



用户手册

78K0S/KU1+

8 位单片微控制器

μ PD78F9200

μ PD78F9201

μ PD78F9202

文档编号: U18172CA1V0UD00 (第一版)
发行日期: 2007 年 6 月 NS CP(K)

© ~~NEC Electronics Corporation~~ 2006
日本印制

[备忘录]

CMOS 设备注意事项

① 输入引脚处的电压波形

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

② 未使用的输入引脚的处理

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

③ ESD防护措施

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

④ 初始化之前的状态

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

⑤ 电源开关顺序

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

⑥ 电源关闭状态下的输入信号

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

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

PC/AT是International Business Machines Corporation的商标。

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

SPARCstation是SPARC International, Inc.的商标。

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

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

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

- 本档信息先于产品的生产周期发布。在最终生产前，产品参数可能会有变化，或者 NEC Electronics Corporation 根据自身的判断在生产前撤消产品。
 - 并非在每个国家都可获得全部型号的产品。请联系 NEC Electronics 销售代理，了解供货和其他相关信息。
 - 未经NEC Electronics的书面许可，不能对本档复制。本档出现的任何错误，NEC Electronics不承担责任。
 - 如果用户在使用本档列出的 NEC Electronics 半导体产品或通过其他途径使用这些产品时，产生侵犯专利、版权以及其他侵犯第三方知识产权的行为，NEC Electronics 不承担责任。对于 NEC Electronics 及其他子公司的任何专利、版权以及其他知识产权，NEC Electronics 没有以许可、明示、暗示以及其他任何方式授权。
 - 文档中电路、软件和其他相关信息的描述，用来说明半导体产品操作和应用的例子。客户在使用这些电路、软件和信息时负全责。客户或第三方在使用这些电路、软件和信息时造成的损失，NEC Electronics不承担责任。
 - NEC Electronics尽力提高半导体产品的质量、可靠性和安全性，但请客户理解错误是不可能完全避免的。为了尽可能减少由于NEC Electronics半导体产品所带来的个人财产及人身安全（包括死亡）的风险，客户在设计过程中应加强安全措施，如容错、耐火性和自检等。
 - NEC Electronics产品分为以下三个质量等级：
 - “标准”、“专业”、“特级”。“特级”质量等级仅用于客户定制的半导体产品。一种半导体产品的应用主要依据它的质量等级。客户在使用某种半导体产品之前应先了解它的质量等级。
 - “标准”：计算机，办公设备，通信设备，测试设备，视频音频设备，家用电子产品，机械工具，个人电子设备和工业机器人
 - “专业”：运输设备（汽车，火车，轮船等），交通控制系统，防灾系统，反犯罪系统，安全设备和医疗设备（不是专用于生命救护的设备）
 - “特级”：飞机，航空设备，水下中继器，核反应堆控制系统，生命救护系统和用于生命救护的医疗设备等。除非在NEC Electronics规格书或规格手册中特别规定，一般的NEC Electronics产品的质量等级都是“标准”的。如果客户希望在不受NEC Electronics产品应用限制的条件下使用NEC Electronics半导体产品，必须事先与NEC Electronics销售代理联系，以确定NEC Electronics是否支持该应用环境。
- (注释)
- (1) “NEC Electronics” 在这里是指NEC Electronics Corporation和它的主要子公司。
 - (2) “NEC Electronics半导体产品” 是指由NEC Electronics或为NEC Electronics开发和制造的半导体产品（如上述定义）。

引言

读者对象

本手册适用于那些希望了解 78K0S/KU1+ 产品功能，并设计开发相关应用系统和程序的用户。

主要产品如下。

- 78K0S/KU1+: μ PD78F9200, 78F9201, 78F9202

目的

本手册用于帮助用户了解下面组织中描述的功能。

组织

78K0S/KU1+ 的手册主要分为两个部分：本手册和指令手册(通用 78K/0S 系列)。

78K0S/KU1+ 用户手册	78K/0S 系列 指令用户手册
<ul style="list-style-type: none">• 引脚功能• 内部模块功能• 中断• 其他内置外设功能• 电气特性	<ul style="list-style-type: none">• CPU 功能• 指令集• 指令描述

手册使用方法

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方面的一般知识。

- ◇ 如何理解 78K0S/KU1+ 的整体功能
 - 按 **目录** 顺序阅读本手册。
- ◇ 如何解释寄存器格式
 - 尖括号(<>)中的二进制位名在 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。
- ◇ 如何获悉某寄存器的详细信息
 - 可参阅 **附录 B 寄存器索引**。
- ◇ 如何获悉 78K/0S 系列指令的详细信息
 - 可参阅 **78K/0S 系列指令用户手册 (U11047E)**。
- ◇ 如何获悉 78K0S/KU1+ 的电气特性
 - 可参阅 **第十八章 电器特性**。

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
数的表示法:	二进制 ... xxxx 或 xxxx B
	十进制 ... xxxx
	十六进制 ... xxxx H

相关文档

本手册中指出的相关文档包括了最初的版本，但未注明。

设备文档

文档名称	文档编号
78K0S/KU1+用户手册	本手册
78K/0S 系列指令用户手册	U11047E

开发软件工具文档 (用户手册)

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

开发硬件工具文档 (用户手册)

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

注意事项 以上列出的相关文档在无任何声明条件下可能会作修改。读者应使用每个文档的最新版本用于设计。

Flash 存储器编程相关文档

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

其它相关文档

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

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

注意事项 以上列出的相关文档在无任何声明条件下可能会作修改。读者应使用每个文档的最新版本用于设计。

目录

第一章 概述.....	13
1.1 特征.....	13
1.2 订购信息.....	14
1.3 引脚配置(俯视图).....	15
1.4 78K0S/Kx1+系列介绍.....	16
1.5 框图.....	17
1.6 功能概述.....	18
第二章 引脚功能.....	19
2.1 引脚功能列表.....	19
2.2 引脚功能.....	21
2.2.1 P20 ~ P23 (端口 2).....	21
2.2.2 P32 和 P34 (端口 3).....	22
2.2.3 P40 和 P43 (端口 4).....	22
2.2.4 $\overline{\text{RESET}}$	22
2.2.5 X1 和 X2.....	22
2.2.6 V _{DD}	22
2.2.7 V _{SS}	22
2.3 引脚 I/O 电路和未使用引脚的建议连接方式.....	23
第三章 CPU 结构.....	25
3.1 存储空间.....	25
3.1.1 内部程序存储空间.....	28
3.1.2 内部数据存储空间.....	29
3.1.3 特殊功能寄存器(SFR)区域.....	29
3.1.4 数据存储器寻址.....	29
3.2 处理器寄存器.....	32
3.2.1 控制寄存器.....	32
3.2.2 通用寄存器.....	35
3.2.3 特殊功能寄存器 (SFRs).....	36
3.3 指令地址寻址.....	39
3.3.1 相对寻址.....	39
3.3.2 立即寻址.....	40
3.3.3 表间接寻址.....	40
3.3.4 寄存器寻址.....	41
3.4 操作数地址寻址.....	42
3.4.1 直接寻址.....	42
3.4.2 短直接寻址.....	43
3.4.3 特殊功能寄存器 (SFR) 寻址.....	44

3.4.4	寄存器寻址	45
3.4.5	寄存器间接寻址	46
3.4.6	基址寻址	47
3.4.7	堆栈寻址	48
第四章	端口功能	49
4.1	端口功能	49
4.2	端口配置	50
4.2.1	端口 2	50
4.2.2	端口 3	54
4.2.3	端口 4	55
4.3	控制端口功能的寄存器	56
4.4	端口功能操作	61
4.4.1	写入 I/O 端口	61
4.4.2	从 I/O 端口读入	61
4.4.3	I/O 端口的操作	61
第五章	时钟发生器	62
5.1	时钟发生器的功能	62
5.1.1	系统时钟振荡器	62
5.1.2	产生间隔时间的时钟振荡器	62
5.2	时钟发生器的配置	63
5.3	时钟发生器的控制寄存器	65
5.4	系统时钟振荡器	68
5.4.1	高速内部振荡器	68
5.4.2	晶体/陶瓷振荡器	68
5.4.3	外部时钟输入电路	70
5.4.4	预分频器	70
5.5	CPU 时钟发生器的操作	71
5.6	外围硬件时钟发生器的操作	77
第六章	16 位定时器/事件计数器 00	79
6.1	16 位定时器/事件计数器 00 的功能	79
6.2	16 位定时器/事件计数器 00 的配置	80
6.3	16 位定时器/事件计数器 00 的控制寄存器	84
6.4	16 位定时器/事件计数器 00 的操作	90
6.4.1	时间间隔定时器操作	90
6.4.2	外部事件计数器操作	92
6.4.3	脉冲宽度测量操作	95
6.4.4	方波输出操作	103
6.4.5	PPG 输出操作	105
6.4.6	单脉冲输出操作	108
6.5	有关 16 位定时器/事件计数器 00 的注意事项	113
第七章	8 位定时器 H1	120

7.1	8 位定时器 H1 的功能.....	120
7.2	8 位定时器 H1 的配置.....	120
7.3	控制 8 位定时器 H1 的寄存器.....	123
7.4	8 位定时器 H1 的操作.....	125
7.4.1	用作时间间隔定时器/方波输出的操作.....	125
7.4.2	用作 PWM 输出模式的操作.....	129
第八章 看门狗定时器.....		135
8.1	看门狗定时器的功能.....	135
8.2	看门狗定时器的配置.....	137
8.3	控制看门狗定时器的寄存器.....	138
8.4	看门狗定时器的操作.....	140
8.4.1	当由选项字节选择为“低速内部振荡不能被停止”时的看门狗定时器操作.....	140
8.4.2	当由选项字节选择为“低速内部振荡可通过软件停止”时的看门狗定时器操作.....	142
8.4.3	STOP 模式下看门狗定时器的操作(当选项字节选项为“可通过软件停止低速内部振荡时钟”).....	144
8.4.4	HALT 模式下的操作(当选项字节选项为“可通过软件停止低速内部振荡时钟”).....	145
第九章 A/D 转换器.....		146
9.1	A/D 转换器的功能.....	146
9.2	A/D 转换器配置.....	148
9.3	转换器中使用的寄存器.....	150
9.4	A/D 转换器的操作.....	155
9.4.1	A/D 转换器的基本操作.....	155
9.4.2	输入电压和转换结果.....	157
9.4.3	A/D 转换操作模式.....	158
9.5	A/D 转换器特征表的阅读方法.....	160
9.6	A/D 转换器使用注意事项.....	162
第十章 中断功能.....		165
10.1	中断功能类型.....	165
10.2	中断源和配置.....	165
10.3	控制中断功能的寄存器.....	167
10.4	中断服务操作.....	170
10.4.1	可屏蔽中断请求的响应操作.....	170
10.4.2	中断嵌套.....	172
10.4.3	中断请求等待.....	174
第十一章 待机功能.....		175
11.1	待机功能及配置.....	175
11.1.1	待机功能.....	175
11.1.2	待机期间用到的寄存器.....	177
11.2	待机功能的操作.....	178
11.2.1	HALT 模式.....	178
11.2.2	STOP 模式.....	181

第十二章 复位功能	185
12.1 确认复位源的寄存器	192
第十三章 上电清零电路	193
13.1 上电清零电路的功能	193
13.2 上电清零电路的配置	194
13.3 上电清零电路的操作	194
13.4 上电清零电路的使用注意事项	195
第十四章 低电压检测器	197
14.1 低电压检测器的功能	197
14.2 低电压检测器的配置	197
14.3 控制低电压检测器的寄存器	198
14.4 低电压检测器的操作	200
14.5 低电压检测器的注意事项	204
第十五章 选项字节	207
15.1 选项字节的功能	207
15.2 选项字节的格式	208
15.3 RESET 用于仅输入端口引脚 (P34)时的注意事项	209
第十六章 FLASH 存储器	210
16.1 特征	210
16.2 存储器配置	211
16.3 功能概要	211
16.4 使用 Flash 编程器的写入	212
16.5 编程环境	213
16.6 On-Board 引脚连接	215
16.6.1 X1 和 X2 引脚	215
16.6.2 $\overline{\text{RESET}}$ 引脚	216
16.6.3 端口引脚	217
16.6.4 电源供应	217
16.7 On-Board 和 Off-Board Flash 存储器编程	218
16.7.1 Flash 存储器编程模式	218
16.7.2 通讯命令	218
16.7.3 安全设置	219
16.8 通过自写入实现 Flash 存储器编程	220
16.8.1 自编程概述	220
16.8.2 自编程注意事项	223
16.8.3 自编程功能所使用的寄存器	223
16.8.4 从普通模式切换至自编程模式的示例	230
16.8.5 从自编程模式切换到普通模式的示例	233
16.8.6 自编程模式下 block 擦除操作示例	236
16.8.7 自编程模式下空白 block 的校验操作示例	239

16.8.8	自编程模式下字节写入操作示例	242
16.8.9	自编程模式下内部校验操作示例	245
16.8.10	自编程模式下命令执行时间最短的操作示例.....	249
16.8.11	自编程模式下中断禁止时间最短的操作过程.....	255
第十七章	指令集	266
17.1	操作使用规则	266
17.1.1	操作数标识符和标识方法	266
17.1.2	“操作”栏的描述	267
17.1.3	“标志”栏的描述	267
17.2	操作列表	268
17.3	按寻址类型列出指令	273
第十八章	电气特性	276
第十九章	封装图	288
第二十章	推荐的焊接条件	289
附录 A	开发工具	290
A.1	软件包	293
A.2	语言处理软件	293
A.3	控制软件	294
A.4	Flash 存储器写入工具	294
A.5	调试工具(硬件)	295
A.5.1	当使用在线仿真器 QB-78K0SKX1 时(开发中).....	295
A.5.2	当使用在线仿真器 QB-MINI2 时	295
A.5.3	当使用在线仿真器 IE-78K0S-NS 或 IE-78K0S-NS-A 时	296
A.5.4	当使用在线仿真器 QB-78K0SKX1MINI 时	296
A.6	调试工具(软件)	297
附录 B	寄存器索引	299
B.1	寄存器索引(按寄存器名称排序).....	299
B.2	寄存器索引(按寄存器符号排序).....	301
附录 C	注意事项列表	303

第一章 概述

1.1 特征

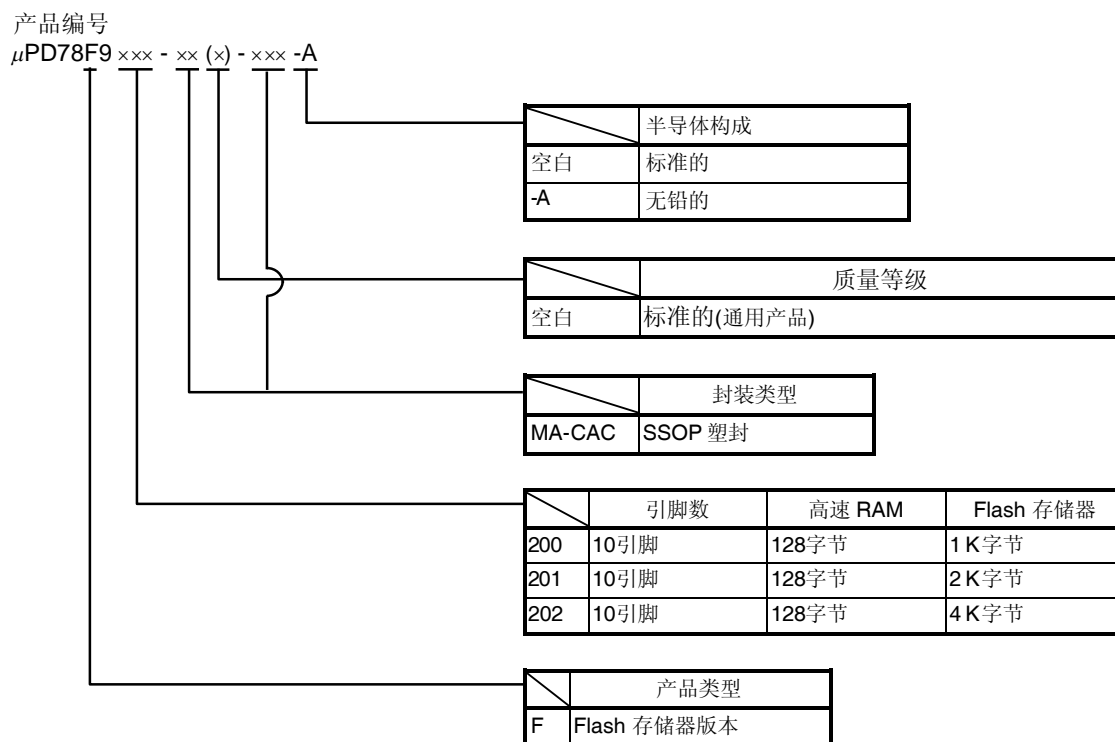
- 指令最短执行时间可以在高速(0.2 μ s)和低速(3.2 μ s)之间改变(系统的操作频率为 10 MHz)
- 通用寄存器: 8 位 \times 8 个寄存器
- ROM 和 RAM 的容量

产品型号 \ 项目	程序存储器 (Flash 存储器)	数据存储器 (内部高速 RAM)
μ PD78F9200	1 KB	128 字节
μ PD78F9201	2 KB	
μ PD78F9202	4 KB	

- 内置上电清零(POC)电路和低电压检测器 (LVI)
- 内置看门狗定时器(使用内置低速振荡器时钟进行操作)
- I/O 端口: 8
- 定时器: 3 通道
 - 16 位时间/事件计数器: 1 通道
 - 8 位定时器: 1 通道
 - 看门狗定时器: 1 通道
- 10 位分辨率 A/D 转换器: 4 通道
- 供应电压: $V_{DD} = 2.0 \sim 5.5 \text{ V}^{\text{注}}$
- 工作环境温度: $T_A = -40 \sim +85^\circ\text{C}$

注 因为上电清零(POC)的检测电压 (VPOC)在 2.1 V \pm 0.1 V 之间, 所以这种产品的电压范围是 2.2 ~ 5.5 V。

1.2 订购信息



请参考NEC公司发布的“NEC半导体设备质量等级”(文档编号 C11531E)，以了解设备的质量等级规范和推荐应用。

[产品编号列表]

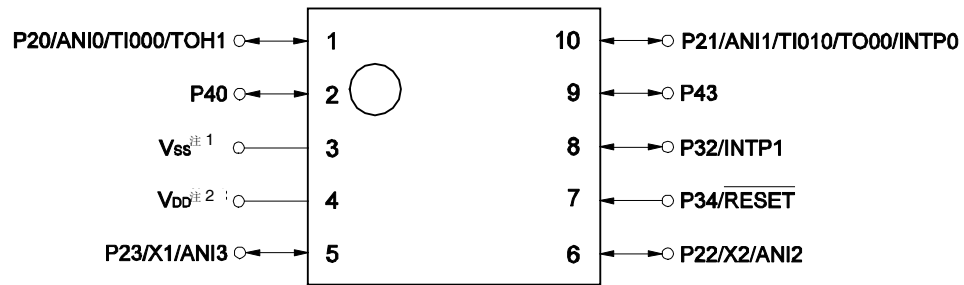
μ PD78F9200MA-CAC-A

μ PD78F9201MA-CAC-A

μ PD78F9202MA-CAC-A

1.3 引脚配置(俯视图)

10 引脚 塑封 SSOP



ANI0 to ANI3:	模拟输入	TI000, TI010:	定时器输入
INTP0, INTP1:	外部中断输入	TO00, TOH1:	定时器输出
P20 to P23:	端口 2	V _{DD} ^{#2} :	电源
P30, P34:	端口 3	V _{SS} ^{#1} :	地
P40, P43:	端口 4	X1, X2:	晶体振荡器 (X1 输入时钟)
$\overline{\text{RESET}}$:	复位		

- 注
1. 在 78K0S/KU1+ 中, V_{SS} 复用功能是作为 A/D 转换器的低电压, 确保连接 V_{SS} 到稳定的 GND (= 0 V)。
 2. 在 78K0S/KU1+ 中, V_{DD} 复用功能是作为 A/D 转换器参考电压输入。当使用 A/D 转换器时, 确保连接 V_{DD} 到稳定的供电电源(2.7 ~ 5.5 V)。

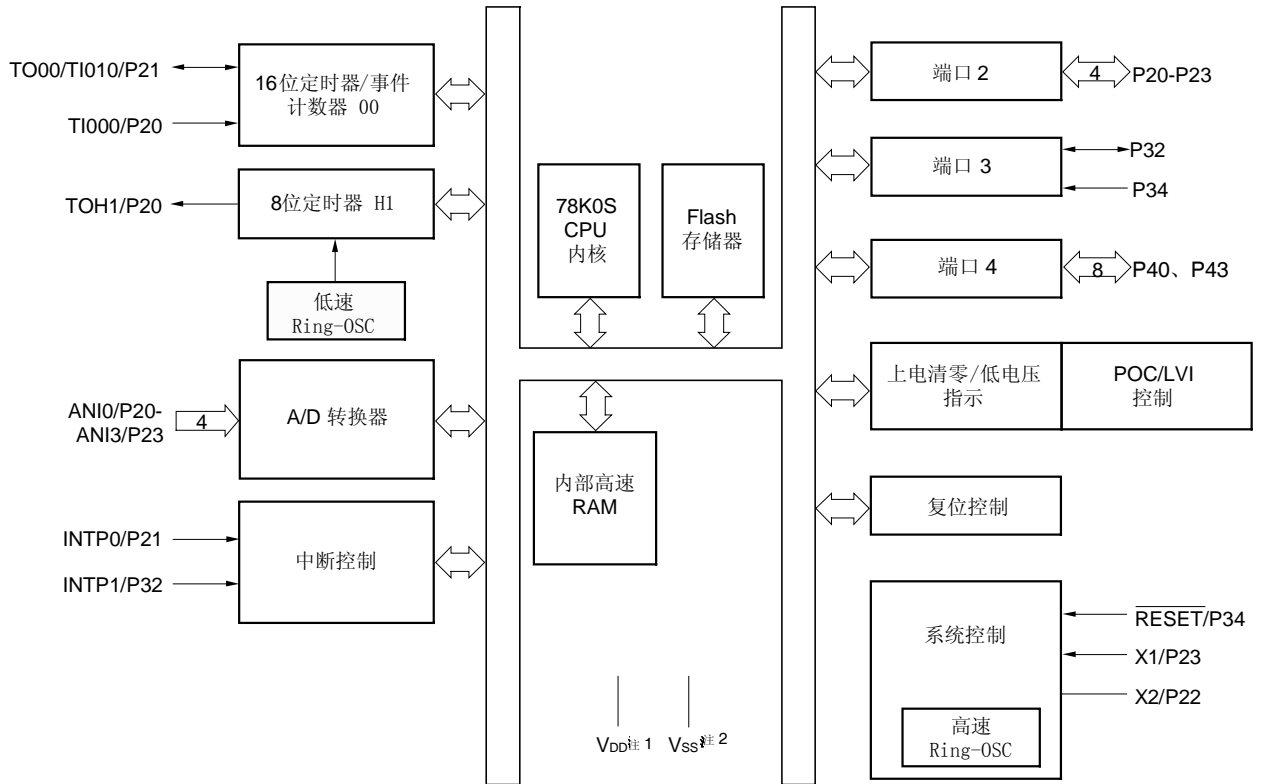
1.4 78K0S/Kx1+系列介绍

下面的表格给出了 78K0S/Kx1+ 的系列产品。

产品型号		78K0S/KU1+	78K0S/KY1+	78K0S/KA1+	78K0S/KB1+
项目					
引脚数目		10 引脚	16 引脚	20 引脚	30 引脚
内部存储器	Flash 存储器	1 KB, 2 KB, 4 KB		2 KB	4 KB, 8 KB
	RAM	128 字节		128 字节	256 字节
电源电压		VDD = 2.0 ~ 5.5 V ^注			
最小指令执行时间		0.20 μ s (10 MHz, VDD = 4.0 to 5.5 V) 0.33 μ s (6 MHz, VDD = 3.0 to 5.5 V) 0.40 μ s (5 MHz, VDD = 2.7 to 5.5 V) 1.0 μ s (2 MHz, VDD = 2.0 to 5.5 V)			
系统时间 (晶振频率)		高速内置振荡器 (8 MHz (典型值)) 晶体/陶瓷振荡器(1 ~ 10 MHz) 外部时钟输入振荡器(1 ~ 10 MHz)			
TMH1 和 WDT 时钟 (晶振频率)		低速内置振荡器(240 kHz (典型值))			
端口	CMOS I/O	7	13	15	24
	CMOS 输入	1	1	1	1
	CMOS 输出	-	-	1	1
定时器	16 位 (TM0)	1 通道			
	8 位 (TMH)	1 通道			
	8 位 (TM8)	-		1 通道	
	WDT	1 通道			
串行接口		-		支持 LIN 总线 UART: 1 通道	
A/D 转换器		10 位: 4 通道 (2.7 ~ 5.5V)			
乘法器 (8 位 \times 8 位)		-			提供
中断	外部	2		4	
	内部	5		9	
复位	RESET 引脚	提供			
	POC	2.1 V \pm 0.1 V			
	LVI	提供 (软件选择)			
	WDT	提供			
工作环境温度		标准产品: -40 ~ +85°C	标准产品, (A) 级别产品: -40 ~ +85°C (A2) 级别产品: -40 ~ +125°C		

注 因为上电清零(POC)的检测电压 (V_{POC})在 2.1 V \pm 0.1 V 之间, 所以本产品的电压范围是 2.2 ~ 5.5 V。

1.5 框图



- 注
1. 在 78K0S/KU1+中, V_{DD} 复用功能是作为 A/D 转换器参考电压输入。当使用 A/D 转换器时, 确保连接 V_{DD} 到稳定的供电电源(2.7 ~ 5.5 V)。
 2. 在 78K0S/KU1+ 中, V_{SS} 复用功能是作为 A/D 转换器的低电压, 确保连接 V_{SS} 到稳定的 GND (= 0 V)。

1.6 功能概述

项目		μ PD78F9200:	μ PD78F9201	μ PD78F9202
内部存储器	Flash 存储器	1 KB	2 KB	4 KB
	高速 RAM	128 字节		
存储空间		64 KB		
X1 输入时钟(晶振频率)		晶体/陶瓷 外部时钟输入: 10 MHz (VDD = 2.0 ~ 5.5 V)		
内部振荡时钟	高速(晶振频率)	内部振荡: 8 MHz (TYP.)		
	低速(TMH1 和 WDT)	内部振荡: 240 kHz (TYP.)		
通用寄存器		8 位 \times 8 个寄存器		
最小指令执行时间		0.2 μ s/0.4 μ s/0.8 μ s/1.6 μ s/3.2 μ s (X1 输入时钟: fX = 10 MHz)		
I/O 端口		全部: 8 引脚 CMOS I/O: 7 引脚 CMOS 输入: 1 引脚		
定时器		<ul style="list-style-type: none"> • 16 位定时器事件计数器: 1 通道 • 8 位定时器(定时器 H1): 1 通道 • 看门狗定时器: 1 通道 		
	定时器输出	2 引脚 (PWM: 1 引脚)		
A/D 转换器		10 位分辨率 \times 4 通道		
向量中断源	外部	2		
	内部	5		
复位		<ul style="list-style-type: none"> • 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 • 使用低电压检测电路进行内部复位 		
供电电源		V _{DD} = 2.0 to 5.5 V*		
工作环境温度		T _A = -40 ~ +85°C		
封装形式		10 引脚塑封 SSOP		

注 因为上电清零(POC)的检测电压 (V_{POC})在 2.1 V \pm 0.1 V 之间, 所以这种产品的电压范围是 2.2 ~ 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
P20	I/O	端口 2 4 位 I/O 端口 可以位选输入/输出方式 可以通过软件设置连接内部上拉电阻		输入	ANI0/TI000/TOH1
P21					ANI1/TI010/ TO00/INTP0
P22 ^{注1}					X2/ANI2 ^{注1}
P23 ^{注1}					X1/ANI3 ^{注1}
P32	I/O	端口 3	可以位选输入/输出方式 可以通过软件设置连接内部上拉电阻	输入	INTP1
P34 ^{注1}			仅输入		
P40, P43 ^{注2}	I/O	端口 4 2 位 I/O 端口 可以位选输入/输出方式 可以通过软件设置连接内部上拉电阻		输入	—

- 注
1. 引脚功能的设置方法，请参见 第十五章 选项字节。
 2. 程序设定中，设置 PM41, PM42, 和 PM44 ~ PM47 为 "0"。

注意事项 引脚 P22/X2/ANI2 和 P23/X1/ANI3 在复位期间为下拉状态。

(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	当指定的有效沿（上升沿、下降沿或双沿）到来时，产生外部中断请求	输入	P21/ANI1/TI010/ TO00
INTP1				P32
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	P20/ANI0/TOH1
TI010				P21/ANI1/TO00/ INTP0
TO00	输出	16 位定时器/事件计数器 00 输出	输入	P21/ANI1/TI010/ INTP0
TOH1	输出	8 位定时器 H1 输出	输入	P20/ANI0/TI000
ANI0	输入	A/D 转换器的模拟输入	输入	P20/TI000/TOH1
ANI1				P21/TI010/TO00/ INTP0
ANI2 ^注				P22/X2 ^注
ANI3 ^注				P23/X1 ^注
RESET ^注	输入	系统复位输入	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	-	P23/ANI3 ^注
X2 ^注	-	系统时钟振荡的晶体/陶瓷振荡器的连接	-	P22/ANI2 ^注
V _{DD}	-	电源电压	-	-
V _{SS}	-	地	-	-

注 引脚功能的设置方法，请参见 第十五章 选项字节。

注意事项 引脚 P22/X2/ANI2 和 P23/X1/ANI3 在复位期间为下拉状态。

2.2 引脚功能

2.2.1 P20 ~ P23 (端口 2)

P20~P23 是一个 4 位 I/O 端口。除了作为 I/O 端口引脚使用外，这些引脚还有输入模拟信号给 A/D 转换器的功能，输入/输出定时器信号和输入外部中断请求信号的功能。

P22 和 P23 也可分别复用为 X2/ANI2 和 X1/ANI3 功能。引脚功能的设置方法，请参见 **第十五章 选项字节**。这些引脚能位选设置以下的工作模式。

(1) 端口模式

P20 ~ P23 作为 4 位 I/O 端口使用。可通过端口模式寄存器 2 (PM2) 位选输入或输出模式。通过使用上拉电阻选择寄存器 2 (PU2)，可以选择连接端口的片内上拉电阻。

(2) 控制模式

P20 ~ P23 有作为 A/D 转换器的模拟信号输入，输入/输出定时器信号，和输入外部中断请求信号的功能。

(a) ANI0 ~ ANI3

这是 A/D 转换器的模拟输入引脚。当使用这些引脚作为模拟输入引脚使用时，请参照 **9.6 A/D 转换器的注意事项 (5) ANI0/P20 ~ ANI3/P23**。

(b) TI000

TI000 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

TI010 将捕捉信号输入到 16 位定时器/计数器 00 的捕捉寄存器(CR000)。

(d) TO00

TO00 从 16 位定时器/计数器 00 输出信号。

(e) TOH1

TOH1 是 8 位定时器 H1 的输出引脚。

(f) INTPO

这是外部中断请求输入引脚，可定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

注意事项 引脚 P22/X2/ANI2 和 P23/X1/ANI3 在复位期间为下拉状态。

2.2.2 P32 和 P34 (端口 3)

P32 是一个 1 位 I/O 端口。除了作为 I/O 端口引脚使用外，还有输入外部中断请求信号的功能。

P34 是一个 1 位输入端口。这个引脚也可作为 $\overline{\text{RESET}}$ 引脚，当电源上电，这时是复位功能。

引脚功能的设置方法，请参见 第十五章 选项字节。

当 P34 用于输入端口因交，连接到上拉电阻。

P32 和 P34 能位选设置以下的工作模式。

(1) 端口模式

P32 作为 1 位 I/O 端口使用。通过设置端口模式寄存器 3 (PM3)，这个引脚能够设置成输入或输出模式。除此之外，通过使用上拉电阻选择寄存器 3 (PU3)，可以选择连接端口的片内上拉电阻。

P34 仅作为 1 位输入端口使用。

(2) 控制模式

P32 可作为外部中断请求输入引脚 (INTP1)，可定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

2.2.3 P40 和 P43 (端口 4)

P40 ~ P47 作为 8 位 I/O 端口使用。通过设置端口模式寄存器 4 (PM4)，这个端口的每一位能够设置成输入或输出模式。除此之外，通过使用上拉电阻选择寄存器 4 (PU4)，可以选择连接端口的片内上拉电阻。

注 程序设定中，设置 PM41, PM42, 和 PM44 ~ PM47 为 "0"。

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

这个引脚输入低有效的系统复位信号。当电源上电，无论选项字节如何设定，这时是复位功能。

2.2.5 X1 和 X2

这两个引脚用于 X1 输入时钟的振荡器连接。

X1 和 X2 也分别用作 P23/ANI3 和 P22/ANI2。引脚功能的设置方法，请参见 第十五章 选项字节。

输入外部时钟信号给 X1。

注意事项 引脚 P22/X2/ANI2 和 P23/X1/ANI3 在复位期间为下拉状态。

2.2.6 V_{DD}

这是电源电压引脚。

在 78K0S/KU1+ 中，V_{DD} 复用功能是作为 A/D 转换器参考电压输入。当使用 A/D 转换器时，确保连接 V_{DD} 于稳定的供电电压 (2.7 ~ 5.5 V)。

2.2.7 V_{SS}

这是信号地引脚。

在 78K0S/KU1+ 中，V_{SS} 复用功能是作为 A/D 转换器的低电压，确保连接 V_{SS} 到稳定的 GND (= 0 V)。

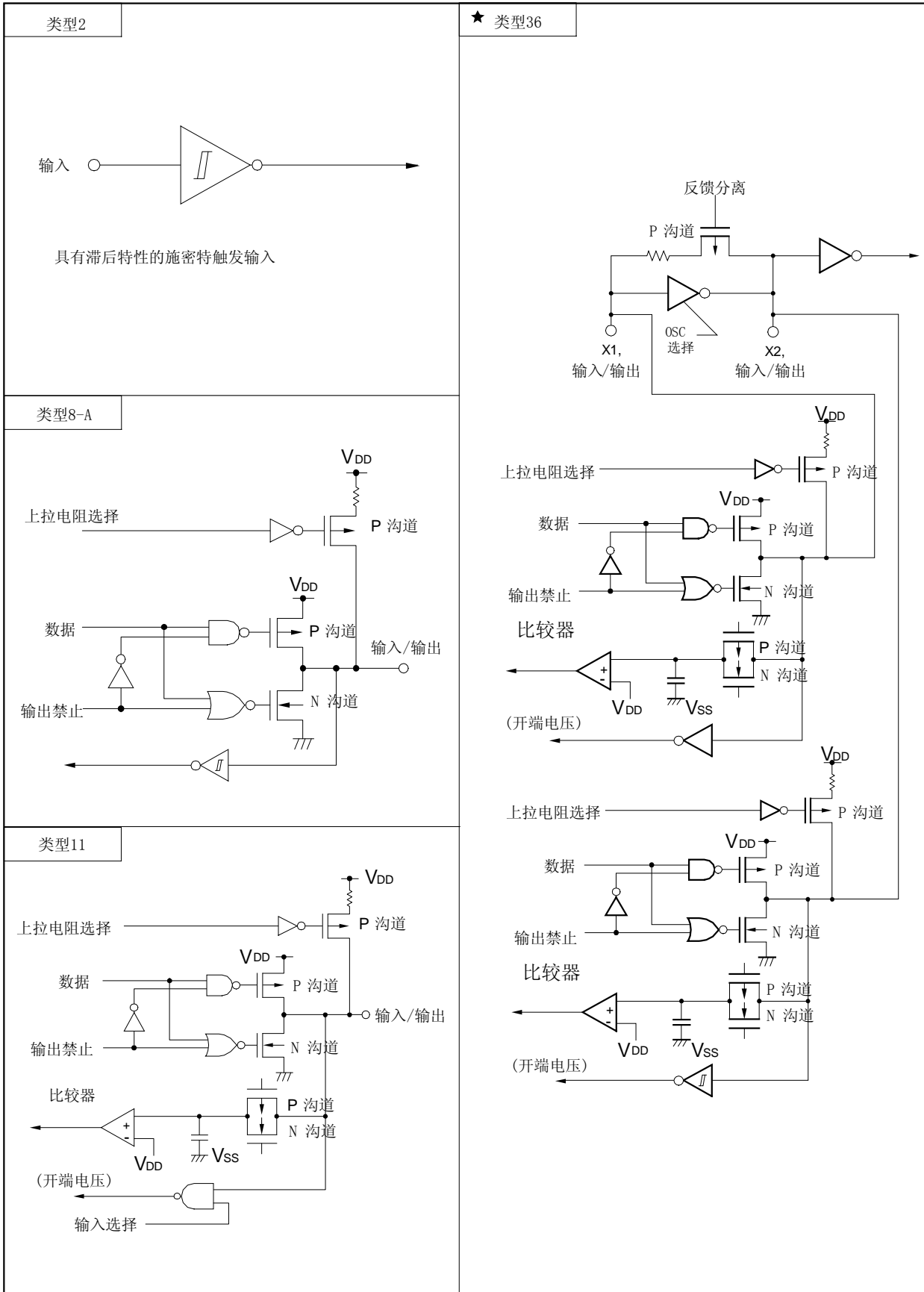
2.3 引脚 I/O 电路和未使用引脚的建议连接方式

表 2-1 给出了每个引脚地 I/O 电路类型和未使用引脚地建议连接方式。
每种类型的 I/O 电路的结构，请参照 图 2-1。

表 2-1. 引脚 I/O 电路的类型和未使用引脚的建议连接方式

引脚名称	I/O 电路类型	I/O	未使用引脚的建议连接方式
P20/ANI0/TI000/TOH1	11	I/O	输入: 串接电阻后单独连接到 V_{DD} 或者 V_{SS}
P21/ANI1/TI010/TO00/ INTP0			输出: 悬空
P22/ANI2/X2	36		输入: 串接电阻后单独连接到 V_{DD} 或者 V_{SS}
P23/ANI3/X1		输出: 悬空	
P32/INTP1	8-A		输入: 串接电阻后单独连接到 V_{DD} 或者 V_{SS} 输出: 悬空
P34/RESET	2	输入	串接电阻后连接到 V_{DD}
P40 and P43	8-A	I/O	输入: 串接电阻后连接到 V_{DD} 或者 V_{SS} 输出: 悬空

图 2-1. 引脚 I/O 电路

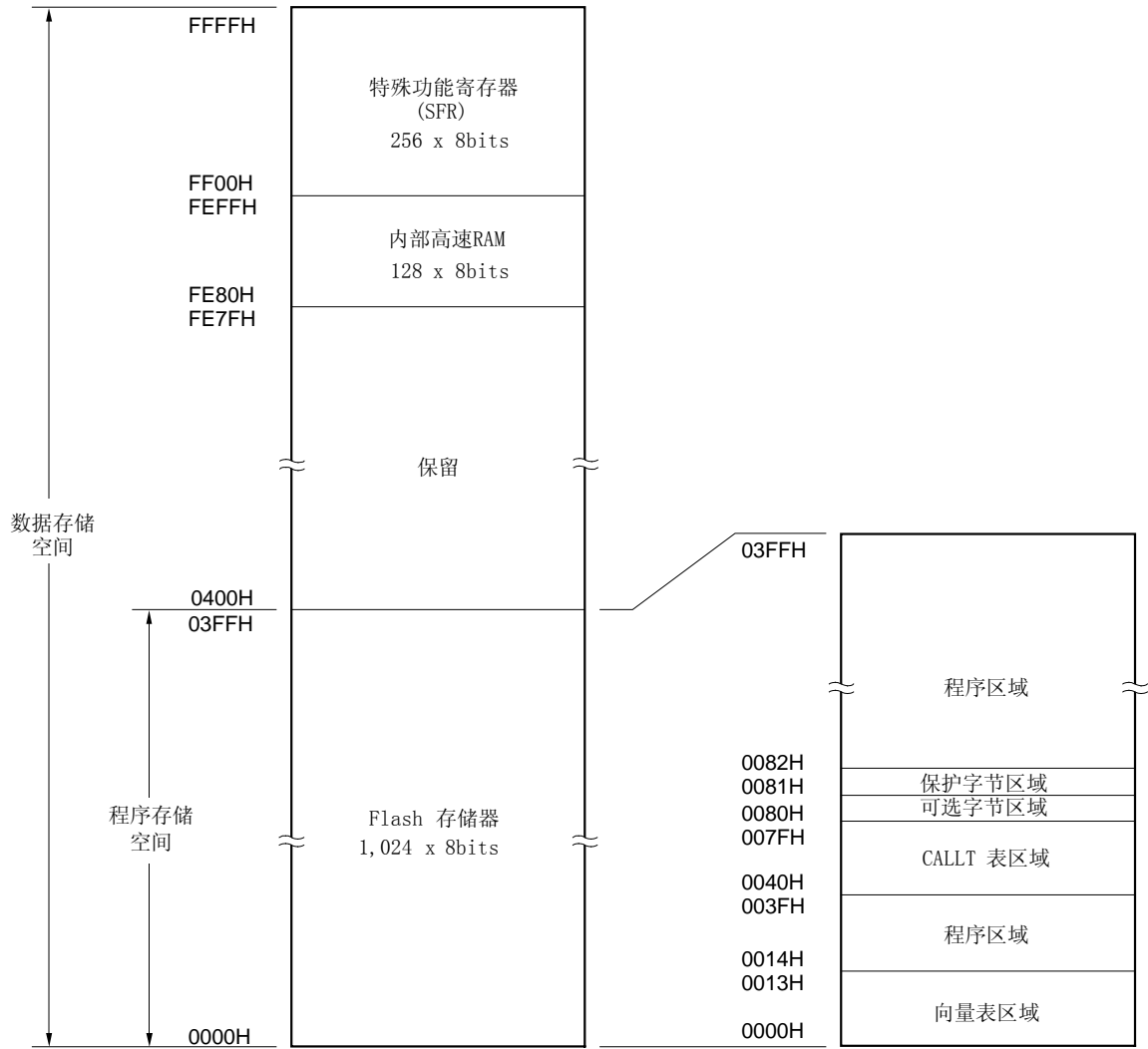


第三章 CPU 结构

3.1 存储空间

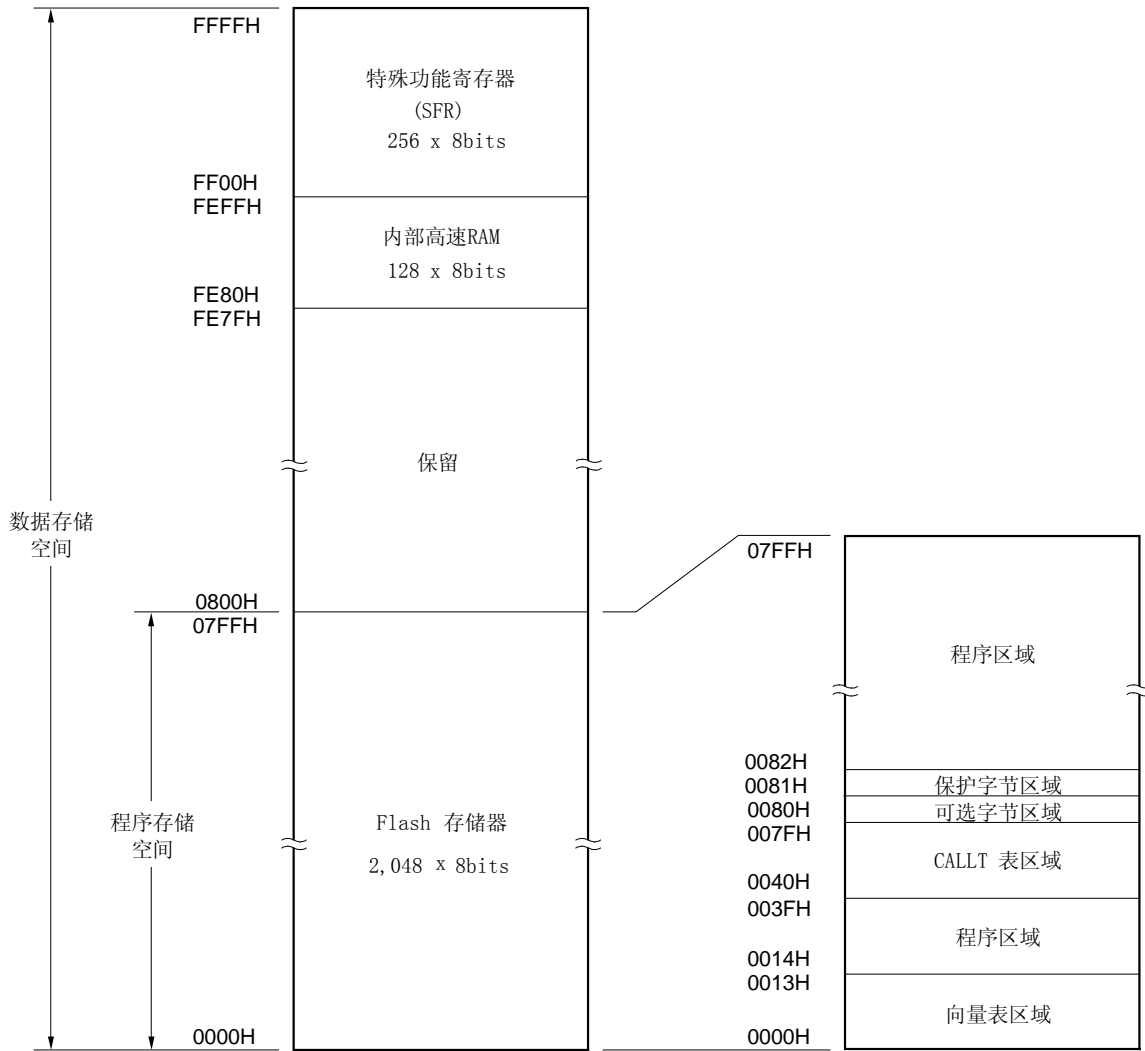
78K0S/KU1+ 可以访问大小为 64KB 的存储空间。图 3-1 ~ 3-3 显示了存储空间映射图。

图 3-1 存储空间映射图 (μ PD78F9200)



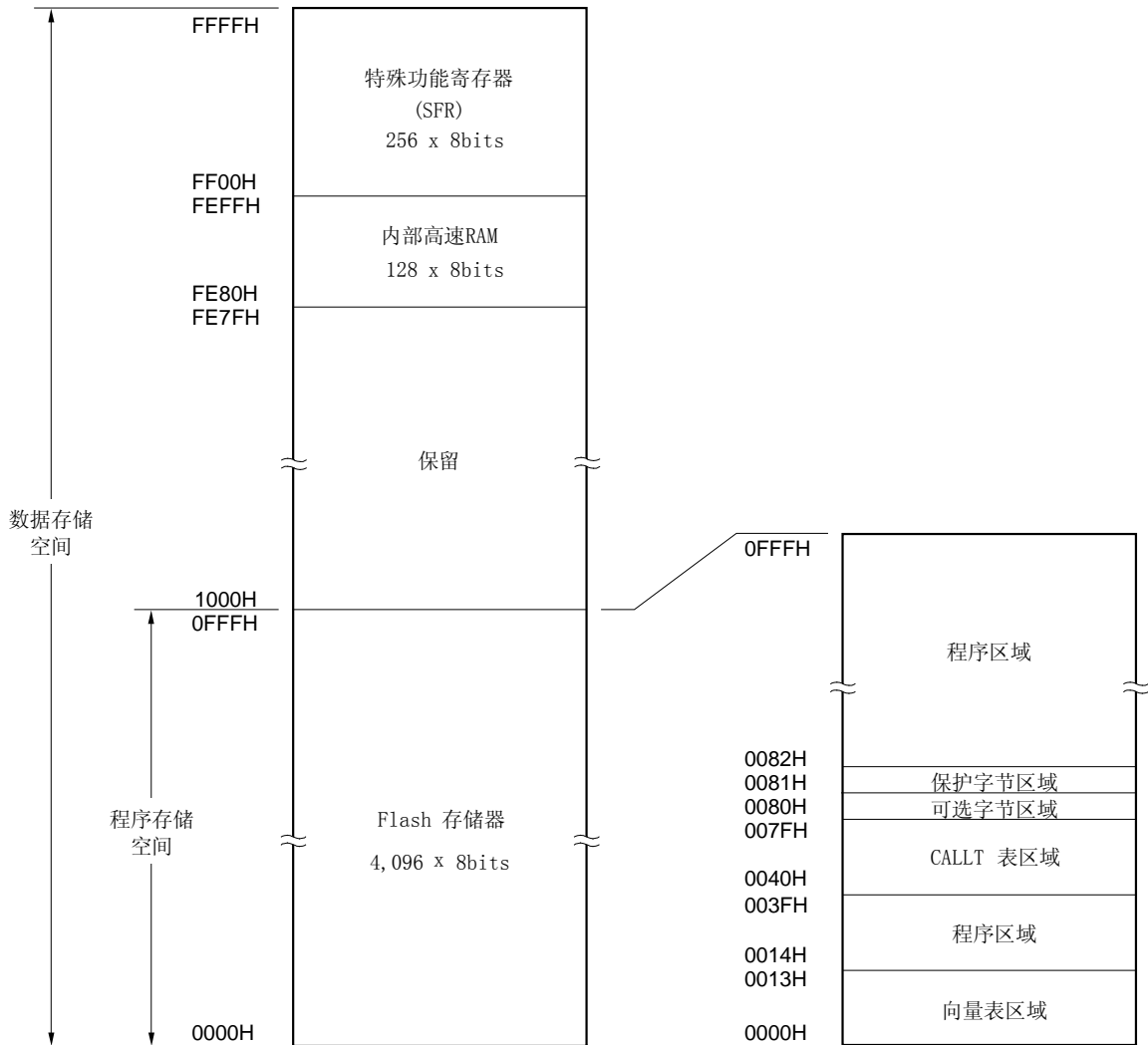
备注 选项字节和保护字节 各为 1 字节。

图 3-2 存储空间映射图 (μPD78F9201)



备注 选项字节和保护字节 各为 1 字节。

图 3-3 存储空间映射图 (μPD78F9202)



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。一般通过程序计数器(PC)来寻址。

78K0S/KU1+ 产品使用内部 ROM(或 FLASH 存储器)的容量情况如下表所示。

表 3-1. 内部 ROM 容量

产品型号	内部 ROM	
	结构	容量
μ PD78F9200	Flash 存储器	1,024 \times 8 位
μ PD78F9201		2,048 \times 8 位
μ PD78F9202		4,096 \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 0000H 到 0013H 总共 20 字节作为向量表区域。在向量表中存放的是根据复位信号输入或每个中断请求的产生进行转移的程序的起始地址。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	000CH	INTTMH1
0006H	INTLVI	000EH	INTTM000
0008H	INTP0	0010H	INTTM010
000AH	INTP1	0012H	INTAD

(2) CALLT 指令表区域

0040H ~ 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。若要了解细节，参考 第十五章 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。若要了解细节，参考 第十六章 FLASH 存储器

3.1.2 内部数据存储空间

78K0S/KU1+提供 128 字节的内部高速 RAM。

内部高速 RAM 也可以被用作堆栈存储器。

3.1.3 特殊功能寄存器(SFR)区域

片上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/KU1+ 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 (FE80H ~ FE7FH) 能够通过与其功能相一致的唯一寻址模式进行访问。图 3-4 至 3-6 举例说明数据存储空间寻址。

图 3-4. 数据存储空间寻址 (μ PD78F9200)

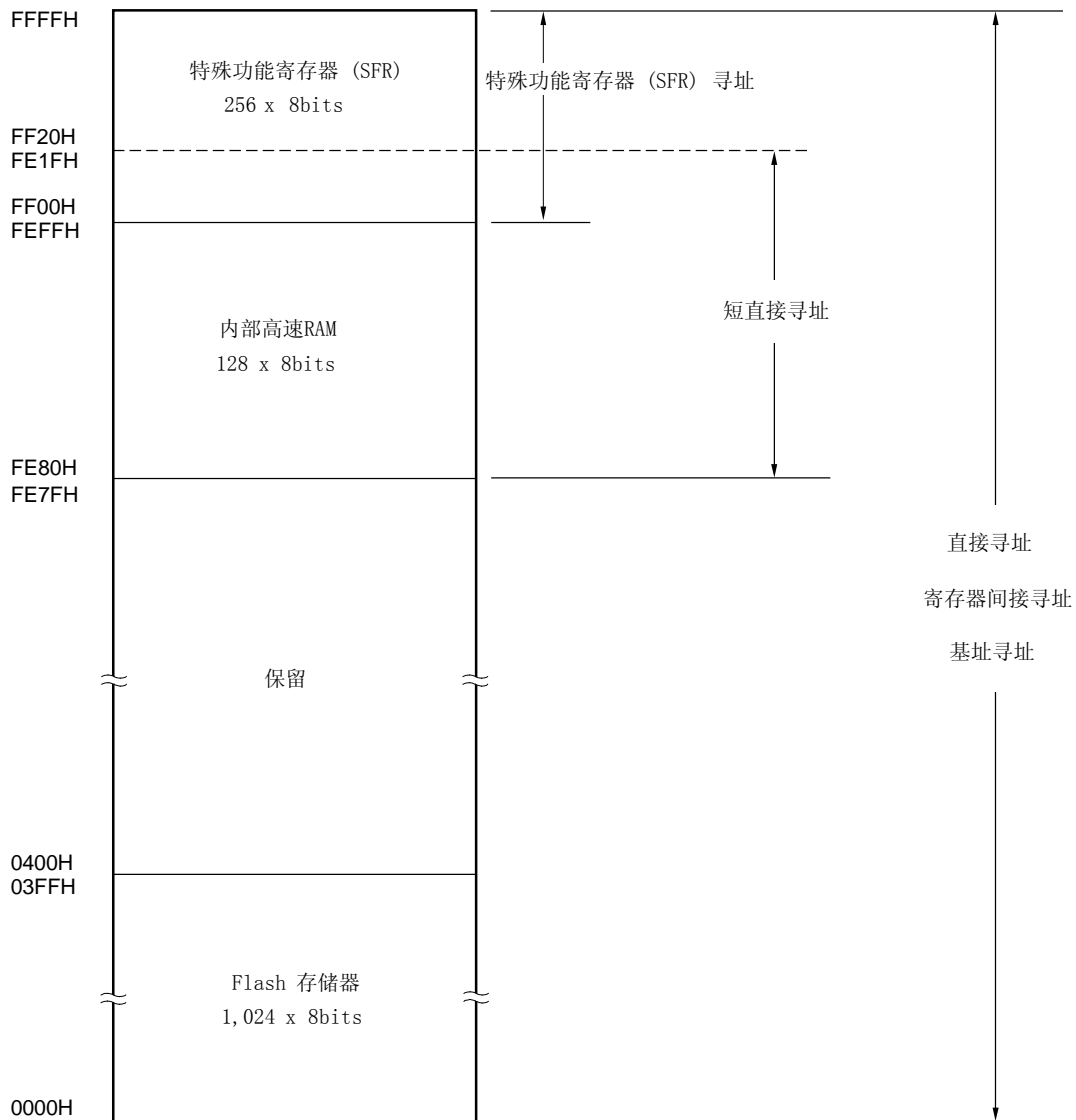


图 3-5. 数据存储空间寻址 (μ PD78F9201)

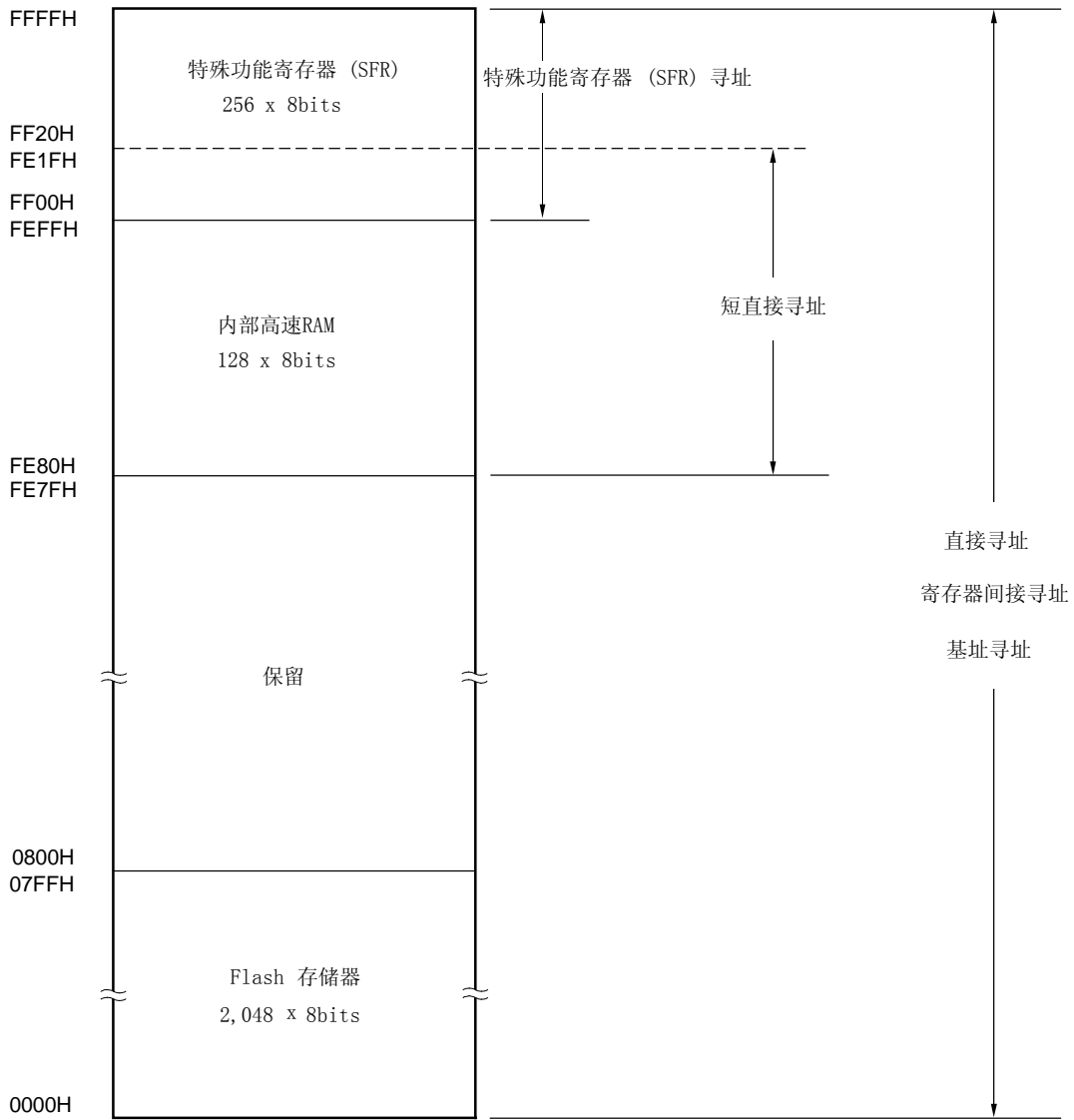
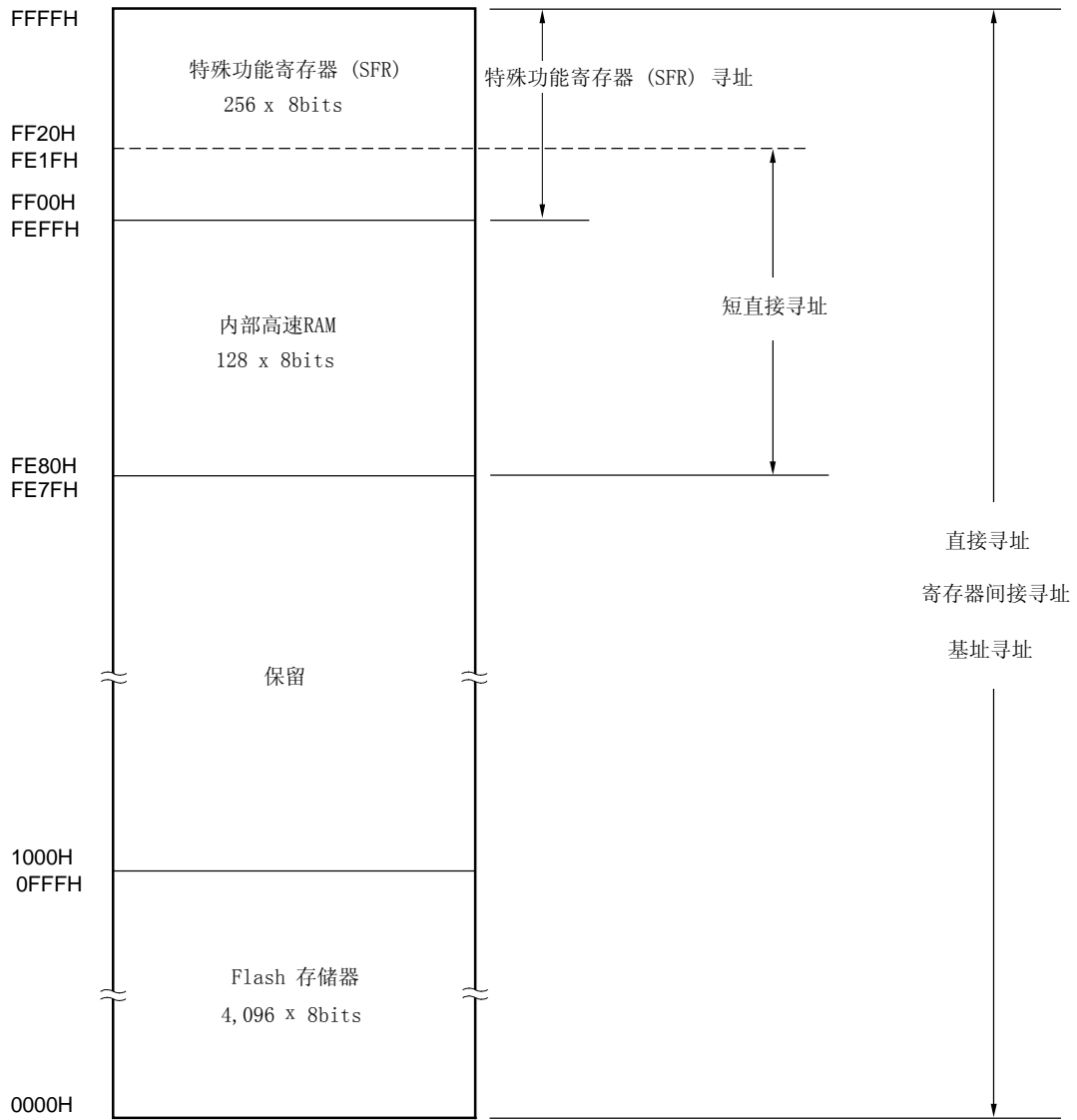


图 3-6. 数据存储空间寻址 (μ PD78F9202)



3.2 处理器寄存器

78K0S/KU1+ 包含以下几种片内处理器寄存器。

3.2.1 控制寄存器

控制寄存器用于控制程序执行的顺序的状态和堆栈空间。控制寄存器包括程序计数器、程序状态字和堆栈指针)。

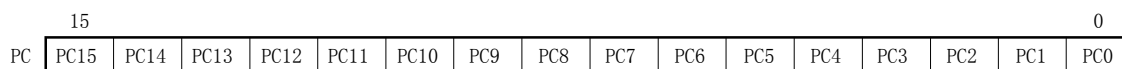
(1) 程序计数器 (PC)

程序计数器是一个 16 位寄存器，用于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，PC 的值会自动累加。当执行分支指令时，则设置为立即数或寄存器的值。

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

图 3-7. 程序计数器结构



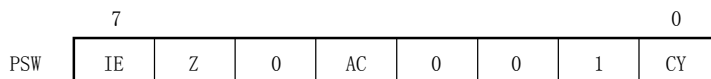
(2) 程序状态字 (PSW)

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

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

复位信号输入将程序状态字的值设定 02H。

图 3-8. 程序状态字结构



(a) 中断允许标志 (IE)

该标志用于控制 CPU 响应中断请求操作。

当 IE=0 时，表示不允许中断 (DI)，即禁止了除不可屏蔽中断外的所有中断。

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

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

(b) 零标志 (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。

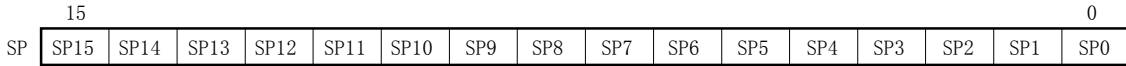
(d) 进位标志 (CY)

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

(3) 堆栈指针 (SP)

这是一个 16 位寄存器，用来存放内部堆栈区的起始地址。只有内部高速 RAM 区才能被设置为堆栈区(除内部高速 RAM 之外，别的区域不能设置为堆栈区)。

图 3-9. 堆栈指针结构



在向堆栈写(存储)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。
堆栈的数据存储/恢复操作过程如图 3-10 和 3-11 所示。

- 注意事项**
1. 由于复位输入使 SP 的内容不确定，所以在使用堆栈前必须先对 SP 初始化。
 2. 堆栈指针只能设置为高速 RAM 区域，并且只有低 10 位可以实际设置。
因此，如果堆栈指针指定为 0FF00H，它会转向高速 RAM 区域的 0FB00H，因为 0FF00H 是在 SFR 区域并且不在高速 RAM 区域。
当值实际推向指针时，从 0FB00H 减 1 为 0FAFFH，但是因为这个值不在高速 RAM 区域，它会转向 0FEFFH，是和当 0FF00H 设置为堆栈指针时的同样的值。

图 3-10. 将数据存入堆栈

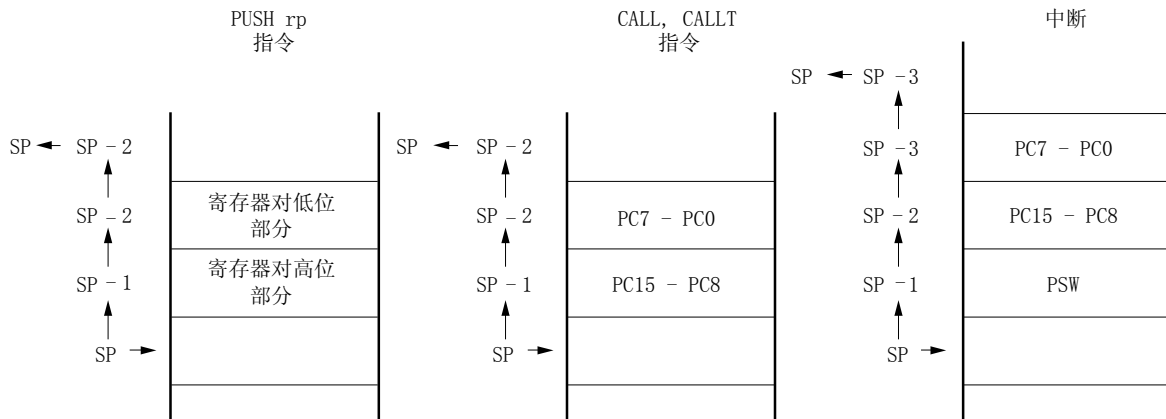
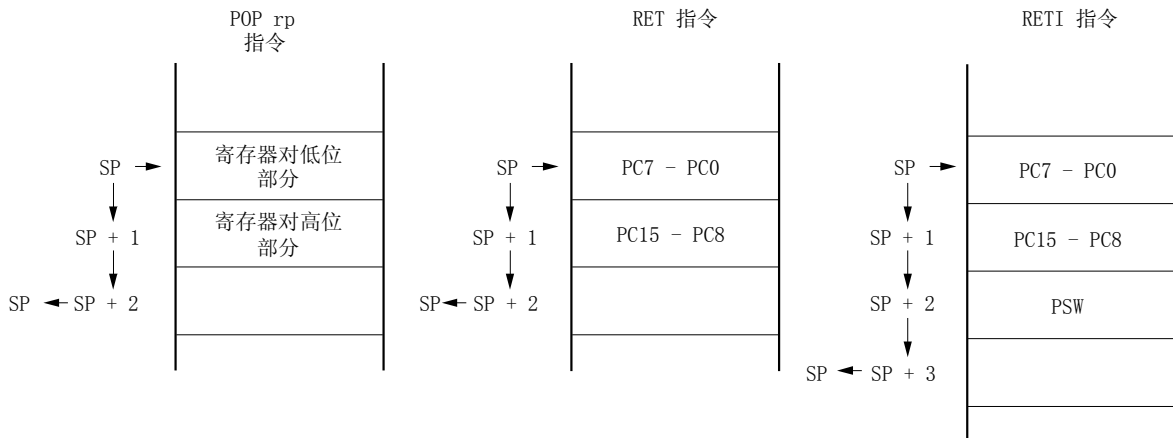


图 3-11. 从堆栈读出数据



3.2.2 通用寄存器

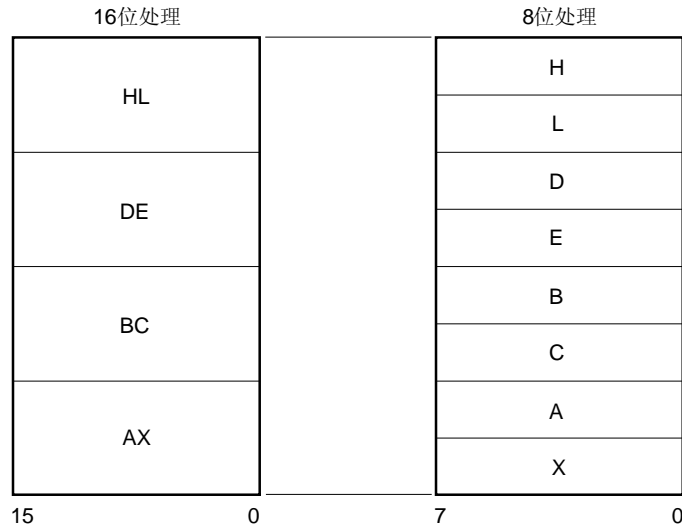
通用寄存器包含了 8 个 8 位寄存器(X, A, C, B, E, D, L 和 H)。

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)。

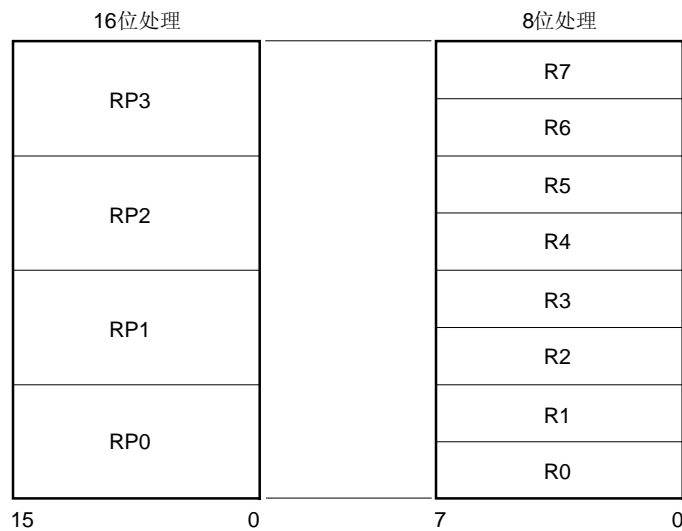
描述通用寄存器可以使用功能名称 (X, A, C, B, E, D, L, H, AX, BC, DE 和 HL) 和绝对名称 (R0 ~ R7 和 RP0 ~ RP3)。

图 3-12. 通用寄存器结构

(a) 功能名称



(b) 绝对名称



3.2.3 特殊功能寄存器 (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。

FF00H ~ FFFFH 总共 256 字节的区域分配给特殊功能寄存器。

特殊功能寄存器可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，可以是 1 位、8 位和 16 位。

每个操作单元描述如下。

- 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。该操作也可以由一个地址来定义。
- 8 位操作
8 位操作指令的操作数(sfr)被描述为汇编程序的保留符号。该操作也可以由一个地址来定义。
- 16 位操作
16 位操作指令的操作数(sfrp)被描述为汇编程序的保留符号。该操作也可以由一个地址(偶地址)来定义。

表 3-3 为特殊功能寄存器列表。表中术语的含义如下。

- 符号
符号表示特殊功能寄存器的地址，在 RA78K0S 中是保留字，并在 CC78K0S 中通过使用 #pragma sfr 伪指令来声明它们作为 sfr 变量使用。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: 可读写
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元(1, 8, 和 16)。
- 复位后
表示复位信号输入后特殊功能寄存器的状态。

表 3-3. 特殊功能寄存器 (1/2)

地址	特殊功能寄存器(SFR)名称	符号	R/W	可同时操作的位单元数目			复位后
				1 位	8 位	16 位	
FF02H	端口寄存器 2	P2	R/W 注 1	√	√	—	00H
FF03H	端口寄存器 3	P3		√	√	—	
FF04H	端口寄存器 4	P4		√	√	—	
FF0EH	8 位定时器 H 比较寄存器 01	CMP01	R/W	—	√	—	
FF0FH	8 位定时器 H 比较寄存器 11	CMP11		—	√	—	
FF12H	16 位定时器计数器 00	TM00	R	—	—	√注 2	0000H
FF13H							
FF14H	16 位定时器捕捉/比较寄存器 000	CR000	R/W	—	—	√注 2	0000H
FF15H							
FF16H	16 位定时器捕捉/比较寄存器 010	CR010		—	—	√注 2	0000H
FF17H							
FF18H	10 位 A/D 转换结果寄存器	ADCR	R	—	—	√注 2	未定义
FF19H							
FF1AH	8 位 A/D 转换结果寄存器	ADCRH		—	√	—	
FF22H	端口模式寄存器 2	PM2	R/W	√	√	—	FFH
FF23H	端口模式寄存器 3	PM3		√	√	—	
FF24H	端口模式寄存器 4	PM4		√	√	—	
FF32H	上拉电阻选择寄存器 2	PU2		√	√	—	
FF33H	上拉电阻选择寄存器 3	PU3		√	√	—	00H
FF34H	上拉电阻选择寄存器 4	PU4		√	√	—	
FF48H	看门狗定时器模式寄存器	WDTM		—	√	—	67H
FF49H	看门狗定时器使能寄存器	WDTE		—	√	—	9AH
FF50H	低电压检测寄存器	LVIM		√	√	—	00H注 3
FF51H	低电压检测电平选择寄存器	LVIS		—	√	—	
FF54H	复位控制标志寄存器	RESF	R	—	√	—	00H注 4
FF58H	低速内部振荡模式寄存器	LSRCM	R/W	√	√	—	00H
FF60H	16 位定时器模式控制寄存器 00	TMC00		√	√	—	
FF61H	预分频模式寄存器 00	PRM00		√	√	—	
FF62H	捕捉/比较控制寄存器 00	CRC00		√	√	—	
FF63H	16 位定时器输出控制寄存器 00	TOC00		√	√	—	
FF70H	8 位定时器 H 模式寄存器 1	TMHMD1		√	√	—	
FF80H	A/D 转换器模式寄存器	ADM		√	√	—	
FF81H	模拟输入通道指定寄存器	ADS		√	√	—	
FF84H	端口模式控制寄存器 2	PMC2		√	√	—	

- 注
1. P34 仅能作为输入端口。
 2. 仅能通过短直接寻址方式进行 16 位访问。
 3. 只有在 LVI 引起的复位后才能保持。
 4. 随复位原因的不同而不同。

表 3-3. 特殊功能寄存器 (2/2)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可同时操作的位数			复位后
				1 位	8 位	16 位	
FFA0H	Flash 保护命令寄存器	PFCMD	W	–	√	–	未定义
FFA1H	Flash 状态寄存器	PFS	R/W	√	√	–	00H
FFA2H	Flash 编程模式控制寄存器	FLPMC		–	√	–	未定义
FFA3H	Flash 编程命令寄存器	FLCMD		√	√	–	00H
FFA4H	Flash 地址指针 L	FLAPL		√	√	–	未定义
FFA5H	Flash 地址指针 H	FLAPH		√	√	–	
FFA6H	Flash 地址指针 H 比较寄存器	FLAPHC		√	√	–	00H
FFA7H	Flash 地址指针 L 比较寄存器	FLAPLC		√	√	–	
FFA8H	Flash 写缓冲寄存器	FLW		–	√	–	
FFE0H	中断请求标志寄存器 0	IF0		√	√	–	
FFE4H	中断屏蔽标志寄存器 0	MK0		√	√	–	FFH
FFECH	外部中断模式寄存器 0	INTM0		–	√	–	00H
FFF3H	预处理器时钟控制寄存器	PPCC		√	√	–	02H
FFF4H	振荡稳定时间选择寄存器	OSTS		–	√	–	未定义 注
FFFBH	处理器时钟控制寄存器	PCC		√	√	–	02H

注 复位释放后的振荡稳定时间可通过选项字节选择。详细信息，请参考 第十五章 选项字节。

3.3 指令地址寻址

一条指令的地址是由程序计数器(PC)决定的。根据执行指令时所获取的下一条指令的字节数，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 **78K/0S 系列指令用户手册(U11047E)**)。

3.3.1 相对寻址

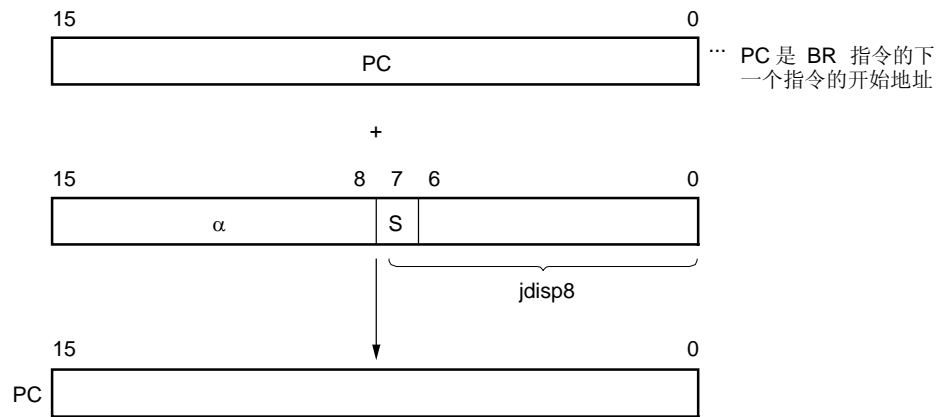
[功能]

将一条指令的 8 位立即数(偏移量: **jdisp8**)与下一条指令的起始地址相加，结果赋给程序计数器(PC)，然后转向相加结果指向的地址。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，分支的范围是从下一条指令起始地址的-128 ~ +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]



当S=0, α的所有位都是0

当S=1, α的所有位都是1

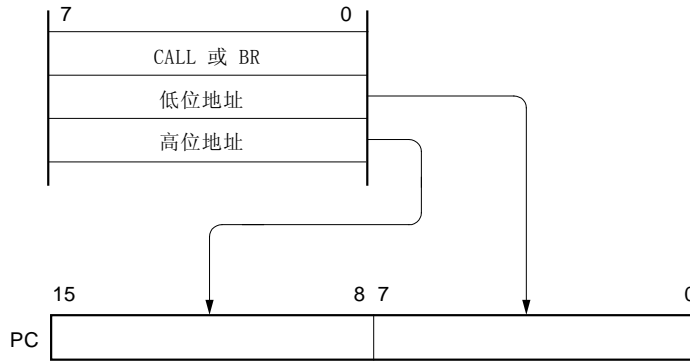
3.3.2 立即寻址

[功能]

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

[图示]

CALL !addr16 指令和 BR !addr16 指令

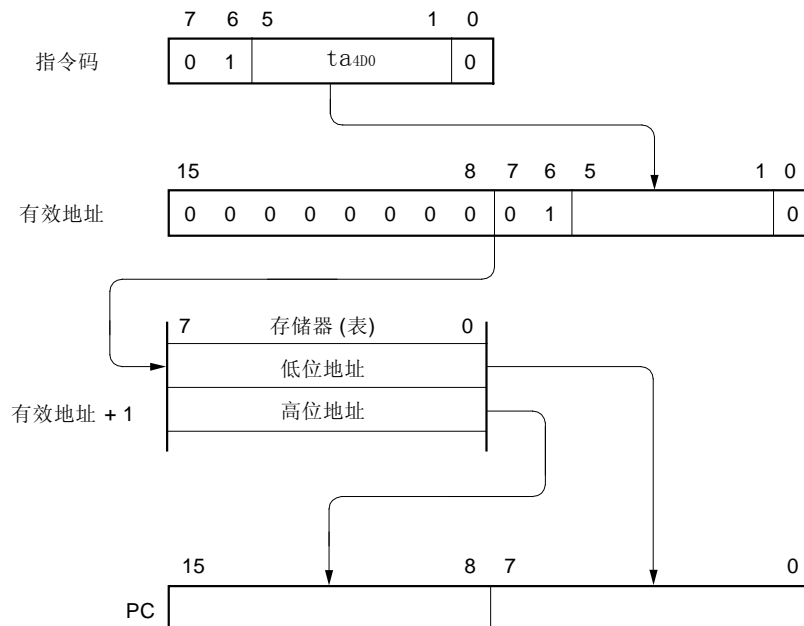


3.3.3 表间接寻址

[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 ~ 7FH 之间所存储的地址，转移地址范围可以是整个存储器空间。

[图示]

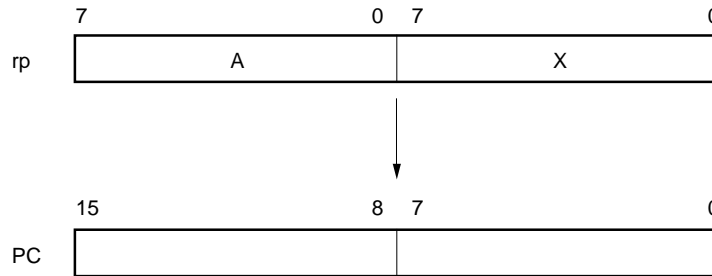


3.3.4 寄存器寻址

[功能]

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

[图示]



3.4 操作数地址寻址

以下方法用来规定指令执行期间寄存器寻址和存储器寻址所进行的操作。

3.4.1 直接寻址

[功能]

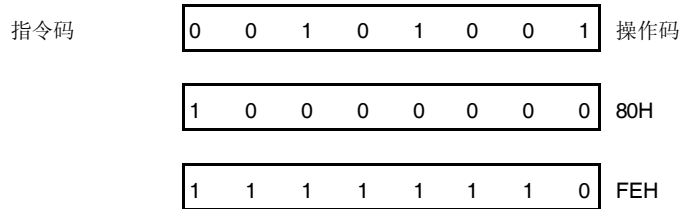
直接寻址方式根据指令中的立即数直接寻址。

[操作数格式]

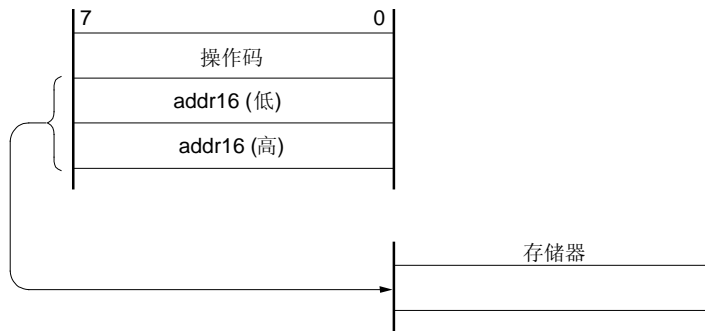
标识符	描述
addr16	标号或 16 位立即数

[举例]

MOV A, !FE80H; 将!addr16 设置为 FE80H



[图示]



3.4.2 短直接寻址

[功能]

用指令中的 8 位立即数直接对内存的固定操作区域寻址。

该方式的寻址范围为 FE80H ~ FF1FH 的 160 字节的空间。内部高速 RAM 映射在地址 FE80H ~ FEFFH，特殊功能寄存器 (SFR) 映射在地址 FF00H ~ FF1FH。

采取短直接寻址的特殊功能寄存器(SFR)区域(FF00H ~ 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。

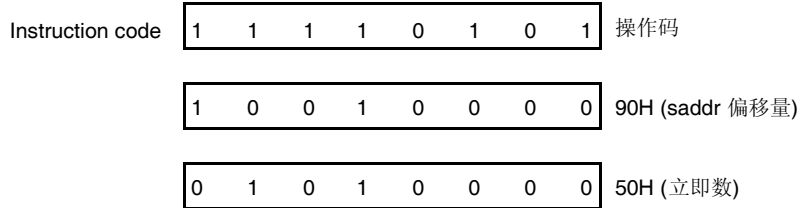
如果 8 位立即数在 80H ~ FFH 之间，则将一个有效地址的第 8 位设置为 0；如果 8 位立即数是在 00H 到 1FH 之间，则把一个有效地址的第 8 位设置为 1。参见下面的[图示]。

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

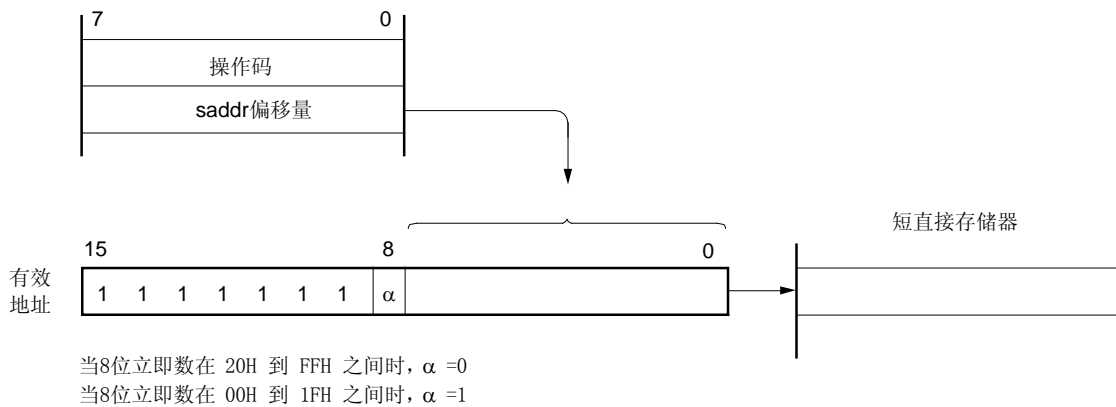
[举例]

EQU DATA1 0FE90H ; DATA1 指向单元 FE90H

MOV DATA1, #50H ; 设置立即数 50H



[Illustration]



3.4.3 特殊功能寄存器 (SFR) 寻址

[功能]

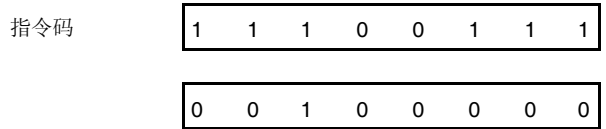
通过指令中的 8 位立即数对内存中的特殊功能寄存器(SFR)区域进行寻址。
 这种寻址方式可以应用到 FF00H ~ FFFFH 共 256 字节的空间。访问 SFR 映射在 FF00H ~ FF1FH 区域的区间需要使用短直接寻址方式。

[操作数格式]

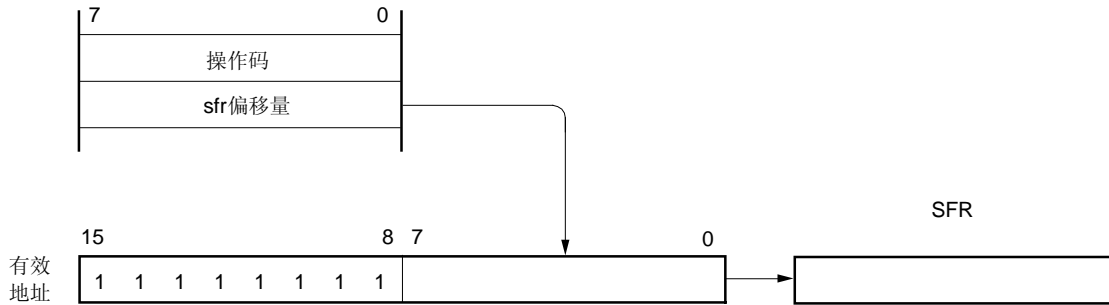
标识符	描述
sfr	特殊功能寄存器名

[举例]

MOV PM0, A; 当选择 PM0 为 sfr 时



[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。

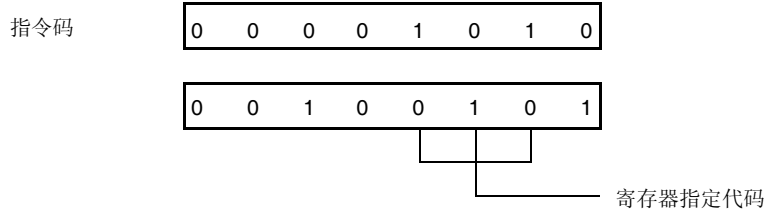
[操作数格式]

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

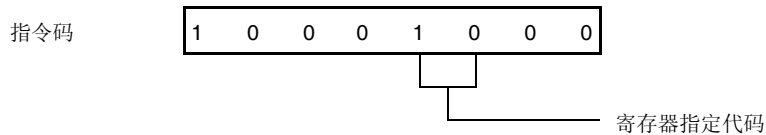
‘r’ 和 ‘rp’ 可用绝对名称 (R0 ~ R7 和 RP0 ~ RP3) 或者功能名称 (X, A, C, B, E, D, L, H, AX, BC, DE, 和 HL) 来描述。

[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。由指令中的寄存器对标识码来指定要访问的寄存器对。可以在所有内存空间使用这种寻址方式。

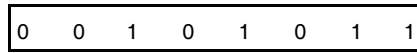
[操作数格式]

标识符	描述
-	[DE], [HL]

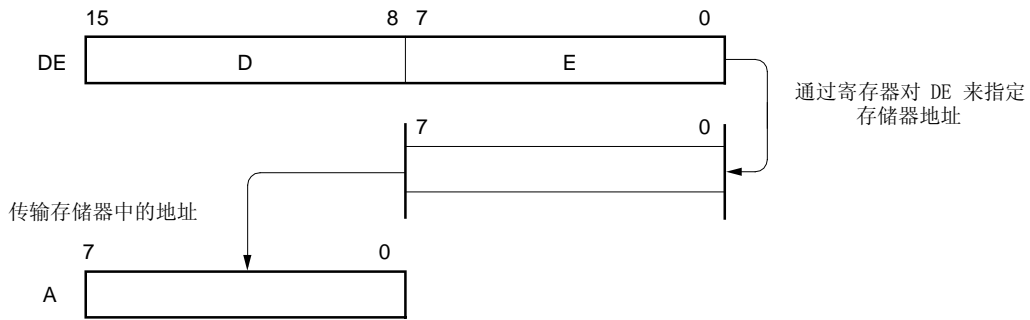
[举例]

MOV A, [DE]; 选择寄存器对 DE 作为操作数

指令码



[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 寄存器对作为基址寄存器，根据相加的结构去寻址。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。这种寻址方式可以对整个内存空间进行。

[操作数格式]

标识符	描述
-	[HL+byte]

[举例]

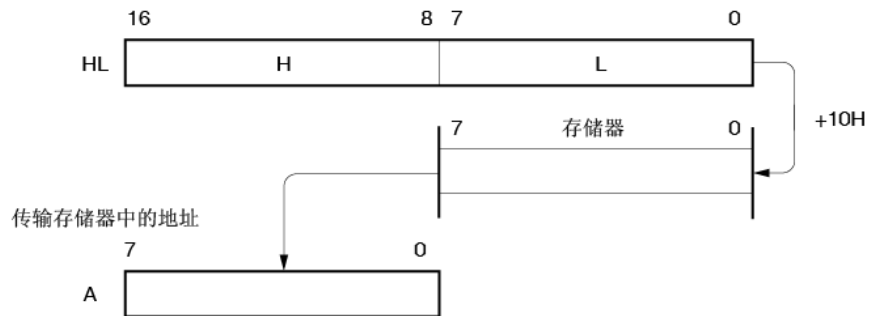
MOV A, [HL+10H]; byte 的值为 10H

指令码

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

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

[图示]



3.4.7 堆栈寻址

[功能]

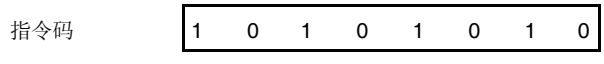
根据堆栈指针(SP)的内容对堆栈区域进行间接寻址。

当执行 **PUSH**、**POP**、子程序调用和 **RETURN** 指令时，或者根据中断请求产生对寄存器进行设置或复位时，将自动使用这种寻址方式。

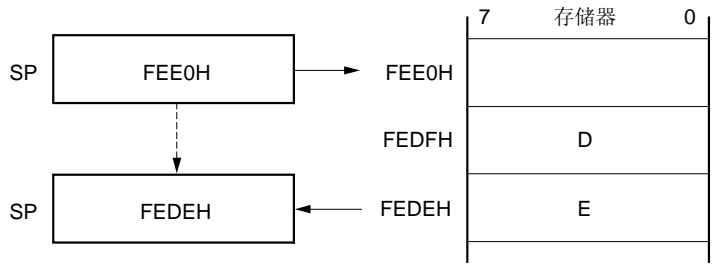
该方式仅能对内部高速 **RAM** 区域进行寻址。

[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/KU1+产品中提供的端口如图 4-1 所示，可被用于不同的控制操作。每个端口功能如表 4-1 所示。除了作为数字 I/O 端口外，这些端口还有复用功能。需要了解这些端口详细的复用功能，请参阅第二章 引脚功能。

图 4-1 端口功能

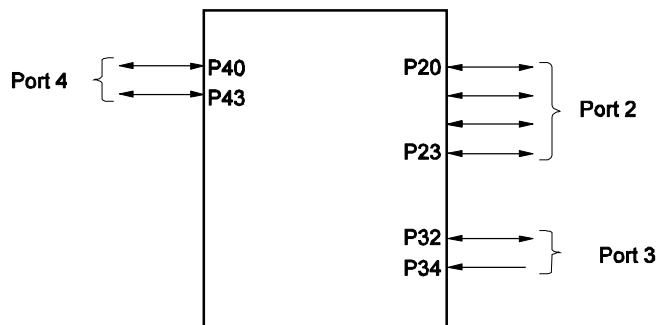


表 4-1 端口功能

引脚名称	I/O	功能		复位后	复用功能引脚
P20	I/O	端口 2 4 位 I/O 端口 可以位选输入/输出方式 可以通过软件设置连接内部上拉电阻		输入	ANI0/TI000/TOH1
P21					ANI1/TI010/TO00/ INTP0
P22 ^{※1}					X2/ANI2 ^{※1}
P23 ^{※1}					X1/ANI3 ^{※1}
P32	I/O	端口 3	可以位选输入/输出方式 可以通过软件设置连接内部上拉电阻	输入	INTP1
P34 ^{※1}					输入
P40 和 P43 ^{※2}	I/O	端口 4 2 位 I/O 端口 可以位选输入/输出方式 可以通过软件设置连接内部上拉电阻		输入	-

- 注
1. 引脚功能的设置方法，请参见 第十五章 选项字节。
 2. 程序设定中，设置 PM41, PM42, 和 PM44 ~ PM47 为 "0"。

注意事项 引脚 P22/X2/ANI2 和 P23/X1/ANI3 在复位期间处于下拉状态。

- 备注**
1. 当高速内部振荡作为系统时钟时 P22 和 P23 可以分配其余用途。
 2. 当外部时钟输入作为系统时钟时 P22 可以分配其余用途。

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 (PM2 ~ PM4) 端口寄存器 (P2 ~ P4) 端口模式控制寄存器 2 (PMC2) 上拉电阻选择寄存器 (PU2 ~ PU4)
端口	总共: 8 (CMOS I/O: 7, CMOS 输入: 1)
上拉电阻	总共: 7

4.2.1 端口 2

端口 2 是一个带输出锁存的 4 位 I/O 端口。可通过端口模式寄存器 2 (PM2) 位选输入或输出模式。当用作输入端口时, 可通过上拉电阻选择寄存器 2 (PU2) 设置为使用内部上拉电阻。

该端口也可以作为内部 A/D 转换器的模拟信号输入口, 定时器 I/O 和外部中断请求信号输入口。

引脚 P22 和 P23 也被用作系统时钟振荡器的 X1 和 X2 引脚。引脚 P22 和 P23 的功能取决于系统时钟振荡器的选择。以下三种时钟振荡器能供选择。

(1) 高速内部时钟电路

引脚 P22 和 P23 能用作 I/O 端口引脚或 A/D 转换器的模拟输入引脚。

(2) 晶体/陶瓷振荡器

当引脚 P22 和 P23 用作 X1 和 X2 引脚时, 不能被用作 I/O 端口引脚或 A/D 转换器的模拟输入引脚。

(3) 外部时钟输入

引脚 P22 能用作 I/O 端口引脚或 A/D 转换器的模拟输入引脚。

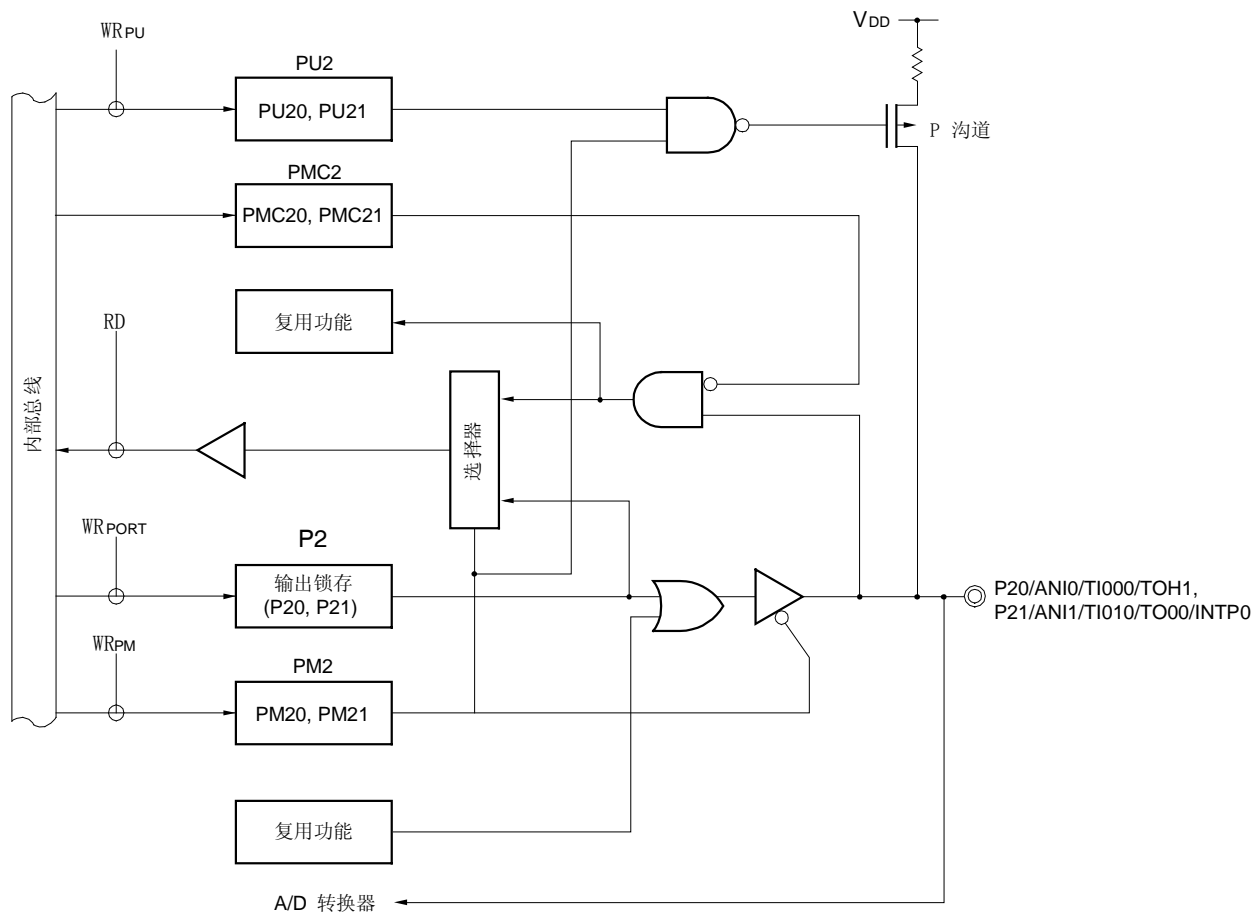
引脚 P23 被用作外部时钟输入引脚 X1 时, 不能被用作 I/O 端口引脚或 A/D 转换器的模拟输入引脚。

可通过选项字节选择系统时钟振荡器。细节描述请参见 **第十五章 选项字节**。

复位后端口 2 为输入模式。

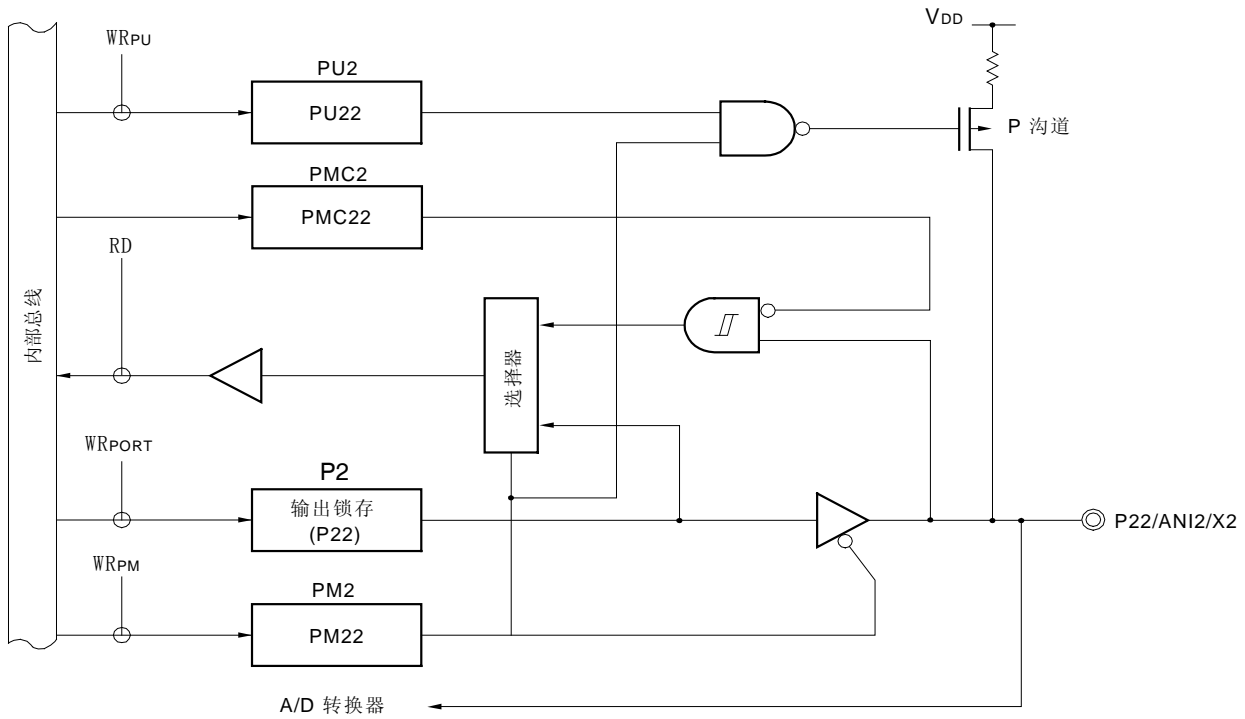
图 4-2 和 4-3 为端口 2 的框图。

图 4-2. P20 和 P21 的框图



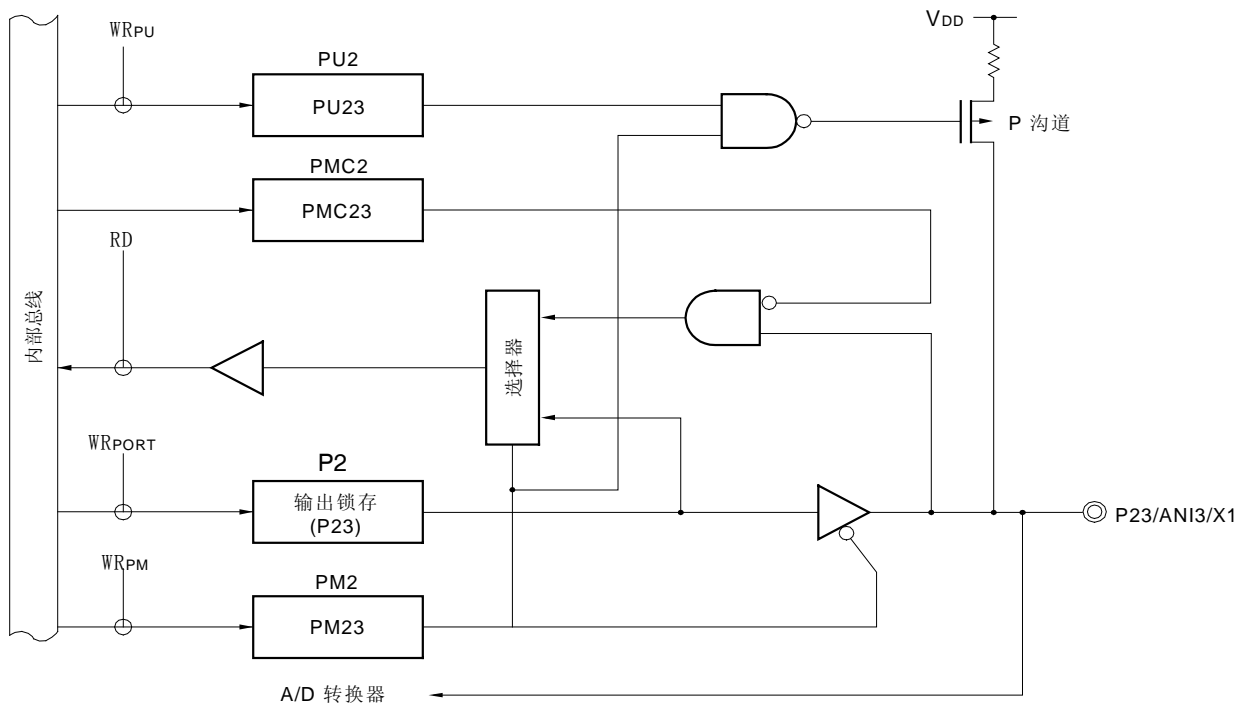
- P2: 端口寄存器 2
- PU2: 上拉电阻选择寄存器 2
- PM2: 断口模式寄存器 2
- PMC2: 端口模式控制寄存器 2
- RD: 读信号
- WR_{xx}: 写信号

图 4-3. P22 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

图 4-4. P23 的框图



- P2: 端口寄存器 2
- PU2: 上拉电阻选择寄存器 2
- PM2: 断口模式寄存器 2
- PMC2: 端口模式控制寄存器 2
- RD: 读信号
- WRxx: 写信号

4.2.2 端口 3

引脚 P32 是具有输出锁存的 1 位 I/O 端口。能通过端口模式寄存器 3 (PM3) 设置成输入或输出模式。当这个引脚被作为输入端口使用时，能通过上拉电阻选择寄存器 3 (PU3) 并以 1 位为单位设置连接上拉电阻。这个引脚也能被用作外部中断请求信号输入引脚。

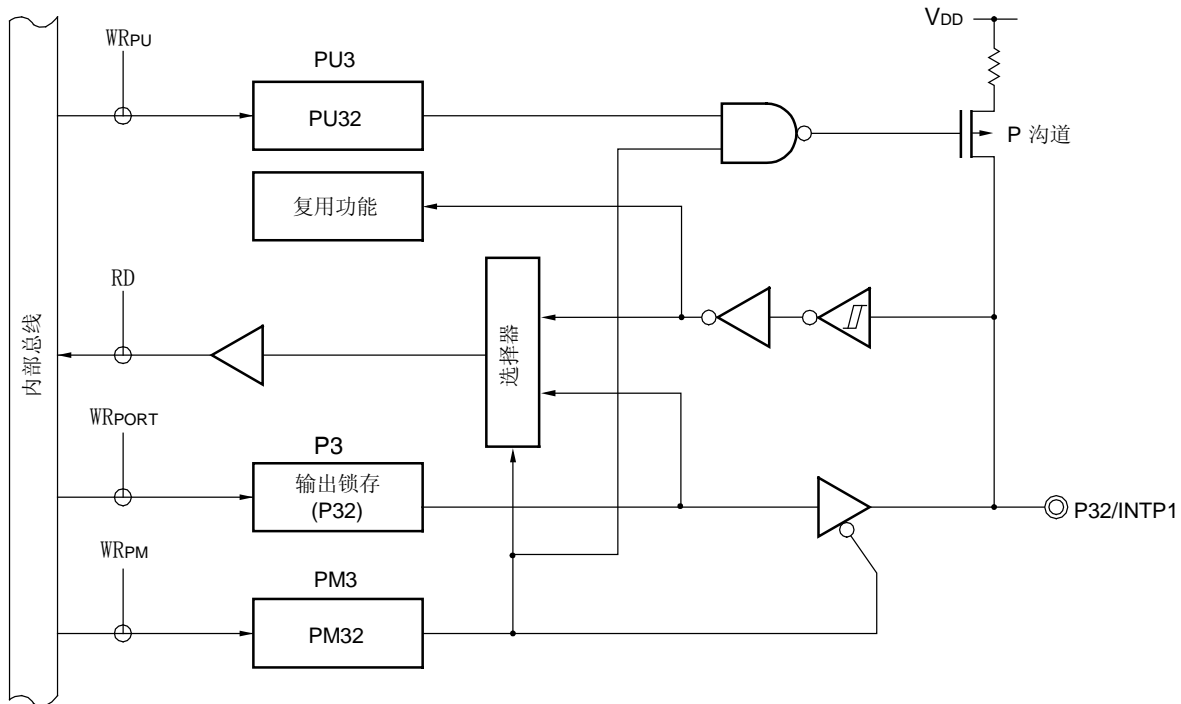
设置端口 3 为输入模式可以从引脚 P32 产生复位信号。

引脚 P34 仅能作为 1 位输入端口，可以作为 $\overline{\text{RESET}}$ 引脚，并且当电源上电时示复位功能。引脚功能的设定方法，请参见 第十五章 选项字节。

当 P34 用作输入端口引脚，连接到上拉电阻。

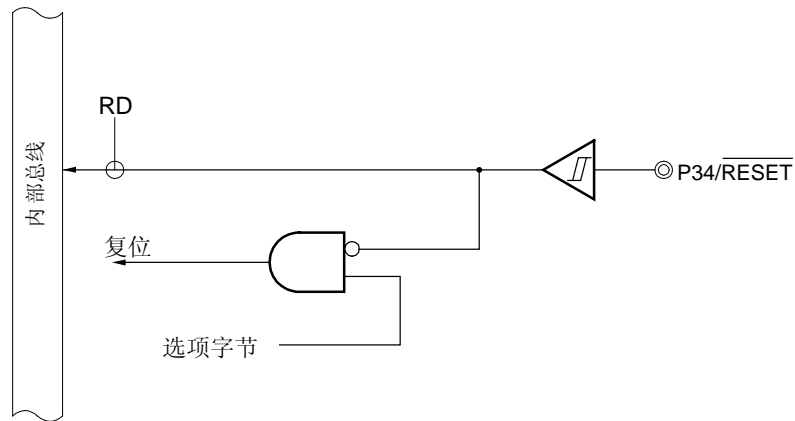
图 4-5 和 4-6 为端口 3 的框图。

图 4-5. P32 的框图



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

图 4-6. P34 的框图



RD: 读信号

注意事项 因为引脚 P34 复用为 $\overline{\text{RESET}}$ 引脚，如果当它作为输入引脚使用的时候，从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，具体细节请参照 第十五章 选项。如果复位状态释放后，再次参照选项字节之前，一个低电平被输入到 $\overline{\text{RESET}}$ 引脚，产生被复位并保持在复位状态直到一个高电平输入到 $\overline{\text{RESET}}$ 引脚。用作输入端口引脚时，连接到上拉电阻。

4.2.3 端口 4

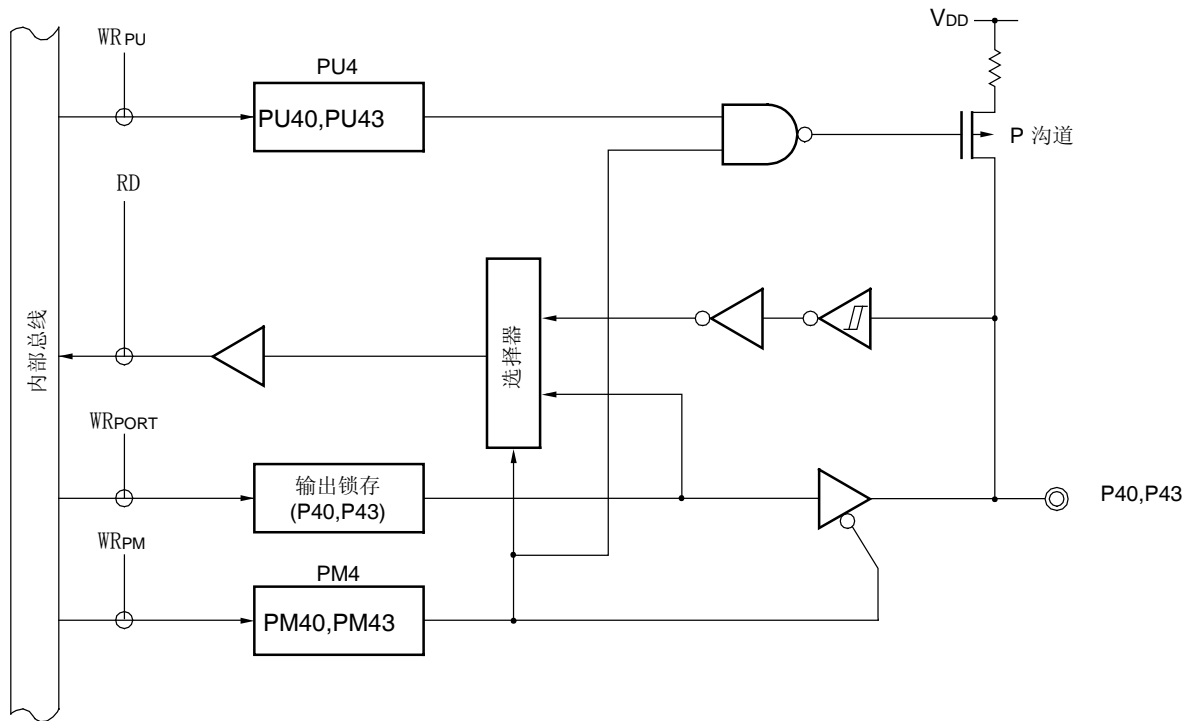
端口 4 是一个带输出锁存的 2 位 I/O 端口。端口的每一位都可以通过设置端口模式寄存器 4(PM4)^注设置成输入或者输出模式。当引脚 P40 和 P43 被当作输入端口使用时，可以以 1 位为单位设置上拉电阻选择寄存器 4(PU4)来连接片内上拉电阻。

复位时端口 4 设置为输入模式。

图 4-7 是端口 4 的框图。

注 程序设定中，设置 PM41, PM42, 和 PM44 ~ PM47 为 "0"。

图 4-7. P40 和 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

4.3 控制端口功能的寄存器

端口被下列四种寄存器所控制。

- 端口模式寄存器 (PM2 ~ PM4)
- 端口寄存器 (P2 ~ P4)
- 端口模式控制寄存器 2 (PMC2)
- 上拉电阻选择寄存器 (PU2 ~ PU4)

(1) 端口模式寄存器 (PM2 ~ PM4)

这些寄存器以 1 位单元来控制相应端口的输入输出模式。

可以由 1 位或者 8 位的操作指令设置这些寄存器。

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 因为引脚 P21 和 P32 同时也能当作外部中断引脚使用，当这些引脚被设置为输出模式并且输出电平被改变的时候，相应的中断请求标志也会被置起来。当要在输出模式下使用这些引脚的时候，应当提前把相应的中断屏蔽标志设置为“1”。

图 4-8. 端口模式寄存器的格式

地址: FF22H 复位后: FFH R/W

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

地址: FF23H 复位后: FFH R/W

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

地址: FF24H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM4	1 ^注	1 ^注	1 ^注	1 ^注	PM43	1 ^注	1 ^注	PM40

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

注 程序设定中，设置 PM41, PM42, 和 PM44 ~ PM47 为 "0"。

(2) 端口寄存器 (P2 ~ P4)

这些寄存器用于设置芯片端口要输出的数据。

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。

P20 ~ P23, P32, P40 ~ P47 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 4-9. 端口寄存器的格式

地址: FF02H 复位后: 00H (输出锁存) R/W

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

地址: FF03H 复位后: 00H^注 (输出锁存) R/W^注

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

地址: FF04H 复位后: 00H (输出锁存) R/W

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

Pmn	m = 2 ~ 4; n = 0 ~ 7	
	输出数据控制 (在输出模式下)	输入数据读取 (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)

这个寄存器指定了端口/复用模式 (A/D 转换功能除外) 或者 A/D 转换模式。

PMC2 寄存器的每一位和端口 2 的每一个引脚都是一一对应的并且可用 1 位控制。

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 4-10. 端口模式控制寄存器 2 的格式

地址: FF84H 复位后: R/W

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

PMC2n	指定的工作模式 (n = 0 ~ 3)
0	端口/复用功能 (A/D 转换功能除外) 模式
1	A/D 转换模式

注意事项 当 PMC20 ~ PMC23 设为 1, 不能使用 P20/ANI0 ~ P23/ANI23 的引脚功能。另外, 设置为 A/D 转换模式时确保设定上拉电阻选项寄存器(PU20 ~ PU23)为 0。

表 4-3. 使用复用功能的时候, 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		PM _{xx}	P _{xx}	PMC2n (n = 0 ~ 3)
	名称	I/O			
P20	ANI0	输入	1	×	1
	TI000	输入	1	×	0
	TOH1	输出	0	0	0
P21	ANI1	输入	1	×	1
	TI010	输入	1	×	0
	TO00	输出	0	0	0
	INTP0	输入	1	×	0
P22	ANI2	输入	1	×	1
P23	ANI3	输入	1	×	1
P32	INTP1	输入	1	×	—

备注 ×: 不考虑

PM_{xx}: 端口模式寄存器, P_{xx}: 端口寄存器 (端口具有输出锁存)

PMC2_x: 端口模式控制寄存器

(4) 上拉电阻选择寄存器 (PU2 ~ PU4)

这些寄存器规定是否使用引脚 P20 ~ P23, P32, P40 ~ P47 的内置上拉电阻。可以通过设置 PU2 ~ PU4, 来定义每一位端口的上拉电阻的使用。

PU2 ~ PU4 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 4-11. 上拉电阻选择寄存器的格式

地址: FF32H 复位后: 00H R/W

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

地址: FF33H 复位后: 00H R/W

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

地址: FF34H 复位后: 00H R/W

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

PU _m n	P _m n 引脚的上拉电阻寄存器的选择 (m = 2 ~ 4; n = 0 ~ 7)
0	不连接上拉电阻
1	连接上拉电阻

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，具体如下所示。

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

4.4.1 写入 I/O 端口

(1) 输出模式

通过传送指令可以把数据写入到输出锁存器中。输出锁存器中的数据通过引脚输出。在新的数据被写入到输出锁存器中以前，输出锁存器中的数据不会改变。

复位后清空输出锁存器中的数据。

(2) 输入模式

通过传送指令可以把数据写入到输出锁存中。因为输出缓冲已经关闭，引脚状态将不再被改变。

一旦数据被写入到输出锁存当中，在新的数据被写入到输出锁存器中以前，输出锁存中的数路不会改变。

复位后清空输出锁存中的数据。

4.4.2 从 I/O 端口读入

(1) 输出模式

可以通过操作指令读取输出锁存器中的数据，输出锁存器中的数据不会被改变。

(2) 输入模式

可以通过操作指令读取引脚状态，输出锁存器中的内容不会被改变。

4.4.3 I/O 端口的操作

(1) 输出模式

对输出锁存的内容执行一个操作，即对输出锁存器进行写操作。输出锁存器的内容从引脚输出。

一旦数据写入输出锁存器，它将一直保存到新数据写入。

复位后清空输出锁存器中的数据。

(2) 输入模式

读取引脚电平，对他的内容执行一个操作，则将引脚内容写入输出锁存，但是如果输出缓冲已经关闭，则引脚状态将不再被改变。

复位后清空输出锁存器中的数据。

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，一是供给 CPU 和外围硬件的时钟（系统时钟）产生电路，二是供给看门狗定时器和 8 位定时器 H1（TMH1）的时钟（间隔时间发生时钟）产生电路。

5.1.1 系统时钟振荡器

系统可以使用下列三种时钟振荡器。

- 高速内部振荡器
该电路可以产生 8MHz（典型值）的时钟信号，通过执行 STOP 指令使其振荡停止。
如果使用高速内部振荡器作为系统时钟，则 X1 和 X2 引脚可以作为输入/输出端口引脚使用。
- 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 1~10MHz。通过执行 STOP 指令使其振荡停止。
- 外部时钟输入电路
外部的集成电路产生的时钟信号，通过 X1 引脚输入单片机。输入振荡频率的范围为 1~10MHz。执行 STOP 指令后，单片机内部停止内部时钟信号。
当外部时钟输入作为系统时钟信号时，X2 引脚可以作为输入/输出引脚使用。

通过设置选项字节可以指定系统时钟源。详细请参见 **第十五章 选项字节**。

如 X1 和 X2 引脚作为输入/输出端口引脚使用时，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。通过对低速内部振荡模式寄存器（LSRCM）设置可以使振荡器停止工作。当设置选项字节时可以通过软件停止振荡器工作。

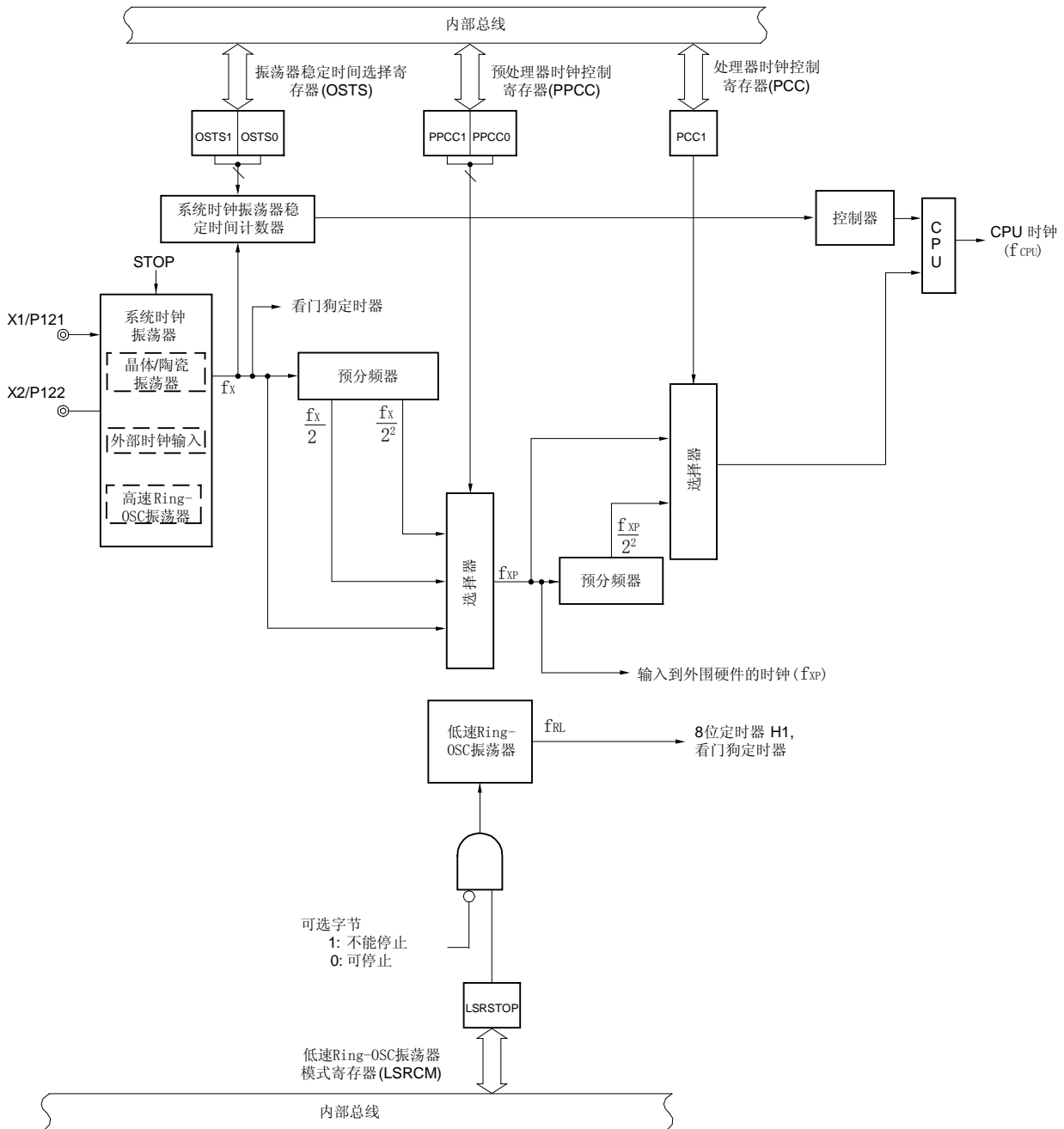
5.2 时钟发生器的配置

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 由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)

这两个寄存器控制系统时钟的分频比。

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 复位后状态: 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 复位后状态: 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 ^{‡1}
0	0	1	f _x /2 ²
1	0	0	f _x /2 ^{2‡2}
0	1	1	f _x /2 ^{3‡1}
1	0	1	f _x /2 ^{4‡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/KU1+最快的指令可以在两个 CPU 时钟周期内完成。因此，CPU 时钟和最小指令执行时间的关系如下表 5-2 所示。

表 5-2 CPU 时钟和最小指令执行时间的关系

CPU 时钟 (f_{CPU}) ^注	最小指令执行时间: $2/f_{CPU}$	
	高速内部振荡时钟 (8.0 MHz (典型值))	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 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) 的选择通过选项字节来控制。

(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。

当通过选项字节设置低速内部振荡器有效时, 可通过软件使低速内部振荡器停止工作。如果设置选项字节使低速内部振荡器不能通过软件来停止工作, 这种寄存器的设置是无效的, 低速内部振荡器继续工作。并且, WDT 的时钟源是由低速内部振荡器给出的, 详细请参见 第八章 看门狗定时器。

LSRCM 可以由 1 位或 8 位存储器操作指令操作。

复位后, LSRCM 的值为 00H。

图 5-4 低速内部振荡模式寄存器(LSRCM)的格式

地址: FF58H 复位后状态: 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
1	停止低速内部振荡

(3) 振荡稳定时间选择寄存器 (OSTS)

这个寄存器用于选择 **STOP** 模式释放时振荡器开始振荡的振荡稳定时间。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，不需要振荡稳定时间。

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的选择由选项字节决定。详情请参见 **第十五章 选项字节**。

OSTS 可以由 8 位存储器操作指令操作。

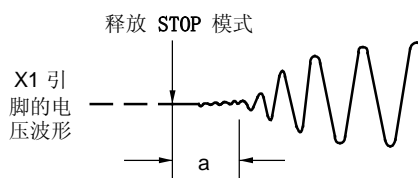
图 5-5 振荡稳定时间选择寄存器 (**OSTS**) 的格式

地址: FFF4H 复位后的状态: 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 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大于等于预期的振荡器的振荡稳定时间。
 2. 无论 **STOP** 模式是被复位信号打破还是被中断打破，**STOP** 模式释放后的等待时间不包括从释放 **STOP** 模式到晶振开始震荡的时间(下图中的 a)。



3. 上电或者复位后的振荡稳定时间可以通过选项字节来设置，详情参见 **第十五章 选项字节**。

- 备注**
1. (): $f_x = 10$ MHz
 2. 可通过查看所使用的振荡器的特性决定振荡器的振荡稳定时间。

5.4 系统时钟振荡器

系统时钟振荡器可以由下列三种振荡器构成。

- 高速内部振荡器: 工作在 8 MHz (典型值)。
- 晶体/陶瓷振荡器: 工作在 1 ~ 10 MHz。
- 外部时钟输入电路: 由 X1 引脚输入 1 ~ 10 MHz 时钟信号。

5.4.1 高速内部振荡器

78K0S/KU1+ 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 引脚可以作为输入/输出端口引脚使用。

关于选项字节的详细情况请参见 第十五章 选项字节。关于输入/输出端口的详细情况, 请参见 第四章 端口功能。

5.4.2 晶体/陶瓷振荡器

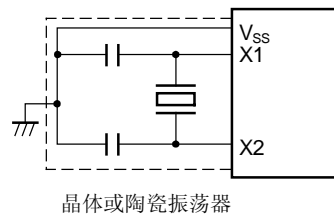
晶体/陶瓷振荡器通过与 X1 和 X2 引脚相连的晶体或陶瓷振荡器产生振荡时钟信号。

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。

关于选项字节的详细情况请参见 第十五章 选项字节。关于输入/输出端口的详细情况, 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



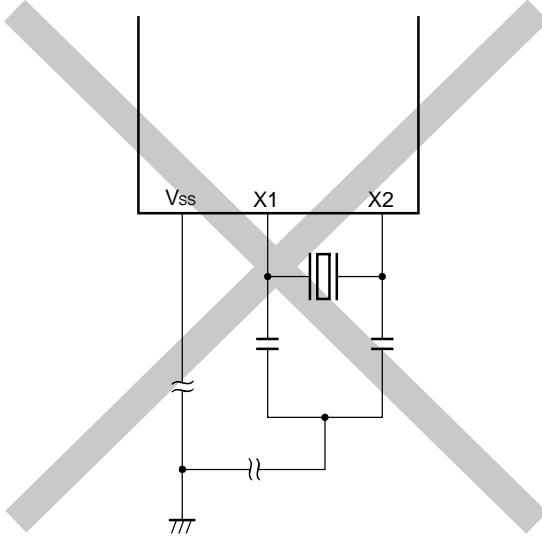
注意事项 当使用晶体/陶瓷振荡器时, 布线应该按照图 5-6 中虚线内的方式连接, 这样可以减小分布电容对振荡器的影响。

- 保持连线尽可能的短。
- 不要使其他的信号线与振荡器导线互相交叉, 不要在振荡器旁边通过大电流的信号线。
- 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。

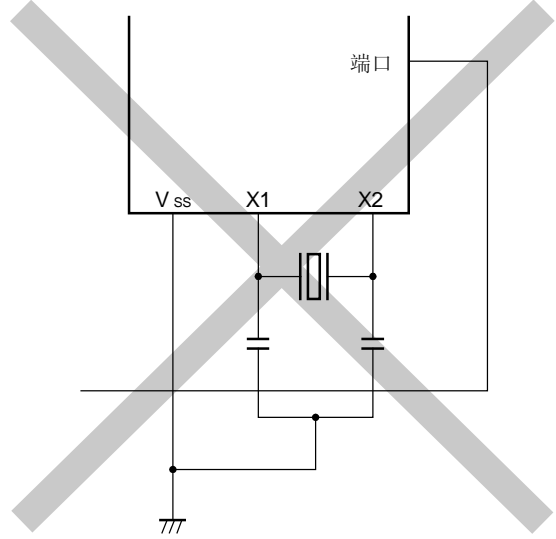
图 5-7 显示了振荡器错误连接方式的示例。

图 5-7 振荡器错误连接方式的示例 (1/2)

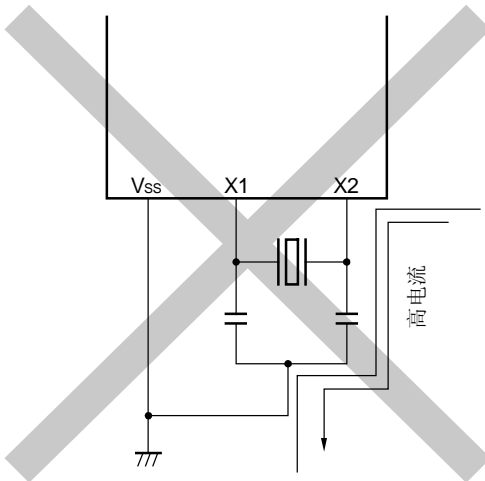
(a) 连接线太长



(b) 信号线与振荡器导线互相交叉



(c) 附近流过大的电流



(d) 大电流通过振荡器的地线 (A, B 和 C 电位波动)

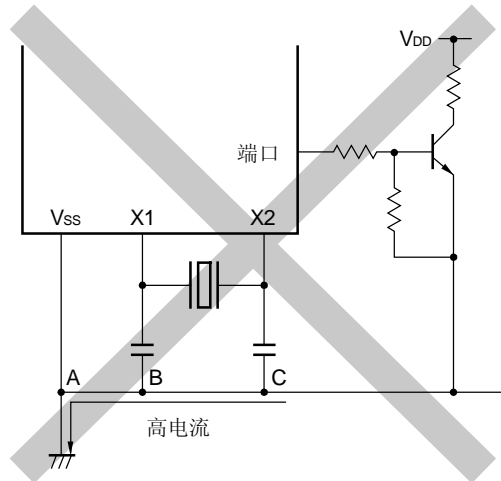
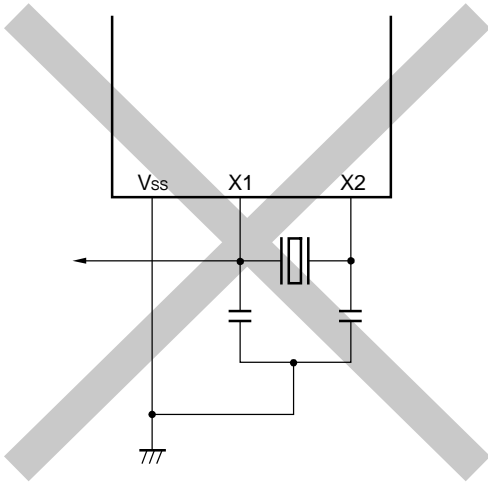


图 5-7 振荡器错误连接方式的示例 (2/2)

(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 引脚可以作为输入/输出端口引脚使用。

关于选项字节的详细情况请参见 **第十五章 选项字节**。关于输入/输出端口的详细情况，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(fx)输出分频后产生一个时钟(fxP)给外围硬件，也能将外围硬件的时钟(fxP)分频后作为 CPU 时钟。

备注 时钟输出是由选项字节选择的振荡器（高速内部振荡器，晶体/陶瓷振荡器或外部时钟输入电路）分频得到。关于选项字节的详细情况请参见 **第十五章 选项字节**。

5.5 CPU 时钟发生器的操作

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

- 高速内部振荡器: 工作在 8 MHz (典型值)。
- 晶体/陶瓷振荡器: 工作在 1 ~ 10 MHz。
- 外部时钟输入电路: 由 X1 引脚输入 1 ~ 10 MHz 时钟信号。

系统时钟源由选项字节控制。关于选项字节的详细情况请参见 **第十五章 选项字节**。

(1) 高速内部振荡器

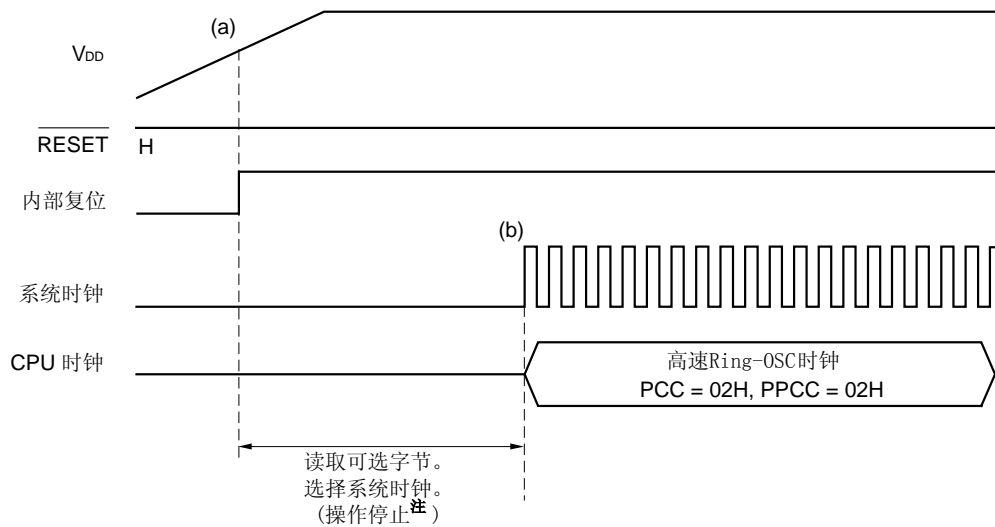
当通过选项字节选择高速内部振荡器时, 情况如下:

- 较短的启动时间
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, 启动时间缩短了。
- 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 引脚就可以作为输入/输出端口引脚使用。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下图 5-8, 5-9 所示。

备注 当使用高速内部振荡器时, 振荡器的精确度为 $\pm 5\%$ 。

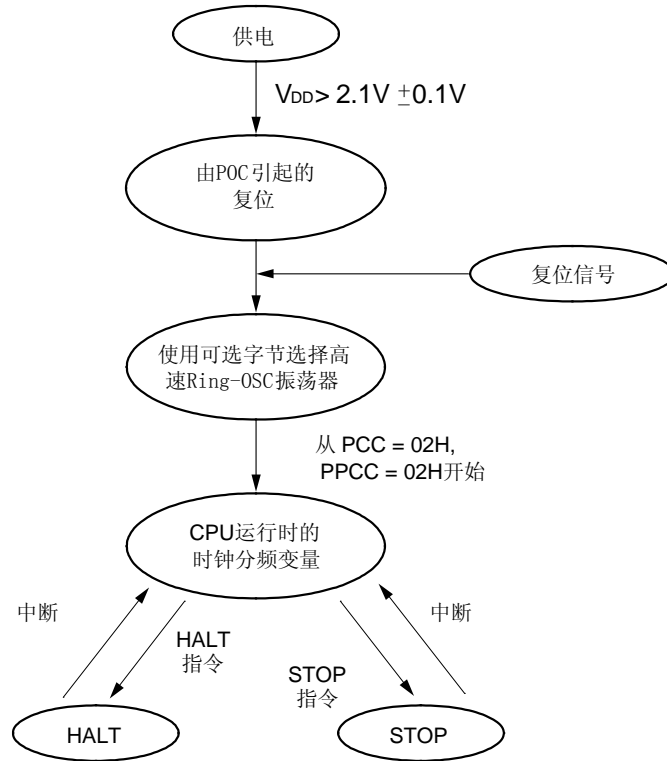
图 5-8 默认由高速内部振荡器启动时的时序图



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) 通过上电时的上电清零功能产生内部复位信号，复位后参照选项字节，选择系统时钟。
- (b) 参照选项字节，选择系统时钟。然后，高速内部时钟作为系统时钟工作。

图 5-9 默认由高速内部振荡器启动时的状态转换框图



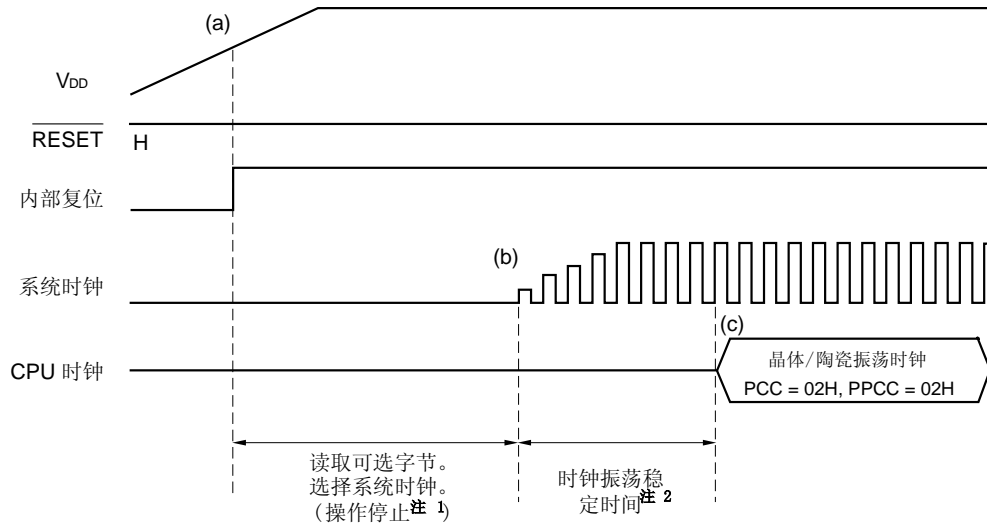
备注 PCC: 处理器时钟控制寄存器
PPCC: 预处理器时钟控制寄存器

(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 1~10MHz 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/陶瓷振荡器启动时的时序图和状态转换框图如下图 5-10 和 5-11 所示。

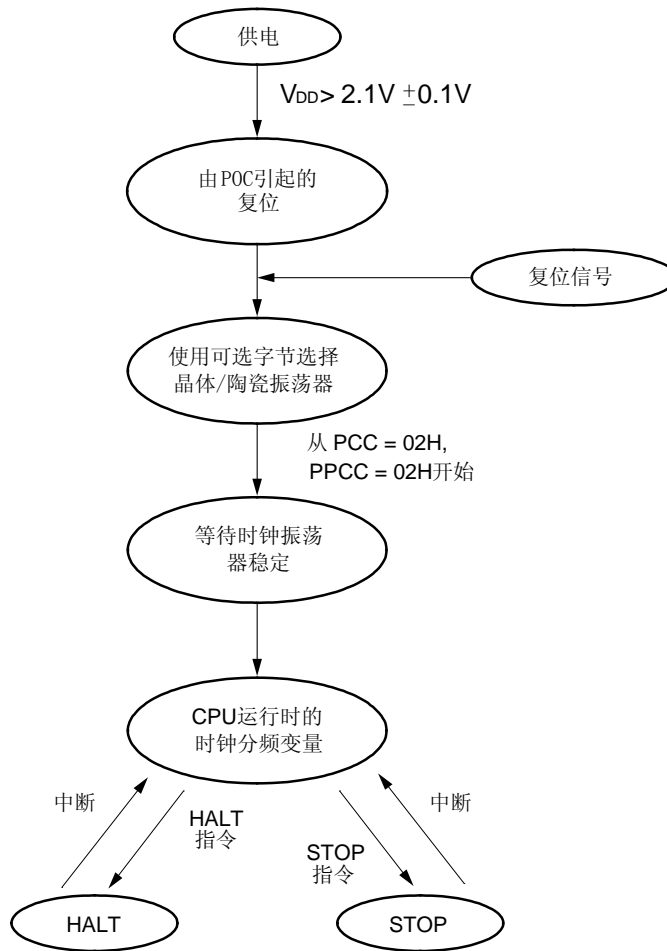
图 5-10 默认由晶体/陶瓷振荡器启动时的时序图



- 注
1. 操作停止时间为：277 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。详细情况请参见 第十五章 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) 通过上电时的上电清零功能产生内部复位信号，复位后参照选项字节，选择系统时钟。
- (b) 在高速内部时钟产生之后，参照选项字节，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。
- (c) 如果选择晶体/陶瓷振荡器为系统时钟，在时钟振荡稳定过后，它作为 CPU 时钟开始工作。等待时间通过选项字节选择。详细情况请参见 第十五章 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，情况如下。

- 高速操作

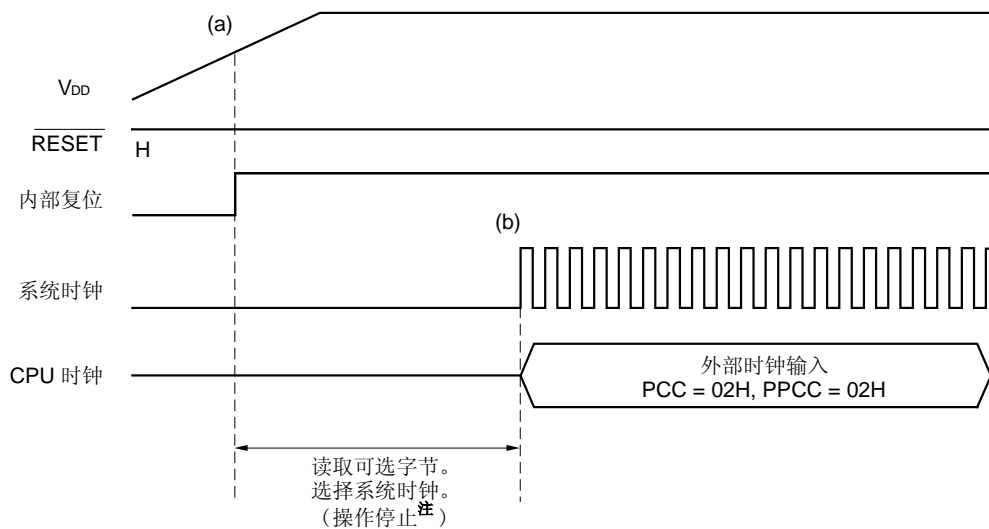
因为外部时钟频率可在 1~10MHz 之间的选择，并且外部时钟的频率偏差较小，所以同高速内部振荡器 (8MHz(典型值))相比，外部时钟输入具有较高的精确度。

- 扩展性提高

如果选择外部时钟输入作为振荡器，则 X2 引脚就可以作为输入/输出端口引脚使用。详情请参见 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下图 5-12 和 5-13 所示。

图 5-12 默认由外部时钟输入启动时的时序图

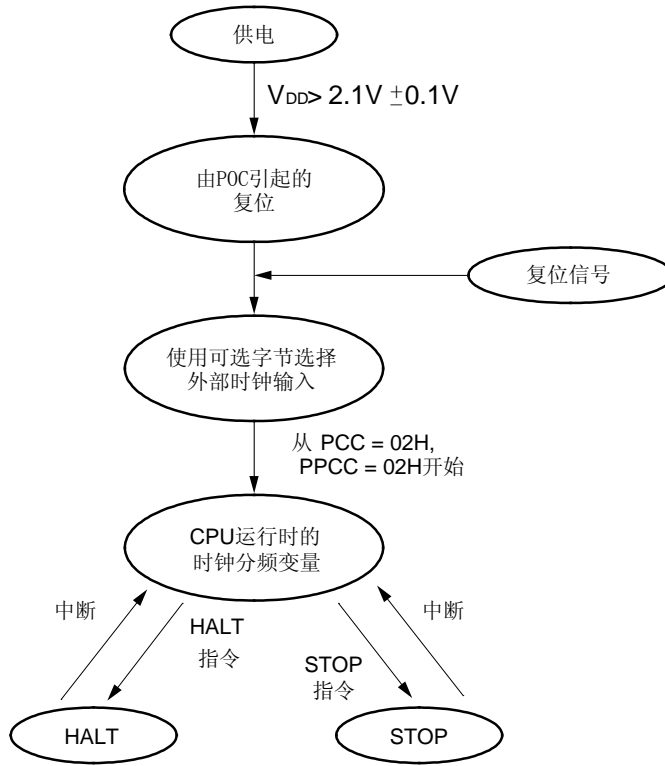


注 操作停止时间为：277 μ s(最小值)，544 μ s(典型值)，1.075ms(最大值)。

(a) 通过上电时的上电清零功能产生内部复位信号，复位后参照选项字节，选择系统时钟。

(b) 参照选项字节，选择系统时钟。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

下列两种时钟信号供给外围硬件使用。

- 外围硬件时钟信号 (f_{xP})
- 低速内部时钟 (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。

外围硬件时钟可以使用如下三种时钟频率： f_x ， $f_x/2$ ， $f_x/4$ 。表 5-3 列举了外围硬件时钟的配置情况。

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

(2) 低速内部时钟

在复位释放后，用作产生间隔时间时钟振荡器的低速内部振荡器开始工作，振荡频率为 240 kHz(典型值)。

可由选项字节指定低速内部振荡器是否能用软件停止。如果设置为通过软件可以停止低速内部振荡器工作，则振荡器可以通过低速内部振荡模式控制寄存器(LSRM)控制启动或停止。如果设定为不能用软件停止低速内部振荡器，WDT 的时钟信号源只能是低速内部时钟(f_{RL})。

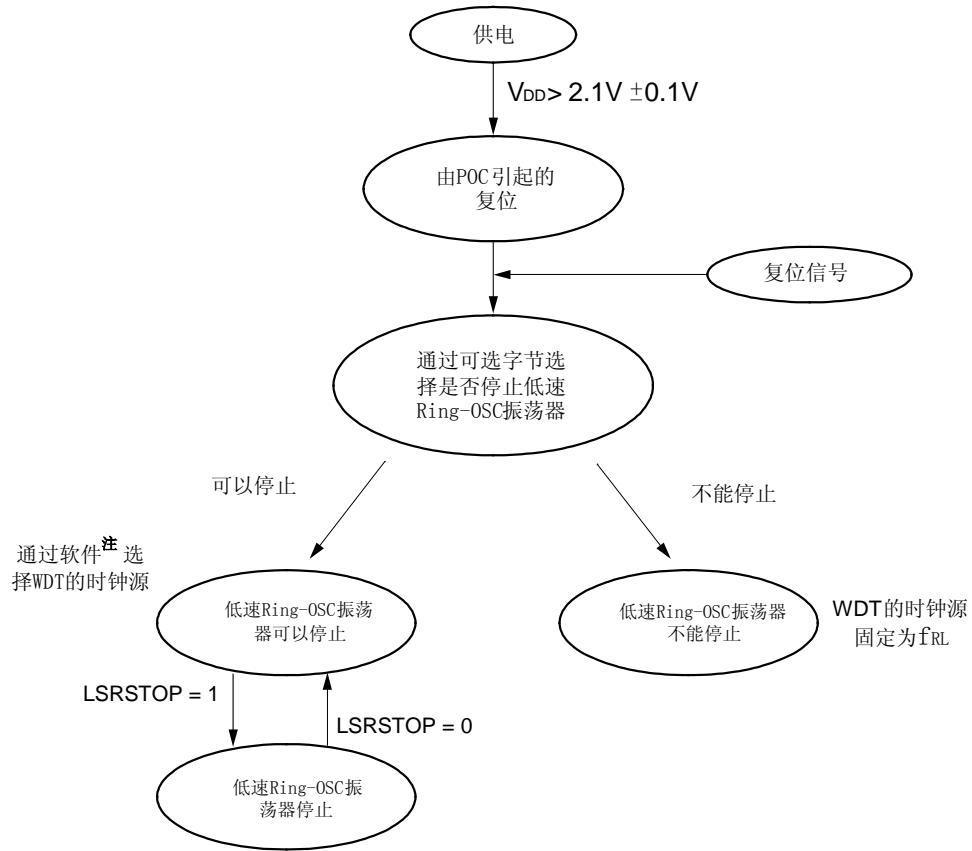
低速内部振荡器时钟独立于 CPU 的时钟。因此，如果用它来作为 WDT 的时钟源，即使 CPU 时钟停止，也可检测到 CPU 挂起。如果把它用作为 8 位定时器 H1 的时钟源，则 8 位定时器 H1 可以在待机状态下工作。

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
可以通过软件设置 停止工作	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(WDT)的时钟源可以为 f_x 或 f_{RL} ，或者它可能停止。详情请参见 第八章 看门狗定时器。

6.1 16 位定时器/事件计数器 00 的功能

16 位定时器/事件计数器 00 有如下功能。

(1) 间隔定时器

16 位定时器/事件计数器 00 以预置时间间隔产生中断请求。

- 计数的范围: $2 \sim 65536$

(2) 外部事件计数器

16 位定时器/事件计数器 00 可以通过有效沿来确定高/低电平宽度来测量外部输入的有效脉冲个数。

- 有效脉冲宽度为: $16/f_{XP}$ 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 可以输出任选频率的方波。

- 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 可以输出任选频率的方波。

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

(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

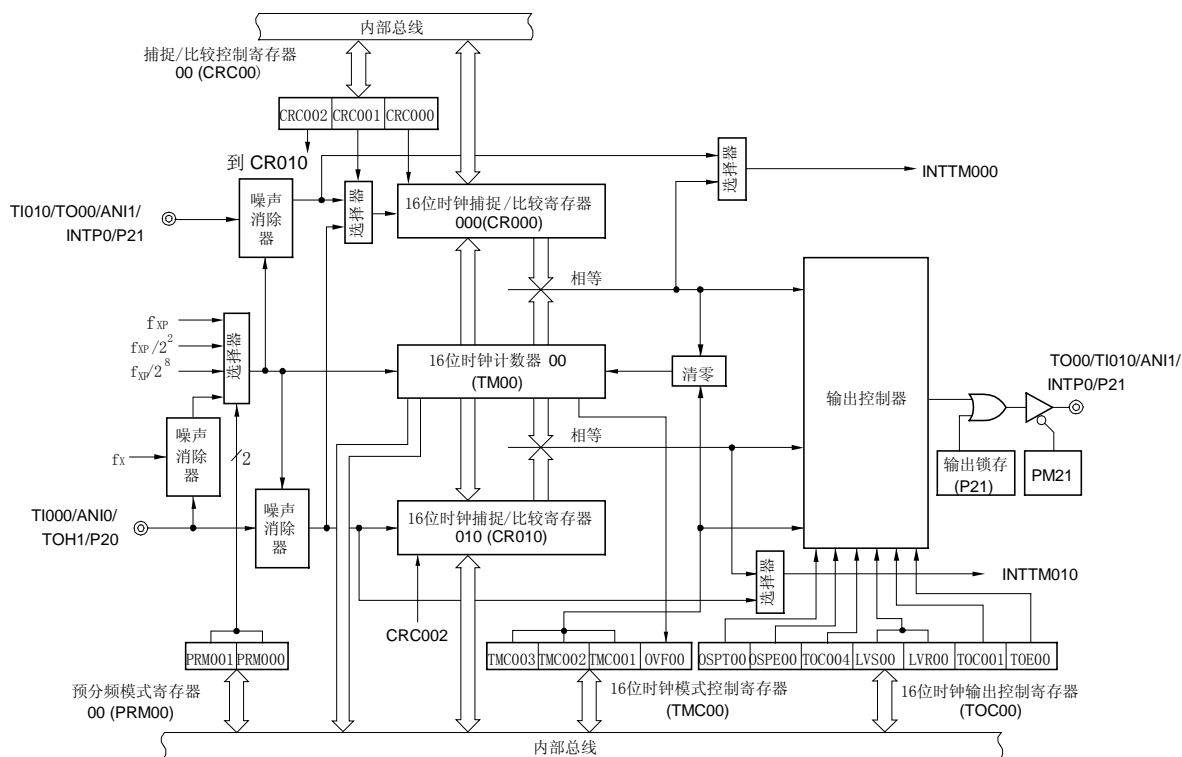
16 位定时器/事件计数器 00 由如下硬件组成。

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 位定时器捕捉/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) 捕捉/比较控制寄存器 00 (CRC00) 16 位定时器输出控制寄存器 00 (TOC00) 预分频模式寄存器 00 (PRM00) 端口模式寄存器 2 (PM2) 端口寄存器 2 (P2) 端口模式控制寄存器 2 (PMC2)

图 6-1 为这些计数器的框图。

图 6-1. 16 位定时器/事件计数器 00 框图

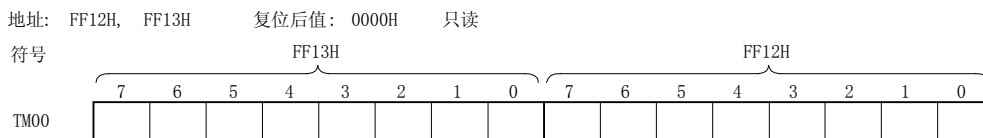


(1) 16 位定时器计数器 00 (TM00)

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



如下情况中计数值置为 0000H。

- <1> 复位输入时
- <2> TMC003 和 TMC002 清零时
- <3> 在清零&启动模式中，如果输入 TI000 的有效沿时
- <4> 在清零&启动模式中，当 TM00 和 CR000 相等时
- <5> 如果 OSPT00 在单脉冲输出模式下置 1

注意事项 1. TM00 被读取时, CR010 无法捕捉其值。

- 2. 如果在定时器计数时参考 TM00，在参考进程中定时器时钟会停止，并且在参考进程结束后定时器时钟恢复计数。

因此，如果进程需要在 TM00 执行时参考它，定时器计数将会发生一个错误。

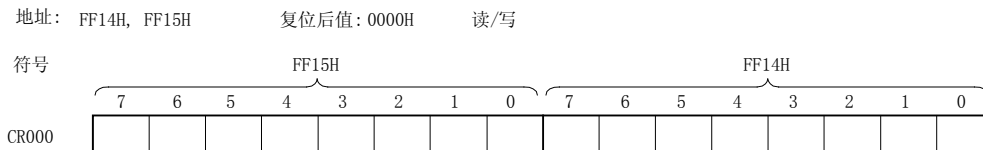
(2) 16 位定时器捕捉/比较寄存器 000 (CR000)

CR000 是 16 位寄存器，具有捕捉寄存器和比较寄存器的功能。无论是作为比较寄存器使用还是作为捕捉寄存器使用，均由捕捉/比较控制寄存器 00(CRC00)的第 0 位(CRC000)来设置。

CR000 由 16 位操作指令设置。

复位信号产生后，CR000 清零。

图 6-3. 16 位定时器捕捉/比较寄存器 000 (CR000)的格式



- **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，当值相等时产生中断请求 (INTTM000)。当设置 TM00 作为间隔定时器时，也可以作为保存间隔时间的寄存器使用。

- 当 CR000 作为捕捉寄存器使用

可选择 TI000 或 TI010 引脚的有效沿作为捕捉触发。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 捕捉触发和 TI000 与 TI010 引脚的有效沿

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

CR000 捕捉触发	TI000 引脚有效沿	ES010	ES000
		下降沿	上升沿
上升沿	下降沿	0	0
无捕捉操作	上升和下降沿均为有效沿	1	1

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

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

备注

- 禁止设置 ES010, ES000 = 1, 0 和 ES110, ES100 = 1, 0。
- ES010, ES000: 预分频模式寄存器 00 (PRM00) 的第 5 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: 捕捉/比较控制寄存器 00 (CRC00) 的第 1 位和第 0 位。

注意事项

- TM00 和 CR000 相等时进入的清零/启动模式中, CR000 置非 0000H 值。当此寄存器作为外部事件计数器使用时, 不能实现仅对一个脉冲的计数。然而, 在自由运行模式和由 TI000 引脚有效沿产生的清零&启动模式中, 如果 CR000 置 0000H, TM00 溢出(FFFFH)后从 0000H 变为 0001H 时, 产生一个中断请求 (INTTM000)。
- 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。
- 当 16 位定时器/事件计数器 00 停止工作后, CR000 的值是不确定的。
- 当 CR000 设为比较模式, 即使有捕捉触发信号输入, 也不响应捕捉操作。
- 当 P21 引脚作为 TI010 有效沿输入端口使用时, P21 引脚则不能作为定时器输出(TO00)使用。而且, 当 P21 用作定时器输出(TO00)时, 也不能作为 TI010 有效沿输入引脚使用。
- 当 CR000 作为捕捉寄存器使用, 如果同时发生寄存器的读操作和输入捕捉触发操作, 则捕捉触发输入有优先权, 而读出的数据为不确定值。同样, 如果定时器停止计数与捕捉触发输入同时发生, 捕捉触发为不确定状态。
- 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 (17) 在定时器操作中改变比较寄存器的注意事项。

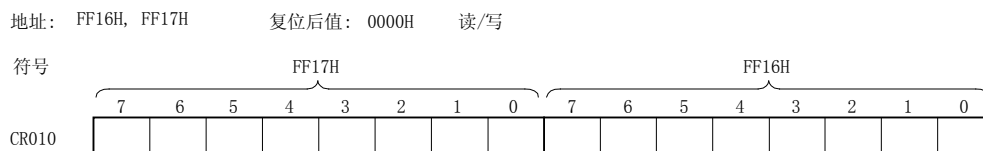
(3) 16位捕捉/比较寄存器 010 (CR010)

CR010 是可以作为捕捉寄存器和比较寄存器使用的 16 位寄存器。无论是作为比较寄存器使用还是作为捕捉寄存器使用，均由捕捉/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。

复位后，CR010 值为 0000H。

图 6-4. 16 位定时器捕捉/比较寄存器 010 (CR010)的格式



- 当 CR010 用于比较寄存器

CR010 中的设置值与 16 位定时器计数器 00 (TM00) 的计数值不断比较，相等时产生中断请求 (INTTM010)。

- 当 CR010 用于捕捉寄存器

可选择 TI000 引脚的有效沿作为捕捉触发。TI000 的有效沿由预分频模式寄存器 00 设置 (PRM00) (参考表 6-3)。

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

CR010 捕捉触发	TI000 引脚有效沿		
	ES010	ES000	
下降沿	下降沿	0	0
上升沿	上升沿	0	1
上升和下降沿均触发	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: 捕捉/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 引脚有效沿产生的清零&启动模式中，如果 CR010 置 0000H，TM00 溢出(FFFFH)后从 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 新值小于原来的值，则必须在改变赋值后重新启动定时器。
 3. 当 16 位定时器/事件计数器 00 停止工作后，CR010 的值是不确定的。
 4. 当 CR010 设为比较模式，即使有捕捉触发信号输入，也不响应捕捉操作。
 5. 当 CR010 作为捕捉寄存器使用，如果同时发生寄存器的读操作和输入捕捉触发操作，则捕捉触发输入有优先权，而读出的数据为不确定值。同样，如果定时器停止计数与捕捉触发输入同时发生，捕捉触发为不确定状态。

注意事项 6. 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 (17) 在定时操作中改变比较寄存器的注意事项。

6.3 16 位定时器/事件计数器 00 的控制寄存器

如下寄存器用于控制 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)
- 捕捉/比较控制寄存器 00 (CRC00)
- 16 位定时器输出控制寄存器 00 (TOC00)
- 预分频模式寄存器 00 (PRM00)
- 端口模式寄存器 2 (PM2)
- 端口寄存器 2 (P2)
- 端口模式控制寄存器 2 (PMC2)

(1) 16 位定时器模式控制寄存器 00 (TMC00)

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

TMC00 由 1 位或 8 位操作指令设置。

复位信号产生后, TMC00 值为 00H。

注意事项 当 TMC002、TMC003 不等于 0, 0 时, 16 位定时器计数器 00 (TM00) 开始工作, 而当 TMC002 和 TMC003 等于 0, 0 时, 停止操作。

图 6-5. 16 位定时器模式控制寄存器 00 (TMC00)的格式

地址: FF60H 复位后值: 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 电平反转条件	中断请求发生
0	0	0	操作停止 (TM00 清零)	无反转	无中断产生
0	0	1			
0	1	0	自由运行模式	当 TM00 和 CR000 相等或 TM00 和 CR010 相等时反转	<当作为比较寄存器时> 当 TM00 和 CR000 相等或 TM00 和 CR010 相等时产生中 断请求<当作为捕捉寄存器时> 在 TI000 引脚和 TI010 引脚有 效沿产生中断请求
0	1	1		当 TM00 和 CR000 相等; TM00 与 CR010 相等; TI000 引脚有效沿时反转	
1	0	0	在 TI000 引脚有效沿上发生清 零& 启动	-	
1	0	1			
1	1	0	当 TM00 和 CR000 相等时发生 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; TM00 与 CR010 相等; 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	无溢出检测
1	检测溢出

- 注意事项**
- 除了 OVF00 标志位, 其他位写入数据时, 定时器必须停止操作。
 - 如果定时器停止, 即使 TI000/TI010 引脚有信号输入, 定时器计数和中断也不响应。
 - 除非当 TI000 引脚有效沿用于计数时钟, 在设置定时器 STOP 模式或系统时钟停止之前, 必须停止定时器工作, 否则, 当系统时钟启动时定时器可能产生错误。
 - 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 位设置 TI000 引脚有效沿。
 - 在 TM00 和 CR00 相等时进入的清零&启动模式中, 当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 后, OVF00 标志位置 1。
 - 即便在 TM00 溢出后而下一个计数时钟计数前 (TM00 变为 0001H 前) 做清除 OVF00 标志操作, OVF00 标志还会被重新置 1, 清除操作无效。
 - 捕捉操作在计数时钟的下降沿执行, 中断请求 (INTTM0n0) 在计数时钟的上升沿产生。

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

(2) 捕捉/比较控制寄存器 00 (CRC00)

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

CRC00 由 1 位或 8 位操作指令设置。

复位后, CRC00 值为 00H。

图 6-6. 捕捉/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	作为捕捉寄存器

CRC001	CR000 捕捉触发选择
0	TI010 引脚有效沿时捕捉触发
1	当 TI000 引脚有效沿反向 ^注 时捕捉触发 ^注

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

注 如果上升沿和下降沿都被选为引脚 TI000 的有效沿, 不执行捕捉操作。

- 注意事项
1. 设置 CRC00 前, 定时器必须停止操作。
 2. 当通过 16 位定时器模式控制寄存器 TMC00 选择为 TM00 和 CR000 相等进入的清零&启动模式时, CR000 就不能被指定为捕捉寄存器。
 3. 为了确保捕捉操作的稳定, 被捕捉的信号宽度要大于 2 个计数时钟(由预分频器模式寄存器 00 (PRM00)选择)的长度才能触发捕捉 (参见图 6-18)。

(3) 16 位定时器输出控制寄存器 00 (TOC00)

此寄存器用于控制 16 位定时器/事件计数器输出的操作。可以设置定时器的输出 F/F set/reset, 输出反转的允许/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, 和通过软件设置的单脉冲输出触发。

TOC00 由 1 位或 8 位存储器操作指令设置。

复位后, 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	单脉冲输出控制							
0	连续脉冲输出模式							
1	单脉冲输出模式 ^注							
TOC004	当 CR010 与 TM00 相等时定时器输出 F/F 控制位							
0	禁止反转操作							
1	允许反转操作							
LVS00	LVR00	定时器输出 F/F 状态设置						
0	0	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(1)						
1	1	禁止设置						
TOC001	当 CR000 与 TM00 相等时定时器输出 F/F 控制位							
0	禁止反转操作							
1	允许反转操作							
TOE00	定时器输出控制							
0	禁止输出 (输出置低电平)							
1	允许输出							

注 单脉冲输出模式只能在自由运行模式和 TI000 引脚作为输入有效沿的清零&开始模式下使用。在 TM00 和 CR000 相等时进入的清零&开始模式下, 由于溢出不会发生, 所以单脉冲输出不可能发生。

- 注意事项**
- 除了 OSPT00 以外, 对其它寄存器设置之前, 定时器应停止工作。
 - 读 LVS00 和 LVR00 时, 读出的数据为 0。
 - 数据被设置后, OSPT00 自动清零, 读出的内容是 0。
 - 除了单脉冲输出模式以外, 不要将 OSPT00 设置为 1。
 - 当连续设置 OSPT00 位时, 需要两个或更多计数时钟周期(由预分频器模式寄存器 00 (PRM00)选择的写入间隔)。

注意事项 6. 当 TOE00 为 0 时, 设置 TOE00, LVS00, 和 LVR00 要用 8 位操作指令同时设置。 当 TOE00 为 1 时, LVS00, LVR00 能够用 1 位操作指令设置。

(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。

复位后, PRM00 值为 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	上升沿和下降沿均为有效沿

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	上升沿和下降沿均为有效沿

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 [*]

备注 1. f_{XP} : 供给外围设备的时钟振荡频率

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

注 外部时钟要求一个比两个内部计数时钟(f_{XP})周期长的脉冲。

注意事项 1. 停止定时器工作后, 向 PRM00 写入数据。

2. 如果 TI000 引脚的有效沿被用于计数时钟, 就不能设置清零启动模式和 TI000 引脚有效沿的捕捉。

注意事项 3. 在下列情况下, 注意 TI0n0 引脚有效沿被检测的警告。

- <1> 系统复位后, 如果一个高电平输入 TI0n0 引脚, 16 位定时器计数器 00 (TM00) 允许操作。
 - 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> 当 TI0n0 为高电平, TM00 停止工作时, TI0n0 引脚如输入一个低电平则 TM00 开始工作。
 - 如果 TI0n0 的下降沿或上升沿和下降沿作为有效沿, 那么 TM00 操作使能后, 下降沿会立即检测到。
 - <3> 当 TI0n0 为低电平, TM00 停止工作时, TI0n0 引脚如输入一个高电平则 TM00 开始工作。
 - 如果 TI0n0 的上升沿或上升沿和下降沿作为有效沿, 那么 TM00 操作使能后, 上升沿会立即检测到。
4. 当 TI000 的有效沿用做计数时钟, 和当 TI000 用于捕捉触发器时, 用于消除噪声的采样时钟是不同的, 对于前者, 由 f_{XP} 做计数时钟, 对于后者, 通过 PRM00 选择的计数时钟。直到采样到有效沿并且连续两次检测到有效电平, 捕捉操作才会完成, 因此可消除小脉冲宽度的噪声。
5. 当 P21 端口作为有效沿的输入引脚(TI010)使用时, 它不能同时作为定时器输出引脚(TO00)使用。当 P21 端口作为定时器输出引脚(TO00)使用时, 它不能同时作为有效沿的输入引脚(TI010)使用

备注 n = 0, 1

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

当引脚 P21/TO00/TI010/ANI1/INTP0 作为定时器输出使用时, 将 PM21, P21 的输出锁存器和 PMC21 清零。

当引脚 P20/TI000/TOH1/ANI0 和引脚 P21/TO00/TI010/ANI1/INTP0 作为定时器输入使用时, 设置 PM20 和 PM21 为 1, 并将 PMC20 和 PMC21 清零。

此时, P20 和 P21 的输出锁存器可以为 1 或 0。

PM2 和 PMC2 可以由 1 位或 8 位存储器操作指令设置。

复位输入设置 PM2 为 FFH, 设置 PMC2 为 00H。

图 6-9. 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 (输出缓冲器开)
1	输入模式 (输出缓冲器关)

图 6-10. 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	工作模式选择 (n = 0 ~ 3)
0	端口/复用功能模式 (A/D 转换器除外)
1	A/D 转换器模式

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) 和捕捉/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 6-11 所示。

设置

基本操作的设置步骤如下。

- <1> 设置 CRC00 寄存器(见图 6-11)。
- <2> 设置 CR000 寄存器任意值。
- <3> 通过使用 PRM00 寄存器设置计数时钟。
- <4> 设置 TMC00 寄存器开始定时器工作 (见图 6-11)。

注意事项 当 TM00 工作时, 改变 CR000 寄存器的设置可能导致错误。要改变设置, 请参考 6.5 16 位定时器/计数器 00 (17) 在定时器操作过程中改变比较寄存器的相关注意事项

备注 关于怎样允许 INTTM000 中断, 请参见 第十章 中断功能。

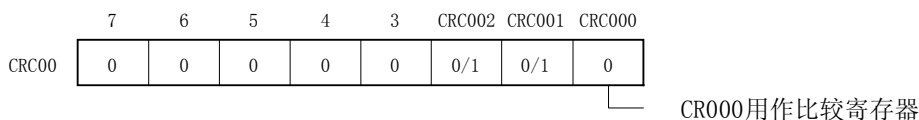
以 16 位定时器捕捉/比较寄存器 000 (CR000) 中预先设定的值为间隔重复产生中断请求。

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)。

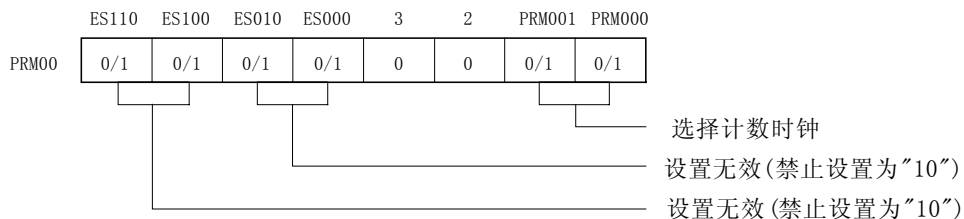
16 位定时器/计数器的计数时钟可以通过预分频模式寄存器 PRM00 的第 0, 1 位 (PRM000,PRM001) 设置。

图 6-11. 设置控制寄存器用于时间间隔定时器操作

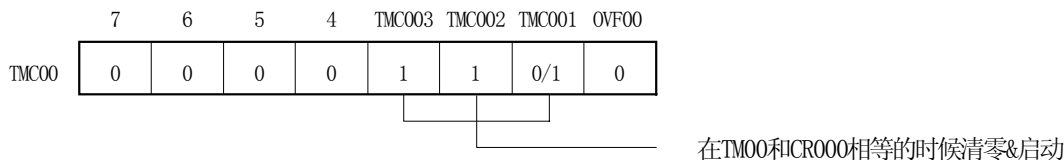
(a) 捕捉/比较控制寄存器 00 (CRC00)



(b) 预分频模式寄存器 00 (PRM00)

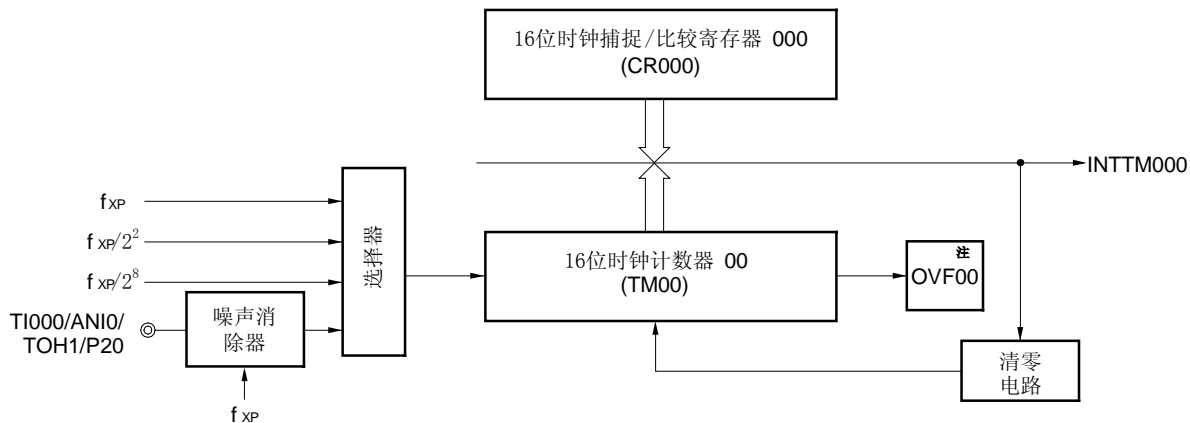


(c) 16位定时器模式控制寄存器 00 (TMC00)



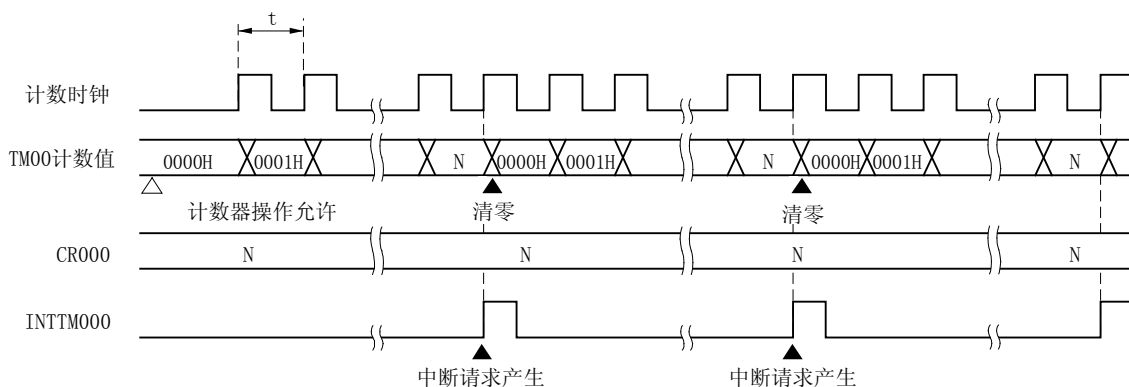
备注 0/1: 设置 0 或 1 可允许其他功能与间隔定时器同时使用。需要了解详细信息，参见各自控制寄存器的描述。

图 6-12. 间隔定时器配置框图



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

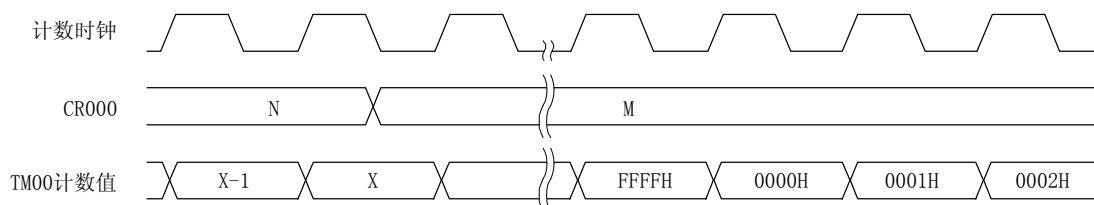
图 6-13. 间隔定时器操作的时序



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 位定时器捕捉/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, 需要重启定时器。

图 6-14. 在定时器计数操作过程中比较寄存器改变之后的时序(N → M: N > M)



备注 $N > X > M$

6.4.2 外部事件计数器操作

设置

基本操作的设置步骤如下。

- <1> 设置 CRC00 寄存器(见 图 6-15)。
- <2> 通过 PRM00 寄存器设置计数时钟。
- <3> 设置 CR000 寄存器(0000H 不能被设置)。
- <4> 设置 TMC00 寄存器开始定时器工作(见 图 6-15)。

备注 1. 设置 TI000 引脚, 请参见 6.3 (5) 端口模式寄存器 2 (PM2) 和端口模式控制寄存器 2 (PMC2)。
 2. 如何使能 INTTM000 中断, 请参见 第十章 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。

每当由预分频模式寄存器 00(PRM00)定义的有效沿输入时，TM00 递增。

当 TM00 的计数值与 16 位定时器捕捉/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)。

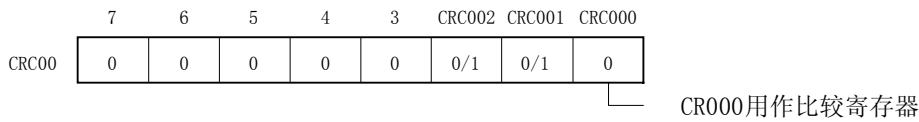
CR000 不能赋值为 0000H(脉冲的计数操作不能完成)。

通过对预分频模式寄存器 PRM00 的第 4, 5 位(ES000,ES010)的设置选择上升沿，下降沿或上升沿和下降沿为有效沿。

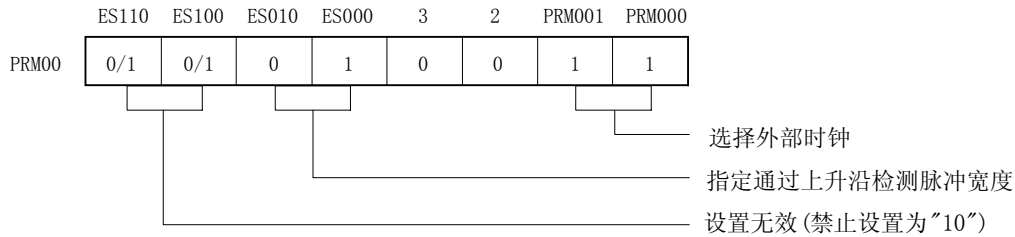
因为仅当使用内部时钟(f_{XP})采样之后检测到两次 TI000 引脚的有效沿时，操作才能进行，所以短脉冲宽度的噪声可以被消除。

图 6-15. 外部事件计数器模式控制寄存器的配置 (指定上升沿)

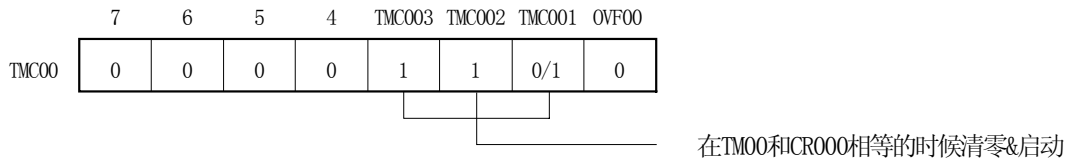
(a) 捕捉/比较控制寄存器 00 (CRC00)



(b) 预分频模式寄存器 00 (PRM00)

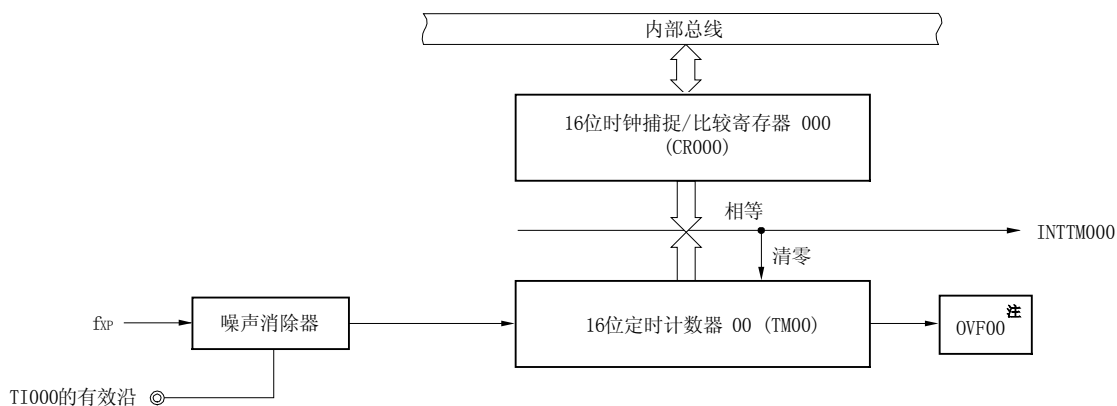


(c) 16 位定时器模式控制寄存器 00 (TMC00)



备注 0/1: 设置 0 或 1 可允许其他功能与间隔定时器同时使用。需要了解详细信息，参见各自控制寄存器的描述。

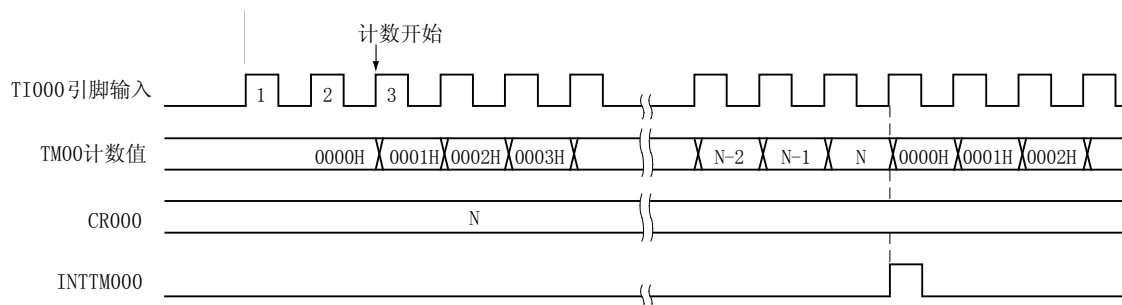
图 6-16. 外部事件计数器配置框图



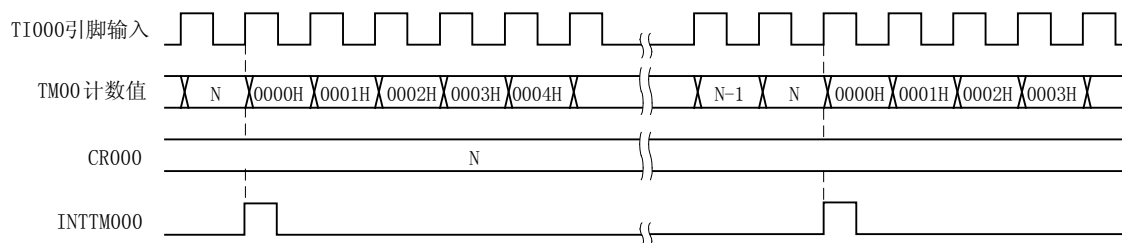
注 只有当 16 位捕获/比较寄存器 000 中的值为 FFFFH 时，OVF00 标志位置 1。

图 6-17. 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 时序
有效沿被检测到两次之后计数开始



(2) INTTM000 产生两次之后的时序



注意事项 当读取外部事件计数器的计数值时，TM00 即为其值。

6.4.3 脉冲宽度测量操作

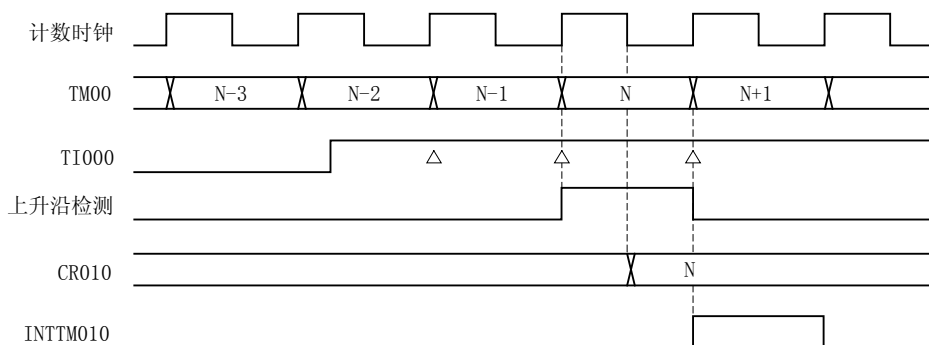
通过 16 位定时器计数器 00 (TM00) 可测量输入 TI000 和 TI010 引脚的信号脉冲宽度。

有两种方法可以测量脉冲宽度：一是在自由运行模式下，通过 TM00 测量；二是与输入到 TI000 引脚的信号边沿同步启动定时器测量。

当发生中断时，读捕捉寄存器的有效值，检测溢出标志位，然后计算脉冲宽度。检测后，溢出标志位清零。

直到在由预分频模式寄存器 00 (PRM00) 选择的计数时钟周期内采样到信号脉冲宽度，且对 TI000 或 TI010 的有效电平检测过两次之后，才执行捕捉操作，因此可以消除短脉冲宽度的噪音。

图 6-18. 指定上升沿的 CR010 捕捉操作



设置

基本操作的设置步骤如下：

- <1> 设置 CRC00 寄存器 (见 图 6-19, 6-22, 6-24 和 6-26)。
- <2> 通过 PRM00 寄存器设置计数时钟。
- <3> 设置 TMC00 寄存器启动操作 (见 图 6-19, 6-22, 6-24 和 6-26)。

注意事项 如要使用两个捕捉寄存器，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 2 (PM2) 和端口模式控制寄存器 2 (PMC2)。
 2. 如何使能 INTTM000 (或 INTTM010) 中断, 参见 第十章 中断功能。

(1) 用自由运行计数器和捕获寄存器测量脉宽

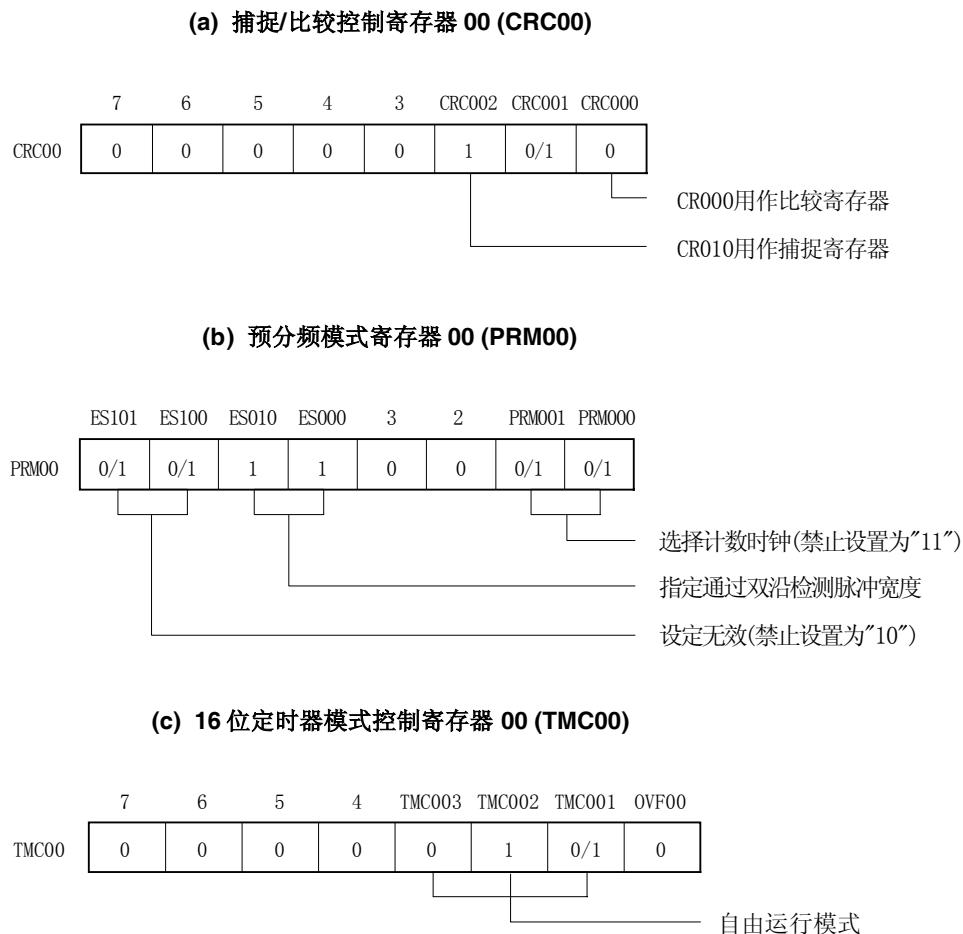
根据 PRM00 的第 4 位和第 5 位 (ES000 and ES010) 指定 TI000 引脚的上升沿和下降沿同为有效沿。

如果在自由运行模式下操作 16 位定时器计数器 00 (TM00)，且将预分频模式寄存器 00 (PRM00) 规定的脉冲沿输入到 TI000 引脚时，TM00 的值被送入 16 位定时器捕捉/比较寄存器 010 (CR010)，并产生外部中断请求信号 (INTTM010)。

使用 PRM00 选择的计数时钟进行采样，并且仅当 TI000 引脚的有效电平检测到两次才执行捕捉操作，因此可以消除短脉冲宽度的噪音。

注意事项 在这个操作例子中的脉冲宽度测量上升到定时器计数器的一个周期。

图 6-19. 通过自由运行计数器和一个捕捉寄存器测量脉冲宽度的控制寄存器的设置 (当使用 TI000 和 CR010 时)



备注 0/1:设置 0 或 1 可允许另外一个功能与间隔定时器同时使用。需要了解详细信息，参见各自控制寄存器的描述。

图 6-20. 通过自由运行计数器测量脉冲宽度的配置框图

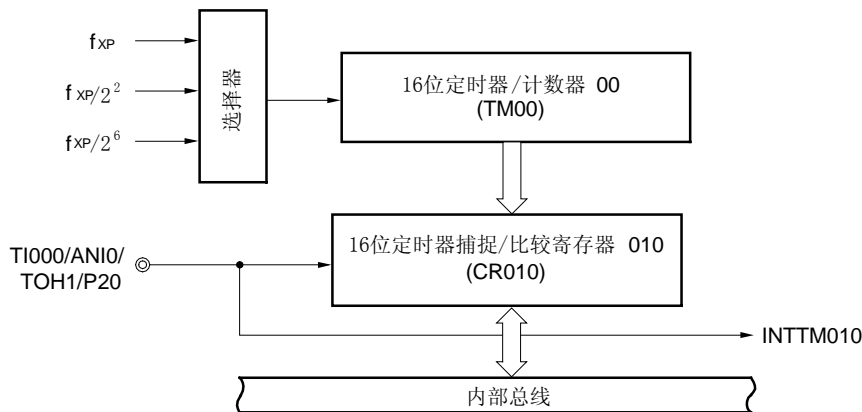
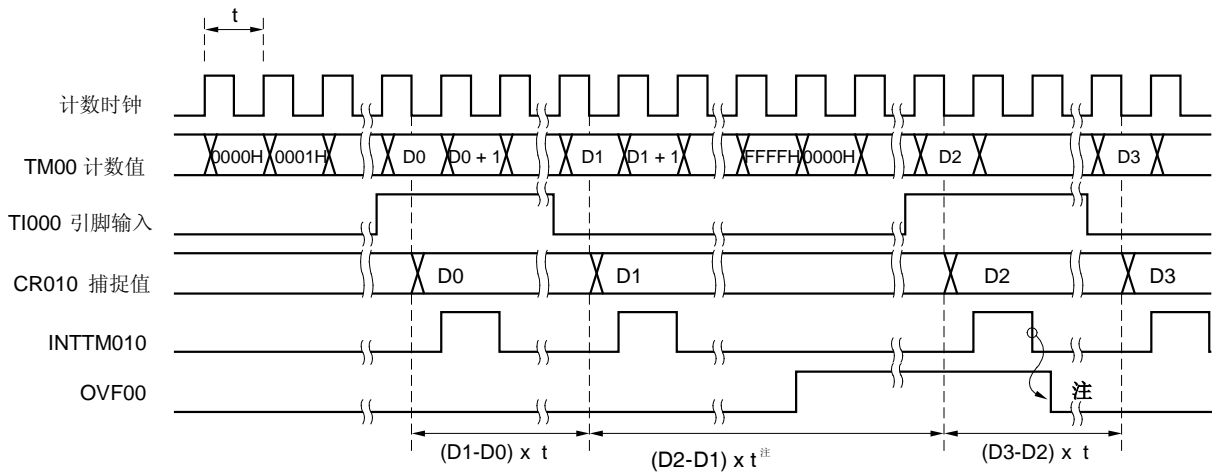


图 6-21. 使用自由运行计数器和一个捕捉寄存器进行脉冲宽度测量的时序（指定上升沿和下降沿）



注 忽略这个设置，进位标志设为 1。

(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。

PRM00 的第 4, 5 位 (ES000,ES010) 和第 6, 7 位 (ES100,ES110) 可以定义 TI000 和 TI010 引脚的上升沿和下降沿都为有效沿。

当有效沿 (由 PRM00 的第四位和第五位(ES000 和 ES010)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 CR010，然后 INTTM010 中断产生。

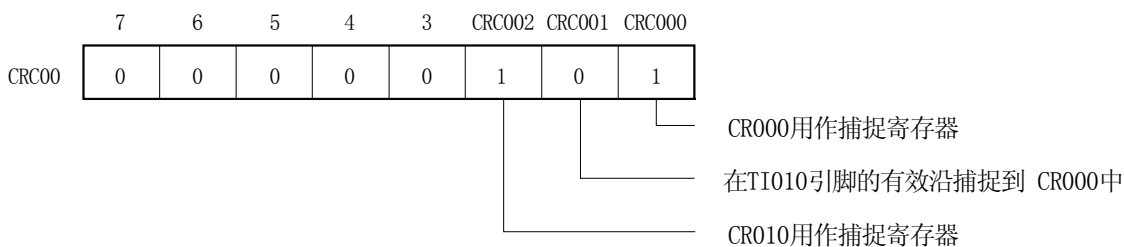
另外，当有效沿 (由 PRM00 的第六位和第七位(ES100 和 ES110)定义) 输入到 TI010 引脚时，TM00 的值写入 16 位定时器捕获/比较寄存器 CR000，然后 INTTM000 中断产生。

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，并且对 TI000 或 TI010 的有效电平检测过两次之后才可执行捕捉操作，因此可以消除短脉冲宽度的噪音。

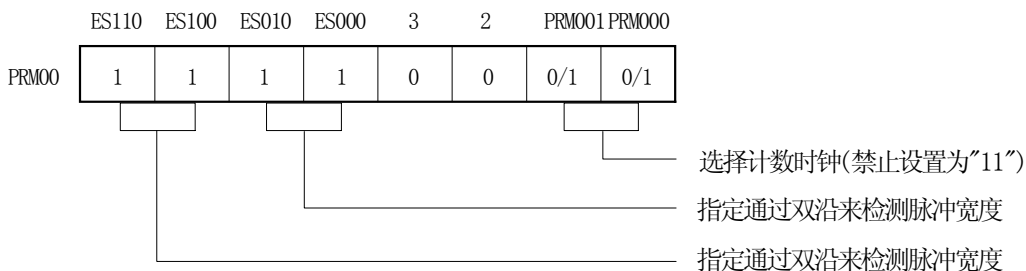
注意事项 在这个操作例子中的脉冲宽度测量上升到定时器计数器的一个周期。

图 6-22. 使用自由运行计数器设置控制寄存器用于两个脉冲宽度测量

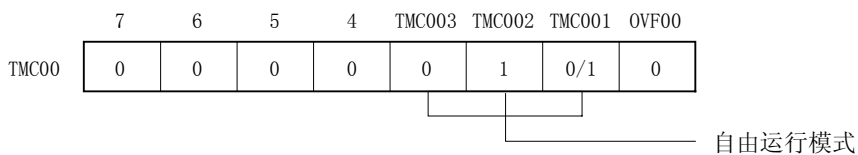
(a) 捕捉/比较控制寄存器 00 (CRC00)



(b) 预分频模式寄存器 00 (PRM00)

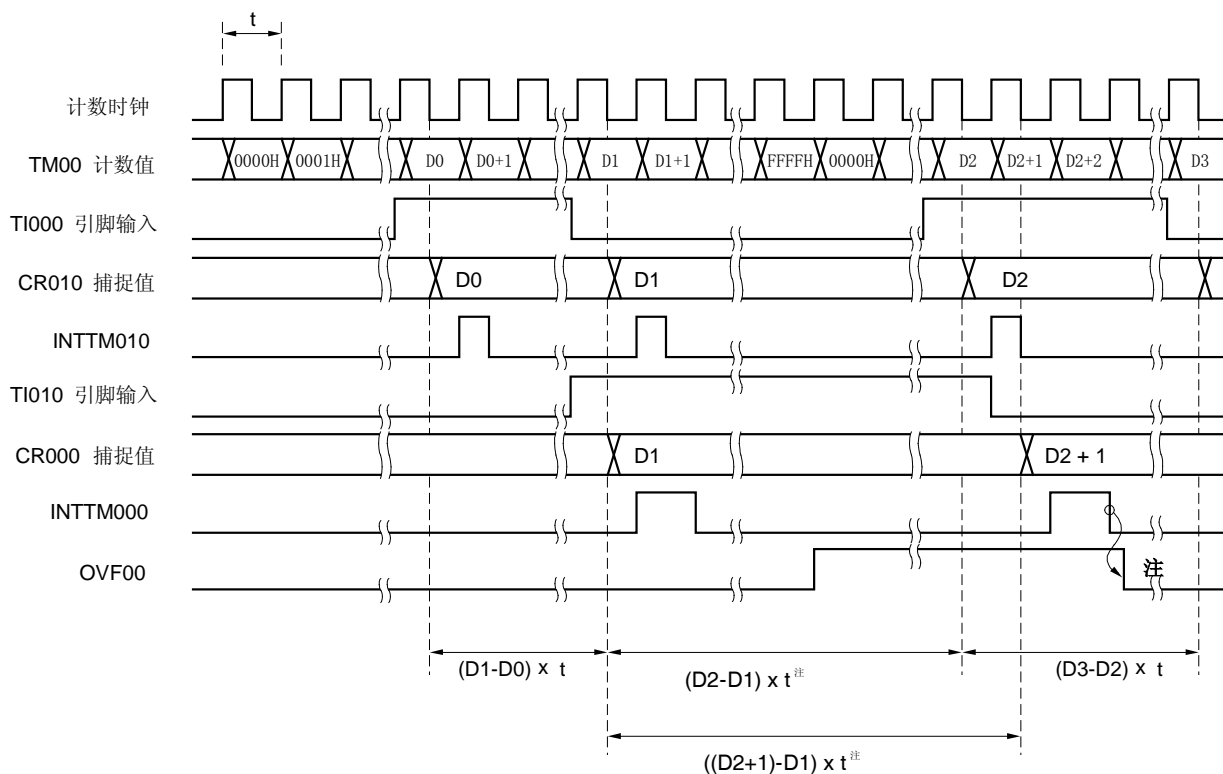


(c) 16 位定时器模式控制寄存器 00 (TMC00)



备注 0/1:设置 0 或 1 可允许另外一个功能与间隔定时器同时使用。需要了解详细信息，参见各自控制寄存器的描述。

图 6-23. 使用自由运行计数器进行脉冲宽度测量的时序（指定上升沿和下降沿）



注 忽略这个设置，进位标志设为 1。

(3) 使用自由运行计数器和两个捕捉寄存器进行脉冲宽度测量

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

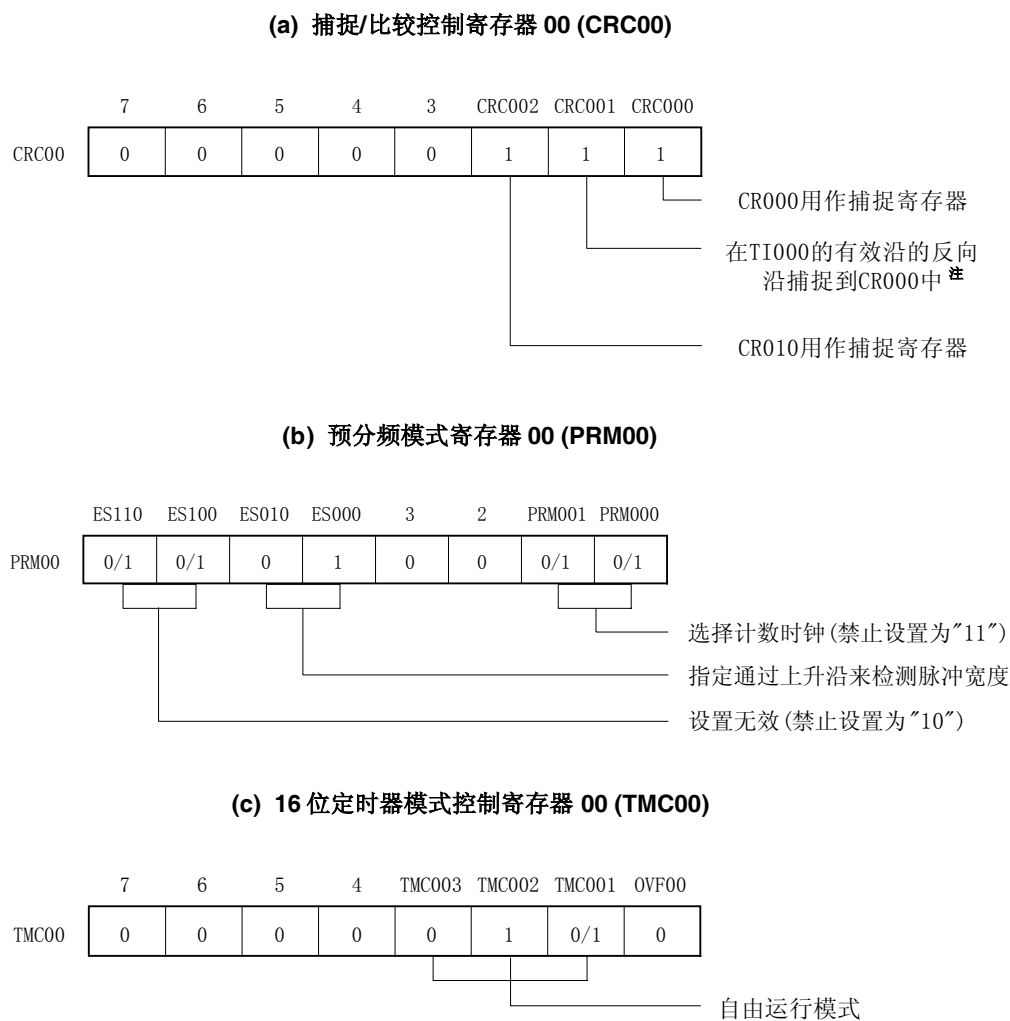
当上升沿或者下降沿（由 PRM00 的第四位和第五位(ES000 和 ES010)定义）输入到 TI000 引脚时，TM00 的值写入 16 位定时器捕捉/比较寄存器 CR010，然后 INTTM010 中断产生。

另外，当捕捉 CR010 的有效沿的反向沿输入时，TM00 的值被送入 16 位定时器捕捉/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，并且对 TI000 的有效电平检测过两次之后才可执行捕捉操作，因此可以消除短脉冲宽度的噪音。

注意事项 在这个操作例子中的脉冲宽度测量上升到定时器计数器的一个周期。

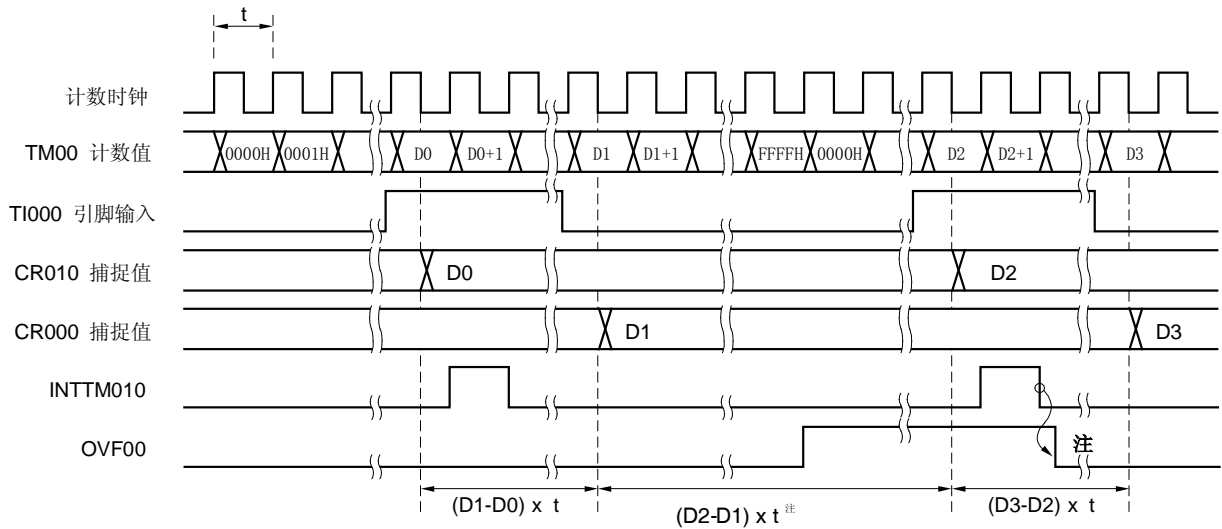
图 6-24. 设置控制寄存器，使用自由运行计数器和两个捕捉寄存器进行两个脉冲宽度的测量（指定上升沿有效）



注 如果指定 TI000 引脚的上升沿和下降沿同时为有效沿，16 位定时器捕捉/比较寄存器 0000(CR000) 不能完成捕捉操作。当 CRC001 位值为 1，TI010 引脚的有效沿被检测时，在 CR000 寄存器中 TM00 的计数值不能被捕捉，但是 TI010 引脚的输入可以作为外部中断源，可以产生 INTTM000 中断。

备注 0/1:设置 0 或 1 可允许另外一个功能与间隔定时器同时使用。需要了解详细信息，参见各自控制寄存器的描述。

图 6-25. 使用自由运行计数器和两个捕捉寄存器进行脉冲宽度测量的时序（指定上升沿有效）



注 忽略这个设置，进位标志设为 1。

(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 and ES010) 可以指定 TI000 引脚的上升沿和下降沿同为有效沿。

当检测有输入到 TI000 引脚的有效沿时，将 16 位定时器计数器 00(TM00)的计数值送入 16 位定时器捕捉/比较寄存器 010 (CR010)，然后通过对 TM00 清零和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，并且对 TI000 的有效电平检测过两次之后才可执行捕捉操作，因此可以消除短脉冲宽度的噪音。

注意事项 在这个操作例子中的脉冲宽度测量上升到定时器计数器的一个周期。

图 6-26. 通过重启定时器设置脉冲宽度测量的控制寄存器（指定上升沿有效）(1/2)

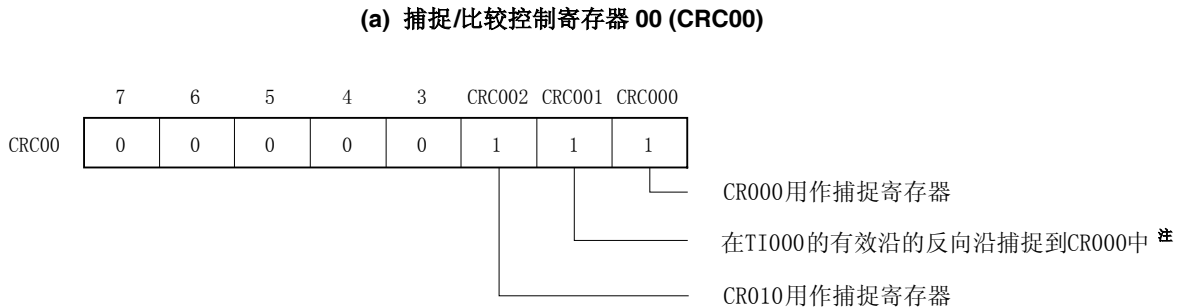
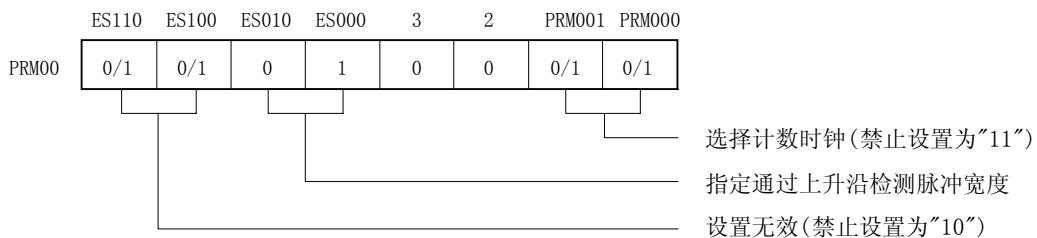
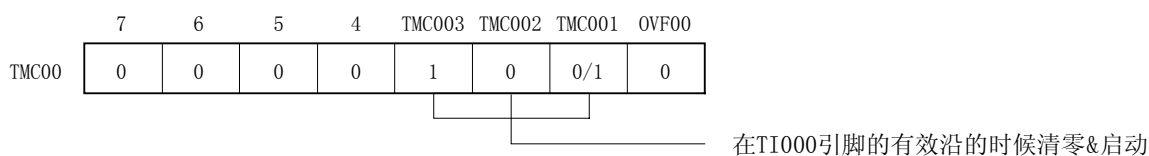


图 6-26. 通过重启定时器设置脉冲宽度测量的控制寄存器（指定上升沿有效） (2/2)

(b) 预分频模式寄存器 00 (PRM00)

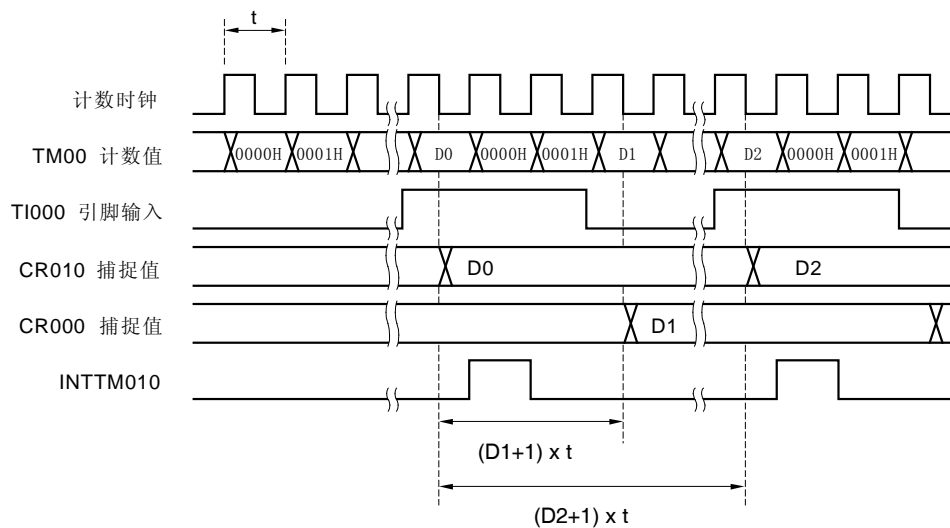


(c) 16 位定时器模式控制寄存器 00 (TMC00)



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

图 6-27. 通过重新设置控制寄存器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 (见 图 6-28)。
- <3> 设置 TOC00 寄存器 (见 图 6-28)。
- <4> 设置 CR000 寄存器 (不能设置成 0000H)。
- <5> 设置 TMC00 寄存器启动操作 (见 图 6-28)。

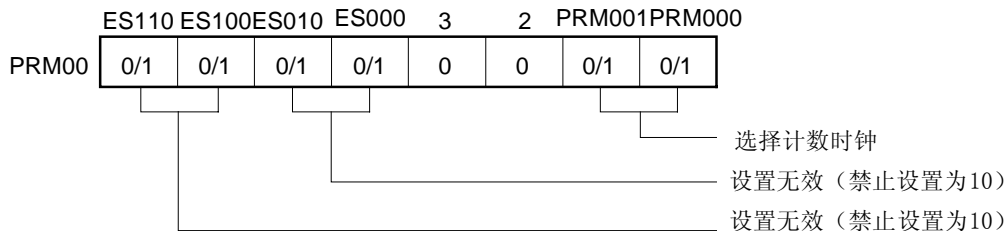
注意事项 如在 TM00 操作中要改变 CR000 的设置可能引起错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 00 (17) 在定时操作中改变比较寄存器的相关注意事项。

- 备注**
- 1. 如需设置 TO00 管脚，请参见 6.3 (5) 端口模式寄存器 2 (PM2) 和端口模式控制寄存器 2 (PMC2)。
 - 2. 如何使能 INTTM000 中断，请参见 第十章 中断功能。

根据 16 位定时器捕捉/比较寄存器 000(CR000)预置计数值所确定的时间间隔，可输出任选频率方波。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 CR00n+1 的预置值为间隔翻转。这样可输出任选频率的方波。

图 6-28. 方波输出模式下控制寄存器的设置(1/2)

(a) 预分频模式寄存器 00 (PRM00)



(b) 捕捉比较控制寄存器 00 (CRC00)

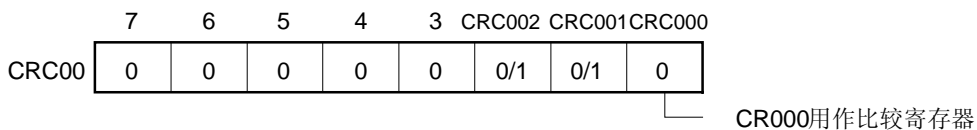
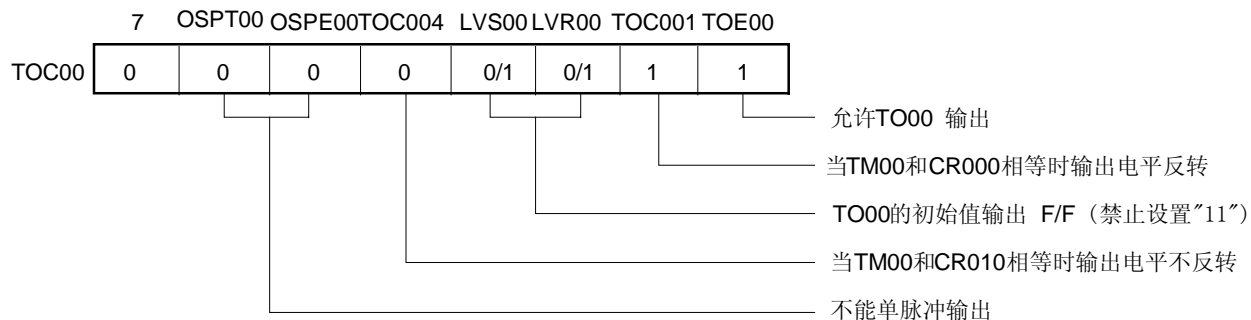
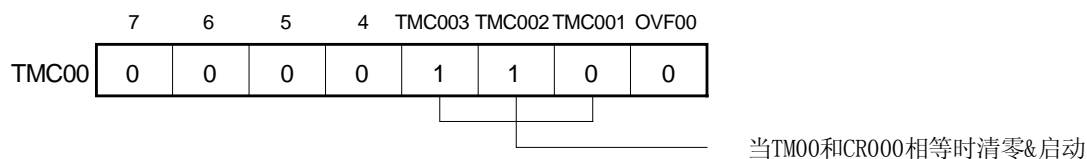


图 6-28. 方波输出模式下控制寄存器的设置 (2/2)

(c) 16 位定时器输出控制寄存器 00 (TOC00)

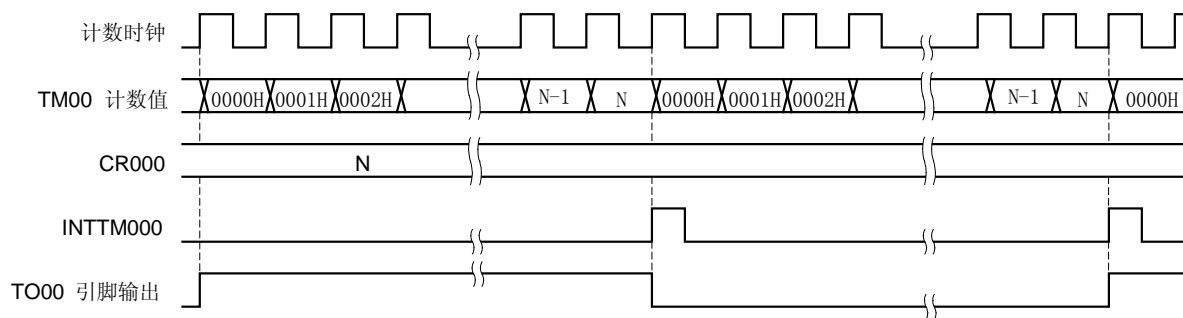


(d) 16 位定时器模式控制寄存器 00 (TMC00)



备注 0/1:设置 0 或 1 可允许另外一个功能与方波输出同时使用。需要了解详细信息，参见各自控制寄存器的描述。

图 6-29. 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) 和捕捉/比较控制寄存器 (CRC00) 可以使用可编程脉冲发生器 PPG 输出, 详见图 6-30。

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 6-30)。
- <2> 设置 CR000 寄存器的内容为任意值, 作为时钟周期。
- <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 6-30)。
- <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 6-30)。

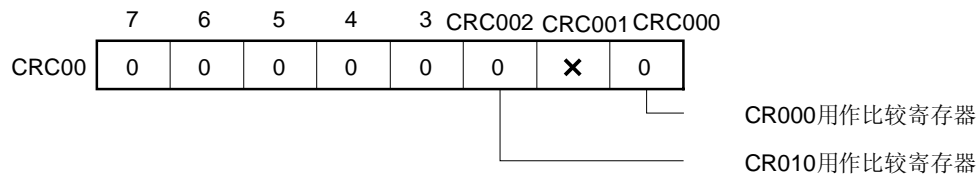
注意事项 如在 TM00 操作中要改变 CR000 的设置可能引起错误。要改变设置, 请参考 6.5 16 位定时器/事件计数器 00 (17) 在定时操作中改变比较寄存器的相关注意事项。

- 备注**
1. 如需设置 TO00 引脚, 请参见 6.3 (5) 端口模式寄存器 2 (PM2) 和端口模式控制寄存器 2 (PMC2)。
 2. 如何使能 INTTM000 中断, 请参见 第十章 中断功能。
 3. $n = 0$ 或 1

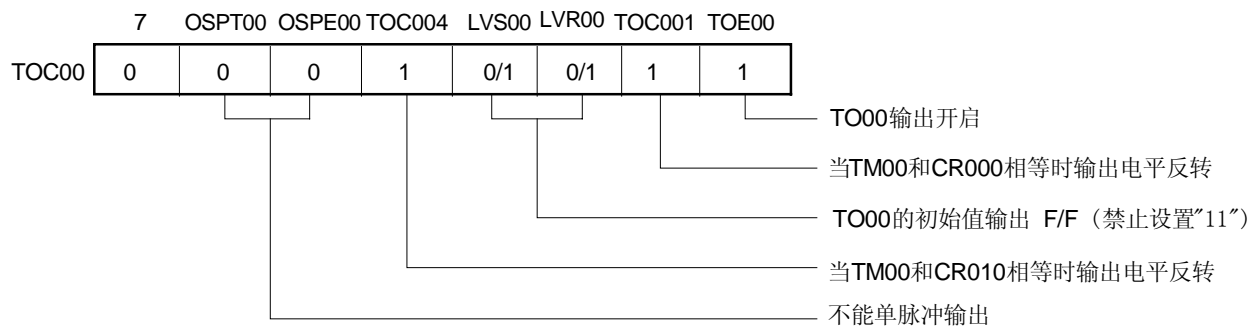
在 PPG 输出操作中, 从 TO00 引脚输出矩形波, 脉冲宽度和周期分别与 16 位定时器捕捉/比较寄存器 010 (CR010) 和 16 位定时器捕捉/比较寄存器 000 (CR000) 中的预置计数值一致。

图 6-30. PPG 输出操作的控制寄存器的设置

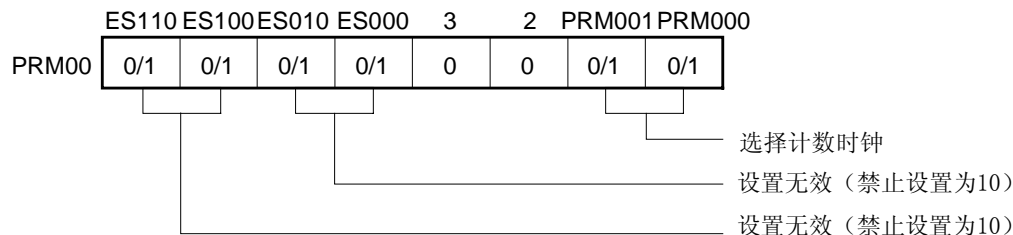
(a) 捕捉/比较控制寄存器 00 (CRC00)



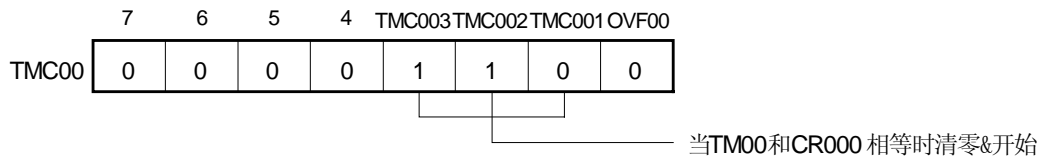
(b) 16 位定时器输出控制寄存器 00 (TOC00)



(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. 由 PPG 输出(CR00n 设置值+ 1)产生的脉冲周期的占空比为 (CR01n 设置值 + 1)/(CR00n 设置值 + 1)。

备注 ×: 不必考虑

图 6-31. PPG 输出的配置框图

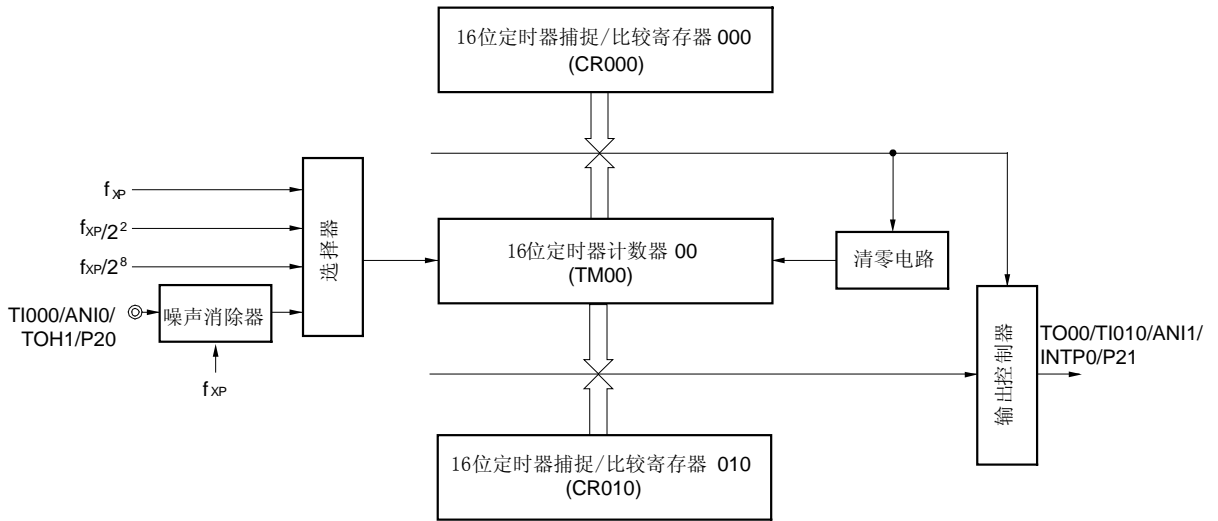
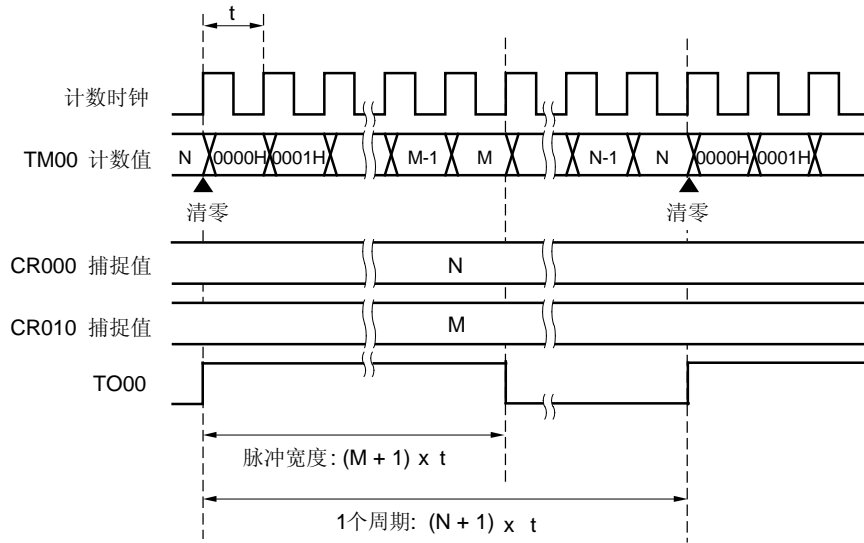


图 6-32. PPG 输出操作的时序



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 6-33 和 6-35)。
- <3> 设置 TOC00 寄存器(设置值参考图 6-33 和 6-35)。
- <4> 设置 CR000 和 CR010 寄存器为任意值(值不可置 0000H)。
- <5> 设置 TMC00 寄存器启动操作(设置值参考图 6-33 和 6-35)。

- 备注**
- 1. 如需设置 TO00 引脚, 参见 6.3 (5) 端口模式寄存器 2 (PM2)。
 - 2. 如何使能 INTTM000 中断(根据需要可选 INTTM010), 请参见第十章 中断功能。

(1) 通过软件触发进行单脉冲输出

通过设置 16 位定时器模式控制寄存器 00 (TMC00), 捕捉/比较控制寄存器 00 (CRC00), 16 位定时器输出控制寄存器 00 (TOC00), 通过软件将 TOC00 寄存器的第 6 位设置成 1, 可以输出单脉冲信号, 如下图 6-33 所示。

将 OSPT00 设置为 1, 可对 16 位定时器/事件计数器 00 清零, 并启动。并且计数值与预先设置到 16 位定时器捕捉/比较寄存器 010(CR010)的值(N)相等后, 输出有效。而在计数值与预先设置到 16 位定时器捕捉/比较寄存器 000(CR000)的值(M)相等后, 输出无效。

单脉冲输出后, 寄存器 TM00 仍然在工作。必须将 TMC00 寄存器的 TMC003 和 TMC002 清零才能中止 TM00 寄存器操作。

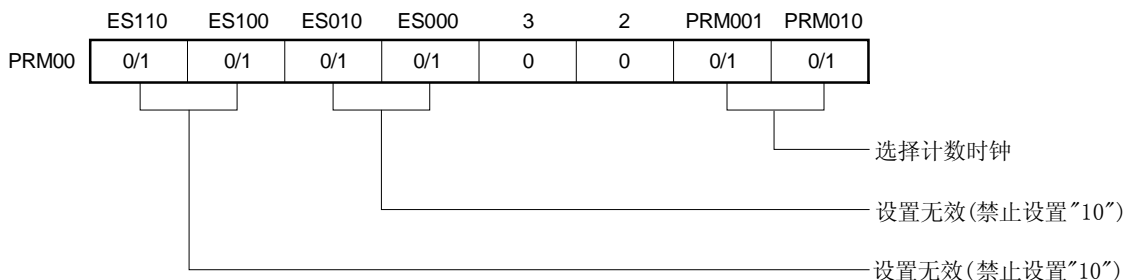
注: 这里所描述的是 $N < M$ 的情况。若 $N > M$, 计数值与 CR000 相等后输出有效, 而与 CR010 相等后输出无效。不能将 N 等于 M。

注意事项

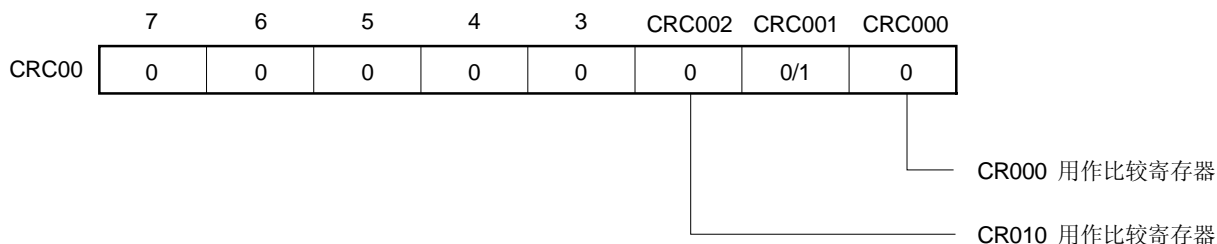
- 1. 当单脉冲输出时, 不要设置 OSPT00 位。在当前单脉冲输出完成后才可再次输出单脉冲。
- 2. 使用软件触发进行 16 位定时器/事件计数器 00 的单脉冲输出时, 不要改变 TI000 及其复用端口引脚的电平。由于在这种情况下外部触发也有效, 所以可使用 TI000 或其复用端口引脚对定时器清零&启动。这样会输出非预期的时序。

图 6-33. 软件触发的单脉冲输出的控制寄存器的设置

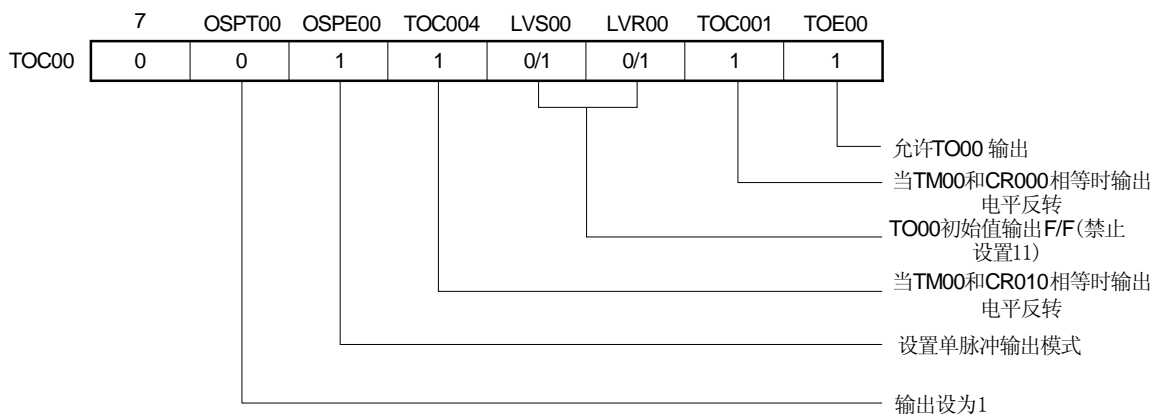
(a) 预分频模式寄存器 00 (PRM00)



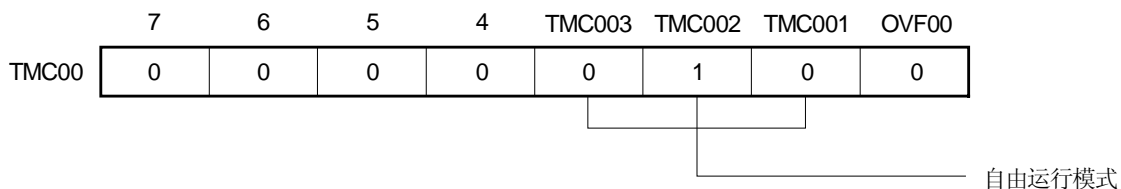
(b) 捕捉/比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)

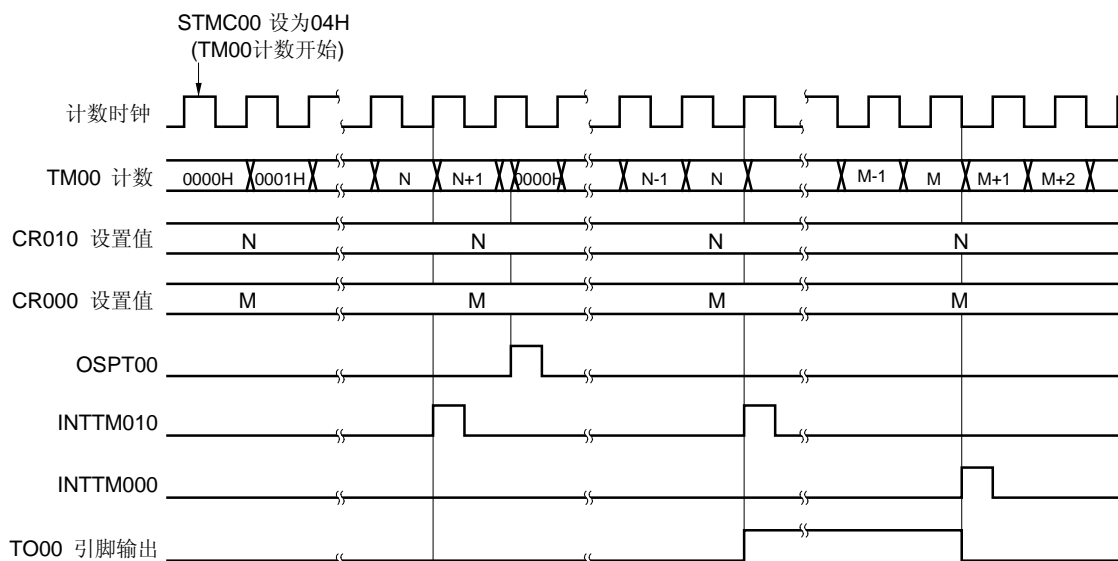


(d) 16 位定时器模式控制寄存器 00 (TMC00)



注意事项 CR000 和 CR010 寄存器不能置 0000H。

图 6-34. 软件触发的单脉冲输出操作的时序



注意事项 一旦为 TMC003 和 TMC002 位赋值 (00 除外, 操作停止模式), 16 位定时器计数器 00 开始计数。

备注 $N < M$

(2) 外部触发的单脉冲输出

通过设置 16 位定时器模式控制寄存器 00 (TMC00), 捕捉/比较控制寄存器 00 (CRC00), 16 位定时器输出控制寄存器 00 (TOC00) 和使用 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, 如下图 6-35 所示。

预分频模式寄存器 00 (PRM00) 寄存器的第 4, 5 位 (ES000 和 ES010) 可以设定 TI000 引脚的有效沿。有效沿可以为上升沿, 下降沿或皆有上升沿和下降沿三种。

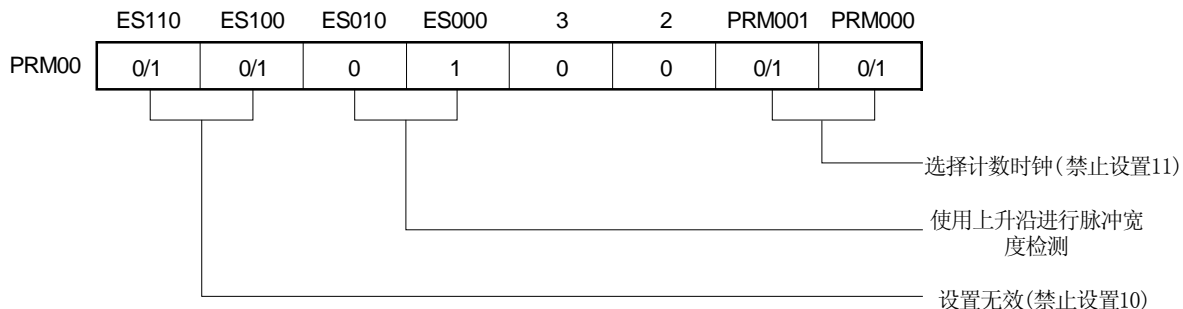
当检测到 TI000 的有效沿时, 将 16 位定时器/事件计数器清零并开始计数。并且计数值与预先设置到 16 位定时器捕捉/比较寄存器 010 (CR010) 的值 (N) 相等后, 输出有效。而在计数值与预先设置到 16 位定时器捕捉/比较寄存器 000 (CR000) 的值 (M) 相等后, 输出无效。

注: 这里所描述的是 $N < M$ 的情况。若 $N > M$, 计数值与 CR000 相等后输出有效, 而与 CR010 相等后输出无效。不能将 N 等于 M。

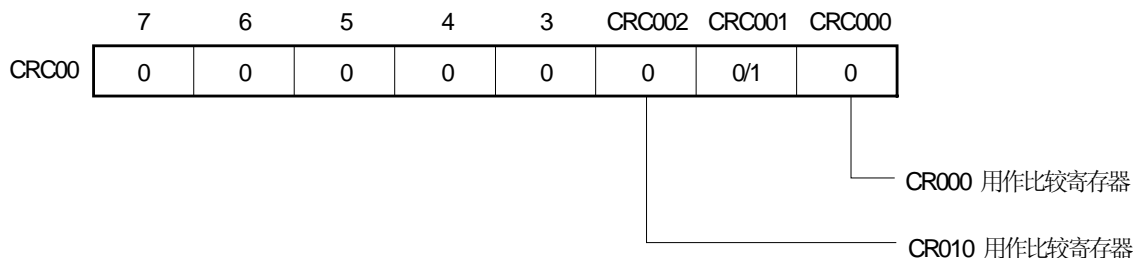
注意事项 当单脉冲正在输出时不要再次再输入外部触发。
在当前单脉冲输出完成后才可再次输出单脉冲。

图 6-35. 外部触发的单脉冲输出的控制寄存器设置(指定上升沿有效)

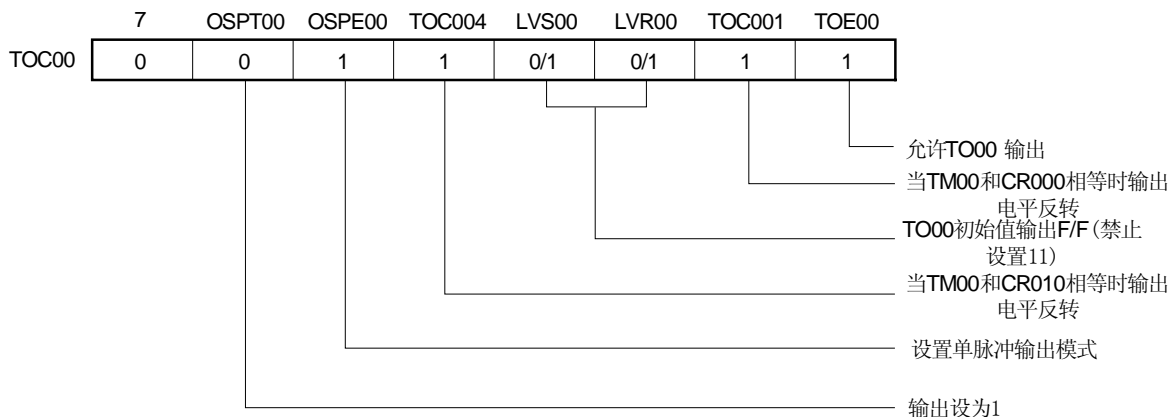
(a) 预分频模式寄存器 00 (PRM00)



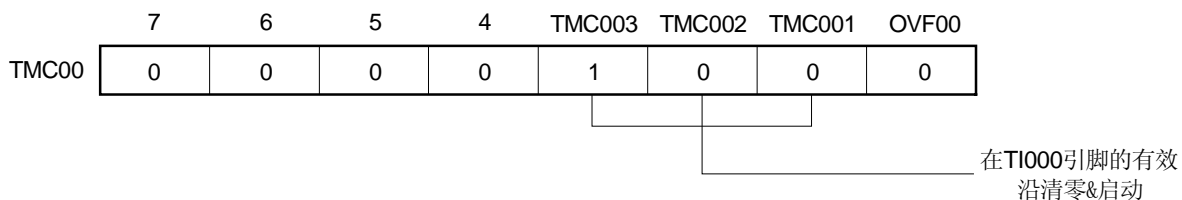
(b) 捕捉/比较控制寄存器 00 (CRC00)



(c) 16位定时器输出控制寄存器 00 (TOC00)

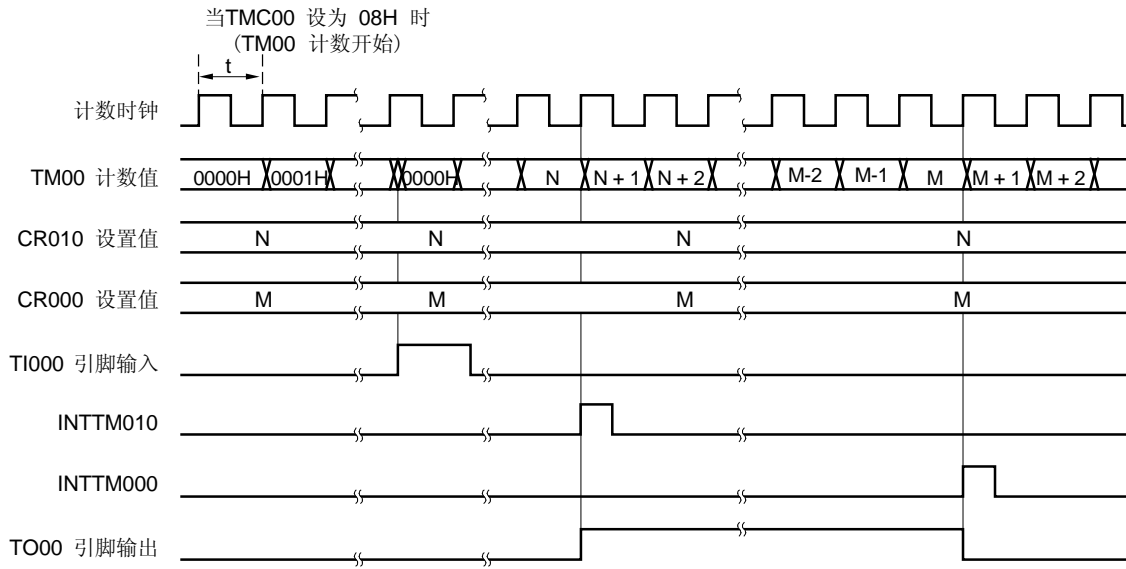


(d) 16位定时器模式控制寄存器 00 (TMC00)



注意事项 CR000 和 CR010 寄存器不能置 0000H。

图 6-36. 外部触发单脉冲输出的时序（指定上升沿有效）



注意事项 一旦为 TMC003 和 TMC002 位赋值（00 除外，操作停止模式），16 位定时器计数器 00 开始计数。

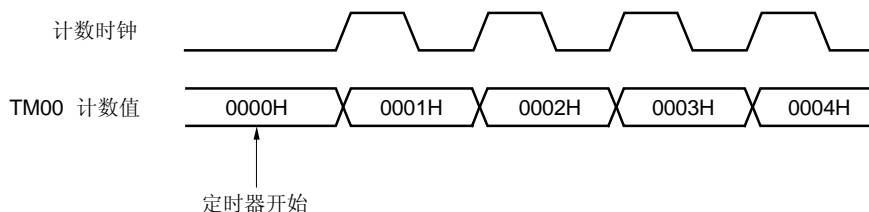
备注 $N < M$

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

定时器启动后在信号匹配产生之前可能发生将近一个时钟的误差，这是由于 16 位定时器计数器 00 (TM00) 的启动与计数时钟不同步。

图 6-37. 16 位定时器计数器 00 (TM00)的启动时序



(2) 16 位定时器计数器 00 (TM00) 操作

- <1> 当 TMC002 和 TMC003 位的值不为 0, 0 (操作停止模式) 时, 16 位定时器 TM00 开始工作。相反, 当 TMC002 和 TMC003 位的值设置为 0, 0 时, 16 位定时器 TM00 停止工作。
- <2> 即使 TM00 的值可读, 这个值也不能被 16 位定时器捕捉/比较寄存器 010 (CR010) 捕捉。
- <3> 当 TM00 中的值被读取时, 计数时钟停止工作。
- <4> 不管 CPU 处于什么工作模式, 当定时器是停止的, 输入 TI000/TI010 引脚的信号不能被响应。

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

- <1> 在 TM00 和 CR000 相等时进入清零&启动模式中, 通过设置 16 位捕捉/比较寄存器 000 (CR000) 为非 0000H 的值。也就是说当此寄存器作为外部事件计数器时, 不能执行 1 个脉冲计数操作。
- <2> TM00 和 CR000 相等时进入清零和启动模式, CR000 寄存器不能作为一个捕捉寄存器使用。
- <3> 在自由运行模式和清零&启动工作模式下 TI000 引脚输入有效沿作为输入时, 如果 CR0n0 寄存器设置为 0000H, 当 CR0n0 从 0000H 变化到 0001H 时, INTTM0n0 中断产生。
- <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 的新值比老值小时, 定时器必须在改变了 CR0n0 的值以后重新启动。

(4) 捕捉寄存器的数据保持

当 16 位定时/计数器 TM00 停止工作后，16 位定时器捕捉/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

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

(6) 捕捉/比较控制寄存器 00 (CRC00) 的设置

设置 CRC00 时，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> 设置定时器前，必须停止工作，OSPT00 除外。

<2> 当读取 LVS00 和 LVR00 时，读出值为 0。

<3> 数据设置之后，OSPT00 自动清零，读该位时，读出值为 0。

<4> 除单脉冲输出模式外，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，总是向 PRM00 写入数据。

(9) 有效沿设置

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

(10) 单脉冲输出

一般只有在自由运行模式或清零&启动模式下，TI000 引脚有效沿输入才能有单脉冲输出。因为在清零&启动模式下，TM00 和 CR000 相等时不发生溢出，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，不要再置 OSPT00 位为 1。等正在进行的单脉冲输出完毕后再次输出单脉冲。

<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 引脚或其复用端口引脚的电平。因为即使在这种情况下外部触发也是有效的，即使是 TI000 引脚或其复用端口引脚的电平也能使定时器清零并重新开始。在非预期的时间内产生脉冲输出。

<3> 16位定时器捕捉/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。
等正在进行的单脉冲输出完毕后再输出单脉冲。

<2> 16位定时器捕捉/比较寄存器 000 和 010 (CR000 和 CR010) 不能设置为 0000H。

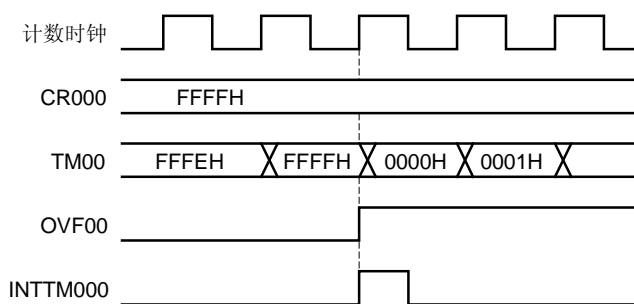
(13) OVF00 标志位操作

<1> OVF00 标志位在下列情况下设置为 1。

在选择清零&启动模式 (由 TM00 和 CR000 相等或者在 T1000 的有效沿时进入), 或选择自由运行模式时。

↓
CR000 置 FFFFH。
↓
当 TM00 从 FFFFH 计数到 0000H 时。

图 6-38. OVF00 标志位的操作时序

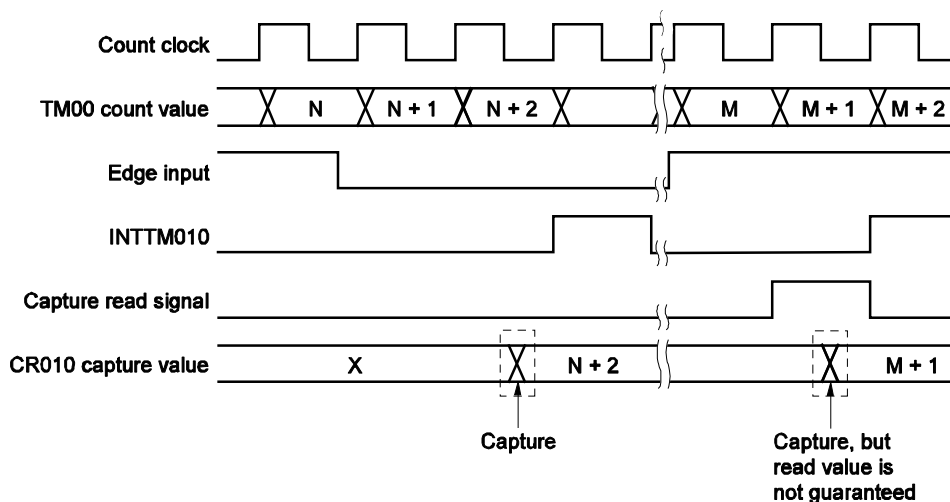


<2> 即便在 TM00 溢出后而下一个计数时钟计数前 (TM00 变为 0001H 前) 做清除 OVF00 标志操作, OVF00 标志还会被重新置 1, 清除操作无效。

(14) 冲突操作

CR000/CR010 被用作为捕捉寄存器时, 如果寄存器读周期和捕捉触发输入相冲突, 那么捕捉触发输入有优先权, 被读数据未定义。同样, 当计数终止输入和捕捉触发输入冲突时, 被捕捉数据不确定。

Figure 6-39. Capture Register Data Retention Timing

**(15) 捕捉操作**

- <1> 如果 TI000 引脚的有效沿被用于计数时钟, 就不能设置清零启动模式和 TI000 引脚有效沿的捕捉。
- <2> 当 CRC001 值为 1, 如果 TI000 引脚的上升沿和下降沿同时作为输入有效沿, 则捕捉操作无效。
- <3> 当 CRC001 值为 1, 如果 TI010 引脚有效沿被检测, TM00 的计数值不能被捕捉到 CR000 寄存器中, 但是 TI010 引脚的输入可以用作外部中断源, 可以产生 INTTM000 中断。
- <4> 为保证捕捉操作的可靠性, 捕捉触发需要用预分频模式寄存器 00 (PRM00) 来选择一个大于两个计数时钟周期的脉冲。
- <5> 在计数时钟下降沿处执行捕捉操作。然而, 在下一个计数时钟的上升沿会出现一个中断请求输入 (INTTM0n0)。
- <6> 要使用两个捕捉寄存器时, 设置 TI000 和 TI010 引脚。

备注 $n = 0, 1$

(16) 比较操作

当 CR0n0 在比较模式下, 即使有捕捉触发输入, 捕捉操作也可能不能执行。

备注 $n = 0, 1$

(17) 在定时器工作时改变比较寄存器的值

<1> 16 位定时器捕捉/比较寄存器 0n0 (CR0n0) 作为比较寄存器使用, 定时器计数时当 16 位定时器计数器 00 (TM00) 和 16 位定时器捕捉/比较寄存器 0n0 (CR0n0) 的值相等阶段修改 CR0n0 的值时, 改变 CR0n0 的时序和相等时序可能会冲突, 这种情况下, 操作不能得到保证。在定时器计数时改变 CR0n0 的值时, INTTM000 中断服务程序有下面操作。

<更改周期 (CR000)>

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

<更改占空比 (CR010)>

1. TM00 和 CR010 (TOC004 = 0) 相等时禁止定时器输出反转操作。
2. 禁止 INTTM000 中断 (TMMK000 = 1)。
3. 重置 CR010。
4. 等待一个 TM00 计数时钟周期。
5. TM00 和 CR010 (TOC004=1) 相等时允许定时器输出反转操作。
6. INTTM000 (TMIF000 = 0) 的中断请求标志位清零。
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 中断后, 定时器时钟和 CPU 时钟的影响。

备注 n = 0, 1

<2> 如果在定时器继续计数过程中, 如果不执行上面的<1>, 则 CR010 中的值将被重新写入两次或更多, 每次写入将引起 TO00 引脚电平的反转。

(18) 脉冲沿检测

<1> 按照如下的方法, 注意 TI0n0 引脚有效沿被检测的警告。

(a) 系统复位后, 如果 TI0n0 引脚输入高电平, 然后 16 位定时器计数器 00 (TM00) 的操作允许, 注意如下。

→ 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 引脚的输入是高电平的时候, 停止 TM00 操作, TI0n0 引脚的输入变低后有设置 TM00 操作允许, 注意如下。

→ 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。

(c) 如果当 TI0n0 引脚的输入是低的时候, 停止 TM00 操作, TI0n0 引脚的输入变高后有设置 TM00 操作允许, 注意如下。

→ 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。

备注 n = 0, 1

<2> 当 TI000 有效沿用作计数时钟和用作捕捉触发时, 所用的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 fx。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。直到采样有效沿并且有效电平被检测到两次之后才能完成捕捉操作, 因此, 可以消除短脉冲宽度噪音。

(19) 外部事件计数器

要想知道外部事件计数器计的值, 应该去读 TM00。

(20) PPG 输出

<1> CR000 和 CR010 中设置的值应在如下范围内:

$0000H < CR010 < CR000 \leq FFFFH$ (禁止设置 CR000 为 0000H)

<2> 通过 PPG 输出产生的脉冲的周期 (CR000 的值 + 1) 的占空比是: $(CR010 \text{ 的值} + 1) / (CR000 \text{ 的值} + 1)$ 。

(21) 停止模式或系统时钟停止模式的设置

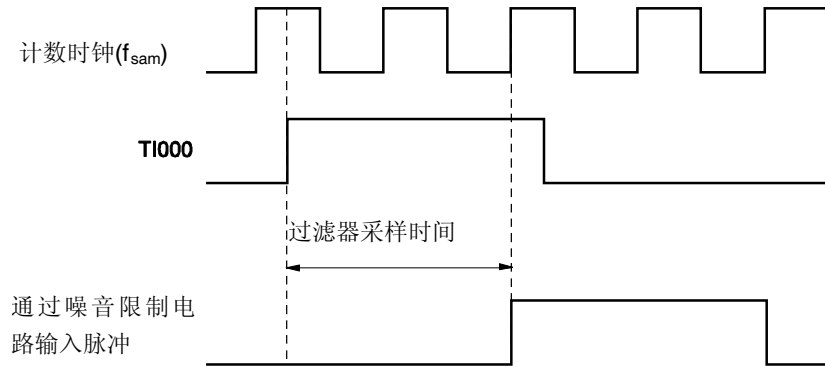
除 TI000 引脚有效沿作为计数器时钟信号输入以外, 在设置停止模式或系统时钟停止模式前停止定时器工作; 否则在系统时钟重新启动时, 定时器将发生误差。

(22) P21/TI010/TO00 引脚

当 P21 用做 TI010 的有效沿输入引脚时, 就不能被用于定时器输出了 (TO00)。同样, 当 P21 被用于 TO00 时, 也不能被用于 TI010 的有效沿输入。

(23) 外部时钟限制

- <1> 当使用 TI000 引脚输入脉冲作为计数时钟(外部触发器), 确保输入满足 AC 特性的的脉冲宽度。关于 AC 特性, 参照第十八章 电器特性。
- <2> 当外部波形输入到 16 位定时器/事件计数器 00, 通过噪音限制电路和时序发生的错误取样成为器件的有效沿。



备注 计数时钟(f_{sam})可以选择使用预分频模式寄存器 00(PRM00)的位 0 和 1(PRM000, PRM001)。

第七章 8 位定时器 H1

7.1 8 位定时器 H1 的功能

8 位定时器 H1 的功能如下:

- 间隔定时器
- PWM 输出
- 方波输出

7.2 8 位定时器 H1 的配置

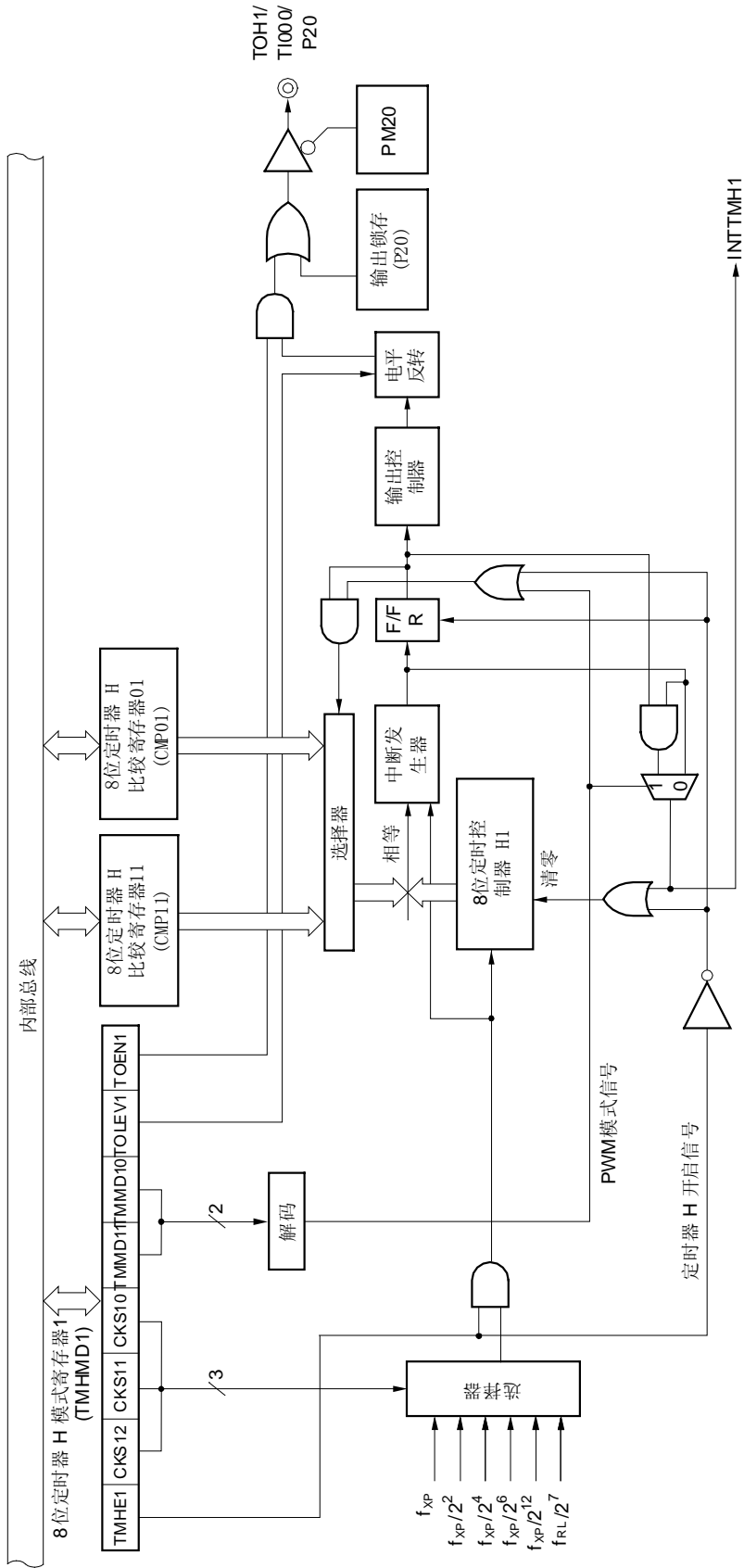
8 位定时器 H1 由以下硬件组成。

表 7-1. 8 位定时器 H1 的配置

项目	配置
定时器寄存器	8 位定时器计数器 H1
寄存器	8 位定时器 H 比较寄存器 01 (CMP01) 8 位定时器 H 比较寄存器 11 (CMP11)
定时器输出	TOH1
控制寄存器	8 位定时器 H 模式寄存器 1 (TMHMD1) 端口模式寄存器 2 (PM2) 端口寄存器 2 (P2) 端口模式控制寄存器 2 (PMC2)

图 7-1 是结构框图

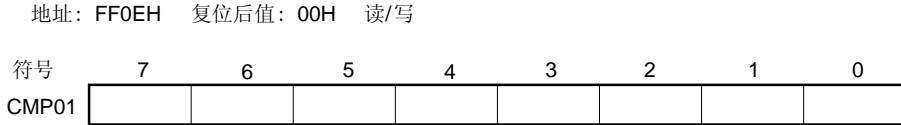
图 7-1. 8 位定时器 H1 的配置框图



(1) 8 位定时器 H 比较寄存器 01 (CMP01)

8 位定时器 H 比较寄存器 01 (CMP01) 可通过 8 位存储操作指令进行读写。
 复位后寄存器值被清零。

图 7-2 8 位定时器 H 比较寄存器 01 (CMP01) 的格式

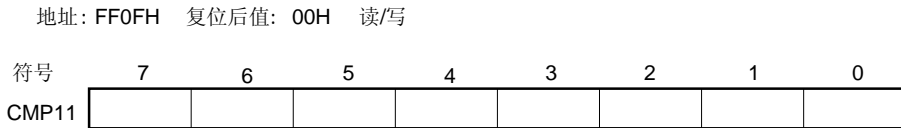


注意事项 在定时器计数期间不能修改 CMP01。

(2) 8 位定时器 H 比较寄存器 11 (CMP11)

8 位定时器 H 比较寄存器 11 (CMP11) 可通过 8 位存储操作指令进行读写。
 复位后寄存器值被清零。

图 7-3 8 位定时器 H 比较寄存器 11 (CMP11) 的格式



CMP11 在定时器计数期间可被重写。

如果在定时器操作期间重写 CMP11 值，当计数值和修改以前的比较值一致时，修改后的比较值才生效。如果计数值和修改以前的比较值一致时 CPU 正向 CMP11 写入新值，则在下一个计数值和修改以前的比较值一致时修改后的比较值生效。

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 (即使前后设置的值相同也必须对 CMP11 重新设置)。

7.3 控制 8 位定时器 H1 的寄存器

以下的寄存器用来控制 8 位定时器 H1：

- 8 位定时器 H 模式寄存器 1 (TMHMD1)
- 端口模式寄存器 2 (PM2)
- 端口寄存器 2 (P2)
- 端口控制寄存器 2 (PMC2)

(1) 8 位定时器 H 模式寄存器 1 (TMHMD1)

该寄存器控制 8 位定时器 H 的模式。

可通过 1 位或 8 位存储操作指令进行设置。

复位后寄存器值被清零 (00H)。

图 7-4 8 位定时器 H 模式寄存器 1 (TMHMD1) 的格式

地址: FF70H 复位后: 00H 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(计数器清零)
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.))
其它			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
其它		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(即使前后设置的值相同也必须对 **CMP11** 重新设置)。

- 备注**
1. f_{XP} : 外围硬件设备时钟振荡频率
 2. f_{RL} : 低速内部时钟振荡频率
 3. 上表中括号内的值对应于 $f_{XP}=10\text{MHz}$, $f_{RL}=240\text{kHz}$ (典型值)。

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

当使用 P20/TOH1/TI000/ANI0 引脚作为定时器输出时，要将 PM20，P20 的输出锁存和 PMC20 清零。

PM2 和 PMC2 可通过 1 位或 8 位存储操作指令进行设置。

复位后 PM2 值为 FFH，PMC2 值为 00H。

图 7-5. 端口模式寄存器 2 (PM2) 的格式

地址: FF22H 复位后: FFH R/W

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

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

图 7-6. 端口模式控制寄存器 2 (PMC2) 的格式

地址: FF84H 复位后: 00H R/W

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

PMC2n	工作模式选择 (n = 0 ~ 3)
0	端口/复用功能模式 (A/D 转换器除外)
1	A/D 转换器模式

7.4 8 位定时器 H1 的操作

7.4.1 用作时间间隔定时器/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。

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

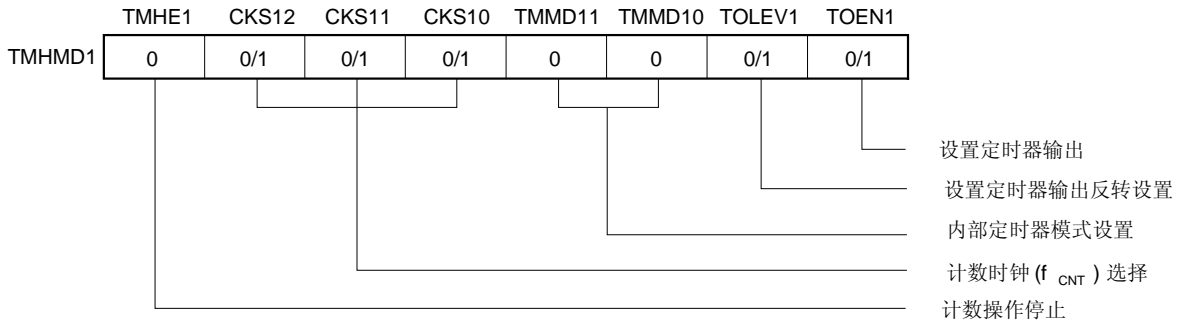
(1) 用法

以相同时间间隔重复产生 INTTMH1 信号。

<1> 各个寄存器的设置

图 7-7. 间隔定时器/方波输出操作时的寄存器设置

(i) 定时器 H 模式寄存器 1 (TMHMD1) 的设置



(ii) CMP01 寄存器设置

- 比较值 (N)

<2> 当 TMHE1=1 开始计数操作。

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

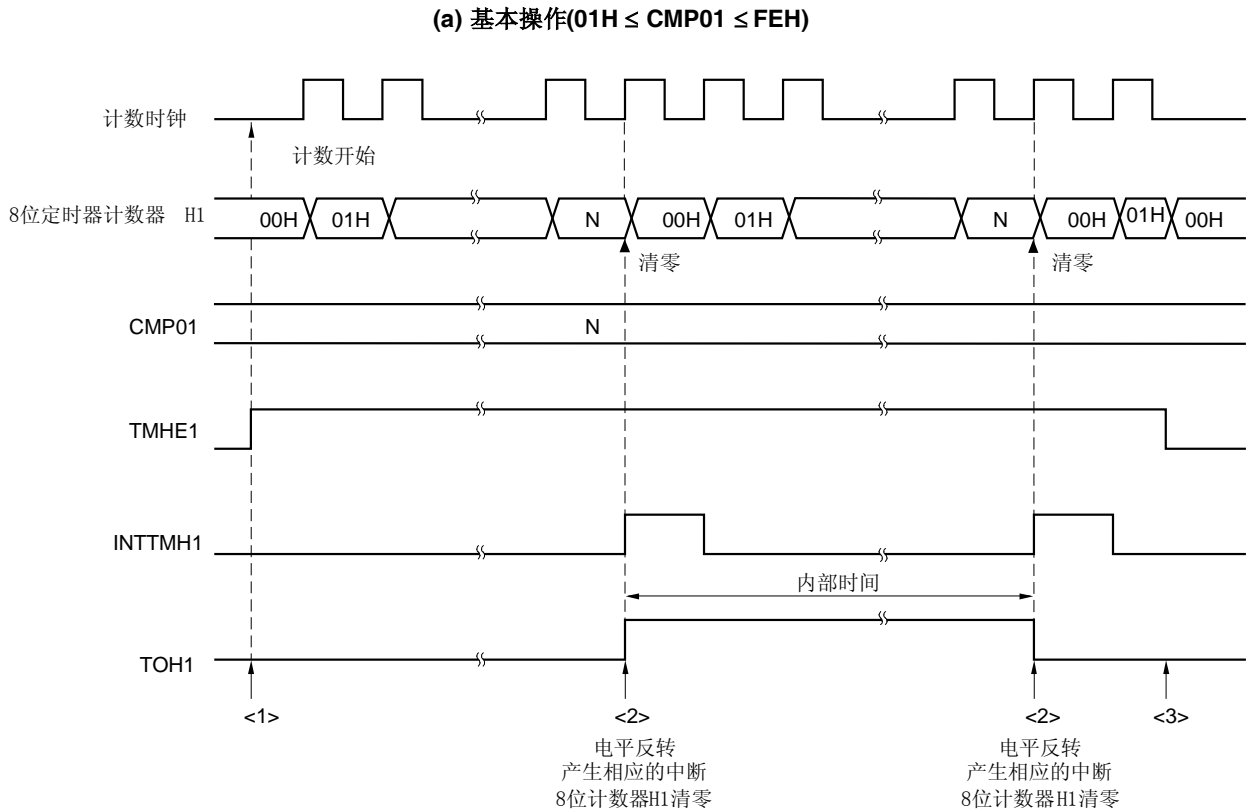
$$\text{间隔时间} = (N + 1) / f_{CNT}$$

<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 7-8. 间隔定时器/方波输出操作的时序图(1/2)



<1> 将 TMHE1 位设置为 1，允许计数操作。操作允许后，在不到 1 个时钟周期的时间内计数时钟开始计数。

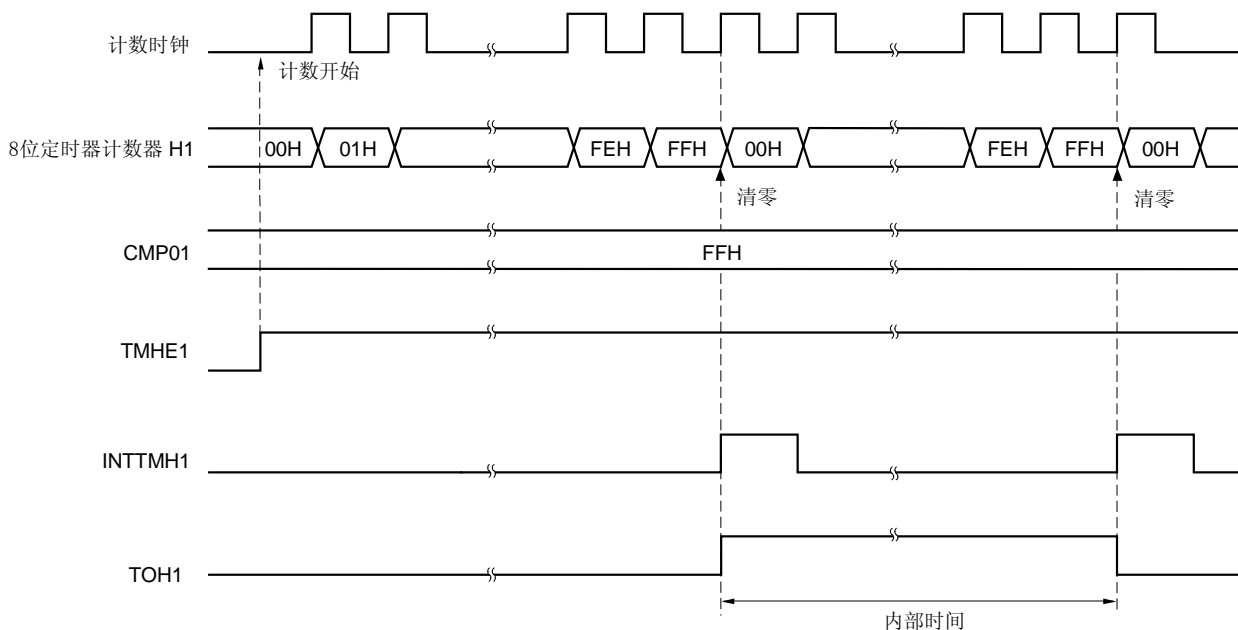
<2> 当 8 位定时器计数器 H1 与 CMP01 寄存器的值相等时，将 8 位定时器计数器 H1 的值清零、反转 TOH1 输出电平并输出 INTTMH1 信号。

<3> 在定时器 H1 操作期间将 TMHE1 位清零，可以使 INTTMH1 信号和 TOH1 输出无效。如果一旦没有 INTTMH1 信号和 TOH1 输出，则电平保持不变。

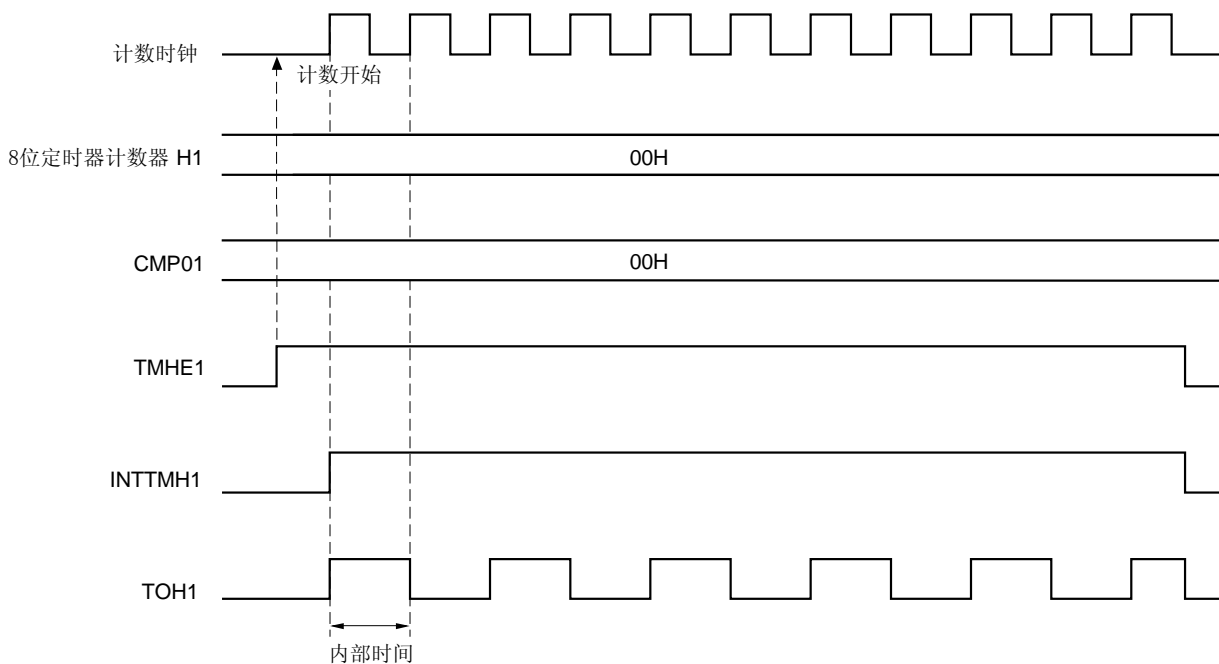
备注 01H ≤ N ≤ FEH

图 7-8. 间隔定时器/方波输出操作的时序图(2/2)

(b) 当 CMP01 = FFH 时的操作



(c) 当 CMP01 = 00H 时的操作



7.4.2 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所示。

在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 开始输出且 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 停止输出。

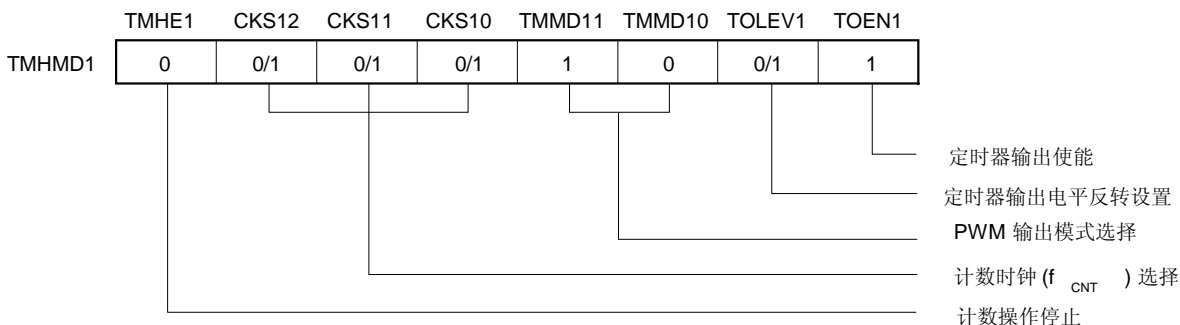
(1) 用法

在 PWM 输出模式中，可输出具有任意占空比和周期的脉冲。

<1> 设置寄存器

图 7-9. PWM 输出模式下的寄存器设置

(i) 设置定时器 H 模式寄存器 1 (TMHMD1)



(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。

<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，将 8 位定时器计数器 H1 清零、产生中断请求信号(INTTMH1)并且 TOH1 开始输出。同时切换比较寄存器（与 8 位定时器比较的寄存器）：由 CMP01 改为 CMP11。

- <4> 当8位定时器计数器H1与CMP11寄存器相等时，停止TOH1输出，并切换比较寄存器（与8位定时器比较的寄存器）：由CMP11改为CMP01。此时不对8位定时器计数器清零，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。

PWM 脉冲输出周期 = $(N+1)/f_{CNT}$
 占空比=有效宽度: PWM 的整个宽度 = $(M + 1) : (N + 1)$

- 注意事项**
1. 在PWM输出模式下，CMP11的设定值可以在计数操作时修改。然而，传送重写CMP11后需要3个（根据TMHMD1寄存器的CKS12和CKS10位选择的信号）或更多的时钟周期。
 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（即使设置的是相同的值，也必须再次赋值）。

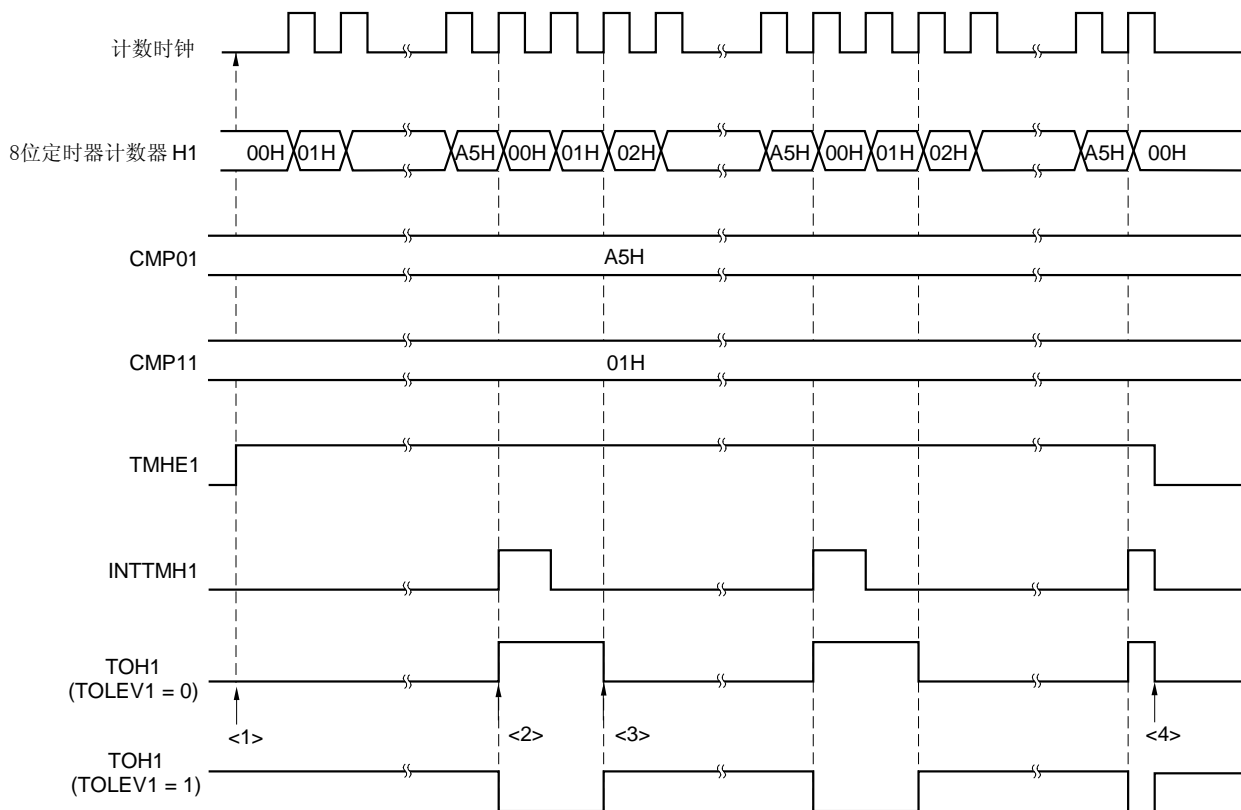
(2)时序图

PWM输出模式中的时序操作过程如下所示。

- 注意事项** CMP11的取值（M）和CMP01的取值(N)应在如下范围内。
 $00H \leq \text{CMP11 (M)} < \text{CMP01 (N)} \leq \text{FFH}$

图 7-10. PWM 输出模式下的时序操作(1/4)

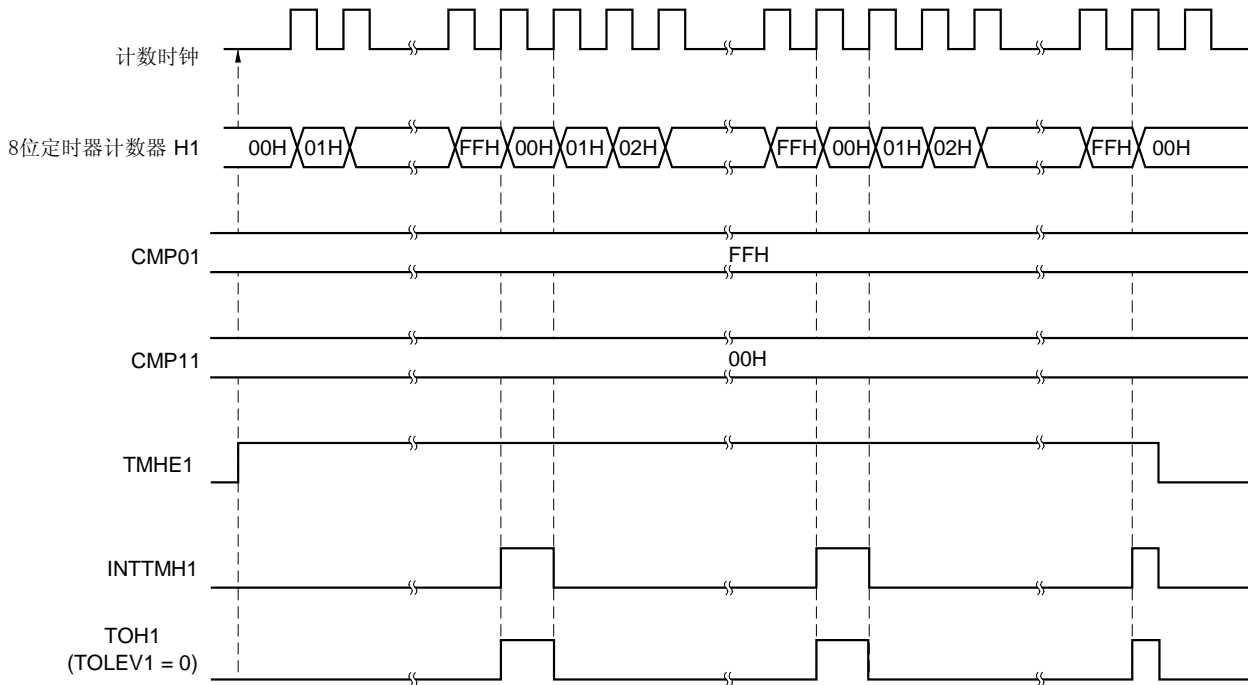
(a) 基本操作(00H < CMP11 < CMP01 < FFH)



- <1> TMHE1 置 1，则允许计数操作。屏蔽一个计数时钟，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，将 TOH1 输出电平反转，对 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。而此时并不对 8 位定时器计数器 H1 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 7-10. 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

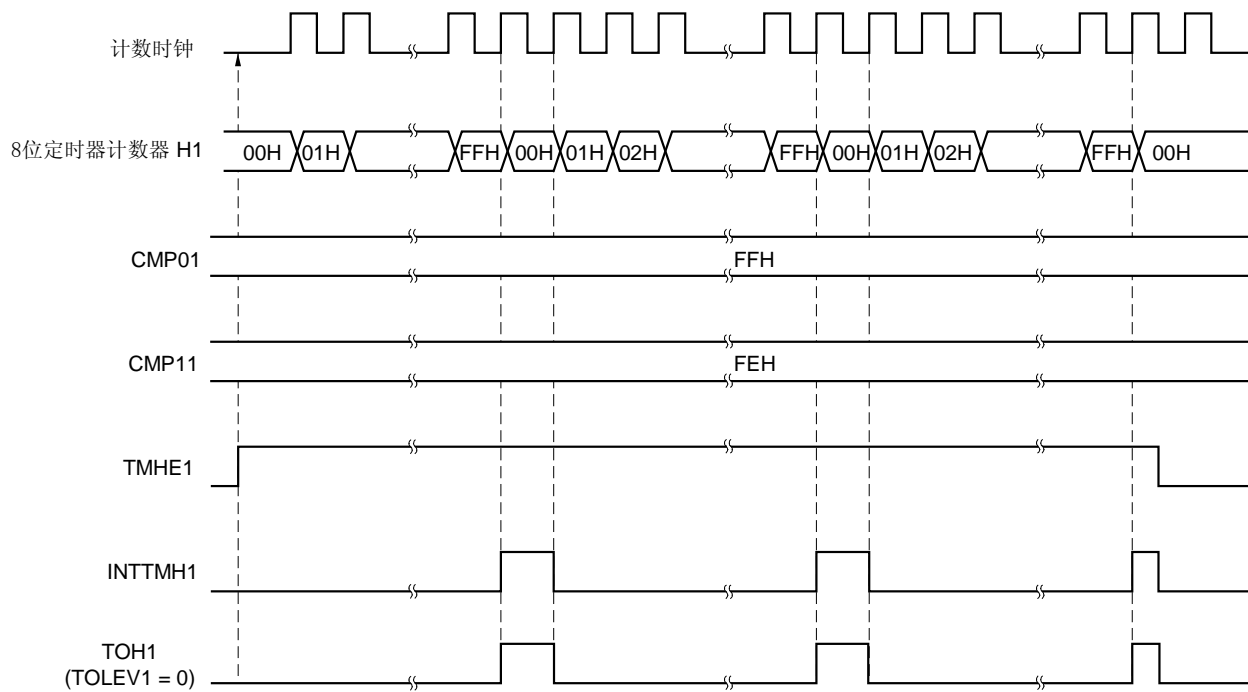


图 7-10. PWM 输出模式下的时序操作(3/4)

(d) 当 CMP01 = 01H, CMP11 = 00H 时的操作

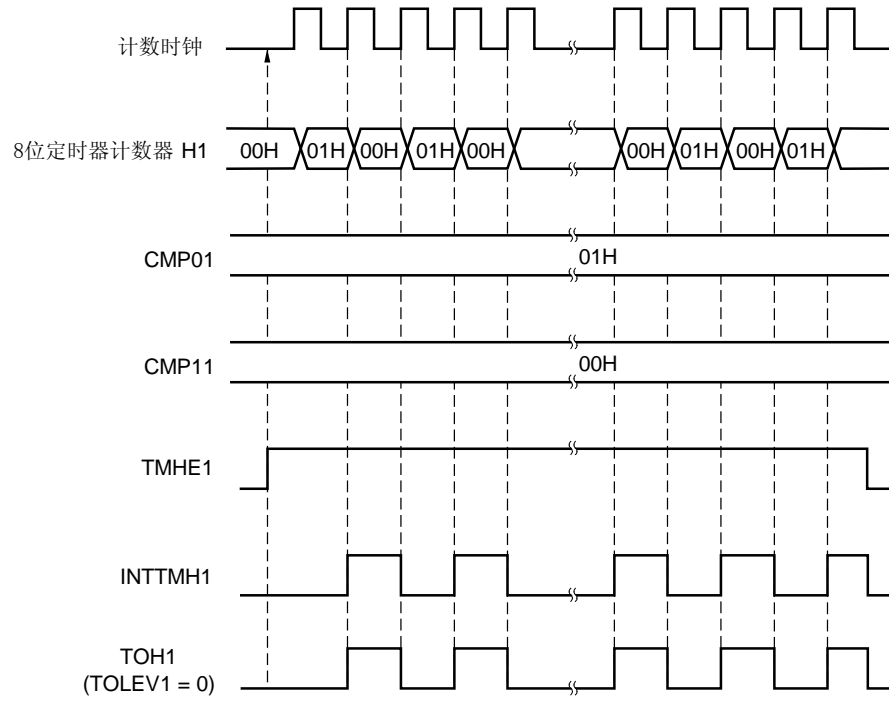
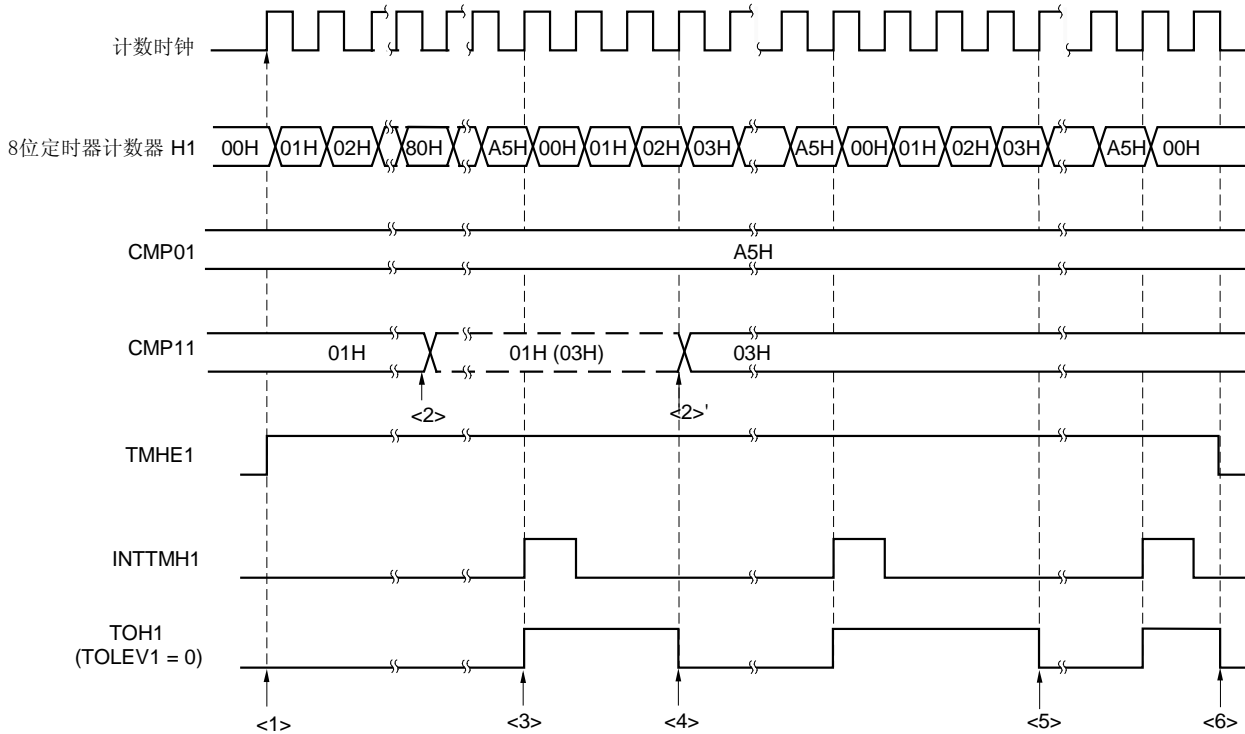


图 7-10. PWM 输出模式下的时序操作(4/4)

(e) 改变 CMP11 的操作(CMP11 = 01H → 03H, CMP01 = A5H)



- <1> TMHE1 置 1，则允许计数操作。屏蔽 1 个计数时钟，启动 8 位定时器计数器 H1。此时 TOH1 输出保持无效。(TOLEV1=0)
- <2> 在定时器计数器操作期间可改变 CMP11 的值。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 的值改变，则改变后的值被锁存，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 寄存器的值相等时，则将值传送至 CMP11，CMP11 的值改变(<2>)。但从 CMP11 的值被改变到将值传送至寄存器需要 3 个或更多的计数时钟。如果在 3 个计数时钟内产生了一个相等信号，则不能将改变后的值传送至寄存器。
- <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 输出无效。此时不对 8 位定时器计数器 H1 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。

第八章 看门狗定时器

8.1 看门狗定时器的功能

看门狗定时器用于检测不期望的程序循环。如果检测到这样的程序循环，将产生一个内部复位信号。

如果由于看门狗定时器而产生复位，则将复位控制标志寄存器(RESF)的第4位(WDTRF)置1。需要了解RESF的详细信息，可参阅第十二章 复位功能。

表 8-1. 看门狗定时器的循环检测时间

循环检测时间	
低速内部时钟操作期间	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} : 低速内部时钟振荡频率。
 2. f_x : 系统时钟振荡频率。
 3. 图中括号里的值对应于 $f_{RL} = 480$ kHz (MAX.), $f_x = 10$ MHz。

根据内置低速内部时钟的选项字节可切换看门狗定时器的操作模式，如表 8-2 所示。

表 8-2. 选项字节的设置和看门狗定时器操作模式

	选项字节设置	
	低速内部时钟不能被停止	低速内部时钟可通过软件停止
看门狗定时器时钟源	恒为 f_{RL} ^{注1}	<ul style="list-style-type: none"> 可由软件选择 (f_x, f_{RL} 或停止) 复位后: 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	看门狗定时器能被停止 ^{注2}

- 注**
- 在供电时不能停止低速内部振荡器(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。
 - 如果时钟源是 f_x , 则在满足以下条件时停止看门狗定时器的供应时钟。
 - 当 f_x 被停止。
 - 在 HALT/STOP 模式中。
 - 在振荡器稳定时间内。
 - 如果时钟源是 f_{RL} , 则在满足以下条件时停止看门狗定时器的供应时钟。
 - 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} : 低速内部时钟振荡频率
 - f_x : 系统时钟振荡频率

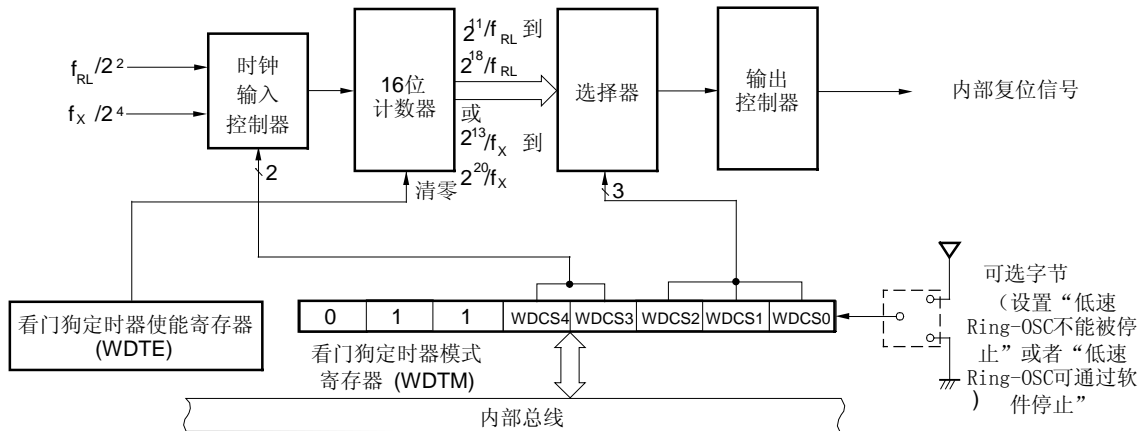
8.2 看门狗定时器的配置

看门狗定时器由以下硬件组成。

表 8-3. 看门狗定时器的配置

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

图 8-1. 看门狗定时器的框图



备注

1. f_{RL} : 低速内部时钟振荡频率。
2. f_X : 系统时钟振荡频率。

8.3 控制看门狗定时器的寄存器

看门狗定时器由以下两种寄存器控制：

- 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，但复位释放后只能被写 1 次。

复位输入将该寄存器置 67H。

图 8-2. 看门狗定时器模式寄存器(WDTM) 的格式

地址: FF48H 复位后值: 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	低速内部时钟 (f _{RL})
0	1	系统时钟 (f _X)
1	×	停止看门狗定时器的操作

WDCS2 ^{注2}	WDCS1 ^{注2}	WDCS0 ^{注2}	溢出时间设置	
			低速内部时钟操作期间	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. 如果选项字节选项为“低速内部振荡不能被停止”，则不能设置该项。无论写入何值，都将选择低速内部振荡时钟。
 2. 以最大周期复位释放(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。不能设置其他值。

注意事项 2. 复位后，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。然而，第一次写入时，WDCS4 和 WDCS3 分别设置为“1”和“x”并且看门狗定时器停止，即使执行下面内容不会产生内部复位信号。

- 第二次写入到 WDTM
 - 1 位存储器操作指令写入到 WDTE1
 - 写入 WDTE 的非“ACH”值
3. 不能用 1 位存储器操作指令对 WDTM 进行操作。
 4. 当使用自写入方式对 flash 存储器进行操作时，为看门狗设置足够的溢出时间 (例如 一字节写入：最小 200 μ s，一个块删除：最小 10 ms)。

- 备注**
1. f_{RL}: 低速内部振荡时钟振荡频率
 2. f_x: 系统时钟振荡频率
 3. x: 不考虑
 4. 图中括号里的值是 f_{RL} = 480 kHz (MAX.)，f_x = 10 MHz. 时的操作结果。

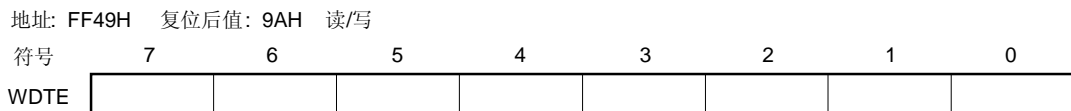
(2) 看门狗定时器使能寄存器 (WDTE)

将 ACH 写入 WDTE 可对看门狗定时器计数器清零并再次启动计数操作。

可由 8 位存储器操作指令设置该寄存器。

复位输入将该寄存器置 9AH。

图 8-3. 看门狗定时器使能寄存器 (WDTE) 的格式



- 注意事项**
1. 如果一个不是 ACH 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)。

8.4 看门狗定时器的操作

8.4.1 当由选项字节选择为“低速内部振荡不能被停止”时的看门狗定时器操作

看门狗定时器的操作时钟恒定为低速内部时钟

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 位为 1 (WDCS2, WDCS1, WDCS0) = 1, 1, 1)启动操作。看门狗定时器操作不能被停止。

以下为复位释放后的看门狗定时器操作。

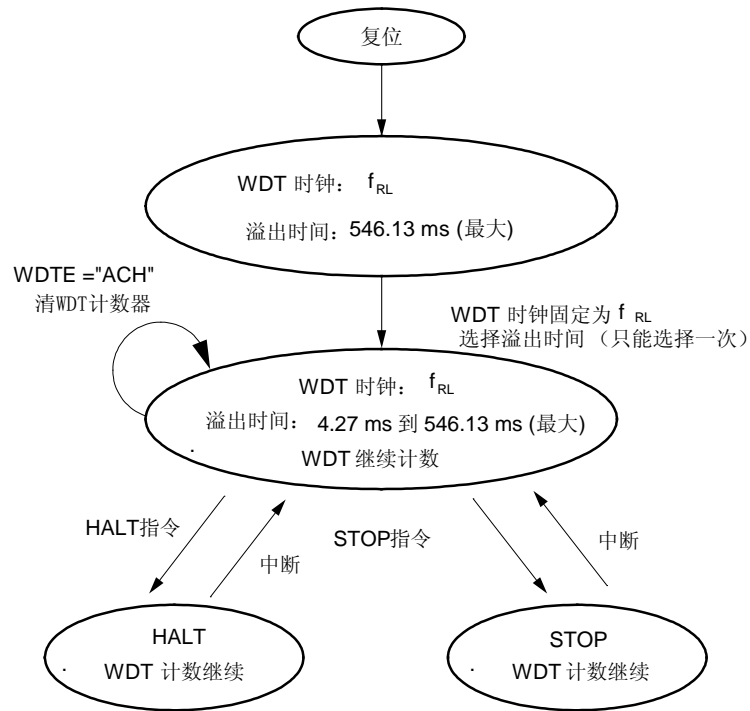
1. 复位后的状态如下。
 - 操作时钟：低速内部时钟
 - ★ • 周期： $2^{18}/f_{RL}$ (546.13 ms: 操作频率 $f_{RL} = 480$ kHz (MAX.))
 - 开始计数
2. 由 8 位存储器操作指令对看门狗定时器模式寄存器(WDTM)中进行以下设置^{注 1, 2}。
 - 周期：用 0~2 位(WDCS2~ WDCS0)设置
3. 在执行上述过程后，将 ACH 写入 WDTE，这样计数器被清零，可重新计数。

- 注
1. 不能修改操作时钟(低速内部时钟)。任何写入 WDTM 第 3 位和第 4 位 (WDCS3, WDCS4)的值将被忽略。
 2. 只要对 WDTM 写入值，看门狗定时器计数器将被清零。

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。

参考如下所示状态转换图

图 8-4. 当由选项字节选择“低速内部振荡不能被停止”时的状态转换图



8.4.2 当由选项字节选择为“低速内部振荡可通过软件停止”时的看门狗定时器操作

可选择看门狗定时器的操作时钟为低速内部振荡时钟或系统时钟。

复位释放后，以低速内部振荡时钟的最大周期(看门狗定时器模式寄存器(WDTM)的第 2, 1 和 0 位 (WDCS2, WDCS1, WDCS0) = 1, 1, 1) 启动操作。

以下为复位后的看门狗定时器操作：

1. 复位释放后的状态如下。
 - 操作时钟：低速内部振荡时钟
 - ★ • 周期： $2^{18}/f_{RL}$ (546.13 ms: 操作频率 $f_{RL} = 480$ kHz (MAX.))
 - 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：由第 3 和第 4 位 (WDCS3 and WDCS4) 选择以下时钟。
 - 低速内部时钟 (f_{RL})
 - 系统时钟 (f_x)
 - 停止看门狗定时器操作
 - 周期：由第 2~0 位 (WDCS2~ WDCS0) 设置
3. 执行上述过程后，将 ACH 写入 WDTE，这样计数器被清零，可重新计数。

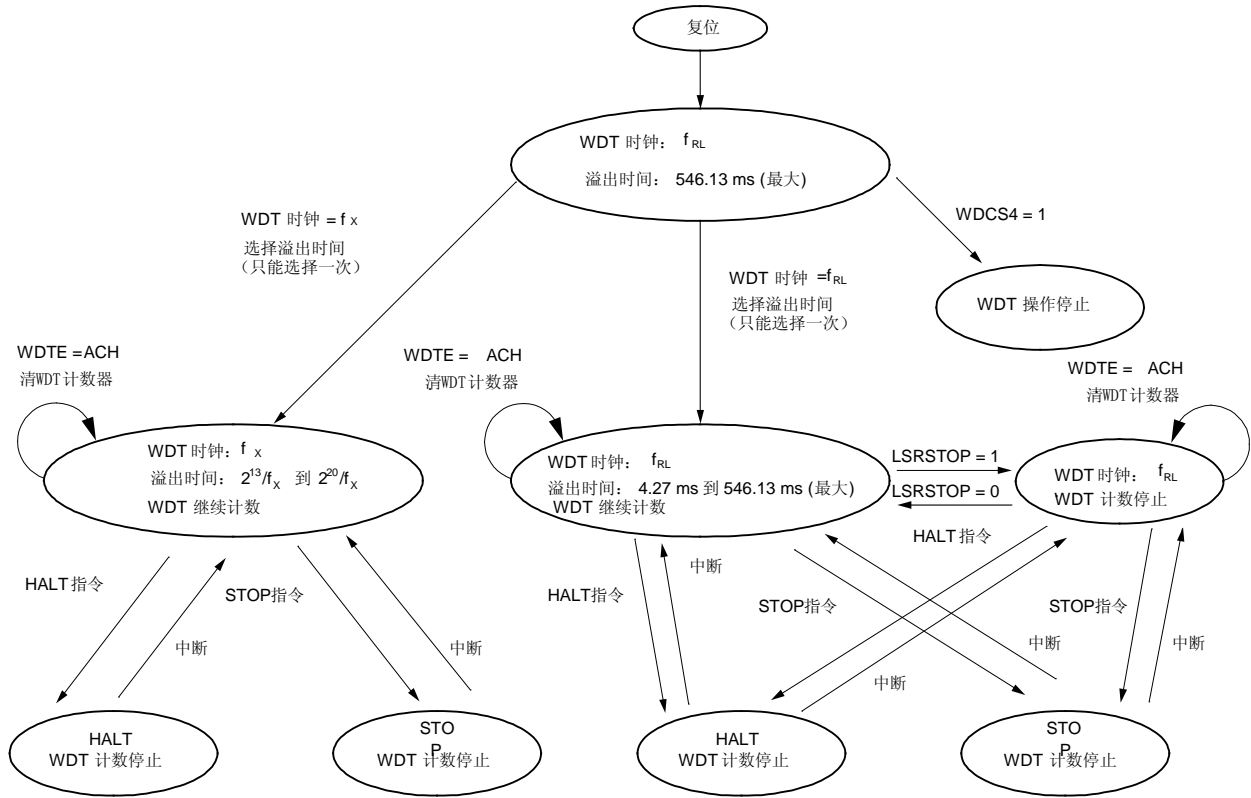
- 注**
1. 只要对 WDTM 进行写入操作，看门狗定时器计数器就将被清零。
 2. 第 7, 6, 5 位分别设置为 0, 1, 1，不能设置其他值。
 3. 如果通过将 WDCS4 和 WDCS3 分别设置为“1”和“x”来停止看门狗定时器操作，即使有下面的操作也不会产生内部复位。
 - 第二次写入 WDTM
 - 用 1 位存储器操作指令对 WDTE 进行操作
 - 给 WDTE 设置一个不同于“ACH”的值

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，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，计数器并没有被清零，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 8.4.3 STOP 模式下看门狗定时器的操作和 8.4.4 HALT 模式下看门狗定时器的操作。

参考如下所示状态转换图

图 8-5. 当由选项字节选择“可通过软件停止低速内部振荡时钟”时的状态转换图



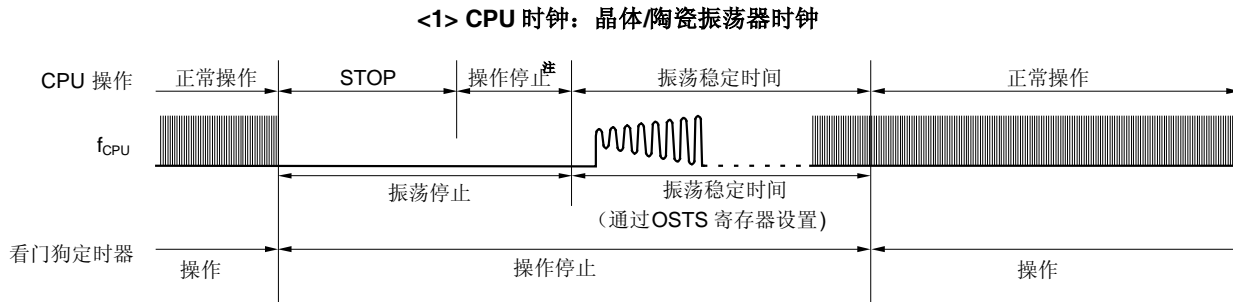
8.4.3 STOP 模式下看门狗定时器的操作(当选项字节选项为“可通过软件停止低速内部振荡时钟”)

无论采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。

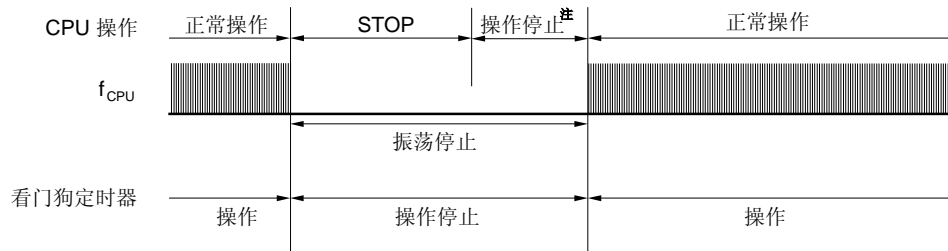
(1) 当看门狗定时器操作时钟是外围硬件时钟(f_x)，且执行 STOP 指令时。

- 当执行 STOP 指令时，要停止看门狗定时器的操作。在释放 STOP 模式后，操作停止 $34\ \mu\text{s}$ (TYP.) (该值是在使用晶体/陶瓷振荡器时由振荡器稳定时间选择寄存器(OSTS)设置的等待振荡器振荡稳定的时间)，然后使用在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 8-6. STOP 模式下的操作(WDT 操作时钟：外围硬件时钟)



<2> CPU 时钟：高速内部振荡时钟或外部时钟输入



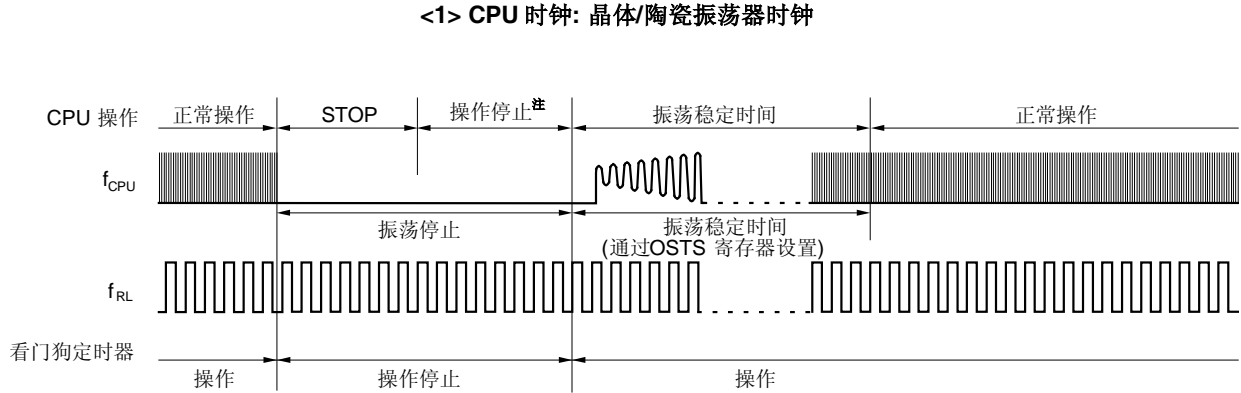
注 操作停止时间是 $17\ \mu\text{s}$ (MIN.), $34\ \mu\text{s}$ (TYP.) 和 $67\ \mu\text{s}$ (MAX.)。

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

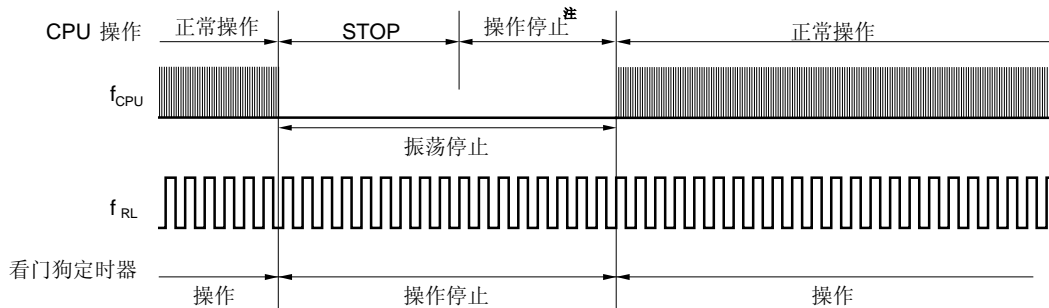
当执行 STOP 指令时, 要停止看门狗定时器的操作。在释放 STOP 模式后, 操作停止 $34 \mu\text{s}$ (TYP.), 然后使用在操作停止前的操作时钟重新开始计数。此时, 计数器没有被清零, 而是保持原值。

★

图 8-7. STOP 模式下的操作(WDT 操作时钟: 低速内部振荡时钟)



<2> CPU 时钟: 高速内部振荡时钟或外部时钟输入

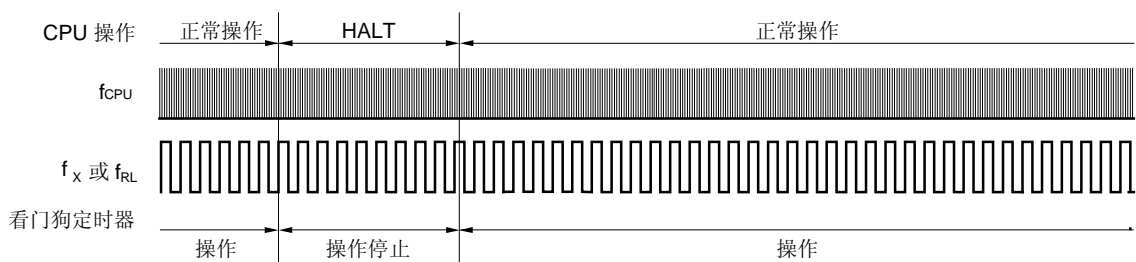


注 操作停止时间是 $17 \mu\text{s}$ (MIN.), $34 \mu\text{s}$ (TYP.)和 $67 \mu\text{s}$ (MAX.)。

8.4.4 HALT 模式下的操作(当选项字节选项为“可通过软件停止低速内部振荡时钟”)

无论采用的是系统时钟(f_x)还是低速内部振荡时钟(f_{RL}), 在执行 HALT 指令时看门狗定时器要停止计数。在释放 HALT 模式后, 使用在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 8-8. HALT 模式下的操作



9.1 A/D 转换器的功能

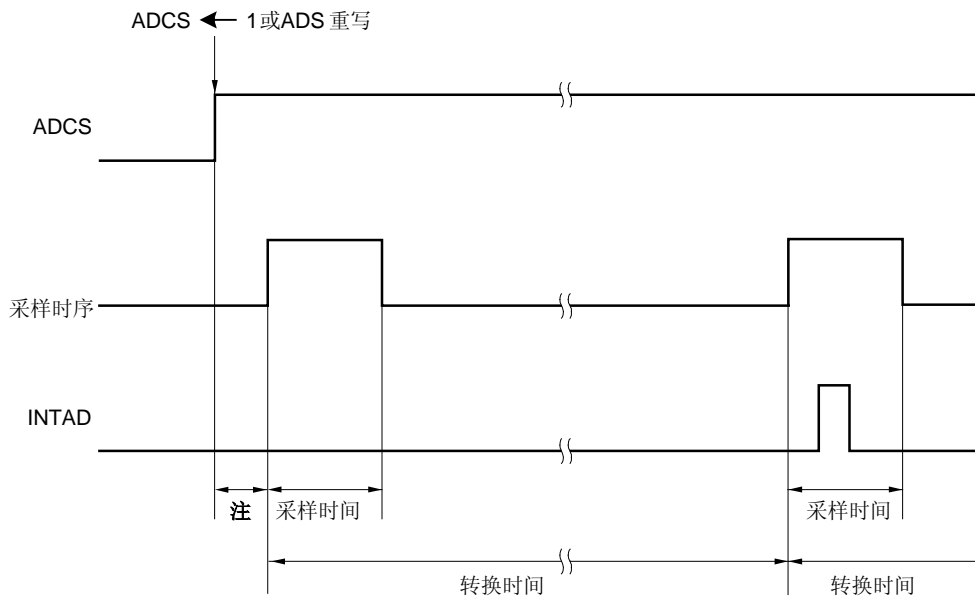
A/D 转换器将模拟信号转换成数字信号，由 4 个通道 (ANI0 ~ ANI3) 组成，具有 10 位分辨率。
A/D 转换器有以下几个功能。

• 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 A/D 转换。每次 A/D 转换结束，都会产生一个中断请求(INTAD)。

图 9-1 为 A/D 采样和转换的时序图，表 9-1 为采样时间和 A/D 转换的时间。

图 9-1. A/D 转换器采样和 A/D 转换的时序



注 ADCS 上升到采样开始需要 2~3 个时钟。

表 9-1. 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 s	4.5 s	1.2 s	3.6 s	0	0	0
AVREF ≥ 4.0 V	24/f _{XP}	72/f _{XP}	3.0 s	9.0 s	2.4 s	7.2 s	1	0	0
AVREF ≥ 2.85 V	96/f _{XP}	144/f _{XP}	12.0 s	18.0 s	9.6 s	14.4 s	1	1	0
	48/f _{XP}	96/f _{XP}	6.0 s	12.0 s	4.8 s	9.6 s	1	0	1
	48/f _{XP}	72/f _{XP}	6.0 s	9.0 s	4.8 s	7.2 s	0	1	0
	24/f _{XP}	48/f _{XP}	3.0 s	6.0 s	禁止设置 (2.4 s)	禁止设置 (4.8 s)	0	0	1
AVREF ≥ 2.7 V	176/f _{XP}	224/f _{XP}	22.0 s	28.0 s	17.6 s	22.4 s	1	1	1
	88/f _{XP}	112/f _{XP}	11.0 s	14.0 s	禁止设置 (8.8 s)	禁止设置 (11.2 s)	0	1	1

注 1. 设置 FR2, FR1 和 FR0 值时, 要与参考电压范围一致, 并且满足以下 2 和 3 的条件。

例如 当 $V_{DD} \geq 2.7 V$, $f_{XP} = 8 MHz$ 时

- 采样时间大于或等于 $11.0 \mu s$, A/D 转换时间大于或等于 $14.0 \mu s$ 并小于等于 $100 \mu s$
- 设置 FR2, FR1 和 FR0 = 0, 1, 1 或 1, 1, 1。

2. 设置采样时间。

- $V_{DD} \geq 4.5 V$: $\geq 1.0 \mu s$
- $V_{DD} \geq 4.0 V$: $\geq 2.4 \mu s$
- $V_{DD} \geq 2.85 V$: $\geq 3.0 \mu s$
- $V_{DD} \geq 2.7 V$: $\geq 11.0 \mu s$

3. 设置 A/D 转换时间。

- $V_{DD} \geq 4.5 V$: $\geq 3.0 \mu s$ 且 $< 100 \mu s$
- $V_{DD} \geq 4.0 V$: $\geq 4.8 \mu s$ 且 $< 100 \mu s$
- $V_{DD} \geq 2.85 V$: $\geq 6.0 \mu s$ 且 $< 100 \mu s$
- $V_{DD} \geq 2.7 V$: $\geq 14.0 \mu s$ 且 $< 100 \mu s$

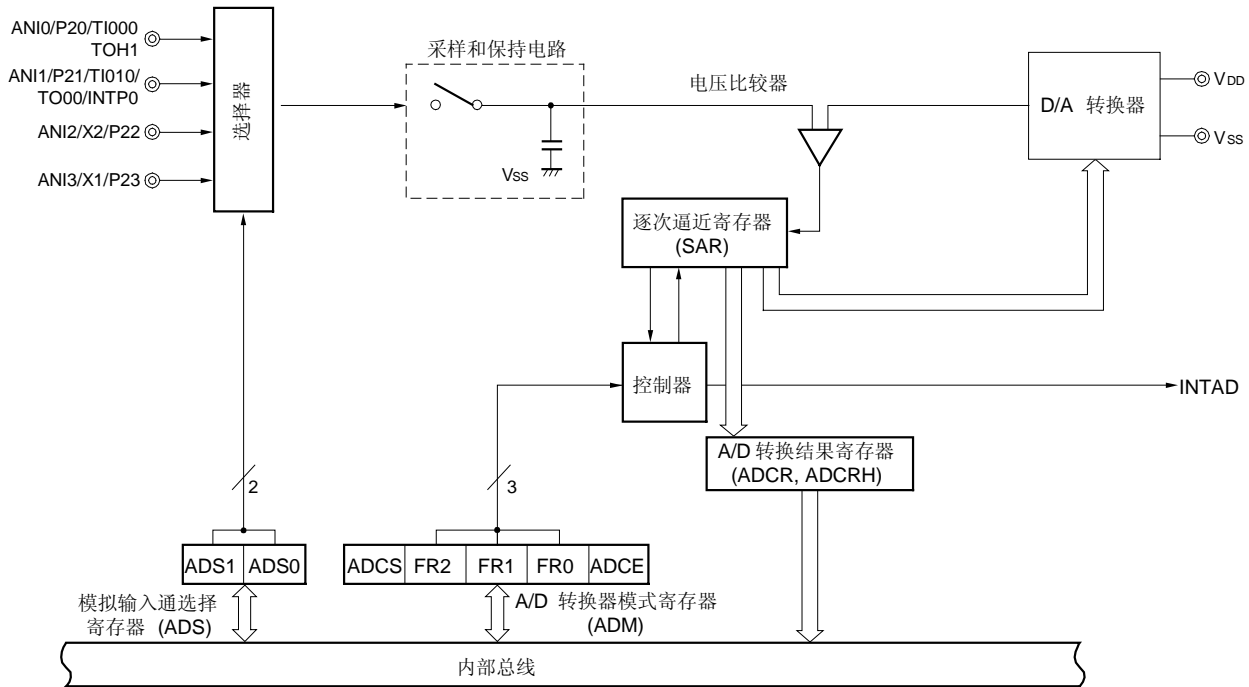
注意事项 以上采样时间和转换时间没有包括时钟频率误差。选择满足上面条件 2 和 3 的采样时间和转换时间时, 要将时钟频率误差考虑在内(高速内部振荡器的最大误差在 $\pm 5\%$)。

备注 1. f_{XP}: 外部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地比较, 直到转换结果输出的时间。

图 9-2 为 A/D 转换器的框图。

图 9-2. A/D 转换器的框图



- 注意事项 1.** 在 78K0S/KU1+ 和 78K0S/KU1+ 中, V_{SS} 复用为 A/D 转换器的接地端。必须将 V_{SS} 与 $GND (= 0 V)$ 相连。
- 2.** 78K0S/KU1+ 中, V_{DD} 复用为 A/D 转换器的参考电源输入。当使用 A/D 转换器时, 所使用的电源电压必须使 V_{DD} 保持稳定 (2.7 ~ 5.5 V)。

9.2 A/D 转换器配置

A/D 转换器由以下硬件组成。

(1) ANI0 ~ ANI3 引脚

ANI0 ~ ANI3 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寄存器 (ADS) 选为模拟输入通道的引脚外, 其他引脚均可用作输入端口引脚。

(2) 采样 & 保持电路

在 A/D 转换开始后, 采样 & 保持电路对由选择器选择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 V_{DD} 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模拟输入电压和 D/A 转换器的输出电压。

(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，并转换结果，从最高有效位(MSB)开始。

当电压转换为数字值，写入最低有效位 (LSB) (A/D 转换结束)，将 SAR 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，并由该寄存器将转换结果存在低 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，该寄存器只保留转换结果的高 8 位。

(8) 控制器

A/D 转换结束，产生中断 INTAD。

(9) V_{DD} 引脚

该引脚为电压引脚。

在 78K0S/KU1+ 和 78K0S/KU1+ 中，V_{DD} 复用为 A/D 转换器的参考电源输入。当使用 A/D 转换器时，所使用的电源电压必须使 V_{DD} 保持稳定 (2.7 ~ 5.5 V)。

(10) V_{SS} 引脚

该引脚为地引脚。

在 78K0S/KU1+ 和 78K0S/KU1+ 中，V_{SS} 复用为 A/D 转换器的接地端，必须将 V_{SS} 与 GND (= 0 V) 相连。

(11) A/D 转换模式寄存器 (ADM)

此寄存器用来设置被转换的模拟输入信号的转换时间，和开始、停止转换操作。

(12) 模拟输入通道选择寄存器 (ADS)

此寄存器用来选择将模拟输入电压转换为数字信号的输入端口。

(13) 端口模式控制寄存器 2 (PMC2)

此寄存器用来定义 P20/ANI0/TI000/TOH1，P21/ANI1/TI010/TO00/INTP0，P22/ANI2 和 P23/ANI3 引脚为 A/D 转换器的模拟输入引脚。

9.3 转换器中使用的寄存器

A/D 转换器使用以下六种寄存器：

- A/D 转换模式寄存器 (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)
- 端口模式寄存器 2 (PM2)
- 端口模式控制寄存器 2 (PMC2)

(1) A/D 转换器模式寄存器 (ADM)

此寄存器用来设置模拟输入的转换时间，和开始/停止转换操作。

可通过 1 位或 8 位存储器操作指令设置 ADM。

复位后该寄存器清零(00H)。

图 9-3. 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 ^{注1}	开始转换操作	

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

ADCE	比较操作控制 ^{注5}	
0 ^{注1}	停止比较操作控制	
1	使能比较操作控制	

备注

1. f_{XP}: 外部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成功地比较, 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范围之内, 因而应该忽略它。

2. 设置 FR2, FR1 和 FR0 值时, 要与参考电压范围一致, 以使其满足以下 2 和 3 的条件。

例如 当 V_{DD} ≥ 2.7 V, f_{XP} = 8 MHz 时

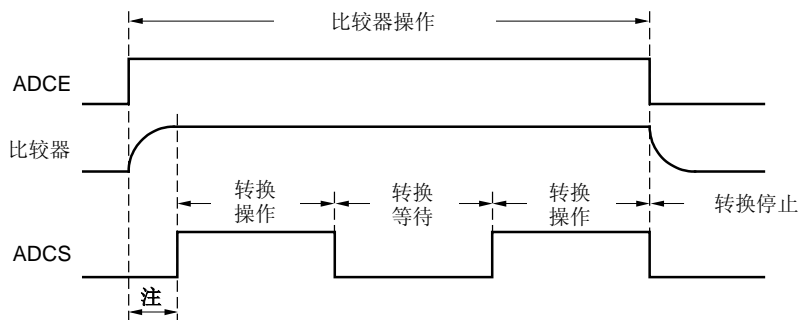
- 采样时间大于或等于 11.0 μs, A/D 转换时间大于或等于 14.0 μs 且小于 100 μs。
- 设置 FR2, FR1 和 FR0 = 0, 1, 1 或 1, 1, 1。

- 注 3. 设置采样时间。
- $V_{DD} \geq 4.5\text{ V}$: $\geq 1.0\ \mu\text{s}$
 - $V_{DD} \geq 4.0\text{ V}$: $\geq 2.4\ \mu\text{s}$
 - $V_{DD} \geq 2.85\text{ V}$: $\geq 3.0\ \mu\text{s}$
 - $V_{DD} \geq 2.7\text{ V}$: $\geq 11.0\ \mu\text{s}$
4. 设置 A/D 转换时间。
- $V_{DD} \geq 4.5\text{ V}$: $\geq 3.0\ \mu\text{s}$ 且 $< 100\ \mu\text{s}$
 - $V_{DD} \geq 4.0\text{ V}$: $\geq 4.8\ \mu\text{s}$ 且 $< 100\ \mu\text{s}$
 - $V_{DD} \geq 2.85\text{ V}$: $\geq 6.0\ \mu\text{s}$ 且 $< 100\ \mu\text{s}$
 - $V_{DD} \geq 2.7\text{ V}$: $\geq 14.0\ \mu\text{s}$ 且 $< 100\ \mu\text{s}$
5. 比较操作通过 ADCS 和 ADCE 来控制, 从开始操作到操作稳定需要 $1\ \mu\text{s}$ 。因此, 从 $\text{ADCE}=1$, 经历 $1\ \mu\text{s}$ 或更长时间, 再将 ADCS 设置为 1, 此时的转换结果比没有等待的转换结果要好。如果 ADCS 设置为 1 前没有等待 $1\ \mu\text{s}$ 或更长时间, 就应该忽略第一次转换的数据。

表 9-3. 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)
0	1	转换等待模式 (仅有比较器电源消耗)
1	×	转换模式

图 9-4. 当使用比较器时的时序图



注 从 ADCE 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没有包括时钟频率误差。选择满足上面条件 2 和 3 的采样时间和转换时间时, 要将时钟频率误差考虑在内(高速内部振荡器的最大误差在 $\pm 5\%$)。
2. 当 A/D 转换结束($\text{ADCS} = 0$)时, 对 ADM 中除了 ADCS 的位进行操作后, 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 $\text{ADCS}=1$, A/D 转换重新开始。
3. 在重写 FR0 至 FR2 时, A/D 转换必须停止 ($\text{ADCS} = 0$)。
4. 一定要把第 6 位、第 2 位和第 1 位清 0。

(2) 模拟输入通道选择寄存器 (ADS)

此寄存器选择被转换的模拟电压输入端口。

可通过 1 位或 8 位存储器操作指令设置 ADS。

复位后该寄存器清零(00H)。

图 9-5. 模拟输入通道选择寄存器 (ADS)的格式



注意事项 ADS 的第 2 位到第 7 位必须清 0。

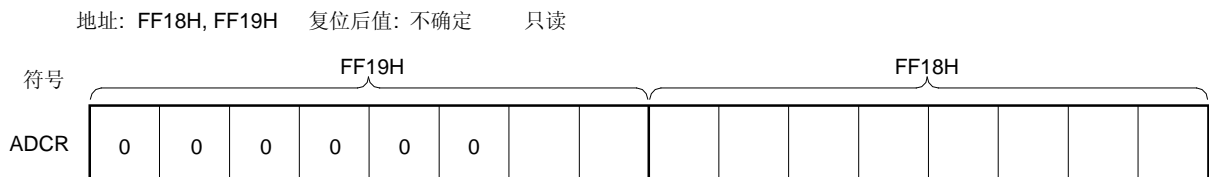
(3) 10 位 A/D 转换结果寄存器 (ADCR)

此寄存器是一个 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，存储从 FF19H 的位 1 开始，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。

复位后 ADCR 的值不确定。

图 9-6. 10 位 A/D 转换结果寄存器 (ADCR)的格式

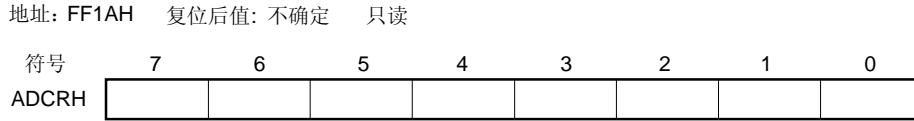


注意事项 当对 A/D 转换模式寄存器(ADM)和模拟输入通道选择寄存器(ADS)进行写操作时，ADCR 的内容可能变得不确定。在对 ADM 和 ADS 执行写操作之前，要在转换完成后读取转换结果。在其他时间读取可能读到错误的转换结果。

(4) 8 位 A/D 转换结果寄存器 (ADCRH)

此寄存器是一个 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。
 可通过 8 位存储器操作指令读取 ADCRH。
 复位后 ADCRH 的值不确定。

图 9-7. 8 位 A/D 转换结果寄存器 (ADCRH) 的格式

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

当使用 P20/ANI0/TI000/TOH1, P21/ANI1/TI010/TO00/INTP0, P22/ANI2 和 P23/ANI3 引脚作为模拟输入时，设置 PM20~PM23 和 PMC20~PMC23 为 1。同时，P20~P23 的输出锁存为 0 或 1。

PM2 和 PMC2 可通过 1 位或 8 位存储器操作指令设置。

复位后 PMC2 为 00H, PM2 为 FFH。

图 9-8. 端口模式寄存器 2 (PM2) 的格式

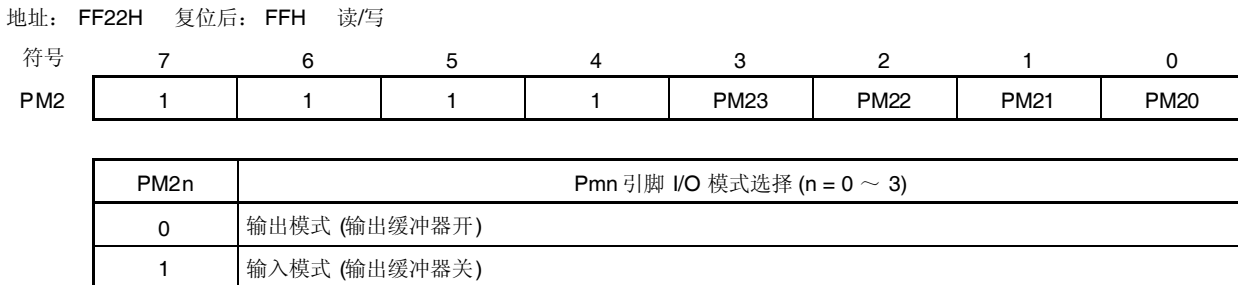
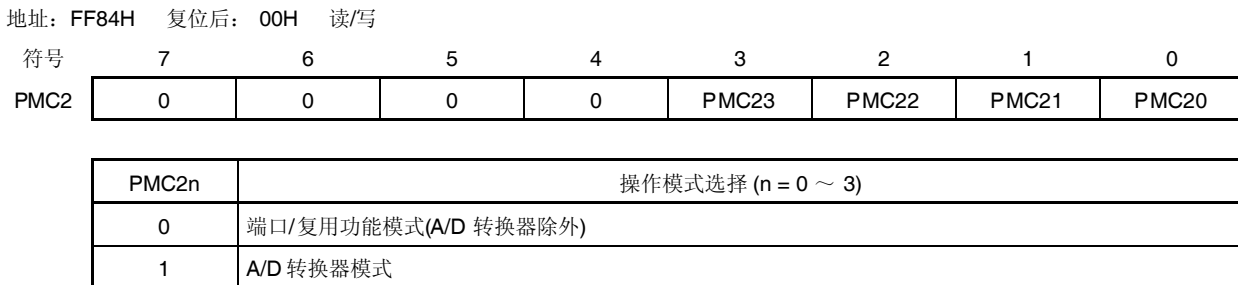


图 9-9. 端口模式控制寄存器 2 (PMC2) 的格式



注意事项 当 PMC20 ~ PMC23 被设置为 1 时，引脚 P20/ANI0/TI000/TOH1，P21/ANI1/TI010/TO00/INTP0，P22/ANI2 和 P23/ANI3 不能用作 A/D 转换器功能之外的任何用途。

这些引脚用作 A/D 转换模式时确保设置上拉电阻选项寄存器(PU20 ~ PU23)为 0。

9.4 A/D 转换器的操作

9.4.1 A/D 转换器的基本操作

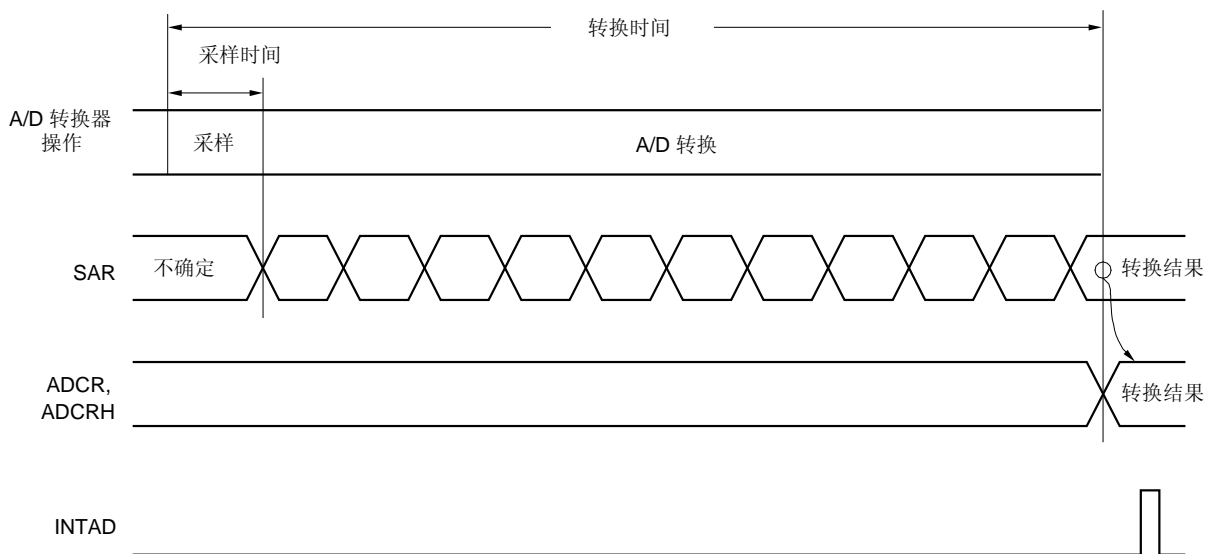
- <1> 使用模拟输入通道选择寄存器(ADS)为 A/D 转换选择一个通道。
- <2> 设置 ADCE 为 1 并等待 $1\ \mu\text{s}$ 或更长。
- <3> 执行两条 NOP 指令或一条相当于两个机器周期的指令。
- <4> 设置 ADCS 为 1 并开始转换操作。
(<5>~<11>操作由硬件实现)
- <5> 采样&保持电路对选中的模拟输入通道的输入电压进行采样。
- <6> 经过一段时间的采样后, 采样&保持电路处于保持状态, 在 A/D 转换结束前, 一直保持输入的模拟电压。
- <7> 设置逐次比较寄存器的第 9 位。通过分接选择器将 D/A 转换电压置为 $(1/2)V_{\text{DD}}$ 。
- <8> 电压比较器比较 D/A 转换电压和模拟输入电压。如果模拟输入电压大于 $(1/2)V_{\text{DD}}$, SAR 中 MSB=1。如果模拟输入电压小于 $(1/2)V_{\text{DD}}$, SAR 中 MSB=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 转换电压。具体模式如下:
 - 第 9 位 = 1: $(3/4)V_{\text{DD}}$
 - 第 9 位 = 0: $(1/4)V_{\text{DD}}$
 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, 并锁存。
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。
ADCS 清零, A/D 转换停止。
如要从 ADCE=1 的状态重新开始 A/D 转换, 从步骤<3>开始。如要从 ADCE=0 的状态重新开始 A/D 转换, 从步骤<2>开始。

- 注意事项**
1. 确保周期 <1> ~ <4> 大于等于 $1\ \mu\text{s}$ 。
 2. <1>和<2>的顺序可以颠倒。

备注 上述过程中使用以下两种类型的 A/D 转换结果寄存器。

- <1> ADCR (16 位): 存储 10 位 A/D 转换值。
- <2> ADCRH (8 位): 存储 8 位 A/D 转换值。

图 9-10. A/D 转换器的基本操作



在软件将 A/D 转换器模式寄存器(ADM)的第 7 位 (ADCS) 复位(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。并且若 ADCS 置 1, 转换将再次从头开始。

产生复位信号后, A/D 转换结果寄存器(ADCR, ADCRH)值不确定。

9.4.2 输入电压和转换结果

输入到模拟输入引脚(AN10~AN13)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。

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

或

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

这里，INT()：该函数返回括号中数值的整数部分

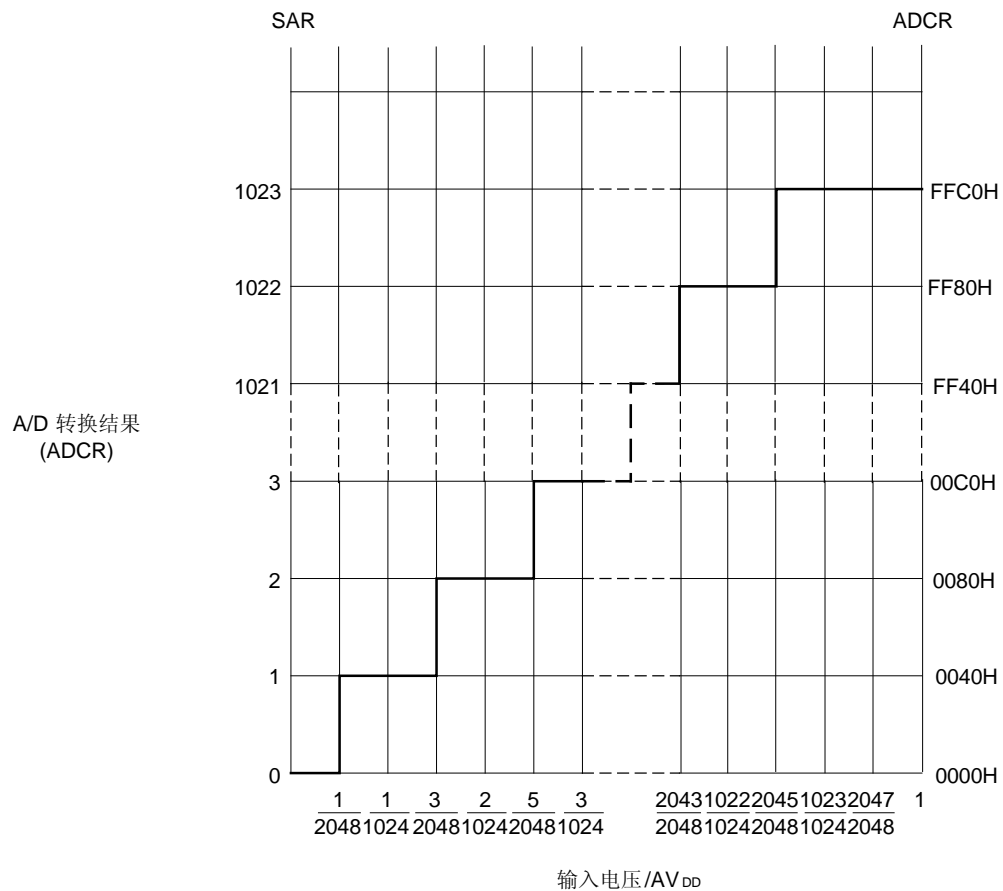
V_{AIN} ：模拟输入电压

V_{DD} ： V_{DD} 引脚电压

ADCR：10 位 A/D 转换结果寄存器 (ADCR) 的值

图 9-11 显示模拟输入电压和 A/D 转换结果之间的关系。

图 9-11. 模拟输入电压和 A/D 转换结果之间的关系



9.4.3 A/D 转换操作模式

由模拟输入通道选择寄存器(ADS)，从 ANI0 ~ ANI3 中选择一个模拟输入通道，并执行 A/D 转换。

(1) A/D 转换操作

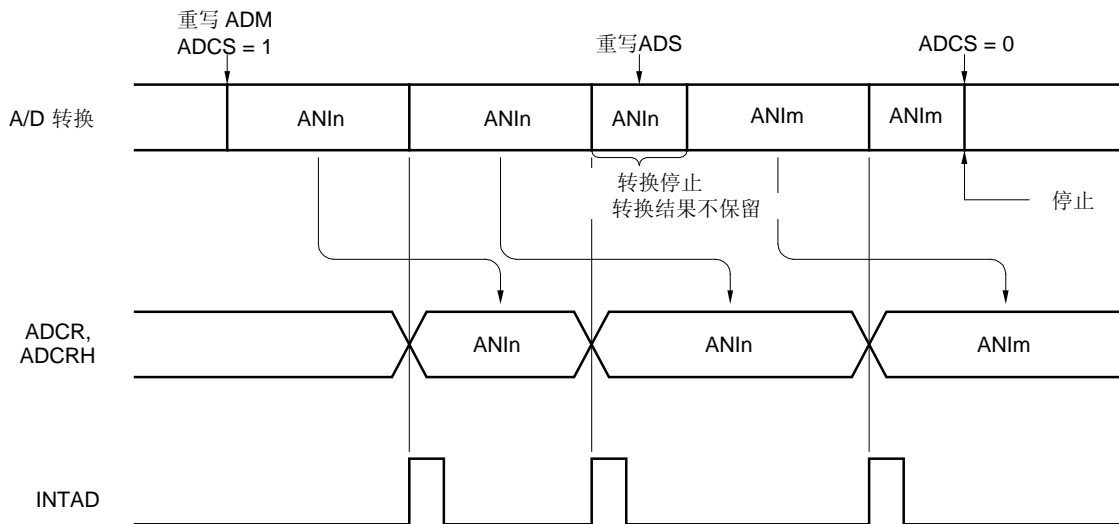
通过设置 A/D 转换模式寄存器 ADM 的第 7 位(ADCS)为 1，可启动电压的 A/D 转换操作。该电压由模拟输入通道选择寄存器(ADS)选择的模拟输入引脚输入。

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)。如果 A/D 转换已经开始而一次 A/D 转换结束时，下一次 A/D 转换立即开始。A/D 转换操作一直持续到有新数据写入到 ADS 为止。

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 被置为 0，则 A/D 转换立即停止。此时的转换结果不确定。

图 9-12. A/D 转换操作



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

设置方式描述如下。

- <1> 设 A/D 转换模式寄存器(ADM)的第 0 位 ADCE=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 到 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 位为 1, 来启动 A/D 转换。
- <5> 产生中断请求信号(INTAD)。
- <6> 将 A/D 转换数据传送至 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 转换数据传送至 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> ~ <4> 的操作时间为 1 μ s 或更长。
 - 2. 若将步骤<1>和<2>次序颠倒, 也不会有问题。
 - 3. <1> 可以被忽略。 但在这种情况下中当步骤<4>执行后忽略第一个转换结果。
 - 4. 步骤<5> 到 <8>所经历的时间与 ADM 的第 5 位和第 3 位(FR2 到 FR0)设置的转换时间不同。步骤<7>~<8>所经历的时间即为转换时间, 由 FR2 到 FR0 设置。

9.5 A/D 转换器特征表的阅读方法

以下，介绍 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。每位数字输出的模拟输入电压的百分比，叫 1LSB(最低有效位)。满度的 1LSB 的百分比用%FSR (满度范围)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、满度误差、积分线性误差和微分线性误差等组合起来表示总误差。

注意量化误差不属于特征表中总误差的范围。

(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。在一个 A/D 转换器中，相差 $\pm 1/2\text{LSB}$ 的模拟输入电压被转换成相同数字代码，因此量化误差不可避免。

注意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 9-13. 总误差

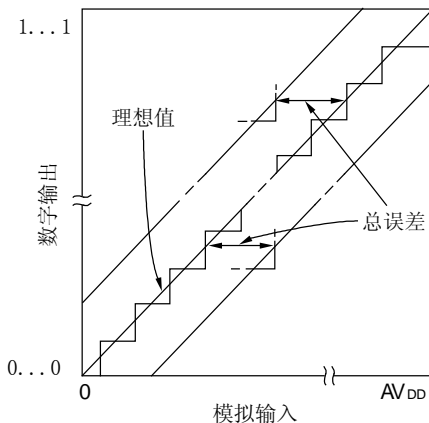
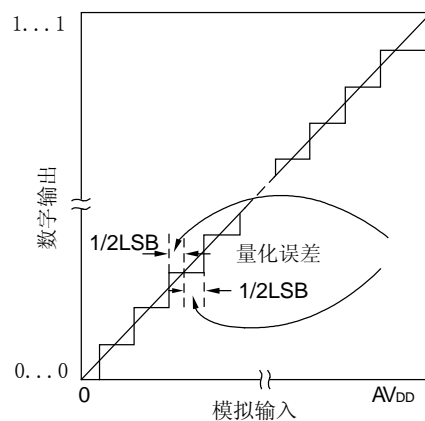


图 9-14. 量化误差



(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) 满度误差

满度误差表示当数字输出范围在 $1\dots\dots110 \sim 1\dots\dots111$ 时, 模拟输入电压的实际测量值和理论值(满度 - $3/2\text{LSB}$)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论线性关系的程度。它表示当零度误差和满度误差为 0 时, 实际测量值和理论线性之间误差的最大值。

(7) 微分线性误差

当代码输出的理论宽度为 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 9-15. 零度误差

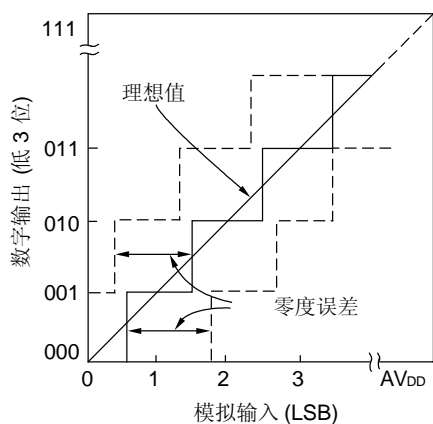


图 9-16. 满度误差

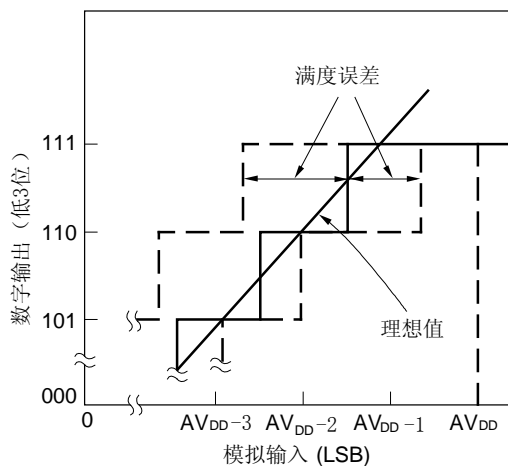


图 9-17. 积分线性误差

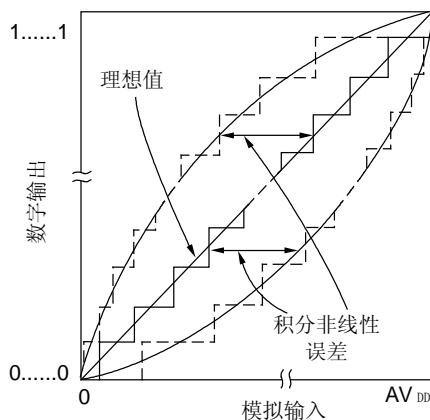
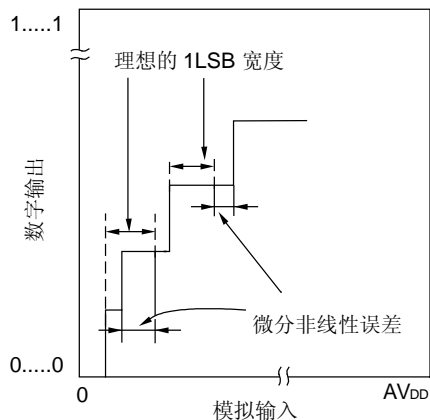


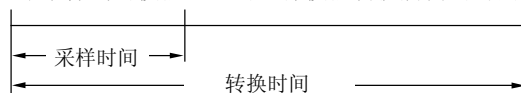
图 9-18. 微分线性误差

**(8) 转换时间**

转换时间表示开始采样到获取数字输出所经历的时间。
采样时间属于特征表中转换时间的范围。

(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。



9.6 A/D 转换器使用注意事项

(1) STOP 模式下的工作电流

在 STOP 模式下, A/D 转换停止操作。此时, 将设置 A/D 转换模式寄存器(ADM)的位 7(ADCS)和位 0(ADCE)清 0, 可以降低操作电流。

(2) ANI0~ANI3 的输入范围

观察 ANI0~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 V_{DD} 或更高, 或者为 V_{SS} 或更低(即在额定的最大范围), 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, 指令对 A/D 转换结果寄存器(ADCR, ADCRH)的写操作和 ADCR, ADCRH 的读操作同时进行, 就为冲突操作。

ADCR, ADCRH 读操作优先。在读操作执行完后, 新的转换结果将写入到 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, 或者对模拟输入通道选择寄存器(ADS)写操作同时进行, 就为冲突操作。

ADM 或 ADS 写操作优先。不执行 ADCR, ADCRH 写操作, 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 V_{DD} 引脚和 ANI0~ANI3 引脚的噪声。

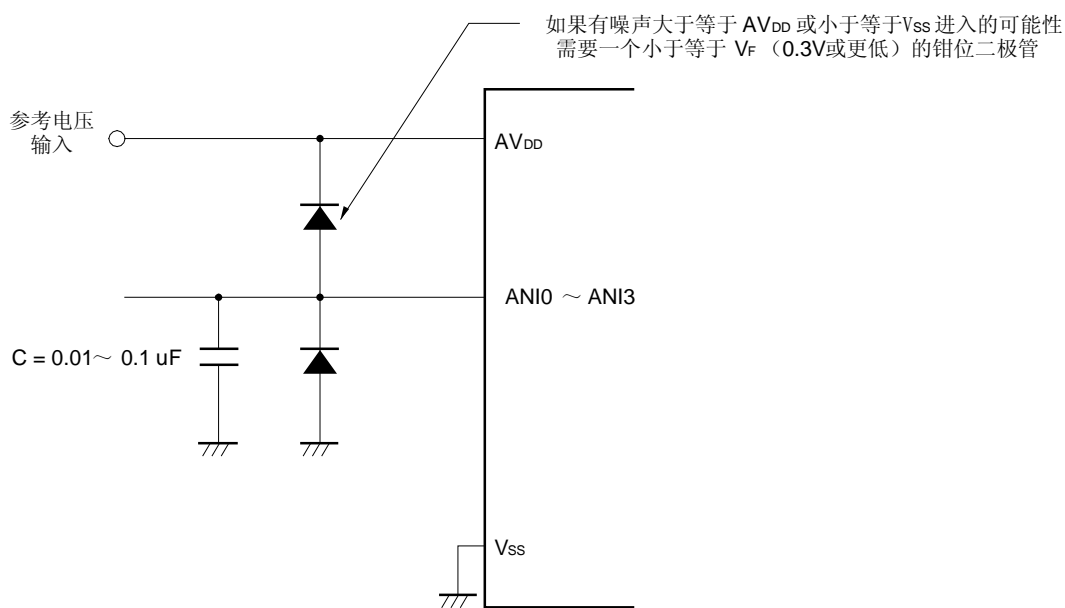
<1> 在低阻抗和高频反馈的电源上连接一个电容。

<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 9-19 所示, 以降低噪声。

<3> 在转换期间, 不要将引脚 ANI0 到 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 9-19. 模拟输入引脚连接



(5) ANI0/P20~ANI3/P23

<1> 模拟输入引脚(ANI0~ANI3) 也可作为输入端口引脚 (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, 否则会降低转换效果。

<2> 如果正在进行 A/D 转换的引脚相邻引脚有数字脉冲, 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

在 A/D 转换器中, 采样期间内内部采样电容充电, 采样时间接近转换时间的 1/6。

由于在采样期间有漏电流和电容充电的电流, 因此输入阻抗在采样期间或其他时候都有波动。

如果在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 1k Ω 或更小, 或者在引脚连接 0.01 μF ~0.1 μF 的电容(见 图 9-19)。

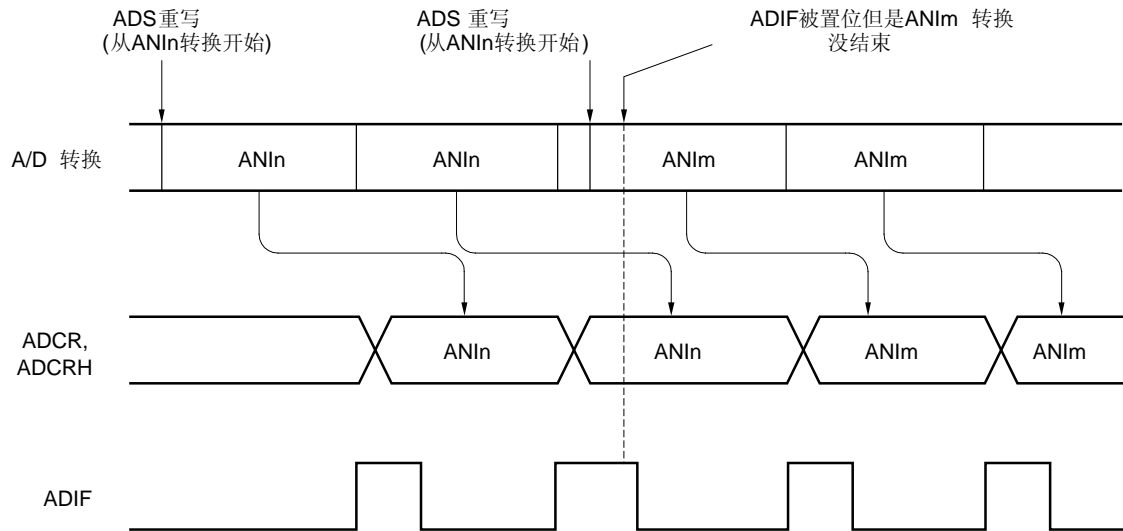
(7) 中断请求信号 (ADIF)

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, 在 A/D 转换期间有模拟输入引脚改变, 在 ADS 修改前, A/D 转换的结果和 ADIF 可能被设置。注意在 ADS 修改后立即读取 ADIF 时, 即使 A/D 转换修改后的模拟输入的 A/D 转换并未结束, 也会设置 ADIF。

在 A/D 转换结束要重新开始转换时, 要在重新开始转换前先将 ADIF 清 0。

图 9-20. 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。可采取措施, 如将 A/D 转换结束中断请求(INTAD)挂起, 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 进行写操作前并且转换结束后, 读取转换结果。若采用和上述不同的时序, 可能会读取到不正确的转换结果。

(10) 转换等待模式的时候的工作电流

在 STOP 模式下操作电流的 DC 特性不满足转换等待模式(只有比较器有功耗), 分别设置 A/D 转换模式寄存器(ADM)的位 7(ADCS)和位 0(ADCE)为 0 和 1。

(11) 内部等效电路

模拟输入模块的等效电路如下所示。

图 9-21. ANIn 引脚的内部等效电路

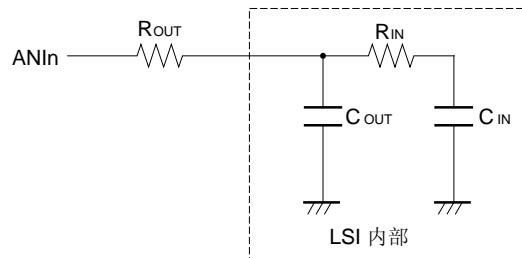


表 9-3. 等效电路的电阻和电容(参考值)

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

备注 1. 表 9-4 所显示的电阻和电容值仅为参考值

2. $n = 0 \sim 3$

3. R_{OUT} : 信号源阻抗

R_{IN} : 模拟输入等效电阻

C_{OUT} : 内部引脚电容

C_{IN} : 模拟输入等效电容

10.1 中断功能类型

有两种中断类型：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，产生一个待机释放信号，并且如果中断是有效的，那么程序根据写入到向量表地址(向量中断服务)的顺序执行。如果多个中断请求同时发生，将根据中断的优先级，优先响应优先级高的中断。如表 10-1 所示。

共有 5 个内部可屏蔽中断源和 2 个外部可屏蔽中断源。

- **复位**

CPU 和 SFR 通过复位信号返回他们的初始状态。复位信号产生的原因如表 10-1 所示。

当产生一个复位信号，程序从写入到 0000H 和 0001H 地址的程序地址开始执行。

10.2 中断源和配置

共有 7 个可屏蔽中断源，和 4 个复位源。(见 表 10-1)。

表 10-1. 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 相等(指定比较寄存器)	内部	000CH	(A)
	5	INTTM000	TM00 和 CR000 相等(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 相等(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	
复位	-	RESET	复位输入	-	0000H	-
		POC	上电清零			
		LVI	低电压检测 ^{注4}			
		WDT	WDT 溢出			

注 1. 优先级是指当有多个中断请求输入时各个中断所对应的优先处理顺序。

1 是最高优先级, 7 为最低优先级。

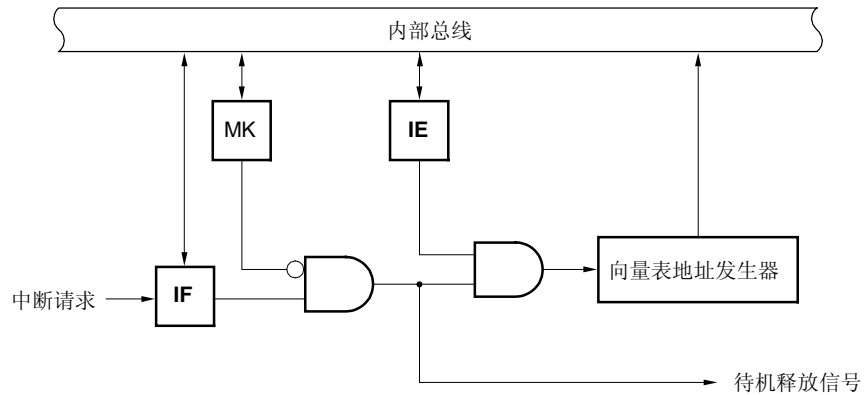
2. (A) 和 (B) 两种基本配置类型, 如图 10-1 中的 (A) 和 (B) 所示。

3. 当低电压检测寄存器 (LVIMD) 的第 1 位 (LVIM) = 0 时。

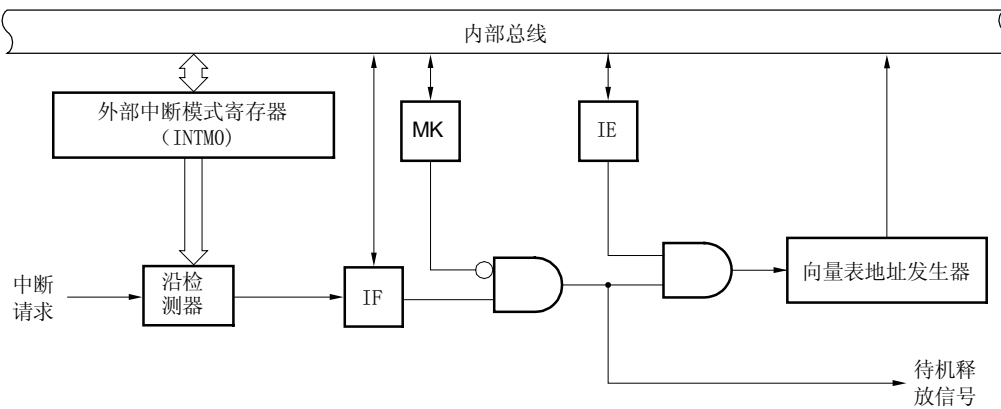
4. 当低电压检测寄存器 (LVIMD) 的第 1 位 (LVIM) = 1 时。

图 10-1. 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

10.3 控制中断功能的寄存器

以下 4 种寄存器用于控制中断功能:

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

表 10-2 列出了中断请求和与其相对应的中断请求标志和中断屏蔽标志。

表 10-2. 中断请求信号及其相应的标志

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

(1) 中断请求标志寄存器 0 (IF0)

当相应的中断请求产生或执行指令时，中断请求标志被置 1。当中断请求被响应或复位信号产生时，通过执行指令将这些标志清零。

可由 1 位或 8 位存储器操作指令设置 IF0。

复位后将 IF0 清零(00H)。

图 10-2. 中断请求标志寄存器 0 (IF0) 的格式

地址: FFE0H 复位后: 00H 读/写

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

××IF×	中断请求标志
0	没有产生中断请求标志
1	产生中断请求，中断请求状态

注意事项 因为 P21 和 P32 有一个复用功能就是作为外部中断输入，所以当指定端口输出模式使输出电平发生变化，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 (MK0)

这些中断屏蔽标志用于允许/禁止相关的可屏蔽中断。

可用 1 位或 8 位存储器操作指令设置 MK0。

复位后将 MK0 设置为 FFH。

图 10-3. 中断屏蔽标志寄存器 (MK0) 的格式

地址: FFE4H 复位后: FFH 读/写

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	0
MK0	ADMK	TMMK010	TMMK000	TMMKH1	PMK1	PMK0	LVIMK	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 P21 和 P32 有一个复用功能就是作为外部中断输入，所以当指定端口输出模式使输出电平发生变化，中断请求标志位置 1。因此，在使用输出模式之前应先将中断屏蔽标志位置 1。

(3) 外部中断模式寄存器 0 (INTM0)

这个寄存器用来设置 INTP0 和 INTP2 的有效沿。

可用 8 位存储器操作指令设置 INTM0。

复位后使 INTM0 清零 (00H)。

图 10-4. 外部中断模式寄存器 0 (INTM0) 的格式

地址: FFECH 复位后: 00H 读/写

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

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	兼有上升沿和下降沿

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	兼有上升沿和下降沿

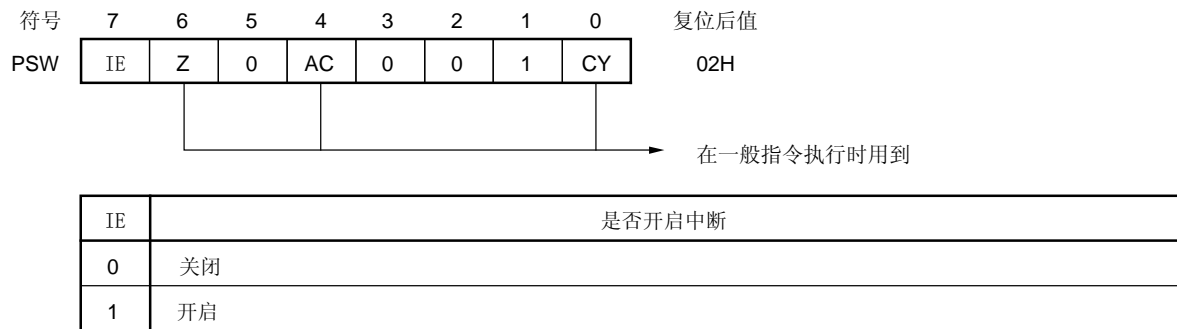
注意事项 1. 寄存器的第 0 位、第 1 位、第 6 位和第 7 位清零。

注意事项 2. 在设置 INTMO 寄存器前，必须设置相应的中断屏蔽标志 (xxMKx = 1) 以禁止中断。INTMO 寄存器设置完成后，首先对中断请求标志 (xxIFx = 0) 进行清零，然后再对中断屏蔽标志 (xxMKx = 0) 进行清零以允许中断。

(4) 程序状态字 (PSW)

PSW 用于保存指令执行结果和中断请求的当前状态。在 PSW 中包含 IE 标志 (允许和禁止可屏蔽中断)。PSW 可用 8 位读写操作指令，还可以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。复位后设置 PSW 为 02H。

图 10-5. 程序状态字 (PSW) 的格式



10.4 中断服务操作

10.4.1 可屏蔽中断请求的响应操作

当中断请求标志被置 1，且与该请求相关的可屏蔽标志被清零时，可以响应这个可屏蔽中断请求。如果处于中断允许状态 (当 IE=1)，向量中断可以被响应。

可屏蔽中断请求产生后开始中断服务所需时间如表 10-3 所示。

图 10-7 和 10-8 为中断请求响应时序。

表 10-3. 从可屏蔽中断请求产生后到开始中断服务所需时间

最小时间	最大时间 ^注
9 个时钟周期	19 个时钟周期

注 此处最大时间是指中断请求产生后到 BT 和 BF 指令执行前等待的最大时间。

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

如果同时产生两个或两个以上的可屏蔽中断请求，则首先响应优先级高的中断请求。

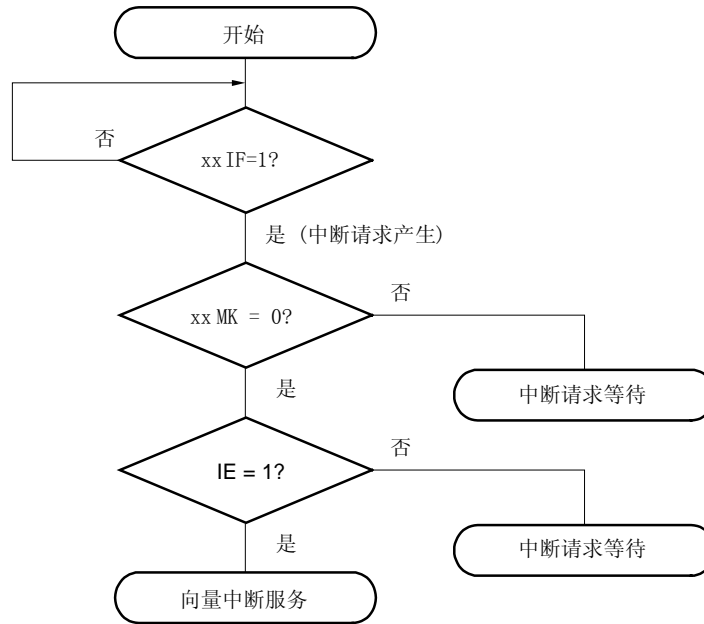
当允许响应中断请求时，处于待机状态中的请求被响应。

图 10-6 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 标志复位清零，通过向量表中的数据来判断响应哪个中断请求并执行响应指令。

可通过执行 RETI 指令从中断返回。

图 10-6. 中断请求响应处理算法

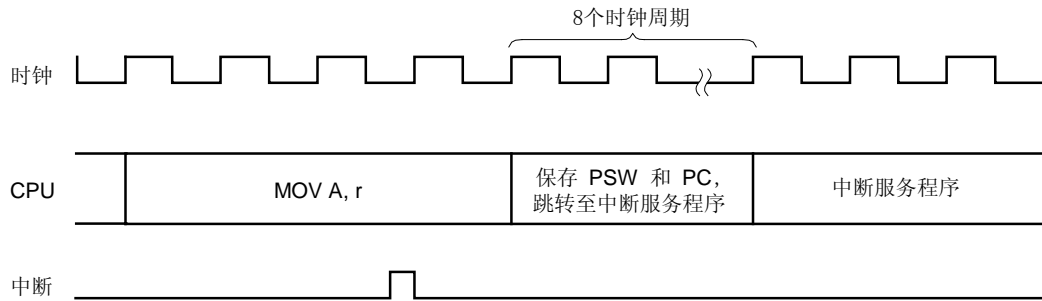


xxIF: 中断请求标志

xxMK: 中断屏蔽标志

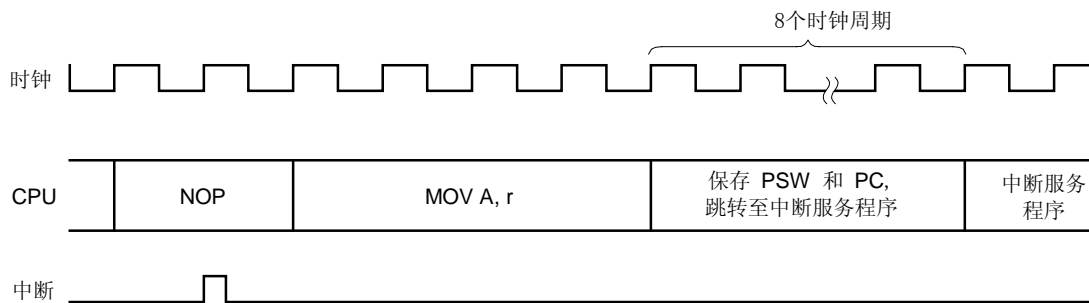
IE: 控制可屏蔽中断请求响应的标志(1=允许, 0=禁止)

图 10-7. 中断请求响应时序 (以 MOV A, r 为例)



如果一条指令执行所需的时钟周期为 n ($n=4-10$)，当执行到 $n-1$ 个时钟周期时中断请求标志 (xxIF) 被置“1”产生中断，则在该指令执行完成后产生中断响应。图 12-8 为 8 位数据传送指令 MOV A, r 中断请求响应时序的示例。该指令执行时钟周期为 4 个时钟周期，中断请求发生在指令执行后的第 3 个时钟周期，中断响应处理在指令 MOV A, r 执行完成后 (即第 4 个时钟周期) 开始。

图 10-8. 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被置位)



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，则中断响应处理在执行下一条指令之后开始。

图 10-8 为 NOP 指令（2 个时钟周期指令）执行时中断请求响应时序，中断请求标志在第 2 个时钟周期被置 1，NOP 指令的下一条指令为 MOV A, r，中断请求响应处理在执行指令 MOV A, r 时开始。

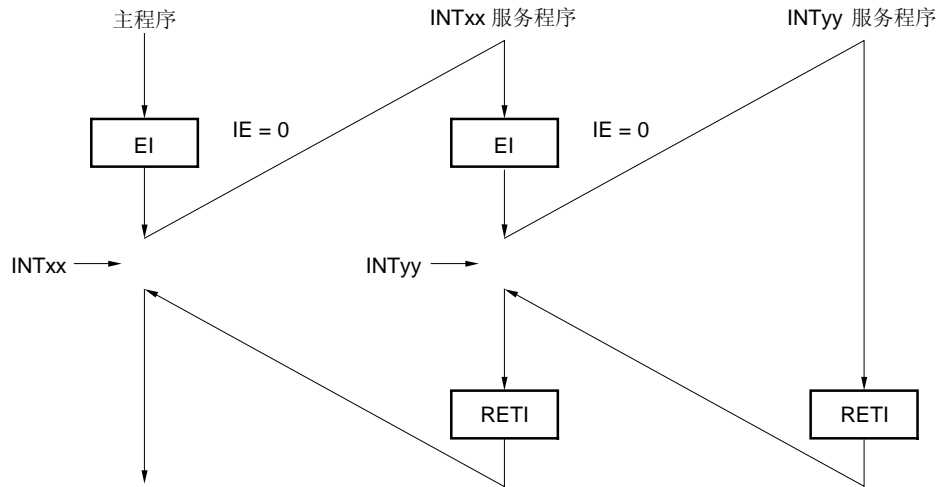
注意事项 当中断请求标志寄存器（IF0）或中断屏蔽标志寄存器（MK0）被访问期间，中断请求等待。

10.4.2 中断嵌套

中断嵌套是指在一个中断服务程序进行的同时使用优先级顺序系统响应另一个中断。当两个或者更多中断同时产生时，根据为每个中断提前指定的优先级执行中断服务程序。

图 10-9. 中断嵌套示例(1/2)

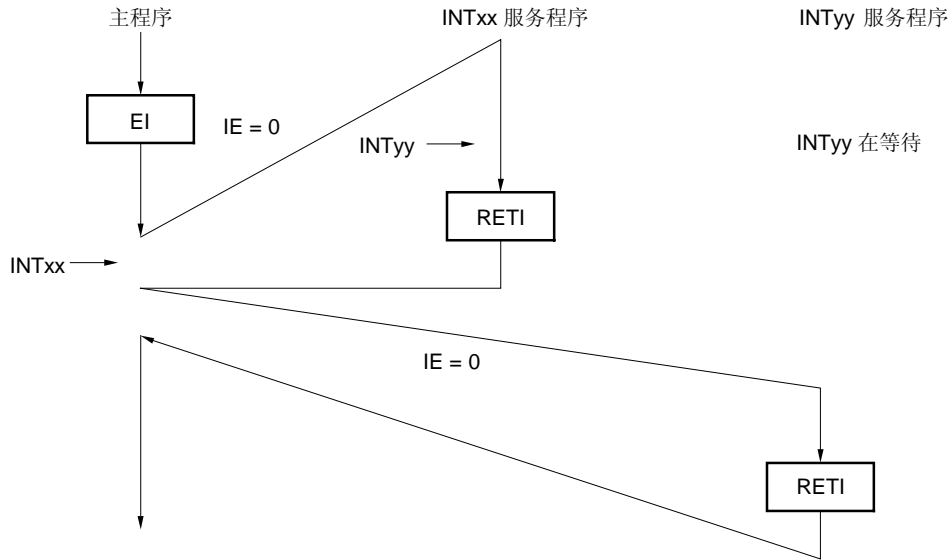
例 1. 响应中断嵌套



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执行 EI 指令，并且设置中断请求响应为允许状态。

注意事项 即使是优先级低的中断也能响应中断嵌套。

例 2. 由于不允许中断没有产生中断嵌套



在 INTxx 中断服务期间不允许中断（不执行 EI 指令），因此不响应中断请求 INTyy，这样也不会产生中断嵌套。中断请求 INTyy 处于等待状态，在执行一条主程序指令后响应该中断请求。

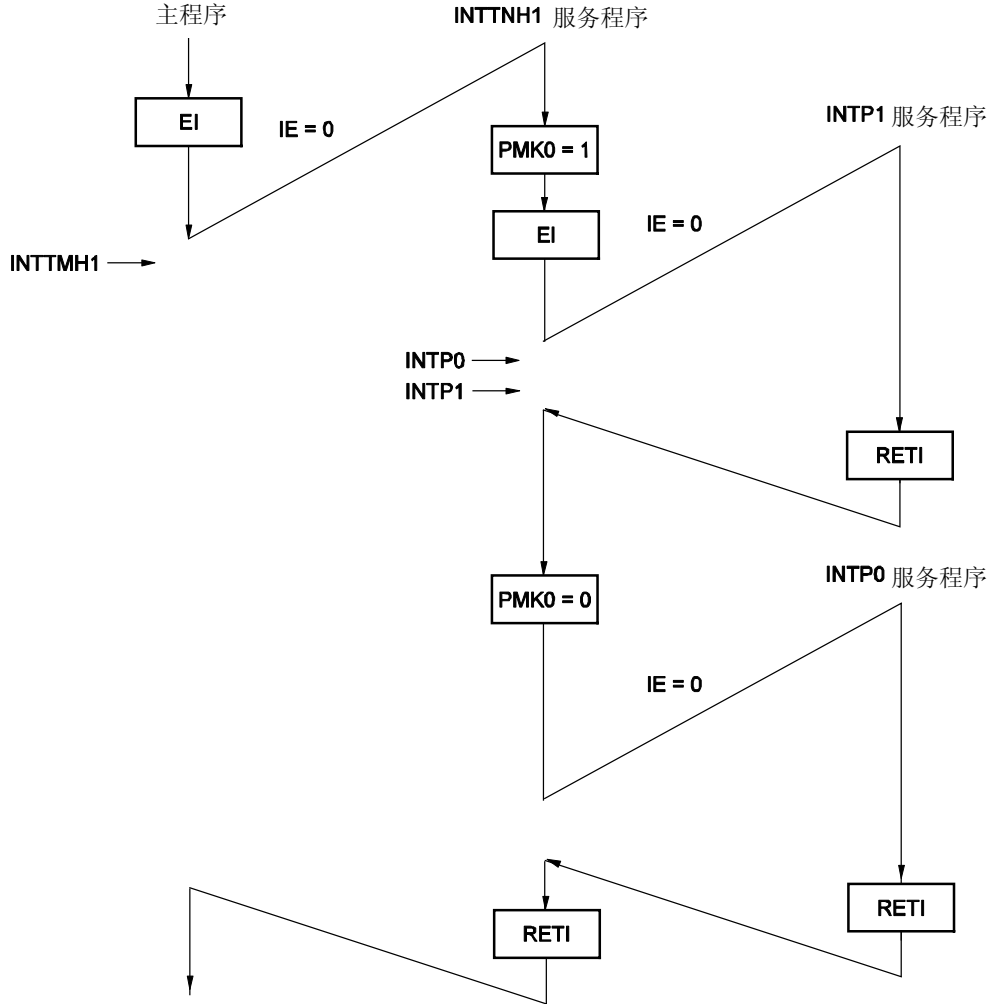
IE = 0: 禁止响应中断请求

图 10-9. 中断嵌套示例(2/2)

例 3. 通过中断嵌套来控制优先级

向量中断允许状态设置如下。

(中断优先级 $INTP0 > INTP1 > INTTMH1$ (参照表 10-1))



在 INTTMH1 中断服务过程中，执行优先级更高的 INTP1 中断服务，因为 INTP0 中断第一被屏蔽。然后，释放了 INTP0 的中断屏蔽，INTP0 进程可以在中断嵌套中执行。

IE = 0: 禁止响应中断请求

10.4.3 中断请求等待

在执行某些指令时，如果有中断请求发生（不管中断请求是可屏蔽中断还是外部中断），应等执行完此条指令和它的下一条指令后才响应中断，此期间中断响应暂时等待。以下为中断请求等待指令。

- 对中断请求标志寄存器 (IF0) 进行操作
- 对中断屏蔽标志寄存器 (MK0) 进行操作

第十一章 待机功能

11.1 待机功能及配置

11.1.1 待机功能

表 11-1. 每种工作状态中操作时钟之间的关系

工作模式 \ 状态	低速内部振荡器			系统时钟	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				振荡	振荡

- 注
1. 当由选项字节选择“不能被停止”低速内部振荡器时。
 2. 当选择了低速内部振荡器“可通过软件停止”时，低速内部振荡器的振荡可被 LSRSTOP 停止。
 3. 如果看门狗定时器的操作时钟为低速内部时钟，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“可被软件停止”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 模式中，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，则内部振荡器继续振荡(参见表 13-1. 由选项字节设置低速内部振荡时钟(是否能通过软件停止))。在此模式中，工作电流不如 STOP 模式中下降得多，但 HALT 模式对于中断请求产生后立即重启操作和进行间断的操作非常有效。

(2) STOP 模式

通过执行 STOP 指令设置 STOP 模式。在 STOP 模式中，系统时钟振荡器停止操作，整个系统的操作终止，这样 CPU 的工作电流将会大幅下降。

可通过中断请求退出该模式，程序可以间断执行。但是，由于在 STOP 模式释放后需要一段等待时间^注以确保振荡器振荡稳定(当使用陶瓷/晶体振荡器时)，因此如果需要响应中断请求后程序立即运行，则应采用 HALT 模式。

注 操作停止时间为 17 μ s(最小),34 μ s(典型),和 67 μ s(最大)。

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. 当改变到 STOP 模式时，一定要在执行 STOP 指令(除了以低速内部振荡时钟工作的外部硬件之外)之前停止外部硬件操作。
 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，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 11-1)。

11.1.2 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、终止以及时钟选择的情况，可参见 **第五章 时钟发生器**。

(1) 振荡稳定时间选择寄存器 (OSTS)

此寄存器用来选择 STOP 模式释放时振荡器时钟的振荡稳定时间。仅当选择晶体/陶瓷振荡时钟作为系统时钟且退出 STOP 模式后，由 OSTS 设置的等待时间才有效。如果选择高速内部振荡或外部时钟输入作为系统时钟源，不用经过等待时间。

由选项字节选择系统时钟振荡器和上电或复位后经过的振荡稳定时间。详情可参见 **第十五章 选项字节**。可由 8 位存储器操作指令设置 OSTS。

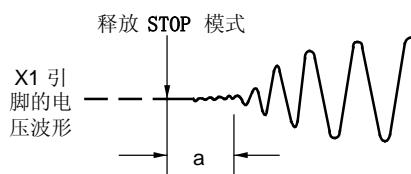
图 11-1. 振荡稳定时间选择寄存器 (OSTS)的格式

地址：FFF4H 复位后：未定义 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 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. 如要设置并退出 STOP 模式，按以下方式设置振荡稳定时间。
预期振荡稳定时间 \leq OSTS 设置的振荡稳定时间
 2. 无论是复位后还是由中断产生退出 STOP 模式，退出 STOP 模式后的等待时间不包括从退出 STOP 模式到时钟振荡启动这段时间 (即下图“a”所示的部分)。



3. 由选项字节选择上电或复位后经过的振荡稳定时间。详情可参见 **第十五章 选项字节**。

- 备注**
1. (): $f_x = 10$ MHz
 2. 通过检查使用的振荡器的特性确定振荡器的振荡稳定时间。

11.2 待机功能的操作

11.2.1 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 11-2. HALT 模式中的工作状态

项目		HALT 模式设置	低速内部振荡器不能被停止 ^注 .	低速内部振荡器可被停止 ^注 .	
				当低速内部振荡继续	当低速内部振荡停止
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 定时器/事件计数器 00			可操作		
8 位 定时器 H1	设置计数时钟为 f_{XP} 到 $f_{XP}/2^{12}$		可操作		
	设置计数时钟为 $f_{RL}/2^7$		可操作	可操作	操作停止
看门狗定 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		可操作 (操作继续)	操作停止	
A/D 转换器			可操作		
POC 电路			可操作		
低电压检测			可操作		
外部中断			可操作		

注 由选项字节可选择低速内部振荡器“不能被停止”或“通过软件停止” (选项字节可参见 第十五章 选项字节)。

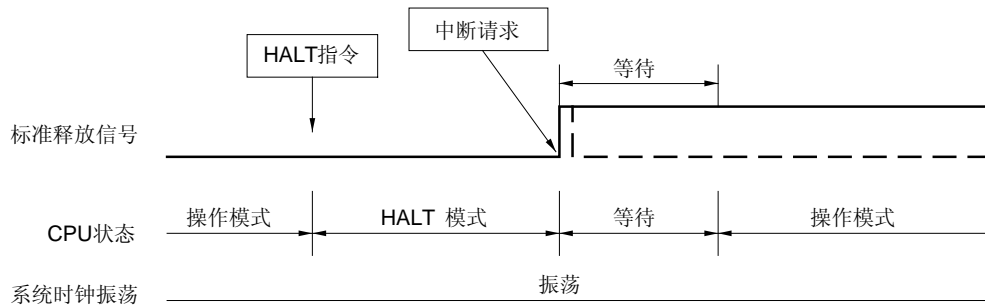
(2) 退出 HALT 模式

可由以下两种中断源退出 HALT 模式。

(a) 由未屏蔽的中断请求释放

当产生一个未屏蔽的中断时，退出 HALT 模式。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。

图 11-2. 通过产生中断请求退出 HALT 模式



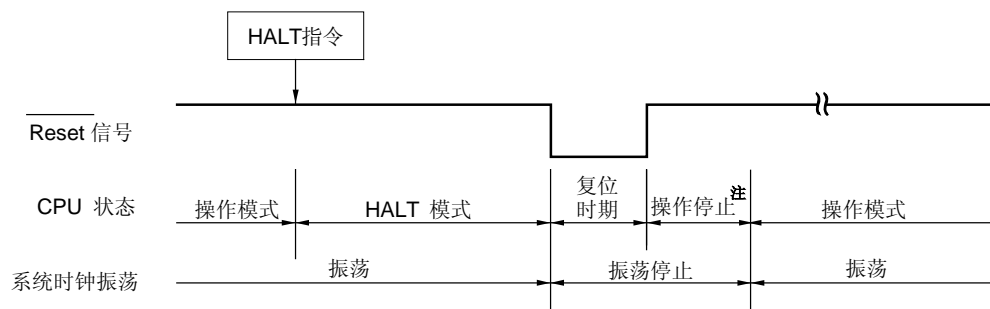
- 备注**
1. 虚线表示退出待机模式的中断请求被响应的情况。
 2. 等待时间如下所示:
 - 执行向量中断服务程序时: 11 ~ 13 个时钟周期
 - 不执行向量中断服务程序时: 3 ~ 5 个时钟周期

(b) 由复位信号退出

当产生复位信号时，释放 HALT 模式，然后在进行正常复位操作后，程序从复位向量指向的地址处执行。

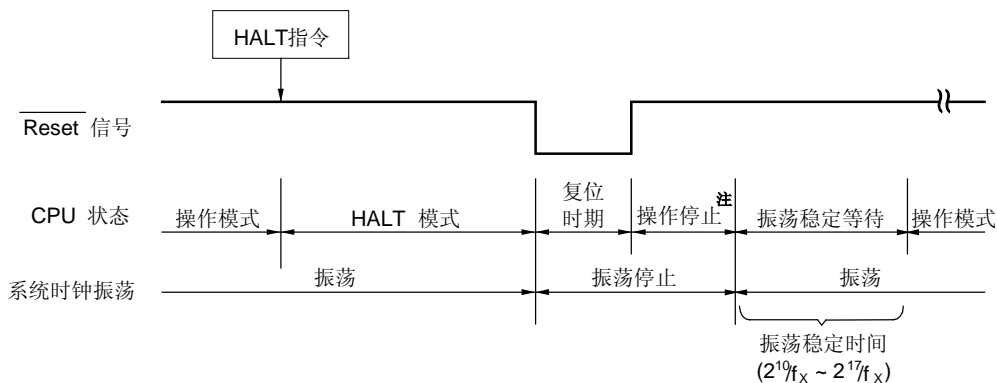
图 11-3. 由复位信号释放 HALT 模式

(1) 当 CPU 时钟为高速内部时钟或外部输入时钟时



注 操作停止(277 μ s(最小),544 μ s(典型),1.075ms(最大)) 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),544 μ s(典型),1.074 ms(最大))参见选项字节。

备注 fx: 系统时钟振荡频率

表 11-3. 在 HALT 模式中中断请求对应的操作

退出 HALT 模式中断源	MK _{xx}	IE	操作
可屏蔽中断请求	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
复位信号产生	-	×	复位处理

x: 不必考虑

11.2.2 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 由于中断请求信号用于释放待机模式，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，然后操作停止 34 μs (典型值) (在经过由振荡稳定时间选择寄存器(OSTS)设置的等待时间后，晶体/陶瓷振荡器启动)。

STOP 模式中的操作状态如下所示。

表 11-4. STOP 模式中的操作状态

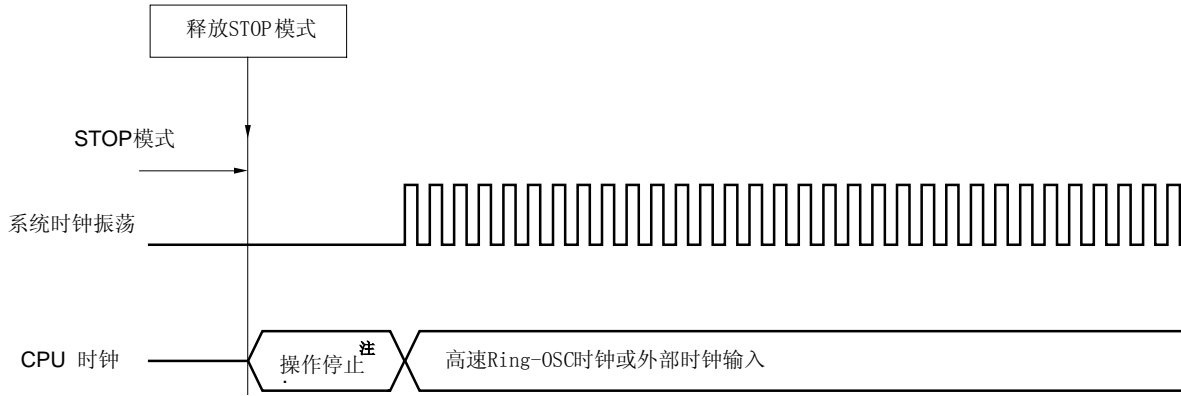
项目		STOP 模式设置	低速内部振荡器不能被停止 ^注 .	
			低速内部振荡器可被停止 ^注 .	
			低速内部振荡继续	低速内部振荡停止
系统时钟			振荡停止	
CPU			终止操作	
端口 (锁存)			保持设置 STOP 模式前的状态	
16 位定时器/事件计数器 00			终止操作	
8 位定时器 H1	设置计数时钟为 f_{XP} 到 $f_{XP}/2^{12}$		终止操作	
	设置计数时钟为 $f_{RL}/2^7$		可操作	终止操作
看门狗定时器	选择“外部硬件时钟”作为操作时钟		禁止设置	终止操作
	选择“低速内部振荡时钟”作为操作时钟		可操作 (操作继续)	终止操作
A/D 转换器			终止操作	
POC 电路			可操作	
低电压检测			可操作	
外部中断			可操作	

注： 选项字节可选择低速内部振荡器“不能被停止”或“通过软件停止” (选项字节可参见 第十五章 选项字节)。

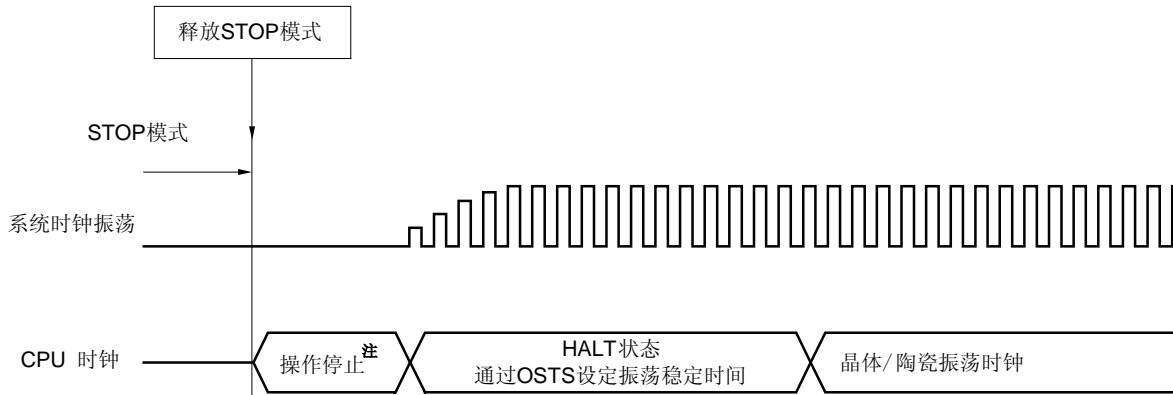
(2) 释放 STOP 模式

图 11-4. 当释放 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 操作停止时间为 $17\mu\text{s}$ (最小), $34\mu\text{s}$ (典型), 和 $67\mu\text{s}$ (最大)。

可由以下两种中断源退出 STOP 模式。

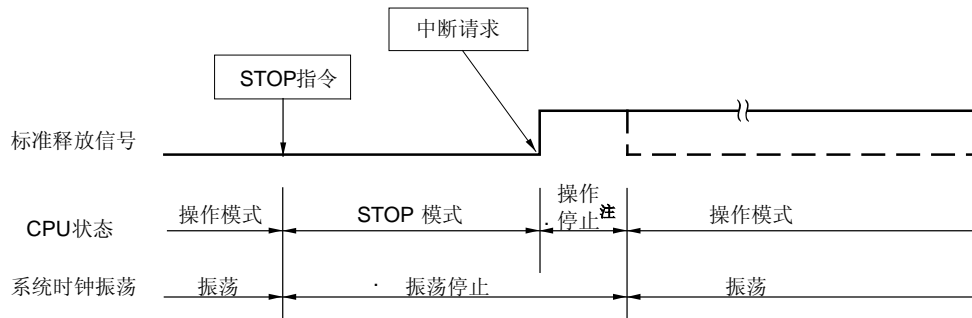
(a) 由未屏蔽的中断请求释放

当产生非屏蔽的中断请求(8 位定时器 H1, 低电压检测器, 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。

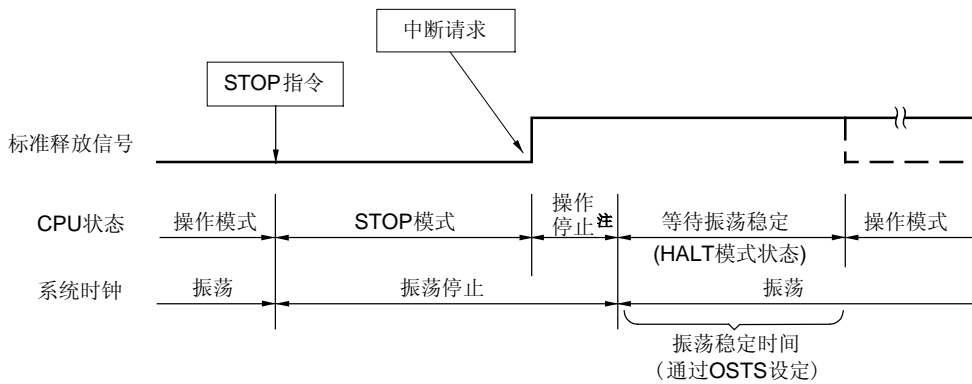
注 仅当计数时钟设置为 $f_{RU}/2^7$ 。

图 11-5. 由中断请求释放 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 操作停止时间为 17 μs (最小), 34 μs (典型), 和 67 μs (最大)。

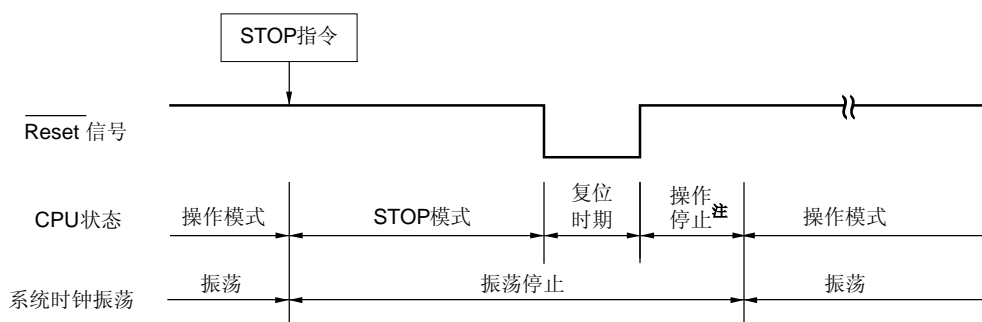
备注 虚线表示退出待机模式的中断请求被响应的情况。

(b) 由复位信号退出

当产生复位信号时，退出 STOP 模式，并在经过振荡稳定时间后执行复位操作。

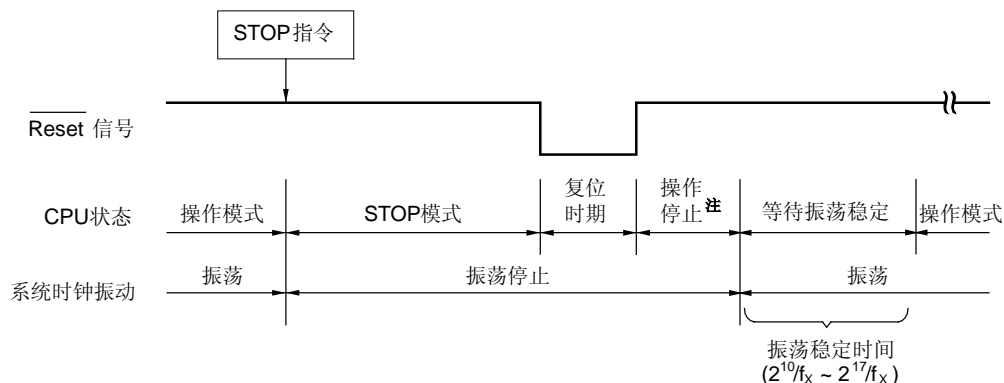
图 11-6. 由复位信号退出 STOP 模式

(1) 如果 CPU 时钟为高速内部振荡时钟或外部输入时钟



注 操作终止(277μs(最小),544μs(典型),1.075ms(最大))参见选项字节。

(2) 如果 CPU 时钟为晶体/陶瓷振荡时钟



注 操作终止(276μs(最小),544μs(典型),1.074ms(最大))参见选项字节。

备注 fx: 系统时钟振荡频率

表 11-5. 在 STOP 模式中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 STOP 模式
复位信号	-	×	进行复位

×: 不必考虑

第十二章 复位功能

以下 4 种操作用于产生复位信号。

- (1) 由 $\overline{\text{RESET}}$ 引脚输入的外部复位信号。
- (2) 由看门狗定时器程序循环检测引起的内部复位。
- (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。

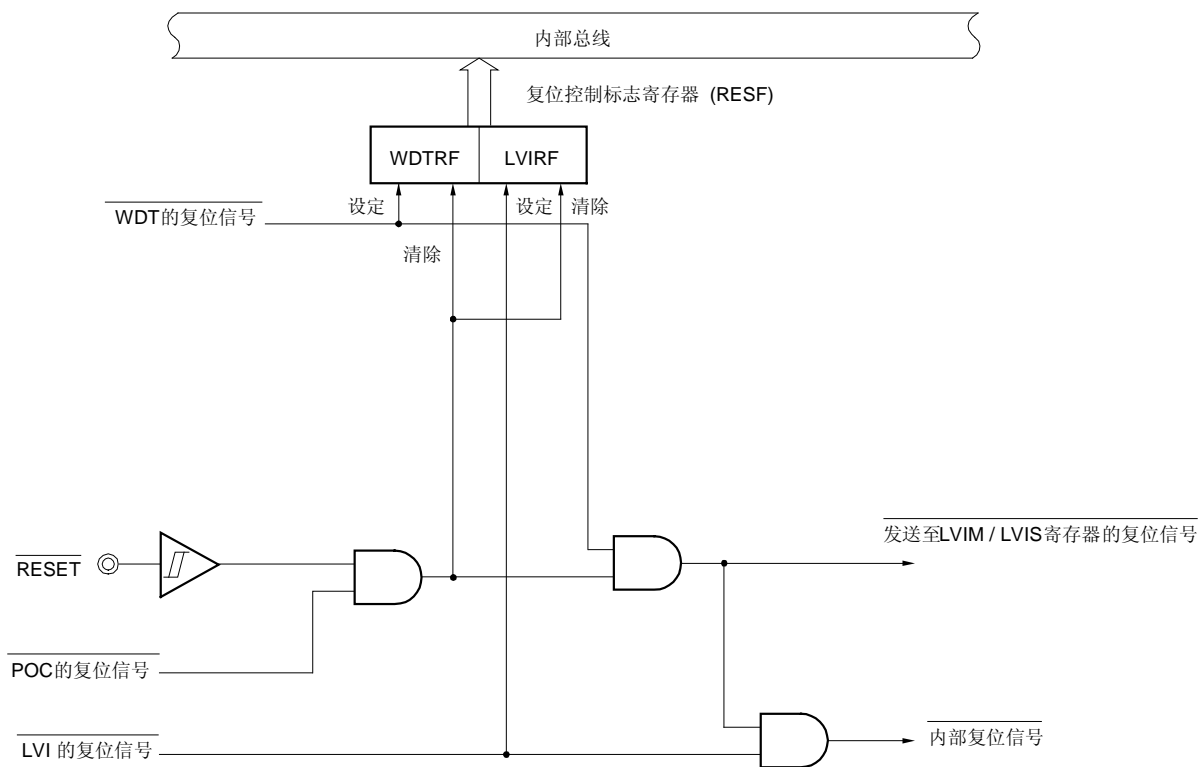
外部复位与内部复位在功能上没有什么区别。在这两种情况中，当输入复位信号时，程序都是从地址 0000H 和 0001H 处开始执行。

当 $\overline{\text{RESET}}$ 引脚输入为低电平或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，硬件的每项设置状态如表 12-1 所示。在复位输入期间或在复位释放后振荡器稳定时间内，每个引脚均为高阻抗，P130(低电平输出)除外。

当 $\overline{\text{RESET}}$ 引脚输入高电平时，复位释放，参考选项字节之后使用 CPU 时钟开始执行程序 (如果选择了晶体/陶瓷振荡器，在参考选项字节经过的时钟振荡稳定时间之后)。由看门狗定时器和时钟监视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，在参考选项字节和经过时钟振荡稳定时间之后)。(参见图 12-2 到 12-4)。由 POC 和 LVI 电路供电检测引起的复位，在复位后，当 $VDD > VPOC$ 或 $VDD > VLVI$ 时自动释放，并且 CPU 开始执行程序 (如果选择了晶体/陶瓷振荡器，在参考选项字节和经过时钟振荡稳定时间之后)。(参见 第十三章 POC 电路 与 第十四章 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. 复位输入期间，系统时钟和低速内部振荡时钟停止振荡。
 3. $\overline{\text{RESET}}$ 引脚只能被复用输入端口引脚(P34)，如果在由 POC 电路释放复位后且在再次参考选项字节之前低电平输入到 $\overline{\text{RESET}}$ 引脚，78K0S/KU1+复位。复位状态保持直到高电平输入到 $\overline{\text{RESET}}$ 引脚。

图 12-1. 复位功能的框图



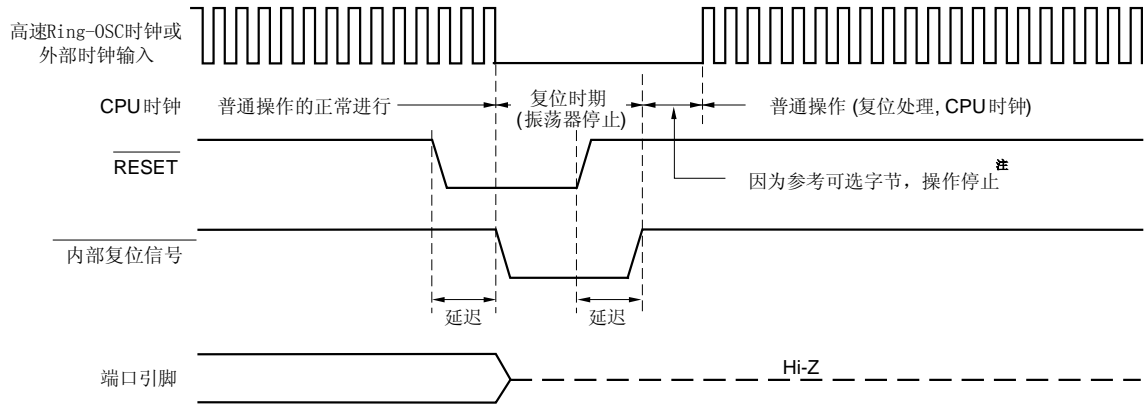
注意事项 LVI 电路的内部复位信号不会使 LVI 电路复位。

- 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

★

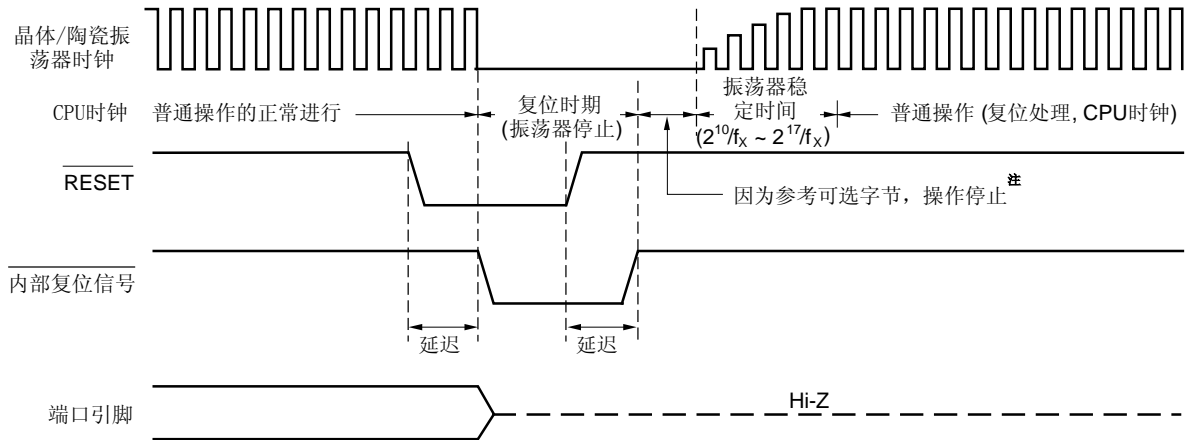
图 12-2. 由 RESET 输入引起的复位时序

<1>高速内部振荡时钟或外部时钟输入



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

<2>晶体/陶瓷振荡时钟

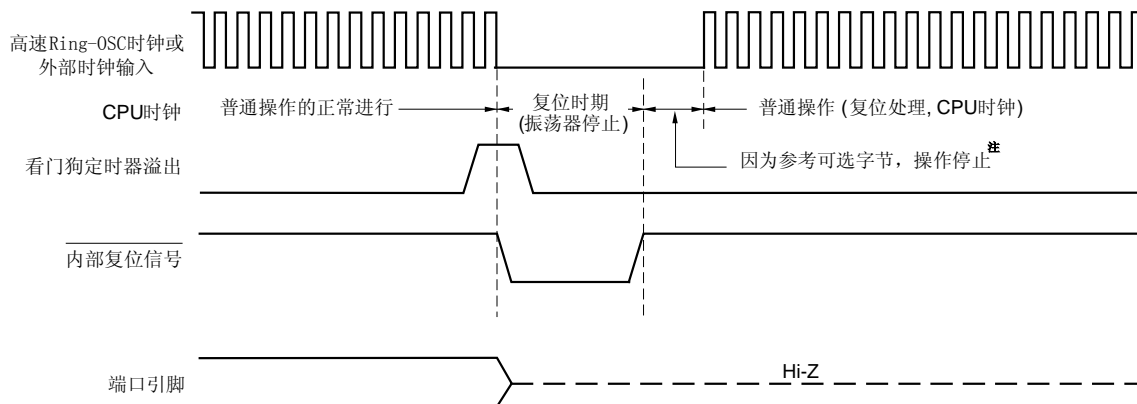


注 操作停止时间为 276 μ S(最小),544 μ S(典型),和 1.074ms(最大)。

备注 f_x : 系统时钟振荡频率

图 12-3. 由看门狗定时器溢出引起的复位时序

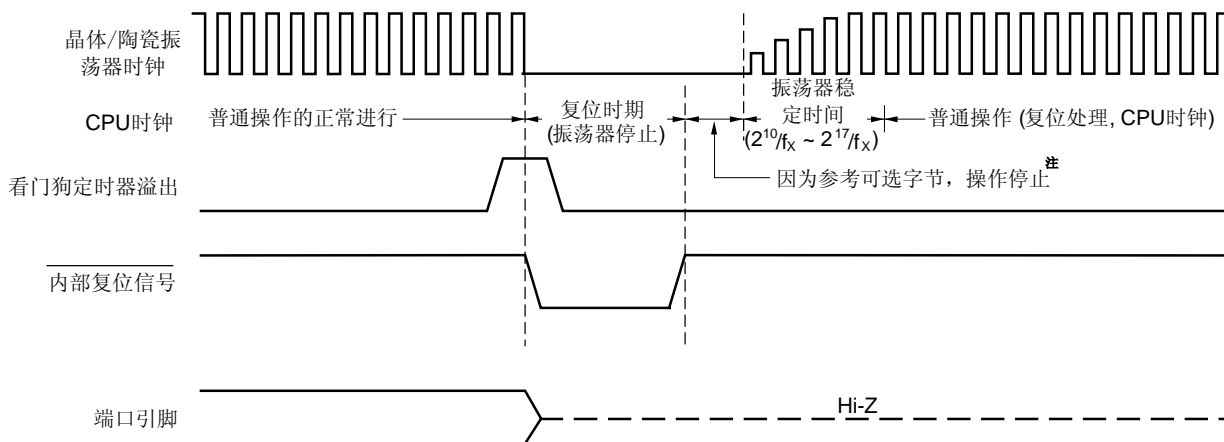
<1> 高速内部振荡时钟或外部时钟输入



注 操作停止时间为 277μs(最小),544μs(典型),和 1.075ms(最大)。

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。

<2> 晶体/陶瓷振荡时钟



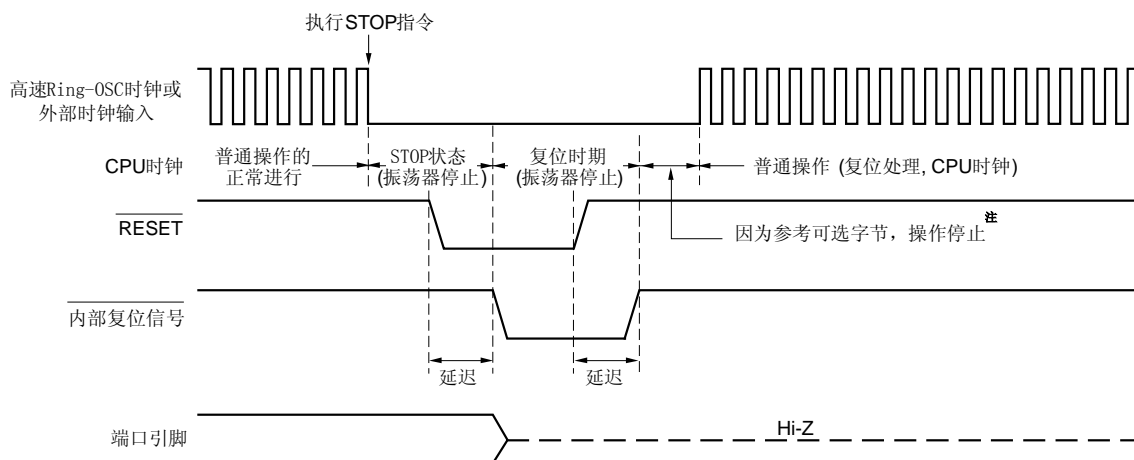
注 操作停止时间为 276μs(最小),544μs(典型),和 1.074ms(最大)。

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 fx: 系统时钟振荡频率

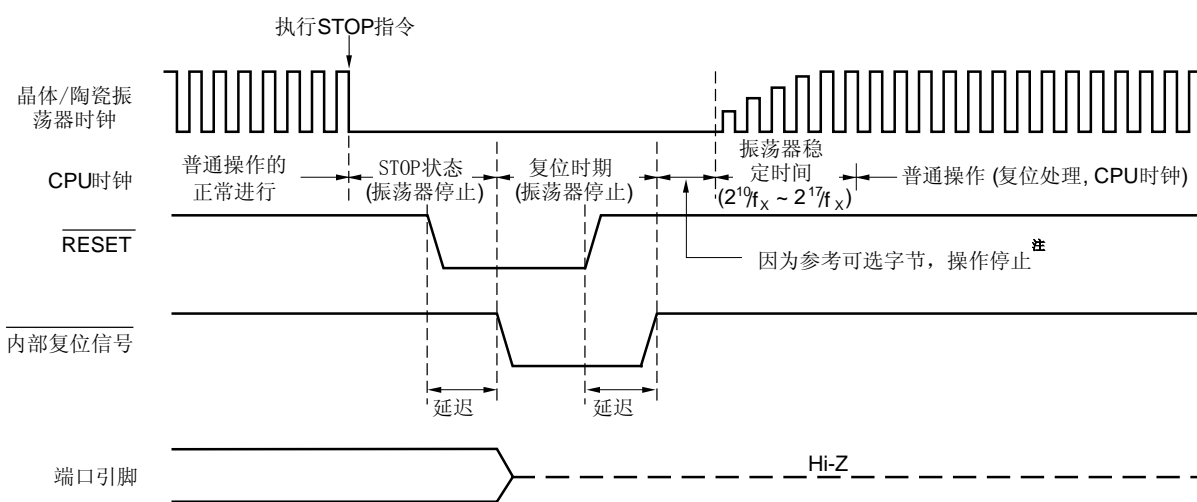
图 12-4. 在 STOP 模式中由 RESET 输入引起的复位时序

<1>高速内部振荡时钟或外部时钟输入



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

<2>晶体/陶瓷振荡时钟



注 操作停止时间为 276 μ s(最小),544 μ s(典型),和 1.074ms(最大)。

备注 1. POC 电路和低电压检测的复位时序，参见 第十三章 POC 电路 和 第十四章低电压检测。
2. f_x : 系统时钟振荡频率

表 12-1. 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}
	通用目的寄存器	不确定 ^{注 2}
端口 (P2 ~ P4) (输出锁存)		00H
端口模式寄存器 (PM2 ~ PM4)		FFH
端口模式控制寄存器 (PMC2)		00H
上拉电阻选择寄存器 (PU2 ~ PU4)		00H
处理器时钟控制寄存器 (PCC)		02H
预处理器时钟控制寄存器 (PPCC)		02H
低速内部振荡模式寄存器 (LSRCM)		00H
振荡稳定时间选择寄存器 (OSTS)		不确定
16 位定时器 00	定时器计数器 00 (TM00)	0000H
	捕捉/比较寄存器 000, 010 (CR000, CR010)	0000H
	模式控制寄存器 00 (TMC00)	00H
	预分频器模式寄存器 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 H1	比较寄存器 (CMP01, CMP11)	00H
	模式寄存器 1 (TMHMD1)	00H
看门狗定时器	模式寄存器 (WDTM)	67H
	使能寄存器 (WDTE)	9AH
A/D 转换器	转换结果寄存器 (ADCR, ADCRH)	Undefined
	模式寄存器 (ADM)	00H
	模拟输入通道选择寄存器 (ADS)	00H

- 注
1. 当产生复位信号和经过振荡稳定时间时，只有 PC 的内容为未定义的。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 12-1. 响应复位后的硬件状态 (2/2)

硬件		复位后的状态
复位功能	复位控制标志寄存器 (RESF)	00H ^注
低电压检测	低电压检测寄存器 (LVIM)	00H ^注
	低电压检测电平选择寄存器 (LVIS)	00H ^注
中断	中断请求标志寄存器 (IF0)	00H
	中断屏蔽标志寄存器 (MK0)	FFH
	外部中断模式寄存器 (INTMO)	00H
Flash 存储器	Flash 保护命令寄存器 (PFCMD)	不确定
	Flash 状态寄存器 (PFS)	00H
	Flash 编程模式控制寄存器 (FLPMC)	不确定
	Flash 编程命令寄存器 (FLCMD)	00H
	Flash 地址指针 L (FLAPL)	不确定
	Flash 地址指针 H (FLAPH)	
	Flash 地址指针 H 比较寄存器 (FLAPHC)	00H
	Flash 地址指针 L 比较寄存器 (FLAPLC)	00H
	Flash 写缓冲寄存器 (FLW)	00H

注 这些值依据复位源的不同而不同如下。

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

12.1 确认复位源的寄存器

78K0S/KU1+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。

可由 8 位存储器操作指令读取 RESF。

$\overline{\text{RESET}}$ 输入、由 POC 电路引起的复位输入以及读取 RESF 可将 RESF 清零。

图 12-5. 复位控制标志寄存器(RESF)的格式

地址: FF54H 复位后: 00H^注 只读

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, 或对 RESF 清零
1	产生内部复位请求

LVIRF	由低电压检测电路产生的内部复位请求(LVI)
0	不产生内部复位请求, 或对 RESF 清零
1	产生内部复位请求

注 复位后的值依据复位源而不同。

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 12-2 所示。

表 12-2. 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)	设置(1)	保持
LVIRF			保持	设置(1)

第十三章 上电清零电路

13.1 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（ $V_{POC} = 2.1V \pm 0.1V$ ），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（ $V_{POC} = 2.1V \pm 0.1V$ ），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，则将复位标志寄存器（RESF）清零。
2. POC 电路的检测电压（ V_{POC} ）值范围为 $2.1V \pm 0.1V$ ，故正常工作电压范围为 $2.2 \sim 5.5 V$ 。

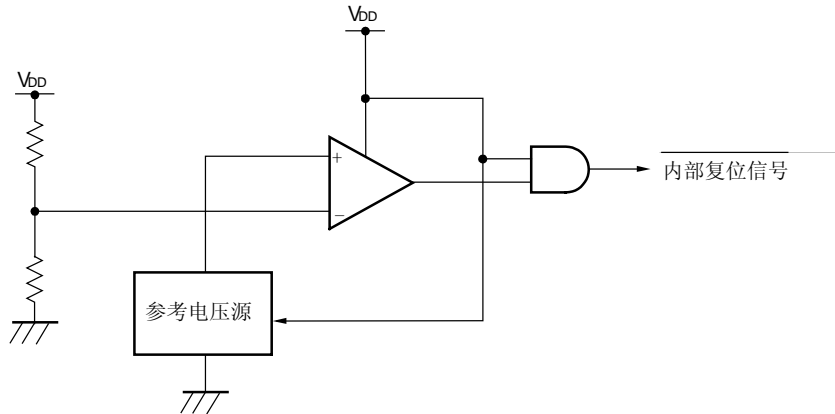
备注 本产品集成了多种用以产生内部复位信号的硬件功能。当看门狗定时器（WDT）和低电压检测电路（LVI）引起复位时，RESF 中有一个标志用于指示复位原因。当 WDT 和 LVI 产生内部复位信号时，RESF 不能被清零，且标志位被置 1。

如需了解 RESF 的详细内容，参见 **第十二章 复位功能**。

13.2 上电清零电路的配置

POC 电路的框图如图 13-1 所示。

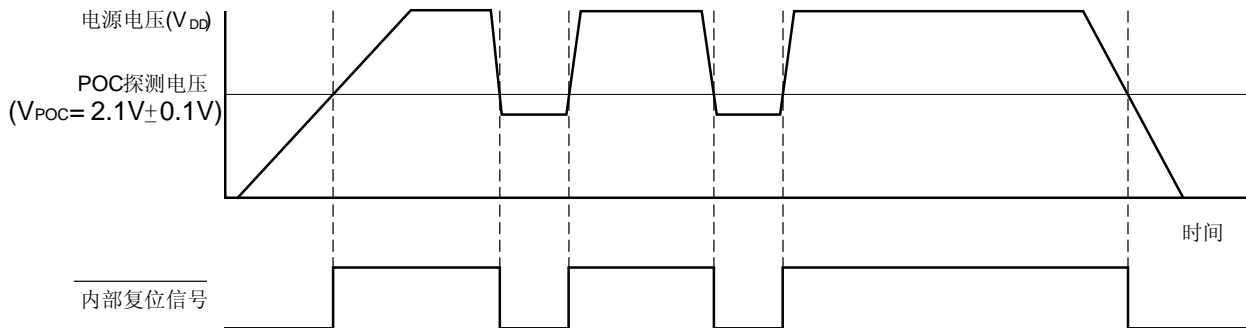
图 13-1. POC 电路的框图



13.3 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 ($V_{POC} = 2.1\text{ V} \pm 0.1\text{ V}$)，当 $V_{DD} < V_{POC}$ 时，产生内部复位信号，当 $V_{DD} \geq V_{POC}$ 时，释放内部复位信号。

图 13-2. POC 电路中产生内部复位信号的时序



13.4 上电清零电路的使用注意事项

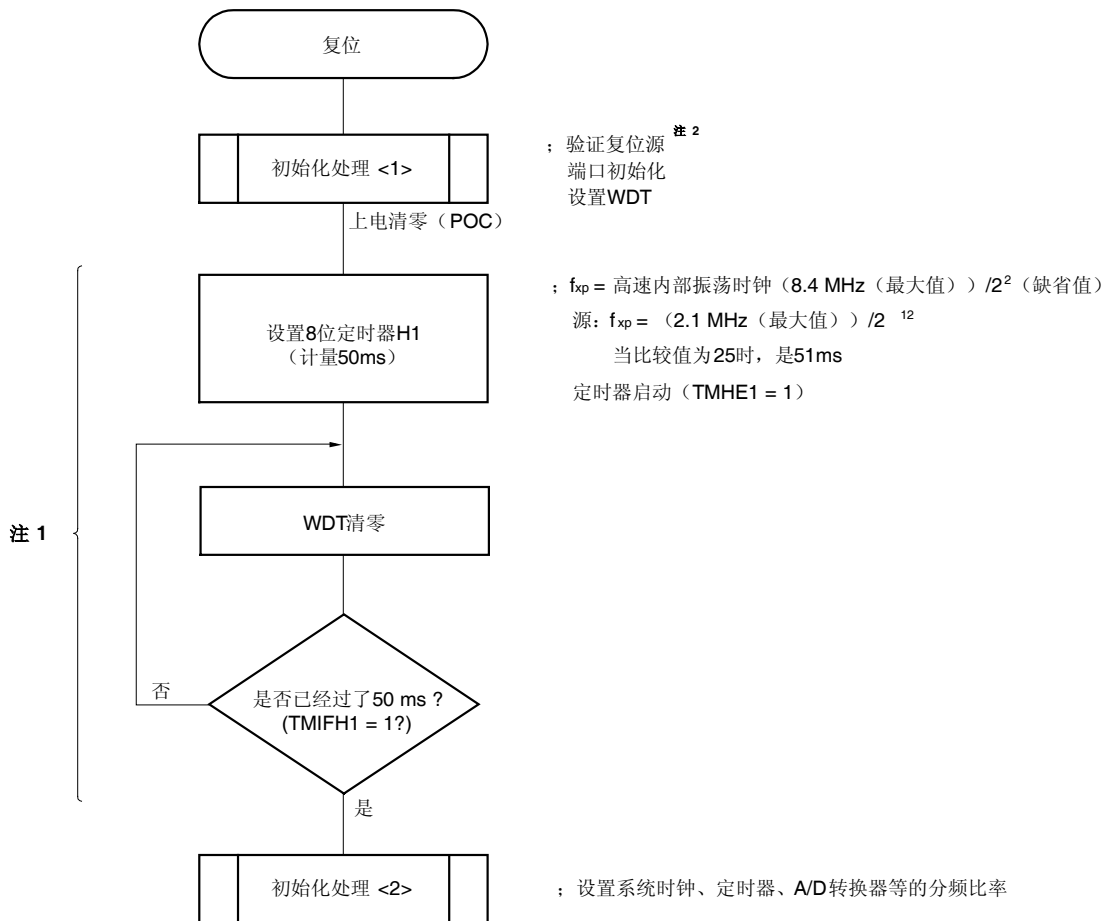
对于一个电源电压在某段时间内在 POC 检测电压 (V_{POC}) 附近波动的系统，电源电压在一段时间内会产生波动，系统可能会反复进行复位和释放复位。在这种情况下，可采用下列方法任意设置从复位释放到微控制器的启动所经历的时间。

<方法>

在释放复位信号后，通过软件计数器（使用一个定时器）等待系统电源电压的波动期，然后对端口初始化。

图 13-3. 复位释放后软件处理过程示例（1/2）

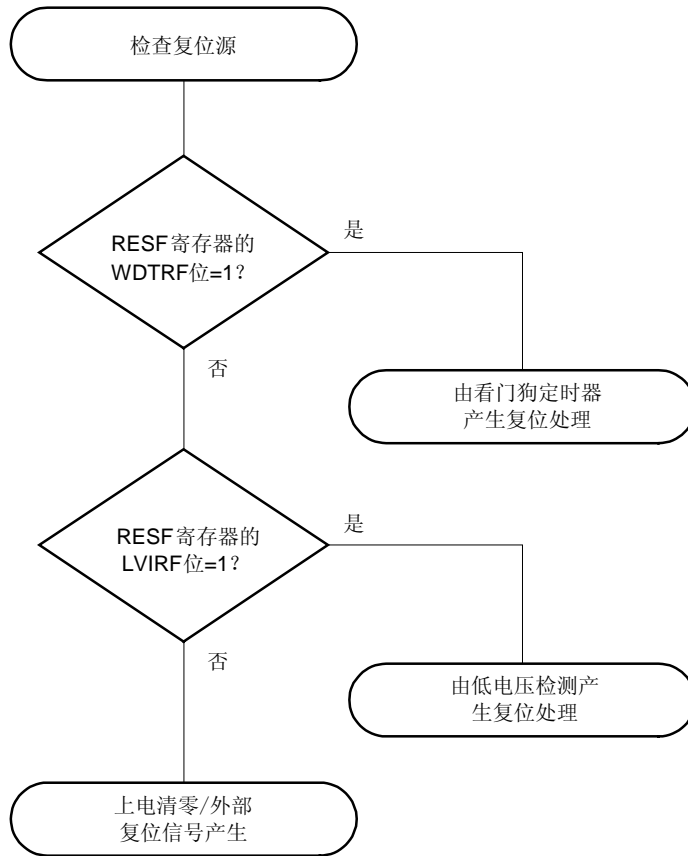
- 如果在电源电压接近 POC 检测电压时，其波动时间小于等于 50ms



- 注
1. 如果在此期间再次产生复位，初始化操作 <2> 不能开始。
 2. 流程图显示在下页。

图 13-3. 复位释放后软件处理过程示例 (2/2)

- 检查复位源



14.1 低电压检测器的功能

低电压检测器(LVI)有以下功能。

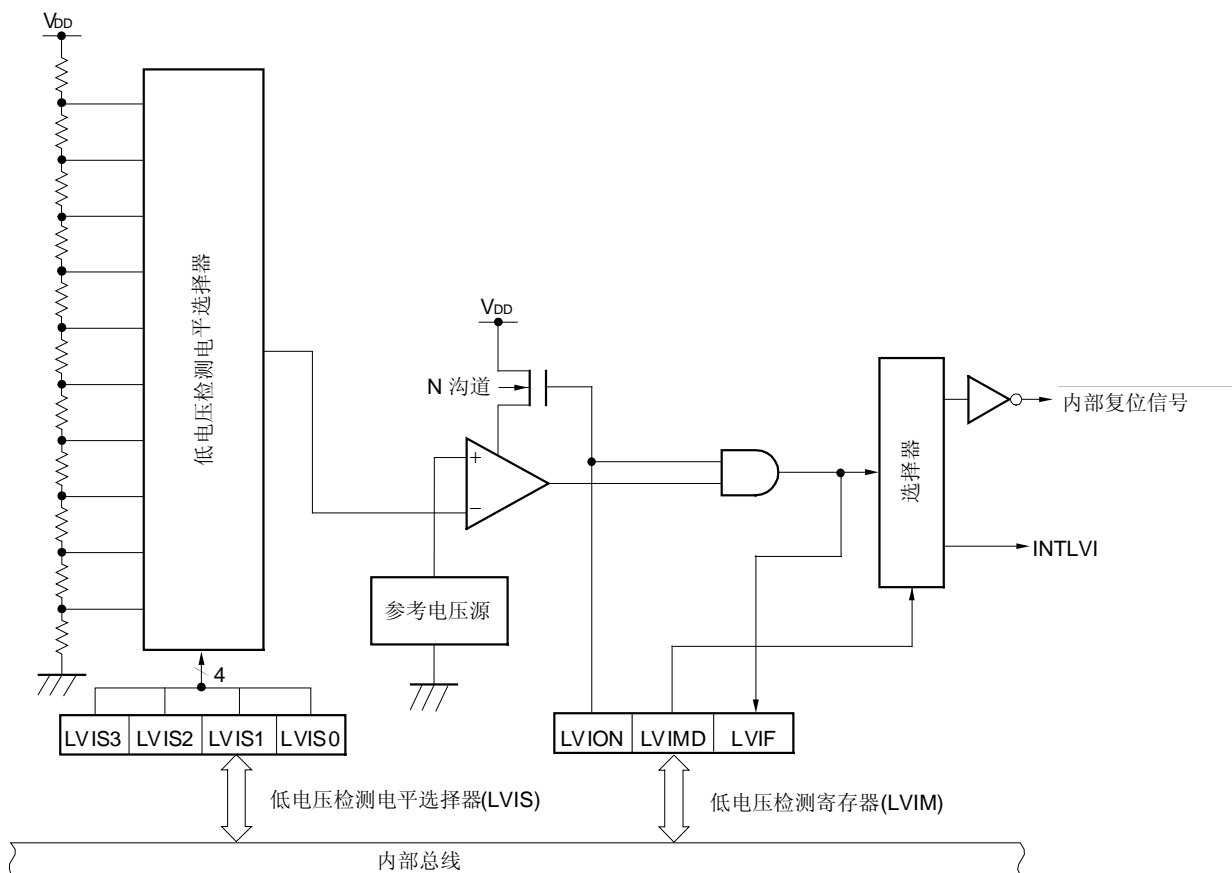
- 比较电源电压 (V_{DD})和检测电压 (V_{LVI}), 当 $V_{DD} < V_{LVI}$ 时, 产生内部中断信号或内部复位信号
- 可由软件改变电源电压的检测电平 (十个电平)
- 可由软件选择中断或复位功能
- 在 STOP 模式下可操作

在使用低电压检测器进行复位时, 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。需要了解 RESF 的详细内容, 可参见 第十二章 复位功能。

14.2 低电压检测器的配置

低电压检测器的框图如下图 14-1 所示。

图 14-1. 低电压检测器的框图



14.3 控制低电压检测器的寄存器

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和操作模式。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位输入将 LVIM 清零^{注1}。

图 14-2. 低电压检测寄存器(LVIM)的格式

地址:FF50H 复位后值: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}	使能低电压检测操作
0	禁止操作
1	使能操作

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}) \geq 检测电压(V_{LVI}), 或当禁止操作时
1	电源电压 (V_{DD}) < 检测电压 (V_{LVI})

- 注
1. 对于由 LVI 产生的一个复位, LVIM 的值未被初始化。
 2. 第 0 位只读。
 3. 当 LVION = 1 时, LVI 电路中的比较器开始运行。从设置 LVION = 1 到 LVIF = 1, 至少经过 0.2 ms。
 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. 可在执行完以下任一过程后终止 LVI。

- 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. 第 2~6 位必须清零。

(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
可由 8 位存储器操作指令设置该寄存器
复位输入将 LVIS 清零^注。

图 14-3. 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	$V_{LV10} (4.3 V \pm 0.2 V)$
0	0	0	1	$V_{LV11} (4.1 V \pm 0.2 V)$
0	0	1	0	$V_{LV12} (3.9 V \pm 0.2 V)$
0	0	1	1	$V_{LV13} (3.7 V \pm 0.2 V)$
0	1	0	0	$V_{LV14} (3.5 V \pm 0.2 V)$
0	1	0	1	$V_{LV15} (3.3 V \pm 0.15 V)$
0	1	1	0	$V_{LV16} (3.1 V \pm 0.15 V)$
0	1	1	1	$V_{LV17} (2.85 V \pm 0.15 V)$
1	0	0	0	$V_{LV18} (2.6 V \pm 0.1 V)$
1	0	0	1	$V_{LV19} (2.35 V \pm 0.1 V)$
其它设置				禁止设置

注 对于由 LVI 产生的一个复位, LVIM 的值未被初始化。

注意事项 1. 第 4~7 位必须清零。

2. 在 LVI 工作时如果有不同于上面的值写入, 写入的大部分时候值是不确定的, 因此确保在写入前停止 LVI(LVIM 寄存器的第七位(LVION) = 0)。

14.4 低电压检测器的操作

低电压检测器可用在以下两种模式中。

- 用作复位
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。
- 用作中断
比较电源电压 (V_{DD})和检测电压(V_{LVI}), 当 $V_{DD} < V_{LVI}$ 产生中断信号。

操作设置如下。

(1) 用作复位

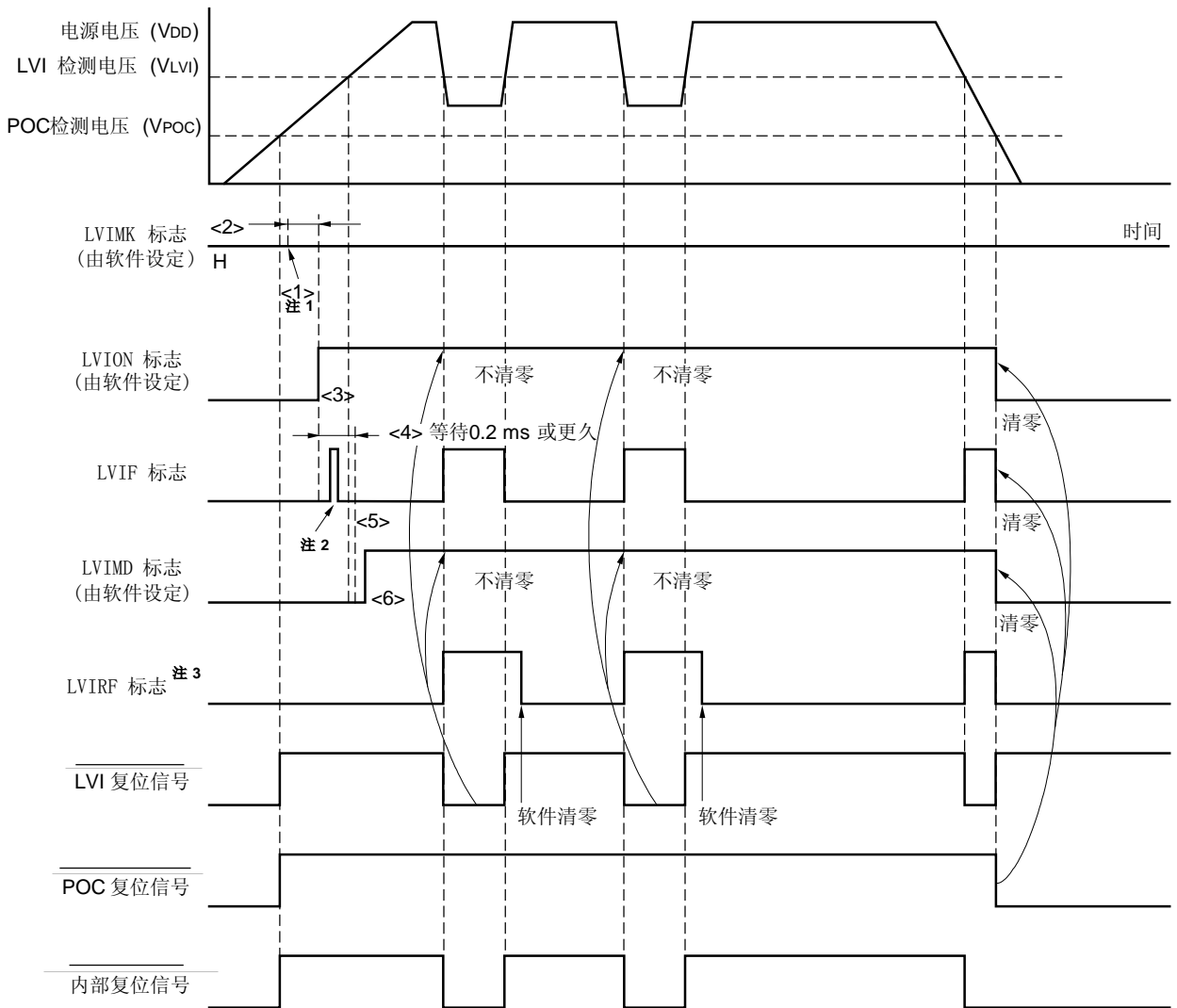
- 启动操作时
 - <1> 屏蔽 LVI 中断 ($LVIMK = 1$)。
 - <2> 使用低电压检测电平选择寄存器(LVIS)的第 3 ~ 0 位 ($LVIS3 \sim LVIS0$) 设置检测电压。
 - <3> 设置低电压检测寄存器(LVIM)的第 7 位($LVION$)=1 (允许 LVI 操作)。
 - <4> 使用软件等待至少 0.2 ms。
 - <5> 当根据 LVIM 的第 0 位($LVIF$)检查到“电源电压(V_{DD}) \geq 检测电压(V_{LVI})”时, 等待过程结束。
 - <6> 设置 LVIM 的第 1 位($LVIMD$)=1(当电源电压(V_{DD})<检测电压(V_{LVI})时产生内部复位信号)。

图 14-4 显示了由低电压检测器产生的内部复位信号的时序。该时序图中的数字与上面的<1>~<6>对应。

- 注意事项**
1. 必须执行过程<1>, 当 $LVIMK = 0$ 时, 在过程<3>之后可能立即会产生中断。
 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。

- 当终止操作时
必须执行以下过程之一:
 - 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 14-4. 低电压检测器内部复位信号产生时序



- 注
1. 由复位输入将 LVIMK 标志设置为“1”。
 2. 可以设置 LVIF 标志为(1)。
 3. LVIRF 是复位控制标志寄存器(RESF)的第 0 位。需要了解 RESF 的详细内容,可参见 第十二章 复位功能。

备注 图 14-4 中的<1>~<6> 与 14.4 (1) 用于复位中“当启动操作时”描述部分的<1>~<6>对应。

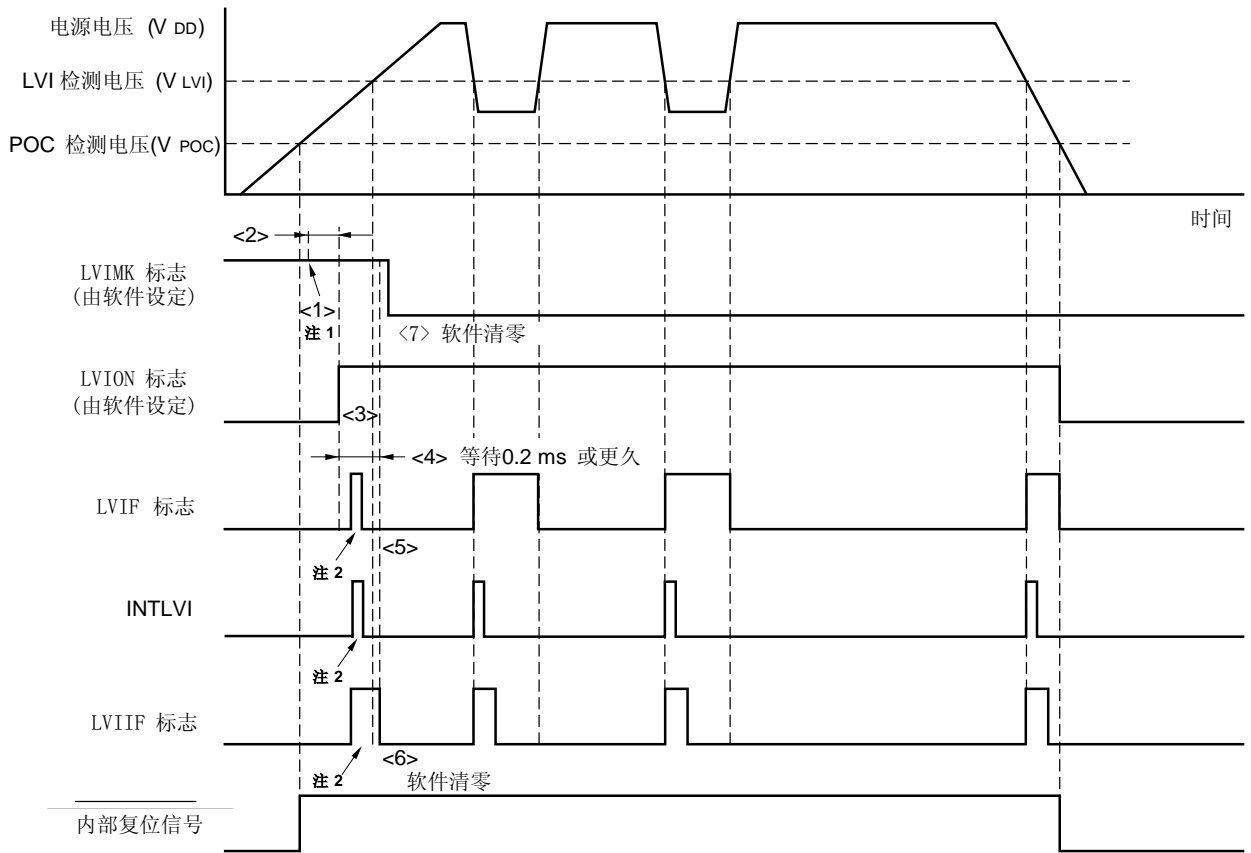
(2) 用于中断时

- 当启动操作时
 - <1> 屏蔽 LVI 中断(LVIMK = 1)。
 - <2> 使用低电压检测电平选择寄存器(LVIS)的第 3~0 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)=1(允许 LVI 操作)。
 - <4> 使用软件等待至少 0.2 ms 的时间。
 - <5> 当根据 LVIM 的第 0 位(LVIF)检查到“电源电压(V_{DD}) ≥ 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 14-5 显示了由低电压检测器产生的中断信号的时序，该时序图的符号与上面的<1>~<7>对应。

- 当终止操作时
必须执行以下过程之一。
 - 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 14-5. 低电压检测器中断信号产生时序



- 注
1. 由复位输入把 LVIMK 标志设置为“1”。
 2. LVIF 和 LVIIF 标志可以设置为“1”。

备注 图 14-5 中的<1>~<7>与 14.4 (2) 用于中断中“当启动操作时”描述部分的<1>~<7>对应。

14.5 低电压检测器的注意事项

在一个电源电压(VDD)在某段时间内在 LVI 检测电压附近波动的系统中，以下操作决定于如何使用低电压检测器。

<1> 用于复位

系统可能会反复进行复位和释放复位。

在这种情况下,可采用下面的方法(1)任意设置从复位释放到微控制器的启动所经历的时间。

<2> 用于中断

可能会频繁产生中断请求。可采用方法(2)。

在该系统中可采用如下方法。

<方法>

(1) 用于复位

在释放复位信号后,通过软件计数器(使用一个定时器)等待电源电压的波动期,然后对端口初始化(参见图 14-6)。

(2) 用于中断

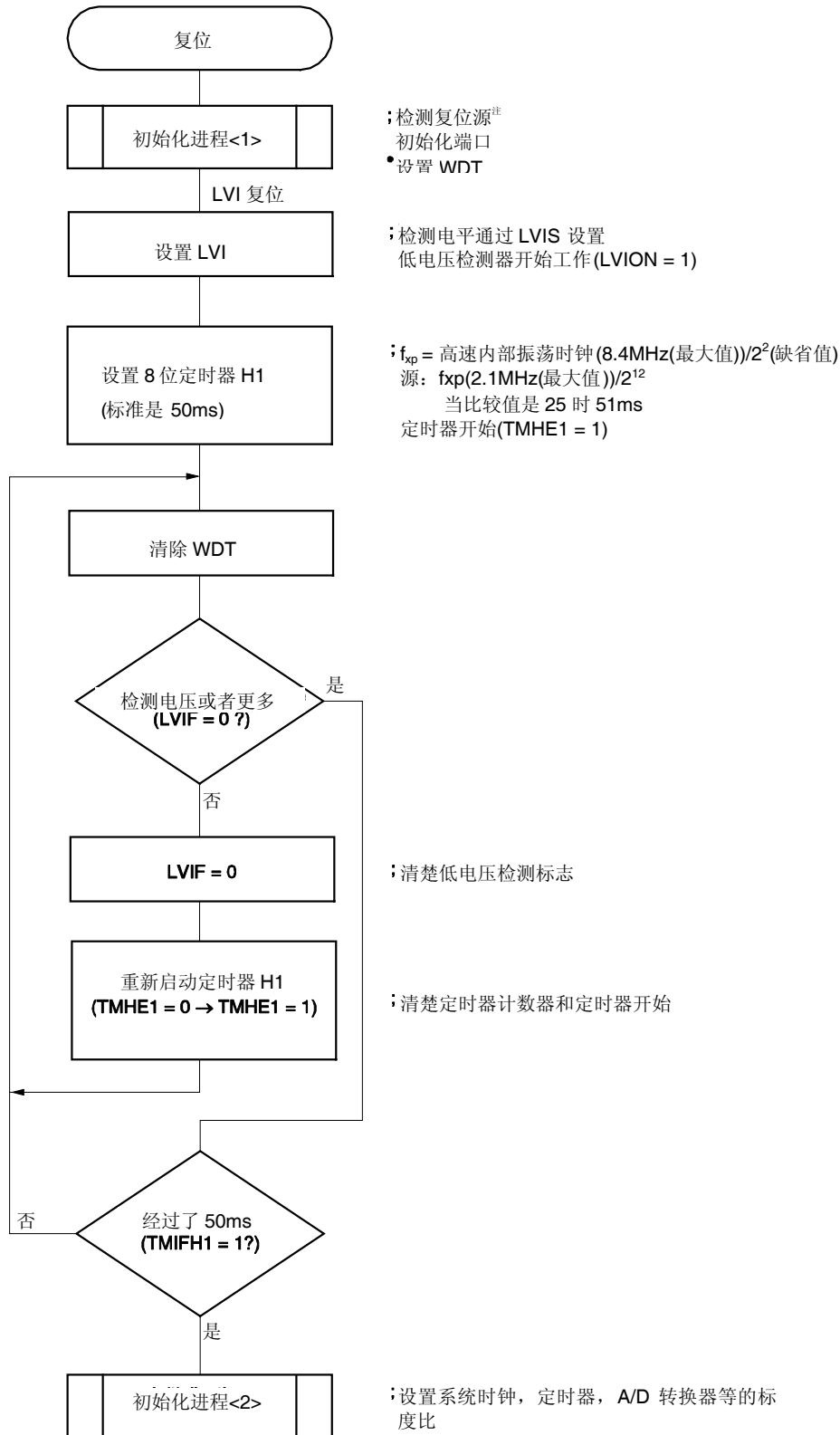
执行低电压检测过程^注,在 LVI 中断服务程序中,利用低电压检测寄存器(LVIM)的第 0 位(LVIF),检查“电源电压(VDD) ≥ 检测电压(VLvi)”,将中断请求标志寄存器(IF0)的第 1 位(LVIIF)清零并允许中断(EI)。

在系统中当电源电压接近 LVI 检测电压,电源电压的波动时间很长时,则要进行等待,利用 LVIF 标志检查“电源电压(VDD) ≥ 检测电压(VLvi)”的情况,然后清 LVIIF 为零,并使能中断(EI)。

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 14-6. 复位释放后软件处理过程示例 (1/2)

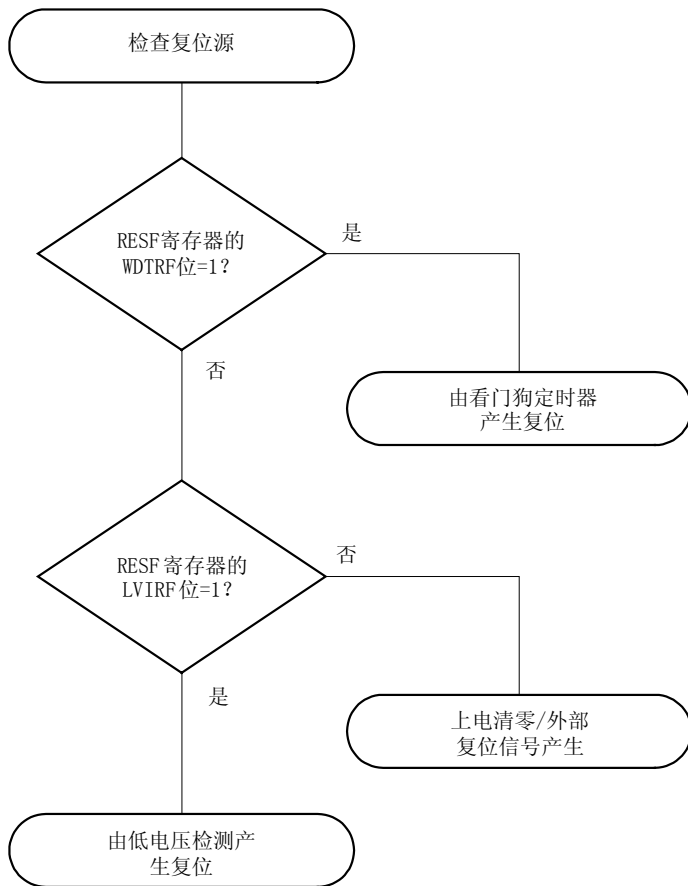
- 如果电源电压接近 LVI 检测电压,其波动时间小于等于 50ms。



注 流程图如下页所示。

图 14-6. 复位释放后软件处理过程示例 (2/2)

• 检查复位源



第十五章 选项字节

15.1 选项字节的功能

78K0S/KU1+有个名为选项字节的区域，在 Flash 存储器的地址 0080H 处。当电源商店或者复位后开始，自动引用选项字节，执行设定的指定功能。使用该产品时，必须通过选项字节设置如下功能。

(1) 系统时钟源选择

- 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

- 不能被停止
- 可被软件停止

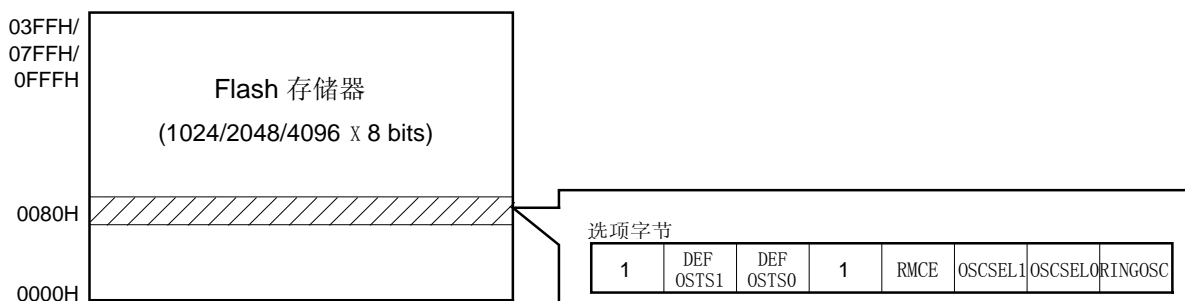
(3) $\overline{\text{RESET}}$ 引脚的控制

- 用作 $\overline{\text{RESET}}$ 引脚
- $\overline{\text{RESET}}$ 引脚可被用作输入引脚(P34)(参考 15.3 当 $\overline{\text{RESET}}$ 引脚用作仅输入端口引脚(P34)的注意事项)。

(4) 上电时或复位释放后振荡稳定时间

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

图 15-1. 选项字节的配置



15.2 选项字节的格式

选项字节的格式如下所示。

图 15-2. 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 仅当选晶体/陶瓷振荡器时钟作为系统时钟源时,这项设定才有效。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时,无需等待时间。

RMCE	RESET 引脚的控制
1	$\overline{\text{RESET}}$ 引脚用作复位输入
0	$\overline{\text{RESET}}$ 引脚用作输入引脚 (P34)

注意事项 由于选项字节在复位释放后参考,如果在选项字节被参考之前一个低电平被输入到 $\overline{\text{RESET}}$ 引脚,复位状态不会释放。

同样,当设置 RMCE 为 0 时,连街到上拉电阻。

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

注意事项 因为 X1 和 X2 引脚也作为 P23/ANI3 和 P22/ANI2 引脚使用, X1 和 X2 引脚使用的条件不同,取决于系统时钟源的选择。

(1) 选择高速内部振荡器时钟

P23/ANI3 和 P22/ANI2 引脚能够用作 I/O 端口引脚或 A/D 转换器的模拟输入引脚。

(2) 选择晶体/陶瓷振荡时钟

因为 X1 和 X2 引脚作为时钟输入引脚,所以它们不能被用作 I/O 端口引脚或 A/D 转换器的模拟输入引脚。

(3) 选择外部时钟输入

因为 X1 引脚作为外部时钟输入引脚, P23/ANI3 不能被用作 I/O 端口引脚或 A/D 转换器的模拟输入引脚。

备注 x :不必考虑

图 15-2. 选项字节的格式 (2/2)

LIOCP	低速内部振荡时钟
1	不能停止 (即使 LSRSTOP 位被写为 1, 振荡也不会停止)
0	通过软件停止 (当 LSRSTOP 位被写为 1 时, 振荡停止)

- 注意事项**
1. 如果选择低速内部振荡时钟振荡不能被停止, 看门狗定时器(WDT)的计数器时钟就会固定为低速内部振荡时钟。
 2. 如果选择低速内部振荡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 的计数器时钟在 HALT/STOP 模式下被停止,。同样地, 当选择除低速内部振荡时钟之外的任何一个时钟作为 WDT 的计数器时钟时, 时钟提供也被停止。
在低速内部振荡工作时 (LSRSTOP = 0),在 STOP 模式下计数时钟可提供给 8 位定时器 H1 使用。

- 备注**
1. (): $f_x = 10 \text{ MHz}$
 2. 需要了解振荡器的振荡稳定时间, 可参考所用的振荡器的特性。
 3. 下面是一个通过软件设置选项字节的例子。

```

OPB CSEG AT 0080H
DB 10010001B          ; 设置选项字节
                      ; 低速内部振荡不能停止
                      ; 系统时钟是晶体或者陶瓷振荡器
                      ; RESET引脚用于输入端口引脚 (P34).
                      ; 最小振荡稳定时间 ( $2^{10}/f_x$ )

```
 4. 可参考的选项字节的时序细节, 参见 第十二章 复位功能。

15.3 $\overline{\text{RESET}}$ 用于仅输入端口引脚 (P34)时的注意事项

不可以通过选项字节擦除/写入(通过使用专门的 flash 存储器编程器来板上编程)一个已经写入设置为 “ $\overline{\text{RESET}}$ 引脚用作仅输入引脚(P34)” 的设备。

目标系统上电之前, 要先给专门的 flash 存储器编程器上电。

如果预先给目标系统上电, 不能切换 flash 存储器编程器模式。

第十六章 FLASH 存储器

16.1 特征

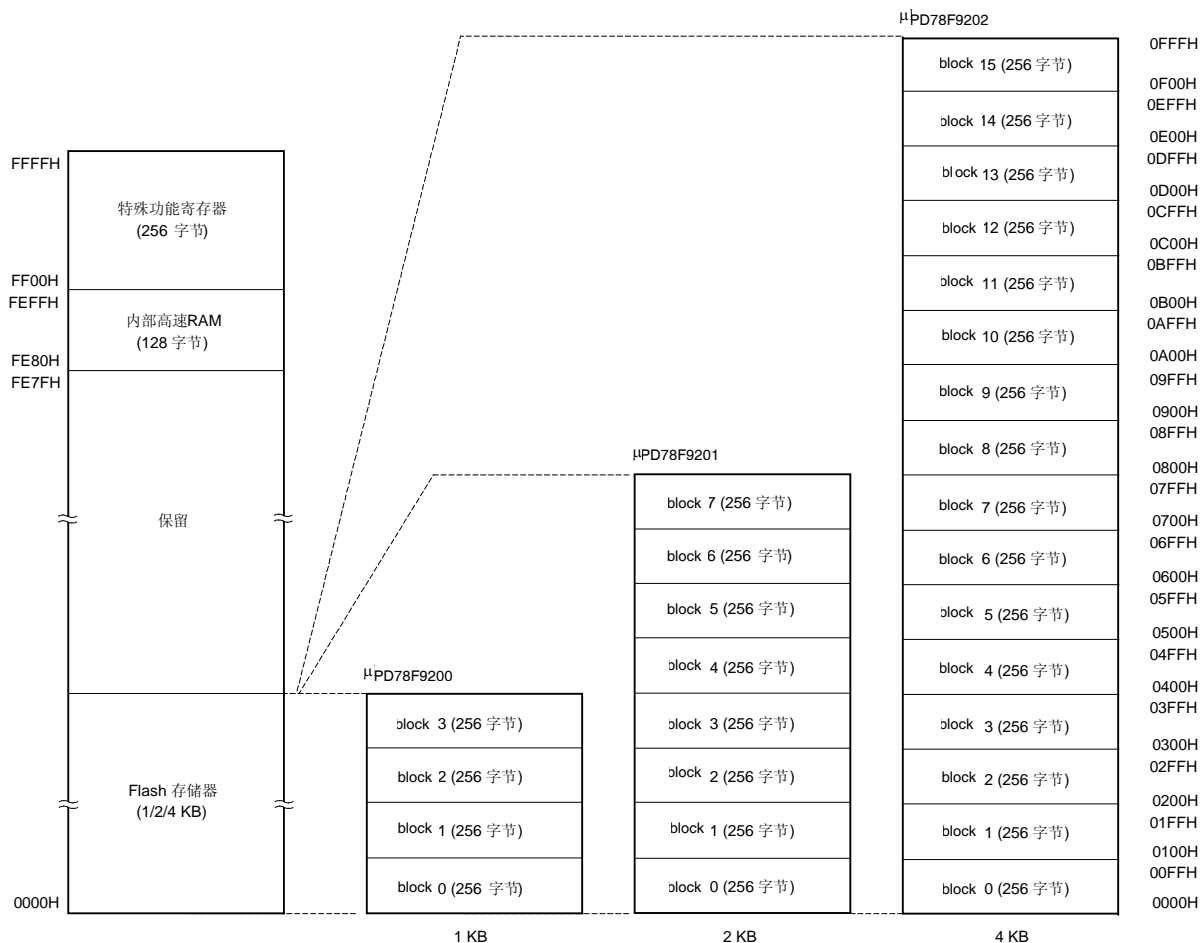
78K0S/KU1+内部 FLASH 存储器有如下特点。

- 不需要为擦除/写入准备单独的专用电源
- 容量: 1/2/4 KB
 - 擦除单元: 1 block (256 字节)
 - 写入单元: 1 block (在 on-board/off-board 编程时间), 1 字节(在自编程时间)
- 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(自编程)
- 支持通过安全功能在 on-board/off-board 编程时间写入 flash 存储器
- 支持通过保护字节在自编程时间在 block 单位中的安全功能

16.2 存储器配置

1/2/4 KB 的 FLASH 内部存储器区域被分为 8/16 个 block，并且能够以 block 为单位写入和擦除数据。所有的 block 也能够被一次性擦除。

图 16-1. Flash 存储器映射表



16.3 功能概要

78K0S/KU1+ 的内部 flash 存储器能够通过专用的 flash 编程器写入，不论 78K0S/KU1+ 是否已经连接到目标板(on-board/off-board 编程)。

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。

Flash 存储器写入控制功能参考表 16-1。

另外，支持安全功能，禁止未经授权的人写入内部 flash 存储器中的用户程序。

参考 16.7.3 安全设置 部分，有详细的安全功能说明。

表 16-1. 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(FA 系列), 写入 flash 存储器。	
自编程	可通过执行一段预先通过 on-board/off-board 编程方法写入到 flash 存储器的用户程序来写入 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论述请参考以下部分。

- 16.7 on-board 和 off-board 存储器编程。
- 16.8 通过自写入实现存储器编程。

16.4 使用 Flash 编程器的写入

下列两种专用 FLASH 编程器能够被用来向 78K0S/KU1+ 的内部 FLASH 存储器写入数据。

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

数据能够通过专用 FLASH 编程器以 on-board 或 off-board 方式写入 FLASH 存储器中。

(1) On-board 编程

78K0S/KU1+ 在安装到目标系统之后, flash 存储器的内容能够被写入。连接专用 flash 编程器和测试点的连接器必须安装到目标系统上。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 16-6 测试点的安装)

(2) Off-board 编程

在 78K0S/KU1+ 被安装到目标系统以前, 数据可以通过专用编程适配器(FA 系列)写入 FLASH 存储器中。

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

16.5 编程环境

编程环境要求如下图所示。

图 16-2. FLASH 存储器的编程环境(FlashPro4)

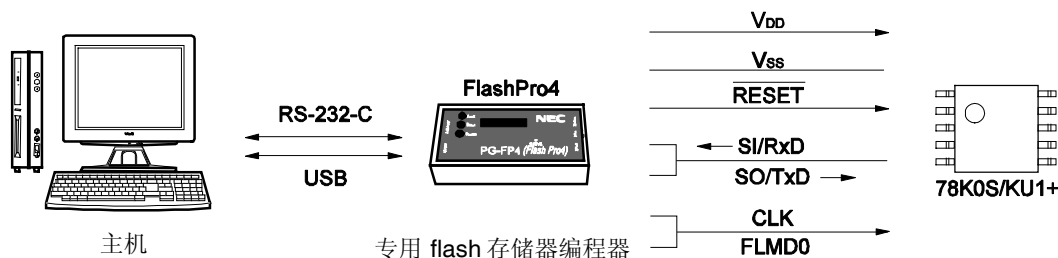
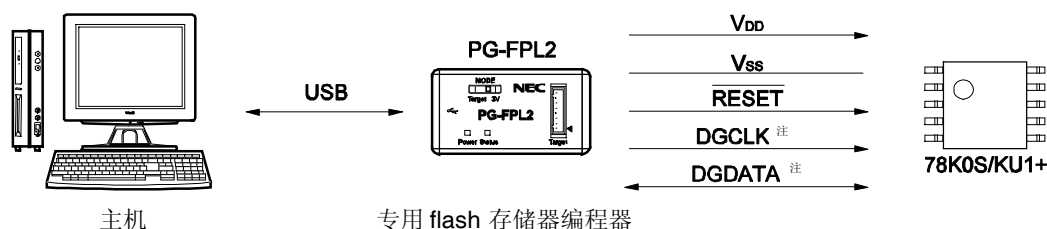


图 16-3. FLASH 存储器的编程环境(PG-FPL2)



注 在 DGDATA 是通讯数据的发送/接收信号时，DGCLK 是通讯时钟。

需要用一台主机来控制专用 flash 编程器。当使用 PG-FP4 或 FL-PR4 时，程序从主机下载到 flash 编程器之后，数据才能被写入。

UART 被用来在专用 flash 编程器与 78K0S/KU1+ 连接时操作写入和擦除。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)。

可以从网络上下载最新的编程器固件，GUI，和参数文件等开发工具，网址 (<http://www.necel.com/micro/ods/jpn/index.html>)。

表 16-2. 78K0S/KU1+和 FlashPro4 之间的连线

FlashPro4 连接引脚			78K0S/KU1+ 连接引脚	
引脚名称	I/O	引脚功能	引脚名称	引脚编号
CLK [Ⓢ]	输出	78K0S/KU1+的时钟	X1/P23/ANI3	5
FLMD0 [Ⓢ]	输出	On-board 模式信号		
SI/RxD [Ⓢ]	输入	接收信号	X2/P22/ANI2	6
SO/TxD [Ⓢ]	输出	发送/接收信号, on-board 模式信号		
/RESET	输出	复位信号	RESET/P34	7
VDD	-	VDD 电压发生/电压监视器	V _{DD}	4
GND	-	地	V _{SS}	3

注 在 78K0S/KU1+中, CLK 和 FLMD0 信号是连到 X1 引脚的, SI/RxD 和 SO/TxD 信号是连到 X2 引脚的;因此, 这些信号需要直接相连起来。

图 16-4. 与 FlashPro4 的连线图

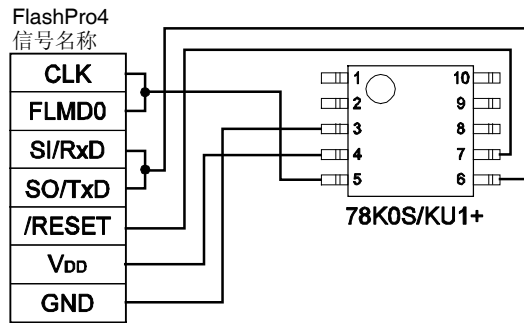
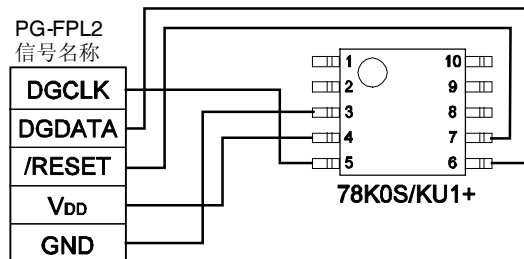


表 16-3. 78K0S/KU1+和 PG-FPL2 之间的连线

PG-FPL2 连接引脚			78K0S/KU1+连接引脚	
引脚名称	I/O	引脚功能	引脚名称	引脚编号
DGCLK	输出	78K0S/KU1+的时钟	X1/P21/ANI3	5
DGDATA	I/O	传输/接收信号, on-board 模式信号	X2/P22/ANI2	6
/RESET	输出	复位信号	RESET/P34	7
V _{DD}	I/O	VDD 电压发生	V _{DD}	4
GND	-	地	V _{SS}	3

图 16-5. 与 PG-FPL2 的连线图



16.6 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 存储器的引脚与复位后的状态相同。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。

16.6.1 X1 和 X2 引脚

X1 和 X2 引脚作为 FLASH 存储器编程的串行接口。因此，如果 X1 和 X2 引脚连接一外部设备，将会产生信号冲突。为了阻止信号发生冲突，应使引脚 X1 和 X2 与外部设备分开连接。

同样的，当一个电容连接到 X1 和 X2 引脚，通讯改变时的波形，通讯可能不依靠电容的容量。确保在 flash 编程时隔离电容的连接。

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，如要在已安装振荡器的情况下使用 on-board 写入方式须采用下列方法(1)和(2)。

- (1) 在器件与振荡器之间加入最少的测试点，通过测试点连接 FLASH 编程器。两者连线尽可能要短 (参考图 16-6 和表 16-4)。
- (2) 可以使用专用 FLASH 编程器的 GUI 软件来设定数据写入的通讯时钟的振荡器频率。查找串/并行振荡器和振荡器使用的反振频率，设定振荡器频率，以便使振荡频率在 $\pm 10\%$ 的范围之外(参考图 16-7 和表 16-5)。

图 16-6. 安装测试点的示例

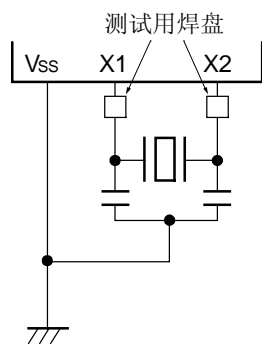


表 16-4. 使用的时钟和测试点的安装

使用的时钟		安装测试点
高速内部震荡器时钟		不需要
外部时钟		
晶体/陶瓷振荡器时钟	在振荡器安装之前	需要
	在振荡器安装之后	

图 16-7. PG-FP4 GUI 软件设置示例

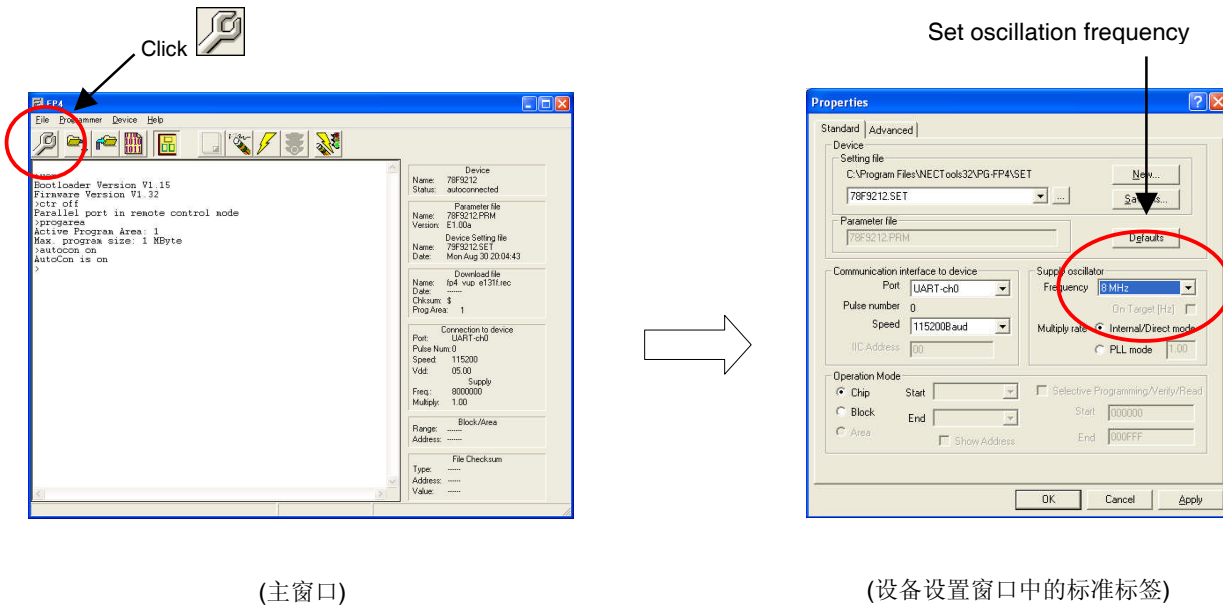


表 16-5. 振荡频率和 PG-FP4 GUI 软件设置值示例

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

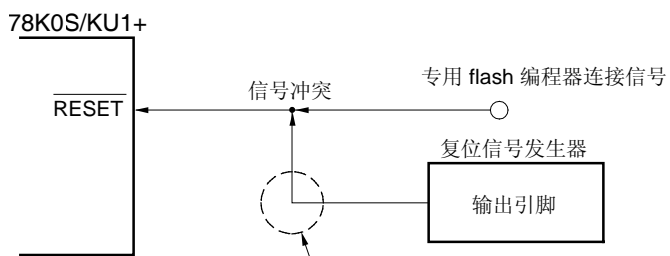
注意事项 以上值为推荐值。由于环境不同，这些值可能改变，所以使用时请先估计一个值。

16.6.2 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 $\overline{\text{RESET}}$ 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 存储器编程模式下从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 16-9. 信号冲突(RESET 引脚)



在 flash 存储器编程模式下，复位信号发生器的信号输出与专用 flash 编程器的信号输出相互冲突。因此，需要隔离复位信号发生器的信号。

16.6.3 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。

16.6.4 电源供应

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。

16.7 On-Board 和 Off-Board Flash 存储器编程

16.7.1 Flash 存储器编程模式

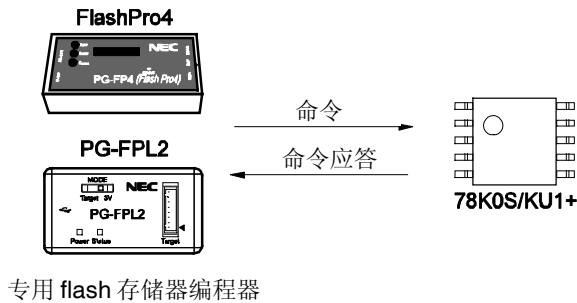
为了使用专用 FLASH 编程器写入 FLASH 存储器里的内容，需要将 78K0S/KU1+ 设定在 FLASH 存储器编程模式。当 78K0S/KU1+ 连接到 FLASH 编程器并且通讯命令发送到微控制器时，微控制器设定在 FLASH 存储器编程模式。

当以 on-board 方式写入 Flash 存储器时，可使用跳线改变操作模式。

16.7.2 通讯命令

通过命令使 78K0S/KU1+ 与专用 flash 编程器通讯。从 flash 编程器发送到 78K0S/KU1+ 的信号称为命令，从 78K0S/KU1+ 发送到 flash 编程器的信号称为应答命令。

图 16-9. 通讯命令



下表列出了 78K0S/KU1+ 的 Flash 存储器控制命令。这些命令都是由编程器发出，78K0S/KU1+ 根据相应命令完成处理。

表 16-6. Flash 存储器控制命令

分类	命令名称	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
	block 擦除命令	擦除指定 block 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
block 检测	block 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，block 擦除命令，和写入命令以阻止第三方操作

78K0S/KU1+ 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 16-7. 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

16.7.3 安全设置

通过使用安全设置命令能够禁止以下操作。

- 禁止一次性擦除(芯片擦除)

禁止对 flash 存储器中的整个 block 执行 block 擦除和一次性擦除(芯片擦除)命令。一旦一次性擦除(芯片擦除)命令被禁止,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)的安全设置设定之后,擦除指令不能执行。另外,即使执行写入命令,与已写入 flash 存储器不同的数据是不能被写入的,因为擦除命令已被禁止。

- 禁止 block 擦除

禁止对 flash 存储器中的某个指定 block 执行 block 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。

- 禁止写入

禁止对 flash 存储器中的整 block 执行写入和 block 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 安全设置当设置下次为编程模式时有效。

FLASH 存储器在默认设置下,一次性擦除(芯片擦除),block 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。

表 16-8 显示了当允许 78K0S/KU1+ 的安全功能后,擦除与写入命令的关系。

表 16-8. 当设置安全功能后命令之间的关系

安全 \ 命令	一次性擦除(芯片擦除)	Block 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 block 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 16-9 显示在每个编程模式下安全设置与操作的关系。

表 16-9. 每个编程模式下安全设置与操作的关系

安全设置	编程模式	On-Board/Off-Board 编程		自编程	
		安全设置	安全操作	安全设置	安全操作
一次性擦除(芯片擦除)	允许	允许	有效 ^{注1}	不允许	无效 ^{注2}
Block 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。

16.8 通过自写入实现 Flash 存储器编程

78K0S/KU1+具有自编程功能，可以通过用户程序来写入 FLASH 存储器，因此具有在线升级程序的功能。

注意事项 在执行自写入以前，自写入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，参考后面的 16.8.4 中的例子
 2. 若想用 78K0S/KU1+的内部 FLASH 当作外部 EEPROM 来存储数据时,请参考 **78K0S/Kx1+ EEPROM 仿真应用笔记 (U17379E)**。

16.8.1 自编程概述

如要执行自编程，需将工作模式由用户程序的正常操作(正常模式)切换到自编程模式。在自编程模式下，先设置特定寄存器，通过执行 HALT 指令执行 flash 存储器的擦写处理程序。当程序结束时，将自动释放 HALT 状态。

如要切换到自编程模式，需以特定顺序对特定寄存器进行操作，详情请参考 **16.8.4 从普通模式切换到自编程模式示例**。

备注 自编程序写入数据可参照 MOV 指令。

表 16-10. 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
普通模式	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 16-10 是自编程框图，图 16-11 是自编程状态转换图，表 16-11 列举了自编程控制命令。

图 16-11. 自编程状态转换图

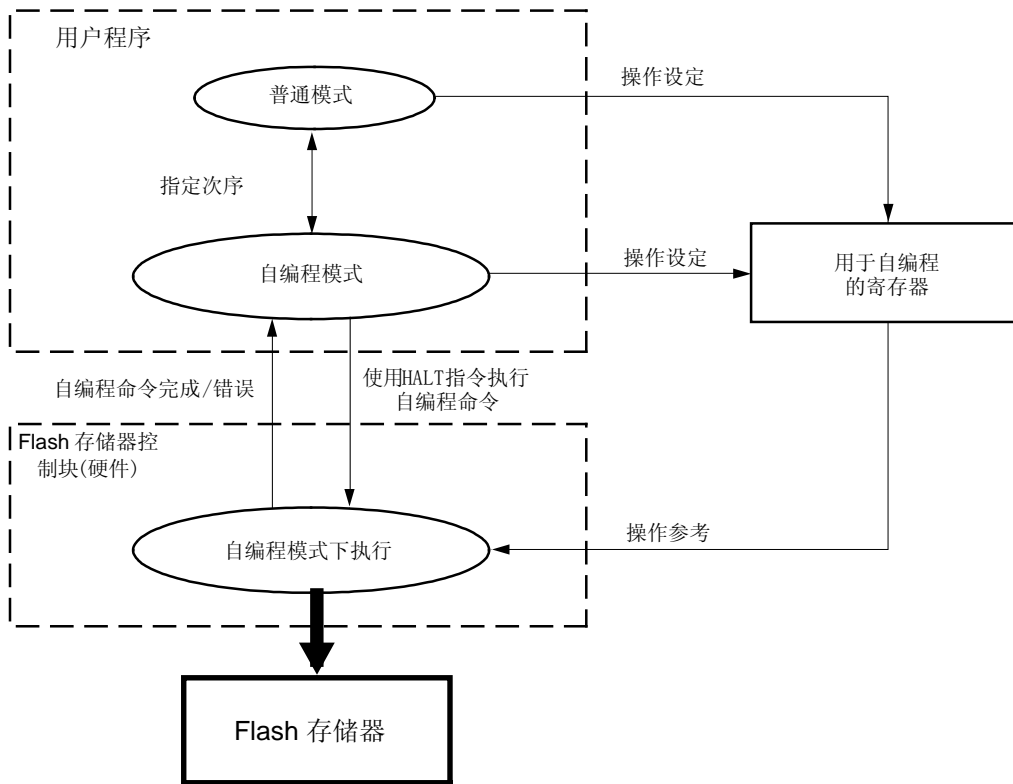


表 16-11. 自编程控制命令

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。用于检测数据是否写入整个 block。	对于 1 个 block 的内部校验 (内部校验命令执行一次): 6.8ms
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。用于检测数据是否写入到相同 block 中的多重地址。	对于 1 个字节的内部校验: 27 μ s
block 擦除	该指令用于擦除指定 block 的内容, 先指定 block 的序号, 然后执行此命令。	8.5 ms
空白 block 检验	该指令用于校验指定 block 是否已被擦除。先指定 block 的序号, 然后执行此命令。	480 μ s
字节写	该指令用于对 FLASH 存储器指定地址写入 1 字节数据。在确定写入地址和写入数据之后, 就可执行此命令。	150 μ s

备注 内部校验 1 指令可以通过相同 block 中的指定地址来执行, 但是内部校验 2 推荐写入相同 block 中的多重地址。

16.8.2 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 16
- 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，应在从普通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 block 擦除期间空白数据的值设为 0FFH。
- 在自编程时预先设置 CPU 时钟为 1MHz 或者更高。
- 执行一个设置自编程模式的指定顺序后，在执行 NOP 和 HALT 指令之后立即执行自编程。这时，HALT 指令 10 μ s (最大值) + 2 CPU 时钟 (fcpu) 自动释放。
- 如果选择振荡器的时钟或者外部始终作为系统时钟，执行一个设置自编程模式的指定顺序之后立即执行 NOP 和 HALT 指令，在 HALT 状态释放后等待 8 μ s，然后执行自编程。
- 使用 1 位操作指令来检测 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。
- 当执行自编程命令时，如果 FLASH 地址指针 H (FLAPH) 和 FLASH 地址指针 H 比较寄存器 (FLAPHC) 的第 4 位到第 7 位为 1，在执行自编程命令之前一定要对这些位清零。
- 在设置自编程模式和普通模式之前立即清除 FLCMD 寄存器的值为 00H。

16.8.3 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 (FLAPH 和 FLAPL)
- Flash 地址指针 H 比较寄存器和 flash 地址指针 L 比较寄存器 (FLAPHC 和 FLAPLC)
- Flash 写缓冲寄存器 (FLW)

78K0S/KU1+ 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)

这个寄存器用于在自编程模式下向 FLASH 存储器写入数据时设定工作模式，并可用于读取保护字节的值。

只有按特定顺序(参考 16.8.3 (2) FLASH 保护命令寄存器(PFCMD)) 数据才能写入 FLPMC，如此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。

该寄存器可通过一个 8 位的存储操作指令来设置。

复位后寄存器内容不确定。

图 16-12. Flash 编程模式控制寄存器(FLPMC)的格式

地址:	FFA2H	复位后:	不确定 ^{※1}	读/写 ^{※2}						
符号	7	6	5	4	3	2	1	0		
FLPMC	0	PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	0	FLSPM		

FLSPM	自编程模式期间操作模式的选择
0	普通模式 这时普通工作状态。执行HALT指令设置待机状态。
1	自编程模式 可以在正常模式下通过执行指定执行顺序来改变模式来执行自编程指令。 设置一个指令，一个地址，一个数据来写入，然后执行HALT指令来执行编程。

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. 注意在设定自编程模式的情况下，可参考 18.8.2 自编程功能的注意事项
 2. 在自编程时预先设置 CPU 时钟为 1MHz 或者更高。
 3. 执行一个设置自编程模式的指定顺序后，在执行 NOP 和 HALT 指令之后立即执行自编程。这时，HALT 指令 10 μ s (最大值) + 2 CPU 时钟 (fCPU)自动释放。
 4. 如果选择振荡器的时钟或者外部始终作为系统时钟，执行一个设置自编程模式的指定顺序之后立即执行 NOP 和 HALT 指令，在 HALT 状态释放后等待 8 μ s，然后执行自编程。
 5. 在设置自编程模式和普通模式之前立即清除 FLCMD 寄存器的值为 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，这样就不会使应用系统由于疏忽导致停止。

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 在自编程模式时不能执行中断服务。在执行设置自编程模式的指定顺序之前和执行改变模式到正常模式的指定顺序之后禁止中断服务(可通过将 MK0 和 MK1 设为 0FFH，还有执行 DI 指令来实现)。

这样重复写入寄存器，可使寄存器不能被非法写入。

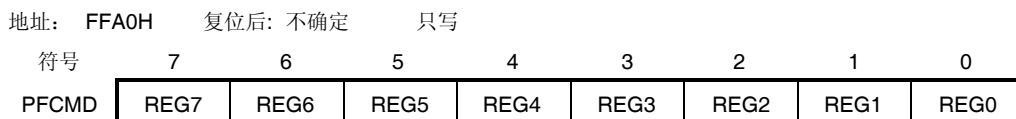
可以通过 FLASH 状态寄存器(PFS)的第 0 位(FPRERR)来验证是否出现错误的存储操作。使用 1 位操作指令来检测 FPRERR。

每当 FLPMC 的值发生改变的时候都必须向 PFCMD 写入值 A5H。

PFCMD 的值可通过一个 8 位的存储器操作指令来设置。

复位后寄存器 PFCMD 状态值不确定。

图 16-15. Flash 保护命令寄存器 (PFCMD)的格式



(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)位反应出来, (VCERR)和 (WEPRERR)可以通过写 0 来清 0。

要验证操作是否执行正确, PFS 寄存器的所有标志位必须被预先清零。

PFS 可以使用 1 位或 8 位存储器操作指令设置。

复位后 PFS 被置为 00H。

注意事项 使用 1 位操作指令来检测 FPRERR。

图 16-14. 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 标志的操作条件

<设置条件>

- 刚执行的对外围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围寄存器而不是 FLPMC。
- 在<2>之后, 如果第一条存储指令操作的是外围寄存器而不是 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围寄存器而不是 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 以上角括号中的数字对应于 (2) Flash 保护命令寄存器 (PFCMD) 中的数字。

<复位条件>

- 如果 FPRERR 标志写 0
- 如果复位信号输入

2. VCERR 标志的操作条件

<设置条件>

- 擦除校验错误
- 内部写入校验错误

如果 VCERR 被设为 1, 则说明 FLASH 没有被正确的擦除或写入。在指定的程序里要再次对 FLASH 进行擦除或写入。

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 的擦除、写入、校验操作。

该寄存器可通过 1 位或 8 位的操作指令来设置。

复位后寄存器值置为 00H。

图 16-15. 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	命令名称	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。用于检测数据是否写入到整个block。如果发生错误，设置flash状态寄存器(PFS)的位1(VCERR)和位2(WEPREERR)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。用于检测数据是否写入到相同block中的多重地址。如果发生错误，设置flash状态寄存器(PFS)的位1(VCERR)和位2(WEPREERR)为1。
0	1	1	Block擦出	此命令用于对指定block的擦除。它可以用在on-board和自编程模式下。
1	0	0	空白block检测	此命令用于检测指定block是否已经被擦除。
1	0	1	字节写	此命令用于将一个字节数据写入flash存储器的指定地址。指令执行前要指定写入地址和写入数据。 如果1个位“1”要写入到一个没有被删除的位时(为0的位)，则FLASH状态寄存器(PFS)的第2位被置为1
其他 ^注			禁止设置	

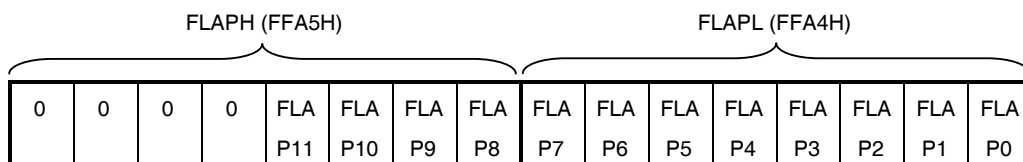
注 如果不是以上列举的值并且工作模式设置为自编程模式，则自编程模式被立刻取消，并且设置 flash 状态寄存器(PFS)的位 1(VCERR)和位 2(WEPREERR)为 1。

(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，当编程命令不执行时，它们递增直到与 FLAPHC 和 FLAPLC 的值相等。因此，当程序命令执行时，PLAPHC 和 FLAPLC 的值需要重新设置。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。复位后寄存器值不确定。

图 16-16. 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 4 位-第 7 位和 FLAPHC 清零。当执行自编程命令时这些位的值为 1。

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

在自编程模式下 FLASH 存储器校验时，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。

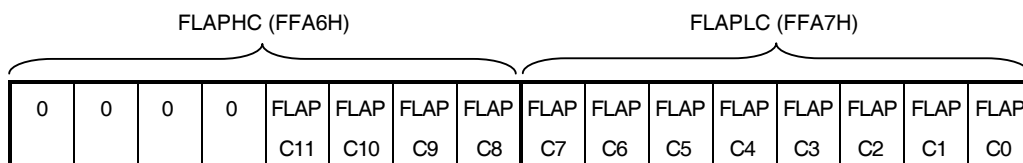
将 FLAPH 的值赋给 FLAPHC。将 flash 存储器校验范围的最末端地址赋给 FLAPLC。

FLAPHC 和 FLAPLC 使用 1 位或 8 位存储器操作指令来设定。

复位后 FLW 寄存器值为 00H。

图 16-19. 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. 在自编程命令执行以前，确保 FLASH 地址指针 H(FLAPH)的第 4 位至第 7 位和 FLASH 地址指针 H 比较寄存器(FLAPHC)清零。
 2. 当进行 block 擦除、校验、空白检测时，将 block 的序号(与 FLAPH 的值相同)赋值给 FLAPHC。
 3. 当 block 擦除完成时 FLAPLC 要清零，当 block 空白检测完成后 FLAPLC 要设为 FFH。

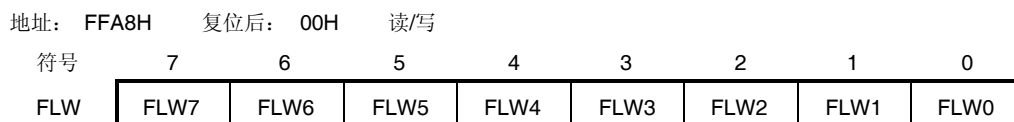
(7) Flash 写缓冲寄存器 (FLW)

该寄存器用于存储要写入 flash 存储器的数据。

该寄存器值可用 1 位或 8 位存储器操作指令设置。

复位后 FLW 寄存器值置为 00H。

图 16-20. FLASH 写缓冲寄存器(FLW)的格式



(8) 保护字节

保护字节用于指定禁止写入或擦除区域。该指定的保护区只在自编程模式下有效。因为受保护区域在自编程模式下是不可用的，所以保护区内的数据是受保护的。

图 16-21. 保护字节的格式 (1/2)



• PD78F9200

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	状态
0	1	1	1	0	Block 3到0是受保护的
0	1	1	1	1	Block 1和0是受保护的 Block 2和3可被写入或擦除
1	1	1	1	1	所有的Block都可被写入或擦除
其它					禁止设置

• PD78F9201

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	状态
0	1	1	0	0	Block 7到0是受保护的
0	1	1	0	1	Block 5到0是受保护的 Block 6到7可被写入或擦除
0	1	1	1	0	Block 3到0是受保护的 Block 4到7可被写入或擦除
0	1	1	1	1	Block 1到0是受保护的 Block 2到7可被写入或擦除
1	1	1	1	1	所有的blockBlock都可被写入或擦除
其它					禁止设置

图 16-21. 保护字节的格式(2/2)

PD78F9202					状态
PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	
0	1	0	0	0	Block 15到0是受保护的
0	1	0	0	1	Block 13到0是受保护的 Block 14到15可被写入或擦除
0	1	0	1	0	Block 11到0是受保护的 Block 12到15可被写入或擦除
0	1	0	1	1	Block 9到0是受保护的 Block 10到15可被写入或擦除
0	1	1	0	0	Block 7到0是受保护的 Block 8到15可被写入或擦除
0	1	1	0	1	Block 5到0是受保护的 Block 6到15可被写入或擦除
0	1	1	1	0	Block 3到0是受保护的 Block 4到15可被写入或擦除
0	1	1	1	1	Block 1到0是受保护的 Block 2到15可被写入或擦除
1	1	1	1	1	所有的Block都可被写入或擦除
其它					禁止设置

16.8.4 从普通模式切换至自编程模式的示例

在执行自编程模式前必须由普通模式切换至自编程模式。

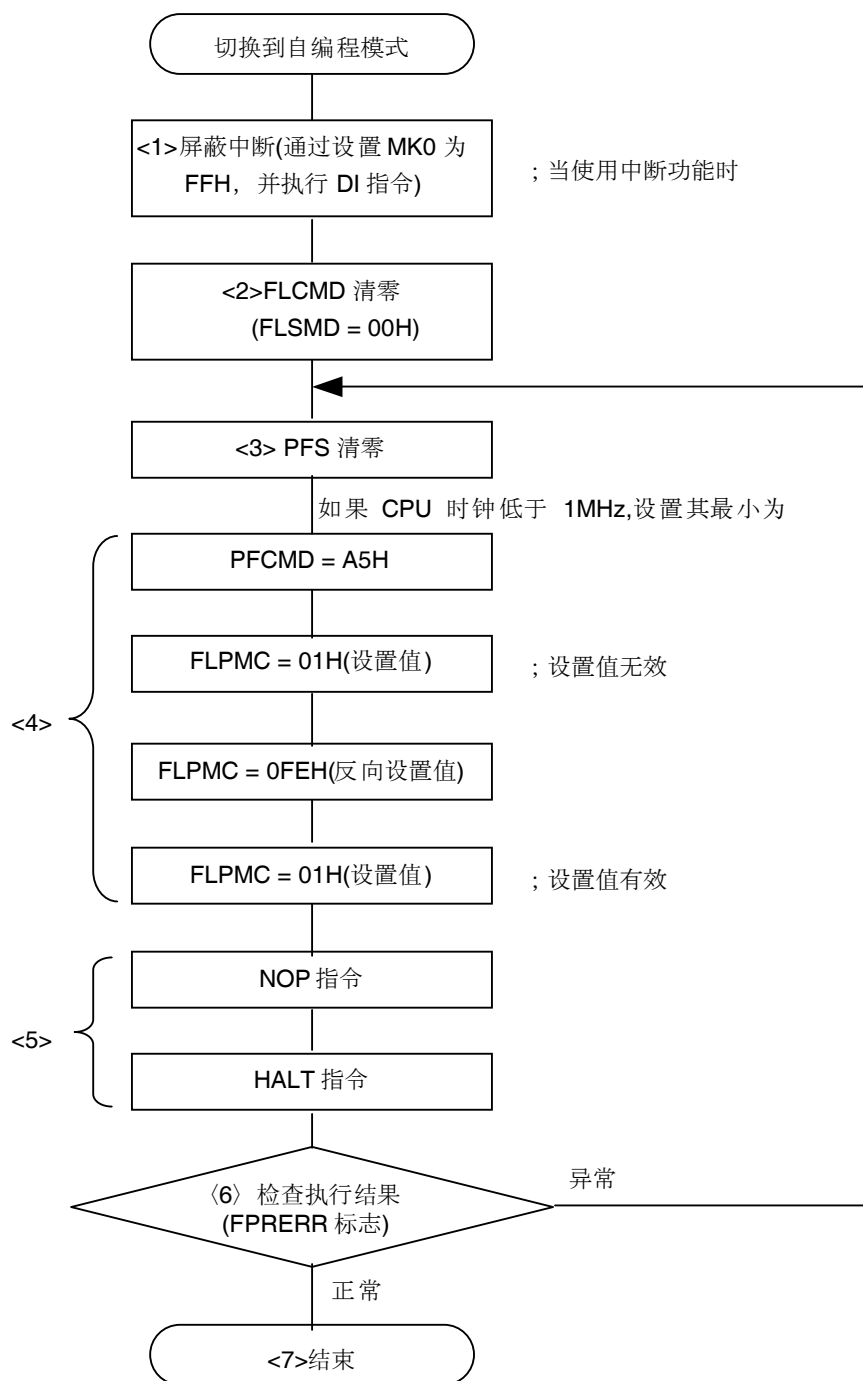
切换至自编程模式的操作过程示例如下：

- <1> 如果使用了中断功能则必须首先屏蔽中断 (通过设置的中断屏蔽标记寄存器(MK0)为 FFH 并执行 DI 指令)。
- <2> 清除 FLCMD (FLCMD = 00H)。
- <3> 清除 flash 状态寄存器(PFS)。
- <4> 按指定顺序设置自编程模式。
 - 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) 检验指定顺序执行的结果是否正确。
错误 → <3> 正常 → <7>
- <7> 模式转换结束。

注 如果 CPU 时钟低于 1MHz，设置它为 1MHz 或者更高。

注意事项 确保执行上面在数据不能擦除和写入地址使用用户编程的工作的顺序。

图 16-20. 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和写入操作发生。

备注 图 16-20 中的<1>到<7> 对应于 16.8.4 中的<1>到<7>(前页)。

以下是切换到自编程模式的示例程序。

```
;-----  
;开始  
;-----  
        MOV     MK0,#11111111B    ;屏蔽全部中断  
        MOV     FLCMD,#00H        ;清除 FLCMD 寄存器  
  
        DI  
  
ModeOnLoop:                ;设定配置来让 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(反相设置值)  
        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.0,$ModeOnLoop    ;验证完成写入指定寄存器  
                                   ;出现错误时重复相同的处理  
  
;-----  
;结束  
;-----
```


16.8.5 从自编程模式切换到普通模式的示例

自编程完成后，操作模式一定要从自编程返回到正常模式。

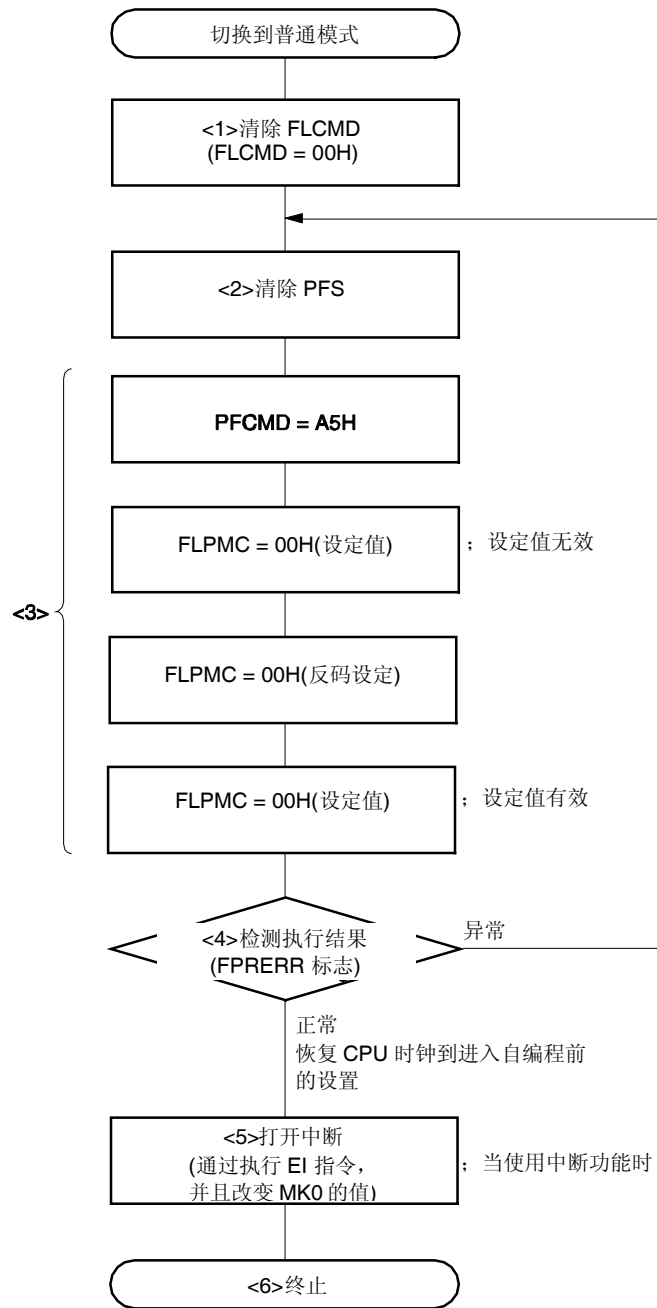
以下是切换到普通模式的过程。

- <1> 清除 FLCMD (FLCMD = 00H).
- <2> 清除状态寄存器(PFS)
- <3> 按指定的顺序设置普通模式
 - 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) 检验指定顺序执行的结果是否正确^注。
错误 → <2> 正常 → <5>
- <5> 开放中断处理服务(执行 EI 指令和改变 MK0 的值)以恢复到原来的状态
- <6> 模式转换结束

注 在自编程之前，在正常指定顺序完成之后，恢复 CPU 时钟到他的设定。

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和写入操作发生。

图 16-21. 切换到普通模式的示例



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和写入操作发生。

备注 图 16-21 中的<1>到<6> 对应于 16.8.5 中的<1>到<6>(前页)。

以下是切换到普通模式的示例程序。

```

;-----
;开始
;-----

        MOV     FLCMD,#00H      ;清除 FLCMD 寄存器

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 (反相设置值)
        MOV     FLPMC,#00H     ;通过 FLPMC 寄存器控制设置普通模式 (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
                                           ;指定序列正常结束后, 自编程开始前恢复 CPU 时钟的设置
                                           ;
                                           ;
        MOV     MK0,#INT_MK0   ;恢复中断屏蔽标志

        EI

;-----
;结束
;-----

```

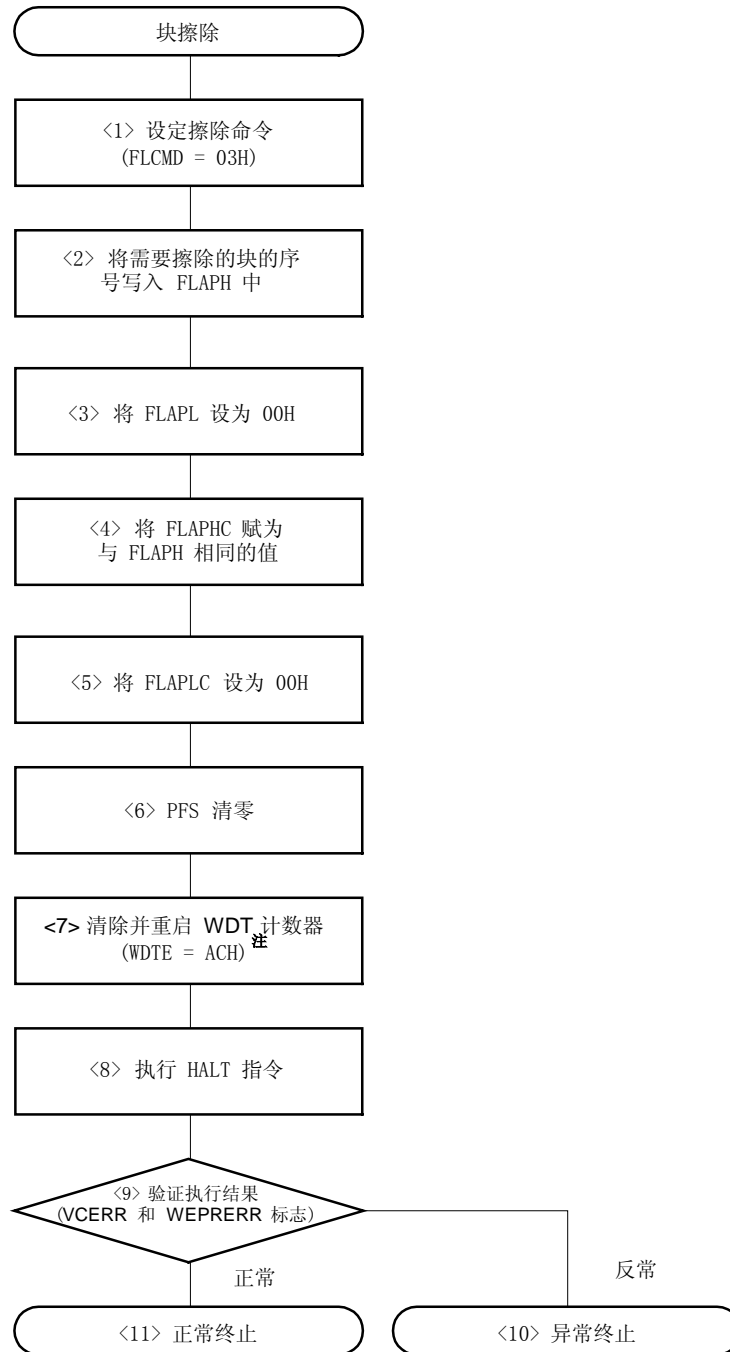
16.8.6 自编程模式下 block 擦除操作示例

自编程模式下 block 擦除操作过程如下：

- <1> 设置编程命令寄存器(FLCMD)值为 03H(block 擦除)。
- <2> 将需要擦除的 block 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。
- <8> 执行 HALT 指令然后开始自编程序(如果自编程序已开始执行,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(WEPERR)检查自编程序是否产生错误。
错误 → <10>
正常 → <11>
- <10> block 擦除处理非正常结束。
- <11> block 擦除处理正常的结束。

注 当看门狗定时器不使用时无需对其进行设置。

图 16-22. 自编程模式下 block 擦除操作流程示例



注 当看门狗定时器不使用时无需对其进行设置。

备注 图 16-22 中的<1>到<11>对应于 16.8.6 中的<1>到<11>(前页)。

以下是自编程模式下 block 擦除程序示例。

```
;-----  
;开始  
;-----  
  
FlashBlockErase:  
    MOV    FLCMD,#03H    ; 设置 flash 控制命令 (block 擦除)  
    MOV    FLAPH,#07H    ; 设置要擦除的 block 号 (这里指定 block 7)  
    MOV    FLAPL,#00H    ; FLAPL 恒为“00H”  
    MOV    FLAPHC,#07H   ; 设置擦除 block 的比较数 (与 FLAPH 值相同)  
    MOV    FLAPLC,#00H   ; FLAPLC 恒为“00H”  
  
    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  
;-----
```

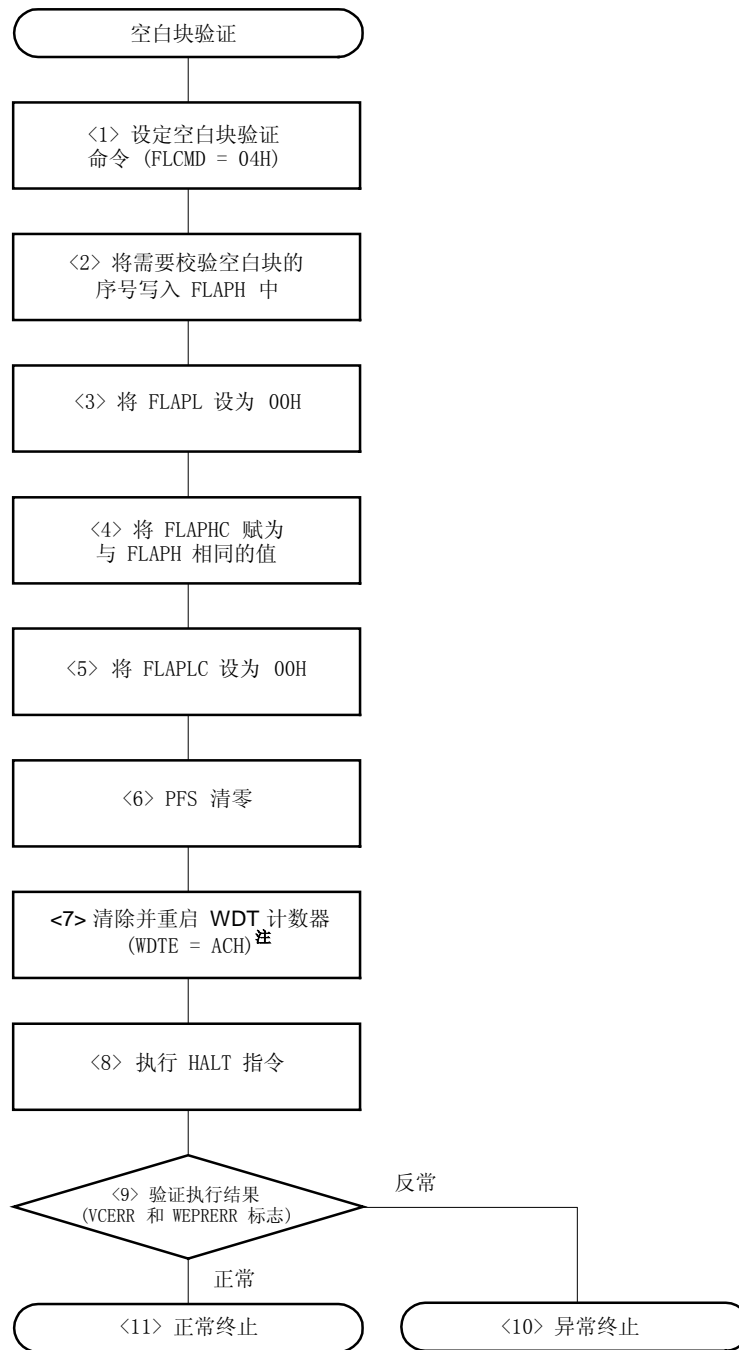
16.8.7 自编程模式下空白 block 的校验操作示例

自编程模式下空白 block 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 block 校验)。
- <2> 将需要执行空白 block 校验的 block 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 的值写回到 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序(如果自编程序已开始执行,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)检查自编程序是否产生错误。
 - 错误 → <10>
 - 正常 → <11>
- <10> 空白 block 校验非正常结束。
- <11> 空白 block 校验正常结束。

注 当看门狗定时器不使用时无需对其进行设置。

图 16-23. 在自编程模式下空白 block 校验操作流程示例



注 当看门狗定时器不使用时无需对其进行设置。

备注 图 16-23 中的<1>到<11> 对应于 16.8.7 中的<1>到<11>(前页)。

以下自编程模式下空白 block 校验程序示例

;-----

;开始

;-----

FlashBlockBlankCheck:

```

MOV    FLCMD,#04H    ;设置 flash 控制命令 (block 空白检查)
MOV    FLAPH,#07H    ;设置空白检查的 block 号 (这里指定 block 7)
;
MOV    FLAPL,#00H    ;FLAPL 恒为“00H”
MOV    FLAPHC,#07H   ;设置空白检查 block 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

;-----

16.8.8 自编程模式下字节写入操作示例

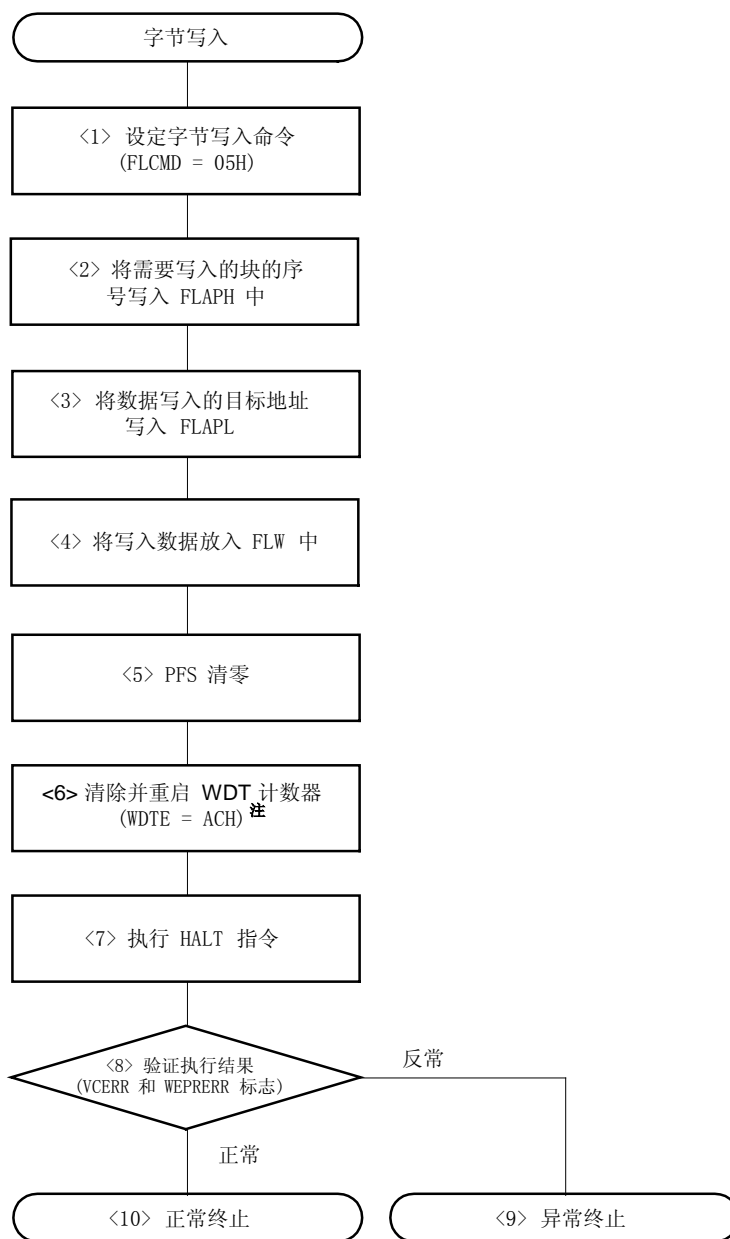
自编程模式下字节写入操作过程如下：

- <1> 设置编程命令寄存器(FLCMD)值为 05H(字节写入)。
- <2> 将被写入数据的 block 的序号放到 FLASH 地址指针 H 寄存器(FLAPH)中。
- <3> 将被写入数据的地址放到 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)[※]。
- <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> 字节写入进程非正常结束。
- <10> 字节写入进程正常结束。

注 当看门狗定时器不使用时无需对其进行设置。

注意事项 如果写入失败,应先擦除 block 内容并再次写入。

图 16-24. 自编程模式下字节写入操作流程示例



注 当看门狗定时器不使用时无需对其进行设置。

备注 图 16-24 中的<1>到<10> 对应于 16.8.8 中的<1>到<10>(前页)。

以下是自编程模式下字节写入程序示例。

```
;-----  
;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 block 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  
;-----
```

16.8.9 自编程模式下内部校验操作示例

自编程模式下内部校验操作过程如下。

- 内部校验 1

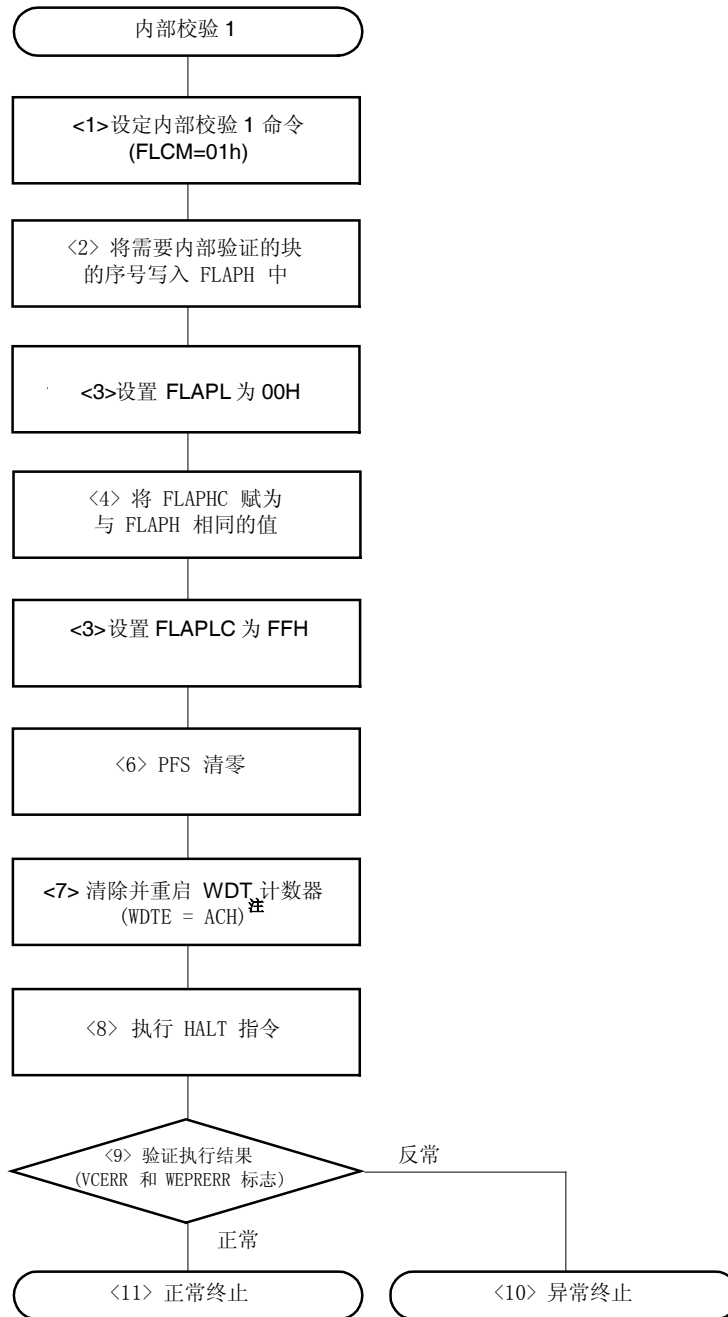
- <1> 设置编程命令寄存器(FLCMD)值为 01H(内部校验 1)。
- <2> 将需要执行内部校验的 block 的序号值放到到 FLASH 地址指针 H(FLAPH)中。
- <3> 设置 FLASH 地址指针 L(FLAPL)为 00H。
- <4> 写入和 FLAPH 相同的值到 FLASH 地址指针 L 比较寄存器(FLAPLC)中。
- <5> 设置 FLASH 地址指针 L 比较寄存器(FLAPL)为 FFH。
- <6> 清除 FLASH 状态寄存器(PFS)。
- <7> 写 ACH 到看门狗定时器开放寄存器(WDTE)(清 0 并重启看门狗定时计数器)^注。
- <8> 执行 HALT 指令然后开始自编程序(如果自编程序已开始执行, 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(WEPREERR)检查自编程序是否产生错误。
错误 → <10>
正常 → <11>
- <10> 内部校验非正常结束。
- <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 block 的序号值放到到 FLASH 地址指针 H(FLAPH)中。
- <3> 设置内部校验的起始地址到 FLASH 地址指针 L(FLAPL)中。
- <4> 将 FLAPH 的值写到的地址指针 H 比较寄存器(FLAPC)中。
- <5> 设置内部校验的结束地址到 FLASH 地址指针 L 比较寄存器(FLAPLC)中。
- <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 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序(如果自编程序已开始执行, 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(WEPREERR)检查自编程序是否产生错误。
错误 → <10>
正常 → <11>
- <10> 内部校验非正常结束。
- <11> 内部校验正常结束。

注 当看门狗定时器不使用时无需对其进行设置。

图 16-25. 自编程模式下内部校验 1 操作示例



注 当看门狗定时器不使用时无需对其进行设置。

备注 图 16-25 中的<1>到<11> 对应于 16.8.9 中的<1>到<11> (前页)。

图 16-27. 自编程模式下内部校验 2 操作示例



注 当看门狗定时器不使用时无需对其进行设置。

备注 图 16-26 中的<1>到<11> 对应于 16.8.9 中的<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 block 号到 FLAPH(例: 这里指定 Block 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 block 号到 FLAPH(例: 这里指定 Block 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
;-----

```

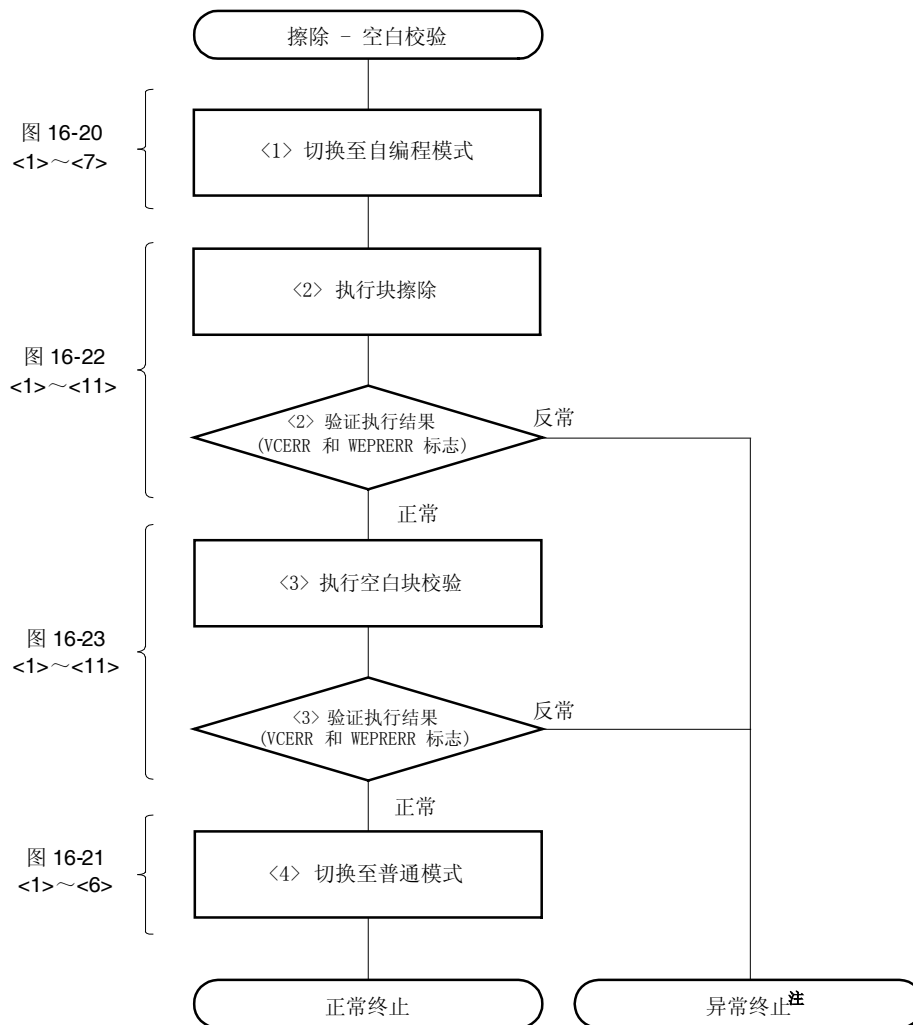

16.8.10 自编程模式下命令执行时间最短的操作示例

自编程模式下命令执行时间最短的操作过程如下。

(1) 从擦除到空白检验

- <1> 将工作模式由普通模式切换至自编程模式(16.8.4 中的<1> 到 <7>)。
- <2> 执行 block 擦除命令——错误检查 (16.8.6 中的<1> 到 <11>)。
- <3> 执行空白 block 检验——错误检查(16.8.7 中的<1> 到 <11>)。
- <4> 将工作模式由自编程模式切换回普通模式(16.8.5 中的<1> 到 <6>)。

图 16-27. 命令执行时间最短的操作流程示例(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普通模式。

备注 图 16-27 中<1>到<4>对应于 16.8.10(1)中的<1>到<4>。

以下是自编程模式下命令执行时间最短(擦除到 block 检测)的操作程序示例

```

;-----
;开始
;-----

        MOV     MK0,#11111111B    ;屏蔽所有中断
        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 (反相设置值)
        MOV     FLPMC,#01H        ;通过 FLPMC 寄存器控制设置自编程模式(设置值)
        ;

        NOP
        HALT
        BT      PFS.0,$ModeOnLoop; 检查写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(block 擦除)
        MOV     FLAPH,#07H        ;设置要擦除的 block 号 (这里指定 block 7)
        ;
        MOV     FLAPL,#00H        ;FLAPL 恒为“00H”
        MOV     FLAPHC,#07H       ;设置 block 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 (block 空白检测)
        MOV     FLAPH,#07H        ;设置空白检测的 block 号 (这里指定 block 7)
        ;
        MOV     FLAPL,#00H        ;FLAPL 恒为 “00H”

        MOV     FLAPHC,#07H       ;设置空白检测 block 比较数(与 FLAPH 值相同)
        ;

```

```

MOV     FLAPLC,#0FFH    ; FLAPLC 恒为 “FFH”
MOV     WDTE,#0ACH     ; 清零& 重启 WDT
HALT
MOV     A,PFS
CMP     A,#00H
BNZ     $StatusError   ; 检查空白检测错误
                                ; 当出现错误时执行异常结束处理
                                ;

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 (反向设置值)
MOV     FLPMC,#00H    ; 通过 FLPMC 寄存器控制设置普通模式)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

EI

BR     StatusNormal

;-----
;结束 (异常结束处理); 为了返回普通模式处理执行切换到普通模式的处理
;-----
StatusError:

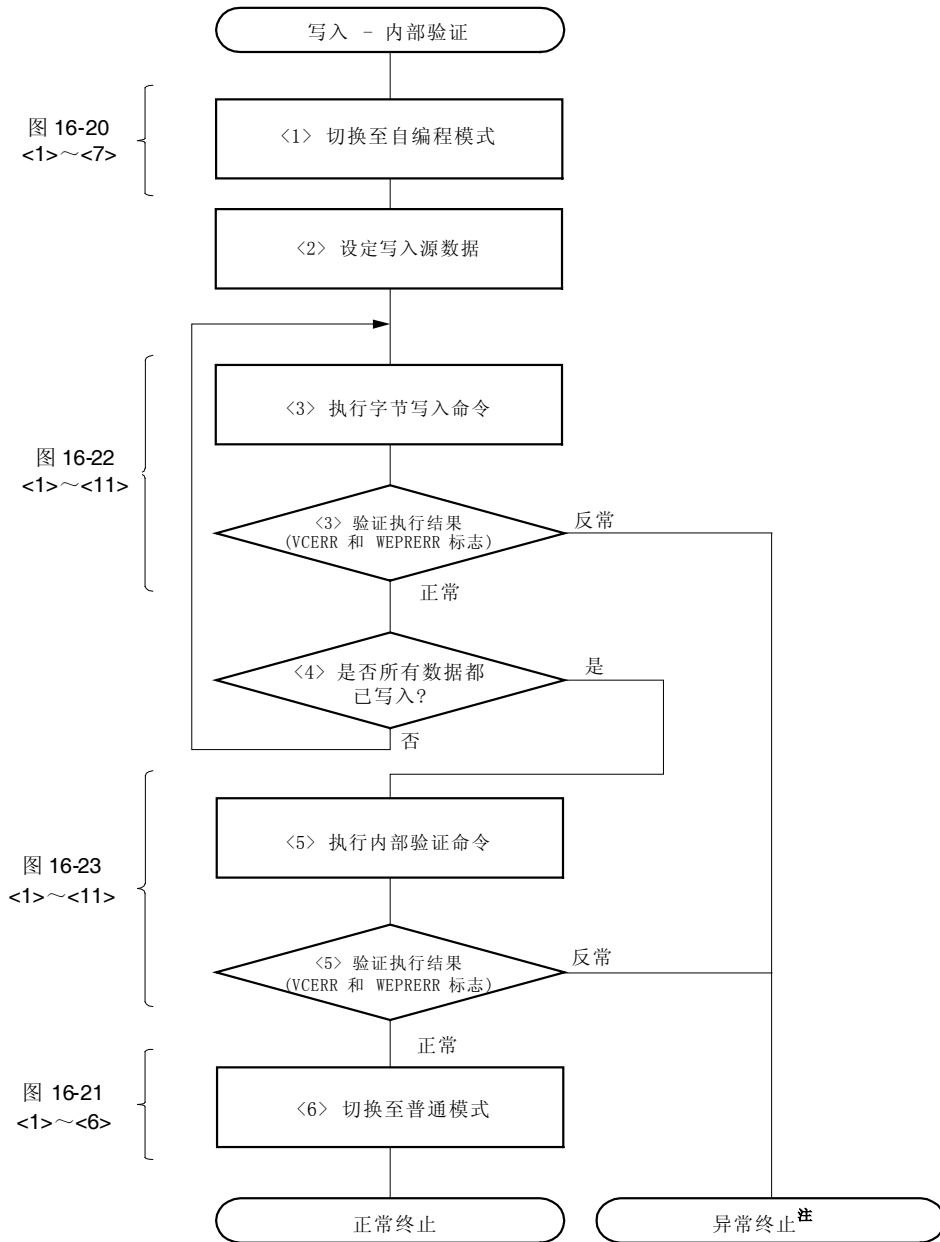
;-----
;结束 (正常结束处理)
;-----
StatusNormal:

```

(2) 从写操作到内部校验

- <1> 将工作模式由普通模式切换至自编程模式 (16.8.4 中的<1> 到 <7>)。
- <2> 确定需要写入的数据。
- <3> 执行 block 字节写入指令——错误检查 (16.8.8 中的<1> 到 <11>)。
- <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 (16.8.9 中的<1> 到 <11>)。
- <6> 将工作模式由自编程模式切换回普通模式 (16.8.5 中的<1> 到 <6>)。

图 16-28. 命令执行时间最短的操作流程(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普通模式。

备注 图 16-28 中<1>到<6>对应于 16.8.10(2)中的<1>到<6>。

以下是自编程模式下命令执行时间最短(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     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 (反向设置值)
        MOV     FLPMC,#01H        ;通过 FLPMC 寄存器控制设置自编程模式(设置值)
                                ;

        NOP
        HALT
        BT     PFS.0,$ModeOnLoop ;检查写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    ;定位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

        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                 ;定位数据要写入的地址 + 1
        MOVW    AX,HL
        CMPW    AX,#DataAdrBtm    ;执行内部校验处理
        BNC     $FlashVerify      ;如果所有数据写入完成

        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 (反向设置值)
MOV     FLPMC,#00H       ; 通过 FLPMC 寄存器控制设置普通模式 (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
                        ; 指定序列正常结束后, 自编程开始前恢复 CPU 时钟的设置
                        ;
                        ;

MOV     MK0,#INT_MK0     ; 恢复中断屏蔽标志

EI

BR     StatusNormal

;-----
; 结束 (异常结束处理); 要返回到普通程序执行切换到普通模式的处理
;-----
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 内部校验 2 用在上面的程序例子中。使用内部校验 1 验证一个完成的 block。

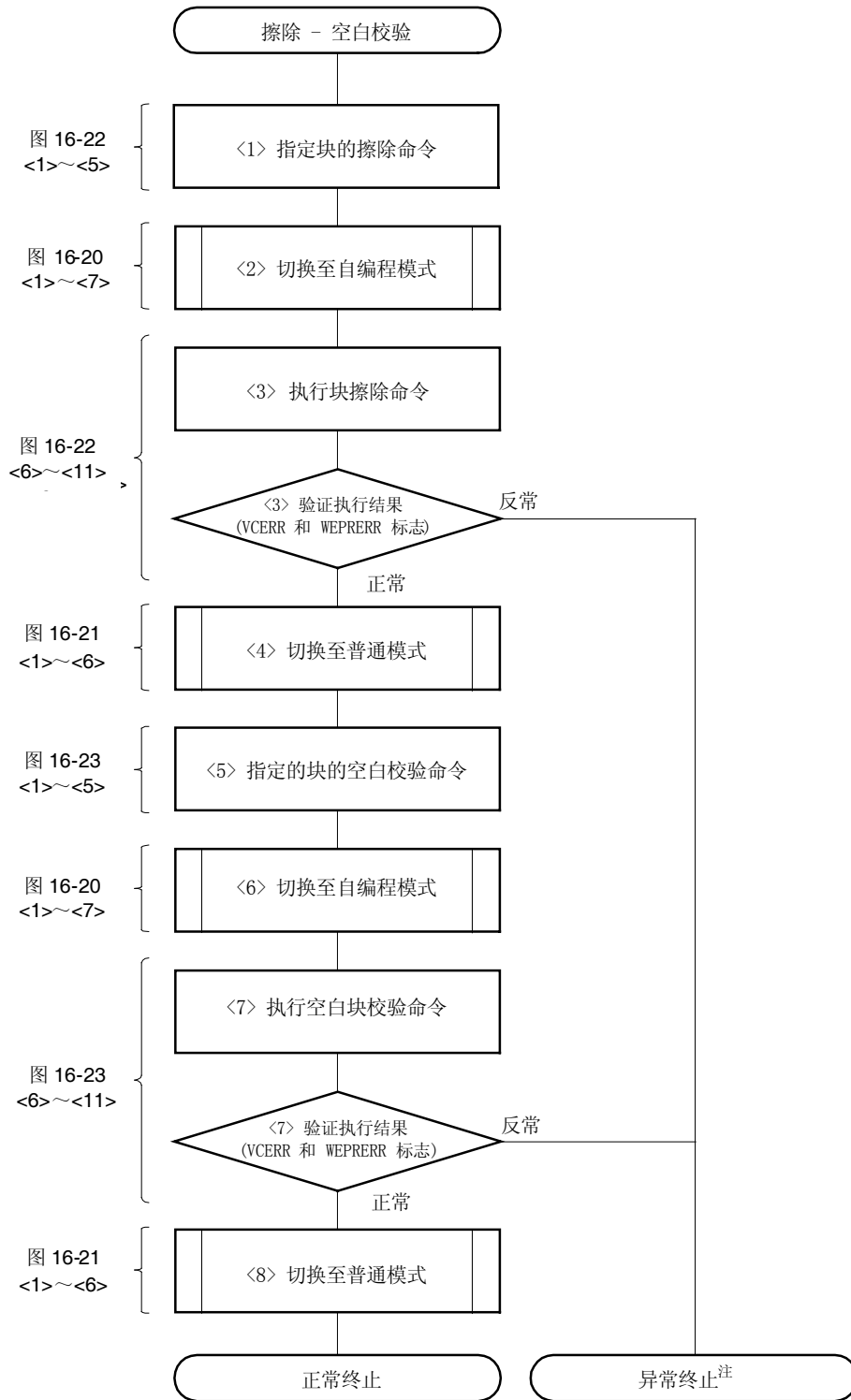
16.8.11 自编程模式下中断禁止时间最短的操作过程

编程模式下中断屏蔽时命令执行时间最短的操作过程如下

(1) 从擦除到空白检验

- <1> 设置控制命令为 block 擦除 (16.8.6 中的<1> 到 <5>)。
- <2> 将工作模式由普通模式切换至自编程模式 (16.8.4 中的<1> 到 <7>)。
- <3> 执行 block 擦除命令——错误检查 (16.8.6 中的<6> 到 <11>)。
- <4> 将工作模式由自编程模式切换至普通模式 (16.8.5 中的<1> 到 <6>)。
- <5> 设置空白 block 检验命令(16.8.7 中的<1> 到 <5>)。
- <6> 将工作模式由普通模式切换至自编程模式 (16.8.4 中的<1> 到 <7>)。
- <7> 执行空白 block 检验——错误检查 (16.8.7 中的<6> 到 <11>)。
- <8> 将工作模式由自编程模式切换回普通模式 (16.8.5 中的<1> 到 <6>)。

图 16-29. 自编程模式下中断屏蔽时命令执行时间最短的操作流程(从擦除到空白 block 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普通模式。

备注 图 16-39 中<1>到<8>对应于 16.8.11(1)中<1>到<8>(前页)。

以下是自编程模式下中断屏蔽时命令(从擦除到空白 block 检验)执行时间最短的程序示例。

```

;-----
;开始
;-----
FlashBlockErase:
    ;设置擦除命令
    MOV     FLCMD,#03H      ;设置 flash 控制命令(block 擦除)
    MOV     FLAPH,#07H     ;设置要擦除的 block 号 (这里指定 block 7)
    MOV     FLAPL,#00H     ;FLAPL 固定为“00H”
    MOV     FLAPHC,#07H    ;设置擦除 block 比较数 (与 FLAPH 值相同)
    MOV     FLAPLC,#00H    ;FLAPLC 固定为“00H”

    CALL    IModeOn        ;切换到自编程模式

    ;执行擦除命令
    MOV     PFS,#00H       ;flash 状态寄存器清零
    MOV     WDTE,#0ACH     ;清零&重启 WDT
    HALT                    ;自编程模式开始
    MOV     A,PFS
    CMP     A,#00H
    BNZ     $StatusError   ;检查擦除错误
                                ;出现错误时执行异常结束处理
                                ;

    CALL    IModeOff       ;切换到普通模式

    ;设置空白检测命令
    MOV     FLCMD,#04H     ;设置 flash 控制命令(block 空白检测)
    MOV     FLAPH,#07H     ;设置进行空白检测的 block 号 (这里指定 block 7)
    MOV     FLAPL,#00H     ;FLAPL 固定为“00H”
    MOV     FLAPHC,#07H    ;设置空白检测的 block 比较数(与 FLAPH 值相同)
                                ;
    MOV     FLAPLC,#0FFH   ;FLAPLC 固定为“FFH”

    CALL    IModeOn        ;切换到自编程模式

    ;执行空白检测命令
    MOV     PFS,#00H       ;flash 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   ;检查空白检测错误
                                ;发生错误时执行异常结束处理

    CALL    IModeOff       ;切换到普通模式

```

```

BR      StatusNormal

;-----
; 结束 (异常结束处理); 要返回到普通程序执行切换到普通模式的处理
;-----
StatusError:

;-----
; 结束 (正常结束处理)
;-----
StatusNormal:

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

    DI

ModeOnLoop:                    ; 配置设置使 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 (反向设置值)
    MOV    FLPMC,#01H        ; 通过 FLPMC 寄存器控制 设置自编程模式 (设置值)
    ;

    NOP
    HALT
    BT     PFS.0,$ModeOnLoop ; 检查写入指定寄存器完成
    ; 出现错误时重复相同的处理

    RET

;-----
; 切换到普通模式的程序
;-----
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 (反向设置值)
MOV    FLPMC,#00H    ; 通过 FLPMC 寄存器控制设置普通模式 (设置值)

BT     PFS.0,$ModeOffLoop ; 检查写入指定寄存器完成
                                ; 出现错误时重复相同的处理
                                ; 指定序列正常完成后, 在自编程前恢复 CPU 时钟设置
                                ;
                                ;
MOV    MK0,#INT_MK0  ; 恢复中断屏蔽标志

EI

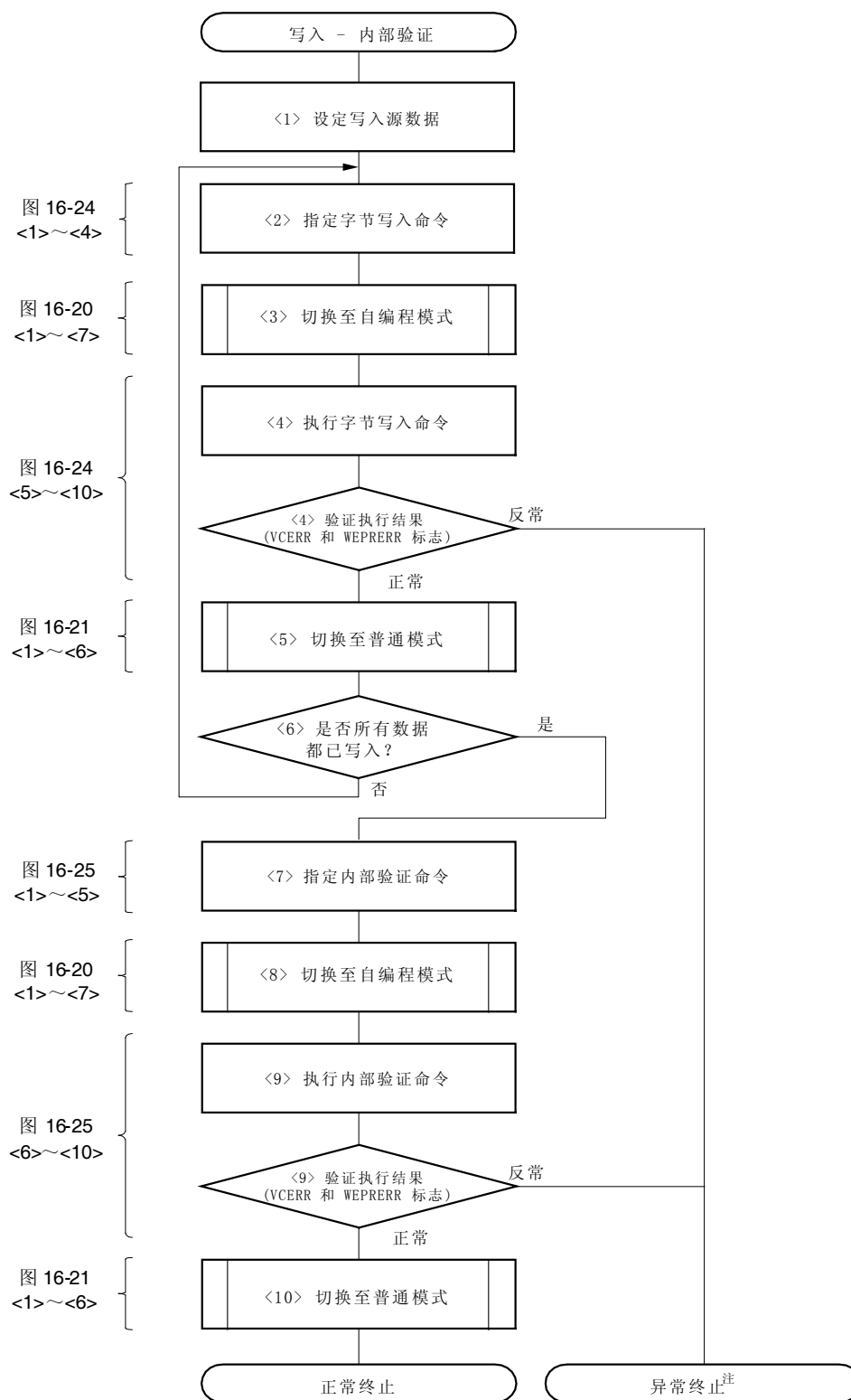
RET

```

(2) 从写操作到内部校验

- <1> 设置需要写入的数据。
- <2> 设置控制命令为字节写入(16.8.8 中的<1> 到 <4>)。
- <3> 将工作模式由普通模式切换到自编程模式(16.8.4 中的<1> 到 <7>)。
- <4> 执行字节写入命令——错误检查(16.8.8 中的<5>到<10>)。
- <5> 将工作模式由自编程模式切换回普通模式(16.8.5 中的<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(16.8.9 中的<1> 到 <5>)。
- <8> 将工作模式由普通模式切换到自编程模式(16.8.4 中的<1> 到 <7>)。
- <9> 执行内部校验指令——错误检查(16.8.9 中的<6> 到 <11>)。
- <10> 将工作模式由自编程模式切换回普通模式(16.8.5 中的<1> 到 <6>)。

图 16-30. 自编程模式下中断屏蔽时命令执行时间最短的操作流程(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普通模式。

备注 图 16-30 中<1>到<10>是相应于 16.8.11(2)中的<1>到<10>(前页)。

以下是自编程模式下中断屏蔽时命令(从写入到内部校验)执行时间最短的操作程序示。

```

;-----
;开始
;-----
        ; 设置写命令
FlashWrite:
        MOVW    HL,#DataAdrTop    ; 指定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          ; 切换到普通模式

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

EI

        ; 写全部数据的判断
        INCW    HL                ; 定位数据要写入的地址 + 1
        MOVW    AX,HL
        CMPW    AX,#DataAdrBtm   ; 执行内部校验处理
        BNC     $FlashVerify     ; 如果所有数据写入完成

        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    IModeOn        ;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
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    IModeOff       ;切换到普通模式

BR     StatusNormal

;-----
; 结束 (异常结束处理); 要返回到普通程序执行切换到普通模式的处理
;-----
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     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 (反向设置值)
MOV    FLPMC,#01H     ; 通过 FLPMC 寄存器控制 设置自编程模式 (设置值)
;

```

```

NOP
HALT
BT     PFS.0,$ModeOnLoop ; 检查写入到指定寄存器是否完成
; 出现错误时重复相同的处理

RET

```

```

;-----
; 切换到普通模式
;-----

```

```

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 (反向设置值)
MOV    FLPMC,#00H     ; 通过 FLPMC 寄存器控制 设置普通模式 (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，在自编程前恢复 CPU 时钟设置
;
;
MOV    MK0,#INT_MK0   ; 恢复中断屏蔽标志

EI

RET

```

```

;-----
; 写入数据
;-----

```

```

DataAdrTop:
DB     XXH
DB     XXH
DB     XXH
DB     XXH

:
:

```



```
        DB      XXH
DataAdrBtm:
;-----
```

备注 内部校验 2 用在上面的程序例子中。使用内部校验 1 验证一个完成的 block。

第十七章 指令集

本章列出了 78K0S/KU1+ 的指令设置。如需了解每种指令的操作和机器语言（操作代码），可参考 **78K/0S 系列指令用户手册（U11047E）**。

17.1 操作使用规则

17.1.1 操作数标识符和标识方法

根据规范确定的指令操作数表示方法（详情可参见汇编程序编程规范），在每种指令的“操作数”栏列出操作数。如果有两种或两种以上的表示方法，可任选其一。大写字母和符号 #, !, \$, 和 [] 是关键字，必须按其原样书写。每种符号的含义如下所示。

- #: 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志。当时用标志时，注意必须加上符号 #, !, \$ 和 []。

对应操作数寄存器标识符 r 和 rp，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）都可用于标识

表 17-1. 操作数标识符和标识方法

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) 专用寄存器符号
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）
word byte bit	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 专用寄存器符号参见 **表 3-3 专用寄存器列表**。

17.1.2 “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 寄存器对; 16 位累加器
BC:	BC 寄存器对
DE:	DE 寄存器对
HL:	HL 寄存器对
PC:	程序计数器
SP:	堆栈指针
PSW:	程序状态字
CY:	进位标志
AC:	辅助进位标志
Z:	零标志
IE:	中断请求使能标志
():	括号中德地址或寄存器所指的存储单元内容
xH, xL:	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 位立即数或标志
jdisp8:	带符号的八位数据 (偏移量)

17.1.3 “标志”栏的描述

(空):	不受影响
0:	清零
1:	设置为 1
x:	根据结果设置/清零
R:	恢复先前保存的值

17.2 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 ← byte			
	saddr, #byte	3	6	(saddr) ← byte			
	sfr, #byte	3	6	sfr ← byte			
	A, r ^{注 1}	2	4	A ← r			
	r, A ^{注 1}	2	4	r ← A			
	A, saddr	2	4	A ← (saddr)			
	saddr, A	2	4	(saddr) ← A			
	A, sfr	2	4	A ← sfr			
	sfr, A	2	4	sfr ← A			
	A, !addr16	3	8	A ← (addr16)			
	!addr16, A	3	8	(addr16) ← A			
	PSW, #byte	3	6	PSW ← byte	x	x	x
	A, PSW	2	4	A ← PSW			
	PSW, A	2	4	PSW ← A	x	x	x
	A, [DE]	1	6	A ← (DE)			
	[DE], A	1	6	(DE) ← A			
	A, [HL]	1	6	A ← (HL)			
	[HL], A	1	6	(HL) ← A			
	A, [HL + byte]	2	6	A ← (HL + byte)			
	[HL + byte], A	2	6	(HL + byte) ← A			
XCH	A, X	1	4	A ↔ X			
	A, r ^{注 2}	2	6	A ↔ r			
	A, saddr	2	6	A ↔ (saddr)			
	A, sfr	2	6	A ↔ sfr			
	A, [DE]	1	8	A ↔ (DE)			
	A, [HL]	1	8	A ↔ (HL)			
	A, [HL, byte]	2	8	A ↔ (HL + byte)			

- 注 1. r = A 除外
2. r = A, X 除外

备注 一个指令周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU})。

助记符	操作数	字节数	时钟	操作	标志		
					Z	AC	CY
MOVW	rp, #word	3	6	rp ← word			
	AX, saddrp	2	6	AX ← (saddrp)			
	saddrp, AX	2	8	(saddrp) ← AX			
	AX, rp ^注	1	4	AX ← rp			
	rp, AX ^注	1	4	rp ← AX			
XCHW	AX, rp ^注	1	8	AX ↔ rp			
ADD	A, #byte	2	4	A, CY ← A + byte	x	x	x
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte	x	x	x
	A, r	2	4	A, CY ← A + r	x	x	x
	A, saddr	2	4	A, CY ← A + (saddr)	x	x	x
	A, laddr16	3	8	A, CY ← A + (addr16)	x	x	x
	A, [HL]	1	6	A, CY ← A + (HL)	x	x	x
	A, [HL + byte]	2	6	A, CY ← A + (HL + byte)	x	x	x
ADDC	A, #byte	2	4	A, CY ← A + byte + CY	x	x	x
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte + CY	x	x	x
	A, r	2	4	A, CY ← A + r + CY	x	x	x
	A, saddr	2	4	A, CY ← A + (saddr) + CY	x	x	x
	A, laddr16	3	8	A, CY ← A + (addr16) + CY	x	x	x
	A, [HL]	1	6	A, CY ← A + (HL) + CY	x	x	x
	A, [HL + byte]	2	6	A, CY ← A + (HL + byte) + CY	x	x	x
SUB	A, #byte	2	4	A, CY ← A - byte	x	x	x
	saddr, #byte	3	6	(saddr), CY ← (saddr) - byte	x	x	x
	A, r	2	4	A, CY ← A - r	x	x	x
	A, saddr	2	4	A, CY ← A - (saddr)	x	x	x
	A, laddr16	3	8	A, CY ← A - (addr16)	x	x	x
	A, [HL]	1	6	A, CY ← A - (HL)	x	x	x
	A, [HL + byte]	2	6	A, CY ← A - (HL + byte)	x	x	x

注 仅当 rp = BC, DE 或 HL 时。

备注 一个指令周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU})。

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	$A, CY \leftarrow A - (\text{HL}) - CY$	×	×	×
	A, [HL + byte]	2	6	$A, CY \leftarrow A - (\text{HL} + \text{byte}) - CY$	×	×	×
AND	A, #byte	2	4	$A \leftarrow A \wedge \text{byte}$	×		
	saddr, #byte	3	6	$(saddr) \leftarrow (saddr) \wedge \text{byte}$	×		
	A, r	2	4	$A \leftarrow A \wedge r$	×		
	A, saddr	2	4	$A \leftarrow A \wedge (saddr)$	×		
	A, !addr16	3	8	$A \leftarrow A \wedge (\text{addr16})$	×		
	A, [HL]	1	6	$A \leftarrow A \wedge (\text{HL})$	×		
	A, [HL + byte]	2	6	$A \leftarrow A \wedge (\text{HL} + \text{byte})$	×		
OR	A, #byte	2	4	$A \leftarrow A \vee \text{byte}$	×		
	saddr, #byte	3	6	$(saddr) \leftarrow (saddr) \vee \text{byte}$	×		
	A, r	2	4	$A \leftarrow A \vee r$	×		
	A, saddr	2	4	$A \leftarrow A \vee (saddr)$	×		
	A, !addr16	3	8	$A \leftarrow A \vee (\text{addr16})$	×		
	A, [HL]	1	6	$A \leftarrow A \vee (\text{HL})$	×		
	A, [HL + byte]	2	6	$A \leftarrow A \vee (\text{HL} + \text{byte})$	×		
XOR	A, #byte	2	4	$A \leftarrow A \nabla \text{byte}$	×		
	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	$A \leftarrow A \nabla (\text{HL})$	×		
	A, [HL + byte]	2	6	$A \leftarrow A \nabla (\text{HL} + \text{byte})$	×		

备注 一个指令周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（ f_{CPU} ）。

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, laddr16	3	8	A – (addr16)	×	×	×
	A, [HL]	1	6	A – (HL)	×	×	×
	A, [HL + byte]	2	6	A – (HL + byte)	×	×	×
ADDW	AX, #word	3	6	AX, CY ← AX + word	×	×	×
SUBW	AX, #word	3	6	AX, CY ← AX – word	×	×	×
CMPW	AX, #word	3	6	AX – word	×	×	×
INC	r	2	4	r ← r + 1	×	×	
	saddr	2	4	(saddr) ← (saddr) + 1	×	×	
DEC	r	2	4	r ← r – 1	×	×	
	saddr	2	4	(saddr) ← (saddr) – 1	×	×	
INCW	rp	1	4	rp ← rp + 1			
DECW	rp	1	4	rp ← rp – 1			
ROR	A, 1	1	2	(CY, A ₇ ← A ₀ , A _{m-1} ← A _m) × 1			×
ROL	A, 1	1	2	(CY, A ₀ ← A ₇ , A _{m+1} ← A _m) × 1			×
RORC	A, 1	1	2	(CY ← A ₀ , A ₇ ← CY, A _{m-1} ← A _m) × 1			×
ROLC	A, 1	1	2	(CY ← A ₇ , A ₀ ← CY, A _{m+1} ← A _m) × 1			×
SET1	saddr.bit	3	6	(saddr.bit) ← 1			
	sfr.bit	3	6	sfr.bit ← 1			
	A.bit	2	4	A.bit ← 1			
	PSW.bit	3	6	PSW.bit ← 1	×	×	×
	[HL].bit	2	10	(HL).bit ← 1			
CLR1	saddr.bit	3	6	(saddr.bit) ← 0			
	sfr.bit	3	6	sfr.bit ← 0			
	A.bit	2	4	A.bit ← 0			
	PSW.bit	3	6	PSW.bit ← 0	×	×	×
	[HL].bit	2	10	(HL).bit ← 0			
SET1	CY	1	2	CY ← 1			1
CLR1	CY	1	2	CY ← 0			0
NOT1	CY	1	2	CY ← \overline{CY}			×

备注 一个指令周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU})。

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	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $PSW \leftarrow (SP + 2), SP \leftarrow SP + 3, NMIS \leftarrow 0$	R	R	R
PUSH	PSW	1	2	$(SP - 1) \leftarrow PSW, SP \leftarrow SP - 1$			
	rp	1	4	$(SP - 1) \leftarrow rp_H, (SP - 2) \leftarrow rp_L, SP \leftarrow SP - 2$			
POP	PSW	1	4	$PSW \leftarrow (SP), SP \leftarrow SP + 1$	R	R	R
	rp	1	6	$rp_H \leftarrow (SP + 1), rp_L \leftarrow (SP), SP \leftarrow SP + 2$			
MOVW	SP, AX	2	8	$SP \leftarrow AX$			
	AX, SP	2	6	$AX \leftarrow SP$			
BR	!addr16	3	6	$PC \leftarrow \text{addr16}$			
	\$addr16	2	6	$PC \leftarrow PC + 2 + \text{jdisp8}$			
	AX	1	6	$PC_H \leftarrow A, PC_L \leftarrow X$			
BC	\$saddr16	2	6	$PC \leftarrow PC + 2 + \text{jdisp8}$ 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, \$saddr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ if (saddr.bit) = 1			
	sfr.bit, \$saddr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ if sfr.bit = 1			
	A.bit, \$saddr16	3	8	$PC \leftarrow PC + 3 + \text{jdisp8}$ if A.bit = 1			
	PSW.bit, \$saddr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ if PSW.bit = 1			
BF	saddr.bit, \$saddr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ if (saddr.bit) = 0			
	sfr.bit, \$saddr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ if sfr.bit = 0			
	A.bit, \$saddr16	3	8	$PC \leftarrow PC + 3 + \text{jdisp8}$ if A.bit = 0			
	PSW.bit, \$saddr16	4	10	$PC \leftarrow PC + 4 + \text{jdisp8}$ if PSW.bit = 0			
DBNZ	B, \$saddr16	2	6	$B \leftarrow B - 1,$ then $PC \leftarrow PC + 2 + \text{jdisp8}$ if B \neq 0			
	C, \$saddr16	2	6	$C \leftarrow C - 1,$ then $PC \leftarrow PC + 2 + \text{jdisp8}$ if C \neq 0			
	saddr, \$saddr16	3	8	$(\text{saddr}) \leftarrow (\text{saddr}) - 1,$ then $PC \leftarrow PC + 3 + \text{jdisp8}$ if (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 一个指令周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU})。

17.3 按寻址类型列出指令

(1) 8 位指令

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

第 2 操作数 第 1 操作数	#byte	A	r	sfr	saddr	laddr16	PSW	[DE]	[HL]	[HL + byte]	\$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV [#] XCH [#] 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
laddr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											
[HL + byte]		MOV											

注 r = A 除外。

(2) 16 位指令

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

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

注 仅当 rp = BC, DE 或 HL 时。

(3) 位操作指令

SET1, CLR1, NOT1, BT, BF

第 1 操作数 \ 第 2 操作数	\$addr16	None
A.bit	BT BF	SET1 CLR1
sfr.bit	BT BF	SET1 CLR1
saddr.bit	BT BF	SET1 CLR1
PSW.bit	BT BF	SET1 CLR1
[HL].bit		SET1 CLR1
CY		SET1 CLR1 NOT1

(4) 调用指令/转移指令

CALL, CALLT, BR, BC, BNC, BZ, BNZ, DBNZ

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

(5) 其他指令

RET, RETI, NOP, EI, DI, HALT, STOP

第十八章 电气特性

最大额定值 (T_A = 25°C)

参数	符号	条件	范围	单位
电源电压	V _{DD}		-0.3 ~ +6.5	V
	V _{SS}		-0.3 ~ +0.3	V
输入电压	V _I	P20 ~ P23, P32, P34, P40, P43	-0.3 ~ V _{DD} + 0.3 ^注	V
输出电压	V _O		-0.3 ~ V _{DD} + 0.3 ^注	V
模拟输入电压	V _{AN}		-0.3 ~ V _{DD} + 0.3 ^注	V
输出电流, 高	I _{OH}	每个引脚	-10.0	mA
		P20 ~ P23, P32, P40, P43 的总和	-44.0	mA
输出电流, 低	I _{OL}	每个引脚	20.0	mA
		P20 ~ P23, P32, P40, P43 的总和	44.0	mA
工作环境温度	T _A	在通常操作模式下	-40 ~ +85	°C
		在 Flash 存储器编程期间		°C
存储温度	T _{stg}	Flash 存储器空白状态	-65 ~ +150	°C
		Flash 存储器已被编程	-40 ~ +125	°C

注 必须为 6.5V 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，因为，必须保证产品在不超过最大额定值的条件下使用。

备注 除了另外的规定，复用功能引脚的特性与端口引脚的相同。

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

振荡器	建议使用的电路	参数	条件	最小	典型	最大	单位
陶瓷振荡器		振荡频率 (f_x) ^{#2}		1		10.0	MHz
晶体振荡器		振荡频率 (f_x) ^{#2}		1		10.0	MHz
外部时钟		X1 输入频率 (f_x) ^{#2}	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	1		10.0	MHz
			$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	1		5.0	
		X1 输入高/低电平宽度 (t_{KH} , t_{KL})	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0.045		0.5	s
			$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	0.09		0.5	

- 注
1. 该产品的电压使用范围为 2.2 ~ 5.5 V，因为上电清零电路(POC)的检测电压(V_{POC})是 2.1 V \pm 0.1 V。
 2. 仅表示振荡器的特性。如需了解指令执行时间，可参见 [交流特性](#)。

注意事项 在使用 X1 振荡器时，上表中的虚线框内部分的连线应遵从如下的连接方式，避免线间电容产生不利影响。

- 连接线越短越好。
- 连接线不与其他信号线交叉。
- 如果信号线流经的电流变化较大，则不要在其周围连线。
- 要保持振荡器电容器的接地点电压与 V_{SS} 相同。
- 避免大电流从电容到地的连线上流过。
- 不要从振荡器获取信号。

备注 对于振荡器选择和振荡器常量，用户需要自己测试振荡器或要求制造商测试。

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	典型	最大	单位
高速内部振荡器	振荡频率($f_X = 8 \text{ MHz}^{\#2}$)偏差	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			± 3	%
			$T_A = -10 \sim +70^\circ\text{C}$			
			$T_A = -40 \sim +85^\circ\text{C}$			± 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. 如需了解指令执行时间，可参见 **交流特性**。

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

振荡器	参数	条件	最小	典型	最大	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 Characteristics ($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = 2.0 \sim 5.5 \text{ V}^{\#}$, $V_{SS} = 0 \text{ V}$) (1/2)

参数	符号	条件		最小	典型	最大	单位
输出电流, 高	I _{OH}	每个引脚	$2.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			-5	mA
		所有引脚	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			-25	mA
			$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$			-15	mA
输出电流, 低	I _{OL}	每个引脚	$2.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			10	mA
		所有引脚	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$			30	mA
			$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$			15	mA
输入电压, 高	V _{IH1}	外部时钟模式下的 P23 和除去 P20、P21 的其它引脚		0.8V _{DD}		V _{DD}	V
	V _{IH2}	非外部时钟模式下的 P23、P20、P21 引脚		0.7V _{DD}		V _{DD}	V
输入电压, 低	V _{IL1}	外部时钟模式下的 P23 和除去 P20、P21 的其它引脚		0		0.2V _{DD}	V
	V _{IL2}	非外部时钟模式下的 P23、P20、P21 引脚		0		0.3V _{DD}	V
输出电压, 高	V _{OH}	所有输出引脚 I _{OH} = -15 mA	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$ I _{OH} = -5 mA	V _{DD} - 1.0			V
		I _{OH} = -100 A	$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	V _{DD} - 0.5			V
输出电压, 低	V _{OL}	所有输出引脚 I _{OL} = 30 mA	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$ I _{OL} = 10 mA			1.3	V
		$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$ I _{OL} = 400 A				0.4	V
输入漏电流, 高	I _{LIH}	V _I = V _{DD}	除 X1 以外的引脚			1	A
输入漏电流, 低	I _{LIL}	V _I = 0 V	除 X1 以外的引脚			-1	A
输出漏电流, 高	I _{LOH}	V _O = V _{DD}	除 X2 以外的引脚			1	A
输出漏电流, 低	I _{LOL}	V _O = 0 V	除 X2 以外的引脚			-1	A
上拉电阻	R _{PU}	V _I = 0 V		10	30	100	kΩ
下拉电阻	R _{PD}	P22, P23, 复位状态		10	30	100	kΩ

注 该产品的电压使用范围为 2.2 ~ 5.5 V，因为上电清零电路(POC)的检测电压(V_{POC})是 2.1 V ± 0.1 V。

备注 除了另外的规定, 复用功能引脚的特性与端口引脚的相同。

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

参数	符号	条件		最小	典型	最大	单位	
电源电流 ^{注2}	I _{DD1} ^{注3}	晶体/陶瓷振荡器, 外部时钟输入振荡器操作模式 ^{注6}	f _x = 10 MHz V _{DD} = 5.0 V ±10% ^{注4}	A/D 转换器终止操作		6.1	12.2	mA
				A/D 转换器正在操作中		7.6	15.2	
			f _x = 6 MHz V _{DD} = 5.0 V ±10% ^{注4}	A/D 转换器终止操作		5.5	11.0	mA
				A/D 转换器正在操作中			14.0	
			f _x = 5 MHz V _{DD} = 3.0 V ±10% ^{注5}	A/D 转换器终止操作		3.0	6.0	mA
				A/D 转换器正在操作中		4.5	9.0	
	I _{DD2}	晶体/陶瓷振荡器, 外部时钟输入 HALT 模式 ^{注6}	f _x = 10 MHz V _{DD} = 5.0 V ±10% ^{注4}	外设操作终止		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}	外设操作终止		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}	外设操作终止		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 转换器终止操作		5.0	10.0	mA
				A/D 转换器正在操作中		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}	外设操作终止		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	A	
			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	A	
			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 ~ 5.5 V, 因为上电清零电路(POC)的检测电压(V_{POC})是 2.1 V ±0.1 V。
 2. 流经内部电源(V_{DD})的总电流, 也包括外设操作电流(但不包括流经端口上拉电阻的电流)。
 3. 包括外设操作电流。
 4. 当处理器时钟控制寄存器 PCC=00H。
 5. 当处理器时钟控制寄存器 PCC=02H。
 6. 由可选字节选择晶体/陶瓷振荡器, 外部时钟输入作为系统时钟源。
 7. 由可选字节选择高速内部振荡器作为系统时钟源。

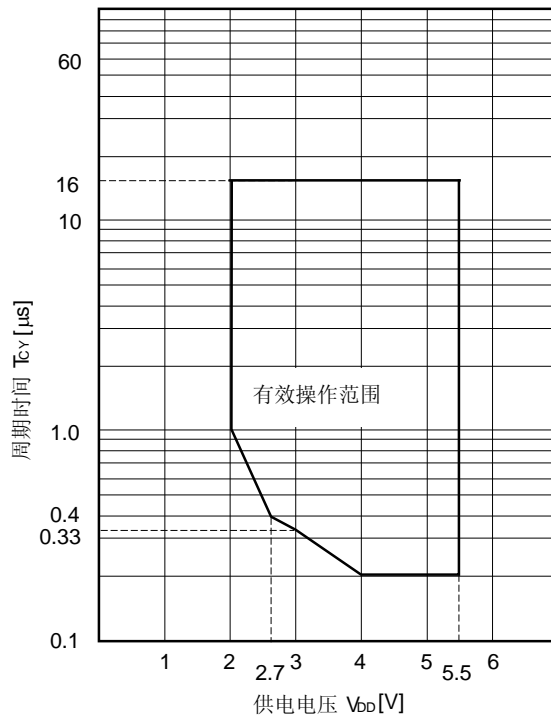
AC 特性

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

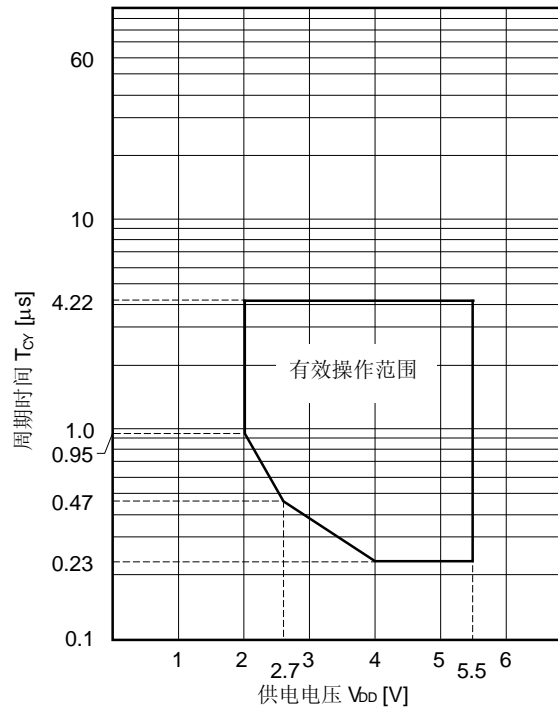
参数	符号	条件	最小	典型	最大	单位	
指令周期 (指令最短执行时间)	T_{CY}	晶体/陶瓷振荡器, 外部时钟输入	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0.2		16	s
			$3.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	0.33		16	s
			$2.7 \text{ V} \leq V_{DD} < 3.0 \text{ V}$	0.4		16	s
			$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	1		16	s
		高速内部振荡器时钟	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0.23		4.22	s
			$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	0.47		4.22	s
$2.0 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	0.95			4.22	s		
TI000/TI010 输入高电平宽度, 低电平宽度	t_{TIH}	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$2/f_{sam} + 0.1^{\#2}$			s	
	t_{TIL}	$2.0 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	$2/f_{sam} + 0.2^{\#2}$			s	
中断输入高电平宽度, 低电平宽度	t_{INTH}		1			s	
	t_{INTL}					s	
RESET 低电平宽度	t_{RSL}		2			s	

- 注
1. 该产品的电压使用范围为 2.2 ~ 5.5 V, 因为上电清零电路(POC)的检测电压(V_{POC})是 2.1 V \pm 0.1 V。
 2. 根据预分频模式寄存器 00(PRM00)的第 0 和第 1 位(PRM000, PRM001)可以选 $f_{sam} = f_{XP}$, $f_{XP}/4$ 或 $f_{XP}/256$ 。注意当选择 TI000/TI010 有效沿作为计数时钟时, $f_{sam} = f_{XP}$ 。

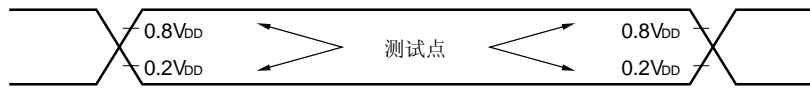
T_{CY} vs. V_{DD} (晶体/陶瓷振荡器时钟, 外部时钟输入)



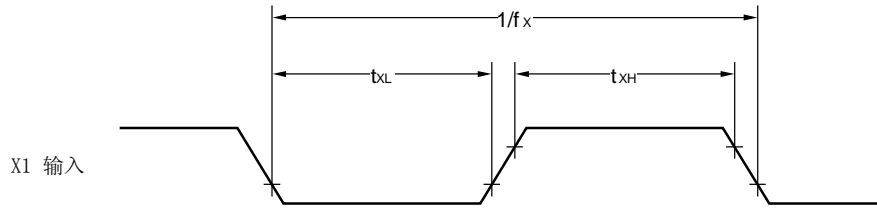
T_{CY} vs. V_{DD} (高速内部振荡器时钟)



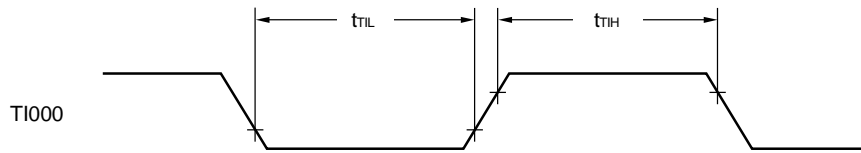
交流时序测试点(不包括 X1 输入)



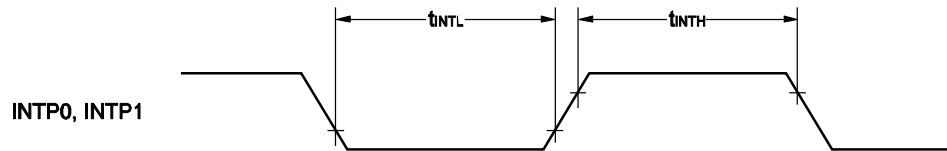
时钟时序



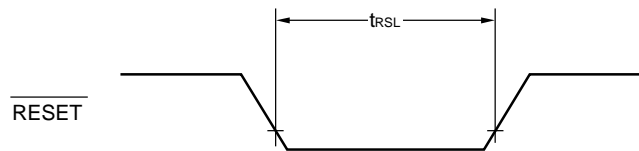
T1000 时序



中断输入时序



$\overline{\text{RESET}}$ 输入时序



A/D 转换器特性 ($T_A = -40 \sim +85^\circ\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}^{\#1}$, $V_{SS} = 0\text{ V}^{\#2}$)

(1) A/D 转换器基本特性

参数	符号	条件	最小	典型	最大	单位
分辨率			10	10	10	bit
转换时间	t_{CONV}	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	3.0		100	μs
		$4.0\text{ V} \leq V_{DD} < 4.5\text{ V}$	4.8		100	μs
		$2.85\text{ V} \leq V_{DD} < 4.0\text{ V}$	6.0		100	μs
		$2.7\text{ V} \leq V_{DD} < 2.85\text{ V}$	14.0		100	μs
模拟输入电压	V_{AIN}		$V_{SS}^{\#2}$		V_{DD}	V

(2) A/D 转换器特性(高速内部振荡器时钟)

参数	符号	条件	最小	典型	最大	单位
总误差 ^{#3, 4}	AINL			$-0.1 \sim +0.2^{\#5}$	$-0.35 \sim +0.45$	%FSR
零度误差 ^{#3, 4}	Ezs			$-0.1 \sim +0.2^{\#5}$	$-0.35 \sim +0.45$	%FSR
满度误差 ^{#3, 4}	Efs			$-0.1 \sim +0.2^{\#5}$	$-0.35 \sim +0.40$	%FSR
积分非线性误差 ^{#4}	ILE			$\pm 1^{\#5}$	± 3	LSB
微分非线性误差 ^{#3}	DLE			$\pm 1^{\#5}$	± 1.5	LSB

(3) A/D 转换器特性(晶体/陶瓷振荡时钟, 外部时钟)

参数	符号	条件	最小	典型	最大	单位
总误差 ^{#1, 2}	AINL	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		$-0.20 \sim +0.35^{\#5}$	$-0.35 \sim +0.65$	%FSR
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		$\pm 0.25^{\#5}$	$-0.35 \sim +0.55$	%FSR
零度误差 ^{#3, 4}	Ezs	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		$-0.20 \sim +0.35^{\#5}$	$-0.35 \sim +0.65$	%FSR
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		$\pm 0.25^{\#5}$	$-0.35 \sim +0.55$	%FSR
满度误差 ^{#3, 4}	Efs	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		$-0.20 \sim +0.35^{\#5}$	$-0.35 \sim +0.55$	%FSR
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		$\pm 0.25^{\#5}$	$-0.35 \sim +0.50$	%FSR
积分非线性误差 ^{#3}	ILE	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		$\pm 1.5^{\#5}$	± 3.0	LSB
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		$\pm 1.5^{\#5}$	± 4.0	LSB
微分非线性误差 ^{#3}	DLE	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		$\pm 1.0^{\#5}$	± 2.5	LSB
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		$\pm 1.0^{\#5}$	± 2.5	LSB

- 注
1. 在 78K0S/KU1+ 中, V_{DD} 复用为 A/D 转换器的参考电压输入。当使用 A/D 转换器时, V_{DD} 使用稳定电源 (2.7 ~ 5.5 V)。
 2. 在 78K0S/KU1+ 中, V_{SS} 复用为 A/D 转换器的地。要确保 V_{SS} 连接到一个稳定的 GND (= 0V)。
 3. 不包括量化误差 ($\pm 1/2$ LSB)。
 4. 该值表示满度的百分比 (%FSR)。
 5. A/D 转换开始后立刻通过指令设置 HALT 模式的值。

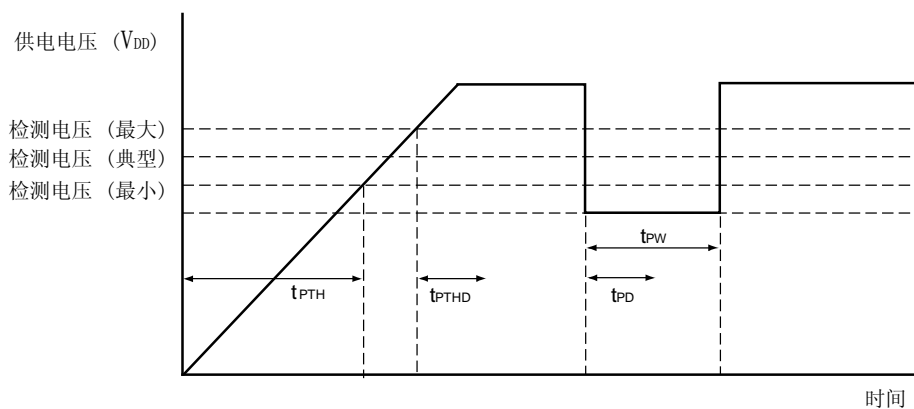
注意事项 当模拟输入引脚被用于复用 I/O 端口时, 或当 A/D 转换时如果未用作 A/D 转换的端口的电平被改变时, 转换精度会降低。

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

参数	符号	条件	最小	典型	最大	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			s
响应延迟时间 1 ^{※1}	t_{PTHD}	上电时, 并达到检测电压(最大值)后			3.0	ms
响应延迟时间 2 ^{※2}	t_{PD}	下电时			1.0	ms
最短脉冲宽度	t_{PW}		0.2			ms

- 注 1. 从电压检测到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



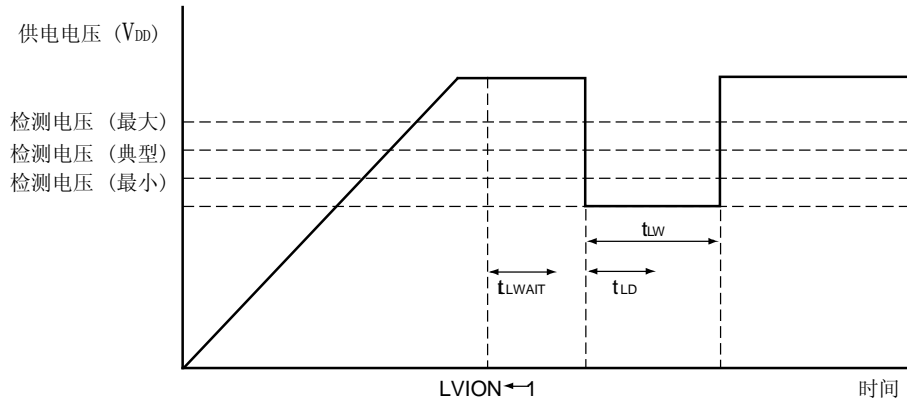
LVI Circuit Characteristics ($T_A = -40 \sim +85^\circ\text{C}$)

参数	符号	条件	最小	典型	最大	单位
检测电压	V_{LV10}		4.1	4.3	4.5	V
	V_{LV11}		3.9	4.1	4.3	V
	V_{LV12}		3.7	3.9	4.1	V
	V_{LV13}		3.5	3.7	3.9	V
	V_{LV14}		3.3	3.5	3.7	V
	V_{LV15}		3.15	3.3	3.45	V
	V_{LV16}		2.95	3.1	3.25	V
	V_{LV17}		2.7	2.85	3.0	V
	V_{LV18}		2.5	2.6	2.7	V
	V_{LV19}		2.25	2.35	2.45	V
响应时间 ^{注1}	t_{LD}			0.2	2.0	ms
最小脉冲宽度	t_{LW}		0.2			ms
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 $LVION = 1$ 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. $V_{POC} < V_{LVm} (m = 0 \sim 9)$

LVI 电路时序

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

参数	符号	条件	最小	典型	最大	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

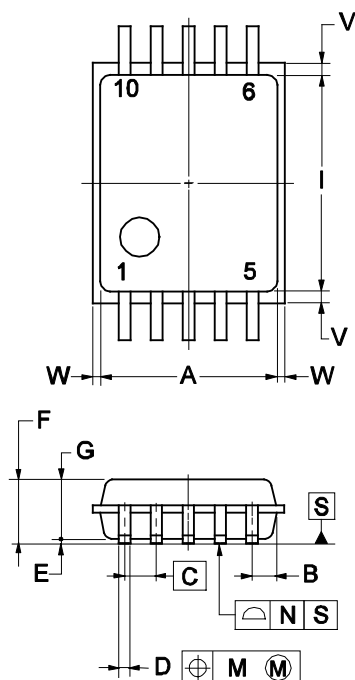
Flash 存储器编程特性 ($T_A = -40 \sim +85^\circ\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$)

参数	符号	条件	最小	典型	最大	单位
电源电压	I_{DD}	$V_{DD} = 5.5\text{ V}$			7.0	mA
擦除次数 ^注 (每 1 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
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		3.3	s
		$T_A = -40 \sim +85^\circ\text{C}$, $N_{ERASE} \leq 100$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0.9	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		1.0	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		1.1	s
		$T_A = -40 \sim +85^\circ\text{C}$, $N_{ERASE} \leq 1000$	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		4.9	s
			$3.5\text{ V} \leq V_{DD} < 4.5\text{ V}$		5.4	s
			$2.7\text{ V} \leq V_{DD} < 3.5\text{ V}$		6.6	s
字节写入时间	T_{WRITE}	$T_A = -40 \sim +85^\circ\text{C}$, $N_{ERASE} \leq 1000$			150	μs
内部验证时间	T_{VERIFY}	每个 block			6.8	ms
		每个字节			27	μs
空白检验	T_{BLKCHK}	每个 block			480	μs
保存年限		$T_A = 85^\circ\text{C}$ ^{注 2} , $N_{ERASE} \leq 1000$	10			年

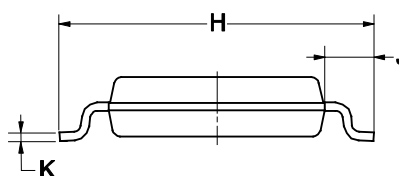
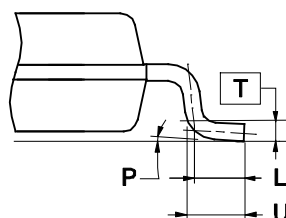
注 1. 该值与擦除计数(N_{ERASE})，擦除时间变化有关，参考芯片擦除时间和块擦除时间参数。
2. 操作和不操作的平均温度是 85°C 。

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。

10 引脚塑封 SSOP(5.72mm(225))



引脚端详图



(单位:mm)

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

P10MA-65-CAG

注

该结构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 0.13mm 内

第二十章 推荐的焊接条件

建议在下列条件下焊接和装贴。
相关技术信息，请浏览下面的网页。

半导体器件装贴手册(<http://www.necel.com/pkg/en/mount/index.html>)

- 注意事项 1.** 产品尾缀 **-A** 是无铅产品。
2. 如不使用以下的焊接和装贴条件请与 **NECEL** 公司销售代理联系。

表 21-1. 表面贴装焊接条件

10 引脚塑封 SSOP(无铅产品)

μ PD78F9200MA-CAC-A, 78F9201MA-CAC-A, 78F9202MA-CAC-A

焊接方法	焊接条件	推荐条件符号
红外回流焊	封装峰值温度: 235°C, 时间: 最大 30 秒。(大于等于 210°C), 次数: 小于等于 3 次, 暴露限制: 7 天 ^注 (之后, 在 125°C 预烘 20 小时)	IR35-207-3
波峰焊	详情请联系 NECEL 销售部门	-
局部加热	引脚温度: 最大 350°C, 时间: 最大 3 秒 (每行引脚)	-

注 打开干燥包之后, 有效存放期内将其存储在温度 25°C (或更低), 湿度 65% RH (或更低) 的环境中。

注意事项 不要将不同的焊接方法一起使用 (局部加热法除外)。

附录 A 开发工具

下面的开发工具在使用 78K0S/KU1+ 的系统开发中使用。图 A-1 显示了开发系统的组成。

- **PC98-NX 系列兼容机**

除非特别说明，IBM PC/AT™ 支持的产品和 PC98-NX 系列兼容。当使用 PC98-NX 系列时可参考 IBM PC/AT 兼容机使用说明。

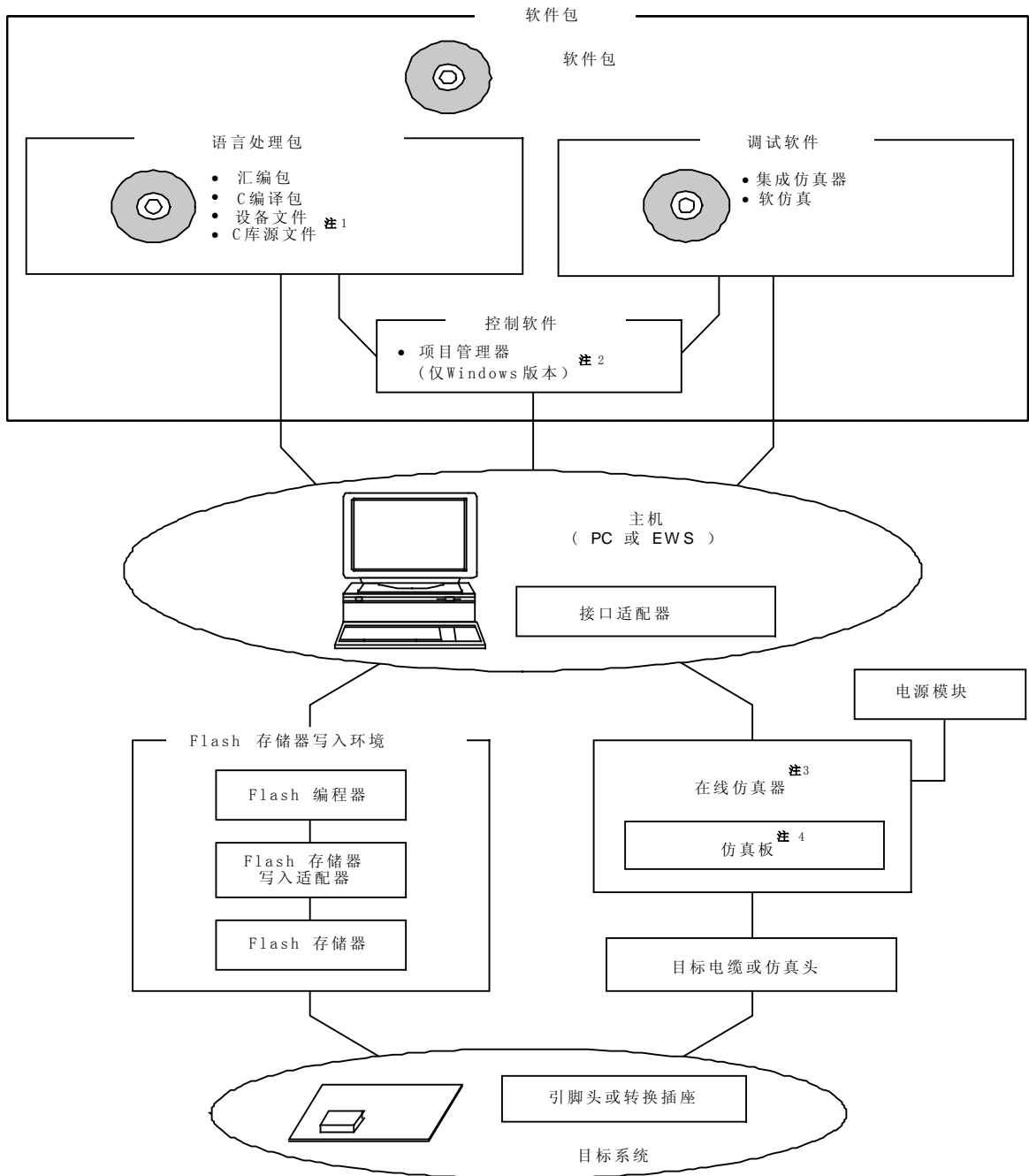
- **Windows™**

除非特别说明，“Windows”是指以下操作系统。

- Windows 98
- Windows NT™ Ver. 4.0
- Windows 2000
- Windows XP

图 A-1. 开发工具(1/2)

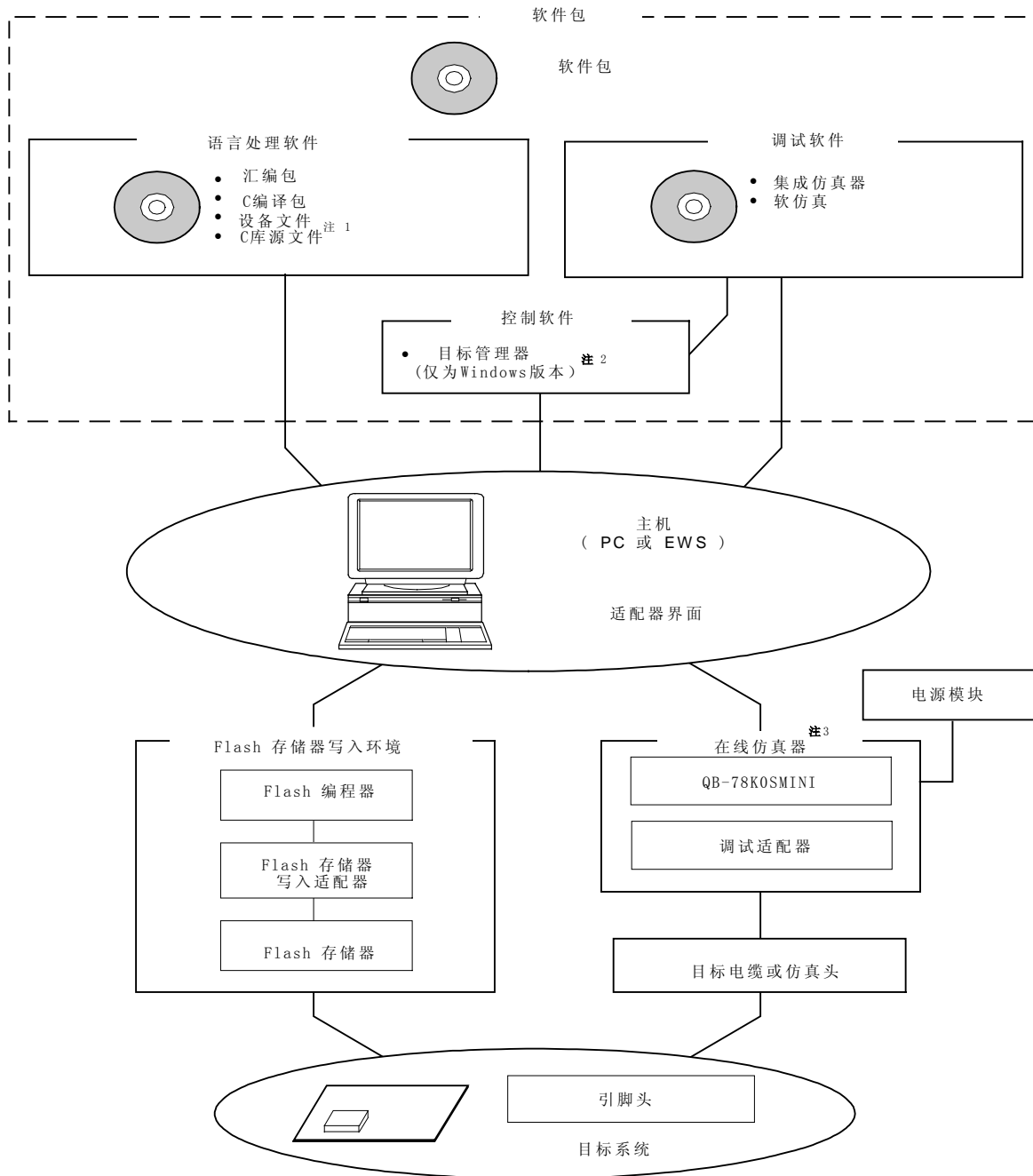
(1) 当使用在线仿真器 IE-78K0S-NS 或 IE-78K0S-NS-A 时



- 注
1. 软件包中不包括 C 库原文件。
 2. 汇编包中包括项目管理器 PM+。
PM+ 应用于 windows 环境。
 3. 除了在线仿真器 IE-78K0S-NS 和 IE-78K0S-NS-A 外的其他产品都是单独销售的。
 4. 在线仿真器 IE-789234-NS-EM1 提供目标电缆。

图 A-1. 开发工具(2/2)

(2) 当使用在线仿真器 QB-78K0SKX1MINI 时



- 注
1. 软件包中不包括 C 库原文件。
 2. 汇编包中包括项目管理器 PM+ 。
PM+ 应用于 windows 环境。
 3. 在线仿真器 QB-78K0SKX1MINI 提供集成调试环境 ID78K0S-QB，Flash 编程器 PG-FPL2，电源模块和目标电缆，其他产品都单独销售。

A.1 软件包

SP78K0S 软件包	此软件包用于开发 78K/0S 系列。 包含如下工具。 RA78K0S, CC78K0S, ID78K0S-NS, SM+ for 78K0S ^{※1} , SM78K0S ^{※2} 和设备文件。
	产品编号: μ SxxxxSP78K0S

- 注
1. SP78K0S Ver. 2.00 或更早的版本中不包括 SM+。
 2. SM78K0S 不支持 78K0S/Kx1+。
 3. 设备文件 DF789234 没有包含在 SP78K0S Ver. 2.00 或更早的版本中。

备注 产品编号中的xxxx 随使用的操作系统而定。

μ SxxxxSP78K0S

xxxx	主机	操作系统	媒体
AB17	PC-9800 系列, IBM PC/AT 和兼容机	日文 Windows	CD-ROM
BB17		英文 Windows	

A.2 语言处理软件

RA78K0S 汇编编译	该程序将编写的助记符程序转换为可在 MCU 中运行的目标代码。 可自动产生符号表和优化转移指令。 和设备文件(DF789234) (单独销售)一起使用。 < PC 环境使用的注意事项 > 该汇编包是基于 DOS 运行的, 但使用 PM+ (包含在汇编包中)可在 windows 中使用。 产品编号: μ SxxxxRA78K0S
CC78K0S C 语言编译	该程序将编写的 C 语言程序转换为可在 MCU 中运行的目标代码。 该编译器应与 汇编包 (RA78K0S) 和设备文件 (DF789234)(二者均独立销售)结合使用。 < PC 环境使用的注意事项 > C 编译包是基于 DOS 运行的, 但使用 PM+ (包含在汇编包中)可在 windows 中使用。 产品编号: μ SxxxxCC78K0S
DF789234 ^{※1} 设备文件	该文件包含该器件特有的信息。 与其他工具联合使用 (RA78K0S, CC78K0S, ID78K0S-NS, ID78K0S-QB, 或 SM+ 对应 78K0S)。 产品编号: μ SxxxxDF789234
CC78K0S-L ^{※2} C 库源文件	函数库的源文件, 可根据用户的特殊需求去改变 C 编译库中的目标库。由于是源文件, 它的使用不随操作系统改变。 产品编号: μ SxxxxCC78K0S-L

- 注 1. μ DF789234 是一个通用文件, 可被 RA78K0S, CC78K0S, ID78K0S-NS, ID78K0S-QB, 和 SM+ 对应 78K0S 使用。
2. SP78K0S 不包括 CC78K0S-L。

备注 产品编号中的xxxx 随使用的操作系统而定。

μ SxxxxRA78K0S

μ SxxxxCC78K0S

μ SxxxxCC78K0S-L

xxxx	主机	操作系统	媒体
AB17	PC-9800 系列, IBM PC/AT 及其兼容机	日文 Windows	CD-ROM
BB17		英文 Windows	
3P17	HP9000 系列 700 TM	HP-UX TM (Rel. 10.10)	
3K17	SPARC 工作站 TM	SunOS TM (Rel. 4.1.4), Solaris TM (Rel. 2.5.1)	

μ SxxxxDF789234

xxxx	主机	操作系统	媒体
AB13	PC-9800 系列, IBM PC/AT 及其兼容机	日文 Windows	3.5" 2HD 软盘
BB13		英文 Windows	

A.3 控制软件

PM+ 项目管理器	这个控制软件可以使用户在 windows 下高效的开发。启动编辑器, 构造程序, 启动调试器, 都能够在 PM+下执行。 < 注意事项 > PM+ 包含在汇编包 (RA78K0S)中。只能在 windows 环境运行。
--------------	---

A.4 Flash 存储器写入工具

Flashpro IV (FL-PR4, PG-FP4) Flash 存储器编程器	只适用与 flash 型程序存储器的 MCU。
PG-FPL2 Flash 存储器编程器	只适用与 flash 型程序存储器的 MCU。
QB-MINI2 具有写入功能的片上调试仿真器	可作为 Flash 存储器编程器专用于微控制器。也可在使用 Flash 微控制器(包括 78K0S/Kx1+)开发应用系统时, 片上调试仿真器用于调试硬件和软件。
FA-78F9212GR-JJG-MX Flash 存储器写入适配器	Flash 存储器写入适配器。用于与 Flash 存储器编程器连接。

备注 FL-PR4 和 FA-78F9212GR-JJG-MX 是 Naito Densei Machida Mfg. Co., Ltd 的产品。
联系: Naito Densei Machida Mfg. Co., Ltd. (电话 +81-42-750-4172)

A.5 调试工具(硬件)

A.5.1 当使用在线仿真器 QB-78K0SKX1 时(开发中)

QB-78K0SKX1 (开发中) 在线仿真器	在使用 78K0S/Kx1+ 开发应用系统时，线仿真器服务于调试硬件和软件。它提供集成调试器 (ID78K0S-QB)。它可通过 USB 接口线连接包括交流适配器，目标插座和主机。
QB-50-EP-01T (开发中) 仿真适配头	仿真适配头是一个灵活的类型，被用于连接在线仿真器和目标系统。
QB-16GR-EA-01T (开发中) 转换适配器	转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-16GR-NQ-01T (开发中) 目标连接器	目标连接器被用来连接目标系统。
目标系统引脚针说明	0.635 mm × 0.635 mm (高: 6 mm)

A.5.2 当使用在线仿真器 QB-MINI2 时

QB-MINI2 具有编程功能的片上调试仿真器	在使用 Flash 微控制器(包括 78K0S/Kx1+)开发应用系统时，片上调试仿真器用于调试硬件和软件。也可作为 Flash 存储器编程器专用于微控制器。
目标系统引脚针说明	16 引脚通用目标连接器(2.54 mm pitch)

A.5.3 当使用在线仿真器 IE-78K0S-NS 或 IE-78K0S-NS-A 时

IE-78K0S-NS 在线仿真器	用于在线调试 78K/0S 系列的硬件和软件应用。支持集成调试器(ID78K0S-NS)。与 AC 适配器，仿真插头，和连接在主机的接口适配器联合使用。
IE-78K0S-NS-A 在线仿真器	覆盖 IE-78K0S-NS 的所有功能，并且提高了调试功能如：跟踪、定时器功能。
IE-70000-MC-PS-B 交流适配器	从 100 ~ 240 VAC 中提供仿真器用 DC 电源。
IE-70000-CD-IF-A PC 接口卡	笔记本电脑做主机时需要的接口卡和电缆(使用 PCMCIA 接口)。
IE-70000-PCI-IF-A 接口适配器	有 PCI 总线的计算机做主机时需要的接口适配器。
IE-789234-NS-EM1 仿真板	用于仿真器件特有的外围设备的仿真板。 结合在线仿真器使用，提供一根目标电缆。
目标系统插座的规格	0.635 mm × 0.635 mm (高: 6 mm)

A.5.4 当使用在线仿真器 QB-78K0SKX1MINI 时

QB-78K0SKX1MINI 在线仿真器	用于在线调试 78K0S/Kx1+ 系列的硬件和软件应用。支持集成调试器(ID78K0S-QB)。和交流适配器，目标电缆，连接主机的 USB 接口电缆联合使用。
目标系统插座的规格	0.635 mm × 0.635 mm (高: 6 mm)

A.6 调试工具(软件)

ID78K0S-NS (支持在线仿真器 IE-78K0S-NS/ IE-78K0S-NS-A) 集成调试器	用于 78K0S 系列的在线仿真。 ID78K0S-NS 是基于 windows 的软件。 该调试器增强了 C 语言的调试功能。通过使用集成窗口功能, 结合源程序, 分解显示, 和跟踪结果的存储区显示, 显示源程序的跟踪结果和设备文件配合使用 (DF789234) (独立销售)。
	产品编号: $\mu S \times \times \times$ ID78K0S-NS
ID78K0S-QB (支持在线仿真器 QB-78K0SKX1MINI) 集成调试器	用于 78K0S/Kx1+ 系列的在线仿真。 ID78K0S-QB 是基于 windows 的软件。 提供 C 语言调试功能, 可以源程序, 分解显示。存储区显示。和设备文件配合使用 (DF789234) (独立销售)。 在线仿真器 QB-78K0SKX1MINI 中提供。
	产品编号: $\mu S \times \times \times$ ID78K0S-QB (不销售)
SM+ for 78K0S 软件仿真器	用于 78K0S 系列的软件仿真。 SM+ for 78K0S 是基于 windows 的软件。 在主机上模拟目标系统的操作, 可执行 C 语言和汇编语言。 使用 SM+ for 78K0S, 可以在不需要外围硬件的情况下仿真程序逻辑和功能, 从而提供了较高的开发效率和质量和设备文件配合使用 (DF789234) (独立销售)。
	产品编号: $\mu S \times \times \times$ SM789234-B
DF789234 [#] 设备文件	该文件包含该器件特有的信息。 与其他工具联合使用 (RA78K0S, CC78K0S, ID78K0S-NS, ID78K0S-QB, 或 SM+ for 78K0S)。
	产品编号: $\mu S \times \times \times$ DF789234

注 DF789234 是一个通用文件, 可被 RA78K0S, CC78K0S, ID78K0S-NS, ID78K0S-QB, 和 SM+ 对于 78K0S 使用。

备注 产品编号中的xxxx随使用的操作系统和媒体而定。

μSxxxxID78K0S-NS

μSxxxxID78K0S-QB

μSxxxxSM789234-NS

xxxx	主机	操作系统	媒体
BB13	PC-9800 系列, IBM PC/AT 及其兼容机	英文 Windows	3.5" 2HD 软盘
AB17		日文 Windows	
BB17		英文 Windows	CD-ROM

μSxxxxDF789234

xxxx	主机	操作系统	媒体
AB13	PC-9800 系列, IBM PC/AT 及其兼容机	日文 Windows	3.5" 2HD 软盘
BB13		英文 Windows	

B.1 寄存器索引(按寄存器名称排序)

- 8 位 A/D 转换结果寄存器(ADCRH) ... 154
- 8 位定时器 H 比较寄存器 01 (CMP01) ... 122
- 8 位定时器 H 比较寄存器 11 (CMP11) ... 122
- 8 位定时器 H 模式寄存器 1 (TMHMD1) ... 123
- 10 位 A/D 转换结果寄存器 (ADCR) ... 153
- 16 位定时器捕捉/比较寄存器 000 (CR000) ... 81
- 16 位定时器捕捉/比较寄存器 010 (CR010) ... 83
- 16 位定时计数器 00 (TM00) ... 81
- 16 位定时器模式控制寄存器 00 (TMC00) ... 84
- 16 位定时器输出控制寄存器 00 (TOC00) ... 87

[A]

- A/D 转换模式寄存器 (ADM) ... 150
- 模拟输入通道指定寄存器 (ADS) ... 153

[C]

- 捕捉/比较控制寄存器 00 (CRC00) ... 86

[E]

- 外部中断模式寄存器 0 (INTM0) ... 169

[F]

- Flash 地址指针 H 比较寄存器 (FLAPHC) ... 228
- Flash 地址指针 L 比较寄存器 (FLAPLC) ... 228
- Flash 地址指针 H (FLAPH) ... 228
- Flash 地址指针 L (FLAPL) ... 228
- Flash 编程命令寄存器 (FLCMD) ... 227
- Flash 编程模式控制寄存器 (FLPMC) ... 223
- Flash 保护命令寄存器 (PFCMD) ... 224
- Flash 状态寄存器 (PFS) ... 225
- Flash 写缓冲寄存器 (FLW) ... 229

[I]

- 中断屏蔽标志寄存器 0 (MK0) ... 169
- 中断请求标志寄存器 0 (IF0) ... 168

[L]

- 低速内部振荡模式寄存器 (LSRCM) ... 66
- 低电压检测寄存器 (LVIM) ... 198
- 低电压检测电平选择寄存器 (LVIS) ... 199

[O]

- 振荡稳定时间选择寄存器 (OSTS) ... 67

[P]

端口模式控制寄存器 2 (PMC2) ... 58, 89, 125, 154

端口模式寄存器 2 (PM2) ... 57, 89, 125, 154

端口模式寄存器 3 (PM3) ... 57

端口模式寄存器 4 (PM4) ... 57

端口寄存器 2 (P2) ... 58

端口寄存器 3 (P3) ... 58

端口寄存器 4 (P4) ... 58

预处理器时钟控制寄存器 (PPCC) ... 65

预分频模式寄存器 00 (PRM00) ... 88

处理器时钟控制寄存器 (PCC) ... 65

上拉电阻选择寄存器 2 (PU2) ... 60

上拉电阻选择寄存器 3 (PU3) ... 60

上拉电阻选择寄存器 4 (PU4) ... 60

[R]

复位控制标志寄存器 (RESF) ... 192

[W]

看门狗定时器使能寄存器 (WDTE) ... 139

看门狗定时器模式寄存器 (WDTM) ... 138

B.2 寄存器索引(按寄存器符号排序)**[A]**

ADCR: 10 位 A/D 转换结果寄存器... 153
 ADCRH: 8 位 A/D 转换结果寄存器... 154
 ADM: A/D 转换模式寄存器 ... 150
 ADS: 模拟输入通道指定寄存器... 153

[C]

CMP01: 8 位定时器 H 比较寄存器 01 ... 122
 CMP11: 8 位定时器 H 比较寄存器 11 ... 122
 CR000: 16 位定时器捕捉/比较寄存器 000 ... 81
 CR010: 16 位定时器捕捉/比较寄存器 010 ... 83
 CRC00: 捕捉/比较控制寄存器 00 ... 86

[F]

FLAPH: Flash 地址指针 H ... 228
 FLAPHC: Flash 地址指针 H 比较寄存器... 228
 FLAPL: Flash 地址指针 L ... 228
 FLAPLC: Flash 地址指针 L 比较寄存器 ... 228
 FLCMD: Flash 编程命令寄存器... 227
 FLPMC: Flash 编程模式控制寄存器 ... 223
 FLW: Flash 写缓冲寄存器 ... 229

[I]

IFO: 中断请求标志寄存器 0 ... 168
 INTM0: 外部中断模式寄存器 0 ... 169

[L]

LSRCM: 低速内部振荡模式寄存器... 66
 LVIM: 低电压检测寄存器... 198
 LVIS: 低电压检测电平选择寄存器... 199

[M]

MK0: 中断屏蔽标志寄存器 0 ... 169

[O]

OSTS: 振荡稳定时间选择寄存器... 67

[P]

P2:	端口寄存器 2 ... 58
P3:	端口寄存器 3 ... 58
P4:	端口寄存器 4 ... 58
PCC:	处理器时钟控制寄存器... 65
PFCMD:	Flash 保护命令寄存器 ... 224
PFS:	Flash 状态寄存器... 225
PM2:	端口模式寄存器 2 ... 57, 89, 125, 154
PM3:	端口模式寄存器 3 ... 57
PM4:	端口模式寄存器 4 ... 57
PMC2:	端口模式控制寄存器 2 ... 58, 89, 125, 154
PPCC:	预处理器时钟控制寄存器... 65
PRM00:	预分频模式寄存器 00 ... 88
PU2:	上拉电阻选择寄存器 2 ... 60
PU3:	上拉电阻选择寄存器 3 ... 60
PU4:	上拉电阻选择寄存器 4 ... 60

[R]

RESF:	复位控制标志寄存器... 192
-------	------------------

[T]

TM00:	16 位定时计数器 00 ... 81
TMC00:	16 位定时器模式控制寄存器 00 ... 84
TMHMD1:	8 位定时器 H 模式寄存器 1 ... 123
TOC00:	16 位定时器输出控制寄存器 00 ... 87

[W]

WDTE:	看门狗定时器使能寄存器... 139
WDTM:	看门狗定时器模式寄存器... 138

附录 C 注意事项列表

这个附录把各个章节的注意事项列表出来。

表中的“分级(硬件/软件)”如下。

硬件： 芯片的内部/外部硬件的注意事项

软件： 软件方面的注意事项，例如寄存器的设置或者编程

(1/14)

章节	分级	功能	功能细节	注意事项	页码
第二章	硬件	引脚功能	P22/X2/ANI2, P23/X1/ANI3	P22/X2/ANI2 和P23/X1/ANI3引脚复位期间为下拉状态。	pp. 19, 20, 21, 22 <input type="checkbox"/>
		存储空间	SP：堆栈指针	由于复位输入使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 <input type="checkbox"/>
第三章	软件			堆栈指针只能设置在高速RAM区域中，并且，只能设置实际设置(SP)的低10位。 因此，如果堆栈指针指定为0FF00H，它将被转换为高速RAM区域中的0FB00H，因为0FF00H在SFR区域，并且不在高速RAM区域。 当有数据被压栈，它的指向将减1，从0FB00H变为0FAFFH，但是因为这个值不在高速RAM区域中，它将被转换成0FEFFH，这与堆栈指针被设置为0FF00H时是一样的。	p. 34 <input type="checkbox"/>
		端口功能	P22/X2/ANI2, P23/X1/ANI3	P22/X2/ANI2和P23/X1/ANI3引脚复位期间为下拉状态。	p. 49 <input type="checkbox"/>
第四章	硬件		P34	因为引脚P34可以复用为RESET引脚使用，如果当它作为输入端口引脚使用，从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，具体细节请参看 第十五章 选项字节。 同时，因为在复位释放后才读取选项字节，如果在读取选项字节之前，一个低电平被输入到RESET引脚，则复位状态不能释放。当用做输入端口时，要连接上拉电阻。	p. 55 <input type="checkbox"/>
			P21, P32	因为P21和P32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。因此使用这个端口为输出模式时，预先设置相应的中断屏蔽标志为1。	p. 57 <input type="checkbox"/>
			PMC2：端口模式控制寄存器2	当PMC20 ~ PMC23设定为1时，P20/ANI0 ~ P23/ANI3引脚的端口功能不能被使用。在引脚设置为A/D转换器模式时要确保上拉电阻选择寄存器(PU20 ~ PU23)设置为0。	p. 59 <input type="checkbox"/>
			—	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，如果一个端口既可输入，又可输出，则对于端口中定义为输入模式的引脚的输出锁存器中的内容是不确定的。	p. 61 <input type="checkbox"/>

章节	分级	功能	功能细节	注意事项	页码
第五章	时钟发生器	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 \geq 预期的振荡器振荡稳定时间。	p. 67 <input type="checkbox"/>
				无论STOP模式是被复位信号打破还是被中断打破, STOP模式释放后的等待时间不包括从释放STOP模式到晶振开始震荡的时间(下图中的a)。	p. 67 <input type="checkbox"/>
				上电或者复位后的振荡稳定时间可以通过选项字节来设置, 详情参见 第十五章 选项字节。	p. 67 <input type="checkbox"/>
第五章	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, 这样可以减小分布电容对振荡器的影响。 <ul style="list-style-type: none"> 保持连线尽可能的短。 不要使其它的信号线与振荡器导线互相交叉, 不要在振荡器旁边通过大电流的信号线。 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。 	p. 68 <input type="checkbox"/>
第六章	硬件	16位定时器/事件计数器00	TM00: 16位定时器计数器00	TM00 被读取时, CR010无法捕捉其值。	pp. 81, 113 <input type="checkbox"/>
				如果TM00在定时器计数时被读取, 在读取期间, 定时器计数将停止, 直到读取工作结束, 才继续计数。因此, 读取时, 定时器计数将产生错误。	pp. 81, 113 <input type="checkbox"/>
	软件	CR000: 16位捕捉/比较寄存器 000		TM00和CR000相等时进入的清零&启动模式中, CR000置非0000H值。当此寄存器作为外部事件计数器使用时, 不能实现仅对一个脉冲的计数。然而, 在自由运行模式和由TI000引脚有效沿产生的清零&启动模式中, 如果CR000置0000H, TM00溢出后从0000H变为0001H时, 产生一个中断请求(INTTM00)。	pp. 82, 113 <input type="checkbox"/>
				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。	p.82, 113 <input type="checkbox"/>
				当16位定时器/事件计数器00停止工作后, CR000的值是不确定的。	pp. 82, 114 <input type="checkbox"/>
				当CR000设为比较模式, 即使有捕捉触发信号输入, 也不响应捕捉操作。	pp. 82, 116 <input type="checkbox"/>
硬件			当P21引脚作为TI010有效沿输入端口使用时, P21引脚则不能作为定时器输出(TO00)使用。而且, 当P21用作 TO00时, 也不能作为TI010有效沿输入引脚使用。	pp. 82, 118 <input type="checkbox"/>	

章节	功能	功能细节	注意事项	页码	
第六章	硬件	16位定时器/事件计数器00	CR000: 16位捕捉/比较寄存器000	当CR000作为捕捉寄存器使用, 如果同时发生寄存器的读操作和输入捕捉触发操作, 则捕捉触发输入有优先权, 而读出的数据为不确定值。同样, 如果定时器停止计数与捕捉触发输入同时发生, 捕捉触发为不确定状态。	pp. 82, 115
				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(17)在定时器操作中改变比较寄存器的注意事项。	p. 82
	软件		CR010: 16位定时器捕捉/比较寄存器010	在自由运行模式和由TI000引脚有效沿产生的清零&启动模式中, 如果CR010置0000H, TM00溢出后从0000H变为0001H时, 产生一个中断请求(INTTM010)。	pp. 83, 113
				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新值小于原来的值, 则必须在改变赋值后重新启动定时器。	pp. 83, 113
				当16位定时器/事件计数器00停止工作后, CR010的值是不确定的。	pp. 83, 114
				当CR010设为比较模式, 即使有捕捉触发信号输入, 也不响应捕捉操作。	pp. 83, 116
	硬件			当CR010作为捕捉寄存器使用, 如果同时发生寄存器的读操作和输入捕捉触发操作, 则捕捉触发输入有优先权, 而读出的数据为不确定值。同样, 如果定时器停止计数与捕捉触发输入同时发生, 捕捉触发为不确定状态。	pp. 83, 115
				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(17)在定时器操作中改变比较寄存器的注意事项。	p. 84
	软件		TMC00: 16位定时器模式控制寄存器00	当TMC002、TMC003不等于0时, 16位定时器计数器00(TM00)开始工作, 而当TMC002和TMC003等于0时, 停止操作。	pp. 85, 113
				除了OVF00标志位, 其他位写入数据时, 定时器必须停止操作。	pp. 85, 114
	软件			如果定时器停止, 即使TI000/TI010引脚有信号输入, 定时器计数和中断也不响应。	pp. 85, 113
				除非当TI000引脚有效沿用于计数时钟, 在设置定时器STOP模式或系统时钟停止之前, 必须停止定时器工作, 否则, 当系统时钟启动时定时器可能产生错误。	pp. 85, 118
				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位设置TI000引脚有效沿。	pp. 85, 114
				在TM00和CR00相等时进入的清零&启动模式中, 当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后, OVF00标志位置1。	p. 85
				即便在TM00溢出后而下一个计数时钟计数前(TM00变为0001H前)做清除OVF00标志操作, OVF00标志还会被重新置1, 清除操作无效。	pp. 85, 115
				捕捉操作在计数时钟的下降沿执行, 中断请求(INTTM0n0)在计数时钟的上升沿产生。	pp. 85, 116

章节	分级	功能	功能细节	注意事项	页码
第六章	软件	16位定时器/事件计数器00	CRC00: 捕捉/比较控制寄存器00	设置CRC00前, 定时器必须停止操作。	pp. 86, 114
				当通过16位定时器模式控制寄存器00(TMC00)选择TM00 和CR000相等进入的清零 & 启动模式时, CR000 就不能被指定为捕捉寄存器。	pp. 86, 113
				为了确保捕捉操作的稳定, 被捕捉的信号宽度要大于2个计数时钟(由预分频器模式寄存器 00(PRM00)选择)的长度才能触发捕捉(参见图 6-18)。	pp. 86, 116
	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, 定时器应停止工作。	pp. 87, 114	<input type="checkbox"/>
			读LVS00和LVR00时, 读出的数据为0。	pp. 87, 114	<input type="checkbox"/>
			数据被设置后, OSPT00 自动清零, 读出的内容是0。	pp. 87, 114	<input type="checkbox"/>
			除了单脉冲输出模式以外, 不要将OSPT00设置为1。	pp. 87, 114	<input type="checkbox"/>
			当连续设置OSPT00位时, 需要两个或更多计数时钟周期(由预分频器模式寄存器00(PRM00)选择)的写入间隔。	pp. 87, 114	<input type="checkbox"/>
			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。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. 88	<input type="checkbox"/>
	硬件	PRM00: 预分频模式寄存器00	停止定时器工作后, 向PRM00写入数据。	pp. 88, 114	<input type="checkbox"/>
			如果TI000 引脚的有效沿被用于计数时钟, 就不能设置清零启动模式和TI000引脚有效沿的捕捉。	pp. 88, 116	<input type="checkbox"/>
			在下列情况下, 注意 TI0n0 引脚有效沿被检测的警告。 <1> 系统复位后, 如果一个高电平输入TI0n0引脚, 16位定时器计数器 00 (TM00) 允许操作。 →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> 当TI0n0为高电平, TM00停止工作时, TI0n0引脚如输入一个低电平则TM00开始工作。 →如果TI0n0的下降沿或上升沿和下降沿作为有效沿, 那么 TM00操作使能后, 下降沿会立即检测到。 <3> 当TI0n0为低电平, TM00停止工作时, TI0n0引脚如输入一个高电平则TM00开始工作。 →如果TI0n0的上升沿或上升沿和下降沿作为有效沿, 那么 TM00操作使能后, 上升沿会立即检测到。	pp. 89, 118	<input type="checkbox"/>
			当TI000的有效沿用做计数时钟, 和当TI000用于捕捉触发器时, 用于消除噪声的采样时钟是不同的, 对于前者, 由f _{XP} 做计数时钟, 对于后者, 通过PRM00选择的计数时钟。直到采样到有效沿并且连续两次检测到有效电平, 捕捉操作才会完成, 因此可消除小脉冲宽度的噪声。	pp. 89, 118	<input type="checkbox"/>

章节	功能	功能细节	注意事项	页码
第六章	16位定时器时钟/事件计数器00	PRM00: 预分频模式寄存器00	当P21端口作为有效沿的输入引脚(TI010)使用时, 它不能同时作为定时器输出引脚(TO00)使用。当P21端口作为定时器输出引脚(TO00)使用时, 它不能同时作为有效沿的输入引脚(TI010)使用。	pp. 89, 118
			间隔定时	当TM00工作时, 改变CR000寄存器的设置可能导致错误。要改变设置, 请参考6.5 16位定时器/事件计数器00(17)在定时器操作过程中改变比较寄存器的相关注意事项。
		外部事件计数器	当读取外部事件计数器的计数值时, TM00即为其值。	pp. 94, 118
		测量脉冲宽度	如要使用两个捕捉寄存器, 设置TI000和TI010引脚。	pp. 95, 116
			例子中可测量脉冲宽度达到定时器计数器的1个周期。	pp. 95, 97, 99, 101
		方波输出	如在TM00操作中要改变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 (17)在定时器操作过程中改变比较寄存器的相关注意事项。	p. 103
		PPG输出	如在TM00操作中要改变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 (17)在定时器操作过程中改变比较寄存器的相关注意事项。	p. 105
			以下为CR000 和 CR010的取值范围: $0000H < CR010 < CR000 \leq FFFFH$	pp. 106, 118
		PPG输出	由PPG输出(CR00n设置值+ 1)产生的脉冲周期的占空比为(CR01n设置值 + 1)/(CR00n设置值 + 1)。	pp. 106, 118
			单脉冲输出: 软件触发	当单脉冲输出时, 不要设置OSPT00 位。在当前单脉冲输出完成后才可再次输出单脉冲。
		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 及其复用端口引脚的电平。 由于在这种情况下外部触发也有效, 所以可使用TI000 或其复用端口引脚对定时器清零&启动。这样会输出非预期的时序。		pp. 108, 114
		CR000 和 CR010 寄存器不能置0000H。		pp. 109, 111
		一旦为TMC003 和TMC002位赋值(00除外, 操作停止模式), 16位定时器计数器00 开始计数。		pp. 110, 114
		单脉冲输出: 外部触发	当单脉冲正在输出时不要再次再输入外部触发。在当前单脉冲输出完成后才可再次输出单脉冲。	pp. 110, 115
			CR000 和CR010寄存器不能置 0000H 。	pp. 111, 117
			一旦为TMC003和TMC002位赋值(00除外, 操作停止模式), 16位定时器计数器00 开始计数。	pp. 112, 113
		定时器启动误差	定时器启动后在信号匹配产生之前可能发生将近一个时钟的误差, 这是由于16位定时器计数器00(TM00)的启动与计数时钟不同步。	p. 113
		单脉冲输出	一般只有在自由运行模式或清零&启动模式下, TI000引脚有效沿输入才能有单脉冲输出。因为在清零&启动模式下, TM00和CR000相等时不发生溢出, 所以不可能输出单脉冲。	p. 114

章节	分节	功能	功能细节	注意事项	页码
第六章	软件	16位定时器时钟/事件计数器00	捕捉操作	当CRC001为1时，如果TI000引脚的上升沿和下降沿同时作为输入有效沿，则捕捉操作无效。	p. 116 <input type="checkbox"/>
				当CRC001值为1，如果TI010引脚有效沿被检测，TM00的计数值不能被捕捉到CR000寄存器中，但是TI010引脚的输入可以用作外部中断源，因为可由外部有效信号输入产生INTTM000中断。	p. 116 <input type="checkbox"/>
			在定时器工作时改变比较寄存器的值	16位定时器捕捉/比较寄存器On0(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位定时器捕捉/比较寄存器On0(CR0n0)的值相等的时序附近修改CR0n0的值时，改变CR0n0时序和相等时序可能会冲突，这种情况下，操作不能得到保证。在定时器计数时改变CR0n0的值时，INTTM000中断服务程序有下面操作。	p. 117 <input type="checkbox"/>
				如果在定时器计数未执行上面的过程<1>期间改变CR010时，那么CR010中的值将被重新写入两次或更多，每次写入将引起TO00引脚电平的反转。	p. 117 <input type="checkbox"/>
			外部事件计数器	在2个有效沿检测后启动计数时序。	p. 118 <input type="checkbox"/>
			外部时钟限制	当使用TI000引脚输入脉冲作为计数时钟(外部触发器)，确保输入满足AC特性的的脉冲宽度。关于AC特性，参照第十八章 电器特性。	p. 119 <input type="checkbox"/>
				当外部波形输入到16位定时器/事件计数器00，通过噪音限制电路和时序发生的错误取样成为器件的有效沿。	p. 119 <input type="checkbox"/>
第七章	软件	8位定时器H1	CMP01: 8位定时器H比较寄存器01	在定时器计数期间不能修改CMP01。	p. 122 <input type="checkbox"/>
			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(即使前后设置的值相同也必须对CMP11重新设置)。	p. 122 <input type="checkbox"/>
			TMHMD18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. 124 <input type="checkbox"/>
				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(即使前后设置的值相同也必须对CMP11重新设置)。	p. 124 <input type="checkbox"/>
	硬件	PWM输出		在PWM输出模式下，在重写寄存器之后，需要3个操作周期(使用TMHMD1寄存器的CKS12到CKS10位选择的信号)来传送CMP11寄存器的值。	p. 130 <input type="checkbox"/>
				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(即使设置的是相同的值，也必须再次赋值)。	p. 130 <input type="checkbox"/>
				CMP11的取值(M)和CMP01的取值(N)应在如下范围内。 $00H \leq \text{CMP11}(M) < \text{CMP01}(N) \leq \text{FFH}$	p. 130 <input type="checkbox"/>

章节	分级	功能	功能细节	注意事项	页码	
第八章	软件	看门狗定时器	WDTE: 看门狗定时器使能寄存器	第7, 6和5 ~ 0位分别设置为0, 1和1。不要设置其它值。	p. 138 <input type="checkbox"/>	
				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。在第一次写入时, 如果“1”和“X”被分别写入到 WDSC4和WDSC3中, 并且看门狗定时器被停止, 这样即使出现以下情况, 也不会产生内部复位信号。 • 第二次写入WDTM • 使用1位存储器操作指令写 WDTE • 对WDTE写入不同于“ACH”的值	p. 139 <input type="checkbox"/>	
				不能用1位存储器操作指令对WDTM进行操作。	p. 139 <input type="checkbox"/>	
				当使用自写入方式对flash存储器进行操作时, 为看门狗设置足够的溢出时间(例如单字节写入: 最小200 μ s, 一个块删除: 最小10 ms)。	p. 139 <input type="checkbox"/>	
				如果一个不是 ACH的值被写入 WDTE, 将产生一个内部复位信号。	p. 139 <input type="checkbox"/>	
				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. 139 <input type="checkbox"/>	
	硬件			当由选项字节选择为“低速 Ring-OSC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. 140 <input type="checkbox"/>
				当由选项字节选择为“低速 Ring-OSC可通过软件停止”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, 计数器并没有被清零, 而是保持原值。	p. 142 <input type="checkbox"/>
	第九章	软件	A/D 转换器	采样时间和A/D转换时间	以上采样时间和转换时间没有包括时钟频率误差。选择采样时间和转换时间例如注2和3是很好的, 要将时钟频率误差考虑在内(当使用高速内部振荡器时最大误差为 $\pm 5\%$)。	p. 147 <input type="checkbox"/>
				框图	在78K0S/KU1+ 中, V _{SS} 复用为A/D转换器的地。要确保V _{SS} 连接到一个稳定的GND(= 0V)。 在78K0S/KU1+ 中, V _{DD} 复用为A/D转换器的参考电压输入。当使用A/D转换器时, V _{DD} 使用稳定电源(2.7 ~ 5.5 V)。	p. 148 <input type="checkbox"/>
硬件			ADM: A/D转换器模式寄存器	以上采样时间和转换时间没有包括时钟频率误差。选择采样时间和转换时间要将时钟频率误差考虑在内(当使用高速内部振荡器时最大误差为 $\pm 5\%$)。	p. 152 <input type="checkbox"/>	
				当A/D转换结束(ADCS = 0)时, 对ADM中除了ADCS的位进行操作后, 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, A/D转换重新开始。	p. 152 <input type="checkbox"/>	
				在重写FR0至FR2时, A/D 转换必须停止 (ADCS = 0)。	p. 152 <input type="checkbox"/>	
				一定要把第6位、第2位和第1位清0。	p. 152 <input type="checkbox"/>	

章节	功能	功能细节	注意事项	页码
第九章	A/D 转换器	ADS: 模拟输入通道选择寄存器	ADS 的第2位到第7位必须清0。	p. 153 <input type="checkbox"/>
		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。在对ADM和ADS执行写操作之前, 要在转换完成后读取转换结果。在其他时间读取可能读到错误的转换结果。	p. 153 <input type="checkbox"/>
		PMC2: 端口模式控制寄存器 2	当PMC20 ~ PMC23被设置为1时, P20/ANI0/TI000/TOH1, P21/ANI1/TI010/TO00/INTP0, P22/ANI2和P23/ANI3引脚只能被用作A/D转换器功能。 确保设置上拉电阻选择寄存器(PU20 ~ PU23)为0, 使引脚设置为A/D转换器模式。	p. 154 <input type="checkbox"/>
		A/D 转换器操作	确保 <1> ~ <4> 的时间为大于等于1 μ s。	p. 159 <input type="checkbox"/>
			如果 <1> 和 <2> 的顺序颠倒也没有关系。	p. 159 <input type="checkbox"/>
			<1> 可以被忽略。 但在这种情况下中当步骤 <4> 执行后忽略第一个转换结果。	p. 159 <input type="checkbox"/>
			步骤 <5> ~ <8> 所经历的时间与ADM的第5 ~ 3位(FR2 ~ FR0)设置的转换时间不同。步骤<7> ~ <8> 所经历的时间即为转换时间, 由FR2 ~ FR0设置。	p. 159 <input type="checkbox"/>
	STOP 模式下的工作电流	在STOP模式下, A/D转换停止操作。此时, 将A/D转换模式寄存器(ADM)的位7(ADCS)和位0(ADCE)清0, 可以降低操作电流。	p. 162 <input type="checkbox"/>	
	ANI0 ~ ANI3的输入范围	观察ANI0 ~ 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V _{DD} 或更高, 或者是V _{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。	p. 162 <input type="checkbox"/>	
	冲突操作	转换结束后, 指令对A/D转换结果寄存器(ADCR, ADCRH)的写操作和ADCR, ADCRH的读操作同时进行, 就为冲突操作。 ADCR, ADCRH读操作优先。在读操作执行完后, 新的转换结果将写入到ADCR、ADCRH。	p. 162 <input type="checkbox"/>	
		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, 或者对模拟输入通道选择寄存器(ADS)写操作同时进行, 就为冲突操作。 ADM或ADS写操作优先。如果ADCR, ADCRH的写操作不完成, 就不会产生转换结束中断信号(INTAD)。	p. 162 <input type="checkbox"/>	
	噪声解决办法	为了保持10位分辨率, 必须注意输入到V _{DD} 引脚和ANI0 ~ ANI3引脚的噪声。 <1>在低阻抗和高频反馈的电源上连接一个电容。 <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9-19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. 162 <input type="checkbox"/>	
	软件	硬件	软件	硬件

章节	分级	功能	功能细节	注意事项	页码
第九章	软件	A/D 转换器	ANI0/P20 ~ ANI3/P23	模拟输入引脚(ANI0 ~ ANI3) 也可用作输入端口引脚 (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, 否则会降低转换效果。	p. 163 <input type="checkbox"/>
				如果正在进行A/D转换的引脚相邻引脚有数字脉冲, 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. 163 <input type="checkbox"/>
			ANI0 ~ ANI3引脚的输入阻抗	在A/D转换器中, 采样期间内部采样电容充电, 采样时间接近转换时间的1/6。由于在采样期间有漏电流和电容充电的电流, 因此输入阻抗在采样期间或其他时候都有波动。 如果在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, 或者在引脚连接0.01μF ~ 0.1μF的电容(见 图 9-19)。	p. 163 <input type="checkbox"/>
			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转换期间有模拟输入引脚改变, 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。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转换结束要重新开始转换时, 要在重新开始转换前先将ADIF清0。	p. 163 <input type="checkbox"/>
			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 μs 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。可采取措施, 如将A/D转换结束中断请求(INTAD)挂起, 并删除第1次转换结果。	p. 163 <input type="checkbox"/>
			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进行写操作前并且转换结束后, 读取转换结果。若采用和上述不同的时序, 可能会读取到不正确的转换结果。	p. 163 <input type="checkbox"/>
第十章	软件	中断功能	IF0: 中断请求标志寄存器, MK0: 中断屏蔽标志寄存器	因为P21和IP32有一个复用功能就是作为外部中断输入, 所以当指定端口输出模式使输出电平发生变化, 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. 168 ~ 169 <input type="checkbox"/>
			INTM0: 外部中断模式寄存器0	寄存器的第0, 1, 6和7位清零。	p. 169 <input type="checkbox"/>
			在设置INTM0寄存器前, 必须设置相应的中断屏蔽标志(×MK× = 1)以禁止中断。INTM0寄存器设置完成后, 首先对中断请求标志(×IF× = 0)进行清零, 然后再对中断屏蔽标志 (×MK× = 0)进行清零以允许中断。	p. 170 <input type="checkbox"/>	

章节	分级	功能	功能细节	注意事项	页码
第十章	软件	中断功能	在中断请求保持期间	当中断请求标志寄存器(IFO)或中断屏蔽标志寄存器(MK0)被访问期间, 中断请求等待。	p. 172 <input type="checkbox"/>
			在中断请求期间	低优先级中断依然可以被响应。	p. 173 <input type="checkbox"/>
第十一章	软件	待机功能	-	仅当选项字节选择低速内部振荡器“可被软件停止”时, LSRSTOP 设置才有效。	p. 175 <input type="checkbox"/>
			STOP模式	当改变到STOP模式时, 一定要在执行STOP指令(除了以低速Ring-OSC振荡时钟工作的外部硬件之外)之前停止外部硬件操作。	p. 176 <input type="checkbox"/>
	硬件		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, 然后执行HALT或STOP指令。	p. 176 <input type="checkbox"/>
			STOP模式	如果在置STOP模式之前低速Ring-OSC振荡器正在工作, 则在STOP模式中不能停止低速Ring-OSC时钟(参见表11-1)。	p. 176 <input type="checkbox"/>
	软件		OSTS: 振荡稳定时间选择寄存器	如要设置并退出STOP模式, 按以下方式设置振荡稳定时间。 预期振荡稳定时间 ≤ OSTS设置的振荡稳定时间	p. 177 <input type="checkbox"/>
				无论是复位后还是由中断产生退出STOP模式, 退出STOP模式后的等待时间不包括从退出STOP模式到时钟振荡启动这段时间(即下图“a”所示的部分)。	p. 177 <input type="checkbox"/>
	由选项字节选择上电或复位后经过的振荡稳定时间。详情可参见第十五章选项字节。			p. 177 <input type="checkbox"/>	
	软件		HALT模式的设定和工作状态	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. 178 <input type="checkbox"/>
STOP模式的设定和工作状态		由于中断请求信号用于释放待机模式, 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, 然后操作停止34 μs(典型值)(在经过由振荡稳定时间选择寄存器(OSTS)设置的等待时间后, 晶体/陶瓷振荡器启动)。	p. 181 <input type="checkbox"/>		

章节	分级	功能	功能细节	注意事项	页码
第十二章	硬件	复位功能	-	对于外部复位, 输入RESET引脚的低电平的时间应为2 μ s或更长。	p. 185 <input type="checkbox"/>
				复位输入期间, 系统时钟和低速内部振荡时钟停止振荡。	p. 185 <input type="checkbox"/>
				RESET引脚只能被复用输入端口引脚(P34), 如果在由POC电路释放复位后且在再次参考选项字节之前低电平输入到RESET引脚, 78K0S/KU1+ 复位。复位状态保持直到高电平输入到RESET引脚。	p. 185 <input type="checkbox"/>
				LVI电路的内部复位信号不会使LVI电路复位。	p. 186 <input type="checkbox"/>
	软件	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. 188 <input type="checkbox"/>	
第十三章	软件	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. 192 <input type="checkbox"/>
	硬件	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, 则将复位标志寄存器(RESF)清零。 POC电路的检测电压(V _{POC})值范围为2.1V \pm 0.1V, 故正常工作电压范围为 2.2 ~ 5.5 V。	p. 193 <input type="checkbox"/> p. 193 <input type="checkbox"/>
			上电清零电路的注意事项	对于一个电源电压在某段时间内在POC检测电压(V _{POC})附近波动的系统, 电源电压在一段时间内会产生波动, 系统可能会反复进行复位和释放复位。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所经历的时间。	p. 195 <input type="checkbox"/>
第十四章	软件	低电压检测器	LVIM: 低电压检测寄存器	可在执行完以下任一过程后终止LVI。 • 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. 198 <input type="checkbox"/>
				第2 ~ 6位必须清零。	p. 198 <input type="checkbox"/>
			LVIS: 低电压检测电平选择寄存器	第4 ~ 7位必须清零。	p. 199 <input type="checkbox"/>
				如果在LVI工作时, 写入了一个不同于上表的值, 则写入时这个值将变的不确定, 因此在写入前必须停止LVI(LVIM寄存器的第7位(LVION)= 0)。	p. 199 <input type="checkbox"/>
			用作复位	必须执行过程 <1>, 当LVIMK = 0时, 在过程 <3> 之后可能立即会产生中断。	p. 200 <input type="checkbox"/>
				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。	p. 200 <input type="checkbox"/>
低电压检测器的注意事项	在一个电源电压(V _{DD})在某段时间内在LVI检测电压附近波动的系统中, 以下操作决定于如何使用低电压检测器。 <1>用于复位 系统可能会反复进行复位和释放复位。 在这种情况下, 可采用下面的方法(1)任意设置从复位释放到微控制器的启动所经历的时间。 <2>用于中断 可能会频繁产生中断请求。可采用方法(2)。	p. 204 <input type="checkbox"/>			

章节	分级	功能	功能细节	注意事项	页码
第十五章	硬件	选项字节	上电时或复位释放后振荡稳定时间	仅当选晶体/陶瓷振荡时钟作为系统时钟源时，这项设定才有效。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时，无需等待时间。	p. 208 <input type="checkbox"/>
			复位引脚的控制	因为选项字节是在复位释放后才被读取的，因此，如果在读取选项字节以前，向复位引脚输入一个低电平，那么复位状态不会释放。当设置RMCE为0时，连接上拉电阻。	p. 208 <input type="checkbox"/>
			系统时钟源的选择	因为X1和X2引脚也作为P23/ANI3和P22/ANI2引脚使用，X1和X2引脚使用的条件不同，取决于系统时钟源的选择。 (1) 选择晶体/陶瓷振荡时钟 因为X1和X2引脚作为时钟输入引脚，所以它们不能被用作I/O端口引脚或A/D转换器的模拟输入引脚。 (2) 选择外部时钟输入 因为X1引脚作为外部时钟输入引脚，P23/ANI3不能被用作I/O端口引脚或A/D转换器的模拟输入引脚。 (3) 选择高速内部振荡器时钟 P23/ANI3和P22/ANI2引脚能够用作I/O端口引脚或A/D转换器的模拟输入引脚。	p. 208 <input type="checkbox"/>
			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)的计数器时钟就会固定为低速内部振荡器时钟。	p. 209 <input type="checkbox"/>
				如果选择低速内部振荡器时钟可以停止，WDT的计数器时钟的提供在HALT/STOP模式下被停止，不管低速内部振荡器模式寄存器(LSRCM)的第0位(LSRSTOP)的设置是什么。同样地，当选择除低速内部振荡器时钟之外的任何一个时钟作为WDT的计数器时钟时，时钟提供也被停止。但是，在低速内部振荡器工作时(LSRSTOP = 0)，即使在STOP模式下，时钟可提供给8位定时器H1使用。	p. 209 <input type="checkbox"/>
当复位引脚被仅用于输入端口引脚时(P34)的注意事项	需要知道以下情况：当重新擦除/写入(通过使用一个专用Flash存储器编程器进行on-board编程)一个已经被写入的设备时，而这个设备在选项字节功能设置下复位引脚仅用于输入端口引脚(P34)。 在为目标系统上电以前，先连接一个专用Flash存储器编程器，并且先将其上电。如果先为目标系统上电，则不能切换到Flash存储器编程模式。	p. 209 <input type="checkbox"/>			
第十六章	软件	FLASH存储器	PG-FP4 GUI软件设置值示例	以上值为推荐值。由于环境不同，这些值可能改变，所以使用时请先估计一个值。	p. 216 <input type="checkbox"/>
			安全设定	在一次性擦除(芯片擦除)的安全设置设定之后，擦除指令不能执行。另外，即使执行写入命令，与已写入Flash存储器不同的数据是不能被写入的，因为擦除命令已被禁止。	p. 219 <input type="checkbox"/>
				当下一次设定编程模式时，安全设置有效。	p. 219 <input type="checkbox"/>
			自编程功能	在执行自写入以前，自写入处理程序必须先被包含在用户程序中。	p. 222 <input type="checkbox"/>
				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16-11。	p. 225 <input type="checkbox"/>
				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，应在从普通模式切换到自编程模式之前以一定顺序终止中断服务(将MK0设定为FFH，并执行DI指令)。	p. 225 <input type="checkbox"/>
当执行自编程命令时，不须使用RAM。	p. 225 <input type="checkbox"/>				
当FLASH存储器正在数据写入/擦除过程中，如果电源电压下降或有复位信号输入，则此次写入/擦除操作不一定成功。	p. 225 <input type="checkbox"/>				

章节	分级	功能	功能细节	注意事项	页码
第十六章	软件	FLASH存储器	自编程功能	在执行自写入以前, 自写入处理程序必须先被包含在用户程序中。	p. 220 <input type="checkbox"/>
				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 16-11。	p. 223 <input type="checkbox"/>
				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, 应在从普通模式切换到自编程模式之前以一定顺序终止中断服务(将MKO设定为FFH, 并执行DI指令)。	p. 223 <input type="checkbox"/>
				当执行自编程命令时, 不须使用RAM。	p. 223 <input type="checkbox"/>
				当FLASH存储器正在数据写入/擦除过程中, 如果电源电压下降或有复位信号输入, 则此次写入/擦除操作不一定成功。	p. 223 <input type="checkbox"/>
				在block擦除中空白数据的值设为FFH。	p. 223 <input type="checkbox"/>
				预先设置CPU时钟, 使在自编程中使CPU时钟大于等于1 MHz。	p. 223 <input type="checkbox"/>
				执行指定序列的命令设定自编程模式后, 立即执行NOP和HALT指令, 以执行自编程。10 μs(最大值)+ 2 CPU时钟(f _{CPU})后HALT指令被自动释放。	p. 223 <input type="checkbox"/>
				如果振荡器时钟/外部时钟被选择为系统时钟, 执行指定序列的命令设定自编程模式后, 立即执行NOP和HALT指令, 在HATL状态释放后等待8 μs, 然后执行自编程。	p. 223 <input type="checkbox"/>
				通过使用1位存储器操作指令验证FPRERR。	p. 223 <input type="checkbox"/>
				自编程模式下和HALT模式下引脚的状态是相同的。	p. 223 <input type="checkbox"/>
				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. 223 <input type="checkbox"/>
				在执行自编程命令前, 确保对Flash地址指针H(FLAPH)的第4 ~ 7位和Flash地址指针H比较寄存器(FLAPHC)清零。如果这些位为1时执行自编程, 设备将会故障。	p. 223 <input type="checkbox"/>
				在设置为自编程模式和普通模式以前, 立即将FLCMD寄存器清零(00H)。	p. 223 <input type="checkbox"/>
		FLPMC: Flash编程模式控制寄存器	注意在设定自编程模式的情况下, 可参考16.8.2自编程功能的注意事项。	p. 224 <input type="checkbox"/>	
			预先设置CPU时钟, 使在自编程中使CPU时钟大于等于1 MHz。	p. 224 <input type="checkbox"/>	
			执行指定序列的命令设定自编程模式后, 立即执行NOP和HALT指令, 以执行自编程。10 μs(最大值)+ 2 CPU时钟(f _{CPU})后HALT指令被自动释放。	p. 224 <input type="checkbox"/>	
			如果振荡器时钟/外部时钟被选择为系统时钟, 执行指定序列的命令设定自编程模式后, 立即执行NOP和HALT指令, 在HATL状态释放后等待8 μs, 然后执行自编程。	p. 224 <input type="checkbox"/>	
			在设置为自编程模式和普通模式以前, 立即将FLCMD寄存器清零(00H)。	p. 224 <input type="checkbox"/>	

章节	分级	功能	功能细节	注意事项	页码	
第十六章	软件	FLASH存储器	PFCMD: Flash保护命令寄存器	在自编程模式下, 不能执行中断服务。在执行指定序列指令设定自编程模式以前, 执行指定序列指令设定工作模式为普通模式以后的两点之间, 停止中断服务(通过在MK0= FFH时, 执行DI指令)。	p. 225	<input type="checkbox"/>
			PFS: Flash状态寄存器	使用1位存储器操作指令验证FPRERR。	p. 225	<input type="checkbox"/>
			FLAPH, FLAPL: Flash地址指针H和L	确保在执行自编程命令以前对Flash地址指针H(FLAPH)的第4 ~ 7位和Flash地址指针H比较寄存器(FLAPHC)清零。如果当执行自编程命令时这些位的值为1。	p. 228	<input type="checkbox"/>
			FLAPH, FLAPL: Flash地址指针H和L	在自编程命令执行以前, 确保FLASH地址指针H(FLAPH)的第4 ~ 7位和FLASH地址指针H比较寄存器(FLAPHC)清零。如果当执行自编程命令时这些位的值为1。	p. 228	<input type="checkbox"/>
				当进行block擦除、校验、空白检测时, 将block的序号(与FLAPH的值相同)赋值给FLAPHC。	p. 228	<input type="checkbox"/>
				当block擦除完成时FLAPLC要清零, 当block空白检测完成后FLAPLC要设为FFH。	p. 228	<input type="checkbox"/>
			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和写入操作发生。	pp. 230, 231, 233, 234	<input type="checkbox"/>
			式切换到普通模式			
字节写入	如果写入失败, 应先擦除block内容并再次写入。	p. 242	<input type="checkbox"/>			
第十八章	硬件	电气特性	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, 因为, 必须保证产品在不超过最大额定值的条件下使用。	p. 276	<input type="checkbox"/>
			X1振荡器特性	在使用X1振荡器时, 上表中的虚线框内部分的连线应遵从如下的连接方式, 避免线间电容产生不利影响。 <ul style="list-style-type: none"> • 连接线越短越好。 • 连接线不与其他信号线交叉。 • 如果信号线流经的电流变化较大, 则不要在其周围连线。 • 要保持振荡器电容器的接地点电压与V_{SS}相同。 • 避免大电流从电容到地的连线上流过。 • 不要从振荡器获取信号。 	p. 277	<input type="checkbox"/>
			A/D转换器	当模拟输入引脚被用于复用I/O端口时, 或当A/D转换时如果未用作A/D转换的端口的电平被改变时, 转换精度会降低。	p. 284	<input type="checkbox"/>
第二十章	硬件	推荐的焊接条件	无铅产品	产品尾缀 -A 是无铅产品。	p. 289	<input type="checkbox"/>
			-	如不使用以下的焊接和装贴条件请与NECEL公司销售代理联系。	p. 289	<input type="checkbox"/>
				不要将不同的焊接方法一起使用(局部加热法除外)。	p. 289	<input type="checkbox"/>

详细信息请联系:

(中国区)

网址:

<http://www.cn.necel.com/>

<http://www.necel.com/>

[北京]

日电电子(中国)有限公司
中国北京市海淀区知春路 27 号
量子芯座 7, 8, 9, 15 层
电话: (+86) 10-8235-1155
传真: (+86) 10-8235-7679

[上海]

日电电子(中国)有限公司上海分公司
中国上海市浦东新区银城中路 200 号
中银大厦 2409-2412 和 2509-2510 室
电话: (+86) 21-5888-5400
传真: (+86) 21-5888-5230

上海恩益禧电子国际贸易有限公司
中国上海市浦东新区银城中路 200 号
中银大厦 2511-2512 室
电话: (+86) 21-5888-5400
传真: (+86) 21-5888-5230

[深圳]

日电电子(中国)有限公司深圳分公司
深圳市福田区益田路卓越时代广场大厦 39 楼
3901, 3902, 3909 室
电话: (+86) 755-8282-9800
传真: (+86) 755-8282-9899

[香港]

香港日电电子有限公司
香港九龙旺角太子道西 193 号新世纪广场
第 2 座 16 楼 1601-1613 室
电话: (+852) 2886-9318
传真: (+852) 2886-9022
2886-9044