

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

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

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

April 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

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

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

## Notice

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

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

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

## 78K0/LC3

### 8 位单片微控制器

---

**$\mu$ PD78F0400**

**$\mu$ PD78F0410**

**$\mu$ PD78F0401**

**$\mu$ PD78F0411**

**$\mu$ PD78F0402**

**$\mu$ PD78F0412**

**$\mu$ PD78F0403**

**$\mu$ PD78F0413**

78K0/LC3具有片上调试功能。

大规模生产时不要使用本产品，因为在使用片上调试功能后，从 flash 存储器可重写的次数受到限制这一点来看，它的可靠性不能保证。NEC Electronics 不接受对于本产品的任何投诉。

[备忘录]

**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 $V_{IL}$ （最大值）和 $V_{IH}$ （最小值）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 $V_{IL}$ （最大值）过渡到 $V_{IH}$ （最小值）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 $V_{DD}$ 或GND。如果有可能尽量定义为输出引脚。对未使用引脚的处理因设备而异，必须遵循与设备相关的规定和说明。

**③ ESD防护措施**

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

**④ 初始化之前的状态**

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

**⑤ 电源开关顺序**

在一个设备的内部操作和外部接口使用不同的电源的情况下，按照规定，应先在接通内部电源之后再接通外部电源。当关闭电源时，按照规定，先关闭外部电源再关闭内部电源。如果电源开关顺序颠倒，可能会导致设备的内部组件过电压，产生异常电流，从而引起内部组件的误操作和性能的退化。

对于每个设备电源的正确开关顺序必须依据设备的规范说明分别进行判断。

**⑥ 电源关闭状态下的输入信号**

不要向没有加电的设备输入信号或提供I/O上拉电源。因为输入信号或提供I/O上拉电源将引起电流注入，从而引起设备的误操作，并产生异常电流，从而使内部组件退化。

每个设备电源关闭时的信号输入必须依据设备的规范说明分别进行判断。

EEPROM 是 NEC 电子公司的注册商标。

SuperFlash 是 Silicon Storage Technology, Inc 的注册商标,在包括美国日本的多个国家注册。

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

- 本文档信息于 2007 年 7 月开始使用。将来可能未经预先通知而更改。在实际进行生产设计时, 请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可, 禁止复制或转载本文件中的内容。本文件所登载内容的错误, 本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的, 对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或其他人所有的专利、版权以及其它知识产权作出任何明示或默示的许可及授权。
- 本文件中的电路、软件以及相关信息仅用以说明半导体产品的运作和应用实例。用户如在设备设计中应用本文件中的电路、软件以及相关信息, 应自行负责。对于用户或其他人因使用了上述电路、软件以及相关信息而引起的任何损失, 本公司概不负责。
- 虽然本公司致力于提高半导体产品的质量及可靠性, 但用户应同意并知晓, 我们仍然无法完全消除出现产品缺陷的可能。为了最大限度地减少因本公司半导体产品故障而引起的对人身、财产造成损害(包括死亡)的危险, 用户务必在其设计中采用必要的安全措施, 如冗余度、防火和防故障等安全设计。
- 本公司产品质量分为:

“标准等级”、“专业等级”以及“特殊等级”三种质量等级。

“特殊等级”仅适用于为特定用途而根据用户指定的质量保证程序所开发的日电电子产品。另外, 各种日电电子产品的推荐用途取决于其质量等级, 详见如下。用户在选用本公司的产品时, 请事先确认产品的质量等级。

“标准等级”: 计算机, 办公自动化设备, 通信设备, 测试和测量设备, 音频·视频设备, 家电, 加工机械以及产业用机器人。

“专业等级”: 运输设备(汽车、火车、船舶等), 交通信号控制设备, 防灾装置, 防止犯罪装置, 各种安全装置以及医疗设备(不包括专门为维持生命而设计的设备)。

“特殊等级”: 航空器械, 宇航设备, 海底中继设备, 原子能控制系统, 为了维持生命的医疗设备、用于维持生命的装置或系统等。

除在本公司半导体产品的数据表或数据手册等资料中另有特别规定以外, 本公司半导体产品的质量等级均为“标准等级”。如果用户希望在本公司设计意图以外使用本公司半导体产品, 务必事先与本公司销售代表联系以确认本公司是否同意为该项应用提供支持。

(注)

- (1) 本声明中的“本公司”是指日本电气电子株式会社(NEC Electronics Corporation)及其控股公司。
- (2) 本声明中的“本公司产品”是指所有由日本电气电子株式会社或为日本电气电子株式会社(定义如上)开发或制造的产品。

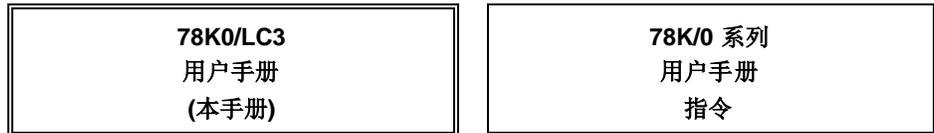
## 引言

**读者对象** 本手册适用于那些希望了解78K0/LC3 功能，并设计开发应用系统和程序的工程师。目标产品如下。

78K0/LC3:  $\mu$ PD78F0400, 78F0401, 78F0402, 78F0403  
 $\mu$ PD78F0410, 78F0411, 78F0412, 78F0413

**目的** 本手册用于帮助用户了解下面**组件**中描述的功能。

**组件** 78K0/LC3 手册主要分为两个部分：手册和指令（与 78K/0 系列通用）。



- |  |   |
|--|---|
| <ul style="list-style-type: none"><li>• 引脚功能</li><li>• 内部模块功能</li><li>• 中断</li><li>• 其它内置外设功能</li><li>• 电气特性</li></ul> | <ul style="list-style-type: none"><li>• CPU 功能</li><li>• 指令集</li><li>• 指令扩展</li></ul> |
|--|---|

**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。

- 如果读者要了解产品功能：  
→ 请按**目录**顺序阅读本手册。
- 如何解释寄存器格式：  
→ 括号中的位名在 RA78K0 中被定义为保留字，并且在 CC78K0 中直接通过 # pragma sfr 定义为一个 sfr 变量。
- 如果读者希望了解 78K/0 系列指令的详细信息：  
→ 可参阅 **78K/0 系列指令用户手册 (U12326E)**。

**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:	$\overline{\text{xxx}}$ (在引脚和信号名称上划一条线)
注:	文中用 <b>注</b> 标注的相关术语的脚注
<b>注意事项:</b>	需要特别关注的信息
<b>备注:</b>	补充信息
数的表示法:	二进制      ... xxxx 或 xxxxB 十进制      ... xxxxx 十六进制    ... xxxxH

## 相关文档

本手册中指出的相关文档可能包括了初级的版本，但未注明。

## 设备文档

文档名称	文档编号
78K0/LC3 用户手册	本手册
78K/0 系列指令用户手册	U12326E

## 与 Flash 存储器编程相关文档 (用户手册)

文档名称	文档编号
PG-FP4 Flash 存储器编程器用户手册	U15260E
PG-FPL3 Flash 存储器编程器用户手册	U17454E

## 其他文档

文档名称	文档编号
半导体选择指南—产品和封装—	X13769X
半导体设备安装手册	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 保护指南	C11892E

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

注意事项 对以上列出的相关文档所做修改恕不另行通知，在设计时请使用每个文档的最新版本。

# 目录

第一章 概述.....	14
1.1 特性.....	14
1.2 应用.....	15
1.3 订购信息.....	15
1.4 引脚配置(俯视图).....	16
1.5 78K0/Lx3 微控制器系列产品.....	19
1.6 框图.....	23
1.7 功能概述.....	24
第二章 引脚功能.....	27
2.1 引脚功能列表.....	27
2.2 引脚功能描述.....	31
2.2.1 P12,P13 (端口 1).....	31
2.2.2 P14 ~ P25 (端口 2).....	31
2.2.3 P31 ~ P34 (端口 3).....	32
2.2.4 P40 (端口 4).....	33
2.2.5 P100,P101(端口 10).....	33
2.2.6 P112 ,P113 (端口 11).....	33
2.2.7 P114 ~ P124 (端口 12).....	34
2.2.8 P140 ~ P143 (端口 14).....	34
2.2.9 P140 ~ P143 (端口 14).....	35
2.2.10 AVREF (仅 $\mu$ PD78F041x).....	35
2.2.11 AVSS (仅 $\mu$ PD78F041x).....	35
2.2.12 COM0 ~ COM7.....	35
2.2.13 $V_{LC0} \sim V_{LC3}$ .....	35
2.2.14 RESET.....	35
2.2.15 REGC.....	36
2.2.16 VDD.....	36
2.2.17 VSS.....	36
2.2.18 FLMD0.....	36
2.3 引脚 I/O 电路和未使用引脚的建议连接方式.....	37
第三章 CPU 结构.....	41
3.1 存储器空间.....	41
3.1.1 内部程序存储器空间.....	47
3.1.2 内部数据存储器空间.....	49
3.1.3 特殊功能寄存器 (SFR) 区域.....	49
3.1.4 数据存储器空间寻址.....	50
3.2 处理器寄存器.....	54
3.2.1 控制寄存器.....	54
3.2.2 通用寄存器.....	58
3.2.3 特殊功能寄存器(SFRs).....	59
3.3 指令地址寻址.....	64

3.3.1 相对寻址.....	64
3.3.2 立即寻址.....	65
3.3.3 表间接寻址.....	66
3.3.4 寄存器寻址.....	90
<b>3.4 操作数地址寻址.....</b>	<b>67</b>
3.4.1 隐含寻址.....	67
3.4.2 寄存器寻址.....	68
3.4.3 直接寻址.....	69
3.4.4 短直接寻址.....	70
3.4.5 特殊功能寄存器 (SFR) 寻址.....	71
3.4.6 寄存器间接寻址.....	72
3.4.7 基址寻址.....	73
3.4.8 基址变址寻址.....	74
3.4.9 堆栈寻址.....	75
<b>第四章 端口功能.....</b>	<b>76</b>
<b>4.1 端口功能.....</b>	<b>76</b>
<b>4.2 端口配置.....</b>	<b>78</b>
4.2.1 端口 1.....	79
4.2.2 端口 2.....	81
4.2.3 端口 3.....	83
4.2.4 端口 4.....	85
4.2.5 端口 10.....	86
4.2.6 端口 11.....	87
4.2.7 端口 12.....	89
4.2.8 端口 14.....	93
4.2.9 端口 15.....	124
<b>4.3 控制端口功能的寄存器.....</b>	<b>95</b>
<b>4.4 端口功能操作.....</b>	<b>102</b>
4.4.1 写入 I/O 端口.....	102
4.4.2 从 I/O 端口读取.....	102
4.4.3 I/O 端口的操作.....	102
<b>4.5 使用复用功能时 PFALL, PF2, PF1, ISC 端口模式寄存器和输出锁存器的设置.....</b>	<b>103</b>
<b>第五章 时钟发生器.....</b>	<b>106</b>
<b>5.1 时钟发生器的功能.....</b>	<b>106</b>
<b>5.2 时钟发生器的配置.....</b>	<b>107</b>
<b>5.3 控制时钟发生器的寄存器.....</b>	<b>109</b>
<b>5.4 系统时钟振荡器.....</b>	<b>120</b>
5.4.1 X1 振荡器.....	120
5.4.2 XT1 振荡器.....	120
5.4.3 不使用副系统时钟.....	123
5.4.4 内部高速振荡器.....	123
5.4.5 内部低速振荡器.....	123
5.4.6 预分频器.....	123
<b>5.5 时钟发生器的操作.....</b>	<b>124</b>
<b>5.6 时钟控制.....</b>	<b>127</b>

5.6.1 控制高速系统时钟示例 .....	127
5.6.2 控制内部高速系统时钟示例 .....	129
5.6.3 控制副系统时钟示例 .....	131
5.6.4 控制内部低速振荡时钟示例 .....	133
5.6.5 CPU 和外部硬件所采用的时钟 .....	133
5.6.6 CPU 时钟状态转换图 .....	134
5.6.7 CPU 时钟切换之前的状况与切换之后的处理 .....	139
5.6.8 CPU 时钟和主系统时钟切换所需的时间 .....	140
5.6.9 时钟振荡停止前的状况 .....	141
5.6.10 外部硬件与源时钟 .....	142
<b>第六章 16 位定时器/事件计数器 00 .....</b>	<b>143</b>
6.1 16 位定时器/事件计数器 00 的功能 .....	143
6.2 16 位定时器/事件计数器 00 的配置 .....	144
6.3 控制 16 位定时器/事件计数器 00 的寄存器 .....	149
6.4 16 位定时器/事件计数器 00 的操作 .....	158
6.4.1 间隔定时器的操作 .....	158
6.4.2 方波输出操作 .....	161
6.4.3 外部事件计数器的操作 .....	164
6.4.4 通过 TI00n 引脚有效沿输入进入清零&启动模式的操作 .....	168
6.4.5 自由运行定时器操作 .....	181
6.4.6 PPG 输出操作 .....	190
6.4.7 单脉冲输出操作 .....	193
6.4.8 脉冲宽度测量操作 .....	198
6.4.9 外部 24 位事件计数器操作 .....	206
6.4.10 外部 24 位事件计数器注意事项 .....	210
6.5 TM00 的特殊用途 .....	212
6.5.1 TM00 操作期间重写 CR010 .....	212
6.5.2 LVS00 和 LVR00 的设置 .....	212
6.6 16 位定时器/事件计数器 00 注意事项 .....	214
<b>第七章 8 位定时器/事件计数器 50, 51 和 52 .....</b>	<b>218</b>
7.1 8 位定时器/事件计数器 50, 51 和 52 的功能 .....	218
7.2 8 位定时器/事件计数器 50, 51 和 52 的配置 .....	218
7.3 控制 8 位定时器/事件计数器 50, 51 和 52 的寄存器 .....	222
7.4 8 位定时器/事件计数器 50, 51 和 52 的操作 .....	260
7.4.1 用作间隔定时器 .....	229
7.4.2 用作外部事件计数器(仅对 TM52) .....	231
7.5 使用 8 位定时器/事件计数器 50, 51 和 52 的注意事项 .....	232
<b>第八章 8 位定时器 H0, H1,和 H2 .....</b>	<b>234</b>
8.1 8 位定时器 H0, H1,和 H2 的功能 .....	234
8.2 8 位定时器 H0, H1,和 H2 的配置 .....	234
8.3 控制 8 位定时器 H0, H1,和 H2 的寄存器 .....	239
8.4 8 位定时器 H0, H1,和 H2 的操作 .....	246
8.4.1 间隔定时器/方波输出操作 .....	246
8.4.2 PWM 输出操作 .....	249

8.4.3 载波发生器操作（仅用于 8 位定时器 H1） .....	255
<b>第九章 实时计数器 .....</b>	<b>262</b>
9.1 实时计数器的功能.....	262
9.2 实时计数器的配置.....	262
9.3 控制实时计数器的寄存器.....	264
9.4 实时计数器的操作.....	276
9.4.1 实时计数器的启动操作 .....	276
9.4.2 实时计数器的读/写 .....	277
9.4.3 设置实时计数器的闹表 .....	277
<b>第十章 看门狗定时器 .....</b>	<b>280</b>
10.1 看门狗定时器的功能 .....	280
10.2 看门狗定时器的配置 .....	281
10.3 控制看门狗定时器的寄存器 .....	282
10.4 看门狗定时器的操作 .....	283
10.4.1 看门狗定时器操作的控制.....	283
10.4.2 门狗定时器溢出时间的设置 .....	284
10.4.3 看门狗定时器窗口打开周期的设置 .....	285
<b>第十一章 蜂鸣器输出控制器 .....</b>	<b>287</b>
11.1 蜂鸣器输出控制器的功能.....	287
11.2 蜂鸣器输出控制器的配置.....	288
11.3 控制蜂鸣器输出控制器的寄存器 .....	288
11.4 蜂鸣器输出控制器的操作.....	290
<b>第十二章 10 位逐次逼近型 A/D 转换器 (仅<math>\mu</math>PD78F045x 和 78F046x) .....</b>	<b>291</b>
12.1 10 位逐次逼近型 A/D 转换器的功能 .....	291
12.2 10 位逐次逼近型 A/D 转换器的配置 .....	292
12.3 10 位逐次逼近型 A/D 转换器使用的寄存器 .....	294
12.4 10 位逐次逼近型 A/D 转换器的操作 .....	302
12.4.1 A/D 转换器的基本操作 .....	302
12.4.2 输入电压和转换结果.....	304
12.4.3 A/D 转换器的操作模式 .....	305
12.5 A/D 转换器特征表的阅读方法.....	307
12.6 10 位逐次逼近型 A/D 转换器的注意事项 .....	309
<b>第十三章 串行接口 UART0.....</b>	<b>313</b>
13.1 串行接口 UART0 的功能 .....	313
13.2 串行接口 UART0 的配置 .....	314
13.3 控制串行接口 UART0 的寄存器.....	317
13.4 串行接口 UART0 的操作 .....	323
13.4.1 操作停止模式.....	323
13.4.2 异步串行接口 (UART) 模式.....	324
13.4.3 专用波特率发生器 .....	330

13.4.4 计算波特率 .....	331
<b>第十四章 串行接口 UART6.....</b>	<b>335</b>
<b>14.1 串行接口 UART6 的功能 .....</b>	<b>335</b>
<b>14.2 串行接口 UART6 的配置 .....</b>	<b>339</b>
<b>14.3 控制串行接口 UART6 的寄存器 .....</b>	<b>342</b>
<b>14.4 串行接口 UART6 的操作 .....</b>	<b>353</b>
14.4.1 操作停止模式 .....	353
14.4.2 异步串行接口(UART)模式 .....	354
14.4.3 专用波特率发生器 .....	368
14.4.4 计算波特率 .....	370
<b>第十五章 LCD 控制器/驱动器.....</b>	<b>376</b>
<b>15.1 LCD 控制器/驱动器的功能 .....</b>	<b>376</b>
<b>15.2 LCD 控制器/驱动器的配置 .....</b>	<b>378</b>
<b>15.3 控制 LCD 控制器/驱动器的寄存器.....</b>	<b>380</b>
<b>15.4 设置 LCD 控制器/驱动器 .....</b>	<b>385</b>
<b>15.5 LCD 显示数据存储寄存器.....</b>	<b>386</b>
<b>15.6 Common 和 Segment 信号.....</b>	<b>387</b>
<b>15.7 显示模式.....</b>	<b>393</b>
15.7.1 静态显示示例 .....	393
15.7.2 2 分时显示示例.....	396
15.7.3 3 分时显示示例.....	399
15.7.4 4 分时显示示例.....	403
<b>15.8 LCD 驱动提供电压 V<sub>LC0</sub>, V<sub>LC1</sub>, V<sub>LC2</sub>, 和 V<sub>LC3</sub>.....</b>	<b>406</b>
15.8.1 内部分压模式 .....	406
15.8.2 外部分压模式 .....	408
<b>第十六章 MANCHESTER 编码发生器 .....</b>	<b>410</b>
<b>16.1 Manchester 编码发生器的功能 .....</b>	<b>410</b>
<b>16.2 Manchester 编码发生器的配置 .....</b>	<b>410</b>
<b>16.3 控制 Manchester 编码发生器的寄存器.....</b>	<b>413</b>
<b>16.4 Manchester 编码发生器的操作 .....</b>	<b>416</b>
16.4.1 操作停止模式 .....	416
16.4.2 Manchester 编码发生器模式 .....	417
16.4.3 位连续缓冲器模式 .....	426
<b>第十七章 中断功能 .....</b>	<b>435</b>
<b>17.1 中断类型.....</b>	<b>435</b>
<b>17.2 中断源及配置.....</b>	<b>435</b>
<b>17.3 控制中断功能的寄存器 .....</b>	<b>440</b>
<b>17.4 中断服务操作.....</b>	<b>447</b>
17.4.1 可屏蔽的中断响应 .....	447
17.4.2 软件中断请求响应 .....	449
17.4.3 中断嵌套 .....	450
17.4.4 保持中断请求 .....	453

第十八章 按键中断功能.....	454
18.1 按键中断功能 .....	454
18.2 按键中断的配置 .....	454
18.3 控制按键中断的寄存器 .....	455
第十九章 待机功能 .....	456
19.1 待机功能及配置 .....	456
19.1.1 待机功能.....	456
19.1.2 控制待机功能的寄存器 .....	457
19.2 待机功能的操作 .....	459
19.2.1 HALT 模式.....	459
19.2.2 STOP 模式 .....	464
第二十章 复位功能.....	470
20.1 确认复位源的寄存器 .....	478
第二十一章 上电清零电路.....	479
21.1 上电清零电路的功能 .....	479
21.2 上电清零电路的配置 .....	480
21.3 上电清零电路的操作 .....	480
21.4 上电清零电路的注意事项 .....	483
第二十二章 低电压检测器 .....	485
22.1 低电压检测器的功能 .....	485
22.2 低电压检测器的配置 .....	486
22.3 控制低电压检测器的寄存器 .....	486
22.4 低电压检测器的操作 .....	489
22.4.1 用于复位.....	490
22.4.2 用于中断.....	495
22.5 低电压检测器的注意事项 .....	500
第二十三章 选项字节 .....	503
23.1 选项字节的功能 .....	503
23.2 选项字节的格式 .....	595
第二十四章 Flash 存储器.....	508
24.1 内部存储器容量切换寄存器 .....	508
24.2 用 Flash 编程器写入数据 .....	509
24.3 编程环境 .....	511
24.4 通信模式 .....	511
24.5 在线方式的引脚处理 .....	513
24.5.1 FLMD0 引脚 .....	513
24.5.2 串行接口引脚.....	513
24.5.3 RESET 引脚.....	515
24.5.4 端口引脚.....	515

24.5.5 REGC 引脚 .....	515
24.5.6 其它信号引脚 .....	516
24.5.7 供电电压 .....	516
<b>24.6 编程方法 .....</b>	<b>517</b>
24.7.1 控制 Flash 存储器 .....	517
24.7.2 Flash 存储器编程模式 .....	517
24.7.3 选择通信模式 .....	518
24.7.4 通信命令 .....	519
<b>24.7 安全性设置 .....</b>	<b>520</b>
<b>24.8 通过自编程进行 Flash 存储器编程(开发中) .....</b>	<b>522</b>
24.8.1 引导交换功能 .....	524
<b>第二十五章 片上调试功能 .....</b>	<b>526</b>
<b>25.1 QB-78K0MINI 和 78K0/LE3 的连接 .....</b>	<b>526</b>
<b>25.2 片上调试安全 ID .....</b>	<b>527</b>
<b>第二十六章 指令集 .....</b>	<b>528</b>
<b>26.1 操作列表使用规则 .....</b>	<b>528</b>
26.1.1 操作数标识符和标识方法 .....	529
26.1.2 操作栏描述 .....	529
26.1.3 标志操作栏的描述 .....	529
<b>26.2 操作列表 .....</b>	<b>530</b>
<b>26.3 按寻址类型列出指令 .....</b>	<b>538</b>
<b>第二十七章 电气特性 (标准产品) .....</b>	<b>541</b>
<b>第二十八章 封装图 .....</b>	<b>559</b>
<b>第二十九章 等待注意事项 .....</b>	<b>560</b>
<b>29.1 等待注意事项 .....</b>	<b>560</b>
<b>29.2 产生等待的外围硬件 .....</b>	<b>561</b>

## 第一章 概述

### 1.1 特征

- 指令最短执行时间可以在高速（0.2  $\mu\text{s}$ : @高速系统时钟的操作频率为 10 MHz）和超低速（122  $\mu\text{s}$ : @副系统时钟的操作频率为 32.768 kHz）之间改变
- 通用寄存器: 8 位  $\times$  32 个寄存器（8 位  $\times$  8 个寄存器  $\times$  4 bank）
- ROM, RAM 的容量

产品型号 \ 项目	程序存储器 (ROM)		数据存储器	
			内部高速 RAM <sup>注</sup>	LCD 显示 RAM
$\mu\text{PD78F0400}$ , 78F0410	Flash 存储器 <sup>注</sup>	8 KB	512 字节	22 $\times$ 4 位 (带有 4 个 com 端口)
$\mu\text{PD78F0401}$ , 78F0411		16 KB	768 字节	
$\mu\text{PD78F0402}$ , 78F0412		24 KB	1 KB	18 $\times$ 8 位 (带有 8 个 com 端口)
$\mu\text{PD78F0403}$ , 78F0413		32 KB		

注 通过使用内部存储器容量切换寄存器（IMS）可以改变内部 Flash 存储器、内部高速 RAM 的容量。

- 内置单电源 Flash 存储器
- 自编程（具有启动交换功能）
- 片上调试功能
- 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（使用内部低速振荡时钟进行操作）
- LCD 控制器/驱动器（内部电压泵，外部电阻分压和内部电阻分压可切换使用）
  - Segment 信号: 22, Common 信号: 4（带有 4 个 com 端口）
  - Segment 信号: 18, Common 信号: 8（带有 4 个 com 端口）
- 内置按键中断功能: 3 通道
- 内置蜂鸣器输出控制器
- I/O 端口: 30
- 定时器: 9 通道
  - 16 位定时/事件计数器: 1 通道
  - 8 位定时/事件计数器: 3 通道
  - 8 位定时器: 3 通道
  - 实时计数器（RTC）: 1 通道
  - 看门狗定时器: 1 通道
- 串行接口: 2 通道
  - UART（支持 LIN（本地互连网络）总线）: 1 通道
  - UART: 1 通道
- 10 位逐次逼近 A/D 转换器: 6 通道（仅限于  $\mu\text{PD78F041x}$ ）
- 曼彻斯特代码发生器
- 电源电压:  $V_{\text{DD}} = 1.8 \sim 5.5 \text{ V}$
- 运行环境温度:  $T_{\text{A}} = -40 \sim +85^\circ\text{C}$

## 1.2 应用

数字相机，AV 设备，家用电器，工业仪表，医疗设备和测量仪器等。

## 1.3 订购信息

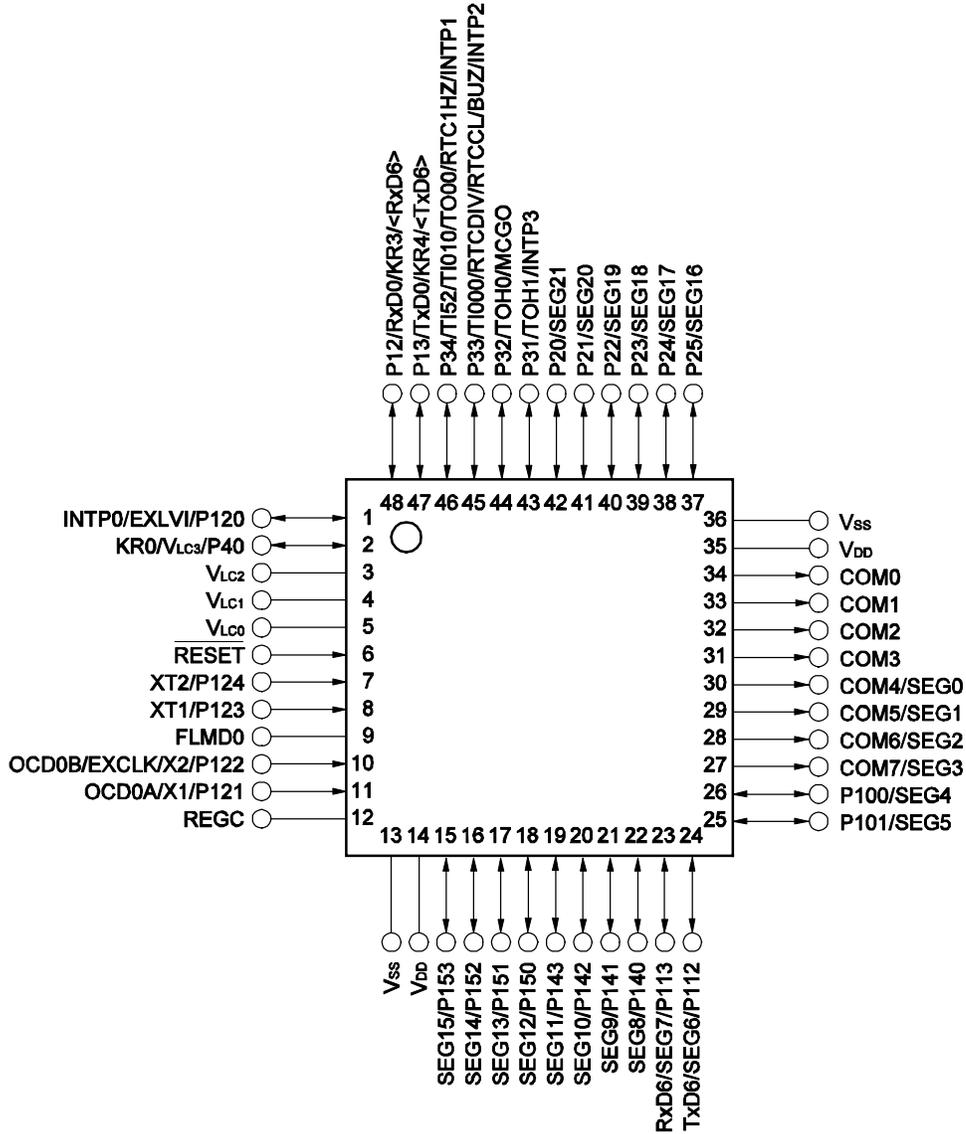
- Flash 存储器版本（无铅产品）

产品型号	封装
$\mu$ PD78F0400GA-GAM-AX	48 引脚塑封 LQFP（微间距）（7 × 7）
$\mu$ PD78F0401GA-GAM-AX	48 引脚塑封 LQFP（微间距）（7 × 7）
$\mu$ PD78F0402GA-GAM-AX	48 引脚塑封 LQFP（微间距）（7 × 7）
$\mu$ PD78F0403GA-GAM-AX	48 引脚塑封 LQFP（微间距）（7 × 7）
$\mu$ PD78F0410GA-GAM-AX	48 引脚塑封 LQFP（微间距）（7 × 7）
$\mu$ PD78F0411GA-GAM-AX	48 引脚塑封 LQFP（微间距）（7 × 7）
$\mu$ PD78F0412GA-GAM-AX	48 引脚塑封 LQFP（微间距）（7 × 7）
$\mu$ PD78F0413GA-GAM-AX	48 引脚塑封 LQFP（微间距）（7 × 7）

### 1.4 引脚配置（俯视图）

#### (1) $\mu$ PD78F0400, 78F0401, 78F0402, 78F0403

- 48 引脚塑封 LQFP（微间距）（7 × 7）

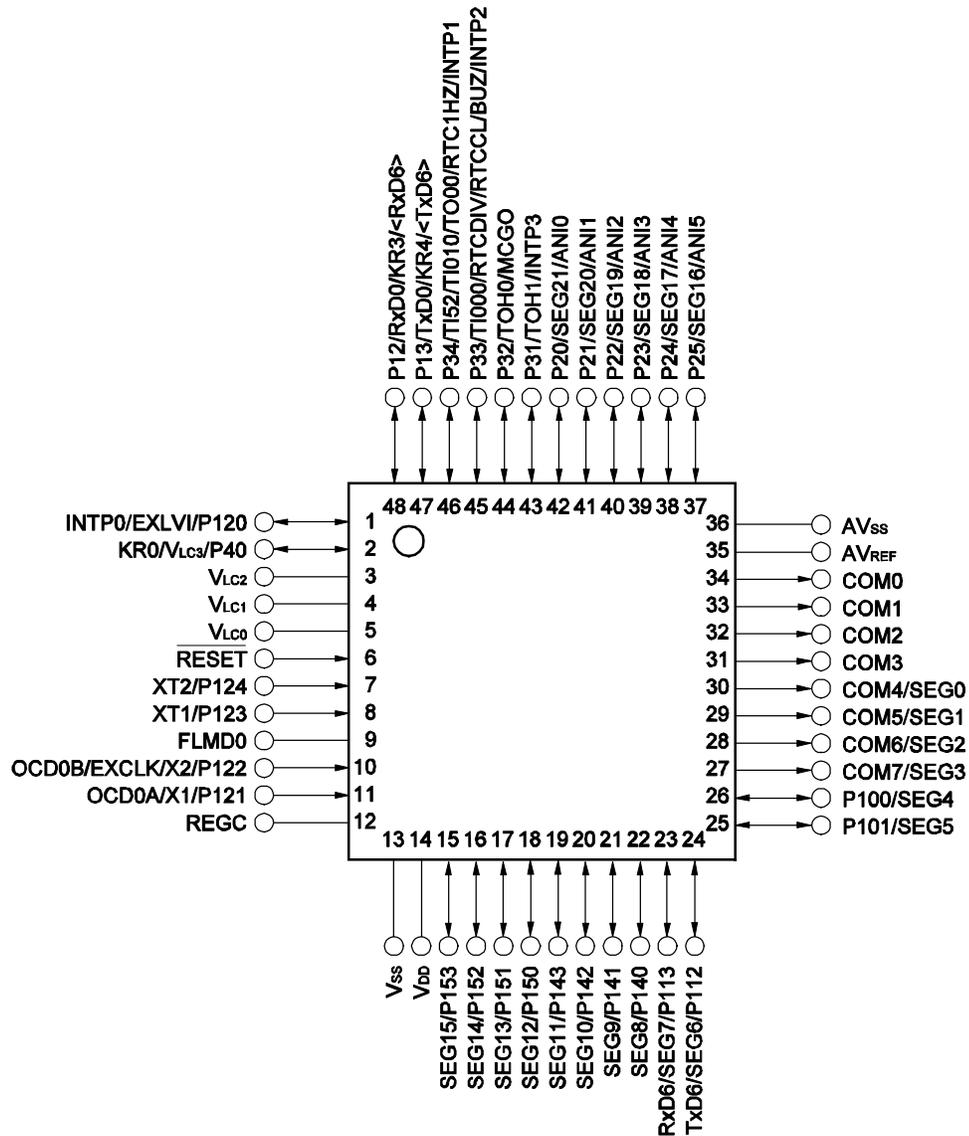


- 注意事项**
1. 通过一个电容（0.47 $\mu$ F ~ 1 $\mu$ F: 推荐）将 REGC 引脚和 V<sub>SS</sub> 连接到一起。
  2. 当使用 Flash 编程器对芯片进行写入操作时，只能使用底部的引脚（引脚 23 和 24）对应为 UART6 引脚（RxD6 和 TxD6），而不能使用顶部的引脚（引脚 47 和 48）。
  3. 确保 V<sub>DD</sub>（引脚 14）和 V<sub>DD</sub>（引脚 35）以及 V<sub>SS</sub>（引脚 13）和 V<sub>SS</sub>（引脚 36）具有相同的电位。

**备注** 带有箭头（<>）的功能可以通过设定输入交换控制寄存器（ISC）分配。

(2)  $\mu$ PD78F0410, 78F0411, 78F0412, 78F0413

- 48 引脚塑封 LQFP (微间距) (7 × 7)



- 注意事项**
1. 使 AV<sub>ss</sub> 与 V<sub>ss</sub> 连接。
  2. 通过一个电容 (0.47  $\mu$ F ~ 1  $\mu$ F: 推荐) 将 REGC 引脚和 V<sub>ss</sub> 连接到一起。
  3. 复位释放后, ANI0/P20~ANI5/P25 被设置为模拟输入模式。
  4. 当使用 Flash 编程器对芯片进行写入操作时, 只能使用底部的引脚 (引脚 23 和 24) 对应为 UART6 引脚 (RxD6 和 TxD6), 而不能使用顶部的引脚 (引脚 47 和 48)。

**备注** 带有箭头 (<>) 的功能可以通过设定输入交换控制寄存器 (ISC) 分配。

## 引脚标识

ANI0~ANI5 <sup>注</sup> :	模拟输入	P150~ P153:	端口 15
AVREF <sup>注</sup> :	模拟参考电压	REGC:	校准电容
AVSS <sup>注</sup> :	模拟地	$\overline{\text{RESET}}$ :	复位
BUZ:	蜂鸣器输出	RxD0, RxD6:	接收数据
COM0~COM7:	Common 输出	RTC1HZ:	实时计数器修正时钟 (1Hz) 输出
EXCLK:	外部时钟输出 (主系统时钟)	RTCCL:	实时计数器时钟 (32.768 kHz 源振荡) 输出
EXLVI:	低电压检测的外部电压输入	RTCDIV:	实时计数器时钟 (32.768 kHz 分频) 输出
FLMD0:	Flash 编程模式	SEG0~SEG21:	Segment 输出
INTP0~INTP3:	外部中断输入	TI000, TI010:	定时器输入
KR0, KR3, KR4:	按键返回	TO00:	定时器输出
MCGO:	曼彻斯特代码发生器输出	TOH0, TOH1:	定时器输出
OCD0A, OCD0B:	片上调试输入/输出	TxD0, TxD6:	发送数据
P12, P13:	端口 1	VDD:	电源
P20~ P25:	端口 2	VSS:	地
P31~P34:	端口 3	VLC0~VLC3:	LCD 电源
P40:	端口 4	X1, X2:	晶体振荡器 (主系统时钟)
P100, P101:	端口 10	XT1, XT2:	晶体振荡器 (副系统时钟)
P112, P113:	端口 11		
P120~P124:	端口 12		
P140~ P143:	端口 14		

注 仅限于 $\mu\text{PD78F041x}$ 。

## 1.5 78K0/Lx3 系列微控制器产品线

ROM	RAM	78K0/LC3	78K0/LD3	78K0/LE3	78K0/LF3
		48 引脚	52 引脚	64 引脚	80 引脚
60 KB	2 KB	–	–	μPD78F0465 μPD78F0455 μPD78F0445	μPD78F0495 μPD78F0485 μPD78F0475
48 KB	2 KB	–	–	μPD78F0464 μPD78F0454 μPD78F0444	μPD78F0494 μPD78F0484 μPD78F0474
32 KB	1 KB	μPD78F0413 μPD78F0403	μPD78F0433 μPD78F0423	μPD78F0463 μPD78F0453 μPD78F0443	μPD78F0493 μPD78F0483 μPD78F0473
24 KB	1 KB	μPD78F0412 μPD78F0402	μPD78F0432 μPD78F0422	μPD78F0462 μPD78F0452 μPD78F0442	μPD78F0492 μPD78F0482 μPD78F0472
16 KB	768 B	μPD78F0411 μPD78F0401	μPD78F0431 μPD78F0421	μPD78F0461 μPD78F0451 μPD78F0441	μPD78F0491 μPD78F0481 μPD78F0471
8 KB	512 B	μPD78F0410 μPD78F0400	μPD78F0430 μPD78F0420	–	–

78K0/Lx3 系列微控制器功能列表如下所示。

(1/3)

产品编号		78K0/LC3								78K0/LD3							
		μPD78F040x				μPD78F041x				μPD78F042x				μPD78F043x			
项目		48 引脚								52 引脚							
Flash 存储器 (KB)		8	16	24	32	8	16	24	32	8	16	24	32	8	16	24	32
RAM (KB)		0.5	0.75	1	1	0.5	0.75	1	1	0.5	0.75	1	1	0.5	0.75	1	1
电源电压		V <sub>DD</sub> = 1.8~5.5 V															
稳压器		提供															
指令最短执行时间		0.2 μs (10 MHz: V <sub>DD</sub> = 2.7~5.5 V)/ 0.4 μs (5 MHz: V <sub>DD</sub> = 1.8~5.5 V)															
时钟	主时钟	高速系统时钟 10 MHz: V <sub>DD</sub> = 2.7~5.5 V/5 MHz: V <sub>DD</sub> = 1.8~5.5 V															
		内部高速振荡时钟 8 MHz (TYP.): V <sub>DD</sub> = 1.8~5.5 V															
	副时钟 32.768 kHz (TYP.): V <sub>DD</sub> = 1.8~5.5 V																
	内部低速振荡时钟 240 kHz (TYP.): V <sub>DD</sub> = 1.8~5.5 V																
端口	总计	30								34							
	定时器	16 位 (TM0) 1 通道 8 位 (TM5) 3 通道 8 位 (TMH) 3 通道 RTC 1 通道 WDT 1 通道															
串行接口	3 线 CSI	-								1 通道 <sup>注 1</sup>							
	UART	1 通道								1 通道 <sup>注 1</sup>							
	支持 LIN 总线的 UART	1 通道 <sup>注 2</sup>								1 通道 <sup>注 3</sup>							
LCD	类型	外部电阻分压和内部电阻分压可切换使用															
	Segment 信号	22 (18) <sup>注 4</sup>								24 (20) <sup>注 4</sup>							
	Common 信号	4 (8) <sup>注 4</sup>															
10-位 逐次逼近 A/D		-				6 通道				-				6 通道			
16 位 ΔΣ 型 A/D		-															
中断	外部	5															
	内部	17				18				19				20			
按键中断		3 通道								5 通道							
复位	RESET 引脚	提供															
	POC	1.59 V ± 0.15 V (上升到 1.8 V 所用时间: 3.6 ms (MAX.))															
	LVI	可选择 16 个电源电压检测级别															
	WDT	提供															
时钟输出		-															
蜂鸣器输出		提供															
远程控制接收器		-								提供							
MCG		提供															
片上调试功能		提供															
运行环境温度		T <sub>A</sub> = -40~+85°C															

注 1. 由于 3 线 CSI 和 UART 使用交替功能引脚, 所以只能使用其中的一个功能。

2. 支持 LIN 总线的 UART 引脚可以变为 UART 引脚(引脚 47 和 48)。

3. 支持 LIN 总线的 UART 引脚可以变为 3 线 CSI/UART 引脚(引脚 50 和 51)。

4. 当使用 8COM 端口时, 括号中的值为信号输出的数量。

产品型号		78K0/LE3														
		$\mu$ PD78F044x					$\mu$ PD78F045x					$\mu$ PD78F046x				
项目		64 引脚														
Flash 存储器(KB)		16	24	32	48	60	16	24	32	48	60	16	24	32	48	60
RAM (KB)		0.75	1	1	2	2	0.75	1	1	2	2	0.75	1	1	2	2
电源电压		V <sub>DD</sub> = 1.8~5.5 V														
稳压器		提供														
指令最短执行时间		0.2 $\mu$ s (10 MHz: V <sub>DD</sub> = 2.7~5.5 V)/ 0.4 $\mu$ s (5 MHz: V <sub>DD</sub> = 1.8~5.5 V)														
时钟	高速系统时钟	10 MHz: V <sub>DD</sub> = 2.7~5.5 V/5 MHz: V <sub>DD</sub> = 1.8~5.5 V														
	内部高速振荡时钟	8 MHz (TYP.): V <sub>DD</sub> = 1.8~5.5 V														
	副时钟	32.768 kHz (TYP.): V <sub>DD</sub> = 1.8~5.5 V														
	内部低速振荡时钟	240 kHz (TYP.): V <sub>DD</sub> = 1.8~5.5 V														
端口	总计	46														
	定时器															
串行接口	16 位(TM0)	1 通道														
	8 位(TM5)	3 通道														
	8 位(TMH)	3 通道														
	RTC	1 通道														
	WDT	1 通道														
LCD	3 线 CSI/UART <sup>注1</sup>	1 通道														
	支持 LIN 总线的 UART <sup>注2</sup>	1 通道														
类型	类型	外部电阻分压和内部电阻分压可切换使用														
	Segment 信号	32 (28) <sup>注3</sup>										24 (20) <sup>注3</sup>				
	Common 信号	4 (8) <sup>注3</sup>														
10 位逐次逼近 A/D		-					8 类型									
16 位 $\Delta\Sigma$ 型 A/D		-										3 类型				
中断	外部	6														
	内部	19					20					21				
按键中断		5 通道														
复位	RESET 引脚	提供														
	POC	1.59 V $\pm$ 0.15 V (上升到 1.8 V 所用时间 : 3.6 ms (MAX.))														
	LVI	可选择 16 个电源电压检测级别														
	WDT	提供														
时钟输出		-														
蜂鸣器输出		提供														
远程控制接收器		提供														
MCG		提供														
片上调试功能		提供														
运行环境温度		T <sub>A</sub> = -40~+85°C														

注 1. 复用功能引脚只能选择一种功能。

2. 支持 LIN 总线的 UART 引脚可以变为 3 线 CSI/UART 引脚(引脚 62 和 63)。

3. 当使用 8COM 端口时, 括号中的值为信号输出的数量。

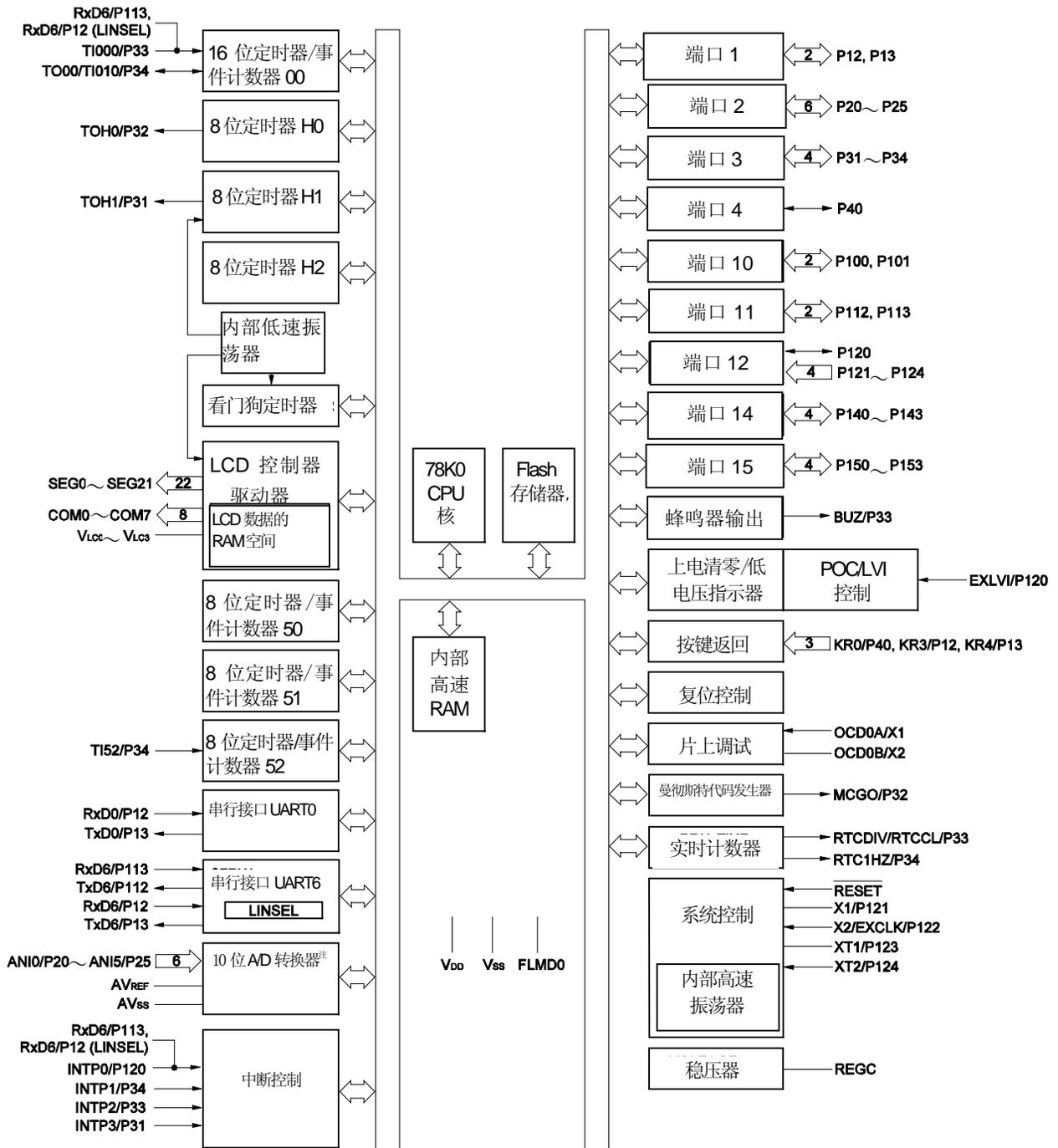
产品型号		78K0/LF3														
		μPD78F047x					μPD78F048x					μPD78F049x				
项目		80 Pins														
Flash 存储器(KB)		16	24	32	48	60	16	24	32	48	60	16	24	32	48	60
RAM (KB)		0.75	1	1	2	2	0.75	1	1	2	2	0.75	1	1	2	2
电源电压		V <sub>DD</sub> = 1.8 ~ 5.5 V														
稳压器		提供														
指令最短执行时间		0.2 μs (10 MHz: V <sub>DD</sub> = 2.7 ~ 5.5 V)/ 0.4 μs (5 MHz: V <sub>DD</sub> = 1.8 ~ 5.5 V)														
时钟	主时钟	高速系统时钟 10 MHz: V <sub>DD</sub> = 2.7 ~ 5.5 V/5 MHz: V <sub>DD</sub> = 1.8 ~ 5.5 V														
		内部高速振荡时钟 8 MHz (TYP.): V <sub>DD</sub> = 1.8 ~ 5.5 V														
	副时钟 32.768 kHz (TYP.): V <sub>DD</sub> = 1.8 ~ 5.5 V															
	内部低速振荡时钟 240 kHz (TYP.): V <sub>DD</sub> = 1.8 ~ 5.5 V															
端口	总计	62														
	定时器	16 位 (TM0) 1 通道														
	8 位 (TM5)	3 通道														
	8 位 (TMH)	3 通道														
	RTC	1 通道														
	WDT	1 通道														
串行接口	3 线 CSI/UART <sup>注1</sup>	1 通道														
	自动发送/ 接收 3 线 CSI	1 通道														
	支持 LIN 总线的 UART <sup>注2</sup>	1 通道														
LCD	类型	外部电阻分压和内部电阻分压可切换使用														
	Segment 信号	40 (36) <sup>注3</sup>										32 (28) <sup>注3</sup>				
	Common 信号	4 (8) <sup>注3</sup>														
10 位逐次逼近 A/D		-					8 通道									
16 位ΔΣ类型 A/D		-										3 通道				
中断	外部	7														
	内部	20					21					22				
按键中断		8 通道														
复位	RESET 引脚	提供														
	POC	1.59 V ±0.15 V (上升到 1.8 V 所用时间: 3.6 ms (MAX.))														
	LVI	可选择 16 个电源电压检测级别														
	WDT	提供														
时钟输出/蜂鸣器输出		提供														
远程控制接收器		提供														
MCG		提供														
片上调试功能		提供														
运行环境温度		T <sub>A</sub> = -40 ~ +85°C														

注 1. 复用功能引脚只能选择一种功能。

2. 支持 LIN 总线的 UART 引脚可以变为自动发送/接收 3 线 CSI/UART 引脚(引脚编号 75 和 76)。

3. 当使用 8COM 端口时, 括号中的值为信号输出的数量。

1.6 框图



注 仅限于μPD78F041x。

## 1.7 功能概述

(1/2)

项目		$\mu$ PD78F0400 $\mu$ PD78F0410	$\mu$ PD78F0401 $\mu$ PD78F0411	$\mu$ PD78F0402 $\mu$ PD78F0412	$\mu$ PD78F0403 $\mu$ PD78F0413
内部存储器	Flash 存储器 (支持自编程功能) <sup>注</sup>	8 KB	16 KB	24 KB	32 KB
	高速 RAM <sup>注</sup>	512 字节	768 字节	1 KB	
	LCD 显示 RAM	22 × 4 位 (带有 4 com) 或 18 × 8 位 (带有 8 com)			
存储器空间		64 KB			
主系统时钟 (振荡频率)	高速系统时钟	X1(晶体/陶瓷)振荡, 外部主系统时钟输入(EXCLK) 2 ~ 10 MHz: $V_{DD} = 2.7 \sim 5.5$ V, 2 ~ 5 MHz: $V_{DD} = 1.8 \sim 5.5$ V			
	内部高速振荡时钟	内部振荡 8 MHz (TYP.): $V_{DD} = 1.8 \sim 5.5$ V			
副系统时钟 (振荡频率)		XT1 (晶体) 振荡 32.768 kHz (TYP.): $V_{DD} = 1.8 \sim 5.5$ V			
内部低速振荡时钟 (提供给 TMH1 和 WDT)		内部振荡 240 kHz (TYP.): $V_{DD} = 1.8 \sim 5.5$ V			
通用目的寄存器		8 位 × 32 寄存器 (8 位 × 8 寄存器 × 4 banks)			
指令最短执行时间		0.2 $\mu$ s (高速系统时钟: @ $f_{XH} = 10$ MHz 操作)			
		0.25 $\mu$ s (内部高速振荡时钟: @ $f_{RH} = 8$ MHz (TYP.) 操作)			
		122 $\mu$ s (副系统时钟: @ $f_{SUB} = 32.768$ kHz 操作)			
指令集		<ul style="list-style-type: none"> <li>• 8 位操作和 16 位操作</li> <li>• 位操作 (置位, 复位, 测试和布尔操作)</li> <li>• BCD 调整, 等</li> </ul>			
I/O 端口		总计: 30 CMOS I/O: 26 CMOS 输入: 4			
定时器		<ul style="list-style-type: none"> <li>• 16 位定时/事件计数器: 1 通道</li> <li>• 8 位定时/事件计数器: 3 通道</li> <li>• 8 位定时器: 3 通道 (其中 2 通道支持 PWM 输出)</li> <li>• 实时计数器: 1 通道</li> <li>• 看门狗定时器: 1 通道</li> </ul>			
		定时器输出	3 (PWM 输出: 2 以及 PPG 输出: 1)		
		RTC 输出	2		
蜂鸣器输出		<ul style="list-style-type: none"> <li>• 1.22 kHz, 2.44 kHz, 4.88 kHz, 9.77 MHz</li> </ul> (外围硬件时钟: @ $f_{PRS} = 10$ MHz 操作)			

注 通过使用内部存储器容量切换寄存器(IMS), 可以改变内部 Flash 存储器和内部高速 RAM 的容量。

项目	$\mu$ PD78F0400 $\mu$ PD78F0410	$\mu$ PD78F0401 $\mu$ PD78F0411	$\mu$ PD78F0402 $\mu$ PD78F0412	$\mu$ PD78F0403 $\mu$ PD78F0413
10 位逐次逼近型 A/D 转换器	<ul style="list-style-type: none"> <li>• <math>\mu</math>PD78F040x: 无</li> <li>• <math>\mu</math>PD78F041x: 6 通道</li> </ul>			
串行接口	<ul style="list-style-type: none"> <li>• 支持 LIN 总线的 UART <sup>注1</sup>: 1 通道</li> <li>• UART: 1 通道</li> </ul>			
LCD 控制器/驱动器	<ul style="list-style-type: none"> <li>• 外部电阻分压和内部电阻分压可切换使用</li> <li>• Segment 信号输出: 22 (18)<sup>注2</sup></li> <li>• Common 信号输出: 4 (8)<sup>注2</sup></li> </ul>			
曼彻斯特代码发生器	提供			
矢量中断源	内部	<ul style="list-style-type: none"> <li>• <math>\mu</math>PD78F040x: 17</li> <li>• <math>\mu</math>PD78F041x: 18</li> </ul>		
	外部	5		
按键中断	按键输入引脚的(KR0, KR3, KR4).下降沿产生按键中断 (INTKR)			
复位	<ul style="list-style-type: none"> <li>• 使用 RESET 引脚复位</li> <li>• 使用看门狗定时器内部复位</li> <li>• 使用 POC 电路内部复位</li> <li>• 使用低电压检测电路内部复位</li> </ul>			
片上调试功能	提供			
电源电压	$V_{DD} = 1.8 \sim 5.5 V$			
工作环境温度	$T_A = -40 \sim +85^\circ C$			
封装形式	48 引脚塑封 LQFP (微间距) (7 × 7)			

- 注 1. 支持 LIN 总线的 UART 引脚可以变为 UART 引脚 (引脚编号 47 和 48)。  
 2. 当使用 8COM 端口时, 括号中的值为信号输出的数量。

以下是定时器概要说明

		16 位定时器/ 事件计数器 00	8 位定时/ 事件计数器 50, 51 和 52			8 位定时器 H0, H1 和 H2			实时计数 器	看门狗定 时器
		TM00	TM50	TM51	TM52	TMH0	TMH1	TMH2		
功能	间隔定时器	1 通道	1 通道	1 通道	1 通道	1 通道	1 通道	1 通道	1 通道 注 1	-
	外部事件计数器	1 通道注 2	-	-	1 通道注 2	-	-	-注 2	-	-
	PPG 输出	1 输出	-	-	-	-	-	-	-	-
	PWM 输出	-	-	-	-	1 输出	1 输出	-	-	-
	脉冲宽度测量	2 输入	-	-	-	-	-	-	-	-
	方波输出	1 输出	-	-	-	1 输出	1 输出	-	-	-
	载波发生器	-	-	-注 3	-	-	1 输出注 3	-	-	-
	日历功能	-	-	-	-	-	-	-	1 通道注 1	-
	RTC 输出	-	-	-	-	-	-	-	2 输出注 4	-
看门狗定时器	-	-	-	-	-	-	-	-	1 通道	
中断源		2	1	1	1	1	1	1	1	-

注 1. 对于实时计数器，间隔定时功能和日历功能可以同时使用。

2. TM52 和 TM00 可以结合在一起作为 24 位计数器。同样，TM52 的外部事件也可以通过 TMH2 来控制是否允许输入。

3. TM51 和 TMH1 可联合使用，作为载波发生器模式。

4. 可以输出 1 Hz 信号，同时输出 512 Hz，16.384 kHz 或是 32.768 kHz 信号。

## 第二章 引脚功能

### 2.1 引脚功能列表

有三种类型的引脚 I/O 缓冲器电源：AV<sub>REF</sub><sup>注</sup>，V<sub>LC0</sub>和 V<sub>DD</sub>。下表显示了这些供电电源与引脚之间的关系。

表 2-1. 引脚 I/O 缓冲器供电电源

电源电压	相关引脚
AV <sub>REF</sub> <sup>注</sup>	P20 ~ P25
V <sub>LC0</sub>	COM0 ~ COM7, SEG0 ~ SEG21, V <sub>LC0</sub> ~ V <sub>LC3</sub>
V <sub>DD</sub>	其它引脚

注 仅限于μPD78F041x only。μPD78F040x 的电源电压为 V<sub>DD</sub>。

#### (1) 端口引脚

(1/2)

引脚名称	I/O	功能	复位后	复用功能
P12	I/O	端口 1。 2 位 I/O 端口。 可以位选输入/输出模式。 通过软件设置，可以定义内置上拉电阻的使用。	输入端口	RxD0/KR3/<RxD6>
P13				TxD0/KR4/<TxD6>
P20	I/O	端口 2。 6 位 I/O 端口。 可以位选输入/输出模式。	数字输入端口	SEG21/ANI0 <sup>注</sup>
P21				SEG20/ANI1 <sup>注</sup>
P22				SEG19/ANI2 <sup>注</sup>
P23				SEG18/ANI3 <sup>注</sup>
P24				SEG17/ANI4 <sup>注</sup>
P25				SEG16/ANI5 <sup>注</sup>
P31	I/O	端口 3。 4 位 I/O 端口。 可以位选输入/输出模式。 通过软件设置，可以定义内置上拉电阻的使用。	输入端口	TOH1/INTP3
P32				TOH0/MCGO
P33				TI000/RTCDIV/ RTCCL/BUZ/INTP2
P34				TI52/TI010/TO00/ RTC1HZ/INTP1
P40	I/O	端口 4。 1 位 I/O 端口 可以位选输入/输出模式。 通过软件设置，可以定义内置上拉电阻的使用。	输入端口	V <sub>LC3</sub> /KR0

注 仅限于μPD78F041x。

备注 带有箭头(< >) 的功能可以通过设定输入转换控制寄存器(ISC)分配。

## (1) 端口引脚

(2/2)

引脚名称	I/O	功能	复位后	复用功能
P100, P101	I/O	端口 10。 2 位 I/O 端口。 可以位选输入/输出模式。 通过软件设置，可以定义内置上拉电阻的使用。	输入端口	SEG4, SEG5
P112	I/O	端口 11。 2 位 I/O 端口 可以位选输入/输出模式。 通过软件设置，可以定义内置上拉电阻的使用。	输入端口	SEG6/TxD6
P113				SEG7/RxD6
P120	I/O	端口 12。 1 位 I/O 端口和 4 位输入端口。 对于 P120，通过软件设置，可以定义内置上拉电阻的使用。	输入端口	INTP0/EXLVI
P121				X1/OCD0A
P122				X2/EXCLK/OCD0B
P123				XT1
P124				XT2
P140 ~ P143	I/O	端口 14。 4 位 I/O 端口。 可以位选输入/输出模式。 通过软件设置，可以定义内置上拉电阻的使用。	输入端口	SEG8 ~ SEG11
P150 ~ P153	I/O	端口 15。 4 位 I/O 端口。 可以位选输入/输出模式。 通过软件设置，可以定义内置上拉电阻的使用。	输入端口	SEG12 ~ SEG15

## (2) 非端口引脚

(1/2)

引脚名称	I/O	功能	复位后	复用功能
ANI0 <sup>注</sup>	输入	10 位逐次逼近型 A/D 转换模拟输入。	数字输入端口	P20/SEG21
ANI1 <sup>注</sup>				P21/SEG20
ANI2 <sup>注</sup>				P22/SEG19
ANI3 <sup>注</sup>				P23/SEG18
ANI4 <sup>注</sup>				P24/SEG17
ANI5 <sup>注</sup>				P25/SEG16
AV <sub>REF</sub> <sup>注</sup>	输入	10 逐次逼近型 A/D 转换器参考电压输入，端口 2 的正电源	-	-
AV <sub>SS</sub> <sup>注</sup>	-	A/D 转换器的地电势。与 V <sub>SS</sub> 电势相同。	-	-
SEG0 ~ SEG3	输出	LCD 控制器/驱动器 segment 信号输出	输出端口	COM4 ~ COM7
SEG4, SEG5			输入端口	P100, P101
SEG6				P112/TxD6
SEG7				P113/RxD6
SEG8 ~ SEG11				P114 ~ P143
SEG12 ~ SEG15				P150 ~ P153
SEG16			数字输入端口	P25/ANI5 <sup>注</sup>
SEG17				P24/ANI4 <sup>注</sup>
SEG18				P23/ANI3 <sup>注</sup>
SEG19				P22/ANI2 <sup>注</sup>
SEG20				P21/ANI1 <sup>注</sup>
SEG21				P20/ANI0 <sup>注</sup>
COM0 ~ COM3			输出	LCD 控制器/驱动器 common 信号输出
COM4 ~ COM7	SEG0 ~ SEG3			
V <sub>LC0</sub> ~ V <sub>LC2</sub>	-	LCD 驱动电压	-	-
V <sub>LC3</sub>			输入端口	P40/KR0
BUZ	输出端口	蜂鸣器输出	输入端口	P33/TI000/RTCDIV/ RTCCL/INTP2
INTP0	输入	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。	输入端口	P120/EXLVI
INTP1				P34/TI52/TI010/ TO00/RTC1HZ
INTP2				P33/TI000/RTCDIV/ RTCCL/BUZ
INTP3				P31/TOH1
KR0	输入	按键中断输入	输入端口	P40/V <sub>LC3</sub>
KR3				P12/RxD0/<RxD6>
KR4				P13/TxD0/<TxD6>
MCGO	输出	曼彻斯特代码输出	输入	P32/TOH0

注 仅限于 $\mu$ PD78F041x。

备注 带有箭头(<>)的功能可以通过设定输入转换控制寄存器(ISC)分配。

## (2) 非端口引脚

(2/2)

引脚名称	I/O	功能	复位后	复用功能
RESET	输入	系统复位输入	-	-
RTCDIV	输出	实时计数器时钟(32 kHz 分频)输出	输入端口	P33/TI000/RTCCCL /BUZ/INTP2
RTCCCL	输出	实时计数器时钟(32 kHz 原始振荡)输出	输入端口	P33/TI000/RTCDIV /BUZ/INTP2
RTC1HZ	输出	实时计数器时钟(1 Hz)输出	输入端口	P34/TI52/TI010/ TO00/INTP1
RxD0	输入	到异步串行接口的串行数据输入。	输入端口	P12/KR3/<RxD6>
RxD6				P113/SEG7
<RxD6>				P12/RxD0/KR3
TI000	输入	到 16 位定时器/事件计数器 00 的外部时钟输入。 16 位定时器/事件计数器 00 的捕捉寄存器(CR000、CR010)的 捕捉触发输入。	输入端口	P33/RTCDIV/ RTCCCL/BUZ/ INTP2
TI010		16 位定时器/事件计数器 00 的捕捉寄存器(CR000)的捕捉触发 输入。		P34/TI52/TO00/ RTC1HZ/INTP1
TI52	输入	8 位定时器/事件计数器 50 的外部计数时钟输入。	输入端口	P34/TI010/TO00/ RTC1HZ/INTP1
TO00	输出	16 位定时器/事件计数器 00 输出	输入端口	P34/TI52/TI010/ RTC1HZ/INTP1
TOH0	输出	8 位定时器 H0 输出	输入端口	P32/MCGO
TOH1		8 位定时器 H1 输出		P31/INTP3
TxD0	输出	从异步串行接口的串行数据输出	输入端口	P13/KR4/<TxD6>
TxD6				P112/SEG6
<TxD6>				P13/TxD0/KR4
EXLVI	输入	用于外部低电压检测的电压输入	输入端口	P120/INTP0
X1	输入	连接主系统时钟振荡器	输入端口	P121/OCD0A
X2	-			P122/EXCLK/ OCD0B
EXCLK	输入	主系统时钟的外部时钟输入	输入端口	P122/X2/OCD0B
XT1	输入	连接副系统时钟振荡器	输入端口	P123
XT2	-			P124
V <sub>DD</sub>	-	正向电源	-	-
V <sub>SS</sub>	-	地电位	-	-
FLMD0	-	Flash 存储器编程模式设置	-	-
OCD0A	输入	用于设置片上调试模式	输入端口	P121/X1
OCD0B	-			P122/X2/EXCLK

备注 带有箭头(< >) 的功能可以通过设定输入转换控制寄存器(ISC)分配。

## 2.2 引脚功能描述

### 2.2.1 P12, P13 (端口 1)

P12 和 P13 作为 2 位 I/O 端口使用。这些引脚也可用于按键中断和串行接口数据 I/O。P13 可由端口功能寄存器 (PF1) 选择作为引脚功能(参见图 4-19)。

以下操作模式可以用位操作。

#### (1) Port mode

P12 和 P13 作为 2 位 I/O 端口使用。通过使用端口模式寄存器 1 (PM1)，可按位设置 P12 和 P13 为输入输出端口。由上拉电阻选择寄存器 1 (PU1) 定义内置上拉电阻的使用。

#### (2) 控制模式

P12 和 P13 可用于按键中断、串行接口数据 I/O。

##### (a) KR3, KR4

按键中断输入引脚。

##### (b) RxD0, RxD6

异步串行接口的串行数据输入引脚。

##### (c) TxD0, TxD6

异步串行接口的串行数据输出引脚。

### 2.2.2 P20 ~ P25 (端口 2)

P20 ~ P25 作为 6 位 I/O 端口使用。这些引脚也可用于 LCD 控制器/驱动器的 segment 信号输出引脚，10 位次逼近型 A/D 转换器模拟输入(仅适用于  $\mu$ PD78F041x)引脚。可由端口功能寄存器(PF2)选择作为 I/O 端口功能或 segment 信号输出功能。

以下操作模式可以用位操作。

#### (1) 端口模式

P20 ~ P25 作为 6 位 I/O 端口使用。通过使用端口模式寄存器 2 (PM2)，可按位设置 P20 ~ P25 为输入输出端口。

#### (2) 控制模式

P20 ~ P25 可用于 LCD 控制器/驱动器的 segment 信号输出，10 位次逼近型 A/D 转换器模拟输入(仅适用于  $\mu$ PD78F041x)。

##### (a) SEG16 ~ SEG21

这些引脚可用于 LCD 控制器/驱动器的 segment 信号输出。

##### (b) ANI0 ~ ANI5 (仅对 $\mu$ PD78F041x)

这些是 10 位次逼近型 A/D 转换器模拟输入引脚。当使用这些引脚作为模拟输入引脚时，参见 12.6 10 位次逼近型 A/D 转换器的注意事项中的(5) ANI0/SEG21/P20 ~ ANI5/SEG16/P25 引脚。

**注意事项** P20 ~ P25 要在复位释放后设置模拟输入模式。

### 2.2.3 P31 ~ P34 (端口 3)

P31 ~ P34 作为 4 位 I/O 端口使用。这些引脚也可用于外部中断请求输入，定时器 I/O，蜂鸣器输出，实时计数器输出和 manchester 编码输出。

以下操作模式可以用位操作。

#### (1) 端口模式

P31 ~ P34 作为 4 位 I/O 端口使用。通过使用端口模式寄存器 3 (PM3)，可按位设置 P31 ~ P34 为输入输出端口。由上拉电阻选择寄存器 3 (PU3)定义内置上拉电阻的使用。

#### (2) 控制模式

P31 ~ P34 可用于外部中断请求输入，定时器 I/O，蜂鸣器输出，实时计数器输出和 manchester 编码输出。

##### (a) INTP1 ~ INTP3

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。

##### (b) TO00, TOH0, TOH1

定时器输出引脚。

##### (c) TI000

将外部计数时钟输入到 16 位定时器/事件计数器 00，也可将捕捉触发信号输入到 16 位定时器/事件计数器 00 的捕捉寄存器（CR000 或 CR010）中。

##### (d) TI010

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

##### (e) TI52

将外部计数时钟输入到 8 位定时器/事件计数器 52。

##### (f) BUZ

蜂鸣器输出引脚。

##### (g) RTCDIV

实时计数器时钟 (32 kHz 分频) 输出引脚。

##### (h) RTCCL

实时计数器时钟 (32 kHz 振荡) 输出引脚。

##### (i) RTC1HZ

实时计数器修正时钟 (1 kHz 振荡) 输出引脚。

##### (j) MCGO

Manchester 编码输出引脚。

### 2.2.4 P40 (端口 4)

P40 作为 1 位 I/O 端口使用。这些引脚也可用于按键中断输入和 LCD 驱动电源。

以下操作模式可以用位操作。

#### (1) 端口模式

P40 作为 1 位 I/O 端口使用。通过使用端口模式寄存器 4 (PM4)，可按位设置 P40 为输入输出端口。由上拉电阻选择寄存器 4 (PU4)定义内置上拉电阻的使用。

#### (2) 控制模式

P40 可用于按键中断输入和 LCD 驱动电源。

##### (a) KR0

按键中断输入引脚。

##### (b) VLc3

LCD 驱动电源。

### 2.2.5 P100, P101 (端口 10)

P100 和 P101 作为 2 位 I/O 端口使用。这些引脚也可以作为 LCD 控制器/驱动器的 segment 信号输出。由端口功能寄存器 ALL (PFALL)选择作为 I/O 端口或 segment 信号输出。

#### (1) 端口模式

P100 和 P101 作为 2 位 I/O 端口使用。通过使用端口模式寄存器 10 (PM10)，可按位设置 P100 和 P101 为输入输出端口。由上拉电阻选择寄存器 10 (PU10)定义内置上拉电阻的使用。

#### (2) 控制模式

P100 和 P101 可作为 LCD 控制器/驱动器的 segment 信号输出。

##### (a) SEG4, SEG5

LCD 控制器/驱动器的 segment 信号输出。

### 2.2.6 P112, P113 (端口 11)

P112 和 P113 作为 2 位 I/O 端口使用。这些引脚也可以作为 LCD 控制器/驱动器的 segment 信号输出和串行接口数据 I/O。由端口功能寄存器 ALL (PFALL)选择作为 I/O 端口(除了 segment 信号输出)或 segment 信号输出。

#### (1) 端口模式

P112 和 P113 作为 2 位 I/O 端口使用。通过使用端口模式寄存器 11 (PM11)，可按位设置 P112 和 P113 为输入输出端口。由上拉电阻选择寄存器 11 (PU11)定义内置上拉电阻的使用。

#### (2) 控制模式

P112 和 P113 可作为 LCD 控制器/驱动器的 segment 信号输出和串行接口数据 I/O。

**(a) SEG6, SEG7**

LCD 控制器/驱动器的 segment 信号输出。

**(b) RxD6**

串行接口 UART6 的串行数据输入引脚。

**(c) TxD6**

串行接口 UART6 的串行数据输出引脚。

**2.2.7 P120 ~ P124 (端口 12)**

P120 作为 1 位 I/O 端口使用。P121 ~ P124 作为 4 位输入端口使用。这些引脚也可以作为外部中断请求输入，外部低电压检测的电压输入，连接主系统时钟振荡器，连接副系统时钟振荡器和外部时钟输入。以下操作模式可以用位操作。

**(1) 端口模式**

P120 作为 1 位 I/O 端口使用和 P121 ~ P124 作为 4 位输入端口使用。仅 P120，可由端口模式寄存器 12(PM12) 设置为输入或输出端口。仅对于 P120，可由上拉电阻选择寄存器 12(PU12) 定义内置上拉电阻的使用。

**(2) 控制模式**

P120 ~ P124 可作为外部中断请求输入、外部低电压检测的电压输入、连接主系统时钟的振荡器、连接副系统时钟的振荡器和外部时钟输入。

**(a) INTPO**

通过定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），它可作为外部中断请求输入(INTPO)使用。

**(b) EXLVI**

用于外部低电压检测的电压输入。

**(c) X1, X2**

用于连接主系统时钟振荡器。

**(d) EXCLK**

用于主系统时钟的外部时钟输入。

**(e) XT1, XT2**

用于连接副系统时钟振荡器。

**备注** 在使用片上调试功能时，X1 和 X2 能够用于片上调试模式设置引脚 (OCD0A, OCD0B)。详情请参看 第二十五章 片上调试功能。

**2.2.8 P140 ~ P143 (端口 14)**

P140 ~ P143 作为 4 位 I/O 端口使用。这些引脚也可以作为 LCD 控制器/驱动器的 segment 信号输出。由端口功能寄存器 ALL (PFALL) 选择作为 I/O 端口或 segment 信号输出。

**(1) 端口模式**

P140 ~ P143 作为 4 位 I/O 端口使用。通过使用端口模式寄存器 14 (PM14)，可按位设置 P140 ~ P143 为输入输出端口。由上拉电阻选择寄存器 14 (PU14)定义内置上拉电阻的使用。

**(2) 控制模式**

P140 ~ P143 可作为 LCD 控制器/驱动器的 segment 信号输出。

**(a) SEG8 ~ SEG11**

LCD 控制器/驱动器的 segment 信号输出。

**2.2.9 P150 ~ P153 (端口 15)**

P150 ~ P153 作为 4 位 I/O 端口使用。这些引脚也可以作为 LCD 控制器/驱动器的 segment 信号输出。由端口功能寄存器 ALL (PFALL)选择作为 I/O 端口或 segment 信号输出。

**(1) 端口模式**

P150 ~ P153 作为 4 位 I/O 端口使用。通过使用端口模式寄存器 15 (PM15)，可按位设置 P150 ~ P153 为输入输出端口。由上拉电阻选择寄存器 15 (PU15)定义内置上拉电阻的使用。

**(2) 控制模式**

P150 ~ P153 可作为 LCD 控制器/驱动器的 segment 信号输出。

**(a) SEG12 ~ SEG15**

LCD 控制器/驱动器的 segment 信号输出。

**2.2.10 AV<sub>REF</sub> (仅对 $\mu$ PD78F041x)**

10 位逐次逼近型 A/D 转换器参考电压输入引脚，端口 2 的正向电源引脚。  
不使用 A/D 转换器时，将该引脚直接连到 V<sub>DD</sub><sup>※</sup>。

**注** 当至少 1 个端口 2 引脚作为数字端口使用时，应使 AV<sub>REF</sub> 引脚电平与 V<sub>DD</sub> 引脚的电平相同。

**2.2.11 AV<sub>SS</sub> (仅对 $\mu$ PD78F041x)**

作为 A/D 转换器的地引脚。即使在不使用 A/D 转换器时，该引脚电平也始终应与 V<sub>SS</sub> 相同

**2.2.12 COM0 ~ COM7**

这些引脚作为 LCD 控制器/驱动器的 common 信号输出引脚。

**2.2.13 V<sub>LC0</sub> ~ V<sub>LC3</sub>**

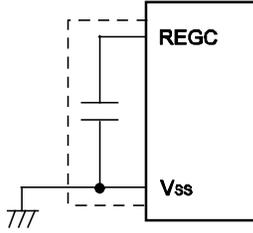
这些引脚作为驱动 LCD 的电源电压引脚。

**2.2.14  $\overline{\text{RESET}}$** 

低电平有效的系统复位输入引脚。

### 2.2.15 REGC

用于内部操作的调节器输出(2.4 V)稳定电容的连接。通过一个电容 (0.47 ~ 1  $\mu$ F : 推荐) 将此引脚连接到 Vss 。



**注意事项** 在上图虚线部分，用尽可能短的线连接。

### 2.2.16 VDD

VDD 为正向供电电源引脚。

### 2.2.17 Vss

Vss 为地引脚。

### 2.2.18 FLMD0

用于 Flash 存储器编程模式设置。

在正常操作模式下将 FLMD0 连接至 Vss 。

在 Flash 存储器编程模式下，将该引脚连接至 Flash 编程器。

## 2.3 引脚 I/O 电路和未使用引脚的建议连接方式

表 2-2 为引脚 I/O 电路类型和未使用引脚的建议连接方式。

参考图 2-1 显示每一种类型的 I/O 电路结构。

表 2-2. 引脚 I/O 电路类型(1/2)

引脚名称	I/O 电路类型	I/O	未使用引脚的建议连接方式
P12/RxD0/KR3/<RxD6>	5-AH	I/O	输入： 通过电阻单独连接至 V <sub>DD</sub> 或 V <sub>SS</sub> 。 输出： 保持开路。
P13/TxD0/KR4/<TxD6>			
P20/SEG21/ANI0 <sup>注1,2</sup> ~ P25/SEG16/ANI5 <sup>注1,2</sup>	17-R		<模拟设置> 连接至 AV <sub>REF</sub> 或 AV <sub>SS</sub> 。 <数字设置> 输入： 通过电阻单独连接至 V <sub>DD</sub> 或 V <sub>SS</sub> 。 <sup>注3</sup> 输出： 保持开路。 <Segment 设置> 保持开路。
P31/TOH1/INTP3	5-AH		输入： 通过电阻单独连接至 V <sub>DD</sub> 或 V <sub>SS</sub> 。 输出： 保持开路。
P32/TOH0/MCGO	5-AG		
P33/TI000/RTCDIV/ RTCCL/BUZ/INTP2	5-AH		
P34/TI52/TI010/TO00/ RTC1HZ/INTP1			
P40/V <sub>LC3</sub> /KR0	5-AO		输入： 通过电阻单独连接至 V <sub>DD</sub> 或 V <sub>SS</sub> 。 输出： 保持开路。
P100/SEG 4, P101/SEG5	17-P		<端口设置> 输入： 通过电阻单独连接至 V <sub>DD</sub> 或 V <sub>SS</sub> 。 输出： 保持开路。 <Segment 设置> 保持开路。
P112/SEG6/TxD6	17-P		<端口设置> 输入： 通过电阻单独连接至 V <sub>DD</sub> 或 V <sub>SS</sub> 。 输出： 保持开路。 <Segment 设置> 保持开路。
P113/SEG7/RxD6	17-Q		

- 注
1. 仅  $\mu$ PD78F041x 提供 ANIX。
  2. 复位释放后，才可设置 P20/SEG21/ANI0 ~ P25/SEG16/ANI5 为数字输入模式。
  3. 对于  $\mu$ PD78F040x，通过电阻单独连接至 V<sub>DD</sub> 或 V<sub>SS</sub>。

备注 箭头(< >)内的功能由输入转换控制寄存器(ISC)分配。

表 2-2. 引脚 I/O 电路类型(2/2)

引脚名称	I/O 电路类型	I/O	未使用引脚的建议连接方式
P120/INTP0/EXLVI	5-AH	I/O	输入： 通过电阻单独连接至 V <sub>DD</sub> 或 V <sub>SS</sub> 。 输出： 保持开路。
P121/X1/OCD0A <sup>注1</sup>	37-A	输入	通过电阻单独连接至 V <sub>DD</sub> 或 V <sub>SS</sub> 。
P122/X2/EXCLK/ OCD0B <sup>注1</sup>			
P123/XT1 <sup>注1</sup>			
P124/XT2 <sup>注1</sup>			
P140/SEG8 ~ P143/SEG11	17-P	I/O	<端口设置> 输入： 通过电阻单独连接至 V <sub>DD</sub> 或 V <sub>SS</sub> 。 输出： 保持开路。 <Segment 设置> 保持开路。
P150/SEG12 ~ P153/SEG15			
COM0 to COM3	18-E	输出	保持开路。
COM4/SEG0 ~ COM7/SEG3	18-F		
V <sub>LC0</sub> to V <sub>LC2</sub>	-		
RESET	2	输入	直接连接或通过电阻连接至 V <sub>DD</sub> 。
FLMD0	38		连接至 V <sub>SS</sub> <sup>注3</sup> 。
AV <sub>REF</sub> <sup>注2</sup>	-	-	直接连接至 V <sub>DD</sub> 。 <sup>注4</sup>
AV <sub>SS</sub> <sup>注2</sup>			直接连接 V <sub>SS</sub> 。

- 注
1. 当不使用这些引脚时，在 I/O 端口模式下使用以上推荐的连接（参看 图 5-2 时钟操作模式选择寄存器 (OSCCTL)的格式）。
  2. 仅适用于  $\mu$ PD78F041x。
  3. 当 flash 存储器写入数据时使用 FLMD0 引脚。当在线重写 flash 存储器数据或执行片上调试，通过电阻连接此引脚到 V<sub>SS</sub> (10 k $\Omega$ : 推荐)。
  4. 当端口 2 作为数字端口使用时，应使它与 V<sub>DD</sub> 引脚的电平相同。

图 2-1. 引脚 I/O 电路列表(1/2)

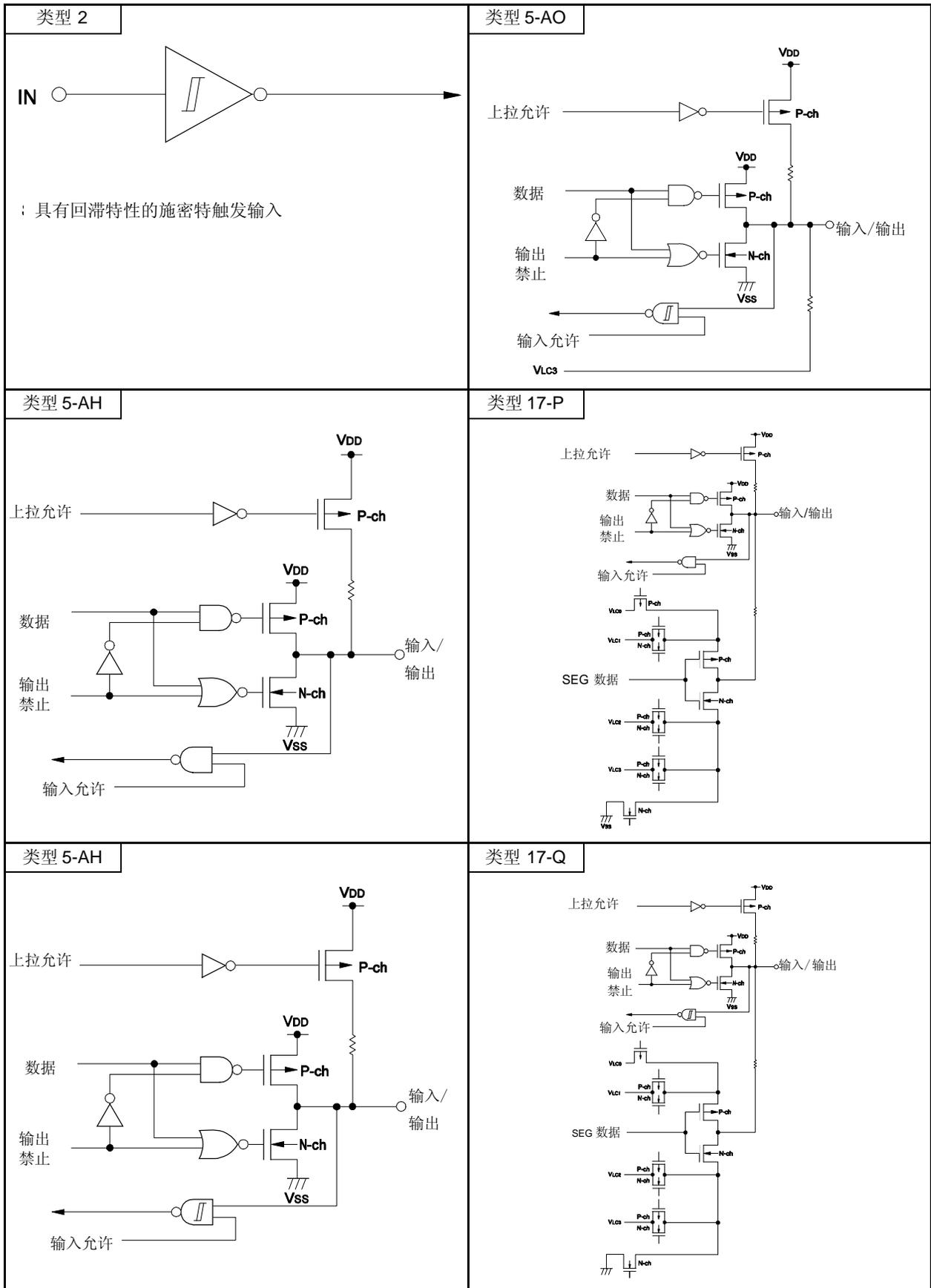
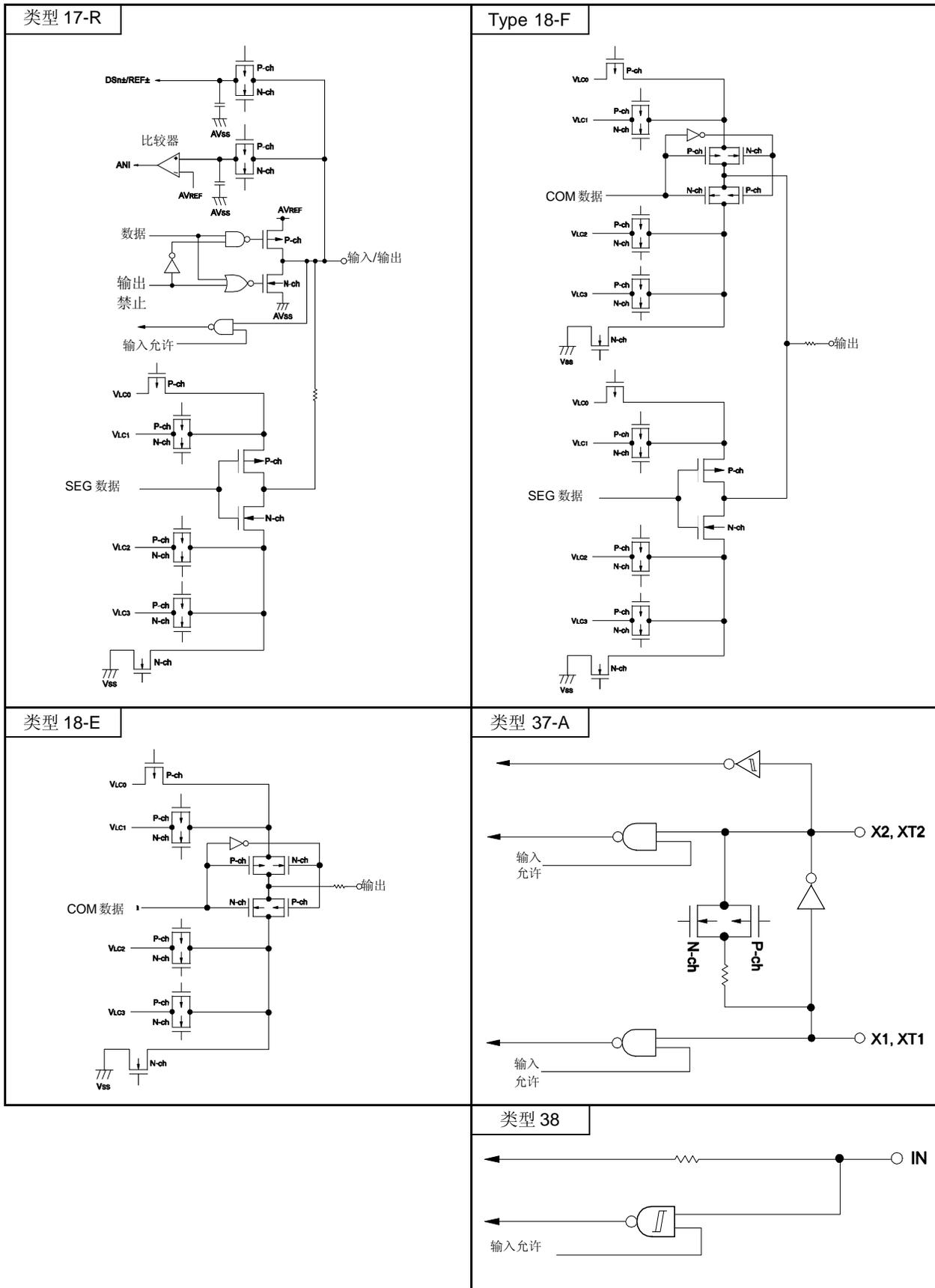


图 2-1. 引脚 I/O 电路列表(2/2)



## 第三章 CPU 结构

### 3.1 存储空间

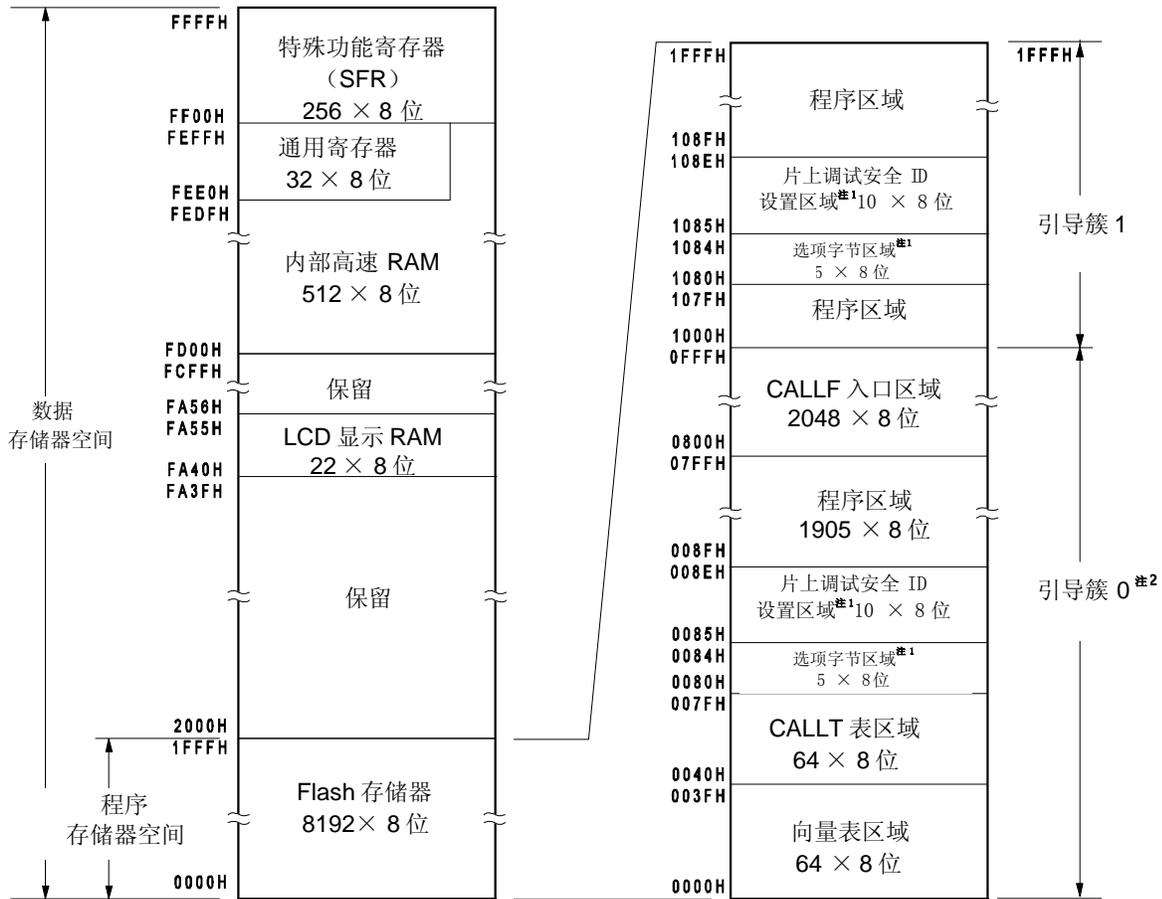
78K0/ LC3 产品可以访问大小为 64KB 的存储空间。图 3-1 ~ 3-4 显示了存储空间映射图。

**注意事项：**不管内部存储器容量有多少，所有 78K0/ LC3 产品的内部存储器容量切换寄存器(IMS)的初始值都是固定的(IMS = CFH)。因此每种产品设置值如下所示。

表 3-1 设置内部存储器容量切换寄存器(IMS) 的值

Flash 存储器版本(78K0/LC3)	IMS	ROM 容量	内部高速 RAM 容量
μPD78F0400, 78F0410	42H	8 KB	512 字节
μPD78F0401, 78F0411	04H	16 KB	768 字节
μPD78F0402, 78F0412	C6H	24 KB	1 KB
μPD78F0403, 78F0413	C8H	32 KB	

图 3-1. 存储空间映射图(μPD78F0400, 78F0410)



- 注
1. 当不使用引导交换时: 设置选项字节为 0080H~0084H, 片上调试安全 ID 为 0085H~008EH。  
当使用引导交换时: 设置选项字节为 0080H~0084H 和 1080H~1084H, 片上调试安全 ID 为 0085H~008EH 和 1085H~108EH。
  2. 根据安全设置可以禁止写入引导簇 0 (参见 24.7 安全设置)。

备注 Flash 存储器被分割成 block (1 个 block = 1 KB)。关于地址值和 block 号, 参见 表 3-2 Flash 存储器中地址值和 Block 号之间的关系。

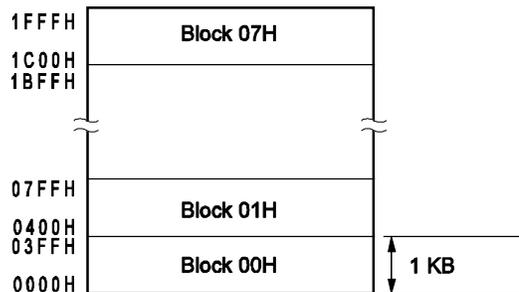
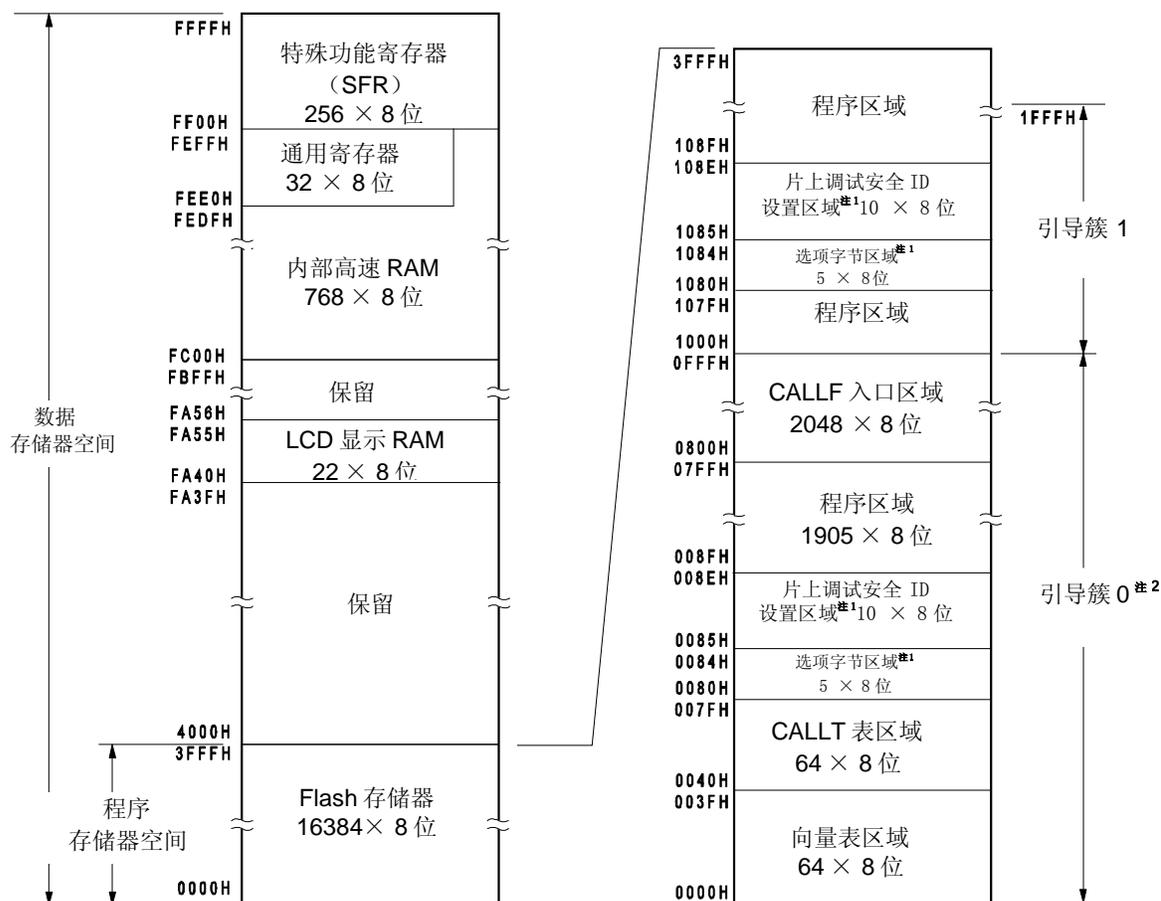


图 3-2. 存储器映射图(μPD78F0401, 78F0411)



- 注
1. 当不使用引导交换时: 设置选项字节为 0080H~0084H, 片上调试安全 ID 为 0085H~008EH。  
当使用引导交换时: 设置选项字节为 0080H~0084H 和 1080H~1084H, 片上调试安全 ID 为 0085H~008EH 和 1085H~108EH。
  2. 根据安全设置可以禁止写入引导簇 0 (参见 24.7 安全设置)。

备注 Flash 存储器被分割成 block (1 个 block = 1 KB)。关于地址值和 block 号, 参见 表 3-2 Flash 存储器中地址值和 Block 号之间的关系。

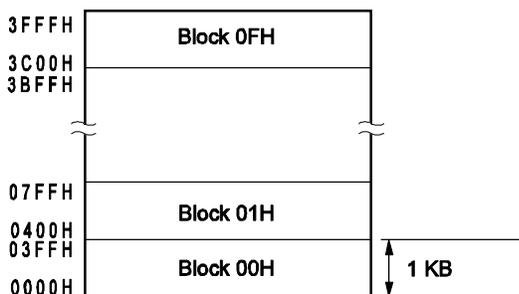
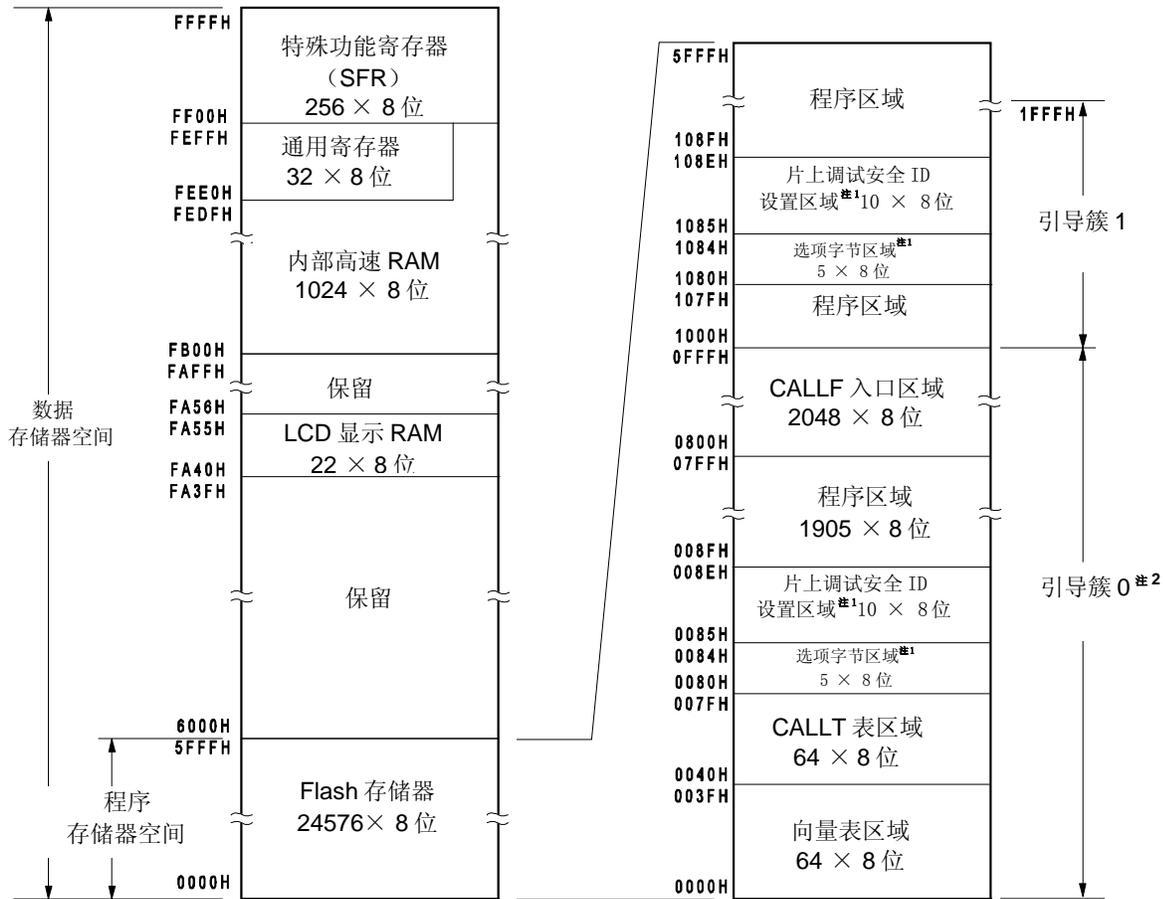


图 3-3. 存储器映射图(μPD78F0402, 78F0412)



- 注
1. 当不使用引导交换时: 设置选项字节为 0080H~0084H, 片上调试安全 ID 为 0085H~008EH。  
当使用引导交换时: 设置选项字节为 0080H~0084H 和 1080H~1084H, 片上调试安全 ID 为 0085H~008EH 和 1085H~108EH。
  2. 根据安全设置可以禁止写入引导簇 0 (参见 24.7 安全设置)。

备注 Flash 存储器被分割成 block (1 个 block = 1 KB)。关于地址值和 block 号, 参见 表 3-2 Flash 存储器中地址值和 Block 号之间的关系。

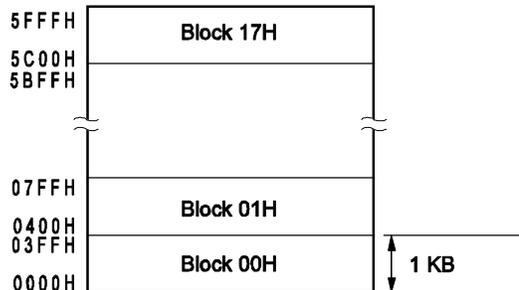
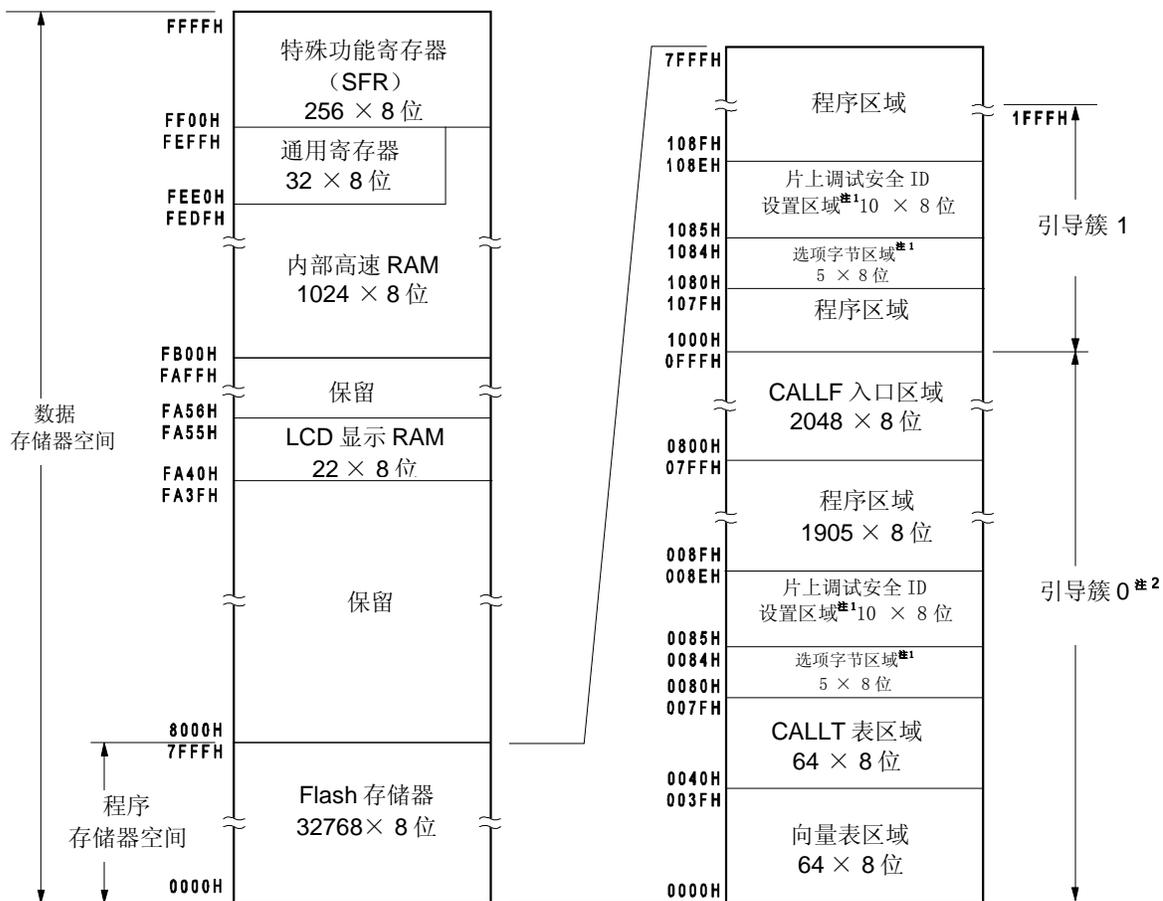
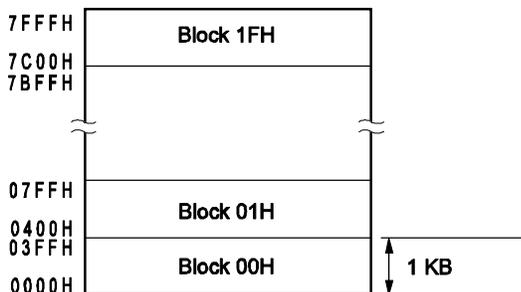


图 3-4. 存储空间映射图( $\mu$ PD78F0403, 78F0413)



- 注
1. 当不使用引导交换时: 设置选项字节为 0080H~0084H, 片上调试安全 ID 为 0085H~008EH。  
当使用引导交换时: 设置选项字节为 0080H~0084H 和 1080H~1084H, 片上调试安全 ID 为 0085H~008EH 和 1085H~108EH。
  2. 根据安全设置可以禁止写入引导簇 0 (参见 24.7 安全设置)。

备注 Flash 存储器被分割成 block (1 个 block = 1 KB)。关于地址值和 block 号, 参见 表 3-2 Flash 存储器中地址值和 Block 号之间的关系。



Flash 存储器中地址值和 Block 号之间的关系如下所示。

表 3-2. Flash 存储器中地址值和 Block 号之间的关系

地址值	Block 号	地址值	Block 号
0000H ~ 03FFH	00H	4000H ~ 43FFH	10H
0400H ~ 07FFH	01H	4400H ~ 47FFH	11H
0800H ~ 0BFFH	02H	4800H ~ 4BFFH	12H
0C00H ~ 0FFFH	03H	4C00H ~ 4FFFH	13H
1000H ~ 13FFH	04H	5000H to 53FFH	14H
1400H ~ 17FFH	05H	5400H ~ 57FFH	15H
1800H ~ 1BFFH	06H	5800H ~ 5BFFH	16H
1C00H ~ 1FFFH	07H	5C00H ~ 5FFFH	17H
2000H ~ 23FFH	08H	6000H ~ 63FFH	18H
2400H ~ 27FFH	09H	6400H ~ 67FFH	19H
2800H ~ 2BFFH	0AH	6800H ~ 6BFFH	1AH
2C00H ~ 2FFFH	0BH	6C00H ~ 6FFFH	1BH
3000H ~ 33FFH	0CH	7000H ~ 73FFH	1CH
3400H ~ 37FFH	0DH	7400H ~ 77FFH	1DH
3800H ~ 3BFFH	0EH	7800H ~ 7BFFH	1EH
3C00H ~ 3FFFH	0FH	7C00H ~ 7FFFH	1FH

备注  $\mu$ PD78F0400, 78F0410: Block 号 00H ~ 07H  
 $\mu$ PD78F0401, 78F0411: Block 号 00H ~ 0FH  
 $\mu$ PD78F0402, 78F0412: Block 号 00H ~ 17H  
 $\mu$ PD78F0403, 78F0413: Block 号 00H ~ 1FH

### 3.1.1 内部程序存储器空间

内部程序存储器空间用于存储程序和表数据，一般通过程序计数器 (PC) 来寻址。

78K0/LC3 产品内部 ROM (Flash 存储器) 的情况如下表所示。

表 3-3. 内部 ROM 容量

产品型号	内部 ROM	
	结构	容量
$\mu$ PD78F0400, 78F0410	Flash 存储器	8192 $\times$ 8 位 (0000H ~ 1FFFH)
$\mu$ PD78F0401, 78F0411		16384 $\times$ 8 位 (0000H ~ 3FFFH)
$\mu$ PD78F0402, 78F0412		24576 $\times$ 8 位 (0000H ~ 5FFFH)
$\mu$ PD78F0403, 78F0413		32768 $\times$ 8 位 (0000H ~ 7FFFH)

内部程序存储器空间主要分为以下几个区域。

#### (1) 向量表区域

从 0000H 到 003FH 总共 64 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。

在 16 位地址中，低 8 位是偶地址，高 8 位是奇地址。

表 3-4. 向量表

向量表地址	中断源	向量表地址	中断源
0000H	RESET 输入, POC, LVI, WDT	0020H	INTTM000
0004H	INTLVI	0022H	INTTM010
0006H	INTP0	0024H <sup>注1</sup>	INTAD <sup>注</sup>
0008H	INTP1	0026H	INTSR0
000AH	INTP2	0028H	INTRTC
000CH	INTP3	002AH	INTTM51
0012H	INTSRE6	002CH	INTKR
0014H	INTSR6	002EH	INTRTCI
0016H	INTST6	0032H	INTTM52
0018H	INTST0	0034H	INTTMH2
001AH	INTTMH1	0036H	INTMCG
001CH	INTTMH0	003EH	BRK
001EH	INTTM50		

注 仅对  $\mu$ PD78F041x。

**(2) CALLT 指令表区域**

0040H 至 007FH 共 64 字节的区域，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**

0080H ~ 0084H 和 1080H ~ 1084H 各 5 字节的区域可以用作选项字节区域。当不使用引导交换功能时，在 0080H ~ 0084H 设置选项字节；而当使用引导交换功能时，在 0080H ~ 0084H 和 1080H ~ 1084H 设置选项字节。详细信息参见 **第二十三章 选项字节**。

**(4) CALLF 指令入口区域**

0800H ~ 0FFFH 的区域，用于子程序的直接调用，通过一个 2 字节的调用指令(CALLF)实现。

**(5) 片上调试安全 ID 设置区域**

0085H ~ 008EH 和 1085H ~ 108EH 各 10 字节区域可用作片上调试安全 ID 设置区域。不使用引导交换功能时，在 0085H ~ 008EH 区域设置片上调试安全 ID；而当使用引导交换功能时，在 0085H ~ 008EH 和 1085H ~ 108EH 区域设置。详细情况请参见 **第二十五章 片上调试功能**。

### 3.1.2 内部数据存储器空间

78K0/LC3 产品包括以下几种 RAM。

#### (1) 内部高速 RAM

表 3-5. 内部高速 RAM 容量

产品型号	内部高速 RAM
$\mu$ PD78F0400, 78F0410	512 × 8 位 (FD00H ~ FEFFH)
$\mu$ PD78F0401, 78F0411	768 × 8 位 (FC00H ~ FEFFH)
$\mu$ PD78F0402, 78F0412	1024 × 8 位 (FB00H ~ FEFFH)
$\mu$ PD78F0403, 78F0413	

这个区域不可以用作写入和执行指令的程序区域。

内部高速 RAM 还可以作为堆栈存储器使用。

#### (2) LCD 显示 RAM

LCD 显示 RAM (22 × 8 位 (FA40H ~ FA55H))是与 LCD 控制器/驱动器组合在一起的 (参见图 15.5 LCD 显示 RAM)。

### 3.1.3 特殊功能寄存器(SFR)区域

片内外围硬件的特殊功能寄存器 (SFR)被分配在 FF00H ~ FFFFH 的区域 (参见 3.2.3 特殊功能寄存器(SFR)中表 3-6 特殊功能寄存器列表)。

**注意事项：** 不要访问那些未分配特殊功能寄存器的地址区域。

### 3.1.4 数据存储空间寻址

寻址是定位地址的方式，需要定位的地址包括下一条指令地址或者与指令执行相关的存储器地址或寄存器地址。

基于可操作性和其他考虑，在 78K0/LC3 中提供了几种用于与指令执行相关的存储器寻址方式。因为有专门的数据存储区域，故可使用一些特殊的寻址方式，具有特殊功能寄存器(SFR)和通用寄存器的功能。图 3-5~3-8 显示了数据存储空间与寻址方式的对应关系。如需了解每种寻址方式的详细内容，参见 3.4 操作数地址寻址。

图 3-5. 数据存储空间与寻址方式的对应关系( $\mu$ PD78F0400, 78F0410)

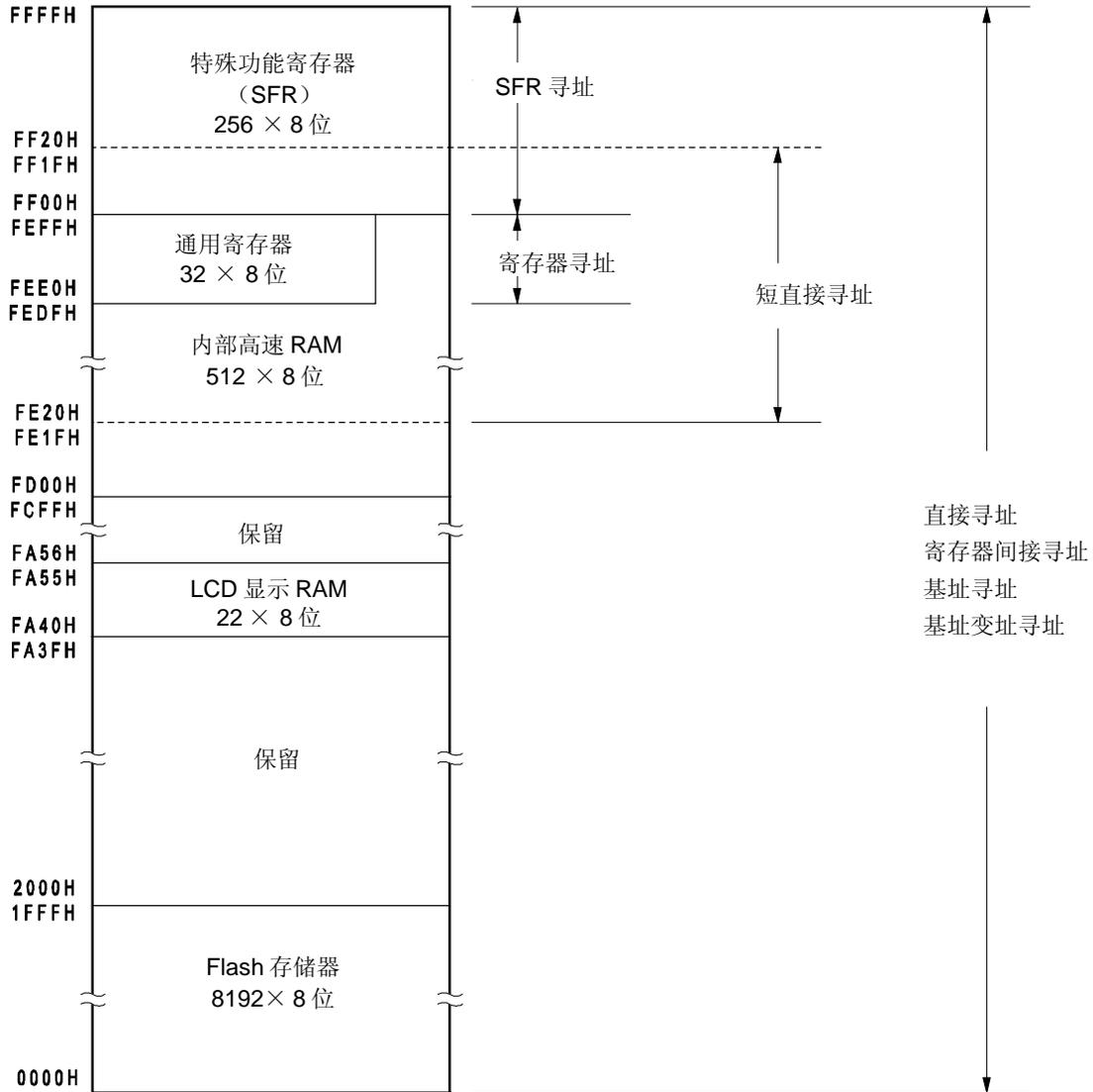


图 3-6. 数据存储空间与寻址方式的对应关系( $\mu$ PD78F0401, 78F0411)

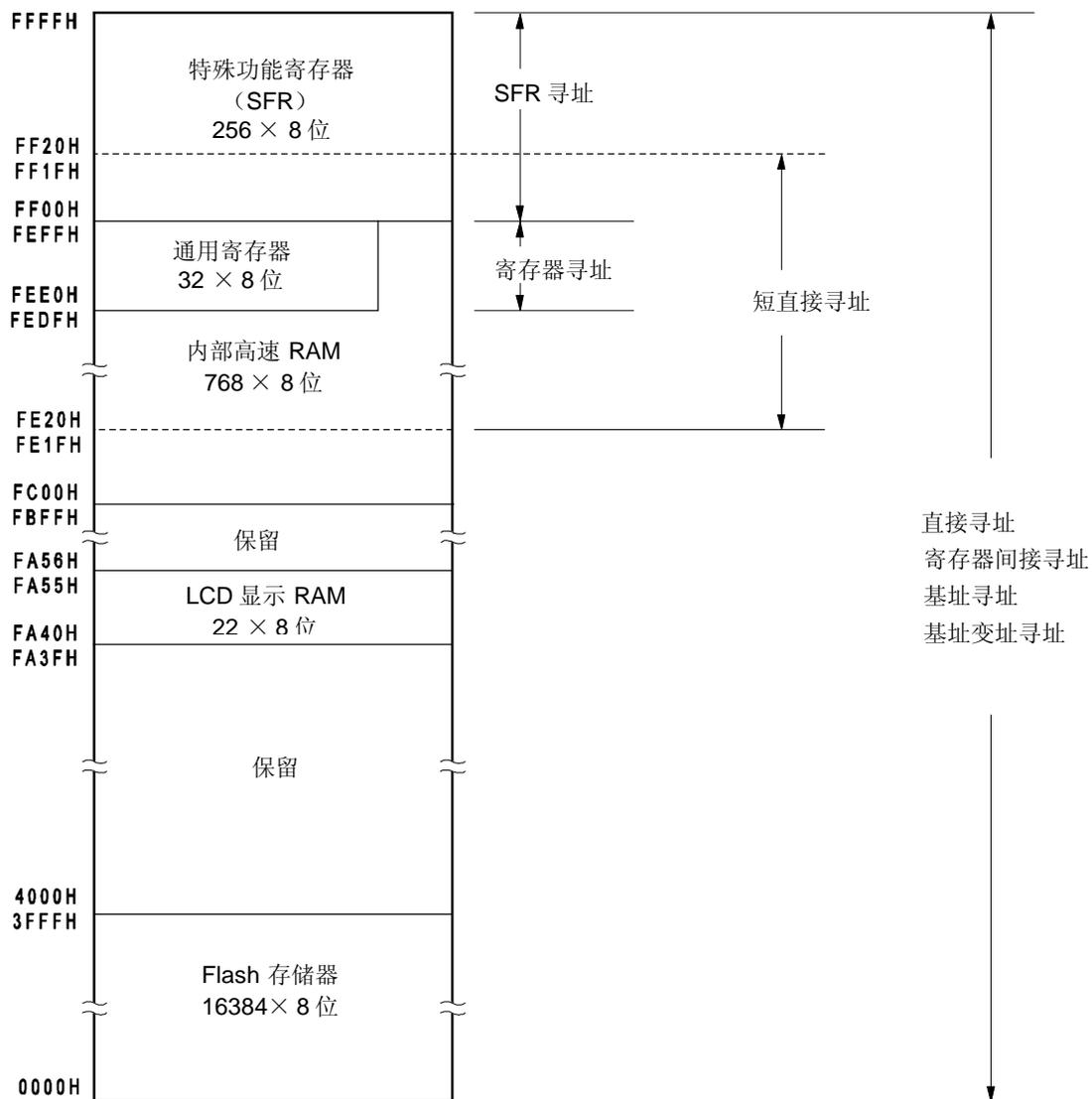


图 3-7. 数据存储空间与寻址方式的对应关系( $\mu$ PD78F0402, 78F0412)

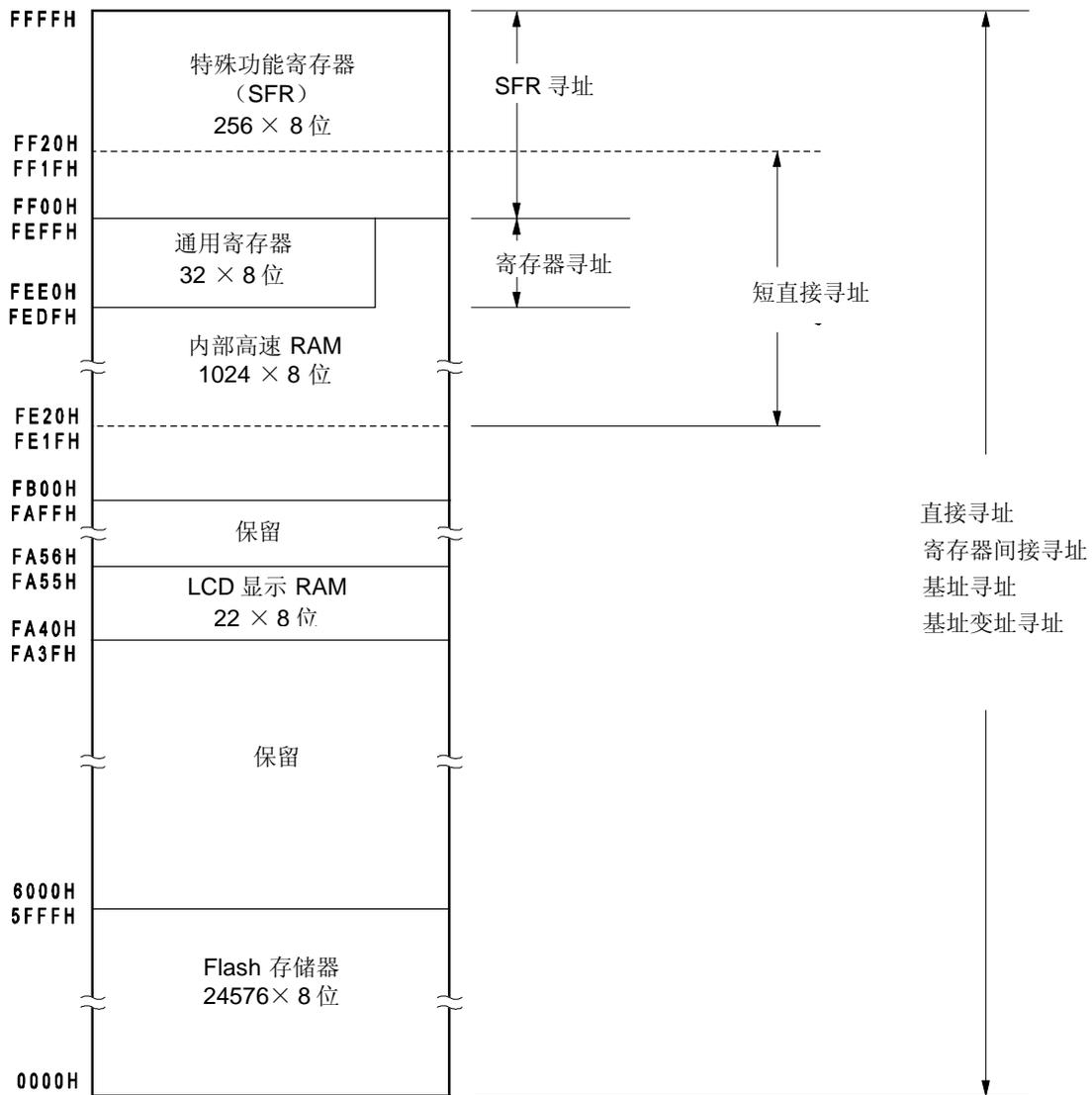
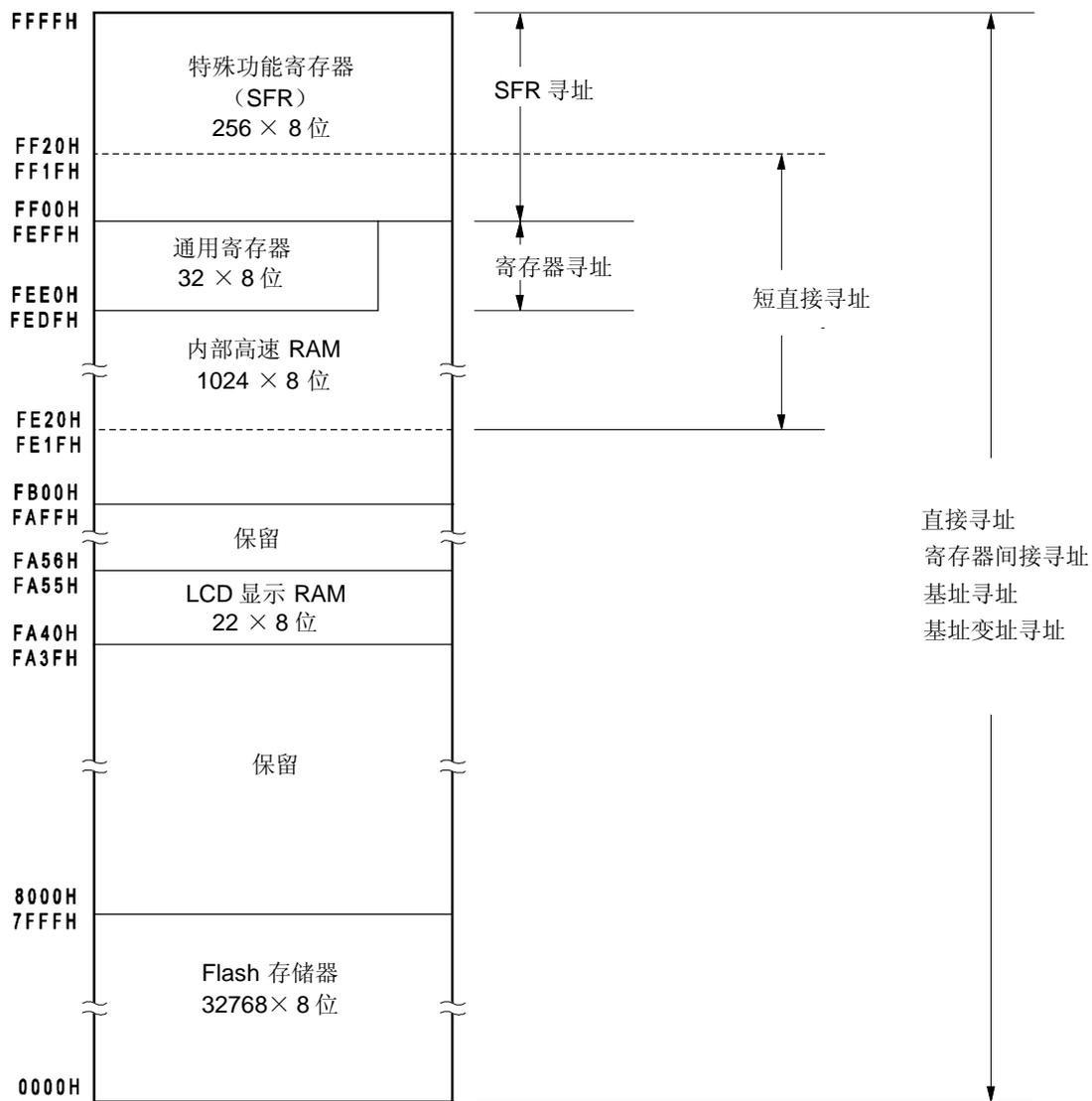


图 3-8. 数据存储空间与寻址方式的对应关系( $\mu$ PD78F0403, 78F0413)



## 3.2 处理器寄存器

78K0/LC3 产品包含以下几种处理器寄存器。

### 3.2.1 控制寄存器

控制寄存器用于控制程序执行的顺序、状态和堆栈空间。程序计数器(PC)、程序状态字(PSW)和堆栈指针寄存器(SP)都属于控制寄存器。

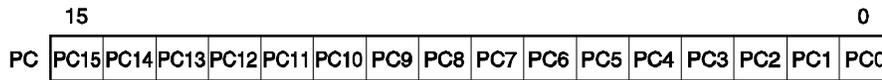
#### (1) 程序计数器 (PC)

程序计数器是一个 16 位寄存器，用于存放下一条即将要执行的指令的地址。

在正常情况下，根据获取的指令字节数，程序计数器(PC)的值会自动累加。当执行分支指令时，则设置立即数和寄存器内容。

复位信号的产生将复位向量表中地址为 0000H 和 0001H 中的值赋给程序计数器。

图 3-9. 程序计数器格式



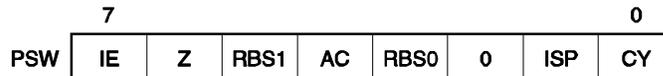
#### (2) 程序状态字 (PSW)

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

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

复位信号的产生将程序状态字的内容设置为 02H。

图 3-10 程序状态字格式



##### (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 为 0 时，表示不允许中断(DI)，即禁止所有可屏蔽中断请求。

当 IE 为 1 时，表示允许中断(EI)，通过优先服务标志(ISP)、用于各种中断源的中断屏蔽标志以及优先级规定标志来完成响应中断请求的控制。

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

**(b) 零标志 (Z)**

当操作结果为 0 时，该标志置 1，其他情况置 0。

**(c) 寄存器组选择标志 (RBS0 和 RBS1)**

寄存器组选择标志有两位，用于选择四组寄存器中的一组。

标志位中存储的信息用来指明执行 SEL Rbn 指令时所选择的寄存器组。

**(d) 半进位标志 (AC)**

如果操作结果中第 3 位有进位或是在第 3 位上有借位，则该标志置 1。其他情况该标志置 0。

**(e) 优先服务标志 (ISP)**

该标志用来管理可屏蔽向量中断响应的优先级。当 ISP 为 0 时，由优先级指定标志寄存器(PR0L, PR0H, PR1L, PR1H) (参见 17.3 (3) 优先级指定标志寄存器(PR0L, PR0H, PR1L, PR1H))指定为低优先级的向量中断请求被禁止响应。对请求的实际响应是由中断允许标志(IE)的状态控制的。

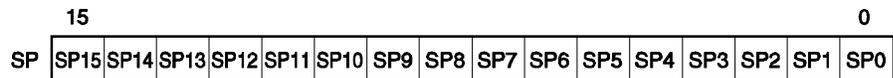
**(f) 进位标志(CY)**

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

**(3) 堆栈指针 (SP)**

这是一个 16 位的寄存器，用来存放存储器堆栈区的起始地址。只有内部高速 RAM 区域才能被设置为堆栈区。

图 3-11 堆栈指针格式



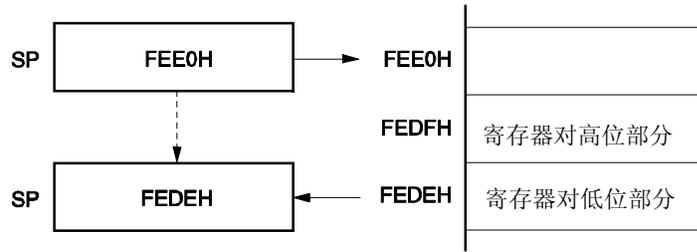
在向堆栈写(存)数据时，堆栈指针 SP 递减，而从堆栈中读出(恢复)数据时，堆栈指针累加。

堆栈的数据存储/恢复操作过程如图 3-12 和 3-13 所示。

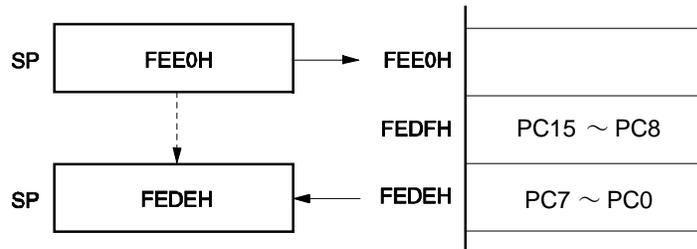
**注意事项：**由于复位信号产生时，SP 的内容不确定，所以在使用堆栈前必须先对 SP 初始化。

图 3-12. 将数据存入堆栈

(a) PUSH rp 指令(当 SP = FEE0H)



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



(c) Interrupt, BRK 指令(当 SP = FEE0H)

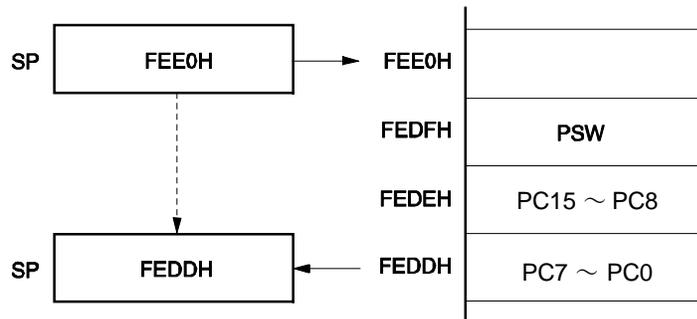
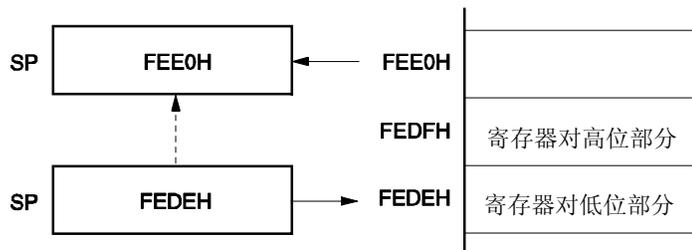
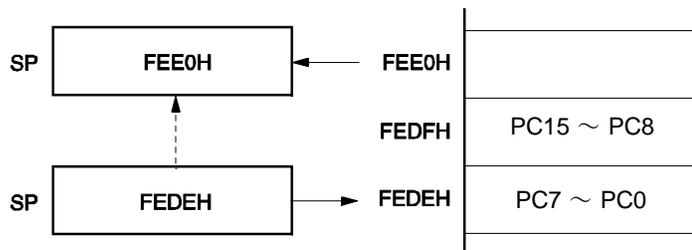


图 3-13. 从堆栈读出数据

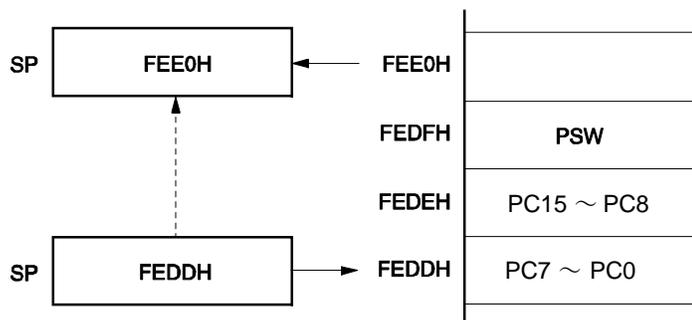
(a) POP rp 指令(当 SP = FEDEH)



(b) RET 指令(当 SP = FEDEH)



(c) RETI, RETB 指令(当 SP = FEDDH)



### 3.2.2 通用寄存器

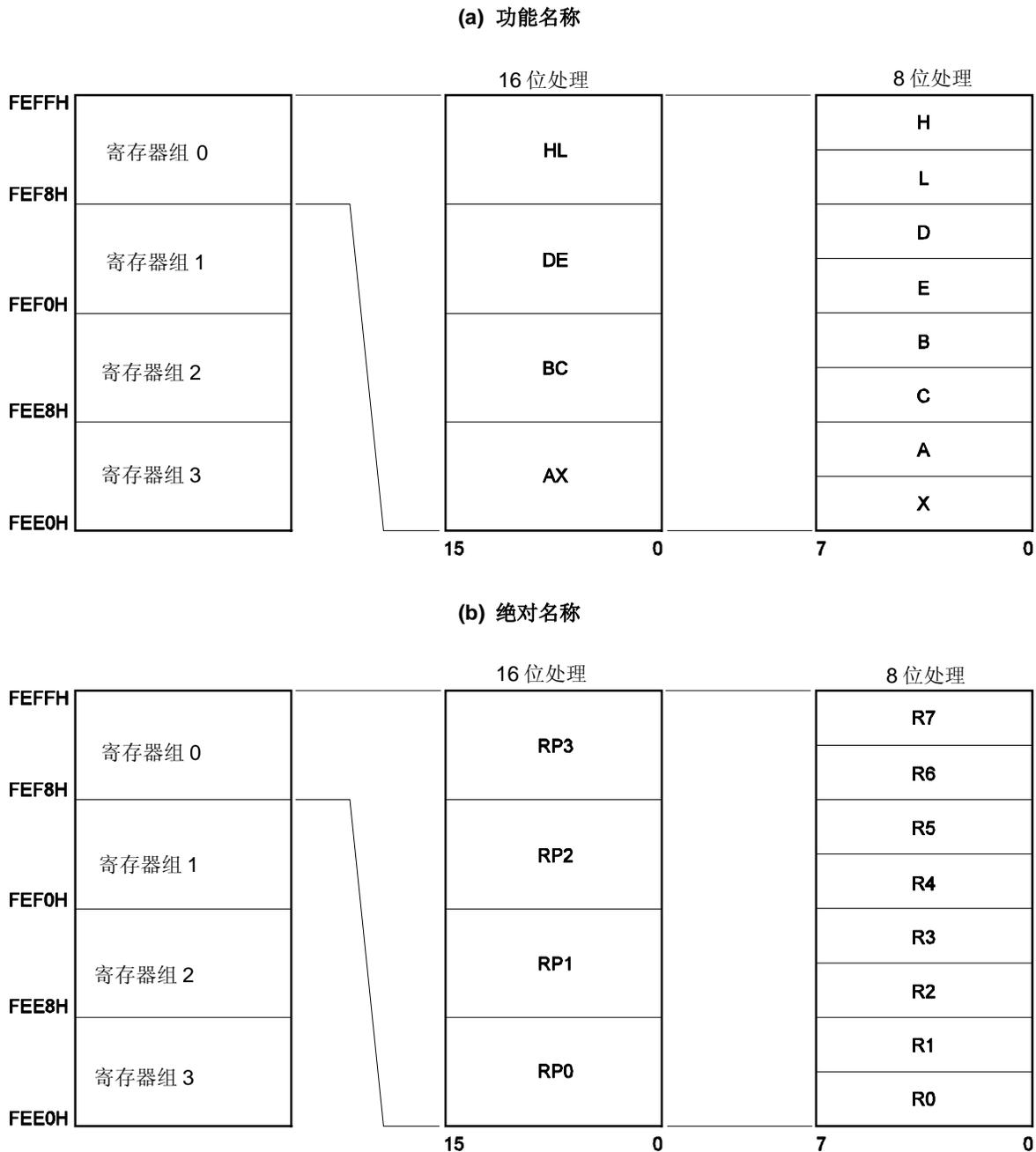
通用寄存器映射到数据存储器特定的地址空间为 FEE0H ~ FEFFH。通用寄存器共有四组，每一组由 8 个 8 位寄存器(X, A, C, B, E, D, L 和 H)组成。

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

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

用于指令执行的寄存器组由 CPU 控制指令(SEL Rn)来设置。由于 4 个寄存器组的结构，通过一个用于正常处理的寄存器和另一个用于中断处理的寄存器之间的切换，可以创建一个高效率的程序。

图 3-14 通用寄存器结构



### 3.2.3 特殊功能寄存器 (SFR)

与通用寄存器不同，每个特殊功能寄存器都有特定的功能。

特殊寄存器指定在 CPU 的 FF00H 到 FFFFH 区域和 LCD 控制器/驱动器 LCDCTL 的 00H 到 03H 区域。

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

每种位单元操作的描述如下。

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

表 3-6 为特殊功能寄存器列表。表中术语的含义如下。

- 符号  
表示特殊功能寄存器地址的符号，在 RA78K0 中是保留字，并使用 CC78K0 中的 #pragma sfr 指令定义为一个 sfr 变量。在使用 RA78K0, ID78K0-QB 和 SM+时，可以将符号作为指令操作数执行写操作。
- 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
- 可操作的位单元  
表示可操作的位单元 (1, 8, 或 16)。“-”表示不可操作的位单元。
- 复位后  
表示复位信号产生后每个寄存器的状态

表 3-6 特殊功能寄存器列表(1/4)

地址	特殊功能寄存器(SFR)名称	符号	R/W	可操作位单元			复位后
				1 位	1 位	1 位	
FF00H	接收缓冲寄存器 6	RXB6	R	-	√	-	FFH
FF01H	端口寄存器 1	P1	R/W	√	√	-	00H
FF02H	端口寄存器 2	P2	R/W	√	√	-	00H
FF03H	端口寄存器 3	P3	R/W	√	√	-	00H
FF04H	端口寄存器 4	P4	R/W	√	√	-	00H
FF05H	发送缓冲寄存器 6	TXB6	R/W	-	√	-	FFH
FF06H	A/D 转换结果寄存器 <sup>※</sup>	ADCR	R	-	-	√	0000H
FF07H	A/D 转换结果寄存器 H <sup>※</sup>	ADCRH	R	-	√	-	00H
FF0AH	端口寄存器 8	P10	R/W	√	√	-	00H
FF0BH	端口寄存器 10	P11	R/W	√	√	-	00H
FF0CH	端口寄存器 11	P12	R/W	√	√	-	00H
FF0EH	端口寄存器 12	P14	R/W	√	√	-	00H
FF0FH	端口寄存器 14	P15	R/W	√	√	-	00H
FF10H	16 位定时器计数器 00	TM00	R	-	-	√	0000H
FF11H							
FF12H	16 位定时器捕获/比较寄存器 000	CR000	R/W	-	-	√	0000H
FF13H							
FF14H	16 位定时器捕获/比较寄存器 010	CR010	R/W	-	-	√	0000H
FF15H							
FF16H	8 位定时器计数器 50	TM50	R	-	√	-	00H
FF17H	8 位定时器比较寄存器 50	CR50	R/W	-	√	-	00H
FF18H	8 位定时器 H 比较寄存器 00	CMP00	R/W	-	√	-	00H
FF19H	8 位定时器 H 比较寄存器 10	CMP10	R/W	-	√	-	00H
FF1AH	8 位定时器 H 比较寄存器 01	CMP01	R/W	-	√	-	00H
FF1BH	8 位定时器 H 比较寄存器 11	CMP11	R/W	-	√	-	00H
FF20H	端口功能寄存器 1	PF1	R/W	√	√	-	00H
FF21H	端口模式寄存器 1	PM1	R/W	√	√	-	FFH
FF22H	端口模式寄存器 2	PM2	R/W	√	√	-	FFH
FF23H	端口模式寄存器 3	PM3	R/W	√	√	-	FFH
FF24H	端口模式寄存器 4	PM4	R/W	√	√	-	FFH
FF2AH	端口模式寄存器 8	PM10	R/W	√	√	-	FFH
FF2BH	端口模式寄存器 10	PM11	R/W	√	√	-	FFH
FF2CH	端口模式寄存器 11	PM12	R/W	√	√	-	FFH
FF2EH	端口模式寄存器 12	PM14	R/W	√	√	-	FFH
FF2FH	端口模式寄存器 14	PM15	R/W	√	√	-	FFH
FF30H	内部高速振荡调整寄存器	HIOTRM	R/W	-	√	-	10H
FF31H	上拉电阻选择寄存器 1	PU1	R/W	√	√	-	00H
FF33H	上拉电阻选择寄存器 3	PU3	R/W	√	√	-	00H
FF34H	端口模式寄存器 4	PU4	R/W	√	√	-	00H
FF3AH	端口模式寄存器 10	PU10	R/W	√	√	-	00H
FF3BH	端口模式寄存器 11	PU11	R/W	√	√	-	00H

注 仅对 $\mu$ PD78F041x。

表 3-6 特殊功能寄存器列表(2/4)

地址	特殊功能寄存器(SFR)名称	符号	R/W	可操作位单元			复位后
				1 位	8 位	16 位	
FF3CH	上拉电阻选择寄存器 12	PU12	R/W	√	√	–	00H
FF3EH	上拉电阻选择寄存器 14	PU14	R/W	√	√	–	00H
FF3FH	上拉电阻选择寄存器 15	PU15	R/W	√	√	–	00H
FF40H	时钟输出选择寄存器	CKS	R/W	√	√	–	00H
FF41H	8 位定时器比较寄存器 51	CR51	R/W	–	√	–	00H
FF42H	8 位定时器 H 模式寄存器 2	TMHMD2	R/W	√	√	–	00H
FF43H	8 位定时器模式控制寄存器 51	TMC51	R/W	√	√	–	00H
FF44H	8 位定时器 H 比较寄存器 02	CMP02	R/W	–	√	–	00H
FF45H	8 位定时器 H 比较寄存器 12	CMP12	R/W	–	√	–	00H
FF47H	MCG 状态寄存器	MC0STR	R	√	√	–	00H
FF48H	外部中断上升沿使能寄存器	EGP	R/W	√	√	–	00H
FF49H	外部中断下降沿使能寄存器	EGN	R/W	√	√	–	00H
FF4AH	MCG 发送缓冲寄存器	MC0TX	R/W	–	√	–	FFH
FF4BH	MCG 发送计数规格寄存器	MC0BIT	R/W	–	√	–	07H
FF4CH	MCG 控制寄存器 0	MC0CTL0	R/W	√	√	–	10H
FF4DH	MCG 控制寄存器 1	MC0CTL1	R/W	–	√	–	00H
FF4EH	MCG 控制寄存器 2	MC0CTL2	R/W	–	√	–	1FH
FF4FH	输入切换控制寄存器	ISC	R/W	√	√	–	00H
FF50H	异步串行接口操作模式寄存器 6	ASIM6	R/W	√	√	–	01H
FF51H	8 位定时器计数器 52	TM52	R	–	√	–	00H
FF53H	异步串行接口接收错误状态寄存器 6	ASIS6	R	–	√	–	00H
FF54H	实时计数器时钟选择寄存器	RTCCL	R/W	√	√	–	00H
FF55H	异步串行接口传送状态寄存器 6	ASIF6	R	–	√	–	00H
FF56H	时钟选择寄存器 6	CKSR6	R/W	–	√	–	00H
FF57H	波特率发生器控制寄存器 6	BRGC6	R/W	–	√	–	FFH
FF58H	异步串行接口控制寄存器 6	ASICL6	R/W	√	√	–	16H
FF59H	8 位定时器比较寄存器 52	CR52	R/W	–	√	–	00H
FF5BH	定时器时钟选择寄存器 52	TCL52	R/W	√	√	–	00H
FF5CH	8 位定时器模式控制寄存器 52	TMC52	R/W	√	√	–	00H
FF60H FF61H	副计数寄存器	RSUBC	R	–	–	√	0000H
FF62H	秒计数寄存器	SEC	R/W	–	√	–	00H
FF63H	分钟计数寄存器	MIN	R/W	–	√	–	00H
FF64H	小时计数寄存器	HOUR	R/W	–	√	–	12H
FF65H	星期计数寄存器	WEEK	R/W	–	√	–	00H

表 3-6 特殊功能寄存器列表(3/4)

地址	特殊功能寄存器(SFR)名称	符号	R/W	可操作位单元			复位后
				1 位	8 位	16 位	
FF66H	日计数器	DAY	R/W	-	√	-	01H
FF67H	月计数器	MONTH	R/W	-	√	-	01H
FF68H	年计数器	YEAR	R/W	-	√	-	00H
FF69H	8 位定时器 H 模式寄存器 0	TMHMD0	R/W	√	√	-	00H
FF6AH	定时器时钟选择寄存器 50	TCL50	R/W	√	√	-	00H
FF6BH	8 位定时器模式控制寄存器 50	TMC50	R/W	√	√	-	00H
FF6CH	8 位定时器 H 模式寄存器 1	TMHMD1	R/W	√	√	-	00H
FF6DH	8 位定时器 H 载波控制寄存器 1	TMCYC1	R/W	√	√	-	00H
FF6EH	按键返回模式寄存器	KRM	R/W	√	√	-	00H
FF6FH	8 位定时器计数器 51	TM51	R	-	√	-	00H
FF70H	异步串行接口操作模式寄存器 0	ASIM0	R/W	√	√	-	01H
FF71H	波特率发生器控制寄存器 0	BRGC0	R/W	-	√	-	1FH
FF72H	接收缓冲寄存器 0	RXB0	R	-	√	-	FFH
FF73H	异步串行接口接收错误状态寄存器 0	ASIS0	R	-	√	-	00H
FF74H	发送移位寄存器 0	TXS0	W	-	√	-	FFH
FF82H	钟表错误修正寄存器	SUBCUD	R/W	√	√	-	00H
FF86H	闹表分钟寄存器	ALARMWM	R/W	-	√	-	00H
FF87H	闹表小时寄存器	ALARMWH	R/W	-	√	-	12H
FF88H	闹表星期寄存器	ALARMWW	R/W	-	√	-	00H
FF89H	实时计数器控制寄存器 0	RTCC0	R/W	√	√	-	00H
FF8AH	实时计数器控制寄存器 1	RTCC1	R/W	√	√	-	00H
FF8BH	实时计数器控制寄存器 2	RTCC2	R/W	√	√	-	00H
FF8CH	定时器时钟选择寄存器 51	TCL51	R/W	√	√	-	00H
FF8DH	A/D 转换器模式寄存器 <sup>注 1</sup>	ADM	R/W	√	√	-	00H
FF8EH	指定模拟输入通道寄存器 <sup>注 1</sup>	ADS	R/W	√	√	-	00H
FF8FH	A/D 端口配置寄存器 0 <sup>注 1</sup>	ADPC0	R/W	√	√	-	08H
FF99H	看门狗定时器使能寄存器	WDTE	R/W	-	√	-	<sup>注 2</sup> 1AH/9AH
FF9FH	时钟操作模式选择寄存器	OSCCTL	R/W	√	√	-	00H
FFA0H	内部振荡模式寄存器	RCM	R/W	√	√	-	80H <sup>注 3</sup>
FFA1H	主时钟模式寄存器	MCM	R/W	√	√	-	00H
FFA2H	主 OSC 控制寄存器	MOC	R/W	√	√	-	80H
FFA3H	振荡稳定时间计数器状态寄存器	OSTC	R	√	√	-	00H
FFA4H	振荡稳定时间选择寄存器	OSTS	R/W	-	√	-	05H
FFACH	复位控制标志寄存器	RESF	R	-	√	-	00H <sup>注 4</sup>

- 注
1. 仅对 $\mu$ PD78F041x。
  2. WDTE 复位值由选项字节设置决定。
  3. 此寄存器值在复位释放后立即为 00H，但当高速内部振荡器振荡精确稳定后自动变为 80H。
  4. RESF 复位值根据复位源变动。

表 3-6 特殊功能寄存器列表(4/4)

地址	特殊功能寄存器(SFR)名称	符号	R/W	可操作位单元			复位后	
				1 Bit	8 Bits	16 Bits		
FFB0H	LCD 模式寄存器	LCDMD	R/W	√	√	–	00H	
FFB1H	LCD 显示模式寄存器	LCDM	R/W	√	√	–	00H	
FFB2H	LCD 时钟控制寄存器 0	LCDC0	R/W	√	√	–	00H	
FFB5H	端口功能寄存器 2	PF2	R/W	√	√	–	00H	
FFB6H	端口功能寄存器 ALL	PFALL	R/W	√	√	–	00H	
FFBAH	16 位定时器模式控制寄存器 00	TMC00	R/W	√	√	–	00H	
FFBBH	预分频模式寄存器 00	PRM00	R/W	√	√	–	00H	
FFBCH	捕捉/比较控制寄存器 00	CRC00	R/W	√	√	–	00H	
FFBDH	16 位定时器输出控制寄存器 00	TOC00	R/W	√	√	–	00H	
FFBEH	低电压检测寄存器	LVIM	R/W	√	√	–	00H <sup>注1</sup>	
FFBFH	低电压检测电平选择寄存器	LVIS	R/W	√	√	–	00H <sup>注1</sup>	
FFE0H	中断请求标志寄存器 0L	IF0	IF0L	R/W	√	√	√	00H
FFE1H	中断请求标志寄存器 0H		IF0H	R/W	√	√		00H
FFE2H	中断请求标志寄存器 1L	IF1	IF1L	R/W	√	√	√	00H
FFE3H	中断请求标志寄存器 1H		IF1H	R/W	√	√		00H
FFE4H	中断屏蔽标志寄存器 0L	MK0	MK0L	R/W	√	√	√	FFH
FFE5H	中断屏蔽标志寄存器 0H		MK0H	R/W	√	√		FFH
FFE6H	中断屏蔽标志寄存器 1L	MK1	MK1L	R/W	√	√	√	FFH
FFE7H	中断屏蔽标志寄存器 1H		MK1H	R/W	√	√		FFH
FFE8H	优先级指定标志寄存器 0L	PR0	PR0L	R/W	√	√	√	FFH
FFE9H	优先级指定标志寄存器 0H		PR0H	R/W	√	√		FFH
FFEAH	优先级指定标志寄存器 1L	PR1	PR1L	R/W	√	√	√	FFH
FFEBH	优先级指定标志寄存器 1H		PR1H	R/W	√	√		FFH
FFF0H	内部存储器容量切换寄存器 <sup>注2</sup>	IMS	R/W	–	√	–	CFH	
FFFBH	处理器时钟控制寄存器	PCC	R/W	√	√	–	01H	

- 注 1. LVIM 和 LVIS 复位值根据复位源变动。
2. 不管内部存储器容量有多少，所有 78K0/ LC3 产品的内部存储器容量切换寄存器(IMS)的初始值都是固定的 (IMS = CFH)。因此每种产品设置值如下所示。

Flash 存储器版本(78K0/LC3)	IMS	ROM 容量	内部高速 RAM 容量
μPD78F0400, 78F0410	42H	8 KB	512 字节
μPD78F0401, 78F0411	04H	16 KB	768 字节
μPD78F0402, 78F0412	C6H	24 KB	1 KB
μPD78F0403, 78F0413	C8H	32 KB	

### 3.3 指令地址寻址

一条指令的地址是由程序计数器(PC)决定的。根据执行指令时所获取的下一条指令字节数，程序计数器(PC)的内容自动增加(每个字节加 1)。在执行转移指令时，将程序计数器(PC)的内容设置为转移目的地址，并按以下寻址方式确定地址。(要了解每条指令的详细信息，请参阅 **78K/0 系列指令用户手册(U12326E)**)。

#### 3.3.1 相对寻址

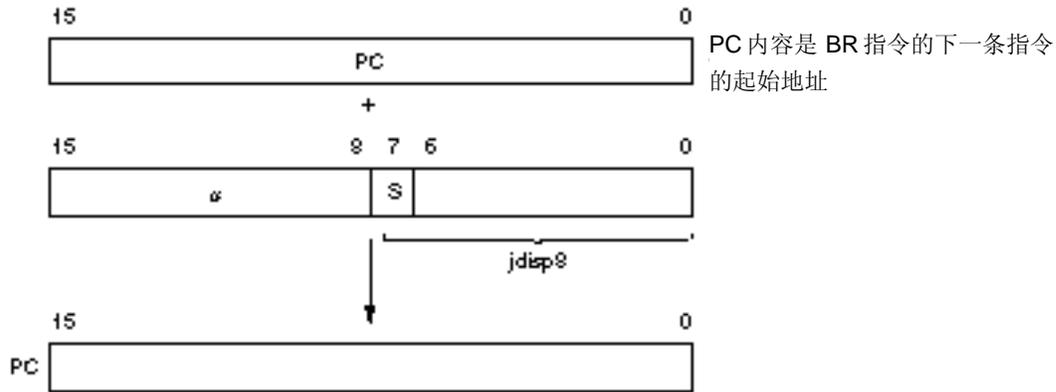
##### [功能]

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数的补码(-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 到+127 之间。

当执行“BR \$addr16”指令或条件转移指令时，将执行相对寻址功能。

##### [图示]



当  $S = 0$ ， $\alpha$  的所有位均为 0

当  $S = 1$ ， $\alpha$  的所有位均为 1

### 3.3.2 立即寻址

**[功能]**

将指令中的立即数赋给程序计数器(PC)，然后转向该地址。

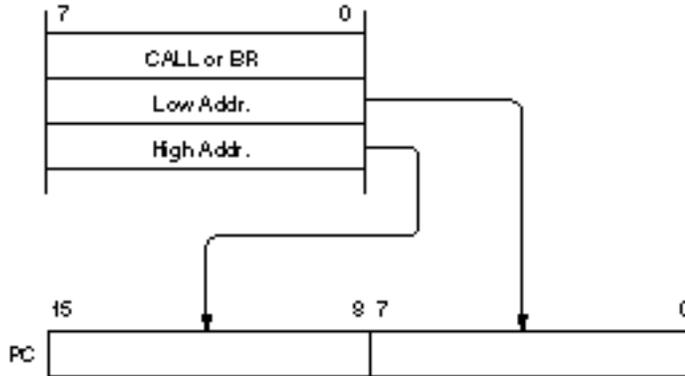
在执行“CALL !addr16”指令、“BR !addr16”指令或“CALLF !addr11”指令时，将执行立即寻址功能。

CALL !addr16 和 BR !addr16 指令的转移地址范围是所有存储空间。

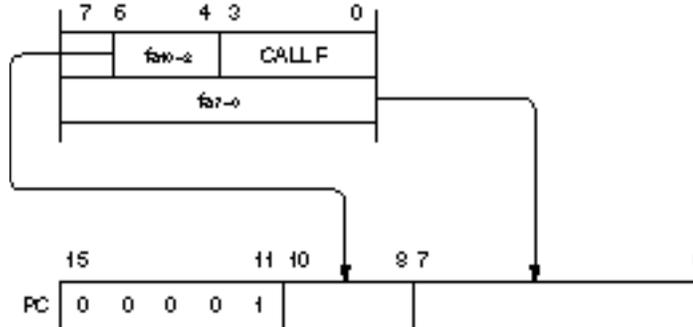
CALLF !addr11 指令的转移地址范围在 0800H 与 0FFFH 之间。

**[图示]**

CALL !addr16 和 BR !addr16 指令



CALLF !addr11 指令



### 3.3.3 表间接寻址

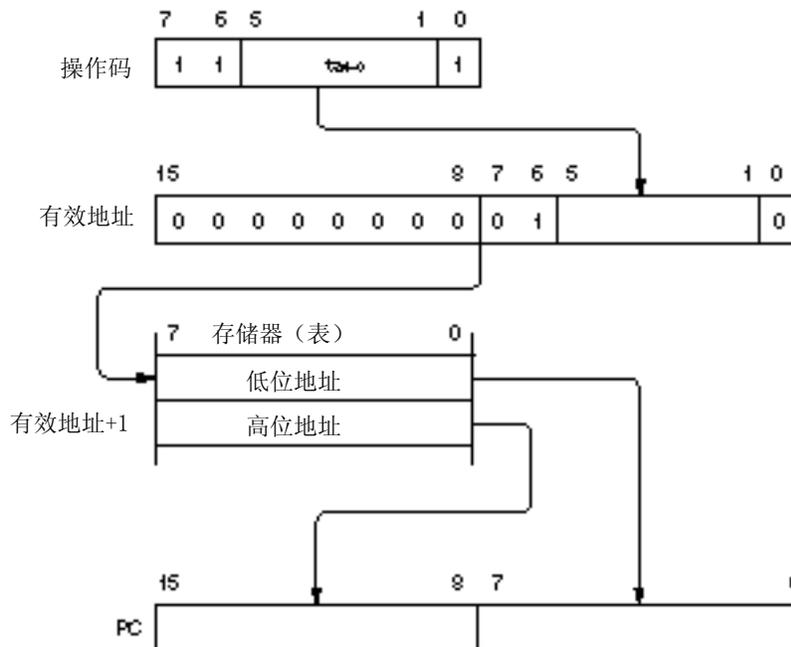
**[功能]**

通过指令码第 1 位到第 5 位的立即数，访问特定存储区中表的内容(转移目的地址)，并将表的内容赋给程序计数器(PC)，然后转向该地址执行程序。

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表 40H~7FH 中所存储的地址，转移地址范围可以是整个存储器空间。

**[图示]**



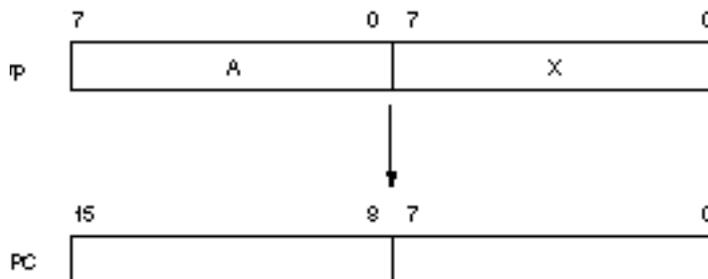
### 3.3.4 寄存器寻址

**[功能]**

将寄存器对(AX)的内容赋给程序计数器(PC)，然后转向该地址。

“BR AX”指令将执行寄存器寻址功能。

**[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 隐含寻址

##### [功能]

这种寻址方式自动寻址通用寄存器中作为累加器(A 和 AX)使用的寄存器。  
在 78K0/LC3 系列指令中下列指令采用隐含寻址方式。

指令	隐含寻址所指定的寄存器
MULU	A 寄存器存放被乘数, AX 寄存器存放运算结果
DIVUW	AX 寄存器用于存放被除数和商
ADJBA/ADJBS	存放进行十进制调整后的数据
ROR4/ROL4	存放用于数字循环的数字数据

##### [操作数格式]

由于指令自动采用隐含寻址方式, 所以没有特定的操作数格式。

##### [举例]

以 MULU X 指令为例, 这是一条 8 位乘 8 位的乘法运算指令, A 寄存器与 X 寄存器相乘的结果存放在 AX 中。在这个例子中 A 寄存器与 AX 寄存器均由隐含寻址方式指定。

### 3.4.2 寄存器寻址

**[功能]**

寄存器寻址方式将通用寄存器作为操作数进行访问，并由寄存器组选择标志(RBS0 ~ RBS1)和指令中的寄存器标识码，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。

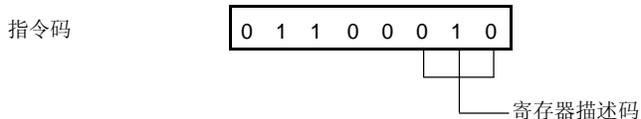
**[操作数格式]**

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

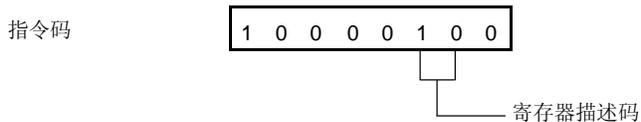
‘r’和‘rp’可用绝对名称(R0 ~ R7 以及 RP0 ~ RP3)和功能名称(X, A, C, B, E, D, L, H, AX, BC, DE 以及 HL)来描述。

**[举例]**

MOV A, C; 当选择通用寄存器 C 为 “r” 时



INCW DE; 当选择通用寄存器组 DE 为 “rp” 时



### 3.4.3 直接寻址

**[功能]**

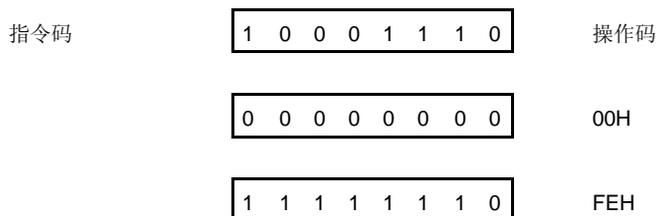
存储器会根据指令字中的操作数地址进行直接寻址操作。

**[操作数格式]**

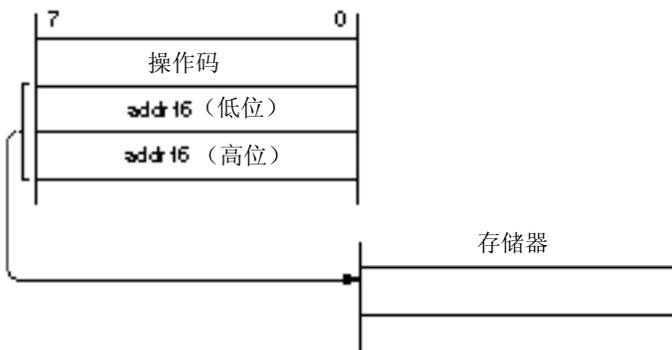
标识符	描述
addr16	标号或 16 位立即数

**[举例]**

MOV A, !0FE00H; 将!addr16 设置为 FE00H 时



**[图示]**



### 3.4.4 短直接寻址

**[功能]**

用指令中 8 位立即数直接对存储器的固定操作区域寻址。

该方式的寻址范围是 FE20H~FF1FH 总共 256 字节的区域。内部高速 RAM 和特殊功能寄存器(SFR)分别映射在 FE20H ~ FEFFH 以及 FF00H ~ FF1FH 的区域。

采用短直接寻址方式的特殊功能寄存器(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、用作定时器和事件计数器的比较和捕捉寄存器都被映射到该区域。这些特殊功能寄存器 (SFR)可以用很少的字节数和时钟数进行操作。

如果 8 位立即数是在 20H 和 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 与 1FH 之间，则一个有效地址的第 8 位设置为 1。参见下面的 [图示]。

**[操作数格式]**

标识符	描述
saddr	标志或从 FE20H ~ FF1FH 的立即数
saddrp	标志或从 FE20H ~ FF1FH 的立即数 (仅使用偶地址)

**[举例]**

MOV 0FE30H, A; 将寄存器 A 的值赋值给地址为 FE30H 的存储器

指令码

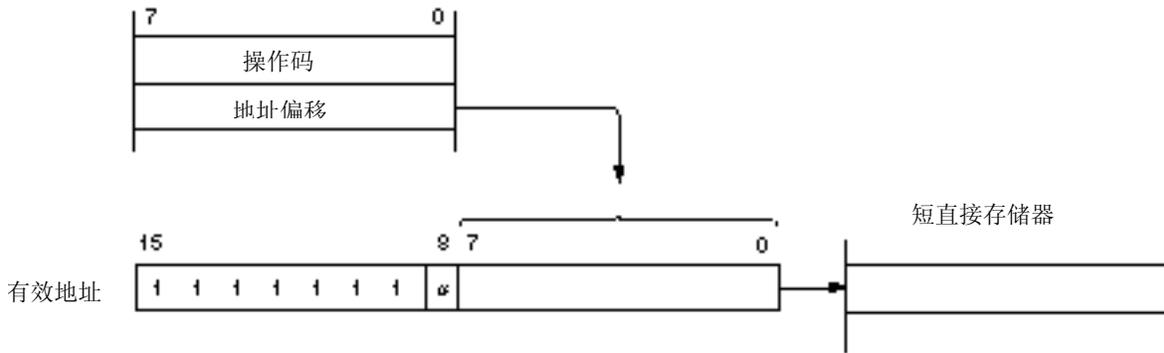
1 1 1 1 0 0 1 0

操作码

0 0 1 1 0 0 0 0

30H (saddr 偏移量)

**[图示]**



当 8 位立即数在 20H ~ FFH 之间时， $\alpha$  等于 0。

当 8 位立即数在 00H ~ 1FH 之间时， $\alpha$  等于 1。

### 3.4.5 特殊功能寄存器 (SFR) 寻址

**[功能]**

通过指令中的 8 位立即数对存储器的特殊功能寄存器 (SFR) 区域进行寻址。

寻址区间为 FF00H ~ FFCFH 以及 FFE0H ~ FFFFH，共 240 字节。而映射在 FF00H ~ FF1FH 区间的特殊功能寄存器 (SFR) 则采用短直接寻址方式。

**[操作数格式]**

标识符	描述
sfr	特殊功能寄存器名
sfrp	16 位可操作特殊功能寄存器名 (仅使用偶地址)

**[举例]**

MOV PM0, A; 选择 PM0(FF20H)作为 sfr

指令码

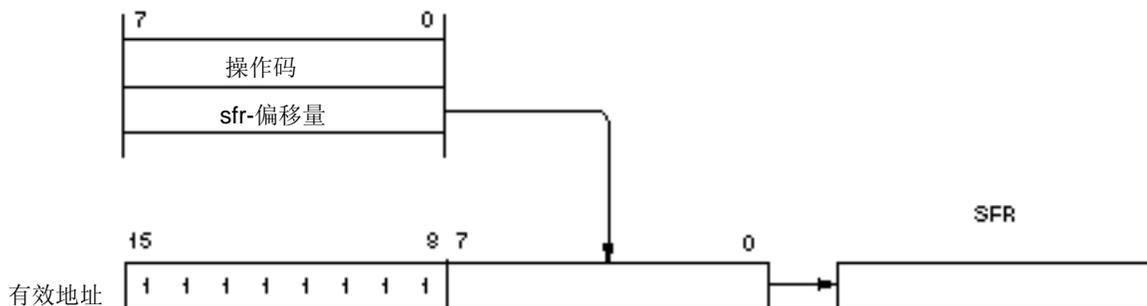
1 1 1 1 0 1 1 0

操作码

0 0 1 0 0 0 0 0

20H (sfr 偏移量)

**[图示]**



### 3.4.6 寄存器间接寻址

**[功能]**

根据寄存器对的内容进行寻址。该寄存器对由寄存器组选择标志(RBS0 和 RBS1)和指令字中的寄存器对指定码指定。

**[操作数格式]**

标识符	描述
-	[DE], [HL]

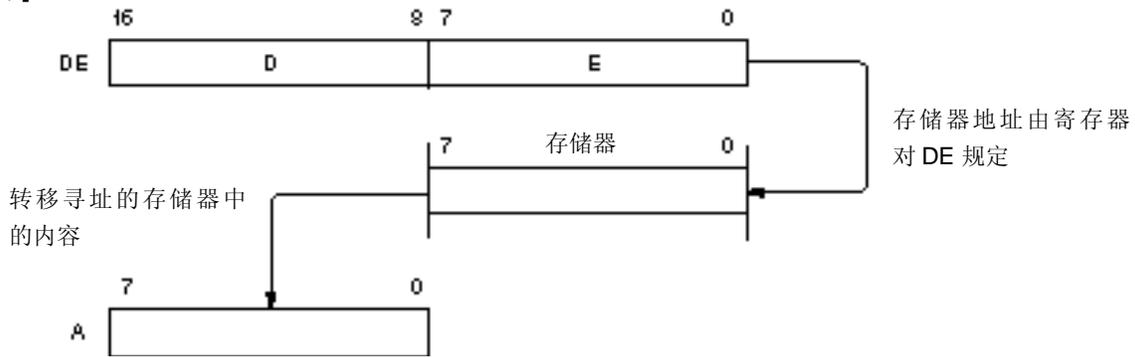
**[举例]**

MOV A, [DE]; 选择 DE 寄存器对作为操作数时

指令码



**[图示]**



### 3.4.7 基址寻址

**[功能]**

将 8 位立即数加到 HL 寄存器对中，HL 寄存器对作为基址寄存器。根据相加结果寻址。需要访问的 HL 寄存器对属于由寄存器组选择标志(RBS0 和 RBS1)确定的寄存器组。通过将偏移量扩展为 16 位正数，来完成加法操作，第 16 位的进位忽略不计。

**[操作数格式]**

标识符	描述
-	[HL + byte]

**[举例]**

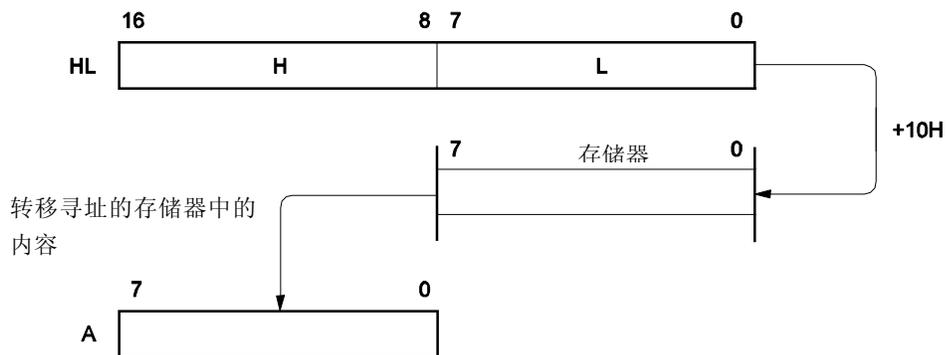
MOV A, [HL + 10H]; byte 的值为 10H 时

指令码

1 0 1 0 1 1 1 0

0 0 0 1 0 0 0 0

**[图示]**



### 3.4.8 基址变址寻址

**[功能]**

将 B 或 C 寄存器的内容加到 HL 寄存器中，HL 寄存器作为基址寄存器，并根据相加结果去寻址。需要访问的 HL、B 和 C 寄存器属于由寄存器组选择标志(RBS0 和 RBS1)确定的寄存器组。通过将 B 或 C 寄存器扩展为一个 16 位的正数来完成加法运算，第 16 位的进位忽略不计。

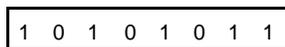
**[操作数格式]**

标识符	描述
-	[HL + B], [HL + C]

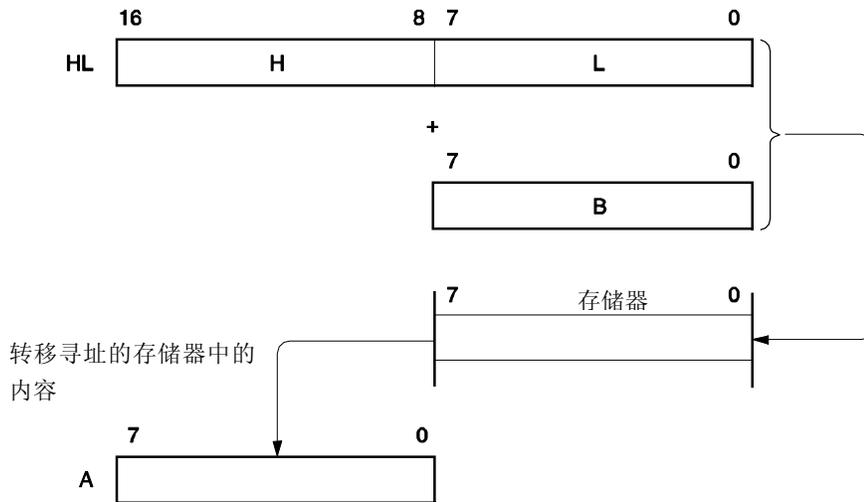
**举例]**

MOV A, [HL +B]; 选择 B 寄存器

指令码



**[图示]**



### 3.4.9 堆栈寻址

**[功能]**

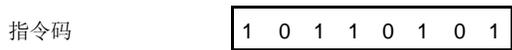
根据堆栈指针(SP)的内容对堆栈区域进行间接寻址。

当执行 PUSH, POP, 子程序调用和返回指令时, 或者产生中断请求时保存或恢复寄存器操作时, 将自动采用这种寻址方式。

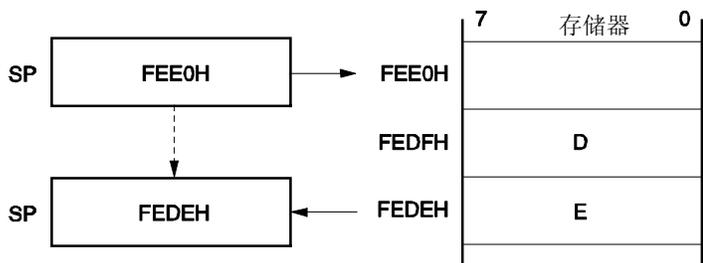
该方式仅对内部高速 RAM 区域进行寻址。

**[举例]**

PUSH DE; 保存在 DE 寄存器中



**[图示]**



## 第四章 端口功能

### 4.1 端口功能

有两种类型的引脚 I/O 缓冲器电源： $AV_{REF}$  和  $V_{DD}$ 。这些电源和引脚之间的关系显示如下。

表 4-1. 引脚 I/O 缓冲器电源

电源	相关引脚
$AV_{REF}$	P20 ~ P25
$V_{DD}$	除 P20 ~ P25 的其它引脚

注 仅适用于  $\mu PD78F041x$ 。  $\mu PD78F040x$  的供电电源是  $V_{DD}$ 。

78K0/LC3 产品中提供的端口如图 4-1 所示，这些端口可以支持多种控制操作。每个端口的功能如表 4-2 所示。

除了作为数字 I/O 端口功能，这些端口还有几个复用功能。如需了解这些端口复用功能的详细信息，请参见第二章引脚功能。

图 4-1. 端口类型

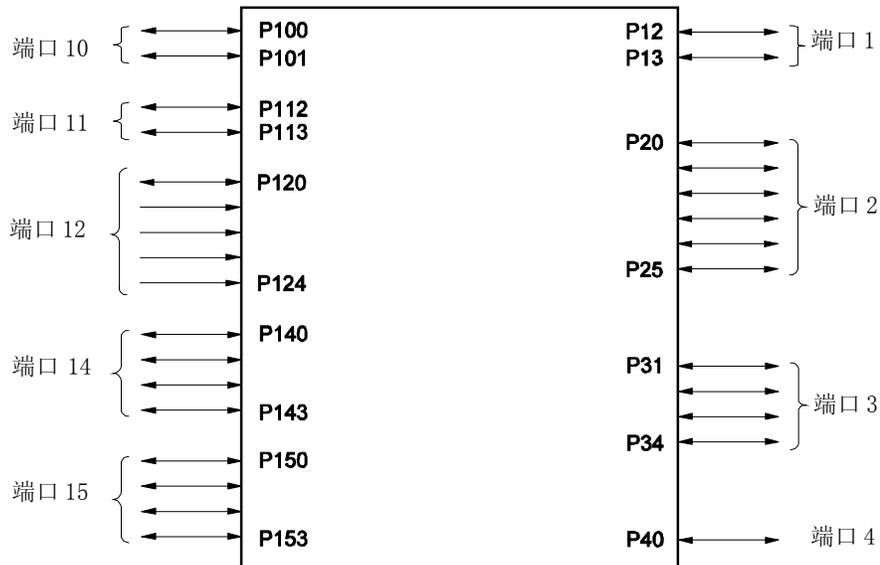


表 4-2. 端口功能

功能名称	I/O	功能	复位后	复位功能
P12	I/O	端口 1 2 位 I/O 端口 可以位选输入输出模式 通过软件设置，可以定义内部上拉电阻的使用	输入端口	RxD0/KR3/<RxD6>
P13				TxD0/KR4/<TxD6>
P20	I/O	端口 2 6 位 I/O 端口 可以位选输入输出模式	数字输入 端口	SEG 21/ANI0 <sup>‡</sup>
P21				SEG 20/ANI1 <sup>‡</sup>
P22				SEG 19/ANI2 <sup>‡</sup>
P23				SEG 18/ANI3 <sup>‡</sup>
P24				SEG 17/ANI4 <sup>‡</sup>
P25				SEG 16/ANI5 <sup>‡</sup>
P31	I/O	端口 3 4 位 I/O 端口 可以位选输入输出模式 通过软件设置，可以定义内部上拉电阻的使用	输入端口	TOH1/INTP3
P32				TOH0/MCGO
P33				TI000/RTCDIV/ RTCCL/BUZ/INTP2
P34				TI52/TI010/TO00/ RTC1HZ/INTP1
P40	I/O	端口 4 1 位 I/O 端口 可以位选输入输出模式 通过软件设置，可以定义内部上拉电阻的使用	输入端口	V <sub>LC3</sub> /KR0
P100, P101	I/O	端口 10 2 位 I/O 端口 可以位选输入输出模式 通过软件设置，可以定义内部上拉电阻的使用	输入端口	SEG4, SEG5
P112	I/O	端口 11 2 位 I/O 端口 可以位选输入输出模式 通过软件设置，可以定义内部上拉电阻的使用	输入端口	SEG6/TxD6
P113				SEG7/RxD6
P120	I/O	端口 12 1 位 I/O 端口和 4 位输入端口 仅 P120 可以通过软件设置，定义内部上拉电阻的使用	输入端口	INTP0/EXLVI
P121				X1/OCD0A
P122				X2/EXCLK/OCD0B
P123				XT1
P124				XT2
P140 ~ P143	I/O	端口 14 4 位 I/O 端口 可以位选输入输出模式 通过软件设置，可以定义内部上拉电阻的使用	输入端口	SEG8 ~ SEG11
P150 ~ P153	I/O	端口 15 4 位 I/O 端口 可以位选输入输出模式 通过软件设置，可以定义内部上拉电阻的使用	输入端口	SEG12 ~ SEG15

注 仅对 $\mu$ PD78F041x。

备注 带有箭头(< >)的功能可以通过设置输入切换控制寄存器 (ISC) 进行分配。

## 4.2 端口配置

端口包括如下硬件。

表 4-3. 端口配置

项目	配置
控制寄存器	端口模式寄存器(PM1 ~ PM4, PM10 ~ PM12, PM14, PM15) 端口寄存器(P1 ~ P4, P10 ~ P12, P14, P15) 上拉电阻选择寄存器 (PU1, PU3, PU4, PU10 ~ PU12, PU14, PU15) 端口功能寄存器 1 (PF1) 端口功能寄存器 2 (PF2) 端口功能寄存器 ALL (PFALL) A/D 端口配置寄存器 0 (ADPC0) <sup>注</sup>
端口	总计: 30
上拉电阻	总计: 20

注 仅对 $\mu$ PD78F041x。

## 4.2.1 端口 1

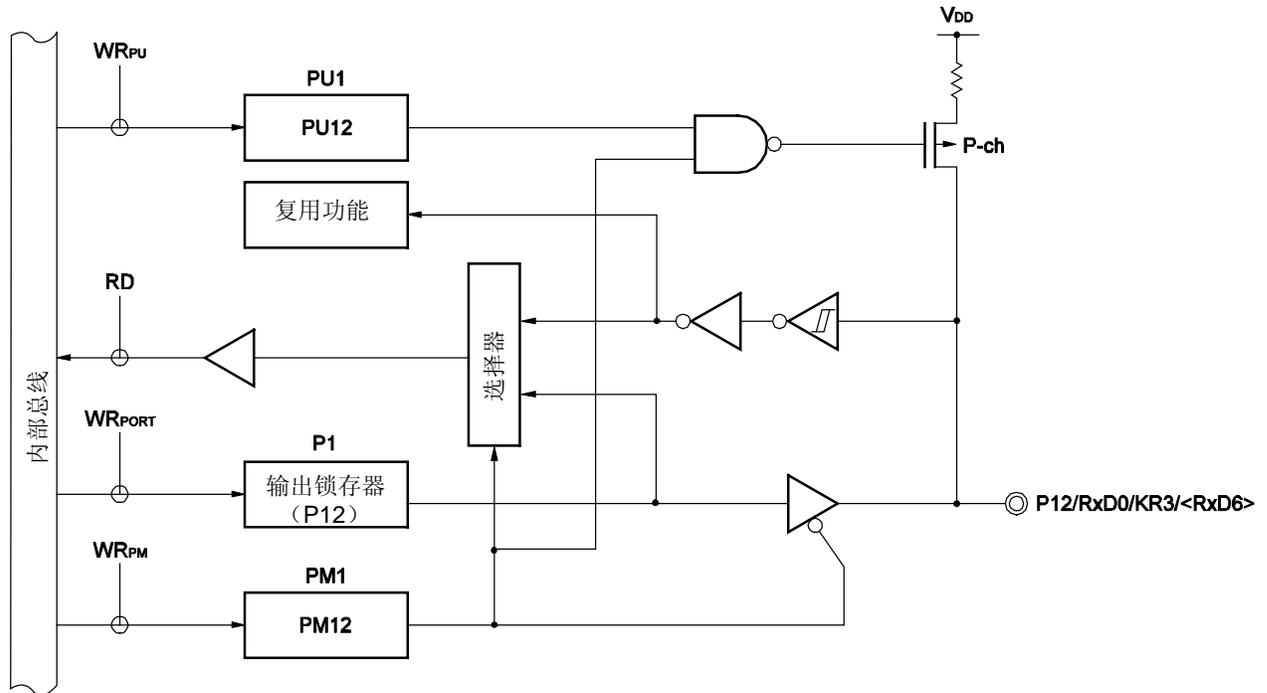
端口 1 是具有输出锁存功能的 2 位 I/O 端口。通过使用端口模式寄存器 1(PM1)，可以位选端口 1 为输入或输出模式。如果 P12 和 P13 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 1(PU1) 以 1 位单元的方式指定。

这个端口还可以用于按键中断输入和串行接口数据 I/O。

复位信号的产生可将端口 1 设置为输入模式。

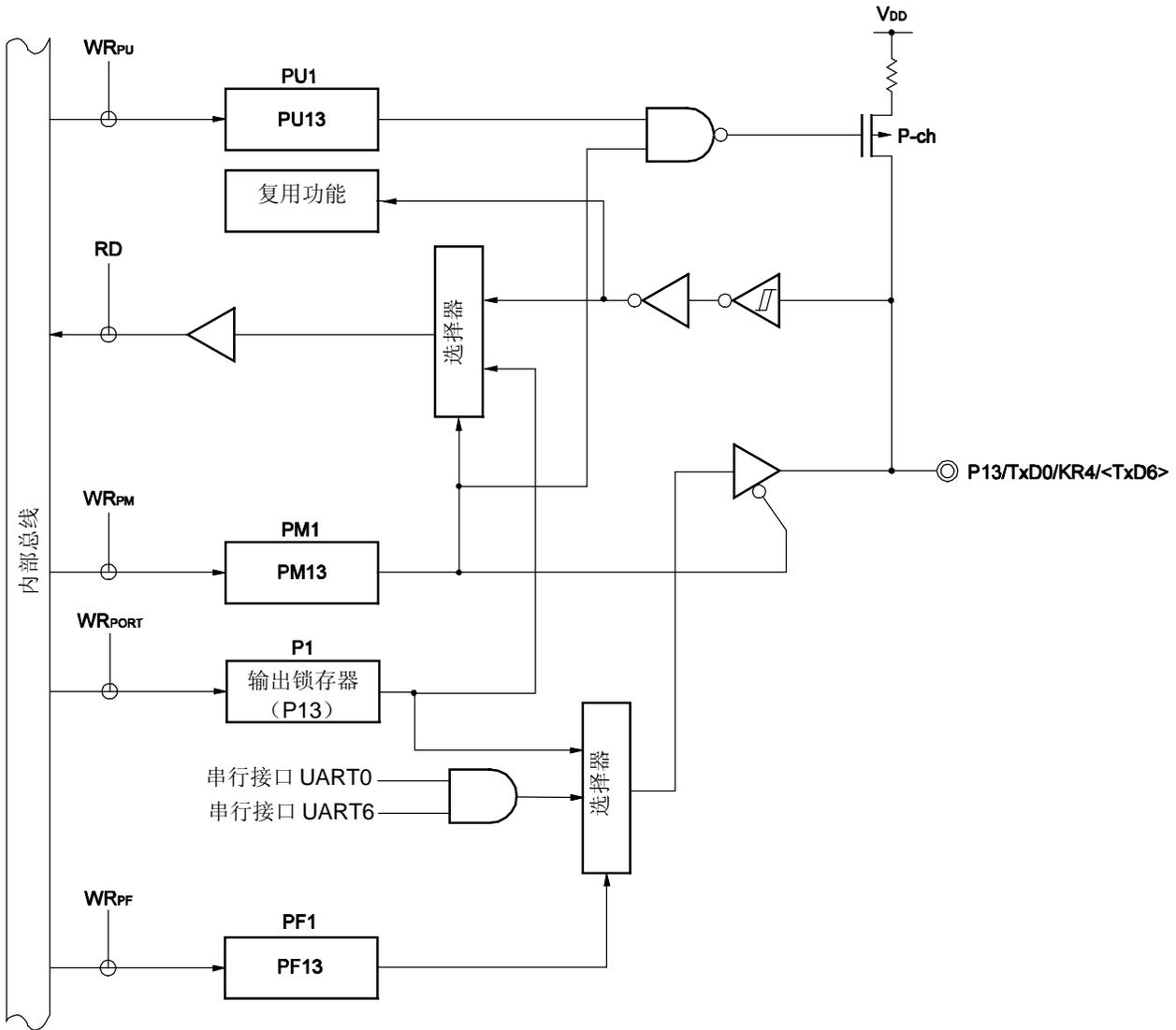
图 4-2 和 4-3 显示端口 1 的框图。

图 4-2. P12 的框图



- P1: 端口寄存器 1
- PU1: 上拉电阻选择寄存器 1
- PM1: 端口模式寄存器 1
- RD: 读信号
- WR<sub>xx</sub>: 写信号

图 4-3. P13 的框图



- P1: 端口寄存器 1
- PU1: 上拉电阻选择寄存器 1
- PM1: 端口模式寄存器 1
- RD: 读信号
- WR<sub>xx</sub>: 写信号

### 4.2.2 端口 2

端口 2 是具有输出锁存功能的 6 位 I/O 端口。通过使用端口模式寄存器 2(PM2)，可以位选端口 2 为输入或输出模式。

这个端口还可以用于 10 位逐次逼近型 A/D 转换器模拟输入(仅对 $\mu$ PD78F041x)和 segment 输出。

P20/SEG21/ANI0<sup>注</sup> ~ P25/SEG16/ANI5<sup>注</sup>，使用端口功能寄存器 2 (PF2)设置端口功能 (除 segment 输出)，使用 ADPC0 设置数字 I/O，使用 PM2 设置输入模式，将其用作数字输入引脚。使用这些引脚从低位开始。

P20/SEG21/ANI0<sup>注</sup> ~ P25/SEG16/ANI5<sup>注</sup>，使用端口功能寄存器 2 (PF2)设置端口功能 (除 segment 输出)，使用 ADPC0 设置数字 I/O，使用 PM2 设置输出模式，将其用作数字输出引脚。使用这些引脚从低位开始。

复位信号的产生可将端口 2 设置为输入模式。

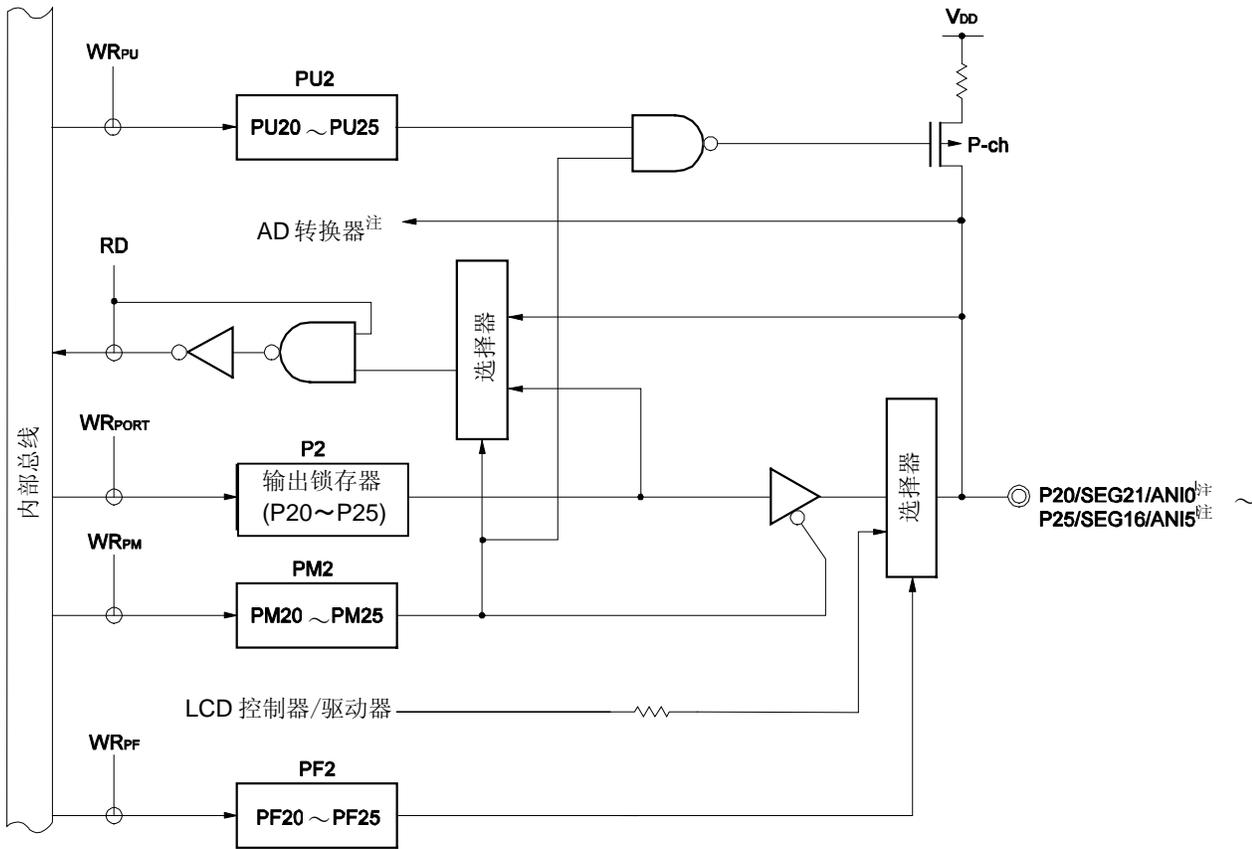
图 4-4 显示端口 2 的框图。

表 4-4. P20/SEG21/ANI0<sup>注</sup> ~ P25/SEG16/ANI5<sup>注</sup>引脚的设置功能

PF2	ADPC0 <sup>注</sup>	PM2	ADS	P20/SEG21/ANI0 <sup>注</sup> ~ P25/SEG16/ANI5 <sup>注</sup>
数字/模拟选择	模拟输入选择	输入模式	不选择 ANI.	模拟输入 (不得修改)
			选择 ANI.	模拟输入 (由逐次逼近型 A/D 转换器修改)
	数字 I/O 选择	输出模式	-	禁止设置
		输入模式	-	数字输入
		输出模式	-	数字输出
SEG 输出选择	-	-	-	Segment 输出

注 仅对 $\mu$ PD78F041x。

图 4-4. P20 ~ P25 的框图



注 仅对 $\mu$ PD78F041x。

- 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
- PF2: 端口功能寄存器 2
- RD: 读信号
- WR $\times\times$ : 写信号

## 4.2.3 端口 3

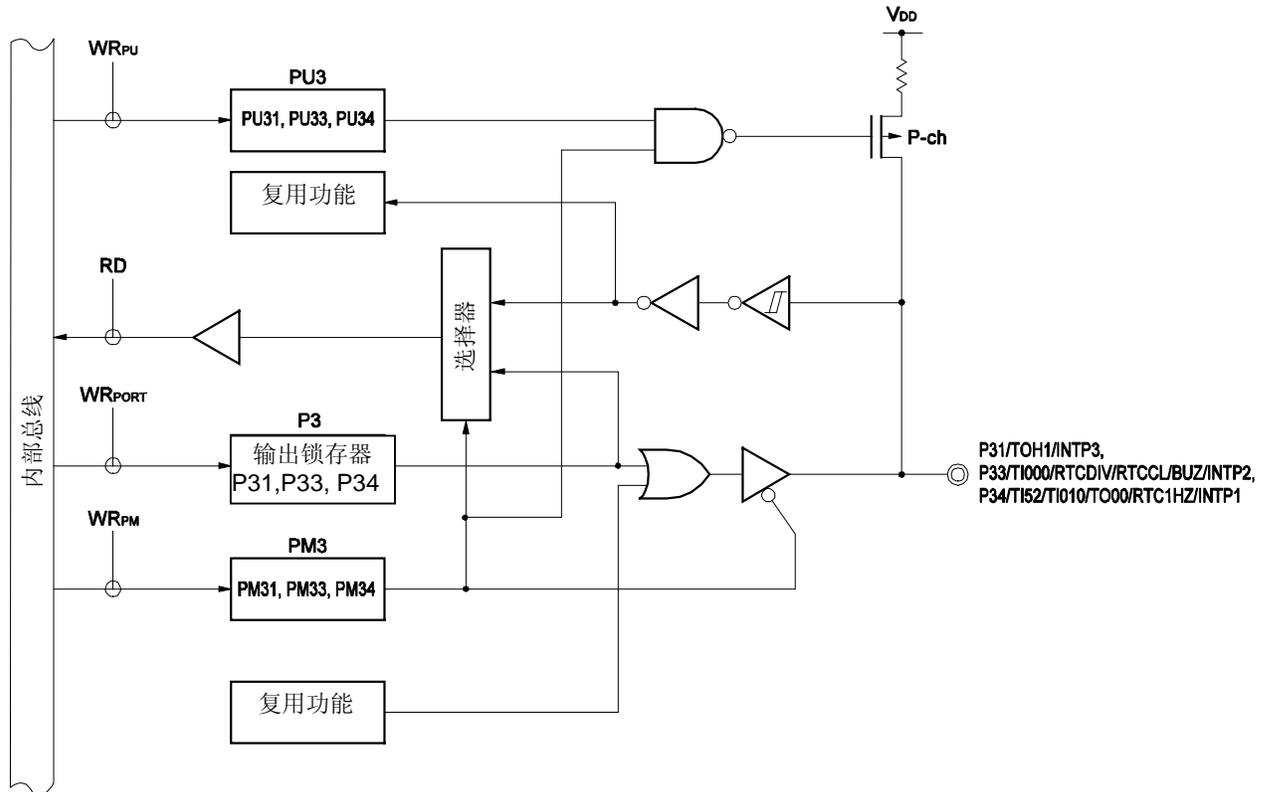
端口 3 是具有输出锁存功能的 4 位 I/O 端口。通过使用端口模式寄存器 3(PM3)，可以位选端口 3 为输入或输出模式。如果 P31 ~ P34 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 3(PU3)以 1 位单元的方式指定。

这个端口还可以用于外部中断请求输入、定时器 I/O、manchester 编码发生器输出、实时计数器输出和蜂鸣器输出。

复位信号的产生可将端口 3 设置为输入模式。

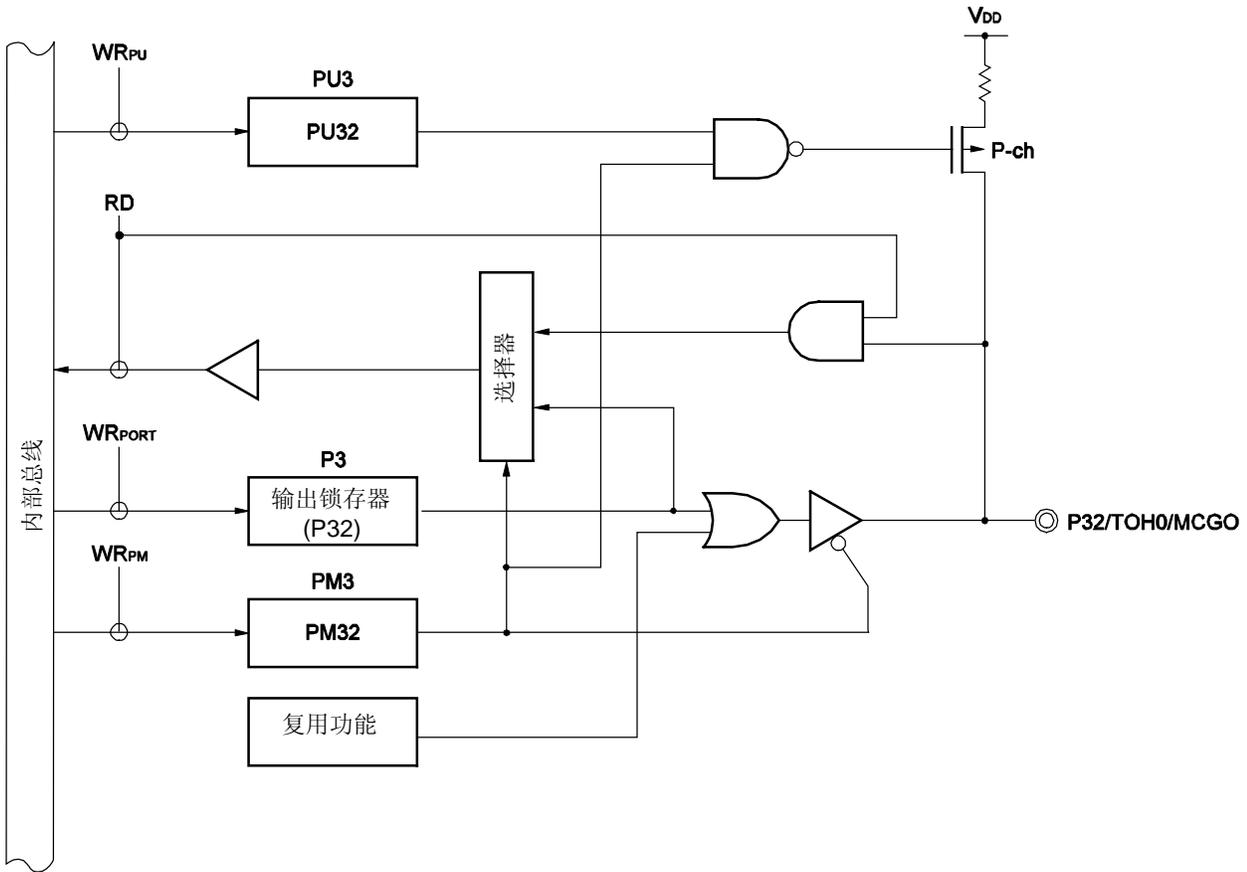
图 4-5 和 4-6 显示端口 3 的框图。

图 4-5. P31, P33, P34 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<sub>xx</sub>: 写信号

图 4-6. 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<sub>xx</sub>: 写信号

## 4.2.4 端口 4

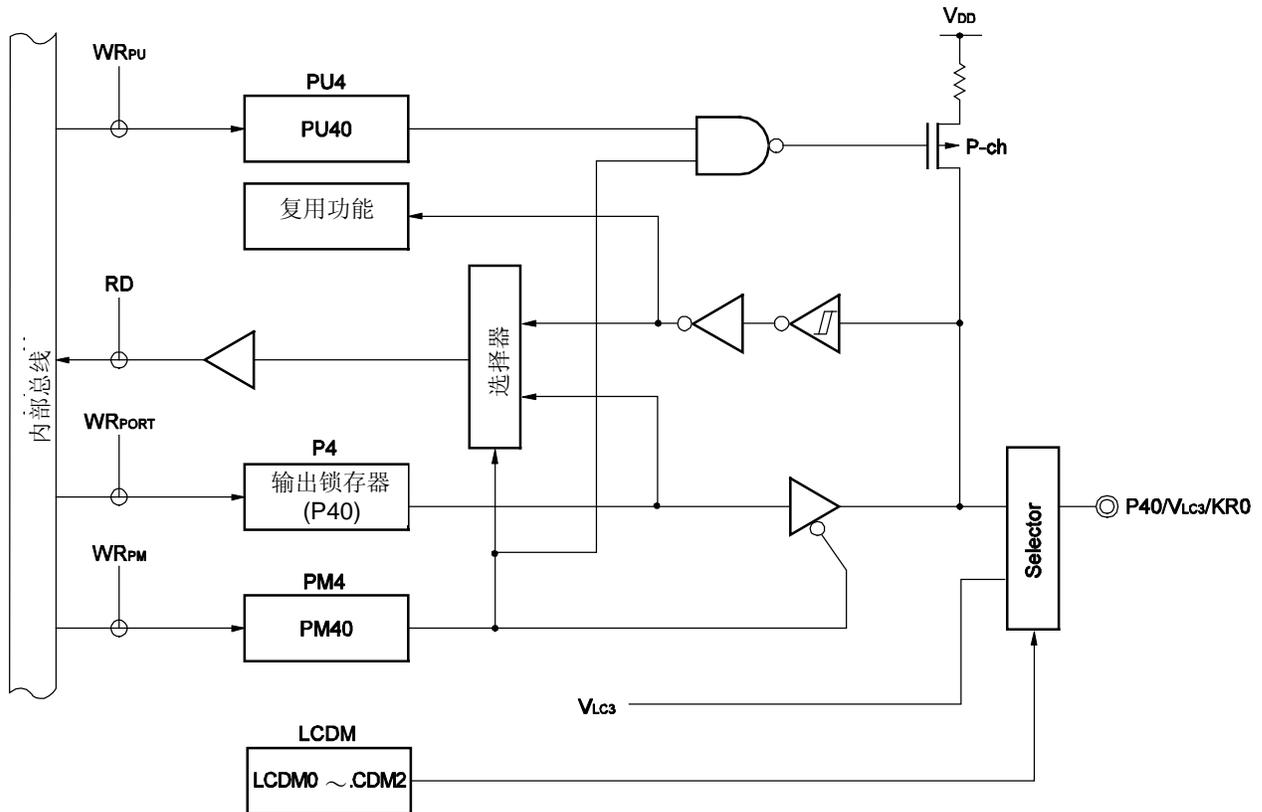
端口 4 是具有输出锁存功能的 1 位 I/O 端口。通过使用端口模式寄存器 4(PM4)，可以位选端口 4 为输入或输出模式。如果 P40 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 4(PU4)以 1 位单元的方式指定。

这个端口还可以用于 LCD 驱动电源引脚和按键中断输入引脚。

复位信号的产生可将端口 4 设置为输入模式。

图 4-7 显示端口 4 的框图。

图 4-7. P40 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
- LCDM: LCD 显示模式寄存器
- RD: 读信号
- WRxx: 写信号

## 4.2.5 端口 10

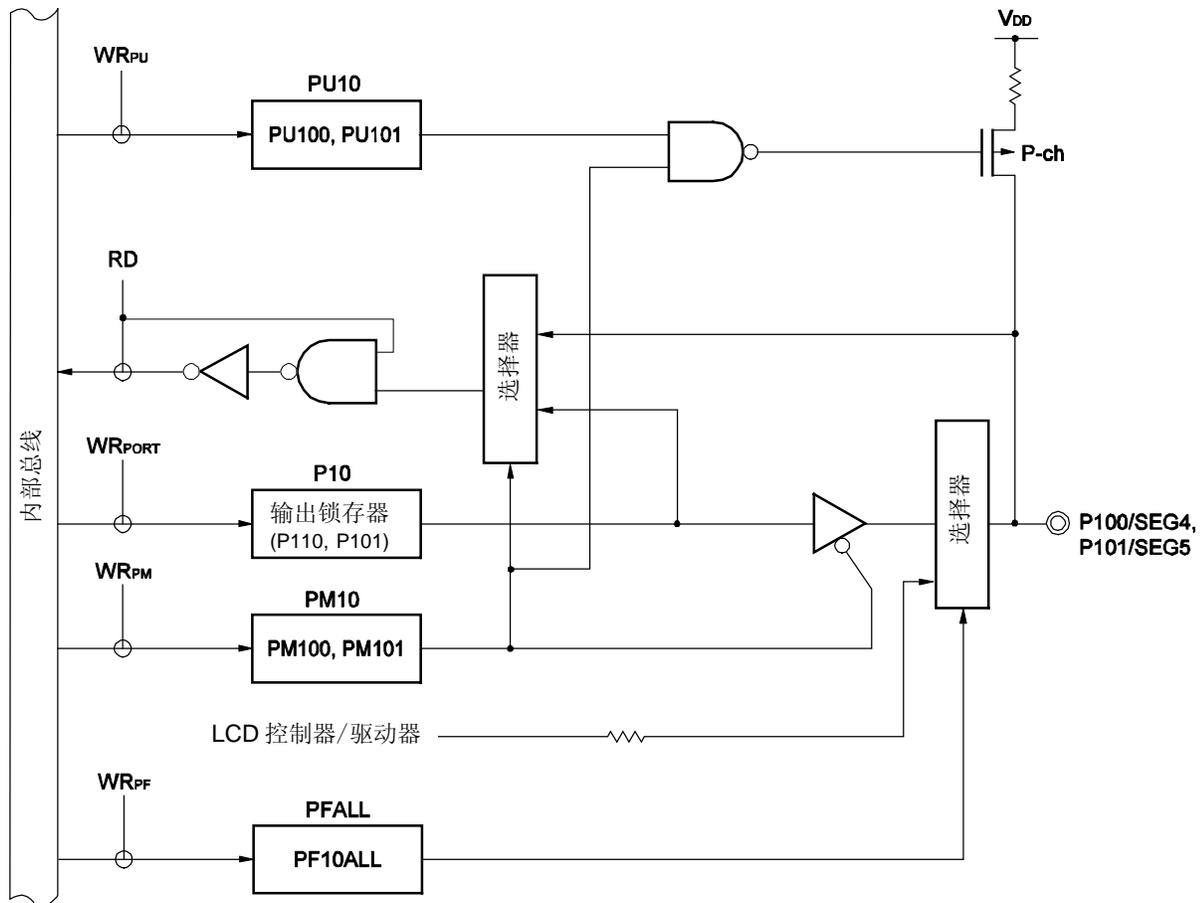
端口 10 是具有输出锁存功能的 2 位 I/O 端口。通过使用端口模式寄存器 10(PM10)，可以位选端口 10 为输入或输出模式。如果 P100 ~ P101 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 10(PU10)以 1 位单元的方式指定。

这个端口还可以用于 segment 输出。

复位信号的产生可将端口 10 设置为输入模式。

图 4-8 显示端口 10 的框图。

图 4-8. P100 ~ P101 的框图



- P10: 端口寄存器 10
- PU10: 上拉电阻选择寄存器 10
- PM10: 端口模式寄存器 10
- PFALL: 端口功能寄存器 ALL
- RD: 读信号
- WRxx: 写信号

## 4.2.6 端口 11

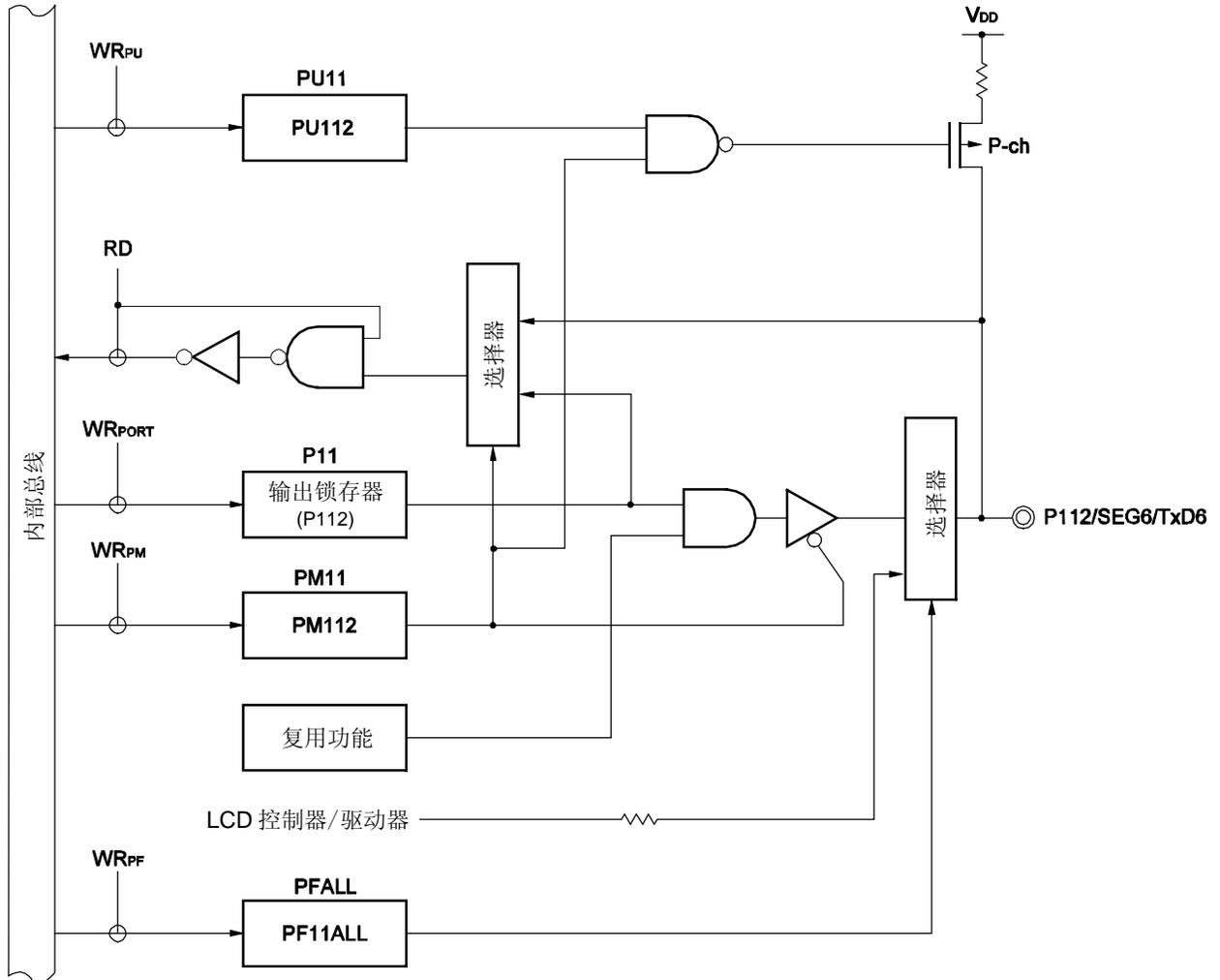
端口 11 是具有输出锁存功能的 2 位 I/O 端口。通过使用端口模式寄存器 11(PM11)，可以位选端口 11 为输入或输出模式。如果 P112，P113 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 11(PU11)以 1 位单元的方式指定。

这个端口还可以用于 segment 输出和串行接口数据 I/O。

复位信号的产生可将端口 11 设置为输入模式。

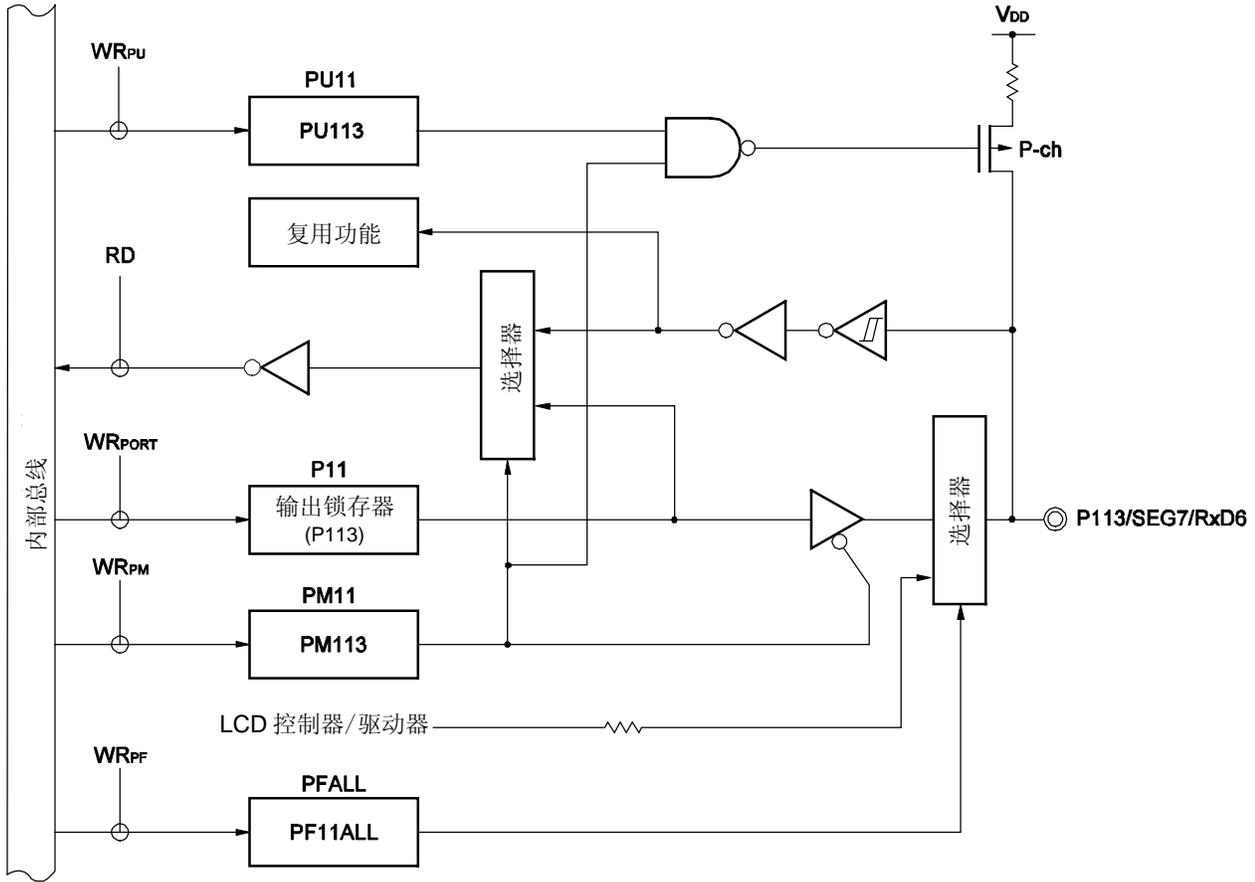
图 4-9~4-10 显示端口 11 的框图。

图 4-9. P112 的框图



- P11: 端口寄存器 11
- PU11: 上拉电阻选择寄存器 11
- PM11: 端口模式寄存器 11
- PFALL: 端口功能寄存器 ALL
- RD: 读信号
- WRxx: 写信号

图 4-10. P113 的框图



- P11: 端口寄存器 11
- PU11: 上拉电阻选择寄存器 11
- PM11: 端口模式寄存器 11
- PFALL: 端口功能寄存器 ALL
- RD: 读信号
- WR<sub>xx</sub>: 写信号

#### 4.2.7 端口 12

端口 12 是具有输出锁存功能的 1 位 I/O 端口和 4 位输入端口。仅 P120 可以使用端口模式寄存器 12(PM12)，位选端口为输入或输出模式。如果 P120 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 12(PU12)指定。

这个端口还可以用于外部中断请求输入、外部低电压检测的电压输入、主系统时钟的振荡器连接、副系统时钟的振荡器连接和主系统时钟的外部时钟输入。

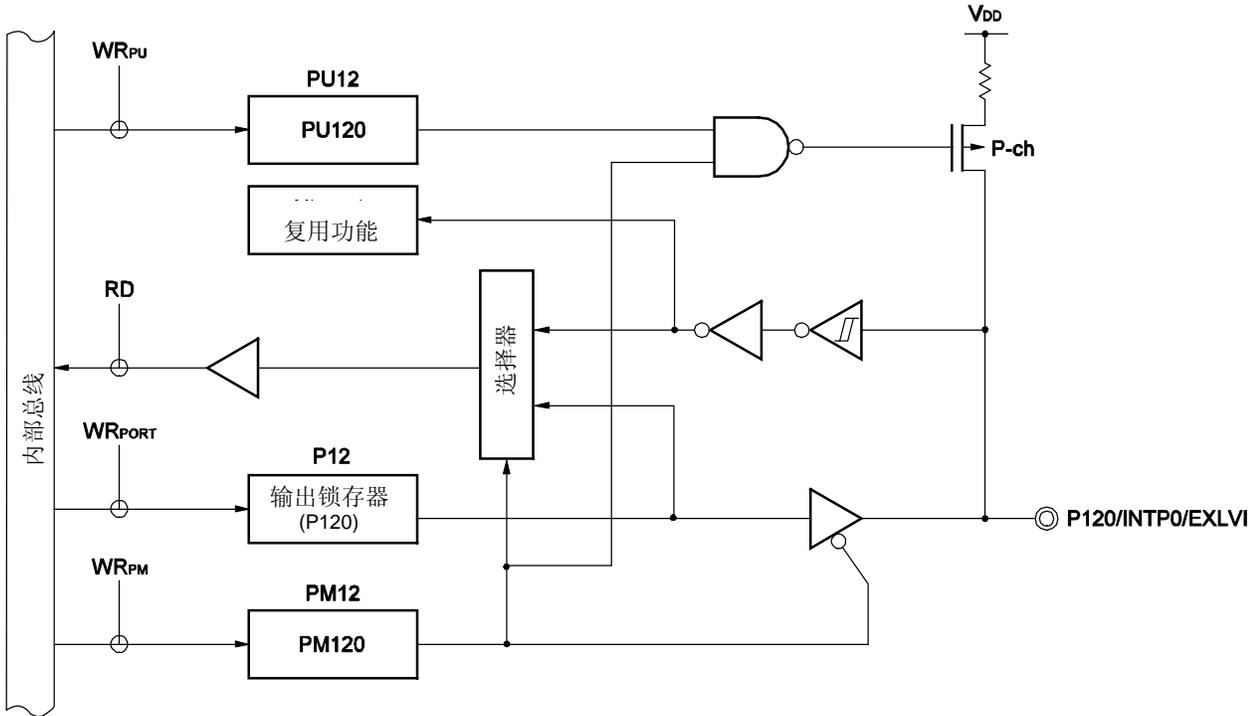
复位信号的产生可将端口 12 设置为输入模式。

图 4-11~4-13 显示端口 12 的框图。

**注意事项** 当使用 P121 ~ P124 引脚连接主系统时钟 (X1,X2) 或副系统时钟 (XT1,XT2) 的振荡器，或者输入该主系统时钟的外部时钟 (EXCLK) 时，必须通过使用时钟操作模式选择寄存器 (OSCCTL) 对 X1 振荡模式、XT1 振荡模式或者外部时钟输入模式进行设置 (如需了解详细信息，可参见 5.3 (1) 时钟操作模式选择寄存器 (OSCCTL) 和 (3) 副系统时钟引脚操作模式设置)。OSCCTL 的复位值为 00H (P121 ~ P124 都用作输入端口引脚)。

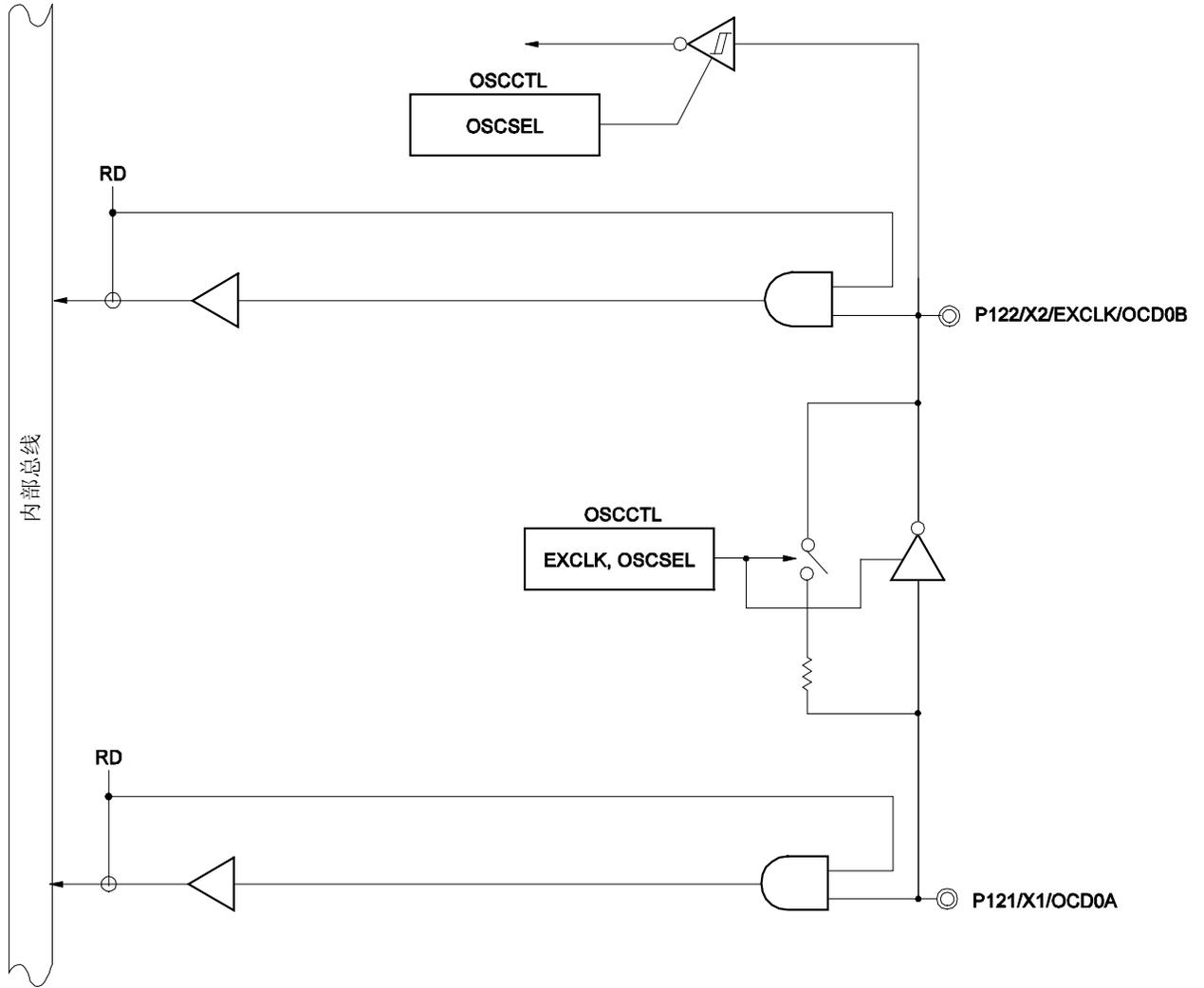
**备注** P121 和 P122 能够用于片上调试模式设定引脚 (OCD0A, OCD0B)。详情请参看 第二十五章 片上调试功能。

图 4-11. P120 的框图



- P12: 端口寄存器 12
- PU12: 上拉电阻选择寄存器 12
- PM12: 端口模式寄存器 12
- RD: 读信号
- WR<sub>xx</sub>: 写信号

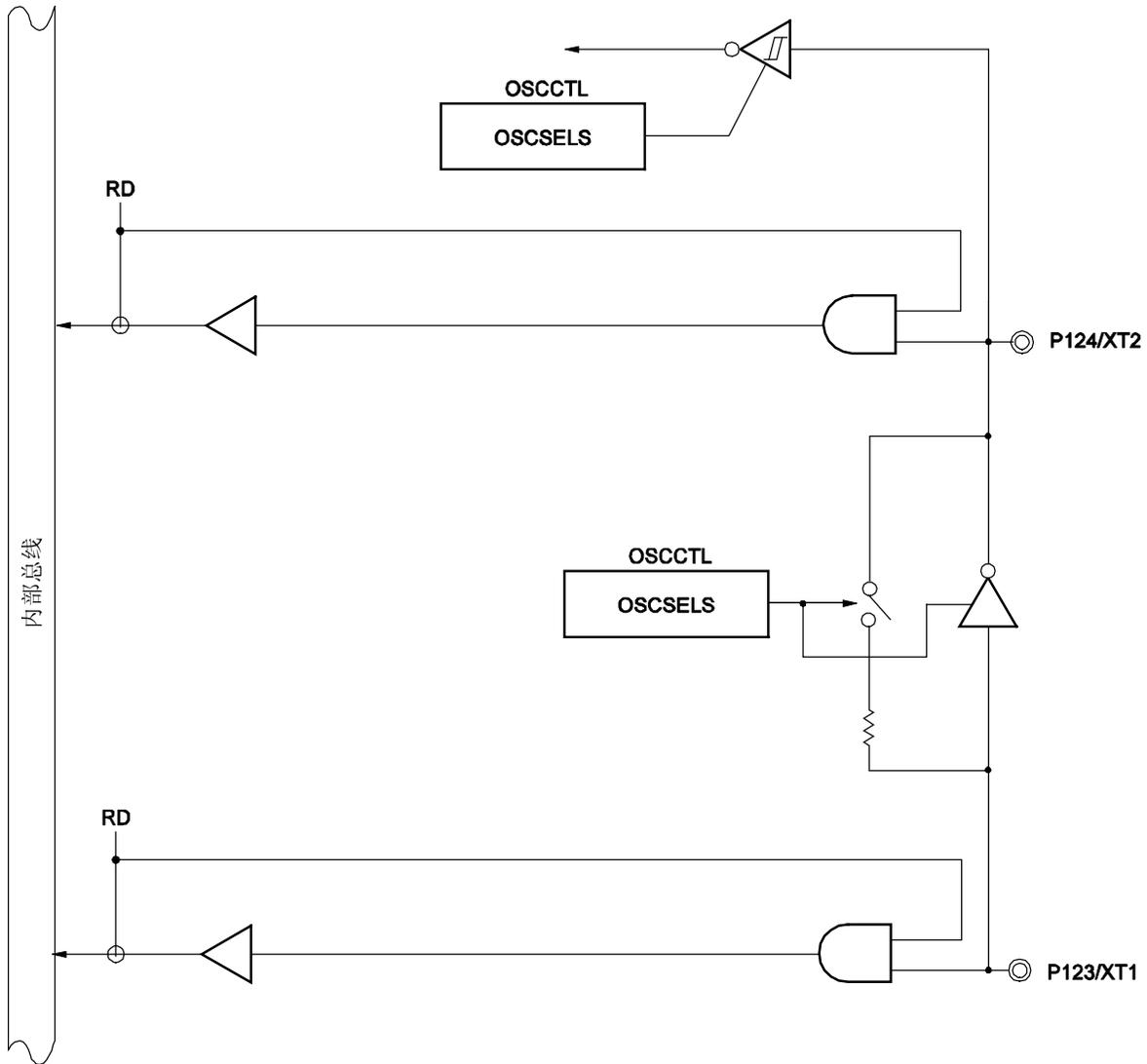
图 4-12. P121 和 P122 的框图



OSCCTL: 时钟操作模式选择寄存器

RD: 读信号

图 4-13. P123 和 P124 的框图



OSCCTL: 时钟操作模式选择寄存器

RD: 读信号

## 4.2.8 端口 14

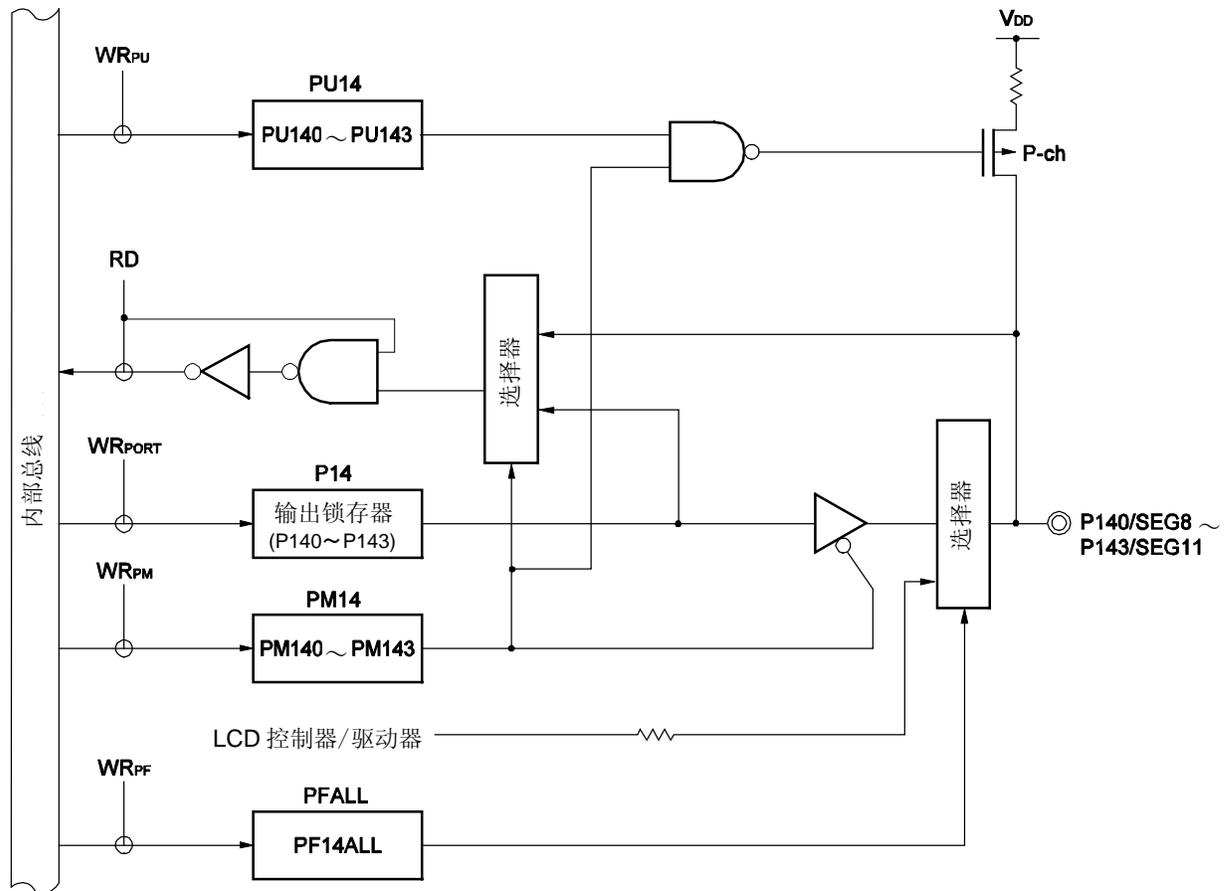
端口 14 是具有输出锁存功能的 4 位 I/O 端口。端口 14 可以使用端口模式寄存器 14(PM14)，位选端口为输入或输出模式。如果 P140~ P143 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 14(PU14)以 1 位单元的方式指定。

这个端口还可以用于 segment 输出。

复位信号的产生可将端口 14 设置为输入模式。

图 4-14 显示端口 14 的框图。

图 4-14. P140 ~ P143 的框图



- P14: 端口寄存器 14
- PU14: 上拉电阻选择寄存器 14
- PM14: 端口模式寄存器 14
- PFALL: 端口功能寄存器 ALL
- RD: 读信号
- WR<sub>xx</sub>: 写信号

## 4.2.9 端口 15

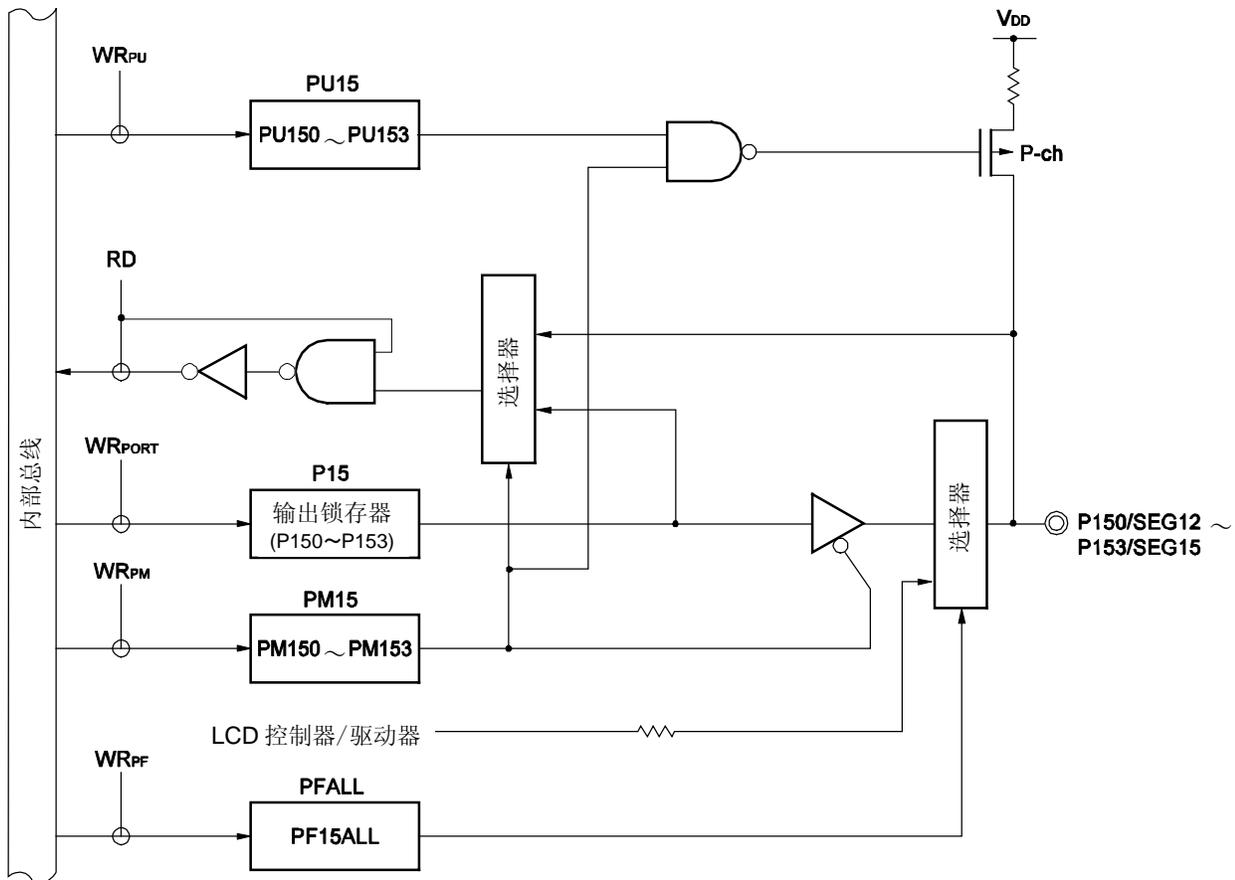
端口 15 是具有输出锁存功能的 4 位 I/O 端口。端口 15 可以使用端口模式寄存器 15(PM15)，位选端口为输入或输出模式。如果 P150~ P153 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 15(PU15)以 1 位单元的方式指定。

这个端口还可以用于 segment 输出。

复位信号的产生可将端口 15 设置为输入模式。

图 4-15 显示端口 15 的框图。

图 4-15. P150 ~ P153 的框图



- P15: 端口寄存器 15
- PU15: 上拉电阻选择寄存器 15
- PM15: 端口模式寄存器 15
- PFALL: 端口功能寄存器 ALL
- RD: 读信号
- WRxx: 写信号

### 4.3 控制端口功能的寄存器

端口功能由如下七种类型寄存器控制。

- 端口模式寄存器(PM1 ~ PM4, PM10 ~ PM12, PM14, PM15)
- 端口寄存器(P1 ~ P4, P10 ~ P12, P14, P15)
- 上拉电阻选择寄存器(PU1, PU3, PU4, PU10 ~ PU12, PU14, PU15)
- 端口功能寄存器 1 (PF1)
- 端口功能寄存器 2 (PF2)
- 端口功能寄存器 ALL (PFALL)
- A/D 端口配置寄存器 0 (ADPC0)<sup>※</sup>

注 仅对 $\mu$ PD78F041x。

#### (1) 端口模式寄存器 (PM1 ~ PM4, PM10 ~ PM12, PM14, PM15)

这类寄存器以 1 位单元的方式定义端口的输入或输出模式。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器的内容设置为 FFH。

当端口引脚使用复用功能时，需要参考 4.5 使用复用功能时，PFALL, PF2, PF1, ISC 端口模式寄存器和输出锁存器的设置。

图 4-16. 端口模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位值	R/W
PM1	1	1	1	1	PM13	PM12	1	1	FF21H	FFH	R/W
PM2	1	1	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	PM34	PM33	PM32	PM31	1	FF23H	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FF24H	FFH	R/W
PM10	1	1	1	1	1	1	PM101	PM100	FF2AH	FFH	R/W
PM11	1	1	1	1	PM113	PM112	1	1	FF2BH	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FF2CH	FFH	R/W
PM14	1	1	1	1	PM143	PM142	PM141	PM140	FF2EH	FFH	R/W
PM15	1	1	1	1	PM153	PM152	PM151	PM150	FF2FH	FFH	R/W

PMmn	Pmn 引脚 I/O 模式选择 (m = 1 ~ 4, 10 ~ 12, 14, 15; n = 0 ~ 5)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

**注意事项** 确保设置 PM1 的第 0 位, 第 1 位和第 4 ~ 7 位、PM2 的第 6 位, 第 7 位、PM3 的第 0 位, 第 5 ~ 7 位、PM4 的第 1 ~ 7 位、PM10 的第 2 ~ 7 位、PM11 的第 0 位, 第 1 位, 第 4 ~ 7 位、PM12 的第 1 ~ 7 位、PM14 的第 4 ~ 7 位和 PM15 的第 4 ~ 7 位为“1”。

**(2) 端口寄存器(P1 to P4, P10 ~ P12, P14, P15)**

这类寄存器用于设置芯片端口要输出的数据。

如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将寄存器清零(00H)。

图 4-17. 端口寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位值	R/W
P1	0	0	0	0	P13	P12	0	0	FF01H	00H (输出锁存)	R/W
P2	0	0	P25	P24	P23	P22	P21	P20	FF02H	00H (输出锁存)	R/W
P3	0	0	0	P34	P33	P32	P31	0	FF03H	00H (输出锁存)	R/W
P4	0	0	0	0	0	0	0	P40	FF04H	00H (输出锁存)	R/W
P10	0	0	0	0	0	0	P101	P100	FF0AH	00H (输出锁存)	R/W
P11	0	0	0	0	P113	P112	0	0	FF0BH	00H (输出锁存)	R/W
P12	0	0	0	P124 <sup>注2</sup>	P123 <sup>注2</sup>	P122 <sup>注2</sup>	P121 <sup>注2</sup>	P120	FF0CH	00H <sup>注1</sup> (输出锁存)	R/W <sup>注1</sup>
P14	0	0	0	0	P143	P142	P141	P140	FF0EH	00H (输出锁存)	R/W
P15	0	0	0	0	P153	P152	P151	P150	FF0FH	00H (输出锁存)	R/W

Pmn	m = 1 ~ 4, 10 ~ 12, 14, 15; n = 0 ~ 5	
	输出数据控制(输出模式下)	输入数据读取(输入模式下)
0	输出 0	输入低电平
1	输出 1	输入低电平

- 注
1. P121 ~ P124 为只读。复位后不确定。
  2. 当引脚的操作模式为时钟输入模式时，读到值始终为 0。

**(3) 上拉电阻选择寄存器(PU1, PU3, PU4, PU10 ~ PU12, PU14, PU15)**

这类寄存器指定是否使用 P12, P13, P31~P34, P40, P100, P101, P112, P113, P120, P140 ~ P143 或 P150 ~ P153 的内部上拉电阻。并且这些引脚已通过 PU1, PU3, PU4, PU10 ~ PU12, PU14 和 PU15 指定为使用内部上拉电阻。不论是否设置了 PU1, PU3, PU4, PU10 ~ PU12, PU14 和 PU15, 设置为输出模式的引脚和用作复用功能的输出引脚都不连接内部上拉电阻。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器清零(00H)。

图 4-18. 上拉电阻选择寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位值	R/W
PU1	0	0	0	0	PU13	PU12	0	0	FF31H	00H	R/W
PU3	0	0	0	PU34	PU33	PU32	PU31	0	FF33H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	FF34H	00H	R/W
PU10	0	0	0	0	0	0	PU101	PU100	FF3AH	00H	R/W
PU11	0	0	0	0	PU113	PU112	0	0	FF3BH	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	FF3CH	00H	R/W
PU14	0	0	0	0	PU143	PU142	PU141	PU140	FF3EH	00H	R/W
PU15	0	0	0	0	PU153	PU152	PU151	PU150	FF3FH	00H	R/W

PUmn	Pmn 引脚内部上拉电阻选择 (m = 1, 3, 4, 10 ~ 12, 14, 15; n = 0 ~ 4)
0	不连接内部上拉电阻
1	连接内部上拉电阻

**(4) 端口功能寄存器 1 (PF1)**

此寄存器用来设置 P13/TxD0/KR4/<TxD6>引脚的功能。

PF1 可以由 1 位或 8 位存储器操作指令设置。

复位信号的产生可将 PF1 清零 (00H)。

**备注** 带有箭头(< >)的功能可以通过设置输入切换控制寄存器 (ISC) 进行分配。

图 4-19. 端口功能寄存器 1 (PF1)的格式

地址: FF20H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PF1	0	0	0	0	PF13	0	0	0

PF13	端口(P13), 案件中断(KR4), UART0, 和 UART6 输出规定
0	用作P13 或 KR4
1	用作TxD0 ~ TxD6

**(5) 端口功能寄存器 2 (PF2)**

此寄存器用来设置 P20 ~ P25 引脚作为端口引脚 (除 segment 输出引脚) 或 segment 输出引脚。

PF2 可以由 1 位或 8 位存储器操作指令设置。

复位信号的产生可将 PF2 清零 (00H)。

图 4-20. 端口功能寄存器 2 (PF2)的格式

地址: FFB5H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PF2	0	0	PF25	PF24	PF23	PF22	PF21	PF20

PF2n	指定端口/segment 输出
0	用作端口(除segment 输出)
1	用作segment 输出

**备注** n = 0 ~ 5

**(6) 端口功能寄存器 ALL (PFALL)**

此寄存器用来设置 P10, P11, P14 和 P15 引脚作为端口引脚（除 segment 输出引脚）或 segment 输出引脚。  
 PFALL 可以由 1 位或 8 位存储器操作指令设置。  
 复位信号的产生可将 PFALL 清零（00H）。

图 4-21. 端口功能寄存器 ALL (PFALL)的格式

地址: FFB6H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PFALL	0	PF15ALL	PF14ALL	0	PF11ALL	PF10ALL	0	0

PFnALL	指定端口/segment 输出
0	用作端口(除segment 输出)
1	用作segment 输出

备注 n = 10, 11, 14, 15

**(7) A/D 端口配置寄存器 0 (ADPC0)**

此寄存器将 P20/ANI0 ~ P25/ANI5 引脚切换为 A/D 转换器的模拟输入或数字 I/O 端口。

ADPC0 可以由 1 位或 8 位存储器操作指令设置。

复位信号的产生可将本寄存器置为 08H。

**注意事项** 复位释放后设置的数据如图 4-22 所示。

图 4-22. A/D 端口配置寄存器 0 (ADPC0)的格式

地址: FF8FH 复位后: 08H R/W

符号	7	6	5	4	3	2	1	0
ADPC0	0	0	0	0	0	ADPC02	ADPC01	ADPC00

ADPC02	ADPC01	ADPC00	数字I/O(D)/模拟(A)输入					
			P25 /ANI5	P24 /ANI4	P23 /ANI3	P22 /AN2	P21 /ANI1	P20 /ANI0
0	0	0	A	A	A	A	A	A
0	0	1	A	A	A	A	A	D
0	1	0	A	A	A	A	D	D
0	1	1	A	A	A	D	D	D
1	0	0	A	A	D	D	D	D
1	0	1	A	D	D	D	D	D
1	1	0	D	D	D	D	D	D
其它情况			禁止设置					

- 注意事项**
1. 通过使用端口模式寄存器 2(PM2)将用于 A/D 转换的通道设置为输入模式。
  2. 由 ADPC 设置数字 I/O，不能由 ADS, ADDS1 或 ADDS0 设置。
  3. 如果将数据写入 ADPC，则产生一个等待周期。当 CPU 使用子系统时钟并且外部硬件时钟停止时，不要将数据写入 ADPC。详细情况可参见 第二十九章 等待注意事项。
  4. 如果 ANI0/P20/SEG21 ~ ANI5/P25/SEG16 引脚经 PF2 设置为 segment 输出，不必考虑 ADPC0 设置(仅适用于  $\mu$ PD78F041x)。

## 4.4 端口功能操作

对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** 1 位操作指令的情况下，虽然只操作了一个信号位，这个端口也要作为 8 位单元来访问。因此，对于一个输入输出引脚混合的端口，指定为输入状态的引脚，其输出锁存器的内容是不确定的，即使是未操作的位。

### 4.4.1 写入 I/O 端口

#### (1) 输出模式

使用传送指令对输出锁存器进行写操作，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被写入。

当复位信号产生时输出锁存器的内容被清零。

#### (2) 输入模式

使用传送指令对输出锁存器进行写操作，因为输出缓冲器处于关闭状态，所以引脚状态不会改变。

一旦数据写入输出锁存器，它将一直保存到新数据被写入。

### 4.4.2 读取 I/O 端口

#### (1) 输出模式

使用传送指令读取的是输出锁存器的内容。输出锁存器的内容不会改变。

#### (2) 输入模式

使用传送指令读取的是引脚状态。输出锁存器的内容不会改变。

### 4.4.3 I/O 端口的操作

#### (1) 输出模式

对输出锁存器执行一个操作时，操作结果写入输出锁存器。而输出锁存器的内容则从引脚输出。

一旦数据写入输出锁存器，它将一直保存到新数据被写入。

当复位信号的产生时输出锁存器的内容被清零。

#### (2) 输入模式

读取引脚电平，对它的内容执行操作，操作的结果写入输出锁存器，因为输出缓冲器处于关闭状态，所以引脚状态不会改变。

## 4.5 使用复用功能时，PFALL, PF2, PF1, ISC, 端口模式寄存器和输出锁存器的设置

为使用端口引脚的复用功能，应按表 4-5 所示对端口模式寄存器和输出锁存器进行设置。

表 4-5. 使用复用功能时，PFALL, PF2, PF1, ISC 端口模式寄存器和输出锁存器的设置(1/2)

引脚名称	复用功能		PFALL, PF2 <sup>注4</sup>	PF1	ISC	PM <sub>xx</sub>	P <sub>xx</sub>
	功能名称	I/O					
P12	KR3	输入	–			1	×
	RxD0	输入	–			1	×
	<RxD6>	输入	–		ISC4 = 0, ISC5 = 1 <sup>注5,7</sup>	1	×
P13 <sup>注9</sup>	KR4	输入	–	PF13 = 0		1	×
	TxD0	输出	–	PF13 = 1		0	×
	<TxD6>	输出	–	PF13 = 1	ISC4 = 0, ISC5 = 1	0	×
P20 ~ P25 <sup>注2</sup>	SEG21 ~ SEG16	输出	1			×	×
	ANI0 ~ ANI5 <sup>注1</sup>	输入	0			1	×
P31	TOH1	输出	–			0	0
	INTP3	输入	–			1	×
P32	TOH0	输出	–			0	0
	MCGO	输出	–			0	0
P33	TI000	输入	–		ISC1 = 0	1	×
	RTCDIV	输出	–			0	0
	RTCCL	输出	–			0	0
	BUZ	输出	–			0	0
	INTP2	输入	–			1	×
P34	TI52	输入	–		注 6	1	×
	TI010	输入	–			1	×
	TO00	输出	–			0	0
	RTC1HZ	输出	–			0	0
	INTP1	输入	–			1	×
P40	KR0	输入	–			1	×
	V <sub>LC3</sub> <sup>注8</sup>	输入	–			×	×
P100, P101	SEG4, SEG5	输出	1			×	×
P112	SEG6	输出	1		ISC3 = 0	×	×
	TxD6	输出	0		ISC3 = 1, ISC4 = ISC5 = 0	0	1

(注和备注在下一页。)

表 4-5. 使用复用功能时, PFALL, PF2, PF1, ISC 端口模式寄存器和输出锁存器的设置(2/2)

引脚名称	复用功能		PFALL, PF2 <sup>注4</sup>	ISC	PM <sub>xx</sub>	P <sub>xx</sub>
	功能名称	I/O				
P113	SEG7	输出	1	ISC3 = 0	×	×
	RxD6	输入	0	ISC3 = 1, ISC4 = ISC5 = 0 <sup>注5,7</sup>	1	×
P120	EXLVI	输入	–		1	×
	INTP0	输入	–	ISC0 = 0	1	×
P121	X1 <sup>注3</sup>	–	–		×	×
	OCD0A	–	–		×	×
P122	X2 <sup>注3</sup>	–	–		×	×
	EXCLK <sup>注3</sup>	输入	–		×	×
	OCD0B	–	–		×	×
P123	XT1 <sup>注3</sup>	–	–		×	×
P124	XT2 <sup>注3</sup>	–	–		×	×
P140 ~ P143	SEG8 ~ SEG11	输出	1		×	×
P150 ~ P153	SEG12 ~ SEG15	输出	1		×	×

(注和备注在下一页。)

- 注 1. 仅适用于 $\mu$ PD78F041x。  
 2. 通过使用端口功能寄存器 2 (PF2)、A/D 端口配置寄存器 0 (ADPC0)、端口模式寄存器 2 (PM2)、模拟输入通道规格寄存器(ADS), 可以选择 P20/ANI0 ~ P25/ANI5 引脚的功能。

表 4-6. 设置 P20/SEG21/ANI0<sup>注</sup> ~ P25/SEG16/ANI5<sup>注</sup>引脚功能

PF2	ADPC0 <sup>Note</sup>	PM2	ADS	P20/SEG21/ANI0 <sup>注</sup> ~ P25/SEG16/ANI5 <sup>注</sup> 引脚
数字/模拟 选择	模拟输入选择	输入模式	不选择 ANI。	模拟输入 (不得修改)
			选择 ANI。	模拟输入 (可由逐次逼近型 A/D 转换器修改)
		输出模式	—	禁止设置
	数字 I/O 选择	输入模式	—	数字输入
		输出模式	—	数字输出
SEG 输出选择	—	—	—	Segment 输出

注 仅对 $\mu$ PD78F041x。

- 当使用 P121 ~ P124 引脚连接主系统时钟 (X1,X2) 或副系统时钟 (XT1,XT2) 的振荡器, 或者输入该主系统时钟的外部时钟 (EXCLK) 时, 必须通过使用时钟操作模式选择寄存器 (OSCCTL) 对 X1 振荡模式、XT1 振荡模式或者外部时钟输入模式进行设置 (如需了解详细信息, 可参见 5.3 (1) 时钟操作模式选择寄存器 (OSCCTL) 和 (3) 副系统时钟引脚操作模式设置)。OSCCTL 的复位值为 00H (P121 ~ P124 都用作输入端口引脚)。
- 每个端口相应的目标寄存器。
- 设置 ISC1 = 1, RxD6 作为 TI000 的输入源。
- 设置 ISC2 = 1, 可由 TMH2 控制 TM52 输入。
- 设置 ISC0 = 1, RxD6 作为 INTP0 的输入源。
- 当 P40/KR0/V<sub>LC3</sub> 引脚用作 V<sub>LC3</sub> 时, 设置为 1/4 分频模式。当它用作端口功能(P40)或按键中断功能 (KR0)时, 设置为其他分频模式。
- 用作端口功能时设置 PF13 = 0。

## 备注

- x: 不必考虑  
 -: 不提供  
 PMxx: 端口模式寄存器  
 Pxx: 端口输出锁存
- 箭头 (< >) 内的功能由输入转换控制寄存器 (ISC)分配。
- X1, X2 引脚用于片上调试模式设定引脚 (OCD1A, OCD1B)。详情请参看 第二十五章 片上调试功能。

## 第五章 时钟发生器

### 5.1 时钟发生器的功能

时钟发生器用于产生时钟，并提供给 CPU 和外部硬件设备。  
可以使用以下三种系统时钟和时钟振荡器。

#### (1) 主系统时钟

##### <1> X1 振荡器

通过连接一个振荡器到 X1 和 X2，该振荡电路产生  $f_x = 2 \sim 10\text{MHz}$  的时钟。  
通过执行 STOP 指令或设置主 OSC 控制寄存器(MOC)，可以停止振荡。

##### <2> 内部高速振荡器

这个振荡电路产生一个  $f_{RH} = 8\text{ MHz (TYP.)}$  的时钟。复位释放后，CPU 总是使用这个内部高速振荡时钟进行操作。可以通过执行 STOP 指令或者使用内部振荡模式寄存器(RCM)停止其振荡。

外部主系统时钟( $f_{EXCLK} = 2 \sim 10\text{MHz}$ )也可以通过 OCD0B/EXCLK/X2/P122 引脚提供。可以通过执行 STOP 指令或者使用 RCM 禁止外部主系统时钟输入。

可以通过使用主时钟模式寄存器(MCM)选择内部高速系统时钟(X1 时钟或者外部主系统时钟)或内部高速振荡时钟，作为主系统时钟。

#### (2) 副系统时钟

##### • 副系统时钟振荡器

通过在 XT1 和 XT2 之间连接一个 32.768kHz 的振荡器，该电路以  $f_{XT} = 32.768\text{ kHz}$  的频率进行振荡。通过使用处理器时钟控制寄存器(PCC)和时钟操作模式选择寄存器(OSCCTL)，可以停止振荡。

备注	1. $f_x$ :	X1 时钟振荡频率
	2. $f_{RH}$ :	内部高速振荡时钟频率
	3. $f_{EXCLK}$ :	外部主系统时钟频率
	4. $f_{XT}$ :	XT1 时钟振荡频率

**(3) 内部低速振荡时钟(看门狗定时器时钟)**

## • 内部低速振荡器

该电路以  $f_{RL} = 240 \text{ kHz}$  (TYP.)的时钟振荡。复位释放后，内部低速振荡时钟总是启动操作。

当通过选项字节设置“内部低速振荡器可由软件停止”时，可以通过使用内部振荡模式寄存器(RCM)停止其振荡。

内部低速振荡时钟不能作为 CPU 时钟。如下硬件使用内部低速振荡时钟。

- 看门狗定时器
- 8 位定时器 H1 (如果选择  $f_{RL}$ ,  $f_{RL}/2^7$  或  $f_{RL}/2^9$  作为计数时钟)
- LCD 控制器/驱动器 (如果选  $f_{RL}/2^3$  为 LCD 时钟源)

**备注**      $f_{RL}$ : 内部低速振荡时钟频率

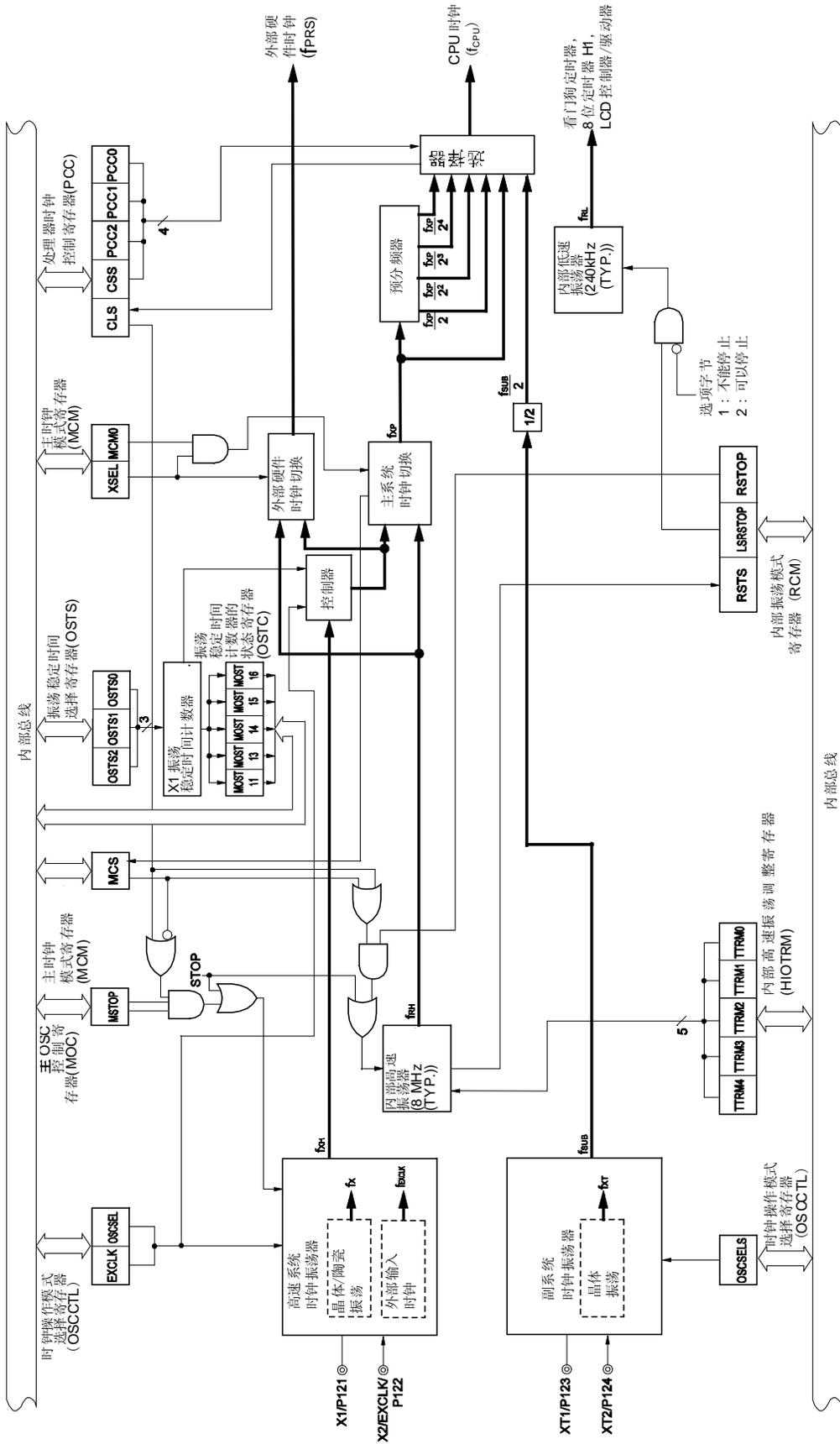
**5.2 时钟发生器的配置**

时钟发生器包括以下硬件。

**表 5-1. 时钟发生器的配置**

Item	构成
控制寄存器	时钟操作模式选择寄存器(OSCCTL) 处理器时钟控制寄存器 (PCC) 内部振荡模式寄存器 (RCM) 主 OSC 控制寄存器 (MOC) 主时钟模式寄存器 (MCM) 振荡稳定时间计数器的状态寄存器(OSTC) 振荡稳定时间选择寄存器(OSTS) 内部高速振荡调整寄存器(HIOTRM)
振荡器	X1 振荡器 XT1 振荡器 内部高速振荡器 内部低速振荡器

图 5-1. 时钟发生器的框图



备注	1. fx:	X1 时钟振荡频率
	2. frH:	内部高速振荡时钟频率
	3. fEXCLK:	外部主系统时钟频率
	4. fXH:	高速系统时钟频率
	5. fXP:	主系统时钟频率
	6. fPRS:	外部硬件时钟频率
	7. fCPU:	CPU 时钟频率
	8. fXT:	XT1 时钟振荡频率
	9. fSUB:	副系统时钟频率
	10. frL:	内部低速振荡时钟频率

### 5.3 控制时钟发生器的寄存器

以下八种寄存器用于控制时钟发生器。

- 时钟操作模式选择寄存器 (OSCCTL)
- 处理器时钟控制寄存器 (PCC)
- 内部振荡模式寄存器 (RCM)
- 主 OSC 控制寄存器 (MOC)
- 主时钟模式寄存器 (MCM)
- 振荡稳定时间计数器的状态寄存器 (OSTC)
- 振荡稳定时间选择寄存器 (OSTS)
- 内部高速振荡调整寄存器 (HIOTRM)

#### (1) 时钟操作模式选择寄存器 (OSCCTL)

该寄存器用于选择高速系统和副系统时钟的操作模式，以及内置振荡器的获取方式。可由一个 1 位或 8 位存储器操作指令设置 OSCCTL。复位信号的产生可将这个寄存器设置为 00H。

图 5-2. 时钟操作模式选择寄存器的格式(OSCCTL)

地址: FF9FH 复位后: 00H R/W

符号	<7>	<6>	5	<4>	3	2	1	0
OSCCTL	EXCLK	OSCSEL	0	OSCSELS	0	0	0	0

EXCLK	OSCSEL	高速系统时钟引脚操作模式	P121/X1 引脚	P122/X2/EXCLK 引脚
0	0	输入端口模式	输入端口	
0	1	X1 振荡模式	晶体/陶瓷振荡器连接	
1	0	输入端口模式	输入端口	
1	1	外部时钟输入模式	输入端口	输入端口

**注意事项** 若要修改 EXCLK 和 OSCSEL, 必须确保主 OSC 控制寄存器(MOC)的第 7 位(MSTOP) =1(X1 振荡器停止或禁止使用来自 EXCLK 引脚的外部时钟)。

必须将位 0 ~ 3 和位 5 清零。

**备注** f<sub>XH</sub>: 高速系统时钟振荡频率

**(2) 处理器时钟控制寄存器(PCC)**

PCC 用于选择 CPU 时钟、分频比和副系统时钟的操作模式。

由 1 位或 8 位存储器操作指令设置 PCC。

复位信号的产生可将 PCC 设置为 01H。

**图 5-3. 处理器时钟控制寄存器(PCC)的格式**

地址: FFFBH 复位后: 01H R/W<sup>※</sup>

符号	7	6	<5>	<4>	3	2	1	0
PCC	0	0	CLS	CSS	0	PCC2	PCC1	PCC0

CLS	CPU 时钟状态
0	主系统时钟
1	副系统时钟

CSS	PCC2	PCC1	PCC0	CPU 时钟 (f <sub>cpu</sub> ) 选择
0	0	0	0	f <sub>XP</sub>
	0	0	1	f <sub>XP</sub> /2 (默认)
	0	1	0	f <sub>XP</sub> /2 <sup>2</sup>
	0	1	1	f <sub>XP</sub> /2 <sup>3</sup>
	1	0	0	f <sub>XP</sub> /2 <sup>4</sup>
1	0	0	0	f <sub>SUB</sub> /2
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
其它情况				禁止设置

注 第 5 位只读。

注意事项 必须将第 3, 6 位和第 7 位清零。

- 备注
1. f<sub>XP</sub>: 主系统时钟振荡频率
  2. f<sub>SUB</sub>: 副系统时钟振荡频率

在 78K0/LC3 中执行速度最快的指令执行时间在 2 个 CPU 时钟以内。CPU 时钟(f<sub>cpu</sub>)与指令最短执行时间的关系如表 5-2 所示。

表 5-2. CPU 时钟与指令最短执行时间的关系

CPU 时钟( $f_{CPU}$ )	指令最短执行时间: $2/f_{CPU}$		
	主系统时钟		副系统时钟
	高速系统时钟 <sup>注</sup>		
	工作频率: 10 MHz	工作频率: 32.768 kHz	工作频率: 32.768 kHz
$f_{XP}$	0.2 $\mu s$	0.25 $\mu s$ (TYP.)	–
$f_{XP}/2$	0.4 $\mu s$	0.5 $\mu s$ (TYP.)	–
$f_{XP}/2^2$	0.8 $\mu s$	1.0 $\mu s$ (TYP.)	–
$f_{XP}/2^3$	1.6 $\mu s$	2.0 $\mu s$ (TYP.)	–
$f_{XP}/2^4$	3.2 $\mu s$	4.0 $\mu s$ (TYP.)	–
$f_{SUB}/2$	–	–	122.1 $\mu s$

注 主时钟模式寄存器(MCM)用于设置提供给 CPU 的主系统时钟(高速系统时钟/ 内部高速振荡时钟) (参见图 5-6)。

### (3) 副系统时钟引脚操作模式的设置

使用时钟操作模式选择寄存器(OSCCTL)的第 4 位(OSCSELS)设置副系统时钟引脚操作模式。

表 5-3. 副系统时钟引脚操作模式的设置

OSCCTL 第 4 位 OSCSELS	副系统时钟引脚操作模式	P123/XT1 引脚	P124/XT2 引脚
0	输入端口模式	输入端口	
1	XT1 振荡模式	晶体振荡器连接	

注意事项 改变 OSCSELS 的当前值时, 应确保处理器时钟控制寄存器(PCC)的第 5 位(CLS) = 0 (CPU 使用主系统时钟)。

**(4) 内部振荡模式寄存器 (RCM)**

内部振荡模式寄存器用于设置内部振荡器操作模式。

由 1 位或 8 位存储器操作指令设置 RCM。

复位信号的产生可将该寄存器设置为 80H<sup>※1</sup>。

图 5-4. 内部振荡模式寄存器(RCM)的格式

地址: FFA0H 复位后: 80H<sup>※1</sup> R/W<sup>※2</sup>

Symbol	<7>	6	5	4	3	2	<1>	<0>
RCM	RSTS	0	0	0	0	0	LSRSTOP	RSTOP

RSTS	内部高速振荡器状态
0	等待内部高速振荡器精确稳定
1	内部高速振荡器的稳定操作

LSRSTOP	内部低速振荡器振荡/停止
0	内部低速振荡器振荡
1	内部低速振荡器停止

RSTOP	内部高速振荡器振荡/停止
0	内部高速振荡器振荡
1	内部高速振荡器停止

- 注
1. 复位释放后寄存器的值立即变为 00H，但在内部高速振荡器稳定后又自动变为 80H。
  2. 第 7 位为只读。

**注意事项** 当设置 RSTOP =1 时，必须确保 CPU 使用的不是内部高速振荡时钟。特别地，在下列任一条件下，可以设置 RSTOP =1。

- 当 MCS = 1 (当 CPU 使用高速系统时钟时)
- 当 CLS = 1 (当 CPU 使用副系统时钟时)

此外，在将 RSTOP 设置为 1 之前应停止正在使用内部高速振荡时钟操作的外部硬件。

**(5) 主 OSC 控制寄存器 (MOC)**

MOC 用于选择高速系统时钟的操作模式。

当 CPU 不使用高速系统时钟时，该寄存器用于停止 X1 振荡器或禁止 EXCLK 引脚的外部时钟输入。

可以用 1 位或 8 位存储器操作指令设置 MOC。

复位信号的产生可将该寄存器设置为 80H。

图 5-5. 主 OSC 控制寄存器 (MOC)的格式

地址: FFA2H 复位后: 80H R/W

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

MSTOP	高速系统时钟操作控制	
	X1 振荡模式	X1 振荡模式
0	X1 振荡器操作	X1 振荡器操作
1	X1 振荡器停止	X1 振荡器停止

- 注意事项**
1. 设置 **MSTOP = 1** 时，必须确保 CPU 使用的不是高速系统时钟。特别地，在下列任一条件下，可以设置 **MSTOP = 1**。
    - 当 **MCS = 0** (当 CPU 使用内部高速振荡时钟时)
    - 当 **CLS = 1** (当 CPU 使用副系统时钟时)
 此外，在设置 **MSTOP** 为 1 之前应停止正在使用高速系统时钟操作的外部硬件。
  2. 当时钟操作模式选择寄存器(**OSCCTL**)的第 6 位(**OSCSEL**)= 0(I/O 端口模式)时，不要将 **MSTOP** 清零。
  3. 外部硬件时钟停止时外部硬件不能操作。外部硬件时钟停止后，若要恢复外部硬件的操作，则必须初始化外部硬件。

**(6) 主时钟模式寄存器 (MCM)**

MCM 用于选择提供给 CPU 的主系统时钟和提供给外部硬件的时钟。

由 1 位或 8 位存储器操作指令设置 MCM。

复位信号的产生可对寄存器清零(00H)。

图 5-6. 主时钟模式寄存器 (MCM) 的格式

符号: FFA1H 复位后: 00H R/W<sup>注</sup>

符号	7	6	5	4	3	<2>	<1>	<0>
MCM	0	0	0	0	0	XSEL	MCS	MCM0

XSEL	MCM0	主系统和外部硬件的时钟选择	
		主系统时钟 (f <sub>XP</sub> )	外部硬件时钟 (f <sub>PRS</sub> )
0	0	内部高速振荡时钟(f <sub>RH</sub> )	内部高速振荡时钟 (f <sub>RH</sub> )
0	1		高速系统时钟 (f <sub>XH</sub> )
1	0	高速系统时钟 (f <sub>XH</sub> )	
1	1		

MCS	主系统时钟状态
0	使用内部高速振荡时钟
1	使用高速系统时钟

注 第 1 位只读。

**注意事项 1.** XSEL 在复位释放后只能被修改一次。

**2.** 无论 XSEL 和 MCM0 如何设置, 除了 f<sub>PRS</sub> 以外的一个时钟将被提供给如下外围硬件功能

- 看门狗定时器(使用内部低速振荡时钟)
- 选择“f<sub>RL</sub>”, “f<sub>RL</sub>/2<sup>7</sup>”, 或者 “f<sub>RL</sub>/2<sup>9</sup>”作为 8 位定时器 H1 的计数时钟(使用内部低速振荡时钟)
- 选择 “f<sub>RL</sub>/2<sup>3</sup>” 作为 LCD 控制器/驱动器的 LCD 时钟源(使用内部低速振荡时钟)
- 选择外部时钟作为外部硬件的时钟源

(以下情况除外: 选择 TM0n (n = 0, 1) 的外部计数时钟(TI00n 有效沿)时)

**(7) (7) 振荡稳定时间计数器的状态寄存器 (OSTC)**

该寄存器用于指示 X1 时钟振荡稳定时间计数器的计数状态。当 X1 时钟振荡启动时，使用内部高速振荡时钟或副系统时钟作为 CPU 时钟，X1 时钟振荡稳定时间可以被检测。

可以由 1 位或 8 位存储器操作指令读取 OSTC 的内容。

复位释放后(由  $\overline{\text{RESET}}$  输入、POC、LVI、和 WDT 进行复位)，通过 STOP 指令，以及设置 MSTOP(MOC 寄存器的第 7 位) = 1，可以对 OSTC 清零(00H)。

图 5-7. 振荡稳定时间计数器的状态寄存器(OSTC)的格式

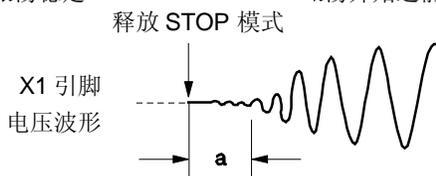
地址: FFA3H 复位后: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST 11	MOST 13	MOST 14	MOST 15	MOST 16	振荡稳定时间状态			
					$f_x = 2 \text{ MHz}$	$f_x = 5 \text{ MHz}$	$f_x = 10 \text{ MHz}$	
1	0	0	0	0	$2^{11}/f_x \text{ min.}$	1.02 ms min.	409.6 $\mu\text{s}$ min.	204.8 $\mu\text{s}$ min.
1	1	0	0	0	$2^{13}/f_x \text{ min.}$	4.10 ms min.	1.64 ms min.	819.2 $\mu\text{s}$ min.
1	1	1	0	0	$2^{14}/f_x \text{ min.}$	8.19 ms min.	3.27 ms min.	1.64 ms min.
1	1	1	1	0	$2^{15}/f_x \text{ min.}$	16.38 ms min.	6.55 ms min.	3.27 ms min.
1	1	1	1	1	$2^{16}/f_x \text{ min.}$	32.77 ms min.	13.11 ms min.	6.55 ms min.

- 注意事项**
1. 在经过上述稳定时间后，从 MOST11 开始的位依次被设置为 1，并一直保持。
  2. 振荡稳定时间计数器值增加到由 OSTC 设置的振荡稳定时间。当内部高速振荡时钟作为 CPU 时钟，如果已进入 STOP 模式，在释放该模式时，按如下方式设置振荡稳定时间。
    - 预期的 OSTC 振荡稳定时间  $\leq$  由 OSTC 设置的振荡稳定时间。
 注意，在 STOP 模式释放后，仅将达到振荡稳定时间(由 OSTC 设置)的状态赋给 OSTC。
  3. X1 时钟振荡稳定等待时间不包括时钟振荡开始之前的时间(下图“a”表示的部分)。



**备注**  $f_x$ : X1 时钟振荡频率

**(8) 振荡稳定时间选择寄存器 (OSTS)**

OSTS 用于选择 X1 时钟振荡稳定等待时间(当 STOP 模式释放时)。

当 CPU 使用 X1 时钟且 STOP 模式释放后，等待由 OSTS 设置的时间。

当 CPU 使用内部高速振荡时钟时，应使用 OSTC 确认 STOP 模式释放后已经经历了所要求的振荡稳定时间。可以使用 OSTC 检测振荡稳定时间。

可以由 8 位存储器操作指令设置 OSTS。

复位信号的产生将 OSTS 内容设置为 05H。

图 5-8. 振荡稳定时间选择寄存器(OSTS)的格式

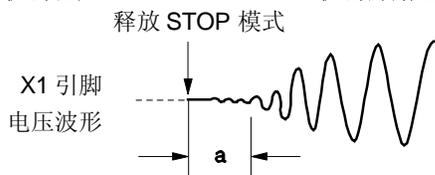
地址: FFA4H 复位后: 05H R/W

符号	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0		振荡稳定时间的选择		
				$f_x = 2 \text{ MHz}$	$f_x = 5 \text{ MHz}$	$f_x = 10 \text{ MHz}$
0	0	1	$2^{11}/f_x$	1.02 ms	409.6 $\mu\text{s}$	204.8 $\mu\text{s}$
0	1	0	$2^{13}/f_x$	4.10 ms	1.64 ms	819.2 $\mu\text{s}$
0	1	1	$2^{14}/f_x$	8.19 ms	3.27 ms	1.64 ms
1	0	0	$2^{15}/f_x$	16.38 ms	6.55 ms	3.27 ms
1	0	1	$2^{16}/f_x$	32.77 ms	13.11 ms	6.55 ms
其它情况			禁止设置			

- 注意事项**
1. 当 CPU 使用 X1 时钟时，若要设置 STOP 模式，则必须在执行 STOP 指令之前设置 OSTS。
  2. 在 X1 时钟振荡稳定时间内不要改变 OSTS 的值。
  3. 振荡稳定时间计数器值增加到由 OSTS 设置的振荡稳定时间。CPU 使用内部高速振荡时钟时，如果已进入 STOP 模式，在释放该模式时，按如下方式设置振荡稳定时间。
    - 预期的 OSTC 振荡稳定时间  $\leq$  由 OSTS 设置的振荡稳定时间。

注意，在 STOP 模式释放后，仅将达到振荡稳定时间的状态(由 OSTS 设置)赋给 OSTC。
  4. X1 时钟振荡稳定等待时间不包括时钟振荡开始之前的时间(下图“a”表示的部分)。



**备注**  $f_x$ : X1 时钟振荡频率

**(9) 内部高速振荡调整寄存器(HIOTRM)**

此寄存器用于调整内部高速振荡器的精确性。通过对内部高速振荡器频率的自测量调整其精确性，使用副系统时钟、晶体振荡器或高精度外部时钟输入定时器，例如实时计数器。

可以由 8 位存储器操作指令设置 HIOTRM。

复位信号的产生将 HIOTRM 设置为 10H。

**注意事项** 如果在精确性调整后温度或  $V_{DD}$  引脚电压变动，频率将发生波动。同样，如果 HIOTRM 寄存器设置了除初值 (10H) 以外的值，由于温度或  $V_{DD}$  电压变动，或 HIOTRM 设置值，内部高速振荡时钟的振荡精确性将超过 MIN. 和 MAX. 值，描述参见第二十七章 电气特性(标准产品)。如果温度或  $V_{DD}$  电压变动，必须调整频率精确性。

图 5-9. 内部高速振荡调整寄存器(HIOTRM)的格式

地址: FF30H 复位后: 10H R/W

符号	7	6	5	4	3	2	1	0
HIOTRM	0	0	0	TTRM4	TTRM3	TTRM2	TTRM1	TTRM0

TTRM4	TTRM3	TTRM2	TTRM1	TTRM0	时钟调整值(目标) (2.7 V ≤ VDD ≤ 5.5 V)		
					MIN.	TYP.	MAX.
0	0	0	0	0	TBD	-4.88%	TBD
0	0	0	0	1	TBD	-4.62%	TBD
0	0	0	1	0	TBD	-4.33%	TBD
0	0	0	1	1	TBD	-4.03%	TBD
0	0	1	0	0	TBD	-3.73%	TBD
0	0	1	0	1	TBD	-3.43%	TBD
0	0	1	1	0	TBD	-3.13%	TBD
0	0	1	1	1	TBD	-2.83%	TBD
0	1	0	0	0	TBD	-2.53%	TBD
0	1	0	0	1	TBD	-2.22%	TBD
0	1	0	1	0	TBD	-1.91%	TBD
0	1	0	1	1	TBD	-1.60%	TBD
0	1	1	0	0	TBD	-1.28%	TBD
0	1	1	0	1	TBD	-0.96%	TBD
0	1	1	1	0	TBD	-0.64%	TBD
0	1	1	1	1	TBD	-0.32%	TBD
1	0	0	0	0	±0% (默认值)		
1	0	0	0	1	TBD	+0.32%	TBD
1	0	0	1	0	TBD	+0.65%	TBD
1	0	0	1	1	TBD	+0.98%	TBD
1	0	1	0	0	TBD	+1.31%	TBD
1	0	1	0	1	TBD	+1.64%	TBD
1	0	1	1	0	TBD	+1.98%	TBD
1	0	1	1	1	TBD	+2.32%	TBD
1	1	0	0	0	TBD	+2.66%	TBD
1	1	0	0	1	TBD	+3.00%	TBD
1	1	0	1	0	TBD	+3.34%	TBD
1	1	0	1	1	TBD	+3.69%	TBD
1	1	1	0	0	TBD	+4.04%	TBD
1	1	1	0	1	TBD	+4.39%	TBD
1	1	1	1	0	TBD	+4.74%	TBD
1	1	1	1	1	TBD	+5.10%	TBD

注意事项 HIOTRM 寄存器指定值增加, 内部高速振荡频率速度增快, 指定值减小速度降低。不会发生值增加频率速度降低, 或值减小速度增加。

## 5.4 系统时钟振荡器

### 5.4.1 X1 振荡器

X1 振荡器采用晶体振荡器或陶瓷振荡器(2 ~ 10MHz)，连接到 X1 和 X2 引脚。

同样可以输入外部时钟。EXCLK 引脚输入时钟信号。

图 5-10 为 X1 振荡器的外部电路示例。

图 5-10. X1 振荡器的外部电路示例

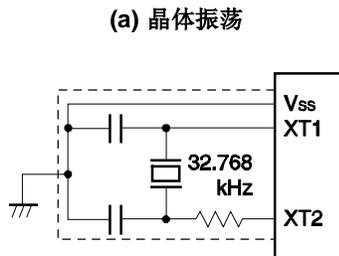


### 5.4.2 XT1 振荡器

XT1 振荡器采用晶体振荡器(标准值为：32.768 kHz)，连接到 XT1 和 XT2 引脚。

图 5-11 为 XT1 振荡器的外部电路示例。

图 5-11. XT1 振荡器的外部电路示例



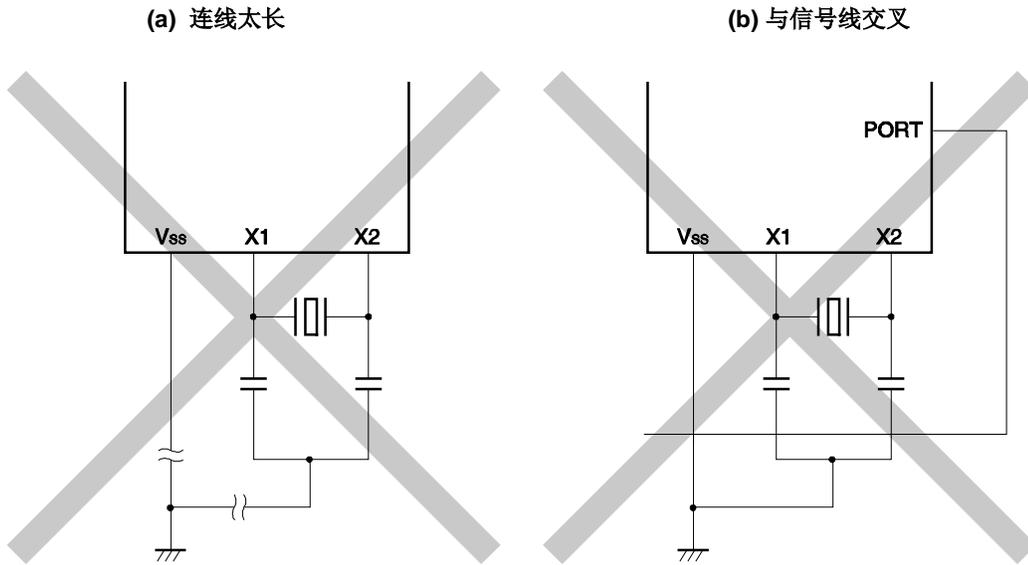
**注意事项 1.** 在使用 X1 振荡器和 XT1 振荡器时，图 5-10 和图 5-11 中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接线电容产生不利影响。

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

**注意** XT1 振荡器被设计成低振幅电路，以降低功耗。

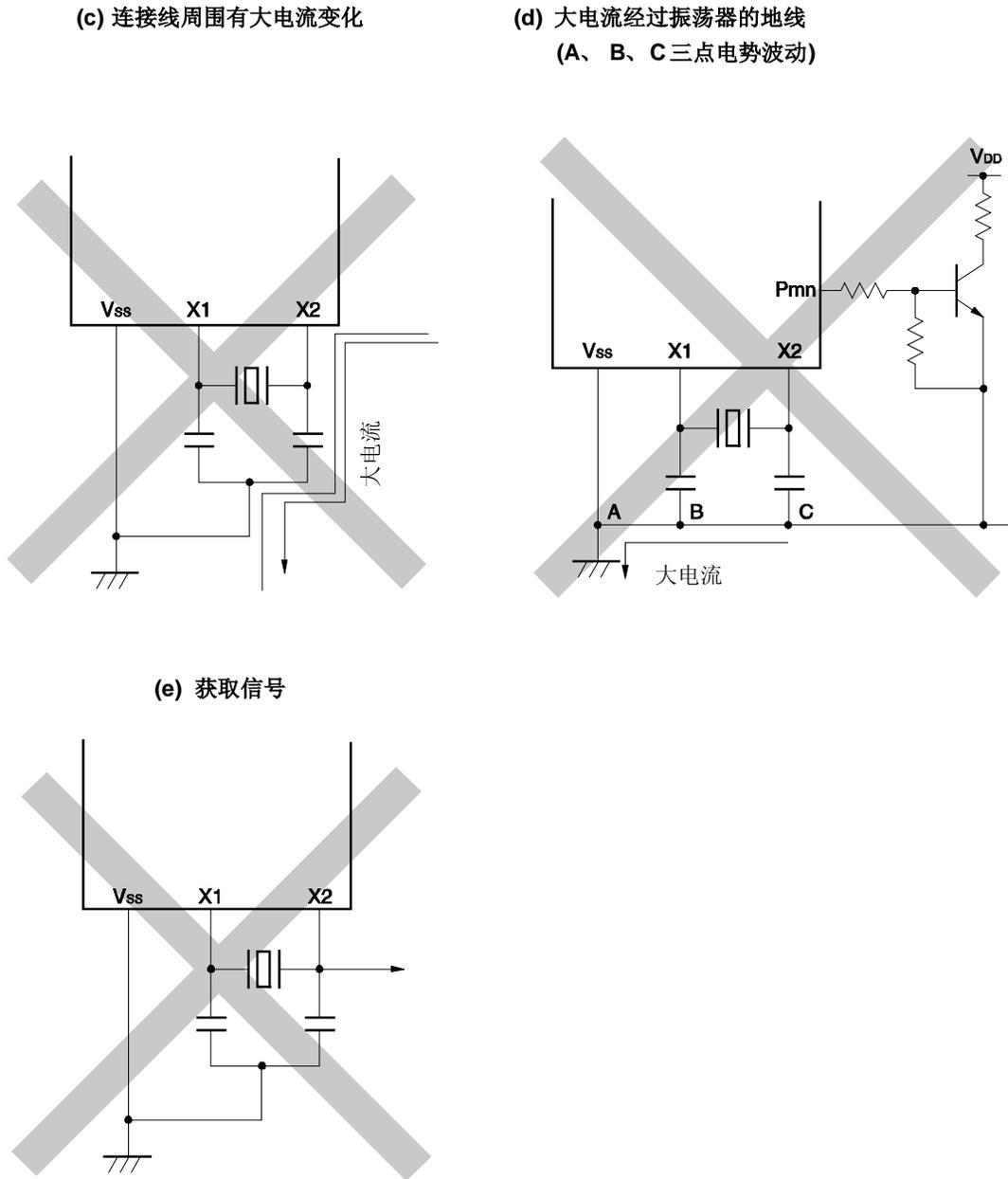
图 5-12 为不正确的振荡器连接示例。

图 5-12. 不正确的振荡器连接示例(1/2)



**备注** 在使用副系统时钟时，分别用 XT1 和 XT2 代替 X1 和 X2。串联电阻也串在 XT2 这边。

图 5-12. 不正确的振荡器连接示例(2/2)



**备注** 在使用副系统时钟时，分别用 XT1 和 XT2 代替 X1 和 X2。串联电阻也插在 XT2 这边。

**注意事项** 2. 当 X2 和 XT1 并行连接时，X2 的串扰噪音会叠加到 XT1，从而产生错误。

### 5.4.3 不使用副系统时钟

为了降低功耗，如果无需使用副系统时钟，或者不使用副系统时钟作为 I/O 端口，则可以设置 XT1 和 XT2 引脚为 I/O 模式(OSCSELS = 0)，并通过一个电阻独立连接到 V<sub>DD</sub> 或 V<sub>SS</sub>。

**备注** OSCSELS:时钟操作模式选择寄存器的第 4 位 (OSCCTL)

### 5.4.4 内部高速振荡器

78K0/LC3 产品中包含内部高速振荡器。可以通过内部振荡模式寄存器(RCM)控制振荡。

复位释放后，内部高速振荡器自动开始振荡(8 MHz (TYP.))。

### 5.4.5 内部低速振荡器

78K0/LC3 产品中包含内部低速振荡器。

内部低速振荡时钟只作为看门狗定时器、8 位定时器 H1 和 LCD 控制器/驱动器的时钟使用。内部低速振荡时钟不能用作 CPU 时钟。

可以通过选项字节选择内部低速振荡器“可由软件停止”或“不能停止”。如果设置“可由软件停止”，则可由内部振荡模式寄存器(RCM)控制振荡。

复位释放后，内部低速振荡器自动产生振荡，同时如果使用选项字节允许看门狗定时器操作，则可以驱动看门狗定时器 (240 kHz (TYP.))。

### 5.4.6 预分频器

当 CPU 使用主系统时钟时，通过分频主系统时钟，预分频器可以产生多种时钟。

## 5.5 时钟发生器的操作

时钟发生器用于产生以下几种时钟，并控制 CPU 的操作模式，如待机模式。(见图 5-1)。

- 主系统时钟  $f_{XP}$ 
  - 高速系统时钟  $f_{XH}$ 
    - X1 时钟  $f_X$
    - 外部主系统时钟  $f_{EXCLK}$
  - 内部高速振荡时钟  $f_{RH}$
- 副系统时钟  $f_{SUB}$ 
  - XT1 时钟  $f_{XT}$
- 内部低速振荡时钟  $f_{RL}$
- CPU 时钟  $f_{CPU}$
- 外部硬件时钟  $f_{PRS}$

在 78K0/LC3 中，当复位释放后内部高速振荡器输出时，CPU 开始操作，因此具有以下特点。

### (1) 增强安全功能

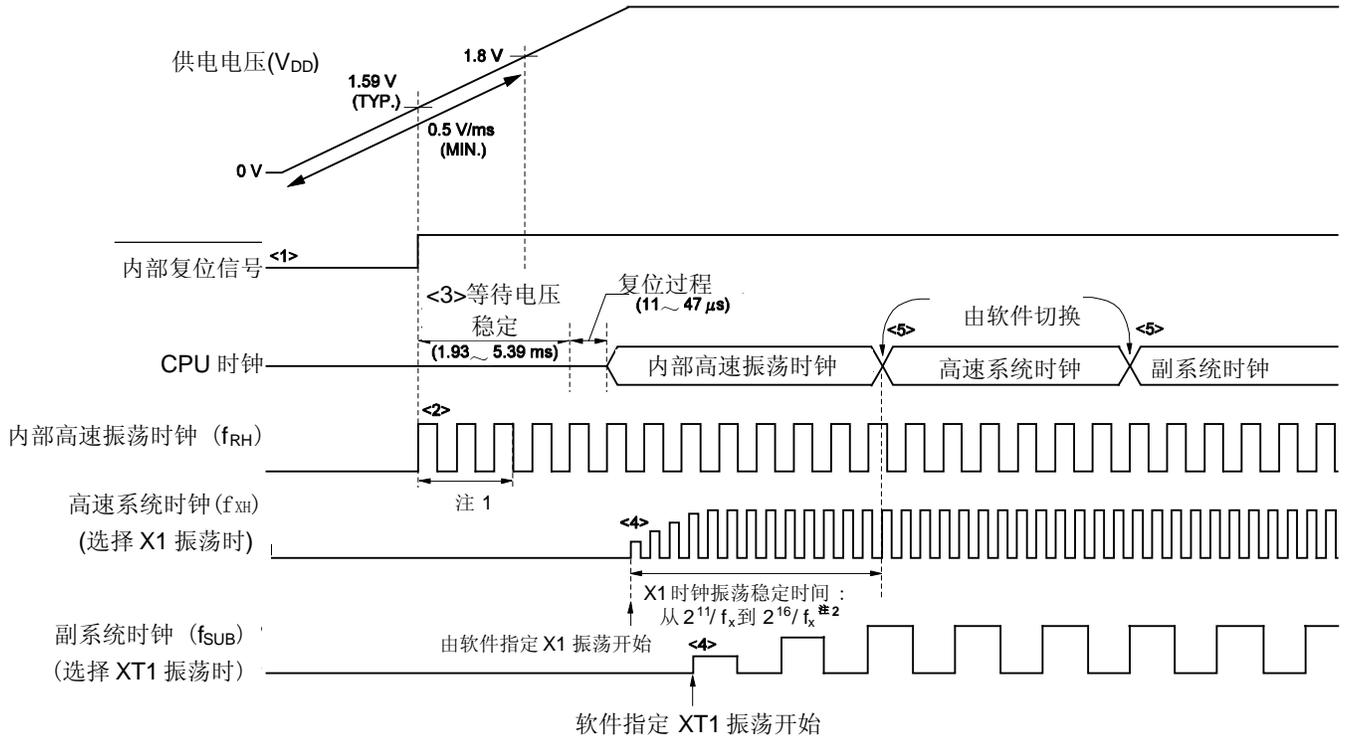
如果 X1 时钟被默认设置为 CPU 时钟，在 X1 时钟遭到损坏或连接错误时设备不能操作，因此复位释放后也不能操作。但是，如果 CPU 的初始时钟是内部高速振荡时钟，则在复位释放后，由内部高速振荡时钟启动设备。这样，系统只需执行最少操作(如由软件确认复位源或在出现故障时执行安全处理)，便可以安全关闭。

### (2) 改善性能

由于 CPU 可以在不必等待 X1 时钟振荡稳定时间情况下就启动，所以总的性能得到了改善。

图 5-13 为上电时，时钟发生器的操作图。

图 5-13. 上电时的时钟发生器操作  
(设置 1.59 V POC 模式 (选项字节: POCMODE = 0))



- <1> 当电源开启时, 通过上电清零 (POC) 电路产生一个内部复位信号。
- <2> 当供电电压超过 1.59 V (TYP.), 复位释放并且内部高速振荡器自动开始振荡。
- <3> 当供电电压上升斜率为 0.5 V/ms (MAX.)时, 复位释放并且在经历了供电电压和稳压器的稳定时间后, CPU 开始使用内部高速振荡时钟, 然后进行复位处理。
- <4> 通过软件设置 X1 或 XT1 时钟的振荡开始(参见 5.6.1 高速系统时钟控制示例中的(1)和 5.6.3 副系统时钟控制示例中的(1))。
- <5> 当 CPU 时钟切换到 X1 或 XT1 时钟时, 等待时钟振荡稳定, 然后通过软件设置切换(参见 5.6.1 高速系统时钟控制示例中的(3)和 5.6.3 副系统时钟控制示例中的(3))。

注 1. 内部电压稳定时间包括内部高速振荡时钟的振荡精确稳定时间。

2. 释放复位后(如上图所示)或在 CPU 使用内部高速振荡时钟时释放 STOP 模式后, 使用振荡稳定时间计数器的状态寄存器(OSTC)来确认 X1 时钟的振荡稳定时间。如果 CPU 使用高速系统时钟(X1 振荡), 则可使用振荡稳定时间选择寄存器(OSTS)设置释放 STOP 模式时的振荡稳定时间。

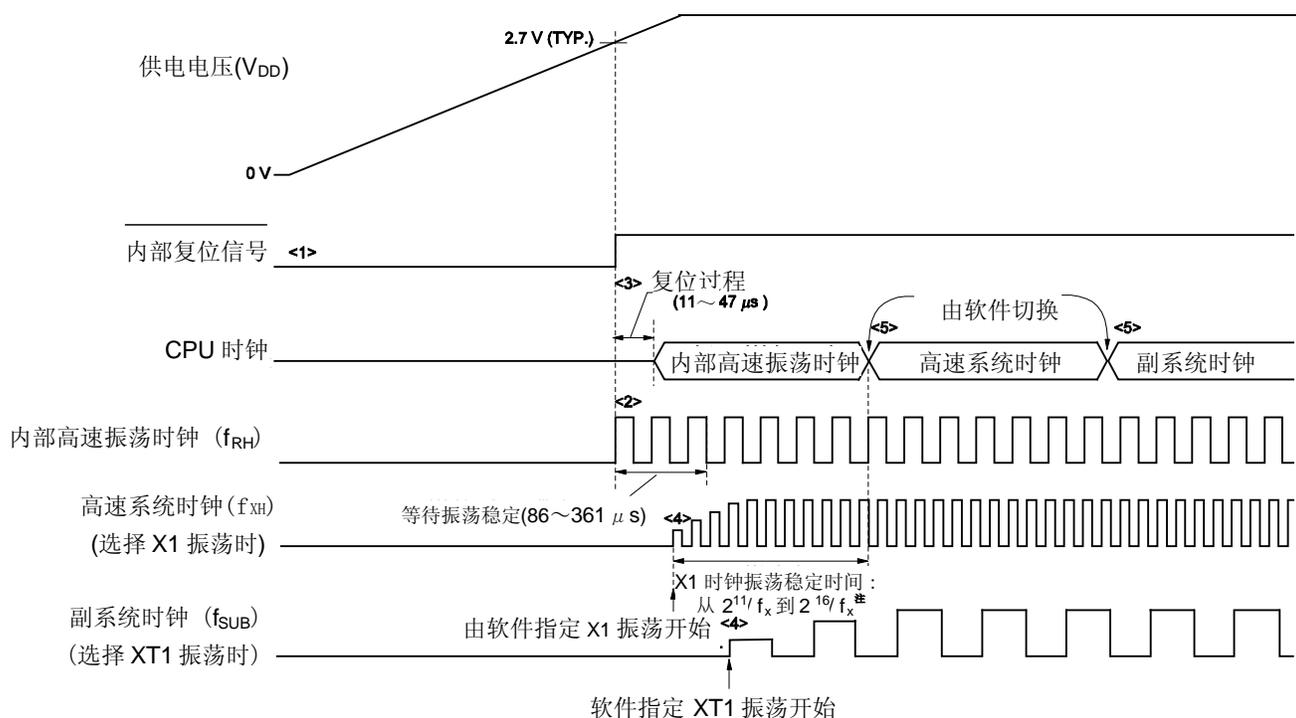
注 1. 内部电压稳定时间包括内部高速振荡时钟的振荡精确稳定时间。

2. 释放复位后(如上图所示)或在 CPU 使用内部高速振荡时钟时释放 STOP 模式后, 使用振荡稳定时间计数器的状态寄存器(OSTC)来确认 X1 时钟的振荡稳定时间。如果 CPU 使用高速系统时钟(X1 振荡), 则可使用振荡稳定时间选择寄存器(OSTS)设置释放 STOP 模式时的振荡稳定时间。

- 注意事项 1. 在供电电压达到 1.8V 之前, 如果电压上升斜率小于 0.5 V/ms (MAX.), 则输入一个低电平到电源的 RESET 引脚直至电压达到 1.8V, 或者通过使用选项字节(POCMODE = 1)设置 2.7 V/1.59 V POC 模式(见图 5-14)。通过这种方式, CPU 的操作时序与 <2> 及由 RESET 引脚释放复位后的时序相同, 如图 5-13 所示。
2. 当使用 EXCLK 引脚的外部时钟输入时, 不需要等待振荡稳定时间。

**备注** 当微控制器正在操作时，不作为 CPU 时钟使用的时钟可以通过软件设置来停止。内部高速振荡时钟和高速系统时钟可以通过执行 STOP 指令来停止(参见 5.6.1 高速系统时钟控制示例中的(4)、内部高速振荡时钟控制示例中的(3)和 5.6.3 副系统时钟控制示例中的(4))。

图 5-14. 上电时的时钟发生器操作  
(设置 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1))



- <1> 当电源开启时，通过上电清零 (POC) 电路产生一个内部复位信号。
- <2> 当供电电压超过 2.7 V (TYP.)，复位释放并且内部高速振荡器自动开始振荡。
- <3> 复位释放并进行了复位处理后，CPU 开始使用内部高速振荡时钟操作。
- <4> 通过软件设置 X1 或 XT1 时钟的振荡开始(参见 5.6.1 高速系统时钟控制示例中的(1)和 5.6.3 副系统时钟控制示例中的(1))。
- <5> 当 CPU 时钟切换到 X1 或 XT1 时钟时，等待时钟振荡稳定，然后通过软件设置切换(参见 5.6.1 高速系统时钟控制示例中的(3)和 5.6.3 副系统时钟控制示例中的(3))。

**注** 释放复位后(如上图所示)或在 CPU 使用内部高速振荡时钟时释放 STOP 模式后，使用振荡稳定时间计数器的状态寄存器(OSTC)来确认 X1 时钟的振荡稳定时间。如果 CPU 使用高速系统时钟(X1 振荡)，则可使用振荡稳定时间选择寄存器(OSTS)设置释放 STOP 模式时的振荡稳定时间。

- 注意事项**
1. 在供电电压达到 1.59V(TYP.)后，必需有 1.93 ~ 5.39 ms 电压振荡稳定时间。如果在 1.93 ms 内电源电压从 1.59 V (TYP.)上升到 2.7 V (TYP.)，在复位处理前自动产生 0 ~ 5.39 ms 的供电电源振荡稳定时间。
  2. 当使用 EXCLK 引脚的外部时钟输入时，不需要等待振荡稳定时间。

**备注** 当微控制器正在操作时，不作为 CPU 时钟使用的时钟可以通过软件设置来停止。内部高速振荡时钟和高速系统时钟可以通过执行 STOP 指令来停止(参见 5.6.1 高速系统时钟控制示例中的(4)、5.6.2 内部高速振荡时钟控制示例中的(3)和 5.6.3 副系统时钟控制示例中的(4))。

## 5.6 时钟控制

### 5.6.1 控制高速系统时钟示例

如下两种高速系统时钟可用。

- X1 时钟: 晶体/陶瓷振荡器连接到 X1 和 X2 引脚。
- 外部主系统时钟: 外部时钟输入到 EXCLK 引脚。

不使用高速系统时钟时，X1/P121 和 X2/EXCLK/P122 引脚可用作 I/O 端口引脚。

**注意事项** 复位释放后，OCD0A/X1/P121 和 OCD0B/X2/EXCLK/P122 引脚为 I/O 端口模式。

下面描述的示例是以下情况时的设置过程示例。

- (1) X1 时钟振荡
- (2) 使用外部主系统时钟
- (3) 将高速系统时钟作为 CPU 时钟和外部硬件时钟
- (4) 停止高速系统时钟

#### (1) X1 时钟振荡时设置过程示例

- <1> 设置 P121/X1 和 P122/X2/EXCLK 引脚，并选择 X1 时钟或外部时钟(OSCCTL 寄存器)，当 EXCLK 清零且 OSCSEL 置 1 时，从端口模式切换到 X1 振荡模式

EXCLK	OSCSEL	高速系统时钟引脚操作模式	P121/X1 引脚	P122/X2/EXCLK Pin
0	1	X1 振荡模式	晶体/陶瓷振荡器连接	

- <2> 控制 X1 时钟振荡 (MOC 寄存器)  
若 MSTOP 被清零，则 X1 振荡器开始振荡。
- <3> 等待 X1 时钟振荡稳定  
检测 OSTC 寄存器并等待所需的时间。  
在等待时间内，其它软件处理过程的执行可以使用内部高速振荡时钟。

**注意事项** 1. 在 X1 时钟运行时，不要改变 EXCLK 和 OSCSEL 的值。

2. 当供电电压达到所用时钟的操作电压后，设置 X1 时钟(参见 第二十七章 电气特性(标准产品))。

**(2) 使用外部主系统时钟时设置过程示例**

<1> 设置 P121/X1 和 P122/X2/EXCLK 引脚并选择操作模式(OSCCTL 寄存器)

当 EXCLK 和 OSCSEL 置 1 时, 从端口模式切换到外部时钟输入模式。

EXCLK	OSCSEL	高速系统时钟引脚操作模式	P121/X1 引脚	P122/X2/EXCLK 引脚
1	1	外部时钟输入模式	I/O 端口	外部时钟输入

<2> 控制外部主系统时钟输入(MOC 寄存器)

MSTOP 清零时, 允许外部主系统时钟输入。

**注意事项 1.** 在外部主系统时钟运行时, 不要改变 EXCLK 和 OSCSEL 的值。

**2.** 当供电电压达到所用时钟的操作电压后, 设置 X1 时钟(参见第二十七章 电气特性(标准产品))。

**(3) 使用高速系统时钟作为 CPU 时钟和外部硬件时钟时设置过程示例**

<1> 设置高速系统时钟振荡<sup>※</sup>

(参见 5.6.1 (1) X1 时钟振荡时设置过程示例和 (2) 使用外部主系统时钟时设置过程示例。)

**注** 当高速系统时钟已经运行时, 不需要设置<1>。

<2> 设置高速系统时钟作为主系统时钟(MCM 寄存器)

XSEL 和 MCM0 置 1 时, 高速系统时钟作为主系统时钟和外部硬件时钟。

XSEL	MCM0	主系统时钟和外部硬件时钟的选择	
		主系统时钟 (f <sub>XP</sub> )	外部硬件时钟 (f <sub>PRS</sub> )
1	1	高速系统时钟 (f <sub>XH</sub> )	高速系统时钟 (f <sub>XH</sub> )

**注意事项** 如果选择高速系统时钟作为主系统时钟, 则只有高速系统时钟可以被设置为外部硬件时钟。

<3> 设置主系统时钟作为 CPU 时钟并选择分频比(PCC 寄存器)

CSS 清零时, 主系统时钟提供给 CPU。若要选择 CPU 时钟分频比, 应使用 PCC0、PCC1 和 PCC2。

CSS	PCC2	PCC1	PCC0	CPU 时钟 (f <sub>CPU</sub> ) 选择
0	0	0	0	f <sub>XP</sub>
	0	0	1	f <sub>XP</sub> /2 (默认)
	0	1	0	f <sub>XP</sub> /2 <sup>2</sup>
	0	1	1	f <sub>XP</sub> /2 <sup>3</sup>
	1	0	0	f <sub>XP</sub> /2 <sup>4</sup>
	其它情况			

**(4) 停止高速系统时钟时设置过程示例**

高速系统时钟可以用如下两种方式停止。

- 执行 STOP 指令以设置为 STOP 模式
- MSTOP 置 1 并停止 X1 振荡(如果使用外部时钟则禁止时钟输入)

**(a) 执行 STOP 指令**

<1> 停止外部硬件的设置

停止那些不能在 STOP 模式下使用的外部硬件(需要了解不能在 STOP 模式下使用的外部硬件, 可参见第十九章 待机功能)

<2> 释放待机模式后设置 X1 时钟振荡稳定时间

当 CPU 使用 X1 时钟时, 在 STOP 指令执行前设置 OSTs。

<3> 执行 STOP 指令

执行 STOP 指令时, 系统处于 STOP 模式下并且 X1 振荡停止 (禁止外部时钟输入)。

**(b) 通过将 MSTOP 设置为 1 来停止 X1 振荡(禁止外部时钟输入)**

<1> 确认 CPU 时钟状态(PCC 和 MCM 寄存器)

根据 CLS 和 MCS 确认 CPU 没有使用高速系统时钟。

当 CLS = 0 且 MCS = 1 时, CPU 采用高速系统时钟, 因此将 CPU 时钟切换为副系统时钟或者内部高速振荡时钟。

CLS	MCS	CPU 时钟状态
0	0	内部高速振荡时钟
0	1	高速系统时钟
1	x	副系统时钟

<2> 停止高速系统时钟(MOC 寄存器)

MSTOP 置 1 时, X1 振荡停止(禁止外部时钟输入)。

注意事项 当 MSTOP= 1 时, 必须确认 MCS = 0 或 CLS = 1。此外, 必须停止正在使用高速系统时钟的外部硬件。

**5.6.2 控制内部高速振荡时钟示例**

下面描述的示例是以下情况时的设置过程示例。

- (1) 内部高速振荡时钟重新开始振荡时
- (2) 使用内部高速振荡时钟作为 CPU 时钟, 并且内部高速振荡时钟或高速系统时钟用作外部硬件时钟时
- (3) 停止内部高速振荡时钟时

**(1) 内部高速振荡时钟重新振荡时设置过程示例<sup>注1</sup>**

<1> 内部高速振荡时钟重新振荡的设置(RCM 寄存器)

RSTOP 清零时, 内部高速振荡时钟开始运行。

<2> 等待内部高速振荡时钟的振荡精确稳定时间(RCM 寄存器)。

一直到  $RSTS = 1$ <sup>注2</sup>。

**注** 1. 复位释放后, 内部高速振荡器自动开始振荡并且内部高速振荡时钟作为 CPU 时钟使用。

2. 如果 CPU 时钟和外部硬件时钟不需要很高的精度, 则无需等待时间。

**(2) 使用内部高速振荡时钟作为 CPU 时钟且内部高速振荡时钟或高速系统时钟作为外部硬件时钟时设置过程示例**

<1> • 内部高速振荡时钟重新振荡<sup>注</sup>

(参见 5.6.2 (1) 内部高速振荡时钟重新振荡时设置过程示例)。

• 高速系统时钟振荡<sup>注</sup>

(使用高速系统时钟作为外部硬件时钟时所需的设置。参见 5.6.1 (1) X1 时钟振荡时设置过程示例和(2) 使用外部主系统时钟时设置过程示例。)

**注** 当内部高速振荡时钟或高速系统时钟已经运行时, 不需要设置<1>。

<2> 选择主系统时钟和外部硬件时钟(MCM 寄存器)

使用 XSEL 和 MCM0 设置主系统时钟和外部硬件时钟。

XSEL	MCM0	主系统时钟和外部硬件时钟的选择	
		主系统时钟 (f <sub>XP</sub> )	外部硬件时钟 (f <sub>PRS</sub> )
0	0	内部高速振荡时钟 (f <sub>RH</sub> )	内部高速振荡时钟 (f <sub>RH</sub> )
0	1		
1	0		高速系统时钟(f <sub>XH</sub> )

<3> 选择 CPU 时钟分频比(PCC 寄存器)

CSS 清零时, 主系统时钟提供给 CPU。若要选择 CPU 时钟分频比, 应使用 PCC0、PCC1 和 PCC2。

CSS	PCC2	PCC1	PCC0	CPU 时钟 (f <sub>CPU</sub> ) 选择
0	0	0	0	f <sub>XP</sub>
	0	0	1	f <sub>XP</sub> /2 (默认)
	0	1	0	f <sub>XP</sub> /2 <sup>2</sup>
	0	1	1	f <sub>XP</sub> /2 <sup>3</sup>
	1	0	0	f <sub>XP</sub> /2 <sup>4</sup>
	其它情况			

**(3) 停止内部高速振荡时钟时设置过程示例**

可以用如下两种方式停止内部高速振荡时钟。

- 执行 STOP 指令设置 STOP 模式
- 将 RSTOP 置 1 且停止内部高速振荡时钟

**(a) 执行 STOP 指令**

<1> 设置外部硬件

停止那些不能在 STOP 模式下使用的外部硬件(如需了解不能在 STOP 模式下使用的外部硬件, 可参见第十九章 待机功能)。

<2> 释放待机模式后 X1 时钟振荡稳定时间的设置

CPU 使用 X1 时钟时, 在 STOP 指令执行前设置 OSTS。

<3> 执行 STOP 指令

执行 STOP 指令时, 系统处于 STOP 模式下并且停止内部高速振荡时钟。

**(b) 通过 RSTOP 置 1 停止内部高速振荡时钟**

<1> 确认 CPU 时钟状态(PCC 和 MCM 寄存器)

根据 CLS 和 MCS 确认 CPU 没有使用内部高速振荡时钟。

CLS = 0 和 MCS = 0 时, CPU 采用内部高速振荡时钟, 因此将 CPU 时钟切换为高速系统时钟或副系统时钟。

CLS	MCS	CPU 时钟状态
0	0	内部高速振荡时钟
0	1	高速系统时钟
1	x	副系统时钟

<2> 停止内部高速振荡时钟(RCM 寄存器)

RSTOP 置 1 时, 停止内部高速振荡时钟。

**注意事项** 当 RSTOP=1 时, 必须确认 MCS = 1 或 CLS = 1。此外, 必须停止正在使用内部高速振荡时钟的外部硬件。

**5.6.3 控制副系统时钟示例**

可用如下两种副系统时钟。

- XT1 时钟: 晶体/陶瓷振荡器连接到 XT1 和 XT2 引脚。

不使用副系统时钟时, XT1/P123 和 XT2/P124 引脚可用作 I/O 端口引脚。

**注意事项** 复位释放后 XT1/P123 和 XT2 /P124 引脚为在输入端口模式。

下面描述的示例是以下情况时的设置过程示例。

- (1) XT1 时钟振荡
- (2) 使用副系统时钟作为 CPU 时钟
- (3) 停止副系统时钟

**(1) XT1 时钟振荡时设置过程示例**

- <1> 设置 XT1 和 XT2 引脚并选择操作模式 (PCC 和 OSCCTL 寄存器)  
当 OSCSELS 设置为如下模式时, 将从端口模式切换到 XT1 振荡模式。

OSCSELS	副系统时钟引脚操作模式	P123/XT1 引脚	P124/XT2 引脚
1	XT1 振荡模式	连接晶体/陶瓷振荡器	

- <2> 等待副系统时钟振荡稳定  
使用定时器功能, 通过软件等待副系统时钟振荡稳定时间。

**注意事项** 在副系统时钟运行时, 不要改变 OSCSELS 的值。

**(2) 使用副系统时钟作为 CPU 时钟时设置过程示例**

- <1> 设置副系统时钟振荡<sup>\*</sup>  
(参见 5.6.3 (1) XT1 时钟振荡时设置过程示例)

**注** 当副系统时钟已经运行时, 不需要设置<1>。

- <2> 切换 CPU 时钟(PCC 寄存器)  
CSS 置 1 时, 副系统时钟提供给 CPU。

CSS	PCC2	PCC1	PCC0	CPU 时钟 (f <sub>CPU</sub> ) 选择
1	0	0	0	f <sub>SUB</sub> /2
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
	其它情况			禁止设置

**(3) 停止副系统时钟时设置过程示例**

- <1> 确认 CPU 时钟状态(PCC 和 MCS 寄存器)  
根据 CLS 和 MCS 确认 CPU 没有使用副系统时钟。  
当 CLS = 1 时, CPU 采用副系统时钟, 因此将 CPU 时钟切换为内部高速振荡时钟或高速系统时钟。

CLS	MCS	CPU 时钟状态
0	0	内部高速振荡时钟
0	1	高速系统时钟
1	x	副系统时钟

- <2> 停止副系统时钟 (OSCCTL 寄存器)  
OSCSELS 清零时, 停止 XT1 振荡。

**注意事项** 1. 清零 OSCSEL 时, 必须确认 CLS = 0。此外, 如果外围硬件使用副系统时钟, 则应停止其操作。  
2. 副系统时钟振荡不能通过使用 STOP 指令来停止。

### 5.6.4 控制内部低速振荡时钟示例

内部低速振荡时钟不能用作 CPU 时钟。

只有如下外部硬件可以使用这个时钟。

- 看门狗定时器
- 8 位定时器 H1 (选择  $f_{RL}$ ,  $f_{RL}/2^7$  或  $f_{RL}/2^9$  作为计数时钟时)
- LCD 控制器/驱动器(选择  $f_{RL}/2^3$  作为 LCD 时钟源时)

此外，可以通过选项字节选择如下操作模式。

- 内部低速振荡器不能停止
- 内部低速振荡器可由软件停止

在复位释放后内部低速振荡器自动开始振荡，并且如果通过选项字节允许看门狗定时器操作，则驱动看门狗定时器 (240 kHz (TYP.))。

#### (1) 停止内部低速振荡时钟时设置过程示例

- <1> 将 LSRSTOP 置 1 (RCM 寄存器)  
LSRSTOP 置 1 时，停止内部低速振荡时钟。

#### (2) 内部低速振荡时钟重新振荡时设置过程示例

- <1> LSRSTOP 清零 (RCM 寄存器)  
LSRSTOP 清零时，内部低速振荡时钟重新振荡。

**注意事项** 如果通过选项字节选择“不能停止内部低速振荡器”，则不能控制内部低速振荡时钟的振荡。

### 5.6.5 CPU 和外部硬件所采用的时钟

下表显示了 CPU 和外部硬件采用的时钟之间的关系及寄存器的设置。

表 5-4. CPU 和外部硬件采用的时钟及寄存器设置

时钟		XSEL	CSS	MCM0	EXCLK
CPU 时钟	外部硬件时钟				
内部高速振荡时钟		0	0	×	×
内部高速振荡时钟	X1 时钟	1	0	0	0
	外部主系统时钟	1	0	0	1
X1 时钟		1	0	1	0
外部主系统时钟		1	0	1	1
副系统时钟	内部高速振荡时钟	0	1	×	×
	X1 时钟	1	1	0	0
		1	1	1	0
	外部主系统时钟	1	1	0	1
		1	1	1	1

备注 1. XSEL: 主时钟模式寄存器(MCM)的第 2 位

2. CSS: 处理器时钟控制寄存器(PCC)的第 4 位

3. MCM0: MCM 的第 0 位

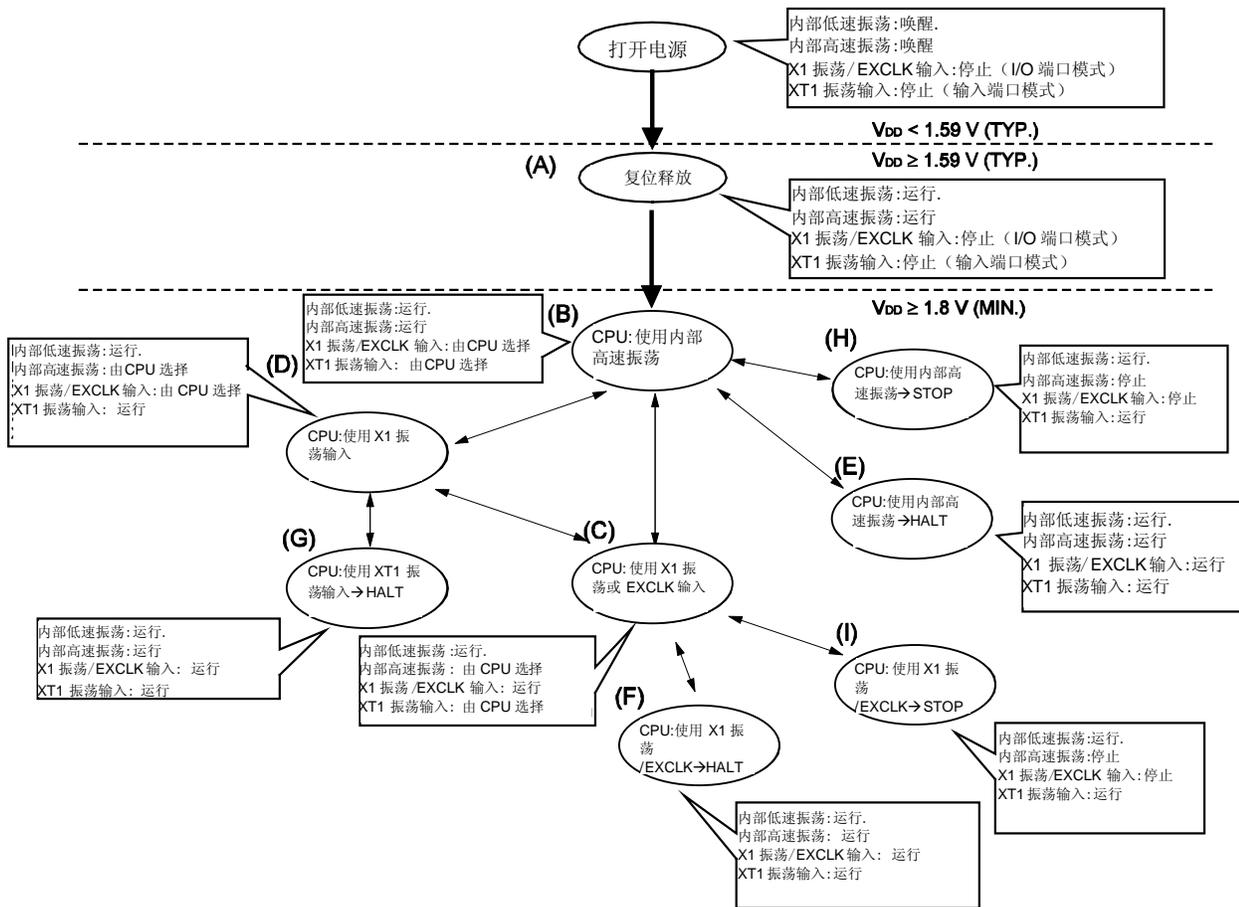
4. EXCLK: 时钟操作模式选择寄存器(OSCCTL)的第 7 位

5. ×: 不必考虑

5.6.6 CPU 时钟状态转换图

图 5-15 显示了该产品 CPU 时钟状态转换图。

图 5-15. CPU 时钟状态转换图  
(设置 1.59 V POC 模式 (选项字节: POCMODE = 0))



**备注** 在 2.7 V/1.59 V POC 模式下 (选项字节: POCMODE = 1), 当供电电压超过 2.7V(TYP.)时 CPU 时钟状态转换为上图所示的(A), 而在复位处理后(11 ~ 47 μs (TYP.))变到(B)。

表 5-5 显示了 CPU 时钟的切换过程与 SFR 寄存器设置示例

表 5-5. CPU 时钟切换与 SFR 寄存器设置示例(1/4)

(1) 复位释放后(A)CPU 使用内部高速振荡时钟(B)

状态转换	SFR 寄存器设置
(A) → (B)	SFR 寄存器不必设置 (复位释放后默认状态).

(2) 复位释放后(A)CPU 使用高速系统时钟(C)

(复位释放后 CPU 立即使用内部高速振荡时钟(B).)

( SFR 寄存器设置序列) →

SFR 寄存器设置标志	EXCLK	OSCSEL	MSTOP	OSTC 寄存器	XSEL	MCM0
状态转换						
(A) → (B) → (C) (X1 时钟)	0	1	0	必须检测	1	1
(A) → (B) → (C) (外部主时钟)	1	1	0	不必检测	1	1

注意事项 供电电压达到所用时钟的操作电压后，设置时钟(参见 第二十七章 电气特性(标准产品) )。

(3) 复位释放后(A)CPU 使用副系统时钟(D)

(复位释放后 CPU 立即使用内部高速振荡时钟(B).)

( SFR 寄存器设置序列) →

SFR 寄存器设置标志	OSCSELS	等待振荡稳定	CSS
状态转换			
(A) → (B) → (D)	1	必须	1

备注 1. 表 5-5 的(A) 到 (I)对应图 5-15 的(A)到(I)。

2. EXCLK, OSCSEL, OSCSELS:

时钟操作模式选择寄存器(OSCCTL)的第 7、6 位和第 4 位

MSTOP: 主 OSC 控制寄存器(MOC)的第 7 位

XSEL, MCM0: 主时钟模式寄存器(MCM)的第 2 位与第 0 位

CSS: 处理器时钟控制寄存器(PCC)的第 4 位

表 5-5. CPU 时钟切换与 SFR 寄存器设置示例 (2/4)

## (4) CPU 时钟从内部高速振荡时钟(B)切换到高速系统时钟(C)

( SFR 寄存器的设置顺序) →

SFR 寄存器的设置标志	EXCLK	OSCSEL	MSTOP	OSTC 寄存器	XSEL <sup>注</sup>	MCM0
状态转换						
(B) → (C) (X1 时钟)	0	1	0	必须检测	1	1
(B) → (C) (外部主时钟)	1	1	0	不必检测	1	1

如果这些寄存器已设置 则不必要
 如果 CPU 使用高速系统 时钟则不必要

注 复位释放后该标志只能被修改一次。如果已经设置了该项，则无需再设置。

注意事项 供电电压达到所用时钟的操作电压后，设置时钟(参见 第二十七章 电气特性(标准产品))。

## (5) CPU 时钟从内部高速振荡时钟(B)切换到副系统时钟(D)

( SFR 寄存器的设置顺序) →

SFR 寄存器的设置标志	OSCSELS	等待振荡稳定	CSS
状态转换			
(B) → (D)	1	必须	1

备注 1. 表 5-5 的(A) 到 (I)对应图 5-15 的(A)到(I)。

2. EXCLK, OSCSEL, OSCSELS:

时钟操作模式选择寄存器(OSCCTL)的第 7、6 位和第 4 位

MSTOP: 主 OSC 控制寄存器(MOC)的第 7 位

XSEL, MCM0: 主时钟模式寄存器(MCM)的第 2 位与第 0 位

CSS: 处理器时钟控制寄存器(PCC)的第 4 位

表 5-5. CPU 时钟切换与 SFR 寄存器设置示例 (3/4)

## (6) CPU 时钟从高速系统时钟(C)切换到内部高速振荡时钟(B)

(SFR 寄存器的设置顺序) →

SFR 寄存器的设置标志	RSTOP	RSTS	MCM0
状态转换			
(C) → (B)	0	确认该标志为 1	0

如果 CPU 使用内部高速振荡时钟则不必要

## (7) CPU 时钟从高速系统时钟(C)切换到副系统时钟(D)

(SFR 寄存器的设置顺序) →

SFR 寄存器的设置标志	OSCSLS	等待振荡稳定	CSS
状态转换			
(C) → (D)	1	必须	1

如果 CPU 使用副系统时钟则不必要

## (8) CPU 时钟从副系统时钟(D)切换到内部高速振荡时钟(B)

(SFR 寄存器的设置序列) →

SFR 寄存器的设置标志	RSTOP	RSTS	MCM0	CSS
状态转换				
(D) → (B)	0	确认该标志为 1	0	0

如果 CPU 使用内部高速振荡时钟则不  
 必要
 ↑
如果 XSEL=0 则  
不必要

- 备注**
1. 表 5-5 的(A) 到 (I) 对应图 5-15 的(A)到(I)。
  2. MCM0: 主时钟模式寄存器(MCM)的第 0 位  
 OSCSELS: 时钟操作模式选择寄存器(OSCCTL)的第 4 位  
 RSTS, RSTOP: 内部振荡模式寄存器(RCM)的第 7 位和第 0 位  
 CSS: 处理器时钟控制寄存器(PCC)的第 4 位

表 5-5. CPU 时钟切换与 SFR 寄存器设置示例 (4/4)

## (9) CPU 时钟从副系统时钟(D)切换到高速系统时钟(C)

( SFR 寄存器的设置序列) ➔

SFR 寄存器的设置标志	EXCLK	OSCSEL	MSTOP	OSTC 寄存器	XSEL <sup>※</sup>	MCM0	CSS
状态转换							
(D) → (C) (X1 时钟)	0	1	0	必须检测	1	1	0
(D) → (C) (外部主时钟)	1	1	0	不必检测	1	1	0

如果这些寄存器已设置 则不必要
 如果 CPU 使用高速系统时钟则不必要

注 复位释放后该标志只能被修改一次。如果已经设置了该项，则无需再设置。

注意事项 供电电压达到所用时钟的操作电压后，设置时钟(参见 第二十七章 电器特性(标准产品))。

## (10) • CPU 使用内部高速振荡时钟(B)时 HALT 模式(E)的设置。

- CPU 使用高速系统时钟(C)时 HALT 模式(F)的设置。
- CPU 使用副系统时钟(D)时 HALT 模式(G)的设置。

状态转换	设置
(B) → (E) (C) → (F) (D) → (G)	执行 HALT 指令

## (11) • CPU 使用内部高速振荡时钟(B)时 STOP 模式(H)的设置。

- CPU 使用高速系统时钟(C)时 STOP 模式(I)的设置。

(设置顺序) ➔

状态转换	设置	
(B) → (H) (C) → (I)	停止那些不能在 STOP 模式下使用的 外部功能	执行 STOP 指令

## 备注

1. 表 5-5 的(A) 到 (I)对应图 5-15 的(A)到(I)。
2. EXCLK, OSCSEL: 时钟操作模式选择寄存器(OSCCTL)的第 7 和 6 位  
MSTOP: 主 OSC 控制寄存器(MOC)的第 7 位  
XSEL, MCM0: 主时钟模式寄存器(MCM)的第 2 位和第 0 位  
CSS: 处理器时钟控制寄存器(PCC) 的第 4 位

## 5.6.7 CPU 时钟切换之前的状况与切换之后的处理

CPU 时钟切换之前的状况与切换之后的处理显示如下。

表 5-6. 切换 CPU 时钟

CPU 时钟		切换前的状况	切换后的处理
切换前	切换后		
内部高速振荡时钟	X1 时钟	X1 振荡稳定 • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • 经历了振荡稳定时间	• 可以停止内部高速振荡器(RSTOP = 1).
	外部主系统时钟	允许来自 EXCLK 引脚的外部时钟输入 • MSTOP = 0, OSCSEL = 1, EXCLK = 1	
X1 时钟	内部高速振荡时钟	内部高速振荡器振荡 • RSTOP = 0	可以停止 X1 振荡(MSTOP = 1).
外部主系统时钟			可以禁止外部主系统时钟输入(MSTOP = 1).
内部高速振荡时钟	XT1 时钟	XT1 振荡稳定 • OSCSELS = 1 • 经历了振荡稳定时间	通过停止内部高速振荡器可以降低操作电流(RSTOP = 1)
X1 时钟			可以停止 X1 振荡 (MSTOP = 1)
外部主系统时钟			可以禁止外部主系统时钟输入(MSTOP = 1)
XT1 时钟	内部高速振荡时钟	内部高速振荡器振荡且选择内部高速振荡时钟作为主系统时钟 • RSTOP = 0, MCS = 0	可停止 XT1 振荡 (OSCSELS = 0).
	X1 时钟	X1 振荡稳定且选择高速系统时钟作为主系统时钟 • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • 经历了振荡稳定时间 • MCS = 1	
	外部主系统时钟	允许来自 EXCLK 引脚的外部时钟输入并选择高速系统时钟作为主系统时钟 • MSTOP = 0, OSCSEL = 1, EXCLK = 1 • MCS = 1	

### 5.6.8 CPU 时钟和主系统时钟切换所需的时间

通过设置处理器时钟控制器(PCC)的第 0 位到第 2 位(PCC0 到 PCC2)以及第 4 位(CSS)，可以切换 CPU 时钟(在主系统时钟和副系统时钟之间)，同时可以改变主系统时钟的分频比。

修改 PCC 后，实际的切换操作不会立即执行；使用切换前的时钟继续操作几个时钟(见 表 5-7)。

可以通过 PCC 寄存器的第 5 位(CLS)来确定 CPU 使用主系统时钟还是副系统时钟。

表 5-7. CPU 时钟切换所需时间与主系统时钟周期分频因子

切换前设置值				切换后设置值																							
CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0																				
				0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	×	×	×
0	0	0	0	/				16 个时钟				2f <sub>XP</sub> /f <sub>SUB</sub> 个时钟															
	0	0	1					8 个时钟				f <sub>XP</sub> /f <sub>SUB</sub> 个时钟															
	0	1	0					4 个时钟				f <sub>XP</sub> /2f <sub>SUB</sub> 个时钟															
	0	1	1					2 个时钟				f <sub>XP</sub> /4f <sub>SUB</sub> 个时钟															
	1	0	0					1 个时钟				f <sub>XP</sub> /8f <sub>SUB</sub> 个时钟															
1	×	×	×	2 个时钟																							

**注意事项** 主系统时钟周期分频因子(PCC0 到 PCC2)的选择，以及从主系统时钟到副系统时钟的切换(CSS 从 0 变到 1)不能同时设置。

但对于主系统时钟周期分频因子(PCC0 到 PCC2)的选择，以及副系统时钟到主系统时钟的切换(CSS 从 1 变到 0)可以同时设置。

#### 备注

1. 表 5-7 列出的时钟数是切换前的 CPU 时钟数。
2. 将 CPU 时钟从主系统时钟切换到副系统时钟，可通过舍入一个时钟并舍弃小数部分来计算时钟数量(如下所示)。

**示例：** 将 CPU 时钟从 f<sub>XP</sub>/2 切换到 f<sub>SUB</sub>/2 (@ 振荡频率 f<sub>SUB</sub> = 32.768 kHz, f<sub>XP</sub> = 10 MHz)  
 $f_{XP}/f_{SUB} = 10000/32.768 \cong 305.1 \rightarrow 306$  个时钟

通过设置主时钟模式寄存器(MCM)的第 0 位(MCM0)，可以切换主系统时钟(在内部高速振荡时钟和高速系统时钟之间)。

修改 MCM0 后，实际的切换操作不会立即执行；使用切换前的时钟继续操作几个时钟(见 表 5-8)。

可以通过 MCM 寄存器的第 1 位(MCS)来确定 CPU 使用内部高速振荡时钟还是高速系统时钟。

表 5-8. 主系统时钟切换所需的最长时间

切换前设置值	切换后设置值	
MCM0	MCM0	
	0	1
0		$1 + 2f_{RH}/f_{XH}$ 时钟
1	$1 + 2f_{XH}/f_{RH}$ 时钟	

**注意事项** 当内部高速振荡时钟切换到高速系统时钟时，MCM 的第 2 位(XSEL)必须预先被设置为 1。XSEL 位在复位释放后只能被修改一次。

**备注**

1. 表 5-8 所列时钟数是切换之前的主系统时钟数。
2. 通过舍去小数部分，计算表 5-8 的时钟数。

**示例：** 将主系统时钟从内部高速振荡时钟切换到高速系统时钟(@振荡频率  $f_{RH} = 8 \text{ MHz}$ ,  $f_{XH} = 10 \text{ MHz}$ )

$$1 + 2f_{RH}/f_{XH} = 1 + 2 \times 8/10 = 1 + 2 \times 0.8 = 1 + 1.6 = 2.6 \rightarrow 2 \text{ 个时钟}$$

### 5.6.9 时钟振荡停止前的状况

下表列出了停止时钟振荡的寄存器标志位设置(禁止外部时钟输入)和时钟振荡停止前的状况。

表 5-9. 时钟振荡停止前的状况及标志位设置

时钟	时钟振荡停止前的状况 (禁止外部时钟输入)	SFR 寄存器的标志位 设置
内部高速振荡时钟	MCS = 1 或 CLS = 1 (CPU 不使用内部高速振荡时钟)	RSTOP = 1
X1 时钟 外部主系统时钟	MCS = 0 或 CLS = 1 (CPU 不使用高速系统时钟)	MSTOP = 1
XT1 时钟	CLS = 0 (CPU 不使用副系统时钟)	OSCSELS = 0

## 5.6.10 外部硬件与源时钟

下表列出了 78K0/LC3 中包含的外部硬件与源时钟。

表 5-10. 外部硬件与源时钟

源时钟		外部硬件时钟( $f_{PRS}$ )	副系统时钟( $f_{SUB}$ )	内部低速振荡时钟( $f_{RL}$ )	TM50 输出	TM52 输出	TMH1 输出	来自外部硬件引脚的外部时钟
外部硬件								
16 位定时器/事件计数器	00	Y	Y	N	N	Y	N	Y (TI000 引脚) <sup>注1</sup>
8 位定时器/事件计数器	50	Y	N	N	N	N	N	N
	51	Y	N	N	N	N	Y	N
	52	Y	N	N	N	N	N	Y (TI52 引脚) <sup>注1</sup>
8 位定时器	H0	Y	N	N	Y	N	N	N
	H1	Y	N	Y	N	N	N	N
	H2	Y	N	N	N	N	N	N
实时计数器		Y	Y	N	N	N	N	N
看门狗定时器		N	N	Y	N	N	N	N
蜂鸣器输出		Y	N	N	N	N	N	N
逐次逼近型 A/D 转换器 <sup>注2</sup>		Y	N	N	N	N	N	N
串行接口	UART0	Y	N	N	Y	N	N	N
	UART6	Y	N	N	Y	N	N	N
LCD 控制器/驱动器		Y	Y	Y	N	N	N	N
Manchester 编码发生器		Y	N	N	N	N	N	N

**注** 1.当 CPU 使用副系统时钟且内部高速振荡时钟已经停止时，不要启动依赖由外部硬件引脚输入外部时钟的这些功能。

2.仅对  $\mu$ PD78F041x。

**备注** Y: 可选, N: 不可选

### 6.1 16 位定时器/事件计数器 00 的功能

16 位定时器/事件计数器 00 具有以下功能。

**(1) 间隔定时器**

16 位定时器/事件计数器 00 以预置的时间间隔产生中断请求。

**(2) 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。

**(3) 外部事件计数器**

16 位定时器/事件计数器 00 可以测量外部输入信号的脉冲数。

**(4) 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。

**(5) PPG 输出**

16 位定时器/事件计数器 00 可以输出矩形波，矩形波的频率和输出宽度可以自由设置。

**(6) 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。

**(7) 24 位外部事件计数器**

通过结合 16 位定时器 00 和 8 位定时器/事件计数器 52，并使用 8 位定时器/事件计数器 52 的外部事件计数器功能，16 位定时器/事件计数器 00 可作为外部 24 位事件计数器操作。

当作为外部 24 位事件计数器，由 8 位定时器 H2 输出控制外部事件输入沿。

## 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 包括以下硬件。

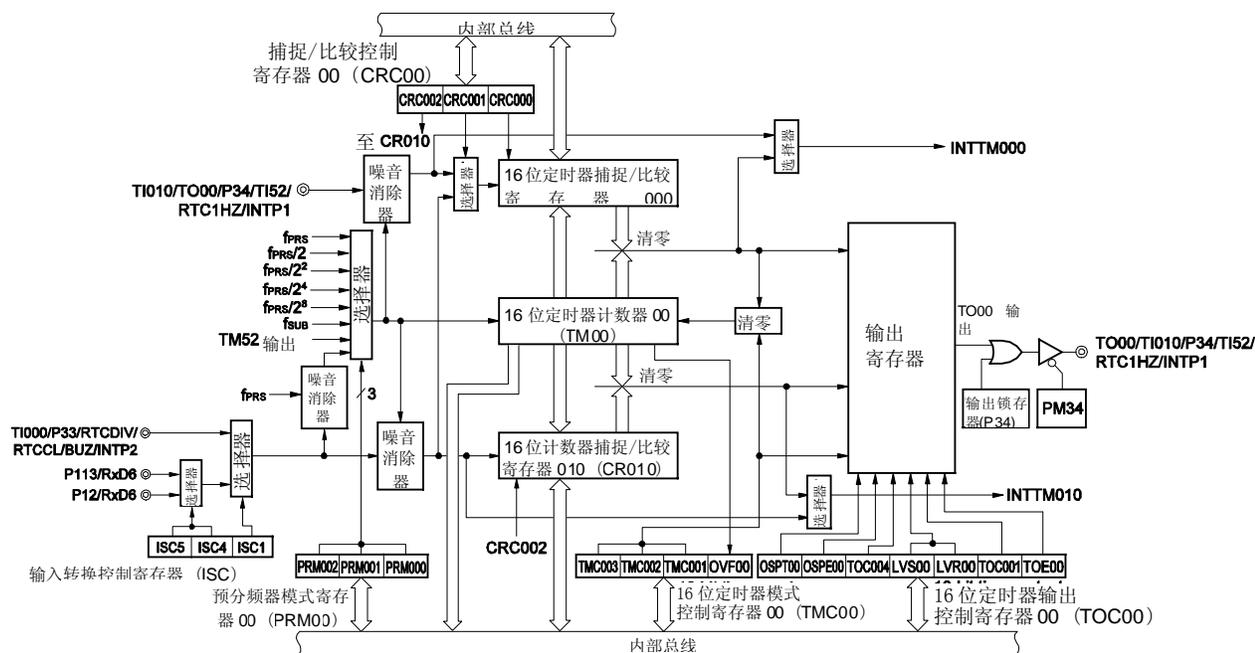
表 6-1. 16 位定时器/事件计数器 00 的配置

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 引脚
定时器输出	TO00 引脚, 输出控制器
控制寄存器	16 位定时器模式控制寄存器 00 (TMC00) 16 位定时器捕捉/比较控制寄存器 00 (CRC00) 16 位定时器输出控制寄存器 00 (TOC00) 预分频器模式寄存器 00 (PRM00) 输入转换控制寄存器 (ISC) 端口模式寄存器 3 (PM3) 端口寄存器 3 (P3)

**备注** 当使用 16 位定时器/事件计数器 00 作为外部 24 位事件计数器时, 要使用 8 位定时器/事件计数器 52 (TM52) 和 8 位定时器计数器 H2 (TMH2)。详细内容, 参见 6.4.9 外部 24 位事件计数器操作。

图 6-1 显示了 16 位定时器/事件计数器的框图。

图 6-1. 16 位定时器/事件计数器 00 的框图



**注意事项 1.** P34 引脚不能同时用于 TI010 有效沿和定时器输出(TO00), 选择其一功能。

- 注意事项 2.** 如果 16 位定时器模式控制寄存器 00 (TMC00) 的第 3 和 2 位(TMC003 和 TMC002)被清零和捕捉触发输入冲突, 则捕捉到的数据不确定。
- 3.** 要从捕捉模式变为比较模式, 首先清零 TMC003 和 TMC002 位, 然后再改变设置。捕捉值被保存在 CR000, 除非设备复位。如果将模式变为比较模式, 确保设置一个比较值。

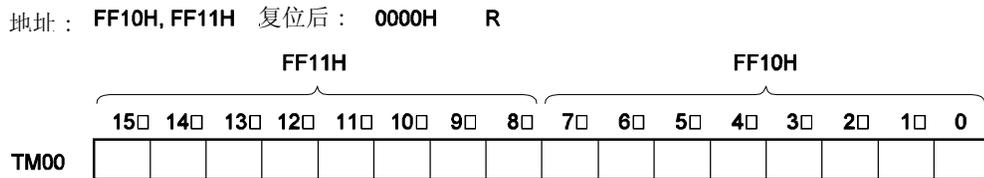
### (1) 16 位定时器计数器 00 (TM00)

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次数同步增加。

如果在操作期间读取计数值, 暂时停止计数时钟输入, 则此时的计数值被读取。

图 6-2. 16 位定时器计数器 00 (TM00)的格式



当 16 位定时器模式控制寄存器 00 (TMC00) 的第 3 位和第 2 位不是 00 时, 可通过读取 TM00 来获取计数值。如果在 TMC003 和 TMC002 = 00 时读取 TM00, 则获取的计数值为 0000H。

如出现以下情况, 计数值复位为 0000H。

- 复位信号的产生
- TMC003 和 TMC002 被清零
- 在 TI000 引脚输入有效沿进入清零和启动模式时
- 在 TM00 和 CR000 相等时进入清零和启动模式时
- 在单脉冲输出模式下如果 OSPT00 被设置为 1 或 TI000 引脚输入有效沿时

**注意事项** 即使读取了 TM00, 也不能通过 CR010 捕捉到该值。

### (2) 16 位定时器 捕捉/比较 寄存器 000 (CR000), 16 位定时器 捕捉/比较 寄存器 010 (CR010)

CR000 与 CR010 是 16 位寄存器, 具有捕捉或比较功能(通过 CRC00 进行选择)。

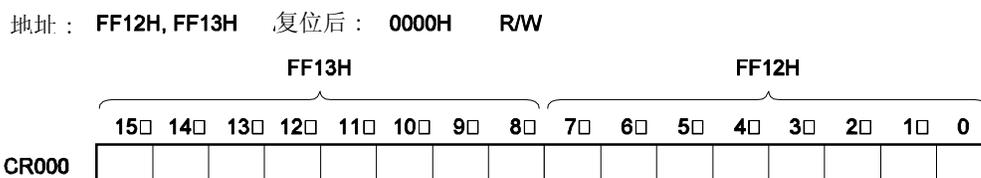
当定时器停止操作时(TMC003 与 TMC002 = 00), 修改 CR000 的值。

如果已按指定方式对 CR010 进行设置, 则在操作期间可以修改 CR010。如需了解详细信息, 可参见 6.5.1 TM00 操作期间重写 CR010。

可按 16 位读写这些寄存器。

复位信号的产生可将这些寄存器设置为 0000H。

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

**(i) 当 CR000 用作比较 寄存器**

CR000 的设置值与 16 位定时器计数器 00 (TM00) 的计数值频繁地相比较，如果相等将产生一个中断请求 (INTTM000)。在 CR000 被重写前一直保持这个值。

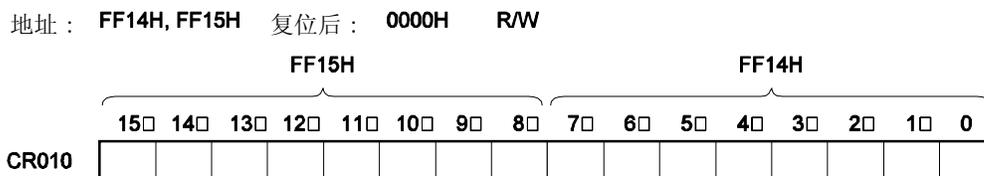
**注意事项** CR000 设置为比较模式时不能执行捕捉操作，即使有捕捉触发输入。

**(ii) 当 CR000 用作捕捉 寄存器**

当输入捕捉触发时，TM00 的计数值被捕捉到 CR000。

可以使用 CRC00 或 PRM00 选择 TI000 引脚的反向脉冲沿或 TI010 引脚的有效沿作为捕捉触发。

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

**(i) 当 CR010 用作比较寄存器**

CR010 的设置值与 TM00 的计数值频繁地相比较，如果相等将产生一个中断请求 (INTTM010)。

**注意事项** CR010 设置为比较模式时不能执行捕捉操作，即使有捕捉触发输入。

**(ii) 当 CR010 用作捕捉 寄存器**

当输入捕捉触发时，TM00 的计数值被捕捉到 CR010。

可以选择 TI000 引脚的有效沿作为捕捉触发。TI000 有效沿由 PRM00 来设置。

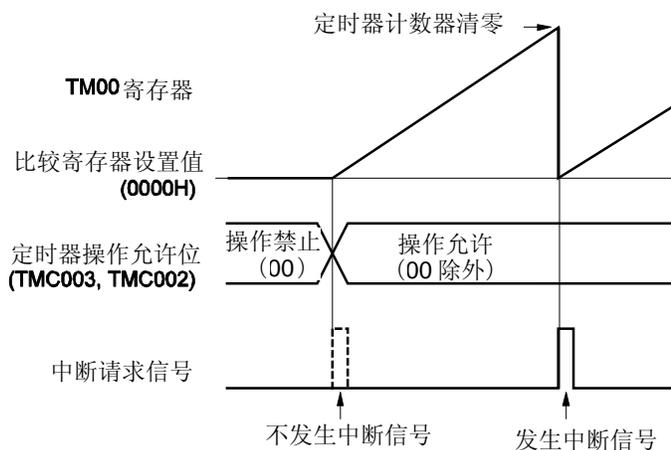
## (iii) CR000 或 CR010 用作比较寄存器时的设置范围

当 CR000 或 CR010 用作比较寄存器时, 按下面所示进行设置。

操作	CR000 寄存器设置范围	CR010 寄存器设置范围
作为间隔定时器操作	0000H < N ≤ FFFFH	0000H <sup>#</sup> ≤ M ≤ FFFFH
作为方波输出操作		通常, 此设置不使用。屏蔽相等中断信号 (INTTM010)。
作为外部事件计数器操作		
由 TI000 引脚有效沿输入进入 清零 & 启动模式	0000H <sup>#</sup> ≤ N ≤ FFFFH	0000H <sup>#</sup> ≤ M ≤ FFFFH
作为自由运行定时器操作		
作为 PPG 输出操作	M < N ≤ FFFFH	0000H <sup>#</sup> ≤ M < N
作为单脉冲输出操作	0000H <sup>#</sup> ≤ N ≤ FFFFH (N ≠ M)	0000H <sup>#</sup> ≤ M ≤ FFFFH (M ≠ N)

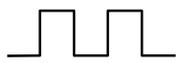
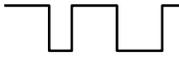
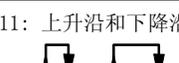
注 当 0000H 被设置时, 定时器不操作并且定时器输出不改变后, 一个相等中断立即产生, 第一个相等时序如下所示。当定时器计数器 (TM00 寄存器) 从 0000H 变为 0001H 时产生相等中断。

- 当定时器计数器由于溢出清零时
- 当定时器计数器由于 TI000 引脚有效沿 (由 TI000 引脚有效沿输入进入清零 & 启动模式) 清零时
- 当定时器计数器由于比较相等 (清零 & 启动模式下 TM00 与 CR000 相等 (CR000 = 非 0000H, CR010 = 0000H)) 清零时



- 备注
1. N: CR000 寄存器设置值, M: CR010 寄存器设置值
  2. TMC003 和 TMC002 的详细说明, 参见 6.3 (1) 16 位定时器 模式控制寄存器 00 (TMC00)。

表 6-2. CR000 和 CR010 的捕捉操作

外部输入信号 捕捉操作	TI000 引脚输入 		TI010 引脚输入 	
CR000 的捕捉操作	CRC001 = 1 TI000 引脚输入 (反相) 	ES001 和 ES000 的设置 值 要捕捉的沿	CRC001 位= 0 TI010 引脚输入 	ES101 和 ES100 的设置 值 要捕捉的沿
		01: 上升沿 		01: 上升沿 
		00: 下降沿 		00: 下降沿 
	11: 上升沿和下降沿 (不能捕捉)		11: 上升沿和下降沿	
中断信号	即使值被捕捉, INTTM000 信号也不发 生。		中断信号	每次值被捕捉时, 产生 INTTM000 信号。
CR010 的捕捉操作	TI000 引脚输入 <sup>注</sup> 	ES001 和 ES000 的设置 值 要捕捉的沿		
		01: 上升沿 		
		00: 下降沿 		
	11: 上升沿和下降沿 			
中断信号	每次值被捕捉时, 产生 INTTM010 信号。			

注 CR010 的捕捉操作不受 CRC001 位设置的影响。

**注意事项** 如果要使用输入到 TI000 引脚的反向脉冲沿来捕捉 TM00 的计数值, 并存入 CR000, 则在捕捉了计数值后, 不产生中断请求信号(INTTM000)。如果在操作期间检测到 TI010 的有效沿, 则不执行捕捉操作, 但会产生 INTTM000 信号作为外部中断信号。当不使用外部中断时, 可以屏蔽 INTTM000 信号。

**备注** CRC001: 参见 6.3 (2) 捕捉/比较控制寄存器 00 (CRC00)。  
ES101, ES100, ES001, ES000: 参见 6.3 (4) 预分频模式寄存器 00 (PRM00)。

### 6.3 寄存器控制 16 位定时器/事件计数器 00

用于 16 位定时器/事件计数器 00 的寄存器如下所示。

- 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)
- 输入切换控制寄存器 (ISC)
- 端口 模式 寄存器 3 (PM3)
- 端口寄存器 3 (P3)

#### (1) 16 位 定时器 模式控制寄存器 00 (TMC00)

TMC00 是一个 8 位寄存器，用于设置 16 位定时器/事件计数器 00 的操作模式、TM00 清零模式和输出时序，及溢出检测。

操作期间(当 TMC003 与 TMC002 不等于 00)禁止重写 TMC00。当 TMC003 与 TMC002 清零(00)(停止操作)并且 OVF00 也清零(0)时，可以修改它的值。

可以由 1 位或 8 位存储器操作指令设置。

复位信号的产生将 TMC00 清零(00H)。

**注意事项** 当设置 TMC002 与 TMC003 为 00（操作停止模式）以外的值时，16 位定时器/事件计数器 00（TM00）开始计数。若要停止操作，则设置 TMC002 和 TMC003 为 00。

图 6-5. 16 位定时器 模式控制寄存器 00 (TMC00) 的格式

地址: FFBAH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	<0>
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

TMC003	TMC002	16 位定时器/事件计数器 00 操作允许
0	0	禁止 16 位定时器/事件计数器 00 操作。停止供应操作时钟。清零 16 位定时器计数器 00 (TM00)。
0	1	自由运行定时器 模式
1	0	通过 TI000 引脚有效沿输入*进入清零 & 启动模式
1	1	TM00 和 CR000 相等时进入清零 & 启动模式

TMC001	反转定时器输出(TO00)的条件
0	<ul style="list-style-type: none"> <li>• TM00 与 CR000 相等或 TM00 与 CR010 相等</li> </ul>
1	<ul style="list-style-type: none"> <li>• TM00 与 CR000 相等或 TM00 与 CR010 相等</li> <li>• TI000 引脚有效沿的触发输入</li> </ul>

OVF00	TM00 溢出标志
清零 (0)	将 OVF00 清零 (0) 或 TMC003 与 TMC002 = 00
设置 (1)	发生溢出

在所有操作模式下(自由运行定时器模式、清零&启动模式(通过 TI000 引脚有效沿输入进入)和清零&启动模式(在 TM00 与 CR000 相等时进入))当 TM00 的值由 FFFFH 变为 0000H 时, OVF00 标志位将被设置为 1。也可以将 1 写入 OVF00 进行设置。

注 通过预分频器模式寄存器 00 (PRM00) 的第 5 位和第 4 位(ES001, ES000)设置 TI000 引脚的有效沿。

**(2) 捕捉/比较控制寄存器 00 (CRC00)**

CRC00 用于控制 CR000 和 CR010 的操作。

操作期间(TMC003 与 TMC002 不等于 00 时)禁止修改 CRC00。

可以由 1 位或 8 位存储器操作指令设置 CRC00。

复位信号的产生将 CRC00 清零(00H)。

图 6-6. 捕捉/比较控制寄存器 00 (CRC00) 的格式

地址: FFBC H 复位后: 00H R/W

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

CRC002	CR010 操作模式的选择
0	用作比较寄存器
1	用作捕捉寄存器

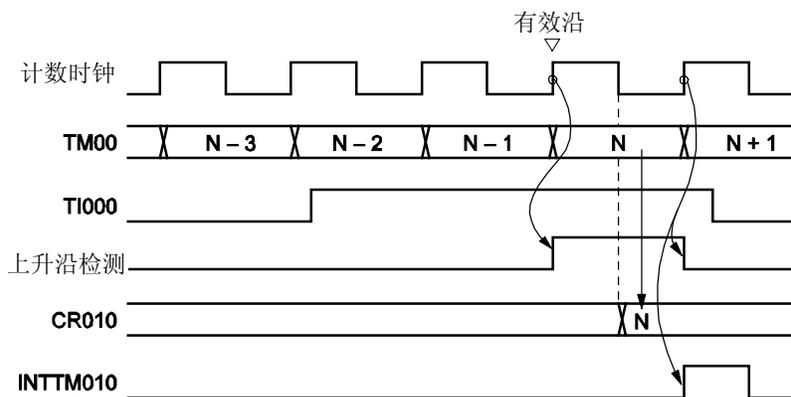
CRC001	CR000 捕捉触发选择
0	在 TI010 引脚的有效沿捕捉
1	在 TI000 引脚的有效沿反向 <sup>注</sup> 时捕捉
由 PRM00 设置 TI010 与 TI000 引脚的有效沿 如果在 CRC001=1 时将 ES001 和 ES000 设置为 11(两种脉冲沿), 则不能检测 TI000 引脚的有效沿	

CRC000	CR000 操作模式选择
0	用作比较寄存器
1	用作捕捉寄存器
如果 TMC003 与 TMC002 被设置为 11 (清零&启动模式(在 TM00 与 CR000 相等时进入)), 则必须将 CRC000 设置为 0	

**注** 当检测到 TI000 的有效沿时, 不执行捕捉操作, 但产生 INTTM000 信号作为外部中断信号。

**注意事项** 为确保捕捉操作正确执行, 捕捉触发需要的脉冲应大于两个计数时钟的时钟周期, 该计数时钟用预分频器模式寄存器 00 (PRM00)选择。

图 6-7. CR010 捕捉操作示例 (指定上升沿有效)

**(3) 16 位 定时器输出控制寄存器 00 (TOC00)**

TOC00 是一个 8 位寄存器，用于控制 TO00 引脚输出。

仅当 OSPT00 操作时才能重写 TOC00(当 TMC003 与 TMC002 不等于 00 时)。操作期间禁止重写其它位。

但可以在定时器操作期间对 TOC00 进行重写，作为重写 CR010 的一种方法(参见 6.5.1 TM00 操作期间 CR010 的重写)。

可以由 1 位或 8 位存储器操作指令设置 TOC00。

复位信号的产生将 TOC00 清零(00H)。

**注意事项** 必须按以下步骤设置 TOC00。

- <1> 设置 TOC004 与 TOC001 为 1。
- <2> 仅设置 TOE00=1。
- <3> 设置 LVS00 或 LVR00 为 1。

图 6-8. 16 位 定时器输出控制寄存器 00 (TOC00) 的格式

地址: FFB DH 复位后: 00H R/W

符号	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
OSPT00	通过软件进行单脉冲输出触发							
0	-							
1	单脉冲输出							
该位的读取值总是为“0”。在所有模式下(单脉冲输出模式除外), 不要将该位设置为 1。如果该位为 1, 则将 TM00 清零并启动。								
OSPE00	单脉冲输出操作控制							
0	连续脉冲输出模式							
1	单脉冲输出							
在自由运行定时器模式或清零&启动模式(通过 TI000 引脚有效沿输入进入)下, 单脉冲输出可以正确进行。在清零&启动模式(由 TM00 与 CR000 相等进入)下不能输出单脉冲。								
TOC004	CR010 与 TM00 相等时 TO00 引脚输出控制							
0	禁止反转操作							
1	允许反转操作							
即使 TOC004 = 0, 也产生中断信号 (INTTM010)。								
LVS00	LVR00	TO00 引脚输出状态设置						
0	0	无变化						
0	1	TO00 引脚输出的初值为低电平 (将 TO00 引脚输出清零(0))						
1	0	TO00 引脚输出的初值为高电平 (将 TO00 引脚输出设置为 1)						
1	1	禁止设置						
<ul style="list-style-type: none"> <li>• LVS00 与 LVR00 可以用来设置 TO00 引脚输出电平的初值。如果不必设初值, 则将保持 LVS00 与 LVR00 为 00。</li> <li>• 当 TOE00 = 1 时, 必须设置 LVS00 与 LVR00。禁止同时将 LVS00、LVR00 和 TOE00 设置为 1。</li> <li>• LVS00 与 LVR00 是触发位。通过将这两位设置为 1, 可以对 TO00 引脚输出电平的初值进行设置。即使将这两位清零(0), TO00 引脚的输出也不会受到影响。</li> <li>• LVS00 与 LVR00 的读取值总是为 0。</li> <li>• 如需了解 LVS00 与 LVR00 的设置, 参见 6.5.2 LVS00 与 LVR00 的设置。</li> <li>• TO00/TI010/P34/TI52/RTC1HZ/INTP1 引脚实际的输出由 PM34 和 P34 决定, TO00 输出除外。</li> </ul>								
TOC001	CR000 与 TM00 相等时 TO00 引脚输出控制							
0	禁止反转操作							
1	允许反转操作							
即使 TOC001 = 0, 也产生中断信号 (INTTM000)								
TOE00	TO00 引脚输出控制							
0	禁止输出 (TO00 引脚输出恒为低电平)							
1	允许输出							

**(4) 预分频模式 寄存器 00 (PRM00)**

PRM00 用于设置 TM00 的计数时钟以及 TI000 与 TI010 引脚输入的有效沿。

操作期间(当 TMC003 与 TMC002 不等于 00 时)禁止重写 PRM00。

可由 1 位或 8 位存储器操作指令设置 PRM00。

复位信号的产生将 PRM00 清零(00H)。

**注意事项** 1. 当 PRM001 与 PRM000 为 11 时(指定 TI000 引脚的有效沿作为计数时钟)，不要进行以下设置。

- 清零 & 启动模式(通过 TI000 引脚有效沿进入)
  - 设置 TI000 引脚作为捕捉触发
2. 如果在 TI000 或 TI010 引脚为高电平并且指定 TI000 或 TI010 引脚的有效沿为上升沿或兼有两种脉冲沿时，允许 16 位定时器/事件计数器 00 操作，TI000 或 TI010 引脚的高电平将被检测为上升沿。当 TI000 或 TI010 引脚被上拉时要注意。然而，一旦定时器操作已经被停止并且之后再次被允许时，则不检测上升沿。
3. 对于 P01 引脚，TI010 的有效沿和定时器输出(TO00)不能同时使用。可以任选其一。

图 6-9. 预分频模式 寄存器 00 (PRM00) 的格式

地址: FFBBH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PRM00	ES101	ES100	ES001	ES000	0	PRM002	PRM001	PRM000

ES101	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	兼有上升沿和下降沿

ES001	ES000	TI000 引脚有效沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	兼有上升沿和下降沿

PRM002	PRM001	PRM000	计数时钟选择 <sup>※1</sup>			
			f <sub>PRS</sub> = 2 MHz	f <sub>PRS</sub> = 5 MHz	f <sub>PRS</sub> = 10 MHz	
0	0	0	f <sub>PRS</sub> <sup>※2</sup>	2 MHz	5 MHz	10 MHz
0	0	1	f <sub>PRS</sub> /2	1 MHz	2.5 MHz	5 MHz
0	1	0	f <sub>PRS</sub> /2 <sup>2</sup>	500 kHz	1.25 MHz	2.5 MHz
0	1	1	f <sub>PRS</sub> /2 <sup>4</sup>	1.25 MHz	2.5 MHz	625 kHz
1	0	0	f <sub>PRS</sub> /2 <sup>8</sup>	7.81 kHz	19.53 kHz	39.06 kHz
1	0	1	f <sub>SUB</sub>	32.768 kHz		
1	1	0	TI000 有效沿 <sup>※3</sup>			
1	1	1	TM52 输出			

- 注
- 如果外围硬件时钟(f<sub>PRS</sub>) 操作在高速系统时钟 (f<sub>XH</sub>) (XSEL = 1), f<sub>PRS</sub> 的工作频率根据供电电压的不同而不同。
    - V<sub>DD</sub> = 2.7~5.5 V: f<sub>PRS</sub> ≤ 10 MHz
    - V<sub>DD</sub> = 1.8~2.7 V: f<sub>PRS</sub> ≤ 5 MHz
  - 如果外围硬件时钟(f<sub>PRS</sub>) 操作在内部高速振荡时钟(f<sub>RH</sub>) (XSEL = 0), 当 1.8 V ≤ V<sub>DD</sub> < 2.7 V 时, PRM002 = PRM001 = PRM000 = 0 (计数时钟: f<sub>PRS</sub>)的设置被禁止。
  - 外部时钟脉冲宽度要求大于内部时钟(f<sub>PRS</sub>)的两个周期宽度。

**注意事项** 在脉冲宽度测量期间不要选择 TI000 的有效沿作为计数时钟。

- 备注
- 通过设置 PRM002, PRM001, PRM000 = 1, 1, 1, 8 位 定时器/事件计数器 52 (TM52)输出可选择作为 TM00 计数时钟。根据 TM52 计数时钟和比较 寄存器的设置值, 可设置任意频率作为 16 位 定时器 (TM00) 计数时钟。
  - f<sub>PRS</sub>: 外围硬件时钟频率  
f<sub>SUB</sub>: 副系统时钟频率

**(5) 输入切换控制寄存器 (ISC)**

通过将 ISC1 设置为 0，TI000 的输入源作为 P33/TI000 引脚的输入信号。

可由 1 位或 8 位存储器操作指令设置 ISC。

复位信号的产生将 ISC 清零为 00H。

图 6-10. 输入切换控制寄存器 (ISC) 的格式

地址: FF4FH 复位后: 00H R/W

符号

	7	6	5	4	3	2	1	0
ISC	0	0	ICS5	ICS4	ICS3	ICS2	ICS1	ICS0

ICS5	ICS4	TxD6, RxD6 输入源 选择
0	0	TxD6:P112, RxD6: P113
1	0	TxD6:P13, RxD6: P12
其他值		禁止设置

ICS3	RxD6/P113 输入允许/禁止
0	RxD6/P113 输入禁止
1	RxD6/P113 输入允许

ICS2	TI52 输入源 控制
0	不控制 TI52 输入 (P34)
1	控制 TI52 输入 (P34) <sup>注1</sup>

ICS1	TI000 输入源选择
0	TI000 (P33)
1	RxD6 (P12 或 P113) <sup>注2</sup>

ICS0	INTP0 输入源选择
0	INTP0 (P120)
1	RxD6 (P12 或 P113) <sup>注2</sup>

- 注
- 1.由 TOH2 输出 信号控制 TI52 输入。
  - 2.TI000 和 INTP0 输入由 ICS5 和 ICS4 选定。

**(6) 端口模式 寄存器 3 (PM3)**

该寄存器按位设置端口 3 为输入/输出模式。

如果使用 P34/TI52/TI010/TO00/RTC1HZ/INTP1 引脚用于 定时器输出, 则需要将 PM34 以及 P34 的输出锁存器清零。

如果使用 P33/TI000/RTCDIV/RTCCL/BUZ/INTP2 和 P34/TI52/TI010/TO00/RTC1HZ/INTP1 引脚用于定时器输入, 则需要将 PM33 和 PM34 设置为 1。此时 P33 和 P34 的输出锁存器可以为 0 也可以为 1。

可由 1 位或 8 位存储器操作指令设置 PM3。

复位信号的产生将 PM3 设置为 FFH。

图 6-11. 端口模式 寄存器 3 (PM3) 的格式

地址 : FF23H 复位后 : FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34	PM33	PM32	PM31	1

PM3n	P3n 引脚 I/O 模式选择(n=1~4)
0	输出模式(输出缓冲器打开)
1	输入模式(输出缓冲器关闭)

## 6.4 16 位定时器/事件计数器 00 的操作

### 6.4.1 间隔定时器操作

如果 16 位定时器模式控制寄存器 (TMC00) 的第 3 位和第 2 位 (TMC003 和 TMC002) 为 11 (清零&启动模式(在 TM00 与 CR000 相等时进入)), 则启动计数操作与计数时钟同步。

当之后的 TM00 值与 CR000 值相等时, 将 TM00 清零 (0000H) 并产生一个相等中断信号 (INTTM000)。这个 INTTM000 信号允许 TM00 作为间隔定时器操作。

- 备注
1. I/O 引脚设置的详细说明, 参见 6.3 (6) 端口模式 寄存器 3 (PM3)。
  2. 要了解如何允许 INTTM000 中断, 参见 第十七章 中断功能。

图 6-12. 间隔定时器操作的框图

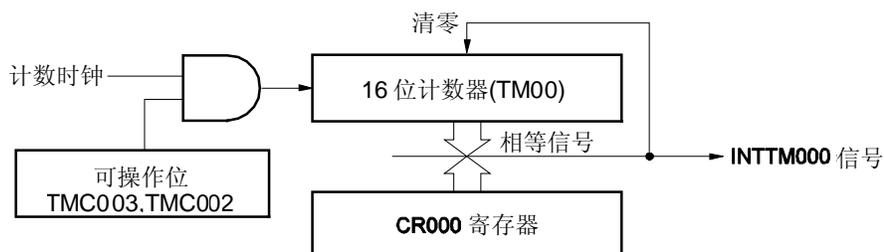


图 6-13. 间隔定时器操作的基本时序示例

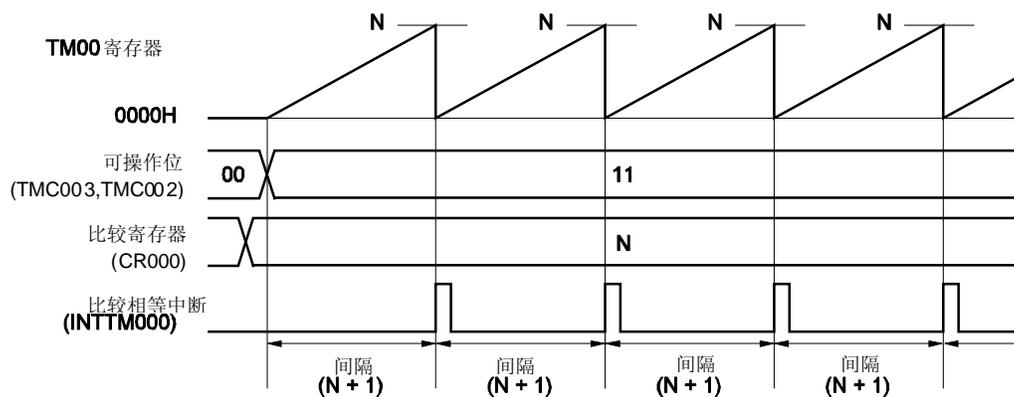
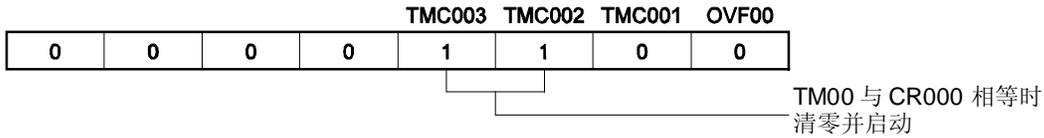
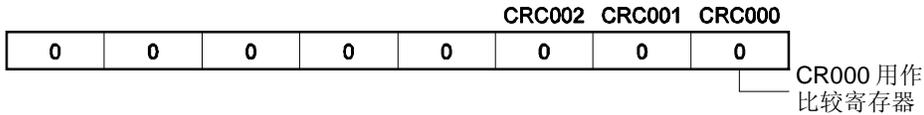


图 6-14. 间隔定时器操作的寄存器设置示例

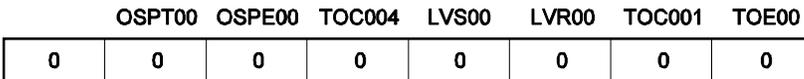
## (a) 16 位 定时器 模式控制寄存器 00 (TMC00)



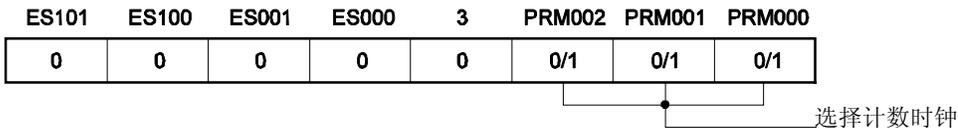
## (b) 捕捉/比较控制寄存器 00 (CRC00)



## (c) 16 位 定时器输出控制寄存器 00 (TOC00)



## (d) 预分频 模式 寄存器 00 (PRM00)



## (e) 16 位定时器计数器 00 (TM00)

通过读取 TM00，获得计数值。

## (f) 16 位 捕捉/比较 寄存器 000 (CR000)

如果  $CR000=M$ ，则间隔时间表示如下。

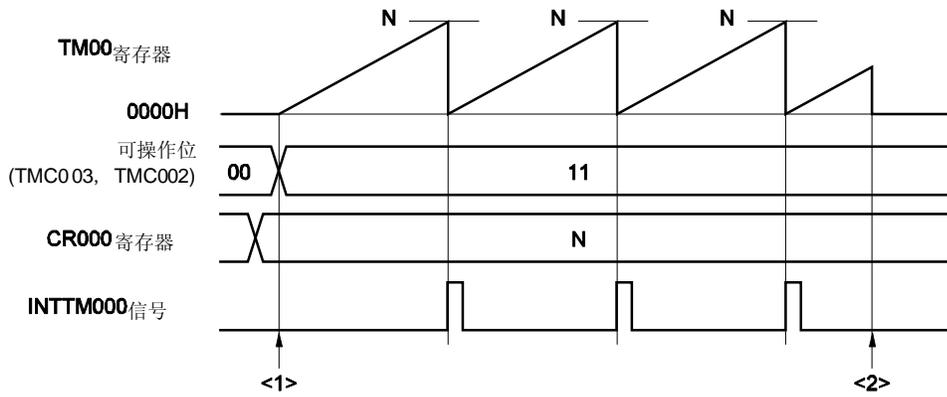
- 间隔时间 =  $(M + 1) \times$  计数时钟周期

禁止将 CR000 设置为 0000H。

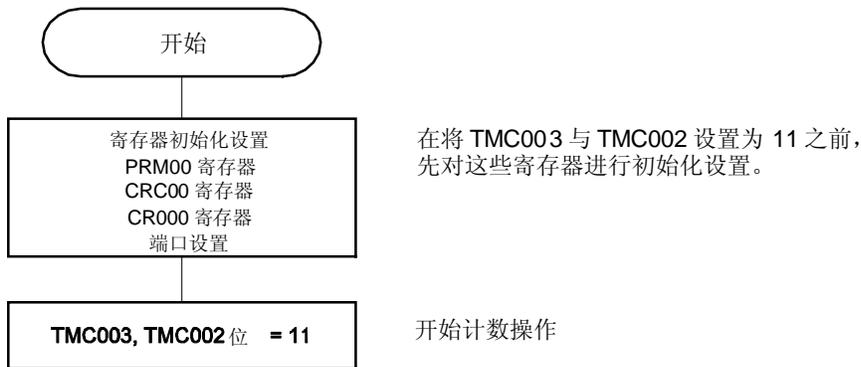
## (g) 16 位 捕捉/比较 寄存器 010 (CR010)

通常 CR010 不用作间隔定时器。但当 CR010 的值与 TM00 的值相等时，会产生比较相等中断(INTTM010)。因此通过使用中断屏蔽标志(TMMK010)屏蔽中断请求。

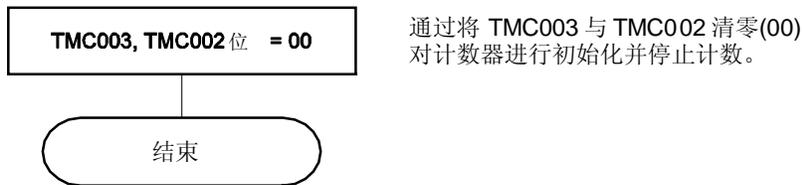
图 6-15. 间隔定时器软件处理示例



<1> 计数操作启动流程



<2> 计数操作停止流程



### 6.4.2 方波输出操作

当 16 位定时器/事件计数器 00 用作间隔定时器时(参见 6.4.1)，通过将 16 位定时器输出控制寄存器 00 (TOC00) 设置为 03H，可以从 TO00 引脚输出一个方波。

当 TMC003 与 TMC002 被设置为 11 时(计数清零&启动模式(在 TM00 与 CR000 相等时进入))，启动计数操作与计数时钟同步。

当之后 TM00 的值与 CR000 的值相等时，将 TM00 清零(0000H)、产生中断信号(INTTM000)并反转 TO00 引脚的输出。反转的 TO00 引脚输出以恒定的间隔允许 TO00 输出一个方波。

- 备注
1. 如需了解 I/O 引脚的设置，可参见 6.3 (6) 端口 模式 寄存器 3 (PM3)。
  2. 要了解如何允许 INTTM000 信号中断，参见 第十七章 中断功能。

图 6-16. 方波输出操作框图

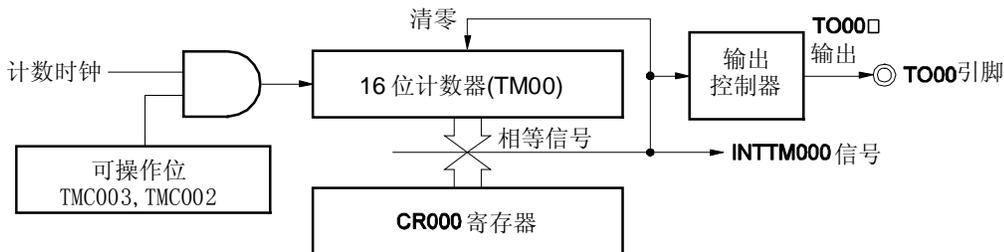


图 6-17. 方波输出操作的基本时序示例

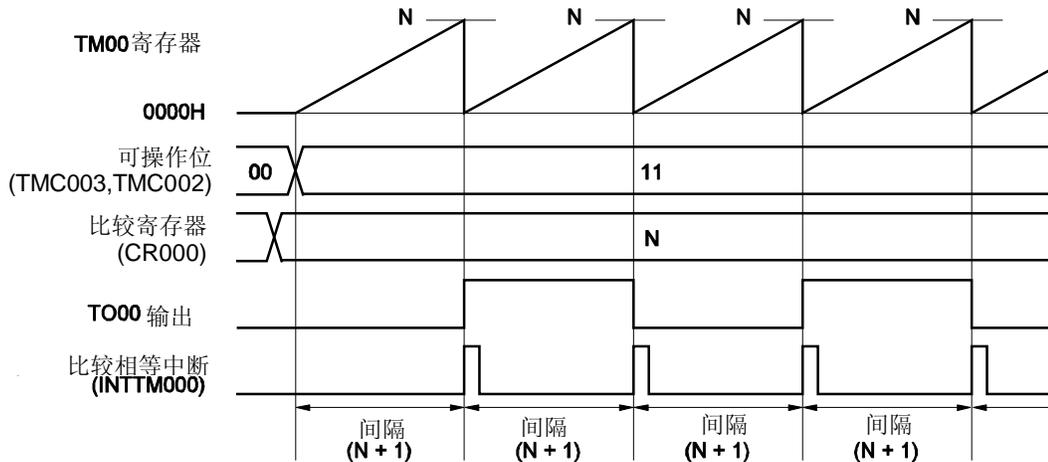
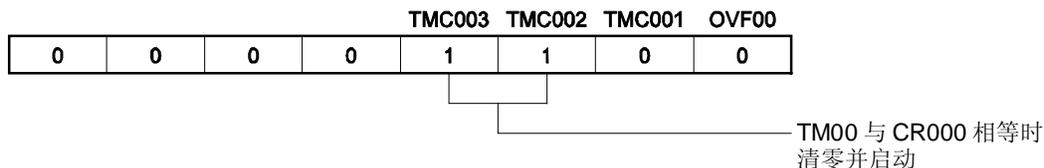
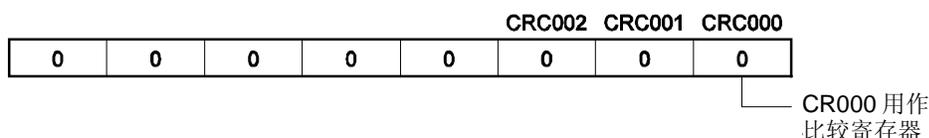


图 6-18. 方波输出操作的寄存器设置示例

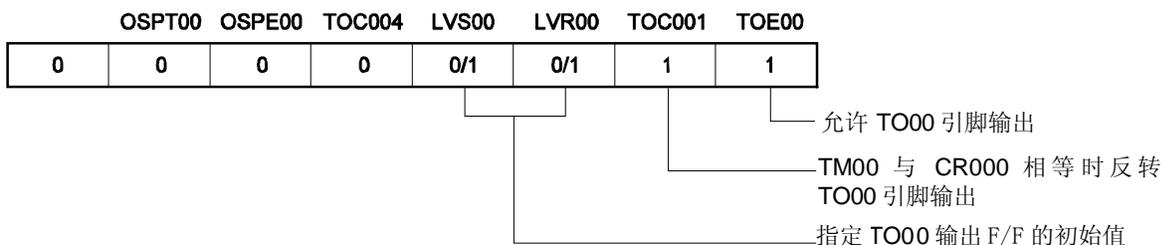
## (a) 16 位 定时器 模式控制寄存器 00 (TMC00)



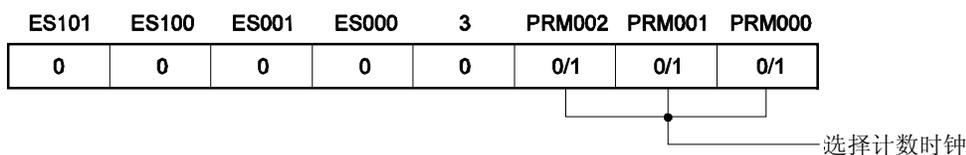
## (b) 捕捉/比较控制寄存器 00 (CRC00)



## (c) 16 位 定时器输出控制寄存器 00 (TOC00)



## (d) 预分频 模式 寄存器 00 (PRM00)



## (e) 16 位定时器计数器 00 (TM00)

通过读取 TM00，获得计数值。

## (f) 16 位 捕捉/比较 寄存器 000 (CR000)

如果 CR000=M，则间隔时间表示如下。

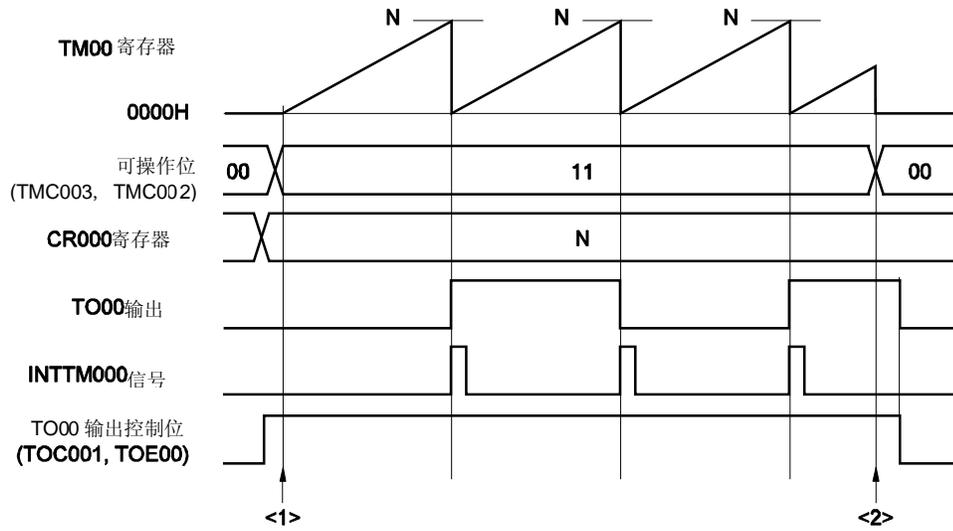
- 方波频率 =  $1 / [2 \times (M + 1) \times \text{计数时钟周期}]$

禁止将 CR000 设置为 0000H。

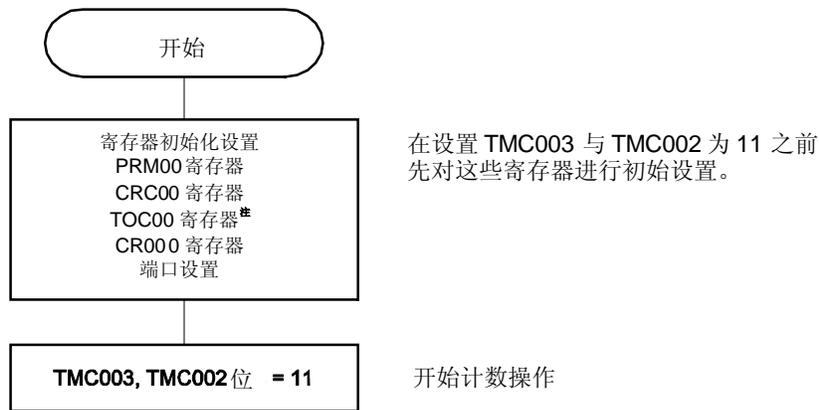
## (g) 16 位 捕捉/比较 寄存器 010 (CR010)

通常 CR010 不用作方波输出。但当 CR010 的值与 TM00 的值相等时，会产生比较相等中断(INTTM010)。因此，通过使用中断屏蔽标志(TMMK010)屏蔽中断请求。

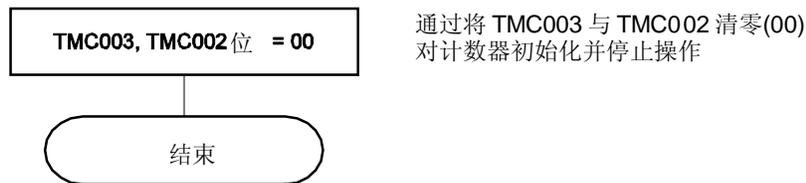
图 6-19. 方波输出功能软件处理示例



<1>计数操作启动流程



<2>计数操作停止流程



注 设置 TOC00 时必须特别小心。如需了解详细信息，参见 6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

### 6.4.3 外部事件计数器操作

当预分频器模式寄存器 00 (PRM00)的第 1 位和第 0 位(PRM001 和 PRM000)被设置为 11(使用 TI000 引脚的有效沿计数)以及 16 位定时器模式控制寄存器 00 (TMC00)的第 3 位和第 2 位(TMC003 和 TMC002)被设置为 11 时,开始对外部事件输入的有效沿进行计数,并产生用于指示 TM00 和 CR000 之间相等的相等中断信号(INTTM000)。

如果要输入外部事件,可以使用 TI000 引脚。因此定时器/事件计数器不能用作清零&启动模式(通过 TI000 引脚有效沿(当 TMC003 与 TMC002 = 10 时)进入)下的外部事件计数器。

INTTM000 信号按下列时序产生。

- INTTM000 信号产生时序 (第 2 次或之后产生)  
= 外部事件输入的有效沿检测次数 × (CR000 的设置值 + 1)

但在定时器/事件计数器开始操作后,将按以下时序立即产生第 1 次相等中断。

- INTTM000 信号产生时序 (仅在第 1 次产生)  
= 外部事件输入的有效沿检测次数 × (CR000 的设置值 + 2)

如果要检测有效沿,则应在 fPRS 时钟周期内对输入到 TI000 引脚的信号进行采样。在一个周期内检测到两次脉冲沿时才认为检测到有效沿,这样可以消除一个窄脉冲宽度的噪音。

- 备注**
1. 如需了解 I/O 引脚的设置,参见 6.3 (6) 端口 模式 寄存器 3 (PM3)。
  2. 要了解如何允许 INTTM000 信号中断,参见第十七章 中断功能。

图 6-20. 外部事件计数器操作框图

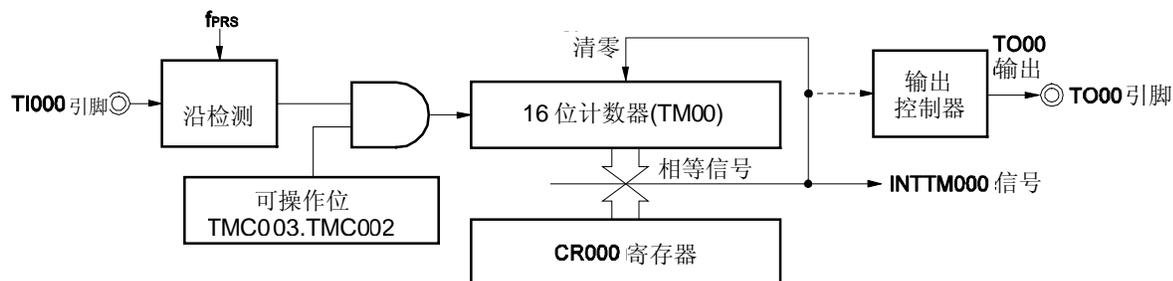
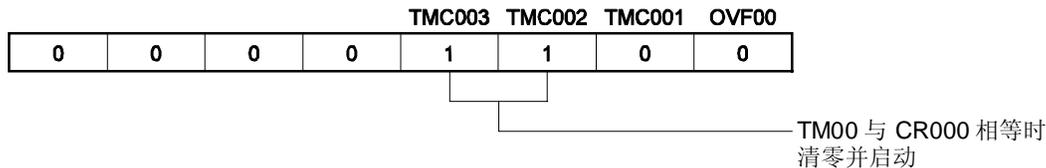
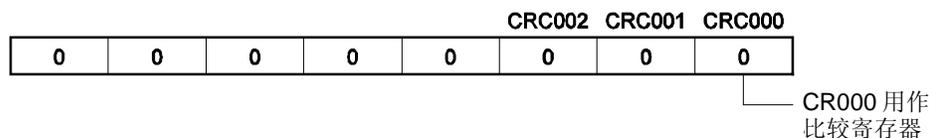


图 6-21. 外部事件计数器模式下寄存器设置示例(1/2)

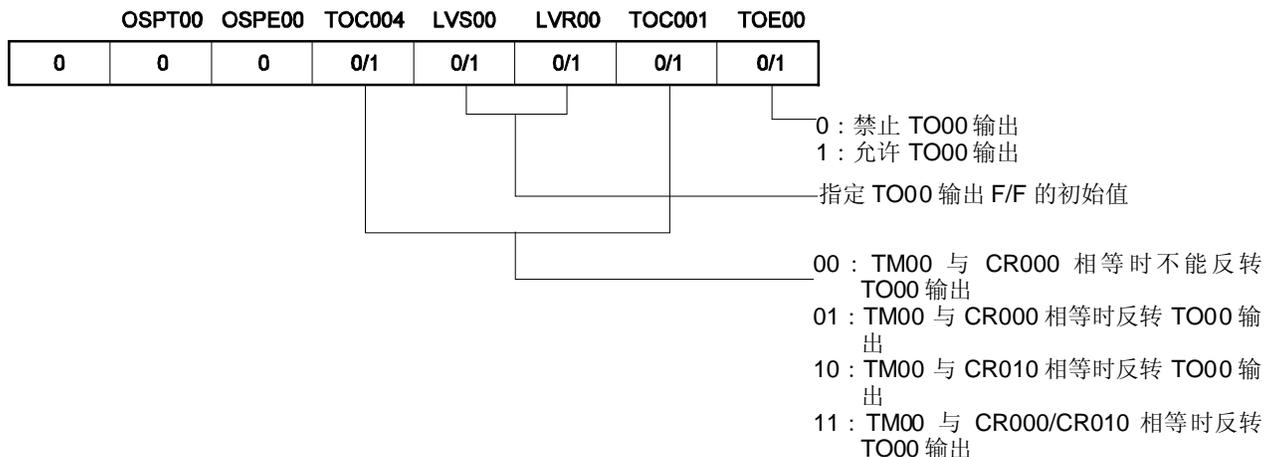
(a) 16 位 定时器 模式控制寄存器 00 (TMC00)



(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) 预分频 模式 寄存器 00 (PRM00)

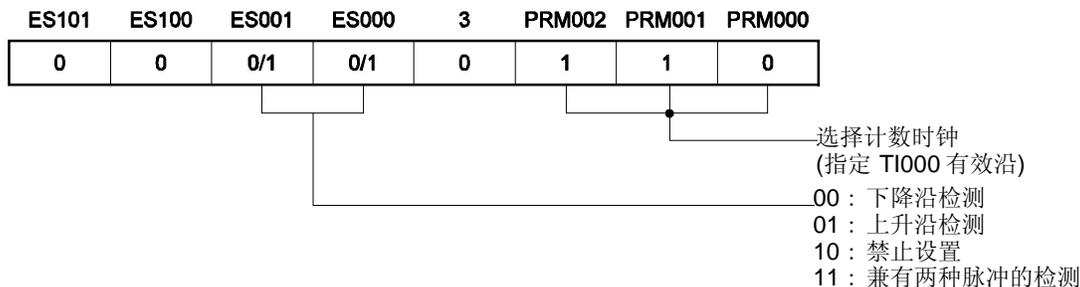


图 6-21. 外部事件计数器模式下寄存器设置示例(2/2)

**(e) 16 位定时器计数器 00 (TM00)**

通过读取 TM00，获得计数值。

**(f) 16 位 捕捉/比较 寄存器 000 (CR000)**

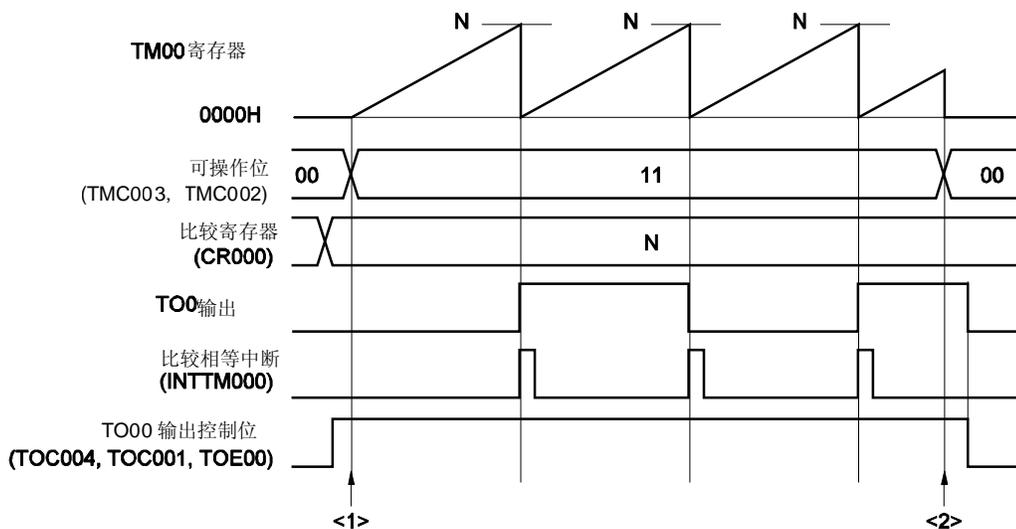
如果 CR000=M，当外部事件达到 (M+1) 时产生中断信号 (INTTM000)。

禁止将 CR000 设置为 0000H。

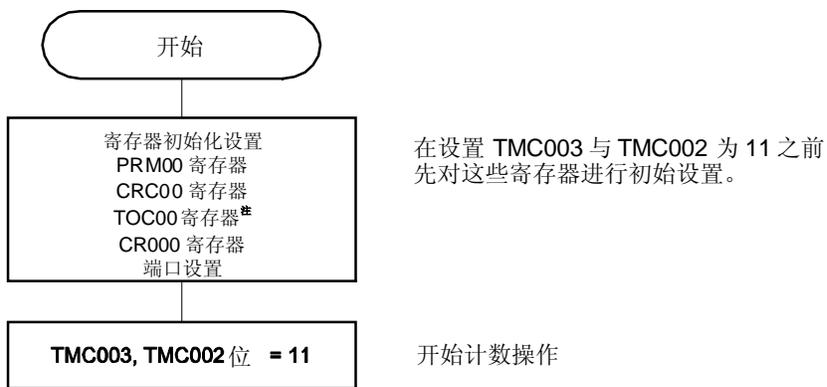
**(g) 16 位 捕捉/比较 寄存器 010 (CR010)**

通常 CR010 不用作外部事件计数器模式。但当 CR010 的值与 TM00 的值相等时，会产生比较相等中断 (INTTM010)。因此，通过使用中断屏蔽标志(TMMK010)屏蔽中断请求。

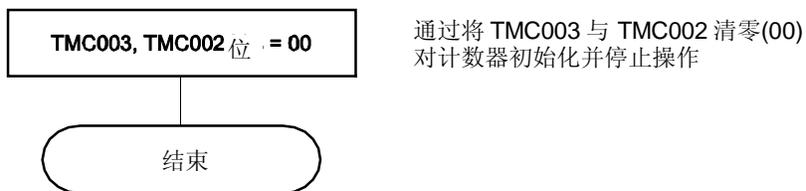
图 6-22. 外部事件计数器模式下的软件处理示例



<1> 计数操作启动流程



<2> 计数操作停止流程



注 设置 TOC00 时必须特别小心。如需了解详细信息，参见 6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。

#### 6.4.4 通过 TI000 引脚有效沿输入进入的清零&启动模式下的操作

当 16 位定时器模式控制寄存器 00 (TMC00) 的第 3 位和第 2 位 (TMC003 和 TMC002) 被设置为 10 (清零&启动模式 (通过 TI000 引脚有效沿输入进入)) 并且向定时器/事件计数器提供计数时钟 (由 PRM00 设置) 时, TM00 开始计数。在计数操作期间, 如果检测到 TI000 引脚的有效沿, 则将 TM00 清零 (0000H) 并再次开始计数。如果没有检测到 TI000 引脚的有效沿, 则 TM00 溢出并继续计数。

可通过 TI000 引脚的有效沿对 TM00 清零。在操作开始后, 不会立即启动计数器。

CR000 与 CR010 可用作比较寄存器和捕捉寄存器。

##### (a) CR000 与 CR010 用作比较寄存器

当 TM00 的值与 CR000、CR010 的值相等时, 产生信号 INTTM000 与 INTTM010。

##### (b) CR000 与 CR010 用作捕捉寄存器

当有效沿输入到 TI010 引脚时 (或者当反向的有效脉冲沿输入到 TI000 引脚时), TM00 的计数值被捕捉到 CR000, 同时产生 INTTM000 信号。

当有效沿输入到 TI000 引脚时, TM00 的计数值被捕捉到 CR010, 同时产生 INTTM010 信号。一旦捕捉了计数值, 计数器即被清零 (0000H)。

**注意事项** 不要将 TI000 引脚的有效沿 (PRM002, PRM001, 和 PRM000= 110) 设置为计数时钟。当 PRM002, PRM001, 和 PRM000= 110 时, 将 TM00 清零。

- 备注**
1. 如需了解 I/O 引脚的设置, 参见 6.3 (6) 端口模式寄存器 3 (PM3)。
  2. 要了解如何允许 INTTM000 信号中断, 参见第十七章 中断功能。

#### (1) 清零&启动模式 (通过 TI000 引脚有效沿输入进入) 下的操作

(CR000: 比较寄存器, CR010: 比较寄存器)

图 6-23. 清零&启动模式 (通过 TI000 引脚有效沿输入进入) 的框图  
(CR000: 比较寄存器, CR010: 比较寄存器)

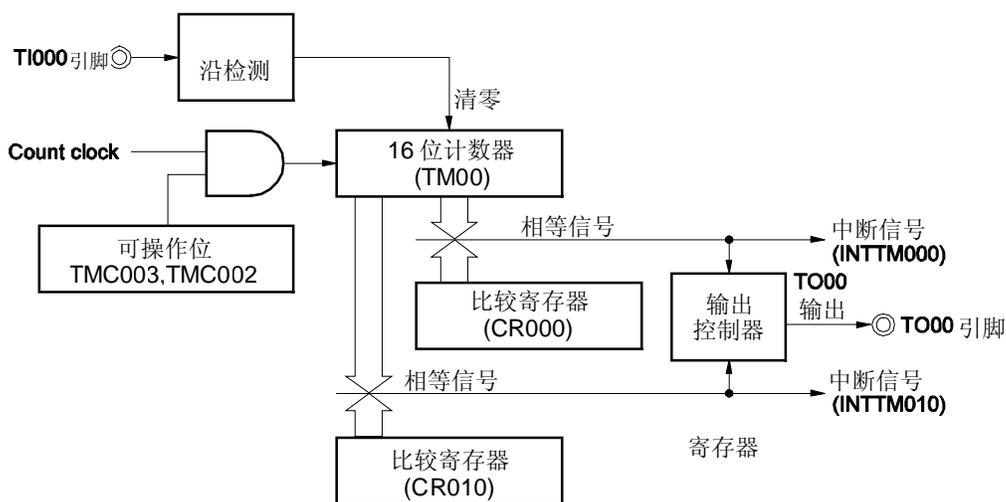
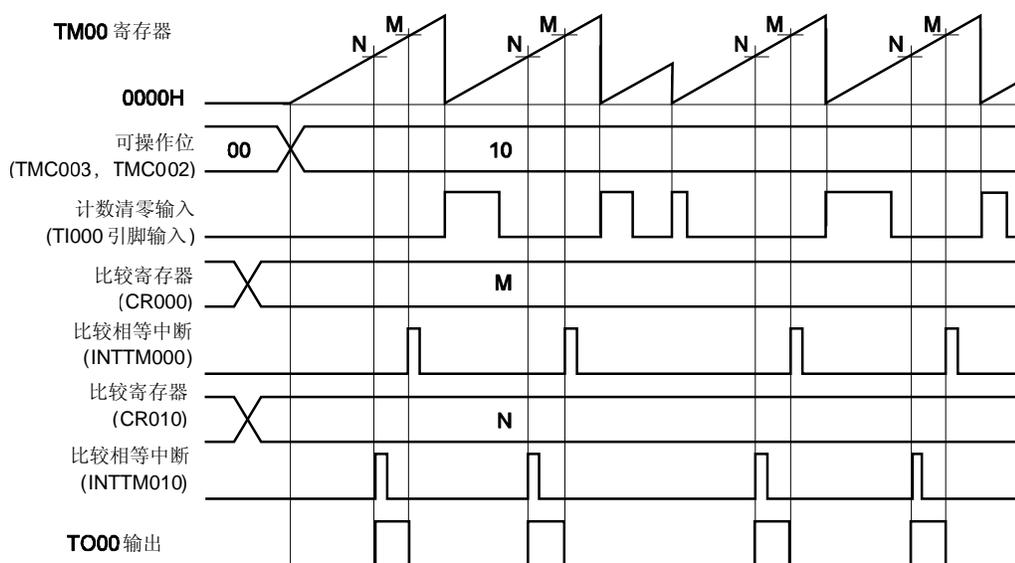
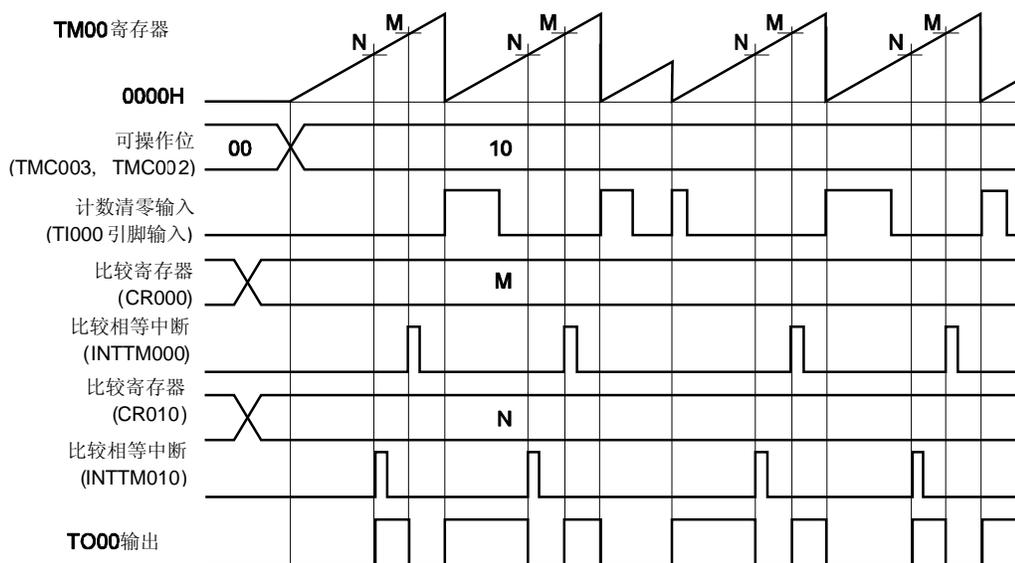


图 6-24. 清零&启动模式(通过 TI000 引脚有效沿输入进入)时序示例  
(CR000: 比较寄存器, CR010: 比较寄存器)

(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 00H, TMC00 = 08H



(b) TOC00 = 13H, PRM00 = 10H, CRC00, = 00H, TMC00 = 0AH



(a) 和 (b) 根据 16 位定时器模式控制寄存器 01 (TMC00) 第 1 位(TMC001) 的设置不同有如下变化。

- (a) 当 TM00 与比较寄存器相等时反转 TO00 引脚的输出电平。
- (b) 当 TM00 与比较寄存器相等或者检测到 TI000 有效沿时反转 TO00 引脚的输出电平。

(2) 清零&启动模式(通过 TI000 引脚有效沿输入进入)下的操作  
(CR000: 比较寄存器, CR010: 捕捉寄存器)

图 6-25. 清零&启动模式(通过 TI000 引脚有效沿输入进入)的框图  
(CR000: 比较寄存器, CR010: 捕捉寄存器)

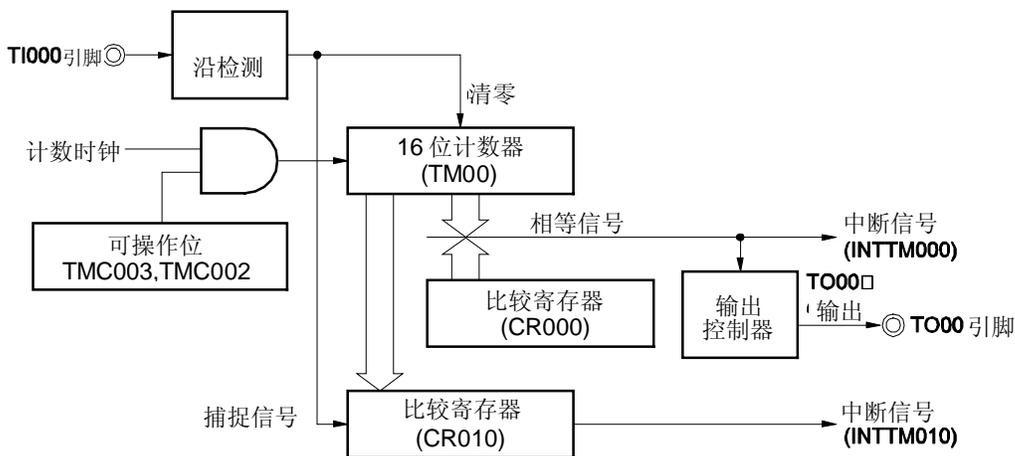
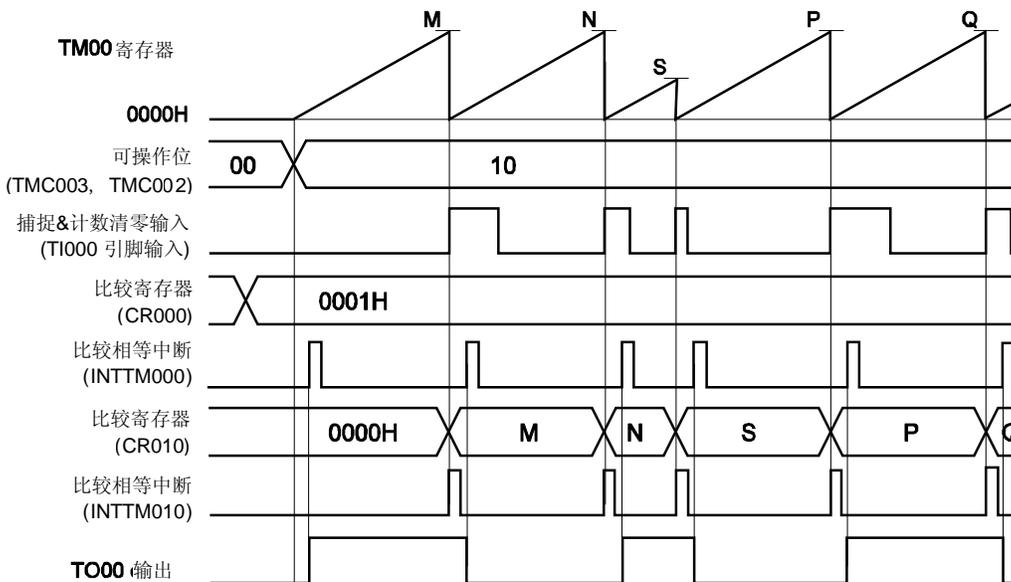


图 6-26. 清零&启动模式(通过 TI000 引脚有效沿输入进入) 时序示例  
(CR000: 比较寄存器, CR010: 捕捉寄存器) (1/2)

(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 04H, TMC00 = 08H, CR000 = 0001H

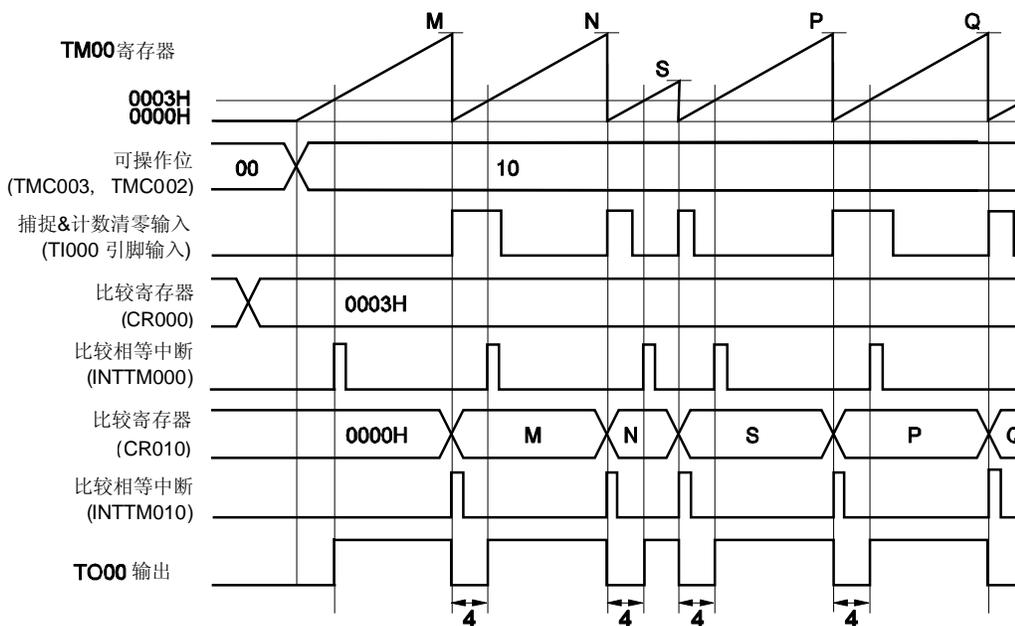


这是一个应用示例：当捕捉到计数值并清零时反转 TO00 引脚的输出电平。

当检测到 TI000 引脚的有效沿时，捕捉到的计数值存入 CR010 并将 TM00 清零(0000H)。当 TM00 的计数值 =0001H 时，产生比较相等中断信号(INTTM000)，并反转 TO00 引脚的输出电平。

图 6-26. 清零&启动模式(通过 TI000 引脚有效沿输入进入)时序示例  
(CR000: 比较 寄存器, CR010: 捕捉 寄存器) (2/2)

(b) TOC00 = 13H, PRM00 = 10H, CRC00, = 04H, TMC00 = 0AH, CR000 = 0003H



这是一个应用示例：当捕捉计数值并清零时，从 TO00 引脚输出 CR000 设置的宽度(本例是 4 个时钟)。

当检测到 TI000 引脚的有效沿时，捕捉到的计数值存入 CR010、产生捕捉中断信号(INTTM010)、将 TM00 清零(0000H)并反转 TO00 引脚的输出电平。当 TM00 的计数值=0003H 时(已经计数 4 个时钟)，产生比较相等中断信号(INTTM000)并反转 TO00 引脚的输出电平。

(3) 清零&启动模式(通过 TI000 引脚有效沿输入进入)的操作  
(CR000: 捕捉寄存器, CR010: 比较寄存器)

图 6-27. 清零&启动模式(通过 TI000 引脚有效沿输入进入)的框图  
(CR000: 捕捉寄存器, CR010: 比较寄存器)

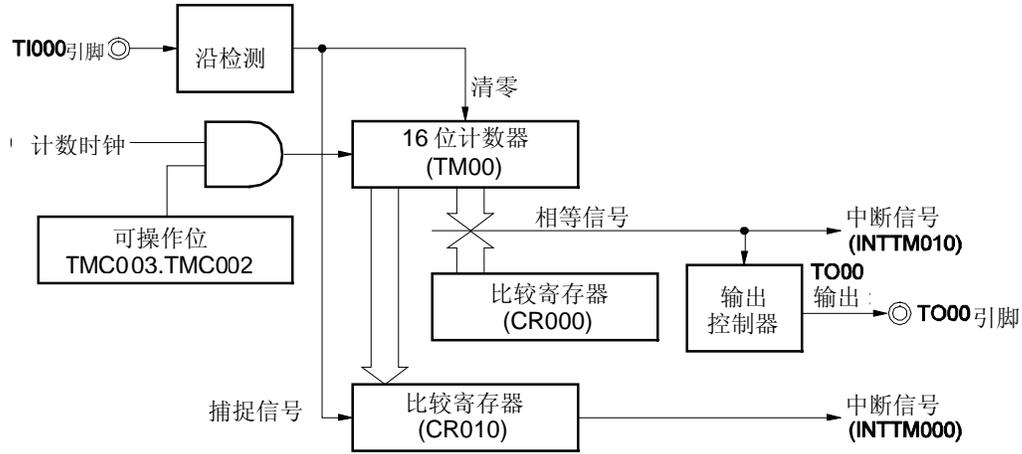
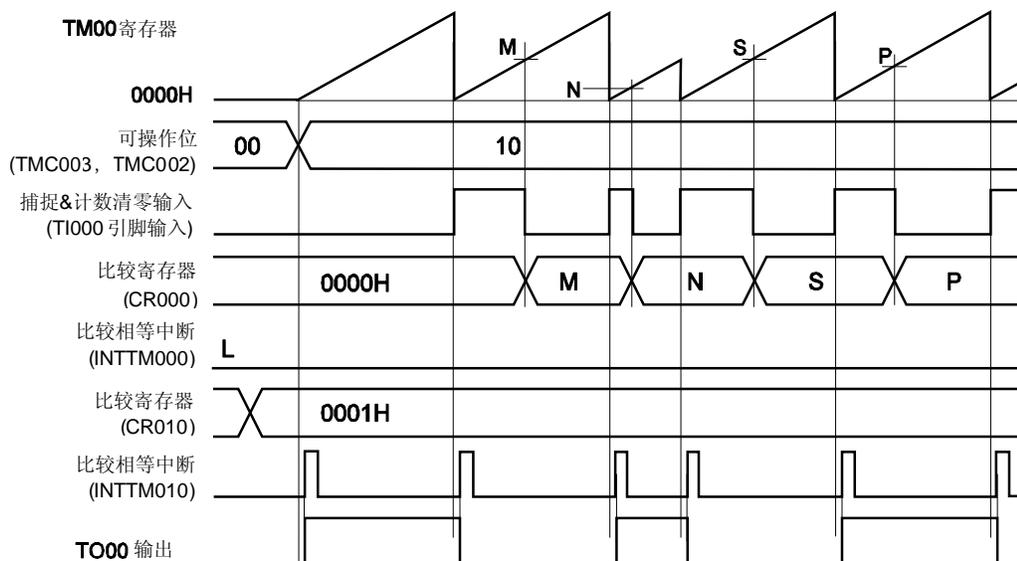


图 6-28. 清零&启动模式(通过 T1000 引脚有效沿输入进入) 时序示例  
(CR000: 捕捉寄存器, CR010: 比较寄存器) (1/2)

(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 03H, TMC00 = 08H, CR010 = 0001H



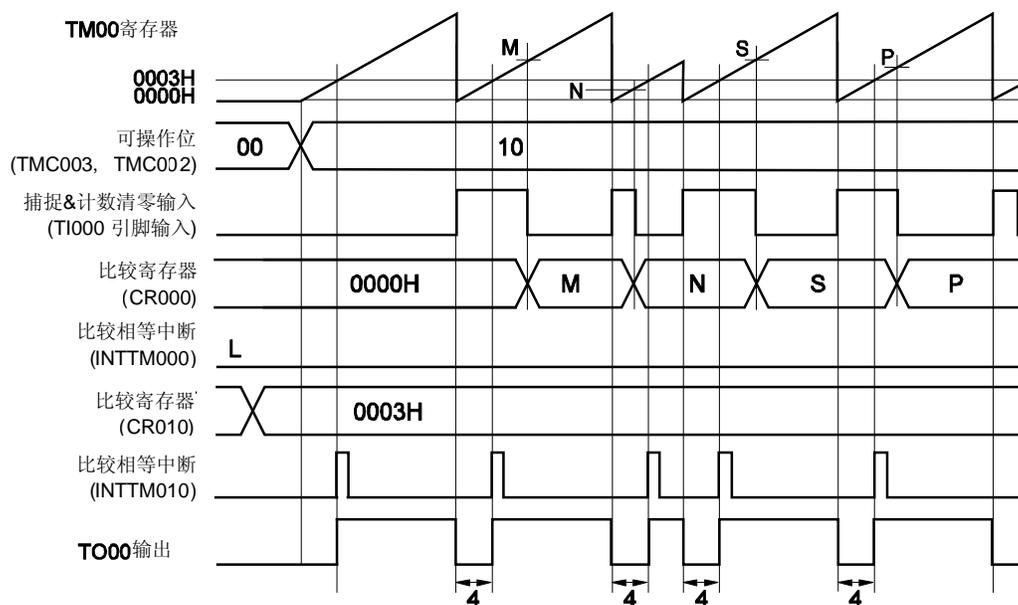
这是一个应用示例：当捕捉计数值并清零时，反转 TO00 引脚的输出电平。

在检测到 T1000 引脚的上升沿时将 TM00 清零，而在检测到 T1000 引脚的下降沿时，TM00 的计数值被捕捉到 CR000。

当捕捉/比较控制寄存器 00 (CRC00)的第 1 位(CRC001)=1 时，TM00 的计数值被捕捉到 CR000(在 T1000 引脚输入信号的反向沿捕捉)，但不产生捕捉中断信号(INTTM000)。然而当检测到 T1010 引脚的有效沿时产生 INTTM000 信号。如果不使用 INTTM000 信号，则将其屏蔽。

图 6-28. 清零&启动模式(通过 TI000 引脚有效沿输入进入) 时序示例  
(CR000: 捕捉 寄存器, CR010: 比较 寄存器) (2/2)

(b) TOC00 = 13H, PRM00 = 10H, CRC00, = 03H, TMC00 = 0AH, CR010 = 0003H



这是一个应用示例：当捕捉计数值并清零时，从 TO00 引脚输出 CR010 设置的宽度(本例是 4 个时钟)。

在检测到 TI000 引脚的上升沿时将 TM00 清零(0000H)，而在检测到 TI000 引脚的下降沿时，TM00 的计数值被捕捉到 CR000。当 TM00 被清零(0000H)后，TO00 引脚的输出电平被反转，因为 TI00 引脚检测到上升沿或者 TM00 的值与比较寄存器 (CR010) 的值相等。

当捕捉/比较控制寄存器 00 (CRC00)的第 1 位 (CRC001) =1 时，TM00 的计数值被捕捉到 CR000(在 TI000 引脚输入信号的反向沿捕捉)，但不产生捕捉中断信号 (INTTM000)。然而当检测到 TI010 引脚的有效沿时产生 INTTM000 中断信号。如果不使用 INTTM000 信号，则将其屏蔽。

(4) 清零&启动模式(通过 TI000 引脚有效沿输入进入)下的操作  
(CR000: 捕捉 寄存器, CR010: 捕捉 寄存器)

图 6-29. 清零&启动模式(通过 TI000 引脚有效沿输入进入)的框图  
(CR000: 捕捉 寄存器, CR010: 捕捉 寄存器)

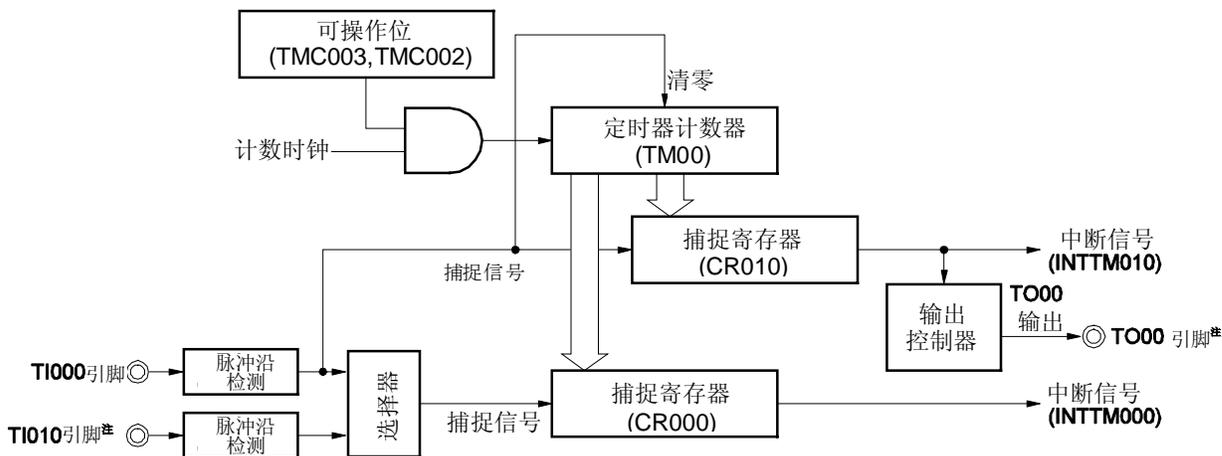
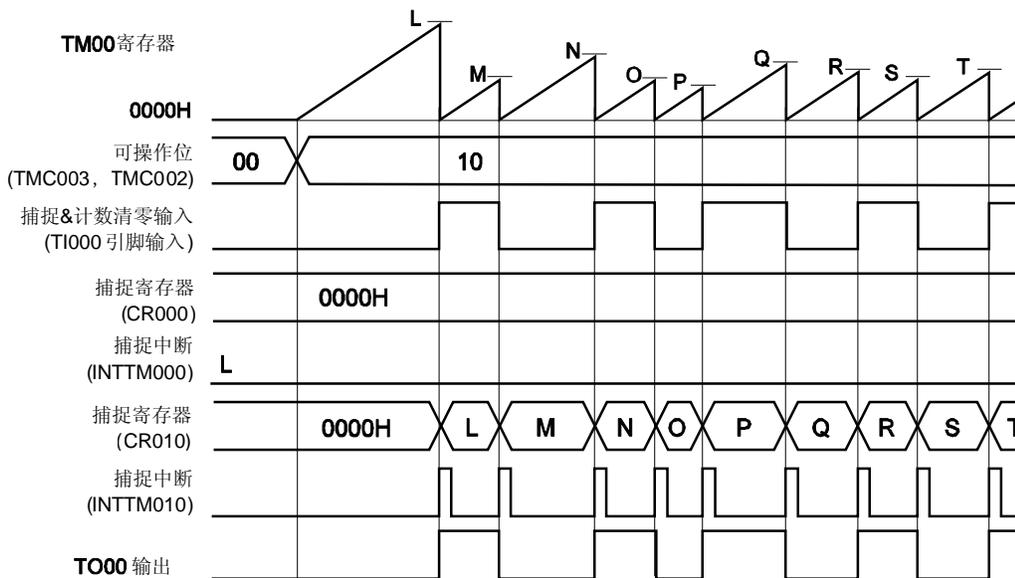


图 6-30. 清零&启动模式(通过 TI000 引脚有效沿输入进入)时序示例  
(CR000: 捕捉 寄存器, CR010: 捕捉 寄存器) (1/3)

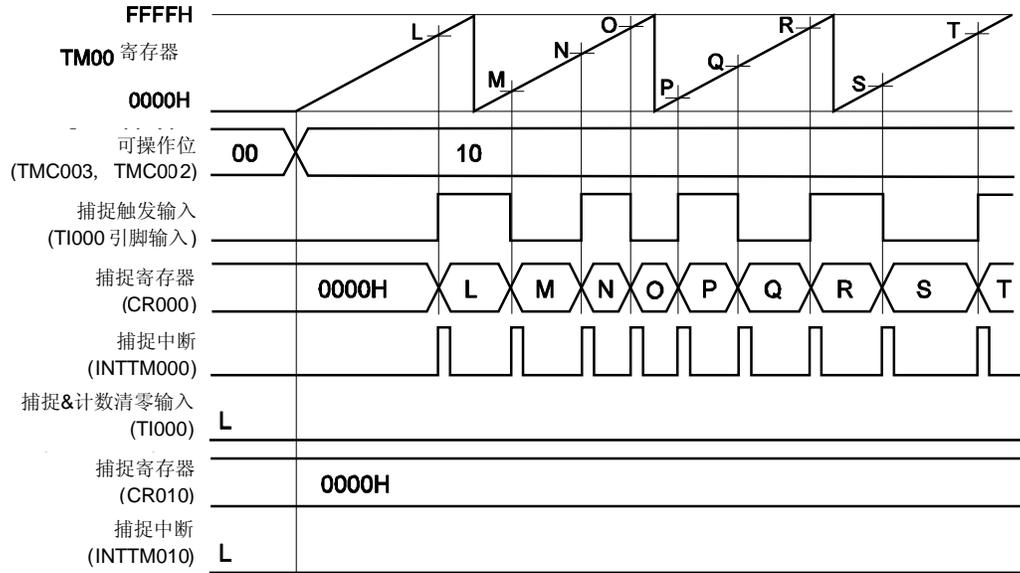
(a) TOC00 = 13H, PRM00 = 30H, CRC00 = 05H, TMC00 = 0AH



这是一个应用示例：当检测到 TI000 引脚的上升沿或下降沿时，捕捉到的计数值存入 CR010、将 TM00 清零并反转 TO00 引脚的输出  
在检测到 TI010 引脚的脉冲沿时产生中断信号(INTTM000)。如果不使用 INTTM000 信号，则将其屏蔽。

图 6-30. 清零&启动模式(通过 TI000 引脚有效沿输入进入)时序示例  
(CR000: 捕捉寄存器, CR010: 捕捉寄存器) (2/3)

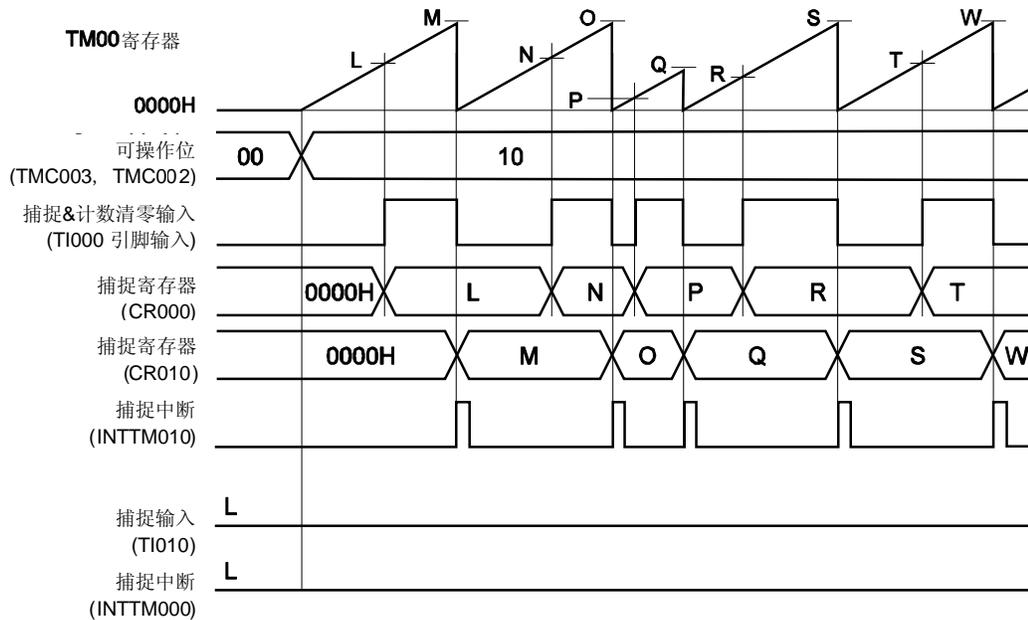
(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 0AH



这是一个时序示例：当脉冲沿不输入到 TI000 引脚时，在检测到 TI010 引脚的上升沿或下降沿时，捕捉到的计数值存入 CR000。

图 6-30. 清零&启动模式(通过 TI000 引脚有效沿输入进入)时序示例  
(CR000: 捕捉 寄存器, CR010: 捕捉 寄存器) (3/3)

(c) TOC00 = 13H, PRM00 = 00H, CRC00 = 07H, TMC00 = 0AH



这是一个应用示例：对输入到 TI000 引脚的信号脉冲宽度进行测量。

通过设置 CRC00，在 TI000 引脚下降沿(也可以是上升沿)的反向沿捕捉到的计数值存入 CR000，而在 TI000 引脚下降沿捕捉到的计数值存入 CR010。

可以通过下列表达式计算输入脉冲的高电平和低电平宽度。

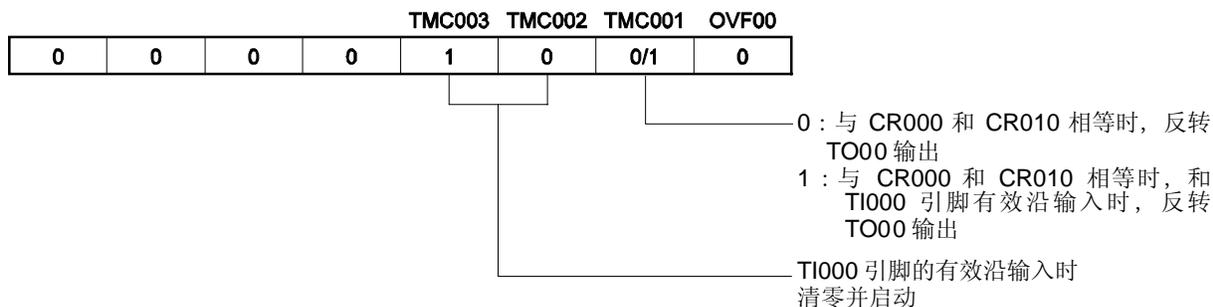
- 高电平宽度 = [CR010 的值] - [CR000 的值] × [计数时钟周期]
- 低电平宽度 = [CR000 的值] × [计数时钟周期]

如果选择 TI000 引脚脉冲的反向沿作为触发，捕捉计数值并存入 CR000，则不会产生 INTTM000 信号。在产生 INTTM010 信号后立即读取 CR000 和 CR010，测量脉冲宽度。

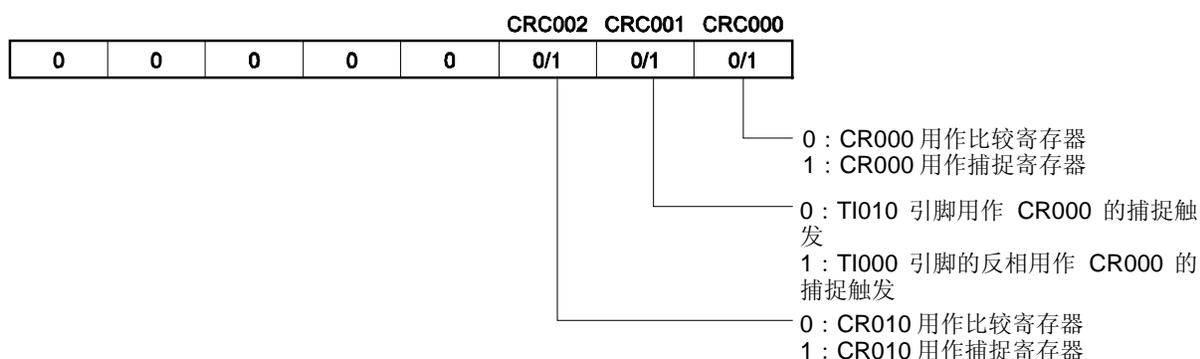
但如果将预分频器模式寄存器 00 (PRM00) 的第 6 位和第 5 位 (ES101 和 ES100) 指示的有效沿输入 TI010 引脚，则不会捕捉计数值，但会产生 INTTM000 信号。如果要测量 TI000 引脚的脉冲宽度，在不使用 INTTM000 信号时将其屏蔽。

图 6-31. 在清零&amp;启动模式(通过 TI000 引脚有效沿输入进入)下寄存器设置示例(1/2)

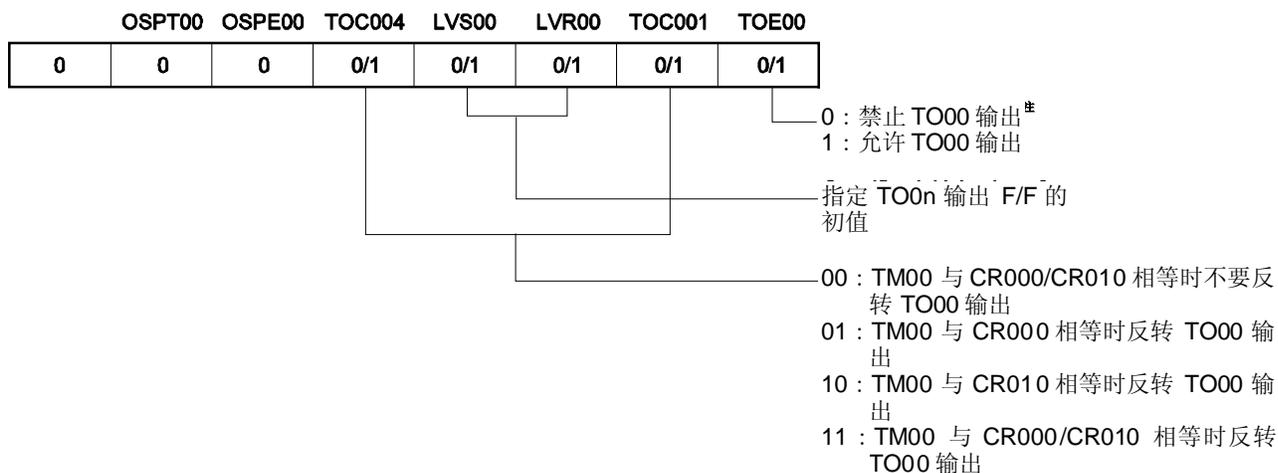
## (a) 16 位 定时器 模式控制寄存器 00 (TMC00)



## (b) 捕捉/比较控制寄存器 00 (CRC00)



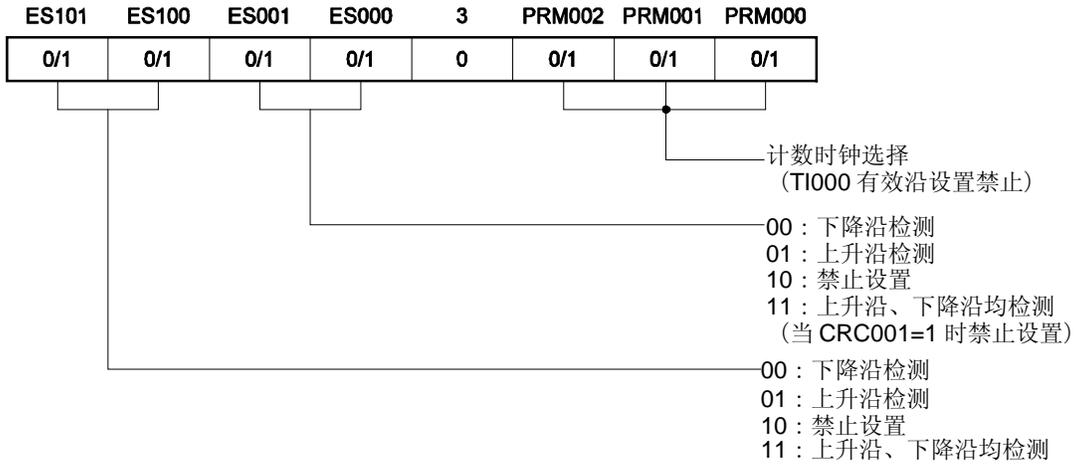
## (c) 16 位 定时器输出控制寄存器 00 (TOC00)



注 当使用检测 TI010 引脚有效沿时不能使用定时器输出(TO00)。

图 6-31. 在清零&amp;启动模式(通过 TI000 引脚有效沿输入进入)下寄存器设置示例(2/2)

## (d) 预分频模式寄存器 00 (PRM00)



## (e) 16 位定时器计数器 00 (TM00)

通过读取 TM00，获得计数值。

## (f) 16 位捕捉/比较寄存器 000 (CR000)

当该寄存器用作比较寄存器并且当其值与 TM00 的值相等时，产生中断信号 (INTTM000)。但不将 TM00 清零。

如果要将该寄存器用作捕捉寄存器，则应选择 TI000 或 TI010 引脚<sup>注</sup>输入作为捕捉触发。当检测到捕捉触发的有效沿时，将 TM00 的计数值存入 CR000。

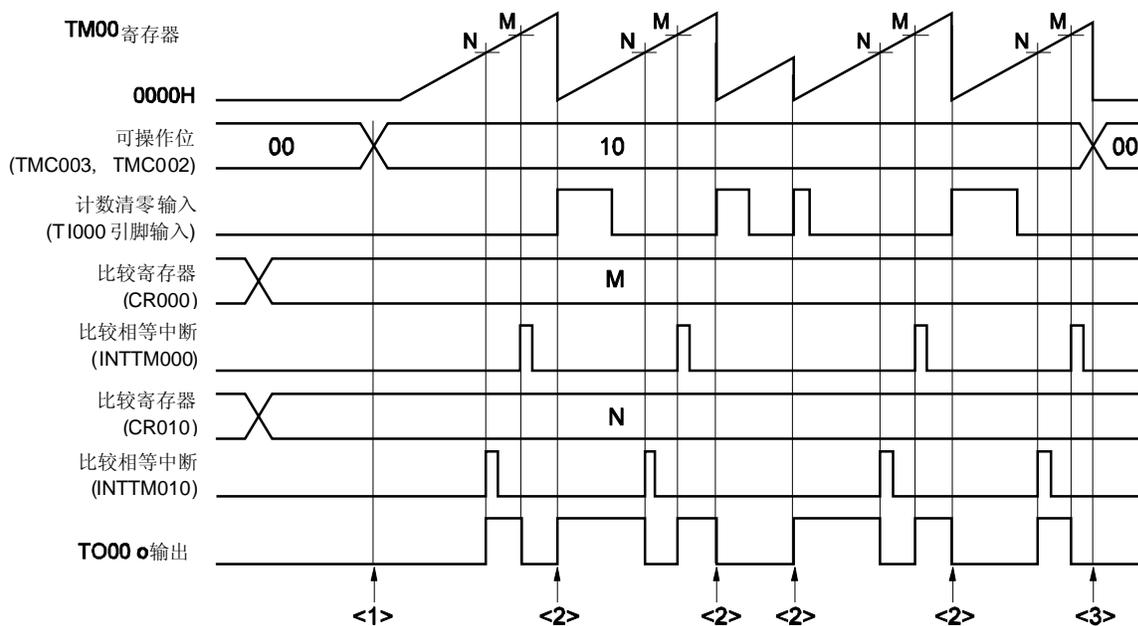
**注** 当使用检测 TI010 引脚的有效沿时不能使用定时器输出(TO00)。

## (g) 16 位捕捉/比较寄存器 010 (CR010)

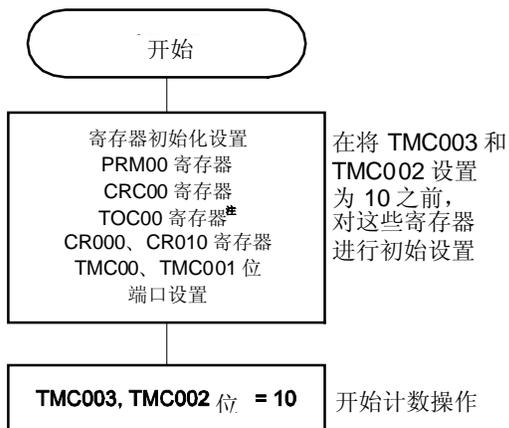
当该寄存器用作比较寄存器并且当其值与 TM00 的值相等时，产生中断信号 (INTTM010)。但不将 TM00 清零。

当该寄存器用作捕捉寄存器，则 TI000 引脚输入作为捕捉触发。当检测到捕捉触发的有效沿时，将 TM00 的计数值存入 CR010。

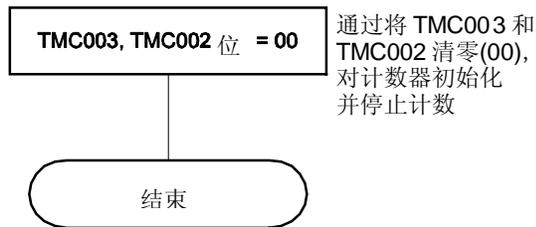
图 6-32. 清零 & 启动模式(通过 TI000 引脚有效沿输入进入)下软件处理示例



<1> 计数操作启动流程



<3> 计数操作停止流程



<2> TM00 寄存器清零&启动流程



注 设置 TOC00 时必须特别小心。如需了解详细信息，可参见 6.3 (3) 16 位 定时器输出控制寄存器 00 (TOC00)。

### 6.4.5 自由运行定时器操作

当 16 位定时器模式控制寄存器 00(TMC00)的第 3 位和第 2 位(TMC003 与 TMC002)为 01(自由运行定时器模式)时, 16 位定时器/事件计数器 00 连续计数并与计数时钟同步。当计数值上升到 FFFFH 时, 在下一个时钟将溢出标志(OVF00)设置为 1、将 TM00 清零(0000H)并继续计数。通过软件执行 CLR 指令将 OVF00 清零(0)。

可以使用以下三种自由运行定时器操作。

- CR000 和 CR010 都用作比较寄存器。
- CR000 与 CR010 一个用作比较寄存器, 另一个用作捕捉寄存器。
- CR000 和 CR010 都用作捕捉寄存器。

**备注**

1. 如需了解 I/O 引脚的设置, 参见 6.3 (6) 端口模式寄存器 3 (PM3)。
2. 要了解如何允许 INTTM000 信号中断, 参见第十七章 中断功能。

#### (1) 自由运行定时器 模式操作

(CR000: 比较 寄存器, CR010: 比较 寄存器)

图 6-33. 自由运行定时器 模式的框图  
(CR000: 比较 寄存器, CR010: 比较 寄存器)

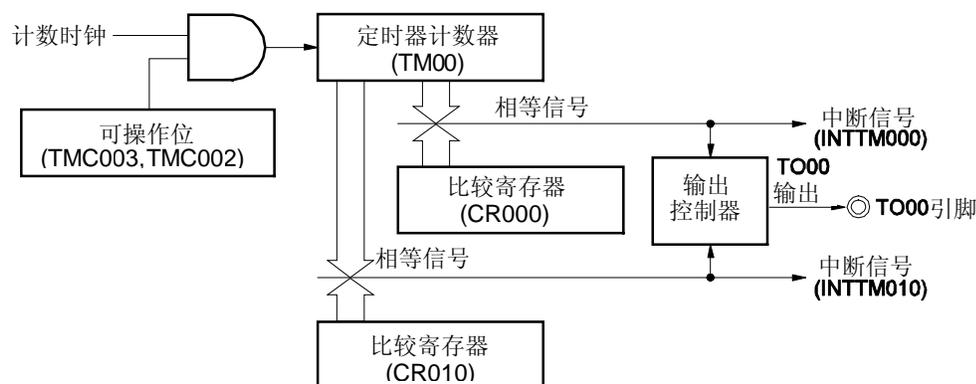
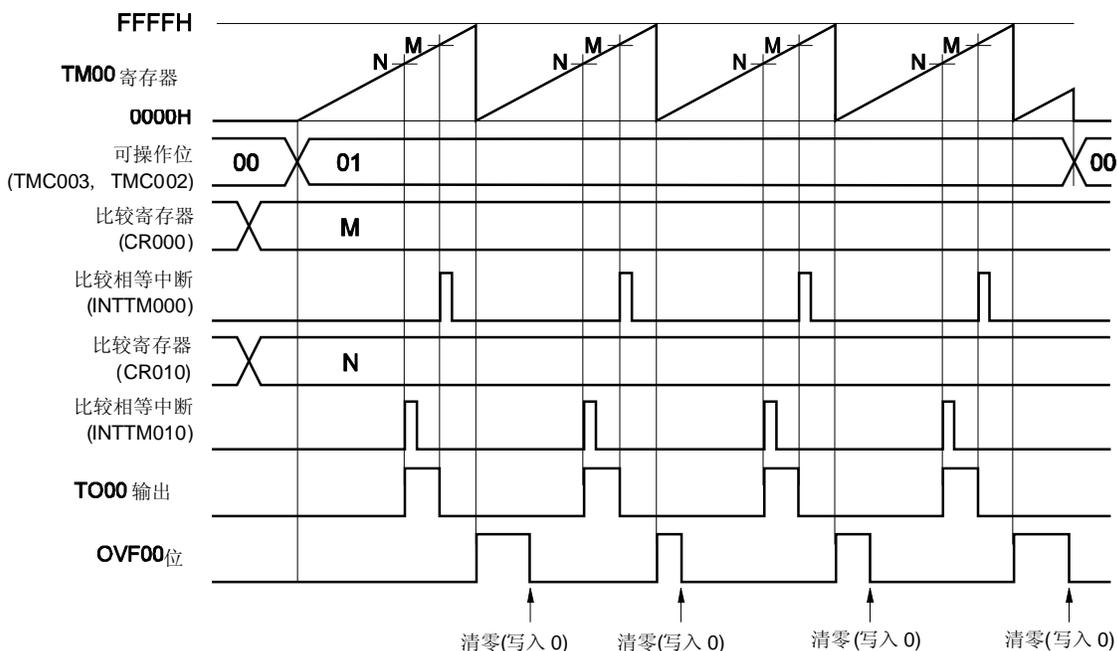


图 6-34. 自由运行定时器 模式时序示例  
(CR000: 比较 寄存器, CR010: 比较 寄存器)

• TOC00 = 13H, PRM00 = 00H, CRC00 = 00H, TMC00 = 04H



这是一个应用示例：两个比较寄存器在自由运行定时器模式下使用。  
每当 TM00 的计数值与 CR000 或 CR010 的值相等时，反转 TO00 引脚的输出电平。当计数值与寄存器值相等时，产生 INTTM000 或 INTTM010 信号。

(2) 自由运行定时器 模式操作  
(CR000: 比较 寄存器, CR010: 捕捉 寄存器)

图 6-35. 自由运行定时器 模式的框图  
(CR000: 比较 寄存器, CR010: 捕捉 寄存器)

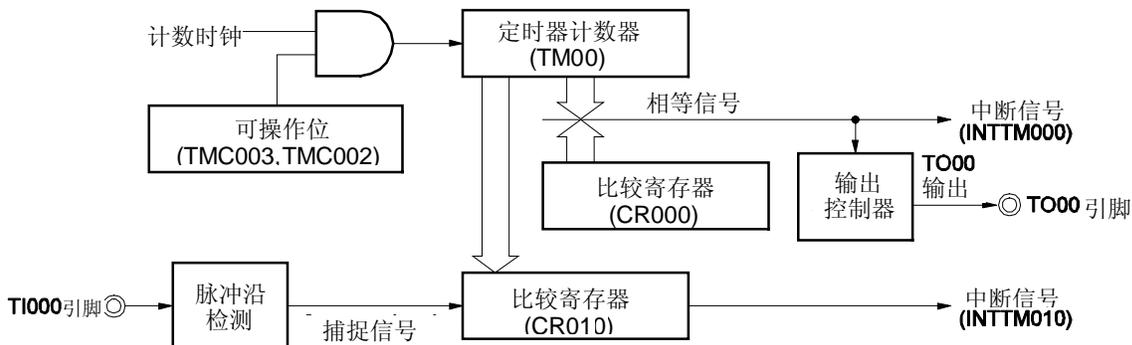
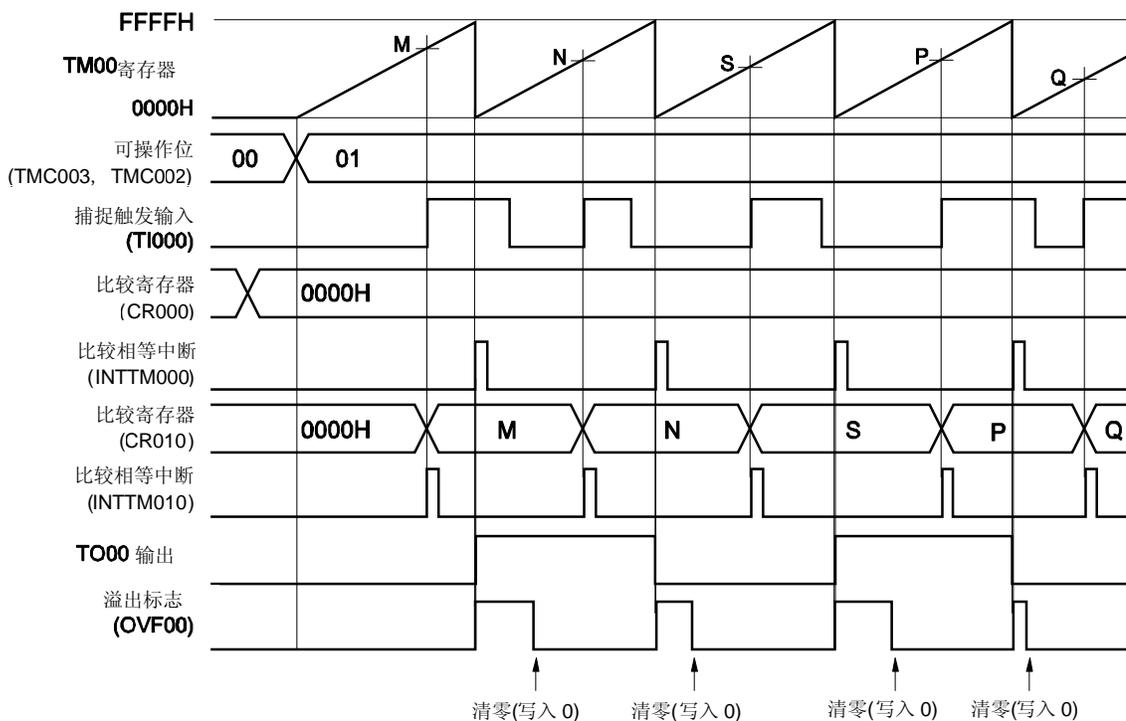


图 6-36. 自由运行定时器 模式时序示例  
(CR000: 比较 寄存器, CR010: 捕捉 寄存器)

• TOC00 = 13H, PRM00 = 10H, CRC00 = 04H, TMC00 = 04H

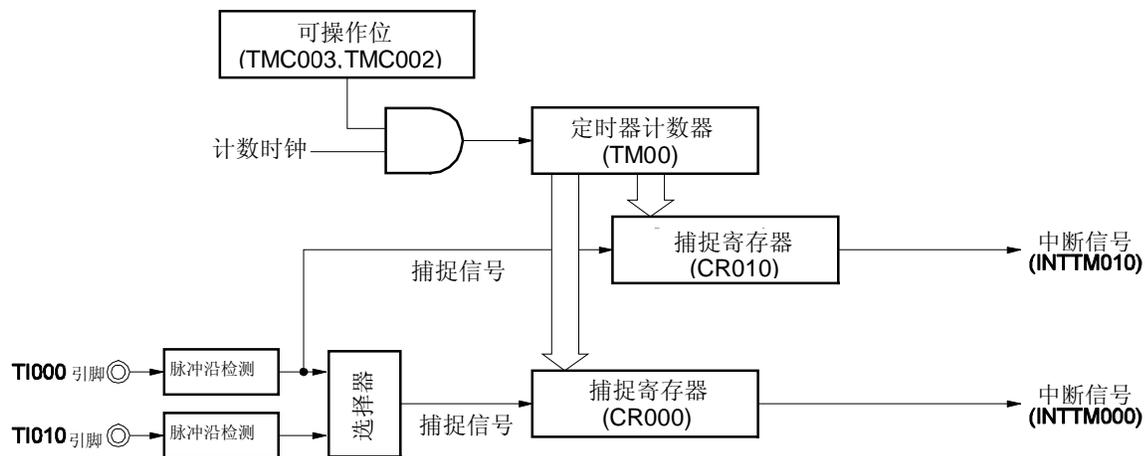


这是一个应用示例：在自由运行定时器模式下同时使用一个比较寄存器和一个捕捉寄存器。  
在本例中每当 TM00 的计数值与 CR000 的值相等时反转 TO00 引脚的输出电平，并且产生中断信号 INTTM000。  
此外每当检测到 TI000 引脚的有效沿时，产生 INTTM010 信号，并捕捉 TM00 的计数值存入 CR010。

## (3) 自由运行定时器 模式操作

(CR000: 捕捉寄存器, CR010: 捕捉寄存器)

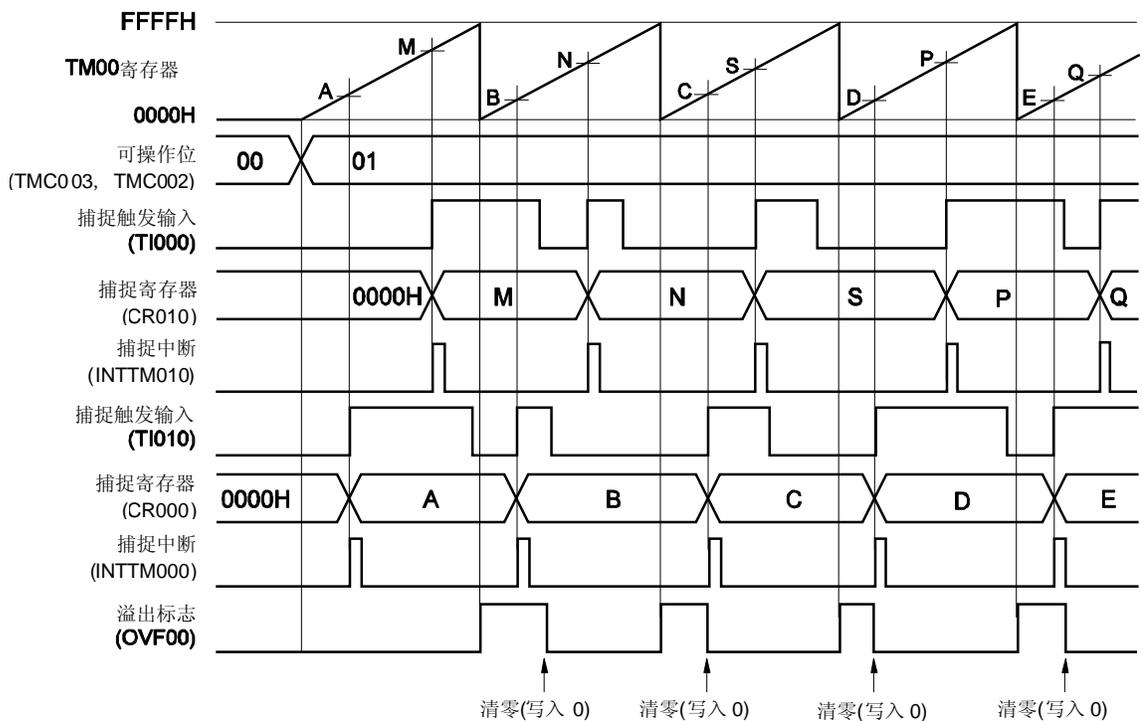
图 6-37. 自由运行定时器 模式的框图  
(CR000: 捕捉寄存器, CR010: 捕捉寄存器)



**备注** 如果在自由运行定时器模式下，CR000 与 CR010 用作捕捉寄存器，则不反转 TO00 引脚的输出电平。但如果 16 位定时器模式控制寄存器 00 (TMC00) 的第 1 位 (TMC001)=1，则每次检测到 TI000 引脚的有效沿时可以反转。

图 6-38. 自由运行定时器 模式时序示例  
(CR000: 捕捉 寄存器, CR010: 捕捉 寄存器) (1/2)

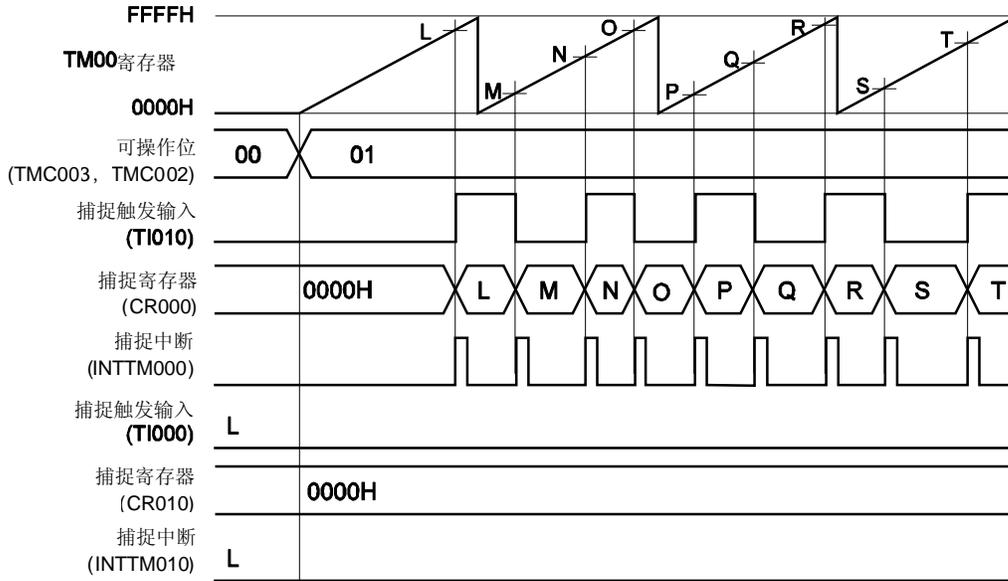
(a) TOC00 = 13H, PRM00 = 50H, CRC00 = 05H, TMC00 = 04H



这是一个应用示例：自由运行定时器模式下，在各自触发信号的有效沿捕捉的计数值存入各自的捕捉寄存器中。当检测到 TI000 引脚输入的有效沿时，捕捉计数值并存入 CR010，而当检测到 TI010 引脚输入的有效沿时，捕捉计数值并存入 CR000。

图 6-38. 自由运行定时器 模式时序示例  
(CR000: 捕捉 寄存器, CR010: 捕捉 寄存器) (2/2)

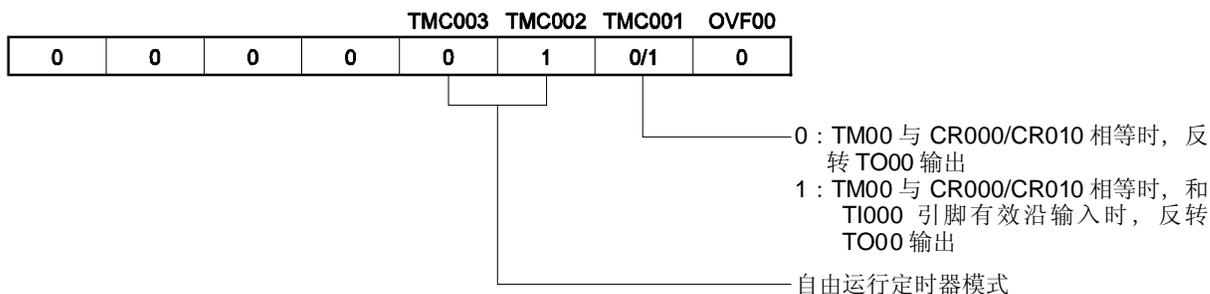
(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 04H



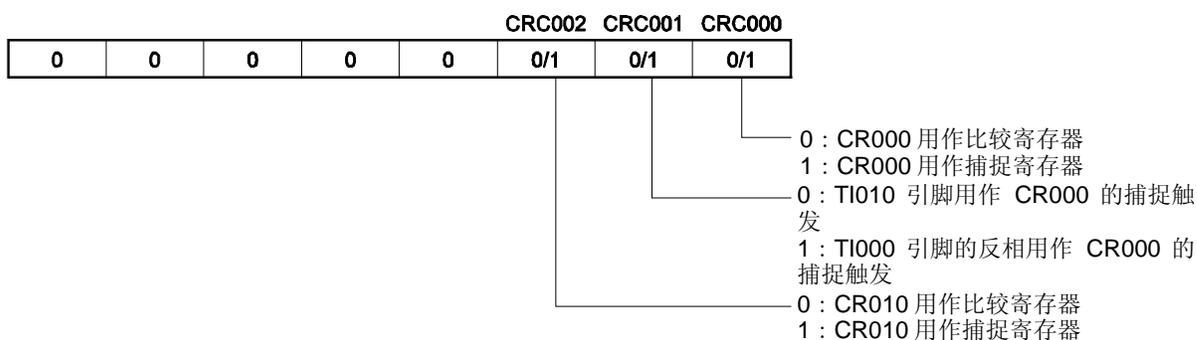
这是一个应用示例：自由运行定时器模式下，检测 TI010 引脚的两种脉冲沿，计数值被捕捉到 CR000。当 CR000 与 CR010 都用作捕捉寄存器并且仅检测到 TI010 引脚的有效沿时，计数值不能被捕捉到 CR010。

图 6-39. 自由运行定时器模式下寄存器设置示例(1/2)

## (a) 16 位 定时器 模式控制寄存器 00 (TMC00)



## (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)

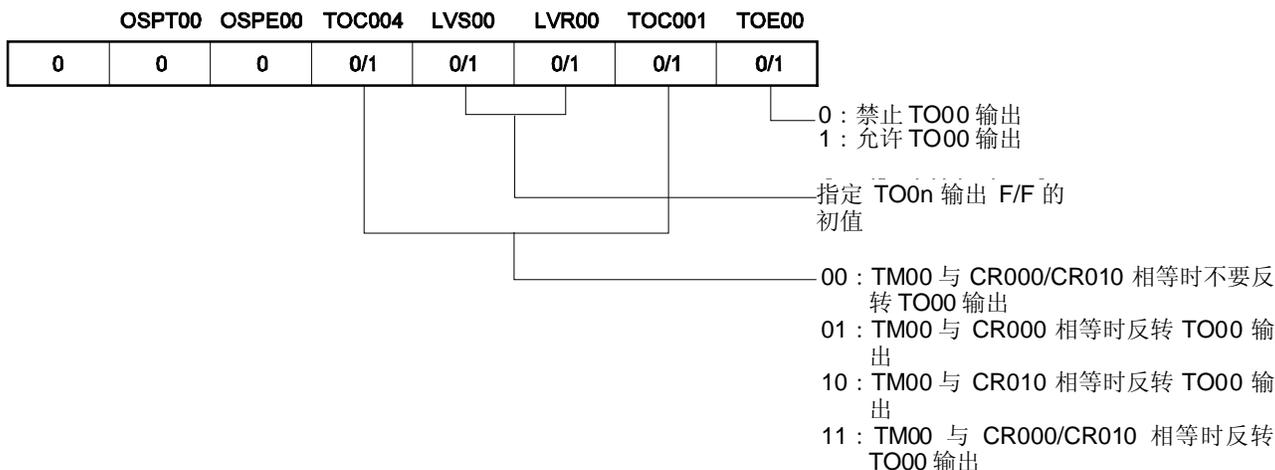
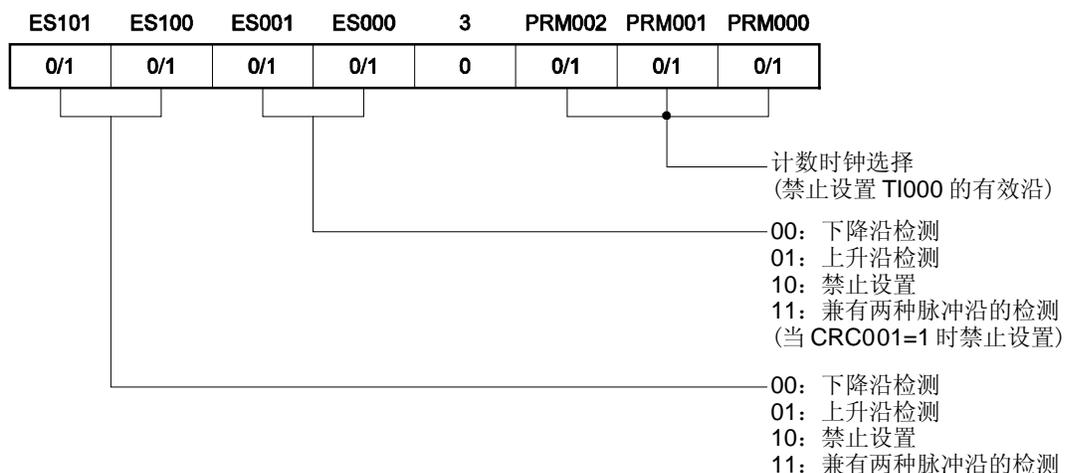


图 6-39. 自由运行定时器模式下寄存器设置示例(2/2)

## (d) 预分频 模式 寄存器 00 (PRM00)



## (e) 16 位定时器计数器 00 (TM00)

通过读取 TM00，获得计数值。

## (f) 16 位 捕捉/比较 寄存器 000 (CR000)

当该寄存器用作比较寄存器并且当其值与 TM00 的计数值相等时，产生中断信号(INTTM000)。但不对 TM00 计数值清零。

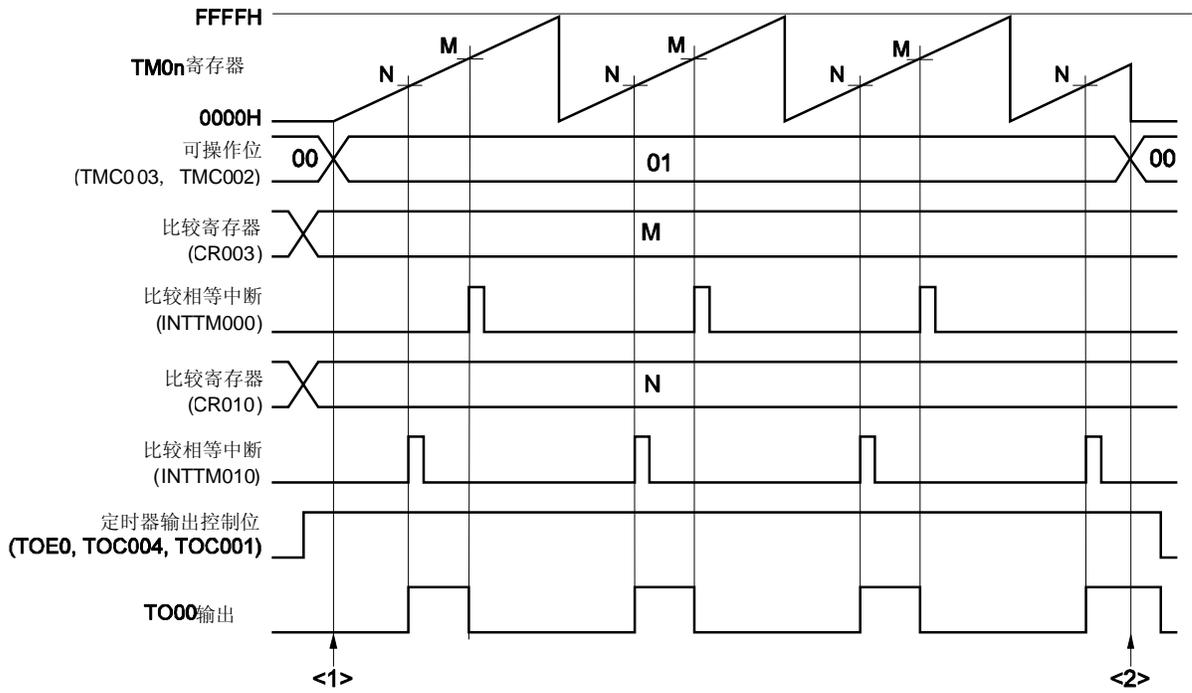
若要用作捕捉寄存器，则应选择 TI000 或 TI010 引脚输入作为捕捉触发。当检测到捕捉触发的有效沿时，将 TM00 的计数值存入 CR000。

## (g) 16 位 捕捉/比较 寄存器 010 (CR010)

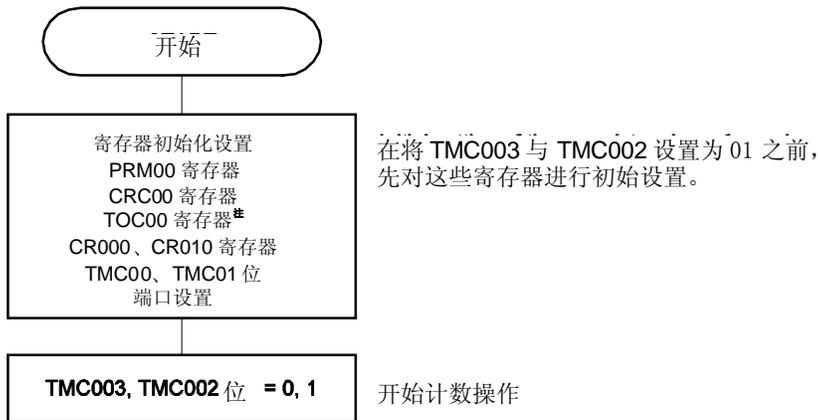
当该寄存器用作比较寄存器并且当其值与 TM00 的计数值相等时，产生中断信号(INTTM010)。但不对 TM00 计数值清零。

当该寄存器用作捕捉寄存器时，将 TI000 引脚输入作为捕捉触发。当检测到捕捉触发的有效沿时，将 TM00 的计数值存入 CR010。

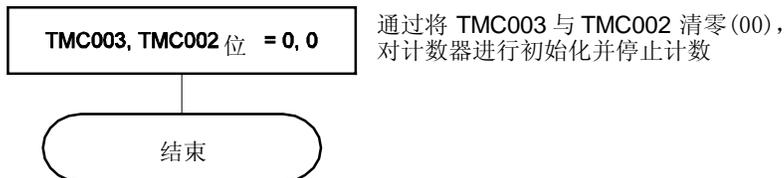
图 6-40. 自由运行定时器模式下软件处理示例



<1> 计数操作启动流程



<2> 计数操作停止流程



注 要特别注意 TOC00 的设置。如需了解详细信息，可参见 6.3 (3) 16 位 定时器输出控制寄存器 00 (TOC00)。

### 6.4.6 PPG 输出操作

当 16 位定时器模式控制寄存器 00 (TMC00) 的第 3 位和第 2 位(TMC003 与 TMC002)=11 时(在 TM00 与 CR000 相等时清零&启动), 在由 CR000 设置的周期内从 TO00 引脚输出方波(具有通过 CR010 预先设置的一个脉冲宽度)作为 PPG (可编程脉冲发生器) 信号。

以下是作为 PPG 输出的脉冲的周期与占空比。

- 脉冲周期 = (CR000 的设置值 + 1) × 计数时钟周期
- 占空比 = (CR010 的设置值 + 1) / (CR000 的设置值 + 1)

**注意事项** 若在操作期间修改占空比 (CR010 寄存器的值), 可参阅 6.5.1 TM00 操作期间 CR010 的重写。

- 备注**
1. 如需了解 I/O 引脚的设置, 参见 6.3 (6) 端口 模式 寄存器 0 (PM0)。
  2. 要了解如何允许 INTTM000 信号中断, 参见第十七章 中断功能。

图 6-41. PPG 输出操作的框图

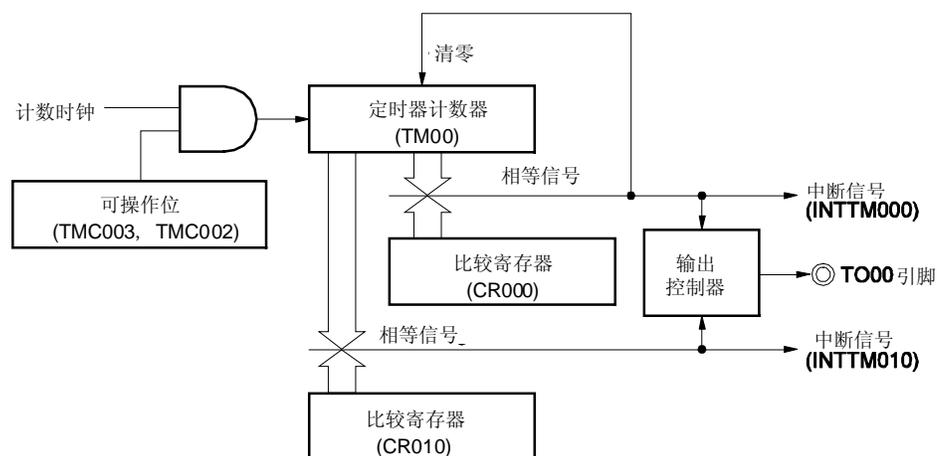
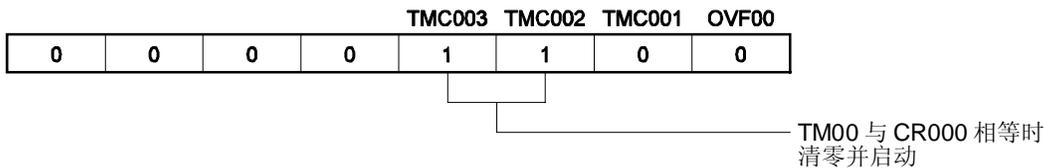
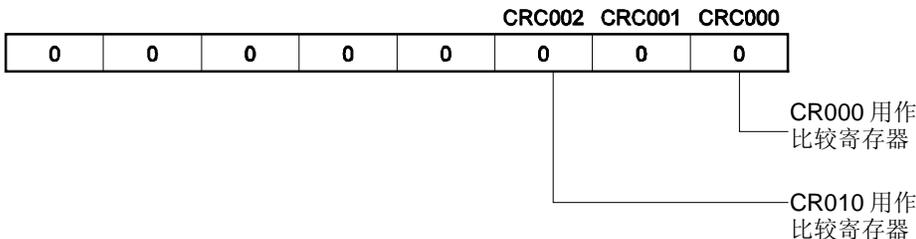


图 6-42. PPG 输出操作的寄存器设置示例

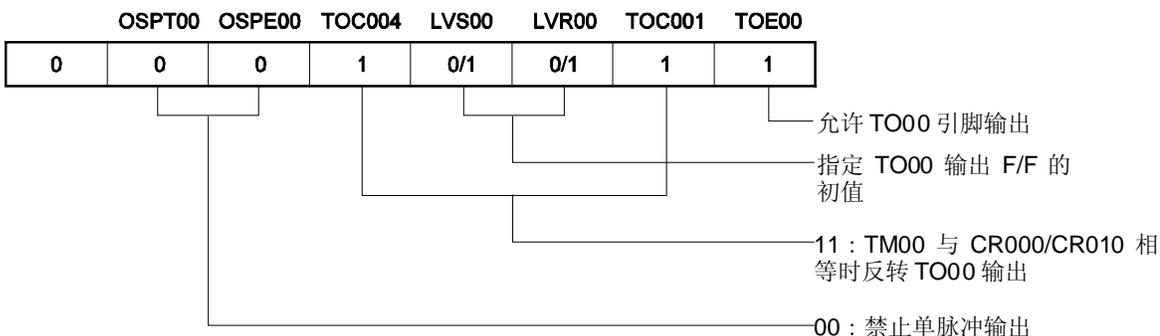
(a) 16 位定时器 模式控制寄存器 00 (TMC00)



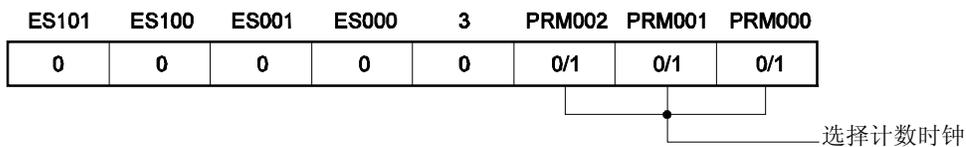
(b) 捕捉/比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频 模式 寄存器 00 (PRM00)



(e) 16 位定时器计数器 00 (TM00)

通过读取 TM00，获得计数值。

(f) 16 位 捕捉/比较 寄存器 000 (CR000)

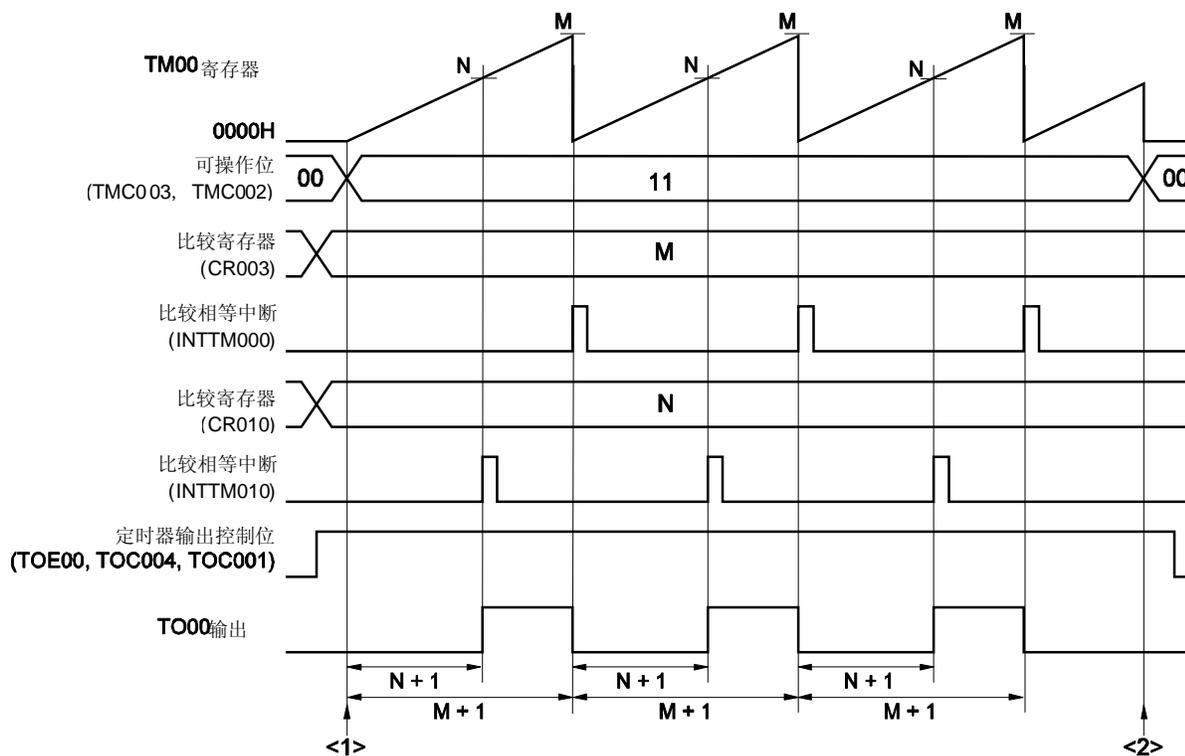
当该寄存器的值与 TM00 的计数值相等时，产生中断信号(INTTM000)。但不将 TM00 的计数值清零。

(g) 16 位 捕捉/比较 寄存器 010 (CR010)

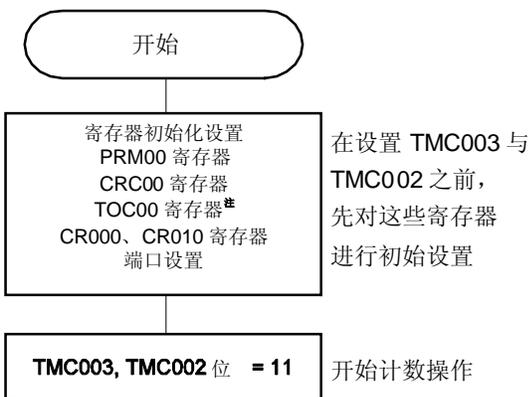
当该寄存器的值与 TM00 的计数值相等时，产生中断信号(INTTM010)。但不将 TM00 的计数值清零。

**注意事项** 对 CR000 与 CR010 进行设置，以满足条件  $0000H \leq CR010 < CR000 \leq FFFFH$ 。

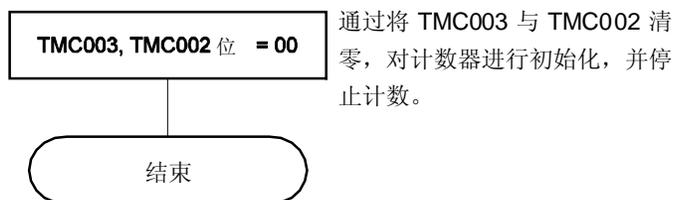
图 6-43. PPG 输出操作软件处理示例



<1> 计数操作启动流程



<2> 计数操作停止流程



**注** 必须特别注意 TOC00 的设置。如需了解详细信息，可参见 6.3 (3) 16 位 定时器输出控制寄存器 00 (TOC00)。

**备注** PPG 脉冲周期 = (M + 1) × 计数时钟周期  
PPG 占空比 = (N + 1)/(M + 1)

### 6.4.7 单脉冲输出操作

通过将 16 位定时器模式控制寄存器 00 (TMC00) 的第 3 位和第 2 位(TMC003 与 TMC002)设置为 01 (自由运行定时器模式)或 10(清零&启动模式 (通过 TI000 引脚有效沿进入)), 以及将 16 位定时器输出控制寄存器 00 (TOC00)的第 5 位 (OSPE00)设置为 1, 可以输出一个单脉冲。

当 TOC00 的第 6 位(OSPT00)=1 时或者在定时器操作期间将有效沿输入到 TI000 引脚时, 触发 TM00 的清零&启动, 并且从 TO00 引脚只输出一次反映 CR000 与 CR010 之间差距的脉冲。

- 注意事项**
1. 当单脉冲输出时, 不要再次输入触发(设置 OSPT00=1 或者检测 TI000 引脚的有效沿)。在当前单脉冲输出完成后产生触发, 才可再次输出单脉冲。
  2. 如果要使用单脉冲输出触发(仅设置 OSPT00=1), 则不要修改 TI000 引脚或其复用功能端口引脚的电平。否则可能会产生异常输出的脉冲。

- 备注**
1. 如需了解 I/O 引脚的设置, 参见 6.3 (6) 端口 模式 寄存器 3 (PM3)。
  2. 要了解如何允许 INTTM000 信号中断, 参见第十七章 中断功能。

图 6-44. 单脉冲输出操作的框图

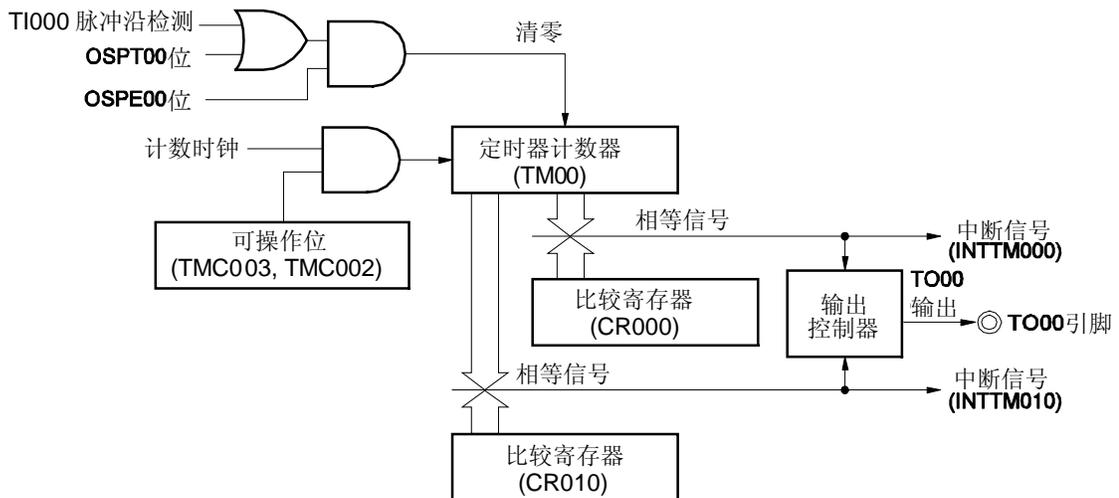


图 6-45. 单脉冲输出操作的寄存器设置示例(1/2)

(a) 16 位 定时器 模式控制寄存器 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	0/1	0/1	0	0

01: 自由运行定时器模式  
10: 使用 TI000 引脚的有效沿清零并启动模式

(b) 捕捉/比较控制寄存器 00 (CRC00)

				CRC002	CRC001	CRC000
0	0	0	0	0	0	0

CR000 用作比较寄存器  
CR010 用作比较寄存器

(c) 16 位 定时器输出控制寄存器 00 (TOC00)

OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0/1	1	1	0/1	0/1	1

允许 TO00 引脚输出  
指定 TO00 引脚输出的初值  
TM00 与 CR000/CR010 相等时反转 T000 输出  
允许单脉冲输出  
通过将该位设置为 1 产生软件触发 (即使设置为 0 也不影响操作)

(d) 预分频 模式 寄存器 00 (PRM00)

ES101	ES100	ES001	ES000	3	PRM002	PRM001	PRM000
0	0	0	0	0	0/1	0/1	0/1

选择计数时钟

图 6-45. 单脉冲输出操作的寄存器设置示例(2/2)

**(e) 16 位定时器计数器 00 (TM00)**

通过读取 TM00，可获得计数值。

**(f) 16 位 捕捉/比较 寄存器 000 (CR000)**

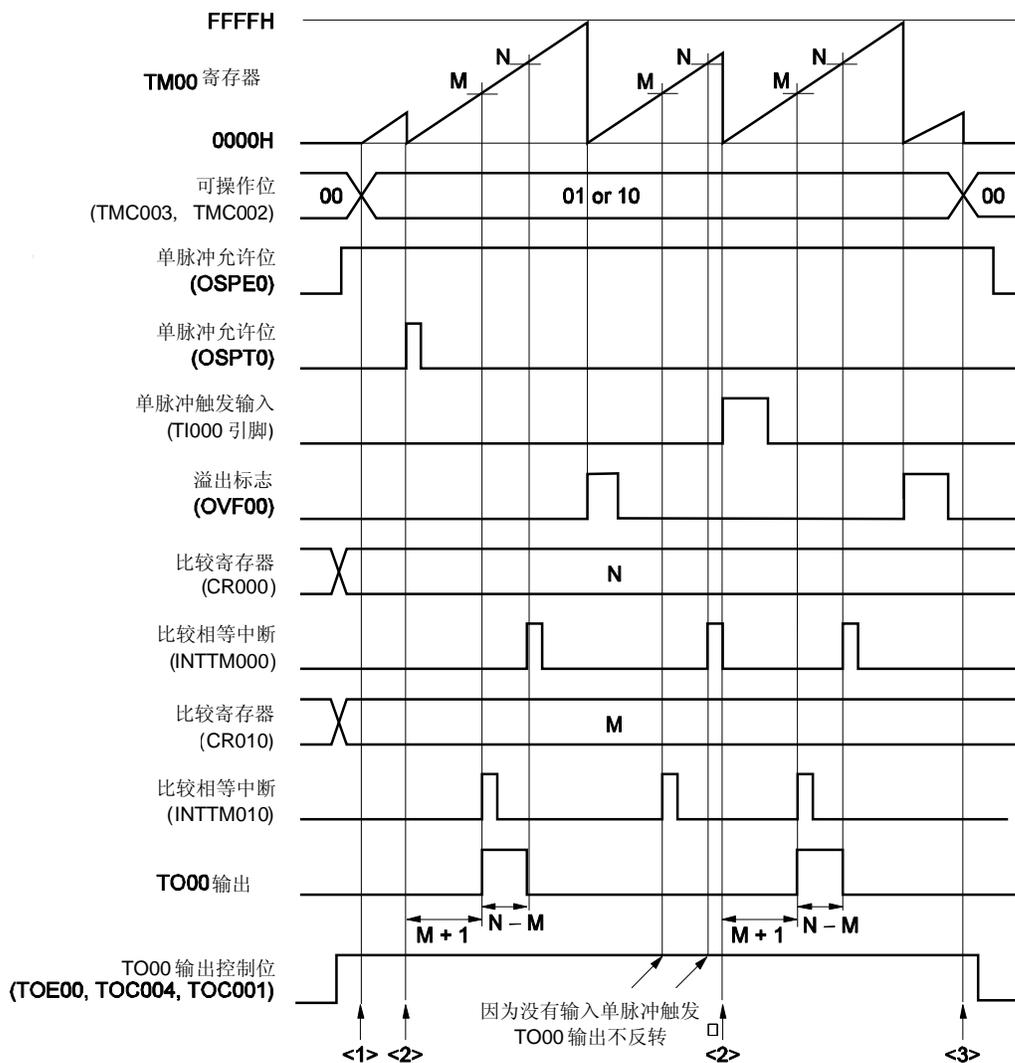
当输出单脉冲时该寄存器用作比较寄存器。当 TM00 的值与 CR000 的值相等时，产生中断信号(INTTM000)并反转 TO00 引脚的输出电平。

**(g) 16 位 捕捉/比较 寄存器 010 (CR010)**

当输出单脉冲时，该寄存器用作比较寄存器。当 TM00 的值与 CR010 的值相等时，产生中断信号(INTTM010)并反转 TO00 引脚的输出电平。

**注意事项** 不要为 CR000 与 CR010 设置相同的值或 0000H。

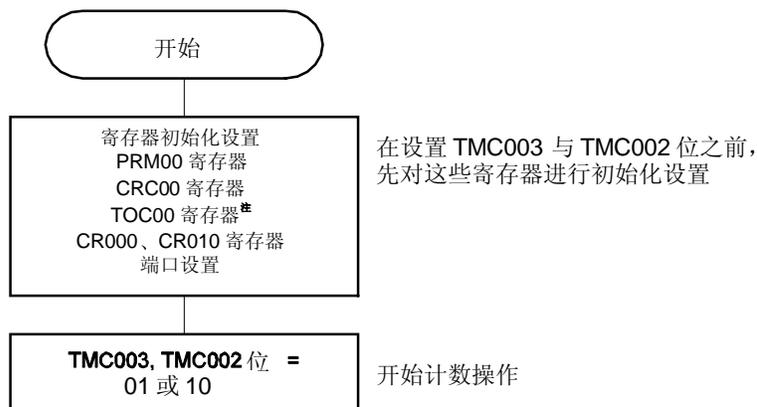
图 6-46. 单脉冲输出操作软件处理示例(1/2)



- 从输入单脉冲触发信号到输出单脉冲的时间  
=  $(M + 1) \times$  计数时钟周期
- 单脉冲输出活动电平宽度  
=  $(N - M) \times$  计数时钟周期

图 6-46. 单脉冲输出操作软件处理示例(2/2)

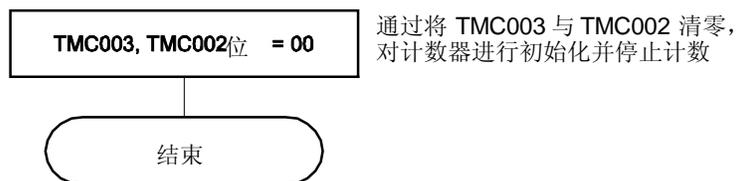
## &lt;1&gt; 计数操作启动流程



## &lt;2&gt; 单脉冲触发输入流程



## &lt;3&gt; 计数操作停止流程



注 必须特别注意 TOC00 的设置。如需了解详细信息，可参见 6.3 (3) 16 位 定时器输出控制寄存器 00 (TOC00)。

## 6.4.8 脉冲宽度测量操作

TM00 可用于测量输入到 TI000 与 TI010 引脚的信号脉冲宽度。

可以通过在自由运行定时器模式下操作 16 位定时器/事件计数器 00 以及通过同步使用 TI000 引脚输入信号的脉冲沿重新启动定时器，来进行测量。

当产生中断时，读取捕捉寄存器的有效值，并测量脉冲宽度。检测 16 位定时器模式控制寄存器 00 (TMC00) 的第 0 位(OVF00)。如果该位被设置为 1，则由软件将其清零。

图 6-47. 脉冲宽度测量 (自由运行定时器 模式)的框图

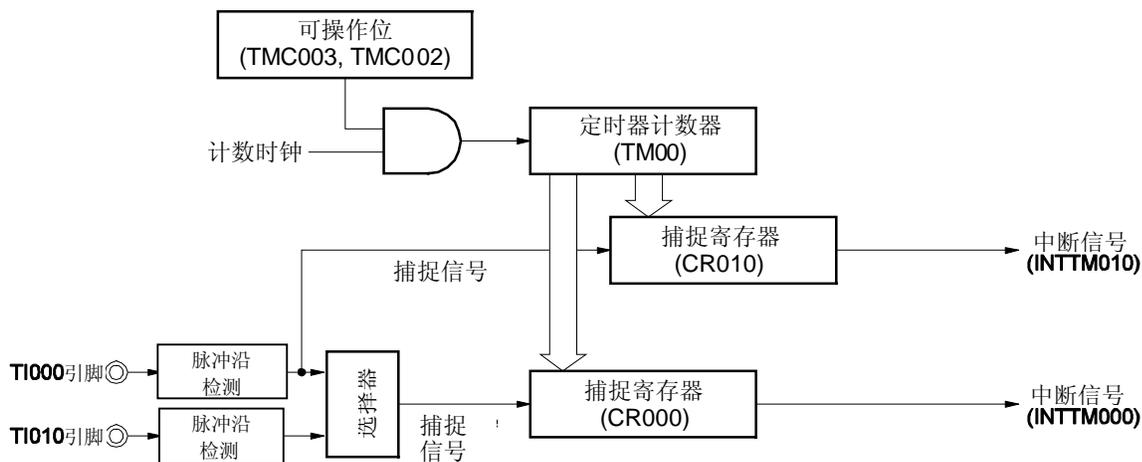
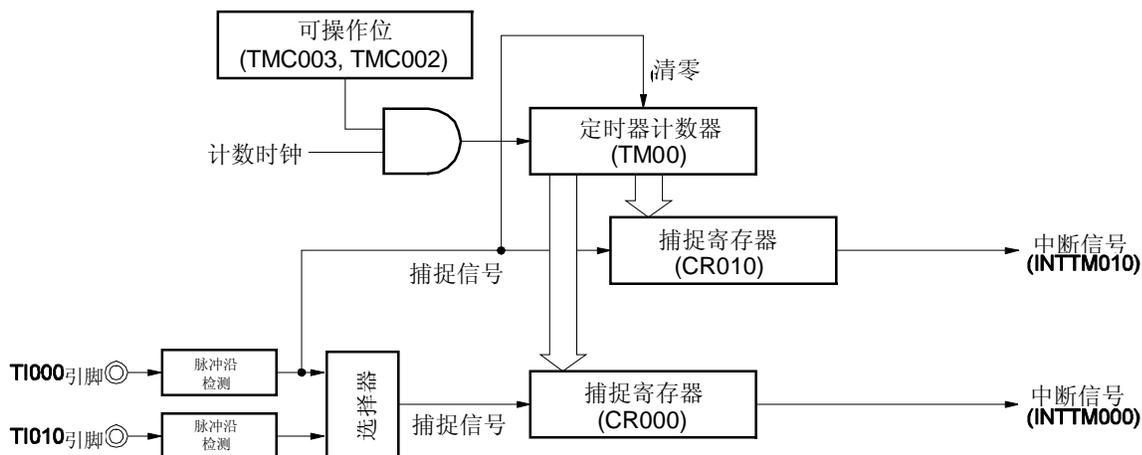


图 6-48. 脉冲宽度测量的框图  
(清零&启动模式(通过 TI000 引脚有效沿输入进入))



可以使用以下三种方法测量脉冲宽度。

- 通过使用 TI000 与 TI010 引脚的两个输入信号测量脉冲宽度(自由运行定时器模式)
- 通过使用 TI000 引脚的一个输入信号测量脉冲宽度(自由运行定时器模式)
- 通过使用 TI000 引脚的一个输入信号测量脉冲宽度(清零&启动模式(通过 TI000 引脚有效沿输入进入))

**注意事项** 当测量脉冲宽度时不要选择 TI000 有效沿作为计数时钟。

- 备注**
1. 如需了解 I/O 引脚的设置，参见 6.3 (6) 端口 模式 寄存器 3 (PM3)。
  2. 要了解如何允许 INTTM000 信号中断，参见第十七章 中断功能。

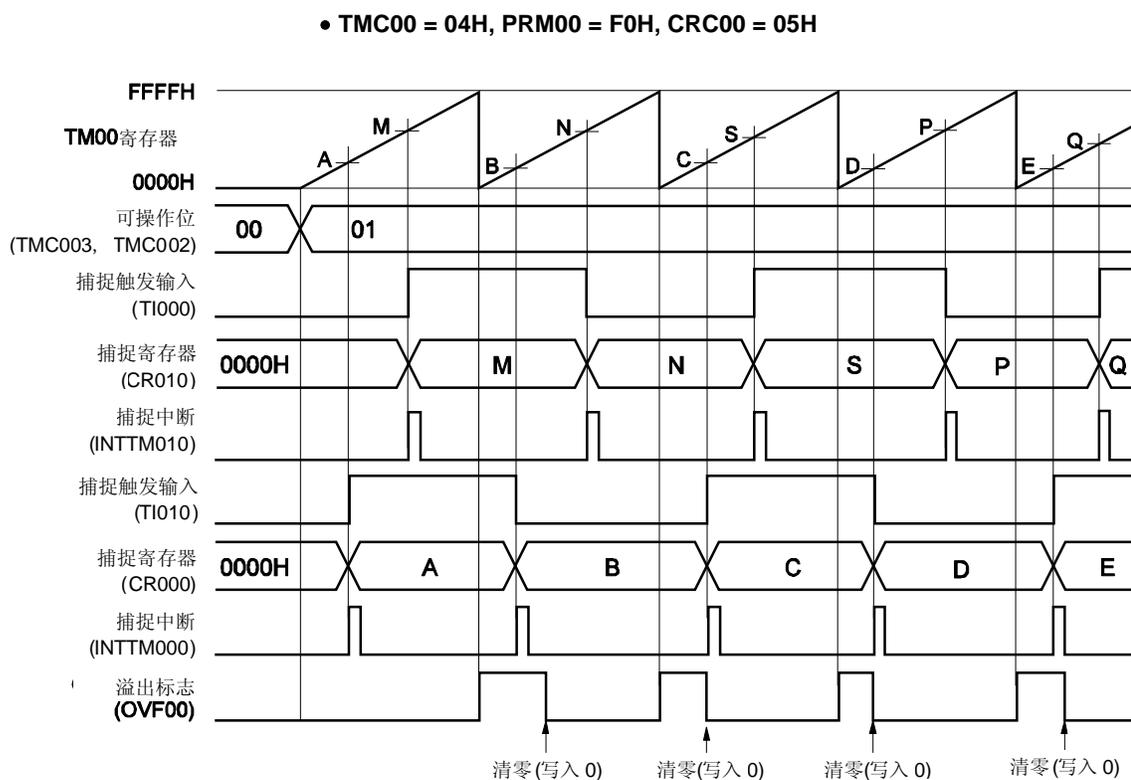
#### (1) 通过使用 TI000 与 TI010 引脚的两个输入信号测量脉冲宽度(自由运行定时器模式)

设置自由运行定时器模式(TMC003 与 TMC002 = 01)。当检测到 TI000 引脚的有效沿时，TM00 的计数值被捕捉到 CR010。当检测到 TI010 引脚的有效沿时，TM00 的计数值被捕捉到 CR000。规定检测 TI000 与 TI010 引脚的两种脉冲沿。

使用这种测量方法，要从每个输入信号的脉冲沿捕捉的计数值中减去预设值。因此要将先前捕捉的值事先存入一个单独的寄存器中。

如果产生溢出，则在当前捕捉值减去先前捕捉的值后计数值为负，并因此产生借位(将程序状态字(PSW)的第 0 位(CY)置 1)。如果发生这种情况，可以忽略 CY 并将计算结果作为脉冲宽度。此外，将 16 位定时器模式控制寄存器 00 (TMC00)的第 0 位(OVF00)清零(0)。

图 6-49. 脉冲宽度测量时序示例 (1)



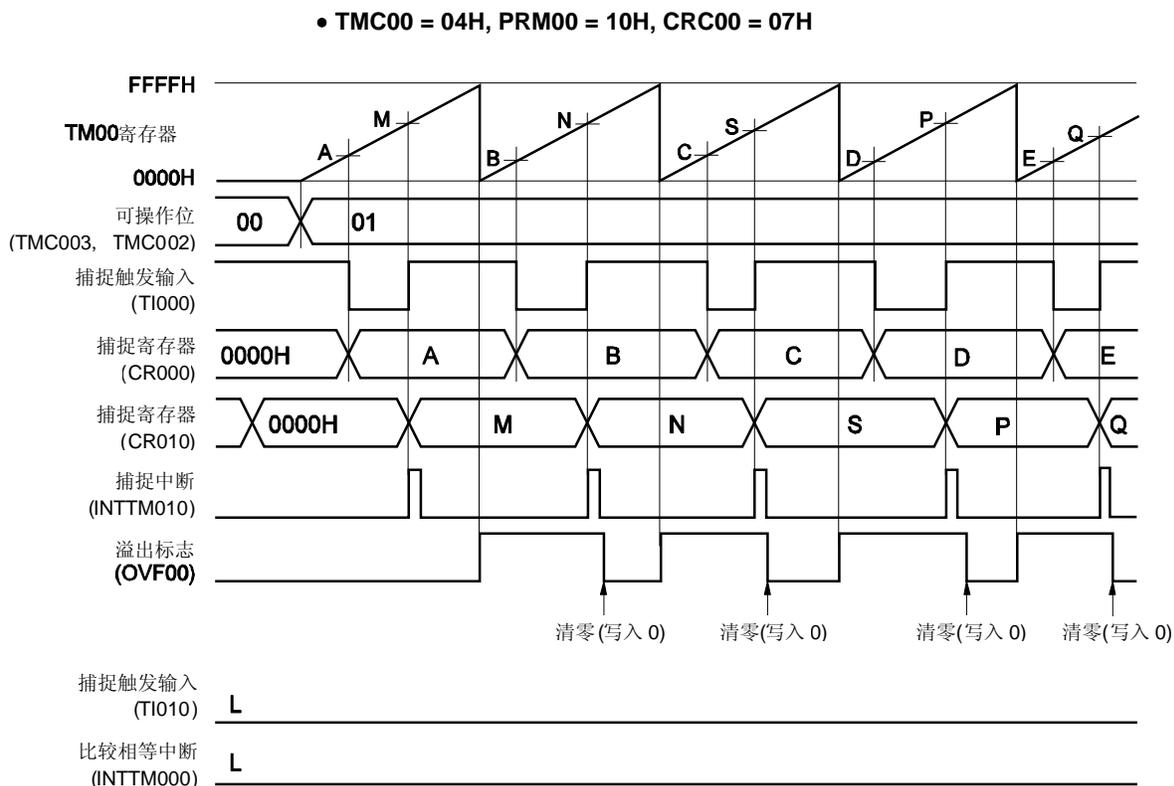
**(2) 通过使用 TI000 引脚的一个输入信号测量脉冲宽度(自由运行模式)**

设置自由运行定时器模式(TMC003 与 TMC002 = 01)。在 TI000 引脚检测到反向的有效沿时, 将 TM00 的计数值捕捉到 CR000。当检测到 TI000 引脚的有效沿时, 将 TM00 的计数值捕捉到 CR010。

使用这种测量方法, 在测量从一个脉冲沿到另一个脉冲沿的宽度时将数值存入分开的两个捕捉寄存器中。因此, 不必保存捕捉值。将两个捕捉寄存器的值相减, 可以计算高电平宽度、低电平宽度和周期。

如果产生溢出, 则当两个捕捉的计数值相减后结果为负, 并因此产生借位(将程序状态字(PSW)的第 0 位(CY)置 1)。如果发生这种情况, 可以忽略 CY 并将计算结果作为脉冲宽度。此外, 将 16 位定时器模式控制寄存器 00 (TMC00)的第 0 位(OVF00)清零(0)。

图 6-50. 脉冲宽度测量时序示例 (2)



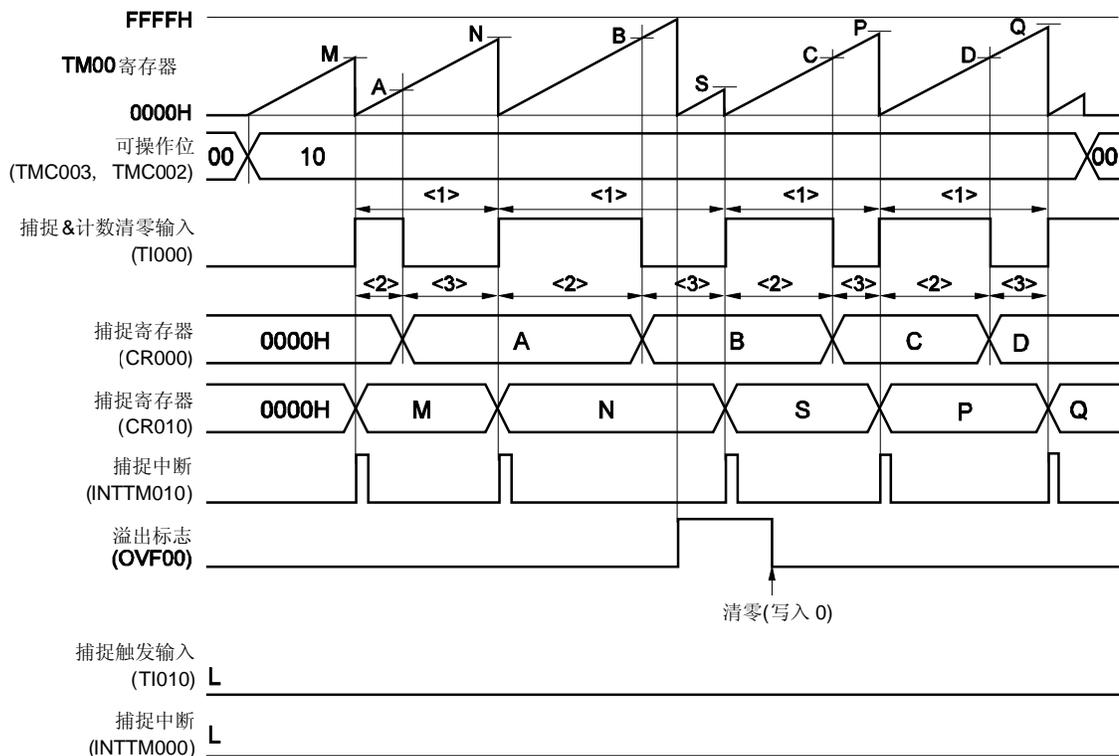
**(3) 通过使用 TI000 引脚的一个输入信号测量脉冲宽度(清零&启动模式(通过 TI000 引脚有效沿输入进入))**

设置清零&启动模式(通过 TI000 引脚有效沿进入)(TMC003 与 TMC002 = 10)。在 TI000 引脚检测到反向的有效沿时, 将 TM00 的计数值捕捉到 CR000。当检测到 TI000 引脚的有效沿时, 将 TM00 的计数值捕捉到 CR010, 并将 TM00 清零(0000H)。因此, 如果 TM00 没有溢出, 则在 CR010 中存入一个周期。

如果产生溢出, 则将 10000H 加上 CR010 的值, 结果作为一个周期。此外, 将 16 位定时器模式控制寄存器 00 (TMC00)的第 0 位(OVF00)清零(0)。

图 6-51. 脉冲宽度测量时序 (3)

• TMC00 = 08H, PRM00 = 10H, CRC00 = 07H



- <1> 脉冲周期 =  $(10000H \times \text{OVF00 位被设置为 1 的次数} + \text{CR010 的捕捉值}) \times \text{计数时钟周期}$   
 <2> 高电平脉冲宽度 =  $(10000H \times \text{OVF00 位被设置为 1 的次数} + \text{CR000 的捕捉值}) \times \text{计数时钟周期}$   
 <3> 低电平脉冲宽度 =  $(\text{脉冲周期} - \text{高电平脉冲宽度})$

图 6-52. 脉冲宽度测量的寄存器设置示例(1/2)

(a) 16 位 定时器 模式控制寄存器 00 (TMC00)

TMC003				TMC002		TMC001	OVF00
0	0	0	0	0/1	0/1	0	0

01: 自由运行定时器模式  
10: 清零并启动模式  
(通过 TI000 引脚的有效沿进入)

(b) 捕捉/比较控制寄存器 00 (CRC00)

CRC002				CRC001		CRC000	
0	0	0	0	0	1	0/1	1

1 : CR000 用作捕捉寄存器  
0 : TI010 引脚用作 CR000 的捕捉触发  
1 : TI000 引脚的反相用作 CR000 的捕捉触发  
1 : CR010 用作捕捉寄存器

(c) 16 位 定时器输出控制寄存器 00 (TOC00)

OSPT00		OSPE00		TOC004		LVS00		LVR00		TOC001		TOE00	
0	0	0	0	0	0	0	0	0	0	0	0	0	0

(d) 预分频 模式 寄存器 00 (PRM00)

ES101		ES100		ES001		ES000		3	PRM002		PRM001		PRM000	
0/1	0/1	0/1	0/1	0/1	0/1	0	0	0	0/1	0/1	0/1	0/1	0/1	0/1

选择计数时钟  
(禁止设置 TI000 的有效沿)  
00 : 下降沿检测  
01 : 上升沿检测  
10 : 禁止设置  
11 : 两种脉冲沿均检测  
(当 CRC001=1 时禁止设置)  
00 : 下降沿检测  
01 : 上升沿检测  
10 : 禁止设置  
11 : 两种脉冲沿均检测

图 6-52. 脉冲宽度测量的寄存器设置示例(2/2)

**(e) 16 位定时器计数器 00 (TM00)**

通过读取 TM00，可获得计数值。

**(f) 16 位 捕捉/比较 寄存器 000 (CR000)**

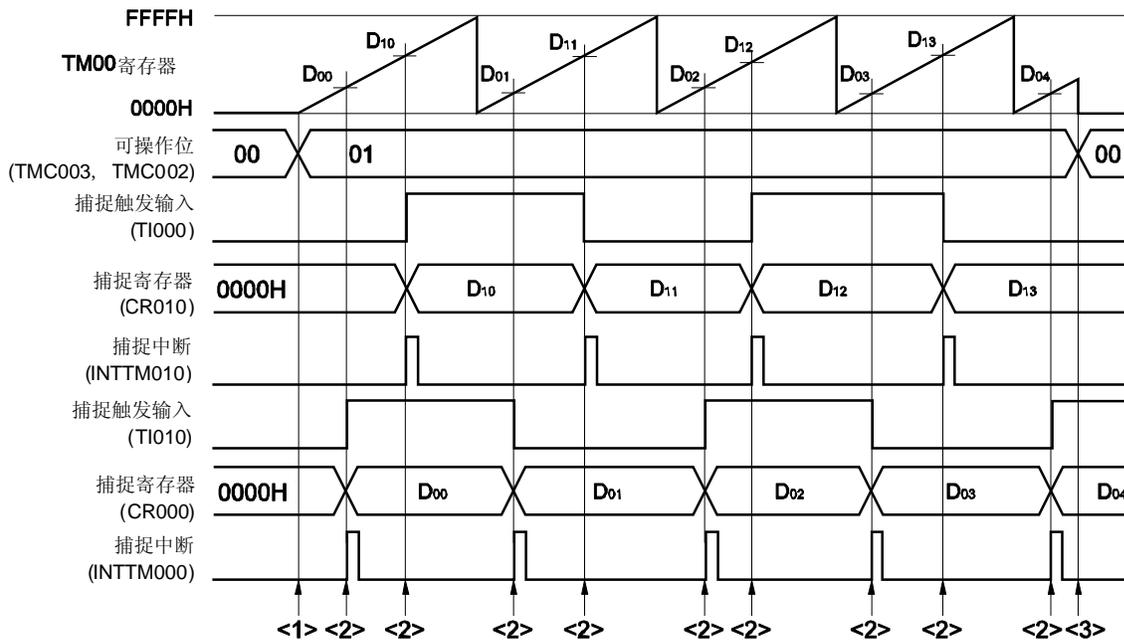
该寄存器用作捕捉寄存器。选择 TI000 或 TI010 引脚作为捕捉触发。当检测到指定脉冲沿的捕捉触发时，将 TM00 的计数值存入 CR000。

**(g) 16 位 捕捉/比较 寄存器 010 (CR010)**

该寄存器用作捕捉寄存器。将 TI000 引脚的输入信号用作捕捉触发。当检测到捕捉触发时，将 TM00 的计数值存入 CR010。

图 6-53. 脉冲宽度测量软件处理示例(1/2)

(a) 自由运行定时器模式示例



(b) 清零&启动模式(通过 TI000 引脚有效沿进入)示例

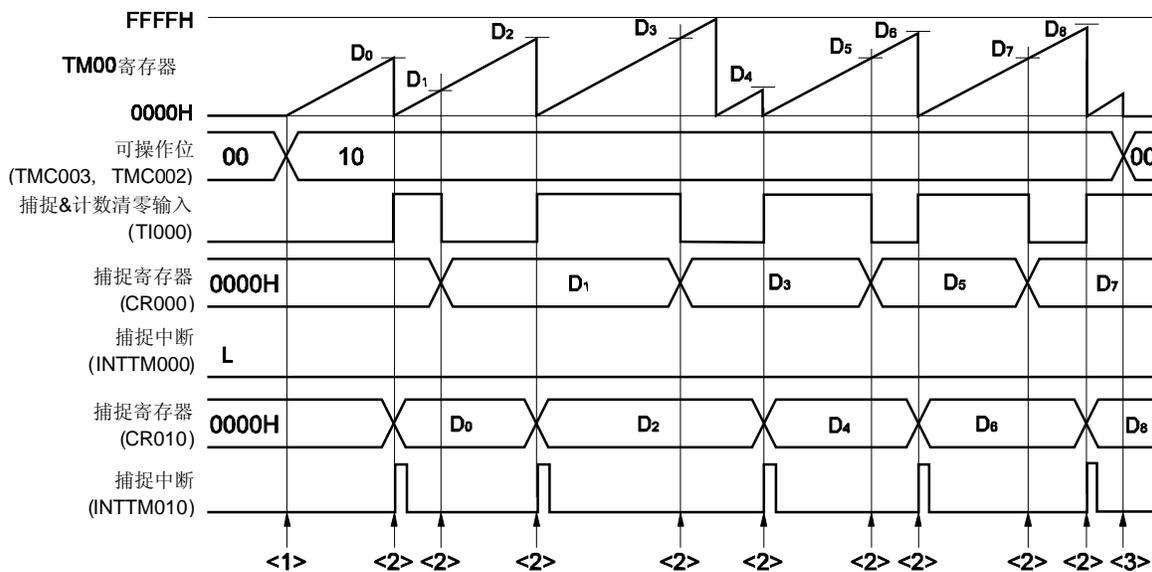
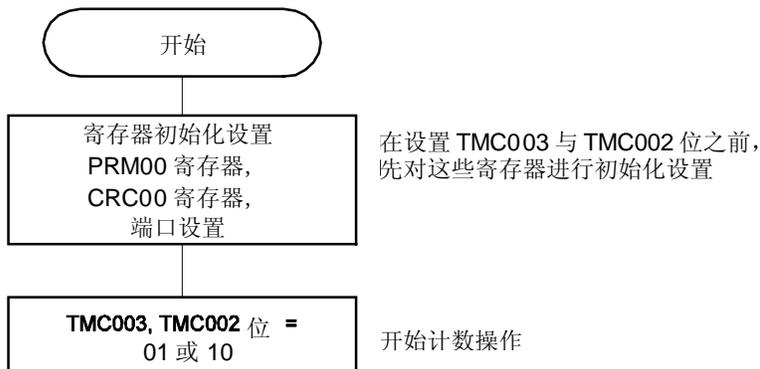
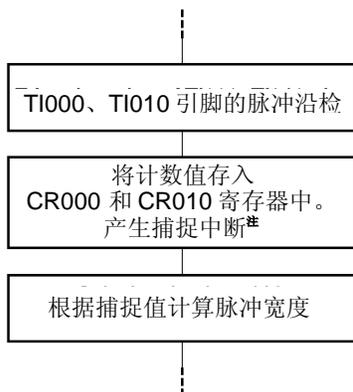


图 6-53. 脉冲宽度测量软件处理示例(2/2)

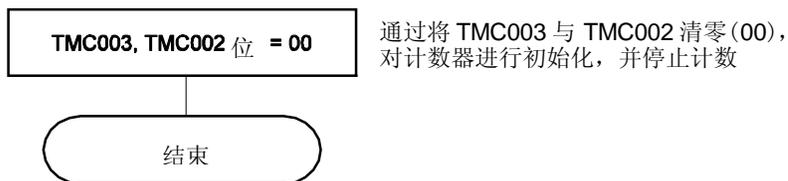
## &lt;1&gt; 计数操作启动流程



## &lt;2&gt; 捕捉触发输入流程



## &lt;3&gt; 计数操作停止流程



注 当选择 TI000 引脚输入的反向脉冲沿作为 CR000 的有效沿时, 不产生捕捉中断信号(INTTM000)。

#### 6.4.9 外部 24 位事件计数器操作

16 位定时器/事件计数器 00 通过级联 16 位定时器/事件计数器 00 和 8 位 定时器/时间计数器 52，可用于外部 24 位事件计数器功能操作，8 位 定时器/事件计数器 52 作为外部事件计数器使用。

作为外部 24 位事件计数器使用，计算经过 8 位定时器 52 (TM52) 输入到 TI52 引脚的外部时钟脉冲数，和计算当 TM52 计数值与 8 位定时器比较寄存器 52 (CR52=FFH<sup>注</sup>) 时由 16 位定时器计数器 00 (TM00) 输出的信号。

当 16 位定时器/事件计数器 00 用于外部 24 位事件计数器时，可通过 8 位定时器计数器 H2 输出控制外部事件输入允许。

输入到 TI52 引脚的有效沿可由 8 位定时器计数器 52 (TM52)的定时器时钟选择寄存器 52 (TCL52) 指定。TM52 外部事件输入允许也可以通过 8 位定时器计数器 H2 输出控制，设置输入切换控制寄存器 (ISC) 的第 2 位 (ISC2) 为“1”。

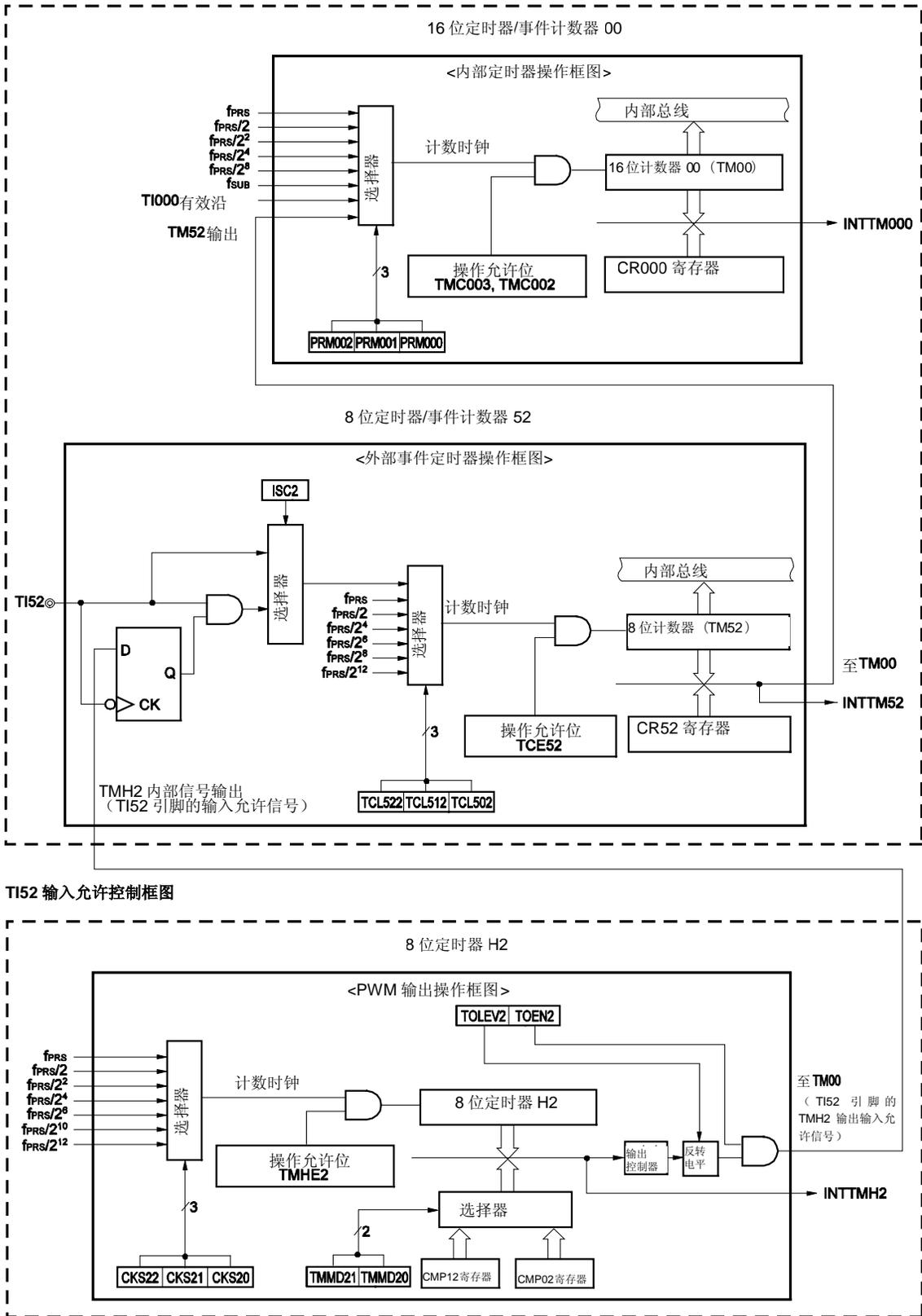
通过设置 16 位定时器/事件计数器 00 的预分频 模式 寄存器 00 (PRM00) 的第 2, 1 和 0 位 (PRM002, PRM001, 和 PRM000) 为 1, 1 和 1, 和设置 16 位 定时器 模式控制寄存器 00 (TMC00)的第 3 和 2 位 (TMC003 和 TMC002) 为 1 和 1 (TM00 和 CR000 相等时进入计数清零&启动模式)，用作计数时钟的 8 位 定时器 52 输出计数操作开始。TM00 计数值与 16 位 定时器 比较 寄存器 000 (CR000)的值相等时 TM00 被清零并且产生一个中断请求信号 (INTTM00)。

随后，TM00 和 CR000 的值相等时产生 INTTM00。

**注** 当 16 位定时器/事件计数器 00 作为外部 24 位事件计数器操作时，8 位 定时器 比较 寄存器 52 (CR52)的值必须设置为 FFH。而且，TM52 中断请求信号 (INTTM52) 必须被屏蔽掉 (TMMK52 = 1)。

图 6-54. 外部 24 位事件计数器的框图

外部 24 位事件计数器的框图



## 设置

<1> TM00 和 TM52 的每种模式的设置。

(a) TM00 设置为间隔定时器。选择 TM52 输出作为计数时钟。

- TMC00: 设置操作禁止。  
(TMC00 = 00000000B)
- CRC00: 设置作为比较 寄存器操作。  
(CRC00 = 000000x0B, x = don't care)
- TOC00: 设置 CR000 和 TM00 相等时 TO00 引脚输出禁止。  
(TOC00 = 00000000B)
- PRM00: TM52 输出 选择作为计数时钟。  
(PRM00 = 00000111B)
- CR000: 设置比较值为 FFFFH。  
如果比较值设置为 M, TM00 计数将到 M。
- CR010: 通常, CR010 不使用, 但是, 当 CR010 的设置值和 TM00 值相等时产生一个比较相等中断 (INTTM010)。因此, 通过中断屏蔽标志(TMMK010)屏蔽中断请求。

(b) 设置 TM52 作为外部事件计数器。

- TCL52: TI52 引脚输入的沿选择  
TI52 引脚的下降沿 → TCL52 = 00H  
TI52 引脚的上升沿 → TCL52 = 01H
- CR52: 设置比较寄存器的值为 FFH。
- TMC52: 计数操作停止。  
(TMC52 = 00000000B)
- TMIF52: 清零此寄存器。

**注意事项** 当 16 位定时器/事件计数器 00 作为外部 24 位事件计数器操作时, INTTM52 必须屏蔽 (TMMK52 = 1)。因此, 比较 寄存器 52 (CR52) 的值必须设置为 FFH。

(c) 为 TI52 引脚设置 TMH2 到输入允许宽度判定模式 (PWM 模式)。\*<sup>‡</sup>

- TMHMD2: 计数操作停止, 选择计数时钟, 模式设置到输入允许宽度判定模式 (PWM 模式), 定时器输出 电平默认值设置为高电平, 定时器输出 设置为允许(TMHMD2 = 0xxx1011B, x = 根据使用条件设置)。
- CMP02: 比较值 (N) 频率设置
- CMP12: 比较值 (M) 占空比设置  
备注  $00H \leq \text{CMP12 (M)} < \text{CMP02 (N)} \leq \text{FFFH}$
- ISC2: 设置 ISC2 = 1 (TI52 引脚输入允许控制)

**注** 如果 TI52 引脚输入允许不控制, 此设置不需要。

<2> TM00, TM52, 和 TMH2 计数操作开始。定时器操作开始必须按照下面的过程。

- (a) 通过设置 TMC003 和 TMC002 位为 1 和 1 启动 TM00 计数器操作。
- (b) 通过设置 TCE52 为 1 启动 TM52 计数器操作。
- (c) 通过设置 TMHE2 为 1 启动 TMH2 计数器操作。\*<sup>‡</sup>

**注** 如果 TI52 引脚输入允许不控制, 此设置不需要。

<3> 当 TM52 和 CR52 (= FFH) 的值相等时, TM52 被清零为 00, 并且相等信号使 TM000 开始计数。然后, 当 TM000 和 CR000 相等时, TM00 被清为 0000H, 并且产生一个相等中断信号(INTTM000)。

如果控制 TI52 引脚输入允许, 通过读取 TM52, TI52 引脚在输入允许期间的外部事件计数可以测量, TM00 计数值, 和由 TMH2 中断请求信号 (INTTMH2) 经过中断服务的 TMIF52。

图 6-55. 外部 24 位事件计数器的操作时序

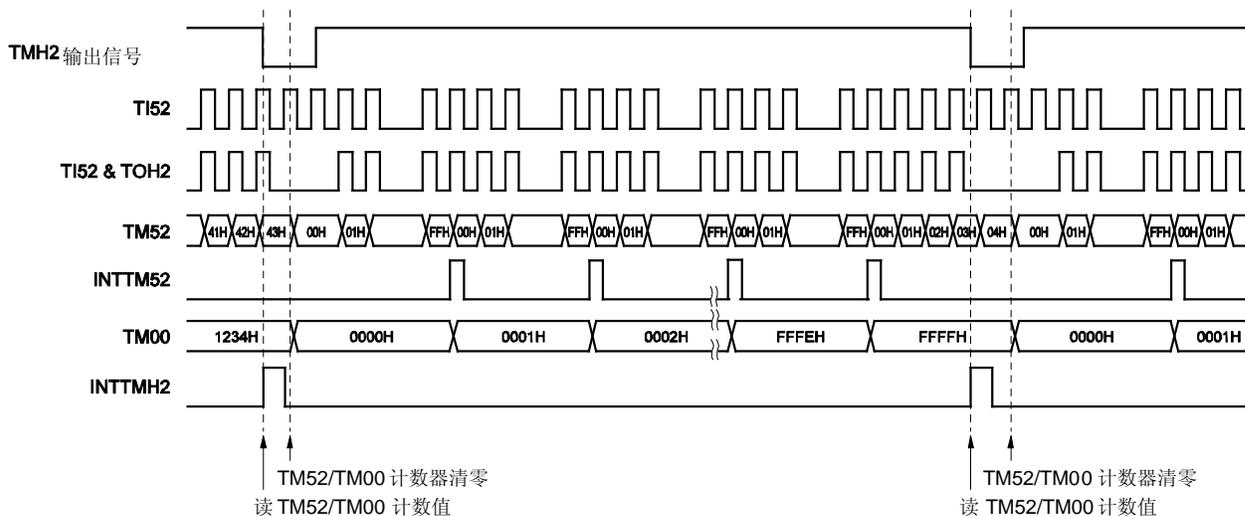
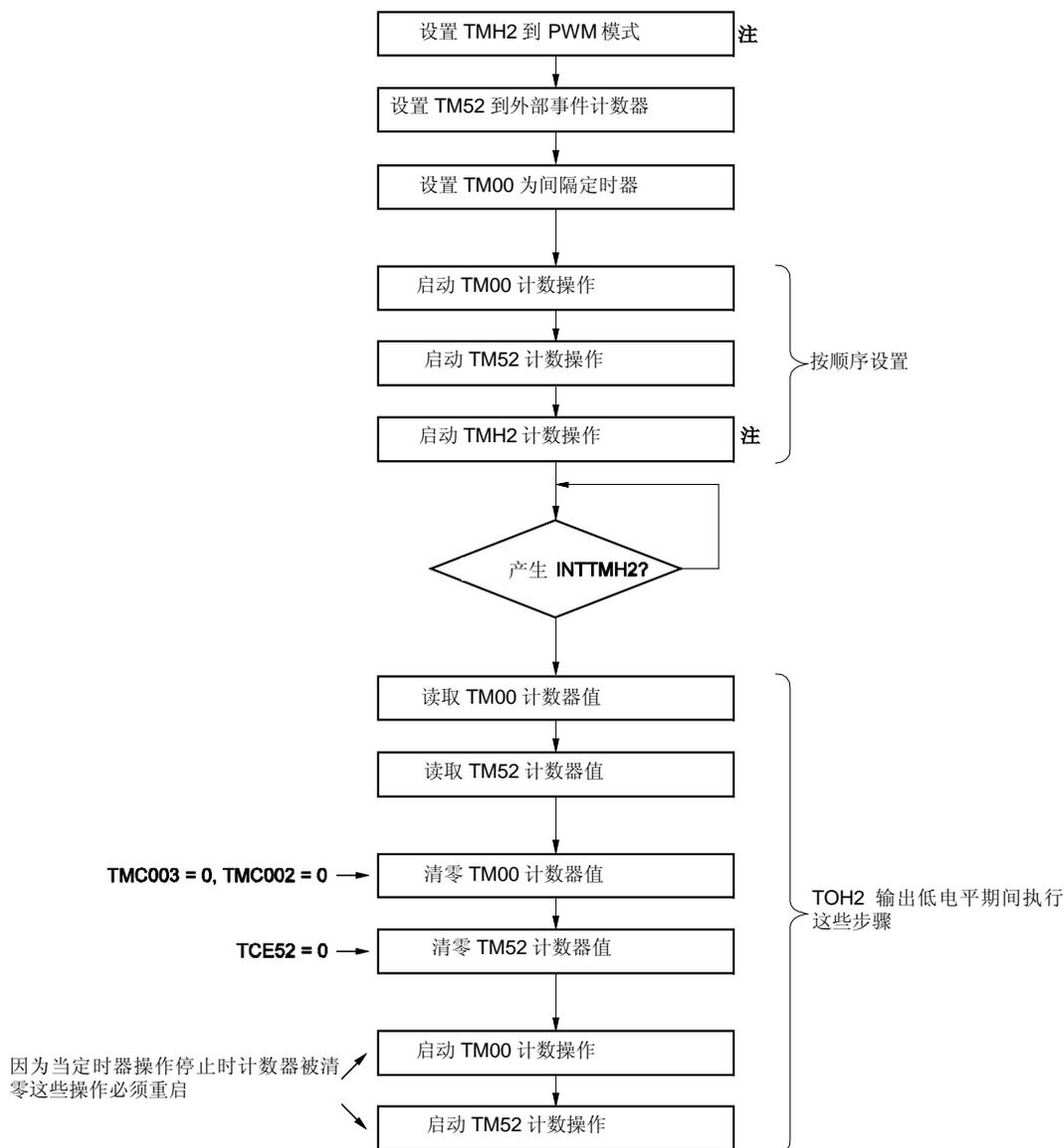


图 6-56. 外部 24 位事件计数器的操作流程



注 如果 TI52 引脚输入允许不控制，此设置不需要。

#### 6.4.10 外部 24 位事件计数器的注意事项

##### (1) 8 位定时器计数器 H2 输出信号

8 位 定时器 H2 的输出电平控制(默认值)，用于控制 TI52 引脚输入允许，必须设置为高电平 (TOLEV2 = 1)。因此，当 TI52 引脚输入允许信号禁止 (TMH2 输出: 低电平)时产生一个中断请求信号(INTTMH2)，并且 TM52 和 TM0 计数值(= 输入允许期间的外部事件计数值)可以由此中断服务读取。

TI52 引脚的输入允许信号的注意事项为高电平 (允许状态)，在 8 位 定时器 H2 操作通过设置(TOLEV2 = 1)已经允许(TMHE2 = 1)后，直到 TMH2 和 CMP02 寄存器值相等。

**(2) TI52 引脚输入允许控制的注意事项**

TI52 引脚的输入允许控制信号 (TMH2 输出信号) 与 TI52 引脚输入时钟同步, 在图 6-54 外部 24 位事件计数器的框图和图 6-55 外部 24 位事件计数器的操作时序中说明。因此, 当计数器作为外部事件计数器操作时, 计数可能会引起错误。

**(3) 外部 24 位事件计数器操作期间 16 位定时器/事件计数器 00 的注意事项**

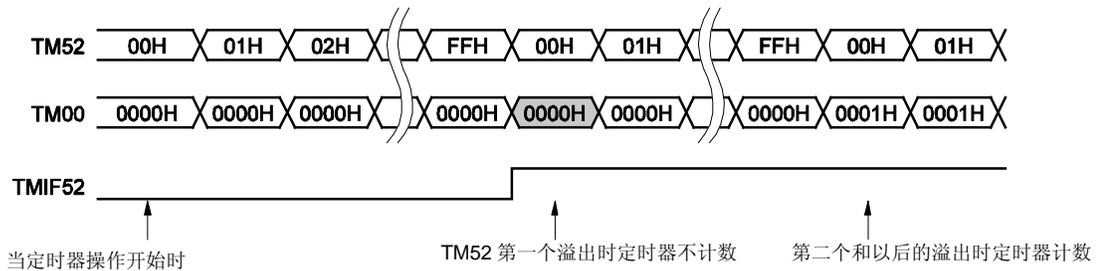
16 位定时器/事件计数器 00 当开始操作时与噪声消除器同步, 在操作开始后第一个时钟不立即计数。

当使用计数器作为 24 位计数器, 通过设置 16 位定时器/事件计数器 00 和 8 位定时器/事件计数器 52 作为高位和低位定时器并将它们级联, 作为低位定时器的 8 位定时器/事件计数器 52 的中断请求标志必须按照下面说明进行检测, 以便正确读出 24 位计数值。

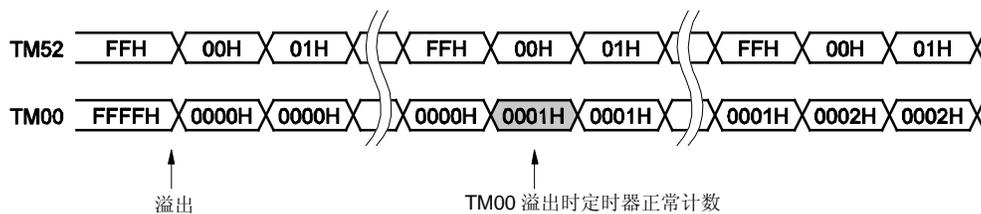
- TMIF52 = 1 时读取 TM52 和 TM00 :  
TM00 实际的计数值是“TM00 的读取值 + 1”。
- TMIF52 = 0 时读取 TM52 和 TM00:  
读取的值是正确的值。

16 位定时器/事件计数器 00 仅当操作开始出现这些情况。当 16 位定时器/事件计数器 00 溢出和计数从 0000H 重启时不发生计数延迟, 因为已经实现同步。

<开始操作时>



<高位定时器的溢出>



## 6.5 TM00 的特殊用途

### 6.5.1 TM00 操作期间重写 CR010

原则上，在 TM00 操作时(TMC003 与 TMC002 不等于 00)禁止重写 78K0/ LC3 的 CR000 与 CR010(当它们用作比较寄存器时)的值。

但如果 CR010 用于 PPG 输出且占空比已被修改(在 CR010 的值与 TM00 的值相等后修改 CR010 的值，如果 CR010 的值与 TM00 的值相等前立即修改 CR010，则可以会产生异常操作)，可以按下列过程修改 CR010，即使 TM00 正在操作。

#### 修改 CR010 值的过程

- <1> 禁止中断 INTTM010 (TMMK010 = 1)。
- <2> 当 TM00 的值与 CR010 的值相等时(TOC004 = 0)禁止反转定时器输出。
- <3> 修改 CR010 的值
- <4> 等待一个 TM00 计数时钟周期
- <5> 允许 TM00 和 CR010 (TOC004 = 1) 相等时进行定时器输出反转操作。
- <6> 将 INTTM010 的中断标志清零 (TMIF010 = 0)。
- <7> 允许中断 INTTM010 (TMMK010 = 0)。

**备注** 如需了解 TMIF010 和 TMMK010，可参见 第十七章 中断功能。

### 6.5.2 设置 LVS00 和 LVR00

#### (1) LVS00 和 LVR00 的使用

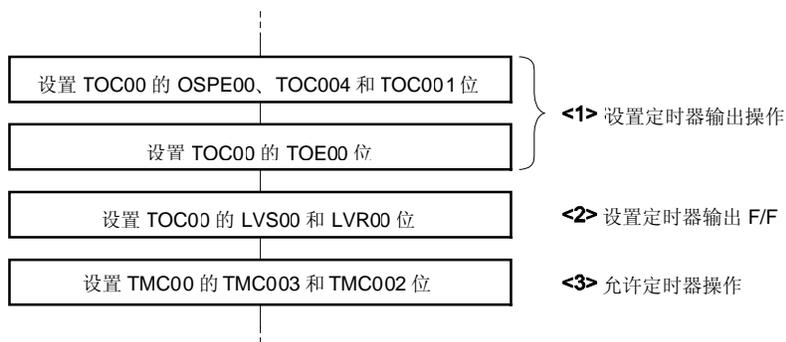
LVS00 与 LVR00 用于设置 TO00 引脚输出的默认值并可以在不需要定时器操作时(TMC003 与 TMC002 = 00)对定时器输出进行反转。当不需要软件控制时将 LVS00 与 LVR00 清零(00) (默认值: 低电平输出)。

LVS00	LVR00	定时器输出 状态
0	0	无变化(低电平输出)
0	1	清零(低电平输出)
1	0	设置(高电平输出)
1	1	设置禁止

**(2) LVS00 和 LVR00 的设置**

可按以下步骤设置 LVS00 和 LVR00。

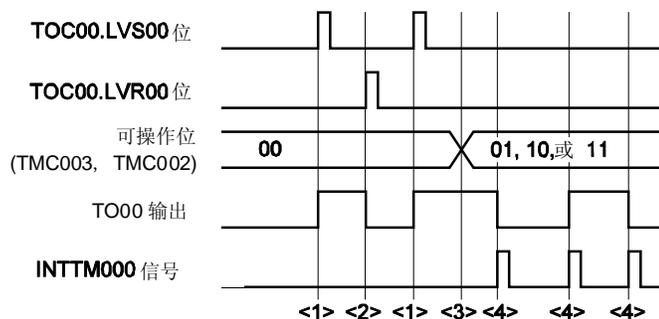
图 6-57. LVS00 和 LVR00 位设置流程示例



**注意事项** 必须按上述<1>、<2>和<3>步骤设置 LVS00 和 LVR00。

步骤<2>可以在<1>之后<3>之前执行。

图 6-58. LVR00 和 LVS00 时序示例



- <1> 当 LVS00 与 LVR00 = 10 时将 TO00 引脚输出变为高电平。
- <2> 当 LVS00 与 LVR00 = 01 时将 TO00 引脚输出变为低电平(即使 LVS00 与 LVR00 被清零(00)则引脚输出恒为高电平)。
- <3> 当 TMC003 与 TMC002=01、10 或 11 时定时器开始操作。由于在开始操作前 LVS00 与 LVR00 被设置为 10，所以 TO00 引脚输出从高电平开始。定时器开始操作后，禁止对 LVS00 与 LVR00 进行设置，直至 TMC003 与 TMC002 = 00 (禁止定时器操作)。
- <4> 每当产生中断信号(INTTM000)时，反转 TO00 引脚的输出电平。

## 6.6 16 位定时器/事件计数器 00 注意事项

## (1) 16 位定时器/事件计数器 00 每个通道的限制条件

表 6-3 显示了每个通道的限制条件。

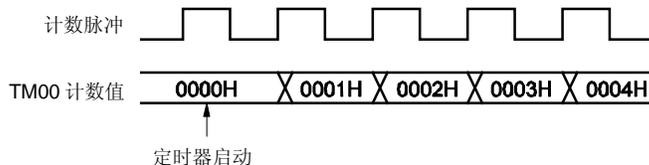
表 6-3. 16 位定时器/事件计数器 00 每个通道的限制条件

操作	限制条件
用作间隔定时器	-
用作方波输出	
用作外部事件计数器	
用于清零&启动模式(通过 TI000 引脚有效沿输入进入)	当使用检测 TI010 引脚有效沿时, 禁止使用定时器输出(TO00)。 TOC00 = 00H
用作自由运行定时器	-
用作 PPG 输出	$0000H \leq CR010 < CR000 \leq FFFFH$
用作单脉冲输出	不能给 CR000 和 CP010 设置相同的值。
用作脉冲宽度测量	禁止使用定时器输出 (TO00) (TOC00 = 00H)

## (2) 定时器启动误差

定时器启动后, 在产生相等信号之前可能发生将近一个时钟周期的误差。这是由于启动 TM00 计数操作与计数脉冲不同步引起的。

图 6-59. TM00 计数启动时序



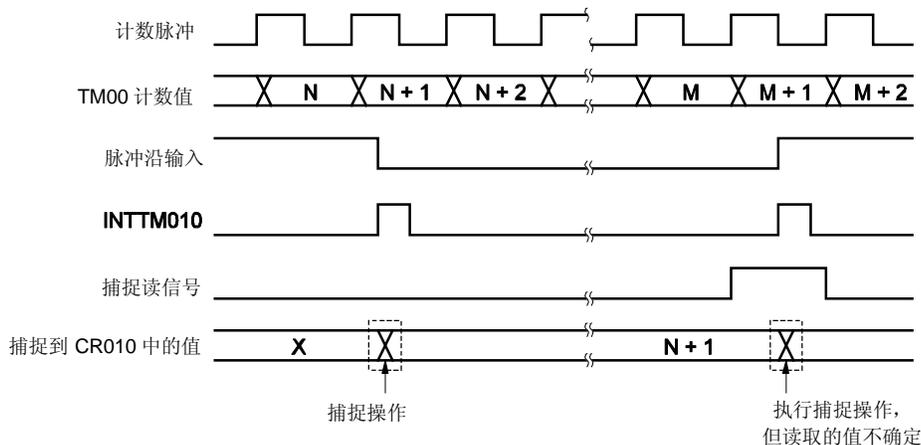
## (3) CR000 与 CR010 的设置 (清零&amp;启动模式(TM00 与 CR000 相等时进入))

为 CR000 与 CR010 设置 0000H 以外的值 (当 TM00 外部事件计数器时不能进行单脉冲计数)。

(4) 捕捉寄存器数据保持时序

- (a) 如果在读取 CR000/CR010 时，有效沿输入到 TI000/TI010 引脚并且检测到 TI000 引脚脉冲的反向沿时，CR010 执行捕捉操作但 CR000/CR010 的读取值不确定。此时当检测到 TI000/TI010 引脚的有效沿时产生中断信号(INTTM000/INTTM010)(当检测到 TI000 引脚的反向脉冲沿时不产生中断信号)。如果由于检测到 TI000/TI010 引脚的有效沿而捕捉到计数值时，则应在产生 INTTM000/INTTM010 信号后读取 CR000/CR010 的值。

图 6-60. 捕捉寄存器数据保持时序



- (b) 当 16 位定时器/事件计数器 00 停止操作后，CR000 与 CR010 的值不确定。

(5) 设置有效沿

当定时器停止操作时(TMC003 与 TMC002 = 00)设置 TI000 引脚的有效沿。使用 ES000 与 ES001 设置有效沿。

(6) 单脉冲的重新触发

必须确定在单脉冲输出模式下正在输出一个活动电平时，没有触发产生。而且必须在输出当前活动电平后输入下一个触发。

**(7) OVF00 标志的操作****(a) 设置 OVF00 标志 (1)**

在以下情况下以及 TM00 溢出时将 OVF00 标志设置为 1。

选择清零&启动模式(在 TM00 与 CR000 相等时进入)。

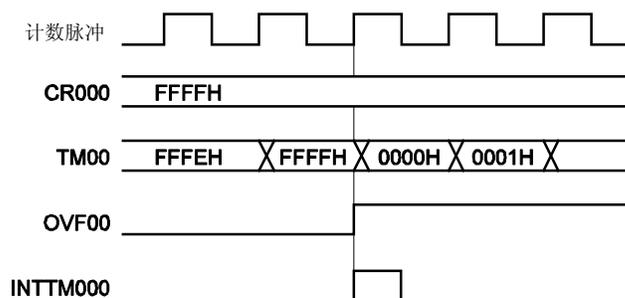
↓

设置 CR000 为 FFFFH。

↓

当 TM00 与 CR000 相等且 TM00 从 FFFFH 变为 0000H 时 TM00 被清零。

图 6-61. OVF00 标志操作时序

**(b) OVF00 标志清零**

即使在 TM00 溢出后下一个计数时钟被计数之前(TM00 的值变为 0001H 之前)将 OVF00 标志清零(0)，OVF00 也会再次被设置为 1，并且对其清零是无效的。

**(8) 单脉冲输出**

在自由运行定时器模式下或者在清零&启动模式(通过 TI000 引脚有效沿进入)下，单脉冲输出能正确操作。而在清零&启动模式(在 TM00 与 CR000 相等时进入)下不能输出单脉冲。

**(9) 捕捉操作****(a) 当指定 TI000 的有效沿作为计数时钟时**

当指定 TI000 的有效沿作为计数时钟时，捕捉寄存器(指定 TI000 作为触发)不能正确操作。

**(b) 通过 TI010 与 TI000 引脚输入信号准确捕捉计数值的脉冲宽度**

为了准确捕捉计数值，作为捕捉触发的 TI000 与 TI010 引脚的输入脉冲宽度必须大于两个计数时钟(由 PRM00 选择)(参见图 6-7)。

**(c) 中断信号的产生**

在计数时钟的下降沿进行捕捉操作，但中断信号(INTTM000 与 INTTM010)则在下一个计数时钟的上升沿产生(参见图 6-7)。

**(d) 当 CRC001 (捕捉/比较控制寄存器 00(CRC00)的第 1 位)=1 时的注释**

在 TI000 引脚输入信号的反向沿将 TM00 的计数值捕捉到 CR000 时，捕捉到计数值后不产生中断信号(INTTM000)。如果在此操作期间，从 TI010 引脚检测到有效沿，则不执行捕捉操作但会产生中断信号(INTTM000)作为外部中断信号。当不使用外部中断时屏蔽 INTTM000 信号。

**(10) 脉冲沿检测****(a) 复位后指定有效沿**

如果复位后 TI000 或 TI010 引脚处于高电平，且当指定上升沿或兼有两种脉冲沿作为 TI000 或 TI010 引脚的有效沿时，允许 16 位定时器/事件计数器 00 的操作，则检测到 TI000 或 TI010 引脚的高电平作为上升沿。上拉 TI000 或 TI010 引脚时要特别注意。但如果操作停止之后又允许操作，则不检测上升沿。

**(b) 用于消除噪音的采样时钟**

根据 TI000 的有效沿是用作计数时钟还是用作捕捉触发，采样时钟(用于消除噪音)会发生变化。在前一种情况中采样时钟恒为  $f_{PRS}$ 。而在后一种情况中则将由 PRM00 选择的计数时钟用于采样。

当采样到 TI000 引脚的输入信号并且在一个周期内两次检测到有效电平时，才认为检测到有效沿。因此可以消除窄脉冲宽度的噪音(参见图 6-7)。

**(11) 定时器操作**

无论 CPU 处于何种操作模式，当定时器停止操作时，不响应 TI000/TI010 引脚的输入信号。

**备注**  $f_{PRS}$ : 外围硬件时钟频率

## 第七章 8 位定时器/事件计数器 50、51 和 52

### 7.1 8 位定时器/事件计数器 50、51 和 52 的功能

8 位定时器/事件计数器 50、51 和 52 具有以下功能

- 间隔定时器
- 外部事件计数器<sup>注</sup>

**注** 仅对 **TM52**。TM52 和 TM00 可以组合在一起用作外部 24 位事件计数器。同样，经 TMH2 控制 TM52 外部事件输入。详细内容，参见第六章 16 位定时器/事件计数器 00。

### 7.2 8 位定时器/事件计数器 50、51 和 52 的配置

8 位定时器/事件计数器 50、51 和 52 包括以下硬件。

表 7-1. 8 位定时器/事件计数器 50、51 和 52 的配置

项目	配置
定时器寄存器	8 位定时器计数器 5n (TM5n)
寄存器	8 位定时器比较寄存器 5n (CR5n)
定时器输入	TI5n
控制寄存器	定时器时钟选择寄存器 5n (TCL5n) 8 位定时器模式控制寄存器 5n (TMC5n) 输入转换控制寄存器 (ISC) 端口模式寄存器 3 (PM3) 端口寄存器 3 (P3)

**备注** n = 0 ~ 2

图 7-1 ~ 7-3 显示了 8 位定时器/事件计数器 50、51 和 52 的框图。

图 7-1. 8 位定时器/事件计数器 50 的框图

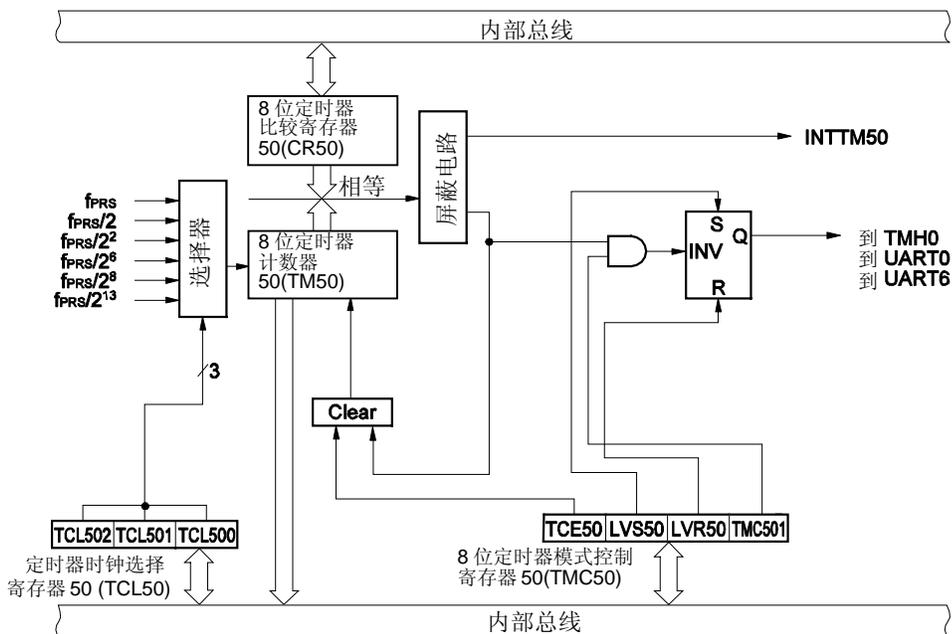


图 7-2. 8 位定时器/事件计数器 51 的框图

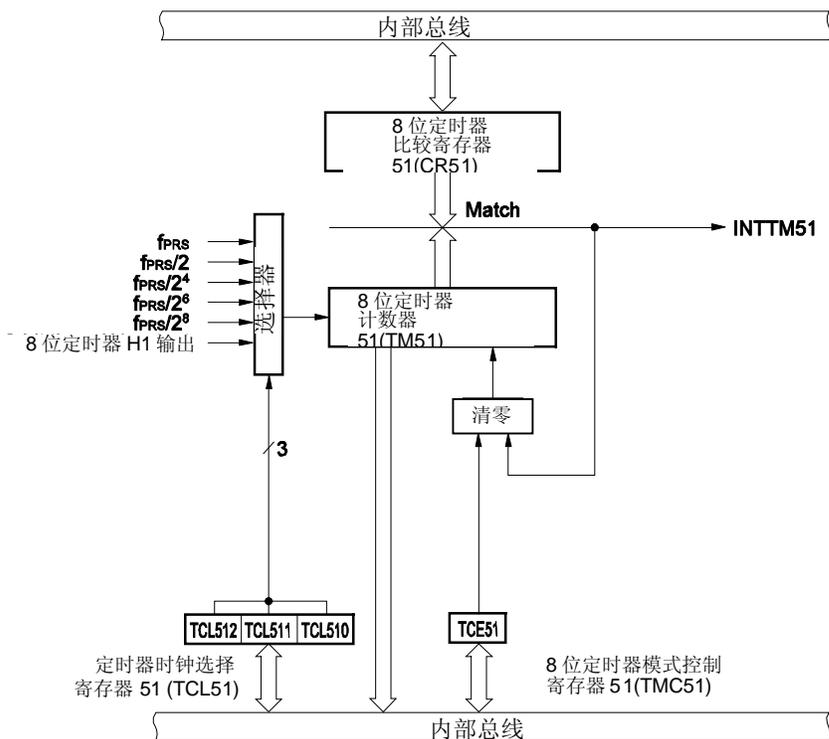
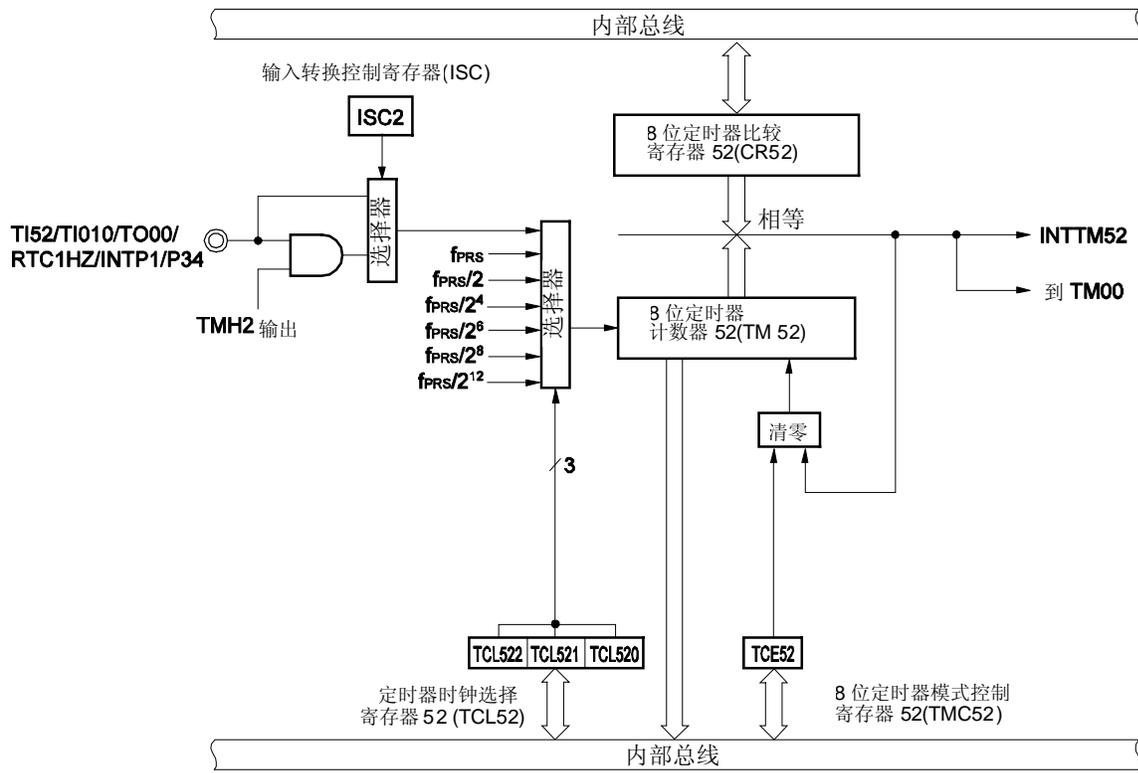


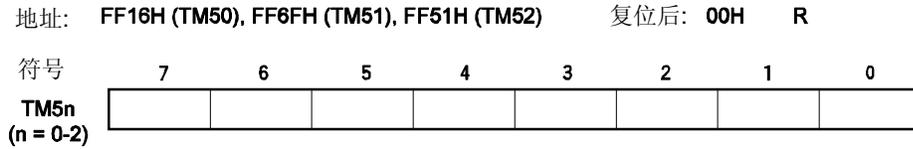
图 7-3. 8 位定时器/事件计数器 52 的框图



**(1) 8 位定时器计数器 5n (TM5n)**

TM5n 是一个 8 位寄存器，用于对计数脉冲进行计数，它是只读的。  
该计数器的计数值随计数时钟的上升沿同步增加。

图 7-4. 8 位定时器计数器 5n (TM5n)的格式



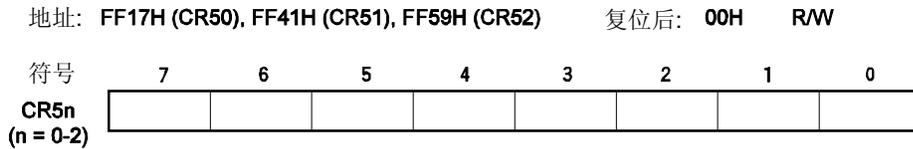
在以下几种情况下将计数值清零(00H)。

- <1> 产生复位信号
- <2> TCE5n 清零
- <3> 当 TM5n 与 CR5n 相等时。

**(2) 8 位定时器比较寄存器 5n (CR5n)**

可以由 8 位存储器操作指令对 CR5n 进行读写  
CR5n 的值将与 8 位定时器计数器 5n(TM5n)的计数值频繁相比较，如果相等将产生中断请求(INTTM5n)。  
CR5n 的取值范围为：00H ~ FFH。  
复位信号的产生将 CR5n 清零(00H)。

图 7-5. 8 位定时器比较寄存器 5n (CR5n)的格式



**注意事项** 操作期间，不要将其它值写入 CR5n。

**备注**      n = 0~2

### 7.3 控制 8 位定时器/事件计数器 50、51 和 52 的寄存器

以下五种寄存器用于控制 8 位定时器/事件计数器 50、51 和 52。

- 定时器时钟选择寄存器 5n (TCL5n)
- 8 位定时器模式控制寄存器 5n (TMC5n)
- 输入转换控制寄存器 (ISC)
- 端口模式寄存器 3 (PM3)
- 端口寄存器 3 (P3)

#### (1) 定时器时钟选择寄存器 5n (TCL5n)

该寄存器设置 8 位定时器/事件计数器 5n 的计数时钟，和 TI5n 引脚输入的有效沿。

可由 1 位或 8 位存储器操作指令设置 TCL5n。

复位信号的产生将 TCL5n 清零(00H)。

备注 n = 0~2

图 7-6. 定时器时钟选择寄存器 50 (TCL50) 的格式

地址: FF6AH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500

TCL502	TCL501	TCL500	计数时钟选择 <sup>※1</sup>			
			f <sub>PRS</sub> = 2 MHz	f <sub>PRS</sub> = 5 MHz	f <sub>PRS</sub> = 10 MHz	
0	0	0	禁止设置			
0	0	1	禁止设置			
0	1	0	F <sub>PRS</sub> <sup>※2</sup>	2 MHz	5 MHz	10 MHz
0	1	1	f <sub>PRS</sub> /2	1 MHz	2.5 MHz	5 MHz
1	0	0	f <sub>PRS</sub> /2 <sup>2</sup>	500 kHz	1.25 MHz	2.5 MHz
1	0	1	f <sub>PRS</sub> /2 <sup>6</sup>	31.25 kHz	78.13 kHz	156.25 kHz
1	1	0	f <sub>PRS</sub> /2 <sup>8</sup>	7.81 kHz	19.53 kHz	39.06 kHz
1	1	1	f <sub>PRS</sub> /2 <sup>13</sup>	0.24 kHz	0.61 kHz	1.22 kHz

注 1. 如果外围硬件时钟(f<sub>PRS</sub>) 操作在高速系统时钟(f<sub>XH</sub>) (XSEL = 1)下， f<sub>PRS</sub> 操作频率依据供电电压不同而变动。

- V<sub>DD</sub> = 2.7 ~ 5.5 V: f<sub>PRS</sub> ≤ 10 MHz
- V<sub>DD</sub> = 1.8 ~ 2.7 V: f<sub>PRS</sub> ≤ 5 MHz

2. 如果外围硬件时钟(f<sub>PRS</sub>) 操作在内部高速振荡时钟(f<sub>XH</sub>) (XSEL = 0)下，当 1.8 V ≤ V<sub>DD</sub> < 2.7 V 时，禁止设置 TCL502, TCL501, TCL500 = 0, 1, 0 (计数时钟: f<sub>PRS</sub>)。

注意事项 1. 如果要向 TCL50 写入其它值，则必须提前停止定时器操作。

2. 必须将第 3 位 ~ 第 7 位清零。

备注 f<sub>PRS</sub>: 外围硬件时钟频率

图 7-7. 定时器时钟选择寄存器 51 (TCL51) 的格式

地址: FF8CH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	计数时钟选择 <sup>注1</sup>			
			f <sub>PRS</sub> = 2 MHz	f <sub>PRS</sub> = 5 MHz	f <sub>PRS</sub> = 10 MHz	
0	0	0	禁止设置			
0	0	1	禁止设置			
0	1	0	F <sub>PRS</sub> <sup>注2</sup>	2 MHz	5 MHz	10 MHz
0	1	1	f <sub>PRS</sub> /2	1 MHz	2.5 MHz	5 MHz
1	0	0	f <sub>PRS</sub> /2 <sup>4</sup>	125 kHz	312.5 kHz	625 kHz
1	0	1	f <sub>PRS</sub> /2 <sup>6</sup>	31.25 kHz	78.13 kHz	156.25 kHz
1	1	0	f <sub>PRS</sub> /2 <sup>8</sup>	7.81 kHz	19.53 kHz	39.06 kHz
1	1	1	定时器 H1 输出信号			

- 注
1. 如果外围硬件时钟 (f<sub>PRS</sub>) 操作在高速系统时钟(f<sub>XH</sub>) (XSEL = 1)下, f<sub>PRS</sub> 操作频率依据供电电压不同而变动。
    - V<sub>DD</sub> = 2.7 ~ 5.5 V: f<sub>PRS</sub> ≤ 10 MHz
    - V<sub>DD</sub> = 1.8 ~ 2.7 V: f<sub>PRS</sub> ≤ 5 MHz
  2. 如果外围硬件时钟(f<sub>PRS</sub>) 操作在内部高速振荡时钟(f<sub>XH</sub>) (XSEL = 0)下, 当 1.8 V ≤ V<sub>DD</sub> < 2.7 V 时, 禁止设置 TCL512, TCL511, TCL510 = 0, 1, 0 (计数时钟: f<sub>PRS</sub>)。

- 注意事项
1. 如果要向 TCL51 写入其它值, 则必须提前停止定时器操作。
  2. 必须将第 3 位 ~ 第 7 位清零。

图 7-8. 定时器时钟选择寄存器 52 (TCL52) 的格式

地址: FF5BH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TCL52	0	0	0	0	0	TCL522	TCL521	TCL520

TCL522	TCL521	TCL520	计数时钟选择 <sup>#1</sup>			
			f <sub>PRS</sub> = 2 MHz	f <sub>PRS</sub> = 5 MHz	f <sub>PRS</sub> = 10 MHz	
0	0	0	由 ISC2 选择时钟下降沿			
0	0	1	由 ISC2 选择时钟上升沿			
0	1	0	F <sub>PRS</sub> <sup>#2</sup>	2 MHz	5 MHz	10 MHz
0	1	1	f <sub>PRS</sub> /2	1 MHz	2.5 MHz	5 MHz
1	0	0	f <sub>PRS</sub> /2 <sup>4</sup>	125 kHz	312.5 kHz	625 kHz
1	0	1	f <sub>PRS</sub> /2 <sup>6</sup>	31.25 kHz	78.13 kHz	156.25 kHz
1	1	0	f <sub>PRS</sub> /2 <sup>8</sup>	7.81 kHz	19.53 kHz	39.06 kHz
1	1	1	f <sub>PRS</sub> /2 <sup>12</sup>	0.49 kHz	1.22 kHz	2.44 kHz

- 注
1. 如果外围硬件时钟 (f<sub>PRS</sub>) 操作在高速系统时钟(f<sub>XH</sub>) (XSEL = 1)下, f<sub>PRS</sub> 操作频率依据供电电压不同而变动。
    - V<sub>DD</sub> = 2.7 ~ 5.5 V: f<sub>PRS</sub> ≤ 10 MHz
    - V<sub>DD</sub> = 1.8 ~ 2.7 V: f<sub>PRS</sub> ≤ 5 MHz
  2. 如果外围硬件时钟(f<sub>PRS</sub>) 操作在内部高速振荡时钟(f<sub>XH</sub>) (XSEL = 0)下, 当 1.8 V ≤ V<sub>DD</sub> < 2.7 V 时, 禁止设置 TCL522, TCL521, TCL520 = 0, 1, 0 (计数时钟: f<sub>PRS</sub>)。

- 注意事项
1. 如果要向 TCL52 写入其它值, 则必须提前停止定时器操作。
  2. 必须将第 3 位 ~ 第 7 位清零。

备注 f<sub>PRS</sub>: 外围硬件时钟频率

**(2) 8位定时器模式控制寄存器 5n (TMC5n)**

此寄存器控制 8-位定时计数器 5n(TM5n)的计数操作。

可以由 1 位或 8 位存储器操作指令设置 TMC5n。

复位信号的产生将该寄存器清零(00H)。

备注 n = 0 ~ 2

图 7-9. 8 位定时器模式控制寄存器 50 (TMC50)的格式

地址: FF6BH 复位后: 00H R/W <sup>#</sup>

符号	<7>	6	5	4	<3>	<2>	1	0
TMC50	TCE50	0	0	0	LVS50	LVR50	TMC501	0

TCE50	TM50 计数操作控制
0	清零后, 禁止计数操作 (计数器停止操作)
1	开始计数

LVS50	LVR50	定时器输出 F/F 状态设置
0	0	无变化
0	1	定时器输出 F/F 复位(0) (TO50 默认输出: 低电平)
1	0	定时器输出 F/F 设置(1) (TO50 默认输出: 高电平)
1	1	禁止设置

TMC501	定时器 F/F 控制
0	禁止翻转操作
1	允许翻转操作

注 第 2 位和第 3 位为只写操作位。

注意事项 1. 一定要把位 0, 和 4 ~ 6 清零 0。

2. 按照下列顺序执行<1> ~ <3>, 不要同时进行。

- <1> 设置 TMC501: 操作模式设置
- <2> 设置 LVS50, LVR50: 定时器 F/F 设置
- <4> 设置 TCE50

备注 如果读取 LVS50 和 LVR50, 数值为 0。

图 7-10. 8 位定时器模式控制寄存器 51 (TMC51) 的格式

地址: FF43H 复位后: 00H R/W\*

符号	<7>	6	5	4	3	2	1	0
TMC51	TCE51	0	0	0	0	0	0	0

TCE51	TM51 计数操作控制
0	清零后禁止计数操作 (计数器停止操作)
1	开始计数

注意事项 必须将位 0 ~ 6 置为 0。

图 7-11. 8 位定时器模式控制寄存器 52 (TMC52)的格式

地址: FF5CH 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	0
TMC52	TCE52	0	0	0	0	0	0	0

TCE52	TM52 计数操作控制
0	清零后禁止计数操作 (计数器停止操作)
1	开始计数

注意事项 必须将位 0 ~ 6 置为 0。

**(3) 输入转换控制寄存器(ISC)**

将 ISC2 置为 1，可经 TOH2 输出信号控制 TI52 输入信号。

可以由 1 位或 8 位存储器操作指令设置。

复位信号的产生将该寄存器清零(00H)。

图 7-12. 输入转换控制寄存器(ISC)的格式

地址: FF4FH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ISC	0	0	ISC5	ISC4	ISC3	ISC2	ISC1	ISC0

ISC5	ISC4	选择 TxD6, RxD6 输入源
0	0	TxD6:P112, RxD6: P113
1	0	TxD6:P13, RxD6: P12
其他情况		禁止设置

ISC3	RxD6/P113 输入允许/禁止
0	禁止 RxD6/P113 输入
1	允许 RxD6/P113 输入

ISC2	控制 TI52 输入源
0	不允许控制 TI52 输入 (P34)
1	允许控制 TI52 输入 (P34) <sup>注1</sup>

ISC1	选择 TI000 输入源
0	TI000 (P33)
1	RxD6 (P12 或 P113) <sup>注2</sup>

ISC0	选择 INTPO 输入源
0	INTPO (P120)
1	RxD6 (P12 或 P113) <sup>注2</sup>

- 注 1. 由 TOH2 输出信号控制 TI52 输入。  
 2. 由 ISC5 和 ISC4 选择 P12 或 P113 。

**(4) 端口模式寄存器 3 (PM3)**

此寄存器可以按位设置端口 3 的输入/输出操作模式。

当 P34/TI52/TI010/TO00/RTC1HZ/INTP1 引脚用作定时器输出时, 将 PM34 置为 1。PM34 此时的输出锁存可以为 0 也可以为 1。

可以由 1 位或 8 位存储器操作指令设置 PM3。

复位信号的产生将寄存器的内容设置为 FFH。

**图 7-13. 端口模式寄存器 3 (PM3) 的格式**

地址: FF23H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34	PM33	PM32	PM31	1

PM3n	P1n 引脚 I/O 模式选择(n = 1~4)
0	输出模式(输出缓冲器打开)
1	输入缓冲器(输出缓冲器关闭)

## 7.4 8位定时器/事件计数器 50、51 和 52 的操作

### 7.4.1 用作间隔定时器

8位定时器/事件计数器 5n 用作间隔定时器，可以在间隔时间内不断产生中断请求，该间隔时间由 8 位定时器比较寄存器 5n(CR5n)的预置值决定。

当 8 位定时器计数器 5n(TM5n)中的计数值与 CR5n 中的值相等时，将 TM5n 清零并继续计数，同时产生中断请求信号(INTTM5n)。

根据定时器时钟选择寄存器 5n(TCL5n)的第 0 位~第 2 位(TCL5n0 ~ TCL5n2)选择 TM5n 的计数时钟。

#### 设置

<1> 设置寄存器

- TCL5n: 设置计数时钟
- CR5n: 比较值
- TMC5n: 停止计数操作

(TMC50 = 0000xxx0B, TMC51 = TMC52 = 00000000B × =不必考虑)

<2> 将 TCE5n 置 1 后，开始计数。

<3> 如果 TM5n 与 CR5n 值相等时，产生中断 INTTM5n (将 TM5n 清零(00H))。

<4> 以相同的时间间隔重复产生 INTTM5n。

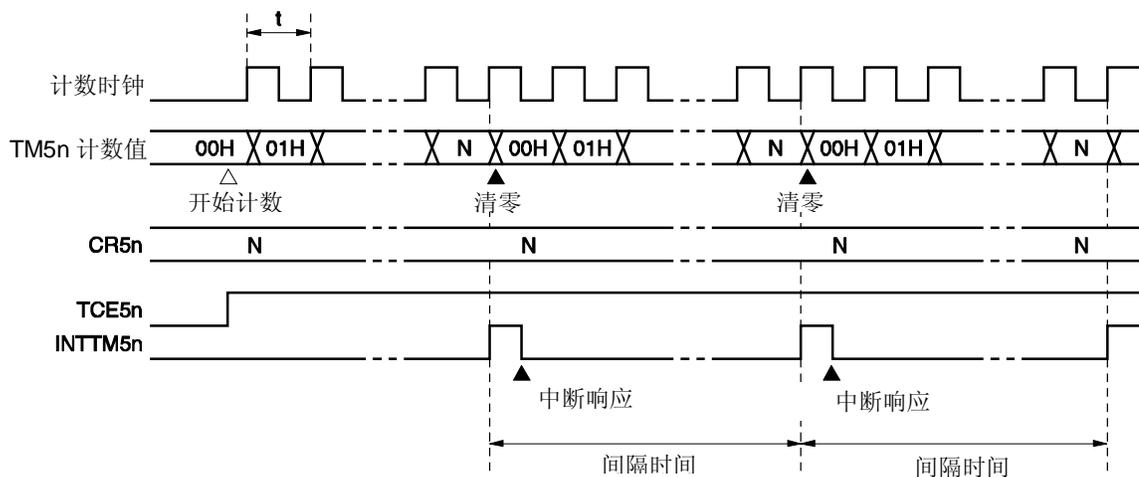
将 TCE5n 清零，停止计数操作。

**注意事项** 操作期间不要向 CR5n 写入其它值。

- 备注**
1. 如需了解如何允许 INTTM5n 中断，请参见 第十七章 中断功能。
  2. n = 0~2

图 7-14. 间隔定时器操作的时序(1/2)

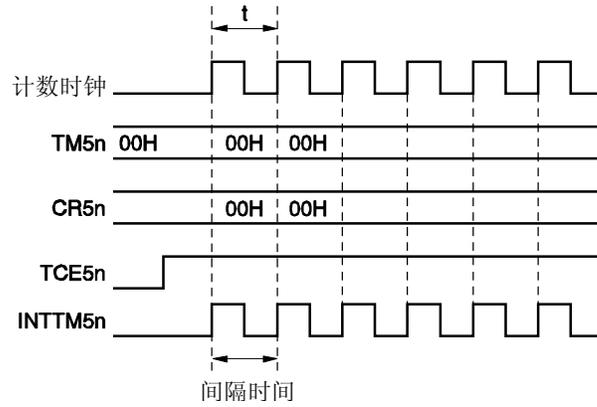
(a) 基本操作



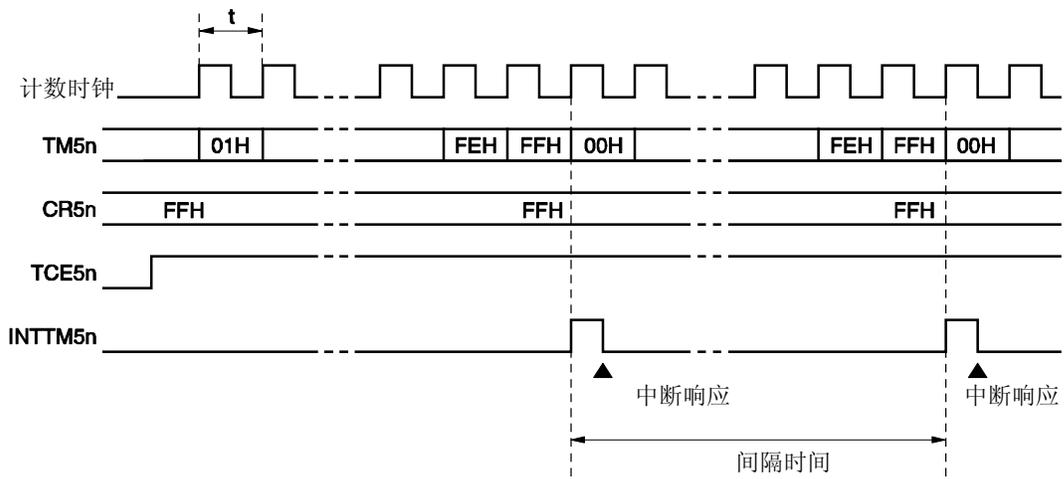
- 备注**
- 间隔时间 =  $(N + 1) \times t$
  - N = 01H ~ FFH
  - n = 0 ~ 2

图 7-14. 间隔定时器操作的时序(2/2)

(b) 当 CR5n = 00H



(c) 当 CR5n = FFH



备注 n = 0 ~ 2

### 7.4.2 用作外部事件计数器的操作(仅对 TM52)

外部事件计数器通过 8 位定时器计数器 52 (TM52) 对输入到 TI52 引脚的外部时钟脉冲计数。

TM52 随输入的有效沿次数递增。有效沿由定时器时钟选择寄存器 52 (TCL52) 规定, 可以选择上升沿或下降沿。

当 TM52 中的计数值与 8 位定时器比较寄存器 52(CR52)中的值相等时, 将 TM52 清零, 并产生中断请求信号 (INTTM52)。

一旦 TM52 中的值与 CR52 的值相等, 将产生 INTTM52 中断请求。

#### 设置

##### <1> 设置寄存器

- 设置端口模式寄存器(PM34)的值为 1。
- TCL52: 选择 TI52 引脚输入脉冲沿  
TI52 引脚下降沿 → TCL52 = 00H  
TI52 引脚上升沿 → TCL52= 01H
- CR52: 比较值
- TMC52:停止计数, 选择 TM5n 与 CR52 相等时清零和启动模式, 禁止定时器 F/F 反转操作, 禁止定时器输出。  
(TMC52= 0000x×00B x =不必考虑)

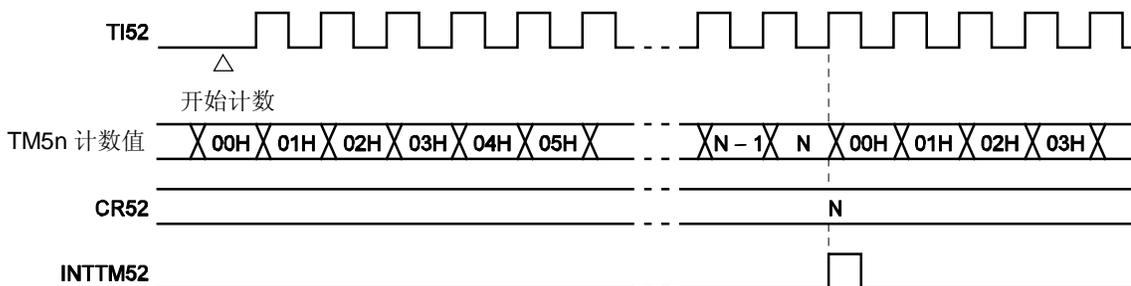
<2> 当 TCE52= 1 时, 开始对从 TI5n 引脚输入的脉冲计数。

<3> 当 TM52 与 CR52 的值相等时, 将产生 INTTM52(TM52 清零(00H))。

<4> 在经过以上设置后, 每当 TM52 与 CR52 的值相等时都会产生 INTTM52 中断请求。

**备注** 如需了解如何允许 INTTM52 中断, 请参见 第十七章 中断功能。

图 7-15. 外部事件计数器操作的时序 (规定上升沿有效)



- 备注**
1. 8 位定时器/事件计数器 52 (TM52) 和 16 位定时器/事件计数器(TM00)组合, 可以用作 24 位定时器/事件计数器。TM52 输入允许由 TMH2 控制。详细内容, 参见 6.4.9 外部 24 位事件计数器操作。
  2. N = 00H ~ FFH

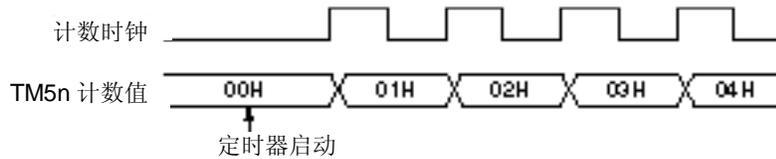
## 7.5 使用 8 位定时器/事件计数器 50、51 和 52 的注意事项

### (1) 定时器启动误差

定时器启动后，在产生相等信号之前可能发生将近一个时钟周期的误差。

这是由于 8 位定时器计数器 50、51 和 52(TM50, TM51 和 TM52)的启动与计数时钟不同步引起的。

图 7-16. 8 位定时器计数器 5n 启动时序



备注  $n = 0 \sim 2$

### (2) 与 16 位定时器/事件计数器 00 一起作为外部 24 位事件计数器操作时的注意事项

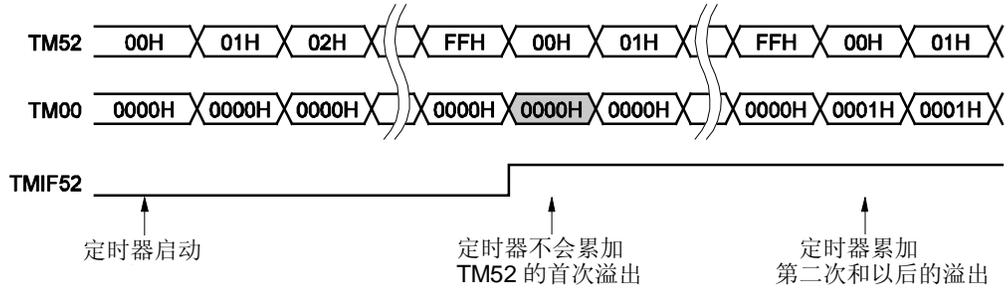
当启动操作时，16 位定时器/事件计数器 00 有内部同步噪声消除电路，并且操作开始后最初的立即时钟将不被计数。

当使用 24 位计数器时，设置 16 定时器/事件计数器 00 和 8 定时器/事件计数器 52，分别作为高位和低位定时器，为了精确读出 24 位计数值，必须检测低位定时器 8 定时器/事件计数器 52 的中断请求信号，描述如下。

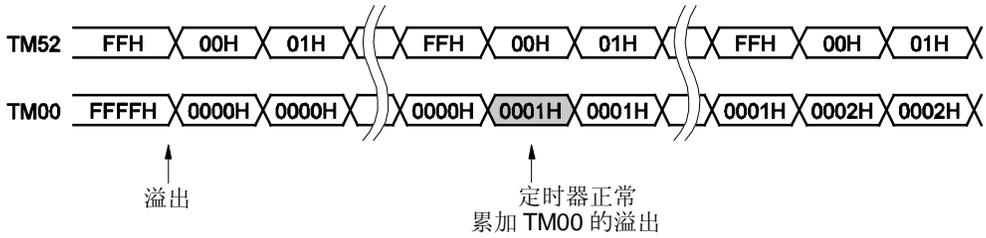
- 如果  $TMIF52 = 1$ ，当读出 TM52 和 TM00:  
TM00 实际计数值为“TM00 + 1 读出值”。
- 如果  $TMIF52 = 0$ ，当读出 TM52 和 TM00:  
读出值为修正值。

当操作启动时，仅发生 16 定时器/事件计数器 00。当 16 定时器/事件计数器 00 溢出并且计数从 0000H 重新开始，因为已经执行同步，不会发生计数延迟。

<当操作启动时>



<高位定时器溢出>



## 第八章 8 位定时器 H0、H1 和 H2

### 8.1 8 位定时器 H0、H1 和 H2 的功能

8 位定时器 H0、H1 和 H2 的功能如下。

- 间隔定时器
- 方波输出<sup>注1</sup>
- PWM 输出模式<sup>注2</sup>
- 载波发生器模式 (仅用于 8 位定时器 H1)<sup>注3</sup>

- 注
1. 仅适用于 TMH0 和 TMH1。
  2. TOHn 仅适用于 TOH0 和 TOH1。
  3. 仅适用于 TMH1。TM51 和 TMH1 可组合用作载波发生器模式。

### 8.2 8 位定时器 H0、H1 和 H2 的配置

8 位定时器 H0、H1 和 H2 包括以下硬件。

表 8-1. 8 位定时器 H0、H1 和 H2 的配置

项目	配置
定时器寄存器	8 位定时器计数 Hn
寄存器	8 位定时器 H 比较寄存器 0n (CMP0n) 8 位定时器 H 比较寄存器 1n (CMP1n)
定时器输出	TOHn <sup>注1</sup> , 输出控制器
控制寄存器	8 位定时器 H 模式寄存器 n (TMHMDn) 8 位定时器 H 载波控制寄存器 1 (TMCYC1) <sup>注2</sup> 端口模式寄存器 3 (PM3) 端口寄存器 3 (P3)

- 注
1. TMH2 没有输出引脚 (TOH2)。TI52 仅可用作内部中断(INTTMH2)或外部事件输入允许信号。
  2. 仅用于 8 位定时器 H1

备注 n = 0-2。TOHn 仅适用于 TOH0 和 TOH1。

图 8-1 ~ 8-3 为定时器框图。

图 8-1. 8 位定时器 H0 的框图

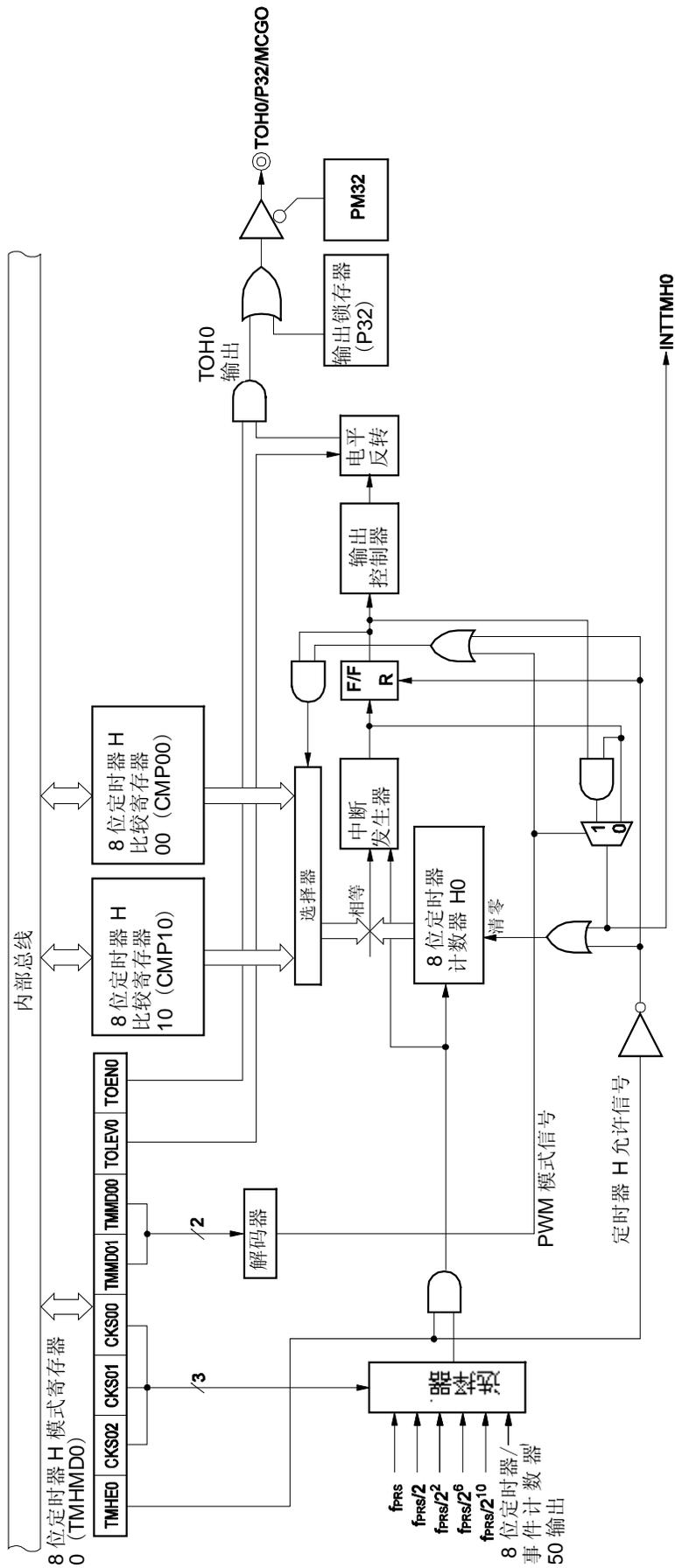


图 8-2. 8 位定时器 H1 的框图

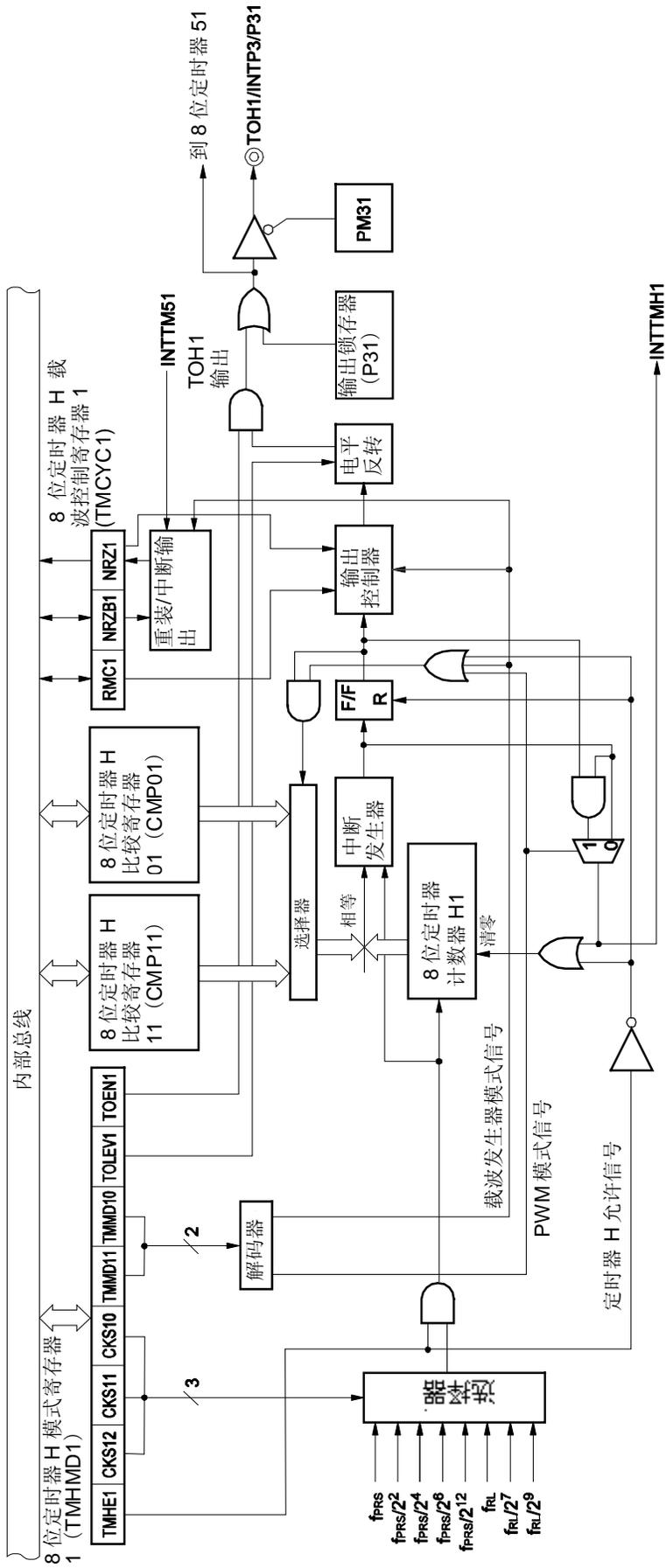
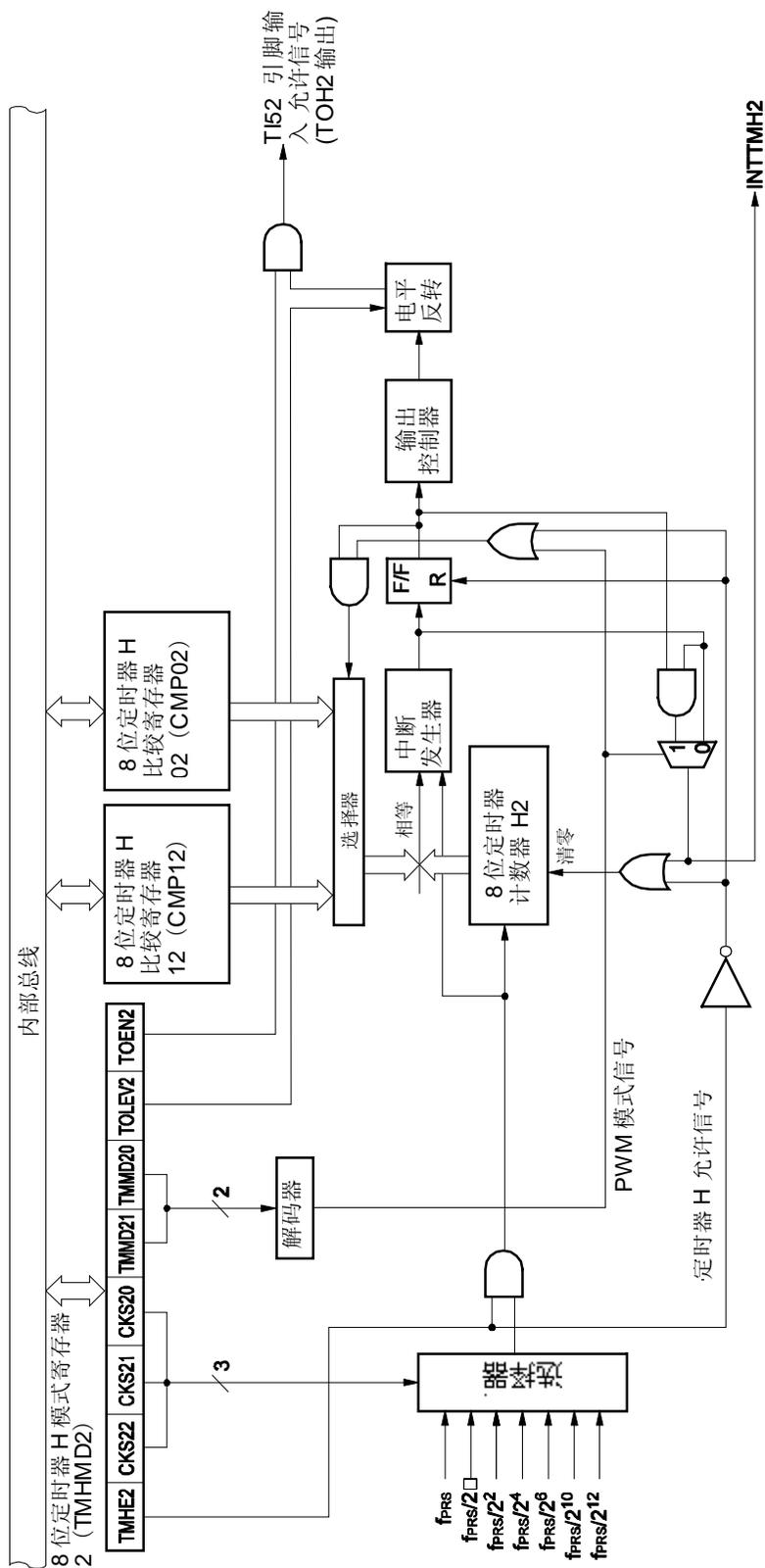


图 8-3. 8 位定时器 H2 的框图



**(1) 8 位定时器 H 比较寄存器 0n (CMP0n)**

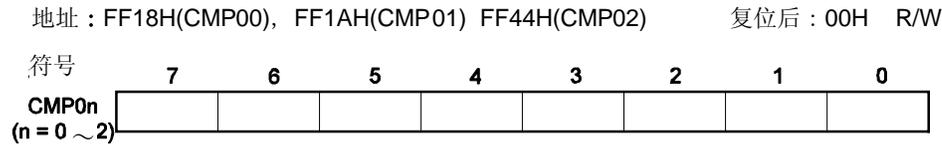
该寄存器可由 8 位存储器操作指令进行读写。该寄存器用于所有的定时器操作模式。

该寄存器不断地将 CMP0n 的值和 8 位定时器计数器 Hn 的计数值进行比较，并且当两个值相等时，产生一个中断请求信号 (INTTMHn) 并反转 TOHn 的输出。

当定时器停止时(TMHEn = 0)重写 CMP0n 的值。

复位信号的产生将该寄存器设置为 00H。

图 8-4. 8 位定时器 H 比较寄存器 0n (CMP0n) 的格式



**注意事项** CMP0n 在定时器计数操作期间不能被重写。CMP0n 在定时器计数操作期间可以重写(写入相同的值)。

**(2) 8 位定时器 H 比较寄存器 1n (CMP1n)**

可由 8 位存储器操作指令读写该寄存器。该寄存器用于 PWM 输出模式和载波发生器模式。

在 PWM 输出模式中，该寄存器不断地将 CMP1n 的设置值和 8 位定时器计数器 Hn 的计数值进行比较，当两个值相等时，反转 TOHn 的输出。不产生中断请求信号。

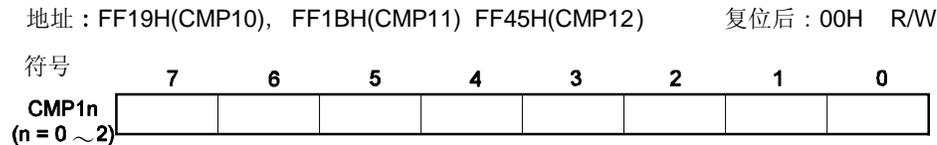
在载波发生器模式中，CMP1n 寄存器不断地将 CMP1n 的设置值和 8 位定时器计数器 Hn 的计数值进行比较，如果两个值相等，则产生一个中断请求信号 (INTTMHn)。同时，计数值清零。

可以在定时器计数期间对 CMP1n 进行更新(相同值)和重写。

如果在定时器操作期间重写 CMP1n 的值，则当定时器计数器的值和 CMP1n 以前的值相等时，锁存新的值并传输给 CMP1n，然后 CMP1n 的值用新值取代。如果计数值和 CMP1n 值的相等操作与 CMP1n 的写操作发生冲突，那么 CMP1n 的值将保持不变。

复位信号的产生对该寄存器清零(00H)。

图 8-5. 8 位定时器 H 比较寄存器 1n 的格式(CMP1n)



**注意事项** 在 PWM 输出模式和载波发生器模式下，当停止定时器计数操作(TMHEn = 0)后再启动定时器计数操作(TMHEn = 1)时，必须设置 CMP1n (即使对 CMP1n 设置相同的值也要再设置一次)。

**备注** n = 0 ~ 2。TOHn 仅适用于 TOH0 和 TOH1。

### 8.3 控制 8 位定时器 H0、H1 和 H2 的寄存器

以下四个寄存器用于控制 8 位定时器 H0、H1 和 H2。

- 8 位定时器 H 模式寄存器 n (TMHMDn)
- 8 位定时器 H 载波控制寄存器 1 (TMCYC1) <sup>註</sup>
- 端口模式寄存器 3 (PM3)
- 端口寄存器 3 (P3)

**注** 仅用于 8 位定时器 H1

#### (1) 8 位定时器 H 模式寄存器 n (TMHMDn)

该寄存器用于控制定时器 H 的模式。

可以由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号的产生对寄存器清零 (00H)。

**备注** n = 0 ~ 2

图 8-6. 8 位定时器 H 模式寄存器 0 (TMHMD0) 的格式

地址：FF69H

复位后：00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD0	TMHE0	CKS02	CKS01	CKS00	TMMD01	TMMD00	TOLEV0	TOEN0

TMHE0	是否允许定时器操作
0	停止定时器计数操作（计数器清零）
1	允许定时器计数（由输入时钟启动计数操作）

CKS02	CKS01	CKS00	计数时钟选择 <sup>注1</sup>			
				f <sub>PRS</sub> = 2 MHz	f <sub>PRS</sub> = 5 MHz	f <sub>PRS</sub> = 10 MHz
0	0	0	f <sub>PRS</sub> <sup>注2</sup>	2 MHz	5 MHz	10 MHz
0	0	1	f <sub>PRS</sub> /2	1 MHz	2.5 MHz	5 MHz
0	1	0	f <sub>PRS</sub> /2 <sup>2</sup>	500 kHz	1.25 MHz	2.5 MHz
0	1	1	f <sub>PRS</sub> /2 <sup>6</sup>	31.25 kHz	78.13 kHz	156.25 kHz
1	0	0	f <sub>PRS</sub> /2 <sup>10</sup>	1.95 kHz	4.88 kHz	9.77 kHz
1	0	1	TM50 输出 <sup>注</sup>			
其它情况			禁止设置			

TMMD01	TMMD00	定时器操作模式
0	0	间隔定时器模式
1	0	引脚输入允许宽度判断模式(PWM 模式)
其它情况		禁止设置

TOLEV0	定时器输出电平控制（默认模式）
0	低电平
1	高电平

TOEN0	定时器输出控制
0	禁止输出
1	允许输出

- 注
- 如果外围硬件时钟(f<sub>PRS</sub>) 操作在高速系统时钟(f<sub>XH</sub>) (XSEL = 1)下， f<sub>PRS</sub> 操作频率依据供电电压不同而变动。
    - V<sub>DD</sub> = 2.7 ~ 5.5 V: f<sub>PRS</sub> ≤ 10 MHz
    - V<sub>DD</sub> = 1.8 ~ 2.7 V: f<sub>PRS</sub> ≤ 5 MHz
  - 如果外围硬件时钟(f<sub>PRS</sub>) 操作在内部高速振荡时钟(f<sub>XH</sub>) (XSEL = 0)下，当 1.8 V ≤ V<sub>DD</sub> < 2.7 V 时，禁止设置 CKS02 = CKS01 = CKS00 = 0 (计数时钟：f<sub>PRS</sub>)。
  - 在选择 TM50 输出作为计数时钟时， TM50 和 CR50 的值相等时产生清零&启动模式 (TMC506 = 0)，先启动 8 位定时器/事件计数器 50，然后允许定时器进行 F/F 反转操作 (TMC501 = 1)。

- 注意事项
1. 当  $TMHE0=1$  时，禁止设置  $TMHMD0$  的其他位。然而，可以更新  $TMHMD0$ （写入相同值）。
  2. 在 PWM 输出模式中，当定时器停止计数后（ $TMHE0 = 0$ ）再次启动计数操作时（ $TMHE0 = 1$ ），必须设置 8 位定时器 H 比较寄存器 10（ $CMP10$ ）的值（即使设置的是相同的值，也必须再次设置）。
  3. 除  $TOH0$  的输出以外， $TOH0/P32/MCGO$  引脚的实际输出由  $PM32$  和  $P32$  决定。

备注 fPRS: 外围硬件时钟频率

图 8-7. 8 位定时器 H 模式寄存器 1 (TMHMD1) 的格式

地址: FF6CH 复位后: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1
	TMHE1	是否允许定时器操作						
	0	停止定时器计数操作 (计数器清零)						
	1	允许定时器计数 (由输入时钟启动计数操作)						
	CKS12	CKS11	CKS10	计数时钟选择 <sup>注1</sup>				
		<input type="checkbox"/>	<input type="checkbox"/>		$f_{PRS} = 2 \text{ MHz}$	$f_{PRS} = 5 \text{ MHz}$	$f_{PRS} = 10 \text{ MHz}$	
		<input type="checkbox"/>	<input type="checkbox"/>					
	0	0	0	$f_{PRS}$ <sup>注2</sup>	2 MHz	5 MHz	10 MHz	
	0	0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	
	0	1	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	
	0	1	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	
	1	0	0	$f_{PRS}/2^{12}$	0.49 kHz	1.22 kHz	2.44 kHz	
	1	0	1	$f_{RL}/2^7$	1.88 kHz (TYP.)			
	1	1	0	$f_{RL}/2^9$	0.47 kHz (TYP.)			
	1	1	1	$f_{RL}$	240 kHz (TYP.)			
	TMMD11	TMMD10	定时器操作模式					
	0	0	间隔定时器模式					
	0	1	载波发生器模式					
	1	0	PWM 输出模式					
	1	1	禁止设置					
	TOLEV1	定时器输出电平控制 (默认模式)						
	0	低电平						
	1	高电平						
	TOEN1	定时器输出控制						
	0	禁止输出						
	1	允许输出						

- 注
1. 如果外围硬件时钟( $f_{PRS}$ ) 操作在高速系统时钟( $f_{XH}$ ) ( $XSEL = 1$ )下,  $f_{PRS}$  操作频率依据供电电压不同而变动。
    - $V_{DD} = 2.7 \sim 5.5 \text{ V}$ :  $f_{PRS} \leq 10 \text{ MHz}$
    - $V_{DD} = 1.8 \sim 2.7 \text{ V}$ :  $f_{PRS} \leq 5 \text{ MHz}$
  2. 如果外围硬件时钟( $f_{PRS}$ ) 操作在内部高速振荡时钟( $f_{XH}$ ) ( $XSEL = 0$ )下, 当  $1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$  时, 禁止设置  $CKS12 = CKS11 = CKS10 = 0$  (计数时钟:  $f_{PRS}$ )。

- 注意事项
1. 当 **TMHE1=1** 时，禁止设置 **TMHMD1** 的其他位。然而，可以更新 **TMHMD1**（写入相同值）。
  2. 在 **PWM** 输出模式和载波发生模式中，当定时器停止计数后（**TMHE1=0**）再次启动计数器操作时（**TMHE1=1**），必须设置 8 位定时器 H 比较寄存器 11（**CMP11**）的值（即使设置的是相同的值，也必须再次设置）。
  3. 当使用载波发生器模式时，设置 **TMH1** 的计数时钟频率至少是 **TM51** 计数时钟频率的 6 倍。
  4. 除 **TOH1** 的输出以外，**TOH1/P31/INTP3** 引脚的实际输出由 **PM31** 和 **P31** 决定。

- 备注
1. **fPRS**: 外围硬件时钟频率
  2. **fRL**: 内部低速振荡时钟频率

图 8-8. 8 位定时器 H 模式寄存器 2 (TMHMD1) 的格式

地址: FF42H 复位后: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD2	TMHE2	CKS22	CKS21	CKS20	TMMD21	TMMD20	TOLEV2	TOEN2
	TMHE2	是否允许定时器操作						
	0	停止定时器计数操作 (计数器清零)						
	1	允许定时器计数 (由输入时钟启动计数操作)						
	CKS22	CKS21	CKS20	计数时钟选择 <sup>注1</sup>				
		•	•		f <sub>PRS</sub> =	f <sub>PRS</sub> =	f <sub>PRS</sub> =	
		•	•		2 MHz	5 MHz	10 MHz	
	0	0	0	f <sub>PRS</sub> <sup>注2</sup>	2 MHz	5 MHz	10 MHz	
	0	0	1	f <sub>PRS</sub> /2	1 MHz	2.5 MHz	5 MHz	
	0	1	0	f <sub>PRS</sub> /2 <sup>2</sup>	500 kHz	1.25 MHz	2.5 MHz	
	0	1	1	f <sub>PRS</sub> /2 <sup>4</sup>	125 kHz	312.5 kHz	625 kHz	
	1	0	0	f <sub>PRS</sub> /2 <sup>6</sup>	31.25 kHz	78.13 kHz	156.25 kHz	
	1	0	1	f <sub>PRS</sub> /2 <sup>10</sup>	1.95 kHz	4.88 kHz	9.77 kHz	
	1	1	0	f <sub>PRS</sub> /2 <sup>12</sup>	0.49 kHz	1.22 kHz	2.44 kHz	
	其他情况			禁止设置				
	TMMD21	TMMD20	定时器操作模式					
	0	0	间隔定时器模式					
	1	0	引脚输入允许宽度判断模式(PWM 输出)					
	其他情况		禁止设置					
	TOLEV2	定时器输出电平控制 (默认模式)						
	0	低电平						
	1	高电平						
	TOEN2	定时器输出控制						
	0	禁止输出						
	1	允许输出 <sup>注3</sup>						

- 注
- 如果外围硬件时钟(f<sub>PRS</sub>) 操作在高速系统时钟(f<sub>XH</sub>) (XSEL = 1)下, f<sub>PRS</sub> 操作频率依据供电电压不同而变动。
    - V<sub>DD</sub> = 2.7 ~ 5.5 V: f<sub>PRS</sub> ≤ 10 MHz
    - V<sub>DD</sub> = 1.8 ~ 2.7 V: f<sub>PRS</sub> ≤ 5 MHz
  - 如果外围硬件时钟(f<sub>PRS</sub>) 操作在内部高速振荡时钟(f<sub>XH</sub>) (XSEL = 0)下, 当 1.8 V ≤ V<sub>DD</sub> < 2.7 V 时, 禁止设置 CKS22 = CKS21 = CKS20 = 0 (计数时钟: f<sub>PRS</sub>)。
  - TMH2 定时器输出仅用作 TM52 的外部事件输入允许信号。不提供引脚的外部输出。

注意事项 当 TMHE2 = 1 时, 禁止设置 TMHMD2 其他位。

备注 f<sub>PRS</sub>: 外围硬件时钟频率

**(2) 8 位定时器 H 载波控制寄存器 1 (TMCYC1)**

该寄存器用于控制 8 位定时器 H1 的遥控输出和载波脉冲输出状态。

可以由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号的产生对寄存器清零 (00H)。

图 8-9. 8 位定时器 H 载波控制寄存器 1 (TMCYC1) 的格式

地址: FF6DH 复位后: 00H R/W<sup>注</sup>

Symbol	7	6	5	4	3	2	1	<0>
TMCYC1	0	0	0	0	0	RMC1	NRZB1	NRZ1

RMC1	NRZB1	遥控输出
0	0	低电平输出
0	1	在 INTTM51 信号输入上升沿高电平输出
1	0	低电平输出
1	1	在 INTTM51 信号输入上升沿载波脉冲输出

NRZ1	载波脉冲输出状态标志
0	禁止载波输出状态 (低电平状态)
1	允许载波输出状态 (RMC1=1: 载波脉冲输出, RMC1=0: 高电平状态)

注 第 0 位只读。

注意事项 当 TMHE1 = 1 时, 不能重写 RMC1。然而, TMCYC1 可以更新 (写入相同的值)。

**(3) 端口模式寄存器 3 (PM3)**

该寄存器按位设置端口 1 为输入输出操作模式。

在将 P32/TOH0/MCGO 和 P31/TOH1/INTP3 引脚用于定时器输出时, 需对 PM32 和 PM31 以及 P32 和 P31 的输出锁存器清零(0)。

可以由 1 位或 8 位存储器操作指令设置 PM3。

复位信号的产生将寄存器置 FFH。

图 8-10. 端口模式寄存器 3 (PM3)的格式

地址: FF23H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34	PM33	PM32	PM31	1

PM3n	P3n 引脚 I/O 模式选择 (n = 0 ~ 4)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

## 8.4 8 位定时器 H0、H1 和 H2 的操作

### 8.4.1 间隔定时器/方波输出操作

当 8 位定时器计数器 Hn 与比较寄存器 0n (CMP0n) 相等时, 将产生中断请求信号 (INTTMHn) 并对 8 位定时器计数器 Hn 清零 (00H)。

在间隔定时器模式中不使用比较寄存器 1n (CMP1n)。即使 CMP1n 寄存器已赋值, 也不会检测 8 位定时器计数器 Hn 与 CMP1n 寄存器的相等情况, 所以不影响定时器的输出。

通过将定时器 H 模式寄存器 n (TMHMDn) 的第 0 位 (TOENn) 置 1, 可以从 TOHn 输出任意频率的方波 (占空比 = 50%)。

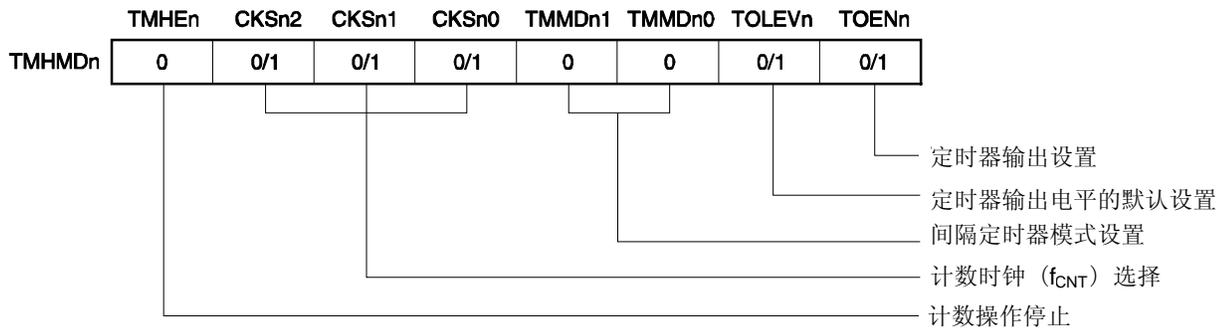
TMH2 定时器输出仅用作 TM52 外部事件输入允许信号。注意, 不提供外部输出引脚。

#### 设置

<1> 寄存器设置。

图 8-11. 间隔定时器/方波输出操作期间的寄存器设置

#### (i) 设置定时器 H 模式寄存器 n (TMHMDn)



#### (ii) CMP0n 寄存器设置

如果 N 作为比较值, 则时间间隔如下:

- 时间间隔 = (N + 1)/f<sub>CNT</sub>

<2> 当 TMHEn = 1 时, 开始计数。

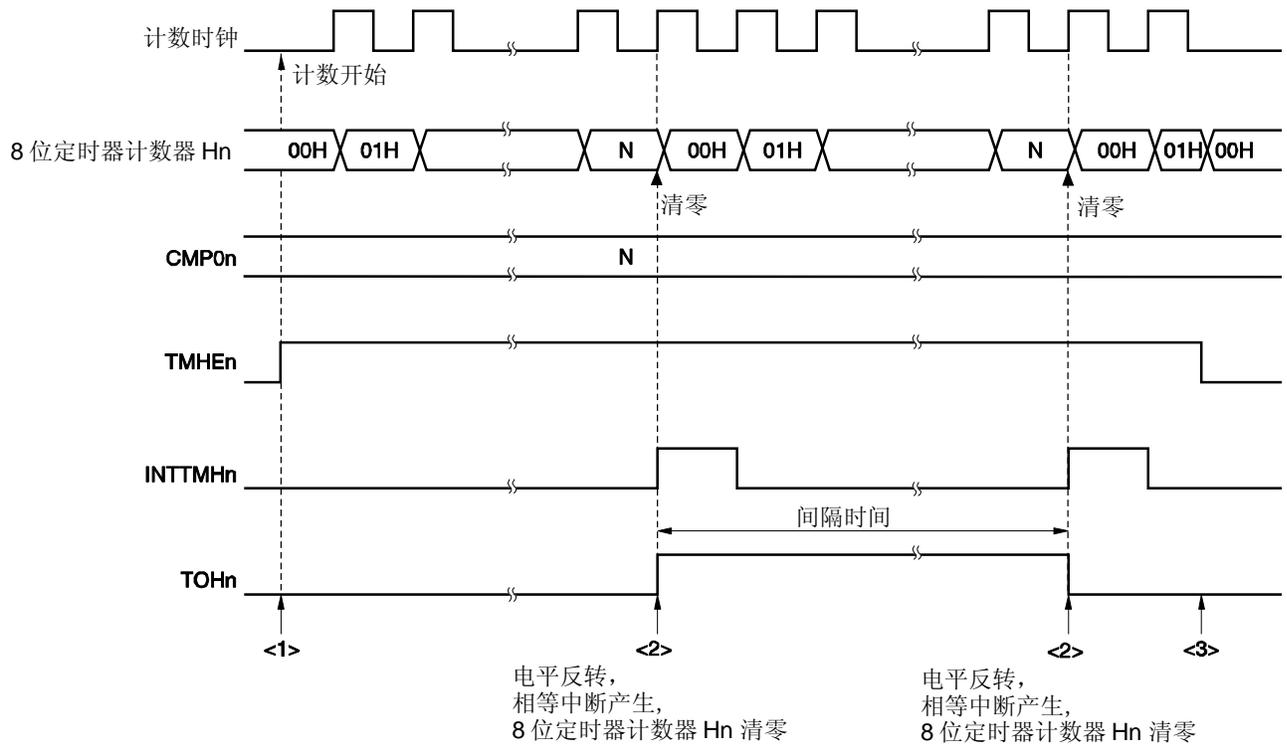
<3> 当 8 位定时器计数器 Hn 与 CMP0n 的值相等时, 将产生 INTTMHn 信号, 并对 8 位定时器计数器 Hn 清零 (00H)。

<4> 随后, 以相同的间隔时间重复产生 INTTMHn 信号。若要停止计数操作, 则将 TMHEn 清零(0)。

#### 备注

1. 如需了解输出引脚的设置情况, 参见 8.3 (3) 端口模式寄存器 3 (PM3)。
2. 如需了解允许 INTTMHn 信号中断的情况, 可参见 第十七章 中断功能。
3. n = 0 ~ 2。TOHn 仅用于 TOH0 和 TOH1。

图 8-12. 间隔定时器/方波输出操作时序(1/2)

(a) 基本操作 (当  $01H \leq CMP0n \leq FEH$  时的操作)

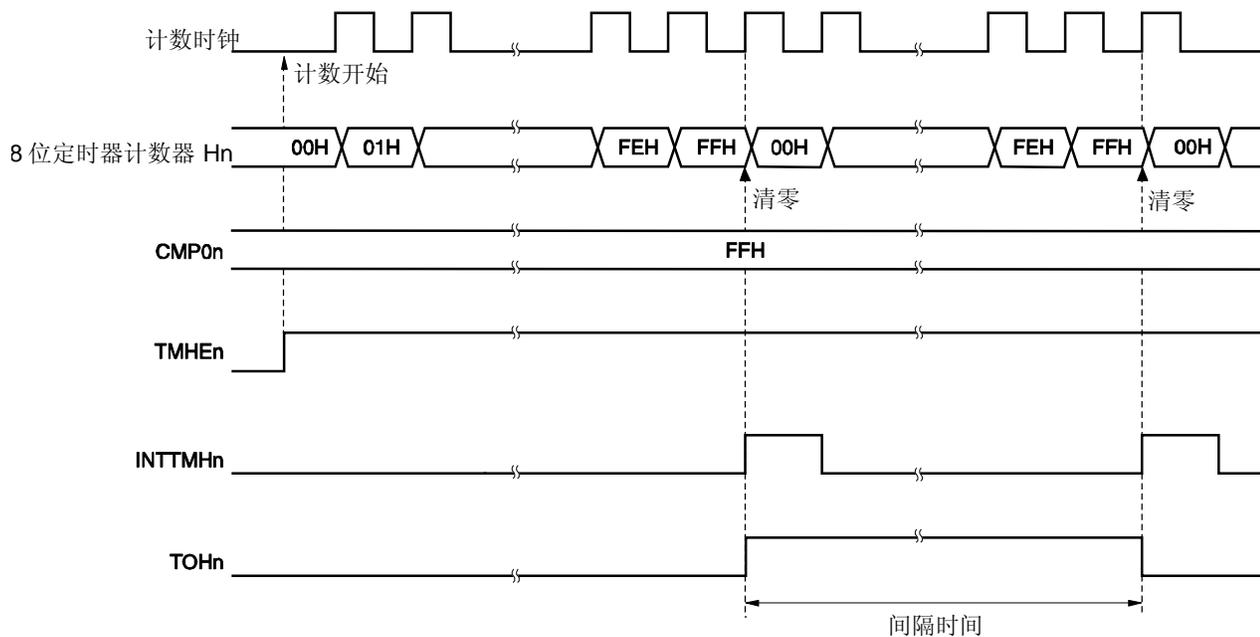
- <1> 通过将 TMHEn 位置 1，允许计数操作。一旦允许计数，则在不到 1 个时钟周期的时间内启动计数操作。
- <2> 当 8 位定时器计数器 Hn 与 CMP0n 寄存器的值相等时，将该定时器计数器清零并反转 TOHn 的输出电平。此外在计数时钟的上升沿输出 INTTMHn 信号。
- <3> 在定时器 Hn 操作期间将 TMHEn 位清零，设置 INTTMHn 信号和 TOHn 的输出为默认电平。如果在将 TMHEn 位清零(0)前它们已经处于默认电平，则电平保持不变。

备注 1.  $n = 0 \sim 2$ 。TOHn 仅用于 TOH0 和 TOH1。

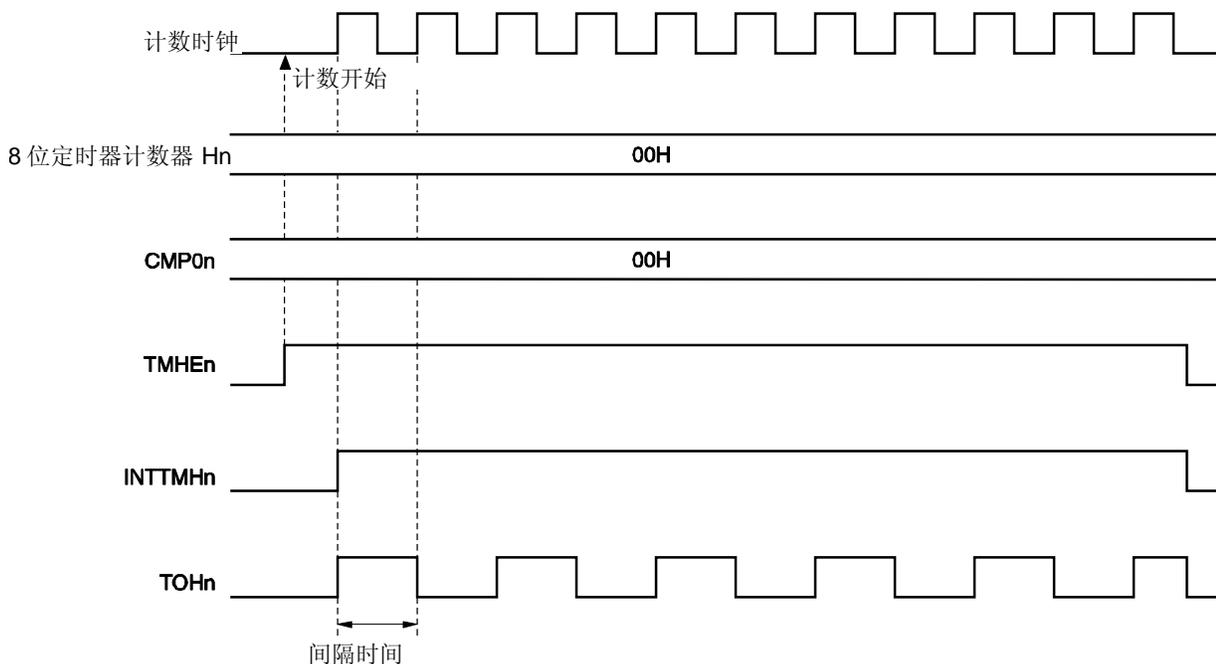
2.  $01H \leq N \leq FEH$

图 8-12. 间隔定时器/方波输出操作时序(2/2)

(b) 当 CMP0n = FFH 时的操作



(c) 当 CMP0n = 00H 时的操作



备注 n = 0 ~ 2。TOHn 仅用于 TOH0 和 TOH1。

### 8.4.2 PWM 输出操作

在 PWM 输出模式中，可以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 0n (CMP0n) 用于控制定时器输出 (TOHn) 的周期。禁止在定时器操作期间修改 CMP0n 寄存器的值。

8 位定时器比较寄存器 1n (CMP1n) 用于控制定时器输出 (TOHn) 的占空比。允许在定时器操作期间修改 CMP1n 寄存器的值。

PWM 输出模式中的操作如下所示。

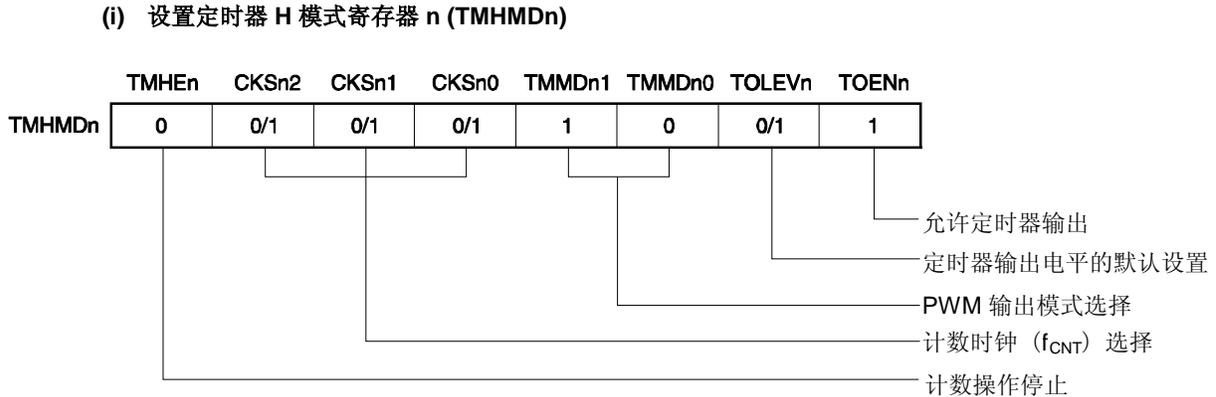
当定时器计数启动后，8 位定时器 Hn 和 CMP0n 寄存器相等，PWM 输出(TOHn 输出)活动电平并将 8 位定时器 Hn 清零。当 8 位定时器 Hn 和 CMP01 寄存器相等，PWM 输出(TOHn 输出) 活动电平。

TMH2 (PWM 输出) 定时器输出仅用作 TM52 的外部事件输入允许信号。注意，没有外部输出引脚。

#### 设置

<1> 寄存器设置。

图 8-13. PWM 输出模式下的寄存器设置



(ii) 设置寄存器 CMP0n

- 比较值(N): 周期设置

(iii) 设置寄存器 CMP1n

- 比较值(M): 占空比设置

- 备注
1. n = 0 ~ 2。TOHn 仅用于 TOH0 和 TOH1。
  2. 00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

<2> 当 TMHEn = 1 时，开始计数。

<3> CMP0n 是在允许计数操作后首次被比较的比较寄存器。当 8 位定时器计数器 Hn 与 CMP0n 寄存器的值相等时，将 8 位定时器计数器 Hn 清零、产生中断请求信号 (INTTMHn)，并且输出活动电平。同时切换比较寄存器（与 8 位定时器比较的寄存器）：由 CMP0n 改为 CMP1n。

<4> 当 8 位定时器计数器 Hn 与 CMP1n 寄存器相等时，输出非活动电平，并切换比较寄存器（与 8 位定时器比较的寄存器）：由 CMP1n 改为 CMP0n。此时不对 8 位定时器计数器 Hn 清零，也不产生 INTTMHn 信号。

<5> 重复执行过程<3>和<4>，可以获取具有任意占空比的脉冲。

<6> 若要停止计数操作，则设置 TMHEn = 0。

若 CMP0n 的值为 N、CMP1n 的值为 M，计数时钟频率为  $f_{CNT}$ ，则 PWM 脉冲的输出周期和占空比如下所示。

- PWM 脉冲输出周期 =  $(N + 1)/f_{CNT}$
- 占空比 =  $(M + 1)/(N + 1)$

**注意事项** 1. 可以在定时器计数操作时改变 CMP1n 的设置值。但从改变 CMP1n 寄存器的值到这个值被传输给寄存器需要经历三个操作时钟（通过 TMHMDn 寄存器的 CKSn2 到 CKSn0 位选择的时钟信号）

2. 在定时器计数操作停止后（TMHEn=0），再次启动定时器操作时（TMHEn=1），必须对 CMP1n 进行设置（即使是设置相同的值，也必须重新设置 CMP1n 寄存器）

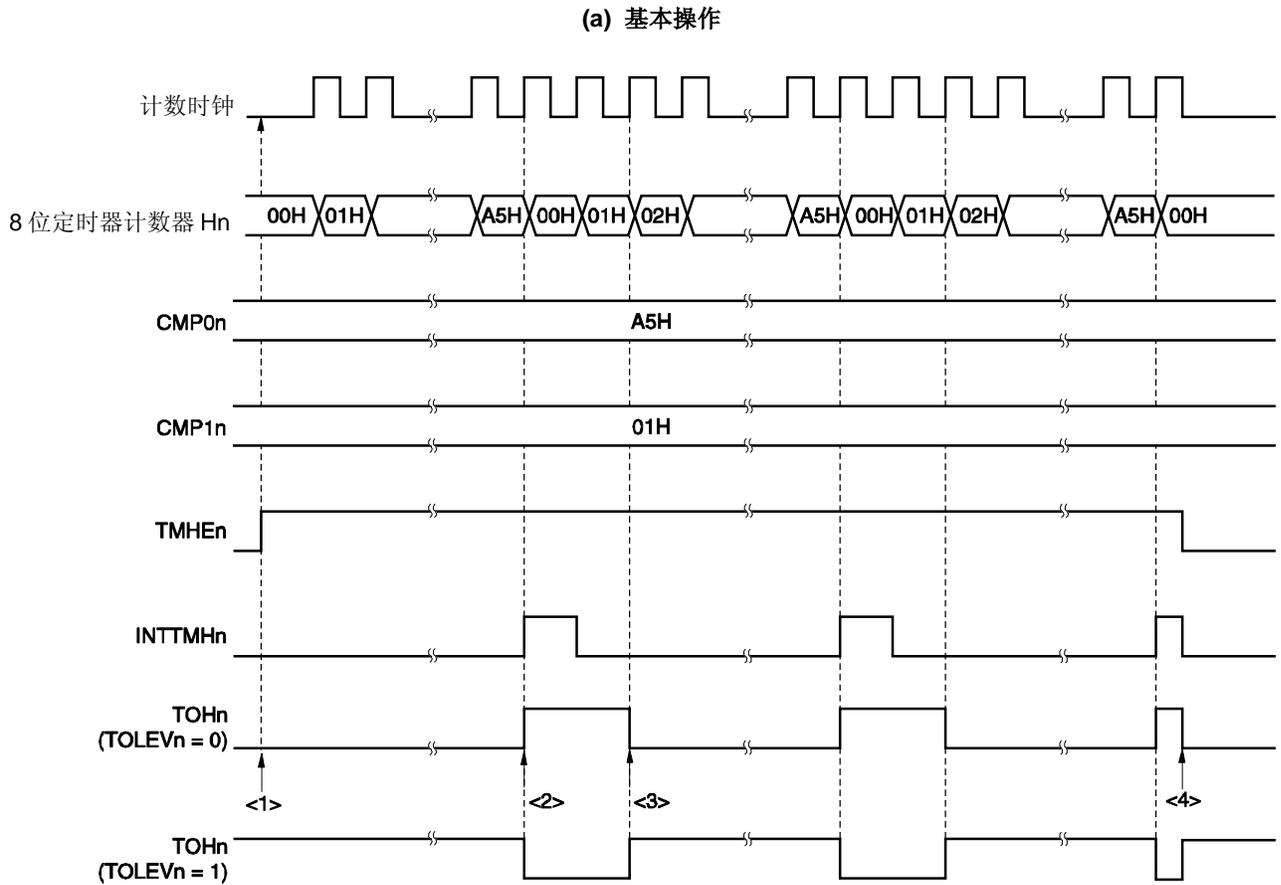
3. CMP1n 寄存器的设置值（M）和 CMP0n 寄存器的设置值（N）必须在以下范围内取值。

$$00H \leq \text{CMP1n (M)} < \text{CMP0n (N)} \leq \text{FFH}$$

**备注**

1. 如需了解输出引脚的设置，参见 8.3 (3) 端口模式寄存器 3 (PM3)。
2. 要了解如何允许 INTTMHn 信号中断的情况，可参见 第十七章 中断功能。
3.  $n = 0 \sim 2$ 。TOHn 仅用于 TOH0 和 TOH1。

图 8-14. PWM 输出模式下的操作时序(1/4)

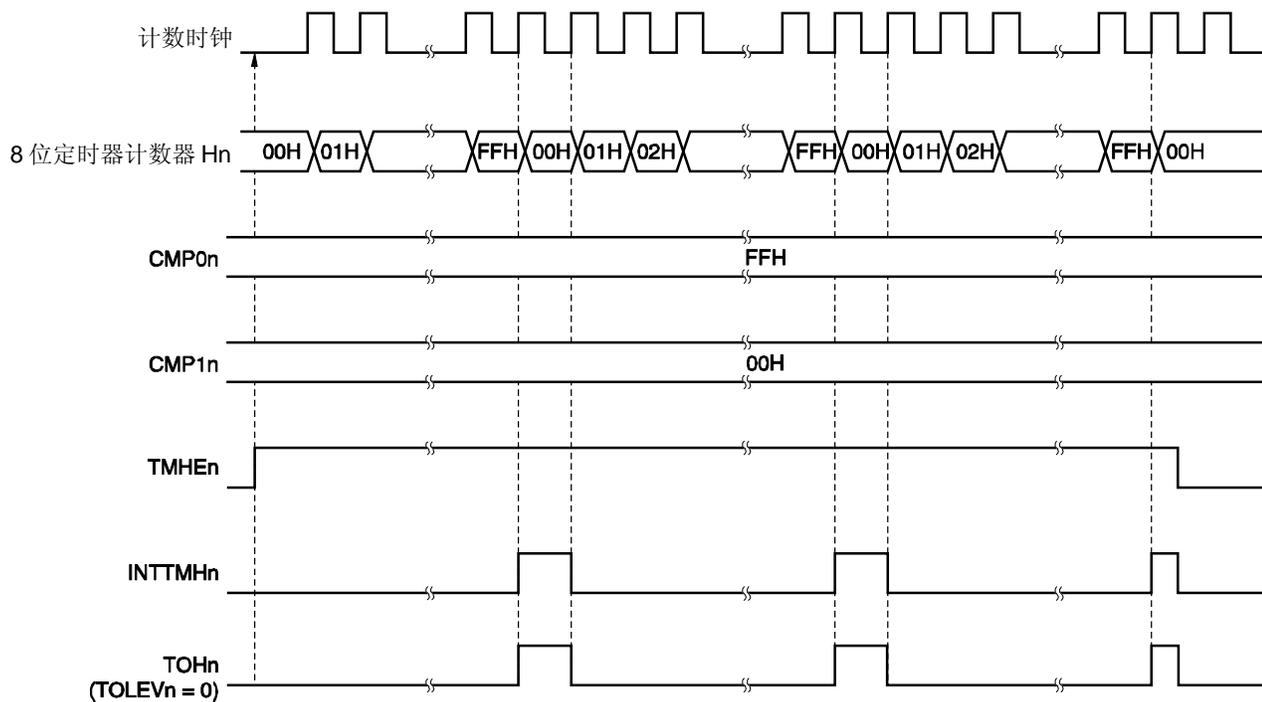


- <1> 通过设置  $TMHEn = 1$ ，允许计数操作。通过屏蔽一个计数时钟，启动 8 位定时器计数器  $Hn$ 。此时，PWM 输出活动电平。
- <2> 当 8 位定时器计数器  $Hn$  与  $CMP0n$  相等时将输出活动电平。此时，对 8 位定时器计数器  $Hn$  清零，并输出  $INTTMHn$  中断信号。
- <3> 当 8 位定时器计数器  $Hn$  与  $CMP1n$  相等时，输出非活动电平。此时，并不对 8 位定时器计数器  $Hn$  清零，也不输出  $INTTMHn$  中断信号。
- <4> 在定时器  $Hn$  操作期间若将  $TMHEn$  位清零，可以设置  $INTTMHn$  信号为默认值和 PWM 输出非活动电平。

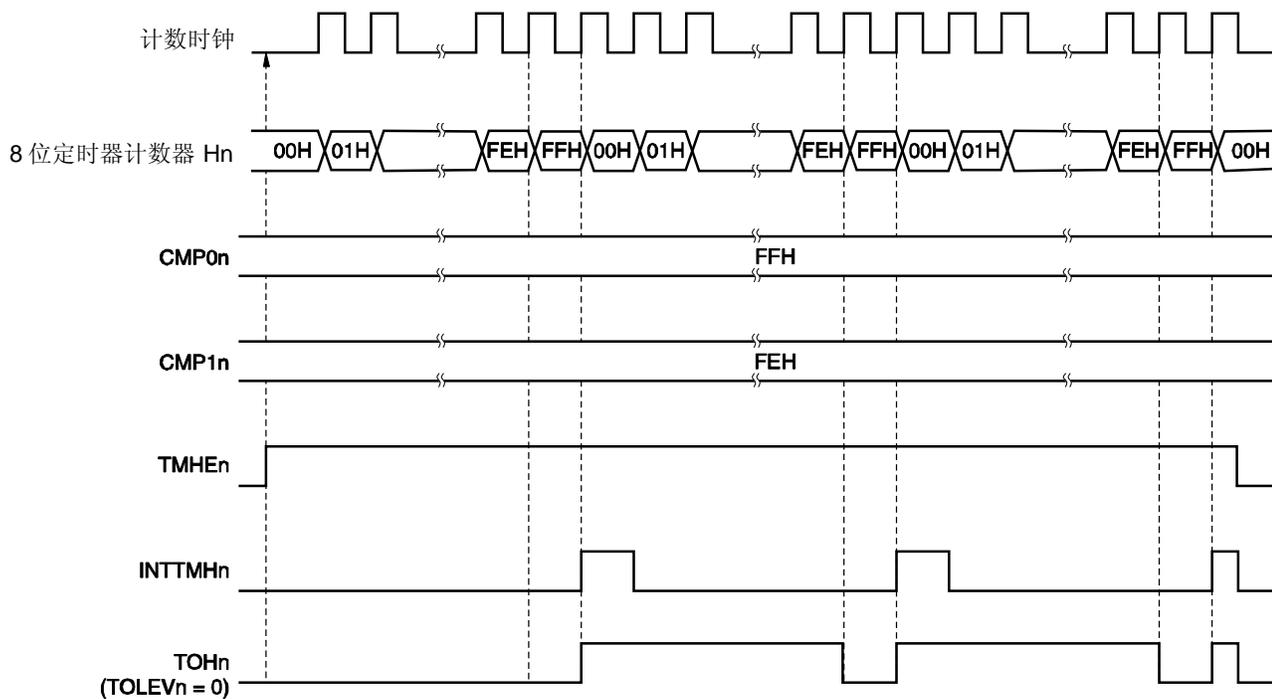
备注  $n = 0 \sim 2$ 。TOHn 仅用于 TOH0 和 TOH1。

图 8-14. PWM 输出模式下的操作时序(2/4)

(b) 当  $CMP0n = FFH$ ,  $CMP1n = 00H$  时的操作



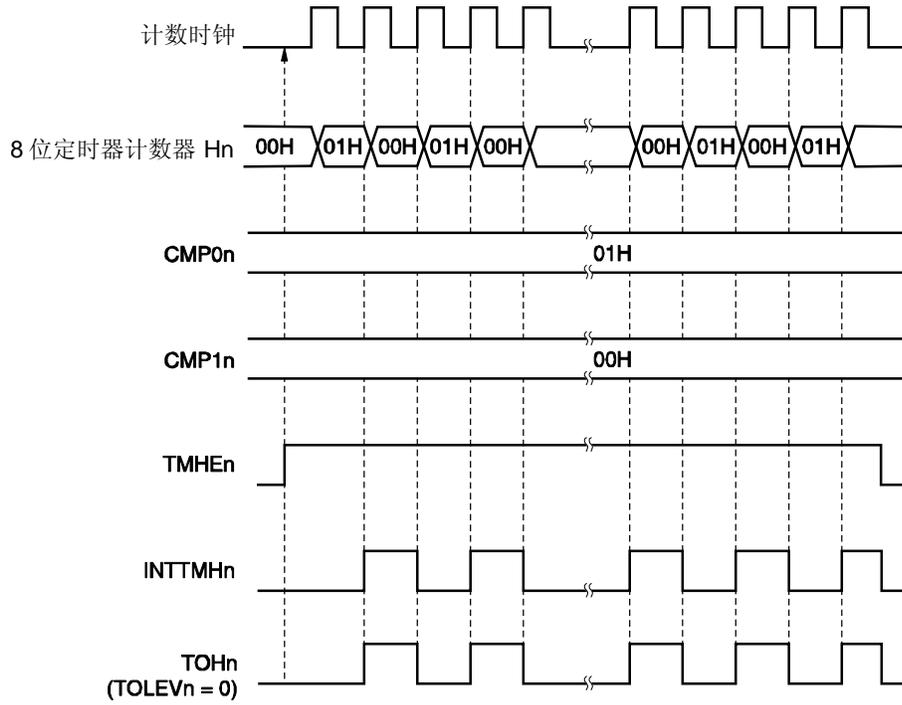
(c) 当  $CMP0n = FFH$ ,  $CMP1n = FEH$  时的操作



备注 n = 0 ~ 2。TOHn 仅用于 TOH0 和 TOH1。

图 8-14. PWM 输出模式下的操作时序(3/4)

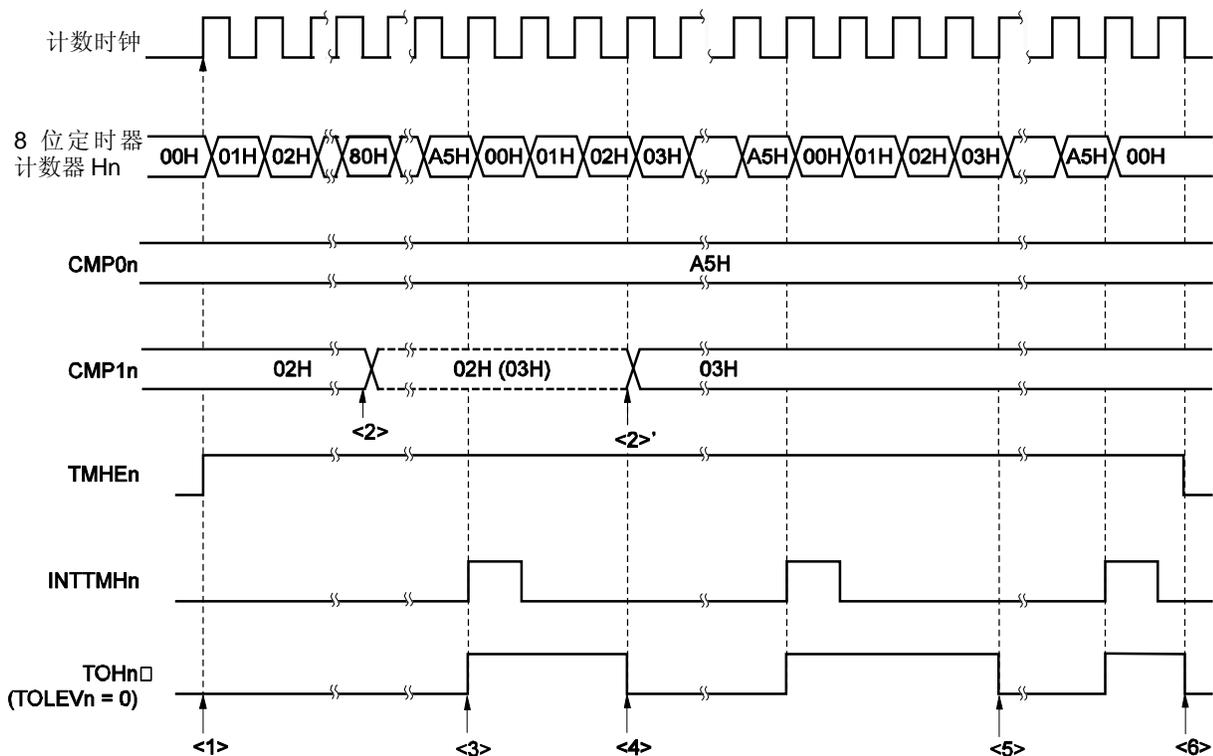
(d) 当  $CMP0n = 01H$ ,  $CMP1n = 00H$  时的操作



备注  $n = 0 \sim 2$ 。TOHn 仅用于 TOH0 和 TOH1。

图 8-14. PWM 输出模式下的操作时序(4/4)

## (e) 修改 CMP1n 的时序操作 (CMP1n = 02H → 03H, CMP0n = A5H)



- <1> 通过设置 TMHEn = 1 允许计数操作。通过屏蔽 1 个计数时钟，启动 8 位定时器计数器 Hn。此时 PWM 输出非活动电平。
- <2> 在定时器计数器操作期间可以修改 CMP1n 的值。修改操作与计数时钟不同步。
- <3> 当 8 位定时器计数器 Hn 与 CMP0n 的值相等时，将 8 位定时器计数器 Hn 的值清零、输出活动电平，并输出 INTTMHn 信号。
- <4> 如果 CMP1n 的值被修改，则修改后的值被锁存，且不传送到寄存器。当 8 位定时器计数器 Hn 的值与修改前 CMP1n 寄存器的值相等时，则将值传送到 CMP1n，这样 CMP1n 的值就被修改了 (<2>')。但从 CMP1n 的值被修改到将值传送到寄存器至少需要 3 个计数时钟。如果在 3 个计数时钟内产生一个相等信号，则不能将修改后的值传送到寄存器。
- <5> 当 8 位定时器计数器 Hn 的值与修改后的 CMP1n 的值相等时，输出非活动电平。此时不对 8 位定时器计数器 Hn 清零，也不产生 INTTMHn 信号。
- <6> 在定时器 Hn 操作期间若将 TMHEn 位清零，可以设置 INTTMHn 信号为默认值和使 PWM 输出非活动电平。

**备注** n = 0 ~ 2。TOHn 仅用于 TOH0 和 TOH1。

### 8.4.3 载波发生器操作（仅用于 8 位定时器 H1）

在载波发生器模式中 8 位定时器 H1 用于产生红外遥控器的载波信号，8 位定时器/事件计数器 51 用于红外遥控信号的产生（计时）

以 8 位定时器/事件计数器 51 设置的周期输出载波时钟。该载波时钟由 8 位定时器 H1 产生。

在载波发生器模式中，8 位定时器 H1 载波脉冲由 8 位定时器/事件计数器 51 控制输出，并从 TOH1 输出。

#### (1) 载波的发生

在载波发生器模式中，8 位定时器 H 比较寄存器 01（CMP01）产生低电平宽度的载波脉冲波形，而 8 位定时器 H 比较寄存器 11（CMP11）产生高电平宽度的载波脉冲波形。

在 8 位定时器 H1 操作期间允许修改 CMP11，但禁止修改 CMP01。

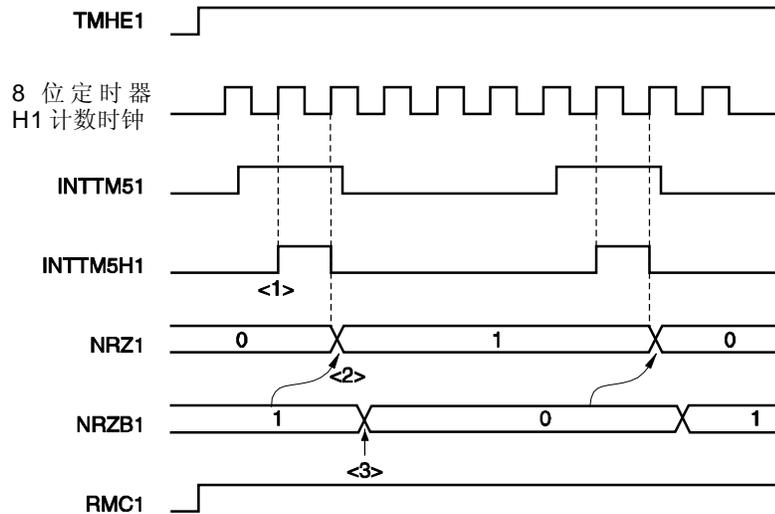
#### (2) 载波输出控制

载波输出由 8 位定时器/事件计数器 51 的中断请求信号（INTTM51）和 8 位定时器 H 载波控制寄存器（TMCYC1）的 NRZB1 和 RMC1 位控制。输出之间的关系如下所示。

RMC1 位	NRZB1 位	输出
0	0	低电平输出
0	1	在 INTTM51 信号输入上升沿高电平输出
1	0	低电平输出
1	1	在 INTTM51 信号输入上升沿载波脉冲输出

为了在计数操作期间控制载波脉冲输出，TMCYC1 寄存器的 NRZ1 和 NRZB1 位有 1 个主位和从位配置。NRZ1 位只读，而 NRZB1 位可读可写。INTTM51 信号与 8 位定时器 H1 计数时钟同步，且作为 INTTM5H1 信号输出。INTTM5H1 信号作为 NRZ1 位的数据传送信号，且 NRZB1 位的值被传送至 NRZ1 位。从 NRZB1 位传送至 NRZ1 位的时序如下所示。

图 8-15. 传送时序



- <1> INTTM51 信号与 8 位定时器 H1 的计数时钟同步，并作为 INTTM5H1 信号输出。
- <2> 在 INTTM5H1 信号上升沿的第 2 个时钟处将 NRZB1 位的值传送至 NRZ1 位。
- <3> 在中断服务程序中将下一个值写入 NRZB1 位。该中断服务程序已由 INTTM5H1 中断启动或通过轮询中断请求标志检测到时序后启动。将下一次要计数的值写入 CR51 中。

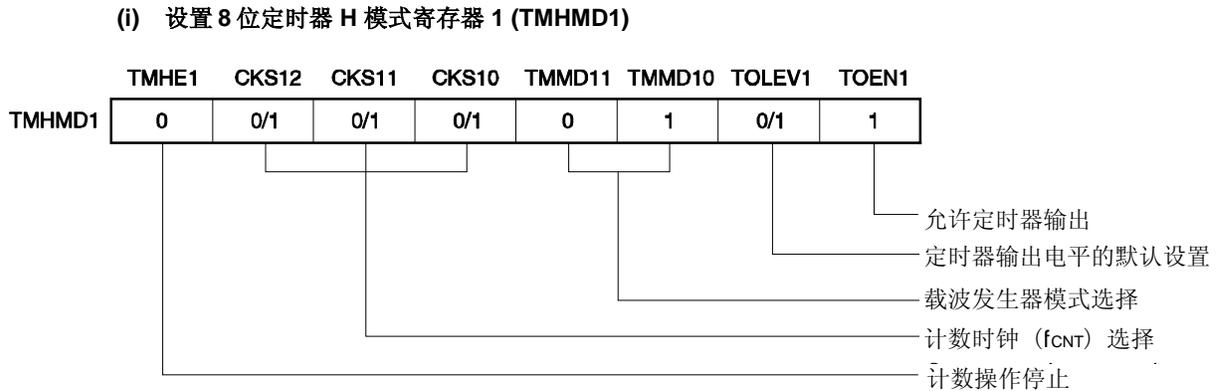
- 注意事项**
1. NRZB1 位被重写后至少到第 2 个时钟时才能被再次重写，否则，从 NRZB1 位到 NRZ1 位的传送不能得到保证。
  2. 在载波发生器模式中使用 8 位定时器/事件计数器 51 时，在<1>时刻将产生中断。如果在其他模式中使用 8 位定时器/事件计数器 51，中断产生的时序则不相同。

**备注** INTTM5H1 是内部信号并不是中断源。

## 设置

## &lt;1&gt; 寄存器设置

图 8-16. 在载波发生器模式下的寄存器设置



## (ii) CMP01 寄存器设置

- 比较值

## (iii) CMP11 寄存器设置

- 比较值

## (iv) TMCYC1 寄存器设置

- RMC1 = 1 ... 遥控输出允许位
- NRZB1 = 0/1 ... 载波输出允许位

## (v) TCL51 和 TMC51 寄存器设置

- 参见 7.3 控制 8 位定时器/事件计数器 50、51 和 52 的寄存器。

<2> 当 TMHE1 = 1 时，8 位定时器 H1 开始计数。

<3> 当 8 位定时器模式控制寄存器 51 (TMC51) 的 TCE51=1 时，8 位定时器/事件计数器 51 开始计数。

<4> 允许计数后，第 1 个用于比较的比较寄存器是 CMP01。当 8 位定时器计数器 H1 的计数值与 CMP01 的值相等时，将产生 INTTMH1 信号，并对 8 位定时器计数器 H1 清零，而此时与 8 位定时器计数器比较的比较寄存器从 CMP01 被切换为 CMP11。

<5> 当 8 位定时器计数器 H1 的计数值与 CMP11 寄存器的值相等时，将产生 INTTMH1 信号，并对 8 位定时器计数器 H1 清零，而此时与 8 位定时器计数器比较的比较寄存器从 CMP11 被切换为 CMP01。

<6> 重复执行过程<4>和<5>，将产生 1 个载波时钟。

<7> INTTM51 信号与 8 位定时器 H1 的计数时钟同步，并作为 INTTM5H1 信号输出。INTTM5H1 信号作为 NRZB1 位的数据传输信号，并将 NRZB1 位的值传送至 NRZ1 位。

<8> 在中断服务程序中将下一个值写入 NRZB1 位。该中断服务程序已由 INTTM5H1 中断启动或通过轮询中断请求标志检测到时序后启动。将下一次要计数的值写入 CR51 中。

<9> 当 NRZ1 位是高电平时，将从 TOH1 引脚输出载波时钟。

<10> 通过执行上述过程，可以获得任意形式的载波时钟。若要停止计数操作，可以将 TMHE1 清零。

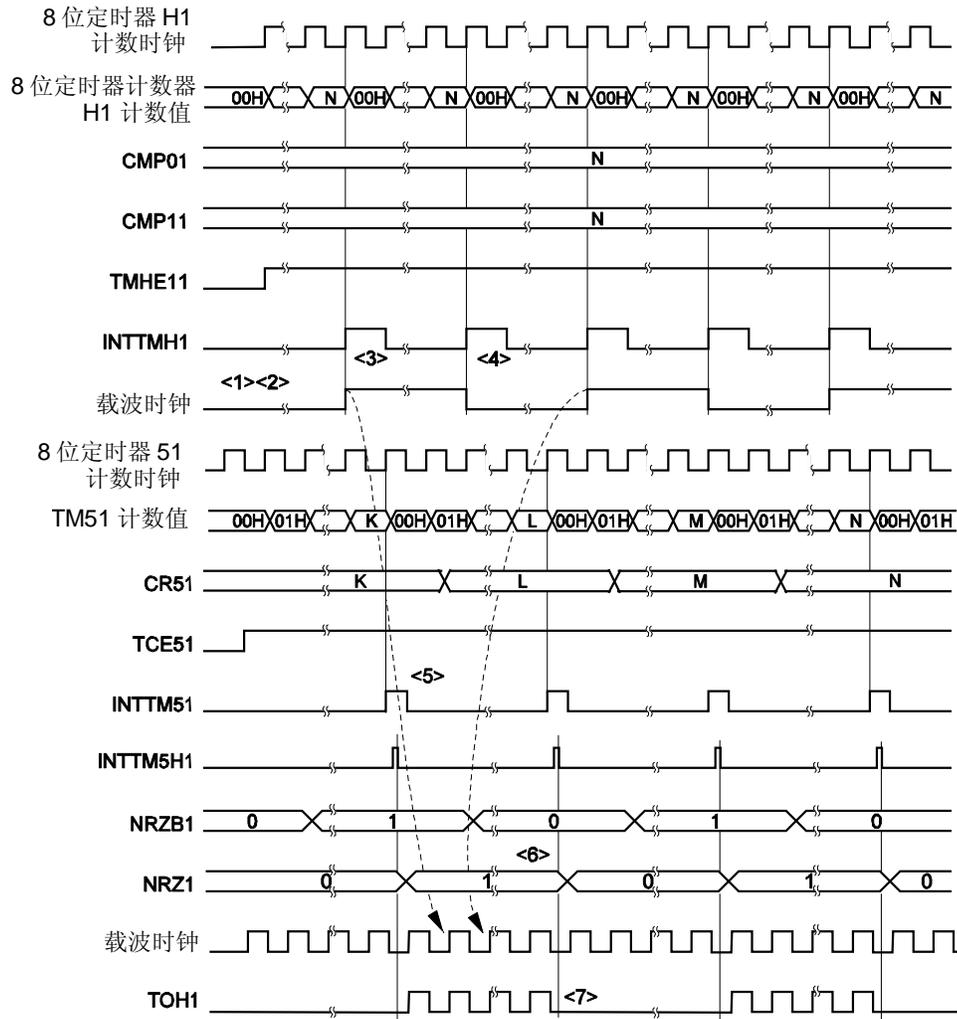
如果 CMP01 的值为 N、CMP11 的值为 M、计数时钟频率为  $f_{CNT}$ ，则载波时钟输出周期与占空比如下所示。

- 载波时钟输出周期 =  $(N + M + 2)/f_{CNT}$
- 占空比=高电平宽度/载波时钟输出宽度 =  $(M + 1) / (N + M + 2)$

- 注意事项**
1. 在定时器计数操作停止后（TMHE1=0），再启动定时器操作（TMHE1=1）时，必须设置 CMP11（即使是相同的值，也必须对 CMP11 重新设置）
  2. 设置 TMH1 的计数时钟频率至少是 TM51 的计数时钟频率的 6 倍。
  3. CMP01 和 CMP11 的值必须在 01H 和 FFH 之间。
  4. 定时器计数期间可以改变 CMP11 寄存器的值。但是，从改变 CMP11 的值到将该值传输到寄存器需要三个操作时钟（由 TMHMD1 寄存器的 CKS12~CKS10 位选择的时钟信号）
  5. 在计时操作开始前必须设置 RMC1。

- 备注**
1. 如需了解输出引脚的设置，参见 8.3 (3) 端口模式寄存器 3 (PM3)。
  2. 要了解如何允许 INTTMH1 信号中断的情况，可参见 第十七章 中断功能。

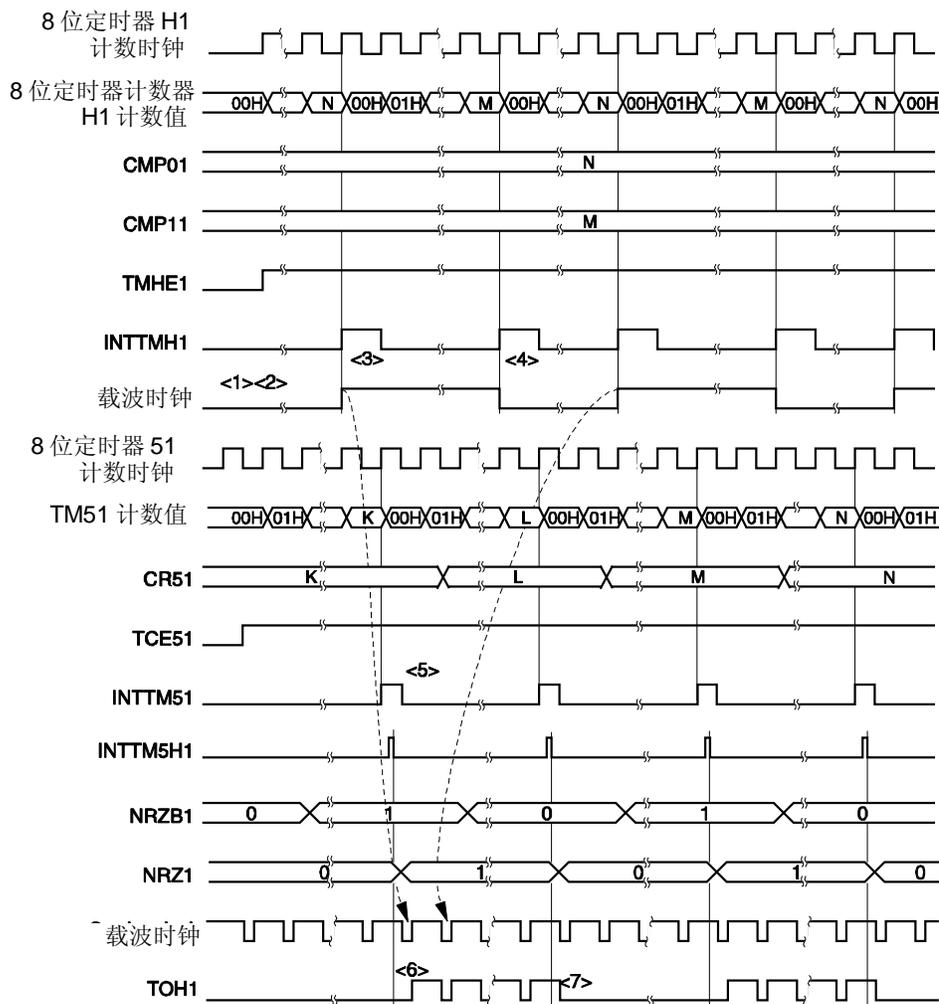
图 8-17. 载波发生器模式操作时序(1/3)

(a) 当  $CMP01 = N$ ,  $CMP11 = N$  时的操作

- <1> 当  $TMHE1 = 0$  且  $TCE51 = 0$  时，停止 8 位定时器计数器 H1 的操作。
- <2> 当  $TMHE1 = 1$  时，8 位定时器计数器 H1 开始计数。此时载波时钟保持默认电平状态。
- <3> 当 8 位定时器计数器 H1 的计数值与  $CMP01$  的值相等时，产生第 1 个  $INTTMH1$  信号并反转载波时钟信号，同时与 8 位定时器计数器 H1 比较的比较寄存器从  $CMP01$  切换为  $CMP11$ 。8 位定时器计数器 H1 的值被清零 (00H)。
- <4> 当 8 位定时器计数器 H1 的计数值与  $CMP11$  的值相等时，产生  $INTTMH1$  信号并反转载波时钟信号，同时与 8 位定时器计数器 H1 比较的比较寄存器从  $CMP11$  切换为  $CMP01$ 。8 位定时器计数器 H1 的值被清零 (00H)。重复执行过程<3>和<4>，将产生 1 个载波时钟，且占空比恒为 50%。
- <5> 当  $INTTM51$  信号产生时，它与 8 位定时器 H1 计数时钟同步，并作为  $INTTM5H1$  信号输出。
- <6>  $INTTM5H1$  信号用作  $NRZB1$  位的数据传输信号，并将  $NRZB1$  位的值传送至  $NRZ1$  位。
- <7> 设置  $NRZ1 = 0$  时， $TOH1$  输出低电平。

**备注**  $INTTM5H1$  是内部信号并不是中断源。

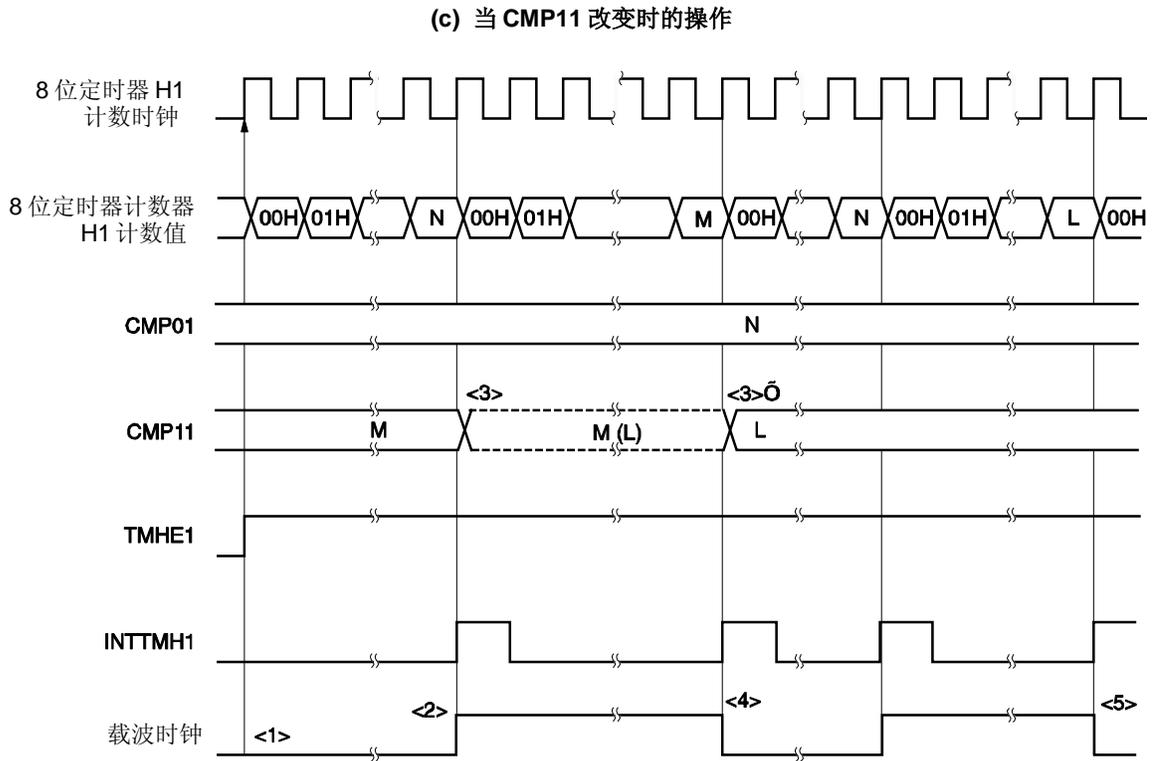
图 8-17. 载波发生器模式操作时序 (2/3)

(b) 当  $CMP01 = N$ ,  $CMP11 = M$  时的操作

- <1> 当  $TMHE1 = 0$  且  $TCE51 = 0$  时，停止 8 位定时器计数器 H1 的操作。
- <2> 当  $TMHE1 = 1$  时，8 位定时器计数器 H1 开始计数。此时载波时钟保持默认电平状态。
- <3> 当 8 位定时器计数器 H1 的计数值与  $CMP01$  的值相等时，产生第 1 个  $INTTMH1$  信号并反转载波时钟信号，同时与 8 位定时器计数器 H1 比较的比较寄存器从  $CMP01$  切换为  $CMP11$ 。8 位定时器计数器 H1 的值被清零 (00H)。
- <4> 当 8 位定时器计数器 H1 的计数值与  $CMP11$  的值相等时，产生  $INTTMH1$  信号并反转载波时钟信号，同时与 8 位定时器计数器 H1 比较的比较寄存器从  $CMP11$  切换为  $CMP01$ 。8 位定时器计数器 H1 的值被清零 (00H)。重复执行过程<3>和<4>，将产生 1 个载波时钟，且占空比不等于 50%。
- <5> 当  $INTTM51$  信号产生时，它与 8 位定时器 H1 计数时钟同步，并作为  $INTTM5H1$  信号输出。
- <6> 若  $NRZ1 = 1$ ，在载波时钟的第 1 个上升沿处输出 1 个载波信号。
- <7> 当  $NRZ1 = 0$  时， $TOH1$  输出保持高电平状态，在载波时钟为高电平时（在过程<6>和<7>中，保证载波时钟波形的高电平宽度）不会转变为低电平。

备注  $INTTM5H1$  是内部信号并不是中断源。

图 8-17. 载波发生器模式操作时序 (3/3)



- <1> 当  $TMHE1 = 1$  时，8 位定时器 H1 开始计数。此时载波时钟保持默认电平状态。
- <2> 当 8 位定时器计数器 H1 的计数值与  $CMP01$  的值相等时，输出  $INTTMH1$  信号、反转载波信号，并将定时器计数器清零(00H)。同时与 8 位定时器计数器 H1 比较的比较寄存器从  $CMP01$  切换为  $CMP11$ 。
- <3>  $CMP11$  与计数时钟不同步，在 8 位定时器 H1 操作期间可以修改  $CMP11$  的值，但修改后的新值 (L) 被锁存。当 8 位定时器计数器 H1 的计数值与  $CMP11$  修改前的值 (M) 相等 (<3>' ) 时，修改  $CMP11$  的值 (<3>' )。
- 但从改变  $CMP11$  的值到将该值传送到寄存器需要 3 个计数时钟。即使在 3 个计数时钟内产生相等信号，新值不会被传送到寄存器
- <4> 当 8 位定时器计数器 H1 的计数值与  $CMP11$  修改前的值 (M) 相等时，输出  $INTTMH1$  信号、反转载波信号，并将定时器计数器清零(00H)。同时与 8 位定时器计数器 H1 比较的比较寄存器从  $CMP11$  切换为  $CMP01$ 。
- <5> 由修改后的值 (L) 确定 8 位定时器计数器 H1 的计数值与  $CMP11$  的值再次相等时的时序。

## 第九章 实时计数器

### 9.1 实时计数器的功能

实时计数器的特性如下。

- 具有年、月、星期、天、小时、分钟和秒的计数器，可以共计 99 年。
- 固定周期中断功能(周期: 1 月到 0.5 秒)
- 闹表中断功能 (闹表: 星期、小时、分钟)
- 间隔中断功能
- 1 Hz 的引脚输出功能
- 512 Hz 或 16.384 kHz 或 32.768 kHz 的引脚输出功能

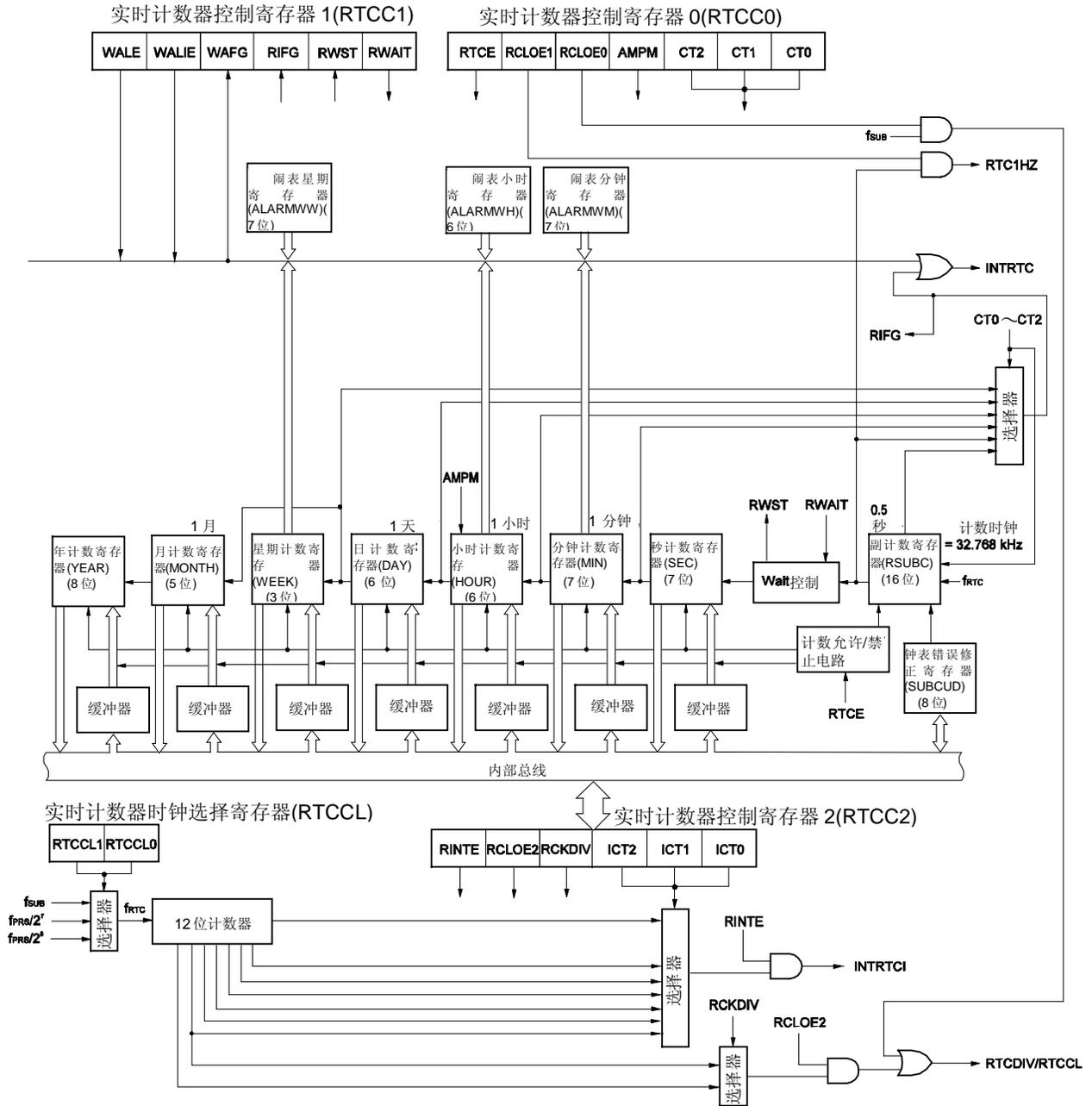
### 9.2 实时计数器的配置

实时计数器包括以下硬件。

表 9-1. 实时计数器的配置

项目	配置
控制寄存器	实时计数器时钟选择寄存器 (RTCCL)
	实时计数器控制寄存器 0 (RTCC0)
	实时计数器控制寄存器 1 (RTCC1)
	实时计数器控制寄存器 2 (RTCC2)
	副计数寄存器 (RSUBC)
	秒计数寄存器 (SEC)
	分钟计数寄存器 (MIN)
	小时计数寄存器 (HOUR)
	日计数寄存器 (DAY)
	星期计数寄存器 (WEEK)
	月计数寄存器 (MONTH)
	年计数寄存器 (YEAR)
	钟表错误修正寄存器 (SUBCUD)
	闹表分钟寄存器 (ALARMWM)
	闹表小时寄存器 (ALARMWH)
	闹表星期寄存器 (ALARMWW)

图 9-1. 实时计数器的框图



### 9.3 控制实时计数器的寄存器

以下 16 个寄存器用于控制实时计数器。

#### (1) 实时计数器时钟选择寄存器(RTCCL)

该寄存器可控制实时计数器模式。

可以由 1 位或 8 位存储器操作指令设置 RTCCL。

复位信号的产生对寄存器清零 (00H)。

图 9-2. 实时计数器时钟选择寄存器(RTCCL)的格式

地址: FF54H 复位后: 00H R/W

符号	7	6	5	4	3	2	<1>	<0>
RTCCL	0	0	0	0	0	0	RTCCL1	RTCCL0

RTCCL1	RTCCL0	控制实时计数器(RTC)输入时钟( $f_{RTC}$ )
0	0	$f_{SUB}$
0	1	$f_{PRS}/2^7$
1	0	$f_{PRS}/2^8$
1	1	禁止设置

- 备注
- 当  $f_{PRS} = 4.19 \text{ MHz}$ ,  $f_{RTC} = f_{PRS}/2^7 = 32.768 \text{ kHz}$
  - 当  $f_{PRS} = 8.38 \text{ MHz}$ ,  $f_{RTC} = f_{PRS}/2^8 = 32.768 \text{ kHz}$

#### (2) 实时计数器控制寄存器 0 (RTCC0)

RTCC0 寄存器是 8 位寄存器，它用于启动或停止实时计数器操作、控制 RTCCL 和 RTC1HZ 引脚、设置 12 或 24 小时系统和固定周期中断功能。

可以由 1 位或 8 位存储器操作指令设置 RTCC0。

复位信号的产生对寄存器清零 (00H)。

图 9-3. 实时计数器控制寄存器 0 (RTCC0)的格式

地址: FF89H 复位后: 00H R/W

符号	<7>	6	<5>	<4>	3	2	1	0
RTCC0	RTCE	0	RCLOE1	RCLOE0	AMPM	CT2	CT1	CT0
RTCE	实时计数器操作控制							
0	停止计数器操作。							
1	启动计数器操作。							
RCLOE1	RTC1HZ 引脚输出控制							
0	禁止 RTC1HZ 引脚输出(1 Hz)。							
1	允许 RTC1HZ 引脚输出 (1 Hz)。							
RCLOE0 <sup>#</sup>	RTCCL 脚输出控制							
0	禁止 RTCCL 引脚输出 (32.768 kHz)。							
1	允许 RTCCL 引脚输出 (32.768 kHz)。							
AMPM	选择12-/24 小时系统							
0	12 小时系统(a.m. 和 p.m. 显示)							
1	24 小时系统							
<ul style="list-style-type: none"> <li>• 改变 AMPM 的值, 要将 RWAIT (RTCC1 的第 0 位)置为 1, 并重置小时计数寄存器(HOUR)。</li> <li>• 表 9-2 表示显示的时间识字显示。</li> </ul>								
CT2	CT1	CT0	选择固定周期中断(INTRTC)					
0	0	0	不用作固定周期中断功能					
0	0	1	每 0.5 s 一次(与秒累加同步)					
0	1	0	每 1 s 一次(与秒累加相同)					
0	1	1	每 1 m 一次(每分钟的 00 秒)					
1	0	0	每 1 小时一次 (每小时的 00 分钟 00 秒)					
1	0	1	每 1 天一次 (每天的 00 小时 00 分钟 00 秒)					
1	1	x	每 1 月一次 (每月的第 1 天 a.m 00 小时 00 分 00 秒)					
CT2 ~ CT0 值改变后, 将中断请求清零。								

注 不能同时允许 RCLOE0 和 RCLOE2 。

注意事项 当 RTCE = 1 时, 如果改变 RCLOE0 和 RCLOE1, 32.768 kHz 和 1 Hz 输出信号将产生一个窄脉冲。

备注 x: 不必考虑

表 9-2. 时间数字显示

24 小时系统	12 小时系统	24 小时系统	12 小时系统
00	12 (AM12)	12	32 (PM12)
01	01 (AM1)	13	21 (PM1)
02	02 (AM2)	14	22 (PM2)
03	03 (AM3)	15	23 (PM3)
04	04 (AM4)	16	24 (PM4)
05	05 (AM5)	17	25 (PM5)
06	06 (AM6)	18	26 (PM6)
07	07 (AM7)	19	27 (PM7)
08	08 (AM8)	20	28 (PM8)
09	09 (AM9)	21	29 (PM9)
10	10 (AM10)	22	30 (PM10)
11	11 (AM11)	23	31 (PM11)

**(3) 实时计数器控制寄存器 1 (RTCC1)**

RTCC1 寄存器是 8 位寄存器，它用于控制闹表中断功能和计数器等待时间。

可以由 1 位或 8 位存储器操作指令设置 RTCC1。

复位信号的产生对寄存器清零 (00H)。

图 9-4. 实时计数器控制寄存器 1 (RTCC1)的格式 (1/2)

地址: FF8AH 复位后: 00H R/W

符号	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	闹表操作控制
0	相等操作无效
1	相等操作有效
设置闹表寄存器 (WALIE 的 RTCC1 位, ALARMWM 寄存器, ALARMWH 寄存器和 ALARMWW 寄存器), 禁止 WALE (将其清零)。	

WALIE	控制闹表中断(INTRTC)功能操作
0	闹表相等时不产生中断。
1	闹表相等时产生中断。

WAFG	闹表检测状态标志
0	闹表不相等
1	检测闹表相等
此状态标志显示闹表相等检测。仅当 WALE = 1 和检测到闹表相等后置一时钟(32.768 kHz)为 1 时有效。此标志在写入“0”后清零, 写入“1”无效。	

图 9-4. 实时计数器控制寄存器 1 (RTCC1)的格式 (2/2)

RIFG	固定周期中断状态标志
0	不产生固定周期中断。
1	产生固定周期中断。
此标志显示固定周期中断产生状态。当固定周期中断产生，其置“1”。 此标志在写入“0”后清零，写入“1”无效。	

RWST	实时计数器等待状态标志
0	计数器操作。
1	读或写计数器值模式。
此状态标志显示 RWAIT 无论是否设置有效。 读或写计数器值前，确认此标志值为 1。	

RWAIT	等待控制实时计数器
0	设置计数器操作。
1	停止 SEC ~ YEAR 计数器。读或写计数器值模式。
此位控制计数器操作。 确保写入“1”，读或写计数器值。 因为 RSUBC 操作继续，在 1 秒内完成读或写，并将此位清零。 当 RWAIT = 1 时，它占用 1 个时钟 (32.768 kHz) 直到计数器值被读出或写入。 当 RWAIT = 1 时，如果 RSUBC 溢出，在 RWAIT = 0 后它累加。如果秒计数寄存器写入，因为 RSUBC 清零，它将不能累加。	

**注意事项** 如果写入 WAFG 标志执行 1 位操作指令，RIFG 标志将清零。因此，执行 WAFG 标志写入，必须使用 8 位操作指令，和此时将 RIFG 标志置为 1。同样，执行 RIFG 标志写入，使用 8 位操作指令和 WAFG 标志置 1。

**备注** 固定周期中断和闹表相等中断使用相同的中断源 (INTRTC)。当同时使用这两种类型中断时，通过检查固定周期中断状态标志(RIFG)和闹表检测状态标志(WAFG)，来判断 INTRTC 产生。

**(4) 实时计数器控制寄存器 2 (RTCC2)**

RTCC2 寄存器是 8 位寄存器，它用于控制间隔中断功能和 RTCDIV 引脚。

可以由 1 位或 8 位存储器操作指令设置 RTCC2。

复位信号的产生对寄存器清零 (00H)。

图 9-5. 实时计数器控制寄存器 2 (RTCC2)的格式

地址: FF8BH 复位后: 00H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
RTCC2	RINTE	RCLOE2	RCKDIV	0	0	ICT2	ICT1	ICT0

RINTE	ICT2	ICT1	ICT0	选择间隔中断(INTRTCI)
0	×	×	×	不产生间隔中断
1	0	0	0	$2^6/f_{RTC}$ (1.953125 ms)
1	0	0	1	$2^7/f_{RTC}$ (3.90625 ms)
1	0	1	0	$2^8/f_{RTC}$ (7.8125 ms)
1	0	1	1	$2^9/f_{RTC}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{RTC}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{RTC}$ (62.5 ms)
1	1	1	×	$2^{12}/f_{RTC}$ (125 ms)

当 RINTE = 0 时，更改 ICT2, ICT1 和 ICT0。

RCLOE2*	RTCDIV 引脚输出控制
0	禁止 RTCDIV 引脚输出。
1	允许 RTCDIV 引脚输出。

RCKDIV	选择RTCDIV 引脚输出频率
0	RTCDIV 引脚输出 512 Hz.
1	RTCDIV 引脚输出 16.384 kHz.

注 不能同时允许 RCLOE0 和 RCLOE2。

注意事项 当 RTCDIV 引脚输出停止，在最多 2 个  $f_{RTC}$  时钟和进入低电平后输出继续。当进入高电平后，输出 512 Hz 和输出立刻停止时，产生至少 1 个  $f_{XT}$  时钟宽度的脉冲。

**(5) 副计数寄存器(RSUBC)**

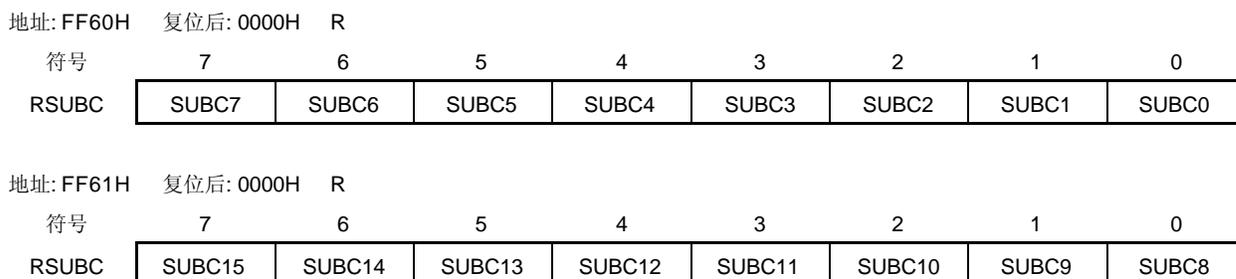
RSUB 寄存器是 16 位寄存器，是实时计数器 1 秒的计数标准时间。它可置 0000H ~ 7FFFH 的值和计数 32.768 kHz 时钟的 1 秒钟。

可以 16 位存储器操作指令设置 RSUBC。

复位信号的产生对寄存器清零 (0000H)。

- 注意事项**
1. 当使用 SUBCU 寄存器纠正，值将变为 8000H 或更多。
  2. 可由写入秒计数寄存器产生复位将该寄存器清零。
  3. 在操作期间读取该寄存器，因为读出值改变，不能保证读出值。

图 9-6. 副计数寄存器(RSUBC)的格式

**(6) 秒计数寄存器 (SEC)**

SEC 寄存器是 8 位寄存器，为 0~59 (十进制) 值和显示计数的秒值。

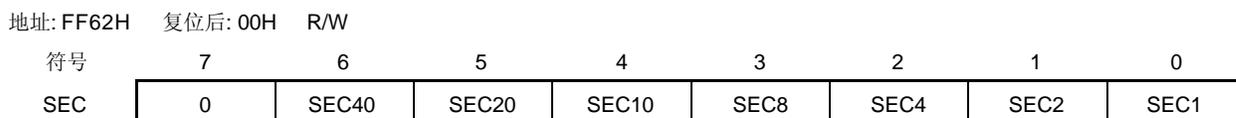
当副计数器溢出它将开始计数。

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。以 BCD 码设置 0~59 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

可以 8 位存储器操作指令设置 SEC。

复位信号的产生对寄存器清零 (00H)。

图 9-7. 秒计数寄存器 (SEC)的格式



**(7) 分钟计数寄存器 (MIN)**

MIN 寄存器是 8 位寄存器，为 0~59（十进制）值和显示计数的分钟值。

当秒计数器溢出它将开始计数。

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。以 BCD 码设置 0~59 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

可以 8 位存储器操作指令设置 MIN。

复位信号的产生对寄存器清零（00H）。

图 9-8. 分钟计数寄存器 (MIN)的格式

地址: FF63H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

**(8) 小时计数寄存器(HOUR)**

HOUR 寄存器是 8 位寄存器，为 0~23 或 1~12（十进制）值和显示计数的小时值。

当分钟计数器溢出它将开始计数。

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。以 BCD 码设置 00~23、01~12 或 21~23 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

可以 8 位存储器操作指令设置 HOUR。

复位信号的产生将寄存器置为 12H。

如果当 AMPM 位置为 1，复位后该寄存器的值清零(00H)。

图 9-9. 小时计数寄存器(HOUR)的格式

地址: FF64H 复位后: 12H R/W

符号	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

**注意事项** 当 AMPM = 0 (如果选择 12 小时系统)，HOUR 的第 5 位 (HOUR20) 显示 AM(0)/PM(1)。

**(9) 日计数寄存器 (DAY)**

HOUR 寄存器是 8 位寄存器，为 31（十进制）值和显示计数的天数值。

当小时计数器溢出它将开始计数。

计数器计数如下。

- 01 ~ 31 (一月、三月、五月、七月、八月、十月、十二月)
- 01 ~ 30 (四月、六月、九月、十一月)
- 01 ~ 29 (二月、闰年)
- 01 ~ 28 (二月、正常年)

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。以 BCD 码设置 00~31 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

可以 8 位存储器操作指令设置 DAY。

复位信号的产生将寄存器置为 01H。

图 9-10. 日计数寄存器 (DAY)的格式

地址: FF66H 复位后: 01H R/W

符号	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

**(10) 星期计数寄存器(WEEK)**

WEEK 寄存器是 8 位寄存器，为 0~6（十进制）值和显示计数的星期值。

当日计数器溢出它将开始计数。

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。以 BCD 码设置 00~06 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

可以 8 位存储器操作指令设置 WEEK。

复位信号的产生将寄存器置为 00H。

图 9-11. 星期计数寄存器(WEEK)的格式

地址: FF65H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

**(11) 月计数寄存器 (MONTH)**

MONTH 寄存器是 8 位寄存器，为 1~12（十进制）值和显示计数的月值。

当日计数器溢出它将开始计数。

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。以 BCD 码设置 01~12 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

可以 8 位存储器操作指令设置 MONTH。

复位信号的产生将寄存器置为 01H。

图 9-12. 月计数寄存器 (MONTH)的格式

地址: FF67H 复位后: 01H R/W

符号	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

**(12) 年计数寄存器 (YEAR)**

YEA 寄存器是 8 位寄存器，为 0~99（十进制）值和显示计数的月值。

当月计数器溢出它将开始计数。

计数器计数如下。

00, 04, 08, ..., 92 和 96 显示闰年

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。以 BCD 码设置 00~99 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

可以 8 位存储器操作指令设置 YEAR。

复位信号的产生将寄存器置为 00H。

图 9-13. 年计数寄存器 (YEAR)的格式

地址: FF68H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

**(13) 钟表计数寄存器 (SUBCUD)**

该寄存器用作纠正副计数器(RSUBC)的计数值。

可以 8 位存储器操作指令设置 SUBCUD。

复位信号的产生将寄存器置为 00H。

图 9-14. 钟表计数寄存器 (SUBCUD)的格式

地址: FF82H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0
DEV	设置钟表纠错时间							
0	当秒数字为 00, 20 或 40 时纠正钟表错误。							
1	仅当秒数字为 00 时纠正钟表错误。							
F6	设置钟表纠错方法							
0	由 $\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2$ 增大							
1	由 $\{(/F5, /F4, /F3, /F2, /F1, /F0) + 1\} \times 2$ 减小							
当 (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *) 时, 钟表错误不能修正。 /F5 ~ /F0 相应的位反向值(000011 当 111100)。								

**(14) 闹表分钟寄存器 (ALARMWM)**

该寄存器用作设置闹表分钟。

可以 8 位存储器操作指令设置 ALARMWM。

复位信号的产生将寄存器置为 00H。

**注意事项** 以 BCD 码设置 00 ~ 59 十进制数到该寄存器。如果设置的值超出范围, 闹表将不检测。

图 9-15. 闹表分钟寄存器 (ALARMWM)的格式

地址: FF86H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

**(15) 闹表小时寄存器(ALARMWH)**

该寄存器用作设置闹表小时。

可以 8 位存储器操作指令设置 ALARMWH。

复位信号的产生将寄存器置为 12H。

**注意事项** 以 BCD 码设置 00 ~ 23、01 ~ 12 或 21 ~ 32 十进制数到该寄存器。如果设置的值超出范围，闹表将不检测。

图 9-16. 闹表小时寄存器(ALARMWH)的格式

地址: FF87H 复位后: 12H R/W

符号	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

**注意事项** 如果 AMPM = 0 (选择 12 小时系统)，ALARMWH 的第 5 位 (WH20) 显示 AM(0)/PM(1)。

**(16) 闹表星期寄存器(ALARMWW)**

该寄存器用作设置闹表数据。

可以 8 位存储器操作指令设置 ALARMWW。

复位信号的产生将寄存器置为 00H。

**注意事项** 以 BCD 码设置 00 ~ 23、01 ~ 12 或 21 ~ 32 十进制数到该寄存器。如果设置的值超出范围，闹表将不检测。

图 9-17. 闹表星期寄存器(ALARMWW)的格式

地址: FF88H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

以下是设置闹表的示例。

闹表时间	日							12 小时显示				24 小时显示			
	星期日	星期一	星期二	星期三	星期四	星期五	星期六	小时 10	小时 1	分钟 10	分钟 1	小时 10	小时 1	分钟 10	分钟 1
	W W 0	W W 1	W W 2	W W 3	W W 4	W W 5	W W 6								
每天 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
每天 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
每天 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
星期一到星期五 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
星期日 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
星期一、星期三、星期五 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

## 9.4 实时计数器的操作

## 9.4.1 启动实时计数器的操作

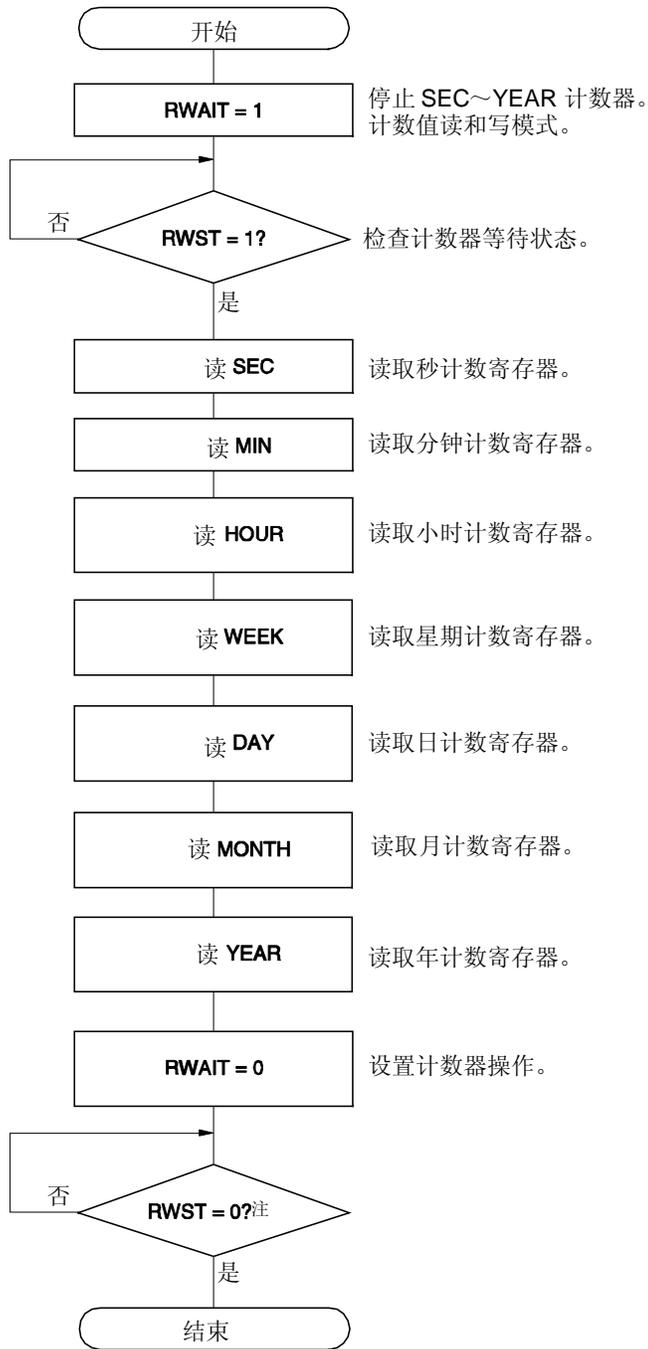
图 9-18. 启动实时计数器的操作流程



## 9.4.2 读/写实时计数器

当  $RWAIT = 1$  时，读出或写入计数器。

图 9-19. 读出实时计数器的操作流程



注 设置 STOP 模式前必须置  $RWST = 0$ 。

注意事项 在 1 秒内完成置  $RWAIT$  为 1 到清  $RWAIT$  为 0 的连续操作。

备注 SEC, MIN, HOUR, WEEK, DAY, MONTH 和 YEAR 将顺次读出。  
不用设置所有的寄存器，仅部分寄存器可被读出。

图 9-20. 写入实时计数器的操作流程



注 设置 STOP 模式前必须置 RWST = 0。

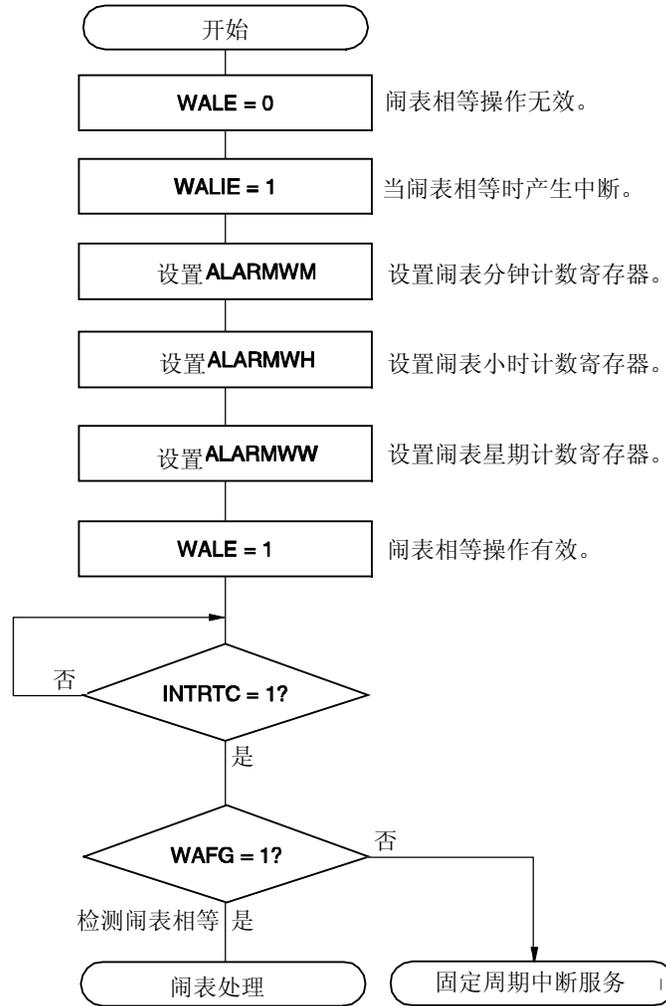
注意事项 在 1 秒内完成置 RWAIT 为 1 到清 RWAIT 为 0 的连续操作。

备注 SEC, MIN, HOUR, WEEK, DAY, MONTH 和 YEAR 将顺次读出。  
不用设置所有的寄存器，仅部分寄存器可被读出。

## 9.4.3 设置实时计数器闹表

当  $WALE = 0$  时，设置闹表时间。

图 9-21. 闹表设置流程图



## 备注

1. 可以以任意顺序写入 ALARMWM, ALARMWH 和 ALARMWW。
2. 固定周期中断和闹表相等中断使用相同的中断源 (INTRTC)。当同时使用这两种类型中断时，通过检查固定周期中断状态标志(RIFG)和闹表检测状态标志(WAFG)，来判断 INTRTC 产生。

## 第十章 看门狗定时器

### 10.1 看门狗定时器的功能

看门狗定时器使用内部低速振荡时钟。

看门狗定时器用于检测不期望出现的程序循环。如果检测到一个程序循环，将产生一个内部复位信号。出现以下情况时检测程序循环。

- 如果看门狗定时器计数器溢出
- 如果对看门狗定时器允许寄存器(WDTE)执行 1 位操作指令。
- 如果将“ACH”以外的数据写入 WDTE
- 如果在窗口关闭期间将数据写入 WDTE
- 如果从没有通过 IMS 和 IXS 寄存器设置的区域获取指令(在 CPU 挂起时检查是否有无效校验)
- 如果因为执行 1 条读/写指令，CPU 访问了没有通过 IMS 和 IXS 寄存器设置的区域(FB00H ~ FFFFH 除外) (CPU 程序循环期间检测是否有非法访问)

当由看门狗定时器产生复位时，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，可以参阅 **第二十章 复位功能**。

## 10.2 看门狗定时器的配置

看门狗定时器包含以下硬件。

表 10-1. 看门狗定时器的配置

项目	配置
控制寄存器	看门狗定时器允许寄存器 (WDTE)

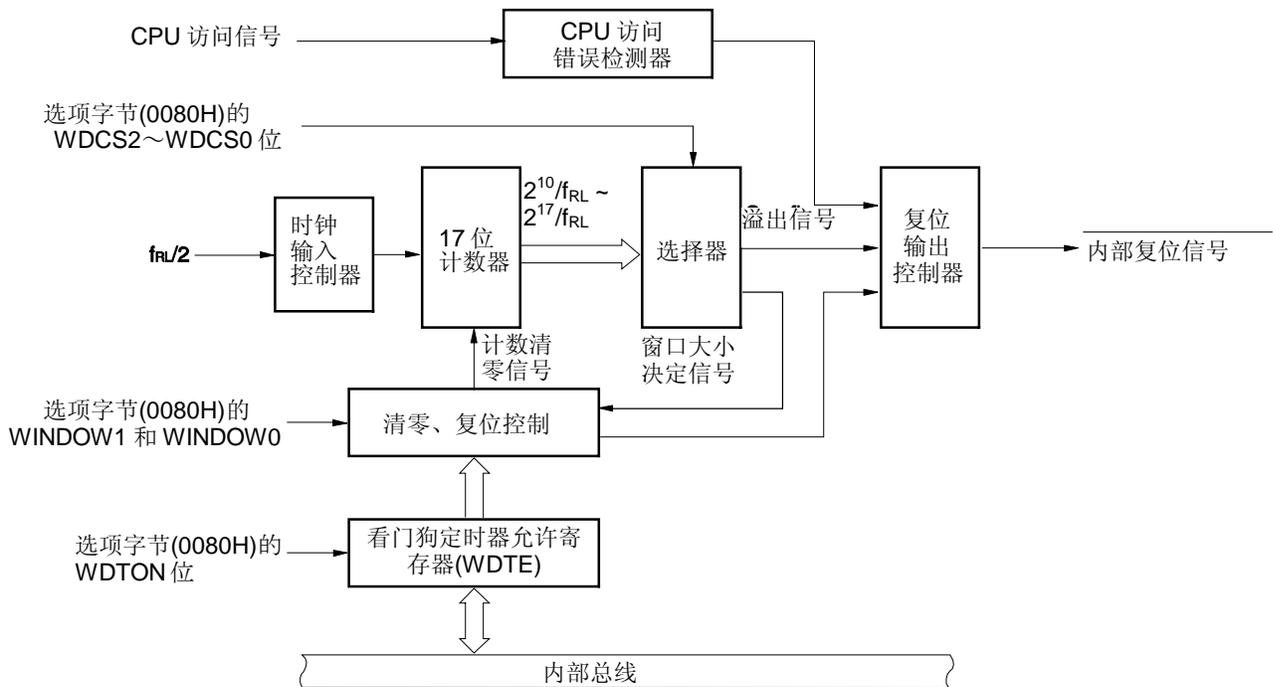
由选项字节设置如何控制计数器的操作、溢出时间和窗口打开周期。

表 10-2. 选项字节和看门狗定时器的设置

看门狗定时器的设置	选项字节 (0080H)
窗口打开周期	第 6 位和第 5 位 (WINDOW1, WINDOW0)
控制看门狗定时器的计数器操作	第 4 位 (WDTON)
看门狗定时器的溢出时间	第 3 位 ~ 第 1 位 (WDCS2 ~ WDCS0)

备注 需要了解选项字节的内容, 可参见 第二十三章 选项字节。

图 10-1. 看门狗定时器的框图



### 10.3 控制看门狗定时器的寄存器

看门狗定时器由看门狗定时器允许寄存器(WDTE)控制。

#### (1) 看门狗定时器允许寄存器(WDTE)

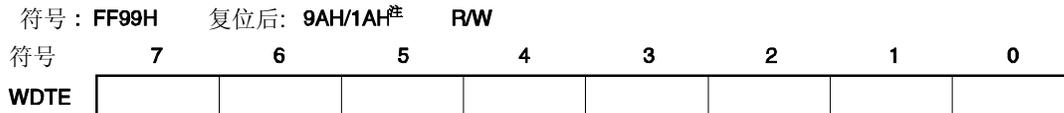
将 ACH 写入 WDTE 可以对看门狗定时器计数器清零并再次开始计数操作。

可以由 8 位存储器操作指令设置该寄存器。

复位信号的产生将该寄存器设置为 9AH 或 1AH<sup>注</sup>。

图 10-2. 看门狗定时器允许寄存器 (WDTE) 的格式

图 10-2. 看门狗定时器允许寄存器 (WDTE) 的格式



**注** WDTE 的复位后的值随选项字节(0080H)的 WDTON 位的设置值而变化。 如果要对看门狗定时器进行操作，设置 WDTON = 1。

WDTON 设置值	WDTE 复位值
0 (禁止看门狗定时器计数的操作)	1AH
1 (允许看门狗定时器计数的操作)	9AH

- 注意事项**
1. 如果写入 WDTE 的值不是 ACH，将产生一个内部复位信号。如果看门狗定时器的时钟源被停止，则当看门狗定时器的时钟源恢复操作时，将产生内部复位信号。
  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。如果看门狗定时器的时钟源被停止，当提供给看门狗定时器的时钟源恢复操作时，将产生内部复位信号。
  3. 从 WDTE 读取的值为 9AH/1AH（与写入值（ACH）不同）。

## 10.4 看门狗定时器的操作

### 10.4.1 看门狗定时器的控制操作

1. 当使用看门狗定时器时，由选项字节(0080H)指定其操作。

- 通过设置选项字节(0080H)的第 4 位(WDTON)=1，允许看门狗定时器的计数操作(复位释放后计数器开始计数)(如需了解详细信息，可参见 第二十三章)。

WDTON	看门狗定时器计数器的操作控制/非法访问检测
0	禁止计数器操作 (复位后停止计数)，禁止非法访问检测操作
1	允许计数器操作 (复位后开始计数)，允许非法访问检测操作

- 通过使用选项字节 (0080H) 的第 3 位~第 1 位 (WDCS2 ~ WDSC0) 设置溢出时间 (如需了解详细信息，参见 11.4.2 和 第二十三章)。
- 通过使用选项字节 (0080H) 的第 6 位和第 5 位 (WINDOW1 和 WINDOW0) 设置窗口打开周期 (如需了解详细信息，参见 11.4.3 和 第二十三章)。

- 复位释放后，看门狗定时器开始计数。
- 在看门狗定时器开始计数后且在达到选项字节设置的溢出时间之前，将“ACH”写入 WDTE，看门狗定时器将被清零并再次开始计数。
- 在这之后，在复位释放后的第二次或之后的窗口打开周期期间写 WDTE。如果在窗口关闭期间写 WDTE，则会产生内部复位信号。
- 如果已经历了溢出时间还没有将“ACH”写入 WDTE，则会产生内部复位信号。  
出现以下情况时会产生内部复位信号。
  - 如果对看门狗定时器允许寄存器(WDTE)执行位操作指令时。
  - 如果将“ACH”以外的数据写入 WDTE。
  - 如果从没有通过 IMS 和 IXS 寄存器设置的区域获取指令(在 CPU 程序循环期间检查是否有无效校验)
  - 如果因为执行 1 条读/写指令，CPU 访问了没有通过 IMS 和 IXS 寄存器设置的区域(FB00H ~ FFFFH 除外)(CPU 程序循环期间检测是否有非法访问)

- 注意事项**
- 复位释放后对 WDTE 进行第一次写操作，将清零看门狗定时器，如果这一操作是在到达溢出时间之前进行的而不管写操作时序，看门狗定时器再次开始计数。
  - 如果通过将“ACH”写入 WDTE 来对看门狗定时器清零，则实际的溢出时间可能与由选项字节设置的溢出时间不同，最高可达  $2/f_{RL}$  秒。
  - 在计数值溢出(FFFFH)之前，可将看门狗定时器立即清零。

注意事项 4. 在 HALT 和 STOP 模式下看门狗定时器的操作依据选项字节第 0 位(LSROSC)的设置值的不同而不同, 如下所示。

	LSROSC = 0 (可由软件停止内部低速振荡器)	LSROSC = 1 (不能停止内部低速振荡器)
HALT 模式	看门狗定时器操作停止	看门狗定时器操作继续
STOP 模式		

如果 LSROSC = 0, 则在 HALT 或 STOP 模式释放后看门狗定时器恢复计数。此时, 不对计数器清零, 而从计数器停止时的值开始计数。

如果在 LSROSC = 0 时通过设置 LSRSTOP(内部振荡模式寄存器(RCM)的第 1 位=1)停止内部低速振荡器的振荡, 则看门狗定时器停止操作。此时不对计数器清零(0)。

5. 在闪存自编程和 EEPROM<sup>TM</sup> 仿真期间看门狗定时器不停止操作。在处理期间中断响应被延迟。设置溢出时间和窗口大小时, 应考虑延迟。

#### 10.4.2 看门狗定时器溢出时间的设置

通过使用选项字节(0080H)的第 3 位~第 1 位(WDCS2 ~ WDCS0), 设置看门狗定时器的溢出时间。

如果出现溢出, 则将产生内部复位信号。在到达溢出时间之前且在窗口打开周期内通过将“ACH”写入 WDTE, 可对当前计数值清零, 且看门狗定时器再次开始计数。

溢出时间设置如下所示。

表 10-3. 看门狗定时器溢出时间的设置

WDCS2	WDCS1	WDCS0	看门狗定时器的溢出时间
0	0	0	$2^{10}/f_{RL}$ (3.88 ms)
0	0	1	$2^{11}/f_{RL}$ (7.76 ms)
0	1	0	$2^{12}/f_{RL}$ (15.52 ms)
0	1	1	$2^{13}/f_{RL}$ (31.03 ms)
1	0	0	$2^{14}/f_{RL}$ (62.06 ms)
1	0	1	$2^{15}/f_{RL}$ (124.12 ms)
1	1	0	$2^{16}/f_{RL}$ (248.24 ms)
1	1	1	$2^{17}/f_{RL}$ (496.48 ms)

注意事项 1. 禁止出现 WDCS2 = WDCS1 = WDCS0 = 0 且 WINDOW1 = WINDOW0 = 0 的情况。

2. 在闪存自编程和 EEPROM 仿真期间看门狗定时器不停止操作。在处理期间中断响应被延迟。设置溢出时间和窗口大小时, 应考虑延迟。

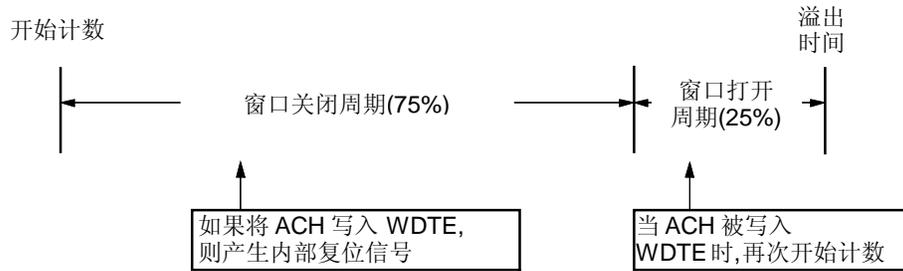
备注 1.  $f_{RL}$ : 内部低速振荡时钟频率  
2. ( ):  $f_{RL} = 264 \text{ kHz (MAX.)}$

### 10.4.3 看门狗定时器窗口打开周期的设置

通过使用选项字节(0080H)的第6位和第5位(WINDOW1, WINDOW0), 设置看门狗定时器的窗口打开周期。窗口使用情况描述如下。

- 如果在窗口打开期间将“ACH”写入 WDTE, 则对看门狗定时器清零并再次开始计数。
- 在窗口关闭期间即使将“ACH”写入 WDTE, 也会检测到异常情况, 并产生内部复位信号。

**举例:** 如果窗口打开周期为 25%



**注意事项** 复位释放后对 WDTE 进行第一次写操作, 将清零看门狗定时器, 如果这一操作是在到达溢出时间之前进行的而不管写操作时序, 看门狗定时器再次开始计数。

窗口打开周期设置如下所示。

表 10-4. 看门狗定时器窗口打开周期的设置

WINDOW1	WINDOW0	看门狗定时器的窗口打开周期
0	0	25%
0	1	50%
1	0	75%
1	1	100%

- 注意事项**
1. 禁止出现  $WDCS2 = WDCS1 = WDCS0 = 0$  且  $WINDOW1 = WINDOW0 = 0$  的情况。
  2. 在闪存自编程和 EEPROM 仿真期间看门狗定时器不停止操作。在处理阶段中断响应被延迟。设置溢出时间和窗口大小时, 应考虑延迟。

**备注** 如果将溢出时间设置为  $2^{10}/f_{RL}$ ，则窗口关闭时间和打开时间如下所示。

	窗口打开周期的设置			
	25%	50%	75%	100%
窗口关闭时间	0 ~ 3.56 ms	0 ~ 2.37 ms	0 ~ 0.119 ms	无
窗口打开时间	3.56 ~ 3.88 ms	2.37 ~ 3.88 ms	0.119 ~ 3.88 ms	0 ~ 3.88 ms

<当窗口打开周期为 25%>

- 溢出时间:

$$2^{10}/f_{RL} (\text{MAX.}) = 2^{10}/264 \text{ kHz} (\text{MAX.}) = 3.88 \text{ ms}$$

- 窗口关闭时间:

$$0 \text{ to } 2^{10}/f_{RL} (\text{MIN.}) \times (1 - 0.25) = 0 \sim 2^{10}/216 \text{ kHz} (\text{MIN.}) \times 0.75 = 0 \sim 3.56 \text{ ms}$$

- 窗口打开时间:

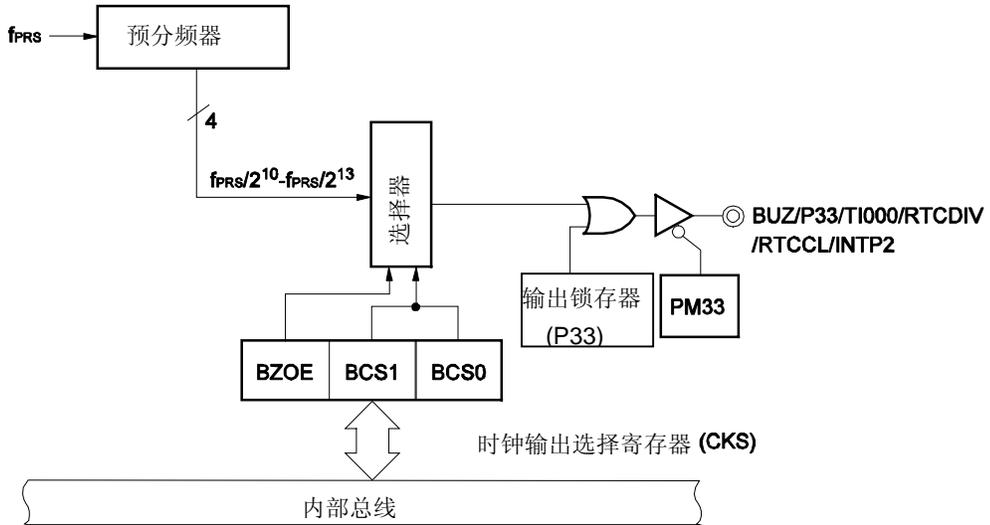
$$2^{10}/f_{RL} (\text{MIN.}) \times (1 - 0.25) \sim 2^{10}/f_{RL} (\text{MAX.}) = 2^{10}/216 \text{ kHz} (\text{MIN.}) \times 0.75 \sim 2^{10}/264 \text{ kHz} (\text{MAX.}) \\ = 3.56 \sim 3.88 \text{ ms}$$

11.1 蜂鸣器输出控制器的功能

蜂鸣器输出使用 CKS 选定的频率输出方波。

图 11-1 是蜂鸣器输出控制器的方框图。

图 11-1. 蜂鸣器输出控制器的方框图



## 11.2 蜂鸣器输出控制器的配置

蜂鸣器输出控制器包括下列硬件。

表 11-1. 蜂鸣器输出控制器的配置

名称	配置
控制寄存器	时钟输出选择寄存器 (CKS) 端口模式寄存器 3 (PM3) 端口寄存器 3 (P3)

## 11.3 控制蜂鸣器输出控制器的寄存器

下列两个寄存器用来控制蜂鸣器输出控制器。

- 时钟输出选择寄存器 (CKS)
- 端口模式寄存器 3 (PM3)

### (1) 时钟输出选择寄存器 (CKS)

此寄存器设置蜂鸣器(BUZ)频率输出的允许/禁止, 并设置输出时钟。

可用 1-位或者 8-位存储器操作指令设置此寄存器。

复位信号将 CKS 清零 00H。

图 11-2. 时钟选择输出寄存器的格式 (CKS)

地址: FF40H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	0
CKS	BZOE	BCS1	BCS0	0	0	0	0	0

BZOE	BUZ 输出允许/禁止规定
0	时钟分频电路操作停止。BUZ 固定为低电平。
1	时钟分频电路操作允许。BUZ 输出允许。

BCS1	BCS0	BUZ 输出时钟选择		
			f <sub>PRS</sub> = 5 MHz	f <sub>PRS</sub> = 10 MHz
0	0	f <sub>PRS</sub> /2 <sup>10</sup>	4.88 kHz	9.77 kHz
0	1	f <sub>PRS</sub> /2 <sup>11</sup>	2.44 kHz	4.88 kHz
1	0	f <sub>PRS</sub> /2 <sup>12</sup>	1.22 kHz	2.44 kHz
1	1	f <sub>PRS</sub> /2 <sup>13</sup>	0.61 kHz	1.22 kHz

**注意事项** 当蜂鸣器输出操作停止时设置 **BCS1** 和 **BCS0** (**BZOE = 0**)。

**备注** f<sub>PRS</sub>: 外围硬件时钟频率

**(2) 端口模式寄存器 3 (PM3)**

此寄存器用 1 位单元设置端口 3 为输入/输出。

当使用 P33/TI000/RTCDIV/RTCCL/BUZ/INTP2 引脚为蜂鸣器输出时, 将 PM33 清零并将 P33 的输出所存清零。

可用 1-位或者 8-位存储器操作指令设置。

复位信号将此寄存器设置为 FFH。

**图 11-3. 端口模式寄存器 3 的格式 (PM3)**

地址: FF23H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34	PM33	PM32	PM31	1

PM3n	P3n 引脚 I/O 模式选择 (n = 1 ~ 4)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

**11.4 蜂鸣器输出控制器的操作**

蜂鸣器频率按下列顺序输出。

- <1> 用时钟输出选择寄存器(CKS) (蜂鸣器输出禁止状态)的位 5 和位 6(BCS0,BCS1)选择蜂鸣器输出频率。
- <2> 设置 CKS 的位 7(BZOE)为 1 允许蜂鸣器输出。

## 第十二章 10位逐次逼近型 A/D 转换器 (仅对 $\mu$ PD78F041x)

### 12.1 10位逐次逼近型 A/D 转换器功能

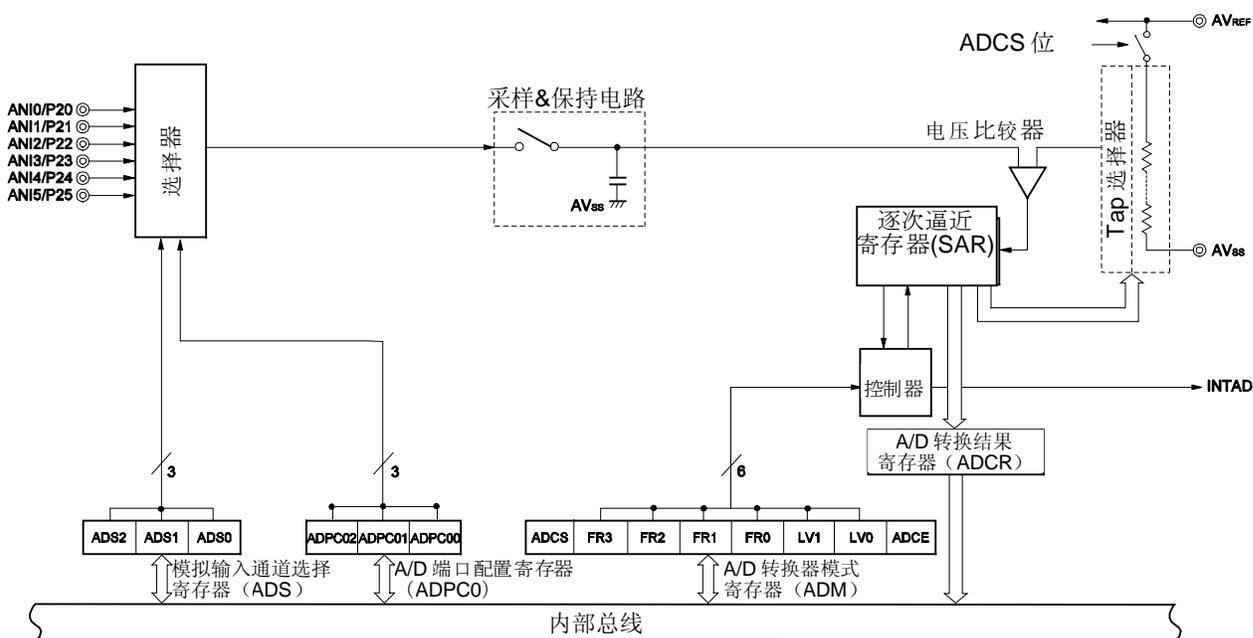
10位逐次逼近型 A/D 转换器用于将模拟输入信号转换为数字信号，最多可由 6 个通道(ANI0 ~ ANI5)组成，具有 10 位分辨率。

A/D 转换器有以下功能。

- 10 位分辨率 A/D 转换

从模拟输入 ANI0~ANI5 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，都可以产生一个中断请求(INTAD)。

图 12-1. 10位逐次逼近型 A/D 转换器的框图



## 12.2 10-位逐次逼近型 A/D 转换器的配置

10-位逐次逼近型 A/D 转换器包括以下硬件。

### (1) ANI0 ~ ANI5 引脚

这些是 6 通道 10-位逐次逼近型 A/D 转换器的模拟输入引脚。用于输入模拟信号，以便转换成数字信号。除了那些被选为模拟输入的引脚外，其它引脚均可用作 I/O 端口引脚。

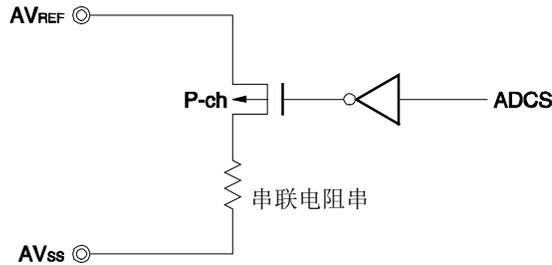
### (2) 采样&保持电路

采样&保持电路在 A/D 转换器启动时用于采样由选择器选择的模拟输入引脚的输入电压，并在 A/D 转换期间保持采样到的电压值。

### (3) 串联电阻串

串联电阻串被连接在  $AV_{REF}$  和  $AV_{SS}$  之间，并产生一个电压，与采样到的电压值进行比较。

图 12-2. 串联电阻串的电路配置



### (4) 电压比较器

电压比较器用于比较采样到的电压值与串联电阻串的输出电压。

### (5) 逐次逼近寄存器 (SAR)

该寄存器用于转换电压比较器所比较的结果，并从最高有效位(MSB)开始转换结果。

当电压被转换成数字值，并写入最低有效位时(A/D 转换结束)，将 SAR 的内容传送到 A/D 转换结果寄存器(ADCR)中。

### (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 转换结果从逐次逼近寄存器(SAR)传送至该寄存器中。并由该寄存器将转换结果保存在它的高 10 位中(低 6 位恒为 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 转换结果从逐次逼近寄存器传送至该寄存器中。并将转换结果保存在 ADCRH 的高 8 位中。

**注意事项** 当从 ADCR 和 ADCRH 读取数据时, 会产生一个等待周期。当 CPU 工作在副系统时钟, 且外围硬件时钟处于停止状态时, 不要从 ADCR 和 ADCRH 读取数据。需要了解详细信息, 可参阅 第二十九章 等待注意事项。

**(8) 控制器**

该电路用于控制模拟输入信号被转换成数字信号的转换时间, 并启动和停止转换操作。当 A/D 转换结束时, 该控制器产生中断 INTAD。

**(9) AVREF 引脚**

该引脚为 A/D 转换器输入一个模拟供电/参考电压。当端口 2 被用作数字端口时, 该引脚的电压应与 V<sub>DD</sub> 的电压保持一致。

根据 AVREF 和 AVSS 的电压, 将输入到 ANI0 ~ ANI4 的信号转换成数字信号。

**(10) AVss 引脚**

这是 A/D 转换器的地电位引脚。即便没有使用 A/D 转换器, 该引脚的电压也应与 V<sub>SS</sub> 的电压保持一致。

**(11) A/D 转换器模式寄存器(ADM)**

该寄存器用于设置被转换的模拟输入信号的转换时间, 并启动或停止转换操作。

**(12) A/D 端口配置寄存器(ADPC)**

该寄存器用于将 ANI0/P20 ~ ANI5/P25 引脚切换成 10-位逐次逼近型 A/D 转换器的模拟输入或数字 I/O 端口。

**(13) 模拟输入通道选择寄存器 (ADS)**

该寄存器用来选择输入模拟电压(将被转换成数字信号)的端口。

**(14) 端口模式寄存器 2 (PM2)**

该寄存器用于将 ANI0/P20 ~ ANI4/P25 引脚切换为输入或输出。

**(15) 端口功能寄存器 2 (PF2)**

此寄存器将 ANI0/P20 ~ ANI5/P25 引脚转换成 I/O 端口, A/D 转换器的模拟输入, 或者 segment 输出。

### 12.3 10-位逐次逼近型 A/D 转换器使用的寄存器

A/D 转换器使用以下七个寄存器。

- A/D 转换器模式寄存器(ADM)
- A/D 端口配置寄存器 0(ADPC0)
- 模拟输入通道选择寄存器(ADS)
- 端口功能寄存器 2 (PF2)
- 端口模式寄存器 2 (PM2)
- 10 位 A/D 转换结果寄存器(ADCR)
- 8 位 A/D 转换结果寄存器(ADCRH)

#### (1) A/D 转换器模式寄存器 (ADM)

该寄存器设置模拟输入的 A/D 转换时间，并启动/停止转换。

可由 1 位或 8 位存储器操作指令设置 ADM。

复位信号产生将该寄存器清零(00H)。

图 12-3. A/D 转换器模式寄存器 (ADM)的格式

地址: FF8DH    复位后: 00H    R/W

符号	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	FR3 <sup>注1</sup>	FR2 <sup>注1</sup>	FR1 <sup>注1</sup>	FR0 <sup>注1</sup>	LV1 <sup>注1</sup>	LV0 <sup>注1</sup>	ADCE

ADCS	A/D 转换操作控制
0	停止转换操作
1	允许转换操作

ADCE	比较器操作控制 <sup>注2</sup>
0	停止比较器的操作
1	允许比较器的操作

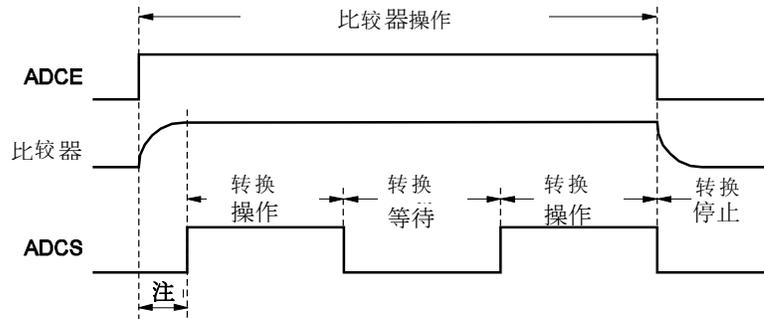
- 注
1. 如需了解 FR3~FR0、LV1、LV0 和 A/D 转换的详细信息，可参见 表 12-2 A/D 转换时间的选择。
  2. 比较器的的操作是由 ADCS 和 ADCE 控制的，从操作开始到操作稳定需要 1μs。因此，当 ADCE 被设置为 1 后，至少经过 1μs 的时间，再将 ADCS 置 1 时，此时的转换结果优先与第一次的转换结果。否则，忽略第一次转换的数据。

表 12-1. ADCS 和 ADCE 的设置

ADCS	ADCE	A/D 转换操作
0	0	停止状态 (不存在直流功耗)
0	1	转换等待模式 (比较器操作，只有比较器产生功耗)
1	0	转换模式 (停止比较器的操作 <sup>注</sup> )
1	1	转换模式 (比较器操作)

注 忽略第 1 次转换的数据。

图 12-4. 使用比较器时的时序图



注 从 ADCE 位上升到 ADCS 位下降所需时间必须至少为  $1 \mu\text{s}$ ，以便稳定内部电路。

- 注意事项
1. 在将 FR0 ~ FR3 位, LV1 和 LV0 位修改成其他值之前, A/D 转换必须停止。
  2. 如果对 ADM 赋值, 会产生 1 个等待周期。当 CPU 工作在副系统时钟, 且外围硬件时钟处于停止状态时, 不要对 ADM 赋值。需要了解详细信息, 可参阅 第二十九章 等待注意事项。

表 12-2. A/D 转换时间的选择

(1)  $2.7\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$

A/D 转换器模式寄存器 (ADM)						转换时间选择				转换时钟 ( $f_{AD}$ )
FR3	FR2	FR1	FR0	LV1	LV0	$f_{PRS} = 2\text{ MHz}$	$f_{PRS} = 8\text{ MHz}$	$f_{PRS} = 10\text{ MHz}$		
1	x	x	x	0	0	$352/f_{PRS}$	禁止设置	$44.0\ \mu\text{s}$	$35.2\ \mu\text{s}$	$f_{PRS}/16$
0	0	0	0	0	0	$264/f_{PRS}$	禁止设置	$33.0\ \mu\text{s}$	$26.4\ \mu\text{s}$	$f_{PRS}/12$
0	0	0	1	0	0	$176/f_{PRS}$		$22.0\ \mu\text{s}$	$17.6\ \mu\text{s}$	$f_{PRS}/8$
0	0	1	0	0	0	$132/f_{PRS}$		$16.5\ \mu\text{s}$	$13.2\ \mu\text{s}$	$f_{PRS}/6$
0	0	1	1	0	0	$88/f_{PRS}$		$44.0\ \mu\text{s}$	$11.0\ \mu\text{s}^{\#}$	$8.8\ \mu\text{s}^{\#}$
0	1	0	0	0	0	$66/f_{PRS}$	$33.0\ \mu\text{s}$	$8.3\ \mu\text{s}^{\#}$	$6.6\ \mu\text{s}^{\#}$	$f_{PRS}/3$
0	1	0	1	0	0	$44/f_{PRS}$	$22.0\ \mu\text{s}$	禁止设置	禁止设置	$f_{PRS}/2$
其它情况						禁止设置				

注 只有当  $4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$  时, 才能设置。

(2)  $2.3\text{ V} \leq AV_{REF} < 2.7\text{ V}$

A/D 转换器模式寄存器 (ADM)						转换时间选择				转换时钟 ( $f_{AD}$ )
FR3	FR2	FR1	FR0	LV1	LV0	$f_{PRS} = 2\text{ MHz}$	$f_{PRS} = 5\text{ MHz}$	$f_{PRS} = 8\text{ MHz}$		
0	0	0	0	0	1	$480/f_{PRS}$	禁止设置	禁止设置	$60.0\ \mu\text{s}$	$f_{PRS}/12$
0	0	0	1	0	1	$320/f_{PRS}$		$64.0\ \mu\text{s}$	$40.0\ \mu\text{s}$	$f_{PRS}/8$
0	0	1	0	0	1	$240/f_{PRS}$		$48.0\ \mu\text{s}$	$30.0\ \mu\text{s}$	$f_{PRS}/6$
0	0	1	1	0	1	$160/f_{PRS}$		$32.0\ \mu\text{s}$	禁止设置	$f_{PRS}/4$
0	1	0	0	0	1	$120/f_{PRS}$	$60.0\ \mu\text{s}$	禁止设置		$f_{PRS}/3$
0	1	0	1	0	1	$80/f_{PRS}$	$40.0\ \mu\text{s}$			$f_{PRS}/2$
其它情况						禁止设置				

注意事项 1. 根据以下条件设置转换时间

- $4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$ : 取样 + 完成转换时间 =  $5 \sim 40\ \mu\text{s}$   
( $f_{AD} = 0.6 \sim 3.6\text{ MHz}$ )
- $2.7\text{ V} \leq AV_{REF} < 4.0\text{ V}$ : 取样 + 完成转换时间 =  $10 \sim 40\ \mu\text{s}$   
( $f_{AD} = 0.6 \sim 1.8\text{ MHz}$ )
- $2.3\text{ V} \leq AV_{REF} < 2.7\text{ V}$ : 取样 + 完成转换时间 =  $25 \sim 75\ \mu\text{s}$   
( $f_{AD} = 0.6 \sim 1.48\text{ MHz}$ )

2. 当修改 FR3 ~ FR0, LV1 和 LV0 为其他值时, 要预先停止 A/D 转换一次 ( $ADCS = 0$ )。

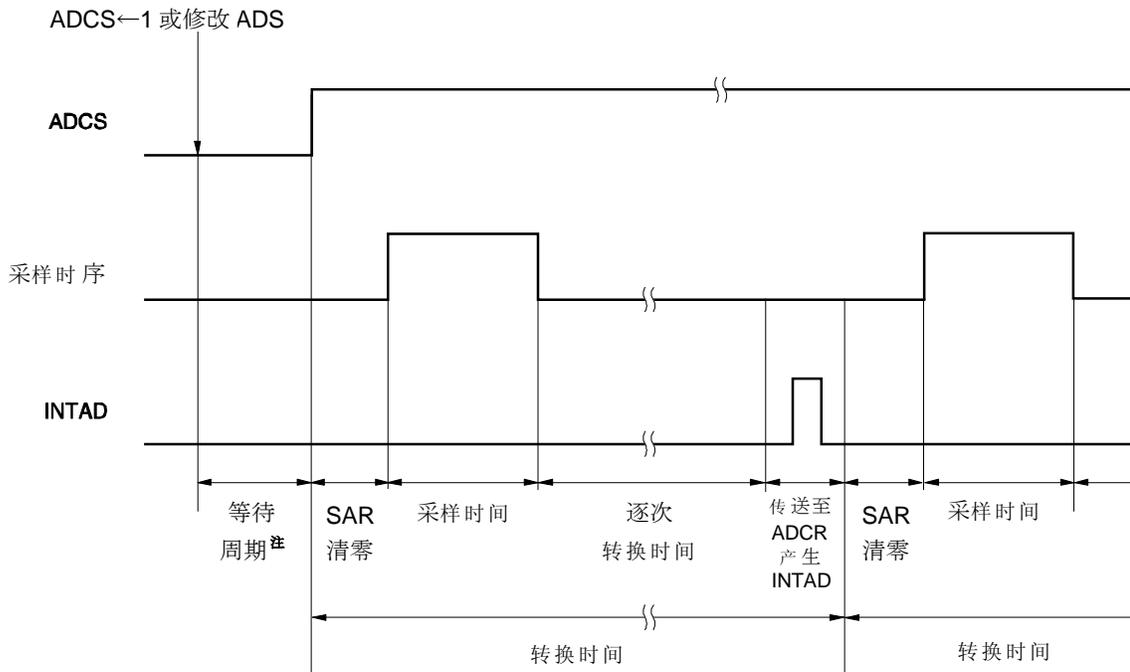
3. 当  $2.3\text{ V} \leq AV_{REF} < 2.7\text{ V}$  时, 修改 LV1 和 LV0 的默认值。

4. 上述的转换时间不包括时钟频率误差。在选择转换时间的时候, 要将时钟频率误差考虑进去。

备注

$f_{PRS}$ : 外围硬件时钟频率

图 12-5. A/D 转换器采样和 A/D 转换时序



注 等待周期的详细信息参见 第二十九章 等待注意事项。

(2) 10 位 A/D 转换结果寄存器 (ADCR)

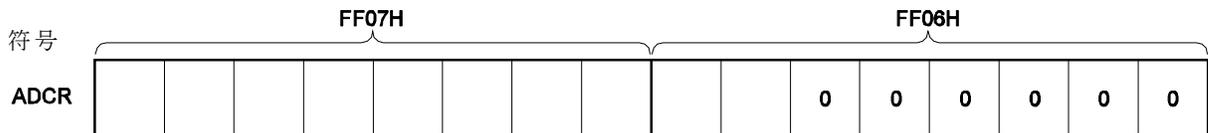
这是一个 16 位寄存器，用来存储 A/D 转换结果。寄存器的低 6 位恒为 0。每次 A/D 转换结束时，将转换结果从逐次逼近寄存器传送至 ADCR 中。转换结果的高 8 位存储在 FF07H 中，而转换结果的低 2 位存储在 FF06H 的高 2 位中。

可由 16 位存储器操作指令读取 ADCR。

复位信号产生将该寄存器清零(0000H)。

图 12-6. 10 位 A/D 转换结果寄存器(ADCR)的格式

地址:FF06H, FF07H      复位后: 0000H R

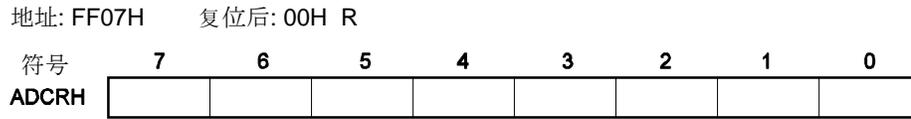


- 注意事项
- 在对 A/D 转换器模式寄存器(ADM)和模拟输入通道选择寄存器(ADS)和 A/D 端口配置寄存器(ADPC0)赋值时，ADCR 的内容可能是不确定的。转换结束后先读取转换结果，再对 ADM, ADS 和 ADPC0 赋值。如果没有按照上述时序，可能会读取到不正确的转换结果。
  - 如果从 ADCR 读取数据，会产生一个等待周期。当 CPU 工作在副系统时钟，且外围硬件时钟处于停止状态时，不要从 ADCR 读取数据。需要了解详细信息，可参阅 第二十九章 等待注意事项。

**(3) 8 位 A/D 转换结果寄存器 (ADCRH)**

这是一个 8 位寄存器，用来存储 A/D 转换结果。可以存储 10 位分辨率的高 8 位。  
 可由 8 位存储器操作指令读取 ADCRH。  
 复位信号产生将该寄存器清零(00H)。

图 12-7. 8 位 A/D 转换结果寄存器 (ADCRH) 的格式



- 注意事项**
1. 在对 A/D 转换器模式寄存器(ADM)、模拟输入通道选择寄存器(ADS)和 A/D 端口配置寄存器(ADPC0)赋值时，ADCRH 的内容可能不确定。转换结束后先读取转换结果，再对 ADM、ADS 和 ADPC0 赋值。如果没有按照上述时序，可能会读取到不正确的转换结果。
  2. 如果从 ADCRH 读取数据，会产生一个等待周期。当 CPU 工作在副系统时钟，且外围硬件时钟处于停止状态时，不要从 ADCRH 读取数据。需要了解详细信息，可参阅 第二十九章 等待注意事项。

**(4) 模拟输入通道选择寄存器 (ADS)**

该寄存器用来选择用于 A/D 转换的模拟电压输入通道。

可由 1 位或 8 位存储器操作指令设置 ADS。

复位信号产生将该寄存器清零(00H)。

图 12-8. 模拟输入通道选择寄存器 (ADS)的格式

地址: FF8EH      复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADS	0	0	0	0	0	ADS2	ADS1	ADS0

ADS2	ADS1	ADS0	模拟输入通道的选择
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	禁止设置
1	1	1	

- 注意事项**
1. 第 3 位 ~ 第 7 位必须清零。
  2. 使用端口模式寄存器 2(PM2)设置一个用于 A/D 转换通道的端口为输入模式。
  3. 不要同时使用 ADPC 和 ADS 设置一个引脚作为数字 I/O 引脚。
  4. 如果对 ADS 赋值, 会产生一个等待周期。当 CPU 工作在副系统时钟, 且外围硬件时钟处于停止状态时, 不要对 ADS 赋值。需要了解详细信息, 可参阅 第二十九章 等待注意事项。

**(5) A/D 端口配置寄存器 (ADPC0)**

这个寄存器用于将 ANI0/P20 ~ ANI4/P25 引脚切换为 A/D 转换器的模拟输入(16-位  $\Delta\Sigma$  型 A/D 转换器或者 10-位逐次逼近型 A/D 转换器的模拟输入)或者数字 I/O 端口。

可由 1 位或 8 位存储器操作指令设置 ADPC0。

复位信号产生将该寄存器清零(08H)。

图 12-9. A/D 端口配置寄存器(ADPC0)的格式

地址: FF8FH 复位后: 08H R/W

符号	7	6	5	4	3	2	1	0
ADPC0	0	0	0	0	0	ADPC02	ADPC01	ADPC00

ADPC02	ADPC01	ADPC00	模拟输入(A)/数字 I/O(D)的切换					
			P25 /ANI5	P24 /ANI4	P23 /ANI3	P22 /AN2	P21 /ANI1	P20 /ANI0
0	0	0	A	A	A	A	A	A
0	0	1	A	A	A	A	A	D
0	1	0	A	A	A	A	D	D
0	1	1	A	A	A	D	D	D
1	0	0	A	A	D	D	D	D
1	0	1	A	D	D	D	D	D
1	1	0	D	D	D	D	D	D
其它情况			禁止设置					

- 注意事项
1. 用端口模式寄存器 2(PM2)设置一个用于 A/D 转换通道的端口为输入模式。
  2. 不要同时使用 ADPC0 和 ADDS1, 以及 ADDS0 设置一个引脚作为数字 I/O 引脚。
  3. 如果将数据写入 ADPC0, 会产生一个等待周期。当 CPU 工作在副系统时钟, 且外围硬件时钟处于停止状态时, 不要将数据写入 ADPC0。需要了解详细信息, 可参阅 第二十九章 等待注意事项。
  4. 如果 ANI0/P20/SEG21 ~ ANI5/P25/SEG16 引脚通过 PF2 寄存器设置为 segment 输出, 输出就转换成 segment 输出, 而不管 ADPC0 设置为何。

**(6) 端口模式寄存器 2 (PM2)**

在使用 ANI0/P20 ~ ANI5/P25 引脚作为模拟输入端口时, 将 PM20 ~ PM25 设为 1。此时 P20 ~ P25 的输出锁存器的内容为 0 或 1。

如果将 PM20 ~ PM25 设为 0, 则它们不能用作模拟输入端口引脚。

可由 1 位或 8 位的存储器操作指令来设置 PM2。

复位信号产生将该寄存器设置为 FFH。

图 12-10. 端口模式寄存器 2 (PM2)的格式

地址: FF8FH 复位后: 08H R/W



根据 PF2、ADPC0、PM2 和 ADS 的设置, ANI0/P25 ~ ANI5/P25 引脚功能如下所示。

表 12-3. ANI0/P20 ~ ANI5/P25 引脚功能的设置

PF2	ADPC0	PM2	ADS	P20/SEG21/ANI0 ~ P25/SEG16/ANI5 引脚
数字/模拟选择	模拟输入选择	输入模式	不选择 ANI.	模拟输入 (不转换)
			选择 ANI.	模拟输入 (由逐次逼近型 A/D 转换器转换)
	数字 I/O 选择	输出模式	—	禁止设置
			—	数字输入
SEG 输出选择	—	—	—	数字输出
			—	Segment 输出

## 12.4 10-逐次逼近型 A/D 转换器的操作

### 12.4.1 A/D 转换器的基本操作

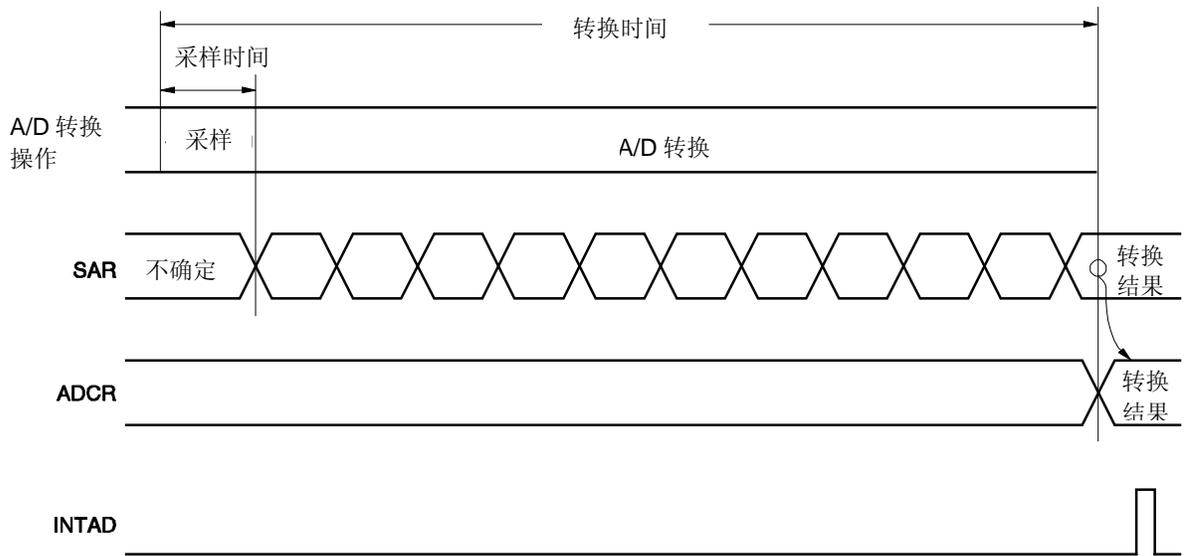
- <1> 把 A/D 转换器模式寄存器(ADM)的第 0 位(ADCE)置 1 启动比较器的操作。
- <2> 通过 A/D 端口配置寄存器(ADPC0)设置通道, 用于 A/D 转换的模拟输入, 并且通过端口模式寄存器 2(PM2)设置为输入模式。
- <3> 根据 ADM 的第 6~1 位(FR3 ~ FR0、LV1 和 LV0)设置 A/D 转换时间。
- <4> 通过模拟输入通道选择寄存器(ADS)选择一个通道用于 A/D 转换
- <5> 设置 ADM 的第 7 位(ADCS)为 1, 启动转换操作。  
(<6> ~ <12> 的操作由硬件完成)
- <6> 由采样&保持电路对输入到已选中的模拟输入通道的电压进行采样。
- <7> 在经过一定时间的采样后, 采样&保持电路处于保持状态, 且在 A/D 转换操作结束前一直保持采样电压。
- <8> 设置逐次逼近寄存器(SAR)的第 9 位。通过分接选择器将串联电阻串的分接电压置为 $(1/2) AV_{REF}$ 。
- <9> 由电压比较器比较串联电阻串的分接电压与采样电压。如果模拟输入电压高于 $(1/2) AV_{REF}$ , 则 SAR 的 MSB = 1; 如果模拟输入电压低于 $(1/2) AV_{REF}$ , 则 SAR 的 MSB = 0。
- <10> 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 位的预置值选择串联电阻串的分接电压, 具体描述如下。
  - 第 9 位 = 1:  $(3/4) AV_{REF}$
  - 第 9 位 = 0:  $(1/4) AV_{REF}$
 比较分接电压与采样电压, 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
- <11> 按此方式继续进行比较, 直至 SAR 的第 0 位。
- <12> 全部 10 位比较完成后, 在 SAR 中保留一个有效的数值结果, 然后将结果传送至 A/D 转换结果寄存器(ADCR, ADCRH)中, 并锁存。  
同时也会产生 A/D 转换结束中断请求 (INTAD)。
- <13> 反复执行步骤 <6> ~ <12>, 直至 ADCS 被清零(0)。  
将 ADCS 清零, 以停止 A/D 转换器操作。  
当 ADCE = 1 时, 若要重新启动 A/D 转换操作, 应从步骤<5>开始。当 ADCE = 0 时, 若要再次启动 A/D 转换操作, 设置 ADCE=1, 等待至少  $1 \mu s$ , 然后从步骤<5>开始操作。如要改变 A/D 转换的通道, 则从步骤<4>开始。

**注意事项** 必须确保 <1> ~ <5> 的操作时间至少为  $1 \mu s$ 。

**备注** 有两种类型的 A/D 转换结果寄存器可以使用。

- ADCR (16 位): 存储 10 位 A/D 转换值
- ADCRH (8 位): 存储 8 位 A/D 转换值

图 12-11. A/D 转换器的基本操作



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

在 A/D 转换期间，如果对模拟输入通道选择寄存器(ADS)进行写操作，则转换操作被初始化，并且若 ADCS 被设置为 1，则转换操作重新开始。

复位信号产生将 A/D 转换结果寄存器(ADCR, ADCRH)的内容设置为 0000H 或 00H。

### 12.4.2 输入电压和转换结果

输入到模拟输入引脚(ANI0 ~ ANI5)的模拟输入电压与理论上的 A/D 转换结果(存储在 10 位 A/D 转换结果寄存器(ADCR)中)之间的关系表示如下。

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

$$ADCR = SAR \times 64$$

或

$$\left( \frac{ADCR}{64} - 0.5 \right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left( \frac{ADCR}{64} + 0.5 \right) \times \frac{AV_{REF}}{1024}$$

其中, INT(): 该函数返回括号中值的整数部分。

$V_{AIN}$ : 模拟输入电压

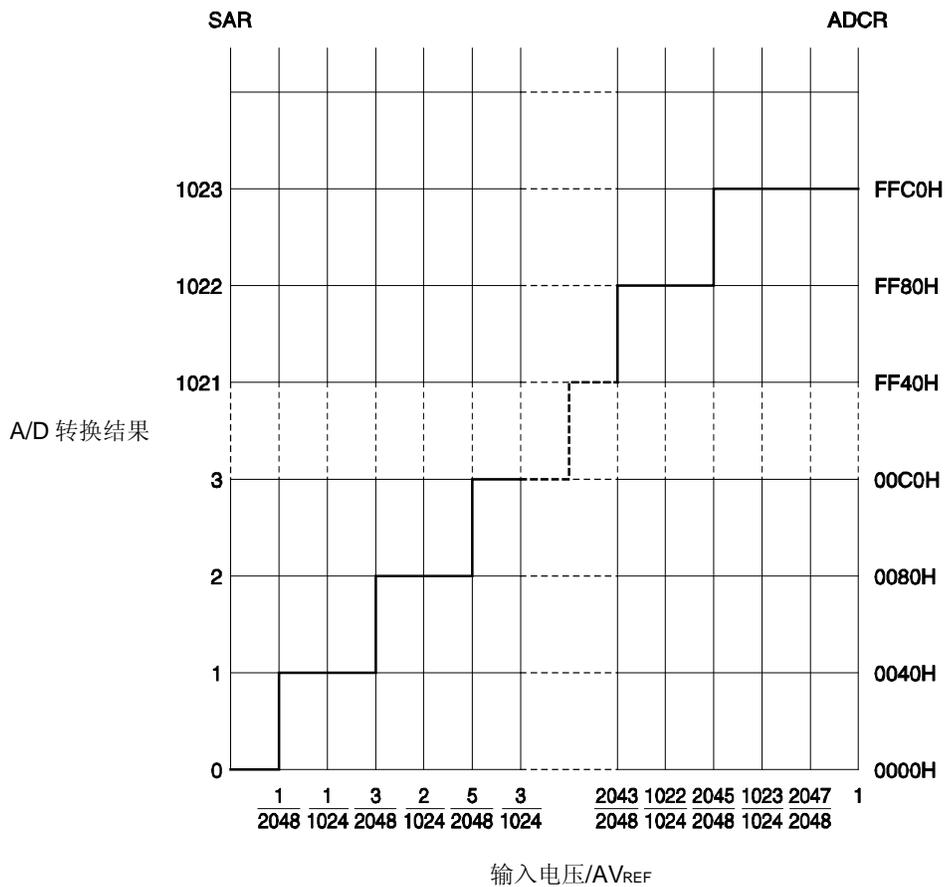
$AV_{REF}$ :  $AV_{REF}$  引脚电压

ADCR: A/D 转换结果寄存器(ADCR)的值

SAR: 逐次逼近寄存器

图 12-12 显示模拟输入电压与 A/D 转换结果之间的关系。

图 12-12. 模拟输入电压和 A/D 转换结果之间的关系



### 12.4.3 12.4.3 A/D 转换器操作模式

A/D 转换器的操作模式为选择模式。由模拟输入通道选择寄存器(ADS)从 ANI0 ~ ANI5 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

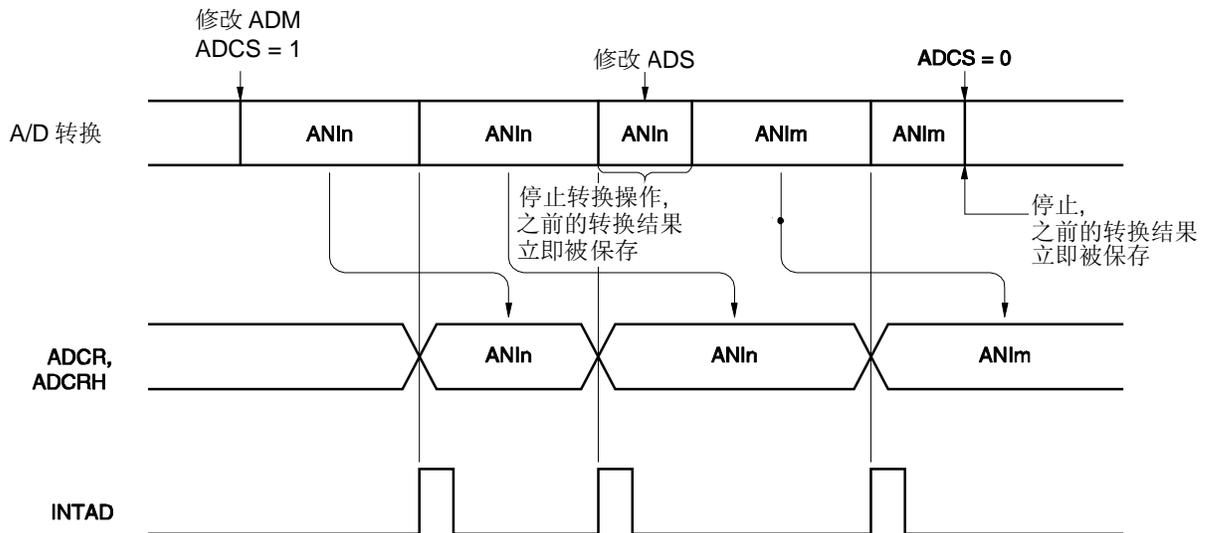
通过将 A/D 转换器模式寄存器(ADM)的第 7 位(ADCS)置 1，可启动电压的 A/D 转换操作，该电压由模拟输入通道选择寄存器(ADS)选择的模拟输入引脚输入。

当 A/D 转换完成时，将 A/D 转换结果存于 A/D 转换结果寄存器(ADCR)中，并产生一个中断请求信号(INTAD)。当一个 A/D 转换结束时，下一个 A/D 转换操作将立即开始。

如果在 A/D 转换期间 ADS 的值被重写，则停止当前正在执行的 A/D 转换操作，重新开始。

如果在 A/D 转换期间 ADCS 被置 0，则 A/D 转换立即停止。此时，之前的转换结果立即被保存。

图 12-13. A/D 转换操作



- 备注
1.  $n = 0 \sim 5$
  2.  $m = 0 \sim 5$

设置方法描述如下。

- <1> 设置 A/D 转换器模式寄存器(ADM)的第 0 位(ADCE)为 1。
- <2> 根据 A/D 端口配置寄存器(ADPC0)的第 2 ~ 0 位(ADPC02 ~ ADPC00)和端口模式寄存器 2(PM2)的第 5 ~ 0 位(PM25 ~ PM20)，设置在模拟输入模式下使用的通道。
- <3> 根据 ADM 的第 6 ~ 1 位(FR3 ~ FR0、LV1 和 LV0)选择转换时间。
- <4> 使用模拟输入通道选择寄存器(ADS)的第 2~0 位(ADS2 ~ ADS0)，选择通道。
- <5> 将 ADM 的第 7 位 (ADCS)置 1，启动 A/D 转换操作。
- <6> 当 A/D 转换结束，产生一个中断请求信号(INTAD)。
- <7> 将 A/D 转换数据传送至 A/D 转换结果寄存器(ADCR，ADCRH)中。

<改变通道>

- <8> 根据 ADS 的第 2 ~ 0 位(ADS2 ~ ADS0)改变通道，启动 A/D 转换操作。
- <9> 当 A/D 转换结束，产生一个中断请求信号(INTAD)。
- <10> 将 A/D 转换数据传送至 A/D 转换结果寄存器(ADCR，ADCRH)中。

<完成 A/D 转换>

- <11> 将 ADCS 清零。
- <12> 将 ADCE 清零。

- 注意事项**
1. 必须确保步骤 <1> ~ <5>的操作时间至少为 1  $\mu$ s。
  2. <1> 可以在<2> 和 <4>之间进行。
  3. <1>可以被省略。但在这种情况中（<5>之后）忽略第一个转换结果。
  4. 步骤<6> ~ <9>所经历的时间与使用 ADM 的第 6 位 ~ 第 1 位(FR3 ~ FR0, LV1, LV0)设置的转换时间不同。步骤<8> ~ <9>所经历的时间为 FR3 ~ FR0, LV1, LV0 设置的转换时间。

## 12.5 A/D 转换器特征表的阅读方法

以下介绍 A/D 转换器中的专用术语。

### (1) 分辨率

这是可识别的最小的模拟输入电压，即每位数字输出的模拟输入电压的百分比，称为 1LSB(最低有效位)。对于满度的 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。

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

精确度与分辨率无关，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。

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

当模拟值转换成数字值时，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的数字代码，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 12-14. 总误差

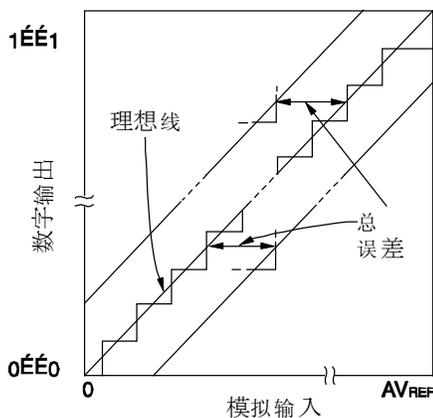
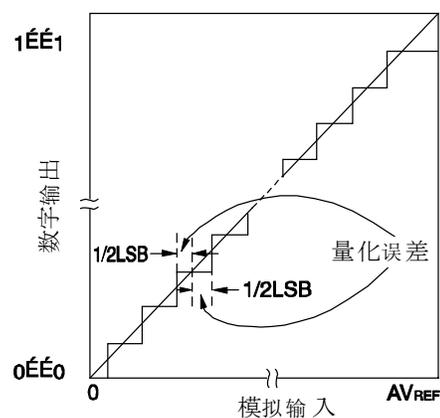


图 12-15. 量化误差



### (4) 零度误差

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

如果实际测量值大于理论值，零度误差表示当数字输出范围在  $0\dots\dots 001 \sim 0\dots\dots 010$  之间时模拟输入电压的实际测量值与理论值( $3/2\text{LSB}$ )之间的误差。

(5) 满度误差

满度误差表示当数字输出范围在  $1\dots\dots110 \sim 1\dots\dots111$  之间时模拟输入电压的实际测量值与理论值(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想线性关系的程度。它表示当零度误差和满度误差均为 0 时实际测量值与理想直线之间误差的最大值。

(7) 微分线性误差

当代码输出的理想宽度为  $1\text{LSB}$  时，微分线性误差表示实际测量值与理想值之间的差距。

图 12-16. 零度误差

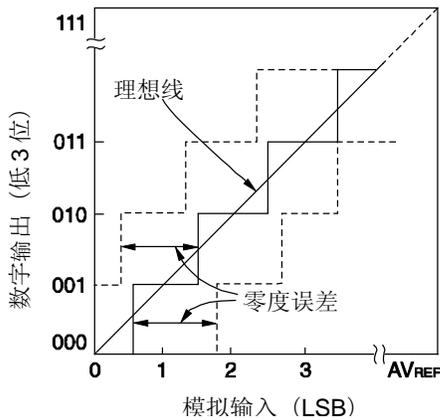


图 12-18. 积分线性误差

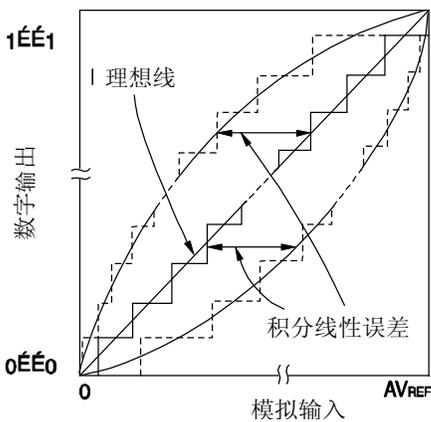


图 12-17. 满度误差

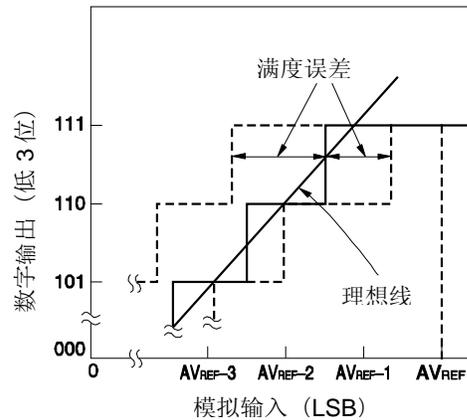
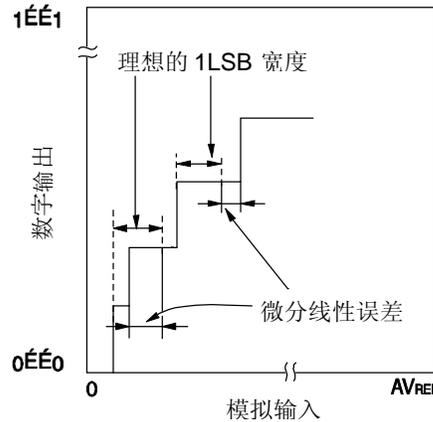


图 12-19. 微分线性误差

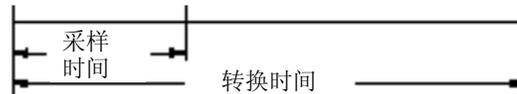


(8) 转换时间

转换时间表示从开始采样到获取数字输出所经历的时间。采样时间包含在特征表中的转换时间中。

(9) 采样时间

采样时间表示模拟电压的模拟开关被打开到模拟电压被采样&保持电路采样所需的时间。



## 12.6 10-位逐次逼近型 A/D 转换器使用注意事项

### (1) STOP 模式中的操作电流

在 STOP 模式中 A/D 转换器停止操作。此时，将 A/D 转换器模式寄存器(ADM)的第 7 位(ADCS)和第 0 位(ADCE)清零，可以降低操作电流。

若要从待机状态重新启动，将中断请求标志寄存器 1L (IF1L)的第 0 位(ADIF)清零(0)，然后开始操作。

### (2) ANI0 ~ ANI5 的输入范围

观察 ANI0 ~ ANI5 输入电压的额定范围。如果输入到模拟输入通道的电压大于等于  $AV_{REF}$ ，或者小于等于  $AV_{SS}$ (即使在绝对最大额定范围之内)，则该通道的转换值不确定。此外，其它通道的转换值也可能受影响。

### (3) 冲突操作

<1> 转换结束后，通过指令对 A/D 转换结果寄存器(ADCR, ADCRH)的写操作和 ADCR 或 ADCRH 的读操作之间的冲突。

ADCR 或 ADCRH 读操作的优先级高。在执行读操作后，才将新的转换结果写入 ADCR 或 ADCRH。

<2> 转换结束后，ADCR 或 ADCRH 的写操作、A/D 转换器模式寄存器(ADM)的写操作以及模拟输入通道选择寄存器(ADS)或 A/D 端口配置寄存器(ADPC)的写操作之间的冲突。

ADM、ADS 或 ADPC 的写操作的优先级高。不执行 ADCR 或 ADCRH 的写操作，也不会产生转换结束中断信号(INTAD)。

### (4) 解决噪音问题的方法

为了保持 10 位分辨率，必须注意输入到  $AV_{REF}$  引脚和 ANI0 ~ ANI5 引脚的噪音。

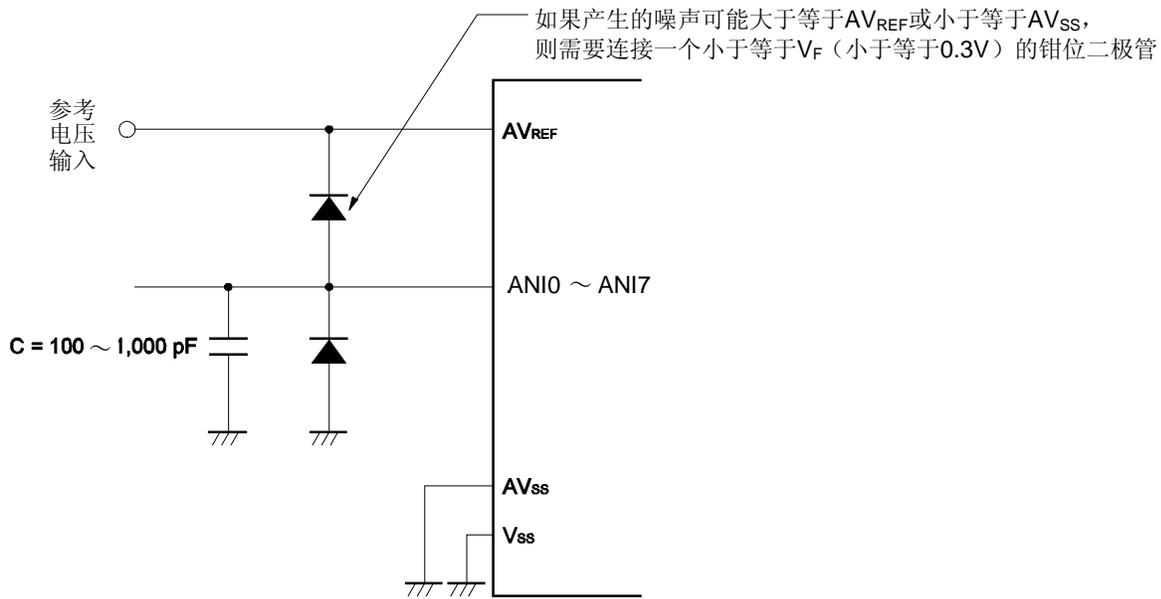
<1> 连接一个低等效电阻和优质频率响应的电容到电源上

<2> 模拟输入源的输出阻抗越大，干扰就越大。为了降低噪音，建议按图 12-20 所示连接外部 C。

<3> 在转换过程中不要切换引脚。

<4> 如果在转换开始后立即设置 HALT 模式，则可以改善精度。

图 12-20. 模拟输入引脚连接



#### (5) ANIO/SEG21/P20 ~ ANI5/SEG16/P25 引脚

- <1> 模拟输入引脚(ANIO ~ ANI5)也可用作输入端口引脚(P20 ~ P25)。  
 当选择 ANIO ~ ANI5 中的任意一个通道执行 A/D 转换时，转换过程中不要访问 P20 ~ P25；否则转换分辨率可能会降低。建议从 ANIO/P20(离 AVREF 最远)开始，选择引脚用作 P20 ~ P25。
- <2> 如果正在进行 A/D 转换的引脚的相邻引脚有数字脉冲输入或输出，或者 segment-输出，则由于噪音耦合，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引用脉冲。

#### (6) ANIO ~ ANI5 引脚的输入阻抗

采样期间 A/D 转换器对采样电容充电，以便进行采样。  
 因此当不进行采样时仅有漏电流经过，而在采样期间则有为电容充电的电流，因此根据是否进行采样，输入阻抗会波动，且无法解决。  
 为了使采样有效，建议模拟输入源的输出阻抗小于等于 10 k $\Omega$ ，并将一个 100 pF 左右的电容连接到 ANIO ~ ANI5 引脚上(参见图 12-20)。

#### (7) AVREF 引脚输入阻抗

在 AVREF 与 AVSS 引脚之间连接几十千欧的串联电阻串。  
 因此，如果参考电压源的输出阻抗很高，当它串联连接到 AVREF 与 AVSS 引脚之间的串联电阻串时，会导致较大的参考电压误差。

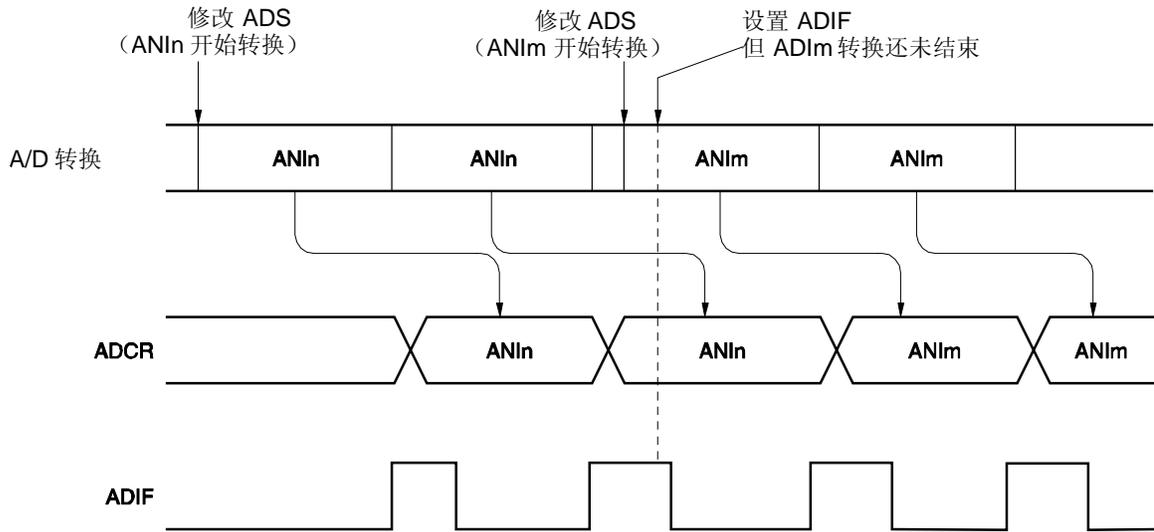
**(8) 中断请求标志(ADIF)**

即使模拟输入通道选择寄存器(ADS)的值被修改, 中断请求标志(ADIF)也不会被清零。

因此, 如果在 A/D 转换期间有一个模拟输入引脚发生变化, 则在 ADS 被修改之前, 模拟输入通道的 A/D 转换结果和改变前的模拟通道的 ADIF 可能被设置。此时需要注意的是, 当 ADS 修改后立即读取 ADIF 时, 即使修改后的模拟输入的 A/D 转换尚未结束, 也会设置 ADIF。

当 A/D 转换停止后又重新开始时, 在启动前先对 ADIF 清零。

图 12-21. A/D 转换结束中断请求产生的时序



- 备注
1.  $n = 0 \sim 5$
  2.  $m = 0 \sim 5$

**(9) A/D 转换刚开始时的转换结果**

在 A/D 转换开始后, 若在 ADCE 置 1 后的  $1 \mu\text{s}$  内对 ADCS 置 1, 或者 ADCE=0 时 ADCS 置 1, 那么第一次的 A/D 转换值可能不在额定范围内。可采取措施, 如悬挂 A/D 转换结束中断请求(INTAD), 并删除第 1 次转换结果。

**(10) 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)、模拟输入通道选择寄存器(ADS)和 A/D 端口配置寄存器(ADPC0)进行写操作时, ADCR 和 ADCRH 的内容可能不确定。在对 ADM、ADS 和 ADPC0 进行写操作前且转换结束后读取转换结果。如果采用与上述不同的时序操作可能会读取到不正确的转换结果。

(11) 内部等效电路

模拟输入模块的等效电路如下所示。

图 12-22. ANIn 引脚的内部等效电路

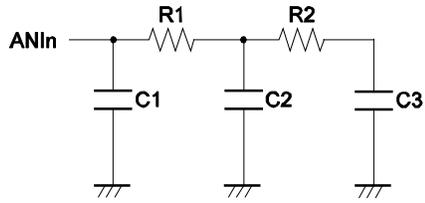


表 12-4. 等效电路的电阻和电容 (参考值)

$AV_{REF}$	R1	R2	C1	C2	C3
2.7 V	TBD	TBD	TBD	TBD	TBD
4.5 V	TBD	TBD	TBD	TBD	TBD

- 备注
1. 表 12-4 所显示的电阻和电容值仅为参考值。
  2.  $n = 0 \sim 5$

### 13.1 串行接口 UART0 的功能

串行接口 UART0 有以下两种模式。

#### (1) 操作停止模式

这个模式在不进行串口通信时使用，以降低功耗。

需要了解详细信息，请参阅 **13.4.1 操作停止模式**。

#### (2) 异步串行接口(UART)模式

该模式的功能概括如下。

需要了解详细信息，请参阅 **13.4.2 异步串行接口(UART)模式** 和 **13.4.3 专用波特率发生器**。

- 最大传输率： 625 kbps
- 双引脚配置 TXD0: 发送数据输出引脚  
RXD0: 接收数据输入引脚
- 可选择通信数据的宽度为 7 或 8 位。
- 专用内置 5 位波特率发生器，允许设置任意大小的波特率。
- 可分别执行发送和接收操作(全双工操作)。
- 恒为 LSB-first 通信

- 注意事项**
1. 如果串行接口 UART0 使用的时钟未停止(例如处于 HALT 模式下)，则可继续正常操作。如果时钟已停止(例如处于 STOP 模式下)，则每个寄存器都停止操作，并且在时钟停止之前及时保存数据。TXD0 引脚在时钟停止之前立即保存数据并输出。但在时钟恢复使用后的操作不能得到保证。因此必须复位电路，使 POWER0 = 0、RXE0 = 0 以及 TXE0 = 0。
  2. 先设置 POWER0 = 1，然后设置 TXE0 = 1(发送)或 RXE0 = 1(接收)开始通信。
  3. 通过由 BRGC0 设置的基本时钟(f<sub>XCLK0</sub>)，可使 TXE0 与 RXE0 同步。为了能够再次发送或接收，在 TXE0 或 RXE0 被清零后，至少要经历两个基本时钟再将 TXE0 或 RXE0 置 1。如果在不到两个基本时钟时设置 TXE0 或 RXE0，则发送或接收电路可能不被初始化。
  4. 设置 TXE0=1 之后经过至少一个基本时钟(f<sub>XCLK0</sub>)才能发送数据到 TXS0。

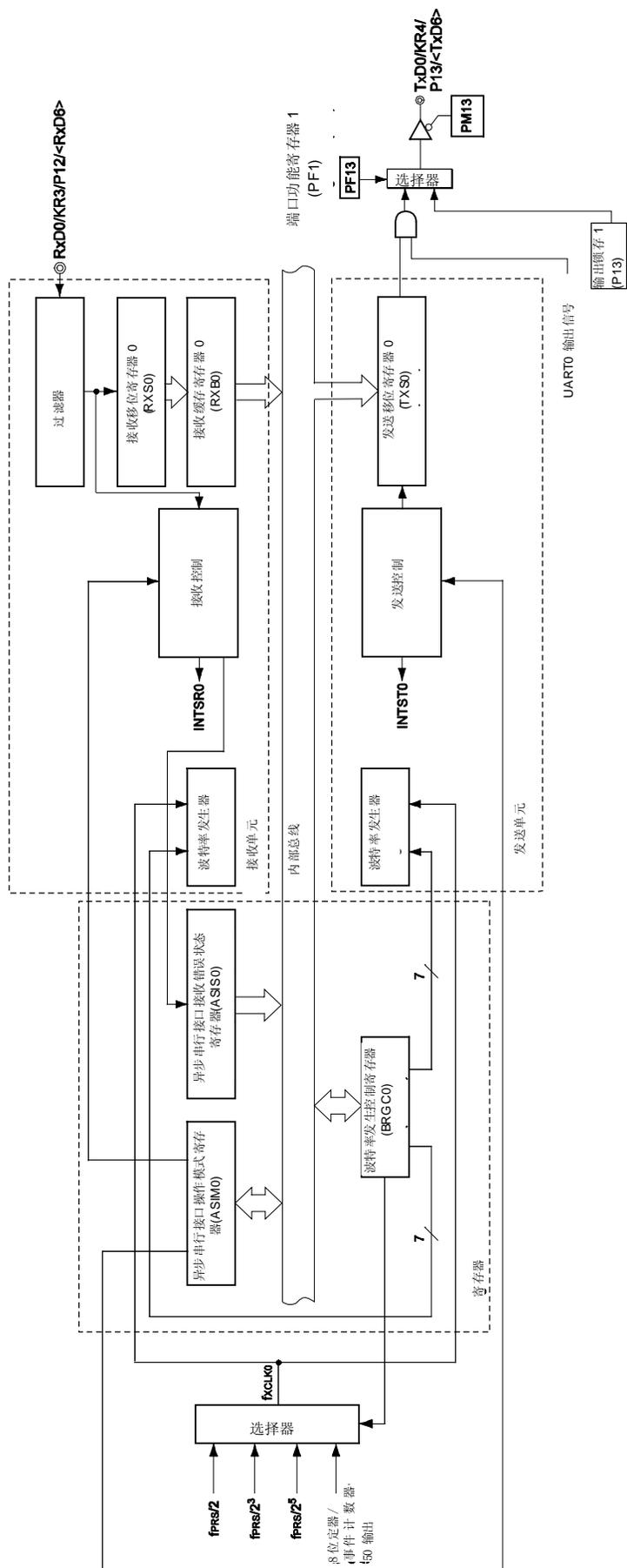
## 13.2 串行接口 UART0 的配置

串行接口 UART0 包括以下硬件。

表 13-1. 串行接口 UART0 的配置

名称	配置
寄存器	接收缓冲寄存器 0 (RXB0) 接收移位寄存器 0 (RXS0) 发送移位寄存器 0 (TXS0)
控制寄存器	异步串行接口操作模式寄存器 0 (ASIM0) 异步串行接口接收错误状态寄存器 0 (ASIS0) 波特率发生器控制寄存器 0 (BRGC0) 端口功能寄存器 1 (PF1) 端口模式寄存器 1 (PM1) 端口寄存器 1 (P1)

图 13-1. 串行接口 UART0 框图



**(1) 接收缓冲寄存器 0 (RXB0)**

该 8 位寄存器用于存储由接收移位寄存器 0(RXS0)转换的并行数据。

每接收到 1 字节的数据，新的接收数据就从接收移位寄存器 0(RXS0)被传送到该寄存器中。

如果数据宽度被设定为 7 位，则将接收到的数据传送到 RXB0 的 0 ~ 6 位，而 MSB 位恒为 0。

如果出现溢出错误(OVE0)，则不把接收数据传送到 RXB0 中。

可由 8 位存储器操作指令读取 RXB0 的内容。不能将数据写入该寄存器中。

复位信号的产生和 POWER0 = 0 将该寄存器设置为 FFH。

**(2) 接收移位寄存器 0 (RXS0)**

该寄存器将输入到 RxD0 引脚的串行数据转换成并行数据。

程序不能直接使用 RXS0。

**(3) 发送移位寄存器 0 (TXS0)**

该寄存器用于设置发送数据。当数据写入 TXS0 时启动发送操作，串行数据从 TxD0 引脚输出。

可由 8 位存储器操作指令对 TXS0 进行写操作。该寄存器的内容不能被读取。

复位信号的产生、POWER0 = 0 和 TXE0 = 0 将该寄存器设置为 FFH。

**注意事项** 1. 设置 TXE0=1 之后经过至少一个基本时钟( $f_{xclk0}$ )才能发送数据到 TXS0。

2. 在发送完成中断信号(INTST0)产生之前，不要把下一个发送数据写入 TXS0。

### 13.3 控制串行接口 UART0 的寄存器

串行接口 UART0 由以下 6 个寄存器控制。

- 异步串行接口操作模式寄存器 0 (ASIM0)
- 异步串行接口接收错误状态寄存器 0 (ASIS0)
- 波特率发生器控制寄存器 0 (BRGC0)
- 端口功能寄存器 1 (PF1)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

#### (1) 异步串行接口操作模式寄存器 0 (ASIM0)

该 8 位寄存器用于控制串行接口 UART0 的串行通信操作。

该寄存器可由 1 位或 8 位存储器操作指令设置。

复位信号的产生将该寄存器设置为 01H。

图 13-2. 异步串行接口操作模式寄存器 0 (ASIM0)的格式 (1/2)

地址: FF70H 复位后: 01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1
POWER0	内部操作时钟的允许/禁止操作							
0 <sup>#1</sup>	禁止内部操作时钟的操作 (时钟恒为低电平) 并异步复位内部电路 <sup>#2</sup>							
1	允许内部操作时钟的操作							
TXE0	允许/禁止发送							
0	禁止发送 (同步复位发送电路)							
1	允许发送							
RXE0	允许/禁止接收							
0	禁止接收 (同步复位接收电路)							
1	允许接收							

注

1. 当 POWER0 = 0 时, 从 RxD0 引脚输入的信号恒为高电平。
2. 对异步串行接口接收错误状态寄存器 0(ASIS0)、发送移位寄存器 0(TXS0)和接收缓冲寄存器 0(RXB0)进行复位。

图 13-2. 异步串行接口操作模式寄存器 0 (ASIM0)的格式 (2/2)

PS01	PS00	发送操作	接收操作
0	0	不输出校验位	无校验接收
0	1	输出零校验	零校验接收 <sup>注</sup>
1	0	输出奇校验	按奇校验判断
1	1	输出偶校验	按偶校验判断

CL0	指定发送/接收数据的字符宽度
0	数据的字符宽度 = 7 位
1	数据的字符宽度 = 8 位

SL0	指定发送数据停止位的个数
0	停止位个数 = 1
1	停止位个数 = 2

**注** 如果选择“按零校验接收”，则不判断校验。因此不要设置异步串行接口接收错误状态寄存器 0(ASIS0)的第 2 位(PE0)，这样就不会产生错误中断。

- 注意事项**
1. 启动发送时，对 **POWER0** 置 1，然后对 **TXE0** 置 1。停止发送时，先将 **TXE0** 清零，然后将 **POWER0** 清零。
  2. 启动接收时，对 **POWER0** 置 1，然后对 **RXE0** 置 1。停止接收时，先将 **RXE0** 清零，然后将 **POWER0** 清零。
  3. 当 **RxD0** 引脚输入为高电平时，先对 **POWER0** 置 1，再对 **RXE0** 置 1。当输入为低电平，**POWER0** 置 1，**RXE0** 置 1 时，启动接收操作。
  4. 通过 **BRGC0** 设置的基本时钟( $f_{xCLK0}$ )同步 **TXE0** 与 **RXE0**。为了能够再次发送或接收，在 **TXE0** 或 **RXE0** 被清零后，至少要经历两个基本时钟再将 **TXE0** 或 **RXE0** 置 1。如果在不到两个基本时钟的时间内设置 **TXE0** 或 **RXE0**，则发送或接收电路可能不被初始化。
  5. 设置 **TXE0=1** 之后经过至少一个基本时钟( $f_{xCLK0}$ )才能设置发送数据到 **TXS0**。
  6. 在对 **PS01**、**PS00** 和 **CL0** 位重写之前，先将 **TXE0** 和 **RXE0** 位清零。
  7. 在重写 **SL0** 位时，必须确保 **TXE0 = 0**。总是以“停止位的个数=1”的格式执行接收操作，因此不会受到 **SL0** 设置值的影响。
  8. 第 0 位必须设置为 1。

**(2) 异步串行接口接收错误状态寄存器 0 (ASIS0)**

该寄存器用于指示串行接口 UART0 在接收操作完成时出现的错误状态，包括 3 个错误标志位(PE0, FE0, OVE0)。可由 8 位存储器操作指令对该寄存器进行只读操作。

复位信号产生，将 ASIS0 寄存器设置为 00H。如果将 ASIS0 的第 5 位(RXE0)和第 7 位(POWER0)置 0，则读取该寄存器的值为 00H。如果产生一个接收错误，则先读取 ASIS0 再读取接收缓冲寄存器 0(RXB0)以便将出错标志位清零。

图 13-3. 异步串行接口接收错误状态寄存器 0 (ASIS0)的格式

地址: FF73H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	指示校验错误的状态标志位
0	如果 POWER0 = 0 且 RXE0 = 0、或者如果读取 ASIS0
1	如果发送数据的校验位与接收完成时的校验位不相等

FE0	指示帧错误的状态标志位
0	如果 POWER0 = 0 且 RXE0 = 0、或者如果读取 ASIS0
1	如果接收完成时未检测到停止位

OVE0	指示溢出错误的状态标志位
0	如果 POWER0 = 0 且 RXE0 = 0、或者如果读取 ASIS0
1	如果接收数据被设置在 RXB0 寄存器中且在读取该数据前下一个接收操作已完成

- 注意事项**
1. 根据异步串行接口操作模式寄存器 0(ASIM0)的 PS01 和 PS00 位的内容的不同，对 PE0 位的操作也有所不同。
  2. 无论停止位的个数是多少，只能将接收数据的第 1 位作为停止位进行检测。
  3. 如果出现溢出错误，则下一个接收数据不写入接收缓冲寄存器 0(RXB0)中，而是被忽略。
  4. 如果从 ASIS0 读取数据，会产生一个等待周期。当 CPU 采用副系统时钟操作而停止外围硬件时钟时，不要从 ASIS0 读取数据。需要了解详细信息，可参见 第二十九章 等待注意事项。

**(3) 波特率发生器控制寄存器 0 (BRGC0)**

该寄存器用于选择串行接口 UART0 的基本时钟和 5 位计数器的分频值。

可由 8 位存储器操作指令设置 BRGC0。

复位信号的产生将该寄存器设置为 1FH。

图 13-4. 波特率发生器控制寄存器 0 (BRGC0)的格式

地址: FF71H 复位后: 1FH R/W

符号	7	6	5	4	3	2	1	0
BRGC0	TPS01	TPS00	0	MDL04	MDL03	MDL02	MDL01	MDL00

TPS01	TPS00	基时钟 ( $f_{XCLK0}$ ) 选择 <sup>注 1</sup>				
		$f_{PRS} = 2 \text{ MHz}$	$f_{PRS} = 5 \text{ MHz}$	$f_{PRS} = 8 \text{ MHz}$	$f_{PRS} = 10 \text{ MHz}$	
0	0	TM50 output <sup>注 2</sup>				
0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	4 MHz	5 MHz
1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1 MHz	1.25 MHz
1	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	250 kHz	312.5 kHz

MDL04	MDL03	MDL02	MDL01	MDL00	k	5-位计数器输出时钟的选择
0	0	×	×	×	×	禁止设置
0	1	0	0	0	8	$f_{XCLK0}/8$
0	1	0	0	1	9	$f_{XCLK0}/9$
0	1	0	1	0	10	$f_{XCLK0}/10$
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	0	1	0	26	$f_{XCLK0}/26$
1	1	0	1	1	27	$f_{XCLK0}/27$
1	1	1	0	0	28	$f_{XCLK0}/28$
1	1	1	0	1	29	$f_{XCLK0}/29$
1	1	1	1	0	30	$f_{XCLK0}/30$
1	1	1	1	1	31	$f_{XCLK0}/31$

- 注 1. 如果外围硬件时钟( $f_{PRS}$ )运行在高速系统时钟( $f_{XH}$ ) ( $XSEL = 1$ ),  $f_{PRS}$ 运行频率依据供电电压的不同而变化。
- $V_{DD} = 2.7 \sim 5.5 \text{ V}$ :  $f_{PRS} \leq 10 \text{ MHz}$
  - $V_{DD} = 1.8 \sim 2.7 \text{ V}$ :  $f_{PRS} \leq 5 \text{ MHz}$
2. 选择 TM50 输出作为基本时钟时, 先启动 8 位定时器/事件计数器 50, 然后允许定时器 F/F 进行反转操作( $TMC501 = 1$ )。

- 注意事项 1. 在重写 MDL04 ~ MDL00 位时, 必须确保 ASIM0 寄存器的第 6 位(TXE0)和第 5 位(RXE0) = 0。
2. 波特率为 5 位计数器输出时钟的 1/2。

- 备注**
1. f<sub>CLK0</sub>: 根据 TPS01 和 TPS00 位选择的基本时钟的频率
  2. f<sub>PRS</sub>: 外围硬件时钟频率
  3. k: 由 MDL04 ~ MDL00 位设置的值 (k = 8, 9, 10, ..., 31)
  4. x: 不必考虑

**(4) 端口功能寄存器 1 (PF1)**

此寄存器设置 P13/TxD0/KR4/<TxD6>引脚的引脚功能。

可由 1-位或 8-位存储器操作指令设置。

复位信号将此寄存器设置为 00H。

**备注** 带有箭头 (< >)的功能可通过输入转换控制寄存器 (ISC)分配。

图 13-5.端口功能寄存器 1 (PF1)的格式

地址: FF20H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PF1	0	0	0	0	PF13	0	0	0

PF13	端口 (P13), UART0, 和 UART6 输出规定
0	用作 P13
1	用作 TxD0 或者 TxD6

**(5) 端口模式寄存器 1 (PM1)**

该寄存器按位设置端口 1 的输入/输出模式。

当 P13/TxD0/KR4/<TxD6> 引脚用于串行接口数据输出时，将 PM10 清零且对 P10 的输出锁存器置 1。

当 P12/RxD0/KR3/<RxD6> 引脚用于串行接口数据输入时，将 PM11 置 1。此时 P11 的输出锁存器的值可以等于 0 或 1。

可由 1 位或 8 位存储器操作指令设置 PM1。

复位信号的产生将该寄存器设置为 FFH。

**备注** 带有箭头 (< >) 的功能可通过输入转换控制寄存器 (ISC) 分配。

**图 13-6. 端口模式寄存器 1(PM1)的格式**

地址: FF21H 复位后: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	1	1	PM13	PM12	1	1

PM1n	P1n 引脚 I/O 模式选择 (n = 2, 3)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

## 13.4 串行接口 UART0 的操作

串行接口 UART0 有以下两种模式。

- 操作停止模式
- 异步串行接口(UART)模式

### 13.4.1 操作停止模式

在此模式下，不能进行串行通信，这样可降低功耗。此外，引脚可作为通用端口引脚使用。通过将 ASIM0 的第 7 位、第 6 位和第 5 位(POWER0、TXE0 和 RXE0)清零，可设置操作停止模式。

#### (1) 使用的寄存器

由异步串行接口操作模式寄存器 0(ASIM0)设置操作停止模式。

可由 1 位或 8 位存储器操作指令设置 ASIM0。

复位信号的产生将该寄存器设置为 01H。

地址: FF70H 复位后: 01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1
POWER0	内部操作时钟的允许/禁止操作							
0 <sup>注1</sup>	禁止内部操作时钟的操作(时钟恒为低电平)且对内部电路进行异步复位 <sup>注2</sup>							
TXE0	允许/禁止发送							
0	禁止发送(同步复位发送电路)							
RXE0	允许/禁止接收							
0	禁止接收(同步复位接收电路)							

- 注
1. 当 POWER0 = 0 时，从 RxD0 引脚输入的信号恒为高电平。
  2. 对异步串行接口接收错误状态寄存器 0(ASIS0)、发送移位寄存器 0(TXS0)和接收缓冲寄存器 0(RXB0)进行复位。

**注意事项** TXE0 和 RXE0 清零后再对 POWER0 清零，可设置操作停止模式。

如果要启动发送或接收，设置 POWER0=1，然后设置 TXE0 或 RXE0=1。

**备注** 如果要将在 RxD0/KR3/<RxD6>/P12 和 TxD0/KR4/<TxD6>/P13 作为通用端口引脚使用，可参见 第四章 端口功能。

### 13.4.2 异步串行接口 (UART)模式

此模式下，在起始位后的 1 字节数据被发送/接收，可执行全双工操作。  
可使用专用 UART 波特率发生器，这样通信选择的波特率范围较大。

#### (1) 使用的寄存器

- 异步串行接口操作模式寄存器 0 (ASIM0)
- 异步串行接口接收错误状态寄存器 0 (ASIS0)
- 波特率发生器控制寄存器 0 (BRGC0)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

在 UART 模式中设置操作的基本过程如下。

- <1> 设置 BRGC0 寄存器 (参见 图 13-4)。
- <2> 设置 ASIM0 寄存器的第 1 位~第 4 位 (SL0、CL0、PS00 和 PS01)(参见 图 13-2)。
- <3> 将 ASIM0 寄存器的第 7 位 (POWER0)置 1。
- <4> 将 ASIM0 寄存器的第 6 位 (TXE0)置 1。 → 允许发送。  
将 ASIM0 寄存器的第 5 位 (RXE0)置 1。 → 允许接收。
- <5> 对 TXS0 寄存器赋值。 → 开始发送数据。

**注意事项** 在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。

寄存器设置与引脚之间的关系如下所示。

表 13-2. 寄存器设置与引脚之间的关系

POWER0	TXE0	RXE0	PM13	P13	PM12	P12	UART0 Operation	引脚功能	
								TxD0/KR4/P13/<TxD6>	RxD0/KR3/P12/<RxD6>
0	0	0	x <sup>注</sup>	x <sup>注</sup>	x <sup>注</sup>	x <sup>注</sup>	停止	KR4/P13/<TxD6>	KR3/P12/<RxD6>
1	0	1	x <sup>注</sup>	x <sup>注</sup>	1	x	接收	KR4/P13	RxD0
	1	0	0	x	x <sup>注</sup>	x <sup>注</sup>	发送	TxD0	KR3/P12
	1	1	0	x	1	x	发送/ 接收	TxD0	RxD0

**注** 能够作为端口，按键中断，或者串行接口 UART6 (只有当 UART0 停止时)使用。

- 备注**
- 1.x: 不必考虑  
POWER0: 异步串行接口操作模式寄存器 0 (ASIM0)的第 7 位  
TXE0: ASIM0 的第 6 位  
RXE0: ASIM0 的第 5 位  
PM1x: 端口模式寄存器  
P1x: 端口输出锁存器
2. 带有箭头 (< >)的功能可通过输入转换控制寄存器 (ISC)分配。

(2) 通信操作

(a) 正常发送/接收数据的格式和波形示例

图 13-6 和 13-7 为正常发送/接收数据的格式和波形示例。

图 13-7. UART 正常发送/接收数据的格式



一个数据帧由以下各位组成。

- 起始位 ... 1 位
- 字符位 ... 7 或 8 位 (LSB-first)
- 校验位 ... 偶校验、奇校验、零校验或无校验
- 停止位 ... 1 或 2 位

由异步串行接口操作模式寄存器 0(ASIM0)规定一个数据帧中字符位的宽度、校验方式和停止位的宽度。

图 13-8. UART 正常发送/接收数据波形示例

1. 数据宽度: 8 位, 校验位: 偶校验, 停止位: 1 位, 通信数据: 55H



2. 数据宽度: 7 位, 校验位: 奇校验, 停止位: 2 位, 通信数据: 36H



3. 数据宽度: 8 位, 校验位: 无, 停止位: 1 位, 通信数据: 87H



**(b) 校验方式与操作**

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

**(i) 偶校验**

## • 发送

控制发送数据，包括校验位，使得数据中“1”的个数为偶数。

校验位的取值如下。

如果发送数据有奇数个“1”：1

如果发送数据有偶数个“1”：0

## • 接收

计算接收数据(包括校验位)中“1”的个数。如果结果为奇数，则产生校验错误。

**(ii) 奇校验**

## • 发送

与偶校验方式不同，奇校验用于控制发送数据，包括校验位，使得数据中“1”的个数为奇数。

校验位取值如下。

如果发送数据有奇数个“1”：0

如果发送数据有偶数个“1”：1

## • 接收

计算接收数据(包括校验位)中“1”的个数。如果结果为偶数，则产生校验错误。

**(iii) 零校验**

无论发送数据为何值，发送时校验位被清零。

接收数据时不检测校验位。因此无论校验位为何值，都不会产生校验错误。

**(iv) 无校验**

发送数据中没有校验位。

进行接收操作时认为接收数据中没有校验位。因此也不会产生校验错误。

**(c) 发送**

当异步串行接口操作模式寄存器 0(ASIM0)的第 7 位(POWER0)=1 且 ASIM0 的第 6 位(TXE0)=1 时, 允许发送。将待发送数据写入发送移位寄存器 0(TXS0), 可启动发送操作。起始位、校验位和停止位会自动添加到数据中。

启动发送操作时, 先从 TxD0 引脚输出起始位, 然后从 LSB 开始依次发送数据其它位。当发送结束时, 由 ASIM0 设置的校验位和停止位被添加到数据中, 并产生发送完成中断请求(INTST0)。

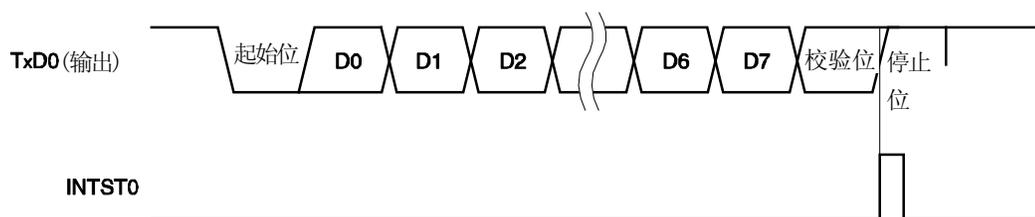
当下一个待发送数据被写入 TXS0 时, 发送操作才结束。

图 13-9 显示了发送完成中断请求(INTST0)的时序。最后一个停止位一输出就产生该中断。

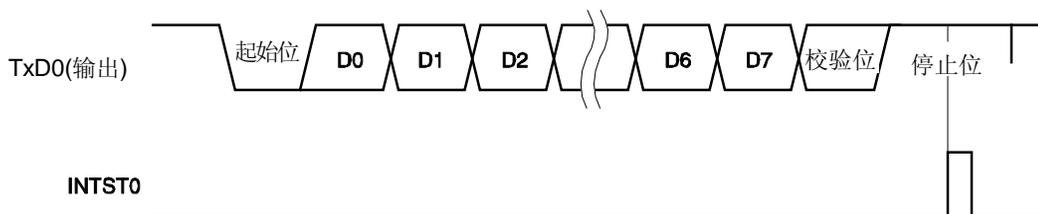
**注意事项** 在将待发送数据写入 TXS0 后, 在发送完成中断信号(INTST0)产生之前, 不要将下一个发送数据写入 TXS0。

图 13-9. 发送完成中断请求的时序

1. 停止位宽度: 1



2. 停止位宽度: 2



**(d) 接收**

当异步串行接口操作模式寄存器 0(ASIM0)的第 7 位(POWER0)=1，然后将 ASIM0 的第 5 位(RXE0)置 1 时，允许接收并对 RxD0 引脚输入进行采样。

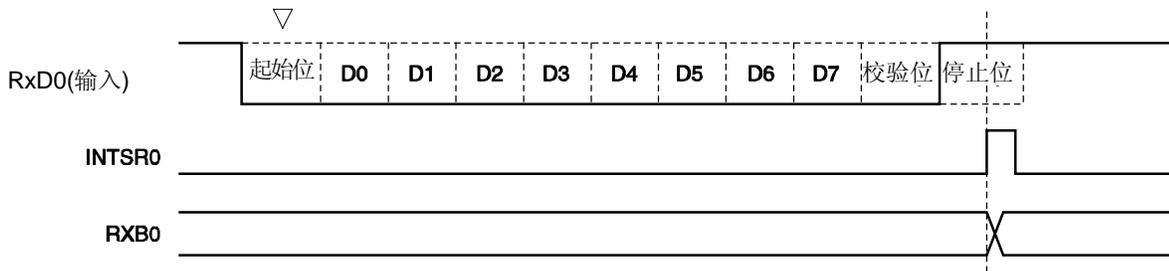
当检测到 RxD0 引脚输入的下降沿时波特率发生器的 5 位计数器开始计数。当计数值等于波特率发生器控制寄存器 0 (BRGC0) 的设置值时，再次采样 RxD0 的输入信号 (如图 14-10 中  $\nabla$  )。如果此时 RxD0 为低电平，则认为输入信号为起始位。

当检测到起始位时，开始接收，以设置的波特率将串行数据依次存入接收移位寄存器 0(RXS0)。当接收到停止位时，会产生接收完成中断信号(INTSR0)，并且将 RXS0 的数据写入接收缓冲寄存器 0(RXB0)。但如果出现溢出错误，则接收到的数据不写入 RXB0。

即使在接收过程中出现校验错(PE0)，接收仍然继续，直至接收到停止位，而在接收完成后会产生一个接收错误中断(INTSR0)。

当接收操作完成且存在接收错误，则产生 INTSR0

图 13-10. 接收完成中断请求时序



- 注意事项**
1. 如果出现接收错误，必须先读取异步串行接口接收错误状态寄存器 0 (ASIS0)再读取接收缓冲寄存器 0 (RXB0)的内容，才能对出错标志清零。否则，当接收到下一个数据时会产生溢出错误，而且接收错误状态保持不变。
  2. 接收始终按“停止位的个数 = 1”的情况执行，第 2 个停止位被忽略。

**(e) 接收错误**

在接收期间有三种出错情况：校验错误、帧错误或溢出错误。如果异步串行接口接收错误状态寄存器 0(ASIS0) 的错误标志位被设置，作为数据接收的结果，则将会产生一个接收错误中断请求信号 (INTSR0)。

在执行接收错误中断服务(INTSR0)时读取 ASIS0 的内容，可确定接收期间出现的是何种错误(参见图 13-3)。读取 ASIS0 后，将其内容清零。

表 13-3. 接收错误产生的原因

接收错误	原因
校验错误	发送数据的校验位与接收数据的校验位不相等
帧错误	未检测到停止位
溢出错误	在从接收缓冲寄存器 0(RXB0)中读取数据前已完成下一个数据的接收操作

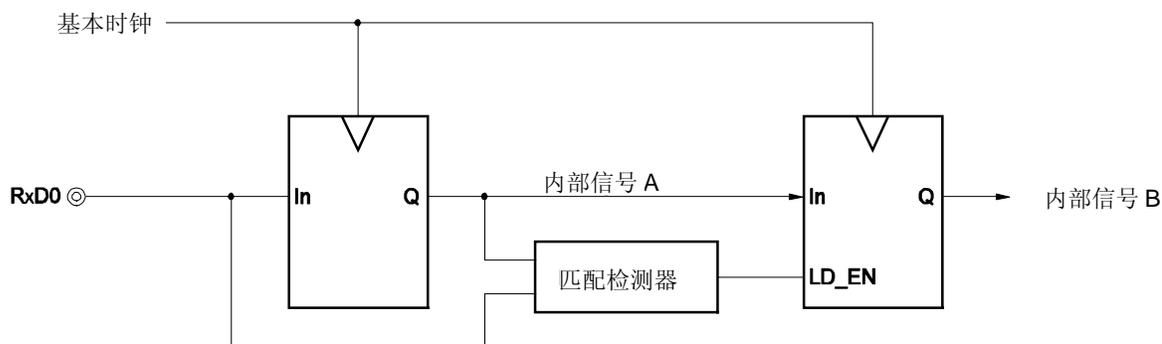
**(f) 接收数据的噪声过滤器**

通过预分频器，使用基本时钟输出，采样 RxD0 信号。

如果两次采样值相同，则相等检测器的输出会发生变化，并把采样的数据作为输入数据。

噪声过滤器的电路结构如图 13-11 所示，接收操作的内部过程与外部信号状态相比被延迟两个时钟执行。

图 13-11. 噪声过滤器电路



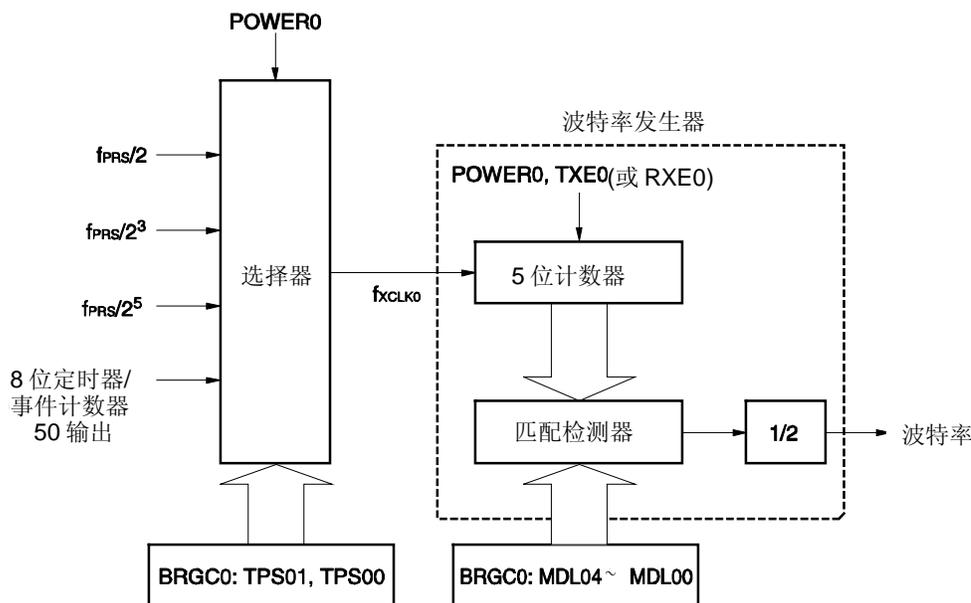
### 13.4.3 专用波特率发生器

专用波特率发生器由一个源时钟选择器和一个 5 位可编程计数器组成，用于产生 UART0 发送/接收的串行时钟。分别使用不同的 5 位计数器用于发送和接收。

#### (1) 波特率发生器的配置

- 基本时钟**  
 该时钟由波特率发生器控制寄存器 0(BRGC0)的第 7 位和第 6 位(TPS01 和 TPS00)选择。当异步串行接口操作模式寄存器 0(ASIM0)的第 7 位(POWER0)=1 时，将该时钟提供给每个模块。该时钟称为基本时钟，它的频率用  $f_{CLK0}$  表示。当 POWER0 = 0 时，基本时钟恒为低电平。
- 发送计数器**  
 当异步串行接口操作模式寄存器 0(ASIM0)的第 7 位(POWER0)或第 6 位(TXE0) = 0 时，该计数器停止计数，并被清零。  
 当 POWER0 = 1 且 TXE0 = 1 时计数器开始计数。  
 当第 1 个发送数据被写入发送移位寄存器 0(TXS0)时，计数器被清零。
- 接收计数器**  
 当异步串行接口操作模式寄存器 0(ASIM0)的第 7 位(POWER0)或第 5 位(RXE0) = 0 时，该计数器停止计数，并被清零。  
 当检测到起始位时，计数器开始计数。  
 在收到一帧数据后计数器停止计数，直到下一个起始位被检测到时再重新开始计数。

图 13-12. 波特率发生器的配置



- 备注**
- POWER0: 异步串行接口操作模式寄存器 0 (ASIM0) 的第 7 位
  - TXE0: ASIM0 的第 6 位
  - RXE0: ASIM0 的第 5 位
  - BRGC0: 波特率发生器控制寄存器 0

**(2) (2) 串行时钟的发生器**

使用波特率发生器控制寄存器 0 (BRGC0)可产生串行时钟。

根据 BRGC0 的第 7 位和第 6 位(TPS01 和 TPS00)，可选择输入给 5 位计数器的时钟。

根据 BRGC0 的第 4 位 ~ 第 0 位(MDL04 ~ MDL00)选择 5 位计数器的分频值( $f_{xCLK0}/8 \sim f_{xCLK0}/31$ )。

**表 13-4. TPS01 和 TPS00 的设置值**

TPS01	TPS00	基时钟 ( $f_{xCLK0}$ ) 选择 <sup>#1</sup>				
		$f_{PRS} = 2 \text{ MHz}$	$f_{PRS} = 5 \text{ MHz}$	$f_{PRS} = 8 \text{ MHz}$	$f_{PRS} = 10 \text{ MHz}$	
0	0	TM50 输出 <sup>#2</sup>				
0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	4 MHz	5 MHz
1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1 MHz	1.25 MHz
1	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	250 kHz	312.5 kHz

- 注
1. 如果外围硬件时钟( $f_{PRS}$ )运行在高速系统时钟( $f_{XH}$ ) ( $XSEL = 1$ ),  $f_{PRS}$  运行频率依据供电电压的不同而变化。
    - $V_{DD} = 2.7 \sim 5.5 \text{ V}$ :  $f_{PRS} \leq 10 \text{ MHz}$
    - $V_{DD} = 1.8 \sim 2.7 \text{ V}$ :  $f_{PRS} \leq 5 \text{ MHz}$
  2. 选择 TM50 输出作为基本时钟时, 先启动 8 位定时器/事件计数器 50, 然后允许定时器 F/F 进行反转操作( $TMC501 = 1$ )。

**(2) 波特率误差**

通过下列公式计算波特率误差。

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

- 注意事项 1. 在发送过程中必须保持波特率误差在接收方允许的误差范围内。  
2. 接收期间的波特率误差必须满足“(4)接收期间允许的波特率范围”中所描述的范围。

举例: 基本时钟的频率 = 2.5 MHz = 2,500,000 Hz  
BRGC0 的 MDL04 ~ MDL00 位 = 10000B(k = 16)  
目标波特率 = 76,800 bps

$$\begin{aligned} \text{波特率} &= 2.5 \text{ M}/(2 \times 16) \\ &= 2,500,000/(2 \times 16) = 78,125 \text{ [bps]} \end{aligned}$$

$$\begin{aligned} \text{误差} &= (78,125/76,800 - 1) \times 100 \\ &= 1.725 \text{ [%]} \end{aligned}$$

### (3) 波特率设置举例

表 13-5. 波特率发生器的数据设置

波特率 [bps]	f <sub>PRS</sub> = 2.0 MHz				f <sub>PRS</sub> = 5.0 MHz				f <sub>PRS</sub> = 10.0 MHz			
	TPS01, TPS00	k	计算值	ERR [%]	TPS01, TPS00	k	计算值	ERR [%]	TPS01, TPS00	k	计算值	ERR [%]
1200	3H	26	1202	0.16	-	-	-	-	-	-	-	-
2400	3H	13	2404	0.16	-	-	-	-	-	-	-	-
4800	2H	26	4808	0.16	3H	16	4883	1.73	-	-	-	-
9600	2H	13	9615	0.16	3H	8	9766	1.73	3H	16	9766	1.73
10400	2H	12	10417	0.16	2H	30	10417	0.16	3H	15	10417	0.16
19200	1H	26	19231	0.16	2H	16	19531	1.73	3H	8	19531	1.73
24000	1H	21	23810	-0.79	2H	13	24038	0.16	2H	26	24038	0.16
31250	1H	16	31250	0	2H	10	31250	0	2H	20	31250	0
33660	1H	15	33333	-0.79	2H	9	34722	3.34	2H	19	32895	-2.1
38400	1H	13	38462	0.16	2H	8	39063	1.73	2H	16	39063	1.73
56000	1H	9	55556	-0.79	1H	22	56818	1.46	2H	11	56818	1.46
62500	1H	8	62500	0	1H	20	62500	0	2H	10	62500	0
76800	-	-	-	-	1H	16	78125	1.73	2H	8	78125	1.73
115200	-	-	-	-	1H	11	113636	-1.36	1H	22	113636	-1.36
153600	-	-	-	-	1H	8	156250	1.73	1H	16	156250	1.73
312500	-	-	-	-	1H	4	312500	1.73	1H	8	312500	0
625000	-	-	-	-	-	-	-	-	1H	4	625000	0

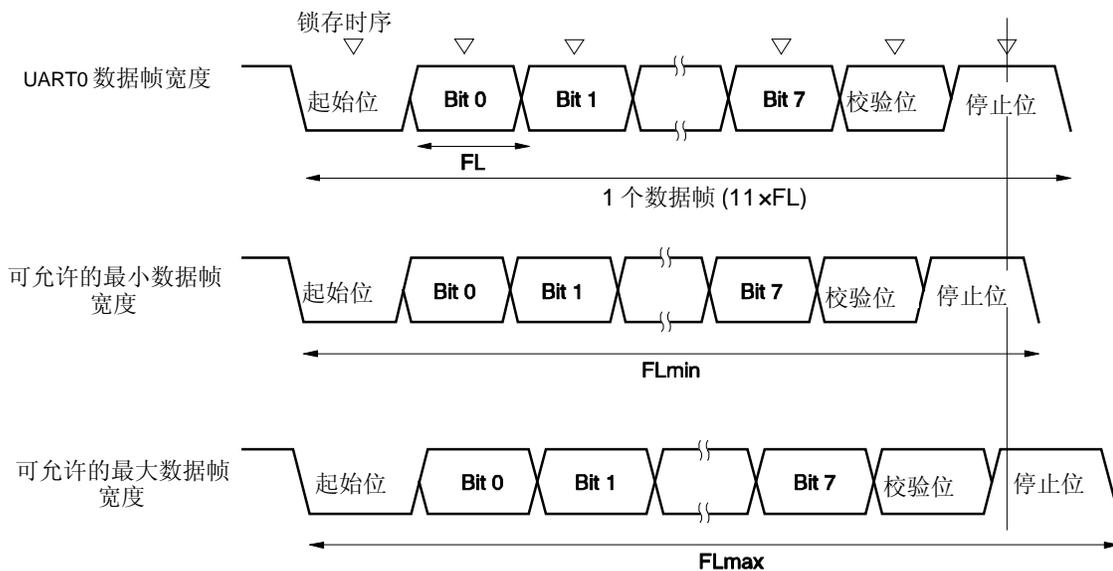
备注 TPS01, TPS00: 波特率发生器控制寄存器 0 (BRGC0) 的第 7 位和第 6 位 (用于设置基本时钟(f<sub>XCLK0</sub>))  
k: 根据 BRGC0 的 MDL04 ~ MDL00 位设置的值 (k = 8, 9, 10, ..., 31)  
f<sub>PRS</sub>: 外围硬件时钟频率  
ERR: 波特率误差

**(4) 接收期间允许的波特率范围**

接收期间来自发送端的波特率可允许的误差范围如下所示。

**注意事项** 必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。

图 13-13. 接收期间允许的波特率范围



如图 13-13 所示，当检测到起始位后，接收数据的锁存时序由波特率发生器控制寄存器 0(BRGCO)设置的计数器来确定。如果数据的最后一位(停止位)满足该锁存时序，则该数据可被正确接收。

假定接收数据为 11 位，各项理论值计算如下。

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

Brate: UART0 的波特率

k: BRGC0 设置的值

FL: 1 位数据宽度

锁存时序的极限值: 2 个时钟

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

因此在发送端设定的可接收的最大波特率如下所示。

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

同样，可计算允许的最大数据帧宽度。

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

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

因此在发送端设定的可接收的最小波特率如下所示。

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

可通过上述最小和最大波特率公式计算在 UART0 与发送端之间允许的波特率误差，如下所示。

表 13-6. 允许的最大/最小波特率误差

分频比 (k)	允许的最大波特率误差	允许的最小波特率误差
8	+3.53%	-3.61%
16	+4.14%	-4.19%
24	+4.34%	-4.38%
31	+4.44%	-4.47%

**备注** 1. 接收时允许的误差与每帧的位数、输入时钟频率和分频比(k)有关。输入时钟频率和分频比(k)越高，允许的误差就越大。

2. k: BRGC0 设置的值。

### 14.1 串行接口 UART6 的功能

串行接口 UART6 有以下两种模式。

#### (1) 操作停止模式

在不进行串行通信时可采用此模式，以降低功耗。

需要了解详细信息，可参见 **14.4.1 操作停止模式**。

#### (2) 异步串行接口(UART) 模式

该模式支持 LIN(局部互联网)-总线。该模式的功能概括如下。

需要了解详细信息，可参见 **14.4.2 异步串行接口(UART)模式** 和 **14.4.3 专用波特率发生器**。

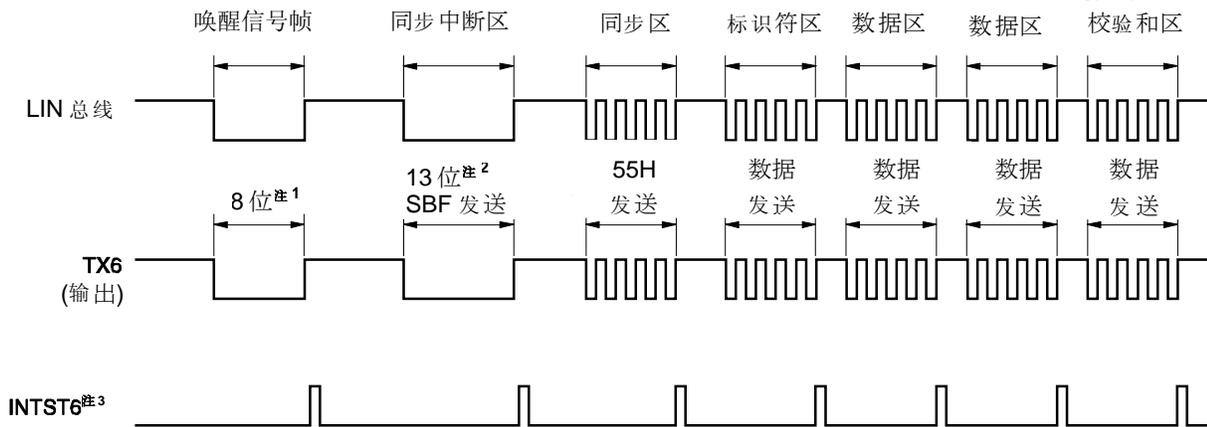
- 最大传输率: 625 kbps
- 双引脚配置 TxD6: 发送数据输出引脚  
RxD6: 接收数据输入引脚
- 可选择通信数据的数据宽度为 7 或 8 位。
- 专用内置 8 位波特率发生器允许设置任意大小的波特率。
- 可分别执行发送和接收操作(全双工操作)。
- 可选 MSB-first 或 LSB-first 通信。
- 反向发送操作
- 同步中断区域发送长度为 13 ~ 20 位
- 可规定多于 11 位的同步中断区域接收(提供 SBF 接收标志位)。

- 注意事项**
1. **TxD6** 输出反向操作仅在发送端进行而不在接收端进行。为了能够使用该功能，接收端应准备好接收已反向的数据。
  2. 如果串行接口 **UART6** 使用的时钟未被停止(例如，处于 **HALT** 模式下)，则继续正常操作。如果串行接口 **UART6** 使用的时钟已被停止(例如，处于 **STOP** 模式下)，则寄存器停止操作，并且在时钟停止之前及时保存数据。**TxD6** 引脚也在时钟停止之前立即保存数据并输出。但在时钟恢复使用后正常操作不能得到保证。因此必须复位电路，使 **POWER6 = 0**、**RxE6 = 0**、**TxE6 = 0**。
  3. 设置 **POWER6 = 1** 然后设置 **TxE6 = 1** (发送)或 **RxE6 = 1** (接收)，以启动通信。
  4. 通过基本时钟( $f_{XCLK6}$ )(由 **CKSR6** 设置)对 **TxE6** 与 **RxE6** 进行同步。若要再次允许发送或接收，必须在 **TxE6** 或 **RxE6** 清零(0)后至少两个基本时钟时将 **TxE6** 或 **RxE6** 设置为 1。如果在两个基本时钟内设置 **TxE6** 或 **RxE6**，则发送电路或接收电路可能不被初始化。
  5. 在设置 **TxE6 = 1** 后至少经历一个基本时钟( $f_{XCLK6}$ )再为 **TxB6** 设置发送数据。
  6. 如果连续发送数据，从停止位到下一个起始位的通信时序将延长两个宏操作时钟。但这不会影响到通信结果，因为接收端会在检测到起始位时对时序初始化。如果 **UART6** 用于 **LIN** 模式，则不能使用连续发送功能。

**备注** LIN 表示局部互联网，是一个低速(1 ~ 20 kbps)串行通信协议，可用于降低网络费用。  
 LIN 是一种单主设备(single-master)通信方式，一个主设备上最多可以连接 15 个从设备。  
 LIN 的从设备用于控制开关、制动器和传感器，这些设备通过 LIN 网与 LIN 主设备相连。  
 通常，LIN 主设备与一个网络，如 CAN(控制器局域网 Controller Area Network)相连。  
 此外，LIN 总线采用单线方式，通过收发器(符合 ISO9141)与各节点相连。  
 在 LIN 协议中，主设备发送带有波特率信息的一帧数据，从设备接收数据并校正波特率误差。因此当从设备端的波特率误差在  $\pm 15\%$  范围内时，可以进行通信。

图 14-1 和 14-2 概括了 LIN 的发送和接收操作。

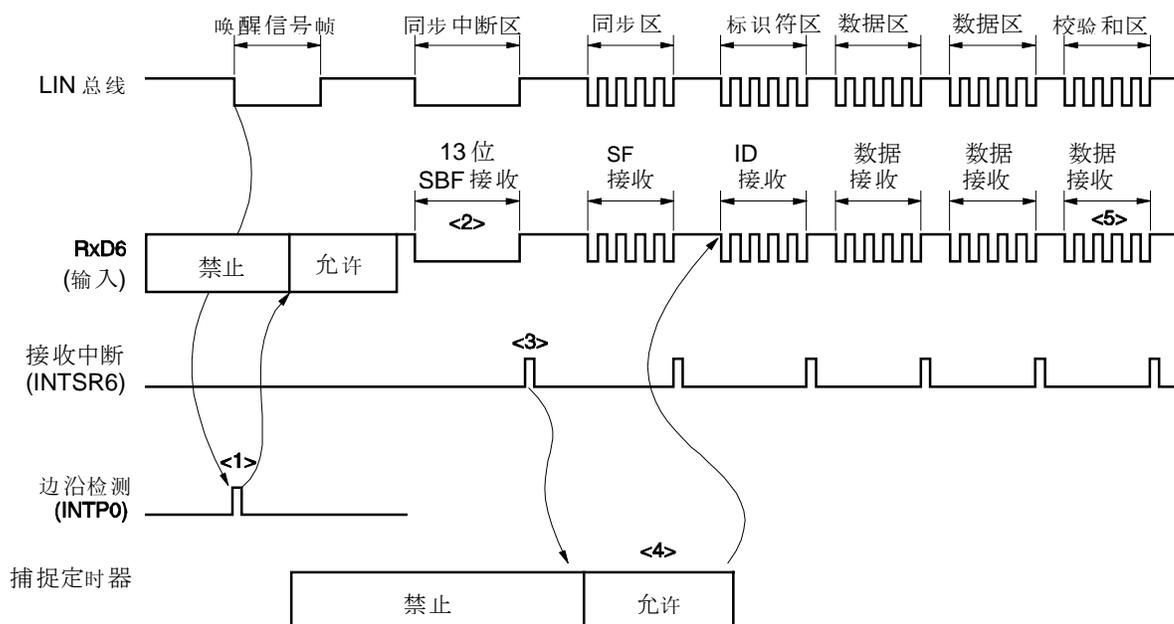
图 14-1. LIN 发送操作



- 注**
1. 在 8 位发送模式中，可通过发送 80H 替代唤醒信号帧。
  2. 使用硬件输出同步中断区。可通过异步串行接口控制寄存器 6(ASICL6)的第 4 位 ~ 第 2 位(SBL62 ~ SBL60)设置输出宽度(等于位长度)(参见 14.4.2 (2) (h) SBF 发送)。
  3. 每次发送完成时输出 INTST6。在发送 SBF 时也输出 INTST6。

**备注** 区域之间的间隔由软件控制。

图 14-2. LIN 接收操作



接收处理过程如下。

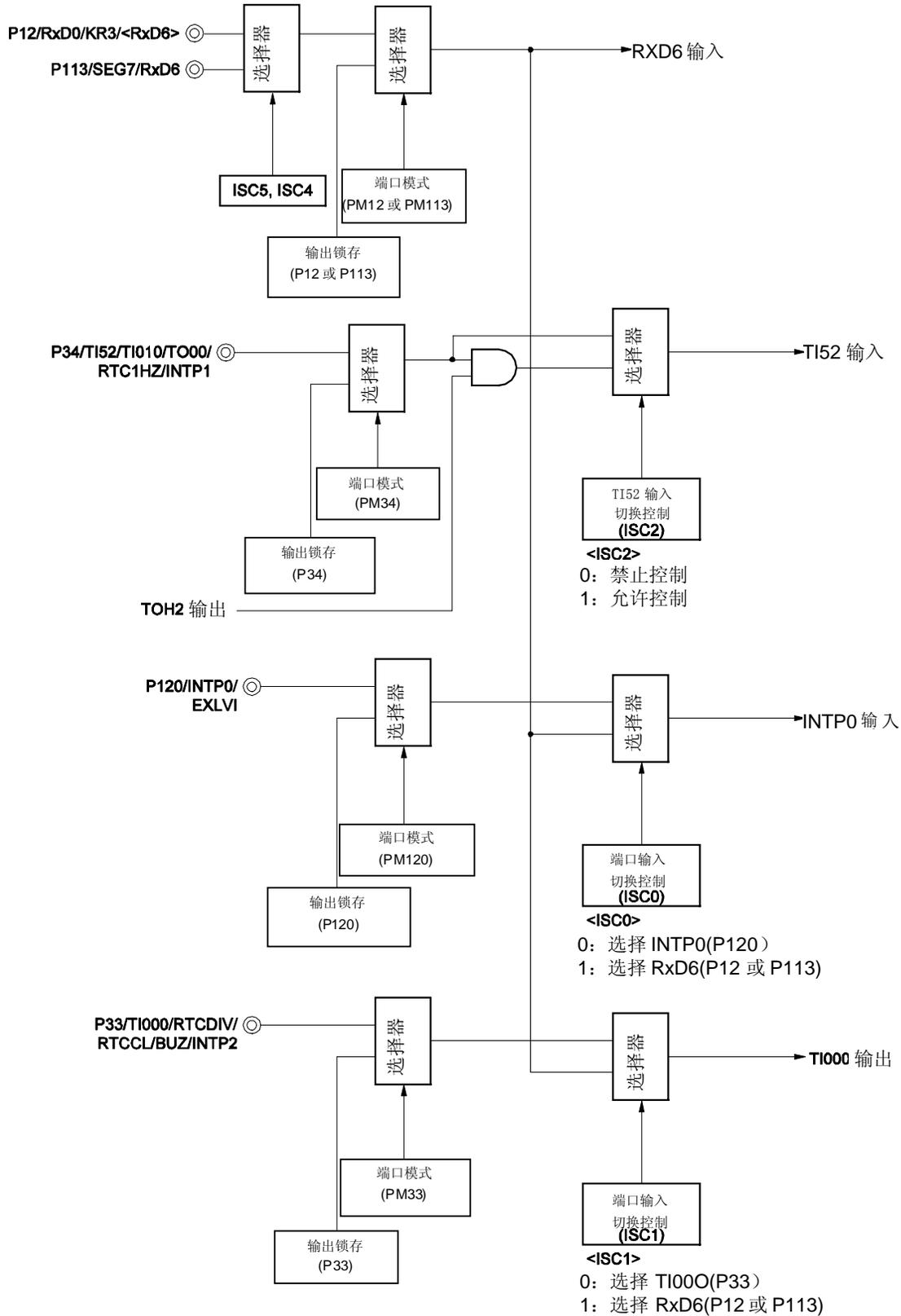
- <1> 在引脚的边沿处检测唤醒信号，允许使用 UART6，并设置 SBF 接收模式。
- <2> 当检测到停止位时才停止接收操作。在检测到至少 11 位低电平数据的 SBF 时，认为 SBF 已被正确接收，并输出一个中断信号。在检测到少于 11 位低电平数据的 SBF 时，认为出现 SBF 接收错误。此时不输出中断信号，而是恢复 SBF 接收模式。
- <3> 如果 SBF 被正确接收，将输出一个中断信号。用这个 SBF 接收完成中断服务来启动 16 位定时器/事件计数器 00 并测量同步区的位长度(脉冲宽度)(参见 6.4.8 脉冲宽度测量操作)。禁止检测 OVE6、PE6 和 FE6 错误，不进行 UART 通信的错误检测，也不执行移位寄存器和 RxB6 之间的数据传送。移位寄存器中保存复位值 FFH。
- <4> 根据同步区的位长度计算波特率误差，在 SF 接收后禁止使用 UART6，然后对波特率发生器控制寄存器 6(BRGC6)重新设置。
- <5> 用软件区分校验和区。在接收到校验和区后通过软件对 UART6 进行初始化，并再次设置 SBF 接收模式。

图 14-3 显示了 LIN 接收操作的端口配置。

通过检测外部中断信号(INTP0)脉冲沿，接收从 LIN 主设备发出的唤醒信号。根据 16 位定时器/事件计数器 00 的外部事件捕捉操作来测量 LIN 主设备发送的同步区的位长度，并计算波特率误差。

不进行外部连接 RxD6 和 INTP0/TI000，而由端口输入切换控制(ISC0/ISC1)可将接收端口的(RxD6)的输入源输入给外部中断(INTP0)和 16 位定时器/事件计数器 00。

图 14-3. LIN 接收操作的端口配置



备注 ISC0, ISC1: 输入切换控制寄存器(ISC)的第0位和第1位 (参见 图 14-11)

以下是 LIN 通信中用到的外部功能。

<用到的外部功能>

- 外部中断(INTP0)：唤醒信号检测  
用途：检测唤醒信号的脉冲沿和通信的开始。
- 16 位定时器/事件计数器 00(TI000)：波特率误差检测  
用途：通过检测同步区(SF)的位长度并按位分频，来检测波特率误差(在捕捉模式下测量 TI000 输入脉冲的间隔)。
- 串行接口 UART6

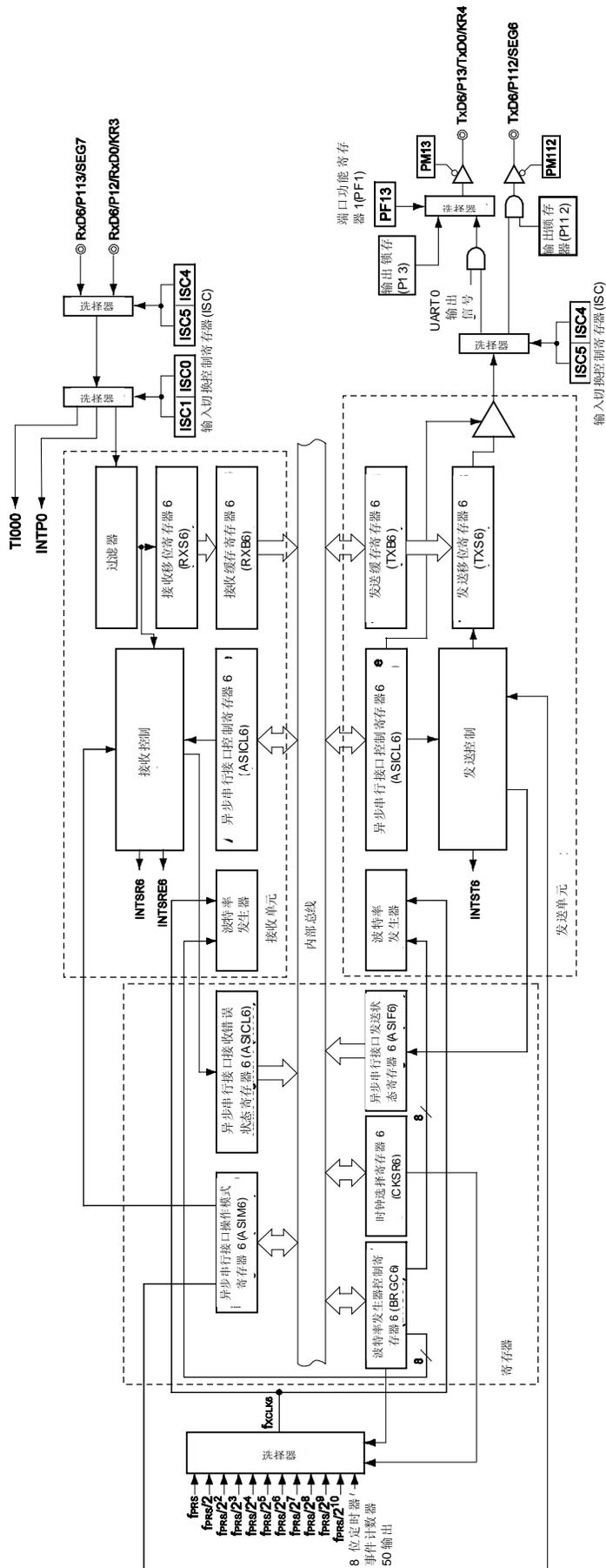
## 14.2 串行接口 UART6 的配置

串行接口 UART6 包括以下硬件。

表 14-1. 串行接口 UART6 的配置

项目	配置
寄存器	接收缓冲寄存器 6 (RXB6) 接收移位寄存器 6 (RXS6) 发送缓冲寄存器 6 (TXB6) 发送移位寄存器 6 (TXS6)
控制寄存器	异步串行接口操作模式寄存器 6 (ASIM6) 异步串行接口接收错误状态寄存器 6 (ASIS6) 异步串行接口发送状态寄存器 6 (ASIF6) 时钟选择寄存器 6 (CKSR6) 波特率发生器控制寄存器 6 (BRGC6) 异步串行接口控制寄存器 6 (ASICL6) 输入切换控制寄存器(ISC) 端口功能寄存器 1 (PF1) 端口模式寄存器 1 (PM1) 端口寄存器 1 (P1) 端口模式寄存器 11 (PM11) 端口寄存器 11 (P11)

图 14-4. 串行接口 UART6 框图



**(1) 接收缓冲寄存器 6 (RXB6)**

该 8 位寄存器用于存储由接收移位寄存器 6(RXS6)移位的并行数据。

每接收到 1 字节的数据，新的接收数据就从 RXS6 传送到该寄存器中。如果数据宽度被设定为 7 位，则按如下方式发送数据。

- 在 LSB-first 接收模式下，将接收到的数据传送到 RXB6 的第 0 位 ~ 第 6 位，而 RXB6 的 MSB 位恒为 0。
- 在 MSB-first 接收模式下，将接收到的数据传送到 RXB6 的第 1 位 ~ 第 7 位，而 RXB6 的 LSB 位恒为 0。

如果出现溢出错误(OVE6)，则不把接收数据传送到 RXB6 中。

可由 8 位存储器操作指令读取 RXB6 的内容。不能将数据写入该寄存器中。

复位信号产生将该寄存器设置为 FFH。

**(2) 接收移位寄存器 6 (RXS6)**

该寄存器将输入到 RxD6 引脚的串行数据转换成并行数据。

程序不能直接使用 RXS6。

**(3) 发送缓冲寄存器 6 (TXB6)**

该寄存器用于设置发送数据。当数据写入 TXB6 时开始发送。

可由 8 位存储器操作指令对 TXB6 进行读或写操作。

复位信号产生将该寄存器设置为 FFH。

**注意事项** 1. 当异步串行接口发送状态寄存器 6(ASIF6)的第 1 位(TXBF6)=1 时不要对 TXB6 进行写操作。

2. 在通信过程中(当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)和第 6 位(TXE6)均为 1，或 ASIM6 的第 7 位(POWER6)和第 5 位(RXE6)均为 1)，不要通过软件刷新(写入相同值)TXB6。

3. 在设置 TXE6 = 1 后，至少经过一个基本时钟(f<sub>XCLK6</sub>)，将发送数据写入 TXB6。

**(4) 发送移位寄存器 6 (TXS6)**

该寄存器将 TXB6 中要传送的数据作为串行数据从 TxD6 移出。在第 1 次发送过程中，当数据写入 TXB6 后立即被传送。或者在持续发送过程中，当发出 1 帧数据后，INTST6 中断产生之前数据立即被传送。利用基本时钟的下降沿从 TxD6 引脚传送来自 TXB6 的数据。

程序不能直接对 TXS6 操作。

### 14.3 控制串行接口 UART6 的寄存器

串行接口 UART6 由以下 9 个寄存器控制。

- 异步串行接口操作模式寄存器 6 (ASIM6)
- 异步串行接口接收错误状态寄存器 6 (ASIS6)
- 异步串行接口发送状态寄存器 6 (ASIF6)
- 时钟选择寄存器 6 (CKSR6)
- 波特率发生器控制寄存器 6 (BRGC6)
- 异步串行接口控制寄存器 6 (ASICL6)
- 输入切换控制寄存器 (ISC)
- 端口功能寄存器 1 (PF1)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)
- 端口模式寄存器 11 (PM11)
- 端口寄存器 11 (P11)

#### (1) 异步串行接口操作模式寄存器 6 (ASIM6)

该 8 位寄存器用于控制串行接口 UART6 的串行通信。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号产生将该寄存器设置为 01H。

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

图 14-5. 异步串行接口操作模式寄存器 6 (ASIM6)的格式 (1/2)

地址: FF50H 复位后: 01H R/W

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

- 注**
1. 发送期间当 POWER6 = 0 时, TxD6 引脚的输出变为高电平, 而来自 RxD6 引脚的输入恒为高电平。
  2. 对异步串行接口接收错误状态寄存器 6(ASIS6)、异步串行接口发送状态寄存器 6(ASIF6)、异步串行接口控制寄存器 6(ASICL6)的第 7 位(SBRF6)和第 6 位(SBRT6), 以及接收缓冲寄存器 6(RXB6)进行复位。

图 14-5. 异步串行接口操作模式寄存器 6 (ASIM6)的格式 (2/2)

PS61	PS60	发送操作	接收操作
0	0	不输出校验位	无校验接收
0	1	输出零校验	按零校验接收 <sup>注</sup>
1	0	输出奇校验	按奇校验判断
1	1	输出偶校验	按偶校验判断

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

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

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

**注** 如果选择“按零校验接收”，则无法判断校验。因此不要设置异步串行接口接收错误状态寄存器 0(ASIS6)的第 2 位(PE6)，这样也不会产生出错中断。

- 注意事项**
1. 启动发送操作时，对 POWER6 置 1，然后对 TXE6 置 1。停止发送操作时，先将 TXE6 清零，然后将 POWER6 清零。
  2. 启动接收操作时，对 POWER6 置 1，然后对 RXE6 置 1。停止接收操作时，先将 RXE6 清零，然后将 POWER6 清零。
  3. 当 RxD6 引脚输入为高电平时，先对 POWER6 置 1，再对 RXE6 置 1。当输入为低电平 POWER6 置 1，RXE6 置 1 时，启动接收操作。
  4. 通过基本时钟(f<sub>XCLK6</sub>)(由 CKSR6 设置)对 TXE6 与 RXE6 进行同步。若要再次允许发送或接收，必须在 TXE6 或 RXE6 清零(0)后至少两个基本时钟时将 TXE6 或 RXE6 设置为 1。如果在两个基本时钟内设置 TXE6 或 RXE6，则发送电路或接收电路可能不被初始化。
  5. 设置 TXE6 = 1 后，至少经过一个基本时钟(f<sub>XCLK6</sub>)，将发送数据写入 TXB6。
  6. 在对 PS61、PS60 和 CL6 位重写之前，先将 TXE6 和 RXE6 位清零。
  7. 工作在 LIN 模式时，PS61 与 PS60 恒为 0。
  8. 在重写 SL6 位时，必须确保 TXE6 = 0。总是以“停止位的个数=1”的格式执行接收操作，因此接收不会受到 SL6 设置值的影响。
  9. 当重写 ISRM6 时，RXE6 必须为 0。

**(2) 异步串行接口接收错误状态寄存器 6 (ASIS6)**

该寄存器用于指示串行接口 UART6 的接收操作完成时出现的错误状态，包括 3 个出错标志位(PE6, FE6, OVE6)。

可由 8 位存储器操作指令对该寄存器进行只读操作。

如果 ASIM6 的第 5 位(RXE6)或第 7 位(POWER6) = 0，则复位信号产生将该寄存器设置为 00H。读取该寄存器的值为 00H。如果产生接收错误，读取 ASIS6 再读取接收缓冲寄存器 6 (RXB6)，可将出错标志清零。

图 14-6. 异步串行接口接收错误状态寄存器 6 (ASIS6)的格式

地址: FF53H 复位后: 00H R

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

PE6	指示校验错误的状态标志位
0	如果 POWER6 = 0 且 RXE6 = 0，或者如果读取 ASIS6
1	如果发送数据的校验位与接收完成时的校验位不相等

FE6	指示帧错误的状态标志位
0	如果 POWER6 = 0 且 RXE6 = 0、或者如果读取 ASIS6
1	如果接收完成时未检测到停止位

OVE6	指示溢出错误的状态标志位
0	如果 POWER6 = 0 且 RXE6 = 0、或 ASIS6 被读取
1	如果将接收数据存于 RXB6 寄存器中且在读取该数据前下一个接收操作已完成

- 注意事项**
1. 根据异步串行接口操作模式寄存器 6(ASIM6)的 PS61 和 PS60 位的设置值不同，对 PE6 位的操作也有所不同。
  2. 无论停止位的个数是多少，只将接收数据的第 1 位作为停止位进行检测。
  3. 如果出现溢出错误，则下一个接收数据不写入接收缓冲寄存器 6(RXB6)中而是被忽略。
  4. 如果从 ASIS6 读取数据，会产生一个等待周期。当 CPU 采用副系统时钟而停止外围硬件时钟操作时，不要从 ASIS6 读取数据。需要了解详细信息，可参见 第二十九章 等待注意事项。

**(3) 异步串行接口发送状态寄存器 6 (ASIF6)**

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

当数据从 TXB6 传送到 TXS6 中后，通过将下一个发送数据写入 TXB6，使得发送能够连续进行而不被中断，即便是在中断产生期间也不受影响。

可由 8 位存储器操作指令对该寄存器进行只读操作。

如果 ASIM6 的第 7 位(POWER6)或第 6 位(TXE6)置 0，则复位信号产生将该寄存器设置为 00H。

图 14-7. 异步串行接口发送状态寄存器 6 (ASIF6)的格式

地址: FF55H 复位后: 00H R

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

TXBF6	发送缓冲数据标志位
0	如果 POWER6 = 0 或 TXE6 = 0，或如果数据被传送到发送移位寄存器 6(TXS6)中
1	当数据被写入发送缓冲寄存器 6(TXB6)中(如果数据已在 TXB6 中)

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

- 注意事项**
1. 为了能够连续发送数据，应将第一个发送的数据(第一个字节)写入 TXB6 中。之后，检查 TXBF6 标志位，必须为“0”。如果正确，将下一个发送数据(第二个字节)写入 TXB6 中。如果 TXBF6 标志为“1”时将数据写入 TXB6，则无法保证发送数据的正确性。
  2. 如果要在连续发送结束时初始化发送单元，则在产生发送完成中断后必须检查 TXSF6 标志位，应为“0”。然后执行初始化操作。如果 TXSF6 标志为“1”时执行初始化操作，则无法保证发送数据的正确性。

**(4) 时钟选择寄存器 6 (CKSR6)**

该寄存器用于选择串行接口 UART6 的基本时钟。

可由 8 位存储器操作指令设置 CKSR6。

复位信号产生将该寄存器清零(00H)。

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

图 14-8. 时钟选择寄存器 6 (CKSR6)的格式

地址: FF56H 复位后: 00H R/W

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

TPS63	TPS62	TPS61	TPS60	基时钟 (f <sub>CLK6</sub> ) 选择 <sup>#1</sup>				
				f <sub>PRS</sub> = 2 MHz	f <sub>PRS</sub> = 5 MHz	f <sub>PRS</sub> = 8 MHz	f <sub>PRS</sub> = 10 MHz	
0	0	0	0	F <sub>PRS</sub> <sup>#2</sup>	2 MHz	5 MHz	8 MHz	10 MHz
0	0	0	1	f <sub>PRS</sub> /2	1 MHz	2.5 MHz	4 MHz	5 MHz
0	0	1	0	f <sub>PRS</sub> /2 <sup>2</sup>	500 kHz	1.25 MHz	2 MHz	2.5 MHz
0	0	1	1	f <sub>PRS</sub> /2 <sup>3</sup>	250 kHz	625 kHz	1 MHz	1.25 MHz
0	1	0	0	f <sub>PRS</sub> /2 <sup>4</sup>	125 kHz	312.5 kHz	500 kHz	625 kHz
0	1	0	1	f <sub>PRS</sub> /2 <sup>5</sup>	62.5 kHz	156.25 kHz	250 kHz	312.5 kHz
0	1	1	0	f <sub>PRS</sub> /2 <sup>6</sup>	31.25 kHz	78.13 kHz	125 kHz	156.25 kHz
0	1	1	1	f <sub>PRS</sub> /2 <sup>7</sup>	15.625 kHz	39.06 kHz	62.5 kHz	78.13 kHz
1	0	0	0	f <sub>PRS</sub> /2 <sup>8</sup>	7.813 kHz	19.53 kHz	31.25 kHz	39.06 kHz
1	0	0	1	f <sub>PRS</sub> /2 <sup>9</sup>	3.906 kHz	9.77 kHz	15.625 kHz	19.53 kHz
1	0	1	0	f <sub>PRS</sub> /2 <sup>10</sup>	1.953 kHz	4.88 kHz	7.513 kHz	9.77 kHz
1	0	1	1	TM50 输出 <sup>#3</sup>				
其它情况				禁止设置				

- 注**
1. 当外围硬件时钟 (f<sub>PRS</sub>) 运行于高速系统时钟(f<sub>XH</sub>) (XSEL = 1), f<sub>PRS</sub> 操作频率依据供电电压的不同而不同。
    - V<sub>DD</sub> = 2.7 ~ 5.5 V: f<sub>PRS</sub> ≤ 10 MHz
    - V<sub>DD</sub> = 1.8 ~ 2.7 V: f<sub>PRS</sub> ≤ 5 MHz
  2. 如果外围硬件时钟 (f<sub>PRS</sub>) 运行于内部高速振荡时钟(f<sub>RH</sub>) (XSEL = 0), 当 1.8 V ≤ V<sub>DD</sub> < 2.7 V, TPS63 = TPS62 = TPS61 = TPS60 = 0 (基时钟: f<sub>PRS</sub>) 这种设置禁止。
  3. 选择 TM50 输出作为基本时钟时,先启动 8 位定时器/事件计数器 50, 然后允许定时器 F/F 反转操作 (TMC501 = 1)。

**注意事项** 在重写 TPS63 ~ TPS60 位时, 必须确保 POWER6 = 0。

**备注** f<sub>PRS</sub>: 外围硬件时钟频率

**(5) 波特率发生器控制寄存器 6 (BRGC6)**

该寄存器用于设置串行接口 UART6 的 8 位计数器的分频值。

可由 8 位存储器操作指令设置 BRGC6。

复位信号产生将该寄存器设置为 FFH。

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

图 14-9. 波特率发生器控制寄存器 6 (BRGC6)的格式

地址: FF57H 复位后: FFH R/W

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

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	k	8 位计数器输出时钟选择
0	0	0	0	0	0	×	×	×	禁止设置
0	0	0	0	0	1	0	0	4	$f_{XCLK6}/4$
0	0	0	0	0	1	0	1	5	$f_{XCLK6}/5$
0	0	0	0	0	1	1	0	6	$f_{XCLK6}/6$
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	$f_{XCLK6}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK6}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK6}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK6}/255$

**注意事项**

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

**备注**

1.  $f_{XCLK6}$ : 根据 CKSR6 寄存器的 TPS63 ~ TPS60 位选择基本时钟的频率
2. k: 由 MDL67 ~ MDL60 位设置的值 (k = 8, 9, 10, ..., 255)
3. ×: 不必考虑

**(6) 异步串行接口控制寄存器 6 (ASICL6)**

该寄存器用于控制串行接口 UART6 的串行通信操作。

可由 1 位或 8 位存储器操作指令设置 ASICL6。

复位信号产生将该寄存器设置为 16H。

**注意事项** 通信过程中(当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)和第 6 位(TXE6)均为 1, 或 ASIM6 的第 7 位(POWER6)和第 5 位(RXE6)均为 1), 可由软件刷新(写入相同值)ASICL6。但是, 由于可能会重新触发 SBF 接收和 SBF 发送, 因此在 SBF 接收期间(SBRT6 = 1)或 SBF 发送期间通过刷新操作不要将 SBRT6 与 SBTT6 都设置为 1(直到由于 SBTT6=1 而产生 INTST6 为止)。

图 14-10. 异步串行接口控制寄存器 6 (ASICL6) 的格式(1/2)

地址: FF58H 复位后: 16H R/W<sup>※</sup>

符号	<7>	<6>	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6

SBRF6	SBF 接收状态标志
0	如果 POWER6 = 0 且 RXE6 = 0, 或者如果 SBF 已正确接收
1	SBF 接收操作正在进行中

SBRT6	SBF 接收触发
0	-
1	SBF 接收触发

SBTT6	SBF 发送触发
0	-
1	SBF 发送触发

注 第 7 位是只读的。

图 15-10. 异步串行接口控制寄存器 6 (ASICL6)的格式 (2/2)

SBL62	SBL61	SBL60	SBF 发送输出宽度控制
1	0	1	SBF 输出宽度: 13 位
1	1	0	SBF 输出宽度: 14 位
1	1	1	SBF 输出宽度: 15 位
0	0	0	SBF 输出宽度: 16 位
0	0	1	SBF 输出宽度: 17 位
0	1	0	SBF 输出宽度: 18 位
0	1	1	SBF 输出宽度: 19 位
1	0	0	SBF 输出宽度: 20 位

DIR6	首位说明
0	MSB
1	LSB

TXDLV6	允许/禁止反向 TxD6 输出
0	TxD6 正常输出
1	TxD6 反向输出

- 注意事项**
1. 如果出现 SBF 接收错误, 则返回到 SBF 接收模式, 并保持 SBRF6 标志位状态(1)不变。
  2. 在设置 SBRT6 之前, 必须确保 ASIM6 的第 7 位(POWER6)和第 5 位(RXE6)为 1。在设置 SBRT6=1 后 SBF 接收完成(产生中断请求信号之前)之前不要将 SBRT6 清零。
  3. SBRT6 的读取值恒为 0。在 SBF 正确接收后 SBRT6 自动清零。
  4. 在设置 SBTT6=1 之前, 必须确保 ASIM6 的第 7 位(POWER6)和第 6 位(TXE6)为 1。在设置 SBTT6=1 之后 SBF 接收完成(产生中断请求信号之前)之前不要将 SBTT6 清零。
  5. SBTT6 的读取值恒为 0。并在 SBF 发送结束后自动清零。
  6. 在接收期间不要设置 SBRT6=1, 而在发送期间也不要设置 SBTT6=1。
  7. 在重写 DIR6 和 TXDLV6 之前, 应对 TXE6 和 RXE6 清零。

**(7) 输入切换控制寄存器(ISC)**

通过设置 ISC5 为 1, UART6 I/O 引脚从 P113/SEG7/RxD6 和 P112/SEG6/TxD6 转换为 P12/RxD0/KR3/RxD6 和 P13/TxD0/KR4/TxD6。

通过设置 ISC3 为 1, P113/SEG7/RxD6 引脚允许输入。当 ISC3 清零 0 时, 外部输入不响应。因此, 复位释放后, 可以避免由于不确定的状态而产生电流流过此引脚, 直到进行输出设置为止。

输入切换控制寄存器(ISC)在 LIN(Local Interconnect Network 本地互连网络)接收期间用于接收从主设备发送的状态信号。

当 ISC0 与 ISC1=1 时, 选择 P12/RxD0/KR3/RxD6 或 P113/SEG7/RxD6 引脚的信号输入作为 INTP0 与 TI000 的输入源。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号产生将该寄存器清零(00H)。

图 14-11. 输入切换控制寄存器(ISC)的格式

地址: FF4FH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ISC	0	0	ISC5	ISC4	ISC3	ISC2	ISC1	ISC0

ISC5	ISC4	TxD6, RxD6 输入源选择
0	0	TxD6:P112, RxD6: P113
1	0	TxD6:P13, RxD6: P12
其它情况		禁止设置

ISC3	RxD6/P113 输入允许/禁止
0	RxD6/P113 输入禁止
1	RxD6/P113 输入允许

ISC2	TI52 输入源控制
0	不允许控制 TI52 输入 (P34)
1	允许控制 TI52 输入 (P34) <sup>注 1</sup>

ISC1	TI000 输入源选择
0	TI000 (P33)
1	RxD6 (P12 或 P113) <sup>注 2</sup>

ISC0	INTP0 输入源选择
0	INTP0 (P120)
1	RxD6 (P12 或 P113) <sup>注 2</sup>

注 1. TI52 输入由 TOH2 输出信号控制。

2. TI000 和 INTP0 输入可以由 ISC5 和 ISC4 选定。

**注意事项** 复位释放后, 当将 P113/SEG7/RxD6 引脚用作 P113 ~ RxD6 引脚时, 把 PF11ALL 设置为 0 并且将 ISC3 设置为 1,  
复位释放后, 当将 P113/SEG7/RxD6 引脚用作 SEG7 引脚时, 把 PF11ALL 设置为 1 并且将 ISC3 设置为 0。

### (8) 端口功能寄存器 1 (PF1)

此寄存器设置 P13/TxD0/KR4/TxD6 引脚的引脚功能。

可由 1-位或 8-位存储器操作指令进行设置。

复位信号将此寄存器 PF1 清零 00H。

图 14-12. 端口功能寄存器 1(PF1)的格式

地址: FF20H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PF1	0	0	0	0	PF13	0	0	0

PF13	Port (P13), 按键中断 (KR4), UART0, UART6 输出规定
0	用作 P13 或 KR4
1	用作 TxD0 ~ TxD6

### (9) 端口模式寄存器 11 (PM1)

该寄存器可以按位设置端口 1 的输入/输出模式。

当 P13/TxD0/KR4/TxD6 引脚用于串行接口数据输出时, 将 PM13 清零且 P13 的输出锁存器的值设置为 1。

当 P12/RxD0/KR3/RxD6 引脚用于串行接口数据输入时, 将 PM12 置 1。此时 P12 的输出锁存器的值= 0 或 1。

可由 1 位或 8 位存储器操作指令设置 PM1。

复位信号产生将该寄存器设置为 FFH。

图 14-13. 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM1	1	1	1	1	PM13	PM12	1	1

PM1n	P1n 引脚 I/O 模式选择 (n = 2, 3)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

**(10) 端口模式寄存器 11 (PM11)**

该寄存器可以按位设置端口 1 的输入/输出模式。

当 P112/SEG6/TxD6 引脚用于串行接口数据输出时，将 PM112 清零且 P112 的输出锁存器的值设置为 1。

当 P113/SEG7/RxD6 引脚用于串行接口数据输入时，将 PM113 置 1。此时 P113 的输出锁存器的值= 0 或 1。

可由 1 位或 8 位存储器操作指令设置 PM11。

复位信号产生将该寄存器设置为 FFH。

图 14-14. 端口模式寄存器 11 (PM11) 的格式

地址: FF2BH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM11	1	1	1	1	PM113	PM112	1	1

PM11n	P11n 引脚 I/O 模式选择 (n = 2, 3)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

## 14.4 串行接口 UART6 的操作

串行接口 UART6 有以下两种模式。

- 操作停止模式
- 异步串行接口(UART) 模式

### 14.4.1 操作停止模式

在此模式下，不能进行串行通信，因此，可降低功耗。此外，引脚可作为一般的端口引脚使用。通过将 ASIM6 的第 7 位、第 6 位和第 5 位(POWER6、TXE6 和 RXE6)清零，可设置操作停止模式。

#### (1) 使用的寄存器

由异步串行接口操作模式寄存器 6(ASIM6)设置操作停止模式。

可由 1 位或 8 位存储器操作指令设置 ASIM6。

复位信号产生将该寄存器设置为 01H。

地址: FF50H 复位后: 01H R/W

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

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

**注意事项** TXE6 和 RXE6 清零后再对 POWER6 清零，可停止操作。

启动通信时，须先设置 POWER6=1，然后设置 TXE6=1 或 RXE6=1。

**备注** 如果将 RxD6/P14 和 TxD6/P13 作为通用端口引脚使用，可参见 第四章 端口功能。

#### 14.4.2 异步串行接口(UART)模式

此模式下，在传送了起始位后，紧接着发送/接收 1 字节数据，然后可执行全双工操作。  
可使用专用 UART 波特率发生器，这样通信选择的波特率范围较大。

##### (1) 使用的寄存器

- 异步串行接口操作模式寄存器 6 (ASIM6)
- 异步串行接口接收错误状态寄存器 6 (ASIS6)
- 异步串行接口发送状态寄存器 6 (ASIF6)
- 时钟选择寄存器 6 (CKSR6)
- 波特率发生器控制寄存器 6 (BRGC6)
- 异步串行接口控制寄存器 6 (ASICL6)
- 输入切换控制寄存器(ISC)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

在 UART 模式中设置操作的基本过程如下。

- <1> 设置 CKSR6 寄存器 (参见图 14-8)
- <2> 设置 BRGC6 寄存器 (参见图 14-9)
- <3> 设置 ASIM6 的第 0 位~第 4 位 (ISRM6、SL6、CL6、PS60、PS61) (参见图 14-5)
- <4> 设置 ASICL6 的第 0 位和第 1 位 (TXDLV6、DIR6) (参见图 14-10)
- <5> 设置 ASIM6 的第 7 位 (POWER6) = 1
- <6> 设置 ASIM6 的第 6 位 (TXE6) = 1。 → 允许发送  
设置 ASIM6 的第 5 位 (RXE6) = 1。 → 允许接收
- <7> 将数据写入发送缓冲寄存器 6 (TXB6) → 开始发送数据

**注意事项** 在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。

寄存器设置与引脚之间的关系如下所示。

表 14-2. 寄存器设置与引脚之间的关系

(a) 当由 ISC 寄存器的位 4, 5(ISC4, ISC5)选定 P12 和 P13 用作 UART6 时

POWER6	TXE6	RXE6	PM13	P13	PM12	P12	UART6 操作	引脚功能	
								TxD6/KR4/TxD0/P13	RxD6/KR3/RxD0/P12
0	0	0	×	×	×	×	停止	KR4/TxD0/P13	KR3/RxD0/P12
1	0	1	×	×	1	×	接收	KR4/P13	RxD6
	1	0	0	×	×	×	传输	TxD6	KR3/P12
	1	1	0	×	1	×	传输/ 接收	TxD6	RxD6

注 可以用作端口功能, 按键中断, 或串行接口 UART0 (只有当 UART6 停止时)。

注意事项 TxD6/SEG6/P112 和 RxD6/SEG7/P113 引脚用作 SEG6/P112 和 SEG7/P113。

备注 ×: 不必考虑  
 POWER6: 异步串行接口操作模式寄存器 6(ASIM6)的第 7 位  
 TXE6: ASIM6 的第 6 位  
 RXE6: ASIM6 的第 5 位  
 PM1×: 端口模式寄存器  
 P1×: 端口输出锁存器

(b) 当由 ISC 寄存器的位 4, 5(ISC4, ISC5)选定 P112 和 P113 用作 UART6 时

POWER6	TXE6	RXE6	PM112	P112	PM113	P113	UART6 操作	引脚功能	
								TxD6/SEG6/P112	RxD6/SEG7/P113
0	0	0	×	×	×	×	停止	SEG6/P112	SEG7/P113
1	0	1	×	×	1	×	接收	SEG6/P112	RxD6
	1	0	0	1	×	×	传输	TxD6	SEG7/P113
	1	1	0	1	1	×	传输/ 接收	TxD6	RxD6

注 可以用作端口寄存器或 segment 输出。

注意事项 TxD6/KR4/TxD0/P13 和 RxD6/KR3/RxD0/P12 引脚用作 KR4/TxD0/P13 和 KR3/RxD0/P12。

备注 ×: 不必考虑  
 POWER6: 异步串行接口操作模式寄存器 6(ASIM6)的第 7 位  
 TXE6: ASIM6 的第 6 位  
 RXE6: ASIM6 的第 5 位  
 PM1×: 端口模式寄存器  
 P1×: 端口输出锁存器

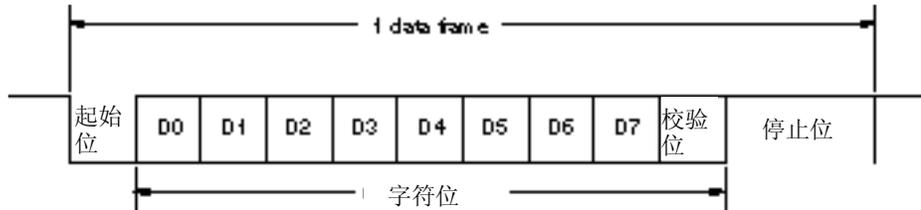
## (2) 通信操作

## (a) 正常发送/接收数据的格式和波形示例

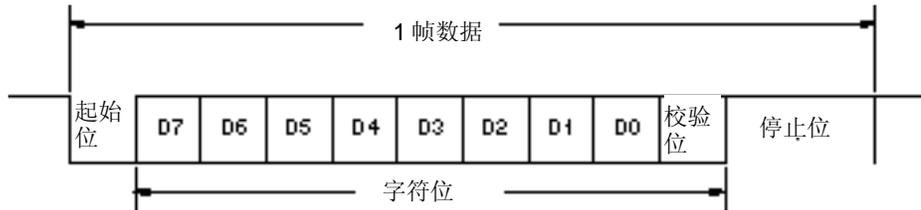
图 14-15 和 14-16 为正常发送/接收数据的格式和波形示例。

图 14-15. 正常 UART 发送/接收数据的格式

## 1. LSB-first 发送/接收



## 2. MSB-first 发送/接收



一帧数据由以下各位组成。

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

由异步串行接口操作模式寄存器 6(ASIM6)规定一个数据帧中字符位的宽度、校验方式和停止位的宽度。

由异步串行接口控制寄存器 6 (ASICL6)的第 1 位(DIR6)规定数据通信方式(LSB-first 或 MSB-first)。

由 ASICL6 的第 0 位(TXDLV6)规定 Tx D6 引脚输出的是正常数据还是反向数据。

图 14-16. 正常 UART 发送/接收数据波形示例

1. 数据宽度: 8 位、LSB-first, 校验位: 偶校验, 停止位: 1 位, 通信数据: 55H



2. 数据宽度: 8 位、MSB-first, 校验位: 偶校验, 停止位: 1 位, 通信数据: 55H



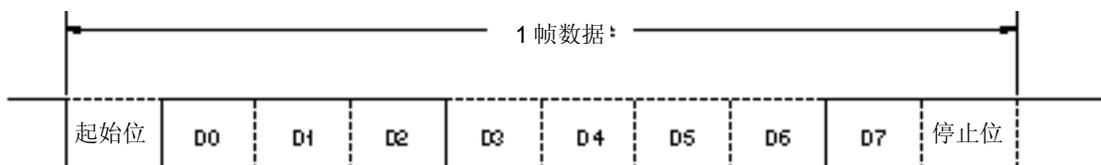
3. 数据宽度: 8 位、MSB-first, 校验位: 偶校验, 停止位: 1 位, 通信数据: 55H, TxD6 引脚输出反向数据



4. 数据宽度: 7 位、LSB-first, 校验位: 奇校验, 停止位: 2 位, 通信数据: 36H



5. 数据宽度: 8 位、LSB-first, 校验位: 无, 停止位: 1 位, 通信数据: 87H



**(b) 校验方式与操作**

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

**注意事项** 在 LIN 模式下，PS61 与 PS60 恒为 0。

**(i) 偶校验**

## • 发送

控制发送数据，包括校验位，使得数据中“1”的个数为偶数。

校验位的取值如下。

如果发送数据有奇数个“1”: 1

如果发送数据有偶数个“1”: 0

## • 接收

计算接收数据(包括校验位)中“1”的个数。如果“1”的个数为奇数，则产生校验错。

**(ii) 奇校验**

## • 发送

跟偶校验不同的是，控制发送数据，包括校验位，使得数据中“1”的个数为奇数。

如果发送数据有奇数个“1”: 0

如果发送数据有偶数个“1”: 1

## • 接收

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

**(iii) 零校验**

无论发送数据为何值，发送时校验位清零。

接收数据时不检测校验位。因此无论校验位为何值，都不会产生校验错。

**(iv) 无校验**

发送数据中没有校验位。

进行接收操作时认为接收数据中没有校验位。因此也不会产生校验错。

**(c) 正常发送**

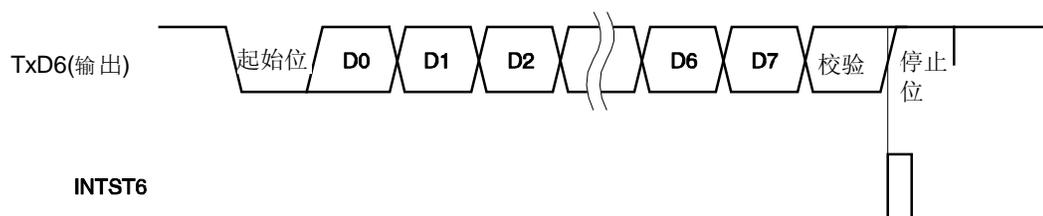
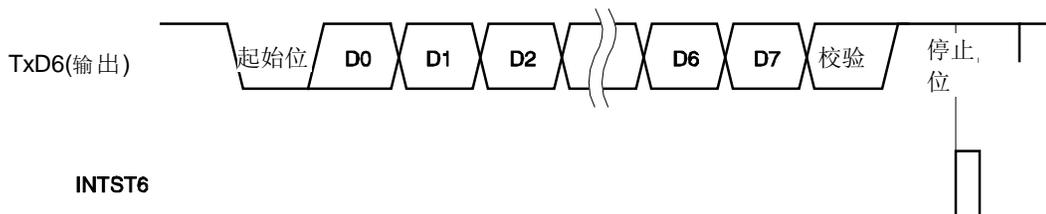
在设置异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)=1, 然后设置 ASIM6 的第 6 位(TXE6)=1 时, 则允许发送。将待发送的数据写入发送缓冲寄存器 6(TXB6)中, 可启动发送操作。起始位、校验位和停止位会自动被添加到数据中。

启动发送操作时, TXB6 中的数据被传送到发送移位寄存器 6(TXS6)中, 然后数据依次从 TXS6 被输出到 TxD6 引脚。当发送结束时, 由 ASIM6 设置的校验位和停止位被添加到数据中, 并产生发送完成中断请求 (INTST6)。

当下一个发送数据写入 TXB6 时, 发送操作才结束。

图 14-17 显示了发送完成中断请求(INTST6)的时序。最后一个停止位一输出就产生该中断。

图 14-17. 正常的发送完成中断请求时序

**1. 停止位宽度: 1****2. 停止位宽度: 2**

**(d) 连续发送**

当发送移位寄存器 6 (TXS6) 一启动移位操作，就可以把下一个发送数据写入发送缓冲寄存器 6(TXB6)。因此，当发送完一帧数据后，即使在执行 INTST6 中断服务时也能够连续发送数据，从而实现高效率的通信速率。此外，当产生发送完成中断时通过读取 ASIF6 的第 0 位(TXSF6)，可以对 TXB6 进行两次有效的写操作(2 个字节)而无须等待发送一帧数据的时间。

为了能够连续发送数据，必须根据 ASIF6 检查发送状态，判断是否可以对 TXB6 进行写操作，如果可以，则写入数据。

- 注意事项**
1. 在连续发送期间，ASIF6 寄存器的 TXBF6 和 TXSF6 标志从“10”变成“11”，再变成“01”。因此检查状态时，不能通过 TXBF6 和 TXST6 来判断。在连续发送期间只读 TXBF6 标志。
  2. 在 LIN 模式下，不能使用连续发送功能。在将发送数据写入发送缓冲寄存器 6(TXB6)之前，异步串行接口发送状态寄存器 6(ASIF6)的值必须为 00H。

TXBF6	对 TXB6 的写操作
0	允许写操作
1	禁止写操作

**注意事项** 为了能够连续发送数据，应将第一个发送的数据(第一个字节)写入 TXB6 中。之后，检查 TXBF6 标志位，必须为“0”。如果正确，将下一个发送数据(第二个字节)写入 TXB6 中。如果 TXBF6 标志为“1”时将数据写入 TXB6，则无法保证发送数据的正确性。

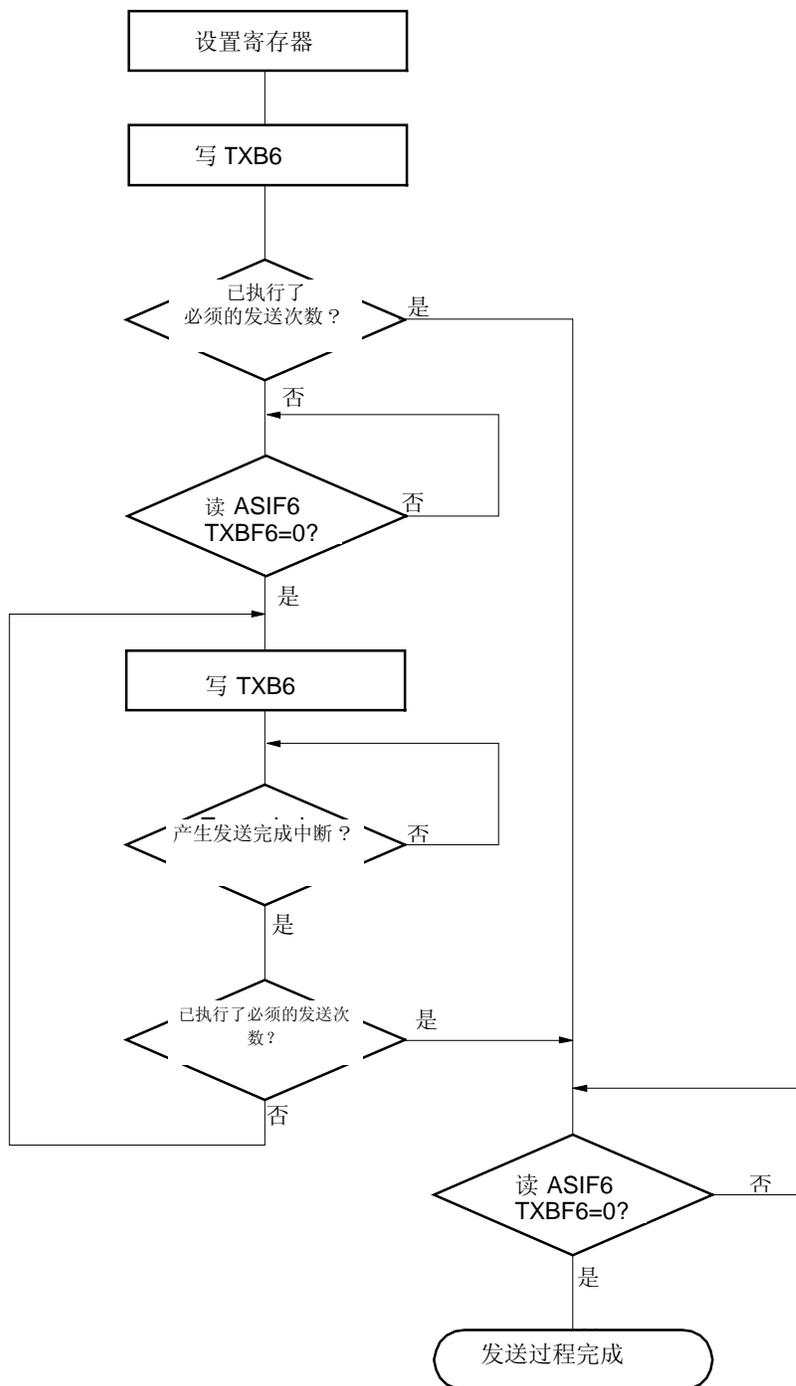
可用 TXSF6 标志位检测通信状态。

TXSF6	发送状态
0	发送操作结束
1	发送操作正在进行中

- 注意事项**
1. 如果要在连续发送结束时初始化发送单元，则在产生发送完成中断后必须检查 TXSF6 标志位，应为“0”。然后执行初始化操作。如果 TXSF6 标志为“1”时执行初始化操作，则无法保证发送数据的正确性。
  2. 在连续发送期间，当发送完一帧数据后执行 INTST6 中断服务之前下一个发送操作可能已经完成。计数测量时，可通过开发一个可以算发送数据的个数的程序或者通过参考 TXSF6 标志来实现检测。

图 14-18 为连续发送操作流程示例。

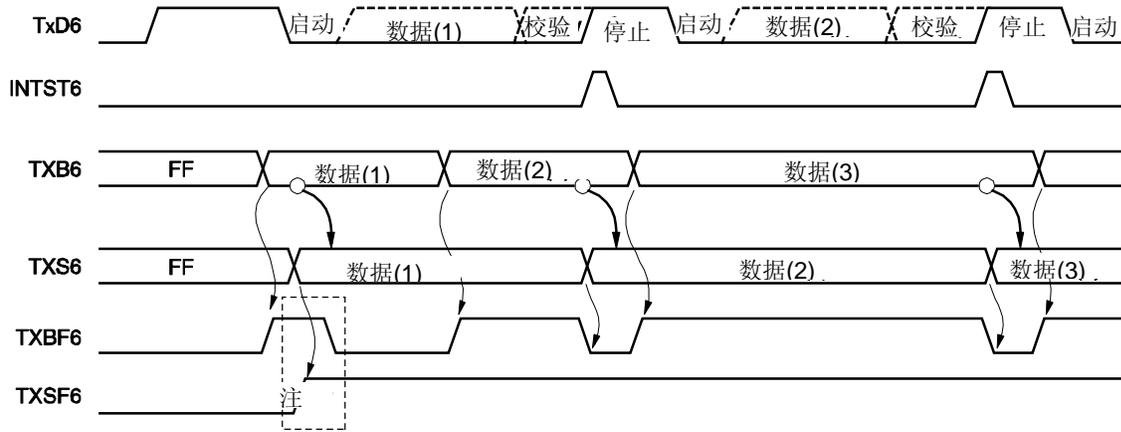
图 14-18. 连续发送操作流程示例



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

图 14-19 显示了启动连续发送操作的时序，而图 14-20 显示了结束连续发送操作的时序。

图 14-19. 启动连续发送操作的时序

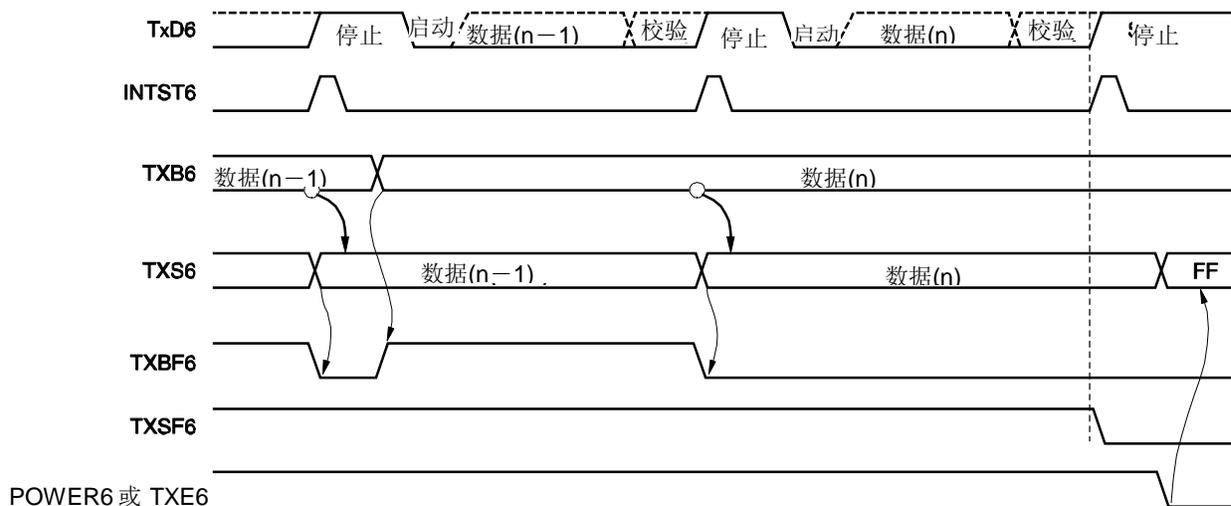


**注** 在读取 ASIF6 时，存在 TXBF6 和 TXSF6 = 1, 1 的情况。因此，仅用 TXBF6 就可以判断是否能进行写操作。

**备注**

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

图 14-20. 结束连续发送操作的时序



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

**(e) 正常接收**

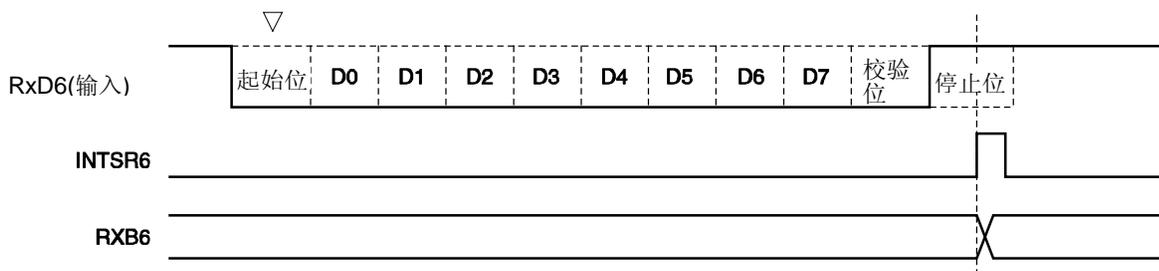
当设置异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)=1, 然后将 ASIM6 的第 5 位(RXE6)置 1 时, 允许接收并对 RxD6 引脚的输入进行采样。

当检测到 RxD6 引脚输入的下降沿时, 波特率发生器的 8 位计数器开始计数。当计数值达到波特率发生器控制寄存器 6(BRGC6)的设置值时, 再次采样 RxD6 的输入信号(如图 14-21 中的▽)。如果此时 RxD6 为低电平, 则认为输入信号为起始位。

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

即使在接收过程中出现校验错(PE6), 接收仍然继续, 直至接收到停止位, 而在接收完成后会产生一个接收错误中断(INTSR6/INTSRE6)。

图 14-21. 接收完成中断请求时序



- 注意事项**
1. 如果出现接收错误, 则先后读取 ASIS6 和 RXB6, 可将出错标志清零。否则, 当接收到下一个数据时会产生溢出错误, 而且接收错误状态保持不变。
  2. 接收始终按“停止位的个数 = 1”的情况执行, 第 2 个停止位被忽略。
  3. 在读取 RXB6 之前, 必须先读取异步串行接口接收错误状态寄存器 6(ASIS6)。

**(f) 接收错误**

在接收期间有三种出错情况: 校验错误、帧错误或溢出错误。作为数据接收的结果, 如果异步串行接口接收错误状态寄存器 6(ASIS6)的出错标志位被置位, 则将会产生一个接收错误中断请求信号 (INTSR6/INTSRE6)。

在执行接收错误中断服务(INTSR6/INTSRE6)时读取 ASIS6 的内容, 可确定接收期间出现的是何种错误(参见图 14-6)。

读取 ASIS6 后, 其内容清零(0)。

表 14-3. 产生接收错误的原因

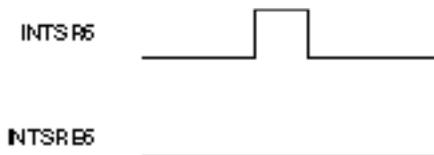
接收错误	产生原因
校验错误	发送数据的校验位与接收到的数据校验位不相等
帧错误	未检测到停止位
溢出错误	在读取接收缓冲寄存器 6(RXB6)的数据前已完成下一个数据的接收操作

通过将异步串行接口操作模式寄存器 6(ASIM6)的第 0 位(ISRM6)清零, 可以把接收错误中断划分为接收完成中断(INTSR6)和错误中断(INTSRE6)两部分。

图 14-22. 接收错误中断

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

(a) 接收期间无错误

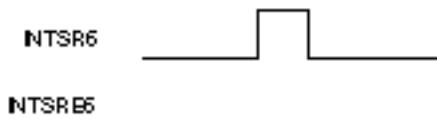


(b) 接收期间产生错误

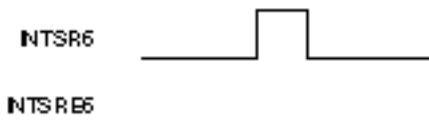


2. 如果 ISRM6=1 (接收错误中断包含在 INTSR6 中)

(a) 接收期间无错误



(b) 接收期间产生错误



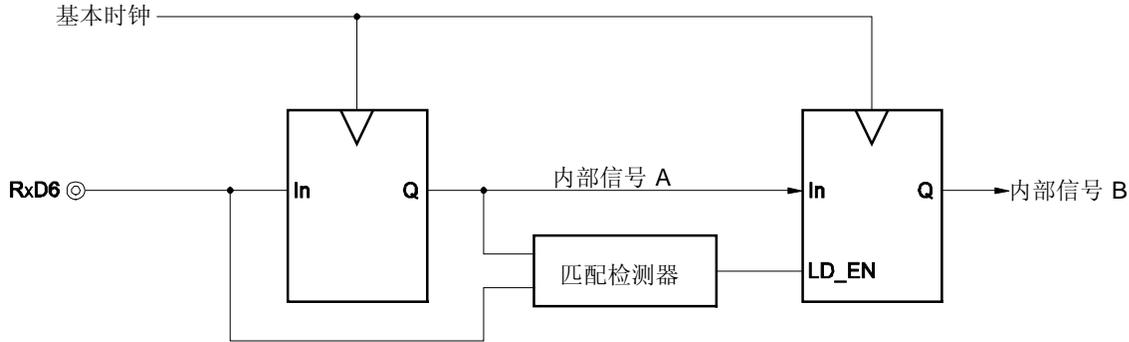
**(g) 接收数据的噪声过滤器**

使用由预分频模块输出的基准时钟采样 RXD6 信号。

如果两次采样值相同，则相等检测器的输出会发生变化，并把采样的数据作为输入数据。

噪声过滤器的电路结构如图 14-23 所示，接收操作的内部过程从外部信号状态开始延迟两个时钟才执行。

图 14-21. 噪声过滤器电路

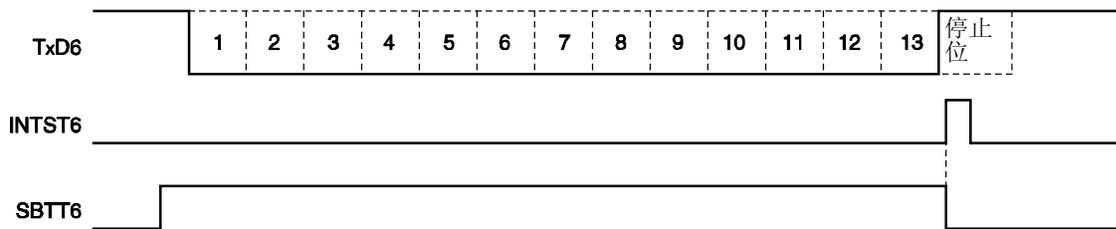
**(h) SBF 发送**

在 LIN 模式中，可使用 SBF (Synchronous Break Field 同步中断区)发送控制功能进行发送。LIN 的发送操作参见图 14-1 LIN 的发送操作。

当异步串行接口模式寄存器 6(ASIM6)的第 7 位(POWER6)被设置为 1 时，TxD6 引脚输出高电平。接下来 ASIM6 的第 6 位(TXE6)被设置为 1 时，进入允许发送状态。且通过设置异步串行接口控制寄存器 6(ASICL6)的第 5 位(SBTT6)为 1，可启动 SBF 发送操作。

开始发送后，第 13 位~第 20 位(由 ASICL6 的第 4 位~第 2 位(SBL62 ~ SBL60)设置)输出低电平。SBF 发送完毕后，产生发送完成中断请求(INTST6)，且 SBTT6 被自动清零。接着，恢复到正常发送模式。直到下一次数据被发送到缓冲寄存器 6(TXB6)或 SBTT6 被设置为 1 时，才停止 SBF 发送。

图 14-24. SBF 发送



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

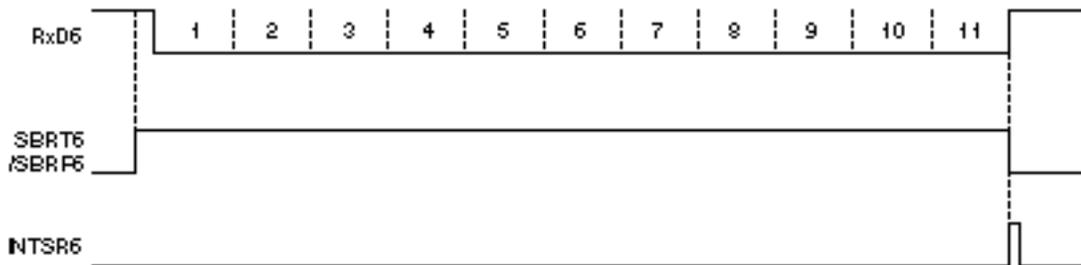
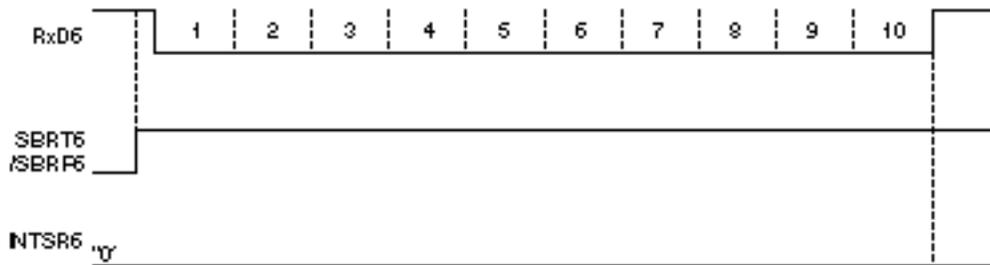
**(i) SBF 接收**

在 LIN 模式下, SBF (Synchronous Break Field 同步中断区)接收控制功能用于接收操作。需要了解 LIN 的接收操作, 可参见图 14-2 LIN 接收操作。

当将异步串行接口模式寄存器 6(ASIM6)的第 7 位(POWER6)被设置为 1, 然后再将 ASIM6 的第 5 位(RXE6)设置为 1 时, 允许接收。当异步串行接口控制寄存器 6(ASICL6)的第 6 位(SBRT6)被设置为 1 时, 允许 SBF 接收。在 SBF 接收允许状态下, 以与正常接收允许状态下同样的方式对 RxD6 引脚进行采样, 并检测起始位。

当检测到起始位时, 启动接收操作, 按照设置的波特率将串行数据依次存入接收移位寄存器 6(RXS6)中。在接收到停止位且 SBF 的宽度大于或等于 11 位时, 与正常接收操作相同, 将产生接收完成中断请求(INTSR6)。此时 SBRF6 和 SBRT6 被自动清零, SBF 接收操作结束。禁止检测 OVE6、PE6 和 FE6(异步串行接口接收错误状态寄存器 6(ASIS6)的第 0 位 ~ 第 2 位)错误, 不进行 UART 通信的错误检测。此外不执行接收移位寄存器 6(RXS6)和接收缓冲寄存器 6(RXB6)之间的数据传送, 并保存复位值 FFH。如果 SBF 的宽度小于或等于 10 位, 则在接收到停止位后不会产生出错中断, 并恢复 SBF 接收模式。此时不对 SBRF6 和 SBRT6 位清零。

图 14-25. SBF 接收

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

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

### 14.4.3 专用波特率发生器

专用波特率发生器由一个源时钟选择器和一个 8 位可编程计数器组成，用于产生 UART6 发送/接收的串行时钟。分别使用不同的 8 位计数器用于发送和接收。

#### (1) 波特率发生器的配置

- 基本时钟

该时钟由时钟选择寄存器 6(CKSR6)的第 3 位 ~ 第 0 位(TPS63 ~ TPS60)选择。当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)=1 时，将该时钟提供给每个模块。该时钟称为基本时钟，它的频率用  $f_{XCLK6}$  表示。当 POWER6 = 0 时，基本时钟恒为低电平。

- 发送计数器

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

当 POWER6 = 1 且 TXE6 = 1 时计数器开始计数。

当第 1 个发送数据写入发送缓冲寄存器 6(TXB6)时，计数器被清零。

如果连续发送数据，则当发送完 1 帧数据时计数器再次被清零。如果后面没有数据要发送，则计数器不被清零，而是继续计数直至 POWER6 或 TXE6 = 0。

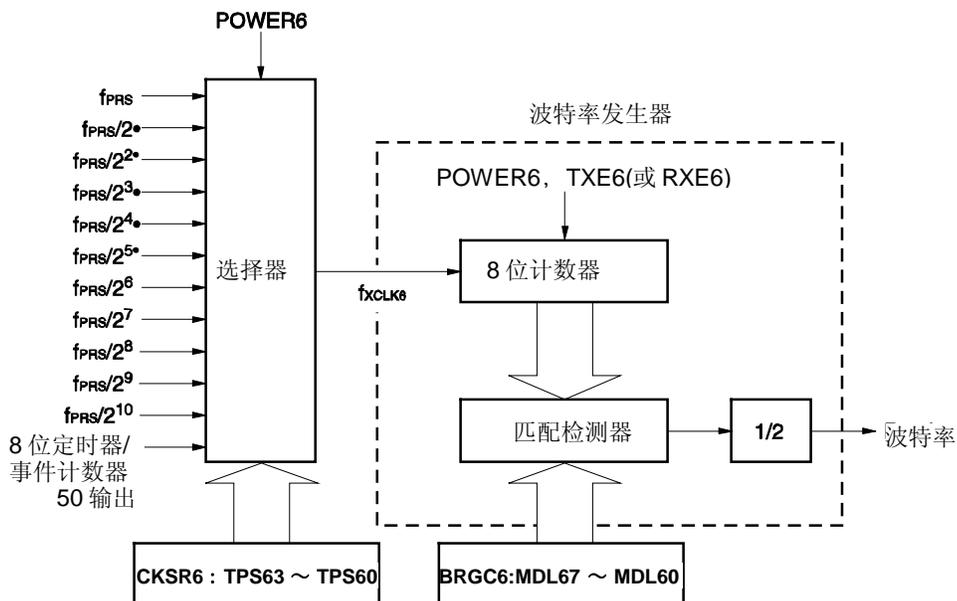
- 接收计数器

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

当检测到起始位时，计数器开始计数。

在收到一帧数据后计数器停止计数，直到下一个起始位被检测到时再重新开始计数。

图 14-26. 波特率发生器的配置



**备注**

- POWER6: 异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位
- TXE6: ASIM6 的第 6 位
- RXE6: ASIM6 的第 5 位
- CKSR6: 时钟选择寄存器 6
- BRGC6: 波特率发生器控制寄存器 6

## (2) 串行时钟的产生

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

使用 CKSR6 的第 3 位 ~ 第 0 位 (TPS63 ~ TPS60)，可以设置输入到 8 位计数器的时钟。使用 BRGC6 的第 7 位 ~ 第 0 位 (MDL67 ~ MDL60) 设置 8 位计数器的分频值 ( $f_{XCLK6}/8 \sim f_{XCLK6}/255$ )。

## 14.4.4 波特率的计算

## (1) 波特率计算公式

波特率可由下列公式计算。

- 波特率 =  $\frac{f_{XCLK6}}{2 \times k}$  [bps]

$f_{XCLK6}$ : 根据 CKSR6 的 TPS63 ~ TPS60 位选择的基本时钟的频率

k: 根据 BRGC6 的 MDL67 ~ MDL60 位设置的值(k = 8, 9, 10, ..., 255)

表 14-4. TPS63 ~ TPS60 的设定值

TPS63	TPS62	TPS61	TPS60	基本时钟 ( $f_{XCLK6}$ ) 选择 <sup>注1</sup>				
				$f_{PRS} =$ 2 MHz	$f_{PRS} =$ 5 MHz	$f_{PRS} =$ 8 MHz	$f_{PRS} =$ 10 MHz	
0	0	0	0	$f_{PRS}$ <sup>Note 2</sup>	2 MHz	5 MHz	8 MHz	10 MHz
0	0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	4 MHz	5 MHz
0	0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2 MHz	2.5 MHz
0	0	1	1	$f_{PRS}/2^3$	250 kHz	625 kHz	1 MHz	1.25 MHz
0	1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	500 kHz	625 kHz
0	1	0	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	250 kHz	312.5 kHz
0	1	1	0	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	125 kHz	156.25 kHz
0	1	1	1	$f_{PRS}/2^7$	15.625 kHz	39.06 kHz	62.5 kHz	78.13 kHz
1	0	0	0	$f_{PRS}/2^8$	7.813 kHz	19.53 kHz	31.25 kHz	39.06 kHz
1	0	0	1	$f_{PRS}/2^9$	3.906 kHz	9.77 kHz	15.625 kHz	19.53 kHz
1	0	1	0	$f_{PRS}/2^{10}$	1.953 kHz	4.88 kHz	7.813 kHz	9.77 kHz
1	0	1	1	TM50 输出 <sup>注3</sup>				
其它				禁止设置				

- 注
1. 当外围硬件时钟 ( $f_{PRS}$ ) 运行于高速系统时钟( $f_{XH}$ ) ( $XSEL = 1$ ),  $f_{PRS}$  操作频率依据供电电压的不同而不同。
    - $V_{DD} = 2.7 \sim 5.5 V$ :  $f_{PRS} \leq 10 MHz$
    - $V_{DD} = 1.8 \sim 2.7 V$ :  $f_{PRS} \leq 5 MHz$
  2. 如果外围硬件时钟 ( $f_{PRS}$ ) 运行于内部高速振荡时钟( $f_{RH}$ ) ( $XSEL = 0$ ), 当  $1.8 V \leq V_{DD} < 2.7 V$ ,  $TPS63 = TPS62 = TPS61 = TPS60 = 0$  (基时钟:  $f_{PRS}$ ) 这种设置禁止。
  3. 选择 TM50 输出作为基本时钟时,先启动 8 位定时器/事件计数器 50, 然后允许定时器 F/F 反转操作 ( $TMC501 = 1$ )。

**(2) 波特率误差**

通过下列公式计算波特率误差。

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

- 注意事项**
1. 发送期间必须保持波特率误差在接收端允许的误差范围内。
  2. 接收期间，波特率误差必须满足“(4)接收期间允许的波特率范围”中所描述的范围。

**举例:**      基本时钟频率 = 10 MHz = 10, 000, 000 Hz  
                  BRGC6 的 MDL67 ~ MDL60 位 = 00100001B (k = 33)  
                  目标波特率 = 153600 bps

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

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

## (3) 波特率设置举例

表 14-5. 波特率发生器的设置数据

波特率 [bps]	f <sub>PRS</sub> = 2.0 MHz				f <sub>PRS</sub> = 5.0 MHz				f <sub>PRS</sub> = 10.0 MHz			
	TPS63- TPS60	k	计算数值	ERR [%]	TPS63- TPS60	k	计算数值	ERR [%]	TPS63- TPS60	k	计算数值	ERR [%]
300	8H	13	301	0.16	7H	65	301	0.16	8H	65	301	0.16
600	7H	13	601	0.16	6H	65	601	0.16	7H	65	601	0.16
1200	6H	13	1202	0.16	5H	65	1202	0.16	6H	65	1202	0.16
2400	5H	13	2404	0.16	4H	65	2404	0.16	5H	65	2404	0.16
4800	4H	13	4808	0.16	3H	65	4808	0.16	4H	65	4808	0.16
9600	3H	13	9615	0.16	2H	65	9615	0.16	3H	65	9615	0.16
19200	2H	13	19231	0.16	1H	65	19231	0.16	2H	65	19231	0.16
24000	1H	21	23810	-0.79	3H	13	24038	0.16	4H	13	24038	0.16
31250	1H	16	31250	0	4H	5	31250	0	5H	5	31250	0
38400	1H	13	38462	0.16	0H	65	38462	0.16	1H	65	38462	0.16
48000	0H	21	47619	-0.79	2H	13	48077	0.16	3H	13	48077	0.16
76800	0H	13	76923	0.16	0H	33	75758	-1.36	0H	65	76923	0.16
115200	0H	9	111111	-3.55	1H	11	113636	-1.36	0H	43	116279	0.94
153600	-	-	-	-	1H	8	156250	1.73	0H	33	151515	-1.36
312500	-	-	-	-	0H	8	312500	0	1H	8	312500	0
625000	-	-	-	-	0H	4	625000	0	1H	4	625000	0

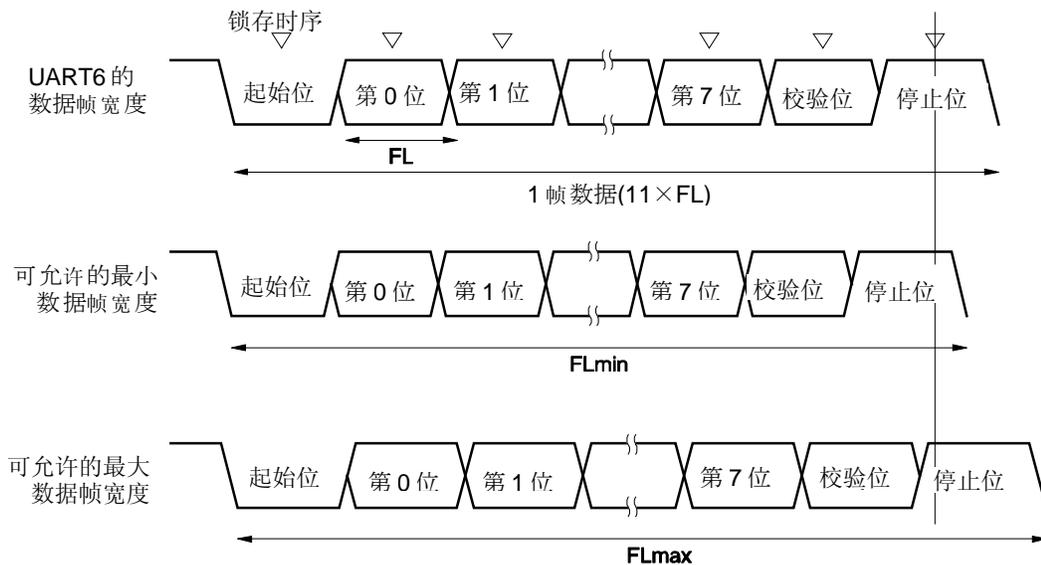
**备注**      TPS63 ~ TPS60: 时钟选择寄存器 6 (CKSR6) 的第 3 位 ~ 第 0 位 (用于设置基本时钟(f<sub>XCLK6</sub>))  
k:              使用波特率发生器控制寄存器 6 (BRGC6) 的 MDL67 ~ MDL60 位设置的值  
                  (k = 8, 9, 10, ..., 255)  
f<sub>PRS</sub>:         外围硬件时钟频率  
ERR:           波特率误差

**(4) 接收期间允许的波特率范围**

接收期间来自发送端的波特率可允许的误差如下所示。

**注意事项** 必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。

图 14-27. 接收期间允许的波特率范围



如图 14-27 所示，当检测到起始位后，接收数据的锁存时序由波特率发生器控制寄存器 6(BRGC6)设置的计数器确定。如果数据的最后一位(停止位)满足该锁存时序，则数据可被正确接收。

假定接收数据为 11 位，各项理论值计算如下。

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

Brate: UART6 的波特率  
 k: BRGC6 的设置值  
 FL: 1 位数据宽度  
 锁存时序的极限: 2 个时钟

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

因此在发送端设定的可接收的最大波特率如下所示。

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

同样，可计算允许的最大数据帧宽度。

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

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

因此在发送端设定的可接收的最小波特率如下所示。

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

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

表 14-6. 允许的最大/最小波特率误差

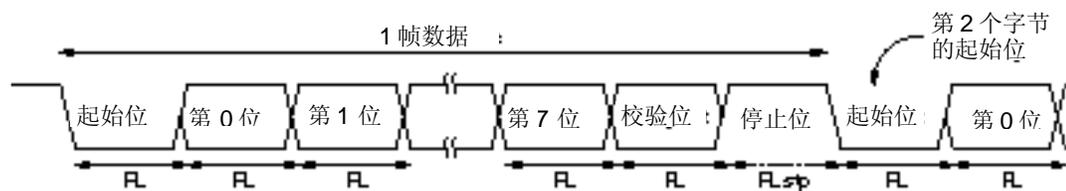
分频比(k)	允许的最大波特率误差	允许的最小波特率误差
4	+2.33%	-2.44%
8	+3.53%	-3.61%
20	+4.26%	-4.31%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.73%

- 备注**
1. 接收时允许的误差与每帧的位数、输入时钟频率和分频比(k)有关。输入时钟频率和分频比(k)越高，允许的误差就越大。
  2. k: BRGC6 设置的值。

**(5) 连续发送期间数据帧的宽度**

当连续发送数据时，从停止位到下一个起始位的数据帧宽度比正常值延长了两个基本时钟。不过，由于在检测到起始位时接收端的时序被初始化，因此通信结果不会受到影响。

图 14-28. 连续发送期间数据帧的宽度



当 1 位数据宽度为  $FL$ 、停止位宽度为  $FL_{stp}$ ，基本时钟频率为  $f_{XCLK6}$  时，有以下公式成立。

$$FL_{stp} = FL + 2/f_{XCLK6}$$

因此，连续发送期间的数据帧宽度为：

$$\text{数据帧宽度} = 11 \times FL + 2/f_{XCLK6}$$

## 第十五章 LCD 控制器/驱动器

### 15.1 LCD 控制器/驱动器的功能

78K0/LC3 的 LCD 控制器/驱动器的功能如下。

- (1) LCD 驱动器的参考电压发生器可转换内部电压泵、外部电阻分压和内部电阻分压。
- (2) 基于自动读取存储器显示数据，自动输出 segment 和 common 信号。
- (3) 5 种不同的显示模式：
  - 静态
  - 1/2 占空比(1/2 偏压)
  - 1/3 占空比(1/2 偏压)
  - 1/3 占空比(1/3 偏压)
  - 1/4 占空比(1/3 偏压)
  - 1/8 占空比(1/4 偏压)
- (4) 每种显示模式都有 4 种不同的频率。
- (5) Segment 信号输出: 22<sup>#</sup> (SEG0 ~ SEG21), Common 信号输出: 8<sup>#</sup> (COM0 ~ COM7)

**注** 有 4 个 segment 信号输出(SEG0 ~ SEG3) 和 4 个 common 信号输出 (COM4 ~ COM7) 位复用引脚。COM4 ~ COM7 只有在 LCD 显示模式寄存器选定 8 分时模式时使用。

表 15-1 列出了每种显示模式的最大显示像素。

表 15-1. 最大像素列表

LCD 驱动器的参考电压发生器	偏压模式	分时	使用的 Common 信号	Segments 数	最大像素
<ul style="list-style-type: none"> <li>•外部电阻分压</li> <li>•内部电阻分压</li> </ul>	-	静态	COM0 (COM1 ~ COM3)	22	22 (22 segment 信号, 1 路 common 信号) <sup># 1</sup>
	1/2	2	COM0, COM1		44 (22 segment 信号, 2 路 common 信号) <sup># 2</sup>
		3	COM0 ~ COM2		66 (22 segment 信号, 3 路 common 信号) <sup># 3</sup>
	1/3	3	COM0 ~ COM2		88 (22 segment 信号, 4 路 common 信号) <sup># 4</sup>
		4	COM0 ~ COM3		
	1/4	8	COM0 ~ COM7	18	144 (18 segment 信号, 8 路 common 信号) <sup># 5</sup>

- 注
1. 2-数字 LCD 面板, 每个 digit 有 8-segment 构造。
  2. 5-数字 LCD 面板, 每个 digit 有 4-segment 构造。
  3. 7-数字 LCD 面板, 每个 digit 有 3-segment 构造。
  4. 11-数字 LCD 面板, 每个 digit 有 2-segment 构造。
  5. 18-数字 LCD 面板, 每个 digit 有 1-segment 构造。

## 15.2 LCD 控制器/驱动器的配置

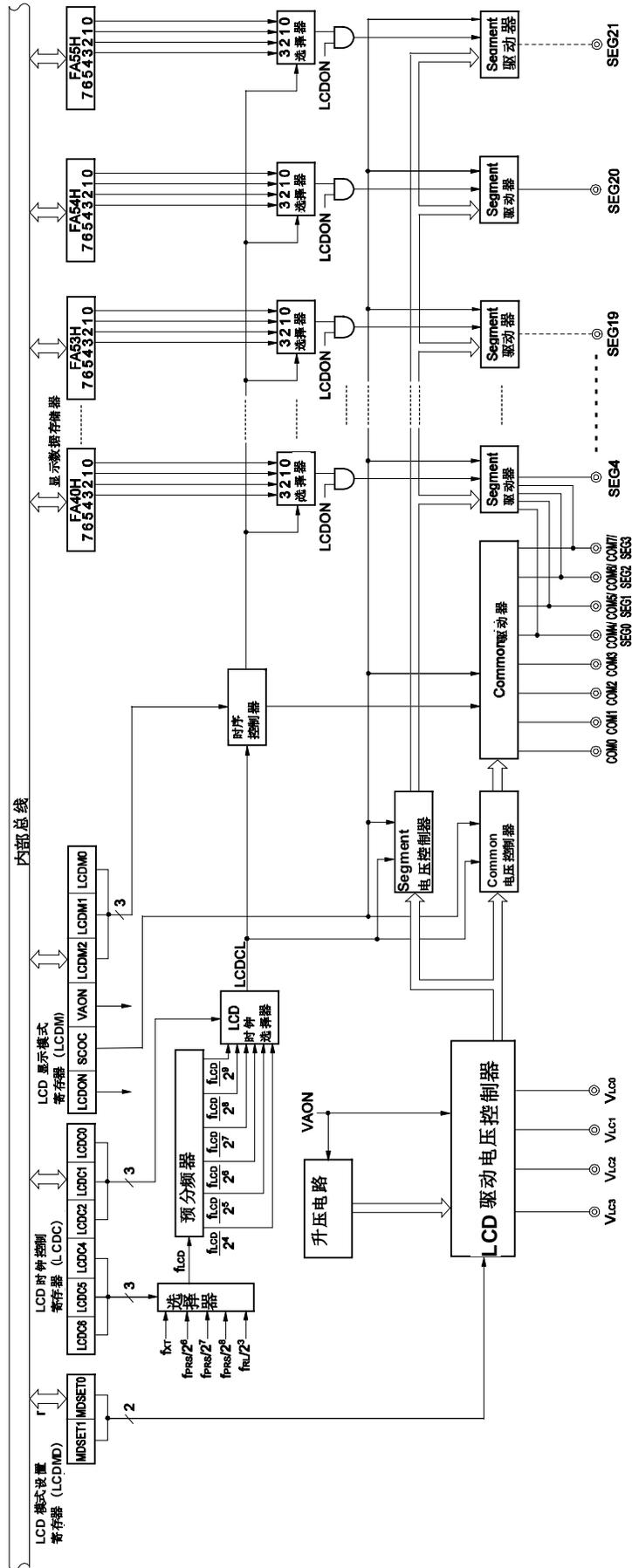
LCD 控制器/驱动器包括以下硬件。

表 15-2. LCD 控制器/驱动器的配置

项目	配置
显示输出	22 segment 信号 <sup>※</sup> (SEG0 ~ SEG21), 8 common 信号 <sup>※</sup> (COM0 ~ COM7)
控制寄存器	LCD 模式寄存器 (LCDMD) LCD 显示模式寄存器 (LCDM) LCD 时钟控制寄存器 (LCDC0) 端口功能寄存器 2 (PF2) 端口功能寄存器 ALL (PFALL)

注 有 4 个 segment 信号输出 (SEG0 ~ SEG3) 和 4 个 common 信号输出 (COM4 ~ COM7) 位复用引脚。  
COM4 ~ COM7 只有在 LCD 显示模式寄存器选定 8 分时模式时使用。

图 16-1.LCD 控制器/驱动器框图



### 15.3 控制 LCD 控制器/驱动器的寄存器

LCD 控制器/驱动器使用以下 5 个寄存器进行控制。

- LCD 模式寄存器 (LCDMD)
- LCD 显示模式寄存器 (LCDM)
- LCD 时钟控制寄存器 (LCDC0)
- 端口功能寄存器 2 (PF2)
- 端口功能寄存器 ALL (PFALL)

#### (1) LCD 模式寄存器(LCDMD)

LCDMD 设置 LCD 驱动电压发生器。

可由 1 位或 8 位存储器操作指令设置 LCDMD。

复位信号将 LCDMD 设置为 00H。

图 15-2. LCD 模式寄存器的格式

地址: FFB0H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
LCDMD	0	0	MDSET1	MDSET0	0	0	0	0

MDSET1	MDSET0	LCD驱动电压发生器选择
0	0	外部电阻分压方式
0	1	内部电阻分压方式(没有等差转换) (当 $V_{LCD} = V_{DD}$ 时使用)
1	1	内部电阻分压方式(等差转换) (当 $V_{LCD} = 3/5V_{DD}$ 时使用)
其它情况		禁止设置

注意事项 位 0~3, 6 和 7 必须为 0。

**(2) LCD 显示模式寄存器 (LCDM)**

LCDM 可以选择液晶是否显示。还可以选择 segment 引脚/common 引脚输出、门升压电路操作和显示模式。可由 1 位或 8 位存储器操作指令设置 LCDM。复位信号将 LDMD 设置为 00H。

图 15-3. LCD 显示模式寄存器格式

地址: FFB1H 复位后: 00H R/W

符号	<7>	<6>	5	<4>	3	2	1	0
LCDM	LCDON	SCOC	0	VAON	0	LCDM2	LCDM1	LCDM0

LCDON	LCD 显示允许/禁止
0	显示关 (取消全部segment 输出)
1	显示

SCOC	Segment 引脚/common 引脚输出控制 <sup>注1</sup>
0	segment/common 引脚输出低电平
1	segment 引脚输出电压取消和common 引脚输出LCD波形

VAON	门升压电路操作 <sup>注 1, 2</sup>
0	无
1	有

LCDM2	LCDM1	LCDM0	LCD 控制器/驱动器显示模式选择	
			电阻分压模式	
			分时数	偏压模式
1	1	1	8	1/4 <sup>注 3</sup>
0	0	0	4	1/3
0	0	1	3	1/3
0	1	0	2	1/2
0	1	1	3	1/2
1	0	0	静态	
其它情况			禁止设置	

(注和注意事项在下一页显示)

- 注
1. LCD 显示不运行或者没必要时, 将 SCOC 和 VAON 设置为 0, 可以降低功耗。
  2. 此位用来控制 LCD 控制器/驱动器的内部门信号的升压。  
如果设置为"内部门升压", LCD 驱动性能得到增强。  
基于下列条件设置 VAON。  
<设置为静态显示模式时>
    - 当  $2.0\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$ : VAON = 0
    - 当  $1.8\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 3.6\text{ V}$ : VAON = 1
 <设置为 1/3 偏压模式时>
    - 当  $2.5\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$ : VAON = 0
    - 当  $1.8\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 3.6\text{ V}$ : VAON = 1
 <设置为 1/2 偏压模式时>
    - 当  $2.7\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$ : VAON = 0
    - 当  $1.8\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 3.6\text{ V}$ : VAON = 1
 <设置为 1/4 偏压模式时>
    - 当  $4.5\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$ : VAON = 0
  3. 当 P40/KR0/VLC3 引脚设置为 1/4 偏压模式时, 它用作  $V_{\text{LC3}}$ 。当引脚设置为另一偏压模式时, 它用作端口功能 (P40) 或者按键中断功能 (KR0)。  
当设置为当引脚为 1/4 偏压模式时, 这样使用引脚:  $4.5\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$ 。

**注意事项** 位 3 和 5 必须设置为 0。

**(3) LCD 时钟控制寄存器 (LCDC)**

LCDC 指定 LCD 时钟源和 LCD 时钟。

依据 LCD 时钟和分时来决定帧频率。

可由 1 位或 8 位存储器操作指令设置 LCDC。

复位信号将 LCDC 设置为 00H。

图 15-4. LCD 时钟控制寄存器的格式

地址: FFB2H 复位后: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDC0	0	LCDC6	LCDC5	LCDC4	0	LCDC2	LCDC1	LCDC0

LCDC6	LCDC5	LCDC4	LCD 源时钟 (f <sub>lcd</sub> ) 选择
0	0	0	f <sub>XT</sub> (32.768 kHz)
0	0	1	f <sub>PRS</sub> /2 <sup>6</sup>
0	1	0	f <sub>PRS</sub> /2 <sup>7</sup>
0	1	1	f <sub>PRS</sub> /2 <sup>8</sup>
1	0	0	f <sub>RL</sub> /2 <sup>3</sup>
其它情况			禁止设置

LCDC2	LCDC1	LCDC0	LCD 时钟 (LCDCL) 选择
0	0	0	f <sub>lcd</sub> /2 <sup>4</sup>
0	0	1	f <sub>lcd</sub> /2 <sup>5</sup>
0	1	0	f <sub>lcd</sub> /2 <sup>6</sup>
0	1	1	f <sub>lcd</sub> /2 <sup>7</sup>
1	0	0	f <sub>lcd</sub> /2 <sup>8</sup>
1	0	1	f <sub>lcd</sub> /2 <sup>9</sup>
其它情况			禁止设置

**注意事项** 位 3 和 7 必须设置为 0。

- 备注**
1. f<sub>XT</sub>: XT1 时钟振荡频率
  2. f<sub>PRS</sub>: 外围硬件时钟频率
  3. f<sub>RL</sub>: 内部低速振荡时钟频率

**(4) 端口功能寄存器 2 (PF2)**

此寄存器设置引脚 P20 ~ P25 用作端口引脚 (不用作 segment 输出引脚) 或者 segment 输出引脚。  
 可由 1 位或 8 位存储器操作指令设置 PF2。  
 复位信号将 PF2 设置为 00H。

图 15-5. 端口功能寄存器 2 的格式

地址: FFB5H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PF2	0	0	PF25	PF24	PF23	PF22	PF21	PF20

PF2n	端口/segment 输出规定
0	用作端口 (不用作segment 输出)
1	用作segment 输出

备注 n = 0 ~ 5

**(5) 端口功能寄存器 ALL (PFALL)**

此寄存器设置引脚 P10, P11, P14 或 P15 用作端口引脚 (不用作 segment 输出引脚) 或者 segment 输出引脚。  
 可由 1 位或 8 位存储器操作指令设置 PFALL。  
 复位信号将 PFALL 设置为 00H。

图 15-6. 端口功能寄存器 ALL 的格式

地址: FFB6H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PFALL	0	PF15ALL	PF14ALL	0	PF11ALL	PF10ALL	0	0

PFnALL	端口/segment 输出规定
0	用作端口 (不用作segment 输出)
1	用作segment 输出

备注 n = 10, 11, 14 或 15

## 15.4 设置 LCD 控制器/驱动器

设置 LCD 控制器/驱动器按以下步骤。

- <1> 设置 (VAON = 1) 内部门升压(LCD 显示模式寄存器(LCDM)的位 4)
- <2> 通过 MDSET0 ~ MDSET1 (LCD 模式寄存器 (LCDMD) 的位 4 和 5) (MDSET0 = 0: 外部电阻分压模式, MDSET0 = 1: 内部电阻分压模式) 设置电阻分压模式。
- <3> 设置用作 segment 输出的引脚为端口功能寄存器 (PF2m, PFnALL)。
- <4> 设置 LCD 显示 RAM 中的 LCD 显示数据初值。
- <5> 通过 LCDM0 ~ LCDM2 (LCD 显示模式寄存器 (LCDM) 的位 0 ~ 2) 设置分分数。
- <6> 通过 LCD 时钟控制寄存器 0 (LCDC0)设置 LCD 源时钟 和 LCD 时钟。
- <7> 设置 (SCOC = 1) SCOC (LCD 显示模式寄存器 (LCDM)的位 6)。  
没有选定的波形从所用的 segment 和 common 引脚输出, 并进入不显示模式。
- <8> 通过设置 (LCDON = 1) LCDON (LCDM 的位 7)为 1, 根据每一个数据存储器的值开始输出。

以后的流程同上, 在数据存储器设置显示数据。

**注** 基于下列条件设置 VAON。

<设置为静态显示模式时>

- 当  $2.0\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$ : VAON = 0
- 当  $1.8\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 3.6\text{ V}$ : VAON = 1

<设置为 1/3 偏压模式时>

- 当  $2.5\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$ : VAON = 0
- 当  $1.8\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 3.6\text{ V}$ : VAON = 1

<设置为 1/2 偏压模式时>

- 当  $2.7\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$ : VAON = 0
- 当  $1.8\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 3.6\text{ V}$ : VAON = 1

<设置为 1/4 偏压模式时>

- 当  $4.5\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$ : VAON = 0

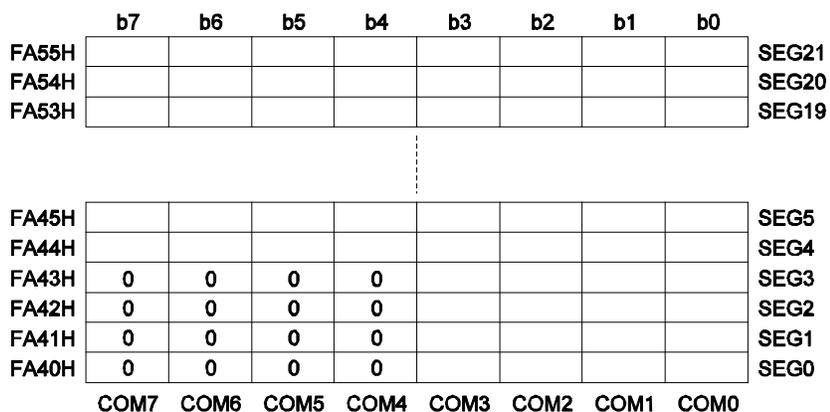
**备注** m = 0 ~ 5, n = 10, 11, 14 或 15

### 15.5 LCD 显示数据存储器

The LCD 显示数据存储器映射在地址 FA40H ~ FA55H。使用 LCD 控制器/驱动器，LCD 显示数据存储器中的数据显示在 LCD 面板上。

图 15-7 显示了 LCD 显示数据存储器 and segment/common 输出的关系。  
不用作显示的区域可以作为普通 RAM 区。

图 15-7. LCD 显示数据存储器 and segment/common 输出的关系



注意事项 FA40H ~ FA43H 的高 4 位为空。一定要将那些位设置为 0。

## 15.6 Common 和 Segment 信号

当相应的 common 和 segment 信号产生高于指定电压(LCD 驱动电压, VLCD)的位差, 开启 LCD 面板上的每个像素。当产生低于 VLCD 位差时关闭像素。

应用 DC 电压到 LCD 面板驱动 common 和 segment 信号, 可能引发坏点。为了避免这个问题, 使用 AC 电压驱动 LCD 面板。

### (1) Common 信号

依照表 18-4 分时列表列出的分时顺序选择 common 信号。静态显示模式时, COM0 ~ COM3 输出相同的信号。

在 2 分时模式, COM2 和 COM3 引脚悬空。3 分时模式, COM3 引脚悬空。

8 分时模式除外, 将 COM4 ~ COM7 引脚悬空或者设置为 segment 引脚。

表 15-3. COM 信号

COM信号 分时数	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
静态显示模式	←	←	←	←	注	注	注	注
2 分时模式	←	←	悬空	悬空	注	注	注	注
3 分时模式	←	←	←	悬空	注	注	注	注
4 分时模式	←	←	←	←	注	注	注	注
8 分时模式	←	←	←	←	←	←	←	←

注 将这些引脚悬空或者 segment 引脚。

### (2) Segment 信号

segment 信号对应 LCD 显示数据存储器的 22 字节 (FA40H ~ FA55H)。每一个字节的位 0, 1, 2, 和 3 分别与 COM0, COM1, COM2, 和 COM3 相对应同步读取。如果为 1, 更改为选择电压;如果为 0, 则取消选定电压。转换结果由 segment(S0 ~ S21) 引脚输出。

检测以上结合 front-surface 电极 (对应于 segment 信号) 和 rear-surface 电极(对应于 common 信号)的信号, 形成 LCD 显示数据存储器的显示模式, 并且按照一对一的方式写入欲显示模式的位数据。

LCD 显示数据存储器的位 1~位 3、位 2 和位 3、位 3 分别不可用于 LCD 静态显示、2 分时和 3 分时模式。这些位用于非显示功能。

高 4 位固定为 0。

**(3) 输出 common 和 segment 信号波形**

表 15-4 列出了 common 和 segment 信号输出电压。

当 common 和 segment 信号都为选定电压，将获得显示器开显示电压 $\pm V_{LCD}$ 。其他情况为关显示电压。

**表 15-5. LCD 驱动电压**
**(a) 静态显示模式**

Segment 信号 \ Common 信号		选定信号电平	取消选定信号电平
		$V_{LSS}/V_{LCO}$	$V_{LCO}/V_{SS}$
$V_{LCO}/V_{SS}$		$-V_{LCD}/+V_{LCD}$	0 V/0 V

**(b) 1/2 偏压模式**

Segment 信号 \ Common 信号		选定信号电平	取消选定信号电平
		$V_{LSS}/V_{LCO}$	$V_{LCO}/V_{SS}$
选定信号电平	$V_{LCO}/V_{SS}$	$-V_{LCD}/+V_{LCD}$	0 V/0 V
取消信号电平	$V_{LC1} = V_{LC2}$	$-\frac{1}{2}V_{LCD}/+\frac{1}{2}V_{LCD}$	$+\frac{1}{2}V_{LCD}/-\frac{1}{2}V_{LCD}$

**(c) 1/3 偏压模式**

Segment 信号 \ Common 信号		选定信号电平	取消选定信号电平
		$V_{LSS}/V_{LCO}$	$V_{LC1}/V_{LC2}$
选定信号电平	$V_{LCO}/V_{SS}$	$-V_{LCD}/+V_{LCD}$	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$
取消信号电平	$V_{LC2}/V_{LC1}$	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$	$+\frac{1}{3}V_{LCD}/-\frac{1}{3}V_{LCD}$

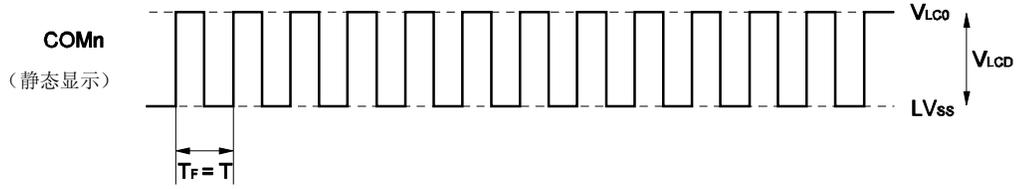
**(d) 1/4 偏压模式**

Segment 信号 \ Common 信号		选定信号电平	取消选定信号电平
		$V_{LSS}/V_{LCO}$	$V_{LC1}/V_{LC2}$
选定信号电平	$V_{LSS}/V_{LCO}$	$+V_{LCD}/-V_{LCD}$	$+\frac{1}{2}V_{LCD}/-\frac{1}{2}V_{LCD}$
取消信号电平	$V_{LC1}/V_{LC3}$	$+\frac{1}{4}V_{LCD}/-\frac{1}{4}V_{LCD}$	$-\frac{1}{4}V_{LCD}/+\frac{1}{4}V_{LCD}$

图 15-8 显示了 common 信号波形，和图 15-9 显示了 common 和 segment 信号的电压和相位。

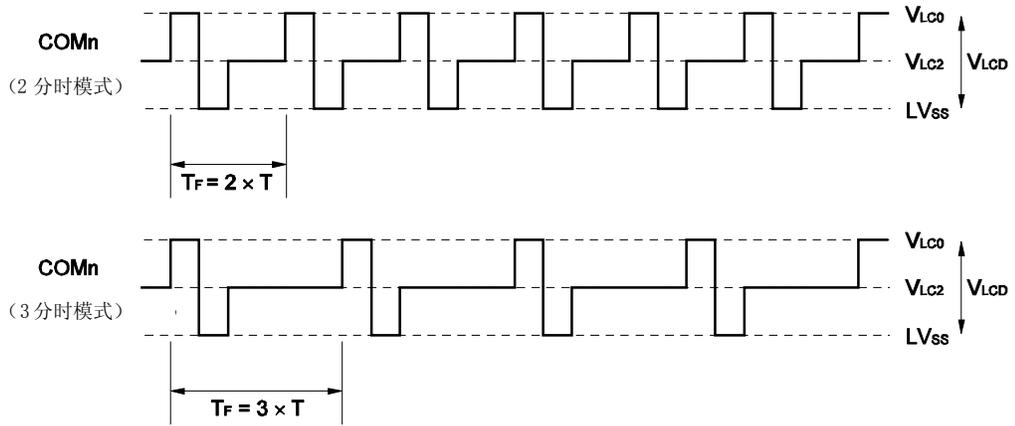
图 15-8. Common 信号波形

(a) 静态显示模式



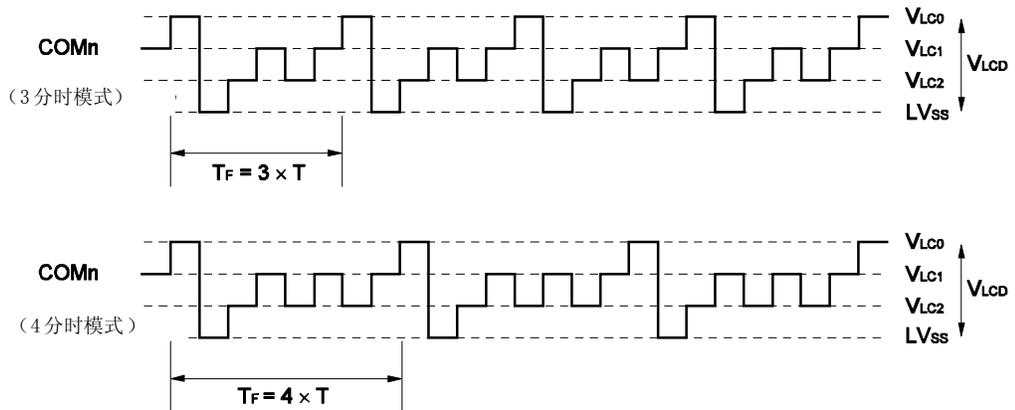
T: 1 个 LCD 时钟周期      T<sub>F</sub>: 帧频率

(b) 1/2 偏压模式



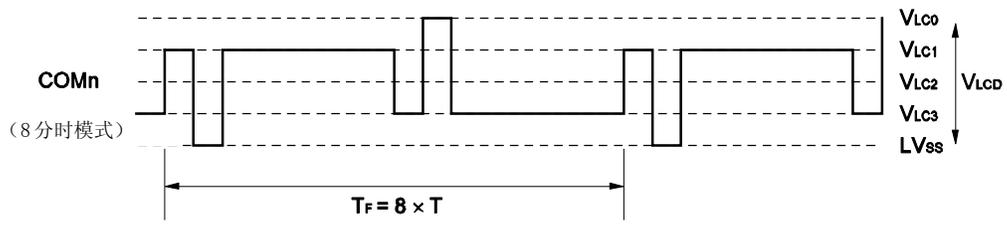
T: 1 个 LCD 时钟周期      T<sub>F</sub>: 帧频率

(c) 1/3 偏压模式



T: 1 个 LCD 时钟周期      T<sub>F</sub>: 帧频率

(d) 1/4 偏压模式

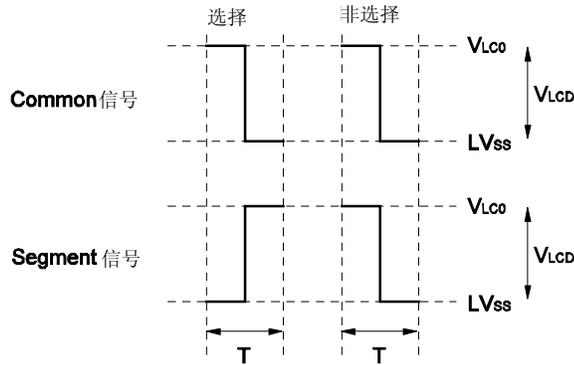


T: 1 个 LCD 时钟周期

$T_F$ : 帧频率

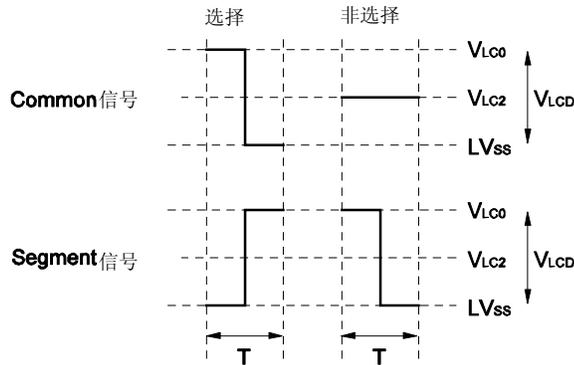
图 15-9. Common 和 Segment 信号的电压和相位

(a) 静态显示模式



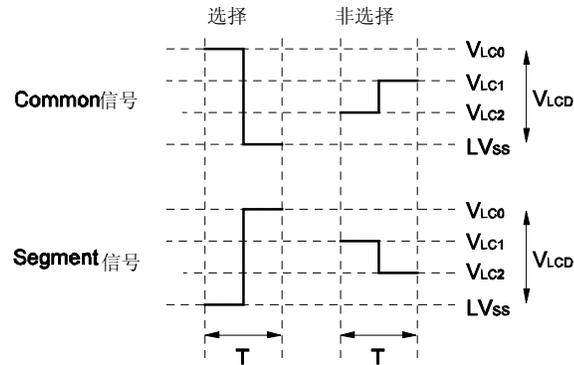
T: 1 个 LCD 时钟周期

(b) 1/2 偏压模式



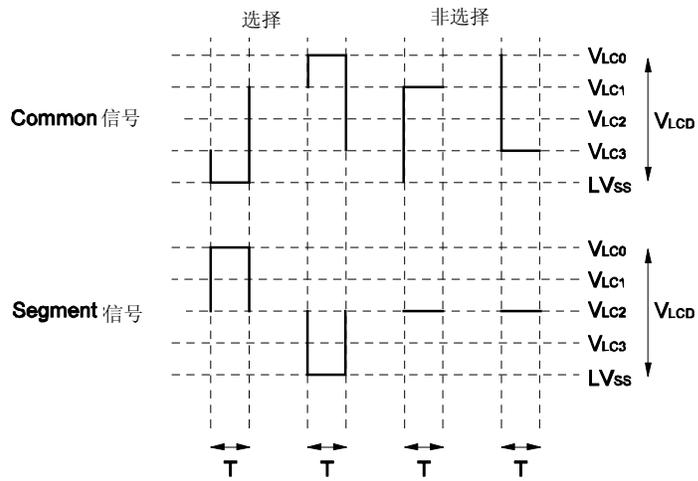
T: 1 个 LCD 时钟周期

(c) 1/3 偏压模式



T: 1 个 LCD 时钟周期

(d) 1/4 偏压模式



T: 1 个 LCD 时钟周期

## 15.7 显示模式

### 15.7.1 静态显示举例

图 15-11 显示的是具有图 15-10 所示的显示模式的 3-位 LCD 面板是怎样连接 78K0/LC3 芯片的 segment 信号 (SEG0 ~ SEG21) 和 common 信号 (COM0)。此例显示的是在 LCD 面板上显示数字“2.3”。显示数据存储区(FA40H to FA55H)的内容与此相对应。

下面描述在第二位数字上显示数字"2." (  )。为了在 LCD 面板上显示 "2."，必须在 common 信号 COM0 时序中根据表 15-5 将电压加在 SEG8~SEG15 上；请看 图 15-10 segment 信号 和 LCD segments 之间的关系。

表 15-5. 选定和取消电压(COM0)

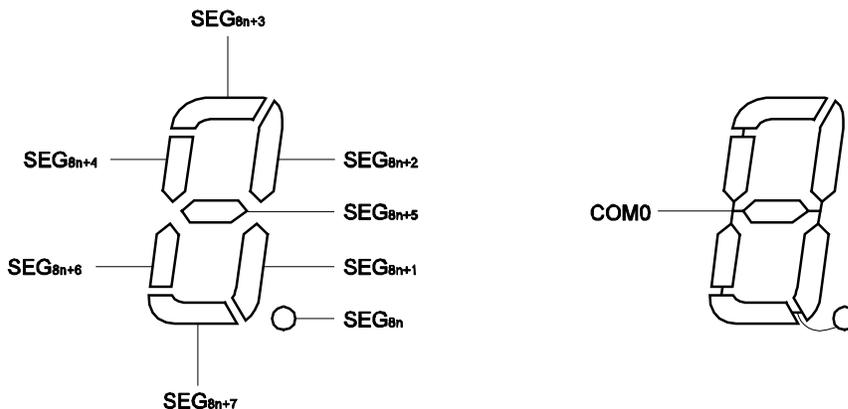
Segment Common	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15
COM0	选定	取消	选定	选定	取消	选定	选定	选定

根据表 15-5，显示数据存储器区的第 0 位 (FA48H ~ FA4FH)依次必须绝对是 10110111。

图 15-12 显示了 S11 和 S12, 和 COM0 的 LCD 驱动波形。当选定电压应用到 COM0 时序的 S11 时，LCD segment 产生相应的一个交替长方形波形，+V<sub>LCD</sub>/-V<sub>LCD</sub>。

COM1 ~ COM3 和 COM0 的波形相同。所以，COM0 ~ COM3 可以连接在一起增加驱动能力。

图 15-10. 静态 LCD 显示图形和电极连接



备注 n = 0, 2

图 15-11. 静态 LCD 面板连接举例

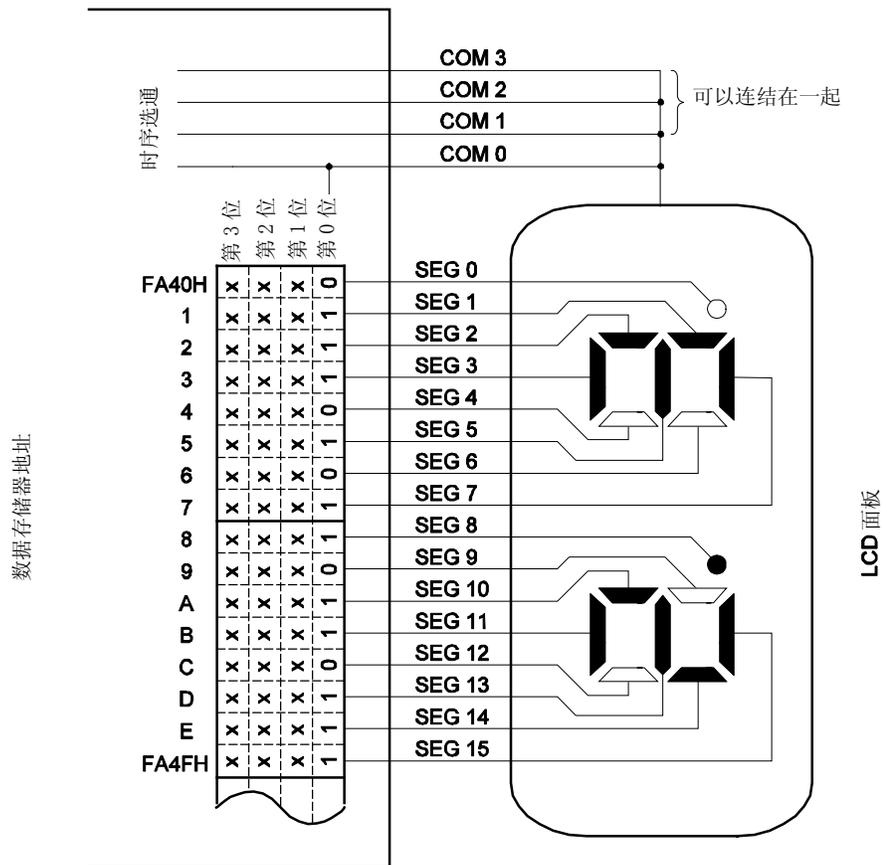
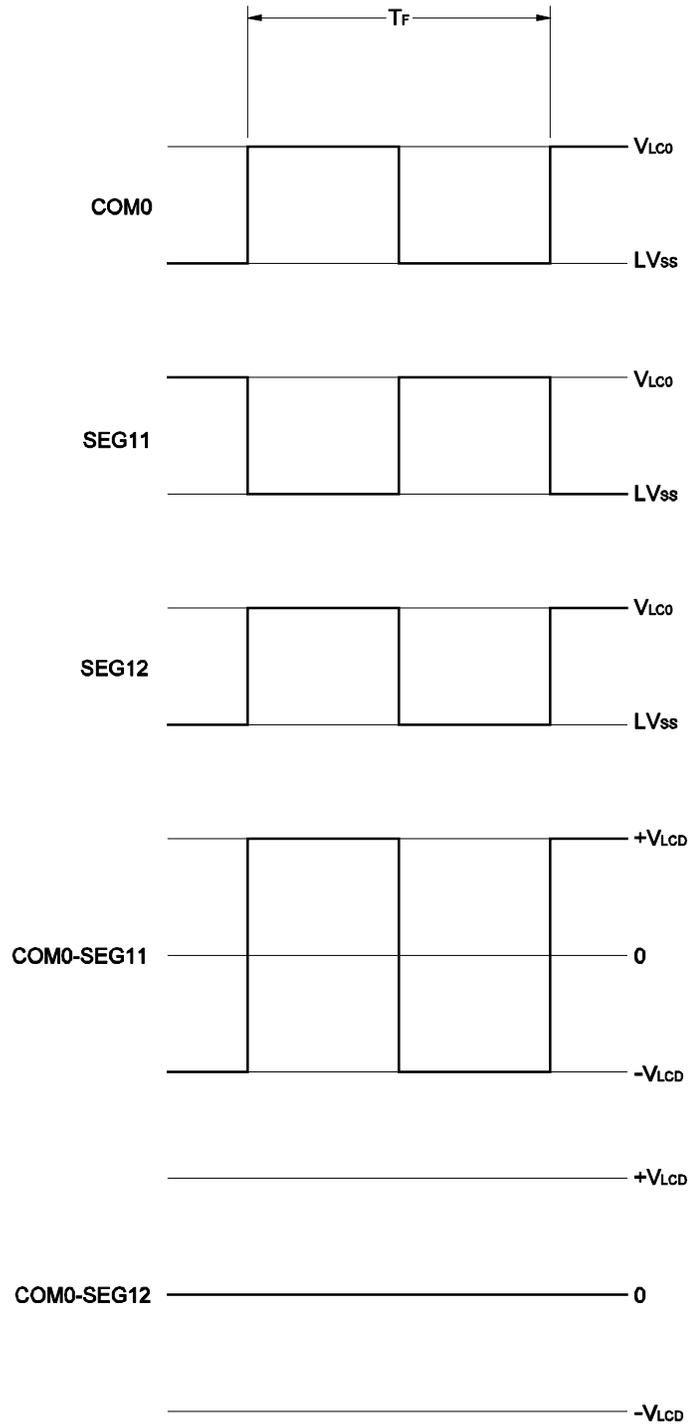


图 15-12. 静态 LCD 驱动波形举例



15.7.2 2-分时显示举例

图 15-14 显示的是具有图 15-13 所示的显示模式的 6-位 LCD 面板是怎样连接 78K0/LC3 芯片的 segment 信号 (SEG0 ~ SEG21) 和 common 信号 (COM0~COM1)。此例显示的是在 LCD 面板上显示数字“2345.6”。显示数据存储寄存器(FA40H to FA55H)的内容与此相对应。

下面描述在第四位数字上显示数字"3." (  )。为了在 LCD 面板上显示 "3."，必须在 common 信号 COM0 时序中根据表 15-6 将电压加在 SEG8~SEG15 上；请看图 15-13 segment 信号和 LCD segments 之间的关系。

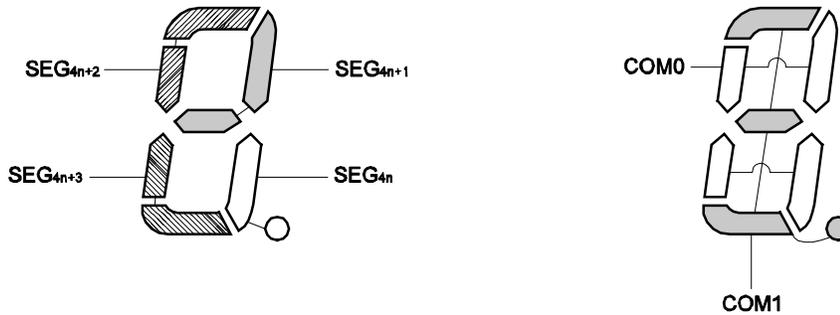
表 15-6. 选定和取消电压 (COM0 和 COM1)

Segment \ Common	SEG12	SEG13	SEG14	SEG15
COM0	选定	选定	取消	取消
COM1	取消	选定	选定	选定

根据表 15-5，显示数据存储器区 (FA4FH)依次必须包含 xx10。

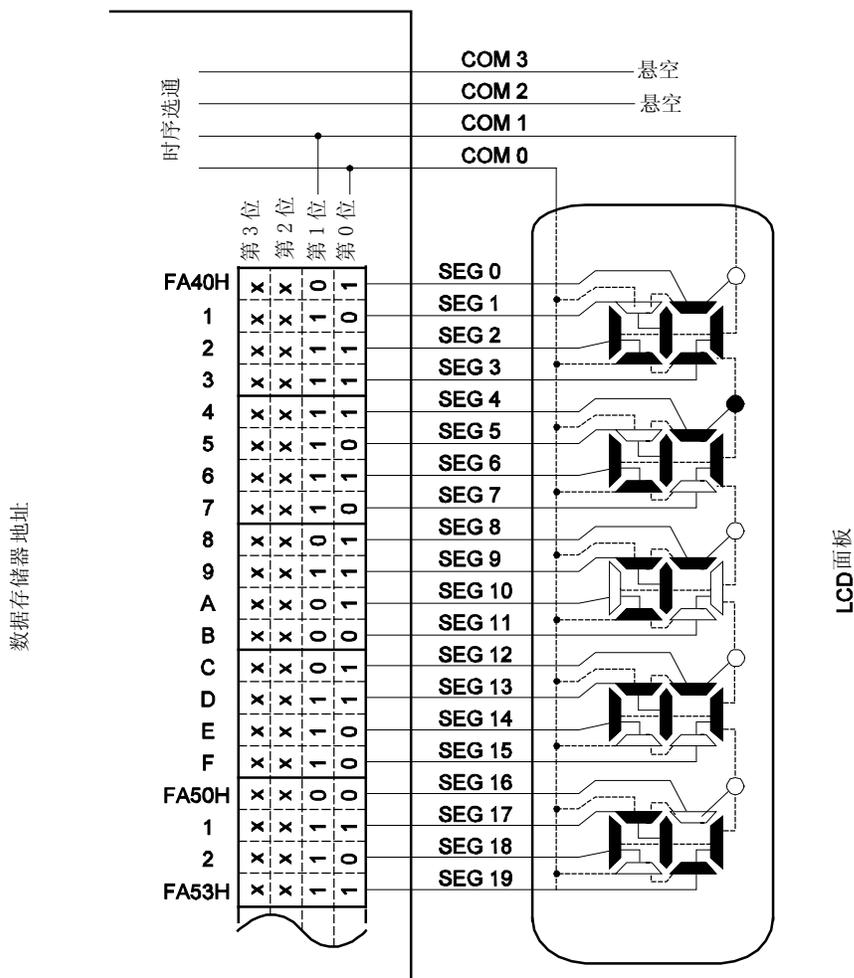
图 15-15 显示了 SEG15 信号和每一个 common 信号的 LCD 驱动波形。当选定电压在 COM0 时序中应用到 SEG15 时，LCD segment 产生相应的一个交替长方形波形，+VLCD/-VLCD。

图 15-13. 2 分时 LCD 显示图形和电极连接



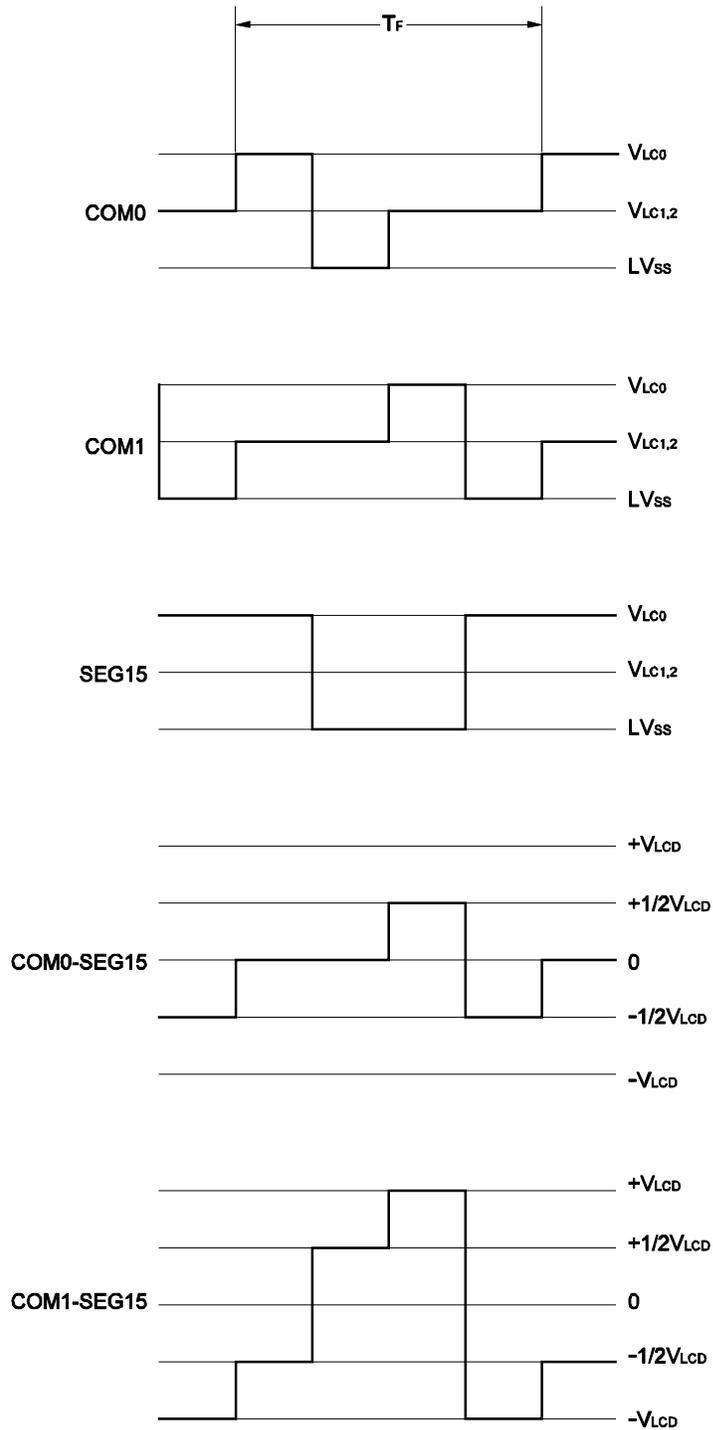
备注 n = 0 ~ 4

图 15-14. 2 分时 LCD 显示图形和电极连接



×: 因为使用 2 分时, 始终存储任意数据。

图 15-15. 2 分时 LCD 驱动波形举例 (1/2 偏压模式)



### 15.7.3 3-分时显示实例

图 15-17 显示的是具有图 15-16 所示的显示模式的 8-位 LCD 面板是怎样连接 78K0/LC3 芯片的 segment 信号 (SEG0 ~ SEG21) 和 common 信号 (COM0 ~ COM2)。此例显示的是在 LCD 面板上显示数字“23456.78”。显示数据存储区(FA40H to FA55H)的内容与此相对应。

下面描述在第三位数字上显示数字“6.” (  )。为了在 LCD 面板上显示 “6.”，必须在 common 信号 COM0 时序中根据表 15-7 将电压加在 SEG6~SEG8 上；请看 图 15-16 segment 信号 和 LCD segments 之间的关系。

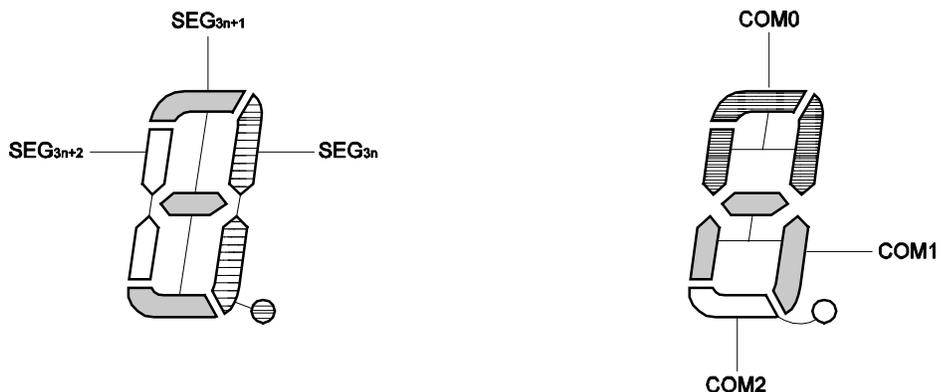
表 15-7. 选定和取消电压 (COM0 ~ COM2)

Segment \ Common	SEG6	SEG7	SEG8
COM0	取消	选定	选定
COM1	选定	选定	选定
COM2	选定	选定	-

根据表 15-7，显示数据存储区 (FA46H) 依次必须包含 x110。

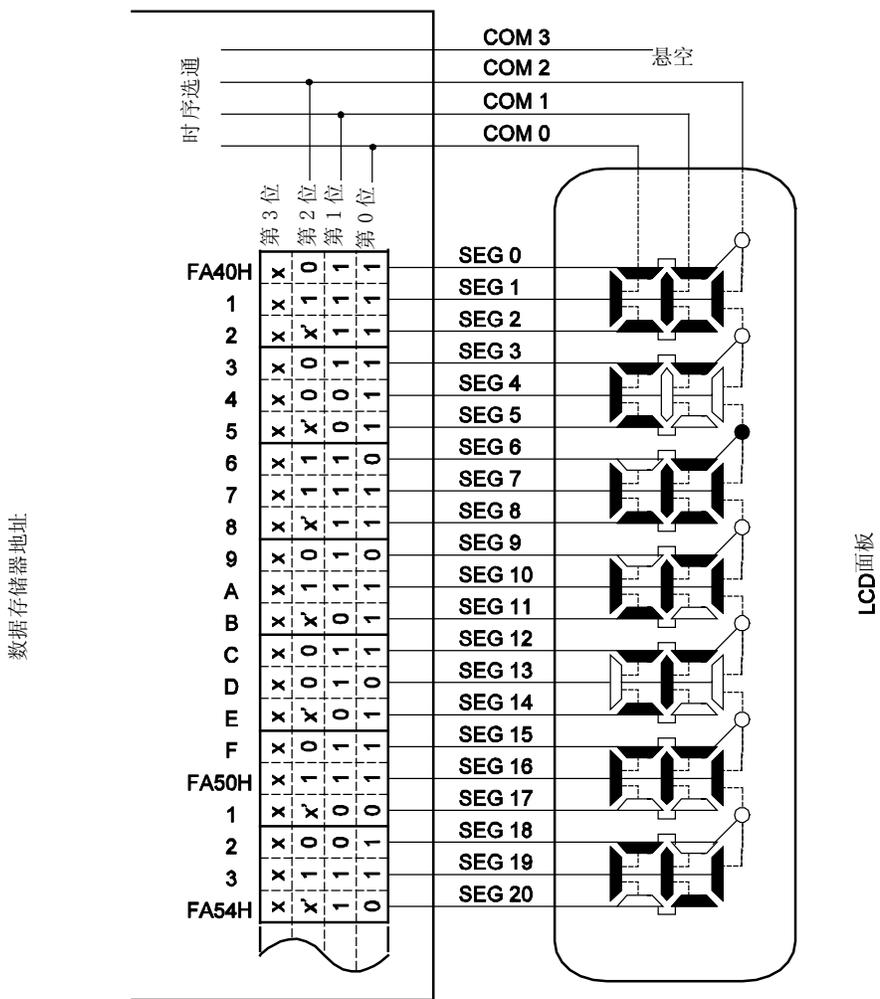
图 15-18 和图 15-9 分别显示的是 1/2 与 1/3 偏压模式下，SEG6 信号和每一个 common 信号的 LCD 驱动波形。当选定电压在 COM0 时序中应用到 SEG6 时，LCD segment 产生相应的一个交替长方形波形， $+V_{LCD}/-V_{LCD}$ 。

图 15-16. 3 分时 LCD 显示图形和电极连接



备注 n = 0 ~ 6

图 15-17. 3 分时 LCD 面板连接举例



x': 因为 LCD 面板没有相应的, 始终存储任意数据。

x: 因为使用 3 分时, 始终存储任意数据。

图 15-18.3 分时 LCD 驱动波形举例 (1/2 偏压模式)

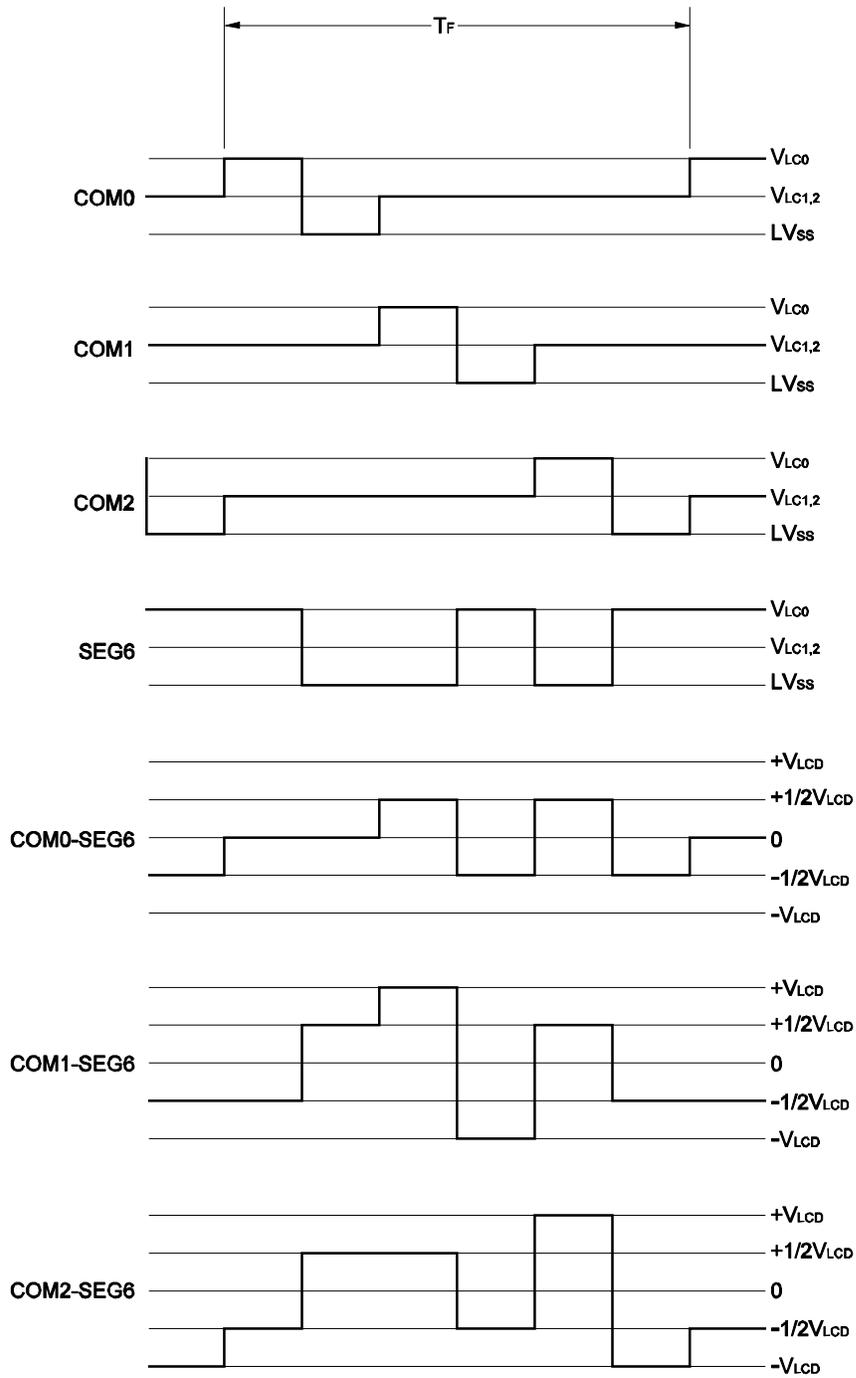
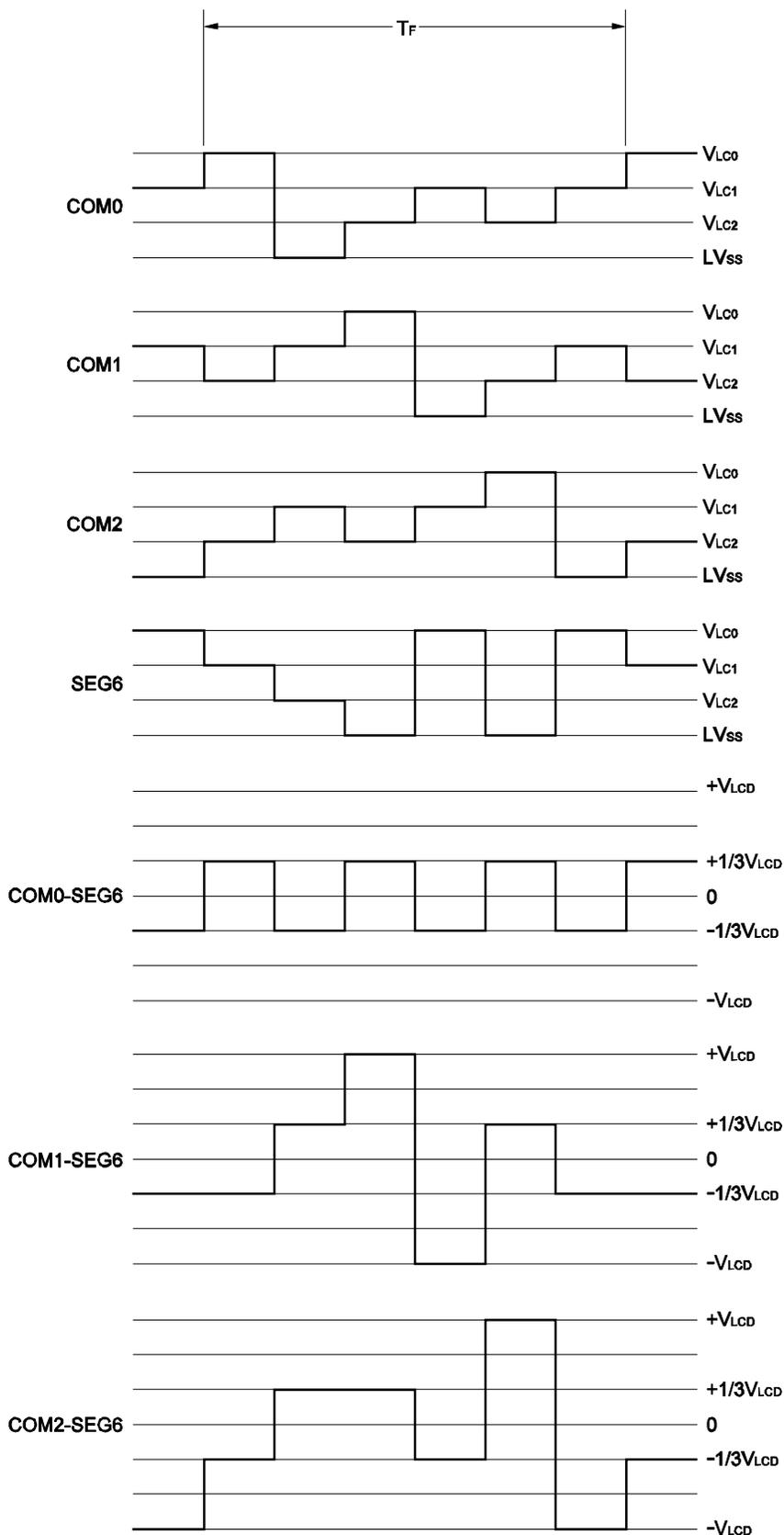


图 15-19. 3 分时 LCD 驱动波形举例 (1/3 偏压模式)



### 15.7.4 4-分时显示实例

图 15-21 显示的是具有图 15-20 所示的显示模式的 12-位 LCD 面板是怎样连接 78K0/LC3 芯片的 segment 信号 (SEG0 ~ SEG21) 和 common 信号 (COM0 ~ COM3)。此例显示的是在 LCD 面板上显示数字“23456.789012”。显示数据存储区(FA40H to FA55H)的内容与此相对应。

下面描述在第七位数字上显示数字“6.” (□)。为了在 LCD 面板上显示“6.”，必须在 common 信号 COM0 时序中根据表 15-8 将电压加在 SEG12~SEG13 上；请看图 15-20 segment 信号和 LCD segments 之间的关系。

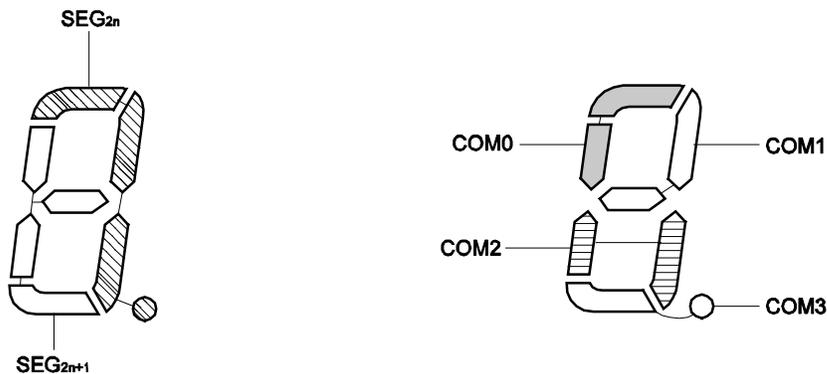
表 15-8. 选定和取消电压 (COM0 to COM3)

Segment \ Common	SEG12	SEG13
COM0	选定	选定
COM1	取消	选定
COM2	选定	选定
COM3	选定	选定

根据表 15-8，显示数据存储区 (FA4CH) 依次必须包含 1101。

图 15-22 显示的是 SEG12 信号和每一个 common 信号的 LCD 驱动波形。当选定电压在 COM0 时序中应用到 SEG12 时，LCD segment 产生相应的一个交替长方形波形， $+V_{LCD}/-V_{LCD}$ 。

图 15-20. 4 分时 LCD 显示图形和电极连接



备注 n = 0 ~ 10

图 15-22. 4 分时 LCD 面板连接举例

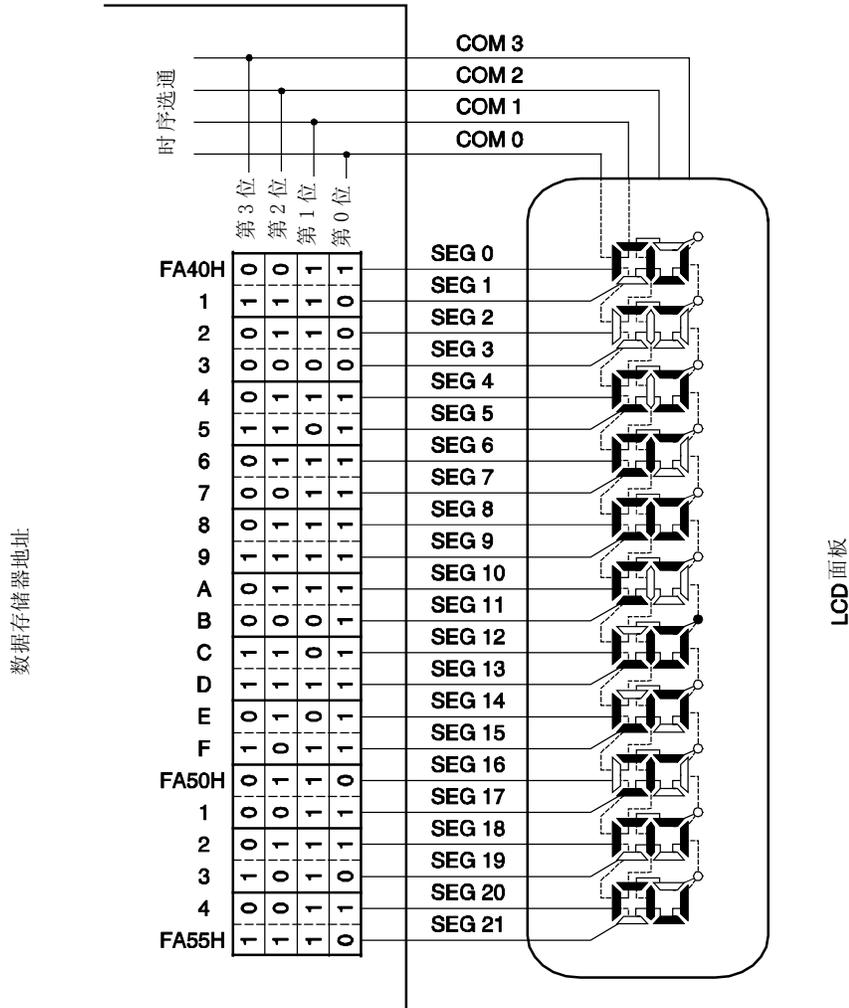
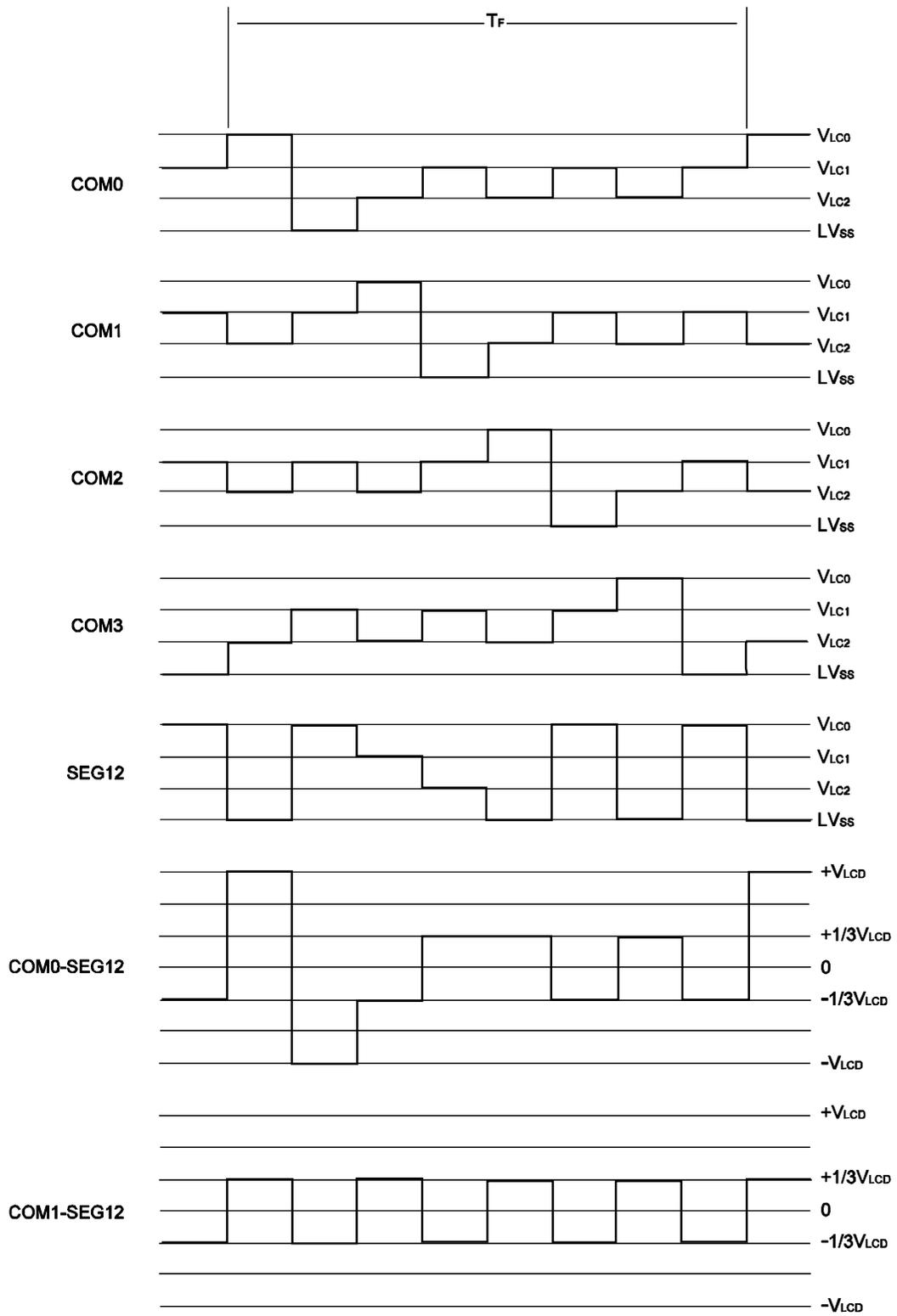


图 15-22. 4 分时 LCD 驱动波形举例 (1/3 偏压模式)



备注 COM2 ~ SEG12 和 COM3 ~ SEG12 的波形没有。

### 15.8 LCD 驱动电压 $V_{LC0}$ , $V_{LC1}$ , $V_{LC2}$ 和 $V_{LC3}$

78K0/LC3 中 LCD 驱动电源提供可由 2 种类型产生：内部电阻分压模式、外部电阻分压模式。

#### 15.8.1 内部电阻分压模式

78K0/LC3 内有分压电阻器提供产生 LCD 驱动电源。使用内部分压电阻器，LCD 驱动电源提供表 15-9 列出的偏压模式，不必使用外部分压电阻器。

表 15-9. LCD 驱动电压 (含片内分压电阻)

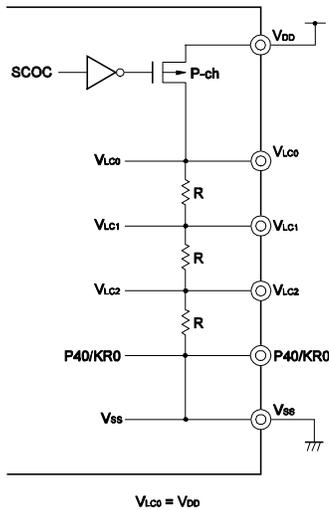
LCD 驱动电压引脚	偏压方式	无偏压 (静态)	1/2 偏压方式	1/3 偏压方式	1/4 偏压方式
$V_{LC0}$		$V_{LCD}$	$V_{LCD}$	$V_{LCD}$	$V_{LCD}$
$V_{LC1}$		$\frac{2}{3}V_{LCD}$	$\frac{1}{2}V_{LCD}$ <sup>Note</sup>	$\frac{2}{3}V_{LCD}$	$\frac{3}{4}V_{LCD}$
$V_{LC2}$		$\frac{1}{3}V_{LCD}$		$\frac{1}{3}V_{LCD}$	$\frac{2}{4}V_{LCD}$
$V_{LC3}$		$V_{SS}$	$V_{SS}$	$V_{SS}$	$\frac{1}{4}V_{LCD}$

注 1/2 偏压模式，必须连接  $V_{LC1}$  和  $V_{LC2}$  引脚。

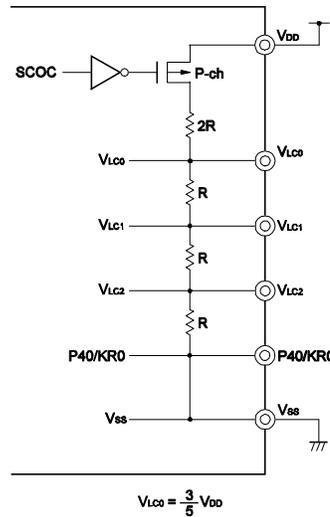
图 15-23 显示按表 15-9 产生 LCD 驱动电压的例子。

图 15-23. LCD 驱动电源连接举例(内部电阻分压模式) (1/2)

(a) 1/3 偏压模式和静态显示模式  
(MDSET1, MDSET0 = 0, 1)  
( $V_{DD} = 5V$ ,  $V_{LC0} = 5V$  的实例)

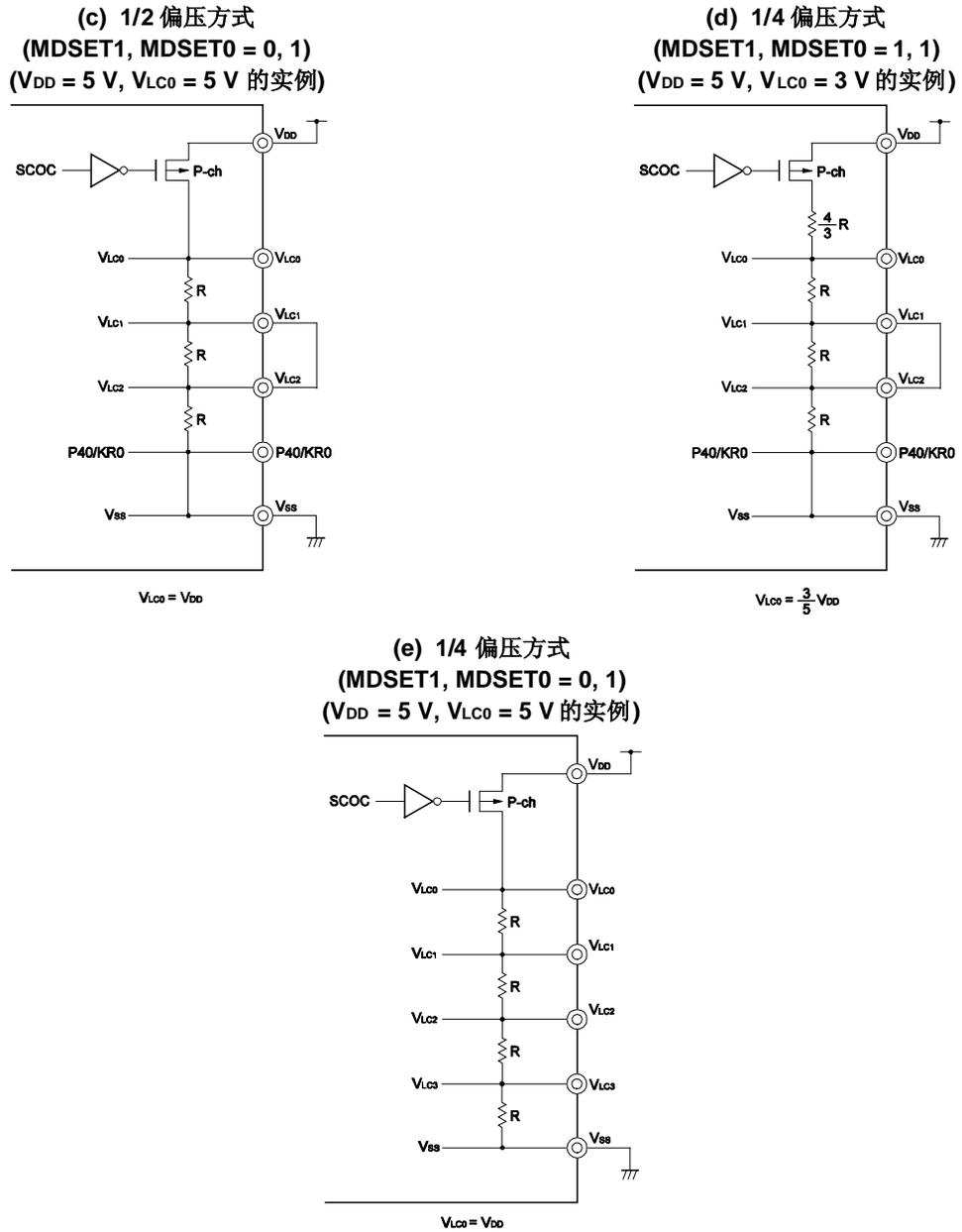


(b) 1/3 偏压模式和静态显示模式  
(MDSET1, MDSET0 = 1, 1)  
( $V_{DD} = 5V$ ,  $V_{LC0} = 3V$  的实例)



备注 在静态模式时推荐使用外部电阻分压方式，为了降低由分压电阻消耗的能耗。

图 15-23. LCD 驱动电源连接举例(内部电阻分压模式) (2/2)

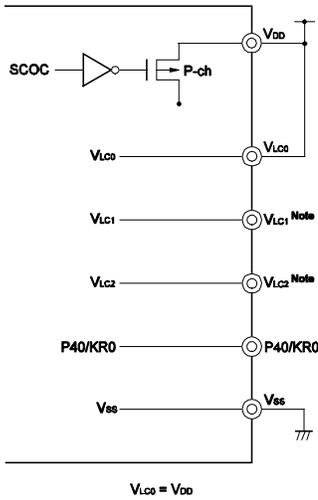


15.8.2 外部电阻分压模式

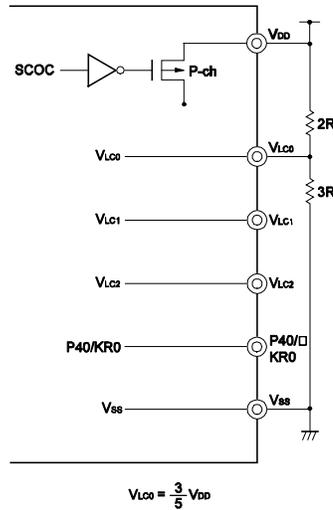
78K0/LC3 也可使用外边分压电阻器产生 LCD 驱动电源, 不使用内部电阻器。图 15-24 显示了相应各种偏压模式下 LCD 驱动电压连接例子。

图 15-24. LCD 驱动电压连接举例 (外部电阻分压模式) (1/2)

(a) 静态显示模式  
(MDSET1, MDSET0 = 0, 0)  
(V<sub>DD</sub> = 5 V, V<sub>LC0</sub> = 5 V 的实例)

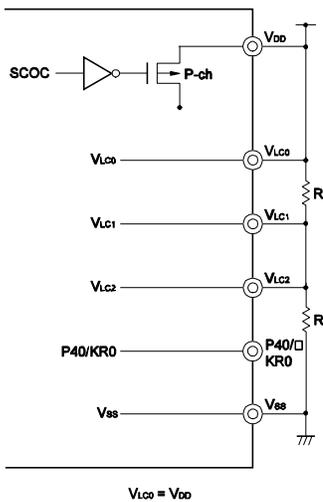


(b) 静态显示模式  
(MDSET1, MDSET0 = 0, 0)  
(V<sub>DD</sub> = 5 V, V<sub>LC0</sub> = 3 V 的实例)



注 直接将 VLC1 和 VLC2 连接到 GND 或 VLC0。

(c) 1/2 偏压方式  
(MDSET1, MDSET0 = 0, 0)  
(V<sub>DD</sub> = 5 V, V<sub>LC0</sub> = 5 V 的实例)



(d) 1/2 偏压方式  
(MDSET1, MDSET0 = 0, 0)  
(V<sub>DD</sub> = 5 V, V<sub>LC0</sub> = 3 V 的实例)

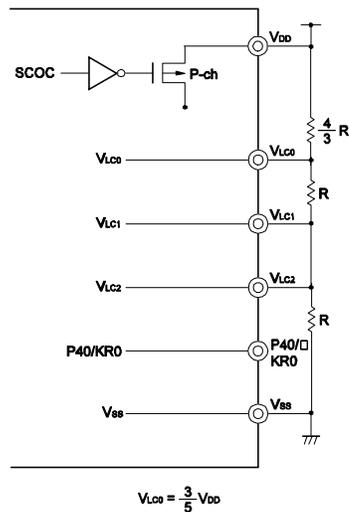
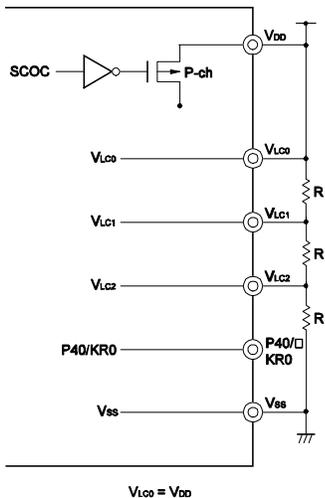
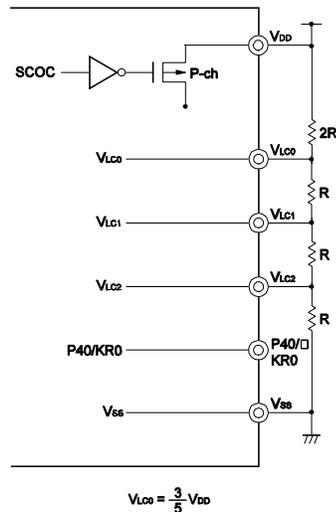


图 15-24. LCD 驱动电压连接举例 (外部电阻分压模式) (2/2)

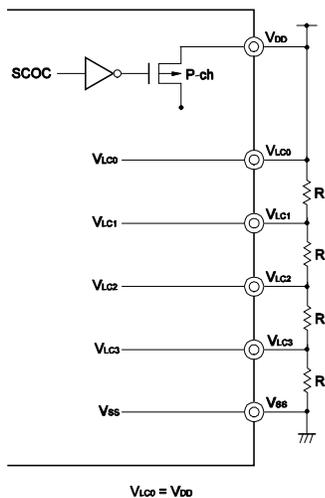
(e) 1/3 偏压方式  
(MDSET1, MDSET0 = 0, 0)  
( $V_{DD} = 5\text{ V}$ ,  $V_{LC0} = 5\text{ V}$  的实例)



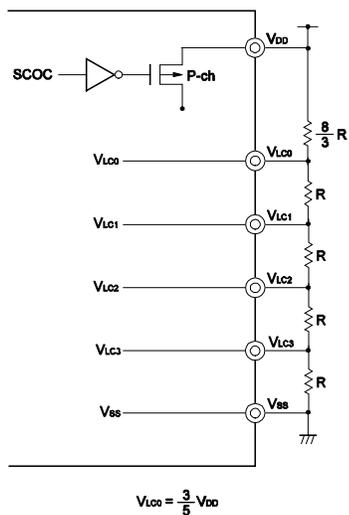
(f) 1/3 偏压方式  
(MDSET1, MDSET0 = 0, 0)  
( $V_{DD} = 5\text{ V}$ ,  $V_{LC0} = 3\text{ V}$  的实例)



(g) 1/4 偏压方式  
(MDSET1, MDSET0 = 0, 0)  
( $V_{DD} = 5\text{ V}$ ,  $V_{LC0} = 5\text{ V}$  的实例)



(h) 1/4 偏压方式  
(MDSET1, MDSET0 = 0, 0)  
( $V_{DD} = 5\text{ V}$ ,  $V_{LC0} = 3\text{ V}$  的实例)



## 第十六章 曼彻斯特发生器

### 16.1 曼彻斯特编码发生器的功能

曼彻斯特编码发生器有下列 3 类模式可以使用。

#### (1) 操作停止模式

这个模式在不运行曼彻斯特编码发生器/位序列缓冲器输出时使用。这个模式降低能耗。

详细内容，请参考 **16.4.1 操作停止模式**。

#### (2) 曼彻斯特编码发生器模式

这个模式用来从 MCGO 引脚传送曼彻斯特编码。

发送的位长度可以设置，并且允许各种不同的位长度。同时，8 位发送数据可以设置数据发送的输出电平和 LSB 优先或者 MSB 优先。

#### (3) 位序列缓冲器模式

这个模式用来从 MCGO 引脚传送位序列数据。

发送的位长度可以设置，并且允许各种不同的位长度。同时，8 位发送数据可以设置数据发送的输出电平和 LSB 优先或者 MSB 优先。

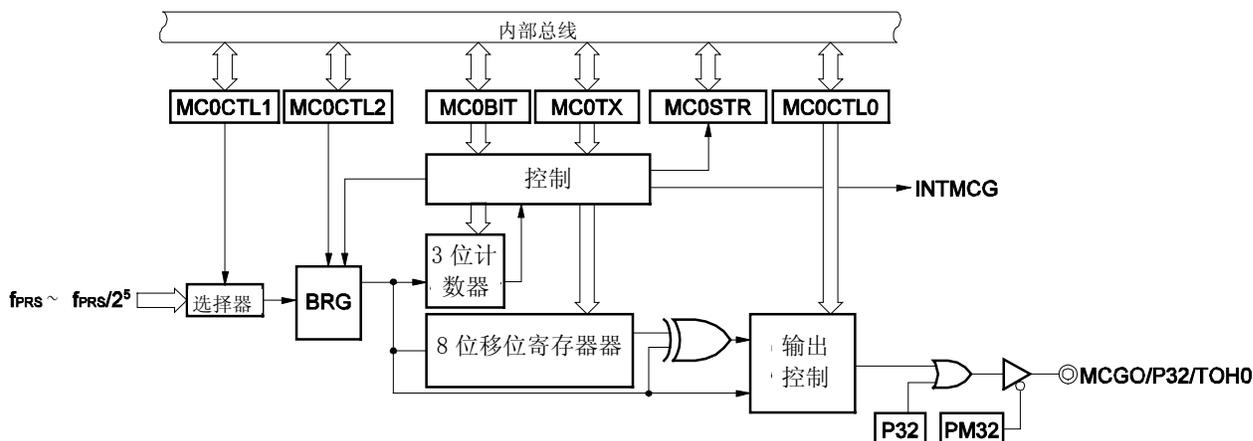
### 16.2 曼彻斯特编码发生器的配置

曼彻斯特编码发生器包括下列硬件。

表 16-1. 曼彻斯特编码发生器的配置

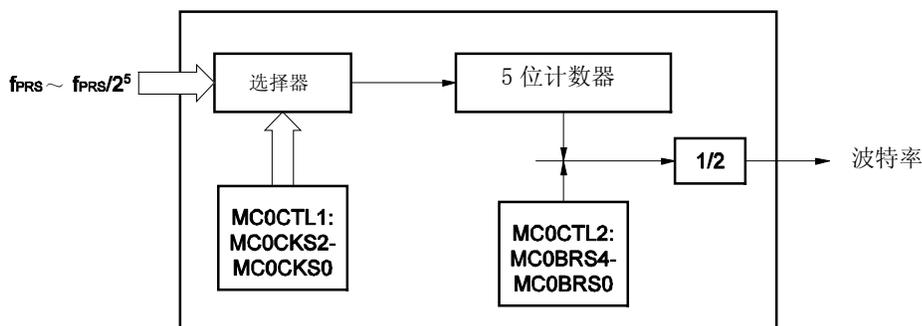
名称	配置
寄存器	MCG 传送缓冲器寄存器 (MC0TX) MCG 传送位计数规定寄存器(MC0BIT)
控制寄存器	MCG 控制寄存器 0 (MC0CTL0) MCG 控制寄存器 1 (MC0CTL1) MCG 控制寄存器 2 (MC0CTL2) MCG 状态寄存器 (MC0STR) 端口模式寄存器 3 (PM3) 端口寄存器 3 (P3)

图 16-1. 曼彻斯特编码发生器的方框图



备注 BRG: 波特率  
 fPRS: 外围硬件时钟频率  
 MC0BIT: MCG 传送位计数规定寄存器  
 MC0CTL2 to MC0CTL0: MCG 控制寄存器 2 ~ 0  
 MC0STR: MCG 状态寄存器  
 MC0TX: MCG 传送缓冲器寄存器

图 16-2. 波特率发生器的方框图



备注 fPRS: 外围硬件时钟频率  
 MC0CTL2, MC0CTL 1: MCG 控制寄存器 2, 1  
 MC0CKS2 to MC0CKS0: MC0CTL1 寄存器的位 2~0  
 MC0BRS4 to MC0BRS0: MC0CTL2 寄存器的位 4~0

(1) MCG 传送缓冲器寄存器 (MC0TX)

这个寄存器用来设置传送数据。MCG 控制寄存器 0 (MC0CTL0)的位 7(MC0PWR)是 1 期间，当数据写入时开始传送操作。写入 MC0TX 的数据通过 8 位移位寄存器转换成串行数据，并且输出给 MCGO 引脚。

曼彻斯特编码或者位序列数据使用 MCG 控制寄存器 0 (MC0CTL0)的位 1(MC0OSL)设置为输出码。

此寄存器由 8 位存储操作指令设置。

复位信号将此寄存器设置为 FFH。

**(2) MCG 传送位计数规定寄存器 (MC0BIT)**

这个寄存器用来设置传送位的个数。

在设置传送数据给 MC0TX 之前设置传送位计数给此寄存器。

连续传输中，发生传送开始中断(INTMCG)之后需要写入要传送的数据位数。但是，如果两次的传送位数相同，就不需要再次写入。

此寄存器由 8 位存储操作指令设置。

复位信号将此寄存器设置为 07H。

图 16-3. MCG 传送位规定寄存器 (MC0BIT)的格式

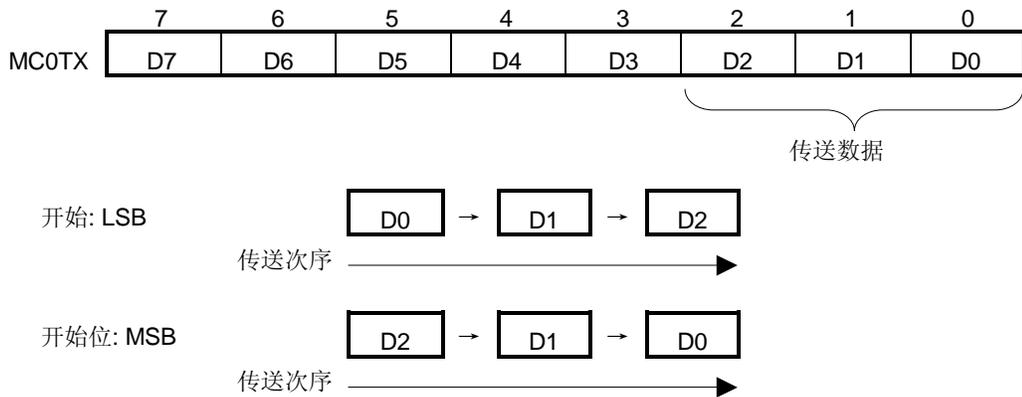
地址: FF4BH 复位后: 07H R/W

符号	7	6	5	4	3	<2>	<1>	<0>
MC0BIT	0	0	0	0	0	MC0BIT2	MC0BIT1	MC0BIT0

MC0BIT2	MC0BIT1	MC0BIT0	传送位计数设置
0	0	0	1 位
0	0	1	2 位
0	1	0	3 位
0	1	1	4 位
1	0	0	5 位
1	0	1	6 位
1	1	0	7 位
1	1	1	8 位

**备注** 当传送位数设置为 7 位或者更小时，不管 MSB/LSB 的设置，较低位总是作为发送开始位传送。

例如 当传送位设置为 3 位时，D7 ~ D0 写入 MCG 传送缓冲器寄存器 (MC0TX)。



### 16.3 控制曼彻斯特编码发生器的寄存器

下列 6 种寄存器用来控制曼彻斯特编码发生器。

- MCG 控制寄存器 0 (MC0CTL0)
- MCG 控制寄存器 1 (MC0CTL1)
- MCG 控制寄存器 2 (MC0CTL2)
- MCG 转台寄存器 (MC0STR)
- 端口模式 3 (PM3)
- 端口寄存器 3 (P3)

#### (1) MCG 控制寄存器 0 (MC0CTL0)

此寄存器用来设置操作模式和允许/禁止操作。

此寄存器可以由 1 位或 8 位存储操作指令设置。

复位信号将此寄存器设置为 10H。

图 16-4. MCG 控制寄存器 0 (MC0CTL0) 的格式

地址: FF4CH 复位后: 10H R/W

符号	<7>	6	5	<4>	3	2	<1>	<0>
MC0CTL0	MC0PWR	0	0	MC0DIR	0	0	MC0OSL	MC0OLV

MC0PWR	操作控制
0	操作停止
1	操作允许

MC0DIR	第一位规定
0	MSB
1	LSB

MC0OSL	数据格式
0	曼彻斯特编码
1	位序列数据

MC0OLV	传输挂起时的输出电平
0	低电平
1	高电平

**注意事项** 重写 MC0DIR, MC0OSL, 和 MC0OLV 位之前将 MC0PWR 位清零(0)(当 MC0PWR 位设置为 1 时, 可以通过 8 位存储操作指令重写这些位)。

**(2) MCG 控制寄存器 1 (MC0CTL1)**

此寄存器用来设置曼彻斯特编码发生器的基时钟。

此寄存器可以通过 8 位存储操作指令设置。

复位信号将此寄存器清零 00H。

**图 16-5. MCG 控制寄存器 1 的格式 (MC0CTL1)**

地址: FF4DH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
MC0CTL1	0	0	0	0	0	MC0CKS2	MC0CKS1	MC0CKS0

MC0CKS2	MC0CKS1	MC0CKS0	基时钟 (f <sub>CLK</sub> ) 选择 <sup>注1</sup>
0	0	0	F <sub>PRS</sub> <sup>注2</sup> (10 MHz)
0	0	1	f <sub>PRS</sub> /2 (5 MHz)
0	1	0	f <sub>PRS</sub> /2 <sup>2</sup> (2.5 MHz)
0	1	1	f <sub>PRS</sub> /2 <sup>3</sup> (1.25 MHz)
1	0	0	f <sub>PRS</sub> /2 <sup>4</sup> (625 kHz)
1	0	1	f <sub>PRS</sub> /2 <sup>5</sup> (312.5 kHz)
1	1	0	禁止设置
1	1	1	

- 注**
1. 如果外围硬件时钟(f<sub>PRS</sub>)运行在高速系统时钟 (f<sub>XH</sub>) (XSEL = 1)状态,f<sub>PRS</sub> 运行频率依据供电电压的不同而不同。
    - V<sub>DD</sub> = 2.7 ~ 5.5 V: f<sub>PRS</sub> ≤ 10 MHz
    - V<sub>DD</sub> = 1.8 ~ 2.7 V: f<sub>PRS</sub> ≤ 5 MHz
  2. 如果外围硬件时钟(f<sub>PRS</sub>) 运行在内部高速振荡时钟(f<sub>RH</sub>) (XSEL = 0), 当 1.8 V ≤ V<sub>DD</sub> < 2.7 V 时, MC0CKS2 = MC0CKS1 = MC0CKS0 = 0 (基时钟: f<sub>PRS</sub>)这种设置必须禁止。

**备注** 重写 MC0CKS2 ~ MC0CKS0 位之前, 先将 MC0CTL0 的位 7(MC0PWR)清零。

- 备注**
1. f<sub>PRS</sub>: 外围硬件时钟频率
  2. 括号内的数字是 f<sub>PRS</sub> = 10 MHz 时的参数。

### (3) MCG 控制寄存器 2 (MC0CTL2)

此寄存器用来设置传输波特率。

此寄存器可以通过 8 位存储操作指令设置。

复位信号将此寄存器设置为 1FH。

图 16-6. MCG 控制寄存器 2 (MC0CTL2)的格式

地址: FF4EH 复位后: 1FH R/W

符号	7	6	5	4	3	2	1	0
MC0CTL2	0	0	0	MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0

MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0	k	5 位计数器的输出时钟选择
0	0	0	×	×	4	f <sub>XCLK</sub> /4
0	0	1	0	0	4	f <sub>XCLK</sub> /4
0	0	1	0	1	5	f <sub>XCLK</sub> /5
0	0	1	1	0	6	f <sub>XCLK</sub> /6
0	0	1	1	1	7	f <sub>XCLK</sub> /7
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	1	0	0	28	f <sub>XCLK</sub> /28
1	1	1	0	1	29	f <sub>XCLK</sub> /29
1	1	1	1	0	30	f <sub>XCLK</sub> /30
1	1	1	1	1	31	f <sub>XCLK</sub> /31

- 注意事项**
1. 重写 MC0BRS4 ~ MC0BRS0 位之前要将 MC0CTL0 寄存器的位 7(MC0PWR)清零
  2. 5 位计数器再除以 2 得到的数值为波特率数值。

- 备注**
1. f<sub>XCLK</sub>: 由 MC0CTL1 寄存器的 MC0CKS2 ~ MC0CKS0 位选择的基时钟频率
  2. k: 由 MC0BRS4 ~ MC0BRS0 位设置的数值 (k = 4, 5, 6, 7, ..., 31)
  3. ×: 无关

### (4) MCG 状态寄存器 (MC0STR)

此寄存器用来指示曼彻斯特编码发生器的操作状态。

此寄存器可以通过 1 位或者 8 存储 操作指令读取。不能对其进行写操作。

发生复位信号或者 MC0PWR = 0 将此寄存器清零 00H。

图 16-7. MCG 状态寄存器 (MC0STR)

地址: FF47H 复位后: 00H R

符号	<7>	6	5	4	3	2	1	0
MC0STR	MC0TSF	0	0	0	0	0	0	0

MC0TSF	数据传输状态
0	<ul style="list-style-type: none"> <li>• 发生复位信号</li> <li>• MC0PWR = 0</li> <li>• 当一次传输完成时，下一个传输数据没有写入 MC0TX</li> </ul>
1	传输进行中

**注意事项** 连续传输过程中此标志一直为 1。在没有确认此标志已经清零的情况下，不要初始化传输操作。

## 16.4 曼彻斯特编码发生器的操作

曼彻斯特编码发生器有下面 3 种工作模式。

- 操作停止模式
- 曼彻斯特编码发生器模式
- 位序列缓冲器模式

### 16.4.1 操作停止模式

操作停止模式时不进行传输操作。因此，能耗就降低了。另外，这种模式下，P32/TOH0/MCGO 引脚可以用作普通的 I/O 端口。

#### (1) 寄存器描述

MCG 控制寄存器 0 (MC0CTL0) 用来设置操作停止模式。

为了设置操作停止模式，将 MC0CTL0 的位 7 (MC0PWR) 清零 0。

##### (a) MCG 控制寄存器 0 (MC0CTL0)

此寄存器可以通过 1 位或者 8 位存储操作指令设置。

复位信号将此寄存器设置为 10H。

地址: FF4CH 复位后: 10H R/W

符号	<7>	6	5	<4>	3	2	<1>	<0>
MC0CTL0	MC0PWR	0	0	MC0DIR	0	0	MC0OSL	MC0OLV

MC0PWR	操作控制
0	操作停止

### 16.4.2 曼彻斯特编码发生器模式

此模式用来使用 MCGO 引脚传输曼彻斯特编码格式的数据。

#### (1) 寄存器描述

MCG 控制寄存器 0 (MC0CTL0), MCG 控制寄存器 1 (MC0CTL1), 和 MCG 控制寄存器 2 (MC0CTL2)用来设置曼彻斯特编码发生器模式。

##### (a) MCG 控制寄存器 0 (MC0CTL0)

此寄存器可以通过 1 位或者 8 位寄存器存储操作指令设置。

复位信号将此寄存器设置为 10H。

地址: FF4CH 复位后: 10H R/W

符号	<7>	6	5	<4>	3	2	<1>	<0>
MC0CTL0	MC0PWR	0	0	MC0DIR	0	0	MC0OSL	MC0OLV

MC0PWR	操作控制
0	操作停止
1	操作允许

MC0DIR	第一位规定
0	MSB
1	LSB

MC0OSL	数据格式
0	曼彻斯特编码
1	位序列数据

MC0OLV	传输挂起时的输出电平
0	低电平
1	高电平

**注意事项** 重写 MC0DIR, MC0OSL, 和 MC0OLV 位之前将 MC0PWR 位清零(0)(当 MC0PWR 位设置为 1 时, 可以通过 8 位存储操作指令重写这些位)。

**(b) MCG 控制寄存器 1 (MC0CTL1)**

此寄存器用来设置曼彻斯特编码发生器的基时钟。

此寄存器可以通过 8 位存储操作指令设置。

复位信号将此寄存器清零 00H。

地址: FF4DH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
MC0CTL1	0	0	0	0	0	MC0CKS2	MC0CKS1	MC0CKS0

MC0CKS2	MC0CKS1	MC0CKS0	基时钟 (f <sub>CLK</sub> ) 选择 <sup>注1</sup>
0	0	0	F <sub>PRS</sub> <sup>注2</sup> (10 MHz)
0	0	1	f <sub>PRS</sub> /2 (5 MHz)
0	1	0	f <sub>PRS</sub> /2 <sup>2</sup> (2.5 MHz)
0	1	1	f <sub>PRS</sub> /2 <sup>3</sup> (1.25 MHz)
1	0	0	f <sub>PRS</sub> /2 <sup>4</sup> (625 kHz)
1	0	1	f <sub>PRS</sub> /2 <sup>5</sup> (312.5 kHz)
1	1	0	禁止设置
1	1	1	

- 注**
1. 如果外围硬件时钟(f<sub>PRS</sub>)运行在高速系统时钟 (f<sub>XH</sub>) (XSEL = 1)状态,f<sub>PRS</sub> 运行频率依据供电电压的不同而不同。
    - V<sub>DD</sub> = 2.7 ~ 5.5 V: f<sub>PRS</sub> ≤ 10 MHz
    - V<sub>DD</sub> = 1.8 ~ 2.7 V: f<sub>PRS</sub> ≤ 5 MHz
  2. 如果外围硬件时钟(f<sub>PRS</sub>) 运行在内部高速振荡时钟(f<sub>RH</sub>) (XSEL = 0), 当 1.8 V ≤ V<sub>DD</sub> < 2.7 V 时, MC0CKS2 = MC0CKS1 = MC0CKS0 = 0 (基时钟: f<sub>PRS</sub>)这种设置必须禁止。

**备注** 重写 MC0CKS2 ~ MC0CKS0 位之前, 先将 MC0CTL0 的位 7(MC0PWR)清零。

- 备注**
1. f<sub>PRS</sub>: 外围硬件时钟频率
  2. 括号内的数字是 f<sub>PRS</sub> = 10 MHz 时的参数。

**(c) MCG 控制寄存器 2 (MC0CTL2)**

此寄存器用来设置传输波特率。

此寄存器可以通过 8 位存储操作指令设置。

复位信号将此寄存器设置为 1FH。

地址: FF4EH 复位后: 1FH R/W

符号	7	6	5	4	3	2	1	0
MC0CTL2	0	0	0	MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0

MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0	k	5 位计数器的输出时钟选择
0	0	0	×	×	4	f <sub>XCLK</sub> /4
0	0	1	0	0	4	f <sub>XCLK</sub> /4
0	0	1	0	1	5	f <sub>XCLK</sub> /5
0	0	1	1	0	6	f <sub>XCLK</sub> /6
0	0	1	1	1	7	f <sub>XCLK</sub> /7
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	1	0	0	28	f <sub>XCLK</sub> /28
1	1	1	0	1	29	f <sub>XCLK</sub> /29
1	1	1	1	0	30	f <sub>XCLK</sub> /30
1	1	1	1	1	31	f <sub>XCLK</sub> /31

**注意事项** 1. 重写 MC0BRS4 ~ MC0BRS0 位之前要将 MC0CTL0 寄存器的位 7(MC0PWR)清零  
2. 5 位计数器再除以 2 得到的数值为波特率数值。

**备注** 1. f<sub>XCLK</sub>: 由 MC0CTL1 寄存器的 MC0CKS2 ~ MC0CKS0 位选择的基时钟频率  
2. k: 由 MC0BRS4 ~ MC0BRS0 位设置的数值 (k = 4, 5, 6, 7, ..., 31)  
3. ×: 无关

**<1> 波特率**

波特率可以通过下列公式计算。

• 波特率 =  $\frac{f_{XCLK}}{2 \times k}$  [bps]

f<sub>XCLK</sub>: 由 MC0CTL1 寄存器的 MC0CKS2 ~ MC0CKS0 位选择的基时钟频率

k: 由 MC0CTL2 寄存器的 MC0BRS4 ~ MC0BRS0 位设置的数值(k = 4, 5, 6, ..., 31)

<2> 波特率的误差

波特率的误差可以通过下列公式计算。

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

注意事项 传输期间要保持波特率误差在接收目的方允许的误差范围之内。

例如: 基时钟频率 = 2.5 MHz = 2,500,000 Hz  
 MC0CTL2 寄存器 MC0BRS4 ~ MC0BRS0 位的设置数值=10000B (k = 16)  
 目标波特率 = 76,800 bps

$$\begin{aligned} \text{波特率} &= 2.5 \text{ M}/(2 \times 16) \\ &= 2,500,000/(2 \times 16) = 78125 \text{ [bps]} \end{aligned}$$

$$\begin{aligned} \text{误差} &= (78,125/76,800 - 1) \times 100 \\ &= 1.725 \text{ [\%]} \end{aligned}$$

<3> 波特率设置的实例

波特率 [bps]	f <sub>PRS</sub> = 10.0 MHz				f <sub>PRS</sub> = 8.38 MHz				f <sub>PRS</sub> = 8.0 MHz				f <sub>PRS</sub> = 6.0 MHz			
	MC0CKS2 ~ MC0CKS0	k	计算数值	ERR [%]	MC0CKS2 ~ MC0CKS0	k	计算数值	ERR [%]	MC0CKS2 ~ MC0CKS0	k	计算数值	ERR [%]	MC0CKS2 ~ MC0CKS0	k	计算数值	ERR [%]
4800	-	-	-	-	5, 6, or 7	27	4850	1.03	5, 6, or 7	26	4808	0.16	5, 6, or 7	20	4688	-2.34
9600	5, 6, or 7	16	9766	1.73	4	27	9699	1.03	5, 6, or 7	13	9615	0.16	4	20	9375	-2.34
19200	5	8	19531	1.73	3	27	19398	1.03	4	13	19231	0.16	4	10	18750	-2.34
31250	4	10	31250	0	2	17	30809	-1.41	4	8	31250	0	2	24	31250	0
38400	4	8	39063	1.73	2	27	38796	1.03	3	13	38462	0.16	2	20	37500	-2.34
56000	3	11	56818	1.46	2	19	55132	-1.55	3	9	55556	-0.79	1	27	55556	-0.79
62500	2	20	62500	0	2	17	61618	-1.41	3	8	62500	0	2	12	62500	0
76800	2	16	78125	1.73	1	27	77592	1.03	2	13	76923	0.16	2	10	75000	-2.34
115200	1	22	113636	-1.36	2	9	116389	1.03	1	17	117647	2.12	1	13	115385	0.16
125000	1	20	125000	0	1	17	123235	-1.41	1	16	125000	0	1	12	125000	0
153600	1	16	156250	1.73	2	7	149643	-2.58	1	13	153846	0.16	1	10	150000	-2.34
250000	1	10	250000	0	1	8	261875	4.75	1	8	250000	0	1	6	250000	0
					0	17	246471	-1.41								

备注 MC0CKS2 ~ MC0CKS0: MCG 控制寄存器 1 (MC0CTL1) 的位 2~0(基时钟 (f<sub>CLK</sub>))的设置  
 k: MCG 控制寄存器 2 (MC0CTL2) (k = 4, 5, 6, ..., 31)位 4~0 (MC0BRS4 ~ MC0BRS0)设置的数值  
 f<sub>PRS</sub>: 外围硬件时钟频率  
 ERR: 波特率误差

**(d) 端口模式寄存器 3 (PM3)**

此寄存器由位单元设置端口 3 为输入/输出模式。

当使用 P32/TOH0/MCGO 引脚用作曼彻斯特编码输出时，将 PM32 清零 0，并且将 P32 的输出锁存设置为 0。

PM3 可以通过 1 位或者 8 位存储操作指令设置。

复位信号将此寄存器设置为 FFH。

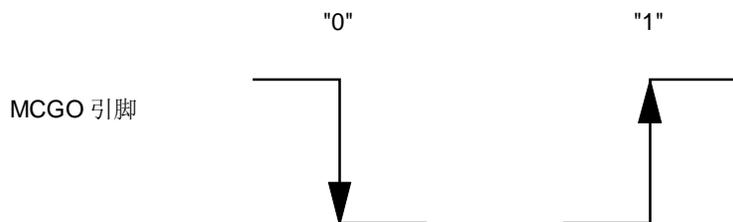
地址: FF23H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34	PM33	PM32	PM31	1

PM3n	P3n 引脚 I/O 模式选择 (n = 1 ~ 4)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

**(2) 曼彻斯特编码输出“0”和“1”的格式**

78K0/LC3 中曼彻斯特编码输出“0”和“1”的格式如下。



(3) 传输操作

曼彻斯特编码发生器模式下，数据能够以 1- ~ 8-位为单元进行传输。数据位以曼彻斯特编码格式传输。如果 MCG 控制寄存器 0 (MC0CTL0) 的位 7(MC0PWR)设置为 1，就允许传输操作。

传输挂起时的输出数值通过 MC0CTL0 寄存器的位 0(MC0OLV)设置。

在将传输数据位长度写入 MCG 传输位计数规定寄存器(MC0BIT)之后，通过在 MCG 传输缓冲器寄存器 (MC0TX) 中写入一个数据，传输方能开始 (MC0TX)。在传输开始时，MC0BIT 数据传入 3-位计数器，并且 MC0TX 数据传入 8-位移位寄存器。在 MC0TX 数值传入 8-位移位寄存器时，产生中断请求信号 (INTMCG)。8-位移位寄存器由波特率时钟连续移位，并且与波特率时钟异或之后从 MCGO 引脚输出。

当执行连续传输时，在数据传输期间，发生 INTMCG 之后，将下一个数据写入 MC0BIT 和 MC0TX。

为了连续传输，在图 16-8 中的(3)和(4)期间必须要将下一个传输数据写入 MC0TX。连续传输期间，在写 MC0TX 之前重写 MC0BIT。

图 16-8. 曼彻斯特编码发生器模式的时序(LSB 优先) (1/4)

(1) 传输时序 (MC0OLV = 1, 总的传输位长度 = 8 位)

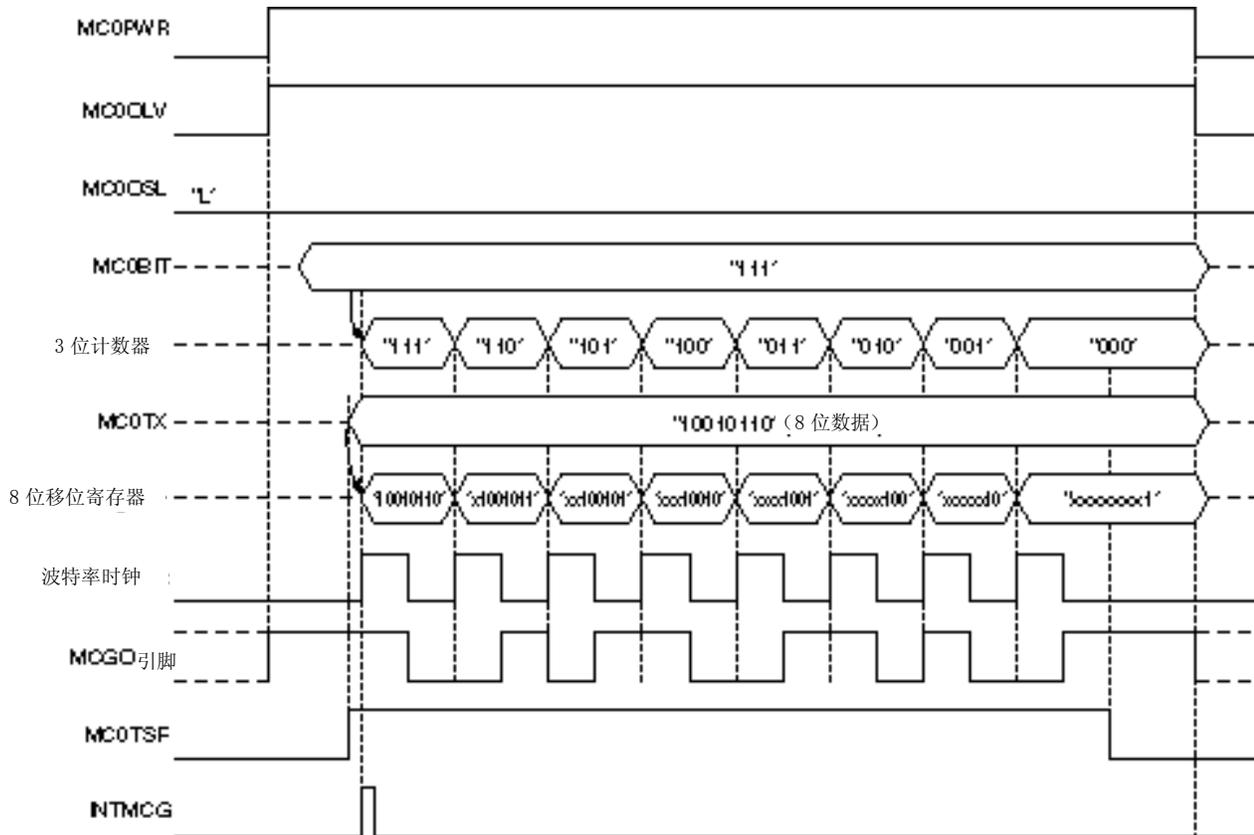


图 16-8. 曼彻斯特编码发生器模式的时序(LSB 优先) (2/4)

(2) 传输时序 (MC0OLV = 0, 总的传输位长度 = 8 位)

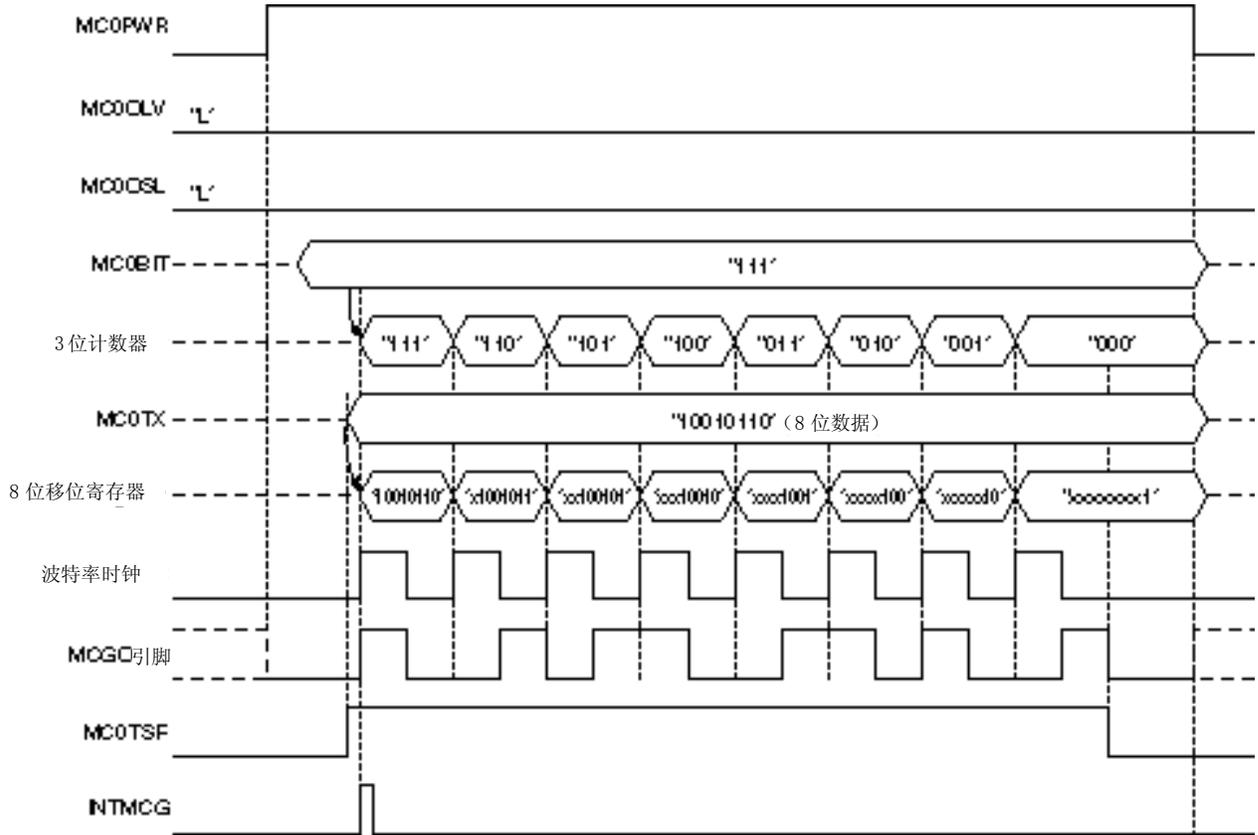
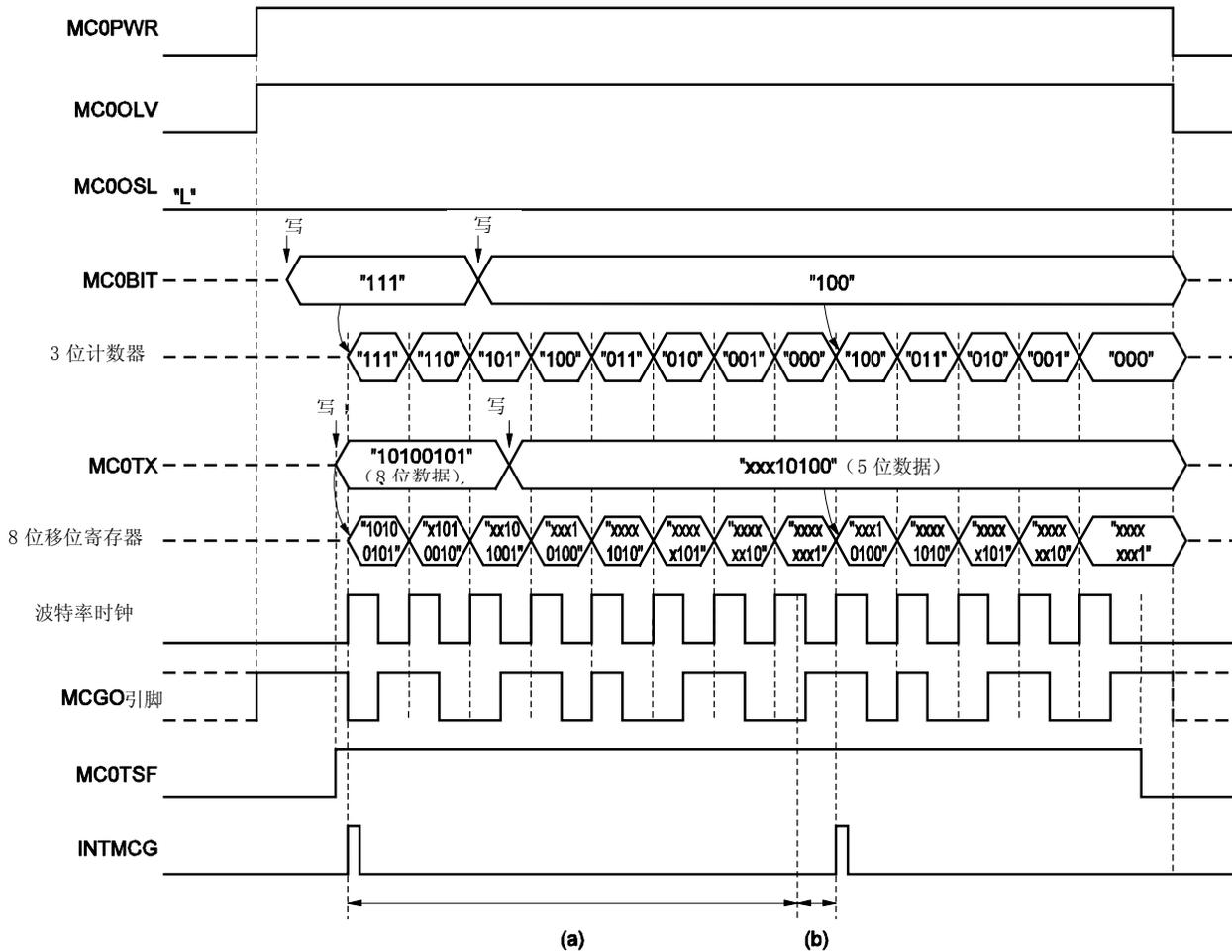


图 16-8. 曼彻斯特编码发生器模式的时序(LSB 优先) (3/4)

(3) 传输时序 (MC0OLV = 1, 总的传输位长度 = 13 位)



(a): “8-位传输期间” - (b)

(b): “波特率的 1/2 周期” + 传输数据的最后一位之前的 1 时钟 (f<sub>xCLK</sub>)

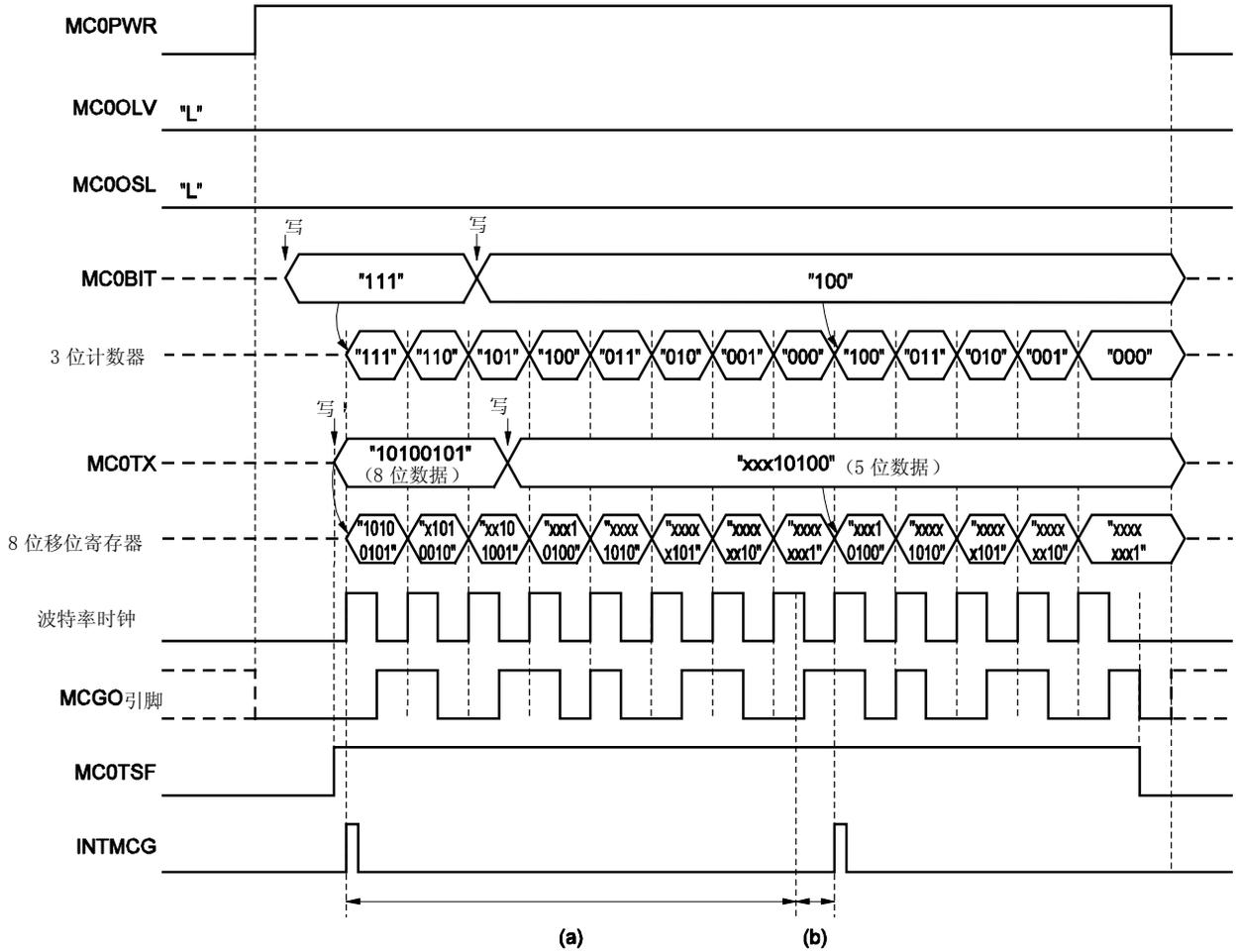
f<sub>xCLK</sub>: 使用 MC0CTL1 寄存器的 MC0CKS2 ~ MC0CKS0 位选择的操作基时钟频率

最后一位: 当 3-位计数器 = 000 时的传输位

**注意事项** 连续传输期间在时段(a)期间必须将下一个传输数据写入 MC0TX。如果在时段(b)期间写入，下一个数据传输在最后一位传输完成之后两个时钟(f<sub>xCLK</sub>)开始。  
连续传输期间，在写 MC0TX 之前重写 MC0BIT。

图 16-8. 曼彻斯特编码发生器模式的时序(LSB 优先) (4/4)

(4) 传输时序 (MC0OLV = 0, 总的传输位长度= 13 bits)



(a): “8-位传输期间” - (b)

(b): “波特率的 1/2 周期 + 传输数据的最后一位之前的 1 时钟 (fxCLK)

fxCLK: 使用 MC0CTL1 寄存器的 MC0CKS2 ~ MC0CKS0 位选择的操作基时钟频率

最后一位:当 3-位计数器 = 000 时的传输位

**注意事项** 连续传输期间在时段(a)期间必须将下一个传输数据写入 MC0TX。如果在时段(b)期间写入，下一个数据传输在最后一位传输完成之后两个时钟 (fxCLK) 开始。  
连续传输期间，在写 MC0TX 之前重写 MC0BIT。

### 16.4.3 位序列缓冲器模式

此模式用 MCGO 引脚输出序列信号。

#### (1) 寄存器描述

MCG 控制寄存器 0 (MC0CTL0), MCG 控制寄存器 1 (MC0CTL1), 和 MCG 控制寄存器 2 (MC0CTL2) 用来设置位序列缓冲器模式。

##### (a) MCG 控制寄存器 0 (MC0CTL0)

此寄存器由 1-位或者 8-位存储操作指令设置。

复位信号将此寄存器设置为 10H。

地址: FF4CH 复位后: 10H R/W

符号	<7>	6	5	<4>	3	2	<1>	<0>
MC0CTL0	MC0PWR	0	0	MC0DIR	0	0	MC0OSL	MC0OLV

MC0PWR	操作控制
0	操作停止
1	操作允许

MC0DIR	第一位规定
0	MSB
1	LSB

MC0OSL	数据格式
0	曼彻斯特编码
1	位序列数据

MC0OLV	传输挂起时的输出电平
0	低电平
1	高电平

**注意事项** 重写 MC0DIR, MC0OSL, 和 MC0OLV 位之前将 MC0PWR 位清零(0)(当 MC0PWR 位设置为 1 的同时, 可以通过 8 位存储操作指令重写这些位)。

**(b) MCG 控制寄存器 1 (MC0CTL1)**

此寄存器用来设置曼彻斯特编码发生器的基时钟。

此寄存器可以通过 8 位存储操作指令设置。

复位信号将此寄存器清零 00H。

地址: FF4DH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
MC0CTL1	0	0	0	0	0	MC0CKS2	MC0CKS1	MC0CKS0

MC0CKS2	MC0CKS1	MC0CKS0	基时钟 (fxCLK) 选择
0	0	0	f <sub>PRS</sub> (10 MHz)
0	0	1	f <sub>PRS</sub> /2 (5 MHz)
0	1	0	f <sub>PRS</sub> /2 <sup>2</sup> (2.5 MHz)
0	1	1	f <sub>PRS</sub> /2 <sup>3</sup> (1.25 MHz)
1	0	0	f <sub>PRS</sub> /2 <sup>4</sup> (625 kHz)
1	0	1	f <sub>PRS</sub> /2 <sup>5</sup> (312.5 kHz)
1	1	0	禁止设置
1	1	1	

**备注** 重写 MC0CKS2 ~ MC0CKS0 位之前, 先将 MC0CTL0 的位 7(MC0PWR)清零。

- 备注**
1. f<sub>PRS</sub>: 外围硬件时钟频率
  2. 括号内的数字是 f<sub>PRS</sub> = 10 MHz 时的参数。

**(c) MCG 控制寄存器 2 (MC0CTL2)**

此寄存器用来设置传输波特率。

此寄存器可以通过 8 位存储操作指令设置。

复位信号将此寄存器设置为 1FH。

地址: FF4EH 复位后: 1FH R/W

符号	7	6	5	4	3	2	1	0
MC0CTL2	0	0	0	MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0

MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0	k	5-位计数器的输出时钟选择
0	0	0	x	x	4	f <sub>XCLK</sub> /4
0	0	1	0	0	4	f <sub>XCLK</sub> /4
0	0	1	0	1	5	f <sub>XCLK</sub> /5
0	0	1	1	0	6	f <sub>XCLK</sub> /6
0	0	1	1	1	7	f <sub>XCLK</sub> /7
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	1	0	0	28	f <sub>XCLK</sub> /28
1	1	1	0	1	29	f <sub>XCLK</sub> /29
1	1	1	1	0	30	f <sub>XCLK</sub> /30
1	1	1	1	1	31	f <sub>XCLK</sub> /31

**注意事项 1.** 重写 MC0BRS4 ~ MC0BRS0 位之前要将 MC0CTL0 寄存器的位 7(MC0PWR)清零

**2.** 5 位计数器再除以 2 得到的数值为波特率数值。

**备注 1.** f<sub>XCLK</sub>:由 MC0CTL1 寄存器的 MC0CKS2 ~ MC0CKS0 位选择的基时钟频率

**2.** k: 由 MC0BRS4 ~ MC0BRS0 位设置的数值 (k = 4, 5, 6, 7, ..., 31)

**3.** x: 无关

**<1> 波特率**

波特率可以通过下列公式计算。

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

f<sub>XCLK</sub>: 由 MC0CTL1 寄存器的 MC0CKS2 ~ MC0CKS0 位选择的基时钟频率

k: 由 MC0CTL2 寄存器的 MC0BRS4 ~ MC0BRS0 位设置的数值(k = 4, 5, 6, ..., 31)

<2> 波特率的误差

波特率的误差可以通过下列公式计算。

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

注意事项 传输期间要保持波特率误差在接收目的方允许的误差范围之内。

例如: 基时钟频率 = 2.5 MHz = 2,500,000 Hz  
 MC0CTL2 寄存器 MC0BRS4 ~ MC0BRS0 位的设置数值=10000B (k = 16)  
 目标波特率 = 76,800 bps

$$\begin{aligned} \text{波特率} &= 2.5 \text{ M}/(2 \times 16) \\ &= 2,500,000/(2 \times 16) = 78125 \text{ [bps]} \end{aligned}$$

$$\begin{aligned} \text{误差} &= (78,125/76,800 - 1) \times 100 \\ &= 1.725 \text{ [\%]} \end{aligned}$$

<3> 波特率设置的实例

波特率 [bps]	f <sub>PRS</sub> = 10.0 MHz				f <sub>PRS</sub> = 8.38 MHz				f <sub>PRS</sub> = 8.0 MHz				f <sub>PRS</sub> = 6.0 MHz			
	MC0CKS2 ~ MC0CKS0	k	计算值	ERR [%]	MC0CKS2 ~ MC0CKS0	k	计算值	ERR [%]	MC0CKS2 ~ MC0CKS0	k	计算值	ERR [%]	MC0CKS2 ~ MC0CKS0	k	计算值	ERR [%]
4800	-	-	-	-	5, 6, or 7	27	4850	1.03	5, 6, or 7	26	4808	0.16	5, 6, or 7	20	4688	-2.34
9600	5, 6, or 7	16	9766	1.73	4	27	9699	1.03	5, 6, or 7	13	9615	0.16	4	20	9375	-2.34
19200	5	8	19531	1.73	3	27	19398	1.03	4	13	19231	0.16	4	10	18750	-2.34
31250	4	10	31250	0	2	17	30809	-1.41	4	8	31250	0	2	24	31250	0
38400	4	8	39063	1.73	2	27	38796	1.03	3	13	38462	0.16	2	20	37500	-2.34
56000	3	11	56818	1.46	2	19	55132	-1.55	3	9	55556	-0.79	1	27	55556	-0.79
62500	2	20	62500	0	2	17	61618	-1.41	3	8	62500	0	2	12	62500	0
76800	2	16	78125	1.73	1	27	77592	1.03	2	13	76923	0.16	2	10	75000	-2.34
115200	1	22	113636	-1.36	2	9	116389	1.03	1	17	117647	2.12	1	13	115385	0.16
125000	1	20	125000	0	1	17	123235	-1.41	1	16	125000	0	1	12	125000	0
153600	1	16	156250	1.73	2	7	149643	-2.58	1	13	153846	0.16	1	10	150000	-2.34
250000	1	10	250000	0	1	8	261875	4.75	1	8	250000	0	1	6	250000	0
					0	17	246471	-1.41								

备注 MC0CKS2 ~ MC0CKS0: MCG 控制寄存器 1 (MC0CTL1) 的位 2~0(基时钟 (f<sub>xCLK</sub>)的设置)  
 k: MCG 控制寄存器 2 (MC0CTL2) (k = 4, 5, 6, ..., 31)位 4~0 (MC0BRS4 ~ MC0BRS0)设置的数值  
 f<sub>PRS</sub>: 外围硬件时钟频率  
 ERR: 波特率误差

**(d) 端口模式寄存器 3 (PM3)**

此寄存器由位单元设置端口 3 为输入/输出模式。

当使用 P32/TOH0/MCGO 引脚用作位序列数据输出时, 将 PM32 清零 0, 并且将 P32 的输出锁存设置为 0。

PM3 可以通过 1 位或者 8 位存储操作指令设置。

复位信号将此寄存器设置为 FFH。

地址: FF23H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34	PM33	PM32	PM31	1

PM3n	P3n 引脚 I/O 模式选择 (n = 1 ~ 4)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

(2) 传输操作

位序列缓冲器模式下，数据能够以 1- ~ 8-位为单元进行传输。如果 MCG 控制寄存器 0 (MC0CTL0) 的位 7(MC0PWR)设置为 1，就允许传输操作。

传输挂起时的输出数值通过 MC0CTL0 寄存器的位 0(MC0OLV)设置。

在将传输数据位长度写入 MCG 传输位计数规定寄存器(MC0BIT)之后，通过在 MCG 传输缓冲器寄存器 (MC0TX) 中写入一个数据，传输方能开始 (MC0TX)。在传输开始时，MC0BIT 数据传入 3-位计数器，并且 MC0TX 数据传入 8-位移位寄存器。在 MC0TX 数值传入 8-位移位寄存器时，产生中断请求信号 (INTMCG)。8-位移位寄存器由波特率时钟连续移位,并且从 MCGO 引脚输出。

当执行连续传输时，在数据传输期间，发生 INTMCG 之后，将下一个数据写入 MC0BIT 和 MC0TX。

为了连续传输，在图 16-9 中的(3)和(4)期间必须要将下一个传输数据写入 MC0TX。连续传输期间，在写 MC0TX 之前重写 MC0BIT。

图 16-9. 位序列缓冲器模式的时序(LSB 优先) (1/4)

(1) 传输时序 (MC0OLV = 1, 总的传输位长度 = 8 位)

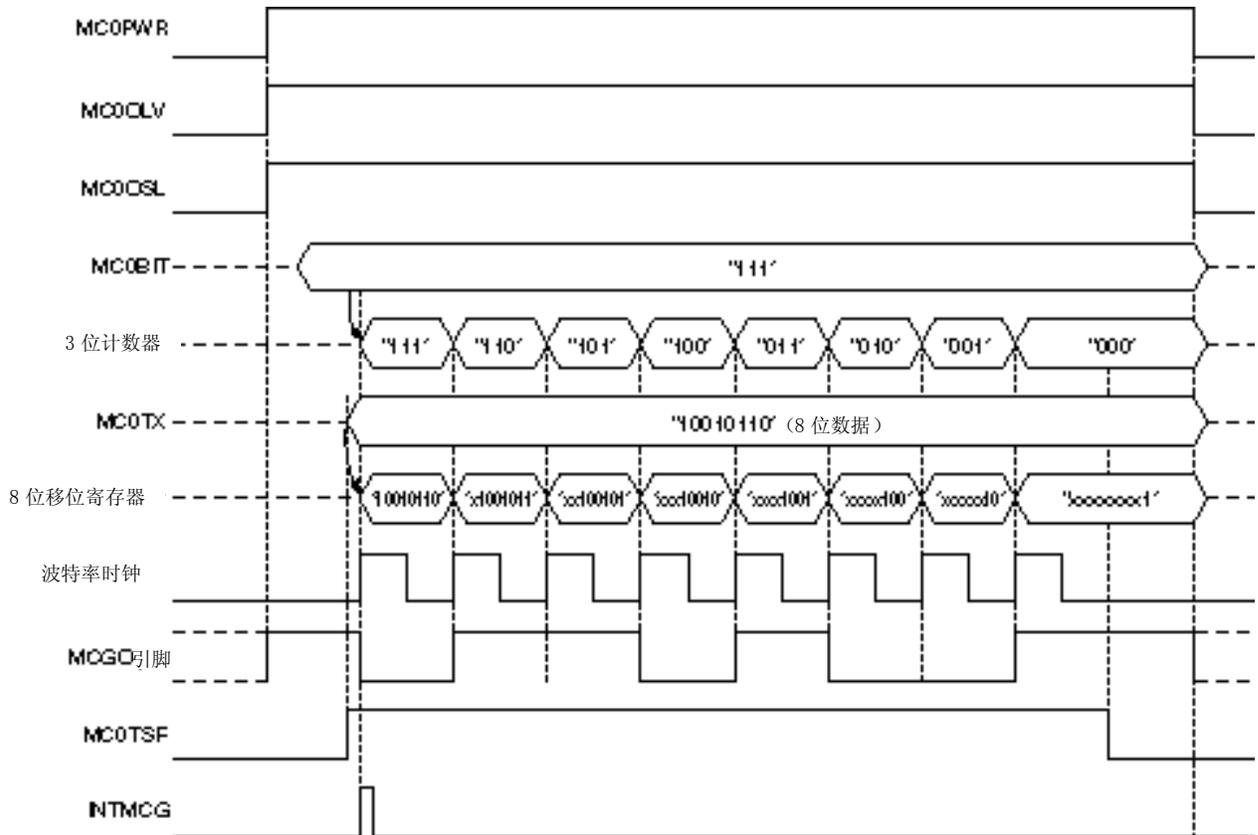


图 16-9. 位序列缓冲器模式的时序(LSB 优先) (2/4)

(2) 传输时序 (MC0OLV = 0, 总的传输位长度 = 8 位)

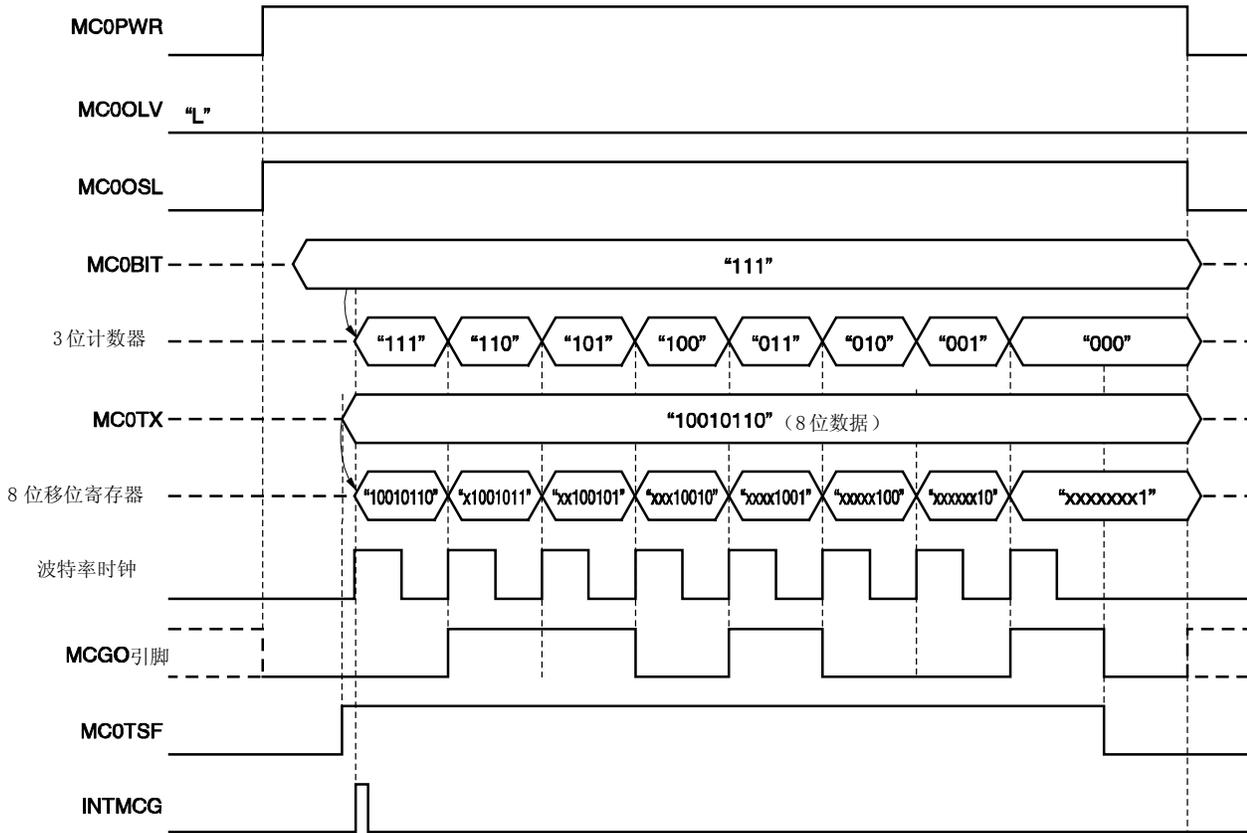
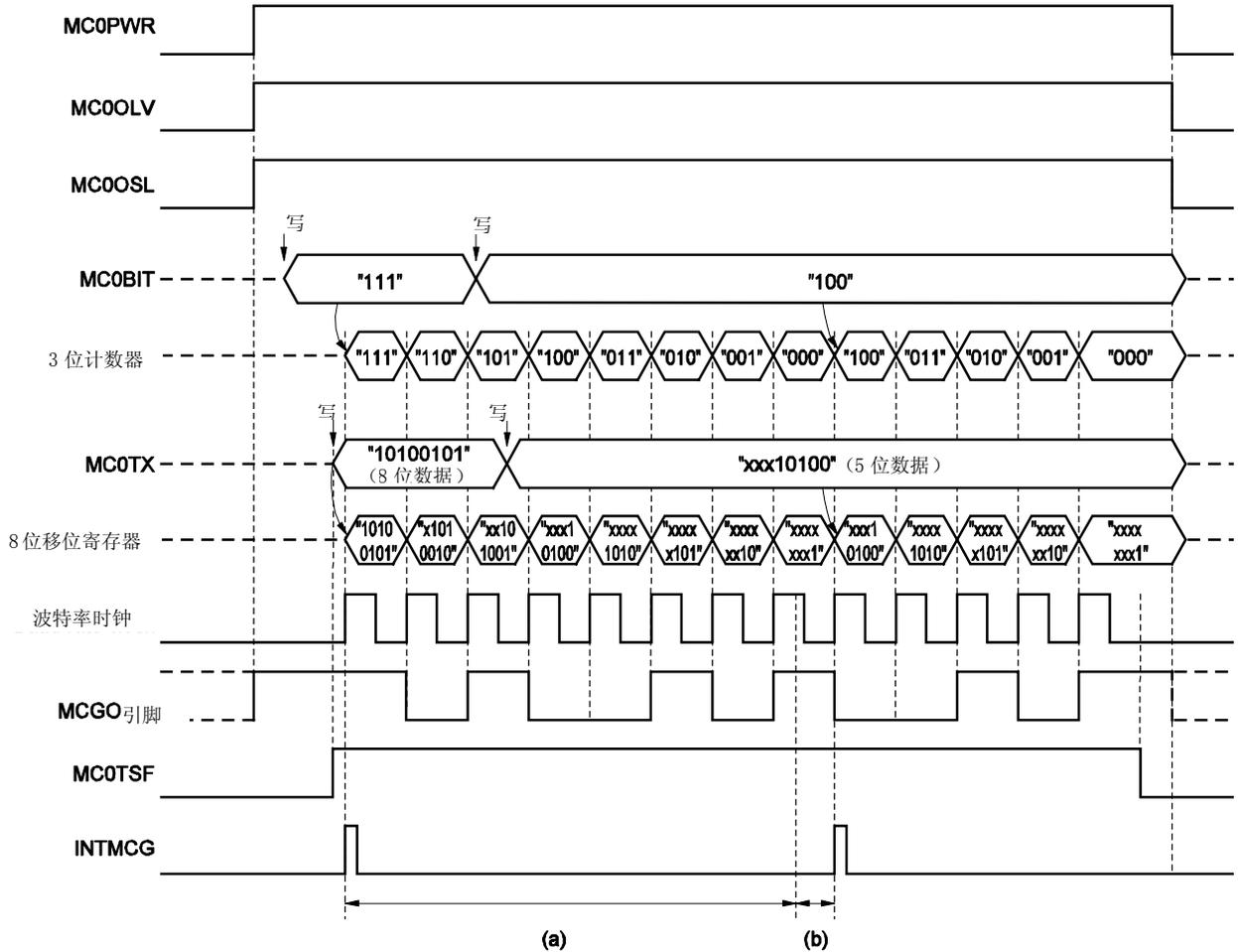


图 16-9. 位序列缓冲器模式的时序(LSB 优先) (3/4)

(3) 传输时序 (MC0OLV = 1, 总的传输位长度= 13 位)



(a): “8-位传输期间” – (b)

(b): “波特率的 1/2 周期” + 传输数据的最后一位之前的 1 时钟 (fxCLK)

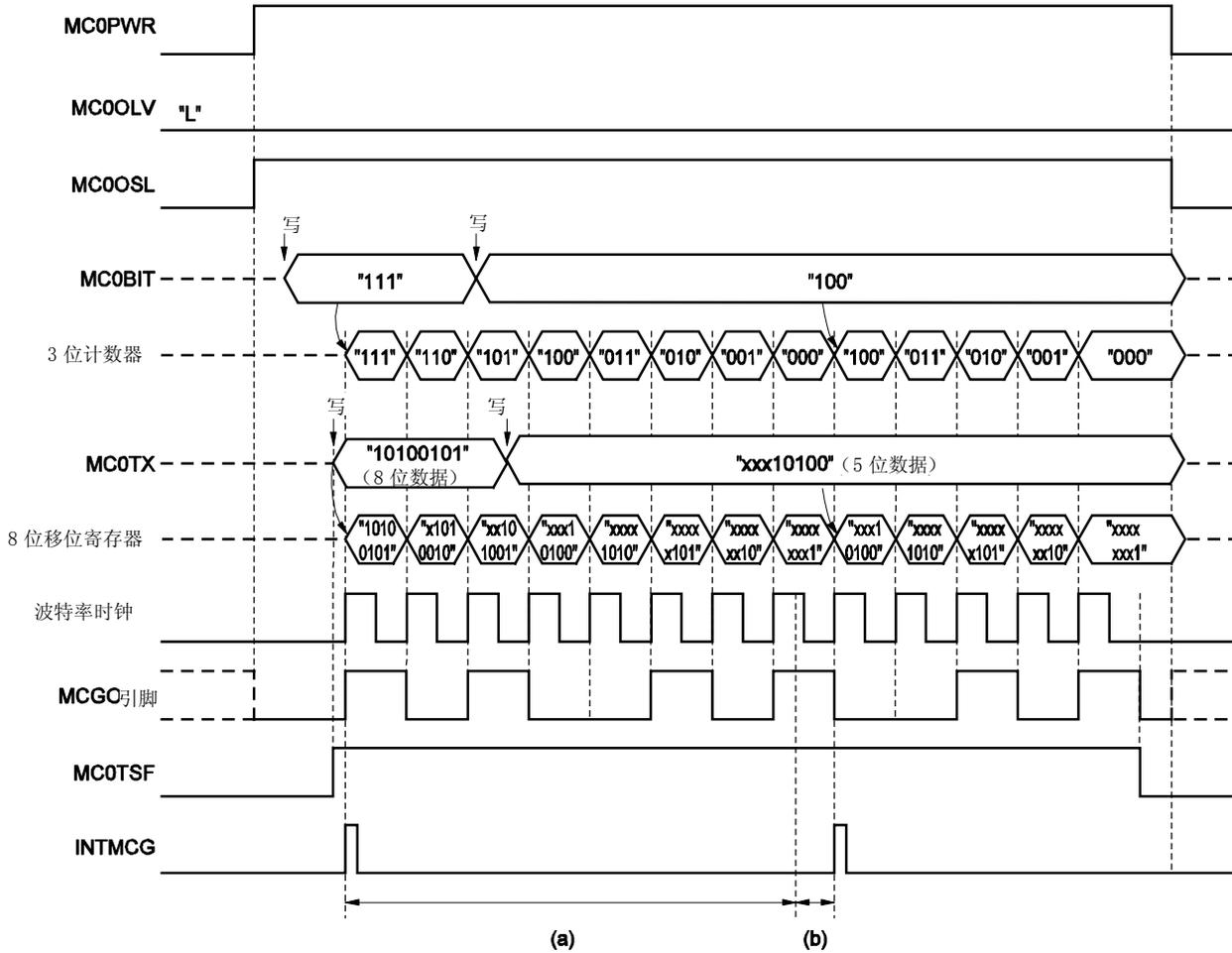
fxCLK: 使用 MC0CTL1 寄存器的 MC0CKS2 ~ MC0CKS0 位选择的操作基时钟频率

最后一位: 当 3-位计数器 = 000 时的传输位

**注意事项** 连续传输期间在时段(a)期间必须将下一个传输数据写入 MC0TX。如果在时段(b)期间写入，下一个数据传输在最后一位传输完成之后两个时钟(fxCLK)开始。  
连续传输期间，在写 MC0TX 之前重写 MC0BIT。

图 16-9. 位序列缓冲器模式的时序(LSB 优先) (4/4)

(4) 传输时序(MC0OLV = 0, 总的传输位长度= 13 位)



(a): “8-位传输期间” – (b)

(b): “波特率的 1/2 周期” + 传输数据的最后一位之前的 1 时钟 (f<sub>XCLK</sub>)

f<sub>XCLK</sub>: 使用 MC0CTL1 寄存器的 MC0CKS2 ~ MC0CKS0 位选择的操作基时钟频率

最后一位: 当 3-位计数器 = 000 时的传输位

**注意事项** 连续传输期间在时段(a)期间必须将下一个传输数据写入 MC0TX。如果在时段(b)期间写入，下一个数据传输在最后一位传输完成之后两个时钟(f<sub>XCLK</sub>)开始。  
连续传输期间，在写 MC0TX 之前重写 MC0BIT。

### 17.1 中断功能的类型

可分为以下两种类型。

#### (1) 可屏蔽中断

这类中断可进行屏蔽控制。通过设置优先级指定标志寄存器(PR0L, PR0H, PR1L, PR1H)将可屏蔽中断分为高优先级中断组和低优先级中断组。

支持中断嵌套，即在低优先级的中断服务子程序中可以响应高优先级的中断请求。如果同时产生两个或两个以上具有相同优先级的中断请求，则根据向量中断服务的优先级进行处理。优先级顺序，请参见表 17-1。产生一个退出待机信号，并退出 STOP 和 HALT 模式。

可屏蔽中断包括外部中断请求和内部中断请求。

- $\mu$ PD78F040x  
外部: 5, 内部: 17
- $\mu$ PD78F041x  
外部: 5, 内部: 18

#### (2) 软件中断

这是通过执行 BRK 指令产生的一类向量中断。即使禁止中断时也可以响应这类中断。软件中断不受中断优先级控制。

### 17.2 中断源及配置

$\mu$ PD78F040x 共有 23 种中断源， $\mu$ PD78F041x 共有 24 种中断源，包括可屏蔽的中断和软件中断。另外，还具有 4 种复位源(参见表 17-1)。

表 17-1. 中断源列表(1/2)

中断类型	默认 优先级 <sup>注1</sup>	中断源		内部/ 外部	向量表 地址	基本配置 类型 <sup>注2</sup>
		名称	触发器			
可屏蔽的	0	INTLVI	低电压检测 <sup>注3</sup>	内部	0004H	(A)
	1	INTP0	引脚输入脉冲沿检测	外部	0006H	(B)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTSRE6	UART6 产生接收错误	内部	0012H	(A)
	6	INTSR6	UART6 接收结束		0014H	
	7	INTST6	UART6 发送结束		0016H	
	8	INTST0	UART0 通信结束		0018H	
	9	INTTMH1	TMH1 与 CMP01 相等 (指定比较寄存器)		001AH	
	10	INTTMH0	TMH0 与 CMP00 相等 (指定比较寄存器)		001CH	
	11	INTTM50	TM50 与 CR50 相等 (指定比较寄存器)		001EH	
	12	INTTM000	TM00 与 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)		0020H	
	13	INTTM010	TM00 与 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)		0022H	
	14	INTAD <sup>注5</sup>	10-位逐次逼近型 A/D 转换结束		0024H	
	15	INTSR0	UART0 接收结束或产生接收错误		0026H	
	16	INTRTC	实时计数器/闹表匹配信号的固定周期检测		0028H	
	17	INTTM51 <sup>注4</sup>	TM51 与 CR51 相等 (指定比较寄存器)		002AH	
	18	INTKR	案件中断检测	外部	002CH	(C)
19	INTRTCI	实时计数器的间隔信号检测	内部	002EH	(A)	

- 注
1. 当两个或两个以上可屏蔽中断同时产生时，采用默认的优先级来决定向量中断的处理顺序。优先级 0 的优先级别最高，而优先级 22 的优先级别最低。
  2. 基本配置类型 (A) ~ (D) 与图 17-1 中 (A) ~ (D) 相对应。
  3. 当低电压检测寄存器(LVIM)的第 1 位(LVIMD)被清零时。
  4. 当 8-位 定时器/事件计数器 51 和 8-位 计数器 H1 用于载波发生器模式时，INTTM5H1 信号发生后产生中断(请看 图 8-15 转移时序)。
  5.  $\mu$ PD78F041x：只有此型号才有。

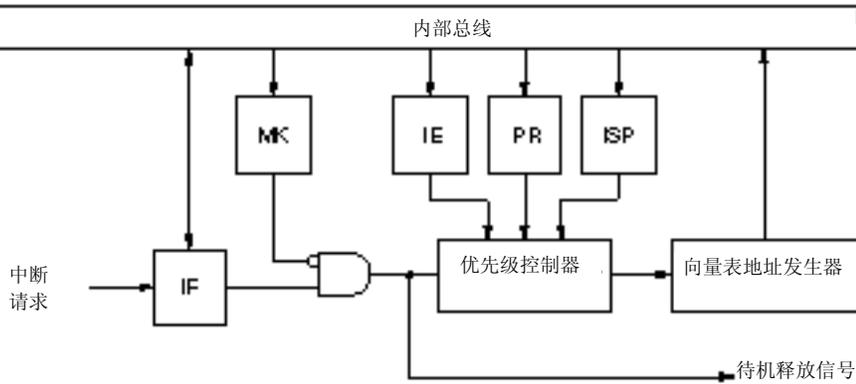
表 17-1. 中断源列表(2/2)

中断类型	默认 优先级 <sup>注1</sup>	中断源		内部/ 外部	向量表 地址	基本配置 类型 <sup>注2</sup>
		名称	触发器			
可屏蔽的	20	INTTM52	TM52 与 CR52 相等 (指定比较寄存器)	内部	0032H	(A)
	21	INTTMH2	TMH2 与 CRH2 相等 (指定比较寄存器)		0034H	
	22	INTMCG	曼彻斯特码接收结束		0036H	
软件中断	-	BRK	执行 BRK 指令	-	003EH	(D)
复位中断	-	RESET	复位输入	-	0000H	-
		POC	上电清零			
		LVI	低电压检测 <sup>注3</sup>			
		WDT	WDT 溢出			

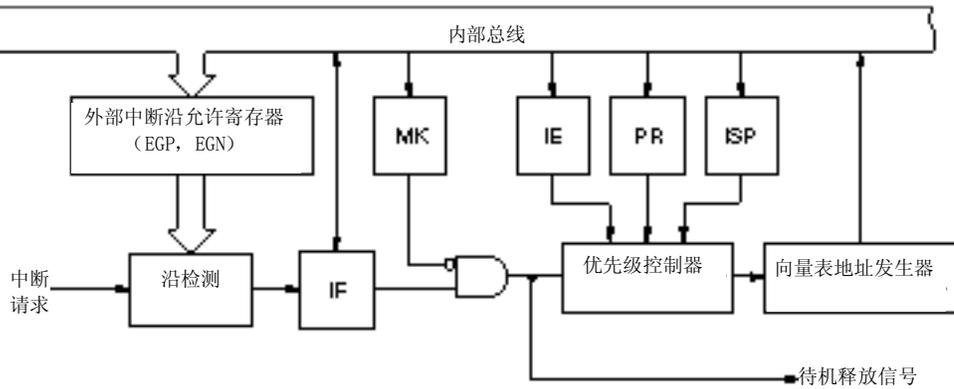
- 注
1. 当两个或两个以上可屏蔽中断同时产生时，采用默认的优先级来决定向量中断的处理顺序。优先级 0 的优先级别最高，而优先级 22 的优先级别最低。
  2. 基本配置类型 (A) ~ (D) 与图 17-1 中 (A) ~ (D) 相对应。
  3. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 设置为 1 时。

图 17-1. 中断功能的基本配置(1/2)

(A) 内部可屏蔽中断



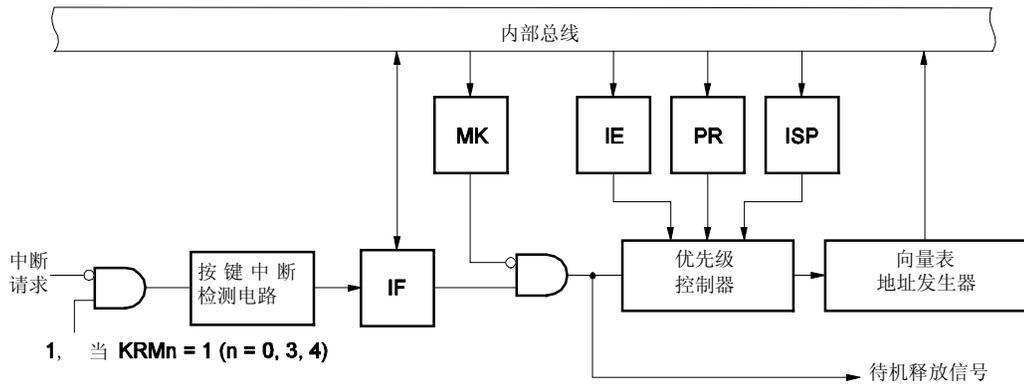
(B) 外部可屏蔽中断 (INTP0 ~ INTP3)



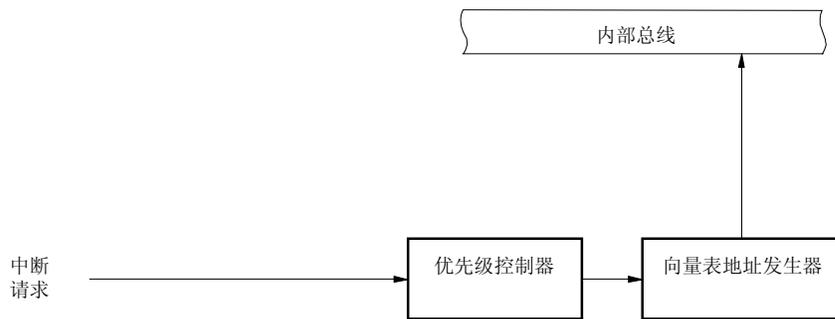
- IF: 中断请求标志
- IE: 中断允许标志
- ISP: 服务优先级标志
- MK: 中断屏蔽标志
- PR: 优先级指定标志

图 17-1. 中断功能的基本配置(2/2)

(C) 外部可屏蔽中断(INTKR)



(D) 软件中断



- IF: 中断请求标志
- IE: 中断允许标志
- ISP: 服务优先级标志
- MK: 中断屏蔽标志
- PR: 优先级指定标志
- KRM: 按键返回模式寄存器

## 17.3 控制中断功能的寄存器

以下 6 种寄存器用于控制中断功能。

- 中断请求标志寄存器 (IF0L、IF0H、IF1L、IF1H)
- 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L、MK1H)
- 优先级指定标志寄存器 (PR0L、PR0H、PR1L、PR1H)
- 外部中断上升沿允许寄存器 (EGP)
- 外部中断下降沿允许寄存器 (EGN)
- 程序状态字 (PSW)

表 17-2 列出了与中断请求源相对应的中断请求标志、中断屏蔽标志和优先级指定标志。

表 17-2. 与中断请求源相对应的标志

中断源	中断请求标志		中断屏蔽标志		优先级指定标志	
		寄存器				寄存器
INTLVI	LVIF	IF0L	LVIMK	MK0L	LVIPR	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTSRE6	SREIF6		SREMK6		SREPR6	
INTSR6	SRIF6	IF0H	SRMK6	MK0H	SRPR6	PR0H
INTST6	STIF6		STMK6		STPR6	
INTST0	STIF0		STMK0		STPR0	
INTTMH1	TMIFH1		TMMKH1		TMPRH1	
INTTMH0	TMIFH0		TMMKH0		TMPRH0	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM000	TMIF000		TMMK000		TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010	
INTAD <sup>※1</sup>	ADIF <sup>※1</sup>	IF1L	ADMK <sup>※1</sup>	MK1L	ADPR <sup>※1</sup>	PR1L
INTSR0	SRIF0		SRMK0		SRPR0	
INTRTC	RTCIF		RTCMK		RTCPR	
INTTM51 <sup>※2</sup>	TMIF51		TMMK51		TMPR51	
INTKR	KRIF		KRMK		KRPR	
INTRTCI	RTCIF		RTCIMK		RTCIPR	
INTTM52	TMIF52		TMMK52		TMPR52	
INTTMH2	TMHIF2	IF1H	TMHMK2	MK1H	TMHPR2	PR1H
INTMCG	MCGIF		MCGMK		MCGPR	

- 注
1.  $\mu$ PD78F041x 只有此型号才有。
  2. 当 8-位 定时器/事件计数器 51 和 8-位 计数器 H1 用于载波发生器模式时，INTTM5H1 信号发生后产生中断 (请看 图 8-15 转移时序)。

**(1) 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H)**

当产生相关的中断请求或执行指令时，这些中断请求标志被置 1。当执行的指令是响应中断请求或复位输入时，这些标志被清零。

当响应中断时，中断请求标志自动清零，然后进入中断服务程序。

可由 1 位或 8 位存储器操作指令设置 IF0L、IF0H、IF1L 和 IF1H。当 IF0L 与 IF0H、IF1L 与 IF1H 组合起来形成 16 位寄存器 IF0 和 IF1 时，可由 16 位存储器操作指令设置这些寄存器。

复位信号的输入将这些寄存器清零(00H)。

图 17-2. 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H) 的格式

地址: FFE0H 复位后: 00H R/W

符号	<7>	6	5	<4>	<3>	<2>	<1>	<0>
IF0L	SREIF6	0	0	PIF3	PIF2	PIF1	PIF0	LVIF

地址: FFE1H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	TMIF010	TMIF000	TMIF50	TMIFH0	TMIFH1	STIF0	STIF6	SRIF6

地址: FFE2H 复位后: 00H R/W

符号	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF52	0	RTCIF	KRIF	TMIF51	RTCIIF	SRIF0	ADIF <sup>#</sup>

地址: FFE3H 复位后: 00H R/W

符号	7	6	5	4	3	2	<1>	<0>
IF1H	0	0	0	0	0	0	MCGIF	TMHIF2

XXIFX	中断请求标志
0	没有产生中断请求信号
1	产生中断请求，中断请求状态

注  $\mu$ PD78F041x 只有此型号才有。

- 注意事项**
1. 必须对 IF0L 的位 5 和位 6, IF1L 的位 6, 和 IF1H 的位 2~位 7 清零。
  2. 当退出待机模式后并要使用定时器、串行接口或 A/D 转换器时，先将中断请求标志清零后再对这些部件进行操作。噪音可能会设置中断请求标志。

**注意事项 3.** 当使用中断请求标志寄存器中的标志时，使用一位存储器操作指令 (CLR1)。当用 C 语言描述时，由于编译后的汇编指令必须是一位存储器操作指令 (CLR1)，所以应该使用一位操作指令，如 “IF0L.0 = 0;” 或 “\_asm(“clr1 IF0L, 0”);”。

如果一条用 C 语言描述的 8 位存储器操作指令如 “IF0L &= 0xfe;”，则编译后将被转换为三条汇编指令：

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

这种情况下，当处于 “mov a, IF0L” 和 “mov IF0L, a” 之间的时序时，即使同一中断请求标志寄存器 (IF0L) 的其他请求标志被设置为 1，则该请求标志也将被 “mov IF0L, a” 清零。因此在 C 语言中使用 8 位存储器操作指令时必须小心。

**(2) 中断屏蔽标志寄存器 (MK0L, MK0H, MK1L, MK1H)**

这些中断屏蔽标志用于允许/禁止相关的可屏蔽中断服务。

可由 1 位或 8 位存储器操作指令设置 MK0L、MK0H、MK1L 和 MK1H。当 MK0L 与 MK0H、MK1L 与 MK1H 组合起来形成 16 位寄存器 MK0 与 MK1 时，可由 16 位存储器操作指令设置这些寄存器。

复位信号的产生将这些寄存器的内容设置为 FFH。

**图 17-3. 中断屏蔽标志寄存器的格式 (MK0L, MK0H, MK1L, MK1H)**

地址: FFE4H 复位后: FFH R/W

符号	<7>	6	5	<4>	<3>	<2>	<1>	<0>
MK0L	SREMK6	1	1	PMK3	PMK2	PMK1	PMK0	LVIMK

地址: FFE5H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	TMMK010	TMMK000	TMMK50	TMMKH0	TMMKH1	STMK0	STMK6	SRMK6

地址: FFE6H 复位后: FFH R/W

符号	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK52	1	RTCMK	KRMK	TMMK51	RTCIMK	SRMK0	ADMK <sup>Ⓢ</sup>

地址: FFE7H 复位后: FFH R/W

符号	7	6	5	4	3	2	<1>	<0>
MK1H	1	1	1	1	1	1	MCGMK	TMHMK2

XXMKX	中断服务控制							
0	允许中断服务							
1	禁止中断服务							

注  $\mu$ PD78F041x 只有此型号才有。

**注意事项** 必需将 MK0L 的位 5 和位 6, MK1L 的位 6, MK1H 的位 2~7 设置为 1。

**(3) 优先级指定标志寄存器(PR0L, PR0H, PR1L, PR1H)**

这些优先级指定标志寄存器用于设置相关的可屏蔽中断优先级次序。

可由 1 位或 8 位存储器操作指令设置 PR0L、PR0H、PR1L 和 PR1H。当 PR0L 与 PR0H、PR1L 与 PR1H 组合起来形成 16 位寄存器 PR0 与 PR1 时，可由 16 位存储器操作指令设置这些寄存器。

复位信号的产生将这些寄存器内容设置为 FFH。

图 17-4. 优先级指定标志寄存器 (PR0L, PR0H, PR1L, PR1H) 的格式

地址: FFE8H 复位后: FFH R/W

符号	<7>	6	5	<4>	<3>	<2>	<1>	<0>
PR0L	SREPR6	1	1	PPR3	PPR2	PPR1	PPR0	LVIPR

地址: FFE9H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0H	TMPR010	TMPR000	TMPR50	TMPRH0	TMPRH1	STPR0	STPR6	SRPR6

地址: FFEAH 复位后: FFH R/W

符号	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PR1L	TMPR52	1	RTCPR	KRPR	TMPR51	RTCIPR	SRPR0	ADPR <sup>Ⓜ</sup>

地址: FFE BH 复位后: FFH R/W

符号	7	6	5	4	3	2	<1>	<0>
PR1H	1	1	1	1	1	1	MCGPR	TMHPR2

XXPRX	优先级选择
0	高优先级
1	低优先级

注  $\mu$ PD78F041x 只有此型号才有。

注意事项 一定要将 PR0L 的位 5 和位 6, PR1L 的位 6, 和 PR1H 的位 2~7 设置为 1。

**(4) 外部中断上升沿允许寄存器(EGP)、外部中断下降沿允许寄存器(EGN)**

这两个寄存器用于指定 INTP0~INTP3 的有效沿。

可由 1 位或 8 位存储器操作指令设置 EGP 和 EGN。

复位信号的产生将这些寄存器清零 (00H)。

**图 17-5. 外部中断上升沿允许寄存器(EGP)和外部中断下降沿允许寄存器(EGN)的格式**

地址: FF48H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGP	0	0	0	0	EGP3	EGP2	EGP1	EGP0

地址: FF49H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGN	0	0	0	0	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn 引脚有效沿选择 (n = 0 ~ 3)
0	0	禁止脉冲沿检测
0	1	下降沿
1	0	上升沿
1	1	兼有上升沿和下降沿

表 17-3 显示了与 EGPn 和 EGNn 相关的端口。

**表 17-3. 与 EGPn 和 EGNn 相关的端口**

检测允许寄存器		脉冲沿检测端口	中断请求信号
EGP0	EGN0	P120/EXLVI	INTP0
EGP1	EGN1	P34/TI52/TI010/TO00/RTC1HZ	INTP1
EGP2	EGN2	P33/TI000/RTCDIV/RTCCL/BUZ	INTP2
EGP3	EGN3	P31/TOH1	INTP3

**注意事项** 在从外部中断功能切换到端口功能时可能会检测到脉冲沿，因此通过将 EGPn 和 EGNn 清零可选择端口模式。

**备注** n = 0 ~ 3

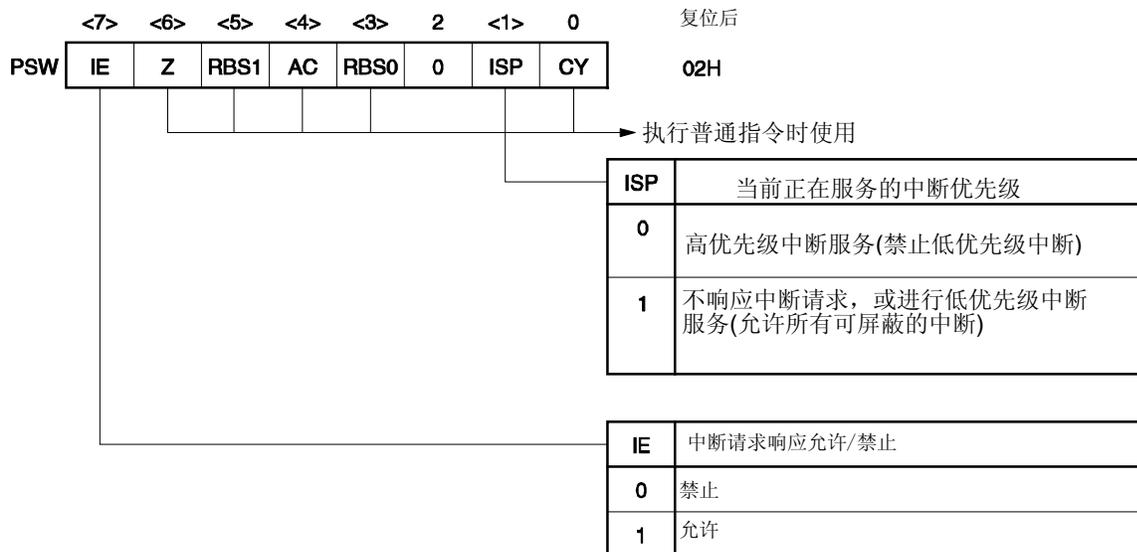
(5) 程序状态字(PSW)

PSW 是用于保存指令执行结果和中断请求的当前状态的寄存器。在 PSW 中包含 IE 标志（设置允许/禁止可屏蔽中断）和 ISP 标志（控制中断嵌套）。

除了 8 位读/写操作指令外，还可使用位操作指令和专用指令（EI 和 DI）对该寄存器进行操作。在响应向量中断请求时，如果执行 BRK 指令，则将 PSW 的内容自动保存到堆栈中，并且将 IE 标志复位为 0。如果响应可屏蔽中断请求，则将被响应中断的优先级指定标志的内容转移到 ISP 标志中。执行 PUSH PSW 指令将 PSW 的内容保存到堆栈中。而执行 RETI、RETB 和 POP PSW 指令可将这些内容从堆栈中恢复。

RESET 输入将 PSW 设置为 02H。

图 17-6. 程序状态字的格式



## 17.4 中断服务操作

### 17.4.1 可屏蔽的中断响应

当中断请求标志=1 且与该中断请求相关的屏蔽标志(MK)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(IE =1), 可以响应向量中断请求。但在一个较高优先级中断请求服务期间 (ISP 标志复位为 0), 不响应低优先级的中断请求。从一个可屏蔽中断请求产生到中断服务执行所经历的时间如表 17-4 所示。

中断请求响应时序可参见图 17-8 和 17-9。

表 17-4. 从可屏蔽中断产生到执行中断服务所需要的时间

	最短时间	最长时间 <sup>注</sup>
当 $\times\times PR = 0$	7 个时钟	32 个时钟
当 $\times\times PR = 1$	8 个时钟	33 个时钟

**注** 如果是在除法指令执行之前产生一个中断请求, 则等待时间会更长。

**备注** 1 个时钟:  $1/f_{CPU}$  ( $f_{CPU}$ : CPU 时钟)

如果同时产生两个或两个以上的可屏蔽中断请求, 则首先响应优先级指定标志中优先级别高的请求。如果两个或两个以上的中断请求具有相同的优先级别, 则首先响应具有最高默认优先级的中断请求。

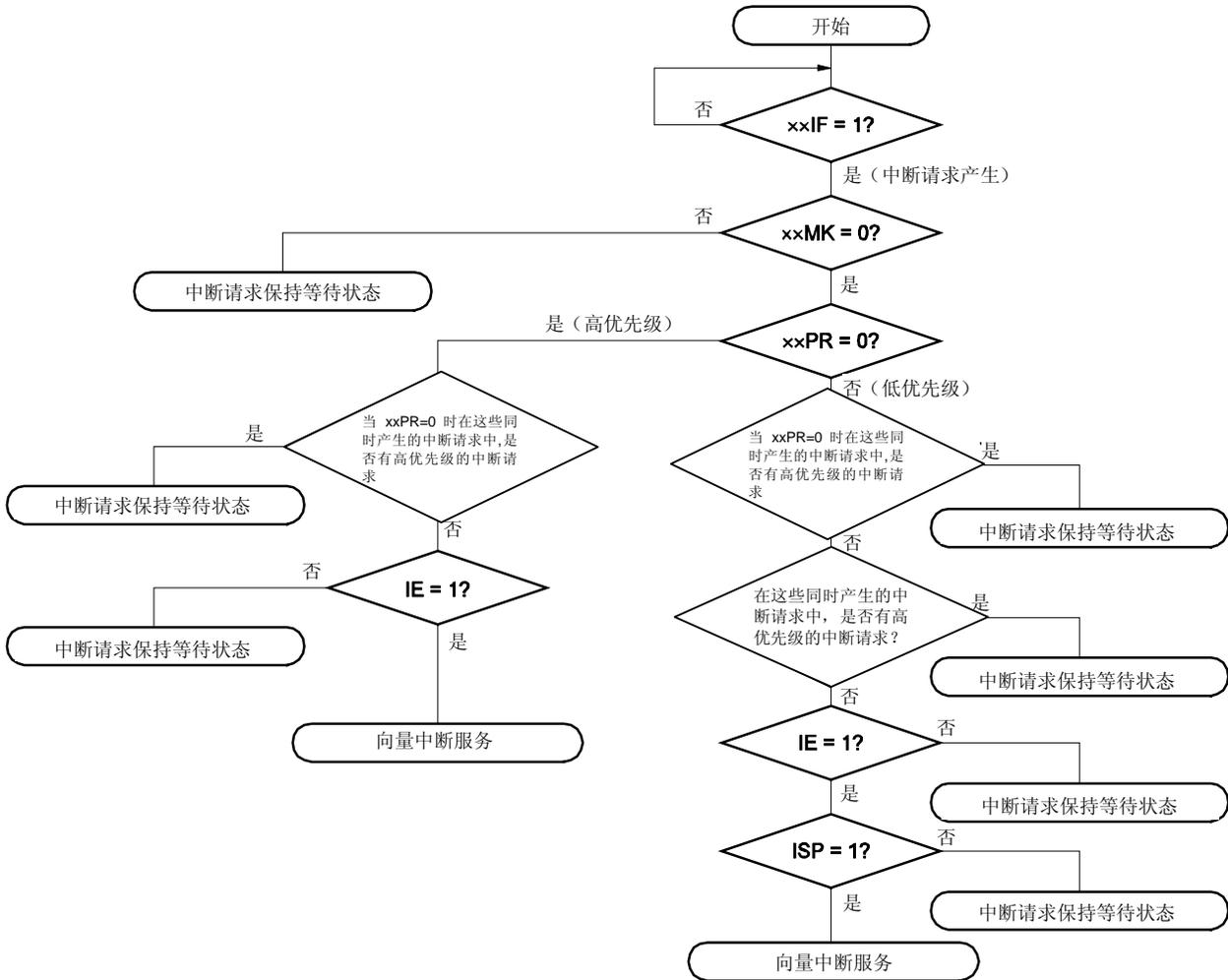
当允许响应中断请求时, 处于等待状态的中断请求被响应。

图 17-7 显示了中断请求响应算法。

如果响应了一个可屏蔽中断请求, 则将 PSW、PC 的内容依次保存到堆栈中, 然后将 IE 标志复位为 0, 并将与被响应的中断相关的优先级指定标志的内容传送到 ISP 中。将用于中断请求的向量表数据传送到 PC 中, 并转移。

可通过执行 RETI 指令从中断返回。

图 17-7. 中断请求响应处理



xxIF: 中断请求标志

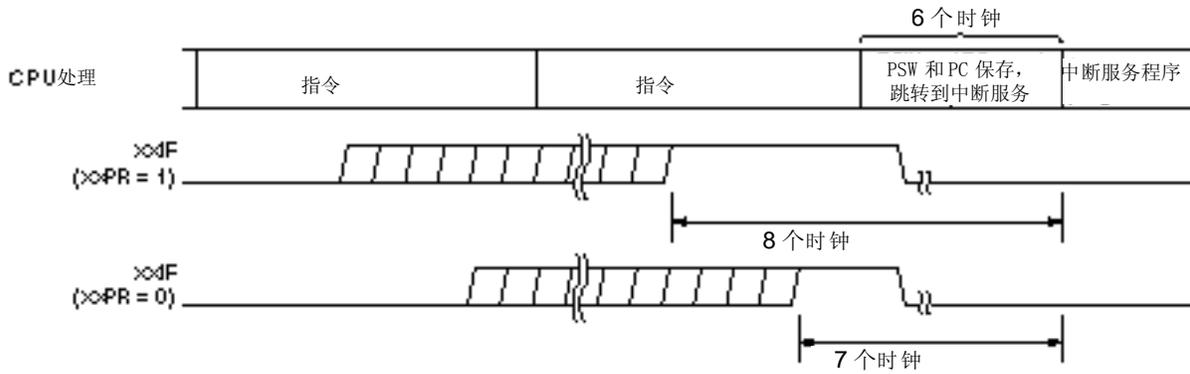
xxMK: 中断屏蔽标志

xxPR: 优先级指定标志

IE: 控制可屏蔽中断请求响应的标志 (1 = 允许、0 = 禁止)

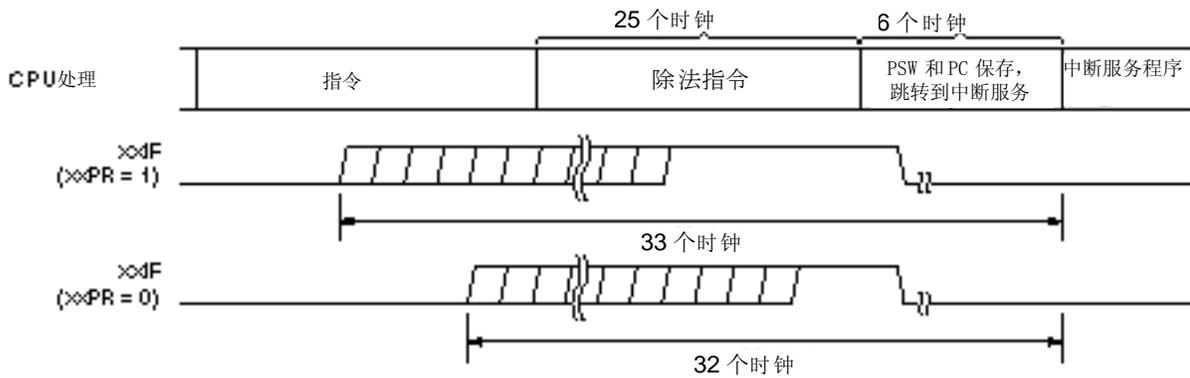
ISP: 指示正在服务的中断优先级别的标志 (0 = 高优先级的中断服务, 1 = 没有中断请求被响应, 或低优先级的中断服务)

图 17-8. 中断请求响应时序 (最短时间)



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

图 17-9. 中断请求响应时序 (最长时间)



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

#### 17.4.2 软件中断请求响应

执行 BRK 指令可响应软件中断。软件中断不能被禁止。

如果响应了一个软件中断请求，则将程序状态字（PSW）和程序计数器（PC）的内容依次保存到堆栈中，然后将 IE 标志复位为 0，并将向量表的内容（003EH、003FH）传送到 PC 中，然后转移。

可通过执行 RETB 指令从软件中断返回。

**注意事项** 不能使用 RETI 指令从软件中断返回。

### 17.4.3 中断嵌套

在执行一个中断服务程序时，又响应了其他中断，这时就产生了中断嵌套。

除非选择允许中断请求响应状态（ $IE = 1$ ），否则不会产生中断嵌套。在响应一个中断请求时，禁止响应其它中断请求（ $IE = 0$ ）。因此，如果要允许中断嵌套，必须在中断服务期间执行 EI 指令，将 IE 标志置 1，从而允许响应其它中断请求。

此外，即使允许中断，也不一定允许中断嵌套，这是因为受到中断优先级控制的限制。可使用两类优先级控制方式：默认优先级控制和可编程优先级控制。可编程优先级控制用于中断嵌套。

在中断允许状态中，如果产生的中断请求的优先级与正在服务的中断优先级相等或高于它，则响应该中断请求，从而产生中断嵌套。如果产生的中断请求的优先级低于正在服务的中断优先级，则不响应该中断请求。由于禁止中断或中断请求的优先级别较低，这些不被允许的中断请求处于等待状态。若当前的中断服务已结束，则在执行至少一条主程序指令后才可响应处于等待状态的中断请求。

表 17-5 显示了允许中断嵌套的中断请求之间的关系，图 17-10 为中断嵌套示例。

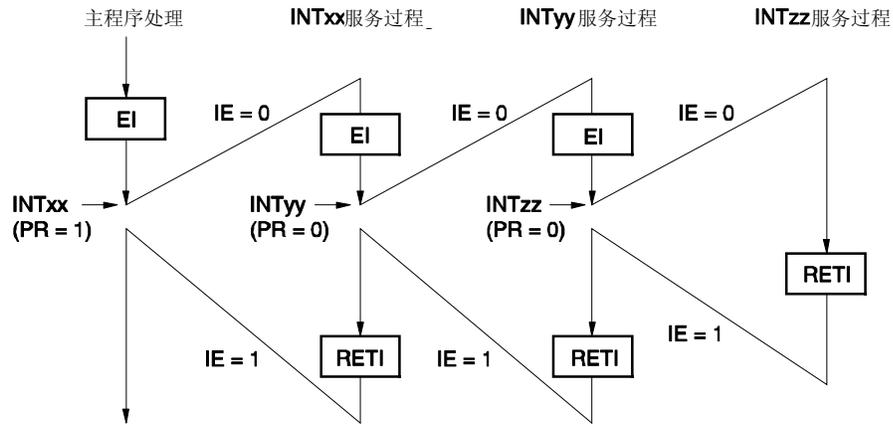
表 17-5. 中断服务期间允许进行中断嵌套的中断请求之间的关系

中断嵌套		可屏蔽中断请求				软件中断请求
		PR = 0		PR = 1		
		IE = 1	IE = 0	IE = 1	IE = 0	
正在服务的中断						
可屏蔽的中断	ISP = 0	○	×	×	×	○
	ISP = 1	○	×	○	×	○
软件中断		○	×	○	×	○

- 备注**
- ：允许中断嵌套
  - ×：禁止中断嵌套
  - ISP 和 IE 为 PSW 中的标志位。  
ISP = 0：正在服务一个高优先级中断。  
ISP = 1：无中断请求被响应，或正在服务一个低优先级中断。  
IE = 0：禁止响应中断请求。  
IE = 1：允许响应中断请求。
  - PR 为 PR0L、PR0H、PR1L 和 PR1H 中的标志位。  
PR = 0：高优先级  
PR = 1：低优先级

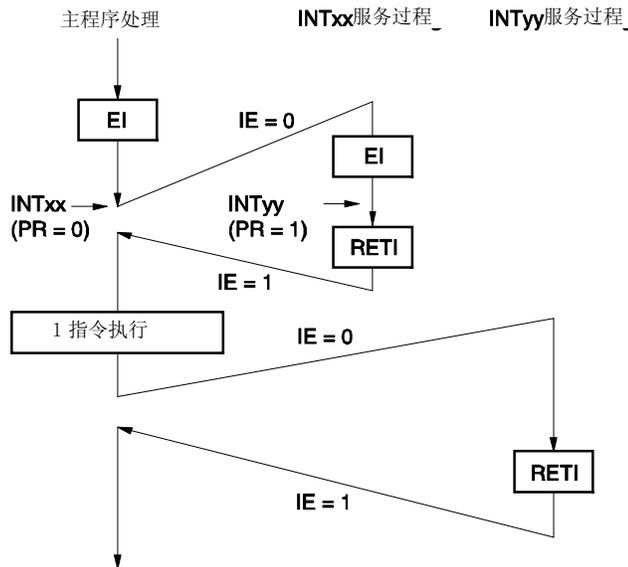
图 17-10. 中断嵌套示例 (1/2)

## 例 1. 产生两次中断嵌套



在进行 INTxx 中断服务期间，响应了两个中断请求 INTyy 和 INTzz，这时就产生了中断嵌套。在响应每个中断之前，必须先执行 EI 指令才能允许响应中断请求。

## 例 2. 由于优先级控制没有产生中断嵌套



在 INTxx 中断服务期间不响应中断请求 INTyy，因为 INTyy 的优先级低于 INTxx，这样就不会产生中断嵌套。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。

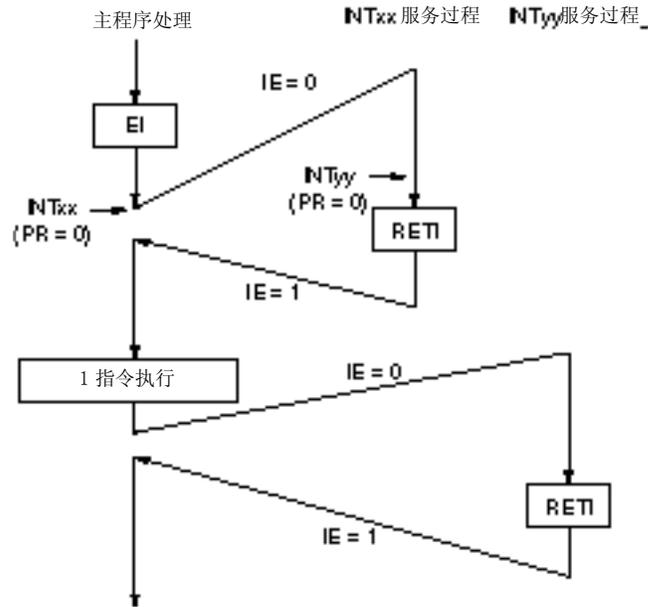
PR = 0: 高优先级

PR = 1: 低优先级

IE = 0: 禁止响应中断请求

图 17-10. 中断嵌套示例 (2/2)

## 例 3. 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。

PR = 0: 高优先级

IE = 0: 禁止响应中断请求

#### 17.4.4 保持中断请求

在某些指令执行期间，即使出现中断请求，请求响应也要保持等待状态，直到下一条指令执行结束。以下列出这类指令（中断请求保持指令）。

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW. bit, CY
- MOV1 CY, PSW. bit
- AND1 CY, PSW. bit
- OR1 CY, PSW. bit
- XOR1 CY, PSW. bit
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW. bit, \$addr16
- BF PSW. bit, \$addr16
- BTCLR PSW. bit, \$addr16
- EI
- DI
- 用于 IF0L, IF0H, IF1L, IF1H, MK0L, MK0H, MK1L, MK1H, PR0L, PR0H, PR1L, 和 PR1H 寄存器的操作指令。

**注意事项** **BRK** 指令不属于上述列出的中断请求保持指令。但通过执行 **BRK** 指令激活的软件中断会将 **IE** 标志清零。因此，即使在执行 **BRK** 指令期间产生可屏蔽中断请求，该中断请求也不会被响应。

图 17-11 显示了处于等待状态的中断请求时序。

图 17-11. 中断请求保持



- 备注**
1. 指令 N: 中断请求保持指令
  2. 指令 M: 除中断请求保持指令之外的指令
  3.  $\times\times PR$ （优先级）的值不会影响  $\times\times IF$ （中断请求）的操作。

## 第十八章 按键中断功能

### 18.1 按键中断的功能

可通过设置按键返回模式寄存器(KRM)和向按键中断输入引脚(KR0, KR3, 和 KR4)输入一个下降沿产生按键中断(INTKR)。

表 18-1. 按键中断检测引脚的分配

标志	描述
KRM0	以 1 位单元控制 KR0 信号。
KRM3	以 1 位单元控制 KR3 信号。
KRM4	以 1 位单元控制 KR4 信号。

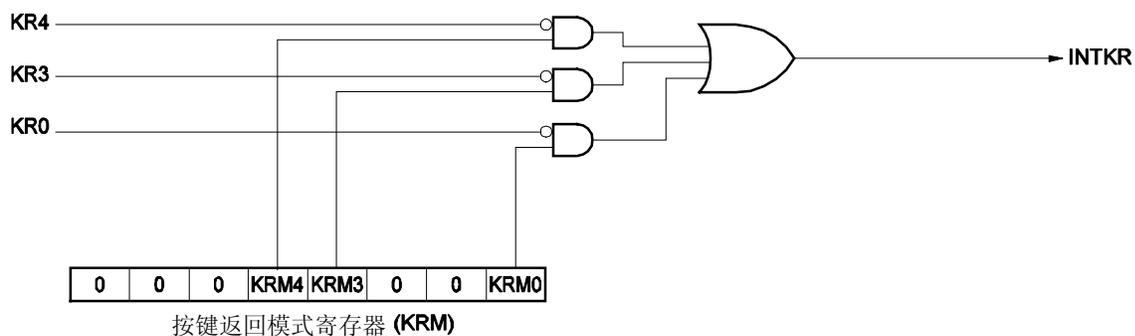
### 18.2 按键中断的配置

按键中断包括以下硬件。

表 18-2 按键中断的配置

项目	配置
控制寄存器	按键返回模式寄存器(KRM)

图 18-1. 按键中断的框图



## 18.3 控制按键中断的寄存器

## (1) 按键返回模式寄存器(KRM)

该寄存器使用 KR0, KR3, 和 KR4 信号分别控制 KRM0, KRM3, 和 KRM4 位。

KRM 可由 1 位或 8 位存储器操作指令设置。

复位信号的产生将 KRM 清零(00H)。

图 18-2. 按键返回模式寄存器 (KRM) 的格式

地址: FF6EH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
KRM	0	0	0	KRM4	KRM3	0	0	KRM0

KRMn	按键中断模式控制 (n = 0, 3, 或 4)
0	不检测按键中断信号
1	检测按键中断信号

- 注意事项**
1. 如果 KRM0, KRM3, 或 KRM4 中任何一位被设置为 1, 则将上拉电阻寄存器 4(PU4)的第 0 位 (PU40), 或者上拉电阻寄存器 1(PU1)的相应位 2 或 3(PU12 或 PU13)设置为 1。
  2. 如果 KRM 发生变化, 则中断请求标志可能被设置。因此可以先禁止中断, 再修改 KRM。先将中断请求标志清零, 再允许中断。
  3. 在按键中断模式中没有被使用的位可用作通用端口。
  4. 当把 P40/KR0/V<sub>LC3</sub> 引脚用作按键中断功能(KR0)时, 将 LCD 显示模式寄存器(LCDM)设置为除 1/4 偏压方式之外的方式。当设置为 1/4 偏压方式时, 它作为 V<sub>LC3</sub> 使用。

## 第十九章 待机功能

### 19.1 待机功能及配置

#### 19.1.1 待机功能

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。如果设置 HALT 模式前，高速系统时钟振荡器、内部高速振荡器、内部低速振荡器或副系统时钟振荡器正在使用，则设置后每种时钟的振荡继续。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频繁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，高速系统时钟振荡器和内部高速振荡器停止操作，整个系统的操作停止，这样 CPU 的工作电流将会大幅下降。

可通过中断请求释放该模式，这样，被中断的操作可以继续执行。由于选择 X1 时钟时，在释放 STOP 模式后需要一段等待时间以确保振荡器振荡稳定，因此如果需要在产生中断请求后立即进行处理，则应选择 HALT 模式。

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 端口输出锁存器和输出缓冲器的状态也将被保持。

- 注意事项**
1. 仅当 CPU 使用主系统时钟时，才能使用 STOP 模式。不能停止副系统时钟的振荡。而当 CPU 使用主系统时钟或副系统时钟时，均可以使用 HALT 模式。
  2. 当切换到 STOP 模式时，在执行 STOP 指令前必须停止使用主系统时钟的外围硬件的操作。
  3. 当使用待机功能时，建议采用以下步骤降低 A/D 转换器的操作电流：首先将 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 和第 0 位 (ADCE) 清零以停止 A/D 转换操作，然后执行 STOP 指令。

### 19.1.2 控制待机功能的寄存器

待机功能由以下两个寄存器控制。

- 振荡稳定时间计数器的状态寄存器（OSTC）
- 振荡稳定时间选择寄存器（OSTS）

**备注** 有关这些寄存器启动、停止以及时钟选择的情况，可参见 **第五章 时钟发生器**。

#### (1) 振荡稳定时间计数器的状态寄存器 (OSTC)

这是 X1 时钟振荡稳定时间计数器的状态寄存器。如果 CPU 使用 X1 振荡时钟、内部高速振荡时钟或副系统时钟，则可以检测 X1 时钟振荡稳定时间。

可由 1 位或 8 位存储器操作指令读取 OSTC。

复位释放（通过  $\overline{\text{RESET}}$  输入、POC、LVI 和 WDT）、STOP 指令以及 MSTOP（MOC 寄存器的第 7 位）= 1 可以将 OSTC 清零（00H）。

图 19-1. 振荡稳定时间计数器的状态寄存器（OSTC）的格式

地址: FFA3H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	振荡稳定时间的状态	
					$f_x = 10 \text{ MHz}$	
1	0	0	0	0	$2^{11}/f_x \text{ min.}$	204.8 $\mu\text{s min.}$
1	1	0	0	0	$2^{13}/f_x \text{ min.}$	819.2 $\mu\text{s min.}$
1	1	1	0	0	$2^{14}/f_x \text{ min.}$	1.64 ms min.
1	1	1	1	0	$2^{15}/f_x \text{ min.}$	3.27 ms min.
1	1	1	1	1	$2^{16}/f_x \text{ min.}$	6.55 ms min.

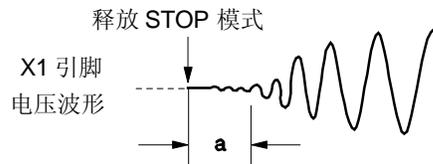
**注意事项** 1. 在经历上述时间后，从 MOST11 起各位被设置为 1，并保持。

2. 振荡稳定时间计数器的计数达到 OSTC 设置的振荡稳定时间。如果已进入 STOP 模式并使用内部高速振荡时钟作为 CPU 时钟时释放 STOP 模式，则可以按以下方式设置振荡稳定时间。

- 预期的 OSTC 振荡稳定时间  $\leq$  OSTC 设置的振荡稳定时间

因此需要注意，在释放 STOP 模式后，只有通过 OSTC 设置的振荡稳定时间期间的状态被设置到 OSTC。

3. X1 时钟振荡稳定等待时间不包括从释放 STOP 模式到时钟振荡启动这段时间（即下图“a”所示的部分）。



**备注**  $f_x$ : X1 时钟振荡频率

**(2) 振荡稳定时间选择寄存器(OSTS)**

该寄存器用于选择释放 STOP 模式后 X1 时钟振荡稳定等待时间。

在释放 STOP 模式后且 CPU 使用 X1 时钟时，等待由 OSTS 设置的时间。

当释放 STOP 模式后且 CPU 使用内部高速振荡时钟时，可通过使用 OSTC 确认是否已经历了预期的振荡稳定时间。振荡稳定时间可根据 OSTC 设置的时间来检测。

可由 8 位存储器操作指令设置 OSTS。

复位信号的产生将 OSTS 设置为 05H。

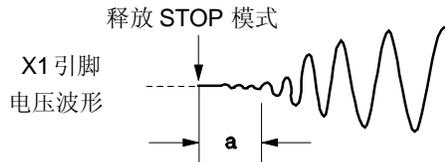
图 19-2. 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFA4H 复位后: 05H R/W

符号	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	振荡稳定时间的选择	
				$f_x = 10 \text{ MHz}$
0	0	1	$2^{11}/f_x$	204.8 $\mu\text{s}$
0	1	0	$2^{13}/f_x$	819.2 $\mu\text{s}$
0	1	1	$2^{14}/f_x$	1.64 ms
1	0	0	$2^{15}/f_x$	3.27 ms
1	0	1	$2^{16}/f_x$	6.55 ms
其它情况			禁止设置	

- 注意事项**
- 如果在 X1 时钟用作 CPU 时钟时要设置 STOP 模式，则应在执行 STOP 指令之前设置 OSTS。
  - 在 X1 时钟振荡稳定期间不要修改 OSTS 寄存器的值。
  - 振荡稳定时间计数器的计数达到 OSTS 设置的振荡稳定时间。如果已进入 STOP 模式，并使用内部高速振荡时钟时作为 CPU 时钟时释放 STOP 模式，则可以按以下方式设置振荡稳定时间。
    - 预期的 OSTC 振荡稳定时间  $\leq$  OSTS 设置的振荡稳定时间
 因此需要注意，在释放 STOP 模式后，只有通过 OSTS 设置的振荡稳定时间期间的状态被设置到 OSTC。
  - X1 时钟振荡等待时间不包括从释放 STOP 模式到时钟振荡启动这段时间（即下图“a”所示的部分）。



**备注** fx: X1 时钟振荡频率

## 19.2 待机功能的操作

### 19.2.1 HALT 模式

#### (1) HALT 模式

通过执行 HALT 指令设置 HALT 模式。无论设置前 CPU 使用的是高速系统时钟、内部高速振荡时钟还是副系统时钟，都可以设置 HALT 模式。

HALT 模式中的操作状态如下所示。

表 19-1. HALT 模式中的操作状态(1/2)

HALT 模式设置		当 CPU 使用主系统时钟并执行 HALT 指令时		
		当 CPU 使用内部高速振荡时钟( $f_{RH}$ )时	当 CPU 使用 X1 时钟( $f_x$ )时	当 CPU 使用外部主系统时钟( $f_{EXCLK}$ )时
项目				
系统时钟		停止 CPU 时钟		
主系统时钟	$f_{RH}$	操作继续 (不能停止)	保持设置 HALT 模式前的状态	
	$f_x$	保持设置 HALT 模式前的状态	操作继续 (不能停止)	保持设置 HALT 模式前的状态
	$f_{EXCLK}$	由外部时钟输入进行操作或停止		操作继续 (不能停止)
副系统时钟	$f_{XT}$	保持设置 HALT 模式前的状态		
$f_{RL}$		保持设置 HALT 模式前的状态		
CPU		操作停止		
Flash 存储器				
RAM		保持设置 HALT 模式前的状态		
端口 (锁存器)				
16-位定时器/事件计数器 00		可操作的		
8-位定时器/事件计数器	50			
	51			
	52			
8-位定时器	H0			
	H1			
	H2			
实时计数器				
看门狗定时器		可操作。当通过选项字节设置“可以由软件停止内部低速振荡器”时，停止看门狗定时器的时钟。		
蜂鸣器输出		可操作的		
10-位逐次逼近型 A/D 转换器 <sup>注</sup>				
串行接口	UART0			
	UART6			
LCD 控制器/驱动器				
曼彻斯特码发生器				
遥控接收器				
上电清零 (POC) 功能				
低电压检测功能				
外部中断				

注 仅限于  $\mu$ PD78F041x。

备注  $f_{RH}$ : 内部高速振荡时钟  
 $f_x$ : X1 时钟  
 $f_{EXCLK}$ : 外部主系统时钟  
 $f_{XT}$ : XT1 时钟  
 $f_{RL}$ : 内部低速振荡时钟

表 19-1. HALT 模式中的操作状态(2/2)

HALT 模式设置		当 CPU 使用副系统时钟并执行 HALT 指令时	
		当 CPU 使用 XT1 时钟 (f <sub>XT</sub> ) 时	
项目			
系统时钟		停止 CPU 时钟	
主系统时钟	f <sub>RH</sub>	保持设置 HALT 模式前的状态	
	f <sub>X</sub>		
	f <sub>EXCLK</sub>	由外部时钟输入进行操作或停止	
副系统时钟	f <sub>XT</sub>	操作继续 (不能停止)	
f <sub>RL</sub>		保持设置 HALT 模式前的状态	
CPU		操作停止	
Flash 存储器			
RAM		保持设置 HALT 模式前的状态	
端口 (锁存器)			
16 位定时器/事件计数器 00 <sup>注1</sup>		可操作的	
8 位定时器/事件计数器	50		
	51		
	52 <sup>Note 1</sup>		
8 位定时器	H0		
	H1		
	H2		
实时计数器			
看门狗定时器		可操作。当通过选项字节设置“可以由软件停止内部低速振荡器”时，停止看门狗定时器的时钟。	
蜂鸣器输出		可操作。然而，当外围硬件时钟停止时 (f <sub>PRS</sub> ) 操作禁止。	
10-位逐次逼近型 A/D 转换器 <sup>注2</sup>			
串行接口	UART0	可操作的	
	UART6		
LCD 控制器/驱动器			
曼彻斯特码发生器			
上电清零 (POC) 功能			
低电压检测功能			
外部中断			

注 1. 当 CPU 使用副系统时钟且内部高速振荡时钟已经停止时，不要启动使用外部时钟的外围硬件的功能。

2. 仅对  $\mu$ PD78F041x。

**备注**

f<sub>RH</sub>: 内部高速振荡时钟  
 f<sub>X</sub>: X1 时钟  
 f<sub>EXCLK</sub>: 外部主系统时钟  
 f<sub>XT</sub>: XT1 时钟  
 f<sub>RL</sub>: 内部低速振荡时钟

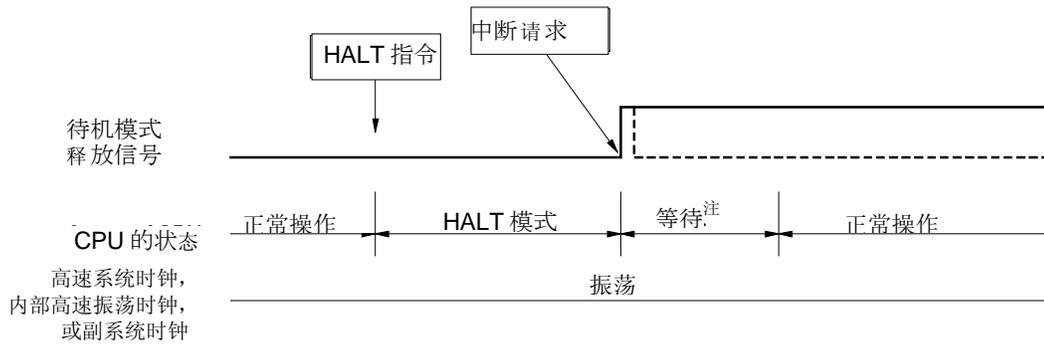
**(2) 释放 HALT 模式**

以下两种方式可以释放 HALT 模式。

**(a) 由没有被屏蔽的中断请求释放**

当产生一个没有被屏蔽的中断时，释放 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 19-3. 通过产生中断请求释放 HALT 模式



**注** 等待时间如下所示：

- 执行向量中断服务程序时： 8 或 9 个 时钟
- 不执行向量中断服务程序时： 2 或 3 个 时钟

**备注** 虚线表示释放待机模式的中断请求被响应的情况。

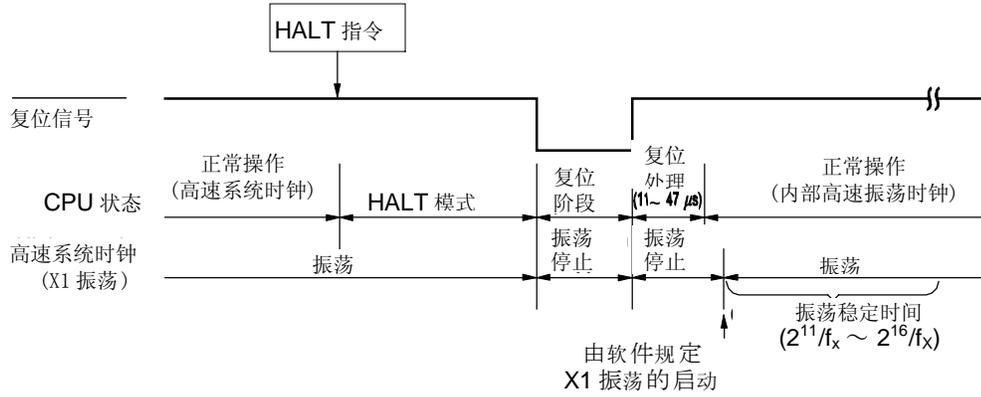
(b) 通过产生的复位信号释放

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开始执行。

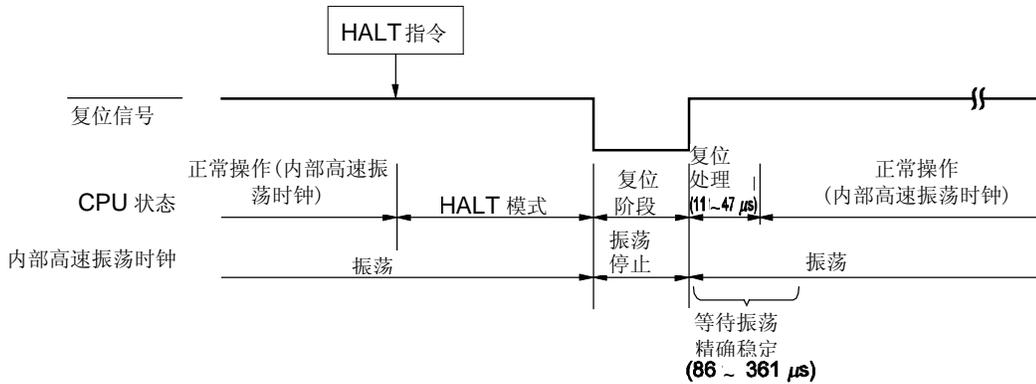
<R>

图 19-4. 通过复位释放 HALT 模式

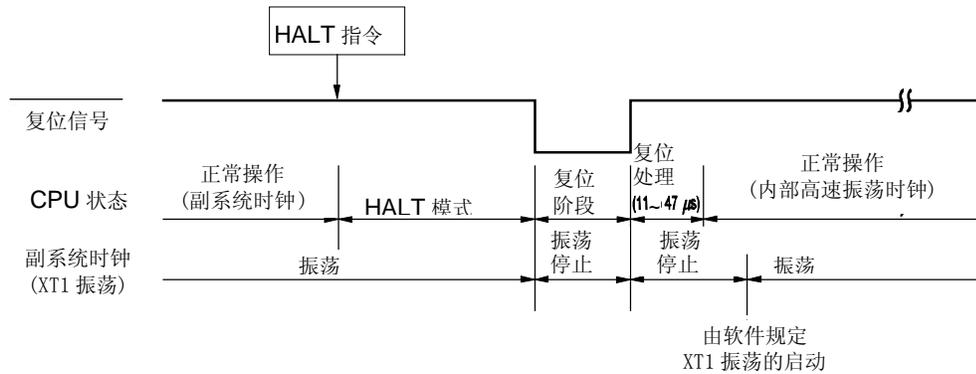
(1) CPU 使用高速系统时钟



(2) CPU 使用内部高速振荡时钟



(3) CPU 使用副系统时钟



备注 fx: X1 时钟振荡频率

表 19-2. 在 HALT 模式中中断请求对应的操作

释放源	MK <sub>xx</sub>	PR <sub>xx</sub>	IE	ISP	操作
可屏蔽中断请求	0	0	0	×	执行下一个地址的指令
	0	0	1	×	执行中断服务程序
	0	1	0	1	执行下一个地址的指令
	0	1	×	0	
	0	1	1	1	执行中断服务程序
	1	×	×	×	保持 HALT 模式
复位	-	-	×	×	复位处理

×: 不必考虑

### 19.2.2 STOP 模式

#### (1) STOP 模式设置及操作状态

通过执行 STOP 指令设置 STOP 模式。仅当设置 STOP 模式前 CPU 使用主系统时钟时才可以设置 STOP 模式。

**注意事项** 由于中断请求信号用于释放待机模式，因此如果一个中断源的中断请求标志被设置且中断屏蔽标志被清零时，则立即释放待机模式。因此，在执行 STOP 指令后，立即将 STOP 模式复位到 HALT 模式，并且在经历了 OSTS 设置的等待时间后，系统返回操作模式。

STOP 模式中的操作状态如下所示。

表 19-3. STOP 模式中的操作状态

STOP 模式设置		当 CPU 使用主系统时钟并执行 STOP 指令时		
		当 CPU 使用内部高速振荡时钟 (f <sub>int</sub> ) 时	当 CPU 使用 X1 时钟 (f <sub>x</sub> ) 时	当 CPU 使用外部主系统时钟 (f <sub>EXCLK</sub> ) 时
项目				
系统时钟		停止 CPU 时钟		
主系统时钟	f <sub>RH</sub>	停止		
	f <sub>x</sub>			
	f <sub>EXCLK</sub>	输入无效		
副系统时钟	f <sub>XT</sub>	保持设置 STOP 模式前的状态		
f <sub>RL</sub>		保持设置 STOP 模式前的状态		
CPU		操作停止		
Flash 存储器				
RAM		保持设置 STOP 模式前的状态		
端口 (锁存器)				
16-位定时器/时间计数器 00 <sup>※1</sup>		只有当 TM52 输出或 TI000 选作计数时钟时才能操作		
8-位定时器/时间计数器	50	保持设置 STOP 模式前的状态		
	51			
	52 <sup>Note 1</sup>	只有当 TI52 选作计数时钟时才能操作		
8-位定时器	H0	只有在 8-位定时器/事件计数器 50 操作期间, 当 TM50 输出选作计数时钟时才能操作		
	H1	只有当 f <sub>RL</sub> , f <sub>RL</sub> /2 <sup>7</sup> , f <sub>RL</sub> /2 <sup>9</sup> 选作计数时钟时才能操作		
	H2	操作停止		
实时计数器		只有当副系统时钟选作计数时钟时才能操作		
看门狗定时器		可操作。当选项字节设置“可由软件停止内部低速振荡器”时, 停止看门狗定时器的时钟。		
蜂鸣器输出		操作停止		
10-位逐次逼近型 A/D 转换器 <sup>※2</sup>				
串行接口	UART0	在 8 位定时器/事件计数器 50 操作期间, 仅当选择 TM50 输出作为计数时钟时可操作		
	UART6			
LCD 控制器/驱动器		仅当选择副系统时钟作为计数时钟时可操作		
曼彻斯特码发生器		操作停止		
上电清零 (POC) 功能		可操作的		
低电压检测功能				
外部中断				

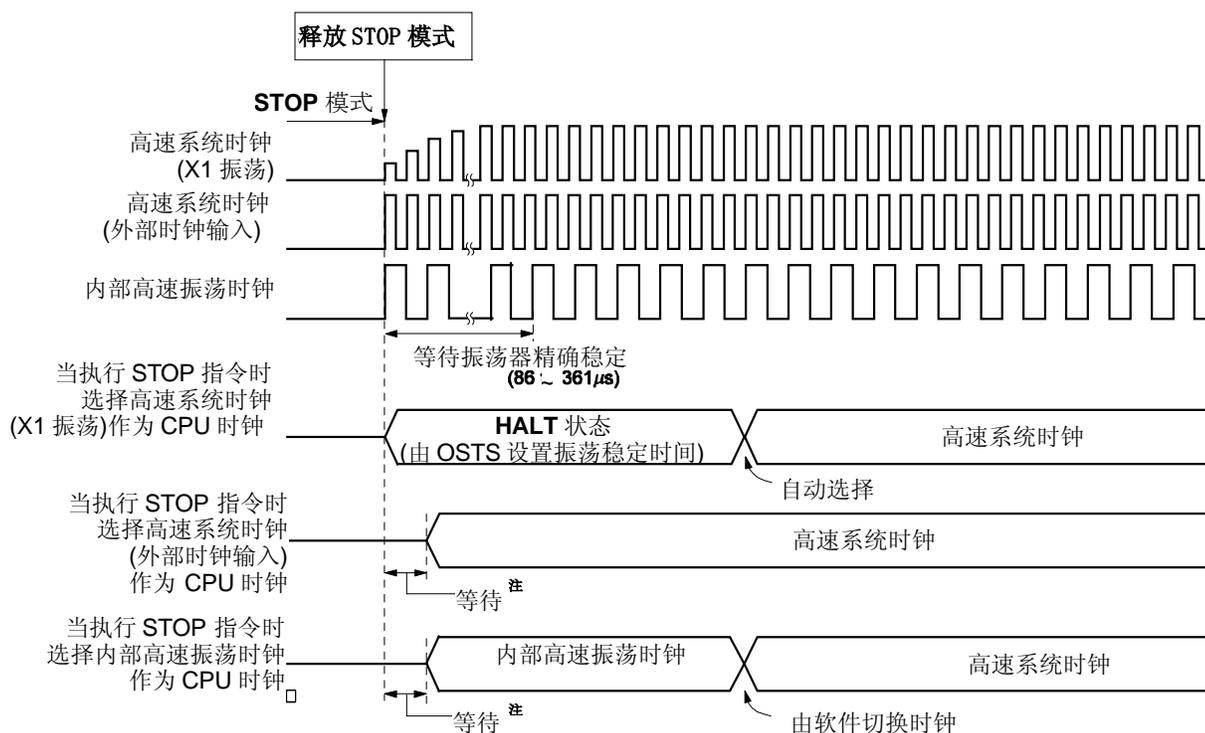
注 1. 当 CPU 使用副系统时钟且内部高速振荡时钟已经停止时, 不要启动使用外部时钟的外围硬件的功能。  
 2. 仅对  $\mu$ PD78F041x。

备注 f<sub>RH</sub>: 内部高速振荡时钟  
 f<sub>x</sub>: X1 时钟  
 f<sub>EXCLK</sub>: 外部主系统时钟  
 f<sub>XT</sub>: XT1 时钟  
 f<sub>RL</sub>: 内部低速振荡时钟

- 注意事项
1. 在 STOP 模式释放后，要使用在 STOP 模式下停止操作的外围硬件，以及在 STOP 模式下时钟停止操作的外围硬件，必须重新启动外围硬件。
  2. 即使通过选项字节选择“可由软件停止内部低速振荡器”，在 STOP 模式下，内部低速振荡时钟仍继续振荡，并保持 STOP 模式设置前的状态。要在 STOP 模式下停止内部低速振荡器的振荡，应通过软件停止该振荡器，然后执行 STOP 指令。
  3. 当 CPU 使用高速系统时钟(X1 振荡)时，为了在 STOP 模式释放后缩短振荡稳定时间，可在执行 STOP 指令前将 CPU 时钟暂时切换到内部高速振荡时钟。STOP 模式释放后，在将 CPU 时钟从内部高速振荡时钟切换到高速系统时钟 (X1 振荡) 之前，应使用振荡稳定时间计数器的状态寄存器(OSTC)检测振荡稳定时间。

(2) 释放 STOP 模式

图 19-5. 释放 STOP 模式时的操作时序(当未屏蔽的中断请求产生时)



注 等待时间如下所示：

- 执行向量中断服务程序时： 8 或 9 个 时钟
- 不执行向量中断服务程序时： 2 或 3 个 时钟

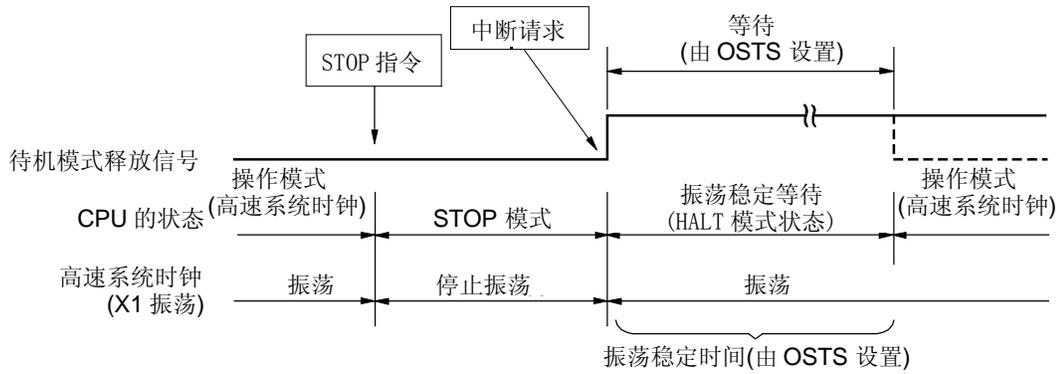
STOP 模式可通过以下两种方式释放。

**(a) 由没有被屏蔽的中断请求释放**

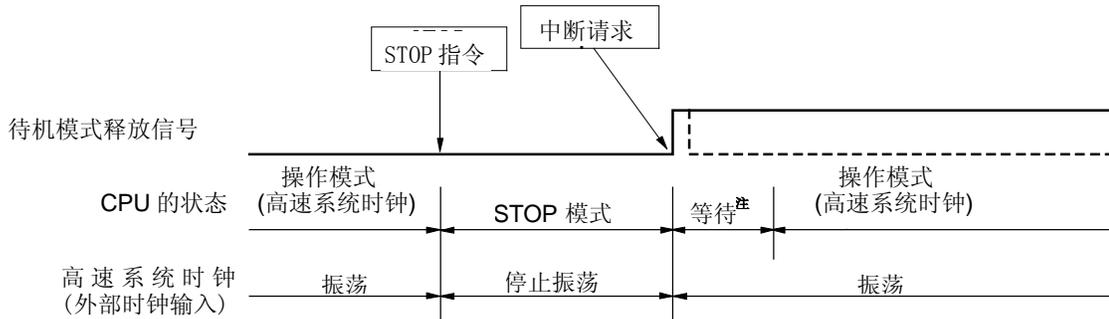
当产生一个没有被屏蔽的中断请求时，释放 STOP 模式。经历振荡稳定时间后，如果允许响应中断，则执行向量中断服务程序。如果禁止响应中断，则执行下一个地址的指令。

图 19-6. 通过产生中断请求释放 STOP 模式 (1/2)

**(1) 当 CPU 使用高速系统时钟 (X1 振荡器)**



**(2) 当 CPU 使用高速系统时钟(外部时钟输入)**



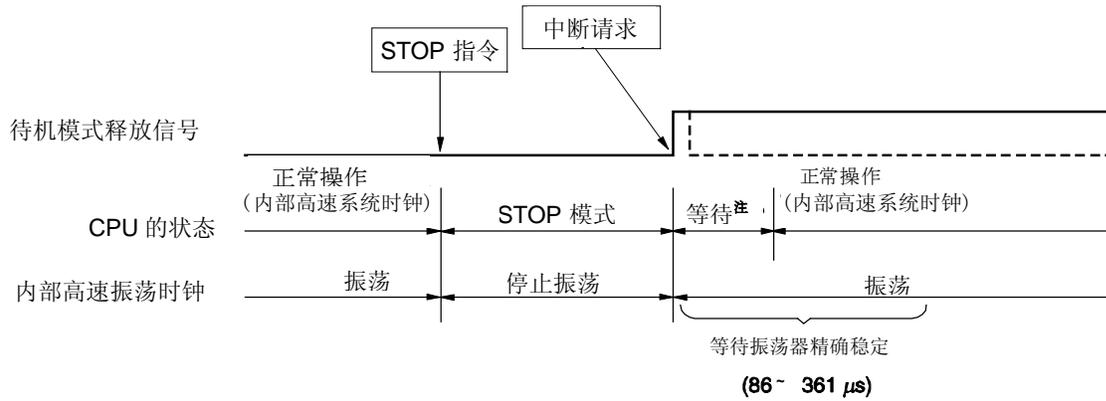
注 等待时间如下所示：

- 执行向量中断服务程序时：8 或 9 个 时钟
- 不执行向量中断服务程序时：2 或 3 个 时钟

备注 虚线表示释放待机模式的中断请求被响应的情况。

图 19-6. 通过产生中断请求释放 STOP 模式(2/2)

(3) 当 CPU 使用内部高速振荡时钟



注 等待时间如下所示:

- 执行向量中断服务程序时: 8 或 9 个 时钟
- 不执行向量中断服务程序时: 2 或 3 个 时钟

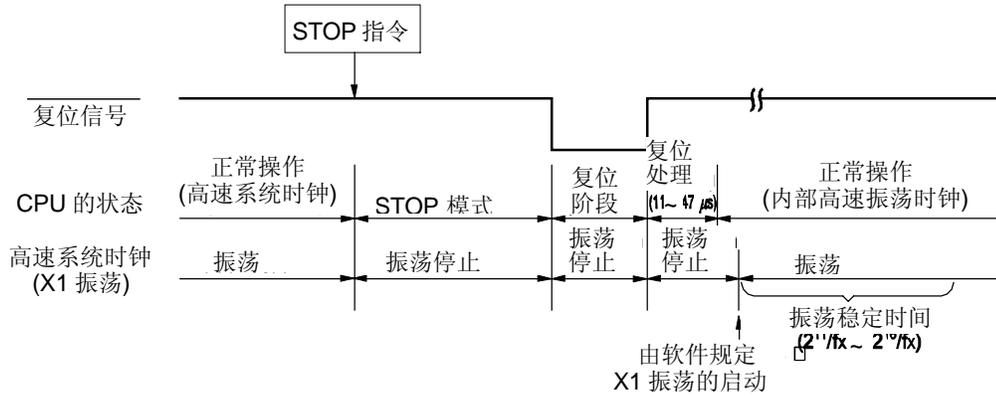
备注 虚线表示释放待机模式的中断请求被响应的情况。

(b) 由复位信号的产生释放

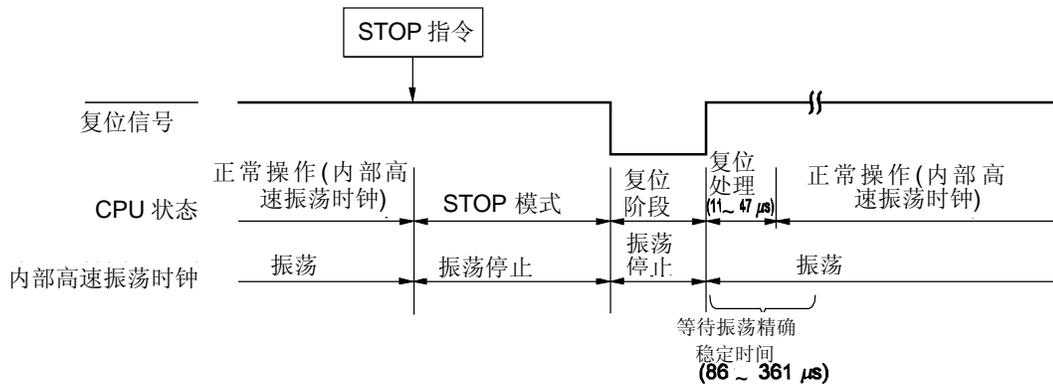
当产生复位信号时，释放 STOP 模式，然后在正常复位操作情况下，当程序转向复位向量地址后执行之。

图 19-7. 由复位释放 STOP 模式

(1) 当 CPU 使用高速系统时钟



(2) 当 CPU 使用内部高速振荡时钟



备注 fx: X1 时钟振荡频率

表 19-4. 在 STOP 模式中中断请求对应的操作

释放源	MKxx	PRxx	IE	ISP	操作
可屏蔽的中断请求	0	0	0	×	执行下一个地址的指令
	0	0	1	×	执行中断服务程序
	0	1	0	1	执行下一个地址的指令
	0	1	×	0	
	0	1	1	1	执行中断服务程序
	1	×	×	×	保持 STOP 模式
复位	-	-	×	×	复位处理

x: 不必考虑

## 第二十章 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电路 (LVI) 的检测电压和供电电压引起的内部复位

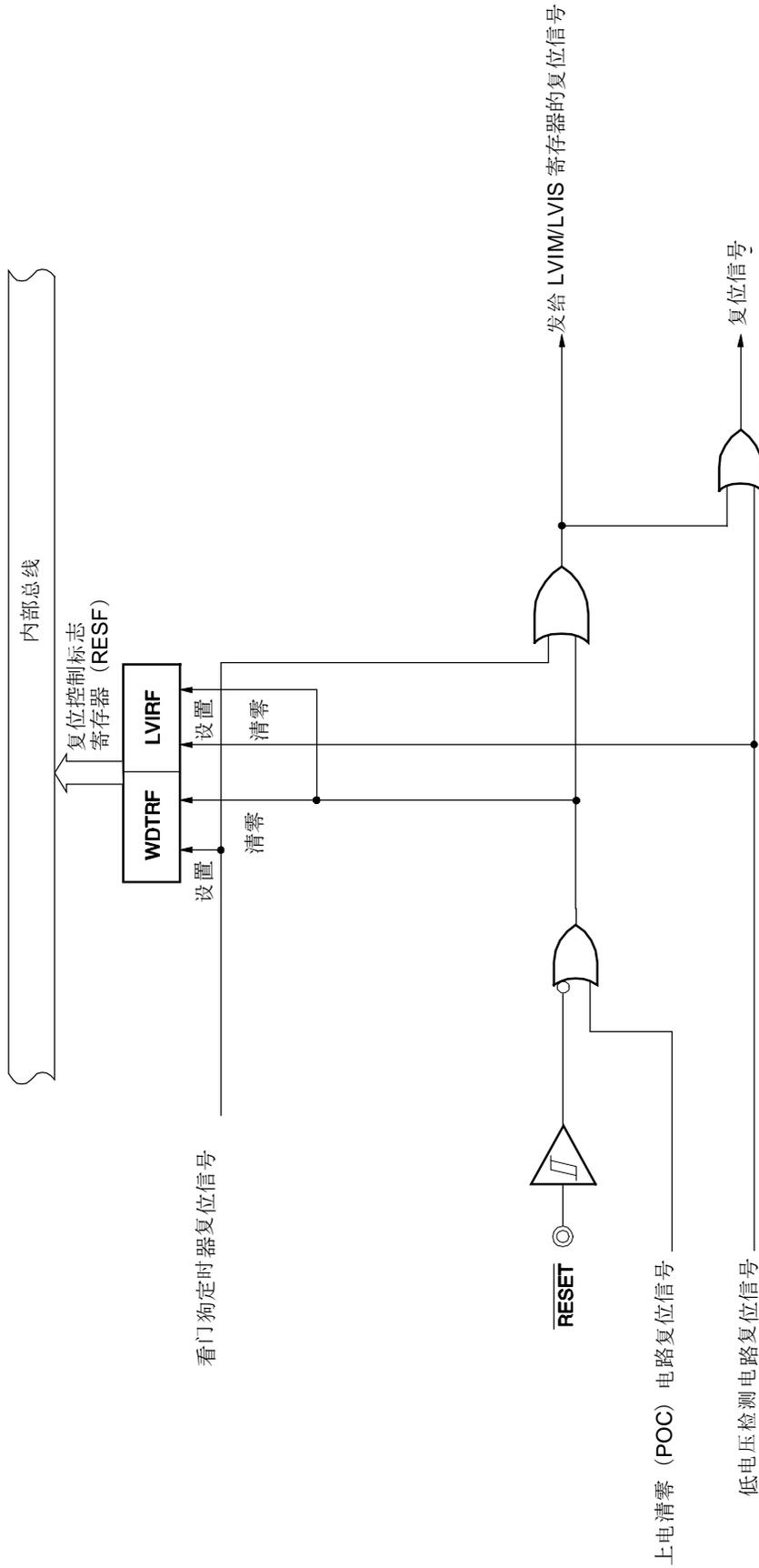
外部复位与内部复位在功能上没有什么区别。在这两种情况中，当产生复位信号时，程序都是从地址 0000H 和 0001H 处开始执行。

当  $\overline{\text{RESET}}$  引脚输入为低电平、看门狗定时器溢出或 POC 和 LVI 电路的电压检测都可以引起复位，每项硬件的设置状态如表 20-1 和 20-2 所示。在复位信号产生期间或在复位释放后振荡稳定时间内，每个引脚均为高阻抗。

当  $\overline{\text{RESET}}$  引脚输入低电平时，设备被复位。当  $\overline{\text{RESET}}$  引脚输入高电平时，设备从复位状态释放，并在复位处理后使用内部高速振荡时钟执行程序。由看门狗定时器引起的复位自动释放，并在复位处理后使用内部高速振荡时钟执行程序（参见图 20-2 ~ 20-4）。由 POC 和 LVI 电路供电电压检测引起的复位，在复位后，当  $V_{DD} \geq V_{POC}$  或  $V_{DD} \geq V_{LVI}$  时自动释放，并使用内部高速振荡时钟执行程序（参见第二十一章 上电清零电路 和 第二十二章 低电压检测电路）。

- 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至少应为 10  $\mu\text{s}$ 。
  2. 复位输入期间，X1 时钟、XT1 时钟、内部高速振荡时钟和内部低速振荡时钟停止振荡。外部主系统时钟输入和外部副系统时钟输入无效。
  3. 在通过复位释放 STOP 模式时，复位输入期间保持 STOP 模式的内容。但端口引脚变为高阻态。

图 20-1. 复位功能的框图



注意事项 LVI 电路内部复位功能不能复位 LVI 电路。

- 备注
1. LVIM: 低电压检测寄存器
  2. LVIS: 低电压检测等级选择寄存器

图 20-2. 由 RESET 输入进行复位的时序

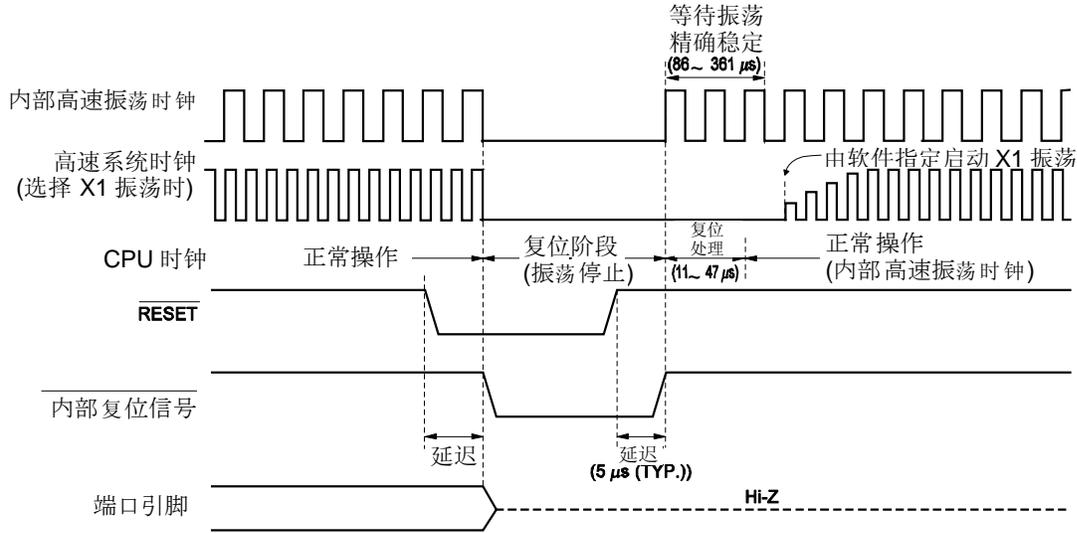
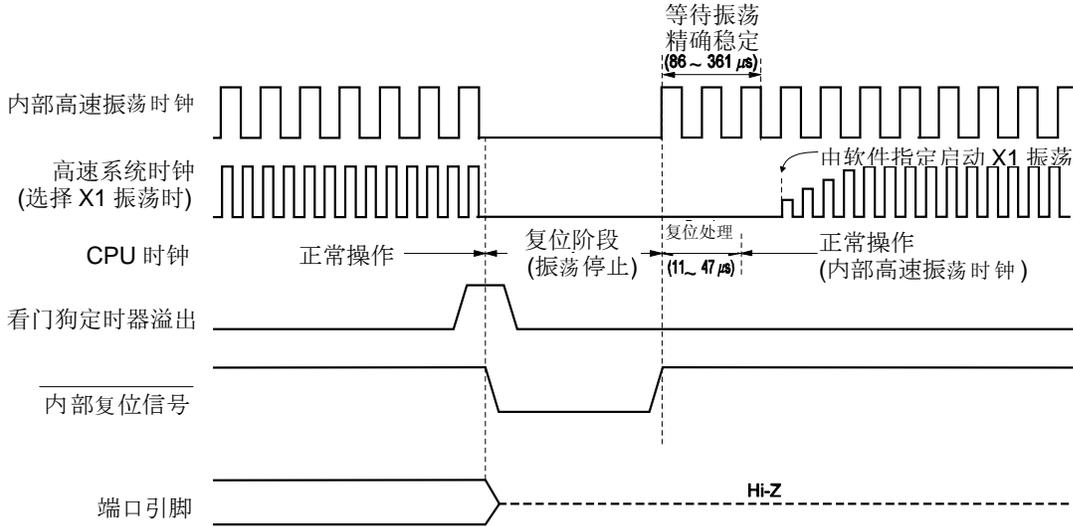
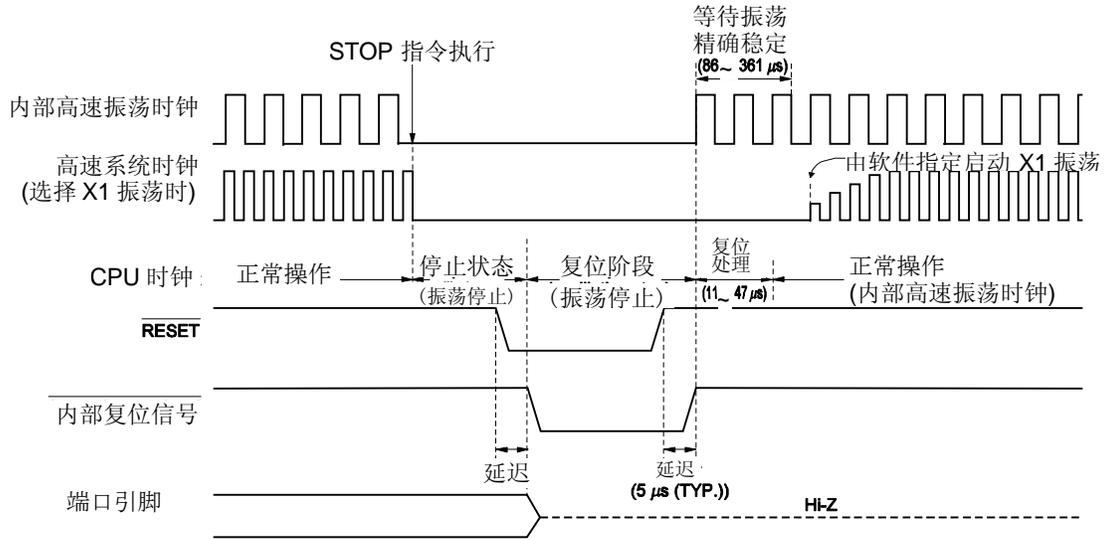


图 20-3. 由于看门狗定时器溢出进行复位的时序



注意事项 看门狗定时器内部复位功能也可以复位看门狗定时器。

图 20-4. 在 STOP 模式下通过 RESET 输入进行复位的时序



**备注** 如需了解上电清零 (POC) 电路和低电压检测电路的复位时序, 可参见第二十一章 上电清零电路 和 第二十二章 低电压检测电路。

表 20-1. 复位期间的操作状态

项目		复位期间
系统时钟		停止 CPU 时钟
主系统时钟	f <sub>RH</sub>	停止操作
	f <sub>X</sub>	停止操作(引脚为 I/O 端口模式)
	f <sub>EXCLK</sub>	时钟输入无效 (引脚为 I/O 端口模式)
副系统时钟	f <sub>XT</sub>	停止操作 (引脚为 I/O 端口模式)
f <sub>RL</sub>		操作停止
CPU		
Flash 存储器		
RAM		
端口 (锁存器)		
16-位定时器/时间计数器	00	
8-位定时器/时间计数器	50	
	51	
	52	
8-位定时器	H0	
	H1	
	H2	
实时计数器		
看门狗定时器		
蜂鸣器输出		
10-位逐次逼近型 A/D 转换器 <sup>注</sup>		
串行接口	UART0	
	UART6	
LCD 控制器/驱动器		
曼彻斯特码发生器		
上电清零功能		可操作
低电压检测功能		停止操作
外部中断		

注 仅对  $\mu$ PD78F041x。

备注

- f<sub>RH</sub>: 内部高速振荡时钟
- f<sub>X</sub>: X1 振荡时钟
- f<sub>EXCLK</sub>: 外部主系统时钟
- f<sub>XT</sub>: XT1 振荡时钟
- f<sub>RL</sub>: 内部低速振荡时钟

表 20-2. 复位响应后的硬件状态(1/3)

硬件		响应复位后的状态 <sup>注1</sup>
程序计数器 (PC)		设置复位向量表的内容 (0000H, 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 <sup>注2</sup>
	通用寄存器	不确定 <sup>注2</sup>
端口寄存器(P1 to P4, P10 ~ P12, P14, P15) (输出锁存器)		00H
端口模式寄存器(PM1 ~ PM4, PM10 ~ PM12, PM14, PM15)		FFH
上拉电阻选择寄存器(PU1, PU3, PU4, PU10 ~ PU12, PU14, PU15)		00H
端口功能寄存器 (PF1)		00H
端口功能寄存器(PF2)		00H
端口功能寄存器(PFALL)		00H
内部存储器容量切换寄存器(IMS)		CFH <sup>注3</sup>
时钟操作模式选择寄存器(OSCCTL)		00H
处理器时钟控制寄存器(PCC)		01H
内部振荡模式寄存器(RCM)		80H
主 OSC 控制寄存器(MOC)		80H
主时钟模式寄存器 (MCM)		00H
振荡稳定时间计数器状态寄存器(OSTC)		00H
振荡稳定时间选择寄存器 (OSTS)		05H
内部高速振荡修正寄存器(HIOTRM)		10H
16-位定时器/事件计数器 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/事件计数器 50, 51, 52	定时器计数器 50, 51, 52 (TM50, TM51, TM52)	00H
	比较寄存器 50, 51, 52 (CR50, CR51, CR52)	00H
	定时器时钟选择寄存器 50, 51, 52 (TCL50, TCL51, TCL52)	00H
	模式控制寄存器 50, 51, 52 (TMC50, TMC51, TMC52)	00H

注 1. 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。  
 2. 当在待机模式中进行复位时，即使复位后也保持复位前的状态。  
 3. 不管内部存储器容量，78K0/LC3 中所有产品的内部存储器容量切换寄存器 (IMS) 初始值都是恒定的 (IMS = CFH)。因此，按以下设置各个产品相应值。

Flash 存储器版本 (78K0/LC3)	IMS	ROM 容量	内部高速 RAM 容量
μPD78F0400, 78F0410	42H	8 KB	512 字节
μPD78F0401, 78F0411	04H	16 KB	768 字节
μPD78F0402, 78F0412	C6H	24 KB	1 KB
μPD78F0403, 78F0413	C8H	32 KB	

表 20-2. 复位响应后的硬件状态(2/3)

硬件		响应复位后的状态 <sup>注1</sup>
8-位计数器 H0, H1, H2	比较寄存器 00, 10, 01, 11, 02, 12 (CMP00, CMP10, CMP01, CMP11, CMP02, CMP12)	00H
	模式寄存器 (TMHMD0, TMHMD1, TMHMD2)	00H
	载波控制寄存器 1 (TMCYC1) <sup>注2</sup>	00H
实时计数器	时钟选择寄存器 (RTCCL)	00H
	副计数寄存器 (RSUBC)	0000H
	秒计数寄存器 (SEC)	00H
	分技术寄存器 (MIN)	00H
	小时技术寄存器 (HOUR)	12H
	周计数寄存器 (WEEK)	00H
	天计数寄存器 (DAY)	01H
	月计数寄存器 (MONTH)	01H
	年计数寄存器 (YEAR)	00H
	钟表误差修正计数器 (SUBCUD)	00H
	闹表分钟寄存器 (ALARMWM)	00H
	闹表小时寄存器 (ALARMWH)	12H
	闹表星期寄存器 (ALARMWW)	00H
	控制寄存器 0 (RTCC0)	00H
控制寄存器 1 (RTCC1)	00H	
控制寄存器 2 (RTCC2)	00H	
蜂鸣器输出控制器	时钟输出选择寄存器 (CKS)	00H
看门狗定时器	使能寄存器 (WDTE)	1AH/9AH <sup>注3</sup>
10-位逐次逼近型 A/D 转换器 <sup>注4</sup>	10 位 A/D 转换结果寄存器 (ADCR)	0000H
	8 位 A/D 转换结果寄存器 (ADCRH)	00H
	模式寄存器 (ADM)	00H
	模拟输入通道选择寄存器 (ADS)	00H
	A/D 端口配置寄存器 (ADPC0)	08H
串行接口 UART0	接收缓冲寄存器 0 (RXB0)	FFH
	发送移位寄存器 0 (TXS0)	FFH
	异步串行接口操作模式寄存器 0 (ASIM0)	01H
	异步串行接口接收错误状态寄存器 0 (ASIS0)	00H
	波特率发生器控制寄存器 0 (BRGC0)	1FH

- 注
1. 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。
  2. 仅用于 8 位定时器 H1。
  3. 通过设置选项字节确定 WDTE 的复位值。
  4. 仅对  $\mu$ PD78F041x。

表 20-2. 复位响应后的硬件状态(3/3)

硬件		响应复位后的状态 <sup>注 1</sup>
串行接口	接收缓冲寄存器 6 (RXB6)	FFH
	发送缓冲寄存器 6 (TXB6)	FFH
	异步串行接口操作模式寄存器 6 (ASIM6)	01H
	异步串行接口接收错误状态寄存器 6 (ASIS6)	00H
	异步串行接口发送状态寄存器 6 (ASIF6)	00H
	时钟选择寄存器 6 (CKSR6)	00H
	波特率发生器控制寄存器 6 (BRGC6)	FFH
	异步串行接口控制寄存器 6 (ASICL6)	16H
	输入切换控制寄存器 (ISC)	00H
LCD 控制器/驱动器	LCD 模式设置寄存器 (LCDMD)	00H
	LCD 显示模式寄存器 (LCDM)	00H
	LCD 时钟控制寄存器 (LDCD0)	00H
曼彻斯特码发生器	发送缓冲器寄存器 (MC0TX)	FFH
	发送位数规定计数器(MC0BIT)	07H
	控制寄存器 0 (MC0CTL0)	10H
	控制寄存器 1 (MC0CTL1)	00H
	控制寄存器 2 (MC0CTL2)	1FH
	状态寄存器 (MC0STR)	00H
按键中断	按键返回模式寄存器 (KRM)	00H
复位功能	复位控制标志寄存器 (RESF)	00H <sup>注 2</sup>
低电压检测电路	低电压检测寄存器 (LVIM)	00H <sup>注 2</sup>
	低电压检测等级选择寄存器 (LVIS)	00H <sup>注 2</sup>
中断	请求标志寄存器 0L, 0H, 1L, 1H (IF0L, IF0H, IF1L, IF1H)	00H
	屏蔽标志寄存器 0L, 0H, 1L, 1H (MK0L, MK0H, MK1L, MK1H)	FFH
	优先级指定标志寄存器 0L, 0H, 1L, 1H (PROL, PROH, PR1L, PR1H)	FFH
	外部中断上升沿使能寄存器 (EGP)	00H
	外部中断下降沿使能寄存器 (EGN)	00H

- 注
1. 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。
  2. 这些值依据复位源的不同而变化。

寄存器		复位源			
		RESET 输入	由 POC 进行复位	由 WDT 进行复位	由 LVI 进行复位
RESF	WDRF 位	清零 (0)	清零 (0)	设置 (1)	RESF
	LVIRF 位			保持	
LVIM		清零 (00H)	清零 (00H)	清零 (00H)	保持
LVIS					

## 20.1 确认复位源的寄存器

78K0/LC3 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。

RESET 输入、由 POC 电路引起的复位输入以及读取 RESF 可将 RESF 清零 (00H)。

图 20-5. 复位控制标志寄存器 (RESF) 的格式

地址: FFACH 复位后: 00H\* R

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

WDTRF	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 (LVI) 产生的内部复位请求
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变化。

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 20-3 所示。

表 20-3. 产生复位请求时的 RESF 状态

标志 \ 复位源	RESET 输入	由 POC 引起的复位	由 WDT 引起的复位	由 LVI 引起的复位
WDTRF	清零(0)	清零(0)	设置(1)	保持
LVIRF			保持	设置(1)

### 21.1 上电清零电路的功能

上电清零电路（POC）有以下功能。

- 在上电时产生内部复位信号。  
在 1.59 V POC 模式下 (选项字节: POCMODE = 0)，当供电电压 ( $V_{DD}$ ) 超过  $1.59\text{ V} \pm 0.15\text{ V}$  时，释放复位信号。  
在 2.7 V/1.59 V POC 模式下 (选项字节: POCMODE = 1)，当供电电压 ( $V_{DD}$ ) 超过  $2.7\text{ V} \pm 0.2\text{ V}$  时，释放复位信号。
- 比较供电电压 ( $V_{DD}$ ) 和检测电压 ( $V_{POC} = 1.59\text{ V} \pm 0.15\text{ V}$ )，并在  $V_{DD} < V_{POC}$  时产生内部复位信号，在  $V_{DD} \geq V_{POC}$  释放复位。

**注意事项** 如果由 POC 电路产生内部复位信号，则将复位控制标志寄存器（RESF）清零（00H）。

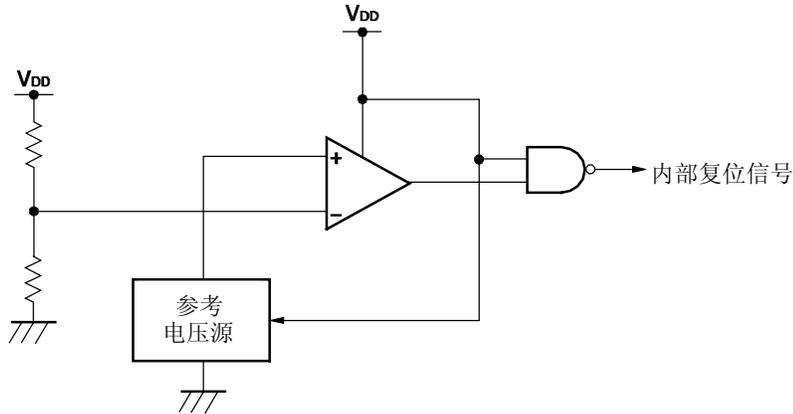
**备注** 78K0/LC3 集成了多种可以产生内部复位信号的硬件功能。RESF 中有一个标志用于指示复位源，用于看门狗定时器（WDT）或低电压检测电路（LVI）产生的内部复位信号。当 WDT 或 LVI 产生内部复位信号时，不对 RESF 清零（00H），标志位被设置为 1。

如需了解 RESF 的详细内容，参见第二十章 复位功能。

## 21.2 上电清零电路的配置

上电清零电路的框图如图 21-1 所示。

图 21-1. 上电清零电路的框图



## 21.3 上电清零电路的操作

### (1) 1.59 V POC 模式 (选项字节: POCMODE = 0)

- 上电时产生内部复位信号。当供电电压( $V_{DD}$ )超过检测电压( $V_{POC} = 1.59\text{ V} \pm 0.15\text{ V}$ )时, 释放复位状态。
- 比较供电电压( $V_{DD}$ )和检测电压( $V_{POC} = 1.59\text{ V} \pm 0.15\text{ V}$ )。当  $V_{DD} < V_{POC}$  时, 产生内部复位信号。当  $V_{DD} \geq V_{POC}$ , 释放内部复位信号。

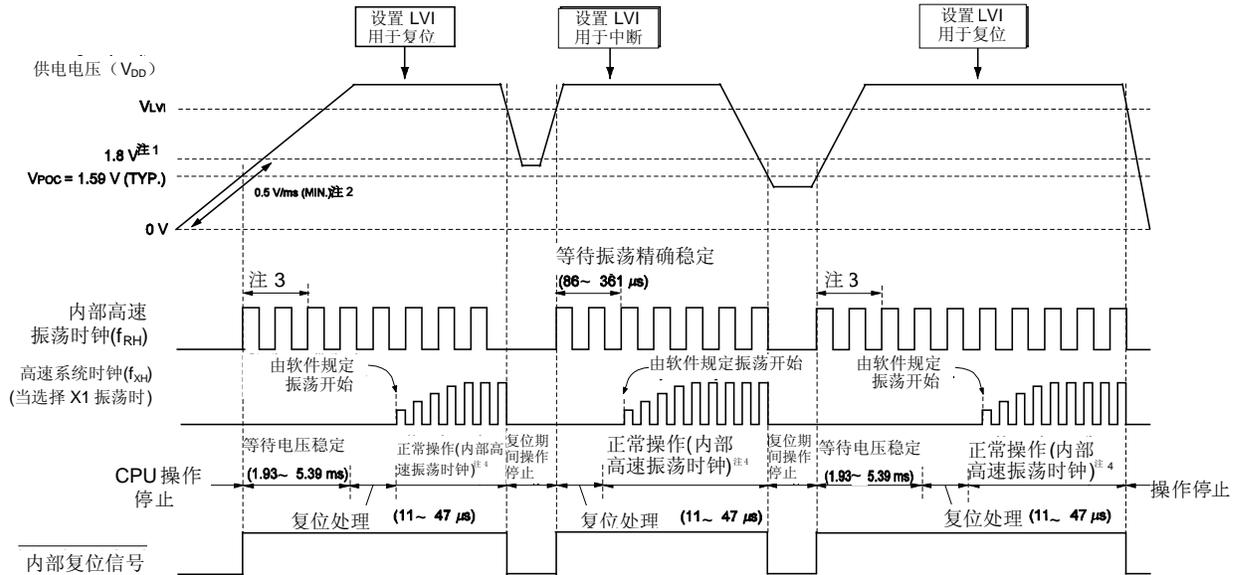
### (2) 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1)

- 上电时产生内部复位信号。当供电电压( $V_{DD}$ )超过检测电压( $V_{DDPOC} = 2.7\text{ V} \pm 0.2\text{ V}$ )时, 释放复位状态。
- 比较供电电压( $V_{DD}$ )和检测电压( $V_{POC} = 1.59\text{ V} \pm 0.15\text{ V}$ )。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。

由上电清零电路和低电压检测电路产生内部复位信号的时序如下所示。

图 21-2. 由上电清零电路和低电压检测电路产生内部复位信号的时序 (1/2)

## (1) 1.59 V POC 模式 (选项字节: POCMODE = 0)



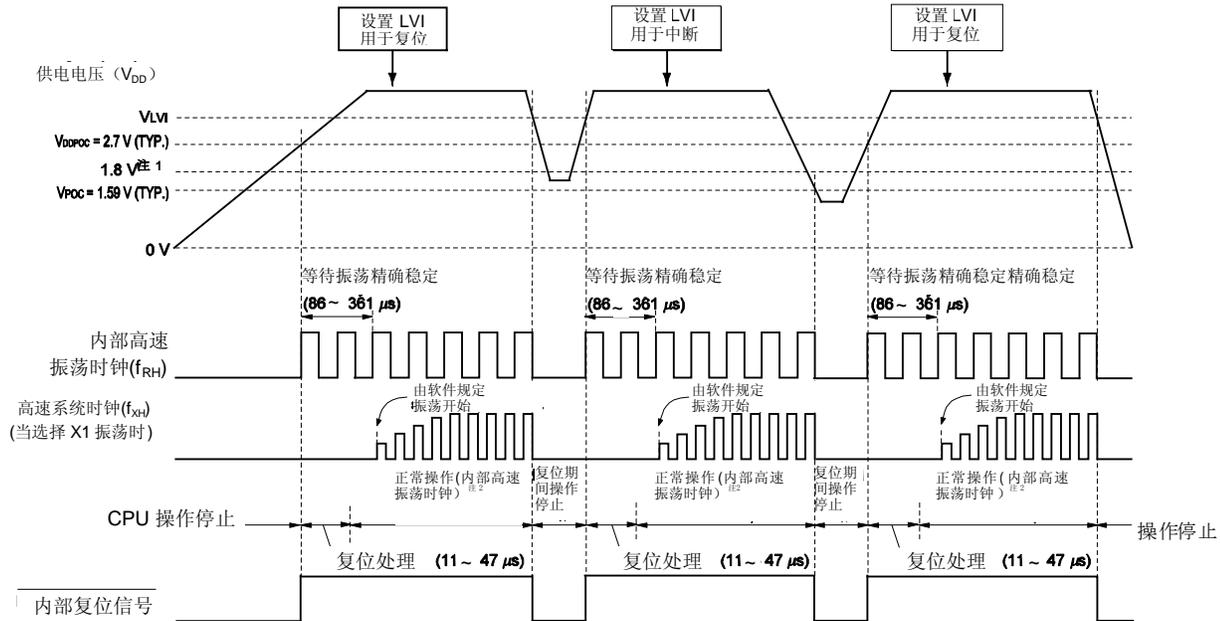
- 注
1. 保证操作的电压范围： $1.8 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$ 。当供电电压下降时，为使状态低于  $1.8 \text{ V}$  复位状态，可以使用低电压检测电路的复位功能，或通过  $\overline{\text{RESET}}$  引脚输入低电平。
  2. 如果上电后电压上升到  $1.8 \text{ V}$  的速率小于  $0.5 \text{ V/ms (MIN.)}$ ，则在上电后且电压抵达  $1.8 \text{ V}$  以前输入一个低电平到  $\overline{\text{RESET}}$  引脚，或者通过使用选项字节 ( $\text{POCMODE} = 1$ ) 设置  $2.7 \text{ V}/1.59 \text{ V}$  POC 模式。
  3. 内部电压稳定时间包括内部高速振荡时钟的振荡精确稳定时间。
  4. 可以选择内部高速振荡时钟和高速系统时钟或副系统时钟作为 CPU 时钟。如果要使用 X1 时钟，可以通过 OSTC 确认是否经历了振荡稳定时间。而如果使用 XT1 时钟，可以通过定时器功能来确认是否经历了振荡稳定时间。
- 3.

**注意事项** 在释放复位状态后通过软件设置低电压检测电路 (参见第二十二章 低电压检测电路)。

**备注**  $V_{LVI}$ : LVI 检测电压  
 $V_{POC}$ : POC 检测电压

图 21-2. 由上电清零电路和低电压检测电路产生内部复位信号的时序(2/2)

## (2) 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1)



- 注
1. 保证操作的电压范围： $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ 。当供电电压下降时，为使状态低于  $1.8\text{ V}$  复位状态，可以使用低电压检测电路的复位功能，或向  $\overline{\text{RESET}}$  引脚输入低电平。
  2. 可以选择内部高速振荡时钟和高速系统时钟或副系统时钟作为 CPU 时钟。如果要使用 X1 时钟，可以通过 OSTC 确认是否经历了振荡稳定时间。而如果要使用 XT1 时钟，可以通过定时器功能来确认是否经历了振荡稳定时间。

- 注意事项
1. 在释放复位状态后通过软件设置低电压检测电路 (参见第二十二章 低电压检测电路)。
  2. 供电电压达到  $1.59\text{ V (TYP.)}$  之后需要  $1.93 \sim 5.39\text{ ms}$  的电压振荡稳定时间。如果在  $1.93\text{ ms}$  之内供电电压从  $1.59\text{ V (TYP.)}$  升到  $2.7\text{ V (TYP.)}$ ，在复位处理之前，自动产生  $0 \sim 5.39\text{ ms}$  的供电电压振荡稳定时间。

备注

$V_{LVI}$ : LVI 检测电压  
 $V_{POC}$ : POC 检测电压

### 21.4 上电清零电路使用注意事项

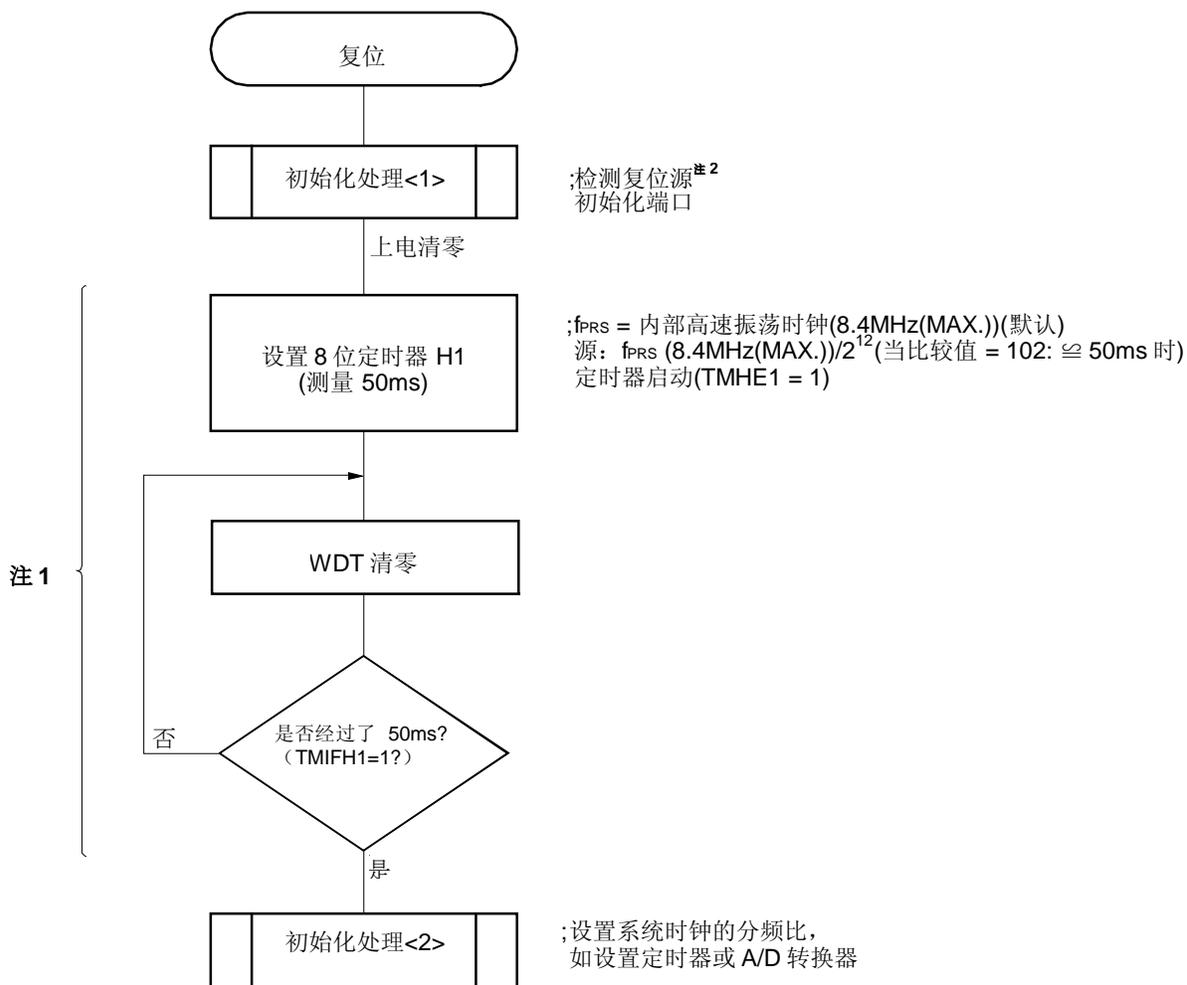
在系统中，如果供电电压 (VDD) 接近 POC 检测电压 (VPOC) 时的一段时间内产生波动，则系统可能会反复复位和释放复位。在这种情况下，可采用下列方法任意设置从复位释放到微控制器的启动所需要的时间。

<方法>

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

图 21-3. 复位释放后软件处理过程示例 (1/2)

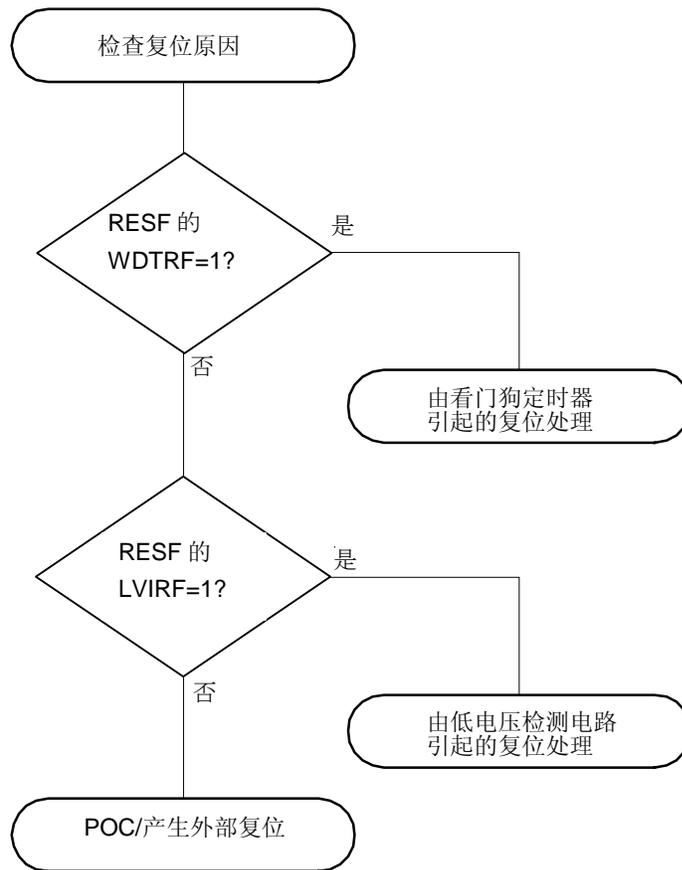
- 如果供电电压接近 POC 检测电压，其波动时间 ≤ 50 ms



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

图 21-3. 复位释放后软件处理过程示例 (2/2)

## •检查复位原因



## 第二十二章 低电压检测电路

### 22.1 低电压检测电路的功能

低电压检测电路 (LVI) 有以下功能。

- 比较供电电压 ( $V_{DD}$ ) 和检测电压 ( $V_{LVI}$ )，并在  $V_{DD} < V_{LVI}$  时产生内部中断信号或内部复位信号。通过软件改变供电电压的检测等级 (16 等级)。
- 比较外部输入引脚的输入电压 ( $EXLVI$ ) 与检测电压 ( $V_{EXLVI} = 1.21 \text{ V (TYP.)}$ )，并在  $EXLVI < V_{EXLVI}$  时产生内部中断信号或内部复位信号。
- 可由软件选择供电电压 ( $V_{DD}$ ) 或外部输入引脚的输入电压 ( $EXLVI$ )。
- 可由软件选择中断或复位功能。
- 在 STOP 模式下可操作。

复位信号和中断信号依据软件的选择按照如下方式产生。

供电电压 ( $V_{DD}$ ) 的点评检测的选择 (LVISEL = 0)		从外部输入引脚的输入电压电平检测的选择 (EXLVI) (LVISEL = 1)	
选择复位 (LVIMD = 1).	选择中断 (LVIMD = 0).	选择复位 (LVIMD = 1).	选择中断 (LVIMD = 0).
当 $V_{DD} < V_{LVI}$ 时产生内部复位信号并且当 $V_{DD} \geq V_{LVI}$ 时释放复位信号。	当 $V_{DD}$ 降落至比 $V_{LVI}$ 低 ( $V_{DD} < V_{LVI}$ ) 时产生内部中断信号或者当 $V_{DD}$ 变成 $V_{LVI}$ 或更高 ( $V_{DD} \geq V_{LVI}$ ) 时。	当 $EXLVI < V_{EXLVI}$ 时产生内部复位信号并且当 $EXLVI \geq V_{EXLVI}$ 时释放复位信号。	当 $EXLVI$ 降落至比 $V_{EXLVI}$ 低 ( $EXLVI < V_{EXLVI}$ ) 时产生内部中断信号或者当 $EXLVI$ 变成 $V_{EXLVI}$ 或更高 ( $EXLVI \geq V_{EXLVI}$ ) 时。

**备注** LVISEL: 低电压检测寄存器 (LVIM) 的位 2  
LVIMD: LVIM 的位 1

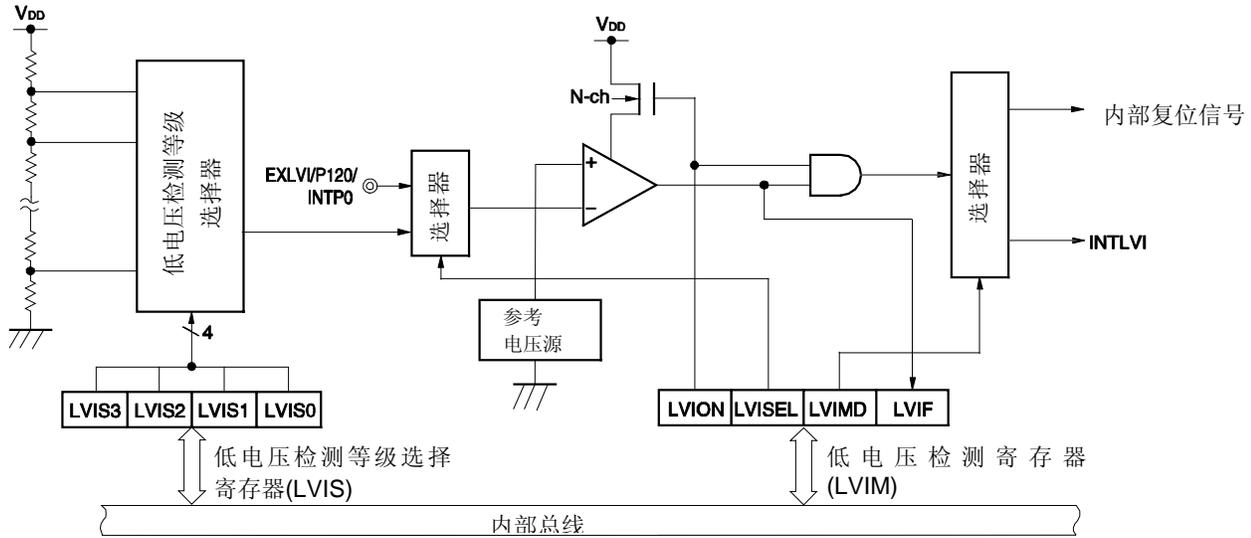
当低电压检测时，供电电压或者外部输入引脚的输入电压可以通过读取低电压检测标志 (LVIM 的位 0: LVIF) 得知其是否高于或低于检测电平。

当低电压检测电路用作产生复位时，如果复位产生，复位控制标志寄存器 (RESF) 的第 0 位 (LVIRF) 被置 1。需要了解 RESF 的详细信息，请参见 **第二十章 复位功能**。

## 22.2 低电压检测电路的配置

低电压检测电路的框图如图 22-1 所示。

图 22-1. 低电压检测电路的框图



## 22.3 控制低电压检测电路的寄存器

低电压检测电路由以下寄存器控制。

- 低电压检测寄存器 (LVIM)
- 低电压检测等级选择寄存器 (LVIS)
- 端口模式寄存器 12(PM12)

### (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。

复位信号的产生将 LVIM 设置为 00H。

图 22-2. 低电压检测寄存器 (LVIM) 的格式

地址 : FFBEH 复位后 : 00H<sup>注1</sup> R/W<sup>注2</sup>

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

LVION <sup>注3,4</sup>	允许低电压检测操作
0	禁止操作
1	允许操作

LVISEL <sup>注3</sup>	电压检测的选择
0	检测供电电压(V <sub>DD</sub> )的等级
1	检测外部输入引脚的输入电压(EXLVI)的等级

LVIMD <sup>注3</sup>	低电压检测操作模式的选择
0	<ul style="list-style-type: none"> <li>LVISEL = 0: 当供电电压(V<sub>DD</sub>) &lt; 检测电压 (V<sub>LVI</sub>), 产生中断信号</li> <li>LVISEL = 1: 当外部输入引脚的输入电压(EXLVI) &lt; 检测电压 (V<sub>LVI</sub>), 产生中断信号</li> </ul>
1	<ul style="list-style-type: none"> <li>LVISEL = 0: 当供电电压(V<sub>DD</sub>) &lt; 检测电压 (V<sub>LVI</sub>), 产生内部复位信号</li> <li>LVISEL = 1: 当外部输入引脚的输入电压(EXLVI) &lt; 检测电压 (V<sub>LVI</sub>), 产生内部复位信号</li> </ul>

LVIF	低电压检测标志
0	<ul style="list-style-type: none"> <li>LVISEL = 0: 供电电压 (V<sub>DD</sub>) ≥ 检测电压 (V<sub>LVI</sub>) 或当禁止操作时</li> <li>LVISEL = 1: 外部输入引脚的输入电压(EXLVI) ≥ 检测电压 (V<sub>EXLVI</sub>), 或当禁止操作时</li> </ul>
1	<ul style="list-style-type: none"> <li>LVISEL = 0: 供电电压 (V<sub>DD</sub>) &lt; 检测电压 (V<sub>LVI</sub>)</li> <li>LVISEL = 1: 外部输入引脚的输入电压(EXLVI) &lt; 检测电压(V<sub>EXLVI</sub>)</li> </ul>

- 注
- 在除了 LVI 之外的复位后, 此位清零 00H。
  - 第 0 位只读。
  - 在产生除 LVI 复位以外的复位时, LVION、LVIMD 和 LVISEL 被清零(0)。在产生 LVI 复位时这些位不被清零。
  - 当 LVION=1 时, LVI 电路中的比较器开始操作。从设置 LVION=1 到操作稳定, 使用软件等待操作稳定时间(10 μs (MAX.))。操作稳定后, 从进入低于 LVI 检测电压状态到 LVIF 置 1(1)需要 200 μs (MIN.) 时间。

- 注意事项
- 在执行以下任一过程后, 停止 LVI。
    - 当使用 8 位操作指令时: 00H → LVIM。
    - 当使用 1 位存储器操作指令时: 对 LVION 清零。
  - 外部输入引脚的输入电压(EXLVI)必须为: EXLVI < V<sub>DD</sub>。
  - 当把 LVI 用作中断, 如果在低于 LVI 检测电压状态下把 LVION 清零, 那么产生 INTLVI 信号, 并且 LVIIF 变为 1。

**(2) 低电压检测等级选择寄存器 (LVIS)**

该寄存器用于选择低电压检测等级。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号将 LVIS 设置为 00H。

图 22-3. 低电压检测等级选择寄存器 (LVIS) 的格式

地址：FFBFH    复位后：00H<sup>※</sup>R/W

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

LVIS3	LVIS2	LVIS1	LVIS0	检测电平
0	0	0	0	V <sub>LV10</sub> (4.24 V ±0.1 V)
0	0	0	1	V <sub>LV11</sub> (4.09 V ±0.1 V)
0	0	1	0	V <sub>LV12</sub> (3.93 V ±0.1 V)
0	0	1	1	V <sub>LV13</sub> (3.78 V ±0.1 V)
0	1	0	0	V <sub>LV14</sub> (3.62 V ±0.1 V)
0	1	0	1	V <sub>LV15</sub> (3.47 V ±0.1 V)
0	1	1	0	V <sub>LV16</sub> (3.32 V ±0.1 V)
0	1	1	1	V <sub>LV17</sub> (3.16 V ±0.1 V)
1	0	0	0	V <sub>LV18</sub> (3.01 V ±0.1 V)
1	0	0	1	V <sub>LV19</sub> (2.85 V ±0.1 V)
1	0	1	0	V <sub>LV110</sub> (2.70 V ±0.1 V)
1	0	1	1	V <sub>LV111</sub> (2.55 V ±0.1 V)
1	1	0	0	V <sub>LV112</sub> (2.39 V ±0.1 V)
1	1	0	1	V <sub>LV113</sub> (2.24 V ±0.1 V)
1	1	1	0	V <sub>LV114</sub> (2.08 V ±0.1 V)
1	1	1	1	V <sub>LV115</sub> (1.93 V ±0.1 V)

注     由 LVI 引起的复位后，LVIS 的数值并不复位，保留其数值。其它原因引起的复位将其清零 00H。

- 注意事项
1. 第 4 位 ~ 第 7 位必须清零。
  2. LVI 操作期间不要修改 LVIS 的值。
  3. 当检测到外部输入引脚的输入电压(EXLVI)时，检测电压(V<sub>EXLVI</sub> = 1.21 V (TYP.))恒定。因此，不必设置 LVIS。

**(3) 端口模式寄存器 12 (PM12)**

当使用 P120/EXLVI/INTP0 引脚进行外部低电压检测电压输入时，设置 PM120=1。此时，P120 的输出锁存器的值可以为 0 或 1。

可由 1 位或 8 位存储器操作指令设置 PM12。

复位信号的产生将 PM12 设置为 FFH。

图 22-4. 端口模式寄存器 12 (PM12)的格式

地址：FF2CH      复位后：FFH R/W

符号：            7            6            5            4            3            2            1            0

PM12	1	1	1	1	1	1	1	PM120
------	---	---	---	---	---	---	---	-------

PM120	P120 引脚 I/O 模式选择
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

**22.4 低电压检测电路的操作**

低电压检测电路能够用于以下两种模式。

**(1) 用于复位**

- 如果 LVISEL = 0，比较供电电压( $V_{DD}$ )与检测电压( $V_{LVI}$ )，当  $V_{DD} < V_{LVI}$  时产生内部复位信号，而当  $V_{DD} \geq V_{LVI}$  时释放内部复位。
- 如果 LVISEL = 1，比较外部输入引脚的输入电压(EXLVI)和检测电压( $V_{EXLVI} = 1.21 \text{ V (TYP.)}$ )，在  $EXLVI < V_{EXLVI}$  时产生内部复位信号，而在  $EXLVI \geq V_{EXLVI}$  时释放内部复位。

**(2) 用于中断**

- 如果 LVISEL = 0，比较供电电压( $V_{DD}$ )与检测电压( $V_{LVI}$ )，当  $V_{DD} < V_{LVI}$  时产生中断信号(INTLVI)。
- 如果 LVISEL = 1，比较外部输入引脚的输入电压(EXLVI)和检测电压( $V_{EXLVI} = 1.21 \text{ V (TYP.)}$ )，当  $EXLVI < V_{EXLVI}$  时产生中断信号(INTLVI)。

**备注**      LVIMD:低电压检测寄存器 (LVIM)的位 1。  
 LVISEL: LVIM 的位 2。

### 22.4.1 用于复位

#### (1) 当检测供电电压(V<sub>DD</sub>)等级时

- 启动操作时
  - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
  - <2> 将低电压检测寄存器(LVIM)的第 2 位(LVISEL)清零(检测供电电压(V<sub>DD</sub>)的等级)(默认值)。
  - <3> 使用低电压检测等级选择寄存器 (LVIS) 的第 3 位~第 0 位 (LVIS3 ~ LVIS0) 设置检测电压。
  - <4> 设置 LVIM 的第 7 位 (LVION) = 1 (允许 LVI 的操作)。
  - <5> 使用软件等待操作稳定时间(10  $\mu$ s (MAX.))。
  - <6> 等待直至根据 LVIM 的第 0 位(LVIF)检测到供电电压(V<sub>DD</sub>)  $\geq$  检测电压(V<sub>LVI</sub>)。
  - <7> 设置 LVIM 的第 1 位 (LVIMD) = 1 (当供电电压 (V<sub>DD</sub>) < 检测电压 (V<sub>LVI</sub>) 时产生内部复位信号)。

图 22-5 显示了由低电压检测电路产生的内部复位信号的时序。该时序图中的标号与上面的<1> ~ <7>对应。

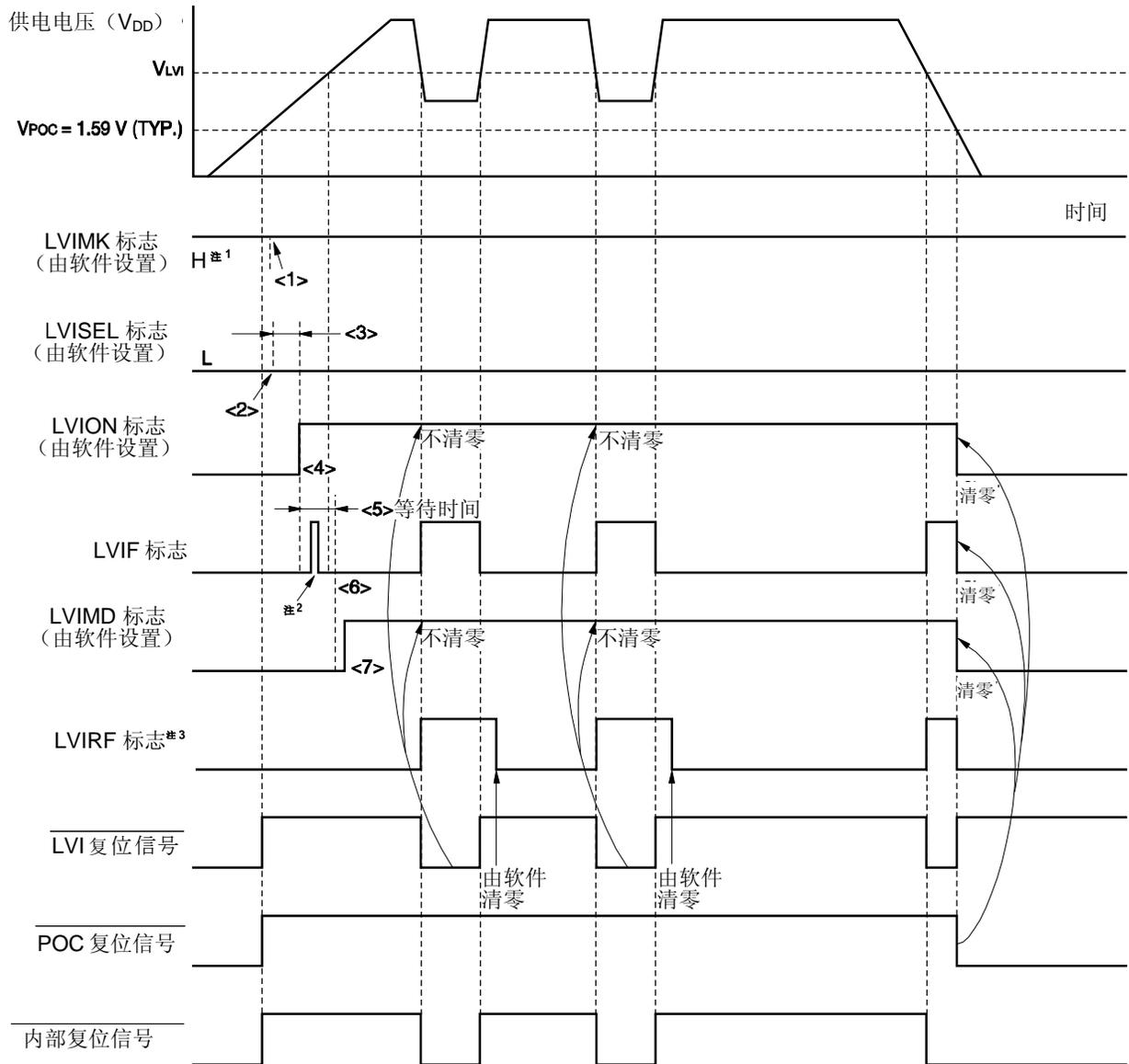
**注意事项 1.** 必须执行过程<1>。当 LVIMK = 0 时，在过程<4>之后可能立即会产生中断。

**2.** 当 LVIMD=1 时如果供电电压 (V<sub>DD</sub>)  $\geq$  检测电压 (V<sub>LVI</sub>)，则不会产生内部复位信号。

- 当停止操作时  
必须执行以下过程之一。
  - 当使用 8 位存储器操作指令时:  
00H  $\rightarrow$  LVIM
  - 当使用 1 位存储器操作指令时:  
依次将 LVIMD 和 LVION 清零。

图 22-5. 低电压检测电路内部复位信号产生时序  
(检测供电电压(V<sub>DD</sub>)等级) (1/2)

(1) 1.59 V POC 模式 (选项字节: POCMODE = 0)

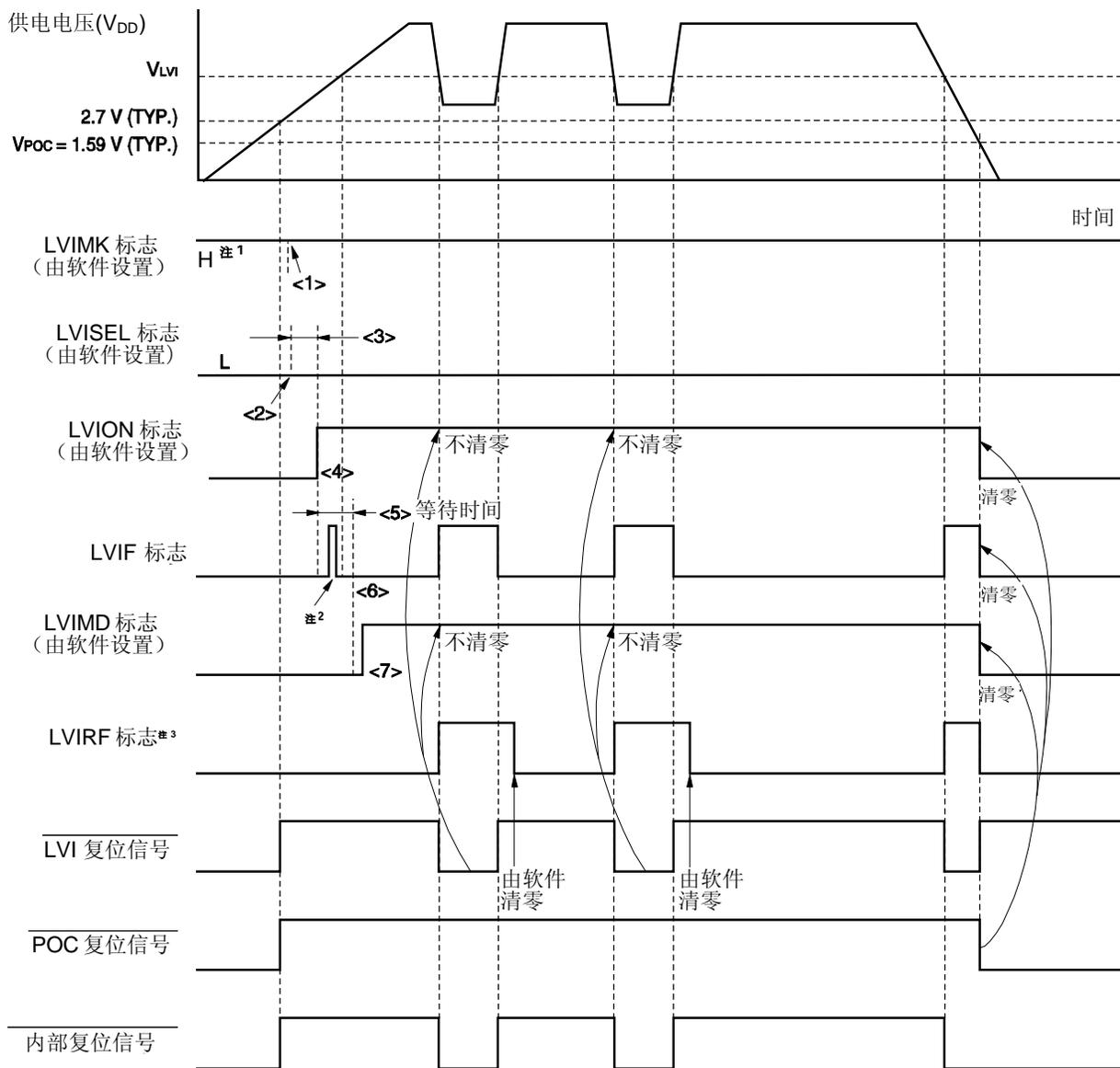


- 注
1. 由复位信号产生将 LVIMK 标志设置为“1”。
  2. 可以设置 LVIF 标志为(1)。
  3. LVIRF 是复位控制标志寄存器 (RESF) 的第 0 位。需要了解 RESF 的详细内容, 可参见第二十章 复位功能。

备注 图 22-5 中的<1> ~ <7>与 22.4.1(1) 当检测供电电压(V<sub>DD</sub>)等级时中“当启动操作时”的描述<1> ~ <7>相对应。

图 22-5. 低电压检测电路内部复位信号产生时序  
(检测供电电压(V<sub>DD</sub>)等级) (2/2)

(2) 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1)



- 注
1. 由复位信号产生将 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, 可参见第二十章 复位功能。

备注 图 22-5 中的<1> ~ <7>与 22.4.1(1) 当检测供电电压(V<sub>DD</sub>)等级时中“当启动操作时”的描述<1> ~ <7>相对应。

**(2) 当检测外部输入引脚的输入电压(EXLVI)等级时**

- 当启动操作时
  - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
  - <2> 设置低电压检测寄存器(LVIM)的第 2 位(LVISEL)=1 (检测外部输入引脚的输入电压 (EXLVI) 等级)。
  - <3> 设置 LVIM 的第 7 位 (LVION) =1 (允许 LVI 的操作)。
  - <4> 使用软件等待操作稳定时间(10  $\mu$ s (MAX.)<sup>#</sup>)。
  - <5> 等待直至根据 LVIM 的第 0 位(LVIF)检测到: 外部输入引脚的输入电压(EXLVI)  $\geq$  检测电压( $V_{EXLVI} = 1.21$  V (TYP.))
  - <6> 设置 LVIM 的第 1 位(LVIMD)=1(当外部输入引脚的输入电压(EXLVI) < 检测电压( $V_{EXLVI} = 1.21$  V (TYP.))时产生内部复位信号)。

图 22-6 显示了由低电压检测电路产生的内部复位信号的时序。该时序图中的标号与上面的<1> ~ <6>对应。

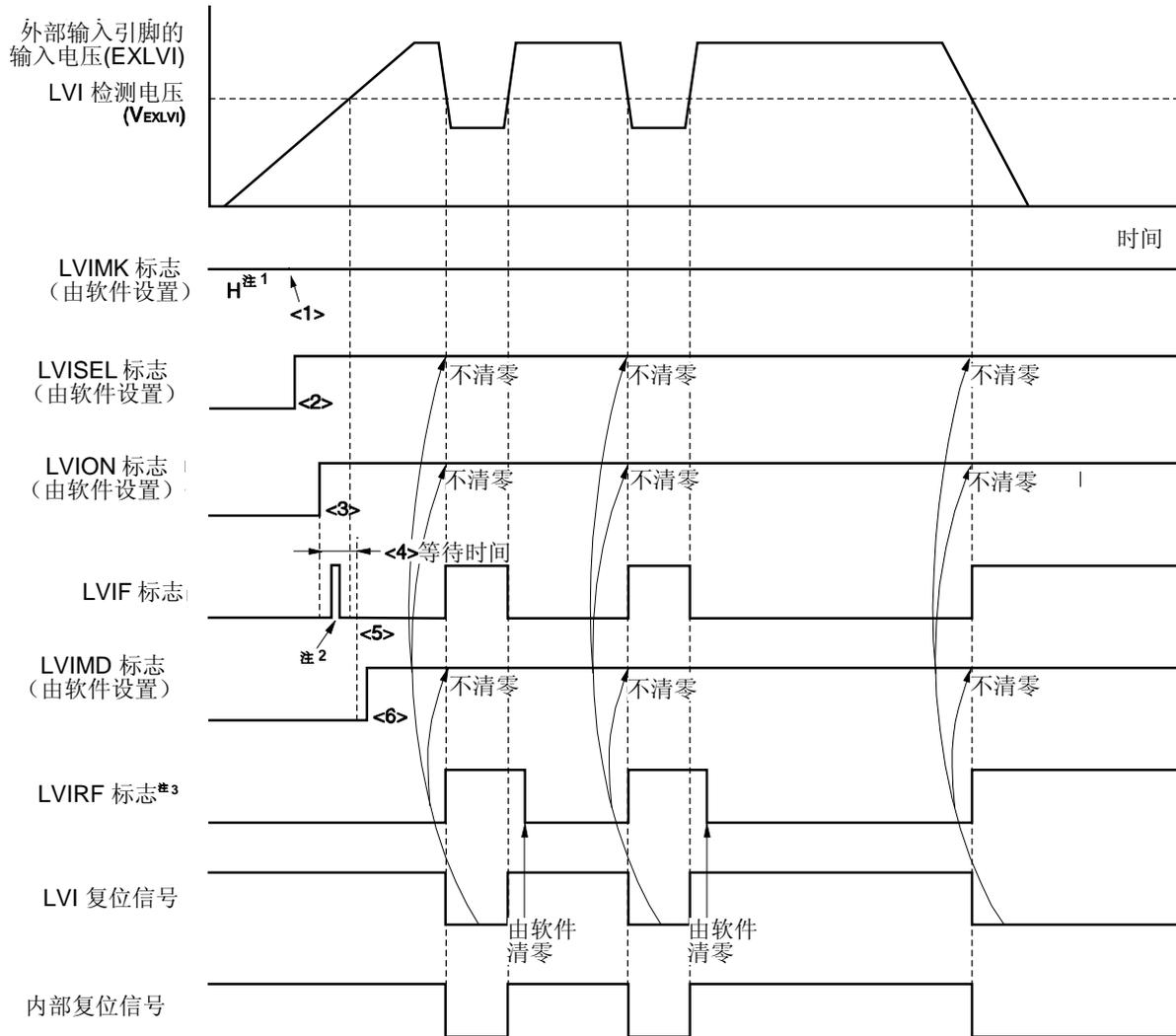
**注意事项 1. 必须执行<1>。当 LVIMK = 0 时，在完成<3>后可能立即会产生中断。**

**2. 如果在 LVIMD=1 时，外部输入引脚的输入电压(EXLVI)  $\geq$  检测电压( $V_{EXLVI} = 1.21$  V (TYP.))，则不会产生内部复位信号。**

**3. 必须保证外部输入引脚的输入电压(EXLVI) :  $EXLVI < V_{DD}$**

- 当停止操作时  
必须执行以下过程之一。
  - 当使用 8 位存储器操作指令:  
00H  $\rightarrow$  LVIM
  - 当使用 1 位存储器操作指令:  
先对 LVIMD 清零(0)，再对 LVION 清零(0)。

图 22-6. 低电压检测电路内部复位信号产生时序  
(检测外部输入引脚的输入电压(EXLVI)的等级)



- 注
1. 由复位信号的产生设置 LVIMK 标志为“1”。
  2. 可以设置 LVIF 标志为(1)。
  3. LVIRF 为复位控制标志寄存器(RESF)的第 0 位。需要了解 RESF 的详细信息，可参见第二十章 复位功能。

备注 图 22-6 中的<1> ~ <6>与 22.4.1 (2) 当检测外部输入引脚的输入电压(EXLVI)等级时中“当启动操作时”的描述<1> ~ <6>相对应

## 22.4.2 用于中断

(1) 当检测供电电压(V<sub>DD</sub>)等级时

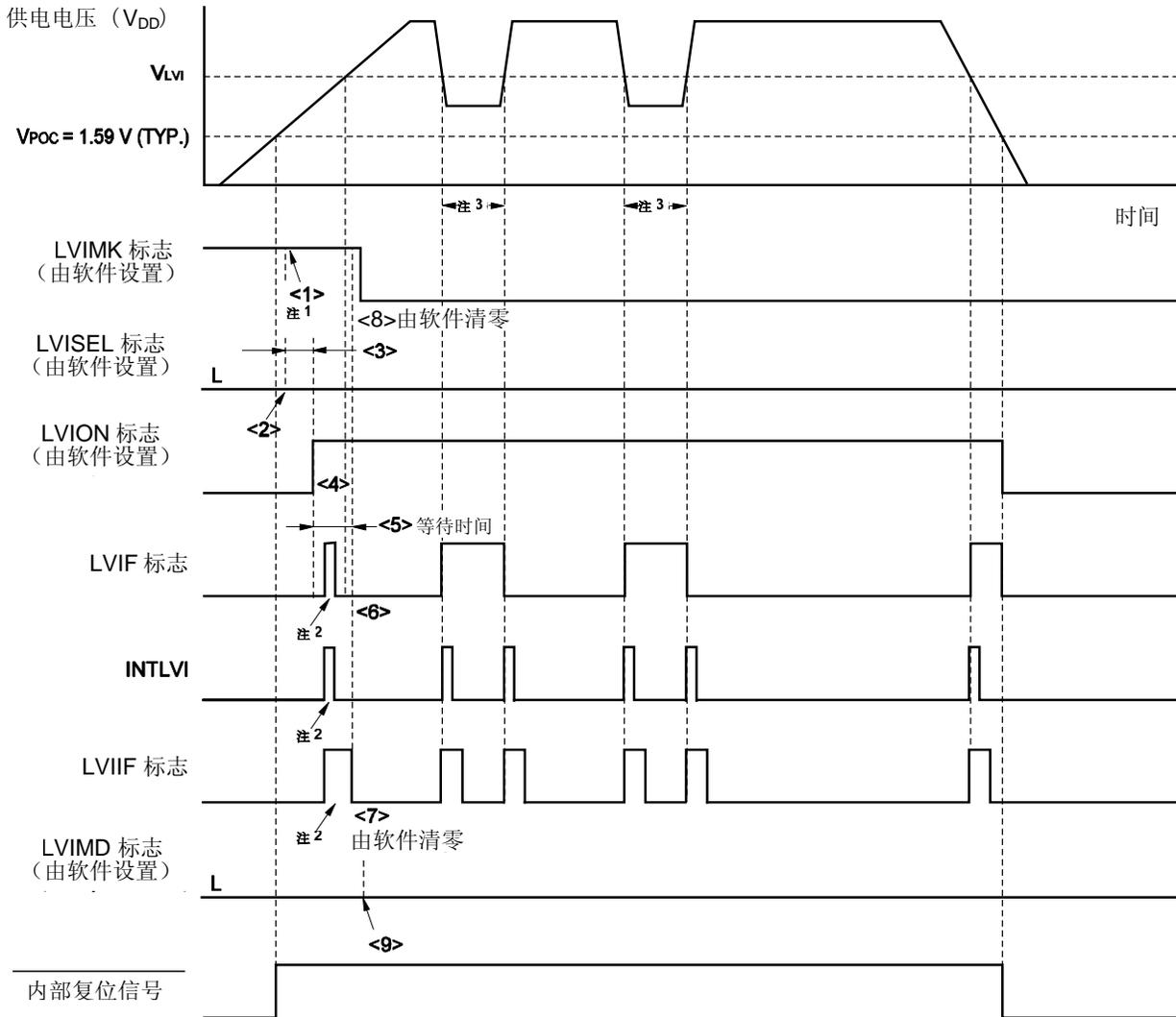
- 当启动操作时
  - <1> 屏蔽 LVI 中断 (LVIMK = 1)
  - <2> 将低电压检测寄存器(LVIM)的第 2 位(LVISEL)清零(0)(检测供电电压(V<sub>DD</sub>)等级)(默认值)。
  - <3> 使用低电压检测等级选择寄存器 (LVIS) 的第 3 位~第 0 位 (LVIS3 ~ LVIS0) 设置检测电压。
  - <4> 设置 LVIM 的第 7 位 (LVION) =1 (允许 LVI 的操作)。
  - <5> 使用软件等待操作稳定时间(10 $\mu$ s (MAX.))。
  - <6> 在 LVIM 的第 0 位(LVIF), 确定“供电电压(V<sub>DD</sub>)  $\geq$  检测电压(V<sub>LVI</sub>)”。
  - <7> 将 LVI 的中断请求标志 (LVIIF) 清零(0)。
  - <8> 释放 LVI 的中断屏蔽标志 (LVIMK) 。
  - <9> 将 LVIM 的第 1 位(LVIMD)清零(0)(当供电电压(V<sub>DD</sub>) < 检测电压(V<sub>LVI</sub>)时产生中断信号)(默认值)。
  - <10> 执行 EI 指令 (当使用向量中断时)。

图 22-7 显示了由低电压检测电路产生的中断信号的时序。该时序图中的标号与上面的<1> ~ <9>对应。

- 当停止操作时  
必须执行以下过程之一。
  - 当使用 8 位存储器操作指令:  
00H  $\rightarrow$  LVIM
  - 当使用 1 位存储器操作指令:  
对 LVION 清零 (0)

图 22-7. 低电压检测电路中断信号产生时序  
(检测供电电压(V<sub>DD</sub>)的等级) (1/2)

(1) 1.59 V POC 模式 (选项字节: POCMODE = 0)

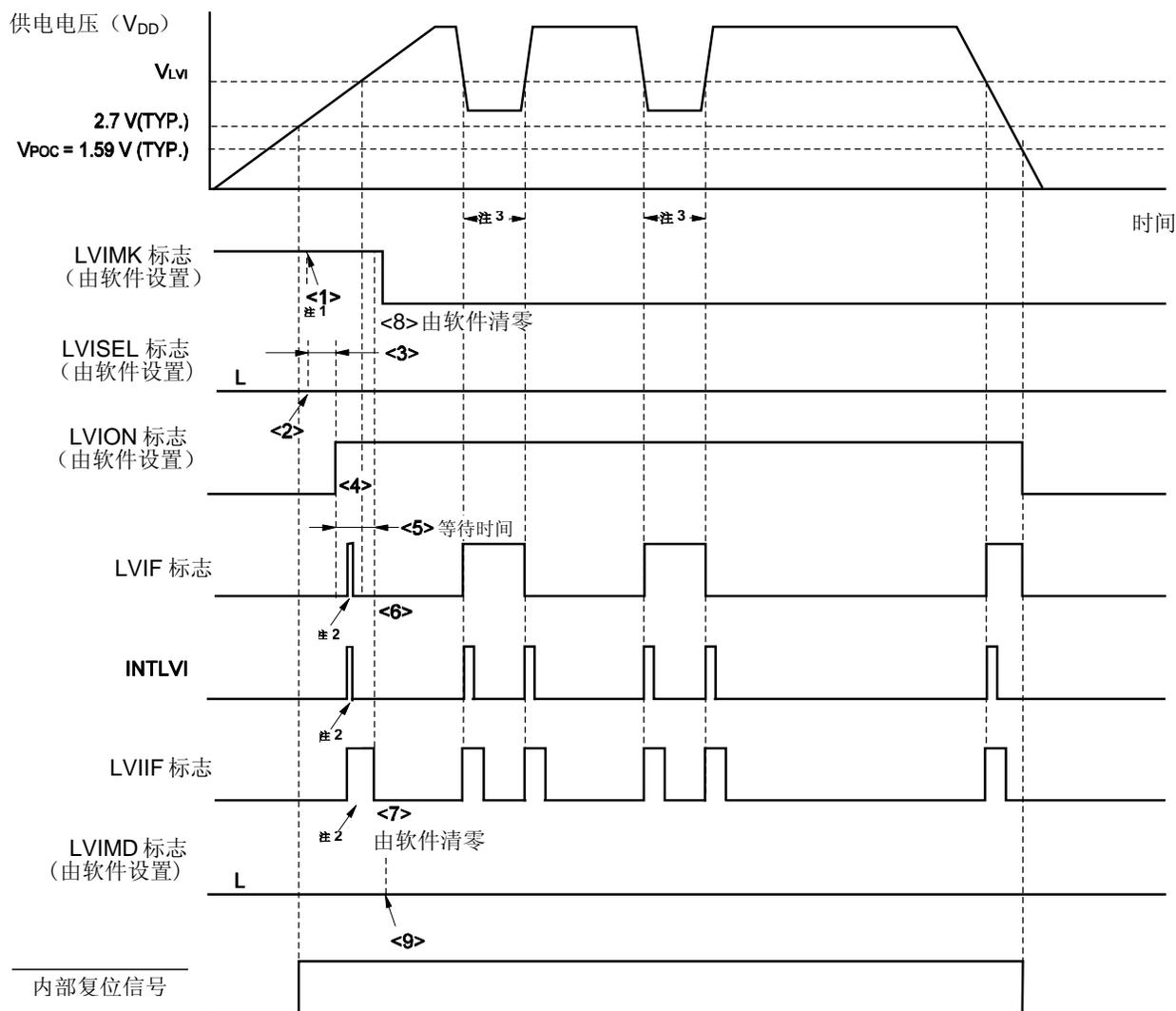


- 注
1. 可由复位信号的产生设置 LVIMK 标志为“1”。
  2. 产生中断请求信号(INTLVI), 且可以设置 LVIF 和 LVIIF 标志为(1)。
  3. 如果在低于 LVI 检测电压状态下把 LVION 清零(0), 那么产生 INTLVI 信号, 并且 LVIIF 变成 1。

备注 图 22-7 中的<1> ~ <9>与 22.4.2 (1) 当检测供电电压(V<sub>DD</sub>)等级时中“当启动操作时”的描述<1> ~ <9>相对应。

图 22-7. 低电压检测电路中中断信号产生时序  
(检测供电电压(V<sub>DD</sub>)的等级) (2/2)

(2) 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1)



- 注
1. 可由复位信号的产生设置 LVIMK 标志为“1”。
  2. 产生中断请求信号(INTLVI)，且可以设置 LVIF 和 LVIIF 标志为(1)。
  3. 如果在低于 LVI 检测电压状态下把 LVION 清零(0)，那么产生 INTLVI 信号，并且 LVIIF 变成 1。

备注 图 22-7 中的<1> ~ <9>与 22.4.2 (1) 当检测供电电压(V<sub>DD</sub>)等级时中“当启动操作时”的描述<1> ~ <9>相对应。

**(2) 当检测外部输入引脚的输入电压(EXLVI)等级时**

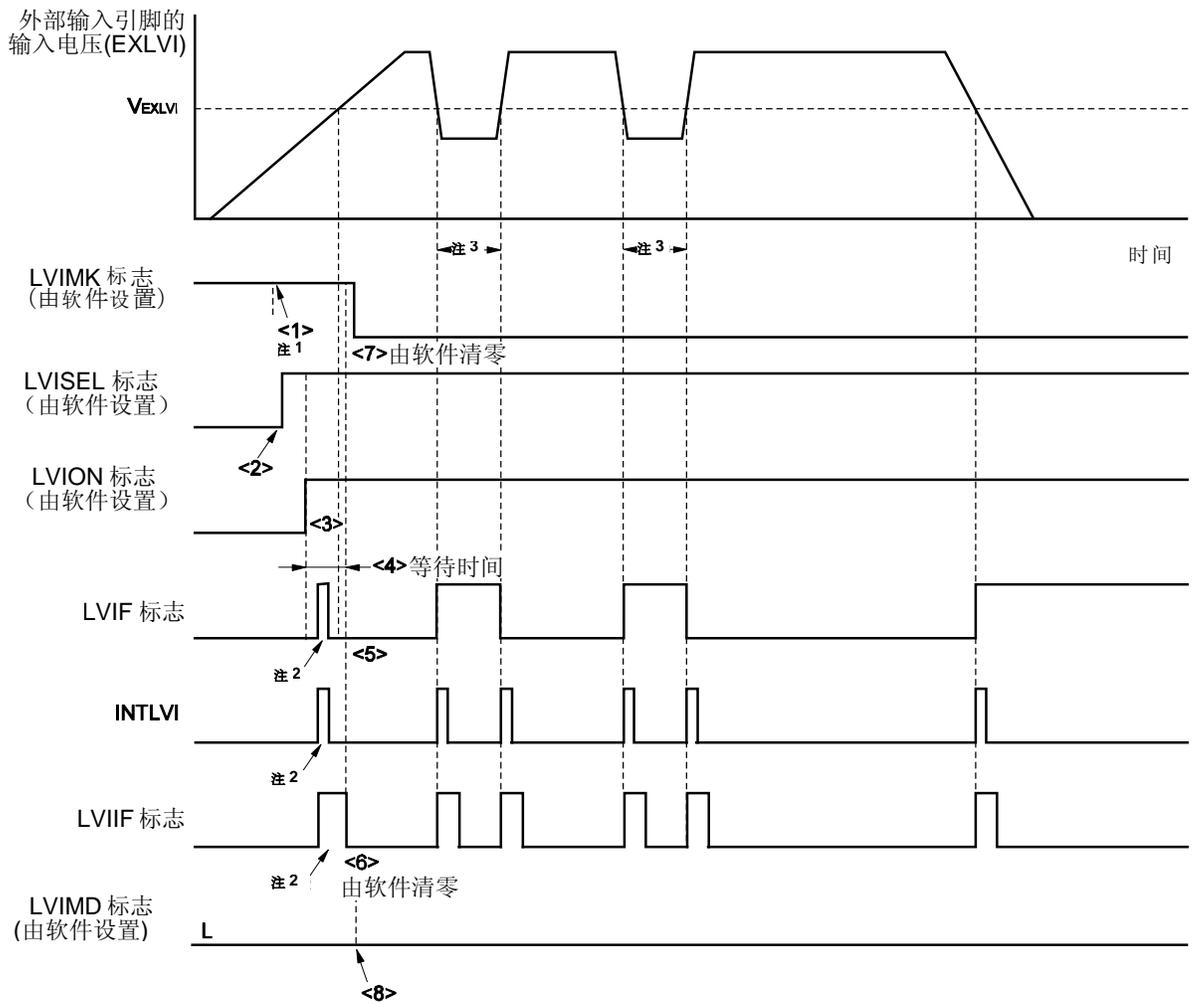
- 当启动操作时
  - <1> 屏蔽 LVI 中断 (LVIMK = 1)
  - <2> 设置低电压检测寄存器(LVIM)的第 2 位(LVISEL)=1(检测外部输入引脚的输入电压(EXLVI)等级)。
  - <3> 设置 LVIM 的第 7 位 (LVION) =1 (允许 LVI 的操作)。
  - <4> 使用软件等待操作稳定时间(10 $\mu$ s (MAX.))。
  - <5> 在 LVIM 的第 0 位(LVIF), 确定“外部输入引脚的输入电压(EXLVI)  $\geq$  检测电压( $V_{EXLVI} = 1.21 \text{ V (TYP.)}$ )”。
  - <6> 将 LVI 的中断请求标志 (LVIIF) 清零(0)。
  - <7> 释放 LVI 的中断屏蔽标志 (LVIMK)。
  - <8> 将 LVIM 的第 1 位(LVIMD)清零(0)(当供电电压( $V_{DD}$ ) < 检测电压( $V_{LVI}$ )时产生中断信号)(默认值)。
  - <9> 执行 EI 指令 (当使用向量中断时)。

图 22-8 显示了由低电压检测电路产生的中断信号的时序。该时序图中的标号与上面的<1> ~ <8>对应。

**注意事项** 外部输入引脚的输入电压(EXLVI)必须满足以下条件:  $EXLVI < V_{DD}$

- 当停止操作时  
必须执行以下过程之一。
  - 当使用 8 位存储器操作指令:  
00H  $\rightarrow$  LVIM
  - 当使用 1 位存储器操作指令:  
对 LVION 清零 (0)

图 22-8. 低电压检测电路中断信号产生时序  
(检测外部输入引脚的输入电压(EXLVI)等级)



- 注
1. 可由复位信号产生设置 LVIMK 标志为“1”。
  2. 产生中断请求信号(INTLVI)，且可以设置 LVIF 和 LVIIF 标志为(1)。
  3. 如果在低于 LVI 检测电压状态下把 LVION 清零(0)，那么产生 INTLVI 信号，并且 LVIIF 变成 1。

备注 图 22-8 中的<1> ~ <8>与 22.4.2 (1) 当检测供电电压( $V_{DD}$ )等级时中“当启动操作时”的描述<1> ~ <8>相对应。

## 22.5 低电压检测电路的注意事项

在系统中，如果供电电压（V<sub>DD</sub>）在接近 LVI 检测电压（V<sub>LVI</sub>）的一段时间内产生波动，这时根据低电压检测电路的使用情况进行如下操作。

### (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）中的(b)。

本系统中采用以下方法。

<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）等待系统的供电电压的波动期，然后对端口初始化(参见图 22-9)。

### (2) 用于中断

(a) 在 LVI 中断的服务程序中通过使用低电压检测寄存器（LVIM）的第 0 位(LVIF)，检测“供电电压(V<sub>DD</sub>) ≥ 检测电压(V<sub>LVI</sub>)”。把中断请求标志寄存器 0L(IFOL)的第 0 位(LVIIF)清零。

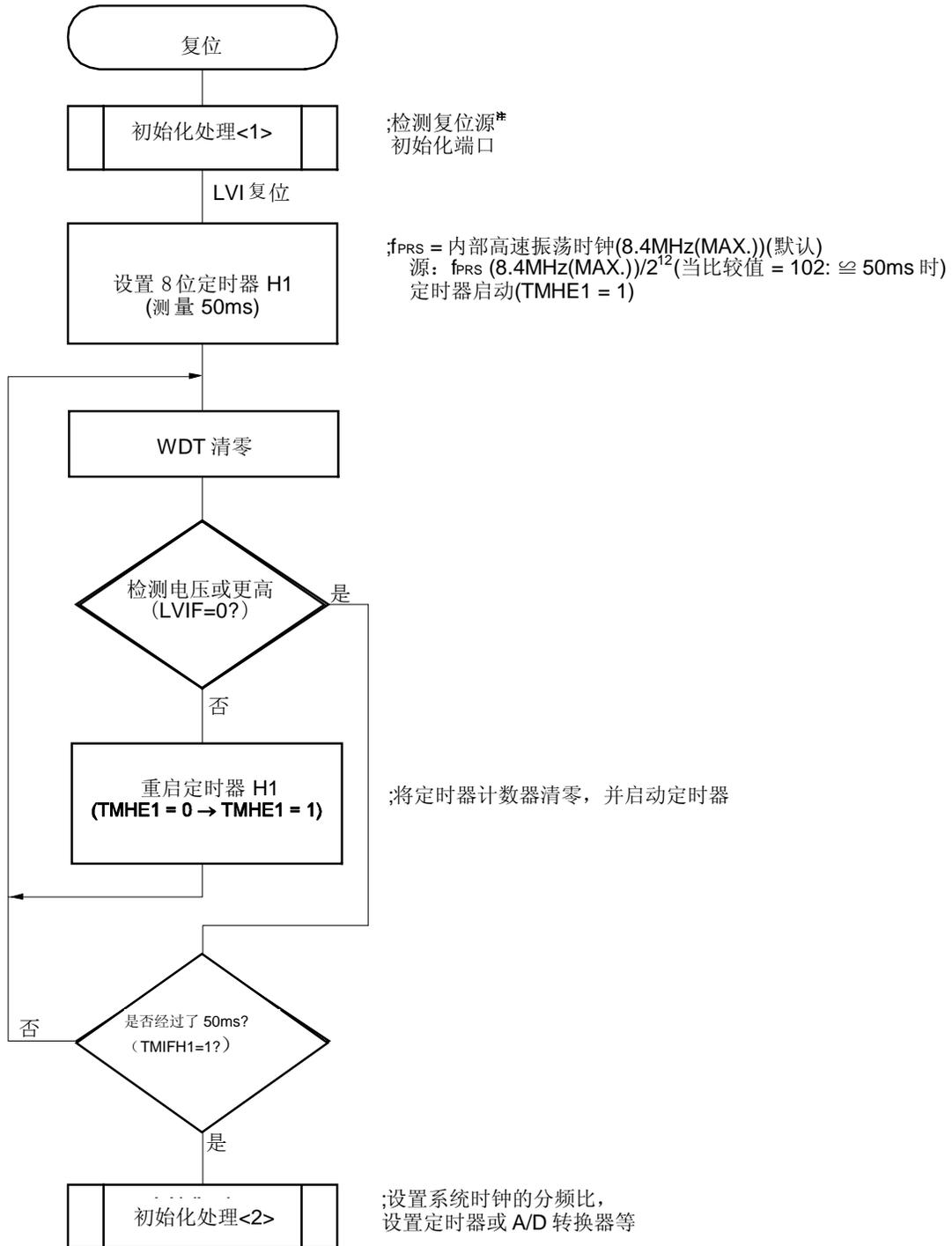
(b) 在系统中，如果供电电压（V<sub>DD</sub>）在接近 LVI 检测电压（V<sub>LVI</sub>）的一段时间内产生波动，等待供电电压波动期间，使用 LVIF 标志来检测“供电电压(V<sub>DD</sub>) ≥ 检测电压(V<sub>LVI</sub>)”，然后把 LVIIF 标志清零。

**备注** 如果设置低电压检测寄存器(LVIM)的第 2 位(LVISEL)为“1”，则对上述方法作如下修改。

- 供电电压(V<sub>DD</sub>) → 外部输入引脚的输入电压(EXLVI)
- 检测电压(V<sub>LVI</sub>) → 检测电压(V<sub>EXLVI</sub> = 1.21 V<sup>※</sup>)

图 22-9. 复位释放后软件处理过程示例(1/2)

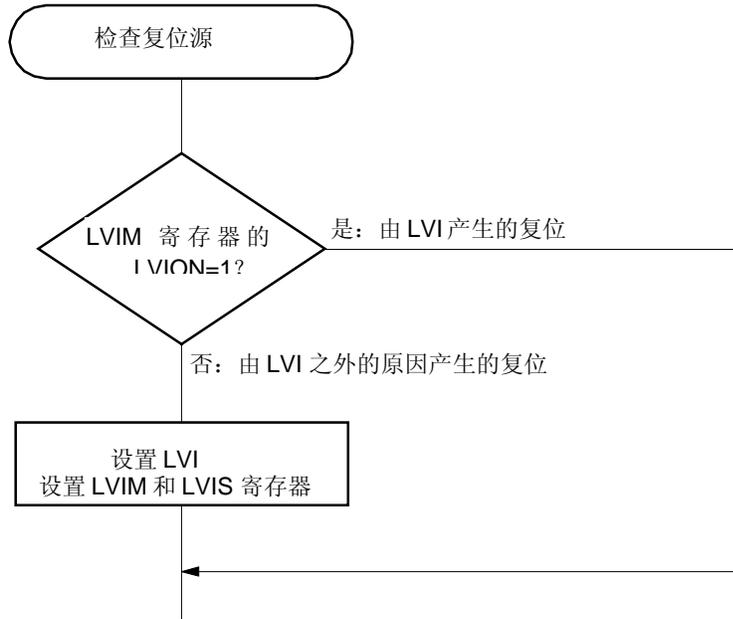
- 如果供电电压接近 LVI 检测电压，其波动时间 $\leq 50$  ms



注 流程图显示在下页

图 22-9. 复位释放后软件处理过程示例(2/2)

•检查复位源



## 第二十三章 选项字节

### 23.1 选项字节的功能

78K0/LC3 中 Flash 存储器的 0080H ~ 0084H 区域为选项字节区域。当打开电源或从复位状态重启设备时，设备自动参考选项字节，并设置指定的功能。使用该产品时，必须使用选项字节设置以下几项功能。

在自编程期间使用引导交换功能时，将 0080H ~ 0084H 切换到 1080H ~ 1084H。因此，事先将 0080H ~ 0084H 和 1080H ~ 1084H 设置为相同的值。

**注意事项** 一定要对 0082H 和 0083H (当使用引导交换功能时为 0082H/1082H 和 0083H/1083H)清零。

#### (1) 0080H/1080H

- 内部低速振荡器的操作
  - 可由软件停止
  - 不能停止
- 看门狗定时器间隔时间的设置
- 看门狗定时器计数器的操作
  - 允许计数器的操作
  - 禁止计数器的操作
- 看门狗定时器窗口打开周期的设置

**注意事项** 将 0080H ~ 1080H 设置成为相同的值，因为在引导交换过程中 0080H ~ 1080H 将被切换。

#### (2) 0081H/1081H

- 选择 POC 模式
  - 2.7 V/1.59 V POC 模式操作期间 (POCMODE = 1)  
上电时该设备处于复位状态，直至供电电压达到 2.7 V (TYP.)。当电压超过 2.7 V (TYP.)时设备从复位状态释放。此后在 2.7 V 不检测 POC，但在 1.59 V (TYP.)检测 POC。  
如果上电后供电电压以低于 0.5 V/ms (MIN.)的速率上升到 1.8 V，建议使用 2.7 V/1.59 V POC 模式。
  - 1.59 V POC 模式操作期间 (POCMODE = 0)  
上电时该设备处于复位状态，直至供电电压达到 1.59 V (TYP.)。当电压超过 1.59 V (TYP.)时设备从复位状态释放。此后采用与上电时相同的方法在 1.59 V (TYP.)检测 POC。

**注意事项** POCMODE 值能通过专用的 Flash 存储器编程器写入。在自编程或是自编程的引导交换操作中不能被设定 (此时，设置 1.59 V POC 模式(缺省))。但是由于在引导交换过程中 1081H 是 0081H 的拷贝，所以在引导交换过程中，推荐将 0081H 和 1081H 设定为相同的值。

**(3) (3) 0084H/1084H**

○ 片上调试操作控制

- 禁止片上调试操作
- 一旦证明片上调试安全 ID 失效时，允许片上调试操作和擦除 Flash 存储器数据。
- 一旦证明片上调试安全 ID 失效时，允许片上调试操作，但不擦除 Flash 存储器数据。

**注意事项** 为了使用引导交换功能，需要对 0084H 单元赋值 02H 或 03H。由于在引导交换时 0084H 和 1084H 要相互切换，因此也要对 1084H 设置与 0084H 相同的值。

## 23.2 选项字节的格式

选项字节格式如下所示。

图 23-1. 选项字节的格式 (1/2)

地址: 0080H/1080H<sup>注</sup>

7	6	5	4	3	2	1	0
0	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	LSROSC
WINDOW1	WINDOW0	看门狗定时器窗口打开周期					
0	0	25%					
0	1	50%					
1	0	75%					
1	1	100%					
WDTON	看门狗定时器计数器的操作控制/非法访问检测						
0	禁止计数器操作 (复位后停止计数), 禁止非法访问检测操作						
1	允许计数器操作 (复位后开始计数), 允许非法访问检测操作						
WDCS2	WDCS1	WDCS0	看门狗定时器溢出时间				
0	0	0	$2^{10}/f_{RL}$ (3.88 ms)				
0	0	1	$2^{11}/f_{RL}$ (7.76 ms)				
0	1	0	$2^{12}/f_{RL}$ (15.52 ms)				
0	1	1	$2^{13}/f_{RL}$ (31.03 ms)				
1	0	0	$2^{14}/f_{RL}$ (62.06 ms)				
1	0	1	$2^{15}/f_{RL}$ (124.12 ms)				
1	1	0	$2^{16}/f_{RL}$ (248.24 ms)				
1	1	1	$2^{17}/f_{RL}$ (496.48 ms)				
LSROSC	内部低速振荡器操作						
0	可由软件停止 (当将 1 写入 RCM 寄存器的第 0 位(LSRSTOP)时停止操作)						
1	不能停止 (即使将 1 写入 LSRSTOP 位也不停止操作)						

注 由于在引导交换时 0080H 和 1080H 要相互切换, 因此要对 0080H 设置与 1080H 相同的值。

- 注意事项
1. 禁止设置:  $WDCS2 = WDCS1 = WDCS0 = 0$  且  $WINDOW1 = WINDOW0 = 0$ 。
  2. Flash 存储器自编程和模拟 EEPROM 期间看门狗定时器继续工作。处理过程中中断响应时间将被延迟。设置溢出时间和窗口大小时应考虑到这一延迟情况。
  3. 如果  $LSROSC = 0$  (可由软件停止振荡), 则无论内部振荡模式寄存器(RCM)的第 0 位(LSRSTOP)为何值, 在 HALT 和 STOP 模式下对看门狗定时器不提供计数时钟。  
当 8 位定时器 H1 使用内部低速振荡时钟时, 即使在 HALT/STOP 模式下也对 8 位定时器 H1 提供计数时钟。
  4. 必须对第 7 位清零(0)。

- 备注
1.  $f_{RL}$ : 内部低速振荡时钟频率
  2. ( ):  $f_{RL} = 264$  kHz (最大)

图 23-1. 选项字节的格式 (2/2)

地址: 0081H/1081H<sup>※1,2</sup>

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	POCMODE

POCMODE	POC 模式选择
0	1.59 V POC 模式 (默认)
1	2.7 V/1.59 V POC 模式

- 注**
1. 使用专用 Flash 存储器编程器只能对 POCMODE 进行写操作。在自编程期间不能设置 POCMODE，或者自编程引导交换(此时，设置 1.59 V POC 模式(默认))操作时也不能设置。但由于引导交换时 1081H 单元的值被复制到 0081H 单元，建议在使用引导交换功能时为 1081H 单元设置与 0081H 相同的值。
  2. 要改变 POC 模式的设置，可在对 Flash 存储器进行一次性擦除(片擦除)后将该值再次写入 0081H 中。在对 Flash 存储器指定 block 擦除后不能修改该设置。

**注意事项** 必须将第 7 位 ~ 第 1 位清零。

地址: 0082H/1082H, 0083H/1083H<sup>※</sup>

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

- 注** 由于 0082H 与 0083H 单元为保留区域，因此必须对这两个存储单元赋值 00H。由于在引导交换时 0082H 与 0083H 要和 1082H 与 1083H 相互切换，因此也要对 1082 与 1083H 设置 00H。

地址: 0084H/1084H<sup>※</sup>

7	6	5	4	3	2	1	0
0	0	0	0	0	0	OCDEN1	OCDEN0

OCDEN1	OCDEN0	片上调试操作控制
0	0	禁止操作
0	1	禁止设置
1	0	允许操作。在确定片上调试安全 ID 失效时不要擦除 Flash 存储器数据。
1	1	允许操作。在确定片上调试安全 ID 失效时可以擦除 Flash 存储器数据。

- 注** 要使用片上调试功能，需要对 0084H 单元赋值 02H 或 03H。由于在引导交换时 0084H 和 1084H 要相互切换，因此也要对 1084H 设置与 0084H 相同的值。

**备注** 如需了解片上调试安全 ID 的详细信息，参见第二十五章 片上调试功能

以下是设置选项字节的软件描述示例。

OPT	CSEG	AT 0080H	
OPTION:	DB	30H	; 允许看门狗定时器的操作 (对非法访问检测的操作), ; 看门狗定时器窗口打开周期: 50%, ; 看门狗定时器的溢出时间: $2^{10}/f_{RL}$ , ; 可由软件停止内部低速振荡器
	DB	00H	; 1.59 V POC 模式
	DB	00H	; 保留区域
	DB	00H	; 保留区域
	DB	00H	; 禁止片上调试

**备注** 复位处理期间参考选项字节。如需了解复位处理的时序，可参见**第二十章 复位功能**。

## 第二十四章 Flash 存储器

78K0/LC3 包含 Flash 存储器，可以安装在电路板上实现在线写入、擦除和覆盖操作。

### 24.1 内部存储器容量切换寄存器

可以使用内部存储器容量切换寄存器（IMS）选择内存容量。

可使用 8 位存储器操作指令设置 IMS。

复位信号的产生将 IMS 设置为 CFH。

**注意事项** 复位释放后必须按照表 24-1 对每个产品进行设置。

图 24-1. 内部存储器容量切换寄存器（IMS）的格式

地址: FFF0H 复位后: CFH R/W

符号

	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM2	RAM1	RAM0	内部高速 RAM 容量选择
0	0	0	768 字节
0	1	0	512 字节
1	1	0	1024 字节
其它情况			禁止设置

ROM3	ROM2	ROM1	ROM0	内部 ROM 容量选择
0	0	1	0	8 KB
0	1	0	0	16 KB
0	1	1	0	24 KB
1	0	0	0	32 KB
其它情况				禁止设置

表 24-1. 内部存储器容量切换寄存器的设置

Flash 存储器版本(78K0/LC3)	IMS 设置	ROM 容量	内部高速 RAM 容量
μPD78F0400, 78F0410	42H	8 KB	512 字节
μPD78F0401, 78F0411	04H	16 KB	768 字节
μPD78F0402, 78F0412	C6H	24 KB	1 K 字节
μPD78F0403, 78F0413	C8H	32 KB	

## 24.2 用 Flash 编程器写入数据

可使用专用 Flash 编程器将数据写入 Flash 存储器，可以 On-board 写入，也可以 Off-board 写入。

### (1) On-board 编程

当目标系统上安装了 78K0/LC3 系列产品后，Flash 存储器的内容可以被重写。必须将连接专用 Flash 编程器的连接器安装在目标系统中。

### (2) Off-board 编程

在将 78K0/LC3 系列产品安装到目标系统之前，可使用专用程序适配器（FA 系列）将数据写入 Flash 存储器。

**备注** FA 系列是 Naito Densai Machida Mfg. Co., Ltd. 的产品。

表 24-2. 78K0/LC3 系列产品与专用 Flash 编程器之间的写操作

专用 Flash 编程器的引脚配置			使用 UART6	
信号名称	I/O	引脚功能	引脚名称	引脚编号
SI/RxD	输入	接收信号	TxD6/SEG6/P112	24
SO/TxD	输出	发送信号	RxD6/SEG7/P113	23
SCK	输出	传输时钟	—	—
CLK	输出	78K0/LC3 的时钟	注 1	注 1
/RESET	输出	复位信号	RESET	6
FLMD0	输出	模式信号	FLMD0	9
V <sub>DD</sub>	I/O	V <sub>DD</sub> 电压产生/ 电压监测	V <sub>DD</sub>	14
			V <sub>DD</sub> <sup>注 2</sup>	35
			AV <sub>REF</sub> <sup>注 3</sup>	
GND	—	地	V <sub>SS</sub>	13
			V <sub>SS</sub> <sup>注 2</sup>	36
			AV <sub>SS</sub> <sup>注 3</sup>	

**注** 1. 在使用 UART6 时，只能使用 X1 时钟(f<sub>x</sub>)或外部主系统时钟(f<sub>EXCLK</sub>)。要使用 Flash 编程器的时钟输出，引脚连接依据所用的专用的 flash 编程器不同而不同。

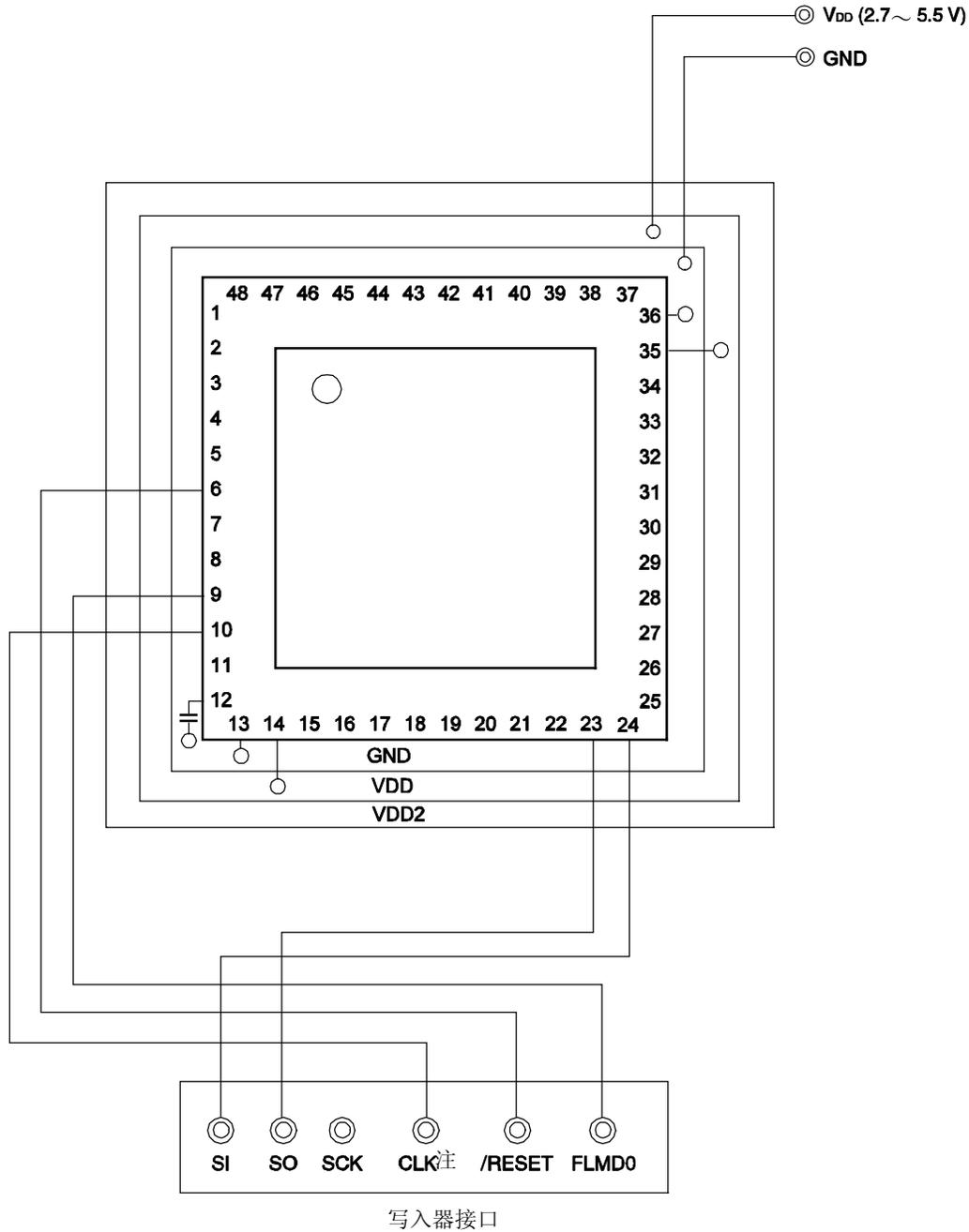
- PG-FP4, FL-PR4: 编程器的 CLK 到 EXCLK/X2/P122 (引脚 10)。
- PG-FPL3, FP-LITE3: 编程器的 CLK 到 X1/P121 (引脚 11)，并且连接其反相信号到 X2/EXCLK/P122 (引脚 10)。

2. 仅对  $\mu$ PD78F040x。

3. 仅对  $\mu$ PD78F041x。

在使用适配器对 Flash 存储器进行写操作时可采用如下所示的连接方式。

图 24-2. UART (UART6) 模式下使用适配器对 Flash 存储器进行写操作示例



**注** 上图显示的是当使用从 PG-FP4 或 FL-PR4 中输出的时钟时的连线例子。  
 当使用从 PG-FPL3 或 FP-LITE3 中输出的时钟时，把 CLK 连接到 X1/P121 (引脚 11)，并且将它的反相信号连接到 X2/EXCLK/P122 (引脚 10)。

### 24.3 编程环境

以下是 78K0/LC3 Flash 存储器所需的编程环境。

图 24-3. Flash 存储器编程环境



需要有一个控制专用 Flash 编程器的主机。

专用 Flash 编程器与 78K0/LC3 的接口使用 UART6 进行写和擦除等操作。Flash 存储器离线写操作时必须使用一个专用编程适配器（FA 系列）。

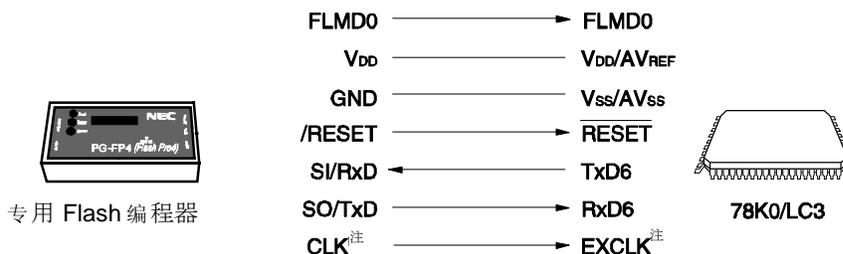
### 24.4 通信模式

通过 78K0/LC3 的 UART6，在专用 Flash 编程器与 78K0/LC3 之间建立串行通信。

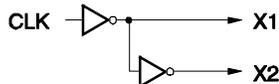
#### • UART6

传输速率: 115200 bps

图 24-4. 专用 Flash 编程器的通信(UART6)



注 上图显示的是当使用从 PG-FP4 或 FL-PR4 中输出时钟时的例子。当使用从 PG-FPL3 或 FP-LITE3 中输出的时钟时，把 CLK 连接到 X1/P121，并且将它的反相信号连接到 X2/EXCLK/P122。



注意事项 当通过 flash 编程器写入时，只有低端引脚(引脚号 23 和 24)响应 UART6(RxD6 和 TxD6)。不要通过高端引脚(引脚号 48 和 47)进行写入。

对于 78K0/LC3，专用 flash 编程器产生下列信号。详细信息，参考 PG-FP4, FL-PR4, PG-FPL3, 或 FP-LITE3 的用户手册。

表 24-3. 引脚连接

专用 Flash 编程器			78K0/LC3	连接
信号名称	I/O	引脚功能	引脚名称	UART6
FLMD0	输出	模式信号	FLMD0	□
V <sub>DD</sub>	I/O	V <sub>DD</sub> 电压产生/电源监测	V <sub>DD</sub> , AV <sub>REF</sub> <sup>注 2</sup>	□
GND	-	地	V <sub>SS</sub> , AV <sub>SS</sub> <sup>注 2</sup>	□
CLK	输出	时钟输出到 78K0/LC3	注 1	○ <sup>注 1</sup>
/RESET	输出	复位信号	RESET	□
SI/RxD	输入	接收信号	TxD6	□
SO/TxD	输出	发送信号	RxD6	□
SCK	输出	传输时钟	-	x

注 1. 在使用 UART6 时，只能使用 X1 时钟(f<sub>x</sub>)或外部主系统时钟(f<sub>EXCLK</sub>)。要使用 Flash 编程器的时钟输出，引脚连接依据所用的专用的 flash 编程器不同而不同。

- PG-FP4, FL-PR4: 编程器的 CLK 到 EXCLK/X2/P122。
- PG-FPL3, FP-LITE3: 编程器的 CLK 到 X1/P121 (引脚 11)，并且连接其反相信号到 X2/EXCLK/P122。

2. 仅对  $\mu$ PD78F041x。

备注 □: 确保连接引脚。

○: 如果在目标板上产生信号，则此引脚不需要连接。

x: 此引脚不需要连接。

## 24.5 在线方式的引脚连接

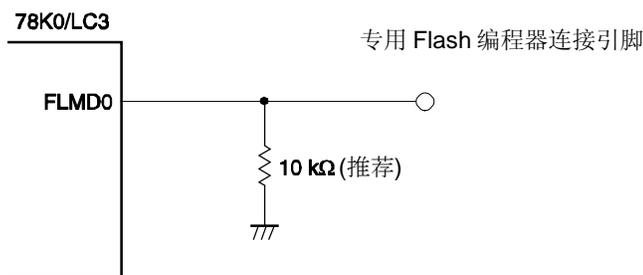
对 Flash 存储器进行 on-board 写入操作时，目标系统必须有连接专用 Flash 编程器的连接器。电路板上首先要提供一个选择功能，可以选择正常操作模式或 Flash 存储器编程模式。

当设置 Flash 存储器编程模式时，那些不用于 Flash 存储器编程的引脚状态与复位后的状态相同。因此如果外部设备不能立即识别复位后的状态，则必须采用如下方式连接引脚。

### 24.5.1 FLMD0 引脚

在普通操作模式下，FLMD0 引脚的输入电压为 0V。在 Flash 存储器编程模式中，将 V<sub>DD</sub> 写电压提供给 FLMD0 引脚。FLMD0 引脚的连接示意图如下所示。

图 24-5. FLMD0 引脚连接示例



### 24.5.2 串行接口引脚

串行接口使用的引脚如下所示。

表 24-4. 串行接口使用的引脚

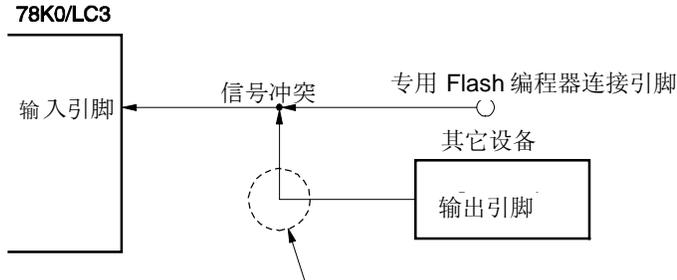
串行接口	使用引脚
UART6	TxD6, RxD6

在将专用 Flash 编程器和串行接口 (已与板上其它设备连接) 的引脚相连时，必须特别注意：信号之间不能冲突，并且另一个设备不能出现误操作。

**(1) 信号冲突**

如果专用 Flash 编程器 (输出端) 与串行接口 (已连接到另一个设备的输出端) 的一个输入引脚相连, 这时会产生信号冲突。为了避免这种情况, 应隔离与另一个设备的连接, 或者使另一个设备处于高阻抗状态。

图 24-6. 信号冲突 (串行接口的输入引脚)

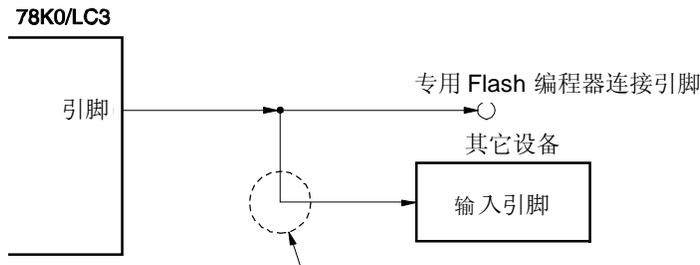


在 Flash 存储器编程模式下, 设备的输出信号与专用 Flash 编程器发送的信号发生冲突。因此必须隔离该设备的信号。

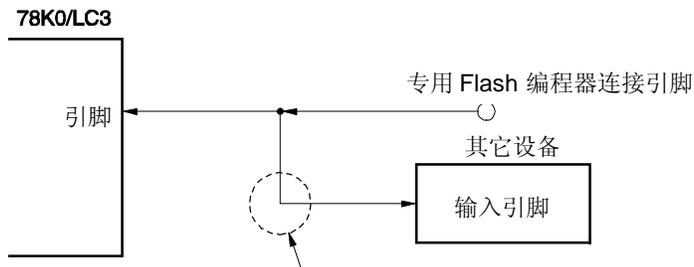
**(2) 其它设备的故障**

如果专用 Flash 编程器的输出或输入端与串行接口 (已连接到另一个设备的输入端) 的一个输入或输出引脚相连, 则信号可能会输出到另一个设备, 从而引起该设备的故障。为了避免这种情况, 应隔离与该设备的连接。

图 24-7. 其它设备的故障



在 Flash 存储器编程模式下, 如果 78K0/LC2 的输出信号影响到另外一个设备, 那么隔离另一设备的信号



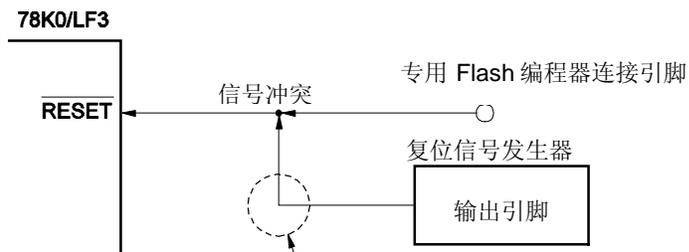
在 Flash 存储器编程模式下, 如果专用 Flash 编程器的输出信号影响到另外一个设备, 那么隔离另一设备的信号

### 24.5.3 RESET 引脚

如果将专用 Flash 编程器的复位信号连接到 RESET 引脚（已连接到板上的复位信号发生器），则会产生信号冲突。为了避免这种情况，应隔离与复位信号发生器的连接。

在 Flash 存储器编程模式下，如果从用户系统输入复位信号，则不能对 Flash 存储器进行正确编程。因此除了专用 Flash 编程器的复位信号外，不要输入其它信号。

图 24-8. 信号冲突(RESET 引脚)



在 Flash 存储器编程模式中，复位信号发生器输出的信号与专用 Flash 存储器编程器的输出信号发生冲突。因此必须隔离复位信号发生器的信号。

### 24.5.4 端口引脚

当设置 Flash 存储器编程模式时，那些不用于 Flash 存储器编程的引脚状态与复位后的状态相同。因此，如果外部设备(与端口相连)不能立即识别复位后的状态，则必须通过一个电阻将端口引脚连接到 V<sub>DD</sub> 或 V<sub>SS</sub>。

### 24.5.5 REGC 引脚

以正常操作相同的方式，通过一个电容 (0.47 ~ 1 μF：推荐)连接 REGC 引脚到 GND。

### 24.5.6 其它信号引脚

使用 on-board 时钟时，将与正常操作模式下相同的状态连接 X1 和 X2。

为了从编程器输入操作时钟，那么，照下列所示连接。

- PG-FP4, FL-PR4: 编程器的 CLK 连接到 EXCLK/X2/P122。
- PG-FPL3, FP-LITE3: 编程器的 CLK 连接到 X1/P121 并且将它的反相信号连接到 X2/EXCLK/P122。

**注意事项** 当 UART6 使用时，只有 X1 时钟( $f_x$ ) 或外部主系统时钟 ( $f_{EXCLK}$ )可以使用。

### 24.5.7 供电电压

要使用 Flash 编程器输出的供电电压，则将 V<sub>DD</sub> 引脚与 Flash 编程器的 V<sub>DD</sub> 相连，将 V<sub>SS</sub> 引脚与 Flash 编程器的 GND 相连。

然而，如果要使用 Flash 编程器的电压监视功能，必须将 V<sub>DD</sub> 和 V<sub>SS</sub> 引脚分别与 Flash 编程器的 V<sub>DD</sub> 和 GND 相连。

要使用板上供电电压，要按正常操作模式连接。

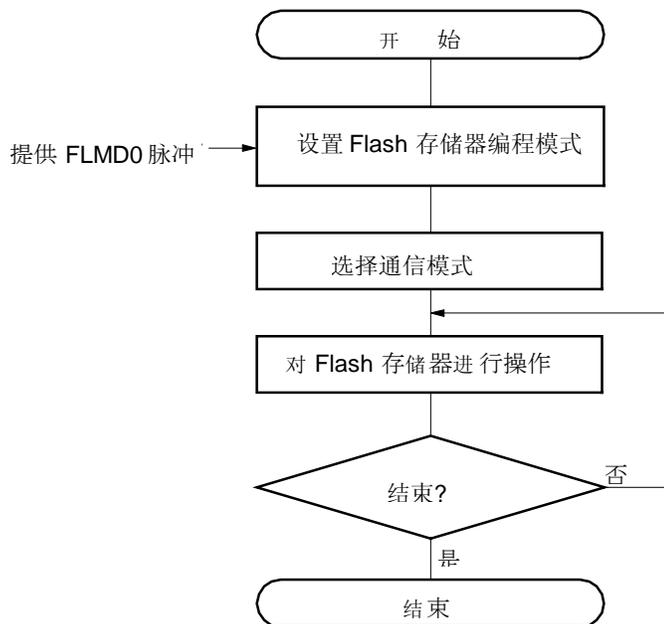
其它的供电电压 (LV<sub>DD</sub>, LV<sub>SS</sub>, AV<sub>REF</sub> 和 AV<sub>SS</sub>) 与正常操作模式中的相同。

## 24.6 编程方法

### 24.6.1 控制 Flash 存储器

下图显示了 Flash 存储器操作过程。

图 24-9. Flash 存储器操作过程



### 24.6.2 Flash 存储器编程模式

如果通过专用 Flash 编程器重写 Flash 存储器的内容，则必须将 78K0/LC3 设置为 Flash 存储器编程模式。而要设置该模式，必须将 FLMD0 引脚连接到  $V_{DD}$ ，并对复位信号清零。

当 on-board 写 Flash 存储器时，使用跳线改变模式。

图 24-10. Flash 存储器编程模式

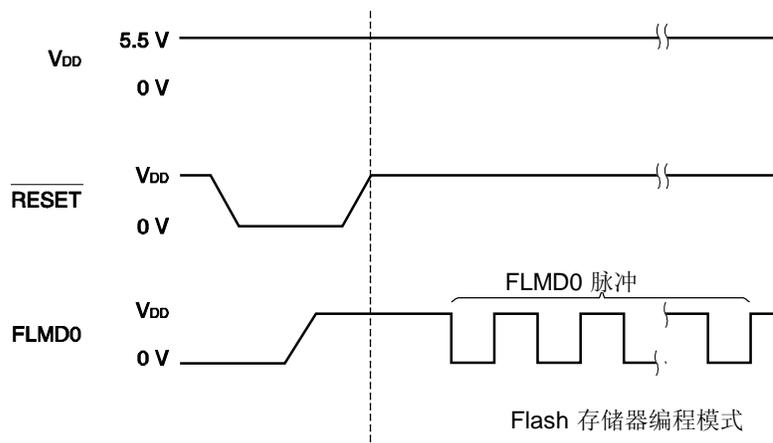


表 24-5. 复位释放后 FLMD0 引脚与操作模式之间的关系

FLMD0	操作模式
0	正常操作模式
$V_{DD}$	Flash 存储器编程模式

### 24.6.3 选择通信模式

在 78K0/LC3 中, 进入专用 Flash 存储器编程模式后, 通过将脉冲输入到 FLMD0 引脚, 来选择通信模式。由 Flash 编程器产生这些 FLMD0 脉冲。

下表显示了脉冲个数与通信模式之间的关系。

表 24-6. 通信模式

通信模式	标准设置 <sup>注1</sup>				使用引脚	外围时钟	FLMD0 脉冲的个数
	端口	速度	频率	倍率			
UART (UART6)	UART-Ext-Osc	115,200 bps <sup>注3</sup>	2 ~ 10 MHz <sup>注2</sup>	1.0	TxD6, RxD6	f <sub>x</sub>	0
	UART-Ext-FP4CK					f <sub>EXCLK</sub>	3

- 注
1. 编程器 GUI 的标准设置选择项目名称。
  2. 电压不同设置范围也不一样。如需了解详细内容, 可参见第二十七章 电气特性(标准产品)。
  3. 波特率误差以外的因素, 如信号波形瞬变, 也会影响 UART 通信, 因此必须象测量波特率误差一样严格测量瞬变。

**注意事项** 当选择 UART6 时, 在接收到 FLMD0 脉冲后, 根据专用 Flash 编程器发送的复位命令计算接收时钟。

**备注**

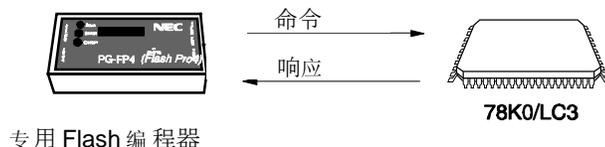
f<sub>x</sub>: X1 时钟

f<sub>EXCLK</sub>: 外部主系统时钟

## 24.6.4 通信命令

78K0/LC3 使用命令与专用 Flash 编程器进行通信。从 Flash 编程器发往 78K0/LC3 的信号称为命令，从 78K0/LC3 发往专用 Flash 编程器的信号称为响应命令。

图 24-11. 通信命令



78K0/LC3 的 Flash 存储器控制命令列表如下。所有这些命令都从编程器发出，78K0/LC3 根据相应的命令进行处理。

表 24-7. Flash 存储器控制命令

类别	命令名称	功能
校验	一次性校验命令	比较特定存储器内容和输入数据。
擦除	片擦除	擦除整个存储器内容。
	Block 擦除	擦除特定区域。
空白检测	Block 空白检测	存储器的一个特定 block 正确擦除，那么就进行检测。
数据写	编程	在 flash 存储器的特定区域写数据。
获取信息	状态	获取当前操作状态 (状态数据)。
	硅标记	获取 78K0/Lx3 信息(例如设备号和存储器配置)。
	获取版本	获取 78K0/Lx3 版本和固件版本。
	校验和	获取特定区域的校验和数据。
安全设置	安全设置	设置安全信息。
其它	复位	用来检测通信的同步状态。
	振荡频率设置	规定振荡频率

78K0/LC3 根据专用 Flash 编程器发出的命令返回一个响应。78K0/LC3 发出的响应列表如下。

表 24-8. 响应命令

响应名称	功能
ACK	响应命令/数据
NAK	响应非法命令/数据

## 24.7 安全性设置

78K0/LC3 禁止用户程序重写内部 flash 存储器，因此未授权用户就不能更改程序。

以下显示使用安全设定命令能够执行的操作。在设置下一个编程模式时，安全性设置有效。

- 禁止一次性擦除 (片擦除)

该项设置禁止对 Flash 存储器整个 block 进行 block 擦除和一次性擦除(片擦除)。一旦禁止执行一次性擦除(片擦除)命令，所有禁止设置项不能被取消。

**注意事项** 在对一次性擦除进行安全性设置后，不能再对该设备执行擦除操作。此外，由于禁止执行擦除命令，即使执行了写命令，与已经写到 Flash 存储器中的数据不同的数据也不能被写入，因为禁止擦除命令。

- 禁止 block 擦除

该项设置禁止对 Flash 存储器中指定 block 执行 block 擦除命令。使用一次性擦除（片擦除）命令时可取消禁止设置。

- 禁止写

该项设置禁止对 Flash 存储器整个 block 执行写命令和 block 擦除命令。使用一次性擦除（片擦除）命令时可取消禁止设置。

- 禁止重写引导簇 0

该项设置禁止对 Flash 存储器中引导簇 0(0000H ~ 0FFFH)执行一次性擦除(片擦除)命令、block 擦除命令和写命令。

**注意事项** 如果对重写引导簇 0 进行了安全性设置，则不能再对该设备的引导簇 0 进行重写。

Flash 存储器在默认设置下，允许一次性擦除(片擦除)、block 擦除、写命令和重写引导簇 0 功能。以上安全性设置仅用于 On-board/Off-board 编程和自编程。各项安全性设置可以结合使用。

通过执行一次性擦除(片擦除)命令，禁止擦除 block 和写入就被清除。

表 24-9 显示了 78K0/LC3 允许使用安全功能时擦除命令与写命令之间的关系。

表 24-9. 允许使用安全功能时命令之间的关系

## (1) On-board/off-board 编程期间

有效的安全功能	执行的命令		
	一次性擦除(片擦除)	Block 擦除	写入
禁止一次性擦除(片擦除)	不能一次性擦除	Blocks 不能擦除。	不能执行 <sup>注</sup> 。
禁止 block 擦除	可以一次性擦除		可以执行。
禁止写入			不能执行。
禁止重写引导簇 0	不能一次性擦除	引导簇 0 不能擦除。	引导簇 0 不能写入。

**注** 确认没有把数据写入已经写入数据的区域。由于禁止使用一次性擦除（片擦除）命令，所以如果数据没有擦除，那么不能写入数据。

## (2) 自编程期间

有效的安全功能	执行的命令	
	Block 擦除	写入
禁止一次性擦除(片擦除)	Blocks 可以擦除。	可以执行。
禁止 block 擦除		
禁止写入		
禁止引导簇 0 重写	引导簇 0 不能擦除。	引导簇 0 不能写入。

表 24-10 显示了在每一种编程模式下怎样进行安全设置。

表 24-10. 每一种编程模式下的安全设置

## (1) On-board/off-board 编程

安全功能	安全设置	怎样禁止安全设置
禁止一次性擦除(片擦除)	通过专用 flash 编程器的 GUI 等进行设置。	设置之后不能禁止。
禁止 block 擦除		执行一次性擦除(片擦除) 命令
禁止写入		
禁止引导簇 0 重写		设置之后不能禁止

## (2) 自编程

安全功能	安全设置	怎样禁止安全设置
禁止一次性擦除(片擦除)	由信息库进行设置。	设置之后不能禁止。
禁止 block 擦除		On-board/off-board 编程期间执行执行一次性擦除(片擦除) 命令(自编程期间不能禁止)。
禁止写入		
禁止引导簇 0 重写		设置之后不能禁止。

## 24.8 通过自写入进行 Flash 存储器编程(开发中)

78K0/LC3 支持自编程功能,即可以通过用户程序重写 Flash 存储器。由于该功能允许用户通过使用使用 78K0/LC3 自编程示例库重写 flash 存储器,因此可用于对程序升级。

如果自编程期间产生中断,可以暂时停止自编程操作,而去执行中断服务程序。要执行中断服务程序,应在停止自编程操作后恢复正常操作模式,并执行 EI 指令。而在恢复自编程模式后,可以继续自编程操作。

**注意事项** 1. 当 CPU 使用副系统时钟时不能使用自编程功能。

2. 自编程期间 FLMD0 引脚应输入高电平。

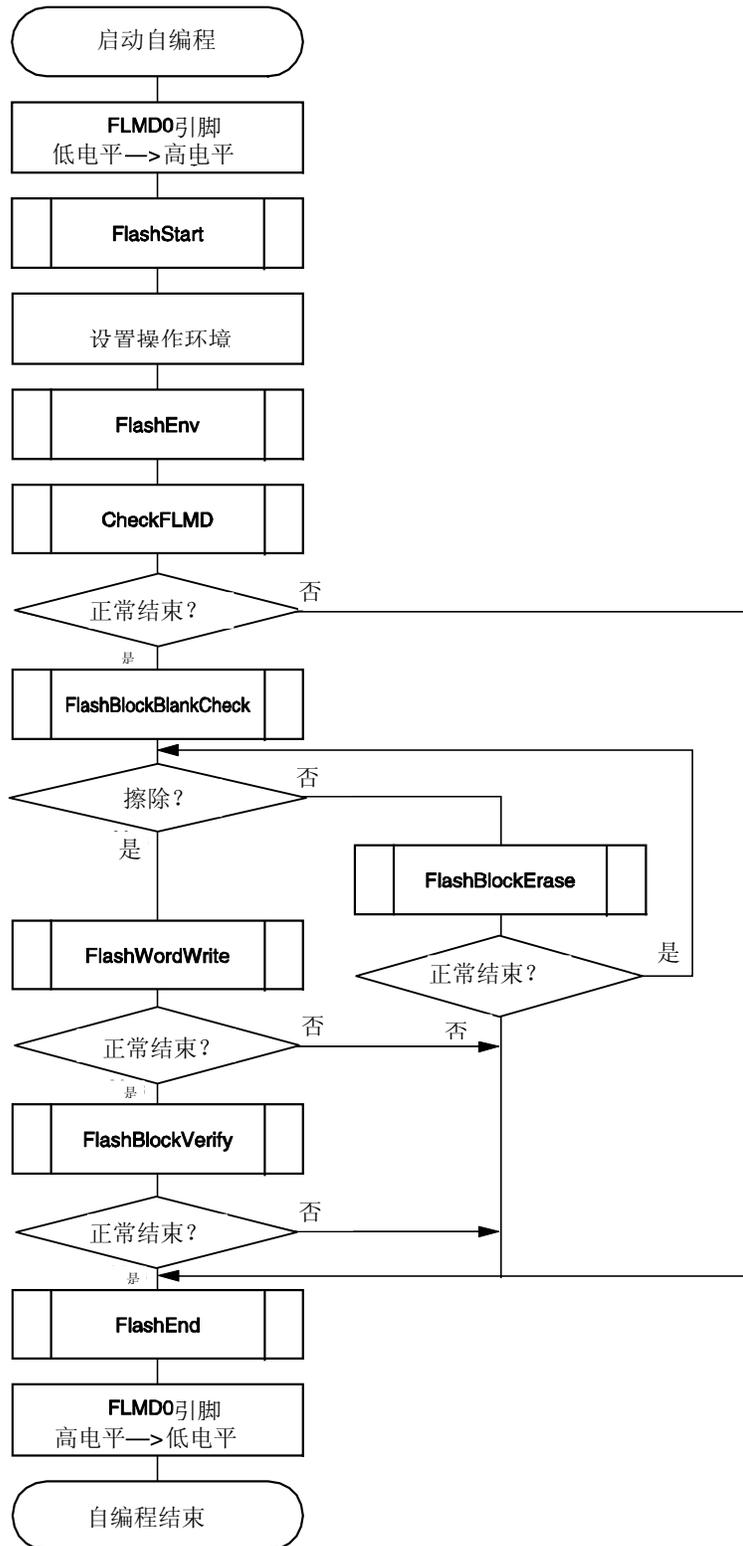
3. 在启动自编程之前必须执行 DI 指令。

自编程功能检查中断请求标志 (IF0L, IF0H, IF1L 和 IF1H)。如果产生了中断请求,则停止自编程操作。

4. 即使在 DI 状态下也可以通过未屏蔽的中断请求停止自编程。要避免这种情况,可以通过使用中断屏蔽标志寄存器(MK0L, MK0H, MK1L 和 MK1H)屏蔽该中断。

下图显示了使用自编程范例库重写 flash 存储器的流程。

图 24-12. 自编程流程 (重写 Flash 编程器)



### 24.8.1 引导交换功能

如果在自编程期间由于电源失效或其它原因导致重写引导区域的操作失败，则引导区域的数据可能会丢失且不能通过复位重启程序。

使用引导交换功能可以避免这种情况。

在擦除引导簇 0(这是一个引导程序区域)\*之前，先通过自编程将一个新的引导程序写入引导簇 1。当将该程序正确写入引导簇 1 后，通过使用 78K0/LC3 固件的设置信息功能将引导簇 1 与引导簇 0 的内容交换，这样引导簇 1 就用作引导区域。之后对初始引导程序区域，即引导簇 0 进行擦除或写操作。

这样，即使在重写引导程序区域时出现电源失效，也能够正确执行程序，这是因为程序在复位并再次执行时实行了从引导簇 1 的引导交换。

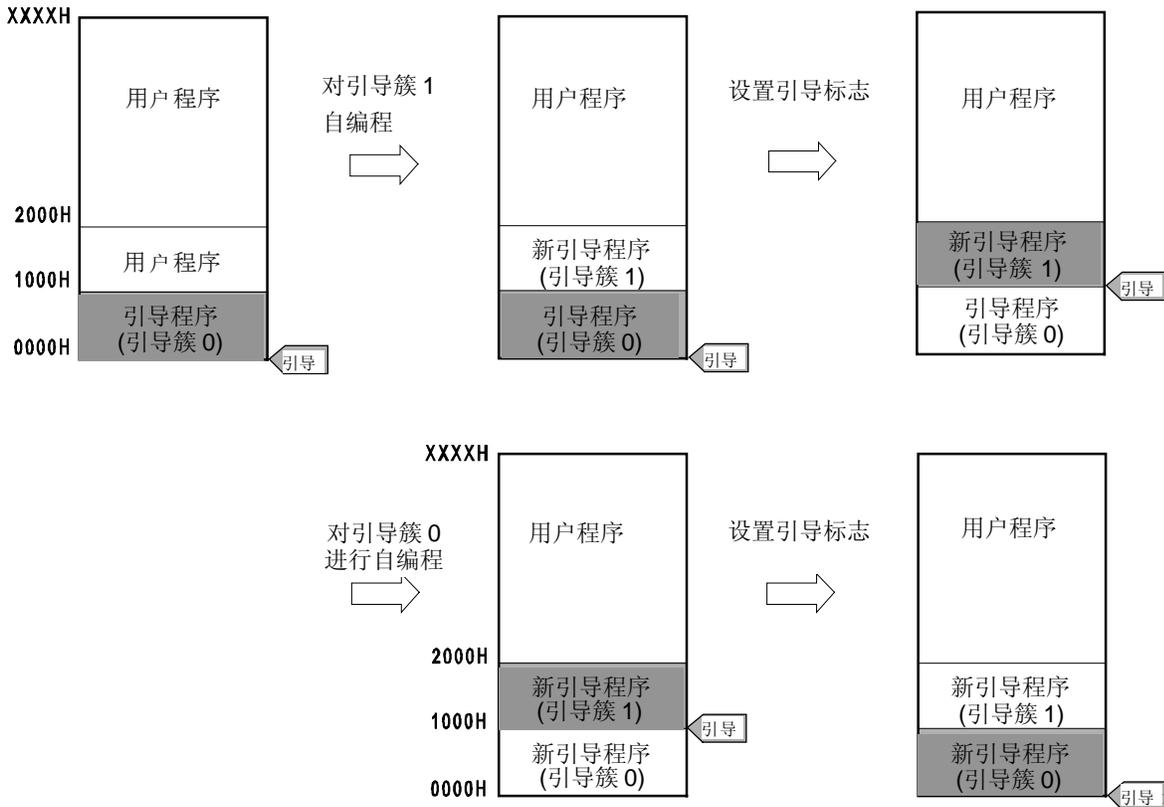
如果已经将程序正确写入引导簇 0，则通过使用 78K0/LC3 固件的设置信息功能恢复初始引导区域。

**注** 一个引导簇的容量为 4 KB，通过引导交换功能交换引导簇 0 与引导簇 1 的内容。

引导簇 0 (0000H ~ 0FFFH): 初始引导程序区域

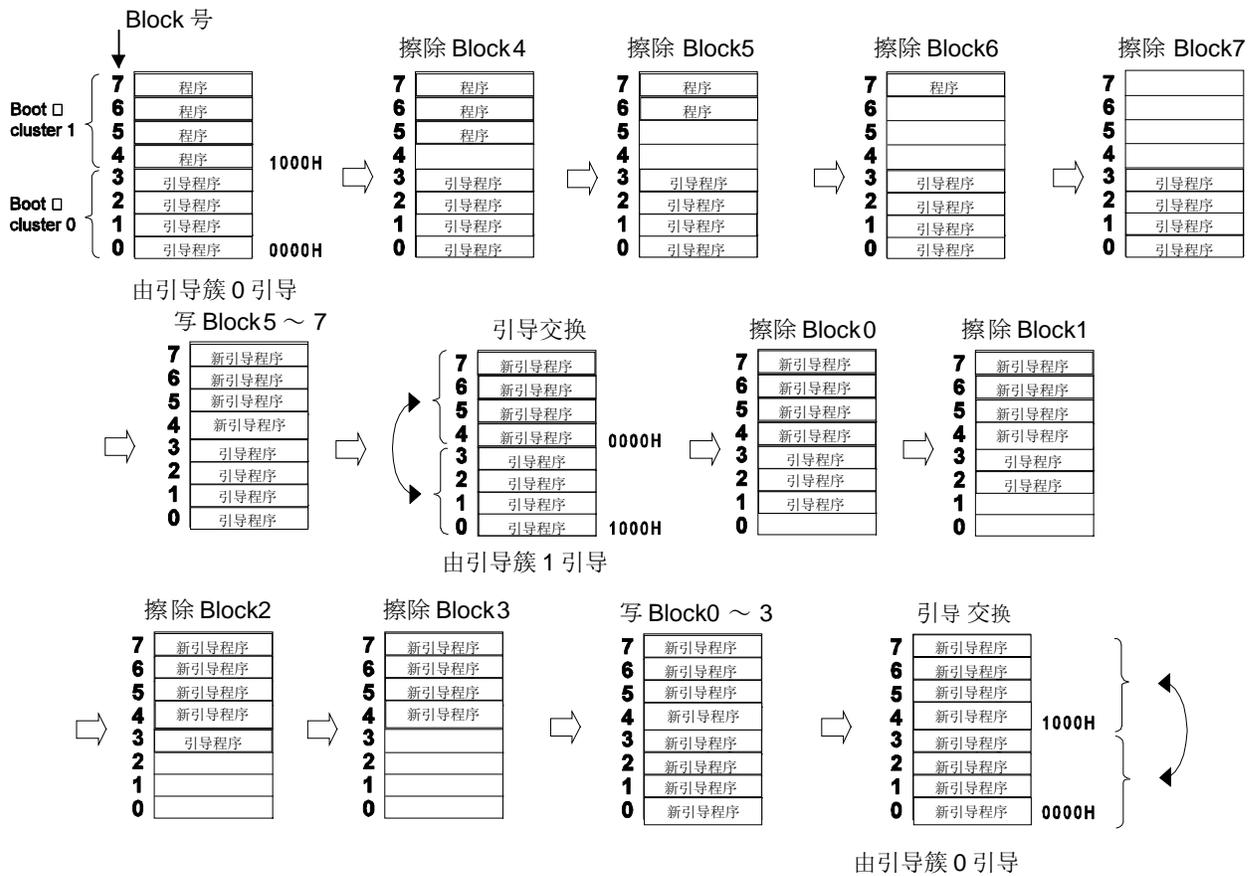
引导簇 1 (1000H ~ 1FFFH): 用于引导交换的区域

图 24-13. 引导交换功能



**备注** 引导标志设置后，当产生复位时，引导簇 1 变成 0000H ~ 0FFFH。

图 24-14. 执行引导交换的实例



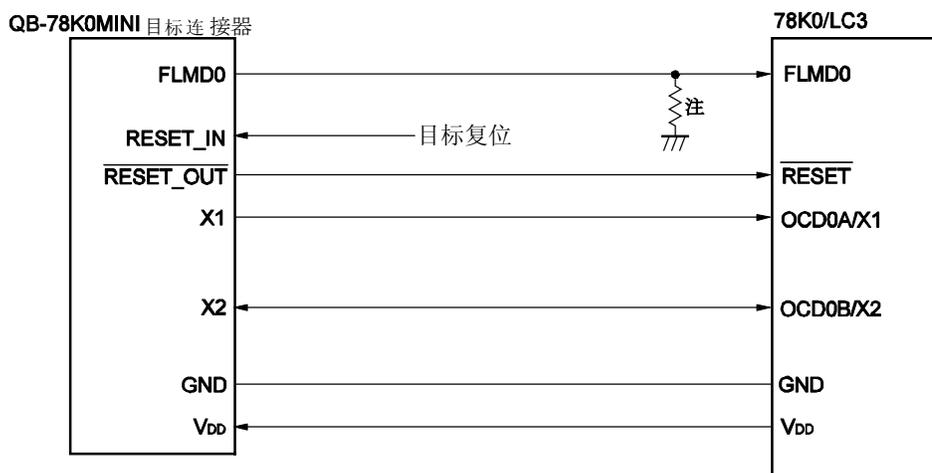
## 第二十五章 片上调试功能

### 25.1 QB-78K0MINI 与 78K0/LC3 的连接

78K0/LC3 使用  $V_{DD}$ 、FLMD0、 $\overline{\text{RESET}}$ 、OCD0A/X1、OCD0B/X2, 和  $V_{SS}$  引脚，通过片上调试仿真器（QB-78K0MINI）与主机通信。

**注意事项** 78K0/LC3 具有片上调试功能。由于在使用片上调试功能后产品的稳定性不能得到保证，因此在给定 flash 存储器可重写次数的情况下不要将该产品用于大规模生产。NEC 电子不接受有关该产品的投诉。

图 25-1. QB-78K0MINI 与 78K0/LC3 的连接示例  
(当使用 OCD0A/X1 和 OCD0B/X2 时)

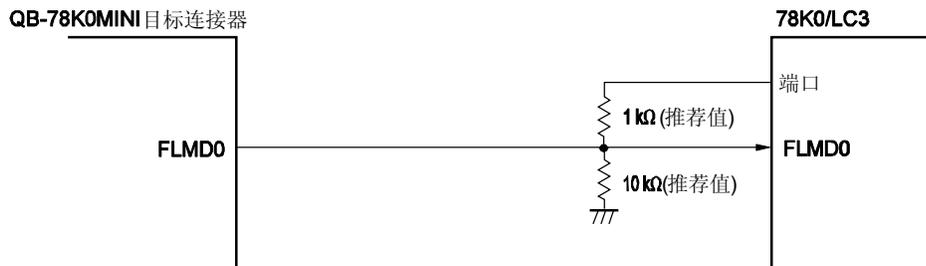


**注** 下拉电阻值至少应为 470  $\Omega$ 。

**注意事项** 在片上调试期间从 OCD0A/X1 引脚输入时钟。

当通过片上调试进行自编程时按下列方式连接 FLMD0 引脚。

图 25-2. 通过片上调试进行自编程时 FLMD0 引脚的连接



## 25.2 片上调试安全 ID

在 78K0/LC3 中，一个片上调试操作控制标志存放在 flash 存储器地址 0084H 中 (参见第二十三章 选项字节)，片上调试安全 ID 设置在地址 0085H ~ 008EH 中。

当使用引导交换功能时，由于 0084H、0085H ~ 008EH 与 1084H 以及 1085H ~ 108EH 的内容被交换，因此预先设置与 1084H 以及 1085H ~ 108EH 的内容相同的值。

如需了解片上调试安全 ID 的详细内容，可参见 QB-78K0MINI 用户手册 (U17029E)。

表 25-1. 片上调试安全 ID

地址	片上调试安全 ID
0085H ~ 008EH	10 字节的 ID 代码
1085H ~ 108EH	

## 第二十六章 指令集

78K0/LC3 本章以表格方式列出了 78K0/LC3 的指令集。如需了解每种指令的操作和操作代码，可参阅 **78K0 系列指令用户手册 (U12326E)**。

### 26.1 操作列表使用规则

#### 26.1.1 操作数标识符和标识方法

根据规范确定的指令操作数标识方法（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标识方法，可选其中之一。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号。当使用标号时，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 26-1. 操作数标识符和标识方法

标识符	标识方法
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	特殊功能寄存器符号 <sup>注</sup>
sfrp	特殊功能寄存器符号（仅用于 16 位可操作寄存器偶地址） <sup>注</sup>
saddr	FE20H ~ FF1FH 立即数或标号
saddrp	FE20H ~ FF1FH 立即数或标号（仅用于偶地址）
addr16	0000H ~ FFFFH 立即数或标号 （仅用于 16 位数据传送指令的偶地址）
addr11	0800H ~ 0FFFH 立即数或标号
addr5	0040H ~ 007FH 立即数或标号（仅用于偶地址）
word	16 位立即数或标号
byte	8 位立即数或标号
bit	3 位立即数或标号
RBn	RB0 ~ RB3

注 不能使用这些操作数访问地址 FFD0H ~ FFD7H。

备注 特殊功能寄存器符号参见表 3-6 特殊功能寄存器列表

**26.1.2 操作栏描述**

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
RBS:	寄存器 bank 选择标志
IE:	中断请求允许标志
( ):	括号中的地址或寄存器所指的存储单元的内容
X <sub>H</sub> , X <sub>L</sub> :	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 (exclusive OR)
—:	数据取反
addr16:	16 位立即数或标号
jdisp8:	带符号的 8 位数据 (偏移量)

**26.1.3 标志操作栏的描述**

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

## 26.2 操作列表

指令组	助记符	操作数	字节数	时钟		操作	标志			
				注 1	注 2		Z	AC	CY	
8 位数据传送	MOV	r, #byte	2	4	-	r ← byte				
		saddr, #byte	3	6	7	(saddr) ← byte				
		sfr, #byte	3	-	7	sfr ← byte				
		A, r <sup>注 3</sup>	1	2	-	A ← r				
		r, A <sup>注 3</sup>	1	2	-	r ← A				
		A, saddr	2	4	5	A ← (saddr)				
		saddr, A	2	4	5	(saddr) ← A				
		A, sfr	2	-	5	A ← sfr				
		sfr, A	2	-	5	sfr ← A				
		A, !addr16	3	8	9	A ← (addr16)				
		!addr16, A	3	8	9	(addr16) ← A				
		PSW, #byte	3	-	7	PSW ← byte		x	x	x
		A, PSW	2	-	5	A ← PSW				
		PSW, A	2	-	5	PSW ← A		x	x	x
		A, [DE]	1	4	5	A ← (DE)				
		[DE], A	1	4	5	(DE) ← A				
		A, [HL]	1	4	5	A ← (HL)				
		[HL], A	1	4	5	(HL) ← A				
		A, [HL + byte]	2	8	9	A ← (HL + byte)				
		[HL + byte], A	2	8	9	(HL + byte) ← A				
	A, [HL + B]	1	6	7	A ← (HL + B)					
	[HL + B], A	1	6	7	(HL + B) ← A					
	A, [HL + C]	1	6	7	A ← (HL + C)					
	[HL + C], A	1	6	7	(HL + C) ← A					
	XCH	A, r <sup>注 3</sup>	1	2	-	A ↔ r				
		A, saddr	2	4	6	A ↔ (saddr)				
		A, sfr	2	-	6	A ↔ (sfr)				
		A, !addr16	3	8	10	A ↔ (addr16)				
		A, [DE]	1	4	6	A ↔ (DE)				
		A, [HL]	1	4	6	A ↔ (HL)				
		A, [HL + byte]	2	8	10	A ↔ (HL + byte)				
		A, [HL + B]	2	8	10	A ↔ (HL + B)				
A, [HL + C]	2	8	10	A ↔ (HL + C)						

- 注
1. 当访问内部高速 RAM 时或针对无数据访问的指令
  2. 当访问内部高速 RAM 以外的区域时
  3. “r = A” 除外

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (fCPU) 的一个周期。
  2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作数	字节数	时钟		操作	标志
				注 1	注 2		Z AC CY
16 位数据 传送	<b>MOVW</b>	rp, #word	3	6	-	rp ← word	
		saddrp, #word	4	8	10	(saddrp) ← word	
		sfrp, #word	4	-	10	sfrp ← word	
		AX, saddrp	2	6	8	AX ← (saddrp)	
		saddrp, AX	2	6	8	(saddrp) ← AX	
		AX, sfrp	2	-	8	AX ← sfrp	
		sfrp, AX	2	-	8	sfrp ← AX	
		AX, rp <sup>注 3</sup>	1	4	-	AX ← rp	
		rp, AX <sup>注 3</sup>	1	4	-	rp ← AX	
		AX, !addr16	3	10	12	AX ← (addr16)	
	!addr16, AX	3	10	12	(addr16) ← AX		
<b>XCHW</b>	AX, rp <sup>注 3</sup>	1	4	-	AX ↔ rp		
8 位操作	<b>ADD</b>	A, #byte	2	4	-	A, CY ← A + byte	x x x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	x x x
		A, r <sup>注 4</sup>	2	4	-	A, CY ← A + r	x x x
		r, A	2	4	-	r, CY ← r + A	x x x
		A, saddr	2	4	5	A, CY ← A + (saddr)	x x x
		A, !addr16	3	8	9	A, CY ← A + (addr16)	x x x
		A, [HL]	1	4	5	A, CY ← A + (HL)	x x x
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	x x x
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	x x x
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C)	x x x
	<b>ADDC</b>	A, #byte	2	4	-	A, CY ← A + byte + CY	x x x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	x x x
		A, r <sup>注 4</sup>	2	4	-	A, CY ← A + r + CY	x x x
		r, A	2	4	-	r, CY ← r + A + CY	x x x
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	x x x
		A, !addr16	3	8	9	A, CY ← A + (addr16) + C	x x x
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	x x x
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	x x x
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	x x x
A, [HL + C]	2	8	9	A, CY ← A + (HL + C) + CY	x x x		

- 注
1. 当访问内部高速 RAM 时或针对无数据访问的指令
  2. 当访问内部高速 RAM 以外的区域时
  3. 仅当 rp = BC, DE 或 HL
  4. “r = A” 除外

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (fcpu) 的一个周期。
  2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8-位操作	SUB	A, #byte	2	4	-	A, CY ← A - byte	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) - byte	x	x	x
		A, r <sup>注 3</sup>	2	4	-	A, CY ← A - r	x	x	x
		r, A	2	4	-	r, CY ← r - A	x	x	x
		A, saddr	2	4	5	A, CY ← A - (saddr)	x	x	x
		A, !addr16	3	8	9	A, CY ← A - (addr16)	x	x	x
		A, [HL]	1	4	5	A, CY ← A - (HL)	x	x	x
		A, [HL + byte]	2	8	9	A, CY ← A - (HL + byte)	x	x	x
		A, [HL + B]	2	8	9	A, CY ← A - (HL + B)	x	x	x
	A, [HL + C]	2	8	9	A, CY ← A - (HL + C)	x	x	x	
	SUBC	A, #byte	2	4	-	A, CY ← A - byte - CY	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) - byte - CY	x	x	x
		A, r <sup>注 3</sup>	2	4	-	A, CY ← A - r - CY	x	x	x
		r, A	2	4	-	r, CY ← r - A - CY	x	x	x
		A, saddr	2	4	5	A, CY ← A - (saddr) - CY	x	x	x
		A, !addr16	3	8	9	A, CY ← A - (addr16) - CY	x	x	x
		A, [HL]	1	4	5	A, CY ← A - (HL) - CY	x	x	x
		A, [HL + byte]	2	8	9	A, CY ← A - (HL + byte) - CY	x	x	x
		A, [HL + B]	2	8	9	A, CY ← A - (HL + B) - CY	x	x	x
	A, [HL + C]	2	8	9	A, CY ← A - (HL + C) - CY	x	x	x	
	AND	A, #byte	2	4	-	A ← A ∧ byte	x		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	x		
		A, r <sup>注 3</sup>	2	4	-	A ← A ∧ r	x		
		r, A	2	4	-	r ← r ∧ A	x		
		A, saddr	2	4	5	A ← A ∧ (saddr)	x		
		A, !addr16	3	8	9	A ← A ∧ (addr16)	x		
		A, [HL]	1	4	5	A ← A ∧ (HL)	x		
		A, [HL + byte]	2	8	9	A ← A ∧ (HL + byte)	x		
		A, [HL + B]	2	8	9	A ← A ∧ (HL + B)	x		
	A, [HL + C]	2	8	9	A ← A ∧ (HL + C)	x			

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
  2. 当访问内部高速 RAM 以外的区域时
  3. “r = A” 除外

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f<sub>cpu</sub>) 的一个周期。
  2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作数	字节数	时钟		操作	标志
				注 1	注 2		Z AC CY
8-位操作	OR	A, #byte	2	4	–	$A \leftarrow A \vee \text{byte}$	x
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	x
		A, r <sup>注3</sup>	2	4	–	$A \leftarrow A \vee r$	x
		r, A	2	4	–	$r \leftarrow r \vee A$	x
		A, saddr	2	4	5	$A \leftarrow A \vee (\text{saddr})$	x
		A, !addr16	3	8	9	$A \leftarrow A \vee (\text{addr16})$	x
		A, [HL]	1	4	5	$A \leftarrow A \vee (\text{HL})$	x
		A, [HL + byte]	2	8	9	$A \leftarrow A \vee (\text{HL} + \text{byte})$	x
		A, [HL + B]	2	8	9	$A \leftarrow A \vee (\text{HL} + B)$	x
	A, [HL + C]	2	8	9	$A \leftarrow A \vee (\text{HL} + C)$	x	
	XOR	A, #byte	2	4	–	$A \leftarrow A \nabla \text{byte}$	x
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \nabla \text{byte}$	x
		A, r <sup>注3</sup>	2	4	–	$A \leftarrow A \nabla r$	x
		r, A	2	4	–	$r \leftarrow r \nabla A$	x
		A, saddr	2	4	5	$A \leftarrow A \nabla (\text{saddr})$	x
		A, !addr16	3	8	9	$A \leftarrow A \nabla (\text{addr16})$	x
		A, [HL]	1	4	5	$A \leftarrow A \nabla (\text{HL})$	x
		A, [HL + byte]	2	8	9	$A \leftarrow A \nabla (\text{HL} + \text{byte})$	x
		A, [HL + B]	2	8	9	$A \leftarrow A \nabla (\text{HL} + B)$	x
	A, [HL + C]	2	8	9	$A \leftarrow A \nabla (\text{HL} + C)$	x	
	CMP	A, #byte	2	4	–	$A - \text{byte}$	x x x
		saddr, #byte	3	6	8	$(\text{saddr}) - \text{byte}$	x x x
		A, r <sup>注3</sup>	2	4	–	$A - r$	x x x
		r, A	2	4	–	$r - A$	x x x
		A, saddr	2	4	5	$A - (\text{saddr})$	x x x
		A, !addr16	3	8	9	$A - (\text{addr16})$	x x x
		A, [HL]	1	4	5	$A - (\text{HL})$	x x x
		A, [HL + byte]	2	8	9	$A - (\text{HL} + \text{byte})$	x x x
		A, [HL + B]	2	8	9	$A - (\text{HL} + B)$	x x x
	A, [HL + C]	2	8	9	$A - (\text{HL} + C)$	x x x	

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
  2. 当访问内部高速 RAM 以外的区域时
  3. “r = A” 除外

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（f<sub>cpu</sub>）的一个周期。
  2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位操作	<b>ADDW</b>	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	<b>SUBW</b>	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	<b>CMPW</b>	AX, #word	3	6	–	AX – word	×	×	×
乘法/ 除法	<b>MULU</b>	X	2	16	–	AX ← A × X			
	<b>DIVUW</b>	C	2	25	–	AX (商), C (余数) ← AX ÷ C			
递增/ 递减	<b>INC</b>	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	<b>DEC</b>	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	<b>INCW</b>	rp	1	4	–	rp ← rp + 1			
<b>DECW</b>	rp	1	4	–	rp ← rp – 1				
循环	<b>ROR</b>	A, 1	1	2	–	(CY, A <sub>7</sub> ← A <sub>0</sub> , A <sub>m-1</sub> ← A <sub>m</sub> ) × 1 次			×
	<b>ROL</b>	A, 1	1	2	–	(CY, A <sub>0</sub> ← A <sub>7</sub> , A <sub>m+1</sub> ← A <sub>m</sub> ) × 1 次			×
	<b>RORC</b>	A, 1	1	2	–	(CY ← A <sub>0</sub> , A <sub>7</sub> ← CY, A <sub>m-1</sub> ← A <sub>m</sub> ) × 1 次			×
	<b>ROLC</b>	A, 1	1	2	–	(CY ← A <sub>7</sub> , A <sub>0</sub> ← CY, A <sub>m+1</sub> ← A <sub>m</sub> ) × 1 次			×
	<b>ROR4</b>	[HL]	2	10	12	A <sub>3-0</sub> ← (HL) <sub>3-0</sub> , (HL) <sub>7-4</sub> ← A <sub>3-0</sub> , (HL) <sub>3-0</sub> ← (HL) <sub>7-4</sub>			
	<b>ROL4</b>	[HL]	2	10	12	A <sub>3-0</sub> ← (HL) <sub>7-4</sub> , (HL) <sub>3-0</sub> ← A <sub>3-0</sub> , (HL) <sub>7-4</sub> ← (HL) <sub>3-0</sub>			
BCD 调整	<b>ADJBA</b>		2	4	–	加法后的十进制调整	×	×	×
	<b>ADJBS</b>		2	4	–	减法后的十进制调整	×	×	×
位操作	<b>MOV1</b>	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY			×
[HL].bit, CY	2	6	8	(HL).bit ← CY					

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
  2. 当访问内部高速 RAM 以外的区域时

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f<sub>CPU</sub>) 的一个周期。
  2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作	字节数	时钟		操作	标志
				注 1	注 2		Z AC CY
位操作	<b>AND1</b>	CY, saddr.bit	3	6	7	$CY \leftarrow CY \wedge (\text{saddr.bit})$	×
		CY, sfr.bit	3	-	7	$CY \leftarrow CY \wedge \text{sfr.bit}$	×
		CY, A.bit	2	4	-	$CY \leftarrow CY \wedge A.\text{bit}$	×
		CY, PSW.bit	3	-	7	$CY \leftarrow CY \wedge \text{PSW.bit}$	×
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \wedge (\text{HL}).\text{bit}$	×
	<b>OR1</b>	CY, saddr.bit	3	6	7	$CY \leftarrow CY \vee (\text{saddr.bit})$	×
		CY, sfr.bit	3	-	7	$CY \leftarrow CY \vee \text{sfr.bit}$	×
		CY, A.bit	2	4	-	$CY \leftarrow CY \vee A.\text{bit}$	×
		CY, PSW.bit	3	-	7	$CY \leftarrow CY \vee \text{PSW.bit}$	×
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \vee (\text{HL}).\text{bit}$	×
	<b>XOR1</b>	CY, saddr.bit	3	6	7	$CY \leftarrow CY \oplus (\text{saddr.bit})$	×
		CY, sfr.bit	3	-	7	$CY \leftarrow CY \oplus \text{sfr.bit}$	×
		CY, A.bit	2	4	-	$CY \leftarrow CY \oplus A.\text{bit}$	×
		CY, PSW.bit	3	-	7	$CY \leftarrow CY \oplus \text{PSW.bit}$	×
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \oplus (\text{HL}).\text{bit}$	×
	<b>SET1</b>	saddr.bit	2	4	6	$(\text{saddr.bit}) \leftarrow 1$	
		sfr.bit	3	-	8	$\text{sfr.bit} \leftarrow 1$	
		A.bit	2	4	-	$A.\text{bit} \leftarrow 1$	
		PSW.bit	2	-	6	$\text{PSW.bit} \leftarrow 1$	× × ×
		[HL].bit	2	6	8	$(\text{HL}).\text{bit} \leftarrow 1$	
	<b>CLR1</b>	saddr.bit	2	4	6	$(\text{saddr.bit}) \leftarrow 0$	
		sfr.bit	3	-	8	$\text{sfr.bit} \leftarrow 0$	
		A.bit	2	4	-	$A.\text{bit} \leftarrow 0$	
		PSW.bit	2	-	6	$\text{PSW.bit} \leftarrow 0$	× × ×
		[HL].bit	2	6	8	$(\text{HL}).\text{bit} \leftarrow 0$	
	<b>SET1</b>	CY	1	2	-	$CY \leftarrow 1$	1
	<b>CLR1</b>	CY	1	2	-	$CY \leftarrow 0$	0
	<b>NOT1</b>	CY	1	2	-	$CY \leftarrow \overline{CY}$	×

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
  2. 当访问内部高速 RAM 以外的区域时

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f<sub>cpu</sub>) 的一个周期。
  2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作数	字节数	时钟		操作	标志
				注 1	注 2		Z AC CY
调用/返回	<b>CALL</b>	!addr16	3	7	-	$(SP - 1) \leftarrow (PC + 3)_H, (SP - 2) \leftarrow (PC + 3)_L,$ $PC \leftarrow \text{addr16}, SP \leftarrow SP - 2$	
	<b>CALLF</b>	!addr11	2	5	-	$(SP - 1) \leftarrow (PC + 2)_H, (SP - 2) \leftarrow (PC + 2)_L,$ $PC_{15-11} \leftarrow 00001, PC_{10-0} \leftarrow \text{addr11},$ $SP \leftarrow SP - 2$	
	<b>CALLT</b>	[addr5]	1	6	-	$(SP - 1) \leftarrow (PC + 1)_H, (SP - 2) \leftarrow (PC + 1)_L,$ $PC_H \leftarrow (00000000, \text{addr5} + 1),$ $PC_L \leftarrow (00000000, \text{addr5}),$ $SP \leftarrow SP - 2$	
	<b>BRK</b>		1	6	-	$(SP - 1) \leftarrow \text{PSW}, (SP - 2) \leftarrow (PC + 1)_H,$ $(SP - 3) \leftarrow (PC + 1)_L, PC_H \leftarrow (003FH),$ $PC_L \leftarrow (003EH), SP \leftarrow SP - 3, IE \leftarrow 0$	
	<b>RET</b>		1	6	-	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $SP \leftarrow SP + 2$	
	<b>RETI</b>		1	6	-	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $\text{PSW} \leftarrow (SP + 2), SP \leftarrow SP + 3$	R R R
	<b>RETB</b>		1	6	-	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $\text{PSW} \leftarrow (SP + 2), SP \leftarrow SP + 3$	R R R
堆栈操作	<b>PUSH</b>	PSW	1	2	-	$(SP - 1) \leftarrow \text{PSW}, SP \leftarrow SP - 1$	
		rp	1	4	-	$(SP - 1) \leftarrow \text{rp}_H, (SP - 2) \leftarrow \text{rp}_L,$ $SP \leftarrow SP - 2$	
	<b>POP</b>	PSW	1	2	-	$\text{PSW} \leftarrow (SP), SP \leftarrow SP + 1$	R R R
		rp	1	4	-	$\text{rp}_H \leftarrow (SP + 1), \text{rp}_L \leftarrow (SP),$ $SP \leftarrow SP + 2$	
	<b>MOVW</b>	SP, #word	4	-	10	$SP \leftarrow \text{word}$	
		SP, AX	2	-	8	$SP \leftarrow \text{AX}$	
AX, SP		2	-	8	$\text{AX} \leftarrow \text{SP}$		
无条件转移	<b>BR</b>	!addr16	3	6	-	$PC \leftarrow \text{addr16}$	
		\$addr16	2	6	-	$PC \leftarrow PC + 2 + \text{jdisp8}$	
		AX	2	8	-	$\text{PCH} \leftarrow \text{A}, \text{PCL} \leftarrow \text{X}$	
条件转移	<b>BC</b>	\$addr16	2	6	-	如果 $\text{CY} = 1$ , 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	
	<b>BNC</b>	\$addr16	2	6	-	如果 $\text{CY} = 0$ , 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	
	<b>BZ</b>	\$addr16	2	6	-	如果 $\text{Z} = 1$ , 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	
	<b>BNZ</b>	\$addr16	2	6	-	如果 $\text{Z} = 0$ , 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
  2. 当访问内部高速 RAM 以外的区域时

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f<sub>cpu</sub>) 的一个周期。
  2. 该时钟周期用于内部 ROM 程序。

指令组	助记符	操作数	字节数	时钟		操作	标志
				注 1	注 2		Z AC CY
条件转移	<b>BT</b>	saddr.bit, \$addr16	3	8	9	如果 (saddr.bit) = 1, 则 $PC \leftarrow PC + 3 + jdisp8$	
		sfr.bit, \$addr16	4	-	11	如果 sfr.bit = 1, 则 $PC \leftarrow PC + 4 + jdisp8$	
		A.bit, \$addr16	3	8	-	如果 A.bit = 1, 则 $PC \leftarrow PC + 3 + jdisp8$	
		PSW.bit, \$addr16	3	-	9	如果 PSW.bit = 1, 则 $PC \leftarrow PC + 3 + jdisp8$	
		[HL].bit, \$addr16	3	10	11	如果 (HL).bit = 1, 则 $PC \leftarrow PC + 3 + jdisp8$	
	<b>BF</b>	saddr.bit, \$addr16	4	10	11	如果 (saddr.bit) = 0, 则 $PC \leftarrow PC + 4 + jdisp8$	
		sfr.bit, \$addr16	4	-	11	如果 sfr.bit = 0, 则 $PC \leftarrow PC + 4 + jdisp8$	
		A.bit, \$addr16	3	8	-	如果 A.bit = 0, 则 $PC \leftarrow PC + 3 + jdisp8$	
		PSW.bit, \$addr16	4	-	11	如果 PSW.bit = 0, 则 $PC \leftarrow PC + 4 + jdisp8$	
		[HL].bit, \$addr16	3	10	11	如果 (HL).bit = 0, 则 $PC \leftarrow PC + 3 + jdisp8$	
	<b>BTCLR</b>	saddr.bit, \$addr16	4	10	12	$PC \leftarrow PC + 4 + jdisp8$ , 如果 (saddr.bit) = 1, 那么复位 (saddr.bit)	
		sfr.bit, \$addr16	4	-	12	$PC \leftarrow PC + 4 + jdisp8$ , 如果 sfr.bit = 1, 那么复位 sfr.bit	
		A.bit, \$addr16	3	8	-	$PC \leftarrow PC + 3 + jdisp8$ , 如果 A.bit = 1, 那么复位 A.bit	
		PSW.bit, \$addr16	4	-	12	$PC \leftarrow PC + 4 + jdisp8$ , 如果 PSW.bit = 1, 那么复位 PSW.bit	x x x
		[HL].bit, \$addr16	3	10	12	$PC \leftarrow PC + 3 + jdisp8$ , 如果 (HL).bit = 1, 那么复位 (HL).bit	
	<b>DBNZ</b>	B, \$addr16	2	6	-	$B \leftarrow B - 1$ , 如果 $B \neq 0$ , 那么 $PC \leftarrow PC + 2 + jdisp8$	
		C, \$addr16	2	6	-	$C \leftarrow C - 1$ , 如果 $C \neq 0$ , 那么 $PC \leftarrow PC + 2 + jdisp8$	
		saddr, \$addr16	3	8	10	$(saddr) \leftarrow (saddr) - 1$ , 如果 $(saddr) \neq 0$ , 那么 $PC \leftarrow PC + 3 + jdisp8$	
CPU 控制	<b>SEL</b>	Rbn	2	4	-	$RBS1, 0 \leftarrow n$	
	<b>NOP</b>		1	2	-	无操作	
	<b>EI</b>		2	-	6	$IE \leftarrow 1$ (允许中断)	
	<b>DI</b>		2	-	6	$IE \leftarrow 0$ (禁止中断)	
	<b>HALT</b>		2	6	-	设置 HALT 模式	
	<b>STOP</b>		2	6	-	设置 STOP 模式	

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
  2. 当访问内部高速 RAM 以外的区域时

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (fcpu) 的一个周期。
  2. 该时钟周期用于内部 ROM 程序。

## 26.3 按寻址类型列出指令

## (1) 8 位指令

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

第 2 操作数 第 1 操作数	#byte	A	R <sup>#</sup>	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	\$addr16	1	无
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

注 “r = A” 除外

(2) 16 位指令

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

第 2 操作数 第 1 操作数	#word	AX	Rp <sup>※</sup>	sfrp	saddrp	!addr16	SP	无
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW <sup>※</sup>						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

注 仅当 rp = BC, DE, HL 时使用

(3) 位操作指令

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

第 2 操作数 第 1 操作数	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	\$addr16	无
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) 调用指令 / 转移指令

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

第 2 操作数 第 1 操作数	AX	!addr16	!addr11	[addr5]	\$addr16
基本指令	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
复合指令					BT BF BTCLR DBNZ

(5) 其它指令

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

## 第二十七章 电气特性 (标准产品)

**注意事项** 78K0/LC3 具有片上调试功能。不要将该产品用于大规模生产，因为在给定 Flash 重写次数的条件下，使用片上调试功能后，产品的可靠性得不到保证。NEC 电子不接受有关该产品的任何投诉。

### 最大额定值( $T_A = 25^\circ\text{C}$ ) (1/2)

参数	符号	条件	额定值	单位
供电电压	$V_{DD}$		$-0.5 \sim +6.5$	V
	$V_{SS}$		$-0.5 \sim +0.3$	V
	$AV_{REF}^{\#2}$		$-0.5 \sim V_{DD} + 0.3^{\#1}$	V
	$AV_{SS}^{\#2}$		$-0.5 \sim +0.3$	V
REGC 引脚输入电压	$V_{IREGC}$		$-0.5 \sim +3.6$ 和 $-0.5 \sim V_{DD}$	V
输入电压	$V_I$	P12, P13, P20 ~ P25, P31 ~ P34, P40, P100, P101, P112, P113, P120 ~ P124, P140 ~ P143, P150 ~ P153, X1, X2, XT1, XT2, FLMD0, RESET	$-0.3 \sim V_{DD} + 0.3^{\#1}$	V
输出电压	$V_O$		$-0.3 \sim V_{DD} + 0.3^{\#1}$	V
模拟输入电压	$V_{AN}$	AN10 ~ AN15 <sup>#2</sup>	$-0.3 \sim AV_{REF} + 0.3^{\#1}$ 和 $-0.3 \sim V_{DD} + 0.3^{\#1}$	V

注 1. 必须小于等于 6.5 V。

2. 仅对  $\mu\text{PD78F041x}$ 。

**注意事项** 任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。

**备注** 除非另外说明，复用功能引脚的特性与端口引脚的相同。

最大额定值( $T_A = 25^\circ\text{C}$ ) (2/2)

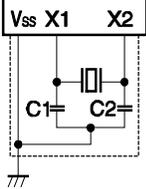
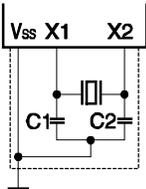
参数	符号	条件		额定值	单位
输出电流, 高	$I_{OH1}$	每一个引脚	P12, P13, P31 ~ P34, P40, P100, P101, P112, P113, P120, P140 ~ P143, P150 ~ P153	-10	mA
		所有引脚总和 -35 mA	P12, P13, P31 ~ P34, P40, P120	-25	mA
			P100, P101, P112, P113, P140 ~ P143, P150 ~ P153	-10	mA
	$I_{OH2}$	每一个引脚	P20 ~ P25	-0.5	mA
		所有引脚总和		-2	mA
输出电流, 低	$I_{OL}$	每一个引脚	P12, P13, P31 ~ P34, P40, P100, P101, P112, P113, P120, P140 ~ P143, P150 ~ P153	30	mA
		所有引脚总和 80 mA	P12, P13, P31 ~ P34, P40, P120	40	mA
			P100, P101, P112, P113, P140 ~ P143, P150 ~ P153	40	mA
		每一个引脚	P20 ~ P25	1	mA
	所有引脚总和		5	mA	
	$T_A$	正常操作模式	-40 ~ +85	$^\circ\text{C}$	
		flash 存储器编程模式			
工作环境温度	$T_{stg}$		-65 ~ +150	$^\circ\text{C}$	
储藏温度					

**注意事项** 任何一项参数哪怕是在瞬间超过最大额定值, 都会使产品质量受到影响。也就是说, 最大额定值是产品濒临物理损坏的临界点, 因而, 必须保证产品在不超过最大额定值的条件下使用。

**备注** 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

## X1 振荡器特性

(T<sub>A</sub> = -40 ~ +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = AV<sub>SS</sub> = 0 V)

振荡器	推荐电路	参数	条件	MIN.	TYP.	MAX.	单位
陶瓷振荡器		X1 时钟振荡频率 (f <sub>x</sub> ) <sup>注</sup>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2.0		10.0	MHz
			1.8 V ≤ V <sub>DD</sub> < 2.7 V	2.0		5.0	
晶体振荡器		X1 时钟振荡频率 (f <sub>x</sub> ) <sup>注</sup>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2.0		10.0	MHz
			1.8 V ≤ V <sub>DD</sub> < 2.7 V	2.0		5.0	

注 表示振荡器的特性。如需了解指令执行时间，可参见 **AC 特性**。

注意事项 1. 在使用 X1 时钟时，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。

- 连接线越短越好。
  - 连接线不应与其他信号线交叉。
  - 流经的电流变化较大的信号线不要在振荡器周围布线。
  - 要保持振荡器电容器的接地点电压与 V<sub>SS</sub> 相同。
  - 不要将电容的地信号接入大电流地。
  - 不要从振荡器获取信号。
2. 由于复位后 CPU 使用内部高速振荡时钟进行操作，因此用户要使用振荡稳定时间计数器的状态寄存器 (OSTC) 检测 X1 时钟振荡稳定时间。在充分评估了所使用的振荡器的振荡稳定时间后再确定 OSTC 和振荡稳定时间选择寄存器 (OSTS) 的振荡稳定时间。

备注 振荡器的选择和振荡器常数，用户可选择任一参数的振荡器或由振荡器制造厂商推荐。

## 内部振荡器特性

(TA = -40 ~ +85°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = AVSS = 0 V)

振荡器	Parameter	条件	MIN.	TYP.	MAX.	单位	
8 MHz 内部振荡器	内部高速振荡时钟频率(fRH) <sup># 1, 2</sup>	RSTS = 1	2.5 V ≤ VDD ≤ 5.5 V	7.6	8.0	8.4	MHz
			1.8 V ≤ VDD < 2.5 V	6.75	8.0	8.4	MHz
		RSTS = 0		2.48	5.6	9.86	MHz
240 kHz 内部振荡器	内部低速振荡时钟频率(fRL)	2.6 V ≤ VDD ≤ 5.5 V	216	240	264	kHz	
		1.8 V ≤ VDD < 2.6 V	192	240	264	kHz	

- 注 1. 仅表示振荡器的特性。如需了解指令执行时间, 可参见 **AC 特性**。  
2. 当设置 HIOTRM = 10H (±0%: 默认) 时

备注 RSTS: 内部振荡模式寄存器(RCM) 的第 7 位

## XT1 振荡器特性

(TA = -40 ~ +85°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = AVSS = 0 V)

振荡器	推荐电路	参数	条件	MIN.	TYP.	MAX.	单位
晶体振荡器		XT1 时钟振荡频率 (fXT) <sup>#</sup>		32	32.768	35	kHz

注 仅表示振荡器的特性。如需了解指令执行时间, 可参见 **AC 特性**。

注意事项 1. 在使用 XT1 时钟时, 上图中被虚线包围的部分的配线应按照如下配线方法配线, 以防止连接线电容产生不利影响。

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

2. XT1 振荡器是一个低振幅电路, 用于降低功耗, 但由于比 X1 振荡器更容易受到噪音干扰, 因此在使用 XT1 时钟时应特别注意连线方式。

备注 振荡器的选择和振荡器常数, 用户可选择任一参数的振荡器或由振荡器制造厂商推荐。

## DC 特性(1/5)

(TA = -40 ~ +85°C, 1.8 V ≤ VDD ≤ 5.5 V, AVREF ≤ VDD, VSS = AVSS = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位			
输出电流高, 高 <sup>注1</sup>	IOH1	P12, P13, P31 ~ P34, P40, P120 的每 一个引脚	4.0 V ≤ VDD ≤ 5.5 V			-3.0	mA		
			2.7 V ≤ VDD < 4.0 V			-2.5	mA		
			1.8 V ≤ VDD < 2.7 V			-1.0	mA		
		P100, P101, P112, P113, P140 ~ P143, P150 ~ P153 的每一个引脚	4.0 V ≤ VDD ≤ 5.5 V			-0.1	mA		
			2.7 V ≤ VDD < 4.0 V			-0.1	mA		
			1.8 V ≤ VDD < 2.7 V			-0.1	mA		
		P12, P13, P31 ~ P34, P40, P120 的所 有引脚 <sup>注3</sup>	4.0 V ≤ VDD ≤ 5.5 V			-20.0	mA		
			2.7 V ≤ VDD < 4.0 V			-10.0	mA		
			1.8 V ≤ VDD < 2.7 V			-5.0	mA		
		P100, P101, P112, P113, P140 ~ P143, P150 ~ P153 的所有引脚 <sup>注3</sup>	4.0 V ≤ VDD ≤ 5.5 V			-2.8	mA		
			2.7 V ≤ VDD < 4.0 V			-2.8	mA		
			1.8 V ≤ VDD < 2.7 V			-2.8	mA		
		所有引脚 <sup>注3</sup>	4.0 V ≤ VDD ≤ 5.5 V			-22.8	mA		
			2.7 V ≤ VDD < 4.0 V			-12.8	mA		
			1.8 V ≤ VDD < 2.7 V			-7.8	mA		
	IOH2	P20 ~ P25 的每一个引脚	AVREF = VDD			-0.1	mA		
输出电流, 低 <sup>注2</sup>	IOL1	P12, P13, P31 ~ P34, P40, P120 的每 一个引脚	4.0 V ≤ VDD ≤ 5.5 V			8.5	mA		
			2.7 V ≤ VDD < 4.0 V			5.0	mA		
			1.8 V ≤ VDD < 2.7 V			2.0	mA		
		P100, P101, P112, P113, P140 ~ P143, P150 ~ P153 的每一个引脚	4.0 V ≤ VDD ≤ 5.5 V			0.4	mA		
			2.7 V ≤ VDD < 4.0 V			0.4	mA		
			1.8 V ≤ VDD < 2.7 V			0.4	mA		
		P12, P13, P31 ~ P34, P40, P120 的所 有引脚 <sup>注3</sup>	4.0 V ≤ VDD ≤ 5.5 V			20.0	mA		
			2.7 V ≤ VDD < 4.0 V			15.0	mA		
			1.8 V ≤ VDD < 2.7 V			9.0	mA		
		P100, P101, P112, P113, P140 ~ P143, P150 ~ P153 的所有引脚 <sup>注3</sup>	4.0 V ≤ VDD ≤ 5.5 V			11.2	mA		
			2.7 V ≤ VDD < 4.0 V			11.2	mA		
			1.8 V ≤ VDD < 2.7 V			11.2	mA		
		所有引脚 <sup>注3</sup>	4.0 V ≤ VDD ≤ 5.5 V			31.2	mA		
			2.7 V ≤ VDD < 4.0 V			26.2	mA		
			1.8 V ≤ VDD < 2.7 V			20.2	mA		
			IOL2	P20 ~ P25 的每一个引脚	AVREF = VDD			0.4	mA

- 注 1. 即使电流从 VDD 流向输出引脚, 设备操作时的电流值也可以得到保证。
2. 即使电流从输出引脚流向 GND, 设备操作时的电流值也可以得到保证。
3. 占空比=70%时 (输出电流的时间为  $0.7 \times t$ , 不输出电流的时间为  $0.3 \times t$ , 其中  $t$  为指定时间) 的规定。可通过以下表达式计算占空比不等于 70% 时的引脚总的输出电流。
- 当 IOH 的占空比为 n%: 引脚总的输出电流 =  $(IOH \times 0.7)/(n \times 0.01)$   
 <举例> 当占空比为 50%, IOH = 20.0 mA  
 引脚总的输出电流 =  $(20.0 \times 0.7)/(50 \times 0.01) = 28.0$  mA
- 但是, 允许流入每个引脚的电流不会随占空比而变化。高于最大额定值的电流不能流入引脚。

备注 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

## DC 特性(2/5)

(TA = -40 ~ +85°C, 1.8 V ≤ VDD ≤ 5.5 V, AVREF ≤ VDD, VSS = AVSS = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
输入电压, 高	V <sub>IH1</sub>	P32, P100, P101, P112, P121 ~ P124, P140 ~ P143, P150 ~ P153	0.7V <sub>DD</sub>		V <sub>DD</sub>	V
	V <sub>IH2</sub>	P12, P13, P31, P33, P34, P40, P113, P120, RESET, EXCLK	0.8V <sub>DD</sub>		V <sub>DD</sub>	V
	V <sub>IH3</sub>	P20 ~ P25 AVREF = VDD	0.7AVREF		AVREF	V
输入电压, 低	V <sub>IL1</sub>	P32, P100, P101, P112, P121 ~ P124, P140 ~ P143, P150 ~ P153	0		0.3V <sub>DD</sub>	V
	V <sub>IL2</sub>	P12, P13, P31, P33, P34, P40, P113, P120, RESET, EXCLK	0		0.2V <sub>DD</sub>	V
	V <sub>IL3</sub>	P20 ~ P25 AVREF = VDD	0		0.3AVREF	V
输出电压, 高	V <sub>OH1</sub>	P12, P13, P31 ~ P34, P40, P120	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, I <sub>OH1</sub> = -3.0 mA		V <sub>DD</sub> - 0.7	V
			2.7 V ≤ V <sub>DD</sub> < 4.0 V, I <sub>OH1</sub> = -2.5 mA		V <sub>DD</sub> - 0.5	V
			1.8 V ≤ V <sub>DD</sub> < 2.7 V, I <sub>OH1</sub> = -1.0 mA		V <sub>DD</sub> - 0.5	V
	P100, P101, P112, P113, P140 ~ P143, P150 ~ P153	I <sub>OH1</sub> = -0.1 mA		V <sub>DD</sub> - 0.5	V	
	V <sub>OH2</sub>	P20 ~ P25 AVREF = VDD, I <sub>OH2</sub> = -0.1 mA		V <sub>DD</sub> - 0.5	V	
输出电压, 低	V <sub>OL1</sub>	P12, P13, P31 ~ P34, P40, P120	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, I <sub>OL1</sub> = 8.5 mA		0.7	V
			2.7 V ≤ V <sub>DD</sub> < 4.0 V, I <sub>OL1</sub> = 5.0 mA		0.7	V
			1.8 V ≤ V <sub>DD</sub> < 2.7 V, I <sub>OL1</sub> = 2.0 mA		0.5	V
			1.8 V ≤ V <sub>DD</sub> < 2.7 V, I <sub>OL1</sub> = 1.0 mA		0.5	V
			1.8 V ≤ V <sub>DD</sub> < 2.7 V, I <sub>OL1</sub> = 0.5 mA		0.4	V
	P100, P101, P112, P113, P140 ~ P143, P150 ~ P153	I <sub>OL1</sub> = 0.4 mA		0.4	V	
	V <sub>OL2</sub>	P20 ~ P25 AVREF = VDD, I <sub>OL2</sub> = 0.4 mA		0.4	V	

备注 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

注意事项 P122/EXCLK 的高电平和低电平输入电压在输入端口模式和外部时钟模式之间变化。

## DC 特性(3/5)

(TA = -40 ~ +85°C, 1.8 V ≤ VDD ≤ 5.5 V, AVREF ≤ VDD, VSS = AVSS = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
输入漏电流, 高	I <sub>LIH1</sub>	P12, P13, P31 ~ P34, P40, P100, P101, P112, P113, P120, P140 ~ P143, P150 ~ P153, FLMD0, RESET	V <sub>I</sub> = V <sub>DD</sub>		1	μA
	I <sub>LIH2</sub>	P20 ~ P25	V <sub>I</sub> = AV <sub>REF</sub> = V <sub>DD</sub>		1	μA
	I <sub>LIH3</sub>	P121 ~ 124 (X1, X2, XT1, XT2)	V <sub>I</sub> = V <sub>DD</sub>	I/O 端口模式	1	μA
			OSC 模式	20	μA	
输入漏电流, 低	I <sub>LIL1</sub>	P12, P13, P31 ~ P34, P40, P100, P101, P112, P113, P120, P140 ~ P143, P150 ~ P153, FLMD0, RESET	V <sub>I</sub> = V <sub>SS</sub>		-1	μA
	I <sub>LIL2</sub>	P20 ~ P25	V <sub>I</sub> = V <sub>SS</sub> , AV <sub>REF</sub> = V <sub>DD</sub>		-1	μA
	I <sub>LIL3</sub>	P121 ~ 124 (X1, X2, XT1, XT2)	V <sub>I</sub> = V <sub>SS</sub>	I/O 端口模式	-1	μA
			OSC 模式	-20	μA	
上拉电阻	R <sub>U</sub>	V <sub>I</sub> = V <sub>SS</sub>	10	20	100	kΩ
FLMD0 供电电压	V <sub>IL</sub>	正常操作模式	0		0.2V <sub>DD</sub>	V
	V <sub>IH</sub>	自编程操作模式	0.8V <sub>DD</sub>		V <sub>DD</sub>	V

**备注** 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

## DC 特性(4/5)

(T<sub>A</sub> = -40 ~ +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, AV<sub>REF</sub> ≤ V<sub>DD</sub>, V<sub>SS</sub> = AV<sub>SS</sub> = 0 V)

参数	符号	条件		MIN.	TYP.	MAX.	单位		
供电电流 注 1	I <sub>DD1</sub>	操作模式	f <sub>XH</sub> = 10 MHz 注 2, V <sub>DD</sub> = 5.0 V	方波输入		1.6	3.0	mA	
				振荡器连接		2.3	3.4		
				f <sub>XH</sub> = 10 MHz 注 2, V <sub>DD</sub> = 3.0 V	方波输入		1.5	2.9	mA
					振荡器连接		2.2	3.3	
				f <sub>XH</sub> = 5 MHz 注 2, V <sub>DD</sub> = 3.0 V	方波输入		0.9	1.7	mA
					振荡器连接		1.3	2.0	
				f <sub>XH</sub> = 5 MHz 注 2, V <sub>DD</sub> = 2.0 V	方波输入		0.7	1.4	mA
				振荡器连接		1.0	1.6		
			f <sub>RH</sub> = 8 MHz, V <sub>DD</sub> = 5.0 V 注 3			1.4	2.3	mA	
			f <sub>SUB</sub> = 32.768 kHz 注 4, V <sub>DD</sub> = 5.0 V	振荡器连接			6.7	26	μA
	I <sub>DD2</sub>	HALT 模式	f <sub>XH</sub> = 10 MHz 注 2, V <sub>DD</sub> = 5.0 V	方波输入		0.4	1.4	mA	
				振荡器连接		1.0	1.7		
				f <sub>XH</sub> = 5 MHz 注 2, V <sub>DD</sub> = 3.0 V	方波输入		0.2	0.7	mA
					振荡器连接		0.5	1.0	
			f <sub>RH</sub> = 8 MHz, V <sub>DD</sub> = 5.0 V 注 3			0.4	1.2	mA	
		f <sub>SUB</sub> = 32.768 kHz 注 4, V <sub>DD</sub> = 5.0 V	振荡器连接			2.4	22	μA	
I <sub>DD3</sub> 注 5	STOP 模式	V <sub>DD</sub> = 5.0 V			1	20	μA		
		V <sub>DD</sub> = 5.0 V, T <sub>A</sub> = -40 ~ +70°C			1	10	μA		

- 注
1. 流入内部供电电源的总电流(V<sub>DD</sub>), 包括外部操作电流和当输入引脚固定为 V<sub>DD</sub> 或 V<sub>SS</sub> 时的输入漏电流。不包括流入上拉电阻的电流和端口的输出电流。
  2. 不包括 8 MHz 内部振荡器、240 kHz 内部振荡器和 XT1 振荡的操作电流和流入 A/D 转换器、看门狗定时器、LVI 电路和 LCD 控制器/驱动器的电流。
  3. 不包括 X1 振荡、XT1 振荡和 240 kHz 内部振荡器的操作电流和流入 A/D 转换器、看门狗定时器、LVI 电路和 LCD 控制器/驱动器的电流。
  4. 不包括 X1 振荡、8 MHz 内部振荡器和 240 kHz 内部振荡器的操作电流和流入 A/D 转换器、看门狗定时器、LVI 电路和 LCD 控制器/驱动器的电流。
  5. 不包括 240 kHz 内部振荡器和 XT1 振荡的操作电流和流入 A/D 转换器、看门狗定时器、LVI 电路和 LCD 控制器/驱动器的电流。

## 备注

1. f<sub>XH</sub>: 高速系统时钟频率 (X1 时钟振荡频率或外部主系统时钟频率)
2. f<sub>RH</sub>: 内部高速振荡时钟频率
3. f<sub>SUB</sub>: 副系统时钟频率 (XT1 时钟振荡频率或外部副系统时钟频率)

## DC 特性 (5/5)

(TA = -40 ~ +85°C, 1.8 V ≤ VDD ≤ 5.5 V, AVREF ≤ VDD, VSS = AVSS = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位	
看门狗定时器操作电流	I <sub>WDT</sub> <sup>注1</sup>	240 kHz 内部低速振荡时钟操作期间		5	10	μA	
LVI 操作电流	I <sub>LVI</sub> <sup>注2</sup>			9	18	μA	
逐次逼近型 A/D 转换器操作电流	I <sub>ADC1</sub> <sup>注3</sup>	2.3 V ≤ AVREF ≤ VDD		0.86	1.9	mA	
LCD 操作电流	I <sub>LCD1</sub> <sup>注4</sup>	LCD 显示关闭 (LCDON = 0, SCOC = 1)	VDD = 5.0 V		3.0	8.0	μA
			VDD = 3.0 V		2.0	5.0	μA
	I <sub>LCD2</sub> <sup>注4</sup>	LCD 显示打开 (LCDON = 1, SCOC = 1)	VDD = 5.0 V		3.0	8.0	μA
			VDD = 3.0 V		2.0	5.0	μA

- 注
1. 仅包括流过看门狗定时器（包括 240 kHz 内部振荡器的操作电流）的电流。当看门狗定时器运行在 HALT 模式或 STOP 模式时，78K0/LC3 的电流值通过 I<sub>WDT</sub> 加 I<sub>DD2</sub> 或 I<sub>DD3</sub> 获得。
  2. 仅包括流过 LVI 电路的电流。当 LVI 电路运行在 HALT 模式或 STOP 模式时，78K0/LE2 的电流值通过 I<sub>LVI</sub> 加 I<sub>DD2</sub> 或 I<sub>DD3</sub> 获得。
  3. 仅包括流过 A/D 转换器的电流。当 A/D 转换器运行在停止模式或 HALT 模式时，78K0/LC3 的电流值通过 I<sub>ADC1</sub>、I<sub>ADC2</sub> 或 I<sub>ADC3</sub> 加 I<sub>DD1</sub> 或 I<sub>DD2</sub> 获得。
  4. 仅包括流过 LCD 控制器/驱动器的电流。不包括流过 LCD 分压电阻的电流。78K0/LC3 的电流值通过供电电流(I<sub>DD1</sub>, I<sub>DD2</sub> 或 I<sub>DD3</sub>) 加 LCD 工作电流 (I<sub>LCD1</sub>, 或 I<sub>LCD2</sub>) 获得。

## AC 特性

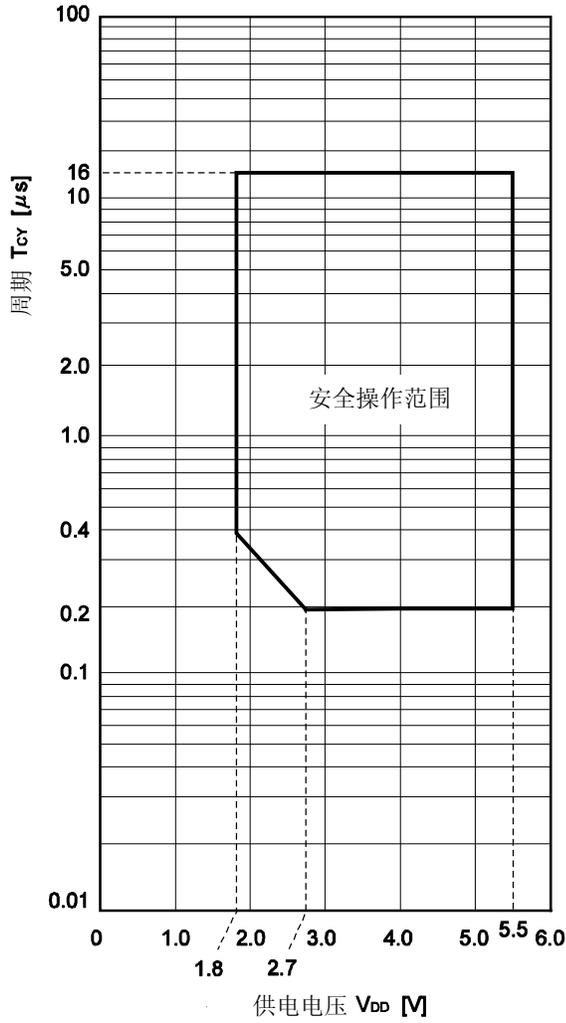
## (1) 基本操作

 $(T_A = -40 \sim +85^\circ\text{C}, 1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}, V_{SS} = AV_{SS} = 0\text{ V})$ 

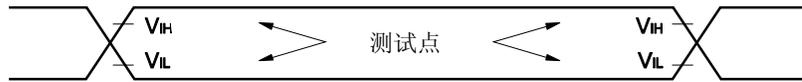
参数	符号	条件	MIN.	TYP.	MAX.	单位	
指令周期 (指令最短执行时间)	$T_{CY}$	主系统时钟 ( $f_{XP}$ ) 操作	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0.2		16	$\mu\text{s}$
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	0.4		16	$\mu\text{s}$
		副系统时钟 ( $f_{SUB}$ ) 操作		114	122	125	$\mu\text{s}$
外围硬件时钟频率	$f_{PRS}$	XSEL = 1	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			10	MHz
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			5	MHz
		XSEL = 0	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	7.6		8.4	MHz
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}^{\#1}$	6.75		8.4	MHz
外部主系统时钟频率	$f_{EXCLK}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.0		10.0	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	2.0		5.0	MHz	
外部主系统时钟 输入高电平宽度, 低电平宽度	$t_{EXCLKH},$ $t_{EXCLKL}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	48		500	ns	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	96		500	ns	
TI000 输入高电平宽度, 低电平 宽度	$t_{TIH0},$ $t_{TIL0}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$2/f_{sam} +$ $0.2^{\#2}$			$\mu\text{s}$	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$2/f_{sam} +$ $0.5^{\#2}$			$\mu\text{s}$	
TI52 输入频率	$f_{TI5}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			16	MHz	
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			10	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			5	MHz	
TI52 输入高电平宽度, 低电平宽 度	$t_{TIH5},$ $t_{TIL5}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	31.25			ns	
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	50			ns	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	100			ns	
中断输入高电平宽度, 低电平宽 度	$t_{INTH},$ $t_{INTL}$		1			$\mu\text{s}$	
按键返回输入低电平宽度	$t_{KR}$		250			ns	
RESET 低电平宽度	$t_{RSL}$		10			$\mu\text{s}$	

- 注
1. 表征主系统时钟频率。用外围功能将时钟分频器设置为  $f_{RH}/2$  或更小。
  2. 使用分频模式寄存器 00(PRM00)的位 0 和位 1(PRM00, PRM001)可以选择选择  $f_{sam} = f_{PRS}, f_{PRS}/4,$   $f_{PRS}/256$ 。注意当选择 TI000 有效沿作为计数时钟时,  $f_{sam} = f_{PRS}$ 。

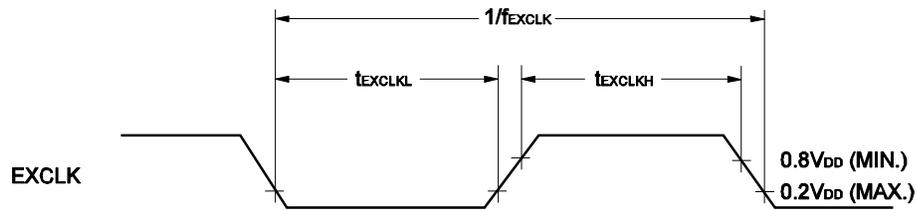
T<sub>cy</sub> vs. V<sub>DD</sub> (主系统时钟操作)



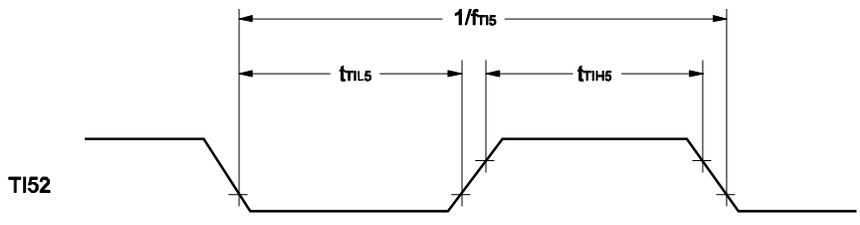
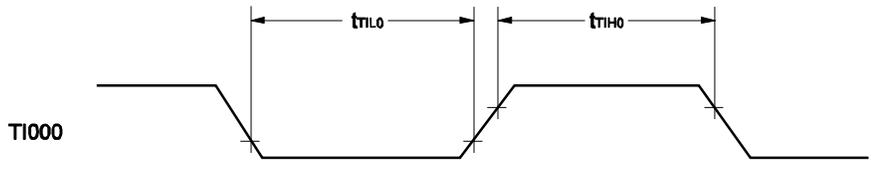
AC 时序测试点(不包括外部主系统时钟)



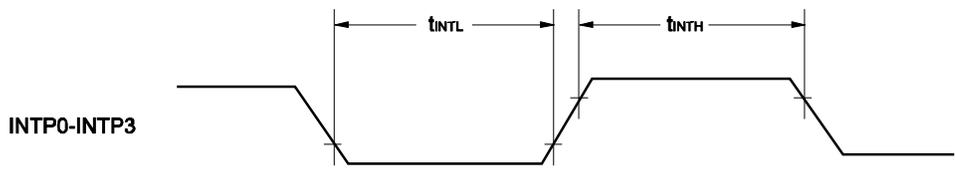
外部主系统时钟时序



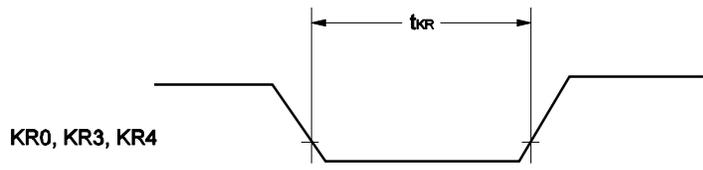
TI 时序



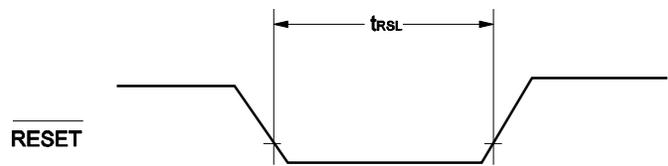
中断请求输入时序



按键中断输入时序



RESET 输入时序



## (2) 曼彻斯特码发生器

(TA = -40 ~ +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = AV<sub>SS</sub> = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
传输率					250	kbps

## (3) 串行接口

(TA = -40 ~ +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = AV<sub>SS</sub> = 0 V)

## (a) UART6 (专用波特率发生器输出)

参数	符号	条件	MIN.	TYP.	MAX.	单位
传输率					625	kbps

## (b) UART0 (专用波特率发生器输出)

参数	符号	条件	MIN.	TYP.	MAX.	单位
传输率					625	kbps

10-位逐次逼近型 A/D 转换器特性 (仅对 $\mu$ PD78F041x)

(TA = -40 ~ +85°C, 2.3 V ≤ AVREF ≤ VDD ≤ 5.5 V, VSS = AVSS = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	RES				10	位
总误差 <sup>注1,2</sup>	AINL	4.0 V ≤ AVREF ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AVREF < 4.0 V			±0.6	%FSR
		2.3 V ≤ AVREF < 2.7 V			±1.2	%FSR
转换时间	tCONV	4.0 V ≤ AVREF ≤ 5.5 V	6.1		36.7	μs
		2.7 V ≤ AVREF < 4.0 V	12.2		36.7	μs
		2.3 V ≤ AVREF < 2.7 V	27		66.6	μs
零度误差 <sup>注1,2</sup>	EzS	4.0 V ≤ AVREF ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AVREF < 4.0 V			±0.6	%FSR
		2.3 V ≤ AVREF < 2.7 V			±0.6	%FSR
满度误差 <sup>注1,2</sup>	EFS	4.0 V ≤ AVREF ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AVREF < 4.0 V			±0.6	%FSR
		2.3 V ≤ AVREF < 2.7 V			±0.6	%FSR
积分非线性误差 <sup>注1</sup>	ILE1	4.0 V ≤ AVREF ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AVREF < 4.0 V			±4.5	LSB
		2.3 V ≤ AVREF < 2.7 V			±6.5	LSB
微分非线性误差 <sup>注1</sup>	DLE1	4.0 V ≤ AVREF ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AVREF < 4.0 V			±2.0	LSB
		2.3 V ≤ AVREF < 2.7 V			±2.0	LSB
模拟输入电压	VAIN1		AVSS		AVREF	V

- 注 1. 不包括量化误差 (±1/2 LSB)  
 2. 该值表示满度的百分比 (%FSR)

## LCD 特性

## (1)电阻分压方式

(a) 静态显示模式 ( $T_A = -40 \sim +85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$ ,  $V_{\text{SS}} = 0\text{ V}$ )<sup>註3</sup>

参数	符号	条件	MIN.	TYP.	MAX.	单位
LCD 驱动电压	$V_{\text{LCD}}$	注 3			$V_{\text{DD}}$	V
LCD 分压电阻 <sup>註1</sup>	$R_{\text{LCD}}$		60	100	150	k $\Omega$
LCD 输出电阻 <sup>註2</sup> (Common)	$R_{\text{ODC}}$				40	k $\Omega$
LCD 输出电阻 <sup>註2</sup> (Segment)	$R_{\text{ODS}}$				200	k $\Omega$

(b) 1/3 偏压方式 ( $T_A = -40 \sim +85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$ ,  $V_{\text{SS}} = 0\text{ V}$ )<sup>註3</sup>

参数	符号	条件	MIN.	TYP.	MAX.	单位
LCD 驱动电压	$V_{\text{LCD}}$	注 3			$V_{\text{DD}}$	V
LCD 分压电阻 <sup>註1</sup>	$R_{\text{LCD}}$		60	100	150	k $\Omega$
LCD 输出电阻 <sup>註2</sup> (Common)	$R_{\text{ODC}}$				40	k $\Omega$
LCD 输出电阻 <sup>註2</sup> (Segment)	$R_{\text{ODS}}$				200	k $\Omega$

(c) 1/2 偏压方式 ( $T_A = -40 \sim +85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$ ,  $V_{\text{SS}} = 0\text{ V}$ )<sup>註3</sup>1/4 偏压方式 ( $T_A = -40 \sim +85^\circ\text{C}$ ,  $4.5\text{ V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$ ,  $V_{\text{SS}} = 0\text{ V}$ )<sup>註3</sup>

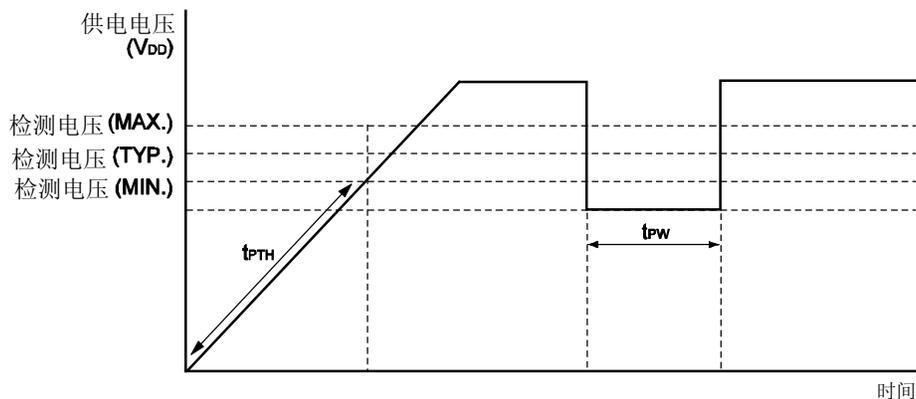
参数	符号	条件	MIN.	TYP.	MAX.	单位
LCD 驱动电压	$V_{\text{LCD}}$	注 3			$V_{\text{DD}}$	V
LCD 分压电阻 <sup>註1</sup>	$R_{\text{LCD}}$		60	100	150	k $\Omega$
LCD 输出电阻 <sup>註2</sup> (Common)	$R_{\text{ODC}}$				40	k $\Omega$
LCD 输出电阻 <sup>註2</sup> (Segment)	$R_{\text{ODS}}$				200	k $\Omega$

- 注
1. 仅对内部分压电阻方式。
  2. 输出电阻是指链接在  $V_{\text{LC0}}$ ,  $V_{\text{LC1}}$ ,  $V_{\text{LC2}}$  和  $V_{\text{SS}}$  引脚之间,和  $\text{SEG}$  和  $\text{COM}$  引脚之间的电阻。
  3. 按照下列条件设置  $VAON$ 。
    - <当设置为静态模式时>
      - 当  $2.0\text{V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$  时:  $VAON = 0$
      - 当  $1.8\text{V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 3.6\text{ V}$  时:  $VAON = 1$
    - <当设置为 1/3 偏压模式时>
      - 当  $2.5\text{V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$  时:  $VAON = 0$
      - 当  $1.8\text{V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 3.6\text{ V}$  时:  $VAON = 1$
    - <当设置为 1/2 偏压模式时>
      - 当  $2.7\text{V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$  时:  $VAON = 0$
      - 当  $1.8\text{V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 3.6\text{ V}$  时:  $VAON = 1$
    - <当设置为 1/4 偏压模式时>
      - 当  $4.5\text{V} \leq V_{\text{LCD}} \leq V_{\text{DD}} \leq 5.5\text{ V}$  时:  $VAON = 0$

1.59 V POC 电路特性 ( $T_A = -40 \sim +85^\circ\text{C}$ ,  $V_{SS} = 0\text{ V}$ )

参数	符号	条件	MIN.	TYP.	MAX.	位
检测电压	$V_{POC}$		1.44	1.59	1.74	V
供电电压上升斜率	$t_{PTH}$	$V_{DD}: 0\text{ V} \rightarrow V_{POC}$ 变化斜率	0.5			V/ms
最小脉宽	$t_{PW}$		200			$\mu\text{s}$

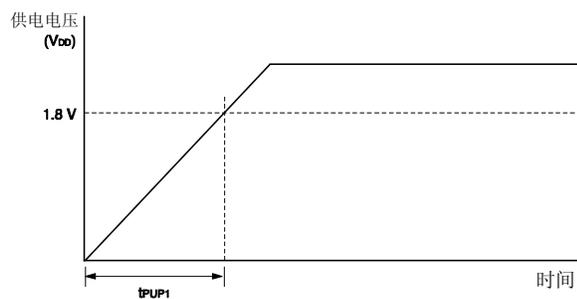
## POC 电路时序

供电电压上升时间 ( $T_A = -40 \sim +85^\circ\text{C}$ ,  $V_{SS} = 0\text{ V}$ )

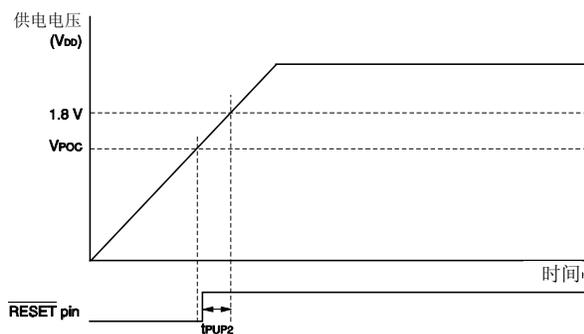
参数	符号	条件	MIN.	TYP.	MAX.	单位
上升至 1.8 V ( $V_{DD}(\text{MIN.})$ ) 所需的最长时间 ( $V_{DD}: 0\text{ V} \rightarrow 1.8\text{ V}$ )	$t_{PUP1}$	当没有使用 RESET 输入时, $\overline{\text{POCMODE}}$ (选项字节) = 0			3.6	ms
上升至 1.8 V ( $V_{DD}(\text{MIN.})$ ) 所需的最长时间 (释放 RESET 输入 $\rightarrow V_{DD}: 1.8\text{ V}$ )	$t_{PUP2}$	当使用 RESET 输入时, $\overline{\text{POCMODE}}$ (选项字节) = 0			1.9	ms

## 供电电压上升时间时序

- 当没有使用 RESET 引脚输入时



- 当使用 RESET 引脚输入时

2.7 V POC 电路特性 ( $T_A = -40 \sim +85^\circ\text{C}$ ,  $V_{SS} = 0\text{ V}$ )

参数	符号	条件	MIN.	TYP.	MAX.	单位
供电电压应用中的检测电压	$V_{DDPOC}$	$\overline{\text{POCMODE}}$ (可选字节) = 1	2.50	2.70	2.90	V

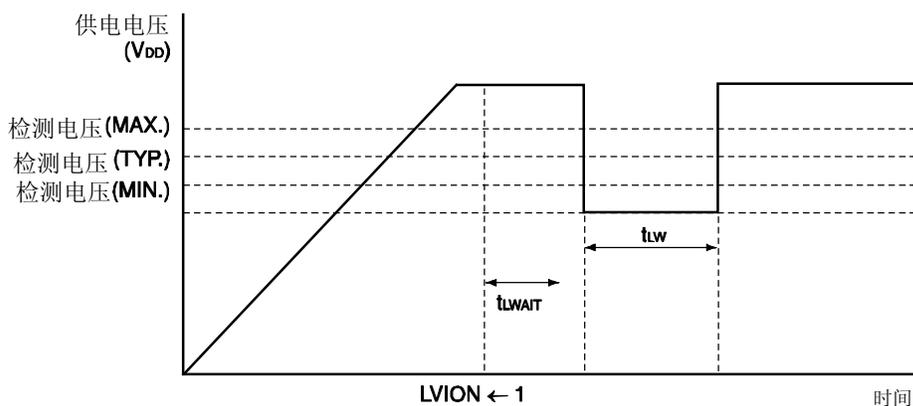
LVI 电路特性 ( $T_A = -40 \sim +85^\circ\text{C}$ ,  $V_{POC} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )

参数	符号	条件	MIN.	TYP.	MAX.	单位	
检测电压	供电电压电平	$V_{LV10}$		4.14	4.24	4.34	V
		$V_{LV11}$		3.99	4.09	4.19	V
		$V_{LV12}$		3.83	3.93	4.03	V
		$V_{LV13}$		3.68	3.78	3.88	V
		$V_{LV14}$		3.52	3.62	3.72	V
		$V_{LV15}$		3.37	3.47	3.57	V
		$V_{LV16}$		3.22	3.32	3.42	V
		$V_{LV17}$		3.06	3.16	3.26	V
		$V_{LV18}$		2.91	3.01	3.11	V
		$V_{LV19}$		2.75	2.85	2.95	V
		$V_{LV10}$		2.60	2.70	2.80	V
		$V_{LV11}$		2.45	2.55	2.65	V
		$V_{LV12}$		2.29	2.39	2.49	V
		$V_{LV13}$		2.14	2.24	2.34	V
		$V_{LV14}$		1.98	2.08	2.18	V
		$V_{LV15}$		1.83	1.93	2.03	V
外部输入引脚 <sup>注1</sup>	EXLVI	$EXLVI < V_{DD}$ , $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	1.11	1.21	1.31	V	
最小脉宽	$t_{LW}$		200			$\mu\text{S}$	
操作振荡稳定等待时间 <sup>注2</sup>	$t_{LWAIT}$				10	$\mu\text{S}$	

- 注 1. 使用 EXLVI/P120/INTP0 引脚。  
 2. 从低电压检测寄存器(LVIM)的位 7(LVION) 设置为 1 到操作稳定所需的时间。

备注  $V_{LV1(n-1)} > V_{LV1n}$ :  $n = 1 \sim 15$

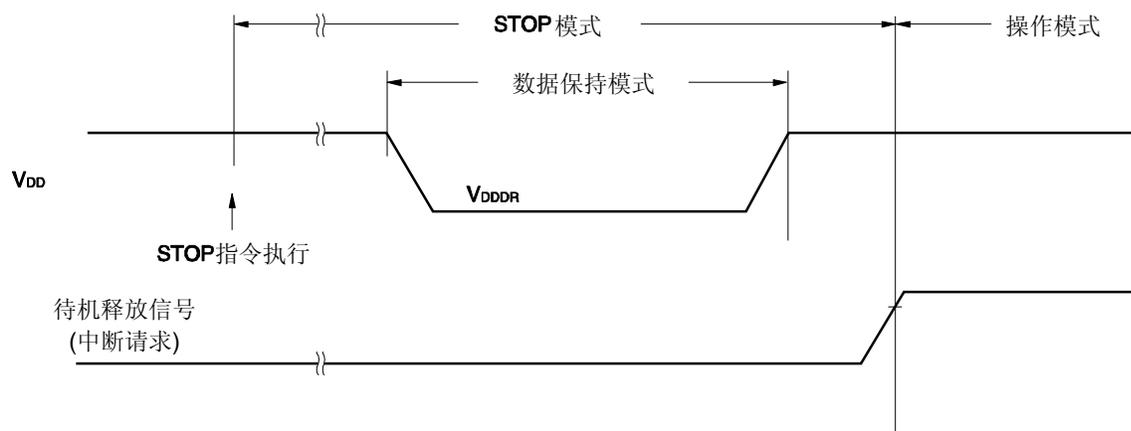
## LVI 电路时序



数据存储器在 STOP 模式下低供电电压时的数据保持特性( $T_A = -40 \sim +85^\circ\text{C}$ )

参数	符号	条件	MIN.	TYP.	MAX.	单位
数据保持供电电压	$V_{DDDR}$		1.44 <sup>注</sup>		5.5	V

注 该值根据 POC 检测电压而变化。当电压降低时，一直保持该数据直至 POC 复位有效，但当 POC 复位有效时数据不能保持。



## Flash 存储器编程特性

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

## • 基本特性

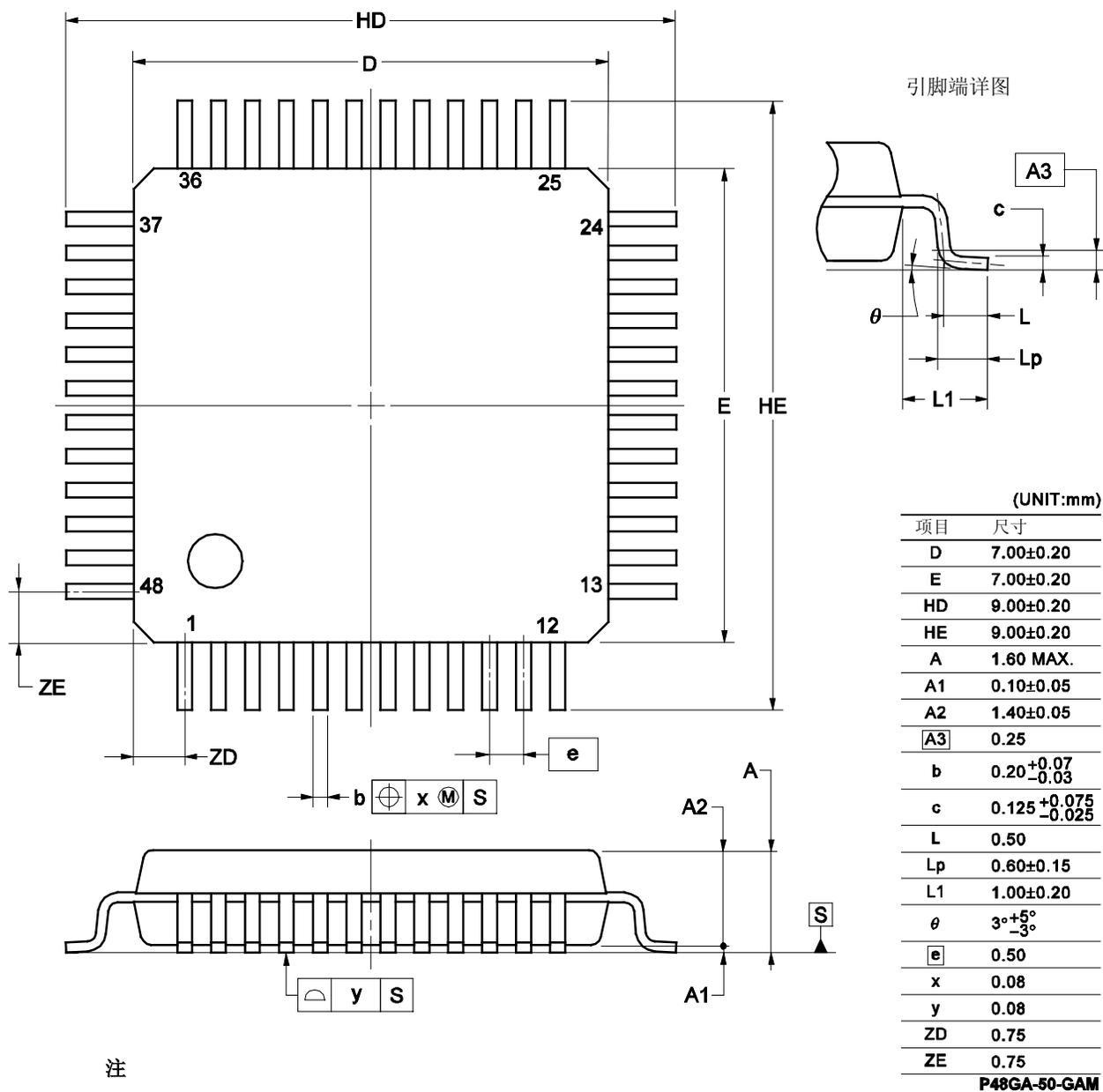
参数	符号	条件	MIN.	TYP.	MAX.	单位
$V_{DD}$ 供电电流	$I_{DD}$			4.5	11.0	mA
擦除时间 <sup>注1</sup>	所有 block	$T_{eraca}$		20	200	ms
	Block 单元	$T_{erasa}$		20	200	ms
写时间(8 位)	$T_{wrwa}$			10	100	$\mu\text{s}$
每片重写次数	$C_{erwr}$	保持时间: 15 年 擦除 1 次 + 擦除后写 1 次 = 1 次重写 <sup>注2</sup>	1000			次数

注 1. 不包括擦除前的预写时间和擦除验证时间 (写回时间)。  
2. 在出货后第一次对产品进行写操作时，“擦除 → 写”和“只写”作为一次重写。

备注  $f_{XP}$ : 主系统时钟振荡频率

## 第二十八章 封装图

48 引脚塑封 LQFP (密脚距) (7×7)



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的 0.08mm 内

© NEC Electronics Corporation 2005

## 第二十九章 等待注意事项

### 29.1 等待注意事项

本产品有两种内部系统总线。

一种是 CPU 总线，另一种是连接低速外围硬件的外设总线。

由于 CPU 总线时钟与外设总线时钟是异步的，因此如果对 CPU 的访问与对外设的访问冲突时，可能会有不期望的非法数据产生。

因此当访问可能会产生冲突的外设时，CPU 反复执行处理，直至出现正确数据。

这样，CPU 并不执行下一条指令，而是等待。如果出现这种情况，指令执行所需的时钟数会依等待的时钟数而增加（如需了解等待时钟数的详细内容，参见表 29-1）。在进行实时操作时必须要注意这一点。

## 29.2 产生等待的外围硬件

表 29-1 列出了当被 CPU 访问时发出等待请求的寄存器和 CPU 等待的时钟数。

表 29-1. 产生等待的寄存器和 CPU 等待时钟数

外围硬件	寄存器	访问方式	等待时钟个数
串行接口 UART0	ASIS0	读	1 个时钟 (恒定)
串行接口 UART6	ASIS6	读	1 个时钟 (恒定)
10-位逐次逼近 型 A/D 转换器	ADM	写	1 ~ 5 个时钟 (当选择 $f_{AD} = f_{PRS}/2$ 时)
	ADS	写	1 ~ 7 个时钟 (当选择 $f_{AD} = f_{PRS}/3$ 时)
	ADPC	写	1 ~ 9 个时钟 (当选择 $f_{AD} = f_{PRS}/4$ 时)
	ADCR	读	2 ~ 13 个时钟 (当选择 $f_{AD} = f_{PRS}/6$ 时) 2 ~ 17 个时钟 (当选择 $f_{AD} = f_{PRS}/8$ 时) 2 ~ 25 个时钟 (当选择 $f_{AD} = f_{PRS}/12$ 时)
<p>上述时钟个数是指在选择相同的源时钟作为 <math>f_{CPU}</math> 和 <math>f_{PRS}</math> 时的值。可以根据以下表达式和条件计算等待时钟的个数。</p> <p>&lt;计算等待时钟数&gt;</p> <ul style="list-style-type: none"> <li>等待时钟数 = <math>(2f_{CPU}/f_{AD}) + 1</math> <ul style="list-style-type: none"> <li>* 如果等待时钟的个数 <math>\leq 0.5</math> 则将小数点以后的部分舍去，而如果等待时钟的个数 <math>&gt; 0.5</math>，则进行四舍五入</li> </ul> </li> </ul> <p><math>f_{AD}</math>: A/D 转换时钟的频率 (<math>f_{PRS}/2 \sim f_{PRS}/12</math>)</p> <p><math>f_{CPU}</math>: CPU 时钟频率</p> <p><math>f_{PRS}</math>: 外围硬件时钟频率</p> <p><math>f_{XP}</math>: 主系统时钟频率</p> <p>&lt;等待时钟的最大/最小个数所需的条件&gt;</p> <ul style="list-style-type: none"> <li>最大个数: CPU 的最高速度(<math>f_{XP}</math>), A/D 转换时钟的最低速度(<math>f_{PRS}/12</math>)</li> <li>最小个数: CPU 的最低速度(<math>f_{SUB}/2</math>), A/D 转换时钟的最高速度(<math>f_{PRS}/2</math>)</li> </ul>			

**注意事项** 当 CPU 工作在副系统时钟和外围硬件时钟停止的状态下，不要使用需等待请求的访问模式访问上述寄存器。

**备注** 这里的时钟是指 CPU 时钟 ( $f_{CPU}$ )。

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