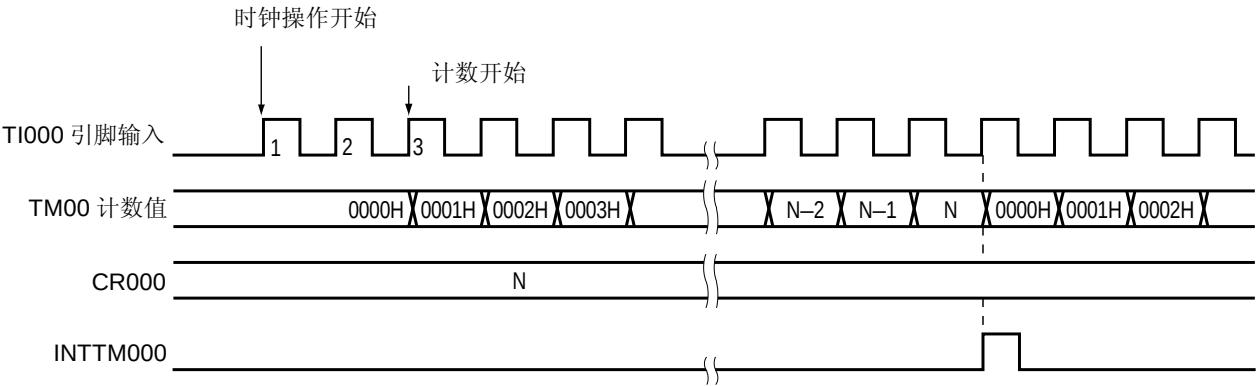
图 6-15. 外部事件计数器配置框图



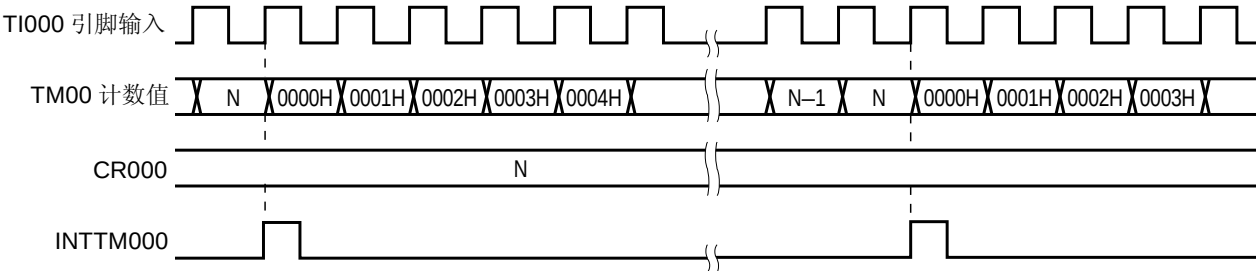
注 只有当 16 位捕获/比较寄存器 CR000 中的值为 FFFFH 时，OVF00 标志位置 1。

图 6-16. 外部事件计数器操作时序（指定上升沿）

(1) 操作启动之后立即产生 INTTM000 的时序
检测到两次有效沿之后，开始计数



(2) INTTM000 产生两次之后 INTTM000 产生的时序



注意事项 需要知道外部事件计数器的计数值，则应该去读 TM00。

6.4.3 脉冲宽度测量操作

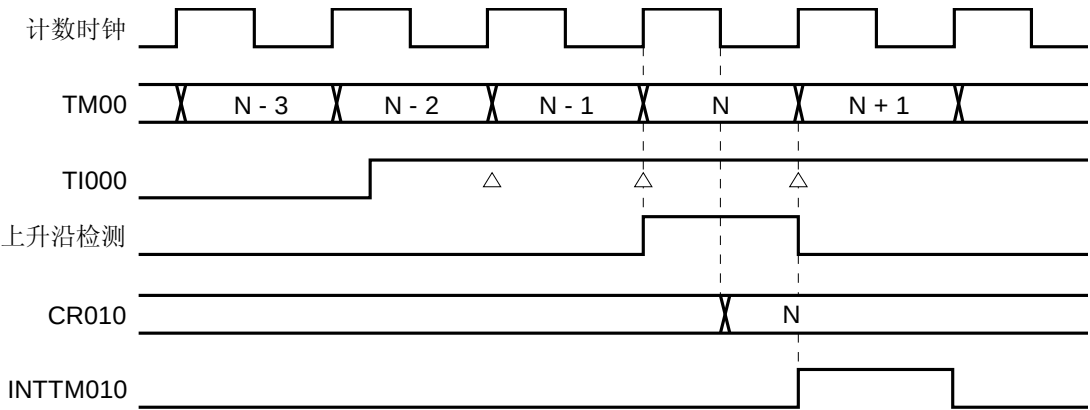
通过 16 位定时器计数器 00（TM00）可测量输入 TI000 和 TI010 引脚的信号脉冲宽度。

有两种测量方法：在自由运行模式下，使用 TM00 测量；或者在 TI000 引脚的输入信号边沿时同步重启定时器进行测量。

当发生中断时，读取捕获寄存器的有效值，检测溢出标志位，然后计算所需测量的脉冲宽度。检测之后，溢出标志位清零。

直到在预分频模式寄存器 00（PRM00）选择的计数时钟周期内采样到信号脉冲宽度，且两次检测到 TI000 或 TI010 的有效电平之后，才执行捕获操作，因此可以消除具有窄脉冲宽度的噪音。

图 6-17. 指定上升沿的 CR010 捕获操作



设置

基本操作的设置步骤如下：

- <1> 设置 CRC00 寄存器（设置值参见图 6-18， 6-21， 6-23， 和 6-25）。
- <2> 通过 PRM00 寄存器设置计数时钟。
- <3> 设置 TMC00 寄存器启动操作（设置值参见图 6-18， 6-21， 6-23， 和 6-25）。

注意事项 要使用两个捕获寄存器时，设置 TI000 和 TI010 引脚。

- 备注**
- 1. 如需设置 TI000（或 TI010）引脚，参见 6.3（5） 端口模式寄存器 3（PM3）。
 - 2. 关于如何允许 INTTM000（或 INTTM010）中断，参见第十三章 中断功能。

(1) 用自由运行计数器和捕获寄存器测量脉宽

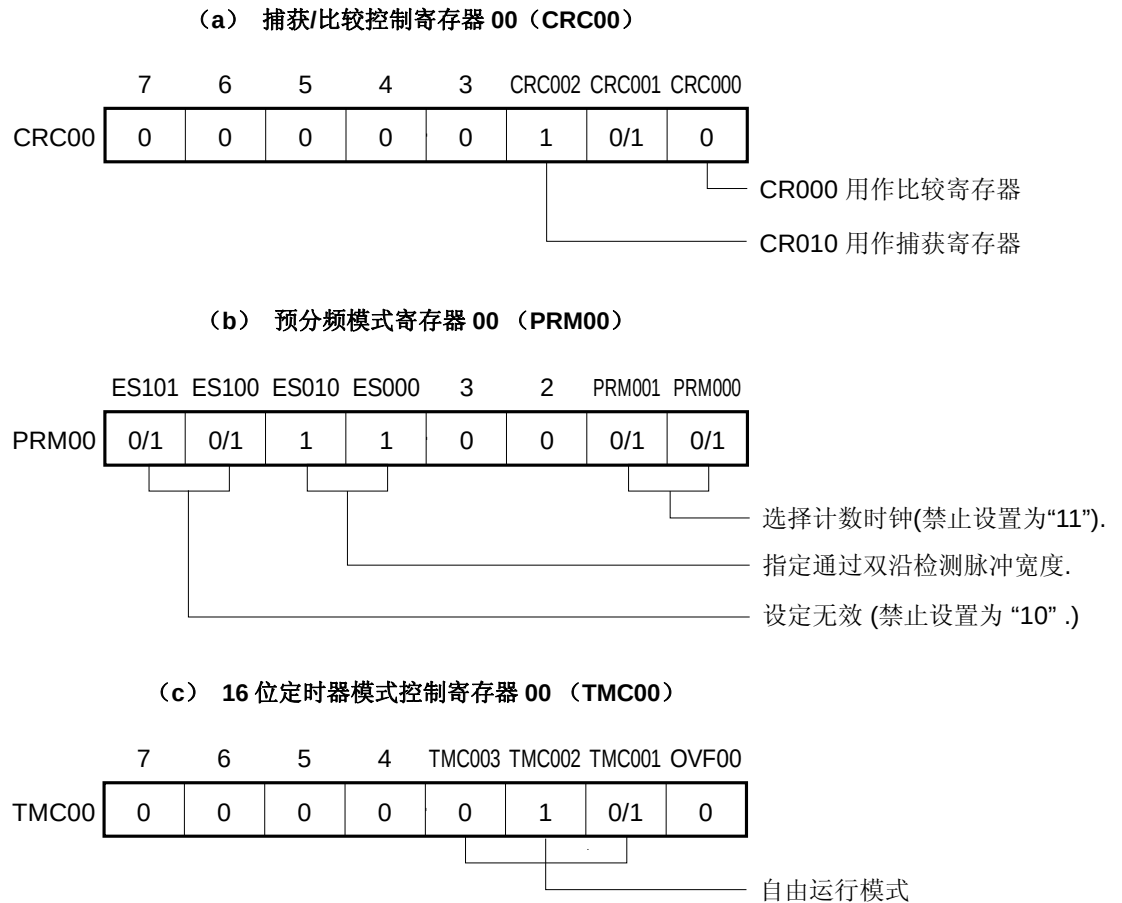
通过 PRM00 的第 4 位和第 5 位 (ES000 和 ES010)，可以指定 TI000 引脚的上升沿和下降沿均为有效沿。

如果 16 位定时器计数器 00 (TM00) 在自由运行模式下工作，且预分频模式寄存器 00 (PRM00) 指定的有效脉冲沿输入到 TI000 引脚时，TM00 的值被送入 16 位定时器捕获/比较寄存器 010 (CR010)，并产生外部中断请求信号 (INTTM010)。

使用由 PRM00 选择的计数时钟进行采样，并且仅当 TI000 引脚检测到两次有效电平才执行捕获操作，因此可以消除具有窄脉冲宽度的噪音。

注意事项 在这个操作示例中，可测量的脉冲宽度可达一个定时计数器周期。

图 6-18. 通过自由运行计数器和一个捕获寄存器测量脉冲宽度的控制寄存器设置（使用 TI000 和 CR010 时）



备注 0/1: 设置 0 或 1 允许其他功能与脉冲宽度测量同时使用。详细信息，请参见各个控制寄存器的描述。

图 6-19. 通过自由运行计数器测量脉冲宽度的配置框图

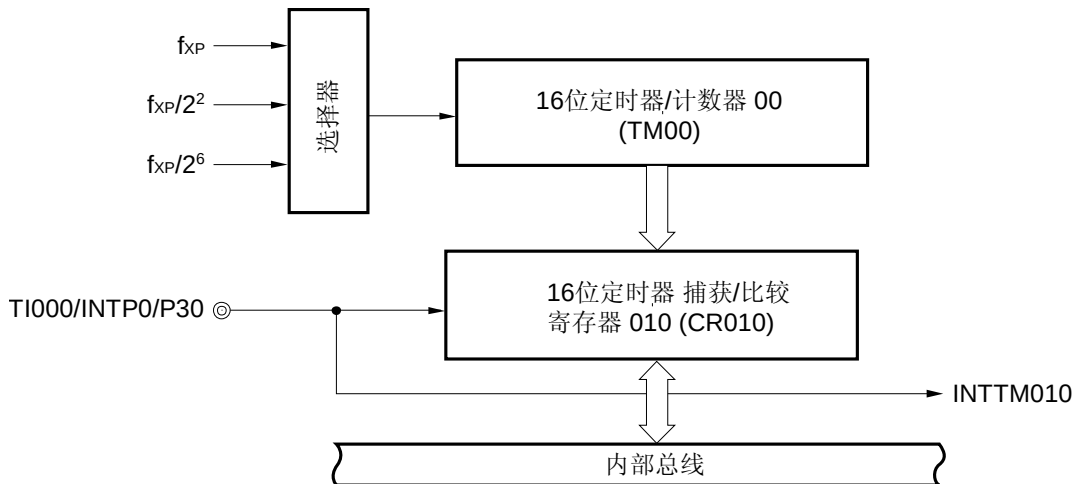
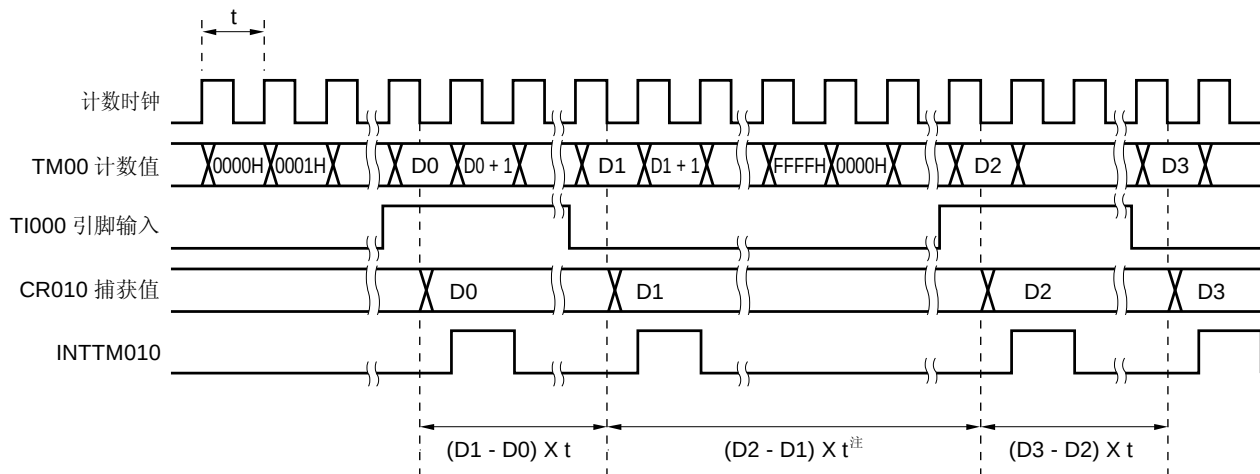


图 6-20. 通过自由运行计数器和一个捕获寄存器（双沿有效）测量脉冲宽度的时序图



注 进位标志位置 1，忽略此设置。

(2) 使用自由运行计数器测量两个脉冲的脉宽

当 16 位定时器 TM00 工作在自由运行模式下时，定时器 TM00 可以同时测量由 TI000 引脚和 TI010 引脚输入的两个信号的脉冲脉宽。

通过 PRM00 的第 4, 5 位 (ES000, ES010) 和第 6, 7 位 (ES100, ES110)，可以指定 TI000 和 TI010 引脚的上升沿和下降沿均为有效沿。

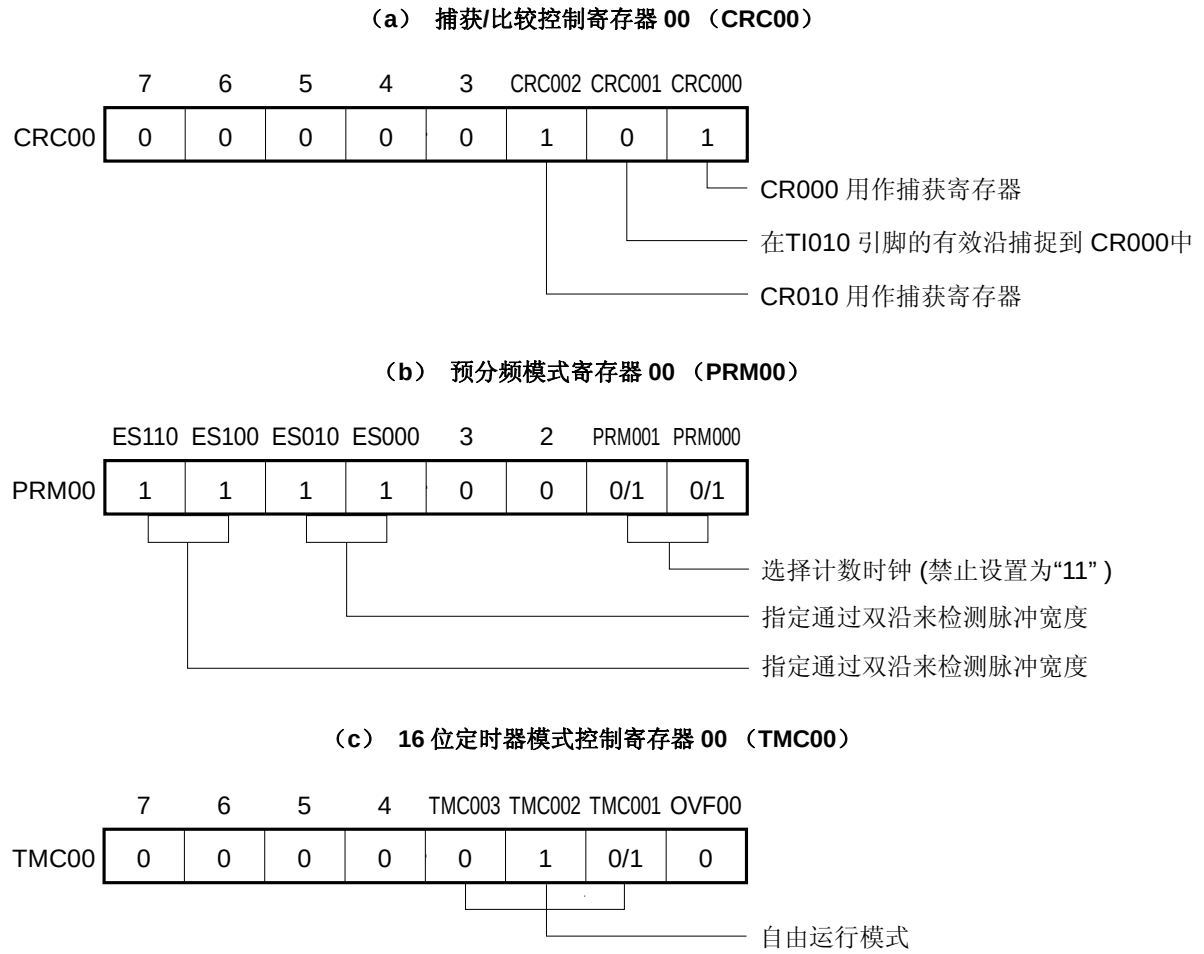
当有效沿（由 PRM00 的第 4, 5 位 (ES000 和 ES010) 定义）输入到 TI000 引脚时，TM00 的值写入 16 位定时器捕获/比较寄存器 CR010，然后产生 INTTM010 中断。

另外，当有效沿（由 PRM00 的第 6, 7 位 (ES100 和 ES110) 定义）输入到 TI010 引脚时，TM00 的值写入 16 位定时器捕获/比较寄存器 CR000，然后产生 INTTM000 中断。

根据预分频模式寄存器 00 (PRM00) 选择的计数时钟周期进行采样，并且在 TI000 或 TI010 引脚检测到两次有效电平之后，才可执行捕获操作，因此可以消除具有窄脉冲宽度的噪音。

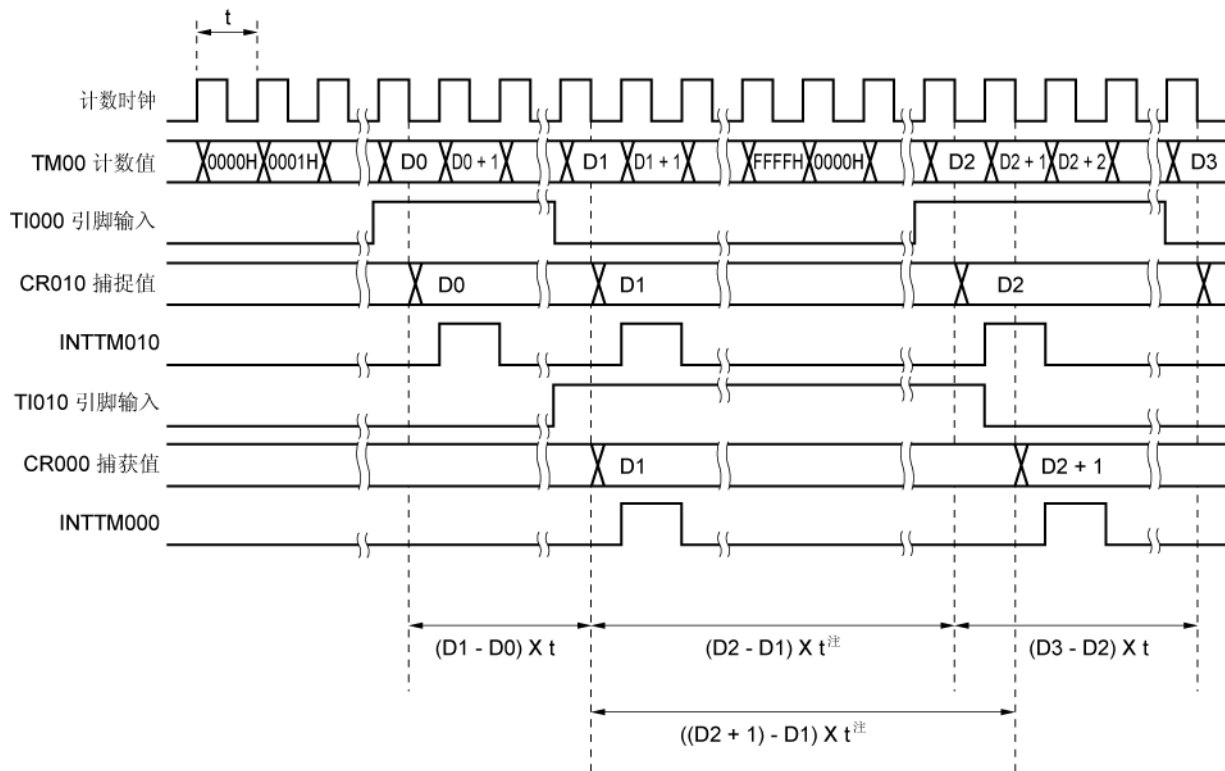
注意事项 在这个操作示例中，可测量的脉冲宽度可达一个定时计数器周期。

图 6-21. 使用自由运行计数器测量两个脉冲宽度的控制寄存器设置



备注 0/1: 设置 0 或 1 允许其他功能与脉冲宽度测量同时使用。详细信息, 请参见各个控制寄存器的描述。

图 6-22. 使用自由运行计数器测量脉冲宽度的时序图（指定双沿有效）



注 进位标志位置 1，忽略此设置。

(3) 使用自由运行计数器和两个捕获寄存器进行脉冲宽度测量

当 16 位定时器 TM00 工作在自由运行模式下时，可以测量由 TI000 引脚输入的信号的脉宽。

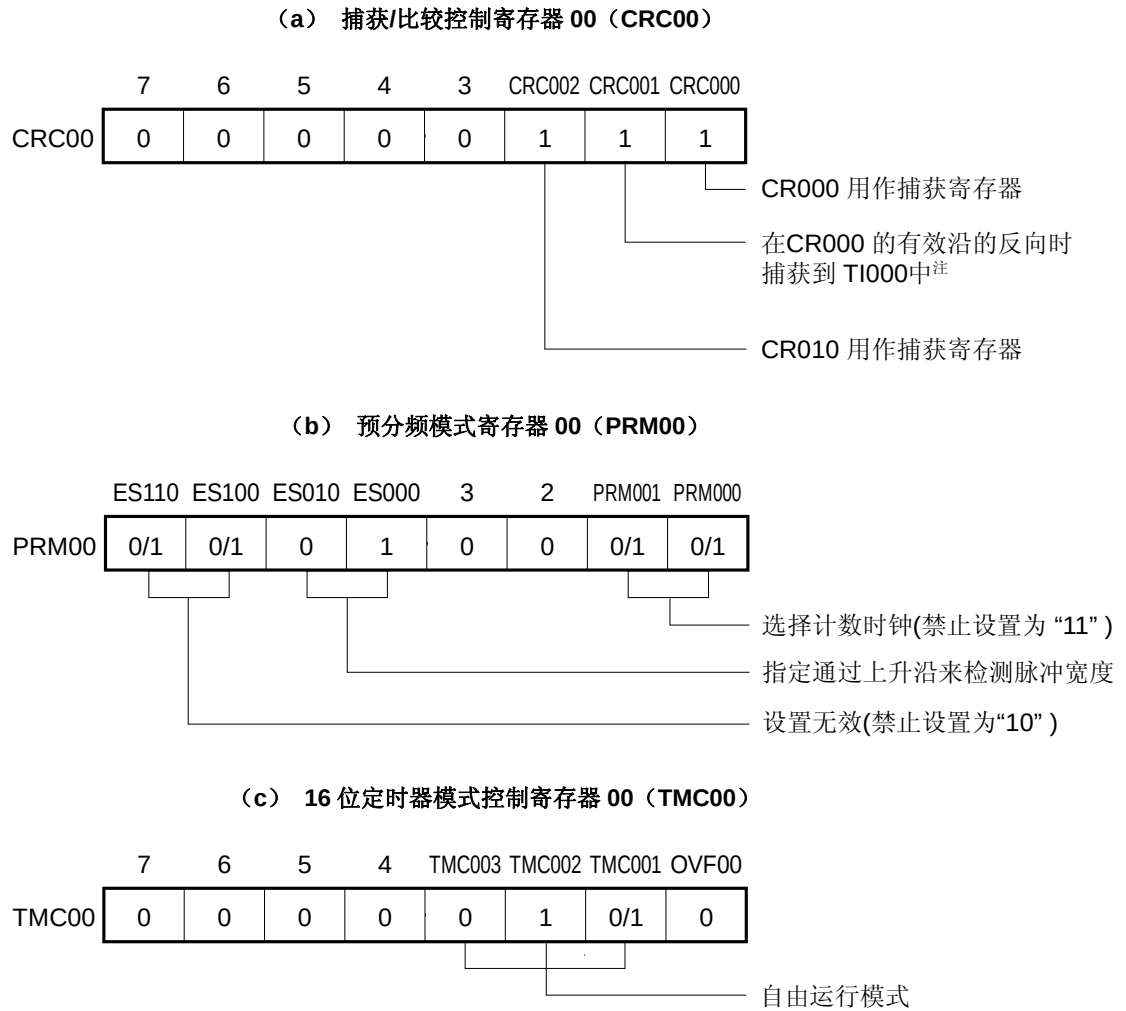
通过 PRM00 的第 4 位和第 5 位（ES000 和 ES010）指定的引脚有效沿输入到 TI000 引脚时，TM00 的值写入 16 位定时器捕获/比较寄存器 CR010，然后产生 INTTM010 中断。

另外，当捕获操作的反向有效沿输入 CR010 的时，TM00 的值被送入 16 位定时器捕获/比较寄存器 000（CR000）。

根据预分频模式寄存器 00（PRM00）选择的计数时钟周期进行采样，并且在 TI000 的引脚检测到两次有效电平之后，才可执行捕获操作，因此可以消除具有窄脉冲宽度的噪音。

注意事项 在这个操作示例中，可测量的脉冲宽度可达一个定时计数器周期。

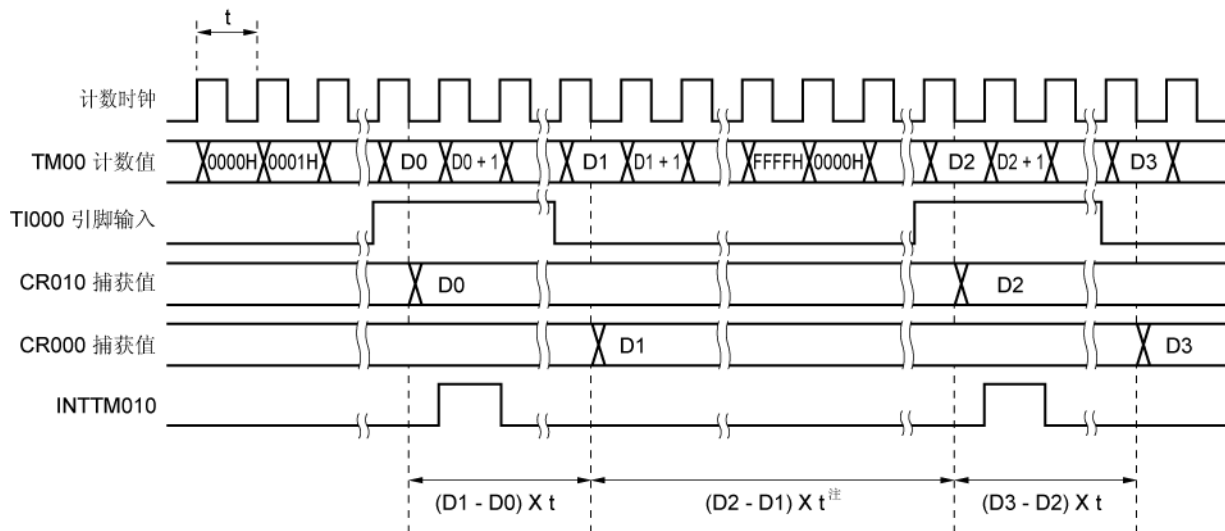
图 6-23. 使用自由运行计数器和两个捕获寄存器进行脉冲宽度测量的控制寄存器设置（指定上升沿有效）



注 如果指定 TI000 引脚的上升沿和下降沿均为有效沿，16 位定时器捕获/比较寄存器 0000 (CR000) 不能完成捕获操作。当 CRC001 值为 1 时，如果 TI010 引脚检测到有效沿，TM00 的计数值不能被捕获到 CR000 寄存器中，但是 TI010 引脚的输入可以用作外部中断源，因为外部有效信号输入时产生 INTTM000 中断。

备注 0/1: 设置 0 或 1 允许其他功能与脉冲宽度测量同时使用。详细信息，请参见各个控制寄存器的描述。

图 6-24. 通过自由运行计数器和两个捕获寄存器测量脉冲宽度的时序图（指定上升沿有效）



注 进位标志位置 1，忽略此设置。

(4) 通过重启进行脉冲宽度测量

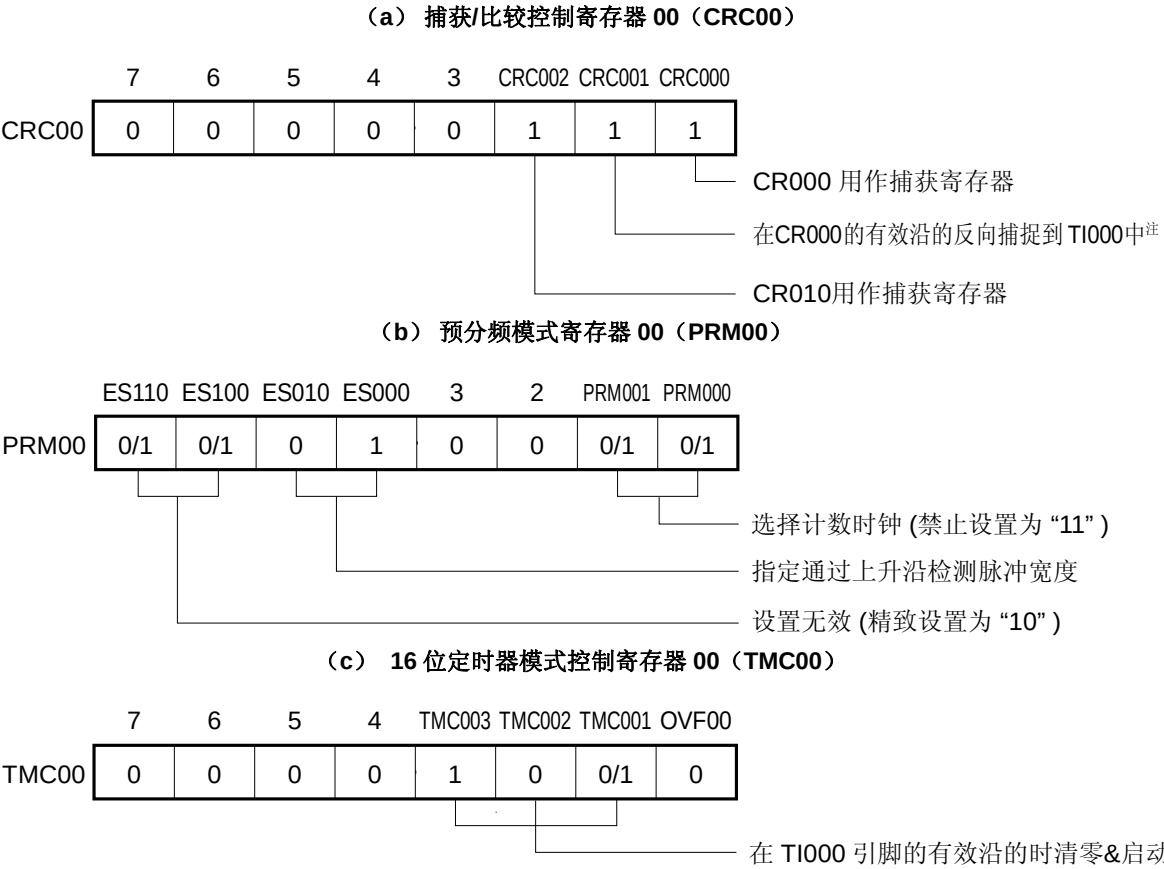
通过 PRM00 的第 4 位和第 5 位 (ES000 和 ES010)，可以指定 TI000 引脚的上升沿和下降沿均为有效沿。

当检测到有效沿输入 TI000 引脚时，将 16 位定时器计数器 00 (TM00) 的计数值送入 16 位定时器捕获/比较寄存器 010 (CR010)，然后对 TM00 清零并重新进行计数操作，来测量输入到 TI000 引脚的信号的脉冲宽度。

根据预分频模式寄存器 00 (PRM00) 选择的计数时钟周期进行采样，并且在 TI000 引脚检测到两次有效电平之后，才可执行捕获操作，因此可以消除具有窄脉冲宽度的噪音。

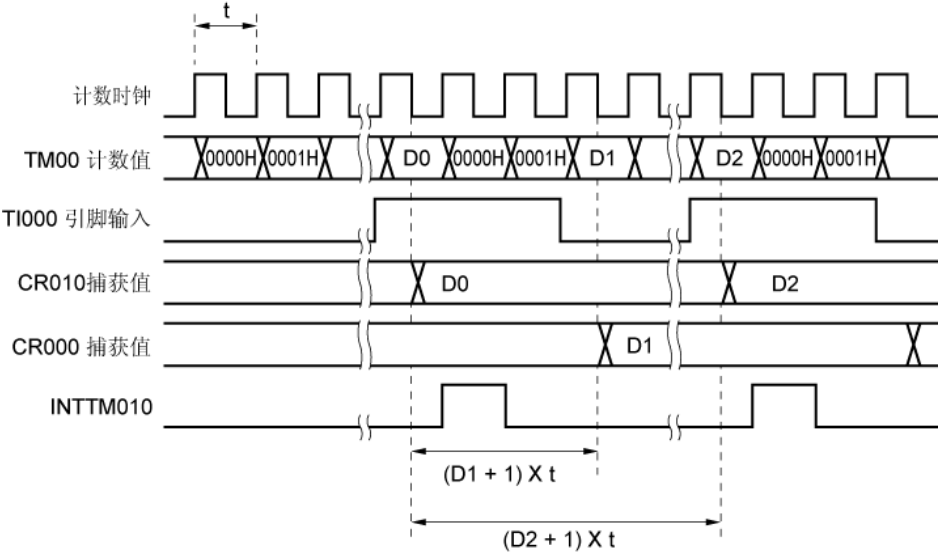
注意事项 在这个操作示例中，可测量的脉冲宽度可达一个定时计数器周期。

图 6-25. 通过重启进行测量脉冲宽度的控制寄存器设置（指定上升沿有效）



注 如果指定 TI000 引脚的上升沿和下降沿均为有效沿，16 位定时器捕获/比较寄存器 0000 (CR000) 不能完成捕获操作。

图 6-26. 通过重启进行测量脉冲宽度的时序（指定上升沿有效）



6.4.4 方波输出操作

设置

基本操作的设置步骤如下：

- <1> 通过 PRM00 寄存器设置计数时钟。
- <2> 设置 CRC00 寄存器（设置值参考图 6-27）。
- <3> 设置 TOC00 寄存器（设置值参考图 6-27）。
- <4> 设置 CR000 寄存器的值（不能设置成 0000H）。
- <5> 设置 TMC00 寄存器，启动操作（设置值参考图 6-27）。

注意事项 如果在 TM00 操作中改变 CR000 的设置，可能引起故障。要改变设置，请参考 6.5 16 位定时器/事件计数器 00 的相关注意事项（17）在定时器操作过程中改变比较寄存器。

- 备注**
- 1. 如需设置 TO00 引脚，参见 6.3（5）端口模式寄存器 3（PM3）。
 - 2. 关于如何允许 INTTM000 中断，参见第十三章 中断功能。

根据 16 位定时器捕获/比较寄存器 000（CR000）预置的计数值所确定的时间间隔，可输出任意频率的方波。

通过将 16 位定时器输出控制寄存器 00（TOC00）的第 0 位（TOE00）和第 1 位（TOC001）设置为 1，可使 TO00 引脚输出状态以预置的 CR000+1 为间隔进行翻转。这样可输出任意频率的方波。

图 6-27. 方波输出模式下控制寄存器的设置（1/2）

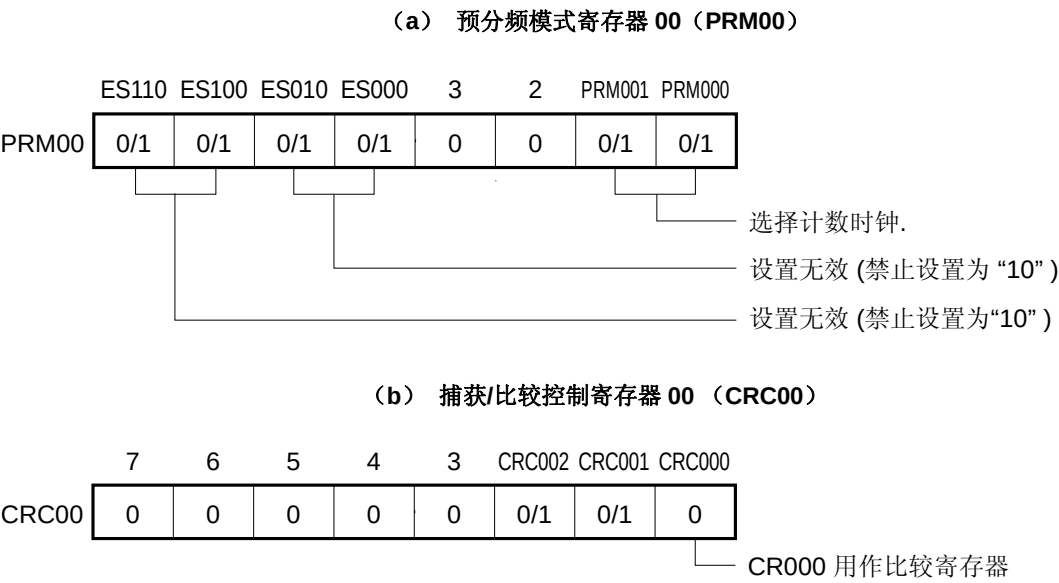
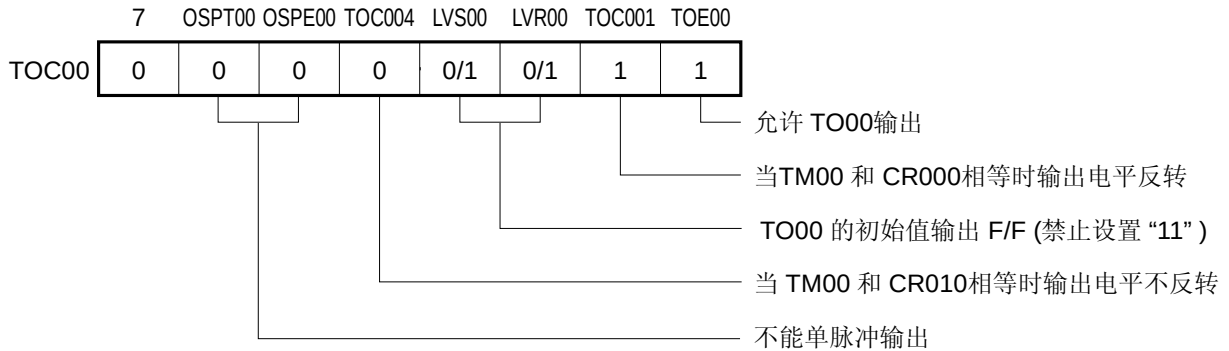
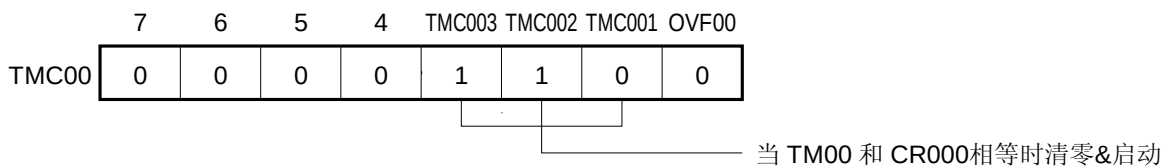


图 6-27. 方波输出模式下控制寄存器的设置 (2/2)

(c) 16 位定时器输出控制寄存器 00 (TOC00)

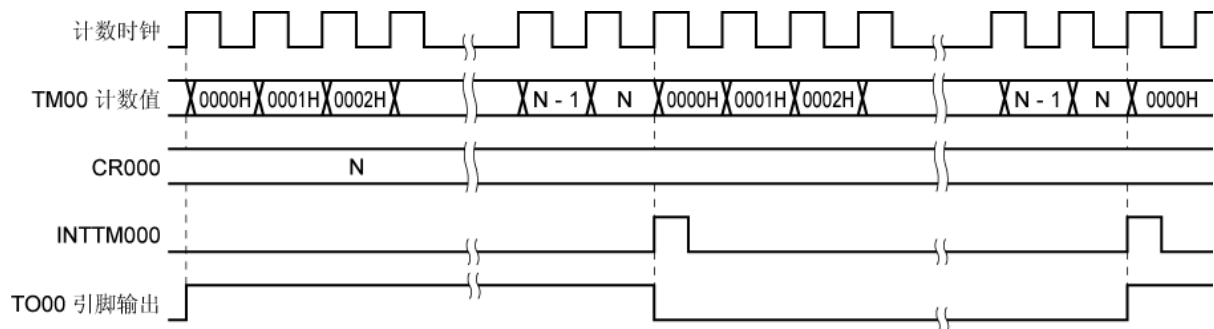


(d) 16 位定时器模式控制寄存器 (TMC00)



备注 0/1: 设置 0 或 1 允许其他功能与方波输出同时使用。详细信息, 请参见各个控制寄存器的描述。

图 6-28. 方波输出操作时序



6.4.5 PPG 输出操作

通过设置 16 位定时器模式控制寄存器（TMC00）和捕获/比较控制寄存器（CRC00），允许可编程脉冲发生器 PPG 输出的操作，详见图 6-29。

设置

基本操作的设置步骤如下：

- <1> 设置 CRC00 寄存器（设置值参考图 6-29）。
- <2> 设置 CR000 寄存器的值，作为周期。
- <3> 设置 CR010 寄存器为任意值，作为占空系数。
- <4> 设置 TOC00 寄存器（设置值参考图 6-29）。
- <5> 通过 PRM00 寄存器设置计数时钟。
- <6> 设置 TMC00 寄存器，启动操作（设置值参考图 6-29）。

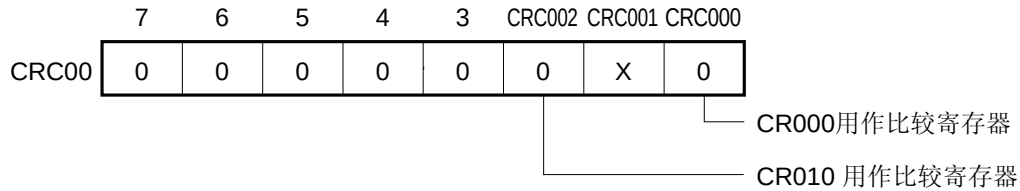
注意事项 如在 TM00 操作中改变 CR000 的设置，可能引起故障。要改变设置，请参考 6.5 16 位定时器/事件计数器 00 的相关注意事项（17）在定时器操作过程中改变比较寄存器。

- 备注**
- 1. 如需设置 TO00 引脚，参见 6.3（5）端口模式寄存器 3（PM3）。
 - 2. 关于如何允许 INTTM000 中断，参见第十三章 中断功能。
 - 3. $n = 0$ 或 1

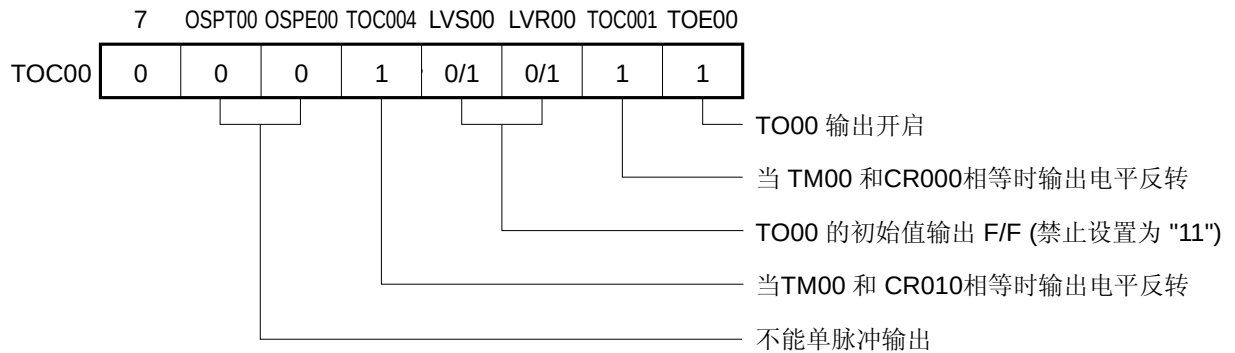
在 PPG 输出操作中，从 TO00 引脚输出矩形波，脉冲宽度和周期分别由 16 位定时器捕获/比较寄存器 010（CR010）和 16 位定时器捕获/比较寄存器 000（CR000）中的预置值来决定。

图 6-29. PPG 输出操作的控制寄存器的设置

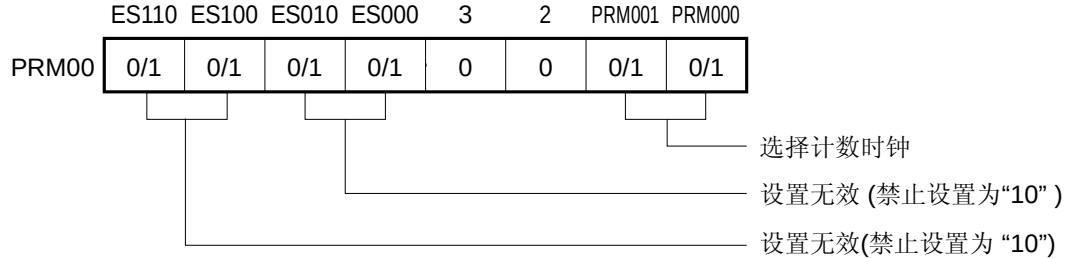
(a) 捕获/比较控制寄存器 00 (CRC00)



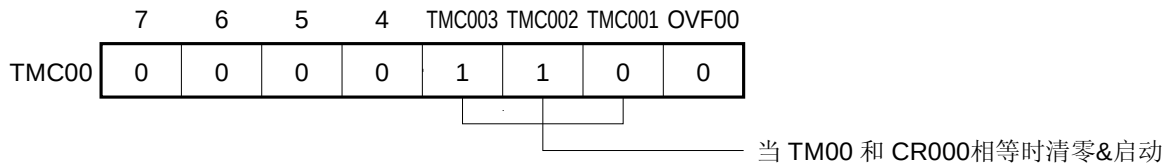
(b) 16 位定时器输出控制寄存器 00 (TOC00)



(c) 预分频模式寄存器 00 (PRM00)



(d) 16 位定时器模式控制寄存器 00 (TMC00)



- 注意事项
1. 以下为 CR000 和 CR010 的取值范围
 $0000H < CR010 < CR000 \leq FFFFH$
 2. 通过 PPG 输出产生的脉冲的周期 (CR000 的设置值 + 1) 的占空比是:
 $(CR010 \text{ 的设置值} + 1) / (CR000 \text{ 的设置值} + 1)$

备注 x: 无需考虑

图 6-30. PPG 输出的配置框图

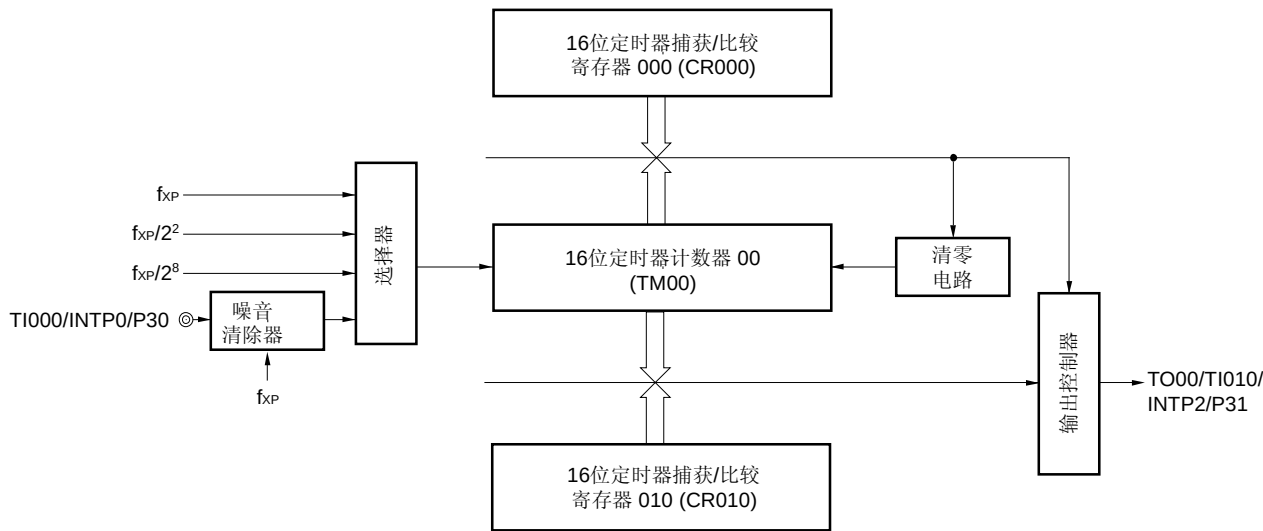
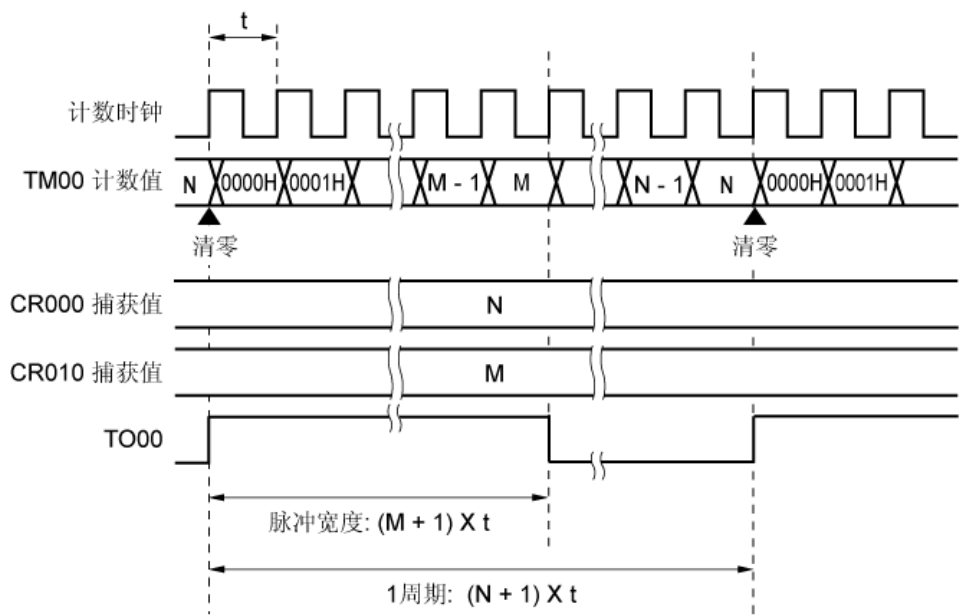


图 6-31. PPG 输出操作的时序



备注 0000H < M < N ≤ FFFFH

6.4.6 单脉冲输出操作

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

设置

基本操作的设置步骤如下：

- <1> 通过 PRM00 寄存器设置计数时钟。
- <2> 设置 CRC00 寄存器（设置值参考图 6-32 和 6-34）。
- <3> 设置 TOC00 寄存器（设置值参考图 6-32 和 6-34）。
- <4> 设置 CR000 和 CR010 寄存器的值（不可设置成 0000H）。
- <5> 设置 TMC00 寄存器启动操作（设置值参考图 6-32 和 6-34）。

- 备注**
- 1. 如需设置 TO00 引脚，参见 6.3（5） 端口模式寄存器 3（PM3）。
 - 2. 关于如何允许 INTTM000 中断（根据需要可选 INTTM010），参见第十三章 中断功能。

（1） 通过软件触发单脉冲输出

通过软件设置 16 位定时器模式控制寄存器 00（TMC00），捕获/比较控制寄存器 00（CRC00），16 位定时器输出控制寄存器 00（TOC00），再将 TOC00 寄存器的第 6 位设置成 1，可以输出单脉冲信号，如图 6-32 所示。

将 OSPT00 设置为 1，可将 16 位定时器/事件计数器 00 清零并启动。计数值与预先设置到 16 位定时器捕获/比较寄存器 010（CR010）的值（N）相等后，输出变为有效。此后，在计数值与预先设置到 16 位定时器捕获/比较寄存器 000（CR000）的值（M）相等后，输出变为无效。^注

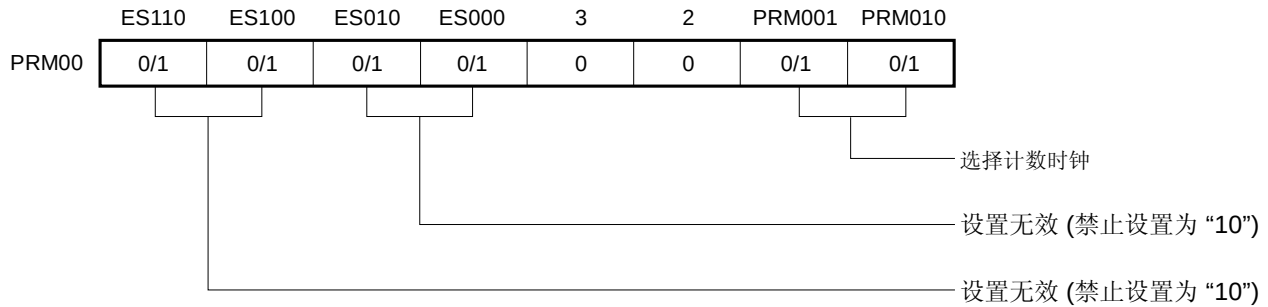
即使单脉冲已经输出后，寄存器 TM00 仍然在继续工作。要停止 TM00 寄存器，必须将 TMC00 寄存器的 TMC003 和 TMC002 清零。

- 注** 这里所描述的是 $N < M$ 的情况。当 $N > M$ 时，计数值与 CR000 相等，输出电平有效，而与 CR010 相等，输出电平无效。不要设置 N 等于 M 。

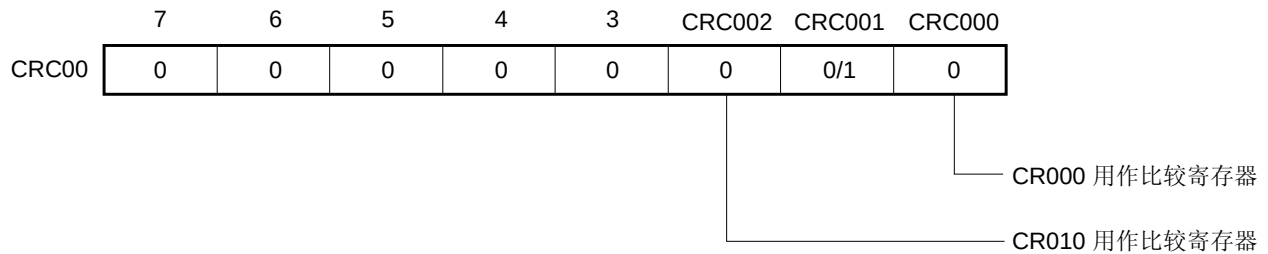
- 注意事项**
- 1. 在单脉冲正在输出情况下，不要再次置 OSPT00 位为 1。再次输出单脉冲，要等待正在进行的单脉冲输出完成之后。
 - 2. 使用软件触发进行 16 位定时器/事件计数器 00 的单脉冲输出时，不要改变 TI000 或其复用端口的引脚电平。
因为在这种情况下外部触发也有效，TI000 的电平 或其复用端口引脚的电平也可以对定时器清零并启动。这样会输出非预期的时序。

图 6-32. 软件触发的单脉冲输出的控制寄存器的设置

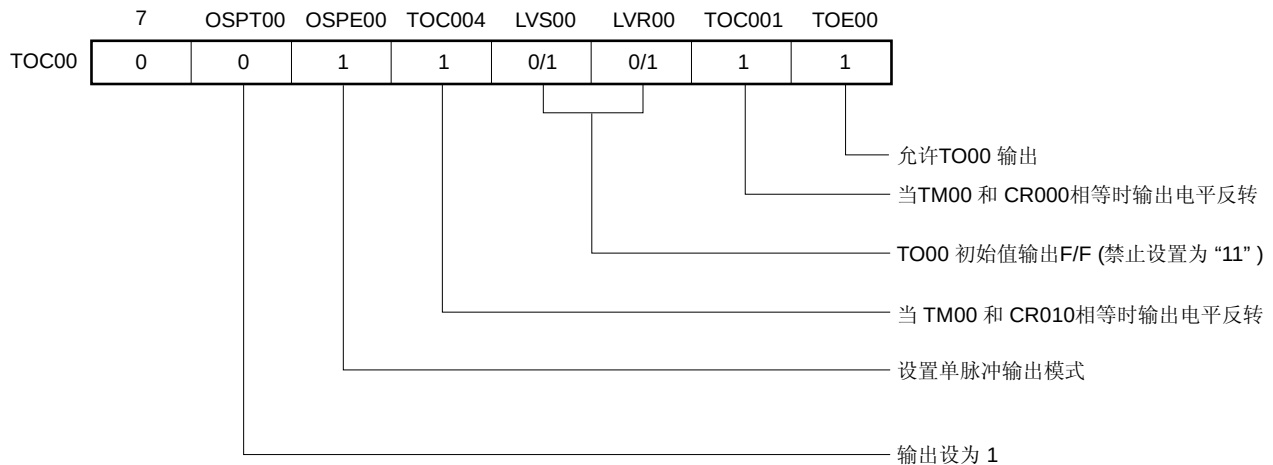
(a) 预分频模式寄存器 00 (PRM00)



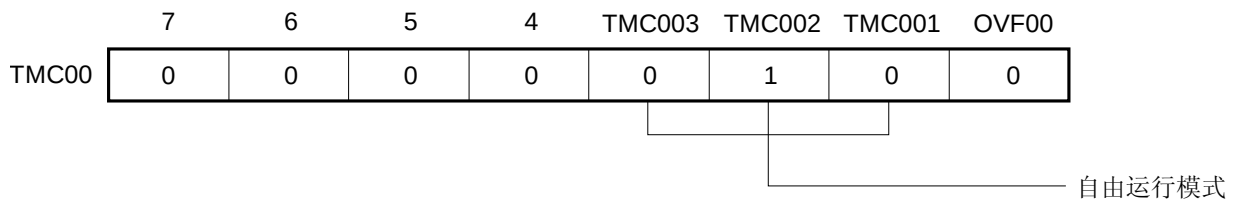
(b) 捕获/比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)

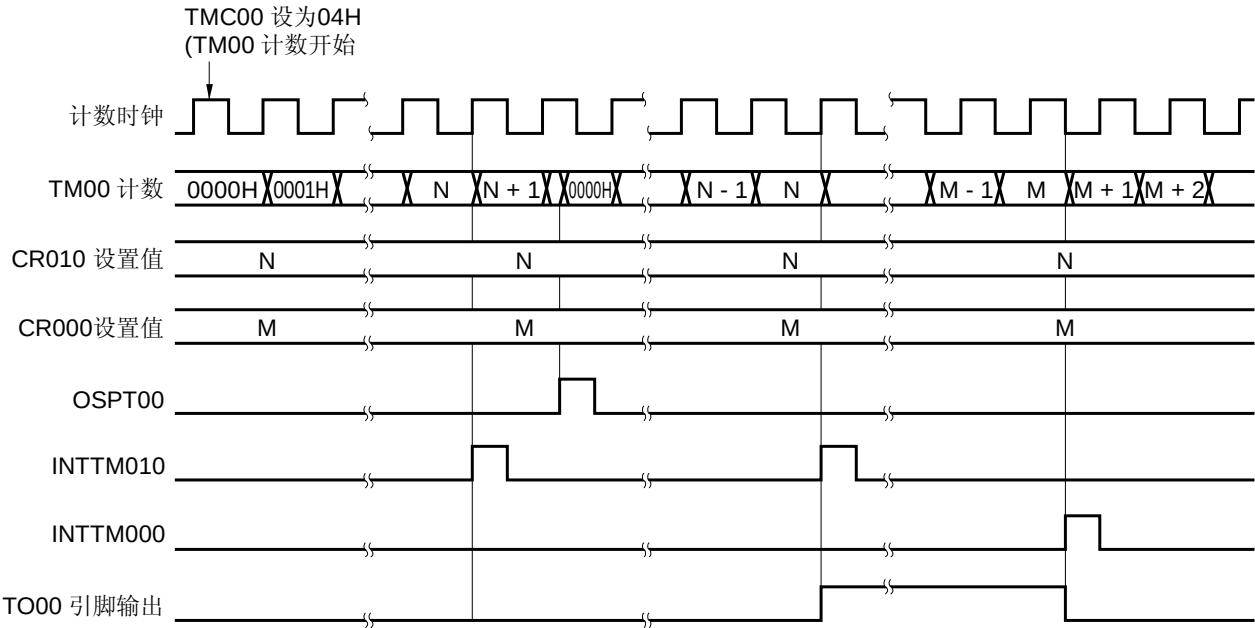


(d) 16 位定时器模式控制寄存器 (TMC00)



注意事项 CR000 和 CR010 寄存器不能设置为 0000H。

图 6-33. 软件触发的单脉冲输出操作的时序



注意事项 只要为 TMC003 和 TMC002 位设置 00 之外的值（操作停止模式），16 位定时器计数器 00 开始操作。

备注 $N < M$

(2) 外部触发的单脉冲输出

通过软件设置 16 位定时器模式控制寄存器 00（TMC00），捕获/比较控制寄存器 00（CRC00），16 位定时器输出控制寄存器 00（TOC00），使用 TI000 引脚的有效沿作为外部触发，可以从 TO00 引脚输出单脉冲信号，如图 6-34 所示。

通过预分频模式寄存器 00（PRM00）寄存器的第 4, 5 位（ES000 和 ES010）指定 TI000 引脚的有效沿。有效沿可以指定为上升沿，下降沿，或上升沿和下降沿均可用。

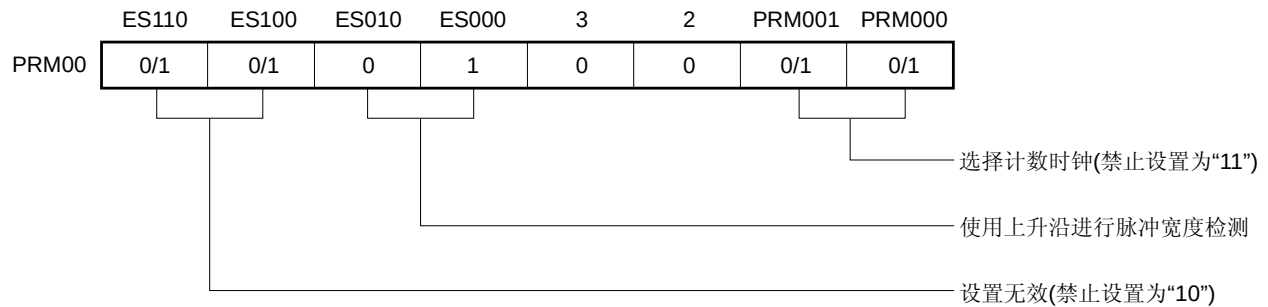
当检测到 TI000 的有效沿时，将 16 位定时器/事件计数器清零并开始计数。计数值与预先设置到 16 位定时器捕获/比较寄存器 010（CR010）的值相等时，输出变为有效。此后，计数值与预先设置到 16 位定时器捕获/比较寄存器 000（CR000）^注的值相等后，输出变为无效。

注 这里所描述的是 $N < M$ 的情况。当 $N > M$ 时，计数值与 CR000 相等，输出电平有效，而与 CR010 相等，输出电平无效。不要设置 N 等于 M。

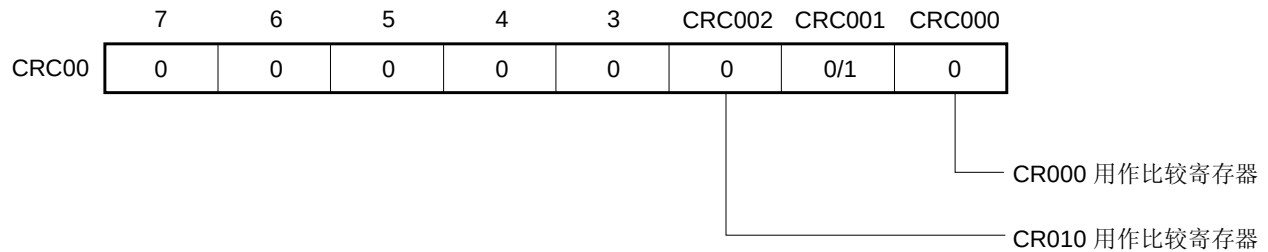
注意事项 在单脉冲输出进行时，不要再次输入外部触发信号。再次输出单脉冲，要等待正在进行的单脉冲输出完成之后。

图 6-34. 外部触发单脉冲输出的控制寄存器设置
(指定上升沿有效)

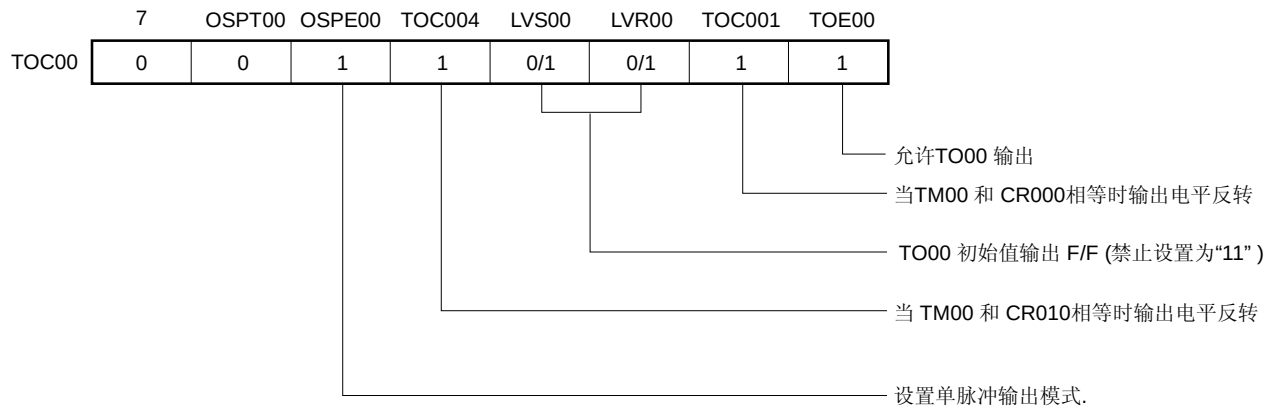
(a) 预分频模式寄存器 00 (PRM00)



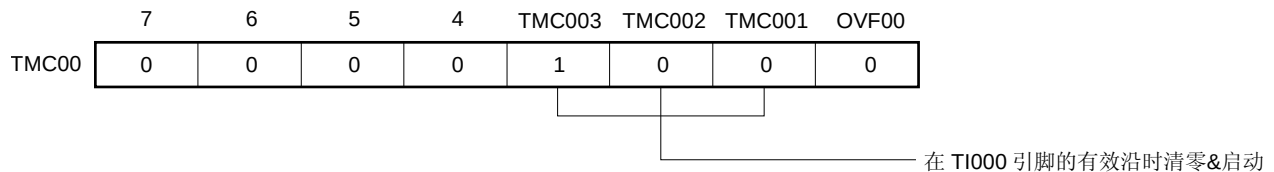
(b) 捕获/比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)

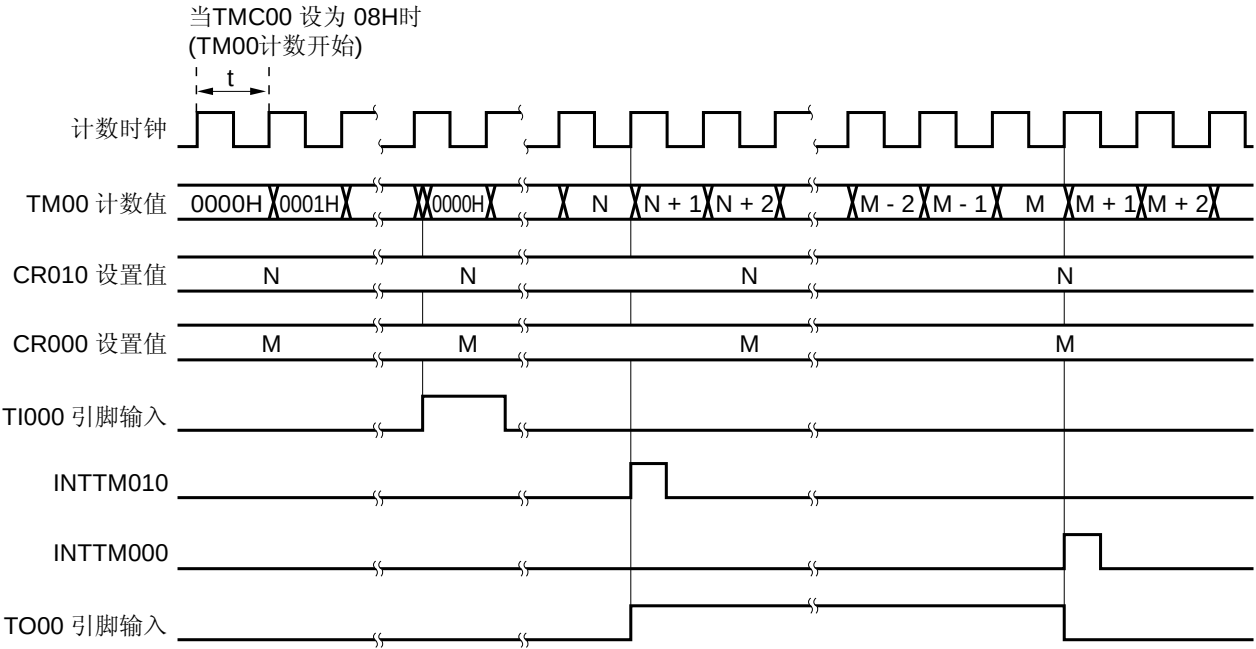


(d) 16 位定时器模式控制寄存器 (TMC00)



注意事项 CR000 和 CR010 寄存器不能设置为 0000H。

图 6-35. 外部触发单脉冲输出操作的时序（指定上升沿有效）



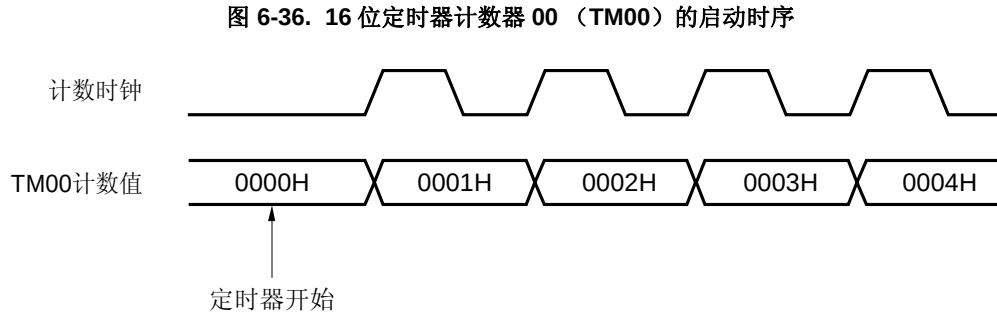
注意事项 只要为 TMC003 和 TMC002 位设置除 0 之外的值（操作停止模式），16 位定时器计数器 00 开始操作。

备注 N < M

6.5 有关 16 位定时器/事件计数器 00 的注意事项

(1) 定时器启动误差

在定时器启动后，产生信号匹配时，可能发生达一个时钟的误差。这是因为 16 位定时器计数器 00 (TM00) 的启动与计数时钟不同步。



(2) 16 位定时器计数器 00 (TM00) 操作

- <1> 当 TMC002 和 TMC003 (操作停止模式) 的值不为 0 时，16 位定时器 TM00 开始工作，置为 0，则不同。相反，当 TMC002 和 TMC003 位的值分别设置为 0 时，用来停止 16 位定时器 TM00。
- <2> 即使 TM00 的值可读，这个值也不会被 16 位定时器捕获/比较寄存器 010 (CR010) 捕获。
- <3> 当 TM00 中的值被读取时，不会发生计数丢失，由于计数时钟暂时停止工作，在读取后重新恢复。
- <4> 如果定时器停止，即使有信号输入 TI000/TI010 引脚，时钟计数和定时器中断都不会发生。

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

- <1> 在 TM00 和 CR000 相等进入清零&启动模式中，设置 16 位捕获/比较寄存器 000 (CR000) 为 0000H 之外的值。也就是说当该寄存器作为外部事件计数器时，不能执行 1 个脉冲的计数操作。
- <2> 当 TM00 和 CR000 相等进入清零&启动模式时，CR000 寄存器不能用作捕获寄存器。
- <3> 在自由运行模式和清零&启动工作模式下，使用 TI000 引脚的有效沿时，如果 CR0n0 寄存器的值设置为 0000H，当 CR0n0 在溢出 (FFFFH) 后从 0000H 变化到 0001H 时，产生 INTTM0n0 中断。
- <4> 如果 CR0n0 的新值比 TM00 的值小时，TM00 继续计数，溢出，然后从 0 开始计数。因此，如果 CR0n0 的新值比原值小，在 CR0n0 的值改变以后，定时器必须复位并重新启动。

(4) 捕获寄存器的数据保持

当 16 位定时/计数器 TM00 停止工作后，16 位定时器捕获/比较寄存器 CR0n0 的值不确定。

备注 n = 0, 1

(5) 16 位定时器模式控制寄存器 00 (TMC00) 的设置

向 16 位定时器模式控制寄存器 TMC00 的各位写入数据前，定时器必须停止工作，OVF00 标志位除外。

(6) 捕获/比较控制寄存器 00 (CRC00) 的设置

设置 CRC00 之前，定时器操作必须停止。

(7) 16 位定时器输出控制寄存器 00 (TOC00) 的设置

<1> 除 OSPT00 除外，设置定时器之前，必须停止工作。

<2> 当读取 LVS00 和 LVR00 时，读出的值为 0。

<3> 数据设置之后，OSPT00 自动清零，读取该位时，读出的值为 0。

<4> 除非在单脉冲输出模式下，OSPT00 不可设置为 1。

<5> 当连续设置 OSPT00 为 1 时，需要用预分频模式寄存器 00 (PRM00) 选择两个或更多计数时钟周期的写入间隔。

(8) 预分频模式寄存器 00 (PRM00) 的设置

定时器停止工作后，才可以向 PRM00 写入数据。

(9) 有效沿设置

定时器停止工作后，通过预分频模式寄存器 00 (PRM00) 的第 4, 5 位 (ES000 和 ES010) 可以设置 TI000 引脚的输入有效沿。

(10) 单脉冲输出

只有在自由运行模式或 TI000 引脚有效沿进入清零&启动模式下，单脉冲输出可以正常操作。因为在 TM00 和 CR000 相等进入清零&启动模式下，不会发生溢出，所以不可能输出单脉冲。

(11) 软件设置的单脉冲输出

<1> 在单脉冲正在输出情况下，不要再次置 OSPT00 位为 1。再次输出单脉冲，要等待正在进行的单脉冲输出完成之后。

<2> 当用软件触发 16 位定时器/事件计数器 00 的单脉冲输出时，不要改变 TI000 引脚或其复用端口的引脚电平。因为在这种情况下外部触发也有效，TI000 或其复用端口引脚的电平也可以对定时器清零并启动。这样会在非预期的时序输出脉冲。

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

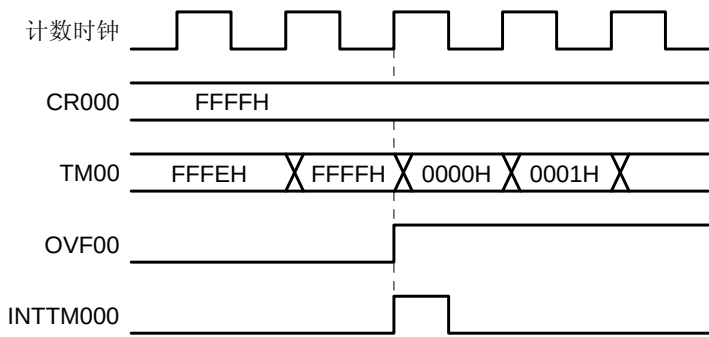
(12) 外部触发的单脉冲输出

- <1> 在单脉冲输出进行时，不要再次输入外部触发信号。再次输出单脉冲，要等待正在进行的单脉冲输出完成之后。
- <2> 16 位定时器捕获/比较寄存器 000 和 010（CR000 和 CR010）不能设置为 0000H。

(13) OVF00 标志位的操作

- <1> OVF00 标志位在下列情况下设置为 1。
在选择在 TM00 和 CR000 相等进入清零&启动模式，在 T1000 的有效沿时进入清零&启动模式，或选择自由运行模式时。
↓
CR000 置为 FFFFH
↓
当 TM00 从 FFFFH 向上计数到 0000H 时。

图 6-37. OVF00 标志位的操作时序

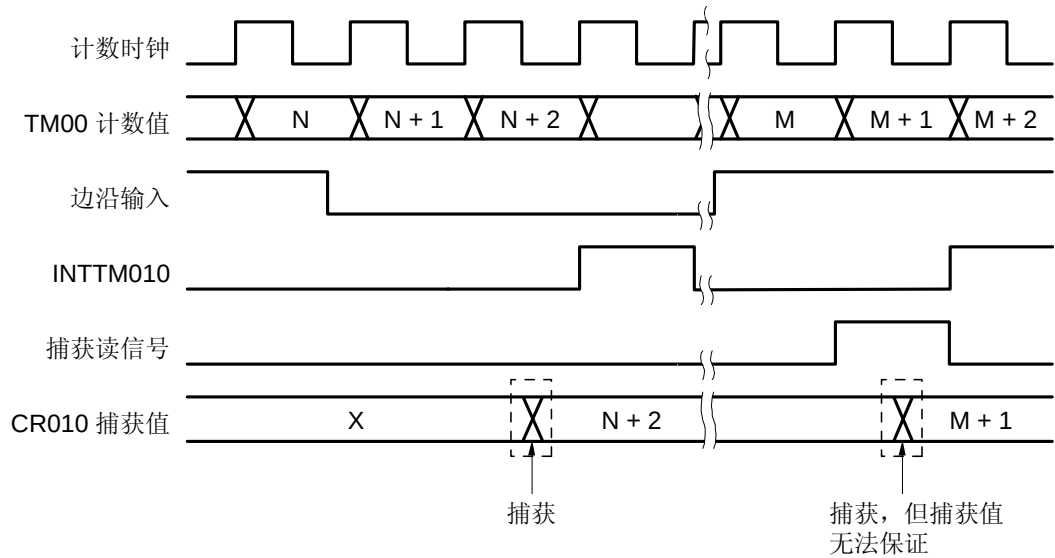


- <2> 即便在 TM00 溢出之后，而在下一个计数时钟计数之前（TM00 变为 0001H 之前），清除 OVF00 标志，OVF00 标志还会被重新置 1，清除操作无效。

(14) 冲突的操作

CR000/CR010 被用作捕获寄存器时，如果寄存器读和捕获触发输入冲突，那么捕获触发输入有优先权，读取的数据不确定。同样，当定时器计数停止和捕获触发输入发生冲突时，捕获到的数据不确定。

图 6-38. 捕获寄存器数据保留时序



(15) 捕获操作

- <1> 如果指定 TI000 引脚有效沿作为计数时钟，就不要设置 TI000 引脚的有效沿时进入清零/开始模式和捕获触发器。
- <2> 如果上升沿和下降沿均被指定为 TI000 引脚的有效触发沿，则捕获无法进行。
- <3> 当 CRC001 值为 1 时，如果 TI010 引脚检测到有效沿，TM00 的计数值不能被捕获到 CR000 寄存器中，但是 TI010 引脚的输入可以用作外部中断源，因为外部有效信号输入时产生 INTTM000 中断。
- <4> 为保证捕获操作的可靠性，捕获触发信号需要的脉冲宽度要大于预分频模式寄存器 00 (PRM00) 选择的两个计数时钟周期。
- <5> 在计数时钟的下降沿时执行捕获操作。然而，在下个计数时钟的上升沿，会产生一个中断请求输入 (INTTM0n0)。
- <6> 要使用两个捕获寄存器时，设置 TI000 和 TI010 引脚。

(16) 比较操作

当 CR0n0 在比较模式下，即使有捕获触发信号输入，捕获操作也可能无法执行。

备注 n = 0, 1

(17) 在定时器工作时改变比较寄存器的值

<1> 16 位定时器捕获/比较寄存器 0n0 (CR0n0) 作为比较寄存器使用，在定时器计数期间，在 16 位定时器计数器 00 (TM00) 和 16 位定时器捕获/比较寄存器 0n0 (CR0n0) 的值相等的时序附近修改 CR0n0 的值时，改变 CR0n0 的时序和两者相等匹配的时序可能会冲突，这种情况下，操作无法得到保证。在定时器计数时改变 CR0n0 的值，INTTM000 中断服务程序会执行如下操作。

<更改周期 (CR000)>

1. TM00 和 CR000 相等时，禁止定时器输出反转操作 (TOC001 = 0)。
2. 禁止 INTTM000 中断 (TMMK000 = 1)。
3. 重写 CR000。
4. 等待一个 TM00 计数时钟周期。
5. TM00 和 CR000 相等时，允许定时器输出反转操作 (TOC001=1)。
6. INTTM000 的中断请求标志位 (TMIF000 = 0) 清零。
7. 允许 INTTM000 中断 (TMMK000 = 0)。

<更改占空比 (CR010)>

1. TM00 和 CR010 相等时，禁止定时器输出反转操作 (TOC004 = 0)。
2. 禁止 INTTM000 中断 (TMMK000 = 1)。
3. 重写 CR010。
4. 等待一个 TM00 计数时钟周期。
5. TM00 和 CR010 相等时，允许定时器输出反转操作 (TOC004=1)。
6. INTTM000 的中断请求标志位 (TMIF000 = 0) 清零。
7. 允许 INTTM000 中断 (TMMK000 = 0)。

当中断和定时器输出反转被禁止时 (上面 1-4 步)，定时器计数仍然继续。如果设置的 CR0n0 的值比较小，TM00 的值可能超过 CR0n0。因此设置 CR0n0 的值时，要考虑当产生 INTTM000 中断之后，定时器时钟和 CPU 时钟的响应时间。

备注 n = 0, 1

<2> 如果在定时器计数期间改变 CR010 时，但未执行上面的过程<1>，那么 CR010 中的值将被重新写入两次或更多次，每次写入都将导致 TO00 引脚输出电平的反转。

(18) 脉冲沿检测

<1> 在如下情况时，要注意 TI0n0 引脚有效沿的检测。

(a) 系统复位后，如果 TI0n0 引脚立即输入高电平，那么允许 16 位定时器计数器 00 (TM00) 操作。

→ 如果上升沿或双沿被指定为 TI0n0 引脚的有效沿，TM00 操作允许后，马上会检测到一个上升沿。

(b) 如果当 TI0n0 引脚是高电平的时候，停止 TM00 操作，则 TI0n0 引脚输入低电平后，允许 TM00 操作。

→ 如果下降沿或双沿被指定为 TI0n0 引脚的有效沿，TM00 操作允许后，立即会检测到一个下降沿。

(c) 如果在 TI0n0 引脚是低电平时停止 TM00 操作，TI0n0 引脚的输入高电平后，允许 TM00 操作。

→ 如果上升沿或上升沿和下降沿均被指定为 TI0n0 引脚的有效沿，允许 TM00 操作后，立即会检测到一个上升沿。

备注 $n = 0, 1$

<2> 当 TI000 有效沿被用作计数时钟，或被用作捕获触发信号时，用来消除噪音的采样时钟是不同的。在前一种情况中，计数时钟为 f_{XP} 。而在后一种情况中，由预分频模式寄存器 00 (PRM00) 选择计数时钟。直到采样到有效沿，并且检测到两次有效电平之后才能完成捕获操作，因此，可以消除具有窄脉冲宽度的噪音。

(19) 外部事件计数器

<1> 检测到两次有效沿后，定时器开始计数。

<2> 需要知道外部事件计数器的计数值，则应该去读 TM00。

(20) PPG 输出

<1> CR000 和 CR010 的取值范围如下：

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

<2> 通过 PPG 输出产生的脉冲的周期 (CR000 的设置值 + 1) 的占空比是：

$$(CR010 \text{ 的设置值} + 1) / (CR000 \text{ 的设置值} + 1)。$$

(21) STOP 模式或系统时钟停止模式的设置

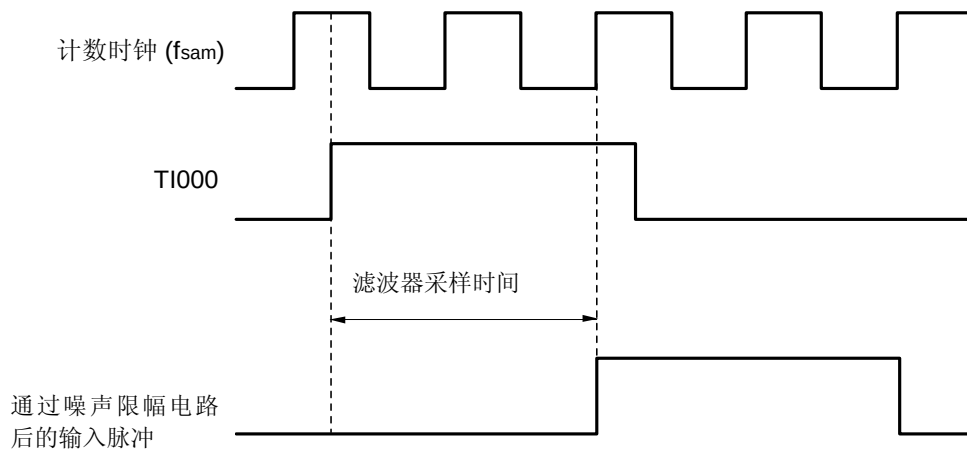
除 TI000 引脚有效沿作为计数时钟以外，在设置 STOP 模式或系统时钟停止模式之前，停止定时器工作；否则在系统时钟启动时，定时器将发生故障。

(22) P31/TI010/TO00 引脚

当 P31 用做 TI010 的有效沿输入引脚时，不能用作定时器输出引脚 (TO00)。同样，当 P31 被用于定时器输出引脚 TO00 时，不能用于 TI010 的有效沿输入引脚。

(23) 外部时钟限制

- <1> 当使用 TI000 引脚的输入脉冲作为计数时钟（外部触发）时，确保输入脉冲宽度能够满足 AC 特性。具体的 AC 特性，参见第二十二章 和 第二十三章 电气特性。
- <2> 当一个外部波形输入到 16 位定时器/事件计数器 00 时，由噪声限幅电路采样，因此在器件内部变为有效的时序会产生错误。



备注 计数时钟 (f_{sam}) 可以通过预分频模式寄存器 00 (PRM00) 的第 0 位和第 1 位 (PRM000, PRM001) 来选择。

第七章 8 位定时器 80

7.1 8 位定时器 80 的功能

8 位定时器 80 具有 8 位间隔定时器的功能，可以按预先指定的时间间隔产生中断。

表 7-1 8 位定时器 80 的间隔时间

	最小间隔时间	最大间隔时间	分辨率
f _{XP} = 8.0 MHz	2 ⁶ /f _{XP} (8 μs)	2 ¹⁴ /f _{XP} (2.05 ms)	2 ⁶ /f _{XP} (8 μs)
	2 ⁸ /f _{XP} (32 μs)	2 ¹⁶ /f _{XP} (8.19 ms)	2 ⁸ /f _{XP} (32 μs)
	2 ¹⁰ /f _{XP} (128 μs)	2 ¹⁸ /f _{XP} (32.7 ms)	2 ¹⁰ /f _{XP} (128 μs)
	2 ¹⁶ /f _{XP} (8.19 ms)	2 ²⁴ /f _{XP} (2.01 s)	2 ¹⁶ /f _{XP} (8.19 ms)
f _{XP} = 10.0 MHz	2 ⁶ /f _{XP} (6.4 μs)	2 ¹⁴ /f _{XP} (1.64 ms)	2 ⁶ /f _{XP} (6.4 μs)
	2 ⁸ /f _{XP} (25.6 μs)	2 ¹⁶ /f _{XP} (6.55 ms)	2 ⁸ /f _{XP} (25.6 μs)
	2 ¹⁰ /f _{XP} (102 μs)	2 ¹⁸ /f _{XP} (26.2 ms)	2 ¹⁰ /f _{XP} (102 μs)
	2 ¹⁶ /f _{XP} (6.55 ms)	2 ²⁴ /f _{XP} (1.68 s)	2 ¹⁶ /f _{XP} (6.55 ms)

备注 f_{XP}: 外围硬件时钟振荡频率

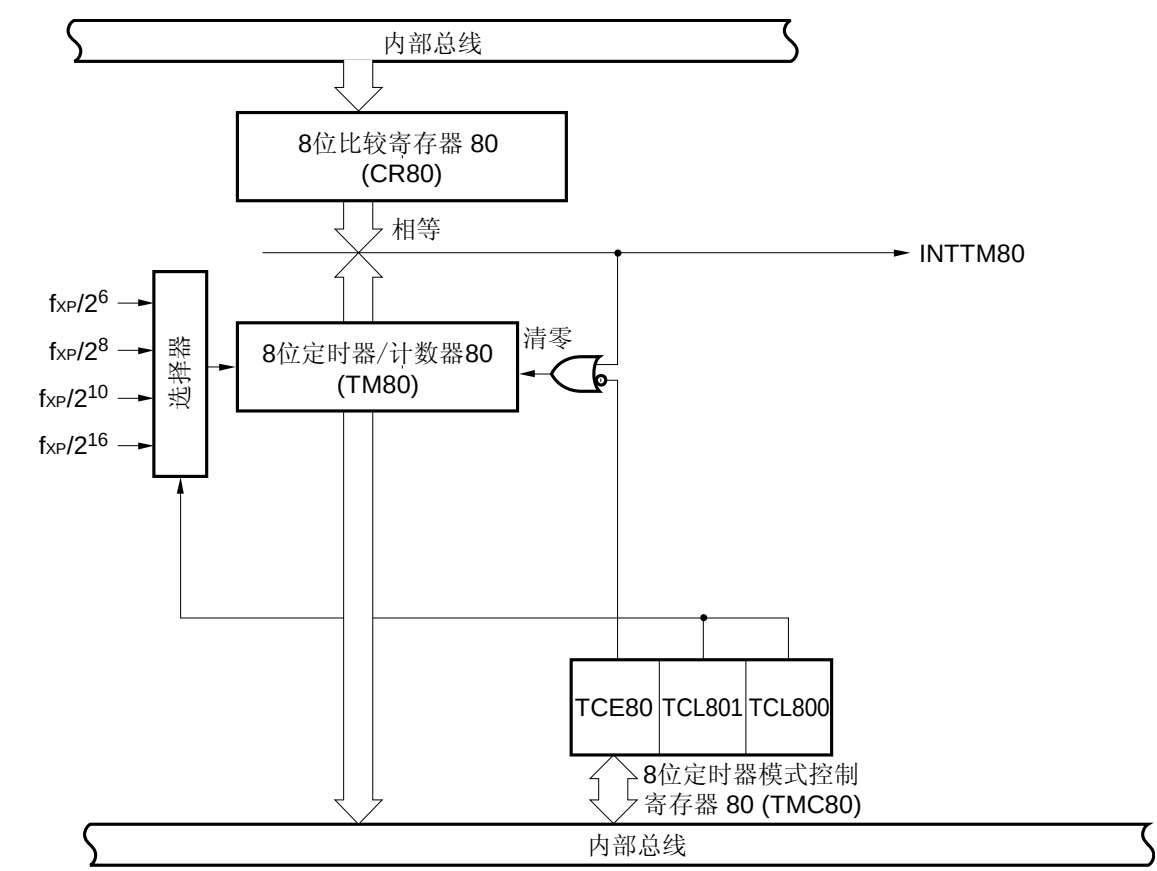
7.2 8 位定时器 80 的配置

8 位定时器 80 由下列硬件组成。

表 7-2 8 位定时器 80 的配置

项目	配置
定时器计数器	8 位定时器计数器 80 (TM80)
寄存器	8 位比较寄存器 80 (CR80)
控制寄存器	8 位定时器模式控制寄存器 80 (TMC80)

图 7-1 8 位定时器 80 的框图



备注 f_{XP}: 外围硬件时钟振荡频率

(1) 8 位比较寄存器 80 (CR80)

8 位比较寄存器 CR80 用来将自身的设定值与 8 位定时器/计数器 80 (TM80) 的计数值进行比较。当两者的值相等时，产生中断请求信号 (INTTM80)。

CR80 可以通过 8 位的存储操作指令来设置。该寄存器的值可以设置为 00H~FFH。

复位信号的产生，会使该寄存器的内容不确定。

图 7-2 8 位比较寄存器 80 (CR80) 格式

地址: FFCDH 复位后: 不确定 写

符号	7	6	5	4	3	2	1	0
CR80								

注意事项 当改变 CR80 值的时候，必须要停止定时器的操作。如果在定时器操作期间改变 CR80 的值，则立刻产生匹配中断请求信号，且定时器可能会被清零。

(2) 8 位定时器计数器 80 (TM80)

该 8 位寄存器用来对计数脉冲进行计数。

TM80 的值可以通过 8 位存储操作指令进行读取。

复位信号的产生，会将 TM80 的值清零。

图 7-3 8 位定时器计数器 80 (TM80) 的格式

地址: FFCEH 复位后: 00H 读

符号	7	6	5	4	3	2	1	0
TM80								

7.3 控制 8 位定时器 80 的寄存器

8 位定时器 80 可以通过 8 位定时器模式控制寄存器 80（TMC80）进行控制。

(1) 8 位定时器模式控制寄存器 80（TMC80）

TMC80 用来允许或停止 8 位定时器/计数器 80（TM80）的操作，并可以设置 TM80 的计数时钟。
该寄存器可用 1 位或 8 位存储操作指令进行设置。
复位信号的产生，会将 TM80 清零。

图 7-4 8 位定时器模式控制寄存器 80（TMC80）的格式

地址： FFCCH 复位后： 00H 读/写

符号 <7> 6 5 4 3 2 1 0

TMC80

TCE80	0	0	0	0	TCL801	TCL800	0
-------	---	---	---	---	--------	--------	---

TCE80	TM80 的控制操作						
0	停止操作（ TM80 被清零）.						
1	允许操作.						

TCL801	TCL800	8 位定时器 80 的计数时钟选择		
			$f_{XP} = 8.0 \text{ MHz}$	$f_{XP} = 10.0 \text{ MHz}$
0	0	$f_{XP}/2^6$	125 kHz	156.3 kHz
0	1	$f_{XP}/2^8$	31.25 kHz	39.06 kHz
1	0	$f_{XP}/2^{10}$	7.81 kHz	9.77 kHz
1	1	$f_{XP}/2^{16}$	0.12 kHz	0.15 kHz

- 注意事项
- 1. 必须在定时器操作停止之后，才能对 TMC80 设置。
 - 2. 确保第 0 位和第 6 位清零。

备注 f_{XP} : 外围硬件时钟振荡频率

7.4 8 位定时器 80 的操作

7.4.1 作为间隔定时器的操作

当 8 位定时器 80 作为间隔定时器时，可以按照 8 位比较寄存器（CR80）中预先设置的计数值作为间隔，重复产生中断。

8 位定时器 80 用作间隔定时器的步骤如下：

- <1> 禁止操作 8 位定时器计数器 80（将 TCE80（8 位定时器模式控制寄存器 80（TMC80）的第 7 位）清 0）。
- <2> 设置 8 位定时器 80 的计数时钟（参考表 7-3 和 7-4）。
- <3> 设置 CR80 的计数值。
- <4> 允许 TM80 操作（设置 TCE80 为 1）。

当 8 位定时器计数器 80（TM80）的计数值等于 CR80 的设置值时，TM80 的值被清零，然后继续计数。同时，产生中断请求信号（INTTM80）。

表 7-3 和表 7-4 列出了间隔时间，图 7-5 表示间隔定时器操作的时序。

- 注意事项**
1. 当改变 CR80 值的时候，必须要停止定时器的操作。如果在定时器工作期间改变 CR80 的值，则可能会立即产生相等中断请求信号。
 2. 当使用 8 位存储操作指令设置 TMC80 的计数时钟，并同时允许 TM80 操作，则在定时器启动后一个周期的误差可能是一个或多个时钟（参见 7.5（1）定时器启动时的错误）。因此，要将 TM80 作为间隔定时器使用，必须按上述顺序进行设置。

表 7-3 8 位定时器 80 的间隔时间（ $f_{XP} = 8.0 \text{ MHz}$ ）

TCL801	TCL800	最小间隔时间	最大间隔时间	分辨率
0	0	$2^6/f_{XP}$ (8 μs)	$2^{14}/f_{XP}$ (2.05 ms)	$2^6/f_{XP}$ (8 μs)
0	1	$2^8/f_{XP}$ (32 μs)	$2^{16}/f_{XP}$ (8.19 ms)	$2^8/f_{XP}$ (32 μs)
1	0	$2^{10}/f_{XP}$ (128 μs)	$2^{18}/f_{XP}$ (32.7 ms)	$2^{10}/f_{XP}$ (128 μs)
1	1	$2^{16}/f_{XP}$ (8.19 ms)	$2^{24}/f_{XP}$ (2.01 s)	$2^{16}/f_{XP}$ (8.19 ms)

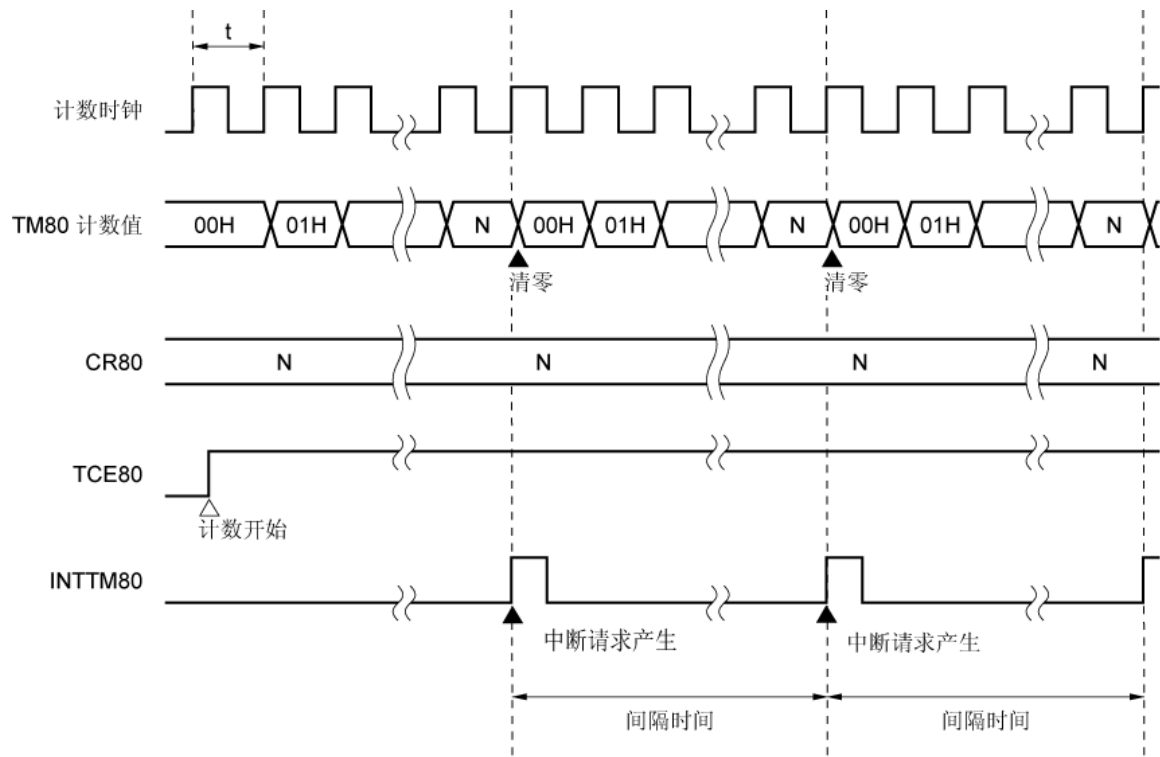
备注 f_{XP} : 外围硬件时钟振荡频率

表 7-4 8 位定时器 80 的间隔时间（ $f_{XP} = 10.0 \text{ MHz}$ ）

TCL801	TCL800	最小间隔时间	最大间隔时间	分辨率
0	0	$26/f_{XP}$ (6.4 μs)	$2^{14}/f_{XP}$ (1.64 ms)	$2^6/f_{XP}$ (6.4 μs)
0	1	$2^8/f_{XP}$ (25.6 μs)	$2^{16}/f_{XP}$ (6.55 ms)	$2^8/f_{XP}$ (25.6 μs)
1	0	$2^{10}/f_{XP}$ (102 μs)	$2^{18}/f_{XP}$ (26.2 ms)	$2^{10}/f_{XP}$ (102 μs)
1	1	$2^{16}/f_{XP}$ (6.55 ms)	$2^{24}/f_{XP}$ (1.68 s)	$2^{16}/f_{XP}$ (6.55 ms)

备注 f_{XP} : 外围硬件时钟振荡频率

图 7-5 间隔定时器操作的时序



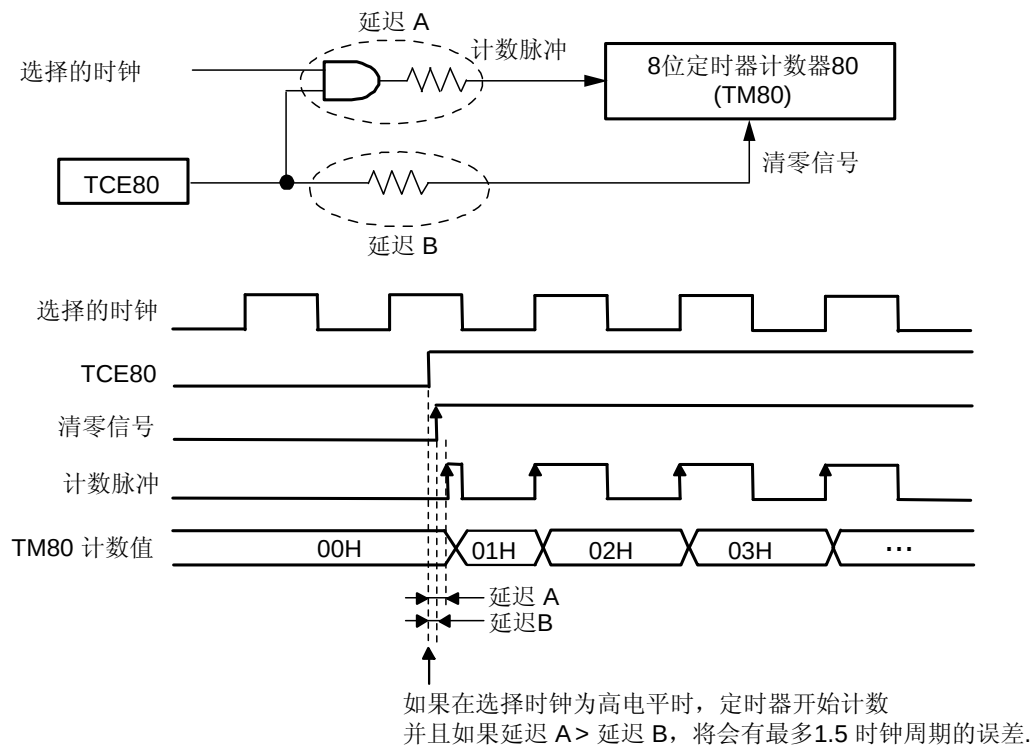
备注 间隔时间 = (N + 1) × t
N = 00H ~ FFH

7.5 8 位定时器 80 的注意事项

(1) 定时器启动时的误差

从启动定时器到产生匹配信号的时间，包含了一个最大 1.5 个时钟周期的误差。这是因为如果在计数时钟为高电平时启动定时器，那么可能立即检测到上升沿，且计数器累加（参见图 7-6）。

图 7-6 1.5 个（最大）时钟周期误差的情形



(2) 8 位比较寄存器 80 的设置

8 位比较寄存器 80 (CR0) 可设置为 00H。

(3) 设置 STOP 模式时注意事项

在执行 STOP 指令之前，必须先停止定时器操作 (TCE80=0)。

第八章 8 位定时器H1

8.1 8 位定时器H1 的功能

8 位定时器 H1 的功能如下

- 间隔定时器·
- PWM 输出模式
- 方波输出·

8.2 8 位定时器H1 的配置

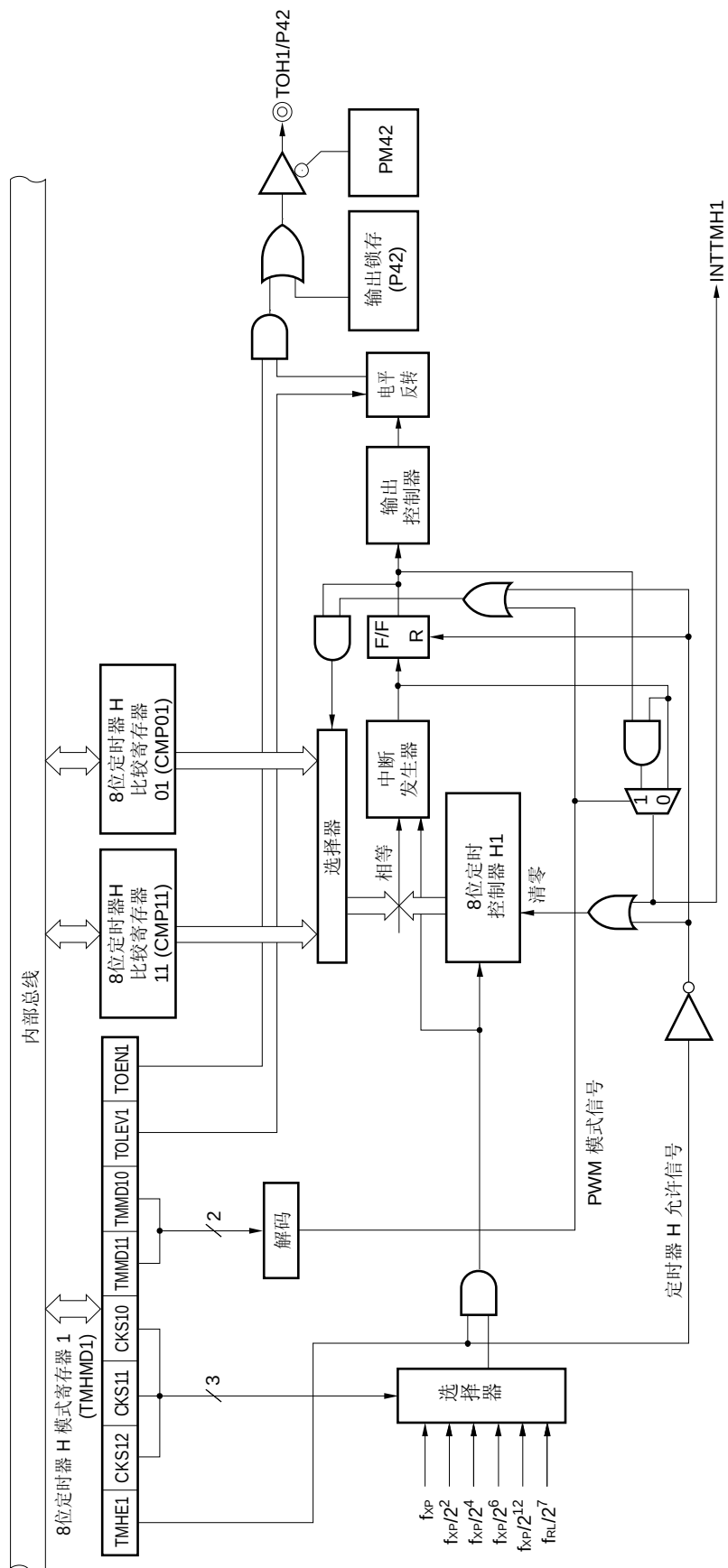
8 位定时器 H1 由以下硬件组成

表 8-1 8 位定时器 H1 的配置

项目	配置
定时器寄存器	8 位定时器计数器 H1
寄存器	8 位定时器 H 比较寄存器 01（CMP01） 8 位定时器 H 比较寄存器 11（CMP11）
定时器输出	TOH1
控制寄存器	8 位定时器 H 模式寄存器 1（TMHMD1） 端口模式寄存器 4（PM4） 端口寄存器 4（P4）

图 8-1 显示结构框图。

图 8-1 8 位定时器 H1 框图



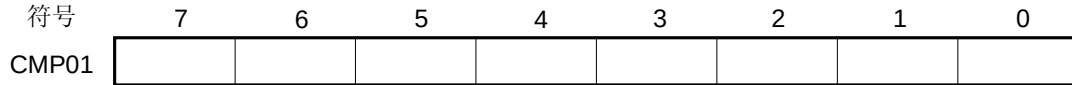
(1) 8 位定时器 H 比较寄存器 01 (CMP01)

该寄存器可以通过 8 位存储操作指令进行读或写。

复位信号的产生，会将该寄存器清为 00H。

图 8-2 8 位定时器 H 比较寄存器 01 (CMP01) 的格式

地址: FF0EH 复位后: 00H 读/写



注意事项 CMP01 在定时器计数期间不能被重写。

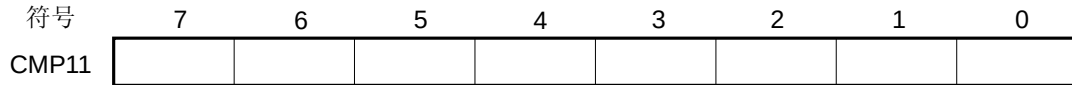
(2) 8 位定时器 H 比较寄存器 11 (CMP11)

该寄存器可以通过 8 位存储操作指令进行读或写。

复位信号的产生，会将该寄存器清为 00H。

图 8-3 8 位定时器 H 比较寄存器 11 (CMP11) 的格式

地址: FF0FH 复位后: 00H 读/写



CMP11 在定时器计数期间可以被重写。

如果在定时器操作期间重写 CMP11 值，在计数值和修改以前的比较值匹配的时刻，修改后的比较值才会生效。如果计数值和修改以前的比较值匹配发生的时刻，和 CPU 向 CMP11 写入的时序发生冲突，则在下一次计数值和修改以前的比较值匹配的时刻，修改后的比较值才会生效。

注意事项 在 PWM 输出模式中，当定时器停止计数后 (TMHE1=0)，再次启动计数操作时 (TMHE1=1)，必须先对 CMP11 进行赋值 (即使 CMP11 前后设置的值相同，也必须重新设置)。

8.3 控制 8 位定时器H1 的寄存器

以下三个寄存器用来控制 8 位定时器 H1。

- 8 位定时器 H 模式寄存器 1 (TMHMD1)
- 端口模式寄存器 4 (PM4)
- 端口寄存器 4 (P4)

(1) 8 位定时器 H 模式寄存器 (TMHMD1)

该寄存器控制 8 位定时器 H 的模式。

该寄存器可以通过 1 位或 8 位存储操作指令进行设置。

复位信号的产生，会将该寄存器清为 00H。

图 8-4 8 位定时器 H 模式寄存器 1 (TMHMD1) 的格式

地址: FF70H 复位后: 00H 读/写

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

TMHE1	定时器操作使能
0	停止定时器计数操作 (计数器清0)
1	允许定时器计数操作(通过输入时钟启动计数操作)

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.))
其他			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
其他		禁止设置

TOLEV1	定时器输出电平控制 (在默认的模式中)
0	低电平
1	高电平

TOEN1	定时器输出控制
0	不允许输出
1	允许输出

- 注意事项**
1. 当 TMHE1 设置为 1 时, 禁止对 TMHMD1 寄存器的其它位进行设置。
 2. 在 PWM 输出模式下, 当定时器停止计数后 (TMHE1=0), 再次启动计数操作时 (TMHE1=1), 必须先对 CMP11 进行赋值 (即使 CMP11 寄存器前后设置的值相同, 也必须重新设置)。

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

(2) 端口模式寄存器 4 (PM4)

该寄存器可以逐位设置端口 4 的输入输出模式。
当使用 P42/TOH1 引脚作为定时器输出时，要对 PM42 进行清零，并将 P42 的输出锁存清零。
PM4 可以通过 1 位或 8 位存储操作指令进行设置。
复位信号的产生，会将该寄存器置为 FFH。

图 8-5 端口模式寄存器 4 (PM4) 的格式

地址： FF24H 复位后： FFH 读/写

符号

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

PM4n	P4n 引脚 I/O 模式选择 (n=0~7)
0	输出模式 (输出缓冲打开)
1	输入模式 (输出缓冲关闭)

8.4 8 位定时器H1 的操作

8.4.1 作为间隔定时器/方波输出的操作

当 8 位定时器计数器 H1 与比较寄存器 01 (CMP01) 相等时, 产生中断请求信号 INTTMH1, 8 位定时器计数器 H1 被清零。

在间隔定时器模式下, 不使用比较寄存器 11 (CMP11)。因为即使设置 CMP11 寄存器, 也不会对 8 位定时器计数器 H1 和 CMP11 寄存器的相等进行检测, 所以, 不影响定时器的输出。

设置定时器 H 模式寄存器 1 (TMHMD1) 的第 0 位 (TOEN1) 为 1, 可从 TOH1 输出任意频率的方波 (占空比为 50%)。

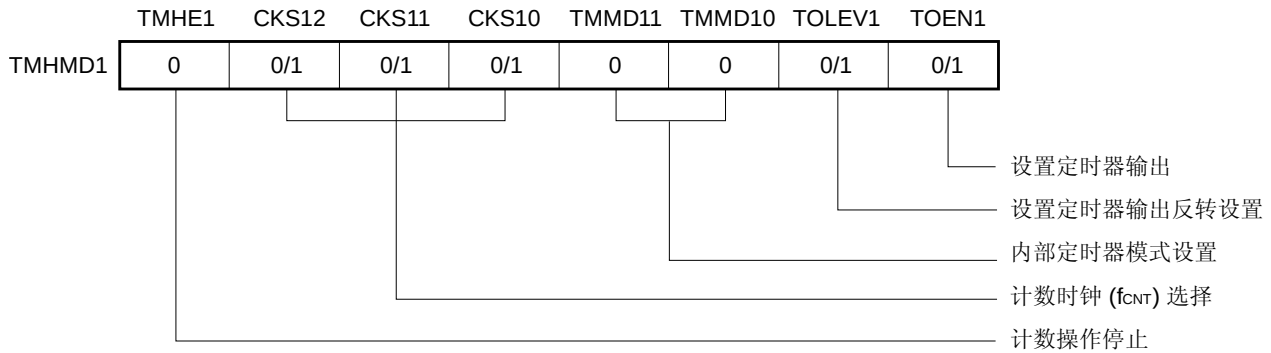
(1) 用法

以相同时间间隔重复产生 INTTMH1 信号。

<1> 设置每个寄存器

图 8-6 间隔定时器/方波输出操作时的寄存器设置

(i) 设置定时器 H 模式寄存器 1 (TMHMD1)



(ii) CMP01 寄存器设置

- 比较值 (N)

<2> 当 TMHE1 = 1 时, 启动计数操作

<3> 当 8 位定时器计数器 H1 的值等于 CMP01 寄存器的设置值时, 产生中断请求信号 INTTMH1, 8 位定时器计数器 H1 被清零。

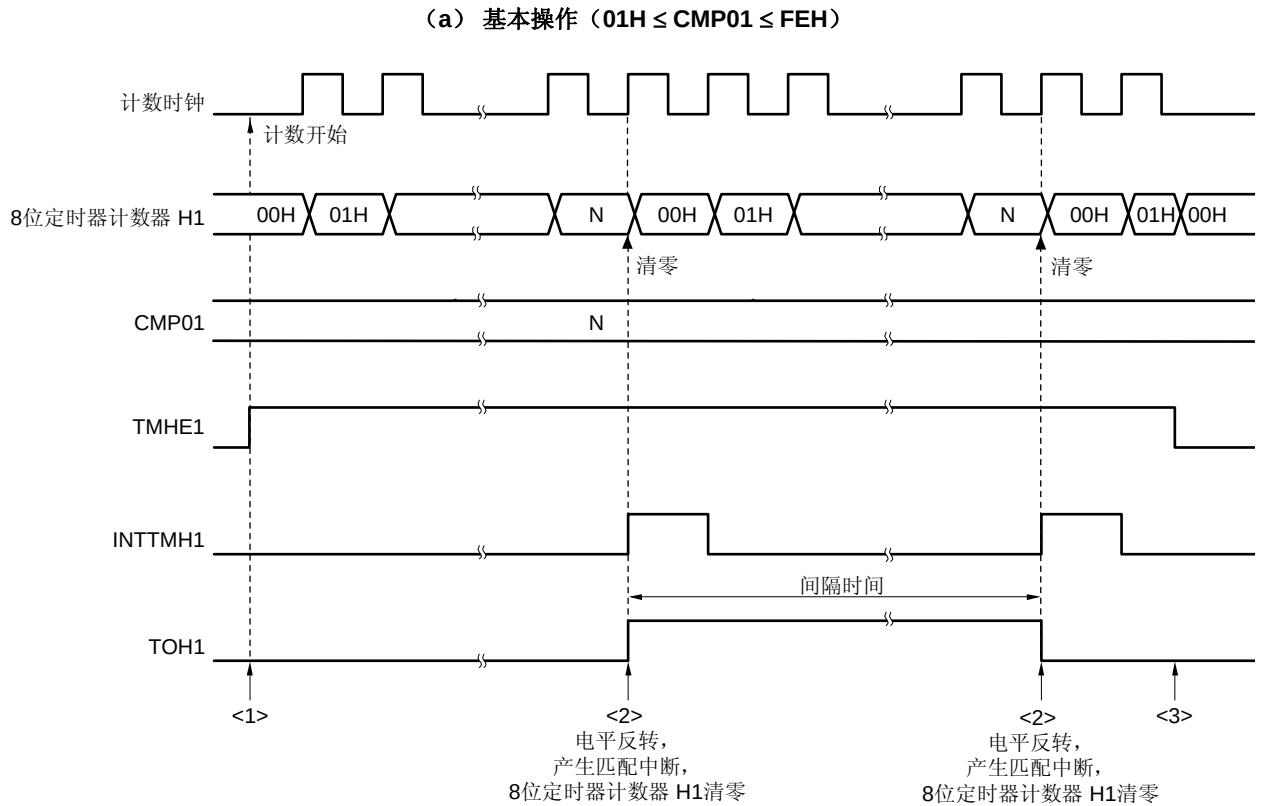
$$\text{间隔时间} = (N + 1) / f_{\text{CNT}}$$

<4> 此后, 以相同的时间间隔产生 INTTMH1 中断。要停止计数操作, 将 TMHE1 清零即可。

(2) 时序图

间隔定时器/方波输出操作的时序图如下：

图 8-7. 间隔定时器/方波输出操作的时序 (1/2)

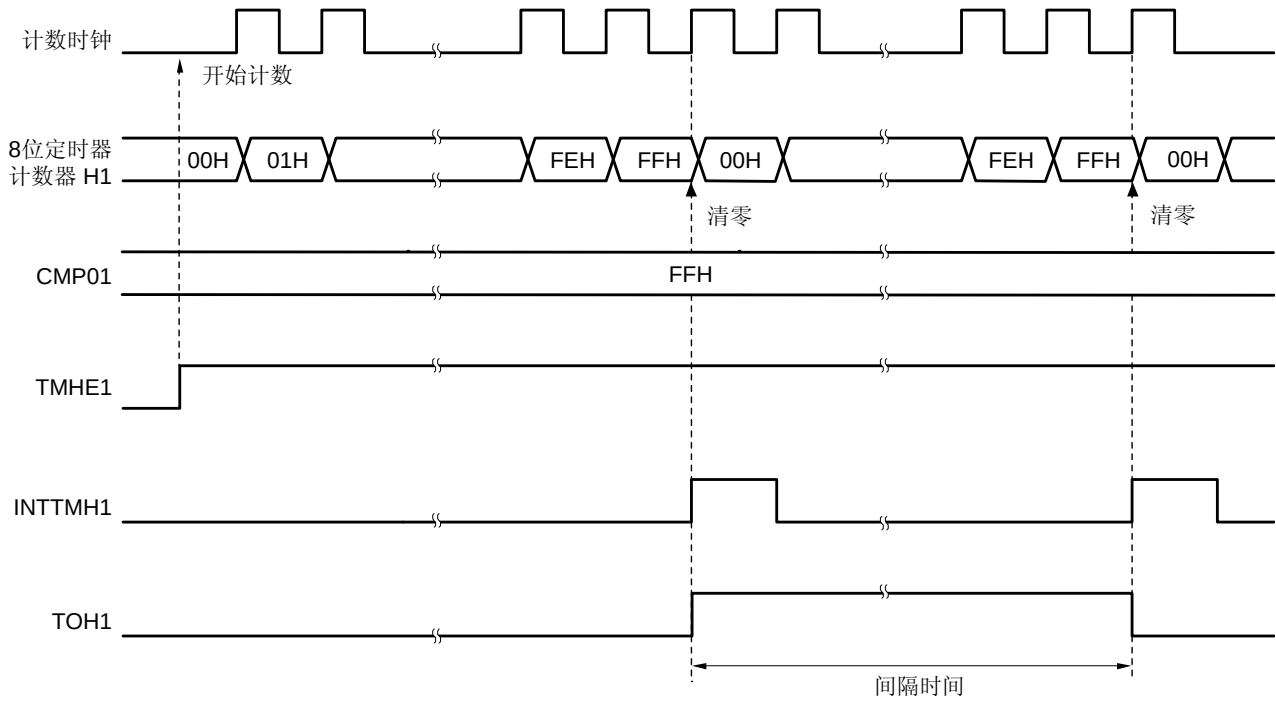


- <1> 将 TMHE1 位设置为 1，允许计数操作。在允许操作后的 1 个时钟周期的时间内，计数时钟开始计数。
- <2> 当 8 位定时器计数器 H1 与 CMP01 寄存器的值相等时，8 位定时器计数器 H1 的值清零，TOH1 输出电平反转，并输出 INTTMH1 信号。
- <3> 在定时器 H1 操作期间，将 TMHE1 位清零可以使 INTTMH1 信号和 TOH1 输出变为无效。如果这些从开始就一直无效，则电平保持不变。

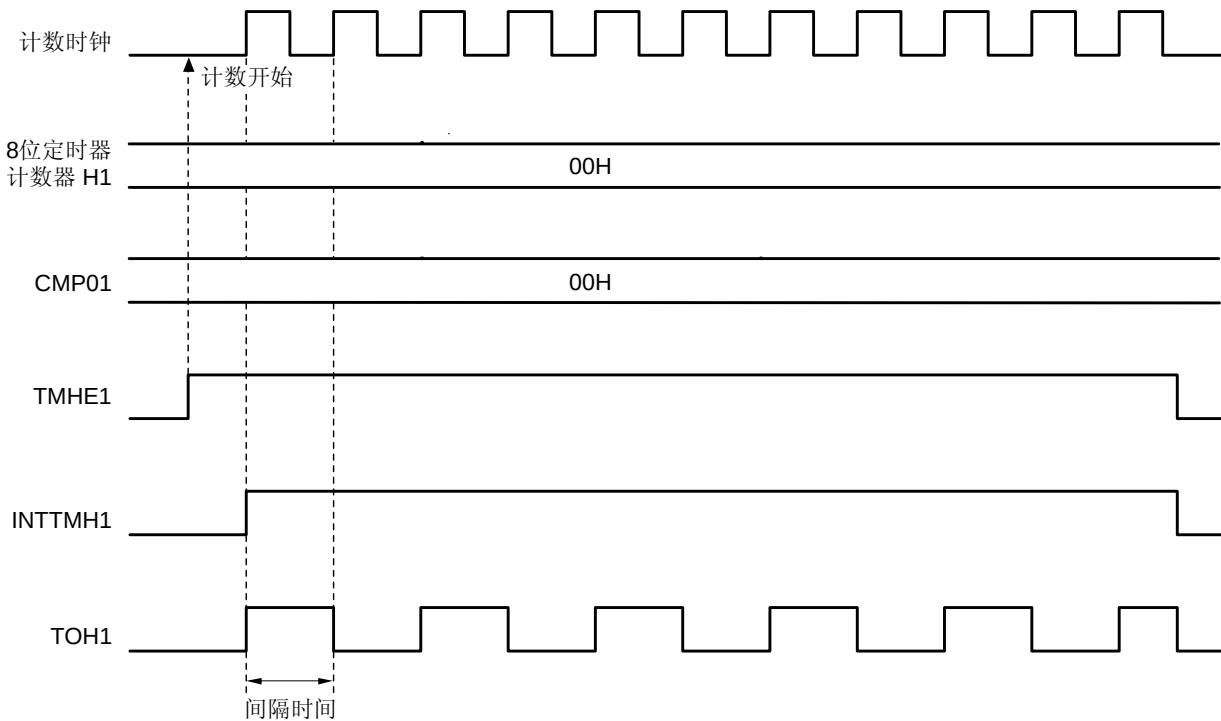
备注 $01H \leq N \leq FEH$

图 8-7. 间隔定时器/方波输出操作的时序 (2/2)

(b) 当 CMP01 = FFH 时的操作



(c) 当 CMP01 = 00H 时的操作



8.4.2 用作PWM输出模式的操作

在 PWM 输出模式中，可输出具有任意占空比和周期的脉冲。

8 位定时器比较寄存器 01（CMP01）用来控制定时器输出（TOH1）的周期。禁止在定时器操作期间重写 CMP01 寄存器。

8 位定时器比较寄存器 11（CMP11）用来控制定时器输出（TOH1）的占空比。允许在定时器操作期间重写 CMP11 寄存器。

PWM 输出模式中的操作如下。

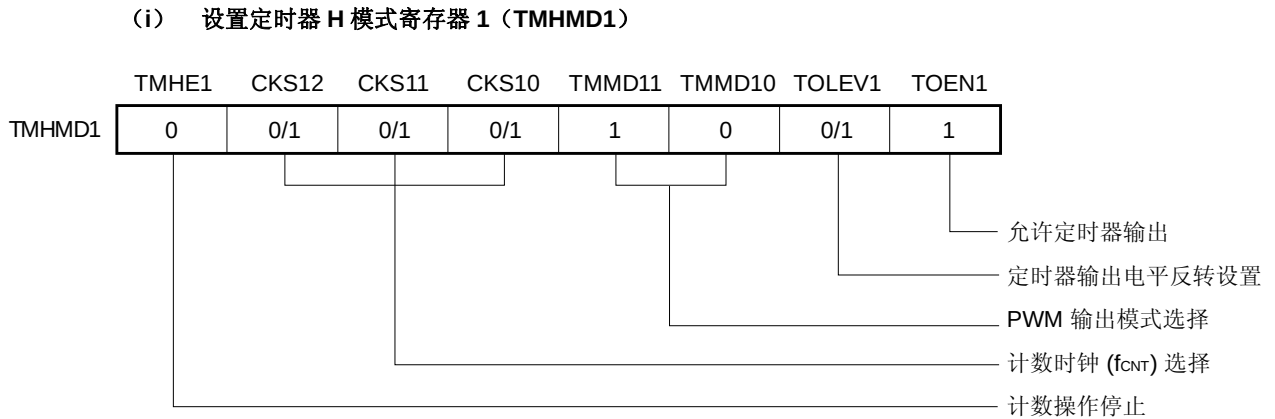
定时器开始计数后，当 8 位定时器计数器 H1 与 CMP01 寄存器的值相等时，TOH1 输出变为有效，且 8 位定时器计数器 H1 清零。当 8 位定时器计数器 H1 与 CMP11 寄存器的值相等时，TOH1 输出变为无效。

（1） 用法

在 PWM 输出模式中，可输出具有任意占空比和周期的脉冲。

<1> 设置每个寄存器

图 8-8 PWM 输出模式下的寄存器设置



（ii） 设置 CMP01 寄存器

- 比较值（N）： 周期设置

（iii） 设置 CMP11 寄存器

- 比较值（M）： 占空比设置

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

<2> 当 TMHE1 = 1 时，开始计数操作。

<3> CMP01 是在允许计数器操作后，首先用于比较的比较寄存器。当 8 位定时器计数器 H1 与 CMP01 寄存器的值相等时，8 位定时器计数器 H1 清零、产生中断请求信号（INTTMH1）并且 TOH1 输出变为有效。同时，用于和 8 位定时器计数器 H1 比较的比较寄存器已经由 CMP01 改为 CMP11。

- <4> 当 8 位定时器计数器 H1 与 CMP11 寄存器相等时，TOH1 输出变为无效，用于和 8 位定时器计数器 H1 比较的比较寄存器已经由 CMP11 改为 CMP01。此时，不对 8 位定时器计数器 H1 清零，也不产生 INTTMH1 信号。
- <5> 通过重复执行过程<3> 和<4>，可获取具有任意占空比的脉冲。
- <6> 若要停止计数操作，设置 TMHE1 = 0。

若 CMP01 的值是 N，CMP11 的值是 M，计数时钟是 f_{CNT} ，则 PWM 脉冲的输出周期和占空比如下所示。

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

- 注意事项**
1. 在 PWM 输出模式下，CMP11 的设置值可以在定时器计数操作期间改变。但是，在重写 CMP11 寄存器的值之后，需要 3 个或更多操作时钟（使用 TMHMD1 寄存器的 CKS12 到 CKS10 位选择的信号）来传送寄存器的值。
 2. 当定时器停止计数后（TMHE1=0），再次启动计数操作时（TMHE1=1），必须对 CMP11 赋值（即使 CMP11 寄存器的设置值相同，也必须重新赋值）。

(2) 时序图

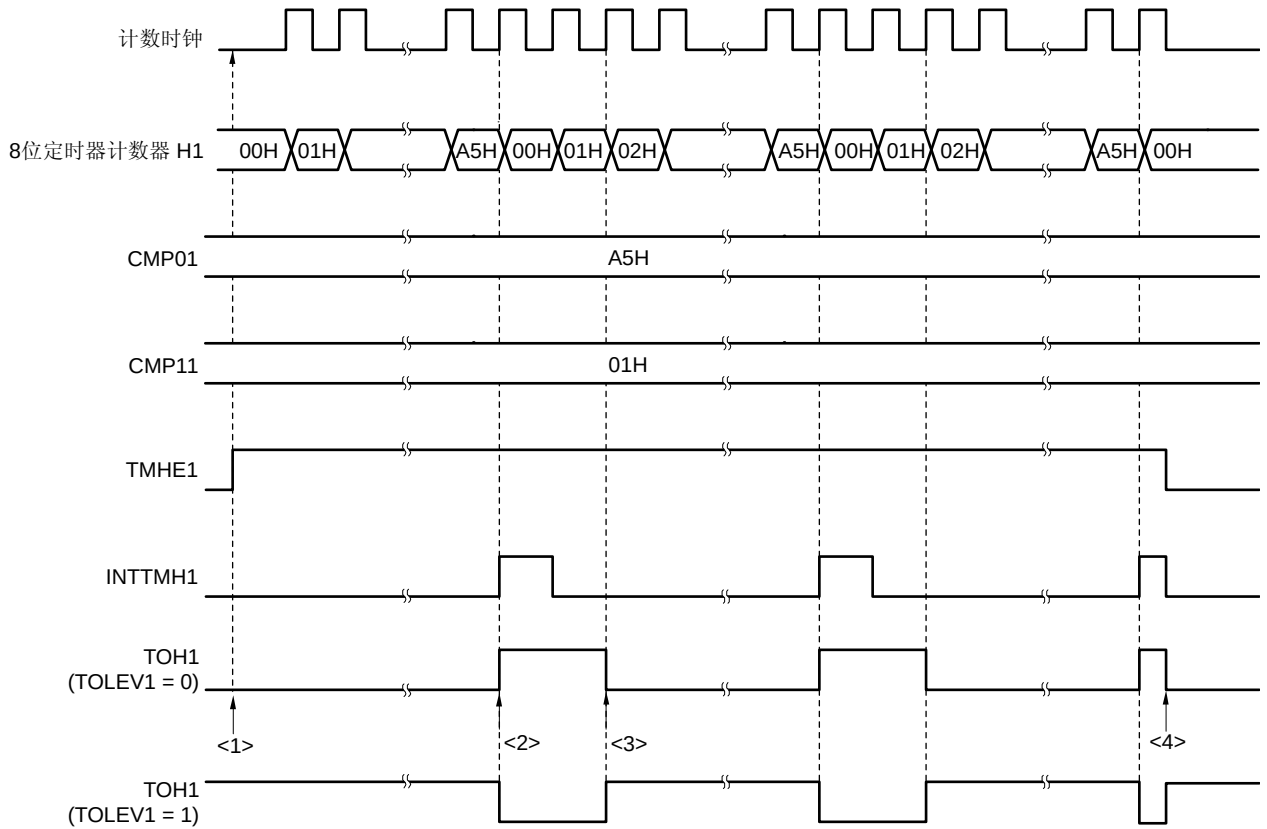
在 PWM 输出模式中，操作时序如下所示

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

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

图 8-9. PWM 输出模式下的时序操作 (1/4)

(a) 基本操作 ($00H < \text{CMP11} < \text{CMP01} < FFH$)



- <1> TMHE1 置 1，则允许计数操作。对齐一个计数时钟来递增计数，启动 8 位定时器计数器 H1。此时，TOH1 输出保持无效 (TOLEV1=0)。
- <2> 当 8 位定时器计数器 H1 和 CMP01 相等时，TOH1 输出电平被反转，8 位定时器计数器 H1 清零，并输出 INTTMH1 中断信号。
- <3> 当 8 位定时器计数器 H1 和 CMP11 相等时，TOH1 输出返回初始电平。此时，8 位定时器计数器 H1 并不清零，也不输出 INTTMH1 中断信号。
- <4> 在定时器 H1 操作期间将 TMHE1 位清零，会使 INTTMH1 信号和 TOH1 输出无效。

图 8-9. PWM 输出模式下的时序操作 (2/4)

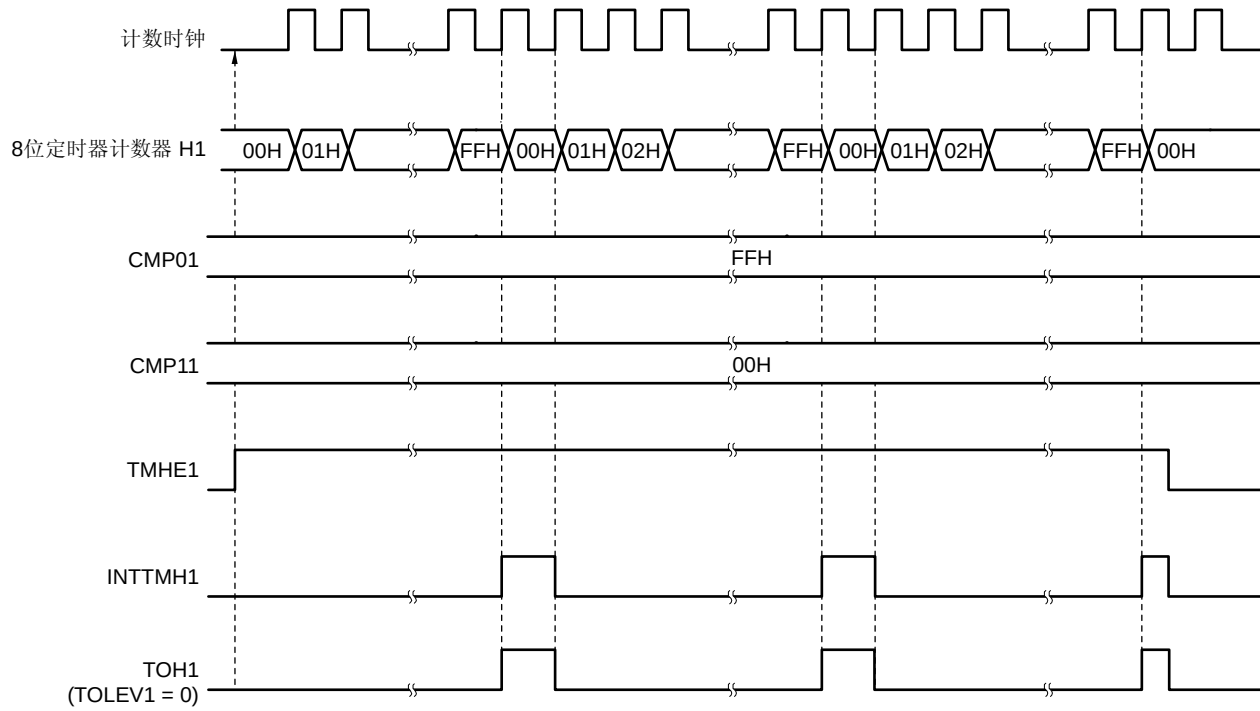
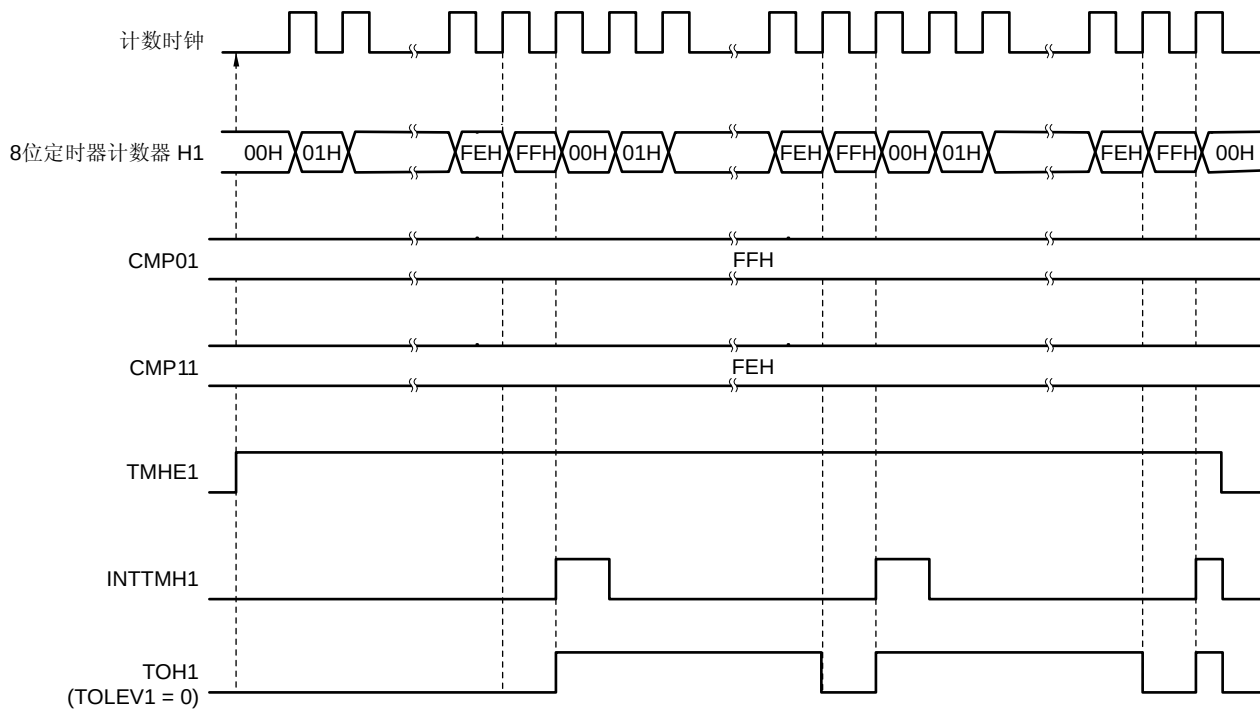
(b) 当 $CMP01 = FFH$, $CMP11 = 00H$ 时的操作(c) 当 $CMP01 = FFH$, $CMP11 = FEH$ 时的操作

图 8-9. PWM 输出模式下的时序操作 (3/4)

(d) 当 CMP01 = 01H, CMP11 = 00H 时的操作

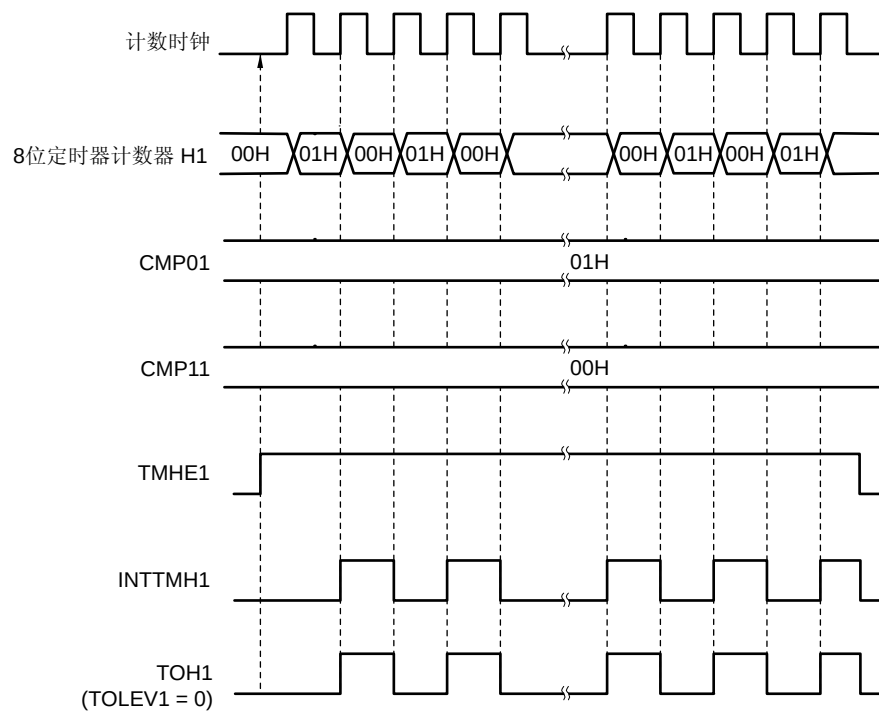
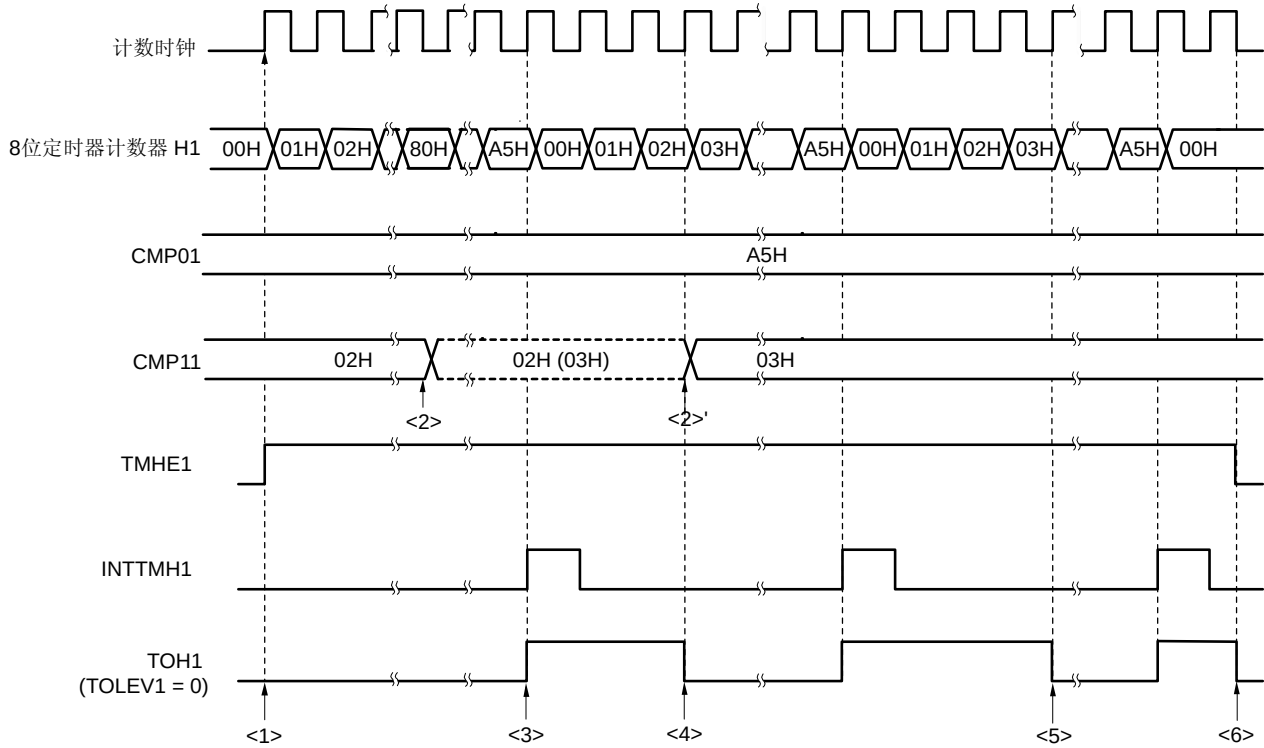


图 8-9. PWM 输出模式下的时序操作 (4/4)

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



- <1> TMHE1 置 1, 则允许计数操作。对齐一个计数时钟来向上计数, 启动 8 位定时器计数器 H1。此时, TOH1 输出保持无效。(当 TOLEV1=0 时)
- <2> CMP11 寄存器的值可以在定时器计数器操作期间改变。该操作与计数时钟不同步。
- <3> 当 8 位定时器计数器 H1 与 CMP01 的值相等时, 8 位定时器计数器 H1 的值清零, TOH1 输出有效, 输出 INTTMH1 信号。
- <4> 如果 CMP11 的值被改变, 则值被锁存, 且不传送到寄存器。当 8 位定时器计数器 H1 的值与修改前 CMP11 寄存器值相等时, 则将修改后的值传送至 CMP11 寄存器, CMP11 的值被改变 (<2>)。但是, 从 CMP11 的值被改变到该值传送到寄存器需要 3 个或更多的计数时钟周期。如果在 3 个计数时钟周期内产生了一个相等信号, 则不能将修改后的值传送到该寄存器。
- <5> 当 8 位定时器计数器 H1 的值与改变后的 CMP11 的值相等时, TOH1 输出变为无效。8 位定时器计数器 H1 不清零, 也不产生 INTTMH1 信号。
- <6> 在定时器 H1 操作期间将 TMHE1 位清零, 会使 INTTMH1 信号和 TOH1 输出无效。

第九章 看门狗定时器

9.1 看门狗定时器的功能

看门狗定时器用于检测意外的程序循环。如果检测到这样的一个程序循环，将产生一个内部复位信号。

如果看门狗定时器产生复位，则复位控制标志寄存器（RESF）的第 4 位（WDTRF）置 1。RESF 的详细信息，请参阅第十五章 复位功能。

表 9-1 看门狗定时器的循环检测时间

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

- 备注
- 1. f_{RL} : 低速内部振荡器的时钟振荡频率
 - 2. f_X : 系统时钟的振荡频率
 - 3. 括号里的值适用于 $f_{RL} = 480$ kHz（最大值）， $f_X = 10$ MHz。

看门狗定时器（WDT）的操作模式可根据片上低速振荡器的选项字节设置进行切换，如表 9-2 所示。

表 9-2 选项字节的设置和看门狗定时器操作模式

	选项字节设置	
	低速内部振荡器不能被停止	低速内部振荡器可以通过软件停止
看门狗定时器时钟源	恒为 f_{RL} ^{注1} .	- 可由软件选择 (f_x, f_{RL} 或停止) - 复位释放后: f_{RL}
复位后的操作	以最大间隔时间启动操作 ($2^{18}/f_{RL}$) .	以最大间隔时间启动操作 ($2^{18}/f_{RL}$) .
操作模式选择	间隔时间只能被修改 1 次	时钟选择/间隔时间只能被修改 1 次
特征	看门狗定时器不能被停止	看门狗定时器可以被停止 ^{注2} .

- 注**
- 只要供电正常，低速内部振荡器就不会停止（复位期间除外）。
 - 如果看门狗定时器时钟源不同，则停止看门狗定时器的供应时钟所需的条件也不同。
 - <1> 如果时钟源是 f_x ，则在满足以下条件时，看门狗定时器的供应时钟停止。
 - 当 f_x 被停止
 - 在 HALT/STOP 模式中。
 - 在振荡器稳定等待时间内
 - <2> 如果时钟源是 f_{RL} ，则在满足以下条件时，看门狗定时器的供应时钟停止。
 - 如果 CPU 时钟是 f_x ，并且在执行 STOP 指令前由软件停止 f_{RL} 。
 - 在 HALT/STOP 模式中。

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

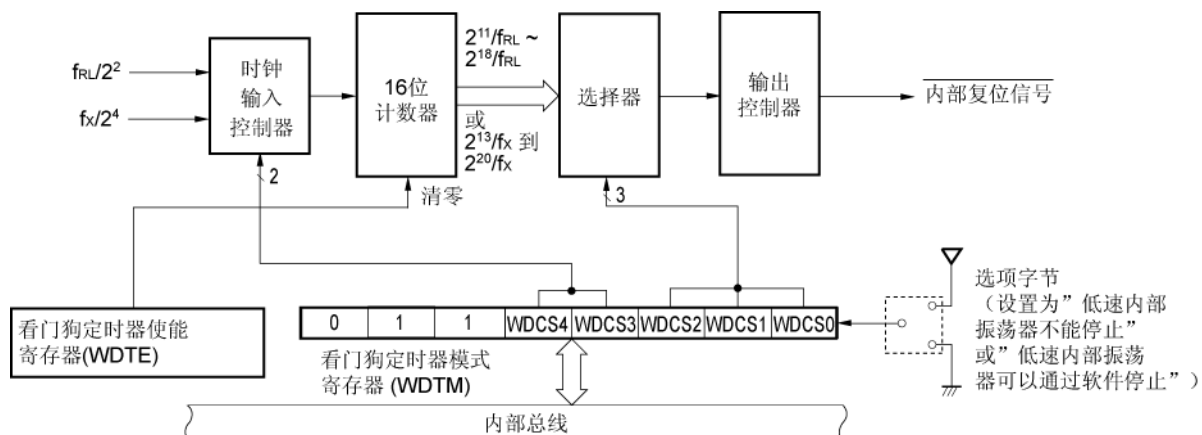
9.2 看门狗定时器的配置

看门狗定时器包括以下硬件。

表 9-3 看门狗定时器的配置

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

图 9-1 看门狗定时器的框图



备注： 1. f_{RL} : 低速内部振荡器的时钟振荡频率
2. f_x : 系统时钟的振荡频率

9.3 控制看门狗定时器的寄存器

看门狗定时器由以下两种寄存器控制

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

(1) 看门狗定时器模式寄存器（WDTM）

该寄存器设置溢出时间和看门狗定时器的操作时钟。

该寄存器可由 8 位存储器操作指令设置，并能被多次读取，但复位释放之后只能被写入 1 次。

复位信号的产生，会将该寄存器置为 67H。

图 9-2. 看门狗定时器模式寄存器的格式（WDTM）

地址: FF48H 复位后: 67H 读/写

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

WDCS4 ^{注1}	WDCS3 ^{注1}	操作时钟选择
0	0	低速内部振荡时钟（f _{RL} ）
0	1	系统时钟（f _x ）
1	×	停止看门狗定时器操作

WDCS2 ^{注2}	WDCS1 ^{注2}	WDCS0 ^{注2}	溢出时间设置	
			低速内部振荡时钟操作期间	系统时钟操作期间
0	0	0	$2^{11}/f_{RL}$ （4.27 ms）	$2^{13}/f_x$ （819.2 μ s）
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. 如果由选项字节指定为“低速内部振荡器不能被停止”，则不能设置该项。无论写入何值，都会选择低速内部振荡器。
 2. 复位释放时为最大周期（WDCS2, 1, 0 = 1, 1, 1）。

- 注意事项**
1. 第 7, 6, 5 位分别设置为 0, 1, 和 1。不能设置为其他值。
 2. 复位释放后, 只能由 8 位存储器操作指令对 **WDTM** 写入一次。如果试图写入第二次, 就会产生内部复位信号。但是, 在第一次写入时, 如果分别设置 **WDSC4** 和 **WDSC3** 为 “1” 和 “x”, 看门狗定时器将停止, 即使有如下操作也不会产生内部复位。
 - 第二次写入 **WDTM**
 - 用 1 位存储器操作指令对 **WDTE** 进行操作
 - 给 **WDTE** 写入一个 “ACH” 以外的值
 3. 不能用 1 位存储器操作指令对 **WDTM** 进行操作。
 4. 当使用自编程方式对 **flash** 存储器进行操作时, 为看门狗设置的溢出时间要足够长 (例如 一个字节写入: 最小 200 μ s, 一个块删除: 最小 10ms)。

- 备注**
1. f_{RL} : 低速内部振荡器的时钟振荡频率
 2. f_x : 系统时钟的振荡频率
 3. x: 无需考虑
 4. 图中括号里的值适用于 $f_{RL} = 480 \text{ kHz}$ (最大值), $f_x = 10 \text{ MHz}$ 。

(2) 看门狗定时器使能寄存器 (WDTE)

将 ACH 写入 WDTE, 可以对看门狗定时器计数器清零并再次启动计数操作。

该寄存器可由 8 位存储器操作指令设置。

复位信号的产生, 会将该寄存器置为 9AH。

图 9-3 看门狗定时器使能寄存器 (WDTE) 的格式

地址: FF49H 复位后: 9AH 读/写

符号	7	6	5	4	3	2	1	0
WDTE								

- 注意事项**
1. 如果一个 ACH 之外的值被写入 **WDTE**, 将产生一个内部复位信号。
 2. 如果使用 1 位存储器操作指令对 **WDTE** 进行操作, 将产生一个内部复位信号。
 3. 从 **WDTE** 读取的值是 9AH (与写入值 (ACH) 不同)。

9.4 看门狗定时器的操作

9.4.1 当选项字节选择“低速内部振荡器不能停止”时，看门狗定时器的操作

看门狗定时器的操作时钟固定为低速内部振荡器时钟

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 位都为 1（WDSC2, WDSC1, WDSC0）=1, 1, 1）启动操作，看门狗定时器操作不能被停止。

复位释放后的看门狗定时器操作如下：

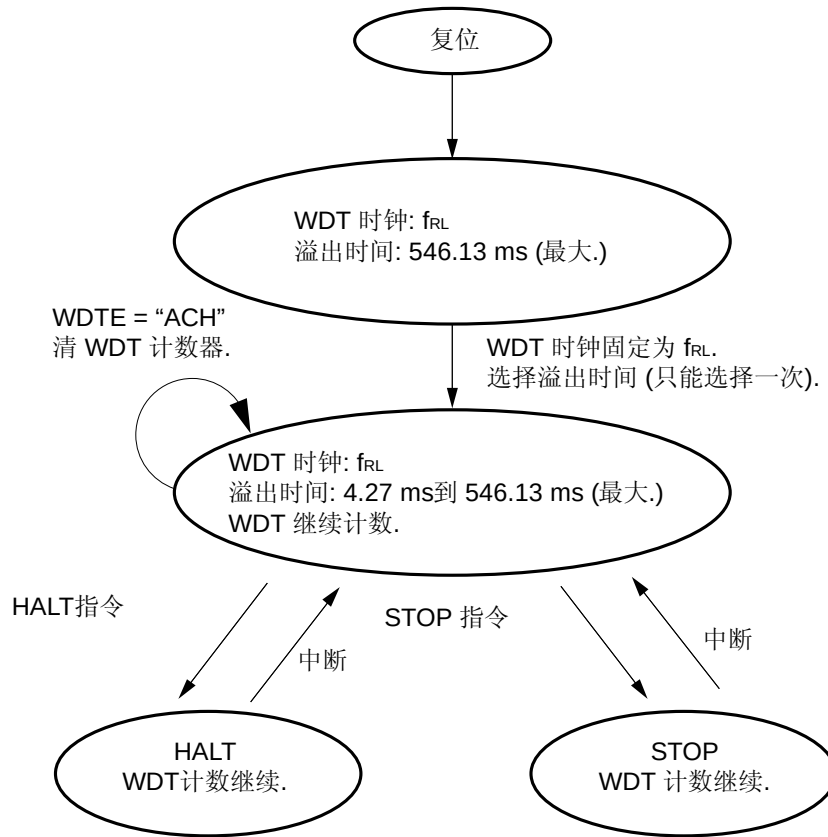
1. 复位释放后的状态如下。
 - 操作时钟： 低速内部振荡器时钟
 - 周期： $2^{18}/f_{RL}$ （546.13 ms） 操作频率为 $f_{RL} = 480 \text{ kHz}$ （最大值）
 - 开始计数
2. 由 8 位存储器操作指令^{注 1, 2}在看门狗定时器模式寄存器（WDTM）中进行以下设置。
 - 周期： 由第 2~0 位（WDSC2~ WDSC0）设置
3. 执行上述过程后，将 ACH 写入 WDTE，计数器清零，允许重新计数。

- 注**
1. 操作时钟（低速内部振荡器时钟）不能改变。任何写入 WDTM 第 3 位和第 4 位（WDSC3, WDSC4）的值将被忽略。
 2. 只要对 WDTM 进行写入操作，看门狗定时器计数器就将立即被清零。

注意事项 在这种模式中，即使是执行 **STOP** 指令，也不能停止看门狗定时器的操作。对于 8 位定时器 H1（TMH1），可选择低速内部振荡器的一个分频作为计数源，因此在 **STOP** 指令执行后，看门狗定时器溢出之前，可使用 TMH1 的中断请求对看门狗定时器清零。如果没有执行该过程，则在 **STOP** 指令执行后，当看门狗定时器溢出时会产生内部复位信号。

状态迁移图如下所示。

图 9-4. 当由选项字节选择“低速内部振荡器不能停止”时的状态迁移图



9.4.2 当选项字节选择“低速内部振荡器可以由软件停止”时，看门狗定时器的操作

看门狗定时器的操作时钟为可选择低速内部振荡时钟或系统时钟。

复位释放后，以低速内部时钟的最大周期（看门狗定时器模式寄存器（WDTM）的第 2 位，第 1 位和第 0 位（WDSC2, WDSC1, WDSC0）= 1, 1, 1）启动操作。

复位释放后的看门狗定时器操作如下：

1. 复位释放后的状态如下。
 - 操作时钟： 低速内部振荡器时钟
 - 周期： $2^{18}/f_{RL}$ （546.13 ms）操作频率为 $f_{RL} = 480$ kHz（最大值）
 - 开始计数
2. 由 8 位存储器操作指令对看门狗定时器模式寄存器（WDTM）进行以下设置^{注1, 2, 3}。
 - 操作时钟： 使用第 3 和第 4 位（WDSC3 和 WDSC4）可以选择以下任何时钟。
 低速内部振荡时钟（ f_{RL} ）
 系统时钟（ f_x ）
 停止看门狗定时器操作
 - 周期： 使用第 2~0 位（WDSC2~WDSC0）设置
3. 执行上述过程后，将 ACH 写入 WDTE，计数器清零，允许重新计数。

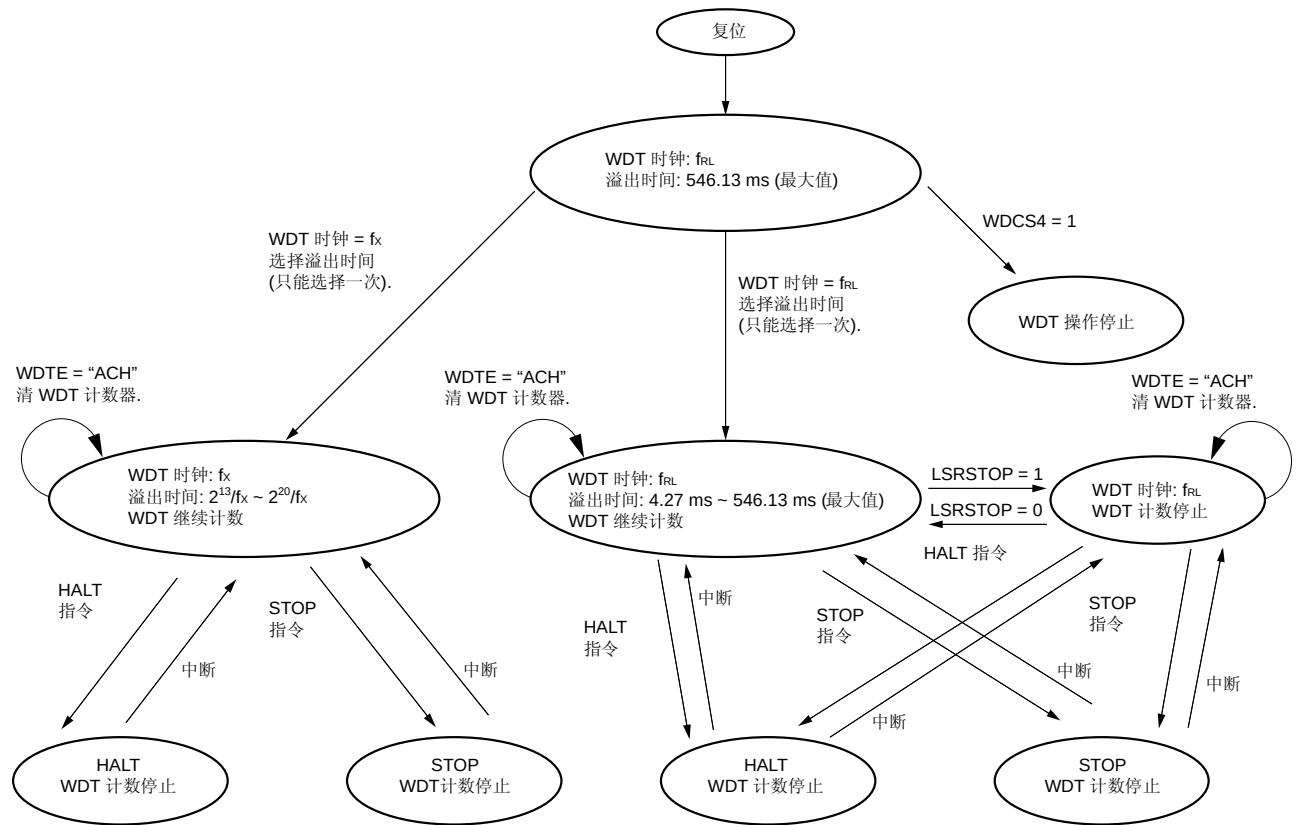
- 注**
1. 只要对 WDTM 进行写入操作，看门狗定时器计数器就将立即被清零。
 2. 第 7, 6, 5 位分别设置为 0, 1, 1。不能设置为其他值。
 3. 在第一次写入时，如果对 WDSC4 和 WDSC3 分别设置为 1 和 x，停止看门狗定时器操作，即使发生下面的操作也不会产生内部复位。”
 - 第二次写入 WDTM
 - 用 1 位存储器操作指令对 WDTE 进行操作
 - 给 WDTE 设置一个 ACH 之外的值

注意事项 此模式中，在 HALT/STOP 指令执行期间，看门狗定时器的操作被停止。在释放 HALT/STOP 模式后，计数器再次启动，使用在执行 HALT/STOP 指令之前由 WDTM 设置的看门狗定时器操作时钟。此时，计数器没有被清零，而是保持原值。

要了解 STOP 和 HALT 模式下每一种状态下看门狗定时器的操作，可参阅 9.4.3 STOP 模式下看门狗定时器的操作（当选项字节选择“低速内部振荡器可以由软件停止”时）和 9.4.4 HALT 模式下看门狗定时器的操作（当选项字节选择“低速内部振荡器不能停止”时）。

状态迁移图如下所示

图 9-5 当选项字节选择“低速内部振荡器可以由软件停止”时的状态迁移图



9.4.3 STOP模式下看门狗定时器的操作（当选项字节选择为“低速内部振荡器可以由软件停止”时）

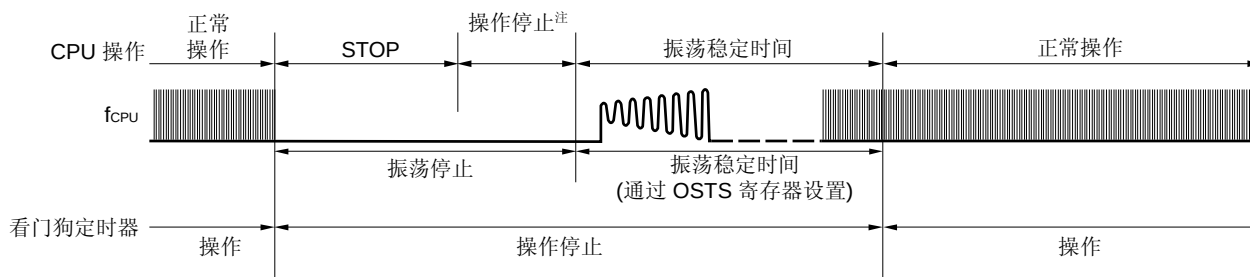
无论使用的是系统时钟还是低速内部振荡时钟，在执行 STOP 指令时，看门狗定时器要停止计数。

（1） 当看门狗定时器操作时钟是外围硬件时钟（fx），且执行 STOP 指令时。

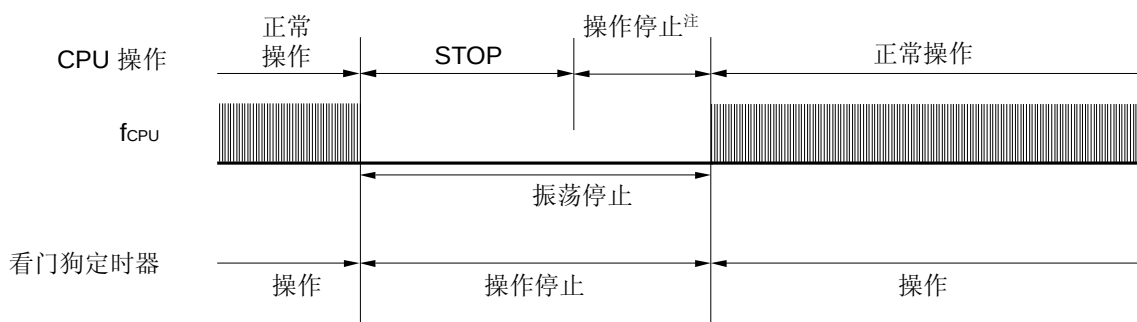
当执行 STOP 指令时，看门狗定时器的操作停止。当 STOP 模式释放后，操作停止 $34\mu\text{s}$ （典型值）（该值是在使用晶体/陶瓷振荡器的情况下，由振荡器稳定时间选择寄存器（OSTS）设置的振荡稳定时间），然后再次开始计数，使用操作停止之前的操作时钟。此时，计数器没有被清零，而是保持原值。

图 9-6 STOP 模式下的操作（WDT 操作时钟：外围硬件时钟）

<1> CPU 时钟：晶体/陶瓷振荡器时钟



<2> CPU 时钟：高速内部振荡时钟或外部时钟输入



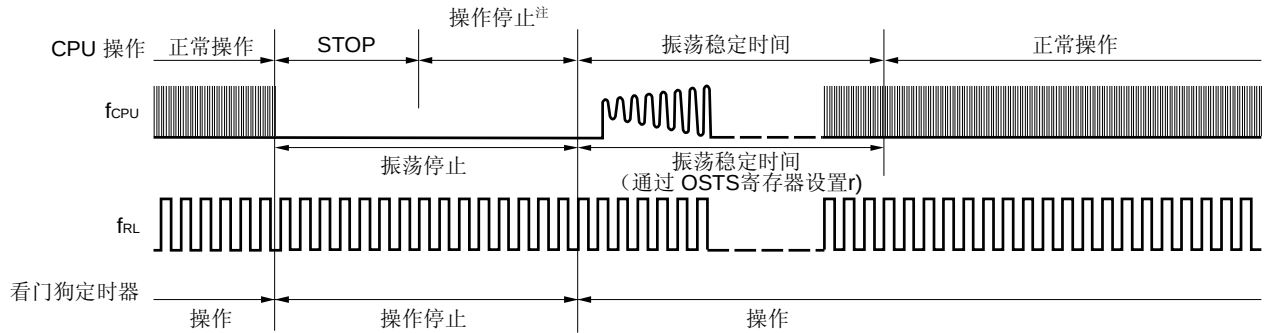
注 操作停止时间是 $17\mu\text{s}$ （最小值）， $34\mu\text{s}$ （典型值）和 $67\mu\text{s}$ （最大值）。

(2) 当看门狗定时器操作时钟是低速内部时钟 (f_{RL})，且执行 STOP 指令时。

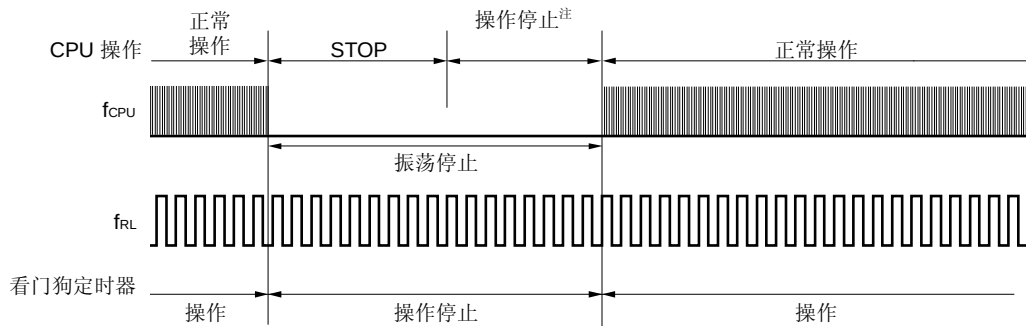
当执行 STOP 指令时，看门狗定时器的操作停止。在释放 STOP 模式后，操作停止 $34\ \mu\text{s}$ （典型值），然后重新开始计数，使用操作停止之前的操作时钟。此时，计数器没有被清零，而是保持原值。

图 9-7. STOP 模式下的操作（WDT 操作时钟：低速内部时钟）

<1> CPU 时钟：晶体/陶瓷振荡器时钟



<2> CPU 时钟：高速内部振荡时钟或外部时钟输入

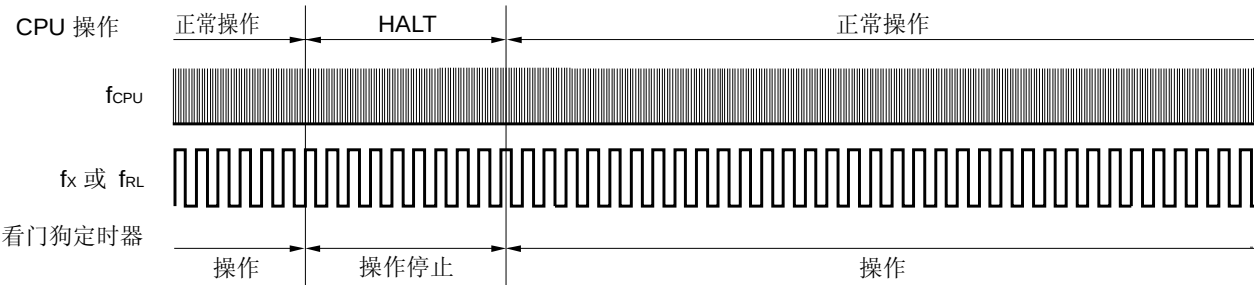


注 操作停止时间是 $17\ \mu\text{s}$ （最小值）， $34\ \mu\text{s}$ （典型值）和 $67\ \mu\text{s}$ （最大值）。

9.4.4 HALT模式下看门狗定时器的操作（当选项字节选择“低速内部振荡器可以由软件停止”时）

无论使用的是系统时钟（f_x）还是低速内部时钟（f_{RL}），在执行 HALT 指令时，看门狗定时器都要停止计数。在释放 HALT 模式后，使用操作停止前的操作时钟重新开始计数。此时，计数器没有被清零，而是保持原值。

图 9-8. HALT 模式下的操作



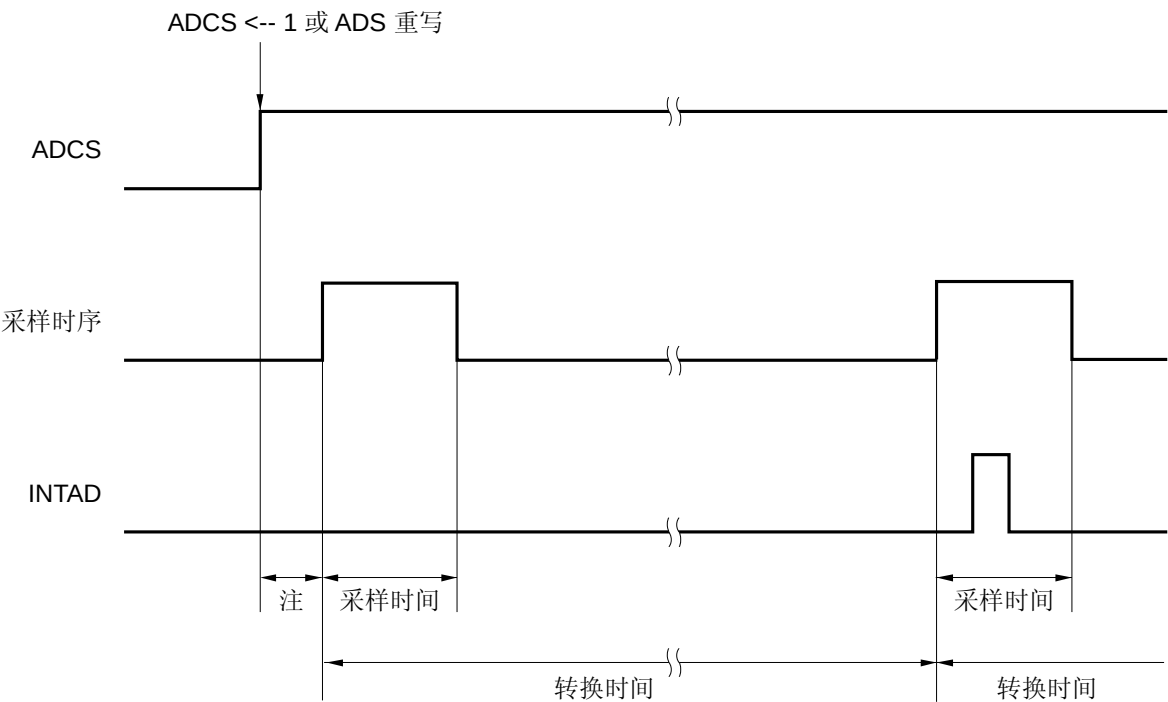
10.1 A/D 转换器的功能

A/D 转换器将模拟信号转换成数字信号，由 4 个通道（ANI0 ~ ANI3）组成，具有 10 位分辨率。
A/D 转换器有以下功能。

- 10 位分辨率 A/D 转换
从模拟输入 ANI0~ANI3 中选择一个通道，重复执行 10 位分辨率的 A/D 转换。每次 A/D 转换结束，都会产生一个中断请求（INTAD）。

图 10-1 为 A/D 采样和 A/D 转换的时序图，表 10-1 为采样时间和 A/D 转换时间。

图 10-1 A/D 转换器采样和 A/D 转换的时序



注 从 ADCS 上升到采样开始，需要 2~3 个时钟。

表 10-1. 采样时间和 A/D 转换时间

参考电压范围 注1	采样时间 注2	转换时间 ^{注3}	f _{XP} = 8 MHz		f _{XP} = 10 MHz		FR2	FR1	FR0
			采样时间 ^{注2}	转换时间 ^{注3}	采样时间 ^{注2}	转换时间 ^{注3}			
AVREF ≥ 4.5 V	12/f _{XP}	36/f _{XP}	1.5 μs	4.5 μs	1.2 μs	3.6 μs	0	0	0
AVREF ≥ 4.0 V	24/f _{XP}	72/f _{XP}	3.0 μs	9.0 μs	2.4 μs	7.2 μs	1	0	0
AVREF ≥ 2.85 V	96/f _{XP}	144/f _{XP}	12.0 μs	18.0 μs	9.6 μs	14.4 μs	1	1	0
	48/f _{XP}	96/f _{XP}	6.0 μs	12.0 μs	4.8 μs	9.6 μs	1	0	1
	48/f _{XP}	72/f _{XP}	6.0 μs	9.0 μs	4.8 μs	7.2 μs	0	1	0
	24/f _{XP}	48/f _{XP}	3.0 μs	6.0 μs	禁止设置 ^{注4} (2.4 μs)	禁止设置 ^{注4} (4.8 μs)	0	0	1
AVREF ≥ 2.7 V	176/f _{XP}	224/f _{XP}	22.0 μs	28.0 μs	17.6 μs	22.4 μs	1	1	1
	88/f _{XP}	112/f _{XP}	11.0 μs	14.0 μs	禁止设置 ^{注4} (8.8 μs)	禁止设置 ^{注4} (11.2 μs)	0	1	1

注 1. 设置 FR2, FR1, 和 FR0 时, 确保要与参考电压范围一致, 来满足以下注 2 和注 3 的条件。

例如 当 AVREF ≥ 2.7 V, f_{XP} = 8 MHz 时,

- 采样时间为 11.0 μs 或更长, A/D 转换时间为 14.0~100 μs。

- 设置 FR2, FR1, 和 FR0 = 0, 1, 1 或 1, 1, 1。

2. 设置采样时间如下。

- AVREF ≥ 4.5 V: 1.0 μs 或更长
- AVREF ≥ 4.0 V: 2.4 μs 或更长
- AVREF ≥ 2.85 V: 3.0 μs 或更长
- AVREF ≥ 2.7 V: 11.0 μs 或更长

3. 设置 A/D 转换时间如下。

- AVREF ≥ 4.5 V: 3.0 μs ~100 μs
- AVREF ≥ 4.0 V: 4.8 μs ~100 μs
- AVREF ≥ 2.85 V: 6.0 μs ~100 μs
- AVREF ≥ 2.7 V: 14.0 μs ~100 μs

<R>

4. 禁止设置, 因为该值不满足注 2 和注 3 的条件。

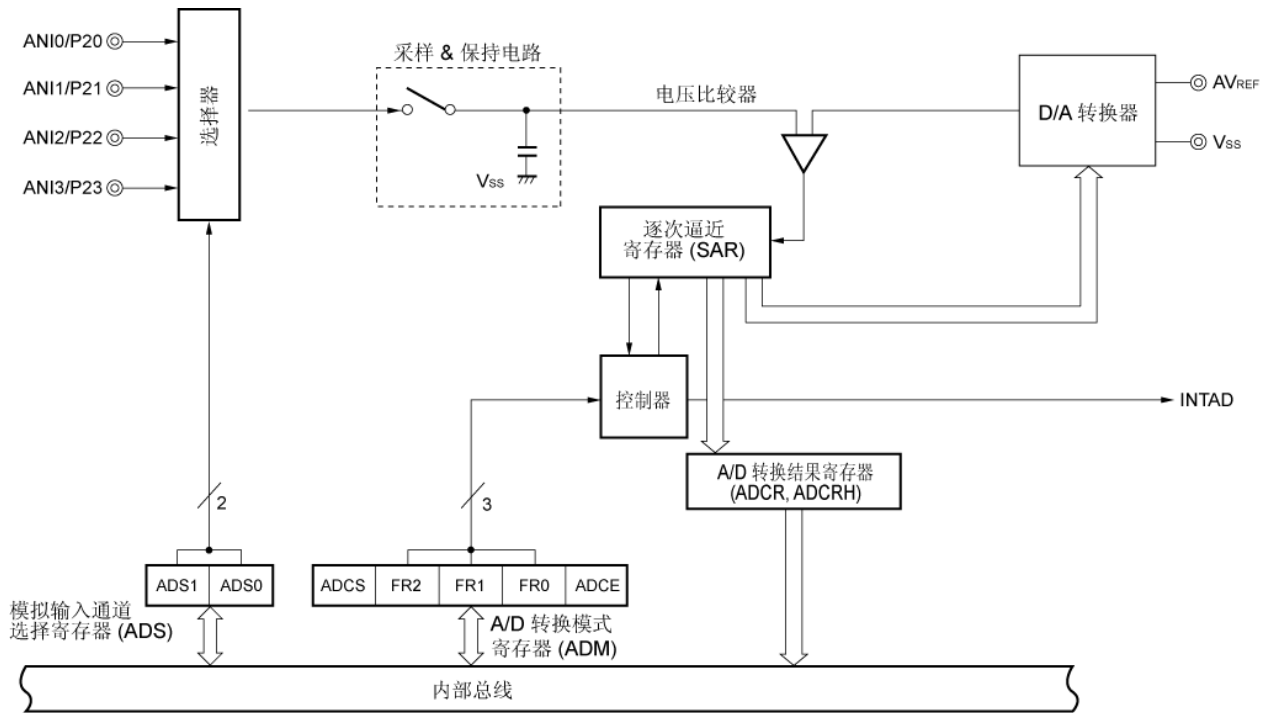
注意事项 以上采样时间和转换时间不包括时钟频率误差。选择采样时间和转换时间时除了要满足注 2 和注 3 外, 还要将时钟频率误差考虑在内 (使用高速内部振荡器时, 最大误差范围为±5%)。

备注 1. f_{XP}: 外围硬件时钟的振荡频率

2. 转换时间包括采样时间和对采样值进行逐次比较直到输出转换结果的时间。

图 10-2 显示了 A/D 转换器的框图。

图 10-2 A/D 转换器的框图



10.2 A/D 转换器的配置

A/D 转换器由下列硬件组成。

(1) ANI0~ANI3 引脚

这些是 4 通道 A/D 转换器的模拟输入引脚。它们输入模拟信号，转换为数字信号。除了那些被模拟输入通道指定寄存器 (ADS) 选为模拟输入的引脚外，其他引脚均可用作 I/O 端口引脚。

(2) 采样 & 保持电路

在 A/D 转换开始后，采样&保持电路对选择器选中的模拟输入引脚的输入信号进行采样，并在 A/D 转换期间保持采样到的模拟输入电压值。

(3) D/A 转换器

D/A 转换器用于连接 AV_{REF} 和 V_{SS} ，并产生一个电压，与模拟输入信号比较。

(4) 电压比较器

电压比较器用于比较采样的模拟输入电压和 D/A 转换的输出电压。

(5) 逐次逼近寄存器 (SAR)

该寄存器用于比较采样的模拟电压和 D/A 转换器的电压，并从最高有效位 (MSB) 开始输出转换结果。

当电压值转换为数字值直到最低有效位 (LSB) (A/D 转换结束) 时，将 SAR 寄存器的内容传送给 A/D 转换结果寄存器 (ADCR)。

(6) 10 位 A/D 转换结果寄存器 (ADCR)

每次 A/D 转换结束时，将 A/D 转换的结果从逐次逼近寄存器传送给该寄存器中，并将转换结果保存在 ADCR 的低 10 位中（高 6 位固定为 0）。

(7) 8 位 A/D 转换结果寄存器 (ADCRH)

每次 A/D 转换结束时，将 A/D 转换的结果从逐次逼近寄存器传送给该寄存器中，并将转换结果保存在 ADCR 的高 8 位中。

(8) 控制器

A/D 转换结束，产生中断 INTAD。

(9) AV_{REF} 引脚

该引脚给 A/D 转换器输入模拟电源/参考电压。不使用 A/D 转换器时，该引脚连接到 V_{DD}。

ANI0 至 ANI3 引脚的输入信号被转换成数字信号，是以 AV_{REF} 和 AV_{SS} 之间的电压为基准。

(10) AV_{SS} 引脚

这是 A/D 转换器的地电位引脚。即使不使用 A/D 转换器时，该引脚的电平应该始终和 V_{SS} 相等。

(11) A/D 转换器模式寄存器 (ADM)

该寄存器用来设置转换模拟输入信号的转换时间，和开始或停止转换操作。

(12) 模拟输入通道选择寄存器 (ADS)

该寄存器用来指定模拟输入端口，将输入到该端口引脚的模拟输入电压转换为数字信号。

(13) 端口模式控制寄存器 2 (PMC2)

当 P20/ANI0 至 P23/ANI3 引脚作为 A/D 转换器的模拟输入引脚时，使用该寄存器。

10.3 A/D 转换器中使用的寄存器

A/D 转换器使用以下六种控制寄存器。

- A/D 转换器模式寄存器 (ADM)
- 模拟输入通道选择寄存器 (ADS)
- 10 位 A/D 转换结果寄存器 (ADCR)
- 8 位 A/D 转换结果寄存器 (ADCRH)
- 端口模式控制寄存器 2 (PMC2)
- 端口模式寄存器 2 (PM2)

(1) A/D 转换器模式寄存器 (ADM)

该寄存器用来设置模拟输入的 A/D 转换时间，以及启动/停止转换操作。

ADM 可以通过 1 位或 8 位存储器操作指令设置。

复位信号的产生，会将该寄存器清为 00H。

图 10-3. A/D 转换器模式寄存器 (ADM) 的格式

地址: FF80H 复位后: 00H 读/写

符号	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	FR2	FR1	FR0	0	0	ADCE

ADCS	A/D 转换操作控制
0	停止转换操作
1 ^{※1}	开始转换操作

FR2	FR1	FR0	参考电压 范围 ^{※2}	采样 时间 ^{※3}	转换 时间 ^{※4}	f _{XP} = 8 MHz		f _{XP} = 10 MHz	
						采样时间 ^{※3}	转换时间 ^{※4}	采样时间 ^{※3}	转换时间 ^{※4}
0	0	0	AVREF ≥ 4.5 V	12/f _{XP}	36/f _{XP}	1.5 μs	4.5 μs	1.2 μs	3.6 μs
1	0	0	AVREF ≥ 4.0 V	24/f _{XP}	72/f _{XP}	3.0 μs	9.0 μs	2.4 μs	7.2 μs
1	1	0	AVREF ≥ 2.85 V	96/f _{XP}	144/f _{XP}	12.0 μs	18.0 μs	9.6 μs	14.4 μs
1	0	1		48/f _{XP}	96/f _{XP}	6.0 μs	12.0 μs	4.8 μs	9.6 μs
0	1	0		48/f _{XP}	72/f _{XP}	6.0 μs	9.0 μs	4.8 μs	7.2 μs
0	0	1		24/f _{XP}	48/f _{XP}	3.0 μs	6.0 μs	禁止设置 ^{※5} (2.4 μs)	禁止设置 ^{※5} (4.8 μs)
1	1	1	AVREF ≥ 2.7 V	176/f _{XP}	224/f _{XP}	22.0 μs	28.0 μs	17.6 μs	22.4 μs
0	1	1		88/f _{XP}	112/f _{XP}	11.0 μs	14.0 μs	禁止设置 ^{※5} (8.8 μs)	禁止设置 ^{※5} (2.4 μs)

ADCE	比较操作控制 ^{※6}
0 ^{※1}	停止比较器的操作
1	允许比较器的操作

- 备注
1. f_{XP}: 外围硬件时钟的振荡频率
 2. 转换时间包括采样时间和对采样值进行连续比较直到转换结果输出的时间。

注 1. 即使当 ADCE = 0 (比较器操作停止) 时, 如果 ADCS = 1, A/D 转换操作也会开始。但是, 第一次转换的数据不在保证的值范围之内, 因此应该忽略第一次的转换结果。

注 2. 设置 FR2, FR1, 和 FR0 时, 确保要与参考电压范围一致, 来满足以下注 3 和 注 4 的条件。

例如 当 $AV_{REF} \geq 2.7 \text{ V}$, $f_{XP} = 8 \text{ MHz}$ 时

- 采样时间为 $11.0 \mu\text{s}$ 或更长, A/D 转换时间为 $14.0 \sim 100 \mu\text{s}$ 。

- 设置 FR2, FR1, 和 $FR0 = 0, 1, 1$ 或 $1, 1, 1$ 。

3. 设置采样时间如下。

- $AV_{REF} \geq 4.5 \text{ V}$: $1.0 \mu\text{s}$ 或更长
- $AV_{REF} \geq 4.0 \text{ V}$: $2.4 \mu\text{s}$ 或更长
- $AV_{REF} \geq 2.85 \text{ V}$: $3.0 \mu\text{s}$ 或更长
- $AV_{REF} \geq 2.7 \text{ V}$: $11.0 \mu\text{s}$ 或更长

4. 设置 A/D 转换时间如下。

- $AV_{REF} \geq 4.5 \text{ V}$: $3.0 \mu\text{s} \sim 100 \mu\text{s}$
- $AV_{REF} \geq 4.0 \text{ V}$: $4.8 \mu\text{s} \sim 100 \mu\text{s}$
- $AV_{REF} \geq 2.85 \text{ V}$: $6.0 \mu\text{s} \sim 100 \mu\text{s}$
- $AV_{REF} \geq 2.7 \text{ V}$: $14.0 \mu\text{s} \sim 100 \mu\text{s}$

<R>

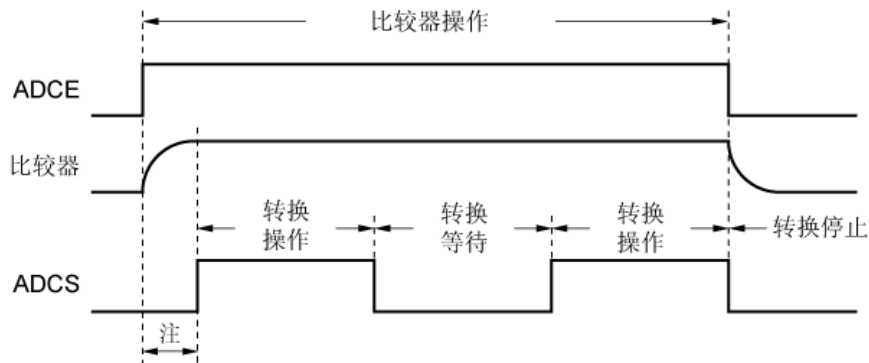
5. 禁止设置, 因为该值不满足注 3 和注 4 的条件。

6. 比较操作通过 ADCS 和 ADCE 来控制, 从操作开始到操作稳定需要 $1 \mu\text{s}$ 的时间。因此, 从 $ADCE=1$, 经过 $1 \mu\text{s}$ 或更长时间, 再将 ADCS 设置为 1, 此时的转换结果比第一次转换结果精确。如果 ADCS 设置为 1 前没有等待 $1 \mu\text{s}$ 或更长时间, 则应该忽略第一次转换得到的数据。

表 10-2. ADCS 和 ADCE 的设置

ADCS	ADCE	A/D 转换操作
0	0	停止状态 (不存在直流电源消耗路径)
0	1	转换等待模式 (仅有比较器电源消耗)
1	×	转换模式

图 10-4. 当使用比较器时的时序图



注 从 ADCE 位上升沿到 ADCS 的上升沿, 至少要等待 $1 \mu\text{s}$ 的时间, 内部电路才能稳定。

注意事项 1. 以上采样时间和转换时间不包括时钟频率误差。选择采样时间和转换时间时除了要满足注 3 和注 4 外, 还要将时钟频率误差考虑在内 (使用高速内部振荡器时, 最大误差范围为 $\pm 5\%$)。

- 注意事项**
2. 当 A/D 转换停止 (ADCS = 0) 时, 对 ADM 中 ADCS 之外的位进行操作, 然后启动 A/D 转换, 需要执行两条 NOP 指令或一条执行时间等于 2 个机器周期的指令, 再设置 ADCS=1。
 3. 在重写 FR0~FR2 时, A/D 转换必须停止 (ADCS = 0)。
 4. 确保将第 6 位、第 2 位和第 1 位清 0。

(2) 模拟输入通道选择寄存器 (ADS)

该寄存器用来指定端口, 将输入到该端口引脚的模拟输入电压转换为数字信号。

ADS 可通过 1 位或 8 位存储器操作指令设置。

复位信号的产生, 会将该寄存器清为 00H。

图 10-5. 模拟输入通道选择寄存器 (ADS) 的格式

地址: FF81H 复位后: 00H 读/写

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

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

注意事项 确保 ADS 的第 2~7 位清 0。

(3) 10 位 A/D 转换结果寄存器 (ADCR)

该寄存器是个 16 位的寄存器, 用来存储 A/D 转换的结果。该寄存器的高 6 位固定为 0。每次 A/D 转换结束后, 转换的结果从逐次比较寄存器中读出存储到 ADCR 中, 在 ADCR 中从 FF19H 的第 1 位开始按顺序存储。FF19H 存放转换结果的高 2 位, FF18H 存放转换结果的低 8 位。

ADCR 的值可以通过 16 位存储器操作指令读取。

复位信号的产生, 使 ADCR 的值不确定。

图 10-6. 10 位 A/D 转换结果寄存器 (ADCR) 的格式

地址: FF18H, FF19H 复位后: 不确定 读

符号	FF19H						FF18H									
ADCR	0	0	0	0	0	0										

注意事项 当对 A/D 转换模式寄存器 (ADM) 和模拟输入通道选择寄存器 (ADS) 进行写操作时, ADCR 的内容可能不确定。要在 ADM 和 ADS 进行写操作之前, 且在转换结束之后, 读取转换结果。在其他时间读取, 可能会读得不正确的转换结果。

(4) 8 位 A/D 转换结果寄存器 (ADCRH)

该寄存器是一个 8 位的寄存器，用来存储 A/D 转换的结果。它存储 10 位转换结果的高 8 位。

ADCRH 的值可以通过 8 位存储器操作指令读取。

复位信号的产生，会使 ADCRH 的值不确定。

图 10-7. 8 位 A/D 转换结果寄存器 (ADCRH) 的格式

地址: FF1AH 复位后: 不确定 读

符号	7	6	5	4	3	2	1	0
ADCRH								

(5) 端口模式控制寄存器 2 (PMC2) 和端口模式寄存器 2 (PM2)

当使用 P20/ANI0 ~ P23/ANI3 引脚作为模拟输入时，设置 PMC20 ~ PMC23 和 PM20 ~ PM23 为 1。此时，P20 ~ P23 的输出锁存可能为 0 或 1。

PMC2 和 PM2 可以通过 1 位或 8 位存储器操作指令设置。

复位信号的产生，会将 PMC2 清为 00H，将 PM2 置为 FFH。

图 10-8 端口模式控制寄存器 2 (PMC2) 的格式

地址: FF84H 复位后: 00H 读/写

符号	7	6	5	4	3	2	1	0
PMC2	0	0	0	0	PMC23	PMC22	PMC21	PMC20

PMC2n	操作模式选择 (n = 0 ~ 3)
0	端口模式
1	A/D 转换模式

注意事项 当 PMC20 ~ PMC23 设置为 1 时，P20/ANI0~P23/ANI3 引脚不能用做端口引脚。将引脚设置为 A/D 转换器模式时，确保上拉电阻选择寄存器 (PU20~PU23) 置为 0。

图 10-9. 端口模式寄存器 2 (PM2) 的格式

地址: FF22H 复位后: FFH 读/写

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

PM2n	P2n 引脚 I/O 模式选择 (n = 0 ~ 3)
0	输出模式 (输出缓冲打开)
1	输入模式 (输出缓冲关闭)

10.4 A/D 转换器的操作

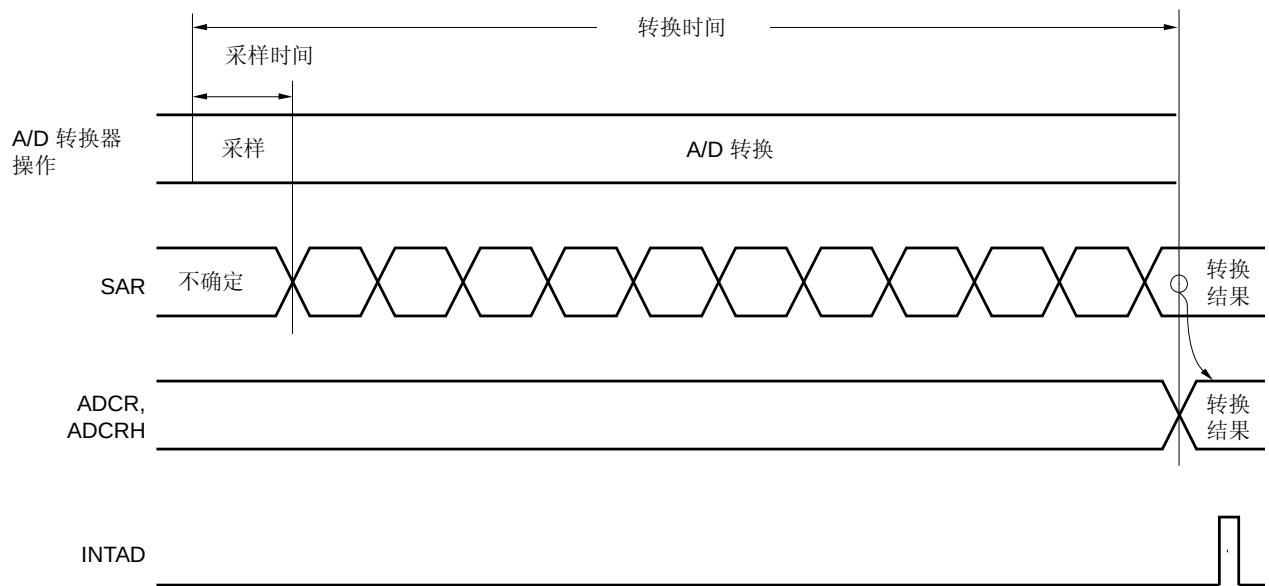
10.4.1 A/D 转换器的基本操作

- <1> 设置 ADCE 为 1。
- <2> 使用模拟输入通道选择寄存器（ADS），选择一个通道进行 A/D 转换。用 FR2~FR0 选择转换时间。
- <3> 执行两条 NOP 指令或一条执行时间等于 2 个机器周期的指令。
- <4> 设置 ADCS 为 1，启动转换操作。
（<5> ~ <11> 操作由硬件实现）
- <5> 采样&保持电路对选中的模拟输入通道的输入电压进行采样。
- <6> 经过一段采样时间后，采样&保持电路处于保持状态，在 A/D 转换结束前，一直保持输入的模拟电压。
- <7> 设置逐次比较寄存器的第 9 位。通过分接选择器将 D/A 转换分接电压置为 $(1/2) AV_{REF}$ 。
- <8> D/A 转换分接电压和模拟输入电压之间的电压差通过电压比较器进行比较。如果模拟输入电压大于 $(1/2) AV_{REF}$ ，SAR 中 MSB 为 1。如果模拟输入电压小于 $(1/2) AV_{REF}$ ，SAR 中 MSB 为 0。
- <9> 接下来，SAR 的第 8 位自动置 1，并进入下一个比较过程。根据第 9 位的预置值选择 D/A 转换分接电压。描述如下：
 - 第 9 位 = 1: $(3/4) AV_{REF}$
 - 第 9 位 = 0: $(1/4) AV_{REF}$
 比较分接电压和模拟输入电压，并设置 SAR 的第 8 位，如下所示。
 - 模拟输入电压 $\geq$ 分接电压: 第 8 位 = 1
 - 模拟输入电压 $<$ 分接电压: 第 8 位 = 0
- <10> 按此方式继续比较，直至 SAR 的第 0 位。
- <11> 全部 10 位都比较完成后，有效的数字结果保存在 SAR 中，并将该结果传输给 A/D 转换结果寄存器（ADCR，ADCRH）中，然后锁存。
同时，产生 A/D 转换结束中断请求（INTAD）。
- <12> 重复执行<5> 到 <11>，直到 ADCS 清 0。
要停止 A/D 转换，ADCS 清 0。
如要从 ADCE=1 的状态重新启动 A/D 转换，从步骤<3>开始。如要从 ADCE=0 的状态重新启动 A/D 转换，从步骤<1>开始（当不改变通道和转换时间时，忽略步骤<1>）。

- 注意事项**
- 1. 必须确保步骤<1> 到 <4> 的操作时间为 1 μ s 或更长。
 - 2. 若将步骤<1>和<2>的次序颠倒，也不会有问题。

- 备注**
- 可以使用以下两种类型的 A/D 转换结果寄存器。
- ADCR（16 位）：存储 10 位 A/D 转换值。
 - ADCRH（8 位）：存储 8 位 A/D 转换值。

图 10-10. A/D 转换器的基本操作



在用软件将 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 重置为 0 之前, 连续执行 A/D 转换操作。

如果在 A/D 转换操作期间, 对 ADM 或模拟输入通道选择寄存器 (ADS) 执行写操作, 则转换操作被初始化。并且如果 ADCS 置 1, 转换将再次从头开始。

复位信号的产生, 会使 A/D 转换结果寄存器 (ADCR, ADCRH) 的值不确定。

10.4.2 输入电压和转换结果

输入到模拟输入引脚（ANI0~ANI3）的模拟输入电压和理论 A/D 转换结果（存储在 10 位 A/D 转换结果寄存器（ADCR））的关系如下。

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

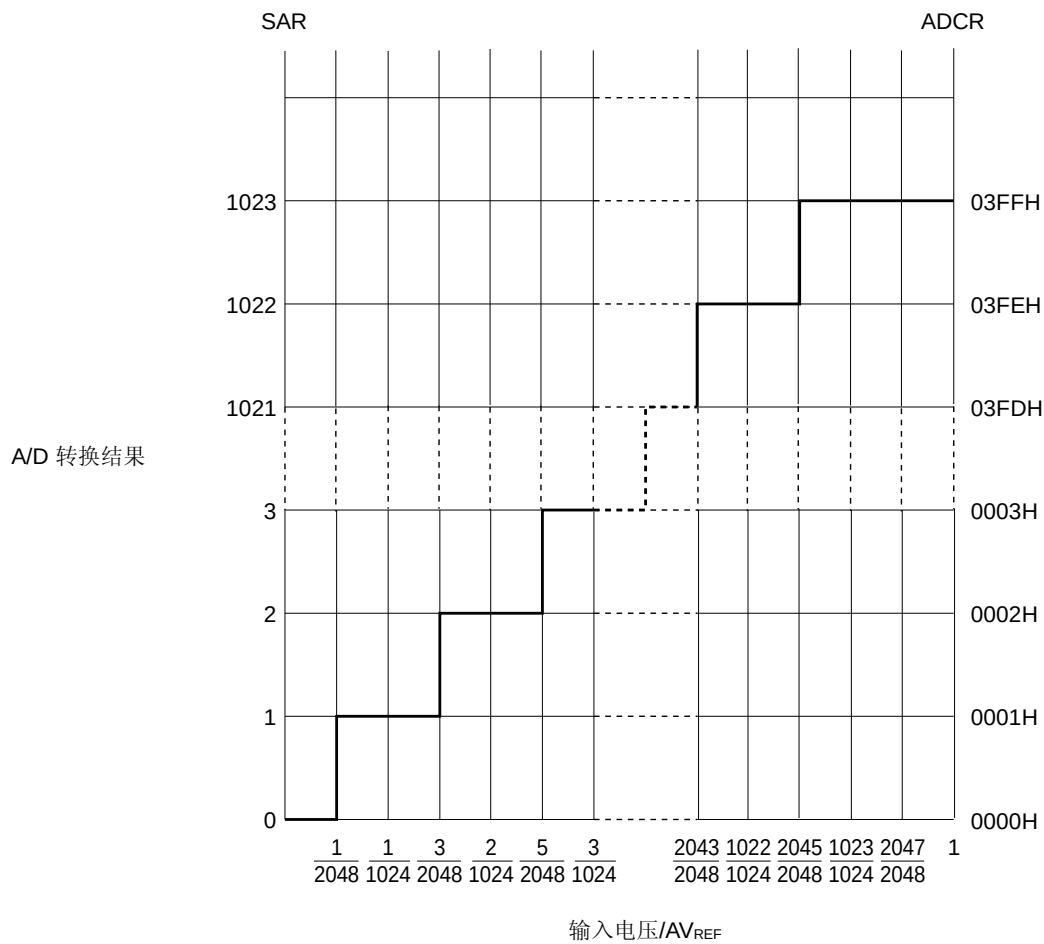
或

$$(\text{ADCR} - 0.5) \times \frac{V_{\text{REF}}}{1024} \leq V_{\text{AIN}} < (\text{ADCR} + 0.5) \times \frac{V_{\text{REF}}}{1024}$$

此处 INT(): 该函数返回括号中数值的整数部分
 V_{AIN} : 模拟输入电压
 V_{REF} : V_{REF} 引脚电压
 ADCR: 10 位 A/D 转换结果寄存器（ADCR）的值

图 10-11 显示模拟输入电压和 A/D 转换结果之间的关系。

图 10-11 模拟输入电压和 A/D 转换结果之间的关系



10.4.3 A/D 转换器操作模式

A/D 转换器的操作模式是选择模式。通过模拟输入通道选择寄存器（ADS），从 ANI0~ANI3 中选择其中一个模拟输入通道，并执行 A/D 转换。

(1) A/D 转换操作

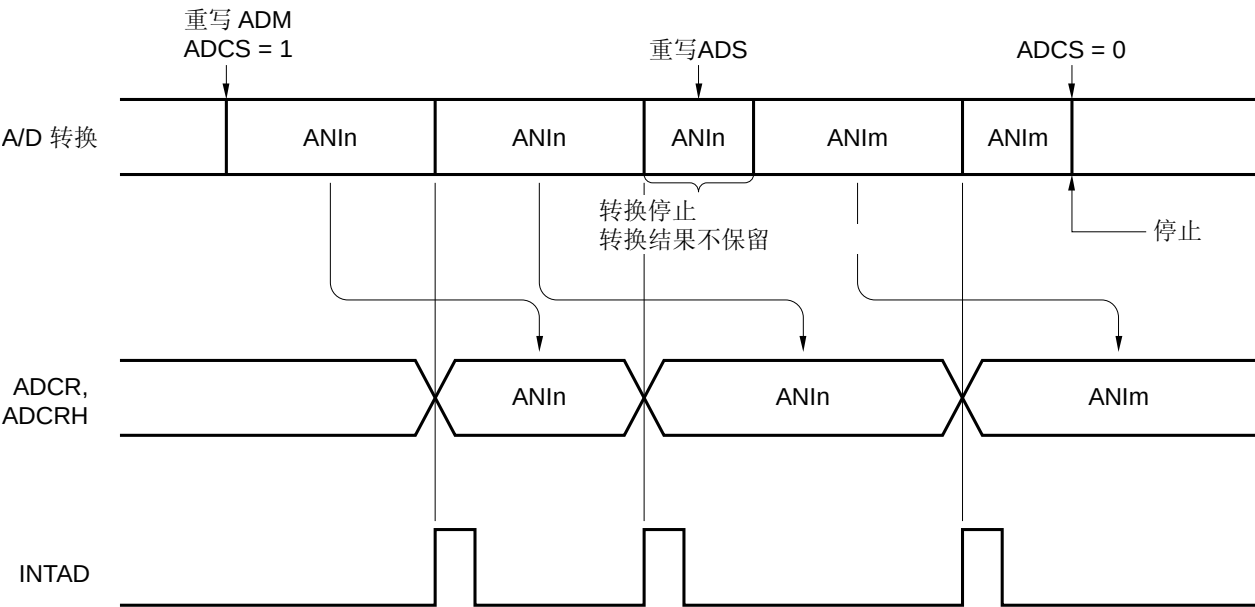
通过设置 A/D 转换模式寄存器（ADM）的第 7 位（ADCS）为 1，可启动对电压的 A/D 转换操作。该电压由模拟输入通道选择寄存器（ADS）选择的模拟输入引脚输入。

当 A/D 转换完成时，转换结果存储在 A/D 转换结果寄存器（ADCR，ADCRH），并且产生一个中断请求信号（INTAD）。A/D 转换一旦开始启动，当一次 A/D 转换结束时，下一次 A/D 转换立即启动。A/D 转换操作会持续进行，直到有新数据写入 ADS。

若在 A/D 转换期间，ADM 或 ADS 值被重写，则要停止当前的 A/D 转换操作，重新开始。

若在 A/D 转换期间，ADCS 被清 0，则 A/D 转换立即停止。此时，转换结果是不确定的。

图 10-12 A/D 转换操作



备注 1. n = 0 ~ 3
2. m = 0 ~ 3

设置方式描述如下。

- <1> A/D 转换模式寄存器 (ADM) 的第 0 位 (ADCE) 设置为 1。
- <2> 通过模拟输入通道选择寄存器 (ADS) 的第 1 位和第 0 位 (ADS1, ADS0) 和 ADM 的第 5 位到第 3 位 (FR2~FR0)，来选择通道和转换时间。
- <3> 执行两条 NOP 指令或一条相当于 2 个机器周期的指令。
- <4> 设置 ADM 的第 7 位 (ADCS) 为 1，启动 A/D 转换。
- <5> 产生中断请求信号 (INTAD)。
- <6> 将 A/D 转换数据传送到 A/D 转换结果寄存器 (ADCR, ADCRH)。

<改变通道>

- <7> 使用 ADS 的第 1 位和第 0 位 (ADS1, ADS0) 改变通道。
- <8> 产生中断请求信号 (INTAD)。
- <9> 将 A/D 转换数据传送到 A/D 转换结果寄存器 (ADCR, ADCRH)。

<A/D 转换完成>

- <10> ADCS 清 0。
- <11> ADCE 清 0。

- 注意事项**
- 1. 必须确保步骤<1> 到 <4> 的操作时间为 1 μ s 或更长。
 - 2. 若将步骤<1>和<2>的次序颠倒，也不会有问题。
 - 3. 步骤<1> 可以忽略。但是，在这种情况下中当步骤<4>执行后，忽略第一次转换结果。
 - 4. 步骤<5>到<8>所经历的时间与 ADM 的第 5 位到第 3 位 (FR2~FR0) 设置的转换时间不同。步骤<7>到步骤<8>所经历的时间即为转换时间，转换时间由 FR2~FR0 设置。

10.5 如何阅读A/D转换器特征表

这里解释 A/D 转换器的专用术语

(1) 分辨率

这是可识别的最小模拟输入电压。也就是说，每位数字输出的模拟输入电压的百分比，叫 1LSB（最低有效位）。1LSB 对满量程的百分比用 %FSR（满量程范围）表示。

当分辨率为 10 位时，1LSB 表示如下。

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

准确度和分辨率无关，而由总误差决定。

(2) 总误差

总误差是指实际测量值与理论值之间的最大误差值。

零度误差、满量程误差、积分线性误差和微分线性误差等，组合起来表示总误差。

需要注意的是，特征表中的总误差不包括量化误差。

(3) 量化误差

当模拟值转换为数字值时，通常会产生 $\pm 1/2\text{LSB}$ 的误差。在 A/D 转换器中，相差 $\pm 1/2\text{LSB}$ 的模拟输入电压会被转换成相同的数字代码，因此量化误差不可避免。

需要注意的是，量化误差不属于特征表中总误差、零度误差、满度误差、积分线性误差和微分线性误差的范围。

图 10-13. 总误差

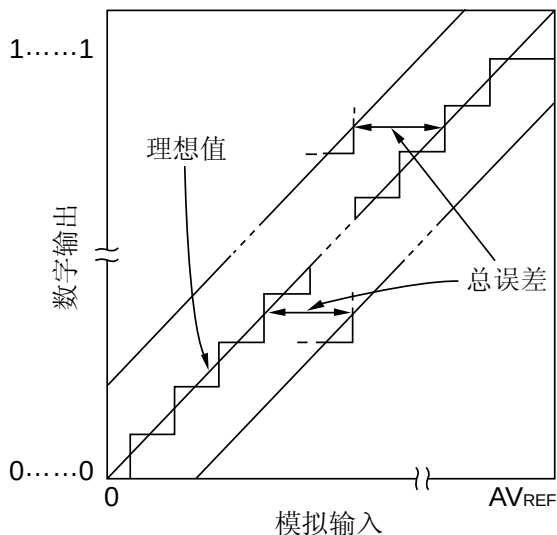
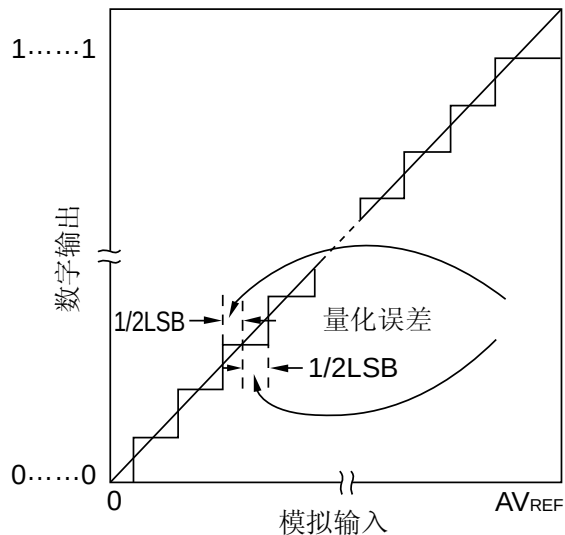


图 10-14. 量化误差



(4) 零度误差

零度误差表示当数字输出范围在 0...000 ~ 0...001 之间时，模拟输入电压的实际测量值和理论值（ $1/2\text{LSB}$ ）之间的误差。

如果实际测量值大于理论值，它表示为当数字输出范围在 0...001 ~ 0...010 之间时，模拟输入电压的实际测量值和理论值（ $3/2\text{LSB}$ ）之间的误差。

(5) 满量程误差

满量程误差表示当数字输出范围在 $1...110 \sim 1...111$ 时，模拟输入电压的实际测量值和理论值（满量程 - $3/2\text{LSB}$ ）之间的误差。

(6) 积分线性误差

积分线性误差说明了转换特征偏离理想线性关系的程度。它表示当零度误差和满量程误差为 0 时，实际测量值和理想直线之间的偏差最大值。

(7) 微分线性误差

当代码输出的理想宽度为 1LSB 时，微分线性误差表示实际测量值和理论值之间的误差。

图 10-15. 零度误差

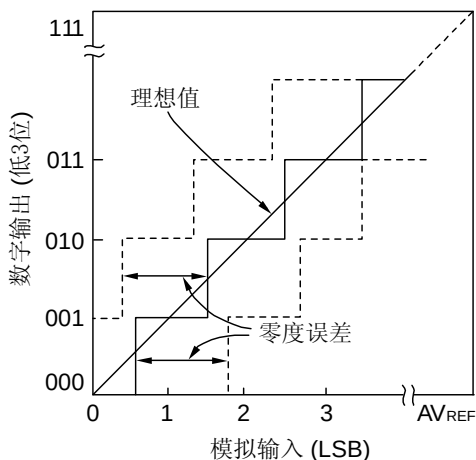


图 10-16. 满量程误差

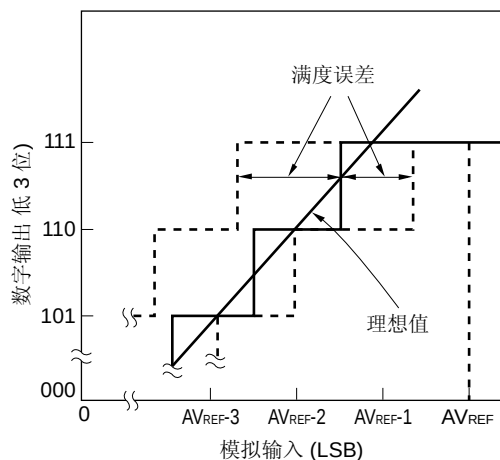


图 10-17. 积分线性误差

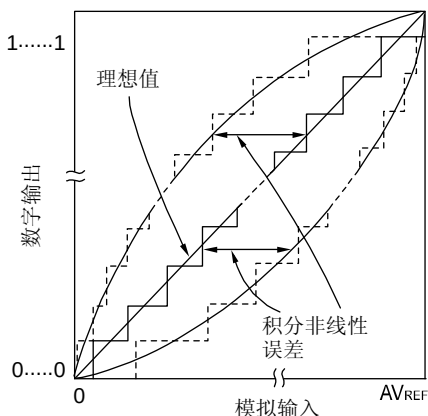
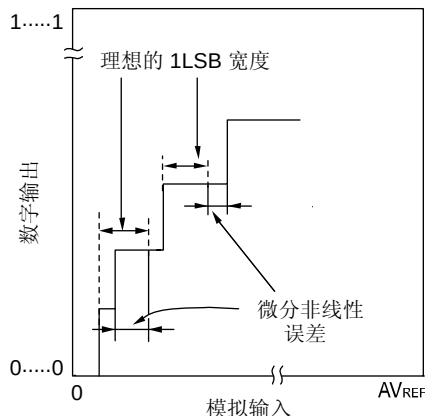


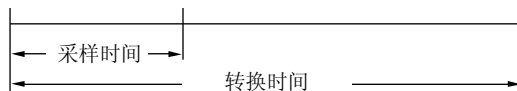
图 10-18. 微分线性误差

**(8) 转换时间**

转换时间表示开始采样到获得数字输出所经历的时间。
特征表中的转换时间包括采样时间。

(9) 采样时间

采样时间表示通过采样&保持电路采样到的模拟电压的进行模拟转换所需的时间。



10.6 A/D转换器的注意事项

(1) STOP 模式下的工作电流

为了满足 STOP 模式下的供电电流 DC 特性，在执行 STOP 指令之前，将 A/D 转换模式寄存器（ADM）的第 7 位（ADCS）和第 0 位（ADCE）清 0。

(2) ANI0 ~ ANI3 的输入范围

观察 ANI0 至 ANI3 的输入电压的额定范围。如果输入到模拟输入通道的电压为 AV_{REF} 或更高，或者为 AV_{SS} 或更低（即使在额定的最大范围内），则该通道的转换值是不确定的。此外，其他通道的转换值也可能受影响。

(3) 冲突操作

<1> 转换结束后，对 A/D 转换结果寄存器（ADCR，ADCRH）的写操作指令和 ADCR，ADCRH 的读操作指令同时进行，就会发生冲突。

ADCR，ADCRH 读操作优先。在读操作执行完成后，将新的转换结果写入 ADCR、ADCRH。

<2> 转换结束后，对 ADCR，ADCRH 写操作和 A/D 转换模式寄存器（ADM）写操作同时进行，或者和模拟输入通道选择寄存器（ADS）写操作同时进行，就会发生冲突。

ADM 或 ADS 写操作优先。ADCR，ADCRH 的写操作不执行，也不会产生转换结束中断信号（INTAD）。

(4) 噪声解决办法

为了保持 10 位分辨率，必须注意输入到 AV_{REF} 引脚和 ANI0~ANI3 引脚的噪声。

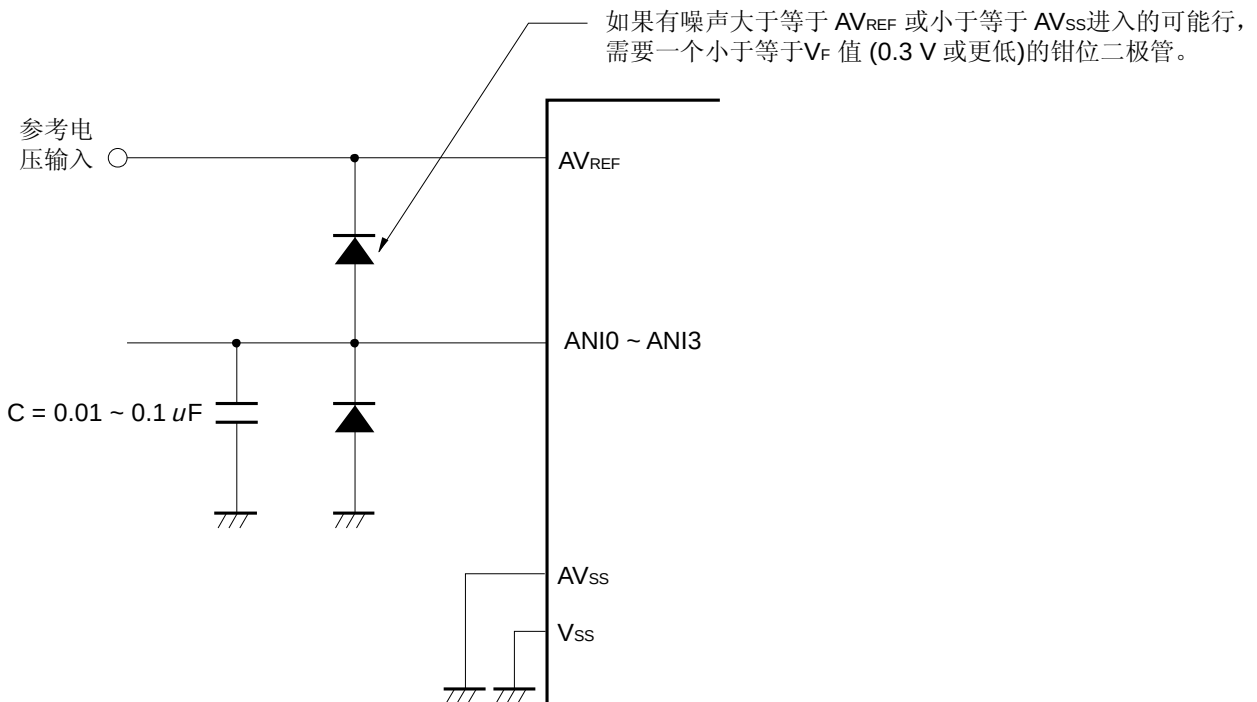
<1> 在电源上连接一个电容，该电容应该是低阻抗和高频响应。

<2> 由于噪声的影响和模拟输入源的输出阻抗成正比，因此建议外接一个电容，如图 10-19 所示，以降低噪声。

<3> 在转换期间，不要将引脚 ANI0~ANI3 的 A/D 转换功能切换到它们的复用功能。

<4> 在转换开始后，立即设置 HALT 模式，可以提高转换精度。

图 10-19. 模拟输入引脚连接



(5) ANI0/P20 ~ ANI3/P23

<1> 模拟输入引脚 (ANI0 ~ ANI3) 也可作为 I/O 端口引脚 (P20 ~ P23)。

当选择 ANI0~ANI3 进行 A/D 转换时, 在转换过程中不要访问 P20~P23, 否则会降低转换精度。

<2> 如果有数字脉冲输入正在进行 A/D 转换的引脚的相邻引脚, 则由于噪声耦合, 有可能得不到预期的 A/D 转换值。因此, 在进行 A/D 转换时不要在相邻引脚输入脉冲信号。

(6) ANI0 ~ ANI3 引脚的输入阻抗

该 A/D 转换器中, 采样期间内对内部采样电容充电并采样。

由于在采样期间之外, 仅有漏电流流过, 在采样期间还有电容的充电电流流过, 因此在采样期间或其他时候输入阻抗都有波动。

如果使用参考电压的最短转换时间, 要进行充足的采样。建议模拟输入源的输出阻抗为 $1\text{k}\Omega$ 或更小, 或者在 ANI0 ~ ANI3 引脚连接 $0.01\mu\text{F}$ 至 $0.1\mu\text{F}$ 的电容器 (见 图 10-19)。

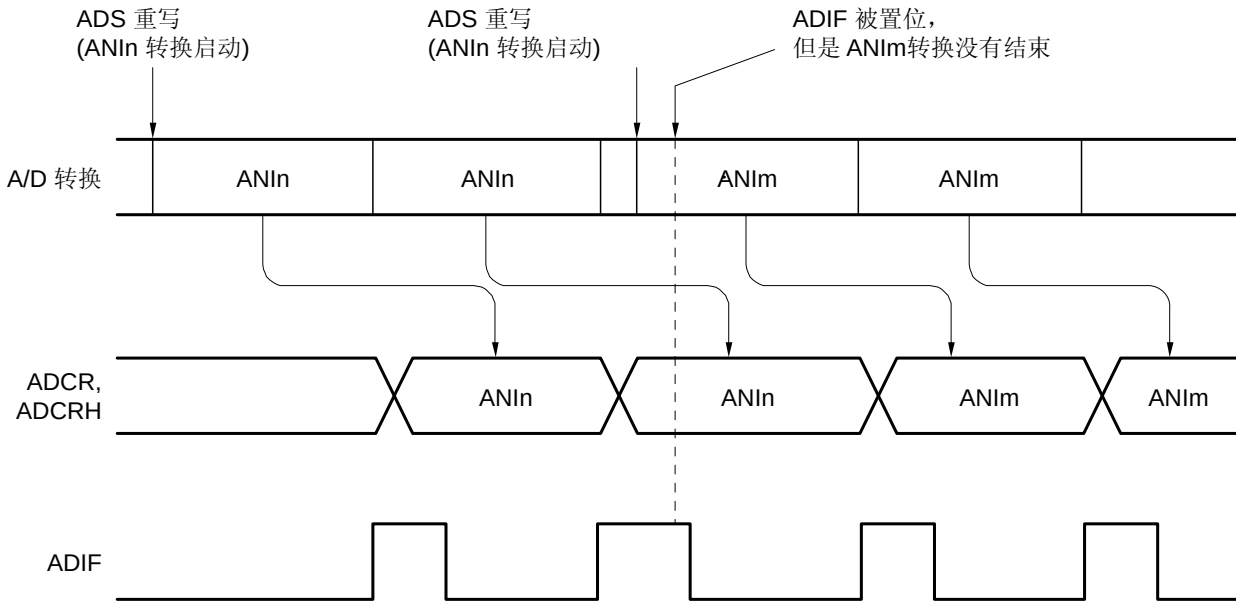
(7) 中断请求信号 (ADIF)

即使模拟输入通道选择寄存器 (ADS) 的值被修改, 中断请求信号 (ADIF) 也不能清零。

因此, 如果在 A/D 转换期间改变模拟输入引脚, 那么在重写 ADS 前, A/D 转换的结果和模拟输入改变前的 ADIF 可能被置 1。所以需要注意, 在 ADS 重写后立即读取 ADIF 时, 尽管模拟输入改变后的 A/D 转换并未结束, 也会设置 ADIF。

在 A/D 转换停止后又重新开始转换时, 要在重新开始转换前先将 ADIF 清 0。

图 10-20. A/D 转换结束中断请求产生的时序



备注 1. $n = 0 \sim 3$
2. $m = 0 \sim 3$

(8) A/D 转换刚开始的转换结果

A/D 转换开始后，如果在 ADCE 置 1 后 $1\mu\text{s}$ 内对 ADCS 置 1，或在 ADCE 为零时对 ADCS 置 1，那么第一次 A/D 转换的值可能不在额定范围内。可以采取的措施，比如查询 A/D 转换结束中断请求（INTAD），并删除第 1 次转换结果。

(9) A/D 转换结果寄存器（ADCR，ADCRH）的读取操作

当对 A/D 转换器模式寄存器（ADM）和模拟输入通道选择寄存器（ADS）进行读取时，ADCR 和 ADCRH 的内容可能变得不确定。要在 ADM 和 ADS 进行写操作之前，且在转换结束之后，读取转换结果。若采用的时序和上述情况不同，可能会读到不正确的转换结果。

(10) 转换等待模式下的工作电流

STOP 模式下工作电流的 DC 特性不能满足，因为当 A/D 转换器模式寄存器（ADM）的第 7 位（ADCS）和第 0 位（ADCE）分别设置为 0 和 1 时，转换等待模式（仅有比较器在耗电）。

(11) 内部等效电路

模拟输入模块的等效电路如下所示。

图 10-21 ANIn 引脚的内部等效电路

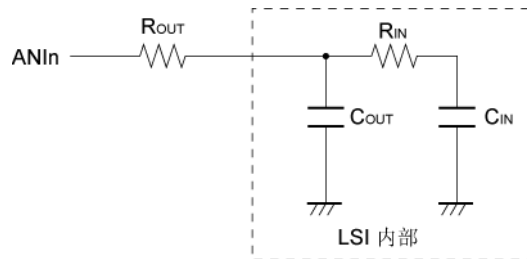


表 10-3. 等效电路的电阻和电容值（参考值）

AV_{REF}	R_{OUT}	R_{IN}	C_{OUT}	C_{IN}
$4.5\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$	$1\text{ k}\Omega$	$3\text{ k}\Omega$	8 pF	15 pF
$2.7\text{ V} \leq AV_{REF} < 4.5\text{ V}$	$1\text{ k}\Omega$	$60\text{ k}\Omega$	8 pF	15 pF

- 备注**
- 表 10-3 所显示的电阻和电容值仅为参考值
 - $n = 0 \sim 3$
 - R_{OUT} : 允许范围的信号源阻抗
 R_{IN} : 模拟输入等效电阻
 C_{IN} : 模拟输入等效电容
 C_{OUT} : 内部引脚电容

11.1 串行接口UART6 的功能

串行接口 UART6 有以下两种模式。

(1) 操作停止模式

在不进行串行通信时，可采用此模式以降低功率消耗。

详细信息，请参见 **11.4.1 操作停止模式**。

(2) 异步串行接口 (UART) 模式

该模式支持 LIN（局部互联网）总线。该模式的功能概括如下。

详细信息，请参见 **11.4.2 异步串行接口 (UART) 模式**和 **11.4.3 专用波特率发生器**。

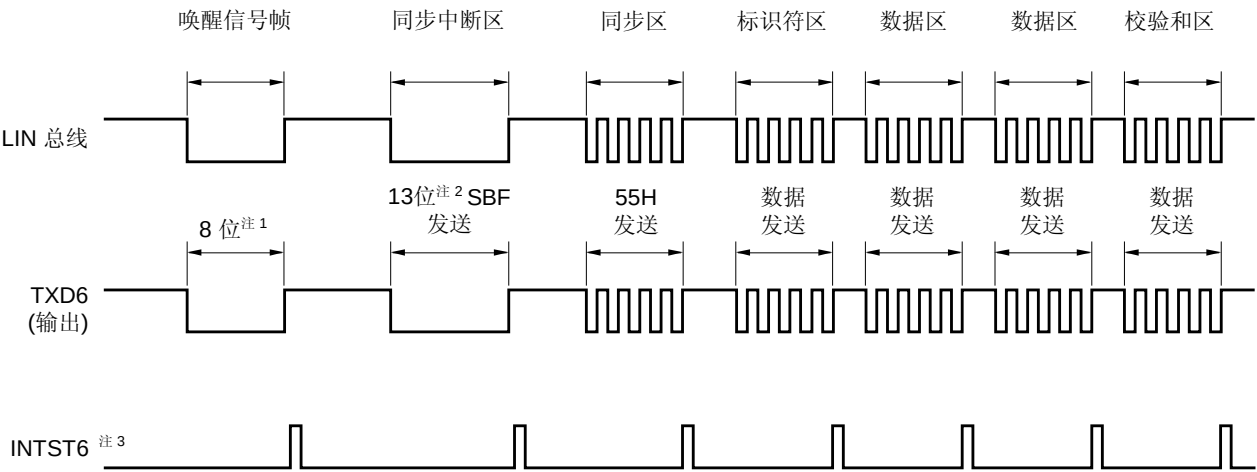
- 双引脚配置 TxD6: 发送数据输出引脚
 RxD6: 接收数据输入引脚
- 可选择通信的数据宽度为 7 或 8 位。
- 专用内置 8 位 波特率发生器，允许设置任意大小的波特率。
- 发送和接收操作可以独立进行。
- 可选 MSB-先行 或 LSB-先行通信。
- 反向传送操作。
- 同步中断域发送范围为 13 ~ 20 位。
- 同步中断区域接收可以识别 11 位以上（提供 SBF 接收标志位）。

- 注意事项**
1. TxD6 输出反向功能的操作仅在发送端进行，而不会在接收端进行。如要使用这个功能，接收端应准备好接收已反向的数据。
 2. 如果串行接口 UART6 使用的时钟未停止（例如，HALT 模式下），则正常操作继续。如果时钟已停止（例如，STOP 模式下），则每个寄存器都停止操作，并且保持时钟停止之前的数值。TxD6 引脚也保持时钟停止之前的数值，并将其输出。但是，在时钟恢复使用后，操作不能保证。因此，对应电路进行复位使 POWER6 = 0, RXE6 = 0, TXE6 = 0。
 3. 如果连续发送数据，从停止位到下一个起始位的通讯时序需要该模块的两个操作时钟周期。但是，这不会影响到通信结果，因为接收端会在检测到起始位时对时序进行初始化。如果该接口用作 LIN 通讯模式，则不能使用连续发送功能。

备注 LIN 表示局部互联网，是一个低速（1 ~ 20 kbps）串行通信协议，可用于降低车载网络的成本。
LIN 通信是一种单主设备通信方式，一个主设备上最多可以连接 15 个从设备。
LIN 的从设备用于控制开关，执行器和传感器，这些设备通过 LIN 网络和 LIN 主设备相连。
通常，LIN 主设备相连到某个网络，比如 CAN（控制器局域网）。
此外，LIN 总线采用单线方式，通过符合 ISO9141 的收发器与各节点相连。
在 LIN 协议中，主设备发送的帧数据带有波特率信息，从设备接收数据并校正波特率误差。因此，当从设备端的波特率误差在±15%范围内，都可以进行通信。

图 11-1 和 11-2 概括了 LIN 的发送和接收操作。

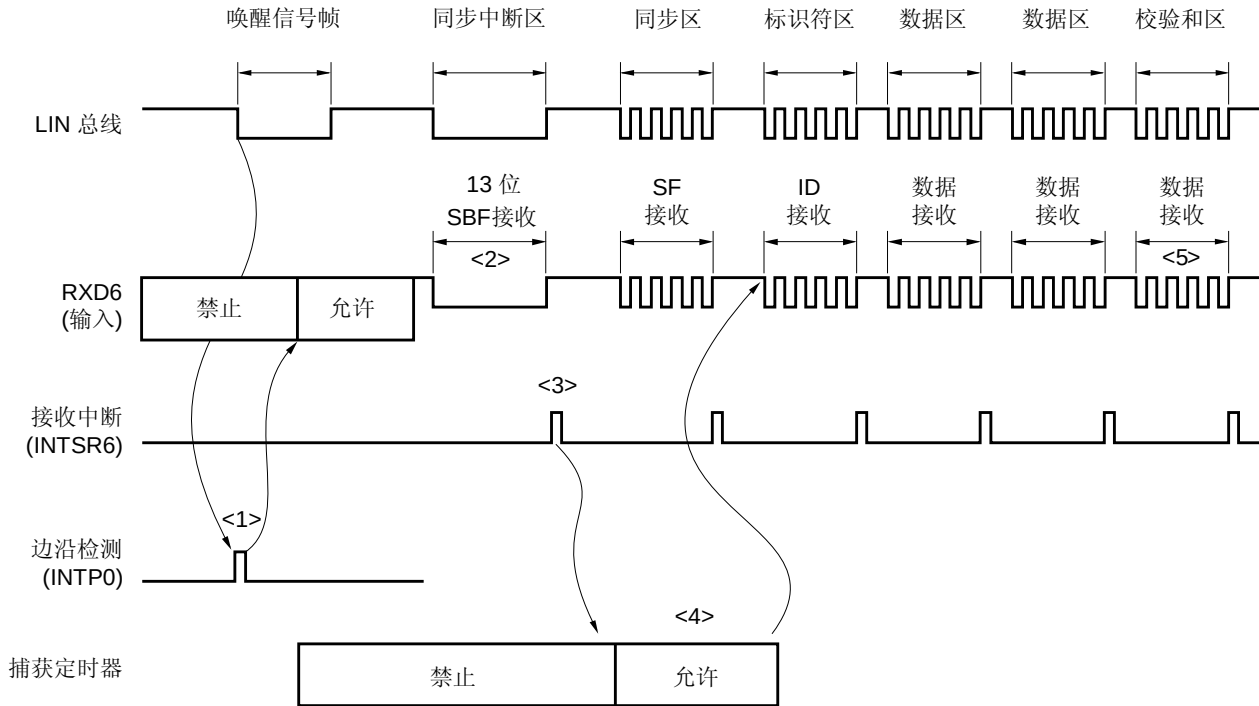
图 11-1. LIN 发送操作



- 注**
- 1. 在 8 位模式中，可通过发送 80H 替代唤醒信号帧。
 - 2. 使用硬件输出同步中断域。输出宽度和异步串行接口控制寄存器 6（ASICL6）的第 4 位~第 2 位（SBL62~SBL60）设置的位宽度相等（参见 11.4.2（2）（h）SBF 发送）。
 - 3. 每次发送完成时，输出 INTST6。在发送 SBF 后，也会输出 INTST6。

备注 每个域之间的间隔由软件控制。

图 11-2. LIN 接收操作



<1> 在引脚的边沿处检测唤醒信号，这样允许使用 UART6，并设置 SBF 接收模式。

<2> 检测到停止位之前，持续接收操作。在检测到具有 11 位或更多位低电平数据的 SBF 时，认为 SBF 接收已正确完成，并输出一个中断信号。在检测到具有少于 11 位低电平数据的 SBF 时，认为出现 SBF 接收错误。不输出中断信号，恢复 SBF 接收模式。

<3> 如果 SBF 被正确接收，将输出一个中断信号。在 SBF 接收完成中断处理中，将启动 16 位定时器/事件计数器 00，并测量同步域的位宽（脉冲宽度）。（参见 6.4.3 脉冲宽度测量操作）。禁止检测 OVE6、PE6 和 FE6 错误，不进行 UART 通信的错误检测过程，也不执行移位寄存器和 RxB6 之间的数据转移。移位寄存器中保持复位后的值 FFH。

<4> 根据同步域的位间隔计算波特率误差，在 SBF 接收后禁止使用 UART6，然后对波特率发生器控制寄存器 6 (BRGC6) 进行复位。

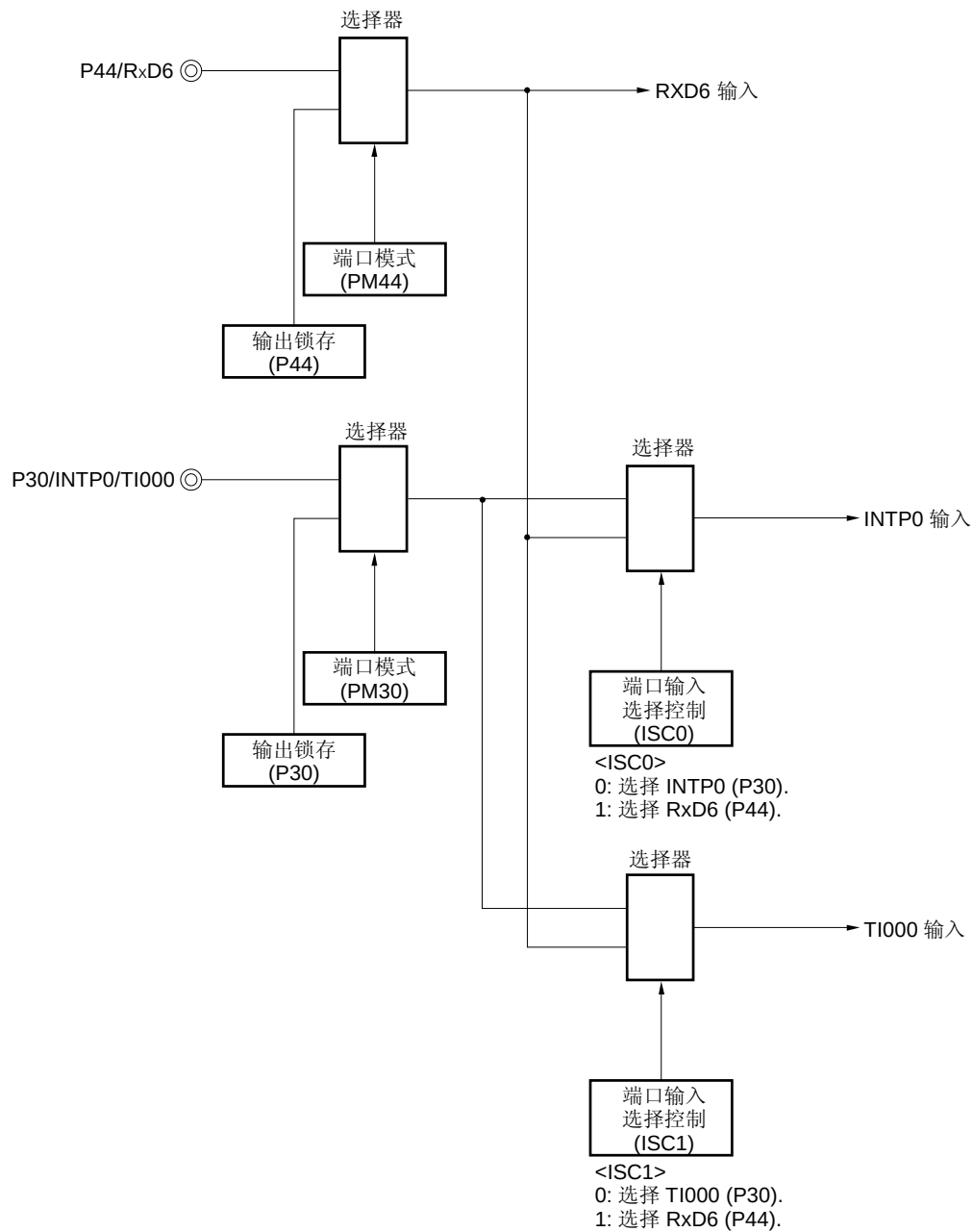
<5> 用软件区分校验和域。在接收到校验和域后，通过软件对 UART6 进行初始化，并再次设置 SBF 接收模式。

图 11-3 显示 LIN 接收操作的端口配置。

通过检测外部中断信号 (INTP0) 脉冲沿，接收主设备发出的唤醒信号。根据 16 位定时器/事件计数器 00 的外部事件捕捉操作来测量主设备发送的同步域长度，由此可以计算波特率误差。

通过端口输入切换控制 (ISC0/ISC1)，不用外部连接 RxD6 和 INTP0/TI000，也可以将接收端口 (RxD6) 的输入信号输入到外部中断 (INTP0) 和 16 位定时器/事件计数器 00。

图 11-3. LIN 接收操作的端口配置



备注 ISC0, ISC1: 输入切换控制寄存器 (ISC) 的第 0 位和第 1 位 (参见图 11-11)。

以下是 LIN 通信操作所使用的外部功能。

<使用的外部功能>

- 外部中断 (INTP0)：唤醒信号检测
用途：检测唤醒信号的脉冲沿，检测通信的开始。
- 16 位定时器/事件计数器 00 (TI000)：波特率误差检测
用途：通过检测同步域 (SF) 的长度并按位的数量分频，来检测波特率误差 (在捕捉模式下，测量 TI000 输入边沿的间隔)。
- 串行接口 UART6

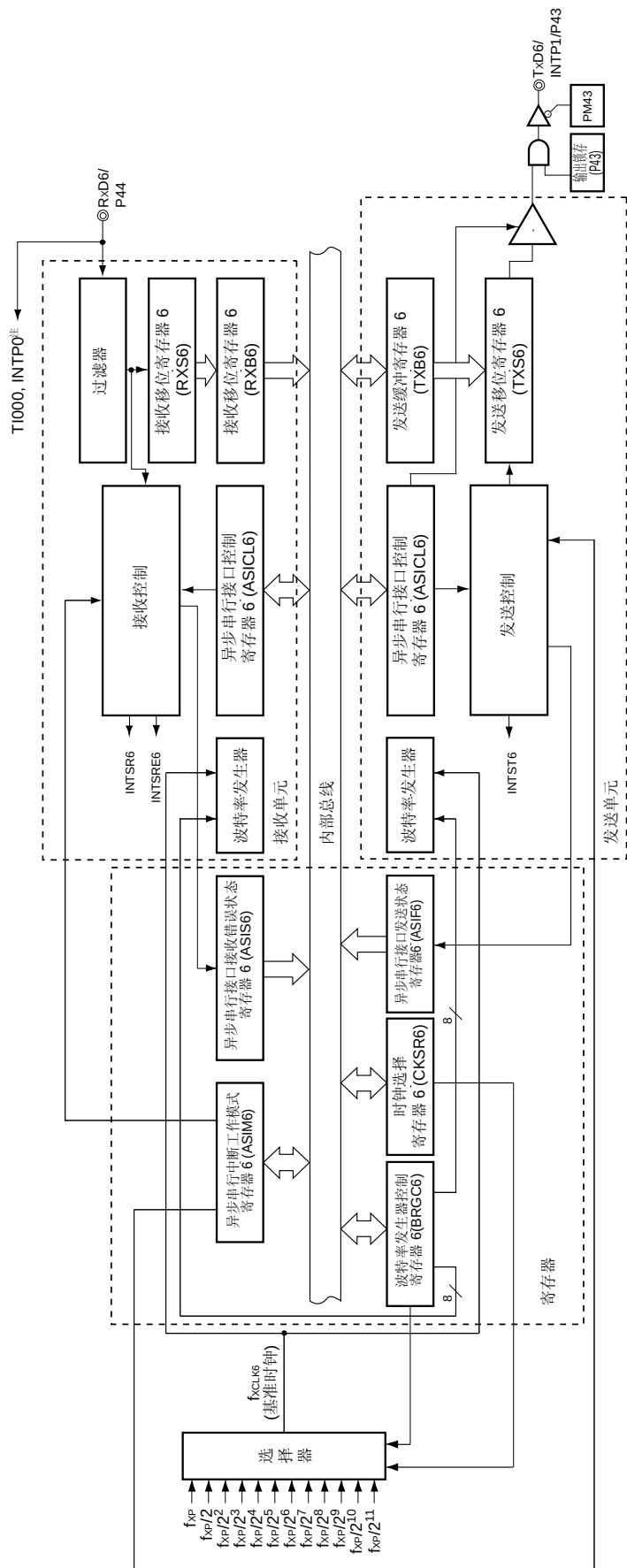
11.2 串行接口UART6 的配置

串行接口 UART6 由以下硬件组成。

表 11-1 串行接口 UART6 的配置

项目	配置
寄存器	接收缓冲寄存器 6（RXB6） 接收移位寄存器 6（RXS6） 发送缓冲寄存器 6（TXB6） 发送移位寄存器 6（TXS6）
控制寄存器	异步串行接口操作模式寄存器 6（ASIM6） 异步串行接口接收错误状态寄存器 6（ASIS6） 异步串行接口发送状态寄存器 6（ASIF6） 时钟选择寄存器 6（CKSR6） 波特率发生器控制寄存器 6（BRGC6） 异步串行接口控制寄存器 6（ASICL6） 输入切换控制寄存器（ISC） 端口模式寄存器 4（PM4） 端口寄存器 4（P4）

图 11-4. 串行接口 UART6 的框图



注 根据输入切换控制寄存器（ISC）进行选择。

(1) 接收缓冲寄存器 6 (RXB6)

该 8 位寄存器用于存储由接收移位寄存器 (RXS6) 转换的并行数据。

每接收到 1 字节的数据, 新的接收数据就从接收移位寄存器 6 (RXS6) 传送到该寄存器中。如果数据宽度被设定为 7 位, 则按如下方式传送数据。

- 在 LSB 先行接收模式中, 将接收到的数据传送到 RXB6 的 0~6 位, RXB6 的 MSB 位恒为 0。
- 在 MSB 先行接收模式中, 将接收到的数据传送到 RXB6 的 7~1 位, RXB6 的 LSB 位恒为 0。

如果出现溢出错误 (OVE6), 则不把接收数据传送到 RXB6 中。

RXB6 的值可由 8 位存储器操作指令读取。不能将数据写入该寄存器中。

复位信号的产生, 会将该寄存器置为 FFH。

注意事项 在将 RXE6 设置为 1, 经过基准时钟 (fXCLK6) 的一个时钟周期后, 进入接收允许状态。

(2) 接收移位寄存器 6 (RXS6)

该寄存器将输入到 RxD6 引脚的串行数据转换成并行数据。

程序不能直接操作 RXS6 寄存器。

(3) 发送缓冲寄存器 6 (TXB6)

该缓冲寄存器用于设置发送数据。当数据写入 TXB6 时, 开始发送。如果数据宽度被设定为 7 位:

- 在 LSB 先行的发送模式中, 数据被传送到 TXB6 的 0~6 位, TXB6 的 MSB 位不发送。
- 在 MSB 先行的发送模式中, 数据被传送到 TXB6 的 7~1 位, TXB6 的 LSB 位不发送。

该寄存器可以通过 8 位存储操作指令进行读或写。

复位信号的产生, 会将该寄存器置为 FFH。

注意事项

1. 当启动发送时, 在 TXE6 设置为 1, 并等待基准时钟 (fxclk6) 一个或多个时钟周期后, 向 TXB6 写入发送数据。
2. 当异步串行接口发送状态寄存器 6 (ASIF6) 的第 1 位 (TXBF6) 为 1 时, 不要向 TXB6 写入数据。
3. 在通信过程中 (当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 和第 6 位 (TXE6) 均为 1, 或 ASIM6 的第 7 位 (POWER6) 和第 5 位 (RXE6) 均为 1 时), 不要通过软件刷新 (写入相同值) TXB6。在连续发送过程中输出相同的值时, 在向 TXB6 写入相同的值之前, 必须确认 TXBF6 为 0。

(4) 发送移位寄存器 6 (TXS6)

该寄存器将 TXB6 中传送来的数据作为串行数据从 TxD6 引脚发出。在第 1 次发送过程中, 当数据写入 TXB6 后立即被传送。或者在持续发送过程中 1 帧数据发送之后, 数据在 INTST6 中断产生之前立即被传送。在基准时钟的下降沿传送来自 TXB6 的数据, 并将其从 TxD6 引脚发出。

程序不能直接操作 TXS6 寄存器。

11.3 串行接口UART6 的控制寄存器

串行接口 UART6 由以下 9 个寄存器控制。

- 异步串行接口操作模式寄存器 6（ASIM6）
- 异步串行接口接收错误状态寄存器 6（ASIS6）
- 异步串行接口发送状态寄存器 6（ASIF6）
- 时钟选择寄存器 6（CKSR6）
- 波特率发生器控制寄存器 6（BRGC6）
- 异步串行接口控制寄存器 6（ASICL6）
- 输入切换控制寄存器（ISC）
- 端口模式寄存器 4（PM4）
- 端口寄存器 4（P4）

(1) 异步串行接口操作模式寄存器 6（ASIM6）

该 8 位寄存器用于控制串行接口 UART6 的串行通信操作。
该寄存器可以通过 1 位或 8 位存储操作指令进行设置。
复位信号的产生，会将该寄存器置为 01H。

备注 在通信过程中（当异步串行接口操作模式寄存器 6（ASIM6）的第 7 位（POWER6）和第 6 位（TXE6）均为 1，或 ASIM6 的第 7 位（POWER6）和第 5 位（RXE6）均为 1 时），可以通过软件刷新（写入相同的值）ASIM6。

图 11-5. 异步串行接口操作模式寄存器 6（ASIM6）的格式（1/2）

地址： FF90H 复位后： 01H 读/写

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	允许/禁止内部操作时钟的操作
0 ^{注1}	禁止内部操作时钟（时钟恒为低电平）的操作及对内部电路进行异步复位 ^{注2}
1 ^{注3}	允许内部操作时钟的操作

- 注**
1. 在发送期间，当 POWER6 清 0 时，TxD6 引脚的输出变为高电平，而 RxD6 引脚的输入恒为高电平。
 2. 异步串行接口接收错误状态寄存器 6（ASIS6）、异步串行接口发送状态寄存器 6（ASIF6）、异步串行接口控制寄存器（ASICL6）的第 7 位（SBRF6）和第 6 位（SBRT6）和接收缓冲寄存器 6（RXB6）都被复位。
 3. 在 POWER6 置 1 时，在经过基准时钟（f_{CLK6}）的一个时钟周期之后，基准时钟作为内部操作时钟提供。

图 11-5. 异步串行接口操作模式寄存器 6（ASIM6）的格式（2/2）

TXE6 ^{注1}	允许/禁止发送
0	禁止发送（同步复位发送电路）
1	允许发送

RXE6 ^{注2}	允许/禁止接收
0	禁止接收（同步复位接收电路）
1	允许接收

PS61	PS60	发送操作	接收操作
0	0	不输出校验位	接收时无校验
0	1	输出零校验	按零校验接收 ^{注3}
1	0	输出奇校验	按奇校验判断
1	1	输出偶校验	按偶校验判断

CL6	指定发送/接收数据的字符宽度
0	数据的字符宽度 = 7 位
1	数据的字符宽度 = 8 位

SL6	指定发送数据停止位的个数
0	停止位的个数 = 1
1	停止位的个数 = 2

ISRM6	允许/禁止出错时产生接收完成中断
0	出错时产生“INTSRE6”（此时，不产生 INTSR6）
1	出错时产生“INTSR6”（此时，不产生 INTSRE6）

- 注
1. 由 CKSR6 设置的基准时钟（f_{CLK6}）使 TXE6 同步。当重新允许发送操作时，将 TXE6 设置为 0 后，经过基准时钟（f_{CLK6}）的一个时钟周期，置 TXE6 为 1。如果在基准时钟（f_{CLK6}）的一个时钟周期内 TXE6 置为 1，发送电路可能无法被初始化。
 2. 由 CKSR6 设置的基准时钟（f_{CLK6}）使 RXE6 同步。当重新允许接收操作时，将 RXE6 设置为 0 后，经过基准时钟（f_{CLK6}）的一个时钟周期，置 RXE6 为 1。如果在基准时钟（f_{CLK6}）的一个时钟周期内 RXE6 被置为 1，接收电路可能无法被初始化。
 3. 如果选择“按零校验接收”，则无法判断校验位。因此，异步串行接口接收错误状态寄存器 0（ASIS6）的第 2 位（PE6）不会被置 1，这样也不会产生出错中断。

注意事项 1. 在启动阶段，对 POWER6 置 1 之后，对 TXE6 置 1，启动发送操作，然后再等待基准时钟（f_{CLK6}）的一个或多个时钟周期之后，设置发送数据到 TXB6。停止发送操作时，先将 TXE6 清零，然后将 POWER6 清零。

- 注意事项
- 2. 在启动阶段，对 **POWER6** 置 1 之后，然后对 **RXE6** 置 1，经过基准时钟（f_{XCLK6}）的一个时钟周期，进入接收允许状态。停止接收操作时，先将 **RXE6** 清零，然后将 **POWER6** 清零。
 - 3. 当 **RxD6** 引脚输入高电平的状态时，设置 **POWER6** = 1 → **RXE6** = 1。如果在低电平输入期间设置 **POWER6** = 1 → **RXE6** = 1，启动接收，不能接收到正确的数据。
 - 4. 在对 **PS61**、**PS60** 和 **CL6** 位重写之前，先将 **TXE6** 和 **RXE6** 位清零。
 - 5. 当该接口用作 **LIN** 通讯操作模式时，**PS61** 与 **PS60** 恒为 0。
 - 6. 在重写 **SL6** 位时，必须确保 **TXE6** = 0。当“停止位的个数 = 1”时，始终执行接收操作，因此设定 **SL6** 的值不会影响接收操作。
 - 7. 当重写 **ISRM6** 时，确保 **RXE6** = 0。

(2) 异步串行接口接收错误状态寄存器 6（ASIS6）

该寄存器用于指示串行接口 UART6 接收操作完成时的错误状态。 包括 3 个出错标志位（**PE6**, **FE6**, **OVE6**）。该寄存器可由 8 位存储器操作指令对其进行只读操作。

如果 **ASIM6** 的第 7 位（**POWER6**）和第 5 位（**RXE6**）均为 0，则复位信号的产生会将该寄存器清为 00H。读取该寄存器时，读得 00H。

图 11-6. 异步串行接口接收错误状态寄存器 6（ASIS6）的格式

地址： FF93H 复位后： 00H 读

符号	7	6	5	4	3	2	1	0
ASIS6	0	0	0	0	0	PE6	FE6	OVE6

PE6	指示校验错误的状态标志位
0	若 POWER6 = 0 且 RXE6 = 0、或读取 ASIS6 寄存器
1	如果发送数据的校验位和接收完成时的校验位不匹配

FE6	指示帧错误的状态标志位
0	若 POWER6 = 0 且 RXE6 = 0、或读取 ASIS6 寄存器
1	如果在接收完成时未检测到停止位

OVE6	指示溢出错误的状态标志位
0	若 POWER6 = 0 且 RXE6 = 0、或读取 ASIS6 寄存器
1	如果将接收数据存于 RXB 寄存器中，且在读取该数据前已经完成下一次接收操作

- 注意事项
- 1. 根据异步串行接口操作模式寄存器 6（**ASIM6**）的 **PS61** 和 **PS60** 位的内容的不同，对 **PE6** 位的操作也有所不同。
 - 2. 无论停止位的个数是多少，只能对接收数据的第 1 位停止位进行校验。
 - 3. 如果出现溢出错误，则下一个接收数据不是被写入接收缓冲寄存器 6（**RXB6**）中，而是被抛弃。
 - 4. 读取接收缓冲寄存器 6（**RXB6**）之前，必须读取 **ASIS6**。

(3) 异步串行接口发送状态寄存器 6 (ASIF6)

该寄存器指示串行接口 UART6 的发送状态。包括两个状态标志位 (TXBF6 和 TXSF6)。

当数据从 TXB6 传送到 TXS6 后，将下一个发送数据写入 TXB6，这样使得发送能够连续进行而不被打断，即使在中断产生期间也不受影响。

该寄存器可由 8 位存储器操作指令对其进行只读操作。

如果 ASIM6 的第 7 位 (POWER6) 和第 6 位 (TXE6) =0，复位信号的产生，会将该寄存器清为 00H。

图 11-7. 异步串行接口发送状态寄存器 6 (ASIF6) 的格式

地址: FF95H 复位后: 00H 读

符号	7	6	5	4	3	2	1	0
ASIF6	0	0	0	0	0	0	TXBF6	TXSF6

TXBF6	发送缓冲数据标志位
0	如果 POWER6 = 0 或 TXE6 = 0，或如果数据被传送到发送移位寄存器 6 (TXS6) 中
1	如果数据写入发送缓冲寄存器 6 (TXB6) 中 (如果有数据存在于 TXB6 中)

TXSF6	发送移位寄存器数据标志位
0	如果 POWER6 = 0 或 TXE6 = 0，或如果发送完成后下一个数据没有从发送缓冲寄存器 6 (TXB6) 发出
1	如果数据从发送缓冲寄存器 6 (TXB6) 发出 (如果数据仍在发送过程中)

- 注意事项
- 1. 如果要连续发送数据，将第一个发送的数据 (第一个字节) 写入 TXB6 寄存器中。检查 TXBF6 标志位，必须为“0”。这样的话，将下一个发送数据 (第二个字节) 写入 TXB6 中。如果在 TXBF6 标志为“1”时将数据写入 TXB6，则无法保证发送数据的正确性。
 - 2. 若要在连续发送结束时初始化发送单元，则在产生发送完成中断后，检查 TXSF6 标志位，确保为“0”，然后执行初始化操作。如果在 TXSF6 标志为“1”时执行初始化操作，则无法保证发送数据的正确性。

(4) 时钟选择寄存器 6 (CKSR6)

该寄存器用于选择串行接口 UART6 的基准时钟。

CKSR6 可由 8 位存储器操作指令设置。

复位信号的产生，会将该寄存器清为 00H。

备注 通信过程中（当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 和第 6 位 (TXE6) 均为 1，或 ASIM6 的第 7 位 (POWER6) 和第 5 位 (RXE6) 均为 1 时），CKSR6 可由软件刷新（写入相同的值）。

图 11-8. 时钟选择寄存器 6 (CKSR6) 的格式

地址： FF96H 复位后： 00H 读/写

符号	7	6	5	4	3	2	1	0
CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60

TPS63	TPS62	TPS61	TPS60	基准时钟 (f _{CLK6}) 选择
0	0	0	0	f _{XP} (10 MHz)
0	0	0	1	f _{XP} /2 (5 MHz)
0	0	1	0	f _{XP} /2 ² (2.5 MHz)
0	0	1	1	f _{XP} /2 ³ (1.25 MHz)
0	1	0	0	f _{XP} /2 ⁴ (625 kHz)
0	1	0	1	f _{XP} /2 ⁵ (312.5 kHz)
0	1	1	0	f _{XP} /2 ⁶ (156.25 kHz)
0	1	1	1	f _{XP} /2 ⁷ (78.13 kHz)
1	0	0	0	f _{XP} /2 ⁸ (39.06 kHz)
1	0	0	1	f _{XP} /2 ⁹ (19.53 kHz)
1	0	1	0	f _{XP} /2 ¹⁰ (9.77 kHz)
1	0	1	1	f _{XP} /2 ¹¹ (4.89 kHz)
其他状态				禁止设置

注意事项 当重写 TPS63 ~ TPS60 时，必须确保 POWER6 = 0。

备注 1. 括号中的值适用于操作频率 f_{XP} = 10 MHz 。
2. f_{XP}: 外围硬件时钟的振荡频率

(5) 波特率发生器控制寄存器 6 (BRGC6)

该寄存器用于设置串行接口 UART6 的 8 位计数器的分频值。

BRGC6 可由 8 位存储器操作指令设置。

复位信号的产生，会将该寄存器置为 FFH。

备注 通信过程中（当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 和第 6 位 (TXE6) 均为 1，或 ASIM6 的第 7 位 (POWER6) 和第 5 位 (RXE6) 均为 1 时），BRGC6 可由软件刷新（写入相同的值）。

图 11-9. 波特率发生器控制寄存器 6 (BRGC6) 的格式

地址： FF97H 复位后： FFH 读/写

符号	7	6	5	4	3	2	1	0
BRGC6	MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	k	8 位计数器输出时钟选择
0	0	0	0	0	×	×	×	×	禁止设置
0	0	0	0	1	0	0	0	8	$f_{XCLK6}/8$
0	0	0	0	1	0	0	1	9	$f_{XCLK6}/9$
0	0	0	0	1	0	1	0	10	$f_{XCLK6}/10$
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	$f_{XCLK6}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK6}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK6}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK6}/255$

- 注意事项**
1. 在重写 MDL67 ~ MDL60 位时，必须确保 ASIM6 的第 6 位 (TXE6) 和第 5 位 (RXE6) = 0。
 2. 波特率为 8 位计数器输出时钟的 1/2。

- 备注**
1. f_{XCLK6} : 根据 CKSR6 寄存器的 TPS63~TPS60 位选择的基准时钟频率
 2. k: MDL67 ~ MDL60 位设置的值 (k = 8, 9, 10, ..., 255)
 3. × 无需考虑

(6) 异步串行接口控制寄存器 6 (ASICL6)

该寄存器用于控制串行接口 UART6 的串行通信操作。

ASICL6 可由 1 位或 8 位存储器操作指令设置。

复位信号的产生，会将该寄存器置为 16H。

注意事项 如果数据 0 通过 SBRT6 和 SBTT6 写入到 ASICL6，在通信操作期间（当 ASIM6 的第 7 位（POWER6）和第 6 位（TXE6）为 1 或 ASIM6 的第 7 位（POWER6）和第 5 位（RXE6）为 1 时），ASICL6 可以通过软件更新（写入相同的值）。

图 11-10. 异步串行接口控制寄存器 6 (ASICL6) 的格式 (1/2)

地址： FF98H 复位后： 16H 读/写[※]

符号

<7>

<6>

5

4

3

2

1

0

ASICL6

SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TxDLV6
-------	-------	-------	-------	-------	-------	------	--------

SBRF6	SBF 接收状态标志
0	如果 POWER6 = 0 且 RXE6 = 0，或者如果 SBF 接收已正确完成
1	SBF 接收正在进行中

SBRT6	SBF 接收触发
0	—
1	SBF 接收触发

SBTT6	SBF 发送触发
0	—
1	SBF 发送触发

注 第 7 位是只读的。

图 11-10. 异步串行接口控制寄存器 6（ASICL6）的格式（2/2）

SBL62	SBL61	SBL60	SBF 发送输出宽度控制
1	0	1	SBF 输出是 13 位长度。
1	1	0	SBF 输出是 14 位长度。
1	1	1	SBF 输出是 15 位长度。
0	0	0	SBF 输出是 16 位长度。
0	0	1	SBF 输出是 17 位长度。
0	1	0	SBF 输出是 18 位长度。
0	1	1	SBF 输出是 19 位长度。
1	0	0	SBF 输出是 20 位长度。

DIR6	起始位规范
0	MSB
1	LSB

TxDLV6	允许/禁止 TxD6 输出反向
0	TxD6 普通输出
1	TxD6 反向输出

- 注意事项
1. 在 SBF 接收错误情况下，则再次返回到 SBF 接收模式下。SBRF6 标志位的状态保持不变（1）。SBF 接收的详细情况，请参见在稍后的 11.4.2 异步串行接口（UART）模式中的（2） - （i）SBF 接收。
 2. 在设置 SBRT6 位为 1 之前，必须确保 ASIM6 的第 7 位（POWER6）和第 5 位（RXE6）为 1。并且，在设置 SBRT6 为 1 之后，不要在 SBF 接收结束（产生中断请求信号）之前，对 SBRT6 位清零。
 3. SBRT6 位的读取值恒为 0。在 SBF 接收正确完成后，SBRT6 自动清零。
 4. 在设置 SBTT6 位为 1 之前，必须确保 ASIM6 的第 7 位（POWER6）和第 6 位（TXE6）为 1。并且，在设置 SBTT6 为 1 之后，不要在 SBF 发送结束（产生中断请求信号）之前，对 SBTT6 位清零。
 5. SBTT6 位的读取值恒为 0。在 SBF 发送结束时 SBRT6 自动清零。
 6. 在重写 DIR6 和 TxDLV6 位之前，将 TXE6 和 RXE6 清零。

(7) 输入切换控制寄存器 (ISC)

输入切换控制寄存器 (ISC) 用于在 LIN (本地内联网络) 接收期间接收主设备发出的状态信号。
通过对 ISC0 和 ISC1 置 1, INTP0 和 TI000 的输入源切换到 P44/RxD6 引脚的输入信号。
该寄存器可以通过 1 位或 8 位存储操作指令进行设置。
复位信号的产生, 会将该寄存器清为 00H。

图 11-11 输入切换控制寄存器 (ISC) 的格式

地址: FF8CH 复位后: 00H 读/写

符号	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	TI000 输入源选择
0	TI000 (P30)
1	RxD6 (P44)

ISC0	INTP0 输入源选择
0	INTP0 (P30)
1	RxD6 (P44)

(8) 端口模式寄存器 4 (PM4)

该寄存器可以逐位设置端口 4 的输入输出模式。
当 P43/TxD6/INTP1 引脚用于串行接口数据输出时, 将 PM43 清零, 且 P43 的输出锁存置 1。
当 P44/RxD6 引脚用于串行接口数据输入时, 将 PM44 置 1。此时 P44 的输出锁存可能是 0 或 1。
PM4 可以通过 1 位或 8 位存储操作指令进行设置
复位信号的产生, 会将该寄存器置为 FFH。

图 11-12 端口模式寄存器 4 (PM4) 的格式

地址: FF24H 复位后: FFH 读/写

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

PM4n	P4n 引脚 I/O 模式选择 (n=0~7)
0	输出模式 (输出缓冲打开)
1	输入模式 (输出缓冲关闭)

11.4 串行接口 UART6 的操作

串行接口 UART6 有以下两种模式。

- 操作停止模式
- 异步串行接口（UART）模式

11.4.1 操作停止模式

在该模式下，不进行串行通信，可以降低电能消耗。此外，该模式下这些引脚可作为普通端口引脚使用。要设置操作停止模式，将 ASIM6 的第 7、6 和 5 位（POWER6、TXE6 和 RXE6）清零。

(1) 使用的寄存器

由异步串行接口操作模式寄存器 6（ASIM6）设置操作停止模式。

ASIM6 可由 1 位或 8 位存储器操作指令设置。

复位信号的产生，会将该寄存器置为 01H。

地址： FF90H 复位后： 01H 读/写

符号

	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6
POWER6		允许/禁止内部操作时钟的操作						
0 ^{註1}		禁止内部操作时钟（时钟恒为低电平）的操作，且对内部电路进行异步复位 ^{註2} 。						
TXE6		允许/禁止发送						
0		禁止发送操作（同步复位发送电路）						
RXE6		允许/禁止接收						
0		禁止接收（同步复位接收电路）						

- 注
1. 在发送期间 POWER6 = 0 时，TxD6 引脚的输出变成高电平，而 RxD6 引脚的输入恒为高电平。
 2. 异步串行接口接收错误状态寄存器 6（ASIS6）、异步串行接口发送状态寄存器 6（ASIF6）、异步串行接口控制寄存器（ASICL6）的第 7 位（SBRF6）和第 6 位（SBRT6）和接收缓冲寄存器 6（RXB6）都被复位。

注意事项 TXE6 和 RXE6 清零后，再对 POWER6 清零，可以设置操作停止模式。
要启动操作，设置 POWER6 为 1，然后设置 TXE6 和 RXE6 为 1。

备注 如果要将 RxD6/P44 和 TxD6/INTP1/P43 作为通用端口引脚使用，请参见第四章 端口功能。

11.4.2 异步串行接口（UART）模式

此模式中，在起始位之后进行 1 个字节数据的发送/接收，可实现全双工操作。

内置专用 UART 波特率发生器，可以在较大的波特率范围内进行通信。

(1) 使用的寄存器

- 异步串行接口操作模式寄存器 6（ASIM6）
- 异步串行接口接收错误状态寄存器 6（ASIS6）
- 异步串行接口发送状态寄存器 6（ASIF6）
- 时钟选择寄存器 6（CKSR6）
- 波特率发生器控制寄存器 6（BRGC6）
- 异步串行接口控制寄存器 6（ASICL6）
- 输入切换控制寄存器（ISC）
- 端口模式寄存器 4（PM4）
- 端口寄存器 4（P4）

在 UART 模式中设置基本操作的步骤如下。

- <1> 设置 CKSR6 寄存器（参见图 11-8）。
- <2> 设置 BRGC6 寄存器（参见图 11-9）。
- <3> 设置 ASIM6 寄存器的第 0~4 位（ISRM6, SL6, CL6, PS60, PS61）（参见图 11-5）。
- <4> 设置 ASICL6 寄存器的第 0 位和第 1 位（TxDLV6, DIR6）（参见图 11-10）。
- <5> 设置 ASIM6 寄存器的第 7 位（POWER6）为 1。
- <6> 设置 ASIM6 寄存器的第 6 位（TXE6）为 1 → 允许发送。
设置 ASIM6 寄存器的第 5 位（RXE6）为 1 → 允许接收。
- <7> 将数据写入发送缓冲寄存器 6（TXB6）→ 开始发送数据。

注意事项 在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。为避免产生非预期的起始位（下降信号），在将 P43 置 1 后，设置 PM43 为 0（输出）。

寄存器设置与引脚的关系如下所示。

表 11-2. 寄存器设置与引脚的关系

POWER6	TXE6	RXE6	PM43	P43	PM44	P44	UART6 操作	引脚功能	
								TxD6/INTP1/P43	RxD6/P44
0	0	0	×	×	×	×	停止	P43	P44
1	0	1	×	×	1	×	接收	P43	RxD6
	1	0	0	1	×	×	发送	TxD6	P44
	1	1	0	1	1	×	发送/ 接收	TxD6	RxD6

注 可以被设置为端口功能。

- 备注
- ×

不必考虑
- POWER6: 异步串行接口操作模式寄存器 6（ASIM6）的第 7 位
- TXE6: ASIM6 的第 6 位
- RXE6: ASIM6 的第 5 位
- PM4x: 端口模式寄存器
- P4x: 端口输出锁存

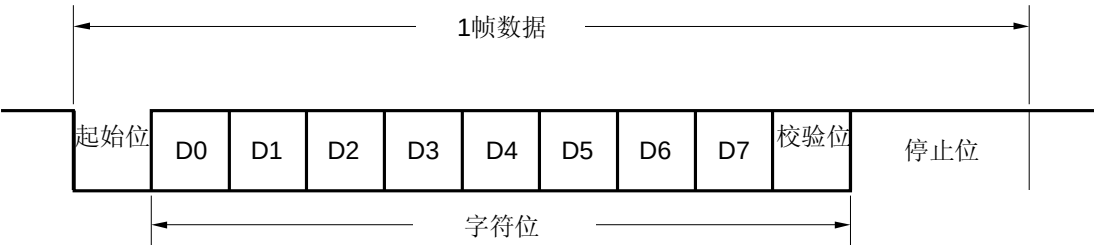
(2) 通信操作

(a) 正常发送/接收数据的格式和波形示例

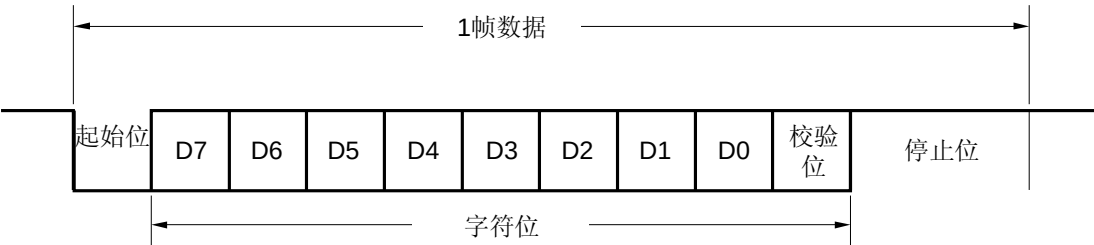
图 11-13 和 11-14 显示了正常发送/接收数据的格式和波形示例。

图 11-13. 正常 UART 发送/接收数据的格式

1. LSB 先行的发送/接收



2. MSB 先行的发送/接收



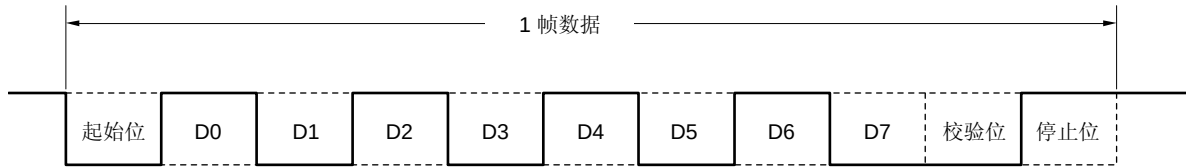
一个数据帧由以下各位组成。

- 起始位 ... 1 位
- 字符位 ... 7 或 8 位
- 校验位 ... 偶校验，奇校验，零校验或无校验
- 停止位 ... 1 或 2 位

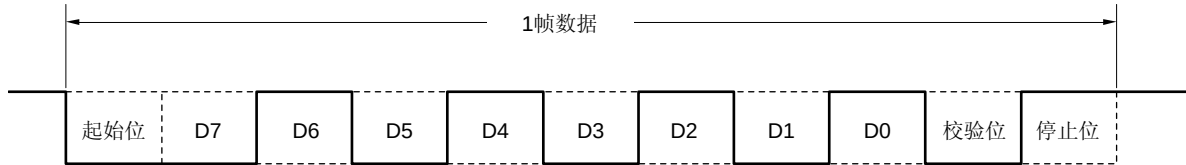
由异步串行接口操作模式寄存器 6 (ASIM6) 指定一个数据帧中字符位的长度、校验方式和停止位的长度。
由异步串行接口控制寄存器 6 (ASICL6) 的第 1 位 (DIR6) 指定数据通信方式是 LSB 先行或 MSB 先行。
由 ASICL6 的第 0 位 (TxDLV6) 指定 TxD6 引脚输出正常数据还是反向数据。

图 11-14. 正常 UART 发送/接收数据波形示例

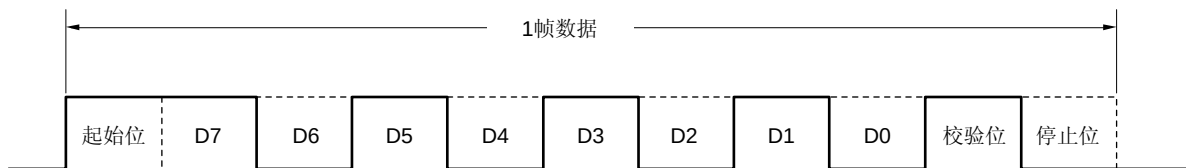
1. 数据长度：8 位, LSB 先行, 校验位：偶校验, 停止位：1 位, 通信数据：55H



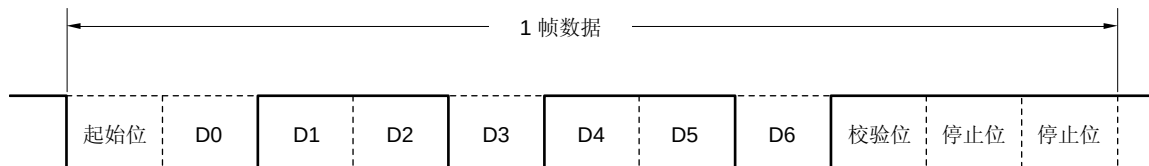
2. 数据长度：8 位, MSB 先行, 校验位：偶校验, 停止位：1 位, 通信数据：55H



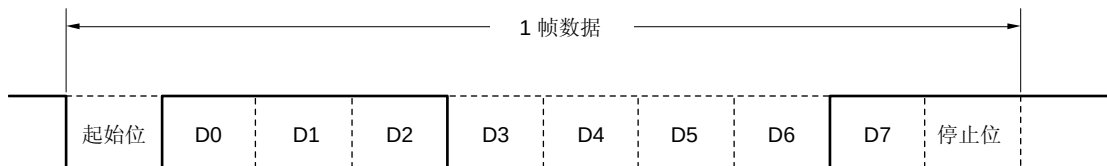
3. 数据长度：8 位, MSB 先行, 校验位：偶校验, 停止位：1 位, 通信数据：55H, TxD6 引脚输出反向数据



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



5. 数据长度：8 位, LSB 先行, 校验位：无, 停止位：1 位, 通信数据：87H



(b) 校验方式与操作

校验位用于检测数据通信中的位错误。通常在发送和接收两端使用相同的校验方式。当采用偶校验和奇校验时，可检测到 1 位（奇数）错误。而采用零校验和无校验时，无法检测到错误。

注意事项 当该接口用作 LIN 通讯操作模式时，PS61 与 PS60 恒为 0。

(i) 偶校验

- 发送
控制发送数据，包括校验位，使得数据中“1”的个数为偶数。
校验位的取值如下。

若发送数据有奇数个“1”： 1
若发送数据有偶数个“1”： 0

- 接收
计算接收数据中“1”的个数，包括校验位。若“1”的个数为奇数，则产生校验错误。

(ii) 奇校验

- 发送
和偶校验不同，控制发送数据，包括校验位，使得数据中“1”的个数为奇数。

如果发送数据有奇数个“1”： 0
如果发送数据有偶数个“1”： 1

- 接收
计算接收数据中“1”的个数，包括校验位。如果“1”的个数为偶数，则产生校验错误。”

(iii) 零校验

发送数据时校验位清零，无论发送数据为何值。
接收数据时，不检测校验位。因此，无论校验位为“0”或“1”，都不会产生校验错误。

(iv) 无校验

发送数据不附加校验位。
进行数据接收操作时，认为没有校验位。因为没有校验位，因此也不会产生校验错误。

(c) 正常发送

当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 为 1 时, 经过基准时钟 (fXCLK6) 的一个时钟周期后, 设置 ASIM6 的第 6 位 (TXE6) 为 1, 则进入发送允许状态。将发送数据写入发送缓冲寄存器 6 (TXB6) 中时, 可启动发送操作。起始位、校验位和停止位会自动附加到数据上。

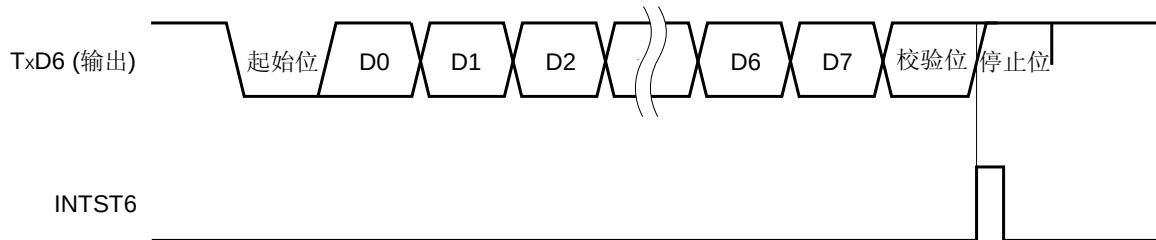
启动发送操作时, TXB6 中的数据传送到发送移位寄存器 6 (TXS6) 中。此后, 将数据依次从 TXS6 输出到 TxD6 引脚。当发送结束时, 由 ASIM6 设置的校验位和停止位附加到数据上, 并产生发送完成中断请求 (INTST6)。

直到下一个发送数据写入 TXB6 时, 发送处于停止。

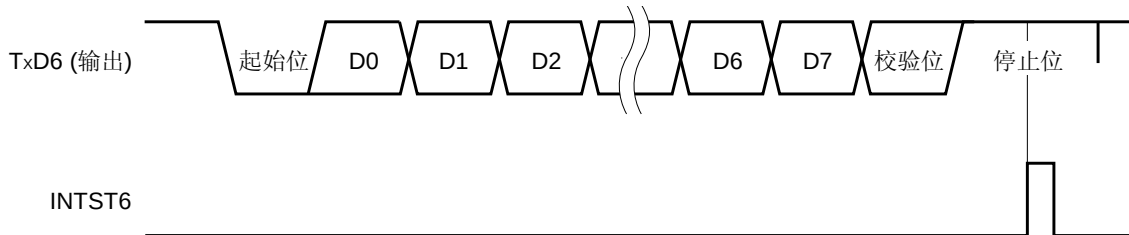
图 11-15 显示了发送完成中断请求 (INTST6) 的时序。最后一个停止位输出时, 立即产生该中断。

图 11-15. 正常发送完成中断请求时序

1. 停止位长度: 1



2. 停止位长度: 2



(d) 连续发送

当发送移位寄存器 6（TXS6）启动移位操作，下一个发送数据立即就可以写入发送缓冲寄存器 6（TXB6）。因此，当发送完一帧数据后，即使在执行 INTST6 中断服务时，数据也能够连续发送，从而实现高效的通信速度。此外，在发送完成中断产生之后，通过确认异步串行接口状态寄存器 6（ASIF6）的第 0 位（TXSF6），可以将下一个发送数据写入 TXB6 寄存器而无须等待一个数据帧的发送时间。

为了能够连续发送数据，必须检查 ASIF6 寄存器来确认发送状态以及是否可以对 TXB6 进行写入操作，然后写入数据。

关于连续发送过程，请参见 图 11-16 连续发送过程流程举例。

- 注意事项**
- 1. 使用 TXBF6 标志的值来判断是否可以连续发送。不要写入下一个发送数据，判断的唯一标准就是 TXSF6 标志已经被置 1。
 - 2. 在该接口用作 LIN 通讯操作时，不能使用连续发送功能。在将发送数据写入发送缓冲寄存器 6（TXB6）之前，异步串行接口发送状态寄存器（ASIF6）的值必须为 00H。

TXBF6	写入 TXB6 寄存器
0	允许写入
1	禁止写入

- 注意事项**
- 若要连续发送数据，将第一个发送的数据（第一个字节）写入 TXB6 寄存器中。检查 TXBF6 标志位，必须为“0”。这样的话，将下一个发送数据（第二个字节）写入 TXB6 中。如果在 TXBF6 标志为“1”时将数据写入 TXB6，则无法保证发送数据的正确性。

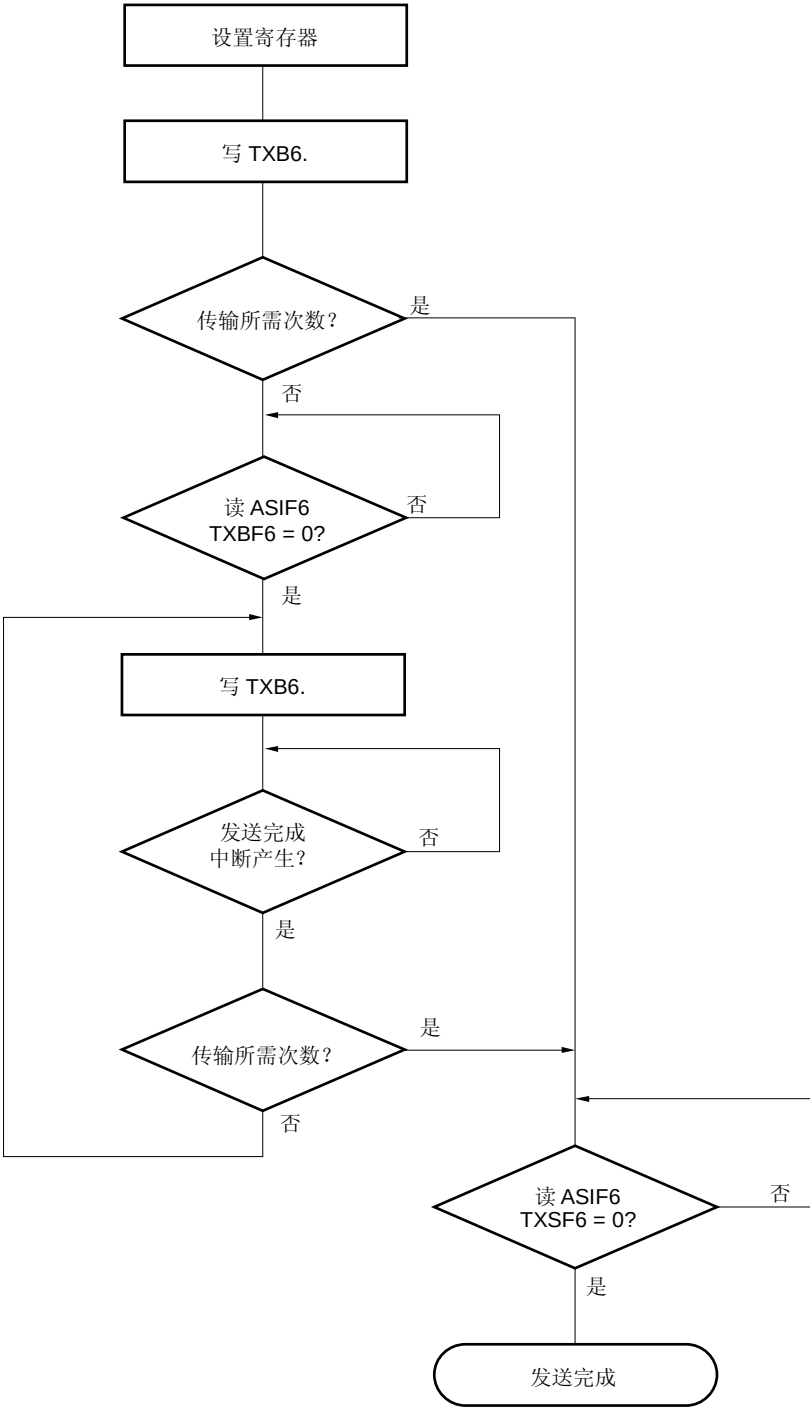
可以用 TXSF6 标志位来检查通信状态。

TXSF6	发送状态
0	发送完成
1	发送正在进行中

- 注意事项**
- 若要在连续发送结束时初始化发送单元，则在产生发送完成中断后，检查 TXSF6 标志位，确保为“0”，然后执行初始化操作。如果在 TXSF6 标志为“1”时执行初始化操作，则无法保证发送数据的正确性。

图 11-16 显示了连续发送操作流程示例。

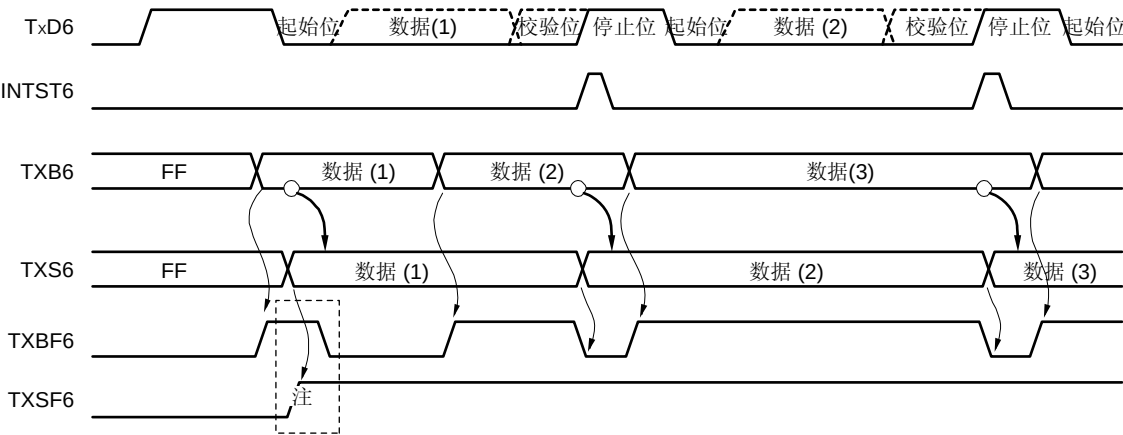
图 11-16. 连续发送操作流程示例



备注 TXB6: 发送缓冲寄存器 6
ASIF6: 异步串行接口发送状态寄存器 6
TXBF6: ASIF6 的第 1 位 (发送缓冲数据标志)
TXSF6: ASIF6 的第 0 位 (发送移位寄存器数据标志)

图 11-17 显示了启动连续发送操作的时序，图 11-18 显示了结束连续发送操作的时序。

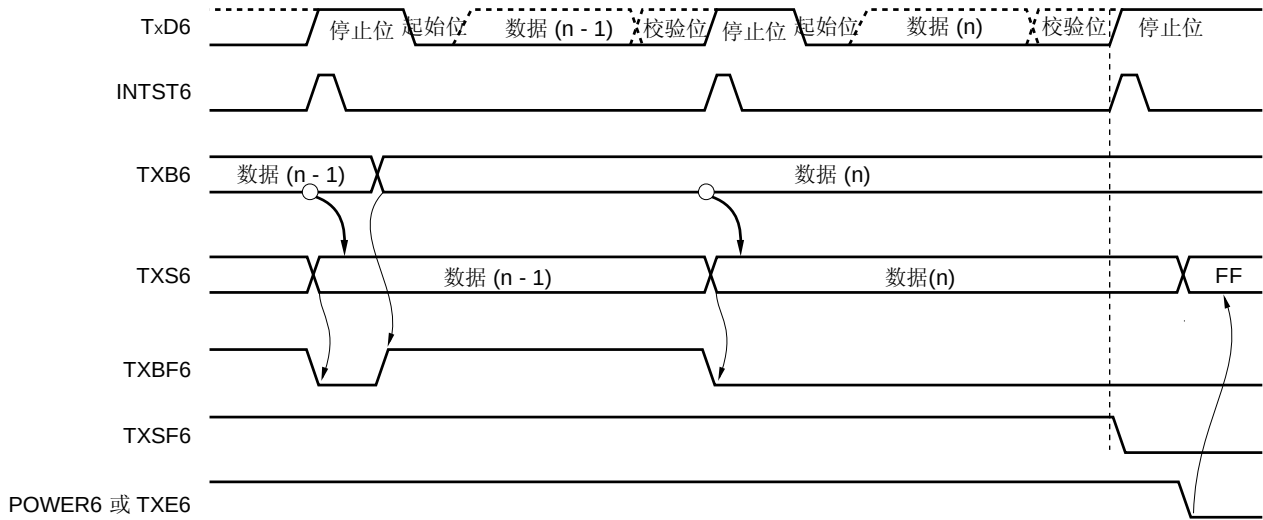
图 11-17. 启动连续发送操作的时序



注 在读取 ASIF6 时，存在 TXBF6 和 TXSF6 = 1, 1 的情况。因此，仅用 TXBF6 来判断是否允许进行写操作。

- 备注**
- TxD6: TxD6 引脚（输出）
 - INTST6: 中断请求信号
 - TXB6: 发送缓冲寄存器 6
 - TXS6: 发送移位寄存器 6
 - ASIF6: 异步串行接口发送状态寄存器 6
 - TXBF6: ASIF6 的第 1 位
 - TXSF6: ASIF6 的第 0 位

图 11-18. 结束连续发送操作的时序



备注

TxD6:	TxD6 引脚（输出）
INTST6:	中断请求信号
TXB6:	发送缓冲寄存器 6
TXS6:	发送移位寄存器 6
ASIF6:	异步串行接口发送状态寄存器 6
TXBF6:	ASIF6 的第 1 位
TXSF6:	ASIF6 的第 0 位
POWER6:	异步串行接口操作模式寄存器（ASIM6）的第 7 位
TXE6:	异步串行接口操作模式寄存器（ASIM6）的第 6 位

(e) 正常接收

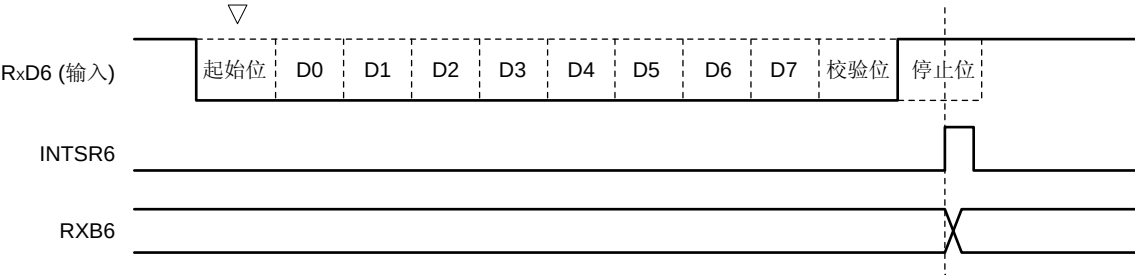
当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 为 1, 然后将 ASIM6 的第 5 位 (RXE6) 置 1 时, 允许接收并对 RxD6 引脚输入进行采样。

当检测到 RxD6 引脚输入的下降沿时, 波特率发生器的 8 位计数器开始计数。当计数到波特率发生器控制寄存器 6 (BRGC6) 的预设值时, 再次采样 RxD6 的输入信号 (如图 11-19 中的▽)。若此时 RxD6 引脚为低电平, 则被认为是起始位。

当检测到起始位时, 开始进行接收操作, 按照设置的波特率将串行数据依次存入接收移位寄存器 6 (RXS6) 中。当接收到停止位时, 会产生接收完成中断信号 (INTSR6), 并且 RXS6 的数据被写入接收缓冲寄存器 6 (RXB6)。但是, 如果出现溢出错误 (OVE6), 则接收数据不能写入 RXB6。

即使在接收过程中出现校验错 (PE6), 接收仍然继续, 直至接收到停止位, 而在接收完成时会产生一个出错中断 (INTSR6/INTSRE6)。

图 11-19. 接收完成中断请求时序



- 注意事项**
1. 即使出现接收错误, 也必须读取接收缓冲寄存器 6 (RXB6)。否则, 当接收到下一个数据时会产生溢出错误, 而且接收错误状态保持不变。
 2. 接收始终按照“停止位的个数 = 1 来执行。第 2 个停止位被忽略。
 3. 在读取 RXB6 之前, 必须先读取异步串行接口接收错误状态寄存器 6 (ASIS6)。

(f) 接收错误

在接收期间可能会产生三种错误： 校验错误，帧错误，或溢出错误。如果异步串行接口接收错误状态寄存器 6（ASIS6）的错误标志在数据接收后被置位，则会产生接收错误中断请求（INTSR6/INTSRE6）。可以通过在接收错误中断（INTSR6/INTSRE6）服务程序中读取 ASIS6 的内容来识别接收期间发生了哪一种错误（参见图 11-6）。当读 ASIS6 时，ASIS6 中的值被重置为 0。

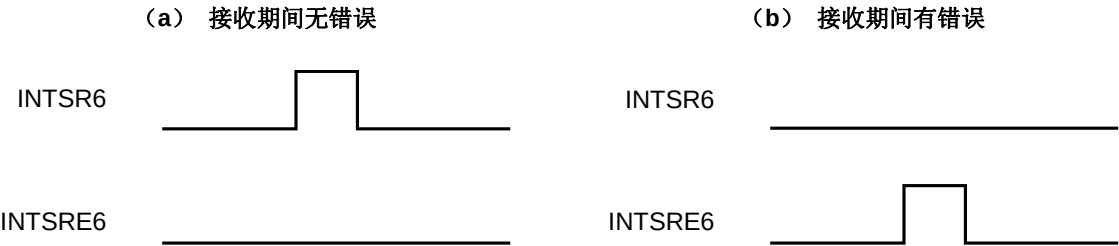
表 11-3. 接收错误的原因

接收错误	原因
校验错误	发送指定的校验和接收数据的校验不匹配
帧错误	未检测到停止位
溢出错误	在从接收缓冲寄存器 6（RXB6）读数据之前，下一个数据接收已经完成

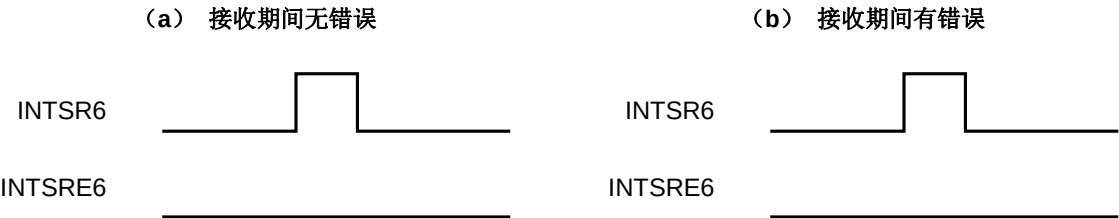
通过异步串行接口操作模式寄存器 6（ASIM6）的第 0 位清零，可把错误中断分为接收完成中断（INTSR6）和错误中断（INTSRE6）。

图 11-20. 接收错误中断

1. 如果 ISRM6 被清零（接收完成中断（INTSR6）和接收错误中断（INTSRE6）是分开的）



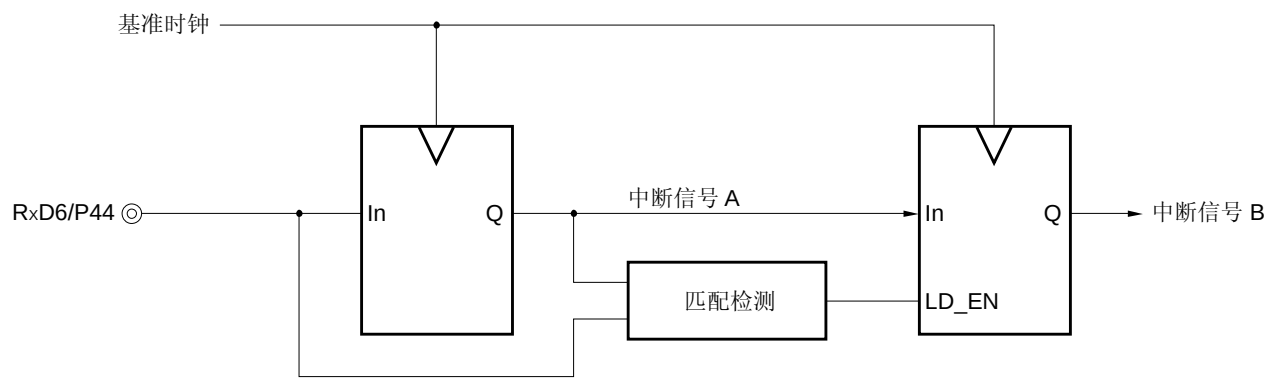
2. 如果 ISRM6 置为 1（错误中断包含在 INTSR6 中）



(g) 接收数据的噪声过滤器

使用预分频模块输出的基准时钟（f_{CLK6}）来采样 RxD6 信号。
如果两次采样值相同，则匹配检测器的输出会发生变化，并把采样的数据作为输入数据。
噪声滤波器的电路结构如图 11-21 所示，接收操作的内部过程从外部信号状态开始，延迟两个时钟执行。

图 11-21. 噪声过滤器电路



(h) SBF 发送

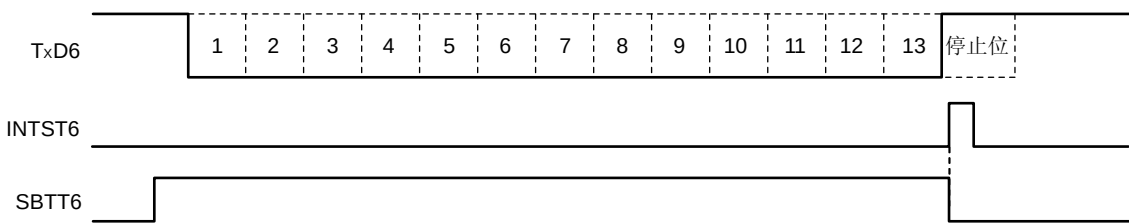
该接口用作 LIN 通讯操作模式，可使用 SBF（同步中断域）发送控制功能进行发送。LIN 的发送操作，如图 11-1 LIN 的发送操作。

当异步串行接口操作模式寄存器 6（ASIM6）的第 7 位（POWER6）置 1 时，TxD6 引脚输出高电平。接下来，当将第 6 位（RXE6）置 1 时，进入发送允许状态，当异步串行接口控制寄存器 6（ASICL6）的第 5 位（SBTT6）设置为 1 时，启动 SBF 发送。

此后，输出 13~20 位（由 ASICL6 的第 4~2 位（SBL62~SBL60）设置）的低电平。SBF 发送结束之后，产生发送完成中断请求（INTST6），并且 SBTT6 自动清零。此后，恢复正常发送模式。

发送被挂起，直到下一个要发送的数据被写入发送缓冲寄存器 6（TXB6），或直到 SBTT6 被置为 1。

图 11-22. SBF 发送



备注 TxD6: TxD6 引脚（输出）
INTST6: 发送完成中断请求
SBTT6: 异步串行接口控制寄存器 6（ASICL6）的第 5 位

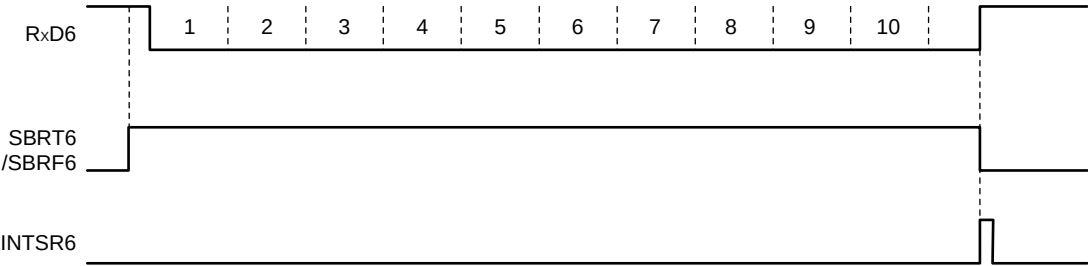
(i) SBF 接收

该接口用作 LIN 通讯操作模式，可使用 SBF（同步间隔域）接收控制功能进行接收。LIN 的接收操作，请参见图 11-2 LIN 接收操作。

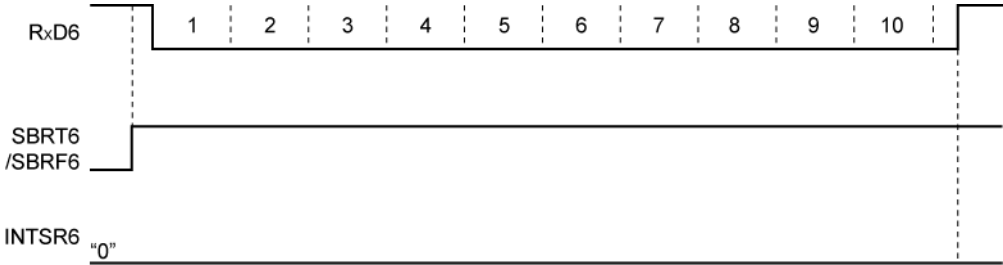
当异步串行接口操作模式寄存器 6（ASIM6）的第 7 位（POWER6）置 1，然后再将第 5 位（RXE6）置 1 时，允许接收。当异步串行接口控制寄存器 6（ASICL6）的第 6 位（SBRT6）置为 1 时，允许进行 SBF 接收。在 SBF 接收允许状态下，对 RxD6 引脚进行采样，以正常接收允许状态下同样的方式检测起始位。当检测到起始位时，启动接收操作，按照设定的波特率将串行数据依次存入接收移位寄存器 6（RXS6）中。接收到停止位时，如果 SBF 的宽度为 11 位或更多，会产生接收完成中断请求（INTSR6），与正常操作相同。此时，SBRF6 和 SBRT6 自动清零，SBF 接收操作结束。OVE6、PE6 和 FE6（异步串行接口接收错误状态寄存器 6（ASIS6）的第 0~2 位）等错误检测被挂起，不进行 UART 通信的错误检测。并且，也不执行接收移位寄存器 6（RXS6）和接收缓冲寄存器 6（RXB6）的数据传送，保存复位值 FFH。如果 SBF 的宽度为 10 位或更少，则在接收到停止位后不会产生中断，也不会进行出错处理，恢复 SBF 接收模式。此时，SBRF6 和 SBRT6 位不清零。

图 11-23. SBF 接收

1. 正常 SBF 接收（检测到停止位，且 SBF 宽度大于 10.5 位）



2. SBF 接收错误（检测到停止位，且 SBF 宽度小于或等于 10.5 位）



备注	RxD6:	RxD6 引脚（输入）
	SBRT6:	异步串行接口控制寄存器 6（ASICL6）的第 6 位
	SBRF6:	ASICL6 的第 7 位
	INTSR6:	接收完成中断请求

11.4.3 专用波特率发生器

专用波特率发生器包括一个源时钟选择器和一个 8 位可编程计数器，用于产生 UART6 发送/接收的串行时钟。
用于发送和接收的 8 位计数器分别提供。

(1) 波特率发生器的配置

- 基准时钟

当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 设置为 1 时，由时钟选择寄存器 6 (CKSR6) 的第 3~0 位 (TPS63~TPS60) 选择的时钟提供给每个模块。该时钟称为基准时钟，它的频率用 f_{CLK6} 表示。当 POWER6 = 0 时，基准时钟恒为低电平。

- 发送计数器

当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 或第 6 位 (TXE6) 为 0 时，该计数器停止计数，并被清零。

当 POWER6 = 1 且 TXE6 = 1 时，开始计数。

当第 1 个发送数据写入发送缓冲寄存器 6 (TXB6) 时，计数器被清零。

如果连续发送数据，则当一帧数据发送完成时，计数器再次被清零。如果后面没有数据要发送，则计数器不被清零，而是继续计数直至 POWER6 或 TXE6 被清零。

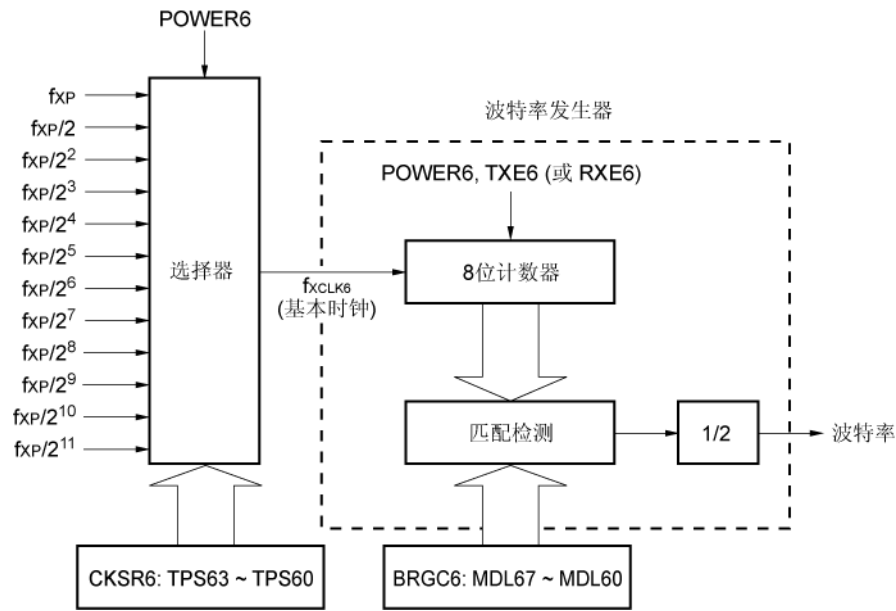
- 接收计数器

当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 或第 5 位 (RXE6) 为 0 时，该计数器停止计数，并被清零。

当检测到起始位时，开始计数。

在收到一帧数据后，计数器停止计数，直到检测到下一个起始位。

图 11-24. 波特率发生器的配置



备注

POWER6: 异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位

TXE6: ASIM6 的第 6 位

RXE6: ASIM6 的第 5 位

CKSR6: 时钟选择寄存器 6

BRGC6: 波特率发生器控制寄存器 6

(2) 串行时钟的产生

使用时钟选择寄存器 6 (CKSR6) 和波特率发生器控制寄存器 6 (BRGC6)，可以产生串行时钟。

根据 CKSR6 的第 3 ~ 0 位 (TPS63 ~ TPS60)，可以选择输入到 8 位计数器的时钟。

根据 BRGC6 的第 7~0 位 (MDL67~MDL60) 选择 8 位计数器的分频值。

(a) 波特率

通过下列公式计算波特率。

$$\bullet \text{ 波特率} = \frac{f_{\text{CLK6}}}{2 \times k} \quad [\text{bps}]$$

f_{CLK6} : 根据 CKSR6 的 TPS63~TPS60 位选择的基准时钟频率

k : 根据 BRGC6 寄存器的 MDL67 ~ MDL60 位设置的值 ($k = 8, 9, 10, \dots, 255$)

(b) 波特率误差

通过下列公式计算波特率误差。

$$\bullet \text{ 误差率}(\%) = \left[\frac{\text{实际波特率 (有误差的波特率)}}{\text{目标波特率 (正确的波特率)}} - 1 \right] \times 100 [\%]$$

- 注意事项**
1. 在接收端传送期间保持波特率误差在允许的误差范围内。
 2. 接收期间的波特率误差必须满足 (4) 接收期间允许的波特率范围 中所描述的范围。

举例: 基准时钟频率 = 10 MHz = 10,000,000 Hz
 设置 BRGC6 寄存器的 MDL67 ~ MDL60 位 = 00100001B ($k = 33$)
 目标波特率 = 153600 bps

$$\begin{aligned} \text{波特率} &= 10 \text{ M} / (2 \times 33) \\ &= 10000000 / (2 \times 33) = 151,515 [\text{bps}] \end{aligned}$$

$$\begin{aligned} \text{误差} &= (151515/153600 - 1) \times 100 \\ &= -1.357 [\%] \end{aligned}$$

(3) 波特率设置举例

表 11-4. 波特率发生器的数据设置

波特率 [bps]	$f_{XP} = 10.0 \text{ MHz}$				$f_{XP} = 8.38 \text{ MHz}$				$f_{XP} = 4.19 \text{ MHz}$			
	TPS63 ~ TPS60	k	计算值	ERR[%]	TPS63 ~ TPS60	k	计算值	ERR[%]	TPS63 ~ TPS60	k	计算值	ERR[%]
600	6H	130	601	0.16	6H	109	601	0.11	5H	109	601	0.11
1200	5H	130	1202	0.16	5H	109	1201	0.11	4H	109	1201	0.11
2400	4H	130	2404	0.16	4H	109	2403	0.11	3H	109	2403	0.11
4800	3H	130	4808	0.16	3H	109	4805	0.11	2H	109	4805	0.11
9600	2H	130	9615	0.16	2H	109	9610	0.11	1H	109	9610	0.11
10400	1H	240	10417	0.16	1H	201	10423	0.22	1H	101	10475	-0.28
19200	1H	130	19231	0.16	1H	109	19220	0.11	0H	109	19220	0.11
31250	0H	160	31250	0.00	0H	134	31268	0.06	0H	67	31268	0.06
38400	0H	130	38462	0.16	0H	109	38440	0.11	0H	55	38090	-0.80
76800	0H	65	76923	0.16	0H	55	76182	-0.80	0H	27	77693	1.03
115200	0H	43	116279	0.94	0H	36	116389	1.03	0H	18	116389	1.03
153600	0H	33	151515	-1.36	0H	27	155185	1.03	0H	14	149643	-2.58
230400	0H	22	227272	-1.36	0H	18	232778	1.03	0H	9	232778	1.03

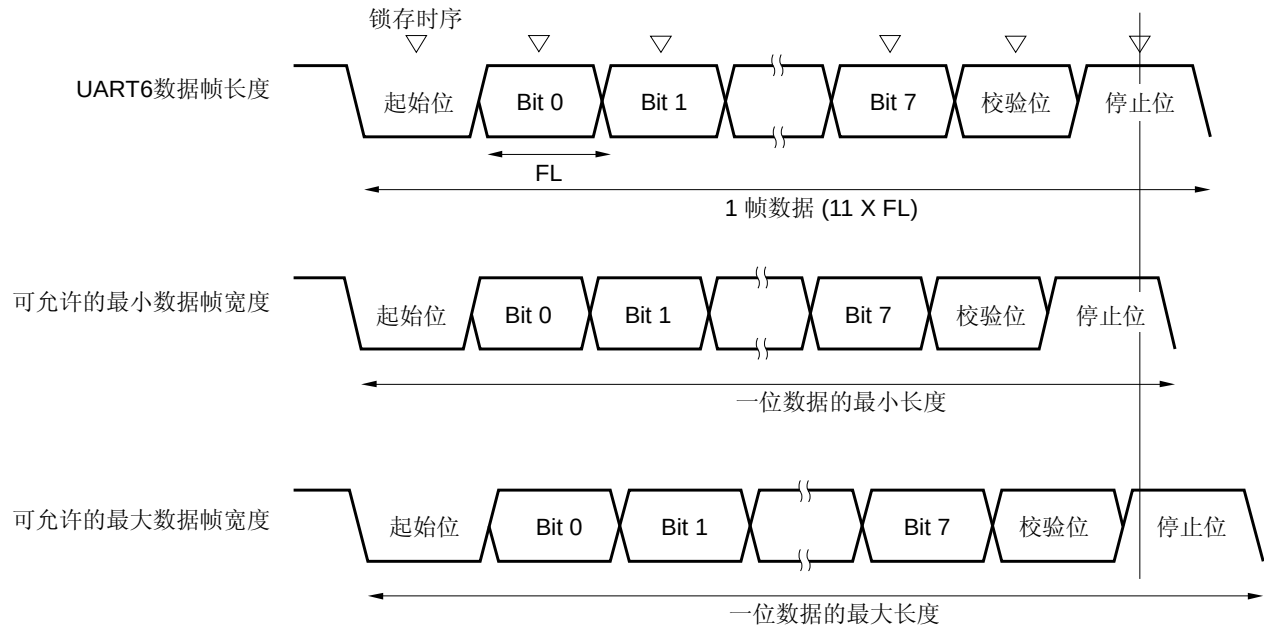
备注 TPS63 ~ TPS60: 时钟选择寄存器 6 (CKSR6) 的第 3 ~ 0 位 (设置基准时钟 (f_{XCLK6}))
 k: 通过 BRGC6 寄存器的 MDL67 ~ MDL60 位设置的值 ($k = 8, 9, 10, \dots, 255$)
 f_{XP} : 外围硬件时钟的振荡频率
 ERR: 波特率误差

(4) 接收期间允许的波特率范围

接收期间来自发送端的波特率可允许的误差如下所示。

注意事项 必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。

图 11-25. 接收期间允许的波特率范围



如图 11-25 所示，当检测到起始位后，接收数据的锁存时序由波特率发生器控制寄存器 6（BRGC6）设置的计数器确定。如果数据的最后一位（停止位）满足该锁存时序，则数据可以正确接收。
假定接收数据为 11 位，各项理论值计算如下。

$$FL = (\text{Brate})^{-1}$$

- Brate: UART6 的波特率
- k: BRGC6 的设置值
- FL: 1 位数据长度
- 锁存时序的容限: 2 个时钟周期

可允许的最小数据帧长度：
$$FL_{min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

因此，在发送端的可接收的最大波特率如下所示。

$$BR_{max} = (FL_{min}/11)^{-1} = \frac{22k}{21k+2} Brate$$

同样，允许的最大数据帧长度计算如下。

$$\frac{10}{11} \times FL_{max} = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FL_{max} = \frac{21k-2}{20k} FL \times 11$$

因此，在发送端可接收的最小波特率如下所示。

$$BR_{min} = (FL_{max}/11)^{-1} = \frac{20k}{21k-2} Brate$$

可通过上述最小和最大波特率公式，计算 UART6 与发送源之间允许的波特率误差，如下所示。

表 11-5. 允许的最大/最小波特率误差

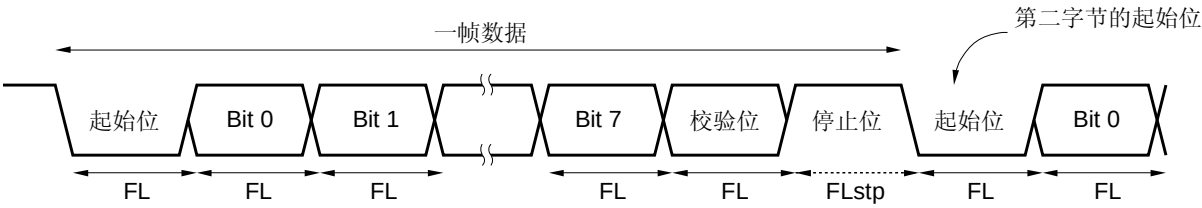
分频比率 (k)	允许的最大波特率误差	允许的最小波特率误差
8	+3.53%	-3.61%
20	+4.26%	-4.31%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.73%

- 备注**
1. 接收时允许的误差由每帧的位数、输入时钟频率和分频比率 (k) 决定。输入时钟频率和分频比率 (k) 越高，允许的误差就越大。
 2. k: BRGC6 的设置值

(5) 连续发送期间的数据帧长度

当连续发送数据时，从停止位到下一个起始位的数据帧长度比正常值延长了两个基准时钟（f_{XCLK6}）周期。但是，由于在检测到起始位时接收端的时序被初始化，因此通信结果不会受到影响。

图 11-26. 连续发送期间的数据帧长度



当 1 位数据长度为 FL、停止位长度为 FLstp，而基准时钟频率为 f_{XCLK6}，有以下公式成立。

$$FLstp = FL + 2/f_{XCLK6}$$

因此，连续发送期间的数据帧长度为：

$$\text{数据帧长度} = 11 \times FL + 2/f_{XCLK6}$$

12.1 乘法器功能

乘法器具有以下功能。

- 计算 8 位 × 8 位 = 16 位

12.2 乘法器配置

(1) 16 位乘法结果存储寄存器 0 (MUL0)

该寄存器存储 16 位乘法结果。

经过 16 个 CPU 时钟后，乘法结果保存在该寄存器中。

MUL0 可以使用 16 位存储器操作指令进行读取。

复位后 MUL0 的值不确定。

注意事项 这个寄存器既能用 16 位存储器操作指令操作，也能用 8 位存储器操作指令操作。但是，使用 8 位存储器操作指令对其进行操作时，要通过直接寻址方式对其进行访问。

(2) 乘法数据寄存器 A 和 B (MRA0 和 MRB0)

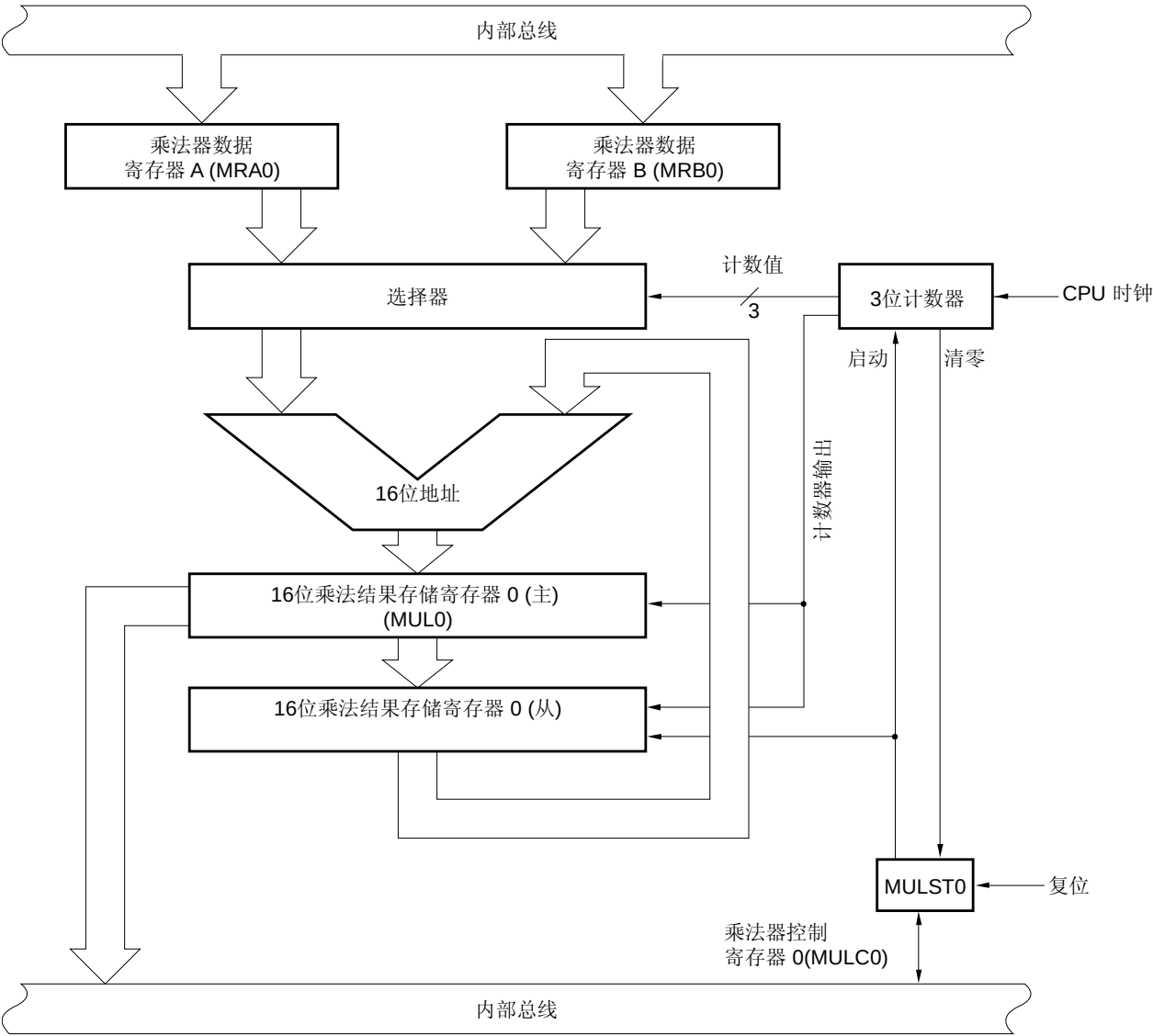
这些是 8 位乘法数据存储寄存器。乘法器将 MRA0 和 MRB0 的值进行相乘。

MRA0 和 MRB0 可以使用 8 位存储器操作指令进行写入。

复位后这两个寄存器的值不确定。

图 12-1 显示了乘法器的框图。

图 12-1. 乘法器的框图



12.3 乘法器控制寄存器

乘法器通过下列寄存器进行控制。

- 乘法器控制寄存器 0（MULC0）

（1）乘法器控制寄存器 0（MULC0）

该寄存器指示乘法器操作后的状态，也可以对乘法器进行控制。

MULC0 可以使用 1 位或 8 位存储器操作指令进行设置。

复位后该寄存器的值为 00H。

图 12-2. 乘法器控制寄存器 0（MULC0）的格式

地址：FFD2H 复位后：00H 读/写

符号	7	6	5	4	3	2	1	<0>
MULC0	0	0	0	0	0	0	0	MULST0

MULST0	乘法器操作启动控制位	乘法器的操作状态
0	在计数器重置为 0 后停止操作	操作停止
1	允许操作	操作进行中

注意事项 确保第 1~7 位清 0。

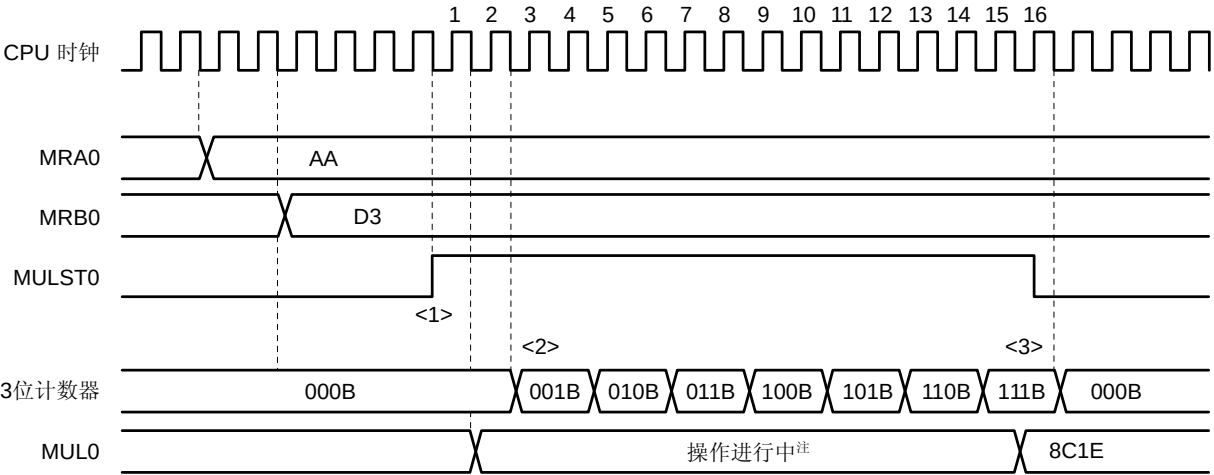
12.4 乘法器操作

78K0S/KB1+ 的乘法器能够执行 8 位 × 8 位= 16 位的计算。图 12-3 显示了当 MRA0 设置为 AAH，MRB0 设置为 D3H 时的乘法器操作时序。

- <1> 通过设置 MULST0 来启动计数，然后 MRA0 和 MRB0 的乘法开始。乘法结果在 16 个时钟后得到，使用 CPU 时钟。
- <2> 选择器产生的数据在每个 CPU 时钟与 MUL0 的数据相加，并且计数器的值加 1。
- <3> 当 3 位计数器的值为 111B 时，MULST0 被清零，操作被停止。此时，MUL0 内保存结果数据。

备注 当 MULST0 为低时，3 位计数器清零。

图 12-3. 乘法器操作时序（示例为 AAH × D3H）



注 计算时的中间结果通过参考 MUL0 读取。

此示例的源代码在下页中显示。

```

=====
; 乘法器使用示例
=====
M_DATA_A EQU 0FE80H ; 乘法器地址 A
M_DATA_B EQU 0FE90H ; 乘法器地址 B

; 操作的设置
MOV M_DATA_A, #0AAH
MOV M_DATA_B, #0D3H

; M_DATA_A 与 M_DATA_B 的乘法
MOV A, M_DATA_A
MOV MRA0, A
MOV A, M_DATA_B
MOV MRB0, A

SET1 MULST0 ; 乘法启动

M_LOOP:
BT MULST0, $M_LOOP ; 等待乘法结束
MOVW AX, MUL0 ; 乘法完成

```

第十三章 中断功能

13.1 中断功能类型

中断有两种类型：可屏蔽中断和复位。

- **可屏蔽中断**

这类中断可进行屏蔽控制。当发生一个中断请求时，产生一个释放待机状态的信号。如果该中断可以被响应，则执行向量表地址上的相应程序（向量中断服务）。如果同时产生多个中断请求，根据中断优先顺序，优先响应高优先级的中断。中断的优先级的详细情况，请参见表 13-1。

共有 9 个内部可屏蔽中断源和 4 个外部可屏蔽中断源。

- **复位**

复位信号使 CPU 和 SFR 返回到初始状态。产生复位信号的原因见表 13-1。

当产生复位信号时，程序从地址 0000H 和 0001H 处开始执行。

13.2 中断源和配置

共有 13 个可屏蔽中断源，和 4 种复位源。（见 表 13-1）

表 13-1. 中断源

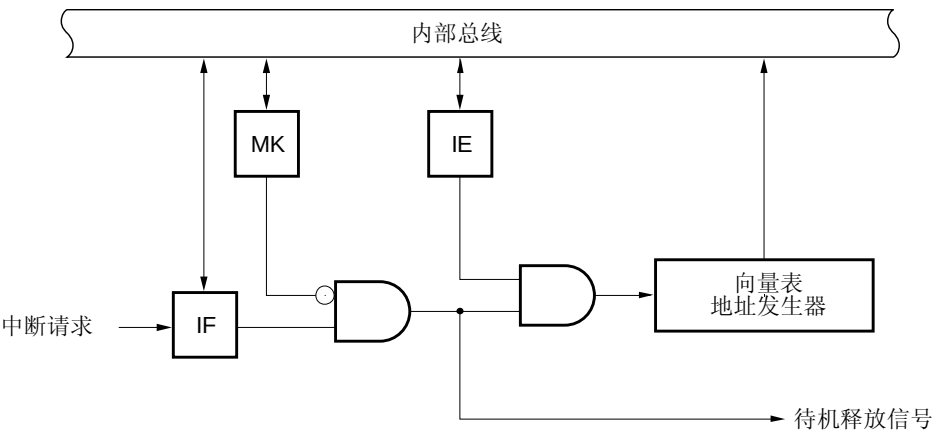
中断类型	优先级 ^{注1}	中断源		内部/ 外部	向量表地址	基本配置类型 ^{注2}
		名称	触发			
可屏蔽的	1	INTLVI	低电压检测 ^{注3}	内部	0006H	(A)
	2	INTP0	引脚输入边沿检测	外部	0008H	(B)
	3	INTP1			000AH	
	4	INTTMH1	TMH1 和 CMP01 匹配	内部	000CH	(A)
	5	INTTM000	TM00 和 CR000 匹配（指定比较寄存器），TI010 引脚有效沿检测（指定捕捉寄存器）		000EH	
	6	INTTM010	TM00 和 CR010 匹配（指定比较寄存器），TI000 引脚有效沿检测（指定捕捉寄存器）		0010H	
	7	INTAD	A/D 转换结束		0012H	
	8	INTP2	引脚输入边沿检测	外部	0016H	(B)
	9	INTP3			0018H	
	10	INTTM80	TM80 和 CR80 匹配	内部	001AH	(A)
	11	INTSRE6	产生 UART6 接收错误		001CH	
	12	INTSR6	UART6 接收结束		001EH	
	13	INTST6	UART6 发送结束		0020H	
复位	-	RESET	复位输入	-	0000H	-
		POC	上电清零			
		LVI	低电压检测 ^{注4}			
		WDT	看门狗溢出			

- 注 1. 优先级是指当有多个中断请求同时输入时，各个中断所对应的优先处理顺序。1 是最高优先级，13 为最低优先级。
2. (A) 和 (B) 这两种基本配置类型，如图 13-1 中的 (A) 和 (B) 所示。
3. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) = 0 时。
4. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) = 1 时。

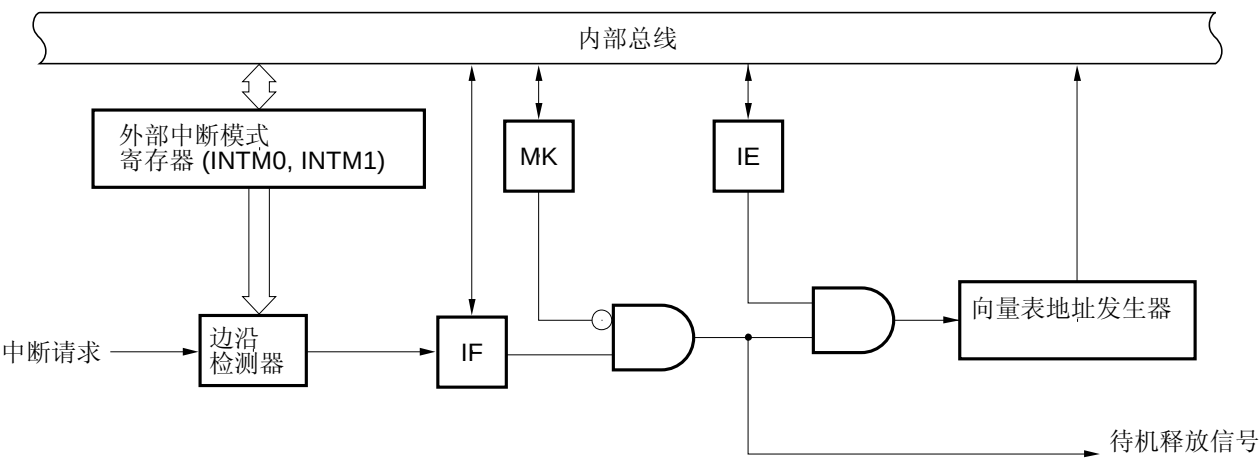
注意事项 向量表地址 0014H 不对应任何中断源。

图 13-1. 中断功能的基本配置

(A) 内部可屏蔽中断



(B) 外部可屏蔽中断



IF: 中断请求标志
IE: 中断允许标志
MK: 中断屏蔽标志

13.3 中断功能的控制寄存器

通过以下 4 种寄存器来控制中断功能。

- 中断请求标志寄存器（IF0，IF1）
- 中断屏蔽标志寄存器（MK0，MK1）
- 外部中断模式寄存器（INTM0，INTM1）
- 程序状态字（PSW）

表 13-2 列出了中断请求，与其相对应的中断请求标志和中断屏蔽标志。

表 13-2. 中断请求信号及对应标志位

中断请求信号	中断请求标志	中断屏蔽标志
INTLVI	LVIF	LVIMK
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTTMH1	TMIFH1	TMMKH1
INTTM000	TMIF000	TMMK000
INTTM010	TMIF010	TMMK010
INTAD	ADIF	ADMK
INTP2	PIF2	PMK2
INTP3	PIF3	PMK3
INTTM80	TMIF80	TMMK80
INTSRE6	SREIF6	SREMK6
INTSR6	SRIF6	SRMK6
INTST6	STIF6	STMK6

(1) 中断请求标志寄存器 0, 1 (IF0, IF1)

当产生相应的中断请求时或执行指令时，中断请求标志被置 1。当中断请求被响应或产生复位信号时，通过执行一条指令，将这些标志清零。

IF0 和 IF1 可由 1 位或 8 位存储器操作指令设置。

复位后将 IF0 和 IF1 清为 00H。

图 13-2. 中断请求标志寄存器 0, 1 (IF0, IF1) 的格式

地址: FFE0H 复位后: 00H R/W

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

地址: FFE1H 复位后: 00H R/W

符号	7	<6>	<5>	<4>	<3>	<2>	<1>	0
IF1	0	STIF6	SRIF6	SREIF6	TMIF80	PIF3	PIF2	0

xxIFx	中断请求标志
0	没有产生中断请求标志
1	产生一个中断请求，中断请求状态

注意事项 因为 P30, P31, P41 和 P43 都有作为外部中断输入的复用功能，所以通过指定端口功能的输出模式来改变输出电平，中断请求标志置 1。因此，在使用输出模式之前，应先将中断屏蔽标志置 1。

(2) 中断屏蔽标志寄存器 0, 1 (MK0, MK1)

这些中断屏蔽标志用于允许/禁止相应的可屏蔽中断。
MK0 和 MK1 可由 1 位或 8 位存储器操作指令设置。
复位后将 MK0 和 MK1 置为 FFH。

图 13-3. 中断屏蔽标志寄存器 0, 1 (MK0, MK1) 的格式

地址: FFE4H 复位后: FFH R/W

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

地址: FFE5H 复位后: FFH R/W

符号	7	<6>	<5>	<4>	<3>	<2>	<1>	0
MK1	1	STMK6	SRMK6	SREMK6	TMMK80	PMK3	PMK2	1

xxMKx	中断服务控制
0	允许中断服务
1	禁止中断服务

注意事项 因为 P30, P31, P41 和 P43 都有作为外部中断输入的复用功能，所以通过指定端口功能的输出模式来改变输出电平，中断请求标志置 1。因此，在使用输出模式之前，应先将中断屏蔽标志置 1。

(3) 外部中断模式寄存器 0 (INTM0)

这个寄存器用来设置 INTP0~INTP2 的有效沿。

INTM0 可用 8 位存储器操作指令设置。

复位后将 INTM0 清为 00H。

图 13-4. 外部中断模式寄存器 0 (INTM0)

地址: FFECH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
INTM0	ES21	ES20	ES11	ES10	ES01	ES00	0	0

ES21	ES20	INTP2 引脚有效沿的选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升和下降沿均为有效沿

ES11	ES10	INTP1 引脚有效沿的选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升和下降沿均为有效沿

ES01	ES00	INTP0 引脚有效沿的选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升和下降沿均为有效沿

- 注意事项
1. 确保第 0 位和第 1 位清零。
 2. 在设置 INTM0 寄存器之前，必须设置相应的中断屏蔽标志 (xxMKx = 1) 以禁止中断。INTM0 寄存器设置之后，首先对中断请求标志 (xxIFx = 0) 清零，然后再对中断屏蔽标志 (xxMKx = 0) 清零，以允许中断。

INTM1 用来设置 INTP3 的有效沿。
INTM1 可以通过 8 位存储器操作指令进行设置。
复位后将 INTM1 清为 00H。

地址: FFEDH 复位后: 00H R/W

ES31	ES30	INTP3 引脚有效沿的选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升和下降沿均为有效沿

(5) 程序状态字 (PSW)

程序状态字用于保存指令执行结果和中断请求的当前状态。**IE** 标志用来允许和禁止可屏蔽中断，被映射到 **PSW** 内。

PSW 可用 8 位读写操作指令，还可以使用位操作指令和专用指令（**EI** 和 **DI**）对该寄存器进行操作。当一个向量中断被响应时，**PSW** 自动被存入堆栈，**IE** 标志被清零。

复位后将程序状态字的值置为 **02H**。

符号 7 6 5 4 3 2 1 0 复位后

PSW IE Z 0 AC 0 0 1 CY 02H

在一般指令执行时使用

13.4 中断服务操作

13.4.1 可屏蔽的中断请求的响应操作

当中断请求标志被置 1，相应的中断屏蔽标志被清零时，可以响应可屏蔽中断请求。如果处于中断允许状态（当 IE 标志位置 1），向量中断可以被响应。

可屏蔽中断请求产生之后，启动中断服务所需时间，如表 13-3 所示。

图 13-8 和 13-9 为中断请求响应时序。

表 13-3. 从可屏蔽中断请求产生后到启动中断服务所需时间

最小时间	最大时间 ^注
9 个时钟周期	19 个时钟周期

注 当中断请求在 BT 和 BF 指令执行之前产生，等待的时间最长。

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

如果同时产生两个或更多的可屏蔽中断请求，则首先响应优先级高的中断请求。

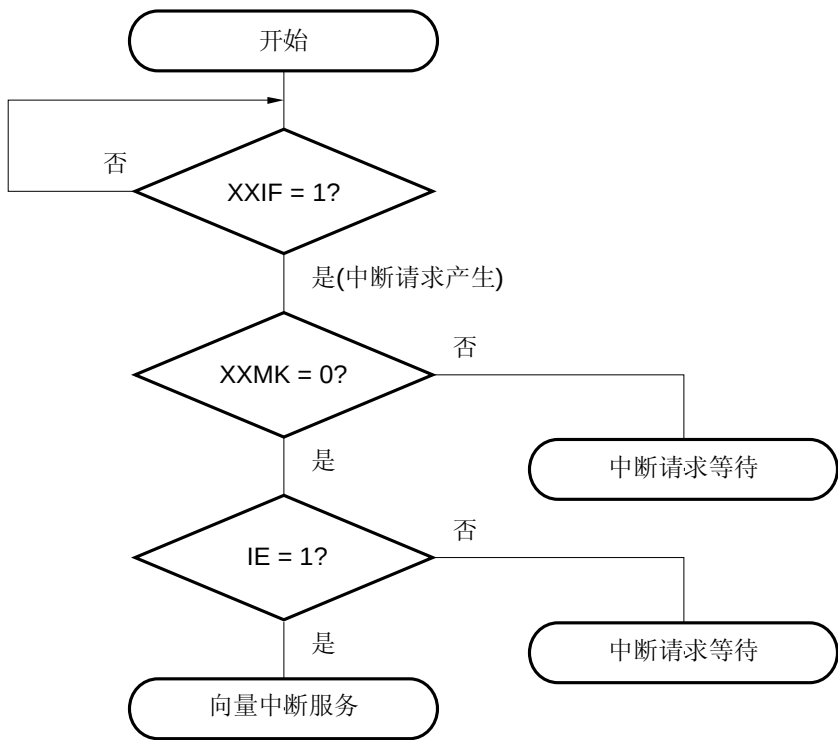
当允许响应中断请求时，处于待机状态中的请求被响应。

图 13-7 显示了中断请求响应算法。

当可屏蔽中断被响应后，PSW 和 PC 的内容被依次保存到堆栈中，IE 标志位清零，通过每个中断请求在向量表中的数据来决定哪个被装载到 PC，并执行跳转。

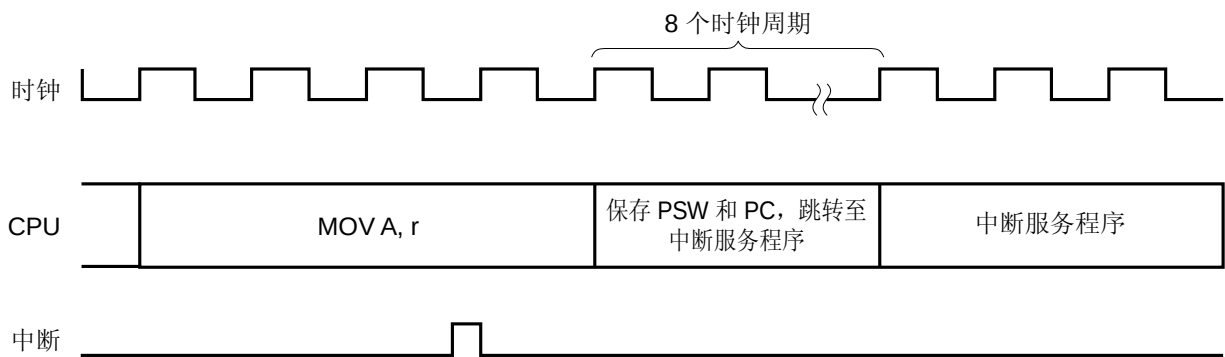
可使用 RETI 指令从中断返回。

图 13-7. 中断请求响应处理算法



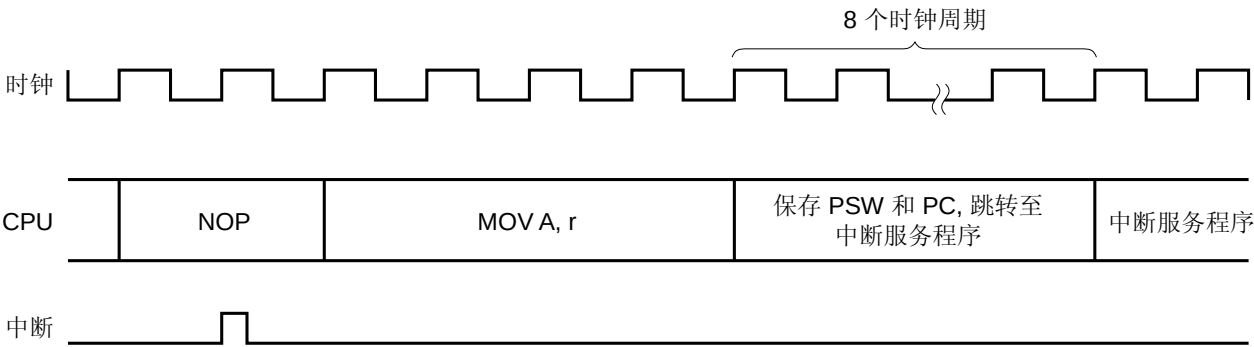
xxIF: 中断请求标志
xxMK: 中断屏蔽标志
IE: 控制可屏蔽中断请求响应的标志（1 = 允许，0 = 禁止）

图 13-8. 中断请求响应时序（以 MOV A, r 为例）



如果执行一条指令所需的时钟周期为 n ($n = 4 \sim 10$)，当执行到 $n-1$ 个时钟周期之前，中断请求标志 (xxIF) 被置位，那么在该指令执行完成后才能响应中断。图 13-8 为 8 位数据传送指令 MOV A, r 的中断请求响应时序示例。该指令执行时钟周期为 4 个时钟周期，如果在指令取指开始后的第 3 个时钟周期发生中断请求，在指令 MOV A, r 执行完成后开始中断响应处理。

图 13-9 中断请求响应时序（中断请求标志在指令执行的最后一个时钟周期置位）



如果中断请求标志在指令执行的最后一个时钟周期被置位，那么在执行完下一条指令之后，才开始中断响应处理。

图 13-9 显示中断请求响应时序的例子，中断请求标志在 NOP 指令（2 时钟周期的指令）的第 2 个时钟周期置位。此时，NOP 指令的下一条指令为 MOV A, r 执行完成后，开始中断请求的响应处理。

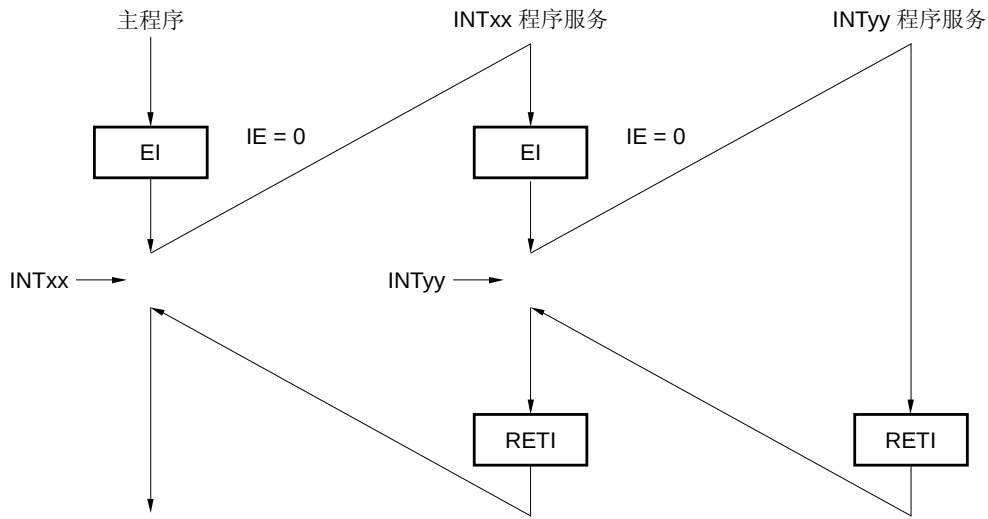
注意事项 当中断请求标志寄存器 0, 1 (IF0, IF1) 或中断屏蔽标志寄存器 0, 1 (MK0, MK1) 被访问期间，中断请求等待。

13.4.2 中断嵌套

中断嵌套是指在一个中断服务进行的同时要响应另一个中断，必须使用中断屏蔽功能来屏蔽低优先级的中断。

图 13-10. 中断嵌套示例 (1/2)

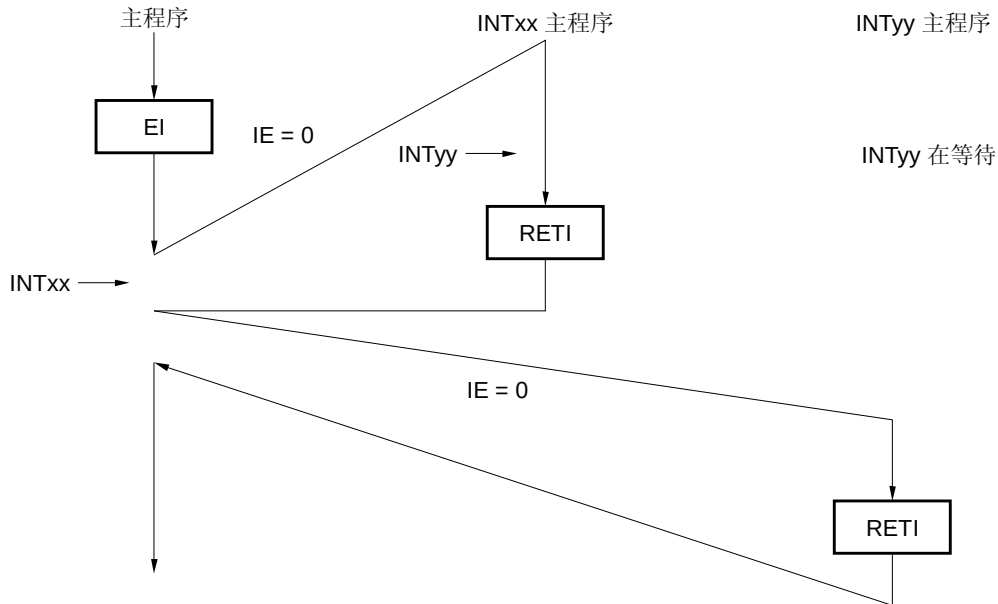
例 1. 响应中断嵌套



在中断 INTxx 处理期间，中断请求 INTyy 被响应，发生中断嵌套。在每个中断响应之前要执行 EI 指令，要释放中断屏蔽，并且设置中断请求响应为允许状态。

注意事项 即使是低优先级中断，中断嵌套也能响应。

例 2. 由于不允许中断没有产生中断嵌套



在 INTxx 中断服务期间没有允许中断（未执行 EI 指令），因此中断请求 INTyy 不响应，这样不会产生中断嵌套。中断请求 INTyy 处于等待状态，在 INTxx 中断服务程序完成后响应该中断请求。

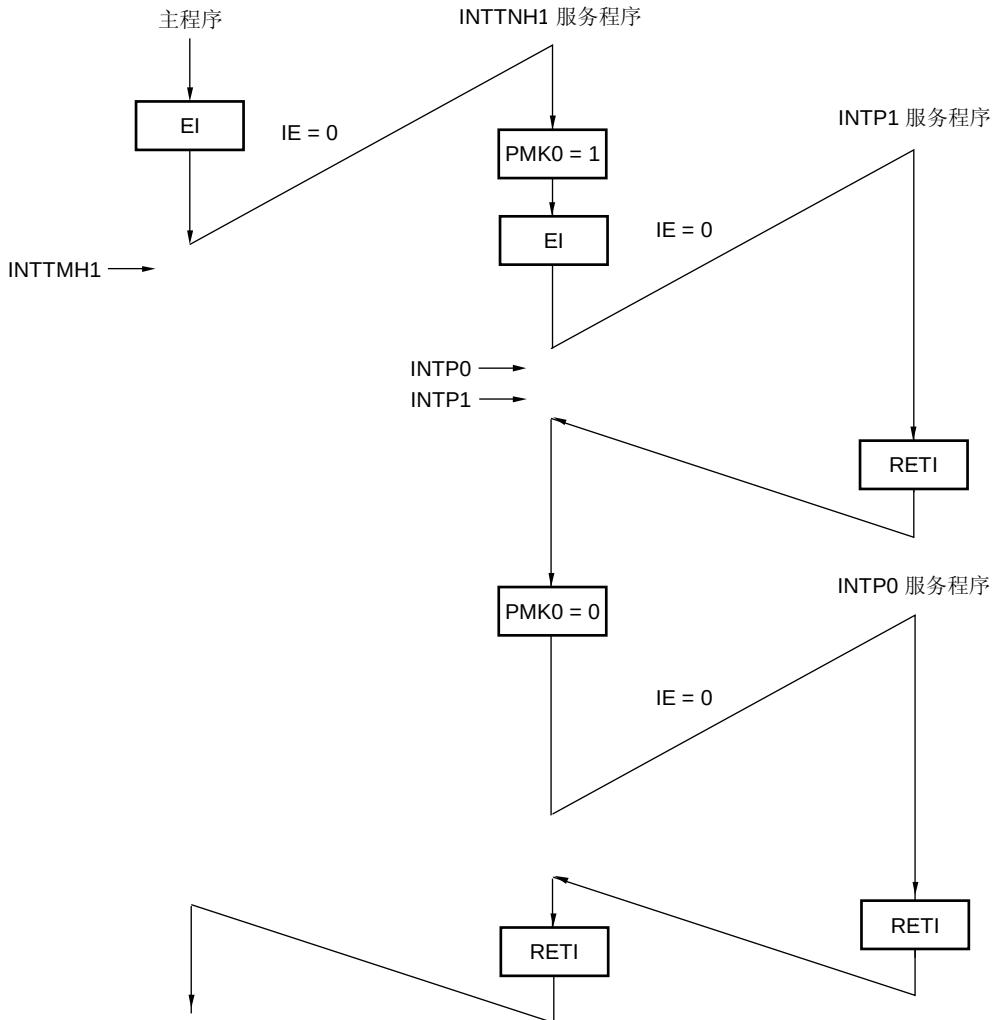
IE = 0: 禁止响应中断请求

图 13-10. 中断嵌套示例 (2/2)

例 3. 通过中断嵌套控制优先级

为 INTP0, INTP1 和 INTTMH1 设置向量中断允许状态。

(中断优先级 INTP0 > INTP1 > INTTMH1 (参照 表 13-1))



在 INTTMH1 中断服务进行时，服务如图进行，INTP1 中断优先，因为 INTP0 最初是被屏蔽的。后来，一旦 INTP0 屏蔽被释放，通过中断嵌套的 INTP0 处理就会执行。

IE = 0: 禁止响应中断请求

13.4.3 中断请求待待

在执行某些指令时，即使有中断请求发生（可屏蔽中断或外部中断），应该等待此条指令和它的下一条指令后执行完毕，才能响应中断，在此期间中断响应暂时等待。以下为这种指令（中断请求等待指令）。

- 对中断请求标志寄存器 0, 1 (IF0, IF1) 进行操作的指令
- 对中断屏蔽标志寄存器 0, 1 (MK0, MK1) 进行操作的指令

第十四章 待机功能

14.1 待机功能及配置

14.1.1 待机功能

表 14-1. 每种工作状态下操作时钟之间的关系

状态 操作模式	低速内部振荡器			系统时钟	供给外部硬件 的时钟
	注 1	注 2			
		LSRSTOP = 0	LSRSTOP = 1		
复位	停止			停止	停止
STOP	振荡	振荡 ^{注 3}	停止		
HALT					

- 注
- 1. 当选项字节选择低速内部振荡器“不能被停止”时。
 - 2. 当选择了低速内部振荡器“可以通过软件停止”时，低速内部振荡器的振荡可通过 LSRSTOP 停止。
 - 3. 如果看门狗定时器的操作时钟是低速内部时钟，看门狗定时器被停止。

注意事项 仅当选项字节选择低速内部振荡器“可通过软件停止”时，LSRSTOP 设置才有效。

备注 LSRSTOP： 低速内部振荡模式寄存器（LSRCM）的第 0 位

待机功能用于减少系统的工作电流。有以下两种模式。

(1) HALT 模式

通过执行 HALT 指令设置 HALT 模式。在 HALT 模式下，CPU 操作时钟停止。系统时钟振荡器继续振荡。如果在设置 HALT 模式之前，低速内部振荡器在工作，则低速内部振荡器继续振荡（参见表 14-1. 由选项字节选择低速内部时钟（是否能通过软件停止））。在此模式中，工作电流没有 STOP 模式中那么低，但 HALT 模式对于中断请求产生后立即重启操作非常有效，也适用于频繁的间歇操作。

(2) STOP 模式

通过执行 STOP 指令设置 STOP 模式。在 STOP 模式中，系统时钟振荡器停止，整个系统的操作停止，这样 CPU 的工作电流将会明显下降。

该模式可通过中断请求退出，允许进行间歇操作。但是，由于在 STOP 模式释放后，会产生操作停止时间^{*}（当使用陶瓷/晶体振荡器时，要经过一段额外的等待时间，以确保振荡器振荡稳定），因此如果需要响应中断请求后立即进行处理，则应采用 HALT 模式。

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

在这两种模式中，寄存器、标志和数据存储器的内容将会保持进入待机模式前的内容。I/O 端口输出锁存和输出缓冲状态也将保持。

- 注意事项**
1. 当切换到 STOP 模式时，一定要在执行 STOP 指令之前停止外部硬件操作（除了以低速内部振荡时钟工作的外部硬件之外）。
 2. 当使用待机功能时，推荐按以下步骤减少 A/D 转换器的工作电流。首先对 A/D 转换器模式寄存器（ADM）的第 7 位（ADCS）和第 0 位（ADCE）清零，以停止 A/D 转换操作，然后执行 HALT 或 STOP 指令。
 3. 如果在置 STOP 模式之前，低速内部振荡器正在工作，则在 STOP 模式中不能停止低速内部振荡时钟（参见表 14-1）。

14.1.2 待机期间用到的寄存器

退出待机模式后的振荡稳定时间，由振荡稳定时间选择寄存器（OSTS）控制。

备注 有关这些寄存器启动、停止以及时钟选择的详细情况，请参见第五章 时钟发生器。

(1) 振荡稳定时间选择寄存器 OSTS

该寄存器用于选择 STOP 模式释放时振荡器提供时钟的振荡稳定时间。仅当晶体/陶瓷振荡器时钟被选为系统时钟源，并且在 STOP 模式释放后，通过 OSTS 设置的等待时间是有效的。如果选择高速内部振荡或外部时钟输入作为系统时钟源，无须等待时间。

上电后或复位模式释放后的系统时钟振荡器和振荡稳定时间，由选项字节选择。详细情况，请参见 第十八章 选项字节。

OSTS 可由 8 位存储器操作指令设置。

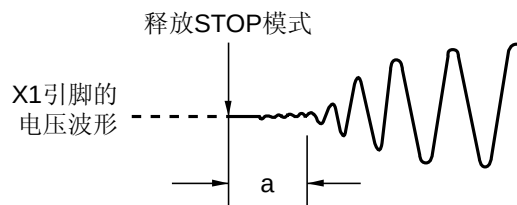
图 14-1 振荡稳定时间选择寄存器（OSTS）的格式

地址：FFF4H 复位后：不确定 读/写

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

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

- 注意事项**
1. 设置和释放 STOP 模式时，按照下面的方法设置振荡稳定时间。
预期的振荡器振荡稳定时间 $\leq$ 通过 OSTS 设置的振荡稳定时间
 2. 无论 STOP 模式是被复位信号打断还是被中断唤醒，STOP 模式释放后的等待时间不包括从释放 STOP 模式到时钟开始振荡的时间（下图中的“a”部分）。



3. 上电或复位释放后的振荡稳定时间，可以通过选项字节来选择。详细情况，请参见 第十八章 选项字节。

- 备注**
1. () : $f_x = 10 \text{ MHz}$
 2. 通过查看所使用振荡器的特性，决定振荡器的振荡稳定时间。

14.2 待机功能的操作

14.2.1 HALT 模式

(1) HALT 模式

执行 HALT 指令设置 HALT 模式。

HALT 模式中的工作状态如下所示。

注意事项 由于中断请求信号可用来退出待机模式，如果某一中断源中断请求标志置位且中断屏蔽标志清零，则立即退出待机模式。

表 14-2. HALT 模式下的工作状态

HALT 模式的设置 项目		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		CPU 时钟停止		
CPU		操作停止		
端口（锁存）		保持设置 HALT 模式前的状态		
16 位 定时器/事件计数器 00		可以操作		
8 位 定时器 80		可以操作		
8 位 定时器 H1	设置计数时钟为 $f_{XP} \sim f_{XP}/2^{12}$	可以操作		
	设置计数时钟为 $f_{RL}/2^7$	可以操作	可以操作	操作停止
看门狗定时器	选择系统时钟作为操作时钟	禁止设置	操作停止	
	选择“低速内部振荡时钟”作为操作时钟	可以操作（操作继续）	操作停止	
A/D 转换器		可以操作		
串行接口 UART6		可以操作		
POC 电路		一直可以操作		
低电压检测		可以操作		
外部中断		可以操作		

注 选项字节可以选择低速内部振荡器为“不能被停止”或“可以通过软件停止”（选项字节请参见第十八章 选项字节）。

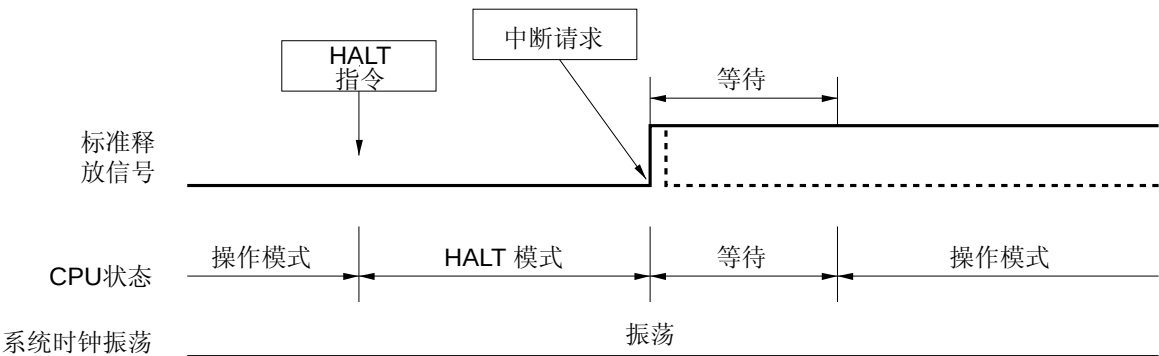
(2) 释放 HALT 模式

HALT 模式可通过以下两种中断源释放。

(a) 由未屏蔽的中断请求释放

当产生一个未屏蔽的中断请求时，释放 HALT 模式。如果允许响应中断，则执行向量中断服务。如果禁止响应中断，则执行下一个地址的指令。

图 14-2. 通过产生中断请求释放 HALT 模式



备注

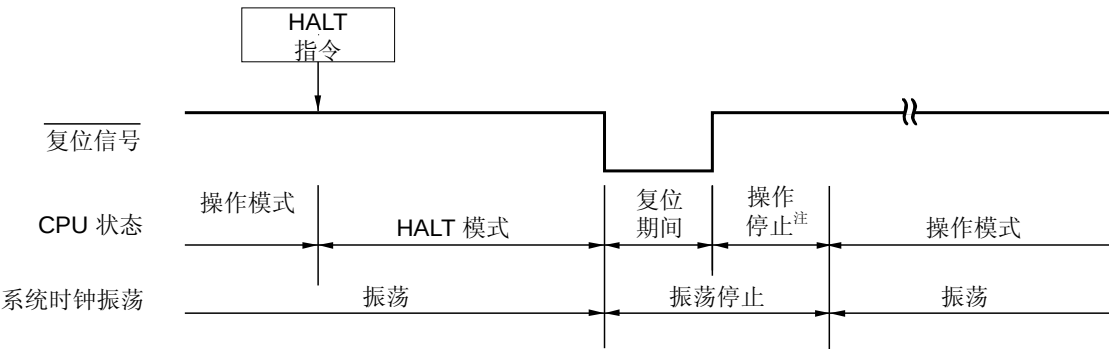
1. 虚线表示释放待机模式的中断请求被响应的情况。
2. 等待时间如下所示：
 - 执行向量中断服务程序时： 11 ~ 13 个时钟周期
 - 不执行向量中断服务程序时： 3 ~ 5 个时钟周期

(b) 由复位信号释放

当产生复位信号时，退出 HALT 模式。然后，和正常复位操作的情况一样，跳转到复位向量指向的地址后，执行程序。

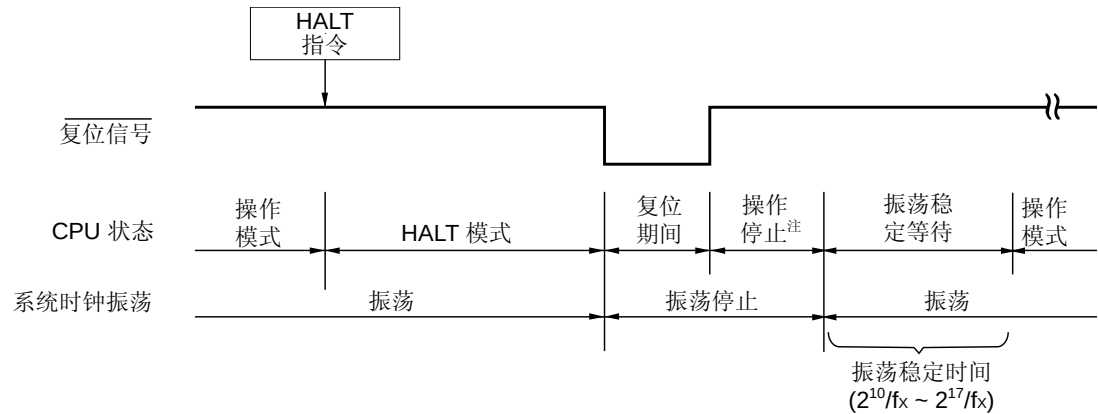
图 14-3. 通过复位信号释放 HALT 模式

(1) 当 CPU 时钟为高速内部振荡时钟或外部输入时钟时



注 操作停止 (277 μ s (最小值), 544 μ s (典型值), 1.075 ms (最大值)), 参见选项字节。

(2) 当 CPU 时钟为晶体/陶瓷振荡时钟



注 操作停止 (276 μ s (最小值), 544 μ s (典型值), 1.074 ms (最大值)), 参见选项字节。

备注 fx: 系统时钟的振荡频率

表 14-3. 在 HALT 模式下中断请求对应的操作

释放源	MK _{xx}	IE	操作
可屏蔽的中断请求	0	0	执行下一个地址的指令
	0	1	执行中断服务程序
	1	×	保持 HALT 模式
产生复位信号	—	×	复位处理

×: 无需考虑

14.2.2 STOP 模式

(1) STOP 模式设置及操作状态

执行 STOP 指令，设置 STOP 模式。

注意事项 因为使用中断请求信号来释放待机模式，如果某一中断源中断请求标志置位且中断屏蔽标志清零，则立即退出待机模式。因此，在 STOP 模式中，执行 STOP 指令后恢复正常操作模式，然后操作会停止 34μs（典型值）（在使用晶体/陶瓷振荡器时，在经过由振荡稳定时间选择寄存器（OSTS）设置的一段额外等待振荡稳定时间后）。

STOP 模式中的操作状态，如下所示。

表 14-4. STOP 模式下的操作状态

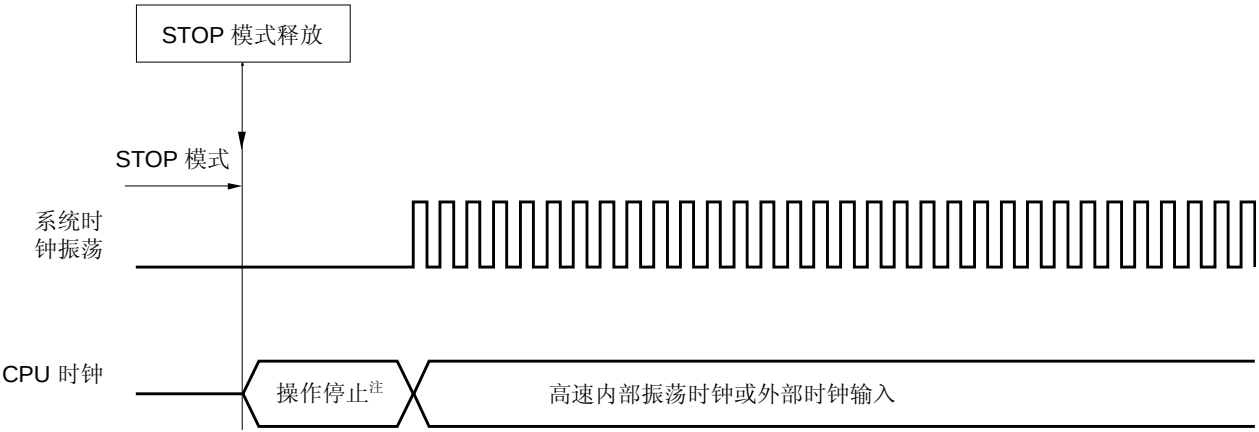
HALT 模式的设置 项目		低速内部振荡器不能被停止 ^注	低速内部振荡器可被停止 ^注	
			低速内部振荡继续	低速内部振荡停止
系统时钟		振荡停止		
CPU		操作停止		
端口（锁存）		保持设置 STOP 模式前的状态		
16 位 定时器/事件计数器 00		操作停止		
8 位 定时器 80		操作停止		
8 位 定时器 H1	设置计数时钟为 $f_{XP} \sim f_{XP}/2^{12}$	操作停止		
	设置计数时钟为 $f_{RL}/2^7$	可以操作	可以操作	操作停止
看门狗定时器	选择“外部硬件时钟”作为操作时钟	禁止设置	操作停止	
	选择“低速内部振荡时钟”作为操作时钟	可以操作（操作继续）	操作停止	
A/D 转换器		操作停止		
串行接口 UART6		操作停止		
POC 电路		一直可以操作		
低电压检测		可以操作		
外部中断		可以操作		

注 选项字节可以选择低速内部振荡器“不能被停止”或“可以通过软件停止”。（关于选项字节，请参见第十八章选项字节）。

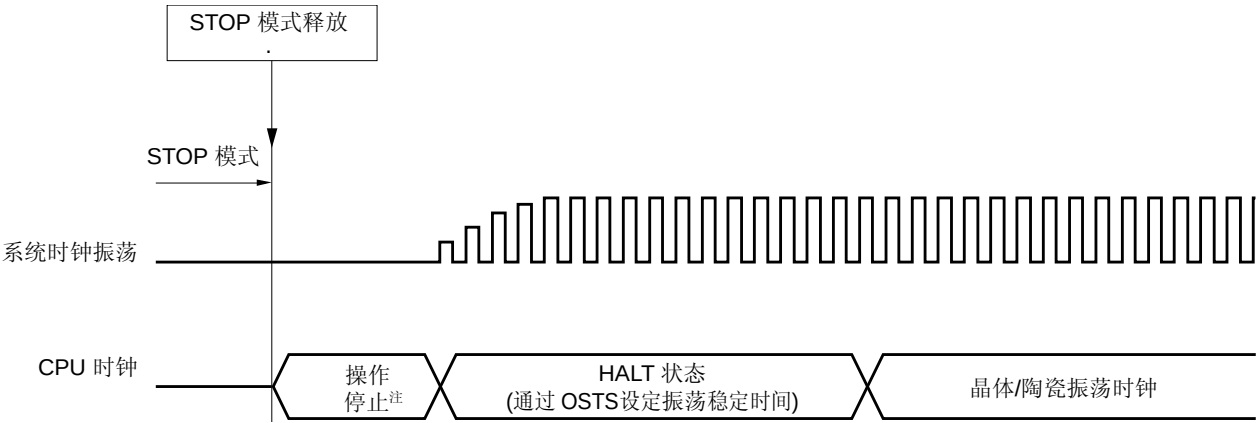
(2) 释放 STOP 模式

图 14-4. 释放 STOP 模式时的操作时序

<1> 如果选择高速内部振荡时钟或外部输入时钟作为系统时钟



<2> 如果选择晶体/陶瓷振荡时钟作为系统时钟



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

STOP 模式可通过以下两种中断源释放。

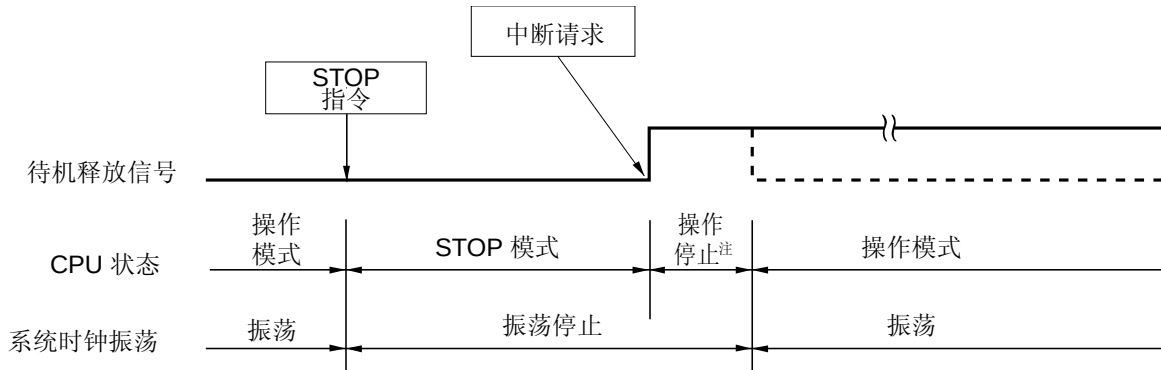
(a) 由未屏蔽的中断请求释放

当产生未屏蔽的中断请求（8 位定时器 H1^注，低电压检测，外部中断请求）时，释放 STOP 模式。经历振荡器稳定时间后，如果允许响应中断，则执行向量中断服务程序。如果禁止响应中断，则执行下一个地址的指令。

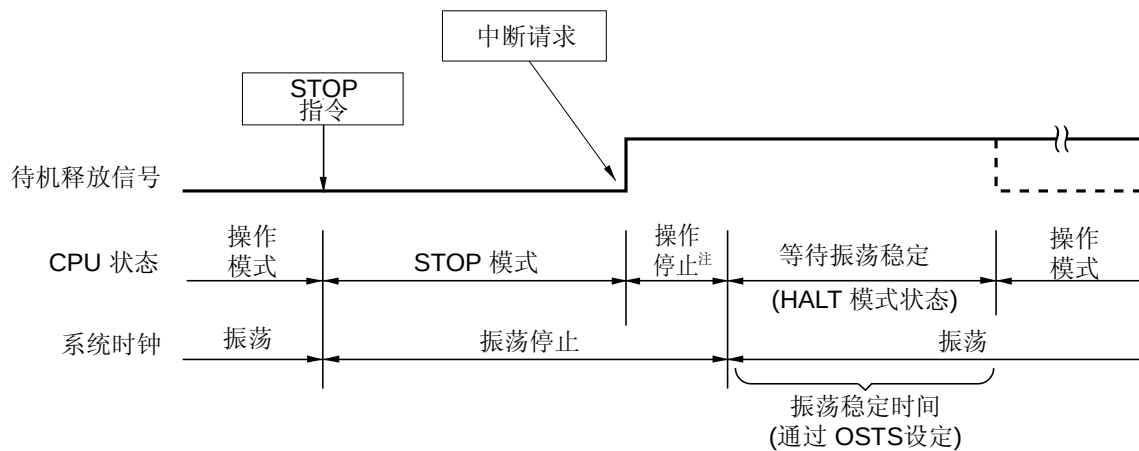
注 仅当计数时钟设置为 $f_{RL}/2^7$

图 14-5. 通过中断请求释放 STOP 模式

(1) 如果 CPU 时钟为高速内部振荡时钟或外部输入时钟



(2) 如果 CPU 时钟为晶体/陶瓷振荡时钟



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

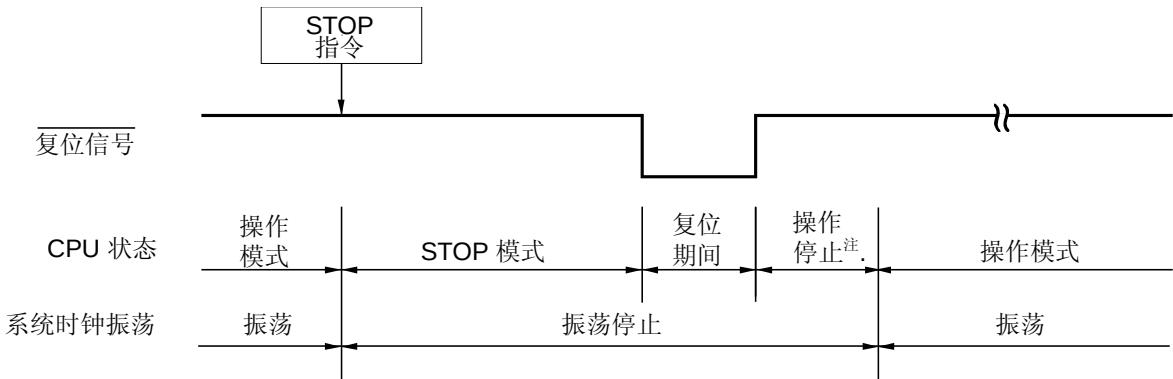
备注 虚线表示释放待机模式的中断请求被响应的情况。

(b) 通过复位信号释放

当产生复位信号时，释放 STOP 模式，并在经过振荡稳定时间后，执行复位操作。

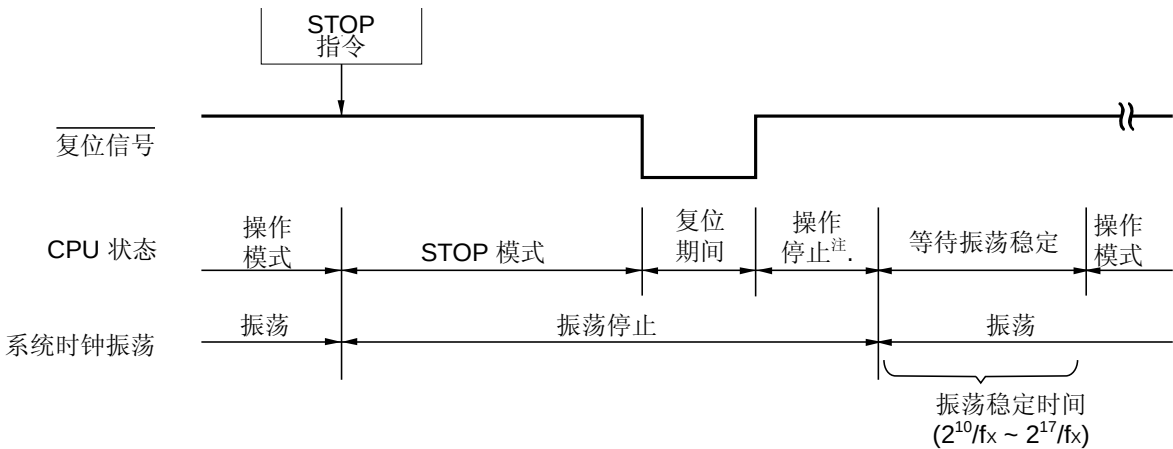
图 14-6. 通过复位信号释放 STOP 模式

(1) 如果 CPU 时钟为高速内部振荡时钟或外部输入时钟



注 操作停止 (277 μ s (最小值), 544 μ s (典型值), 1.075 ms (最大值)), 参见选项字节。

(2) 如果 CPU 时钟为晶体/陶瓷振荡时钟



注 操作停止 (276 μ s (最小值), 544 μ s (典型值), 1.074 ms (最大值)), 参见选项字节。

备注 fx: 系统时钟的振荡频率

表 14-5. 在 STOP 模式下中断请求对应的操作

释放源	MKxx	IE	操作
可屏蔽的中断请求	0	0	执行下一个地址的指令
	0	1	执行中断服务程序
	1	x	保持 STOP 模式
产生复位信号	—	x	复位处理

x: 无需考虑

第十五章 复位功能

有以下 4 种操作可以产生复位信号。

- (1) 由 $\overline{\text{RESET}}$ 引脚输入的外部复位信号。
- (2) 由看门狗定时器溢出引起的内部复位。
- (3) 通过比较上电清零 (POC) 电路的检测电压和供电电压引起的内部复位。
- (4) 通过比较低电压检测器 (LVI) 的检测电压和供电电压引起的内部复位。

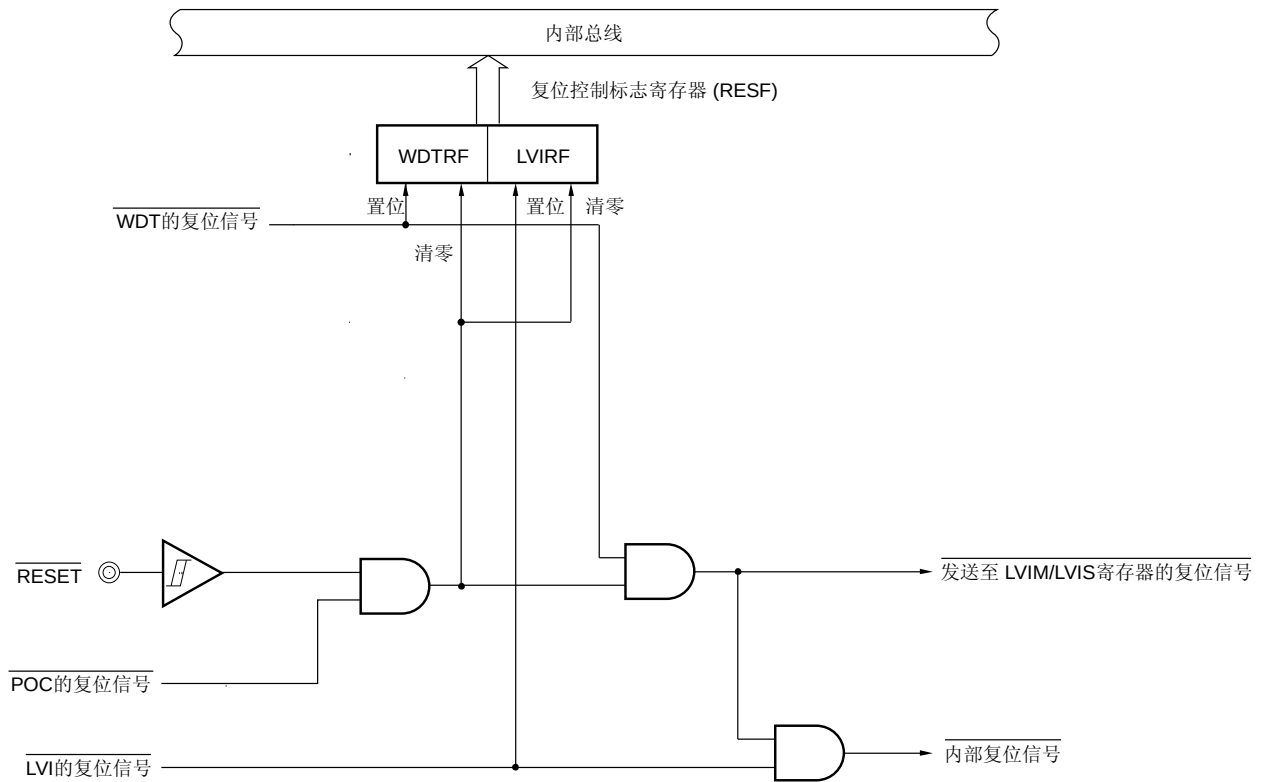
外部复位与内部复位在功能上没有区别。在这两种情况中，当产生复位信号时，程序都是从地址 0000H 和 0001H 处开始执行。

当 $\overline{\text{RESET}}$ 引脚输入为低电平，看门狗定时器溢出，或 POC 和 LVI 电路的电压检测都可以引起复位，硬件的每项状态的设置都会如表 15-1 所示。在复位输入期间或在复位释放后振荡器稳定时间内，每个引脚都是高阻抗，P130 除外，它是低电平输出。

当 $\overline{\text{RESET}}$ 引脚输入低电平时，复位产生。当 $\overline{\text{RESET}}$ 引脚输入高电平时，复位释放，参考选项字节之后，CPU 开始执行程序（如果选择了晶体/陶瓷振荡器，在参考选项字节之后，并经过时钟振荡稳定时间）。由看门狗定时器引起的复位在复位后自动释放，并且 CPU 开始执行程序（如果选择了晶体/陶瓷振荡器，在参考选项字节之后，并经过时钟振荡稳定时间）。（参见图 15-2 到 15-4）。POC 和 LVI 电路供电检测引起的复位，在复位后当 $V_{DD} > V_{POC}$ 或 $V_{DD} > V_{LVI}$ 时自动释放，且 CPU 开始执行程序（如果选择了晶体/陶瓷振荡器，在参考选项字节之后，并经过时钟振荡稳定时间）。（参见第十六章 POC 电路和第十七章 低电压检测）。

- 注意事项**
1. 对于外部复位，输入 $\overline{\text{RESET}}$ 引脚的低电平的时间应为 $2\mu\text{s}$ 或更长。
 2. 在复位信号产生期间，系统时钟和低速内部振荡时钟停止振荡。
 3. 当 $\overline{\text{RESET}}$ 引脚用作只能输入的端口引脚 (P34) 时，由 POC 电路、LVI 电路和看门狗定时器释放复位后，且再次参考选项字节之前，如果有低电平输入到 $\overline{\text{RESET}}$ 引脚，78K0S/KB1+ 复位。保持复位状态直到高电平输入到 $\overline{\text{RESET}}$ 引脚。

图 15-1. 复位功能的框图

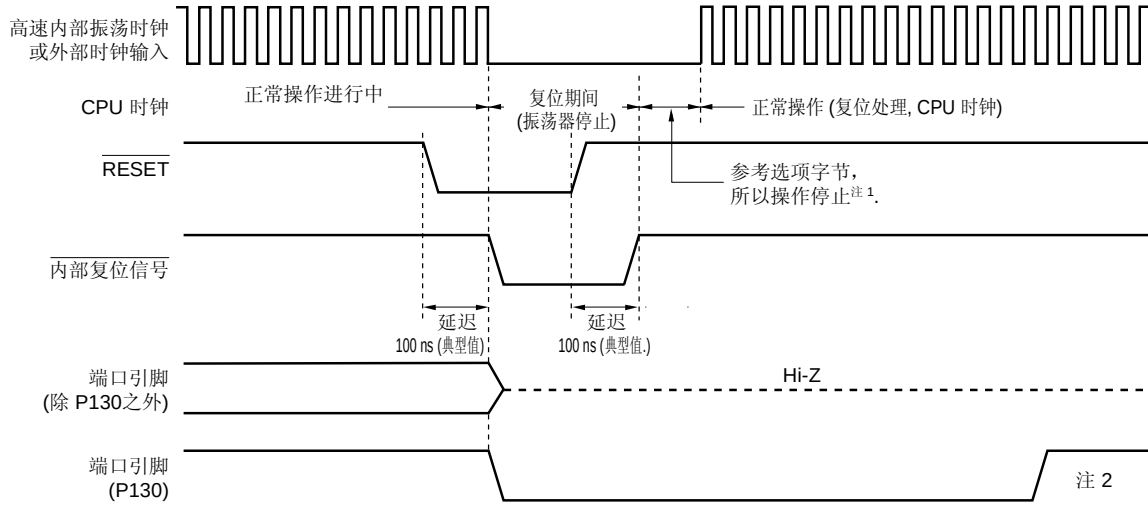


注意事项 LVI 电路的内部复位信号不会使 LVI 电路复位。

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

图 15-2. 由RESET 输入引起的复位时序

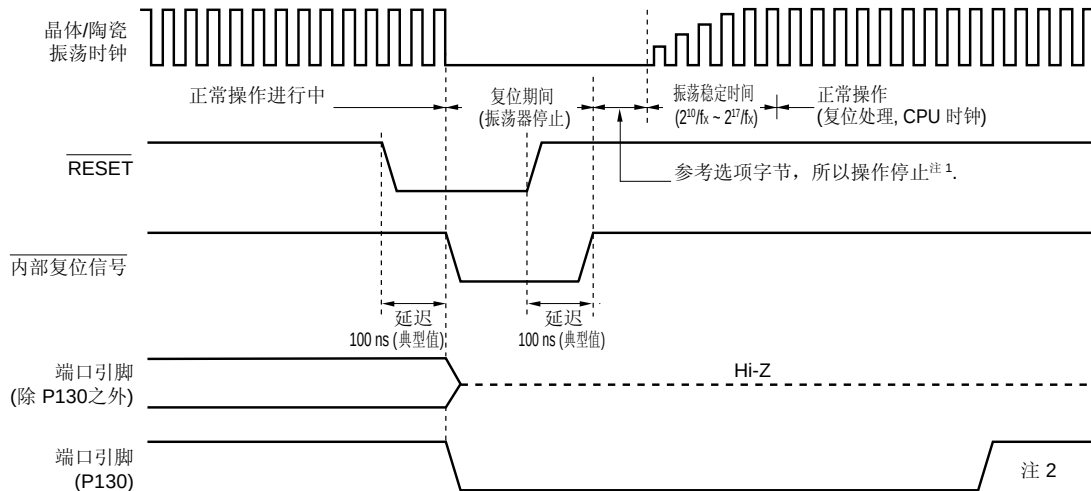
<1>高速内部振荡时钟或外部时钟输入



- 注 1. 操作停止时间为 $277\mu\text{s}$ (最小值), $544\mu\text{s}$ (典型值), 和 1.075 ms (最大值)。
 2. 用软件设置高电平输出。

备注 当复位有效时, P130 输出低电平。如果 P130 在复位有效前设置为输出高电平, P130 的输出信号可以作为虚拟输出, 就象 CPU 的复位信号。

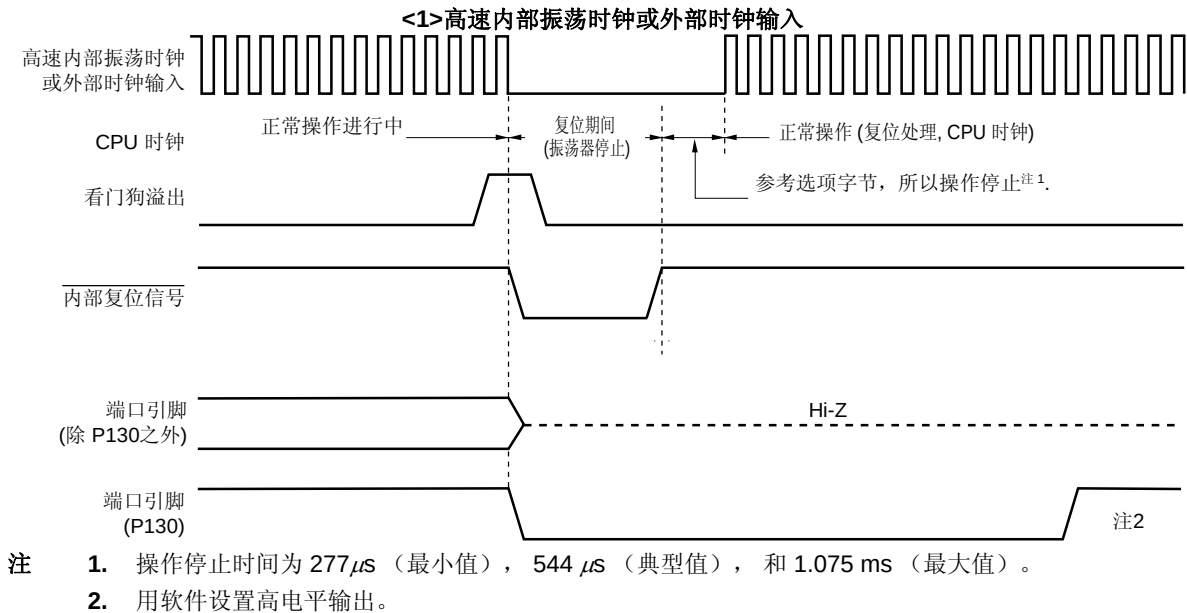
<2> 晶体/陶瓷振荡时钟



- 注 1. 操作停止时间为 $276\mu\text{s}$ (最小值), $544\mu\text{s}$ (典型值), 和 1.074 ms (最大值)。
 2. 用软件设置高电平输出。

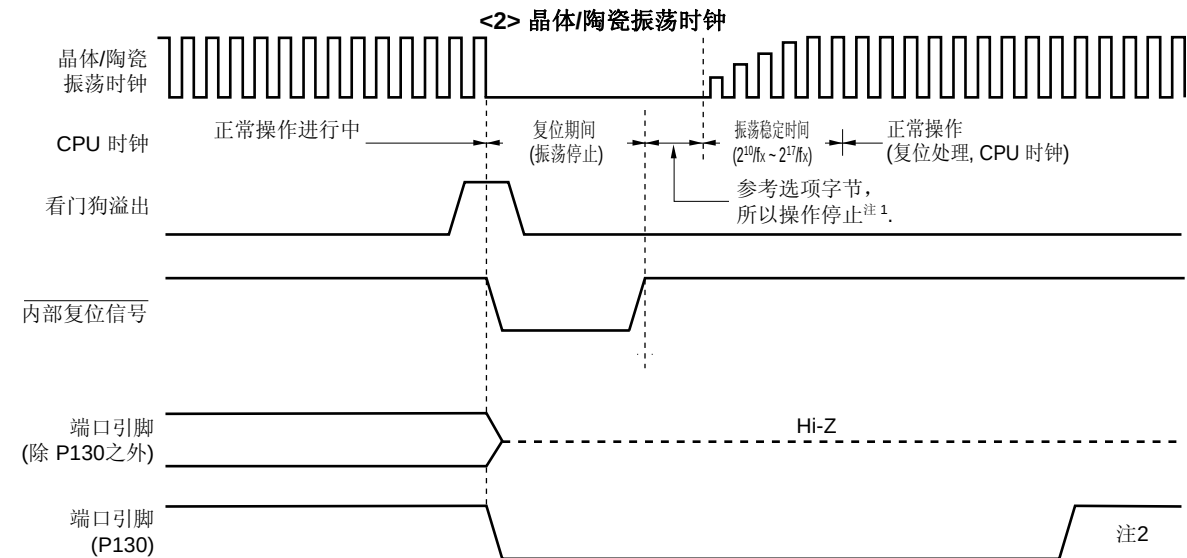
备注 1. f_x : 系统时钟的振荡频率
 2. 当复位有效时, P130 输出低电平。如果 P130 在复位有效前设置为输出高电平, P130 的输出信号可以作为虚拟输出, 就象 CPU 的复位信号。

图 15-3. 由看门狗定时器溢出引起的复位时序



注意事项 在看门狗定时器内部复位的情况下, 看门狗定时器也复位。

备注 当复位有效时, P130 输出低电平。如果 P130 在复位有效前设置为输出高电平, P130 的输出信号可以作为虚拟输出, 就象 CPU 的复位信号。



注 1. 操作停止时间为 $276\mu\text{s}$ (最小值), $544\mu\text{s}$ (典型值), 和 1.074ms (最大值)。

注 2. 用软件设置高电平输出。

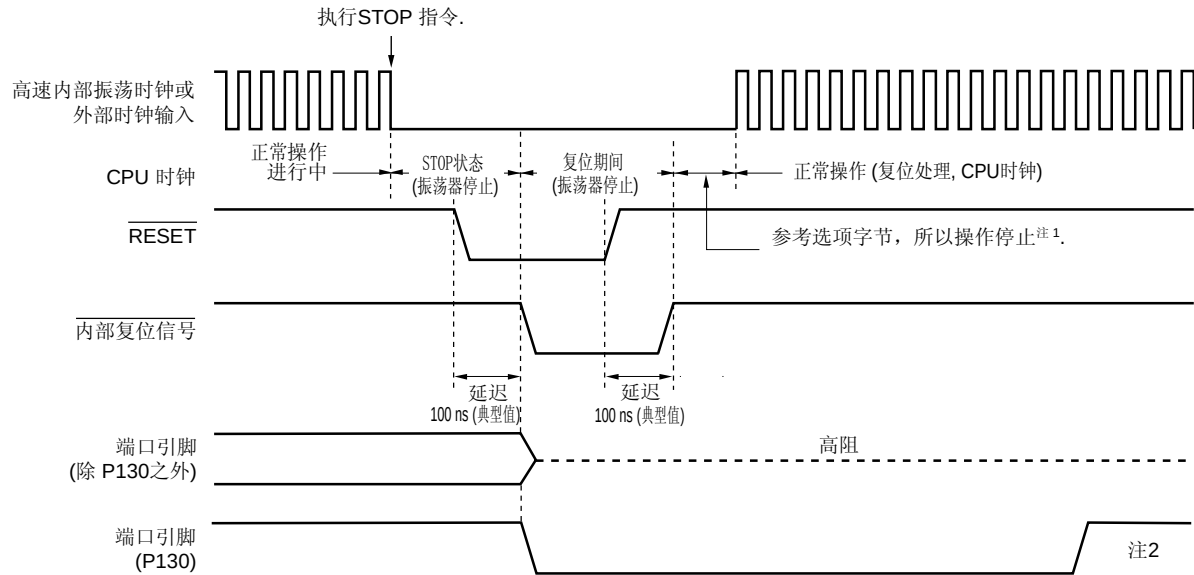
注意事项 在看门狗定时器内部复位的情况下, 看门狗定时器也复位。

备注

1. f_x : 系统时钟的振荡频率
2. 当复位有效时, P130 输出低电平。如果 P130 在复位有效前设置为输出高电平, P130 的输出信号可以作为虚拟输出, 就象 CPU 的复位信号。

图 15-4. 在 STOP 模式下由 RESET 输入引起的复位时序

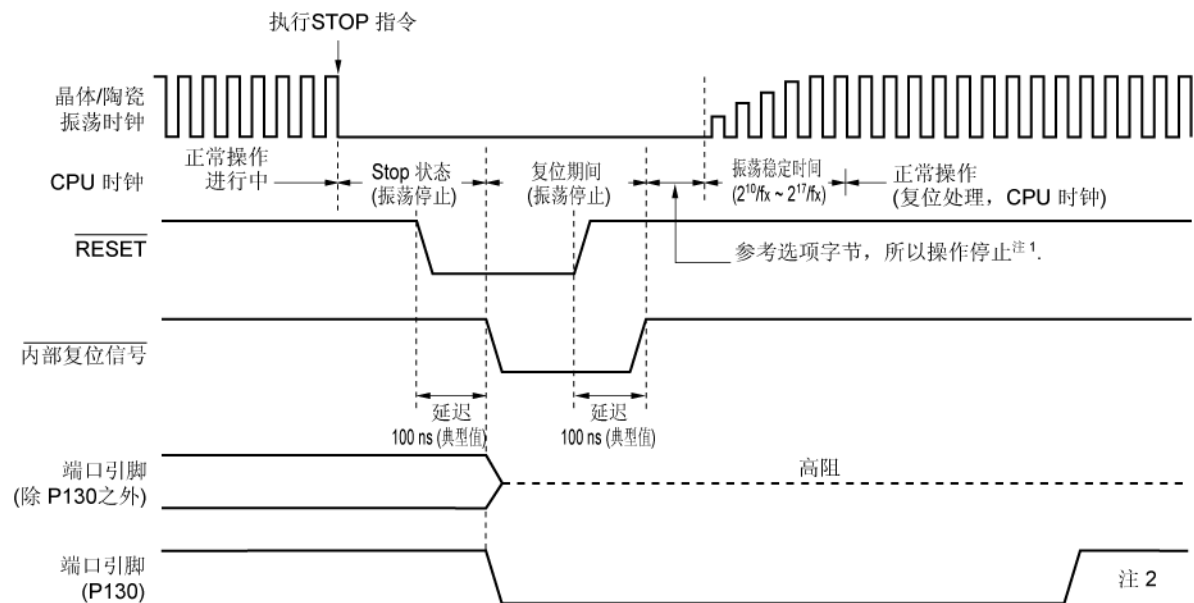
<1> 高速内部振荡时钟或外部时钟输入



- 注 1. 操作停止时间为 $277\mu\text{s}$ (最小值), $544\mu\text{s}$ (典型值), 和 1.075 ms (最大值)。
2. 用软件设置高电平输出。

备注 当复位有效时, P130 输出低电平。如果 P130 在复位有效前设置为输出高电平, P130 的输出信号可以作为虚拟输出, 就象 CPU 的复位信号。

<2> 晶体/陶瓷振荡时钟



- 注 1. 操作停止时间为 $276\mu\text{s}$ (最小值), $544\mu\text{s}$ (典型值), 和 1.074 ms (最大值)。
2. 用软件设置高电平输出。

- 备注
1. POC 电路和低电压检测的复位时序，参见第十六章 POC 电路和第十七章 低电压检测。
 2. f_x : 系统时钟的振荡频率
 3. 当复位有效时，P130 输出低电平。如果 P130 在复位有效前设置为输出高电平，P130 的输出信号可以作为虚拟输出，就象 CPU 的复位信号。

表 15-1. 响应复位后的硬件状态 (1/2)

硬件		复位后的状态
程序计数器 (PC) ^{注 1}		设置复位向量表的内容 (0000H 和 0001H)
堆栈指针 (SP)		不确定
程序状态字 (PSW)		02H
RAM	数据存储器	不确定 ^{注 2}
	通用寄存器	不确定 ^{注 2}
端口 (P0, P2 ~ P4, P12, P13) (输出锁存)		00H
端口模式寄存器 (PM0, PM2 ~ PM4, PM12)		FFH
端口模式控制寄存器 (PMC2)		00H
上拉电阻选择寄存器 (PU0, PU2, PU3, PU4, PU12)		00H
处理器时钟控制寄存器 (PCC)		02H
预处理器时钟控制寄存器 (PPCC)		02H
低速内部振荡模式寄存器 (LSRCM)		00H
振荡稳定时间选择寄存器 OSTs		不确定
16 位定时器 00	定时器计数器 00 (TM00)	0000H
	捕捉/比较寄存器 000, 010 (CR000, CR010)	0000H
	模式控制寄存器 00 (TMC00)	00H
	预分频模式寄存器 00 (PRM00)	00H
	捕捉/比较控制寄存器 00 (CRC00)	00H
	定时器输出控制寄存器 00 (TOC00)	00H
8 位定时器 80	定时器计数器 80 (TM80)	00H
	比较寄存器 (CR80)	不确定
	模式控制寄存器 80 (TMC80)	00H
8 位定时器 H1	比较寄存器 (CMP01, CMP11)	00H
	模式寄存器 1 (TMHMD1)	00H
看门狗定时器	模式寄存器 (WDTM)	67H
	允许寄存器 (WDTE)	9AH
A/D 转换器	转换结果寄存器 (ADCR, ADCRH)	不确定
	模式寄存器 (ADM)	00H
	模拟输入通道选择寄存器 (ADS)	00H

- 注
1. 只有在复位信号产生和振荡稳定时间内，PC 的内容是不确定的。其他硬件单元的状态保持不变。
 2. 在待机模式中，保持复位后的状态。

表 15-1. 响应复位后的硬件状态 (2/2)

硬件		复位后的状态
串行接口 UART6	接收缓冲寄存器 6 (RXB6)	FFH
	发送缓冲寄存器 6 (TXB6)	FFH
	异步串行接口操作模式寄存器 6 (ASIM6)	01H
	异步串行接口接收错误状态寄存器 6 (ASIS6)	00H
	异步串行接口发送错误状态寄存器 6 (ASIF6)	00H
	时钟选择寄存器 6 (CKSR6)	00H
	波特率发生器控制寄存器 6 (BRGC6)	FFH
	异步串行接口控制寄存器 6 (ASICL6)	16H
	输入切换控制寄存器 (ISC)	00H
乘法器	16 位乘法结果存储器寄存器 (MUL0)	不确定
	数据寄存器 (MRA0, MRB0)	不确定
	控制寄存器 (MULC0)	00H
复位功能	复位控制标志寄存器 (RESF)	00H ^注
低电压检测	低电压检测寄存器 (LVIM)	00H ^注
	低电压检测电平选择寄存器 (LVIS)	00H ^注
中断	中断请求标志寄存器 (IF0, IF1)	00H
	中断屏蔽标志寄存器 (MK0, MK1)	FFH
	外部中断模式寄存器 (INTM0, INTM1)	00H
Flash 存储器	Flash 保护命令寄存器 (PFCMD)	不确定
	Flash 状态寄存器 (PFS)	00H
	Flash 编程模式控制寄存器 (FLPMC)	不确定
	FLASH 编程命令寄存器 (FLCMD)	00H
	Flash 地址指针 L (FLAPL)	不确定
	Flash 地址指针 H (FLAPH)	
	Flash 地址指针 H 比较寄存器 (FLAPHC)	00H
	Flash 地址指针 L 比较寄存器 (FLAPLC)	00H
	Flash 写缓冲寄存器 (FLW)	00H

注 根据复位源的不同, 这些值会有如下改变。

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

15.1 确认复位源的寄存器

78K0S/KB1+中有许多内部复位源。复位控制标志寄存器（RESF）用于存储产生复位请求的复位源。
RESF 的值可由 8 位存储器操作指令读取。
由 $\overline{\text{RESET}}$ 输入、由 POC 电路引起的复位输入，以及读取 RESF 产生的复位信号，都会将 RESF 清为 00H。

图 15-5. 复位控制标志寄存器（RESF）的格式

地址： FF54H	复位后： 00H ^注	读						
符号	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LVIRF

WDTRF	由看门狗定时器产生的内部复位请求（WDT）
0	不产生内部复位请求，或 RESF 清零
1	产生内部复位请求

LVIRF	由低电压检测（LVI）产生的内部复位请求
0	不产生内部复位请求，或 RESF 清零
1	产生内部复位请求

注 复位后的值会根据复位源而不同。

注意事项 不能通过 1 位存储器操作指令读取数据。

产生复位请求时的 RESF 状态如表 15-2 所示。

表 15-2. 产生复位请求时的 RESF 状态

复位源 标志	$\overline{\text{RESET}}$ 输入	由 POC 复位	由 WDT 复位	由 LVI 复位
WDTRF	清零（0）	清零（0）	置位（1）	保持
LVIRF			保持	置位（1）

第十六章 上电清零电路

16.1 上电清零电路的功能

上电清零电路（POC）有如下功能。

- 在上电时产生内部复位信号
- 比较电源电压（ V_{DD} ）和检测电压（ $V_{POC} = 2.1\text{ V}$ （典型值）），并在 $V_{DD} < V_{POC}$ 时产生内部复位信号
- 比较电源电压（ V_{DD} ）和检测电压（ $V_{POC} = 2.1\text{ V}$ （典型值）），并在 $V_{DD} \geq V_{POC}$ 时释放内部复位信号

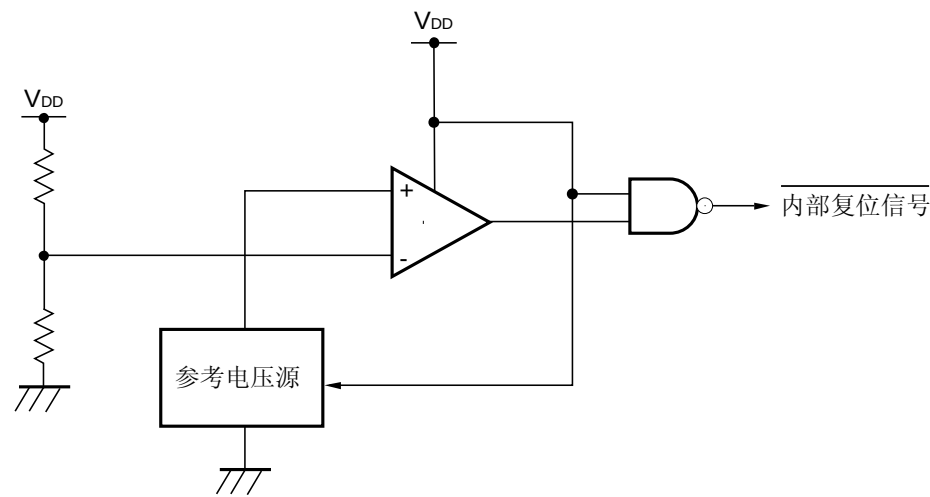
注意事项 1. 如果在 POC 电路中产生内部复位信号，则将复位控制标志寄存器（RESF）清零。
2. 这些产品应该在如下电压范围内使用，因为 POC 电路检测电压（ V_{POC} ）决定供电电压范围。
标准产品，（A）等级产品：2.2 ~ 5.5 V（（A2）等级产品）2.26 ~ 5.5 V

备注 本产品集成了多种可以产生内部复位信号的硬件功能。当看门狗定时器（WDT）和低电压检测电路（LVI）引起复位时，复位控制标志寄存器（RESF）中有一个标志用于指示复位原因。当内部复位信号由 WDT 或 LVI 产生时，RESF 不能被清零，且标志位被置 1。
RESF 的详细内容，请参见第十五章 复位功能。

16.2 上电清零电路的配置

POC 电路的框图如图 16-1 所示

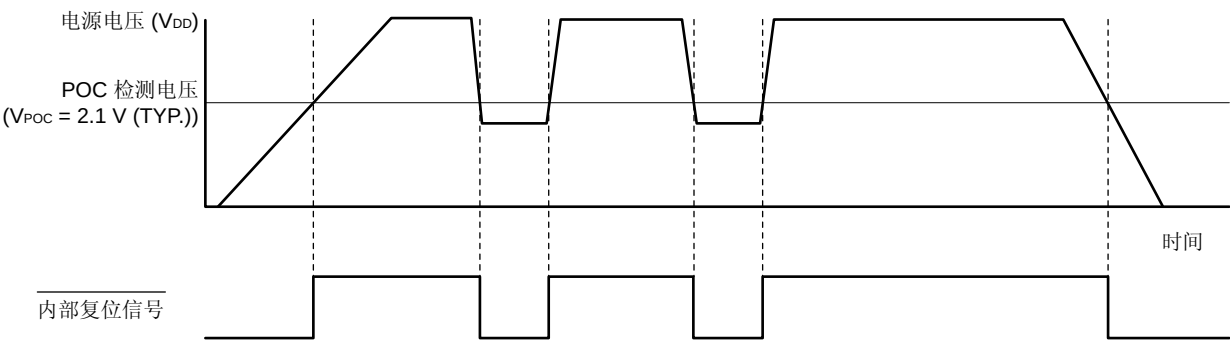
图 16-1. POC 电路的框图



16.3 上电清零电路的操作

在 POC 电路中，比较电源电压 (V_{DD}) 和检测电压 ($V_{POC} = 2.1\text{ V}$ (典型值))，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。

图 16-2. POC 电路中产生内部复位信号的时序



16.4 上电清零电路的使用注意事项

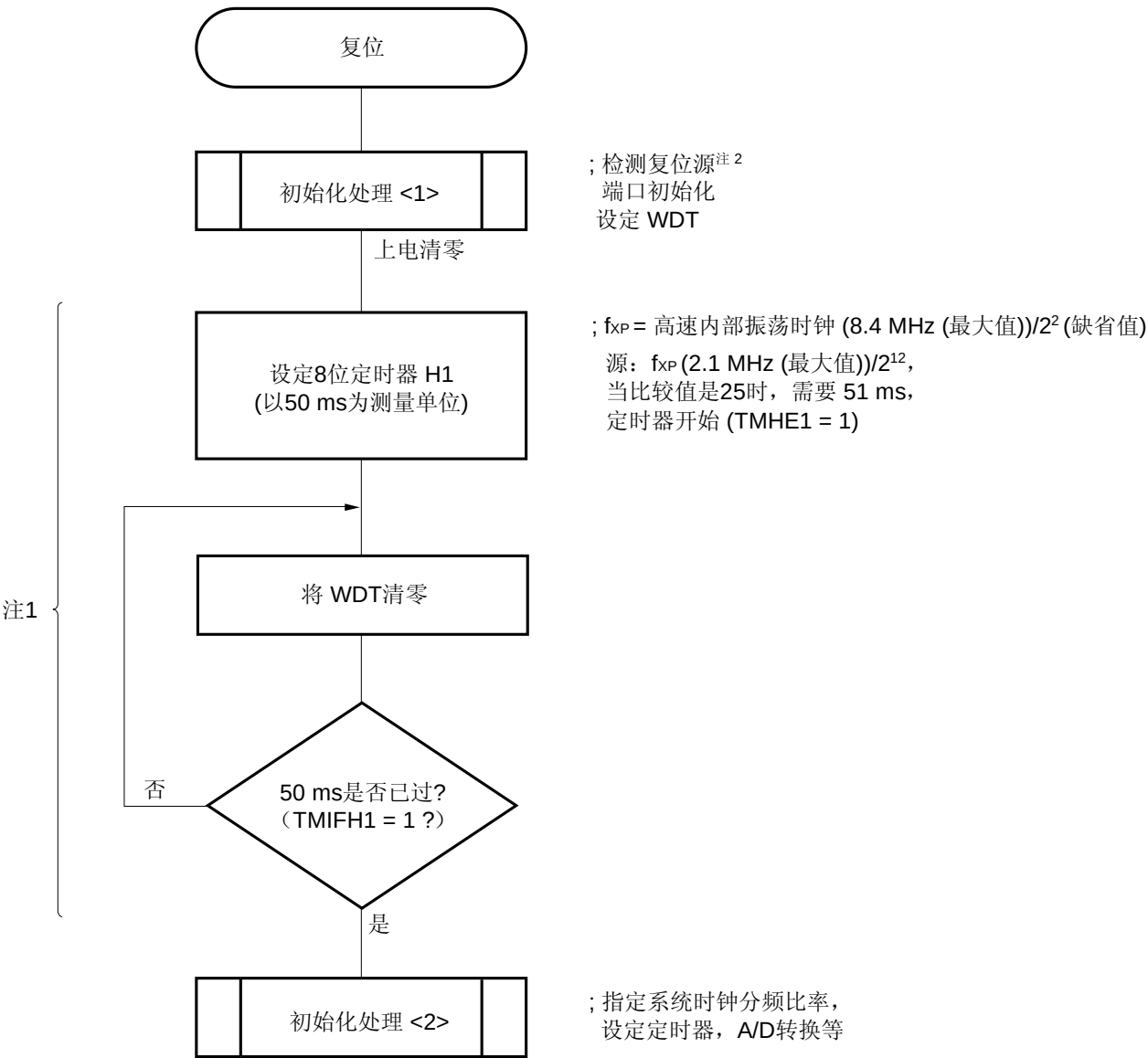
对于一个系统，如果电源电压（V_{DD}）某段时间内在 POC 检测电压（V_{POC}）附近波动，该系统可能会反复进行复位和复位状态的释放。在这种情况下，可采用下列方法任意设置从复位释放到微控制器的启动操作所经历的时间。

<动作>

在释放复位信号后，通过使用一个定时器的软件计数器来等待系统电源电压的波动期，然后对端口初始化。

图 16-3. 复位释放后软件处理过程示例（1/2）

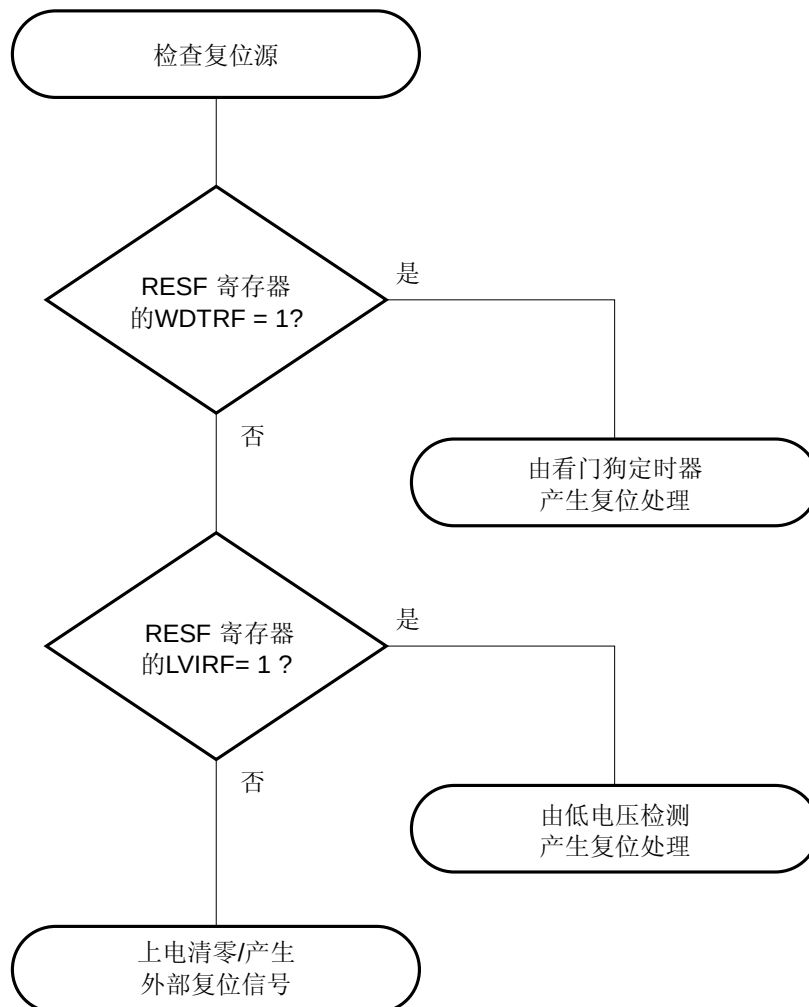
- 如果在电源电压接近 POC 检测电压时，波动时间小于等于 50ms



注 1. 如果在此期间再次产生复位，则不启动初始化处理<2>。
2. 流程图如下页所示

图 16-3. 复位释放后软件处理过程示例（2/2）

- 检查复位原因



第十七章 低电压检测器

17.1 低电压检测器的功能

低电压检测器（LVI）有以下功能。

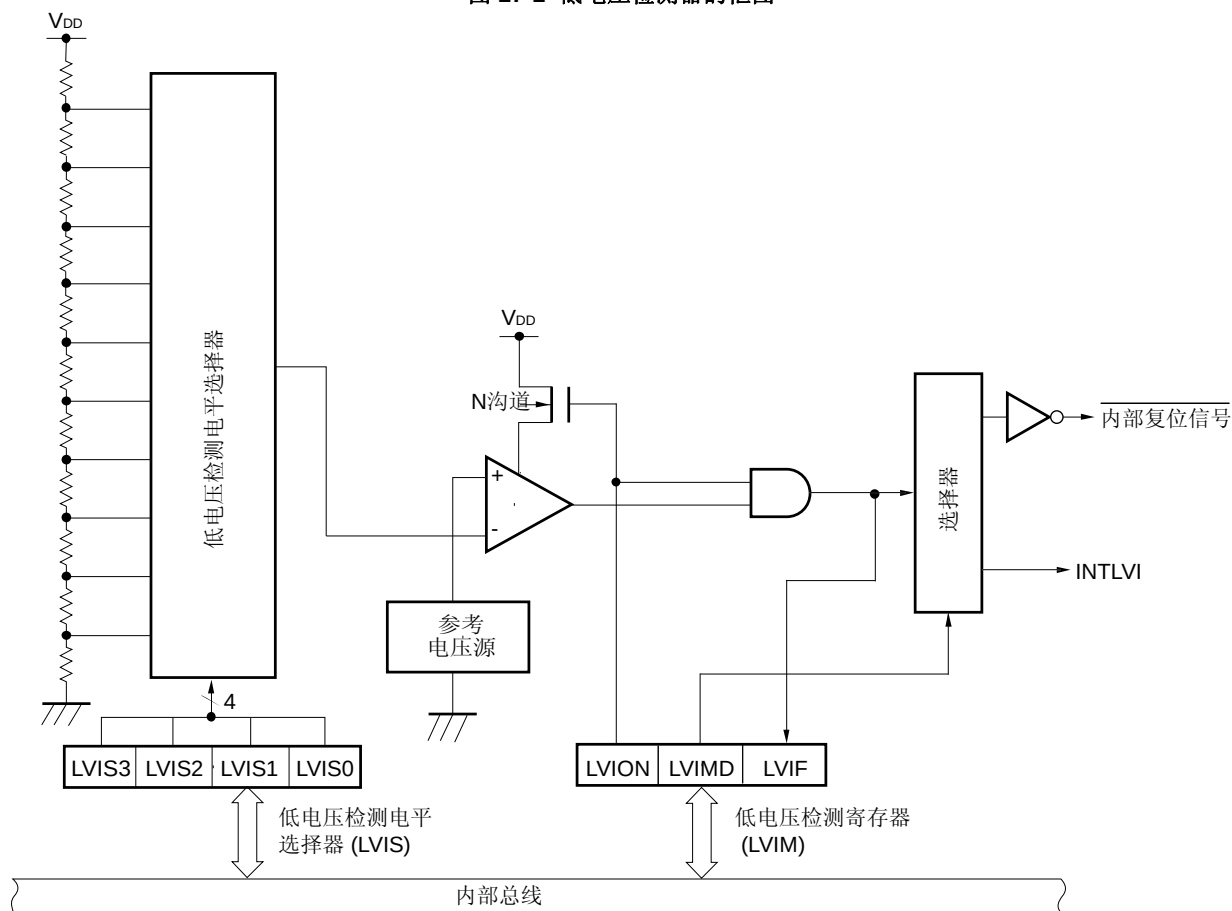
- 比较电源电压（ V_{DD} ）和检测电压（ V_{LVI} ），当 $V_{DD} < V_{LVI}$ 时，产生内部中断信号或内部复位信号
- 可由软件改变的电源电压检测电平（十个电平）
- 可由软件选择中断或复位功能
- 在 STOP 模式下可以操作

低电压检测器用作复位时，如果产生复位，则复位控制标志寄存器（RESF）的第 0 位（LVIRF）被设置为 1。RESF 的详细内容，请参见第十五章 复位功能。

17.2 低电压检测器的配置

低电压检测器的框图如下图 17-1 所示。

图 17-1 低电压检测器的框图



17.3 低电压检测器的控制寄存器

低电压检测器由以下寄存器控制

- 低电压检测寄存器（LVIM）
- 低电压检测电平选择寄存器（LVIS）

(1) 低电压检测寄存器（LVIM）

该寄存器用于设置低电压检测的方式和操作模式。
该寄存器可以通过 1 位或 8 位存储操作指令进行设置。
复位输入会将该寄存器清零^{注 1}

图 17-2. 低电压检测寄存器（LVIM）的格式

地址: FF50H 复位后: 00H^{注 1} 读/写^{注 2}

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

LVION ^{注 3}	允许低电压检测操作
0	禁止操作
1	允许操作

LVIMD	低电压检测操作模式的选择
0	当电源电压（V _{DD} ）< 检测电压（V _{LVI} ）时，产生中断信号
1	当电源电压（V _{DD} ）< 检测电压（V _{LVI} ）时，产生内部复位信号

LVIF ^{注 4}	低电压检测标志
0	当电源电压（V _{DD} ）≥检测电压（V _{LVI} ）时，或禁止操作时
1	电源电压（V _{DD} ）<检测电压（V _{LVI} ）

- 注
1. 只有在 LVI 产生的复位后保持。
 2. 第 0 位是只读位。
 3. 当 LVION 被置 1 时，LVI 电路中的比较器开始工作。从设置 LVION 为 1 到 LVIF 确认电压，至少经过 0.2ms，使用软件进行等待。
 4. 当 LVION = 1 且 LVIMD = 0 时，LVIF 的值作为中断请求信号 INTLVI 输出。

- 注意事项
1. 要停止 LVI，可执行以下任一过程。
 - 当使用 8 位存储器操作指令时：写 00H 到 LVIM
 - 当使用 1 位存储器操作指令时：对 LVION 清零
 2. 确保第 2~6 位清零

(2) 低电压检测电平选择寄存器 (LVIS)

该寄存器用于选择低电压检测电平。
该寄存器可由 8 位存储器操作指令设置。
复位输入会将该寄存器清零^注

图 17-3. 低电压检测电平选择寄存器 (LVIS) 的格式

地址: FF51H, 复位后: 00H^注 读/写

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

LVIS3	LVIS2	LVIS1	LVIS0	检测电平
0	0	0	0	V _{LVI0} (4.3 V ± 0.2 V)
0	0	0	1	V _{LVI1} (4.1 V ± 0.2 V)
0	0	1	0	V _{LVI2} (3.9 V ± 0.2 V)
0	0	1	1	V _{LVI3} (3.7 V ± 0.2 V)
0	1	0	0	V _{LVI4} (3.5 V ± 0.2 V)
0	1	0	1	V _{LVI5} (3.3 V ± 0.15 V)
0	1	1	0	V _{LVI6} (3.1 V ± 0.15 V)
0	1	1	1	V _{LVI7} (2.85 V ± 0.15 V)
1	0	0	0	V _{LVI8} (2.6 V ± 0.1 V)
1	0	0	1	V _{LVI9} (2.35 V ± 0.1 V)
其他状态				禁止设置

注 只有在 LVI 产生的复位后保持。

- 注意事项
1. 确保第 4~7 位清零。
 2. 在 LVI 工作期间, 如果写入不同的值, 该值在写入的时刻将变得不确定, 因此确保在写入前停止 LVI 的操作 (LVIM 寄存器的第 7 位 (LVION) = 0)。

17.4 低电压检测器的操作

低电压检测器可用于以下两种模式

- 用于复位
比较电源电压 (V_{DD}) 和检测电压 (V_{LVI})，当 $V_{DD} < V_{LVI}$ 产生内部复位信号，当 $V_{DD} \geq V_{LVI}$ 释放内部复位信号。
- 用于中断
比较电源电压 (V_{DD}) 和检测电压 (V_{LVI})，当 $V_{DD} < V_{LVI}$ 时，产生中断信号 (INTLVI)。

操作设置如下

(1) 用作复位

- 当启动操作时
 - <1> 屏蔽 LVI 中断 ($LVIMK = 1$)
 - <2> 使用低电压检测电平选择寄存器 (LVIS) 的第 3~0 位 ($LVIS3 \sim LVIS0$) 设置检测电压。
 - <3> 设置低电压检测寄存器 (LVIM) 的第 7 位 ($LVION$) 为 1 (允许 LVI 操作)。
 - <4> 使用软件等待至少 0.2 ms 的时间
 - <5> 根据 LVIM 的第 0 位 ($LVIF$) 确定“电源电压 (V_{DD}) $\geq$ 检测电压 (V_{LVI})”时，等待过程结束。
 - <6> 设置 LVIM 的第 1 位 ($LVIMD$) 为 1。(当电源电压 (V_{DD}) $<$ 检测电压 (V_{LVI}) 时，产生内部复位信号)

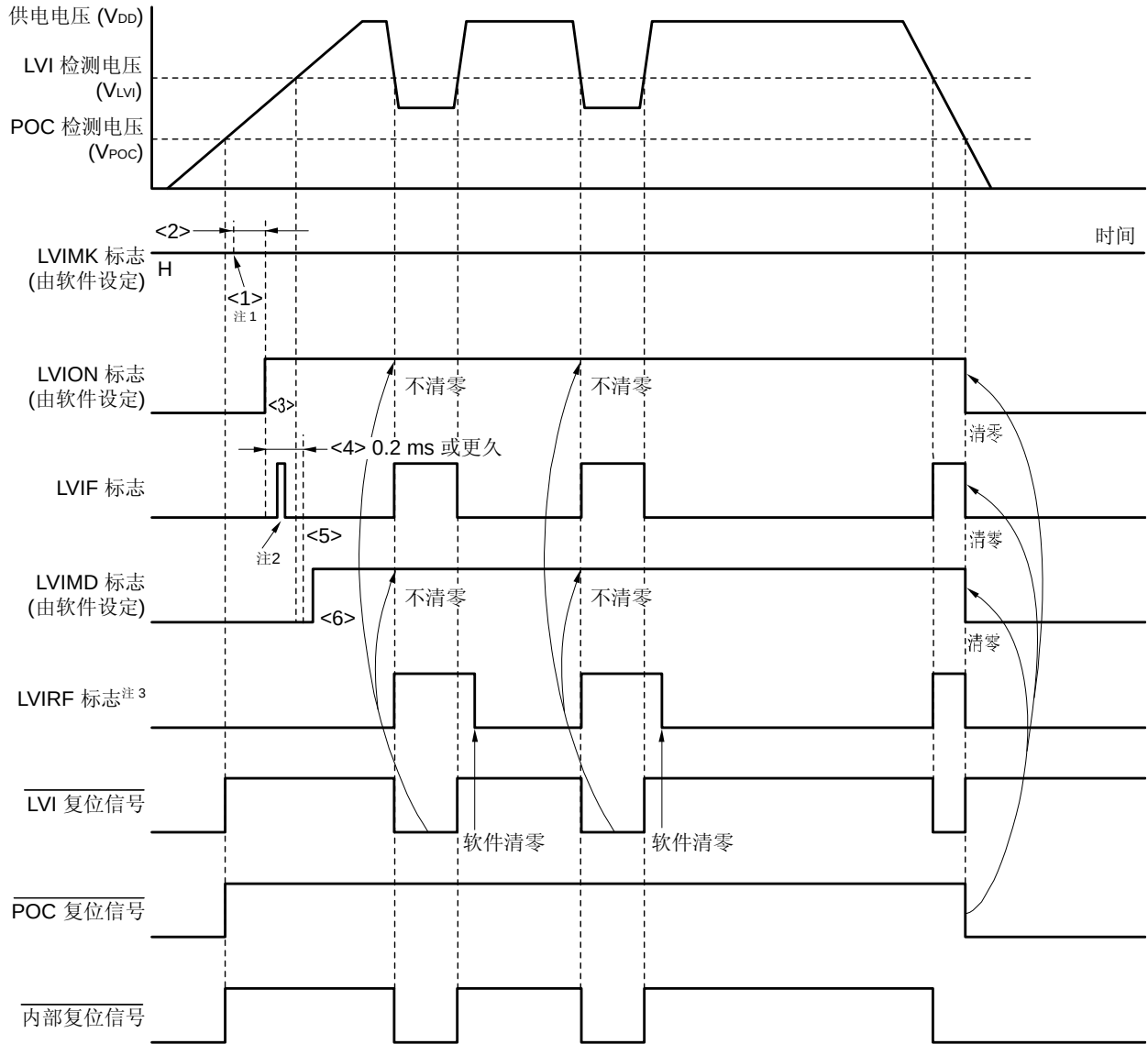
图 17-4 显示了由低电压检测器产生内部复位信号的时序。该时序图中的<1>~<6>对应上面的步骤<1>~<6>。

注意事项

1. 步骤<1> 必须执行。当 $LVIMK = 0$ 时，在步骤<3>之后，可能会立即产生中断。
2. 当 LVIM 被设置为 1 时，如果电源电压 (V_{DD}) $\geq$ 检测电压 (V_{LVI})，则不会产生内部复位信号。

- 要停止 LVI，可执行以下任一过程
 - 当使用 8 位存储器操作指令时： 写 00H 到 LVIM。
 - 当使用 1 位存储器操作指令时： 依次将 LVIMD 和 LVION 清零。

图 17-4. 低电压检测器内部复位信号产生时序



- 注
1. RESET 输入将 LVIMK 标志设置为“1”。
 2. LVIF 标志可以设置为 (1)。
 3. LVIRF 是复位控制标志寄存器 (RESF) 的第 0 位。RESF 的详细内容, 请参见第十五章 复位功能。

备注 图 17-4 中的<1>~<6>对应 17.4 (1) 用作复位 中“当启动操作时”描述的步骤<1>~<6>。

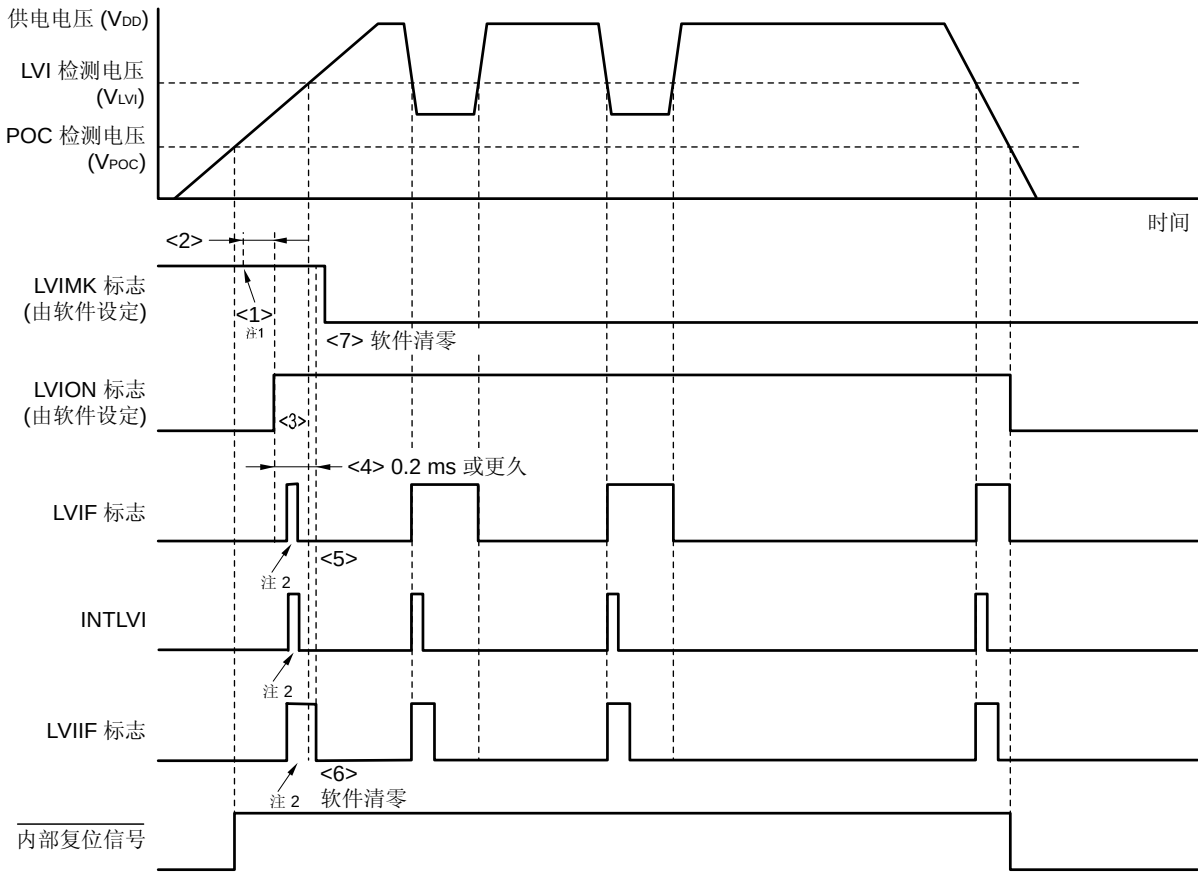
(2) 用作中断

- 当启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)
 - <2> 使用低电压检测电平选择寄存器 (LVIS) 的第 3~0 位 (LVIS3~LVIS0) 设置检测电压。
 - <3> 设置低电压检测寄存器 (LVIM) 的第 7 位 (LVION) 为 1 (允许 LVI 操作)。
 - <4> 使用软件等待至少 0.2 ms 的时间
 - <5> 根据 LVIM 的第 0 位 (LVIF) 确定电源电压 (V_{DD}) $\geq$ 检测电压 (V_{LVI}) 时, 等待过程结束。”
 - <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 (当使用向量中断时)

图 17-5 显示了由低电压检测器产生中断信号的时序。该时序图中的<1>~<7>对应上面的步骤<1>~<7>。

- 要停止 LVI, 可执行以下任一过程
 - 当使用 8 位存储器操作指令时: 写 00H 到 LVIM
 - 当使用 1 位存储器操作指令时: 对 LVION 清零

图 17-5. 低电压检测器中断信号产生时序



- 注
- 1. RESET 输入将 LVIMK 标志设置为“1”。
 - 2. 可以产生中断请求信号 (INTLVI), LVIF 和 LVIIF 标志可能设置为 1。

备注 图 17-5 中的<1>~<7>对应 17.4 (2) 用作中断中 “当启动操作时” 描述的步骤<1>~<7>。

17.5 低电压检测器的注意事项

在系统中，如果电源电压（VDD）某段时间内在 LVI 检测电压（VLVI）附近波动，可根据低电压检测器的使用情况进行如下操作。

<1> 用作复位

系统可能会反复进行复位和复位状态的释放。

在这种情况下，采用下面的动作（1）来任意设置从复位释放到微控制器启动操作所需要的时间。

<2> 用作中断

可能会频繁产生中断请求。采用动作（2）的（b）。

在该系统中，可采用如下动作。

<动作>

（1） 用作复位

在复位信号释放后，通过使用定时器的软件计数器等待电源电压的波动期，然后对端口初始化（参见图 17-6）。

（2） 用作中断

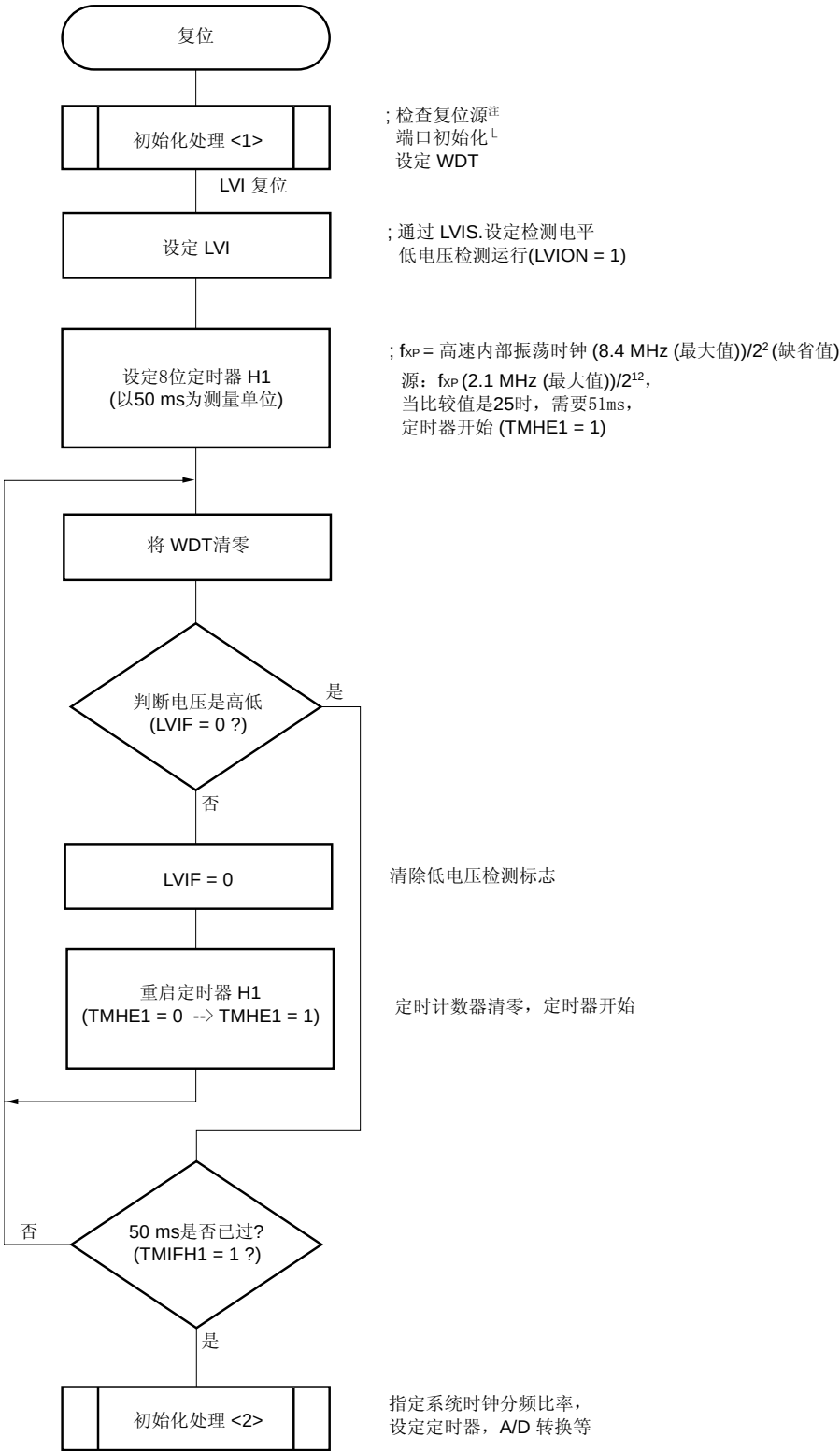
（a）为低电压检测执行处理过程^注。通过低电压检测寄存器（LVIM）的第 0 位（LVIF），在 LVI 中断的服务程序中检查“电源电压（VDD）≥ 检测电压（VLVI）”，将中断请求标志寄存器（IF0）的第 1 位（LVIIF）清零。

（b）在系统中，当电源电压在 LVI 检测电压附近的波动时间很长时，则要进行等待电源电压波动期，通过 LVIF 标志检查“电源电压（VDD）≥ 检测电压（VLVI）”的情况，然后清 LVIIF 标志为零。

注 对于低电压检测过程，CPU 时钟速度切换至低速，并且 A/D 转换器停止，等。

图 17-6. 复位释放后软件处理过程示例（1/2）

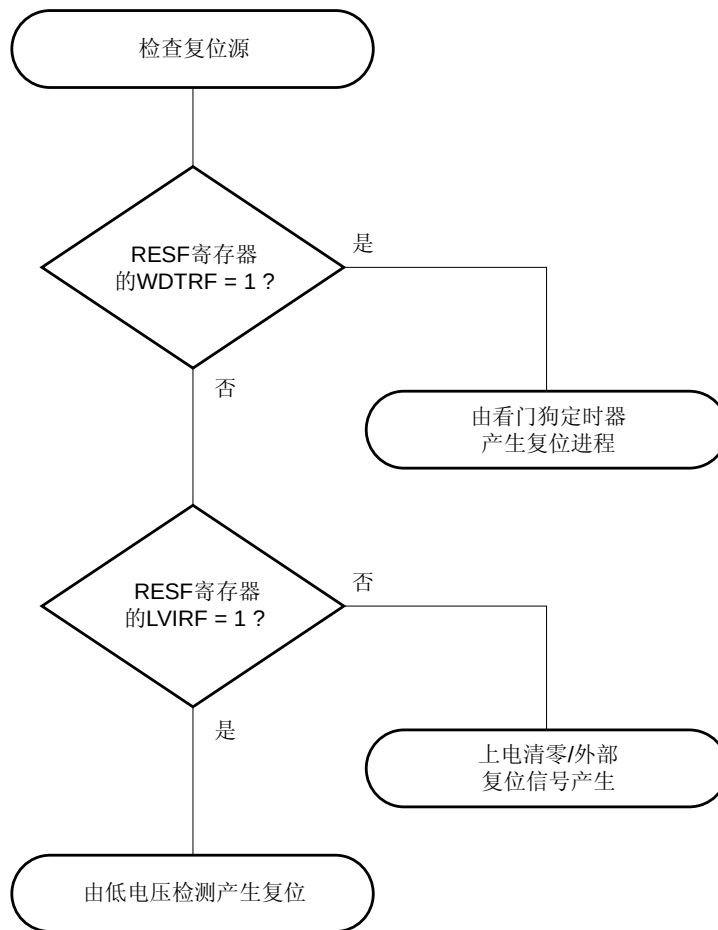
- 如果电源电压在 LVI 检测电压附近的波动时间小于等于 50 ms



注 流程图如下页所示

图 17-6. 复位释放后软件处理过程示例（2/2）

• 检查复位源

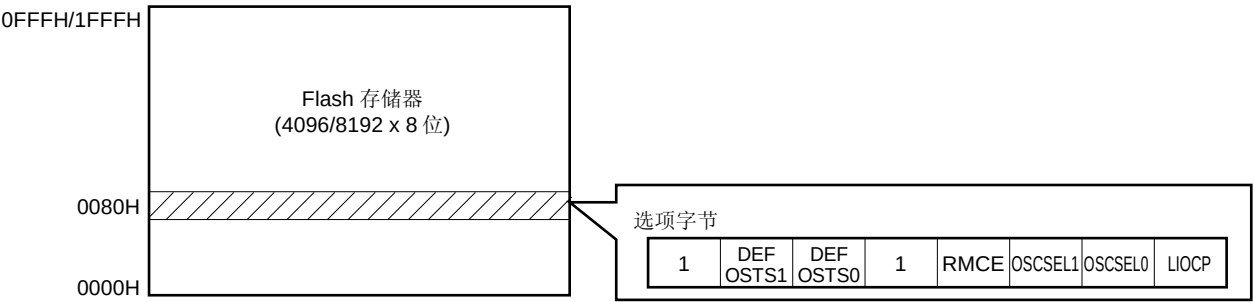


18.1 选项字节的功能

78K0S/KB1+ 的 Flash 存储器地址 0080H，是选项字节区域。当上电或复位后启动时，选项字节自动被参考，并且执行设定的指定功能。使用该产品时，确保使用选项字节设置如下功能。

- (1) 系统时钟源的选择
 - 高速内部振荡时钟
 - 晶体/陶瓷振荡时钟
 - 外部时钟输入
- (2) 低速内部振荡时钟的振荡
 - 不能被停止
 - 可以通过软件停止
- (3) $\overline{\text{RESET}}$ 引脚的控制
 - 用作 $\overline{\text{RESET}}$ 引脚
 - $\overline{\text{RESET}}$ 引脚用作输入端口引脚(P34)（参见 18.3 $\overline{\text{RESET}}$ 引脚用作只能输入的端口引脚（P34）的注意事项）
- (4) 上电时或复位释放后的振荡稳定时间
 - $2^{10}/f_x$
 - $2^{12}/f_x$
 - $2^{15}/f_x$
 - $2^{17}/f_x$

图 18-1. 选项字节的位置



18.2 选项字节的格式

选项字节的格式如下所示。

图 18-2. 选项字节的格式(1/2)

地址: 0080H

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

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

注意事项 仅当选择晶体/陶瓷振荡时钟作为系统时钟源时，该选项的设置才有效。如果选用高速内部振荡器时钟或外部时钟输入作为系统时钟源，无需等待时间。

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

注意事项 因为在复位释放之后，才能参考选项字节。如果在参考选项字节之前，低电平输入到复位引脚，这样的复位状态不会被释放。

并且，当 RMCE 设置为 0 时，连接上拉电阻。

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

注意事项 因为 X1 和 X2 引脚也可以作为 P121 和 P122 引脚使用，根据系统时钟源的选择，X1 和 X2 引脚可以使用的条件不同。

(1) 选择晶体/陶瓷振荡时钟

X1 和 X2 引脚不能作为 I/O 端口使用，因为它们被作为时钟输入端口。

(2) 选择外部时钟输入

因为 X1 引脚作为外部时钟输入口，P121 口不能作为 I/O 端口引脚使用。

(3) 选择高速内部振荡时钟

P121 和 P122 端口可作为 I/O 端口引脚。

备注 x: 无需考虑

图 18-2. 选项字节的格式(2/2)

LIOCP	低速内部振荡器工作
1	不能停止（即使 LSRSTOP 位写入 1，振荡也不会停止）
0	可以通过软件停止(当 LSRSTOP 位写入 1 时，振荡停止)

注意事项

- 1. 如果选择低速内部振荡器时钟不能停止，看门狗定时器(WDT)的计数时钟就会固定为低速内部振荡器时钟。
- 2. 如果选择低速内部振荡器时钟可以通过软件停止，不管低速内部振荡模式寄存器(LSRCM)的第 0 位 (LSRSTOP)设置如何，WDT 的计数时钟在 HALT/STOP 模式下都会被停止。同样地，当 WDT 的计数时钟选择低速内部振荡器时钟之外的时钟信号时，时钟提供也被停止。
当低速内部振荡器工作(LSRSTOP = 0)时，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时间，可参考所使用的振荡器的特性。
- 3. 设置选项字节的软件代码示例如下所示。

```
OPB CSEG AT 0080H
DB 10010001B      ; 数据设置到选项字节
                   ; 低速内部振荡器不能停止
                   ; 系统时钟是晶体或陶瓷振荡器。
                   ; RESET引脚用作只能输入的端口引脚(P34)
                   ; 最短振荡稳定时间( $2^{10}/f_x$ )
```

4. 选项字节被参考时序的详细内容，请参见第十五章 复位功能。

18.3 RESET 引脚用作只能输入的端口引脚(P34)时的注意事项

对于已经通过选项字节功能被设置为“RESET 引脚用作只能输入的端口引脚(P34)”并完成写入的设备，当重新擦除/重新写入时(使用专用 flash 存储器编程器在板编程)，要留意如下内容：
给目标系统上电前，连接专用 flash 存储器编程器并打开编程器的电源。
如果目标系统已经提前上电，无法切换到 flash 存储器编程模式。

第十九章 FLASH存储器

19.1 特点

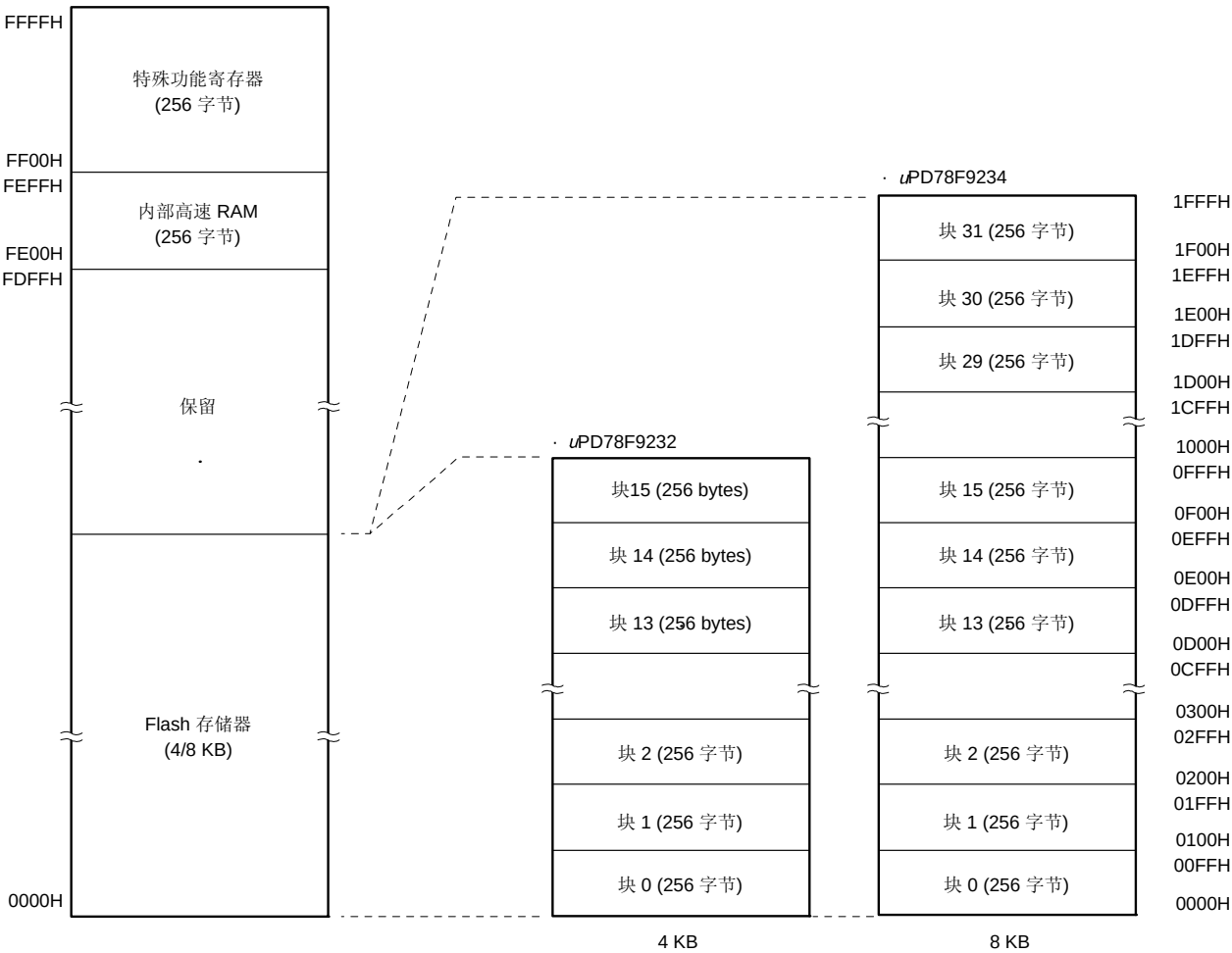
78K0S/KB1+ 内部 FLASH 存储器有如下特点:

- 不需要为擦除/写入准备单独的专用电源
- 容量: 4 KB / 8 KB
 - 擦除单元: 1 块 (Block) (256 字节)
 - 写入单元: 1 块 (Block) (以 onboard/offboard 编程时), 1 字节 (自编程时)
- 写入方法
 - 使用专用的 FLASH 编程器写入 (on-board/off-board 编程)
 - 使用用户程序写入 FLASH 存储器 (自编程写入)
- 支持带有安全功能的 onboard/offboard FLASH 存储器的写入
- 通过保护字节, 在自编程时支持以块为单位的安全功能

19.2 存储器配置

4/8 KB 的 FLASH 内部存储器区域被分为 16/32 个块，并且能够以块为单位写入和擦除数据。通过专用的 FLASH 编程器，也可以一次性擦除所有的块。

图 19-1. Flash 存储器图表



19.3 功能概要

78K0S/KB1+ 内部 FLASH 存储器能够通过专用的 FLASH 编程器写入，不论 78K0S/KB1+是否已经连接到目标板（on-board/off-board 编程模式）。

提供使用用户程序写入程序的功能（自编程），当目标系统已经生产/出货后要改变程序时，该功能为用户提供了一种较为理想的解决办法。

参考表 19-1 FLASH 存储器的写入控制功能。

另外，安全功能禁止未经授权将用户程序写入内部 flash 存储器。

详细的安全功能说明，请参考 19.7.3 安全设置。

表 19-1. 写入方法

写入方法	功能概要	操作模式
On-board 编程	在器件已经安装在目标系统之后，可通过专用 flash 编程器，写入 flash 存储器。	Flash 存储器编程模式
Off-board 编程	在器件安装在目标系统之前，可通过专用 flash 编程器和专用编程适配板，写入 flash 存储器。	
自编程	通过执行一段预先通过 on-board/off-board 编程方法写入到 flash 存储器的用户程序，可以写入 flash 存储器。	自编程模式

- 备注
1. FA 系列是 Naito Densei Machida Mfg. Co., Ltd 公司的产品。

2. flash 存储器写入控制功能的详细描述，请参考以下部分。
 - 19.7 On-Board 和 Off-Board Flash 存储器编程
 - 19.8 通过自编程进行 Flash 存储器编程

19.4 使用FLASH编程器的写入

可以使用下列两种专用 FLASH 编程器向 78K0S/KB1+的内部 FLASH 存储器写入数据。

- <R>
- FlashPro5 (PG-FP5, FL-PR5)
 - QB-MINI2

通过专用 FLASH 编程器，数据能够以 on-board 或 off-board 方式写入 FLASH 存储器中。

(1) On-board 编程

78K0S/KB1+ 安装到目标系统后，可以写入 FLASH 存储器。目标系统上必须安装连接器，用来连接专用 FLASH 编程器和测试点。只有在使用晶体/陶瓷振荡器写入数据时才要求测试点（参考图 19-4 测试点的安装）。

(2) Off-board 编程

在 78K0S/KB1+被安装到目标系统以前，可以通过专用编程适配器（FA 系列）将数据写入 FLASH 存储器中。

- <R>
- 备注

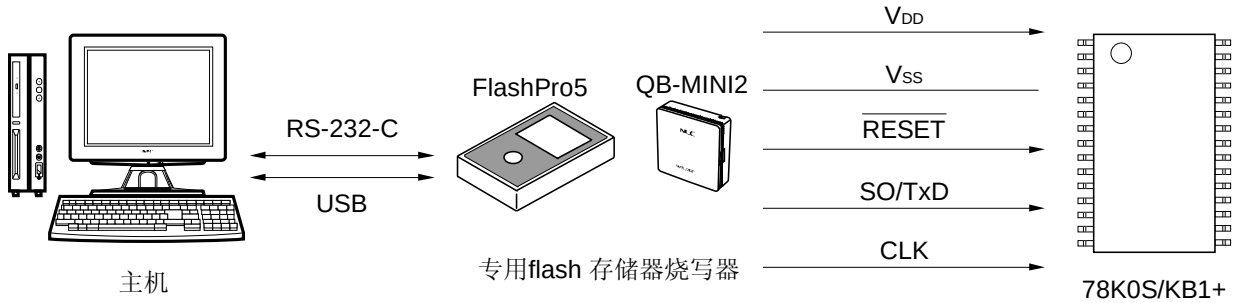
FL-PR5 和 FA 系列是 Naito Densei Machida Mfg. Co., Ltd 的产品。

19.5 编程环境

将程序写入 flash 存储器的编程环境要求如下。

<R>

图 19-2. 将程序写入 flash 存储器的环境（使用 FlashPro5/QB-MINI2）



<R>

备注 对于 QB-MINI2, SO/TxD 信号名称是 DATA。

必须使用主机来控制专用flash存储器编程器。当使用PG-FP5、或FL-PR5时，程序从主机下载之后，数据才能通过flash编程器写入。

专用 FLASH 编程器与 78K0S/KB1+接口连接时，使用 UART 来操作数据的写入和擦除。当使用 off-board 方式写入 FLASH 存储器时，需要一个专用编程适配器（FA 系列）。

可以从开发工具网站下载最新的编程器固件，GUI，和参数文件等。

(<http://www.necel.com/micro/ods/eng/index.html>)

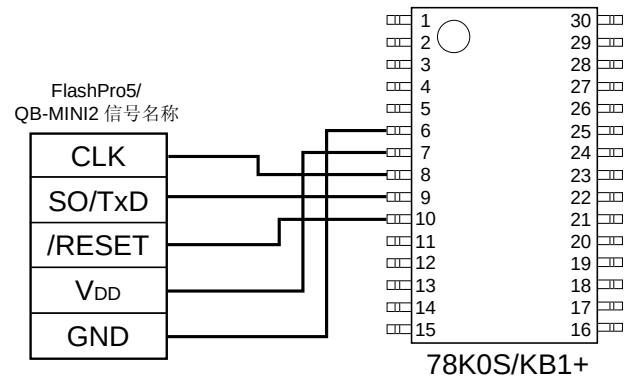
<R>

表 19-2. 78K0S/KB1+ 和 FlashPro5/QB-MINI2 之间的连线

FlashPro5/QB-MINI2 引脚			78K0S/KB1+连接引脚
引脚名称	I/O	引脚功能	引脚名称
CLK	输出	78K0S/KB1+的时钟	X1/P121
SO/TxD	输出	接收信号/on-board 模式信号	X2/P122
RESET	输出	复位信号	RESET/P34
V _{DD}	—	V _{DD} 电压/电压监视	V _{DD}
GND	—	接地	V _{SS}

<R>

图 19-3. 和 FlashPro5/QB-MINI2 的连线图（30 引脚产品）



备注 对于 QB-MINI2，SO/TxD 信号名称是 DATA。

19.6 On-Board 引脚连接

如要以 on-board 方式写入 flash 存储器，目标系统必须提供连接到专用 flash 编程器的连接器。首先目标板要可以选择正常操作模式或者 flash 存储器写入模式。

当设定为 flash 存储器写入模式时，所有不参与写入 flash 存储器的引脚都处于复位后的状态。因此，如果外部器件不能在复位后识别这个状态，引脚必须以如下方式进行处理。

在自写入模式下引脚的状态和 HALT 模式的状态是相同的。

19.6.1 X1 和 X2 引脚

X1 和 X2 引脚被用作 FLASH 存储器编程的串行接口。因此，如果 X1 和 X2 引脚连接外部设备，将会产生信号冲突。为了阻止信号发生冲突，此时应该断开引脚 X1 和 X2 与外部设备的连接。

同样，当一个电容连接到 X1 和 X2 引脚时，通讯期间的波形被改变，因此电容器的容量可能导致通讯被禁止。在 flash 编程期间确保断开电容的连接。

当使用晶体或陶瓷振荡器作为系统时钟，且隔离振荡器比较困难时，使用 on-board 编程方式须采用下列方法（1）和（2）：

- （1）在设备与振荡器之间加入测试点，测试点数量要尽可能少，在测试点连接 FLASH 编程器。两者连线尽要可能短（参见图 19-4 和表 19-3）。
- （2）使用专用 FLASH 编程器的 GUI 软件来设定数据写入的通讯时钟振荡频率。查找所使用振荡器的串/并谐振频率和反谐振频率，设定的振荡器频率应该在谐振频率 $\pm 10\%$ 的范围之外（参见图 19-5 和表 19-4）

图 19-4. 安装测试点的示例

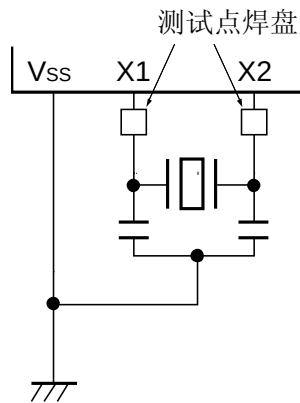


表 19-3. 使用的时钟和测试点的安装

时钟源		测试点的安装
高速内部振荡时钟		不需要
外部时钟		
晶体/陶瓷振荡	在振荡器安装之前	
	在振荡器安装之后	需要

图 19-5. PG-FP5 图形用户界面软件设置演示

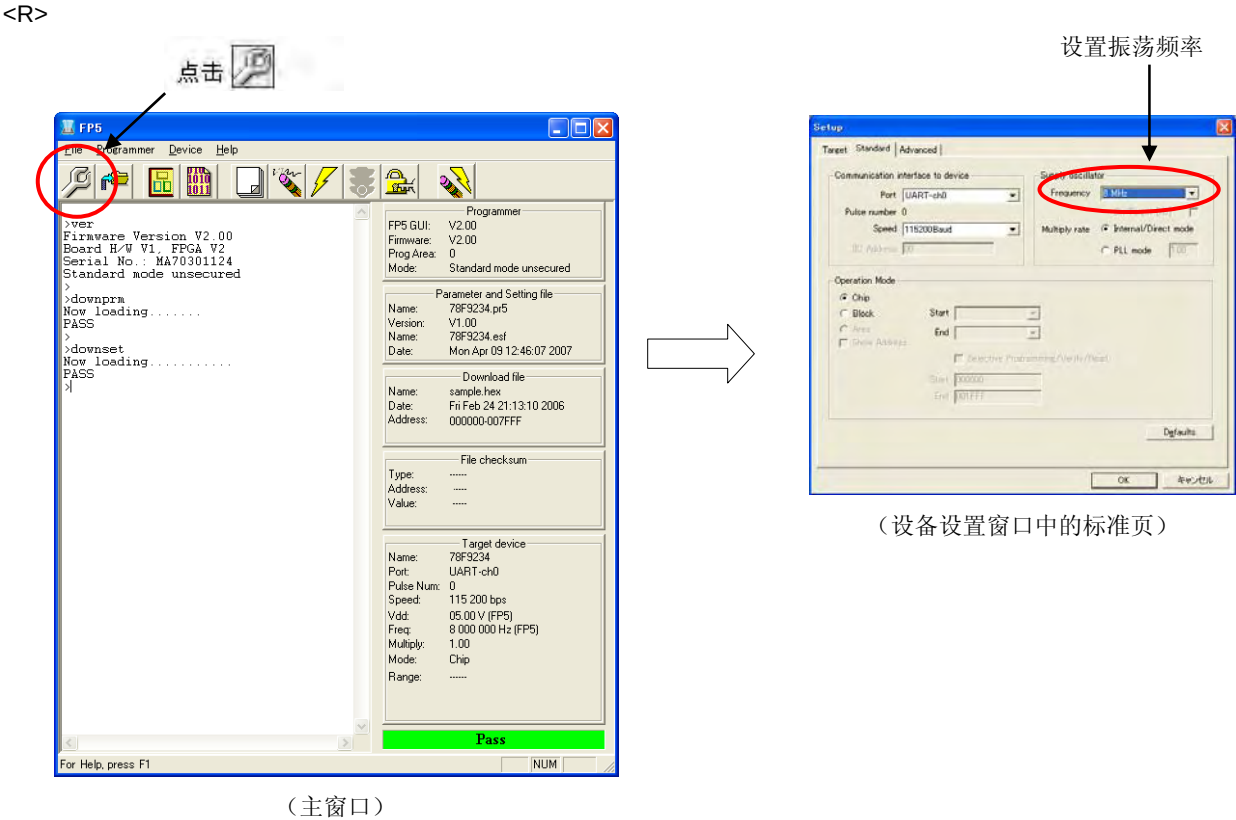


表 19-4. 振荡频率和 PG-FP5 GUI 软件设置值示例

晶振频率	PG-FP5 软件设定值举例（通讯频率）
$2\text{ MHz} \leq f_x < 4\text{ MHz}$	8 MHz
$4\text{ MHz} \leq f_x < 8\text{ MHz}$	9 MHz
$8\text{ MHz} \leq f_x < 9\text{ MHz}$	10 MHz
$9\text{ MHz} \leq f_x \leq 10\text{ MHz}$	8 MHz

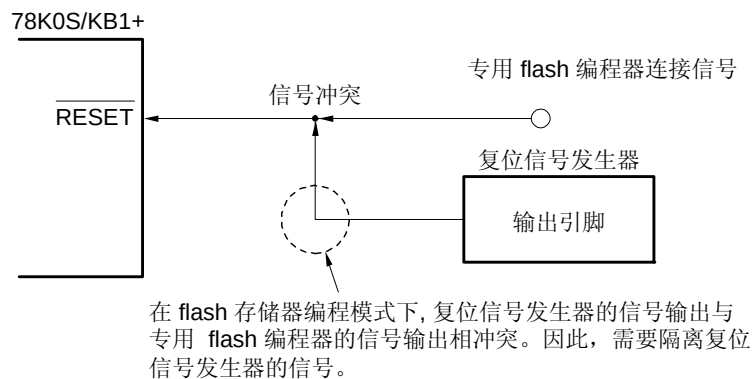
注意事项 以上值为推荐值。由于环境不同，这些值可能改变，所以设置之前请先充分评估。

19.6.2 RESET 引脚

若专用 FLASH 编程器的复位信号与已经连接到板上信号发生器的RESET引脚相连，将会发生信号冲突。为了防止这种冲突发生，需要隔离复位信号发生器。

如果在 FLASH 存储器编程模式设定期间用户系统输入复位信号，则不能够对 FLASH 存储器正确编程。因此不要输入除专用 FLASH 编程器复位信号外的任何信号。

图 19-6. 信号冲突 (RESET 引脚)



19.6.3 端口引脚

当 FLASH 存储器编程模式设定后, 所有不参与写入的引脚都进入与复位后相同的状态。如果连接这些端口引脚的外部设备不能在复位后立即识别这个状态, 则此引脚必须通过一个电阻连接到 V_{DD} 或 V_{SS} 上。

引脚的状态在自编程模式下和 HALT 模式下是相同的。

19.6.4 供电

将 V_{DD} 引脚连接到 FLASH 编程器的 V_{DD} 引脚, 将 V_{SS} 引脚连接到 FLASH 编程器的 V_{SS} 引脚。

提供的电源和正常工作模式下的其它电源相同 (AV_{REF} 和 AV_{SS})。

19.7 On-Board 和 Off-Board Flash存储器编程

19.7.1 Flash 存储器编程模式

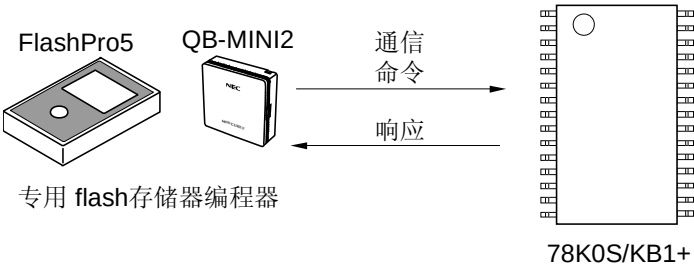
为了使用专用 FLASH 编程器写入 FLASH 存储器里的内容，需要将 78K0S/KB1+设定在 FLASH 存储器编程模式。
当 78K0S/KB1+连接到 FLASH 编程器并且通讯命令发送到微控制器时，微控制器设定在 FLASH 存储器编程模式。
当 on-board 写入 FLASH 存储器时，可使用跳线改变操作模式。

19.7.2 通讯命令

通过命令使 78K0S/KB1+与专用 flash 编程器通讯。从 flash 编程器发送到 78K0S/KB1+的信号称为命令，从 78K0S/KB1+发送到 FLASH 编程器的命令称为应答命令。

<R>

图 19-7. 通讯命令



下表列出了通讯命令。这些命令都是由编程器发出，78K0S/KB1+根据相应命令完成处理。

表 19-5. 通讯命令

分 类	命令名	功 能
擦除	一次性擦除（芯片擦除）命令	擦除整个存储器的内容
	块擦除命令	擦除指定块的内容
写入	写入命令	将数据写入指定地址范围中，并进行校验
校验和	校验和命令	读取指定地址范围的校验和，并与写入数据进行比较
块检测	块检测命令	确定整个存储器的擦除状态
安全	安全设置命令	禁止一次性擦除（芯片擦除）命令，块擦除命令，和写入命令以阻止第三方操作

78K0S/KB1+对专用 FLASH 编程器发出的命令将会返回一个应答命令，应答命令如下表所示：

表 19-6. 应答命令

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

19.7.3 安全设定

通过使用安全设定命令能够禁止以下操作

- 禁止一次性擦除（芯片擦除）
禁止块擦除和一次性擦除（芯片擦除）。一旦一次性擦除（芯片擦除）被禁止，所有的保护设定将不能取消。

注意事项 在一次性擦除（芯片擦除）的安全设定后，擦除指令不能执行。另外，即使一个写入命令被执行，与已存在的数据不同的部分是不能被写入的，因为擦除命令已不能执行。

- 块擦除保护
禁止块擦除命令，该命令设置可通过一次性擦除（芯片擦除）命令取消

- 写保护
禁止在 FLASH 中执行整块写入和整块擦除命令。该命令设置也可通过一次性擦除（芯片擦除）命令取消

备注 当下一次设定编程模式时，安全设置有效。

FLASH 存储器在默认设置下，一次性擦除（芯片擦除），块擦除和写入命令是有效的。以上安全设定只是在 on-board/off-board 编程时有效。每一个安全设定都能够和其他设定一起使用。

图 19-7 以下图为当设置了 78K0S/KB1+的安全功能后擦除与写入命令的关系。

表 19-7. 设置了 78K0S/KB1+的安全功能后擦除与写入命令的关系

命令 安全	一次性擦除（芯片擦除）	块擦除命令	写命令
当一次性擦除（芯片擦除）安全操作有效时	禁止	禁止	有效 ^注
当块擦除安全操作有效时	有效		有效
当写入命令安全操作有效时			禁止

注 因为擦除命令被禁止，所以与已写入 flash 存储器不同的数据是不能被写入的

表 19-8 显示在每个编程模式下安全设置与操作的关系

表 19-8. 每个编程模式下安全设定与操作的关系

编程模式 安全设定	On-Board/Off-Board 编程		自编程	
	安全设定	安全操作	安全设定	安全操作
一次性擦除（芯片擦除）	允许	有效 ^{注1}	不允许	无效 ^{注2}
块擦除				
写入				

注 1. 安全设定可以禁止任何指令的执行
 2. 不管安全设定如何，自编程指令都可以执行

19.8 通过自编程方式对FLASH进行编程

780K0S/KB1+支持自编程功能，可以通过用户程序来写入 FLASH 存储器，因此 780K0S/KB1+具有在线升级程序的功能。

注意事项 在执行自编程以前，自编程处理程序必须先被包含在用户程序中。

备注 1. 自编程的用法，参见后面 19.8.4. 中的例子。
 2. 若想用 78K0S/KB1+的内部 FLASH 当作外部 EEPROM 来存储数据时，请参考《78K0S/Kx1+ EEPROM 模拟应用笔记》（U17379E）。

19.8.1 自编程概述

如要执行自编程， 需要将工作模式由用户程序模式（正常模式）切换到自编程模式。在自编程模式下，先设置特定寄存器，通过执行 HALT 指令执行擦写处理程序。当程序结束时，将自动退出 HALT 状态。

关于通过对特定寄存器的操作使模式切换到自编程模式的详情请参考 19.8.4 正常模式与自编程模式切换示例。

备注 自编程写入数据，可参照 MOV 指令。

表 19-9 自编程模式

模式	用户程序执行	通过 HALT 指令对 FLASH 执行 写入/擦除命令
正常模式	允许	—
自编程模式	允许 ^注	允许

注 在自编程模式下，可屏蔽中断处理服务程序被禁止。

图 19-8 是自编程框图，图 19-9 是自编程状态迁移图，表 19-10 列举了自编程控制命令。

图 19-8 自编程框图

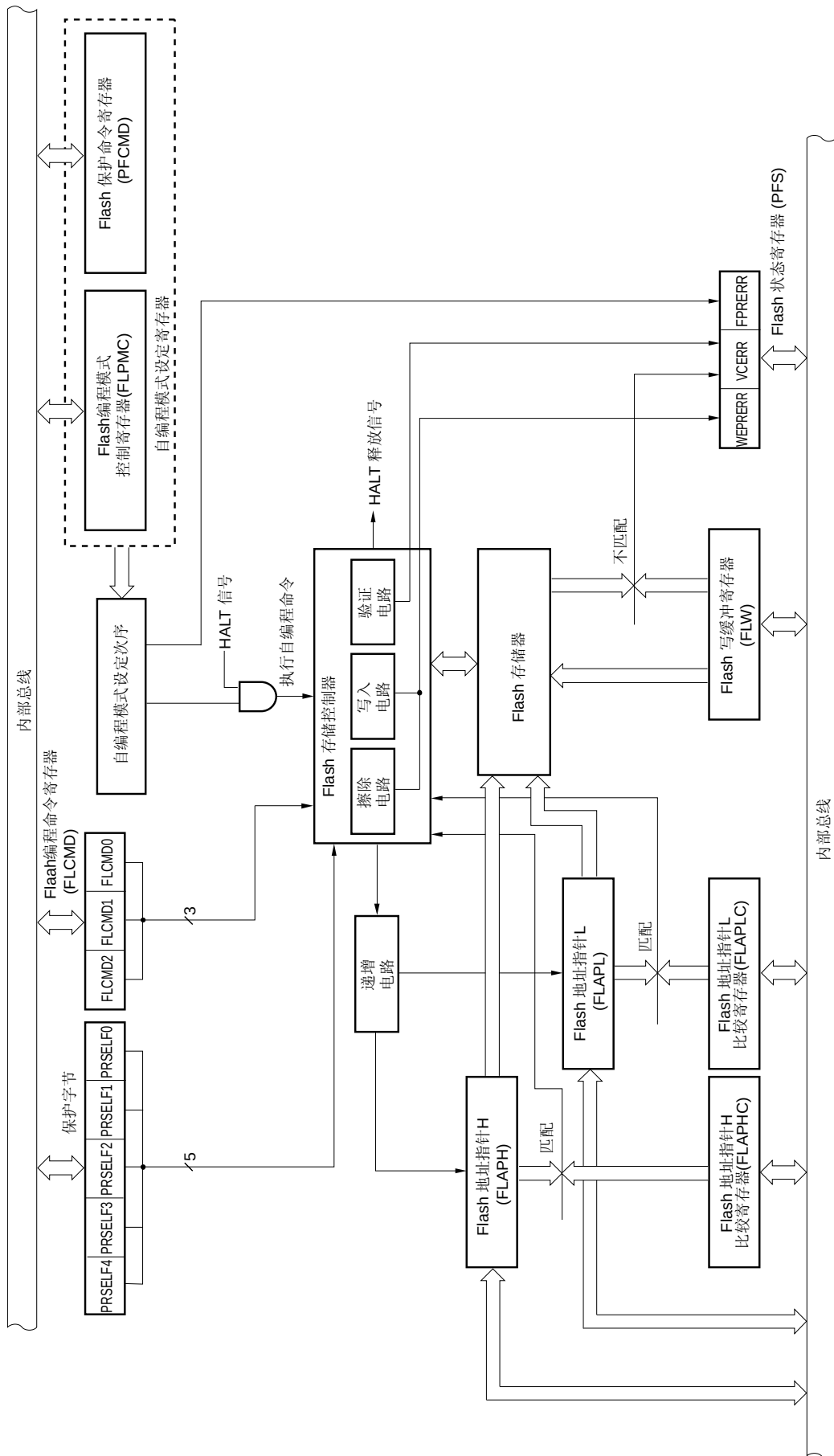


图 19-9 自编程状态迁移图

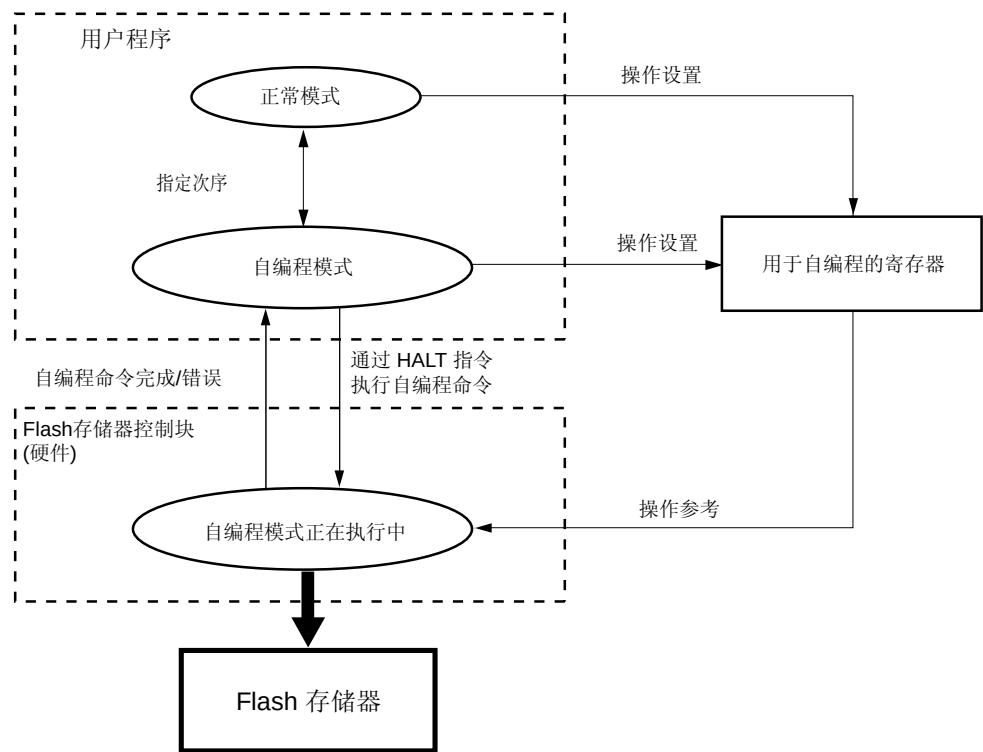


表 19-10 自编程控制命令

命令名称	功能	从 HALT 指令执行开始到命令执行结束的时间
内部校验 1	该指令用于验证数据是否已经被正确的写入 FLASH 存储器中。它用于检查数据是否被写入到一个完整的块。	内部校验 1 个块： 6.8ms（内部校验命令执行一次）
内部校验 2	该指令用于验证数据是否已经被正确的写入 FLASH 存储器中。它用于检查数据是否被写入到相同块中的多个地址。	内部校验 1 字节：27 μ s
块擦除 ^注	该指令用于擦除指定块的内容，先指定块的序号，然后执行此命令	8.5 ms
空白块检验	该指令用于校验指定块是否已被擦除。先指定块的序号，然后执行此命令	480 μ s
1 字节写入	该指令用于对 FLASH 存储器指定地址写入 1 字节数据。在确定写入地址和数据之后，就可执行此命令	150 μ s

<R> 注 为单次擦除设置重试的数量要大于块擦除时间除以 8.5 ms 得到的值，和 flash 存储器块擦除所需的时间（最大值）一致。

备注 内部校验 1 命令可以在相同的块中由指定地址执行，如果数据写入到相同块中的多个地址时推荐使用内部校验 2 命令。

19.8.2 自编程注意事项

- 当自编程命令执行时，其它命令都不能执行。因此，要先清除和重启看门狗时钟计数器，以便在自编程时计数器不会溢出。关于执行自编程需要的时间请参考表 19-10。
- 在自编程期间发生的中断，在自编程模式结束后才能被响应。为了避免这种情况，应在从正常模式切换到自编程模式之前以一定顺序终止中断服务（通过将 MK0 和 MK1 设定为 0FFH，并执行 DI 指令）
- 当执行自编程命令时，不使用 RAM。
- 当 FLASH 存储器正在数据写入或擦除过程中，如果供电电压下降或者有复位信号输入，则此次写入或擦除操作不一定成功。
- 在块擦除中空白数据的值设为 FFH。
- 预先设置 CPU 时钟以使其在自编程期间至少为 1 MHz。
- 执行指定序列到自编程模式后，在执行 NOP 和 HALT 指令后立即执行自编程。此时，HALT 指令在 10 μ s（最大值）+ 2 个 CPU 时钟（f_{cpu}）后自动释放。
- 如果选择晶振或外部时钟作为系统时钟，执行指定序列到自编程模式后执行 NOP 和 HALT 指令，释放 HALT 状态后等待 8 μ s，然后执行自编程。
- 使用 1 位存储器操作指令检查 FPRERR。
- 自编程模式下和 HALT 模式下引脚的状态是相同的。
- 在自编程模式中，禁止通过 on-board/off-board 编程设置安全功能，无论安全功能如何设定，自编程命令都可执行。想要在自编程期间禁止写入或擦除进程，则需要设定保护字节。
- 执行自编程命令之前，要确保 flash 地址指针 H（FLAPH）的第 5 位 ~ 第 7 位和 flash 地址指针比较寄存器（FLAPHC）清零。如果在这些位为 1 时执行自编程命令，则设备可能出现故障。
- 就在设置自编程模式和正常模式之前，将 FLCMD 寄存器的值清零为 00H。

19.8.3 自编程功能所使用的寄存器

自编程使用以下的寄存器

- Flash 编程模式控制寄存器（FLPMC）
- Flash 保护命令寄存器（PFCMD）
- Flash 状态寄存器（PFS）
- Flash 编程命令寄存器（FLCMD）
- Flash 地址指针 H 和 L 寄存器（FLAPH 和 FLAPL）
- Flash 地址指针 H 比较寄存器，和 FLASH 地址指针 L 比较寄存器（FLAPH 和 FLAPLC）
- Flash 写入缓存寄存器（FLW）

78K0S/KB1+ 在 FLASH 存储器的 0081H 地址有一个区域被称为保护字节

（1）Flash 编程模式控制寄存器（FLPMC）

在自编程模式下，该寄存器用来设置将数据写入 FLASH 存储器的工作模式，并可用于读取保护字节的值。

只有按特定顺序（请参阅 19.8.3（2）FLASH 保护命令寄存器（PFCMD））才能写入数据，这样就不会因为偶然的噪声干扰或者程序挂起等故障导致应用系统意外停止。

该寄存器可通过 8 位的存储操作指令来设置。
复位信号的产生，会使该寄存器内容不确定。

图 19-10. Flash 编程模式控制寄存器（FLPMC）的格式

地址： FFA2H 复位后： 不确定^{注1} 读/写^{注2}

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

FLSPM	自编程模式下操作模式的选择
0	正常模式 这是正常操作状态，执行HALT指令设置待机状态
1	自编程模式 在执行HALT指令之前，首先要设定FLASH编程命令、地址偏移量、写入数据并设置FLSPM为1。在完成以上操作后，执行HALT指令，FLASH存储器的模式将从普通转为编程模式。

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	保护字节的设定值 可通过这些位读出
---------	---------	---------	---------	---------	----------------------

- 注 1. 当复位释放时，第 0 位（FLSPM）被清 0。当复位释放后，保护字节的设定值可从第 2 位到第 6 位（PRSELF0 到 RESELF4）读出。
2. 第 2 位到第 6 位（PRSELF0 到 RESELF4）是只读的。

- 注意事项 1. 设置自编程模式的情况下的注意事项，请参考 19.8.2 自编程功能的注意事项。
2. 预先设置 CPU 时钟，使其在自编程期间至少为 1 MHz 。
3. 执行指定序列到自编程模式之后，立即执行 NOP 和 HALT 指令，然后再执行自编程。此时，HALT 指令在 10 μs（最大值）+ 2 个 CPU 时钟（f_{CPU}）后自动释放。
4. 如果选择晶振或外部时钟作为系统时钟，执行指定序列到自编程模式后执行 NOP 和 HALT 指令，释放 HALT 状态后等待 8 μs，然后执行自编程。
5. 就在设置自编程模式和正常模式之前，将 FLCMD 寄存器的值清零为 00H。

(2) Flash 保护命令寄存器（PFCMD）

如果由于噪声或程序挂起等原因引起的故障导致应用系统停止，此时对 FLASH 编程模式控制寄存器（FLPMC）进行写入操作将对系统产生严重影响。PFCMD 用来保护 FLPMC 的写入，这样应用系统就不会意外停止。

- 必须按以下顺序对 FLPMC 执行写入操作
- <1> 写入一特定的值（A5H）到 PFCMD 中
- <2> 写入设定值到 FLPMC 的第 0 位（FLSPM）中（此时写入操作是无效的）

<3> 将设定值的反码写入 FLPMC 的第 0 位（FLSPM）（此时写入操作是无效的）

<4> 将设定值写入 FLPMC 的第 0 位（FLSPM）（此时写入操作是有效的）

注意事项 在自编程模式下中断服务不能执行。在执行设置自编程模式的指定序列前和执行切换正常模式的指定序列后禁止中断服务（当 MK0 和 MK1 =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。

可以通过 FLASH 状态寄存器（PFS）的第 0 位（FPRERR）来验证是否出现非法的存储操作。

每当 FLPMC 的值发生改变时，都必须向 PFCMD 写入值 A5H。用 1 位存储器操作指令检查 FPRERR。

PFCMD 的值可通过 8 位的存储器操作指令来设置。

复位信号的产生，会使寄存器 PFCMD 的值不确定。

图 19-11. Flash 保护命令寄存器（PFCMD）的格式

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

（3）Flash 状态寄存器（PFS）

如果数据不能按正确顺序（写入 FLASH 保护命令寄存器 PFCMD）写入受保护的 FLASH 编程模式控制寄存器（FLPMC）中，FLPMC 不能被写入并且出现保护错误，PFS 的第 0 位（FPRERR）将被置 1。

当 FPRERR 为 1 时，可以通过直接写 0 来清 0。

自编程模式期间可能产生的错误能够通过 PFS 的第 1 位（VCERR）和第 2 位（WEPRERR）获得，（VCERR）和（WEPRERR）可以通过写 0 来清 0。

要验证操作是否执行正确，PFS 寄存器的必须被预先清零。

PFS 可以使用 1 位或 8 位存储器操作指令设定。

复位信号的产生，会将 PFS 清为 00H。

注意事项 使用 1 位存储器操作指令检查 FPRERR。

图 19-12. Flash 状态寄存器（PFS）的格式

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

1. FPRERR 标记的操作条件

<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。

- 在<1>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<2>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<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 以上尖括号中的编号对应于（2）Flash 保护寄存器（PFCMD）。

<复位条件>

- 如果 FPRERR 标志写 0
- 复位信号输入

2. VCERR 标志的操作条件

<设定条件>

- 擦除校验错误
- 内部写入校验错误

如果 VCERR 置 1，则说明 FLASH 没有被正确的擦除或写入。按照指定程序，对 FLASH 进行再次擦除或写入。

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>

- VCERR 标记被置为 0
- 外部复位信号输入

3. WEPRERR 标记操作条件

<设置条件>

- 使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”被写入到一个没有被擦除的位（一个为“0”的位）

<复位条件>

- WEPRERR 标记被置 0
- 外部复位信号输入

(4) FLASH 编程命令寄存器 (FLCMD)

在自编程模式下，FLCMD 被用来确定 FLASH 的擦、写、校验操作。

该寄存器可通过 1 位或 8 位的操作指令来设置。

复位信号的产生，会将该寄存器清为 00H。

图 19-13. Flash 编程命令寄存器 (FLCMD) 的格式

地址: FFA3H	复位后: 00H	读/写						
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	功 能
0	0	1	内部校验1	此命令用来验证写入FLASH内的数据是否正确。它用于检查数据是否已写入到一个完整的块。如果验证中发现数据写入有错误，FLASH状态寄存器（PFS）的第1位（VCERR）或第2位（WEPRERR）将被置为1。
0	1	0	内部校验2	此命令用来验证写入FLASH内的数据是否正确。它用于检查数据是否被写入到相同块中的多个地址。如果验证中发现数据写入有错误，FLASH状态寄存器（PFS）的第1位（VCERR）或第2位（WEPRERR）将被置为1。
0	1	1	块擦除	此命令用于对指定块的擦除。它可以用在on-board和自编程模式下。
1	0	0	块空白检测	此命令用于检测指定的块是否已经被擦除。
1	0	1	字节写入	此命令用于将一个字节数据写入FLASH的指定地址。指令执行前要指定写入数据和写入地址。 如果数据“1”写入到一个没有被删除的位（该位内容为0），则FLASH状态寄存器（PFS）的第2位被置为1。
其他 [※]			禁止设定	

注 如果执行的命令不在上表列举范围，命令的执行立刻终止，flash 状态寄存器的第 1 位和第 2 位（WEPRERR 和 VCERR）被置为 1。

(5) Flash 地址指针 H 和 L (FLAPH 和 FLAPL)

Flash 地址指针 H 和 L (FLAPH 和 FLAPL) 用于在自编程模式下指定 FLASH 的删除、写入、验证的开始地址。

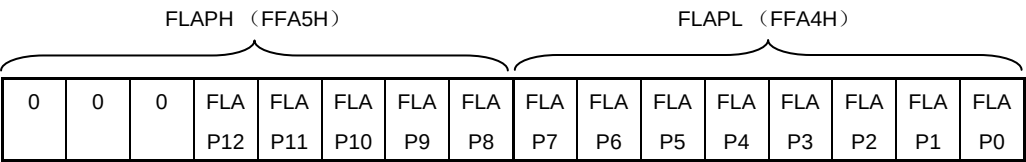
FLAPH 和 FLAPL 由计数器组成，当不执行编程命令时，它们递增到与 FLAPHC 和 FLAPLC 的值相等。因此，当程序命令执行时，PLAPHC 和 FLAPLC 的值都需要重新设置。

Flash 地址指针 H 和 L (FLAPH 和 FLAPL) 可通过 1 位或 8 位操作指令来设置。

复位信号的产生，会使该寄存器的值不确定。

图 19-14. Flash 地址指针 H/L（FLAPH/FLAPL）格式

地址： FFA4H, FFA5H 复位后： 不确定 读/写



注意事项 执行自编程命令之前，要确保 flash 地址指针 H（FLAPH）的第 5 位 ~ 第 7 位和 flash 地址指针比较寄存器（FLAPHC）清零。如果在这些位为 1 时执行自编程命令，则设备可能出现故障。

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

在自编程模式下校验 FLASH 存储器时，FLASH 地址指针比较寄存器 H 和 L（FLAPHC 和 FLAPLC）用于指定内部序列操作的地址范围。

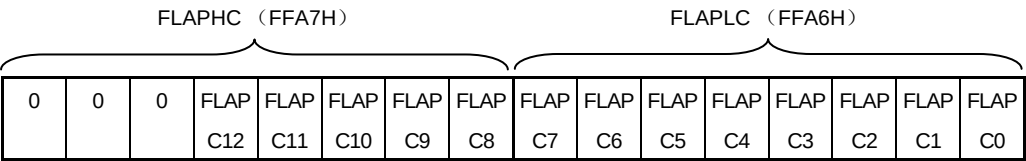
将 FLAPH 的值赋给 FLAPHC。将 FLASH 存储器校验范围的最末端地址赋给 FLAPLC。

FLAPHC 和 FLAPLC 使用 1 位或 8 位存储器操作指令来设定。

复位信号的产生，会将这些寄存器的值清为 00H。

图 19-15. Flash 地址指针 H/L 比较寄存器（FLAPHC/FLAPLC）的格式

地址： FFA6H, FFA7H 复位后： 00H 读/写



- 注意事项
1. 执行自编程命令之前，要确保 flash 地址指针 H（FLAPH）的第 5 位 ~ 第 7 位和 flash 地址指针比较寄存器（FLAPHC）清零。如果在这些位为 1 时执行自编程命令，则设备可能出现故障。
 2. 当进行块擦除、校验、空白检测时，将块序号（与 FLAPH 的值相同）赋值给 FLAPHC。
 3. 当进行块擦除时，FLAPLC 要清零，当块空白检测时，FLAPLC 要设为 FFH。

(7) Flash 写缓冲寄存器 (FLW)

数据在写入 FLASH 之前，先存储在 FLW 缓存内

该寄存器值可用 1 位或 8 位存储操作指令设置

复位信号的产生，会将该寄存器的值清为 00H。

图 19-16. FLASH 写入缓存 (FLW) 的格式

地址: FFA8H	复位后: 00H	读/写						
符号	7	6	5	4	3	2	1	0
FLW	FLW7	FLW6	FLW5	FLW4	FLW3	FLW2	FLW1	FLW0

(8) 保护字节

保护字节用于指定区域，该区域在写入或擦除受到保护。该指定的保护区域只在自编程模式下有效。因为受保护区域在自编程模式下不可用，所以可以保障保护区域内的数据。

图 19-17. 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 PD78F9232

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	状 态
0	1	0	0	0	块15 ~ 块0受到保护
0	1	0	0	1	块13 ~ 块0受到保护 块14 ~ 块15可被写入或擦除
0	1	0	1	0	块11 ~ 块0受到保护 块12 ~ 块15可被写入或擦除
0	1	0	1	1	块9 ~ 块0受到保护 块10 ~ 块15可被写入或擦除
0	1	1	0	0	块7 ~ 块0受到保护 块8 ~ 块15可被写入或擦除
0	1	1	0	1	块5 ~ 块0受到保护 块6 ~ 块15可被写入或擦除
0	1	1	1	0	块3 ~ 块0受到保护 块4 ~ 块15可被写入或擦除
0	1	1	1	1	块1 ~ 块0受到保护 块2 ~ 块15可被写入或擦除
1	1	1	1	1	所有的块都可被写入或擦除
其它					禁止

图 19-17. 保护字节的格式 (2/2)

• μ PD78F9234

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	状 态
0	0	0	0	0	块31 ~ 块0受到保护
0	0	0	0	1	块29 ~ 块0受到保护 块30 和 块31可被写入或擦除
0	0	0	1	0	块27 ~ 块0受到保护 块28 ~ 块31可被写入或擦除
0	0	0	1	1	块25 ~ 块0受到保护 块26 ~ 块31可被写入或擦除
0	0	1	0	0	块23 ~ 块0受到保护 块24 ~ 块31可被写入或擦除
0	0	1	0	1	块21 ~ 块0受到保护 块22 ~ 块31可被写入或擦除
0	0	1	1	0	块19 ~ 块0受到保护 块20 ~ 块31可被写入或擦除
0	0	1	1	1	块17 ~ 块0受到保护 块18 ~ 块31可被写入或擦除
0	1	0	0	0	块15 ~ 块0受到保护 块16 ~ 块31可被写入或擦除
0	1	0	0	1	块13 ~ 块0受到保护 块14 ~ 块31可被写入或擦除
0	1	0	1	0	块11 ~ 块0受到保护 块12 ~ 块31可被写入或擦除
0	1	0	1	1	块9 ~ 块0受到保护 块10 ~ 块31可被写入或擦除
0	1	1	0	0	块7 ~ 块0受到保护 块8 ~ 块31可被写入或擦除
0	1	1	0	1	块5 ~ 块0受到保护 块6 ~ 块31可被写入或擦除
0	1	1	1	0	块3 ~ 块0受到保护 块4 ~ 块31可被写入或擦除
0	1	1	1	1	块1 和 块0受到保护 块2 ~ 块31可被写入或擦除
1	1	1	1	1	所有的块都可被写入或擦除
其它					禁止

19.8.4 从正常模式切换到自编程模式的示例

在执行自编程模式前必须由正常模式切换至自编程模式。

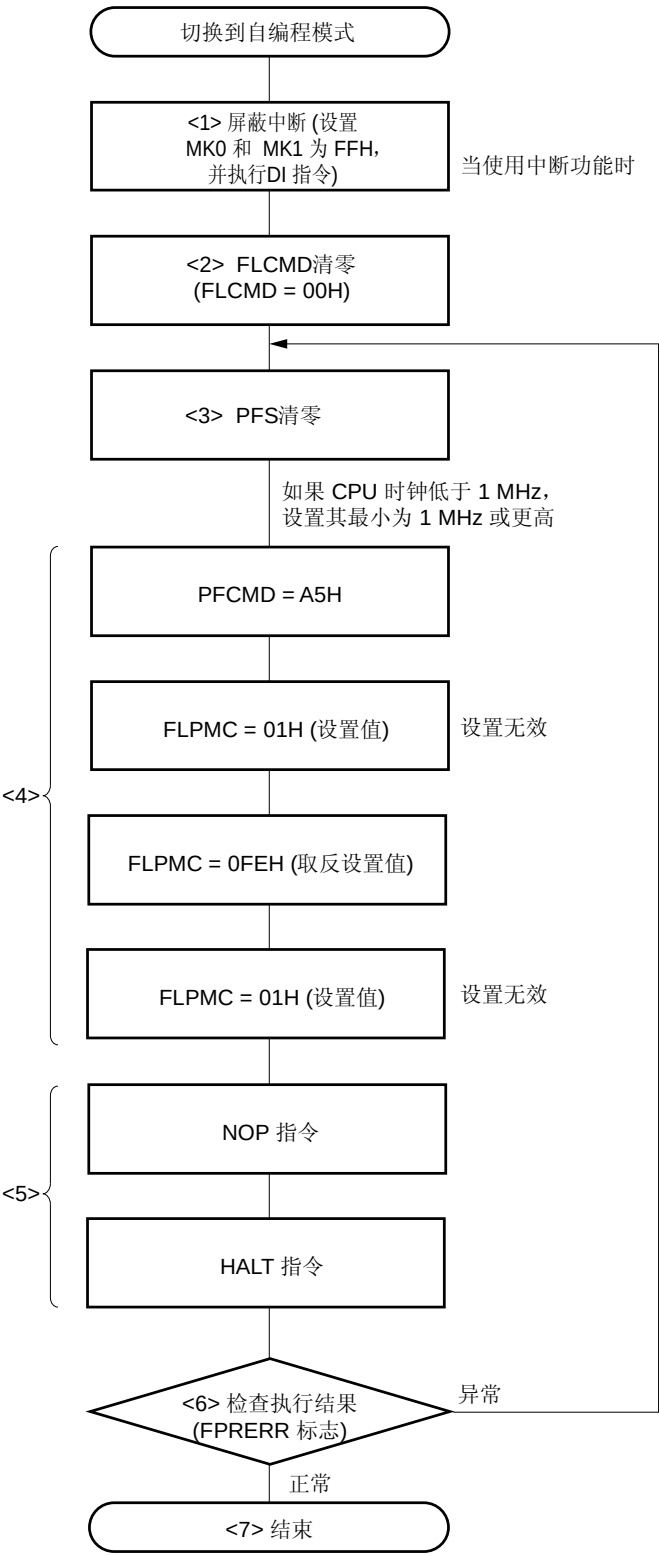
以下是切换至自编程模式的操作过程示例。

- <1> 如果使用了中断功能，则必须首先屏蔽中断（通过设置中断屏蔽标记寄存器（MK0， MK1）为 FFH，并执行 DI 指令）。
- <2> 清零 FLCMD（FLCMD = 00H）。
- <3> 清除状态寄存器（PFS）。
- <4> 按指定顺序^注设置工作模式为自编程模式
 - 设定 PFCMD 值为 A5H
 - 设定 FLPMC 值为 01H（此时写入操作是无效的）
 - 设定 FLPMC 值为 0FEH（01H 的反码）（此时写入操作是无效的）
 - 设定 FLPMC 值为 01H（此时写入操作是有效的）
- <5> 执行 NOP 指令和 HALT 指令。
- <6> 用 PFS 的第 0 位（FPRERR）检验指定序列的执行结果是否正确
异常 → <3>，正常 → <7>
- <7> 模式转换结束。

注 如果 CPU 时钟低于 1 MHz，将其设置为至少 1 MHz。

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 19-18. 切换至自编程模式示例



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 19-18 中<1> 到<7> 对应于 19.8.4 中的<1> 到<7>（前页）。

例 切换到自编程模式的示例程序

```

; -----
; 开始
; -----
    MOV     MK0, #11111111B    ; 屏蔽所有中断
    MOV     MK1, #11111111B
    MOV     FLCMD, #00H        ; FLCMD 寄存器清零

    DI

ModeOnLoop:
    MOV     PFS, #00H          ; flash 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        ; 使用 FLPMC 寄存器设置自编程模式
                                ; 控制（设置值）

    NOP
    HALT
    BT      PFS.O, $ModeOnLoop ; 验证完成写入指定的寄存器
                                ; 当发生错误重复相同的处理

; -----
; 结束
; -----

```

19.8.5 从自编程模式切换到正常模式的示例

自编程完成后，操作模式一定要从自编程返回到正常模式。

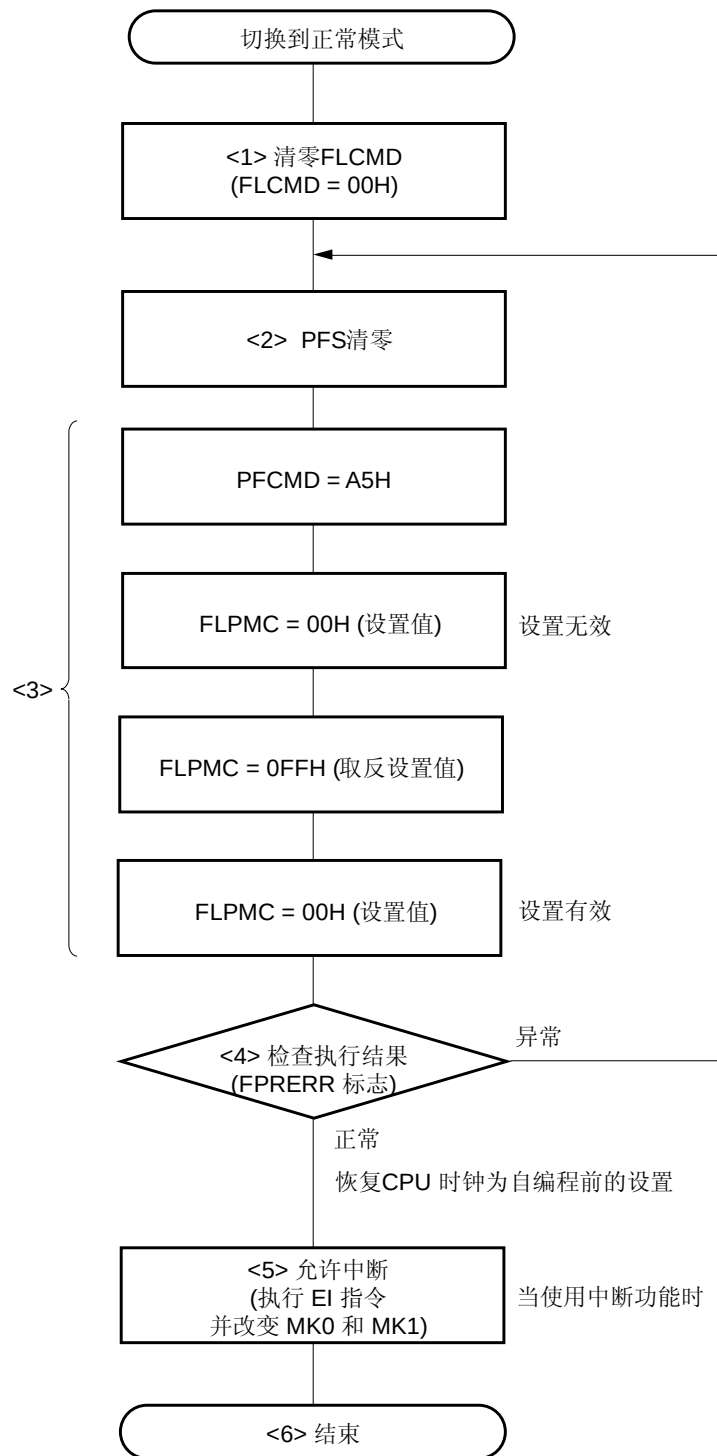
例 切换到正常模式的过程。

- <1> 清零 FLCMD (FLCMD = 00H)。
- <2> 清除状态寄存器 (PFS)
- <3> 按指定的顺序设置工作模式为正常模式
 - 设定 PFCMD 值为 A5H
 - 设定 FLPMC 值为 00H (此时写入操作是无效的)
 - 设定 FLPMC 值为 0FFH (00H 的反码) (此时写入操作是无效的)
 - 设定 FLPMC 值为 00H (此时写入操作是有效的)
- <4> 用 PFS 的第 0 位 (FPRERR) 检验执行指定顺序运行的结果是否正确^注。
错误→ <2>，正确→ <5>
- <5> 开放中断处理服务 (执行 EI 指令，并改变 MK0 和 MK1 的值) 以恢复到原来的状态。
- <6> 模式转换结束。

注 在指定顺序正常执行后，恢复自编程前的 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 19-19. 切换到正常模式示例



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 19-19 中<1> 到<6> 对应于 19.8.5 中的 <1> 到<6>（前页）

例 切换到正常模式的示例程序

```
; -----
; 开始
; -----

        MOV     FLCMD,#00H      ;FLCMD 寄存器清零
ModeOffLoop:
        MOV     PFS,#00H       ;flash 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（对设置值取反）
        MOV     FLPMC, #00H     ; 通过 FLPMC 寄存器控制设置正常模式（设置值）

        BT      PFS.0,$ModeOffLoop ;验证完成写入指定的寄存器
                                           ;当发生错误时重复相同的处理
                                           ;恢复 CPU 时钟到它的设置前
                                           ;正常完成指定寄存器后编程
                                           ;顺序

        MOV     MK0,#INT_MK0    ;恢复中断屏蔽标志
        MOV     MK1,#INT_MK1

        EI

; -----
; 结束
; -----
```

19.8.6 自编程模式下块擦除操作示例

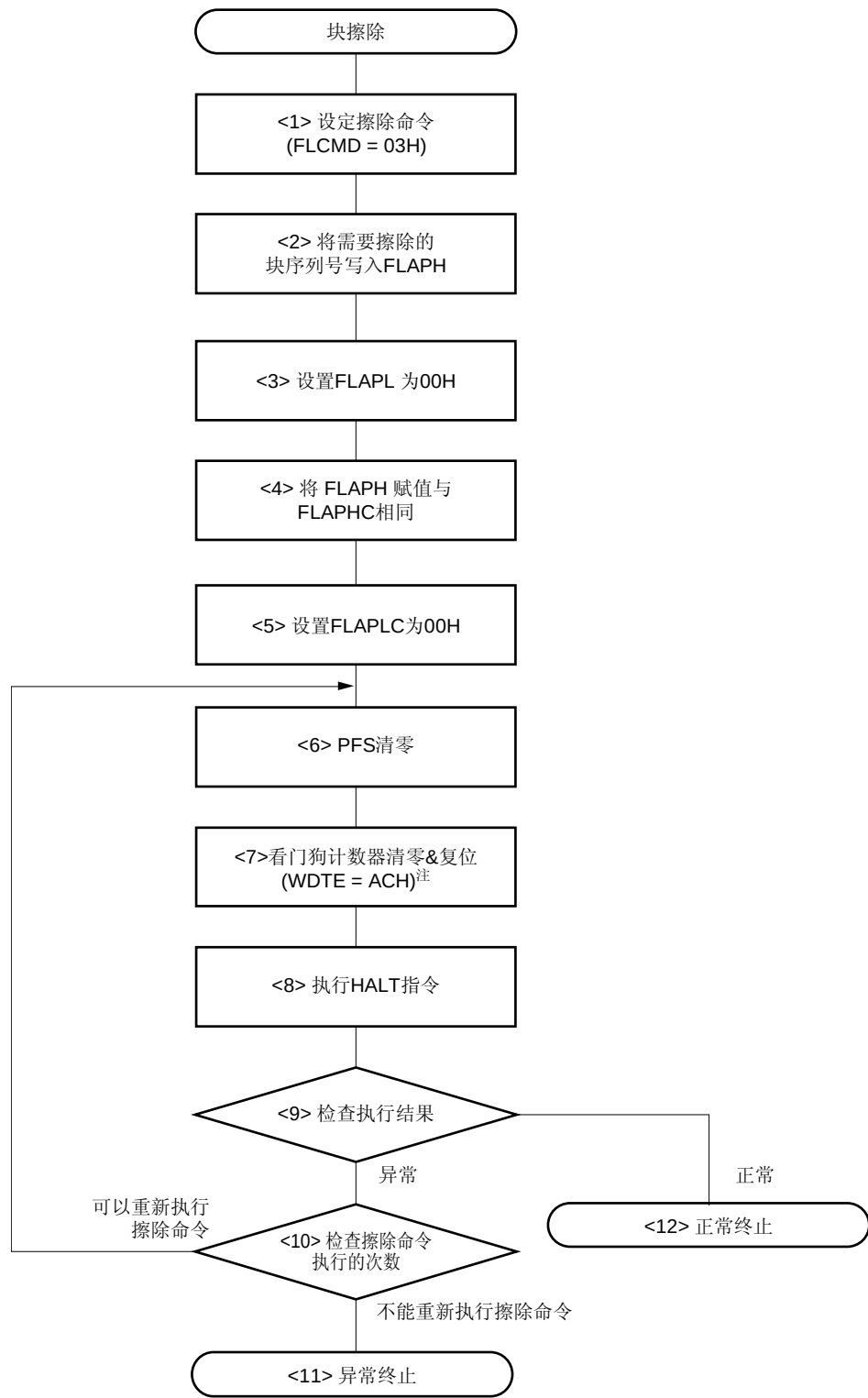
自编程模式下块擦除操作过程如下

- <1> 设置编程命令寄存器（FLCMD）值为 03H（块擦除）
- <2> 将需要擦除的块的序号值放到 FLASH 地址指针 H（FLAPH）中
- <3> 设置地址指针 L（FLAPL）为 00H
- <4> 将 FLAPH 的值写到地址指针 H 比较寄存器（FLAPHC）中
- <5> 设置地址指针 L 比较寄存器（FLAPLC）值为 00H
- <6> 清除状态寄存器（PFS）
- <7> 写 ACH 到看门狗定时器开放寄存器（WDTE）（清零并重启看门狗定时计数器）^{注1}
- <8> 执行 HALT 指令然后开始自编程（如果自编程已开始执行，则在 HALT 指令后立刻执行一条指令）
- <9> 用 PFS^{注2}的第 1 位（VCERR）和第 2 位（WEPRERR）检查自编程是否产生错误
 - 异常 → <10>
 - 正常 → <11>
- <10> 如果执行擦除命令的次数没有执行，返回到步骤 <6>，再次执行命令。如果执行擦除命令的次数执行，块擦除结束会有异常。
- <11> 块擦除处理异常结束。
- <12> 块擦除处理正常结束。

注 1.当不使用看门狗定时器时，无需对其进行设置。

2.在对外部禁止区域写入时候，单独检查 WEPRERR 位来检查执行中的错误。

图 19-20. 自编程模式下块擦除操作流程



注 当不使用看门狗定时器时，无需对其进行设置。

备注 图 19-20 中<1> 到<12> 对应于 19.8.6<1> 到<12>（前页）

例 自编程模式下块擦除程序

```

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

<R>          MOV      B,#48          ; Specifies the number of times the erase command can be
                                           ; executed.
                                           ; (4.0 V to 5.5 V Time for executing block erasure 100 times)

FlashBlockErase:
          MOV      FLCMD,#03H      ; Sets flash control command (block erase)
          MOV      FLAPH,#07H      ; Sets number of block to be erased (block 7 is specified here)
          MOV      FLAPL,#00H      ; Fixes FLAPL to "00H"
          MOV      FLAPHC,#07H     ; Sets erase block compare number (same value as that of FLAPH)
          MOV      FLAPLC,#00H     ; Fixes FLAPLC to "00H"

EraseRetry:
          MOV      PFS,#00H        ; Clears flash status register
          MOV      WDTE,#0ACH      ; Clears & restarts WDT
          HALT                    ; Self programming is started
          MOV      A,PFS
<R>          CMP      A,#00H        ; Checks execution result
          BZ        $StatusNormal ; Normal termination
          DBNZ      B,$EraseRetry ; Checks whether to re-execute the erase command.

;-----
;END (abnormal termination processing); Perform processing to shift to
normal mode in order to return to normal processing
;-----
StatusError:

;-----
;END (normal termination processing)
;-----
StatusNormal:

```

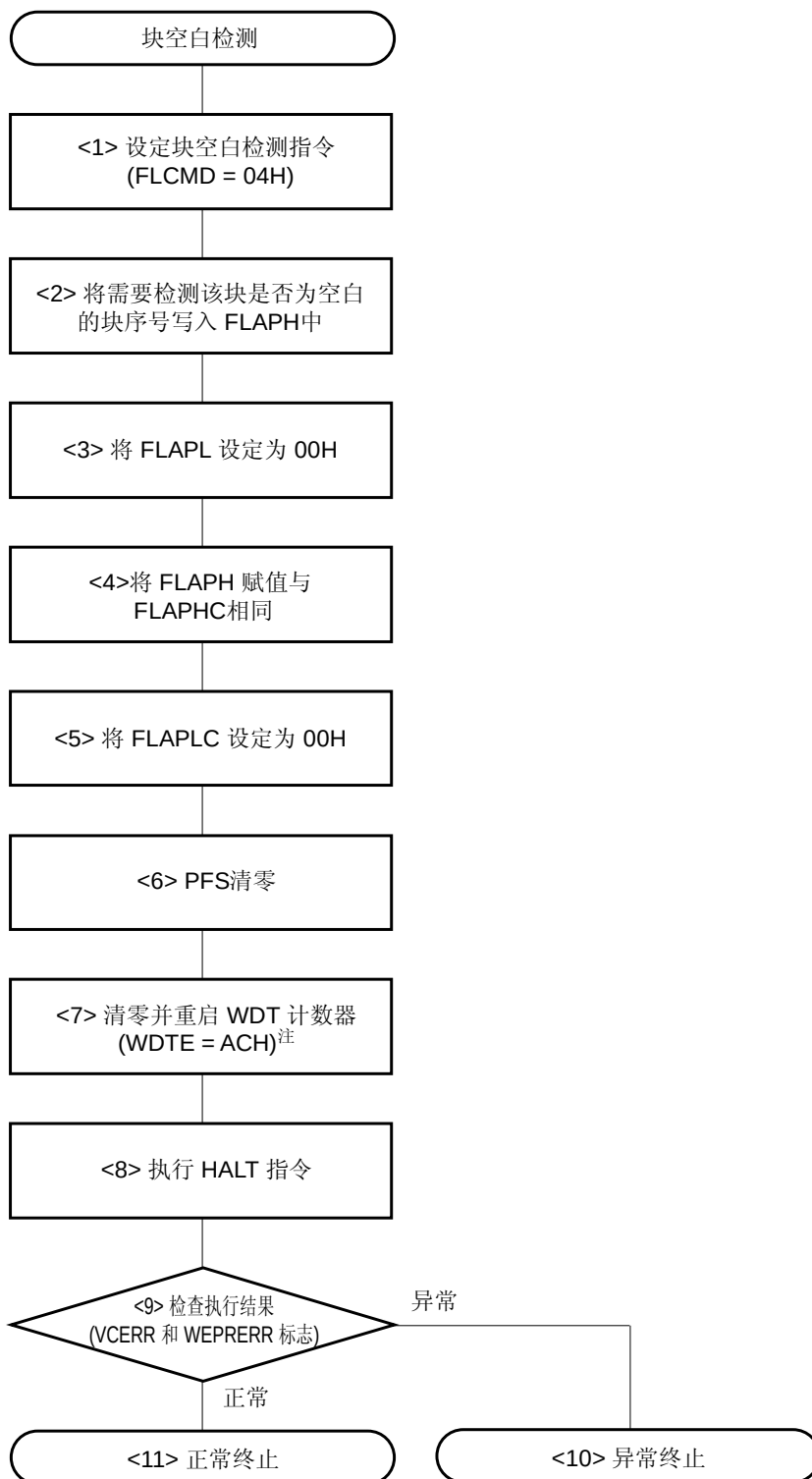
19.8.7 自编程模式下空白块的校验操作示例

自编程模式下空白块的校验操作过程如下

- <1> 设置编程命令寄存器（FLCMD）值为 04H（空白块校验）
- <2> 将需要执行空白块校验的块的序号放到 FLASH 地址指针 H 寄存器（FLAPH）中
- <3> 设置地址指针 L（FLAPL）为 00H
- <4> 将 FLAPH 的值写到处 FLASH 地址指针 H 比较寄存器（FLAPC）中
- <5> 设置 FLASH 地址指针 L 比较寄存器（FLAPLC）为 FFH
- <6> 清除 FLASH 状态寄存器（PFS）
- <7> 写 ACH 到看门狗定时器开放寄存器（WDTE）（清零并重启看门狗定时计数器）^注
- <8> 执行 HALT 指令然后开始自编程（如果自编程已开始执行，则在 HALT 指令后立刻执行一条指令）
- <9> 用 PFS 的第 1 位（VCERR）和第 2 位（WEPRERR）检查自编程是否产生错误
 - 错误 → <10>
 - 正常 → <11>
- <10> 空白块校验异常结束.
- <11> 空白块校验正常结束

注 当不使用看门狗定时器时，无需对其进行设置。

图 19-21. 在自编程模式下空白块校验操作流程



注 当不使用看门狗定时器时，无需对其进行设置。

备注 图 19-21 中<1> 到<11>对应于 19.8.7<1> 到<11>（前页）

例 自编程模式下空白块校验程序示例

```
; -----
; 开始
; -----
```

FlashBlockBlankCheck:

```
MOV    FLCMD, #04H    ; 设置 flash 控制命令（块 空白检查）
MOV    FLAPH, #07H     ; 设置空白检查的块序号（这里指定块 7）

MOV    FLAPL, #00H     ; FLAPL 恒为“00H”
MOV    FLAPHC, #07H    ; 设置空白检查 块 比较数（值与 FLAPH 相同）

MOV    FLAPLC, #0FFH   ; FLAPLC 恒为“FFH”

MOV    PFS, #00H       ; flash 状态寄存器清零
MOV    WDTE, #0ACH     ; 清零&复位 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: 正常结束， 除 0 外：异常结束）
```

```
; -----
; 结束
; -----
```

19.8.8 自编程模式下字节写入操作示例

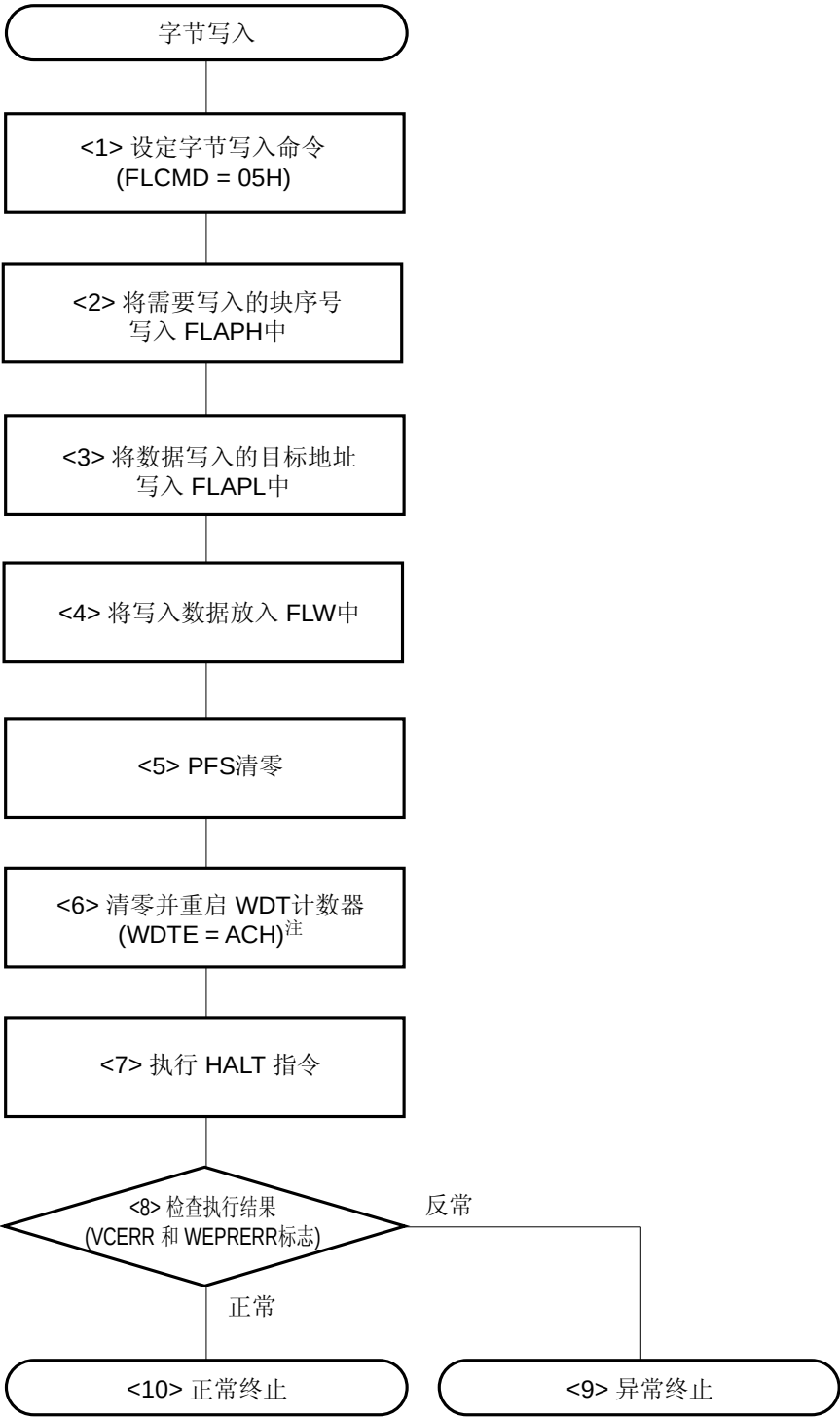
自编程模式下字节写入操作过程如下：

- <1> 设置编程命令寄存器（FLCMD）值为 05H（字节写入）
- <2> 将需要被写入数据的块序号放到 FLASH 地址指针 H 寄存器（FLAPH）中
- <3> 将需要被写入数据的目标地址放到 FLASH 地址指针 L 寄存器（FLAPL）中
- <4> 将需要写入的数据放到 FLASH 写入缓存寄存器（FLW）中
- <5> 清除 FLASH 状态寄存器（PFS）
- <6> 写 ACH 到看门狗定时器开放寄存器（WDTE）（清零并重启看门狗定时计数器）^注
- <7> 执行 HALT 指令然后开始自编程（如果自编程已开始执行，则在 HALT 指令后立刻执行一条指令）
- <8> 用 PFS 的第 1 位（VCERR）和第 2 位（WEPRERR）检查自编程是否产生错误。
 - 错误 → <9>
 - 正常 → <10>
- <9> 字节写入进程异常结束。
- <10> 字节写入进程正常结束。

注 当不使用看门狗定时器时，无需对其进行设置。

注意事项 如果写入失败，应该先擦除块内容并再次写入。

图 19-22. 自编程模式下字节写入操作流程



注 当不使用看门狗定时器时不必使用这个设置。

备注 图 19-22 中<1> 到<10> 对应于 19.8.8<1> 到<10>（前页）。

例 自编程模式下字节写入程序

```

; -----
; 开始
; -----
FlashWrite:
    MOV     FLCMD, #05H    ; 设置 flash 控制命令（字节写）
    MOV     FLAPH, #07H    ; 通过 FLAPH 设置数据要写入的地址（这里指定块 7）

    MOV     FLAPL, #20H    ; 通过 FLAPL 设置写入数据的地址（这里指定地址 20H）

    MOV     FLW, #10H      ; 设置要写入的数据（这里指定 10H）

    MOV     PFS, #00H      ; flash 状态寄存器清零
    MOV     WDTE, #0ACH    ; 清零&复位 WDT
    HALT                                ; 自编程开始
    MOV     A, PFS
    MOV     CmdStatus, A    ; 执行结果存储在变量中
                                ; （CmdStatus = 0: 正常结束， 除 0 外： 异常结束）

; -----
; 结束
; -----

```

19.8.9 自编程模式下内部校验操作示例

自编程模式下内部校验 1 和 2 的操作过程如下。

• 内部校验 1

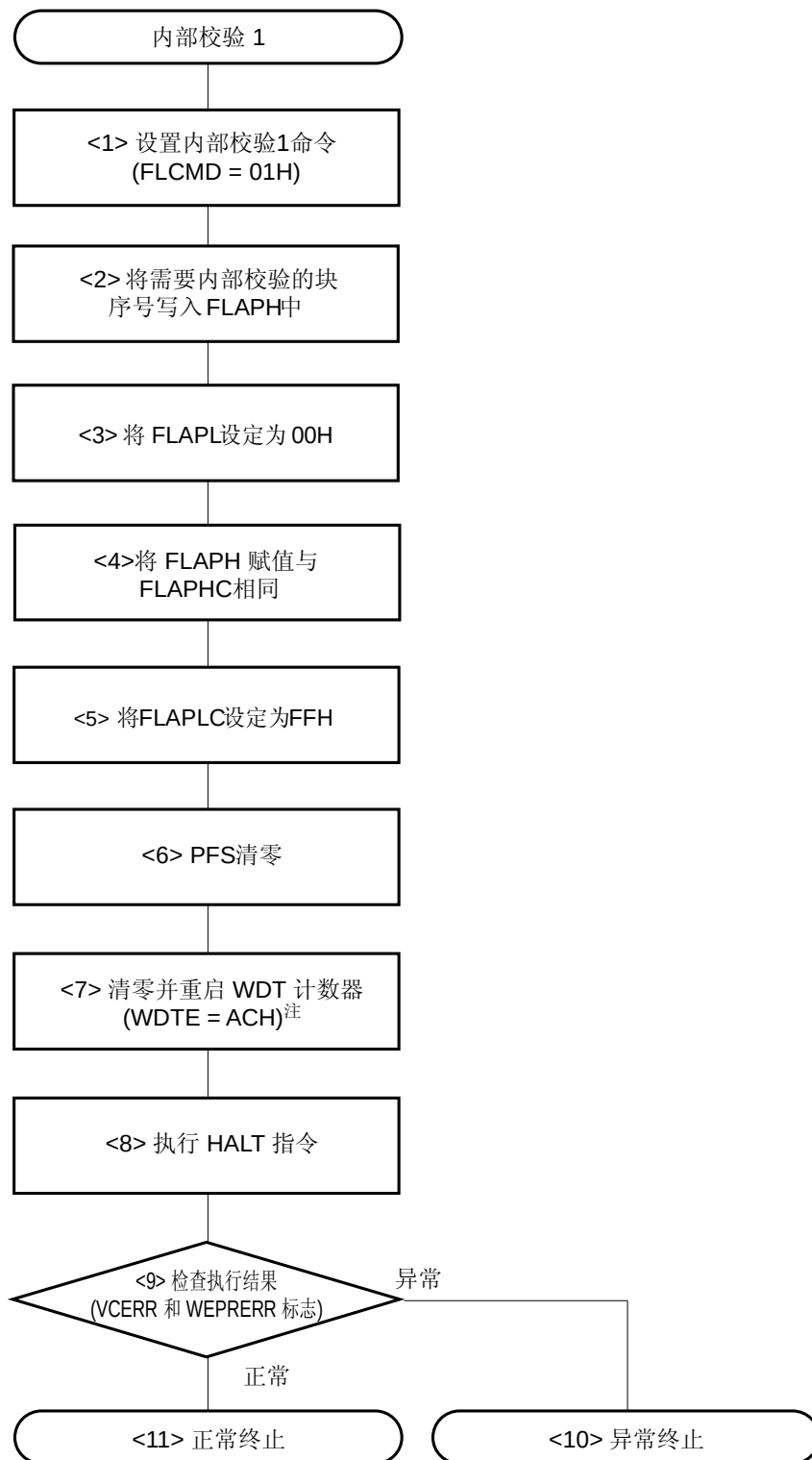
- <1> 设置编程命令寄存器（FLCMD）值为 01H（内部校验 1）。
- <2> 将需要执行内部校验的块的序号值放到到 FLASH 地址指针 H（FLAPH）中。
- <3> 设置 00H 到 flash 地址指针 L（FLAPL）。
- <4> 将 FLAPH 的值写到 flash 地址指针 H 比较寄存器（FLAPHC）。
- <5> 设置 FFH 到 flash 地址指针 L 比较寄存器（FLAPLC）。
- <6> flash 状态寄存器（PFS）清零。
- <7> 写 ACH 到看门狗定时器开放寄存器（WDTE）（清 0 并重启看门狗定时计数器）^注。
- <8> 执行 HALT 指令然后开始自编程（如果自编程已开始执行，则在 HALT 指令后立刻执行一条指令）。
- <9> 用 PFS 的第 1 位（VCERR）和第 2 位（WEPRERR）检查自编程是否产生错误。
 - 异常 → <10>
 - 正常 → <11>
- <10> 内部校验异常结束。
- <11> 内部校验正常结束。

• 内部校验 2

- <1> 设置编程命令寄存器（FLCMD）值为 02H（内部校验 2）。
- <2> 将需要执行内部校验的块的序号值放到到 FLASH 地址指针 H（FLAPH）中。
- <3> 设置校验起始地址到 flash 地址指针 L（FLAPL）。
- <4> 将 FLAPH 的值写到 flash 地址指针 H 比较寄存器（FLAPHC）。
- <5> 设置校验结束地址到 flash 地址指针 L 比较寄存器（FLAPLC）。
- <6> flash 状态寄存器（PFS）清零。
- <7> 写 ACH 到看门狗定时器开放寄存器（WDTE）（清 0 并重启看门狗定时计数器）^注。
- <8> 执行 HALT 指令然后开始自编程（如果自编程已开始执行，则在 HALT 指令后立刻执行一条指令）。
- <9> 用 PFS 的第 1 位（VCERR）和第 2 位（WEPRERR）检查自编程是否产生错误。
 - 异常 → <10>
 - 正常 → <11>
- <10> 内部校验异常结束。
- <11> 内部校验正常结束。

注 当不使用看门狗定时器时，无需对其进行设置。

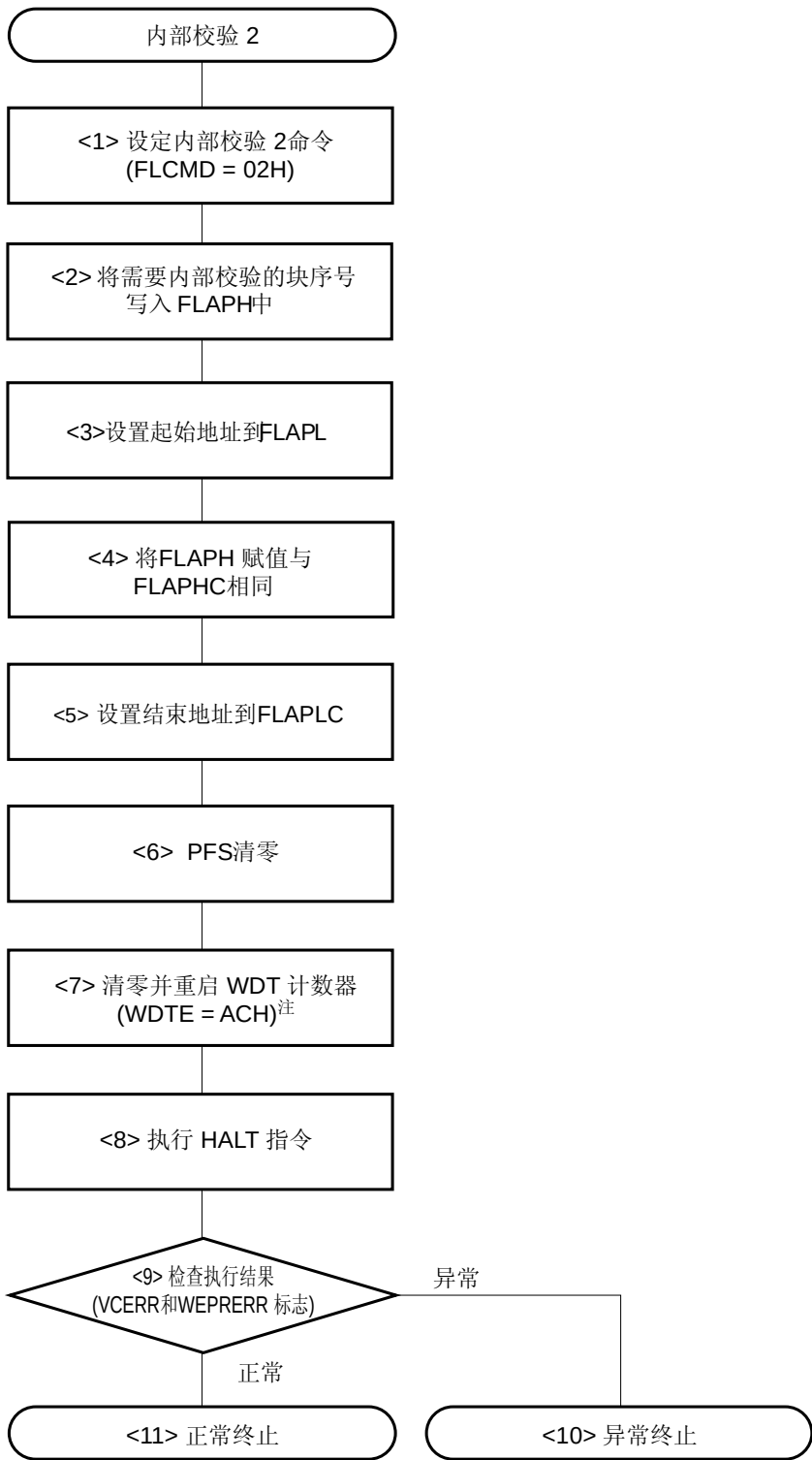
图 19-23. 自编程模式下内部校验 1 流程



注 当不使用看门狗定时器时，无需对其进行设置。

备注 图 19-23 中<1> 到<11> 对应于 19.8.9 内部校验 1 的<1> 到<11>（前页）。

图 19-24. 自编程模式下内部校验 2 流程



注 当不使用看门狗定时器时，无需对其进行设置。

备注 图 19-24 中<1> 到<11> 对应于 19.8.9 内部校验 2 的<1> 到<11>（前页）。

自编程模式下，内部校验 1 和 2 操作程序示例如下

- 内部校验 1

```
; -----
; 开始
; -----
FlashVerify:
    MOV     FLCMD, #01H    ; 设置 flash 控制命令（内部校验 1）
    MOV     FLAPH, #07H    ; 设置需要进行校验的块序号到 FLAPH（例：这里指定块 7）

    MOV     FLAPL, #00H    ; 设置 00H
    MOV     FLAPHC, #07H
    MOV     FLAPLC, #FFH   ; 设置 FFH

    MOV     PFS, #00H      ; flash 状态寄存器清零
    MOV     WDTE, #0ACH    ; 清零&复位 WDT
    HALT                                ; 开始自编程
    MOV     A, PFS
    MOV     CmdStatus, A    ; 执行结果保存在变量中
                                ; （CmdStatus = 0: 正常结束， 除 0 外： 异常结束）

; -----
; 结束
; -----;
```

- 内部校验 2

```
; -----
; 开始
; -----
FlashVerify:
    MOV     FLCMD, #02H    ; 设置 flash 控制命令（内部校验 2）
    MOV     FLAPH, #07H    ; 设置需要进行校验的块序号到 FLAPH（例：这里指定块 7）

    MOV     FLAPL, #00H    ; 设置 FLAPL 到开始地址校验（例：这里指定地址 00H）

    MOV     FLAPHC, #07H
    MOV     FLAPLC, #20H   ; 设置 FLAPLC 到结束地址校验（例：这里指定地址 20H）

    MOV     PFS, #00H      ; flash 状态寄存器清零
    MOV     WDTE, #0ACH    ; 清零&复位 WDT
    HALT                                ; 自编程开始
    MOV     A, PFS
    MOV     CmdStatus, A    ; 运行结果保存在变量中
                                ; （CmdStatus = 0: 正常结束， 除 0 外： 异常结束）

; -----
; 结束
; -----;
```

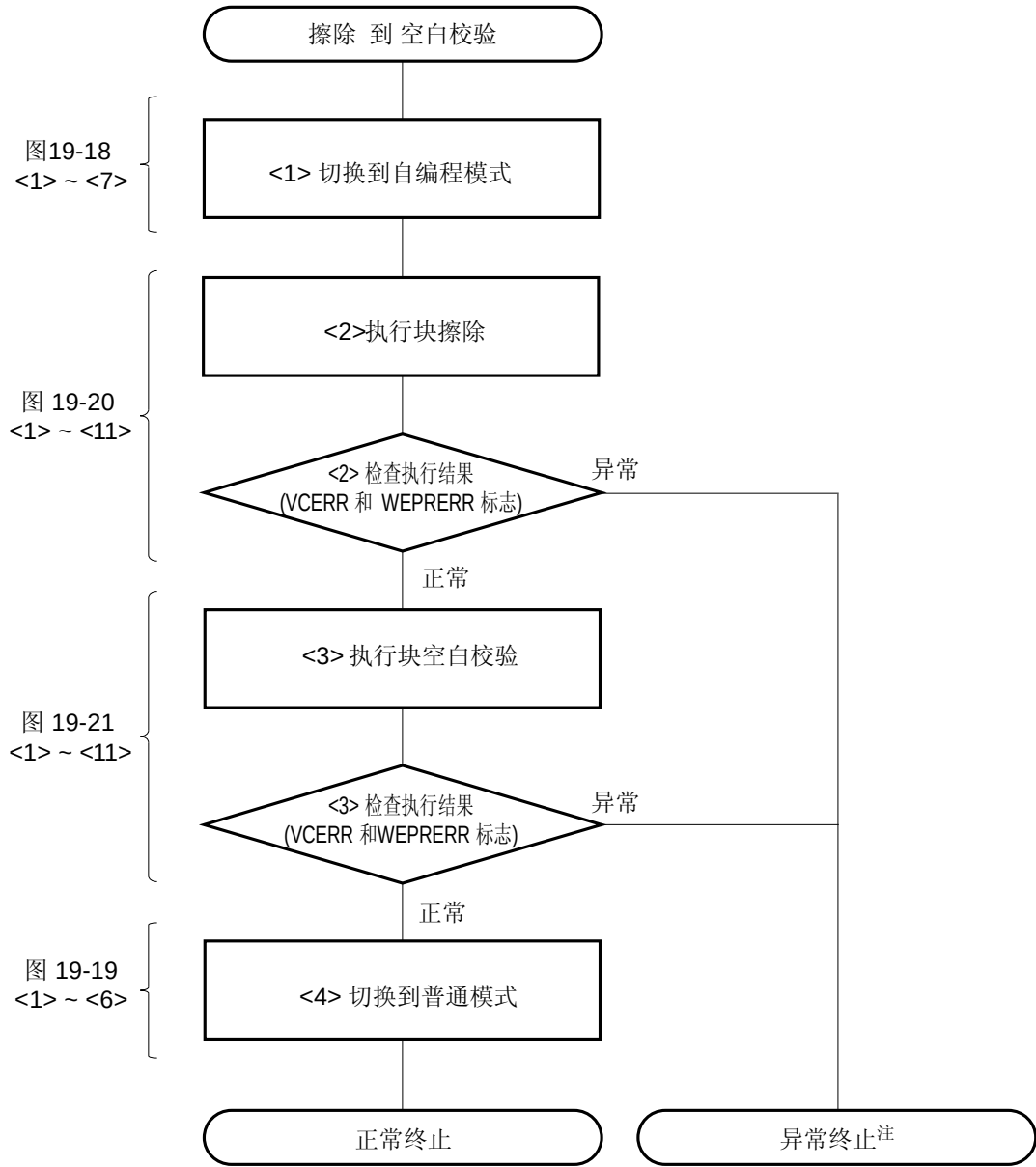
19.8.10 自编程模式下最短命令执行时间的操作示例

自编程模式下最短命令执行时间的操作过程如下：

(1) 从擦除到空白检验

- <1> 将工作模式由正常模式切换至自编程模式（在 19.8.4<1> 到 <7>）
- <2> 执行块擦除命令 → 错误检查（在 19.8.6<1>到<11>）
- <3> 执行块空白块检验 → 错误检查（在 19.8.7<1>到<11>）
- <4> 将工作模式由自编程模式切换回正常模式（在 19.8.5<1>到<6>）

图 19-25 最短命令执行时间的操作流程（从擦除到空白检验）



注 为了让程序返回到正常的处理中需将工作模式切换回正常模式。

备注 图 19-25 中<1> 到<4> 对应于 19.8.10<1> 到<4>。

自编程模式下，最短命令执行时间（从擦除到空白检验）的操作程序示例。

```

; -----
; 开始
; -----

    MOV     MK0, #11111111B    ; 屏蔽所有中断
    MOV     MK1, #11111111B
    MOV     FLCMD, #00H        ; 清零 FLCMD 寄存器

    DI

ModeOnLoop:
; 配置设置使 CPU 时钟 ≥ 1 MHz
    MOV     PFS, #00H          ; flash 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 寄存器控制设置自编程模式（设置值）
;

    NOP
    HALT
    BT      PFS.0, $ModeOnLoop ; 检查写入到指定寄存器完成
; 当出现错误时重复相同的处理

FlashBlockErase:
    MOV     FLCMD, #03H        ; 设置 flash 控制命令（块擦除）
    MOV     FLAPH, #07H        ; 设置要擦除的块序号（这里指定块 7）

    MOV     FLAPL, #00H        ; FLAPL 恒为“00H”
    MOV     FLAPHC, #07H       ; 设置块比较数（与 FLAPH 值相同）

    MOV     FLAPLC, #00H       ; FLAPLC 恒为“00H”

    MOV     WDTE, #0ACH        ; 清零&复位 WDT
    HALT                        ; 开始自编程
    MOV     A, PFS
    CMP     A, #00H
    BNZ     $StatusError       ; 检查擦除错误
; 发生错误时执行异常结束处理

FlashBlockBlankCheck:
    MOV     FLCMD, #04H        ; 设置 flash 控制命令（块空白检测）
    MOV     FLAPH, #07H        ; 设置空白检测的块序号（这里指定块 7）
    MOV     FLAPL, #00H        ; FLAPL 恒为“00H”

```

```

MOV    FLAPHC, #07H      ; 设置空白检测块比较数（与 FLAPH 值相同）
MOV    FLAPLC, #0FFH     ; FLAPLC 恒为 "FFH"
MOV    WDTE, #0ACH       ; 清零& 重启 WDT
HALT                                ; 自编程开始
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结束处理

MOV    FLCMD, #00H       ; FLCMD 寄存器清零
ModeOffLoop:
MOV    PFS, #00H         ; flash 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（对设置值取反）
MOV    FLPMC, #00H       ; 通过 FLPMC 寄存器控制设置正常模式）设置值

BT     PFS.0, $ModeOffLoop ; 检查写入指定寄存器完成
                                ; 发生错误时重复相同的处理
                                ; 指定序列正常完成后，自编程开始前恢复 CPU 时钟设置

MOV    MK0, #INT_MK0     ; 恢复中断屏蔽标志
MOV    MK1, #INT_MK1

EI

BR     StatusNormal

; -----
; END（异常结束处理）； 为了返回正常模式处理，执行切换到正常模式的处理
; -----
StatusError:

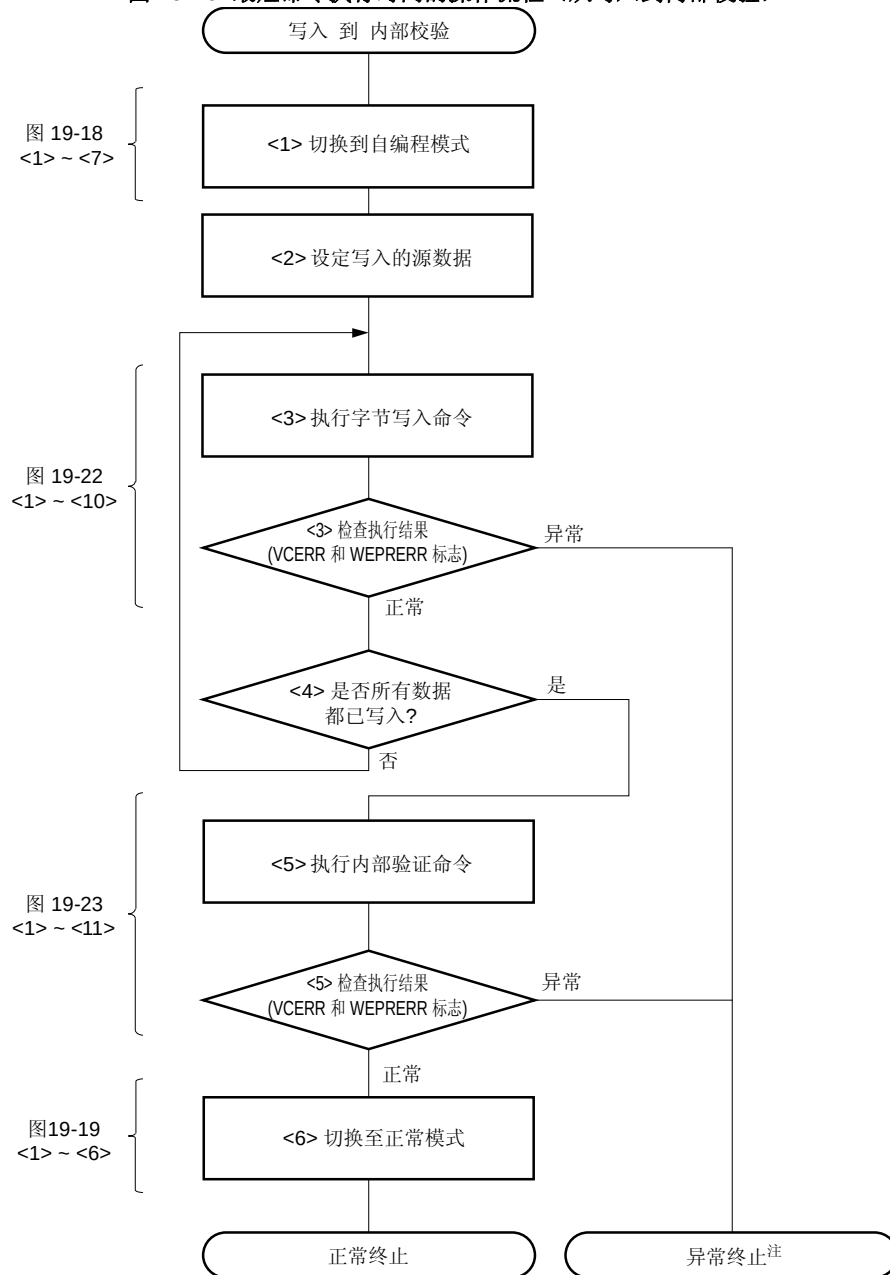
; -----
; END（正常结束处理）
; -----
StatusNormal:

```

(2) 从写入字节到内部校验

- <1> 将工作模式由正常模式切换至自编程模式（在 19.8.4<1> 到 <7>）
- <2> 确定需要写入的数据
- <3> 执行块字节写入指令 → 错误检查（在 19.8.8<1>到<10>）
- <4> 重复操作<3>直到将所有数据写入
- <5> 执行内部校验指令 → 错误检查（在 19.8.9<1>到<11>）
- <6> 将工作模式由自编程模式切换回正常模式（在 19.8.5<1>到<6>）

图 19-26 最短命令执行时间的操作流程（从写入到内部校验）



注 为了让程序返回到正常的处理中需将工作模式切换回正常模式。

备注 图 19-26 中<1> 到<6> 对应于 19.8.10 (2) 中的<1>到<6>。

自编程模式下最短命令执行时间（从写入到内部校验）的操作程序示例。

```

; -----
; 开始
; -----

    MOV     MK0, #11111111B    ; 屏蔽所有中断
    MOV     MK1, #11111111B
    MOV     FLCMD, #00H        ; FLCMD 寄存器清零

    DI

ModeOnLoop:                          ; 配置设置使 CPU 时钟 ≥ 1 MHz
    MOV     PFS, #00H          ; flash 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 寄存器控制设置自编程模式
                                ; （设置值）

    NOP
    HALT
    BT      PFS.0, $ModeOnLoop ; 检查写入到指定寄存器完成
                                ; 出现错误时重复相同的处理

FlashWrite:
    MOVW    HL, #DataAdrTop    ; 设置数据写入地址
    MOVW    DE, #WriteAdr      ; 设置数据要写入的地址
FlashWriteLoop:
    MOV     FLCMD, #05H        ; 设置 flash 控制命令（字节写）
    MOV     A, D
    MOV     FLAPH, A            ; 设置数据要写入的地址
    MOV     A, E
    MOV     FLAPL, A            ; 设置数据要写入的地址
    MOV     A, [HL]
    MOV     FLW, A              ; 设置要写入的数据

    MOV     WDTE, #0ACH        ; 清零 & 重启 WDT
    HALT                          ; 自编程开始
    MOV     A, PFS
    CMP     A, #00H
    BNZ     $StatusError        ; 检查写错误
                                ; 发生错误时执行异常结束处理

    INCW    HL                  ; 数据写入地址+1
    MOVW    AX, HL

```

CMPW	AX, #DataAdrBtm	: 执行内部校验处理
BNC	\$FlashVerify	: 如果所有数据写入完成
INCW	DE	: 数据要写入的地址 + 1
BR	FlashWriteLoop	

FlashVerify:

MOVW	HL, #WriteAdr	: 设置验证地址
MOV	FLCMD, #02H	: 设置 flash 控制命令 (内部校验 2)
MOV	A, H	
MOV	FLAPH, A	: 设置校验开始地址
MOV	A, L	
MOV	FLAPL, A	: 设置校验开始地址
MOV	A, D	
MOV	FLAPHC, A	: 设置校验结束地址
MOV	A, E	
MOV	FLAPLC, A	: 设置校验结束地址
MOV	WDTE, #0ACH	: 清零 & 重启 WDT
HALT		: 自编程开始
MOV	A, PFS	
CMP	A, #00H	
BNZ	\$StatusError	: 检查内部校验错误 : 出现错误时执行异常结束处理
MOV	FLCMD, #00H	: FLCMD 寄存器清零

ModeOffLoop:

MOV	PFS, #00H	: flash 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 (对设置值取反)
MOV	FLPMC, #00H	: 通过 FLPMC 寄存器控制设置正常模式 (设置值)
BT	PFS.0, \$ModeOffLoop	: 检查写入到指定寄存器完成 : 出现错误时重复相同处理 : 指定序列正常完成后, 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
MOV	MK1, #INT_MK1	
EI		
BR	StatusNormal	

```
; -----
; 结束（异常结束处理）； 要返回到正常程序处理，执行切换到正常模式的处理
; -----
StatusError:

; -----
; 结束（正常结束处理）
; -----
StatusNormal:

; -----
; 写入数据
; -----
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。

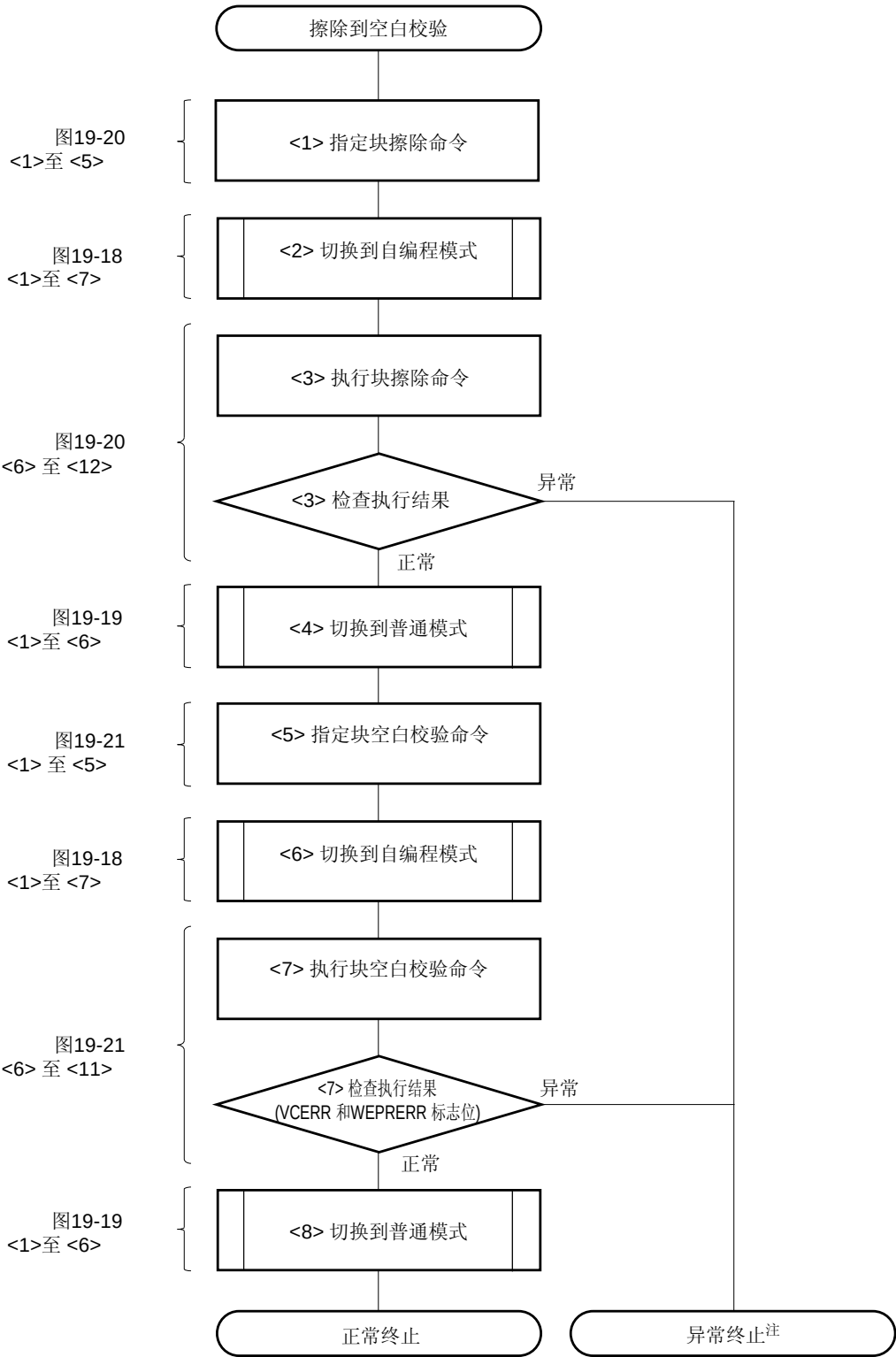
19.8.11 自编程模式下最短中断禁止时间的操作过程

自编程模式下最短中断禁止时间的操作过程如下：

(1) 从擦除到空白检验

- <1> 设置控制命令为块擦除（在 19.8.6 <1> 到 <5>）
- <2> 将工作模式由正常模式切换至自编程模式（在 19.8.4 <1>到<7>）
- <3> 执行块擦除命令 → 错误检查（在 19.8.6 <6>到<11>）
- <4> 将工作模式由自编程模式切换至正常模式（在 19.8.5 <1>到<6>）
- <5> 设置控制命令为空白块检验（在 19.8.7 <1>到<5>）
- <6> 将工作模式由正常模式切换至自编程模式（在 19.8.4 <1>到<7>）
- <7> 执行块空白块检验 → 错误检查（在 19.8.7 <6>到<11>）
- <8> 将工作模式由自编程模式切换回正常模式（在 19.8.5 <1>到<6>）

图 19-27 自编程模式下最短中断禁止时间的操作流程（从擦除到空白块检验）



注 为了让程序返回到正常的处理中，需将工作模式切换回正常模式。

备注 图 19-27 中<1> 到<8> 对应于 19.8.11<1> 到<8>（前页）

下面所示为自编程模式下最短中断禁止时间（从擦除到空白块检验）的程序示例。

```

;-----
;START
;-----
<R>      MOV      B,#48          ; Specifies the number of times the erase command can be
                                   ; executed.
                                   ; (4.0 V to 5.5 V Time for executing block erasure 100 times)

FlashBlockErase:
    ; Sets erase command
    MOV      FLCMD,#03H          ; Sets flash control command (block erase)
    MOV      FLAPH,#07H          ; Sets number of block to be erased (block 7 is specified here)
    MOV      FLAPL,#00H          ; Fixes FLAPL to "00H"
    MOV      FLAPHC,#07H         ; Sets erase block compare number (same value as that of FLAPH)
    MOV      FLAPLC,#00H         ; Fixes FLAPLC to "00H"

    CALL     !ModeOn             ; Shift to self programming mode

EraseRetry:
    ; Execution of erase command
    MOV      PFS,#00H            ; Clears flash status register
    MOV      WDTE,#0ACH          ; Clears & restarts WDT
    HALT                                ; Self programming is started
    MOV      A,PFS
    CMP      A,#00H              ; Checks execution result
    BNZ      $RetryCheck         ; Checks erase error
                                   ; Performs abnormal termination processing when an error
                                   ; occurs.

    CALL     !ModeOff            ; Shift to normal mode

    ; Sets blank check command
    MOV      FLCMD,#04H          ; Sets flash control command (block blank check)
    MOV      FLAPH,#07H          ; Sets block number for blank check (block 7 is specified here)
    MOV      FLAPL,#00H          ; Fixes FLAPL to "00H"
    MOV      FLAPHC,#07H         ; Sets blank check block compare number (same value as that of
                                   ; FLAPH)
    MOV      FLAPLC,#0FFH        ; Fixes FLAPLC to "FFH"

    CALL     !ModeOn             ; Shift to self programming mode

    ; Execution of blank check command
    MOV      PFS,#00H            ; Clears flash status register
    MOV      WDTE,#0ACH          ; Clears & restarts WDT
    HALT                                ; Self programming is started
    MOV      A,PFS
    CMP      A,#00H              ; Checks execution result

```

```

        BNZ      $StatusError    ; Checks blank check error
                                   ; Performs abnormal termination processing when an error occurs

        CALL     !ModeOff        ; Shift to normal mode

        BR       StatusNormal

<R> RetryCheck:
        DBNZ     B,$EraseRetry

;-----
;END (abnormal termination processing); Perform processing to shift to
        normal mode in order to return to normal processing
;-----
StatusError:

;-----
;END (normal termination processing)
;-----
StatusNormal:

;-----
;Processing to shift to self programming mode
;-----
ModeOn:
        MOV      MK0,#11111111B    ; Masks all interrupts
        MOV      MK1,#11111111B
        MOV      FLCMD,#00H        ; Clears FLCMD register

        DI

ModeOnLoop:
                                   ; Configure settings so that the CPU clock ≥ 1 MHz
        MOV      PFS,#00H          ; Clears flash status register
        MOV      PFCMD,#0A5H       ; PFCMD register control
        MOV      FLPMC,#01H        ; FLPMC register control (sets value)
        MOV      FLPMC,#0FEH       ; FLPMC register control (inverts set value)
        MOV      FLPMC,#01H        ; Sets self programming mode via FLPMC register control (sets
                                   ; value)

        NOP
        HALT

        BT       PFS.0,$ModeOnLoop ; Checks completion of write to specific registers
                                   ; Repeats the same processing when an error occurs

        RET

```

```

;-----
; Processing to shift to normal mode
;-----
ModeOffLoop:
    MOV     FLCMD,#00H      ; Clears FLCMD register
    MOV     PFS,#00H       ; Clears flash status register
    MOV     PFCMD,#0A5H    ; PFCMD register control
    MOV     FLPMC,#00H     ; FLPMC register control (sets value)
    MOV     FLPMC,#0FFH    ; FLPMC register control (inverts set value)
    MOV     FLPMC,#00H     ; Sets normal mode via FLPMC register control (sets value)

    BT      PFS.0,$ModeOffLoop ; Checks completion of write to specific registers
                                ; Repeats the same processing when an error occurs
                                ; Restore the CPU clock to its setting before the self
                                ; programming, after normal completion of the specific
                                ; sequence

    MOV     MK0,#INT_MK0    ; Restores interrupt mask flag
    MOV     MK1,#INT_MK1

    EI

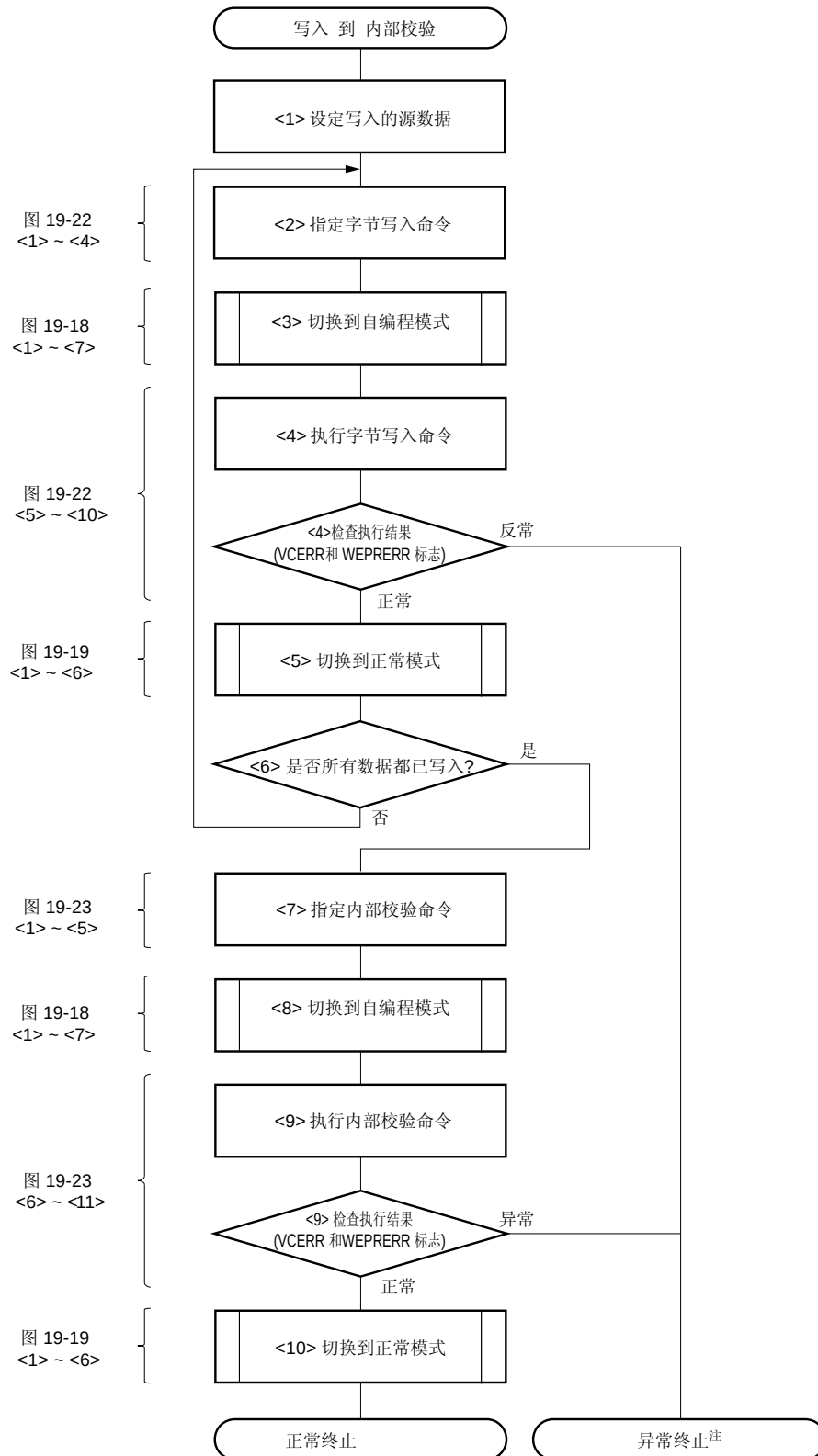
    RET

```

(2) 从写入字节到内部校验

- <1> 设置需要写入的数据
- <2> 设置控制命令为字节写入（在 **19.8.8**<1> 到 <4>）
- <3> 将工作模式由正常模式切换到自编程模式（在 **19.8.4**<1> 到 <7>）
- <4> 执行字节写入命令 → 错误检查（在 **19.8.8**<5> 到 <10>）
- <5> 将工作模式由自编程模式切换回正常模式（在 **19.8.5**<1>到 <6>）
- <6> 重复操作<2>到<5>直到将所有数据写入
- <7> 设定内部校验命令 （在 **19.8.9**<1>到<5>）
- <8> 将工作模式由正常模式切换到自编程模式（在 **19.8.4**<1> 到 <7>）
- <9> 执行内部校验指令 → 错误检查（在 **19.8.9**<6> 到 <11>）
- <10>将工作模式由自编程模式切换回正常模式（在 **19.8.5**<1>到<6>）

图 19-28. 自编程模式下最短中断禁止时间的操作流程（从写入到内部校验）



注 为了让程序返回到正常的处理中需将工作模式切换回正常模式。

备注 图 19-28 中<1> 到<10> 是相应于 19.8.11<1> 到<10>（前页）。

如下所示为自编程模式下最短中断禁止时间（从写入到内部校验）的操作程序示。

```

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

; 设置写入命令
FlashWrite:
    MOVW    HL, #DataAdrTop ; 设置数据要写入的地址
    MOVW    DE, #WriteAdr   ; 设置数据要写入的地址

FlashWriteLoop:
    MOV     FLCMD, #05H      ; 设置 flash 控制命令（字节写）
    MOV     A, D
    MOV     FLAPH, A         ; 设置数据要写入的地址
    MOV     A, E
    MOV     FLAPL, A         ; 设置数据要写入的地址
    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 状态寄存器清零
    MOV     WDTE, #0ACH      ; 清零&重启 WDT
    HALT                                ; 自编程开始
    MOV     A, PFS
    CMP     A, #00H
    BNZ     $StatusError     ; 检查写错误
                                ; 出现错误时执行异常结束处理

    CALL    !ModeOff          ; 切换到正常模式

    MOV     MK0, #INT_MK0    ; 恢复中断屏蔽标志
    MOV     MK1, #INT_MK1

    EI

; 判断是否所有数据写入
    INCW    HL                ; 数据写入地址 + 1
    MOVW    AX, HL
    CMPW    AX, #DataAdrBtm ; 执行内部校验处理
    BNC     $FlashVerify     ; 如果所有数据写入完成

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

```


; 设置内部校验命令

FlashVerify:

```

MOVW    HL, #WriteAdr    ; 设置校验地址

MOV     FLCMD, #02H      ; 设置 flash 控制命令（内部校验 2）
MOV     A, H
MOV     FLAPH, A          ; 设置校验开始地址
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

```

; 执行内部校验命令

```

MOV     PFS, #00H        ; Flash 状态寄存器清零
MOV     WDTE, #0ACH       ; 清零&重启 WDT
HALT                                ; 自编程开始

MOV     A, PFS
CMP     A, #00H
BNZ     $StatusError      ; 检查内部校验错误
                                ; 出现错误时执行异常结束处理

CALL    !ModeOff          ; 切换到正常模式

BR      StatusNormal

```

```

; -----
; 结束（异常结束处理）；要返回到正常程序处理，执行切换到正常模式的处理
; -----

```

StatusError:

```

; -----
; 结束（正常结束处理）
; -----

```

StatusNormal:

```

; -----
; 切换到自编程模式
; -----
ModeOn:
    MOV     MK0, #11111111B ; 屏蔽所有中断
    MOV     MK1, #11111111B
    MOV     FLCMD, #00H ; FLCMD 寄存器清零

    DI

ModeOnLoop:
; 配置设置使 CPU 时钟 ≥ 1 MHz
    MOV     PFS, #00H ; Flash 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 寄存器控制设置自编程模式 (设置值)

    NOP
    HALT
    BT      PFS.0, $ModeOnLoop ; 检查写入到指定寄存器是否完成
; 出现错误时重复相同的处理

    RET

; -----
; 切换到正常模式
; -----
ModeOffLoop:
    MOV     FLCMD, #00H ; FLCMD 寄存器清零
    MOV     PFS, #00H ; Flash 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 (对设置值取反)
    MOV     FLPMC, #00H ; 通过 FLPMC 寄存器控制设置正常模式 (设置值)

    BT      PFS.0, $ModeOffLoop ; 检查写入到指定寄存器是否完成
; 出现错误时重复相同的处理
; 指定序列正常完成后, 恢复在自编程前的 CPU 时钟设置

    MOV     MK0, #INT_MK0 ; 恢复中断屏蔽标志
    MOV     MK1, #INT_MK1

    EI

```

RET

```

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

    :
    :

    DB      XXH
DataAdrBtm:
; -----
    
```

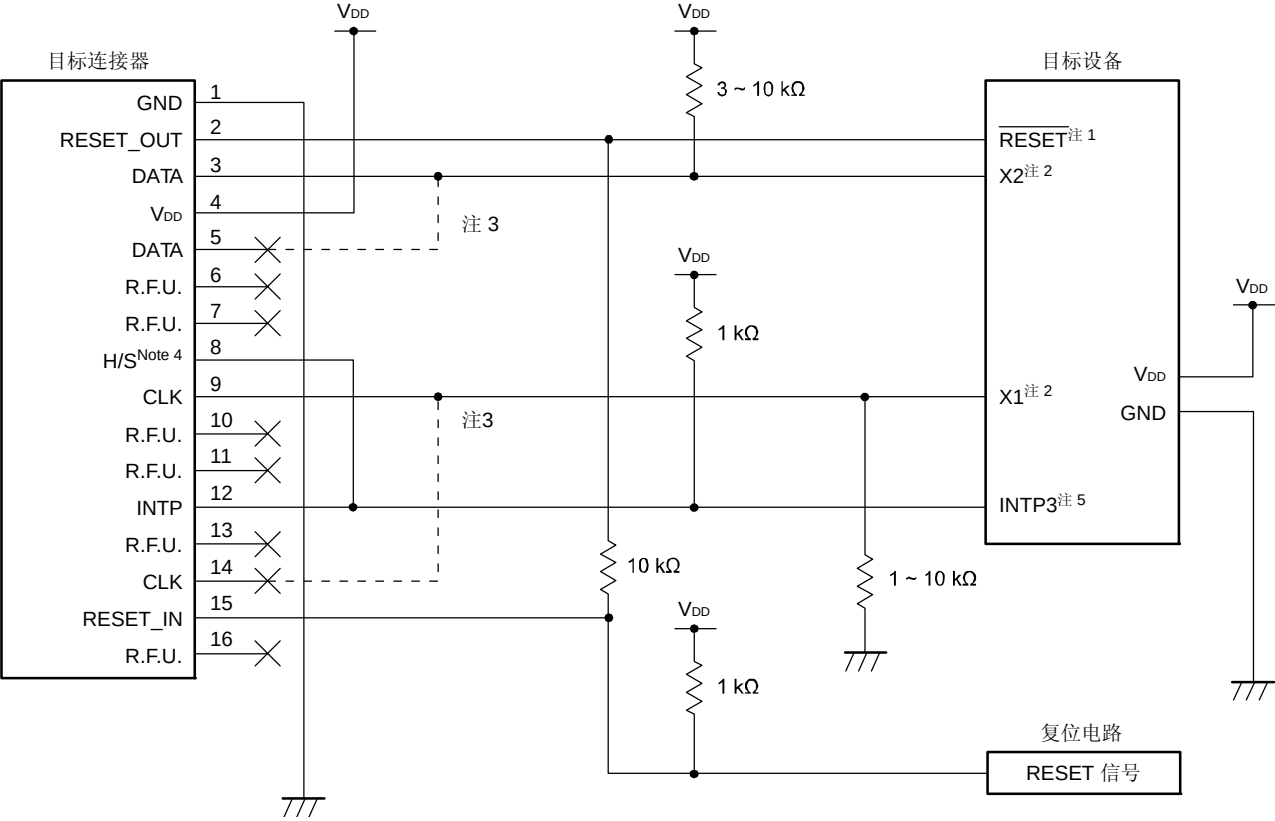
备注 在上面的程序例子中使用内部校验 2。验证一个完整的块，使用内部校验 1。

20.1 连接QB-MINI2 和 78K0S/KB1+

78K0S/KB1+ 通过片上调试仿真器（QB-MINI2）和主机通讯，使用 $\overline{\text{RESET}}$ ，X1，X2，INTP3，V_{DD}，和 GND 引脚。

注意事项 **The 78K0S/KB1+ 具有片上调试功能，可用于产品的开发和评估。在用于量产时，不要使用产品的片上调试功能，因为使用该功能的话，可能会超过 Flash 存储器的保证重写次数，这样的产品可靠性无法保障。使用片上调试功能时，NEC 电子不接受本产品的相关投诉。**

图20-1. 推荐电路连接



注意事项 在电路连接示例中描述的常数值都是参考值。如果打算进行大批量生产的 flash 编程，请认真评估目标设备的规格是否能够满足要求。

- 注**
- 1. 在调试器启动或者执行强制复位时， $\overline{\text{RESET}}$ 引脚被用来下载监控程序。因此，该引脚作为 $\overline{\text{RESET}}$ 引脚的复用功能不能使用。关于 reset 引脚的连接，请参考 **QB-MINI2 用户手册（U18371E）**。
 - 2. 这是在目标系统中未使用 X1 和 X2 时的引脚连接。当使用 X1 和 X2 引脚时，请参考 **20.1.2 X1 和 X2 的引脚连接**。
 - 3. 如果连接虚线部分，也不会出现问题。
 - 4. 连接该引脚可增强调试时 run 和 break 之间的时间测量精度。即使该引脚开路，也可以调试，但是会产生几个 ms 单位的测量误差。

注 5. 在调试期间，INTP3 引脚用来 MINICUBE2 和目标设备间的通讯。因此，当使用 MINICUBE2 进行调试时，INTP3 引脚和它的复用功能引脚都不能使用。对于 INTP3 引脚的连接，请参考 20.1.1 INTP3 引脚的连接。

通讯使用的引脚决定于监控程序是否已经被写入（请参考表 20-1）。X1 和 X2 引脚可以被用作 I/O 端口引脚或振荡引脚，在监控程序已经写入之后。

表 20-1. QB-MINI2 通讯使用的引脚

监控程序写入之前	监控程序写入之后
X1, X2, RESET, INTP3, VDD, VSS	RESET, INTP3, VDD, VSS

20.1.1 INTP3 引脚的连接

INTP3 引脚只能用于 QB-MINI2 和目标设备之间调试时的通讯。设计电路时，根据实际情况从下列选择最相符的电路。

- (1) 目标系统未使用 INTP3 引脚（如图 20-1. 推荐电路连接所示）
→ 参见 图 20-2。
- (2) QB-MINI2 仅用来编程，不作调试
→ 参见 图 20-3。
- (3) QB-MINI2 用来调试，INTP3 引脚的调试只能通过实际设备来执行
→ 参见 图 20-4。

图20-2. 当目标系统未使用INTP3引脚时的线路连接

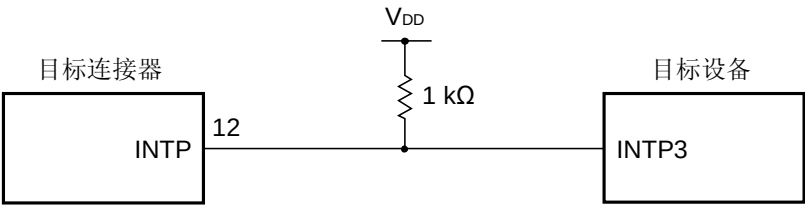
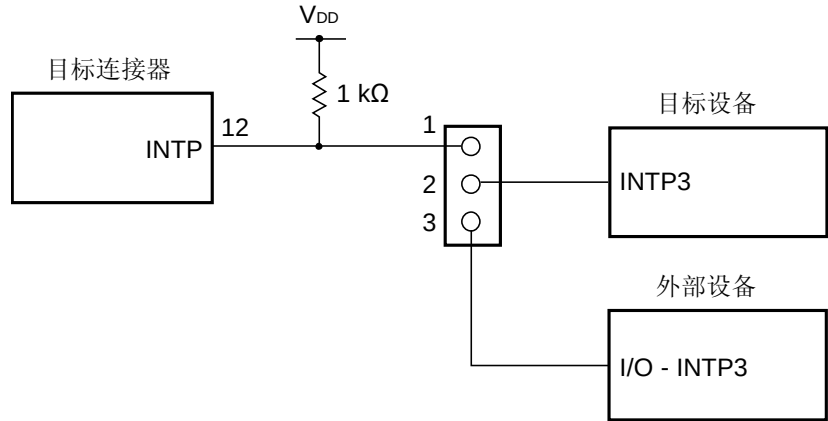


图20-3. QB-MINI2仅用作编程时的线路连接



图20-4. QB-MINI2用来调试且INTP3引脚的调试只能通过实际设备来执行时的线路连接



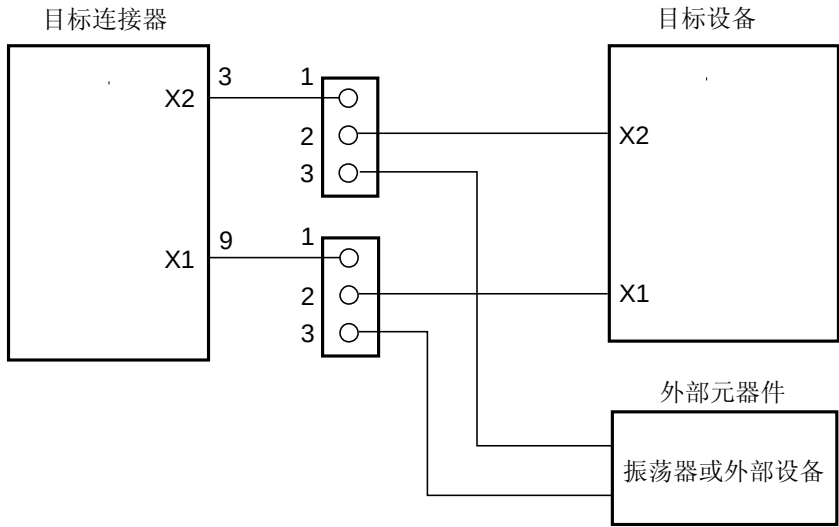
* 跳线设置
连接 QB-MINI2 进行调试：1-2 短接
其他情况：2-3 短接

注意事项 如果不使用 QB-MINI2 而是通过实机运行来调试，使用 QB-Programmer 写入用户程序。通过调试器下载的程序包括监控程序，如果不通过 QB-MINI2 控制，该程序会有故障。

20.1.2 X1 和X2 引脚的连接

调试器第一次启动（下载监控程序）时，使用 X1 和 X2 引脚。用 QB-Programmer 编程时，使用 X1 和 X2 引脚。

图20-5. 当目标系统使用X1和X2引脚的情况下电路连接



* 跳线设置
第一次启动调试器
（下载监控程序）或者当
用 QB-Programmer 编程时：1-2 短接
其他情况：2-3 短接

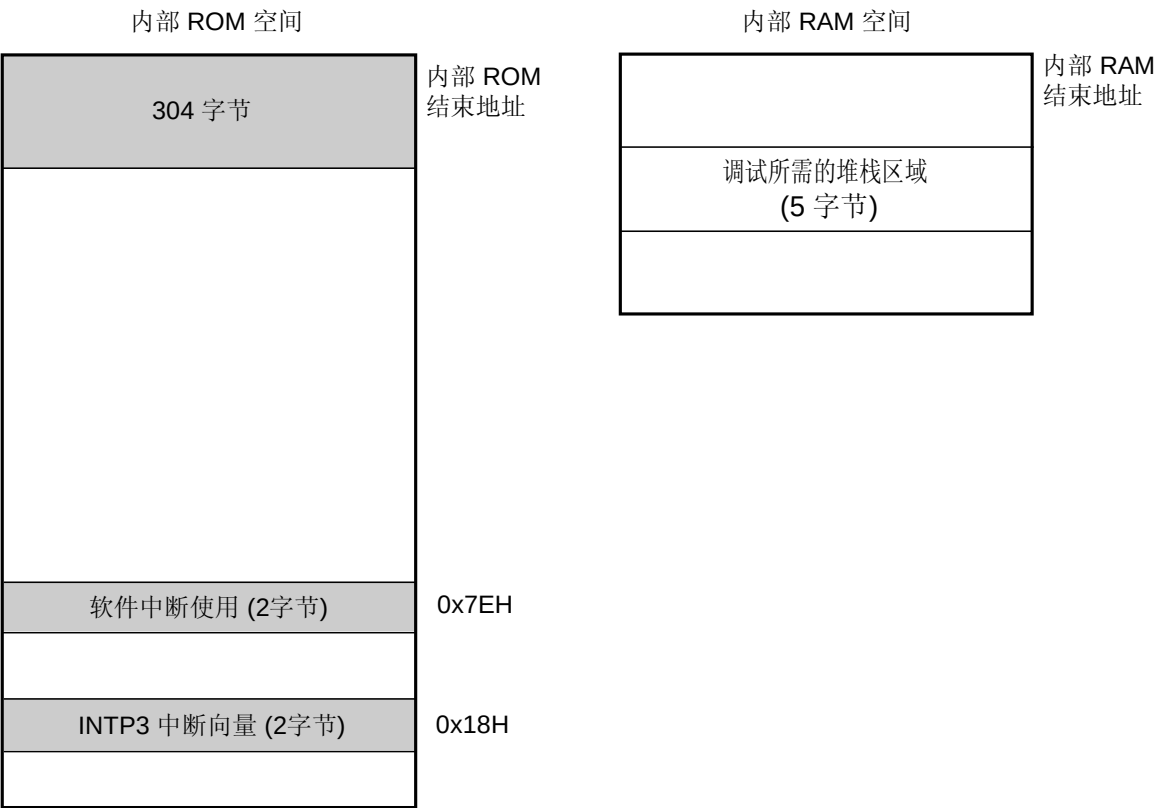
20.2 用户资源的保留

用户必须准备下面的事项来执行 QB-MINI2 和目标设备之间的通讯，并实现每一个调试功能。设置的详细信息，请参考 **QB-MINI2 用户手册（U18371E）**。

□ 存储器空间的保留

图 20-6 中的阴影部分是存放调试监控程序的保留区域，因此用户程序不能占用到这些空间。

图20-6. 存放调试监控程序的存储器空间



□ 通讯串行接口的保留

千万不要改变由调试监控程序执行，用作 QB-MINI2 与目标设备之间通讯的 INTP3 引脚的寄存器设置。

第二十一章 指令集概述

本章列出 78K0S/KB1+ 的指令集。如需了解每种指令的操作和机器语言（操作代码），请参考 **78K/0S 系列指令用户手册 (U11047E)**。

21.1 操作

21.1.1 操作数标识符和描述方法

按照指令操作数标识符的描述方法(详情请参考汇编程序规范),操作数在每种指令的“操作数”栏列出。如果有两种或两种以上的描述方法，可任选其一。大写字母和符号 #, !, \$, 和[] 是关键字，必须按其原样描述。每种符号的含义如下。

- #: 立即数标识
- !: 绝对地址标识
- \$: 相对地址标识
- []: 间接地址标识

立即数用来描述一个数值型数据或标号。当使用标号时，确保描述时加上符号#, !, \$ 和[]。

对于操作数寄存器标识符 r 和 rp, 可以使用功能名称(X, A, C 等)或绝对名称(下表括号中的名称，R0, R1, R2 等) 来描述。

表 21-1. 操作数标识符和描述方法

标识符	描述方法
r rp sfr	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) 特殊功能寄存器符号
saddr saddrp	FE20H ~ FF1FH 立即数或标号 FE20H ~ FF1FH 立即数或标号(仅用于偶地址)
addr16 addr5	0000H ~ FFFFH 立即数或标号(仅用于 16 位数据传输指令的偶地址) 0040H ~ 007FH 立即数或标号(仅用于偶地址)
字 字节 位	16 位立即数或标号 8 位立即数或标号 3 位立即数或标号

备注 特殊功能寄存器符号，参见表 3-3 特殊功能寄存器。

21.1.2 “操作”栏的描述

A:	A 寄存器; 8 位累加器
X:	X 寄存器
B:	B 寄存器
C:	C 寄存器
D:	D 寄存器
E:	E 寄存器
H:	H 寄存器
L:	L 寄存器
AX:	AX 寄存器组; 16 位累加器
BC:	BC 寄存器组
DE:	DE 寄存器组
HL:	HL 寄存器组
PC:	程序计数器
SP:	堆栈指针
PSW:	程序状态字
CY:	进位标志
AC:	辅助进位标志
Z:	零标志
IE:	中断请求使能标志
():	括号中的地址或寄存器所指的存储单元的元内容
×H, ×L:	16 位寄存器的高 8 位和低 8 位
∧:	逻辑与(AND)
∨:	逻辑或(OR)
⊕:	逻辑异或
—:	数据取反
addr16:	16 位立即数或标号
jdisp8:	带符号的八位数据(偏移量)

21.1.3 “标志”栏的描述

空白	不改变
0:	清为 0
1:	置为 1
×:	根据结果置位/清零
R:	恢复先前保存的值

21.2 操作列表

助记符	操作数	字节数	时钟	操作	标志
					Z AC CY
MOV	r, #byte	3	6	$r \leftarrow \text{byte}$	
	saddr, #byte	3	6	$(\text{saddr}) \leftarrow \text{byte}$	
	sfr, #byte	3	6	$\text{sfr} \leftarrow \text{byte}$	
	A, r 注 1	2	4	$A \leftarrow r$	
	r, A 注 1	2	4	$r \leftarrow A$	
	A, saddr	2	4	$A \leftarrow (\text{saddr})$	
	saddr, A	2	4	$(\text{saddr}) \leftarrow A$	
	A, sfr	2	4	$A \leftarrow \text{sfr}$	
	sfr, A	2	4	$\text{sfr} \leftarrow A$	
	A, !addr16	3	8	$A \leftarrow (\text{addr16})$	
	!addr16, A	3	8	$(\text{addr16}) \leftarrow A$	
	PSW, #byte	3	6	$\text{PSW} \leftarrow \text{byte}$	x x x
	A, PSW	2	4	$A \leftarrow \text{PSW}$	
	PSW, A	2	4	$\text{PSW} \leftarrow A$	x x x
	A, [DE]	1	6	$A \leftarrow (\text{DE})$	
	[DE], A	1	6	$(\text{DE}) \leftarrow A$	
	A, [HL]	1	6	$A \leftarrow (\text{HL})$	
	[HL], A	1	6	$(\text{HL}) \leftarrow A$	
	A, [HL + byte]	2	6	$A \leftarrow (\text{HL} + \text{byte})$	
	[HL + byte], A	2	6	$(\text{HL} + \text{byte}) \leftarrow A$	
XCH	A, X	1	4	$A \leftrightarrow X$	
	A, r 注 2	2	6	$A \leftrightarrow r$	
	A, saddr	2	6	$A \leftrightarrow (\text{saddr})$	
	A, sfr	2	6	$A \leftrightarrow \text{sfr}$	
	A, [DE]	1	8	$A \leftrightarrow (\text{DE})$	
	A, [HL]	1	8	$A \leftrightarrow (\text{HL})$	
	A, [HL, byte]	2	8	$A \leftrightarrow (\text{HL} + \text{byte})$	

注 1. r = A 除外。
2. r = A, X。除外"

备注 一个指令时钟周期是指一个 CPU 时钟周期(f_{CPU})，由处理器时钟控制寄存器(PCC)选择。

助记符	操作数	字节数	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 一个指令时钟周期是指一个 CPU 时钟周期(f_{CPU})，由处理器时钟控制寄存器(PCC)选择。

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

备注 一个指令时钟周期是指一个 CPU 时钟周期(f_{CPU})，由处理器时钟控制寄存器(PCC)选择。

助记符	操作数	字节数	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	$r \leftarrow r + 1$	×	×	
	saddr	2	4	$(saddr) \leftarrow (saddr) + 1$	×	×	
DEC	r	2	4	$r \leftarrow r - 1$	×	×	
	saddr	2	4	$(saddr) \leftarrow (saddr) - 1$	×	×	
INCW	rp	1	4	$rp \leftarrow rp + 1$			
DECW	rp	1	4	$rp \leftarrow rp - 1$			
ROR	A, 1	1	2	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$			×
ROL	A, 1	1	2	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$			×
RORC	A, 1	1	2	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			×
ROLC	A, 1	1	2	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			×
SET1	saddr.bit	3	6	$(saddr.bit) \leftarrow 1$			
	sfr.bit	3	6	$sfr.bit \leftarrow 1$			
	A.bit	2	4	$A.bit \leftarrow 1$			
	PSW.bit	3	6	$PSW.bit \leftarrow 1$	×	×	×
	[HL].bit	2	10	$(HL).bit \leftarrow 1$			
CLR1	saddr.bit	3	6	$(saddr.bit) \leftarrow 0$			
	sfr.bit	3	6	$sfr.bit \leftarrow 0$			
	A.bit	2	4	$A.bit \leftarrow 0$			
	PSW.bit	3	6	$PSW.bit \leftarrow 0$	×	×	×
	[HL].bit	2	10	$(HL).bit \leftarrow 0$			
SET1	CY	1	2	$CY \leftarrow 1$			1
CLR1	CY	1	2	$CY \leftarrow 0$			0
NOT1	CY	1	2	$CY \leftarrow \overline{CY}$			×

备注 一个指令时钟周期是指一个 CPU 时钟周期(f_{CPU})，由处理器时钟控制寄存器(PCC)选择。

助记符	操作数	字节数	时钟	操作	标志
					Z AC CY
CALL	!addr16	3	6	$(SP - 1) \leftarrow (PC + 3)_H, (SP - 2) \leftarrow (PC + 3)_L,$ $PC \leftarrow \text{addr16}, SP \leftarrow SP - 2$	
CALLT	[addr5]	1	8	$(SP - 1) \leftarrow (PC + 1)_H, (SP - 2) \leftarrow (PC + 1)_L,$ $PC_H \leftarrow (00000000, \text{addr5} + 1),$ $PC_L \leftarrow (00000000, \text{addr5}), SP \leftarrow SP - 2$	
RET		1	6	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP), SP \leftarrow SP + 2$	
RETI		1	8	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP), PSW \leftarrow (SP + 2),$ $SP \leftarrow SP + 3$	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	$PC \leftarrow PC + 2 + \text{jdisp8}$ if CY = 1	
BNC	\$saddr16	2	6	$PC \leftarrow PC + 2 + \text{jdisp8}$ if CY = 0	
BZ	\$saddr16	2	6	$PC \leftarrow PC + 2 + \text{jdisp8}$ if Z = 1	
BNZ	\$saddr16	2	6	$PC \leftarrow PC + 2 + \text{jdisp8}$ if Z = 0	
BT	saddr.bit, \$addr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ if (saddr.bit) = 1	
	sfr.bit, \$addr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ if sfr.bit = 1	
	A.bit, \$addr16	3	8	$PC \leftarrow PC + 3 + \text{jdisp8}$ if A.bit = 1	
	PSW.bit, \$addr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ if PSW.bit = 1	
BF	saddr.bit, \$addr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ if (saddr.bit) = 0	
	sfr.bit, \$addr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ if sfr.bit = 0	
	A.bit, \$addr16	3	8	$PC \leftarrow PC + 3 + \text{jdisp8}$ if A.bit = 0	
	PSW.bit, \$addr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ if PSW.bit = 0	
DBNZ	B, \$addr16	2	6	$B \leftarrow B - 1$, then $PC \leftarrow PC + 2 + \text{jdisp8}$ if $B \neq 0$	
	C, \$addr16	2	6	$C \leftarrow C - 1$, then $PC \leftarrow PC + 2 + \text{jdisp8}$ if $C \neq 0$	
	saddr, \$addr16	3	8	$(\text{saddr}) \leftarrow (\text{saddr}) - 1$, then $PC \leftarrow PC + 3 + \text{jdisp8}$ if $(\text{saddr}) \neq 0$	
NOP		1	2	No Operation	
EI		3	6	$IE \leftarrow 1$ (Enable Interrupt)	
DI		3	6	$IE \leftarrow 0$ (Disable Interrupt)	
HALT		1	2	Set HALT Mode	
STOP		1	2	Set STOP Mode	

备注 一个指令时钟周期是指一个 CPU 时钟周期(f_{CPU})，由处理器时钟控制寄存器(PCC)选择。

21.3 按寻址类型的指令列表

(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	!addr16	PSW	[DE]	[HL]	[HL + byte]	\$addr16	1	无
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV [#] XCH [#] ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV 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
!addr16		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	无
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 2 操作数 第 1 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

第二十二章 电气特性（标准产品，（A）等级产品）

最大额定值（ $T_A = 25^\circ\text{C}$ ）

参数	符号	条件	额度	单位
供电电压	V_{DD}		$-0.3 \sim +6.5$	V
	V_{SS}		$-0.3 \sim +0.3$	V
	AV_{REF}		$-0.3 \sim V_{DD} + 0.3^{\text{注}}$	V
	AV_{SS}		$-0.3 \sim +0.3$	V
输入电压	V_{I1}	P00 ~ P03, P30 ~ P34, P40 ~ P47, P120 ~ P123	$-0.3 \sim V_{DD} + 0.3^{\text{注}}$	V
	V_{I2}	P20 ~ P23	$-0.3 \sim AV_{REF} + 0.3^{\text{注}}$ 和 $-0.3 \sim V_{DD} + 0.3^{\text{注}}$	V
输出电压	V_O		$-0.3 \sim V_{DD} + 0.3^{\text{注}}$	V
模拟输入电压	V_{AN}		$-0.3 \sim AV_{REF} + 0.3^{\text{注}}$ 和 $-0.3 \sim V_{DD} + 0.3^{\text{注}}$	V
输出电流，高	I_{OH}	每个引脚	-10.0	mA
		除 P20 ~ P23 外的所有引脚总和	-44.0	mA
		P20 ~ P23 引脚总和	-44.0	mA
输出电流，低	I_{OL}	每个引脚	20.0	mA
		除 P20 ~ P23 外的所有引脚总和	44.0	mA
		P20 ~ P23 引脚总和	44.0	mA
工作环境温度	T_A	在正常操作模式下	$-40 \sim +85$	$^\circ\text{C}$
		在 Flash 存储器编程期间		
存储温度	T_{stg}	Flash 存储器空白状态	$-65 \sim +150$	$^\circ\text{C}$
		已经进行 Flash 存储器编程	$-40 \sim +125$	$^\circ\text{C}$

注 必须为 6.5 V 或更低。

注意事项 任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因此，必须保证产品在不超过最大额定值的条件下使用。

备注 除非另外指定，复用功能引脚的特性与端口引脚的特性相同。

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

振荡器	推荐电路	参数	条件	最小值	典型值	最大值	单位
陶瓷振荡器		振荡频率 (f_x) ^{注2}		2.0		10.0	MHz
晶体振荡器		振荡频率 (f_x) ^{注2}		2.0		10.0	MHz
外部时钟		X1 输入频率 (f_x) ^{注2}	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	2.0		10.0	MHz
			$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	2.0		5.0	
		X1 输入高/低电平宽度 (t_{XH} , t_{XL})	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0.045		0.25	μs
			$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	0.09		0.25	

注 1. 该产品的电压使用范围为 $2.2 \sim 5.5 \text{ V}$ ，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. 仅表示振荡器的特性。关于指令执行时间，请参见 **AC 特性**。

注意事项 在使用 X1 振荡器时，上图中虚线框部分的连线应遵从如下的连接方式，避免线间电容的负面影响。

- 连线要保证尽可能的短。
- 振荡器导线不要和其它的信号线互相交叉。
- 不要在流经大电流的信号线旁边进行振荡器的布线。
- 保持振荡器电容的地电位和 V_{SS} 相同。
- 不要将电容的接地点连接在大电流通过的地模式。
- 不要从振荡器取信号。

备注 对于振荡器的选择和振荡器的常数，用户需要自己评估振荡器或要求制造商评估测试。

标准产品，(A) 等级产品 $T_A = -40 \sim +85^{\circ}\text{C}$

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

振荡器	参数	条件	最小值	典型值	最大值	单位
高速内置振荡器	振荡频率（ $f_x = 8\text{ MHz}^{\#2}$ ）	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$T_A = -10 \sim +80^{\circ}\text{C}$		± 3	%
	偏差		$T_A = -40 \sim +85^{\circ}\text{C}$		± 5	%
	振荡频率（ f_x ） $^{\#2}$	$2.0\text{ V} \leq V_{DD} < 2.7\text{ V}$	5.5			MHz

注 1. 该产品的电压使用范围为 $2.2 \sim 5.5\text{ V}$ ，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. 仅表示振荡器的特性。关于指令执行时间，请参见 **AC 特性**。

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

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

注 该产品的电压使用范围为 $2.2 \sim 5.5\text{ V}$ ，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 ($T_A = -40 \sim +85^{\circ}\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}^{*1}$, $V_{SS} = 0 \text{ V}$) (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_{OH1}	除 P20~P23 外的引脚	每个引脚	$2.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			-5	mA
			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
	I_{OH2}	P20 ~ P23	每个引脚	$2.0 \text{ V} \leq AV_{REF} \leq 5.5 \text{ V}$			-5	mA
			总和	$2.0 \text{ V} \leq AV_{REF} \leq 5.5 \text{ V}$			-15	mA
	I_{OL1}	除 P20~P23 外的引脚	每个引脚	$2.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			10	mA
输出电流，低	I_{OL1}	除 P20~P23 外的引脚	总和	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			30	mA
				$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$			15	mA
	I_{OL2}	P20 ~ P23	每个引脚	$2.0 \text{ V} \leq AV_{REF} \leq 5.5 \text{ V}$			10	mA
			总和	$4.0 \text{ V} \leq AV_{REF} \leq 5.5 \text{ V}$			30	mA
				$2.0 \text{ V} \leq AV_{REF} \leq 4.0 \text{ V}$			15	mA
输入电压，高	V_{IH1}	P00 ~ P03, P30 ~ P34, P40 ~ P47, P120, P123			$0.8V_{DD}$		V_{DD}	V
	V_{IH2}	P20 ~ P23			$0.7AV_{REF}$		AV_{REF}	V
	V_{IH3}	P121, P122			$0.8V_{DD}$		V_{DD}	V
输入电压，低	V_{IL1}	P00 ~ P03, P30 ~ P34, P40 ~ P47, P120, P123			0		$0.2V_{DD}$	V
	V_{IL2}	P20 ~ P23			0		$0.3AV_{REF}$	V
	V_{IL3}	P121, P122			0		$0.2V_{DD}$	V
输出电压，高	V_{OH1}	除 P20 ~ P23 外所有引脚		$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$V_{DD} - 1.0$			V
		总和 $I_{OH1} = -15 \text{ mA}$		$I_{OH1} = -5 \text{ mA}$				
	V_{OH2}	$I_{OH1} = -100 \text{ A}$		$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	$V_{DD} - 0.5$			V
		P20 ~ P23 所有引脚总和		$4.0 \text{ V} \leq AV_{REF} \leq 5.5 \text{ V}$	$AV_{REF} - 1.0$			V
		$I_{OH2} = -10 \text{ mA}$		$I_{OH2} = -5 \text{ mA}$				
		$2.0 \text{ V} \leq AV_{REF} < 4.0 \text{ V}$		$I_{OH2} = -100 \mu\text{A}$	$AV_{REF} - 0.5$			V
输出电压，低	V_{OL1}	除 P20~P23 外所有引脚		$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			1.3	V
		总和 $I_{OL1} = 30 \text{ mA}$		$I_{OL1} = 10 \text{ mA}$				
	V_{OL2}	$I_{OL1} = 400 \mu\text{A}$		$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$			0.4	V
		P20 ~ P23 所有引脚总和		$4.0 \text{ V} \leq AV_{REF} \leq 5.5 \text{ V}$			1.3	V
		$I_{OL2} = 30 \text{ mA}$		$I_{OL2} = 10 \text{ mA}$				
		$I_{OL2} = 400 \mu\text{A}$		$2.0 \text{ V} \leq AV_{REF} \leq 4.0 \text{ V}$			0.4	V
输入漏电流，高	I_{LIH}	$V_i = V_{DD}$		除 X1 外的引脚			1	μA
输入漏电流，低	I_{LIL}	$V_i = 0 \text{ V}$		除 X1 外的引脚			-1	μA
输出漏电流，高	I_{LOH}	$V_o = V_{DD}$		除 X2 外的引脚			1	μA
输出漏电流，低	I_{LOL}	$V_o = 0 \text{ V}$		除 X2 外的引脚			-1	μA
上拉电阻	R_{PU}	$V_i = 0 \text{ V}$			10	30	100	$\text{k}\Omega$
下拉电阻	R_{PD}	P121, P122, 复位状态			10	30	100	$\text{k}\Omega$

注 该产品的电压使用范围为 $2.2 \sim 5.5 \text{ V}$ ，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 除非另外指定，复用功能引脚的特性与端口引脚的特性相同。

DC 特性 ($T_A = -40 \sim +85^{\circ}\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}$ ^{注1}, $V_{SS} = 0 \text{ V}$) (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}	晶体/陶瓷振荡，外部时钟输入振荡，操作模式 ^{注6}	f _X = 10 MHz V _{DD} = 5.0 V ±10% ^{注4}	A/D 转换器停止操作		6.1	12.2	mA
				A/D 转换器正在操作中 ^{注8}		7.6	15.2	
			f _X = 6 MHz V _{DD} = 5.0 V ±10% ^{注4}	A/D 转换器停止操作		5.5	11.0	mA
				A/D 转换器正在操作中 ^{注8}			14.0	
			f _X = 5 MHz V _{DD} = 3.0 V ±10% ^{注5}	A/D 转换器停止操作		3.0	6.0	mA
				A/D 转换器正在操作中 ^{注8}		4.5	9.0	
	I _{DD2}	晶体/陶瓷振荡，外部时钟输入 HALT 模式 ^{注6}	f _X = 10 MHz V _{DD} = 5.0 V ±10% ^{注4}	外设操作停止		1.7	3.8	mA
				外设正在操作中			6.7	
			f _X = 6 MHz V _{DD} = 5.0 V ±10% ^{注4}	外设操作停止		1.3	3.0	mA
				外设正在操作中			6.0	
			f _X = 5 MHz V _{DD} = 3.0 V ±10% ^{注5}	外设操作停止		0.48	1	mA
				外设正在操作中			2.1	
	I _{DD3} ^{注3}	高速内部振荡器操作模式 ^{注7}	f _X = 8 MHz V _{DD} = 5.0 V ±10% ^{注4}	A/D 转换器停止操作		5.0	10.0	mA
				A/D 转换器正在操作中 ^{注8}		6.5	13.0	
	I _{DD4}	高速内部振荡器 HALT 模式 ^{注7}	f _X = 8 MHz V _{DD} = 5.0 V ±10% ^{注4}	外设操作停止		1.4	3.2	mA
				外设正在操作中			5.9	
	I _{DD5}	STOP 模式	V _{DD} = 5.0 V ±10%	低速内部振荡器停止		3.5	20.0	μA
				低速内部振荡器工作		17.5	32.0	
			V _{DD} = 3.0 V ±10%	低速内部振荡器停止		3.5	15.5	μA
				低速内部振荡器工作		11.0	26.0	

- 注
1. 该产品的电压使用范围为 $2.2 \sim 5.5 \text{ V}$ ，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. 流经内部电源（ V_{DD} ）的总电流。包括外设操作电流（但是，不包括流经端口上拉电阻的电流）。
 3. 包括外围模块操作电流。
 4. 当处理器时钟控制寄存器（PCC）被清为 00H 时。
 5. 当处理器时钟控制寄存器（PCC）被置为 02H 时。
 6. 使用选项字节选择晶体/陶瓷振荡器，外部时钟输入作为系统时钟源。
 7. 使用选项字节选择高速内部振荡器时钟作为系统时钟源。
 8. 该电流包括流经 AV_{REF} 引脚的电流。

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

参数	符号	条件	最小值	典型值	最大值	单位
指令周期（指令最短执行时间）	T_{CY}	晶体/陶瓷振荡时钟， 外部时钟输入	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0.2	16	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
		高速内部振荡器时钟	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	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
TI000 输入的高电平宽度， 低电平宽度	t_{TIH}	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$2/f_{sam} + 0.1^{\#2}$			μs
	t_{TIL}	$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	$2/f_{sam} + 0.2^{\#2}$			μs
中断输入的高电平宽度， 低电平宽度	t_{INTH}		1			μs
	t_{INTL}					μs
RESET 输入低电平宽度	t_{RSL}		2			μs

- 注 1. 该产品的电压使用范围为 $2.2 \sim 5.5 \text{ V}$ ，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. 根据预分频模式寄存器 00（PRM00）的第 0 位和第 1 位（PRM000，PRM001），可以选择 $f_{sam} = f_{XP}$ ， $f_{XP}/4$ ，或 $f_{XP}/256$ 。注意当选择 TI000 有效沿作为计数时钟时， $f_{sam} = f_{XP}$ 。

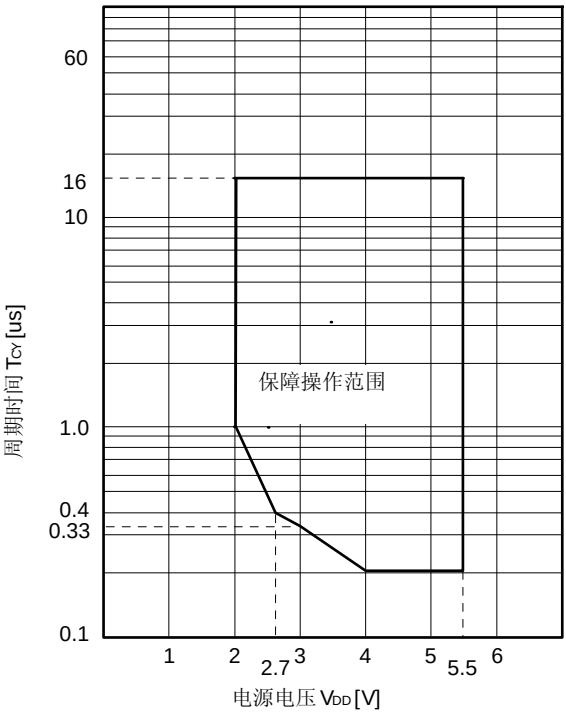
<R>

CPU 时钟频率，外围设备时钟频率

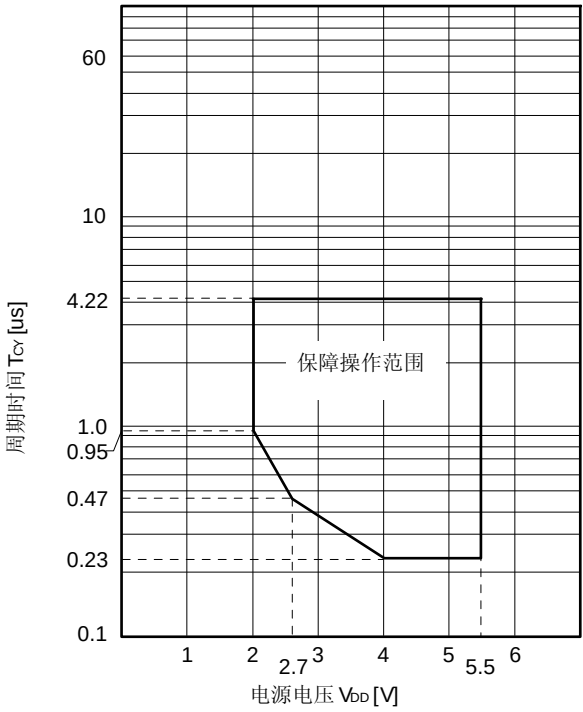
参数	条件	CPU 时钟（ f_{CPU} ）	外围时钟（ f_{XP} ）
陶瓷振荡器， 晶体振荡器， 外部时钟	$4.0 \text{ V} \sim 5.5 \text{ V}$	$125 \text{ kHz} \leq f_{CPU} \leq 10 \text{ MHz}$	$500 \text{ kHz} \leq f_{XP} \leq 10 \text{ MHz}$
	$3.0 \text{ V} \sim 4.0 \text{ V}$	$125 \text{ kHz} \leq f_{CPU} \leq 6 \text{ MHz}$	
	$2.7 \text{ V} \sim 3.0 \text{ V}$	$125 \text{ kHz} \leq f_{CPU} \leq 5 \text{ MHz}$	
	$2.0 \text{ V} \sim 2.7 \text{ V}^{\#}$	$125 \text{ kHz} \leq f_{CPU} \leq 2 \text{ MHz}$	$500 \text{ kHz} \leq f_{XP} \leq 5 \text{ MHz}$
高速内部振荡器	$4.0 \text{ V} \sim 5.5 \text{ V}$	500 kHz （典型值） $\leq f_{CPU} \leq 8 \text{ MHz}$ （典型值）	2 MHz （典型值） $\leq f_{XP} \leq 8 \text{ MHz}$ （典型值）
	$2.7 \text{ V} \sim 4.0 \text{ V}$	500 kHz （典型值） $\leq f_{CPU} \leq 4 \text{ MHz}$ （典型值）	
	$2.0 \text{ V} \sim 2.7 \text{ V}^{\#}$	500 kHz （典型值） $\leq f_{CPU} \leq 2 \text{ MHz}$ （典型值）	2 MHz （典型值） $\leq f_{XP} \leq 4 \text{ MHz}$ （典型值）

- 注 该产品的电压使用范围为 $2.2 \sim 5.5 \text{ V}$ ，因为上电清零电路（POC）的检测电压（ V_{POC} ）是 $2.1 \text{ V} \pm 0.1 \text{ V}$ 。

T_{CY} VS. V_{DD} （晶体/陶瓷振荡时钟，外部时钟输入）



T_{CY} VS. V_{DD} （高速内部振荡时钟）



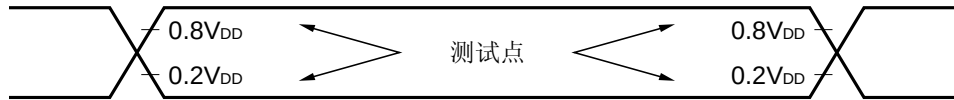
(2) 串行接口 ($T_A = -40 \sim +85^{\circ}\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}^{\#1}$, $V_{SS} = 0 \text{ V}$)

UART 模式（UART6，专用波特率发生器输出）

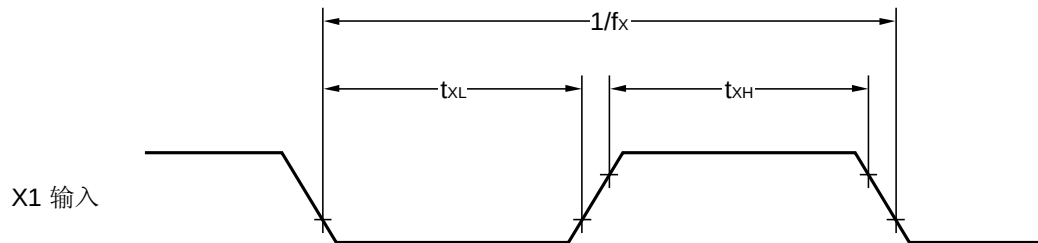
参数	符号	条件	最小值	典型值	最大值	单位
传输速率					312.5	kbps

注 该产品的电压使用范围为 $2.2 \sim 5.5 \text{ V}$ ，因为上电清零电路（POC）的检测电压（ V_{POC} ）是 $2.1 \text{ V} \pm 0.1 \text{ V}$ 。

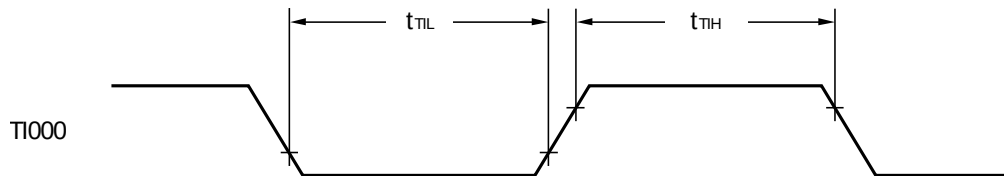
AC 时序测试点（不包括 X1 输入）



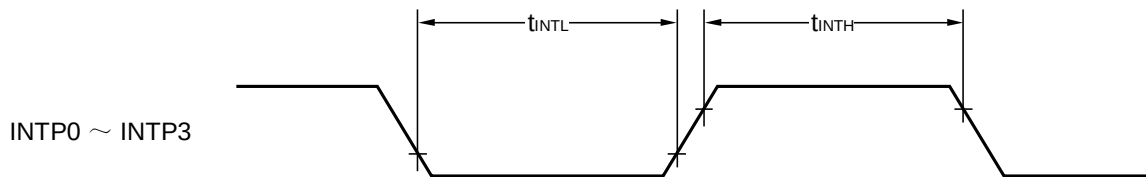
时钟时序



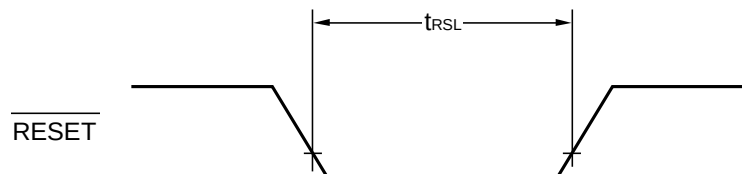
TI000 时序



中断输入时序



RESET 输入时序



A/D 转换器特性 ($T_A = -40 \sim +85^{\circ}\text{C}$, $2.7\text{ V} \leq AV_{\text{REF}} \leq V_{\text{DD}} \leq 5.5\text{ V}$, $V_{\text{SS}} = AV_{\text{SS}} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
分辨率			10	10	10	bit
总误差 ^{注 1, 2}	AINL	$4.0\text{ V} \leq AV_{\text{REF}} \leq 4.5\text{ V}$		± 0.2	± 0.4	%FSR
		$2.7\text{ V} \leq AV_{\text{REF}} < 4.0\text{ V}$		± 0.3	± 0.6	%FSR
转换时间	t_{CONV}	$4.5\text{ V} \leq AV_{\text{REF}} \leq 5.5\text{ V}$	3.0		100	μs
		$4.0\text{ V} \leq AV_{\text{REF}} < 4.5\text{ V}$	4.8		100	μs
		$2.85\text{ V} \leq AV_{\text{REF}} < 4.0\text{ V}$	6.0		100	μs
		$2.7\text{ V} \leq AV_{\text{REF}} < 2.85\text{ V}$	14.0		100	μs
零度误差 ^{注 1, 2}	EzS	$4.0\text{ V} \leq AV_{\text{REF}} \leq 5.5\text{ V}$			± 0.4	%FSR
		$2.7\text{ V} \leq AV_{\text{REF}} < 4.0\text{ V}$			± 0.6	%FSR
满量程误差 ^{注 1, 2}	Efs	$4.0\text{ V} \leq AV_{\text{REF}} \leq 5.5\text{ V}$			± 0.4	%FSR
		$2.7\text{ V} \leq AV_{\text{REF}} < 4.0\text{ V}$			± 0.6	%FSR
积分非线性误差 ^{注 1}	ILE	$4.0\text{ V} \leq AV_{\text{REF}} \leq 5.5\text{ V}$			± 2.5	LSB
		$2.7\text{ V} \leq AV_{\text{REF}} < 4.0\text{ V}$			± 4.5	LSB
微分非线性误差 ^{注 1}	DLE	$4.0\text{ V} \leq AV_{\text{REF}} \leq 5.5\text{ V}$			± 1.5	LSB
		$2.7\text{ V} \leq AV_{\text{REF}} < 4.0\text{ V}$			± 2.0	LSB
模拟输入电压	V_{AIN}		V_{SS} ^{注 1}		AV_{REF}	V

- 注 1. 不包括量化误差 ($\pm 1/2$ LSB)。
2. 该值表示满量程的百分比 (%FSR)。

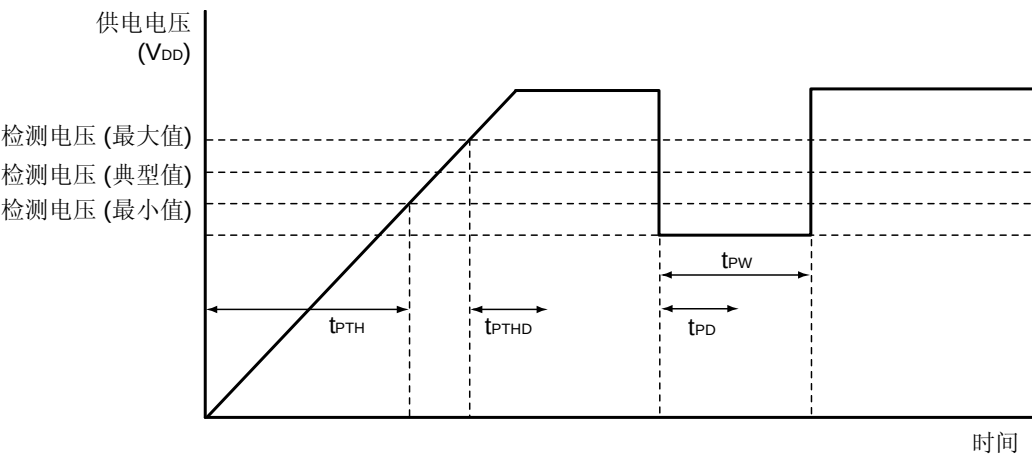
注意事项 当模拟输入引脚用作复用 I/O 端口，或在 A/D 转换期间端口发生改变时，转换精度可能降低。

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

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V_{POC}		2.0	2.1	2.2	V
电源启动时间	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}	供电电压上升时，达到检测电压（MAX.）后			3.0	ms
响应延迟时间 2 ^{注 2}	t_{PD}	供电电压下降			1.0	ms
最短脉冲宽度	t_{PW}		0.2			ms

- 注 1. 从电压检测到内部复位释放所需要的时间。
 2. 从电压检测到内部复位信号产生所需要的时间。

POC 电路时序



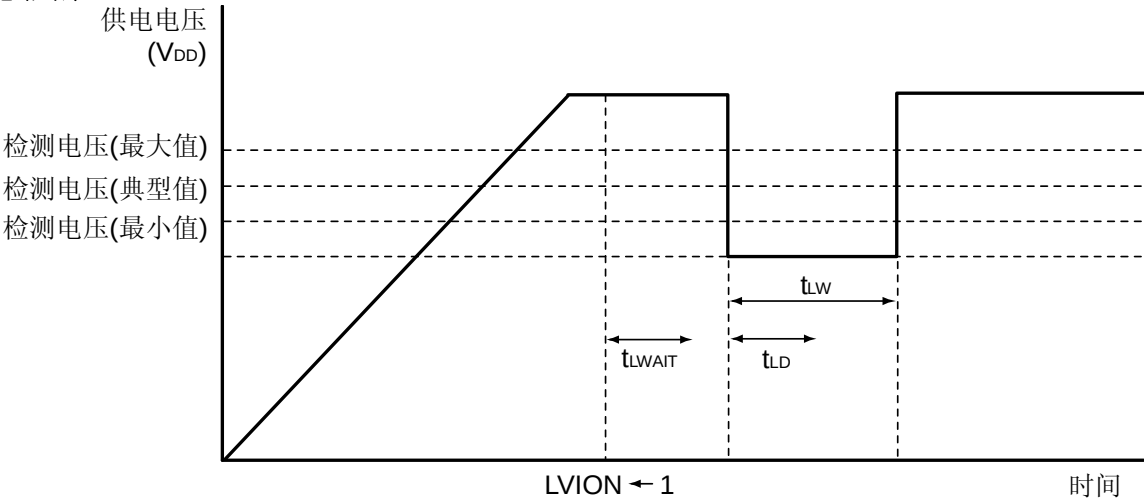
LVI 电路特性（ $T_A = -40 \sim +85^{\circ}\text{C}$ ）

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V_{LV10}		4.1	4.3	4.5	V
	V_{LV11}		3.9	4.1	4.3	V
	V_{LV12}		3.7	3.9	4.1	V
	V_{LV13}		3.5	3.7	3.9	V
	V_{LV14}		3.3	3.5	3.7	V
	V_{LV15}		3.15	3.3	3.45	V
	V_{LV16}		2.95	3.1	3.25	V
	V_{LV17}		2.7	2.85	3.0	V
	V_{LV18}		2.5	2.6	2.7	V
	V_{LV19}		2.25	2.35	2.45	V
响应时间 ^{注 1}	t_{LD}			0.2	2.0	ms
最短脉冲宽度	t_{LW}		0.2			ms
操作稳定等待时间 ^{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产生，所需的时间。
2. 从设置 LVION 为 1 到操作稳定，所需的时间。

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

LVI 电路时序



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

参数	符号	条件	最小值	典型值	最大值	单位
数据保持电源电压	V_{DDDR}		2.0		5.5	V
释放信号放置时间	t_{SREL}		0			μs

Flash 存储器编程特性 ($T_A = -40 \sim +85^{\circ}\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
电源电流	I_{DD}	$V_{DD} = 5.5\text{ V}$			7.0	mA
擦除次数 ^{‡1} (每个 Block)	N_{ERASE}	$T_A = -40 \sim +85^{\circ}\text{C}$	1000			次
芯片擦除时间	T_{CERASE}	$T_A = -10 \sim +85^{\circ}\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0.8	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		1.0	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		1.2	s
		$T_A = -10 \sim +85^{\circ}\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		4.8	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		5.2	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		6.1	s
		$T_A = -40 \sim +85^{\circ}\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		1.6	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		1.8	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		2.0	s
		$T_A = -40 \sim +85^{\circ}\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		9.1	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		10.1	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		12.3	s
Block 擦除时间	T_{BERASE}	$T_A = -10 \sim +85^{\circ}\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0.4	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		0.5	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		0.6	s
		$T_A = -10 \sim +85^{\circ}\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		2.6	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		2.8	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		2.3	s
		$T_A = -40 \sim +85^{\circ}\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0.9	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		1.0	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		1.1	s
		$T_A = -40 \sim +85^{\circ}\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		4.9	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		5.4	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		6.6	s
写字节时间	T_{WRITE}	$T_A = -40 \sim +85^{\circ}\text{C}$, $N_{ERASE} \leq 1000$			150	μs
内部校验	T_{VERIFY}	每个 block			6.8	ms
		每个字节			27	μs
空白检测	T_{BLKCHK}	每个 block			480	μs
保存年限		$T_A = 85^{\circ}\text{C}$ ^{‡2} , $N_{ERASE} \leq 1000$	10			年

- 注
1. 根据擦除计数 (N_{ERASE})，擦除时间不同。参考芯片擦除时间和 block 擦除时间参数。
 2. 操作和不操作的平均温度是 85°C 时。

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

第二十三章 电气特性（目标值）（（A2）等级产品）

注意事项 这些特性显示的都是目标值，在设备评测后可能会改变。操作电压范围同样也可能改变。

最大额定值（ $T_A = 25^\circ\text{C}$ ）

参数	符号	条件	额度	单位
供电电压	V_{DD}		$-0.3 \sim +6.5$	V
	V_{SS}		$-0.3 \sim +0.3$	V
	AV_{REF}		$-0.3 \sim V_{DD} + 0.3$ ^{注1}	V
	AV_{SS}		$-0.3 \sim +0.3$	V
输入电压	V_{I1}	P00 ~ P03, P30 ~ P34, P40 ~ P47, P120 ~ P123	$-0.3 \sim V_{DD} + 0.3$ ^{注1}	V
	V_{I2}	P20 ~ P23	$-0.3 \sim AV_{REF} + 0.3$ ^{注1} 和 $-0.3 \sim V_{DD} + 0.3$ ^{注1}	V
输出电压	V_O		$-0.3 \sim V_{DD} + 0.3$ ^{注1}	V
模拟输入电压	V_{AN}		$-0.3 \sim AV_{REF} + 0.3$ ^{注1} 和 $-0.3 \sim V_{DD} + 0.3$ ^{注1}	V
输出电流，高	I_{OH}	每个引脚	-7.0	mA
		除 P20 ~ P23 外的所有引脚总和	-30.0	mA
		P20 ~ P23 引脚总和	-30.0	mA
输出电流，低	I_{OL}	每个引脚	14.0	mA
		除 P20 ~ P23 外的所有引脚总和	30.0	mA
		P20 ~ P23 引脚总和	30.0	mA
工作环境温度	P_T ^{注2}	$T_A = -40 \sim +85^\circ\text{C}$	120	mW
		$T_A = +85 \sim +125^\circ\text{C}$	110	mW
存储温度	T_A	在正常操作模式下	$-40 \sim +125$	$^\circ\text{C}$
		在 Flash 存储器编程期间	$-40 \sim +105$	$^\circ\text{C}$
供电电压	T_{stg}	Flash 存储器空白状态	$-65 \sim +150$	$^\circ\text{C}$
		已经进行 Flash 存储器编程	$-40 \sim +125$	$^\circ\text{C}$

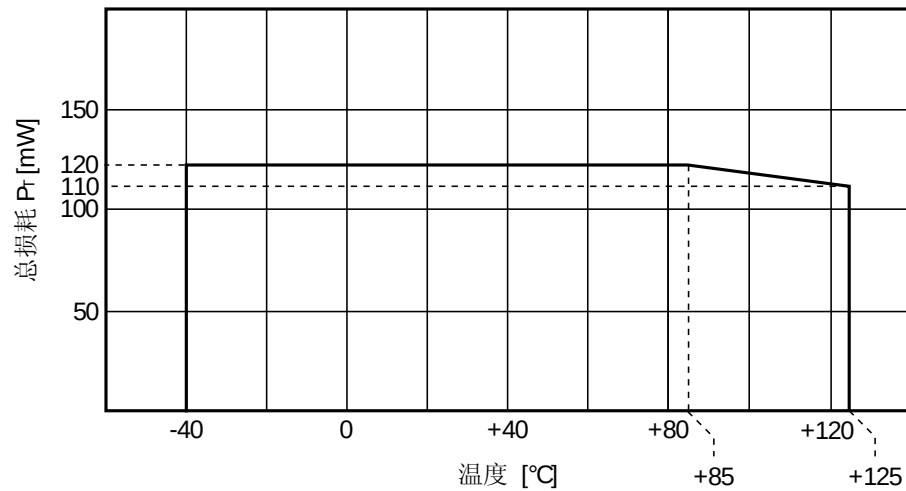
注 1 必须为 6.5 V 或更低。

注意事项 任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因此，必须保证产品在不超过最大额定值的条件下使用。

备注 除非另外指定，复用功能引脚的特性与端口引脚的特性相同。

（注2 在下页列出。）

注 2. 根据允许的总损耗而变化（参见下图）。



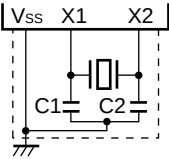
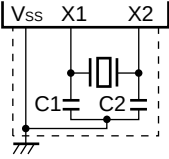
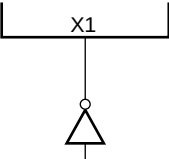
使用如下的公式设计器件的功耗总和，要小于或等于器件总损耗 P_T （推荐使用小于等于 80% 的额定值）。

- 总功耗 = $V_{DD} \times \{I_{DD} - \Sigma I_{OH}\} + \Sigma \{ (V_{DD} - V_{OH}) \times I_{OH} \} + \Sigma (V_{OH} \times I_{OL})$

当使用内部上拉电阻时，使用如下公式计算它的功耗，并且将结果加到上面的结果中。

- 内部上拉电阻的功耗 = $\Sigma (V_{DD}/R_{PU} \times V_{DD})$

<R> X1 振荡器特性 ($T_A = -40 \sim +125^{\circ}\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}$ ^{註1}, $V_{SS} = 0 \text{ V}$)

振荡器	推荐电路	参数	条件	最小值	典型值	最大值	单位
陶瓷振荡器		振荡频率 (f_x) ^{註2}		2.0		8.0	MHz
晶体振荡器		振荡频率 (f_x) ^{註2}		2.0		8.0	MHz
外部时钟		X1 输入频率 (f_x) ^{註2}	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	2.0		8.0	MHz
			$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	2.0		5.0	
		X1 输入高/低电平宽度 (t_{xH} , t_{xL})	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0.057		0.25	μs
			$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	0.09		0.25	

注 1. 该产品的电压使用范围为 2.26 ~ 5.5 V，因为上电清零电路（POC）的检测电压（ V_{POC} ）是 2.26 V（最大值）。

2. 仅表示振荡器的特性。关于指令执行时间，请参见 **AC 特性**。

注意事项 在使用 X1 振荡器时，上图中虚线框部分的连线应遵从如下的连接方式，避免线间电容的负面影响。

- 连线要保证尽可能的短。
- 振荡器导线不要和其它的信号线互相交叉。
- 不要在流经大电流的信号线旁边进行振荡器的布线。
- 保持振荡器电容的地电位和 V_{SS} 相同。
- 不要将电容的接地点连接在大电流通过的地模式。
- 不要从振荡器取信号。

备注 对于振荡器的选择和振荡器的常数,用户需要自己评估振荡器或要求制造商评估测试。

(A2) 等级产品 $T_A = -40 \sim +125^{\circ}\text{C}$

高速内部振荡器特性（ $T_A = -40 \sim +125^{\circ}\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}^{\#1}$, $V_{SS} = 0 \text{ V}$ ）

振荡器	参数	条件	最小值	典型值	最大值	单位
高速内置振荡器	振荡频率（ $f_X = 8 \text{ MHz}^{\#2}$ ） 偏差	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$ $T_A = -10 \sim +80^{\circ}\text{C}$ $T_A = -40 \sim +125^{\circ}\text{C}$			± 3	%
					± 5	%
	振荡频率（ f_X ） $\#2$	$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	5.5			MHz

注 1. 该产品的电压使用范围是 $2.26 \sim 5.5 \text{ V}$ ，因为上电清零（POC）电路的检测电压（ V_{POC} ）是 2.26 V （最大值）。

2. 仅表示振荡器的特性。关于指令执行时间，请参见 **AC 特性**。

低速内部振荡器特性（ $T_A = -40 \sim +125^{\circ}\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}^{\#}$, $V_{SS} = 0 \text{ V}$ ）

振荡器	参数	条件	最小值	典型值	最大值	单位
低速内部振荡器	振荡频率（ f_{RL} ）		120	240	495	kHz

注 该产品的电压使用范围是 $2.26 \sim 5.5 \text{ V}$ ，因为上电清零（POC）电路的检测电压（ V_{POC} ）是 2.26 V （最大值）。

DC 特性 ($T_A = -40 \sim +125^{\circ}\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}$, $V_{SS} = 0 \text{ V}$) (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 _{OH1}	除 P20~P23 之外的引脚	每个引脚	$2.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$		-3.5	mA
			总和	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$		-17.5	mA
				$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$		-10.5	mA
	I _{OH2}	P20 ~ P23	每个引脚	$2.0 \text{ V} \leq AV_{REF} \leq 5.5 \text{ V}$		-3.5	mA
			总和	$2.0 \text{ V} \leq AV_{REF} \leq 5.5 \text{ V}$		-10.5	mA
输出电流, 低	I _{OL1}	除 P20~P23 外的引脚	每个引脚	$2.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$		7.0	mA
			总和	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$		21.0	mA
				$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$		10.5	mA
	I _{OL2}	P20 ~ P23	每个引脚	$2.0 \text{ V} \leq AV_{REF} \leq 5.5 \text{ V}$		7.0	mA
			总和	$4.0 \text{ V} \leq AV_{REF} \leq 5.5 \text{ V}$		21.0	mA
				$2.0 \text{ V} \leq AV_{REF} \leq 4.0 \text{ V}$		10.5	mA
输入电压, 高	V _{IH1}	P00 ~ P03, P30 ~ P34, P40 ~ P47, P120, P123		$0.8V_{DD}$		V_{DD}	V
	V _{IH2}	P20 ~ P23		$0.7AV_{REF}$		AV_{REF}	V
	V _{IH3}	P121, P122		$0.8V_{DD}$		V_{DD}	V
输入电压, 低	V _{IL1}	P00 ~ P03, P30 ~ P34, P40 ~ P47, P120, P123		0		$0.2V_{DD}$	V
	V _{IL2}	P20 ~ P23		0		$0.3AV_{REF}$	V
	V _{IL3}	P121, P122		0		$0.2V_{DD}$	V
输出电压, 高	V _{OH1}	除 P20 ~ P23 外所有引脚的总和	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$ I _{OH1} = -10.5 mA	$V_{DD} - 1.0$			V
		I _{OH1} = -100 A	$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$ I _{OH1} = -3.5 mA	$V_{DD} - 0.5$			V
	V _{OH2}	P20 ~ P23 所有引脚总和	$4.0 \text{ V} \leq AV_{REF} \leq 5.5 \text{ V}$ I _{OH2} = -7 mA	$AV_{REF} - 1.0$			V
		I _{OH2} = -100 μA	$2.0 \text{ V} \leq AV_{REF} < 4.0 \text{ V}$ I _{OH2} = -3.5 mA	$AV_{REF} - 0.5$			V
输出电压, 低	V _{OL1}	除 P20 ~ P23 外所有引脚的总和	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$ I _{OL1} = 21 mA			1.3	V
		I _{OL1} = 400 μA	$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$ I _{OL1} = 7 mA			0.4	V
	V _{OL2}	P20 ~ P23 所有引脚总和	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$ I _{OL2} = 21 mA			1.3	V
		I _{OL2} = 400 μA	$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$ I _{OL2} = 7 mA			0.4	V
输入漏电流, 高	I _{LIH}	V _I = V _{DD}	除 X1 外的引脚			10	μA
输入漏电流, 低	I _{LIL}	V _I = 0 V	除 X1 外的引脚			-10	μA
输出漏电流, 高	I _{LOH}	V _O = V _{DD}	除 X2 外的引脚			10	μA
输出漏电流, 低	I _{LOL}	V _O = 0 V	除 X2 外的引脚			-10	μA
上拉电阻	R _{PU}	V _I = 0 V		10	30	120	kΩ
下拉电阻	R _{PD}	P121, P122, 复位状态		10	30	120	kΩ

注 该产品的电压使用范围是 2.26 ~ 5.5 V，因为上电清零（POC）电路的检测电压（V_{POC}）是 2.26 V（最大值）。

备注 除非另外指定, 复用功能引脚的特性与端口引脚的特性相同。

DC 特性 ($T_A = -40 \sim +125^\circ\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}$ ^{注2}, $V_{SS} = 0 \text{ V}$) (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}	晶体/陶瓷振荡，外部时钟输入振荡，操作模式 ^{注6}	f _x = 8 MHz V _{DD} = 5.0 V ±10% ^{注4}	A/D 转换器停止操作		5.8	12.8	mA		
				A/D 转换器正在操作中 ^{注8}		7.3	15.8			
			f _x = 6 MHz V _{DD} = 5.0 V ±10% ^{注4}	A/D 转换器停止操作		5.5	12.2	mA		
				A/D 转换器正在操作中 ^{注8}			15.2			
			f _x = 5 MHz V _{DD} = 3.0 V ±10% ^{注5}	A/D 转换器停止操作		3.0	6.6	mA		
				A/D 转换器正在操作中 ^{注8}		4.5	9.6			
			I _{DD2}	晶体/陶瓷振荡，外部时钟输入 HALT 模式 ^{注6}	f _x = 8 MHz V _{DD} = 5.0 V ±10% ^{注4}	外设操作停止		1.5	4.6	mA
						外设正在操作中			7.6	
	f _x = 6 MHz V _{DD} = 5.0 V ±10% ^{注4}	外设操作停止				1.3	4.2	mA		
		外设正在操作中					7.2			
	f _x = 5 MHz V _{DD} = 3.0 V ±10% ^{注5}	外设操作停止				0.48	1.6	mA		
		外设正在操作中					2.7			
	I _{DD3} ^{注3}	高速内部振荡操作模式 ^{注7}			f _x = 8 MHz V _{DD} = 5.0 V ±10% ^{注4}	A/D 转换器停止操作		5.0	12.2	mA
						A/D 转换器正在操作中 ^{注8}		6.5	15.2	
	I _{DD4}	高速内部振荡器 HALT 模式 ^{注7}	f _x = 8 MHz V _{DD} = 5.0 V ±10% ^{注4}	外设操作停止		1.4	4.4	mA		
				外设正在操作中			7.1			
	I _{DD5}	STOP 模式	V _{DD} = 5.0 V ±10%	低速内部振荡器停止		3.5	1200	μA		
				低速内部振荡器工作		17.5	1300			
			V _{DD} = 3.0 V ±10%	低速内部振荡器停止		3.5	600	μA		
				低速内部振荡器工作		11.0	700			

- 注
1. 该产品的电压使用范围是 $2.26 \sim 5.5 \text{ V}$ ，因为上电清零（POC）电路的检测电压（ V_{POC} ）是 2.26 V （最大值）。
 2. 流经内部电源（ V_{DD} ）的总电流。包括外设操作电流（但是，不包括流经端口上拉电阻的电流）。
 3. 包括外围模块操作电流。
 4. 当处理器时钟控制寄存器（PCC）被清为 00H 时。
 5. 当处理器时钟控制寄存器（PCC）被置为 02H 时。
 6. 使用选项字节选择晶体/陶瓷振荡器，外部时钟输入作为系统时钟源。
 7. 使用选项字节选择高速内部振荡时钟作为系统时钟源。
 8. 该电流包括流经 AV_{REF} 引脚的电流。

AC 特性

(1) 基本操作 ($T_A = -40 \sim +125^{\circ}\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}^{\#1}$, $V_{SS} = 0 \text{ V}$)

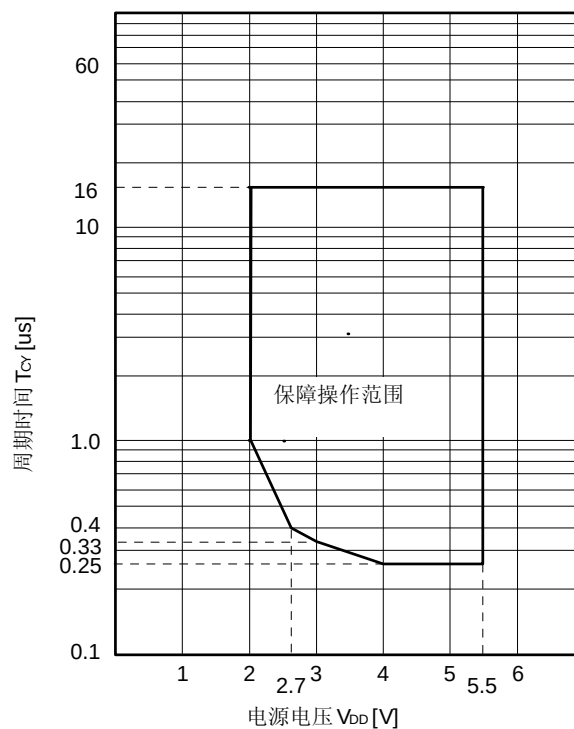
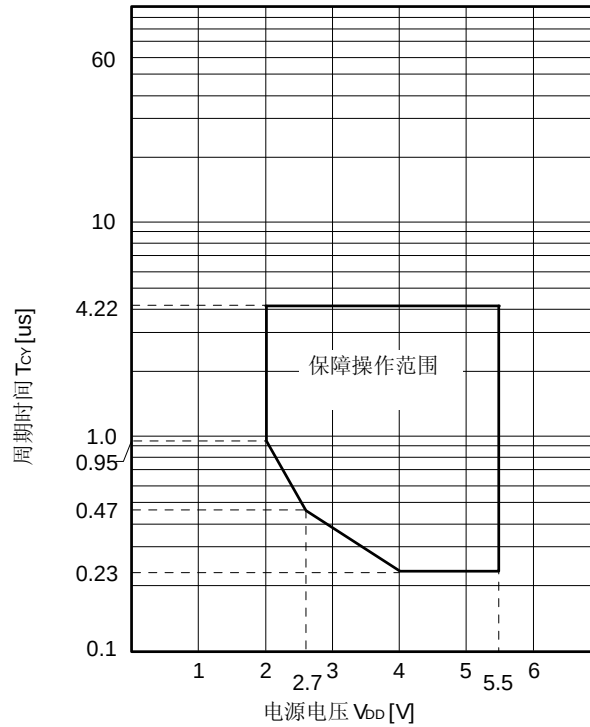
参数	符号	条件	最小值	典型值	最大值	单位
指令周期（指令最短执行时间）	T_{CY}	晶体/陶瓷振荡时钟， 外部时钟输入	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0.25	16	μs
			$3.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	0.33	16	μs
			$2.7 \text{ V} \leq V_{DD} < 3.0 \text{ V}$	0.4	16	μs
			$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	1	16	μs
		高速内部振荡器时钟	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	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
TI000 输入的高电平宽度， 低电平宽度	t_{TIH}	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$2/f_{sam} + 0.1^{\#2}$			μs
	t_{TIL}	$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	$2/f_{sam} + 0.2^{\#2}$			μs
中断输入的高电平宽度， 低电平宽度	t_{INTH}		1			μs
	t_{INTL}					μs
RESET 输入低电平宽度	t_{RSL}		2			μs

- 注 1. 该产品的电压使用范围是 $2.26 \sim 5.5 \text{ V}$ ，因为上电清零（POC）电路的检测电压（ V_{POC} ）是 2.26 V （最大值）。
2. 根据预分频模式寄存器 00（PRM00）的第 0 位和第 1 位（PRM000, PRM001），可以选择 $f_{sam} = f_{XP}$, $f_{XP}/4$, 或 $f_{XP}/256$ 。注意当选择 TI000 有效沿作为计数时钟时， $f_{sam} = f_{XP}$ 。

CPU 时钟频率, 外围设备时钟频率

参数	条件	CPU 时钟 (f_{CPU})	外围时钟 (f_{XP})
陶瓷振荡器, 晶体振荡器, 外部时钟	$4.0 \text{ V} \sim 5.5 \text{ V}$	$125 \text{ kHz} \leq f_{CPU} \leq 8 \text{ MHz}$	$500 \text{ kHz} \leq f_{XP} \leq 8 \text{ MHz}$
	$3.0 \text{ V} \sim 4.0 \text{ V}$	$125 \text{ kHz} \leq f_{CPU} \leq 6 \text{ MHz}$	
	$2.7 \text{ V} \sim 3.0 \text{ V}$	$125 \text{ kHz} \leq f_{CPU} \leq 5 \text{ MHz}$	
	$2.0 \text{ V} \sim 2.7 \text{ V}^{\#}$	$125 \text{ kHz} \leq f_{CPU} \leq 2 \text{ MHz}$	$500 \text{ kHz} \leq f_{XP} \leq 5 \text{ MHz}$
高速内部振荡器	$4.0 \text{ V} \sim 5.5 \text{ V}$	500 kHz （典型值） $\leq f_{CPU} \leq 8 \text{ MHz}$ （典型值）	2 MHz （典型值） $\leq f_{XP} \leq 8 \text{ MHz}$ （典型值）
	$2.7 \text{ V} \sim 4.0 \text{ V}$	500 kHz （典型值） $\leq f_{CPU} \leq 4 \text{ MHz}$ （典型值）	
	$2.0 \text{ V} \sim 2.7 \text{ V}^{\#}$	500 kHz （典型值） $\leq f_{CPU} \leq 2 \text{ MHz}$ （典型值）	2 MHz （典型值） $\leq f_{XP} \leq 4 \text{ MHz}$ （典型值）

- 注 该产品的电压使用范围是 $2.26 \sim 5.5 \text{ V}$ ，因为上电清零（POC）电路的检测电压（ V_{POC} ）是 2.26 V （最大值）。

T_{CY} VS. V_{DD}（晶体/陶瓷振荡时钟，外部时钟输入）**T_{CY} VS. V_{DD}**（高速内部振荡时钟）

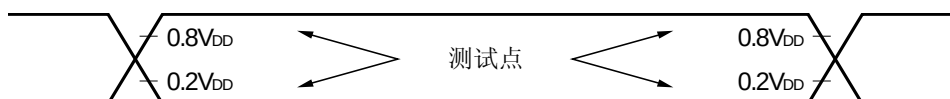
(2) 串行接口 ($T_A = -40 \sim +125^{\circ}\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}^{\#}$, $V_{SS} = 0 \text{ V}$)

UART 模式（UART6，专用波特率发生器输出）

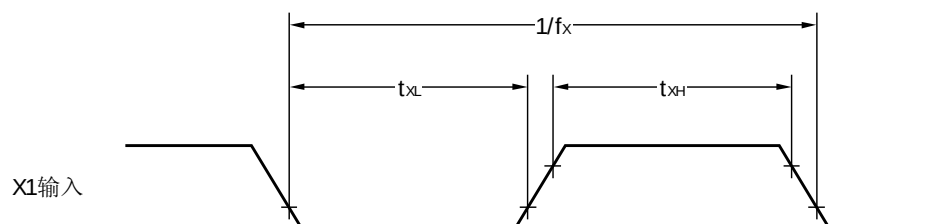
参数	符号	条件	最小值	典型值	最大值	单位
传输速率					312.5	kbps

注 该产品的电压使用范围是 $2.26 \sim 5.5 \text{ V}$ ，因为上电清零（POC）电路的检测电压（ V_{POC} ）是 2.26 V （最大值）。

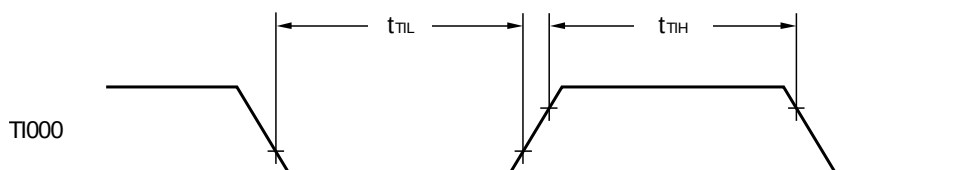
AC 时序测试点（不包括 X1 输入）



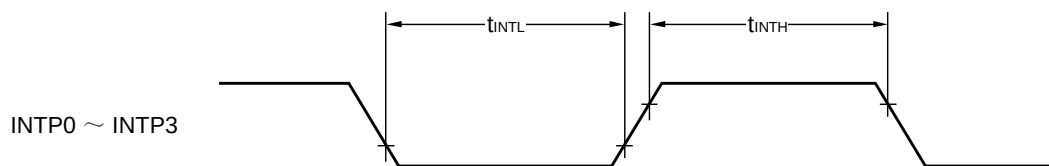
时钟时序



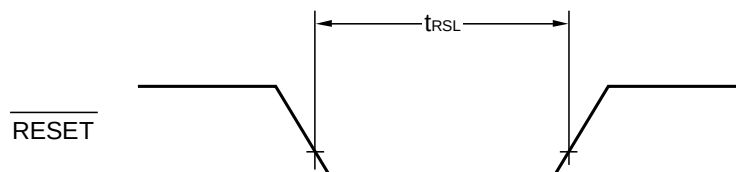
TI000 时序



中断输入时序



RESET 输入时序



A/D 转换器特性 ($T_A = -40 \sim +125^{\circ}\text{C}$, $2.7\text{ V} \leq AV_{\text{REF}} \leq V_{\text{DD}} \leq 5.5\text{ V}$, $V_{\text{SS}} = AV_{\text{SS}} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
分辨率			10	10	10	bit
总误差 ^{註1, 2}	AINL	$4.0\text{ V} \leq AV_{\text{REF}} \leq 4.5\text{ V}$		± 0.2	± 0.7	%FSR
		$2.7\text{ V} \leq AV_{\text{REF}} < 4.0\text{ V}$		± 0.3	± 0.9	%FSR
转换时间	t_{CONV}	$4.5\text{ V} \leq AV_{\text{REF}} \leq 5.5\text{ V}$	3.0		30	μs
		$4.0\text{ V} \leq AV_{\text{REF}} < 4.5\text{ V}$	4.8		30	μs
		$2.85\text{ V} \leq AV_{\text{REF}} < 4.0\text{ V}$	6.0		30	μs
		$2.7\text{ V} \leq AV_{\text{REF}} < 2.85\text{ V}$	14.0		30	μs
零度误差 ^{註1, 2}	Ezs	$4.0\text{ V} \leq AV_{\text{REF}} \leq 5.5\text{ V}$			± 0.7	%FSR
		$2.7\text{ V} \leq AV_{\text{REF}} < 4.0\text{ V}$			± 0.9	%FSR
满量程误差 ^{註1, 2}	Efs	$4.0\text{ V} \leq AV_{\text{REF}} \leq 5.5\text{ V}$			± 0.7	%FSR
		$2.7\text{ V} \leq AV_{\text{REF}} < 4.0\text{ V}$			± 0.9	%FSR
积分非线性误差 ^{註1}	ILE	$4.0\text{ V} \leq AV_{\text{REF}} \leq 5.5\text{ V}$			± 5.5	LSB
		$2.7\text{ V} \leq AV_{\text{REF}} < 4.0\text{ V}$			± 7.5	LSB
微分非线性误差 ^{註1}	DLE	$4.0\text{ V} \leq AV_{\text{REF}} \leq 5.5\text{ V}$			± 2.5	LSB
		$2.7\text{ V} \leq AV_{\text{REF}} < 4.0\text{ V}$			± 3.0	LSB
模拟输入电压	V_{AIN}		V_{SS} ^{註1}		AV_{REF}	V

注 1. 不包括量化误差 ($\pm 1/2$ LSB)。

2. 该值表示满量程的百分比 (%FSR)。

注意事项 当模拟输入引脚用作复用 I/O 端口，或在 A/D 转换期间端口发生改变时，转换精度可能降低。

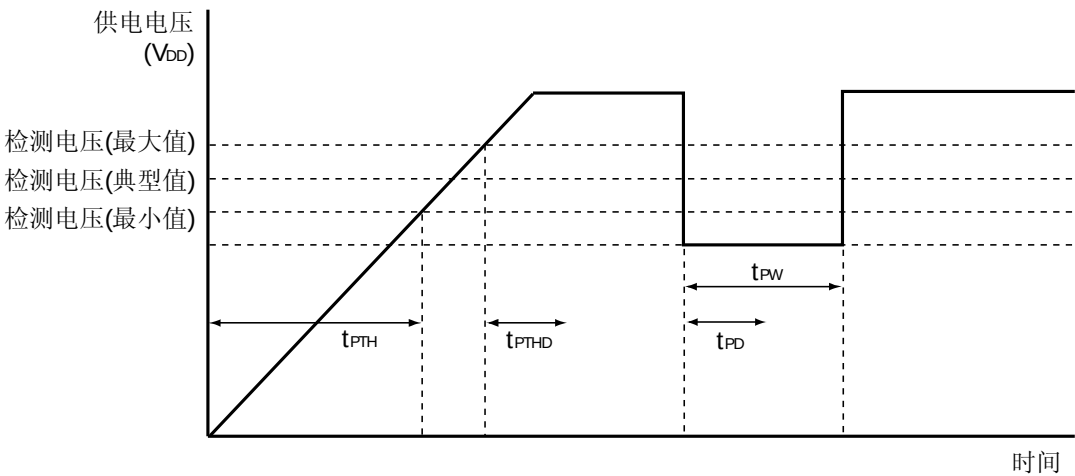
(A2) 等级产品 $T_A = -40 \sim +125^{\circ}\text{C}$

POC 电路特性（ $T_A = -40 \sim +125^{\circ}\text{C}$ ）

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	2.26	V
电源启动时间	t_{PTH}	$V_{\text{DD}}: 0\text{ V} \rightarrow 2.1\text{ V}$	1.5			μs
响应延迟时间 1 ^{※1}	t_{PTHD}	供电电压上升时,达到检测电压 (MAX.) 后			3.0	ms
响应延迟时间 2 ^{※2}	t_{PD}	供电电压下降			1.0	ms
最短脉冲宽度	t_{PW}		0.2			ms

- 注
1. 从电压检测到内部复位释放所需要的时间。
 2. 从电压检测到内部复位信号产生所需要的时间。

POC 电路时序



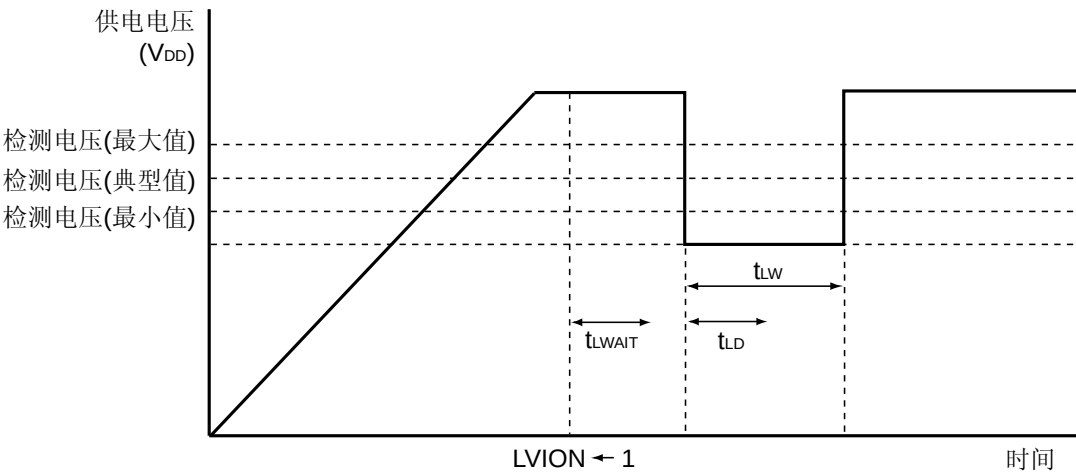
LVI 电路特性（ $T_A = -40 \sim +125^{\circ}\text{C}$ ）

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	4.65	V
	V_{LV11}		3.9	4.1	4.45	V
	V_{LV12}		3.7	3.9	4.25	V
	V_{LV13}		3.5	3.7	4.05	V
	V_{LV14}		3.3	3.5	3.85	V
	V_{LV15}		3.15	3.3	3.60	V
	V_{LV16}		2.95	3.1	3.40	V
	V_{LV17}		2.70	2.85	3.15	V
	V_{LV18}		2.50	2.60	2.85	V
	V_{LV19}		2.25	2.35	2.60	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
操作稳定等待时间 ^{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产生，所需的时间。
 2. 从设置 LVION 为 1 到操作稳定，所需的时间。

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

LVI 电路时序



数据存储器 STOP 模式低电源电压数据保持特性（ $T_A = -40 \sim +125^{\circ}\text{C}$ ）

参数	符号	条件	最小值	典型值	最大值	单位
数据保持电源电压	V_{DDDR}		2.0		5.5	V
释放信号放置时间	t_{SREL}		0			μs

Flash 存储器编程特性 ($T_A = -40 \sim +105^{\circ}\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$)

参数	符号	条件		最小值	典型值	最大值	单位
电源电流	I_{DD}	$V_{DD} = 5.5\text{ V}$				7.0	mA
擦除次数 ^{註1} (每个 Block)	N_{ERASE}	$T_A = -40 \sim +105^{\circ}\text{C}$		1000			次
芯片擦除时间	T_{CERASE}	$T_A = -10 \sim +105^{\circ}\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			0.8	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$			1.0	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$			1.2	s
		$T_A = -10 \sim +105^{\circ}\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			4.8	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$			5.2	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$			6.1	s
		$T_A = -40 \sim +105^{\circ}\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			1.6	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$			1.8	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$			2.0	s
		$T_A = -40 \sim +105^{\circ}\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			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
Block 擦除时间	T_{BERASE}	$T_A = -10 \sim +105^{\circ}\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			0.4	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$			0.5	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$			0.6	s
		$T_A = -10 \sim +105^{\circ}\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			2.6	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$			2.8	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$			3.3	s
		$T_A = -40 \sim +105^{\circ}\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			0.9	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$			1.0	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$			1.1	s
		$T_A = -40 \sim +105^{\circ}\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			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
写字节时间	T_{WRITE}	$T_A = -40 \sim +105^{\circ}\text{C}$, $N_{ERASE} \leq 1000$				150	μs
内部校验	T_{VERIFY}	每个 block				6.8	ms
		每个字节				27	μs
空白检测	T_{BLKCHK}	每个 block				480	μs
总功耗	P_T ^{註3}	$T_A = -40 \sim +105^{\circ}\text{C}$				120	mW
保存年限		$T_A = 85^{\circ}\text{C}$ ^{註2} , $N_{ERASE} \leq 1000$		10			年

注 1. 根据擦除计数 (N_{ERASE}) ,擦除时间不同。参考芯片擦除时间和 block 擦除时间参数。
2. 操作和不操作的平均温度是 85°C 时。

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

(注3 在下页列出。)

注 3. 需要进行自编程时，使用如下公式设计器件的功耗总和，要使器件的功耗小于或等于其总耗散功率 P_T （推荐使用小于等于 80% 的额定值）。

$$\bullet \text{ 总功耗} = V_{DD} \times \{I_{DD} - \Sigma I_{OH}\} + \Sigma \{ (V_{DD} - V_{OH}) \times I_{OH} \} + \Sigma (V_{OH} \times I_{OL})$$

当使用内部上拉电阻时，使用如下公式计算它的功耗，并且将结果加到上面的结果中。

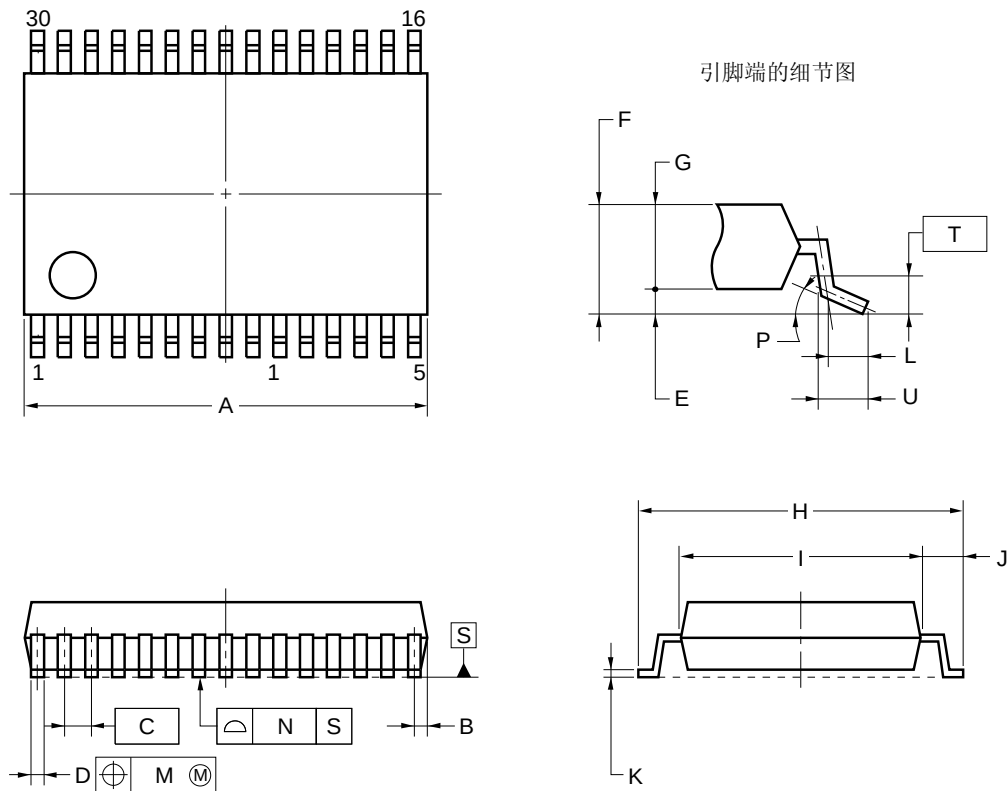
$$\bullet \text{ 内部上拉电阻的功耗} = \Sigma (V_{DD}/R_{PU} \times V_{DD})$$

备注 在 flash 存储器编程时， $I_{DD} = 7.0 \text{ mA}$ （最大值）。

第二十四章 封装图

- μ PD78F9232MC-5A4-A, 78F9234MC-5A4-A, 78F9232MC(A)-5A4-A, 78F9234MC(A)-5A4-A, 78F9232MC(A2)-5A4-A, 78F9234MC(A2)-5A4-A

30 脚塑封 SSOP (7.62 mm (300))



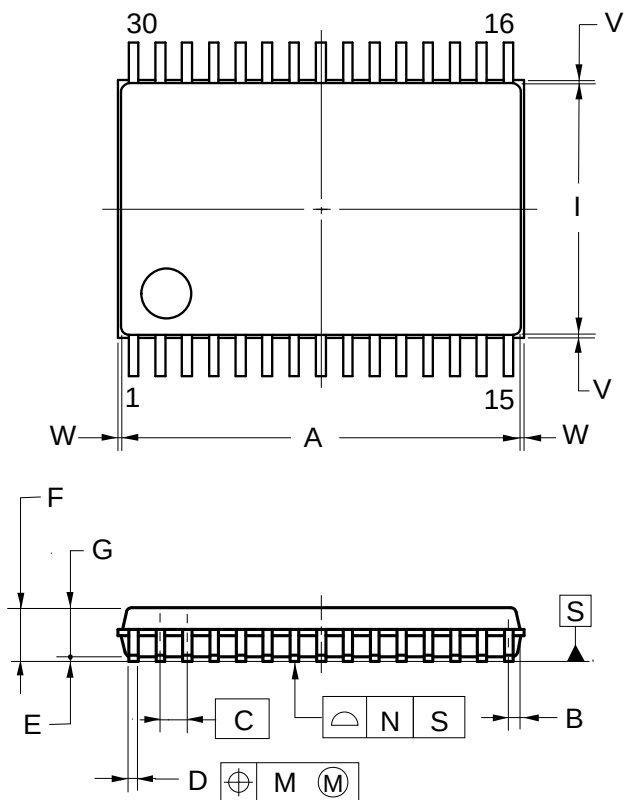
注
在最大使用材料情况下，每条引脚的中心线
位于其实际位置(T.P.)的0.13 mm 内。

项目	毫米
A	9.85±0.15
B	0.45 MAX.
C	0.65 (T.P.)
D	0.24 ^{+0.08} _{-0.07}
E	0.1±0.05
F	1.3±0.1
G	1.2
H	8.1±0.2
I	6.1±0.2
J	1.0±0.2
K	0.17±0.03
L	0.5
M	0.13
N	0.10
P	3° ^{+5°} _{-3°}
T	0.25
U	0.6±0.15

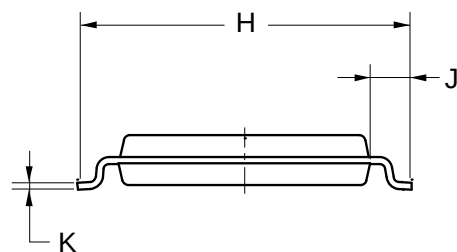
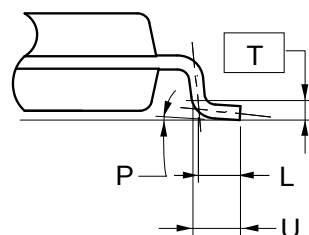
S30MC-65-5A4-2

- μ PD78F9232MC(A)-CAB-AX, 78F9234MC(A)-CAB-AX, 78F9232MC(A2)-CAB-AX, 78F9234MC(A2)-CAB-AX

30 脚塑封 SSOP (7.62mm (300))



引脚端的细节图



(单位:mm)

项目	尺寸
A	9.70±0.10
B	0.30
C	0.65 (T.P.)
D	0.22 ^{+0.10} _{-0.05}
E	0.10±0.05
F	1.30±0.10
G	1.20
H	8.10±0.20
I	6.10±0.10
J	1.00±0.20
K	0.15 ^{+0.05} _{-0.01}
L	0.50
M	0.13
N	0.10
P	3° ^{+5°} _{-3°}
T	0.25(T.P.)
U	0.60±0.15
V	0.25 MAX.
W	0.15 MAX.

P30MC-65-CAB

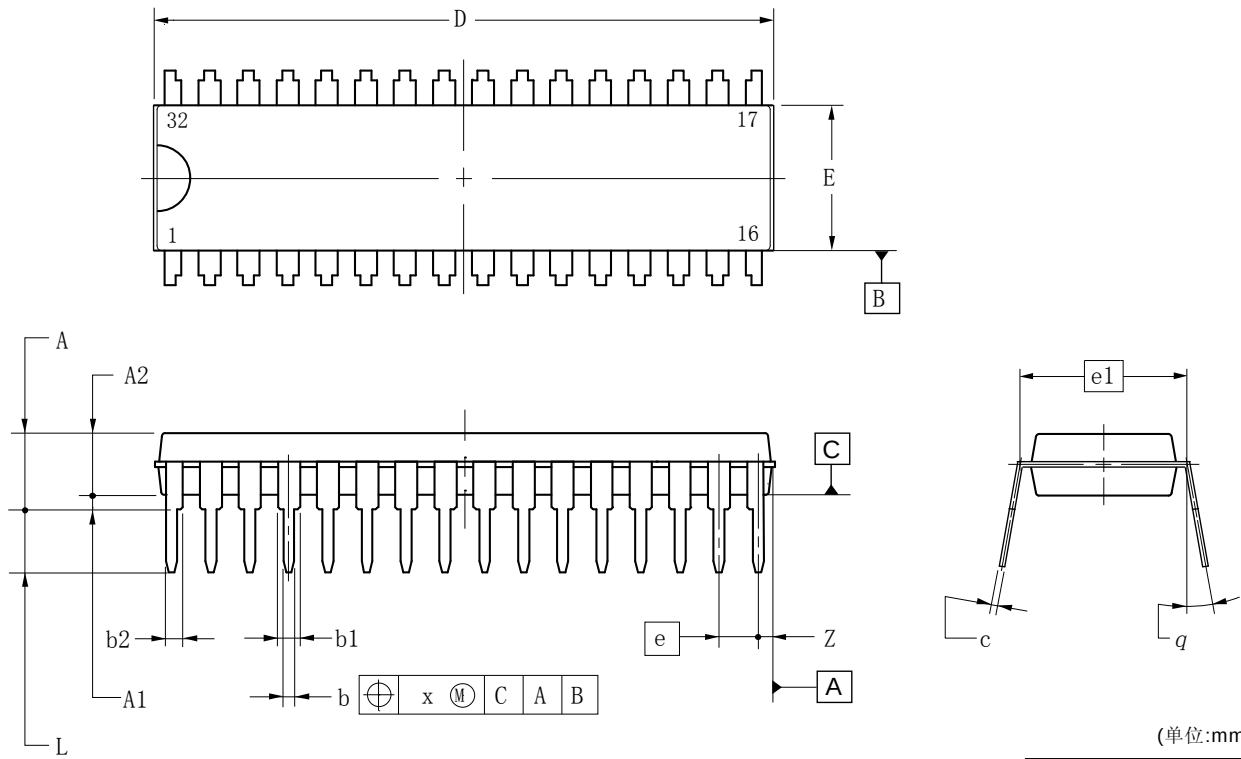
© NEC Electronics Corporation 2008

注

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

• μ PD78F9232CS-CAA-A, 78F9234CS-CAA-A

32 脚塑封 SDIP (7.62mm (300))



(单位:mm)

项目	尺寸
D	28.05 $\pm$ 0.15
E	6.60 $\pm$ 0.20
A	3.45 $\pm$ 0.15
A1	0.65 $\pm$ 0.10
A2	2.80
e	1.778
e1	7.62
b	0.50 $\pm$ 0.10
b1	1.02 $\pm$ 0.10
b2	0.75 $\pm$ 0.10
c	0.25 $^{+0.10}_{-0.05}$
L	2.86 $\pm$ 0.20
x	0.25
q	0° to 15°
Z	0.615

P32CS-70CAA

© NEC Electronics Corporation 2008

第二十五章 推荐焊接条件

这些产品建议在下列条件下进行焊接和安装。

相关技术信息，请浏览下面的网页。

半导体器件装配手册 (<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 或 -AX，是无铅产品。
 2. 如果不使用以下的推荐焊接方法和条件，请与 NEC 电子公司销售代理联系。

表 25-1. 表面贴装类型焊接条件 (1/2)

● 30 脚塑封 SSOP

μPD78F9232MC-5A4-A, 78F9234MC-5A4-A, 78F9232MC(A)-5A4-A, 78F9234MC(A)-5A4-A,
78F9232MC(A2)-5A4-A, 78F9234MC(A2)-5A4-A

焊接方法	焊接条件	推荐条件符号
红外回流	封装峰值温度: 260°C, 时间: 最长 60 秒 (大于等于 220°C), 次数: 小于等于 3 次, 曝光限制: 7 天 [※] (此后, 在 125°C 预烘 10 ~ 72 小时)	IR60-107-3
波峰焊	金属熔化温度: 最大 260°C, 时间: 最长 10 秒, 次数: 一次, 预热温度: 最高 120°C (包装表面温度), 曝光限制: 7 天 [※] (此后, 在 125°C 预烘 10 ~ 72 小时)	WS60-107-1
局部加热	引脚温度: 350°C 最大值, 时间: 最大 3 秒 (每行引脚)	—

注 打开干燥包之后，在有效存放期内将其存储在 25°C 或更低温度，65% RH 或更低湿度的环境中。

注意事项 不要一起使用不同的焊接方法（局部加热法除外）。

表 25-1. 表面贴装类型焊接条件 (2/2)

● 30 脚塑封 SSOP (无铅产品)

μPD78F9232MC(A)-CAB-AX, 78F9234MC(A)-CAB-AX, 78F9232MC(A2)-CAB-AX,
78F9234MC(A2)-CAB-AX

焊接方法	焊接条件	推荐条件符号
红外回流	封装峰值温度: 260°C, 时间: 最长 60 秒 (大于等于 220°C), 次数: 小于等于 3 次, 曝光限制: 7 天 [※] (此后, 在 125°C 预烘 10 ~ 72 小时)	IR60-207-3
波峰焊	焊锡炉温度: 260°C 最大值, 时间: 最长 10 秒, 次数: 1 次, 预热温度: 120°C 最大值 (封装表面温度), 曝光限制: 7 天 [※] (此后, 在 125°C 预烘 10 ~ 72 小时)	WS60-107-1
局部加热	引脚温度: 350°C 最大值, 时间: 最大 3 秒 (每行引脚)	—

注 1. 开发中

2. 开干燥包之后, 在有效存放期内将其存储在 25°C 或更低温度, 65% RH 或更低湿度的环境中。

注意事项 不要一起使用不同的焊接方法 (局部加热法除外)。

● 32 脚塑封 SDIP (无铅产品)

μPD78F9232CS-CAA-A, 78F9234CS-CAA-A

焊接方法	焊接条件	推荐条件符号
波峰焊 (仅对引脚)	焊锡炉温度: 260°C, 时间: 最长 10 秒。	—
局部加热	引脚温度: 350°C 最大值., 时间: 最长 3 秒 (每个引脚)	—

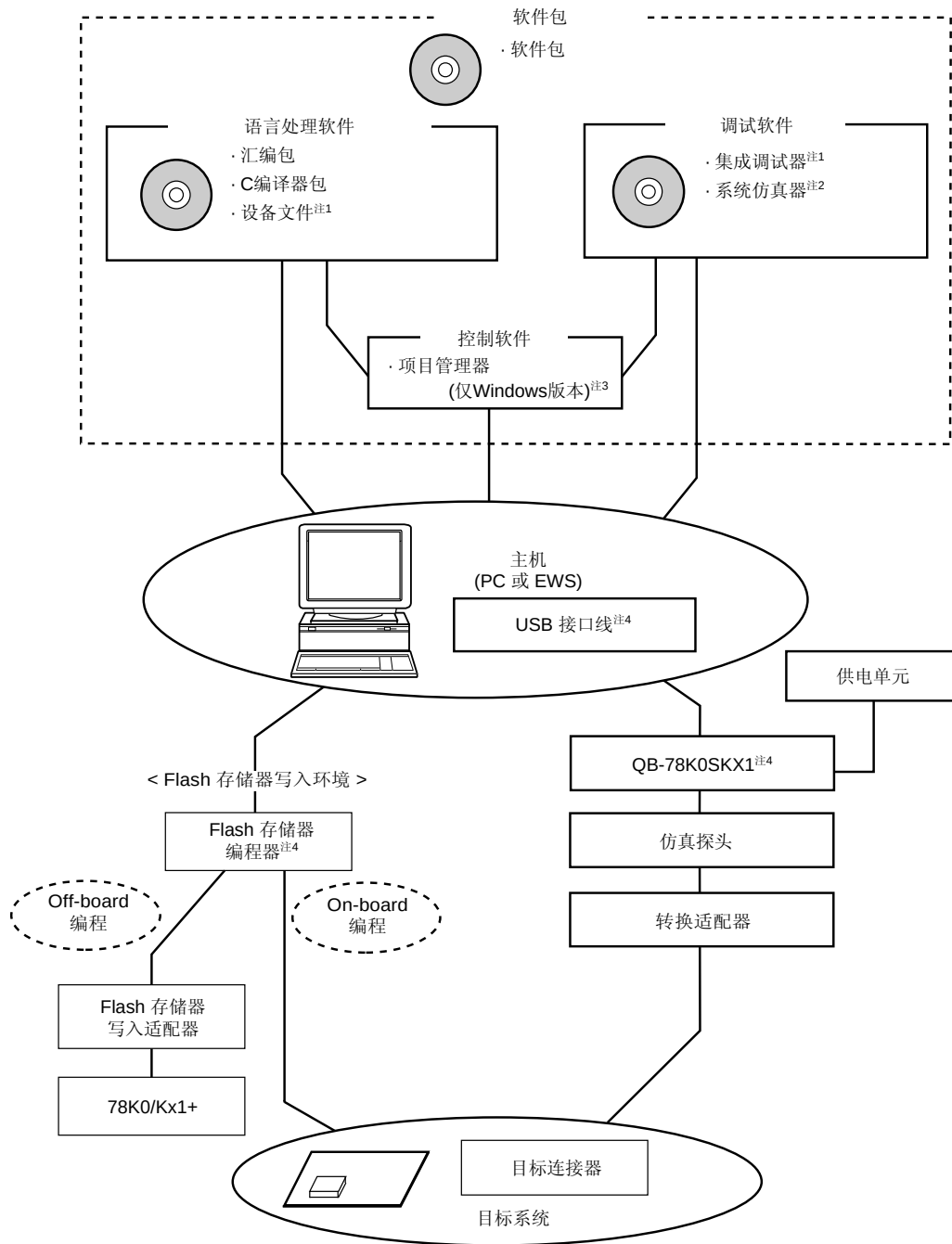
注意事项 在进行波峰焊时, 只对过孔插装元件 (THD) 的引脚加热。
确保波峰焊料不要接触封装。

附录A 开发工具

在 78K0S/KB1+的系统开发中可以使用下列开发工具。图 A-1 显示了开发系统的组成。

图 A-1. 开发工具 (1/2)

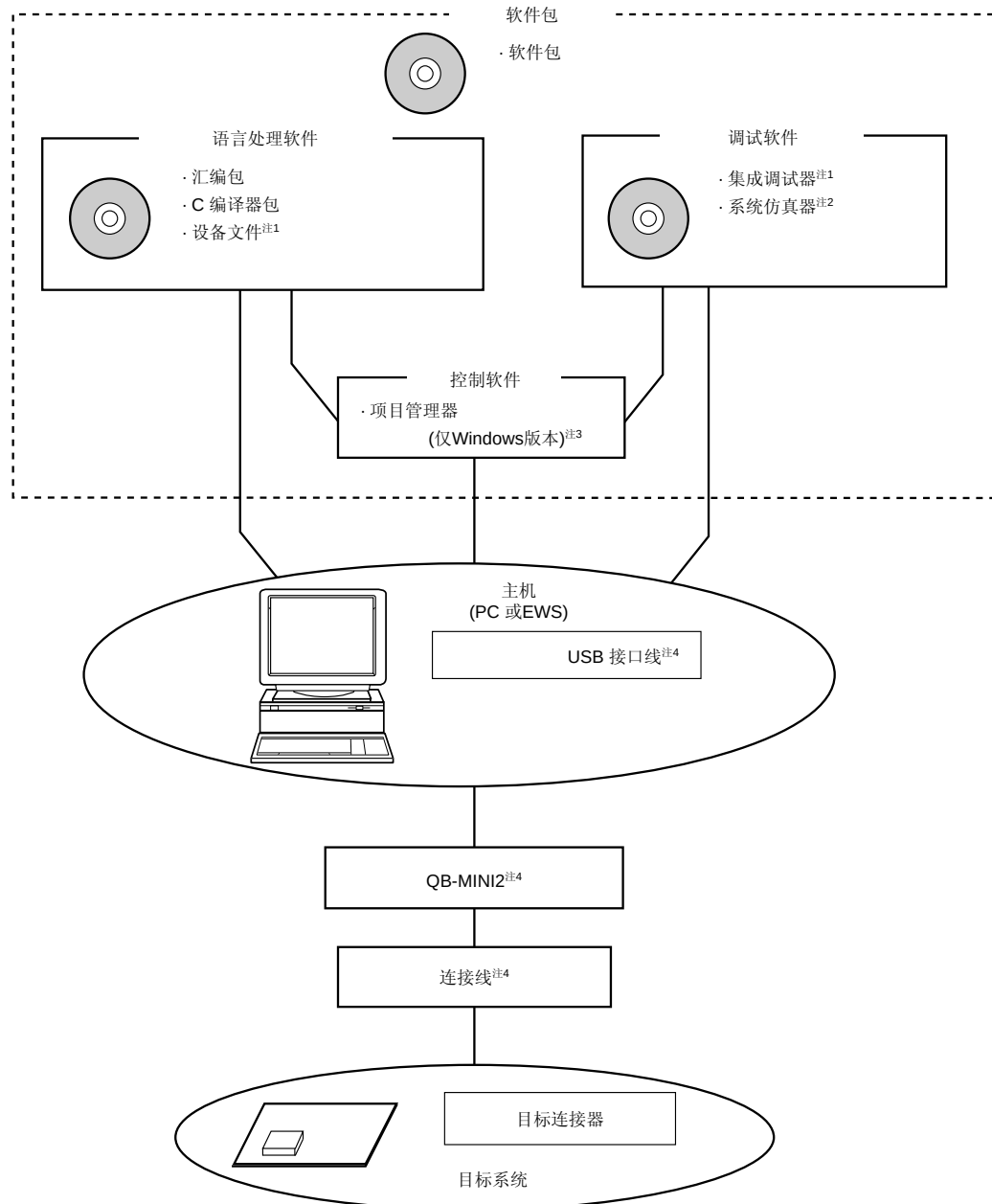
(1) 当使用在线仿真器 QB-78K0SKX1 时



- 注
1. 从下载网站下载 78K0S/Kx1+ 微控制器的设备文件 (DF789234) 和集成调试器 ID78K0S-QB 开发工具。
(<http://www.necel.com/micro/en/ods/>)。
 2. 78K0S 的 SM+ (指令仿真版本) 包含在软件包里。78K0S/Kx1+ 的 SM+ (指令+外围仿真版本) 不包含在其中。
 3. 项目管理器 PM+ 包含在汇编包中。PM+ 不能用于 WindowsTM 以外的操作系统。
 4. QB-78K0SKX1 包含集成调试器 ID78K0S-QB, USB 接口线, 带编程功能 QB-MINI2 的片上仿真器和连接线。其他产品单独出售。

图 A-1. 开发工具 (2/2)

(2) 当使用带有编程功能的片上调试仿真器 QB-MINI2 时



- 注
1. 从下载网站下载 78K0S/Kx1+微控制器的设备文件 (DF789234) 和集成调试器ID78K0S-QB开发工具。
(<http://www.necel.com/micro/en/ods/>)。
 2. 78K0S 的 SM+ (指令仿真版本) 包含在软件包里。78K0S/Kx1+的 SM+ (指令+外围仿真版本) 不包含在其中。
 3. 项目管理器 PM+包含在汇编包中。PM+不能用于 Windows™ 以外的操作系统。
 4. QB-MINI2 提供USB接口线和连接线, 其他产品单独出售。此外, QB-MINI2 的操作软件可以从开发工具下载网站下载 (<http://www.necel.com/micro/en/ods/>)

A.1 软件包

SP78K0S 78K0S 微控制器软件包	78K0S 微控制器通用开发工具（软件）合成在这个包里。
--------------------------	------------------------------

A.2 语言处理软件

RA78K0S ^{注1} 汇编程序包	该汇编程序将有助记符方式编写的程序转换为微控制器可执行的目标代码。 该汇编程序也拥有能够自动创建符号表和最佳分支指令的功能。 该汇编程序应该与一个设备文件（DF789234）联合使用。 <在 PC 环境中使用 RA78K0S 时的预防措施> 该汇编程序包是一个基于 DOS 系统的应用程序。但是，通过 Windows 上的项目管理器（PM+），它也可以用于 Windows。PM+ 包含在汇编程序包里。
CC78K0S ^{注1} C 编译程序包	该编译器将以 C 语言格式编写的程序转换为微控制器可执行的目标代码。 该编译器应该与一个设备文件联合使用。 <在 PC 环境中使用 CC78K0S 时的预防措施> 该 C 编译程序包是一个基于 DOS 系统的应用程序。但是，通过 Windows 上的项目管理器（PM+），它也可以用于 Windows。PM+ 包含在汇编程序包里。
DF789234 ^{注2} 设备文件	该文件包含设备特有的信息。 该设备文件应该与一个工具联合使用（RA78K0S, CC78K0S, ID78K0S-QB, 和系统仿真器）。 相应的 OS 和主机因使用的工具而不同。

- 注
1. 如果 RA78K0S 和 CC78K0S 的版本是 2.00 版本或之后的版本，RA78K0S 和 CC78K0S 的不同版本可以安装在相同的机器上。
 2. DF789234 可以和 RA78K0S, CC78K0S, ID78K0S-QB 及系统仿真器一起使用。从开发工具下载网站下载 DF789234（<http://www.necel.com/micro/en/ods/>）。

A.3 Flash存储器写入工具

A.3.1 当使用flash存储器编程器PG-FP5 和FL-PR5 时

FL-PR5, PG-FP5 Flash 存储器编程器	这是一个专门用于集成了 flash 存储器的微控制器的 flash 编程器。
FA-78F9202MA-CAC-RX Flash 存储写入适配器	这是一个用于连接 flash 存储器编程器的 flash 存储器写入适配器。

备注 1. FL-PR5 和FA-78F9202MA-CAC-RX是Naito Densei Machida Mfg. Co., Ltd的产品（<http://www.ndk-m.co.jp/>, e-mail: info@ndk-m.co.jp）。

2. 请使用 flash 存储器编程适配器的最新版本。

A.3.2 当使用带编程功能的片上调试仿真器QB-MINI2 时

QB-MINI2 带编程功能的片上调试仿真器	这是一个专门用于带片上 flash 存储器微控制器的 flash 存储器编程器。当使用 78K0S/Kx1+微控制器开发应用系统时，它也可以用于调试硬件和软件的片上调试仿真器。当将其用作 flash 存储器编程器时，应该与一个连接线和用于连接主机的 USB 线一起使用。
目标连接器说明	16 引脚通用连接器（2.54 mm 间距）

备注 从下载网上下载操作QB-MINI2 的开发工具软件（<http://www.necel.com/micro/en/ods/>）。

A.4 调试工具（硬件）

A.4.1 当使用在线仿真器QB-78K0SKX1 时

QB-78K0SKX1 在线仿真器	当使用 78K0S/Kx1+微控制器开发应用系统时，该在线仿真器用作调试硬件和软件的片上调试仿真器。它支持集成调试器（ID78K0S-QB）。该仿真器应该与电源单元和仿真探头一起使用，并使用 USB 将该仿真器连接到主机上。
QB-50-EP-01T [※] 仿真探头	这是一个柔性仿真探头，用于连接在线仿真器和目标系统。
QB-10MA-EA-01T [※] 交换适配器	该交换适配器用于执行从在线仿真器到目标连接器的引脚转换。
QB-10MA-NQ-01T [※] 目标连接器	该目标连接器用于在目标系统上安装。
目标系统上引脚的说明	0.635 mm × 0.635 mm （高度： 6 mm）

（注和备注列在下一页或后面。）

注 交换适配器和目标连接器的产品编号和目标设备的封装如下所示

封装	交换适配器	目标连接器
30 引脚塑封 SSOP (MC-5A4 和 MC-CAB 型)	QB-30MC-EA-04T	QB-30MC-NQ-02T
32 引脚塑封 SDIP (CS-CAA 型)	无	无

- 备注**
1. 对于 SDIP 封装，使用目标电线。
 2. QB-78K0SKX1 提供集成调试器 ID78K0S-QB, 一个 USB 接口线，片上调试仿真器 QB-MINI2 和一个连接线。
当使用 QB-MINI2 时，从开发工具下载网站下载操作 QB-MINI2 的软件
(<http://www.necel.com/micro/en/ods/>)。
 3. QB-78K0SKX1 打包的内容因产品编号而不同，列表如下。

打包的内容 产品编号	在线仿真器	仿真探头	交换适配器	目标连接器
QB-78K0SKX1-ZZZ	QB-78K0SKX1	无		
QB-78K0SKX1-T10MA		QB-50-EP-01T	QB-10MA-EA-01T	QB-10MA-NQ-01T

A.4.2 当使用带编程功能的片上调试仿真器QB-MINI2 时

QB-MINI2 带编程功能的片上调试仿真器	当使用 78K0S/Kx1+微控制器开发应用系统时，该片上调试仿真器用于调试硬件和软件。它也可以专门用作带片上 flash 存储器微控制器的 flash 编程器。当用作片上调试仿真器，应该与连接线和用于连接主机的 USB 接口线一起使用。
目标连接器说明	16 引脚通用连接器（2.54 mm 间距）

备注 从开发工具下载网站下载操作QB-MINI2 的软件（<http://www.necel.com/micro/en/ods/>）。

A.5 调试工具（软件）

ID78K0S-QB ^注 （支持 QB-78K0SKX1, QB- MINI2）集成调试器	该调试器支持 78K0S/Kx1+微控制器的在线仿真器。ID78K0S-QB 是基于 Windows 操作系统的软件。 提供的调试功能支持 C 语言、源程序编辑、反汇编显示和存储器显示。 它应该和设备文件（DF789234）一起使用。
SM+ for 78K0S SM+ for 78K0S/Kx1+ ^注 系统仿真器	系统仿真器是基于 Windows 操作系统的软件。 当在主机上仿真目标系统的操作时，它用于执行在 C 源程序水平或汇编程序水平的调试。 系统仿真器的使用允许应用逻辑测试的执行和与硬件开发无关的性能测试，所以可以提供更高的开发效率和软件质量。 系统仿真器应该和设备文件（DF789234）一起使用。 可以使用下列两类支持 78K0S/Kx1+的系统仿真器类型。 • SM+ for 78K0S（指令仿真版本） 它只能仿真一个 CPU。它包含在软件包里。 (SM+ for 78K0S/Kx1+ (指令+外围仿真版本)) 它可以仿真一个 CPU 和外围硬件（端口、定时器、串行接口等）。

注 从开发工具下载网站下载ID78K0S-QB（<http://www.necel.com/micro/en/ods/>）。

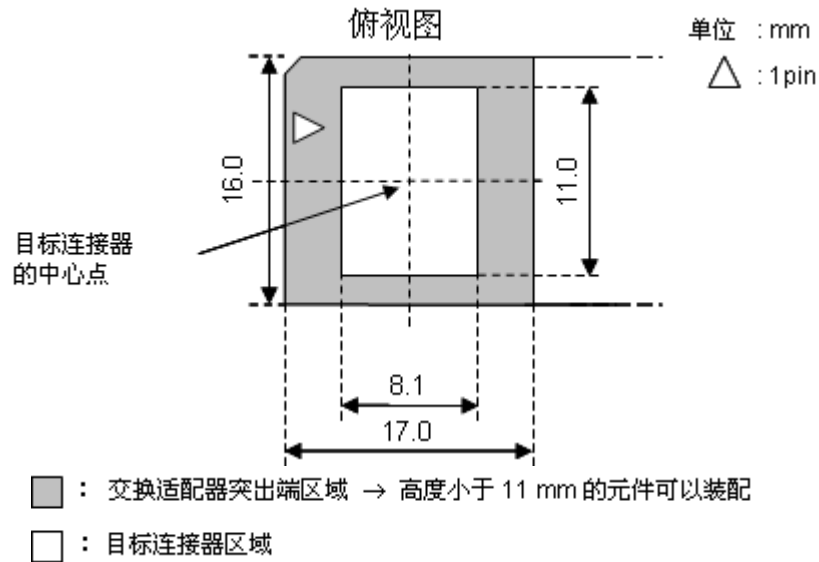
附录B 目标系统设计注意事项

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。

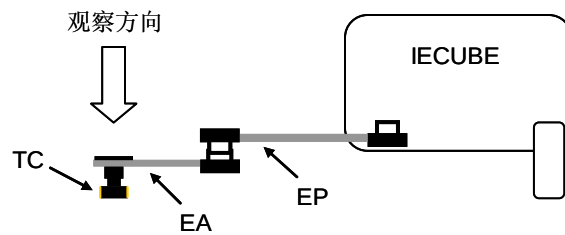
关于目标连接器、交换适配器、和仿真探头的封装图，请参见如下网址。

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

图 B-1. 当使用 78K0S/Kx1+ 仿真探头时 (30 脚 MC 封装)

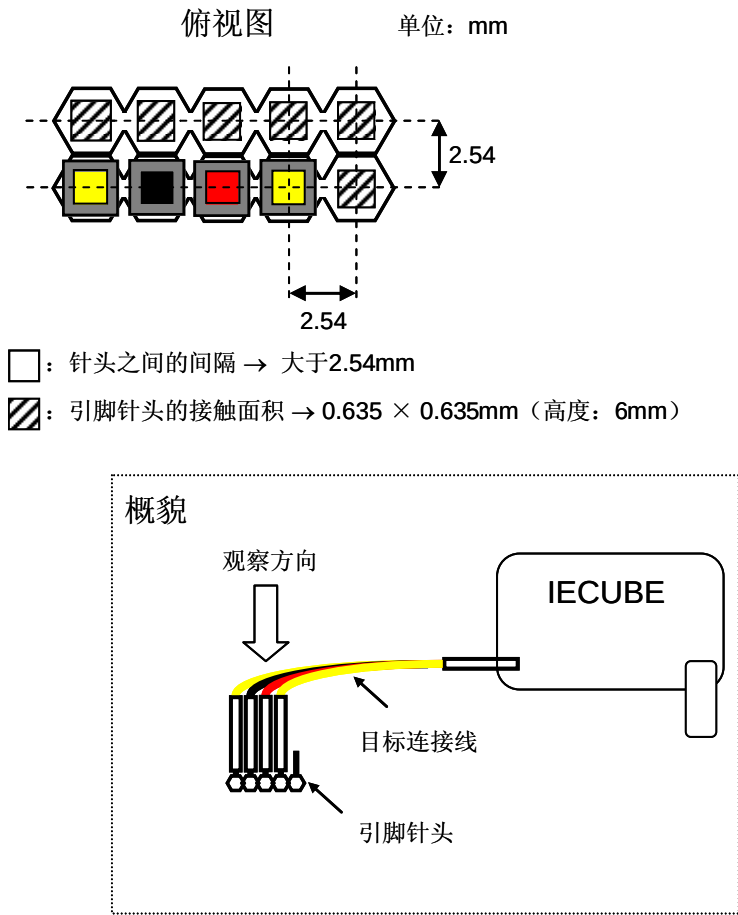


概貌



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+ 目标线(单道)



C.1 寄存器索引(寄存器名称)

8 位 A/D 转换结果寄存器 (ADCRH) ... 171
 8 位比较寄存器 80 (CR80) ... 131
 8 位定时器计数器 80 (TM80) ... 131
 8 位定时器 H 比较寄存器 01 (CMP01) ... 138
 8 位定时器 H 比较寄存器 11 (CMP11) ... 138
 8 位定时器 H 模式寄存器 1 (TMHMD1) ... 139
 8 位定时器模式控制寄存器 80 (TMC80) ... 132
 10 位 A/D 转换结果寄存器 (ADCR) ... 170
 16 位定时器捕捉/比较寄存器 000 (CR000) ... 90
 16 位定时器捕捉/比较寄存器 010 (CR010) ... 92
 16 位定时器计数器 00 (TM00) ... 90
 16 位定时器模式控制寄存器 00 (TMC00) ... 93
 16 位定时器输出控制寄存器 00 (TOC00) ... 96
 16 位乘法结果存储寄存器 0(MUL0)... 220

[A]

A/D 转换器模式寄存器(ADM) ... 168
 模拟输入通道指定寄存器 (ADS) ... 170
 异步串行接口控制寄存器 6 (ASICL6) ... 195
 异步串行接口操作模式寄存器 6 (ASIM6) ... 189
 异步串行接口接收错误状态寄存器 6 (ASIS6) ... 191
 异步串行接口发送状态寄存器 6 (ASIF6) ... 192

[B]

波特率发生器控制寄存器 6 (BRGC6) ... 194

[C]

捕捉/比较控制寄存器 00 (CRC00) ... 95
 时钟选择寄存器 6 (CKSR6) ... 193

[E]

外部中断模式寄存器 0 (INTM0) ... 231
 外部中断模式寄存器 1 (INTM1) ... 232

[F]

Flash 地址指针 H (FLAPH) ... 290
 Flash 地址指针 L (FLAPL) ... 290
 Flash 地址指针 H 比较寄存器 (FLAPHC) ... 290
 Flash 地址指针 L 比较寄存器 (FLAPLC) ... 290
 Flash 编程命令寄存器 (FLCMD) ... 289
 Flash 编程模式控制寄存器 (FLPMC) ... 286
 Flash 保护命令寄存器(PFCMD) ... 287
 Flash 状态寄存器(PFS) ... 287

Flash 写入缓冲器寄存器(FLW) ... 291

[I]

输入切换控制寄存器(ISC) ... 197

中断屏蔽标志寄存器 0 (MK0) ... 230

中断屏蔽标志寄存器 1 (MK1) ... 230

中断请求标志寄存器 0 (IF0) ... 229

中断请求标志寄存器 1 (IF1) ... 229

[L]

低电压检测寄存器(LVIM) ... 261

低电压检测电平选择寄存器(LVIS) ... 262

低速内部振荡模式寄存器(LSRCM) ... 76

[M]

乘法数据寄存器 A (MRA0) ... 220

乘法数据寄存器 B (MRB0) ... 220

乘法器控制寄存器 0 (MULC0) ... 222

[O]

振荡稳定时间选择寄存器(OSTS) ... 77, 240

[P]

端口模式控制寄存器 2 (PMC2) ... 69, 171

端口模式寄存器 0 (PM0) ... 67

端口模式寄存器 2 (PM2) ... 67, 171

端口模式寄存器 3 (PM3) ... 67, 98

端口模式寄存器 4 (PM4) ... 67, 141, 197

端口模式寄存器 12 (PM12) ... 67

端口寄存器 0 (P0) ... 68

端口寄存器 2 (P2) ... 68

端口寄存器 3 (P3) ... 68

端口寄存器 4 (P4) ... 68

端口寄存器 12 (P12) ... 68

端口寄存器 13 (P13) ... 68

预处理器时钟控制寄存器(PPCC) ... 75

预分频模式寄存器 00 (PRM00) ... 97

处理器时钟控制寄存器(PCC) ... 75

上拉电阻选择寄存器 0 (PU0) ... 70

上拉电阻选择寄存器 2 (PU2) ... 70

上拉电阻选择寄存器 3 (PU3) ... 70

上拉电阻选择寄存器 4 (PU4) ... 70

上拉电阻选择寄存器 12 (PU12) ... 70

[R]

接收缓冲器寄存器 6 (RXB6) ... 188

接收移位寄存器 6 (RXS6) ... 188

复位控制标志寄存器(RESF) ... 255

[T]

发送缓冲寄存器 6 (TXB6) ... 188

发送移位寄存器 6 (TXS6) ... 188

[W]

看门狗定时器允许寄存器(WDTE) ... 155

看门狗定时器模式寄存器(WDTM) ... 154

C.2 寄存器索引 (符号)

[A]

ADCR: 10 位 A/D 转换结果寄存器... 170
ADCRH: 8 位 A/D 转换结果寄存器... 170
ADM: A/D 转换器模式寄存器... 168
ADS: 模拟输入通道指定寄存器... 170
ASICL6: 异步串行接口控制寄存器 6 ... 195
ASIF6: 异步串行接口发送状态寄存器 6 ... 192
ASIM6: 异步串行接口操作模式寄存器 6 ... 189
ASIS6: 异步串行接口接收错误状态寄存器 6 ... 191

[B]

BRGC6: 波特率发生器控制寄存器 6 ... 194

[C]

CKSR6: 时钟选择寄存器 6 ... 193
CMP01: 8 位定时器 H 比较寄存器 01 ... 138
CMP11: 8 位定时器 H 比较寄存器 11 ... 138
CR000: 16 位定时器捕捉/比较寄存器 000 ... 90
CR010: 16 位定时器捕捉/比较寄存器 010 ... 92
CR80: 8 位比较寄存器 80 ... 131
CRC00: 捕捉/比较控制寄存器 00 ... 95

[F]

FLAPH: Flash 地址指针 H ... 290
FLAPHC: Flash 地址指针 H 比较寄存器... 290
FLAPL: Flash 地址指针 L ... 290
FLAPLC: Flash 地址指针 L 比较寄存器... 290
FLCMD: Flash 编程命令寄存器... 289
FLPMC: Flash 编程模式控制寄存器... 286
FLW: Flash 写入缓冲器寄存器... 291

[I]

IF0: 中断请求标志寄存器 0 ... 229
IF1: 中断请求标志寄存器 1 ... 229
INTM0: 外部中断模式寄存器 0 ... 231
INTM1: 外部中断模式寄存器 1 ... 232
ISC: 输入切换控制寄存器... 197

[L]

LSRCM: 低速内部振荡模式寄存器... 76
LVIM: 低电压检测寄存器... 261
LVIS: 低电压检测电平选择寄存器 ... 262

[M]

MK0: 中断屏蔽标志寄存器 0 ... 230
MK1: 中断屏蔽标志寄存器 1 ... 230
MRA0: 乘法数据寄存器 A ... 220
MRB0: 乘法数据寄存器 B... 220
MUL0: 16 位乘法结果存储寄存器 0... 220
MULC0: 乘法器控制寄存器 0 ... 222

[O]

OSTS: 振荡稳定时间选择寄存器... 77, 240

[P]

P0: 端口寄存器 0 ... 68
P2: 端口寄存器 2 ... 68
P3: 端口寄存器 3 ... 68
P4: 端口寄存器 4 ... 68
P12: 端口寄存器 12 ... 68
P13: 端口寄存器 13 ... 68
PCC 处理器时钟控制寄存器... 75
PFCMD: Flash 保护命令寄存器... 287
PFS: Flash 状态寄存器... 287
PM0: 端口模式寄存器 0 ... 67
PM2: 端口模式寄存器 2 ... 67, 171
PM3: 端口模式寄存器 3 ... 67, 98
PM4: 端口模式寄存器 4 ... 67, 141, 197
PM12: 端口模式寄存器 12 ... 67
PMC2: 端口模式控制寄存器 2 ... 69, 171
PPCC 预处理器时钟控制寄存器... 75
PRM00: 预分频模式寄存器 00 ... 97
PU0: 上拉电阻选择寄存器 0 ... 70
PU2: 上拉电阻选择寄存器 2 ... 70
PU3: 上拉电阻选择寄存器 3 ... 70
PU4: 上拉电阻选择寄存器 4 ... 70
PU12: 上拉电阻选择寄存器 12 ... 70

[R]

RESF: 复位控制标志寄存器... 255
RXB6: 接收缓冲器寄存器 6 ... 188
RXS6: 接收移位寄存器 6 ... 188

[T]

TM00: 16 位定时计数器 00 ... 90
TM80: 8 位定时计数器 80 ... 131
TMC00: 16 位定时器模式控制寄存器 00 ... 93
TMC80: 8 位定时器模式控制寄存器 80 ... 132
TMHMD1: 8 位定时器 H 模式寄存器 1 ... 139
TOC00: 16 位定时器输出控制寄存器 00 ... 96

TXB6: 发送缓冲寄存器 6 ... 188

TXS6: 接收移位寄存器 6 ... 188

[W]

WDTE: 看门狗定时器允许寄存器... 155

WDTM: 看门狗定时器模式寄存器... 154

附录 D 注意事项列表

本附录列出本文档中描述的注意事项。

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：软件注意事项，比如寄存器设置或编程

(1/20)

章节	类别	功能	功能详解	注意事项	页码
第一章	硬件	引脚配置	AVss 引脚	连接 AVss 引脚到 Vss。	p.16 ☐
第二章	硬件	引脚功能	P121/X1 和 P122/X2 引脚	在复位期间，P121/X1 和 P122/X2 引脚处于下拉状态。	pp. 21, 22, 24, 25 ☐
第三章	硬件	存储器空间	向量表地址	向量表地址 0014H 不对应任何中断源。	p.30 ☐
			SP：堆栈指针	由于复位信号的产生会使 SP 的内容不确定，所以在使用堆栈前必须先对 SP 初始化。	p.34 ☐
				堆栈指针只能设置到高速 RAM 区域，并且只有低 10 位可以真正设置。因此，如果堆栈指针被指定到 0FF00H，它要转换成高速 RAM 区域的 0FB00H，因为 0FF00H 在 SFR 区域内，并且不是高速 RAM 区域。当值被真正压进栈时，0FB00H 减 1 变成 0FAFFH，但是因为该值不在高速 RAM 区域，它被转换成 0FEFFH，与当 0FF00H 设置到堆栈指针时的值相同。	p.34 ☐
第四章	硬件	端口功能	P121/X1和 P122/X2引脚	在复位期间，P121/X1 和 P122/X2 引脚为下拉状态。	p.53 ☐
			P34 引脚	因为引脚 P34 可以作为 $\overline{\text{RESET}}$ 引脚使用，所以当它作为输入端口引脚使用的时候，外部输入一个复位信号也不能实现复位功能。这个端口的功能通过选项字节来选择。详情请参见第十八章 选项字节。 同样，因为在复位释放之后参考选项字节，如果在参考选项字节前，有一个低电平输入到 $\overline{\text{RESET}}$ 引脚，那么复位状态不会被释放。当其作为输入端口引脚时，应连接上拉电阻。	p.59 ☐
			P30, P31, 和 P43 引脚	因为 P30, P31 和 P43 引脚也被用作外部中断引脚，这些引脚的任何一个被设置成输出模式，并且它的输出电平被改变，则相应的中断请求标志被置位。因此使用这个端口引脚为输出模式时，于是，预先设置相应的中断屏蔽标志为 1。	p.66 ☐
			PMC：端口模式控制寄存器 2	当 PMC20 ~ PMC23 设置为 1 时，P20/ANI0 ~ P23/ANI3 引脚不能用做端口引脚。将引脚设置为 A/D 转换器模式时，确保上拉电阻选择寄存器（PU20 ~ PU23）置为 0。	p.69 ☐
			—	虽然 1 位存储器操作指令只对 1 位进行操作，但实际是以 8 位形式访问端口。因此，如果一个端口既有输入又有输出引脚，则对于端口中定义为输入模式的引脚，其输出锁存器中的内容是不确定的。	p.71 ☐

章节	类别	功能	功能详解	注意事项	页码
第五章	软件	主时钟	OSTS: 振荡稳定时间选择寄存器	设置和释放 STOP 模式时, 按照下面的方法设置振荡稳定时间。 OSTS 设置的振荡稳定时间大于等于预期的振荡器振荡稳定时间。	p.77 ☐
				无论 STOP 模式是被复位信号唤醒还是被中断唤醒, STOP 模式释放后的等待时间不包括从释放 STOP 模式到时钟开始振荡的时间 (下图中的“a”部分)。	p.77 ☐
				复位释放后的振荡稳定时间可以通过选项字节来选择。详情请参见 第十八章 选项字节。	p.77 ☐
	硬件	晶体/陶瓷振荡器	—	当使用晶体/陶瓷振荡器时, 布线应该按照图 5-6 中虚线内的方式连接, 这样可以减小线间电容的负面影响。 ● 保证连线要尽可能的短。 ● 不要使其它的信号线与振荡器导线互相交叉。不要在通过大电流的信号线旁边进行振荡器的布线。 ● 保持振荡器电容的地和 Vss 同电位。不要将电容的接地点连接在大电流通过的地模式。 ● 不要从振荡器上取信号。	p.78 ☐
第六章	硬件	16 位定时器/事件计数器 00	TM00: 16 位定时器计数器 00	读取 TM00 时, CR010 无法捕获其值。	pp. 90, 122 ☐
				当读取 TM00 时, 不会发生计数丢失, 因为计数时钟暂时停止工作, 在读取后重新恢复。	pp. 90, 122 ☐
	软件		CR000: 16 位定时器捕获/比较寄存器 000	当 TM00 和 CR000 相等而进入清零&启动模式时, CR000 的值不要设置为 0000H。也就是说当此寄存器作为外部事件计数器时, 不能执行 1 个脉冲的计数操作。	pp. 91, 122 ☐
				然而, 在自由运行模式和由 TI000 引脚有效沿进入的清零&启动模式中, 如果 CR000 清为 0000H, 则 TM00 溢出后 (FFFFH), CR000 从 0000H 变为 0001H 时, 产生一个中断请求 (INTTM000)。	pp. 91, 122 ☐
				如果 CR000 的新值小于 16 位定时器计数器 0 (TM00) 的值, TM00 继续计数, 溢出, 然后从 0 开始重新计数。因此, 如果 CR000 的新值小于原来的值, 在 CR000 的值改变后, 定时器必须复位重新启动。	pp. 91, 122 ☐
				当 16 位定时器/事件计数器 00 停止工作后, CR000 的值是无法保证的。	pp. 91, 123 ☐
	硬件			当 CR000 设为比较模式, 即使有捕获触发信号输入, 也不执行捕获操作。	pp. 91, 126 ☐
				当 P31 用做 TI010 的有效沿输入引脚时, 不能用作定时器输出引脚 (TO00)。同样, 当 P31 被用于定时器输出引脚 TO00 时, 不能用于 TI010 的有效沿输入引脚。	pp. 91, 127 ☐

章节	类别	功能	功能详解	注意事项	页码	
第六章	硬件	16 位定时器/事件计数器 00	CR000: 16 位定时器捕获/比较寄存器 000	当 CR000 作为捕获寄存器使用, 如果同时发生寄存器的读操作和输入捕获触发操作, 则捕获触发输入有优先权, 而读出的数据为不确定值。同样, 如果定时器停止计数与捕获触发输入信号发生冲突, 捕获触发为不确定状态。	pp. ☐ 91, 125	
	软件			如果在 TM00 操作中改变 CR000 的设置, 可能引起错误。要改变设置, 请参考 6.5 16 位定时器/事件计数器 00 的相关注意事项 (17) 在定时器操作过程中改变比较寄存器。	p.92 ☐	
	硬件		CR010: 16 位定时器捕获/比较寄存器 010	在自由运行模式和由 TI000 引脚有效沿进入的清零&启动模式中, 如果 CR010 清为 0000H, 则 TM00 溢出后 (FFFFH), CR010 从 0000H 变为 0001H 时, 产生一个中断请求 (INTTM010)。	pp. ☐ 92, 122	
				如果 CR010 新值小于 16 位定时器计数器 0 (TM00) 的值, TM00 继续计数, 溢出, 然后从 0 开始重新计数。因此, 如果 CR010 的新值小于原来的值, 在 CR010 的值改变后, 定时器必须复位重新启动。	pp. ☐ 92, 122	
				当 16 位定时器/事件计数器 00 停止工作后, CR010 的值是无法保证的。	pp. ☐ 92, 123	
			当 CR010 设为比较模式, 即使有捕获触发信号输入, 也不执行捕获操作。	pp. ☐ 92, 126		
			当 CR010 作为捕获寄存器使用, 如果同时发生寄存器的读操作和输入捕获触发操作, 则捕获触发输入有优先权, 而读出的数据为不确定值。同样, 如果定时器停止计数与捕获触发输入发生冲突, 捕获触发为不确定状态。	pp. ☐ 92, 125		
			如果在 TM00 操作中改变 CR010 的设置, 可能引起错误。 要改变设置, 请参考 6.5 16 位定时器/事件计数器 00 的相关注意事项 (17) 在定时器操作过程中改变比较寄存器。	p.93 ☐		
	软件		TMC00: 16 位定时器模式控制寄存器 00	当 TMC002 和 TMC003 (操作停止模式) 的值不为 0 时, 16 位定时器 TM00 开始工作, 置为 0, 则不同。相反, 当 TMC002 和 TMC003 位的值设置为 0 时, 0 用来停止 16 位定时器 TM00。	pp. ☐ 93, 122	
				向 16 位定时器模式控制寄存器 TMC00 的各位写入数据前, 定时器必须停止工作, OVF00 标志位除外。	pp. ☐ 94, 123	
				如果定时器停止, 时钟计数和定时器中断都不会发生, 即使有信号输入 TI000/TI010 引脚。	pp. ☐ 94, 122	
				除 TI000 引脚有效沿作为计数时钟以外, 在设置 STOP 模式或系统时钟停止模式之前, 停止定时器工作; 否则在系统时钟启动时, 定时器将发生故障。	pp. ☐ 94, 127	
				当定时器停止工作后, 才能通过预分频模式寄存器 00 (PRM00) 的第 4 位和第 5 位来设置 TI000 引脚有效沿。	pp. ☐ 94, 123	
				如果 TM00 和 CR00 相等进入清零&启动模式时, 或因 TI000 有效沿进入清零&启动模式时, 或者定时器工作在自由工作模式时, 当 CR000 的值为 FFFFH, TM00 的值从 FFFFH 变为 0000H 时, OVF00 标志位置 1。	p.94 ☐	
				即便在 TM00 溢出之后, 而在下一个计数时钟计数之前 (TM00 变为 0001H 之前), 清除 OVF00 标志, OVF00 标志还会被重新置 1, 清除操作无效。	pp. ☐ 94, 124	

章节	类别	功能	功能详解	注意事项	页码
第六章	软件	16 位定时器/事件计数器 00	TMC00: 16 位定时器模式控制寄存器 00	在计数时钟的下降沿处执行捕获操作。然而，在下个计数时钟的上升沿，会产生一个中断请求输入（INTTM0n0）。	pp. ☐ 94, 125
			CRC00: 捕获/比较控制寄存器 00	设置 CRC00 之前，定时器操作必须停止。	pp. ☐ 95, 123
				当通过 16 位定时器模式控制寄存器 TMC00 选择 TM00 和 CR000 相等进入清零&启动模式时，CR000 就不能被指定为捕获寄存器。	pp. ☐ 95, 122
				为了确保捕获操作的可靠性，要求被捕获的信号宽度要大于 2 个计数时钟（由预分频器模式寄存器 00（PRM00）选择）（参见图 6-17）。	pp. ☐ 95, 125
	软件	TOC00: 16 位定时器输出控制寄存器 00	除了 OSPT00 以外，对其它寄存器设置之前，定时器必须停止工作。	pp. ☐ 96, 123	
			当读取 LVS00 和 LVR00 时，读出值为 0。	pp. ☐ 96, 123	
			数据设置之后，OSPT00 自动清零，读取该位时，读出值为 0。	pp. ☐ 96, 123	
			除单脉冲输出模式外，OSPT00 不可设置为 1。	pp. ☐ 96, 123	
	硬件		当连续设置 OSPT00 为 1 时，需要用预分频模式寄存器 00（PRM00）选择两个或更多计数时钟周期的写入间隔。	pp. ☐ 96, 123	
			当 TOE00 为 0 时，设置 TOE00，LVS00，和 LVR00，要用 8 位存储器操作指令同时设置。当 TOE00 为 1 时，LVS00，LVR00 能够用 1 位存储器操作指令设置。	p.96 ☐	
	软件	PRM00: 预分频模式寄存器 00	定时器停止工作后，才可以向 PRM00 写入数据。	pp. ☐ 98, 123	
			如果指定 TI000 引脚的有效沿作为计数时钟，就不要设置 TI000 引脚的有效沿时进入清零/开始模式和捕获触发器。	pp. ☐ 98, 125	
			在如下情况时，要注意 TI0n0 引脚有效沿的检测。 <1> 系统复位后，如果 TI0n0 引脚立即输入高电平，那么允许 16 位定时器计数器 00（TM00）操作。 → 如果上升沿或双沿被指定为 TI0n0 引脚的有效沿，TM00 操作允许后，马上会检测到一个上升沿。 <2> 如果当 TI0n0 引脚是高电平的时候，停止 TM00 操作，则 TI0n0 引脚输入低电平后，允许 TM00 操作。 → 如果下降沿或双沿被指定为 TI0n0 引脚的有效沿，TM00 操作允许后，马上会检测到一个下降沿 <3> 当 TI0n0 为低电平，TM00 停止工作时，则 TI0n0 引脚输入高电平后，允许 TM00 操作。 → 如果上升沿或双沿被指定为 TI0n0 引脚的有效沿，TM00 操作允许后，马上会检测到一个上升沿。		pp. ☐ 98, 127

(5/20)

章节	类别	功能	功能详解	注意事项	页码
第六章	硬件	16 位定时器/事件计数器 00	PRM00: 预分频模式寄存器 00	当 TI000 有效沿用作计数时钟和用作捕获触发时, 用来消除噪音的采样时钟是不同的。在前一种情况中, 计数时钟为 f _{XP} 。而在后一种情况中, 由预分频模式寄存器 00 (PRM00) 选择计数时钟。直到采样到有效沿, 并且连续两次检测到有效电平, 捕获操作才会完成, 因此可通过窄脉冲宽度消除噪声。	pp. 98, 127 ☐
				当 P31 用做 TI010 的有效沿输入引脚时, 不能用作定时器输出引脚 (TO00)。同样, 当 P31 被用于定时器输出引脚 TO00 时, 不能用于 TI010 的有效沿输入引脚。	pp. 98, 127 ☐
	软件		间隔定时器	如果在 TM00 操作中改变 CR000 的设置, 可能引起错误。要改变设置, 请参考 6.5 16 位定时器/事件计数器 00 的相关注意事项 (17) 在定时器操作过程中改变比较寄存器。	p.99 ☐
			外部事件计数器	需要知道外部事件计数器的计数值, 则应该去读 TM00。	pp. 103, 127 ☐
			脉冲宽度测量	要使用两个捕获寄存器时, 设置 TI000 和 TI010 引脚。	pp. 104, 125 ☐
				在这个操作示例中, 可测量的脉冲宽度可达一个定时器计数周期。	pp. 105, 107, 108, 110 ☐
	方波输出		如果在 TM00 操作中改变 CR000 的设置, 可能引起故障。要改变设置, 请参考 6.5 16 位定时器/事件计数器 00 的相关注意事项 (17) 在定时器操作过程中改变比较寄存器。	p.112 ☐	
	PPG 输出		如在 TM00 操作中改变 CR000 的设置, 可能引起故障。 要改变设置, 请参考 6.5 16 位定时器/事件计数器 00 的相关注意事项 (17) 在定时器操作过程中改变比较寄存器。	p.114 ☐	
			以下为 CR000 和 CR010 的取值范围: 0000H < CR010 < CR000 ≤ FFFFH	pp. 115, 127 ☐	
			通过 PPG 输出产生的脉冲的周期 (CR000 的设置值 + 1) 的占空比是: (CR010 的设置值+ 1) / (CR000 的设置值 + 1)。	pp. 115, 127 ☐	
	硬件	单脉冲输出: 软件触发	在单脉冲正在输出情况下, 不要再次置 OSPT00 位为 1。 再次输出单脉冲, 要等待正在进行的单脉冲输出完成之后。	pp. 117, 123 ☐	
			使用软件触发进行 16 位定时器/事件计数器 00 的单脉冲输出时, 不要改变 TI000 或其复用端口的引脚电平。 因为在这种情况下外部触发也有效, TI000 或其复用端口引脚的电平也可以对定时器清零并启动。这样会输出非预期的时序。	pp. 117, 123 ☐	
			CR000 和 CR010 寄存器不能设置为 0000H。	pp. 118, 123 ☐	
			只要为 TMC003 和 TMC002 位设置 00 之外的值 (操作停止模式), 16 位定时器计数器 00 开始操作。	pp. 119, 122 ☐	
软件					

章节	类别	功能	功能详解	注意事项	页码
第六章	硬件	16 位定时器/事件计数器 00	单脉冲输出： 外部触发	在单脉冲输出进行时，不要再次输入外部触发信号。再次输出单脉冲，要等待正在进行的单脉冲输出完成之后。	pp. ☐ 119, 124
				CR000 和 CR010 寄存器不能设置为 0000H 。	pp. ☐ 120, 124
				只要为 TMC003 和 TMC002 位设置 00 之外的值（操作停止模式），16 位定时器计数器 00 开始操作。	pp. ☐ 121, 122
	软件	硬件	定时器启动时的误差	在定时器启动后，产生信号匹配时，可能发生达一个时钟的误差。这是因为 16 位定时器计数器 00（TM00）的启动与计数时钟不同步。	p.122 ☐
			单脉冲输出	只有在自由运行模式或 TI000 引脚有效沿进入清零&启动模式下，单脉冲输出可以正常操作。因为在 TM00 和 CR000 相等进入清零&启动模式下，不会发生溢出，所以不可能输出单脉冲。	p.123 ☐
				捕获操作	如果上升沿和下降沿均被指定为 TI000 引脚的有效触发沿，则捕获无法进行。
		当 CRC001 值为 1，如果 TI010 引脚检测到有效沿，TM00 的计数值不能被捕获到 CR000 寄存器中，但是 TI010 引脚的输入可以用作外部中断源，因为外部有效信号输入时产生 INTTM000 中断。			p.125 ☐
		在定时器运行期间改变比较寄存器	16 位定时器捕获/比较寄存器 0n0（CR0n0）作为比较寄存器使用，在定时器计数期间，在 16 位定时器计数器 00（TM00）和 16 位定时器捕获/比较寄存器 0n0（CR0n0）的值相等的时序附近修改 CR0n0 的值时，改变 CR0n0 的时序和两者相等匹配的时序可能会冲突，这种情况下，操作无法得到保证。在定时器计数时改变 CR0n0 的值时，INTTM000 中断服务程序会执行如下操作。	p.126 ☐	
			如果在定时器计数期间改变 CR010 时，但未执行上面的过程<1>，那么 CR010 中的值将被重新写入两次或更多次，每次写入都将导致 TO00 引脚输出电平的反转。	p.126 ☐	
			外部事件计数器	检测到两次有效沿后，定时器开始计数。	p.127 ☐
			外部时钟限制	当使用 TI000 引脚的输入脉冲作为计数时钟（外部触发）时，确保输入脉冲宽度能够满足 AC 特性。 具体的 AC 特性，参见 第二十二章 和 第二十三章 电气特性 。	p.128 ☐
		当一个外部波形输入到 16 位定时器/事件计数器 00 时，由噪声限幅电路采样，因此在器件内部变为有效的时序会产生错误。		p.128 ☐	
第七章	软件	8 位定时器 80	CR80： 比较寄存器 80	当改变 CR80 值的时候，必须要停止定时器的操作。如果在定时器操作期间改变 CR80 的值，则立刻产生匹配中断请求信号，且定时器可能会清除。	p.131 ☐
			TMC80： 8 位定时器模式控制寄存器 80	对 TMC80 的设置，必须在定时器操作停止后。	p.132 ☐
				确保第 0 位和第 6 位清零。	p.132 ☐

章节	类别	功能	功能详解	注意事项	页码
第七章	软件	8 位定时器 80	间隔定时器	当改变 CR80 值的时候，必须要停止定时器的操作。如果在定时器工作期间改变 CR80 的值，则可能会立即产生相等中断请求信号。	p.133 ☐
				当使用 8 位存储操作指令设置 TMC80 的计数时钟，并同时允许 TM80 操作，则在定时器启动后一个周期的误差可能是一个或多个时钟（参见 7.5（1）定时器启动时的错误）。因此，要将 TM80 作为间隔定时器使用，必须按上述顺序进行设置。	p.133 ☐
	硬件		定时器启动时的误差	从启动定时器到产生匹配信号的时间，包含了一个最大 1.5 个时钟周期的误差。这是因为如果在计数时钟为高电平时启动定时器，那么可能立即检测到上升沿，且计数器累加（参见图 7—6）。	p.135 ☐
	软件		CR80: 比较寄存器 80	8 位比较寄存器 80（CR0）可设置为 00H。	p.135 ☐
			STOP 模式	在执行 STOP 指令之前，必须先停止定时器操作（TCE80=0）。	p.135 ☐
第八章	软件	8 位定时器 H1	CMP01: 8位定时器H 比较寄存器01	CMP01 在定时器计数期间不能被重写。	p.138 ☐
			CMP11: 8位定时器H 比较寄存器 11	在 PWM 输出模式中，当定时器停止计数后（TMHE1=0），再次启动计数操作时(TMHE1=1)，必须先对 CMP11 进行赋值（即使 CMP11 前后设置的值相同，也必须重新设置）。	p.138 ☐
			TMHMD1: 8 位定时器 H 模式寄存器 1	当 TMHE1 设置为 1 时，禁止对 TMHMD1 寄存器的其它位进行设置。	p.140 ☐
				在 PWM 输出模式下，当定时器停止计数后（TMHE1=0），再次启动计数操作时(TMHE1=1)，必须先对 CMP11 进行赋值（即使 CMP11 寄存器前后设置的值相同，也必须重新设置）。	p.140 ☐
	硬件	PWM 输出	在 PWM 输出模式下，CMP11 的设置值可以在定时器计数操作期间改变。但是，在重写 CMP11 寄存器的值之后，需要 3 个或更多操作时钟（使用 TMHMD1 寄存器的 CKS12 到 CKS10 位选择的信号）来传送寄存器的值。	p.146 ☐	
			当定时器停止计数后（TMHE1=0），再次启动计数操作时（TMHE1=1），必须对 CMP11 赋值（即使 CMP11 寄存器的设置值相同，也必须重新赋值）。	p.146 ☐	
			确保 CMP11 寄存器的设置值（M）和 CMP01 寄存器的设置值(N)在如下范围内： 00H ≤ CMP11 (M) < CMP01 (N) ≤ FFH	p.147 ☐	
第九章	软件	看门狗定时器	WDTM: 看门狗定时器模式寄存器	第 7，6，5 位分别设置为 0，1，和 1。不能设置为其他值。	p.155 ☐
				复位释放后，仅能由 8 位存储器操作指令对 WDTM 写入一次。如果试图写入第二次，就会产生内部复位信号。但是，在第一次写入时，如果分别设置 WDSC4 和 WDSC3 为 “1”和“x”，看门狗定时器将停止，即使有如下操作也不会产生内部复位。 • 第二次写入 WDTM • 用 1 位存储器操作指令对 WDTE 进行操作 • 给 WDTE 写入一个“ACH”以外的值	p.155 ☐

章节	类别	功能	功能详解	注意事项	页码
第九章	软件	看门狗定时器	WDTM: 看门狗定时器模式寄存器	不能用 1 位存储器操作指令对 WDTM 进行操作	p.155 ☐
				当使用自编程方式对 flash 存储器进行操作时, 为看门狗设置的溢出时间要足够长 (例如 一个字节写入: 最小 200 μ s, 一个块删除: 最小 10ms)。	p.155 ☐
			WDTE: 看门狗定时器允许寄存器	如果一个 ACH 之外的值被写入 WDTE, 将产生一个内部复位信号。	p.155 ☐
				如果使用 1 位存储器操作指令对 WDTE 进行操作, 将产生一个内部复位信号。	p.155 ☐
				从 WDTE 读取的值是 9AH (与写入值 (ACH) 不同)。	p.155 ☐
	硬件		通过选项字节设置“低速内部振荡器不能被停止”时	在这种模式中, 即使是执行 STOP 指令, 也不能停止看门狗定时器的操作。对于 8 位定时器 H1 (TMH1), 可选择低速内部振荡时钟的一个分频作为计数源, 因此在 STOP 指令执行后, 看门狗定时器溢出之前, 可使用 TMH1 的中断请求对看门狗定时器清零。如果没有执行该过程, 则在 STOP 指令执行后, 当看门狗定时器溢出时会产生内部复位信号。	p.156 ☐
第十章	软件	A/D 转换器	通过选项字节设置“低速内部振荡器可以被软件停止”时	此模式中, 在 HALT/STOP 指令执行期间, 看门狗定时器的操作被停止。在释放 HALT/STOP 模式后, 通过使用在 HALT/STOP 指令执行前 WDTM 设置的看门狗定时器操作时钟, 将使计数器再次启动。此时, 计数器没有被清零, 而是保持原值。	p.158 ☐
			采样时间和 A/D 转换时间	以上采样时间和转换时间不包括时钟频率误差。选择采样时间和转换时间时除了要满足注 2 和注 3 外, 还要将时钟频率误差考虑在内 (使用高速内部振荡器时, 最大误差范围为 $\pm 5\%$)。	p.164 ☐
			ADM: A/D 转换器模式寄存器	以上采样时间和转换时间不包括时钟频率误差。选择采样时间和转换时间时除了要满足注 3 和注 4 外, 还要将时钟频率误差考虑在内 (使用高速内部振荡器时, 最大误差范围为 $\pm 5\%$)。	p.169 ☐
				当 A/D 转换停止 (ADCS = 0) 时, 对 ADM 中 ADCS 之外的位进行操作, 然后启动 A/D 转换, 需要执行两条 NOP 指令或一条相当于 2 个机器周期的指令, 再设置 ADCS=1。	p.170 ☐
				在重写 FR0~FR2 时, A/D 转换必须停止 (ADCS = 0)。	p.170 ☐
				确保将第 6 位、第 2 位和第 1 位清 0。	p.170 ☐
			ADS: 模拟输入通道指定寄存器	确保 ADS 的第 2~7 位清 0。	p.170 ☐
			ADCR: 10 位 A/D 转换结果寄存器	当对 A/D 转换模式寄存器 (ADM) 和模拟输入通道选择寄存器 (ADS) 进行写操作时, ADCR 的内容可能不确定。要在 ADM 和 ADS 进行写操作前, 并且转换结束后, 读取转换结果。在其他时间读取, 可能读到不正确的转换结果。	p.170 ☐

章节	类别	功能	功能详解	注意事项	页码
第十章	软件	A/D 转换器	PMC2: 端口模式控制寄存器 2	当 PMC20 ~ PMC23 设置为 1 时, P20/ANI0~P23/ANI3 引脚不能用做端口引脚。将引脚设置为 A/D 转换器模式时, 确保上拉电阻选择寄存器 (PU20~PU23) 置为 0。	p.171 ☐
			A/D 转换器操作	必须确保步骤<1> 到 <4> 的操作时间为 1 μ s 或更长。	pp. ☐ 172, 176
				若将步骤<1>和<2>的次序颠倒, 也不会有问题。	pp. ☐ 172, 176
				<1> 可以被忽略。 但是, 在这种情况下中当步骤<4>执行后, 忽略第一次转换结果。	p.176 ☐
				步骤<5>到<8>所经历的时间与 ADM 的第 5 位到第 3 位 (FR2~FR0) 设置的转换时间不同。步骤<7>到<8>所经历的时间即为转换时间, 由 FR2~FR0 设置。	p.176 ☐
	硬件	STOP 模式的工作电流	为了满足 STOP 模式下的供电电流 DC 特性, 在执行 STOP 指令之前, 将 A/D 转换模式寄存器 (ADM) 的第 7 位 (ADCS) 和第 0 位 (ADCE) 清 0。	p.179 ☐	
		ANI0 ~ ANI3 的输入范围	观察 ANI0 至 ANI3 的输入电压的额定范围。如果输入到模拟输入通道的电压为 AVREF 或更高, 或者为 AVSS 或更低 (即使在额定的最大范围内), 则该通道的转换值是不确定的。此外, 其他通道的转换值也可能受影响。	p.179 ☐	
	软件	操作冲突	ADCR, ADCRH 读操作优先。在读操作执行完成后, 将新的转换结果写入 ADCR、ADCRH。	p.179 ☐	
			ADM 或 ADS 写操作优先。 ADCR, ADCRH 的写操作不执行, 也不会产生转换结束中断信号 (INTAD)。	p.179 ☐	
	硬件	噪声应对策略	为了保持 10 位分辨率, 必须注意输入到 AVREF 引脚和 ANI0~ANI3 引脚的噪声。 <1> 在电源上连接一个电容, 该电容应该是低阻抗和高频响应。 <2> 由于噪声的影响和模拟输入源的输出阻抗成正比, 因此建议外接一个电容, 如图 10-19 所示, 以降低噪声。 <3> 在转换期间, 不要将引脚 ANI0~ANI3 的 A/D 转换功能切换到它们的复用功能。 <4> 在转换开始后, 立即设置 HALT 模式, 可以提高转换精度。	p.179 ☐	
		ANI0/P20 ~ ANI3/P23	模拟输入引脚 (ANI0 ~ ANI3) 也可作为 I/O 端口引脚 (P20 ~ P23)。当选择 ANI0~ANI3 进行 A/D 转换时, 在转换过程中不要访问 P20~P23, 否则会降低转换精度。	p.180 ☐	
			如果有数字脉冲输入正在进行 A/D 转换的引脚的相邻引脚, 则由于噪声耦合, 有可能得不到预期的 A/D 转换值。因此, 在进行 A/D 转换时不要在相邻引脚输入脉冲信号。	p.180 ☐	

章节	类别	功能	功能详解	注意事项	页码
第十章	硬件	A/D 转换器	ANIO ~ ANI3 引脚的输入阻抗	该 A/D 转换器中，采样期间内对内部采样电容充电并采样。由于在采样期间之外，仅有漏电流流过，在采样期间还有电容的充电电流流过，因此在采样期间或其他时候输入阻抗都有波动。如果使用参考电压的最短转换时间，要进行充足的采样。建议模拟输入源的输出阻抗为 1kΩ 或更小，或者在 ANIO ~ ANI3 引脚连接 0.01μF 至 0.1μF 的电容（参见图 10-19）。	p.180 ☐
	软件		中断请求标志 (ADIF)	即使模拟输入通道选择寄存器 (ADS) 的值被修改，中断请求信号 (ADIF) 也不能清零。因此，如果在 A/D 转换期间改变模拟输入引脚，那么在重写 ADS 前，A/D 转换的结果和模拟输入改变前的 ADIF 可能被设置。所以需要注意，在 ADS 重写后立即读取 ADIF 时，尽管模拟输入改变后的 A/D 转换并未结束，也会设置 ADIF。在 A/D 转换停止后又重新开始转换时，要在重新开始转换前先将 ADIF 清 0。	p.180 ☐
			A/D 转换开始后的即时转换结果	A/D 转换开始后，如果在 ADCE 置 1 后 1μs 内对 ADCS 置 1，或在 ADCE 为零时对 ADCS 置 1，那么第一次 A/D 转换的值可能不在额定范围内。可以采取措，比如查询 A/D 转换结束中断请求 (INTAD)，并删除第 1 次转换结果。	p.181 ☐
			A/D 转换结果寄存器 (ADCR, ADCRH) 读操作	当对 A/D 转换器模式寄存器 (ADM) 和模拟输入通道选择寄存器 (ADS) 进行读取时，ADCR 和 ADCRH 的内容可能变得不确定。要在 ADM 和 ADS 进行写操作前，并且转换结束后，读取转换结果。若采用的时序和上述情况不同，可能会读到不正确的转换结果。	p.181 ☐
	硬件		转换等待模式的工作电流	STOP 模式下的 DC 特性中的工作电流不能满足，因为转换等待模式（仅有比较器在耗电），当 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 和第 0 位 (ADCE) 分别设置为 0 和 1 时。	p.181 ☐
第十一章	硬件	串行接口 UART6	UART 模式	TxD6 输出反向功能的操作仅在发送端进行，而不会在接收端进行。如要使用这个功能，接收端应准备好接收已反向的数据。	p.182 ☐
	软件			如果串行接口 UART6 使用的时钟未停止（例如，处于 HALT 模式下），则正常操作继续。如果时钟已停止（例如，处于 STOP 模式下），则每个寄存器都停止操作，并且保持时钟停止之前的数值。TxD6 引脚也保持时钟停止之前的数值，并将其输出。但是，在时钟恢复使用后，操作不能保证。因此，对电路进行复位使 POWER6 = 0, RXE6 = 0, TXE6 = 0。	p.182 ☐
				如果连续发送数据，从停止位到下一个起始位的通讯时序将延长两个该模块的操作时钟。但是，这不会影响到通信结果，因为接收端会在检测到起始位时对时序进行初始化。如果该接口用作 LIN 通讯模式，则不能使用连续发送功能。	p.182 ☐

(11/20)

章节	类别	功能	功能详解	注意事项	页码
第十一章	软件	串行接口 UART6	RXB: 接收缓冲寄存器 6	在将 RXE6 设置为 1, 经过基准时钟 (f _{CLK6}) 的一个时钟周期后, 进入接收允许状态。	p.188 ☐
			TXB6: 发送缓冲寄存器 6	当启动发送时, 在 TXE6 设置为 1, 并等待基准时钟 (f _{CLK6}) 一个或多个时钟周期后, 向 TXB6 写入发送数据。	p.188 ☐
				当异步串行接口发送状态寄存器 6 (ASIF6) 的第 1 位 (TXBF6) 为 1 时, 不要向 TXB6 写入数据。	p.188 ☐
				在通信过程中 (当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 和第 6 位 (TXE6) 均为 1, 或 ASIM6 的第 7 位 (POWER6) 和第 5 位 (RXE6) 均为 1 时), 不要通过软件刷新 (写入相同值) TXB6。在连续发送过程中输出相同的值时, 在向 TXB6 写入相同的值之前, 必须确认 TXBF6 为 0。	p.188 ☐
			ASIM6: 控制串行接口 UART6 的寄存器	启动时, 对 POWER6 置 1 之后, 对 TXE6 置 1, 启动发送操作, 然后再等待在基准时钟 (f _{CLK6}) 的一个或多个时钟周期之后, 设置发送数据到 TXB6。停止发送操作时, 先将 TXE6 清零, 然后将 POWER6 清零。	p.190 ☐
				启动时, 对 POWER6 置 1 之后, 然后对 RXE6 置 1, 经过基本时钟 (f _{CLK6}) 的一个时钟周期, 进入接收允许状态。停止接收操作时, 先将 RXE6 清零, 然后将 POWER6 清零。	p.191 ☐
				当 RxD6 引脚输入高电平的状态时, 设置 POWER6 = 1 → RXE6 = 1。如果在低电平输入期间设置 POWER6 = 1 → RXE6 = 1, 启动接收, 但不能接收到正确的数据。	p.191 ☐
				在对 PS61、PS60 和 CL6 位重写之前, 先将 TXE6 和 RXE6 位清零。	p.191 ☐
				当该接口用作 LIN 通讯操作模式时, PS61 与 PS60 恒为 0。	p.191 ☐
				在重写 SL6 位时, 必须确保 TXE6 = 0。当“停止位的个数 = 1”时, 始终执行接收操作, 因此设定 SL6 的值不会影响接收操作。	p.191 ☐
				当重写 ISRM6 时, 确保 RXE6 = 0。	p.191 ☐
			ASIS6: 异步串行接口接收错误状态寄存器 6	根据异步串行接口操作模式寄存器 6 (ASIM6) 的 PS61 和 PS60 位的内容的不同, 对 PE6 位的操作也有所不同。	p.191 ☐
				无论停止位的个数是多少, 只能对接收数据的第 1 位停止位进行校验。	p.191 ☐
				如果出现溢出错误, 则下一个接收数据不写入接收缓冲寄存器 6 (RXB6) 中, 而是被抛弃。	p.191 ☐
				读取接收缓冲寄存器 6 (RXB6) 之前, 必须读取 ASIS6。	pp. 191, 209 ☐

(12/20)

章节	类别	功能	功能详解	注意事项	页码
第十一章	软件	串行接口 UART6	ASIF6: 异步串行接口发送状态寄存器 6	若要连续发送数据, 将第一个发送的数据 (第一个字节) 写入 TXB6 寄存器中。检查 TXBF6 标志位, 必须为“0”。如果这样, 将下一个发送数据 (第二个字节) 写入 TXB6 中。如果在 TXBF6 标志为“1”时将数据写入 TXB6, 则无法保证发送数据的正确性。	p.192 ☐
				若要在连续发送结束时初始化发送单元, 则在产生发送完成中断后检查 TXSF6 标志位, 确保为“0”, 然后执行初始化操作。如果在 TXSF6 标志为“1”时执行初始化操作, 则无法保证发送数据的正确性。	p.192 ☐
			CKSR6: 时钟选择寄存器 6	当重写 TPS63 ~ TPS60 时, 必须确保 POWER6 = 0。	p.193 ☐
			BRGC6: 波特率发生器控制寄存器 6	在重写 MDL67 ~ MDL60 位时, 必须确保 ASIM6 的第 6 位 (TXE6) 和第 5 位 (RXE6) = 0。	p.194 ☐
	硬件	串行接口控制寄存器 6	ASICL6: 异步串行接口控制寄存器 6	波特率为 8 位计数器输出时钟的 1/2。	p.194 ☐
				如果数据 0 通过 SBRT6 和 SBTT6 写入到 ASICL6, 在通信操作期间 (当 ASIM6 的第 7 位 (POWER6) 和第 6 位 (TXE6) 为 1 或 ASIM6 的第 7 位 (POWER6) 和第 5 位 (RXE6) 为 1 时), ASICL6 可以通过软件更新 (写入相同的值)。	p.195 ☐
				在 SBF 接收错误情况下, 则再次返回到 SBF 接收模式下。SBRF6 标志位的状态保持不变 (1)。SBF 接收的详细情况, 请参见在稍后的 11.4.2 异步串行接口 (UART) 模式中的 (2) - (i) SBF 接收。	p.196 ☐
				在设置 SBRT6 位为 1 之前, 必须确保 ASIM6 的第 7 位 (POWER6) 和第 5 位 (RXE6) 为 1。并且, 在设置 SBRT6 为 1 之后, 不要在 SBF 接收结束 (产生中断请求信号) 之前, 对 SBRT6 位清零。	p.196 ☐
				SBRT6 位的读取值恒为 0。在 SBF 接收正确完成后, SBRT6 自动清零。	p.196 ☐
				在设置 SBTT6 位为 1 之前, 必须确保 ASIM6 的第 7 位 (POWER6) 和第 6 位 (TXE6) 为 1。并且, 在设置 SBTT6 为 1 之后, 不要在 SBF 发送结束 (产生中断请求信号) 之前, 对 SBTT6 位清零。	p.196 ☐
				SBTT6 位的读取值恒为 0。在 SBF 发送结束时 SBRT6 自动清零。	p.196 ☐
				在重写 DIR6 和 TxDLV6 位之前, 将 TXE6 和 RXE6 清零。	p.196 ☐
			ASIM6 的第 7, 6, 和 5 位 (POWER6, TXE6, 和 RXE6)	TXE6 和 RXE6 清零后, 再对 POWER6 清零, 可以设置操作停止模式。要启动操作, 设置 POWER6 为 1, 然后设置 TXE6 和 RXE6 为 1。	p.198 ☐

(13/20)

章节	类别	功能	功能详解	注意事项	页码
第十一章	软件	串行接口 UART6	UART 模式	在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。为避免产生非预期的起始位（下降信号），在将 P43 置 1 后，设置 PM43 为 0（输出）。	p.199 ☐
			校验类型和操作	当该接口用作 LIN 通讯操作模式时，PS61 与 PS60 恒为 0。	p.203 ☐
			连续发送	使用 TXBF6 标志的值来判断是否可以连续发送。不要写入下一个发送数据，判断的唯一标准就是 TXSF6 标志已经被置 1。	p.205 ☐
				在该接口用作 LIN 通讯操作时，不能使用连续发送功能。在将发送数据写入发送缓冲寄存器 6（TXB6）之前，异步串行接口发送状态寄存器（ASIF6）的值必须为 00H。	p.205 ☐
			连续发送期间的 TXBF6: ASIF6 的第 1 位	若要连续发送数据，将第一个发送的数据（第一个字节）写入 TXB6 寄存器中。检查 TXBF6 标志位，必须为“0”。如果这样，将下一个发送数据（第二个字节）写入 TXB6 中。如果在 TXBF6 标志为“1”时将数据写入 TXB6，则无法保证发送数据的正确性。	p.205 ☐
			连续发送期间的 TXSF6: ASIF6 的第 0 位	若要在连续发送结束时初始化发送单元，则在产生发送完成中断后检查 TXSF6 标志位，确保为“0”，然后执行初始化操作。如果在 TXSF6 标志为“1”时执行初始化操作，则无法保证发送数据的正确性。	p.205 ☐
			正常接收	即使出现接收错误，也必须读取接收缓冲寄存器 6（RXB6）。否则，当接收到下一个数据时会产生溢出错误，而且接收错误状态保持不变。	p.209 ☐
				接收始终按照“停止位的个数 = 1 来执行。第 2 个停止位被忽略。	p.209 ☐
				在读取 RXB6 之前，必须先读取异步串行接口接收错误状态寄存器 6（ASIS6）。	p.209 ☐
			串行时钟的产生	在接收端传送期间保持波特率误差在允许的误差范围内。	p.215 ☐
				接收期间的波特率误差必须满足（4）接收期间允许的波特率范围 中所描述的范围。	p.215 ☐
			接收期间允许的波特率范围	必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。	p.217 ☐
第十二章	软件	乘法器	MUL0: 16 位乘法结果存储寄存器 0	这个寄存器既能用 16 位存储器操作指令操作，也能用 8 位存储器操作指令操作。但是，使用 8 位存储器操作指令对其进行操作时，要通过直接寻址方式对其进行访问。	p.220 ☐
			MULC0: 控制寄存器 0	确保第 1~7 位清 0。	p.222 ☐

章节	类别	功能	功能详解	注意事项	页码
第十三章	硬	中断功能	向量表地址	向量表地址 0014H 不对应任何中断源。	p.226 ☐
	软件		IF0, IF1: 中断请求标志寄存器0,1, MK0, MK1: 中断屏蔽标志寄存器 0, 1	因为 P30, P31, P41, 和 P43 都有作为外部中断输入的复用功能, 所以通过指定端口功能的输出模式来改变输出电平, 中断请求标志置 1。因此, 在使用输出模式之前, 应先将中断屏蔽标志置 1。	pp. 229, 230 ☐
			INTM0: 外部中断模式寄存器 0	确保第 0 位和第 1 位清零。	p.231 ☐
				在设置 INTM0 寄存器之前, 必须设置相应的中断屏蔽标志 (××MK× = 1) 以禁止中断。INTM0 寄存器设置之后, 首先对中断请求标志 (××IF× = 0) 清零, 然后再对中断屏蔽标志 (××MK× = 0) 清零, 以允许中断。	p.231 ☐
			INTM1: 外部中断模式寄存器 1	第 2 位至第 7 位必须清零。	p.232 ☐
				设置 INTM1 之前, 先设定 PMK3 为 1, 以禁止中断。要允许中断, PIF3 清 0, 然后对 PMK3 清 0。	p.232 ☐
			中断请求挂起	当中断请求标志寄存器 0, 1 (IF0, IF1) 或中断屏蔽标志寄存器 0, 1 (MK0, MK1) 被访问期间, 中断请求被挂起。	p.235 ☐
第十四章			多重中断服务	即使是低优先级中断, 多重中断也能响应。	p.236 ☐
	软件	待机功能	—	仅当选项字节选择低速内部振荡器“可以通过软件停止”时, LSRSTOP 设置才有效。	p.238 ☐
	硬件		STOP 模式	当切换到 STOP 模式时, 一定要在执行 STOP 指令之前停止外部硬件操作 (除了以低速内部振荡时钟工作的外部硬件之外)。	p.239 ☐
			STOP 模式, HALT 模式	当使用待机功能时, 推荐按以下步骤减少 A/D 转换器的工作电流。首先对 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 和第 0 位 (ADCE) 清零, 以停止 A/D 转换操作, 然后执行 HALT 或 STOP 指令。	p.239 ☐
			STOP 模式	如果在置 STOP 模式之前, 低速内部振荡器正在工作, 则在 STOP 模式中不能停止低速内部振荡时钟 (参见表 14-1)。	p.239 ☐
	软件		OSTS: 振荡稳定时间选择寄存器	设置和释放 STOP 模式时, 按照下面的方法设置振荡稳定时间。预期的振荡器振荡稳定时间 ≤ 通过 OSTS 设置的振荡稳定时间。	p.240 ☐
	硬件			无论 STOP 模式是被复位信号打断还是被中断唤醒, STOP 模式释放后的等待时间不包括从释放 STOP 模式到时钟开始振荡的时间 (下图中的“a”部分)。	p.240 ☐
	软件			上电或复位释放后的振荡稳定时间, 可以通过选项字节来选择。详细情况, 请参见 第十八章 选项字节。	p.240 ☐

章节	类别	功能	功能详解	注意事项	页码
第十四章	软件	待机功能	HALT 模式设置和操作状态	由于中断请求信号可用来退出待机模式，如果某一中断源中断请求标志置位且中断屏蔽标志清零，则立即退出待机模式。	p.241 ☐
			STOP 模式设置和操作状态	因为使用中断请求信号来释放待机模式，如果某一中断源中断请求标志置位且中断屏蔽标志清零，则立即退出待机模式。因此，在 STOP 模式中，执行 STOP 指令后恢复正常操作模式，然后操作会停止 34 μ s（典型值）（在使用晶体/陶瓷振荡器时，在经过由振荡稳定时间选择寄存器（OSTS）设置的一段额外等待振荡稳定时间后）。	p.244 ☐
第十五章	硬件	复位功能	—	对于外部复位，输入 引脚的低电平的时间应为 2 μ s 或更长。	p.248 ☐
				在复位信号产生期间，系统时钟和低速内部振荡时钟停止振荡。	p.248 ☐
				当 RESET 引脚用作只能输入的端口引脚（P34）时，由 POC 电路、LVI 电路和看门狗定时器释放复位后，且再次参考选项字节之前，如果有低电平输入到 RESET 引脚，78K0S/KB1+复位。保持复位状态直到高电平输入到 RESET 引脚。	p.248 ☐
				LVI 电路的内部复位信号不会使 LVI 电路复位。	p.249 ☐
			由看门狗定时器溢出引起的复位时序	在看门狗定时器内部复位的情况下，看门狗定时器也复位。	p.251 ☐
			RESF：复位控制标志寄存器	不能通过 1 位存储器操作指令读取数据。	p.255 ☐
第十六章	软件	上电清零电路	上电清零电路的功能	如果在 POC 电路中产生内部复位信号，则将复位控制标志寄存器（RESF）清零。	p.256 ☐
	硬件			这些产品应该在如下电压范围内使用，因为 POC 电路检测电压（V _{POC} ）决定供电电压范围。 标准产品，(A) 等级产品：2.2 ~ 5.5 V，((A)等级产品) 2.26 ~ 5.5 V	p.256 ☐
	软件	上电清零电路的注意事项		对于一个系统，如果电源电压（V _{DD} ）某段时间内在 POC 检测电压（V _{POC} ）附近波动，该系统可能会反复进行复位和复位状态的释放。在这种情况下，可采用下列方法任意设置从复位释放到微控制器的启动操作所经历的时间。	p.258 ☐
第十七章	软件	低电压检测器	LVIM：低电压检测寄存器	要停止 LVI，可执行以下任一过程： • 当使用 8 位存储器操作指令时：写 00H 到 LVIM • 当使用 1 位存储器操作指令时：对 LVION 清零	p.261 ☐
				确保第 2~6 位清零。	p.261 ☐
		低电压检测电平选择寄存器	LVIS：低电压检测电平选择寄存器	确保第 4~7 位清零。	p.262 ☐
				在 LVI 工作期间，如果写入不同的值，该值在写入的时刻将变得不确定，因此确保在写入前停止 LVI（LVIM 寄存器的第 7 位（LVION）= 0）。	p.262 ☐

章节	类别	功能	功能详解	注意事项	页码
第十七章	软件	低电压检测器	用作复位时	步骤<1> 必须执行。当 LVIMK = 0 时，在步骤<3>之后，可能会立即产生中断。 当 LVIM 被设置为 1 时，如果电源电压 (VDD) ≥ 检测电压 (VLVI)，则不会产生内部复位信号。	p.263 ☐
			低电压检测器的注意事项	在系统中，如果电源电压 (VDD) 某段时间内在 LVI 检测电压 (VLVI) 附近波动，可根据低电压检测器的使用情况进行如下操作。 <1> 用作复位 系统可能会反复进行复位和复位状态的释放。 在这种情况下，采用下面的动作 (1) 来任意设置从复位释放到微控制器启动操作所需要的时间。 <2> 用作中断 可能会频繁产生中断请求。采用动作 (2) 的 (b)。	p.266 ☐
第十八章	硬件	选项字节	上电或复位释放后振荡稳定时间	仅当选择晶体/陶瓷振荡器时钟作为系统时钟源时，该选项的设置才有效。如果选用高速内部振荡器时钟或外部时钟输入作为系统时钟源，无需等待时间。	p.270 ☐
			复位引脚的控制	因为在复位释放之后，才能参考选项字节。如果在参考选项字节之前，低电平输入到 RESET 引脚，这样的复位状态不会被释放。 并且，当 RMCE 设置为 0 时，连接上拉电阻。	p.270 ☐
			系统时钟源的选择	因为 X1 和 X2 引脚也可以作为 P121 和 P122 引脚使用，根据系统时钟源的选择，X1 和 X2 引脚可以使用的条件不同。 (1) 选择晶体/陶瓷振荡器时钟 X1 和 X2 引脚不能作为 I/O 端口使用，因为它们被作为时钟输入端口。 (2) 选择外部时钟输入 因为 X1 引脚作为外部时钟输入，P121 口不能作为 I/O 端口引脚使用。 (3) 选择高速内部振荡器时钟 P121 和 P122 端口可作为 I/O 端口引脚。	p.270 ☐
			低速内部振荡	如果选择低速内部振荡器时钟不能停止，看门狗定时器(WDT)的计数时钟就会固定为低速内部振荡器时钟。	p.271 ☐
				如果选择低速内部振荡器时钟可以通过软件停止，WDT 的计数时钟在 HALT/STOP 模式下被停止，不管低速内部振荡模式寄存器(LSRCM)的第 0 位 (LSRSTOP)如何设置。同样地，当 WDT 的计数时钟选择低速内部振荡器时钟之外的时钟信号时，时钟提供也被停止。 当低速内部振荡器工作(LSRSTOP = 0)时，即使在 STOP 模式下，时钟也可提供给 8 位定时器 H1。	p.271 ☐

章节	类别	功能	功能详解	注意事项	页码
第十八章	硬件	选项字节	当RESET引脚仅用作输入端口引脚（P34）时的注意事项	对于已经通过选项字节功能被设置为“RESET引脚用作只能输入的端口引脚（P34）”并完成写入的设备，当重新擦除/重新写入时(使用专用 flash 存储器编程器在板编程)，要留意如下内容。 给目标系统上电前，连接专用 flash 存储器编程器并打开电源。如果目标系统已经提前上电，flash 存储器编程模式无法切换。	p.271 ☐
第十九章	软件	Flash 存储器	PG-FP5 GUI 软件设置值举例	以上值为推荐值。由于环境不同，这些值可能改变，所以设置之前请先充分评估。	p.278 ☐
			安全设置	在一次性擦除（芯片擦除）的安全设定后，擦除指令不能执行。另外，即使一个写入命令被执行，与已存在的数据不同的部分是不能被写入的，因为擦除命令已不能执行。	p.281 ☐
			自编程功能	在执行自编程以前，自编程处理程序必须先被包含在用户程序中。	p.282 ☐
				当自编程命令执行时，其它命令都不能执行。因此，要先清除和重启看门狗时钟计数器，以便在自编程时计数器不会溢出。关于执行自编程需要的时间请参考表 19-10。	p.285 ☐
				在自编程期间发生的中断，在自编程模式结束后才能被响应。为了避免这种情况，应在从正常模式切换到自编程模式之前以一定顺序终止中断服务（通过将 MK0 和 MK1 设定为 0FFH，并执行 DI 指令）	p.285 ☐
				当执行自编程命令时，不使用 RAM。	p.285 ☐
				当 FLASH 存储器正在数据写入或擦除过程中，如果供电电压下降或者有复位信号输入，则此次写入或擦除操作不一定成功。	p.285 ☐
				在块擦除中空白数据的值设为 0FFH。	p.285 ☐
				预先设置 CPU 时钟以使其在自编程期间至少为 1 MHz。	p.285 ☐
				执行指定序列到自编程模式后，在执行 NOP 和 HALT 指令后立即执行自编程。此时，HALT 指令在 10 μ s（最大值）+ 2 个 CPU 时钟（f _{cpu} ）后自动释放。	p.285 ☐
				如果选择晶振或外部时钟作为系统时钟，执行指定序列到自编程模式后执行 NOP 和 HALT 指令，释放 HALT 状态后等待 8 μ s，然后执行自编程。	p.285 ☐
				使用 1 位存储器操作指令检查 FPRERR。	p.285 ☐
				自编程模式下和 HALT 模式下引脚的状态是相同的。	p.285 ☐

章节	类别	功能	功能详解	注意事项	页码
第十九章	软件	Flash 存储器	自编程功能	在自编程模式中，禁止通过 on-board/off-board 编程设置安全功能，无论安全功能如何设定，自编程命令都可执行。想要在自编程期间禁止写入或擦除进程，则需要设定保护字节。	p.285 ☐
				执行自编程命令之前，要确保 flash 地址指针 H (FLAPH) 的第 5 位 ~ 第 7 位和 flash 地址指针比较寄存器 (FLAPHC) 清零。如果在这些位为 1 时执行自编程命令，则设备可能出现故障。	p.285 ☐
				在即将设置自编程模式和正常模式之前，将 FLCMD 寄存器的值清为 00H。	p.285 ☐
			FLPMC: Flash 编程模式控制寄存器	设置自编程模式的情况下的注意事项，请参考 19.8.2 自编程功能的注意事项。	p.286 ☐
				预先设置 CPU 时钟，使其在自编程期间至少为 1 MHz。	p.286 ☐
				执行指定序列到自编程模式之后，立即执行 NOP 和 HALT 指令，然后再执行自编程。此时，HALT 指令在 10 μ s (最大值) + 2 个 CPU 时钟 (f _{cpu}) 后自动释放。	p.286 ☐
				如果选择晶振或外部时钟作为系统时钟，执行指定序列到自编程模式后执行 NOP 和 HALT 指令，释放 HALT 状态后等待 8 μ s，然后执行自编程。	p.286 ☐
				在即将设置自编程模式和正常模式之前，将 FLCMD 寄存器的值清为 00H。	p.286 ☐
			PFCMD: Flash 保护命令寄存器	在自编程模式下中断服务不能执行。在执行设置自编程模式的指定序列前和执行切换正常模式的指定序列后禁止中断服务 (当 MK0 和 MK1 = FFH 时通过执行 DI 指令)。	p.287 ☐
			PFS: Flash 状态寄存器	使用 1 位存储器操作指令检查 FPRERR。	p.287 ☐
			FLAPH, FLAPL: Flash 地址指针 H 和 L	执行自编程命令之前，要确保 flash 地址指针 H (FLAPH) 的第 5 位 ~ 第 7 位和 flash 地址指针比较寄存器 (FLAPHC) 清零。如果在这些位为 1 时执行自编程命令，则设备可能出现故障。	p.290 ☐
			FLAPHC, FLAPLC: Flash 地址指针 H/L 比较寄存器	执行自编程命令之前，要确保 flash 地址指针 H (FLAPH) 的第 5 位 ~ 第 7 位和 flash 地址指针比较寄存器 (FLAPHC) 清零。如果在这些位为 1 时执行自编程命令，则设备可能出现故障。	p.290 ☐
				当进行块擦除、校验、空白检测时，将块序号 (与 FLAPH 的值相同) 赋值给 FLAPHC。	p.290 ☐
				当进行块擦除时，FLAPLC 要清零，当块空白检测时，FLAPLC 要设为 0FFH。	p.290 ☐

章节	类别	功能	功能详解	注意事项	页码
第十九章	软件	Flash 存储器	切换到自编程模式	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。	pp. ☐ 293,294, 296, 297
			切换到正常模式		
			字节写入操作	如果写入失败，应该先擦除块内容并再次写入。	p.305 ☐
第二十章	硬件	片上调试功能	将 QB-MINI2 连接到 78K0S/KB1+	在用于大批量生产时，不要使用产品的片上调试功能，因为使用该功能的话，可能会超过 Flash 存储器的保证重写次数，这样的产品可靠性无法保障。使用片上调试功能时，NEC 电子不接受本产品的相关投诉。	p.330 ☐
				在电路连接示例中描述的常数值都是参考值。如果打算进行大批量生产的 flash 编程，请认真评估目标设备的规格是否能够满足要求。	p.330 ☐
				如果不使用 QB-MINI2 而是通过实机运行来调试，使用 QB-Programmer 写入用户程序。通过调试器下载的程序包括监控程序，如果不通过 QB-MINI2 控制，该程序会有故障。	p.332 ☐
第二十二章	硬件	电气特性 (标准产品, (A) 等级产品)	最大额定值	任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因此，必须保证产品在不超过最大额定值的条件下使用。	p.344 ☐
			X1 振荡器	在使用 X1 振荡器时，上图中虚线框部分的连线应遵从如下的连接方式，避免线间电容的负面影响。 ● 保证连线要尽可能的短。 ● 不要使其它的信号线与振荡器导线互相交叉。 ● 不要在通过大电流的信号线旁边进行振荡器的布线。 ● 保持振荡器电容的地和 Vss 同电位。 ● 不要将电容的接地点连接在大电流通过的地模式。 ● 不要从振荡器上取信号。	p.345 ☐
			A/D 转换器	当模拟输入引脚用作复用 I/O 端口，或在 A/D 转换期间端口发生改变时，转换精度可能降低。	p.352 ☐

章节	类别	功能	功能详解	注意事项	页码
第二十三章	硬件	电气特性 (目标值) (A2) 等级产品)	—	这些特性显示的都是目标值，在设备评测后可能会改变。操作电压范围同样也可能改变。	p.356 ☐
			最大额定值	任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因此，必须保证产品在不超过最大额定值的条件下使用。	p.356 ☐
			X1 振荡器	在使用 X1 振荡器时，上图中虚线框部分的连线应遵从如下的连接方式，避免线间电容的负面影响。 • 保证连线要尽可能的短。 • 不要使其它的信号线与振荡器导线互相交叉。 • 不要在通过大电流的信号线旁边进行振荡器的布线。 • 保持振荡器电容的地和 Vss 同电位。 • 不要将电容的接地点连接在大电流通过的地模式。 • 不要从振荡器上取信号。	p.358 ☐
			A/D 转换器	当模拟输入引脚用作复用 I/O 端口，或在 A/D 转换期间端口发生改变时，转换精度可能降低。	p.365 ☐
第二十五章	硬件	推荐焊接条件	无铅产品	产品的器件编号尾缀带有 -A 或 -AX，是无铅产品。	p.373 ☐
			—	如果不使用以下的推荐焊接方法和条件，请与 NEC 电子公司销售代理联系。	p.373 ☐
				不要一起使用不同的焊接方法（局部加热法除外）。	pp. ☐ 373, 374
				在进行波峰焊时，只对过孔插装元件（THD）的引脚加热。确保波峰焊料不要接触封装。	p.374 ☐

附录 E 修订历史

E.1 该版本的主要修订信息

页码	说明
全部	删除 μ PD78F9232MC-5A4, 78F9234MC-5A4, 78F9232MC(A)-5A4, 78F9234MC(A)-5A4, 78F9232MC(A2)-5A4, 78F9234MC(A2)-5A4
p. 6	修改 开发软件工具相关文档 (用户手册)
p. 18	修改 1.4 78K0S/Kx1+产品线
pp. 38 至 41	修改 表 3-3 特殊功能寄存器
p. 274	19.4 使用 FLASH 编程器的写入 - 删除 FlashPro4, 增加 QB-MINI2 - 修改 备注
p. 275	修改增加备注到 图 19-2 将程序写入 flash 存储器的环境 (使用 FlashPro5/QB-MINI2)
p. 276	修改表 19-2 78K0S/KB1+ 和 FlashPro5/QB-MINI2 之间的连线
p. 276	修改增加备注到 图 19-2 和 FlashPro5/QB-MINI2 的连线图 (30 引脚产品)
p. 280	修改图 19-7 通讯命令
pp. 299 至 301	19.8.6 自编程模式下块擦除操作示例 - 修改说明, 增加注 2 - 修改 图 19-20 自编程模式下块擦除操作流程 - 修改样例程序
pp. 320 至 322	19.8.11 自编程模式下最短中断禁止时间的操作过程 - 修改 图 19-27 自编程模式下最短中断禁止时间的操作流程 (从擦除到空白块检验) - 修改样例程序
p. 330	修改 20.1 连接 QB-MINI2 到 78K0S/KB1+的注意事项
p. 356	第 23 章 电气特性 ((A2)等级产品) 目标规格 → 规格
p. 375	修改 附录 A 开发工具

E.2 以往版本的修订历史

下表显示了该版本的修订历史，“适用于”栏指出相关修订版本的章节。

(1/6)

版本	说明	适用于
第二版	在 1.3 订购信息中增加部分产品型号	第一章 概述
	修改 1.5 78K0S/Kx1+ 产品系列的操作温度范围	
	增加 2.1 引脚功能列表中 P34/RESET, P121/X1, 和 P122/X2 的注	第二章 引脚功能
	增加 2.2.3 P30~P34 (端口 3) 描述	
	增加 2.2.7 RESET 描述	
	增加和修改 4.2.3 端口 3 中的注意事项	第四章 端口功能
	在 6.1 16 位定时器/事件计数器 00 的功能中修改(2) 外部事件计数器的描述	第六章 16 位定时器/事件计数器 00
	修改 图 6-2. 16 位定时器计数器 00 (TM00)的格式	
	修改图 6-5. 16 位定时器模式控制寄存器 00 (TMC00)的格式中的注意事项 2	
	增加 6.4.3 脉冲宽度测量操作中的(1), (2), (3), 和(4)的注意事项	
	修改图 6-19. 通过自由运行计数器测量脉冲宽度的配置框图	
	修改图 6-20. 使用自由运行计数器和一个捕获寄存器进行脉冲宽度测量的时序（指定上升沿和下降沿）和注	
	修改图 6-22. 使用自由运行计数器进行脉冲宽度测量的时序（指定双沿）和注	
	修改图 6-24. 使用自由运行计数器和两个捕获寄存器进行脉冲宽度测量的时序（指定上升沿有效）和注	
	修改 6.5 有关 16 位定时器/事件计数器 00 的注意事项中的 (2) 16 位定时器计数器 00 (TM00) 操作的<3>和<4>	
	修改 7.2 8 位定时器 80 的配置中的(1)8 位比较寄存器 80 (CR80) 的注意事项	第七章 8 位定时器 80
	修改 8.2 8 位定时器 H1 的配置的(2)8 位定时器 H 比较寄存器 11 (CMP11)	第八章 8 位定时器 H1
	修改 8.4.2 用作 PWM 输出模式的操作中的 注意事项 1	
	修改图 8-9.PWM 输出模式下的时序操作中的(e) 改变 CMP11 的操作(CMP11 = 02H → 03H, CMP01 = A5H)	
	增加图 9-2. 看门狗定时器模式寄存器的格式 (WDTM)中的注意事项 2	第九章 看门狗定时器
	修改表 10-1. 采样时间和 A/D 转换时间和 注 1	第十章 A/D 转换器
	修改图 10-3. A/D 转换器模式寄存器 (ADM) 的格式和 注 2	
	增加 11.3 控制串行接口 UART6 的寄存器中(7) 输入切换控制寄存器 (ISC)的描述	第十一章 串行接口 UART6
	修改表 11-4. 波特率发生器的数据设置中的值	

(2/6)

版本	说明	适用于
第二版	修改 13.4.2 多重中断服务	第十三章 中断功能
	增加 图 13-10 中断嵌套示例(1/2) 中 示例 1 的注意事项	
	增加 图 13-10 中断嵌套示例(2/2) 中 示例 3	
	修改 图 14-3. 由复位信号释放 HALT 模式中的复位信号	第十四章 待机功能
	修改 并增加 14.2.2 STOP 模式的(2)中的(a)由未屏蔽的中断请求释放的注	
	修改图 14-6. 由复位信号释放 STOP 模式中的复位信号	
	修改图 15-1 复位功能框图	第十五章 复位功能
	修改图 16-3.复位释放后软件处理过程示例(1/2)	第十六章 上电清零电路
	修改 图 17-1 低电压检测器的框图	第十七章 低电压检测器
	修改图 17-5. 低电压检测器中断信号产生时序中的 INTLVI 和注	
	修改 17.5 低电压检测器的注意事项的 <动作>中的(2)	
	修改图 17-6.复位释放后软件处理过程示例(1/2)	
	修改第十八章 选项字节的描述和结构	第十八章 选项字节
	修改和增加 19.1 特性	第十九章 FLASH 存储器
	将图 19-2 写入 FLASH 存储器的环境分为两个图, FlashPro4 和 PG-FPL2	
	删除 19.7.1 Flash 存储器编程模式	
	修改并增加 19.8.2 自编程功能的注意事项	
	增加 19.8.3 自编程功能所使用的寄存器(3) 中的 3. WEPRERR 标志的操作条件中的 <设置条件>	
	增加图 19-15 Flash 编程命令寄存器(FLCMD)的格式中的描述	
	完全修改第二十一章 电气特性 (目标值)	第二十一章 电气特性 (目标值)
	增加第二十五章 封装标记信息 和 第二十六章 推荐焊接条件	第二十五章 封装标记信息
		第二十六章 推荐焊接条件
	增加 A.1 软件包 包括软件工具	附录 A 开发工具
	增加附录 D 注意事项列表	附录 D 注意事项列表

版本	说明	适用于
第三版	删除(T) 产品, (S) 产品, (R) 产品, (T2) 产品的描述	通篇
	修改 1.4 78K0S/Kx1+ 产品系列	第一章 概述
	增加 3.2.1 (3) 堆栈指针 (SP) 中的注意事项 2	第三章 CPU 结构
	修改图 4-7 P43 的框图	第四章 端口功能
	增加图 4-17 端口模式控制寄存器 2 的格式中的注意事项	
	修改图 6.2(1) . 16 位定时器计数器 00 (TM00)的格式中的注意事项 2	第六章 16 位定时器/事件计数器 00
	增加 6.5 (23) 外部时钟限制	
	修改图 10-8 端口模式控制寄存器 2 的格式 (PMC2) 中的注意事项	第十章 A/D 转换器
	增加 10.6 (10) 转换等待模式的工作电流	
	增加 11.2 (1) 接收缓冲寄存器 6 (RXB6) 中的注意事项	第十一章 串行接口 UART6
	增加 11.2 (3) 发送缓冲寄存器 6 (TXB6) 中的注意事项 1, 并且修改 注意事项 3	
	修改图 11-5 异步串行接口操作模式寄存器 6 (ASIM6) (1/2) 中的注 3	
	增加图 11-5 异步串行接口操作模式寄存器 6 (ASIM6) (2/2) 中的注 1 和 2, 修改注意事项 1, 2 和 3	
	修改 11.3 (6) 异步串行接口控制寄存器 6 (ASICL6) 中的注意事项	
	修改图 11-10 异步串行接口控制寄存器 6 的格式 (ASICL6) 中的注意事项 1	
	修改 11.4.2 (1) 寄存器使用中的注意事项	
	部分修改 11.4.2 (2) (c) 正常发送的描述	
	部分修改 11.4.2 (2) (d) 连续发送中的描述并增加注意事项 1	
	增加 17.3 (2) 低电压检测电平选择寄存器(LVIS) 中的注意事项 2	第十七章 低电压检测器
	增加 18.3 当 RESET 引脚仅用作输入端口引脚(P34)时的注意事项	第十八章 选项字节
	增加 19.6.1 X1 和 X2 引脚的描述	第十九章 FLASH 存储器
	增加 19.8 Flash 存储器由自写入编程的备注 1	
	修改并增加表 19-11 自编程控制命令内部校验 1 的描述, 增加内部校验 2 的备注	
	部分修改并增加 19.8.2 自编程功能的注意事项	
	增加图 19-12. Flash 编程模式控制寄存器(FLPMC)的格式的的注意事项 2, 3 和 5 并修改注意事项 4	
	修改 19.8.3 (2) Flash 保护命令寄存器 (PFCMD) 中的注意事项, 并增加 FPRERR 的描述	
	增加 19.8.3 (3) Flash 状态寄存器(PFS) 中的注意事项	

(4/6)

版本	说明	适用于
第三版	修改图 19-15. Flash 编程命令寄存器 (FLCMD)的格式中的内部校验 1 的注, 并增加内部校验 2 的描述	第十九章 FLASH 存储器
	修改图 19-16. Flash 地址指针 H/L (FLAPH/FLAPL)格式中的注意事项和 Flash 地址指针 H/L 比较寄存器 (FLAPHC/FLAPLC)的格式中的注意事项 1	
	增加 19.8.4 从正常模式切换到自编程模式的示例和 19.8.5 从自编程模式切换到正常模式的示例的描述	
	增加 19.8.9 自编程模式下内部校验操作示例中的内部校验 1 和 2 的描述	
	增加 19.8.10 自编程模式下最短命令执行时间的操作示例和 19.8.11 自编程模式下中断屏蔽时最短命令执行时间的操作过程的描述	
	- 修改低电平输入电压 (V_{IL3})的最大值 - 修改高电平输入电压 (V_{OH2})的条件 - 修改高电平输入漏电流, 低电平输入漏电流, 高电平输出漏电流和低电平输出漏电流的最大值 - 修改 STOP 模式下供电电流(I_{DD5})的最大值 - 在 AC 特性中增加 CPU 时钟和外围时钟频率的设置范围 - 修改 A/D 转换器特性中的注意事项	第二十一章 电气特性(标准产品, (A) 等级产品)
	- 修改低电平输入电压 (V_{IL3})的最大值 - 修改高电平输入电压 (V_{OH2})的条件 - 增加内部上拉电阻的功耗计算公式 - 在 AC 特性中增加 CPU 时钟和外围时钟频率的设置范围 - 修改 A/D 转换器特性中的注意事项	第二十二章 电气特性 (目标值) ((A2) 等级产品)
	删除 第二十五章 封装标注信息	第二十五章 封装标记信息
	修改 A.4 Flash 存储器写入工具	附录 A 开发工具
	增加 A.5.1 当使用在线仿真器 QB-78K0SKX1 (开发中) 和 A.5.2 当使用在线仿真器 QB-MINI2	

(5/6)

版本	说明	适用于
第四版	- 增加 SDIP 封装产品 (μPD78F9232CS-CAA-A, 78F9234CS-CAA-A) - 增加(μPD78F9232MC(A)-CAB-AX, 78F9234MC(A)-CAB-AX, 78F9232MC(A2)-CAB-AX, 78F9234MC(A2)-CAB-AX) 产品的 SSOP 封装	通篇
	修改 1.1 特征	第一章 概述
	修改 1.2 订购信息	
	1.3 引脚配置 (俯视图)	
	修改 AV_{REF} 的描述	
	修改 1.4 78K0S/Kx1+ 产品系列	
	修改 1.6 功能概述	第二章 引脚功能
	2.1 引脚功能列表, 2.2.9 AV _{REF}	
	修改 AV_{REF} 功能的描述	第十章 A/D 转换器
	- 在表 10-1 采样时间和 A/D 转换时间中添加注 4 - 在图 10-3 A/D 转换器模式寄存器(ADM)的格式中添加注 5	
	修改注意事项 3	第十五章 复位功能
	19.4 使用 FLASH 编程器的写入	第十九章 FLASH 存储器
	- 修改专用 flash 存储器编程器 (增加 FlashPro5 , 并删除 PG-FPL2) - 修改备注	
	19.5 编程环境	
	- 在专用 flash 存储器编程器中增加 FlashPro5 和 QB-MINI2 - 增加 FlashPro5/QB-MINI2 的连接描述 - 从专用 flash 存储器编程器中删除 PG-FPL2	
	修改图 19-5 PG-FP5 图形界面软件设置举例	
	图 19-7. 通讯命令	
	- 在专用 flash 存储器编程器中增加 FlashPro5 和 QB-MINI2 - 从专用 flash 存储器编程器中删除 PG-FPL2	第二十章 片上调试功能
	在表 19-10 自编程控制命令中增加注	
	增加此章	第二十二章 电气特性(标准产品, (A) 等级产品)
	- 修改 X1 振荡器特性的数值 - 在 AC 特性中增加 CPU 时钟和外围时钟频率的设置范围	
	- 修改 X1 振荡器特性的数值 - 在 AC 特性中增加 CPU 时钟和外围时钟频率的设置范围	第二十三章 电气特性 (目标值) ((A2) 等级产品)
	增加 新添产品的封装图	第二十四章 封装图
	增加 新添产品的推荐焊接条件	第二十五章 推荐焊接条件

(6/6)

版本	说明	适用于
第四版	修改 图 A-1 开发工具	附录 A 开发工具
	A.4 Flash 存储器写入工具 • 增加 FlashPro5 和 QB-MINI2 • 删除 PG-FPL2	
	修改 A.5.1 当使用在线仿真器 QB-78K0SKX1 时	
	删除 A.5.3 当使用在线仿真器 IE-78K0S-NS 或 IE-78K0S-NS-A 时 和 A.5.4 当使用在线仿真器 QB-78K0SKX1MINI 时	
	修改 A.6 调试工具（软件）	
	修改此章	附录 B 目标系统设计注意事项

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.renesas.com/>（中文）

<http://www.renesas.com/>（英文）

[北京]

瑞萨电子（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]

瑞萨电子（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]

瑞萨电子（中国）有限公司上海分公司

中国上海市浦东新区陆家嘴环路 1233 号
汇亚大厦 205 室

电话：（+86）21-5877-1818

传真：（+86）21-6887-6100

[香港]

香港瑞萨电子有限公司

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

瑞萨电子（上海）有限公司

中国上海市浦东新区陆家嘴环路 1233 号
汇亚大厦 205 室

电话：（+86）21-5877-1818

传真：（+86）21-6887-6100

[成都]

瑞萨电子（中国）有限公司成都分公司

四川省成都市二环路南三段 15 号
天华大厦 608 室

电话：(+86)28-8512-5224

传真：(+86)28-8512-5334

[长春]

瑞萨电子（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]

瑞萨电子（中国）有限公司大连分公司

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

[青岛]

瑞萨电子（中国）有限公司青岛分公司

中国山东青岛市宁夏路 288 号 G3 楼 607 室
电话：(+86)532-8872-7900/8872-7901
传真：(+86)532-8872-7902