

To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

用户手册

RENESAS

V850ES/KE2

32 位单片微控制器

硬件

μ PD70F3726

文档编号 U17705CA2V0UD00 (第二版)

发行日期 2007 年 12 月 CP(K)

© NEC Electronics Corporation 2005

日本印刷

[备忘录]

CMOS 设备的注释

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 $V_{IL}(\text{MAX})$ 和 $V_{IH}(\text{MIN})$ 之间, 设备可能发生错误。在输入电平固定时以及输入电平从 $V_{IL}(\text{MAX})$ 过渡到 $V_{IH}(\text{MIN})$ 时的传输期间, 要防止散射噪声影响设备。

②未使用的输入引脚的处理

CMOS 设备的输入端保持开路可能导致误操作。如果一个输入引脚未被连接, 则由于噪音等原因可能会产生内部输入电平, 从而导致误操作。CMOS 设备的操作特性与 Bipolar 或 NMOS 设备不同。

CMOS 设备的输入电平必须借助上拉或下拉电路固定在高电平或低电平。每一个未使用的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格和说明。

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。操作员应佩戴静电消除腕带以保证良好接地。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。

注意：μPD70F3726 使用 Silicon Storage Technology, Inc. 公司许可的 SuperFlash® 技术。

IECUBE 是日电电子有限公司在日本和德国的注册商标。

MINICUBE 是日电电子公司（德国）在美国的注册商标。

EEPROM 是日电电子有限公司的商标。

Windows and Windows NT 是微软公司在美国和/或其他国家的注册商标或商标。

SuperFlash 是 Silicon Storage Technology, Inc.公司在包括美国和日本在内的几个国家的注册商标。

PC/AT 是 International Business Machines Corporation 的商标。

SPARCstation 是 SPARC International, Inc的商标。

Solaris 和**SunOS**是Sun Microsystems, Inc的商标

TRON 是实时操作系统核心的缩写。

ITRON 是工业实时操作系统核心的缩写。

- 本文档中的信息为2006年5月更新。今后文档中的信息被更新时，不会有特别的提示，因此，如阁下正使用本文档中所描述的器件进行设计开发，请您留意日电电子有限公司最新发布的数据表单(data sheet)、数据文本(data book)或其他相关文档，以获取最新公布的产品信息。并非在每个国家都可获得全部型号的产品。请联系NEC Electronics 销售代理，了解供货和其他相关信息。
- 未经NEC Electronics的书面许可，不能对本文档复制。本文档出现的任何错误，NEC Electronics不承担责任。
- 如果用户在使用本文档列出的NEC Electronics 半导体产品或通过其他途径使用这些产品时，产生侵犯专利、版权以及其他侵犯第三方知识产权的行为，NEC Electronics 不承担责任。对于NEC Electronics 及其他子公司的任何专利、版权以及其他知识产权，NEC Electronics 没有以许可、明示、暗示以及其他任何方式授权。
- 文档中电路、软件和其他相关信息的描述，用来说明半导体产品操作和应用的例子。客户在使用这些电路、软件和信息时负全责。客户或第三方在使用这些电路、软件和信息时造成的损失，NEC Electronics不承担责任。
- NEC Electronics尽力提高半导体产品的质量、可靠性和安全性，但请客户理解错误是不可能完全避免的。为了尽可能减少由于NEC Electronics半导体产品所带来的个人财产及人身安全（包括死亡）的风险，客户在设计过程中应加强安全措施，如容错、耐火性和自检等。
- NEC Electronics产品分为以下三个质量等级：
“标准”、“专业”、“特级”。“特级”质量等级仅用于客户定制的半导体产品。一种半导体产品的应用

主要依据它的质量等级。客户在使用某种半导体产品之前应先了解它的质量等级。

“标准”：计算机，办公设备，通信设备，测试设备，视频音频设备，家用电子产品，机械工具，个人电子设备和工业机器人

“专业”：运输设备（汽车，火车，轮船等），交通控制系统，防灾系统，反犯罪系统，安全设备和医疗设备（不是专用于生命救护的设备）

“特级”：飞机，航空设备，水下中继器，核反应堆控制系统，生命救护系统和用于生命救护的医疗设备等。

除非在NEC Electronics规格书或规格手册中特别规定，一般的NEC Electronics产品的质量等级都是“标准”的。如果客户希望在不受NEC Electronics产品应用限制的条件下使用NEC Electronics半导体产品，必须事先与NEC Electronics销售代理联系，以确定NEC Electronics是否支持该应用环境。

（注释）

（1）“NEC Electronics”在这里是指NEC Electronics Corporation和它的主要子公司。

（2）“NEC Electronics半导体产品”是指由NEC Electronics或为NEC Electronics开发和制造的半导体产品（如上述定义）。

M8E 02.11-1

引言

读者对象 本手册适用于那些希望了解 V850ES/KE2 产品的功能并准备利用其进行应用系统开发的用户。

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 V850ES/KE2 产品手册主要分为两个部分： 硬件(本手册)和架构(**V850ES 用户架构手册**)。

硬件	架构
<ul style="list-style-type: none">• 引脚功能• CPU 功能• 片上外围功能• Flash 存储器编程• 电气特性	<ul style="list-style-type: none">• 数据类型• 寄存器设置• 指令格式和指令集• 中断和异常• 流水线操作

手册使用方法 在阅读本手册前，读者应掌握电子工程、逻辑电路和微控制器等方面的一般知识。

如何获悉 V850ES/KE2 的全部功能
→ 请依照**目录**阅读本手册。

如何获取某寄存器的详细信息
→ 请参考**附录 C 寄存器索引**

寄存器格式
→ 寄存器格式图中，一些寄存器位所对应的数字被尖括号(<>)包围，这表示该位的名
字是设备文件中的保留字。

如何了解指令的详细功能
→ 请参阅**V850ES 架构用户手册**。

如何了解 V850ES/KE2 产品的电气指标
→ 请参考第二十三章**电器特性**。

在本手册中“xxx 寄存器的 yyy 位”是以“xxx.yyy 位”的形式表示的。需要注意的是编译器和汇编器是不能识别“xxx.yyy”的形式的描述的。

文档较前一版本的主要修改之处用<R> 符号标注。

规定

数据规则:	数据的高位部分在左侧，低位部分在右侧。
低电平有效表示方法:	$\overline{\text{xxx}}$ (在引脚或信号名称上加上划线)
存储器映射地址:	顶部: 高地址, 底部: 低地址
注:	文中用"注"标注的相关术语的脚注
注意事项:	需要特别注意的信息
备注:	补充信息
数的标识法:	二进制... xxxx 或 xxxxB 十进制... xxxx 十六进制... xxxxH
2 的幂表示(地址空间, 内存容量):	
	K (千): $2^{10} = 1, 024$
	M (兆): $2^{20} = 1, 024^2$
	G (吉): $2^{30} = 1, 024^3$

相关文档

本手册中指出的相关文档包括了最初的版本，但未注明。

V850ES/KE2 产品相关的文档

文档名称	文档编号
V850ES 架构用户手册	U15943E
V850ES/KE2 硬件用户手册	本手册

开发工具相关的文档（用户手册）

文档名称	文档编号	
QB-V850ESX1H 在线仿真器	U17214E	
QB-V850MINI 片上调试仿真器	U17638E	
QB-MINI2 带 flash 存储器编成功能的片上调试仿真器	准备中	
CA850 Ver. 3.00 C 编译器安装包	操作	U17293E
	C 语言	U17291E
	汇编语言	U17292E
	链接伪指令	U17294E
PM+ Ver. 6.20 工程管理器	U17990E	
ID850QB Ver. 3.20 集成调试器	操作	U17964E
SM850 Ver. 2.50 系统软仿真器	操作	U16218E
SM850 Ver. 2.00 或更高版系统软仿真器	开放式外部用户接口规范	U14873E
RX850 Ver. 3.20 实时操作系统	基础	U13430E
	安装	U17419E
	技术	U13431E
	任务调试器	U17420E
RX850 Pro Ver. 3.20 实时操作系统	基础	U13773E
	安装	U17421E
	技术	U13772E
	任务调试器	U17422E
AZ850 Ver. 3.30 系统性能分析器	U17423E	
PG-FP4 Flash 存储器编程器	U15260E	

目录

第一章 绪论	17
1.1 V850ES/Kx2 产品列表	17
1.2 性能	18
1.3 应用领域	19
1.4 订货信息	19
1.5 引脚图（俯视图）	20
1.6 功能模块配置	22
1.7 性能指标	25
第二章 引脚功能	26
2.1 引脚功能列表	26
2.2 I/O 引脚电路和不使用引脚的推荐连接法	30
2.3 引脚 I/O 电路	32
第三章 CPU 功能	34
3.1 特点	34
3.2 CPU 寄存器组	35
3.2.1 程序寄存器组	36
3.2.2 系统寄存器组	37
3.3 操作模式	43
3.4 地址空间	44
3.4.1 CPU 地址空间	44
3.4.2 CPU 地址空间的绕回	45
3.4.3 存储器映射	46
3.4.4 区域	48
3.4.5 地址空间的推荐使用方法	50
3.4.6 片上外围 I/O 区域	52
3.4.7 特殊寄存器	58
3.4.8 注意事项	61
第四章 端口功能	65
4.1 特点	65
4.2 端口的基本配置	65
4.3 端口配置	66
4.3.1 端口 0	72
4.3.2 端口 3	74
4.3.3 端口 4	80
4.3.4 端口 5	82
4.3.5 端口 7	85
4.3.6 端口 9	86
4.3.7 端口 CM	92
4.3.8 端口 DL	94
4.4 框图	96
4.5 当使用复用功能时的端口寄存器设置	114
4.6 注意事项	118

4.6.1	端口 n 寄存器 (Pn) 位操作指令的注意事项	118
4.6.2	滞后性	119
第五章	时钟发生器功能	120
5.1	概述	120
5.2	结构图	121
5.3	寄存器	123
5.4	操作	127
5.4.1	各时钟的操作	127
5.4.2	时钟输出功能	127
5.4.3	外部时钟输入功能	127
5.5	PLL 功能	128
5.5.1	概述	128
5.5.2	寄存器	128
5.5.3	使用方法	129
第六章	16 位定时器/事件计数器 P (TMP)	130
6.1	概述	130
6.2	功能	130
6.3	配置	131
6.4	寄存器	133
6.5	操作	144
6.5.1	间隔定时器模式 (TP0MD2 ~ TP0MD0 位 = 000)	145
6.5.2	外部事件计数模式 (TP0MD2 ~ TP0MD0 位 = 001)	155
6.5.3	外部触发脉冲输出模式 (TP0MD2 ~ TP0MD0 位 = 010)	163
6.5.4	单脉冲输出模式 (TP0MD2 ~ TP0MD0 位 = 011)	175
6.5.5	PWM 输出模式 (TP0MD2 ~ TP0MD0 位 = 100)	182
6.5.6	自由运行定时器模式 (TP0MD2 ~ TP0MD0 位 = 101)	191
6.5.7	脉宽测量模式 (TP0MD2 ~ TP0MD0 位 = 110)	208
6.5.8	定时器输出操作	214
6.6	捕捉触发输入引脚(TIP0a)的噪声消除	215
6.7	注意事项	217
第七章	16 位定时器/事件计数器 0	218
7.1	功能	218
7.2	配置	219
7.3	寄存器	224
7.4	操作	231
7.4.1	间隔定时器操作	231
7.4.2	方波输出操作	234
7.4.3	外部事件计数器的操作	237
7.4.4	通过 TI010 引脚有效边沿进入的清零和开始模式的操作	240
7.4.5	自由运行定时器操作	256
7.4.6	PPG 输出操作	265
7.4.7	单脉冲输出操作	268
7.4.8	脉冲宽度测量操作	273
7.5	TM01 的特殊应用	281
7.5.1	TM01 操作过程中改写 CR011 寄存器	281

7.5.2	设置 LVS01 和 LVR01 位	281
7.6	注意事项	283
第八章	8 位定时器/事件计数器 5	287
8.1	功能	287
8.2	配置	288
8.3	寄存器	291
8.4	运行	294
8.4.1	作为间隔定时器运行	294
8.4.2	作为外部事件计数器运行	296
8.4.3	方波输出操作	297
8.4.4	8 位 PWM 输出操作	299
8.4.5	作为间隔定时器运行（16 位）	302
8.4.6	作为外部事件计数器运行（16 位）	304
8.4.7	方波输出操作（16 位分辨率）	305
8.4.8	注意事项	306
第九章	8 位定时器 H	307
9.1	功能	307
9.2	配置	307
9.3	寄存器	310
9.4	运行	314
9.4.1	作为间隔定时器/方波输出	314
9.4.2	PWM 输出模式操作	317
9.4.3	载波发生器模式操作	323
第十章	间隔定时器和钟表定时器	330
10.1	间隔定时器 BRG	330
10.1.1	功能	330
10.1.2	配置	330
10.1.3	寄存器	332
10.1.4	操作	334
10.2	钟表定时器	335
10.2.1	功能	335
10.2.2	配置	335
10.2.3	寄存器	336
10.2.4	操作	338
10.3	注意事项	339
第十一章	看门狗定时器的功能	341
11.1	看门狗定时器 1	341
11.1.1	功能	341
11.1.2	设置	343
11.1.3	寄存器	343
11.1.4	操作	345
11.2	看门狗定时器 2	347
11.2.1	功能	347
11.2.2	配置	348

11.2.3	寄存器.....	348
11.2.4	操作	350
第十二章	实时输出功能（RTO）	351
12.1	功能	351
12.2	配置	352
12.3	寄存器.....	353
12.4	操作	355
12.5	用法	356
12.6	注意事项	356
12.7	安全功能	357
第十三章	A/D 转换器.....	359
13.1	概述	359
13.2	功能	359
13.3	配置	360
13.4	寄存器.....	362
13.5	操作	370
13.5.1	基本操作.....	370
13.5.2	触发模式.....	371
13.5.3	操作模式.....	372
13.5.4	掉电检测功能.....	375
13.5.5	设置方法.....	376
13.6	注意事项	377
13.7	A/D 转换器特征表的阅读方法.....	383
第十四章	异步串行接口（UART）	387
14.1	特点	387
14.2	配置	388
14.3	寄存器.....	390
14.4	中断请求	396
14.5	操作	397
14.5.1	数据格式.....	397
14.5.2	发送操作.....	398
14.5.3	连续发送操作.....	400
14.5.4	接收操作.....	404
14.5.5	接收错误.....	405
14.5.6	校验位的类型和相应的操作.....	407
14.5.7	接收数据噪音过滤器	408
14.6	专门的波特率生成器 n（BRGn）	409
14.6.1	波特率生成器 n（BRGn）的配置	409
14.6.2	串行时钟的生成	410
14.6.3	波特率设置实例	413
14.6.4	接收过程中允许的波特率范围	414
14.6.5	连续发送过程中的发送速率.....	416
14.7	注意事项	416
第十五章	同步串行接口 0（CSIO）	417

15.1	特点	417
15.2	配置	418
15.3	寄存器	421
15.4	操作	430
15.4.1	发送/接收完成中断请求信号 (INTCSI0n)	430
15.4.2	单发送模式	432
15.4.3	连续发送模式	435
15.5	输出引脚	443
第十六章 IIC 总线		444
16.1	特点功能	444
16.2	配置	447
16.3	寄存器	449
16.4	功能	463
16.4.1	引脚配置	463
16.5	IIC 总线的定义和控制方法	464
16.5.1	开始条件	464
16.5.2	地址	465
16.5.3	发送方向指定	466
16.5.4	ACK	467
16.5.5	停止条件	468
16.5.6	等待状态	469
16.5.7	等待状态取消方法	471
16.6	I ² C 中断请求信号 (INTIIC0)	472
16.6.1	主设备运行	473
16.6.2	从设备运行 (当接收从设备地址数据 (地址匹配) 时)	476
16.6.3	从设备运行 (当接收扩展代码时)	480
16.6.4	无通讯运行	484
16.6.5	仲裁失败运行 (在仲裁失败后作为从设备运行)	485
16.6.6	当出现仲裁失败时的运行 (在仲裁失败后无通讯)	487
16.7	中断请求信号 (INTIIC0) 生成配时和等待控制	494
16.8	地址匹配检测方法	495
16.9	错误检测	495
16.10	扩展代码	496
16.11	仲裁	497
16.12	唤醒功能	498
16.13	通讯预定	499
16.13.1	当启用通讯预定功能时 (IICF0.IICRSV0 位 = 0)	499
16.13.2	当通讯预定功能被禁用时 (IICF0.IICRSV0 位 = 1)	502
16.14	注意事项	503
16.15	通讯操作	504
16.15.1	单主系统中的主设备运行	505
16.15.2	多主系统中的主设备运行	506
16.15.3	从设备运行	509
16.16	数据通讯的配时	512
第十七章 中断/异常处理功能		519
17.1	概述	519
17.1.1	特点	519

17.2	非屏蔽中断	522
17.2.1	操作	525
17.2.2	还原	526
17.2.3	NP 标志	527
17.3	可屏蔽中断	528
17.3.1	操作	528
17.3.2	恢复	530
17.3.3	可屏蔽中断的优先级	531
17.3.4	中断控制寄存器 (xxICn)	535
17.3.5	中断屏蔽寄存器 0, 1, 3 (IMR0, IMR1, IMR3)	537
17.3.6	当前服务的优先级寄存器 (ISPR)	538
17.3.7	ID 标志.....	539
17.3.8	看门狗定时器模式寄存器 1 (WDTM1)	540
17.4	外部中断请求输入引脚 (NMI 和 INTPO 到 INTP7)	541
17.4.1	噪声消除.....	541
17.4.2	边沿检测.....	543
17.5	软件异常	547
17.5.1	操作	547
17.5.2	恢复	548
17.5.3	EP 标志	549
17.6	异常陷阱	550
17.6.1	非法的操作代码定义	550
17.6.2	调试陷阱	552
17.7	多中断服务控制	554
17.8	中断响应时间	556
17.9	CPU 不对中断进行确认的时期	557
17.10	注意事项	557
第十八章	按键中断功能	558
18.1	功能	558
18.2	寄存器	559
第十九章	待机功能	560
19.1	概述	560
19.2	寄存器	563
19.3	HALT 模式	566
19.3.1	设置和操作状态	566
19.3.2	释放 HALT 模式	566
19.4	IDLE 模式	568
19.4.1	设置和操作状态	568
19.4.2	释放 IDLE 模式	569
19.5	STOP 模式	571
19.5.1	设置和操作状态	571
19.5.2	释放 STOP 模式.....	572
19.5.3	当释放 STOP 模式时保证振荡稳定时间.....	574
19.6	副时钟操作模式	575
19.6.1	设置和操作状态	575
19.6.2	释放副时钟操作模式	575
19.7	副 IDLE 模式	577

19.7.1	设置和操作状态	577
19.7.2	释放副 IDLE 模式	578
第二十章	复位功能	580
20.1	概述	580
20.2	配置	580
20.3	操作	581
第二十一章	Flash 存储器	585
21.1	特点	585
21.2	内存配置	586
21.3	功能概述	587
21.4	通过专门的 Flash 存储器编程器重写	591
21.4.1	编程环境	591
21.4.2	通讯模式	592
21.4.3	Flash 存储器控制	597
21.4.4	通讯模式的选择	598
21.4.5	通讯命令	599
21.4.6	引脚连接	600
21.5	通过自编程重写	605
21.5.1	概述	605
21.5.2	特点	606
21.5.3	标准自编程流程	607
21.5.4	Flash 存储器功能	608
21.5.5	引脚处理	608
21.5.6	使用的内部资源	609
第二十二章	片上调试功能	610
22.1	不使用 DCU 调试	611
22.1.1	电路连接示例	611
22.1.2	可屏蔽功能	612
22.1.3	保护用户资源	613
22.1.4	注意事项	618
22.2	ROM 安全功能	619
22.2.1	安全 ID	619
22.2.2	设置	620

	第二十三章 电气特性.....	622
	第二十四章 封装图	644
<R>	第二十五章 推荐焊接条件	645
<R>	附录 A 开发工具.....	646
	A.1 软件包.....	648
	A.2 语言处理软件.....	648
	A.3 控制软件	648
	A.4 调试工具 (硬件).....	649
	A.4.1 当使用 IECUBE® QB-V850ESKX1H 时	649
	A.4.2 当使用 MINICUBE QB-V850MINI 时	651
	A.4.3 当使用 MINICUBE2 QB-MINI2 时.....	653
	A.5 D 调试工具 (软件).....	654
	A.6 嵌入式软件.....	655
	A.7 Flash 存储器写入工具.....	655
	附录 B 指令集列表.....	656
	B.1 常规指令	656
	B.2 指令集(按字母顺序)	659
	附录 C 寄存器索引.....	666
<R>	附录 D 注意事项列表.....	672
<R>	附录 E 修订履历.....	699
	E.1 在此版本中的主要修订	699

第一章 概述

1.1 V850ES/Kx2 产品列表

产品名称		V850ES/KE2	V850ES/KF2	V850ES/KG2	V850ES/KJ2
引脚数目		64 引脚	80 引脚	100 引脚	144 引脚
存储器 (KB)	Flash 存储器	128	128 256	128 256	128 256
	RAM	4	6 12	6 16	6 16
供电电压		2.7 ~ 5.5 V			
最短执行指令时间		50 ns @20 MHz			
主时钟	X1 输入	2 ~ 10 MHz			
	副时钟	32.768 kHz			
端口	CMOS 输入	8	8	8	16
	CMOS I/O	41 (4) ^注	57 (6) ^注	72 (8) ^注	106 (12) ^注
	N 沟道漏极开路 I/O	2	2	4	6
定时器	16 位 (TMP)	1 通道	1 通道	1 通道	1 通道
	16 位 (TM0)	1 通道	2 通道	4 通道	6 通道
	8 位 (TM5)	2 通道	2 通道	2 通道	2 通道
	8 位 (TMH)	2 通道	2 通道	2 通道	2 通道
	间隔定时器	1 通道	1 通道	1 通道	1 通道
	钟表	1 通道	1 通道	1 通道	1 通道
	WDT1	1 通道	1 通道	1 通道	1 通道
	WDT2	1 通道	1 通道	1 通道	1 通道
RTO		6 位×1 通道	6 位×1 通道	6 位×1 通道	6 位×2 通道
串行接口	CSI	2 通道	2 通道	2 通道	3 通道
	自动传输/ 接收 3 线 CSI	–	1 通道	2 通道	2 通道
	UART	2 通道	2 通道	3 通道	3 通道
	I ² C	1 通道	1 通道	1 通道	2 通道
外部总线	地址空间	–	128 KB	3 MB	15 MB
	地址总线	–	16 位	22 位	24 位
	模式	–	仅复用	复用/单独	
DMA 控制器		–	–	4 通道	4 通道
10 位 A/D 转换器		8 通道	8 通道	8 通道	16 通道
8 位 D/A 转换器		–	–	2 通道	2 通道
中断	外部	9	9	9	9
	内部	26	29	41	47
按键返回输入		8 通道	8 通道	8 通道	8 通道
复位	RESET 引脚	提供			
	WDT1	提供			
	WDT2	提供			
稳压器		无	提供		
待机功能		HALT/IDLE/STOP/副 IDLE 模式			
适用工作温度		TA = -40 ~ +85°C			

注 括号中的数据仅表明可选 N 沟道漏极开路输出的引脚个数。

1.2 性能

- 最短指令执行时间：50 ns（工作在主时钟（f_{xx}）= 20 MHz 下）
- 通用寄存器：32 位 × 32 寄存器
- CPU 特性：
 - 带符号乘法（16 × 16 → 32）：1 ~ 2 时钟周期
 - 对寄存器没有破坏的指令可以不断地同时执行。
 - 饱和操作（包含上溢和下溢的检测功能）
 - 32 位移位指令：1 时钟周期
 - 位操作指令
 - 区分长短格式的读取/存储指令
- 存储器空间：为程序和数据提供 64 MB 的线性地址空间
 - 内部存储器：
 - μPD70F3726（单电源 flash 存储器：128 KB/RAM：4 KB）
- 中断和异常：
 - 不可屏蔽中断：3 个
 - 可屏蔽中断：32 个
 - 软件异常：32 个
 - 异常陷阱：1 个
- I/O 线：共：51 个
- 按键中断功能
- 定时器功能
 - 16 位定时器/事件计数器 P：1 通道
 - 16 位定时器/事件计数器 0：1 通道
 - 8 位定时器/事件计数器 5：2 通道
 - 8 位定时器 H：2 通道
 - 8 位间隔定时器 BRG：1 通道
 - 时钟定时器/间隔定时器：1 通道
 - 看门狗定时器
 - 看门狗定时器 1（可作为稳定振荡定时器）：1 通道
 - 看门狗定时器 2：1 通道
- 串行端口
 - 异步串行接口（UART）：2 通道
 - 3 线串行 I/O（CSIO）：2 通道
 - I²C 总线接口（I²C）：1 通道
- A/D 转换器：10 位分辨率 × 8 通道
- 实时输出接口：6 位 × 1 通道
- 待机功能：HALT/IDLE/STOP 模式，副时钟/副 IDLE 模式
- 时钟发生器
 - 主时钟振荡（f_x）/副时钟振荡（f_{xT}）
 - 可选择 7 个级别的 CPU 时钟（f_{xx}，f_{xx}/2，f_{xx}/4，f_{xx}/8，f_{xx}/16，f_{xx}/32，f_{xT}）
 - 时钟直连模式/PLL 模式可选
- 复位
 - 通过 $\overline{\text{RESET}}$ 引脚复位
 - 通过看门狗定时器 1 的溢出复位（WDTRES1）
 - 通过看门狗定时器 2 的溢出复位（WDTRES2）
- 封装：64 引脚塑料 LQFP（密脚距）（10 × 10）

1.3 应用领域

- 家庭音响
- AV 设备
- 电脑附属设备（键盘等）
- 家用电器
 - 空调的外部装备
 - 微波炉，电饭锅
- 工业设备
 - 泵
 - 售货机
 - FA

1.4 订货信息

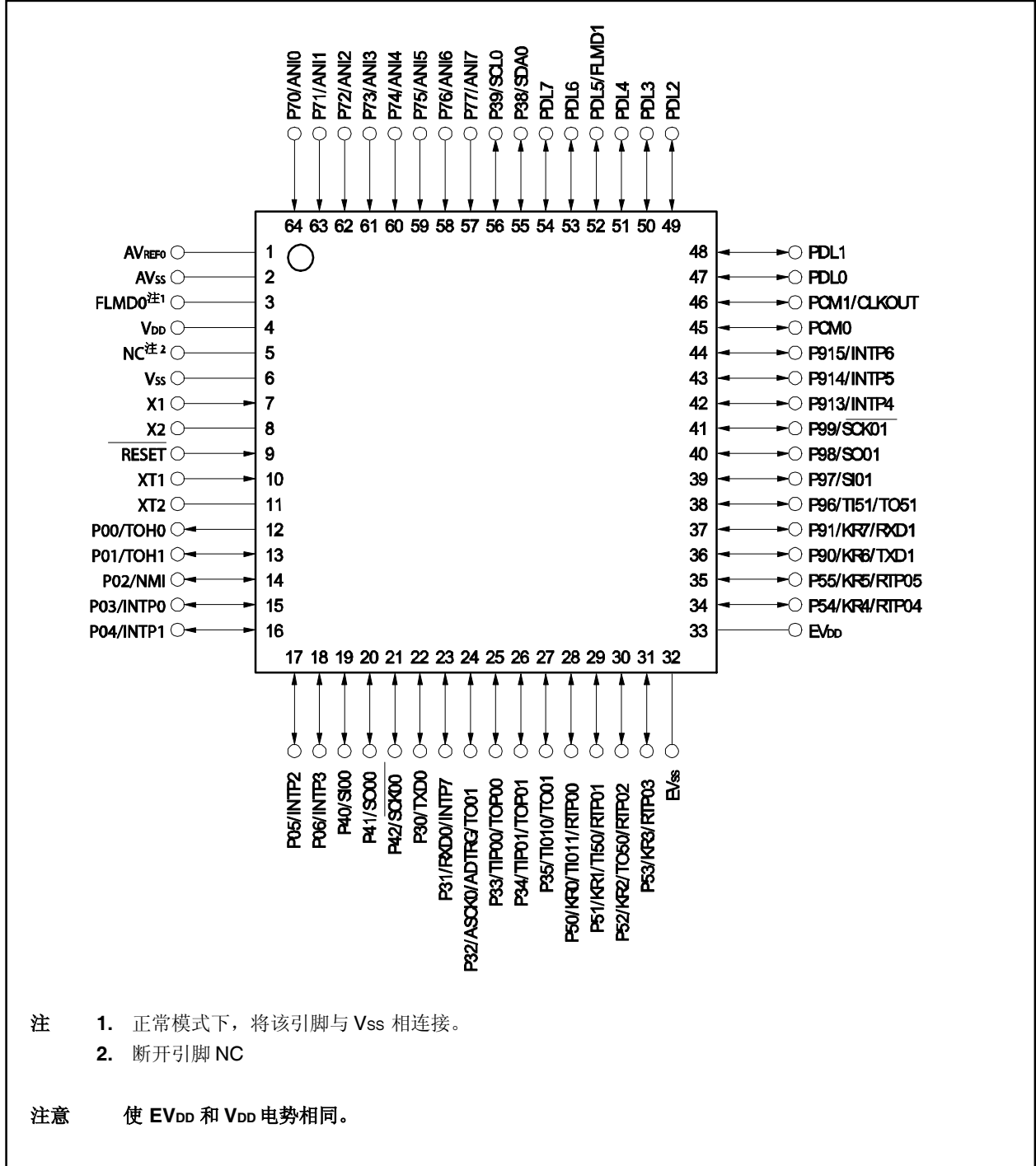
产品代号	封装
μ PD70F3726GB-8EU-A	64 引脚塑料 LQFP（密脚距）（10 × 10）

备注 代号末尾带 A 的产品属于无铅产品

1.5 引脚图（俯视图）

64 引脚塑料 LQFP（密脚距）（10 × 10）

μPD70F3726GB-8EU-A

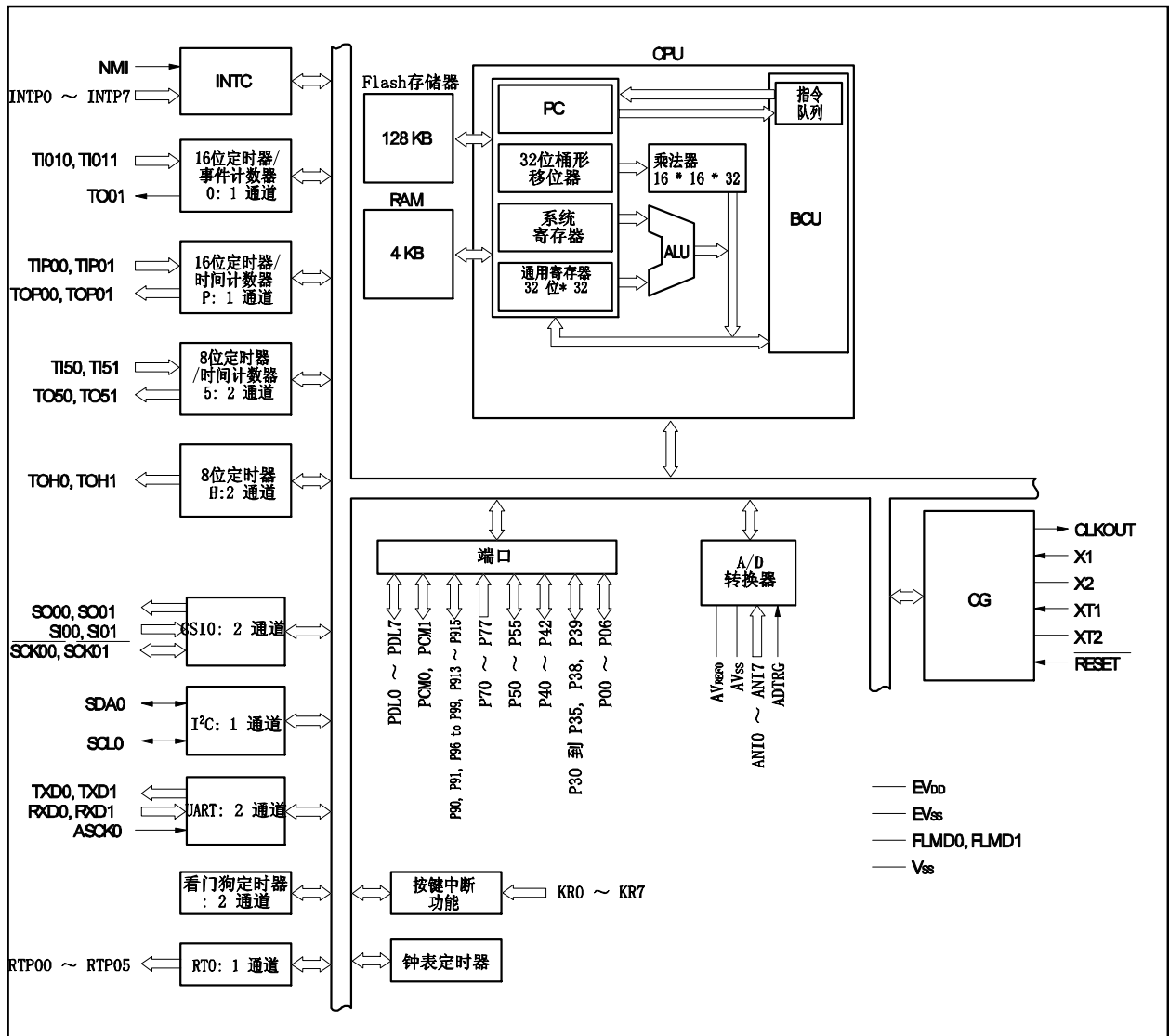


引脚名称

ADTRG:	A/D 触发信号输入	PDL0 ~ PDL7:	端口 DL
ANI0 ~ ANI7:	模拟信号输入	RESET:	复位
ASCK0:	异步串行时钟	RTP00 ~ RTP05:	实时输出端口
AVREF0:	模拟参考电压	RXD0, RXD1:	接收数据
AVSS:	模拟地	SCK00, SCK01:	串行时钟
CLKOUT:	时钟输出	SCL0:	串行时钟
EVDD:	端口电源	SDA0:	串行数据
EVSS:	端口地	SI00, SI01:	串行输入
FLMD0, FLMD1:	Flash 编程模式	SO00, SO01:	串行输出
INTP0 ~ INTP7:	外部中断输入	TI010, TI011,	
KR0 ~ KR7:	按键返回	TI50, TI51,	
NC:	不能连接	TIP00, TIP01:	定时器输入
NMI:	不可屏蔽中断请求	TO01,	
P00 ~ P06:	端口 0	TO50, TO51,	
P30 ~ P35, P38, P39:	端口 3	TOH0, TOH1,	
P40 ~ P42:	端口 4	TOP00, TOP01:	定时器输出
P50 ~ P55:	端口 5	TXD0, TXD1:	发送数据
P70 ~ P77:	端口 7	VDD:	电源
P90, P91, P96 ~ P99,		VSS:	地
P913 ~ P915:	端口 9	X1, X2:	主时钟晶振
PCM0, PCM1:	端口 CM	XT1, XT2:	副时钟晶振

1.6 功能模块配置

(1) 内部结构框图



(2) 内部单元

(a) CPU

CPU 通过使用 5 级流水线控制，实现了地址计算、算术逻辑运算、数据传输以及其他几乎所有指令的单时钟执行。

另外，乘法器（16 位 × 16 位 → 32 位）和桶型移位寄存器（32 位）等其它片上专用硬件的集成，也大大加快了复杂操作处理的速度。

(b) 总线控制单元 (BCU)

总线控制单元控制内部主线。

(c) ROM

V850ES/Kx2 产品提供 128KB 的 Flash 存储器，映射地址为 0000000H~001FFFFH。在取指令过程中，该存储器可由 CPU 在一个时钟周期内访问。

(d) RAM

V850ES/Kx2 产品提供 4KB 的 RAM，映射地址为 3FFE000H~3FFEFFFH 在数据访问过程中，该 RAM 可由 CPU 在一个时钟周期内访问。

(e) 中断控制器 (INTC)

该控制器用来处理由片上外围硬件和外部硬件发出的硬件中断请求（NMI，INTP0 ~ INTP7）。可以对这些中断请求指定 8 个级别的中断优先级，同时也可进行多重中断服务的处理。

(f) 时钟发生器 (CG)

V850ES/Kx2 产品提供一个主时钟振荡器和一个副时钟振荡器分别用于产生主时钟振荡频率（fx）和副时钟振荡频率（fx_T）。

共有两种模式可供选择：时钟直连模式，fx 被直接用作主时钟频率（f_{xx}）；PLL 模式，fx 在倍频 4 倍后被用作主时钟频率。

CPU 时钟频率（f_{cpu}）可以从 f_{xx}，f_{xx}/2，f_{xx}/4，f_{xx}/8，f_{xx}/16，f_{xx}/32 和 f_{xxT} 频率中进行选择。

(g) 定时器与计数器

片上集成了 1 个 O 通道 16 位定时器/计数器，1 个 P 通道 16 位定时器/事件计数器，2 个 5 通道 8 位定时器/事件计数器，可以测量脉冲间隔，频率以及可编程的脉冲输出。

2 个 5 通道 8 位定时器/事件计数器可以通过级联配置成 1 个 16 位定时器。

片上提供了 2 个 8 位定时器 H 通道，可以输出可编程脉冲。

(h) 时钟定时器

该定时器可以 32.768KHz 的副时钟或由预分频器 3 提供的 32.768KHz 的 f_{BRG} 为计数时钟提供用于时钟定时的参考时间间隔（0.5s）。也可以使时钟定时器工作于主时钟来实现间隔定时器功能。

(i) 看门狗定时器

V850ES//KE2 产品提供 2 个片上看门狗定时器，该定时器用于检测程序死锁，系统异常等错误状态。

看门狗定时器 1 可作为间隔定时器使用。当其作为看门狗定时器使用时，在发生溢出后，它将生成一个不可屏蔽的中断请求信号（INTWDT1）或者系统复位信号（WDTRES1）。当其作为间隔定时器使用时，在发生溢出后，它将生成一个可屏蔽的中断请求信号（INTWDTM1）。

看门狗定时器 2 在复位释放后执行默认操作。

在发生溢出后，它将生成一个不可屏蔽的中断请求信号（INTWDT2）或者系统复位信号（WDTRES2）。

(j) 串行接口（SIO）

V850ES//KE2 具有 3 种形式的串行接口：异步串行接口（UARTn），定时串行接口（CSI0n）和一个 I²C 总线接口（I²C0），可同时使用 5 个通道。

使用 UARTn 时，数据通过 TXDn ~ RXDn 引脚传输。

使用 CSI0n 时，数据通过 SO0n ~ SI0n 引脚，以及 SCK0n 引脚。

使用 I²C0 时，数据通过 SDA0 ~ SCL0 引脚传输。

备注 n = 0, 1

(k) A/D 转换器

该 A/D 转换器是一个具有 8 个模拟输入引脚的 10 位高速，高分辨率 A/D 转换器，采用逐次比较法进行转换。

(l) 按键中断功能

通过向按键输入引脚（8 通道）输入下降沿信号，可以产生一个按键中断请求信号（INTKR）。

(m) 实时输出功能

实时输出功能可以在定时器比较匹配信号产生时将输出锁存器中预先设置的 6 位数据输出。

片上集成了 1 通道 6 位数据实时输出功能。

(n) 端口

如下所示，可使用以下通用端口功能和控制引脚功能。

端口	I/O	复用功能
P0	7 位 I/O	NMI，外部中断，定时器输出
P3	8 位 I/O	串行接口，定时器 I/O，外部中断，A/D 转换器触发信号
P4	3 位 I/O	串行接口
P5	6 位 I/O	定时器 I/O、按键中断功能、实时输出功能
P7	8 位输入	A/D 转换器模拟输入
P9	9 位 I/O	串行接口、定时器 I/O、外部中断、按键中断功能
PCM	2 位 I/O	时钟输出
PDL	8 位 I/O	-

1.7 性能指标

产品代号		μ PD70F3726
内存	ROM	128 KB (单电源 flash 存储器)
	高速 RAM	4 KB
存储空间		64 MB
通用寄存器		32 位 \times 32 个寄存器
主时钟 (振荡频率)		陶瓷/晶体/外部时钟
		不用 PLL: 2 ~ 10 MHz (2.7 ~ 5.5 V) 使用 PLL: 2 ~ 5 MHz (4.5 ~ 5.5 V), 2 ~ 2.5 MHz (2.7 ~ 5.5 V)
副时钟 (振荡频率)		晶体/外部时钟 (32.768 kHz)
最短指令执行时间		50 ns (主时钟频率为 20MHz)
DSP 功能		32 \times 32 = 64: 200 ~ 250 ns (20 MHz) 32 \times 32 + 32 = 32: 300 ns (20 MHz) 16 \times 16 = 32: 50 ~ 100 ns (20 MHz) 16 \times 16 + 32 = 32: 150 ns (20 MHz)
I/O 端口		51 • 输入: 8 • I/O: 43 (N 沟道漏极开路输出选择: 4, 安装到 N 沟道漏极开路输出: 2)
定时器		16 位定时器/事件计数器 P: 1 通道 16 位定时器/事件计数器 0: 1 通道 8 位定时器/事件计数器 5: 2 通道 (16 位定时器/事件计数器 P: 1 通道不可用) 8 位定时器 H: 2 通道 看门狗定时器: 2 通道 时钟定时器: 1 通道 8 位间隔定时器: 1 通道
实时输出端口		4 位 \times 1, 2 位 \times 1, 或 6 位 \times 1
A/D 转换器		10 位分辨率 \times 8 通道
串行端口		CSI: 2 通道 UART: 2 通道 I ² C 总线: 1 通道 专用波特率发生器: 2 通道
中断源		外部: 9 (9) [*] , 内部: 26
节能功能		STOP/IDLE/HALT/副 IDLE 模式
工作电压		4.5 ~ 5.5 V (20 MHz) / 2.7 ~ 5.5 V (8 MHz)
封装		64 引脚 塑料 LQFP (密脚距) (10 \times 10 mm)

注 括号中的数据表示可以释放 STOP 模式的外部中断的个数。

第二章 引脚功能

V850ES/KE2 的引脚名称和功能如下所示，可分为端口引脚和非端口引脚。

引脚 I/O 缓冲式供电电源有两种：AVREF0 和 EVDD。这些电源与引脚的关系如下所示。

表 2-1. 引脚 I/O 缓冲式供电电源

供电电源	对应引脚
AVREF0	端口 7
EVDD	复位, 端口 0, 3 ~ 5, 9, CM, DL

2.1 引脚功能列表

(1) 端口引脚

(1/2)

引脚名称	引脚编号	I/O	上拉电阻	功能	复用功能
P00	12	I/O	有	端口 0 I/O 端口 可以位选输入/输出模式。	TOH0
P01	13				TOH1
P02	14				NMI
P03	15				INTP0
P04	16				INTP1
P05	17				INTP2
P06	18				INTP3
P30	22	I/O	有	端口 3 I/O 端口 可以位选输入/输出模式。 P38 和 P39 固定为 N 沟道开漏输出。	TXD0
P31	23				RXD0/INTP7
P32	24				ASCK0/ADTRG/TO01
P33	25				TIP00/TOP00
P34	26				TIP01/TOP01
P35	27				TI010/TO01
P38	55				无
P39	56		无	SCL0	
P40	19	I/O	有	端口 4 I/O 端口 可以位选输入/输出模式。P41 和 P42 可以位选 N 沟道漏极开路输出。	SI00
P41	20				SO00
P42	21				SCK00

(2/2)

引脚名称	引脚编号	I/O	上拉电阻	功能	复位功能
P50	28	I/O	有	端口 5 I/O 端口 可以位选输入/输出模式。	TI011/RTP00/KR0
P51	29				TI50/RTP01/KR1
P52	30				TO50/RTP02/KR2
P53	31				RTP03/KR3
P54	34				RTP04/KR4
P55	35				RTP05/KR5
P70	64	输入	无	端口 7 输入端口	ANI0
P71	63				ANI1
P72	62				ANI2
P73	61				ANI3
P74	60				ANI4
P75	59				ANI5
P76	58				ANI6
P77	57				ANI7
P90	36	I/O	有	端口 9 I/O 端口 可以位选输入/输出模式。 P98 和 P99 可以位选 N 沟道输出。	TXD1/KR6
P91	37				RXD1/KR7
P96	38				TI51/TO51
P97	39				SI01
P98	40				SO01
P99	41				SCK01
P913	42				INTP4
P914	43				INTP5
P915	44				INTP6
PCM0	45	I/O	有	端口 CM I/O 端口 可以位选输入/输出模式。	-
PCM1	46				CLKOUT
PDL0	47	I/O	有	端口 DL I/O 端口 可以位选输入输出模式。	-
PDL1	48				-
PDL2	49				-
PDL3	50				-
PDL4	51				-
PDL5	52				FLMD1
PDL6	53				-
PDL7	54				-

(2) 非端口引脚

(1/2)

引脚名称	引脚编号	I/O	上拉电阻	功能	复位功能
ADTRG	24	输入	有	A/D 转换器外部触发输入	P32/ASCK0/TO01
ANI0	64	输入	无	A/D 转换器模拟电压输入	P70
ANI1	63				P71
ANI2	62				P72
ANI3	61				P73
ANI4	60				P74
ANI5	59				P75
ANI6	58				P76
ANI7	57				P77
ASCK0	24	输入	有	UART0 串行时钟输入	P32/ADTRG/TO01
AV _{REF0}	1	-	-	A/D 转换器的参考电压/复位功能的正电源提供	-
AV _{SS}	2	-	-	A/D 转换器的地电平/复位功能端口	-
CLKOUT	46	输出	无	内部系统时钟输出	PCM1
EV _{DD}	33	-	-	外部存储的正电源供应	-
EV _{SS}	32	-	-	外部存储的地电平	-
FLMD0	3	输入	无	flash 存储器编程模式设置引脚	-
FLMD1	52		有		PDL5
INTP0	15	输入	有	外部中断请求输入 (可屏蔽, 模拟噪音消除电路)	P03
INTP1	16				P04
INTP2	17				P05
INTP3	18			外部中断请求输入 (可屏蔽, 数字+模拟噪声消除电路)	P06
INTP4	42			外部中断请求输入 (可屏蔽, 模拟噪声消除电路)	P913
INTP5	43				P914
INTP6	44				P915
INTP7	23				P31/RXD0
KR0	28	输入	有	按键返回输入	P50/TI011/RTP00
KR1	29				P51/TI50/RTP01
KR2	30				P52/TO50/RTP02
KR3	31				P53/RTP03
KR4	34				P54/RTP04
KR5	35				P55/RTP05
KR6	36				P90/TXD1
KR7	37				P91/RXD1
NC	5	-	-	非内部连接。保持开路。	-
NMI	14	输入	有	外部中断输入 (不可屏蔽, 模拟噪声消除电路)	P02
RESET	9	输入	-	系统复位输入	-

引脚名称	引脚编号	I/O	上拉电阻	功能	复位功能		
RTP00	28	输出	有	实时输出端口	P50/TI011/KR0		
RTP01	29				P51/TI50/KR1		
RTP02	30				P52/TO50/KR2		
RTP03	31				P53/KR3		
RTP04	34				P54/KR4		
RTP05	35				P55/KR5		
RXD0	23	输入	有	UART0 串行接收数据输入	P31/INTP7		
RXD1	37			UART1 串行接收数据输入	P91/KR7		
SCK00	21	I/O	有	CSI00 和 CSI01 的 I/O 串行时钟 可以位选 N 沟道开漏输出	P42		
SCK01	41				P99		
SCL0	56	I/O	无	I ² C0 的 I/O 串行时钟 I ² C0 的 I/O 连接到 N 沟道开漏输出	P39		
SDA0	55	I/O	无	I ² C0 的 I/O 串行发送/接收数据 I ² C0 的 I/O 连接到 N 沟道开漏输出	P38		
SI00	19	输入	有	CSI00 的串行接收数据输入	P40		
SI01	39			CSI01 的串行接收数据输入	P97		
SO00	20	输出	有	串行发送数据输出(CSI00 和 CSI01)可位选 N 沟道开漏输出。	P41		
SO01	40				P98		
TI010	27	输入	有	TM01 的捕捉触发输入/外部事件输入	P35/TO01		
TI011	28			TM01 的捕捉触发输入	P50/RTP00/KR0		
TI50	29			TM50 的外部事件输入	P51/RTP01/KR1		
TI51	38			TM51 的外部事件输入	P96/TO51		
TIP00	25			TMP0 的捕捉触发输入/外部事件输入	P33/TOP00		
TIP01	26			TMP0 的捕捉触发输入	P34/TOP01		
TO01	24			输出	有	TM01 的定时器输出	P32/ASCK0/ADTRG
	27		P35/TI010				
TO50	30	TM50 的定时器输出	P52/RTP02/KR2				
TO51	38	TM51 的定时器输出	P96/TI51				
TOH0	12	TMH0 的定时器输出	P00				
TOH1	13	TMH1 的定时器输出	P01				
TOP00	25	TMP0 的定时器输出	P33/TIP00				
TOP01	26		P34/TIP01				
TXD0	22	输出	有			UART0 的串行发送数据输出	P30
TXD1	36					UART1 的串行发送数据输出	P90/KR6
V _{DD}	4	-	-	内部供电电源引脚	-		
V _{SS}	6	-	-	内部地电平	-		
X1	7	输入	无	连接主时钟振荡器	-		
X2	8	-	无		-		
XT1	10	输入	无	连接副时钟振荡器	-		
XT2	11	-	无		-		

2.2 I/O 引脚电路和不使用引脚的推荐连接法

(1/2)

引脚	复用功能	引脚编号	I/O 电路类型	推荐连接	
P00	TOH0	12	5-A	输入：通过一个电阻单独连接到 EV _{DD} 或者 EV _{SS} 。 输出：保持开路。	
P01	TOH1	13			
P02	NMI	14	5-W		
P03 ~ P06	INTP0 ~ INTP3	15 ~ 18			
P30	TXD0	22	5-A		
P31	RXD0/INTP7	23	5-W		
P32	ASCK0/ADTRG	24			
P33	TIP00/TOP00	25			
P34	TIP01/TOP01	26			
P35	TI010/TO01	27			
P38	SDA0	55	13-AD		
P39	SCL0	56			
P40	SI00	19	5-W		
P41	SO00	20	10-E		
P42	$\overline{\text{SCK00}}$	21	10-F		
P50	TI011/RTP00/KR0	28	8-A		
P51	TI50/RTP01/KR1	29			
P52	TO50/RTP02/KR2	30			
P53	RTP03/KR3	31			
P54	RTP04/KR4	34			
P55	RTP05/KR5	35			
P70 ~ P77	ANI0 ~ ANI7	64 ~ 57	9-C	连接到 AV _{REF0} 或者 AV _{SS}	
P90	TXD1/KR6	36	8-A	输入：通过一个电阻单独连接到 EV _{DD} 或者 EV _{SS} 。 输出：保持开路。	
P91	RXD1/KR7	37			
P96	TI51/TO51	38			
P97	SI01	39	5-W		
P98	SO01	40	10-E		
P99	$\overline{\text{SCK01}}$	41	10-F		
P913 ~ P915	INTP4 ~ INTP6	42 ~ 44	5-W		
PCM0	-	45	5-A		
PCM1	CLKOUT	46			
PDL0 ~ PDL4	-	47 ~ 51			
PDL5	FLMD1	52			
PDL6, PDL7	-	53, 54			
AV _{REF0}	-	1	-	直接连接到 V _{DD} 。	
AV _{SS}	-	2	-	-	
EV _{DD}	-	33	-	-	
EV _{SS}	-	32	-	-	

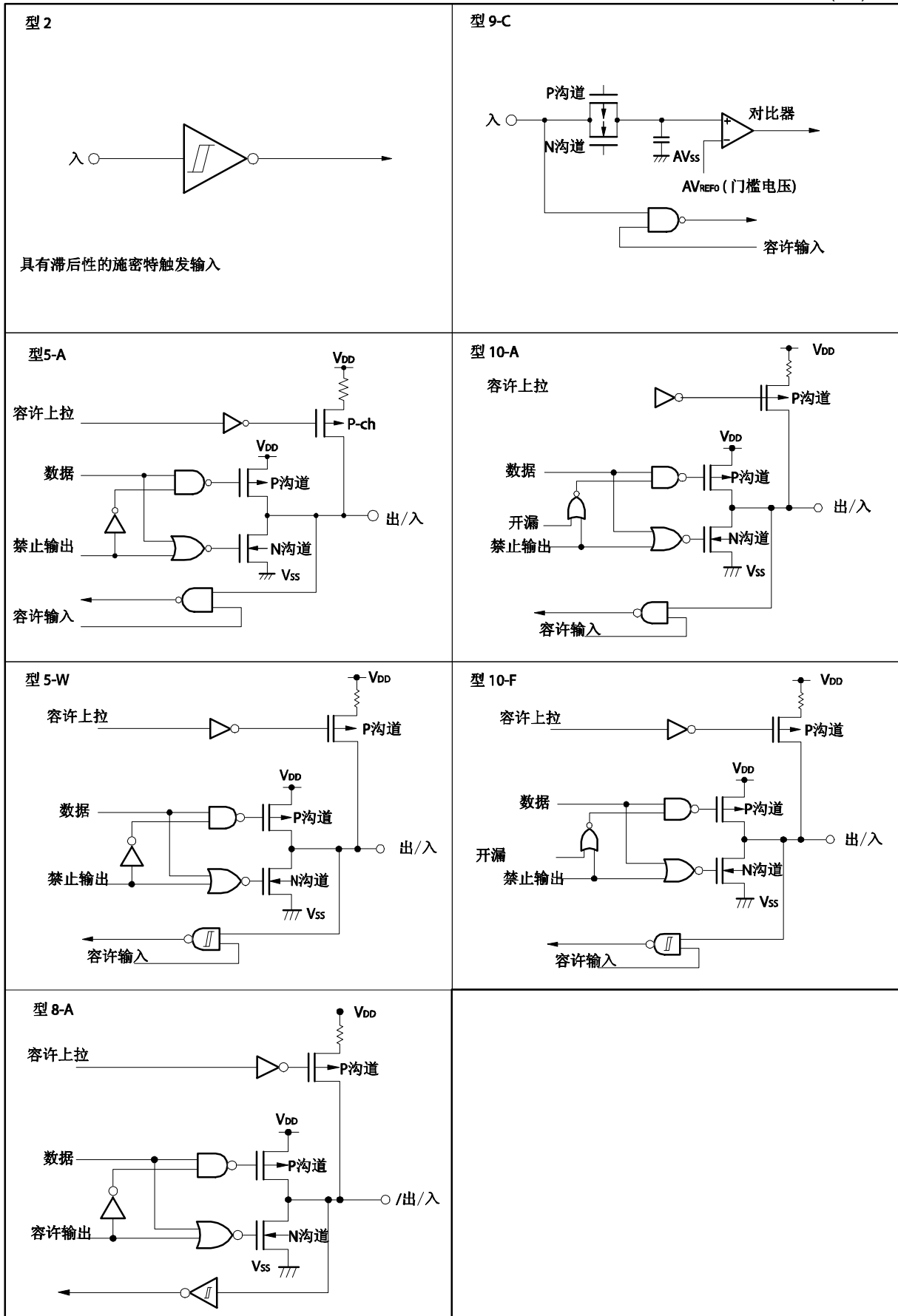
(2/2)

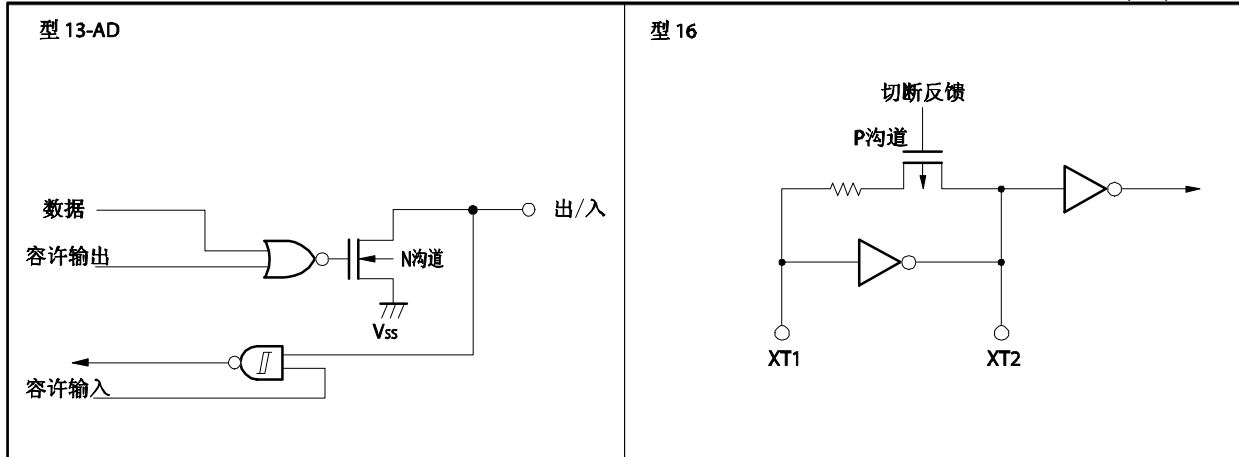
引脚	复用功能	引脚编号	I/O 电路类型	推荐连接
NC	–	5	–	保持开路。
RESET	–	9	2	–
FLMD0	–	3	–	直接与 EV _{SS} 或 V _{SS} 相连或用一个 10 kΩ 下拉电阻。
V _{DD}	–	4	–	–
V _{SS}	–	6	–	–
X1	–	7	–	–
X2	–	8	–	–
XT1	–	10	16	直接连接 V _{SS} [#] 。
XT2	–	11	16	保持开路。

注 当引脚不用时，务必将 PSMR.XTSTP 设置为 1 位。

2.3 引脚 I/O 电路

(1/2)





备注 视 V_{DD} 为 EV_{DD} ，视 V_{SS} 为 EV_{SS} 。

第三章 CPU 功能

V850ES/KE2 系列产品的 CPU 是基于 RISC 架构设计的，具备 5 级流水线控制，几乎所有指令均可在一个时钟周期内完成。

3.1 特点

- 指令数目: 83
- 最小指令执行时间: 50.0 ns (@ 20 MHz 操作: 4.5 ~ 5.5 V)
100 ns (@ 10 MHz 操作: 2.7 ~ 5.5 V)
- 存储器空间 程序 (物理地址) 空间: 64 MB 线性
 数据 (逻辑地址) 空间: 4 GB 线性
- 通用寄存器: 32 位 × 32 个
- 内部 32 位架构
- 5 级流水线控制
- 乘除法指令
- 饱和运算指令
- 32 位移位指令: 1 时钟
- 长/短格式的载入/存储指令
- 4 种位操作指令
 - SET1
 - CLR1
 - NOT1
 - TST1

3.2 CPU 寄存器组

V850ES/KE2 产品的 CPU 寄存器组可分为两组：通用程序寄存器组和专用系统寄存器组。所有的寄存器组的宽度均为 32 位。

关于更多详细信息，敬请参阅 **V850ES 架构用户手册**。

(1)程序寄存器组		(2)系统寄存器组	
31		31	
0		0	
r0	(零寄存器)	EIPC	(中断状态保存寄存器)
r1	(汇编器保存寄存器)	EIPSW	(中断状态保存寄存器)
r2			
r3	(堆栈指针 (SP))	FEPC	(NMI 状态保存寄存器)
r4	(全局指针 (GP))	FEPSW	(NMI 状态保存寄存器)
r5	(文本指针 (TP))		
r6		ECR	(中断源寄存器)
r7			
r8		PSW	(程序状态词)
r9			
r10		CTPC	(CALLT 执行状态保存寄存器)
r11		CTPSW	(CALLT 执行状态保存寄存器)
r12			
r13			
r14		DBPC	(异常陷阱状态保存寄存器)
r15		DBPSW	(异常陷阱状态保存寄存器)
r16			
r17		CTBP	(CALLT 基指针)
r18			
r19			
r20			
r21			
r22			
r23			
r24			
r25			
r26			
r27			
r28			
r29			
r30	(元素指针 (EP))		
r31	(连接指针 (LP))		
31			
0			
PC	(程序计数器)		

3.2.1 程序寄存器组

程序寄存器组包含通用寄存器和程序计数器。

(1) 通用寄存器(r0 ~ r31)

共有 32 个通用寄存器(r0~r31)可供选择。这些寄存器均可用于存放数据变量或地址变量。

但是，r0 和 r30 是被指令隐含使用的，因此要慎用这两个寄存器。寄存器 r0 始终保持 0 值，用于使用了 0 值的操作和偏移量为 0 的寻址操作。寄存器 r30 被指令 SLD 和 SST 所使用，在访问存储器时被用作基指针。

r1, r3~r5 以及 r31 被汇编器和 C 编译器隐含使用。因此，在使用这些寄存器之前要对它们的内容进行保存，以免发生数据丢失。使用之后，要对寄存器的值进行恢复。寄存器 r2 在一些情况下会被实时操作系统所使用。当它没有被实时操作系统使用时可作为变量寄存器。

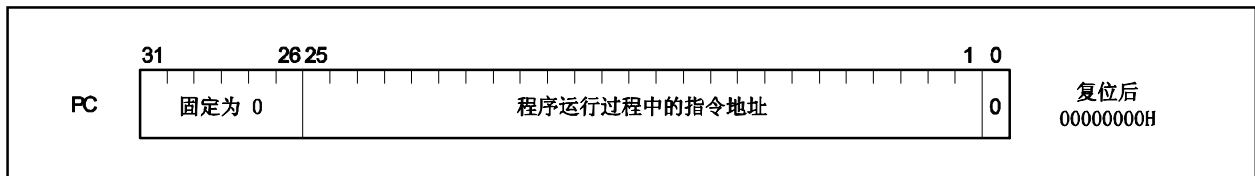
表 3-1. 程序寄存器

名称	用途	操作
r0	零寄存器	始终保持为零
r1	汇编器保留寄存器	用于生成 32 位立即数据的工作寄存器
r2	地址/数据变量寄存器(当 r2 不被实时操作系统所使用时)	
r3	堆栈指针	当函数被调用时用于生成堆栈架构
r4	全局指针	用于访问数据区域内的全局变量
r5	文本指针	用于表示文本区域(放置程序程序代码的区域)的开始部分的寄存器
r6 ~ r29	地址/数据变量寄存器	
r30	元素指针	访问存储器时用于生成地址的基指针
r31	连接指针	用于编译器调用函数时
PC	程序计数器	程序执行时存放指令地址

(2) 程序计数器(PC)

该寄存器在程序执行时用于存放指令地址。它的低 26 位有效，第 26~31 位恒为 0。若第 25 位~第 26 位产生进位，那么这个进位将被忽略。

第 0 位固定为 0，因此，向奇地址的跳转将不会被执行。



3.2.2 系统寄存器组

系统寄存器控制 CPU 的状态并且保存中断信息。

使用系统寄存器载入/存储(load/store)指令(LDSR 或 STSR)并指定一个系统寄存器编号(如下所示)可以对这些系统寄存器进行读/写操作。

表 3-2. 系统寄存器编号

系统寄存器编号	系统寄存器名	可否指定操作数	
		LDSR 指令	STSR 指令
0	中断状态保存寄存器 (EIPC) ^{注1}	是	是
1	中断状态保存寄存器 (EIPSW) ^{注1}	是	是
2	NMI 状态保存寄存器 (FEPC) ^{注1}	是	是
3	NMI 状态保存寄存器 (FEPSW) ^{注1}	是	是
4	中断源寄存器 (ECR)	否	是
5	程序状态字 (PSW)	是	是
6 ~ 15	预留(访问这些寄存器号的操作结果不能保证。)	否	否
16	CALLT 指令状态保存寄存器 (CTPC)	是	是
17	CALLT 指令状态保存寄存器 (CTPSW)	是	是
18	异常/调试陷阱状态保存寄存器 (DBPC)	是 ^{注2}	是 ^{注2}
19	异常/调试陷阱状态保存寄存器 (DBPSW)	是 ^{注2}	是 ^{注2}
20	CALLT 基指针 (CTBP)	是	是
21 ~ 31	预留(访问这些寄存器号的操作结果不能保证。)	否	否

- 注
1. 由于系统寄存器只有一组，因此，要实现多重中断需用程序对这组寄存器进行保存。
 2. 这些寄存器只能在 DBTRAP 指令或非法操作代码和 DBRET 指令之间被访问。

注意事项 即使 EIPC, FEPC 或 CTPC 的第 0 位被 LDSR 指令置 1，在中断处理程序执行后程序由 RETI 指令返回时，第 0 位仍然被忽略(这是因为 PC 的第 0 位固定为 0)。因此请使用偶数数值(第 0 位为 0)来设置 EIPC, FEPC 或 CTPC。

(1) 中断状态保存寄存器 (EIPC 和 EIPSW)

共有两个中断状态保存寄存器，EIPC 和 EIPSW。

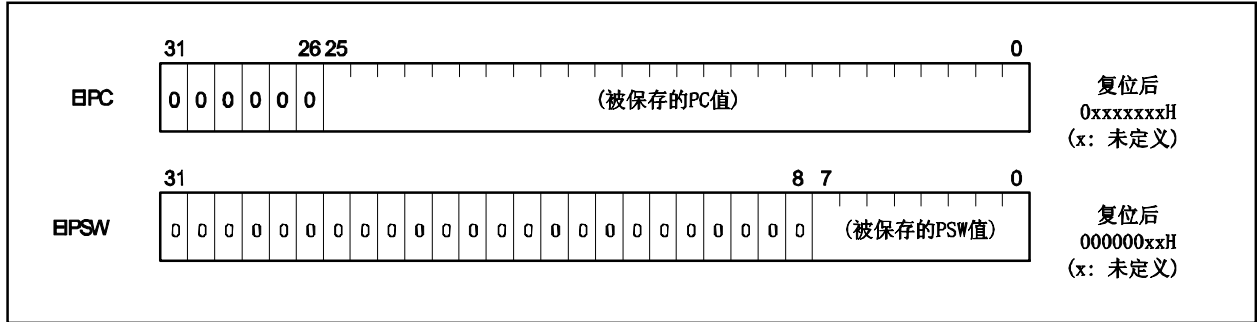
当软件异常或可屏蔽中断发生时，程序计数器 PC 的值被保存在 EIPC 中，程序状态字 PSW 的值被保存在 EIPSW 中(当不可屏蔽中断(NMI)发生时，PC 和 PSW 的值则会被分别保存在不可屏蔽中断状态保存寄存器 FEPC 和 FEPSW 中)。软件异常和可屏蔽中断发生时，当前指令的下一条指令的地址会被保存到 EIPC 中（一些指令除外，关于更多细节，敬请参阅 17.9 CPU 不对中断进行确认的时期）。

当前的 PSW 值被保存在 EIPSW 中。

由于中断状态保存寄存器只能对一组中断状态进行保存，因此当允许多重中断服务时，系统环境需要由程序自行保存。

EIPC 的第 26~31 位和 EIPSW 的第 8~31 位为扩展功能保留(这些位始终为 0)。

当 RETI 指令被执行时，EIPC 和 EIPSW 的值将被分别恢复到 PC 和 PSW。



(2) 不可屏蔽中断状态保存寄存器 (FEPC 和 FEPSW)

共有两个 NMI 状态保存寄存器，FEPC 和 FEPSW。

当不可屏蔽中断(NMI)发生时，程序寄存器 PC 的值被保存在 FEPC 中，程序状态字 PSW 的值被保存在 FEPSW 中。

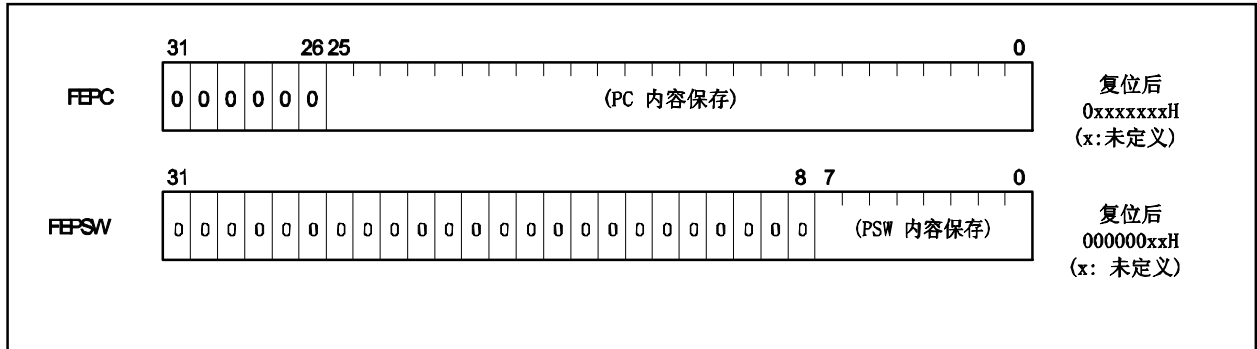
不可屏蔽中断发生时，当前指令的下一条指令（除了某些指令之外）的地址会被保存到 FEPC 中。

当前的 PSW 值被保存在 FEPSW 中。

由于只有一组 NMI 状态保存寄存器，因此当多重中断发生时，系统环境需要由程序自行保存。

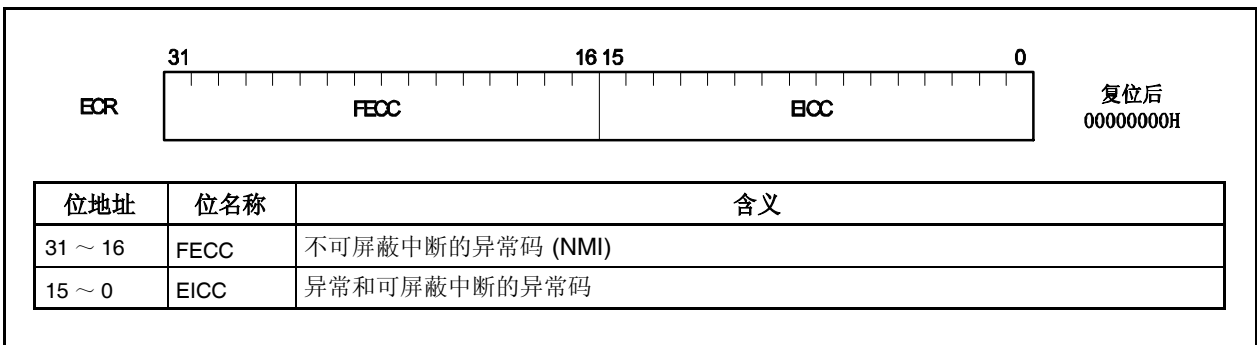
FEPC 的第 26~31 位和 FEPSW 的第 8~31 位为扩展功能保留(这些位始终为 0)。

当 RETI 指令被执行时，FEPC 和 FEPSW 的值将被分别恢复到 PC 和 PSW。



(3) 中断源寄存器 (ECR)

中断源寄存器(ECR)用来在异常或中断发生时保持异常或中断源。它保存每一个中断源的异常码。因为这个寄存器是一个只读寄存器，所以使用 LDSR 指令不能将数据写入该寄存器。



(4) 程序状态字 (PSW)

程序状态字(PSW)是由代表程序(指令执行结果)和 CPU 状态的一组标志位组成。

如果使用 LDSR 指令更改该寄存器的内存,那么新的状态将在 LDSR 指令执行后立即生效。当用 LDSR 指令对 PSW 进行写操作时,中断请求响应始终保持无响应状态。寄存器的第 8~31 位为扩展功能保留(始终为 0)。

(1/2)

31
8 7 6 5 4 3 2 1 0

PSW

RFU

NP	EP	ID	SAT	CY	OV	S	Z
----	----	----	-----	----	----	---	---

复位后
00000020H

位地址	标志名称	含义
31 ~ 8	RFU	保留区域。固定为零。
7	NP	表示正在进行非可屏蔽中断(NMI)服务。响应 NMI 后该标志被设置,同时禁止多重中断的响应。 0: 未执行 NMI 服务。 1: 正在执行 NMI 服务。
6	EP	表示正在进行异常处理。异常产生后该标志被设置。该标志被设置后仍可以对中断请求进行响应。 0: 未执行异常处理。 1: 正在执行异常处理。
5	ID	显示可屏蔽中断能否被响应。 0: 允许中断。 1: 禁止中断。
4	SAT ^注	显示由于产生了溢出,饱和运算处理指令的结果是饱和的。因为该标志是累加性标志,所以运算结果为饱和的。饱和运算指令将把该位置 1,而即使下一条指令的运算结果为不饱和,该位也不会被清零。向 PSW 中装入数据可将该位清零。注意在普通算术运算过程中,该位既不会被置 1,也不会被清零。 0: 不饱和。 1: 饱和。
3	CY	运算产生进位或借位时,该标志会被置 1。 0: 未产生进位或借位。 1: 产生进位或借位。
2	OV ^注	运算过程中产生溢出时,该标志被置 1。 0: 未产生溢出。 1: 产生溢出。
1	S ^注	运算结果为负值时,该标志被置 1。 0: 运算结果为正值或零。 1: 运算结果为负值。
0	Z	运算结果为零时,该标志被设置。 0: 运算结果不为 0。 1: 运算结果为 0。

备注 请阅读下页中的注。

注 饱和运算的操作结果是由 OV 和 S 标志的内容决定的。在饱和运算过程中只有 OV 标志被置 1，SAT 标志才会被置位(1)。

运算结果的状态	标志状态			饱和和处理运算结果
	SAT	OV	S	
超过正值上界	1	1	0	7FFFFFFFH
超过负值下界	1	1	1	80000000H
正值 (未超上界)	保持原值不变	0	0	运算结果本身
负值 (未超下界)			1	

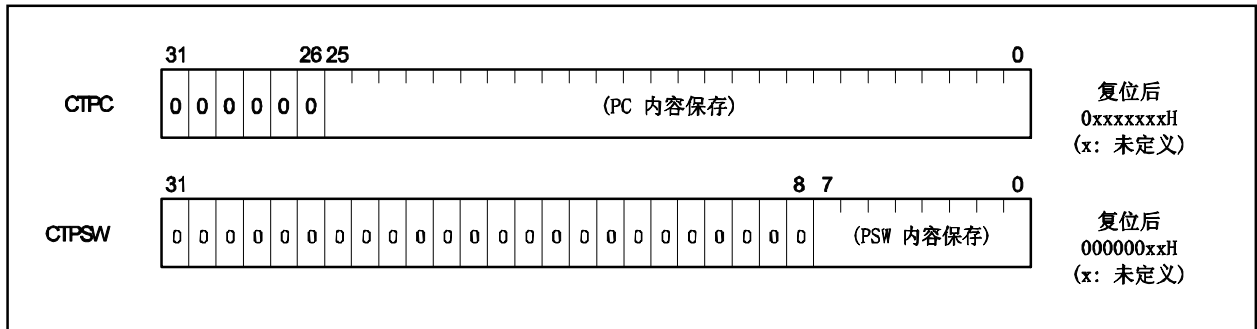
(5) CALLT 执行状态保存寄存器 (CTPC 和 CTPSW)

共有两个 CALLT 执行状态保存寄存器，CTPC 和 CTPSW。

当 CALLT 指令执行时，程序计数器 PC 值被保存在 CTPC 中，程序状态字 PSW 值被保存在 CTPSW 中。被保存在 CTPC 中的值是 CALLT 下一条指令的地址。

保存在 CTPSW 中的值是当前的 PSW 值。

CTPC 的第 26~31 位和 CTPSW 的第 8~31 位为扩展功能保留(始终为 0)。



(6) 异常/调试陷阱状态保存寄存器 (DBPC 和 DBPSW)

共有两个异常/调试陷阱状态保存寄存器，DBPC 和 DBPSW。

当异常陷阱或调试陷阱发生时，PC 值被保存在 DBPC 中，PSW 值被保存在 DBPSW 中。

被保存在 DBPC 中的值是异常陷阱或调试陷阱发生时正在执行的指令的下一条指令的地址。

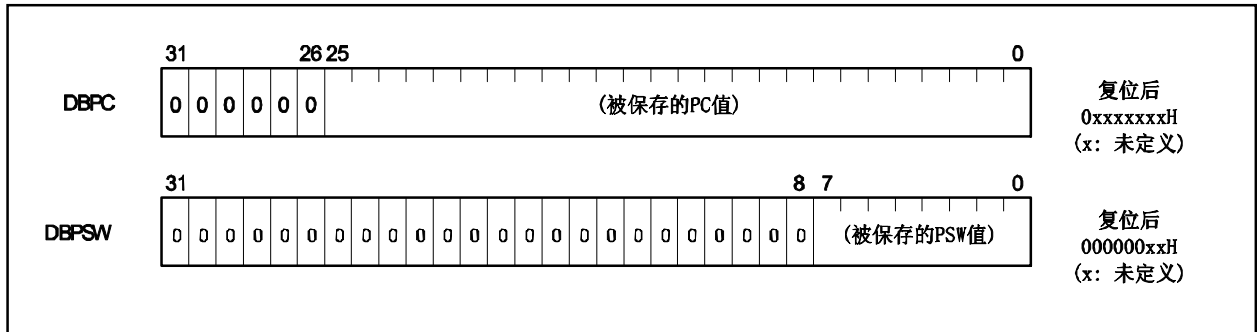
保存在 DBPSW 中的值是当前的 PSW 值。

<R>

该寄存器只有在执行 DBTRAP 指令或非法操作代码和 DBRET 指令之间才能够被读写。

DBPC 的第 26~31 位和 DBPSW 的第 8~31 位为扩展功能保留(始终为 0)。

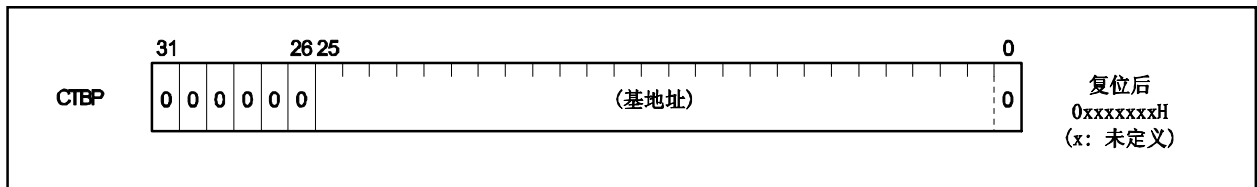
当 DBRETI 指令被执行时，DBPC 和 DBPSW 中的值将被分别恢复到 PC 和 PSW 中。



(7) CALLT 基址针 (CTBP)

CALLT 基址指针(CTBP)用于指定一个表地址或生成一个目标地址(第 0 位始终为 0)。

第 26~31 位为扩展功能保留(始终为 0)。



3.3 操作模式

V850ES/ KE2 具有以下几种操作模式。

(1) 正常操作模式

在这个模式下，所有与总线接口相关的引脚都会在系统复位后被设为端口模式。程序转至内部 ROM 的复位入口地址，然后指令操作开始执行。

(2) flash 存储器编程模式

在这个模式下，内部 flash 存储器可以用 flash 存储器编写器进行编写。

(a) 指定操作模式

根据引脚 FLMD0 和 FLMD1 的状态（输入电平）可对器件进行操作模式的指定。

正常模式下，要确保 FLMD0/1 引脚在系统复位期间输入低电平。

flash 存储器编程模式下，如果连接编程器，编程器应该输入 FLMD0 脚高电平信号，但是在自编程模式下这个信号必须通过一个外部电路输入。

在应用系统中设定这些引脚的参数，在操作中不要改变引脚的设置。

FLMD0	FLMD1	操作模式
L	x	正常操作模式
H	L	flash 存储器编程模式
H	H	禁止设置

备注 H: 高电平

L: 低电平

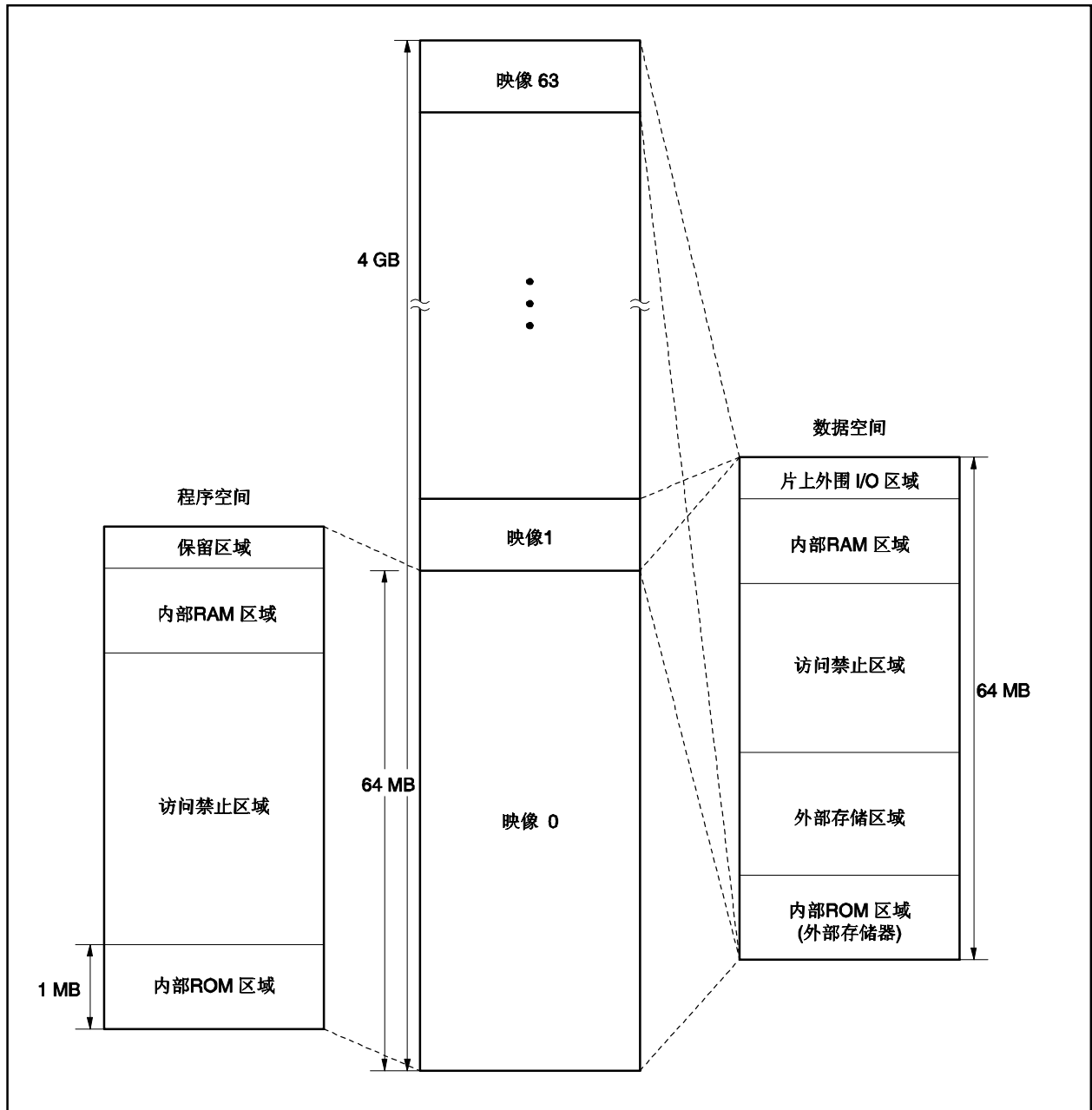
x: 忽略

3.4 地址空间

3.4.1 CPU 地址空间

对于指令寻址，支持 1 MB 的内部 ROM 区域和 64 MB 线性地址空间（程序空间）的内部 RAM 区域。支持对操作数的最大 4GB 的线形地址空间。这个 4GB 的线性地址空间可视为 64 个 64MB 的物理地址空间。这意味着无论地址位的第 26~31 位为何值，系统总是在同样的 64MB 的物理地址空间中进行寻址。

图 3-1. 地址空间图



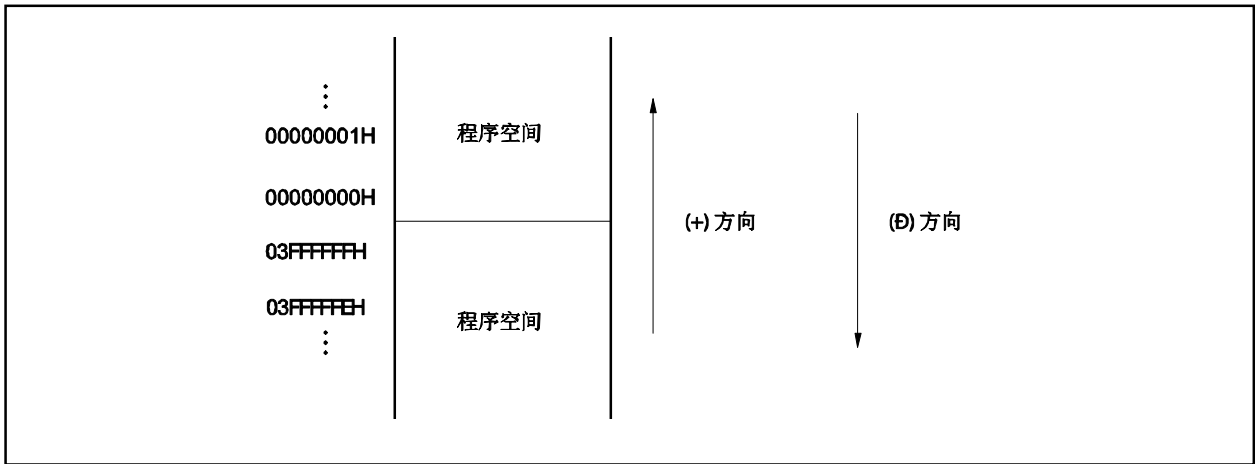
3.4.2 CPU 地址空间的回绕

(1) 程序空间

PC(程序计数器)的 32 位中, 高 6 位固定为 0, 只有低 26 位有效。即使分支地址计算过程中产生第 25~26 位的进位或借位, 也会被高 6 位忽略, 高 6 位保持 0。

因此程序空间的地址下限(地址 00000000H)和地址上限(地址 03FFFFFFH)成为连续的地址。回绕所指的就是这种地址下限和地址上限连续的情况。

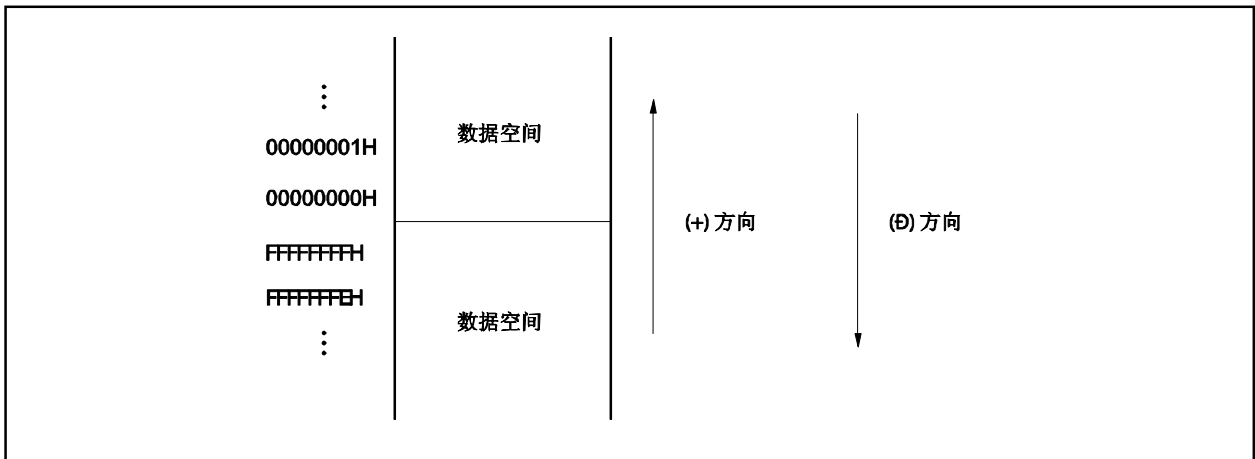
注意事项 由于从地址 03FFF000H~03FFFFFFH 的 4KB 范围是片上外围 I/O 设备区, 这个区域不能进行取指令操作。因此, 要避免执行可能使分支地址计算结果为此 4KB 中值的操作。



(2) 数据空间

结果超过 32 位的操作数地址计算操作将被忽略。

因此, 数据空间的最低地址 0000000H 和最高地址 FFFFFFFFH 之间是相连的, 回绕现象会发生在这两个地址附近。



3.4.3 存储器映射

V850ES/ KE2 存储器的保留区域如下所示。

图 3-2. 数据存储器映射 (物理地址)

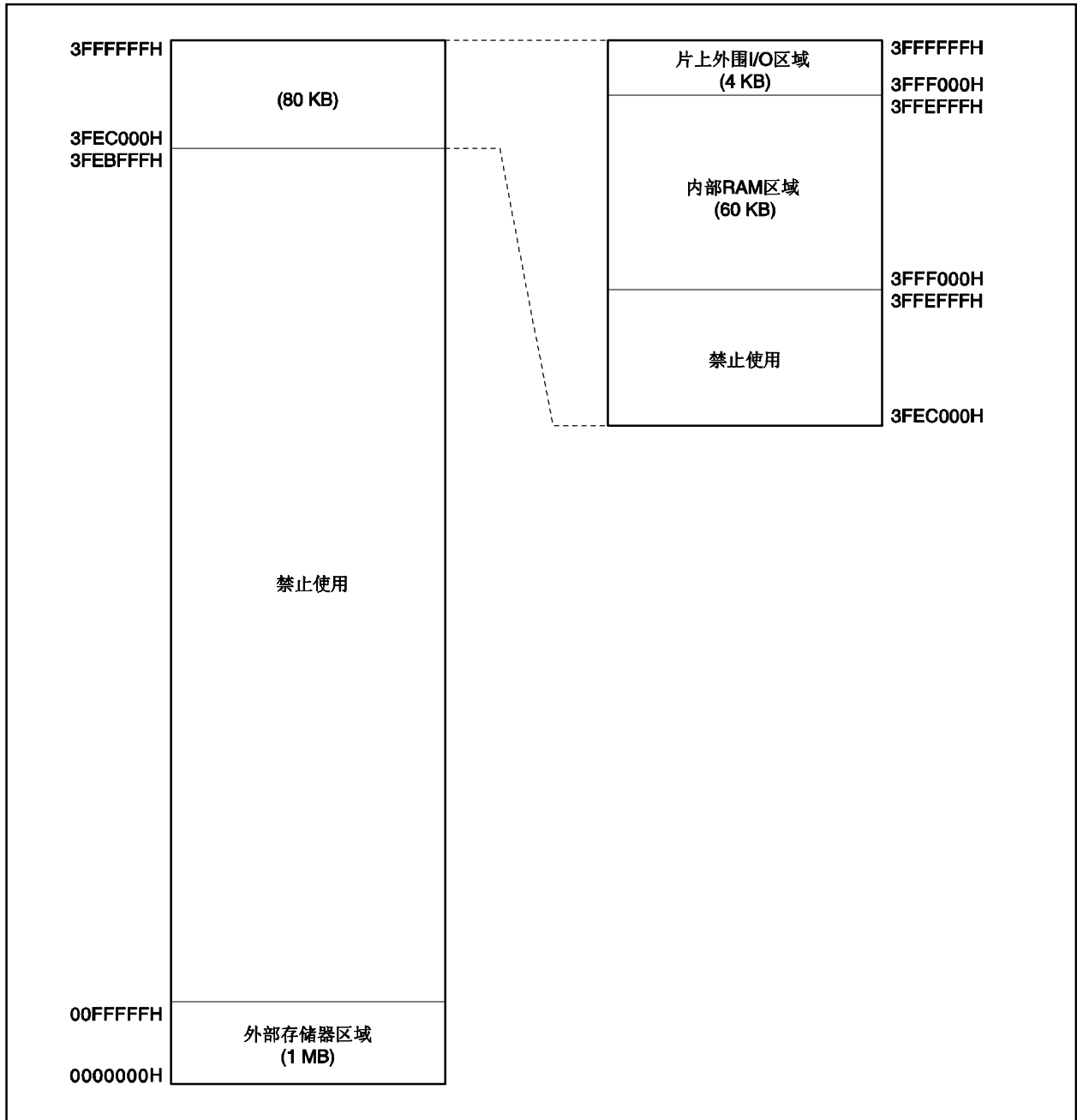
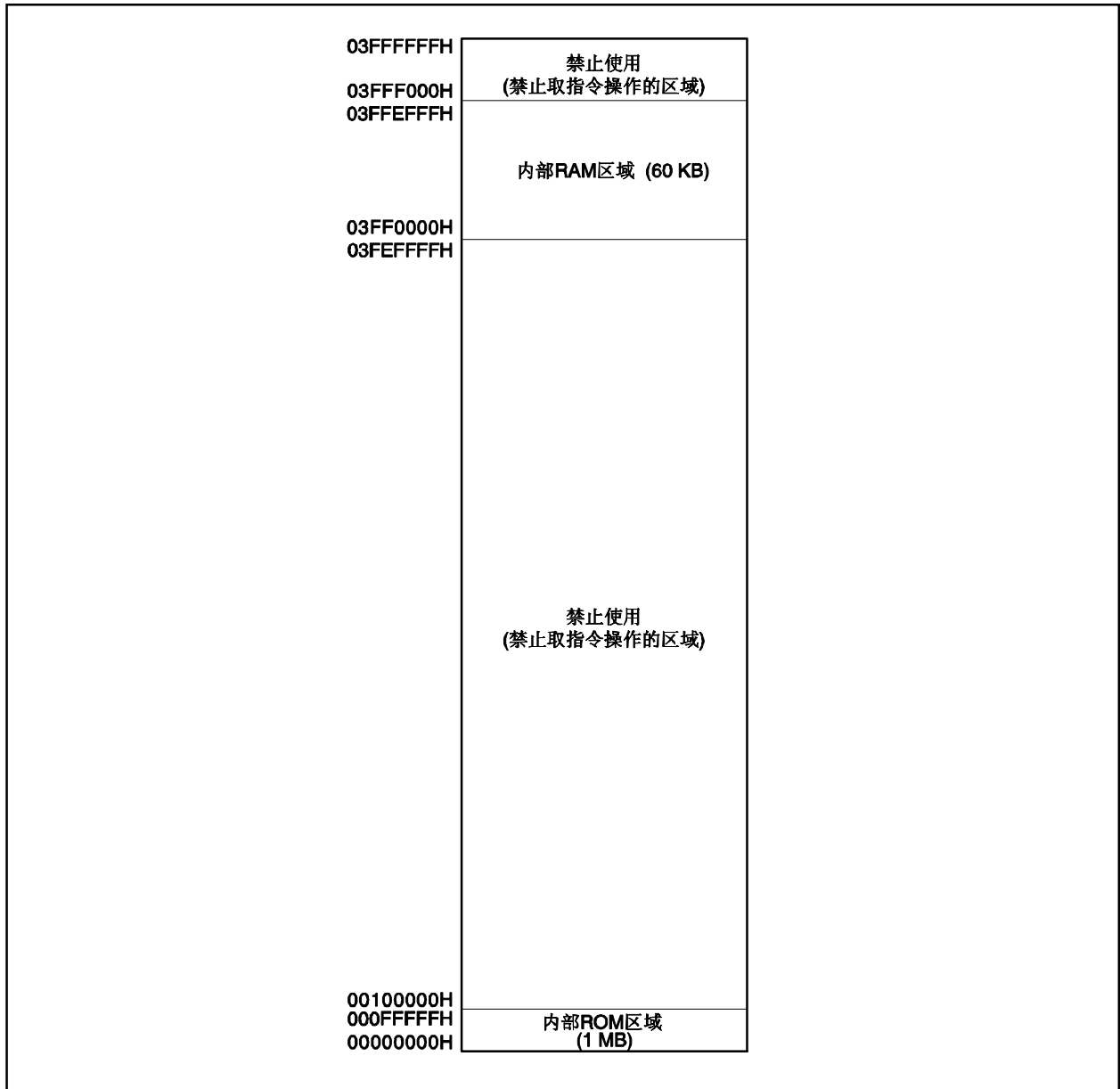


图 3-3. 程序存储器映射



3.4.4 区域

(1) 内部 ROM 区域

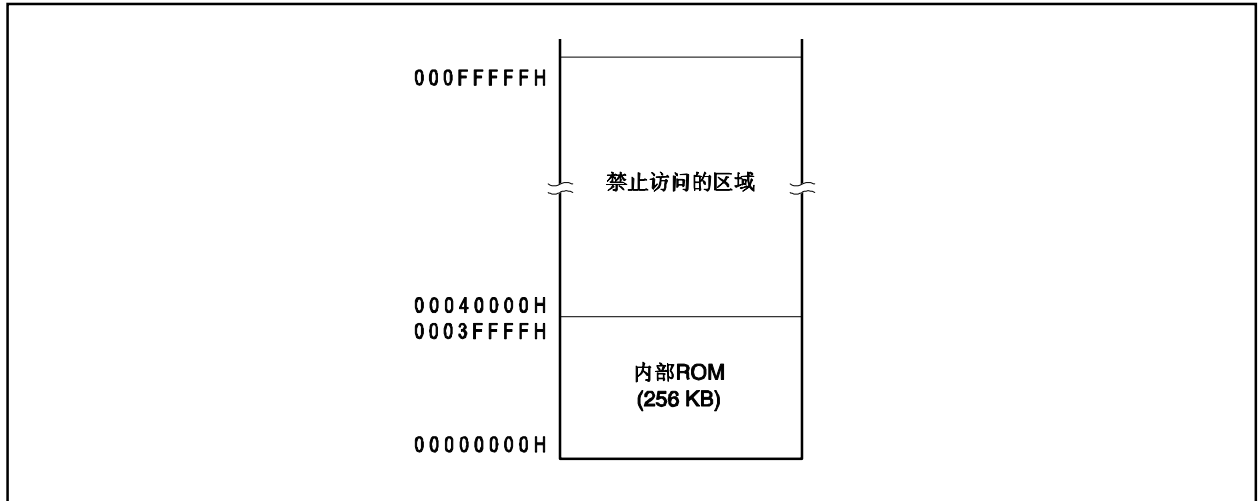
地址 0000000H~00FFFFFFH 1MB 的空间被保留为内部 ROM 区域。

(a) 内部 ROM (128KB)

μPD70F3726 版本的芯片的内部 ROM 区域为地址 0000000H ~ 001FFFFH 的 128KB 区域。

对地址 0020000H~00FFFFFFH 的操作是被禁止的。

图 3-4. 内部 ROM 区域 (128 KB)



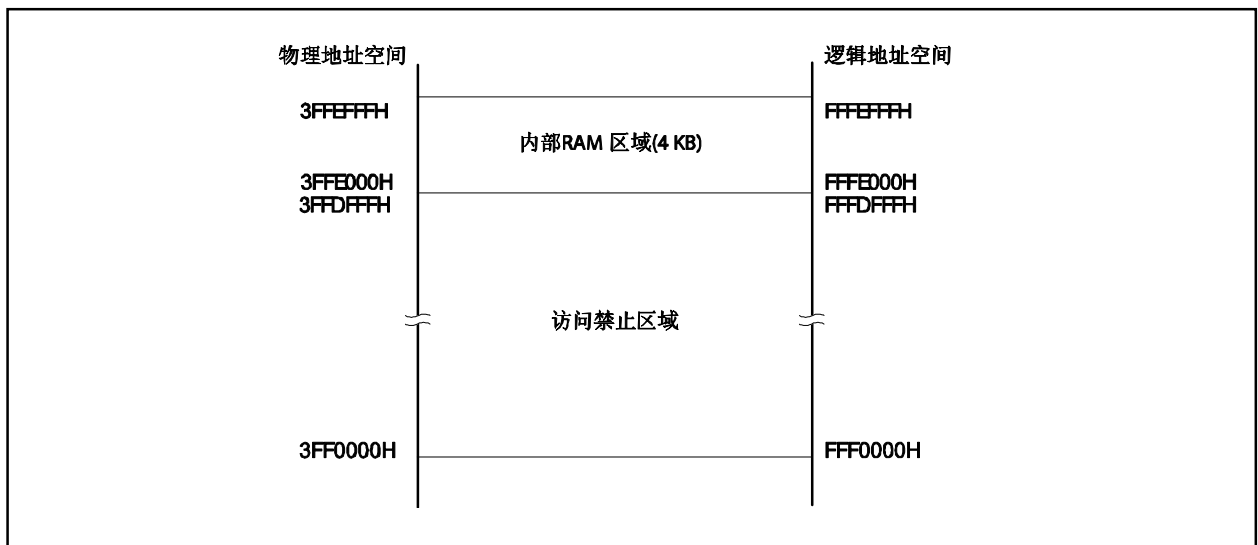
(2) 内部 RAM 区域

地址为 3FF0000H~3FFFFFFFFH 的 60KB 空间被保留为内部 RAM 区域。

地址为 3FFE000H~3FFFFFFFFH 的 4KB 区域为内部 RAM 的物理区域。

禁止对地址 3FF0000H~3FFDFFFH 进行操作。

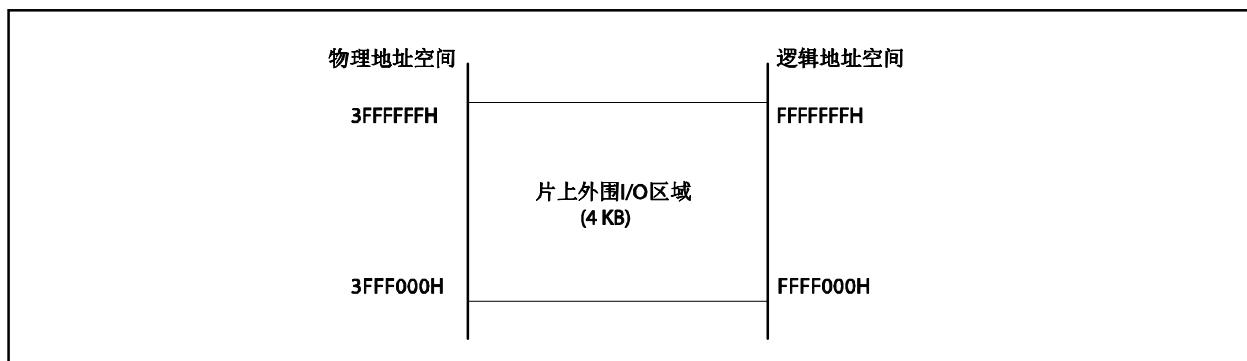
图 3-5. 内部 RAM 区域 (4 KB)



(3) 片上外围 I/O 区域

地址 3FFF000H~3FFFFFFH 的 4KB 空间是片上外围 I/O 区域。

图 3-6. 片上外围 I/O 区域



外围 I/O 寄存器被映射到片上外围 I/O 区间，这些寄存器是用来指定片上外围 I/O 的操作模式和对它们的状态进行监控的。程序不能在这个区间内取址。

- 注意事项**
1. 当寄存器被一个字操作指令读写时，一个字的空间被分为低 16 位和高 16 位分两次被字节操作先后读写，其中最低 2 位地址被忽略。
 2. 若对能进行字节操作的寄存器进行半字长(halfword)操作，那么读取时，高 8 位数据无效，写入时，低 8 位数据被写入寄存器。
 3. 未被指定为寄存器的地址为未来扩展保留。对这些地址操作结果的有效性是无法保证的。

<R>

(4) 访问所需的时钟数

下表给出访问各个资源所需的基时钟个数。

总线周期类型 \ 区域 (总线宽度)	内部 ROM (32 位)	内部 RAM (32 位)	片上外围 I/O (16 位)
取指令 (普通访问)	1	1 ^{#1}	-
取指令 (分支)	2	2 ^{#1}	-
操作数访问	3	1	3 ^{#2}

- 注**
1. 如果上述访问与数据访问冲突，时钟数加 1。
 2. 根据 VSWC 寄存器设置的变化，该值有所变化。

备注 单位：时钟数/一次访问

3.4.5 地址空间的推荐使用方法

V850ES/KE2 的架构要求在对数据区域中的操作数进行读写时，要确保有一个寄存器可以作为地址指针来使用。这个指针中的地址值的 $\pm 32\text{KB}$ 的地址范围内的操作数可以被指令直接访问。由于能够用来充当指针的通用寄存器数量有限，当指针值改变引起地址计算时，为了不影响系统的性能，要指定尽可能多的通用寄存器来保存变量值，这样也会减少程序所占的空间。

(1) 程序空间

PC（程序计数器）的 32 位中，高 6 位固定为 0，只有低 26 位有效。因此对于程序空间，从 00000000H 开始的 64 MB 连续地址空间无条件的与内存映射相对应。

可使用以下地址从 3FFE000H~3FFEFFFH (4 KB)作为内部 RAM 区间作为程序区间。

(2) 数据空间

在 V850ES/KE2 系列器件中，4GB 的 CPU 地址空间可被看作 64 个 64MB 的地址空间，因此 26 位地址的最低有效位(位 25)是作为符号扩展位被扩展至 32 位而被用作地址值的。

(a) 回绕式内存空间的应用举例

如果 R=r0(零寄存器)被用来指定 LD/ST disp16[R]指令，那么 00000000H $\pm 32\text{KB}$ 范围的空间都可通过带符号扩展的 16 位偏移量来寻址。包括内部硬件的所有资源都可用一个指针来寻址。

零寄存器(r0)是一个由硬件指定的值固定为零的寄存器，它可以有效地减少指定指针时对寄存器的需求量。

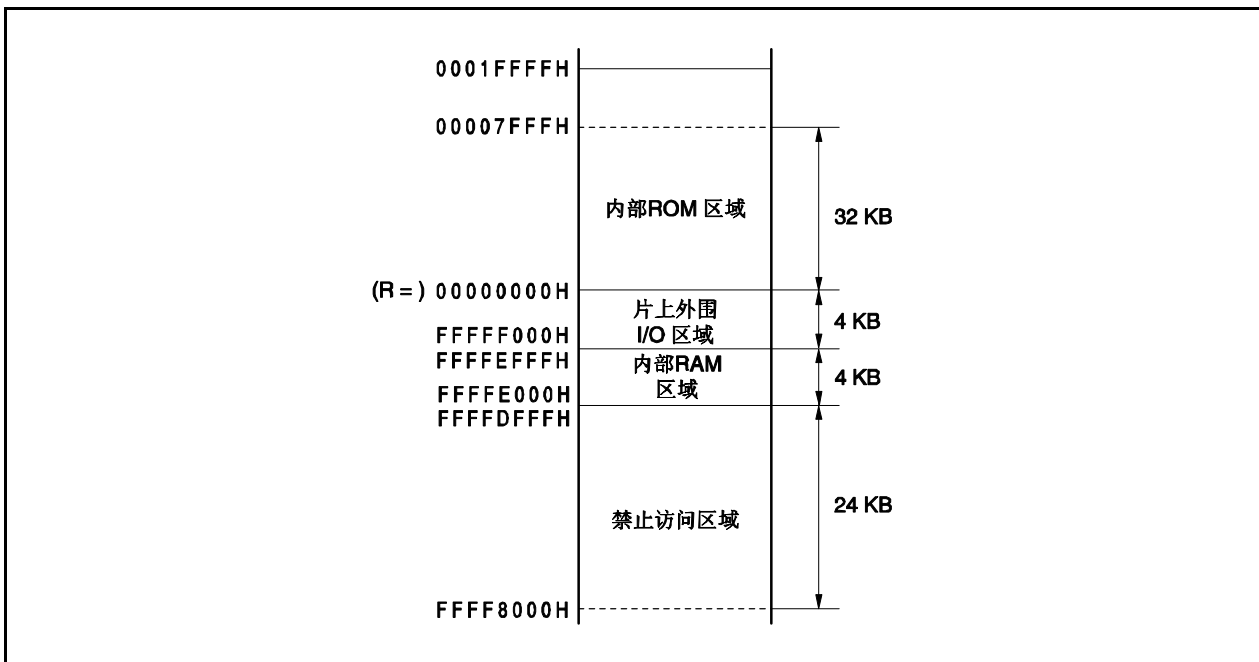
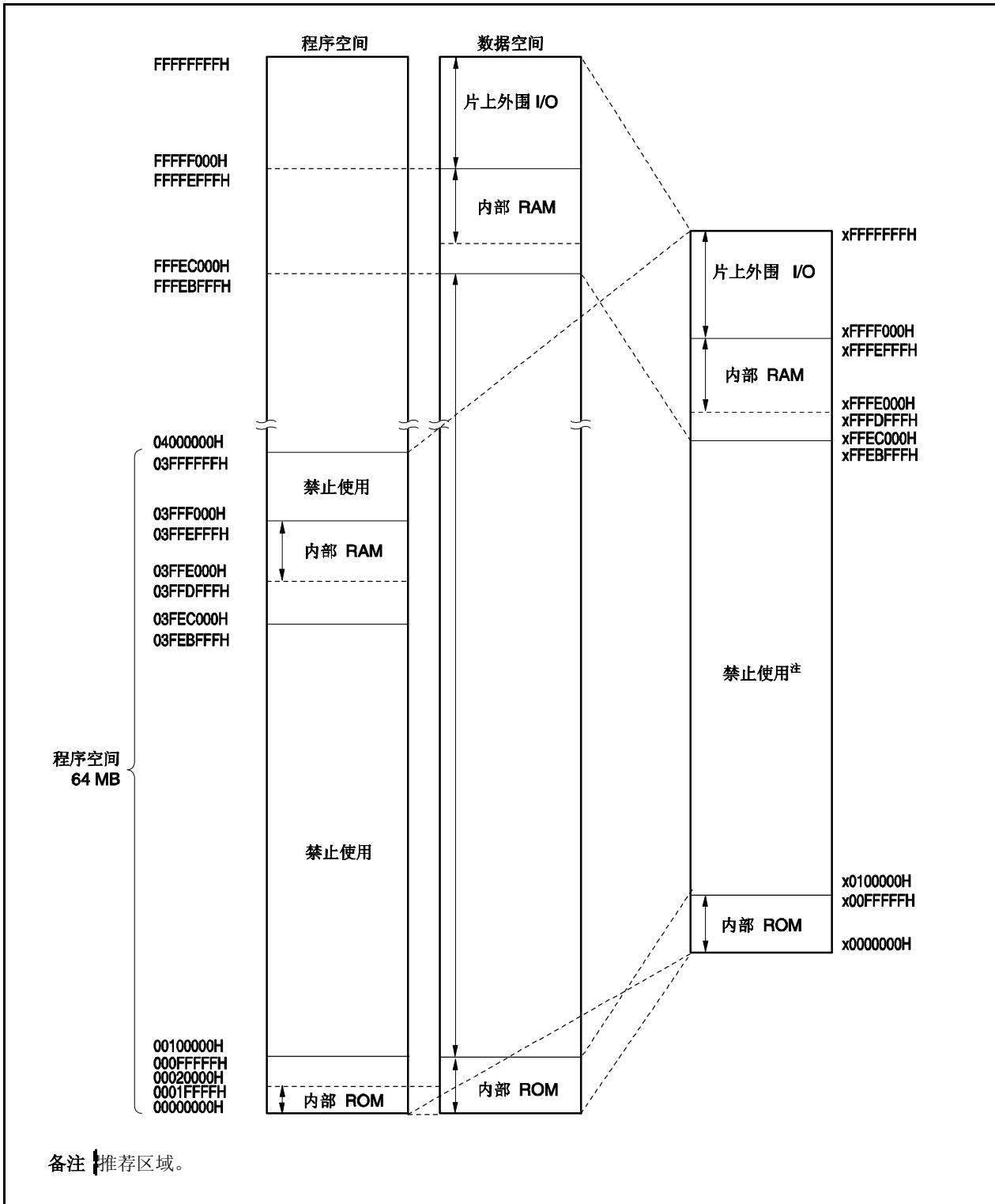


图 3-7. 推荐的存储器映射



3.4.6 外围 I/O 寄存器

(1/6)

地址	寄存器功能名称	符号	R/W	可操作位单元			复位后
				1	8	16	
FFFFF004H	端口 DL 寄存器	PDL	R/W	√	√		00H [※]
FFFFF00CH	端口 CM 寄存器	PCM	R/W	√	√		00H [※]
FFFFF024H	端口 DL 模式寄存器	PMDL	R/W	√	√		FFH
FFFFF02CH	端口 CM 模式寄存器	PMCM	R/W	√	√		FFH
FFFFF04CH	端口 CM 模式控制寄存器	PMCCM	R/W	√	√		00H
FFFFF06EH	系统等待控制寄存器	VSWC	R/W	√	√		77H
FFFFF100H	中断屏蔽寄存器 0	IMR0	R/W			√	FFFFH
FFFFF100H	中断屏蔽寄存器 0L	IMR0L	R/W	√	√		FFH
FFFFF101H	中断屏蔽寄存器 0H	IMR0H	R/W	√	√		FFH
FFFFF102H	中断屏蔽寄存器 1	IMR1	R/W			√	FFFFH
FFFFF102H	中断屏蔽寄存器 1L	IMR1L	R/W	√	√		FFH
FFFFF103H	中断屏蔽寄存器 1H	IMR1H	R/W	√	√		FFH
FFFFF106H	中断屏蔽寄存器 3	IMR3	R/W			√	FFFFH
FFFFF106H	中断屏蔽寄存器 3L	IMR3L	R/W	√	√		FFH
FFFFF110H	中断控制寄存器	WDT1IC	R/W	√	√		47H
FFFFF112H	中断控制寄存器	PIC0	R/W	√	√		47H
FFFFF114H	中断控制寄存器	PIC1	R/W	√	√		47H
FFFFF116H	中断控制寄存器	PIC2	R/W	√	√		47H
FFFFF118H	中断控制寄存器	PIC3	R/W	√	√		47H
FFFFF11AH	中断控制寄存器	PIC4	R/W	√	√		47H
FFFFF11CH	中断控制寄存器	PIC5	R/W	√	√		47H
FFFFF11EH	中断控制寄存器	PIC6	R/W	√	√		47H
FFFFF124H	中断控制寄存器	TM0IC10	R/W	√	√		47H
FFFFF126H	中断控制寄存器	TM0IC11	R/W	√	√		47H
FFFFF128H	中断控制寄存器	TM5IC0	R/W	√	√		47H
FFFFF12AH	中断控制寄存器	TM5IC1	R/W	√	√		47H
FFFFF12CH	中断控制寄存器	CSI0IC0	R/W	√	√		47H
FFFFF12EH	中断控制寄存器	CSI0IC1	R/W	√	√		47H
FFFFF130H	中断控制寄存器	SREIC0	R/W	√	√		47H
FFFFF132H	中断控制寄存器	SRIC0	R/W	√	√		47H
FFFFF134H	中断控制寄存器	STIC0	R/W	√	√		47H
FFFFF136H	中断控制寄存器	SREIC1	R/W	√	√		47H
FFFFF138H	中断控制寄存器	SRIC1	R/W	√	√		47H
FFFFF13AH	中断控制寄存器	STIC1	R/W	√	√		47H
FFFFF13CH	中断控制寄存器	TMHIC0	R/W	√	√		47H
FFFFF13EH	中断控制寄存器	TMHIC1	R/W	√	√		47H
FFFFF142H	中断控制寄存器	IICIC0	R/W	√	√		47H
FFFFF144H	中断控制寄存器	ADIC	R/W	√	√		47H
FFFFF146H	中断控制寄存器	KRIC	R/W	√	√		47H

注 输出锁存器的值为 00H。当这些寄存器处于输入模式时，读取到的数值就是引脚的状态。

(2/6)

地址	寄存器功能名	符号	R/W	可操作位单元			复位后
				1	8	16	
FFFFF148H	中断控制寄存器	WTIIC	R/W	√	√		47H
FFFFF14AH	中断控制寄存器	WTIC	R/W	√	√		47H
FFFFF14CH	中断控制寄存器	BRGIC	R/W	√	√		47H
FFFFF172H	中断控制寄存器	PIC7	R/W	√	√		47H
FFFFF174H	中断控制寄存器	TP0OVIC	R/W	√	√		47H
FFFFF176H	中断控制寄存器	TP0CCIC0	R/W	√	√		47H
FFFFF178H	中断控制寄存器	TP0CCIC1	R/W	√	√		47H
FFFFF1FAH	当前服务优先级寄存器	ISPR	R	√	√		00H
FFFFF1FCH	命令寄存器	PRCMD	W		√		未定义
FFFFF1FEH	节能控制寄存器	PSC	R/W	√	√		00H
FFFFF200H	A/D 转换器模式寄存器	ADM	R/W	√	√		00H
FFFFF201H	模拟输入通道参数寄存器	ADS	R/W	√	√		00H
FFFFF202H	电压不足比较模式寄存器	PFM	R/W	√	√		00H
FFFFF203H	电压不足比较阈值寄存器	PFT	R/W		√		00H
FFFFF204H	A/D 转换器结果寄存器	ADCR	R			√	未定义
FFFFF205H	A/D 转换器结果寄存器 H	ADCRH	R		√		未定义
FFFFF300H	寄存器按键返回模式寄存器	KRM	R/W	√	√		00H
FFFFF30AH	选择操作控制寄存器 1	SELCNT1	R/W	√	√		00H
FFFFF318H	数字噪声消除控制寄存器	NFC	R/W	√	√		00H
FFFFF400H	端口 0 寄存器	P0	R/W	√	√		00H ^注
FFFFF406H	端口 3 寄存器	P3	R/W			√	0000H ^注
FFFFF406H	端口 3 寄存器 L	P3L	R/W	√	√		00H ^注
FFFFF407H	端口 3 寄存器 H	P3H	R/W	√	√		00H ^注
FFFFF408H	端口 4 寄存器	P4	R/W	√	√		00H ^注
FFFFF40AH	端口 5 寄存器	P5	R/W	√	√		00H ^注
FFFFF40EH	端口 7 寄存器	P7	R		√		未定义
FFFFF412H	端口 9 寄存器	P9	R/W			√	0000H ^注
FFFFF412H	端口 9 寄存器 L	P9L	R/W	√	√		00H ^注
FFFFF413H	端口 9 寄存器 H	P9H	R/W	√	√		00H ^注
FFFFF420H	端口 0 模式寄存器	PM0	R/W	√	√		FFH
FFFFF426H	端口 3 模式寄存器	PM3	R/W			√	FFFFH
FFFFF426H	端口 3 模式寄存器 L	PM3L	R/W	√	√		FFH
FFFFF427H	端口 3 模式寄存器 H	PM3H	R/W	√	√		FFH
FFFFF428H	端口 4 模式寄存器	PM4	R/W	√	√		FFH
FFFFF42AH	端口 5 模式寄存器	PM5	R/W	√	√		FFH
FFFFF432H	端口 9 模式寄存器	PM9	R/W			√	FFFFH
FFFFF432H	端口 9 模式寄存器 L	PM9L	R/W	√	√		FFH
FFFFF433H	端口 9 模式寄存器 H	PM9H	R/W	√	√		FFH
FFFFF440H	端口 0 模式控制寄存器	PMC0	R/W	√	√		00H

注 输出锁存器的值为 00H 或 0000H。当这些寄存器处于输入模式时，读取到的数值就是引脚的状态。

地址	寄存器功能名	符号	R/W	可操作位单元			复位后
				1	8	16	
FFFFF446H	端口 3 模式控制寄存器	PMC3	R/W			√	0000H
FFFFF446H	端口 3 模式控制寄存器 L	PMC3L	R/W	√	√		00H
FFFFF447H	端口 3 模式控制寄存器 H	PMC3H	R/W	√	√		00H
FFFFF448H	端口 4 模式控制寄存器	PMC4	R/W	√	√		00H
FFFFF44AH	端口 5 模式控制寄存器	PMC5	R/W	√	√		00H
FFFFF452H	端口 9 模式控制寄存器	PMC9	R/W			√	0000H
FFFFF452H	端口 9 模式控制寄存器 L	PMC9L	R/W	√	√		00H
FFFFF453H	端口 9 模式控制寄存器 H	PMC9H	R/W	√	√		00H
FFFFF466H	端口 3 功能控制寄存器	PFC3	R/W	√	√		00H
FFFFF46AH	端口 5 功能控制寄存器	PFC5	R/W	√	√		00H
FFFFF472H	端口 9 功能控制寄存器	PFC9	R/W			√	0000H
FFFFF472H	端口 9 功能控制寄存器 L	PFC9L	R/W	√	√		00H
FFFFF473H	端口 9 功能控制寄存器 H	PFC9H	R/W	√	√		00H
FFFFF580H	8 位定时器 H 模式寄存器 0	TMHMD0	R/W	√	√		00H
FFFFF581H	8 位定时器 H 载体控制寄存器 0	TMCYC0	R/W	√	√		00H
FFFFF582H	8 位定时器 H 比较寄存器 00	CMP00	R/W		√		00H
FFFFF583H	8 位定时器 H 比较寄存器 01	CMP01	R/W		√		00H
FFFFF590H	8 位定时器 H 模式寄存器 1	TMHMD1	R/W	√	√		00H
FFFFF591H	8 位定时器 H 载体控制寄存器 1	TMCYC1	R/W	√	√		00H
FFFFF592H	8 位定时器 H 比较寄存器 10	CMP10	R/W		√		00H
FFFFF593H	8 位定时器 H 比较寄存器 11	CMP11	R/W		√		00H
FFFFF5A0H	TMP0 控制寄存器 0	TP0CTL0	R/W	√	√		00H
FFFFF5A1H	TMP0 控制寄存器 1	TP0CTL1	R/W	√	√		00H
FFFFF5A2H	TMP0 I/O 控制寄存器 0	TP0IOC0	R/W	√	√		00H
FFFFF5A3H	TMP0 I/O 控制寄存器 1	TP0IOC1	R/W	√	√		00H
FFFFF5A4H	TMP0 I/O 控制寄存器 2	TP0IOC2	R/W	√	√		00H
FFFFF5A5H	TMP0 选择寄存器 0	TP0OPT0	R/W	√	√		00H
FFFFF5A6H	TMP0 捕捉/比较寄存器 0	TP0CCR0	R/W			√	0000H
FFFFF5A8H	TMP0 捕捉/比较寄存器 1	TP0CCR1	R/W			√	0000H
FFFFF5AAH	TMP0 计数器读取缓存寄存器	TP0CNT	R			√	0000H
FFFFF5C0H	16 位定时器定时器 5	TM5	R			√	0000H
FFFFF5C0H	8 位定时器定时器 50	TM50	R		√		00H
FFFFF5C1H	8 位定时器定时器 51	TM51	R		√		00H
FFFFF5C2H	16 位定时器比较寄存器 5	CR5	R/W			√	0000H
FFFFF5C2H	8 位定时器比较寄存器 50	CR50	R/W		√		00H
FFFFF5C3H	8 位定时器比较寄存器 51	CR51	R/W		√		00H
FFFFF5C4H	5 时钟定时器选择寄存器 5	TCL5	R/W			√	0000H
FFFFF5C4H	时钟定时器选择寄存器 50	TCL50	R/W		√		00H
FFFFF5C5H	时钟定时器选择寄存器 51	TCL51	R/W		√		00H

地址	寄存器功能名称	符号	R/W	可操作位单元			复位后
				1	8	16	
FFFFF5C6H	16 位定时器模式控制寄存器 5	TMC5	R/W			√	0000H
FFFFF5C6H	8 位定时器模式控制寄存器 50	TMC50	R/W	√	√		00H
FFFFF5C7H	8 位定时器模式控制寄存器 51	TMC51	R/W	√	√		00H
FFFFF610H	16 位定时器计数器 01	TM01	R			√	0000H
FFFFF612H	16 位定时器捕捉/比较 寄存器 010	CR010	R/W			√	0000H
FFFFF614H	16 位定时器捕捉/比较 寄存器 011	CR011	R/W			√	0000H
FFFFF616H	16 位定时器模式控制寄存器 01	TMC01	R/W	√	√		00H
FFFFF617H	预分频模式寄存器 01	PRM01	R/W	√	√		00H
FFFFF618H	捕捉/比较控制寄存器 01	CRC01	R/W	√	√		00H
FFFFF619H	16 位定时器输出控制寄存器 01	TOC01	R/W	√	√		00H
FFFFF680H	时钟定时器操作模式寄存器	WTM	R/W	√	√		00H
FFFFF6C0H	振荡稳定时间选择寄存器	OSTS	R/W		√		01H
FFFFF6C1H	看门狗定时器时钟选择寄存器	WDCS	R/W		√		00H
FFFFF6C2H	看门狗定时器模式寄存器 1	WDTM1	R/W	√	√		00H
FFFFF6D0H	看门狗定时器模式寄存器 2	WDTM2	R/W		√		67H
FFFFF6D1H	看门狗定时器允许寄存器	WDTE	R/W		√		9AH
FFFFF6E0H	实时输出缓存寄存器 L0	RTBL0	R/W	√	√		00H
FFFFF6E2H	实时输出缓存寄存器 H0	RTBH0	R/W	√	√		00H
FFFFF6E4H	实时输出端口模式寄存器 0	RTPM0	R/W	√	√		00H
FFFFF6E5H	实时输出端口控制寄存器 0	RTPC0	R/W	√	√		00H
FFFFF706H	端口 3 功能控制扩展寄存器	PFCE3	R/W	√	√		00H
FFFFF802H	系统状态寄存器	SYS	R/W	√	√		00H
FFFFF806H	PLL 控制寄存器	PLLCTL	R/W	√	√		01H
FFFFF820H	节能模式寄存器	PSMR	R/W	√	√		00H
FFFFF828H	处理器时钟控制寄存器	PCC	R/W	√	√		03H
FFFFF8B0H	间隔定时器 BRG 模式寄存器	PRSM	R/W		√		00H
FFFFF8B1H	间隔定时器 BRG 比较寄存器	PRSCM	R/W		√		00H
FFFFFA00H	异步串行接口模式寄存器 0	ASIM0	R/W	√	√		01H
FFFFFA02H	接收缓存寄存器 0	RXB0	R		√		FFH
FFFFFA03H	异步串行接口状态寄存器 0	ASIS0	R		√		00H
FFFFFA04H	发送缓存寄存器 0	TXB0	R/W		√		FFH
FFFFFA05H	异步串行接口发送状态寄存器 0	ASIF0	R	√	√		00H
FFFFFA06H	时钟选择寄存器 0	CKSR0	R/W		√		00H
FFFFFA07H	波特率生成控制寄存器 0	BRGC0	R/W		√		FFH
FFFFFA10H	异步串行接口模式寄存器 1	ASIM1	R/W	√	√		01H
FFFFFA12H	接收缓存寄存器 1	RXB1	R		√		FFH
FFFFFA13H	异步串行接口状态寄存器 1	ASIS1	R		√		00H
FFFFFA14H	发送缓存寄存器 1	TXB1	R/W		√		FFH
FFFFFA15H	异步串行接口发送状态寄存器 1	ASIF1	R	√	√		00H
FFFFFA16H	时钟选择寄存器 1	CKSR1	R/W		√		00H
FFFFFA17H	波特率生成控制寄存器 1	BRGC1	R/W		√		FFH

地址	寄存器功能名称	符号	R/W	可操作位单元			复位后
				1	8	16	
FFFFB00H	TIP00 噪声清除控制寄存器	P0NFC	R/W	√	√		00H
FFFFB04H	TIP01 噪声清除控制寄存器	P1NFC	R/W	√	√		00H
FFFFC00H	外部中断下降沿指定寄存器 0	INTF0	R/W	√	√		00H
FFFFC06H	外部中断下降沿指定寄存器 3	INTF3	R/W	√	√		00H
FFFFC13H	外部中断下降沿指定寄存器 9H	INTF9H	R/W	√	√		00H
FFFFC20H	外部中断上升沿指定寄存器 0	INTR0	R/W	√	√		00H
FFFFC26H	外部中断上升沿指定寄存器 3	INTR3	R/W	√	√		00H
FFFFC33H	外部中断上升沿指定寄存器 9H	INTR9H	R/W	√	√		00H
FFFFC40H	上拉电阻选择寄存器 0	PU0	R/W	√	√		00H
FFFFC46H	上拉电阻选择寄存器 3	PU3	R/W	√	√		00H
FFFFC48H	上拉电阻选择寄存器 4	PU4	R/W	√	√		00H
FFFFC4AH	上拉电阻选择寄存器 5	PU5	R/W	√	√		00H
FFFFC52H	上拉电阻选择寄存器 9	PU9	R/W			√	0000H
FFFFC52H	上拉电阻选择寄存器 9L	PU9L	R/W	√	√		00H
FFFFC53H	上拉电阻选择寄存器 9H	PU9H	R/W	√	√		00H
FFFFC67H	端口 3 功能寄存器 H	PF3H	R/W	√	√		00H
FFFFC68H	端口 4 功能寄存器	PF4	R/W	√	√		00H
FFFFC73H	端口 3 功能寄存器 H	PF9H	R/W	√	√		00H
FFFFD00H	计时 串行接口模式寄存器 00	CSIM00	R/W	√	√		00H
FFFFD01H	计时串行接口时钟选择 寄存器 0	CSIC0	R/W	√	√		00H
FFFFD02H	计时串行接口接收缓冲寄存器 0	SIRB0	R			√	0000H
FFFFD02H	计时串行接口接收缓冲寄存器 0L	SIRB0L	R		√		00H
FFFFD04H	计时串行接口发送缓冲 寄存器 0	SOTB0	R/W			√	0000H
FFFFD04H	计时串行接口发送缓冲 寄存器 0L	SOTB0L	R/W		√		00H
FFFFD06H	计时串行接口只读接收缓冲寄存器 0	SIRBE0	R			√	0000H
FFFFD06H	计时串行接口只读接收缓冲寄存器 0L	SIRBE0L	R		√		00H
FFFFD08H	计时串行接口初始发送缓冲寄存器 0	SOTBF0	R/W			√	0000H
FFFFD08H	计时串行接口初始发送缓冲寄存器 0L	SOTBF0L	R/W		√		00H
FFFFD0AH	串行 I/O 移位寄存器 0	SIO00	R/W			√	00H
FFFFD0AH	串行 I/O 移位寄存器 0L	SIO00L	R/W		√		0000H
FFFFD10H	计时串行接口模式寄存器 01	CSIM01	R/W	√	√		00H
FFFFD11H	计时串行接口时钟选择 寄存器 1	CSIC1	R/W	√	√		00H
FFFFD12H	计时串行接口接收缓冲寄存器 1	SIRB1	R			√	0000H
FFFFD12H	计时串行接口接收缓冲寄存器 1L	SIRB1L	R		√		00H
FFFFD14H	计时串行接口发送缓冲 寄存器 1	SOTB1	R/W			√	0000H
FFFFD14H	计时串行接口发送缓冲 寄存器 1L	SOTB1L	R/W		√		00H
FFFFD16H	计时串行接口只读接收缓冲寄存器 1	SIRBE1	R			√	0000H
FFFFD16H	计时串行接口只读接收缓冲寄存器 1L	SIRBE1L	R		√		00H
FFFFD18H	计时串行接口初始发送缓冲寄存器 1	SOTBF1	R/W			√	0000H
FFFFD18H	计时串行接口初始发送缓冲寄存器 1L	SOTBF1L	R/W		√		00H

(6/6)

地址	寄存器功能名称	符号	R/W	可操作位单元			复位后
				1	8	16	
FFFFD1AH	串行 I/O 移位寄存器 1	SIO01	R/W			√	00H
FFFFD1AH	串行 I/O 移位寄存器 1L	SIO01L	R/W		√		0000H
FFFFD80H	IIC 移位寄存器 0	IIC0	R/W		√		00H
FFFFD82H	IIC 控制寄存器 0	IICC0	R/W	√	√		00H
FFFFD83H	地址寄存器 0	SVA0	R/W		√		00H
FFFFD84H	IIC 时钟选择寄存器 0	IICCL0	R/W	√	√		00H
FFFFD85H	IIC 功能扩展寄存器 0	IICX0	R/W	√	√		00H
FFFFD86H	IIC 状态寄存器 0	IICS0	R	√	√		00H
FFFFD8AH	IIC 标志寄存器 0	IICF0	R/W	√	√		00H
FFFFFF44H	上拉电阻选择寄存器 DL	PUDL	R/W	√	√		00H
FFFFFF4CH	上拉电阻选择寄存器 CM	PUCM	R/W	√	√		00H

3.4.7 特殊寄存器

特殊寄存器是为了防止由非有意的程序循环引起的无效数据写入而受到保护的寄存器。V850ES/KE2 中共有如下 3 个特殊寄存器。

- 节能控制寄存器 (PSC)
- 处理器时钟控制寄存器 (PCC)
- 看门狗定时器模式寄存器(WDTM1)

此外，器件还提供了一个命令寄存器(PRCMD)来阻截对特殊寄存器的非法写操作，这样应用系统就不会因为程序挂起而停止工作了。对特殊寄存器的正常写操作是通过特定的流程实现的，非法的写操作将被记录在系统状态寄存器(SYS)中。

(1) 向特殊寄存器中写入数据

请按照以下流程向特殊寄存器中写入数据。

- <1> 将要向特殊寄存器中写入的数据储存在通用寄存器中。
- <2> 将步骤<1>中存储的数据写入 PRCMD 寄存器。
- <3> 将设置的数据写入特殊寄存器 中(用下面的指令)。
 - 存指令(ST/STT 指令)
 - 位操作指令(SET1/CLR1/NOT1 指令)
- <4> ~ <8>插入 NOP 指令 (5 个指令)^注。

[举例描述] 写 PSC 寄存器 (设置待机模式)

```

ST.B r11, PSMR[r0]    ; 设置 PSMR 寄存器 (设置 IDLE1, IDLE2, 和 STOP 模式)
<1> MOV 0x02, r10
<2> ST.B r10, PRCMD[r0] ; 写 PRCMD 寄存器
<3> ST.B r10, PSC[r0]  ; 设置 PSC 寄存器
<4> NOP 注              ; 哑指令
<5> NOP 注              ; 哑指令
<6> NOP 注              ; 哑指令
<7> NOP 注              ; 哑指令
<8> NOP 注              ; 哑指令
(下一条指令)

```

读取特殊寄存器时不需经过特定流程。

注 设置 IDLE 模式 或 STOP 模式(通过将 PSC.STP 位置 1)后，要立即插入 5 条 NOP 指令。

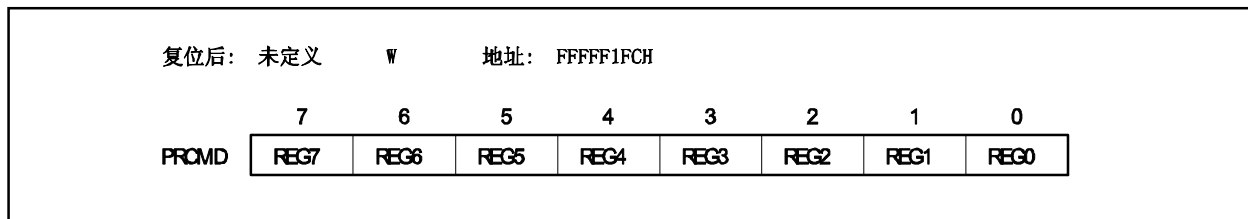
- 注意事项**
1. 对于 PRCMD 寄存器的存储指令，系统将不响应中断。存储指令应该是按上述的第<2>和<3>步骤连续执行的。若在步骤<2>和<3>之间执行了其他指令，而这个指令又进行了中断的响应，则将打乱上述的顺序导致设备误操作。
 2. 尽管对 PRCMD 写入的数据为虚数据，但还是请使用与设置特殊寄存器(例子中的<3>)的通用寄存器相同的寄存器来向 PRCMD 寄存器(例子中的<2>)写入虚数据。这个过程也同样可以用于使用通用寄存器进行寻址的操作中。

(2) 命令寄存器 (PRCMD)

命令寄存器 (PRCMD) 有 8 位，用于保护那些会严重影响系统运行状态的寄存器不被轻易写入，从而使应用系统不会因为无意的程序循环引起的程序挂起而停止工作。除非数据先被写入 PRCMD，否则向特殊寄存器中写入的数据是无效的。

数据只有通过特定的流程才可能被写入特殊寄存器，这样就能够防止对特殊寄存器的非法写操作。

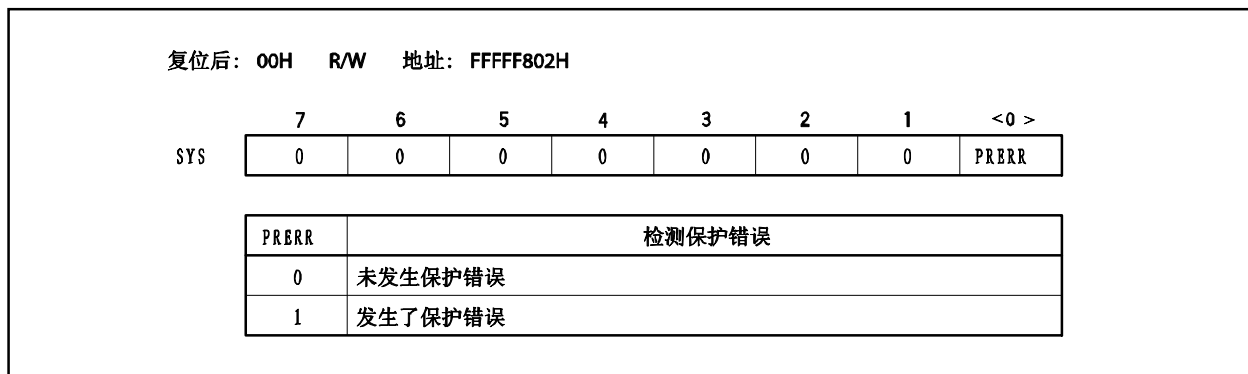
该寄存器仅支持 8 位写操作 (若对其进行读取，则所读数据不确定)。



(3) 系统状态寄存器 (SYS)

该寄存器中包含了用于指定整个系统的操作状态的状态标志。

该寄存器支持 8 位或 1 位读写方式。



PRERR 标志会在以下情况下改变。

(a) 置位条件 (PRERR 标志 = 1)

- (i) 当向特殊寄存器写入数据之前没有对 PRCMD 寄存器进行写入时 (在执行 **3.4.7 (1) 向特殊寄存器中写入数据** 中的步骤<3>之前没有执行步骤<2>)。
- (ii) 当对 PRCMD 寄存器进行写入操作后 (包括位操作指令), 没有进行对特殊寄存器的写入, 而是写入了片上外围 I/O 寄存器时 (如果 **3.4.7 (1) 向特殊寄存器中写入数据** 中的步骤<3>不是对特殊寄存器的写入操作)。

备注 对于 WDTM 寄存器以外其他的特殊寄存器(PCC 和 PSC 寄存器), 即使在写 PRCMD 寄存器和写一个特殊寄存器之间执行读取片上外围 I/O 寄存器指令 (除了位操作指令) (内部 RAM 的访问等), PRERR 标志也不设置, 并且设置的数据可以写入特殊寄存器。

(b) 清零条件 (PRERR 标志 = 0)

- (i) 当向 PRERR 标志写入 0 时
- (ii) 当系统被复位时

- 注意事项**
1. 如果在对 PRCMD 寄存器进行写入之后, 向 SYS 寄存器(并非特殊寄存器)的 PRERR 位写入 0, 那么 PRERR 位将被清零(写入命令优先)。
 2. 如果在对 PRCMD 寄存器进行写入之后, 又执行了对 PRCMD 寄存器(并非特殊寄存器)的写入操作, 那么 PRERR 位将被置 1。

3.4.8 注意事项

(1) 需要首先设置的寄存器

在使用 V850ES/KE2 器件时，需要首先设置的寄存器。

- 系统等待控制寄存器 (VSWC)
- 看门狗定时器模式寄存器 2 (WDTM2)

在设置好 VSWC 和 WDTM2 寄存器之后，再根据需要对其他寄存器进行设置。

当使用了外部总线时，设置好上述寄存器之后，再通过设置各端口相关的寄存器将总线引脚设置为复用功能中的总线控制引脚模式。

(a) 系统等待控制寄存器 (VSWC)

VSWC 寄存器用于控制总线访问片上外围 I/O 寄存器的等待时间。

访问一个片上外围 I/O 寄存器需要 3 个时钟周期的时间(不包含等待周期)。V850ES/KE2 需要根据不同的操作频率插入不同的等待周期数。因此，请根据所使用的内部系统时钟，将以下对应的数值设置到 VSWC 寄存器中。

该寄存器支持 8 位读写方式。(地址：FFFFF06EH，复位后：77H)

操作条件	系统内部时钟频率(f_{CLK})	VSWC 寄存器设置	等待周期数
$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$32\text{ kHz} \leq f_{CLK} < 16.6\text{ MHz}$	00H	0 (无等待)
	$16.6\text{ MHz} \leq f_{CLK} \leq 20\text{ MHz}$	01H	1
$4.0\text{ V} \leq V_{DD} < 4.5\text{ V}$	$32\text{ kHz} \leq f_{CLK} \leq 16\text{ MHz}$	00H	0 (无等待)
$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	$32\text{ kHz} \leq f_{CLK} < 8.3\text{ MHz}$	00H	0 (无等待)
	$8.3\text{ MHz} \leq f_{CLK} \leq 10\text{ MHz}$	01H	1

(b) 看门狗定时器模式寄存器 2 (WDTM2)

WDTM2 寄存器用于设置看门狗定时器 2 的溢出时间和操作时钟。

看门狗定时器 2 将在器件复位后自动从复位模式启动。向 WDTM2 寄存器写入相应数值以确认该操作。

关于更多细节，敬请参阅第 11 章 看门狗定时器 2 的功能。

(2) 访问特定的片上外围 I/O 寄存器

本产品具有两种内部系统总线。

一种是 CPU 总线，另一种是用于与低速外围硬件进行通信的外围总线。

CPU 总线和外围总线的时钟是异步的。如果发生 CPU 访问和外围硬件访问的冲突，那么总线上传输的数据就可能为错误数据。因此，在有可能发生总线冲突的情况下，对外围硬件进行访问时，器件会改变 CPU 访问的时钟周期，这样就保证了访问所传输数据的正确性。正因如此，CPU 不会进行对下一条指令的处理，而是进入等待状态。当这种等待状态产生时，执行一条指令的所需的时钟周期数就需要加上等待周期。

注意对于那些对实时处理要求较高的应用来说，一定要考虑上述的情况。

当片上外围 I/O 寄存器被访问时，插入的等待周期数可能要多于 VSWC 寄存器所设置的数量。

此时的访问情况以及计算所插入的等待周期数量(CPU 时钟数)的方法如下所示。

待插入的等待周期数 = $(2 + m) \times k$ (时钟)

寄存器访问特定的片上外围 I/O 寄存器的次数 = $3 + m + (2 + m) \times k$ (时钟)

外围功能	寄存器名称	访问	k
看门狗定时器 1 (WDT1)	WDTM1	写	1~ 5
	<等待周期数目的计算注> $k = \{(1/f_x) \times 2 / ((2 + m) / f_{CPU})\} + 1$ f_x : 主时钟振荡频率		
看门狗定时器 2 (WDT2)	WDTM2	写	3 (固定)
16-定时器/事件计数器 P0 (TMP0)	TP0CCR0, TP0CCR1, TP0CNT	读	1
	<等待周期数目的计算注> $k = \{(1/f_{xx}) / ((2 + m) / f_{CPU})\} + 1$		
	TP0CCR0, TP0CCR1	写	0 ~ 2
	<等待周期数目的计算注> $k = \{(1/f_{xx}) \times 5 / ((2 + m) / f_{CPU})\}$ 寄存器在连续写入同一个寄存器时会发生等待		
16 位定时器 /事件计数器 01(TM01)	TMC01	读-修改-写	1 (固定) 在写入时发生等待
I ² C0	IICS0	读	1 (固定)
异步串行接口 0 和 1(UART0, UART1)	ASIS0, ASIS1	读	1 (固定)
实时输出功能 0 (RTO0)	RTBL0, RTBH0	写 (当 RTPC0.RTPOE0 位= 0)	1
A/D 转换器	ADM, ADS, PFM, PFT	写	1 或 2
	ADCR, ADCRH	读	1 或 2
	<等待周期数目计算注> $\{(1/f_{xx}) \times 2 / ((2 + m) / f_{CPU})\} + 1$		

注 在计算等待数时，结果的小数部分乘 $(1/f_{CPU})$ ，若小于等于 $(1/f_{CPU})/(2 + m)$ 则舍，大于则入。

- 注意事项**
1. 如果取自内部 ROM 或内部 RAM，等待数如下所示。如果取自外部存储器，等待数则可能减少。外部存储器访问周期的影响根据等待设置等的变化有所不同。但是上述等待数是最大值，因此不会产生更大的值。
 2. 当 CPU 工作在副时钟且 X1 引脚上没有时钟输入时，不要访问发生等待的寄存器。如果等待发生，则只能由复位释放。

备注 在计算等待周期数目时：
 f_{CPU} : CPU 时钟频率
 f_{xx} : 主时钟频率
 m : 把 VSWC 的寄存器额位 2 设置为 0

当 VSWC 寄存器 = 00H: $m = 0$

当 VSWC 寄存器 = 01H: $m = 1$

(3) 指令与中断请求的冲突限制

(a) 描述

如果在<1>中指令后 **sld** 前的<2>中的指令解码运行出现冲突，<1>中指令前的中断请求完整，则<1>中指令的执行结果不能保存在寄存器中。

指令<1>

- **ld** 指令: ld.b, ld.h, ld.w, ld.bu, ld.hu
- **sld** 指令: sld.b, sld.h, sld.w, sld.bu, sld.hu
- **n** 乘法指令: mul, mulh, mulhi, mulu

指令<2>

mov reg1, reg2	not reg1, reg2	satsubr reg1, reg2	satsub reg1, reg2
satadd reg1, reg2	satadd imm5, reg2	或 reg1, reg2	xor reg1, reg2
and reg1, reg2	reg2	subr reg1, reg2	sub reg1, reg2
add reg1, reg2	tst reg1, reg2	cmp reg1, reg2	cmp imm5, reg2
mulh reg1, reg2	add imm5, reg2	sar imm5, reg2	shl imm5, reg2
	shr imm5, reg2		

<例>

<i> ld.w [r11], r10	如果 ld 指令 <i> 的执行完成之前， sld 指令<iii>之前的 mov 指令<ii>的解码过程和一个中断请求发生冲突，那么指令<i>的操作结果有可能不被保存在寄存器中。
:	
:	
<ii> mov r10, r28	
<iii> sld.w 0x28, r10	

(b) 定时器解决方法

<1> 使用编译器(CA850)时

请使用CA850 Ver. 2.61或更高版本，因为这些版本的编译器可以自动避免生成上述情形的指令序列

<2> 通过汇编器测量计数器

当执行指令 <ii>后立即执行 **sld** 指令时，用如下两种方法中的任意一种避免上述操作。

- 在 **sld** 指令之前插入一条 **nop** 指令。
- 在上述的指令<ii>中，不要与 **sld** 指令使用相同的寄存器。

第四章 端口功能

4.1 特点

- 只输入端口：8 引脚
- I/O 端口：43 引脚
 - 固定为 N 沟道开漏输出：2
 - 可选择 N 沟道开漏输出：6
- 输入或者输出可以以位为单位指定

4.2 端口的基本配置

V850ES/KE2 由端口 0, 3 ~ 5, 7, 9, CM 和 DL 共 51 个 I/O 端口引脚（包括 8 仅输入端口引脚）组成。端口配置情况如下图所示。

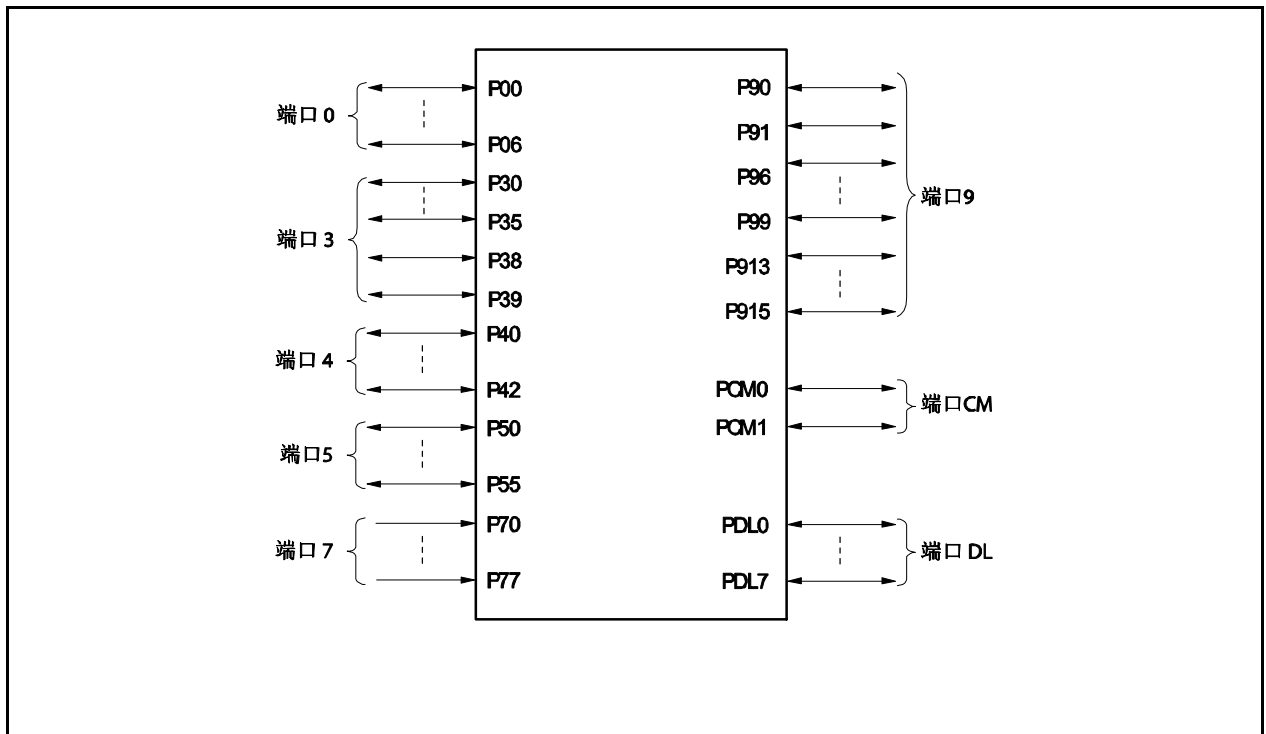


表 4-1. V850ES/KE2 引脚的 I/O 缓冲器电源

电源	对应引脚
AV _{REF0}	端口 7
EV _{DD}	RESET, 端口 0, 3 ~ 5, 9, CM, DL

4.3 端口配置

表 4-2. 端口配置

项目	配置
控制寄存器	端口 n 寄存器 (Pn: n = 0, 3 ~5, 7, 9, CM, DL) 端口 n 模式寄存器 (PMn: n = 0, 3 ~5, 9, CM, DL) 端口 n 模式控制寄存器 (PMcn: n = 0, 3 ~5, 9, CM) 端口 n 功能控制寄存器 (PFCn: n = 3, 5, 9) 端口 n 功能寄存器 (PFn: n = 3, 4, 9) 端口 3 功能控制扩展寄存器 (PFCE3) 上拉电阻选项寄存器 (PUn: n = 0, 3 ~5, 9, CM, DL)
端口	只输入: 8 I/O: 43
上拉电阻	软件控制: 41

(1) 端口 n 寄存器 (Pn)

通过读写 Pn 寄存器, 将数据输出到外部设备或从外部设备输入。Pn 寄存器由保持输出数据的端口锁存器和读引脚状态的电路组成。

Pn 寄存器的每一位都与端口 n 的一个引脚对应。该寄存器支持 1 位读写方式。

复位后: 00H (输出锁存) R/W								
	7	6	5	4	3	2	1	0
Pn	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0
Pnm	输出数据的控制 (输出模式下)							
0	输出 0							
1	输出 1							

注 只输入端口引脚未定义。

根据各寄存器的设置, 从 Pn 寄存器读写数据如下。

表 4-3. 读取/写入 Pn 寄存器

PMCn 寄存器的设置	PMn 寄存器设置	向 Pn 寄存器写入	从 Pn 寄存器读取
端口模式 (PMCnm 位 = 0)	输出模式 (PMnm 位 = 0)	数据写入输出锁存器 ^注 。 在端口模式下 (PMCn = 0)，输出锁存器的内容从引脚输出。	读取输出锁存的值。
	输入模式 (PMnm 位 = 1)	数据写入输出锁存器 ^注 。 引脚状态不受影响。	读取引脚状态。
复用功能模式 (PMCnm 位 = 1)	输出模式 (PMnm 位 = 0)	数据写入输出锁存器 ^注 。 引脚状态不受影响。 引脚按复用功能引脚操作。	<ul style="list-style-type: none"> 当复用功能输出时 读取复用功能的输出状态。 当复用功能输入时 读取输出锁存值。
	输入模式 (PMnm 位 = 1)	数据写入输出锁存器 ^注 。 引脚状态不受影响。 引脚按复用功能引脚操作。	读取引脚状态。

注 除非有新的值写入，否则写到输出锁存的值会一直保持。

(2) 端口 n 模式寄存器 (PMn)

PMn 寄存器指定对应端口引脚的输入或输出模式。

PMn 寄存器的每一位都与端口 n 的一个引脚对应，可以以位为单位指定模式。

复位后: FFH R/W								
	7	6	5	4	3	2	1	0
PMn	PMn7	PMn6	PMn5	PMn4	PMn3	PMn2	PMn1	PMn0
	PMnm	输入/输出模式的控制						
	0	输出模式						
	1	输入模式						

(3) 端口 n 模式控制寄存器 (PMcN)

PMcN 寄存器指定端口模式或复用功能。

PMcN 寄存器的每一位都与端口 n 的一个引脚对应，可以以位为单位指定端口模式。

复位后: 00H		R/W						
	7	6	5	4	3	2	1	0
PMcN	PMcN7	PMcN6	PMcN5	PMcN4	PMcN3	PMcN2	PMcN1	PMcN0
	PMcNm	指定操作模式						
	0	端口模式						
	1	复用功能模式						

(4) 端口 n 功能控制寄存器 (PFCn)

如果引脚有两个或更多复用功能，PFCn 寄存器指定要使用引脚的复用功能。

此寄存器的每一位都与端口 n 的一个引脚对应，可以以位为单位指定端口引脚的复用功能。

复位后: 00H		R/W						
	7	6	5	4	3	2	1	0
PFCn	PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0
	PFCnm	指定复用功能						
	0	复用功能 1						
	1	复用功能 2						

(5) 端口 n 功能控制扩展寄存器 (PFCEn)

如果引脚有三个或更多的复用功能，PFCEn 寄存器指定要使用引脚的复用功能。

此寄存器的每一位都与端口 n 的一个引脚对应，可以以位为单位指定端口引脚的复用功能。

复位后: 00H		R/W						
	7	6	5	4	3	2	1	0
PFCEn	PFCEn7	PFCEn6	PFCEn5	PFCEn4	PFCEn3	PFCEn2	PFCEn1	PFCEn0
	7	6	5	4	3	2	1	0
PFCn	PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0
PFCEnm	PFCnm	复用功能详述						
0	0	复用功能 1						
0	1	复用功能 2						
1	0	复用功能 3						
1	1	复用功能 4						

(6) 端口 n 功能寄存器 (PFn)

PFn 寄存器指定正常输出或 N 沟道开漏输出。

此寄存器的每一位都与端口 n 的一个引脚对应，可以以位为单位指定端口引脚的输出模式。

复位后: 00H		R/W						
	7	6	5	4	3	2	1	0
PFn	PFn7	PFn6	PFn5	PFn4	PFn3	PFn2	PFn1	PFn0
PFnm ^注	正常输出 /N-沟开漏输出控制							
0	正常输出 (CMOS输出)							
1	N沟开漏输出							

注 无论 PMCn 寄存器的设置如何，只有在 PMn.PMnm 位为 0（输出模式）时，PFnm 位才有效。当 PFnm 位为 1 时，（输入模式），在 PFn 寄存器的设置值是无效的。

例

<1> 当 PFn 寄存器值有效时
 PFnm 位 = 1...设置 N 沟道开漏输出。
 PMnm 位 = 0...设置输出模式。
 PMCnm 位 = 0 或 1

<2> 当 PFn 寄存器值无效时
 PFnm 位 = 0...设置 N 沟道开漏输出。
 PMnm 位 = 1...设置输入模式。
 PMCnm 位 = 0 或 1

(7) 上拉电阻选项寄存器 (PUn)

Pun 寄存器指定了与片上上拉电阻的连接。

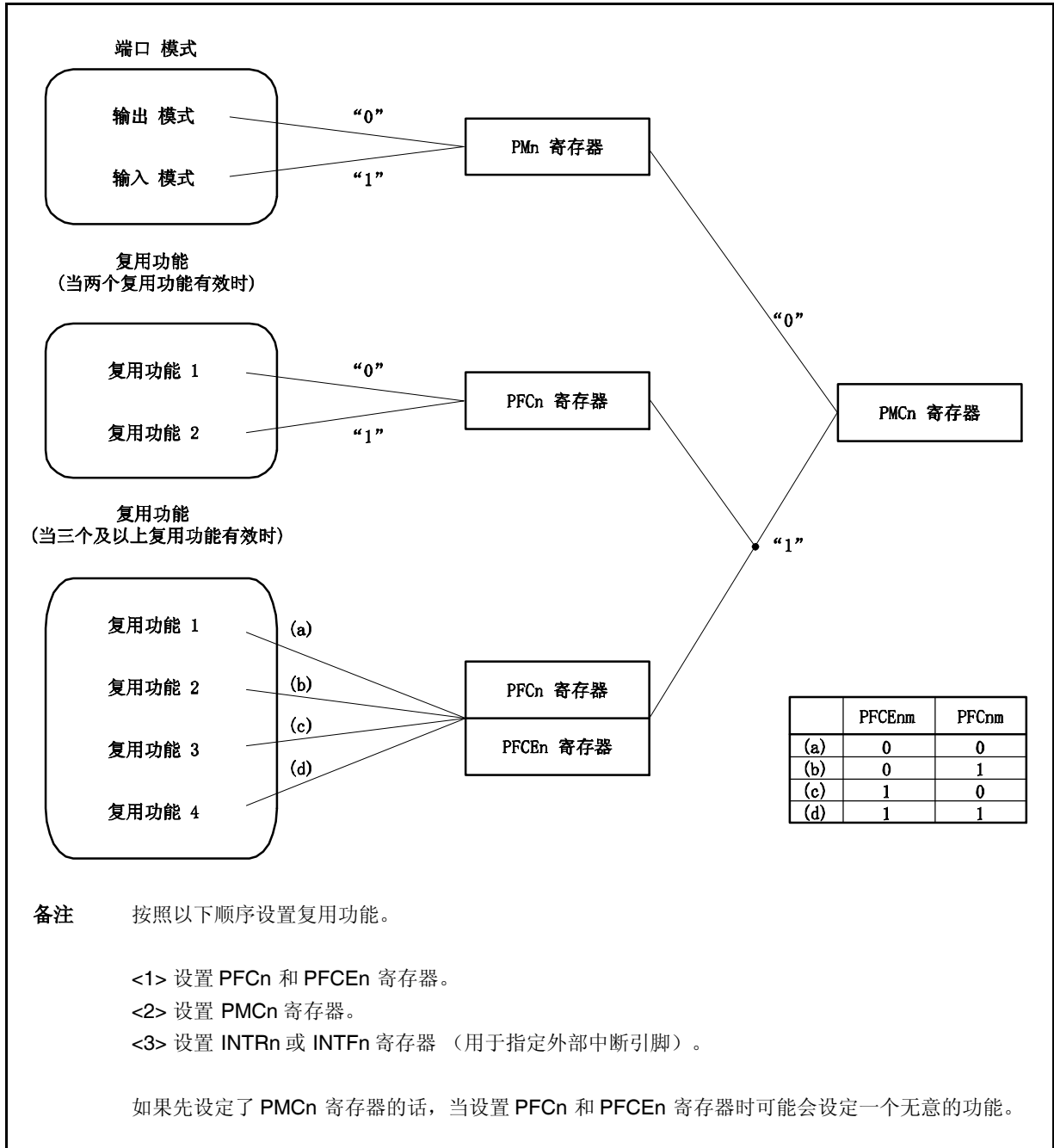
Pun 寄存器的每一位与端口 n 的一位对应。可以位为单位对其进行指定。

复位后: 00H R/W								
PUn	7	6	5	4	3	2	1	0
	PUn7	PUn6	PUn5	PUn4	PUn3	PUn2	PUn1	PUn0
PUnm	控制片上上拉电阻的连接							
0	不接							
1	连接							

(8) 端口设置

按照下图设置端口。

图 4-1. 每个寄存器和引脚功能的设置



4.3.1 端口 0

端口 0 是一个可以以位为单位控制 I/O 设置的 7 位 I/O 端口。

端口 0 包含以下复用功能引脚。

表 4-4. 端口 0 复用功能引脚

引脚编号	引脚名称	复用功能	I/O	PULL [*]	备注	模块类型
12	P00	TOH0	输出	是	-	D0-U
13	P01	TOH1	输出			D0-U
14	P02	NMI	输入		模拟噪声清除	D1-SUIL
15	P03	INTP0	输入			D1-SUIL
16	P04	INTP1	输入			D1-SUIL
17	P05	INTP2	输入			D1-SUIL
18	P06	INTP3	输入		模拟/数字噪声清除	D1-SUIL

注 软件上拉功能

注意事项 P02 ~ P06 引脚在复用功能的输入模式具有滞后性，但在端口模式没有滞后性。

(1) 端口 0 寄存器 (P0)

复位后: 00H (输出锁存) R/W 地址: FFFF400H								
	7	6	5	4	3	2	1	0
P0	0	P06	P05	P04	P03	P02	P01	P00
P0n	输出数据控制 (输出模式) (n = 0 ~ 6)							
0	输出0							
1	输出1							

(2) 端口 0 模式寄存器 (PM0)

复位后: FFH R/W 地址: FFFF420H								
	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00
PM0n	I/O 模式控制 (n = 0 ~ 6)							
0	输出模式							
1	输入模式							

(3) 端口 0 模式控制寄存器 (PMC0)

复位后: 00H R/W 地址: FFFFF440H

	7	6	5	4	3	2	1	0
PMC0	0	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	PMC00
PMC06	P06 引脚操作模式的规格							
0	I/O 端口							
1	INTP3 输入							
PMC05	P05 引脚操作模式的规格							
0	I/O 端口							
1	INTP2 输入							
PMC04	P04 引脚操作模式的规格							
0	I/O 端口							
1	INTP1 输入							
PMC03	P03 引脚操作模式的规格							
0	I/O 端口							
1	INTP0 输入							
PMC02	P02 引脚操作模式的规格							
0	I/O 端口							
1	NMI 输入							
PMC01	P01 引脚操作模式的规格							
0	I/O 端口							
1	TOH1 输出							
PMC00	P00 引脚操作模式的规格							
0	I/O 端口							
1	TOH0 输出							

(4) 上拉电阻选项寄存器 0 (PU0)

复位后: 00H R/W 地址: FFFFFC40H

	7	6	5	4	3	2	1	0
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00
PU0n	片上上拉电阻连接的控制 (n = 0 ~ 6)							
0	不接							
1	连接							

4.3.2 端口 3

端口 3 是一个可以以位为单位控制 I/O 设置的 8 位 I/O 端口。

端口 3 包含以下复用功能。

表 4-5. 端口 3 复用功能引脚

引脚编号	引脚名称	复用功能	I/O	PULL [*]	备注	模块类型
22	P30	TXD0	输出	是	-	D-U
23	P31	RXD0/INTP7	输入			D1-SUIHL
24	P32	ASCK0/ADTRG/TO01	I/O			E10-SUL
25	P33	TIP00/TOP00	I/O			Gxx10-SUL
26	P34	TIP01/TOP01	I/O			Gxx10-SUL
27	P35	TI010/TO01	I/O			E10-SUL
55	P38	SDA0	I/O	否	N 沟道开漏输出	D2-SNFH
56	P39	SCL0	I/O			D2-SNFH

注 软件上拉功能

注意事项 P31 ~ P35, P38 及 P39 在复用功能期间有滞后性, 但在端口模式下不会。

(1) 端口 3 寄存器 (P3)

复位后: 00H (输出锁存)		R/W	地址:		P3 FFFF406H, P3L FFFF406H, P3H FFFF407H			
P3 (P3H ^注)	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	P39	P38
(P3L)	7	6	5	4	3	2	1	0
	0	0	P35	P34	P33	P32	P31	P30
P3n	输出数据的控制(输出模式) (n = 0 ~ 5, 8, 9)							
0	输出0							
1	输出1							

注 要以 8 位 或 1 位为单位读写 P3 寄存器的位 8 ~ 15，需要将他们指定为 P3H 寄存器的位 0 ~ 7。

备注 P3 寄存器可以 16 位为单元读写。
此外当使用 P3 寄存器的高 8 位作为 P3H 寄存器，低 8 位作为 P3L 寄存器时，P3 可以以 8 位或 1 位为单位读写。

(2) 端口 3 模式寄存器 (PM3)

复位后: FFFFH		R/W	地址:		PM3 FFFF426H, PM3L FFFF426H, PM3H FFFF427H			
PM3 (PM3H ^注)	15	14	13	12	11	10	9	8
	1	1	1	1	1	1	PM39	PM38
(PM3L)	7	6	5	4	3	2	1	0
	1	1	PM35	PM34	PM33	PM32	PM31	PM30
PM3n	I/O 模式的控制 (n = 0 ~ 5, 8, 9)							
0	输出模式							
1	输入模式							

注 要以 8 位 或 1 位为单位读写 PM3 寄存器的位 8 ~ 15，需要将他们指定为 PM3H 寄存器的位 0 ~ 7。

备注 PM3 寄存器支持 16 位读写方式。
此外当使用 PM3 寄存器的高 8 位作为 PM3H 寄存器，低 8 位作为 PM3L 寄存器时，PM 寄存器支持 8 位或 1 位读写方式。

(3) 端口 3 模式控制寄存器 (PMC3)

复位后: 0000H		R/W	地址: PMC3 FFFF446H, PMC3L FFFF446H, PMC3H FFFF447H					
PMC3 (PMC3H ^{注1})	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	PMC39	PMC38
(PMC3L)	7	6	5	4	3	2	1	0
	0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30
PMC39	P39引脚操作模式的规格							
0	I/O 端口							
1	SCL0 I/O							
PMC38	P38引脚操作模式的规格							
0	I/O 端口							
1	SDA0 I/O							
PMC35	P35引脚操作模式的规格							
0	I/O 端口							
1	TI010 输入/TO01输出							
PMC34	P34引脚操作模式的规格							
0	I/O端口							
1	TIP01输出/TOP01 输出							
PMC33	P33 引脚操作模式的规格							
0	I/O 端口							
1	TIP00 输入/TOP00 输出							
PMC32	P32 引脚操作模式的规格							
0	I/O 端口							
1	ASCK0 输入/ADTRG 输入/TO01输出							
PMC31	P31引脚操作模式的规格							
0	I/O 端口							
1	RXD0 输入/INTP7 input ^{注2}							
PMC30	P30 引脚操作模式的规格							
0	I/O 端口							
1	TXD0 输出							

注

- 要以 8 位 或 1 位为单位读写 PMC3 寄存器的位 8 ~ 15，需要将他们指定为 PMC3H 寄存器的位 0 ~ 7。
- INTP7 引脚和 RXD0 引脚为复用功能引脚。当使用引脚的 RXD0 功能，需要禁止复用功能 INTP7 引脚的边沿检测（将 INTF3.INTF31 和 INTR3.INTR31 清 0）。当使用引脚的 INTP7 功能，停止 UART0 接收操作（将 ASIM0.RXE0 位清零）。

备注 PMC3 寄存器支持 16 位读写方式。
此外当使用 PMC3 寄存器的高 8 位作为 PMC3H 寄存器，低 8 位作为 PMC3L 寄存器时，PMC3 寄存器支持 8 位或 1 位读写方式。

(4) 端口 3 功能控制寄存器 H (PF3H)

复位后: 00H R/W 地址: FFFFC67H

	7	6	5	4	3	2	1	0
PF3H	0	0	0	0	0	0	PF39	PF38

PF3n	正常端口/复用功能的规格 (n = 8, 9)
0	作为正常端口使用时 (N沟道开漏输出)
1	作为复用功能使用时 (N沟道开漏输出)

注意事项 当 P38 和 P39 作为 N 沟道开漏输出复用功能引脚使用时, 请按下列顺序设置。
一定要先将接口锁存器设置为 1, 再将引脚设置 N 沟道开漏输出。
P3n 位 = 1 → PF3n 位 = 1 → PMC3n 位 = 1

(5) 端口 3 功能控制寄存器 (PFC3)

复位后: 00H R/W 地址: FFFF466H

	7	6	5	4	3	2	1	0
PFC3	0	0	PFC35	PFC34	PFC33	PFC32	0	0

备注 有关设定复用功能的详细信息, 请参考 4.3.2 (7) 指定端口 3 的复用功能引脚。

(6) 端口 3 功能控制扩展寄存器 (PFCE3)

复位后: 00H R/W 地址: FFFF706H

	7	6	5	4	3	2	1	0
PFCE3	0	0	0	PFCE34	PFCE33	0	0	0

备注 有关设定复用功能的详细信息, 请参考 4.3.2 (7) 指定端口 3 的复用功能引脚。

(7) 指定端口 3 的复用功能引脚

PFC35	P35 引脚复用功能的指定	
0	TI010 输入	
1	TO01 输出	

PFCE34	PFC34	P34 引脚复用功能的指定
0	0	禁止设置
0	1	禁止设置
1	0	TIP01 输入
1	1	TOP01 输出

PFCE33	PFC33	P33 引脚复用功能的指定
0	0	禁止设置
0	1	禁止设置
1	0	TIP00 输入
1	1	TOP00 输出

PFC32	P32 引脚复用功能的指定	
0	ASCK0/ADTRG [※] 输入	
1	TO01 输出	

注 ASCK0 和 ADTRG 引脚是复用功能引脚。当使用引脚的 ASCK0 功能时，禁止复用功能 ADTRG 引脚的触发输入（将 ADS.TRG 位清零或将 ADS.ADTMD 位设置为 1）。当使用引脚的 ADTRG 功能时，不要将 UART0 操作时钟设置为外部输入（设置 CKSR0.TPS03 ~ CKSR0.TPS00 位为 1011 以外的数）。

注意事项 在 PFC3n 和 PFCE3n 位保持初始值 0 时，通过 PMC3.PMC3n 位指定 P3n 引脚的复用功能时，输出变得不确定。因此，要设置 P3n 引脚的复用功能，首先将 PFC3n 和 PFCE3n 位设置为 1，再将 PMC3n 位设置为 1（n = 3, 4）。

(8) 上拉电阻选项寄存器 3 (PU3)

复位后: 00H R/W 地址: FFFFC46H

	7	6	5	4	3	2	1	0
PU3	0	0	PU35	PU34	PU33	PU32	PU31	PU30

PU3n	片上上拉电阻连接的控制(n=0~5)	
0	不接	
1	连接	

4.3.3 端口 4

端口 4 为可以以位为单位控制 I/O 设置的 3 位 I/O 端口。
 端口 4 包含以下复用功能引脚。

表 4-6. 端口 4 复用功能引脚

引脚编号	引脚名称	复用功能	I/O	PULL [※]	备注	模块类型
19	P40	SI00	输入	是	-	D1-SUL
20	P41	SO00	输出		可选择 N 沟道开漏输出。	D0-UF
21	P42	$\overline{\text{SCK00}}$	I/O		D2-SUFL	

注 软件上拉功能

注意事项 P40 和 P42 引脚在复用功能的输入模式具有滞后性，但在端口模式没有滞后性。

(1) 端口 4 寄存器 (P4)

复位后: 00H (输出锁存) R/W 地址: FFFFF408H								
	7	6	5	4	3	2	1	0
P4	0	0	0	0	0	P42	P41	P40
P4n	输出数据的控制 (输出模式) (n = 0 ~ 2)							
0	输出0							
1	输出1							

(2) 端口 4 模式寄存器 (PM4)

复位后: FFH R/W 地址: FFFFF428H								
	7	6	5	4	3	2	1	0
PM4	1	1	1	1	1	PM42	PM41	PM40
PM4n	I/O模式的控制 (n = 0 ~ 2)							
0	输出模式							
1	输入模式							

(3) 端口 4 模式控制寄存器 (PMC4)

(1/2)

复位后: 00H R/W 地址: FFFF448H								
	7	6	5	4	3	2	1	0
PMC4	0	0	0	0	0	PMC42	PMC41	PMC40
PMC42	P42引脚操作模式的规格							
0	I/O 端口							
1	SCK00 I/O							
PMC41	P41 引脚操作模式的规格							
0	I/O 端口							
1	SO00 输出							
PMC40	P40 引脚操作模式的规格							
0	I/O 端口							
1	SI00 输入							

(4) 端口 4 功能寄存器 (PF4)

复位后: 00H R/W 地址: FFFF68H								
	7	6	5	4	3	2	1	0
PF4	0	0	0	0	0	PF42	PF41	0
PF4n	正常输出/N沟道开漏输出的控制 (n = 1, 2)							
0	正常输出							
1	N沟道开漏输出							
<p>注意事项 当 P41 和 P42 作为 N 沟道开漏输出复用功能引脚, 按如下顺序设置。 一定要先将端口锁存器设置为 1, 然后再将引脚设置为 N 沟道开漏输出。 P4n 位 = 1 → PF4n 位 = 1 → PMC4n 位 = 1</p>								

(5) 上拉电阻选项寄存器 4 (PU4)

复位后: 00H R/W 地址: FFFF48H								
	7	6	5	4	3	2	1	0
PU4	0	0	0	0	0	PU42	PU41	PU40
PU4n	片上上拉电阻连接的控制 (n = 0 ~ 2)							
0	不接							
1	连接							

4.3.4 端口 5

端口 5 为可以以位为单位控制 I/O 设置的 6 位 I/O 端口。

端口 5 包含以下复用功能引脚。

表 4-7. 端口 5 复用功能引脚

引脚编号	引脚名称	复用功能	I/O	PULL [※]	备注	模块类型
28	P50	TI011/RTP00/KR0	I/O	是	-	E10-SULT
29	P51	TI50/RTP01/KR1	I/O			E10-SULT
30	P52	TO50/RTP02/KR2	I/O			E00-SUT
31	P53	RTP03/KR3	I/O			Ex0-SUT
34	P54	RTP04/KR4	I/O			Ex0-SUT
35	P55	RTP05/KR5	I/O			Ex0-SUT

注 软件上拉功能

(1) 端口 5 寄存器 (P5)

复位后: 00H (输出锁存) R/W 地址: FFFF40AH								
	7	6	5	4	3	2	1	0
P5	0	0	P55	P54	P53	P52	P51	P50
P5n	输出数据的控制 (输出模式) (n = 0 ~ 5)							
0	输出0							
1	输出1							

(2) 端口 5 模式寄存器 (PM5)

复位后: FFH R/W 地址: FFFF42AH								
	7	6	5	4	3	2	1	0
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50
PM5n	I/O 模式的控制 (n = 0 ~ 5)							
0	输出模式							
1	输入模式							

(3) 端口 5 模式控制寄存器 (PMC5)

复位后: 00H R/W 地址: FFFF44AH

	7	6	5	4	3	2	1	0
PMC5	0	0	PMC55	PMC54	PMC53	PMC52	PMC51	PMC50
PMC55	P55 引脚操作模式的规格							
0	I/O端口/KR5 输入							
1	RTP05 输出							
PMC54	P54 引脚操作模式的规格							
0	I/O 端口/KR4输入							
1	RTP04 输出							
PMC53	P53引脚操作模式的规格							
0	I/O 端口/KR3 输入							
1	RTP03 输出							
PMC52	P52 引脚操作模式的规格							
0	I/O 端口/KR2 输入							
1	TO50 输出/RTP02 输出							
PMC51	P51 引脚操作模式的规格							
0	I/O 端口/KR1输入							
1	TI50 输入/RTP01 输出							
PMC50	P50引脚操作模式的规格							
0	I/O 端口/KR0 输入							
1	TI011 输入/RTP00 输出							

(4) 端口 5 功能控制寄存器 (PFC5)

注意事项 在 PFC5n 位保持初始值 0 时, 通过 PMC5.PMC5n 位指定 P5n 引脚的复用功能时, 输出变得不确定。因此, 要设置 P5n 引脚的复用功能 2, 首先将 PFC5n 位设置为 1, 再将 PMC5n 引脚设置为 1 (n = 3 ~ 5)。

复位后: 00H R/W 地址: FFFF46AH								
	7	6	5	4	3	2	1	0
PFC5	0	0	PFC55	PFC54	PFC53	PFC52	PFC51	PFC50
PFC55	P55 复用功能引脚的规格							
1	RTP05 输出							
PFC54	P54 复用功能引脚的规格							
1	RTP04 输出							
PFC53	P53 复用功能引脚的规格							
1	RTP03 输出							
PFC52	复用功能引脚的规格							
0	TO50 输出							
1	RTP02 输出							
PFC51	P51 复用功能引脚的规格							
0	TI50 输入							
1	RTP01 输出							
PFC50	P50 复用功能引脚的规格							
0	TI011 输入							
1	RTP00 输出							

(5) 上拉电阻选项寄存器 5 (PU5)

复位后: 00H R/W 地址: FFFF44AH								
	7	6	5	4	3	2	1	0
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50
PU5n	片上上拉电阻连接的控制 (n = 0 ~ 5)							
0	不接							
1	连接							

4.3.5 端口 7

端口 7 为 8 位仅输入端口，所有引脚只可以输入。

端口 7 包含以下复用功能引脚。

表 4-8. 端口 7 复用功能引脚

引脚编号	引脚名称	复用功能	I/O	PULL [※]	备注	模块类型
64	P70	ANI0	输入	否	-	A-A
63	P71	ANI1	输入			A-A
62	P72	ANI2	输入			A-A
61	P73	ANI3	输入			A-A
60	P74	ANI4	输入			A-A
59	P75	ANI5	输入			A-A
58	P76	ANI6	输入			A-A
57	P77	ANI7	输入			A-A

注 软件上拉功能

(1) 端口 7 寄存器 (P7)

复位后: 未定义		R	地址: FFFFF40EH					
	7	6	5	4	3	2	1	0
P7	P77	P76	P75	P74	P73	P72	P71	P70
P7n	读取输入数据 (n = 0 ~ 7)							
0	输入低电平							
1	输入高电平							

4.3.6 端口 9

端口 9 可以以位为单位控制 I/O 设置的 9 位 I/O 端口。

端口 9 包含以下复用功能引脚。

表 4-9. 端口 9 用功能引脚

引脚编号	引脚名称	复用功能	I/O	PULL [※]	备注	模块类型
36	P90	TXD1/KR6	I/O	是	-	Ex0-SUT
37	P91	RXD1/KR7	输入			Ex1-SUHT
38	P96	TI51/TO51	I/O			Ex0-SUT
39	P97	SI01	输入			Ex1-SUL
40	P98	SO01	输出		可指定 N 沟道开漏输出。	Ex0-UF
41	P99	$\overline{\text{SCK01}}$	I/O		Ex2-SUFL	
42	P913	INTP4	输入		模拟噪声清除	Ex1-SUILZ
43	P914	INTP5	输入			Ex1-SUILZ
44	P915	INTP6	输入			Ex1-SUILZ

注 软件上拉功能

注意事项 P97, P99 和 P913~P915 在复用功能下具有滞后性, 在端口模式下则无。

(1) 端口 9 寄存器 (P9)

复位后: 00H (输出锁存) R/W 地址: P9 FFFF412H,
P9L FFFF412H, P9H FFFF413H

	15	14	13	12	11	10	9	8
P9 (P9H ^注)	P915	P914	P913	0	0	0	P99	P98
	7	6	5	4	3	2	1	0
(P9L)	P97	P96	0	0	0	0	P91	P90
P9n	输出数据的控制 (输出模式) (n = 0, 1, 6 ~ 9, 13 ~ 15)							
0	输出0							
1	输出1							

注 要以 8 位或 1 位读写 P9 寄存器的位 8~15，需要将它们指定为 P9H 寄存器的位 0~7。

备注 P9 寄存器支持 16 位读写方式。
此外，当使用 P9 的高 8 位作为 P9H 寄存器，
低 8 位作为 P9L 寄存器时，P9 寄存器支持 8 位或 1 位读写方式。

(2) 端口 9 模式寄存器 (PM9)

复位后: FFFFH R/W 地址: PM9 FFFF432H,
PM9L FFFF432H, PM9H FFFF433H

	15	14	13	12	11	10	9	8
PM9 (PM9H ^注)	PM915	PM914	PM913	1	1	1	PM99	PM98
	7	6	5	4	3	2	1	0
(PM9L)	PM97	PM96	1	1	1	1	PM91	PM90
PM9n	I/O 模式的控制 (n = 0, 1, 6 ~ 9, 13 ~ 15)							
0	输出模式							
1	输入模式							

注 要以 8 位或 1 位读写 PM9 寄存器的位 8~15，需要将它们指定为 PM9H 寄存器的位 0~7。

备注 PM9 寄存器支持 16 位读写方式。
此外，当使用 PM9 的高 8 位作为 PM9H 寄存器，低 8 位作为 PM9L 寄存器时，PM9 寄存器支持 8 位或 1 位读写方式。

(3) 端口 9 模式控制寄存器 (PMC9)

复位后: 0000H		R/W	地址:		PMC9 FFFF452H, PMC9L FFFF452H, PMC9H FFFF453H							
PMC9 (PMC9H ^注)	15	14	13	12	11	10	9	8				
	PMC915	PMC914	PMC913	0	0	0	PMC99	PMC98				
(PMC9L)	7	6	5	4	3	2	1	0				
	PMC97	PMC96	0	0	0	0	PMC91	PMC90				
PMC915	P915 引脚操作模式的规格											
0	I/O 端口											
1	INTP6 输入											
PMC914	P914 引脚操作模式的规格											
0	I/O 端口											
1	INTP5 输入											
PMC913	P913 引脚操作模式的规格											
0	I/O 端口											
1	INTP4 输入											
PMC99	P99 引脚操作模式的规格											
0	I/O 端口											
1	SCK01 I/O											
PMC98	P98 引脚操作模式的规格											
0	I/O 端口											
1	SO01 输出											
PMC97	P97 引脚操作模式的规格											
0	I/O 端口											
1	SI01 输入											
PMC96	P96 引脚操作模式的规格											
0	I/O 端口/TI51 输入											
1	TO51 输出											
PMC91	P91 引脚操作模式的规格											
0	I/O 端口/KR7 输入											
1	RXD1 输入											
PMC90	P90 引脚操作模式的规格											
0	I/O 端口port/KR6 输入											
1	TXD1 输出											

注 要以 8 位或 1 位读写 PMC9 寄存器的位 8~15, 需要将它们指定为 PMC9H 寄存器的位 0~7。

备注 PMC9 寄存器支持 16 位读写方式。
此外, 当使用 PMC9 的高 8 位作为 PMC9H 寄存器, 低 8 位作为 PMC9L 寄存器时, PMC9 寄存器支持 8 位或 1 位读写方式。

(4) 端口 9 功能寄存器 H (PF9H)

复位后:00H R/W 地址:FFFFC73H

	7	6	5	4	3	2	1	0
PF9H	0	0	0	0	0	0	PF99	PF98

PF9n	正常输出/N沟道开漏输出的控制 (n = 8, 9)
0	正常输出
1	N沟道开漏输出

注意事项 当将 P98 和 P99 作为 N 沟道开漏复用功能引脚时，按以下顺序设置 N 沟道开漏。
一定要先设置端口锁存器为 1，后设置 N 沟道开漏输出。

P9n 位 = 1 → PFC9n 位 = 0/1 → PF9n 位 = 1 → PMC9n 位 = 1

(5) 端口 9 功能控制寄存器 (PFC9)

注意事项 在 PFC9n 位保持初始值 0 时, 通过 PMC9.PMC9n 位指定端口 9 的复用功能时, 输出变得不确定。因此, 要设置端口 9 的复用功能 2 时, 首先设置 PFC9n 位为 1, 再设置 PMC9n 位为 1 (n = 0, 1, 6 ~ 9, 13 ~ 15)。

复位后 0000H		R/W	地址:		PFC9 FFFFF472H, PFC9L FFFFF472H, PFC9H FFFFF473H							
PFC9 (PFC9H ^注)	15	14	13	12	11	10	9	8				
	PFC915	PFC914	PFC913	0	0	0	PFC99	PFC98				
(PFC9L)	7	6	5	4	3	2	1	0				
	PFC97	PFC96	0	0	0	0	PFC91	PFC90				
PFC915	P915复用功能引脚的规格											
1	INTP6 输入											
PFC914	P914 复用功能引脚的规格											
1	INTP5 输入											
PFC913	P913 复用功能引脚的规格											
1	INTP4 输入											
PFC99	P99复用功能引脚的规格											
1	SCK01 I/O											
PFC98	P98 复用功能引脚的规格											
1	SO01 输出											
PFC97	P97 复用功能引脚的规格											
1	SI01 输入											
PFC96	P96 复用功能引脚的规格											
1	TO51 输出											
PFC91	P91复用功能引脚的规格											
1	RXD1 输入											
PFC90	P90复用功能引脚的规格											
1	TXD1 输出											
注	要以 8 位或 1 位读写 PFC9 寄存器的位 8~15, 需要将它们指定为 PFC9H 寄存器的位 0~7。											
备注	PFC9 寄存器支持 16 位读写方式。 此外, 当使用 PFC9 的高 8 位作为 PFC9H 寄存器, 低 8 位作为 PFC9L 寄存器时, PFC9 寄存器支持 8 位或 1 位读写方式。											

(6) 上拉电阻选项寄存器 9 (PU9)

复位后:0000H R/W 地址: PU9 FFFFFFFC52H,
PU9L FFFFFFFC52H, PU9H FFFFFFFC53H

	15	14	13	12	11	10	9	8
PU9 (PU9H ^注)	PU915	PU914	PU913	0	0	0	PU99	PU98
	7	6	5	4	3	2	1	0
(PU9L)	PU97	PU96	0	0	0	0	PU91	PU90
PU9n	片上上拉电阻连接的控制 (n = 0, 1, 6 ~ 9, 13 ~ 15)							
0	不接							
1	连接							

注 以 8 位或 1 位读写 PU9 寄存器的位 8~15，需要将它们指定为 PU9H 寄存器的位 0~7。

备注 PU9 该寄存器支持 16 位读写方式。

此外，当使用 PU9 的高 8 位作为 PU9H 寄存器，低 8 位作为 PU9L 寄存器时，PU9 寄存器支持 8 位或 1 位读写方式。

4.3.7 端口 CM

CM 端口为可以以位为单位控制 I/O 设置 2 位 I/O 端口。
CM 端口包括以下复用功能引脚。

表 4-10. 端口 CM 复用功能引脚

引脚编号	引脚名称	复用功能	I/O	PULL ^注	备注	模块类型
45	PCM0	-	-	是	-	C-U
46	PCM1	CLKOUT	输出			D0-U

注 软件上拉功能

(1) 端口 CM 寄存器 (PCM)

复位后: 00H (输出锁存器) R/W 地址: FFFF00CH								
	7	6	5	4	3	2	1	0
PCM	0	0	0	0	0	0	PCM1	PCM0
	PCMn 输出数据的控制 (输出模式) (n = 0, 1)							
	0	输出0						
	1	输出1						

(2) 端口 CM 模式寄存器 (PMCM)

复位后: FFH R/W 地址: FFFF02CH								
	7	6	5	4	3	2	1	0
PMCM	1	1	1	1	1	1	PMCM1	PMCM0
	PMCMn I/O 模式的控制 (n = 0, 1)							
	0	输出模式						
	1	输入模式						

(3) 端口 CM 模式控制寄存器 (PMCCM)

复位后: 00H R/W 地址: FFFF04CH								
	7	6	5	4	3	2	1	0
PMCCM	0	0	0	0	0	0	PMCCM1	0
	PMCCM1 PCM1 引脚操作模式的规格							
	0	I/O 端口						
	1	CLKOUT输出						

(4) 上拉电阻选项寄存器 CM (PUCM)

复位后: 00H R/W 地址: FFFFFFF4CH

	7	6	5	4	3	2	1	0
PUCM	0	0	0	0	0	0	PUCM1	PUCM0

PUCMn	片上上拉电阻连接的控制 (n = 0, 1)
0	不接
1	连接

4.3.8 端口 DL

DL 端口为可以以位为单位控制 I/O 设置 8 位 I/O 端口。

DL 端口包括以下复用功能引脚。

表 4-11. 端口 DL 复用引脚功能

引脚编号	引脚名称	复用功能引脚名称	I/O	PULL ^注	备注	模块类型
47	PDL0	-	-	是	-	C-U
48	PDL1	-	-			C-U
49	PDL2	-	-			C-U
50	PDL3	-	-			C-U
51	PDL4	-	-			C-U
52	PDL5	-	-			C-U
53	PDL6	-	-			C-U
54	PDL7	-	-			C-U

注 软件上拉功能

(1) 端口 DL 寄存器 (PDL)

复位后: 00H (输出锁存器) R/W 地址: FFFF004H

	7	6	5	4	3	2	1	0
PDL	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0
PDLn	输出数据的控制 (输出模式) (n=0~7)							
0	输出0							
1	输出1							

(2) 端口 DL 模式寄存器 (PMDL)

复位后: FFH R/W 地址: FFFF024H

	7	6	5	4	3	2	1	0
PMDL	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0
PMDLn	I/O 模式的控制 (n=0~7)							
0	输出模式							
1	输入模式							

(3) 上拉电阻选项寄存器 DL (PUDL)

复位后: 00H R/W 地址: FFFFFFF44H

	7	6	5	4	3	2	1	0
PUDL	PUDL7	PUDL6	PUDL5	PUDL4	PUDL3	PUDL2	PUDL1	PUDL0
PUDLn	片上上拉电阻连接的控制 (n=0~7)							
0	不接							
1	连接							

4.4 框图

图 4-2. A-A 类型的框图

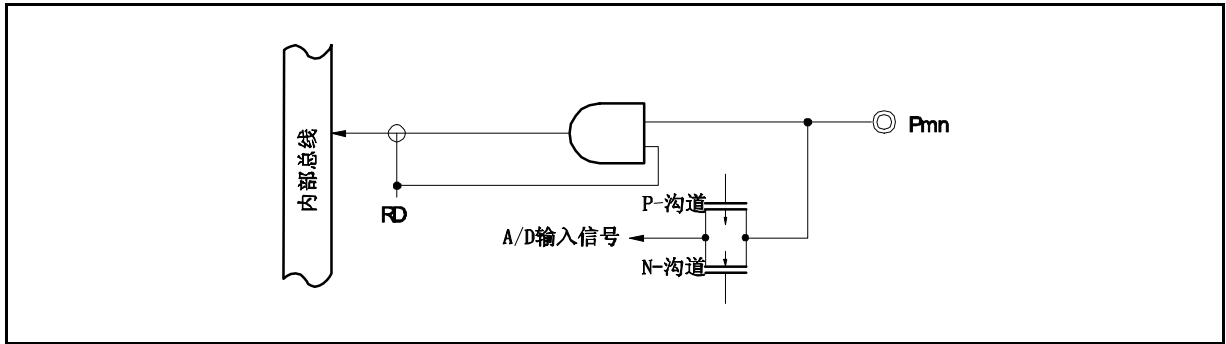


图 4-3. C-U 类型的框图

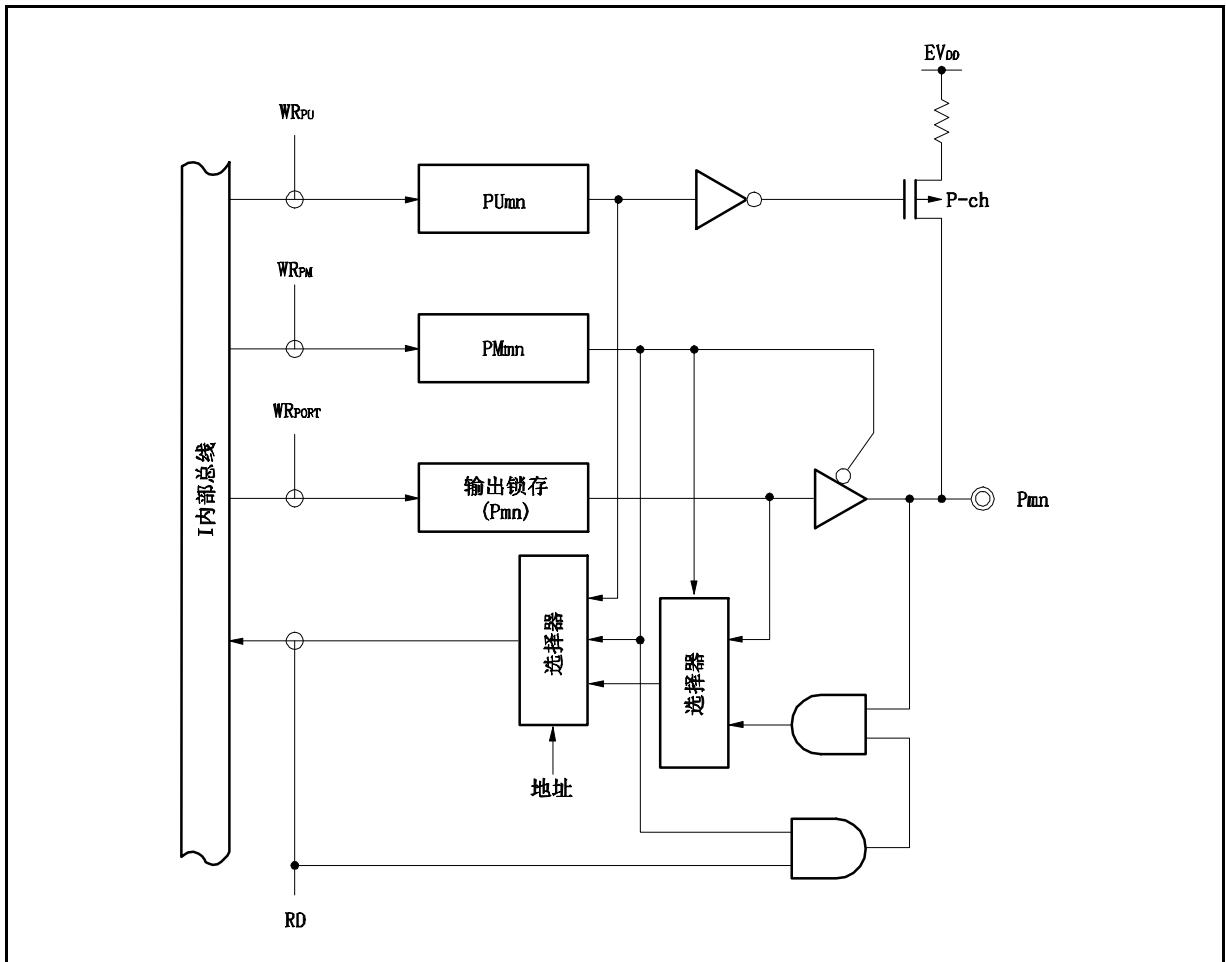


图 4-4. D0-U 类型的框图

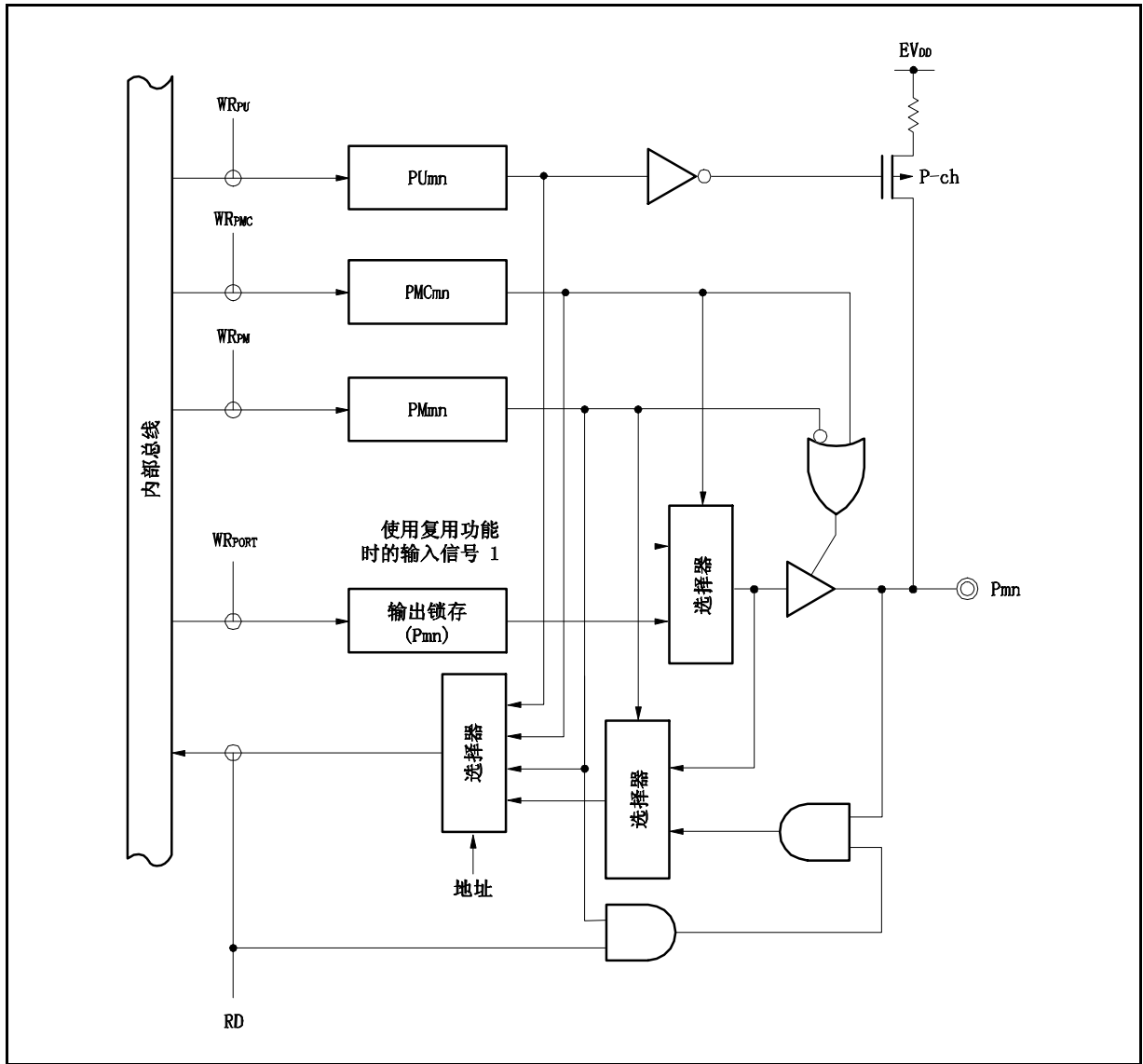
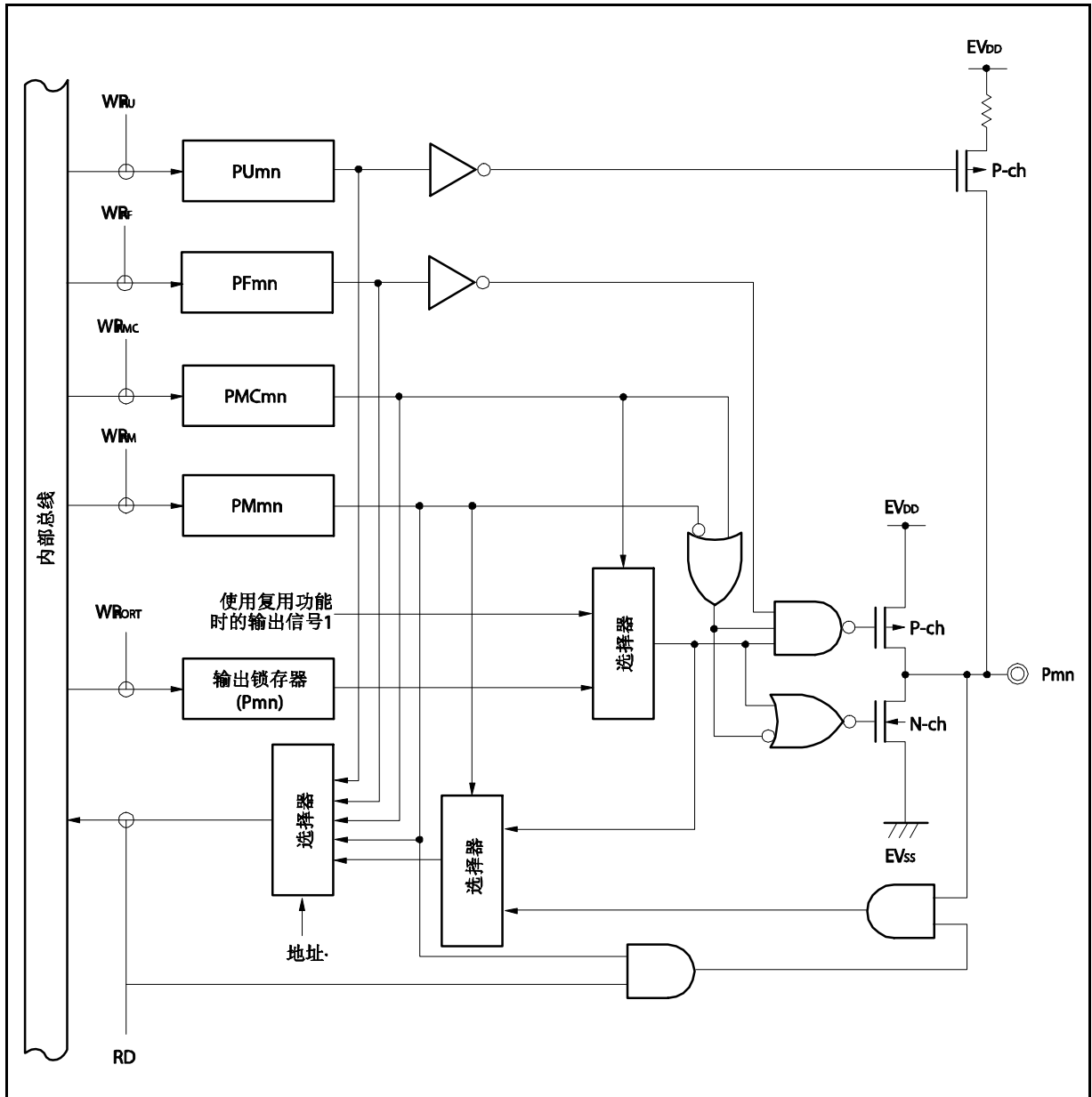
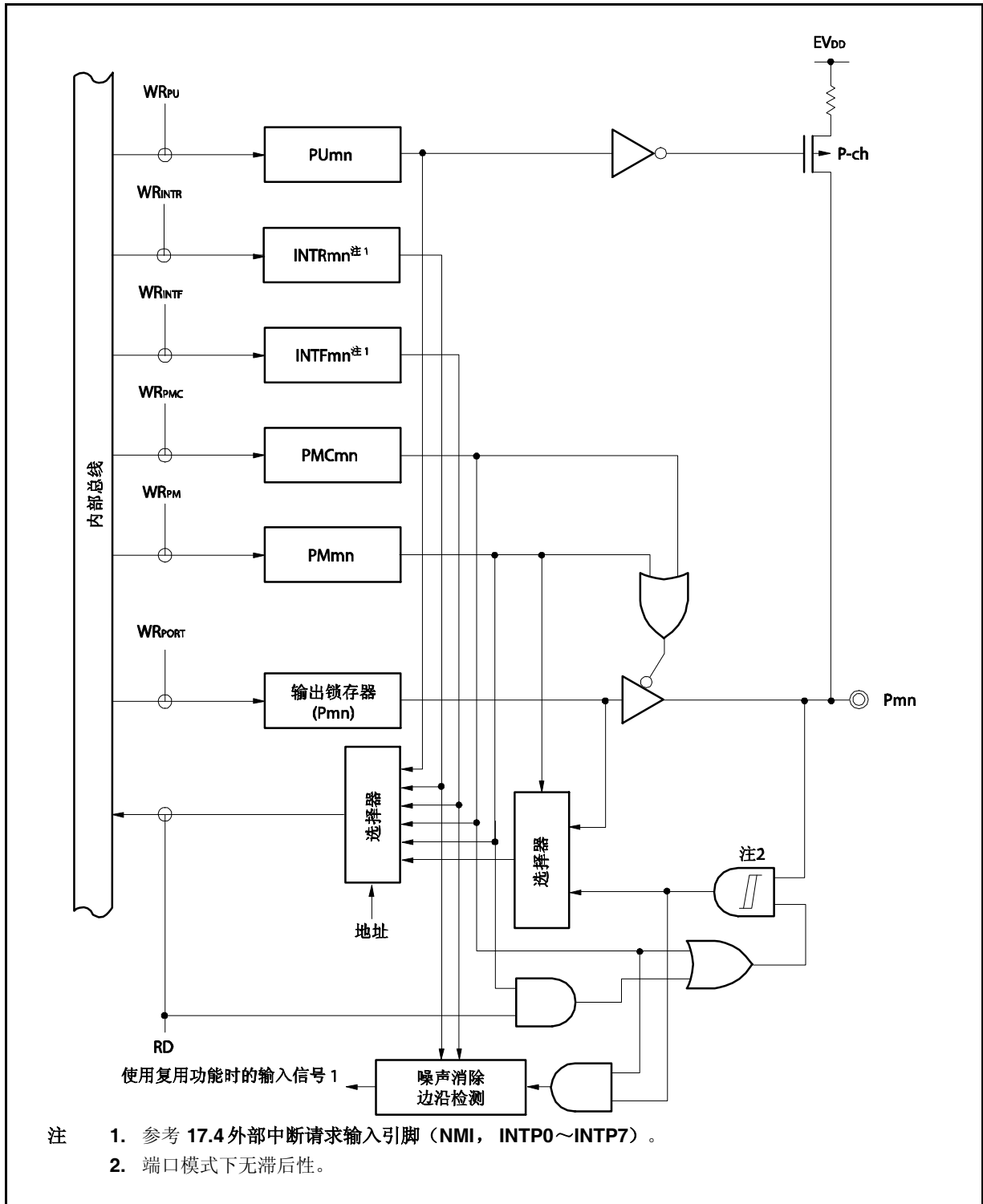


图 4-5. D0-UF 类型的框图



★

图 4-6. D1-SUIL 类型的框图



★

图 4-7. D1-SUIHL 类型的框图

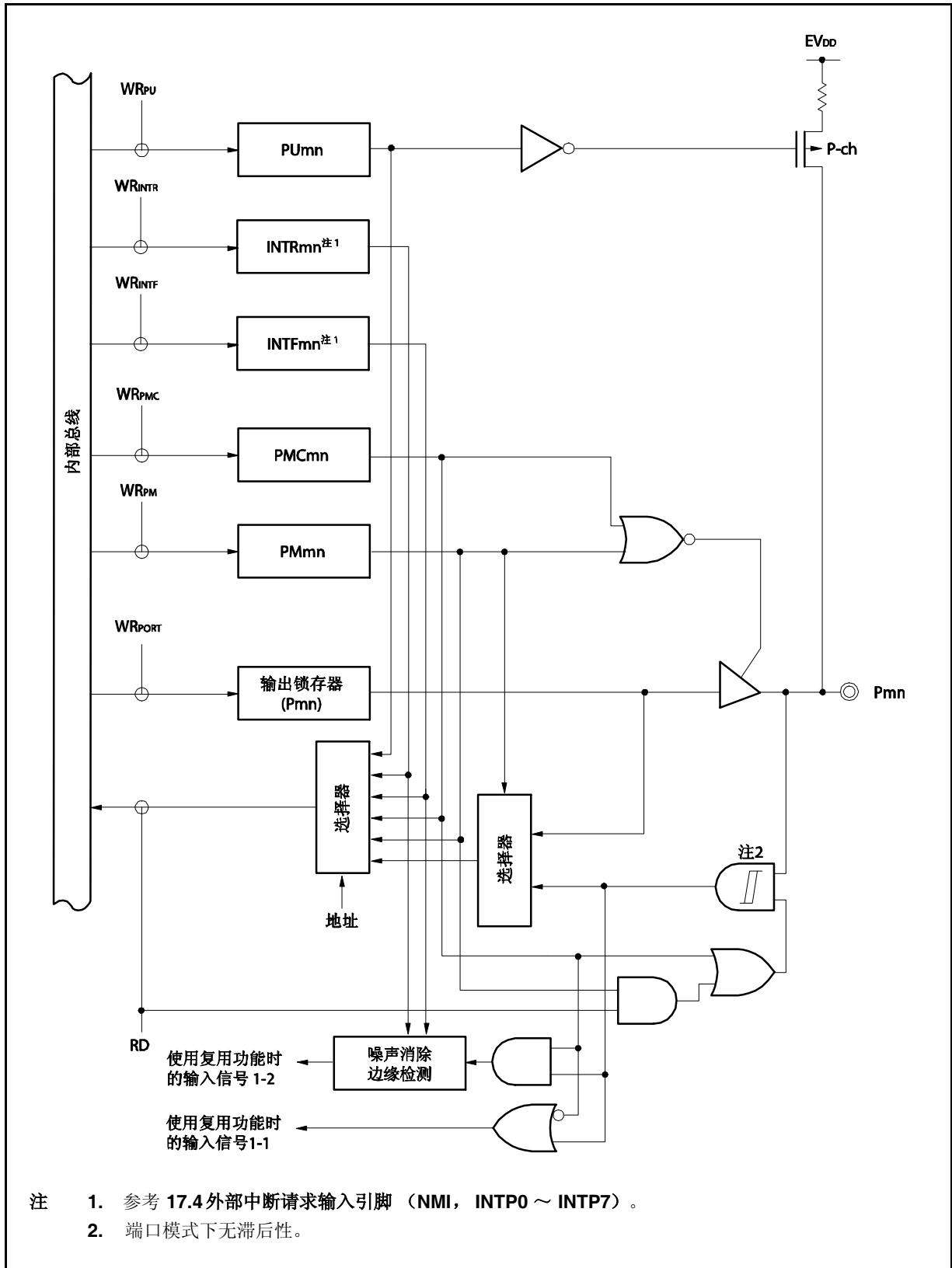


图 4-8. D1-SUL 类型的框图

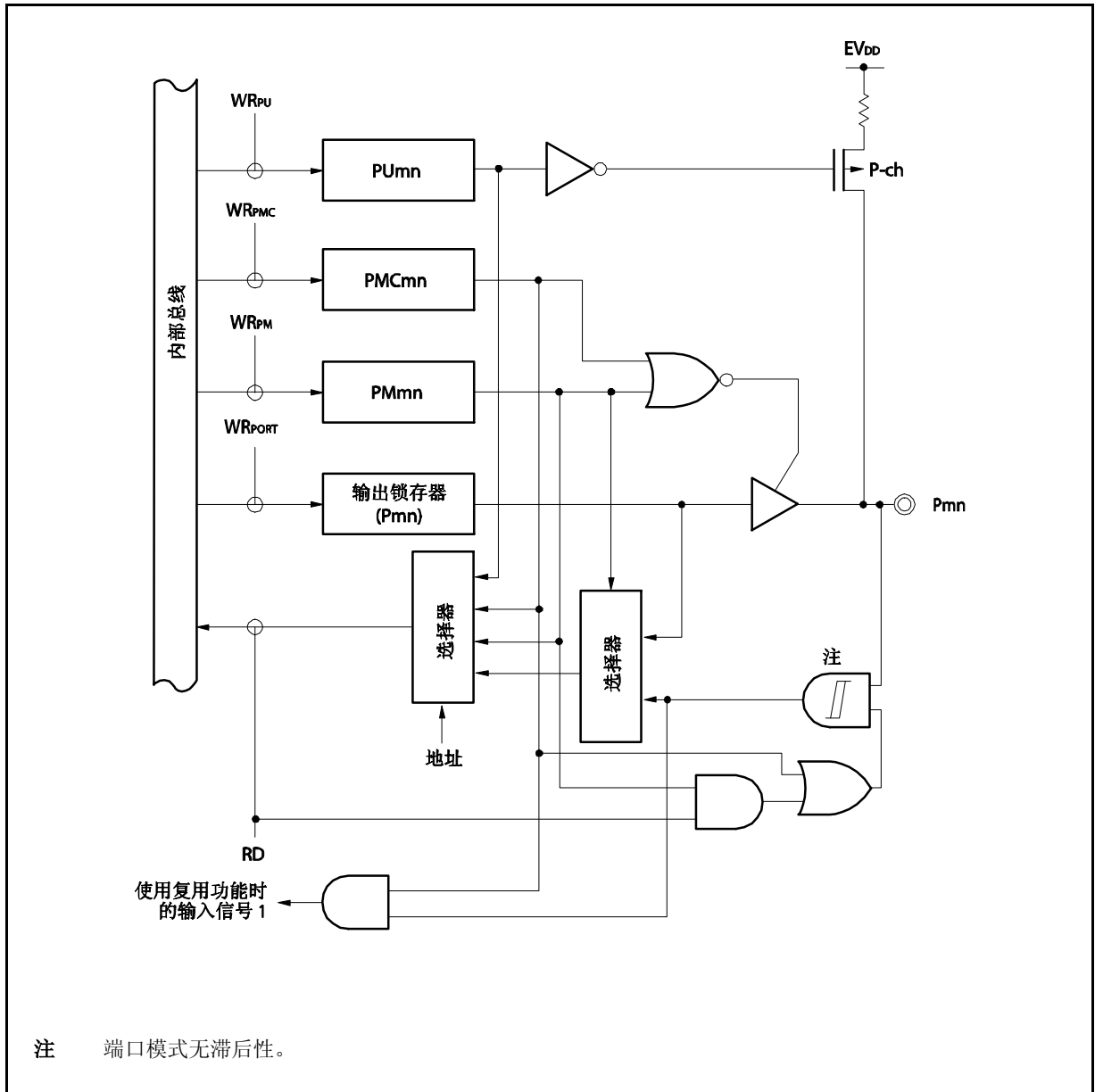


图 4-9. D2-SNFH 类型的框图

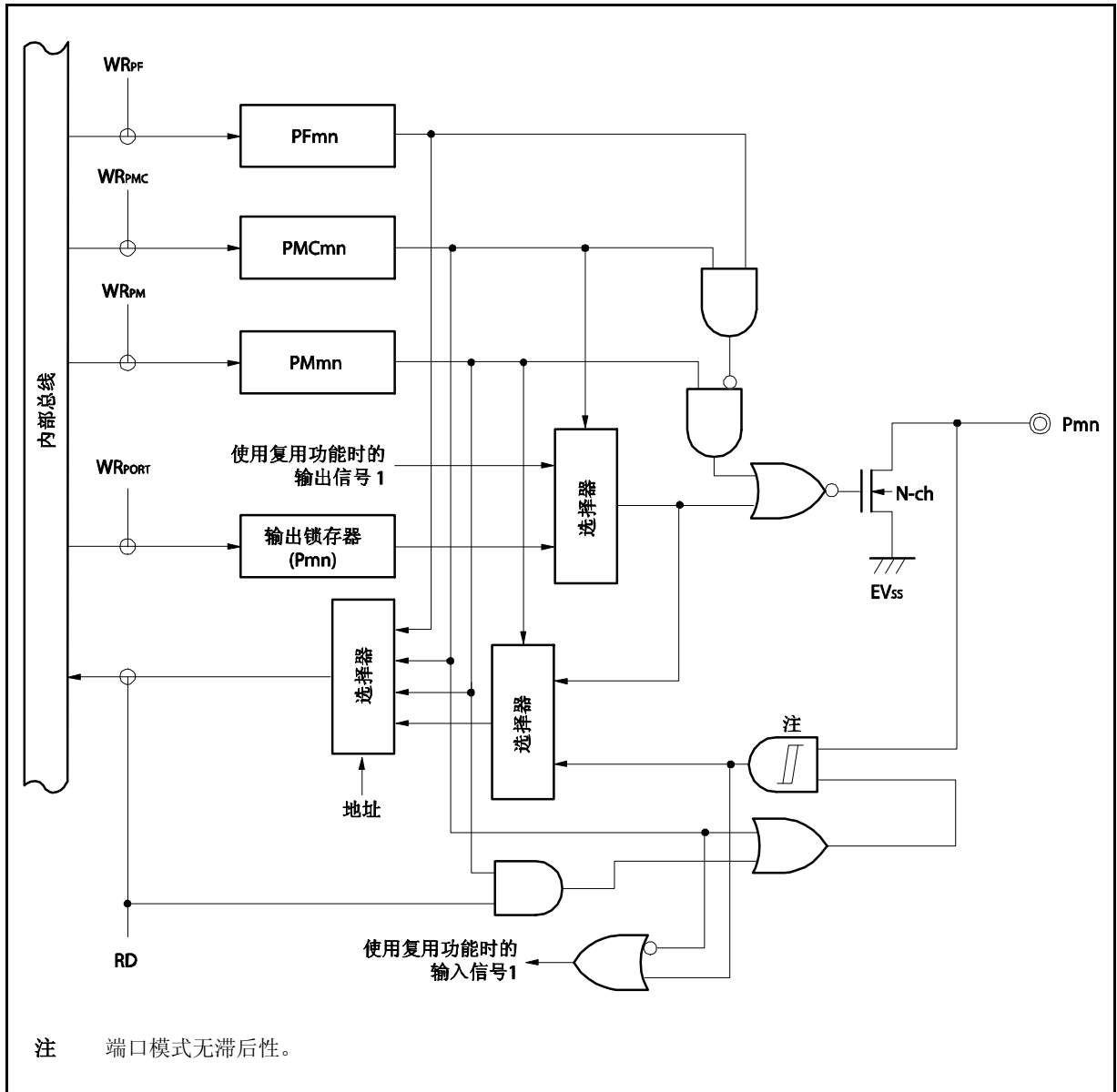


图 4-10. D2-SUFL 类型的框图

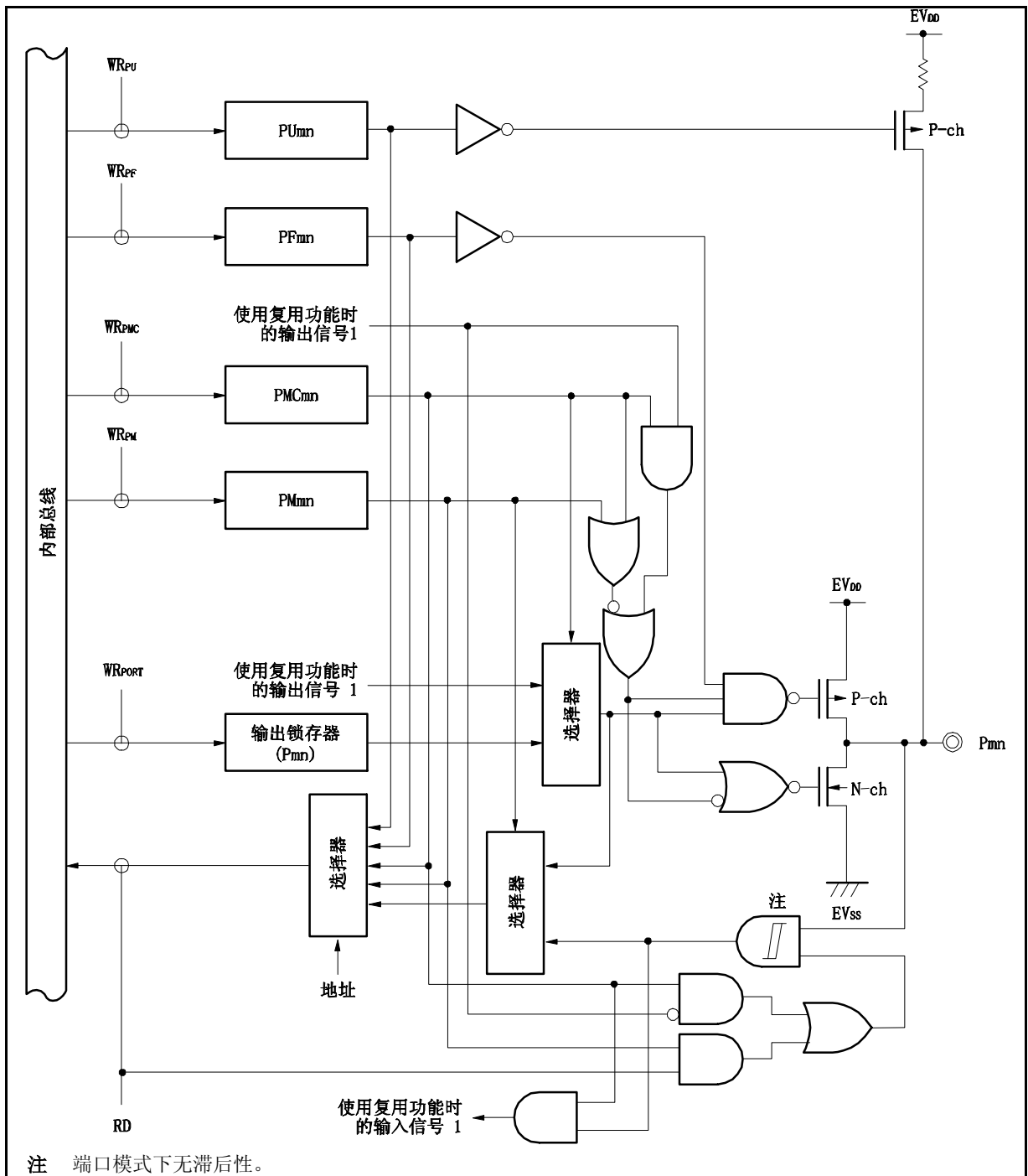
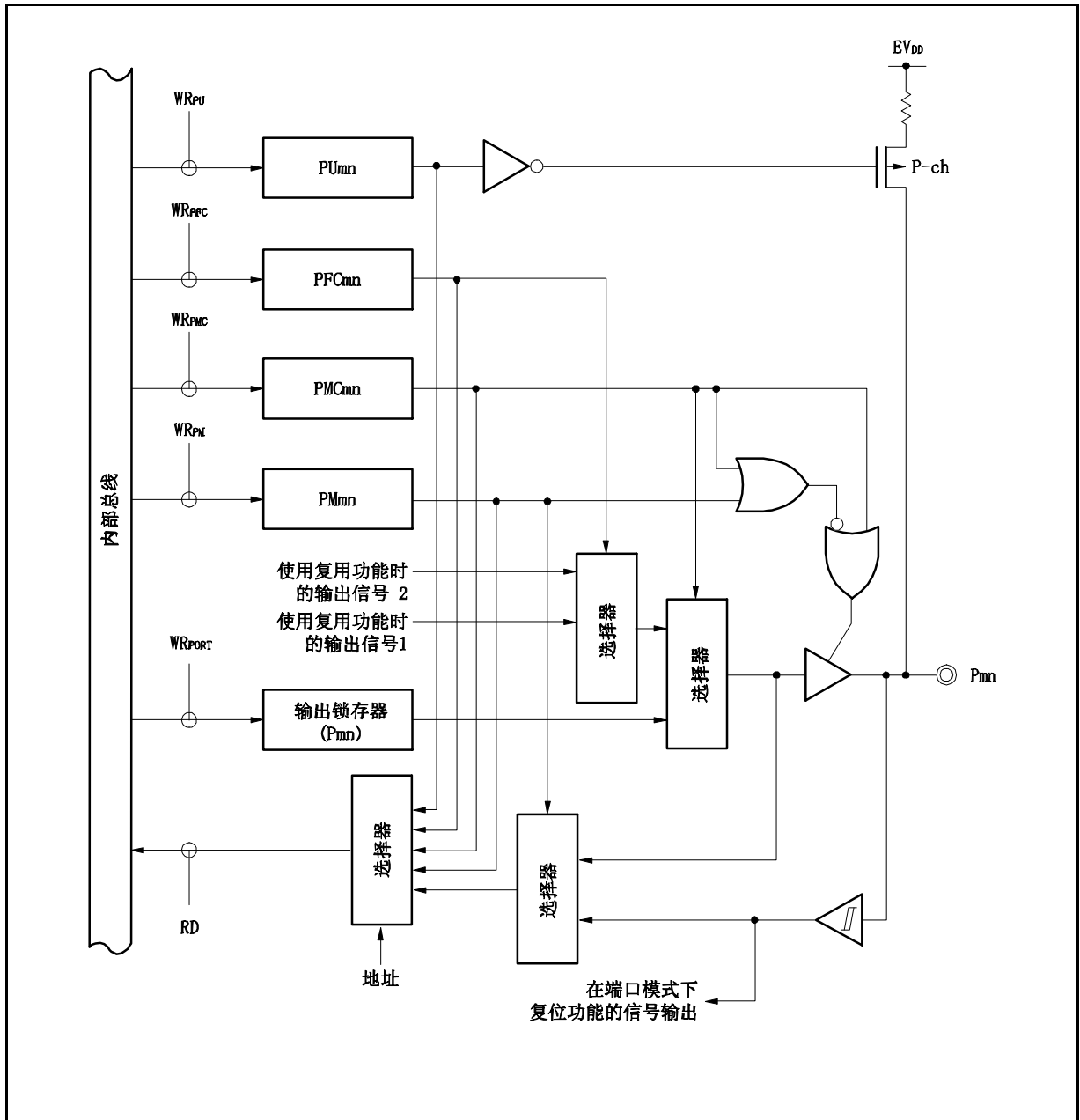
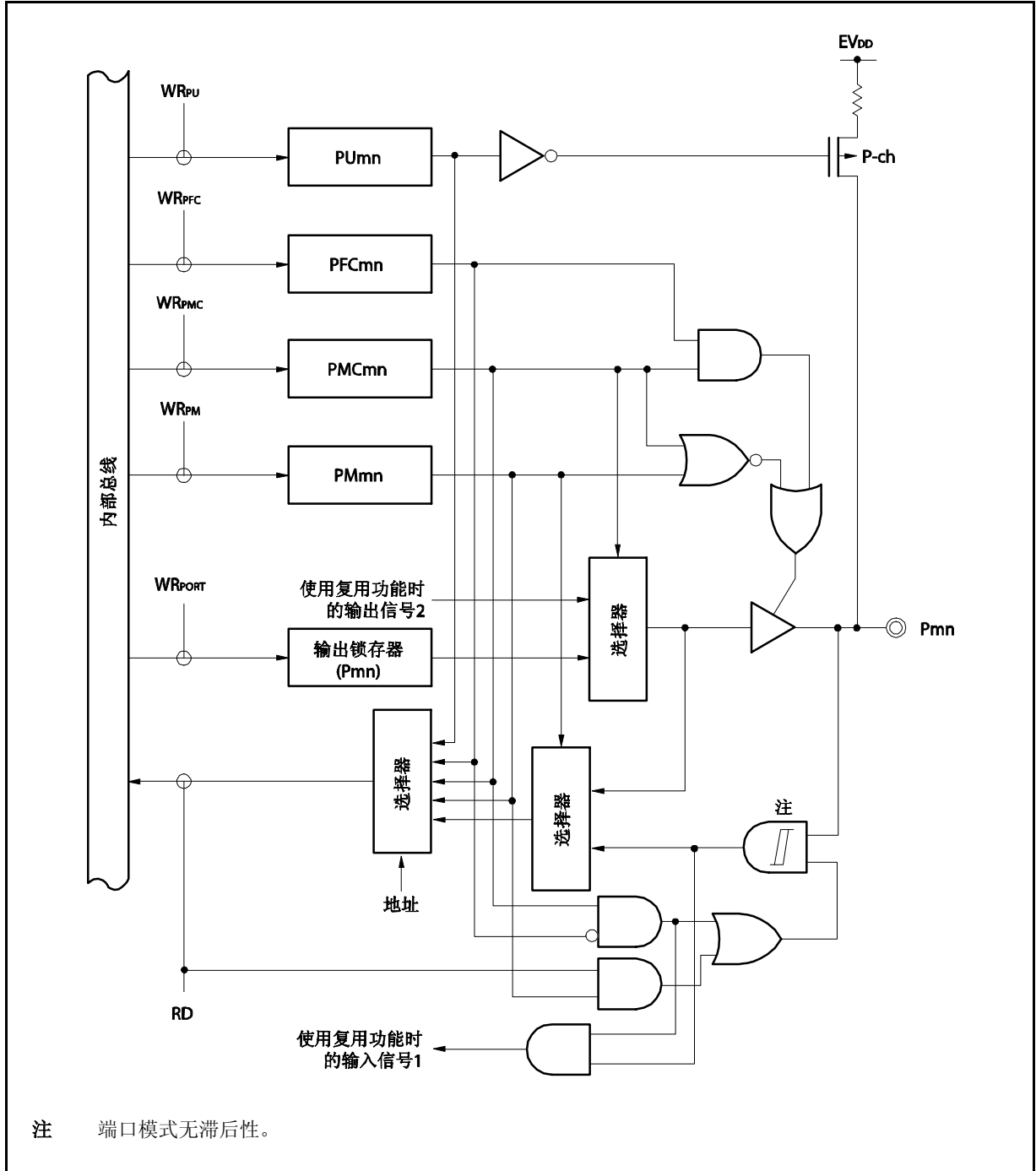


图 4-11. E00-SUT 类型的框图



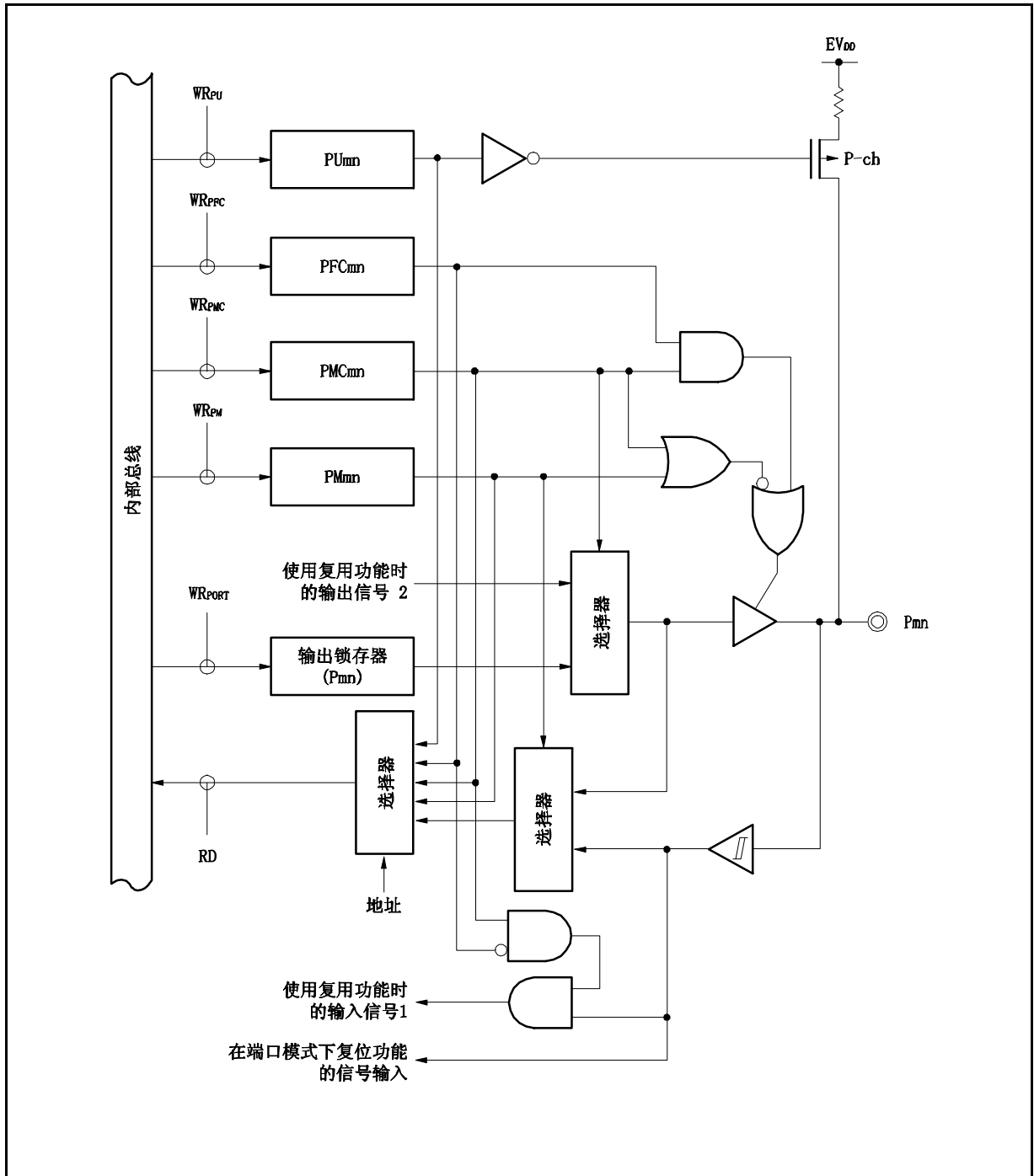
★

图 4-12. E10-SUL 类型的框图



★

图 4-13. E10-SULT 类型的框图



★

图 4-14. Ex0-SUT 类型的框图

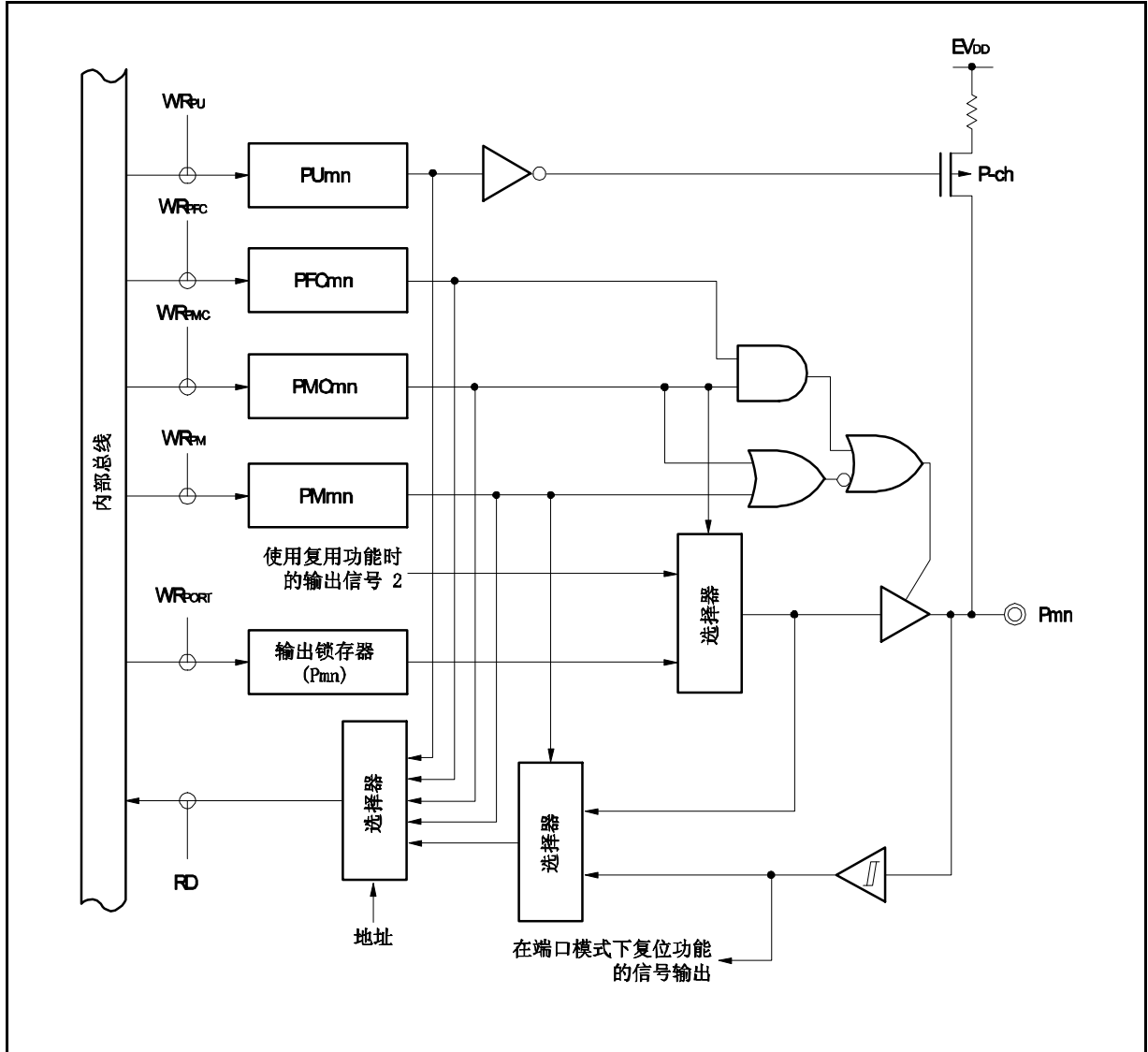


图 4-15. Ex0-UF 类型的框图

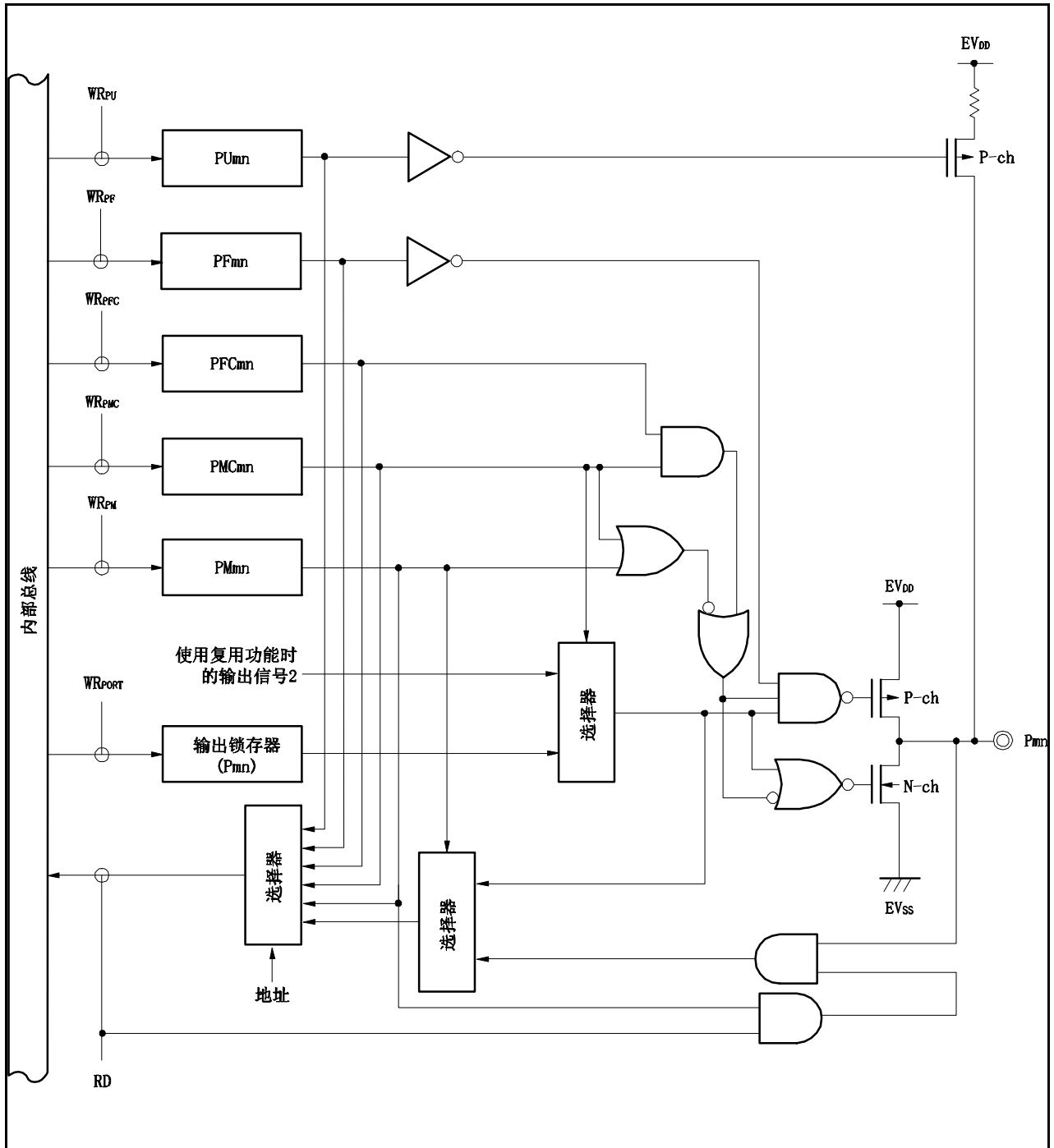
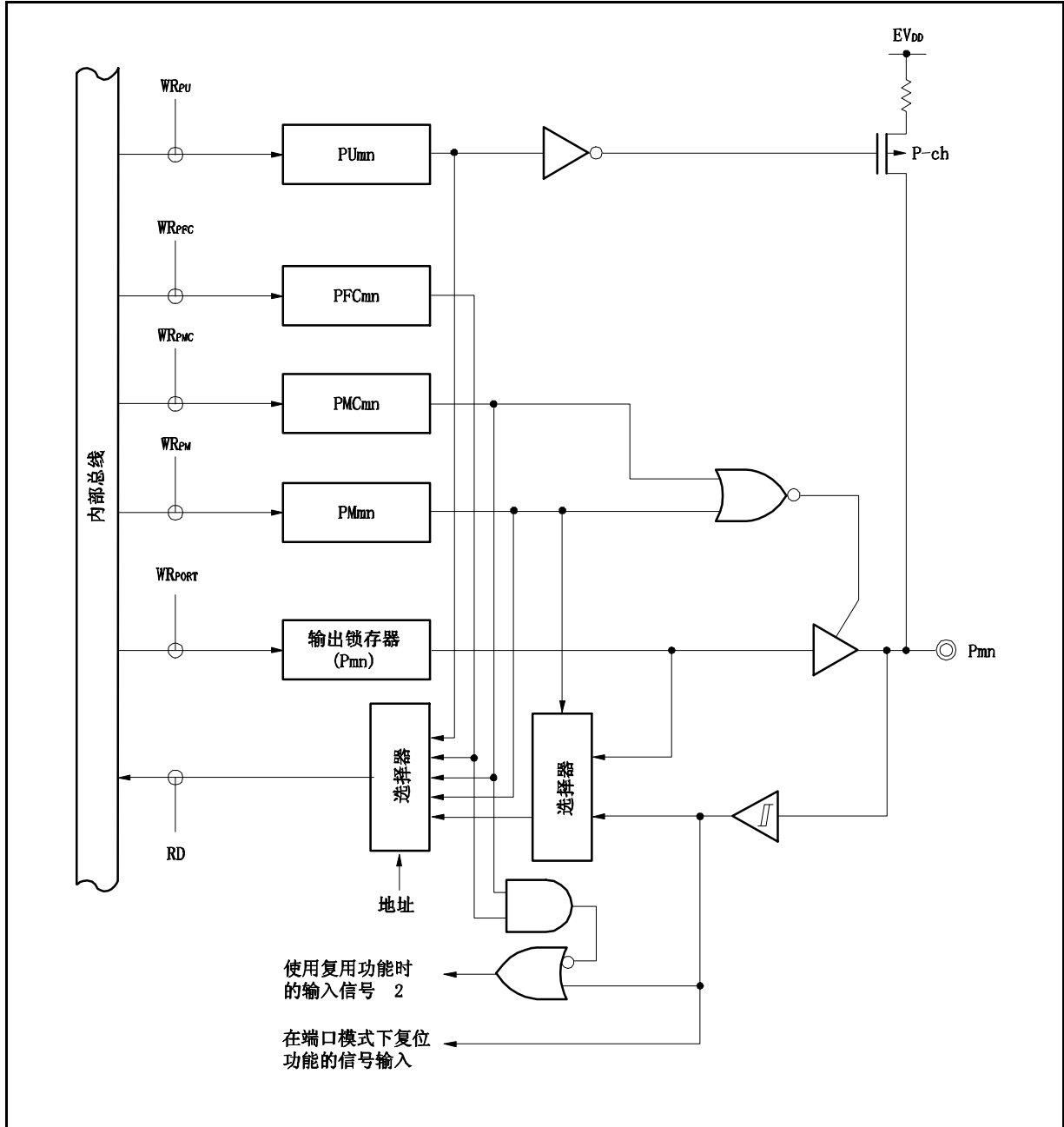
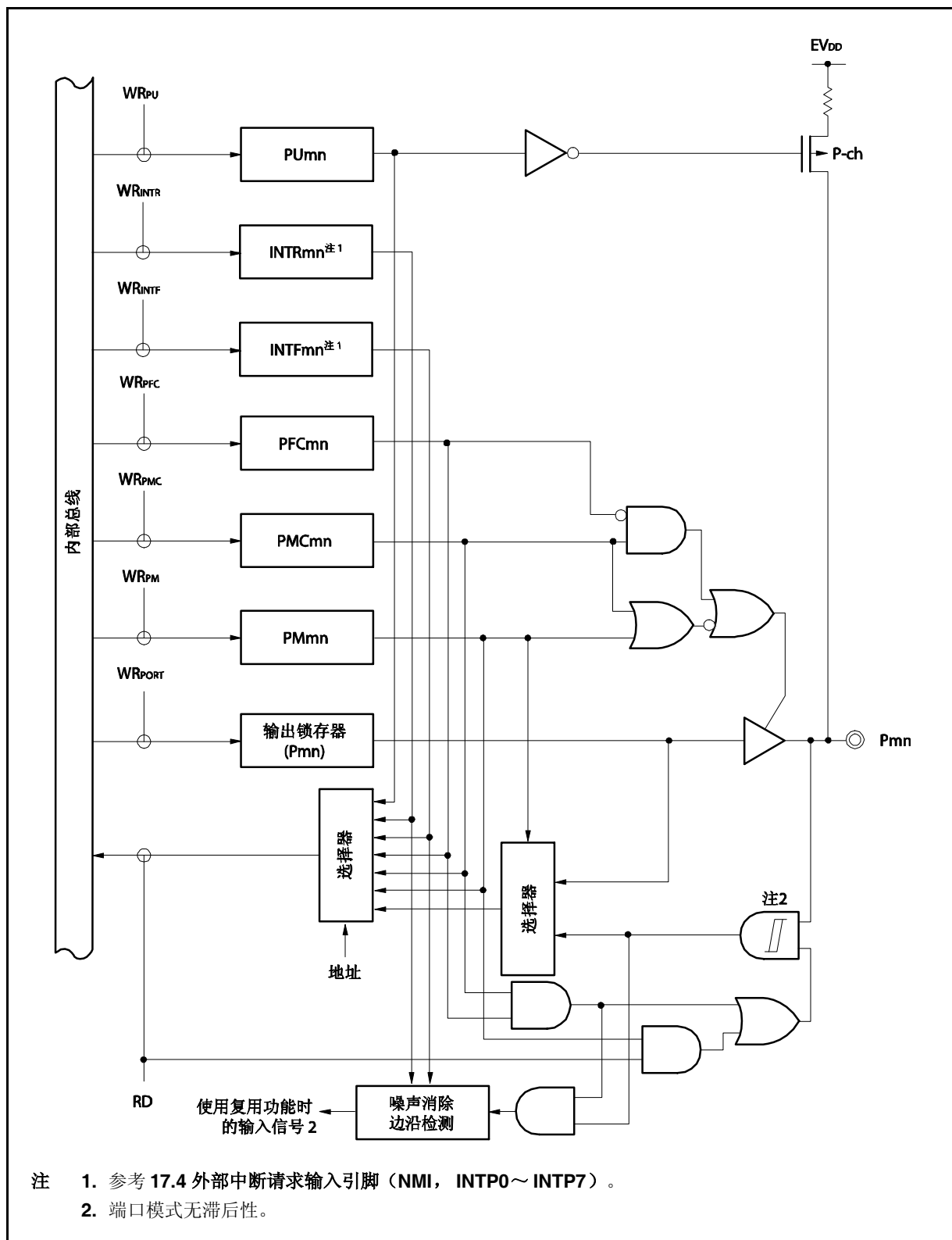


图 4-16. Ex1-SUHT 类型的框图



★

图 4-17. Ex1-SUIL 类型的框图



★

图 4-18. Ex1-SUL 类型的框图

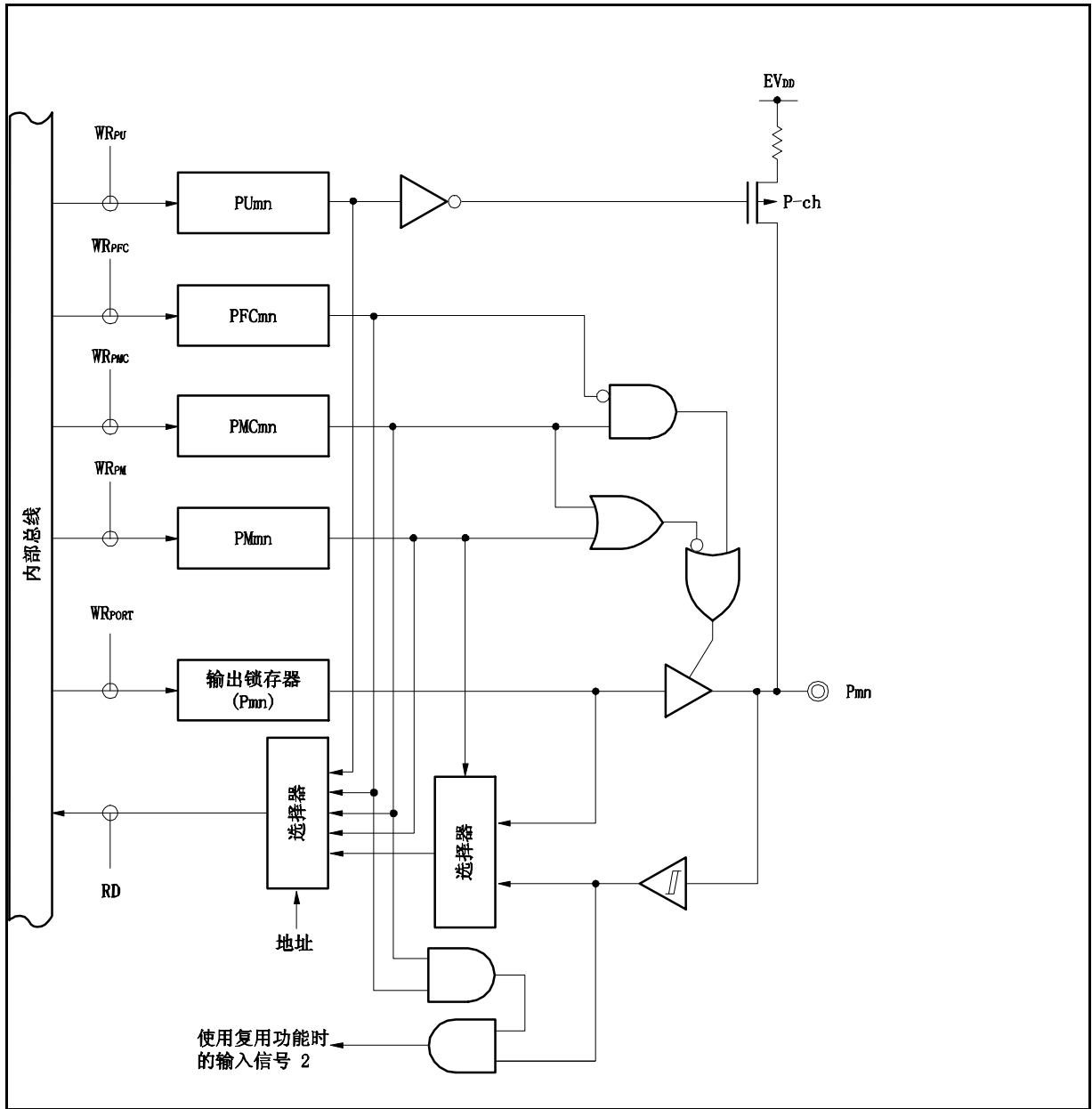


图 4-19. Ex2-SUFL 类型的框图

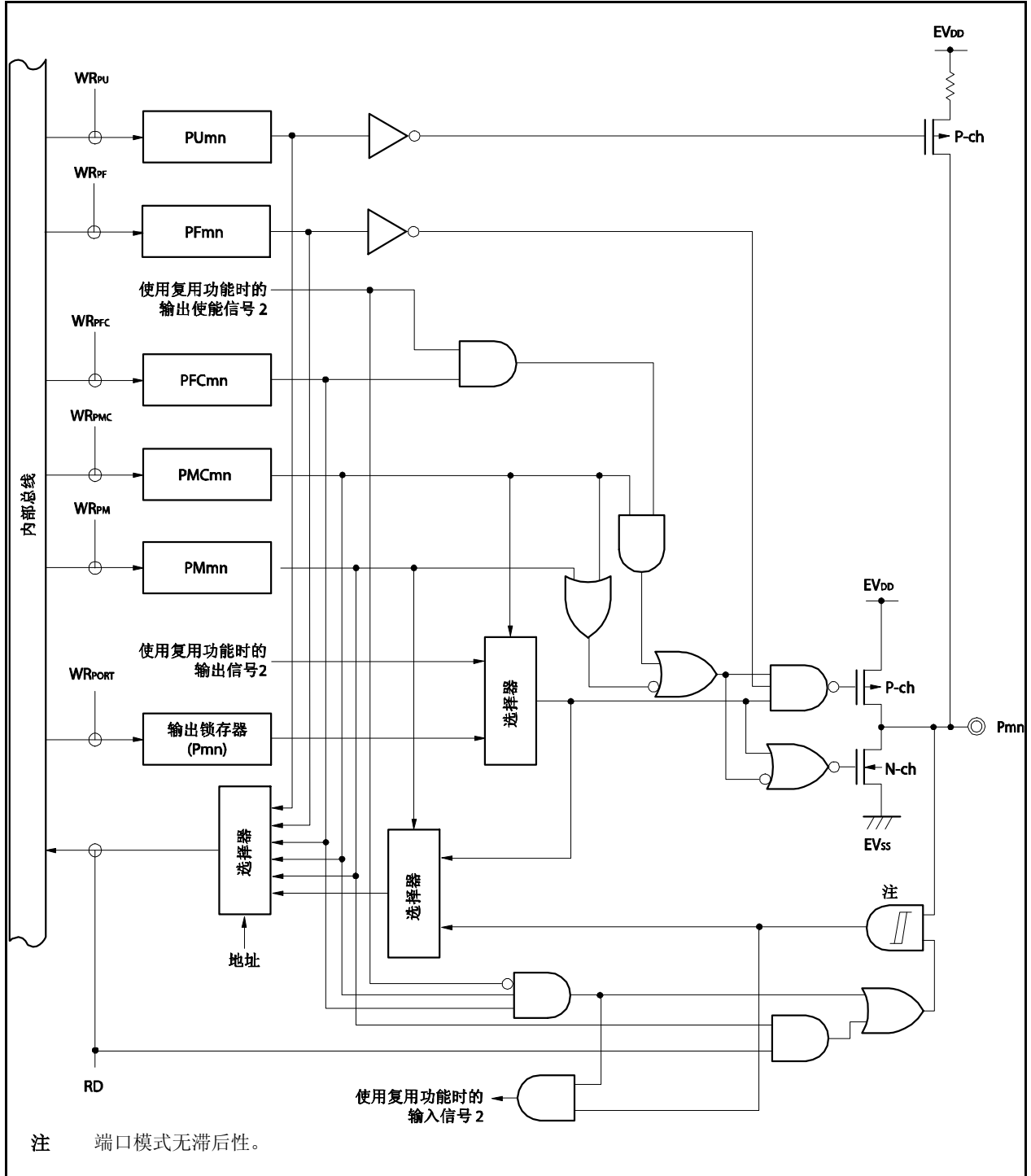
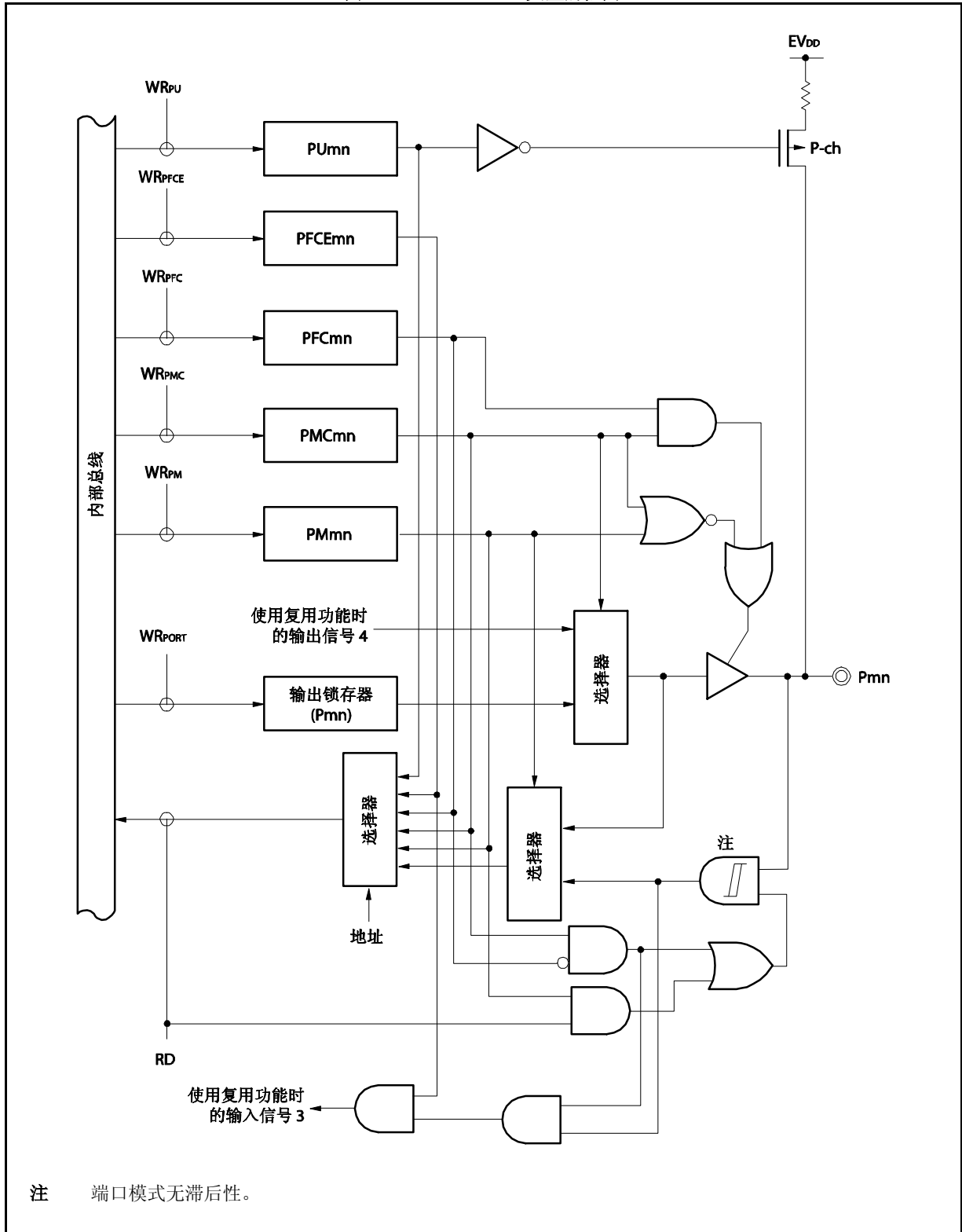


图 4-20. Gxx10-SUL 类型的框图



注 端口模式无滞后性。

4.5 当使用复用功能时的端口寄存器设置

表 4-12 显示每个端口用作复用功能时的端口寄存器设置。
当端口引脚用作复用功能引脚时，请参考每个引脚的说明。

表 4-12. 端口引脚用作复用功能时的设置 (1/3)

引脚名称	复用功能		Pn寄存器的Pnx位	PMn寄存器的PMnx位	PMCn寄存器的PMCnx位	PFCEn寄存器的PFCEnx位	PFCn寄存器的PFCnx位	其它位 (寄存器)
	功能名称	I/O						
P00	TOH0	输出	P00 = 不要求设置	PM00 = 不要求设置	PMC00 = 1	-	-	-
P01	TOH1	输出	P01 = 不要求设置	PM01 = 不要求设置	PMC01 = 1	-	-	-
P02	NMI	输入	P02 = 不要求设置	PM02 = 不要求设置	PMC02 = 1	-	-	-
P03	INTP0	输入	P03 = 不要求设置	PM03 = 不要求设置	PMC03 = 1	-	PFC03 = 0	-
P04	INTP1	输入	P04 = 不要求设置	PM04 = 不要求设置	PMC04 = 1	-	-	-
P05	INTP2	输入	P05 = 不要求设置	PM05 = 不要求设置	PMC05 = 1	-	-	-
P06	INTP3	输入	P06 = 不要求设置	PM06 = 不要求设置	PMC06 = 1	-	-	-
P30	TXD0	输出	P30 = 不要求设置	PM30 = 不要求设置	PMC30 = 1	-	PFC30 = 0	-
P31	RXD0	输入	P31 = 不要求设置	PM31 = 不要求设置	PMC31 = 1	-	注1, PFC31 = 0	-
	INTP7	输入	P31 = 不要求设置	PM31 = 不要求设置	PMC31 = 1	-	注1, PFC31 = 0	-
P32	ASCK0	输入	P32 = 不要求设置	PM32 = 不要求设置	PMC32 = 1	-	注2, PFC32 = 0	-
	ADTRG	输入	P32 = 不要求设置	PM32 = 不要求设置	PMC32 = 1	-	注2, PFC32 = 0	-
	TO01	输出	P32 = 不要求设置	PM32 = 不要求设置	PMC32 = 1	-	PFC32 = 1	-
P33	TIPO0	输入	P33 = 不要求设置	PM33 = 不要求设置	PMC33 = 1	PFCE33 = 1	PFC33 = 0	-
	TOPO0	输出	P33 = 不要求设置	PM33 = 不要求设置	PMC33 = 1	PFCE33 = 1	PFC33 = 1	-
P34	TIPI0	输入	P34 = 不要求设置	PM34 = 不要求设置	PMC34 = 1	PFCE34 = 1	PFC34 = 0	-
	TOPI0	输出	P34 = 不要求设置	PM34 = 不要求设置	PMC34 = 1	PFCE34 = 1	PFC34 = 1	-
P35	TI010	输入	P35 = 不要求设置	PM35 = 不要求设置	PMC35 = 1	-	PFC35 = 0	-
	TO01	输出	P35 = 不要求设置	PM35 = 不要求设置	PMC35 = 1	-	PFC35 = 1	-

注 1. INTP7 及 RXD0 引脚为复用功能引脚。当把引脚作为 RXD0 引脚时，禁止复用功能引脚 INTP7 的边沿检测 (将 INTF3.INTF31 位以及 INTR3 .INTR31 位清零)。当将引脚作为 INTP7 引脚使用时，停止 UART0 接收操作 (将 ASIM0.RXE0 位清零)。

2. ASCK0 及 ADTRG 引脚为复用功能引脚。当把引脚作为 ASCK0 引脚时，禁用复用功能引脚 ADTRG 的触发输入 (将 ADS.TRG 位清零或将 ADS.ADTMD 位置 1)。当将引脚作为 ADTRG 引脚使用时，勿将 UART0 操作时钟设置为外部输入 (将 CKSR0 .TPSR03~CKSR0 .TPSR00 位设置为除了 1011 以外的数)。

表 4-12. 端口引脚用作复用功能时的设置 (2/3)

引脚名称	复用功能		P 寄存器的Pnx位	Pm _n 寄存器的Pm _{nx} 位	PMC _n 寄存器的PMC _{nx} 位	PFC _n 寄存器的PFC _{nx} 位	其它位 (寄存器)
	功能名称	I/O					
P38	SDA0	I/O	P38 = 不要求设置	PM38 = 不要求设置	PMC38 = 1	-	PF38 (PF3H) = 1
P39	SCL0	I/O	P39 = 不要求设置	PM39 = 不要求设置	PMC39 = 1	-	PF39 (PF3H) = 1
P40	SI00	输入	P40 = 不要求设置	PM40 = 不要求设置	PMC40 = 1	-	-
P41	SO00	输出	P41 = 不要求设置	PM41 = 不要求设置	PMC41 = 1	-	PF41 (PF4) = 无关
P42	SCR00	I/O	P42 = 不要求设置	PM42 = 不要求设置	PMC42 = 1	-	PF42 (PF4) = 无关
P50	TI011	输入	P50 = 不要求设置	PM50 = 不要求设置	PMC50 = 1	PFC50 = 0	-
	RTP00	输出	P50 = 不要求设置	PM50 = 不要求设置	PMC50 = 1	PFC50 = 1	-
	KR0	输入	P50 = 不要求设置	PM50 = 1	PMC50 = 0	PFC50 = 不要求设置	KRM0 (KRM) = 1
P51	TI150	输入	P51 = 不要求设置	PM51 = 不要求设置	PMC51 = 1	PFC51 = 0	-
	RTP01	输出	P51 = 不要求设置	PM51 = 不要求设置	PMC51 = 1	PFC51 = 1	-
	KR1	输入	P51 = 不要求设置	PM51 = 1	PMC51 = 0	PFC51 = 不要求设置	KRM1 (KRM) = 1
P52	TO50	输出	P52 = 不要求设置	PM52 = 不要求设置	PMC52 = 1	PFC52 = 0	-
	RTP02	输出	P52 = 不要求设置	PM52 = 不要求设置	PMC52 = 1	PFC52 = 1	-
	KR2	输入	P52 = 不要求设置	PM52 = 1	PMC52 = 0	PFC52 = 不要求设置	KRM2 (KRM) = 1
P53	RTP03	输出	P53 = 不要求设置	PM53 = 不要求设置	PMC53 = 1	PFC53 = 1	-
	KR3	输入	P53 = 不要求设置	PM53 = 1	PMC53 = 0	PFC53 = 不要求设置	KRM3 (KRM) = 1
P54	RTP04	输出	P54 = 不要求设置	PM54 = 不要求设置	PMC54 = 1	PFC54 = 1	-
	KR4	输入	P54 = 不要求设置	PM54 = 1	PMC54 = 0	PFC54 = 不要求设置	KRM4 (KRM) = 1
P55	RTP05	输出	P55 = 不要求设置	PM55 = 不要求设置	PMC55 = 1	PFC55 = 1	-
	KR5	输入	P55 = 不要求设置	PM55 = 1	PMC55 = 0	PFC55 = 不要求设置	KRM5 (KRM) = 1

表 4-12. 端口引脚用作复用功能时的设置 (3/3)

引脚名称	复用功能		f Pn 寄存器的Pnx位	寄存器的Pmx位	PMCn 寄存器的PMCnx位	PFCn寄存器的PFCnx位	其它位 (寄存器)
	功能名称	I/O					
P70	ANIO	输入	P70 = 不要求设置	-	-	-	-
P71	ANI1	输入	P71 = 不要求设置	-	-	-	-
P72	ANI2	输入	P72 = 不要求设置	-	-	-	-
P73	ANI3	输入	P73 = 不要求设置	-	-	-	-
P74	ANI4	输入	P74 = 不要求设置	-	-	-	-
P75	ANI5	输入	P75 = 不要求设置	-	-	-	-
P76	ANI6	输入	P76 = 不要求设置	-	-	-	-
P77	ANI7	输入	P77 = 不要求设置	-	-	-	-
P90	TXD1	输出	P90 = 不要求设置	PM90 = 不要求设置	PMC90 = 1	PFC90 = 1	-
	KR6	输入	P90 = 不要求设置	PM90 = 1	PMC90 = 0	PFC90 = 不要求设置	KRM6 (KRM) = 1
P91	RXD1	输入	P91 = 不要求设置	PM91 = 不要求设置	PMC91 = 1	PFC91 = 1	-
	KR7	输入	P91 = 不要求设置	PM91 = 1	PMC91 = 0	PFC91 = 不要求设置	KRM7 (KRM) = 1
P96	TI51	输入	P96 = 不要求设置	PM96 = 1	PMC96 = 0	PFC96 = 不要求设置	-
	T051	输出	P96 = 不要求设置	PM96 = 不要求设置	PMC96 = 1	PFC96 = 1	-
P97	SI01	输入	P97 = 不要求设置	PM97 = 不要求设置	PMC97 = 1	PFC97 = 1	-
P98	S001	输出	P98 = 不要求设置	PM98 = 不要求设置	PMC98 = 1	PFC98 = 1	PF98 (PF9) = 无关
P99	SCR01	I/O	P99 = 不要求设置	PM99 = 不要求设置	PMC99 = 1	PFC99 = 1	PF99 (PF9) = 无关
P913	INTP4	输入	P913 = 不要求设置	PM913 = 不要求设置	PMC913 = 1	PFC913 = 1	-
P914	INTP5	输入	P914 = 不要求设置	PM914 = 不要求设置	PMC914 = 1	PFC914 = 1	-
P915	INTP6	输入	P915 = 不要求设置	PM915 = 不要求设置	PMC915 = 1	PFC915 = 1	-
PCM1	CLKOUT	输出	PCM1 = 不要求设置	PMCM1 = 不要求设置	PMCM1 = 1	-	-

4.6 注意事项

4.6.1 端口 n 寄存器 (Pn) 位操作指令的注意事项

当位操作指令执行在一个同时提供输入和输出功能的端口时，不属于操作对象目标位的输入端口的输出锁存的值也可能被写入。

因此，当端口从输入模式切换到输出模式时，推荐重写输出锁存。

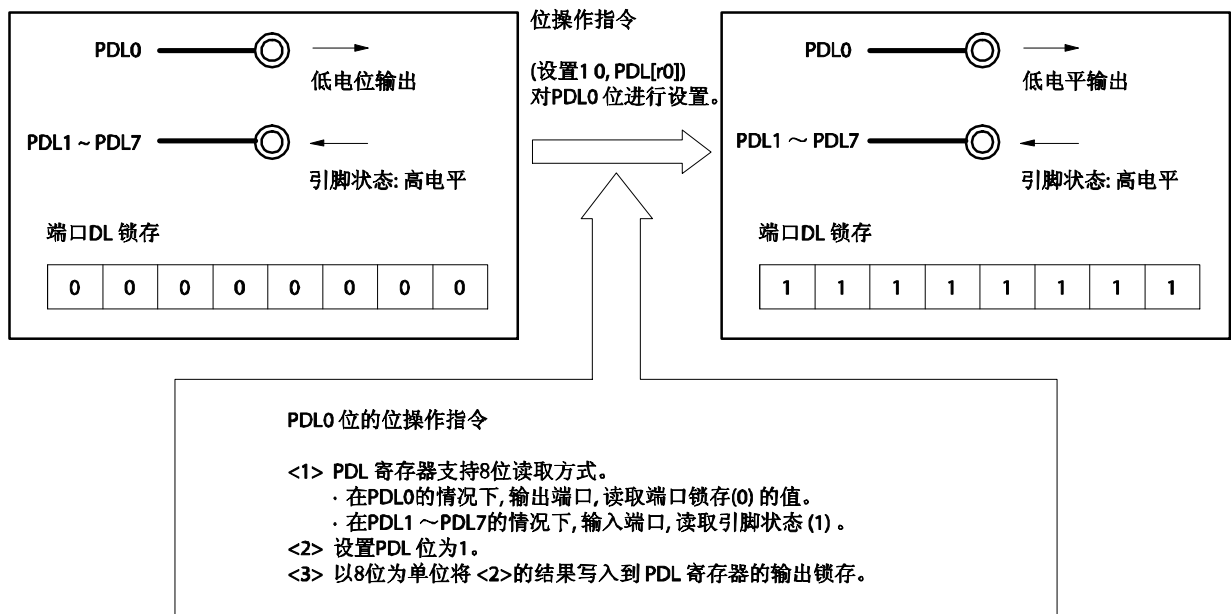
<例> 当 PDL0 引脚为输出端口，PDL1 ~ PDL7 引脚为输入端口（全部引脚状态为高电平），端口锁存的值为 00H，如果通过位操作指令将 PDL0 引脚的输出从低电平变为高电平，则端口锁存的值为 FFH。

说明：PMnm 位 = 1 的端口的 Pn 寄存器的写入和读取的对象，分别是输出锁存器和引脚状态。V850ES/KE2 位操作指令按照以下步骤执行。

- <1> 以 8 位为单位读 Pn 寄存器。
- <2> 操作目标位。
- <3> 以 8 位为单位写 Pn 寄存器。

在步骤 <1>中，PDL0 引脚作为输出端口，输出锁存的值 (0) 被读取 PDL1 ~ PDL7 引脚作为输入端口，引脚状态被读取。如果此时 PDL1 ~ PDL7 引脚的状态为高电平，则读取的值为 FEH。通过步骤 <2>，此值变为 FFH。在步骤 <3>，FFH 写到输出锁存。

图 4-21. 位操作指令 (PDL0)



4.6.2 滞后性

在端口模式下，以下端口没有滞后性。

P02 ~ P06

P31 ~ P35, P38, P39

P40, P42

P97, P99, P913 ~ P915

第五章 时钟发生器功能

5.1 概述

有如下时钟发生器功能。

○ 主时钟振荡器

<IPLL (×4) 模式>

- $f_X = 2 \sim 5 \text{ MHz}$ ($f_{XX} = 8 \sim 20 \text{ MHz}$: $4.5 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$)
- $f_X = 2 \sim 4 \text{ MHz}$ ($f_{XX} = 8 \sim 16 \text{ MHz}$: $4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$)
- $f_X = 2 \sim 2.5 \text{ MHz}$ ($f_{XX} = 8 \sim 10 \text{ MHz}$: $2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$)

<时钟直连模式>

- $f_X = 2 \sim 10 \text{ MHz}$ ($f_{XX} = 2 \sim 10 \text{ MHz}$: $2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$)

○ 副时钟振荡器

- $f_{XT} = 32.768 \text{ kHz}$

○ PLL (锁相环) 倍频 (×4) 功能

- 可选择时钟直连模式/PLL 模式
- 可用电压: $V_{DD} = 2.7 \text{ to } 5.5 \text{ V}$

○ 内部系统时钟发生器

- 7 级 (f_{XX} , $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, f_{XT})

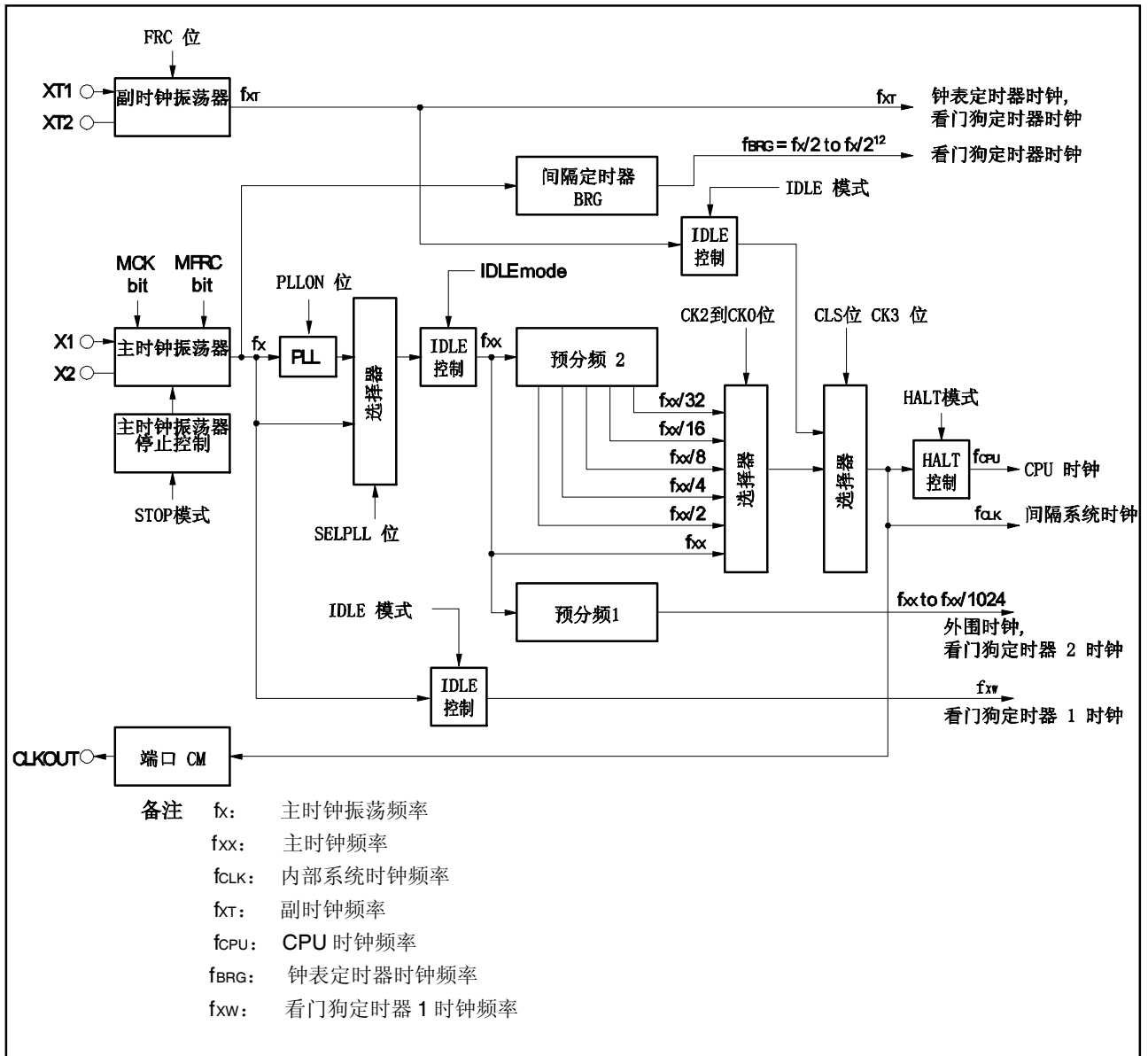
○ 外围时钟发生器功能

○ 时钟输出功能

- 备注**
- f_X : 主时钟振荡频率
 - f_{XX} : 主时钟频率
 - f_{XT} : 副时钟频率

5.2 结构图

图 5-1. 时钟发生器



(1) 主时钟振荡器

主振荡器产生以下振荡频率 (f_x) :

- $f_x = 2 \sim 5$ MHz ($V_{DD} = 4.5 \sim 5.5$ V, PLL 模式)
- $f_x = 2 \sim 4$ MHz ($V_{DD} = 4.0 \sim 5.5$ V, PLL 模式)
- $f_x = 2 \sim 2.5$ MHz ($V_{DD} = 2.7 \sim 5.5$ V, PLL 模式)
- $f_x = 2 \sim 10$ MHz ($V_{DD} = 2.7 \sim 5.5$ V, 时钟直连模式)

(2) 副时钟振荡器

副时钟振荡器产生频率为 32.768 kHz 的振荡信号 (f_{XT})。

(3) 主时钟振荡器停止控制电路

该电路生成一个可停止主时钟振荡器振荡的控制信号。

主时钟振荡器的振荡在 STOP 模式或 PCC.MCK 位 = 1 (仅在 PCC.CLS 位 = 1 时有效) 时被停止。

(4) 预分频器 1

该预分频器生成的时钟信号 ($f_{xx} \sim f_{xx}/1024$) 将提供给以下片上外围器件: TMP0, TM01, TM50, TM51, TMH0, TMH1, CSI00, CSI01, UART0, UART1, I²C0, ADC 和 WDT2。

(5) 预分频器 2

该电路对主时钟 (f_{xx}) 进行分频。

预分频器 2 生成的时钟信号 ($f_{xx} \sim f_{xx}/32$) 将提供给用于生成 CPU 时钟 (f_{CPU}) 和内部系统时钟 (f_{CLK}) 的选择器。

f_{CLK} 是提供给 INTC, ROM 和 RAM 模块的时钟信号, 并可以由 CLKOUT 引脚向外部输出。

(6) 间隔定时器 BRG

该电路将主时钟振荡器产生的时钟信号 (f_x) 分频为一个指定的频率 (32.768 kHz) 并将这个时钟频率提供给钟表定时器模块。关于更多细节, 敬请参阅第十章 间隔定时器和钟表定时器。

(7) PLL

该电路对主时钟 (f_x) 进行倍频。

PLL 可在两种模式下进行工作: 时钟直连模式, 在该模式中, f_x 按原有频率输出; PLL 模式, 在该模式中, f_x 被倍频后输出。以上模式可通过 PLLCTL.SELPLL 位进行选择。

PLL 工作的开始和停止由 PLLCTL.PLLON 位选择。

5.3 寄存器

(1) 处理器时钟控制寄存器 (PCC)

PCC 寄存器是一个特殊寄存器。只有通过特定的流程才能将数据写入该寄存器（见 3.4.7 特殊寄存器）。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 03H。

(1/2)

复位后: 03H R/W 地址: FFFF828H

	7	<6>	5	<4>	<3>	2	1	0
PCC	FRC	MCK	MFRC	CLS ^注	CK3	CK2	CK1	CK0

FRC	子时钟片上反馈电阻的使用
0	使用
1	不用

MCK	主时钟振荡器的控制
0	容许振荡
1	停止振荡
<ul style="list-style-type: none"> 当系统在主时钟作为CPU时钟操作时, 即使MCK位设置为1, 主设置操作也不停止。它在CPU时钟转变为子时钟后停止。 当主时钟停止, 设备在子时钟下操作, 清零MCK位然后在返回主时钟前等待直至程序确定振荡稳定时间。 	

MFRC	主时钟片上反馈电阻的使用
0	使用
1	不用

CLS ^注	CPU时钟状态 (f _{cpu})
0	主时钟操作
1	子时钟操作

注 CLS 位只读。

CK3	CK2	CK1	CK0	时钟选择 (f_{CLK}/f_{CPU})
0	0	0	0	f_x
0	0	0	1	$f_x/2$
0	0	1	0	$f_x/4$
0	0	1	1	$f_x/8$ (缺省值)
0	1	0	0	$f_x/16$
0	1	0	1	$f_x/32$
0	1	1	'	禁止设置
1	'	'	'	f_x

- 注意事项
1. 在通过 CLKOUT 向外输出时钟信号时，不要改变 CPU 时钟的频率（通过 CK3~CK0 位）。
 2. 对 CK3 位操作时，请使用位操作指令。如果使用 8 位操作指令，不要改变 CK2~CK0 位的值。
 3. 当 CPU 在副时钟下工作且 X1 引脚没有时钟输入时，不要访问寄存器，否则将发生等待。
（访问方法详细内容请参考 3.4.8 (2) 特殊片上外围 I/O 寄存器的访问）如果等待发生，则只能由复位释放。

备注 ×: 忽略

(a) 从主时钟操作切换~副时钟操作的举例

- <1> CK3 位 ← 1: 推荐使用位操作指令。不要改变 CK2~CK0 位的值。
 <2> 副时钟操作: 读取 CLS 位的值以检查副时钟是否已经启动。设置 CK3 位后要经过以下的时间副时钟操作才能启动。

最大值: $1/f_{XT}$ (1/副时钟频率)

- <3> MCK 位 ← 1: 只有在需要停止主时钟工作的时候才可将 MCK 位置 1。

- 注意事项** 1. 当停止主时钟工作时,也要同时停止 PLL。
 2. 如果不满足下面的条件,那么请改变 CK2~CK0 位的设置以满足该条件,然后再切换~副时钟工作模式。

内部系统时钟 (f_{CLK}) > 副时钟 (f_{XT} : 32.768 kHz) × 4

备注 内部系统时钟 (f_{CLK}): 由 CK2~CK0 位所设定的主时钟 (f_{XX}) 产生的时钟信号

[示例]

```
<1> _SET_SUB_RUN :
    st.b      r0, PRCMD[r0]
    set1      3, PCC[r0]          -- CK3 位 ← 1
<2> _CHECK_CLS :
    tst1      4, PCC[r0]          --等待副时钟工作的开始。
    bz        _CHECK_CLS
<3> _STOP_MAIN_CLOCK :
    st.b      r0, PRCMD[r0]
    set1      6, PCC[r0]          -- MCK 位 ← 1, 主时钟停止
```

备注 以上的记述只是简单的示例。请注意上述步骤<2>,对 CLS 位的检查是在一个死循环中进行的。

(b) 从副时钟操作切换~主时钟操作的举例

- <1> MCK 位 ← 0: 主时钟开始振荡
- <2> 通过软件插入等待时间, 等待主时钟振荡稳定。
- <3> CK3 位 ← 0: 推荐使用位操作指令。不要改变 CK2~CK0 位的值。
- <4> 主时钟操作: 设置 CK3 位后要经过以下的时间, 主时钟操作才能启动。

最大值: $1/f_{xt}$ (1/副时钟频率)

因此, 请在将 CK3 位清零后立即插入 NOP 指令或通过读取 CLS 位的方法检查主时钟是否开始工作。

[示例]

```

<1> _START_MAIN_OSC :
    st.b          r0, PRCMD[r0]          --解除对特殊寄存器的保护
    clr1          6, PCC[r0]             --启动主时钟振荡
<2> movea        0x55, r0, r11          --等待振荡稳定时间
    _WAIT_OST :
    nop
    nop
    nop
    addi          -1, r11, r11
    mp            r0, r11
    bne           _PROGRAM_WAIT
<3> st.b          r0, PRCMD[r0]
    clr1          3, PCC[r0]             -- CK3 ← 0
<4> _CHECK_CLS :
    tst1          4, PCC[r0]             --等待主时钟开始工作
    bnz           _CHECK_CLS

```

备注 以上的记述只是简单的示例。请注意上述步骤<4>, 对 CLS 位的检查是在一个死循环中进行的。

5.4 操作

5.4.1 各时钟的操作

下表显示了各时钟的操作状态。

表 5-1. 各时钟的操作状态

目标时钟	PCC 寄存器								
	CLS 位 = 0, MCK 位 = 0					CLS 位 = 1, MCK 位 = 0		CLS 位 = 1, MCK 位 = 1	
	复位 过程中	振荡稳定 时间等待 过程中	HALT 模式	IDLE 模式	STOP 模式	副时钟 模式	副 IDLE 模 式	副时钟 模式	副 IDLE 模式
主时钟振荡器 (fx)	×	○	○	○	×	○	○	×	×
副时钟振荡器 (fxT)	○	○	○	○	○	○	○	○	○
CPU 时钟 (fcPU)	×	×	×	×	×	○	×	○	×
内部系统时钟 (fCLK)	×	×	○	×	×	○	×	○	×
外围时钟 (fxx to fxx/1024)	×	×	○	×	×	○	×	×	×
WT 时钟 (主)	×	○	○	○	×	○	○	×	×
WT 时钟 (副)	○	○	○	○	○	○	○	○	○
WDT1 时钟 (fxw)	×	○	○	○	×	○	○	×	×
WDT2 时钟 (主):	×	×	○	×	×	○	×	×	×
WDT2 时钟 (副):	○	○	○	○	○	○	○	○	○

备注 ○: 可操作
×: 停止

5.4.2 时钟输出功能

时钟输出功能用于将内部系统时钟 (fCLK) 从 CLKOUT 引脚输出。

内部系统时钟 (fCLK) 可通过 PCC.CK3~PCC.CK0 位进行选择。

CLKOUT 引脚是作为 PCM1 引脚和时钟输出引脚复用使用的，因此使用之前请按需求设置端口 CM 的控制寄存器。

CLKOUT 引脚的状态与表 5-1 中所示的内部系统时钟的状态是一致的，该引脚可以在内部系统时钟为可操作状态时，输出时钟信号。当内部系统时钟处于停止状态时，CLKOUT 引脚输出为低电平。复位释放后，该引脚的默认模式为端口模式 (PCM1 引脚: 输入模式)。因此，直~该引脚被设置为输出模式之前，它的引脚状态保持为高阻态 (Hi-Z)。

5.4.3 外部时钟输入功能

外部时钟可以直接输入~振荡器。将时钟从 X1 引脚，其反相信号从 X2 引脚输入。将 PCC.MFRC 位置 1 (片上反馈电阻不用)。注意，即使在外部时钟模式，振荡稳定时间仍然需要写入。

5.5 PLL 功能

5.5.1 概述

PLL 功能用于输出 CPU 和片上外围功能的工作时钟，可以选择频率为晶振频率的 4 倍或选择时钟直连模式。

使用 PLL 功能时： 输入时钟 = 2 ~ 5 MHz (f_{xx}: 8 ~ 20 MHz)

时钟直连模式： 输入时钟 = 2 ~ 10 MHz (f_{xx}: 2 ~ 10 MHz)

5.5.2 寄存器

(1) PLL 控制寄存器 (PLLCTL)

PLLCTL 寄存器是用来控制 PLL 功能的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 01H。

复位后 : 01H R/W 地址: FFFFF806H

	7	6	5	4	3	<2>	<1>	<0>
PLLCTL	0	0	0	0	0	RTOST0 ^注	SELPLL	PLLON

PLLON	PLL 操作控制
0	PLL 停止
1	操作PLL

SELPLL	PLL 设置选择
0	时钟直连操作
1	PLL 操作

注 对于 RTOST0 位，参考第十二章 实时输出功能 (RTO)。

注意事项 请务必将 3~7 位清零。改变第 3 位的值并不影响操作。

5.5.3 使用方法

(1) 使用 PLL 时

- 复位信号被释放之后，PLL 处于工作状态（PLLCTL.PLLON 位 = 1），但由于默认模式为时钟直连模式（PLLCTL.SELPLL 位 = 0），为使 PLL 有效，请选择 PLL 模式（SELPLL 位 = 1）。
- 为了设置 STOP 模式（主时钟停止）或 IDLE 模式，应首先选择时钟直连模式再停止 PLL 工作。为了从 IDLE 或 STOP 模式返回之前的模式，应首先使能 PLL 操作（PLLON 位 = 1），再选择 PLL 模式（SELPLL 位 = 1）。
- 为了使能 PLL 操作，应首先将 PLLON 位置 1，等待 200 μ s，再将 SELPLL 位置 1。为了停止 PLL，应首先选择时钟直连模式（SELPLL 位 = 0），等待 8 个时钟以上，再停止 PLL（PLLON 位 = 0）。

(2) 不使用 PLL 时

- 复位信号释放后，时钟直连模式（SELPLL 位 = 0）将被默认选中，但 PLL 处于工作状态（PLLON 位 = 1），所以请务必手动关闭 PLL（PLLON 位 = 0）。

备注 PLL 在 IDLE 模式下可操作。停止 PLL 可实现低功耗。在切换到 STOP 模式时，请务必停止 PLL。

第六章 16 位定时器/事件计数器 P (TMP)

定时器 P (TMP) 是一个 16 位的定时器/事件计数器。

6.1 概述

以下是 TMP0 的概况。

- 时钟选择：： 8 通道
- 捕捉/触发输入引脚： 2 个
- 外部事件计数输入引脚： 1 个
- 外部触发输入引脚： 1 个
- 定时器/计数器：： 1 个
- 捕捉/比较寄存器： 2 个
- 捕捉/比较匹配中断请求信号： 2 个
- 定时器输出引脚：： 2 个

6.2 功能

TMP0 具有以下功能。

- 间隔定时器
- 外部事件计数器
- 外部触发脉冲输出
- 单脉冲输出
- PWM 输出
- 自由运行定时器
- 脉宽测量

6.3 配置

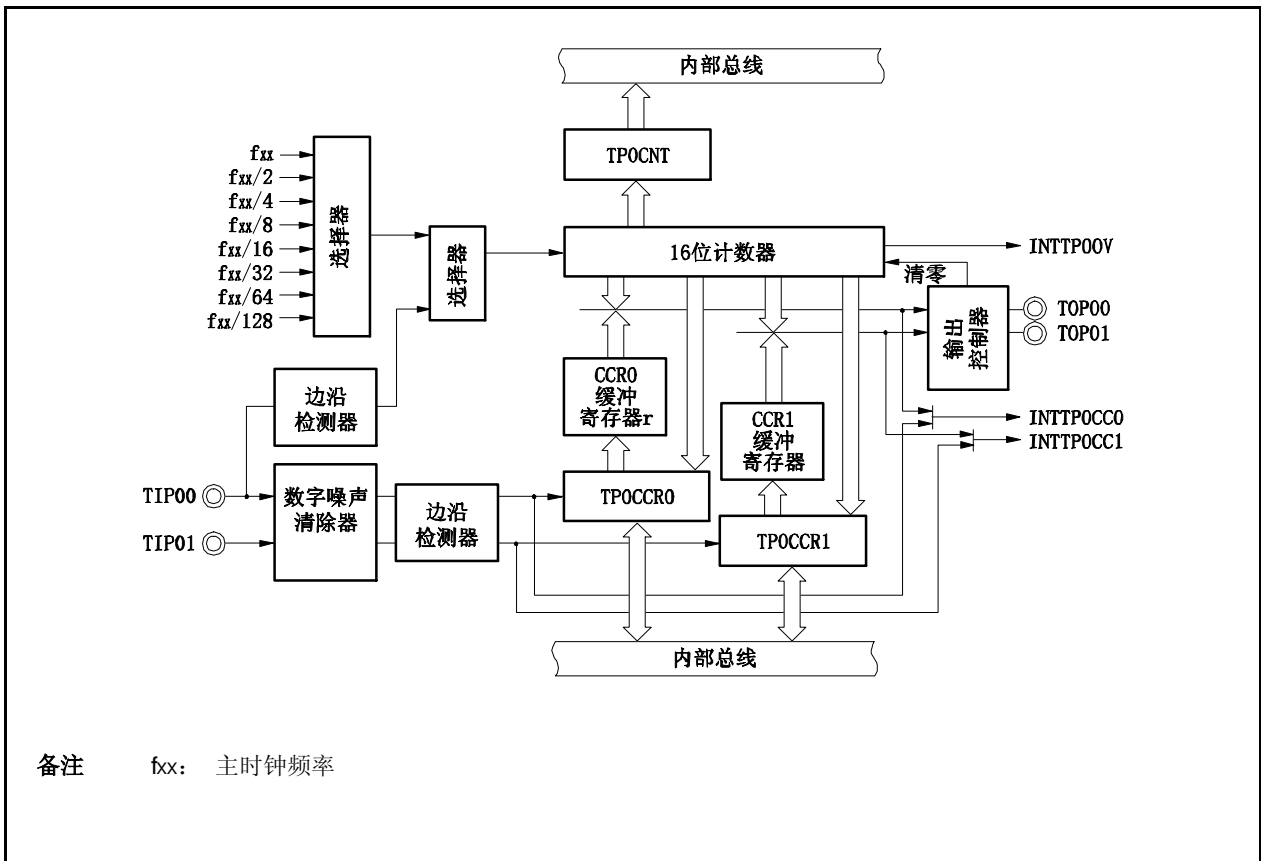
TMP0 包含以下硬件。

表 6-1. TMP0 的配置

器件	配置
定时器寄存器	16 位计数器
寄存器	TMP0 捕捉/比较寄存器 0, 1 (TP0CCR0, TP0CCR1) TMP0 计数器读取缓冲寄存器 (TPOCNT) CCR0, CCR1 缓冲寄存器
定时器输入	2 (TIP00 ^注 , TIP01 引脚)
定时器输出	2 (TOP00, TOP01 引脚)
控制寄存器	TMP0 控制寄存器 0, 1 (TP0CTL0, TP0CTL1) TMP0 I/O 控制寄存器 0 ~ 2 (TP0IOC0 ~ TP0IOC2) TMP0 选项寄存器 0 (TP0OPT0)

注 TIP00 引脚的复用功能包括捕捉触发输入信号，外部事件计数输入信号和外部触发输入信号。

图 6-1. TMP0 的框图



(1) 16 位计数器

该 16 位计数器可以对内部时钟或外部事件进行计数。

可以使用 TP0CNT 寄存器读取该计数器的计数值。

当 TP0CTL0.TP0CE 位 = 0 时，16 位计数器的值为 FFFFH。如果此时对 TP0CNT 寄存器进行读取，则读取值将为 0000H。

复位输入将 TP0CE 位清零。因此，16 位计数器也会同时被置为 FFFFH。

(2) CCR0 缓冲寄存器

CCR0 寄存器是一个用于比较 16 位计数器的计数值的 16 位比较寄存器。

当 TP0CCR0 寄存器被作为比较寄存器使用时，向 TP0CCR0 寄存器写入的数据将被传送至 CCR0 缓冲寄存器。如果 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配，会产生一个比较匹配中断请求信号 (INTTP0CC0)。

CCR0 缓冲寄存器不能被直接读/写。

复位后 CCR0 缓冲寄存器和 TP0CCR0 寄存器都将被清零。

(3) CCR1 缓冲寄存器

CCR1 寄存器是一个用于比较 16 位计数器的计数值的 16 位比较寄存器。

当 TP0CCR1 寄存器被作为比较寄存器使用时，向 TP0CCR1 寄存器写入的数据将被传送至 CCR1 缓冲寄存器。如果 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配，会产生一个比较匹配中断请求信号 (INTTP0CC1)。

CCR1 缓冲寄存器不能被直接读/写。

复位后 CCR1 缓冲寄存器和 TP0CCR1 寄存器都将被清零。

(4) 边沿检测器

该电路用于检测 TIP00 和 TIP01 引脚输入的有效边沿。通过 TP0IOC1 和 TP0IOC2 寄存器选择无有效沿，上升沿，下降沿或双沿有效。

(5) 输出控制器

该电路用于控制 TOP00 和 TOP01 引脚的输出。输出控制器是由 TP0IOC0 寄存器控制的。

(6) 选择器

该选择器用于选择 16 位计数器的计数时钟。有 8 种内部时钟和一个外部事件可供选择。

(7) 数字噪声消除器

该电路仅当 TIP0a 引脚用于捕捉触发输入引脚时有效。

该电路由 TIP0a 噪声消除寄存器 (PaNFC) 控制。

备注 a=0, 1

6.4 寄存器

(1) TMP0 控制寄存器 0 (TPOCTL0)

TPOCTL0 寄存器是用于控制 TMP0 操作的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

可使用软件向 TPOCTL0 寄存器写入相同的数值。

复位后: 00H R/W 地址: FFFF5A0H

	<7>	6	5	4	3	2	1	0
TPOCTL0	TPOCE	0	0	0	0	TPOCKS2	TPOCKS1	TPOCKS0

TPOCE	TMP0 操作控制
0	禁止TMP0 操作 (TMP0异步复位 产).
1	容许TMP0 操作. TMP0 操作开始.

TPOCKS2	TPOCKS1	TPOCKS0	内部计数时钟选择
0	0	0	fx
0	0	1	fx/2
0	1	0	fx/4
0	1	1	fx/8
1	0	0	fx/16
1	0	1	fx/32
1	1	0	fx/64
1	1	1	fx/128

注 TPOOPT0.TPOOVF 位, 16 位计数器, 定时器输出 (TOP00, TOP01 引脚)

- 注意事项
1. 在 TPOCE 位 = 0 时才可对 TPOCKS2 ~ TPOCKS0 位进行设置。
当将 TPOCE 位的值从 0 改变为 1 时, 可同时对 TPOCKS2 ~ TPOCKS0 位进行设置。
 2. 请务必将第 3~第 6 位清零。

备注 fx: 主时钟频率

(2) TMP0 控制寄存器 1 (TPOCTL1)

TPOCTL1 寄存器是用于控制 TMP0 操作的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF5A1H

	7	<6>	<5>	4	3	2	1	0
TPOCTL1	0	TPOEST	TPOEEE	0	0	TP0MD2	TP0MD1	TP0MD0

TPOEST	软件触发控制
0	-
1	为外部触发输入生成一有效信号。 ·在单脉冲输出模式下:输出一单脉冲, 数字1作为触发写入进TPOEST位。 ·在外部触发脉冲输出模式下: 输出一PWM位波形, 数字1作为触发写入进TPOEST位。

TPOEEE	计数时钟选择
0	禁止在外部事件计数输入时操作。 (在TPOCTL0.TPOCK0到TPOCTL0.TPOCK2位选择的计数时钟下进行计数操作)
1	禁止在外部事件计数输入时操作。 (在外部事件计数输入信号的有效边沿下进行计数操作)

TPOEEE位选在内部计数时钟或外部事件计数输入TPOEEE的有效边沿下进行计数操作。

TP0MD2	TP0MD1	TP0MD0	定时器模式选择
0	0	0	间隔定时器模式
0	0	1	外部事件计数模式
0	1	0	外部触发脉冲输出模式
0	1	1	单脉冲输出模式
1	0	0	PWM 输出模式
1	0	1	自由运行定时器模式
1	1	0	脉冲宽度测量模式
1	1	1	禁止设置

- 注意事项**
1. 对 TPOEST 位的设置只有在外部触发脉冲输出模式或单脉冲输出模式才有效。在其它模式下, 对该位的置 1 操作将被忽略。
 2. 在外部事件计数模式下, 无论 TPOEEE 位如何设置, 外部事件计数输入都是被选中的。
 3. 在 TPOCTL0.TPOCE 位 = 0 时才可对 TPOEEE 和 TP0MD2 ~ TP0MD0 位进行设置 (TPOCE 位 = 1 时只可以对这些位写入相同的值)。如果在 TPOCE 位 = 1 时对这些位进行了改写, 那么操作结果将不能被保证。若由于误操作引起了改写, 则要将 TPOCE 位清零然后再次设置这些寄存器位。
 4. 请务必将第 3, 4 和 7 位清零。

(3) TMP0 I/O 控制寄存器 0 (TP0IOC0)

TP0IOC0 寄存器是用于控制定时器输出的 8 位寄存器 (TOP00, TOP01 引脚)。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF5A2H

	7	6	5	4	3	<2>	1	<0>
TP0IOC0	0	0	0	0	TP0OL1	TP0OE1	TP0OL0	TP0OE0

<R>

TP0OL1	TOP01 引脚输出水平设置 ^注
0	TOP01 引脚高端启动
1	TOP01 引脚低端启动

<R>

TP0OE1	TOP01 引脚输出设置
0	禁止定时器输出 · TP0OL1 位 = 0: 低端从 TOP01 引脚输出 · TP0OL1 位 = 1: 高端从 TOP01 引脚输出
1	容许定时器输出 (一方波从 TOP01 引脚输出)。

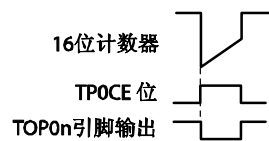
TP0OL0	TOP00 引脚输出水平设置 ^注
0	TOP00 引脚高端启动
1	TOP00 引脚低端启动

TP0OE0	TOP00 引脚输出设置
0	禁止定时器输出 · TP0OL0 位 = 0: 低端从 TOP00 引脚输出 · TP0OL0 位 = 1: 高端从 TOP00 引脚输出
1	容许定时器输出 (一方波从 TOP00 引脚输出)。

注 定时器输出引脚 (TOP0n) 的输出电平由 TP0OLn 位指定, 如下所示。

· 当 TP0OLn 位 = 0

· 当 TP0OLn 位 = 1



- 注意事项
1. 在 TP0CTL0.TP0CE 位 = 0 时才可对 TP0OL1, TP0OE1, TP0OL0 和 TP0OE0 位进行改写 (TP0CE 位 = 1 时只可以向这些位写入相同的值)。若在 TP0CE 位 = 1 由于误操作引起了改写, 则要将 TP0CE 位清零然后再次设置这些寄存器位。
 2. 在 TP0CE 位和 TP0OEm 位为 0 时, 即使对 TP0OLm 位进行了操作, TOP0m 引脚的输出电平也不能确定。(a = 0, 1)。

(4) TMP0 I/O 控制寄存器 1 (TP0IOC1)

TP0IOC1 寄存器是用于控制捕捉触发输入信号(TIP00, TIP01 引脚)有效沿的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFFF5A3H

	7	6	5	4	3	2	1	0
TP0IOC1	0	0	0	0	TP0IS3	TP0IS2	TP0IS1	TP0IS0

TP0IS3	TP0IS2	捕捉触发输入信号(TIP01 引脚)的有效边沿设置
0	0	没检测到边沿(捕捉操作无效)
0	1	检测到上升边沿
1	0	检测到下降边沿
1	1	检测到双边沿

TP0IS1	TP0IS0	捕捉触发输入信号 (TIP00 引脚)有效边沿设置
0	0	没检测到边沿(捕捉操作无效)
0	1	检测到上升边沿
1	0	检测到下降边沿
1	1	检测到双边沿

- 注意事项**
1. 在 TP0CTL0.TP0CE 位 = 0 时才可对 TP0IS3 ~ TP0IS0 位进行改写 (TP0CE 位 = 1 时只可以向这些位写入相同的值)。若在 TP0CE 位 = 1 由于误操作引起了改写, 则要将 TP0CE 位清零然后再次设置这些寄存器位。
 2. TP0IS3 ~ TP0IS0 位只有在自由运行定时器模式和脉宽测量模式下才有效。其它模式下, 无法进行捕捉操作。

(5) TMP0 I/O 控制寄存器 2 (TP0IOC2)

TP0IOC2 寄存器是用于控制外部事件计数输入信号(TIP00 引脚)有效沿和外部触发输入信号有效沿的 8 位寄存器(TIP00 引脚)。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF5A4H

	7	6	5	4	3	2	1	0
TP0IOC2	0	0	0	0	TP0EES1	TP0EES0	TP0ETS1	TP0ETS0

TP0EES1	TP0EES0	外部事件计数输入信号 (TIP00 引脚)的有效边沿设置
0	0	没检测到边沿 (外部事件计数无效)
0	1	检测到上升边沿
1	0	检测到下降边沿
1	1	检测到双边沿

TP0ETS1	TP0ETS0	外部触发输入信号 (TIP00 引脚)的有效边沿
0	0	没检测到边沿 (外部事件计数无效)
0	1	检测到上升边沿
1	0	检测到下降边沿
1	1	检测到双边沿

- 注意事项**
1. 在 TP0CTL0.TP0CE 位 = 0 时才可对 TP0EES1, TP0EES0, TP0ETS1 和 TP0ETS0 位进行改写 (TP0CE 位 = 1 时只可以向这些位写入相同的值)。若在 TP0CE 位 = 1 由于误操作引起了改写, 则要将 TP0CE 位清零然后再次设置这些寄存器位。
 2. TP0EES1 和 TP0EES0 位只有在 TP0CTL1.TP0EEE 位 = 1 或设置了外部事件计数模式 (TP0CTL1.TP0MD2 ~ TP0CTL1.TP0MD0 位 = 001)时才有效。
 3. TP0ETS1 和 TP0ETS0 位只有在外部触发脉冲输出模式 (TP0CTL1.TP0MD2 ~ TP0CTL1.TP0MD0 位 = 010) 或单脉冲输出模式 (TP0CTL1.TP0MD2 ~ TP0CTL1.TP0MD0 = 011) 下才有效。

(6) TMP0 选项寄存器 0 (TP0OPT0)

TP0OPT0 寄存器是用于设置捕捉/比较操作和检测溢出的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF5A5H

	7	6	5	4	3	2	1	<0>
TP0OPT0	0	0	TP0CCS1	TP0CCS0	0	0	0	TP0OVF

TP0CCS1	TP0CCR1 寄存器捕捉/对比选择
0	选择对比寄存器
1	选择捕捉寄存器
TP0CCS1 位设置只在自由运行模式下有效。	

TP0CCS0	TP0CCR0 寄存器捕捉/对比选择
0	选择对比寄存器
1	选择捕捉寄存器
TP0CCS0 位设置只在自由运行模式下有效。	

TP0OVF	溢出检测标志
设置(1)	发生溢出
复位(0)	TP0OVF 位写入 0 或 TP0CTL0.TP0CE 位 = 0
<ul style="list-style-type: none"> • 当在自由运行模式下或在脉冲宽度测量模式下 16 位计数器的计数值从 FFFFH 溢出到 0000H 时, 设置 TP0OVF 位。 • 在 TP0OVF 位设置为 0 的同时, 将生成一中断请求信号 (INTTP0OV)。中断请求信号 (INTTP0OV) 只在自由运行模式下或在脉冲宽度测量模式下生成。 • 当 TP0OVF 位等于 0 时, 读取 TP0OVF 或 TP0OPT0 寄存器也不能使 TP0OVF 位清零。 • TP0OVF 位可读写, 但其不能通过软件设置为 1。写入 1 对 TMP0 的操作无影响。 	

- 注意事项**
1. 在 TP0CTL0.TP0CE 位 = 0 时才可对 TP0CCS1 和 TP0CCS0 位进行改写 (TP0CE 位 = 1 时只可以向这些位写入相同的值)。若在 TP0CE 位 = 1 由于误操作引起了改写, 则要将 TP0CE 位清零然后再次设置这些寄存器位。
 2. 请务必将第 1~3 位, 第 6 和第 7 位清零。

(7) TMP0 捕捉/比较寄存器 0 (TPOCCR0)

根据选择模式的不同，TPOCCR0 可用作捕捉寄存器或比较寄存器。

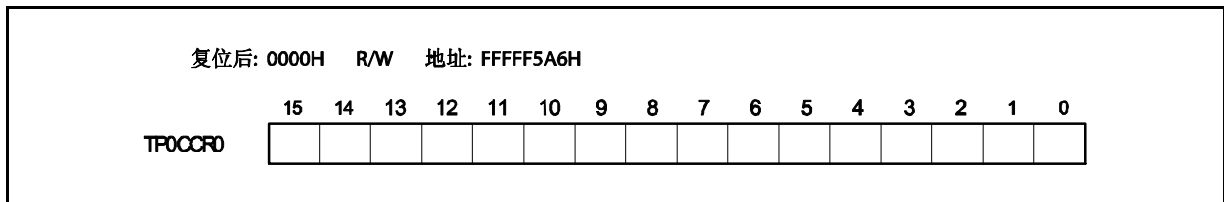
该寄存器只有在自由运行模式下根据 TP0OPT0.TP0CCS0 位设置的不同，可以用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TPOCCR0 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。

TPOCCR0 寄存器可以在操作过程中进行读写。

该寄存器支持 16 位读写方式。

该寄存器复位后的值为 0000H。

注意事项 当 CPU 工作于副时钟且主时钟振荡停止时禁止访问 TPOCCR0 寄存器。关于更多细节，敬请参阅 3.4.8 (2)。



(a) 作为比较寄存器时的功能

TP0CCR0 寄存器可以在 TP0CTL0.TP0CE 位 = 1 时被改写。

向 TP0CCR0 寄存器中设置的数值将被传送到 CCR0 缓冲寄存器中。当 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相匹配时，会产生一个比较匹配中断请求信号(INTTP0CC0)。如果允许了 TOP00 引脚输出，那么 TOP00 引脚输出的电平将被反转。

当 TP0CCR0 在间隔定时器模式，外部事件计数模式，外部触发脉冲输出模式，单脉冲输出模式或 PWM 输出模式下被用作周期寄存器时，如果 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相匹配，那么 16 位计数器将被清为 0000H。

(b) 作为捕捉寄存器时的功能

当 TP0CCR0 寄存器在自由运行定时器模式下被用作捕捉寄存器时，如果检测到了捕捉触发输入引脚 (TIP00 引脚) 的有效边沿，那么 16 位计数器的计数值将被保存到 TP0CCR0 寄存器中。在脉宽测量模式下，如果检测到了捕捉触发输入引脚 (TIP00 引脚) 的有效边沿，那么 16 位计数器的计数值将被保存到 TP0CCR0 寄存器中并且同时 16 位计数器被清为 0000H。

即使捕捉操作和读取 TP0CCR0 寄存器操作冲突，仍可正确读出 TP0CCR0 寄存器中的值。

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 6-2. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	-

(8) TMP0 捕捉/比较寄存器 1 (TP0CCR1)

根据选择模式的不同，TP0CCR1 可用作捕捉寄存器或比较寄存器。

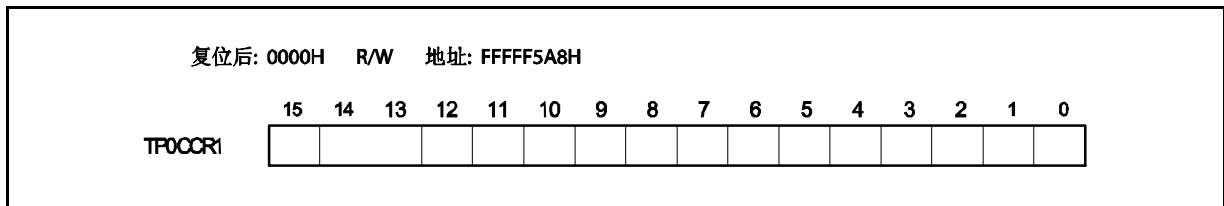
该寄存器只有在自由运行模式下根据 TP0OPT0.TP0CCS1 位设置的不同，可以用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TP0CCR1 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。

TP0CCR1 寄存器可以在操作过程中进行读写。

该寄存器支持 16 位读写方式。

该寄存器复位后的值为 0000H。

注意事项 当 CPU 工作于副时钟且主时钟振荡停止时禁止访问 TP0CCR1 寄存器。关于更多细节，敬请参阅 3.4.8 (2)。



(a) 作为比较寄存器时的功能

TP0CCR1 寄存器可以在 TP0CTL0.TP0CE 位 = 1 时被改写。

向 TP0CCR1 寄存器中设置的数值将被传送到 CCR1 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 缓冲寄存器中的数值相匹配时，会产生一个比较匹配中断请求信号(INTTP0CC1)。如果允许了 TOP01 引脚输出功能，那么 TOP01 引脚输出的电平将被反转。

(b) 作为捕捉寄存器时的功能

当 TP0CCR1 寄存器在自由运行定时器模式下被用作捕捉寄存器时，如果检测到了捕捉触发输入引脚 (TIP01 引脚) 的有效边沿，那么 16 位计数器的计数值将被保存到 TP0CCR1 寄存器中。在脉宽测量模式下，如果检测到了捕捉触发输入引脚 (TIP01 引脚) 的有效边沿，那么 16 位计数器的计数值将被保存到 TP0CCR1 寄存器中并且同时 16 位计数器被清为 0000H。

即使捕捉操作和读取 TP0CCR1 寄存器的操作产生冲突，仍可正确读出 TP0CCR1 寄存器中的值。

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 6-3. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	-

(9) TMP0 计数器读取缓冲寄存器 (TP0CNT)

可以通过读取缓冲寄存器 TP0CNT 来读取 16 位计数器的计数值。

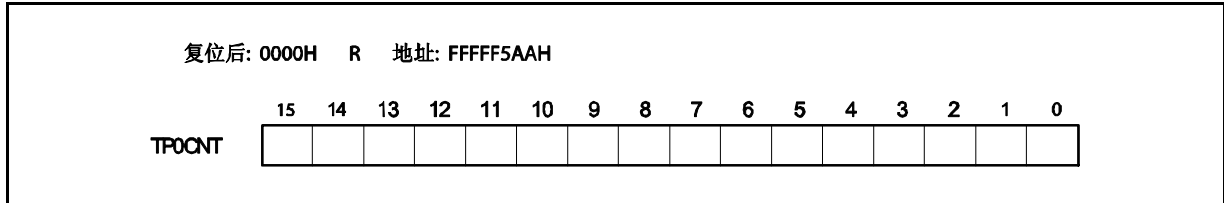
如果该寄存器在 TP0CTL0.TP0CE 位 = 1 时被读取，那么读取的值就是 16 位计数器的计数值。

该寄存器支持 16 位只读方式。

当 TP0CE 位 = 0 时，TP0CNT 寄存器将被清零。如果此时对 TP0CNT 寄存器进行读取，读取到的值将是 0000H 而不是 16 位计数器的值(FFFFH)。

复位后，TP0CE 位被清零的同时 TP0CNT 寄存器的值也被清零。

注意事项 以下情况下禁止访问 TP0CNT 寄存器。关于更多细节，敬请参阅 3.4.8 (2) 访问特定的片上外围 I/O 寄存器。



6.5 操作

TMP0 可进行以下操作。

操作	TP0CTL1.TP0EST 位 (软件触发位)	TIP00 引脚 (外部触发输入)	捕捉/比较寄存器设置	比较寄存器写入
间隔定时器模式	无效	无效	仅比较	可随时写入
外部事件计数模式 ^{注 1}	无效	无效	仅比较	可随时写入
外部触发脉冲输出模式 ^{注 2}	有效	有效	仅比较	同时写入
单脉冲输出模式 ^{注 2}	有效	有效	仅比较	可随时写入
PWM 输出模式	无效	无效	仅比较	同时写入
自由运行定时器模式	无效	无效	可转换	可随时写入
脉宽测量模式 ^{注 2}	无效	无效	仅捕捉	不可操作

- 注**
1. 使用外部事件计数模式时，请关闭对 TIP00 引脚捕捉触发输入的有效边沿的检测(通过将 TP0IOC1.TP0IS1 和 TP0IOC1.TP0IS0 位清零)。
 2. 使用外部触发脉冲输出模式，单脉冲输出模式和脉宽测量模式时，请选择内部时钟作为计数时钟(通过将 TP0CTL1.TP0EEE 位清零)。

6.5.1 间隔定时器模式 (TP0MD2 ~ TP0MD0 位 = 000)

在间隔定时器模式下，当 TP0CTL0.TPOCE 位 = 1 时，每隔指定间隔时间都会产生一个中断请求信号 (INTTP0CC0)，同时可以从 TOP00 引脚输出一个周期为 2 倍间隔时间的方波。

通常情况下，间隔定时器模式中不使用 TPOCCR1 寄存器。

图 6-2. 间隔定时器的配置图

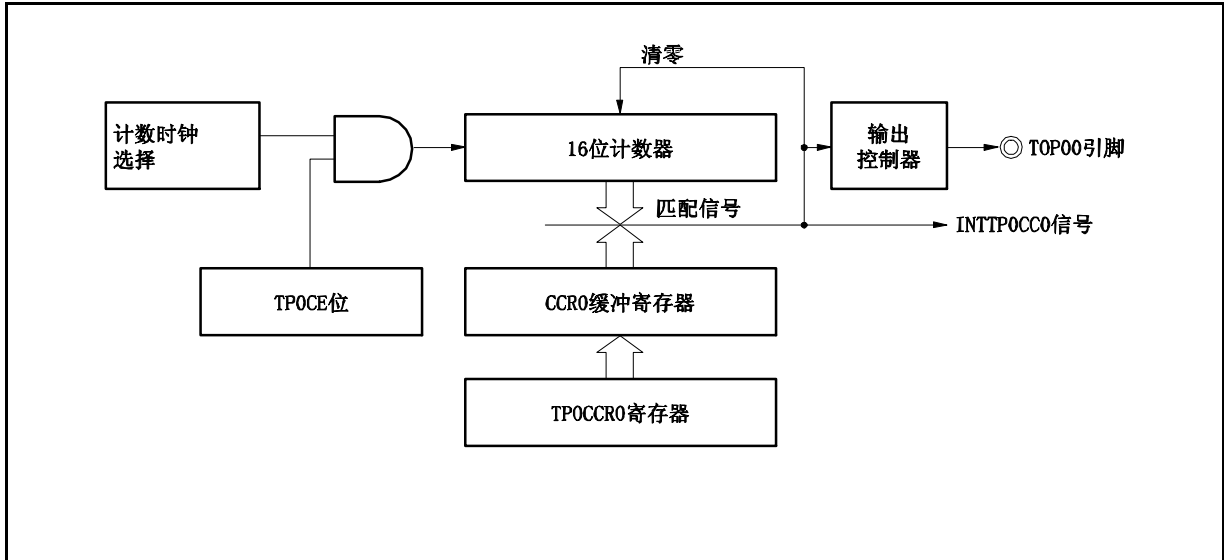
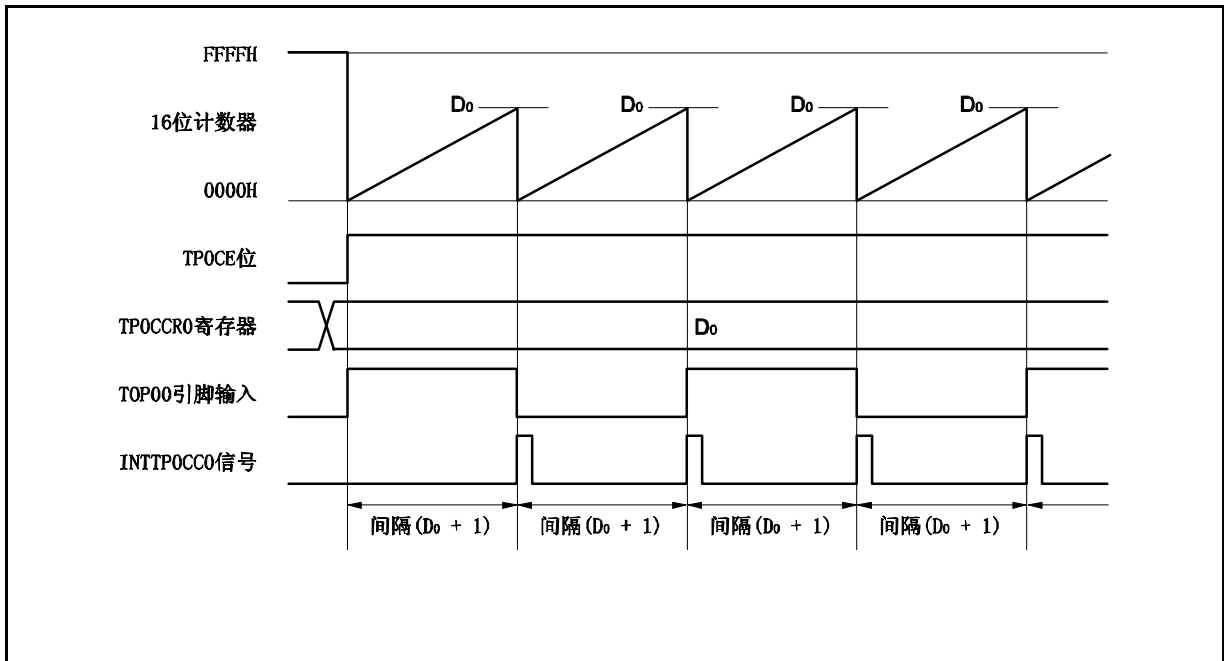


图 6-3. 间隔定时器模式操作的基本时序



16 位计数器将在 TP0CE 位被置 1 时，与计数时钟同步从 FFFFH 清零为 0000H 并开始计数。此时，TOP00 引脚的输出电平被反转。另外，TP0CCR0 寄存器的设置值也会在此时被传送到 CCR0 缓冲寄存器。

当 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值匹配时，16 位计数器清零，TOP00 引脚输出电平反转并且产生一个比较匹配中断请求信号 (INTTP0CC0)。

间隔时间可由以下算式求得。

$$\text{间隔时间} = (\text{TP0CCR0 寄存器的设定值} + 1) \times \text{计数时钟周期}$$

图 6-4. 间隔定时器模式操作的寄存器设置(1/2)

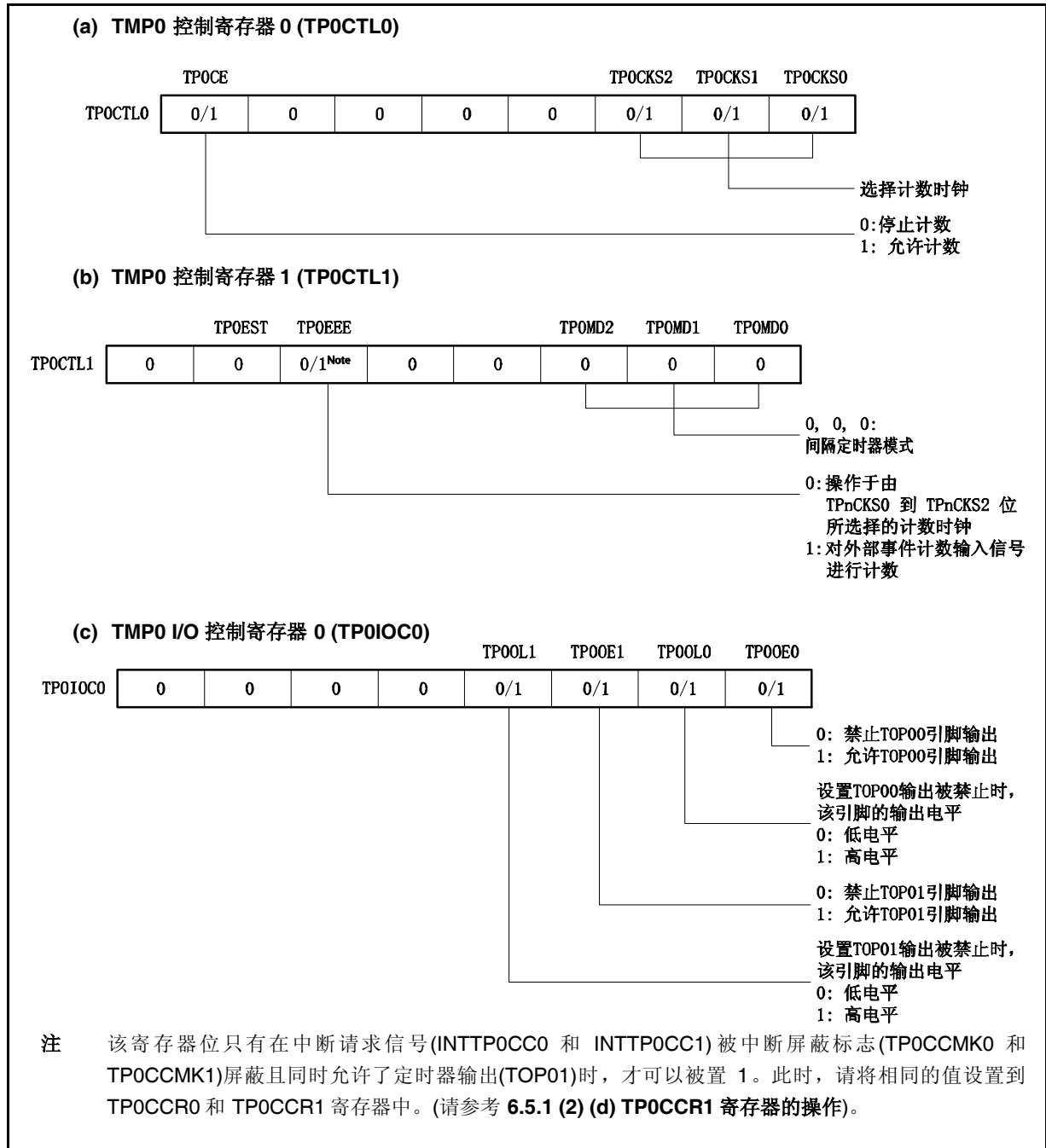


图 6-4. 间隔定时器模式操作的寄存器设置 (2/2)

(d) TMP0 计数器读取缓冲寄存器 (TP0CNT)

可以通过读取 TP0CNT 寄存器来读取 16 位计数器的计数值。

(e) TMP0 捕捉/比较寄存器 0 (TP0CCR0)

如果将 TP0CCR0 寄存器设置为 D_0 ，那么间隔时间如下。

间隔时间 = $(D_0 + 1) \times$ 计数时钟周期

(f) TMP0 捕捉/比较寄存器 1 (TP0CCR1)

通常情况下，间隔定时器模式中不使用 TP0CCR1 寄存器。但对 TP0CCR1 寄存器设置的数值也将被传送到 CCR1 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 缓冲寄存器中的数值匹配时，也会产生一个比较匹配中断请求信号(INTTP0CC1)。

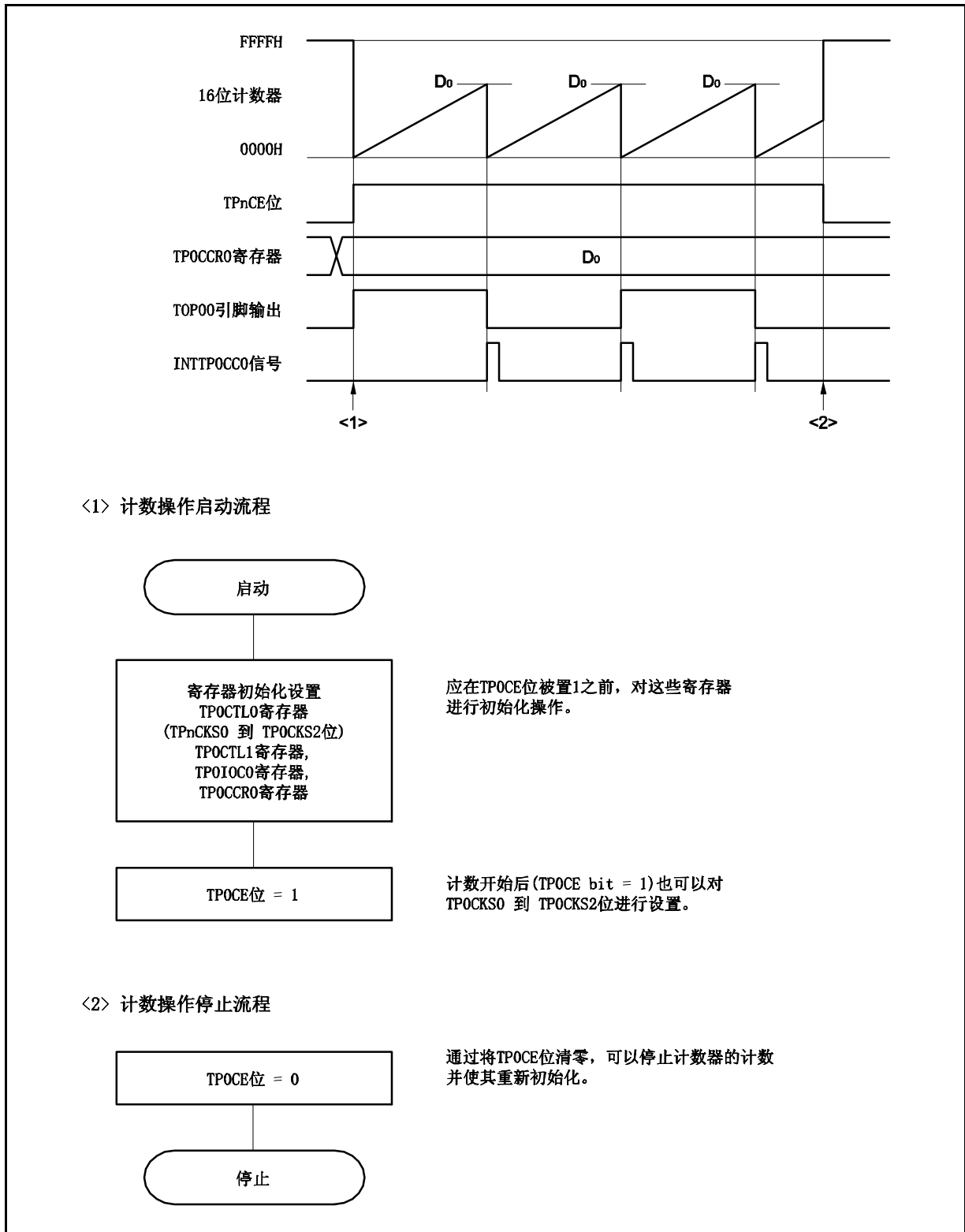
因此，请使用相应的中断屏蔽标志(TP0CCMK1)将中断请求屏蔽。

备注 间隔定时器模式下，并未使用 TMP0 I/O 控制寄存器 1 (TP0IOC1)，TMP0 I/O 控制寄存器 2 (TP0IOC2) 和 TMP0 选项寄存器 0 (TP0OPT0)。

需要设置 TP0IOC2 寄存器以使用外部事件计数输入。

(1) 间隔定时器模式操作流程

图 6-5. 间隔定时器模式下的软件处理流程

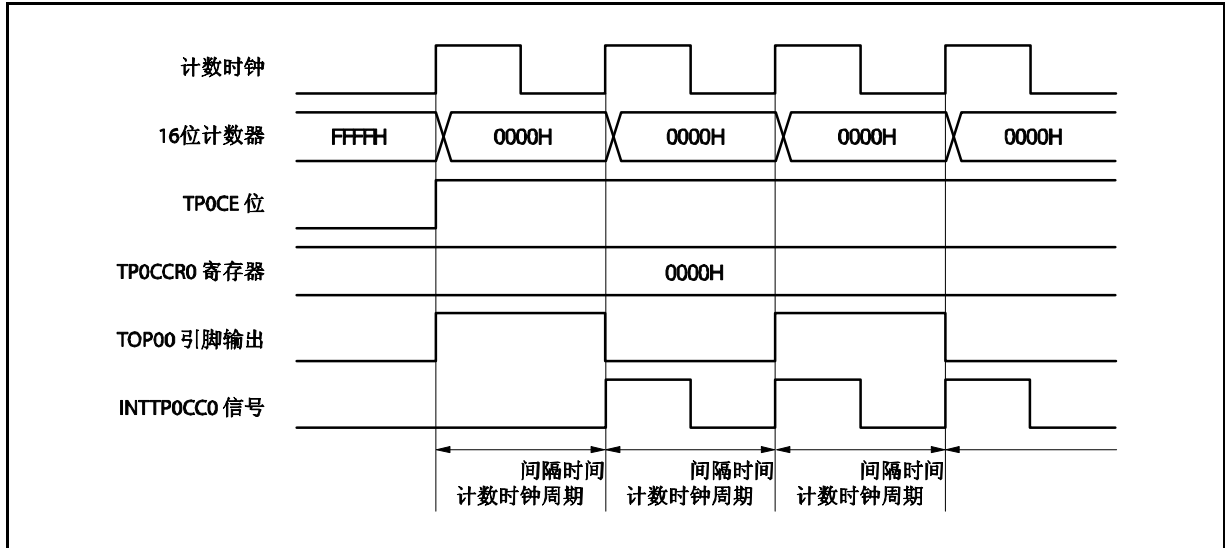


(2) 间隔定时器模式操作时序

(a) TP0CCR0 寄存器设置为 0000H 时的操作

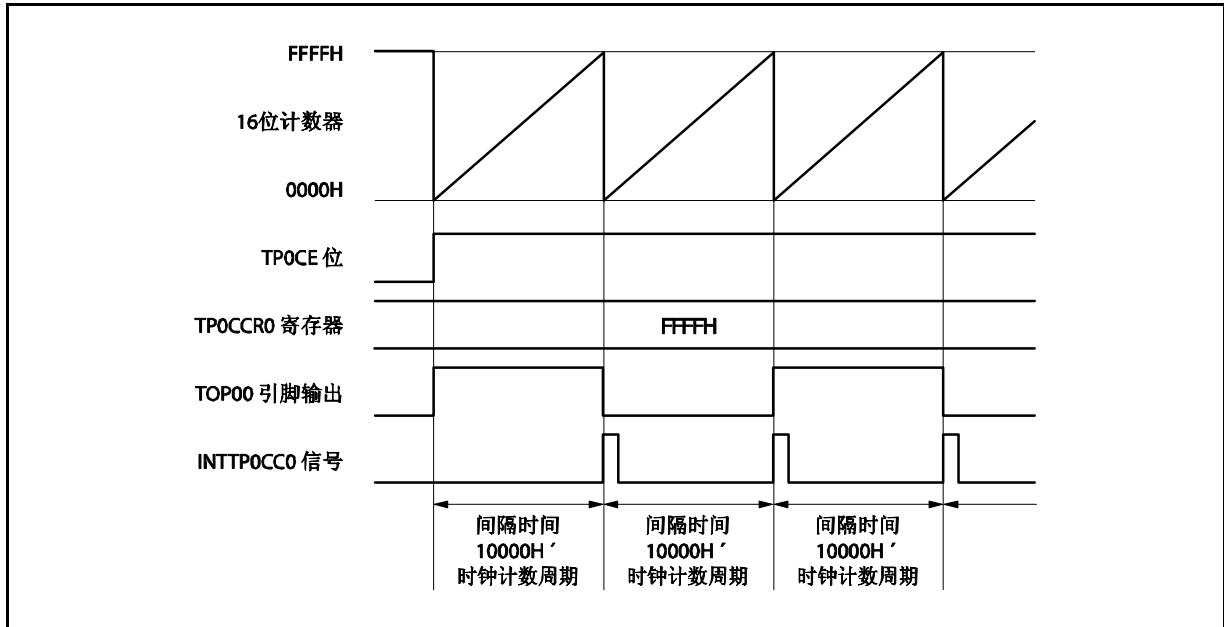
如果 TP0CCR0 寄存器被设置为 0000H，那么在第二个计数时钟之后(含第二个)每当计数时钟来临，都会产生 INTTP0CC0 信号，并且 TOP00 引脚的电平都会反转。

16 位计数器的计数值始终保持为 0000H。



(b) TP0CCR0 寄存器被设置为 FFFFH 时的操作

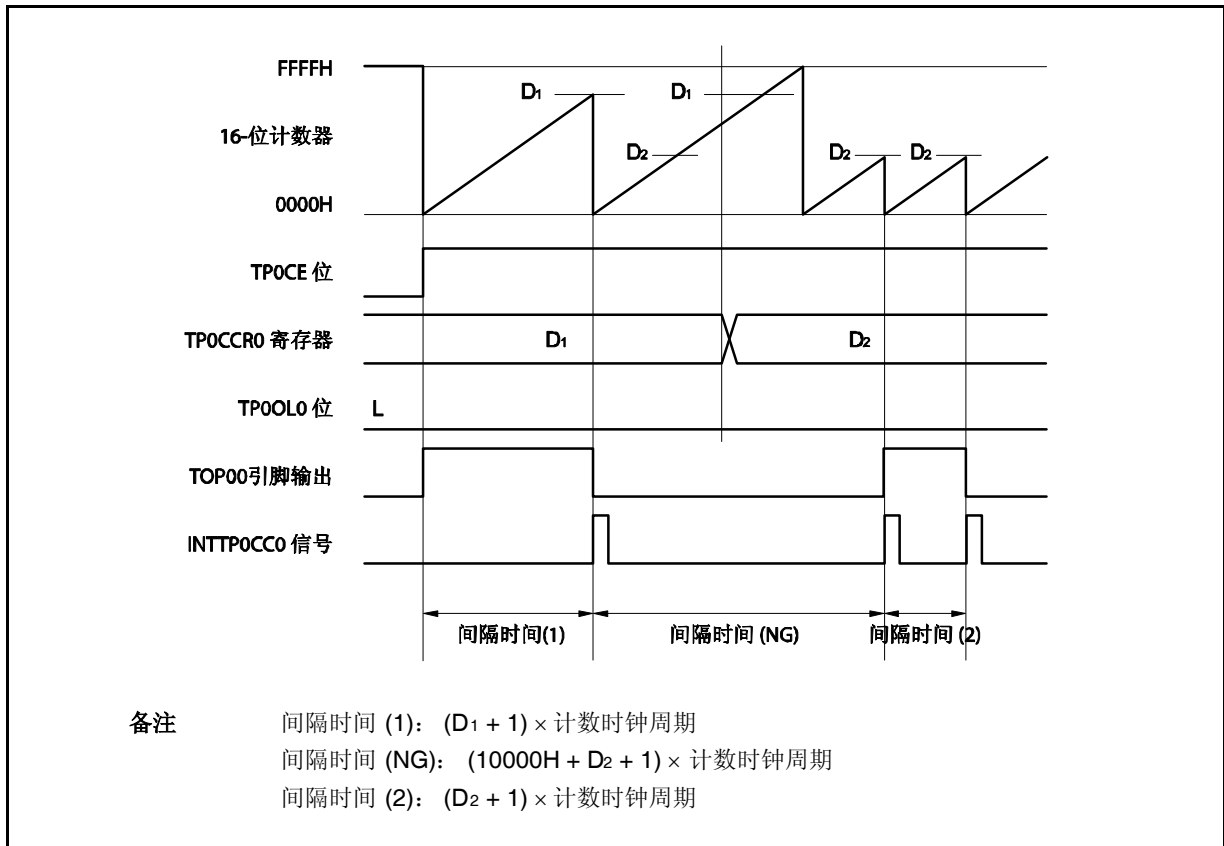
如果 TP0CCR0 寄存器被设置为 FFFFH，那么 16 位计数器将计数到 FFFFH。并在下一个计数时钟时刻同步被清零。同时产生 INTTP0CC0 信号，TOP00 引脚电平反转。此时，不会产生溢出中断请求信号 (INTTP0OV)，溢出标志位 (TP0OPT0.TP0OVF 位) 也不会被置 1。



(c) 改写 TP0CCR0 寄存器时需要注意的事项

将 TP0CCR0 寄存器的值减小时，要先停止计数，再对设定值进行改变。

如果在计数过程中将 TP0CCR0 寄存器的值减小，那么 16 位计数器可能产生溢出。

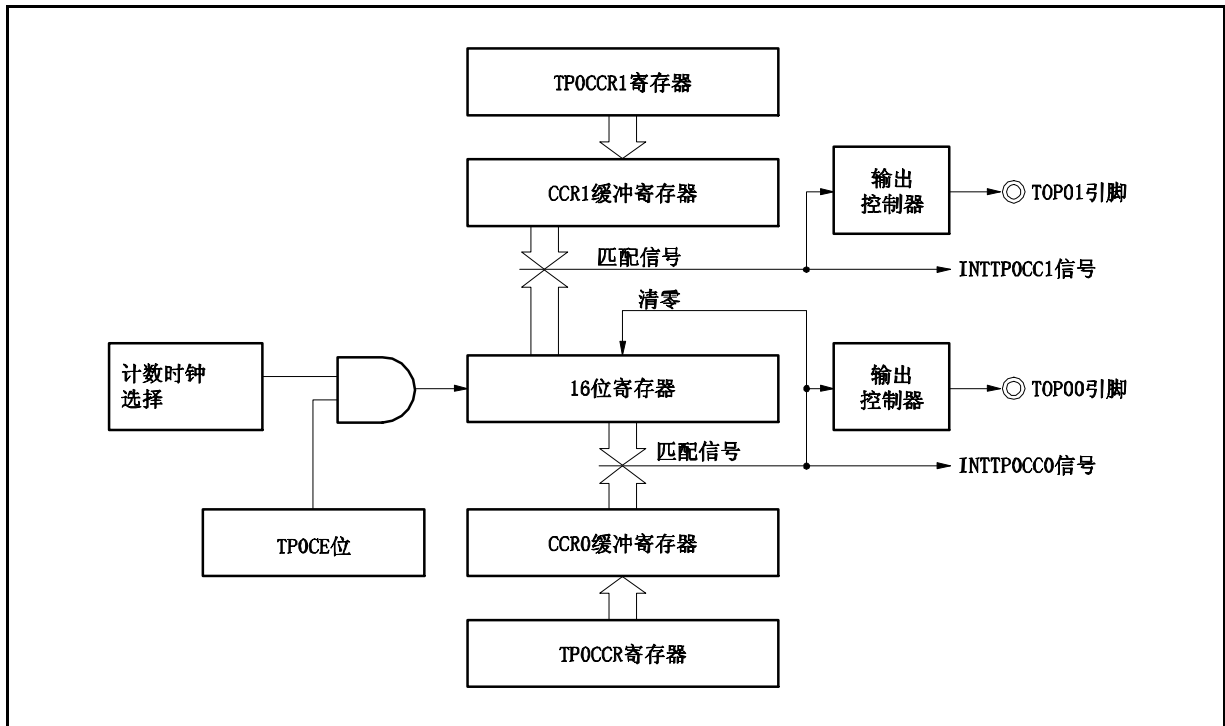


如果当前计数值大于 D_2 但小于 D_1 ，将 TP0CCR0 寄存器的值从 D_1 改变为 D_2 ，新的设定值将在改写后立即从 TP0CCR0 寄存器传送到 CCR0 缓冲寄存器中，那么，16 位计数器的计数值将与 D_2 进行比较。由于计数值已经超过了 D_2 ，因此 16 位计数器会继续计数到 FFFFH 并产生溢出，之后重新从 0000H 开始计数。在重新计数过程中，计数值与 D_2 匹配时，才会产生 INTTP0CC0 信号，同时 TOP00 引脚输出电平反转。

因此可见，INTTP0CC0 信号不会在预期的间隔时间“ $(D_1 + 1) \times$ 计数时钟周期”或“ $(D_2 + 1) \times$ 计数时钟周期”时产生，而是在一个“ $(10000H + D_2 + 1) \times$ 计数时钟周期”的间隔时间产生。

(d) TP0CCR1 寄存器的操作

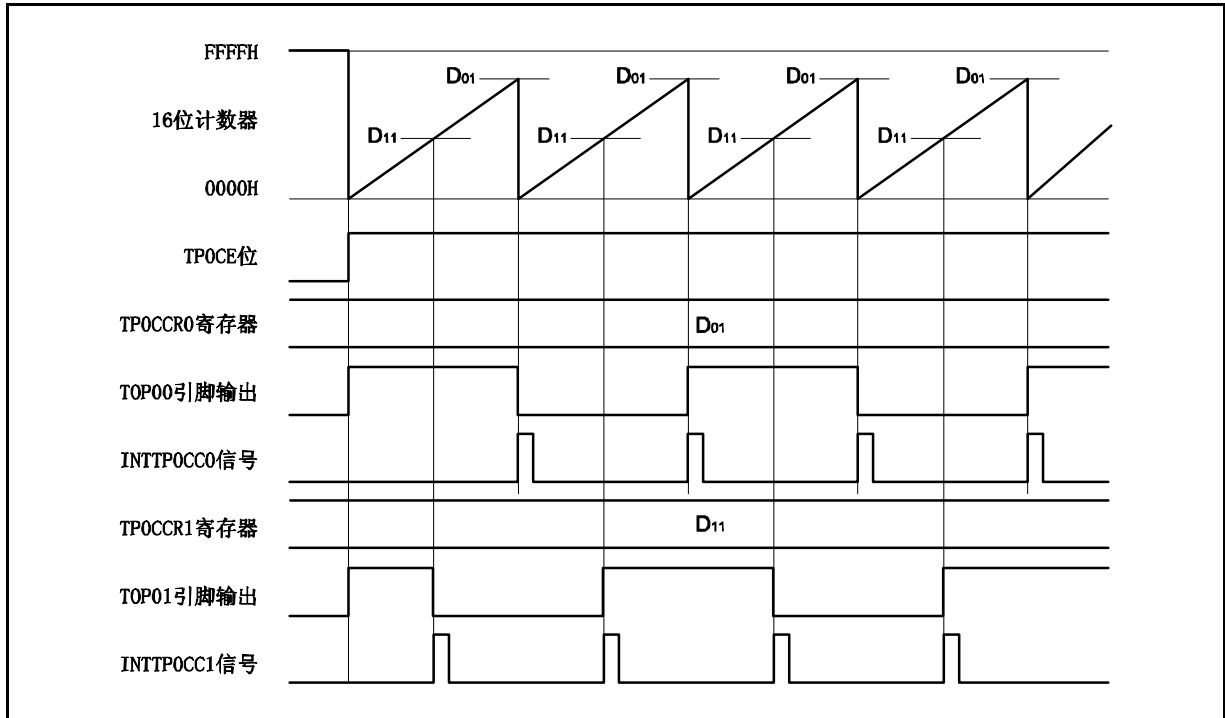
图 6-6. TP0CCR1 寄存器的配置图



如果 TP0CCR1 寄存器的值小于 TP0CCR0 寄存器的值，那么 INTTP0CC1 信号将每周期产生一次。与此同时，TOP01 引脚的电平也会反转。

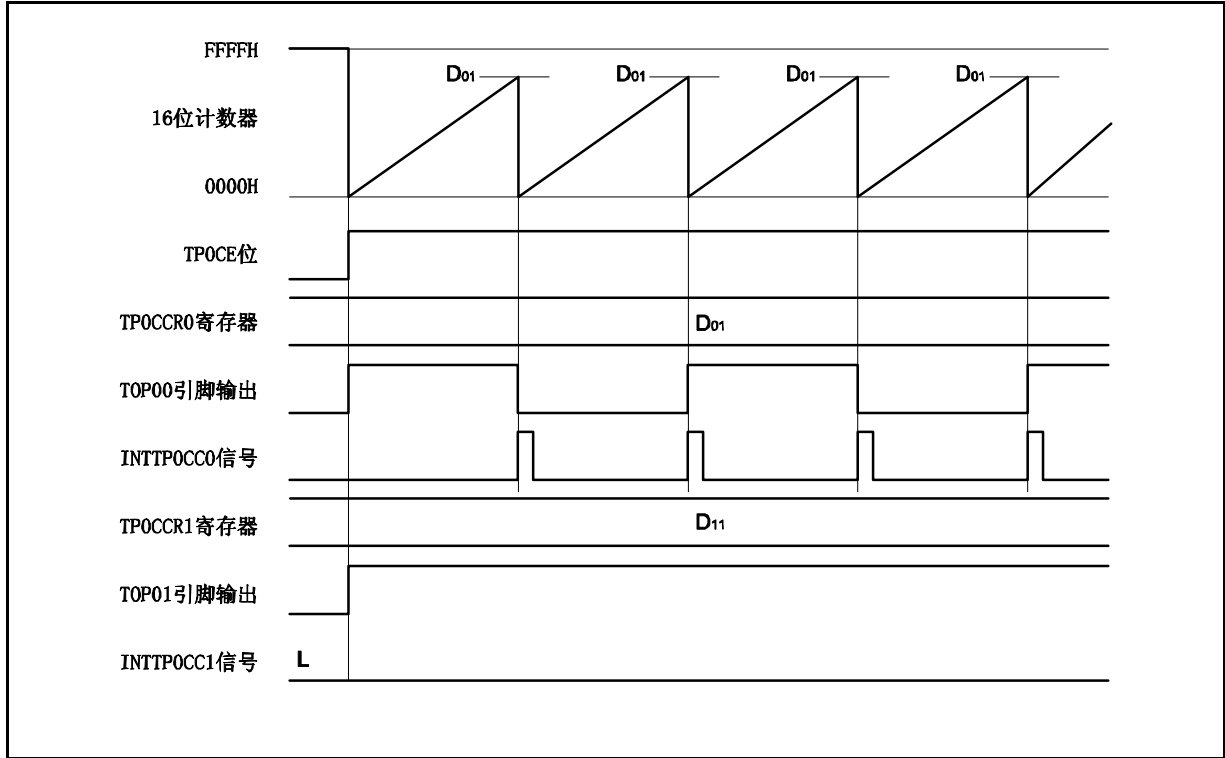
TOP01 引脚输出的方波周期与 TOP00 引脚的方波周期相同。

图 6-7. $D_{01} \geq D_{11}$ 时的时序图



如果 TP0CCR1 寄存器的值大于 TP0CCR0 寄存器的值，那么 16 位计数器的计数值不会与 TP0CCR0 寄存器中的值相匹配。因此，不会产生 INTTP0CC1 信号，TOP01 引脚的电平也不会反转。

图 6-8. $D_{01} < D_{11}$ 时的时序图



6.5.2 外部事件计数模式 (TP0MD2 ~ TP0MD0 位 = 001)

在外部事件计数模式下，当 TPOCTL0.TPOCE 位被设置为 1 时，外部事件的有效沿将被计数，并在指定数量的有效沿被计数时产生一个中断请求信号(INTTP0CC0)。该模式下不可使用 TOP00 引脚。

通常情况下，外部事件计数模式下不使用 TP0CCR1 寄存器。

图 6-9. 外部事件计数模式的配置图

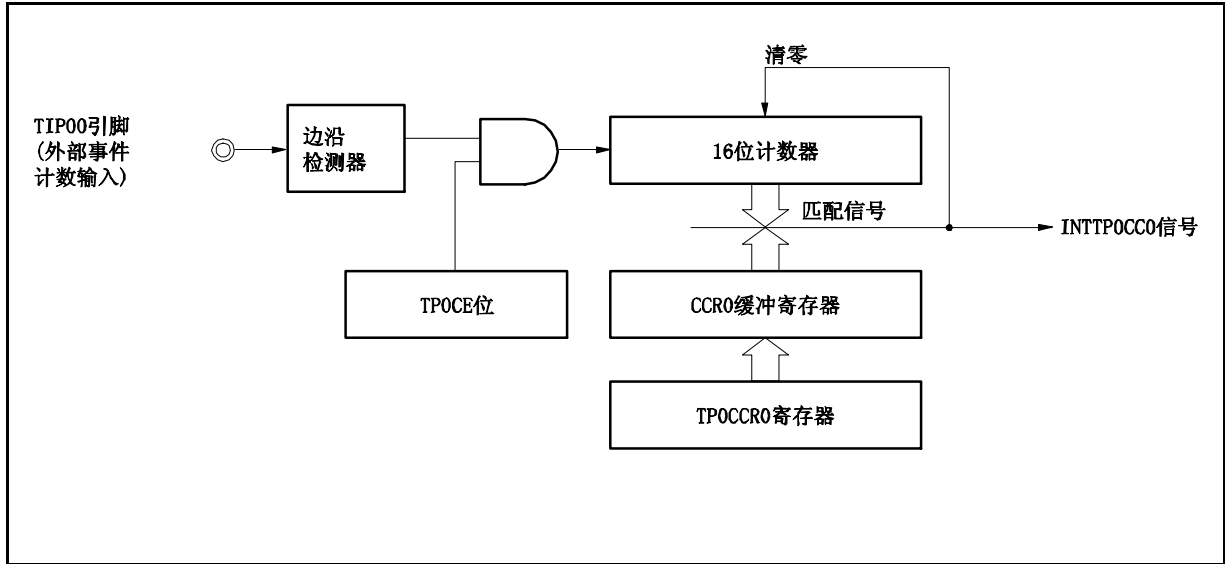
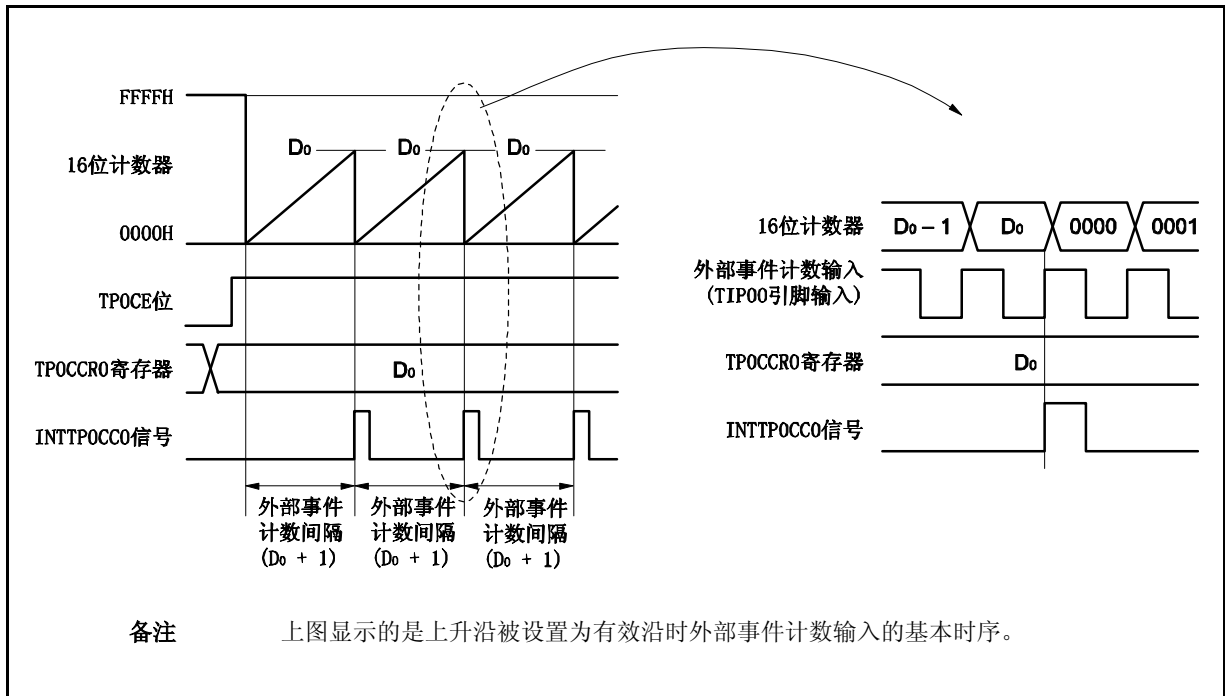


图 6-10. 外部事件计数模式的基本时序



当 TPOCE 位被设置为 1 时，16 位计数器的计数值从 FFFFH 清零为 0000H。每次外部事件计数输入的有效沿被检测时，计数器计数一次。此外，TPOCCR0 寄存器的设定值会被传送到 CCR0 缓冲寄存器中。

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值相匹配时，16 位计数器将被清零，并且产生一个比较匹配中断请求信号(INTTPOCC0)。

每当检测到(TPOCCR0 寄存器设置值 + 1)次外部事件计数输入的有效沿时，都会产生 INTTPOCC0 信号。

图 6-11. 外部事件计数模式操作的寄存器设置 (1/2)

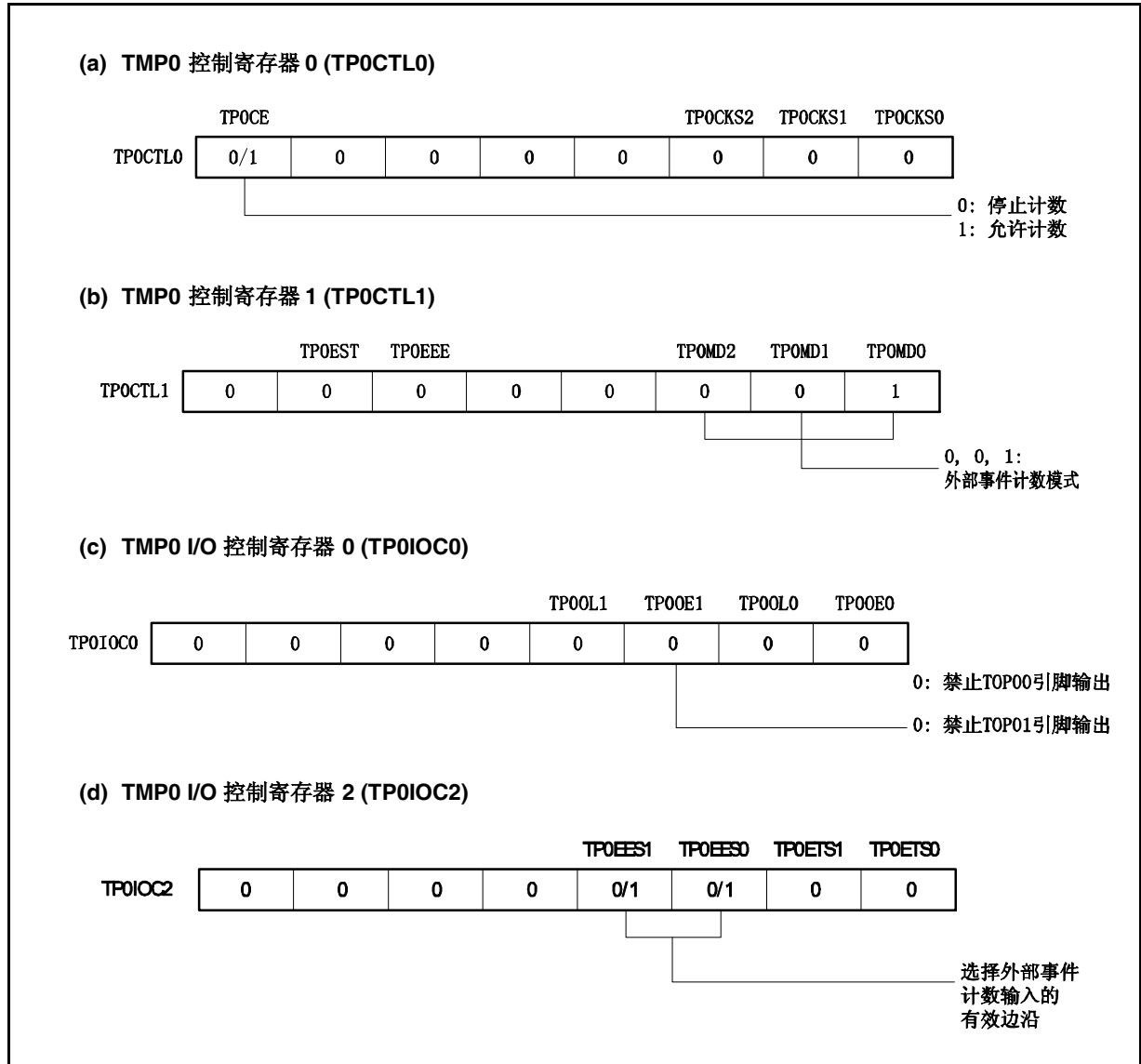


图 6-11. 外部事件计数模式操作的寄存器设置 (2/2)

(e) TMP0 计数器读取缓冲寄存器 (TP0CNT)

可以通过读取 TP0CNT 寄存器来读取 16 位计数器的计数值。

(f) TMP0 捕捉/比较寄存器 0 (TP0CCR0)

如果向 TP0CCR0 寄存器写入 D_0 ，则计数器被清零，并且将在外部事件计数达到 $(D_0 + 1)$ 时产生比较匹配中断请求信号 (INTTP0CC0)。

(g) TMP0 捕捉/比较寄存器 1 (TP0CCR1)

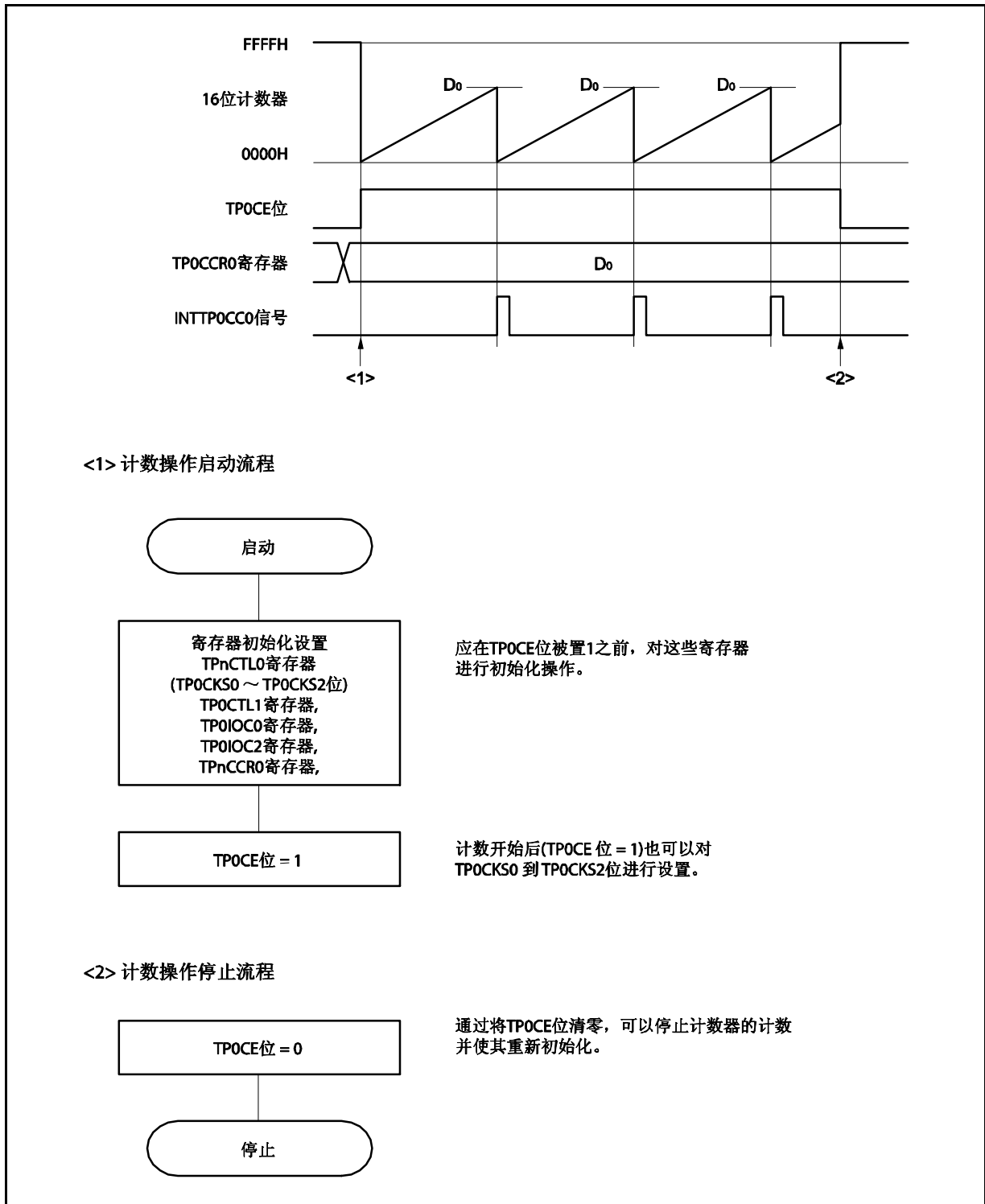
通常情况下，外部事件计数模式中不使用 TP0CCR1 寄存器。但对 TP0CCR1 寄存器设置的数值也将被传送到 CCR1 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 缓冲寄存器中的数值匹配时，也会产生一个比较匹配中断请求信号 (INTTP0CC1)。

因此，请使用相应得中断屏蔽标志 (TP0CCMK1) 将中断请求屏蔽。

备注 外部事件计数模式下，并未使用 TMP0 I/O 控制寄存器 1 (TP0IOC1) 和 TMP0 选项寄存器 0 (TP0OPT0)。

(1) 外部事件计数模式操作流程

图 6-12. 外部事件计数模式的软件处理流程

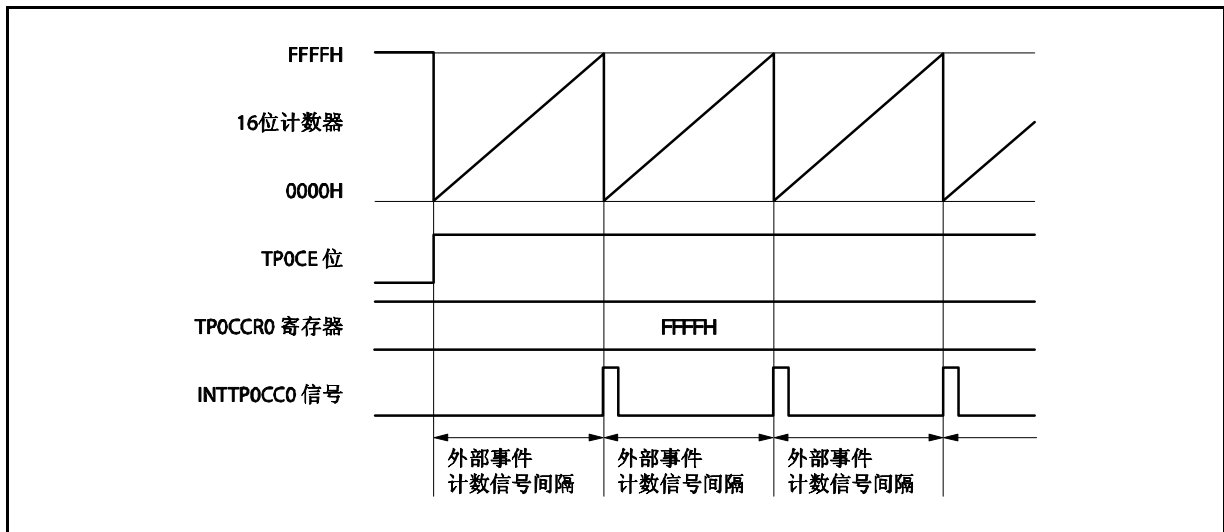


(2) 外部事件计数模式的操作时序

- 注意事项
1. 在外部事件计数模式中，不要将 TP0CCR0 和 TP0CCR1 寄存器设置为 0000H。
 2. 在外部事件计数模式中，禁止使用定时器输出功能。如果要将外部事件计数输入作为定时器输出使用，那么请将工作模式设置为间隔定时器模式并将外部事件计数输入设定为计数时钟 (TP0CTL1.TP0MD2 ~ TP0CTL1.TP0MD0 位 = 000, TP0CTL1.TP0EEE 位 = 1)。

(a) TP0CCR0 寄存器被设置为 FFFFH 时的操作

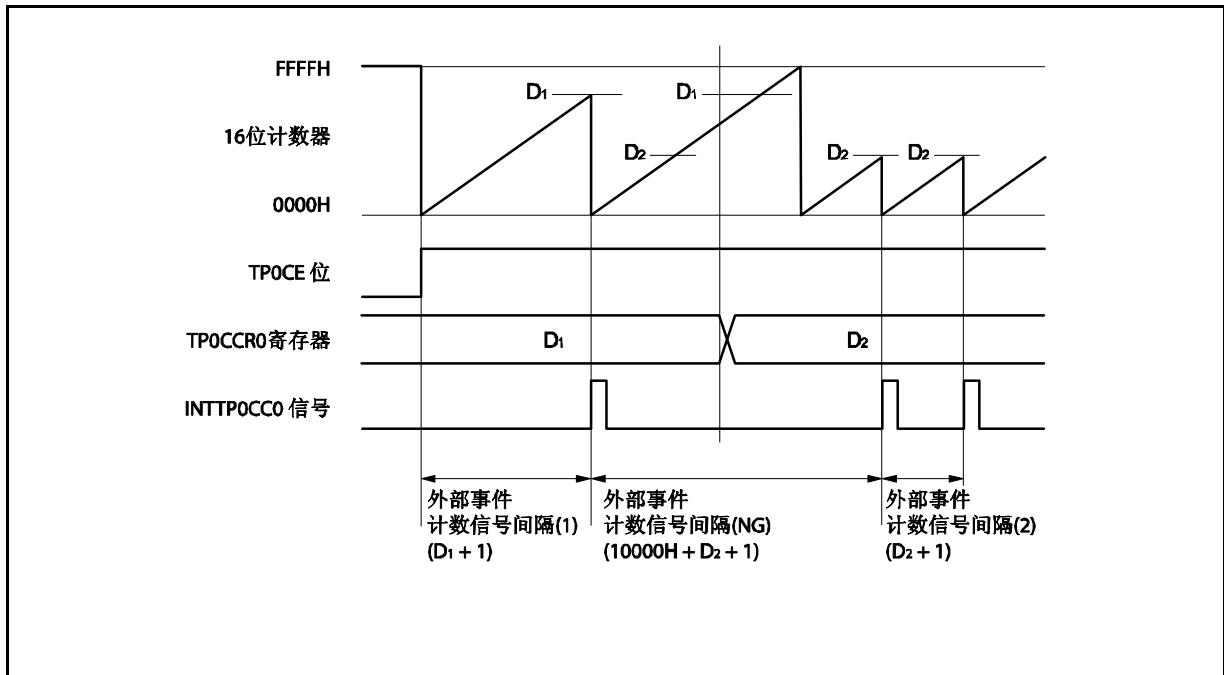
如果 TP0CCR0 寄存器被设置为 FFFFH，那么 16 位计数器将在每次检测到外部事件计数信号的有效沿处计数，直到 FFFFH。并在下一个计数事件时刻被同步清零，同时产生 INTTP0CC0 信号。此时，溢出标志位 (TP0OPT0.TP0OVF 位) 不会被置 1。



(b) 改写 TP0CCR0 寄存器时需要注意的事项

将 TP0CCR0 寄存器的值减小时，要先停止计数，再对设定值进行改变。

如果在计数过程中将 TP0CCR0 寄存器的值减小，那么 16 位计数器可能产生溢出。



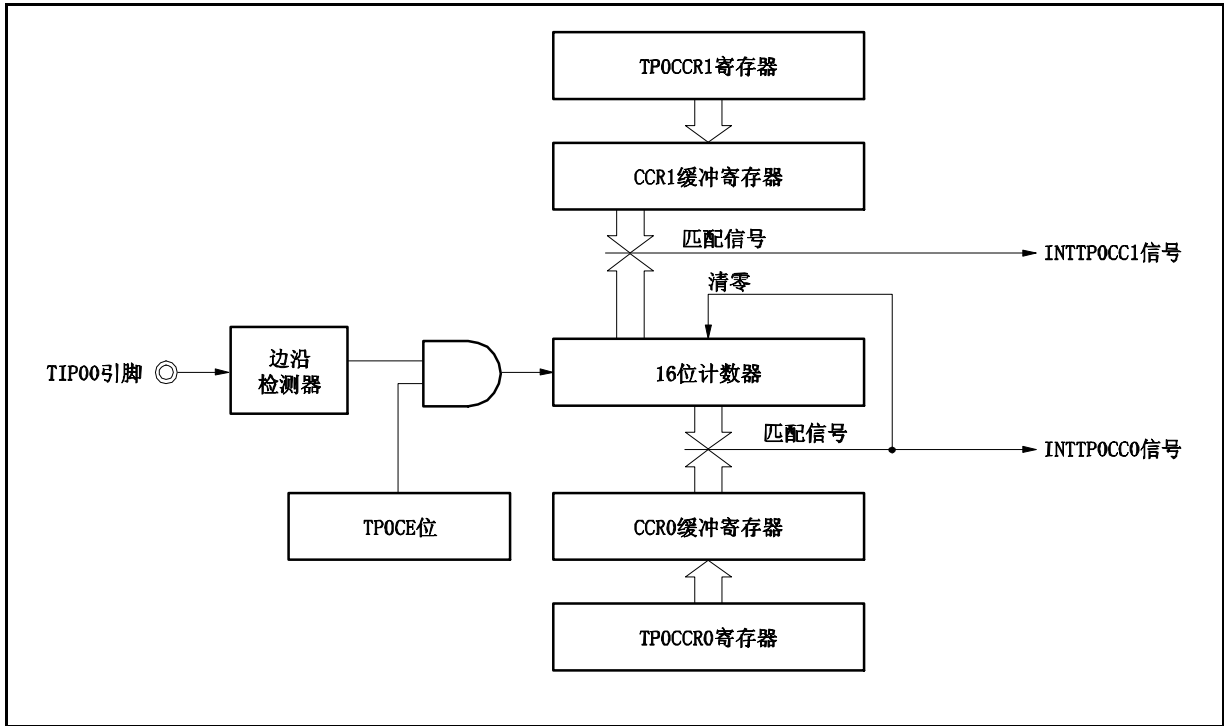
如果当前计数值大于 D_2 但小于 D_1 ，将 TP0CCR0 寄存器的值从 D_1 改变为 D_2 ，新的设定值将在改写后立即从 TP0CCR0 寄存器传送到 CCR0 缓冲寄存器中，那么，16 位计数器的计数值将与 D_2 进行比较。

由于计数值已经超过了 D_2 ，因此 16 位计数器会继续计数到 FFFFH 并产生溢出，之后重新从 0000H 开始计数。在重新计数过程中，计数值与 D_2 匹配时，才会产生 INTTP0CC0 信号。

由此可见，INTTP0CC0 信号不会在预期的事件计数次数“ $(D_1 + 1)$ 次”或“ $(D_2 + 1)$ 次”时产生，而是在一个“ $(10000H + D_2 + 1)$ 次”的计数次数时产生。

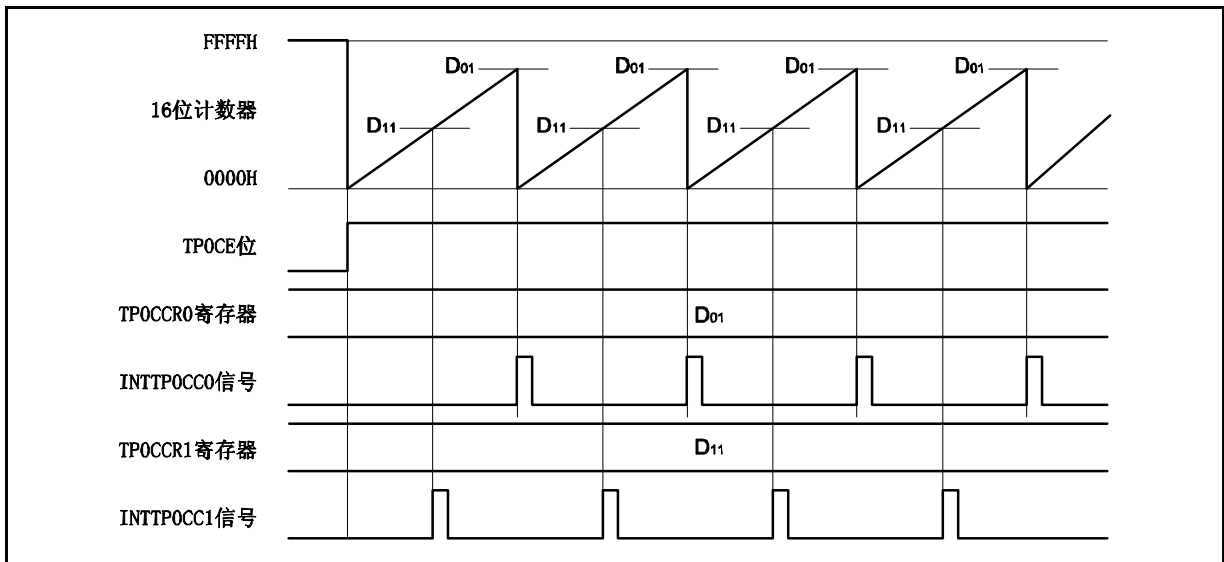
(c) TP0CCR1 寄存器的操作

图 6-13. TP0CCR1 寄存器的配置



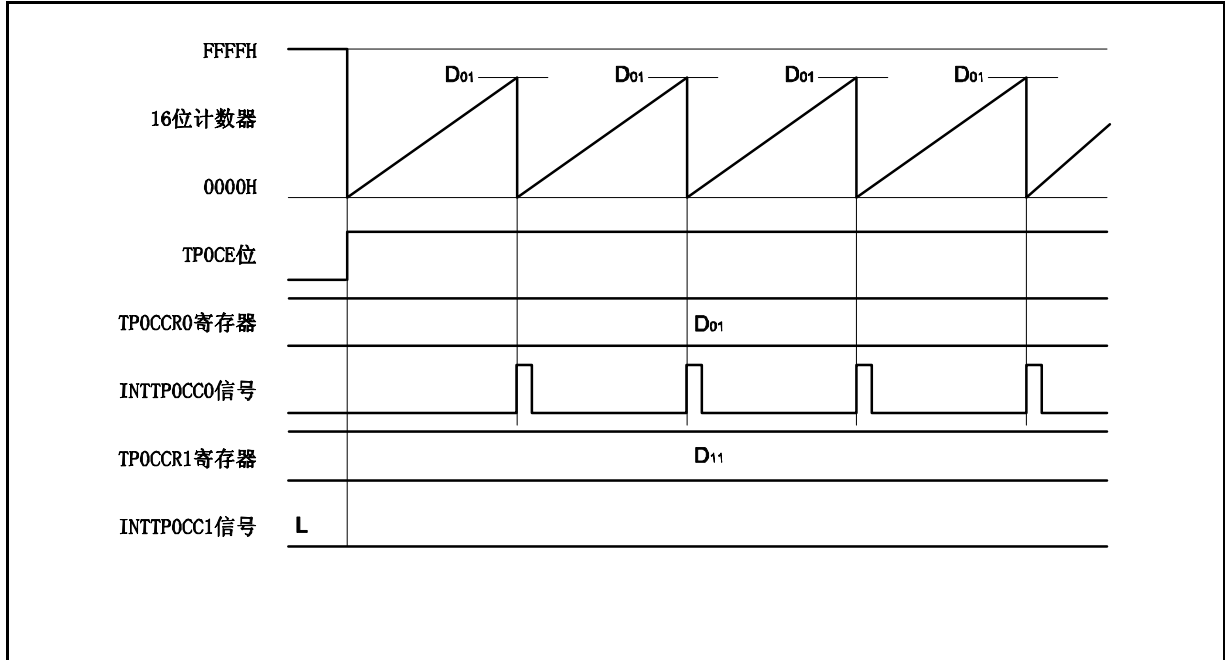
如果 TP0CCR1 寄存器的值小于 TP0CCR0 寄存器的值，那么 INTTP0CC1 信号将每周期产生一次。

图 6-14. $D_{01} \geq D_{11}$ 时的时序图



如果 TP0CCR1 寄存器的值大于 TP0CCR0 寄存器的值，那么 16 位计数器的计数值不会与 TP0CCR0 寄存器中的值相匹配。因此，不会产生 INTTP0CC1 信号。

图 6-15. $D_{01} < D_{11}$ 时的时序图



6.5.3 外部触发脉冲输出模式 (TP0MD2 ~ TP0MD0 位 = 010)

在外部触发脉冲输出模式中，TP0CTL0.TP0CE 位被置 1 后，16 位定时器/事件计数器 P 进入等待触发信号的状态。当检测到一个外部触发输入信号的有效沿时，16 位定时器/事件计数器 P 开始计数，并由 TOP01 引脚输出一个 PWM 波形。

也可以使用软件触发代替外部触发来开始脉冲的输出。当使用软件触发时，还可由 TOP00 引脚输出一个以 PWM 波形周期的 2 倍为周期的方波。

图 6-16. 外部触发脉冲输出模式的配置

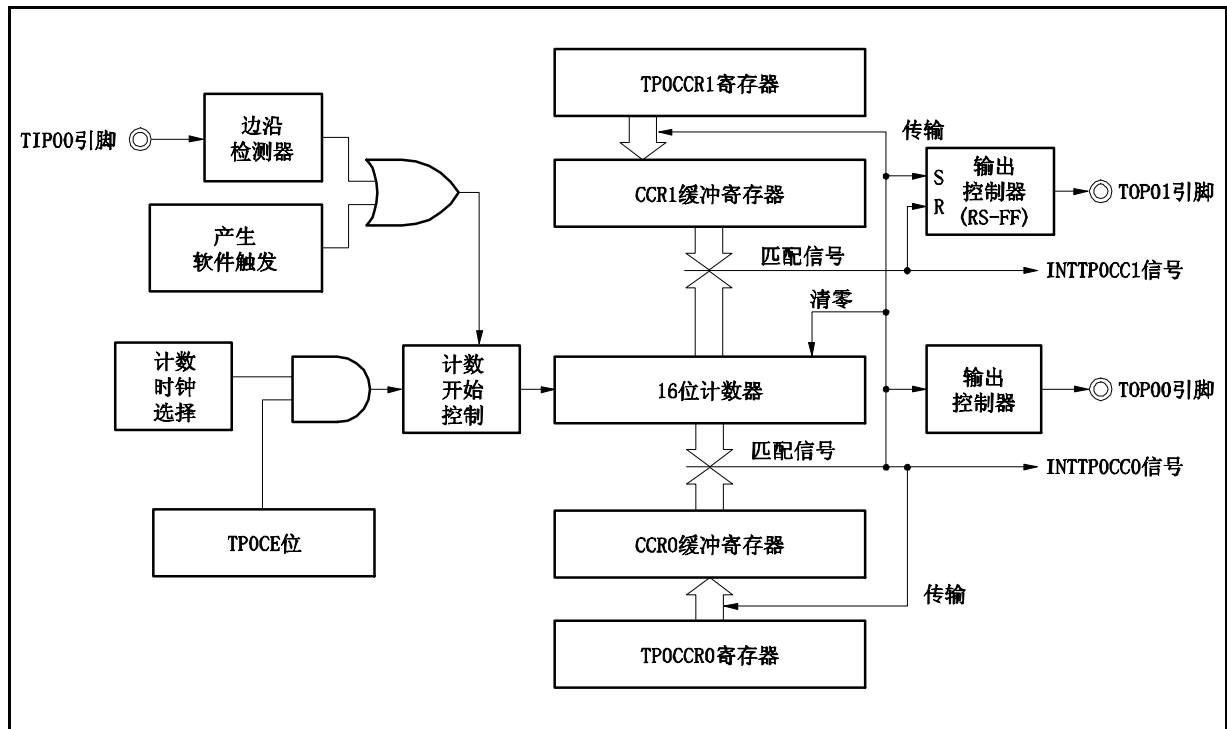
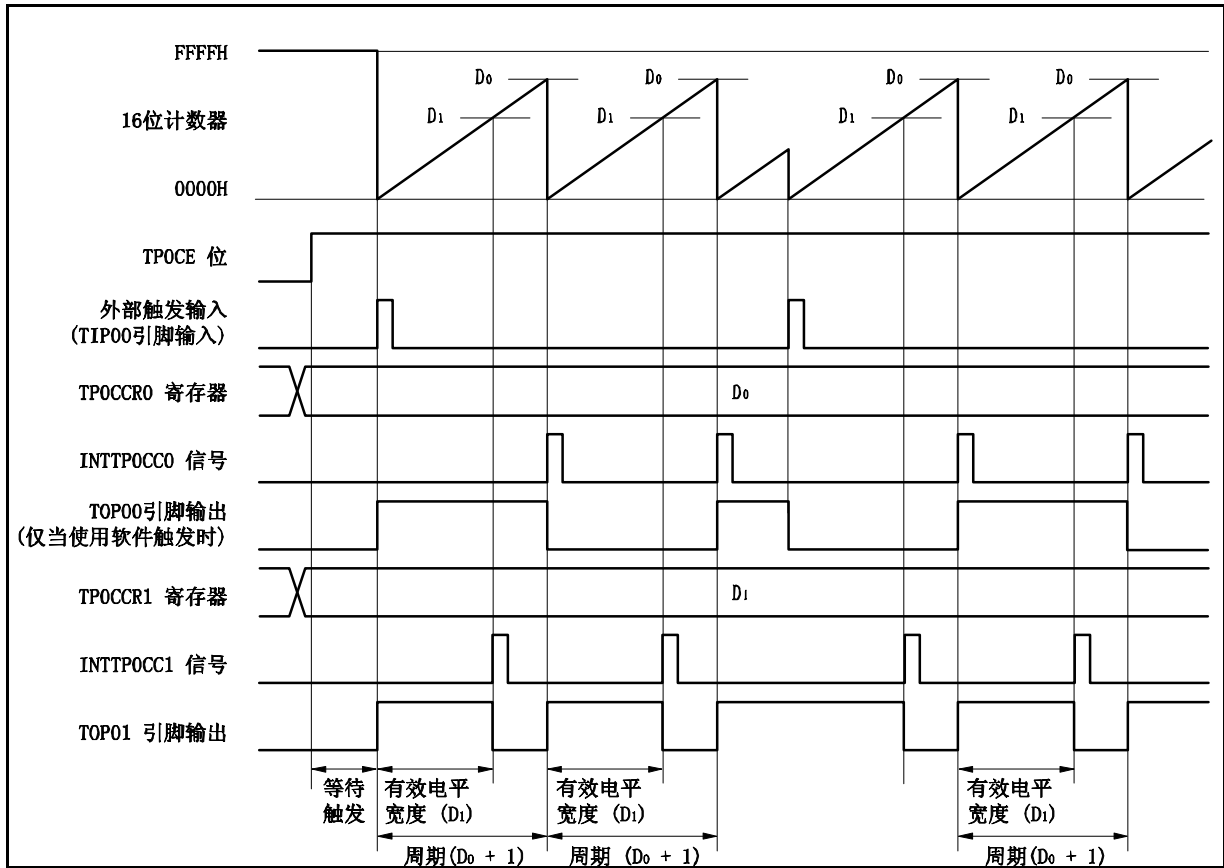


图 6-17. 外部触发脉冲输出模式的基本时序



TPOCTL0.TPOCE 位被置 1 后, 16 位定时器/事件计数器 P 进入等待触发信号的状态。触发被检测后, 16 位计数器从 FFFFH 清零为 0000H, 同时开始计数并从 TOP01 引脚输出一个 PWM 波形。如果在计数器计数过程中, 再次检测到触发信号, 那么计数器将被清零并重新开始计数。(TOP00 引脚的输出电平反转。TOP01 在触发产生时输出高电平(无论当前状态为高/低)。)

PWM 波形的有效电平宽度, 周期以及占空比系数可由以下的算式得出。

有效电平宽度 = (TPOCCR1 寄存器的设置值) × 计数时钟周期

周期 = (TPOCCR0 寄存器的值 + 1) × 计数时钟周期

占空比系数 = (TPOCCR1 寄存器的值)/(TPOCCR0 寄存器的值 + 1)

比较匹配中断请求信号 INTTP0CC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器中的值相匹配后产生, 同时 16 位计数器也会被清零。比较匹配中断请求信号 INTTP0CC1 会在 16 位计数器的计数值与 CCR1 缓冲寄存器中的值相匹配时产生。

当 16 位计数器的计数值与 CCRm 缓冲寄存器中的值相匹配时, TPOCCRm 寄存器中设置的值会被传送到 CCRm 缓冲寄存器, 同时 16 位计数器被清零。

触发源可以从外部触发输入信号的有效沿和设置软件触发(TPOCTL1.TP0EST 位 = 1)的方式中进行选择。

备注 a=0, 1

图 6-18. 外部触发脉冲输出模式的寄存器设置 (1/2)

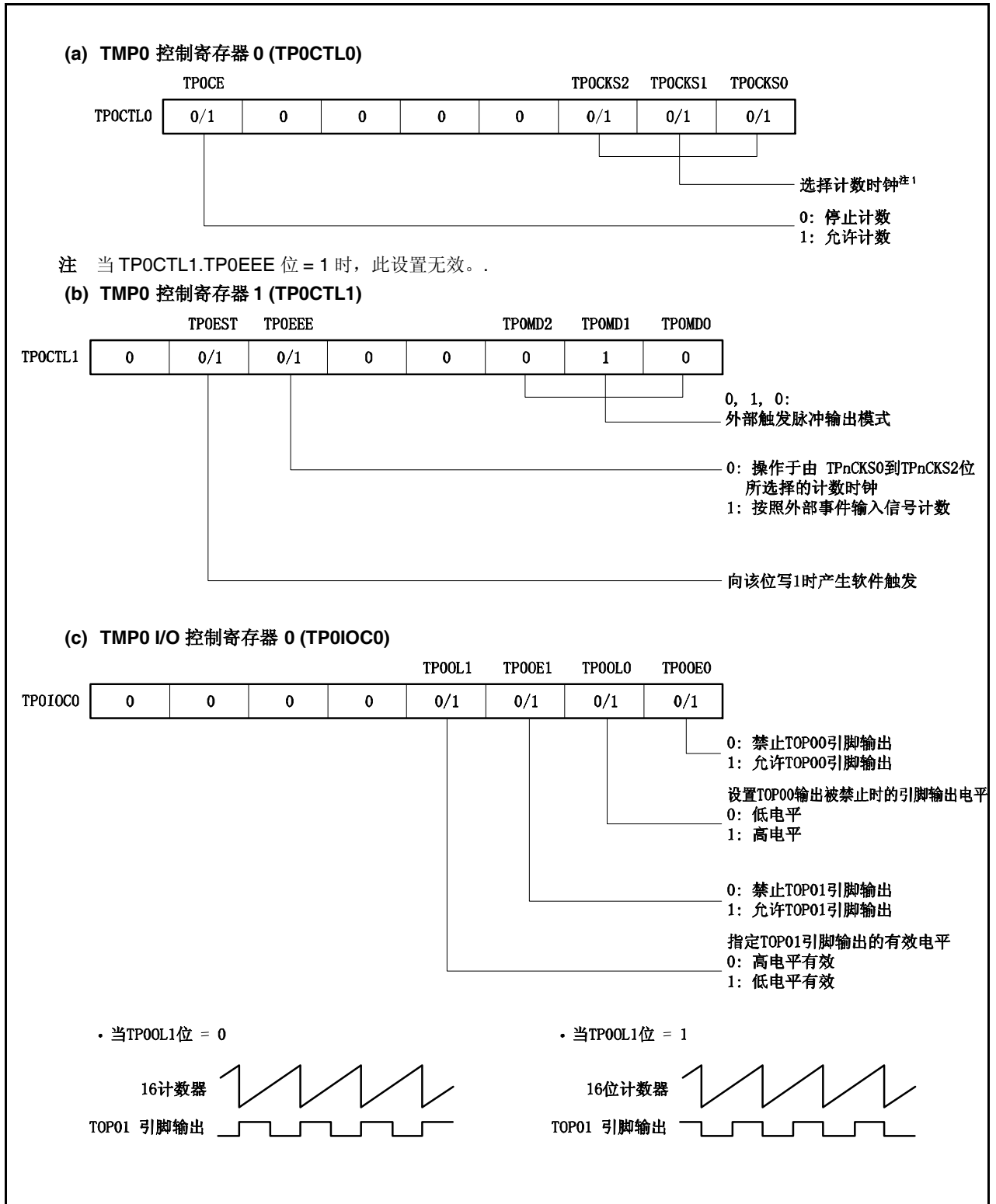
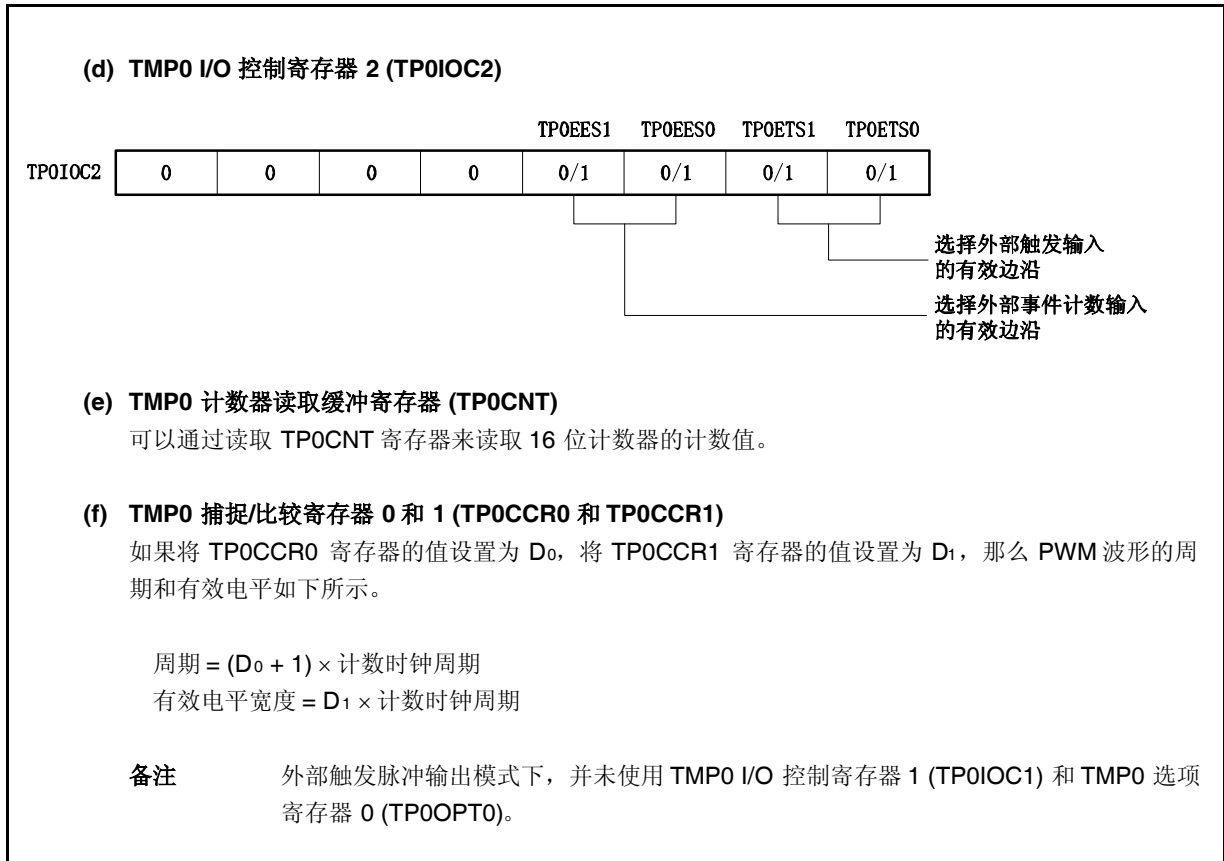


图 6-18. 外部触发脉冲输出模式的寄存器设置 (2/2)



(1) 外部触发脉冲输出模式的操作流程

图 6-19. 外部触发脉冲输出模式的软件处理流程 (1/2)

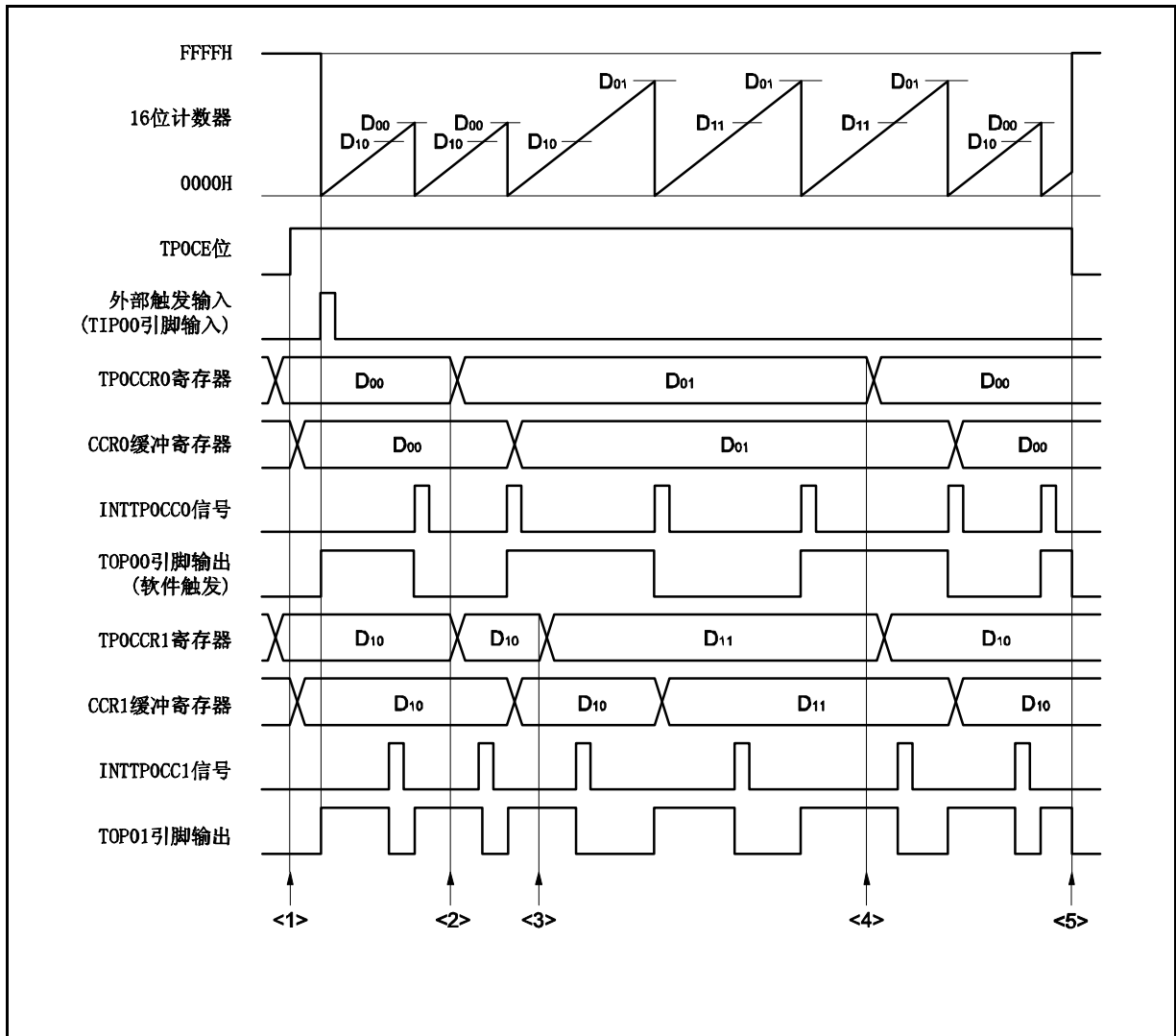
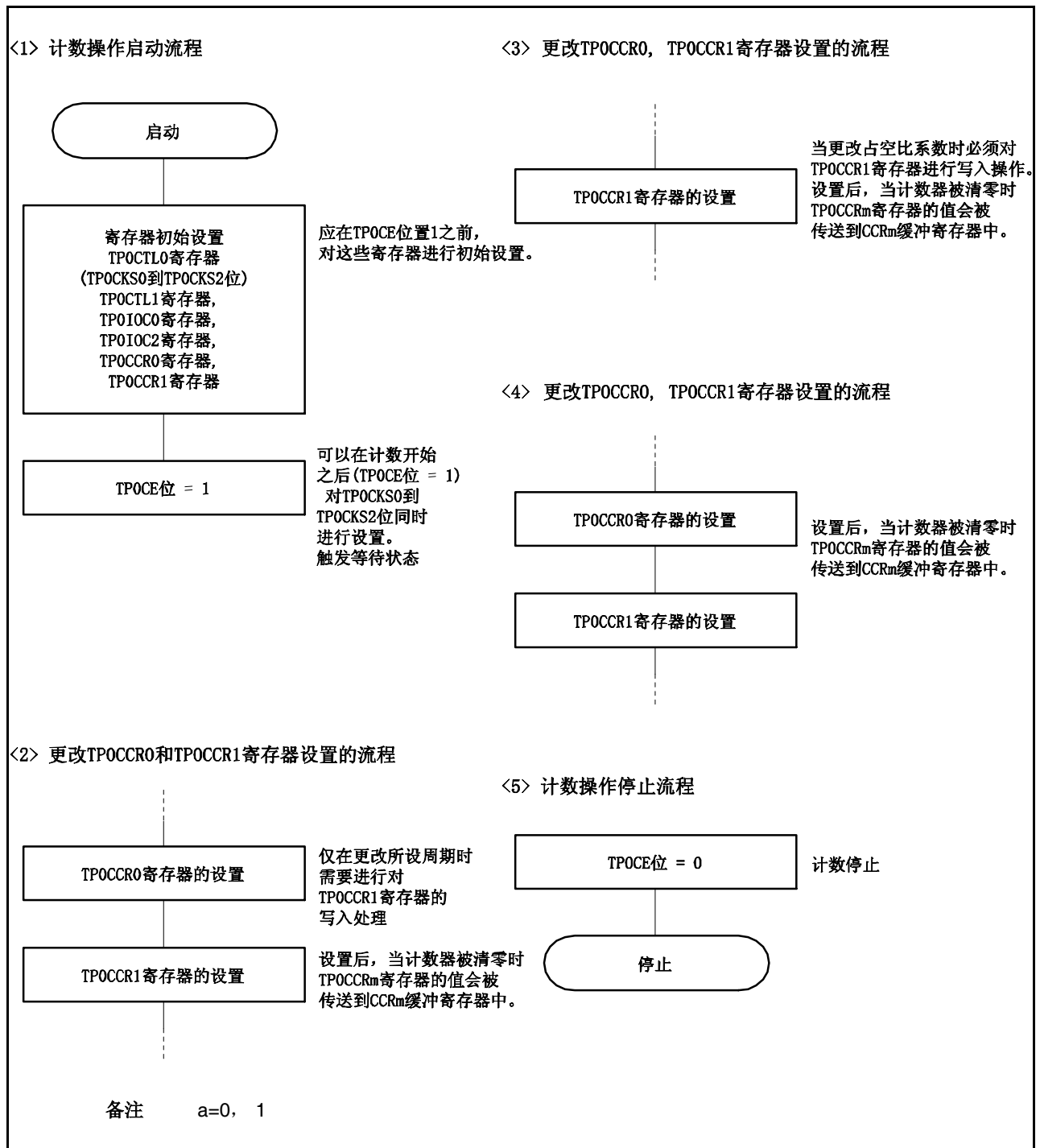


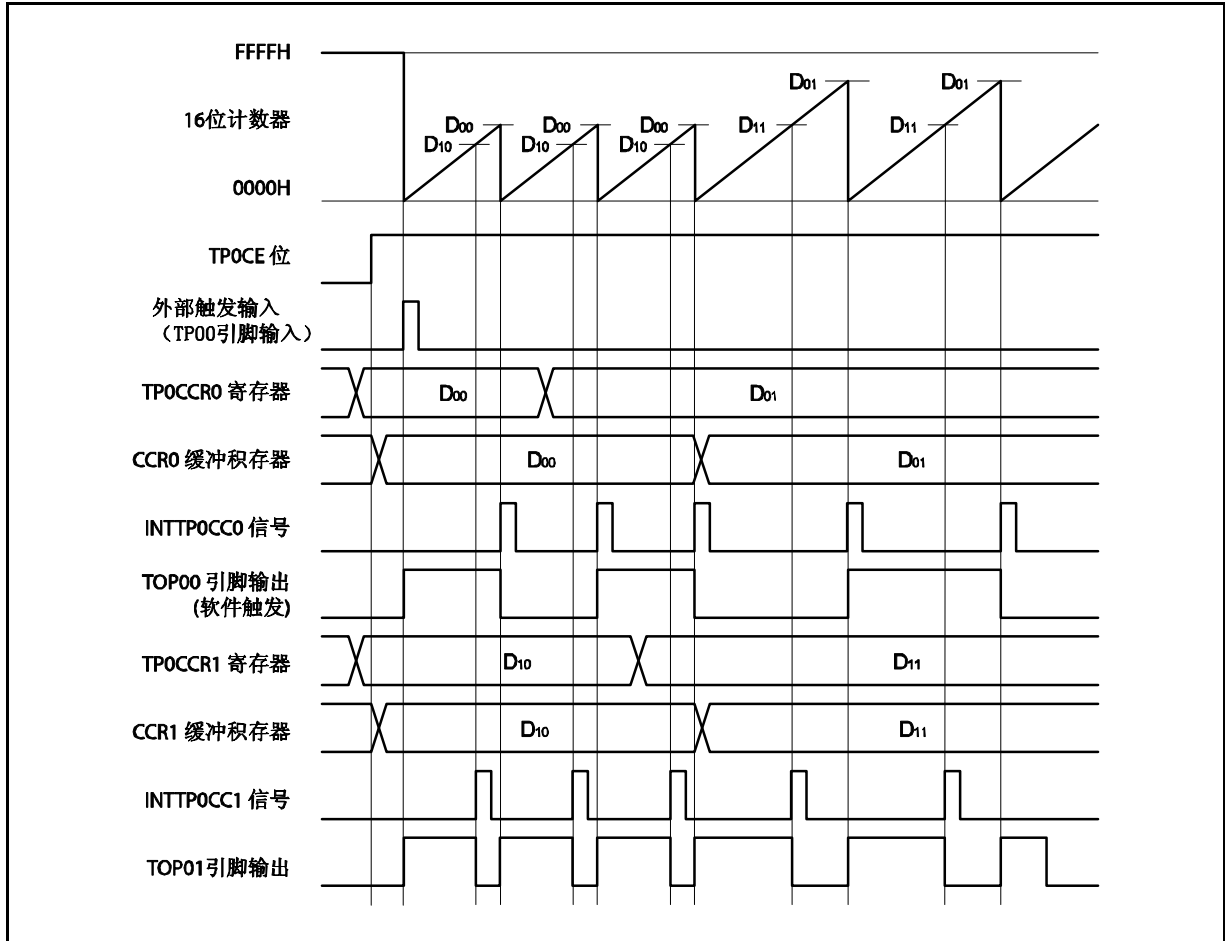
图 6 -19. 外部触发脉冲输出模式的软件处理流程 (2/2)



(2) 外部触发脉冲输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

计数器计数过程中，若改变 PWM 波形，则要在最后进行对 TP0CCR1 寄存器的写入操作。
对 TP0CCR1 寄存器进行写入操作之后，若需再次更改 TP0CCRm 寄存器的值，须等待下一个 INTTP0CC0 信号被检测到后，再进行相关操作。



若要将 TP0CCRa 寄存器中设置的数据传送至 CCRa 缓冲寄存器，必须对 TP0CCR1 寄存器进行写入操作。

同时改变 PWM 波形的周期和有效电平时，先将周期值写入 TP0CCR0 寄存器，再将有效电平宽度值写入 TP0CCR1 寄存器。

只改变 PWM 波形的周期时，先将周期值写入 TP0CCR0 寄存器，再向 TP0CCR1 寄存器写入与原来相同的值。

只改变 PWM 波形的有效电平宽度(占空比系数)时，只需重新设置 TP0CCR1 寄存器的值即可。

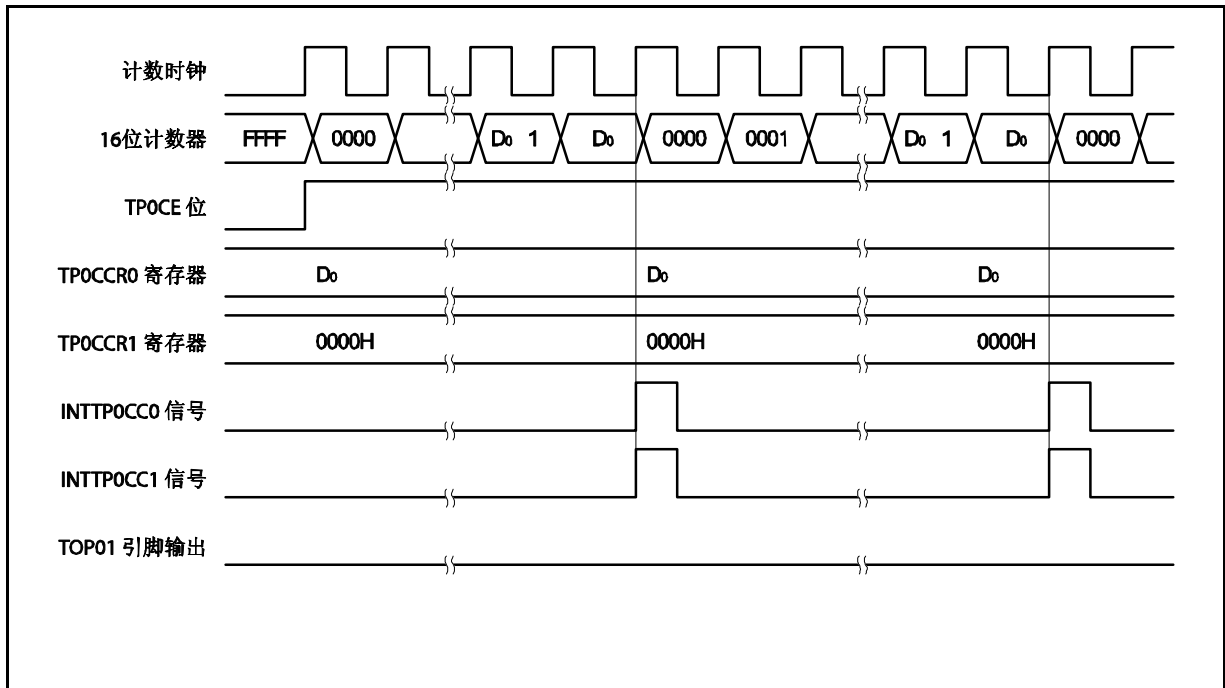
数据被写入 TP0CCR1 寄存器之后，当 16 位计数器被清零时，TP0CCRa 寄存器中的数值才会被传送至 CCRa 缓冲寄存器中。此时 CCRa 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

向 TP0CCR1 寄存器的写入操作完成后，若需再次改写 TP0CCR0 或 TP0CCR1 寄存器的值，则要在产生 INTTP0CC0 信号之后再进行改写。否则，由于数据从 TP0CCRa 寄存器传送至 CCRa 缓冲寄存器的时序与 TP0CCRa 寄存器的写入时序相冲突，CCRa 缓冲寄存器中所设置的数值可能无法确定。

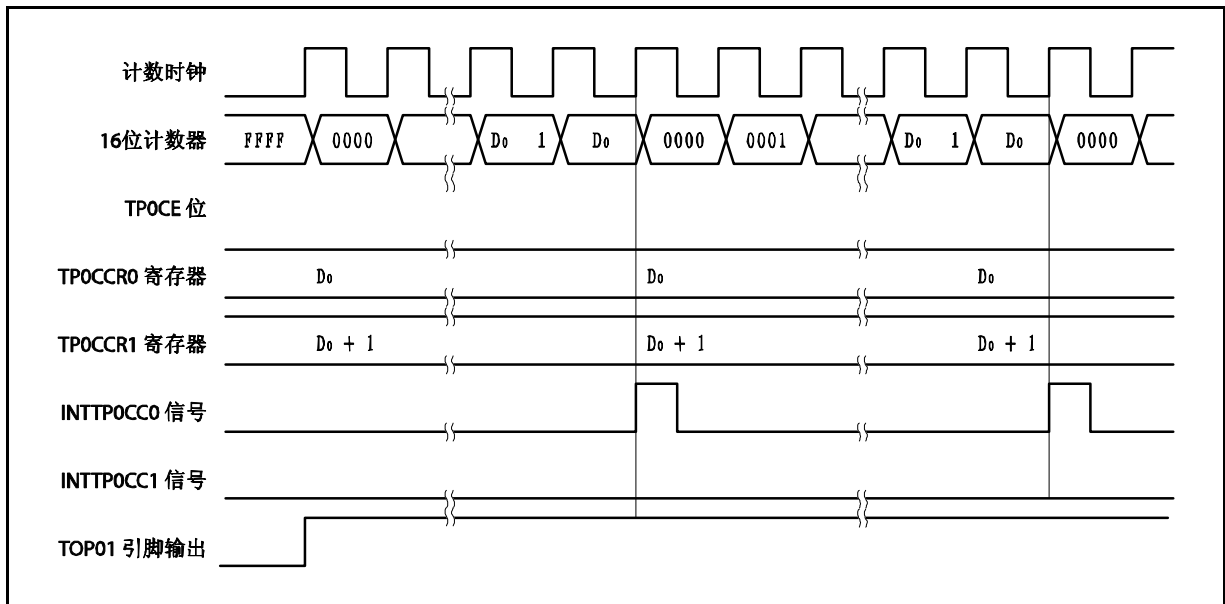
备注 a=0, 1

(b) PWM 波形的 0%/100% 输出

若要输出 0% 的波形，则需将 TP0CCR1 寄存器设置为 0000H。如果 TP0CCR0 寄存器被设置为 FFFFH，那么 INTTP0CC1 信号将周期性产生。

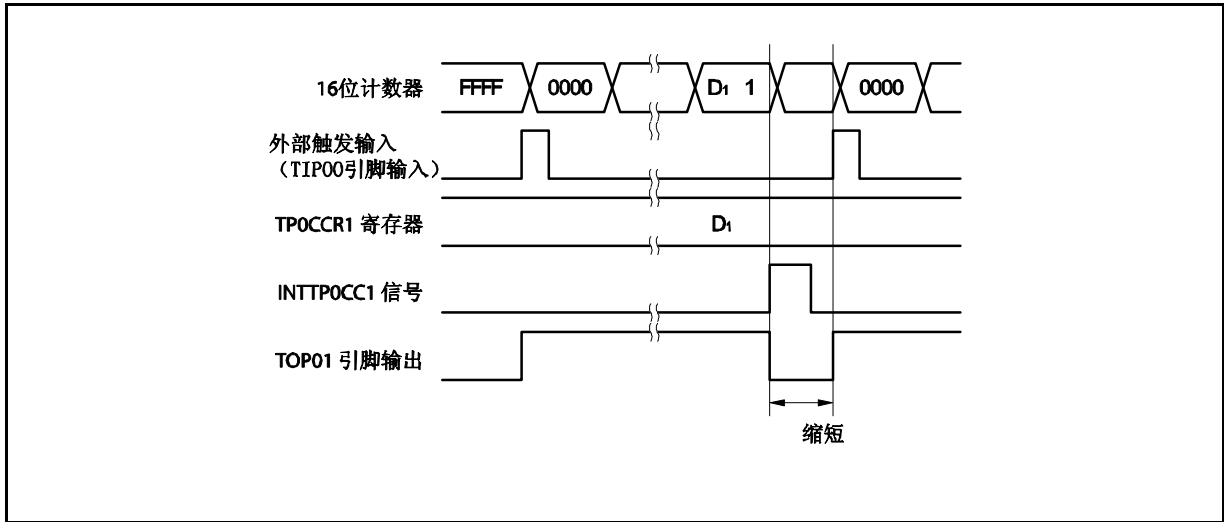


若要输出 100% 波形，则需将 TP0CCR1 寄存器设置为 (TP0CCR0 寄存器值 + 1)。TP0CCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。

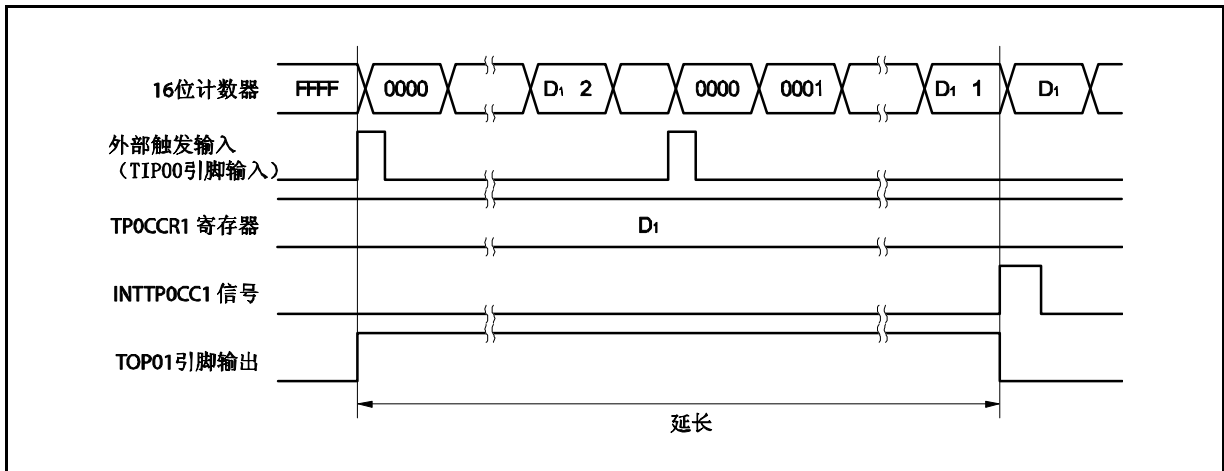


(c) 触发检测和 TP0CCR1 寄存器匹配之间的冲突

如果触发信号紧随 INTTP0CC1 中断信号的产生被检测，那么 16 位计数器将被立即清零并重新开始计数，TOP01 引脚输出也会改变为有效电平。这种情况下，PWM 波形的非有效电平时间就会被缩短。

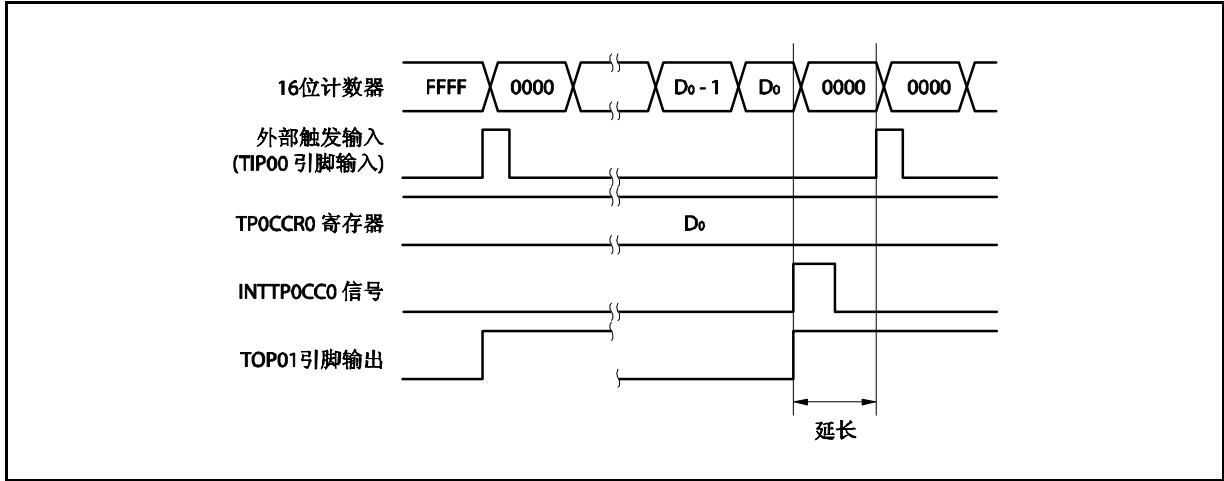


如果在 INTTP0CC1 信号产生之前检测到触发信号，那么 INTTP0CC1 信号将不会产生，此时，16 位计数器被清零并重新开始计数。TOP01 引脚的输出也会继续保持为有效电平。这种情况下，PWM 波形的有效电平时间就会被延长。

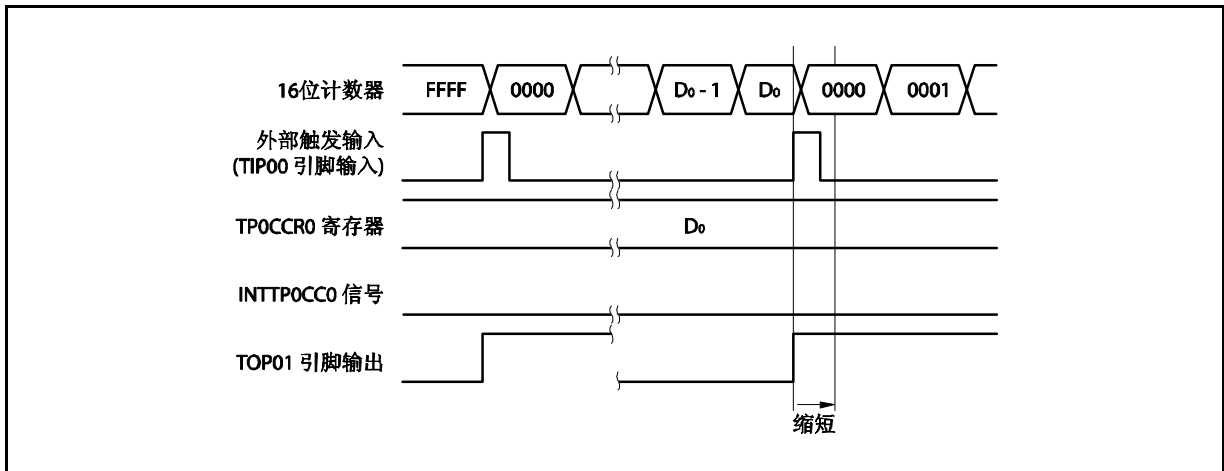


(d) 触发检测和 TP0CCR0 寄存器匹配之间的冲突

如果触发信号紧随 INTTP0CC0 中断信号的产生被检测，那么 16 位计数器将被清零并重新开始计数。因此，TOP01 引脚输出的有效电平时间就会被延长(从 INTTP0CC0 信号产生到触发信号被检测的时间)。

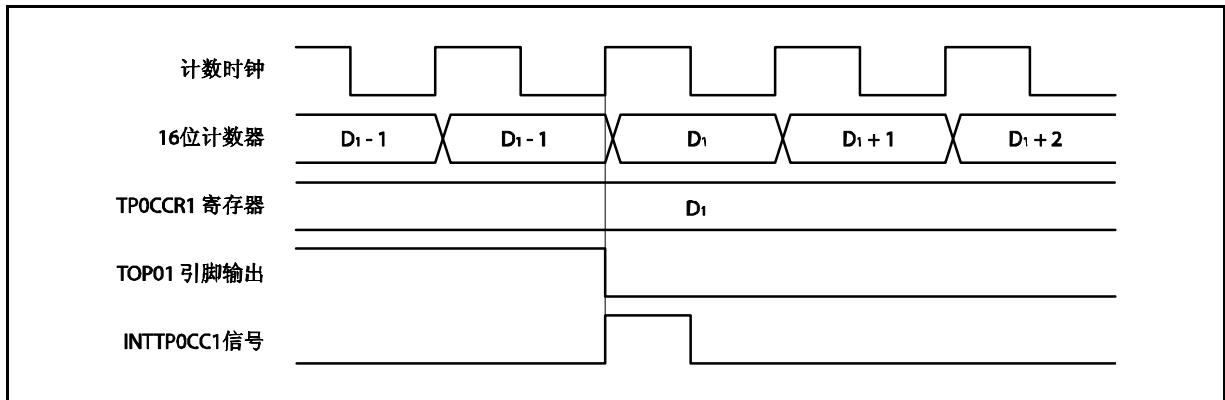


如果在 INTTP0CC0 信号产生之前检测到触发信号，那么 INTTP0CC0 信号将不会产生。16 位计数器将被清零并重新开始计数，TOP01 引脚输出也会改变为有效电平。这种情况下，PWM 波形的非有效电平时间就会被缩短。



(e) 比较匹配中断请求信号的产生时序 (INTTP0CC1)

外部触发脉冲输出模式下，INTTP0CC1 信号的产生时序与其它模式下 INTTP0CC1 信号的产生时序有所不同；INTTP0CC1 信号将在 16 位计数器的计数值与 TP0CCR1 寄存器的值相匹配时产生。



通常情况下，INTTP0CC1 信号会在 16 位计数器计数值与 TP0CCR1 寄存器的值匹配之后的下一个计数时钟同步产生。

但在外部触发脉冲输出模式下，INTTP0CC1 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOP01 引脚输出信号变化的时序而改变的。

6.5.4 单脉冲输出模式 (TP0MD2 ~ TP0MD0 位 = 011)

在单脉冲输出模式中，TPOCTL0.TPOCE 位被置 1 后，16 位定时器/事件计数器 P 进入等待触发信号的状态。当检测到一个外部触发输入信号的有效沿时，16 位定时器/事件计数器 P 开始计数，并由 TOP01 引脚输出一个单脉冲波形。

也可以使用软件触发代替外部触发来开始脉冲的输出。当使用软件触发时，TOP00 引脚会在 16 位计数器计数时输出有效电平，在计数器停止计数时(等待触发时)输出非有效电平。

图 6-20. 单脉冲输出模式的配置图

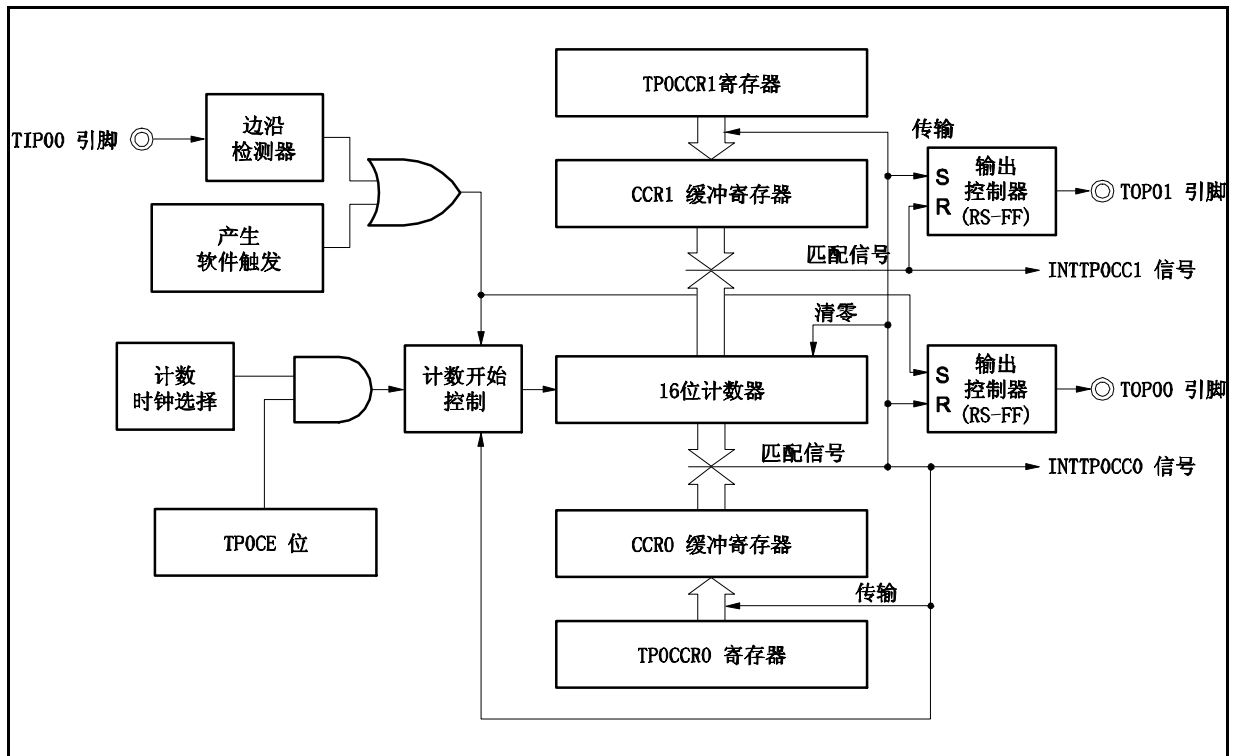
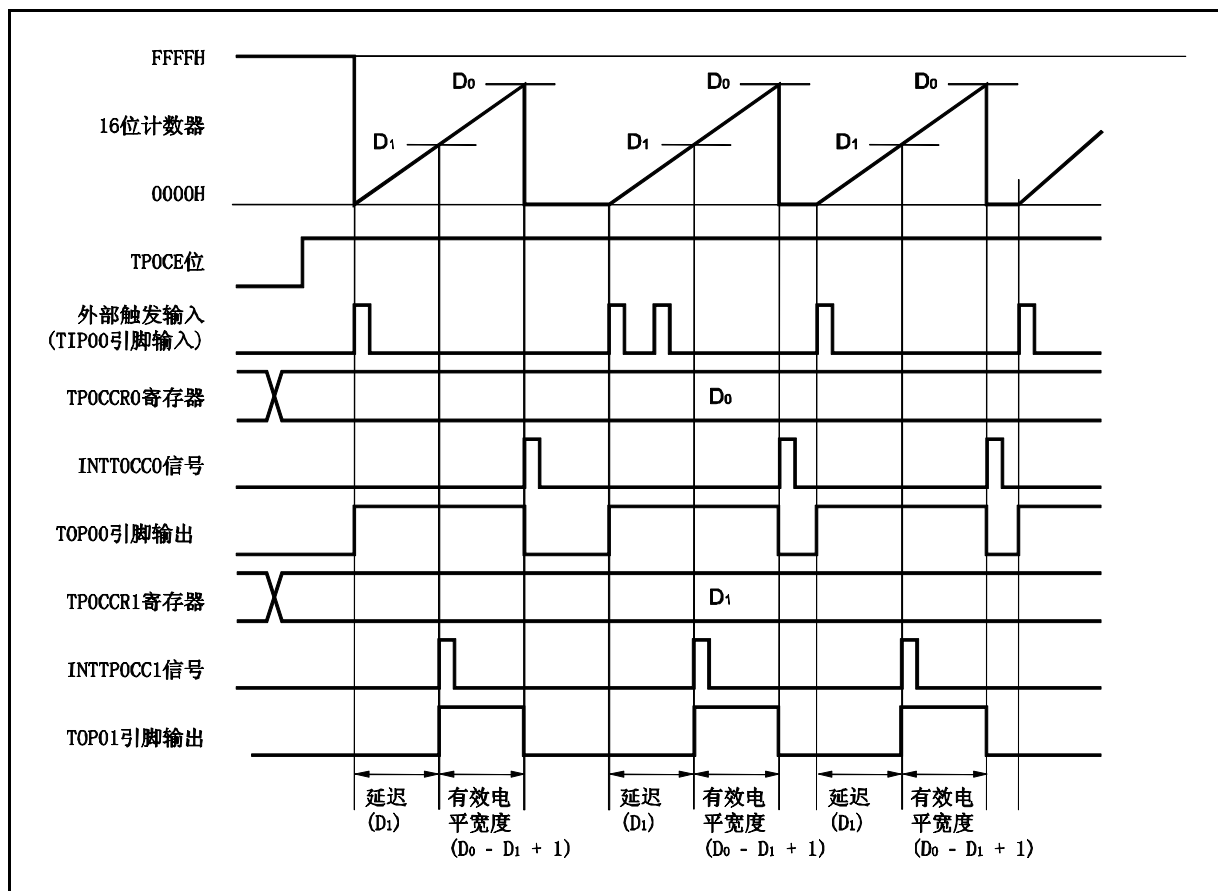


图 6-21. 单脉冲输出模式的基本时序



TPOCTL0.TPOCE 位被置 1 后，16 位定时器/事件计数器 P 进入等待触发信号的状态。触发被检测后，16 位计数器从 FFFFH 清零为 0000H，同时开始计数并从 TOP01 引脚输出一个单脉冲波形。单脉冲波形输出后，16 位计数器将被重置为 FFFFH 并停止计数，16 位定时器/时间计数器 P 重新进入等待触发信号的状态。如果在单脉冲波形输出过程中再次检测到触发信号，那么这个触发信号将被忽略。

单脉冲波形的输出延迟时间以及有效电平宽度可由以下算式得出。

输出延迟时间 = (TPOCCR1 寄存器的设置值) × 计数时钟周期

有效电平宽度 = (TPOCCR0 寄存器的值 - TPOCCR1 寄存器的值 + 1) × 计数时钟周期

比较匹配中断请求信号 INTTPOCC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器中的值相匹配时的下一个计数时钟时产生。比较匹配中断请求信号 INTTPOCC1 会在 16 位计数器的计数值与 CCR1 缓冲寄存器中的值相匹配时产生。

触发源可以从外部触发输入信号的有效沿和设置软件触发(TPOCTL1.TPOEST 位 = 1)的方式中进行选择。

图 6-22. 单脉冲输出模式的寄存器设置 (1/2)

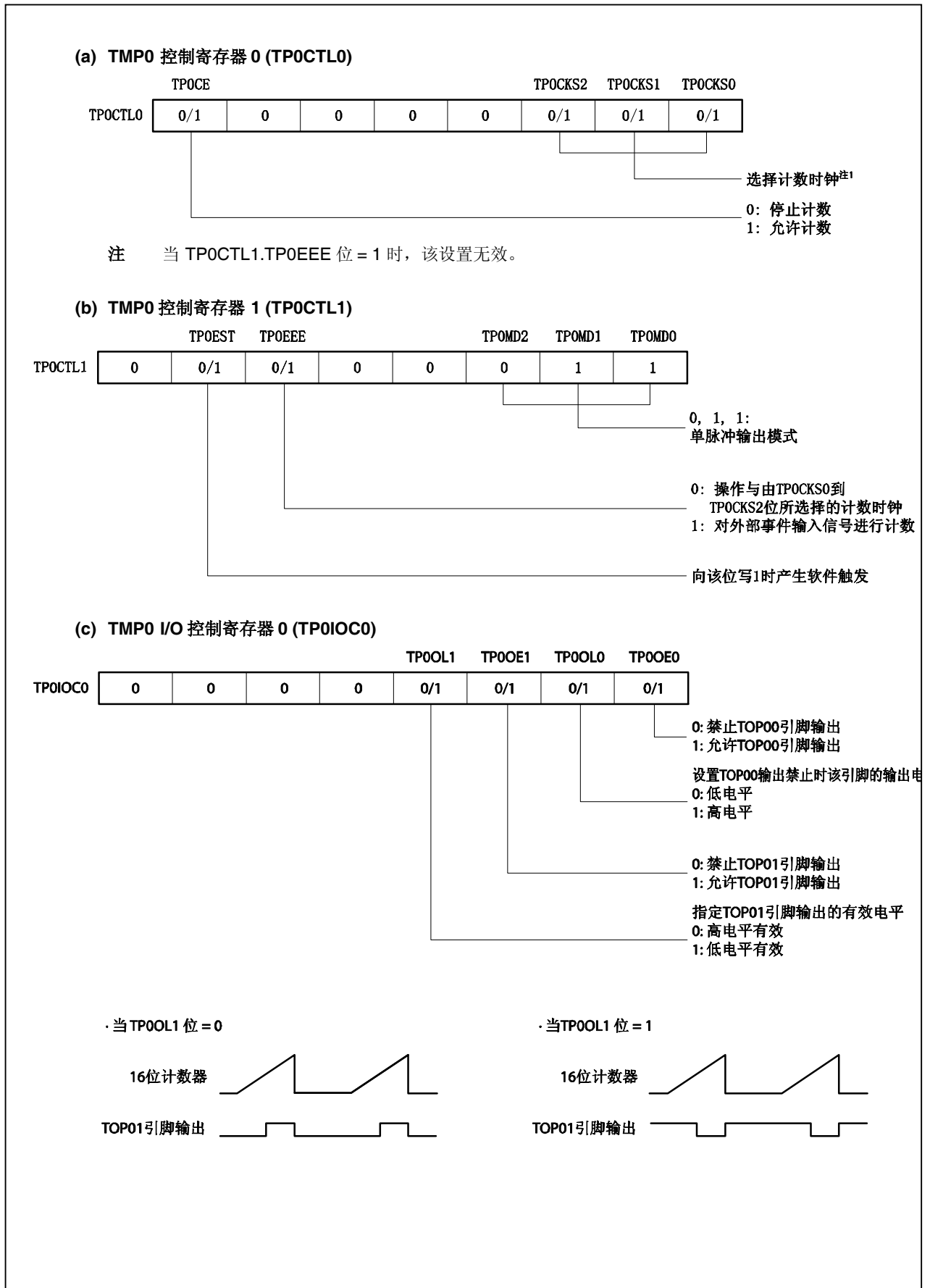
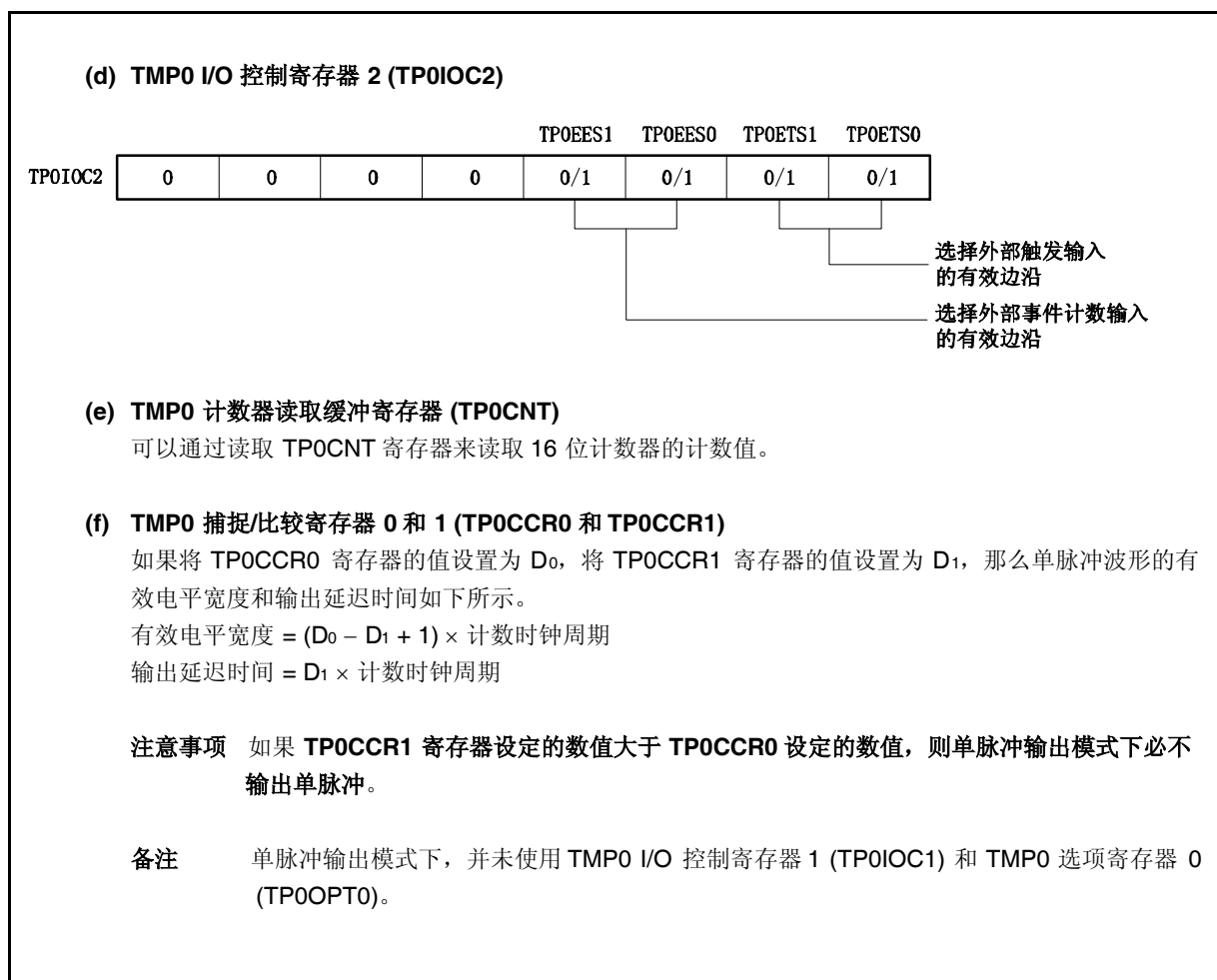
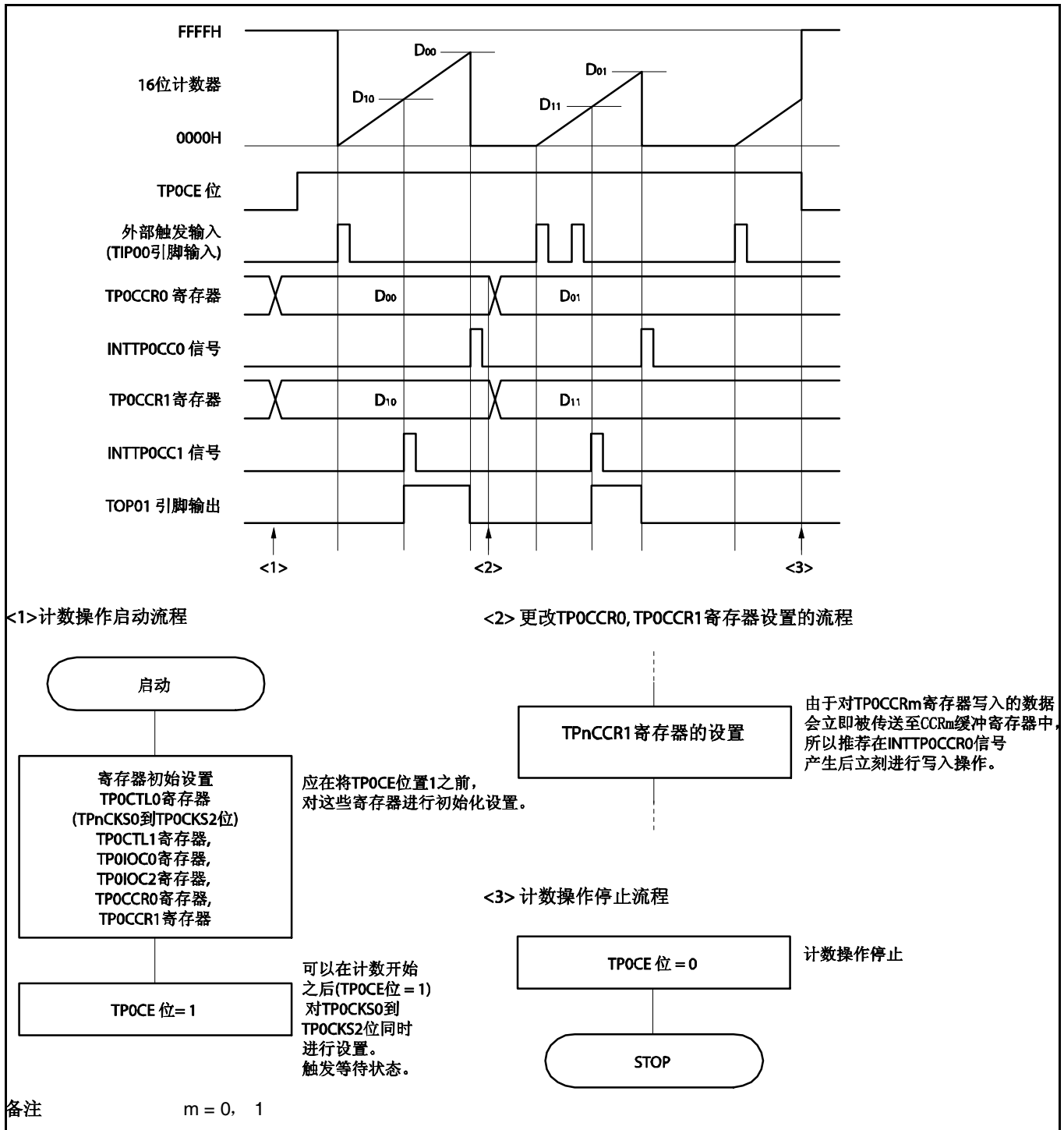


图 6-22. 单脉冲输出模式的寄存器设置 (2/2)



(1) 单脉冲输出模式的操作流程

图 6-23. 单脉冲输出模式的软件操作流程

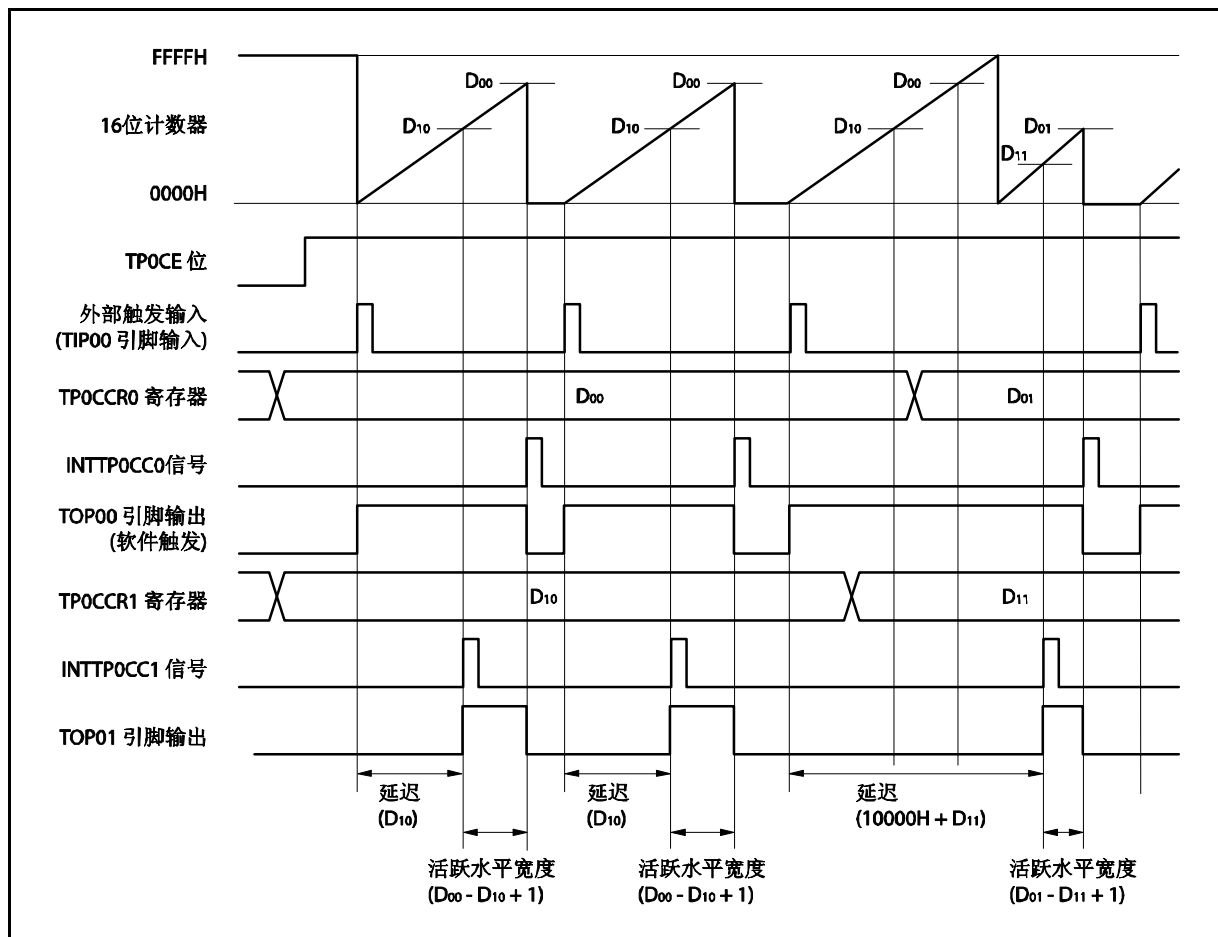


(2) 单脉冲输出模式的操作时序

(a) 改写 TP0CCRaMa 寄存器时需要注意的事项

将 TP0CCRaMa 寄存器的值减小时，要先停止计数，再对设定值进行改变。

如果在计数过程中将 TP0CCRaMa 寄存器的值减小，那么 16 位计数器可能产生溢出。



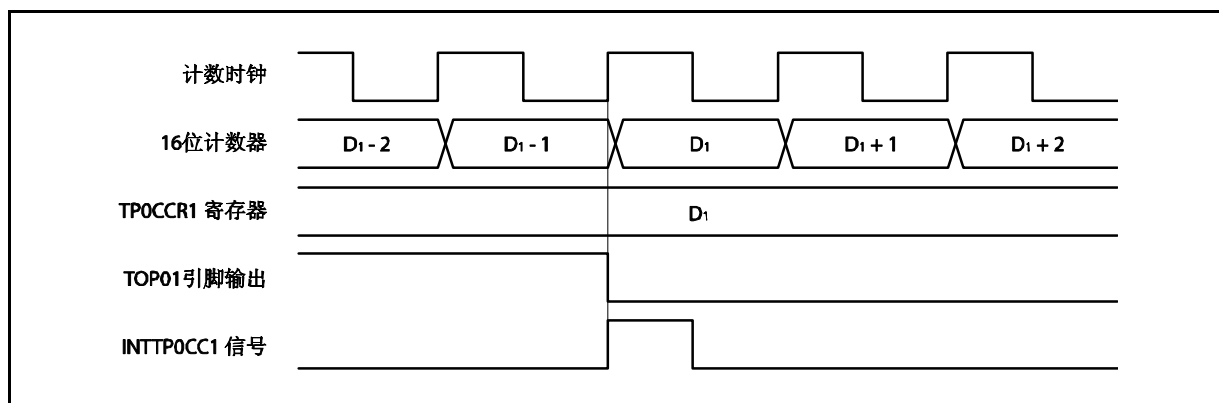
当将 TP0CCR0 寄存器的值从 D₀₀ 改写为 D₀₁，TP0CCR1 寄存器的值从 D₁₀ 改写为 D₁₁ 时，这里假设 D₀₀ > D₀₁ 且 D₁₀ > D₁₁，如果 TP0CCR1 寄存器被改写时，16 位计数器的计数值大于 D₁₁ 且小于 D₁₀，并且 TP0CCR0 寄存器被改写时，计数器的计数值大于 D₀₁ 且小于 D₀₀，两个设定值都将在改写后立即生效并与计数值进行比较。此时计数器会计数到 FFFFH 然后从 0000H 重新开始计数。当计数值与 D₁₁ 匹配时，将产生 INTTP0CC1 信号，同时 TOP01 引脚电平变为有效电平。当计数值与 D₀₁ 匹配时，将产生 INTTP0CC0 信号，同时 TOP01 引脚电平变为非有效电平并且计数器停止计数。

因此可见，此时定时器输出的单脉冲波形的延迟时间或有效电平宽度会与预期数值有所差异。

备注 a = 0, 1

(b) 比较匹配中断请求信号的产生时序 (INTTP0CC1)

单脉冲输出模式下，INTTP0CC1 信号的产生时序与其它模式下 INTTP0CC1 信号的产生时序有所不同。这里，INTTP0CC1 信号将在 16 位计数器的计数值与 TP0CCR1 寄存器的值相匹配时产生。



通常情况下，INTTP0CC1 信号会在 16 位计数器计数值与 TP0CCR1 寄存器的值匹配之后的下一个计数时钟同步产生。

但在单脉冲输出模式下，INTTP0CC1 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOP01 引脚输出信号变化的时序而改变的。

6.5.5 PWM 输出模式 (TP0MD2 ~ TP0MD0 位 = 100)

PWM 输出模式下，TPOCTL0.TPOCE 位被置 1 后，TOP01 引脚将输出一个 PWM 波形。
另外，TOP00 引脚将输出一个以 PWM 波形周期的 2 倍为周期的脉冲信号。

图 6-24. PWM 输出模式的配置图

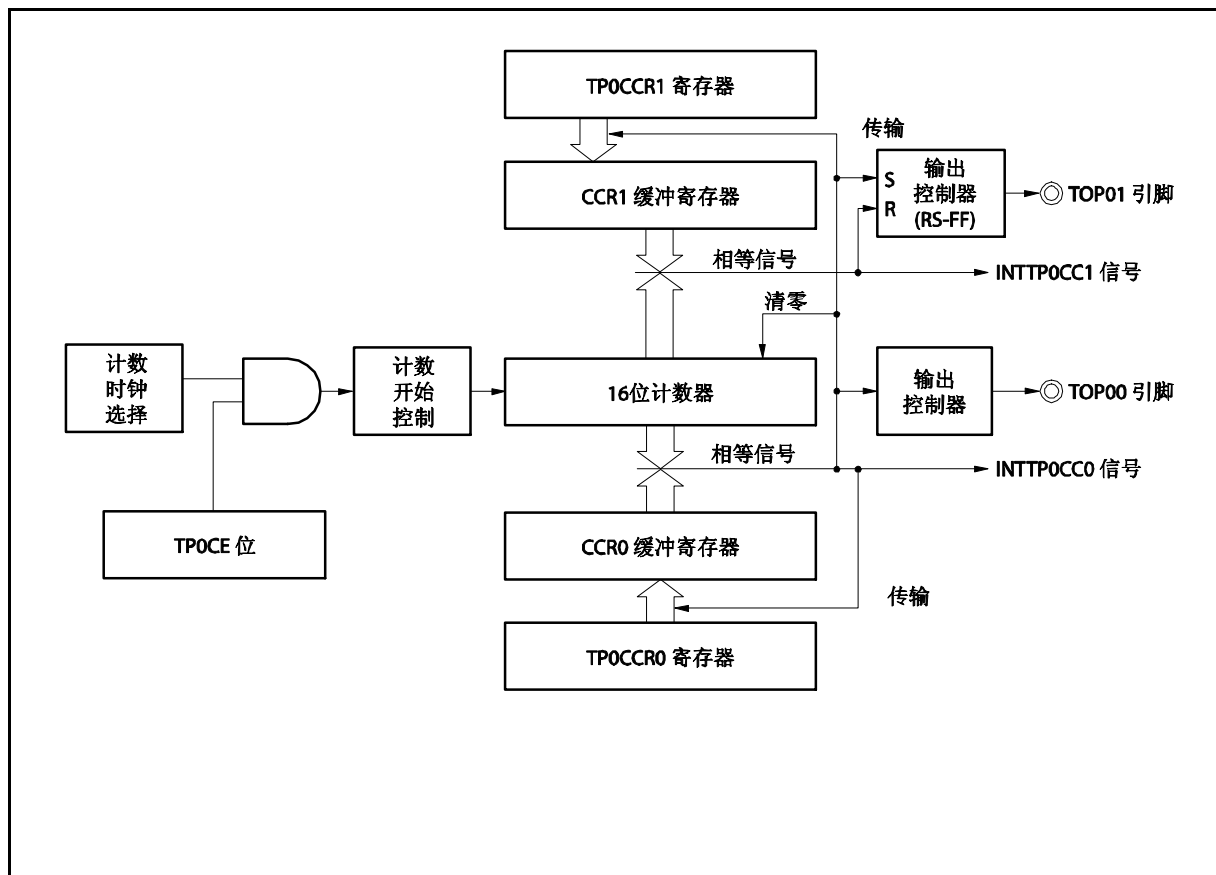
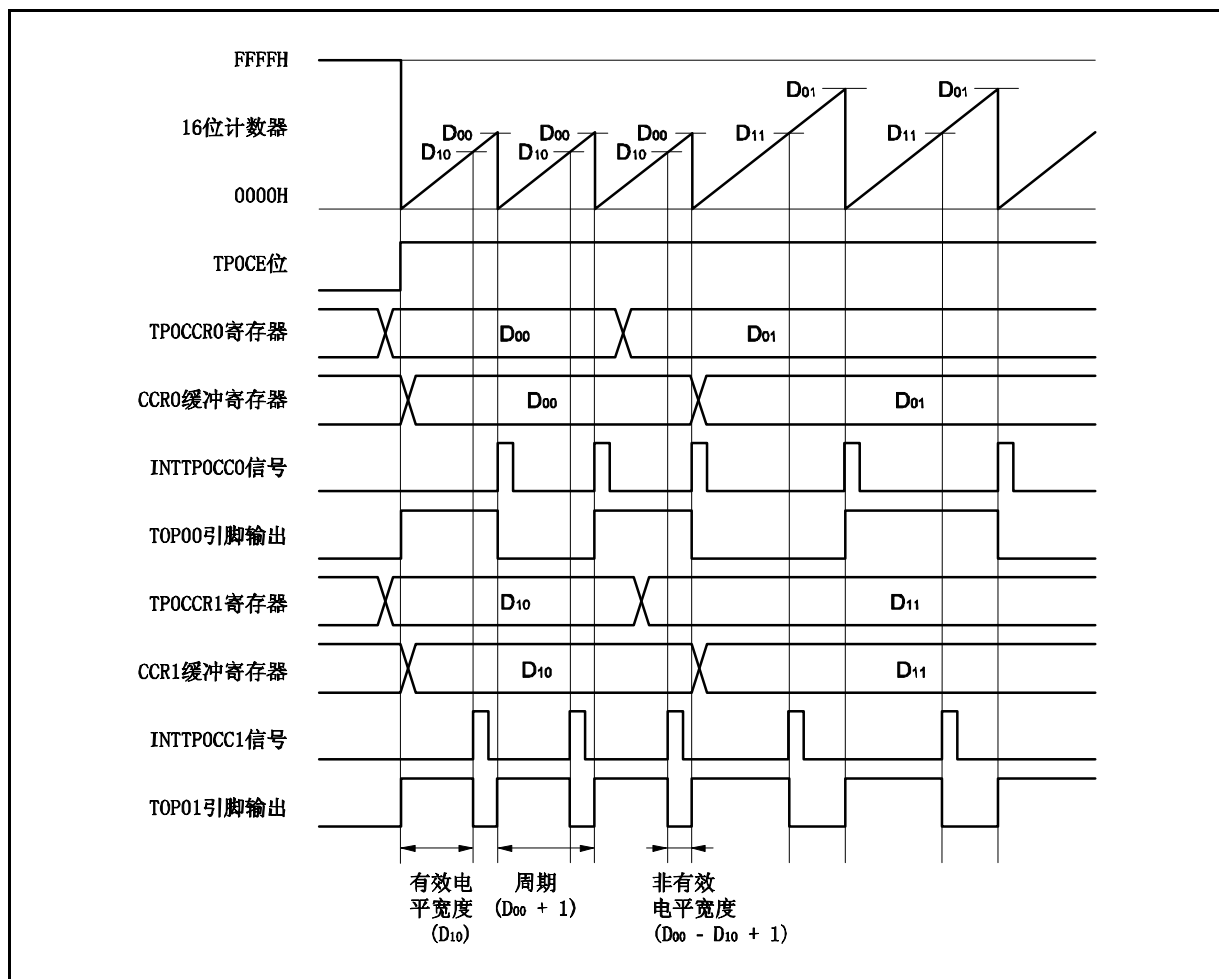


图 6-25. PWM 输出模式的基本时序



TPOCE 位被置 1 后，16 位计数器从 FFFFH 清零为 0000H 并开始计数，同时从 TOP01 引脚输出一个 PWM 波形。

PWM 波形的有效电平宽度，周期以及占空比系数可由以下算式得出。

有效电平宽度 = (TPOCCR1 寄存器的值) × 计数时钟周期

周期 = (TPOCCR0 寄存器的值 + 1) × 计数时钟周期

占空比系数 = (TPOCCR1 寄存器的值)/(TPOCCR0 寄存器的值 + 1)

可以在计数器计数过程中，通过改写 TPOCCR_a 寄存器的值来改变 PWM 波形的参数。改写后的数值将在 16 位计数器的计数值与当前 CCR₀ 缓冲寄存器的值匹配之后生效，同时 16 位计数器也会被清零。

比较匹配中断请求信号 INTTPOCC₀ 会在 16 位计数器的计数值与 CCR₀ 缓冲寄存器的值相匹配后的下一个计数时钟时产生，同时 16 位计数器将被清零。比较匹配中断请求信号 INTTPOCC₁ 会在 16 位计数器的计数值与 CCR₁ 缓冲寄存器的值相匹配时产生。

当 16 位计数器的计数值与 CCR_a 缓冲寄存器中的值相匹配时，TPOCCR_a 寄存器中设置的值会被传送至 CCR_a 缓冲寄存器，同时 16 位计数器被清零。

备注 a=0, 1

图 6-26. PWM 输出模式的寄存器设置 (1/2)

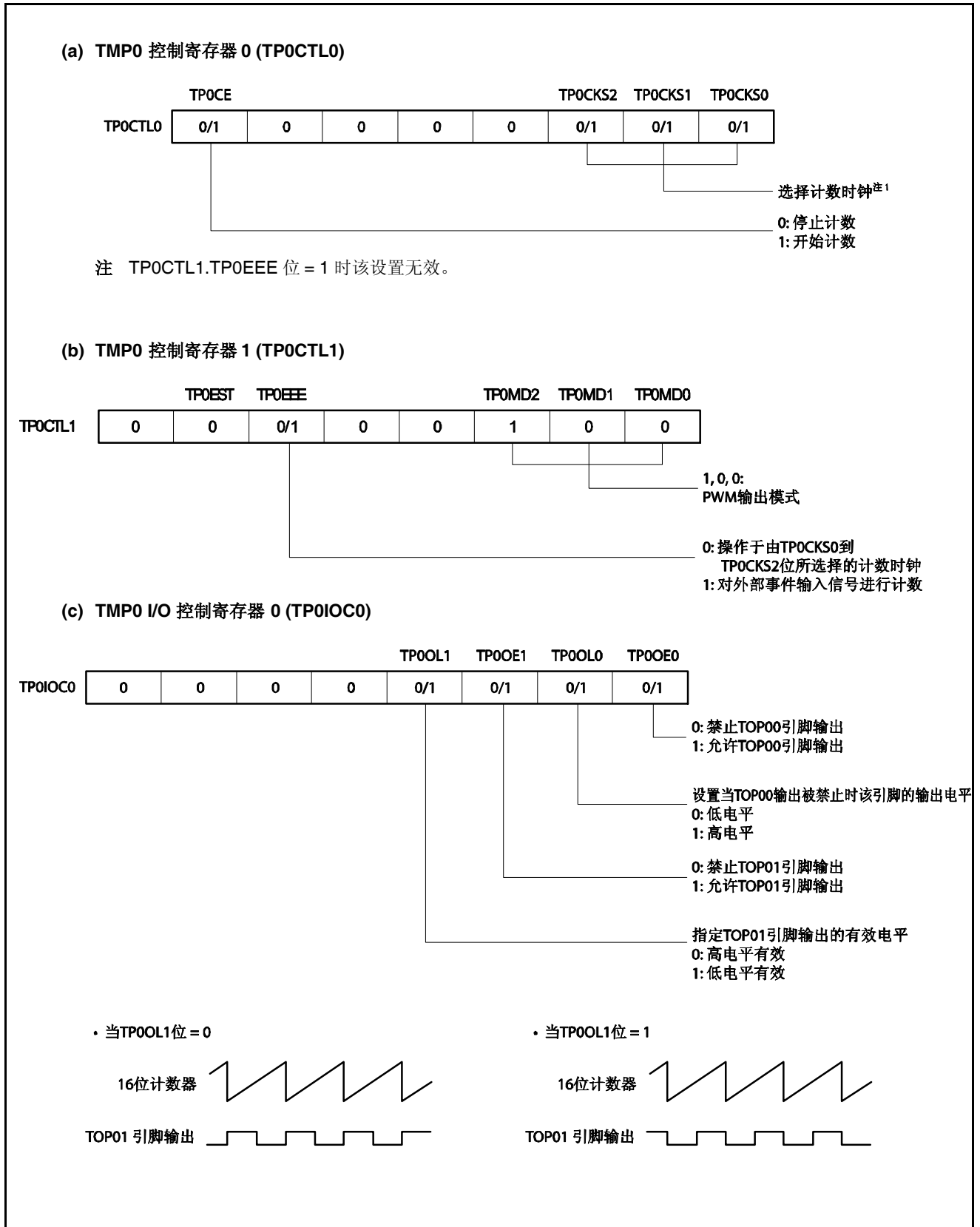
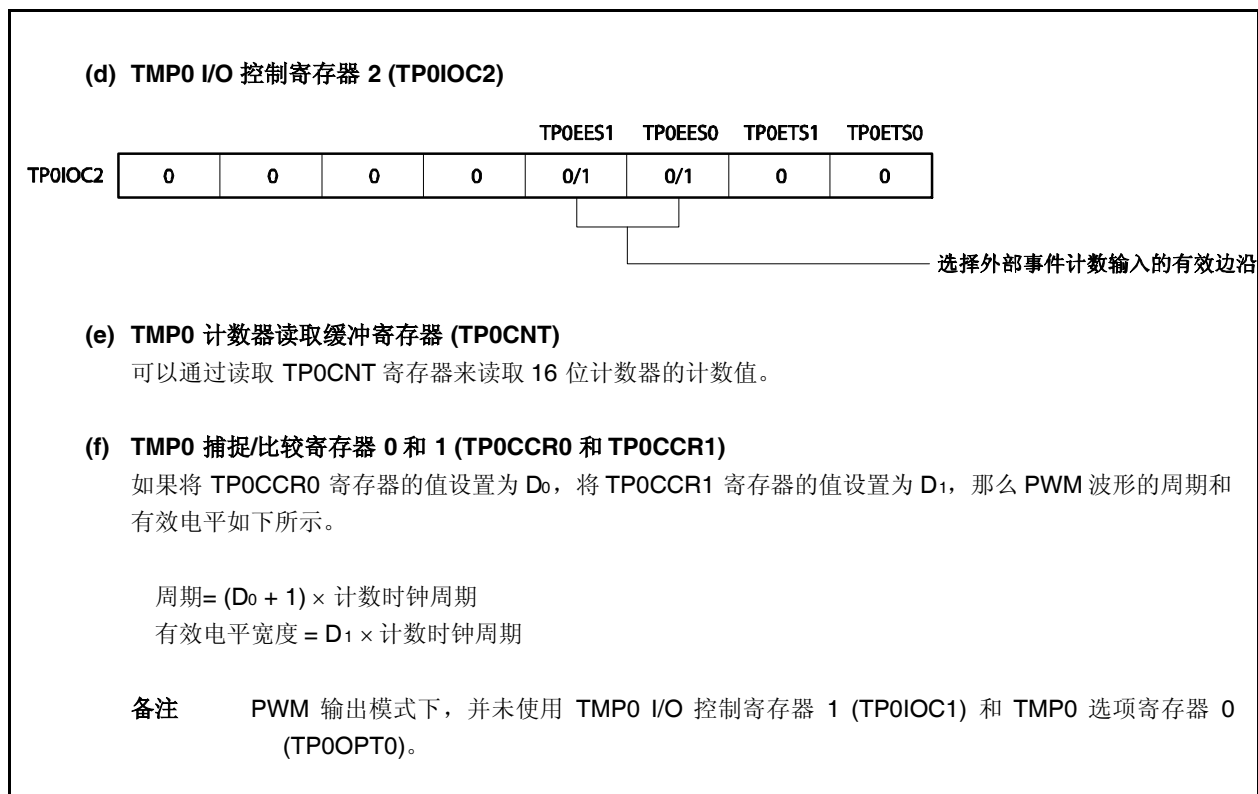


图 6-26. PWM 输出模式的寄存器设置 (2/2)



(1) PWM 输出模式的操作系统

图 6-27. PWM 输出模式的软件处理流程 (1/2)

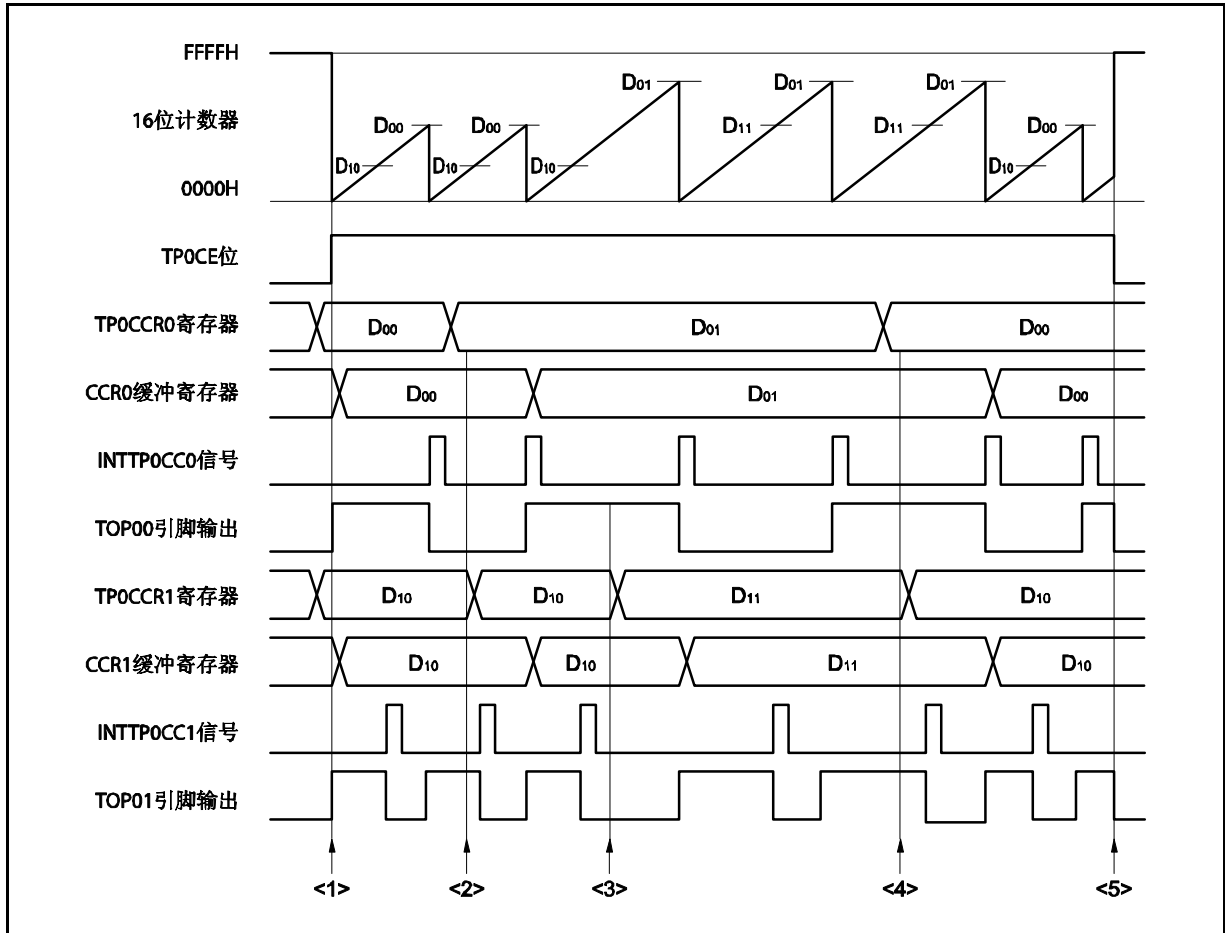
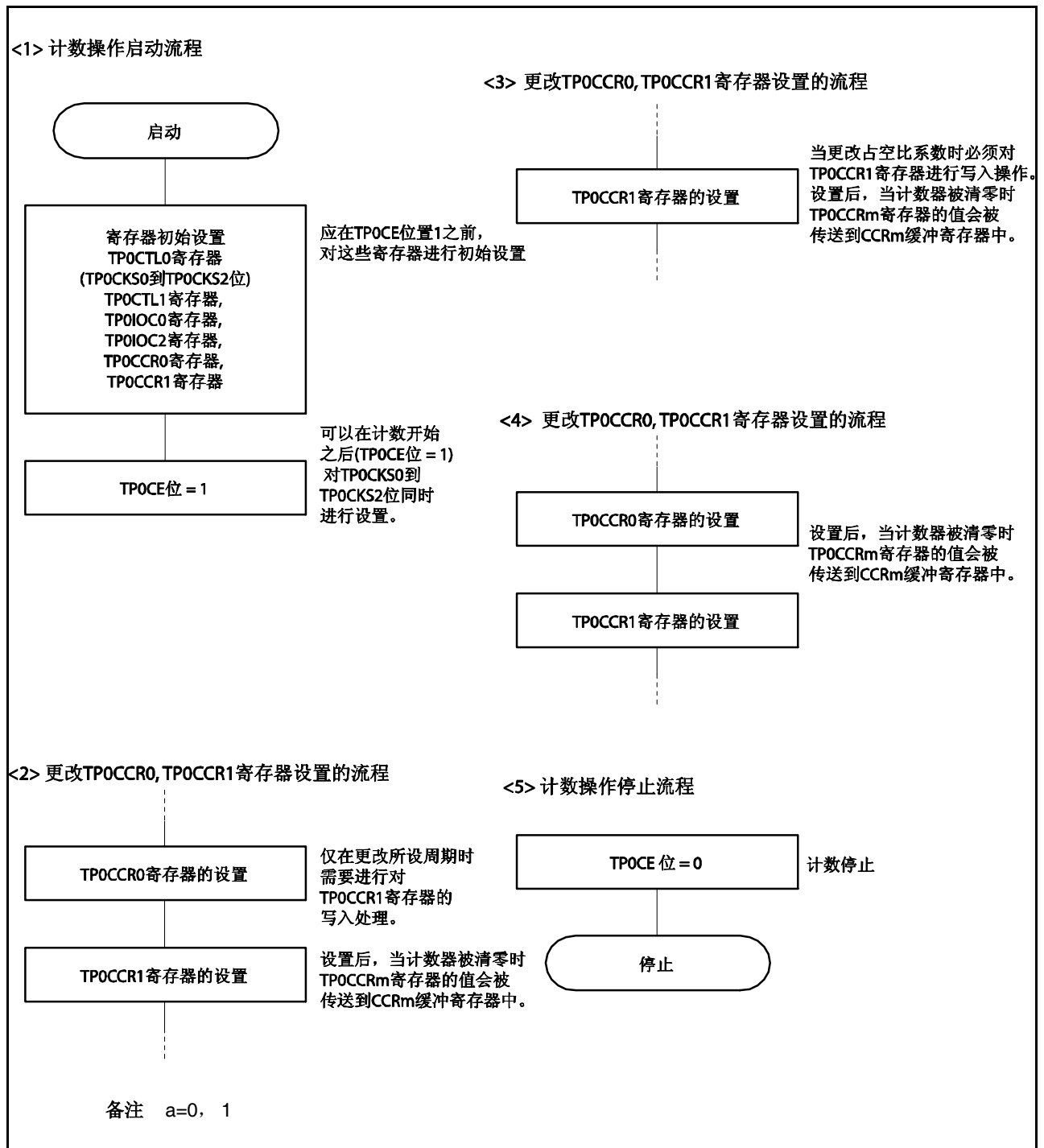


图 6-27. PWM 输出模式的软件处理流程 (2/2)

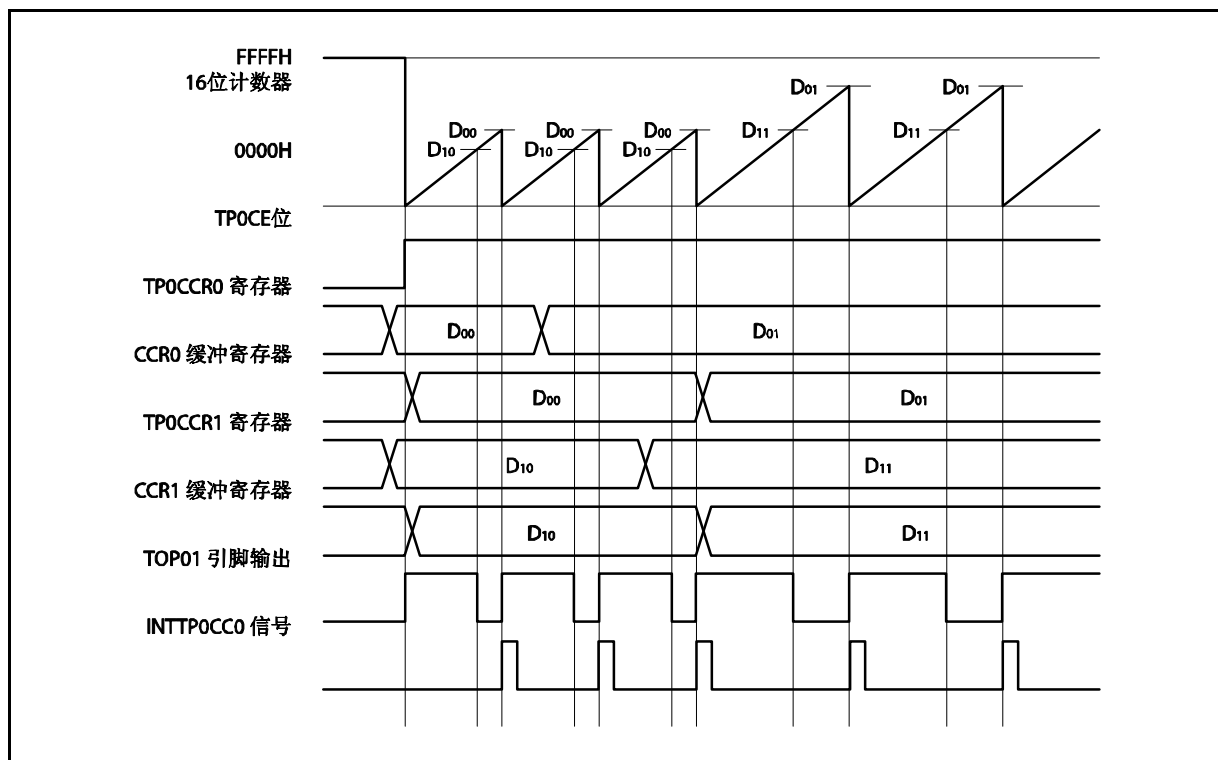


(2) PWM 输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

计数器计数过程中，若改变 PWM 波形，则要在最后进行对 TP0CCR1 寄存器的写入操作。

对 TP0CCR1 寄存器进行写入操作之后，若需再次更改 TP0CCRa 寄存器的值，须等待下一个 INTTP0CC1 信号被检测到后，再进行相关操作。



若要将 TP0CCRa 寄存器中设置的数据传送至 CCRa 缓冲寄存器，必须对 TP0CCR1 寄存器进行写入操作。

同时改变 PWM 波形的周期和有效电平时，先将周期值写入 TP0CCR0 寄存器，再将有效电平宽度值写入 TP0CCR1 寄存器。

只改变 PWM 波形的周期时，先将周期值写入 TP0CCR0 寄存器，再向 TP0CCR1 寄存器写入与原来相同的值。

只改变 PWM 波形的有效电平宽度(占空比系数)时，只需重新设置 TP0CCR1 寄存器的值即可。

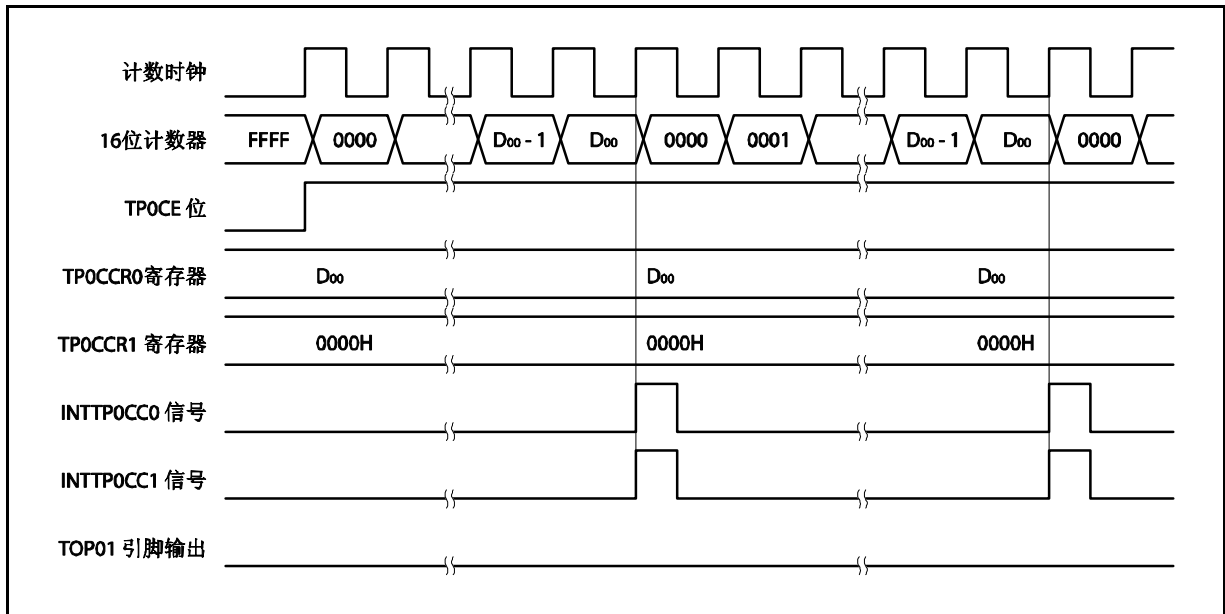
数据被写入 TP0CCR1 寄存器之后，16 位计数器被清零时，TP0CCRa 寄存器中的数值才会被同步传送至 CCRa 缓冲寄存器中，此时 CCRa 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

向 TP0CCR1 寄存器的写入操作完成后，若需再次改写 TP0CCR0 或 TP0CCR1 寄存器的值，则要在产生 INTTP0CC0 信号之后再进行改写。否则，由于数据从 TP0CCRa 寄存器传送至 CCRa 缓冲寄存器的时刻与 TP0CCRa 寄存器的写入时刻相冲突，CCRa 缓冲寄存器中所设置的数值可能无法确定。

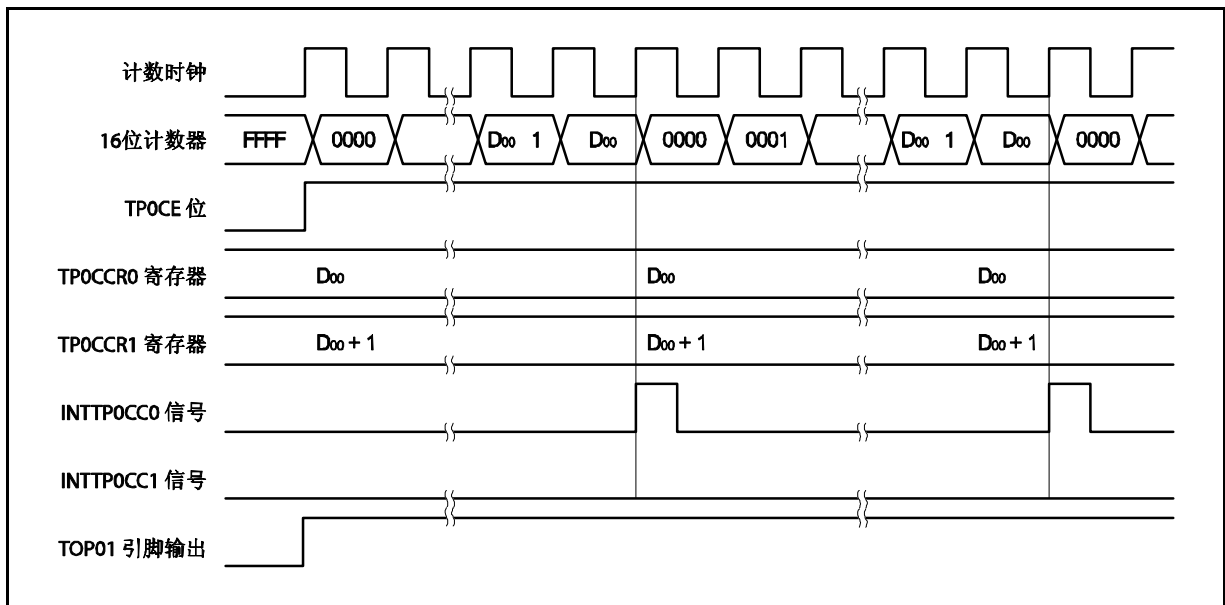
备注 a = 0, 1

(b) PWM 波形的 0%/100% 输出

若要输出 0% 的波形，则需将 TP0CCR1 寄存器设置为 0000H。如果 TP0CCR0 寄存器被设置为 FFFFH，那么 INTTP0CC1 信号将周期性产生。

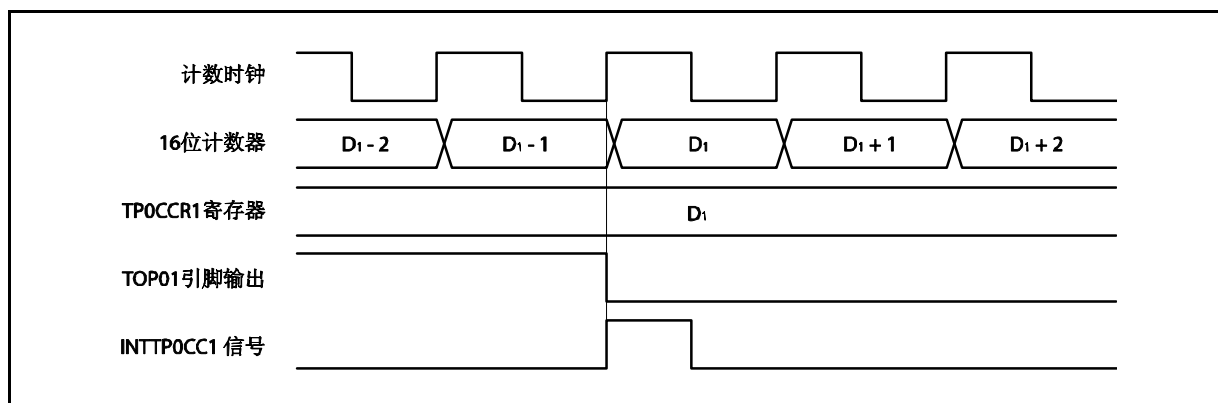


若要输出 100% 波形，则需将 TP0CCR1 寄存器设置为 (TP0CCR0 寄存器值 + 1)。TP0CCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。



(c) 比较匹配中断请求信号的产生时序 (INTTP0CC1)

PWM 输出模式下，INTTP0CC1 信号的产生时序与其它模式下 INTTP0CC1 信号的产生时序有所不同，这里，INTTP0CC1 信号将在 16 位计数器的计数值与 TP0CCR1 寄存器的值相匹配时产生。



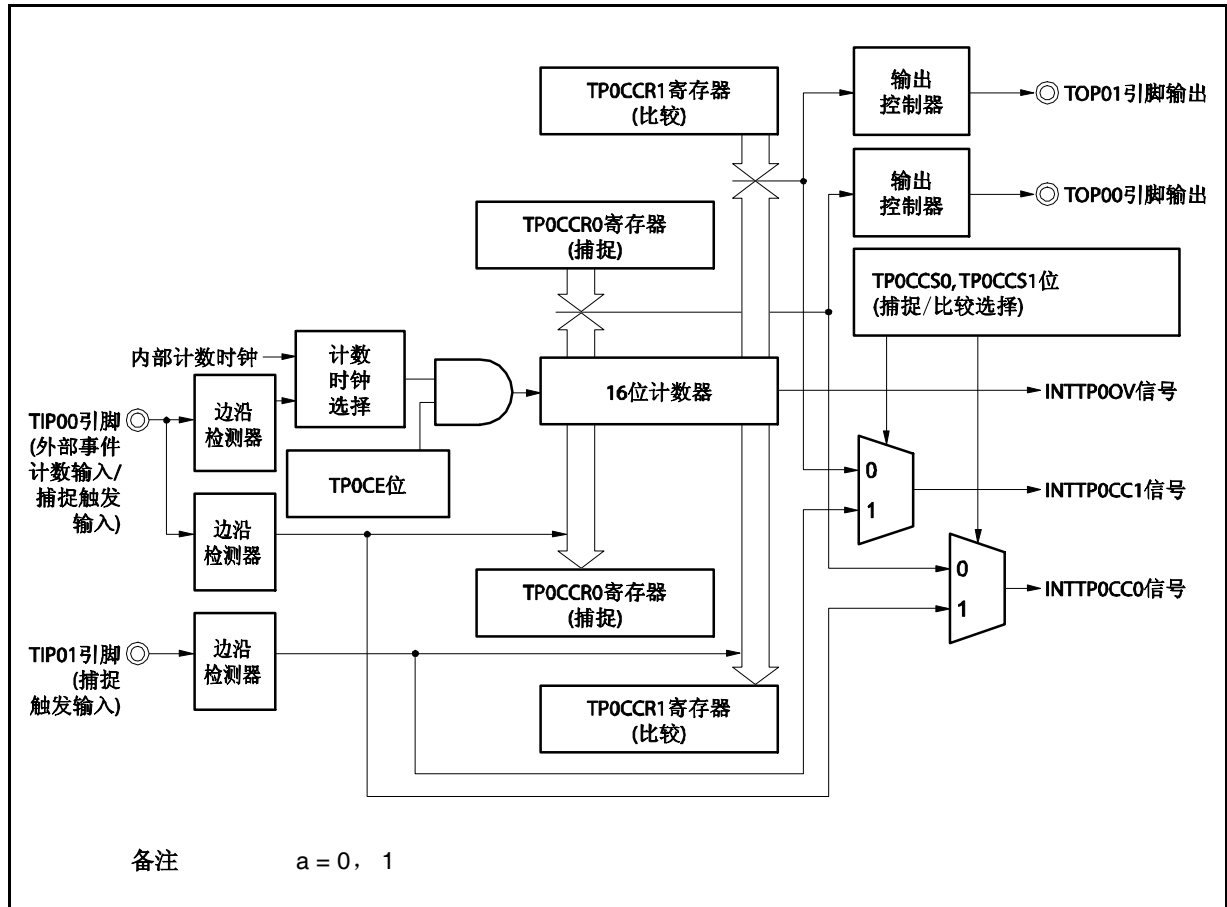
通常情况下，INTTP0CC1 信号会在 16 位计数器计数值与 TP0CCR1 寄存器的值匹配之后的下一个计数时钟同步产生。

但在 PWM 输出模式下，INTTP0CC1 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOP01 引脚输出信号变化的时序而改变的。

6.5.6 自由运行定时器模式 (TP0MD2 ~ TP0MD0 位 = 101)

在自由运行定时器模式中，TP0CTL0.TP0CE 位被置 1 后，16 位定时器/时间计数器 P 开始计数。此时，根据 TP0OPT0.TP0CCS0 和 TP0OPT0.TP0CCS1 位的设置，TP0CCRaRa 寄存器的功能可以在比较寄存器和捕捉寄存器之间进行选择。

图 6-28. 自由运行定时器模式的配置图

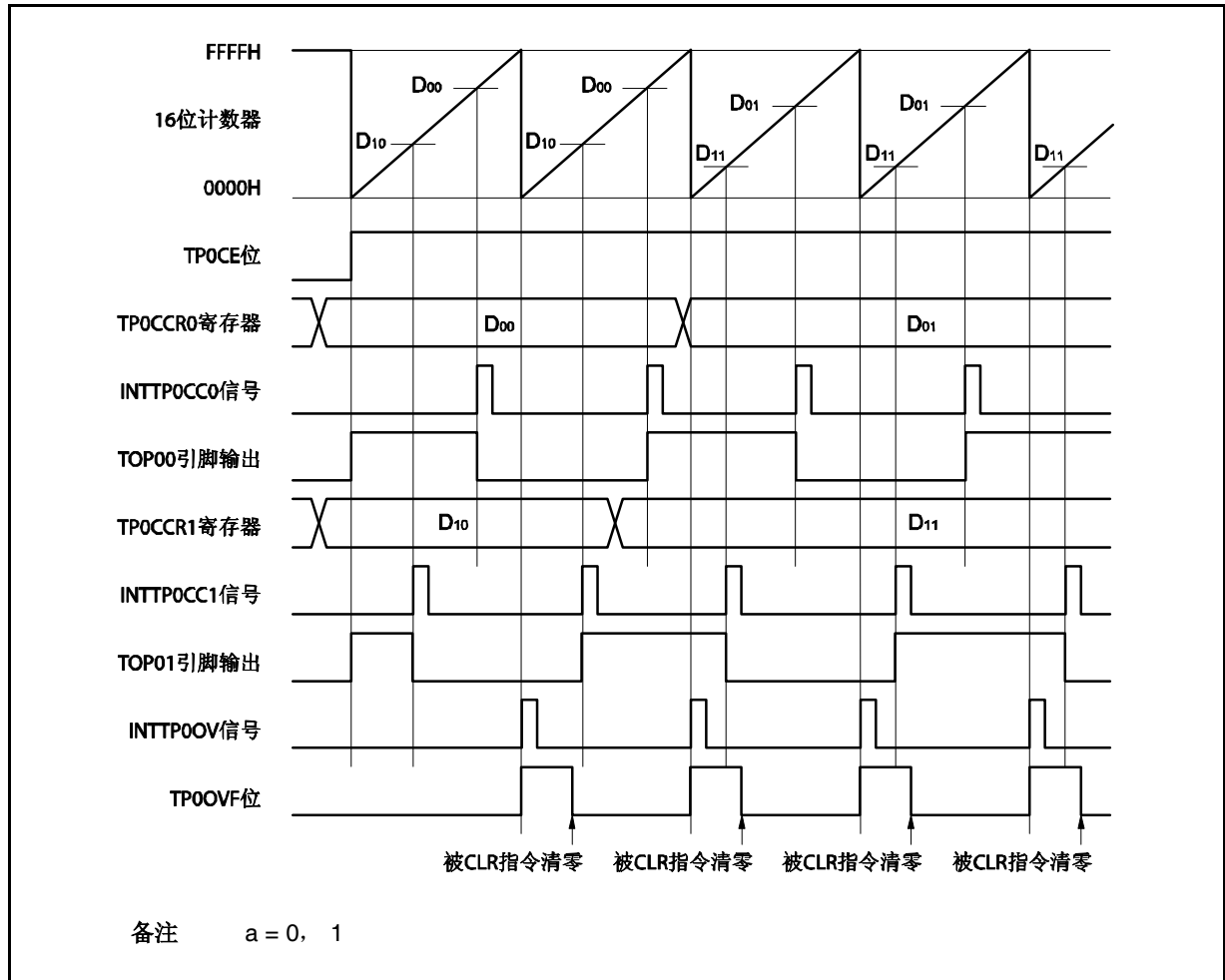


TPOCE 位被置 1 后，16 位计数器开始计数，同时 TOP00 和 TOP01 引脚输出信号的电平反转。当 16 位计数器的计数值与 TP0CCRa 寄存器的值匹配时将产生一个比较匹配中断请求信号(INTTP0CCa)，同时 TOP0a 引脚的输出电平将反转。

此后，16 位计数器将继续与计数时钟同步计数。当它计数到 FFFFH 时，将在下一个计数时钟时产生一个溢出中断请求信号(INTTP0OV)，同时计数器被清零并重新开始计数。此时，溢出标志(TP0OPT0.TP0OVF)也会被置 1。请通过软件使用 CLR 指令将溢出标志位清零。

可以在计数器计数过程中改写 TP0CCRa 寄存器的值。如果对 TP0CCRa 寄存器的值进行了改写，那么新的值将立即生效并用于与计数值的比较。

图 6-29. 自由运行定时器模式的基本时序 (比较功能)



TPOCE 位被置 1 后, 16 位计数器开始计数。当检测到 TIP0a 引脚输入信号的有效沿时, 16 位计数器的计数值被保存在 TP0CCRa 寄存器中, 同时产生一个捕捉中断请求信号(INTTP0CCa)。

此后, 16 位计数器将继续与计数时钟同步计数。当它计数到 FFFFH 时, 将在下一个计数时钟时产生一个溢出中断请求信号(INTTP0OV), 同时计数器被清零并重新开始计数。此时, 溢出标志(TP0OPT0.TP0OVF)也会被置 1。请通过软件使用 CLR 指令将溢出标志位清零。

图 6-30. 自由运行定时器模式的基本时序 (捕捉功能)

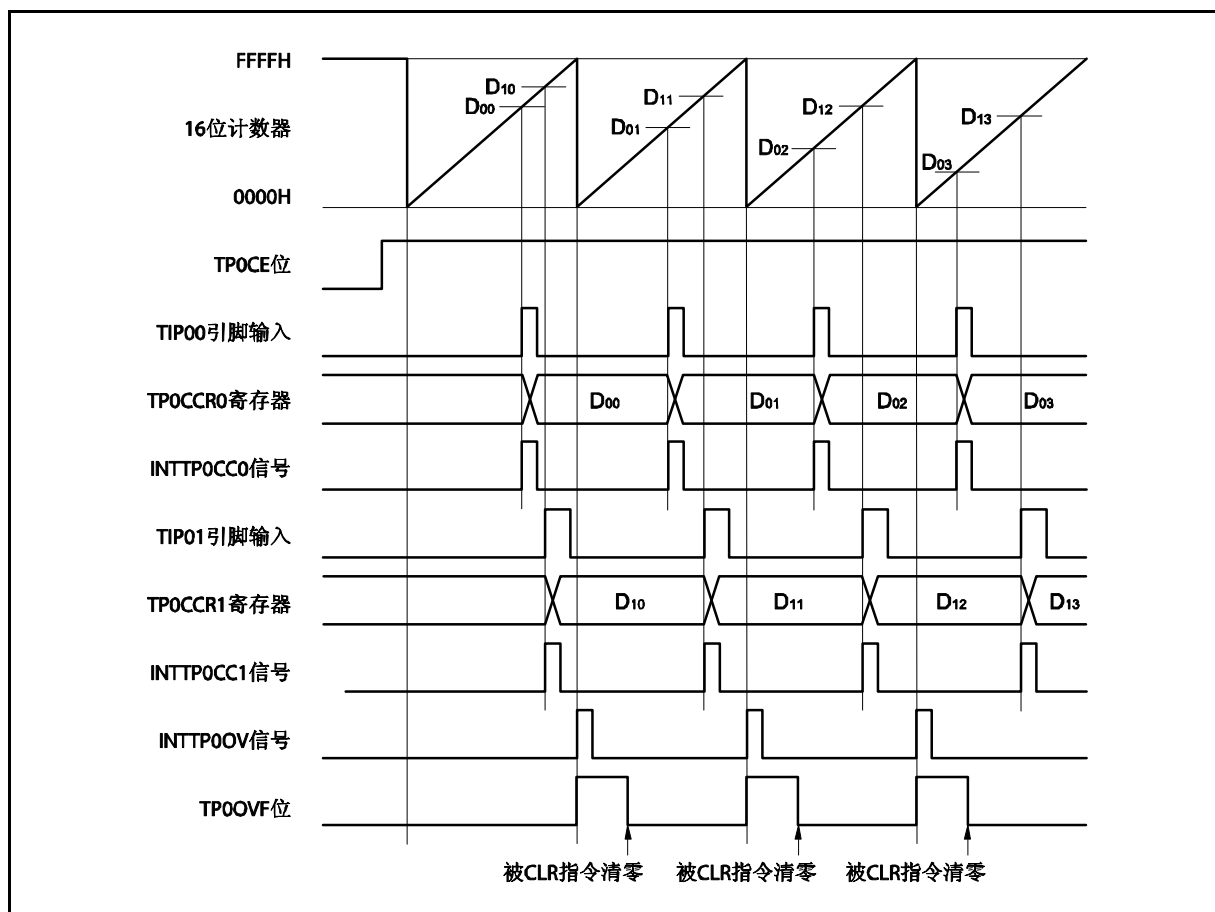


图 6-31. 自由运行定时器模式的寄存器设置 (1/2)

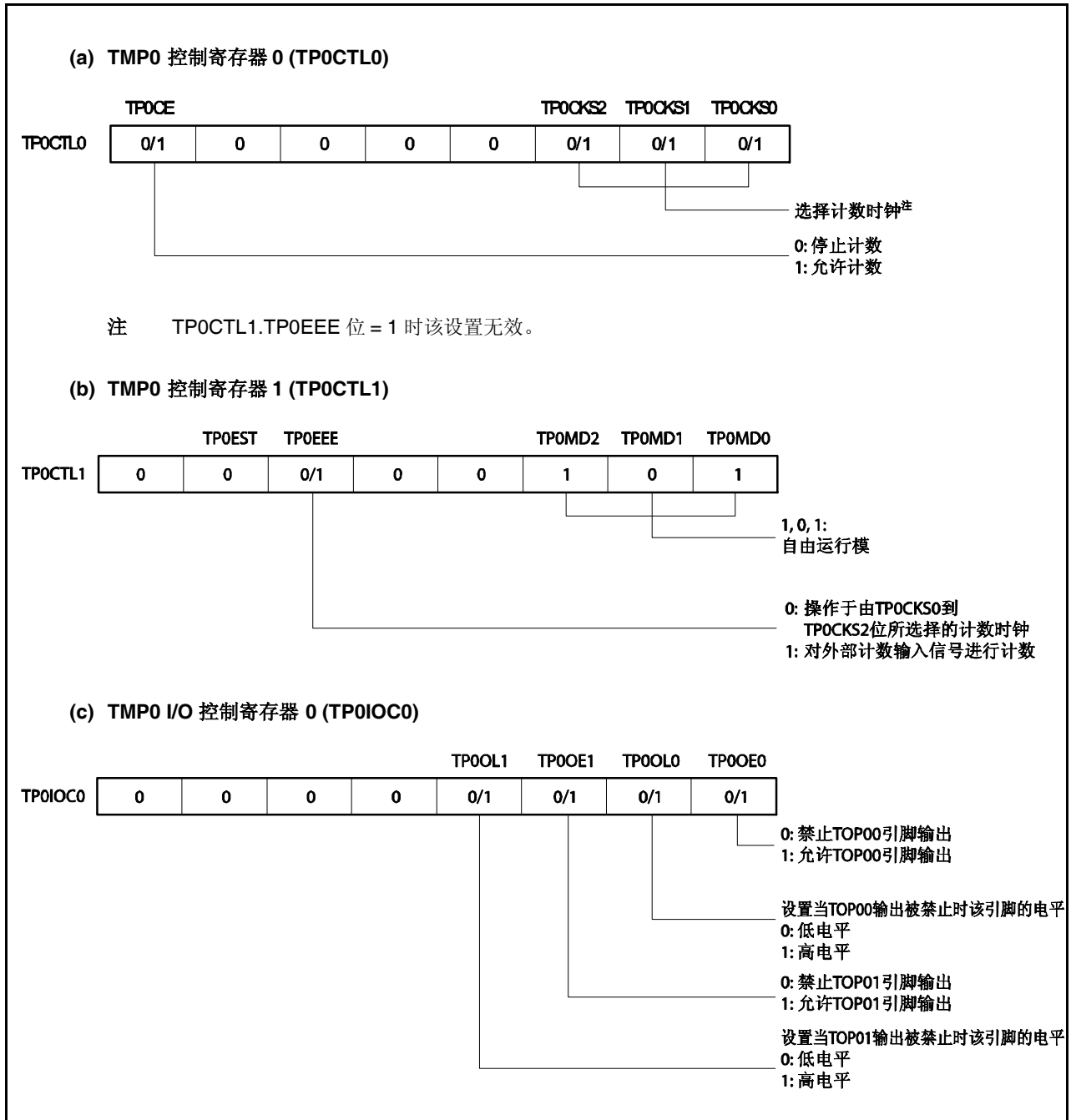
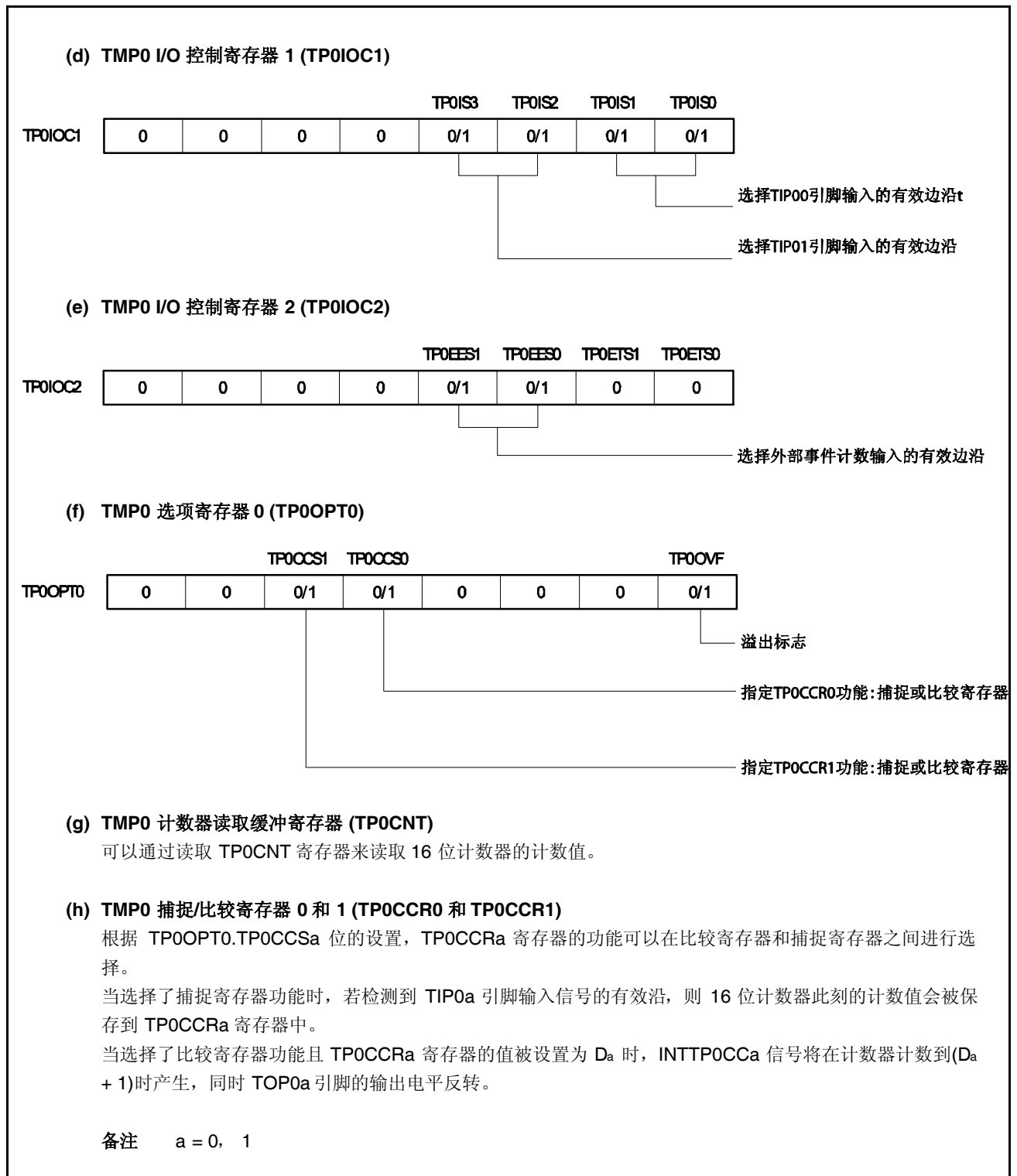


图 6-31. 自由运行定时器模式的寄存器设置 (2/2)



(1) 自由运行定时器模式的操作系统

(a) 当捕捉/比较寄存器用作比较寄存器时

图 6-32. 自由运行定时器模式的软件处理流程(比较功能) (1/2)

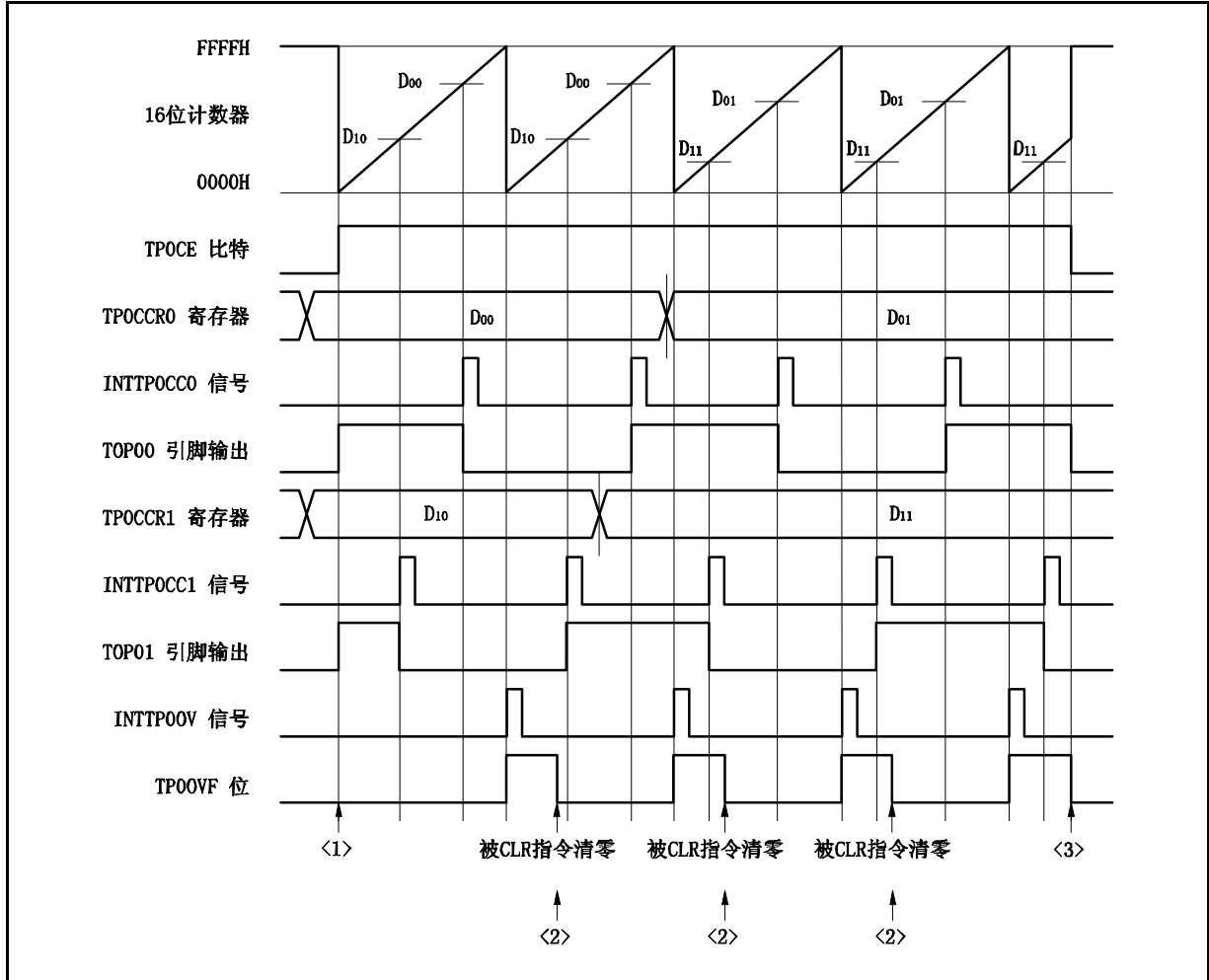
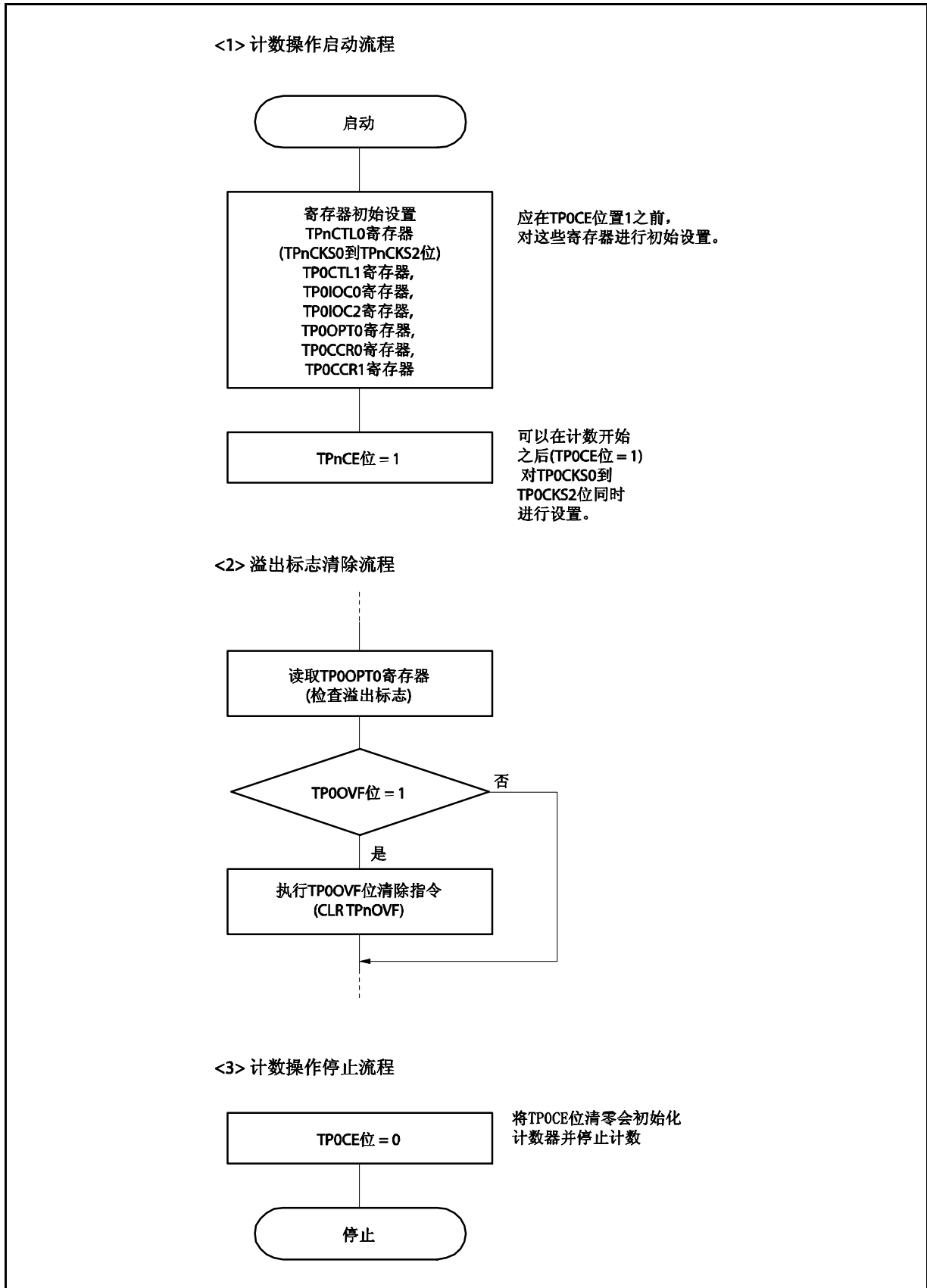


图 6-32. 自由运行定时器模式的软件处理流程 (比较功能) (2/2)



(b) 当捕捉/比较寄存器用作捕捉寄存器时

图 6-33. 自由运行定时器模式的软件处理流程 (捕捉功能) (1/2)

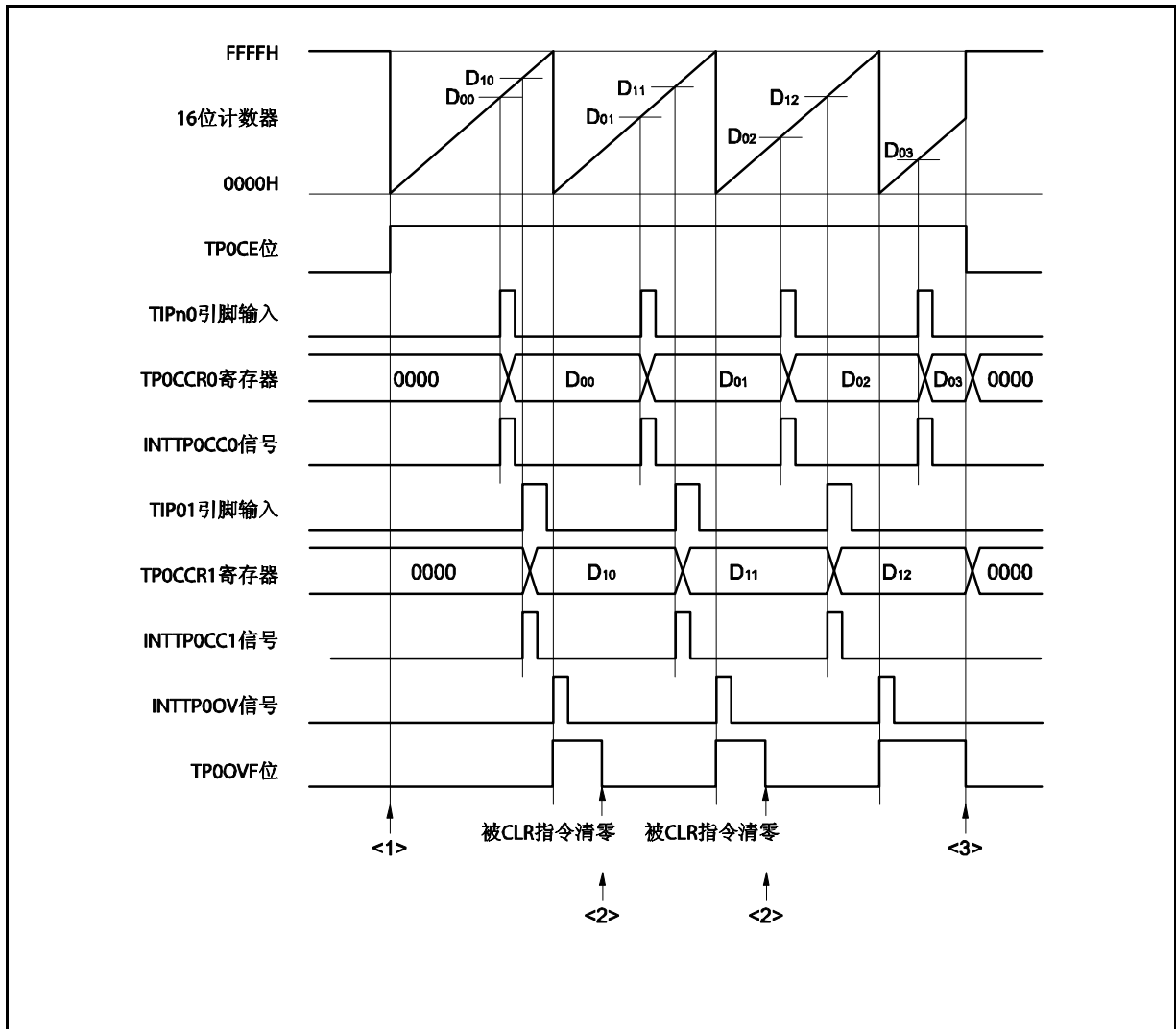
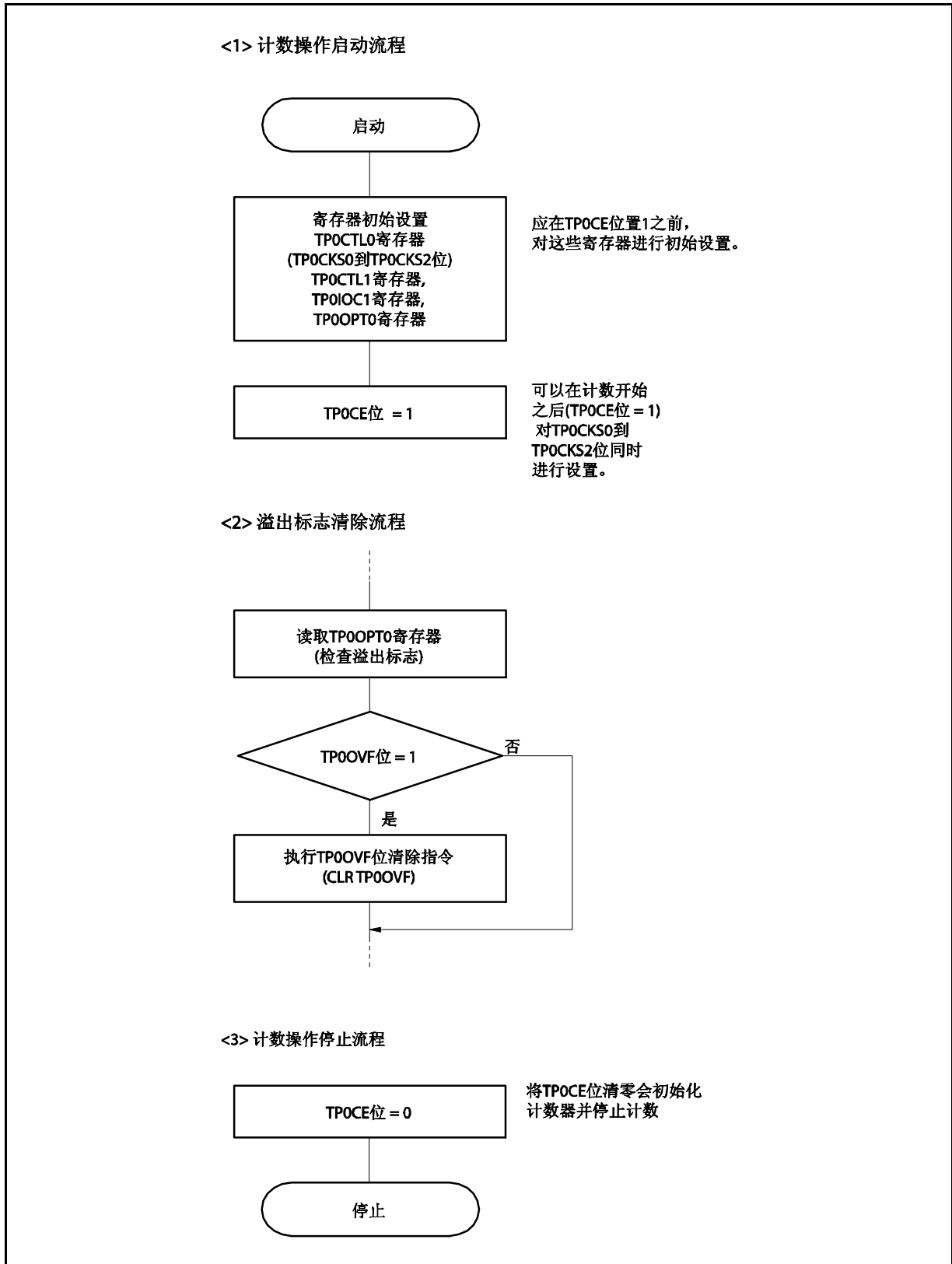


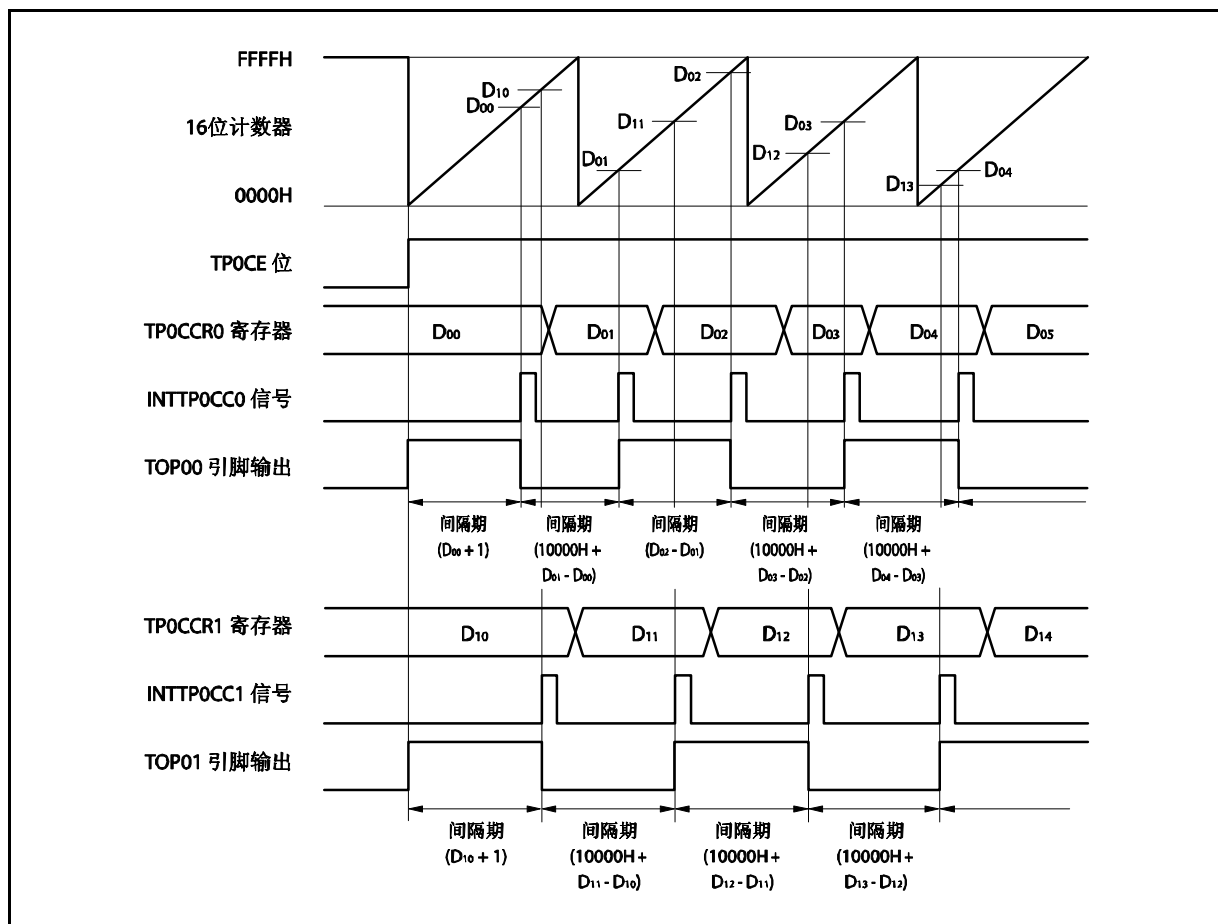
图 6-33. 自由运行定时器模式的软件处理流程 (捕捉功能) (2/2)



(2) 自由运行定时器模式的操作时序

(a) 使用比较寄存器功能的时间间隔操作

当 TPOCCRa 寄存器被作为比较寄存器使用时，16 位定时器/事件计数器 P 可作为间隔定时器使用，这时，每当检测到 INTTP0CCa 信号时，都需要通过软件处理的方式来设置比较值以指定下一次中断请求产生的间隔时间。



当使用自由运行定时器模式进行间隔时间操作时，可以在同一通道指定两个不同的间隔时间。进行间隔时间操作时，必须在每次检测到 INTTP0CCa 信号后执行的中断服务程序中将相应的值设置到 TPOCCRa 寄存器。

向 TPOCCRa 寄存器中重新设置的值可由以下算式求出，这里假设间隔时间为“Da”。

比较寄存器的最初设置值： $D_a - 1$

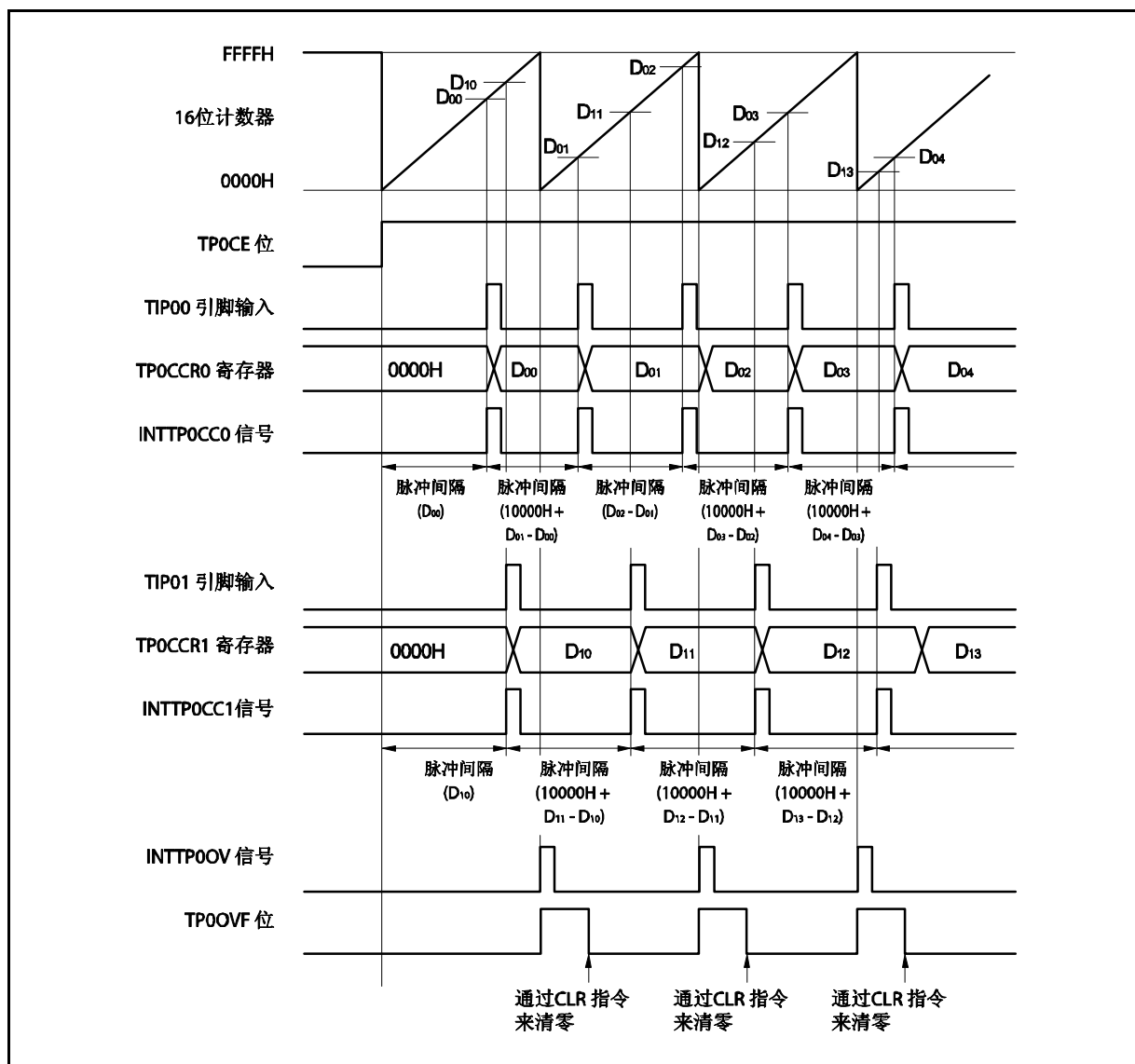
第二次设置比较寄存器以后(含第二次)的设置值：前次设置值 + D_a

(如果计算结果大于 FFFFH，那么将结果减 10000H 后再设置到寄存器中。)

备注 a = 0, 1

(b) 使用捕捉寄存器进行脉宽测量

当 TP0CCRa 寄存器被作为捕捉寄存器使用时, 可使用 16 位定时器/事件计数器 P 进行脉宽测量操作, 这里, 需要通过软件处理在每次检测到中断信号时对捕捉寄存器进行读取并计算间隔时间。

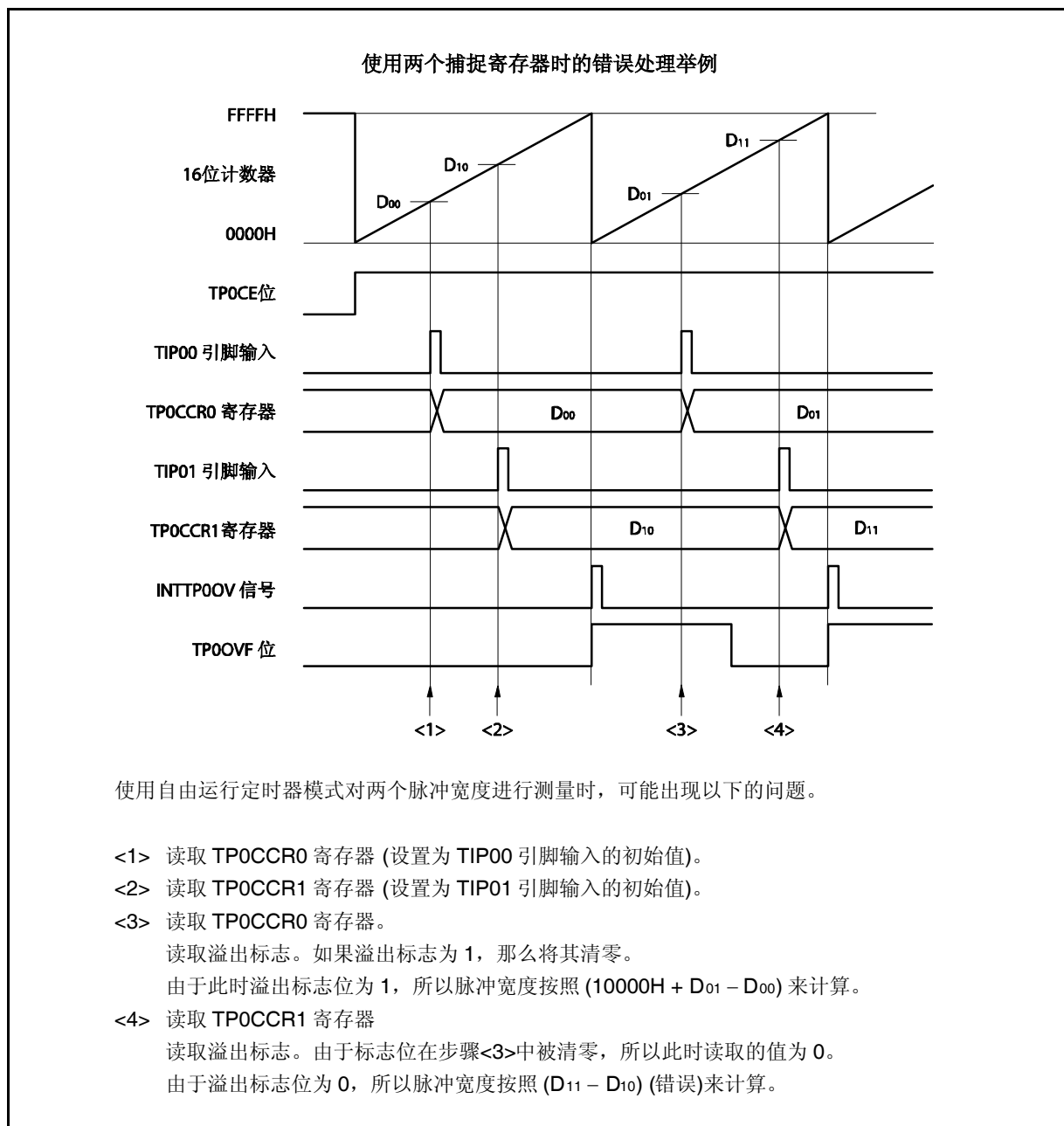


当使用自由运行定时器模式进行脉宽测量操作时, 可以在同一通道对两个不同的脉冲宽度进行测量。通过与 INTTP0CCa 信号同步读取 TP0CCRa 寄存器的值并计算该值与上一次读取值的差, 就可以求得脉冲的宽度。

备注 a = 0, 1

(c) 使用两个捕捉寄存器产生溢出时的操作

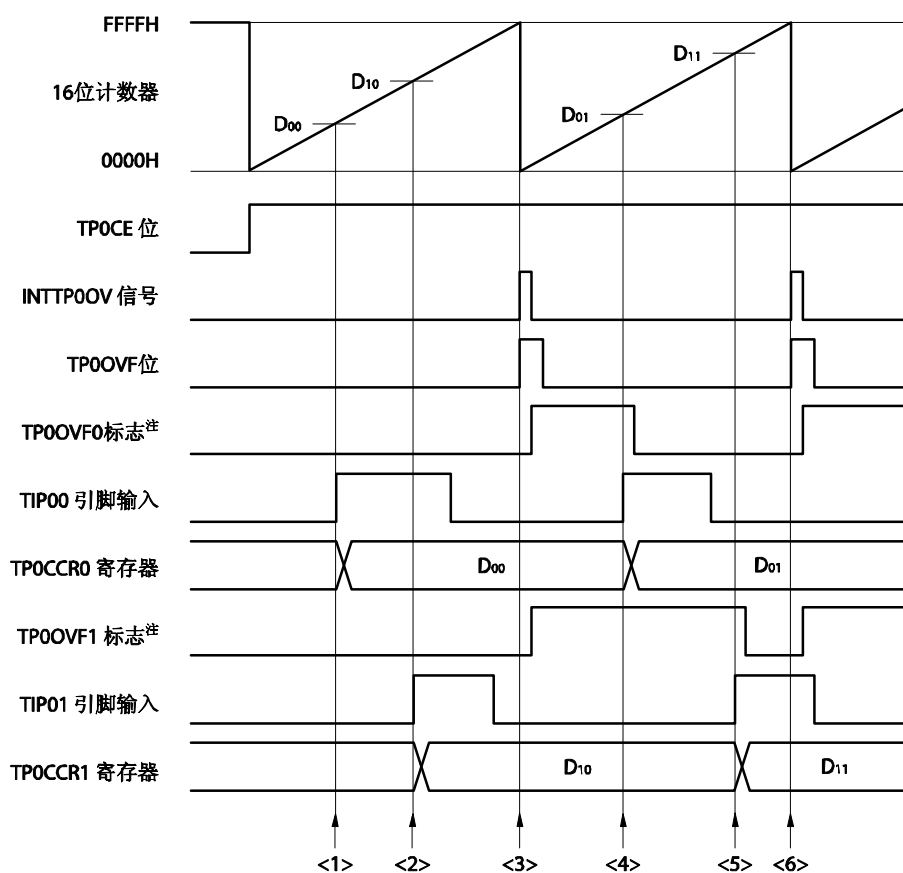
使用两个捕捉寄存器时，请务必慎重处理溢出标志位。首先，给出一个错误处理的例子，如下图所示。



当使用了两个捕捉寄存器时，如果溢出标志被其中一个捕捉寄存器清零，那么另一个捕捉寄存器将无法求得正确的脉冲宽度。

因此，使用两个捕捉寄存器时，需要通过软件来进行控制。下面给出一个软件控制的例子。

使用两个捕捉寄存器时的处理举例(利用溢出中断)



注 这里的 TP0OVF0 和 TP0OVF1 标志是建立于内置 RAM 上的变量，可由软件任意进行设置。

<1> 读取 TP0CCR0 寄存器 (设置为 TIP00 引脚输入的初始值)。

<2> 读取 TP0CCR1 寄存器 (设置为 TIP01 引脚输入的初始值)。

<3> 产生了溢出。在溢出中断服务程序中将 TP0OVF0 和 TP0OVF1 标志位置 1，同时将硬件溢出标志位清零。

<4> 读取 TP0CCR0 寄存器。

读取 TP0OVF0 标志位，如果该标志位为 1，那么将其清零。

由于 TP0OVF0 标志位为 1，脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。

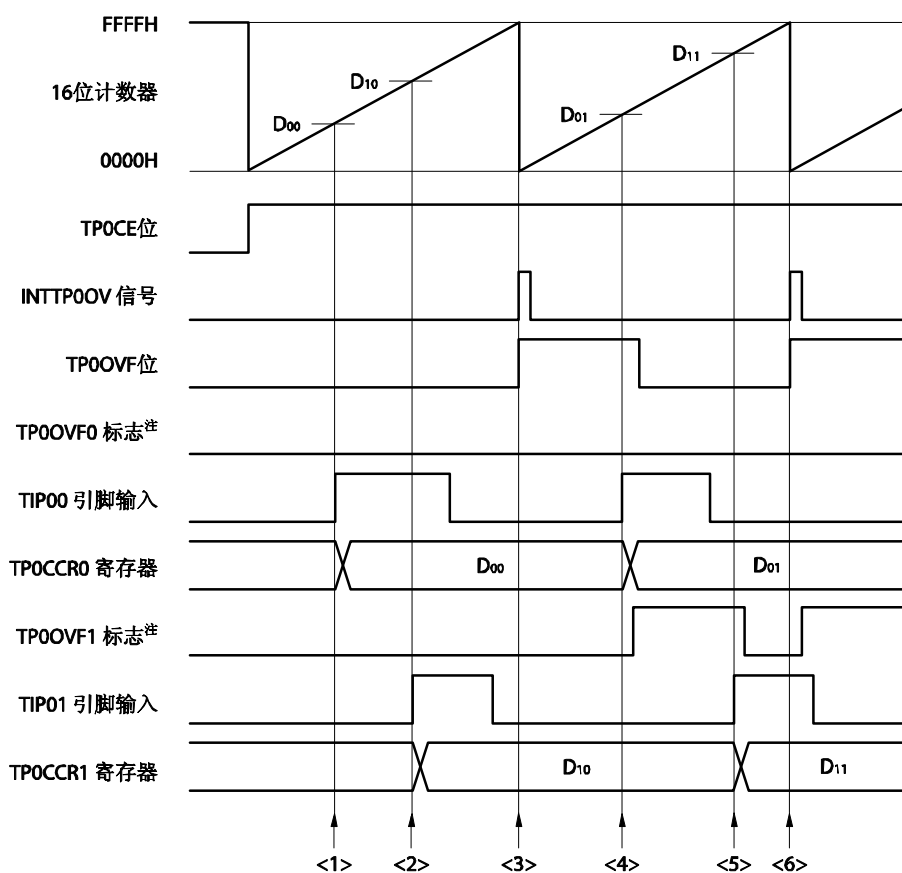
<5> 读取 TP0CCR1 寄存器。

读取 TP0OVF1 标志位，如果该标志位为 1，那么将其清零。(步骤<4>中将 TP0OVF0 位清零但 TP0OVF1 位始终保持为 1。)

由于 TP0OVF1 标志为 1，脉冲宽度按照 $(10000H + D_{11} - D_{10})$ (正确)来计算。

<6> 同步骤 <3>

使用两个捕捉寄存器时的处理举例(不利用溢出中断)



注 这里的 TP0OVF0 和 TP0OVF1 标志是建立于内置 RAM 上的变量，可由软件任意进行设置。

<1> 读取 TP0CCR0 寄存器 (设置为 TIP00 引脚输入的初始值)。

<2> 读取 TP0CCR1 寄存器 (设置为 TIP01 引脚输入的初始值)。

<3> 产生了溢出。但不进行任何软件处理。

<4> 读取 TP0CCR0 寄存器。

读取溢出标志。如果溢出标志为 1，那么只把 TP0OVF1 标志设置为 1，同时将溢出标志清零。

由于溢出标志位为 1，脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。

<5> 读取 TP0CCR1 寄存器。

读取溢出标志。由于溢出标志位在步骤<4>中被清零，所以读取结果为 0。

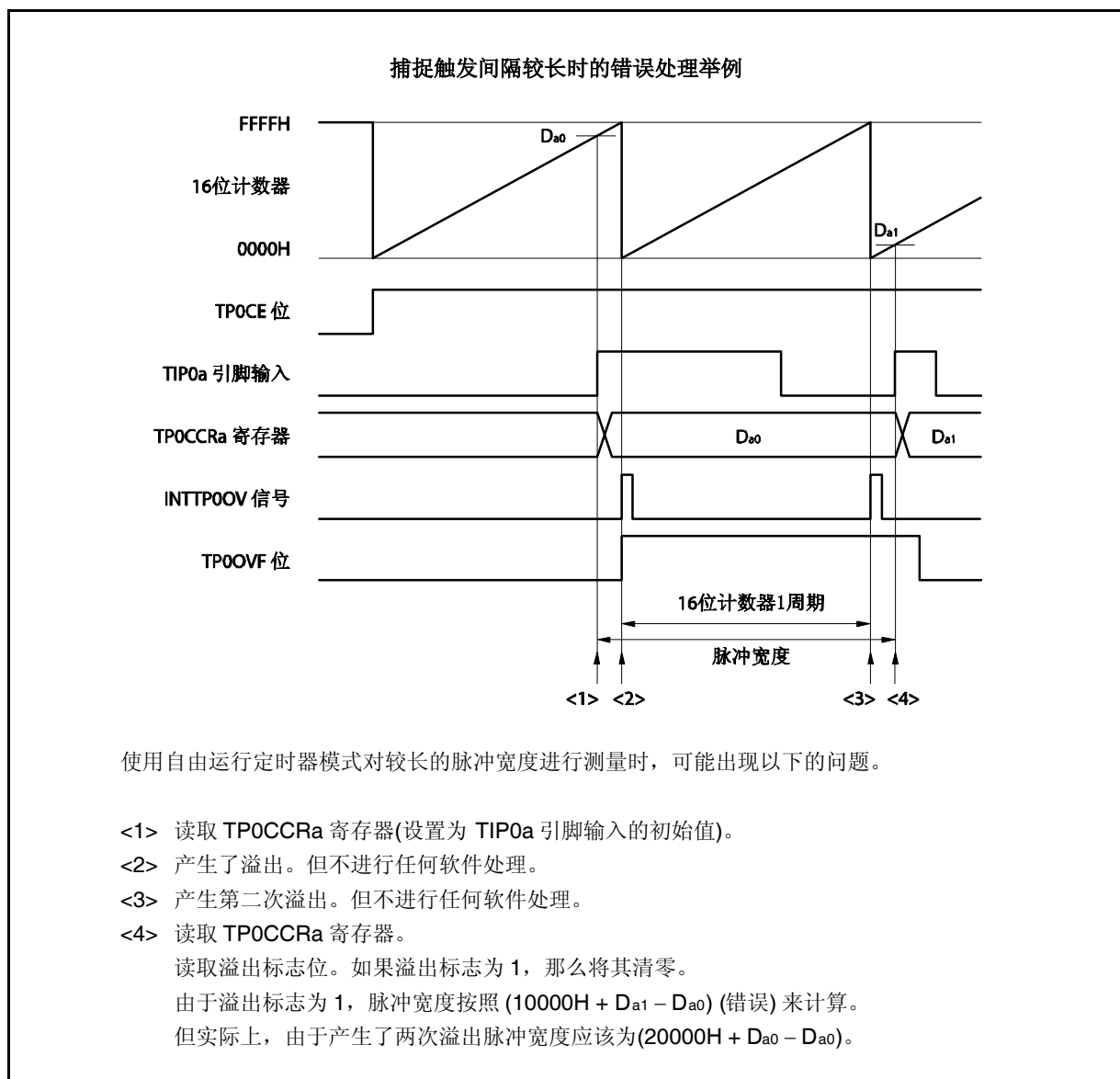
读取 TP0OVF1 位。如果 TP0OVF1 标志为 1，那么将其清零。

由于 TP0OVF1 标志为 1，脉冲宽度按照 $(10000H + D_{11} - D_{10})$ (正确)来计算。

<6> 同步步骤 <3>

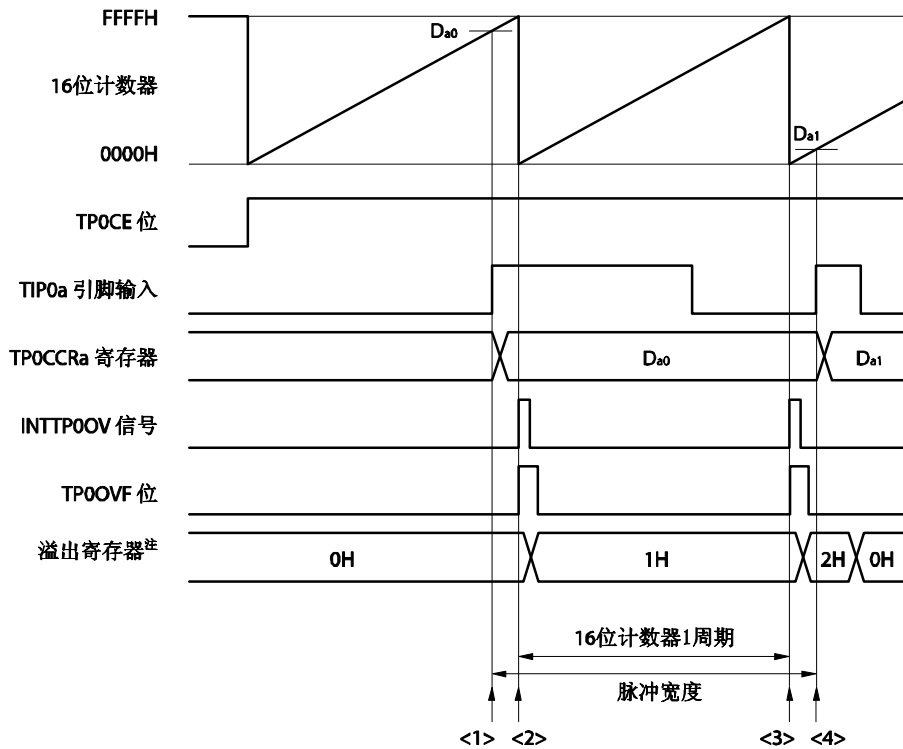
(d) 捕捉触发间隔较长时的溢出处理

若脉冲宽度大于 16 位计数器的计数周期，则请务必慎重处理溢出标志，因为在两次捕捉触发信号之间，可能产生不只一次的溢出。首先给出一个错误处理的例子，如下所示。



如果捕捉触发间隔时间较长以至产生两次或更多次溢出，那么计算结果可能为错误的脉冲宽度。因此，在捕捉触发间隔时间较长时，请减慢 16 位计数器的计数时钟以延长计数周期或者通过软件的方式避免错误的产生。下面给出一个使用软件方式避免错误的例子。

捕捉触发间隔较长时的处理举例

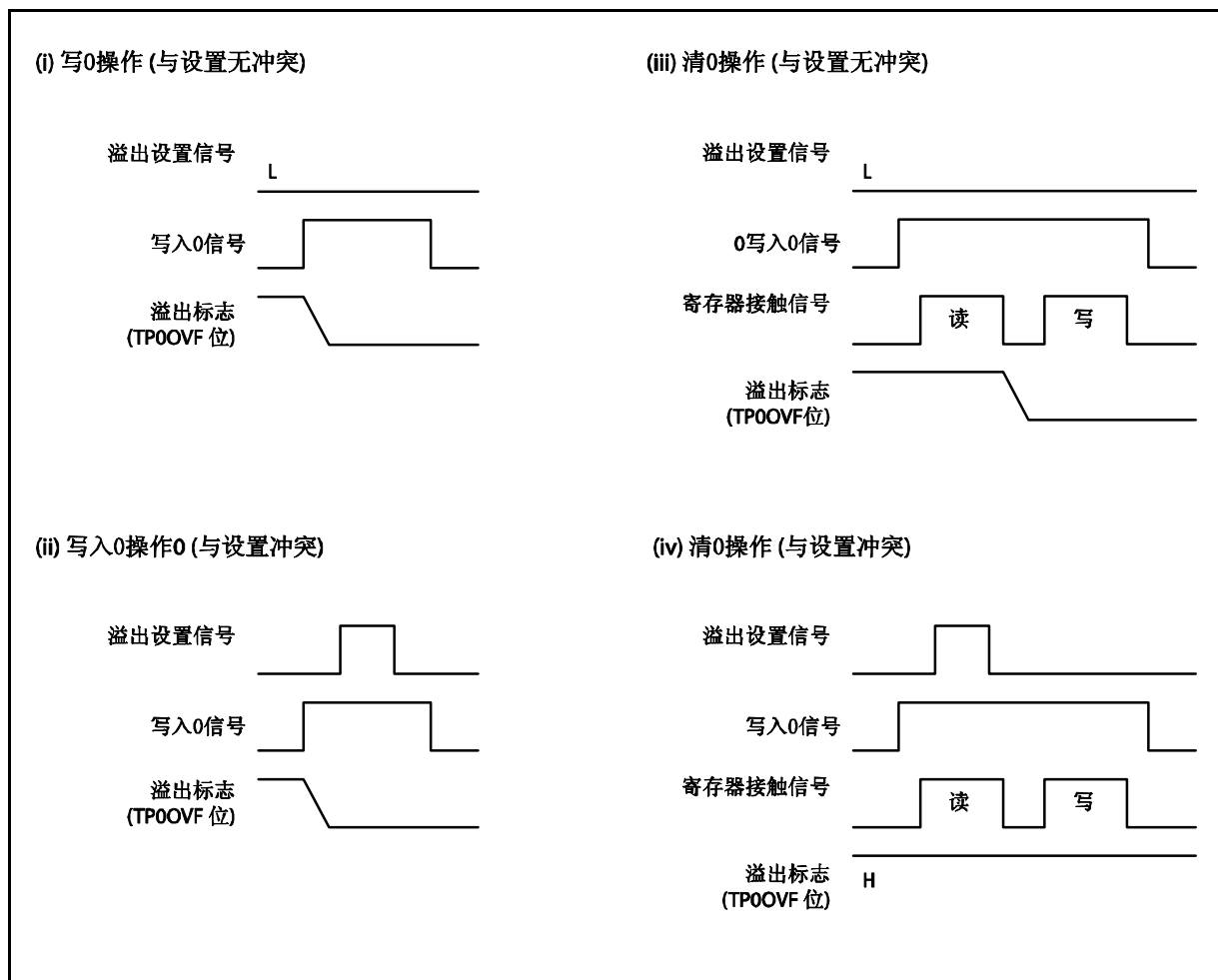


注 这里的溢出次数计数器是建立在内置 RAM 上的变量，可由软件任意进行设置。

- <1> 读取 TP0CCRa 寄存器 (设置为 TIP0a 引脚输入的初始值)。
- <2> 产生了溢出。在溢出中断服务程序中将溢出次数计数器加 1 并将溢出标志清零。
- <3> 产生了第二次溢出。在溢出中断服务程序中将溢出次数计数器加 1 并将溢出标志清零。
- <4> 读取 TP0CCRa 寄存器。
读取溢出次数计数器的值。
→ 假设此时溢出计数器的值为“N”，则脉冲宽度可按 $(N \times 10000H + D_{a1} - D_{a0})$ 计算求得。
在本例中，由于溢出产生了两次，所以脉冲宽度为 $(20000H + D_{a1} - D_{a0})$ 。
将溢出次数计数器清零(0H)。

(e) 将溢出标志位清零

可以通过使用 CLR 指令清零 TP0OVF 位或向 TP0OPT0 寄存器写入 8 位数据(第 0 位为 0)的方法将溢出标志位清零。为了能够准确检测到溢出的发生, 请务必在每次读取到 TP0OVF 位为 1 后, 立即使用位操作指令将该位清零。



当读取到溢出标志位为 1 后, 使用 CLR 指令将其清零以清除溢出标志。如果不读取溢出标志就直接向其写入 0, 那么溢出信息将可能被擦除(上图中的(ii))。这种情况下, 即使产生过溢出, 软件也会按照无溢出来处理。

如果 CLR 指令的执行与溢出产生的时刻发生冲突, 那么即使执行了 CLR 指令, 溢出标志也会保持为 1。

6.5.7 脉宽测量模式 (TP0MD2 ~ TP0MD0 位 = 110)

在脉宽测量模式中，TP0CTL0.TP0CE 位被置 1 后，16 位计数器开始计数。每当检测到 TIP0a 引脚输入信号的有效沿时，16 位计数器的计数值就会被保存到 TP0CCRa 寄存器中，同时计数器清零为 0000H。

两个有效沿之间的间隔时间可以在捕捉中断请求信号 (INTTP0CCa) 产生后通过读取 TP0CCRa 寄存器的值来计算。

可以选择 TIP00 或 TIP01 引脚中的任意一个作为捕捉触发输入引脚。通过 TP0IOC1 寄存器将未使用的引脚设置为“无边沿检测”。

当使用外部时钟来测量脉宽时，只能使用 TIP01 引脚作为捕捉触发输入引脚，因为外部时钟输入已经固定为 TIP00 引脚。此时要将 TP0IOC1.TP0IS1 和 TP0IOC1.TP0IS0 位清零 (捕捉触发输入 (TIP00 引脚)：无边沿检测)。

图 6-34. 脉宽测量模式的配置图

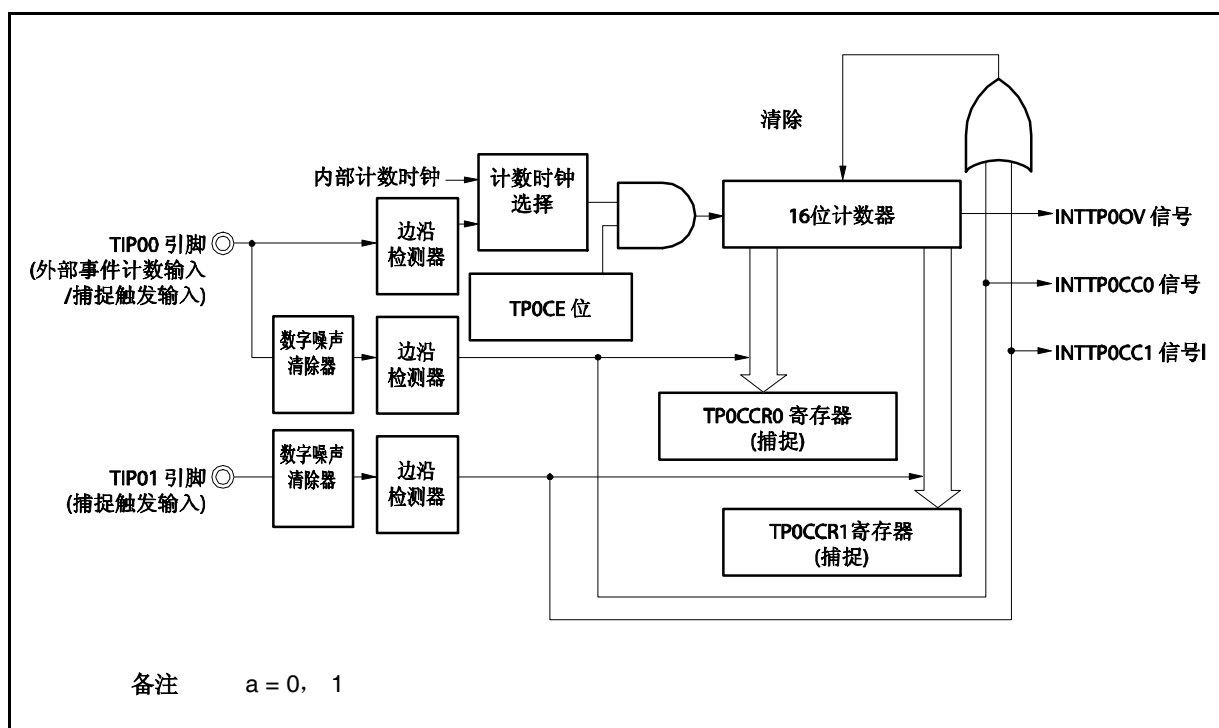
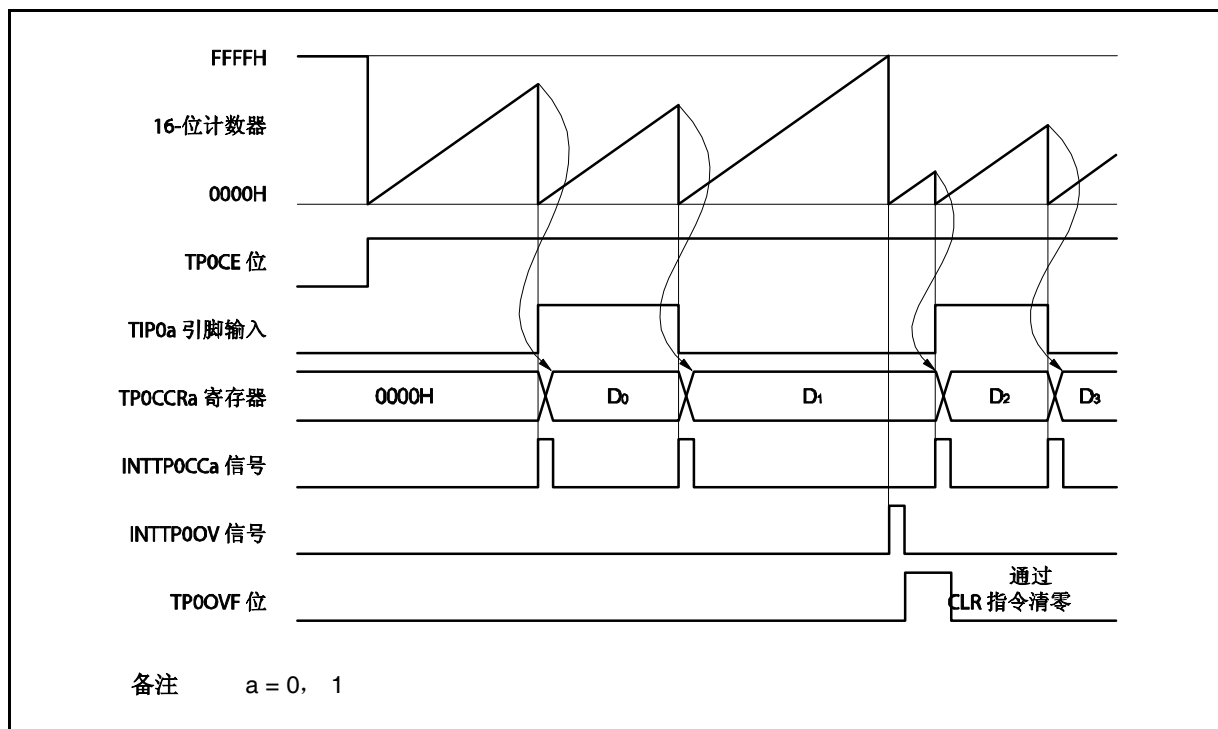


图 6-35. 脉宽测量模式的基本时序



当 TPOCE 位被置 1 后，16 位计数器开始计数。当检测到 TIP0a 引脚的有效沿时，16 位计数器的计数值将被保存到 TP0CCRA 寄存器中，同时计数器被清零并产生一个捕捉中断请求信号(INTTP0CCa)。

脉冲宽度可由下式求得。

$$\text{脉冲宽度} = \text{捕捉计数值} \times \text{计数时钟周期}$$

如果直到 16 位计数器计数到 FFFFH 仍没有检测到 TIP0a 引脚的有效沿，那么将在下一个时钟时产生一个溢出中断请求信号(INTTP0OV)，同时计数器被清零并重新开始计数。这种情况下，溢出标志位(TP0OPT0.TP0OVF 位)也将被设置为 1。请通过软件使用 CLR 指令将溢出标志清零。

如果溢出标志被设置为 1，那么脉冲宽度可由下式求得。

$$\text{脉冲宽度} = (10000H \times \text{TP0OVF 位被设置为 1 的次数} + \text{捕捉计数值}) \times \text{计数时钟周期}$$

备注 a = 0, 1

图 6-36. 脉宽测量模式的寄存器设置 (1/2)

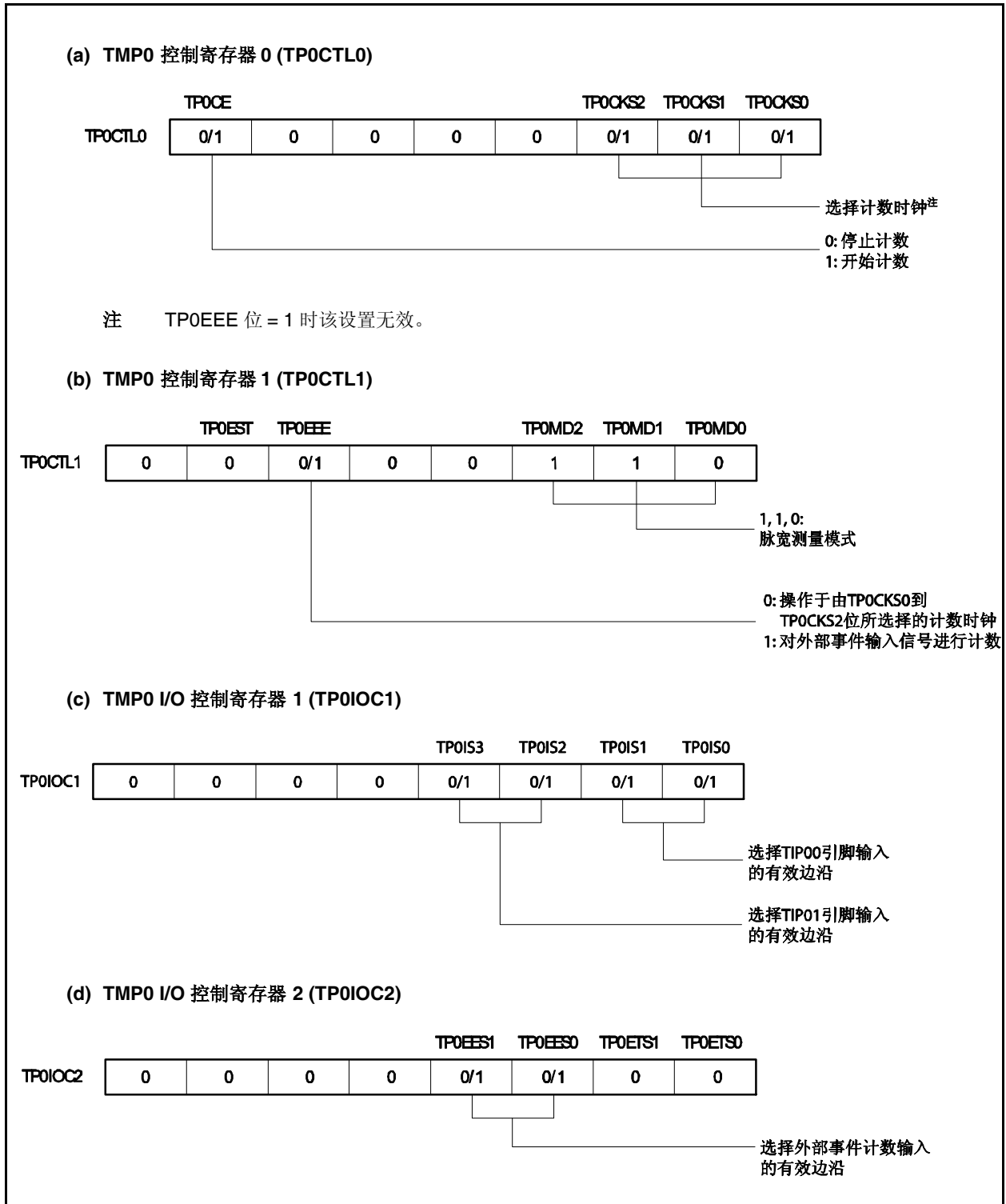
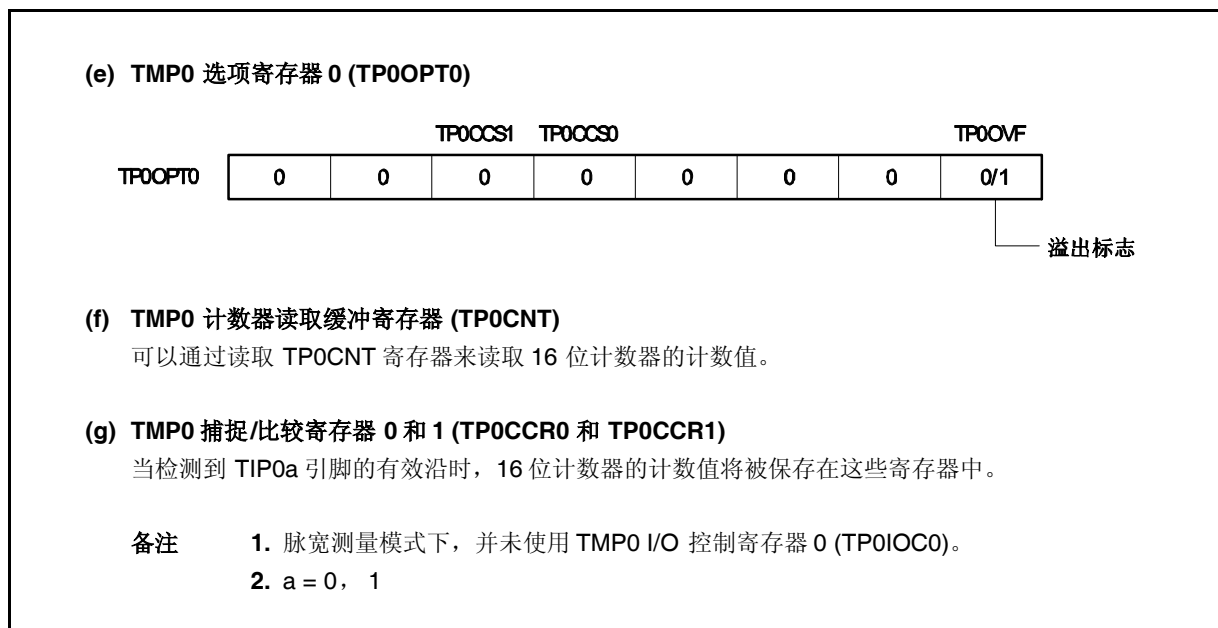
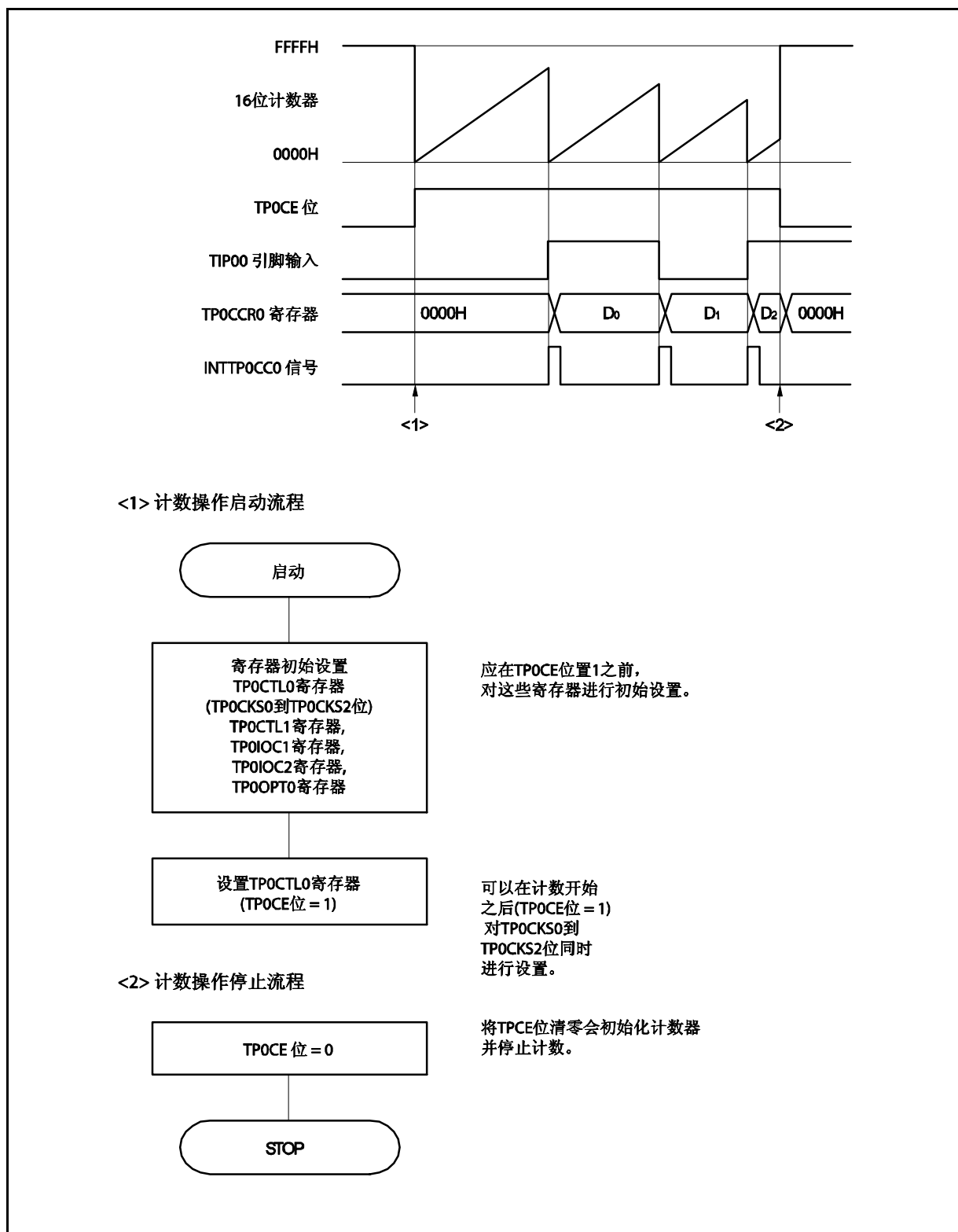


图 6-36. 脉宽测量模式的寄存器设置 (2/2)



(1) 脉宽测量模式的操作系统

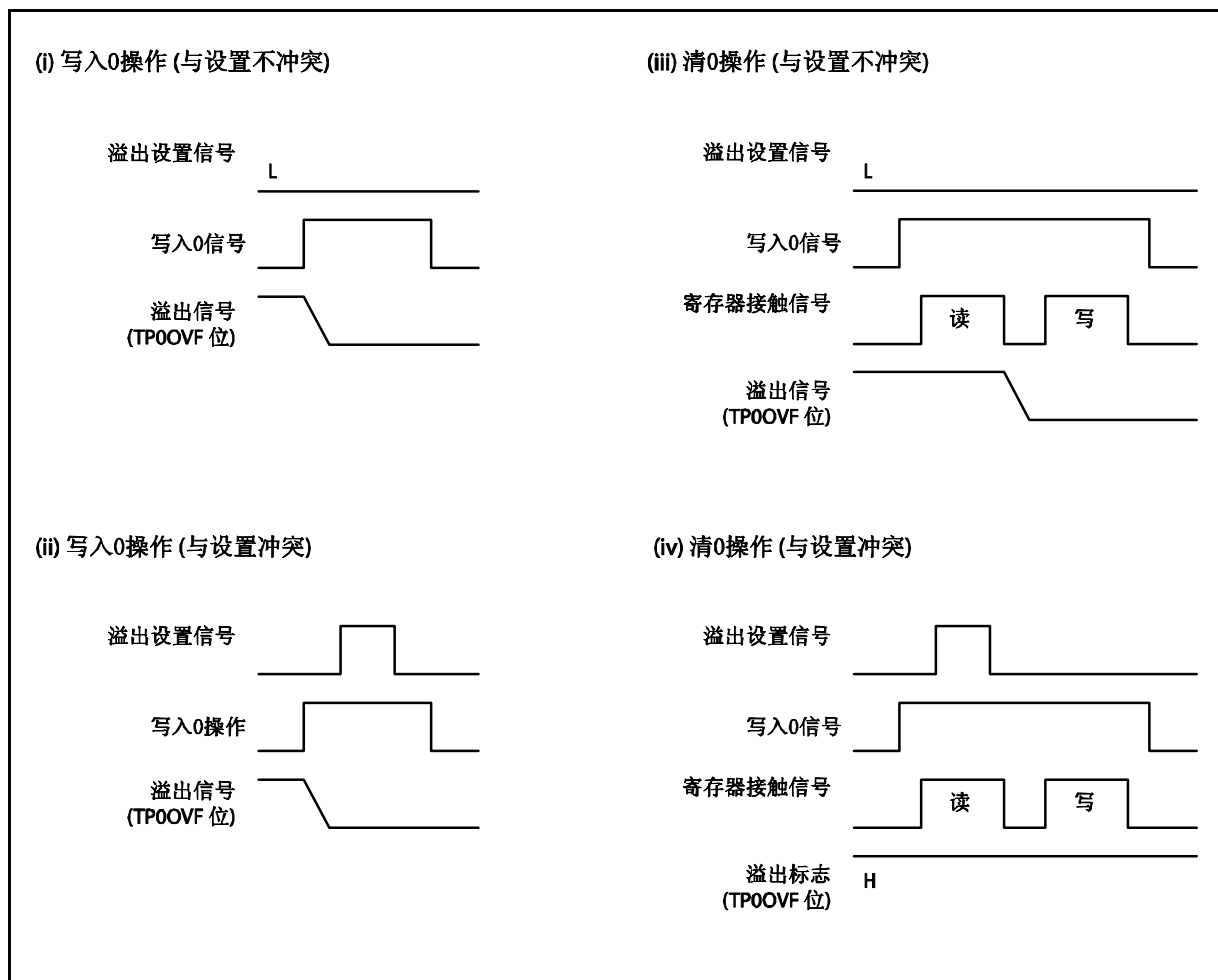
图 6-37. 脉宽测量模式的软件处理流程



(2) 脉宽测量模式的操作时序

(a) 将溢出标志位清零

可以通过使用 CLR 指令清零 TP0OVF 位或向 TP0OPT0 寄存器写入 8 位数据(第 0 位为 0)的方法将溢出标志位清零。为了能够准确检测到溢出的发生, 请务必在每次读取到 TP0OVF 位为 1 后, 立即使用位操作指令将该位清零。



当读取到溢出标志位为 1 后, 使用 CLR 指令将其清零以清除溢出标志。如果不读取溢出标志就直接向其写入 0, 那么溢出信息将可能被擦除(上图中的(ii))。这种情况下, 即使产生过溢出, 软件也会按照无溢出来处理。

如果 CLR 指令的执行与溢出产生的时刻发生冲突, 那么即使执行了 CLR 指令, 溢出标志也会保持为 1。

6.5.8 定时器输出操作

TOP00 和 TOP01 引脚的工作模式与输出电平的关系如下表所示。

表 6-4. 各模式下的定时器输入控制

操作模式	TOP01 引脚	TOP00 引脚
间隔定时器模式	方波输出	
外部事件计数模式	方波输出	—
外部触发脉冲输出模式	外部触发脉冲输出	方波输出
单脉冲输出模式	单脉冲输出	
PWM 输出模式	PWM 输出	
自由运行定时器模式	方波输出 (仅限使用比较功能时)	
脉宽测量模式		—

表 6-5. TOP00 和 TOP01 引脚在定时器输出控制位控制下的真值表

TP0IOC0.TP0OLm 位	TP0IOC0.TP0OEm 位	TP0CTL0.TP0CE 位	TOP0m 引脚的电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前为低电平, 计数过程中为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平, 计数过程中为低电平

备注 a = 0, 1

6.6 捕捉触发输入引脚(TIP0a)的噪声消除

引脚有一个数字噪声消除器。

但是仅当此引脚作为捕捉触发输入引脚时噪声消除电路有效。此引脚作为外部事件计数输入引脚或者外部触发输入引脚时此电路无效。

当通过 PMC3, PFC3, 和 PFCE3 寄存器指定 TIP0a 引脚的复用功能时可以消除数字噪声。

采样次数可以使用 PaNFC.PaNFSTS 位选择, 可选值为 2 或 3。采样时钟可以通过 PaNFC.PaNFC2 ~ PaNFC.PaNFC0 位从 fxx, fxx/2, fxx/4, fxx/16, fxx/32, 或 fxx/64 中选择。

(1) TIP0a 噪声消除控制寄存器(PaNFC)

该寄存器用于选择采样时钟和消除数字噪声的采样次数。

该寄存器可以进行 8 位或 1 位读写操作。

复位操作将此寄存器清为 00H。

复位后: 00H R/W 地址: P0NFC FFFFFB00H, P1NFC FFFFFB04H								
	7	6	5	4	3	2	1	0
PaNFC (a = 0, 1)	0	PaNFSTS	0	0	0	PaNFC2	PaNFC1	PaNFC0
PaNFSTS	清除数字噪声的抽样次数设置							
0	抽样次数 = 3							
1	抽样次数 = 2							
PaNFC2	PaNFC1	PaNFC0	抽样时钟选择					
0	0	0	fxx					
0	0	1	fxx/2					
0	1	0	fxx/4					
0	1	1	fxx/16					
1	0	0	fxx/32					
1	0	1	fxx/64					
上面所列除外			禁止设置					

注意事项 1. 经过采样时钟周期×采样次数的时间后, 再将 TMP0 的 16 位计数器使能(TP0CTL.TP0CE 位 = 1)。

2. 务必将位 7, 5~3 全部清零。

<设置过程>

- <1> 用 PaNFC 寄存器选择采样次数和采样时钟。
- <2> 用 PMC3, PFC3, 和 PFCE3 寄存器选择 (TIP0a 引脚的) 复用功能
- <3> 设置 TMP0 的工作模式(如捕捉模式或捕捉触发的有效边沿)。
- <4> 使能 TMP0 计数操作。

<噪声消除宽度>

数字噪声消除宽度(t_{WTIPa})如下。T 是采样时钟周期, M 是采样倍数。

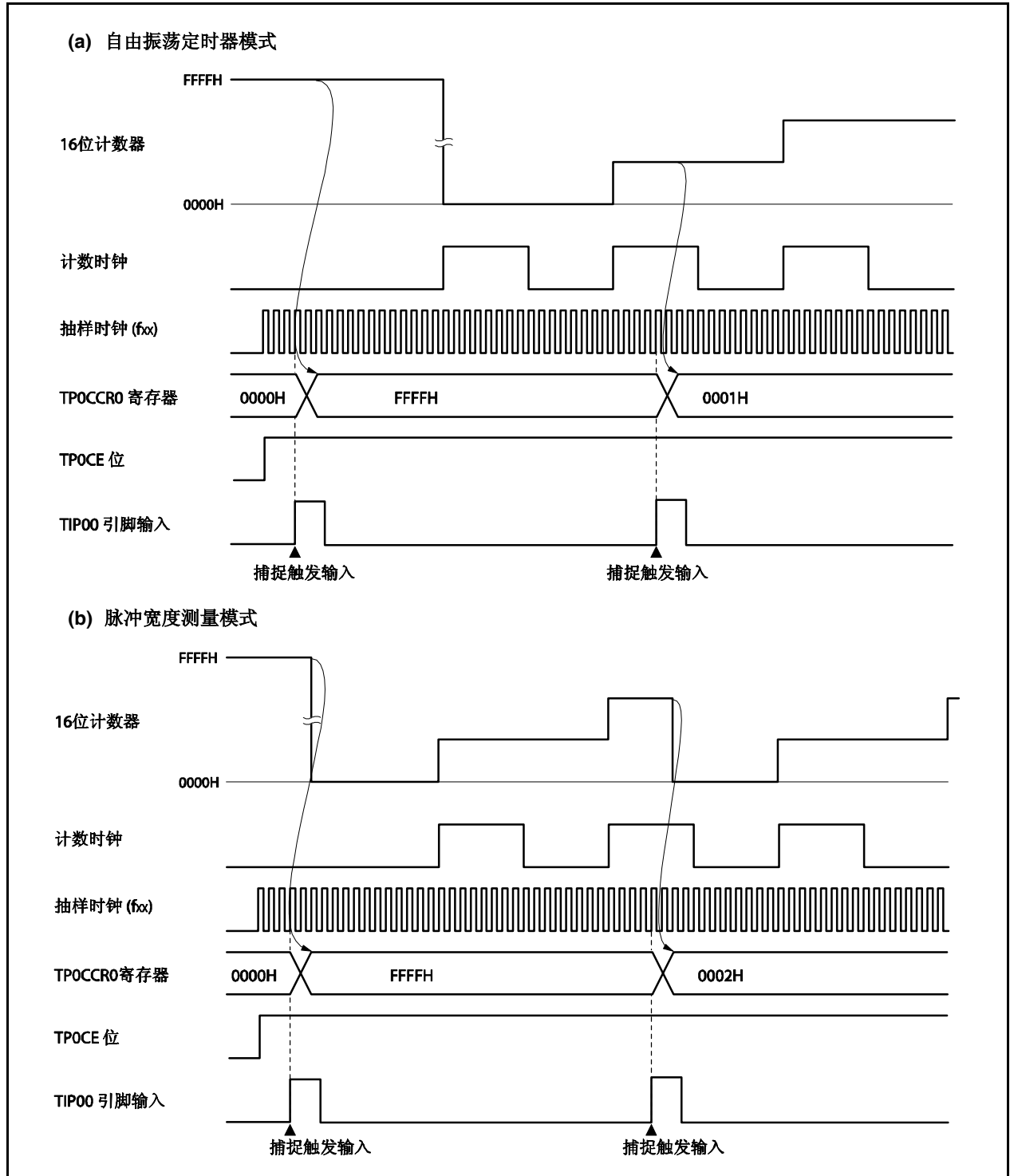
- $t_{WTIPa} < (M - 1)T$: 确认为噪声
- $(M - 1)T \leq t_{WTIPa} < MT$: 视为噪声消除或视为有效边沿
- $t_{WTIPa} \geq MT$: 确认为有效边沿

因此, 必须输入脉宽大于等于 MT 的脉冲, 才能被精确的检测为捕捉触发输入的有效边沿。

6.7 注意事项

(1) 捕捉操作

当使用捕捉操作，且选择 $f_{xx}/8$ ， $f_{xx}/16$ ， $f_{xx}/32$ ， $f_{xx}/64$ ， $f_{xx}/128$ ，或外部事件计数器(TP0CLT1.TP0EEE 位 = 1)为计数时钟时，如果 TP0CE 位置 1 后马上输入捕捉触发源，TP0CCRn 寄存器可能会捕捉到 FFFFH 而不是 0000H。



第七章 16 位定时器/事件计数器 0

V850ES/KE2，提供了 1 个单通道 16 位定时器/事件计数器 0。

7.1 功能

16 位定时器/事件计数器 01 有如下功能。

(1) **间隔定时器**

16 位定时器/事件计数器 01 每隔指定间隔时间都会产生一个中断请求信号。

(2) **方波输出**

16 位定时器/事件计数器 01 可以输出任意指定频率的方波。

(3) **外部事件计数器**

16 位定时器/事件计数器 01 可以测量外部输入信号的脉冲个数。

(4) **单脉冲输出**

16 位定时器/事件计数器 01 可以输出自由设定脉冲宽度的单脉冲。

(5) **PPG 输出**

16 位定时器/事件计数器 01 可以输出自由设定频率和脉冲宽度的矩形波。

(6) **脉冲宽度测量**

16 位定时器/事件计数器 01 可以测量外部输入信号的脉冲宽度。

7.2 配置

16 位定时器/事件计数器 01 包括以下硬件。

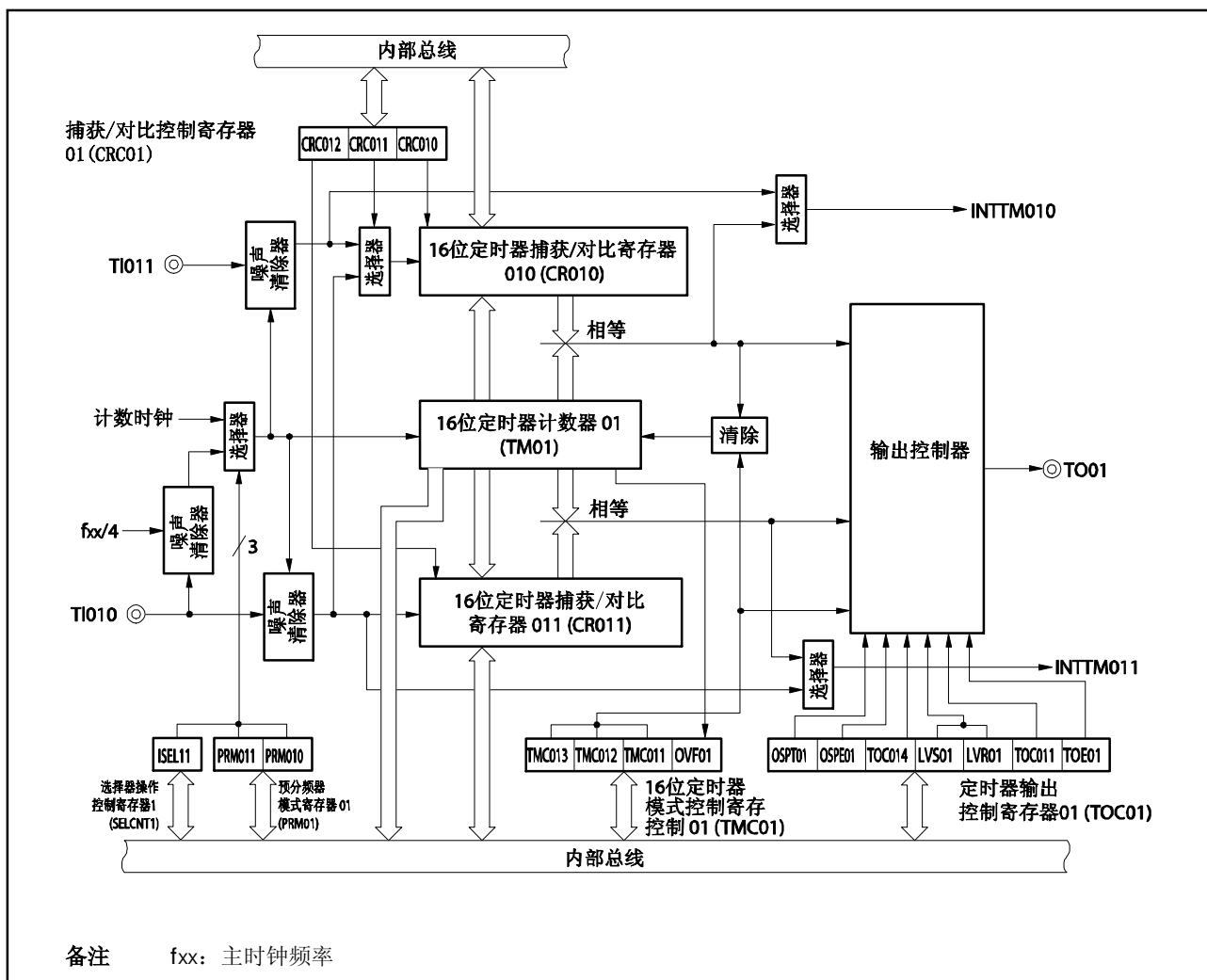
表 7-1. 16 位定时器/事件计数器 01 的配置

项目	配置
定时器/计数器	16 位定时器计数器 01 (TM01)
寄存器	16 位定时器捕捉/比较寄存器: 16 位 × 2 (CR010, CR011)
定时器输入	2 (TI010, TI011 引脚)
定时器输出	1 (TO01 引脚), 输出控制器
控制寄存器 ^注	16 位定时器模式控制寄存器 01 (TMC01) 捕捉/比较控制寄存器 01 (CRC01) 16 位定时器输出控制寄存器 01 (TOC01) 预分频器模式寄存器 01 (PRM01) 选择操作控制寄存器 1 (SELCNT1)

注 TI010, TI011, 及 TO01 引脚功能的使用请参考表 4-12 端口引脚用作复用功能时的设定。

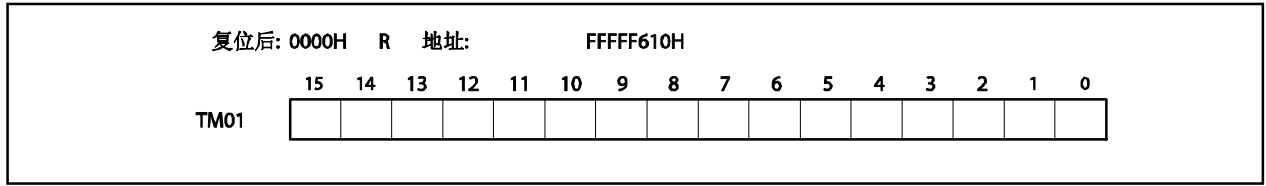
如下所示框图。

图 7-1. 16 位定时器/事件计数器 01 的框图



(1) 16位定时器计数器 01 (TM01)

TM01 寄存器是 16 位只读寄存器，它对计数脉冲进行计数。
该计数器的计数值随计数时钟的上升沿同步增加。



当 TMC01.TMC013 及 TMC01.TMC012 位值非 00 时，可通过 TM01 寄存器来读取其计数值。当读取的 TMC013 及 TMC012 位为 00 时，TM01 寄存器的读取值为 0000H。
在下列情况下计数值复位为 0000H。

- 生成复位信号
- 如果 TMC013 及 TMC012 位为 00
- TI010 引脚的有效边沿输入时清零 并开始的模式时 TI010 引脚输入有效边沿
- TM01 和 CR010 寄存器相等时清零 并开始的模式时 TM01 和 CR010 寄存器相等
- 在单脉冲输出模式下 TOC01.OSPT01 位设置为 1 或输入有效边沿到 TI010 引脚

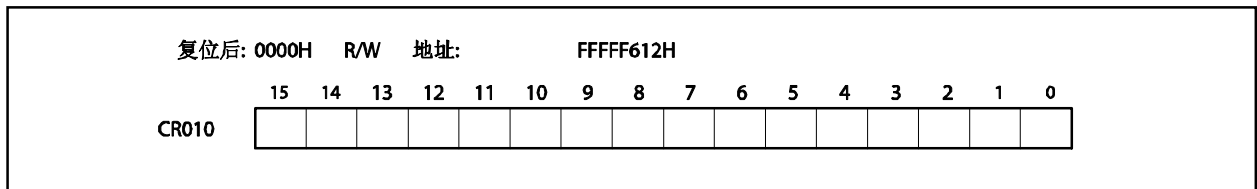
(2) 16位定时器捕捉/比较对比寄存器 010，16位定时器捕捉/比较寄存器 011 (CR011)

CR010 及 CR011 为 16 位寄存器，通过使用 CR010 寄存器可选择具有捕捉功能或比较功能。

当定时器在运行时 (TMC01.TMC013 和 TMC01.TMC012 位 = 00 以外的数) 禁止改变 CR010 寄存器的值。
如果用指定的设置方式，CR011 寄存器的值可以在操作中改变。关于更多细节，敬请参阅 **7.5.1 TM01 操作时改写 CR011 寄存器**。

该寄存器支持 16 位读写方式。

该寄存器复位后的值为 0000H。

(a) 16位定时器捕捉/比较寄存器 010 (CR010)

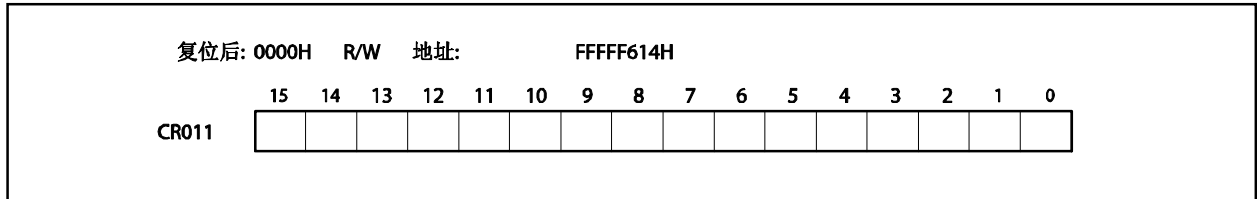
(i) 当 **CR010** 寄存器用作比较寄存器

CR010 寄存器的设置值始终与 TM01 寄存器的计数值对比，如果两值相等，则生成中断请求信号（INTTM010），数值保留至 CR010 寄存器重新写入。

(ii) 当 **CR010** 寄存器用作捕捉寄存器

当捕捉触发器输入，TM01 寄存器的计数值被捕捉入 CR010 寄存器。可以通过 CRC01 或 PRM01 寄存器选择 TI010 引脚的反相边沿或者 TI011 引脚的有效边沿作为捕捉触发源。

(b) 16 位定时器捕捉/比较寄存器 011（CR011）



(i) 当 **CR011** 寄存器用作比较寄存器

CR011 寄存器的设置值始终与 TM01 寄存器的计数值对比，如果两值相等，则生成中断请求信号（INTTM011）。

(ii) 当 **CR011** 寄存器用作捕捉寄存器

通过输入捕捉触发器，可将 TM01 寄存器的计数值捕捉入 CR011 寄存器。
TI010 引脚的有效边沿可选作捕捉触发器。TI010 引脚的有效边沿可用 PRM01 寄存器设置。

- 注意事项**
1. 当 P35 引脚用作 TI010 的有效边沿且使用定时器输出功能，将 P32 引脚设置为定时器输出引脚（TO01）。
 2. 如清零 TMC013 及 TMC012 位与捕捉触发器输入冲突，则捕捉数据不确定。
 3. 将捕捉模式转至对比模式时，先将 TMC013 及 TMC012 位清至 00，然后在修改设置。
除非设备复位，捕捉的数值将保存在 CR010 及 CR011 计数器内。如果转换到对比模式，要先设置对比值。

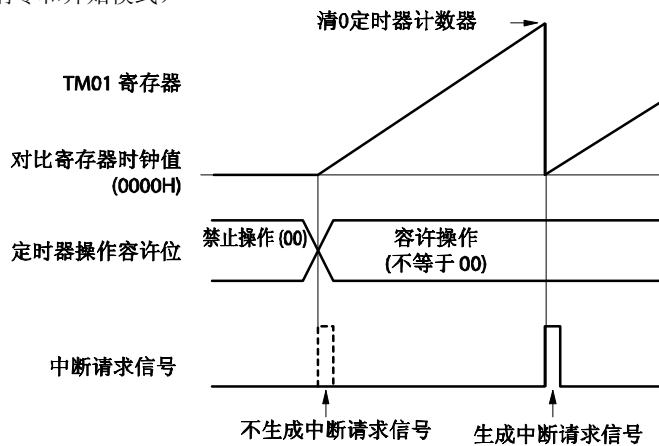
(c) 用作比较寄存器时的设定范围

当 CR010 或 CR011 寄存器用作比较寄存器时，设置如下。

操作	CR010寄存器	CR011寄存器
<ul style="list-style-type: none"> 按间隔定时器操作 按方波输出操作 按外部事件计数器操作 	$0000H \ll N (FFFFH)$	$0000H^{\#} \leq M \leq FFFFH$ 通常情况下，该设置不用。屏蔽相等中断信号 (INTTM011)
<ul style="list-style-type: none"> 通过TI010引脚有效边沿输入在清零并开始的模式下操作 按自由运行定时器操作 	$0000H^{\#} \leq N \leq FFFFH$	$0000H^{\#} \leq M \leq FFFFH$
按PPG输出操作	$M < N \leq FFFFH$	$0000H^{\#} \leq M < N$
按单脉冲输出操作	$0000H^{\#} \leq N \leq FFFFH (N \neq M)$	$0000H^{\#} \leq M \leq FFFFH (M \neq N)$

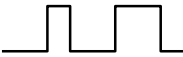
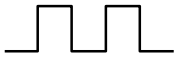


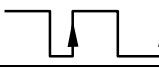



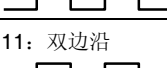



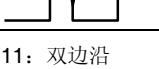
注 当设置 0000H 时，在定时器操作后，相等中断不会马上发生，定时器输出没有改变，第一个相等计时如下。当计数器定时器 (TM01 寄存器) 从 0000H 转至 0001H，在计时时相等中断发生。

- 由于溢出定时器计数器清零
- 由于 TI010 引脚有效边沿 (通过 TI010 引脚有效边沿输入进入清零和开始模式) 使得定时器计数器清零
- 由于对比相等使得定时器计数器清零 (通过 TM01 及 CR01 之间的相等 (CR010 \neq 0000H, CR011 = 0000H) 进入清零和开始模式)



- 备注**
1. N: CR010 寄存器设置值
M: CR011 寄存器设置值
 2. 关于操作容许位 (TMC01.TMC013, TMC01.TMC012 位) 的更多细节，敬请参阅 7.3 (1) 16 位定时器模式控制寄存器 01 (TMC01)。

表 7-2. CR010 及 CR011 寄存器的捕捉操作

外部输入信号 捕捉操作	TI010 引脚输入 		TI011 引脚输入 	
CR010 寄存器的捕捉操作	CRC011 位= 1 TI010 引脚输入 (反相) 	给ES101及 ES100 设值 将捕捉的边沿位置	CRC011位= 0 TI011 引脚输入 	给ES111及 ES110 设值 将捕捉的边沿位置
		01: 上升沿 		01: 上升沿 
		00: 下降沿 		00: 下降沿 
		11: 双边沿 (不能被捕捉)		11: 双边沿 
	中断信号	即使捕捉到定时值也不会生成INTTM010信号	中断信号	每当捕捉到定时值时,就会生成INTTM010信号
CR011 寄存器的捕捉操作	TI010 引脚输入 ^注 	给ES101及 ES100 设值 将捕捉的边沿位置		
		01: 上升沿 		
		00: 下降沿 		
		11: 双边沿 		
	中断信号	每当捕捉到定时值时,就会生成INTTM011信号		

注 CRC011 位的设置对 CR011 寄存器的捕捉操作没有影响。

注意事项 如果用 TI010 引脚输入信号的反相信号来捕捉 TM01 寄存器的计数值并存入 CR010 寄存器, 则捕捉完数值后, 不产生中断请求信号 INTTM010。如果 TI010 引脚输入信号的反相信号作为触发源时检测到 TI011 引脚的有效边沿, 不执行捕捉, 但是产生 INTTM010 信号, 作为外部中断信号。如果不用外部中断, 需要屏蔽 INTTM010 信号。

备注 CRC011: 参见 7.3 (2) 捕捉/比较控制寄存器 01 (CRC01)。
ES111, ES110, ES101, ES100: 参见 7.3 (4) 预分频器模式寄存器 01 (PRM01)。

7.3 寄存器

用以控制 16 位定时器/事件计数器 01 的寄存器如下所示。

- 16 位定时器模式控制寄存器 01 (TMC01)
- 捕捉/比较控制寄存器 01 (CRC01)
- 16 位定时器输出控制寄存器 01 (TOC01)
- 预分频器模式寄存器 01 (PRM01)
- 选择器操作控制寄存器 1 (SELCNT1)

备注 TI010, TI011 及 TO01 引脚功能的使用, 参考表 4-12 端口引脚用作复位功能时的设置。

(1) 16 位定时器模式控制寄存器 01 (TMC01)

TMC01 寄存器有 8 位, 用于设定 16 位定时器/事件计数器 01 的工作模式、TM01 清零模式和输出时序, 并用于检测溢出。

工作时 (TMC013 和 TMC012 位 $\neq 00$) 禁止改写 TMC01。但当 TMC013 和 TMC012 位被清为 00 (停止工作时) 并且当 OVF01 位清零时, 可改写 TMC01。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

- 注意事项**
1. 16 位定时器/事件计数器 01 在 TMC012 和 TMC013 不再为 00 (停止工作模式) 的时刻开始工作。将 TMC012 和 TMC013 置为 00 将使工作停止。
 2. 当主时钟停止副时钟工作时, 不要访问 TMC01 寄存器。
关于更多细节, 敬请参阅 3.4.8 (2)。

复位后: 00H R/W 地址: FFFFF616H

	7	6	5	4	3	2	1	<0>
TMC01	0	0	0	0	TMC013	TMC012	TMC011	OVF01

TMC013	TMC012	16 位定时器/事件计数器 01 的容许操作
0	0	禁止 TM01 操作。 停止提供操作时钟。将 16 位定时器计数器 (TM01) 清零。
0	1	自由运行定时器模式
1	0	通过 TI010 引脚有效边缘输入 ^{注1} 进入清零及开始模式
1	1	通过 TM01 及 CR010 的相等进入清零及开始模式

TMC011 ^{注2}	使定时器输出反相的条件 (TO01)
0	(TM01 与 CR010 的相等或 TM01 与 CR011 的相等
1	(TM01 与 CR010 的相等或 TM01 与 CR011 的相等 (TI010 引脚有效边沿的触发器输入

OVF01	TM01 寄存器溢出标志
清除 (0)	清除 OVF01 至 0 或 TMC01.TMC013 及 TMC01.TMC012 = 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, 通过 TI010 引脚有效边沿输入进入的清零及开始模式, 以及通过 TM01 和 CR010 相等进入的清零及开始模式) TM01 的值从 FFFFH 变为 0000H 时, OVF01 置 1。
通过把 OVF01 位写入 1 来将其设置为 1。

- 注
1. 通过设置 PRM01 来设置 TI010 引脚的有效边沿。
 2. 当交替使用 TO01 引脚及 TI010 引脚时一定要将 TMC011 位清零。

(2) 捕捉/比较控制寄存器 01 (CRC01)

CRC01 寄存器是用于控制 CR010 及 CR011 寄存器操作的。

在操作中严禁改变 CRC01 寄存器的值 (当 TMC01.TMC013 及 TMC01.TMC012 位 ≠ 00)。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: CRC01 FFFFF618H

	7	6	5	4	3	2	1	0
CRC01	0	0	0	0	0	CRC012	CRC011	CRC010

CRC012	CR011 寄存器操作模式选择
0	按比较寄存器操作
1	按捕捉寄存器操作

CRC011	CR010 寄存器捕捉触发选择
0	通过 TI011 引脚有效边沿捕捉
1	通过 TI010 引脚有效边沿的相反相位 ^注 捕捉

通过 PRM01 寄存器设置 TI011 及 TI010 的有效边沿。
当 CRC011 为 1 时, 如果 PRM01.ES101 及 PRM01.ES100 均设置为 11 (双沿), TI010 引脚的有效边沿不能被检测到。

CRC010	CR010 寄存器操作模式选择
0	按比较寄存器操作
1	按捕捉寄存器操作

如果 TMC013 及 TMC012 均设置为 11 (通过相等 TM01 及 CR010 进入清除及开始模式), 一定将 CRC010 位设置为 0。

注 当从 TI011 引脚检测到有效边沿时, 不进行捕捉操作, 但生成作为外部中断信号的 INTTM010 信号。

注意事项 为确保捕捉操作正确进行, 捕捉触发器需要一个比 PRM01 或 SELCNT1 寄存器选择的计数时钟长两个周期的脉冲。

(3) 16 位定时器输出控制寄存器 01 (TOC01)

TOC01 寄存器是控制 TO01 引脚输出的 8 位寄存器。

只有当 OSPT01 位在操作时 (当 TMC01.TMC013 及 TMC01.TMC012 位 ≠ 00)，TOC01 寄存器可被写入。在操作中严禁写入其它位。

然而，TOC014 在定时器操作时，可作为重新写入 CR011 寄存器的一种方式而被重新写入。(参见 7.5.1 **TM01 操作时改写 CR011 寄存器**)。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

注意事项 在以下步骤中一定要设置 TOC01 寄存器。

- <1> TOC014 及 TOC011 位设置为 1。
- <2> 仅将 TOE01 位设置为 1。
- <3> 将 LVS01 位或 LVR01 位设置为 1。

(1/2)

复位后: 00H	R/W	地址: FFFFF619H							
	7	<6>	<5>	4	<3>	<2>	1	<0>	
TOC01	0	OSPT01	OSPE01	TOC014	LVS01	LVR01	TOC011	TOE01	

OSPT01	通过软件触发单脉冲输出
0	-
1	单脉冲输出
该位的读出值始终为“0”。	
如该位设置为 1，TM01 被清零，然后开启。	

OSPE01	单脉冲输出操作控制
0	连续脉冲输出
1	单脉冲输出 如该位设置为 1，TM01 被清零，然后开启。
单脉冲输出在由引脚的有效边沿输入进入的自由运行定时器模式或清零及开启模式下操作正确。	
单脉冲在由 TM01 及 CR010 寄存器相等进入的清零及开启模式下不能输出。	

TOC014	TO01 引脚输出对 CR011 及 TM01 寄存器相等的控制
0	禁止反向操作
1	容许反向操作
即使 TOC014 位=0，中断信号 (INTTM011) 也会生成。	

LVS01	LVR01	设置 TO01 引脚的输出状态
0	0	无变化
0	1	TO01 引脚输出的初始值 偏低 (TO01 引脚输出清零)。
1	0	TO01 引脚输出的初始值 偏高 (TO01 引脚输出设置为 1)。
1	1	禁止设置

- LVS01 及 LVR01 可用来设置 TO01 引脚输出水平的初始值。如初始值无需设置, LVS01 及 LVR01 位为 001。
- 当 TOE01 = 1, 一定要设置 LVS01 及 LVR01 位。
禁止同时设置 LVS01, LVR01 及 TOE01 位为 1。
- LVS01 及 LVR01 位为触发位。通过设置这些位为 1 可对 TO01 引脚输出水平的初始值进行设置。即使这些位清零, TO01 引脚输出不受影响。
- LVS01 及 LVR01 位的读取值始终为 0。
- 关于如何设置 LVS01 及 LVR01 位, 参见 7.5.2 设置 LVS01 及 LVR01 位。

TOC011	TO01 引脚输出控制 CR010 及 TM01 寄存器的相等
0	禁止反向操作
1	容许反向操作

当 TOC011 位=0 时, 中断信号 (INTTM010) 生成。

TOE01	TO01 引脚输出控制
0	禁止输出 (TO01 引脚输出设置为低电平)
1	容许输出

(4) 预分频模式寄存器 01 (PRM01)

PRM01 寄存器是用来设置 TM01 寄存器计数时钟和 TI010 及 TI011 引脚输入有效边沿的寄存器。PRM011 及 PRM010 位与 SELCNT1.ISEL11 位组合在一起被设置。详情参考 7.3 (6) 16 位定时器/事件计数器 01 的计数时钟设置。

工作时 (TMC01.TMC013 和 TMC01.TMC012 位 ≠ 00)，禁止改写 PRM01 寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

注意事项 1. 当设置 PRM011 和 PRM010 位为 11 (指定 TI010 引脚的有效边沿为计数时钟) 时，不要应用如下设置。

- 从 TI010 引脚有效边沿输入清零和开始模式
 - 设置 TI010 引脚为捕捉触发引脚
- 2.** 如果在 TI010 或 TI011 引脚为高电平，且 TI010 或 TI011 引脚的上升沿或双边沿为有效边沿时使能 16 位定时器/事件计数器 01，则 TI010 或 TI011 引脚的高电平会被视为一个上升沿。当 TI010 或 TI011 引脚上拉时注意这点。但在定时器工作停止后再重新使能的情况下，不会检测到上升沿。
- 3.** 当 P35 引脚作为 TI010 的有效边沿且使用定时器输出功能时，设置 P32 引脚为定时器输出引脚 (TO01)。

复位后:	00H	R/W	地址:	FFFFFF617H				
	7	6	5	4	3	2	1	0
PRM01	ES111	ES110	ES101	ES100	0	0	PRM011	PRM010

ES111	ES110	TI011 引脚有效边沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	双边沿

ES101	ES100	TI010 引脚有效边沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	双边沿

备注 要设置 PRM011 和 PRM010 位，参见 7.3 (6) 16 位定时器/事件计数器 01 的计数时钟设置。

(5) 选择器控制寄存器 1 (SELCNT1)

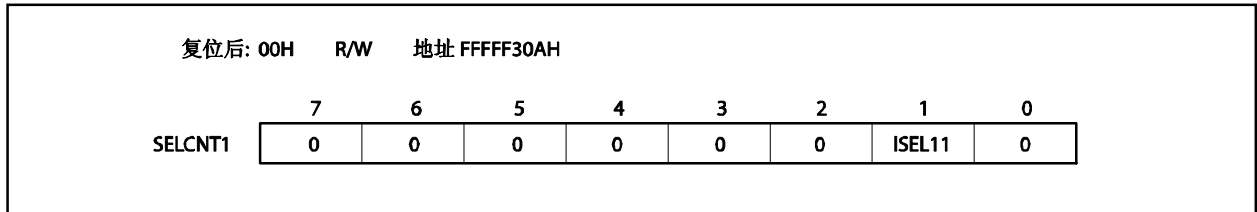
SELCNT1 寄存器设置 16 位定时器/事件计数器 01 的计数时钟。

SELCNT1 寄存器的设置需要与 PRM01.PRM101 和 PRM01.PRM100 位的设置共同进行。详情请参见 7.

(6) 16 位定时器/事件计数器 01 的计数时钟的设置

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。



(6) 16 位定时器/事件计数器 01 计数时钟的设置

16 位定时器/事件计数器 01 计数时钟设置是通过 PRM01.PRM011, PRM01.PRM010 和 SELCNT1.ISEL11 位共同进行的。

SELCNT1 寄存器	PRM01 寄存器		计数时钟 ^{#1} 的选择					
	ISEL11 位	PRM011 位	PRM010 位	计数时钟	f _{xx} = 20 MHz	f _{xx} = 16 MHz	f _{xx} = 10 MHz	
0	0	0	f _{xx}	禁止设置	禁止设置	100 ns		
0	0	1	f _{xx} /4	200 ns	250 ns	400 ns		
0	1	0	INTWT	-	-	-		
0	1	1	Ti0101 ^{#2} 的有效边沿	-	((
1	0	0	f _{XX} /2	100 ns	125 ns	200 ns		
1	0	1	f _{XX} /8	400 ns	500 ns	800 ns		
1	1	0	f _{XX} /16	800 ns	1.0 (s)	1.6 (s)		
1	1	1	禁止设置					

- 注
1. 选择内部时钟时，请按此设置以满足如下要求：
 V_{DD} = 4.0 到 5.5 V: 计数时钟 ≤ 10 MHz
 V_{DD} = 2.7 到 4.0 V: 计数时钟 ≤ 5 MHz
 2. 外部时钟脉冲要大于内部时钟 (f_{xx}/4) 周期的 2 倍。

7.4 操作

7.4.1 间隔定时器操作

如果 TMC01.TMC013 和 TMC01.TMC012 位设置为 11，（TM01 寄存器和 CR010 寄存器相等时进入清零和开始模式），计数操作的开始与计数时钟是同步的。

当 TM01 寄存器计数至与 CR010 寄存器相等，TM01 寄存器清为 0000H，并产生相等中断信号 (INTTM010)。INTTM010 信号使 TM01 寄存器作为间隔定时器工作。

- 备注**
1. 关于复用功能引脚设置，参见表 4-12 端口引脚使用复用功能时的设置。
 2. 关于 INTTM010 中断的使能，参见第十七章 中断/异常处理功能。

图 7-2. 间隔定时器操作框图

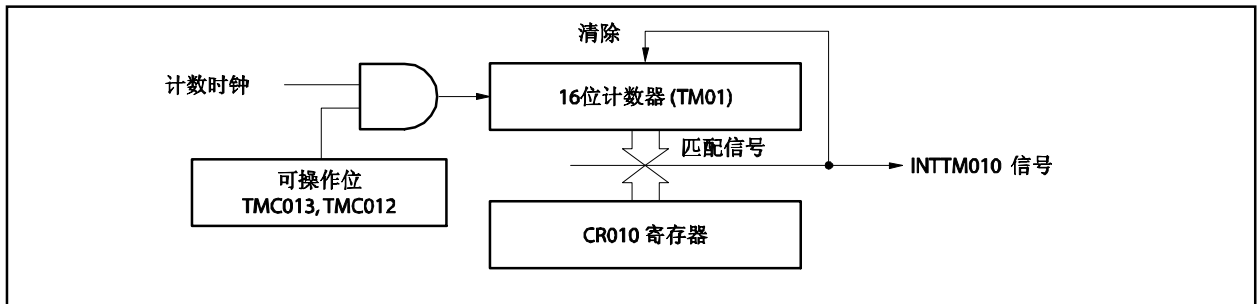


图 7-3. 间隔定时器操作的基本时序示例

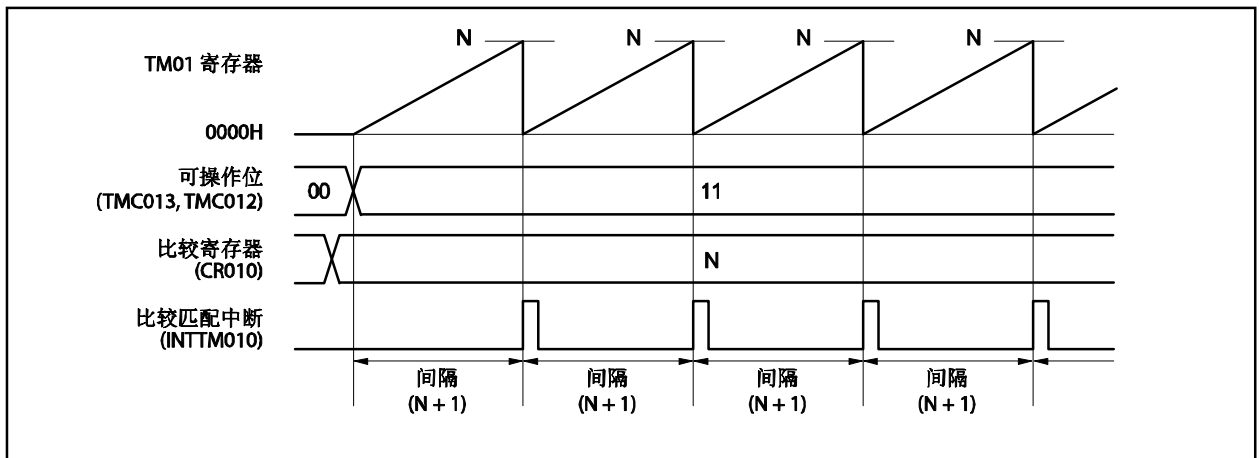


图 7-4. 间隔定时器操作的寄存器设置示例

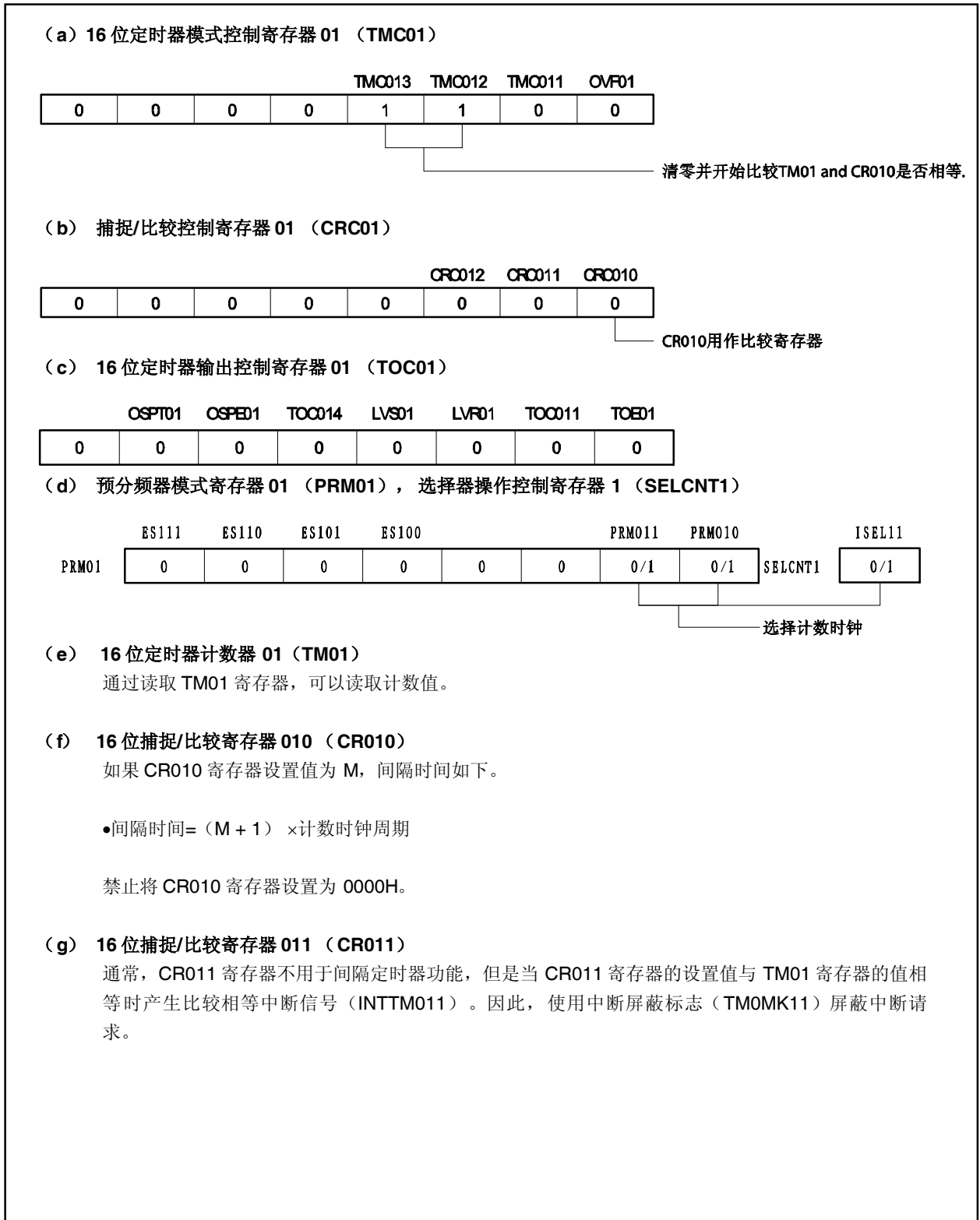
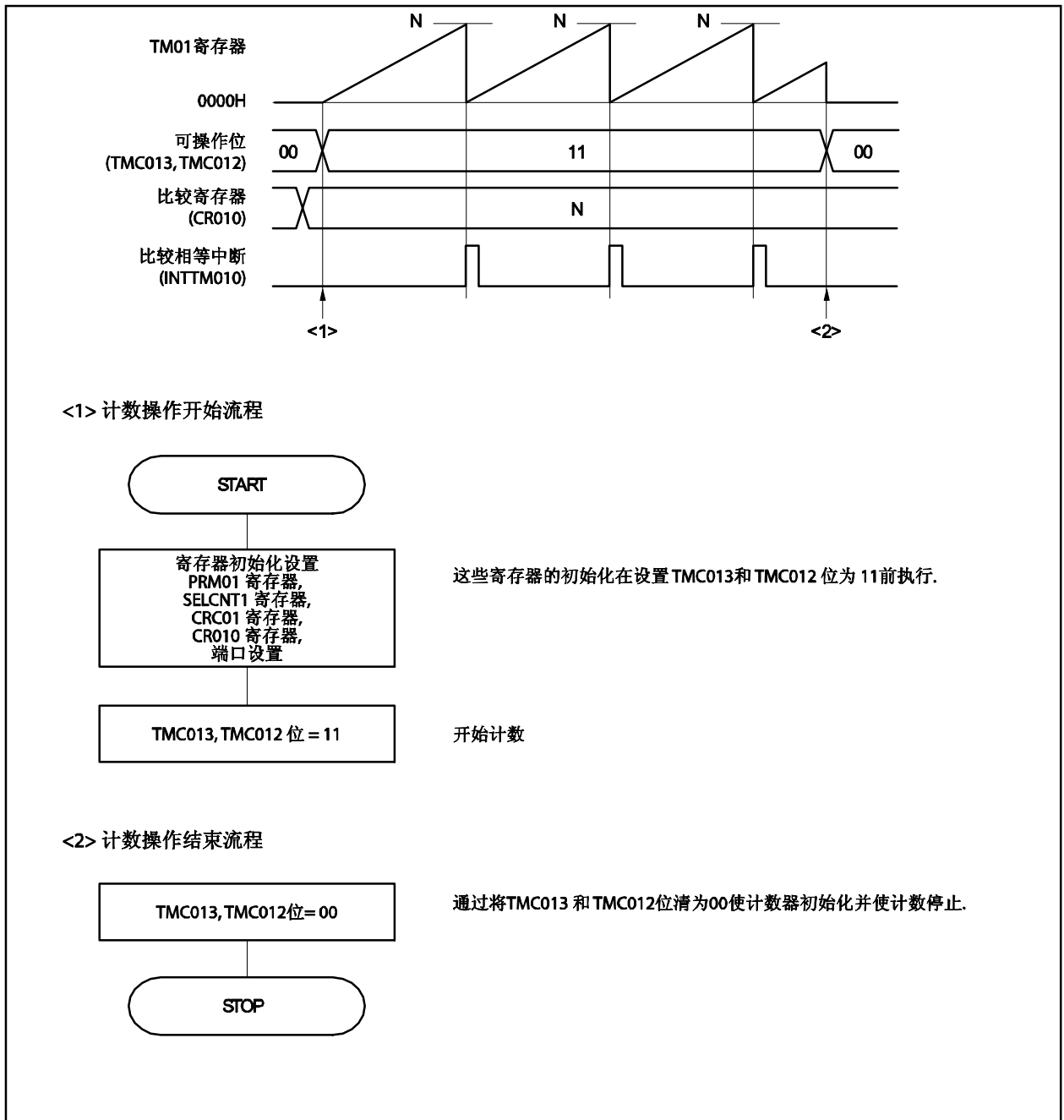


图 7-5. 间隔定时器功能的软件处理示例



7.4.2 方波输出操作

当 16 位定时器/事件计数器 01 作为间隔定时器工作时（参见 7.4.1），如果设置 TOC01 寄存器为 03H，TO01 引脚可以输出方波。

当 TMC01.TMC013 和 TMC01.TMC012 位设置为 11 时（TM01 寄存器和 CR010 寄存器相等时进入计数清零和开始模式），计数操作的开始与计数时钟同步。

TM01 寄存器计数至与 CR010 寄存器相等时，TM01 寄存器清为 0000H，产生中断信号（INTTM010），TO01 引脚的输出反相。这样，TO01 引脚输出值以固定的间隔反相，使得 TO01 输出方波。

- 备注**
1. 关于复用功能引脚设置，参见表 4-12 端口引脚使用复用功能时的设置。
 2. 关于 INTTM010 中断的使能，参见第十七章 中断/异常处理功能。

图 7-6. 方波输出操作框图

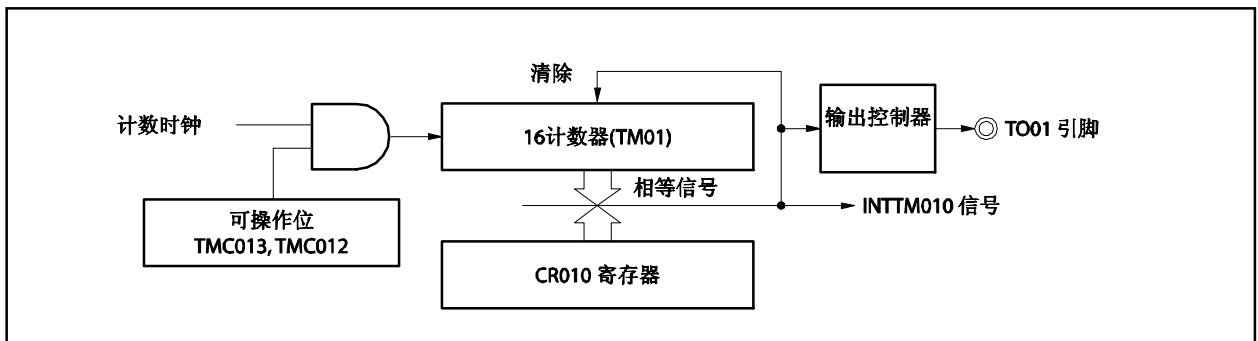


图 7-7. 方波输出操作基本时序示例

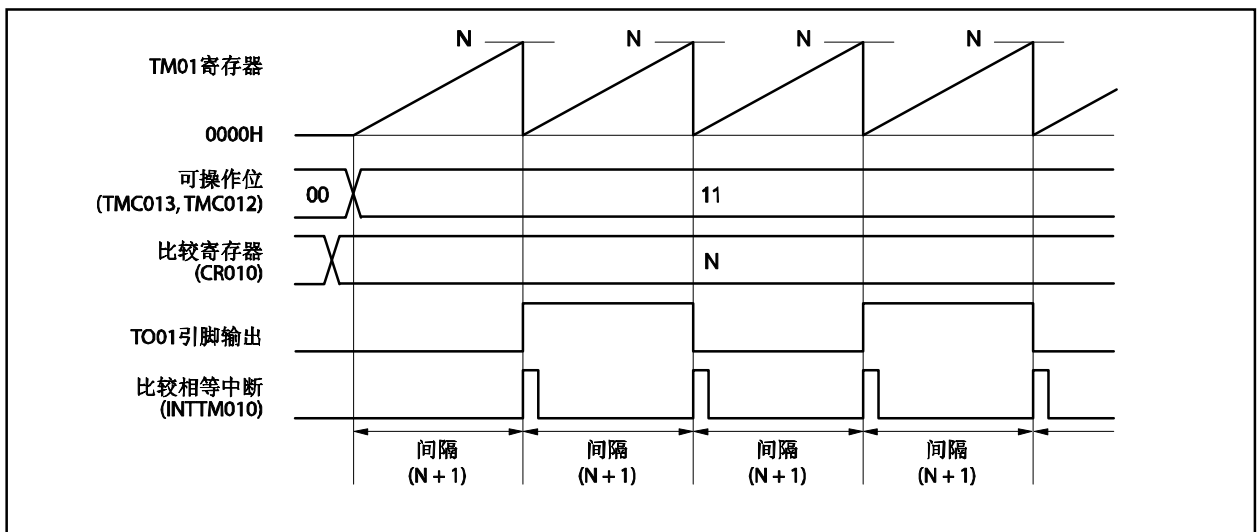


图 7-8. 方波输出操作寄存器设置示例

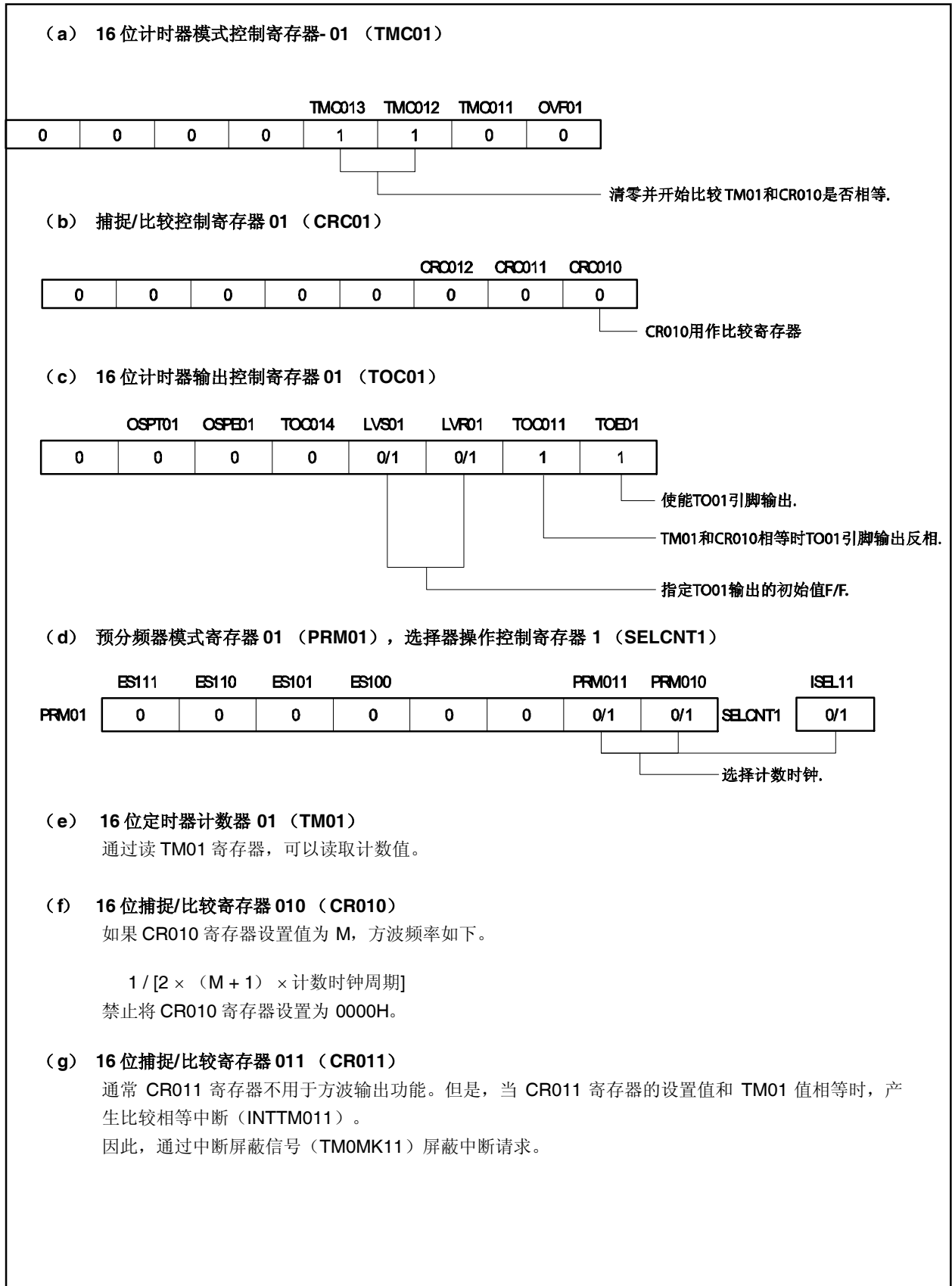
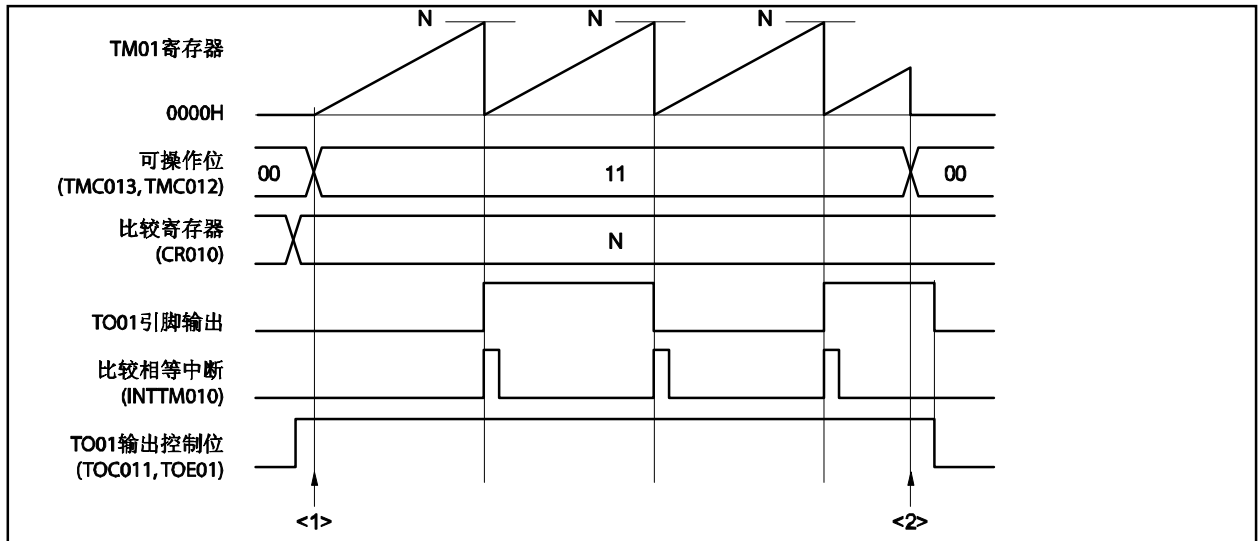


图 7-9. 方波输出功能的软件处理示例



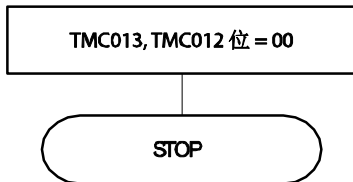
<1> 计数操作开始流程



这些寄存器的初始化在设置TMC013和TMC012位为11前执行。

开始计数操作。

<2> 计数操作结束流程



通过将TMC013 和 TMC012位清为00使计数器初始化并使计数停止。

注 需要特别注意 TOC01 寄存器的设置。关于更多细节，敬请参阅 7.3 (3) 16位定时器输出控制寄存器 01 (TOC01)。

7.4.3 外部事件计数器的操作

当 PRM01.PRM011 和 PRM01.PRM010 位设置为 11（在 TI010 引脚的有效边沿时向上计数）且 TMC01.TMC013 和 TMC01.TMC012 位设置为 11 时，外部事件输入的有效边沿被计数，TM01 寄存器和 CR010 寄存器相等时，产生相等中断信号（INTTM010）。

外部事件需要从 TI010 引脚输入。因此，通过 TI010 引脚输入有效边沿（TMC013 和 TMC012 位 = 10）进入清零和开始模式时，定时器/事件计数器不能作为外部事件计数器。

INTTM010 信号由如下时间产生。

- INTTM010 信号产生的时间（第 2 次或其后）
=检测到外部事件的有效边沿次数×（CR010 寄存器的设置值+1）

但定时器/事件计数器启动后的第 1 次相等中断是由如下时间产生的。

- 检测到外部事件输入的有效边沿次数×（CR010 寄存器的设置值+2）

为了检测有效边沿，输入到 TI010 引脚的信号在 fPRS 的时钟周期内采样。连续检测到两次才视为有效边沿。因此，脉冲较窄的噪声会被消除。

- 备注**
1. 关于复用功能引脚（TI010）设置，参见表 4-12 端口引脚使用复用功能时的设置。
 2. 关于 INTTM010 中断的使能，参见第十七章 中断/异常处理功能。

图 7-10. 外部事件计数器操作框图

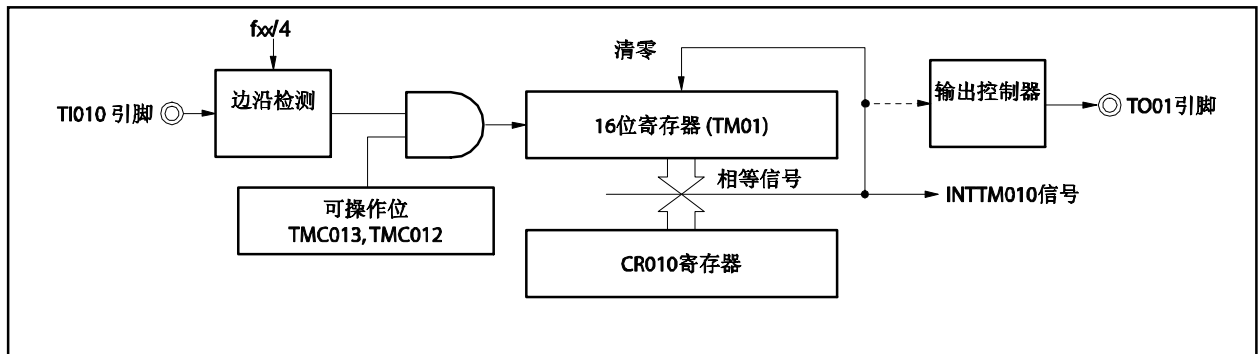


图 7-11. 外部事件计数器模式的寄存器设置示例

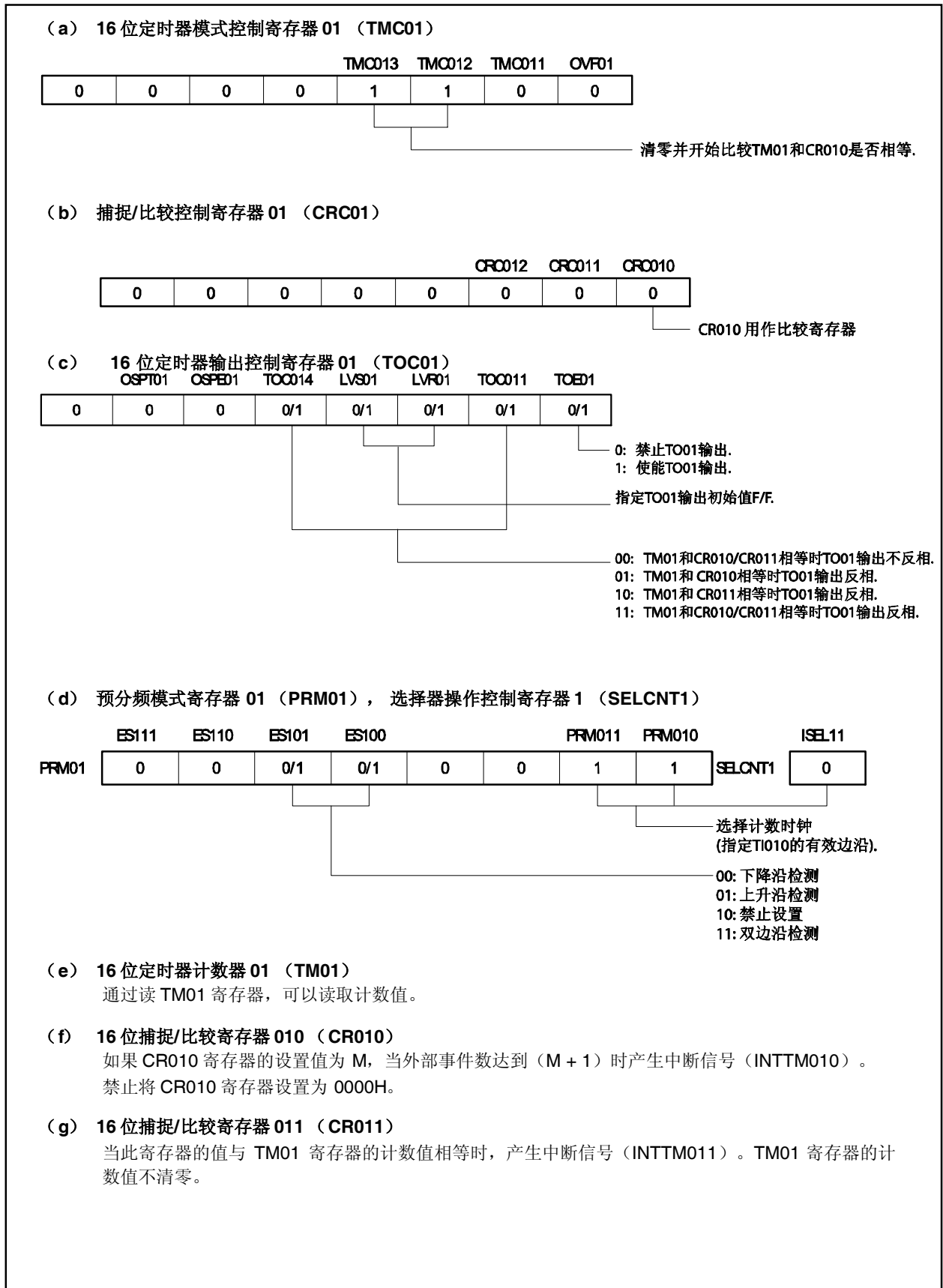
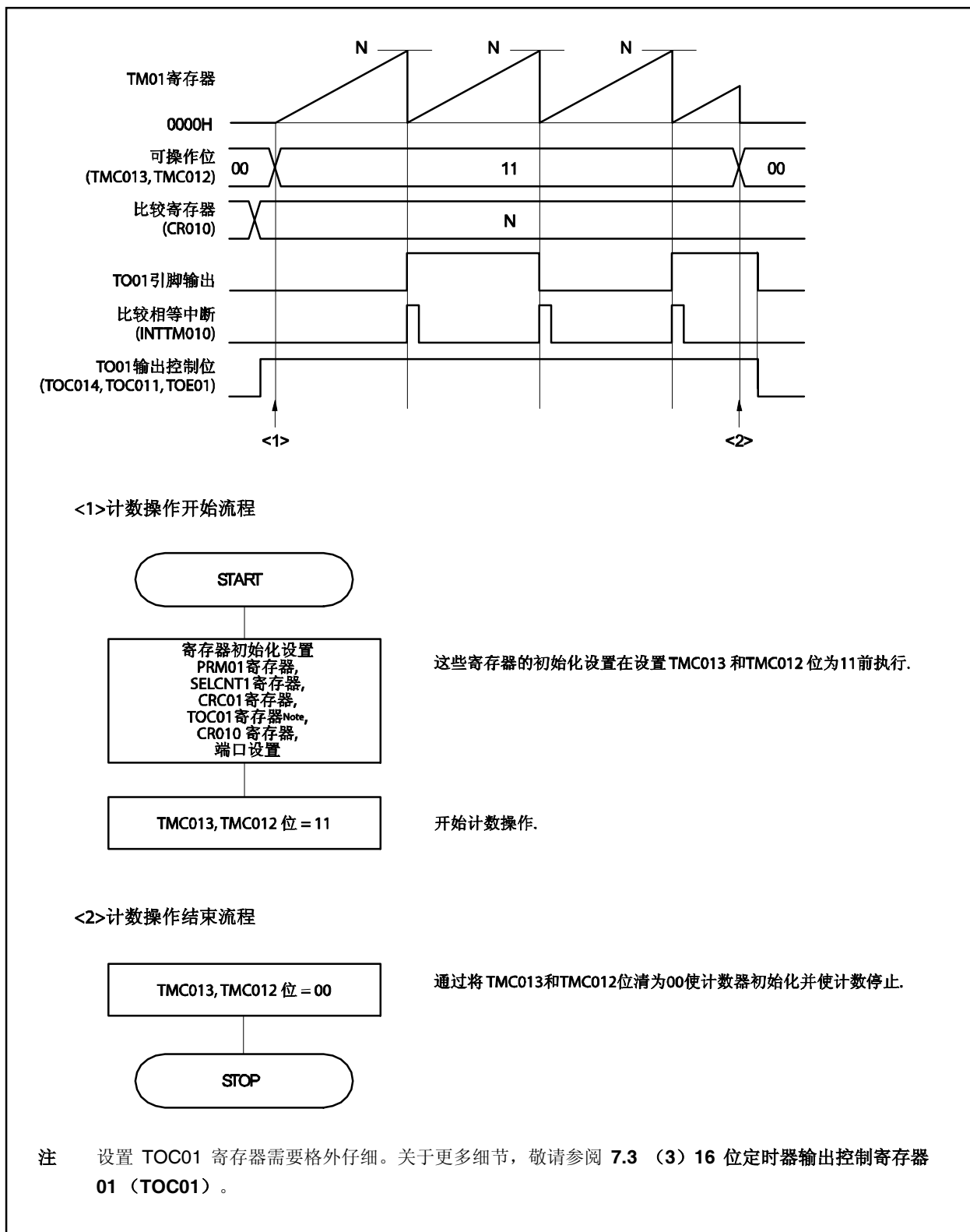


图 7-12. 外部事件计数器模式的软件处理示例



7.4.4 通过 TI010 引脚有效边沿的输入进入清零和开始模式的操作

当 TMC01.TMC013 和 TMC01.TMC012 位设置为 10（通过 TI010 引脚有效边沿的输入进入清零和开始模式）且在定时器/事件计数器上加计数时钟（由 PRM01， SELCNT1 寄存器设置）时，TM01 寄存器开始向上计数。如果在计数时检测到 TI010 引脚的有效边沿，TM01 寄存器清为 0000H 并重新开始向上计数。如果没有检测到 TI010 有效边沿，TM01 寄存器溢出后继续计数。

TI010 引脚的有效边沿是清零 TM01 寄存器的一种方式。操作刚开始时，无法控制计数器的启动。

CR010 和 CR011 寄存器作为比较寄存器或捕捉寄存器。

(a) CR010 和 CR011 寄存器作为比较寄存器时

TM01 寄存器与 CR010 和 CR011 寄存器的值相等时产生 INTTM010 和 INTTM011 信号。

(b) CR010 和 CR011 寄存器作为捕捉寄存器时

TM01 寄存器的计数值捕捉入 CR010 寄存器且当 TI011 引脚输入有效边沿（或该有效边沿的反相信号）时产生 INTTM010 信号。

当 TI010 引脚输入有效边沿时，TM01 寄存器的计数值被 CR011 捕捉且产生 INTTM011 信号。一旦计数值被捕捉，计数器立即清为 0000H。

注意事项 不要设置计数时钟为 TI010 引脚的有效边沿（RPM01.PRM011 和 RPM01.PRM010 位 = 11）。PRM011 和 PRM010 位为 11 时，TM01 寄存器清零。

备注

1. 关于复用功能引脚的设置，参见表 4-12 端口引脚使用复用功能时的设置。
2. 关于 INTTM010 中断的使能，参见第十七章 中断/异常处理功能。

- (1) 过 TI010 引脚输入有效边沿进入的清零和开始模式的操作
 (CR010 寄存器: 比较寄存器, CR011 寄存器: 比较寄存器)

图 7-13. 通过 TI010 引脚输入有效边沿进入的清零和开始模式框图
 (CR010 寄存器: 比较寄存器, CR011 寄存器: 比较寄存器)

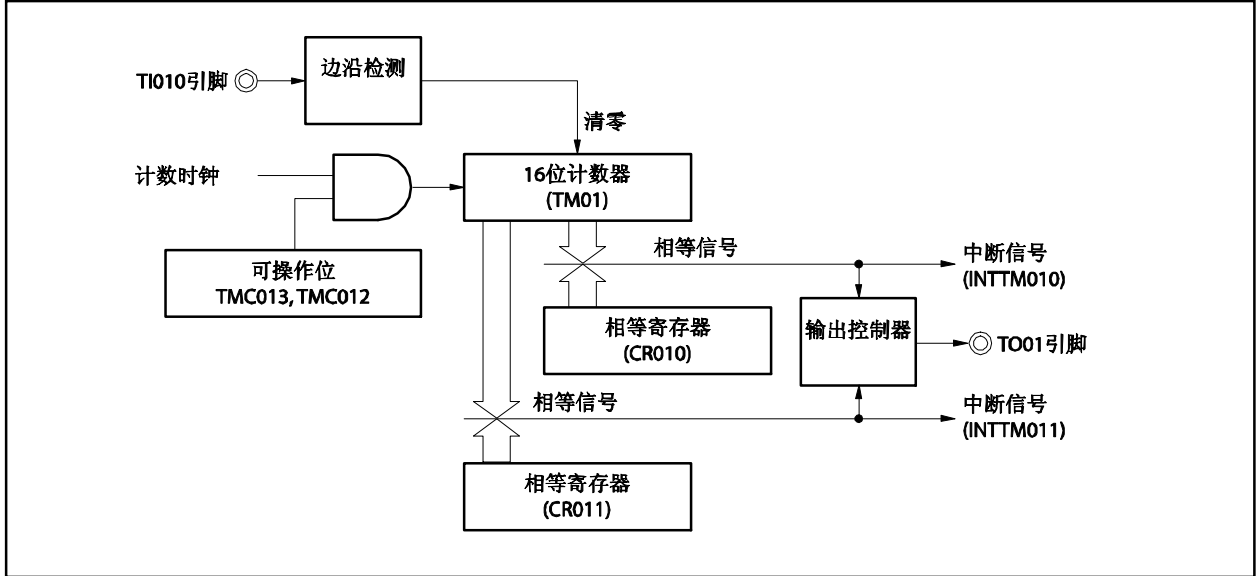
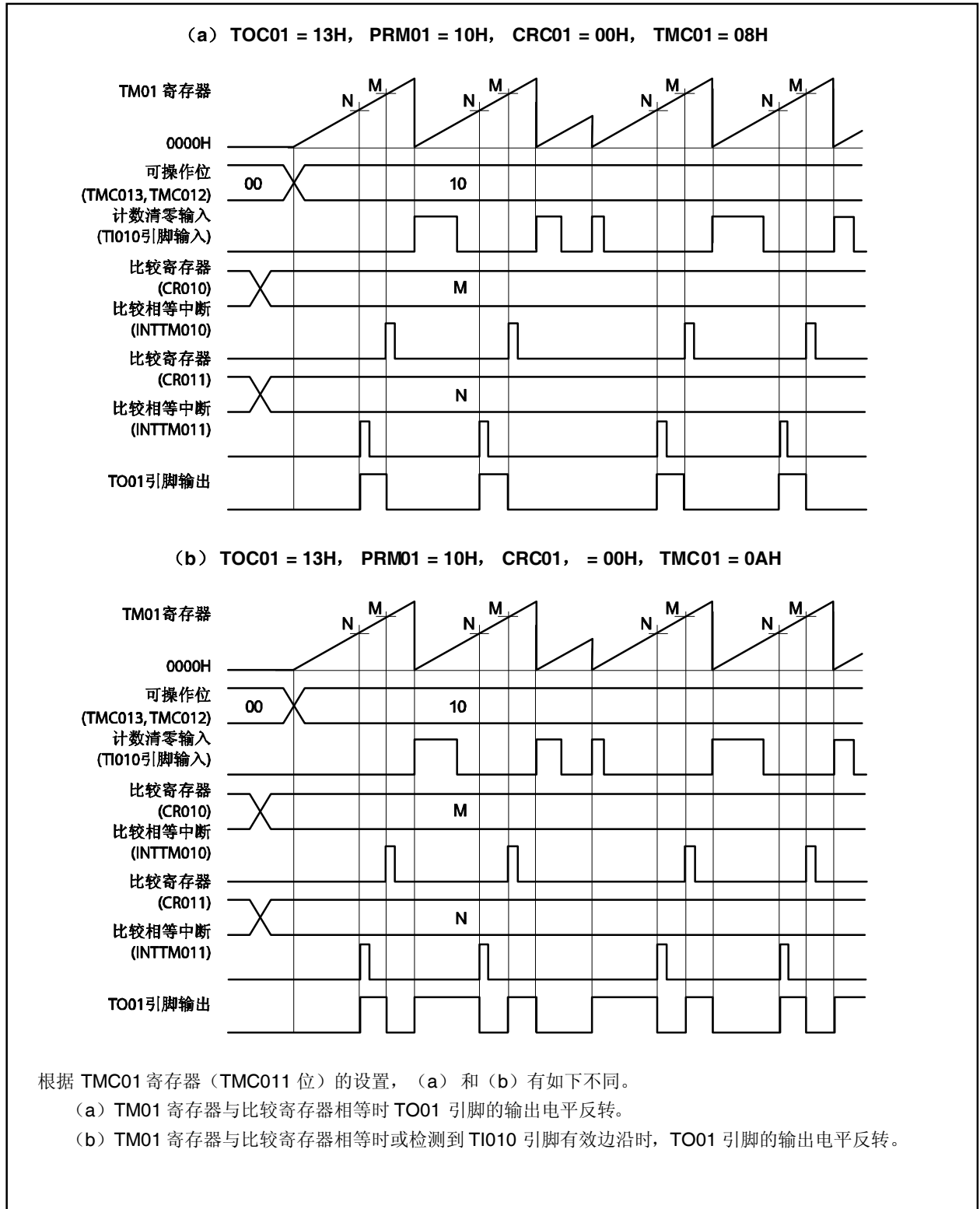


图 7-14. 通过 TI010 引脚输入有效边沿进入的清零和开始模式时序示例
(CR010 寄存器: 比较寄存器, CR011 寄存器: 比较寄存器)



- (2) 通过 TI010 引脚输入有效边沿进入的清零和开始模式的操作
 (CR010 寄存器: 比较寄存器, CR011 寄存器: 捕捉寄存器)

图 7-15. 通过 TI010 引脚输入有效边沿进入的清零和开始模式框图
 (CR010 寄存器: 比较寄存器, CR011 寄存器: 捕捉寄存器)

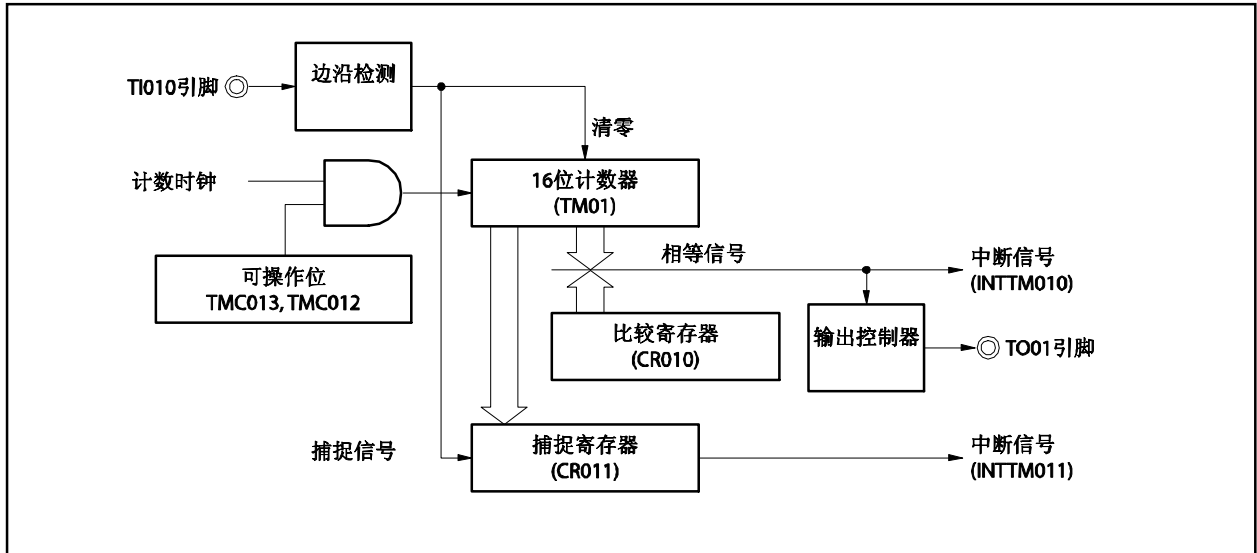


图 7-16. 通过 TI010 引脚输入有效边沿进入的清零和开始模式时序
 (CR010 寄存器: 比较寄存器, CR011 寄存器: 捕捉寄存器) (1/2)

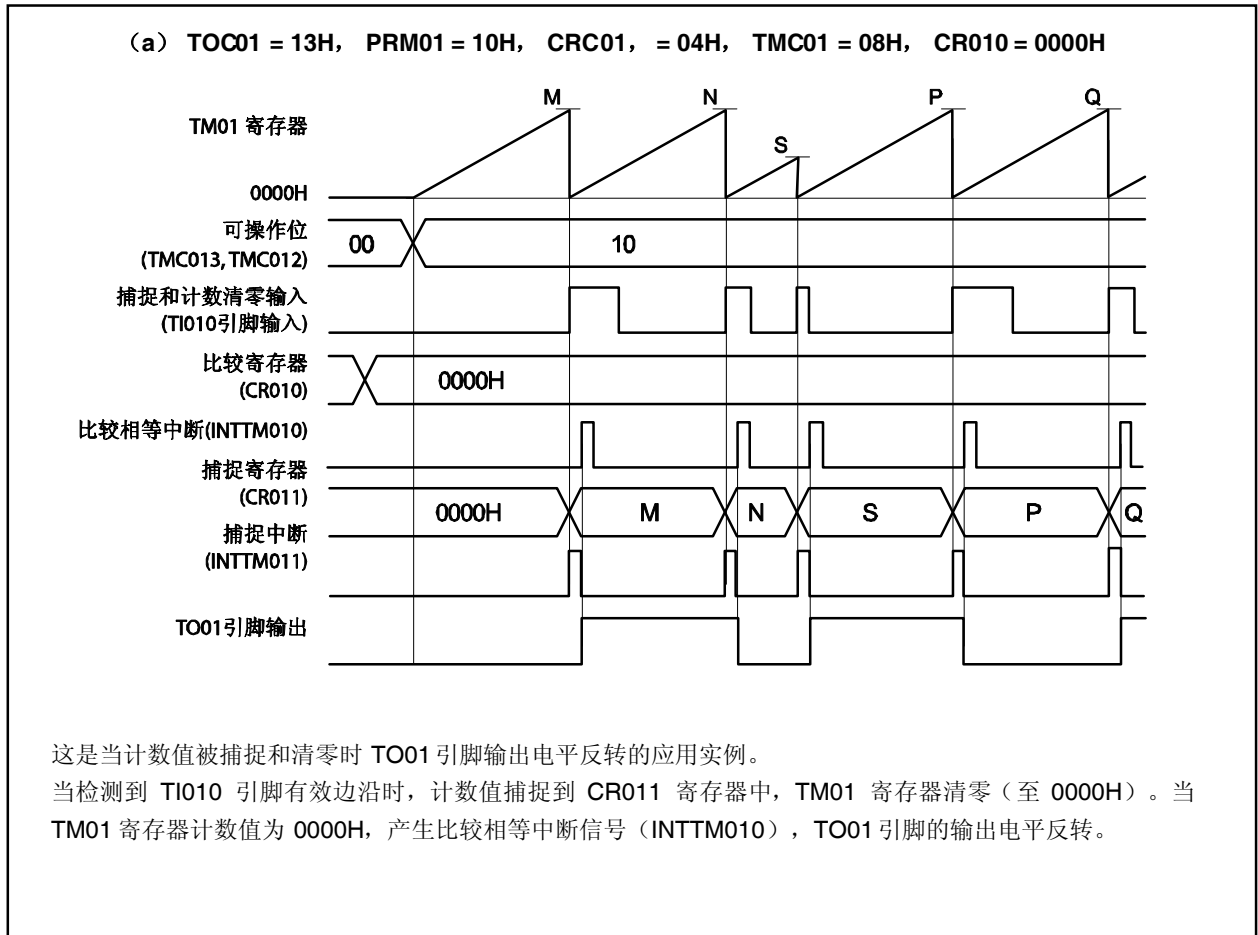
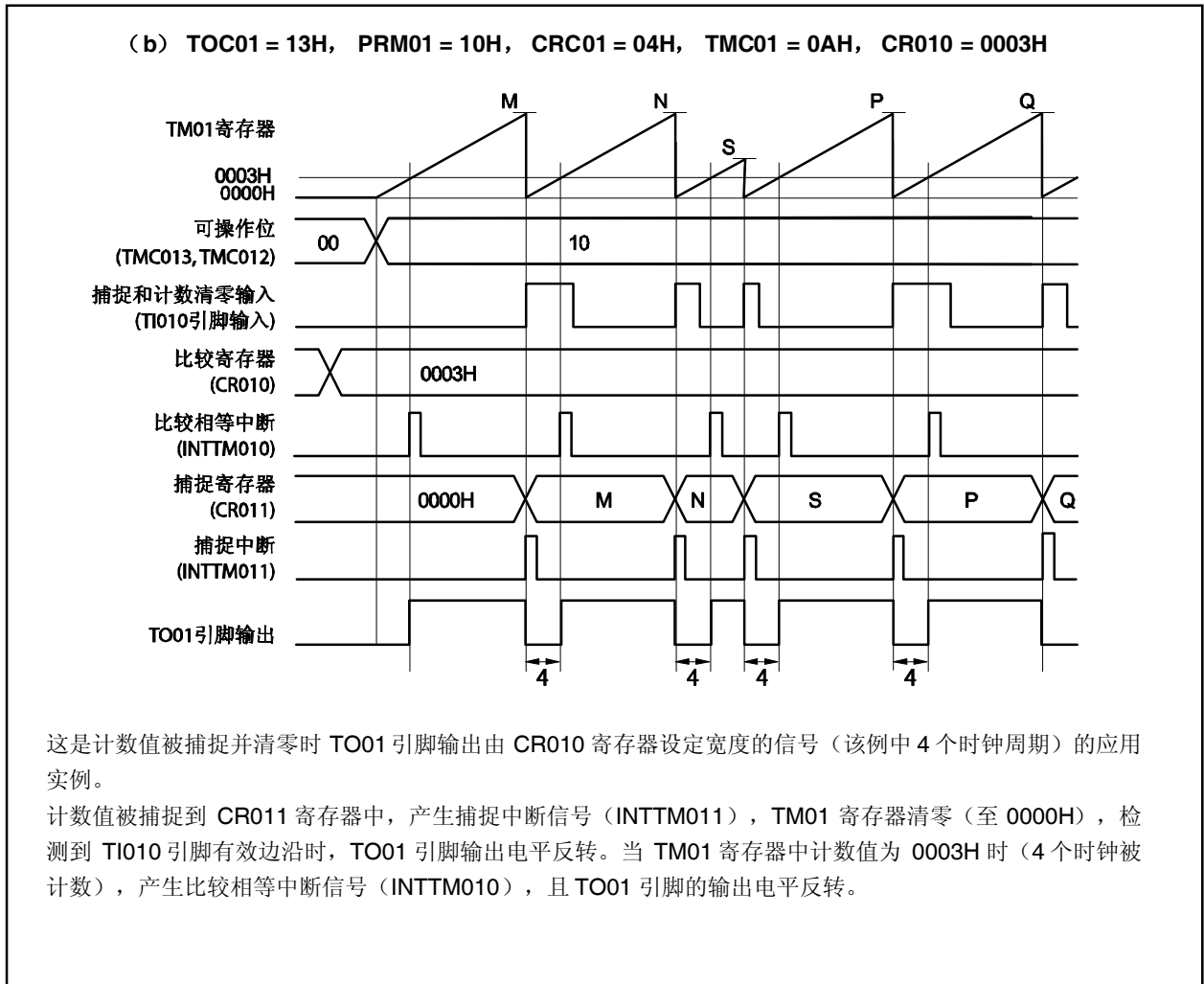


图 7-16. 通过 TI010 引脚输入有效边沿进入的清零和开始模式时序
 (CR010 寄存器: 比较寄存器, CR011 寄存器: 捕捉寄存器) (2/2)



这是计数值被捕捉并清零时 TO01 引脚输出由 CR010 寄存器设定宽度的信号（该例中 4 个时钟周期）的应用实例。

计数值被捕捉到 CR011 寄存器中，产生捕捉中断信号 (INTTM011)，TM01 寄存器清零（至 0000H），检测到 TI010 引脚有效边沿时，TO01 引脚输出电平反转。当 TM01 寄存器中计数值为 0003H 时（4 个时钟被计数），产生比较相等中断信号 (INTTM010)，且 TO01 引脚的输出电平反转。

- (3) 通过 TI010 引脚输入有效边沿进入的清零和开始模式的操作
 (CR010 寄存器: 捕捉寄存器, CR011 寄存器: 比较寄存器)

图 7-17. 通过 TI010 引脚输入有效边沿进入的清零和开始模式框图
 (CR010 寄存器: 捕捉寄存器, CR011 寄存器: 比较寄存器)

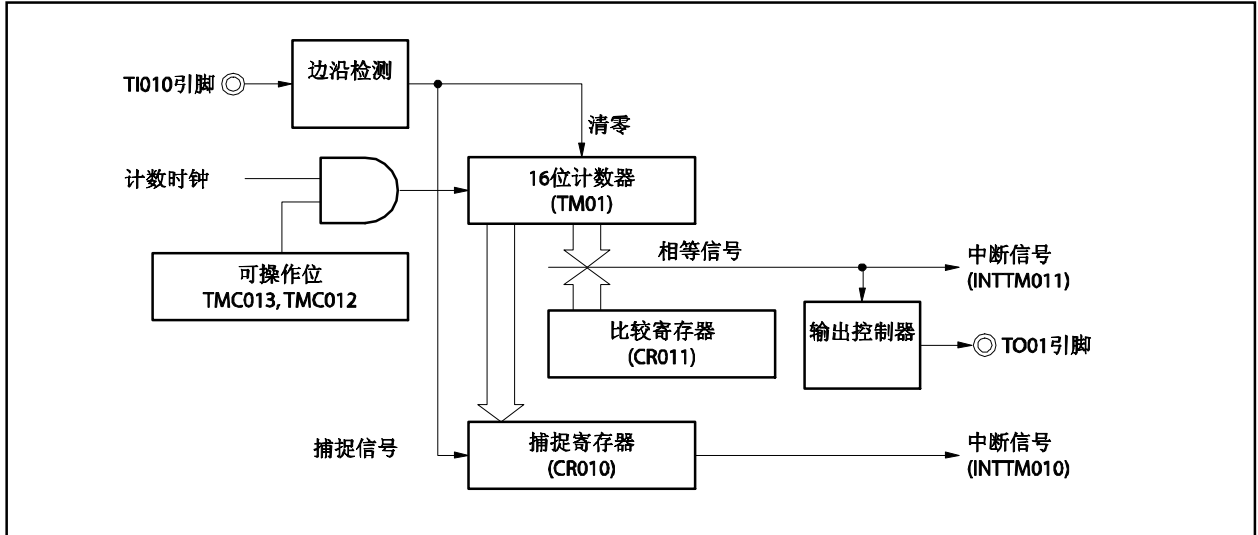


图 7-18. 通过 TI010 引脚输入有效边沿进入的清零和开始模式时序
 (CR010 寄存器: 捕捉寄存器, CR011 寄存器: 比较寄存器) (1/2)

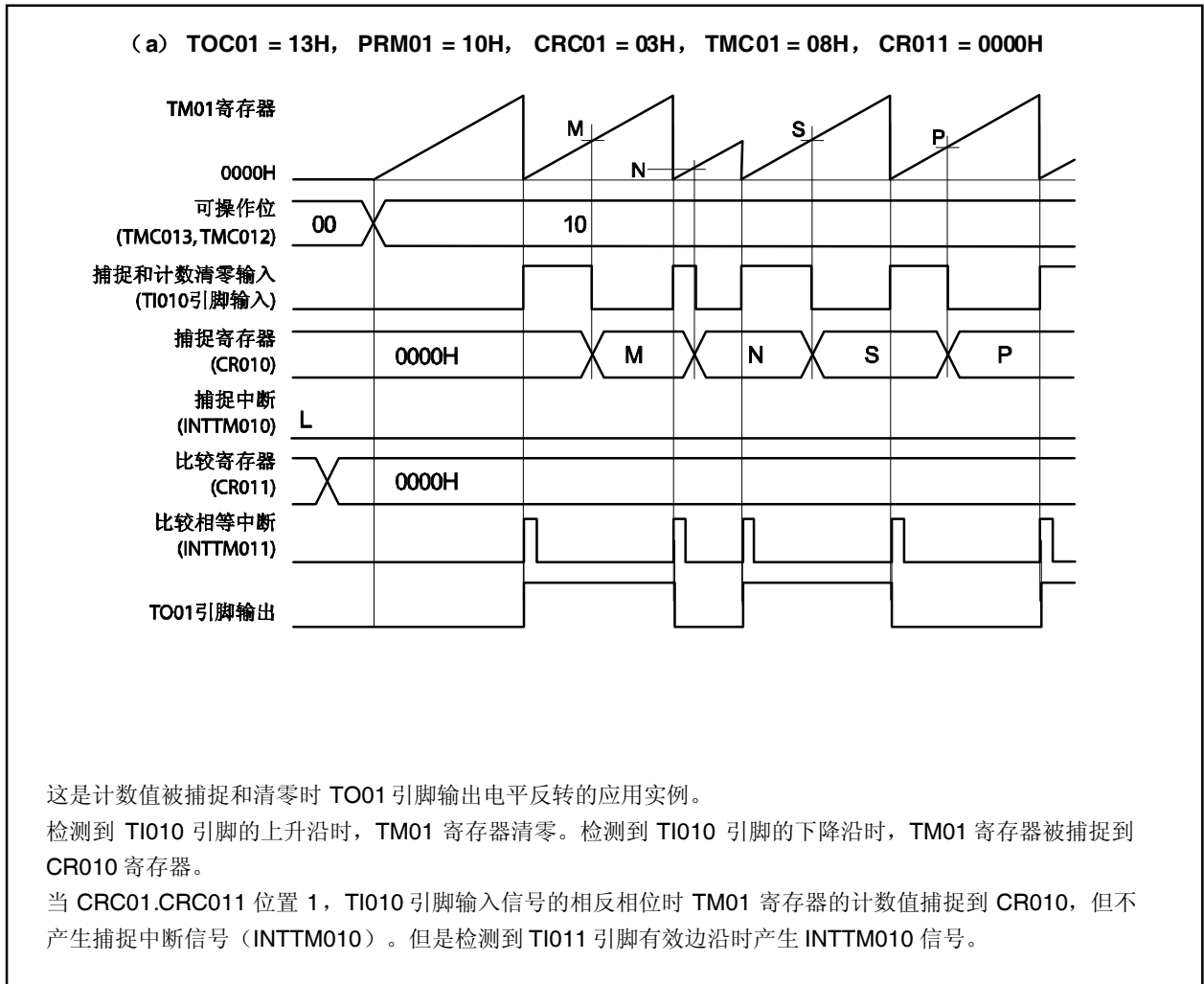
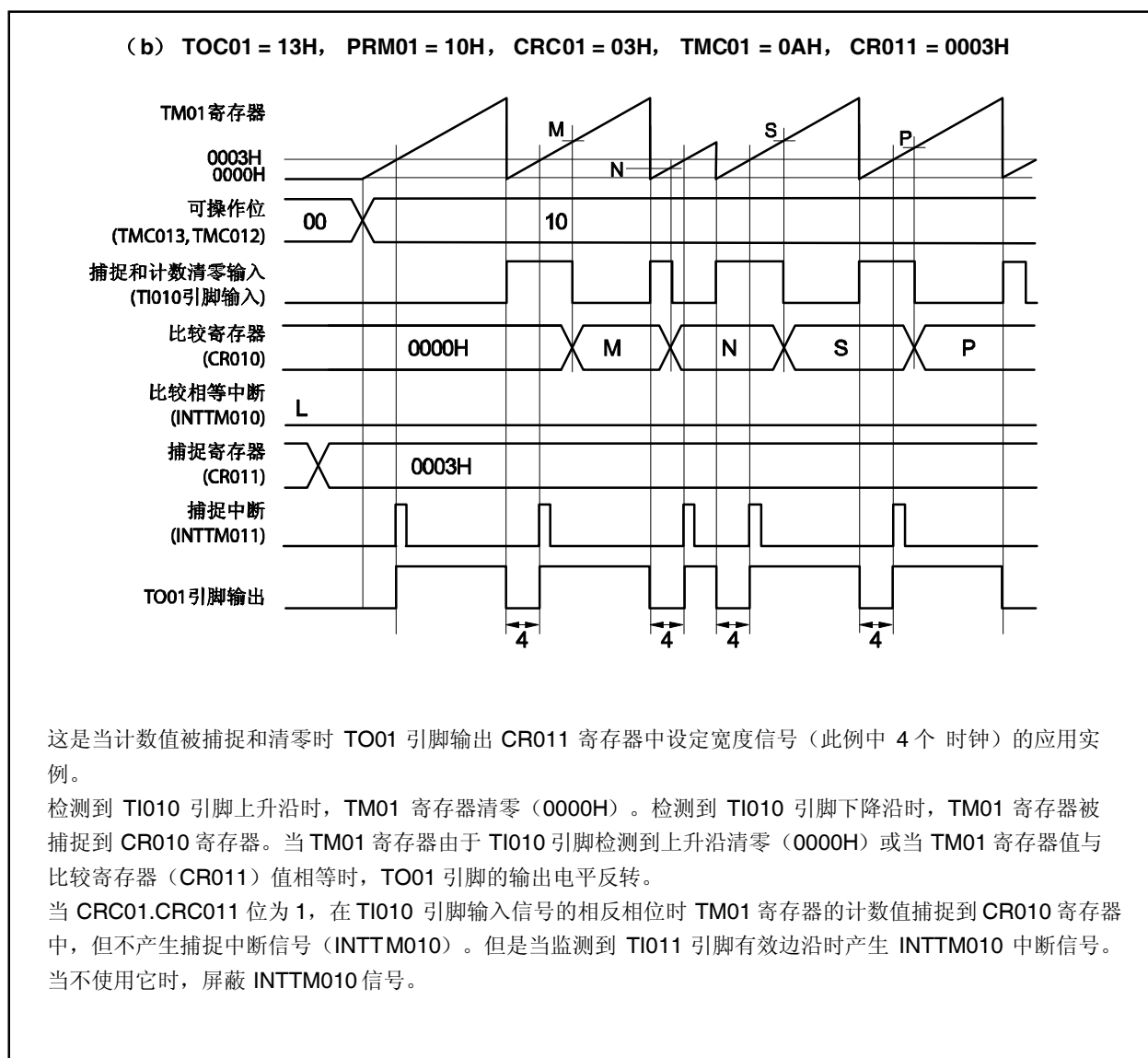


图 7-18. 通过 TI010 引脚输入有效边沿进入的清零和开始模式时序
(CR010 寄存器: 捕捉寄存器, CR011 寄存器: 比较寄存器) (2/2)



- (4) 通过 TI010 引脚输入有效边沿进入的清零和开始模式的操作
 (CR010 寄存器: 捕捉寄存器, CR011 寄存器: 捕捉寄存器)

图 7-19. 通过 TI010 引脚输入有效边沿进入的清零和开始模式框图
 (CR010 寄存器: 捕捉寄存器, CR011 寄存器: 捕捉寄存器)

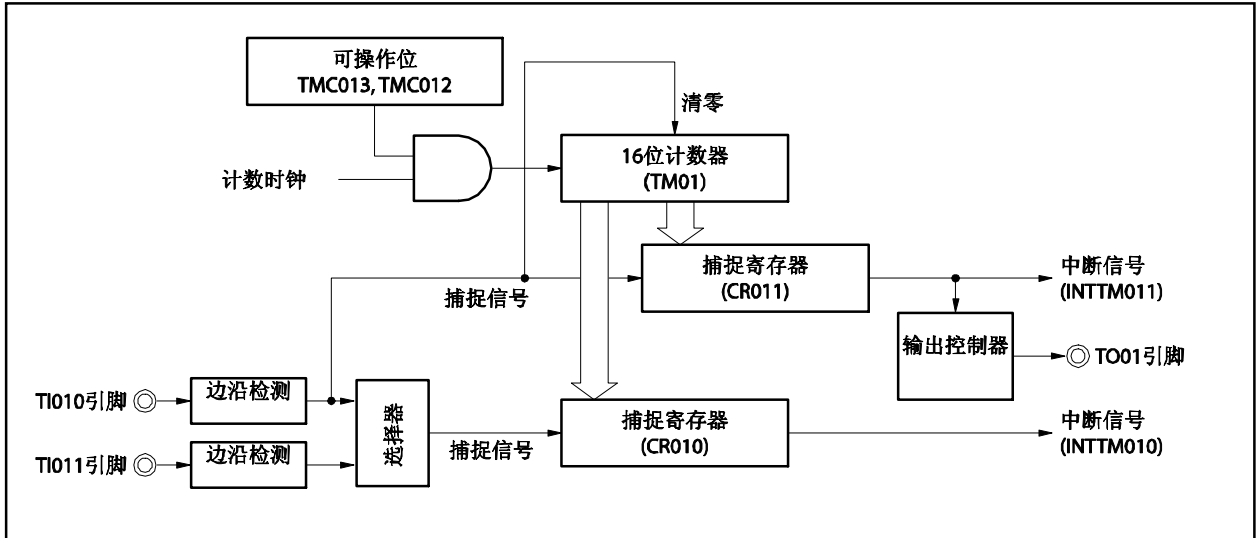


图 7-20. 通过 TI010 引脚输入有效边沿进入的清零和开始模式的时序示例
 (CR010 寄存器: 捕捉寄存器, CR011 寄存器: 捕捉寄存器) (1/3)

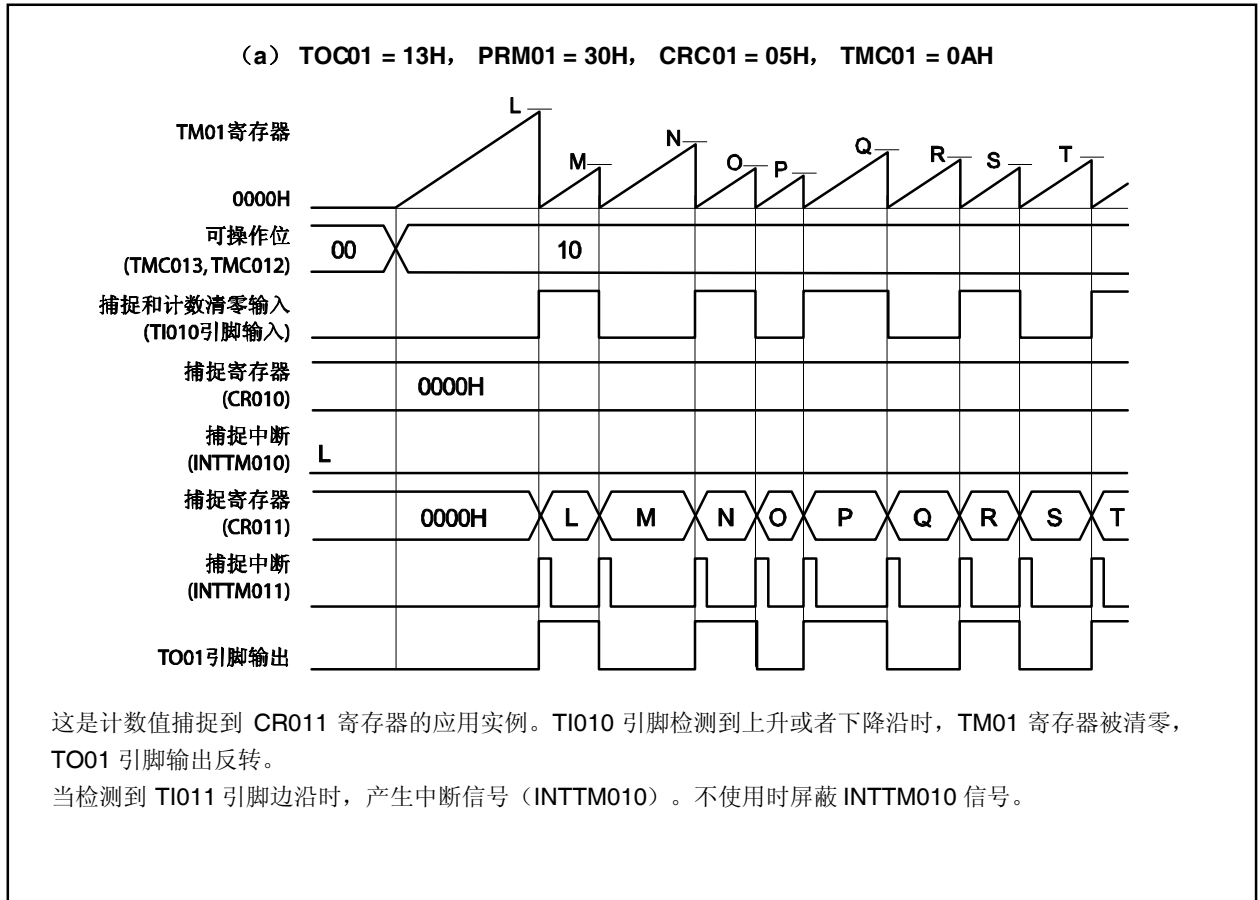


图 7-20. 通过 TI010 引脚输入有效边沿进入的清零和开始模式的时序示例
 (CR010 寄存器: 捕捉寄存器, CR011 寄存器: 捕捉寄存器) (2/3)

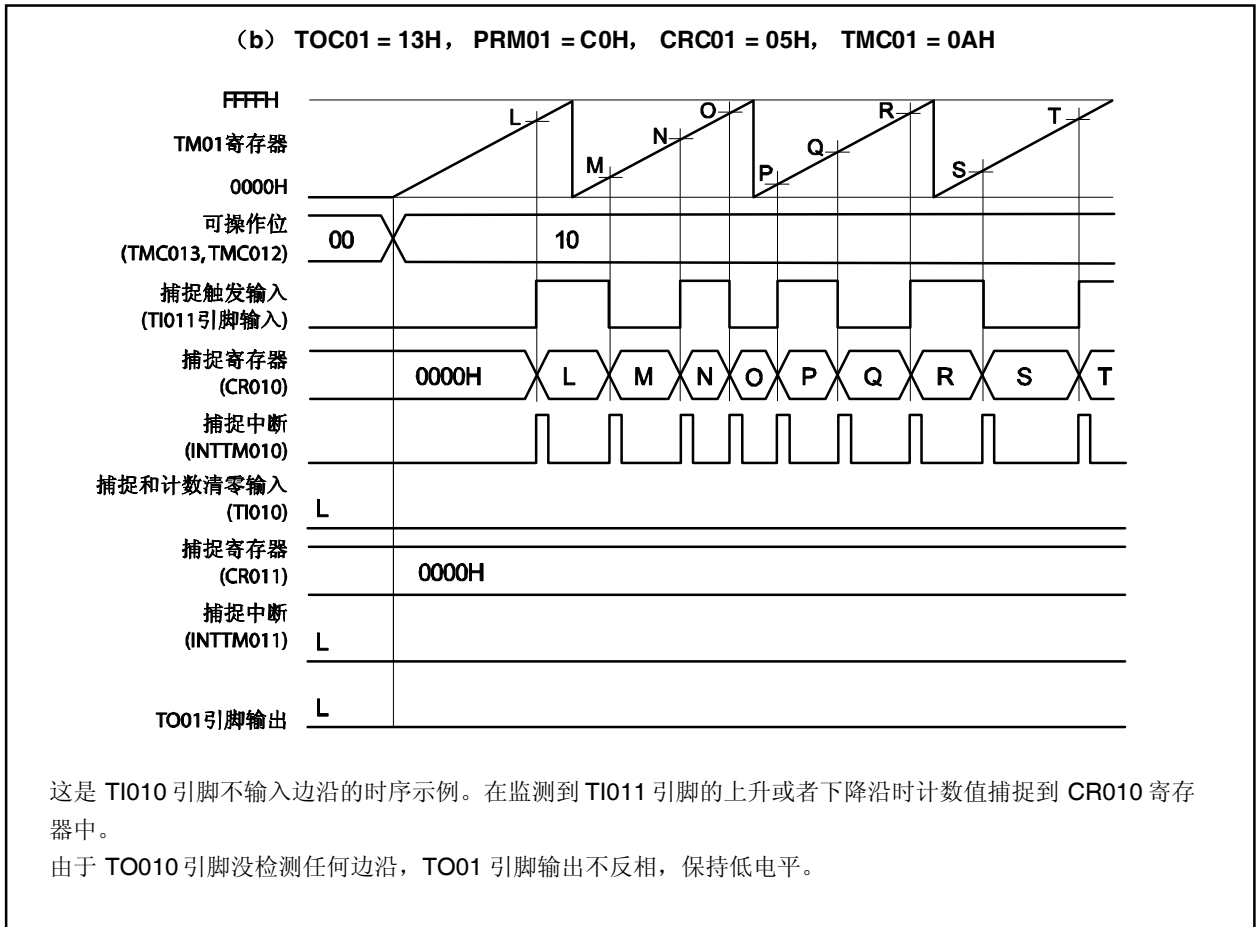
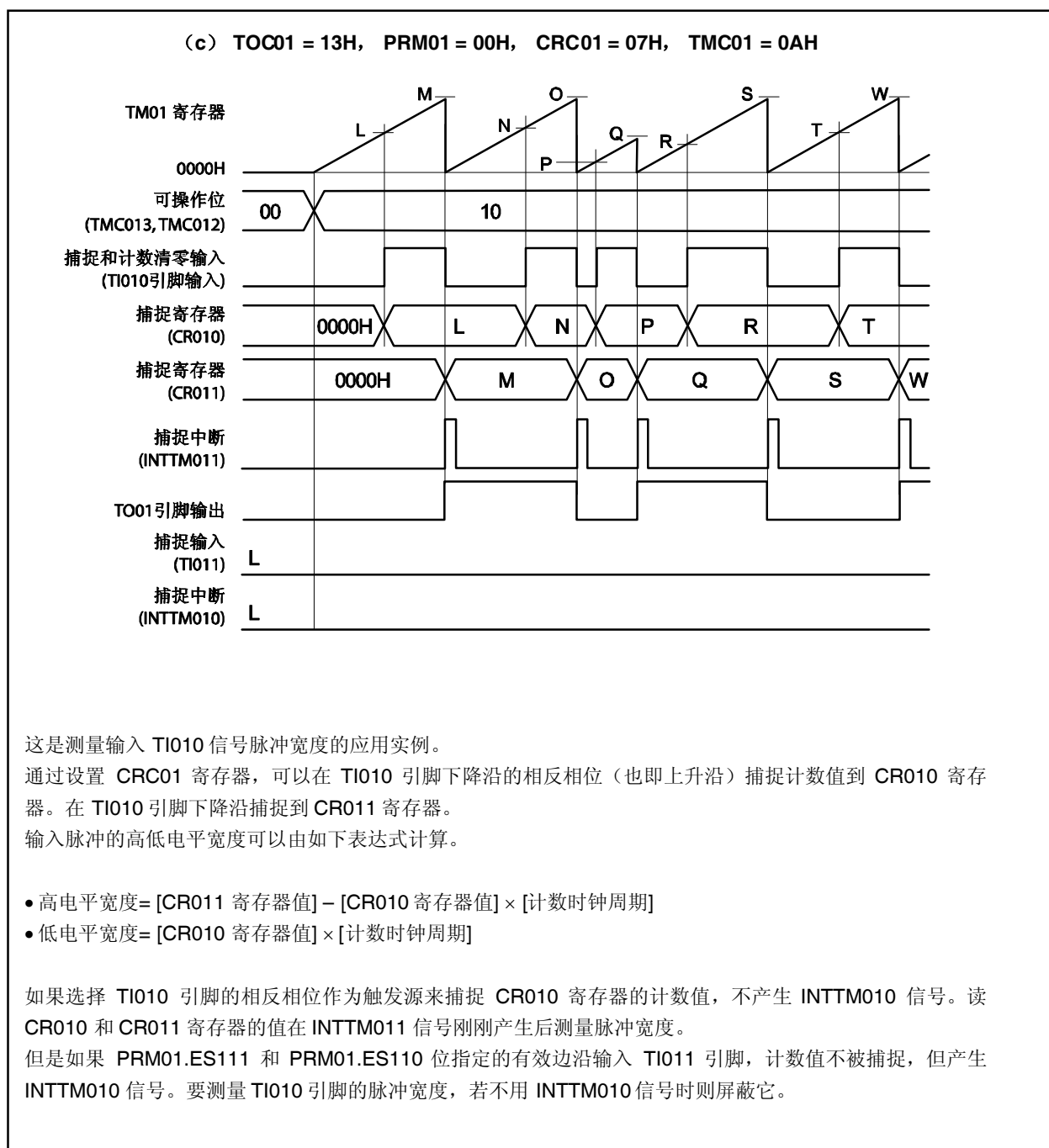


图 7-20. 通过 TI010 引脚输入有效边沿进入的清零和开始模式的时序示例
(CR010 寄存器: 捕捉寄存器, CR011 寄存器: 捕捉寄存器) (3/3)



这是测量输入 TI010 信号脉冲宽度的应用实例。

通过设置 CRC01 寄存器, 可以在 TI010 引脚下降沿的相反相位 (也即上升沿) 捕捉计数值到 CR010 寄存器。在 TI010 引脚下降沿捕捉到 CR011 寄存器。

输入脉冲的高低电平宽度可以由如下表达式计算。

- 高电平宽度 = [CR011 寄存器值] - [CR010 寄存器值] × [计数时钟周期]
- 低电平宽度 = [CR010 寄存器值] × [计数时钟周期]

如果选择 TI010 引脚的相反相位作为触发源来捕捉 CR010 寄存器的计数值, 不产生 INTTM010 信号。读 CR010 和 CR011 寄存器的值在 INTTM011 信号刚刚产生后测量脉冲宽度。

但是如果 PRM01.ES111 和 PRM01.ES110 位指定的有效边沿输入 TI011 引脚, 计数值不被捕捉, 但产生 INTTM010 信号。要测量 TI010 引脚的脉冲宽度, 若不用 INTTM010 信号时则屏蔽它。

图 7-21. 通过 TI010 引脚输入有效边沿进入的清零和开始模式的寄存器设置示例 (1/2)

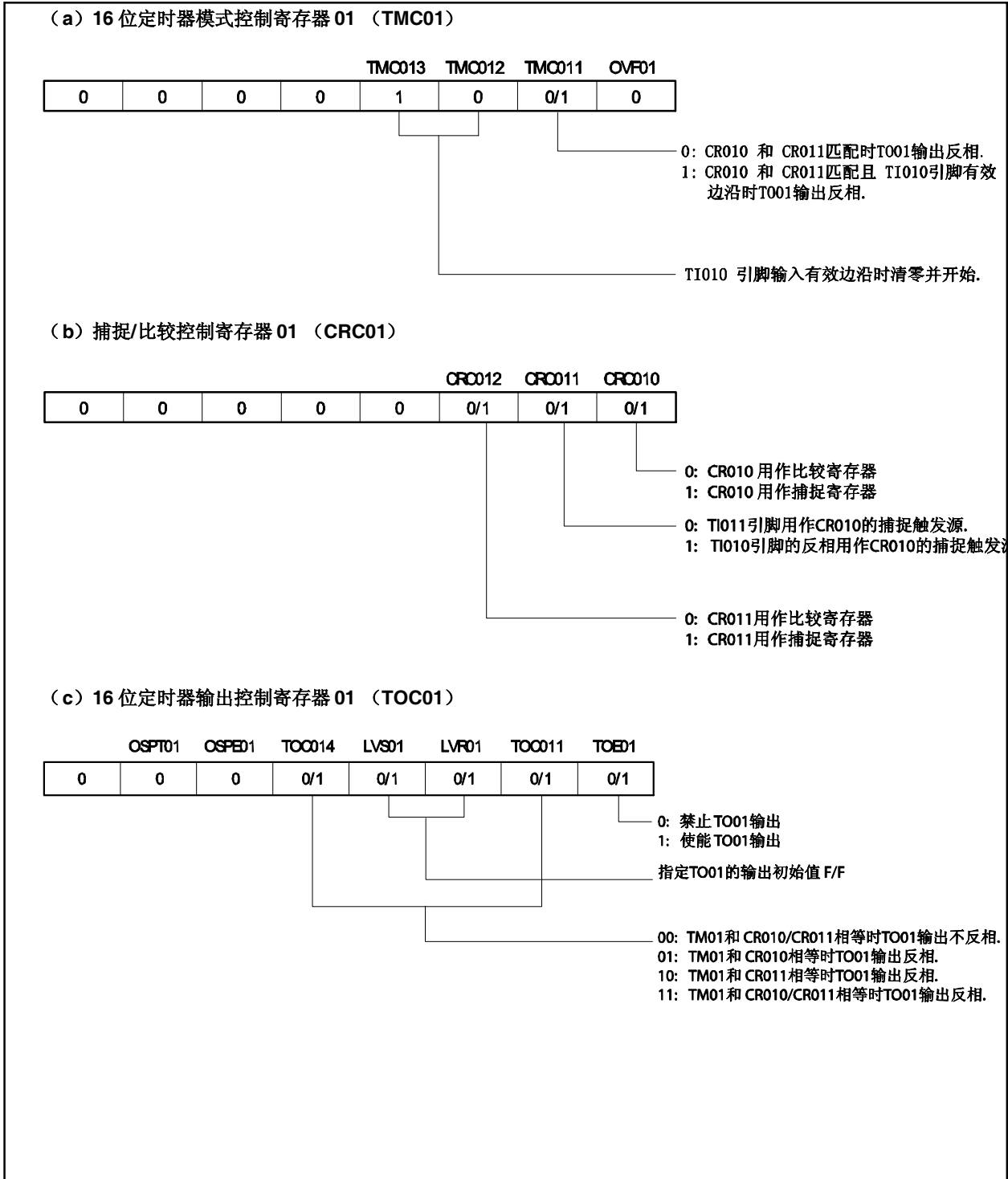


图 7-21. 通过 TI010 引脚输入有效边沿进入的清零和开始模式的寄存器设置示例 (2/2)

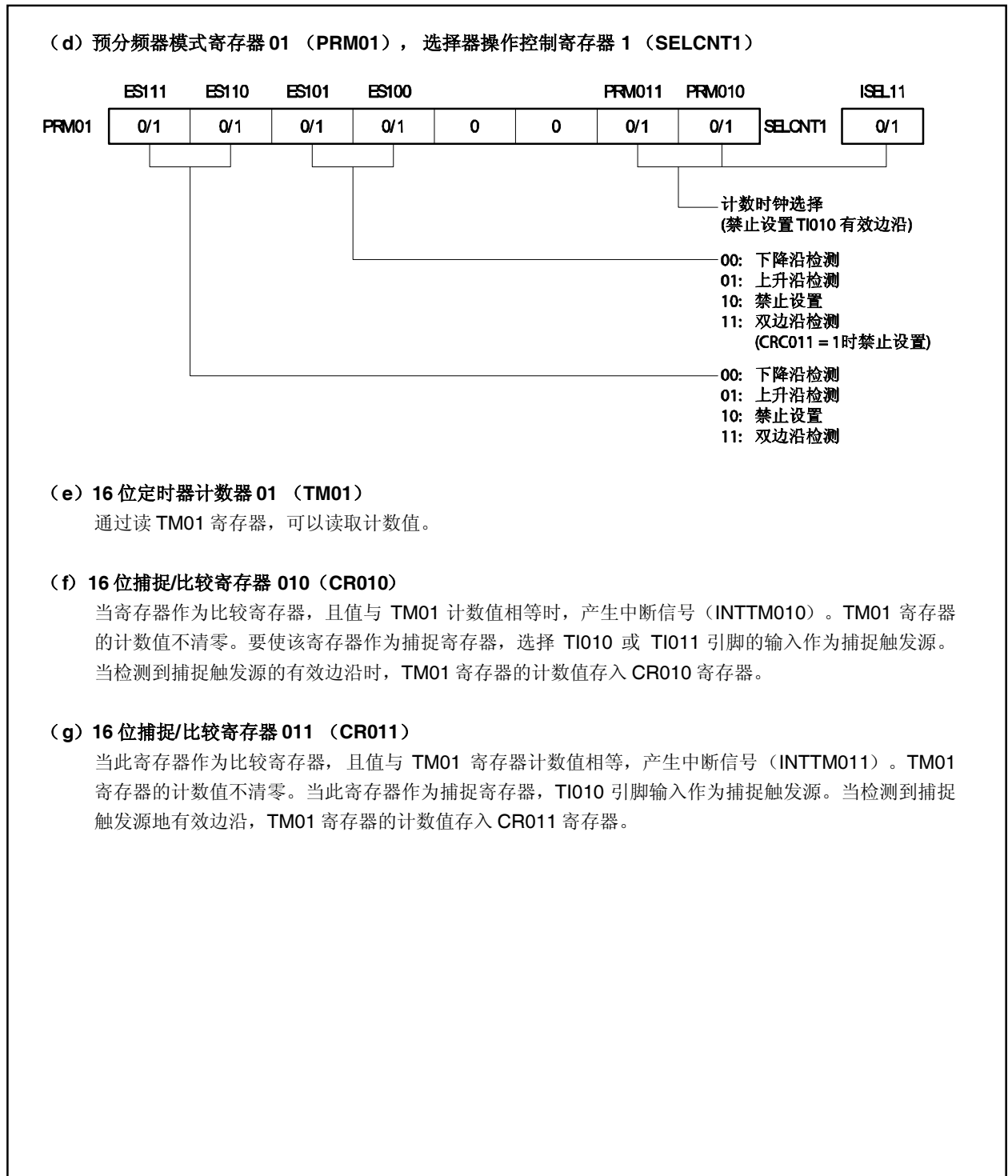
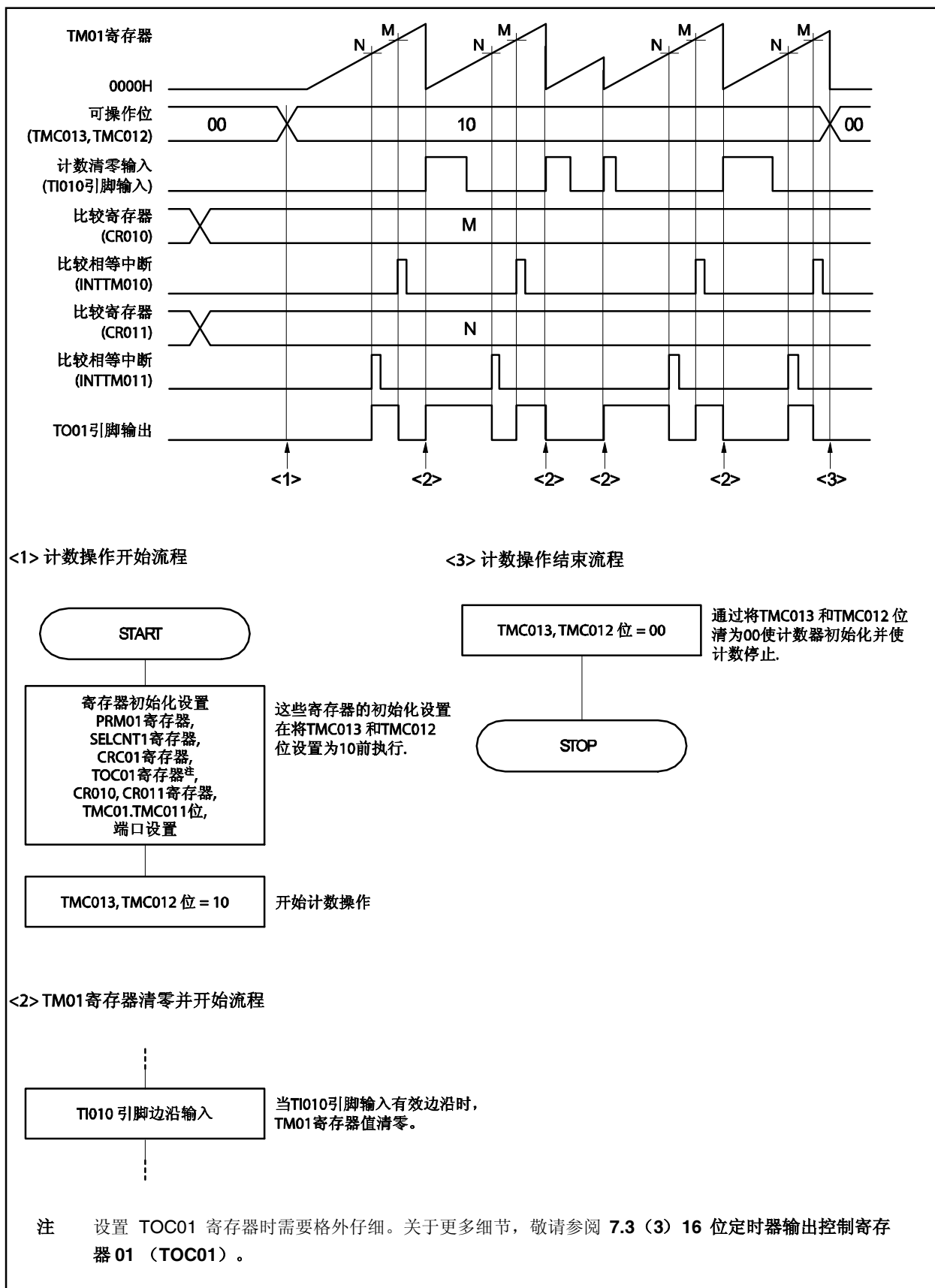


图 7-22. 通过 TI010 引脚输入有效边沿进入的清零和开始模式的软件处理示例



7.4.5 自由运行定时器操作

当 TMC01.TMC013 和 TMC01.TMC012 位设置为 01（自由运行定时器模式），16 位定时器/事件计数器 01 与计数时钟同步继续向上计数。当向上计数至 FFFFH 时，溢出标志（TMC01.OVF01 位）在下一个时钟设置为 1，TM01 寄存器清零（为 0000H）且继续计数。通过软件执行 CLR 指令可以清零 OVF01 位。

可以实现如下三种的自由运行定时器操作类型。

- CR010 和 CR011 寄存器均作为比较寄存器。
- CR010 和 CR011 寄存器之一作为比较寄存器，另外一个作为捕捉寄存器。
- CR010 和 CR011 寄存器均作为捕捉寄存器。

备注

1. 关于复用功能引脚（TO01）的设置，参见表 4-12 端口引脚使用复用功能时的设置。
2. 关于 INTTM010 和 INTTM011 中断的使能，参见第十七章 中断/异常处理功能。

(1) 自由运行定时器模式操作

（CR010 寄存器：比较寄存器， CR011 寄存器：比较寄存器）

图 7-23. 自由运行定时器模式框图
（CR010 寄存器：比较寄存器， CR011 寄存器：比较寄存器）

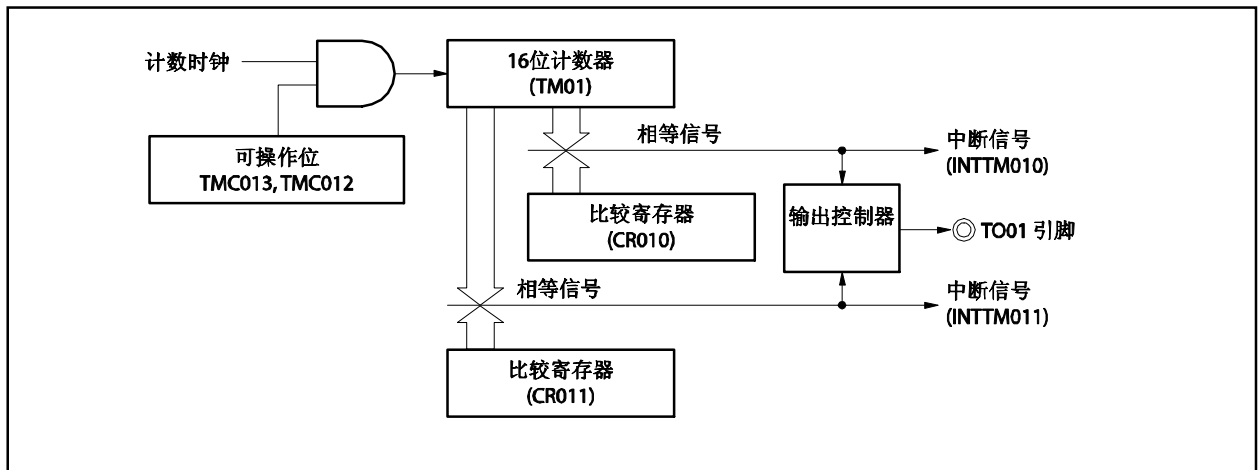
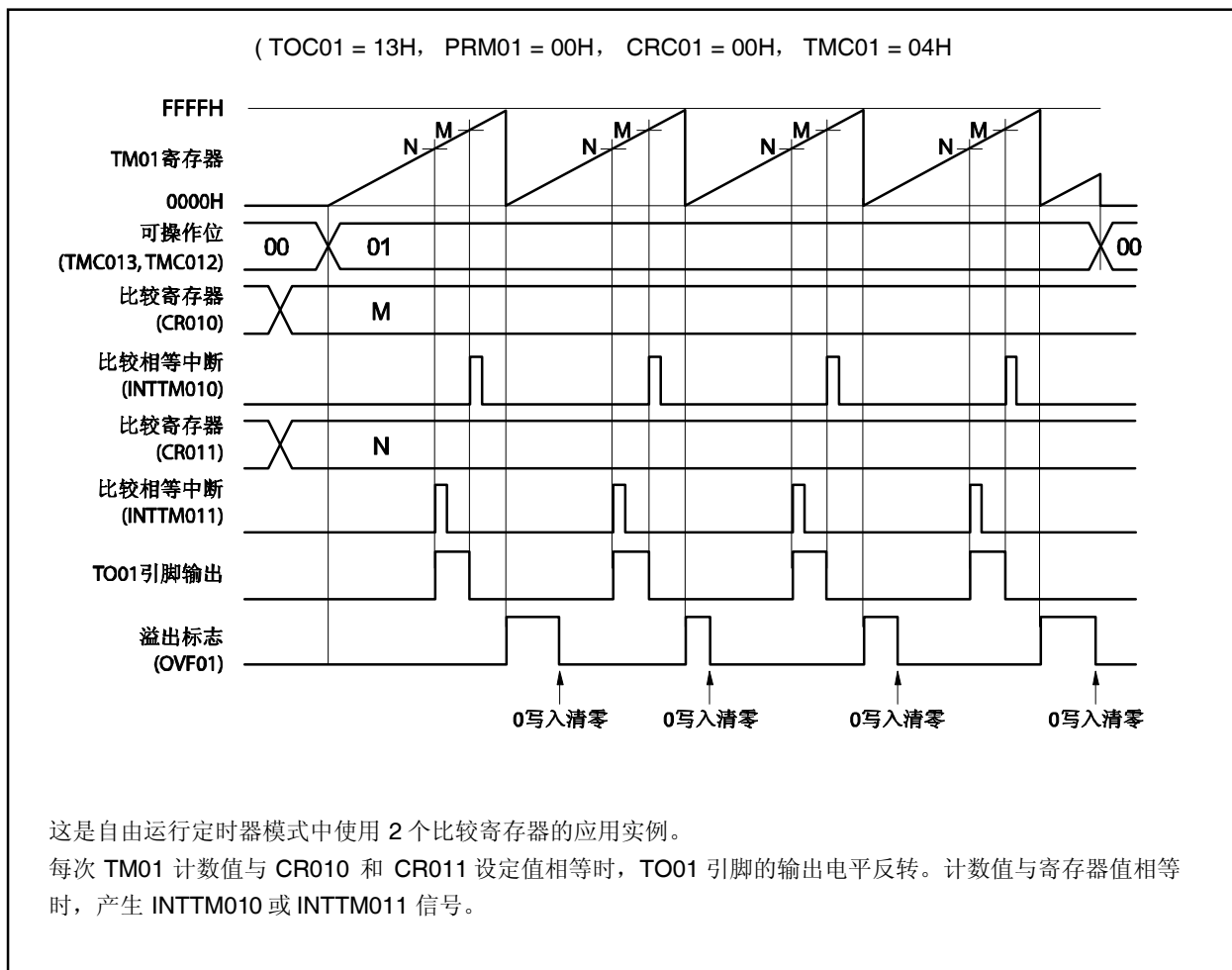


图 7-24. 自由运行定时器模式时序示例
(CR010 寄存器: 比较寄存器, CR011 寄存器: 比较寄存器)



(2) 自由运行定时器模式操作

(CR010 寄存器: 比较寄存器, CR011 寄存器: 捕捉寄存器)

图 7-25. 自由运行定时器模式框图
(CR010 寄存器: 比较寄存器, CR011 寄存器: 捕捉寄存器)

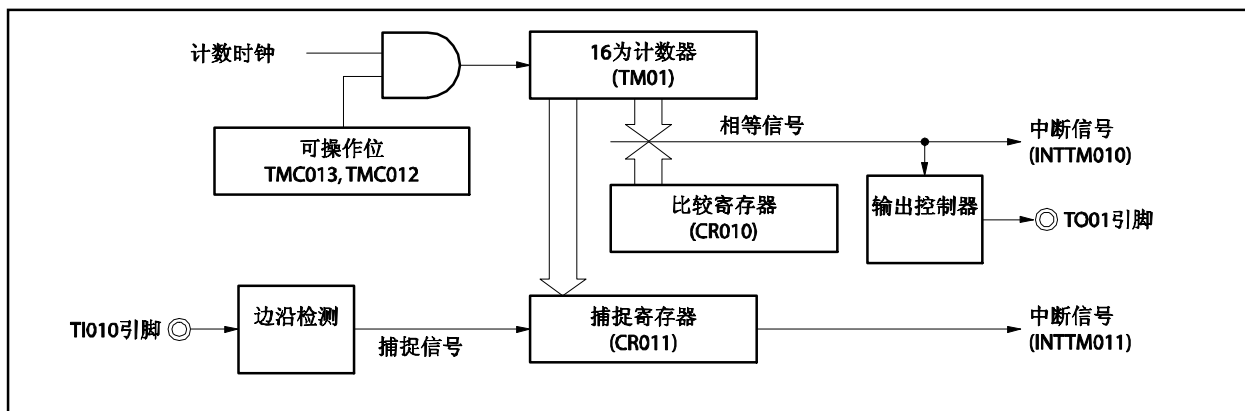
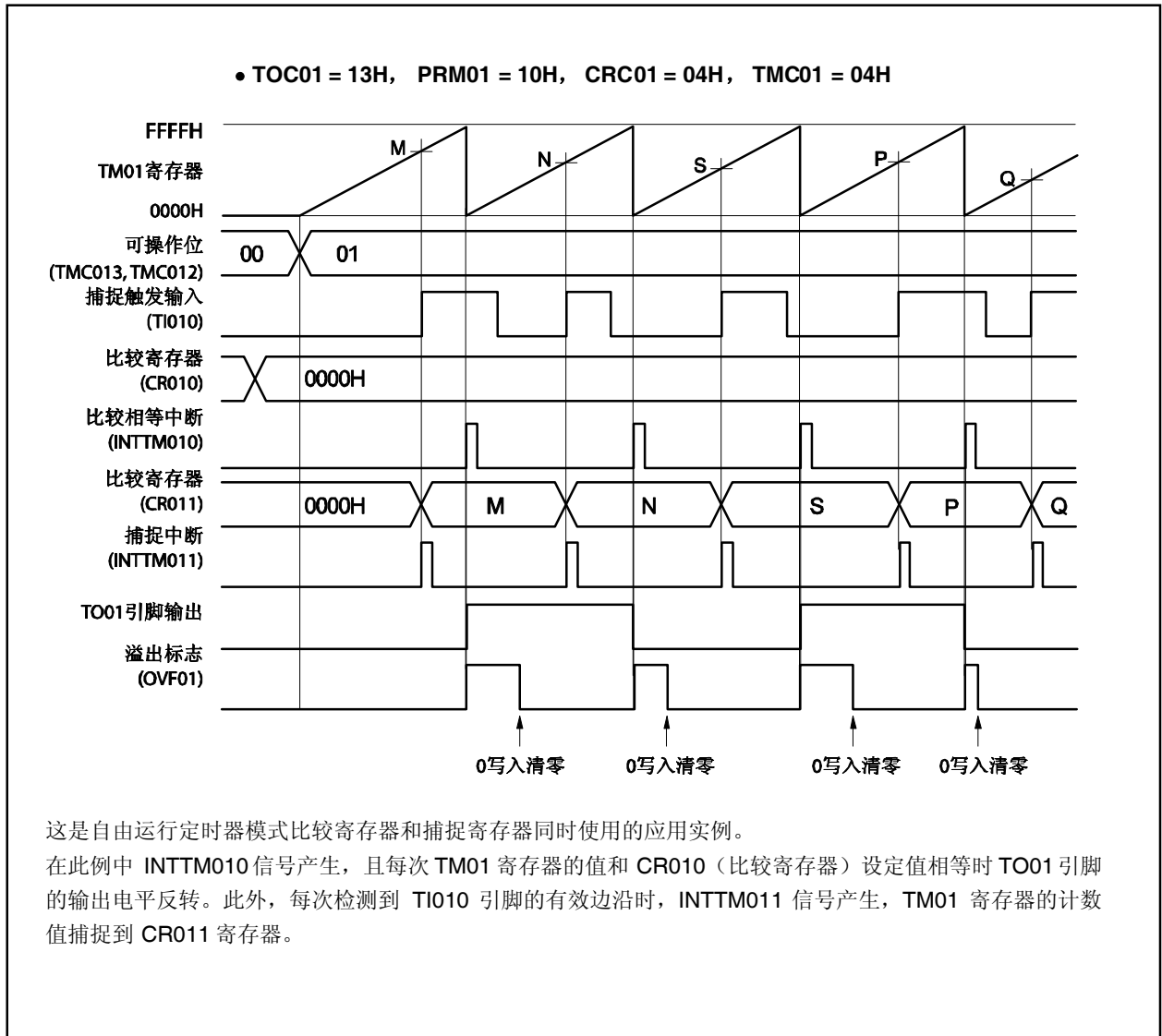


图 7-26. 自由运行定时器模式时序示例
 (CR010 寄存器: 比较寄存器, CR011 寄存器: 捕捉寄存器)



(3) 自由运行定时器模式操作

(CR010 寄存器: 捕捉寄存器, CR011 寄存器: 捕捉寄存器)

图 7-27. 自由运行定时器模式框图
(CR010 寄存器: 捕捉寄存器, CR011 寄存器: 捕捉寄存器)

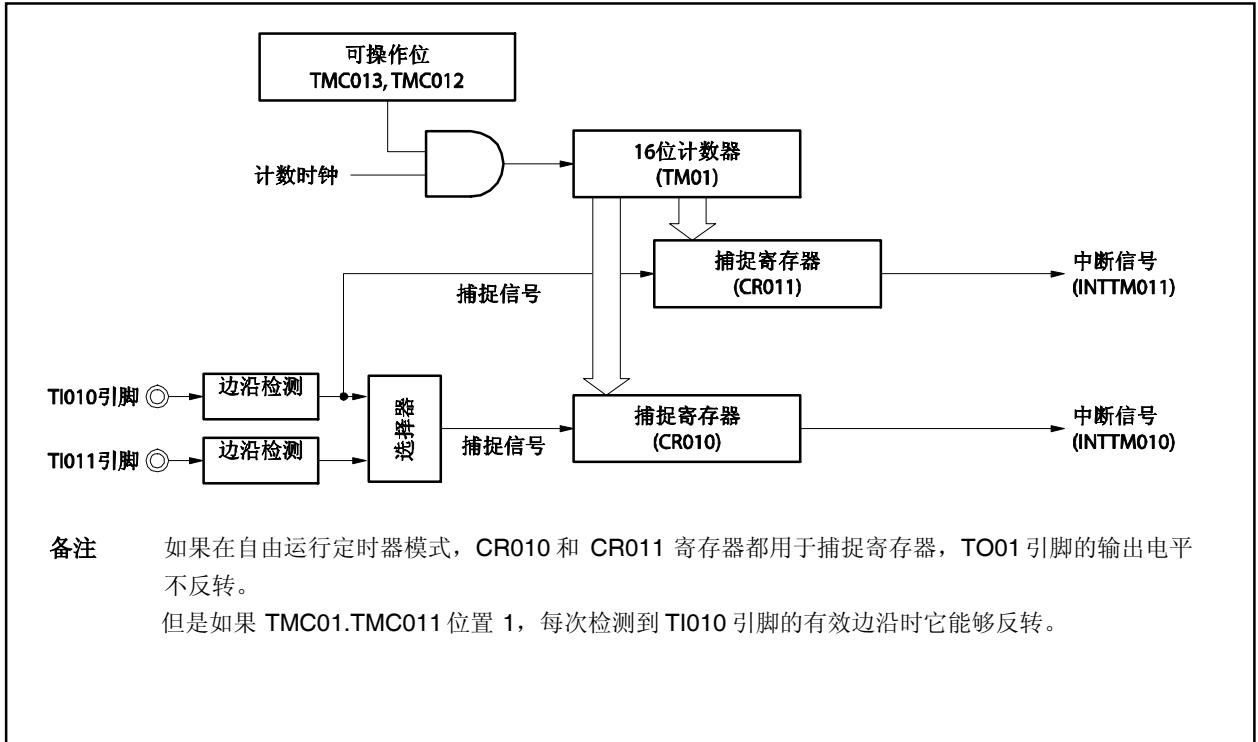


图 7-28. 自由运行定时器模式时序示例
 (CR010 寄存器: 捕捉寄存器, CR011 寄存器: 捕捉寄存器) (1/2)

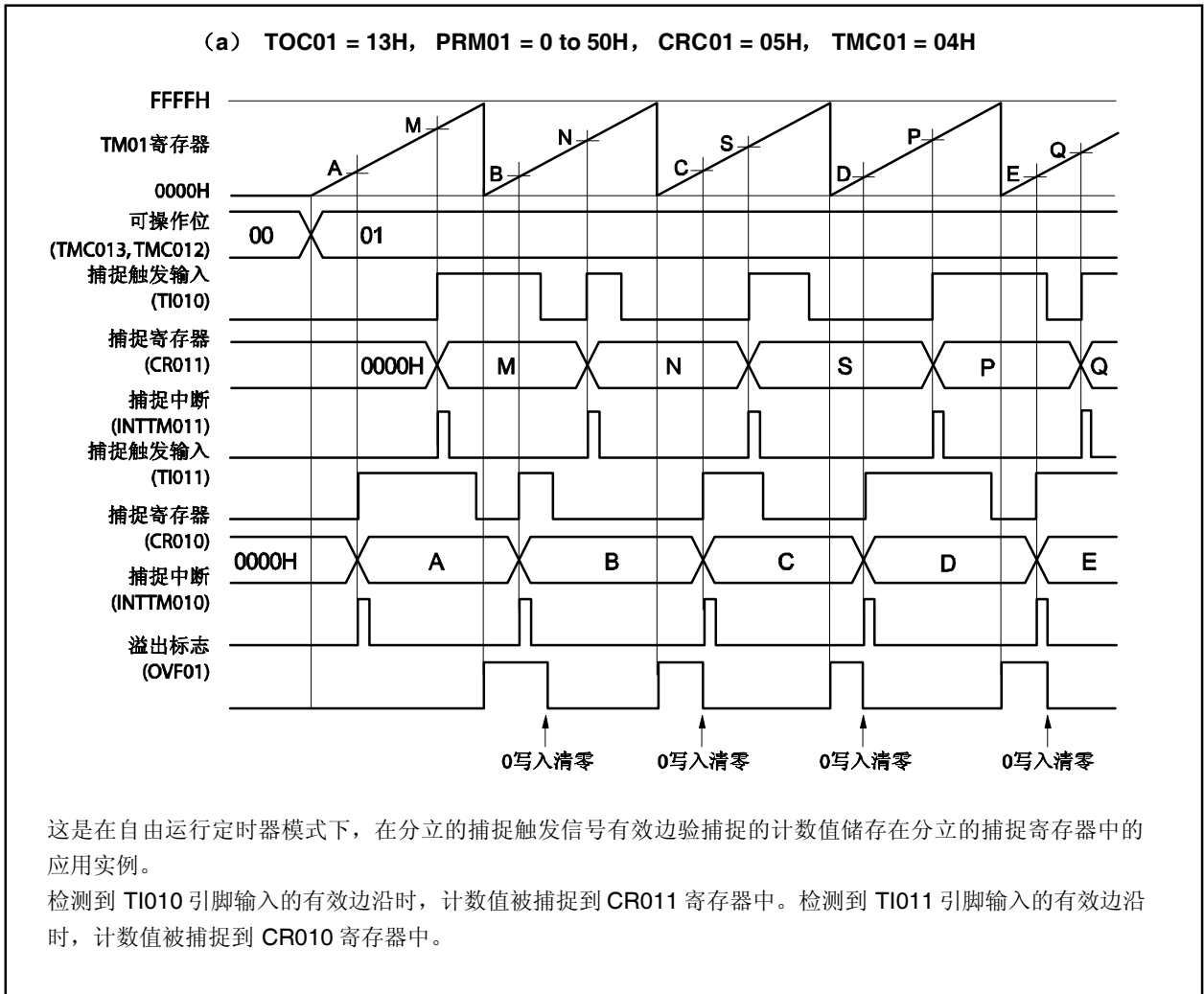


图 7-28. 自由运行定时器模式时序示例
 (CR010 寄存器: 捕捉寄存器, CR011 寄存器: 捕捉寄存器) (2/2)

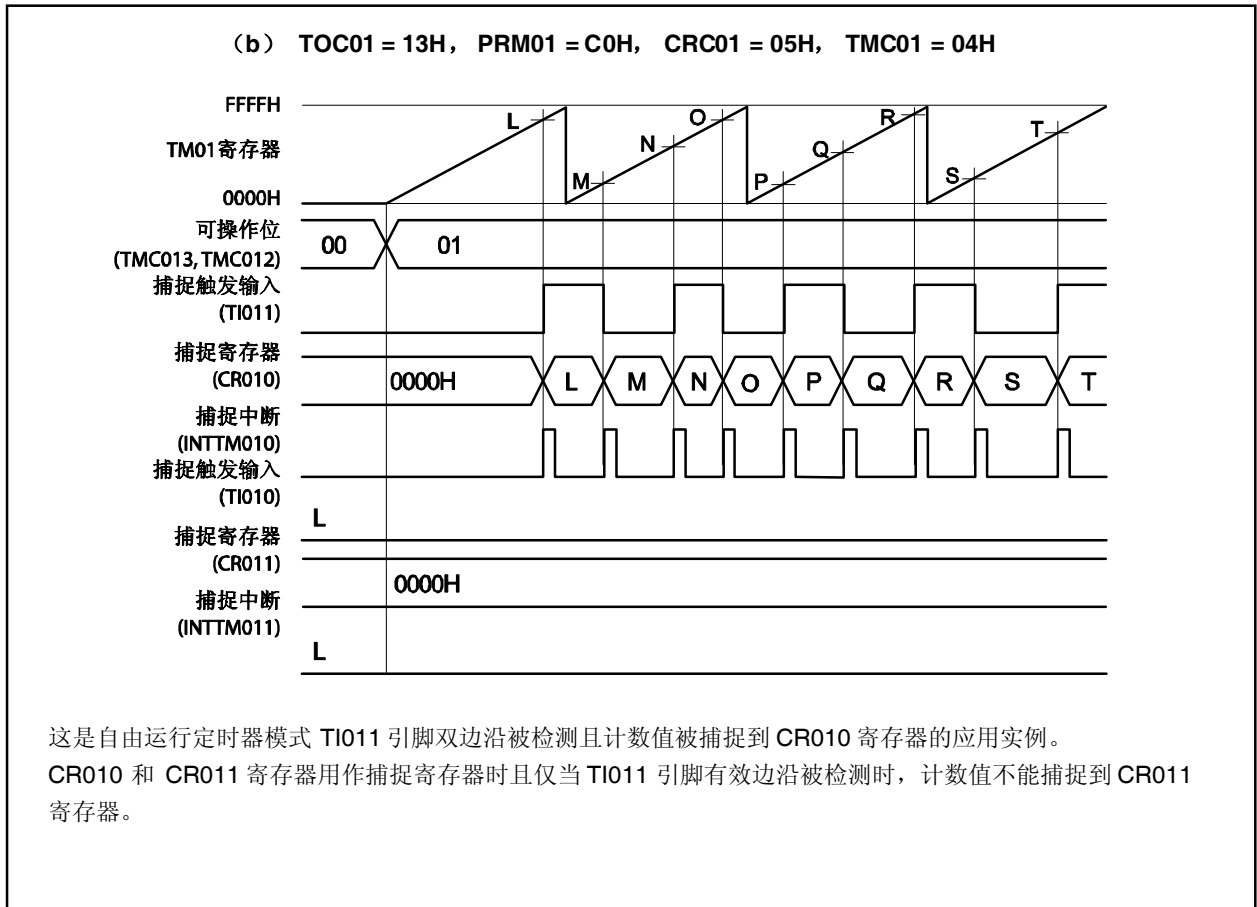


图 7-29. 自由运行定时器模式寄存器设置示例 (1/2)

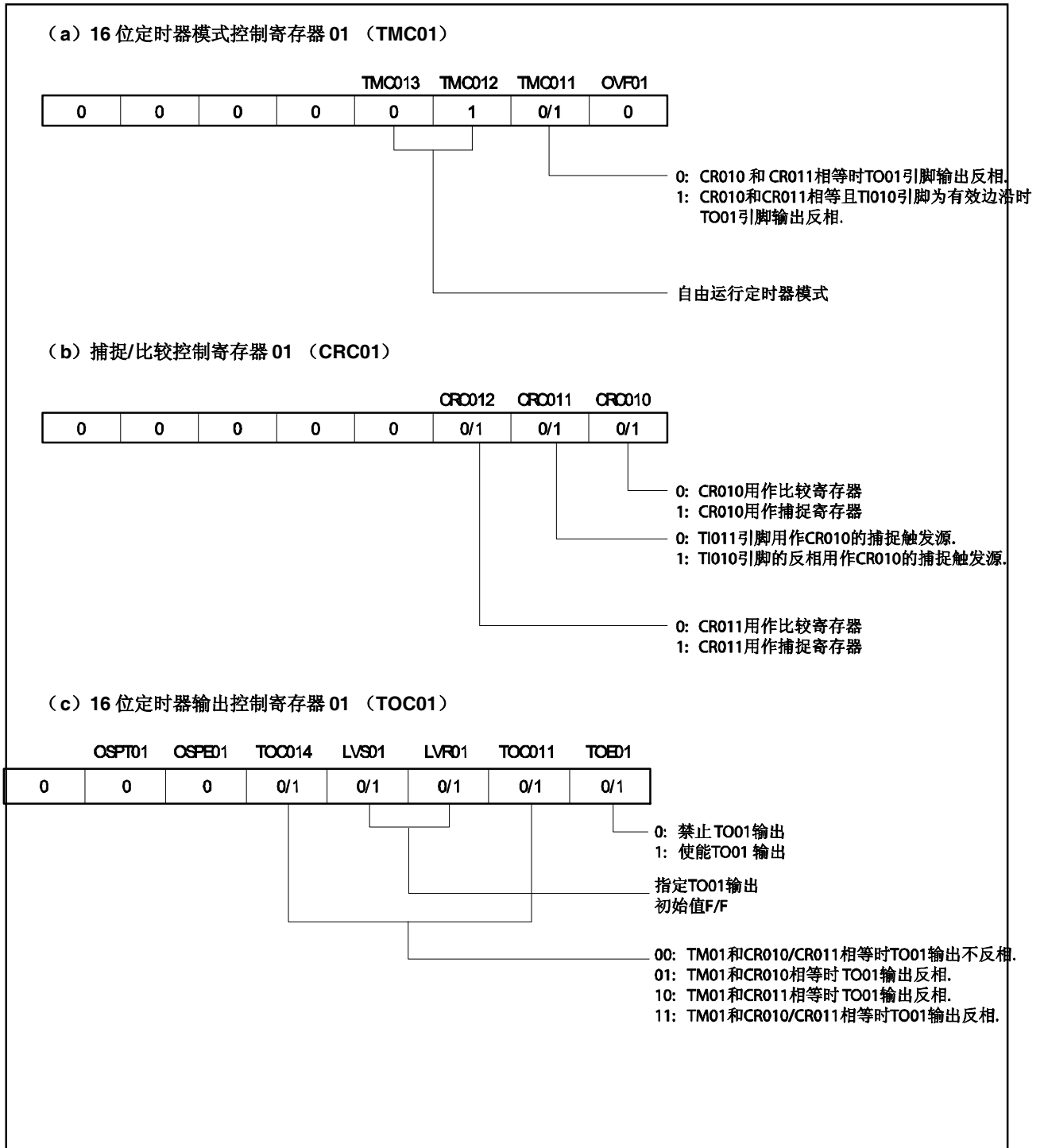


图 7-29. 自由运行定时器模式寄存器设置示例 (2/2)

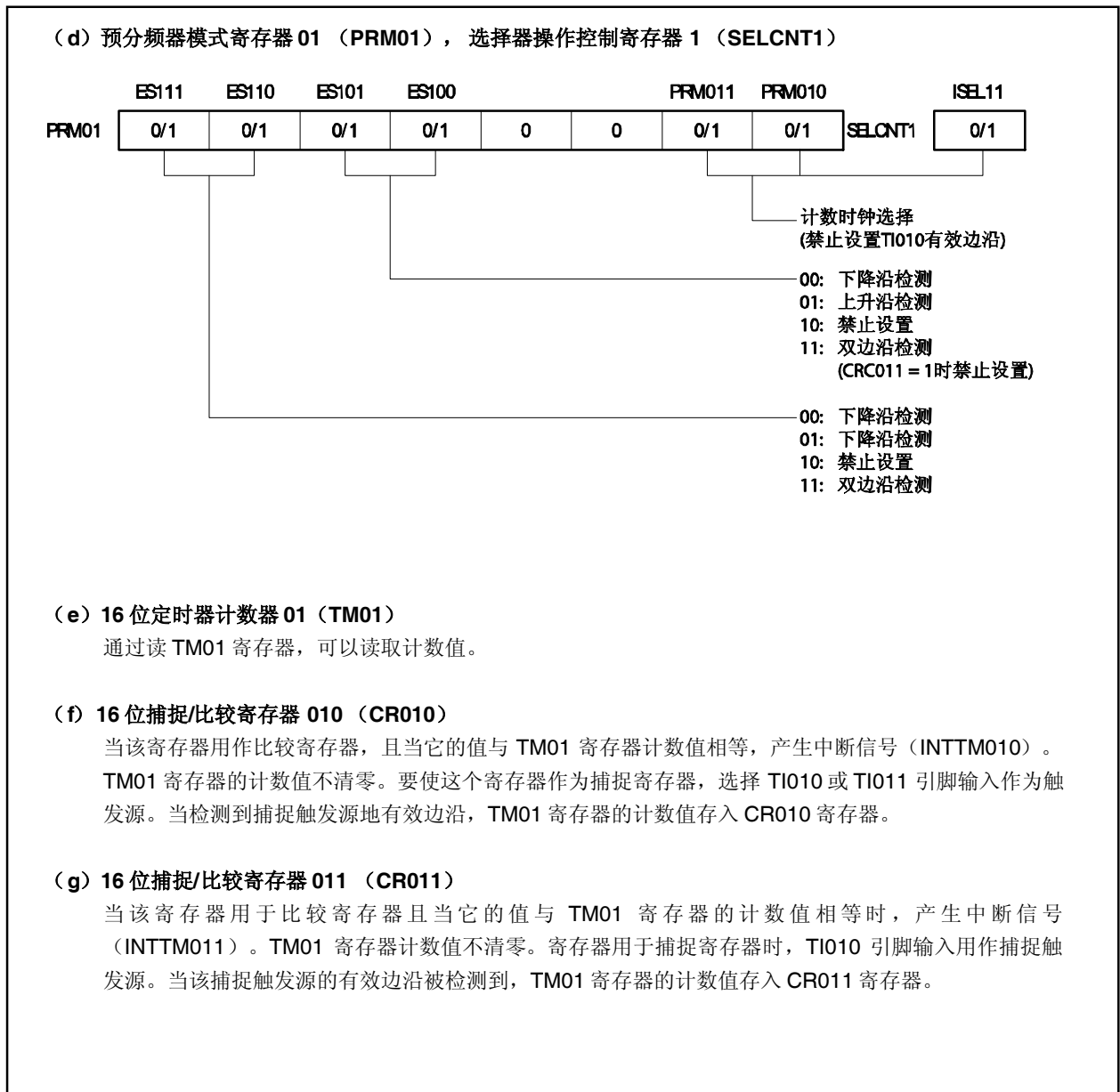
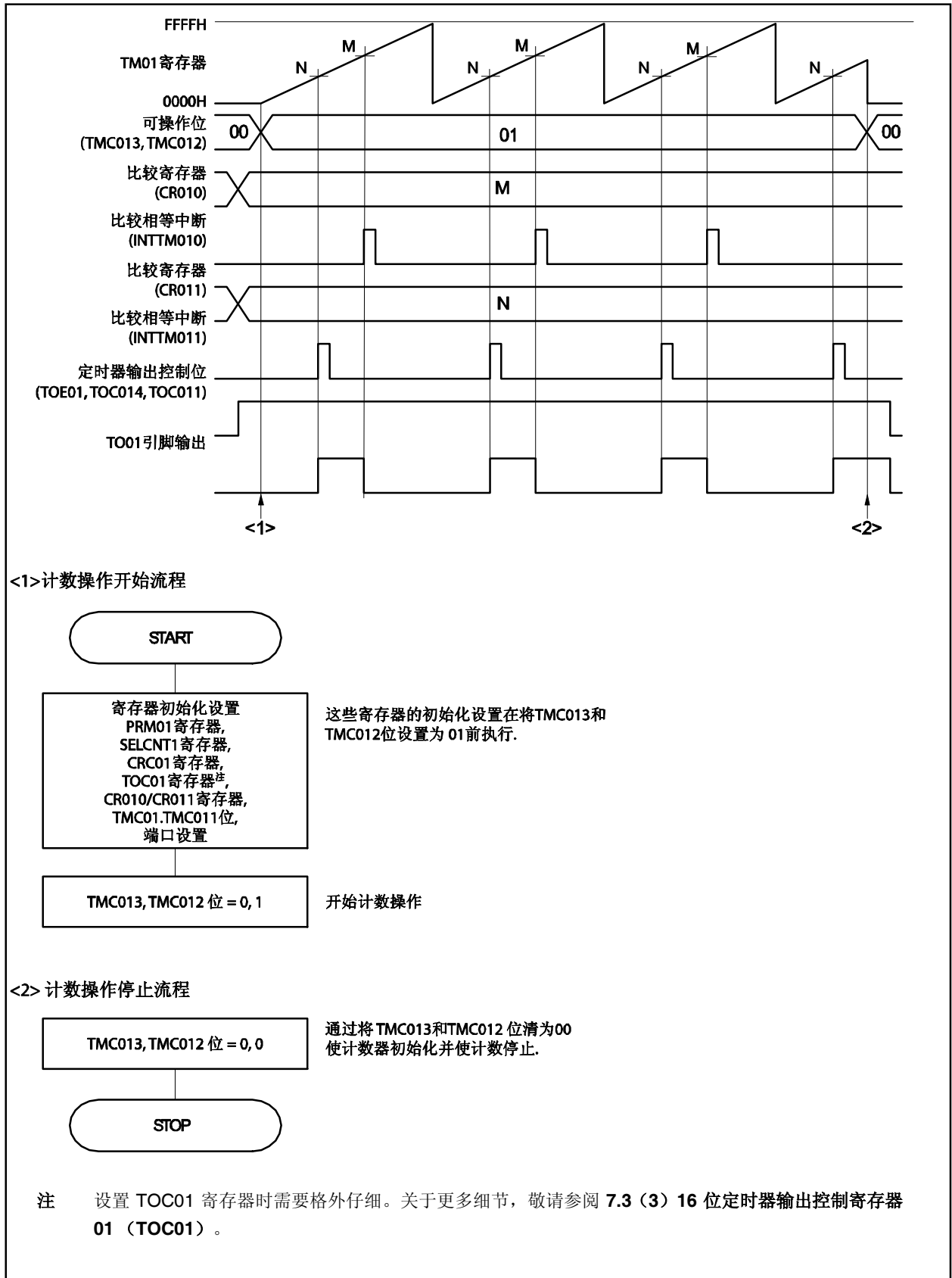


图 7-30. 自由运行定时器模式软件处理示例



7.4.6 PPG 输出操作

当 TMC01.TMC013 和 TMC01.TMC012 位设置为 11 时（TM01 寄存器和 CR010 寄存器相等时清零并开始），TO01 引脚可以输出一个由 CR011 寄存器预先设置脉宽的矩形波，称为 PPG（可编程脉冲发生器），周期由 CR010 寄存其设置。

PPG 输出的脉冲周期和占空比因子如下。

- 脉冲周期 = (CR010 寄存器设置值 + 1) × 计数时钟周期
- 占空比 = (CR011 寄存器设置值 + 1) / (CR010 寄存器设置值 + 1)

注意事项 要在操作期间改变占空比因子（CR011 寄存器值），参见 7.5.1 TM01 操作期间改写 CR011 寄存器。

- 备注**
1. 关于复用功能引脚设置，参见表 4-12 端口引脚使用复用功能时的设置。
 2. 关于 INTTM010 和 INTTM011 中断的使能，参见第十七章 中断/异常处理功能。

图 7-31. PPG 输出操作框图

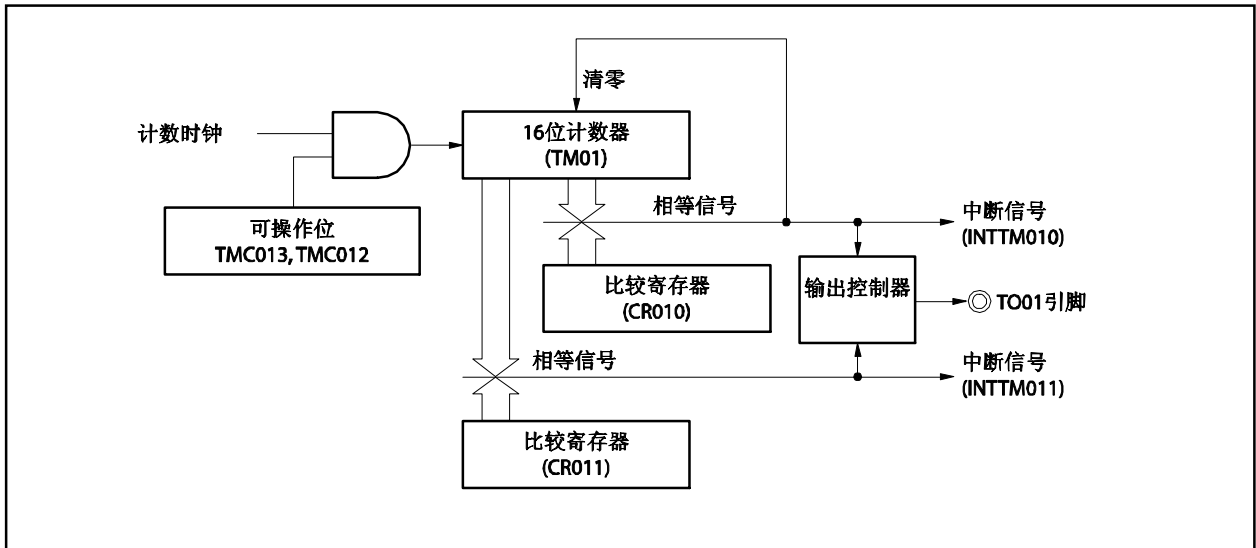


图 7-32. PPG 输出操作的寄存器设置示例

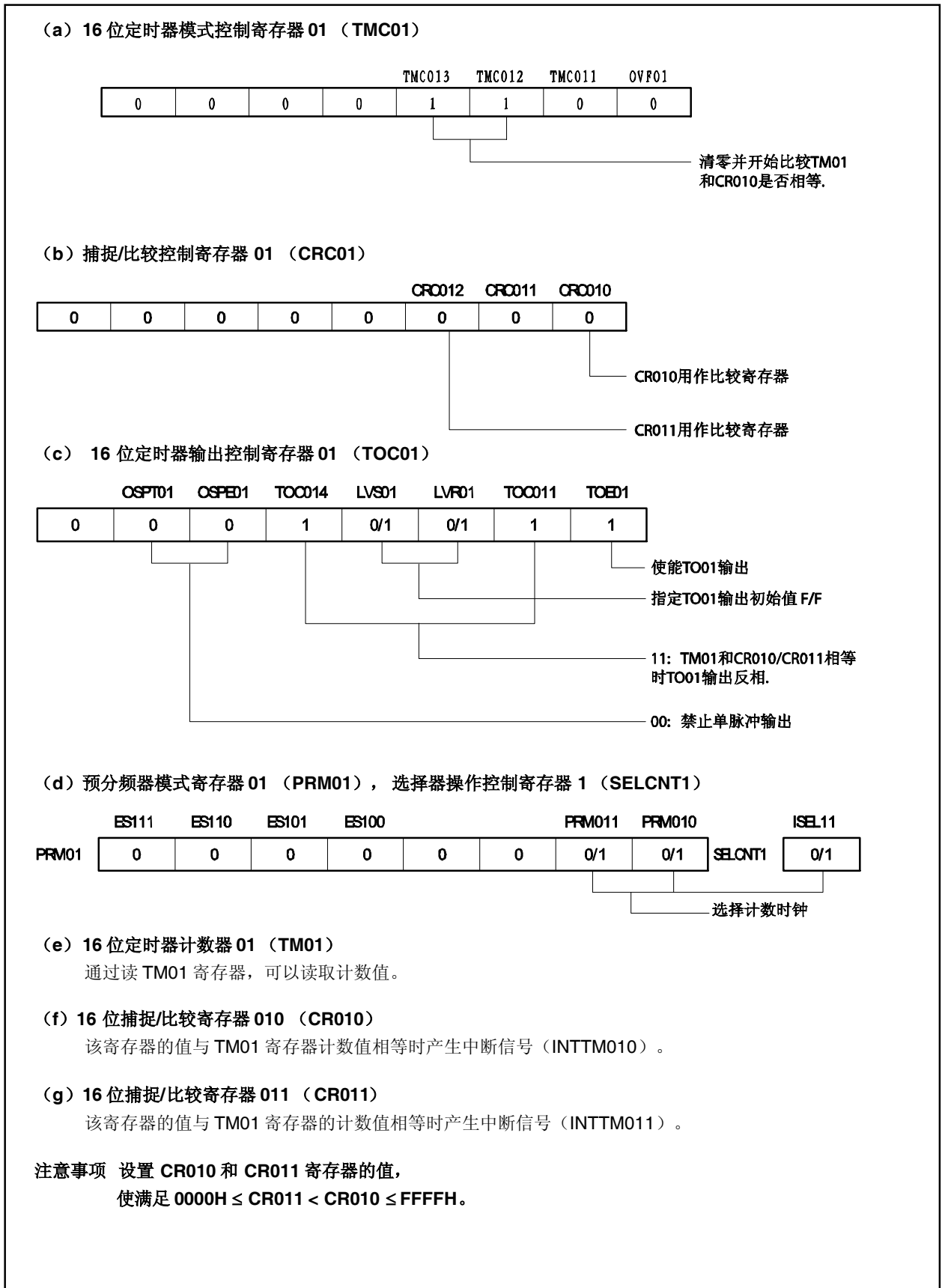
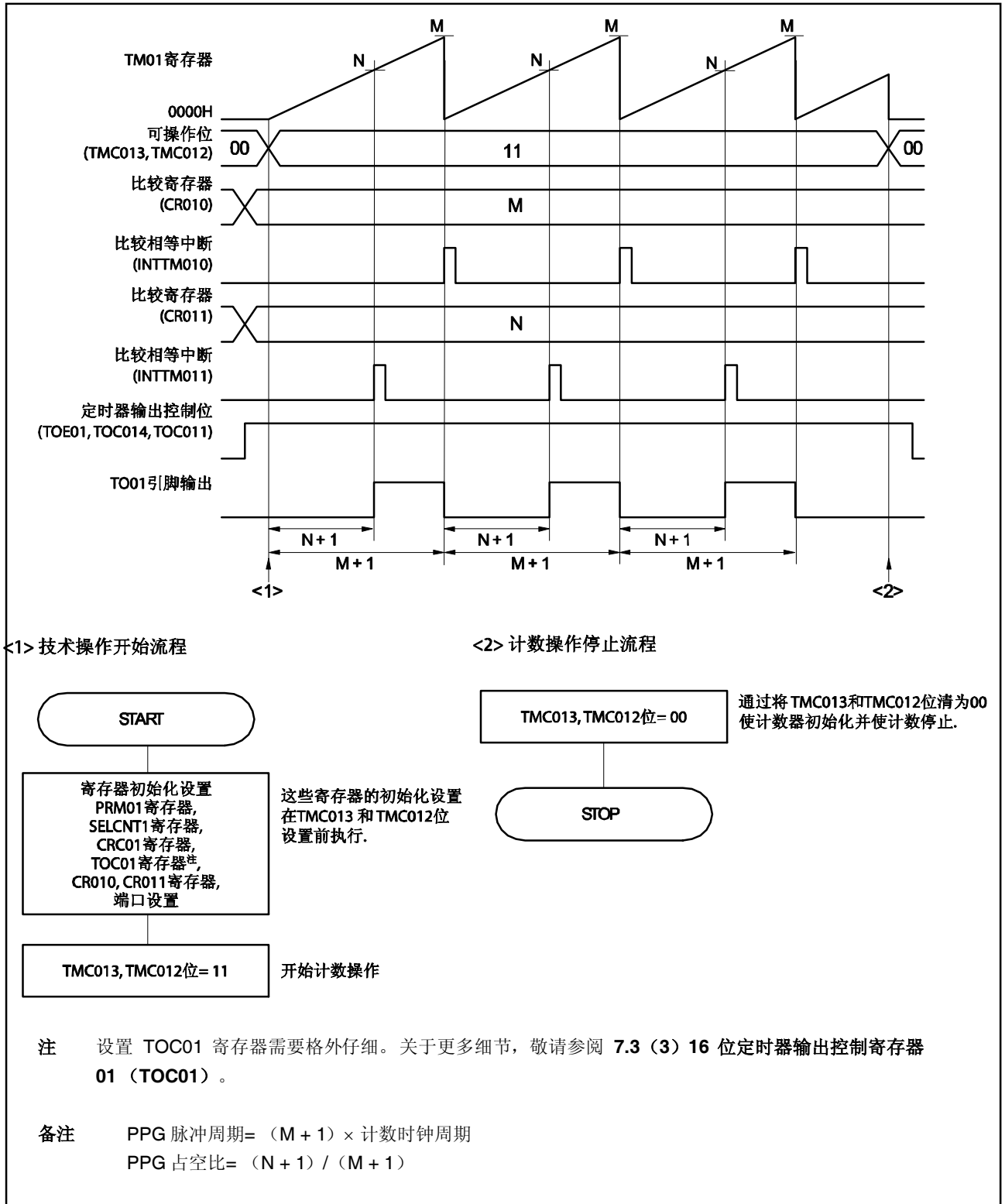


图 7-33. PPG 输出操作的软件处理示例



7.4.7 单脉冲输出操作

设置 TMC01.TMC013 和 TMC01.TMC012 位为 01（自由运行定时器模式）或 10（通过 TI010 引脚有效边沿进入清零和开始模式），并且设置 TOC01.OSPE01 位为 1 时，可以输出单脉冲。

当 TOC01.OSPT01 设置为 1，或者当定时器操作过程中 TI010 引脚输入有效边沿 TM01 寄存器的清零和开始被触发，TO01 引脚输出 CR010 和 CR011 寄存器差值的脉冲，且仅输出一次。

注意事项 在单脉冲正在输出时，不要再一次输入触发（设置 OSPT01 为 1 或检测 TI010 引脚的有效边沿）。要再次输出单脉冲，需要在当前单脉冲输出完成后再产生新的触发。

- 备注**
1. 关于复用功能引脚设置 参见表 4-12 端口引脚使用复用功能时的设置。
 2. 关于 INTTM010 和 INTTM011 中断的使能，参见第十七章 中断/异常处理功能。

图 7-34. 单脉冲输出操作框图

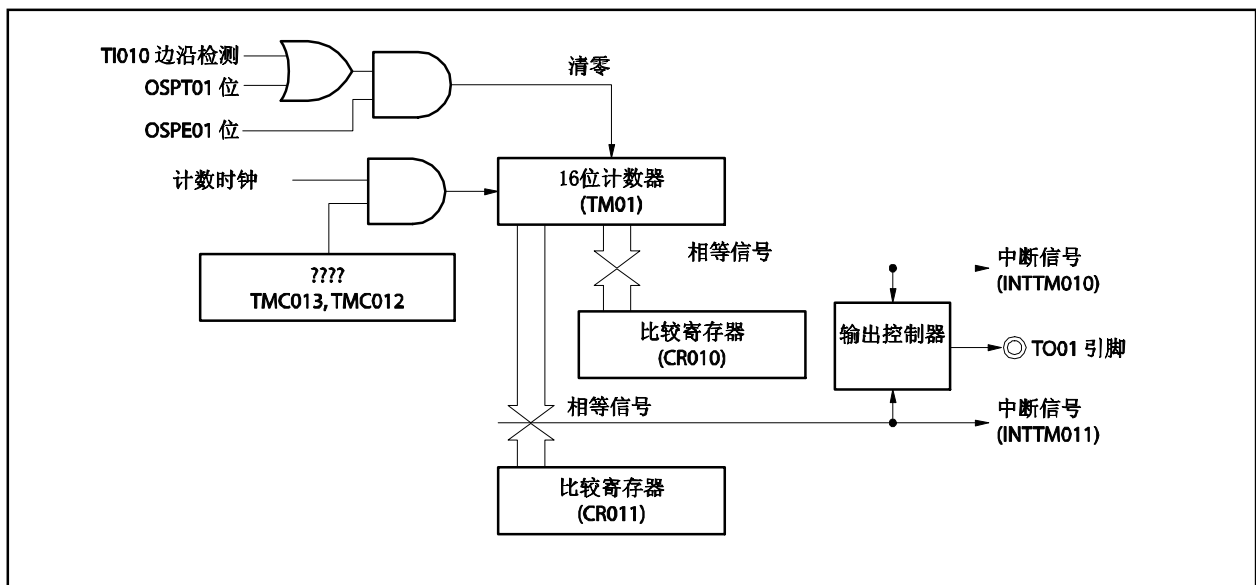


图 7-35. 单脉冲输出操作的寄存器设置示例 (1/2)

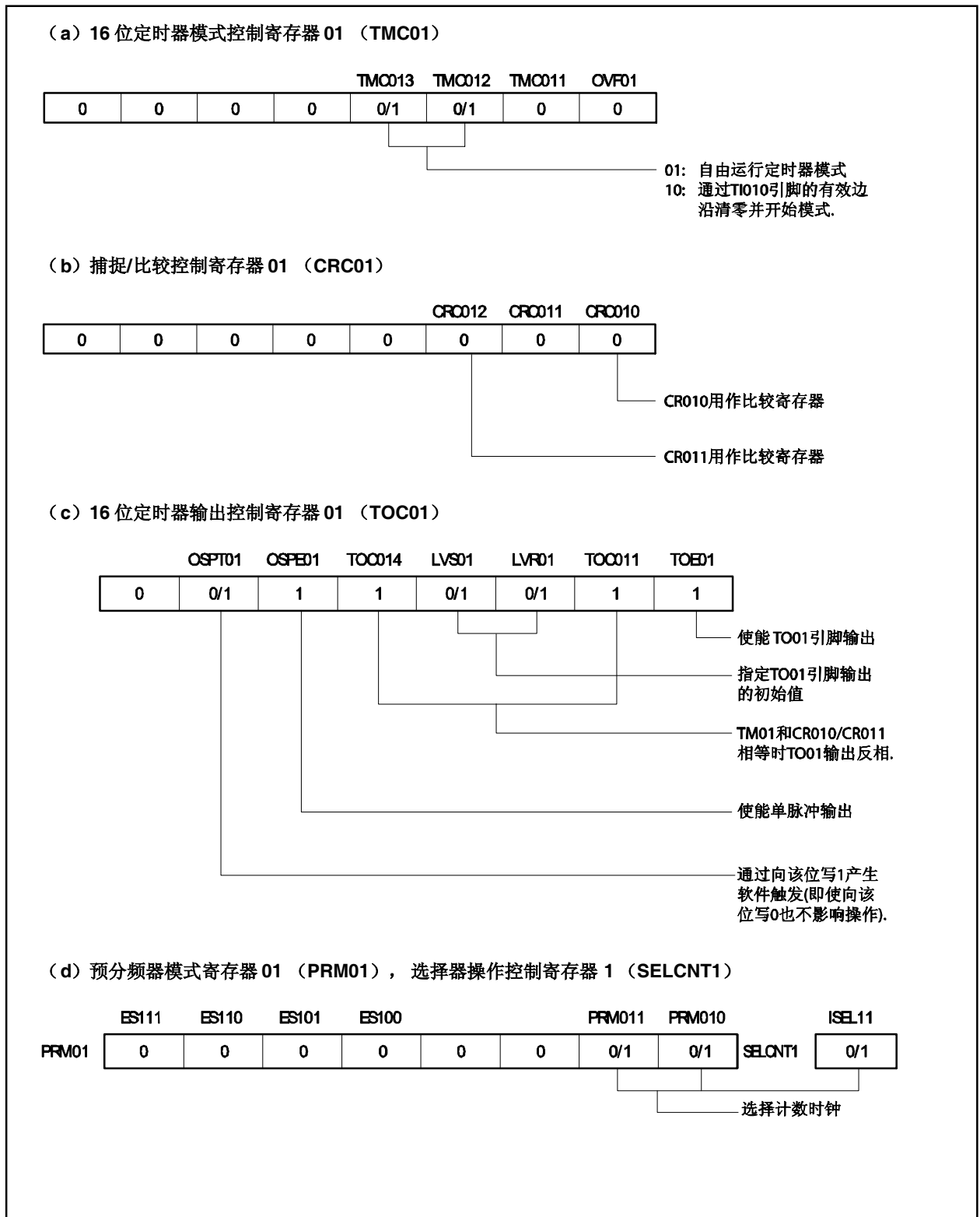


图 7-35. 单脉冲输出操作的寄存器设置示例 (2/2)

(e) 16 位定时器计数器 01 (TM01)

通过读 TM01 寄存器，计数值可被读取。

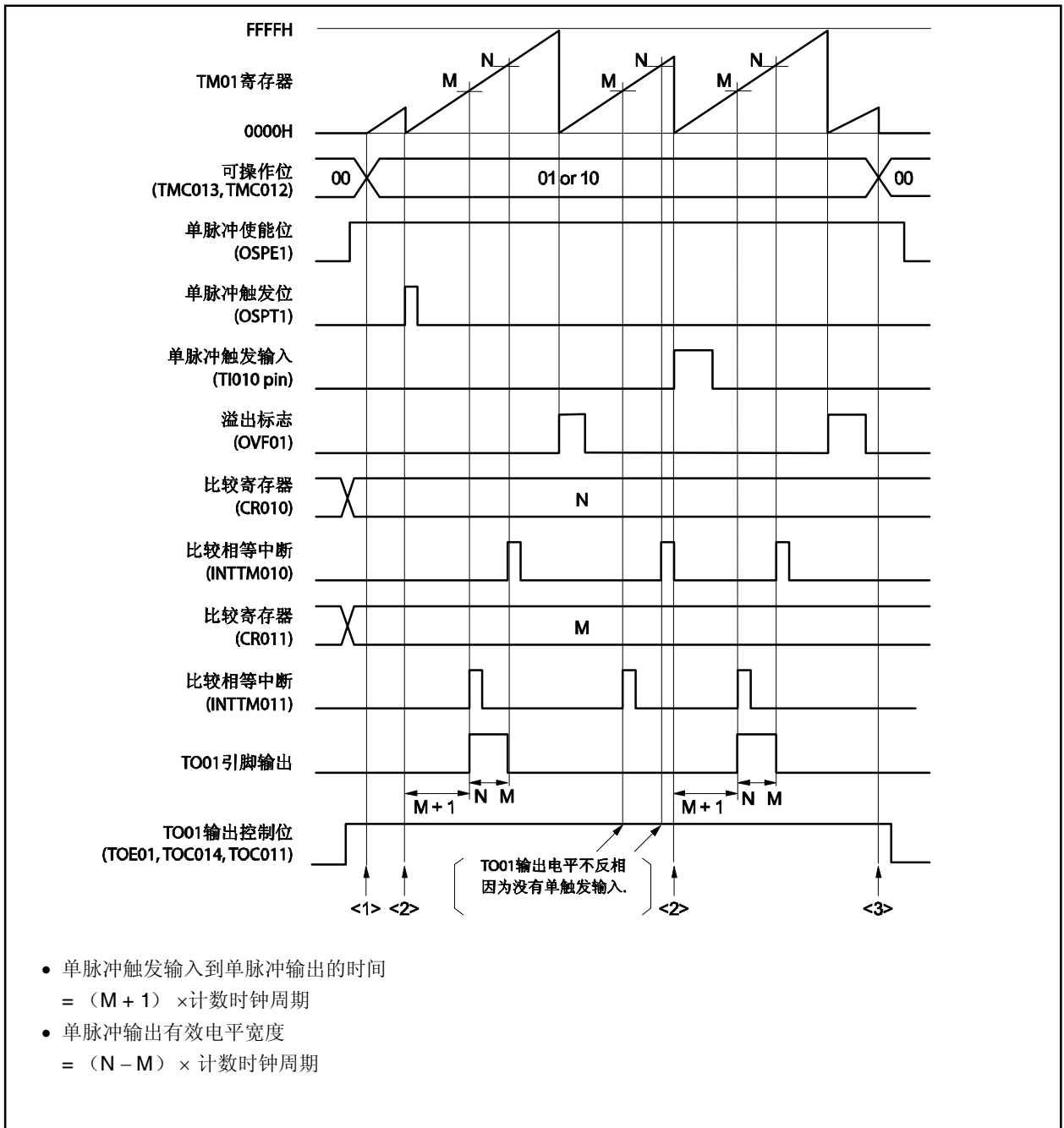
(f) 16 位捕捉/比较寄存器 010 (CR010)

输出单脉冲时，该寄存器可以用于比较寄存器。当 TM01 值与 CR010 寄存器的指相等时，产生中断信号 (INTTM010)，TO01 引脚输出电平反转。

(g) 16 位捕捉/比较寄存器 011 (CR011)

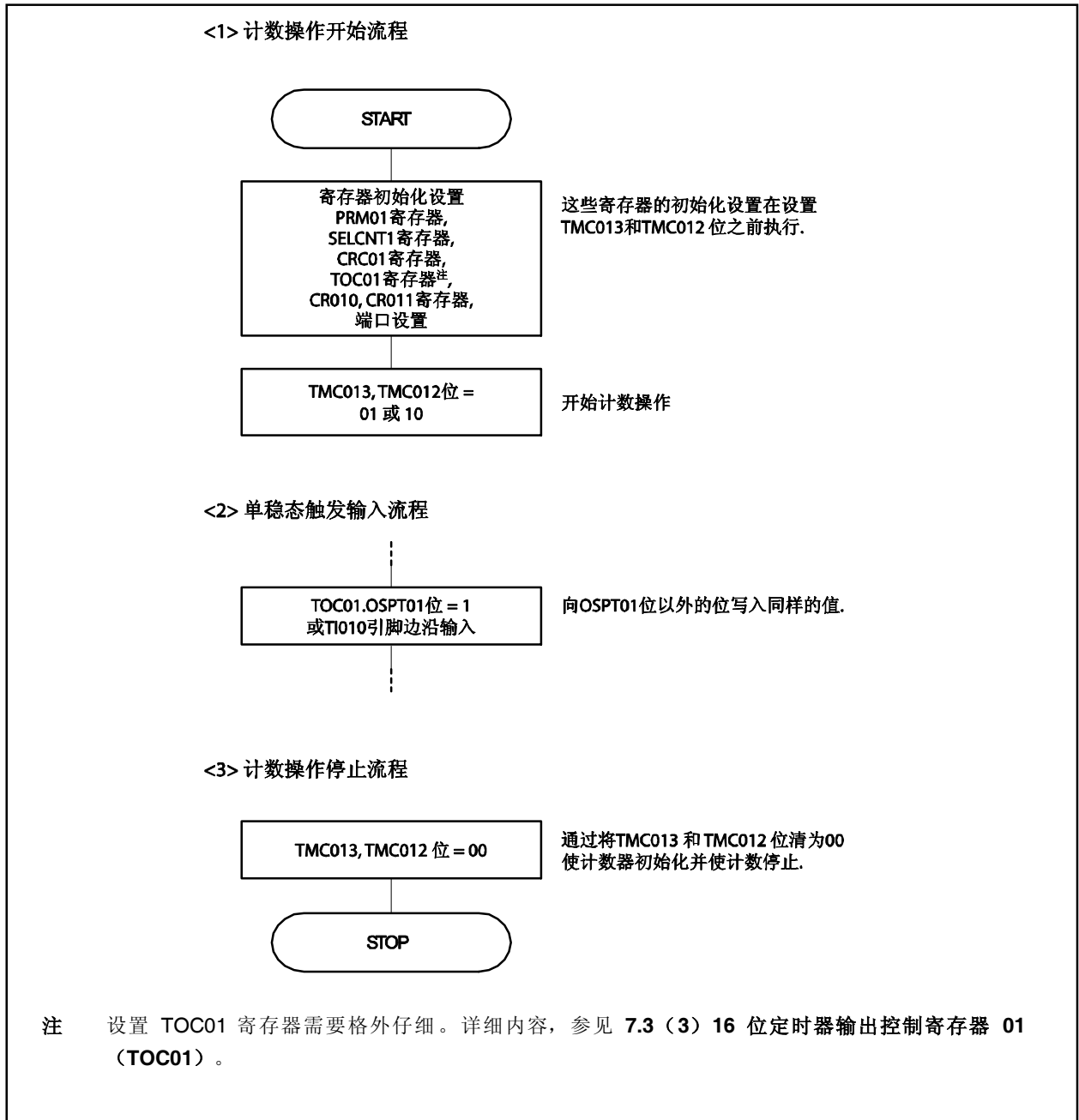
输出单脉冲时，这个寄存器用于比较寄存器。当 TM01 寄存器的值与 CR011 寄存器的指相等时，产生中断信号 (INTTM011)，TO01 引脚的输出电平反转。

图 7-36. 单脉冲输出操作的软件处理示例 (1/2)



- 单脉冲触发输入到单脉冲输出的时间
= $(M + 1) \times \text{计数时钟周期}$
- 单脉冲输出有效电平宽度
= $(N - M) \times \text{计数时钟周期}$

图 7-36. 单脉冲输出操作的软件处理示例 (2/2)



7.4.8 脉冲宽度测量操作

TM01 寄存器可以用于测量 TI010 和 TI011 引脚输入的信号脉冲宽度。

使 16 位定时器/事件计数器 01 工作在自由运行定时器模式，或者从 TI010 引脚输入信号同步重新启动定时器，可以实现测量。

当中断产生时，读取有效捕捉寄存器的值，并测量脉宽。检查 TMC01.OVF01 标志。如果置 (1)，则用软件使之清零。

图 7-37. 脉冲宽度测量框图（自由运行定时器模式）

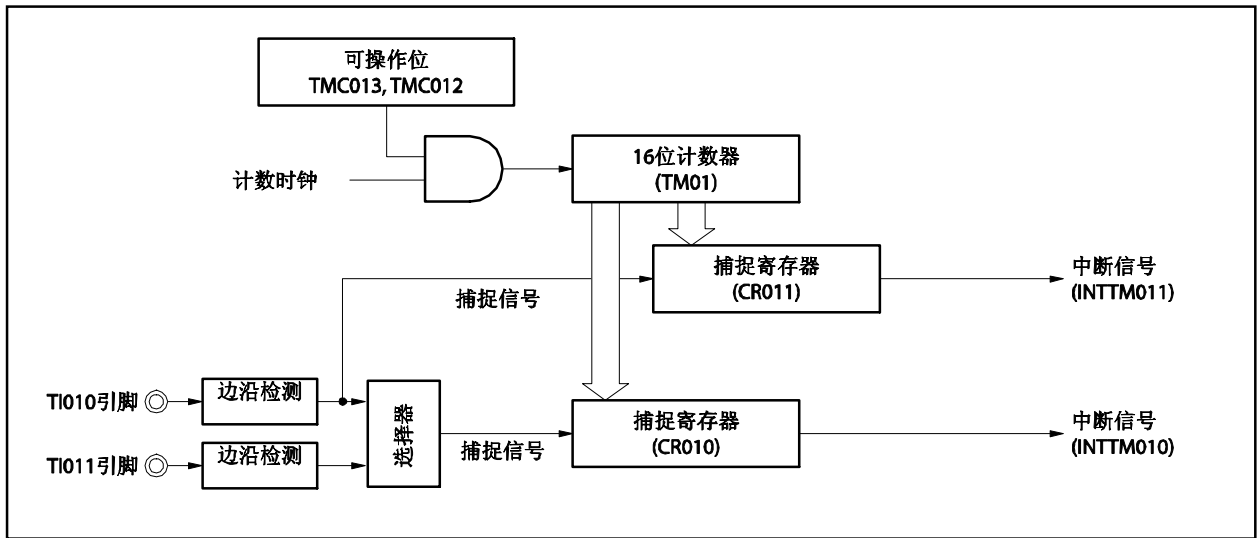
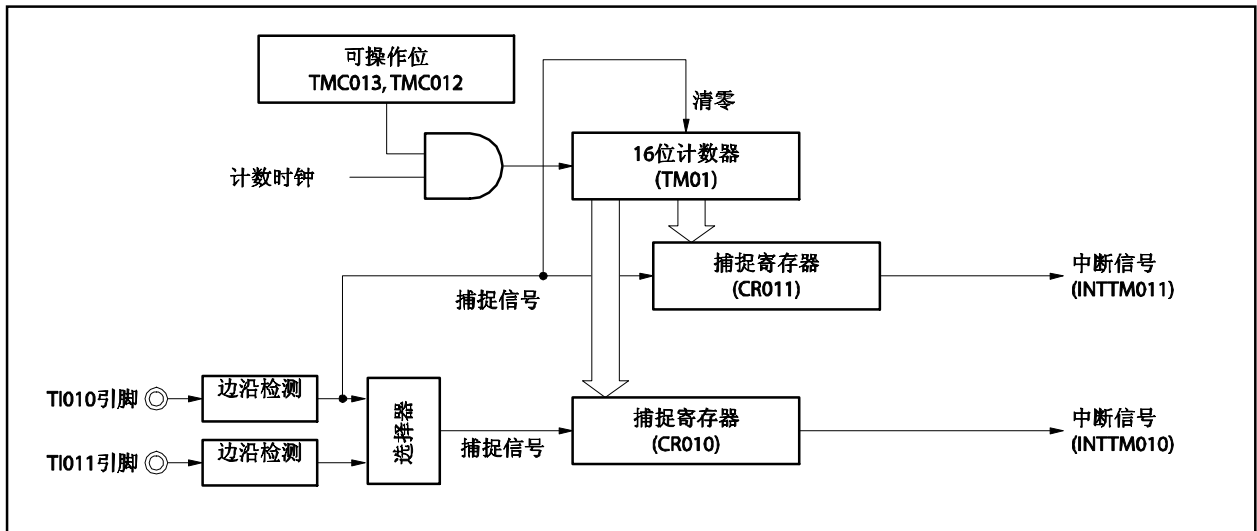


图 7-38. 脉冲宽度测量框图
(通过 TI010 引脚有效边沿输入进入清零并开始模式)



脉冲宽度可用如下三种方法测量。

- 用 TI010 和 TI011 引脚两个输入信号测量脉冲宽度（自由运行定时器模式）
- 用 TI010 引脚一个输入信号测量脉冲宽度（自由运行定时器模式）
- 用 TI010 引脚一个输入信号测量脉冲宽度（通过 TI010 引脚有效边沿输入进入清零并开始模式）

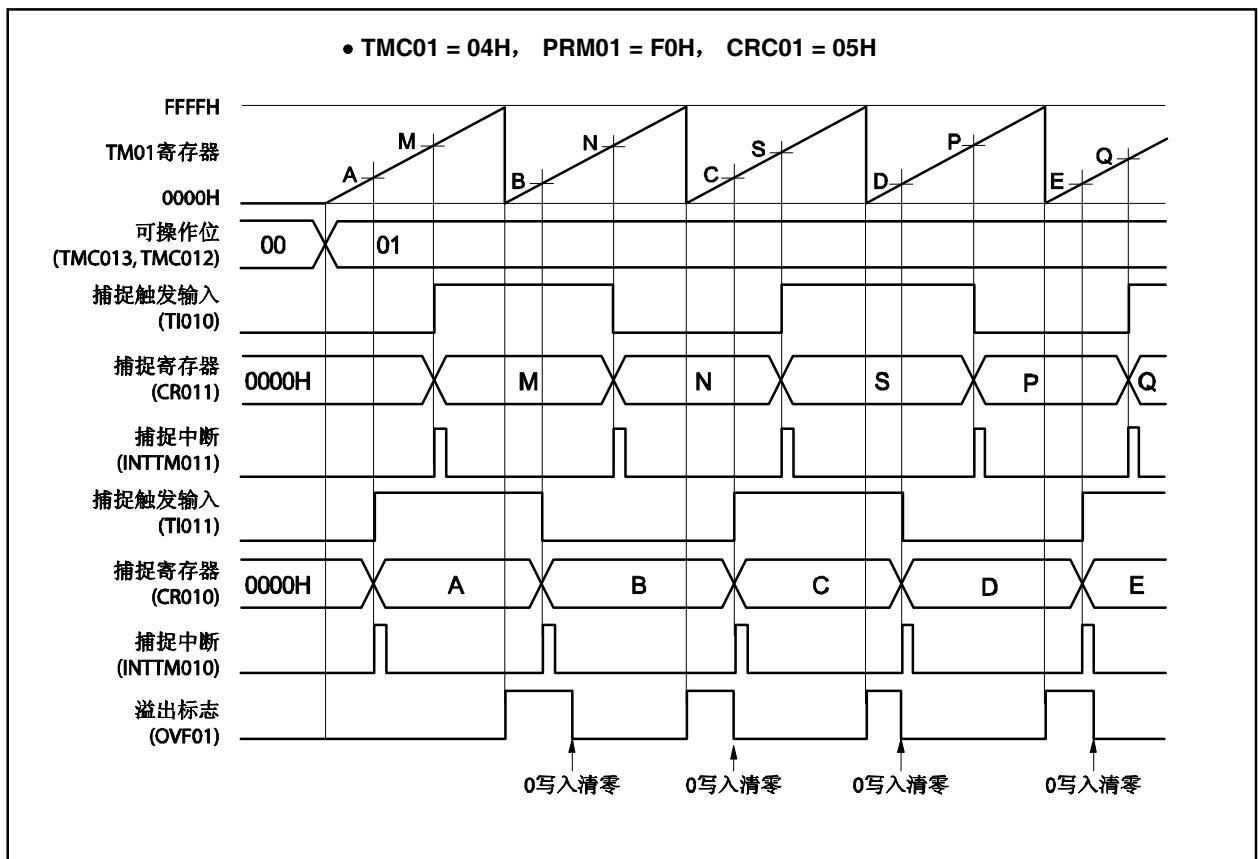
(1) 用 TI010 和 TI011 引脚两个输入信号测量脉冲宽度（自由运行定时器模式）

设置自由运行定时器模式（TMC01.TMC013 和 TMC01.TMC012 位= 01）。当检测到 TI010 引脚的有效边沿时，TM01 寄存器的计数值被捕捉到 CR011 寄存器。当检测到 TI011 引脚的有效边沿时，TM01 寄存器的计数值被捕捉到 CR010 寄存器。指定检测 TI010 和 TI011 引脚的双边沿。

通过这种测量方法，每个输入信号边沿捕捉到的计数值减去之前的计数值，因此需要单独一个寄存器预先保存之前捕捉到的数值。

如果发生溢出，依然减去之前捕捉的值，则发生借位（PSW.CY 位置 1），结果变为负数。如果发生这种情况，忽略 CY，取脉宽的计算值。此外，要将 TMC01.OVF01 位清零。

图 7-39. 脉宽测量时序示例 (1)



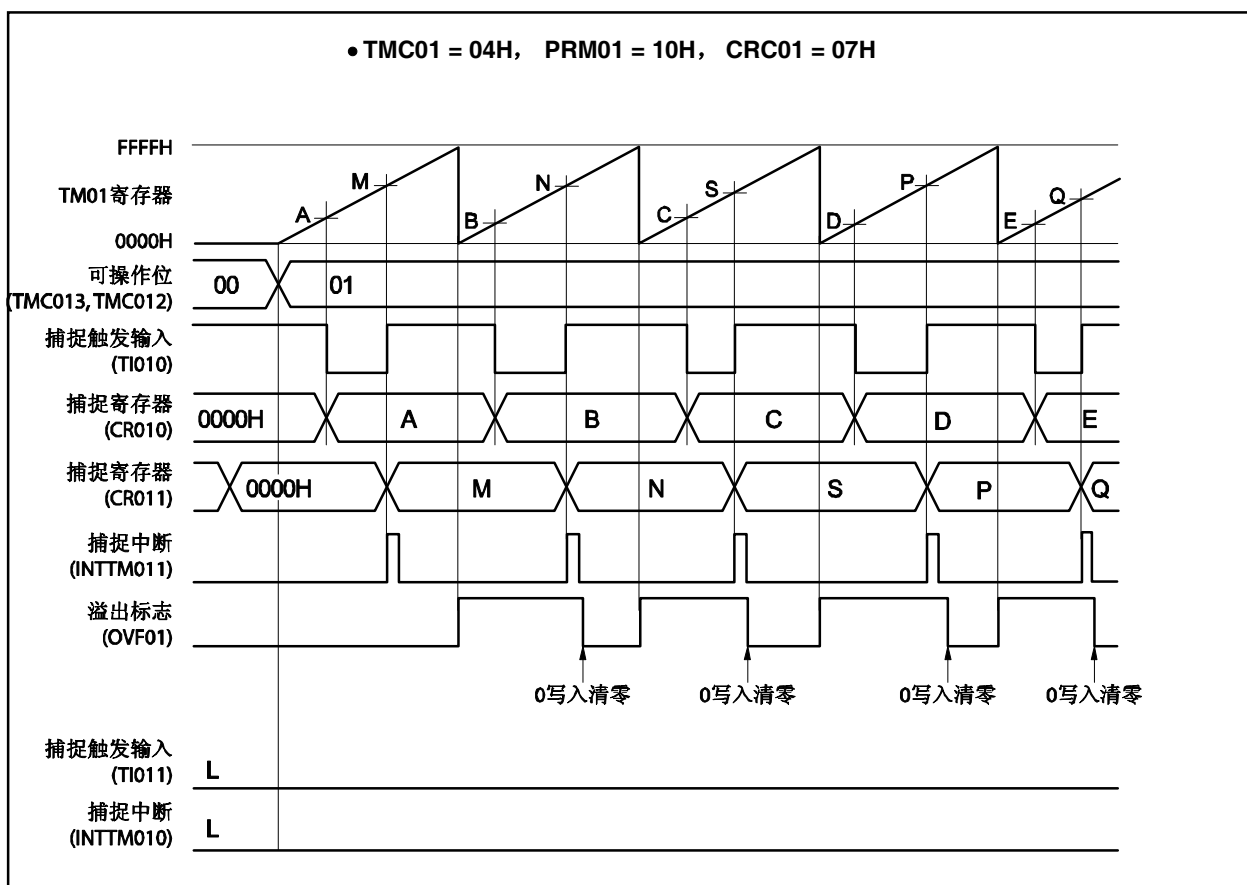
(2) 用 TI010 引脚输入信号测量脉冲宽度 (自由运行定时器模式)

设置自由运行定时器模式 (TMC01.TMC013 和 TMC01.TMC012 位 = 01)。检测到 TI010 引脚有效边沿的相反相位时, 将 TM01 寄存器的计数值捕捉到 CR010 寄存器中。当检测到 TI010 引脚的有效边沿时, TM01 寄存器的计数值捕捉到 CR011 寄存器中。

通过这种测量方法测量脉宽的一个边沿到另一个边沿时, 数值存入两个独立的捕捉寄存器。因此, 捕捉值不需要存储。一个捕捉寄存器的值减去另一个捕捉寄存器的值, 测出一个高电平宽度, 一个低电平宽度, 即可计算周期。

如果发生溢出, 仍然从一个寄存器的值中减去另一个的值, 则产生借位 (PSW.CY 位置 1), 结果变为负数。如果发生了这种情况, 忽略 CY, 取脉冲宽度的计算值。此外, 清零 TMC01.OVF01 位。

图 7-40. 脉冲宽度测量的时序示例 (2)



(3) 用 TI010 引脚一个输入信号测量脉冲宽度 (通过 TI010 引脚输入有效边沿进入清零并开始模式)

设置通过 TI010 引脚有效边沿进入清零并开始模式 (TMC01.TMC013 和 TMC01.TMC012 位 = 10)。检测到 TI010 引脚有效边沿的相反相位时, TM01 寄存器的计数值被捕捉到 CR010 寄存器中。检测到 TI010 引脚的有效边沿时, TM01 寄存器的计数值被捕捉到 CR011 寄存器中且 TM01 寄存器清零 (0000H)。因此, 如果 TM01 寄存器没有溢出, 一个周期就存入 CR011 寄存器中。
如果发生溢出, 将存入 CR011 寄存器中的数值加 10000H, 即为一个周期。清零 TMC01.OVF01 位。

图 7-41. 脉冲宽度测量的时序示例 (3)

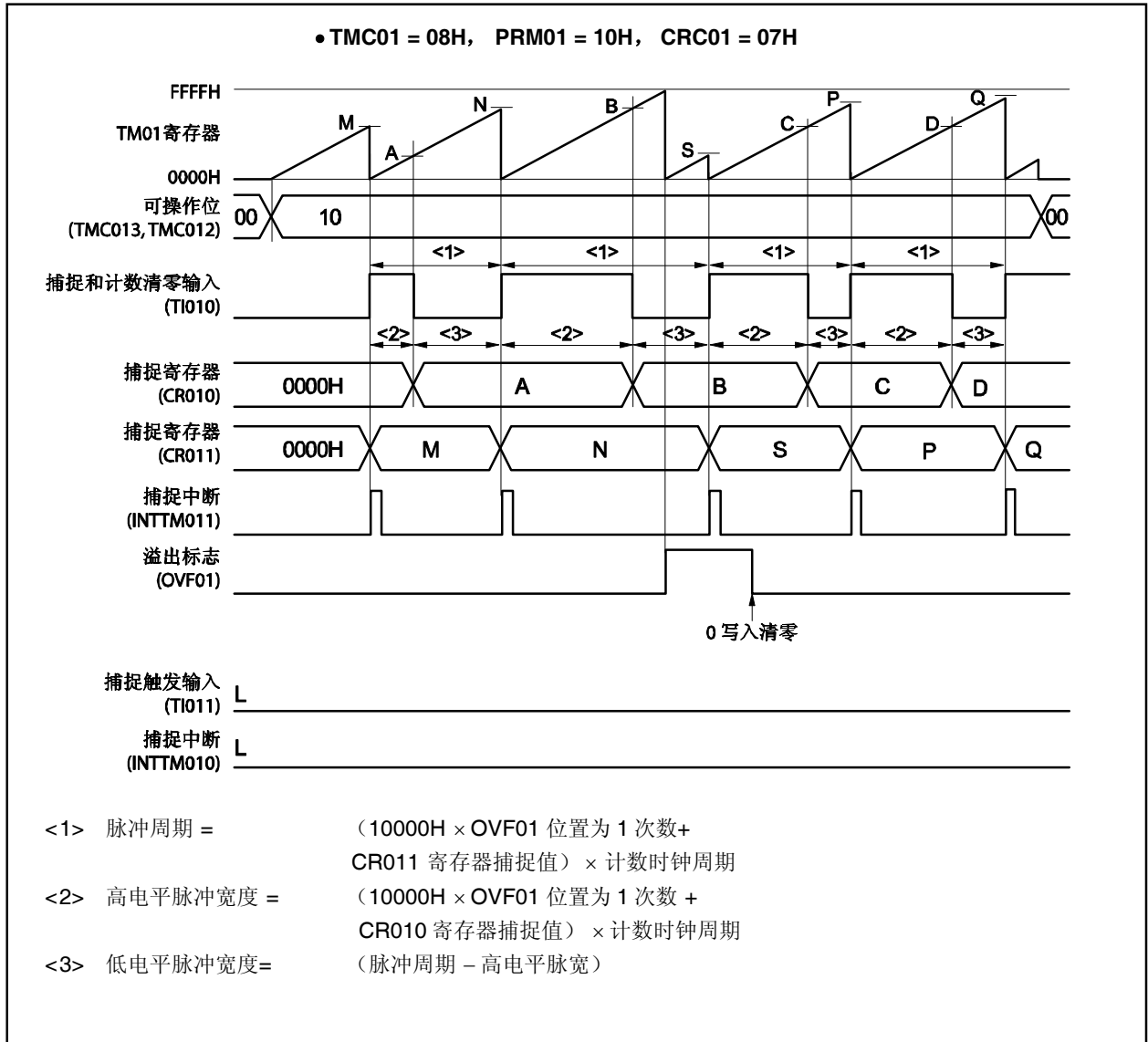


图 7-42. 脉冲宽度测量寄存器设置示例 (1/2)

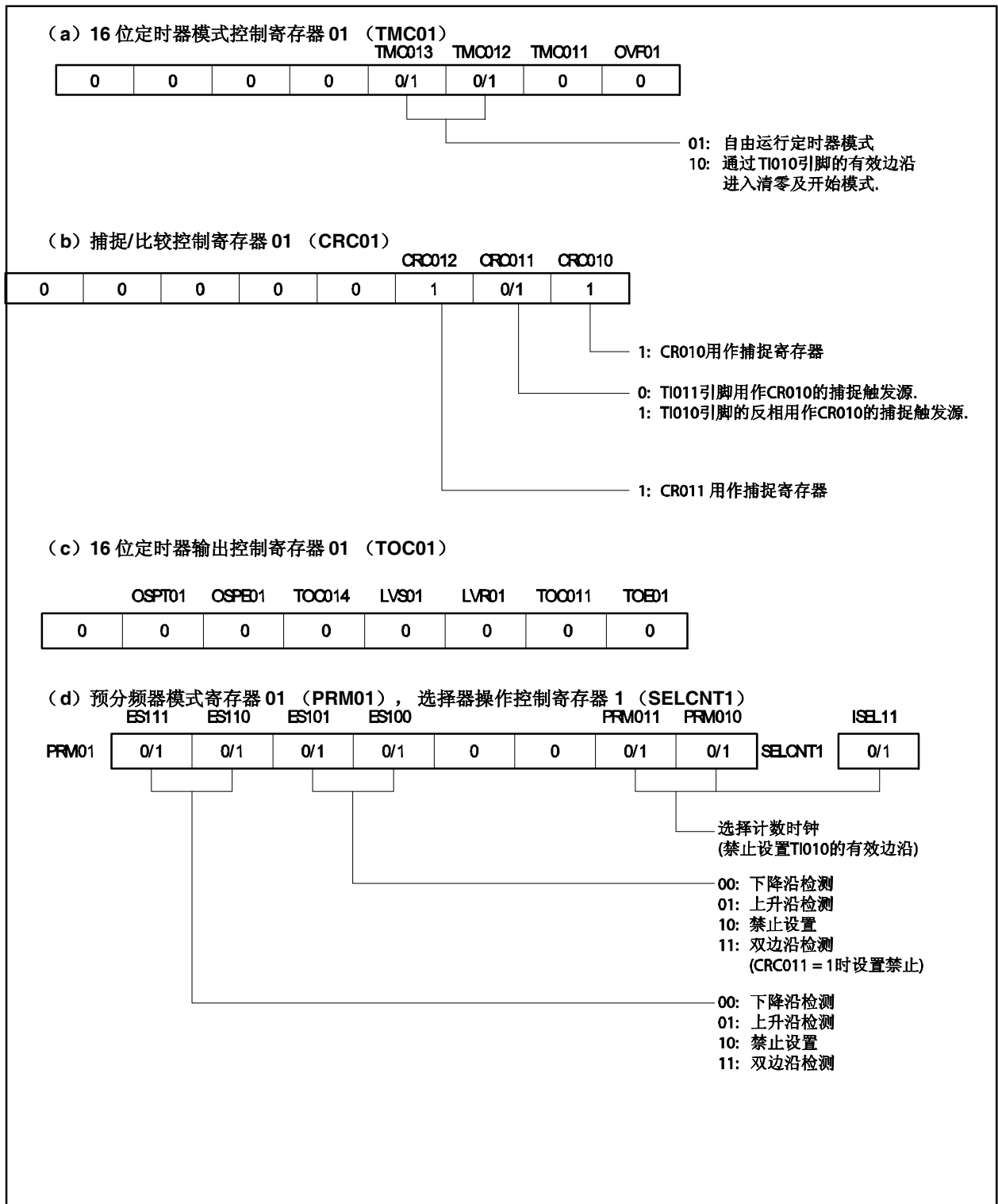


图 7-42. 脉冲宽度测量寄存器设置示例 (2/2)

(e) 16 位定时器计数器 01 (TM01)

通过读 TM01 寄存器，可以读取计数值。

(f) 16 位捕捉/比较寄存器 010 (CR010)

该寄存器可以用于捕捉寄存器。TI010 或 TI011 引脚可以选择作为捕捉触发源。当检测到捕捉触发源的指定边沿时，TM01 寄存器的计数值存入 CR010 寄存器中。

(g) 16 位捕捉/比较寄存器 011 (CR011)

该寄存器用于捕捉寄存器。输入 TI010 引脚的信号用于捕捉触发源。当检测到捕捉触发源，TM01 寄存器的计数值存入 CR011 寄存器。

图 7-43. 脉冲宽度测量软件处理示例 (1/2)

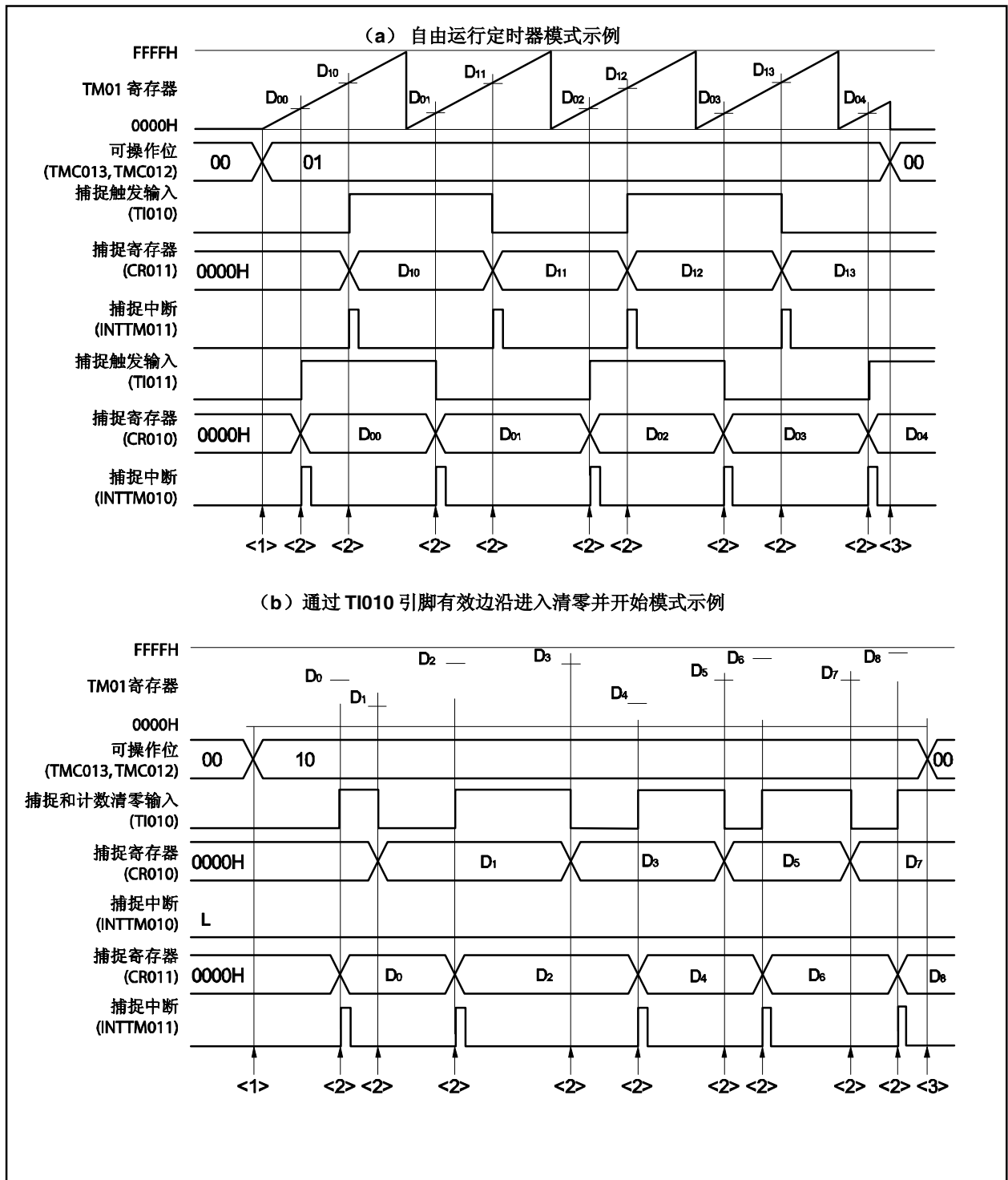
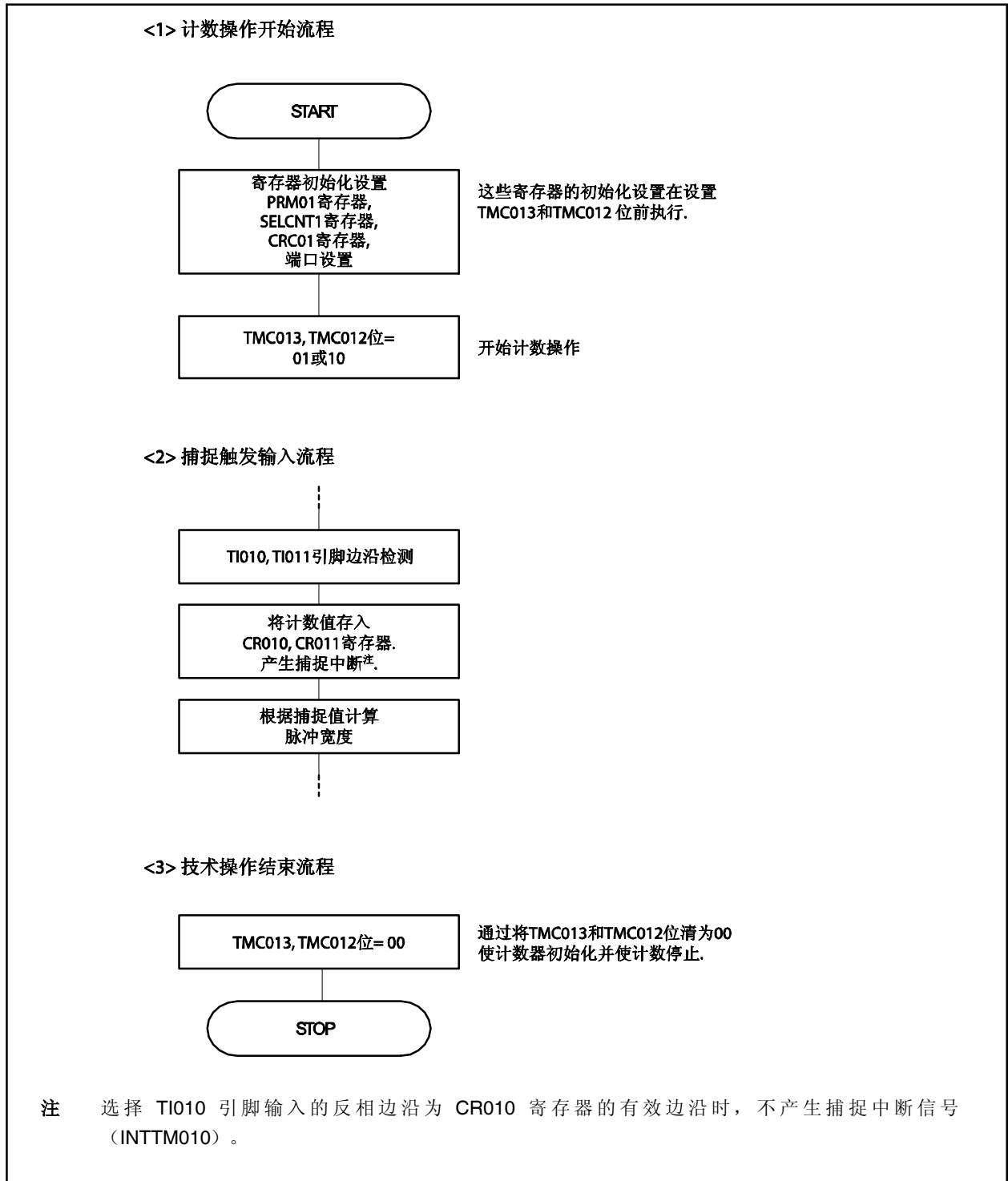


图 7-43. 脉冲宽度测量软件处理示例 (2/2)



7.5 TM01 的特殊应用

7.5.1 TM01 操作过程中改写 CR011 寄存器

原则上说，V850ES/KE2 中的 CR010 和 CR011 寄存器用作比较寄存器时，禁止在 TM01 寄存器操作时（TMC01.TMC013 和 TMC01.TMC012 位= 00 以外的数）改写 CR010 和 CR011 寄存器。

但如果 CR011 寄存器用于 PPG 输出，则如下过程可以改变 CR011 寄存器的值并改变占空比因子，即使在 TM01 寄存器操作中也是如此。（在 CR011 寄存器的值与 TM01 寄存器的值相等后立即改变 CR011 寄存器的值。如果 CR011 寄存器的值在即将与 TM01 寄存器相等之前被改变，则会执行一个不可预料的操作）。

改变 CR011 寄存器的过程

- <1> 禁止中断 INTTM011（TM0IC10.TM0MK11 位 = 1）。
- <2> TM01 寄存器的值与 CR011 寄存器的值相等时禁止定时器输出电平反转。（TOC01.TOC014 位= 0）。
- <3> 改变 CR011 寄存器的值。
- <4> 等待 TM01 寄存器的一个时钟周期。
- <5> 使能 TM01 寄存器的值与 CR011 寄存器的值相等时定时器输出电平反转（TOC01.TOC014 位= 1）。
- <6> 将 INTTM011 的中断标志清零（TM0IC10.TM0IF11 位= 0）。
- <7> 使能中断 INTTM011（TM0IC10.TM0MK11 位= 0）。

备注 关于 TM0IC10 寄存器，参见第十七章 中断/异常处理功能。

7.5.2 设置 LVS01 和 LVR01 位

(1) LVS01 和 LVR01 位的用途

TOC01.LVS01 和 TOC01.LVR01 位用于设置 TO01 引脚输出的默认值并在不使能定时器操作时（TMC01.TMC013 和 TMC01.TMC012 位 = 00）反转定时器输出电平。在不需要软件控制时，将 LVS01 和 LVR01 清为 00（缺省值：低电平输出）。

LVS01 位	LVR01 位	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	禁止设置

(2) 设置 LVS01 和 LVR01 位

用如下步骤设置 LVS01 和 LVR01 位。

图 7-44. 设置 LVS01 和 LVR01 位流程示例

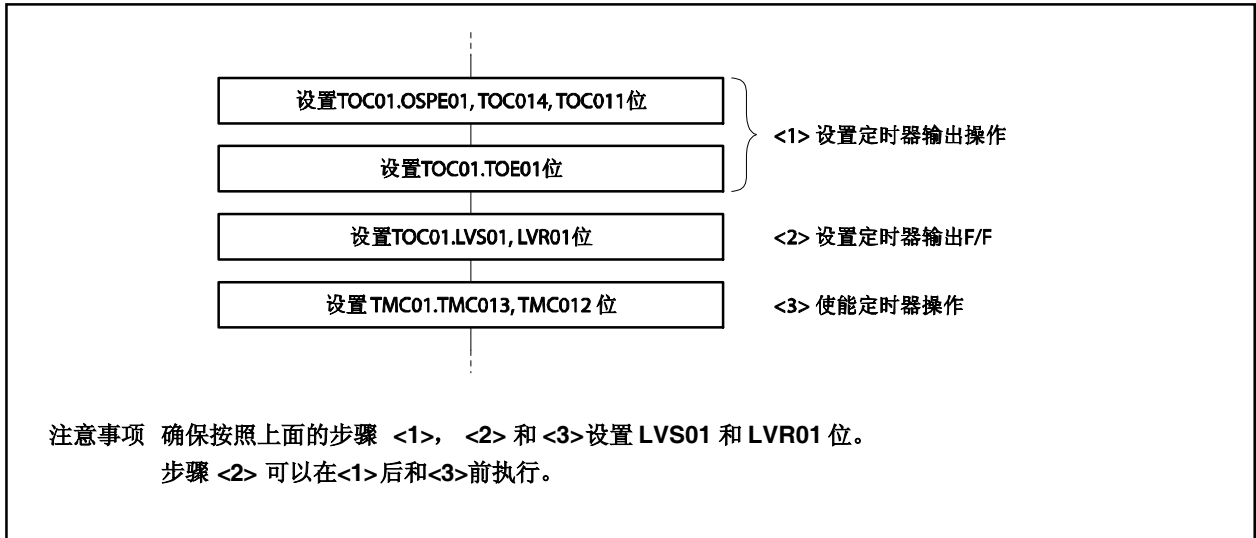
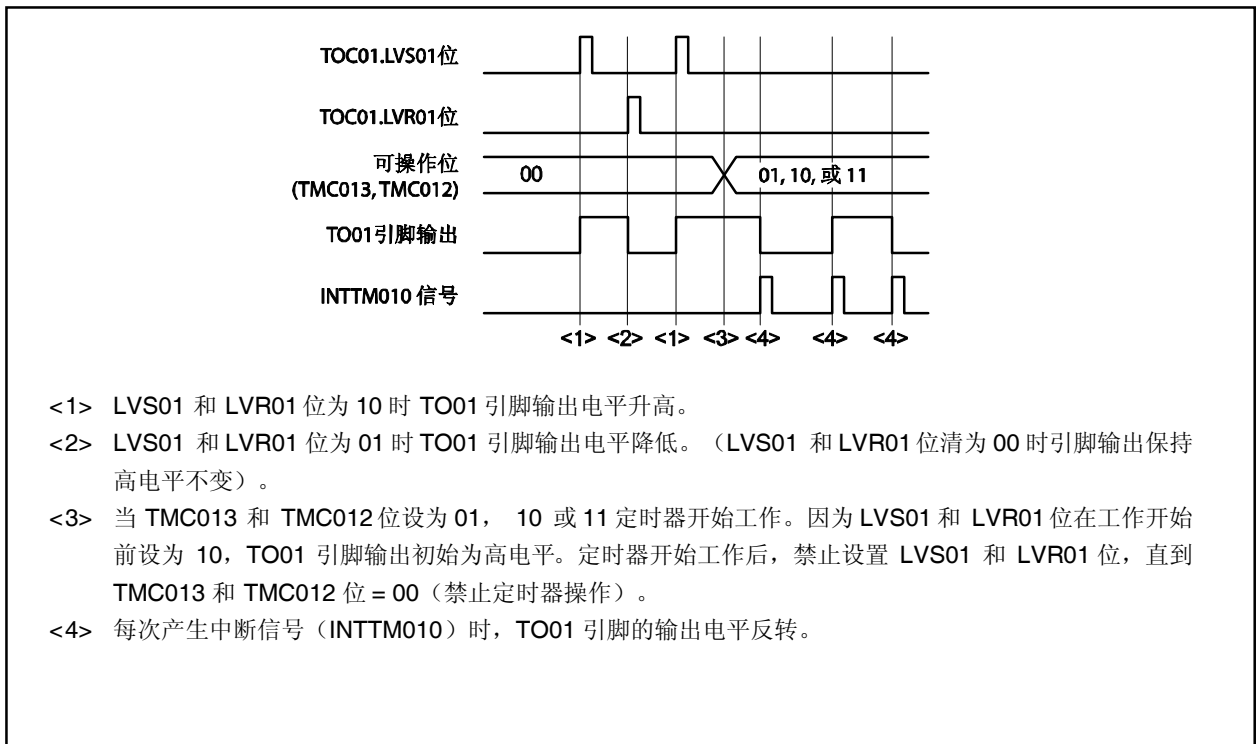


图 7-45. LVR01 和 LVS01 位时序示例



7.6 注意事项

(1) TI010/TO01 引脚复用功能

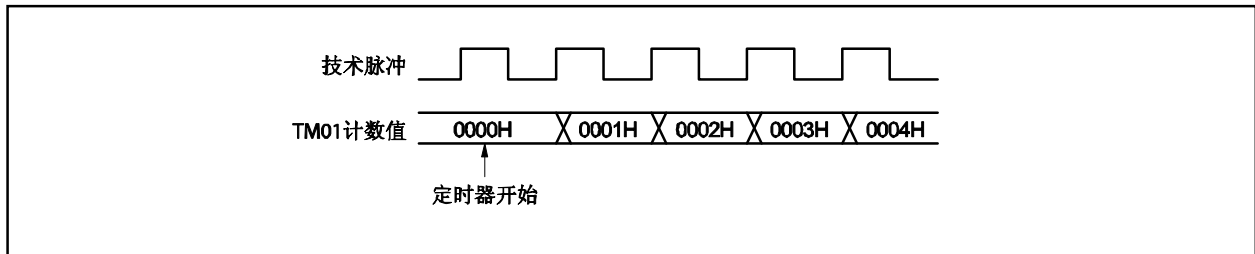
通道	引脚	复用功能	备注
TM01	TI010	P35/TO01	与 TO01 共用此引脚。
	TI011	P50/KR0/RTP00	-
	TO01	P32/ASCK0/ADTRG	.连接到 2 个引脚, P32 和 P35。
		P35/TI010	

- 在检测 TI010 引脚作为触发源的有效边沿时执行单脉冲输出，需用 TO01 输出，复用为 P32。
当使用 TO01 引脚输出且复用为 P35 时，如果 TI010 引脚同样复用为 P35 则不可使用。
当仅使用软件触发源（置（1） TOC01.OSPT01 位）作为单脉冲输出的启动触发源时，P32 和 P35 任意一个可以作为 TO01 引脚的输出。
- 通过检测 TI010 引脚输入的有效边沿执行 TO01 引脚输出反转操作时，用 TO01 输出，复用为 P32。
当使用 TO01 引脚输出且复用为 P35 时，如果 TI010 引脚同样复用为 P35 则不可使用。使用 TO01 引脚且复用为 P35 时，将 TMC01.TMC011 位清零。

(2) 启动定时器的误差

定时器启动后，相等信号产生前，最高可产生 1 个时钟周期的误差。这是因为 TM01 寄存器的计数操作的启动是与计数脉冲同步的。

图 7-46. TM01 寄存器计数启动时序



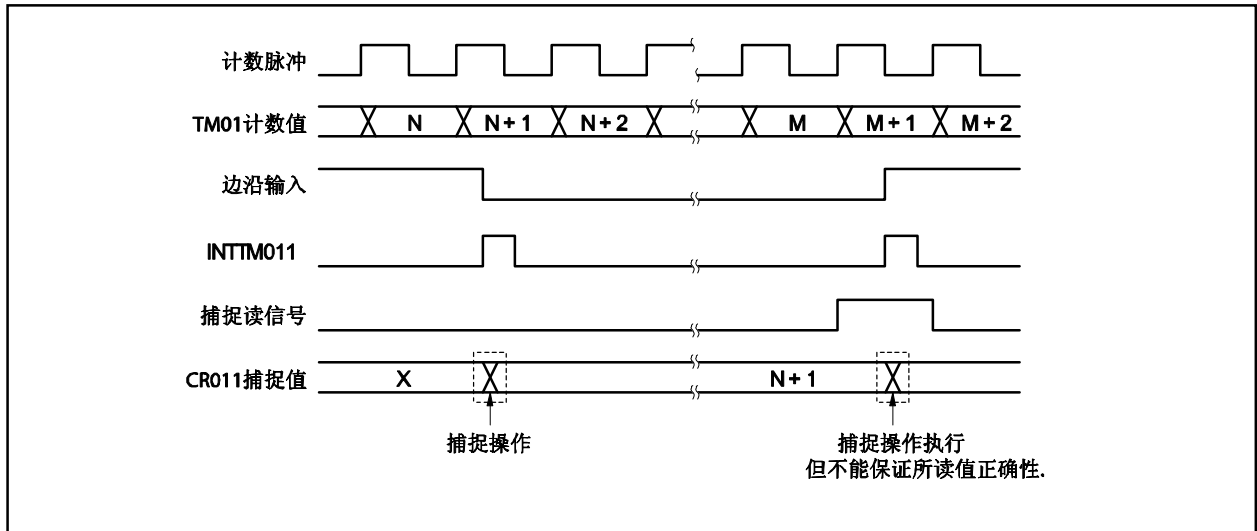
(3) 设置 CR010 和 CR011 寄存器（TM01 寄存器和 CR010 寄存器相等时清零并开始的模式下）

设置 CR010 和 CR011 寄存器为 0000H 以外的值（当这些寄存器作为外部事件计数器时，单脉冲计数操作是不能实现的）。

(4) 捕捉寄存器的数据保持时间

- (a) 如果在读 CR010/CR011 寄存器时，TI011/TI010 引脚输入有效边沿，则 CR010/CR011 执行捕捉操作，但此时所读值正确性不受保证。但是由于检测到有效边沿，产生中断请求信（INTTM010/INTTM011）。

图 7-47. 捕捉寄存器的数据保持时间



- (b) 在 16 位定时器/事件计数器 01 已经停止后，CR010 和 CR011 寄存器的值不能得到保证。

(5) 设置有效边沿

在定时器操作停止时（TMC01.TMC013 和 TMC01.TMC012 位 = 00）设置 TI010 引脚的有效边沿。通过 PRM01.ES100 和 PRM01.ES101 位设置有效边沿。

(6) 重触发单脉冲

保证在单脉冲输出模式正在输出有效电平时不产生触发。
保证在当前有效电平输出完毕后输入下一个触发源。

(7) OVF01 标志的操作

(a) OVF01 标志的设置

除了 TM01 寄存器溢出的情况以外，以下情况也使 TMC01.OVF01 标志设置为 1。

选择如下模式：TM01 寄存器和 CR010 寄存器相等时清零并开始。

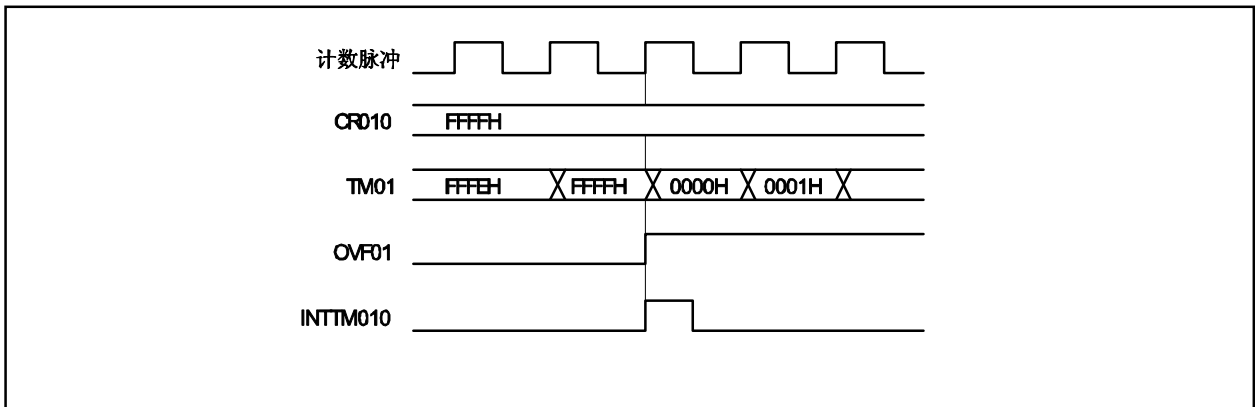


将 CR010 寄存器设置为 FFFFH



当 TM01 寄存器从 FFFFH（与 CR010 寄存器相等）清为 0000H 时

图 7-48. OVF01 标志操作时序



(b) OVF01 标志的清零

TM01 寄存器溢出后，清零 OVF01 标志无效，即使清零是在下一个计数时钟被计数操作前进行（在 TM01 寄存器变为 0001H 之前进行）。

(8) 单脉冲输出

单脉冲输出通常在如下两种模式之一，自由运行定时器模式和清零并开始发生在 TI010 引脚有效边沿时的模式。在如下模式单脉冲输出无法实现，清零并开始发生在 TM01 寄存器和 CR010 寄存器相等时的模式。

(9) 捕捉操作**(a) 如果指定 TI010 引脚的有效边沿为计数时钟**

如果指定 TI010 引脚的有效边沿为计数时钟，指定 TI010 引脚为触发源的捕捉寄存器工作不正常。

(b) 保证 TI011 和 TI010 引脚输入的信号被正确捕捉

为了精确的捕捉计数值，TI010 和 TI011 引脚输入的捕捉触发源脉冲宽度必须大于两个计数时钟周期，计数时钟由 PRM01 和 SELCNT1 寄存器选择。

(c) 中断信号产生

尽管捕捉操作在计数时钟的下降沿执行，中断请求信号（INTTM010，INTTM011）在下一个计数时钟的上升沿产生。

(d) 注意当 CRC01.CRC011 位置为 1 时

当 TM01 寄存器的计数值在 TI010 引脚输入信号的相反相位被捕捉到 CR010 寄存器，计数值被捕捉后不产生中断信号（INTTM010）。如果在此操作中 TI011 引脚上检测到有效边沿，捕捉操作不执行，但产生 INTTM010 信号，作为外部中断信号。

(10) 边沿检测**(a) 复位后指定有效边沿**

如果复位后 16 位定时器/事件计数器 01 的操作被禁止且 TI010 或 TI011 引脚处于高电平状态且上升沿或者双边沿指定为 TI010 或 TI011 引脚的有效边沿，则 TI010 或 TI011 引脚的高电平被检测为上升沿。注意到这点当 TI010 或 TI011 引脚上拉时。但是当操作先停止再重新使能时，上升沿不被检测。

(b) 噪声消除采样时钟

根据 TI010 引脚的有效边沿用于计数时钟或者捕捉触发源，噪声消除的采样时钟有所不同。在前者的情况，采样以 $f_{xx}/4$ 执行，在后者的情况，采样计数时钟是由 PRM01 和 SELCNT1 寄存器选择的。当输入到 TI010 引脚的信号被采样，且连续两次检测到有效边沿，则认为检测到有效边沿。因此，脉冲宽度较短的噪声可以被消除。

备注 fxx: 主时钟频率

第八章 8 位定时器/事件计数器 5

V850ES/KE2 提供两信道 8 位定时器/事件计数器 5。

8.1 功能

8 位定时器/事件计数器 5n 具有以下两个模式 ($n = 0, 1$)。

- 仅使用 8 位定时器/事件计数器的模式 (单独模式)
- 使用级联的模式 (16 位分辨率: 级联模式)

这两种模式描述如下:

(1) 仅使用 8 位定时器/事件计数器的模式 (单独模式)

8 位定时器/事件计数器 5n 作为一个 8 位定时器/事件计数器运行。使用以下功能。

- 内部定时器
- 外部事件计数器
- 方形波输出
- PWM 输出

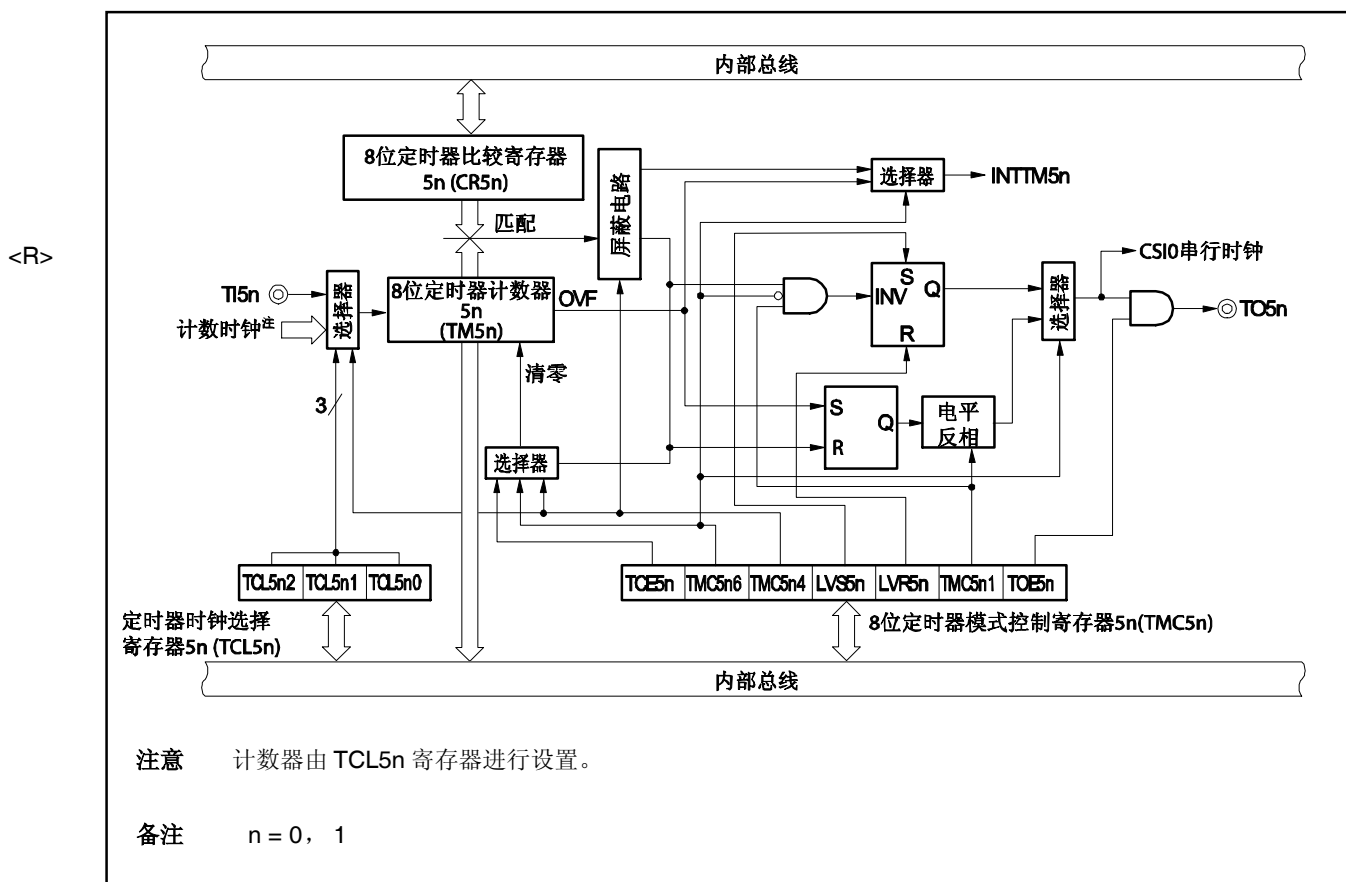
(2) 使用级联的模式 (16 位分辨率: 级联模式)

8 位定时器/事件计数器 5n 以级联模式连接 TM5n 寄存器, 作为一个 16 位定时器/事件计数器运行。它使用以下功能。

- 内部定时器具有 16 位分辨率
- 外部事件计数器具有 16 位分辨率
- 方形波输出具有 16 位分辨率

8 位定时器/事件计数器 5n 的模块图显示如下。

图 8-1. 8 位定时器/事件计数器 5n 的模块图



8.2 配置

8 位定时器/事件计数器 5n 包括以下硬件。

表 8-1. 8 位定时器/事件计数器 5n 的配置

项目	配置
定时器寄存器	8 位定时器计数器 5n (TM5n) 16 位定时器计数器 (TM5)：仅用于级联模式
寄存器	8 位定时器比较寄存器 5n (CR5n) 16 位定时器比较寄存器 5 (CR5)：仅用于级联模式
定时器输出	1 (TO5n 引脚)
控制寄存器 ^{说明}	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) 16 位定时器模式控制寄存器 5 (TMC5)：仅用于级联模式

注 当使用 TI5n 和 TO5n 引脚的功能时并且端口引脚用于复用功能时，参考表 4-12 当端口引脚用于复用功能时的设置。

备注 n = 0, 1

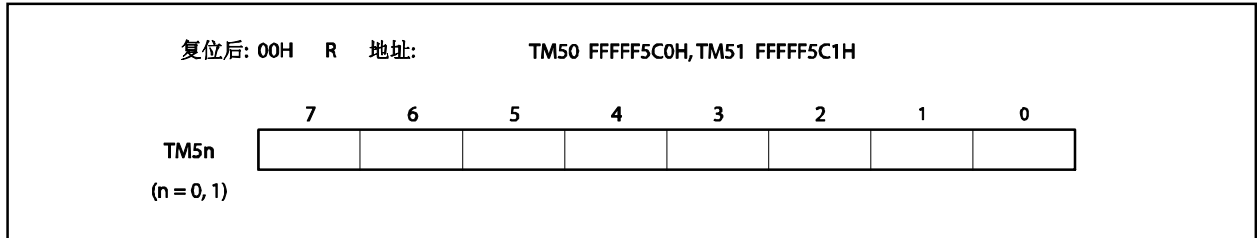
(1) 8 位定时器计数器 5n (TM5n)

TM5n 寄存器是一个 8 位只读寄存器，其对脉冲计数。

此计数器与计数时钟的上升沿同步增长。

通过级联连接方式，TM5n 寄存器可以用于 16 位定时器。

当以级联方式将 TM50 和 TM51 寄存器用作 16 位定时器，这些寄存仅在 16 位设备中可以读取。因此，读取这些寄存器两次并且比较他们的值，考虑在计数变化中进行读取。



在以下情况下，计数值被复位为 00H。

<1> 复位

<2> 当使用 TMC5n 时，TCE5n 位被清除 (0)。

<3> 当 TM5n 寄存器和 CR5n 寄存器之间相等时，如果进行清除 & 启动动作，则 TM5n 寄存器和 CR5n 寄存器相等。

注意事项 当以级联方式连接时，甚至最低定时器 (TM50) 被清除时，这些寄存器成为 000H。

备注 n = 0, 1

(2) 8 位定时器比较寄存器 5n (CR5n)

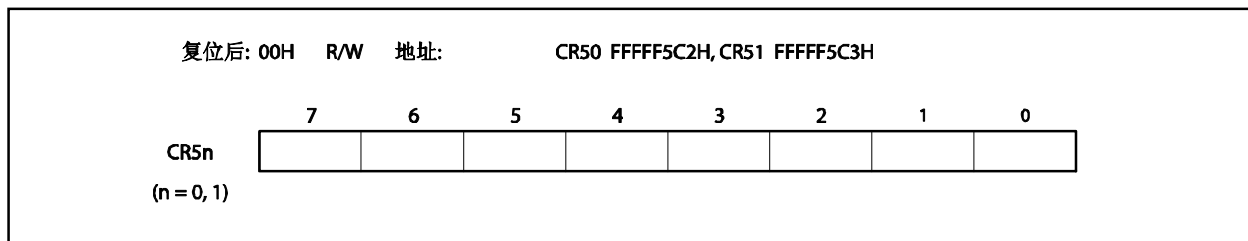
在 8 位设备中，CR5n 寄存器可以进行读取并写入。

在非 PWM 模式中，CR5n 寄存器中设置的值总是与 TM5n 寄存器的计数值相比较，并且如果两个值相等，则产生中断请求信号 (INTTM5n)。

在 PWM 模式中，TM5n 溢出会使 TO5n 引脚输出变为有效电平，当 TM5n 寄存器的值与 CR5n 寄存器的值相等时，TO5n 引脚输出变为非有效电平。

CR5n 寄存器的值可以设置为 00H ~ FFH。

当以级联方式使用 TM50 寄存器和 TM51 寄存器作为 16 位定时器时，CR50 寄存器和 CR51 寄存器作为 16 位比较寄存器 5 (CR5) 运行。比较 16 位长度的计数器值与寄存器值，如果他们相等，则产生中断请求信号 (INTTM50)。



- 注意事项**
1. 当 TM5n 寄存器和 CR5n 寄存器 (TMC5n.TMC5n6 位 = 0) 相等并且进行清除&启动时，在计数操作中不要写不同值到 CR5n 寄存器中。
 2. 在 PWM 模式中，设置 CR5n 寄存器重写间隔为三个或更多计数时钟 (用 TCL5n 寄存器选择时钟)。
 3. 当使用级联连接方式时，在改变 CR5n 寄存器值之间，确保停止定时器。

备注 n = 0, 1

8.3 寄存器

以下两个寄存器用于控制 8 位定时器/事件计数器 5n。

- 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)

备注 为了使用 TI5n 和 TO5n 引脚的功能之间，当端口引脚用于交换功能时，参考表 4-12 当端口引脚用于复用功能时的设置。

(1) 定时器时钟选择寄存器 5n (TCL5n)

TCL5n 寄存器设置 8 位定时器/事件计数器 5n 的计数时钟和 TI5n 引脚输入的有效边。

TCL5n 寄存器支持 8 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: TCL50 FFFF5C4H, TCL51 FFFF5C5H

	7	6	5	4	3	2	1	0
TCL5n	0	0	0	0	0	TCL5n2	TCL5n1	TCL5n0

(n = 0, 1)

TCL5n2	TCL5n1	TCL5n0	计数时钟选择 ^注		
			Clock	f _{xx}	
				20 MHz	10 MHz
0	0	0	TI5n的降沿	-	-
0	0	1	TI5n的升沿	-	-
0	1	0	f _{xx}	禁止设置	100 ns
0	1	1	f _{xx} /2	100 ns	200 ns
1	0	0	f _{xx} /4	200 ns	0.4 ns
1	0	1	f _{xx} /64	3.2 ns	6.4 ns
1	1	0	f _{xx} /256	12.8 ns	25.6 ns
1	1	1	INTTM010	-	-

注 当选择间隔时钟时，设置此值以满足以下条件。

V_{DD} = 4.0 ~ 5.5 V: 计数时钟 ≤ 10 MHz

V_{DD} = 2.7 ~ 4.0 V: 计数时钟 ≤ 5 MHz

注意事项 在用不同数据重写 TCL5n 寄存器时，停止定时器操作。

备注 当 TM5n 寄存以级联方式连接时，TCL51 寄存器设置有效。

(2) 8 位定时器模式控制寄存器 5n (TMC5n)

TMC5n 寄存器执行以下 6 种设置。

- 控制 TM5n 寄存器的计数
- 选择 TM5n 寄存器的运行模式
- 选择单独模式或级联连接模式
- 选择定时器输出双稳态多谐振荡器的状态
- 控制定时器输出双稳态多谐振荡器或在 PWM 模式（空闲运行定时器）中选择有效电平
- 控制定时器输出

TMC5n 寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: TMC50 FFFF5C6H, TMC51 FFFF5C7H

	<7>	6	5	4	<3>	<2>	1	<0>
TMC5n (n = 0, 1)	TCE5n	TMC5n6	0	TMC514 [‡]	LVS5n	LVR5n	TMC5n1	TOE5n
	TCE5n	8位定时器计数操作及事件计数器 5n的控制						
	0	计数器清0后禁止计数(禁用计数器)						
	1	开始计数操作						
	TMC5n6	8位定时器及事件计数器5n操作模式的选择						
	0	清0及开启开始匹配TM5n和CR5n模式						
	1	PWM(自由运行定时器)模式						
	TMC514	8位定时器及事件计数器51单联模式或续联模式的选择						
	0	单联模式						
	1	续联模式(与8位定时器及事件计数器50的连接)						
	LVS5n	LVR5n	定时器输出F/F状态设置					
	0	0	不变					
	0	1	定时器输出F/F复位到0					
	1	0	定时器输出F/F设置为1					
	1	1	禁止设置					
	TMC5n1	非PWM(自由运行定时器)模式(TMC5n6位=0)			PWM(自由运行定时器)模式(TMC5n6位=1)			
		控制定时器 F/F			选择活跃位			
	0	禁止反向操作			高活性			
	1	容许反向操作			低活性			
	TOE5n	定时器输出控制						
	0	禁止输出(TO5n 引脚处于低端)						
	1	容许输出						

注 TMC50 寄存器的第 4 位固定为 0。

- 注意事项
1. 因为 TO51 和 TI51 是相同引脚的交换功能, 在一个时刻仅一个引脚可以使用。
 2. 在非 PWM 模式下, LVS5n 和 LVR5n 位设置有效。
 3. 不要同时设置以下的<1> ~ <4>项, 设置如下。

<1> 设置 TMC5n1, TMC5n6, 和 TMC514 [‡] 位:	运行模式的设置
<2> 为定时器输出使能设置 TOE5n 位。:	定时器输出使能
<3> 设置 LVS5n 和 LVR5n 位 (小于 2):	定时器输出 F/F 的设置
<4> 设置 TCE5n 位	

- 备注
1. 在 PWM 模式, 通过 TCE5n 位=0 设置 PWM 输出为非有效电平。
 2. 当 LVS5n 和 LVR5n 位读出时, 读出 0 值。
 3. TMC5n6, LVS5n, LVR5n, TMC5n1, 和 TOE5n 位的值反映 TO5n 输出, 不管 TCE5n 位的值是多少。

8.4 运行

8.4.1 作为间隔定时器运行

8 位定时器/事件计数器 5n 作为一个间隔定时器运行，其以 CR5n 寄存器中预设的计数间隔重复产生中断。如果 TM5n 寄存器中的计数值与 CR5n 寄存器的设置的值相等，TM5n 寄存器的值清除并设置为 00H，计数继续，同时，产生一个中断请求信号（INTTM5n）。

设置方法

<1> 设置每个寄存器

- TCL5n 寄存器： 选择计数时钟（t）
- CR5n 寄存器： 比较值（N）
- TMC5n 寄存器： 停止计数操作并选择一种模式，在此模式中，当 TM5n 寄存器和 CR5n 寄存器之间相等时，则发生清除 & 启动动作。（TMC5n 寄存器 = 0000xx00B，x：忽略）。

<2> 当 TMC5n.TCE5n 位设置为 1 时，计数操作开始。

<3> 当 TM5n 寄存器和 CR5n 寄存器的值相等时，则产生 INTTM5n 信号（TM5n 寄存器清除为 00H）。

<4> 然后，以相同时间间隔重复产生 INTTM5n 信号。为了停止计数，设备 TCE5n 位为 0。

$$\text{间隔时间} = (N + 1) \times t; \quad N = 00H \sim FFH$$

注意事项 在间隔定时器操作期间，不要重写 CR5n 寄存器的值。

备注 n = 0, 1

图 8-2. 间隔定时器操作的定时（1/2）

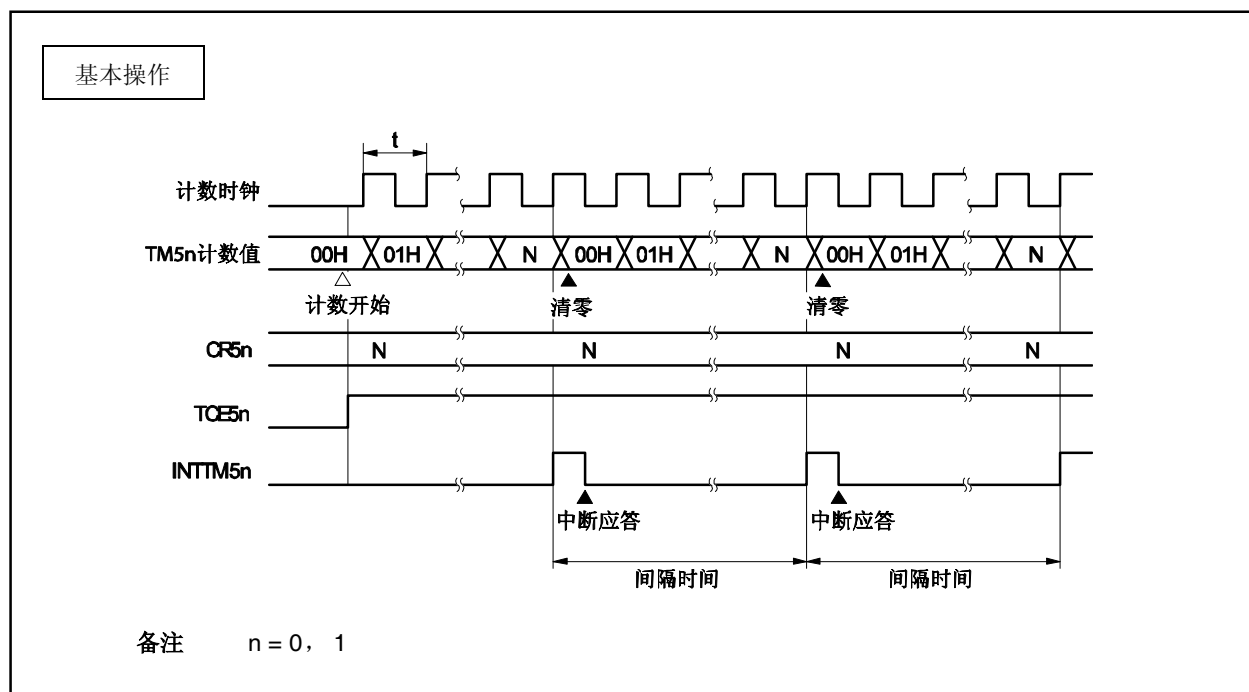
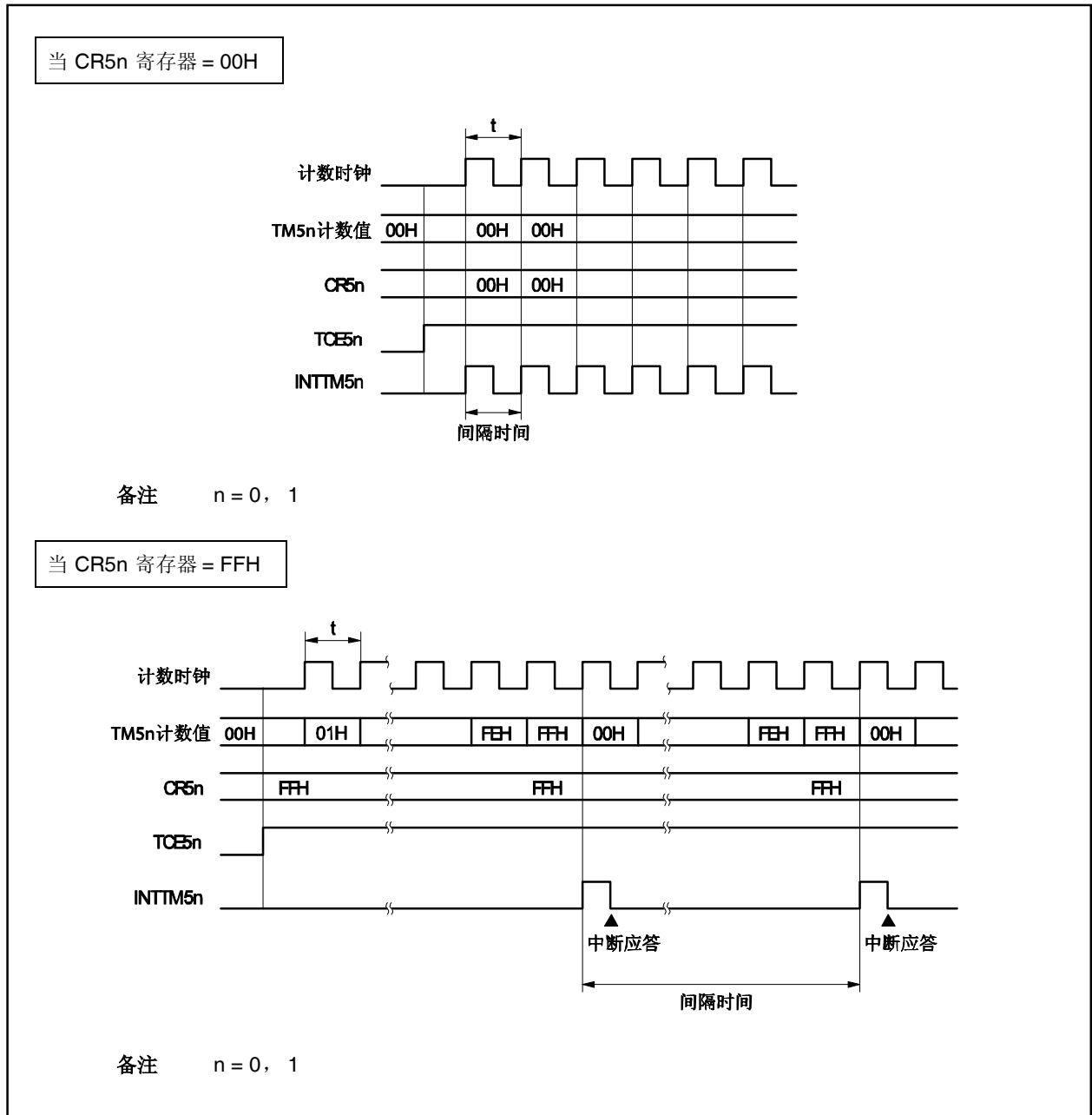


图 8-2. 间隔定时器操作的定时 (2/2)



8.4.2 作为外部事件计数器运行

通过使用 TM5n 寄存器，外部事件计数器对从外部源到 TI5n 引脚的时钟脉冲输入计数。

每次由 TCL5n 寄存器指定的有效沿是 TI5n 引脚的输入，TM5n 寄存器增加，可以指定上升沿或下降沿为有效沿。

当 TM5n 寄存器的计数值与 CR5n 寄存器的值相等时，TM5n 寄存器清除为 00H 并且产生一个中断请求信号 (INTTM5n)。

设置方法

<1> 设置每个寄存器。

- TCL5n 寄存器：选择 TI5n 引脚输入沿。
TI5n 引脚的下降沿 → TLC5n 寄存器 = 00H
TI5n 引脚的上升沿 → TCL5n 寄存器 = 01H
- CR5n 寄存器：比较值 (N)
- TMC5n 寄存器：停止计数运行，选择一个模式，在此模式中，当 TM5n 寄存器和 CR5n 寄存器之间发生相等时，则发生清除 & 启动动作，禁止定时器输出 F/F 倒置操作并禁止定时器输出。
(TMC5n 寄存器 = 0000xx00B, x: 忽略)
- 对于交换功能引脚设置，当端口引脚用于交换功能时，参考表 4-12 当端口引脚用于复用功能时的设置。

<2> 当 TMC5n.TCE5n 位设置为 1 时，计数器对 TI5n 引脚的脉冲输入计数。

<3> 当 TM5n 寄存器和 CR5n 寄存器的值相等时，则产生 INTTM5n 信号 (TM5n 寄存器清除为 00H)。

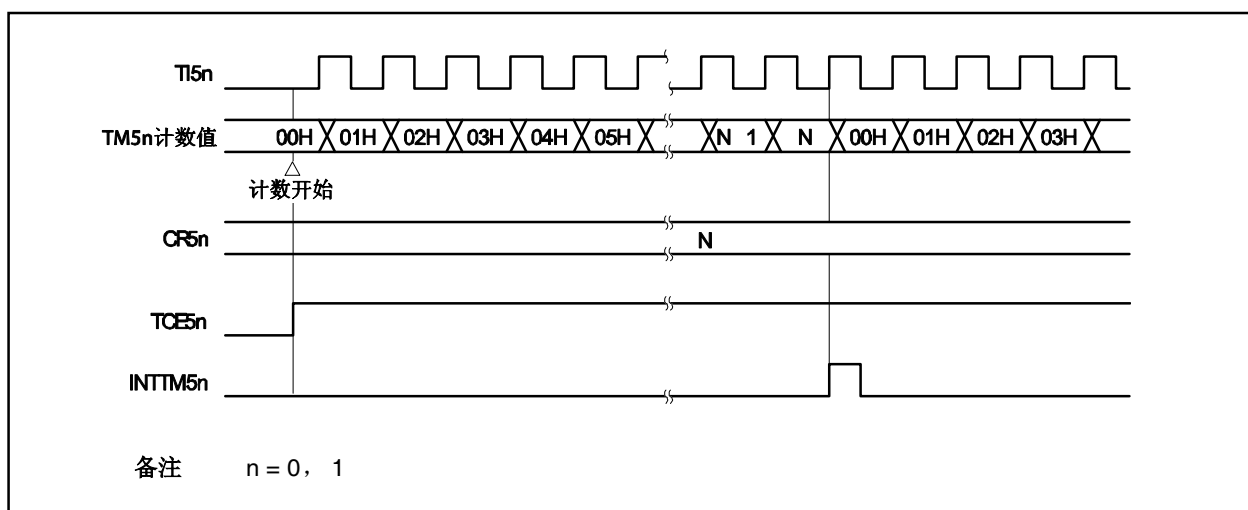
<4> 然后，每次 TM5n 寄存器和 CR5n 寄存器的值相等时，则产生 INTTM5n 信号。

当 TI5n 引脚的有效沿是输入 N+1 次 (N = 00H ~ FFH)，则产生 INTTM5n 信号

注意事项 当外部事件计数器操作期间，不要重写 CR5n 寄存器的值。

备注 n = 0, 1

图 8-3. 事件计数器操作的定时 (带有指定的上升沿)



8.4.3 方波输出操作

任何频率的方波可以以一定间隔输入，间隔由 CR5n 寄存器中预设的值所决定。

通过设置 TCM5n 和 TOE5n 位为 1，TO5n 引脚的输出状态以一定时间间隔倒置，时间间隔由预设的 CR5n 寄存器中的计数值所决定。以这种方式，任何频率的方波可以是输出（负载=50%）（n=0, 1）。

设置方法

<1> 设置每个寄存器。

- TCL5n 寄存器： 选择计数时钟 (t) 。
- CR5n 寄存器： 比较值 (N)
- TMC5n 寄存器： 停止计数操作，选择一种模式，在此模式中，当 TM5n 寄存器和 CR5n 寄存器相等时，则进行清除和启动动作，设置定时器输出的初始值，开始定时器输出 F/F 倒置操作，并且开始定时器输出。
(TMC5n 寄存器 = 00001011B 或 00000111B)
- 对于复用功能引脚设置，当端口引脚用于复用功能时，参考表 4-12 当端口引脚用于复用功能时的设置。

<2> 当 TMC5n.TCE5n 位设置为 1，计数开始。

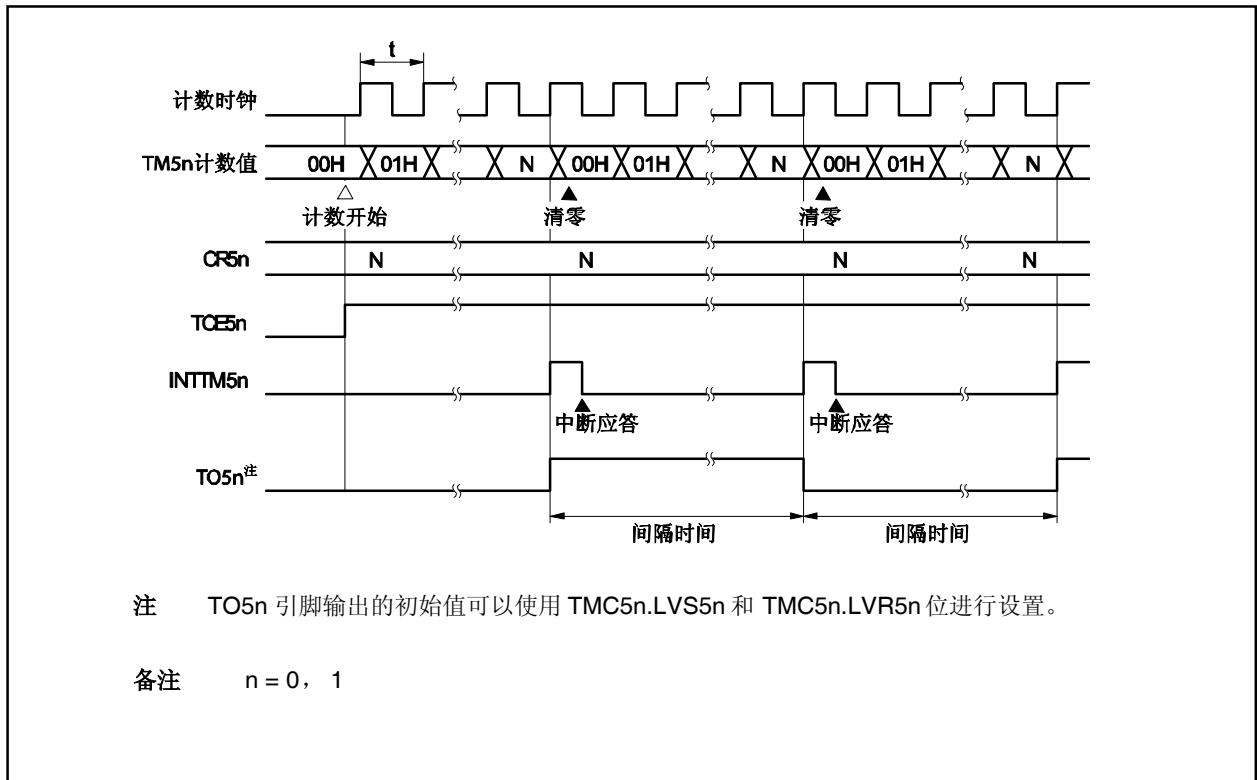
<3> 当 TM5n 寄存器和 CR5n 寄存器值相等时，定时器输出 F/F 倒置。
而且，则生产 INTTM5n 信号并且 TM5n 寄存器清除为 00H。

<4> 然后，在相同的时间间隔内，定时器输出倒置，并且方波从 TO5n 引脚输出。

$$\text{频率} = 1/2t (N + 1) : N = 00H \sim FFH$$

注意事项 在方波输出期间，不要重写 CR5n 寄存器的值。

图 8-4. 方波输出操作的定时



8.4.4 8 位 PWM 输出操作

通过设置 TMC5n.TMC5n6 位为 1，8 位定时器/事件计数器 5n 执行 PWM 输出。

带负载系数的脉冲为 TO5n 引脚的输出，其中负载系数由 CR5n 寄存器中设置的值决定。

设置 CR5n 寄存器中的 PWM 脉冲的有效电平，有效电平可以使用 TMC5n.TMC5n1 位进行选择。

计数时钟可以使用 TCL5n 寄存器进行选择。

PWM 输出可以通过 TMC5n.TOE5n 位进行开启/禁止。

注意事项 CR5n 寄存器重写时间间隔必须为三个或更多操作时钟（由 TCL5n 寄存器进行设置）。

使用方法

<1> 设置每个寄存器。

- TCL5n 寄存器：选择计数时钟（t）。
- CR5n 寄存器：比较值（N）。
- TMC5n 寄存器：停止计数操作，选择 PWM 模式，并且保持定时器输出 F/F 不变，设置有效电平，并且启用定时器输出。

（TMC5n 寄存器= 01000001B 或 01000011B）

- 对于复用功能引脚设置，当端口引脚用于复用功能时，参考表 4-12 当端口引脚用于复用功能时的设置。

<2> 当 TMC5n.TCE5n 位设置为 1 时，计数开始。

PWM 输出操作

<1> 当计数开始时，PWM 输出（从 TO5n 引脚输出）输出非有效电平直到发生溢出。

<2> 当发生溢出时，则输出设置方法<1>设置的有效电平，有效电平输出直到 CR5n 寄存器的值与 TM5n 寄存器的计数值相等，产生一个中断请求信号（INTTM5n）。

<3> 当 CR5n 寄存器的值与 TM5n 寄存器的计数值相等，非有效电平输出并且继续输出直到再次发生溢出。

<4> 然后，重复步骤<2>和<3>直到计数停止。

<5> 当通过清除 TCE5Nw 位为 0 停止计数时，PWM 输出变为非激活。

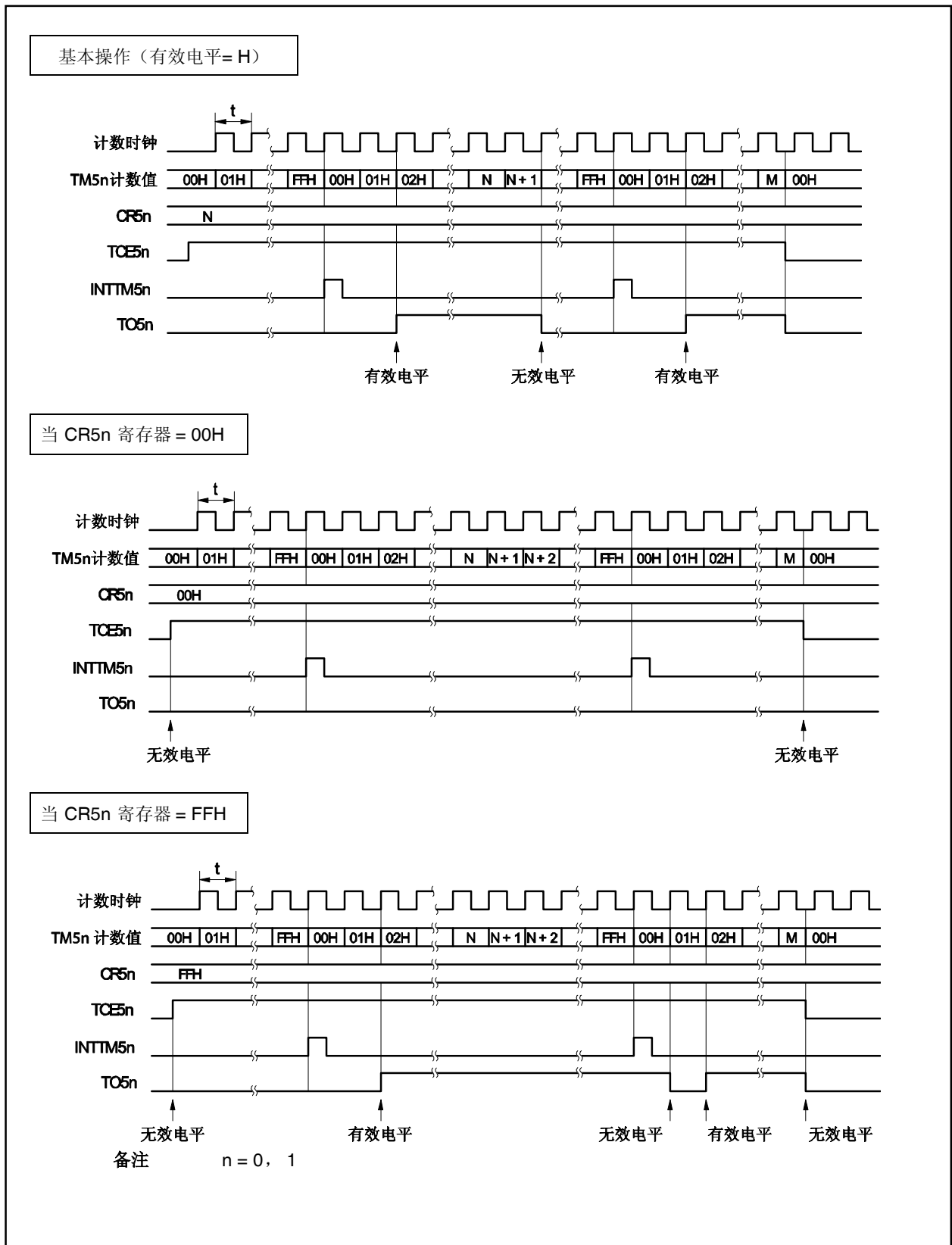
周期 = 256t，有效电平宽度 = Nt，负载 = N/256：N = 00H ~ FFH

备注 1. n = 0, 1

2. 关于详细的定时，参考图 8-5 PWM 输出操作时序和图 8-6 基于 CR5n 寄存器转换的操作时序。

(a) PWM 输出的基本操作

图 8-5. PWM 输出操作的定时

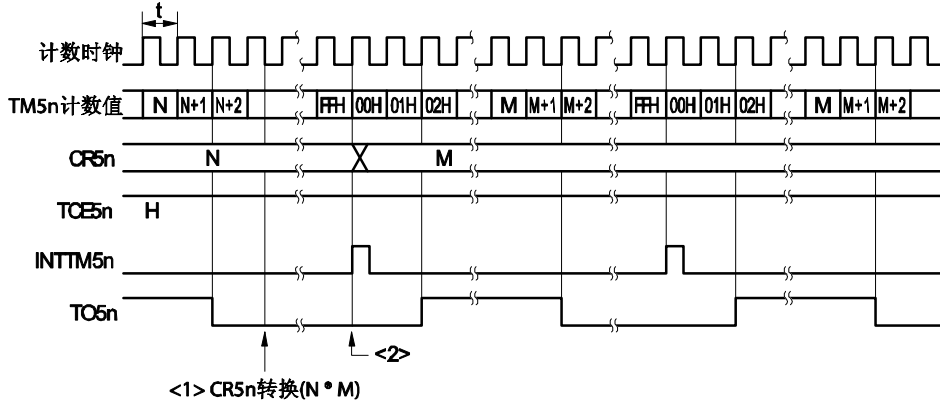


(b) 基于 CR5n 寄存器转换的操作

图 8-6. 基于 CR5n 寄存器转换的操作定时

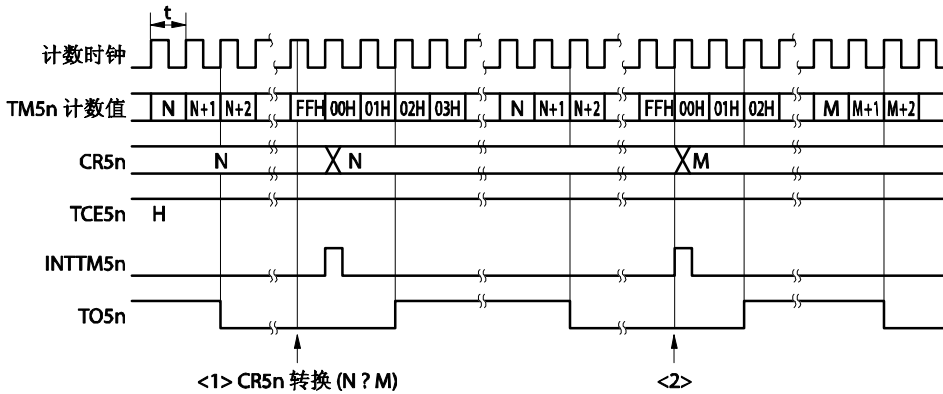
在 FFH 时钟的上升沿之间 CR5n 寄存器的值由 N 变为 M

→在发生溢出之后，CR5n 寄存器的值立即进行传输。



在 FFH 时钟的上升沿之后，当 CR5n 寄存器的值从 N 变为 M 时。

→在第二次溢出时，CR5n 寄存器的值被传输。



注意事项 在步骤<1>和<2>之间，当从 CR5n 寄存器重加载时，实际上使用的值不同（已读值：M；CR5n 寄存器的真实值：N）。

备注 n = 0, 1

8.4.5 作为间隔定时器运行（16 位）

通过设置 TMC51.TMC514 位为 1 可以选择 16 位分辨率定时器/事件计数器模式。

通过使用 16 位定时器比较寄存器 5（CR5）预设的计数值为时间间隔，可以重复产生中断，从而 8 位定时器/事件计数器 5n 可以作为一个间隔定时器运行。

设置方法

<1> 设置每个寄存器

- TCL50 寄存器: 选择计数时钟 (t)
(在级联连接方式下, TCL51 寄存器不需要进行设置)
- CR50 寄存器: 比较值 (N) ... 低 8 位 (可以设置为 00H ~ FFH)
- CR51 寄存器: 比较值 (N) ... 高 8 位 (可以设置为 00H ~ FFH)
- TMC50, TMC51 寄存器: 选择一个模式, 在此模式中, 当 TM5 寄存器和 CR5 寄存器 (x: 忽略) 相等时清除&启动动作不发生。

$$\left. \begin{array}{l} \text{TMC50 寄存器} = 0000\text{xx}00\text{B} \\ \text{TMC51 寄存器} = 0001\text{xx}00\text{B} \end{array} \right\}$$

<2> 设置 TMC51.TCE51 位为 1, 然后设置 TMC50.TCE50 位为 1 以启动计数操作。

<3> 当以级联方式连接的 TM5 寄存器和 CR5 寄存器的值相等时, 则产生 INTTM50 信号 (TM5 寄存器清除为 0000H)。

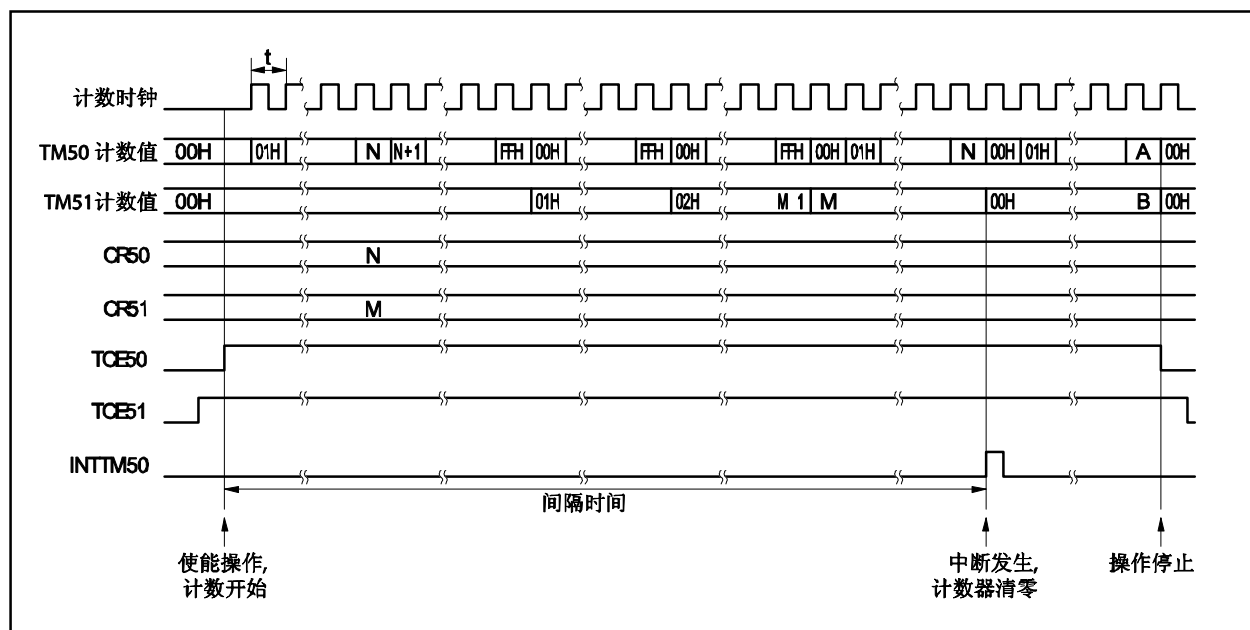
<4> 然后以相同的时间间隔重复产生 INTTM50 信号。

$$\text{间隔时间} = (N + 1) \times t; N = 0000\text{H} \sim \text{FFFFH}$$

- 注意事项**
1. 为了在级联连接中使用 8 位访问进行写入, 在操作开始时设置 TCE51 位为 1, 然后设置 TCE50 位为 1。当操作停止时, 清除 TCE50 位为 0, 然后清除 TCE51 位为 0。
 2. 在级联连接中, TI50 输入, TO50 输出, 并且使用 INTTM50 信号。不要使用 TI51 输出, TO51 输出和 INTTM51 信号, 而是将他们掩膜 (关于更多细节, 敬请参阅第 17 章 中断/异常处理功能)。将 LVS51, LVR51, TMC511, 和 TOE51 位清零。
 3. 在定时器操作中, 不要改变 CR5 寄存器的值。

图 8-7 显示了具有 16 位分辨率的级联连接模式的一个定时示例。

图 8-7. 带 16 位分辨率的级联连接模式



8.4.6 作为外部事件计数器运行（16 位）

通过设置 TMC51.TMC514 位为 1，可以选择 16 位分辨率定时器/事件计数器模式。

使用 16 位定时器计数器 5（TM5），外部事件计数器可以对从一个外部源到 TI50 引脚的时钟脉冲输入计数。

设置方法

<1> 设置每个寄存器。

- TCL50 寄存器：选择 TI50 引脚输出边。
（在级联连接中，不必设置 TCL51 寄存器）。
TI50 引脚的下降沿 → TCL50 寄存器 = 00H
TI50 引脚的上升沿 → TCL50 寄存器 = 01H
- CR50 寄存器：比较值（N）... 低 8 位（可以设置为 00H ~ FFH）
- CR51 寄存器：比较值（N）... 高 8 位（可以设置为 00H ~ FFH）
- TMC50, TMC51 寄存器：停止计数操作，当 TM5 寄存器和 CR5 寄存器之间相等时，选择输入的清除&停止模式，禁止定时器输出 F/F 倒置，并且禁止定时器输出。
（x：忽略）

$$\left(\begin{array}{l} \text{TMC50 寄存器} = 0000\text{xx}00\text{B} \\ \text{TMC51 寄存器} = 0001\text{xx}00\text{B} \end{array} \right)$$

- 对于复用功能引脚设置，当端口引脚用于复用功能时，参考表 4-12 当端口引脚用于复用功能时的设置。

<2> 设置 TMC51.TCE51 位为 1，然后设置 TMC50.TCE50 位为 1，并且对来自 TI50 引脚的脉冲输入计数。

<3> 当以级联连接模式连接的 TM5 和 CR5 寄存器相等时，则产生 INITTM50 信号（TM5 寄存器清除为 0000H）。

<4> 每次 TM5 和 CR5 寄存器的值相等时，则产生 INTTM50 信号。

当 TI50 引脚是输出 N+1 倍：N = 0000H ~ FFFFH 时，则产生 INTTM50 信号

- 注意事项**
1. 在外部事件计数器操作期间，不要重写 CR5n 寄存器的值。
 2. 在级联连接模式中，要使用 8 位访问进行写入，设置 TCE51 位为 1，然后设置 TCE50 位为 1。当操作停止时，清除 TCE50 位为 0，然后清除 TCE51 位为 0（n = 0, 1）。
 3. 在级联连接模式中，使用 TI50 输出和 INTTM50 信号。不要使用 TI51 输入，TO51 输出和 INTTM51 信号，而是将他们掩膜（关于更多细节，敬请参阅第 17 章 中断/异常处理功能）。将 LVS51, LVR51, TMC511, 和 TOE51 位清除为 0。
 4. 在外部事件计数器操作期间，不要改变 CR5 寄存器的值。

8.4.7 方波输出操作（16 位分辨率）

通过设置 TMC51.TMC514 位为 1，可以选择 16 位分辨率定时器/事件计数器模式。

使用 16 位定时器比较寄存器 5（CR5）预设的时间间隔，8 位定时器/事件计数器 5n 可以输出任意频率的方波。

设置方法

<1> 设置每个寄存器。

- TCL50 寄存器： 选择计数时钟（t）
（在级联连接模式中不必设置 TCL51 寄存器）
- CR50 寄存器： 比较值（N）... 低 8 位（可以设置为 00H~FFH）
- CR51 寄存器： 比较值（N）... 高 8 位（可以设置为 00H~FFH）
- TMC50， TCM51 寄存器： 停止计数操作，选择一种模式，在此模式中，当 TM5 寄存器和 CR5 寄存器相等时发生清除和启动操作。

LVS50	LVR50	定时器输出 F/F 状态设置
1	0	高电平输出
0	1	低电平输出

启用定时器输出 F/F 倒置并且启用定时器输出。

$\left[\begin{array}{l} \text{TMC50 寄存器} = 00001011\text{B 或 } 00000111\text{B} \\ \text{TMC51 寄存器} = 00010000\text{B} \end{array} \right]$

- 关于复用功能引脚设置，当端口引脚用于复用功能时，参考表 4-12 当端口引脚用于复用功能时的设置。

<2> 设置 TMC51.TCE51 位为 1，然后设置 TMC50.TCE50 位为 1 以启动计数操作。

<3> 当以级联方式连接的 TM5 寄存器和 CR5 寄存器的值相等时，TO50 定时器输出 F/F 倒置。而且，产生 INTTM50 信号，TM5 寄存器清除为 0000H。

<4> 然后，在相同的时间间隔内，定时器输出 F/F 倒置，并且从 TO50 引脚输出方波。

$$\text{频率} = 1/2t (N + 1) : N = 0000\text{H} \sim \text{FFFFH}$$

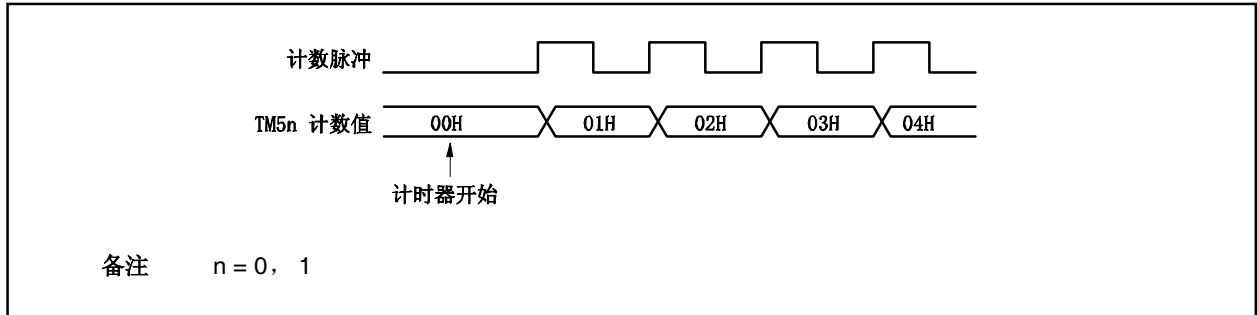
注意事项 在操作期间，不要将不同的值写入到 CR5 寄存器中。

8.4.8 注意事项

(1) 启动定时器错误

在定时器启动之后且在产生相等信号之间，1 个时钟发生错误。这是因为 TM5n 寄存器与计数脉冲没有同步启动。

图 8-8. TM5n 寄存器的计数启动定时



第九章 8 位定时器 H

在 V850ES/KE2 中，提供两个 8 位定时器 H 的通道。

9.1 功能

8 位定时器 Hn 具有以下几种功能（n = 0, 1）。

- 间隔定时器
- 方波输出
- PWM 输出
- 载波发生器

9.2 配置

8 位定时器 Hn 具有以下硬件。

表 9-1. 8 位定时器 Hn 的配置

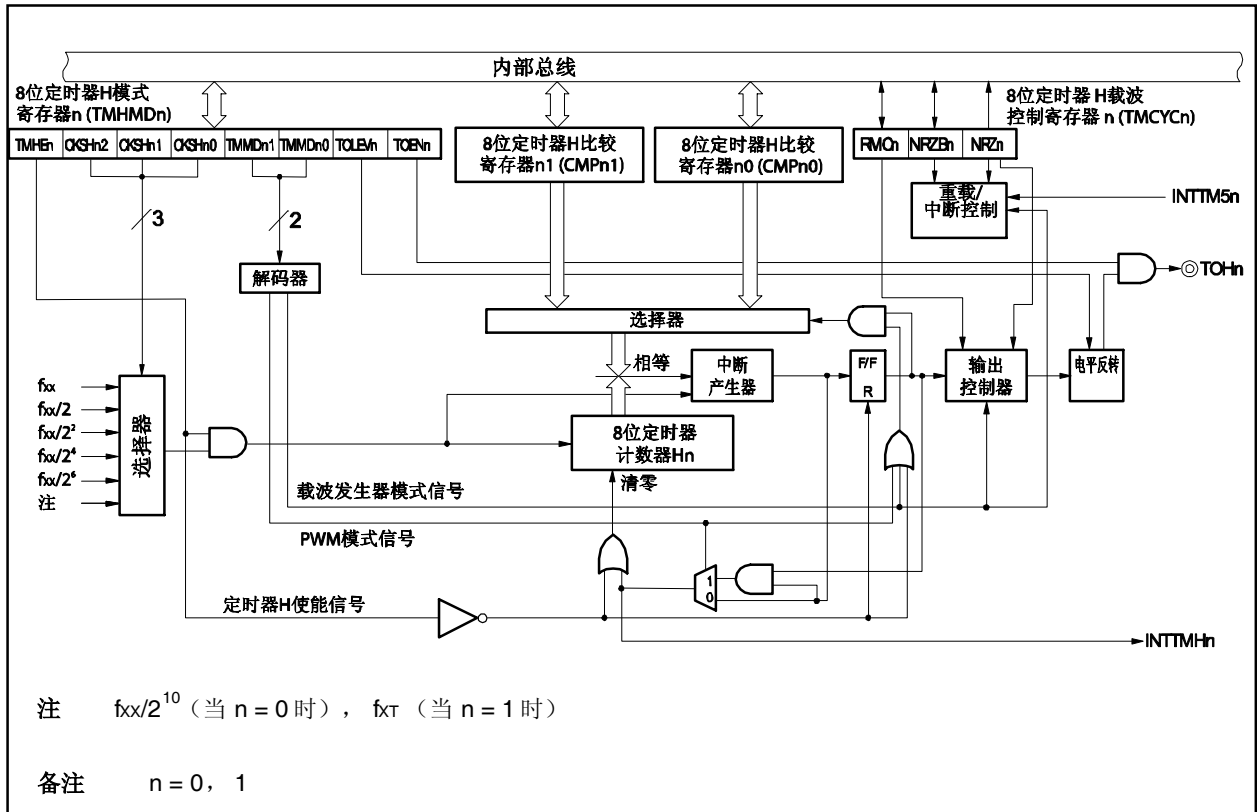
项目	配置
定时器寄存器	8 位定时器计数器 Hn: 1
寄存器	8 位定时器 H 比较寄存器 n0 (CMPn0) : 1 8 位定时器 H 比较寄存器 n1 (CMPn1) : 1
定时器输出	TOHn, 输出控制器
控制寄存器 ^注	8 位定时器 H 模式寄存器 n (TMHMDn) 8 位定时器 H 载波控制寄存器 n (TMCYCn)

注 关于 TOHn 引脚功能的使用方法，敬请参阅表 4-12 当端口引脚用于交替功能时的设置。

备注 n = 0, 1

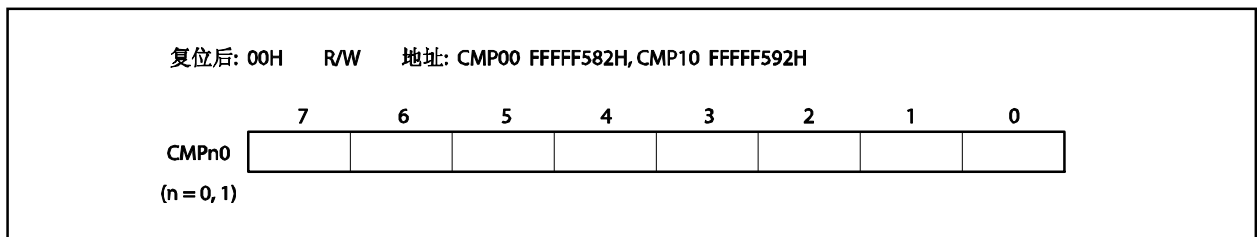
框图如下。

图 9-1. 8 位定时器 Hn 的框图



(1) 8 位定时器 H 比较寄存器 n0 (CMPn0)

该寄存器支持 8 位读写方式。此寄存器可以在定时器的所有运行模式下使用。此寄存器不断将 CMPn0 寄存器中设定的值与 8 位定时器计数器 Hn 的计数值进行比较。当这两个值相等时，将产生一个中断请求信号 (INTTMHn) 并颠倒 TOHn 引脚的输出电平。
 当定时器终止时 (TMHMDn.TMHEn 位 = 0) 要重新写入 CMPn0 寄存器的值。
 该寄存器复位后的值为 00H。



注意事项 在定时器计数操作被禁止期间对 CMPn0 寄存器进行数据重写。

(2) 8 位定时器 H 比较寄存器 n1 (CMPn1)

该寄存器支持 8 位读写方式。

此寄存器可以在 PWM 输出模式和载波发生器模式下使用。

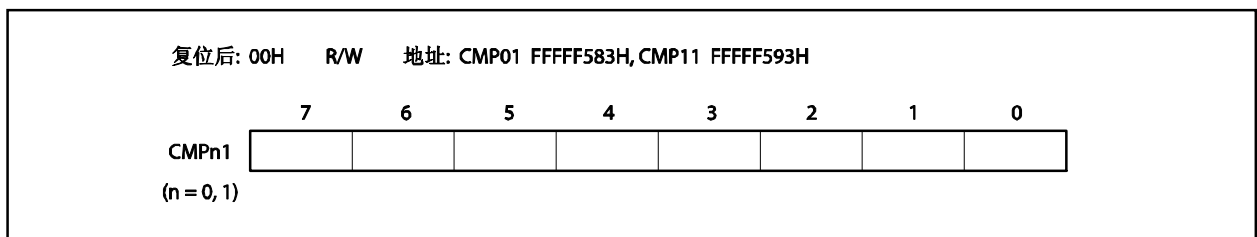
在 PWM 输出模式下，此寄存器可以在定时器的所有运行模式下使用。此寄存器不断将 CMPn1 寄存器中设定的值与 8 位定时器计数器 Hn 的计数值进行比较。当这两个值相等时，颠倒 TOHn 引脚的输出电平。不产生中断请求信号。

在载波发生器模式下，此 CMPn1 寄存器不断将 CMPn1 寄存器中设定的值与 8 位定时器计数器 Hn 的计数值进行比较。当这两个值相等时，产生一个中断请求信号 (INTTMHn)。同时，清除计数值。

在定时器计数执行期间可以对 CMPn1 进行重写。

如果 CMPn1 寄存器的值在定时器运行期间进行了重写，则当定时器的计数值与 CMPn1 寄存器中原有的值相等时，此重写的新值将被锁存起来并传输给 CMPn1 寄存器，然后 CMPn1 寄存器的值就变成为此新值。如果计数值和 CMPn1 寄存器的值相等并且写入的值与 CMPn1 寄存器发生了冲突，则 CMPn1 的值并没有被改变。

该寄存器复位后的值为 00H。



可以在定时器计数期间对 CMPn1 进行重写。

在载波发生器模式下，对 CMPn1 寄存器设置之后，如果 8 位定时器计数器 Hn 的计数值和 CMPn1 寄存器中设定的值相等，则产生一个中断请求信号 (INTTMHn)。同时，8 位定时器计数器的值被清为 00H。

如果在定时器运行期间对 CMPn1 寄存器中设定的值进行了重写，则重新加载时间选择在 8 位定时器计数器 Hn 的计数值和 CMPn1 寄存器中设定的值相等的时候。如果传输时间和 CPU 中寄存器发送冲突，则不进行传输。

注意事项 在 PWM 输出模式和载波发生器模式下，定时器计数操作被终止之后 (TMHEn 位 = 0)，当启动定时器计数操作 (TMHMDn.TMHEn 位 = 1) 时要对 CMPn1 寄存器进行设置。(即使对 CMPn1 寄存器设定的是相同的值也要再次设置)。

9.3 寄存器

控制 8 位定时器 Hn 的寄存器如下:

- 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 n (TMCYCn)

备注 1. 关于 TOHn 引脚功能的使用方法, 敬请参阅表 4-12 当端口引脚用于交替功能时的设置。
 2. n = 0, 1

- (1) **8 位定时器 H 模式寄存器 n (TMHMDn)**
TMHMDn 寄存器控制 8 位定时器 Hn 的模式。
TMHMDn 寄存器支持 8 位或 1 位读写方式。
该寄存器复位后的值为 00H。

备注 n = 0, 1

(a) 8 位定时器 H 模式寄存器 0 (TMHMD0)

复位后: 00H R/W 地址: FFFF580H

	<7>	6	5	4	3	2	<1>	<0>
TMHMD0	TMHE0	CKSH02	CKSH01	CKSH00	TMMD01	TMMD00	TOLEVO	TOEN0

TMHE0	容许8位定时器H0 操作
0	停止定时器计数操作(8位定时器计数器H0 = 00H)
1	容许定时器计数操作(时钟输入时开始计数)

CKSH02	CKSH01	CKSH00	计数时钟选择			
			计数时钟 ^注	$f_{\text{CK}} = 20 \text{ MHz}$	$f_{\text{CK}} = 16.0 \text{ MHz}$	$f_{\text{CK}} = 10.0 \text{ MHz}$
0	0	0	f_{CK}	禁止设置	禁止设置	100 ns
0	0	1	$f_{\text{CK}}/2$	100 ns	125 ns	200 ns
0	1	0	$f_{\text{CK}}/4$	200 ns	250 ns	400 ns
0	1	1	$f_{\text{CK}}/16$	800 ns	1 μs	1.6 μs
1	0	0	$f_{\text{CK}}/64$	3.2 μs	4 μs	6.4 μs
1	0	1	$f_{\text{CK}}/1024$	51.2 μs	64 μs	102.4 μs
以上除外			禁止设置			

TMMD01	TMMD00	8位定时器H0操作模式
0	0	间隔定时器模式
0	1	载波生成器模式
1	0	PWM 输出模式禁止设置
1	1	禁止设置

TOLEVO	定时器输出水平控制(缺省)
0	低电平
1	高电平

TOEN0	定时器输出控制
0	禁止输出
1	容许输出

注 设置时满足以下条件。

$V_{\text{DD}} = 4.0 \sim 5.5 \text{ V}$: 计数时钟频率 $\leq 10 \text{ MHz}$

$V_{\text{DD}} = 2.7 \sim 4.0 \text{ V}$: 计数时钟频率 $\leq 5 \text{ MHz}$

- 注意事项
1. 当 TMHE0 位=1 时, 设置位 (除了 TMHMD0 寄存器中的设置位) 被禁止。
 2. 在 PWM 输出模式下, 定时器计数操作被终止 之后 (TMHE0 位 = 0), 当启动定时器计数操作 (TMHE0 位 = 1) 时要对 CMP01 寄存器进行设置。(即使对 CMP01 寄存器设定的是相同的值也要再次设置)。
 3. 当使用的是载波发生器模式, 则要将 8 位定时器 H0 计数时钟频率设置为 8 位定时器/事件计数器 50 计数时钟频率的 6 倍或更高。

(b) 8 位定时器 H 模式寄存器 1 (TMHMD1)

复位后: 00H R/W 地址: FFFF590H

	<7>	6	5	4	3	2	<1>	<0>
TMHMD1	TMHE1	CKSH12	CKSH11	CKSH10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	容许8位定时器 H1 操作
0	停止定时器计数操作 (8位定时器计数器H1 = 00H)
1	容许定时器计数操作 (当时钟输入时开始计数)

CKSH12	CKSH11	CKSH10	计数时钟选择			
			计数时钟 ^注	f _{CK} = 20.0 MHz	f _{CK} = 16.0 MHz	f _{CK} = 10.0 MHz
0	0	0	f _{CK}	禁止设置	禁止设置	100 ns
0	0	1	f _{CK} /2	100 ns	125 ns	200 ns
0	1	0	f _{CK} /4	200 ns	250 ns	400 ns
0	1	1	f _{CK} /16	800 ns	1 ns	1.6 ns
1	0	0	f _{CK} /64	3.2 ns	4 ns	6.4 ns
1	0	1	f _{CK} (副时钟)			
以上除外			禁止设置			

TMMD11	TMMD10	8位定时器H1操作模式
0	0	间隔定时器模式
0	1	载波生成器模式
1	0	PWM 输出模式
1	1	禁止设置

TOLEV1	定时器输出水平控制 (缺省)
0	低电平
1	高电平

TOEN1	定时器输出控制
0	禁止输出
1	容许输出

注 设置时满足以下条件。

V_{DD} = 4.0 ~ 5.5 V: 计数时钟频率 ≤ 10 MHz

V_{DD} = 2.7 ~ 4.0 V: 计数时钟频率 ≤ 5 MHz

- 注意事项
1. 当 TMHE1 位=1 时, 设置位 (除了 TMHMD1 寄存器中的设置位) 被禁止。
 2. 在 PWM 输出模式和载波发生器下, 定时器计数操作被终止 之后 (TMHE1 位 = 0), 当启动定时器计数操作 (TMHE1 位 = 1) 时要对 CMP11 寄存器进行设置。(即使对 CMP11 寄存器设定的是相同的值也要再次设置)。
 3. 当使用的是载波发生器模式, 则要将 8 位定时器 H1 计数时钟频率设置为 8 位定时器/事件计数器 51 计数时钟频率的 6 倍或更高。

(2) 8 位定时器 H 载波控制寄存器 n (TMCYCn)

此寄存器控制 8 位定时器 Hn 远程控制输出和输出状态的载波脉冲。

TMCYCn 寄存器支持 8 位或 1 位读写方式。NRZn 位是只读位。

TMCYCn 寄存器复位后的值为 00H。

备注 n = 0, 1

复位后: 00H		R/W	地址: TMCYC0 FFFF581H, TMCYC1 FFFF591H							
			7	6	5	4	3	2	1	<0>
TMCYCn			0	0	0	0	0	RMCn	NRZBn	NRZn
(n = 0, 1)										
	RMCn	NRZBn	远程控制输出							
	0	0	低电平输出							
	0	1	高电平输出							
	1	0	低电平输出							
	1	1	载波脉冲输出							
	NRZn	运载脉冲输出状态标志								
	0	禁止载波输出状态 (低电平状态)								
	1	容许载波输出状态								

9.4 运行

9.4.1 作为间隔定时器/方波输出

当 8 位定时器计数器 Hn 的计数值与 CMPn0 寄存器中设定的值相等时，将产生一个中断请求信号 (INTTMHn) 并将 8 位定时器计数器 Hn 清为 00H。

CMPn1 不能在间隔定时器模式下使用。即使对 CMPn1 寄存器进行了设置，也不会对定时器的输出造成影响，因为 8 位定时器计数器 Hn 和 CMPn1 寄存器的相等情况不能探测出来。

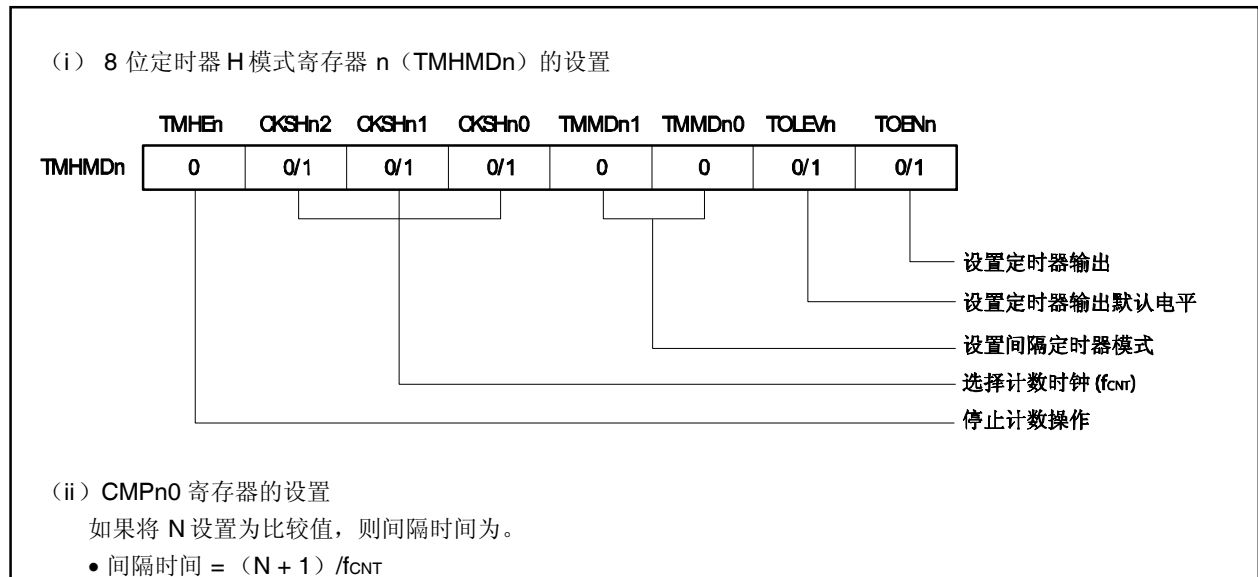
通过将 TMHMDn.TOENn 设置为 1，可以使所需频率（负载 = 50%）的方波从 TOHn 引脚中输出。

- 备注**
1. 关于交替功能引脚 (TOHn) 的设置，参见表 4-12 当端口引脚用于交替功能时的设置
 2. 关于 INTTMHn 的使能中断，请参见第十七章 中断/异常处理功能。

设置

<1> 设置每个寄存器。

图 9-2.在间隔定时器模式下设置寄存器

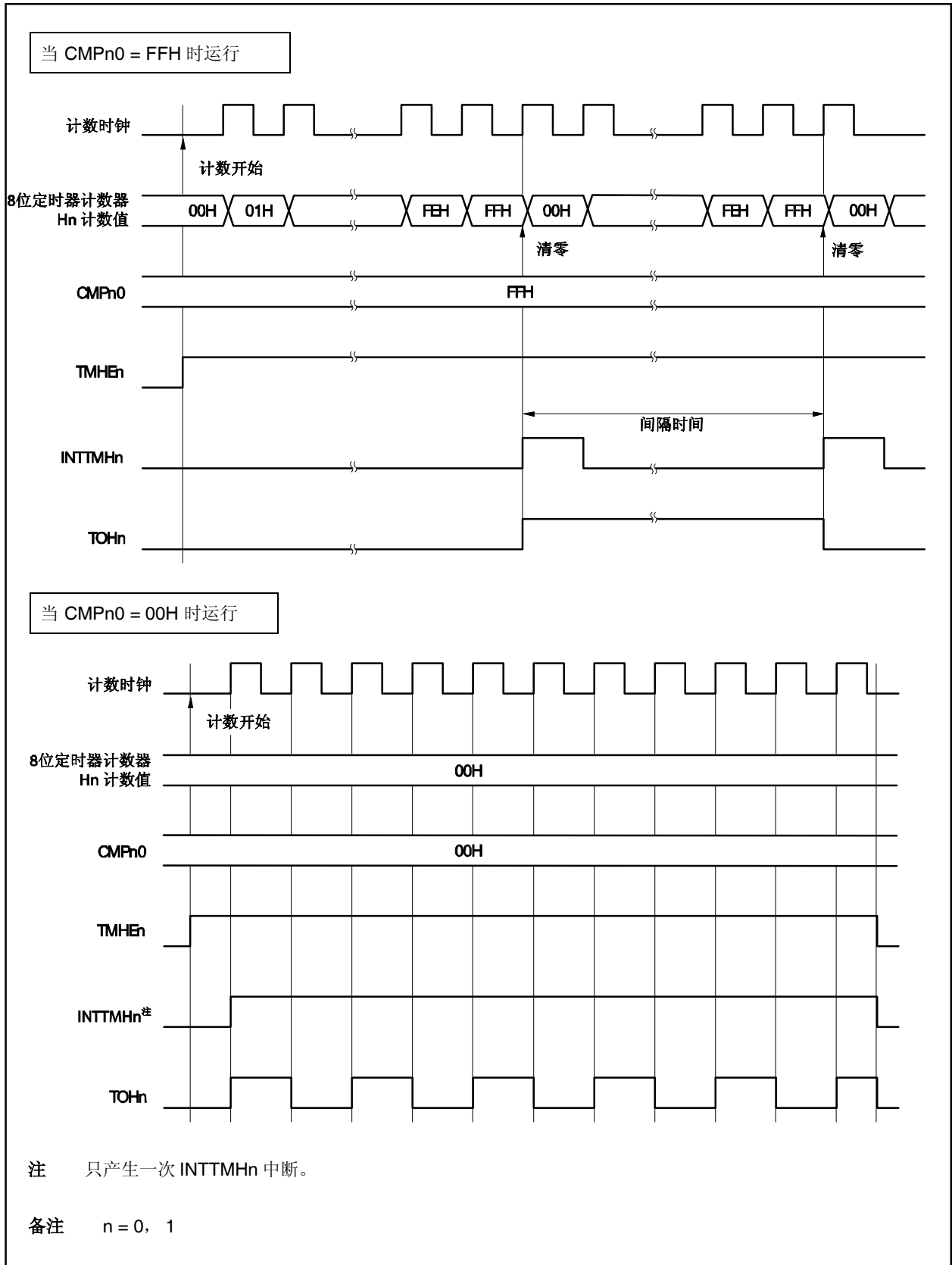


<2> 当 TMHEn 位被设置为 1 时，则启动计数。

<3> 当 8 位定时器计数器 Hn 的计数值和 CMPn0 寄存器中设定的值相等时，将产生 INTTMHn 信号并且 8 位定时器计数器被清为 00H。

<4> 然后在相同的间隔内产生 INTTMHn 信号。将 TMHEn 位清为 0，可以终止计数。

图 9-3. 间隔定时器/方波输出操作的时间选择 (2/2)



9.4.2 PWM 输出模式操作

在 PWM 输出模式下，可以输出任何占空比和周期的脉冲。

CMPn0 寄存器控制定时器输出 (TOHn) 周期。在定时器运行期间不能对 CMPn0 寄存器进行重写。

CMPn1 寄存器控制定时器输出 (TOHn) 占空比。CMPn1 寄存器可以在定时器运行期间进行重写。

在 PWM 输出模式中的操作如下。

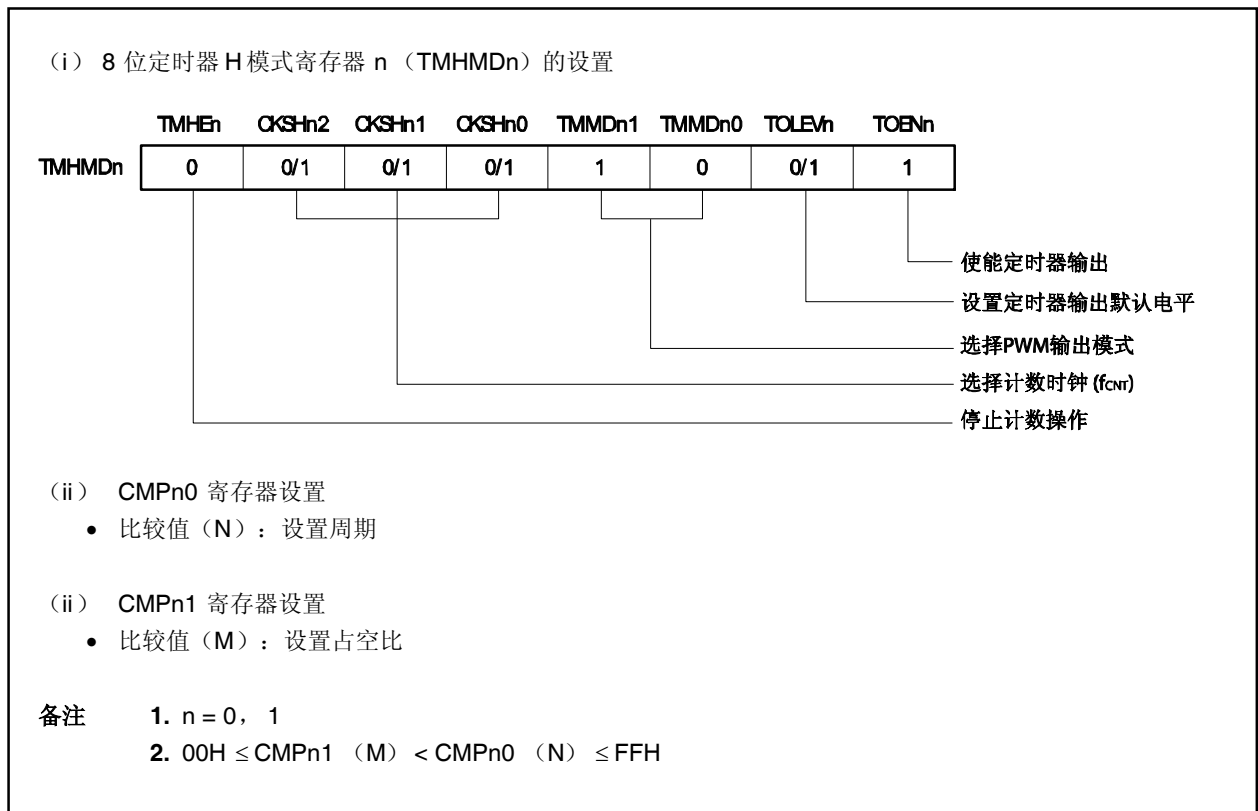
启动定时器计数后，当 8 位定时器计数器 Hn 的计数值和 CMPn0 寄存器中设定的值相等时，颠倒 TOHn 输出电平并将 8 位定时器计数器 Hn 清为 00H。当 8 位定时器计数器的计数值和 CMPn1 寄存器中设定的值相等时，颠倒 TOHn 输出电平。

- 备注**
1. 关于交替功能引脚 (TOHn) 的设置，请参见表 4-12 当端口引脚用于交替功能时的设置。
 2. 关于 INTTMHn 的使能中断，请参见第十七章 中断/异常处理功能。

设置

<1> 设置每个寄存器。

图 9-4. 在 PWM 输出模式下设置寄存器



<2> 当 TMHEn 位被设置为 1 时，启动计数。

- <3> 启动计数操作之后，首先要比较的是 CMPn0 寄存器。当 8 位定时器计数器 Hn 的计数值和 CMPn0 寄存器中设定的值相等时，8 位定时器计数器 Hn 被清除，产生一个中断请求信号（INTTMHn）并颠倒 TOHn 输出电平。同时，与 8 位定时器计数器 Hn 相比较的寄存器将从 CMPn0 寄存器改变为 CMPn1 寄存器。
- <4> 当 8 位定时器计数器 Hn 的计数值和 CMPn1 寄存器中设定的值相等时，颠倒 TOHn 输出电平。同时，与 8 位定时器计数器 Hn 相比较的寄存器将从 CMPn1 寄存器改变为 CMPn0 寄存器。此时 8 位定时器计数器 Hn 不被清除并且不产生 INTTMHn 信号。
- <5> 重复执行步骤<3> 和<4>可以获取任何负载的脉冲。
- <6> 将 TMHEn 位清为 0，终止计数操作。

指定 CMPn0 寄存器中设定的值为 (N)、CMPn1 中设定的值为 (M)、计数时钟频率为 f_{CNT} 。PWM 脉冲输出周期和占空比如下。

$$\begin{aligned} \text{PWM 脉冲输出周期} &= (N + 1) / f_{CNT} \\ \text{占空比} &= \text{未激活宽度} : \text{激活宽度} = (M + 1) : (N + 1) \end{aligned}$$

- 注意事项**
1. 当定时器计数器正在运行时，可以改变 CMPn1 寄存器中设定的值。但是这至少需要 3 个运行时钟（此信号由 TMHMDn 寄存器的 CKSHn2 到 CKSHn0 位来选择）。直到这个值被传输给寄存器时 CMPn1 寄存器的值才被改变。
 2. 定时器计数操作被终止（TMHEn 位 = 0）之后再启动计数器计数操作时（TMHEn 位 = 1）一定要对寄存器进行设置。（即使对寄存器设置的是相同的值也要再次设置）。
 3. 要确保 CMPn1 寄存器中设定的值 (M) 和 CMPn0 寄存器中设定的值 (N) 在下面这个范围之内。
 $00H \leq \text{CMPn1 (M)} < \text{CMPn0 (N)} \leq FFH$ 。

图 9-5. 在 PWM 输出模式下的运行时间选择 (1/4)

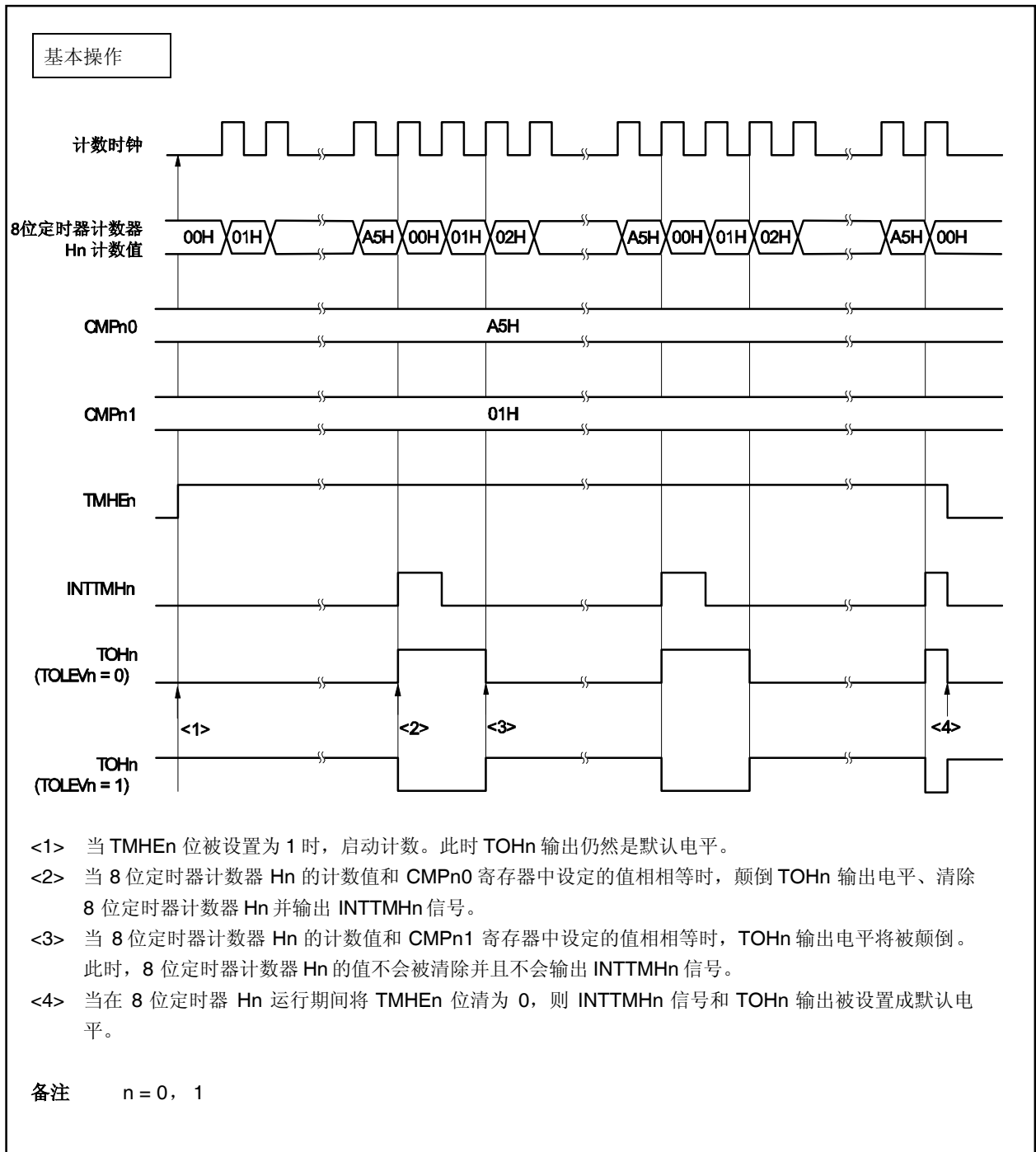


图 9-5. 在 PWM 输出模式下的运行时机选择 (2/4)

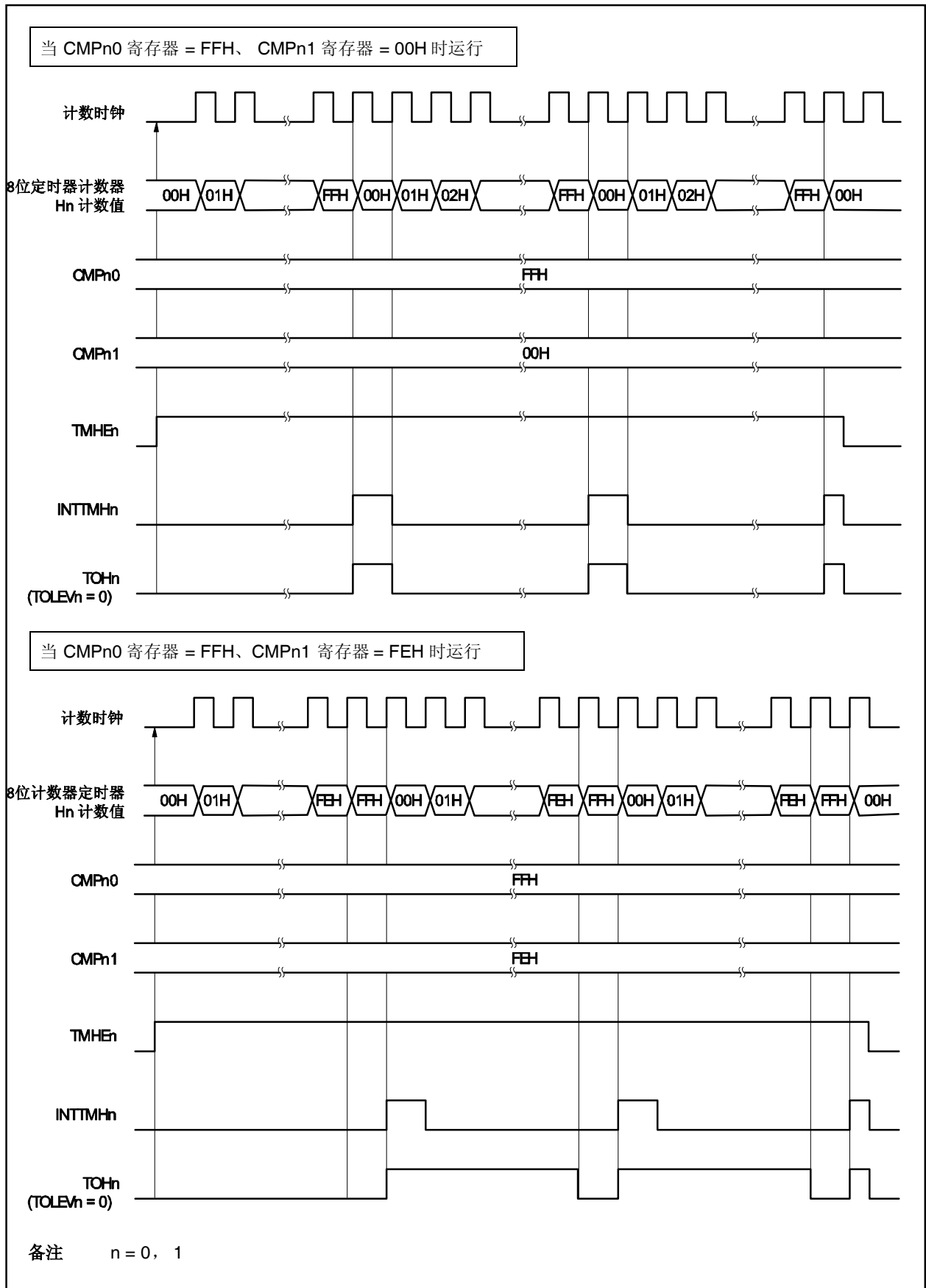


图 9-5. 在 PWM 输出模式下的运行时机选择 (3/4)

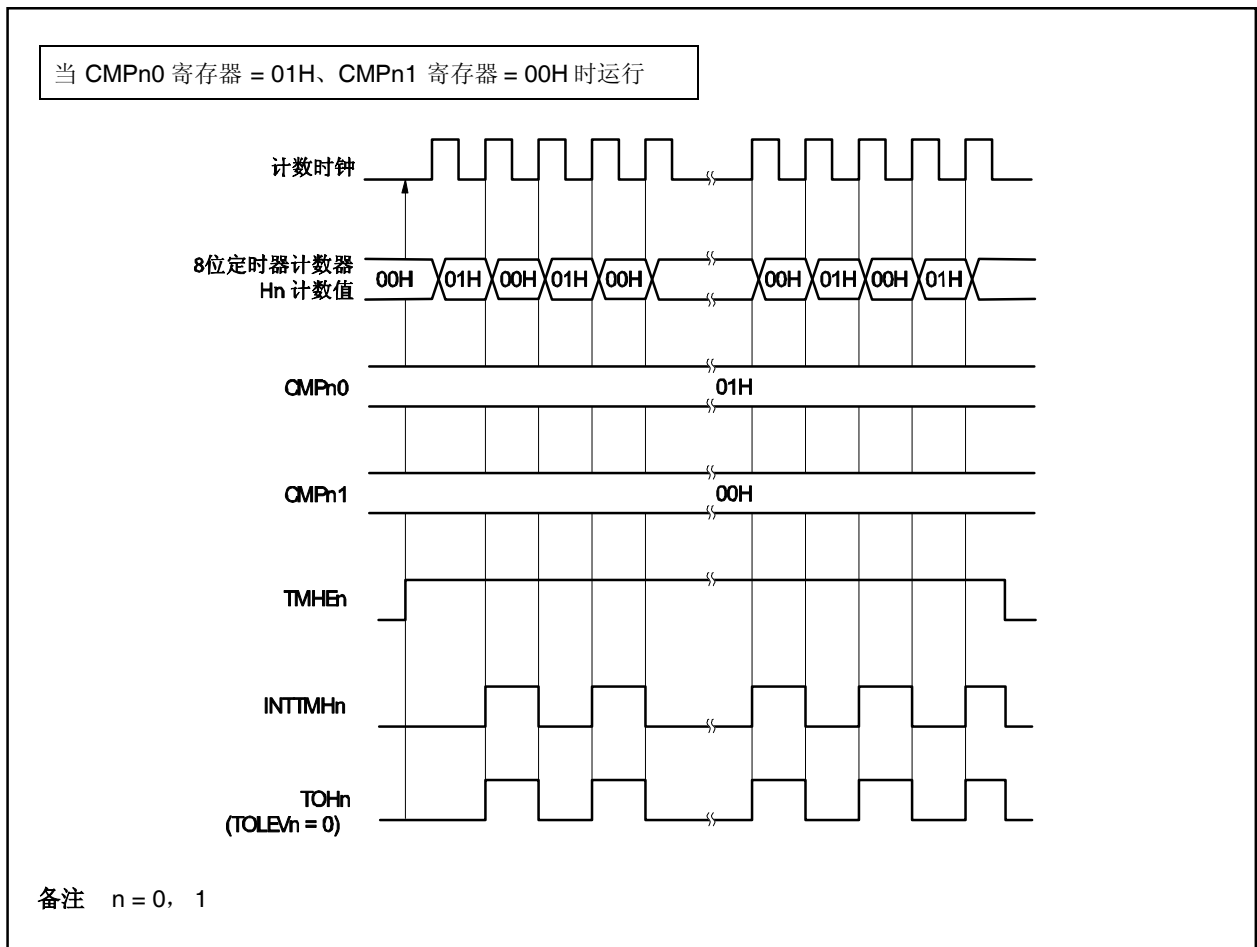
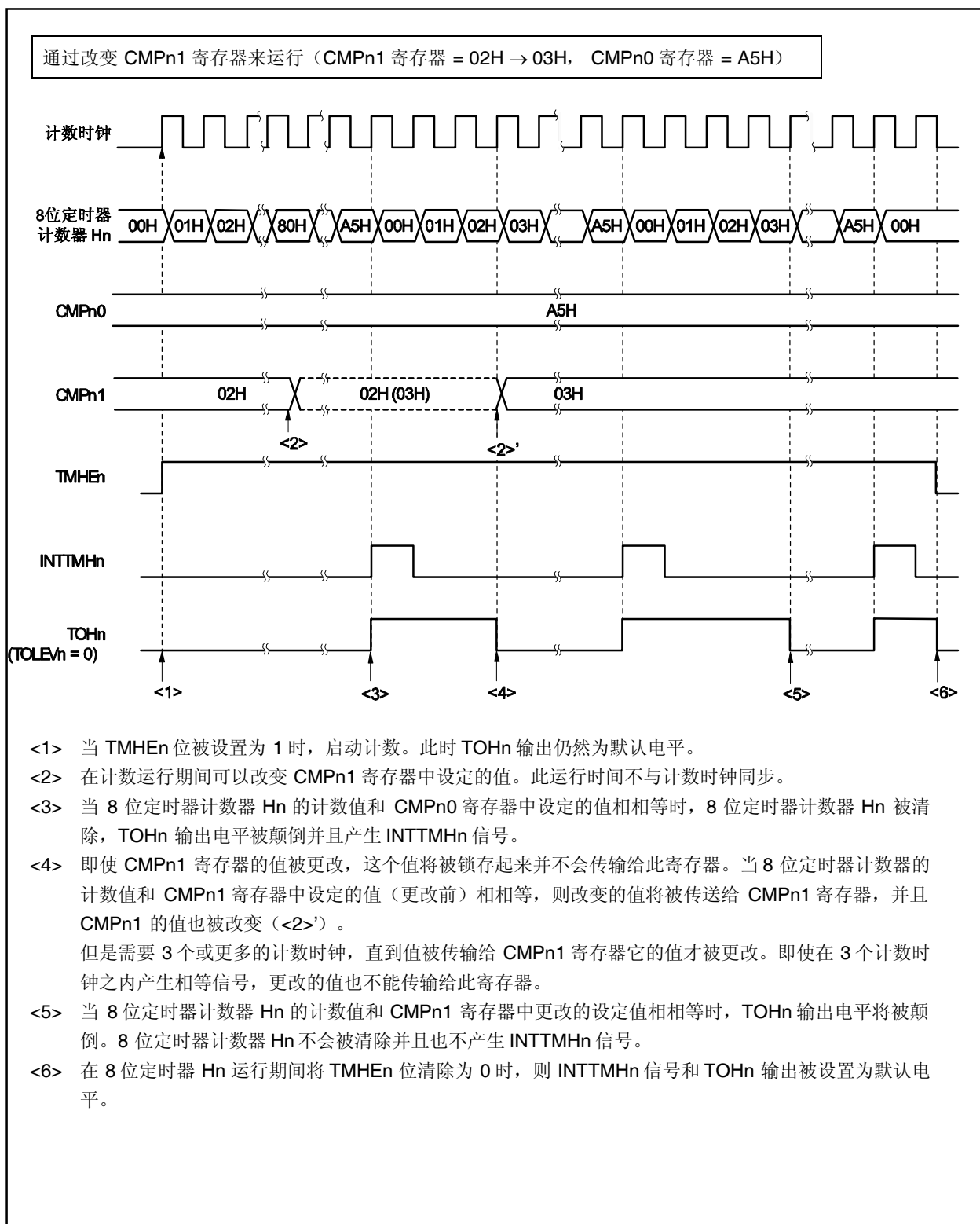


图 9-5. 在 PWM 输出模式下的运行时机选择 (4/4)



9.4.3 载波发生器模式操作

使用循环设置（具有 8 位定时器/事件计数器 5n）将 8 位定时器 Hn 产生的载波时钟输出。

在载波发生器模式下，8 位定时器/事件计数器 5n 被用来控制扩展：输出 8 位定时器 Hn 的载波脉冲，从 TOHn 输出端口中输出载波脉冲。

- 备注**
1. 关于交替功能引脚（TOHn）的设置，请参见表 4-12 当端口引脚用于交替功能时的设置。
 2. 关于 INTTMHn 的使能中断，请参见第十七章 中断/异常处理功能。

(1) 产生载波

在载波发生器模式下，CMPn0 寄存器产生一个具有低电平宽度的载波脉冲波形，CMPn1 寄存器产生一个具有高电平宽度的载波脉冲波形。

在 8 位定时器 Hn 运行期间，可以对 CMPn1 寄存器进行重写，但是不能对 CMPn0 寄存器进行重写。

(2) 载波输出控制

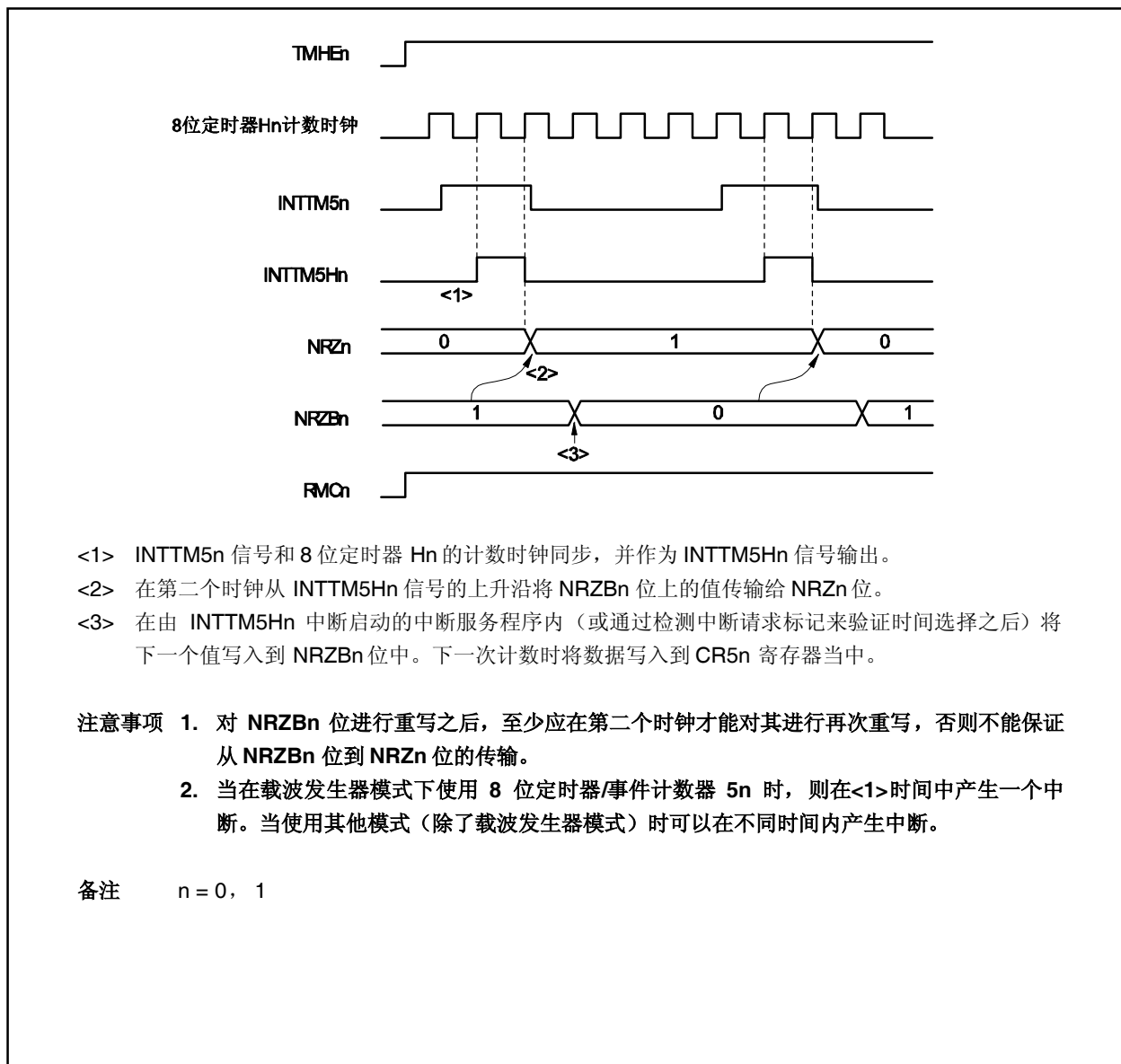
利用 8 位定时器/事件计数器 5n 的中断请求信号（INTTM5n）和 TMCYCn.NRZBn 与 TMCYCn.RMCn 位来进行载波输出控制。输出关系如下。

RMCn 位	NRZBn 位	输出
0	0	低电平输出
0	1	高电平输出
1	0	低电平输出
1	1	载波脉冲输出

备注 n = 0, 1

为了在计数运行期间控制载波脉冲输出，则 $TMCYCn.NRZn$ 和 $TMCYCn.NRZBn$ 位具有一个主次位配置。当 $NRZBn$ 位被读取或写入时 $NRZn$ 位是只读的。 $INTTM5n$ 信号和 8 位定时器 Hn 时钟同步，输出时作为 $INTTM5Hn$ 信号。 $INTTM5Hn$ 信号变为 $NRZn$ 位的数据传输信号， $NRZBn$ 位上的值被传输给 $NRZn$ 位。从 $NRZBn$ 位传输到 $NRZn$ 位的时间选择如下。

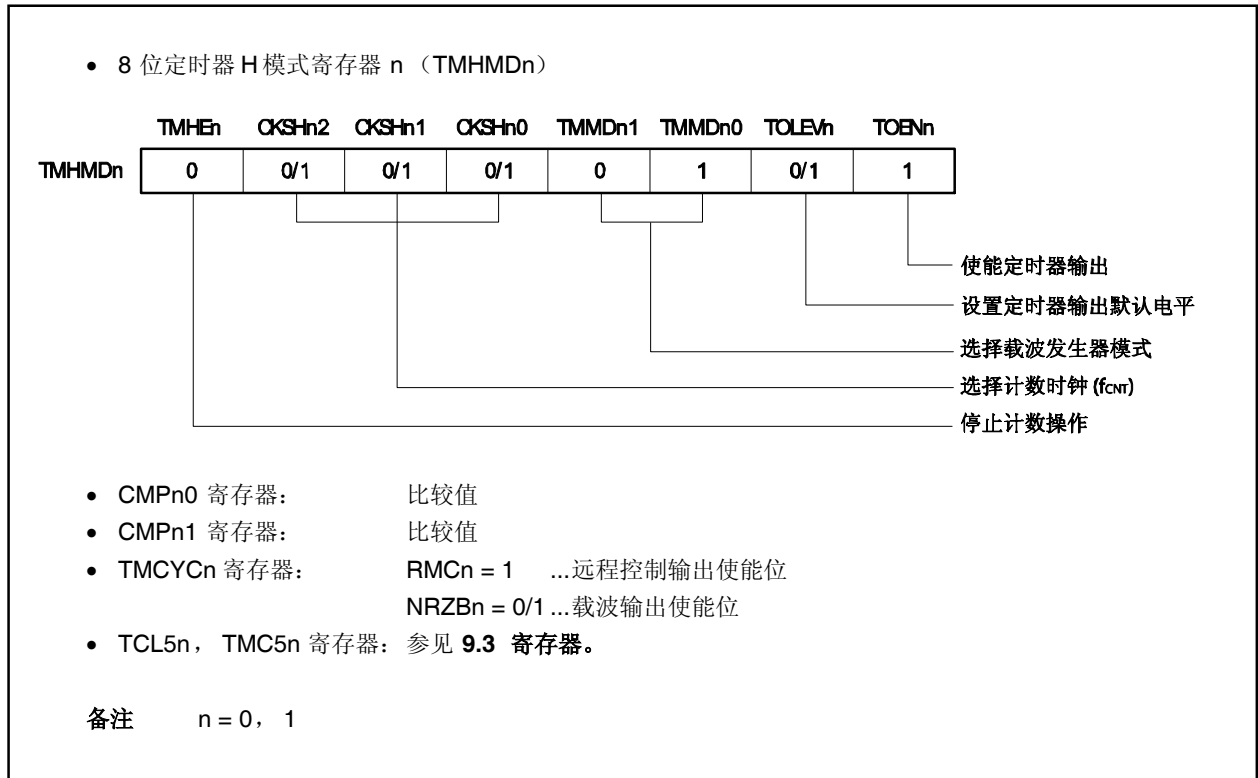
图 9-6. 传输时间选择



设置

<1> 设置每个寄存器。

图 9-7. 在载波发生器模式下设置寄存器



<2> 当 TMHEn 位被设置为 1 时, 将启动 8 位定时器 Hn 计数操作。

<3> 当 TMC5n.TCE5n 位被设置位 1 时, 将启动 8 位定时器/时间计数器 5n 计数操作。

<4> 启动计数操作之后, 首先要比较 CMPn0 寄存器。当 8 位定时器计数器 Hn 的计数值和 CMPn0 寄存器中设定的值相等时, 将产生 INTTMHn 信号并且 8 位定时器计数器 Hn 被清除。同时与 8 位定时器计数器 Hn 相比较的寄存器从 CMPn0 寄存器变为 CMPn1 寄存器。

<5> 当 8 位定时器计数器 Hn 的计数值和 CMPn1 寄存器中设定的值相等时, 将产生 INTTMHn 信号并且 8 位定时器计数器 Hn 被清除。同时与 8 位定时器计数器 Hn 相比较的寄存器从 CMPn1 寄存器变为 CMPn0 寄存器。

<6> 重复执行步骤<4> 和 <5> 获取载波时钟。

<7> INTTM5n 信号和 8 位定时器 Hn 同步并作为 INTTM5Hn 信号输出。此信号变为 NRZBn 位的数据传输信号并且 NRZBn 位上的值被传输给 NRZn 位。

<8> 在由 INTTM5Hn 中断启动的中断服务程序内 (或通过检测中断请求标记来验证时间选择之后) 将下一个值写入到 NRZBn 位中。下一次计数时将数据写入到 CR5n 寄存器当中。

<9> 当 NRZn 位变成高电平时, 载波时钟从 TOHn 引脚输出。

<10> 重复执行上述步骤可以获取任何载波时钟。将 TMHEn 位清为 0 关闭计数操作。

指定 CMPn0 寄存器中设定的值为 (N)、CMPn1 中设定的值为 (M)、计数时钟频率为 f_{CNT} 。载波时钟输出输出周期和负载如下。

$$\begin{aligned} \text{载波时钟输出周期} &= (N + M + 2) / f_{CNT} \\ \text{占空比}=\text{高电平宽度: 载波时钟输出宽度} &= (M + 1) : (N + M + 2) \end{aligned}$$

- 注意事项**
1. 定时器计数操作被终止之后 (TMHEn 位 = 0) 再次启动定时器计数操作时 (TMHEn 位 = 1) 一定要设置 CMPn1 寄存器。(即使对 CMPn1 寄存器设置相同的值也要再次设置)。
 2. 将 CMPn0 和 CMPn1 寄存器的值设置在 01H ~ FFH 范围之内。
 3. 在载波发生器模式下, CMPn1 寄存器被重写之后, 将新值传输给此寄存器需要 3 个操作时钟 (通过 TMHMDn.CKSHn0 ~ TMHMDn.CKSHn2 位选择的信号)。
 4. 启动计数操作之前要对 TMCYCn.RMCn 位进行设置。
 5. 当使用载波发生器模式时, 将 8 位 Hn 计数时钟频率设置为 8 位定时器/事件计数器 5n 时钟频率的 6 倍或更高。

图 9-8. 载波发生器模式 (1/3)

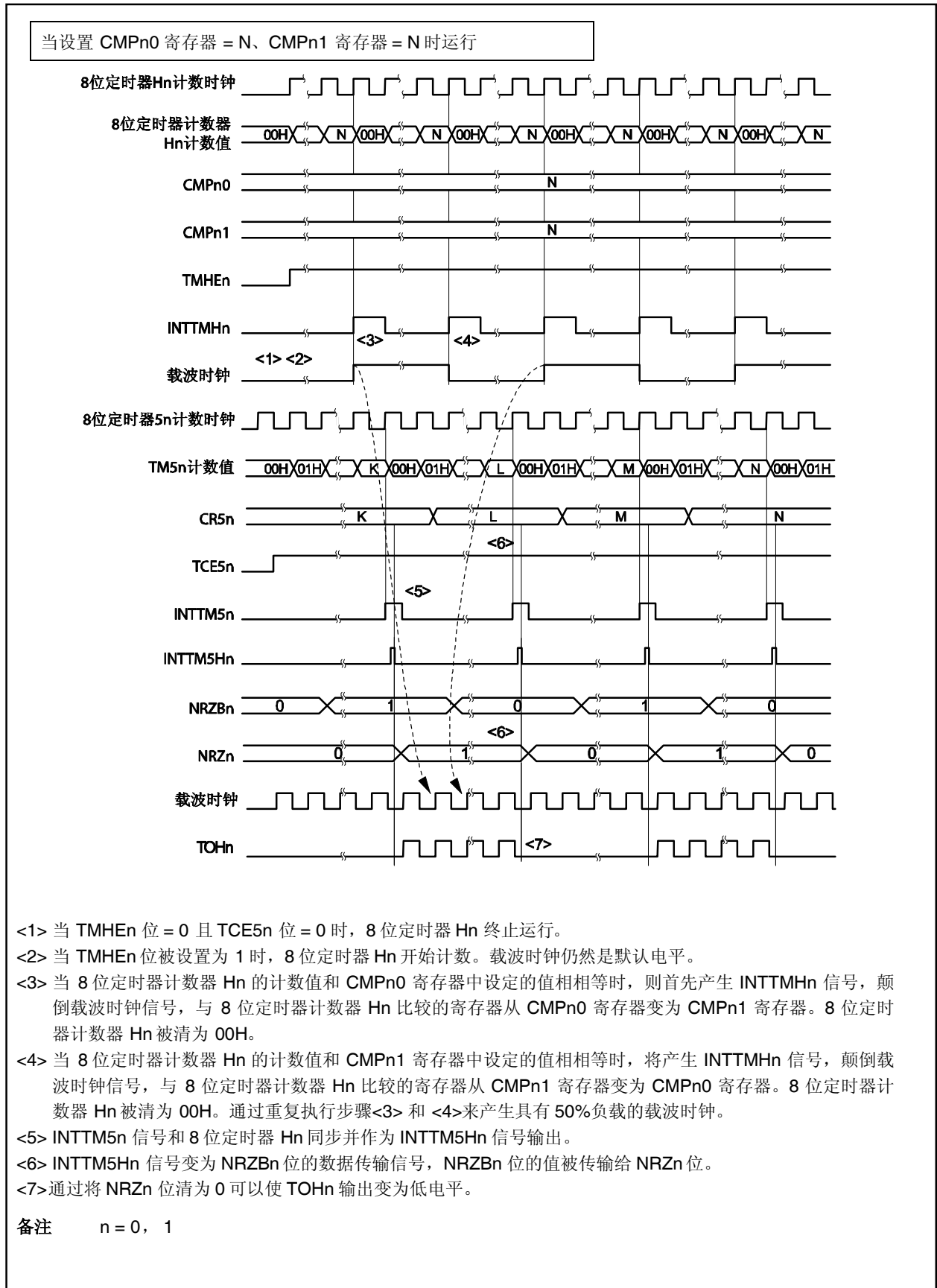


图 9-8. 载波发生器模式 (2/3)

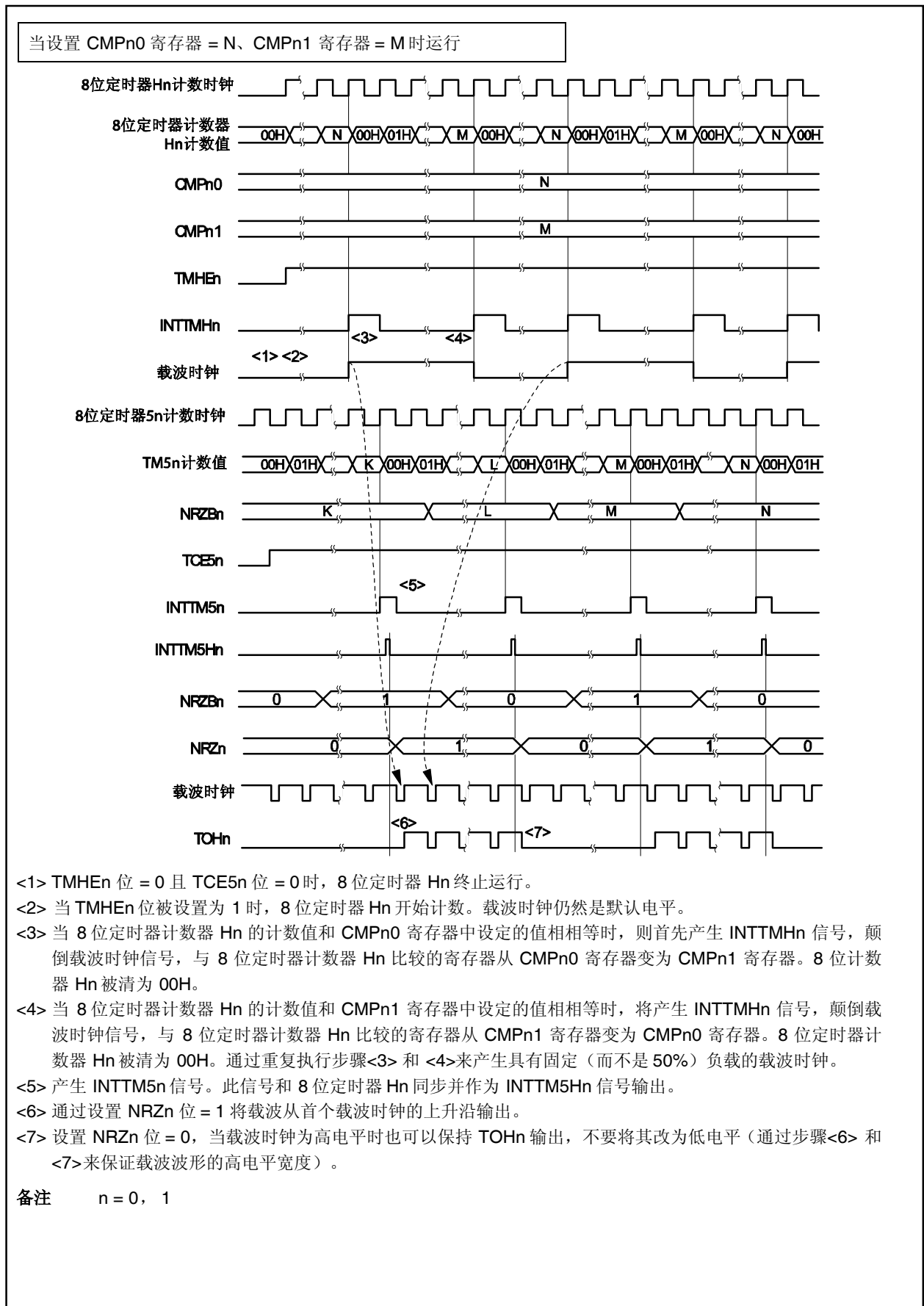
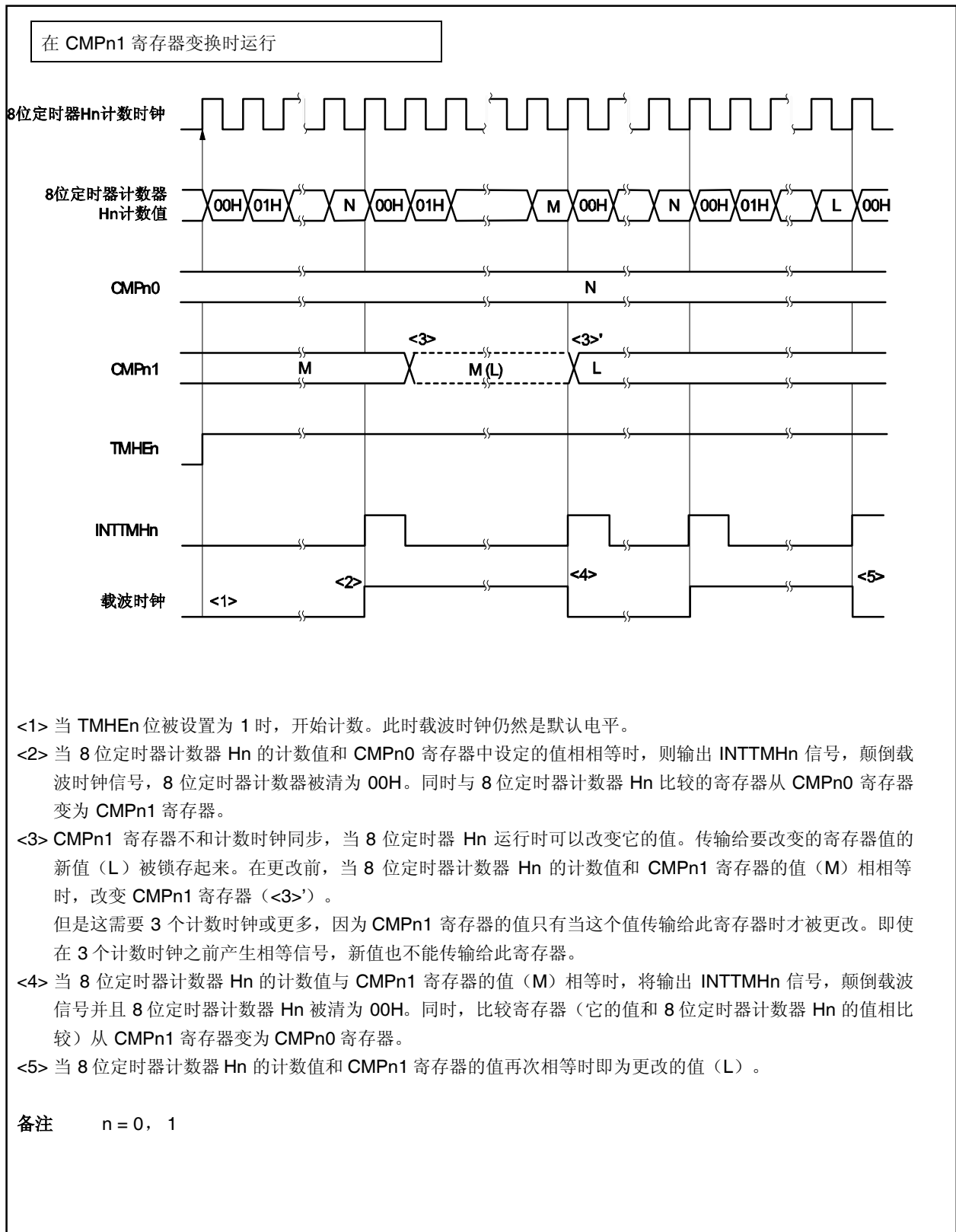


图 9-8. 载波发生器模式 (3/3)



第十章 间隔定时器和钟表定时器

V850ES/KE2 具有间隔定时器 BRG 和一个钟表定时器。间隔定时器 BRG 可以作为时钟定时器的时钟源来使用。钟表定时器也可以作为间隔定时器 WT 来使用。

两个间隔定时器通道和一个钟表定时器通道可以同时使用。

10.1 间隔定时器 BRG

10.1.1 功能

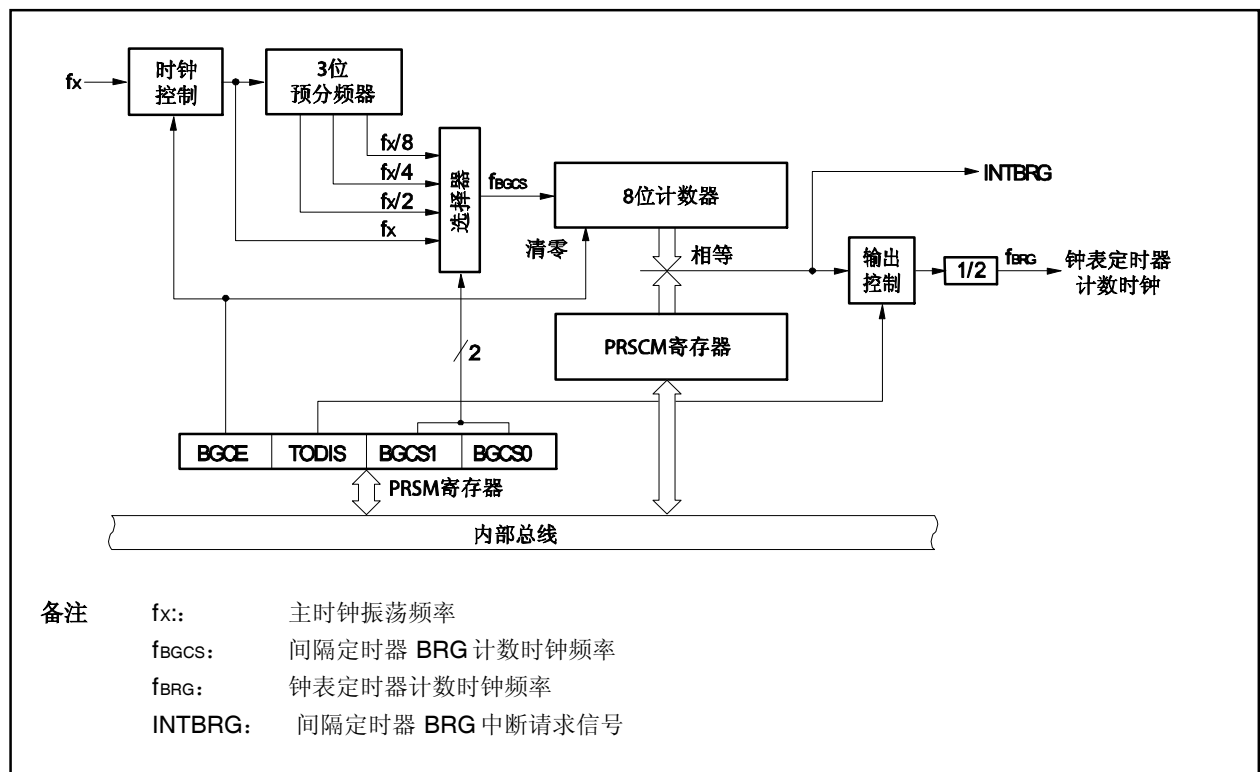
间隔定时器具有以下功能。

- 间隔定时器 BRG: 在指定的时间间隔内产生中断请求信号 (INTBRG)。
- 为钟表定时器生成计数时钟: 当主时钟作为计数时钟为钟表时钟提供时钟供给来使用时将产生一个计数时钟 (f_{BRG})。

10.1.2 配置

下面是间隔定时器 BRG 的框图。

图 10-1. 间隔定时器 BRG 框图



(1) 时钟控制

时钟控制可以控制间隔定时器 BRG 的操作。

(2) 3 位预分频器

3 位预分频器划分 f_x , 产生 $f_x/2$, $f_x/4$, 和 $f_x/8$ 。

(3) 选择器

选择器可以从 f_x , $f_x/2$, $f_x/4$, 和 $f_x/8$ 中为间隔定时器选择计数时钟(f_{BGS})。

(4) 8 位计数器

8 位计数器对计数时钟进行计数(f_{BGS})。

(5) 输出控制

输出控制可以控制计数时钟(f_{BRG})对钟表定时器的时钟供给。

(6) PRSCM 寄存器

PRSCM 寄存器是一个用于设置间隔时间的 8 位比较寄存器。

(7) PRSM 寄存器

PRSM 寄存器控制间隔定时器 BRG 与选择器的操作和时钟定时器的时钟供给。

10.1.3 寄存器

间隔定时器 BRG 包括以下几种寄存器。

(1) 间隔定时器 BRG 模式寄存器 (PRSM)

PRSM 控制间隔定时器 BRG 的操作、计数时钟的选择和对钟表定时器的时钟供给。

该寄存器支持 8 位或 1 位读写方式。

PRSM 寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF8B0H

	7	6	5	<4>	3	2	1	0
PRSM	0	0	0	BGCE	0	TODIS	BGCS1	BGCS0

BGCE	间隔定时器操作的控制
0	停止操作, 8位计数器清为01H
1	操作

TODIS	监视定时器时钟供应的控制
0	提供时钟给监视定时器
1	不提供时钟给监视定时器

BGCS1	BGCS0	输入时钟的选择 (f_{BGCS}) ^注	10 MHz	5 MHz	4 MHz
0	0	f_x	100 ns	200 ns	250 ns
0	1	$f_x/2$	200 ns	400 ns	500 ns
1	0	$f_x/4$	400 ns	800 ns	1 μ s
1	1	$f_x/8$	800 ns	1.6 μ s	2 μ s

注 对这些位进行设置, 使其满足以下条件。

$$V_{DD} = 4.0 \sim 5.5 \text{ V}: f_{BGCS} \leq 10 \text{ MHz}$$

$$V_{DD} = 2.7 \sim 4.0 \text{ V}: f_{BGCS} \leq 5 \text{ MHz}$$

注意事项 1. 当间隔定时器 BRG 正在运行时 (BGCE 位=1), 不要改变 TODIS、BGCS1 和 BGCS0 位的值。在设置 BGCE 位之前要先对 TODIS、BGCS1 和 BGCS0 位进行设置。

2. 当 BGCE 位被清除时 (清为 0), 则 8 位计数器即被清除。

(2) 间隔定时器 BRG 比较寄存器 (PRSCM)

PRSCM 是一个 8 位比较寄存器。

该寄存器支持 8 位读写方式

PRSCM 寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF8B1H

	7	6	5	4	3	2	1	0
PRSCM	PRSCM7	PRSCM6	PRSCM5	PRSCM4	PRSCM3	PRSCM2	PRSCM1	PRSCM0

注意事项 当间隔定时器 **BRG** 正在运行时，不要对 **PRSCM** 寄存器进行重写（**PRSM.BGCE** 位=1）。对 **BGCE** 位进行设置(1)之前要先设置 **PRSCM** 寄存器。

10.1.4 操作

(1) 间隔定时器 BRG 的操作

使用 PRSM 的 BGCS1 和 BGCS0 位对计数时钟进行设置，使用 PRSCM 寄存器来设置 8 位比较值。

当对 PRSM.BGCE 位进行设置时(1)，间隔定时器 BRG 就开始运行了。

每次 8 位计数器的计数值和 PRSCM 寄存器中设定的值相等时，都会产生一个中断请求信号 (INTBRG)。同时 8 位计数器被清为 01H 并继续计数。

<R>

间隔时间可以从下面这个方程中获取。

$$\text{间隔时间} = 2^m \times N / f_x$$

备注 m: 分度值(设置 BGCS1 和 BGCS0 位的值) = 0 ~ 3
 N: 在 PRSCM 寄存器^注中的集合值=1~256。(当在 PRSCM 寄存器中其设定的值为 00H 时, 则 N=256)
 f_x: 主时钟振荡频率

<R>

注 当 PRSCM 寄存器= 01H 时, 只产生一次 INTBRG 中断。

(2) 为钟表定时器提供计数时钟

使用 PRSM 的 BGCS1 和 BGCS0 位对计数时钟进行设置，使用 PRSCM 设置 8 位比较值，这样就可以使钟表定时器的计数时钟频率值为 32.768 kHz。同时要清除(0) PRSM.TODIS 位。

当对 PRSM.BGCE 位进行设置时(1)，则将提供给钟表定时器。

f_{BRG} 从下面这个方程中获取。

$$f_{BRG} = f_x / (2^{m+1} \times N)$$

为了将 f_{BRG} 设置为 32.768 kHz，需要进行以下运算，来设置 BGCS1 与 BGCS0 位和 PRSCM 寄存器。

- <1> 设置 N = f_x/65,536 (对小数进行四舍五入) 使 m = 0。
- <2> 如果 N 为偶数, 则 N = N/2, m = m + 1
- <3> 重复执行步骤 <2>, 直到 N 为偶数或 m = 3
- <4> 对 PRSCM 设置为 N, 对 BGCS1 和 BGCS0 位设置为 m。

例: 当 f_x = 4.00 MHz

- <1> N = 4,000,000/65,536 = 61 (对小数进行四舍五入), m = 0
- <2>, <3> 因为 N 是奇数, 所以这些值仍然是 N = 61, m = 0
- <4> 在 PRSCM 寄存器中的设定值: 3DH (61), 在 BGCS1 和 BGCS0 位中的设定值: 00

备注 m: 分度值 (设置 BGCS1 和 BGCS0 位的值) = 0 ~ 3
 N: 在 PRSCM 寄存器中的集合值=1~256 (当在 PRSCM 寄存器中其设定的值为 00H 时, 则 N=256)。
 f_x: 主时钟振荡频率

10.2 钟表定时器

10.2.1 功能

钟表定时器具有以下功能。

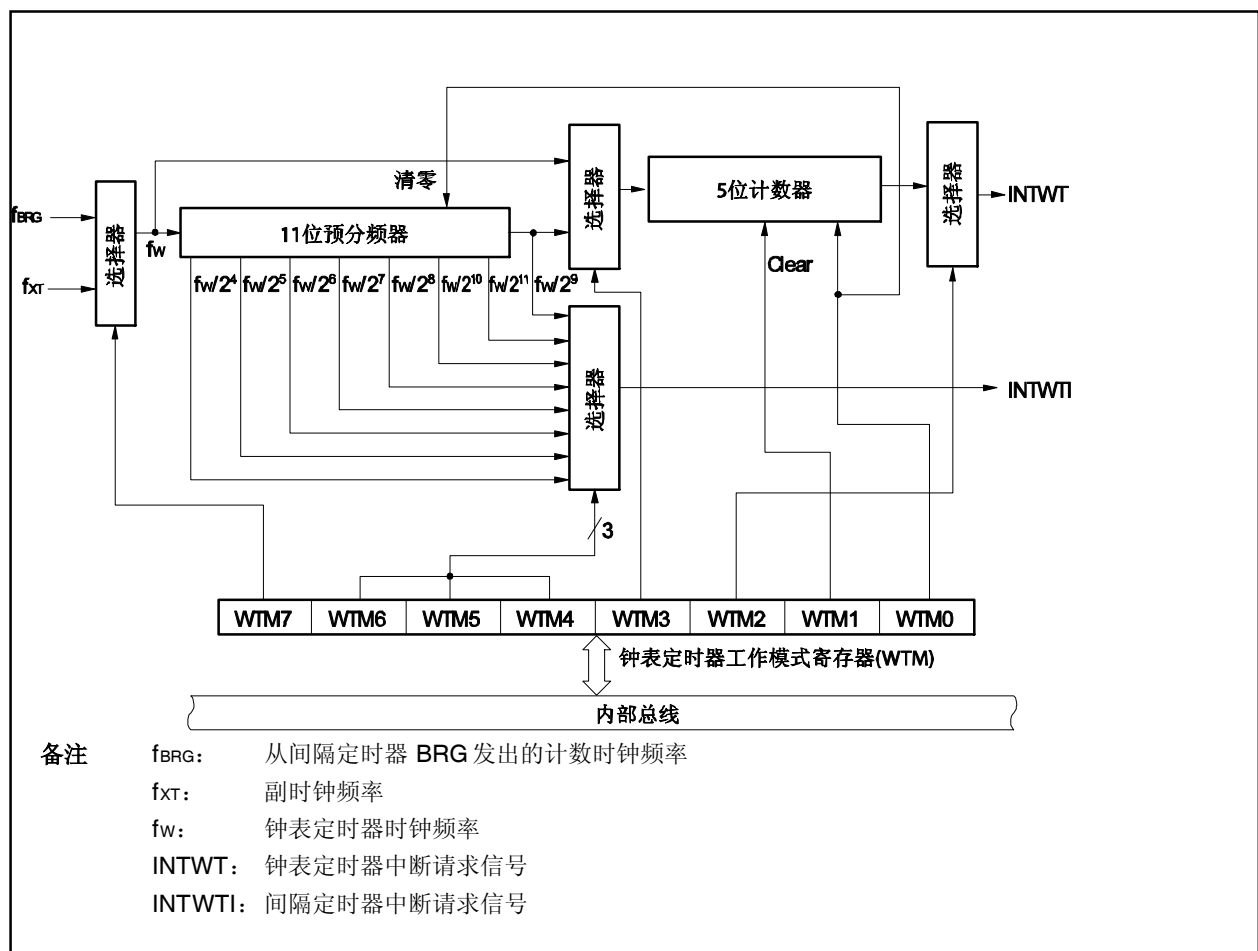
- 钟表定时器： 使用主时钟或副时钟每隔 0.5 秒或 0.25 秒产生一个中断请求信号(INTWT)。
- 间隔定时器： 在预设置时间间隔内产生一个中断请求信号(INTWT)。

钟表定时器和间隔定时器的功能可以同时使用。

10.2.2 配置

下面是钟表定时器的框图。

图 10-2. 钟表定时器框图



(1) 11 位预分频器

11 位预分频器通过对 f_w 进行划分来产生 $f_w/2^4$ 到 $f_w/2^{11}$ 的时钟。

(2) 5 位计数器

5 位计数器通过 f_w 或 $f_w/2^9$ 对进行计数每隔 $2^4/f_w$, $2^5/f_w$, $2^{13}/f_w$, 或 $2^{14}/f_w$ 产生一个钟表定时器中断请求信号 (INTWT)。

(3) 选择器

钟表定时器具有以下 4 个选择器。

- 选择主时钟（从间隔定时器 BRG(f_{BRG})中产生的时钟）或副时钟 (fXT) 作为钟表定时器时钟的选择器。
- 选择 f_w 或 $f_w/2^9$ 作为 5 位计数器计数时钟频率的选择器。
- 选择 $2^4/f_w$ 或 $2^{13}/f_w$, 或 $2^5/f_w$ 或 $2^{14}/f_w$ 作为 INTWT 信号产生时间间隔的选择器。
- 从 $2^4/f_w$ 到 $2^{11}/f_w$ 中选择间隔定时器 WT 中断请求信号 (INTWTI) 产生时间间隔的选择器。

(4) 8 位计数器

8 位计数器对计数时钟进行计数(f_{BGCS})。

(5) WTM 寄存器

WTM 寄存器是一个 8 位寄存器，它控制钟表定时器/间隔定时器 WT 的操作，并设置中断请求信号产生的时间间隔。

10.2.3 寄存器

钟表定时器中具有下面这个寄存器。

(1) 钟表定时器操作模式寄存器(WTM)

该寄存器可以启用或禁用钟表定时器的计数时钟和操作，设置 11 位预分频器的时间间隔，控制 5 位计数器的操作，设置钟表定时器中断请求信号 (INTWT) 的产生时间。

WTM 寄存器支持 8 位或 1 位读写方式。

WTM 寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFFF680H

	7	6	5	4	3	2	<1>	<0>
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	WTM6	WTM5	WTM4	间隔定时器中断(INTWTI) 时间选择
0	0	0	0	$2^4/f_w$ (488 ns: $f_w = f_{XT}$)
0	0	0	1	$2^5/f_w$ (977 ns: $f_w = f_{XT}$)
0	0	1	0	$2^6/f_w$ (1.95 ms: $f_w = f_{XT}$)
0	0	1	1	$2^7/f_w$ (3.91 ms: $f_w = f_{XT}$)
0	1	0	0	$2^8/f_w$ (7.81 ms: $f_w = f_{XT}$)
0	1	0	1	$2^9/f_w$ (15.6 ms: $f_w = f_{XT}$)
0	1	1	0	$2^{10}/f_w$ (31.3 ms: $f_w = f_{XT}$)
0	1	1	1	$2^{11}/f_w$ (62.5 ms: $f_w = f_{XT}$)
1	0	0	0	$2^4/f_w$ (488 ns: $f_w = f_{BRG}$)
1	0	0	1	$2^5/f_w$ (977 ns: $f_w = f_{BRG}$)
1	0	1	0	$2^6/f_w$ (1.95 ms: $f_w = f_{BRG}$)
1	0	1	1	$2^7/f_w$ (3.91 ms: $f_w = f_{BRG}$)
1	1	0	0	$2^8/f_w$ (7.81 ms: $f_w = f_{BRG}$)
1	1	0	1	$2^9/f_w$ (15.6 ms: $f_w = f_{BRG}$)
1	1	1	0	$2^{10}/f_w$ (31.3 ms: $f_w = f_{BRG}$)
1	1	1	1	$2^{11}/f_w$ (62.5 ms: $f_w = f_{BRG}$)

WTM7	WTM3	WTM2	监视定时器中断(INTWT)时间选择
0	0	0	$2^{14}/f_w$ (0.5 s: $f_w = f_{XT}$)
0	0	1	$2^{13}/f_w$ (0.25 s: $f_w = f_{XT}$)
0	1	0	$2^5/f_w$ (977 ns: $f_w = f_{XT}$)
0	1	1	$2^4/f_w$ (488 ns: $f_w = f_{XT}$)
1	0	0	$2^{14}/f_w$ (0.5 s: $f_w = f_{BRG}$)
1	0	1	$2^{13}/f_w$ (0.25 s: $f_w = f_{BRG}$)
1	1	0	$2^5/f_w$ (977 ns: $f_w = f_{BRG}$)
1	1	1	$2^4/f_w$ (488 ns: $f_w = f_{BRG}$)

WTM1	5位计数器操作的控制
0	操作停止后清零
1	开始

WTM0	容许监视定时器操作
0	停止操作 (清零预分频器及5位计数器)
1	容许操作

注意事项 当 WTM0 和 WTM1 位设置为 0 时, 将 WTM2 重新写入到 WTM7 位中。

备注 1. f_w : 钟表定时器时钟频率

2. 当 $f_w = 32.768$ kHz 时, 应用圆括号中的值

10.2.4 操作

(1) 作为钟表定时器的操作

钟表定时器在固定的时间间隔内产生一个中断请求。钟表定时器运行时副时钟(32.768 kHz)时间间隔是 0.25 或 0.5 秒。

当 WTM.WTM0 和 WTM.WTM1 位被设置为 11 时启动计数。当这些位被清为 00 时，则 11 位预分频器和 5 位计数器被清除，并且终止计数操作。

当钟表定时器和间隔定时器 WT 同时运行时，通过将 WTM1 位清为 0 使 5 位计数器与时间同步。此时在钟表定时器中也许会产生 15.6ms 的错误，但是间隔定时器 WT 并不会受到影响。

(2) 作为间隔定时器的操作

钟表定时器也可以作为间隔定时器来使用，它会在指定的时间间隔（此时间间隔通过提前设置的计数值来确定）内重复地产生中断请求信号（INTWTI）。

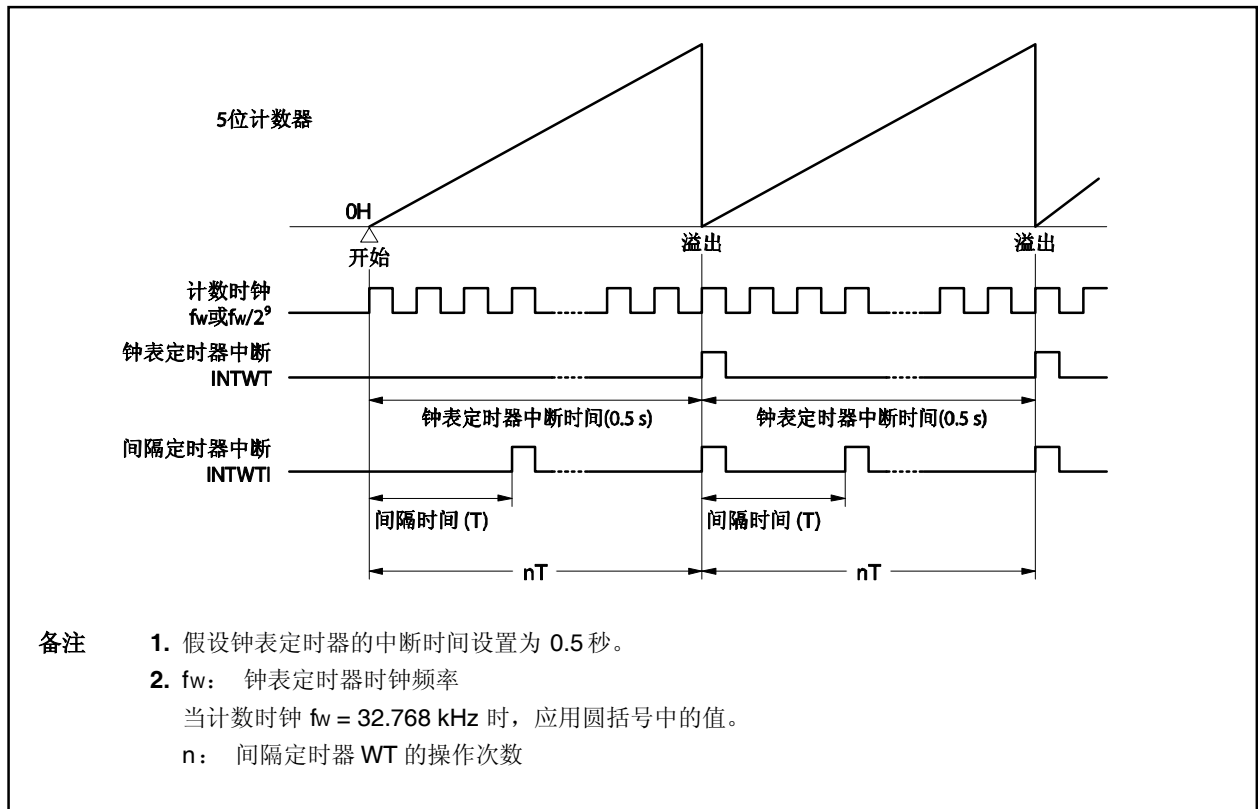
通过 WTM.WTM4 到 WTM.WTM7 位来选择时间间隔。

表 10-1. 间隔定时器的时间间隔

WTM7	WTM6	WTM5	WTM4	时间间隔	
0	0	0	0	$2^4 \times 1/f_w$	488 μ s ($f_w = f_{XT} = 32.768$ kHz)
0	0	0	1	$2^5 \times 1/f_w$	977 μ s ($f_w = f_{XT} = 32.768$ kHz)
0	0	1	0	$2^6 \times 1/f_w$	1.95 ms ($f_w = f_{XT} = 32.768$ kHz)
0	0	1	1	$2^7 \times 1/f_w$	3.91 ms ($f_w = f_{XT} = 32.768$ kHz)
0	1	0	0	$2^8 \times 1/f_w$	7.81 ms ($f_w = f_{XT} = 32.768$ kHz)
0	1	0	1	$2^9 \times 1/f_w$	15.6 ms ($f_w = f_{XT} = 32.768$ kHz)
0	1	1	0	$2^{10} \times 1/f_w$	31.3 ms ($f_w = f_{XT} = 32.768$ kHz)
0	1	1	1	$2^{11} \times 1/f_w$	62.5 ms ($f_w = f_{XT} = 32.768$ kHz)
1	0	0	0	$2^4 \times 1/f_w$	488 μ s ($f_w = f_{BRG} = 32.768$ kHz)
1	0	0	1	$2^5 \times 1/f_w$	977 μ s ($f_w = f_{BRG} = 32.768$ kHz)
1	0	1	0	$2^6 \times 1/f_w$	1.95 ms ($f_w = f_{BRG} = 32.768$ kHz)
1	0	1	1	$2^7 \times 1/f_w$	3.91 ms ($f_w = f_{BRG} = 32.768$ kHz)
1	1	0	0	$2^8 \times 1/f_w$	7.81 ms ($f_w = f_{BRG} = 32.768$ kHz)
1	1	0	1	$2^9 \times 1/f_w$	15.6 ms ($f_w = f_{BRG} = 32.768$ kHz)
1	1	1	0	$2^{10} \times 1/f_w$	31.3 ms ($f_w = f_{BRG} = 32.768$ kHz)
1	1	1	1	$2^{11} \times 1/f_w$	62.5 ms ($f_w = f_{BRG} = 32.768$ kHz)

备注 fw: 钟表定时器时钟频率

图 10-3. 钟表定时器/间隔定时器的定时操作

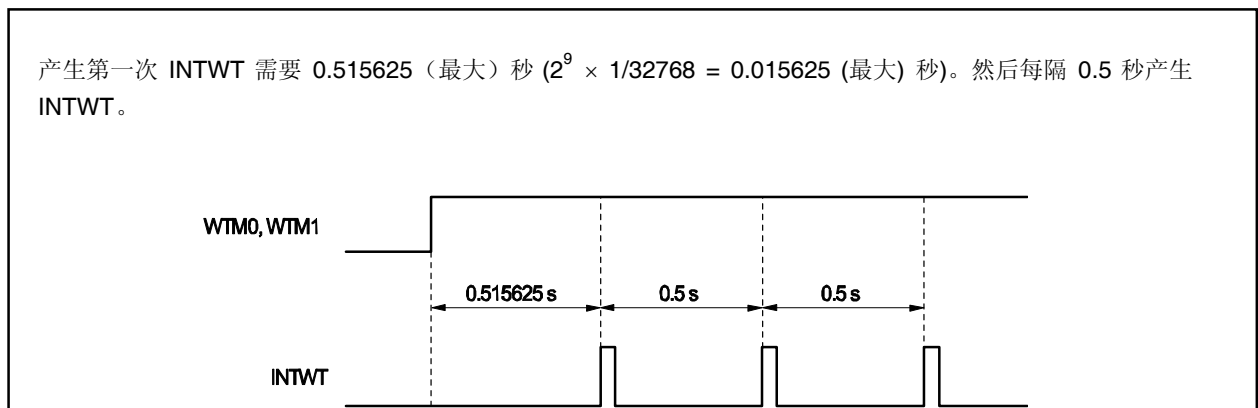


10.3 注意事项

(1) 作为钟表定时器的操作

启动之后, 产生第一个钟表定时器中断请求(INTWT)之前需要一段时间(WTM.WTM1 和 WTM.WTM0 位 = 11)。

图 10-4. 产生钟表定时器中断请求的实例 (INTWT)
(当中断周期= 0.5 秒时)



(2) 当钟表定时器和间隔定时器 BRG 同时运行时

当对钟表定时器将副时钟设置为计数时钟时，间隔定时器 BRG 的时间间隔可以设置为任意值。改变此间隔时间并不会对钟表定时器造成影响（改变间隔时间前要终止其运行）。

当对钟表定时器使用主时钟作为计数时钟时，要将间隔定时器 BRG 的时间间隔设置为 65.536 kHz（近似值）。不要更改此数值。

(3) 当间隔定时器 BRG 和间隔定时器 WT 同时运行时

当对间隔定时器 WT 使用副时钟作为计数时钟时，间隔定时器 BRG 和 WT 的间隔时间可以被设置为任意值。也可以在以后的过程中对其进行更改（改变它的值之前要终止其运行）。

当对间隔定时器 WT 使用主时钟作为计数时钟时，间隔定时器 BRG 的间隔时间可以设置为任意值，但是以后不能再对其进行改变（只有当间隔定时器 WT 终止运行时才能对其进行更改）。间隔定时器 WT 的时间间隔可以设置成 $\times 2^5$ 到 $\times 2^{12}$ （间隔定时器 BRG 中的集合值）。它也可以在以后的过程中进行更改。

(4) 当钟表定时器和间隔定时器 WT 同时运行时

间隔定时器 WT 的间隔时间可以设置为 488 μ s 和 62.5 ms 之间的值。以后不能对其进行更改。

当钟表定时器正在运行时不要终止间隔定时器 WT（清除(0) WTM.WTM0 位）。如果在清除(0)之后对 WTM0 位进行设置(1)，则钟表定时器将会产生 0.5 或 0.25 秒的偏差。

(5) 当钟表定时器、间隔定时器 BRG 和间隔定时器 WT 同时运行时

当对钟表定时器使用副时钟作为计数时钟时，间隔定时器 BRG 和 WT 的时间间隔可以设置为任意值。在以后的过程中可以对间隔定时器 BRG 的间隔时间进行改变（改变前要终止其运行）。

当对钟表定时器使用主时钟作为计数时钟时，将间隔定时器 BRG 的时间间隔设置约为 65.536 kHz。以后不能对其进行改变。间隔定时器 WT 的时间间隔可以设置为 488 μ s 和 62.5 ms 之间的值。以后也不能对其进行改变。

当钟表定时器正在运行时，不要终止间隔定时器 BRG（清除(0) PRSM.BGCE 位）或间隔定时器 WT（清除(0) WTM.WTM0 位）。

第十一章 看门狗定时器的功能

11.1 看门狗定时器 1

11.1.1 功能

看门狗定时器 1 的操作模式如下。

- 看门狗定时器
- 时间间隔定时器

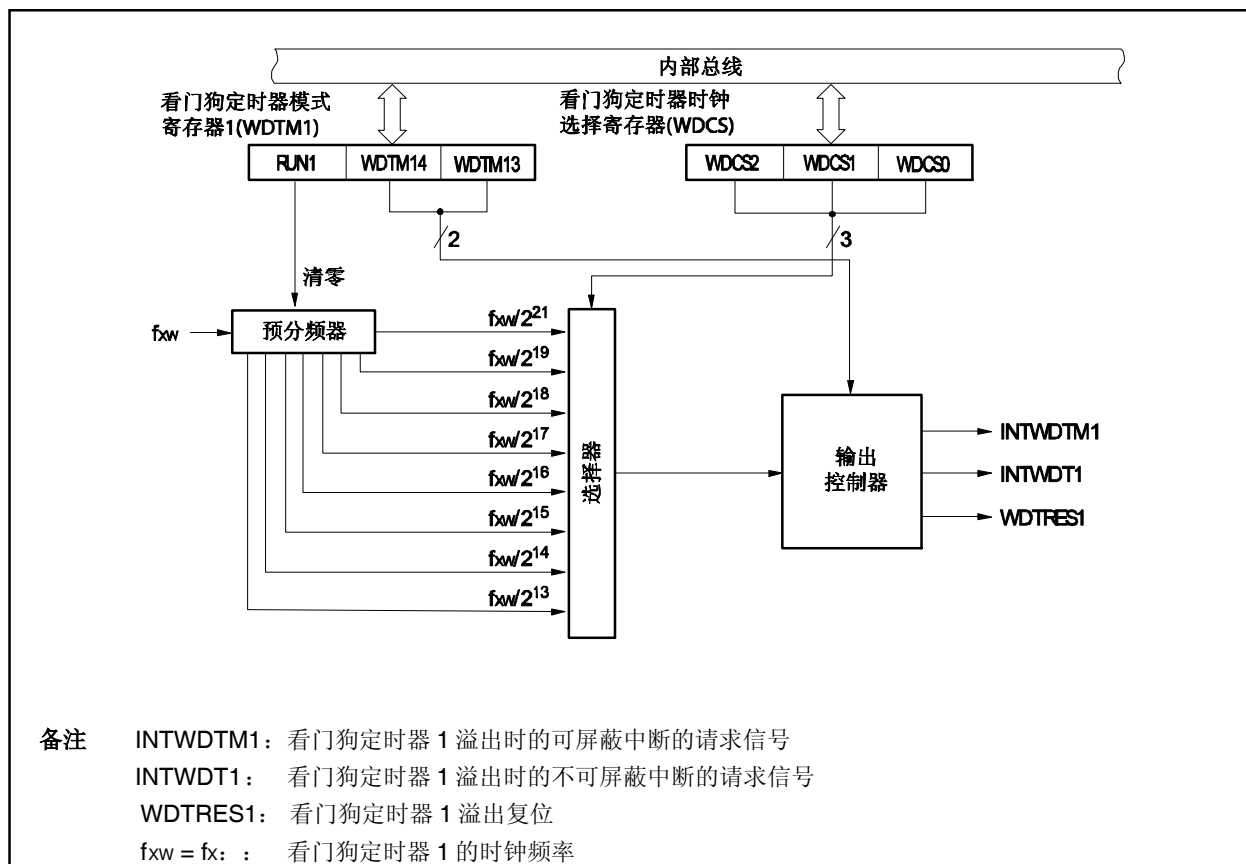
下述功能可由上面列出的模式实现。

- 看门狗定时器 1^注溢出时不可屏蔽中断的请求信号 (INTWDT1) 的产生
- 看门狗定时器 1 溢出时的系统复位信号 (WDTRES1) 的产生
- 时间间隔定时器溢出时的可屏蔽中断的请求信号 (INTWDTM1) 的产生

注 由不可屏蔽中断的请求引起的不可屏蔽中断的请求信号 (INTWDT1, INTWDT2)，参见 17.10 注意事项。

备注 选择是否使用在看门狗定时器 1 的模式下的看门狗定时器 1 或与 WDTM1 寄存器一起的时间间隔定时器模式。

图 11-1. 看门狗定时器 2 框图



11.1.2 设置

看门狗定时器 1 包括以下硬件。

表 11-1. 看门狗定时器 1 的设置

项目	设置
控制寄存器	看门狗定时器选择寄存器 (WDCS) 看门狗定时器模式寄存器 1 (WDTM1)

11.1.3 寄存器

控制看门狗定时器 1 的寄存器如下。

- 看门狗定时器时钟选择寄存器 (WDCS)
- (WDTM1) 看门狗定时器 1 模式寄存器

(1) 看门狗定时器时钟选择寄存器 (WDCS)

该寄存器设定看门狗定时器 1 的溢出时间和间隔时间定时器。

WDCS 寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF6C1H								
	7	6	5	4	3	2	1	0
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0
	WDCS2	WDCS1	WDCS0	看门狗定时器1及间隔定时器的溢出时间				
				f _{xw}				
				4 MHz	5 MHz	10 MHz		
	0	0	0	2 ¹³ /f _{xw}	2.048 ms	1.638 ms	0.819 ms	
	0	0	1	2 ¹⁴ /f _{xw}	4.096 ms	3.277 ms	1.638 ms	
	0	1	0	2 ¹⁵ /f _{xw}	8.192 ms	6.554 ms	3.277 ms	
	0	1	1	2 ¹⁶ /f _{xw}	16.38 ms	13.11 ms	6.554 ms	
	1	0	0	2 ¹⁷ /f _{xw}	32.77 ms	26.21 ms	13.11 ms	
	1	0	1	2 ¹⁸ /f _{xw}	65.54 ms	52.43 ms	26.2 ms	
	1	1	0	2 ¹⁹ /f _{xw}	131.1 ms	104.9 ms	52.43 ms	
	1	1	1	2 ²¹ /f _{xw}	524.3 ms	419.4 ms	209.7 ms	
备注 f _{xw} = f _x : 看门狗定时器时钟频率								

(2) 看门狗定时器模式寄存器 (WDTM1)

该寄存器设定看门狗定时器 1 的操作模式和使能计数操作。

该寄存器是一个只能按照特殊序列写入的特殊寄存器 (参见 3.4.7 特殊寄存器)。

WDTM1 寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

注意事项 当主时钟被停止且 CPU 运行在次时钟状态下时, 请勿访问 WDTM1 寄存器。

关于更多细节, 敬请参阅 3.4.8 (2)。

复位后: 00H R/W 地址: FFFF6C2H

<?>	6	5	4	3	2	1	0
WDTM1	RUN1	0	0	WDTM14	WDTM13	0	0

RUN1	看门狗定时器 1 操作模式的选择 ^{注1}
0	停止计数
1	清零计数器及开始计数

WDTM14	WDTM13	看门狗定时器 1 操作模式的选择 ^{注2}
0	0	间隔定时器模式 (在生成溢出,可屏蔽中断 INTWDTM1时.)
0	1	
1	0	看门狗定时模式 1 ^{注3} (在生成溢出,不可屏蔽中断 INTWDT1 时.)
1	1	看门狗定时模式 2 (在启用溢出,复位操作WDTRES1 时.)

注

1. 一旦 RUN1 位被设定 (为 1) 时, 它不能被软件清零。
因此, 当计数被启动后, 它只能被复位操作停止。
2. 一旦 WDTM13 和 WDTM14 位被复位 (为 1), 它们就不能被软件清除 (为 0), 仅可以通过复位清除。
3. 由不可屏蔽中断的请求信号 (INTWDT1) 引起的不可屏蔽中断的服务, 参见 17.10 注意事项。

11.1.4 操作

(1) 看门狗定时器 1 的操作

操作看门狗定时器 1 以侦测一个由设定 WDTM1.WDTM14 位数为 1 而选择好的程序循环。

看门狗定时器 1 计数时钟（由程序循环侦测的时间间隔）能用从 DCS.WDCS0 到 WDCS.WDCS2 的位数来选择。通过设置 WDTM1.RUN1 为 1 来开始计数操作。当开始以后，RUN1 在设置程序循环侦测的时间间隔时也被置位为 1，看门狗定时器 1 即被清零且计数操作重新开始。

如果 RUN1 位没有置 1，并且超出了程序循环监测时间，则通过 WDTM1.WDTM13 值决定产生一个系统复位信号（WDTRES1），或者产生一个非屏蔽中断请求信号（INTWDT1）。

看门狗定时器 1 在 STOP 和 IDLE 模式停止计数。为了清零看门狗定时器 1，在进入 STOP 和 IDLE 模式之前将 RUN1 位置 1。

因为看门狗定时器 1 在 HALT 模式工作，务必保证 HALT 模式下看门狗定时器 1 不发生溢出。

- 注意事项**
1. 当选择副时钟作为 CPU 时钟时，看门狗定时器 1 计数停止（看门狗定时器 1 数值保持）。
 2. INTWDT1 信号的非屏蔽中断服务，参见 17.10 注意事项。

表 11-2. 看门狗定时器 1 的程序循环检测时间

时钟	程序循环检测时间		
	f _{xw} = 4 MHz	f _{xw} = 5 MHz	f _{xw} = 10 MHz
2 ¹³ /f _{xw}	2.048 ms	1.638 ms	0.819 ms
2 ¹⁴ /f _{xw}	4.096 ms	3.277 ms	1.683 ms
2 ¹⁵ /f _{xw}	8.192 ms	6.554 ms	3.277 ms
2 ¹⁶ /f _{xw}	16.38 ms	13.11 ms	6.554 ms
2 ¹⁷ /f _{xw}	32.77 ms	26.21 ms	13.11 ms
2 ¹⁸ /f _{xw}	65.54 ms	52.43 ms	26.21 ms
2 ¹⁹ /f _{xw}	131.1 ms	104.9 ms	52.43 ms
2 ²¹ /f _{xw}	524.3 ms	419.4 ms	209.7 ms

备注 f_{xw} = f_x: 看门狗定时器 1 时钟频率

(2) 用于间隔定时器的操作

当 WDTM1.WDTM14 位清零时，看门狗定时器 1 作为间隔定时器，以预先设定好的时间间隔不断产生中断。

在间隔定时器模式，WDTIC 寄存器中的中断屏蔽标志 (WDTMK) 和优先级指定标志 (WDTPR0 ~WDTPR2) 有效，并可以产生可屏蔽中断请求信号 (INTWDTM1)。INTWDTM1 信号在所有可屏蔽中断中默认优先级最高。

间隔定时器在 HALT 模式仍然工作，但在 STOP 及 IDLE 模式停止。

- 注意事项**
1. 一旦设置 WDTM14 位为 1 (即选择看门狗定时器 1 模式)，只有复位才能使其进入间隔定时器模式。
 2. 当选择副时钟作为 CPU 时钟时，看门狗定时器 1 计数停止 (看门狗定时器 1 数值保持)。

表 11-3. 间隔定时器的间隔时间

时钟	间隔时间		
	$f_{xw} = 4 \text{ MHz}$	$f_{xw} = 5 \text{ MHz}$	$f_{xw} = 10 \text{ MHz}$
$2^{13}/f_{xw}$	2.048 ms	1.638 ms	0.819 ms
$2^{14}/f_{xw}$	4.096 ms	3.277 ms	1.638 ms
$2^{15}/f_{xw}$	8.192 ms	6.554 ms	3.277 ms
$2^{16}/f_{xw}$	16.38 ms	13.11 ms	6.554 ms
$2^{17}/f_{xw}$	32.77 ms	26.21 ms	13.11 ms
$2^{18}/f_{xw}$	65.54 ms	52.43 ms	26.21 ms
$2^{19}/f_{xw}$	131.1 ms	104.9 ms	52.43 ms
$2^{21}/f_{xw}$	524.3 ms	419.4 ms	209.7 ms

备注 $f_{xw} = f_x$: 看门狗定时器 1 时钟频率

11.2 看门狗定时器 2

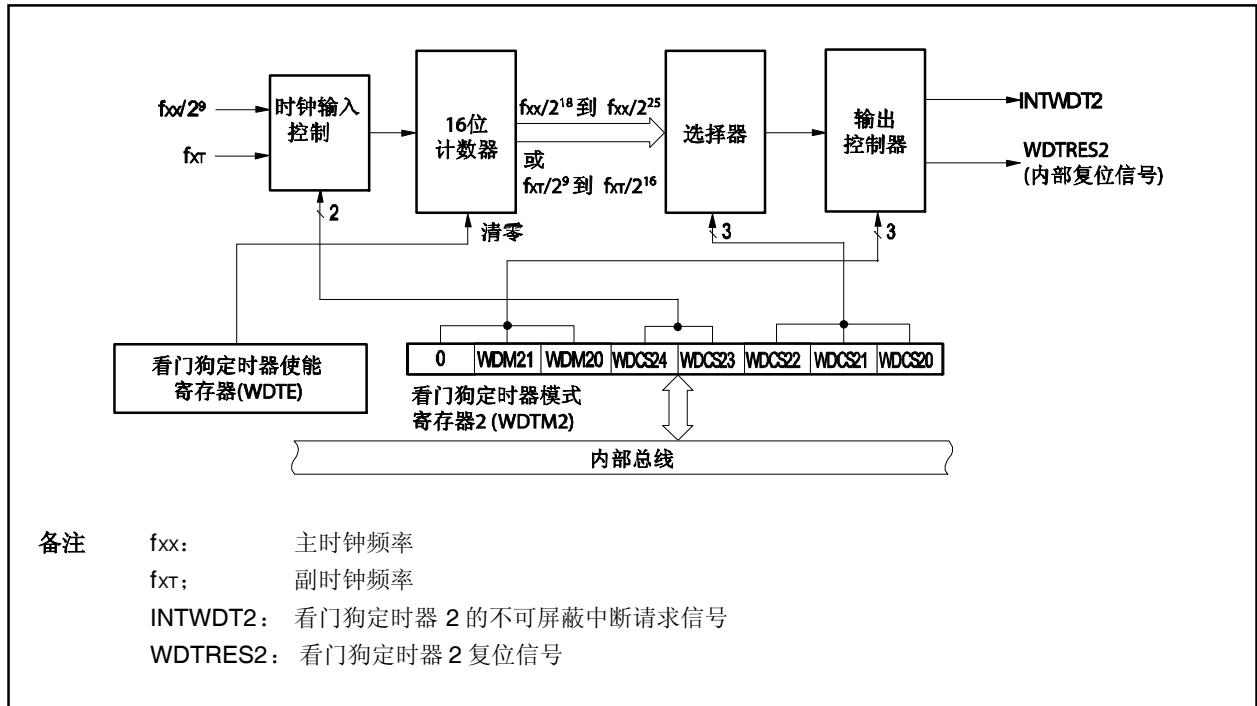
11.2.1 功能

看门狗定时器 2 的功能如下。

- 看门狗定时器默认开始工作^{注 1}
 - 复位模式：看门狗定时器 2 溢出时复位（产生信号 WDT2RES）
 - 不可屏蔽中断请求模式：看门狗定时器 2 溢出时 NMI 操作（产生信号 INTWDT2）^{注 2}
- 源时钟可选主时钟和副时钟

- 注**
- 复位释放后，看门狗定时器 2 会自动启动。
 当不使用看门狗定时器 2，在复位前通过该功能使其停止工作，或清除看门狗定时器 2 并在下一次间隔开始前使其停止工作。
 为了确认操作的正确性，即使在不需要改变默认设置（复位模式，间隔时间： $f_{xx}/2^{25}$ ）的情况下，也要对寄存器 WDTM2 执行一次写操作。
 - 由不可屏蔽中断请求信号（INTWDT2）引发的不可屏蔽中断服务，可参见 17.10 注意事项。

图 11-2. 看门狗定时器 2 的框图



11.2.2 配置

看门狗定时器 2 由以下寄存器控制。

表 11-4. 看门狗定时器 2 的配置

项目	配置
控制寄存器	看门狗定时器模式寄存器 2 (WDTM2) 看门狗定时器使能寄存器 (WDTE)

11.2.3 寄存器

(1) 看门狗定时器模式寄存器 2 (WDTM2)

该寄存器设置溢出时间和看门狗定时器 2 的操作时钟。

可由 8 位存储器操作指令设置该寄存器。该寄存器可多次读取，但复位释放后只能写入一次。

该寄存器复位后的值为 67H。

注意事项 当主时钟停止，CPU 工作在副时钟时，不要访问 WDTM2 寄存器。

关于更多细节，敬请参阅 3.4.8 (2)。

复位后: 67H R/W 地址: FFFFF6D0H

WDTM2	7	6	5	4	3	2	1	0
	0	WDM21	WDM20	WDCS24	WDCS23	WDCS22	WDCS21	WDCS20

WDM21	WDM20	看门狗定时器2操作模式的选择
0	0	停止操作
0	1	不可屏蔽中断请求模式 (NTWDT2的生成)
1	-	复位模式 (WDTRES2的生成)

注意事项

1. 若要停止看门狗计时器 2，向 WDTM2 寄存器写入“1FH”。
2. WDCS0 ~WDCS4 的详细情况，参见表 11-5 看门狗定时器 2 的时钟选择。
3. 如果复位后 2 次写 WDTM2 寄存器，则会强制输出溢出信号。
4. 若要有意产生一个溢出信号，向 WDTM2 寄存器仅写入两次数据，或向 WDTE 寄存器仅写入一次不等于“ACH”的数据。

然而，当看门狗计时器 2 停止工作时不产生溢出信号，即使向 WDTM2 寄存器仅写入两次数据，或向 WDTE 寄存器仅写入一次不等于“ACH”的数据。

<R

表 11-5. 看门狗定时器 2 的时钟选择

WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	Selected Clock	f _{xx} = 20 MHz	f _{xx} = 16 MHz	f _{xx} = 10 MHz
0	0	0	0	0	2 ¹⁸ /f _{xx}	13.1 ms	16.4 ms	26.2 ms
0	0	0	0	1	2 ¹⁹ /f _{xx}	26.2 ms	32.8 ms	52.4 ms
0	0	0	1	0	2 ²⁰ /f _{xx}	52.4 ms	65.5 ms	104.9 ms
0	0	0	1	1	2 ²¹ /f _{xx}	104.9 ms	131.1 ms	209.7 ms
0	0	1	0	0	2 ²² /f _{xx}	209.7 ms	262.1 ms	419.4 ms
0	0	1	0	1	2 ²³ /f _{xx}	419.4 ms	524.3 ms	838.9 ms
0	0	1	1	0	2 ²⁴ /f _{xx}	838.9 ms	1048.6 ms	1677.7 ms
0	0	1	1	1	2 ²⁵ /f _{xx}	1677.7 ms	2097.2 ms	3355.4 ms
0	1	0	0	0	2 ⁹ /f _{XT}	15.625 ms (f _{XT} = 32.768 kHz)		
0	1	0	0	1	2 ¹⁰ /f _{XT}	31.25 ms (f _{XT} = 32.768 kHz)		
0	1	0	1	0	2 ¹¹ /f _{XT}	62.5 ms (f _{XT} = 32.768 kHz)		
0	1	0	1	1	2 ¹² /f _{XT}	125 ms (f _{XT} = 32.768 kHz)		
0	1	1	0	0	2 ¹³ /f _{XT}	250 ms (f _{XT} = 32.768 kHz)		
0	1	1	0	1	2 ¹⁴ /f _{XT}	500 ms (f _{XT} = 32.768 kHz)		
0	1	1	1	0	2 ¹⁵ /f _{XT}	1000 ms (f _{XT} = 32.768 kHz)		
0	1	1	1	1	2 ¹⁶ /f _{XT}	2000 ms (f _{XT} = 32.768 kHz)		
1	x	x	x	x	操作停止			

(2) 看门狗定时器使能寄存器 (WDTE)

将“ACH”写入寄存器 WDTE，则看门狗定时器 2 的计数器被清零，计数重新开始。

该寄存器支持 8 位读写方式。

该寄存器复位后的值为 9AH。

复位后: 9AH R/W 地址: FFFFF6D1H

7	6	5	4	3	2	1	0

WDTE

注意事项

1. 如果非“ACH”的数值被写入寄存器 WDTE，必定产生溢出信号。
2. 当向寄存器 WDTE 执行 1 位存储器操作指令，必定产生溢出信号。
3. 读取寄存器 WDTE 的值为“9AH”（与写入值“ACH”不同）。
4. 若若有意产生一个溢出信号，向 WDTM2 寄存器仅写入两次数据，或向 WDTE 寄存器仅写入一次不等于“ACH”的数据。

然而，当看门狗计时器 2 停止工作时不产生溢出信号，即使向 WDTM2 寄存器仅写入两次数据，或向 WDTE 寄存器仅写入一次不等于“ACH”的数据。

<R>

11.2.4 操作

复位释放后，看门狗定时器 2 自动开始运行。

复位后寄存器 WDTM2（使用字节访问操作）只可被写入一次。要使用看门狗定时器 2，使用 8 位操作指令，向寄存器 WDTM2 写入操作模式和间隔时间。之后，将无法停止看门狗定时器 2 的操作。

看门狗定时器 2 程序循环监测时间可由 WDTM2.WDCS24 到 WDTM2.WDCS20 位选择。向寄存器 WDTE 写入 ACH，就会使看门狗定时器 2 的计数器清零，并开始重新计数操作。在计数器开始操作后，循环检测时间间隔内向寄存器 WDTE 写入 ACH。

如果在定时间隔记满时没有对寄存器 WDTE 进行 ACH 写操作，就会根据 WDM21 和 WDTM2.WDM20 的设置值，产生复位信号（WDT2RES）或不可屏蔽中断请求信号（INTWDT2）。

要不使用看门狗定时器 2，向寄存器 WDTM2 写入 1FH。

不可屏蔽中断请求模式下的不可屏蔽中断请求服务可参见 17.10 注意事项。

注意事项

如果选择主时钟作为看门狗定时器 2 的时钟源，看门狗定时器在 IDLE/STOP 模式将停止工作。因此，在设置 IDLE/STOP 模式前，通过向 WDTE 寄存器写入 ACH 将看门狗定时器 2 清零。

在 HALT 模式时，或 IDLE/STOP 模式下选择副时钟作为其源时钟时，看门狗定时器 2 工作。因此，HALT 模式需要小心不要使定时器溢出。

第十二章 实时输出功能 (RTO)

12.1 功能

实时输出功能 (RTO) 发送预先设置的数据到寄存器 RTBLO 和 RTBH0, 然后在产生定时器中断时由硬件通过实时输出锁存发送这些数据到外部设备。发送这些数据到外部设备的引脚组成了实时输出功能 (RTO) 端口。

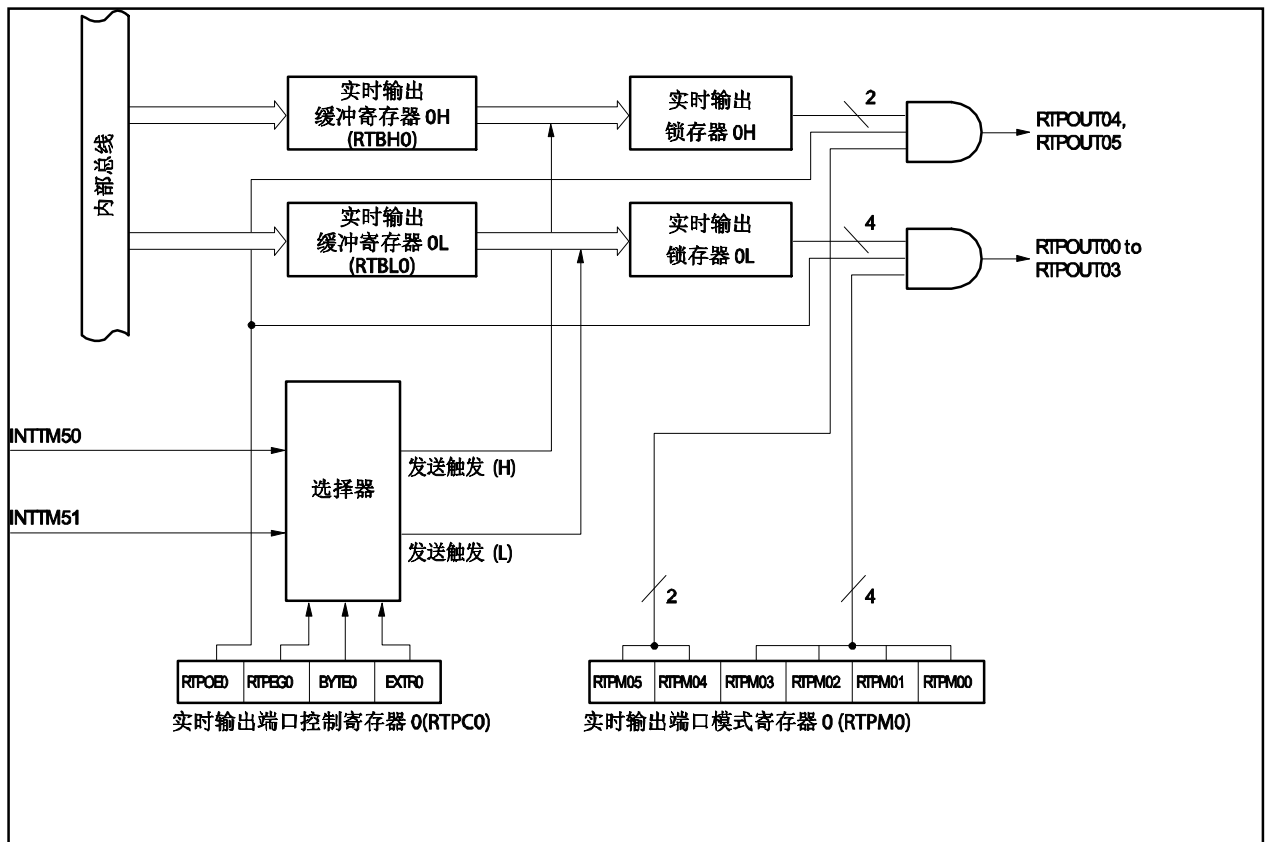
由于 RTO 可以稳定的输出信号, 所以适用于控制步进电机。

V850ES/KE2 提供一个 6 位的实时输出端口通道。

可由 1 位存储器操作指令设置实时输出端口工作在普通端口模式或实时输出端口模式。

RTO 的框图如下。

图 12-1. RTO 的框图



12.2 配置

RTO 包含以下硬件。

表 12-1. RTO 的配置

项目	配置
寄存器	实时输出缓冲寄存器 0 (RTBL0, RTBH0)
控制寄存器	实时输出端口模式寄存器 0 (RTPM0) 实时输出端口控制寄存器 0 (RTPC0)

(1) 实时输出缓冲寄存器 0 (RTBL0, RTBH0)

寄存器 RTBL0 和 RTBH0 是 4 位寄存器，用于保持预设输出数据。

该寄存器被映射到外围 I/O 寄存器区域的独立地址。

该寄存器支持 8 位或 1 位读写方式。

如果指定了操作模式为 4 位 × 1 通道或 2 位 × 1 通道 (RTPC0.BYTE0 位 = 0)，数据可被分别设置到寄存器 RTBL0 和 RTBH0。通过指定这两个寄存器中任意一个的地址，这两个寄存器的数据可被立刻读取。

如果指定了操作模式为 6 位 × 1 通道 (BYTE0 位 = 1)，8 位长的数据可被设置到寄存器 RTBL0 和 RTBH0 中，写入这两个寄存器中的任意一个。另外，通过指定这两个寄存器中的任意一个的地址，这两个寄存器中的数据可被立刻读取。

表 12-2 展示了操作寄存器 RTBL0 和 RTBH0 的过程。

复位后: 00H		R/W		地址:		RTBL0 FFFF6E0H, RTBH0 FFFF6E2H			
		7	6	5	4	3	2	1	0
RTBL0						RTBL03	RTBL02	RTBL01	RTBL00
RTBH0		0	0	RTBH05	RTBH04				

注意事项

1. 当向寄存器 RTBH0 的第 6、7 位进行写操作时，写入的数据恒为 0。
2. 当主时钟停止，CPU 使用副时钟进行操作时，禁止访问寄存器 RTBL0 和 RTBH0。关于更多细节，敬请参阅 3.4.8 (2)。

表 12-2. 寄存器 RTBL0 和 RTBH0 的操作

操作模式	被操作的寄存器	读		写 ^注	
		高 4 位	低 4 位	高 4 位	低 4 位
4 位 × 1 通道, 2 位 × 1 通道	RTBL0	RTBH0	RTBL0	无效	RTBL0
	RTBH0	RTBH0	RTBL0	RTBH0	无效
6 位 × 1 通道	RTBL0	RTBH0	RTBL0	RTBH0	RTBL0
	RTBH0	RTBH0	RTBL0	RTBH0	RTBL0

注 设置实时输出端口后，要在产生实时输出触发之前，将欲输出的数据设置到寄存器 RTBL0 和 RTBH0。

12.3 寄存器

RTO 由下面两种类型的寄存器控制。

- 实时输出端口模式寄存器 0 (RTPM0)
- 实时输出端口控制寄存器 0 (RTPC0)

(1) 实时输出端口模式寄存器 0 (RTPM0)

该寄存器选择实时输出端口模式或普通端口模式，可由 1 位存储器操作指令完成操作。

RTPM0 寄存器支持 8 位或 1 位读写方式。

RTPM0 寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF6E4H

	7	6	5	4	3	2	1	0
RTPM0	0	0	RTPM05	RTPM04	RTPM03	RTPM02	RTPM01	RTPM00

RTPM0m	实时输出端口的控制 (m = 0 到 5)
0	禁止实时输出
1	容许实时输出

- 注意事项**
1. 为了将实时输出信号 (RTPOUT00 到 RTPOUT05) 从引脚 (RTP00 到 RTP05) 输出，需要通过 PMC5 和 PFC5 寄存器将这些引脚设置为实时输出端口。
 2. 通过允许时输出操作 (RTPC0.RTPOE0 位 = 1)，允许实时输出的那些位将执行实时输出操作，禁止实时输出的那些位将输出 0。
 3. 如果禁止实时输出 (RTPOE0 位 = 0)，实时输出信号 (RTPOUT00 到 RTPOUT05) 全部输出 0，无论 RTPM0 寄存器如何设置。

(2) 实时输出端口控制寄存器 0 (RTPC0)

RTPC0 寄存器用于设置实时输出端口的操作模式和输出触发模式。

实时输出端口的操作模式和输出触发模式的关系如表 12-3 所示。

RTPC0 寄存器支持 8 位或 1 位读写方式。

RTPC0 寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF6E5H

<7>	6	5	4	3	2	1	0
RTPC0	RTPOE0	RTPEG0 ^{注1}	BYTE0	EXTR0 ^{注2}	0	0	0

RTPOE0	实时输出操作的控制
0	禁止操作 ^{注3}
1	容许操作

BYTE0	实时输出通道配置规格
0	4 位 '1 通道, 2 位 '1 通道
1	6 位 '1 通道

注

1. RTPEG0 位的值不影响操作。
2. 关于 EXTR0, 参考表 12-3。
3. 当禁止实时输出操作 (RTPOE0 位 = 0) 时, 实时输出信号的所有位 (RTPOUT00 到 RTPOUT05) 输出“0”。

注意事项 只有当 RTPOE0 = 0 时, 设置 BYTE0 和 EXTR0 位。

表 12-3. 实时输出端口的操作模式和输出触发

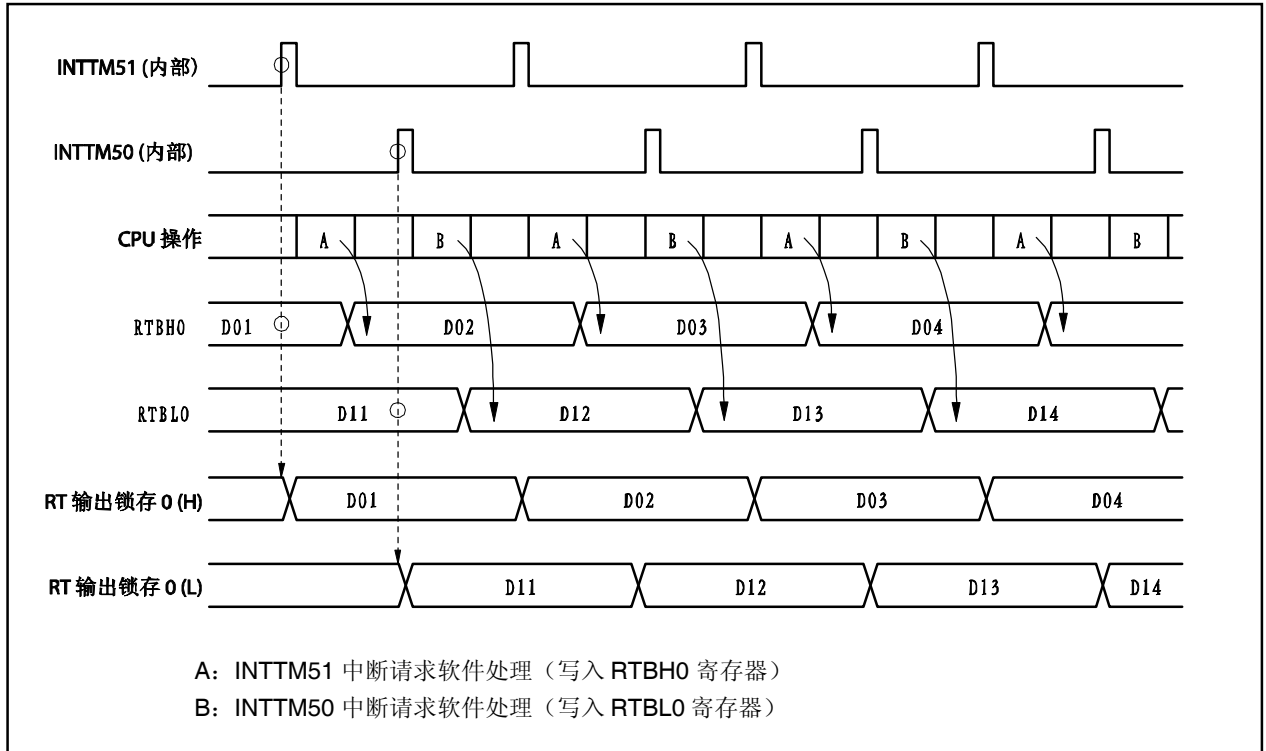
BYTE0	EXTR0	操作模式	RTBH0 (RTP04, RTP05)	RTBL0 (RTP00 到 RTP03)
0	0	4 位 × 1 通道,	INTTM51	INTTM50
	1	2 位 × 1 通道	INTTM50	没有触发
1	0	6 位 × 1 通道	INTTM50	
	1		禁止设置	

12.4 操作

如果设置 `RTPC0.RTPOE0` 为 1, 实时输出操作被允许, 寄存器 `RTBH0` 和 `RTBL0` 的数据将与产生所选择的发送触发 (由 `RTPC0.EXTR0` 和 `RTPC0.BYTE0` 设置) 同步被发送到实时输出锁存器。对于被发送的数据, 只有那些由寄存器 `RTPM0` 设定的允许实时输出的位才从 `RTPOUT00` 到 `RTPOUT05` 位输出。寄存器 `RTPM0` 设定的禁止实时输出的位输出为 0。

如果通过将 `RTPOE0` 清零的方法禁止实时输出操作, 那么无论寄存器 `RTPM0` 如何设置, 信号 `RTPOUT00` 到 `RTPOUT05` 的输出都为 0。

图 12-2. RTO0 操作时序的示例 (当 `EXTR0` 位 = 00, `BYTE0` 位 = 00)



备注 待机模式下的操作, 参见第十九章 待机功能。

12.5 用法

- (1) 禁止实时输出。
RTPC0.RTPOE0 清零。
- (2) 初始化如下。
 - 可由 1 位存储器操作指令设置实时输出口模式或普通端口模式。
设置寄存器 RTPM0。
 - 通道配置：选择触发和有效沿。
设置 RTPC0.EXTR0, RTPC0.BYTE0 和 RTPC0.RTPEG0 位。
 - 设置寄存器 RTBH0 和 RTBL0^{注1} 的初始值。
- (3) 允许实时输出。
设置 RTPOE0 = 1。
- (4) 在产生所选择的发送触发^{注2}时，设置寄存器 RTBH0 和 RTBL0 的下一输出值。
- (5) 在产生所选择的发送触发相应的中断服务中，设置寄存器 RTBH0 和 RTBL0 的下一输出值。

- 注**
1. 如果当 RTPOE0 = 0 时写寄存器 RTBH0 和 RTBL0，数据被分别发送到实时输出锁存 0H 和 0L。
 2. 即使当 RTPOE0 = 1 时写寄存器 RTBH0 和 RTBL0，数据也不被发送到实时输出锁存 0H 和 0L。

注意事项 为了将实时输出信号 (RTPOUT00 到 RTPOUT05) 从引脚输出，需要通过 PMC5 和 PFC5 寄存器设置实时输出口 (RTP00 到 RTP05)。

12.6 注意事项

- (1) 通过软件避免下述冲突。
 - 实时输出允许/禁止转换 (RTPOE0) 和所选择的实时输出触发之间的冲突。
 - 实时输出允许状态下写寄存器 RTBH0 和 RTBL0，与所选择的实时输出触发之间的冲突。
- (2) 在操作初始化前，禁止实时输出 (RTPOE0 位=0)。
- (3) 一旦实时输出被禁止 (RTPOE0 = 0)，确保在再次允许实时输出 (RTPOE0 = 0 → 1) 前初始化寄存器 RTBH0 和 RTBL0。

12.7 安全功能

芯片提供了安全功能电路，在 RTO 控制的步进电机发生故障时，该电路可设置引脚输出高阻。它通过外部中断 INTPO 引脚边沿检测，强制将分配给 RTP00~RTP05 的引脚复位，使它们处于高阻状态。

被 INTPO^{注1} 引脚置为高阻的端口 (P50~P55 引脚) 将初始化^{注2}，因此必须重新设置这些端口。

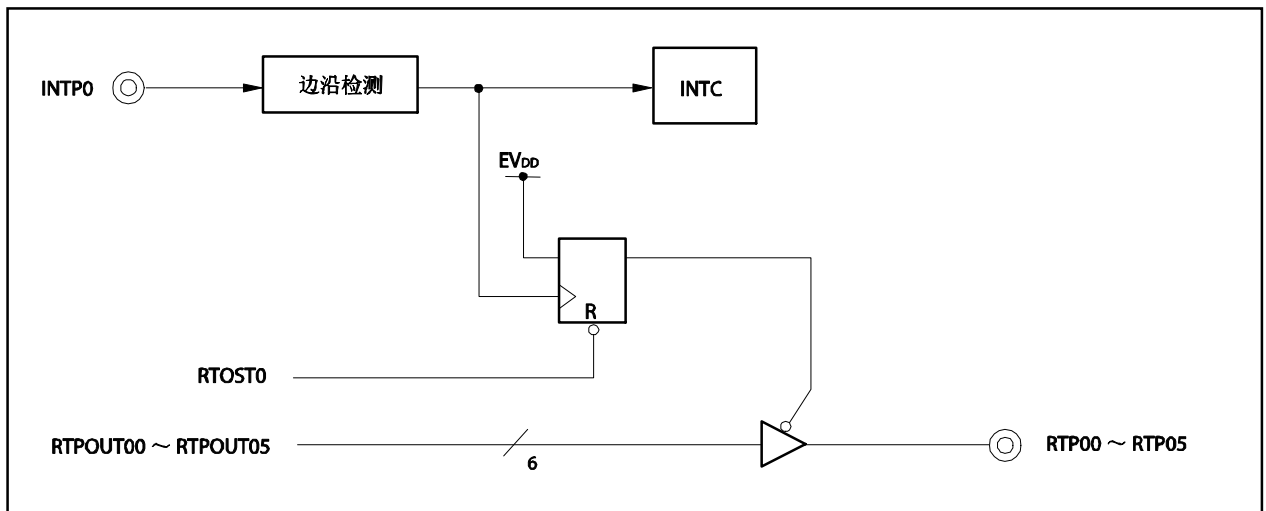
注 1. 无论端口设置如何，P50~P55 引脚将全部通过 INTPO 引脚置为高阻。

2. 以下寄存器中与 P50~P55 引脚相关的所有位初始化。

- P5 寄存器
- PM5 寄存器
- PMC5 寄存器
- PU5 寄存器
- PFC5 寄存器

安全功能的框图如下。

图 12-3.安全功能框图



该功能由 PLLCTL.RTOST0 位设置。

(1) PLL 控制寄存器 (PLLCTL)

PLLCTL 寄存器是 8 位寄存器，控制 RTO 安全功能和 PLL。

该寄存器支持 8 位或 1 位读写方式。

PLLCTL 寄存器复位后的值为 01H。

复位后: 01H R/W 地址: FFFF806H

	7	6	5	4	3	<2>	<1>	<0>
PLLCTL	0	0	0	0	0	RTOST0	SELPLL ^注	PLLON ^注

RTOST0	RTP00到RTP05 安全功能的控制
0	INTP0 引脚不作安全功能使用
1	INTP0 引脚作安全功能使用

注 关于 SELPLL 和 PLLON 位的详细情况，参考第五章 时钟产生功能。

- 注意事项
1. 从实时输出口 (RTP00~RTP05) 输出数值前，选择 INTP0 引脚中断边沿检测，并设置 RTOST0 位。
 2. 通过 INTP0 引脚将 P50~P55 引脚置为高阻之后，需要先取消安全功能，再重新将端口 (P50~P55 引脚) 设置为实时输出口。

[重新设置端口步骤]

- <1> 取消安全功能，并通过将 RTOST0 位清零禁止端口设置。
 - <2> 将 RTOST0 位置 1 (仅在必要时)。
 - <3> 重新设置为实时输出口。
3. 务必将 4 至 7 位清零。改变第 3 位的值不影响操作。

第十三章 A/D 转换器

13.1 概述

A/D 转换器将模拟输入信号转换成数字值，有 8 个通道（AN10 ~ AN17）。

A/D 转换器有以下功能。

- 工作电压（ AV_{REF0} ）：2.7~5.5 V
- 逐次逼近法 10 位 A/D 转换器
- 模拟输入引脚数：8
- 触发模式：
 - 软件触发模式
 - 定时器触发模式（INTTM010）
 - 外部触发模式（ADTRG 引脚）
- 操作模式
 - 选择模式
 - 扫描模式
- A/D 转换时间：
 - 标准模式：
 - 14 ~ 100 μ S @ $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$
 - 17 ~ 100 μ S @ $2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$
 - 高速模式：
 - 3 ~ 100 μ S @ $4.5\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$
 - 4.8 ~ 100 μ S @ $4.0\text{ V} \leq AV_{REF0} < 4.5\text{ V}$
 - 6 ~ 100 μ S @ $2.85\text{ V} \leq AV_{REF0} < 4.0\text{ V}$
 - 14 ~ 100 μ S @ $2.7\text{ V} \leq AV_{REF0} < 2.85\text{ V}$
- 掉电检测功能

注意事项 使用 A/D 转换器时， AV_{REF0} 需要与 V_{DD} 和 EV_{DD} 处于同一电势。

13.2 功能

(1) 10 位分辨率 A/D 转换

从 AN10 ~ AN17 引脚选择一个模拟通道，A/D 转换操作以 10 位分辨率重复进行。每当 A/D 转换结束，都产生一个中断请求信号（INTAD）。

(2) 掉电检测功能

该功能用于检测电池电压低的情况。A/D 转换的结果（寄存器 ADCRH 的值）与寄存器 PFT 的值进行比较，只有当指定的比较条件相等时，才产生中断信号 INTAD。

13.3 配置

A/D 转换器包含以下硬件。

图 13-1. A/D 转换器的框图

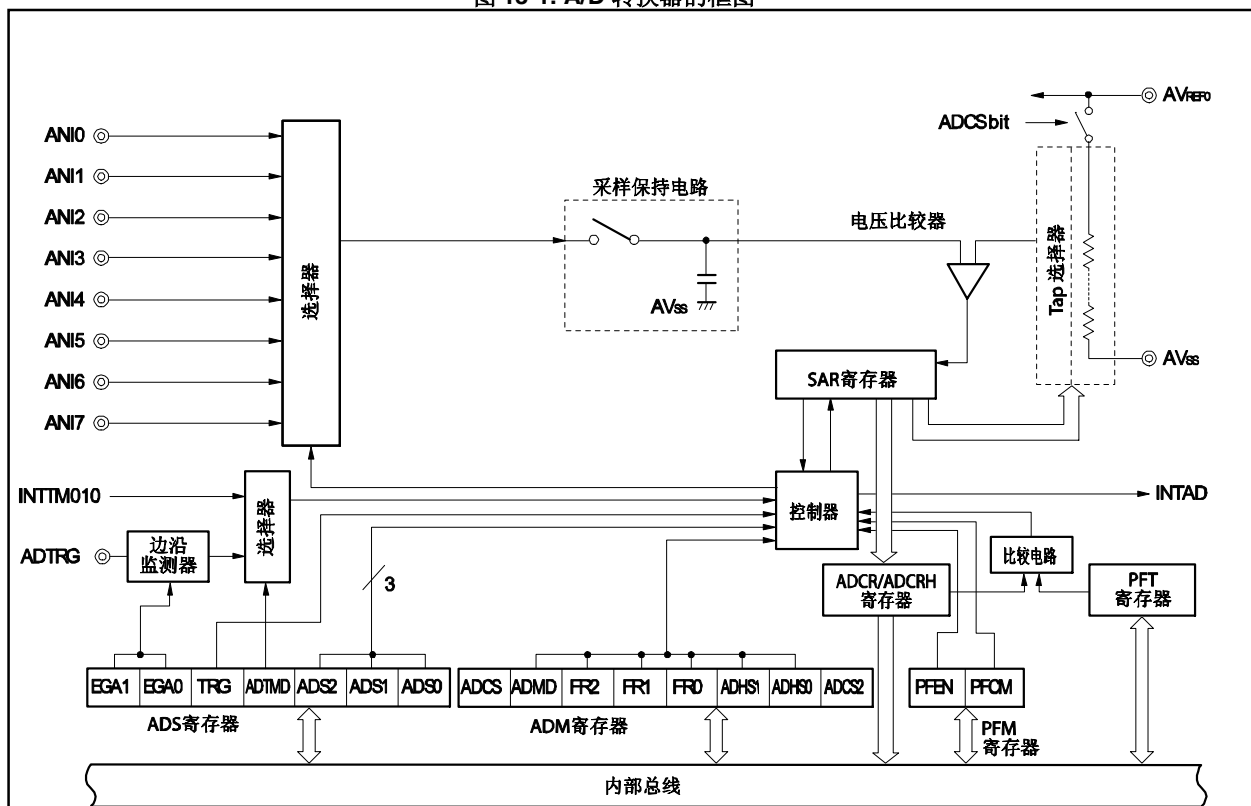


表 13-1. 软件中使用的 A/D 转换器寄存器

项目	配置
寄存器	A/D 转换结果寄存器 (ADCR) A/D 转换结果寄存器 H (ADCRH)：仅可读高 8 位 掉电比较阈值寄存器 (PFT) A/D 转换器模式寄存器 (ADM) 模拟输入通道指定寄存器 (ADS) 掉电比较模式寄存器 (PFM)

(1) ANI0~ANI7 引脚

ANI0~ ANI7 是 8 个 A/D 转换器通道模拟输入的引脚，用于输入待转换成数字信号的模拟信号。没有被寄存器 ADS 选定作为模拟输入的引脚可以用于普通输入端口。

(2) 采样与保持电路

采样与保持电路采集输入电路选择的模拟输入信号，发送采样数据到电压比较器。该电路在 A/D 转换期间保持被采样的模拟输入信号的电压。

(3) 串联电阻串

串联电阻串连接 AV_{REF0} 和 AV_{SS} 并产生一个与模拟输入信号相比较的电压。

(4) 电压比较器

电压比较器对被采样保持的电压值及串联电阻串的输出电压值进行比较。

(5) 逐次逼近寄存器 (SAR)

寄存器将采样的模拟输入电压值与串联电阻串的电压值进行比较，保持比较结果。比较结果从最高有效位 (MSB) 开始。

当比较结果被致以最低有效位 (LSB) (也就是说，当 A/D 转换完全)，寄存器 SAR 的结果被传送到寄存器 ADCR。

SAR 寄存器的值不可直接读写。

(6) A/D 转换结果寄存器 (ADCR)，A/D 转换结果寄存器 H (ADCRH)

每次当 A/D 转换结束，转换结果从逐次逼近寄存器载入并保存在该寄存器的高 10 位 (低 6 位恒为 0)。

(7) 控制器

当 A/D 转换完成或使用掉电监测功能时，控制器将 A/D 转换结果 (ADCRH 寄存器的值) 与 PFT 的值进行比较，只有当指定的比较条件满足时，才产生中断信号 INTAD。

(8) AV_{REF0} 引脚

该引脚用于输入 A/D 转换器的模拟电源/参考电压。即使没有使用 A/D 转换器，也应保持该引脚电压与 V_{DD} 引脚电势相同。

输入到引脚 ANI0 ~ ANI7 的信号基于引脚 AV_{REF0} 和 AV_{SS} 之间的电压被转换成数字信号。

(9) AV_{SS} 引脚

这是 A/D 转换器的信号地电位引脚。即便没有使用 A/D 转换器，该引脚的电压也应与 V_{SS} 的电压保持一致。

(10) A/D 转换模式寄存器 (ADM)

该寄存器设置模拟输入转换成数字信号的转换时间和转换操作的启动和停止。

(11) 模拟输入通道指定寄存器 (ADS)

该寄存器指定待转换为数字信号的模拟电压的输入端口。

(12) 掉电比较模式寄存器 (PFM)

该寄存器设置掉电检测模式。

(13) 掉电比较阈值寄存器 (PFT)

该寄存器设置与 ADCR 寄存器比较的阈值。

13.4 寄存器

A/D 转换器使用以下寄存器进行控制。

- A/D 转换器模式寄存器 (ADM)
- 模拟输入通道指定寄存器 (ADS)
- 掉电比较模式寄存器 (PFM)
- 掉电比较阈值寄存器 (PFT)
- A/D 转换结果寄存器, A/D 转换结果寄存器 H (ADCR, ADCRH)

(1) A/D 转换模式寄存器 (ADM)

该寄存器设置模拟信号到数字信号的转换时间, 以及转换的启动和停止。

ADM 寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

表 13-2. A/D 转换时间

复位后: 00H R/W 地址: FFFF200H

	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	ADMD	FR2 ^{注1}	FR1 ^{注1}	FR0 ^{注1}	ADHS1 ^{注1}	ADHS0 ^{注1}	ADCS2
ADCS	A/D转换操作的控制							
0	停止转换操作							
1	容许转换操作							
ADMD	操作模式控制							
0	选择模式							
1	扫描模式							
ADHS1	5V A/D转换时间模式的选择 (AV _{REF0} ³ 4.5 V)							
0	正常模式							
1	高速模式 (仅在AV _{REF0} ³ 4.5 V时有效)							
ADHS0	3V A/D转换时间模式的选择 (AV _{REF0} ³ 2.7 或 2.85 V)							
0	正常模式							
1	高速模式 (仅在 AV _{REF0} ³ 2.7 或 2.85 V时有效)							
ADCS2	参考电压生成器增压的控制 ^{注2}							
0	停止参考电压生成器							
1	容许参考电压生成器模式							

注 1. 关于 FR2~FR0 位及 A/D 转换, 敬请参阅表 13-2 A/D 转换时间。

- 用于升压的参考电压生成器的操作由 ADCS 位控制, 在操作开始后需 1 μ S (高速模式) 或 14 μ S (标准模式) 才能使其稳定。因此在 ADCS2 位设置为 1 (用于升压的参考电压生成器开启) 后至少 1 μ S (高速模式) 或 14 μ S (标准模式), ADCS2 位才设置为 1 (A/D 转换开始), 第一个转换结果有效。

- <R> 注意事项
- 标准模式下 (ADHS1, ADHS0 位= 00), 在 A/D 转换操作 (ADCS 位=1) 中, 禁止写入 ADM 寄存器。
高速模式下 (ADHS1, ADHS0 位= 10 或 01), 在 A/D 转换操作时, 如在 ADM 中写入同值, 转换中止并重新启动。在 A/D 操作中严禁写入 FR2 ~ FR0, ADHS1, 及 ADHS0 位。
 - 禁止设置 ADHS1 及 ADHS0 位为 11。
 - 在主时钟停止, 副时钟操作时, 禁止访问 ADM 寄存器。关于更多细节, 敬请参阅 3.4.8 (2) 访问特殊片上外围 I/O 寄存器。

表 13-2. A/D 转换时间

ADHS1	ADHS0	FR2	FR1	FR0	A/D 转换时间 (μs)				转换时间模式	
					20 MHz@ AV _{REF0} ≥ 4.5 V	16 MHz@ AV _{REF0} ≥ 4.0 V	8 MHz@ AV _{REF0} ≥ 2.85 V	8 MHz@ AV _{REF0} ≥ 2.7 V		
0	0	0	0	0	288/f _{xx}	14.4	18.0	36.0	36.0	标准模式 AV _{REF0} ≥ 2.7 V
0	0	0	0	1	240/f _{xx}	禁止设置	15.0	30.0	30.0	
0	0	0	1	0	192/f _{xx}	禁止设置	禁止设置	24.0	24.0	
0	0	0	1	1	禁止设置					
0	0	1	0	0	144/f _{xx}	禁止设置	禁止设置	18.0	18.0	标准模式 AV _{REF0} ≥ 2.7 V
0	0	1	0	1	120/f _{xx}	禁止设置	禁止设置	禁止设置	禁止设置	
0	0	1	1	0	96/f _{xx}	禁止设置	禁止设置	禁止设置	禁止设置	
0	0	1	1	1	禁止设置					
0	1	0	0	0	96/f _{xx}	4.8	6.0	12.0	禁止设置	高速模式 AV _{REF0} ≥ 2.85 V
0	1	0	0	1	72/f _{xx}	禁止设置	禁止设置	9.0		
0	1	0	1	0	48/f _{xx}	禁止设置	禁止设置	6.0	禁止设置	
0	1	0	1	1	24/f _{xx}	禁止设置	禁止设置	禁止设置	禁止设置	
0	1	1	0	0	224/f _{xx}	11.2	14.0	28.0	28.0	高速模式 AV _{REF0} ≥ 2.7 V
0	1	1	0	1	168/f _{xx}	禁止设置	10.5	21.0	21.0	
0	1	1	1	0	112/f _{xx}	禁止设置	禁止设置	禁止设置	禁止设置	
0	1	1	1	1	56/f _{xx}	禁止设置	禁止设置	禁止设置	禁止设置	
1	0	0	0	0	72/f _{xx}	3.6	禁止设置	禁止设置	禁止设置	高速模式 AV _{REF0} ≥ 4.5 V
1	0	0	0	1	54/f _{xx}	禁止设置	禁止设置	禁止设置	禁止设置	
1	0	0	1	0	36/f _{xx}	禁止设置	禁止设置	禁止设置	禁止设置	
1	0	0	1	1	18/f _{xx}	禁止设置	禁止设置	禁止设置	禁止设置	
1	0	1	×	×	禁止设置					
1	1	×	×	×	禁止设置					

(a) 控制用于升压的参考电压生成器

当 ADCS2 位=0, A/D 转换器的电压下降。在 ADCS2 位设置为 1 后, 转换器需 1 μ s (高速模式) 或 14 μ s (正常模式) 或更多的建立时间。

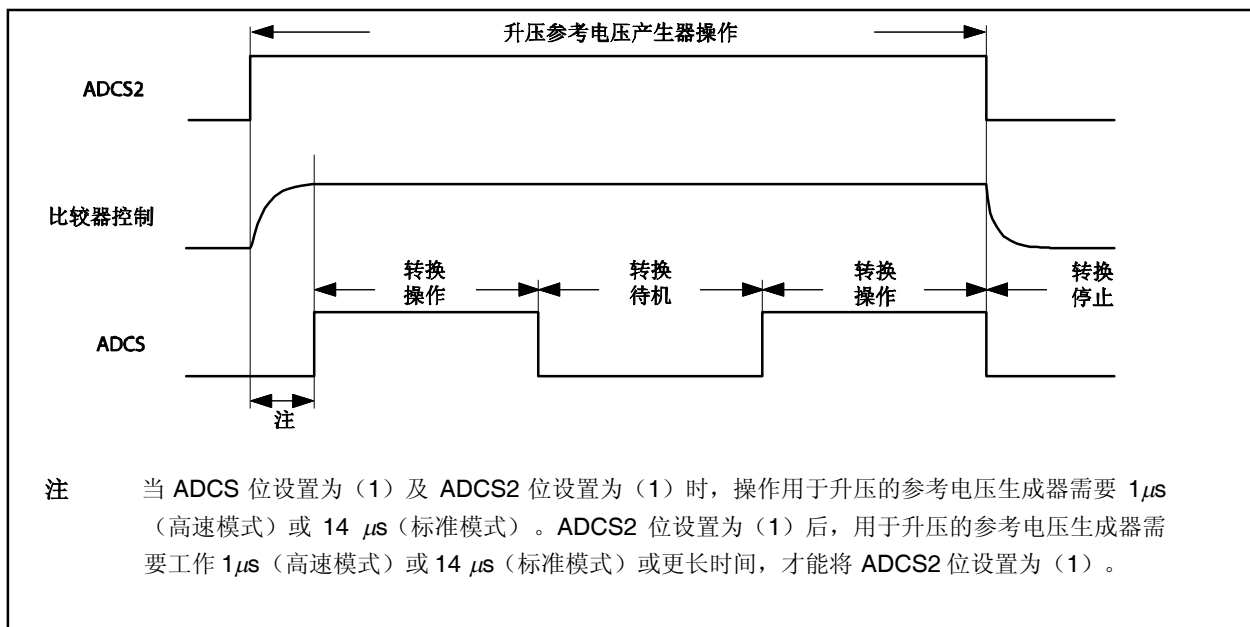
因此, 在 ADCS2 位设为 1 至少 1 μ s (高速模式) 或 14 μ s (正常模式) 后, 通过设置 ADCS 位为 1, A/D 转换结果才变为有效。

表 13-3. 设置 ADCS 位及 ADCS2 位

ADCS	ADCS2	A/D 转换操作
0	0	停止状态 (DC 电源耗电通道不存在)
0	1	转换等待模式 (只有用于提高电压的参考电压生成器耗电)
1	0	转换模式 (参考电压生成器停止操作 ^{#1})
1	1	转换模式 (进行参考电压生成器操作 reference ^{#2})

- 注 1. 如果 ADCS 及 ADCS2 位从 00B 变为 10B 用于提升, 参考电压生成器自动开启。如果 ADCS 位清零而 ADCS2 为 0, 电压生成器自动关闭。在软件触发模式下 (ADS.TRG 位 = 0), 严禁使用第一个 A/D 转换结果。
在硬件触发模式下 (TRG 位=1), 只有在用于升压的参考电压生成器的振稳定时间结束后启动 A/D 转换时, 才可使用 A/D 转换结果。
2. 如果 ADCS 及 ADCS2 位从 00B 变为 11B, 用于升压的参考电压生成器自动开启。如果 ADCS 位清零而 ADCS2 为 1, 电压生成器保持开启。在软件触发模式下 (TRG 位 = 0), 禁止使用 A/D 转换结果。
在硬件触发模式下 (TRG 位 = 1), 只有在用于升压的参考电压生成器的振稳定时间结束后启动 A/D 转换时, 才可使用 A/D 转换结果。

图 13-2. 操作顺序



(2) 模拟输入通道指定寄存器 (ADS)

该寄存器指定了 A/D 转换的模拟电压输入端口。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF201H

	7	6	5	4	3	2	1	0
ADS	EGA1 ^{注1}	EGA0 ^{注1}	TRG	ADTMD ^{注2}	0	ADS2	ADS1	ADS0

EGA1 ^{注1}	EGA0 ^{注1}	外部触发信号 (ADTRG) 边沿规格
0	0	没检测到边沿
0	1	降沿
1	0	升沿
1	1	双沿

TRG	触发模式选择
0	软件触发模式
1	硬件触发模式

ADTMD ^{Note2}	硬件触发模式规格 ^h
0	外部触发 (ADTRG 硬件输入)
1	定时器触发 (INTTM010 信号生成)

ADS2	ADS1	ADS0	模拟输入通道规格	
			选择模式	扫描模式
0	0	0	ANI0	ANI0
0	0	1	ANI1	ANI0, ANI1
0	1	0	ANI2	ANI0 to ANI2
0	1	1	ANI3	ANI0 to ANI3
1	0	0	ANI4	ANI0 to ANI4
1	0	1	ANI5	ANI0 to ANI5
1	1	0	ANI6	ANI0 to ANI6
1	1	1	ANI7	ANI0 to ANI7

- 注
1. EGA1 及 EGA0 位仅在硬件触发模式 (TRG 位 = 1) 及外部触发模式下 (ADTRG 引脚输入 ADTMD 位 = 1) 有效。
 2. ADTMD 位仅在硬件触发模式下有效 (TRG 位 = 1)。

- <R> 注意事项
1. 标准模式下 (ADM.ADHS1, ADM.ADHS0 位 = 00) 在 A/D 转换操作中 (ADM.ADCS 位 = 1), 禁止写入 ADS 寄存器。
 2. 标准模式下 ADHS1, ADHS0 位 = 00) 在 A/D 转换操作中 (ADCS 位 = 1), 禁止冗余输入软/硬件触发。
 3. 在主时钟停止副时钟运行时禁止访问 ADS 寄存器。敬请参阅 3.4.8 (2) 访问特殊片上外围 I/O 寄存器。
 4. 一定要将位 3 清至“0”。

(3) A/D 转换结果寄存器，A/D 转换结果寄存器 H (ADCR, ADCRH)

ADCR 及 ADCRH 寄存器存储 A/D 转换结果。

该寄存器是只读的，可由 16 位或 8 位存储器操作指令进行读写。但是，由 16 位存储器操作指令访问寄存器 ADCR，8 位存储器操作指令访问寄存器 ADCRH。寄存器 ADCR 的高 10 位读取 10 位转换结果，其低 6 位的读取值为 0。ADCR 寄存器读取转换结果的高 8 位。

这些寄存器复位后的值未定义。

复位后: 未定 R 地址: FFFFF204H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCR	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

复位后: 未定 R 地址: FFFFF205H

	7	6	5	4	3	2	1	0
ADCRH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

注意事项 在主时钟停止副时钟运行时禁止访问 ADCR 和 ADCRH 寄存器。敬请参阅 3.4.8 (2) 访问特殊片上外围 I/O 寄存器。

如下为输入到模拟输入引脚的模拟输入电压（ANI0 ~ ANI17）和 A/D 转换结果（ADCR 寄存器）的关系。

$$SAR = INT \left(\frac{V_{IN}}{AV_{REF0}} \times 1024 + 0.5 \right)$$

$$ADCR^{注} = SAR \times 64$$

或

$$(SAR - 0.5) \times \frac{AV_{REF0}}{1024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{REF0}}{1024}$$

INT ()：该函数返回括号中值的整数部分。

V_{IN} ：模拟输入电压

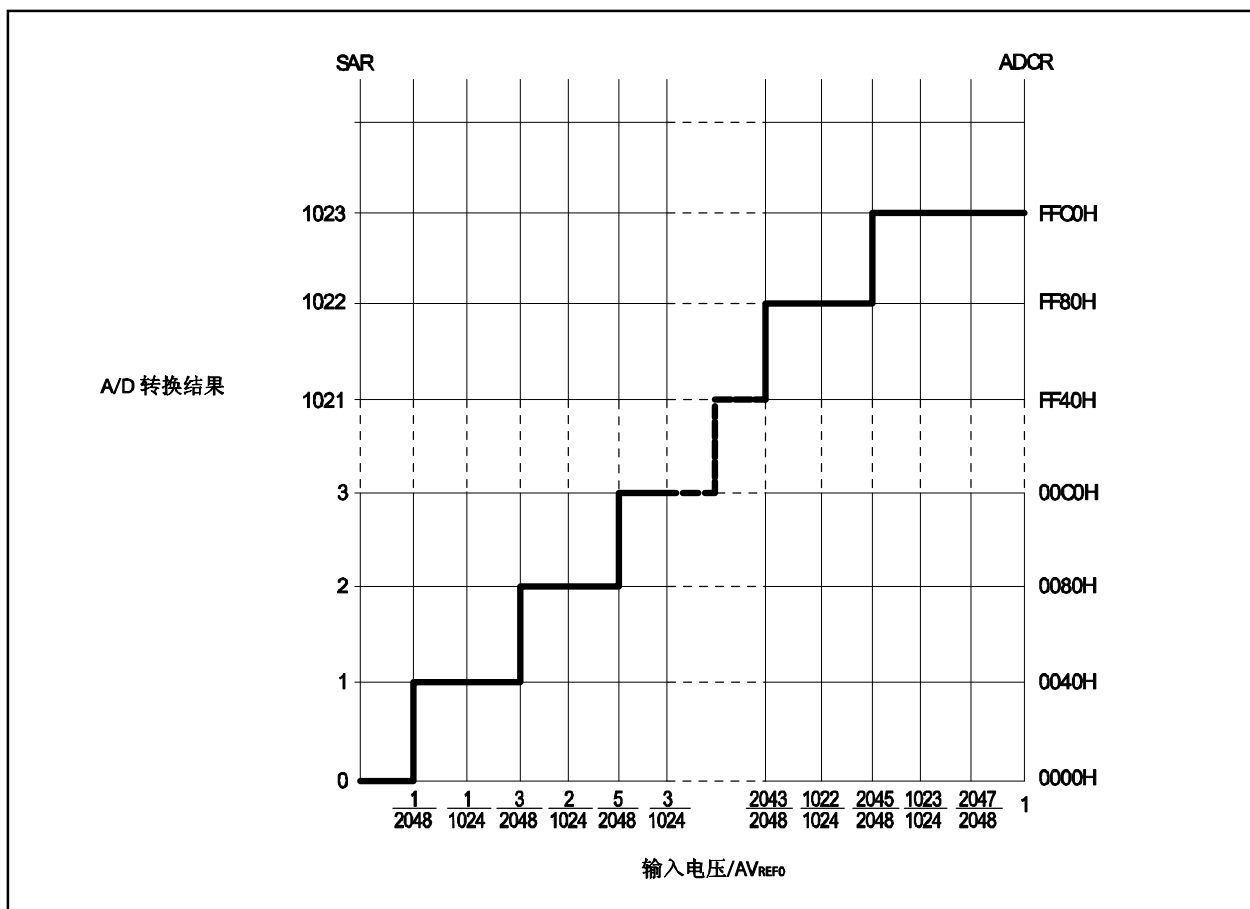
AV_{REF0} ： AV_{REF0} 引脚电压

ADCR：寄存器 ADCR 的值

注 寄存器 ADCR 的低 6 位恒为 0。

如下显示了模拟输入电压和 A/D 转换结果之间的关系。

图 13-3. 模拟输入电压和 A/D 转换结果之间的关系



(4) 掉电比较模式寄存器 (PFM)

该寄存器设置掉电检测模式。

PFM 寄存器用于比较寄存器 PFT 值及寄存器 ADCRH 值。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF202H

	<7>	<6>	5	4	3	2	1	0
PFM	PFEN	PFCM	0	0	0	0	0	0

PFEN	电源故障对比容许/禁止的选择
0	禁止电源故障对比
1	容许电源故障对比

PFCM	电源故障对比模式选择
0	在 ADCR ³ PFT 下生成中断请求信号 (INTAD)
1	当 ADCR < PFT 时生成中断请求信号 (INTAD)

- <R>
- 注意事项 1. 标准模式下 (ADM.ADHS1, ADM.ADHS0 位 = 00) 在 A/D 转换操作中 (ADM.ADCS 位 = 1), 禁止写入 PFM 寄存器。
2. 在主时钟停止副时钟运行时禁止访问 PFM 寄存器。敬请参阅 3.4.8 (2) 访问特殊片上外围 I/O 寄存器。

(5) 掉电比较阈值寄存器 (PFT)

该寄存器用于设置掉电检测模式的对比值。

该寄存器的 8 位数据设置用于与 ADCRH 寄存器的值进行对比。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF203H

	7	6	5	4	3	2	1	0
PFT								

- <R>
- 注意事项 1. 标准模式下 (ADM.ADHS1, ADM.ADHS0 位 = 00) 在 A/D 转换操作中 (ADM.ADCS 位 = 1), 禁止写入 PFT 寄存器。
2. 在主时钟停止副时钟运行时禁止访问 PFT 寄存器。敬请参阅 3.4.8 (2) 访问特殊片上外围 I/O 寄存器。

13.5 操作

13.5.1 基本操作

- <1> 通过使用 ADS 寄存器选择其模拟信号待转换为数字信号的通道。
设置 ADM.ADHS1 或 ADM.ADHS0 位。
- <2> 设置 ADM.ADCS2 位为 1 和等待 1 μs （高速模式）或 14 μs （标准模式）或更长时间。
- <3> 设置 ADM.ADCS 位为 1 以启动 A/D 转换。
（硬件执行步骤<4>~<10>。）
- <4> 采样和保持电路对被选模拟输入通道的输入电压进行采样。
- <5> 在一段特定采样时间后，采样和保持电路进入保持状态，输入模拟电压一直保持到其被转换为数字信号。
- <6> 设置逐次逼近寄存器（SAR）的第 9 位为 1。分接选择器选择 $(1/2) \times AV_{\text{REF0}}$ 作为串联电阻串的分接电压。
- <7> 电压比较器将串联电阻串的电压和模拟输入电压的电压差进行比较。如果模拟输入电压高于 $(1/2) \times AV_{\text{REF0}}$ ，则 SAR 的 MSB=1。如果模拟输入电压低于 $(1/2) \times AV_{\text{REF0}}$ ，则 SAR 的 MSB = 0。
- <8> 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 位的所得值选择串联电阻串的分接电压，具体描述如下。
 - 第 9 位 = 1: $(3/4) AV_{\text{REF0}}$
 - 第 9 位 = 0: $(1/4) AV_{\text{REF0}}$
 比较分接电压与采样电压，根据结果，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 \leq 分接电压: 第 8 位= 0
- <9> 按此方式继续进行比较，直至 SAR 的第 0 位。
- <10> 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 中锁存。同时也会产生 A/D 转换结束中断请求信号（INTAD）。
- <11> 重复步骤<4> ~<10>直到 ADCS 位清零。
进行另外 A/D 转换，从步骤<3>开始，然而，在 ADCS2 位清为 0 时操作 A/D 转换器，从<2>步骤开始。

13.5.2 触发模式

V850ES/KE2 系列用以下 3 种触发模式来设置 A/D 转换的开启时间。这些触发模式由 ADS 寄存器设置。

- 软件触发模式
- 外部触发模式（硬件触发模式）
- 定时器触发模式（硬件触发模式）

(1) 软件触发模式

当 ADS.TRG 位为 0 时通过设置 ADM.ADCS 位为 1 该模式可启动 A/D 转换。

在 A/D 完成之后，如 ADCS 位不清零，转换将不断重复进行。

<R> 在高速模式下（ADM.ADHS1，ADM.ADHS0 位 = 01 或 10），如果在转换过程中对寄存器 ADM，ADS，PFM，或 PFT 进行写操作，转换中止并重新开始。标准模式下（ADHS1，ADHS0 位 = 00）在转换过程中，禁止对 ADM，ADS，PFM，及 PFT 寄存器进行写操作。

(2) 外部触发模式（硬件触发模式）

<R> 通过设置高速模式（ADHS1，ADHS0 位 = 10 或 01）来使用该模式。标准模式下（ADHS1，ADHS0 位 = 00）。在标准模式下 A/D 转换过程中，禁止输入有效边沿到 ADTRG 引脚。

在 ADCS 位设置为 1，TRG 位设置为 1，ADS.ADTMD 位设置为 0 后，通过检测外部触发（ADTRG），该模式用来启动 A/D 转换。

在 ADCS 位设置为 1 后，A/D 转换器等待外部触发（ADTRG）。

通过 ADS.EGA1 及 ADS.EGA0 位指定输入到 ADTRG 引脚的有效边沿信号。当检测到指定的有效边沿时，A/D 转换启动。

A/D 转换完成时，A/D 转换器再次等待外部触发（ADTRG）。

在高速模式下（ADHS1，ADHS0 位 = 01 或 10）A/D 转换过程中，如输入有效边沿到 ADTRG 引脚，A/D 转换中止并重新开始。

在高速模式下（ADHS1，ADHS0 位 = 01 或 10）A/D 转换过程中，如对 ADM，ADS，PFM，或 PFT 寄存器进行写操作，A/D 转换中止，等待外部触发（ADTRG）。

(3) 定时器触发模式（硬件触发模式）

<R> 通过设置高速模式（ADHS1，ADHS0 位 = 10 或 01）来使用该模式。标准模式下（ADHS1，ADHS0 位 = 00）。在 A/D 转换过程中，禁止输入有效边沿到 ADTRG 引脚。

在 ADCS 位设置为 1，TRG 及 ADTMD 位都被设置为 1 后，通过检测定时器触发（INTTM010），该模式用来启动 A/D 转换。

在 ADCS 位设置为 1 后，A/D 转换器等待定时器触发（INTTM010）。

A/D 转换在生成 INTTM010 信号时启动。

A/D 转换完成时，A/D 转换器再次等待定时器触发（INTTM010）。

在高速模式下（ADHS1，ADHS0 位 = 01 或 10）A/D 转换过程中，如生成 INTTM010 信号，A/D 转换中止并重新开始。

在高速模式下（ADHS1，ADHS0 位 = 01 或 10）A/D 转换过程中，如对 ADM，ADS，PFM，或 PFT 寄存器进行写操作，A/D 转换中止，等待定时器触发（INTTM010）。

13.5.3 操作模式

有 2 种操作可供使用。这些操作模式由 ADM 寄存器设置。

- 选择模式
- 扫描模式

(1) 选择模式

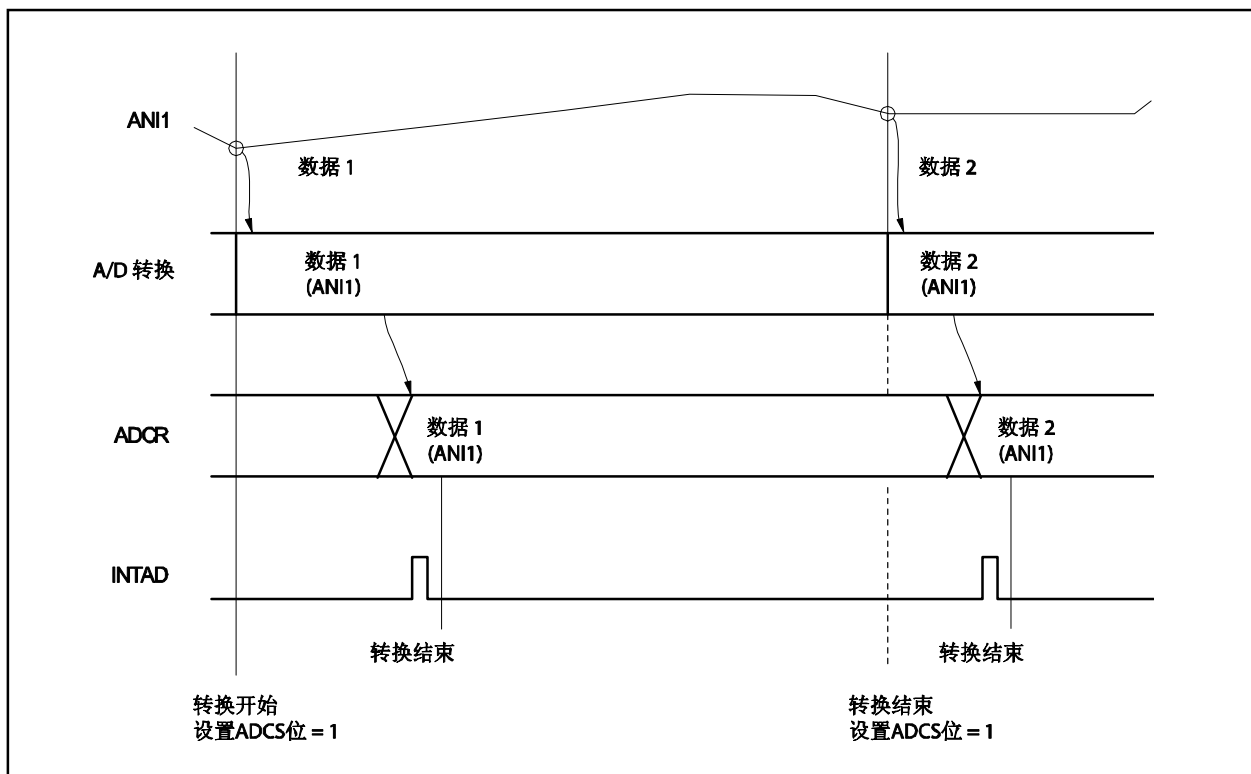
ADM.ADM位=0 时由 ADS 寄存器指定的输入模拟信号被转换。转换过程完成后，转换结果保存到 ADCR 寄存器。

同时，生成 A/D 转换结束中断请求信号 (INTAD)。然而，该信号能否生成取决于 PFM 及 PFT 寄存器的设置。敬请参阅 13.5.4 掉电检测功能。

<R> 在高速模式下 (ADM.ADHS1, ADHS1, ADHS0 位 = 01 或 10) A/D 转换过程中，如对 ADM, ADS, PFM 和 PFT 寄存器进行写操作，A/D 转换中止。在软件触发模式下，A/D 转换器重新开始。在硬件触发模式下，A/D 转换器等待触发。在标准模式下 (ADHS1, ADHS0 位 = 00) 在转换过程中，禁止对 ADM, ADS, PFM, 或 PFT 寄存器进行写操作。

<R> 在高速模式下 (ADHS1, ADHS0 位 = 01 或 10) 在转换过程中，如在硬件触发模式下检测到触发，A/D 转换中止，重新开始。在标准模式下 (ADHS1, ADHS0 位 = 00) A/D 转换过程中，禁止再次输入触发。

图 13-4. 选择模式操作时序示例 (ADS.ADS2 ~ ADS.ADS0 位 = 001B)



(2) 扫描模式

在该模式下，当 ADM.ADMD 位=1 时，从 ANI0 引脚输入的信号和由 ADS 寄存器指定的模拟信号连续不断的选择及转换。

当模拟输入信号转换过程完成，转换结果保存到 ADCR 寄存器。同时，生成 A/D 转换结束中断请求信号 (INTAD)。

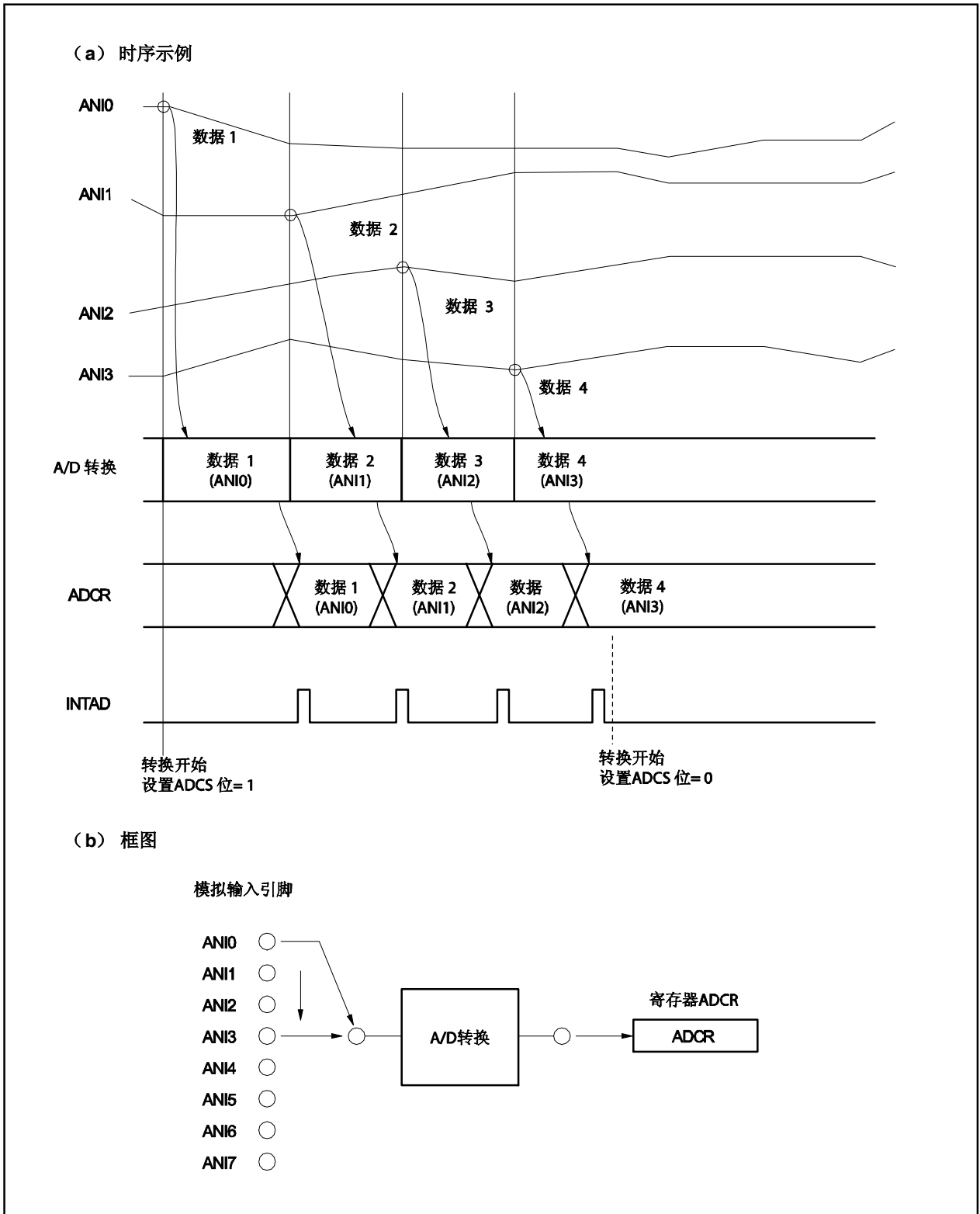
所有模拟输入信号的 A/D 转换结果都保存到 ADCR 寄存器。因此，一旦一个模拟输入信号 A/D 转换结束，建议将其保存到 RAM。

<R> 在硬件触发模式下 (ADS.TRG 位 = 1)，A/D 转换器转换完成后等待触发。在高速模式下 (ADM.ADHS1, ADM.ADHS0 位 = 01 或 10) A/D 转换过程中，如对 ADM, ADS, PFM 和 PFT 寄存器进行写操作，A/D 转换中止。在软件触发模式下，A/D 转换器重新开始。在硬件触发模式下，A/D 转换器等待触发。转换从 ANI0 引脚重新开始。在标准模式下 (ADHS1, ADHS0 位 = 00) 在转换过程中，禁止输入有效边沿到 ADTRG 引脚。

<R> 在高速模式下 (ADHS1, ADHS0 位 = 01 或 10) 在转换过程中，如在硬件触发模式下检测到触发，A/D 转换中止，重新开始 (ANI0 引脚)。在标准模式下 (ADHS1, ADHS0 位 = 00) A/D 转换过程中，禁止对 ADM, ADS, PFM, 或 PFT 寄存器进行写操作。

图 13-5. 扫描模式操作时序示例 (ADS.ADS2 ~ ADS.ADS0 位 = 011B)

<R>

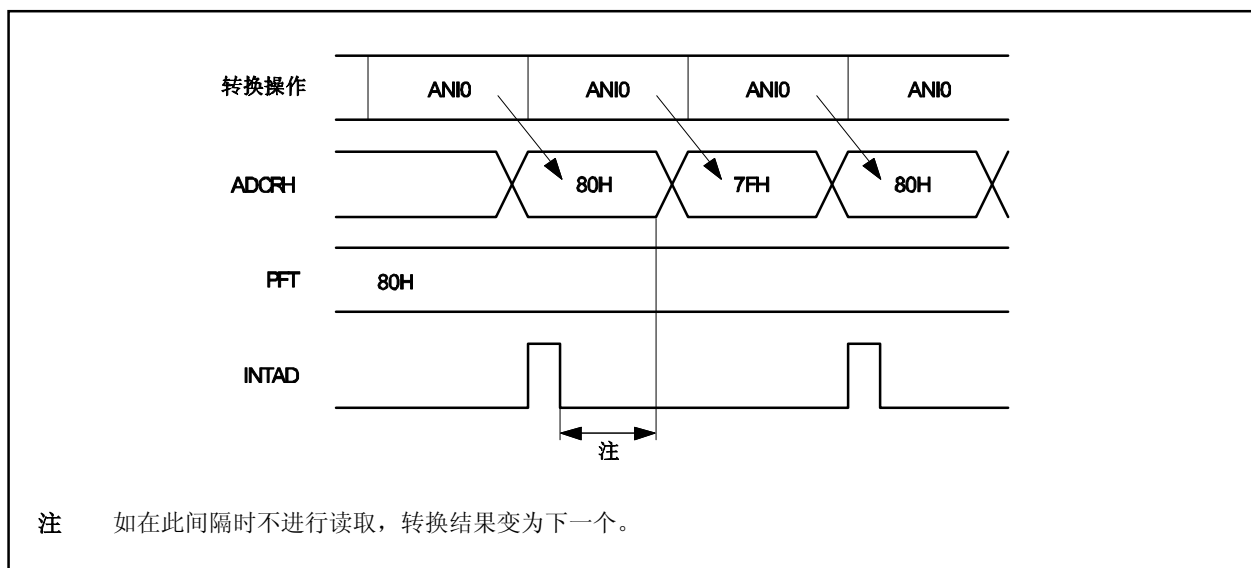


13.5.4 掉电检测功能

通过使用 PFM 及 PFT 寄存器（如下）来控制转换结束中断请求信号（INTAD）。

- 如 PFM.PFEN 位设置为 0，在每次转换结束时总生成 INTAD 信号。
- 如 PFEN 位 = 1 及 PFM.PFCM 位 = 0，在转换结束时，转换结果（ADCRH 寄存器）及 PFT 寄存器的值进行比较，只有 $ADCRH \geq PFT$ ，才会生成 INTAD 信号。
- 如 PFEN 位及 PFM.PFCM 位皆等于 1，在转换结束时，转换结果及 PFT 寄存器的值进行比较，只有 $ADCRH < PFT$ ，才会生成 INTAD 信号。
- 因为，当 PFEN 位 = 1 时，除非转换值在下一个转换结束之前读取，否则在 INTAD 信号生成后转换值会被重写，在一些情形实际操作可能与上述有所不同（参考图 13-6）。

图 13-6. 掉电检测功能（PFCM 位 = 0）



13.5.5 设置方法

以下描述了如何设置寄存器。

(1) 在 A/D 转换中使用 A/D 转换器

<1> 设置 (1) ADM.ADCS2 位。

<2> 通过设置 ADS.ADS2~ADS.ADS0 位和 ADM.ADHS1, ADM.ADHS0 和 ADM.FR2 ~ADM.FR0 位来选择通道及转换时间。

<3> 设置 (1) ADM.ADCS 位。

<4> A/D 转换数据传输到 ADCR 寄存器。

<5> 生成中断请求信号 (INTAD)。

<改变通道>

<6> 通过设置 ADS2 ~ADS0 位来改变通道。

<7> A/D 转换数据传输到 ADCR 寄存器。

<8> 生成中断请求信号 (INTAD)。

<结束 A/D 转换>

<9> 清 (0) ADCS 位。

<10> 清 (0) ADCS2 位。

注意事项 1. <1> ~<3>之间的时间间隔必须为 1 μ s (高速模式) 或 14 μ s (标准模式) 或更多。

2. 步骤<1>和<2>可调换。

3. 步骤<1> 可忽略。然而, 如忽略步骤<1>, 在步骤<3>后不要使用首次转换结果。

4. 步骤<4> ~<7>之间的时间间隔与 ADHS1, ADHS0, 及 FR2~ FR0 位设置的转换时间不同。

步骤<6> ~<7>之间的时间间隔与 ADHS1, ADHS0, 及 FR2~ FR0 位设置的转换时间相同。

(2) 使用 A/D 转换器的掉电检测功能

<1> 设置 PFM.PFEN 位为 (1)。

<2> 通过 PFM.PFCM 位设置掉电比较条件。

<3> 设置 (1) ADM.ADCS2 位。

<4> 通过设置 ADS.ADS2~ADS.ADS0 位及 ADM.ADHS1, ADM.ADHS0, 及 ADM.FR2 ~ADM.FR0 位来选择通道及转换时间。

<5> 在 PFT 寄存器中设置阈值。

<6> 设置 (1) ADM.ADCS 位。

<7> A/D 转换数据传输到 ADCR 寄存器。

<8> 比较 ADCRH 寄存器和 PFT 寄存器。当条件相等时, 生成中断请求信号 (INTAD)。

<改变通道>

<9> 通过设置 ADS2 到 ADS0 位改变通道。

<10> A/D 转换数据传输到 ADCR 寄存器。

<11> 对比 ADCRH 及 PFT 寄存器。当条件相等时, 生成中断请求信号 (INTAD)。

<结束 A/D 转换>

<12> 清 (0) ADSC 位。

<13> 清 (0) ADSC2 位。

备注 如果容许掉电检测功能操作, 无论设置选择模式还是扫描模式, 将比较所有 A/D 转换结果。

13.6 注意事项

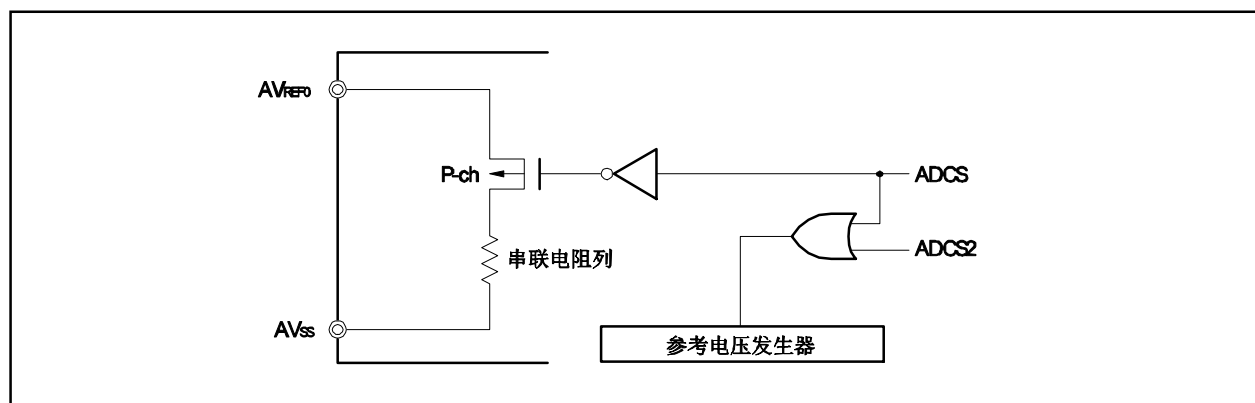
(1) 待机模式下的耗电量

在待机模式下 A/D 转换器操作停止。此时，可通过停止转换模式（ADM.ADCS 位 = 0）及参考电压生成器（ADM.ADCS2 位 = 0）来减少耗电量。

图 13-7 所示的是在待机模式下如何减少耗电量的示例。

<R>

图 13-7. 在待机模式下如何减少耗电量的示例



(2) ANI0 ~ ANI7 引脚的输入范围

输入指定范围的电压值到引脚 ANI0 ~ ANI17。如果大于等于 AV_{REF0} 或小于等于 AV_{SS} （即使在最大绝对额定值范围内）的电压被输入到这些引脚，则那个通道的转换值不确定，其它通道的转换值也会受到影响。

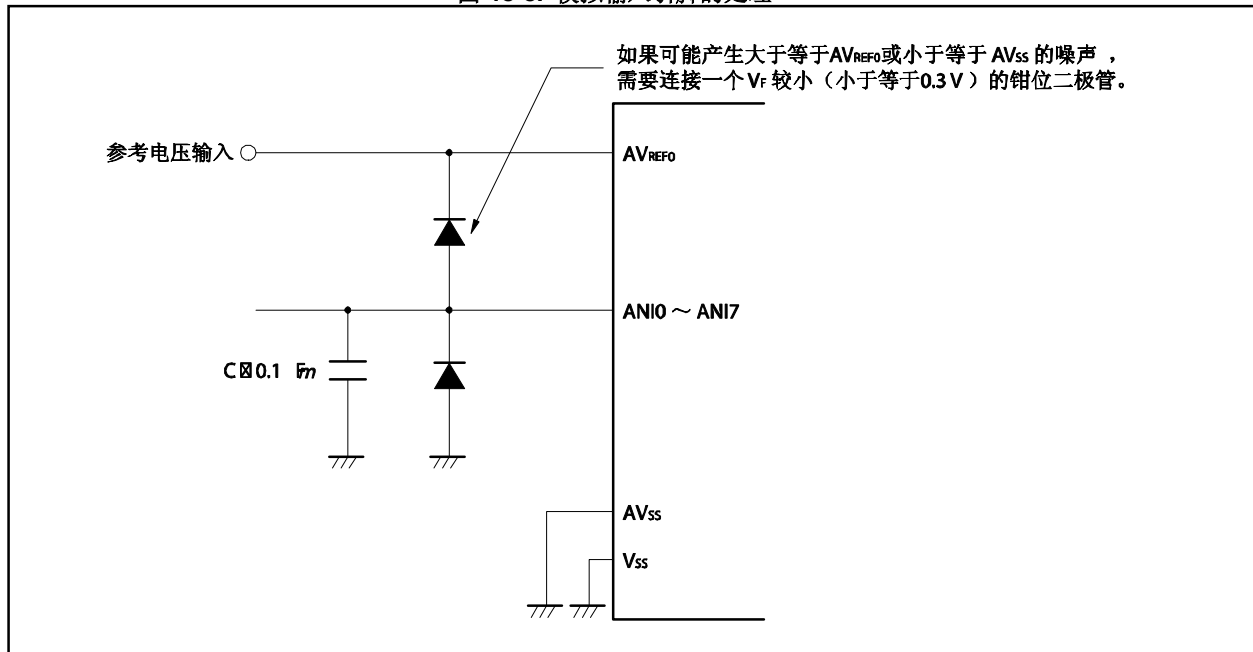
(3) 冲突的操作

- (a) 在转换结束时，写入 ADCR 寄存器与读取 ADCR 寄存器冲突。
读取 ADCR 优先。读取之后，新的转换结果写入 ADCR 寄存器。
- (b) 在转换结束时，写入 ADCR 寄存器与写入 ADM 寄存器或写入 ADS 寄存器冲突。
写入 ADM 寄存器或写入 ADS 寄存器优先。没写入 ADCR 或生成转换结束中断请求信号（INTAD）。

(4) 抑制噪声的方法

为了确保 10 位分辨率，注意 AV_{REF0} 及 $ANI0 \sim ANI7$ 引脚上的噪声。噪声影响随着模拟输入源的输出阻抗的增加而提高。为了降低噪声，推荐连接一个外部电容，如图 13-8 所示。

图 13-8. 模拟输入引脚的处理

**(5) $ANI0/P70 \sim ANI7/P77$ 引脚**

模拟输入引脚（ $ANI0 \sim ANI17$ ）可功能复用为输入端口引脚（ $P70 \sim P77$ ）。

当从引脚 $ANI0 \sim ANI17$ 中选择一个执行 A/D 转换时，不要在转换期间对端口 7 执行输入指令，否则转换分辨率将下降。

如果某引脚正在进行 A/D 转换，其临近引脚输入一个数字脉冲，则由于耦合噪声的影响，A/D 转换值可能与预期值有出入。因此，在 A/D 转换过程中，确保不使用临近引脚传输脉冲信号。

(6) AV_{REF0} 引脚的输入阻抗

AV_{REF0} 引脚及 AV_{SS} 引脚之间串联着数十个 $k\Omega$ 的电阻。

因此，如果参考电压源的输出阻抗较高，输出阻抗将与 AV_{REF0} 引脚及 AV_{SS} 引脚之间的串联电阻串联，导致产生较大的参考电压误差。

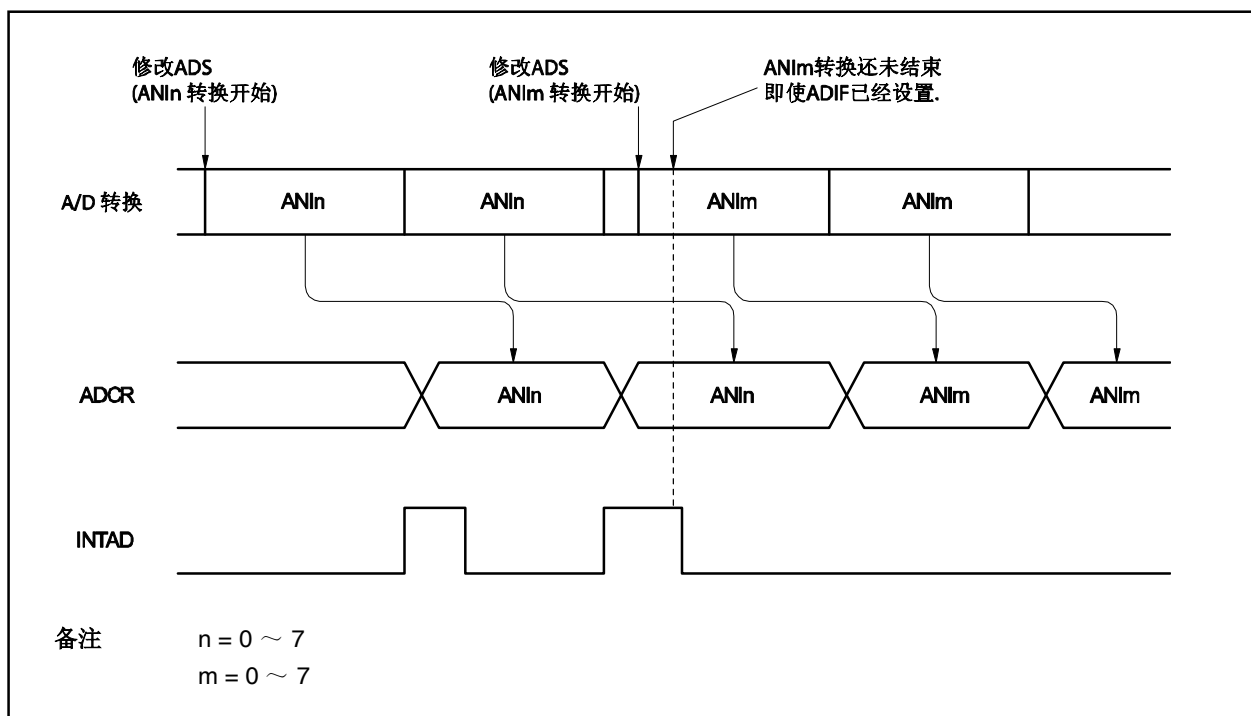
(7) 中断请求标志 (ADIC.ADIF 位)

即使寄存器 ADS 的内容改变, ADIF 位也不清零。

因此, 如果在 A/D 转换过程中模拟输入引脚改变, ADIF 位可能设置为 (1), 因为在上一次模拟输入引脚的 A/D 转换结束后 ADS 寄存器才立即被复写。在该情况下, 如果寄存器 ADS 复写后立即读取标志 ADIF, 标志 ADIF 可能置 1, 即使新选择的模拟输入信号的转换还未完成。

当 A/D 转换停止, 在再次转换前使标志 ADIF 清零。

图 13-9. A/D 转换结束中断请求产生时序

**(8) A/D 转换后的首次转换值**

在 ADM.ADCS2 位设置为 1 后如果 ADM.ADCS 位在 $1\ \mu\text{s}$ (高速模式) 或 $1\ 4\ \mu\text{s}$ (标准模式) 内设置为 1, 或如果 ADCS2 清零后设置 ADCS 位为 1, A/D 转换操作启动后的首次转换值可能不会符合标准。采取相关措施如发出 A/D 转换结束中断请求信号 (INTAD) 及丢弃首次转换结果。

(9) 读取 A/D 转换结果寄存器 (ADCR)

当执行寄存器 ADM 或 ADS 的写指令, 寄存器 ADCR 的内容可能不确定。当转换操作完成后, 在写入 ADM 或 ADS 寄存器前读取转换值。如果不按照以上时序读取, 可能无法读取正确的转换值。

当主时钟 (fx) 振荡停止, CPU 在副时钟下操作时, 禁止访问 ADCR 及 ADCRH 寄存器。敬请参阅 3.4.8

(2) 访问特殊片上外围 I/O 寄存器。

(10) A/D 转换器采样时间及 A/D 转换启动延迟时间

A/D 转换采样时间因 ADM 的设置值不同有所变化。A/D 转换器操作使能后，经过一定延迟时间后，才开始实际采样。

在使用 A/D 转换时间需严格遵守的设置时，并注意图 13-10 和表 13-4 所示的内容。

图 13-10. A/D 转换器采样及 A/D 转换启动延时

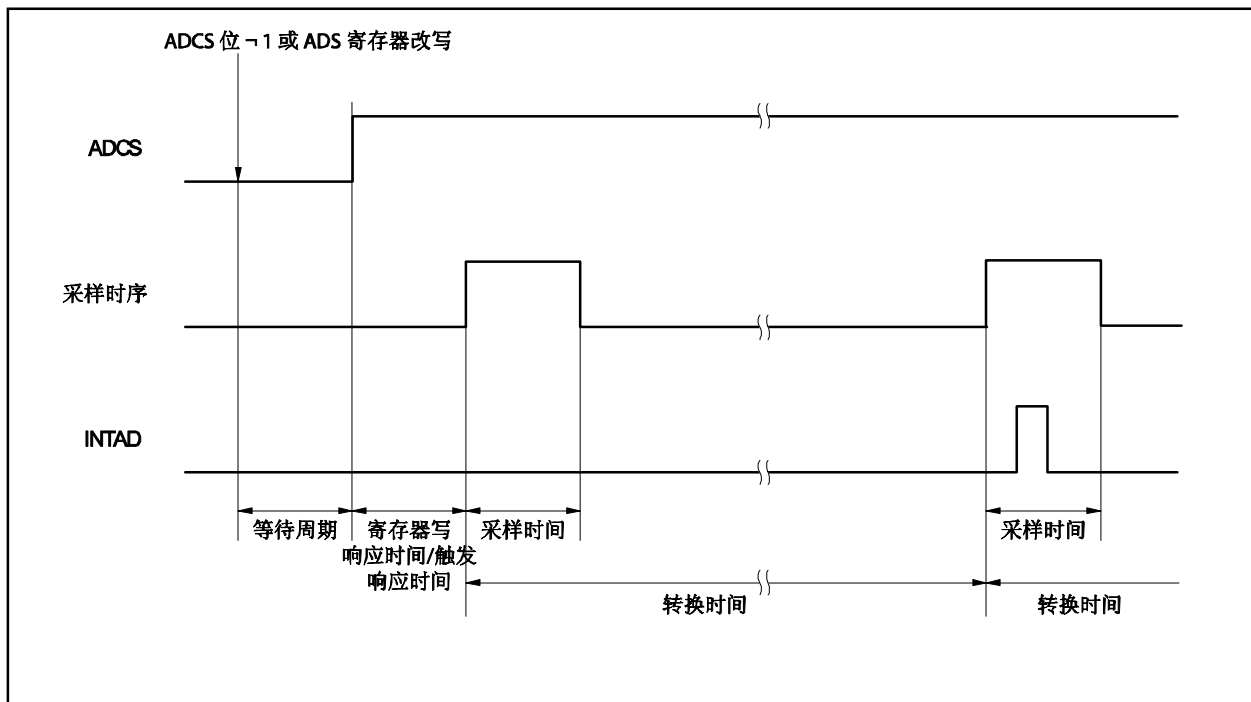


表 13-4. A/D 转换器转换时间

ADHS1	ADHS0	FR2	FR1	FR0	转换时间	采样时间	寄存器写入反应时间 ^注		触发反应时间 ^注	
							MIN.	MAX.	MIN.	MAX.
0	0	0	0	0	288/f _{xx}	176/f _{xx}	11/f _{xx}	12/f _{xx}	7/f _{xx}	8/f _{xx}
0	0	0	0	1	240/f _{xx}	176/f _{xx}	11/f _{xx}	12/f _{xx}	7/f _{xx}	8/f _{xx}
0	0	0	1	0	192/f _{xx}	132/f _{xx}	10/f _{xx}	11/f _{xx}	6/f _{xx}	7/f _{xx}
0	0	1	0	0	144/f _{xx}	88/f _{xx}	9/f _{xx}	10/f _{xx}	5/f _{xx}	6/f _{xx}
0	0	1	0	1	120/f _{xx}	88/f _{xx}	9/f _{xx}	10/f _{xx}	5/f _{xx}	6/f _{xx}
0	0	1	1	0	96/f _{xx}	48/f _{xx}	11/f _{xx}	12/f _{xx}	7/f _{xx}	8/f _{xx}
0	1	0	0	0	96/f _{xx}	48/f _{xx}	11/f _{xx}	12/f _{xx}	7/f _{xx}	8/f _{xx}
0	1	0	0	1	72/f _{xx}	36/f _{xx}	10/f _{xx}	11/f _{xx}	6/f _{xx}	7/f _{xx}
0	1	0	1	0	48/f _{xx}	24/f _{xx}	9/f _{xx}	10/f _{xx}	5/f _{xx}	6/f _{xx}
0	1	0	1	1	24/f _{xx}	12/f _{xx}	8/f _{xx}	9/f _{xx}	4/f _{xx}	5/f _{xx}
0	1	1	0	0	224/f _{xx}	176/f _{xx}	11/f _{xx}	12/f _{xx}	7/f _{xx}	8/f _{xx}
0	1	1	0	1	168/f _{xx}	132/f _{xx}	10/f _{xx}	11/f _{xx}	6/f _{xx}	7/f _{xx}
0	1	1	1	0	112/f _{xx}	88/f _{xx}	9/f _{xx}	10/f _{xx}	5/f _{xx}	6/f _{xx}
0	1	1	1	1	56/f _{xx}	44/f _{xx}	8/f _{xx}	9/f _{xx}	4/f _{xx}	5/f _{xx}
1	0	0	0	0	72/f _{xx}	24/f _{xx}	11/f _{xx}	12/f _{xx}	7/f _{xx}	8/f _{xx}
1	0	0	0	1	54/f _{xx}	18/f _{xx}	10/f _{xx}	11/f _{xx}	6/f _{xx}	7/f _{xx}
1	0	0	1	0	36/f _{xx}	12/f _{xx}	9/f _{xx}	10/f _{xx}	5/f _{xx}	6/f _{xx}
1	0	0	1	1	18/f _{xx}	6/f _{xx}	8/f _{xx}	9/f _{xx}	4/f _{xx}	5/f _{xx}
以上除外					禁止设置	-	-	-	-	-

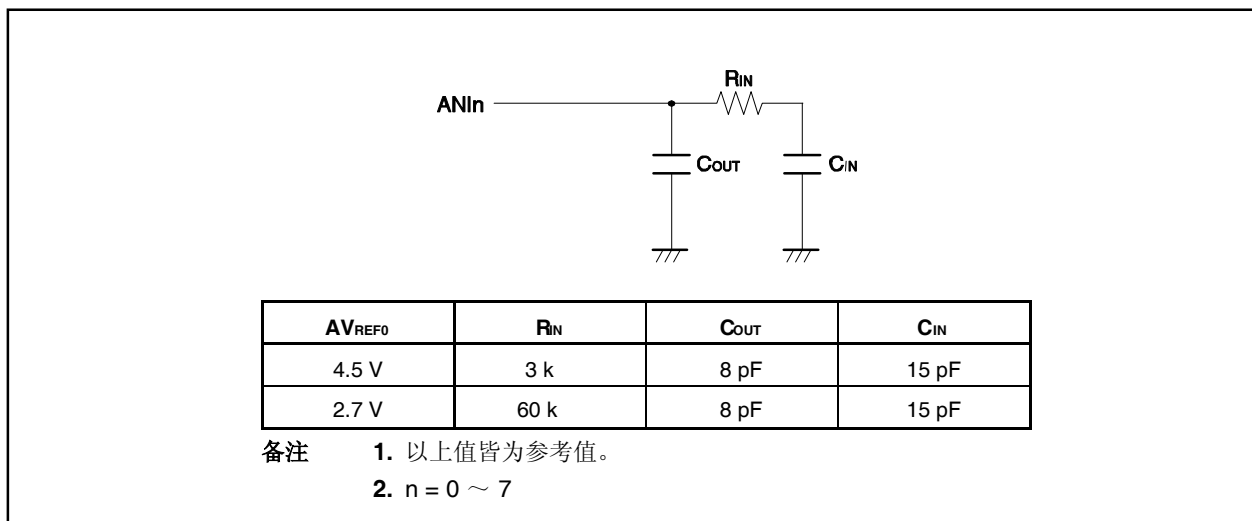
注 反应时间是指等待期之后的时间。关于等待功能，参考 3.4.8 (2) 访问特殊片上外围 I/O 寄存器。

备注 f_{xx}: 主时钟频率

(11) 内部等效电路

如下所示为模拟输入模块的等效电路。

图 13-11. 内部等效电路的 ANIn 引脚

**(12) A/D 转换结果的浮动**

由于供电电压的波动，A/D 转换结果可能产生变化，或着可能受到噪声影响。为了减小浮动，使用多次测量求平均值的方法。

(13) A/D 转换结果滞后特性

逐次逼近 A/D 转换器维持内部采样和保持电容的模拟输入电压，同时进行 A/D 转换。A/D 转换结束后模拟输入电压保留在内部采样和保持电容中。结果，可能产生下述现象。

- 当同一个通道用于 A/D 转换，如果电压高于或低于先前 A/D 转换，则出现滞后特征，转换结果受到上次转换值影响。因此，即使同一个模拟输入电压也会产生不同结果。
- 当改变模拟输入通道，可能出现滞后特征，转换结果受到上次转换通道影响。这是因为只有一个 A/D 转换器用于 A/D 转换。因此，即使同一个模拟输入电压也会产生不同结果。

因此，为获得更为精确的转换值，在同一通道下连续两次操作 A/D 转换过程，丢弃首次转换结果。

<R>

(14) 标准模式下的 A/D 转换操作

- 软件触发模式：
在标准模式下的转换过程中，禁止写入 ADM， ADS， PFM， 或 PFT 寄存器。（ADM.ADHS1， ADM.ADHS0 位 = 00）。
- 硬件触发模式（外部触发/定时器触发）：
在标准模式下无法使用该模式（ADHS1， ADHS0 位 = 00），可在高速模式下使用（ADHS1， ADHS0 位 = 10 或 01）。

13.7 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 **1 LSB**（最低有效位）。对于满度的 **1 LSB** 的百分比用 **%FSR**（满度范围）表示。**%FSR** 是以百分比形式表示的一定范围的可变模拟输入电压的比率。与分辨率无关，可用如下公式表示。

$$\begin{aligned} 1\%FSR &= (\text{可转换模拟输入电压的最大值} - \text{可转换模拟输入电压的最小值}) / 100 \\ &= (AV_{REF0} - 0) / 100 \\ &= AV_{REF0} / 100 \end{aligned}$$

当分辨率为 10 位时 **1LSB** 表示如下。

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

精确度与分辨率无关，而由总误差决定。

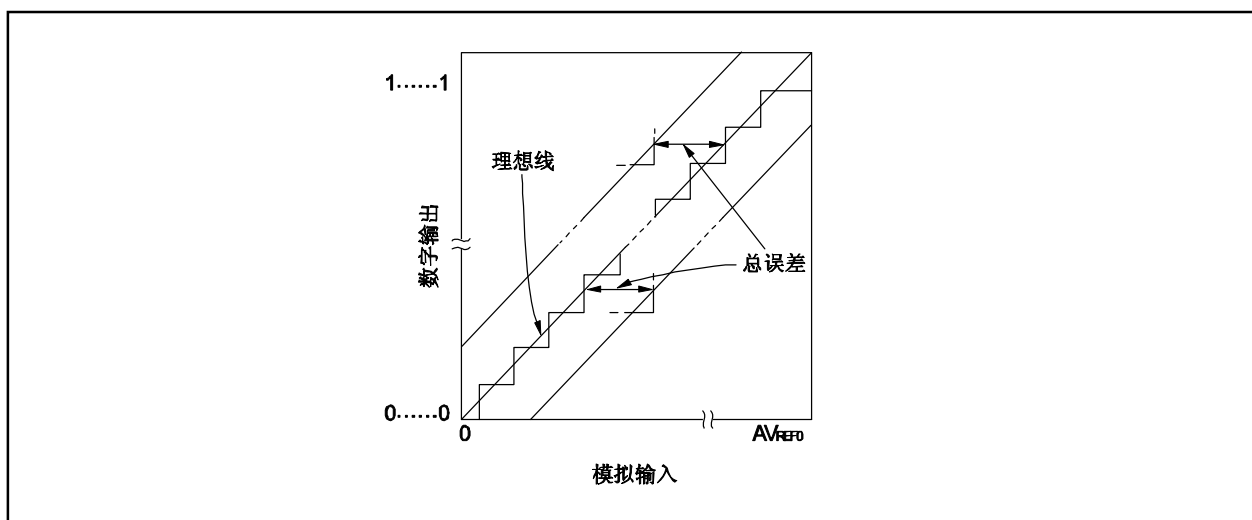
(2) 总误差

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、积分线性误差和积分线性误差等组合起来表示总误差。

量化误差不属于特征表中总误差的范围。

图 13-12. 总误差

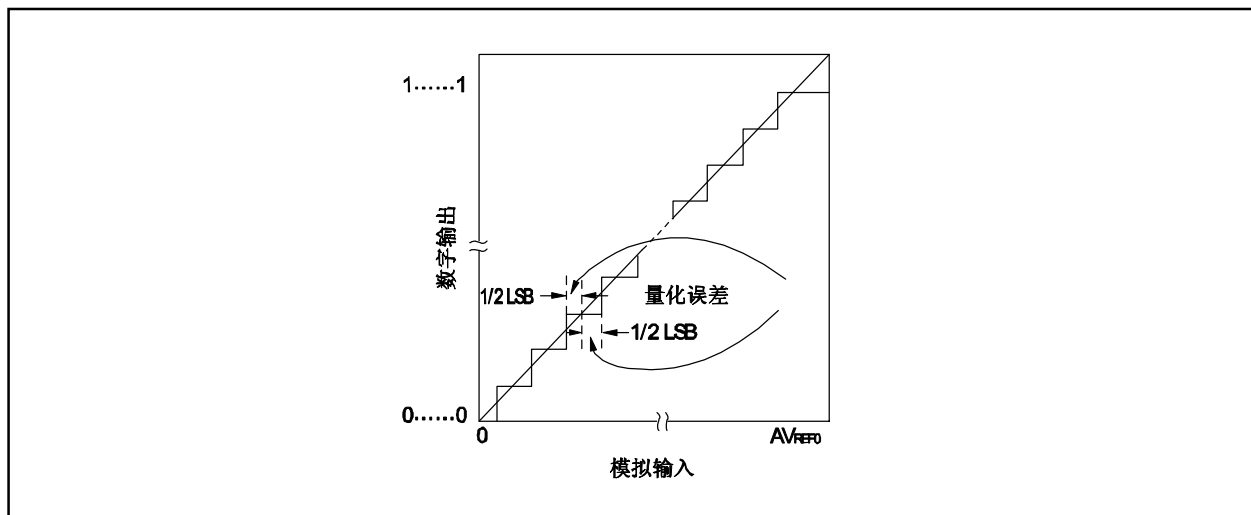


(3) 量化误差

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。

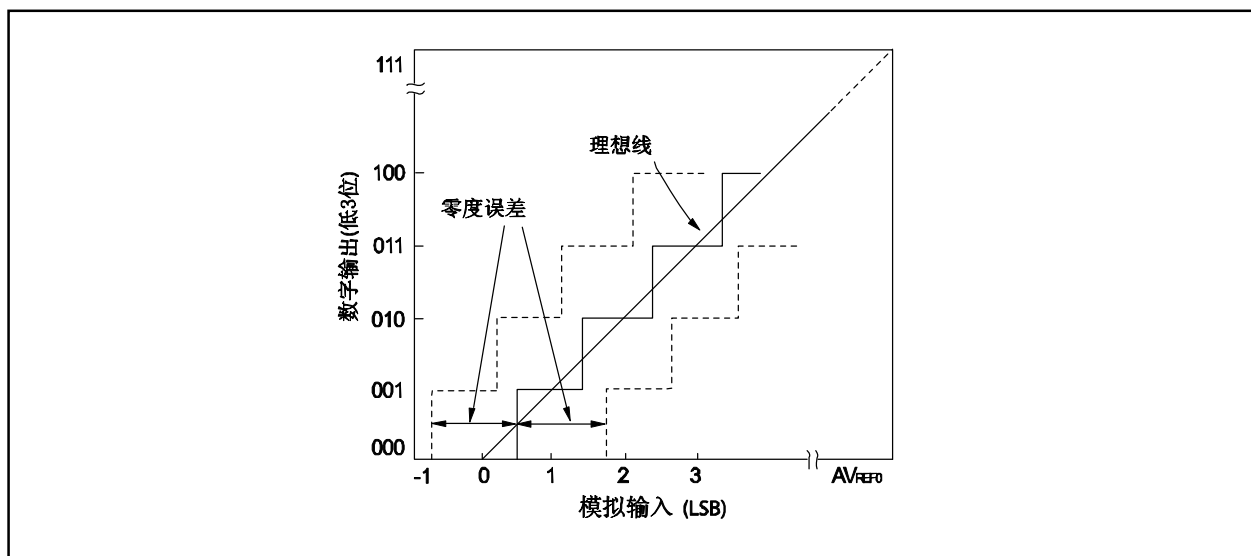
量化误差不属于特征表中总误差、零度误差、满度误差、积分线性误差和微分线性误差的范围。

图 13-13. 量化误差

**(4) 零度误差**

零度误差表示当数字输出范围在 $0.....000 \sim 0.....001$ 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。

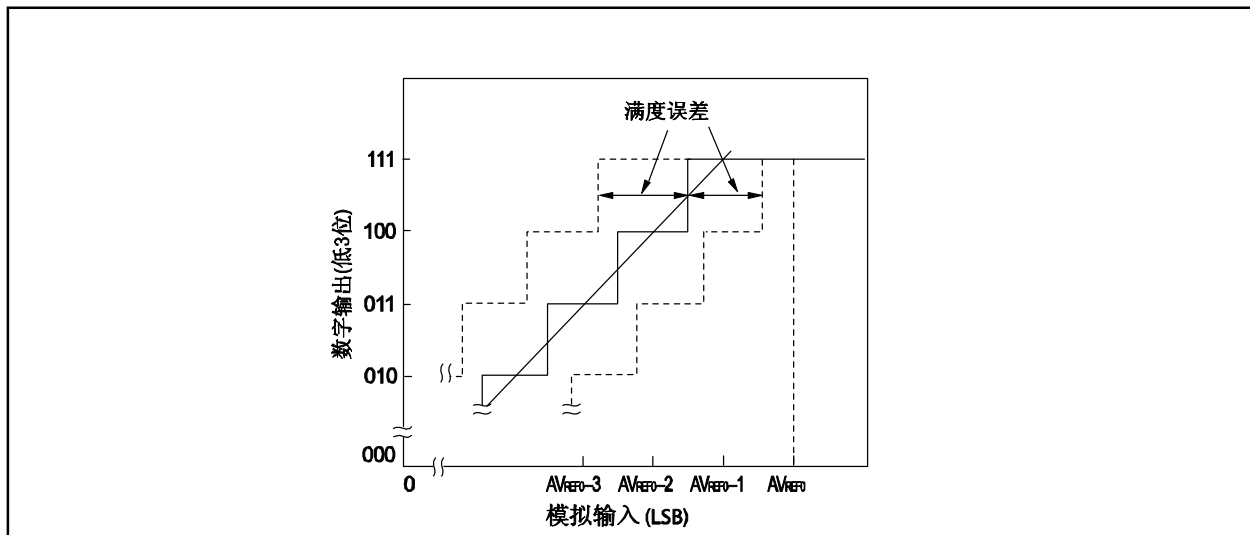
图 13-14. 零度误差



(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。

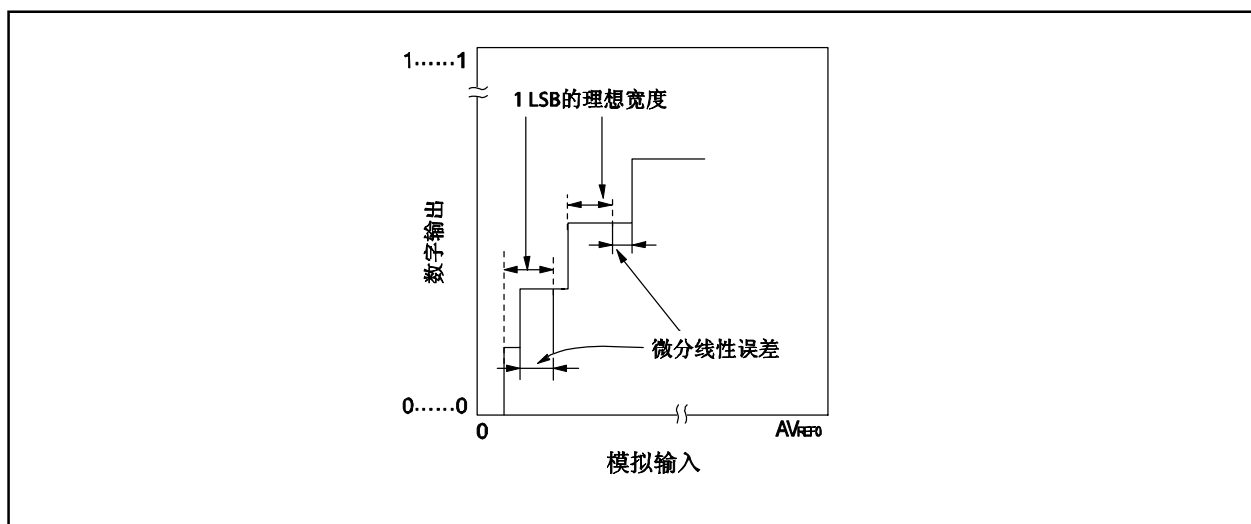
表 13-15. 满度误差



(6) 微分线性误差

理论上，代码输出的宽度为 1LSB 。当输出一个特定码，微分线性误差表示实际测量值与理想值之间的差距。当同一个通道的模拟输入引脚的电压逐渐地从 AV_{SS} 到 AV_{REF0} 持续增加，这显示了 A/D 转换的基本特征。当输入电压增加或减少，或当两个或更多的通道被使用时，参见 13.7 (2) 总误差。

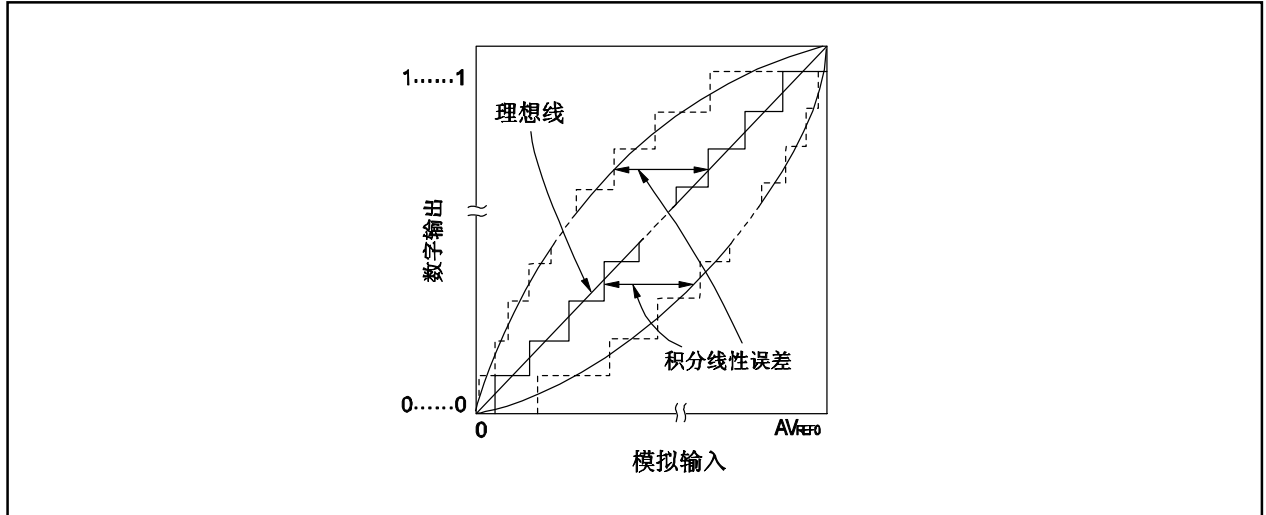
图 13-16. 微分线性误差



(7) 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。

图 13-17. 积分线性误差

**(8) 转换时间**

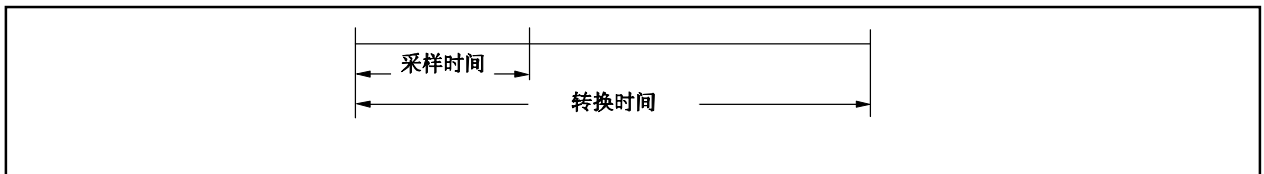
转换时间表示从产生触发到获取数字输出所经历的时间。

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和保持电路采样所需的时间。

图 13-18. 采样时间



第十四章 异步串行接口 (UART)

在 V850ES/KE2 中提供了两个异步串行接口 (UART) 的通道。

14.1 特点

- 最大发送速度：312.5 kbps
- 全双工通讯
 - 片上 RXBn 寄存器
 - 片上 TXBn 寄存器
- 两引脚配置^注
 - TXDn：发送数据输出引脚
 - RXDn：接收数据输入引脚
- 接收错误检测功能
 - 奇偶校验错误
 - 帧错误
 - 过载错误
- 中断源：三种类型
 - 接收错误中断请求信号 (INTSREn)：
 - 接收完成中断请求信号 (INTSRn)：
 - 发送完成中断请求信号 (INTSTn)：
- 字符长度：7 或 8 位
- 奇偶校验功能：奇校验、偶校验 0 校验或无校验
- 发送停止位：1 或 2 位
- 片上专门波特率生成器

根据三种类型的接收错误的逻辑 OR 生成的中断。

在接收允许状态下完成串行发送后当接收数据从接收移位寄存器转移到 RXBn 寄存器时生成的中断。

当从发送移位寄存器中发送串行发送数据 (8 或 7 位) 完成时生成的中断。

注 ASCK0 引脚 (外部时钟引脚) 只对 UART0 可用。

14.2 配置

表 14-1. UARTn 的配置

项目	配置
寄存器	接收缓冲寄存器 n (RXBn) 发送缓冲寄存器 n (TXBn) 接收移位寄存器 发送移位寄存器 异步串行接口模式寄存器 n (ASIMM) 异步串行接口状态寄存器 n (ASISn) 异步串行接口发送状态寄存器 n (ASIFn)
其他	接收控制奇偶校验 添加发送控制校验

备注 n = 0, 1

图 14-1 显示的是 UARTn 的配置。

(1) 异步串行接口模式寄存器 n (ASIMn)

ASIMn 寄存器是指定 UARTn 运行的 8 位寄存器。

(2) 异步串行接口状态寄存器 n (ASISn)

ASISn 寄存器包含一套在出现接收错误时指出错误内容的标志。各种接收错误标志在出现接收错误时设为 1，在读取 ASISn 寄存器时清零。

(3) 异步串行接口发送状态寄存器 n (ASIFn)

ASIFn 寄存器是在进行发送操作时状态的 8 位寄存器。

该寄存器包含发送缓冲数据标志 (表示 TXBn 寄存器数据的保持状态) 和发送移位寄存器数据标志 (表示发送是否在进行中)。

(4) 接收控制奇偶校验

接收操作是根据 ASIMn 寄存器的内容控制的。在接收过程中也会检查奇偶校验错误；如果检测到错误，则会在 ASISn 寄存器中对应错误内容设置一个值。

(5) 接收移位寄存器

这是一个将输入到 RXDn 引脚的串行数据转换到并行数据的移位寄存器。接收一个字节的的数据，如果检测到停止位，接收数据被发送到 RXBn 寄存器。

不能对该寄存器进行直接操作。

(6) 接收缓冲寄存器 n (RXBn)

RXBn 寄存器是一个保持接收数据的 8 位缓冲寄存器。当收到 7 个字符时，在最高有效位储存 0。

在接收允许状态中，接收数据从接收移位寄存器转移至 RXBn 寄存器，与一个帧的移入流程结束同步。

同时，发送数据到 RXBn 寄存器也会生成接收完成中断请求信号 (INTSRn)。

(7) 发送移位寄存器

这是一个将来自 TXBn 寄存器的并行数据转换到串行数据的移位寄存器。
 当从 TXBn 寄存器发送一个字节的的数据时，移位寄存器数据从 TXDn 引脚输出。
 接收完成中断请求信号 (INTSTn) 会在一个帧发送完成的同时产生。
 不能对该寄存器进行直接操作。

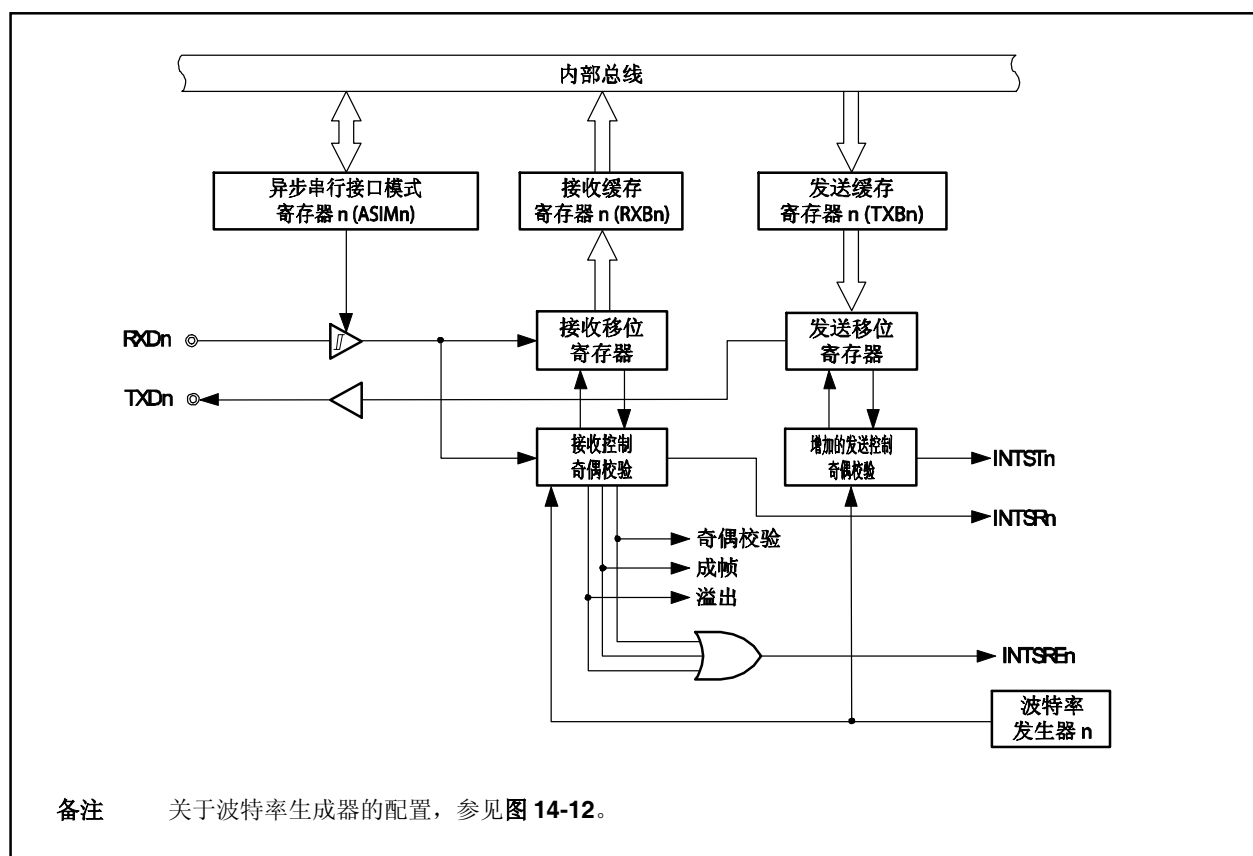
(8) 发送缓冲寄存器 n (TXBn)

TXBn 寄存器是一个发送数据的 8 位缓冲器。写入发送数据到 TXBn 寄存器会启动发送操作。

(9) 添加发送控制校验

发送操作是通过添加一个起始位、奇偶校验位或停止位到写入到 TXBn 寄存器的数据，根据 ASIMn 寄存器中设置的内容控制的。

图 14-1. UARTn 的框图



14.3 寄存器

(1) 异步串行接口模式寄存器 n (ASIMn)

ASIMn 寄存器是一个控制 UARTn 发送操作的 8 位寄存器。

该寄存器支持 8 位或 1 位只读方式。

该寄存器复位后的值为 01H。

- 注意事项**
1. 在使用 UARTn 时，确保在设置 CKSRn 和 BRGCn 寄存器前设置与 UARTn 功能相关的外部引脚到控制模式，然后设置 UARTEn 位到 1，在开始设置其它位。
 2. 当输入高电平到 RXDn 引脚时将 UARTEn 和 RXEn 位设为 1。当输入低电平到 RXDn 引脚将这些位设置为 1，接收将会开始。

(1/2)

复位后: 01H R/W 地址: ASIM0 FFFFA00H, ASIM1 FFFFA10H

	<7>	<6>	<5>	4	3	2	1	0
ASIMn	UARTEn	TXEn	RXEn	PSn1	PSn0	CLn	SLn	ISRMn

(n = 0, 1)

UARTEn	运行时钟的控制
0	停止向 UARTn 的时钟供应。
1	向 UARTn 时钟供应。
<ul style="list-style-type: none"> • 如果 UARTEn 位清零，UARTn 同步复位[✱]。 • 如果 UARTEn 位 = 0，UARTn 复位。如果要操作 UARTn，首先将 UARTEn 位设为 1。 • 如果将 UARTEn 位从 1 清为 0，所有 UARTn 寄存器被初始化。要将 UARTEn 位重新设置为 1，必须复位 UARTn 寄存器。 <p>不论 UARTEn 位如何设置，TXDn 引脚的输出在禁止发送时会升高。</p>	

TXEn	发送允许/禁止
0	禁止发送。
1	允许发送
<ul style="list-style-type: none"> • 在启动时将 UARTEn 位设为 1 后将 TXEn 位设为 1。在将 TXEn 位清零后将 UARTEn 位清零即可停止。 • 要初始化发送单元的状态，将 TXEn 清零；在经过两个时钟周期（基础时钟）后，将 TXEn 设为 1。如果没有再设置 TXEn 位，初始化无法成功。（关于更多细节，敬请参阅 14.6.1 (1) 基础时钟。） 	

注 复位 ASISn、ASIFn 和 RXBn 寄存器。

RXEn	接收允许/禁止
0	禁止接收 [#]
1	允许接收

- 在启动时将 **UARTEn** 位设为 1 后将 **RXEn** 位设为 1。在将 **RXEn** 位清零后将 **UARTEn** 位清零即可停止。
- 要初始化接收单元的状态，将 **RXEn** 清零；在经过两个时钟周期（基础时钟）后，将 **RXEn** 设为 1。如果没有再设置 **RXEn** 位，初始化无法成功。（关于基础时钟的更多细节，敬请参阅 **14.6.1 (1) 基础时钟**。）

PSn1	PSn0	发送操作	接收操作
0	0	不输出校验位	不接收校验位
0	1	输出 0 校验位	接收为 0 校验位
1	0	输出奇数校验位	判为奇数校验位
1	1	输出偶数校验位	判为偶数校验位

- 要重写 **PSn1** 和 **PSn0** 位，首先将 **TXEn** 和 **RXEn** 位清零。
- 如果为接收选择“0 校验位”，不校验位判定。由于没有设置 **ASISn.PEn** 位，而不会生成错误中断。

CLn	1 帧发送/接收数据字符长度的指定
0	7 位
1	8 位

- 要重写 **CLn** 位，首先将 **TXEn** 和 **RXEn** 位清零。

S Ln	发送数据的停止位长度指定
0	1 位
1	2 位

- 要重写 **S Ln** 位，首先将 **TXEn** 位清零。
- 由于接收总是要以长度为 1 的停止位结束，**S Ln** 位的设置不影响接收操作。

ISRMn	当发生错误时，接收完成中断请求信号产生的禁止/使能
0	当出现错误时生成一个接收错误中断请求信号（ INTSREn ）作为一个中断。 在此情况下，不生成接收完成中断请求信号（ INTSRn ）。
1	当出现错误时生成一个接收完成中断请求信号（ INTSRn ）作为一个中断。 在此情况下，不生成接收错误中断请求信号（ INTSREn ）。

- 要重写 **ISRMn** 位，首先将 **RXEn** 位清零。

注 当禁止接收时，接收移位寄存器不检测开始位。不对 **RXBn** 寄存器进行移入处理或发送处理，同时保留 **RXBn** 寄存器的内容。

当允许接收时，接收移位操作开始，并与开始位的检测同步。当一个帧的接收完成时，接收移位寄存器的内容被传送到 **RXBn** 寄存器。与此同时，产生一个接收完成中断请求信号（**INTSRn**）。

(2) 异步串行接口状态寄存器 n (ASISn)

包含三个错误标志位 (Pen、Fen 和 OVEN) 的 ASISn 寄存器在 UARTn 接收完成时显示错误状态。通过一个读取操作将 ASISn 寄存器清除为 00H。当出现接收错误时, 应读取 RXBn 寄存器, 并在读取 ASISn 寄存器后清除错误标志。
 该寄存器支持 8 位只读方式。
 该寄存器复位后的值为 00H。

- 注意事项**
1. 当在 ASIMn.UARTEn 位或 ASIMn.RXEn 位清零或者读取 ASISn 寄存器时, Pen、Fen 和 OVEN 位清零。
 2. 禁止使用位操作指令进行操作。
 3. 当停止主时钟, CPU 运行在副时钟上时, 不要访问 ASISn 寄存器。关于更多细节, 请参见 3.4.8 (2)。

复位后: 00H R 地址: ASIF0 FFFFA05H, ASIF1 FFFFA15H

	7	6	5	4	3	2	<1>	<0>
ASISn	0	0	0	0	0	0	TXBFn	TXSFn

(n = 0, 1)

Pen	奇偶校验错误的状态标志
0	当 UARTEn 或 RXEn 清零时或者读取了 ASISn 寄存器后
1	当接收完成时, 接收数据的奇偶校验与奇偶校验位不相等
• Pen 位的操作根据 ASIMn.PSn1 和 ASIMn.PSn0 位的设置而不同。	

FEn	帧错误的状态标志
0	当 UARTEn 或 RXEn 清零时或者读取了 ASISn 寄存器后
1	当接收完成时, 没有检测到停止位
• 对于接收数据的停止位, 只检查第一位而不论停止位的长度。	

OVEn	过载错误的状态标志
0	当 UARTEn 或 RXEn 清零时或者读取了 ASISn 寄存器后。
1	UARTn 在读取 RXBn 寄存器的接收数据前完成了下一个接收操作。
• 当出现过载错误后, 下一个接收的数据值不重写到 RXBn 寄存器, 数据被丢弃。	

(3) 异步串行接口发送状态寄存器 n (ASIFn)

ASIFn 寄存器包括两个状态标志位，表示发送进行过程中的状态。

通过在将数据从 TXBn 寄存器转移到发送移位寄存器后写入下一个数据到 TXBn 寄存器，可以确保在出现中断间隔的情况下发送操作连续无中断地进行。当进行连续操作时，应在参照 TXBFn 位后写入数据以避免错误地写入数据到 TXBn 寄存器。

该寄存器支持 8 位或 1 位只读方式。

该寄存器复位后的值为 00H。

复位后: 00H R 地址: ASIF0 FFFFA05H, ASIF1 FFFFA15H

	7	6	5	4	3	2	<1>	<0>
ASIFn	0	0	0	0	0	0	TXBFn	TXSFn

(n = 0, 1)

TXBFn	发送缓冲数据标志
0	没有下一次发送到 TXBn 寄存器的数据 (当 ASIMn.UARTEn 或 ASIMn.TXEn 位清零时或者数据已经发送到发送移位寄存器时)
1	TXBn 寄存器中存在下次要发送的数据 (当已经写入数据到 TXBn 寄存器时, 数据在 TXBn 寄存器中存在)

- 当连续发送数据时, 应在确认此标志为 0 后将数据写入到 TXBn 寄存器。如果当本标志为 1 时写入 TXBn 寄存器, 则不能保证数据的发送。

TXSFn	发送移位寄存器数据标志 (表示 UARTn 的发送状态)
0	初始状态或等待发送 (当 UARTEn 或 TXEn 位清零时位清零时, 或者当发送完成时没有从 TXBn 寄存器发送下一个数据)
1	发送在进行中 (当数据已经从 TXBn 寄存器发送时)

- 当发送单元初始化时, 在出现发送完成中断请求信号 (INTSTn) 后应先确认在确认此标志为 0 再执行初始化。如果当本标志为 1 时进行初始化, 则不能保证数据的发送。

(4) 接收缓冲寄存器 n (RXBn)

RXBn 寄存器是一个储存已经由接收移位寄存器转换的并行数据的 8 位缓冲寄存器。

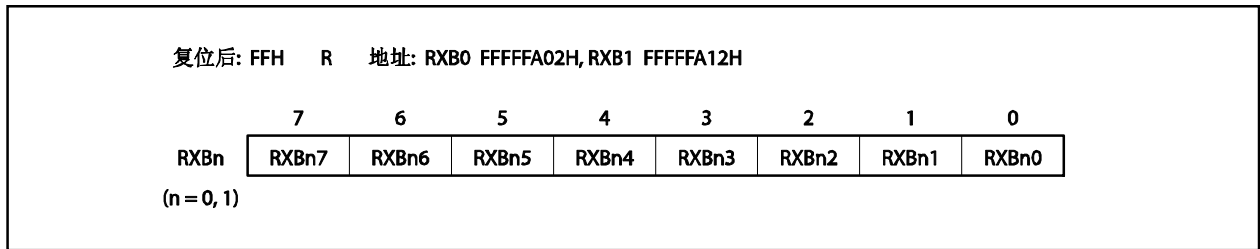
当允许接收时 (ASIMn.RXEn 位 = 1)，接收数据从接收移位寄存器转移到 RXBn 寄存器，并同步完成一个帧的移入流程。同时，到 RXBn 寄存器的转移会生成一个转移完成中断请求信号 (INTSRn)。关于生成此中断请求时间的信息，请参见 14.5.4 接收操作。

当禁止接收时 (ASIMn.RXEn 位 = 0)，保留 RXBn 寄存器的内容，即使一个帧的移入流程完成时对到 RXBn 寄存器的数据转移也不进行任何处理。同时生成 INTSRn 信号。

当指定了 7 位为数据长度，为接收数据发送 RXBn 寄存器的 6 到 0 位，最高有效位 (7 位) 总是 0。但如果出现过载错误 (ASISn.OVEn 位 = 1)，则不发送此时的接收数据到 RXBn 寄存器。

当输入复位或者 ASIMn.UARTEn 位 = 0 时，RXBn 寄存器变成 FFH。

该寄存器支持 8 位只读方式。



(5) 发送缓冲寄存器 n (TXBn)

TXBn 寄存器是用于设置发送数据的 8 位缓冲寄存器。

当允许发送时 (ASIMn.TXEn 位 = 1)，通过写入数据到 TXBn 寄存器开始发送操作。

当禁止发送时 (TXEn 位 = 1)，即使写入数据到 TXBn 寄存器，还是忽略数据。

转移 TXBn 寄存器的数据到发送移位寄存器，生成发送完成断请求信号 (INTSTn)，与完成从发送移位寄存器发送一帧同步。关于生成中断请求的信息，敬请参阅 **14.5.2 发送操作**。

当 ASIFn.TXBfn 位 = 1 时，不得写入数据到 TXBn 寄存器。

该寄存器支持 8 位只读方式。

该寄存器复位后的值为 FFH。

复位后: FFH R/W 地址: TXB0 FFFFA04H, TXB1 FFFFA14H								
	7	6	5	4	3	2	1	0
TXBn	TXBn7	TXBn6	TXBn5	TXBn4	TXBn3	TXBn2	TXBn1	TXBn0
(n = 0, 1)								

14.4 中断请求

UARTn 生成以下三类中断请求信号。

- 接收错误中断请求信号 (INTSREn)
- 接收完成中断请求信号 (INTSRn)
- 发送完成中断请求信号 (INTSTn)

以上三类中断请求信号之间的默认优先级从高到低是：接收错误中断、接收完成中断和发送完成中断。

表 14-2.生成的中断请求信号和默认的优先级

中断请求信号	优先级
接收错误中断请求信号 (INTSREn)	1
接收完成中断请求信号 (INTSRn)	2
发送完成中断请求信号 (INTSTn)	3

(1) 接收错误中断请求信号 (INTSREn)

当允许接收时，则根据为 ASISn 寄存器解释的三种类型的接收错误的逻辑 OR 生成 INTSREn 信号。可以根据 ASIMn.ISRMn 位指定在出现错误时是否生成 INTSREn 信号或 INTSRn 信号。
当禁止接收时，则不生成 INTSREn 信号。

(2) 接收完成中断请求信号 (INTSRn)

当允许接收时，当移入数据到接收移位寄存器或者发送数据到 RXBn 寄存器时生成 INTSRn 信号。即使在出现接收错误时也可以根据 ASIMn.ISRMn 位生成 INTSRn 信号代替 INTSREn 信号。
当禁止接收时，则不生成 INTSRn 信号。

(3) 发送完成中断请求信号 (INTSTn)

当包含 7 位或 8 位字符的一帧发送数据移出发送移位寄存器时生成 INTSTn 信号。

14.5 操作

14.5.1 数据格式

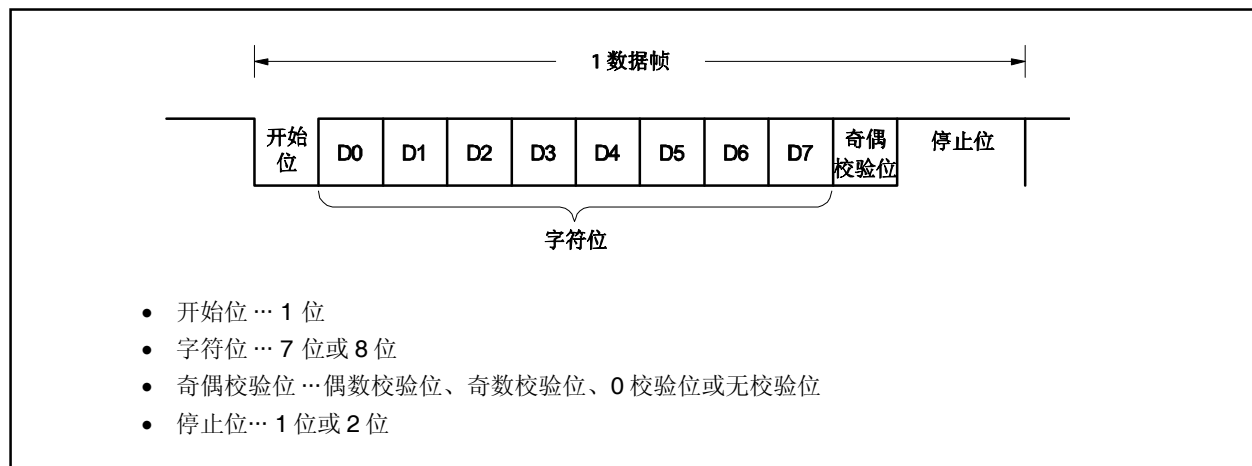
可以进行全双工串行数据发送和接收。

发送和接收的数据格式包括包含一个开始位、字符位、奇偶校验位和停止位的数据帧，如图 14-2 所示。

一个数据帧内的字符位长度、校验位的类型和停止位的长度根据 ASIMn 寄存器指定。

同时，数据首先发送最低有效位。

图 14-2. UARTn 发送/接收数据的格式



14.5.2 发送操作

当 ASIMn.UARTEn 位设为 1 时，从 TXDn 引脚输出一个高电平。

当 ASIMn.TXEn 位设为 1 时，允许发送，通过写入发送数据到 TXBn 寄存器开始发送操作。

(1) 发送允许状态

通过设置 TXEn 位设定此状态。

- TXEn 位 = 1: 发送允许状态
- TXEn 位 = 0: 发送禁止状态

由于 UARTn 没有发送允许信号 (CTS) 输入引脚应用端口，应用一个端口确认目的地是否处于接收允许状态。

(2) 开始发送操作

在发送允许状态，发送操作是通过写入发送数据到 TXBn 寄存器开始的。当开始发送操作时，TXBn 寄存器中的数据被转移到发送移位寄存器。然后发送移位寄存器输出数据到 TXDn 引脚（发送数据从开始位开始按顺序发送）。开始位、奇偶校验位和停止位是自动添加的。

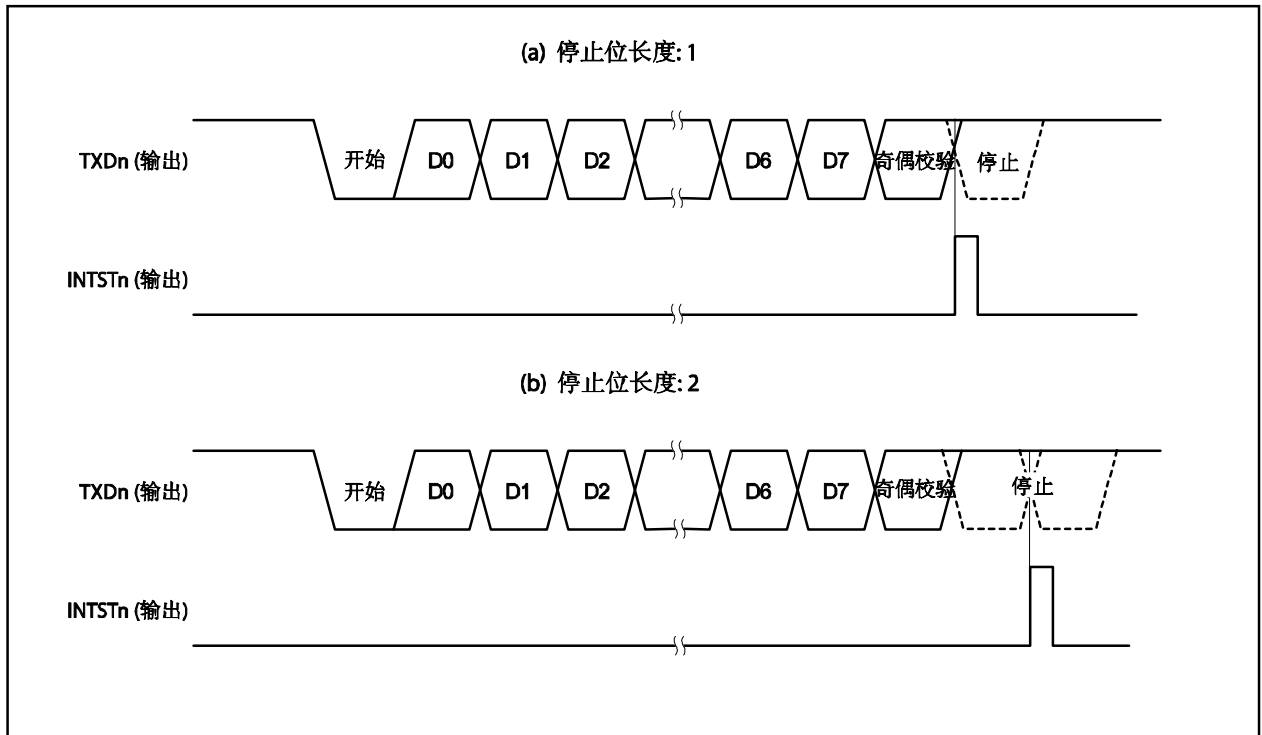
(3) 发送中断

当发送移位寄存器变空时，则生成发送完成中断请求信号 (INTSTn)。INTSTn 信号生成的时序根据停止位的长度指定。INTSTn 信号在输出最后停止位的同时生成。

如果下一个要发送的数据没有写入到 TXBn 寄存器，发送操作挂起。

注意事项 正常情况下，当发送移位寄存器变空时，则生成 INTSTn 信号。但如果发送移位寄存器是由于复位而变空的，则不生成 INTSTn 信号。

图 14-3.UARTn 发送完成中断时序



14.5.3 连续发送操作

UARTn 可以在发送移位寄存器开始移位操作时写入下一个发送数据到 TXBn 寄存器。这样即使在发送一帧数据后发送完成中断服务的过程中也可以实现高效的发送速率。此外，在出现在发送完成中断请求信号 (INTSTn) 读取 ASIFn.TXSFn 位可以使 TXBn 寄存器连续被高效地写入两次而不必等待 1 个数据帧的发送。

当进行连续数据发送时，应在参照 ASIFn 寄存器后写入数据以确认发送状态以及确定是否可以写入数据到 TXBn 寄存器。

注意事项 在连续发送中，ASIF.TXBFn 和 ASIF.TXSFn 位的值变化如下 10 → 11 → 01。
因此，不要基于 TXBFn 和 TXSFn 位的组合确认状态。
在连续发送的过程中仅读取 TXBFn 位。

TXBFn	是否允许写入到 TXBn 寄存器
0	允许写入
1	不允许写入

注意事项 在连续进行发送时，将第一个发送数据（第一个字节）写入到 TXBn 寄存器并确认 TXBFn 位是 0，然后将下一个发送数据（第二个字节）写入到 TXBn 寄存器。如果当 TXBFn 位是 1 时写入到 TXBn 寄存器，则无法保证发送数据。

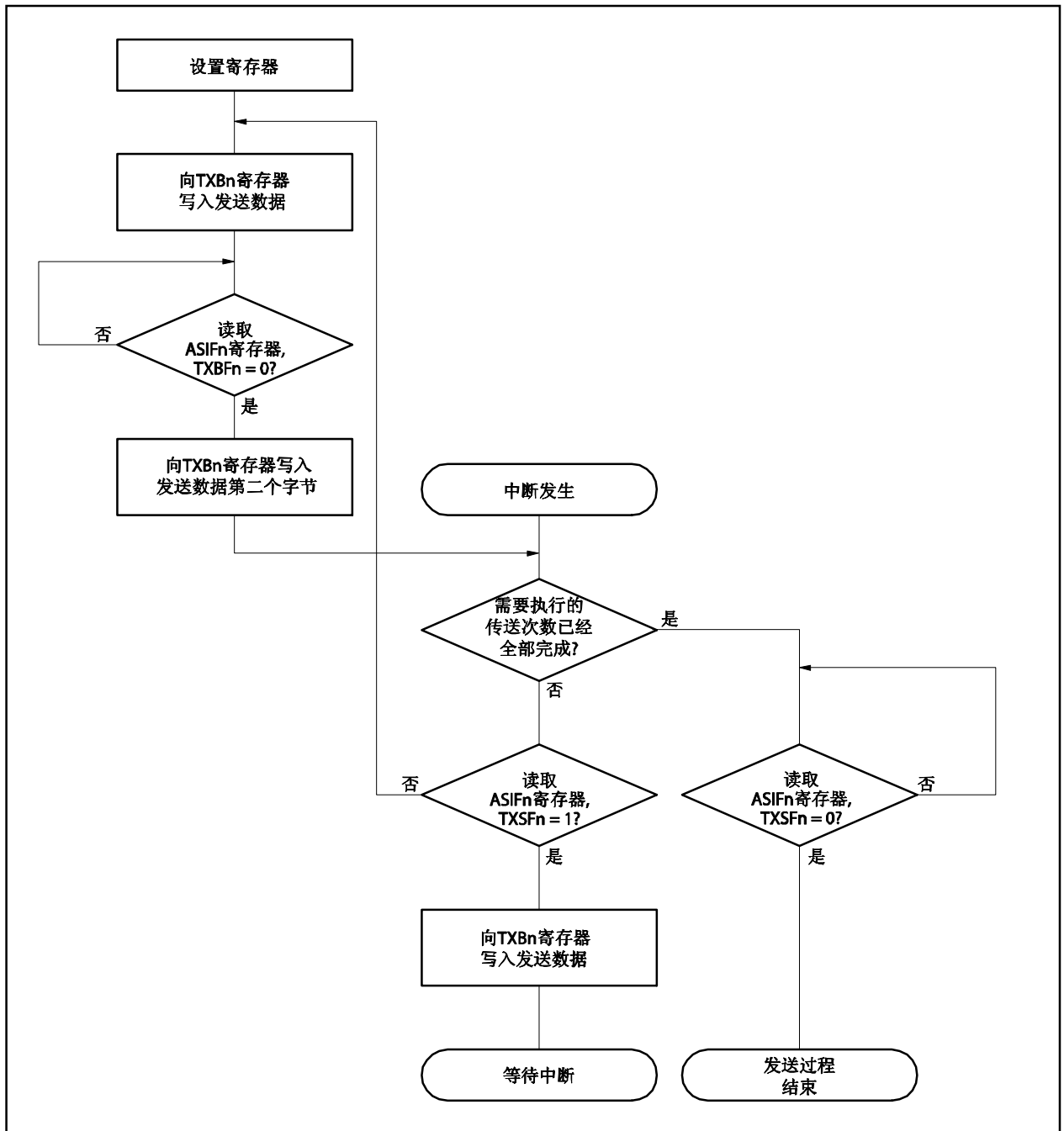
可以参照 TXSFn 位确认通讯状态。

TXSFn	发送状态
0	发送完成
1	发送中

注意事项

1. 在连续发送完成的情况下初始化发送单元时，确认在出现发送完成中断后 TXSFn 位是 0，然后执行初始化。如果在 TXSFn 位是 1 时进行初始化，则无法保证发送数据。
2. 在连续进行发送时，如果在执行一个数据帧的发送后的 INTSTn 中断服务前完成了下一次发送，则可能出现过载错误。可以通过嵌入可以计数发送数据和参照 TXSFn 位次数的程序检测过载错误。

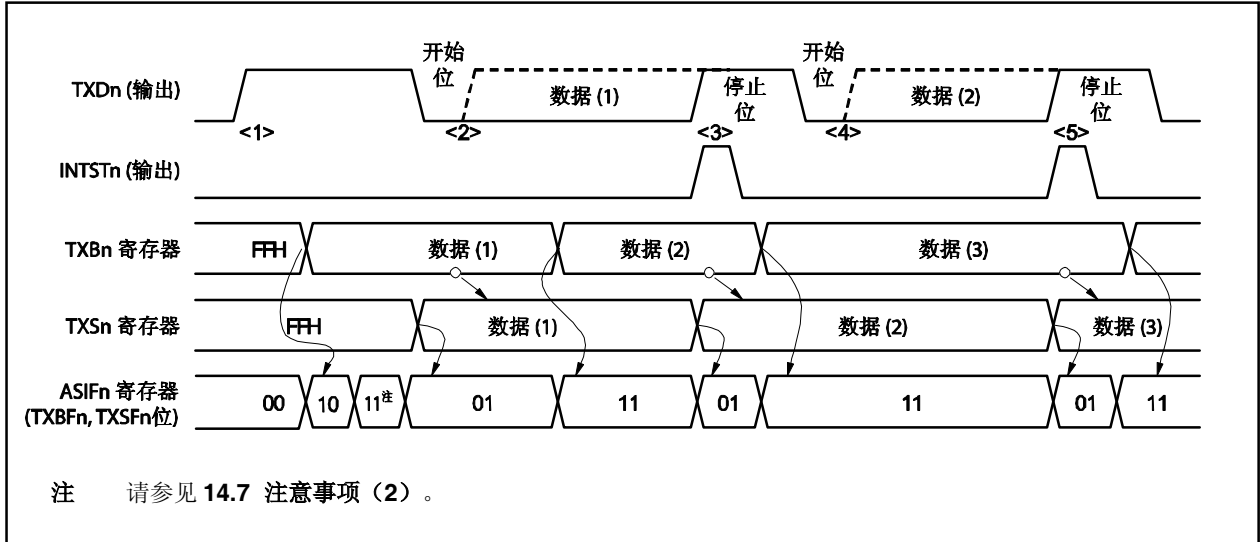
图 14-4. 连续发送流程



(1) 启动程序

启动连续发送的程序如下。

图 14-5. 连续发送启动程序



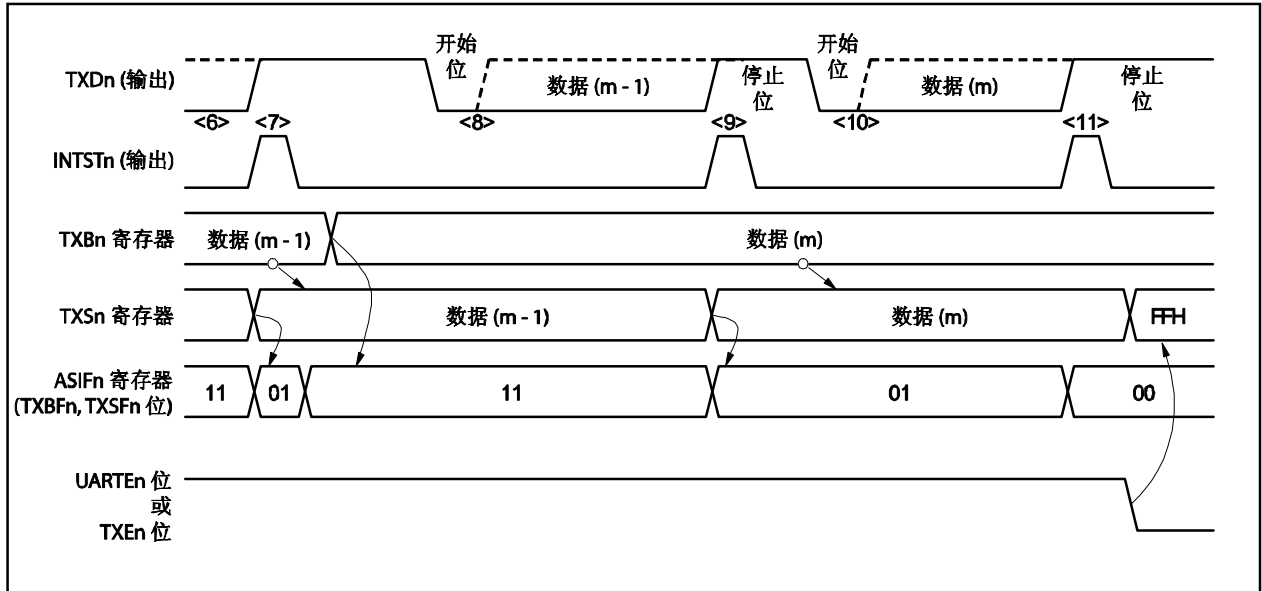
发送开始程序	内部操作	ASIFn 寄存器	
		TXBFn	TXSFn
<ul style="list-style-type: none"> 设置发送模式 写入数据 (1) 	<1> 启动发送单元	0	0
	<2> 生成开始位	1	1*
	开始 (1) 发送	0	1
<ul style="list-style-type: none"> 读取 ASIFn 寄存器 (确认 TXBFn 位 = 0) 	←	<u>0</u>	1
<ul style="list-style-type: none"> 写入数据 (2) 	<<发送进行中>>	1	1
	<3> INTSTn 中断出现	0	1
<ul style="list-style-type: none"> 读取 ASIFn 寄存器 (确认 TXBFn 位 = 0) 	←	<u>0</u>	1
<ul style="list-style-type: none"> 写入数据 (3) 	<<发送进行中>>	1	1
	<4> 生成开始位 开始 (2) 发送 <<发送进行中>>		
	<5> INTSTn 中断出现	0	1
<ul style="list-style-type: none"> 读取 ASIFn 寄存器 (确认 TXBFn 位 = 0) 	←	<u>0</u>	1
<ul style="list-style-type: none"> 写入数据 (4) 		1	1

注 参见 14.7 注意事项 (2)。

(2) 结束程序

结束连续发送的程序如下所示。

图 14-6. 连续发送结束程序



发送结束程序	内部操作	ASIFn 寄存器	
		TXBFn	TXSFn
<ul style="list-style-type: none"> 读取 ASIFn 寄存器 (确认 TXBFn 位 = 0) 写入数据 (m) 	<6> 数据 (m-2) 的发送进行中 <7> INTSTn 中断出现	1	1
	<8> 生成开始位 开始数据 (m-1) 的发送 <<发送进行中>>	0	1
<ul style="list-style-type: none"> 读取 ASIFn 寄存器 (确认 TXSFn 位 = 1) 没有写入数据 	<9> INTSTn 中断出现	0	<u>1</u>
	<10> 生成开始位 开始数据 (m) 的发送 <<发送进行中>>	0	0
<ul style="list-style-type: none"> 读取 ASIFn 寄存器 (确认 TXSFn 位 = 0) 将 UARTEn 位或 TXEn 位清零 	<11> 生成 INTSTn 中断 初始化内部电路	0	<u>0</u>

14.5.4 接收操作

通过设置 `ASIMn.UARTEn` 位为 1 时设置等待接收状态，然后再设置 `ASIMn.RXEn` 位为 1。如要开始接收操作，当检测到 `RxDn` 引脚下降沿时，在下降沿处开始采样。如果在开始位采样点的 `RxDn` 引脚处于低电平，则开始位被识别。当接收操作开始时，串行数据在接收移位寄存器根据设定的波特率按顺序存储。每次接收一个帧的数据完成后生成一个接收完成中断请求信号 (`INTSRn`)。正常情况下，接收数据是通过此中断服务从 `RxBn` 寄存器发送到内存的。

(1) 接收允许状态

通过将 `RXEn` 位设为 1 设置接收操作为接收允许状态。

- `RXEn` 位 = 1: 接收允许状态
- `RXEn` 位 = 0: 接收禁止状态

在接收允许状态，接收硬件在初始状态等待。此时，`RxBn` 寄存器的内容会被保留，不会生成接收完成中断或接收错误中断。

(2) 开始接收操作

接收操作通过检测开始位启动。

`RxDn` 引脚是通过运用串行时钟从波特率生成器 `n` (`BRGn`) 采样的。

(3) 接收完成中断

当 `RXEn` 位 = 1 并完成一个帧的数据的接收时（当检测到停止位时），生成 `INTSRn` 信号，同时接收移位寄存器中的接收数据被转移到 `RxBn` 寄存器。

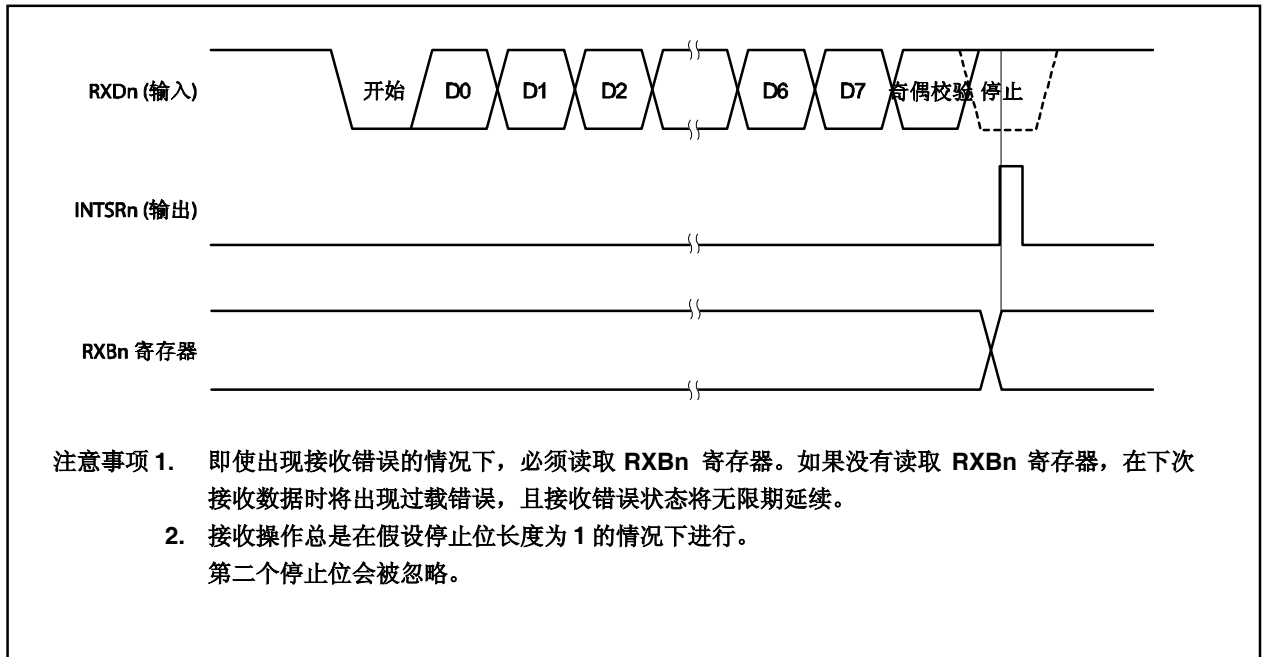
此外，当出现过载错误 (`ASISn.OVEn` 位 = 1) 时，接收数据不转移到 `RxBn` 寄存器，根据 `ASIMn.ISRMn` 位的设置生成 `INTSRn` 信号或接收错误中断请求信号 (`INTSREn`)。

即使在出现接收操作的过程中出现奇偶校验错误 (`ASISn.PEn` 位 = 1) 或帧错误 (`ASISn.FEn` 位 = 1)，接收操作会继续执行直到接收到停止位。在接收完成后，根据 `ISRMn` 位的设置生成 `INTSRn` 信号或 `INTSREn` 信号（接收移位寄存器中的接收数据被转移到 `RxBn` 寄存器）。

如果在接收操作过程中 `RXEn` 位被清零，接收操作立即停止。此时 `RxBn` 寄存器和 `ASISn` 寄存器的内容不会改变，也不生成 `INTSRn` 信号或 `INTSREn` 信号。

当 `RXEn` 位 = 0（禁止接收）时不生成 `INTSRn` 信号或 `INTSREn` 信号。

图 14-7. UARTn 接收完成中断时序



14.5.5 接收错误

在接收操作中可能出现三类错误是奇偶校验错误、帧错误和过载错误。数据接收使得 **ASISn** 寄存器的各种标志被设为 1, 同时生成接收错误中断请求信号 (**INTSREn**) 或接收完成中断请求信号 (**INTSRn**)。 **ASIMn.ISRMn** 位指定是否生成 **INTSREn** 信号或 **INTSRn** 信号。

在接收过程中出现的错误类型可以通过在 **INTSREn** 或 **INTSRn** 中断服务过程中读取 **ASISn** 寄存器的内容检测。可以通过读取 **ASISn** 寄存器将 **ASISn** 寄存器的内容清零。

表 14-3. 接收错误的原因

错误标志	接收错误	原因
PEn	奇偶校验错误	发送中的校验位与接收数据的校验位不相等
FEn	帧错误	没有检测到停止位
OVEEn	过载错误	在从 RXBn 寄存器读取数据前完成了下一个数据的接收

(1) 接收错误中断请求信号分离

可以通过将 ISRMn 位清零把接收错误中断请求信号从 INTSRn 信号中分离出来并生成 INTSREn 信号。

图 14-8. 当从 INTSRn 信号分离接收错误中断请求信号 (ISRMn 位 = 0) 时

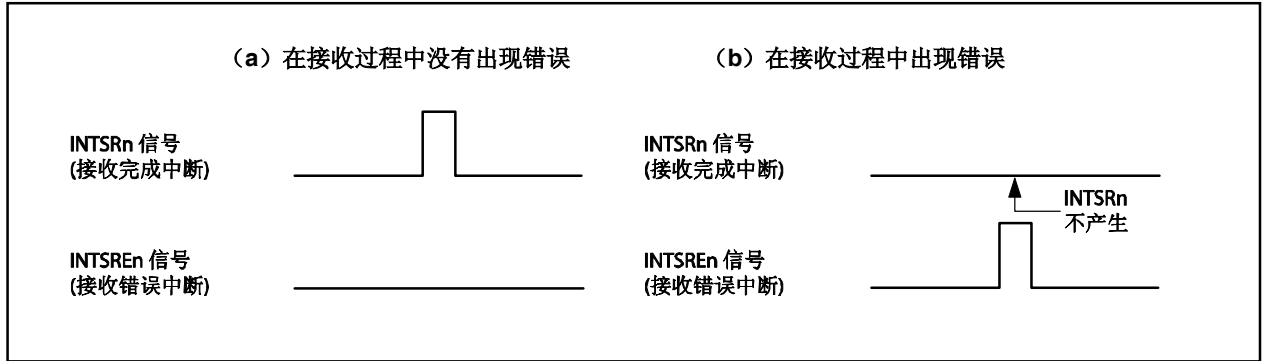
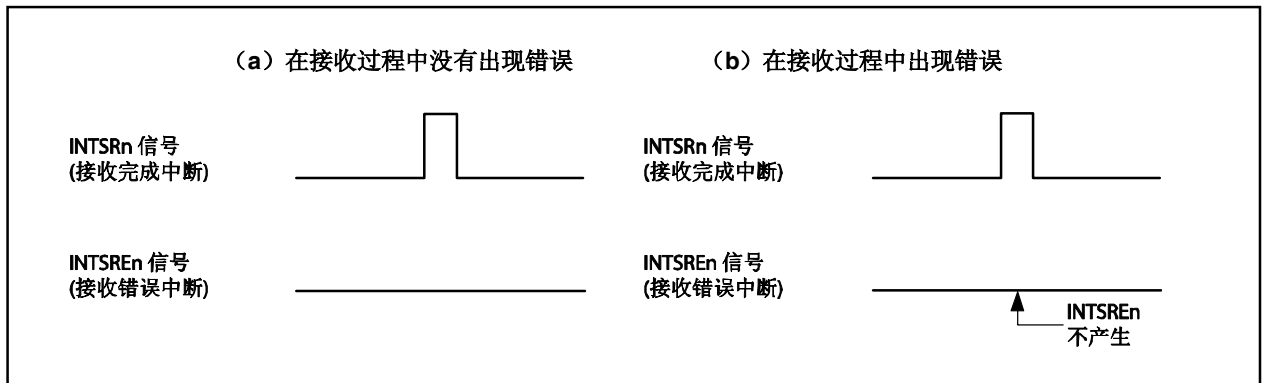


图 14-9. 当接收错误中断请求信号包括在 INTSRn 信号中 (ISRMn 位 = 1) 时



14.5.6 校验位的类型和相应的操作

校验位用于检测通讯数据中的位错误。在正常情况下，发送和接收侧使用相同类型的校验位。

(1) 奇校验

(i) 发送过程中

控制奇数校验位以实现.带有校验位的发送数据中带有值为 1 的位数为奇数。校验位的值如下。

- 如果发送数据中带有值为 1 的位数为奇数： 1
- 如果发送数据中带有值为 1 的位数为偶数： 0

(ii) 接收过程中

计数带有校验位的发送数据中带有值为 1 的位数，如果位数是奇数，则产生奇偶校验错误。

(2) 偶校验

(i) 发送过程中

与偶数校验位相对，控制偶数校验位以实现.带有校验位的发送数据中带有值为 1 的位数为奇数。校验位的值如下。

- 如果发送数据中带有值为 1 的位数为奇数： 0
- 如果发送数据中带有值为 1 的位数为偶数： 1

(ii) 接收过程

计数带有校验位的发送数据中带有值为 1 的位数，如果位数是偶数，则产生奇偶校验错误。

(3) 0 校验

在发送过程中，不论发送数据如何，校验位被设为 0。

在接收过程中，不进行校验位的检查。因此，不论校验位是 0 还是 1，不生成奇偶校验错误。

(4) 无校验

不添加校验位到发送数据。

在接收过程中，如同没有校验位一样进行接收操作，由于没有校验位，不产生奇偶校验错误。

14.5.7 接收数据噪声过滤器

RXDn 信号从预分频器输出基础时钟的 (f_{uclk}) 下降沿采样获得。如果同样的采样值获得了两次，则相等检测器输出改变，对此输出进行采样作为输入数据。因此判定超出 1 个时钟宽度的数据为噪声，不输出到内部电路 (参见图 14-11)。参见关于基础时钟的 14.6.1 (1) 基础时钟。

同时，由于电路如图 14-10 所示，接收操作中的内部处理根据外部信号状态被延迟两个时钟。

图 14-10. 噪声过滤器电路

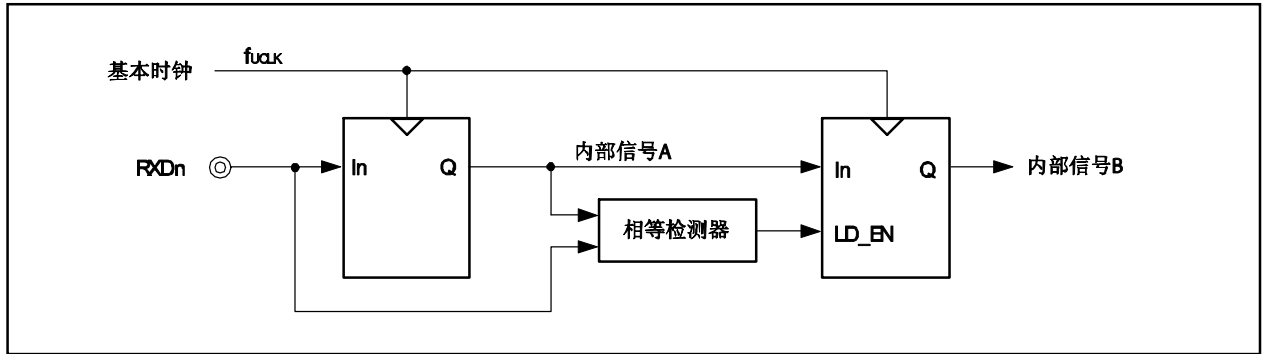
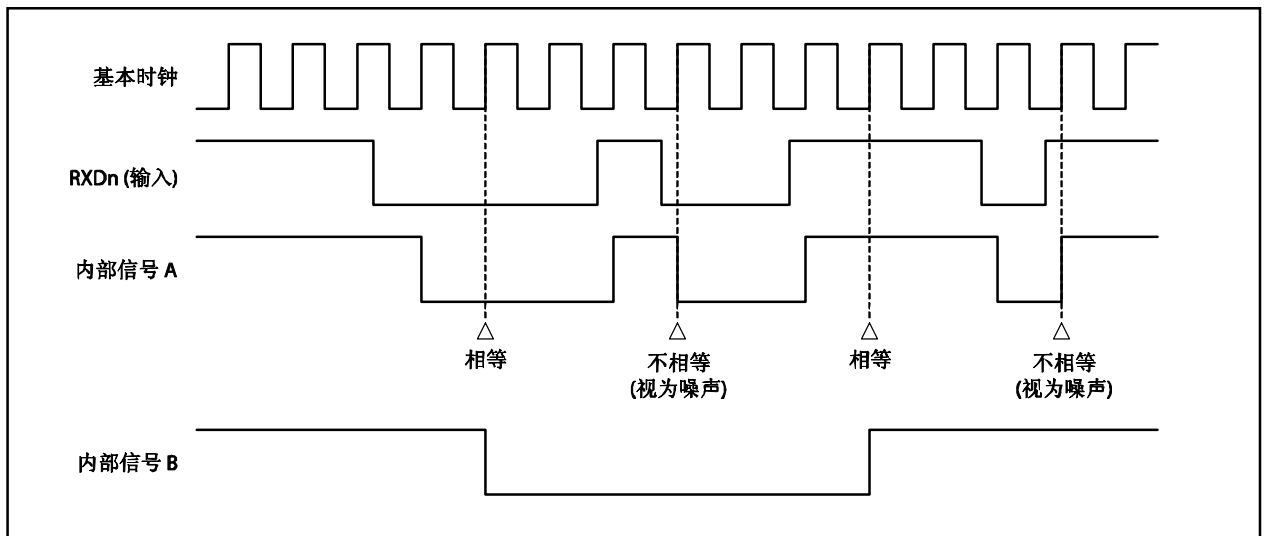


图 14-11. 判定 RXDn 信号为噪声的时序



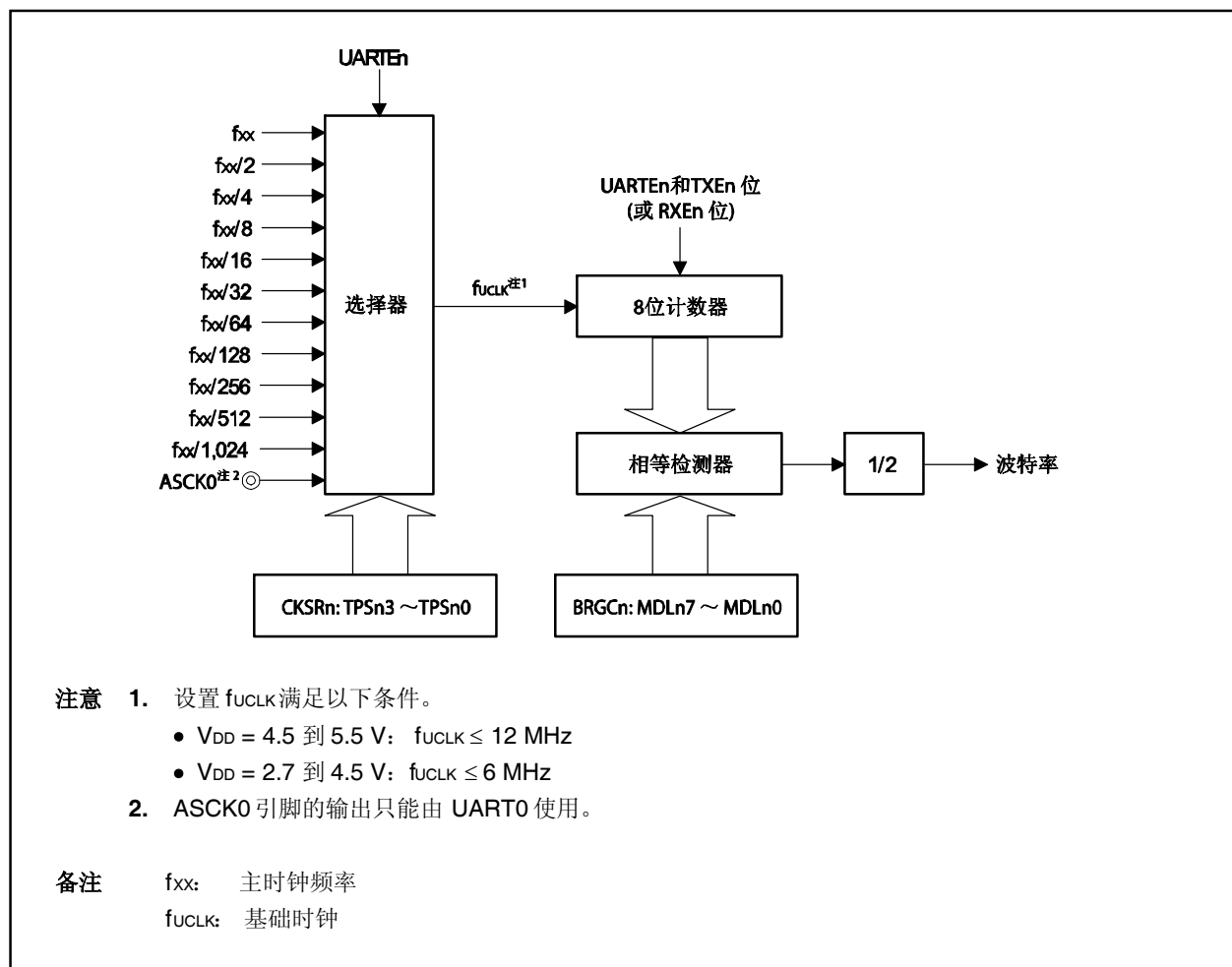
14.6 专门的波特率生成器 n (BRGn)

专门的波特率生成器包括一个源时钟选择器和 8 位可编程计数器，在发送/接收过程中由 UARTn 生成串行时钟。可以为每个通道选择专门的波特率生成器的输出作为串行数据。

有针对发送和针对接收的单独 8 位计数器。

14.6.1 波特率生成器 n (BRGn) 的配置

图 14-12. 波特率生成器 n (BRGn) 的配置



(1) 基础时钟

当 ASIMn.UARTEn 位 = 1 时，根据 CKSRn.TPSn3 到 CKSRn.TPSn0 位选择的时钟被送到发送/接收单元。此时钟称为基础时钟 (f_{uCLK})。当 UARTEn 位 = 0 时， f_{uCLK} 设为低电平。

14.6.2 串行时钟的生成

可以根据 CKSRn 和 BRGCn 寄存器的设置生成串行时钟。

8 位计数器的基础时钟由 CKSRn.TPSn3~CKSRn.TPSn0 位选择。

8 位计数器的除数值可通过 BRGCn.MDLn7~BRGCn.MDLn0 位设置。

(1) 时钟选择 寄存器 n (CKSRn)

CKSRn 寄存器是使用 TPSn3~TPSn0 位选择基础时钟的 8 位寄存器。TPSn3~TPSn0 位选择的时钟成为发送/接收模块的基础时钟 (f_{UCLK})。

该寄存器支持 8 位读写方式。

该寄存器复位后的值为 00H。

注意事项 在重新写入到 TPSn3~TPSn0 位前将 ASIMn.UARTEn 位清零。

复位后: 00H R/W 地址: CKSR0 FFFFA06H, CKSR1 FFFFA16H

	7	6	5	4	3	2	1	0
CKSRn	0	0	0	0	TPSn3	TPSn2	TPSn1	TPSn0

(n = 0, 1)

TPSn3	TPSn2	TPSn1	TPSn0	基础时钟 (f _{UCLK}) ^{※1}
0	0	0	0	f _{xx}
0	0	0	1	f _{xx} /2
0	0	1	0	f _{xx} /4
0	0	1	1	f _{xx} /8
0	1	0	0	f _{xx} /16
0	1	0	1	f _{xx} /32
0	1	1	0	f _{xx} /64
0	1	1	1	f _{xx} /128
1	0	0	0	f _{xx} /256
1	0	0	1	f _{xx} /512
1	0	1	0	f _{xx} /1, 024
1	0	1	1	外部时钟 ^{※2} (ASCK0 引脚)
以上以外的				禁止设置

注意 1. 设置 f_{UCLK} 满足以下条件。

- V_{DD} = 4.5 ~ 5.5 V: f_{UCLK} ≤ 12 MHz。
- V_{DD} = 2.7 ~ 4.5 V: f_{UCLK} ≤ 6 MHz。

2. ASCK0 引脚的输入时钟只能由 UART0 使用。
禁止设置 UART1 和 UART2。

备注 f_{xx}: 主时钟频率

(2) 波特率生成器控制寄存器 n (BRGCn)

BRGCn 寄存器是控制 UARTn 波特率的 (串行发送速度) 8 位寄存器。

该寄存器支持 8 位读写方式。

该寄存器复位后的值为 FFH。

注意事项 如果要重写 MDLn7 到 MDLn0 位, 应首先将 ASIMn.TXEn 和 ASIMn.RXEn 位清零。

复位后: FFH R/W 地址: BRGC0 FFFFA07H, BRGC1 FFFFA17H



MDLn7	MDLn6	MDLn5	MDLn4	MDLn3	MDLn2	MDLn1	MDLn0	设置值 (k)	串行时钟
0	0	0	0	0	×	×	×	—	禁止设置
0	0	0	0	1	0	0	0	8	f _{uclk} /8
0	0	0	0	1	0	0	1	9	f _{uclk} /9
0	0	0	0	1	0	1	0	10	f _{uclk} /10
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	0	1	0	250	f _{uclk} /250
1	1	1	1	1	0	1	1	251	f _{uclk} /251
1	1	1	1	1	1	0	0	252	f _{uclk} /252
1	1	1	1	1	1	0	1	253	f _{uclk} /253
1	1	1	1	1	1	1	0	254	f _{uclk} /254
1	1	1	1	1	1	1	1	255	f _{uclk} /255

- 备注**
1. f_{uclk}: CKSR0.TPSn3~CKSR0.TPSn0 位选择基础时钟的频率[Hz]
 2. k: MDLn7~MDLn0 位 (k = 8, 9, 10, ..., 255) 设置的值
 3. 波特率是 8 位计数器的输出时钟除以 2。
 4. ×: 忽略

(3) 波特率

波特率是通过以下公式获得的值。

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

$f_{\text{uCLK}} = \text{CKSRn.TPSn3} \sim \text{CKSRn.TPSn0}$ 位选择基础时钟的频率[Hz]。

$k = \text{MDLn7} \sim \text{MDLn0}$ 位 ($k = 8, 9, 10, \dots, 255$) 设置的值

(4) 波特率错误

波特率错误是通过以下公式获得的。

$$\text{错误 (\%)} = \left(\frac{\text{实际波特率 (带有错误的波特率)}}{\text{目标波特率 (正常波特率)}} - 1 \right) \times 100 [\%]$$

- 注意事项**
1. 确保发送过程中的波特率错误不超过接收目标的允许错误。
 2. 确保接收过程中的波特率错误在接收过程中允许的波特率范围内，如“14.6.4.接收过程中允许的波特率范围”中所述。

例： 基本时钟频率 = 10 MHz = 10, 000, 000 Hz
 BRGCn.MDLn7~BRGCn.MDLn0 位的设置 = 00100001B ($k = 33$)
 Target 波特率 = 153, 600 bps

$$\begin{aligned} \text{波特率} &= 10, 000, 000 / (2 \times 33) \\ &= 151, 515 [\text{bps}] \end{aligned}$$

$$\begin{aligned} \text{错误} &= (151, 515 / 153, 600 - 1) \times 100 \\ &= -1.357 [\%] \end{aligned}$$

14.6.3 波特率设置示例

表 14-4. 波特率生成器设置数据

波特率 (bps)	f _{xx} = 20 MHz			f _{xx} = 16 MHz			f _{xx} = 10 MHz		
	f _{uCLK}	k	ERR	f _{uCLK}	k	ERR	f _{uCLK}	k	ERR
300	f _{xx} /512	41H (65)	0.16	f _{xx} /1024	1AH (26)	0.16	f _{xx} /256	41H (65)	0.16
600	f _{xx} /256	41H (65)	0.16	f _{xx} /1024	0DH (13)	0.16	f _{xx} /128	41H (65)	0.16
1200	f _{xx} /128	41H (65)	0.16	f _{xx} /512	0DH (13)	0.16	f _{xx} /64	41H (65)	0.16
2400	f _{xx} /64	41H (65)	0.16	f _{xx} /256	0DH (13)	0.16	f _{xx} /32	41H (65)	0.16
4800	f _{xx} /32	41H (65)	0.16	f _{xx} /128	0DH (13)	0.16	f _{xx} /16	41H (65)	0.16
9600	f _{xx} /16	41H (65)	0.16	f _{xx} /64	0DH (13)	0.16	f _{xx} /8	41H (65)	0.16
10400	f _{xx} /64	0FH (15)	0.16	f _{xx} /64	0CH (12)	0.16	f _{xx} /32	0FH (15)	0.16
19200	f _{xx} /8	41H (65)	0.16	f _{xx} /32	0DH (13)	0.16	f _{xx} /4	41H (65)	0.16
24000	f _{xx} /32	0DH (13)	0.16	f _{xx} /2	A7H (167)	-0.20	f _{xx} /16	0DH (13)	0.16
31250	f _{xx} /32	0AH (10)	0.00	f _{xx} /32	08H (8)	0.00	f _{xx} /16	0AH (10)	0
33600	f _{xx} /2	95H (149)	-0.13	f _{xx} /2	77H (119)	0.04	f _{xx}	95H (149)	-0.13
38400	f _{xx} /4	41H (65)	0.16	f _{xx} /16	0DH (13)	0.16	f _{xx} /2	41H (65)	0.16
48000	f _{xx} /16	0DH (13)	0.16	f _{xx} /2	53H (83)	0.40	f _{xx} /8	0DH (13)	0.16
56000	f _{xx} /2	59H (89)	0.32	f _{xx} /2	47H (71)	0.60	f _{xx}	59H (89)	0.32
62500	f _{xx} /16	0AH (10)	0.00	f _{xx} /16	08H (8)	0.00	f _{xx} /8	0AH (10)	0.00
76800	f _{xx} /2	41H (65)	0.16	f _{xx} /8	0DH (13)	0.16	f _{xx}	41H (65)	0.16
115200	f _{xx} /2	2BH (43)	0.94	f _{xx} /2	23H (35)	-0.79	f _{xx}	2BH (43)	0.94
153600	f _{xx} /2	21H (33)	-1.36	f _{xx} /4	0DH (13)	0.16	f _{xx}	21H (33)	-1.36
312500	f _{xx} /4	08H (8)	0	f _{xx} /2	0DH (13)	-1.54	f _{xx} /2	08H (8)	0.00

注意事项 基础时钟 (f_{uCLK}) 的允许频率如下。

- V_{DD} = 4.5~5.5 V: f_{uCLK} ≤ 12 MHz
- V_{DD} = 2.7~4.5 V: f_{uCLK} ≤ 6 MHz

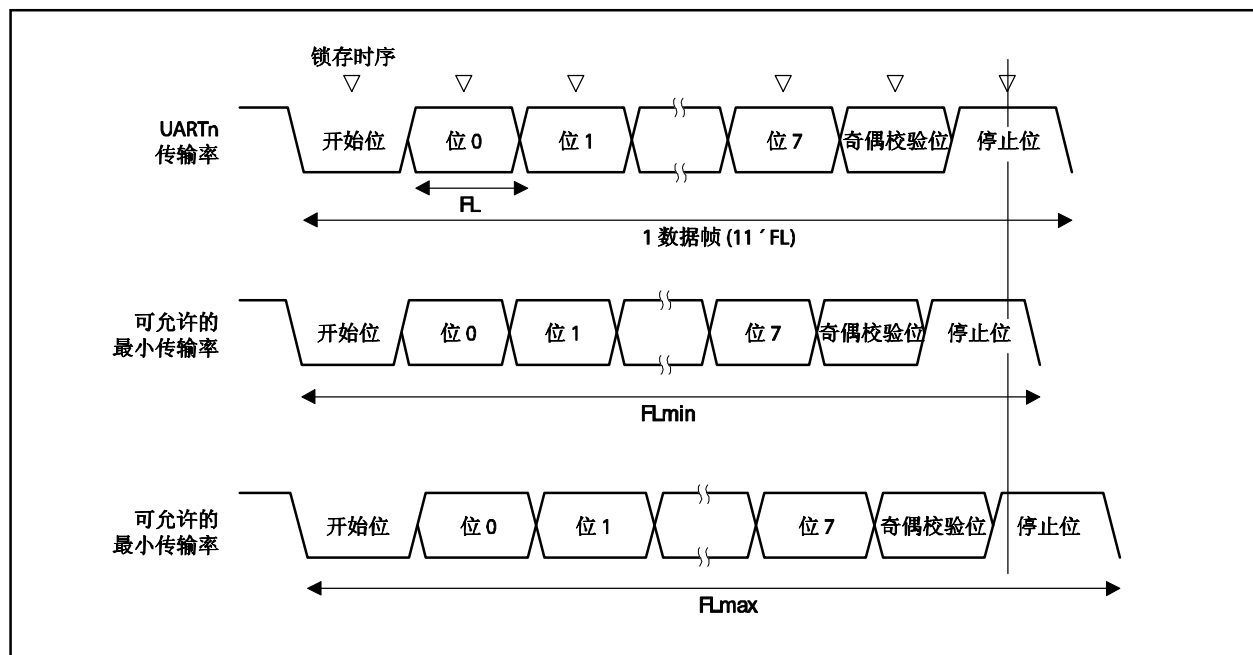
备注 f_{xx}: 主时钟频率
f_{uCLK}: 基本时钟频率
k: BRGCn.MDLn7~BRGCn.MDLn0 位设定的值
ERR: 波特率错误[%]
n = 0~2

14.6.4 接收过程中允许的波特率范围

对在接收过程中允许的发送目标的波特率的偏差程度如下所示。

注意事项 下述的等式应用于设定接收过程的波特率，使得其始终在允许的错误范围内。

图 14-13. 接收过程中允许的波特率范围



如图 14-13 所示，在检测到开始位后，接收数据的锁存时间裕度是根据 BRGCn 寄存器设置的计数器确定的。如果一直到最终数据（停止位）的所有数据在锁宽限的范围内，可以正常接收数据。

如果此应用到 11 位的接收，以下公式在理论上正确。

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

Brate: UARTn 波特率

K: BRGCn 寄存器设定的值

FL: 1-位数据长度

当锁存时间裕度是两个基础时钟时，最低允许的发送速率 (FLmin) 如下。

$$FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{2k+2}{2k} FL$$

因此，发送目标的最大可接收的波特率 (BRmax) 如下。

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

类似的，最大允许的发送速率 (FLmax) 可以通过以下等式获得。

$$\begin{aligned} \frac{10}{11} \times FL_{max} &= 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL \\ FL_{max} &= \frac{21k - 2}{20k} FL \times 11 \end{aligned}$$

因此，发送目标的最低可接收的波特率 (BRmin) 如下。

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

UARTn 的允许波特率错误和发送目标可以根据上述计算最高和最低波特率值的公式如下得出。

表 14-5. 最高和最低允许的波特率错误

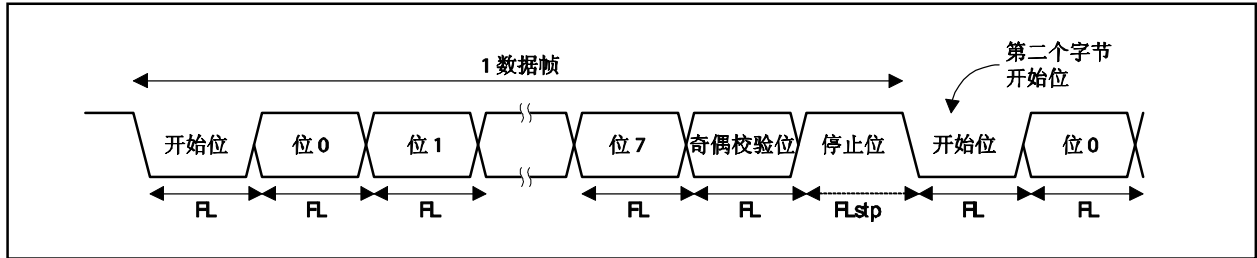
分频比 (k)	最高允许的波特率错误	最低允许的波特率错误
8	+3.53%	-3.61%
20	+4.26%	-4.31%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.73%

- 备注**
1. 接收精确度取决于一个帧中的位数、基本时钟频率、分频比 (k)。基本时钟频率越大和分频比 (k) 越高，精确度越高。
 2. k: BRGCn 寄存器设置值

14.6.5 连续发送过程中的发送速率

在连续发送的过程中，从一个停止位到下一个开始位的发送率比正常情况下延长两个基础时钟。在接收侧，发送结果不受影响，因为时序是通过检测开始位初始化的。

图 14-14. 连续发送过程中的发送速率



用 FL 代表 1 位数据的长度、 FL_{stp} 代表停止位长度、 f_{uCLK} 代表基本时钟频率，得出以下公式。

$$FL_{stp} = FL + 2/f_{uCLK}$$

因此，连续发送过程中的发送速率如下（当停止位长度=1 时）。

$$\text{发送速率} = 11 \times FL + (2/f_{uCLK})$$

14.7 注意事项

使用 $UARTn$ 时的注意事项如下所述。

- (1) 当向 $UARTn$ 供应时钟停止时（比如在空闲或停止模式），操作停止，每个寄存器保留在停止供应时钟前的值。 $TXDn$ 引脚输出同样保留和输出在停止提供时钟前的值。在时钟供应重新启动后，则不能保证操作。因此，在供应重新启动后，应通过将 $ASIMn.UARTEn$ 、 $ASIMn.RXEn$ 和 $ASIMn.TXEn$ 位清为 000。
- (2) $UARTn$ 有两个阶段的缓冲配置，包括 $TXBn$ 寄存器和发送移位寄存器以及显示每个缓冲器状态的状态标志 ($ASIFn.TXBFn$ 和 $ASIFn.TXSFn$ 位)。如果 $TXBFn$ 和 $TXSFn$ 位在连续发送中读取，值的变化如下。10 → 11 → 01。对于写入下一个数据到 $TXBn$ 寄存器的时序，在连续发送过程中仅读取 $TXBFn$ 位。

第十五章 同步串行接口 0 (CSIO)

在 V850ES/KE2 中提供了两个同步串行接口 (CSIO) 的通道。

15.1 特点

- 最大传输速度: 5 Mbps
- 主模式/从模式可选择
- 传输数据长度: 8 位或 16 位均可设置
- 可为传输数据选择最高有效位 (MSB) /最低有效位 (LSB)
- 可以选择八个时钟信号 (七个主时钟和一个从时钟)
- 三线型: SO0n: 串行传输数据输出
 SI0n: 串行接收数据输入
 SCK0n: 串行时钟输入/输出
- 输入源: 1 型
 - 传输/接收完成中断请求信号 (INTCSI0n)
- 可选择传输/接收模式或仅接收模式
- 在芯片上提供两个传输缓冲寄存器 (SOTBFn/SOTBFLn 和 SOTBn/SOTBLn) 和两个接收缓冲寄存器 (SIRBn/SIRBLn 和 SIRBEn/SIRBELn)
- 可选择单传输模式/连续传输模式

备注: n = 0, 1

15.2 配置

CSI0n 通过 CSIM0n 寄存器控制。

(1) 同步串行接口模式寄存器 0n (CSIM0n)

CSIM0n 寄存器是一个指定 CSIM0n 运行的 8 位寄存器。

(2) 同步串行接口时钟选择寄存器 n (CSICn)

CSICn 寄存器是一个控制 CSICn 串行传输运行的 8 位寄存器。

(3) 串行输入/输出移位寄存器 0n (SIO0n)

SIO0n 寄存器是将并行数据转换为串行数据的 16 位移位寄存器。

SIO0n 寄存器可用于传输和接收。

数据从最高有效位或最低有效位侧移入（接收）和移出（传输）的。
实际的传输/接收操作是通过访问缓冲寄存器启动的。

(4) 串行输出/出入移位寄存器 0nL (SIO0nL)

SIO0nL 寄存器是将并行数据转换为串行数据的 8 位移位寄存器。

SIO0nL 寄存器可用于传输和接收。

数据从最高有效位或最低有效位侧移入（接收）和移出（传输）的。
实际的传输/接收操作是通过访问缓冲寄存器启动的。

(5) 同步串行接口接收缓冲寄存器 n (SIRBn)

SIRBn 寄存器是存储接收数据的 16 位缓冲寄存器。

(6) 同步串行接口接收缓冲寄存器 nL (SIRBnL)

SIRBnL 寄存器是存储接收数据的 8 位缓冲寄存器。

(7) 同步串行接口只读接收缓冲寄存器 n (SIRBEn)

SIRBEn 寄存器是存储接收数据的 16 位缓冲寄存器。

SIRBEn 寄存器与 SIRBn 寄存器相同，用于读取 SIRBn 寄存器的内容。

(8) 同步串行接口只读接收缓冲寄存器 nL (SIRBEnL)

SIRBEnL 寄存器是存储接收数据的 8 位缓冲寄存器。

SIRBEnL 寄存器与 SIRBnL 寄存器相同，用于读取 SIRBnL 寄存器的内容。

(9) 同步串行接口传输缓冲寄存器 n (SOTBn)

SOTBn 寄存器是存储传输数据的 16 位缓冲寄存器。

(10) 同步串行接口传输缓冲寄存器 nL (SOTBnL)

SOTBnL 寄存器是存储传输数据的 8 位缓冲寄存器。

(11) 同步串行接口初始传输缓冲寄存器 n (SOTBFn)

SOTBFn 寄存器是存储连续传输模式下的初始传输数据的 16 位缓冲寄存器。

(12) 同步串行接口初始传输缓冲寄存器 nL (SOTBFnL)

SOTBFnL 寄存器是存储连续传输模式下的初始传输数据的 8 位缓冲寄存器。

(13) 选择器

选择器选择要使用的串行数据。

(14) 串行时钟控制器

控制移位寄存器的串行时钟供应，还控制在使用内部时钟时到 $\overline{\text{SCK0n}}$ 引脚的时钟输出。

(15) 串行时钟计数器

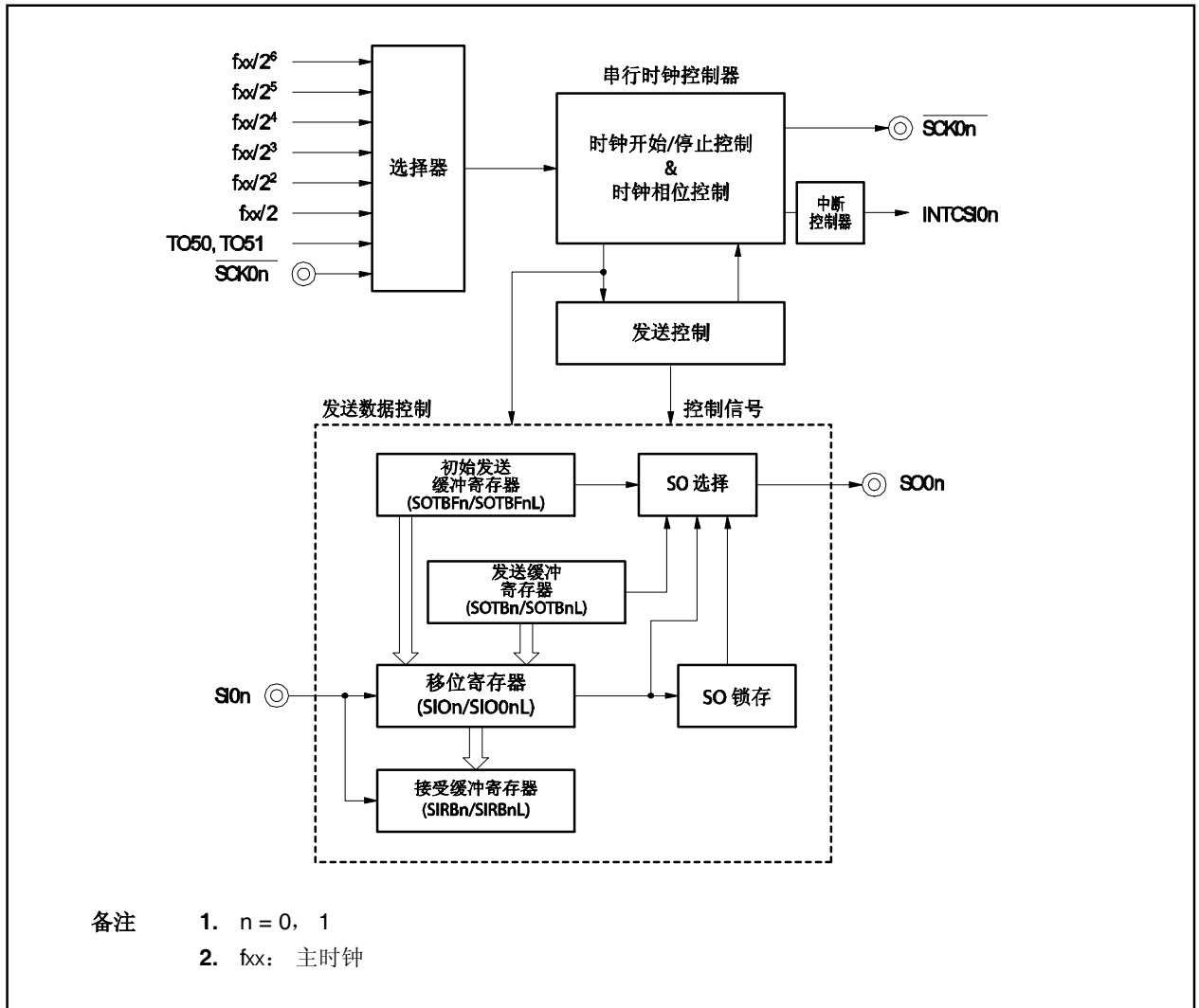
计数传输和接收过程中的串行输出或输入，检查是否进行了 8 位或 16 位数据传输和接收。

(16) 中断控制器

控制中断请求时序。

备注 n = 0, 1

图 15-1. 同步串行接口框图



15.3 寄存器

(1) 同步串行接口模式寄存器 0n (CSIM0n)

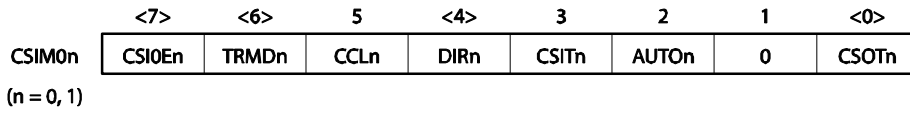
CSIM0n 寄存器控制 CSIOn 的运行。

该寄存器支持 8 位或 1 位读写方式（但 CSOTn 位是只读的）。

CSIM0n 该寄存器复位后的值为 00H。

注意事项 只有当 CSOTn 位 = 0 时才能重写 CSIM0n.TRMDn、CSIM0n.CCLn、CSIM0n.DIRn、CSIM0n.CSITn 和 CSIM0n.AUTOn 位。如果在其他时间重写了这些位，则不能保证运行。

复位后: 00H R/W 地址: CSIM00 FFFFFFFD00H, CSIM01 FFFFFFFD10H



CSI0En	CSI0n 运行允许/禁止
0	禁止 CSI0n 运行。
1	允许 CSI0n 运行。
可以通过将 CSI0En 位清零同步复位*内部 CSI0n 电路。关于当 CSI0En 位=0时的 SCK0n 和 SO0n 引脚输出状态，参见 15.5 输出引脚。	

TRMDn	传输/接收模式的指定
0	只接收模式
1	传输/接收模式
当 TRMDn 位 = 0 时，执行接收操作，SO0n 引脚输出一个低电平。数据接收是通过读取 SIRBn 寄存器开始的。 当 TRMDn 位 = 1 时，通过写入数据到 SOTBn 寄存器开始传输/接收。	

CCLn	数据长度的指定
0	8 位
1	16 位

DIRn	传输方向模式的指定 (最低有效位/最高有效位)
0	传输数据的第一个位是最高有效位
1	传输数据的第一个位是最低有效位

CSITn	中断请求信号延迟的控制
0	无延迟
1	延迟模式 (相对于串行时钟，中断请求信号会延迟 1/2 个周期)
只有在主模式 (CSICn.CKS0n2 到 CSICn.CSK0n0 位不是 111B) 下延迟模式 (CSITn 位 = 1) 才有效。在从模式 (CKS0n2 到 CKS0n0 位是 111B) 下不要设置延迟模式。	

AUTOn	单传输模式或连续传输模式的指定
0	单传输模式
1	连续传输模式

CSOTn	通讯状态标记
0	通讯停止
1	通讯进行中
通过将 0 写入到 CSI0En 位将 CSOTn 位清零。	

注 复位 CSOTn 位和 SIRBn、SIRBnL、SIRBE、SIRBE nL、SIO n 和 SIO nL 寄存器。

(2) 同步串行接口时钟选择寄存器 n (CSICn)

CSICn 寄存器是一个控制 CSI0n 传输运行的 8 位寄存器。
 该寄存器支持 8 位或 1 位读写方式。
 CSICn 寄存器复位后的值为 00H。

注意事项 只有在 CSIM0n.CSI0En 位=0 时才能重写 CSICn 寄存器。

复位后: 00H R/W 地址: CSIC0 FFFFD01H, CSIC1 FFFFD11H

7	6	5	4	3	2	1	0
0	0	0	CKPn	DAPn	CKS0n2	CKS0n1	CKS0n0

(n = 0, 1)

CKPn	DAPn	从 SCK0n 接收数据或向 SCK0n 传输数据时序的指定
0	0	(1 型)
0	1	(2 型)
1	0	(3 型)
1	1	(4 型)

CKS0n2	CKS0n1	CKS0n0	串行时钟 ^a	模式
0	0	0	$f_{xx}/2$	主模式
0	0	1	$f_{xx}/2^2$	主模式
0	1	0	$f_{xx}/2^3$	主模式
0	1	1	$f_{xx}/2^4$	主模式
1	0	0	$f_{xx}/2^5$	主模式
1	0	1	$f_{xx}/2^6$	主模式
1	1	0	TO5n 生成的时钟	主模式
1	1	1	外部时钟 (SCK0n 引脚)	从模式

注 设置系列时钟以满足以下条件。

- $V_{DD} = 4.0 \sim$ 到 5.5 V : 串行时钟 $\leq 5 \text{ MHz}$
- $V_{DD} = 2.7 \sim 4.0 \text{ V}$: 串行时钟 $\leq 2.5 \text{ MHz}$

备注 f_{xx} : 主时钟频率

(3) 同步串行接口接收缓冲寄存器 n, nL (SIRBn, SIRBnL)

SIRBn 寄存器是存储接收数据的 16 位缓冲寄存器。

当设置只接收模式时 (CSIM0n.TRMDn 位 = 0)，通过从 SIRBn 寄存器读取数据开始接收操作。

该寄存器支持 16 位只读方式。但当下 8 位用于 SIRBnL 寄存器时，该寄存器支持 8 位只读方式。

除了复位输入外，可以通过清零 CSIM0n.CSI0En 位初始化该寄存器。

注意事项 1. 只有当设置了 16 位数据长度 (CSIM0n.CCLn 位 = 1) 时才读取 SIRBn 寄存器。

只有当设置了 8 位数据长度 (CCLn 位 = 0) 时才读取 SIRBnL 寄存器。

2. 当设置了单传输模式时 (CSIM0n.AUTOn 位 = 0)，只有在空闲状态 (CSIM0n.CSOTn 位 = 0) 才进行读取操作。如果在数据传输中读取了 SIRBn 或 SIRBnL 寄存器，则不能保证数据。

(a) SIRBn 寄存器

地址: 0000H R 地址: SIRB0 FFFFFFFD02H, SIRB1 FFFFFFFD12H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRBn	SIRBn	SIRBn	SIRBn	SIRBn	SIRBn	SIRBn	SIRBn	SIRBn	SIRBn	SIRBn	SIRBn	SIRBn	SIRBn	SIRBn	SIRBn	SIRBn
(n = 0, 1)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

(b) SIRBnL 寄存器

复位后: 00H R 地址: SIRB0L FFFFFFFD02H, SIRB1L FFFFFFFD12H

	7	6	5	4	3	2	1	0
SIRBnL	SIRBn7	SIRBn6	SIRBn5	SIRBn4	SIRBn3	SIRBn2	SIRBn1	SIRBn0
(n = 0, 1)								

(4) 同步串行接口只读接收缓冲寄存器 n, nL (SIRBEn, SIRBEnL)

SIRBEn 寄存器是存储接收数据的 16 位缓冲寄存器。

SIRBEn 寄存器与 SIRBn 寄存器相同。即使读取了 SIRBEn 寄存器，也不会开始下一个操作。当没有继续串行接收时，SIRBEn 寄存器用于读取 SIRBn 寄存器的内容。

该寄存器支持 16 位只读方式。但当下 8 位用于 SIRBEnL 寄存器时，该寄存器支持 8 位只读方式。

除了复位输入外，可以通过清零 CSIM0n.CSI0En 位初始化该寄存器。

注意事项 1. 即使从 SIRBEn 和 SIRBEnL 寄存器读取了数据也不开始接收操作。

2. 只有当设置了 16 位数据长度 (CSIM0n.CCLn 位 = 1) 时才读取 SIRBEn 寄存器。

只有当设置了 8 位数据长度 (CCLn 位 = 0) 时才读取 SIRBEnL 寄存器。

(a) SIRBEn 寄存器

复位后: 0000H R 地址: SIRBE0 FFFFD06H, SIRBE1 FFFFD16H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRBEn	SIRBEn	SIRBEn	SIRBEn	SIRBEn	SIRBEn	SIRBEn	SIRBEn	SIRBEn	SIRBEn	SIRBEn	SIRBEn	SIRBEn	SIRBEn	SIRBEn	SIRBEn	SIRBEn
(n = 0, 1)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

(B) SIRBEnL 寄存器

复位后: 00H R 地址: SIRBE0L FFFFD06H, SIRBE1L FFFFD16H

	7	6	5	4	3	2	1	0
SIRBEnL	SIRBEn7	SIRBEn6	SIRBEn5	SIRBEn4	SIRBEn3	SIRBEn2	SIRBEn1	SIRBEn0
(n = 0, 1)								

(5) 同步串行接口传输缓冲寄存器 n, nL (SOTBn, SOTBnL)

SOTBn 寄存器是存储传输数据的 16 位缓冲寄存器。

当设置传输/接收模式 (CSIM0n.TRMDn 位 = 1) 时, 通过写入数据到 SOTBn 寄存器开始传输操作。

该寄存器支持 16 位只读方式。但当下 8 位用于 SOTBnL 寄存器时, 该寄存器支持 8 位只读方式。

该寄存器在初始化后复位。

注意事项 1. 只有当设置了 16 位数据长度 (CSIM0n.CCLn 位 = 1) 时才读取 SOTBn 寄存器。

只有当设置了 8 位数据长度 (CCLn 位 = 0) 时才读取 SOTBnL 寄存器。

2. 当设置了单传输模式时 (CSIM0n.AUTOn 位 = 0), 只有在空闲状态 (CSIM0n.CSOTn 位 = 0) 才进行访问。如果在数据传输中访问了 SOTBn 和 SOTBnL 寄存器, 则不能保证数据。

(a) SOTBn 寄存器

复位后: 0000H R/W 地址: SOTB0 FFFFFFFD04H, SOTB1 FFFFFFFD14H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOTBn	SOTBn	SOTBn	SOTBn	SOTBn	SOTBn	SOTBn	SOTBn	SOTBn	SOTBn	SOTBn	SOTBn	SOTBn	SOTBn	SOTBn	SOTBn	SOTBn
(n = 0, 1)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

(b) SOTBnL 寄存器

复位后: 00H R/W 地址: SOTB0L FFFFFFFD04H, SOTB1L FFFFFFFD14H

	7	6	5	4	3	2	1	0
SOTBnL	SOTBn7	SOTBn6	SOTBn5	SOTBn4	SOTBn3	SOTBn2	SOTBn1	SOTBn0
(n = 0, 1)								

(6) 同步串行接口初始传输寄存器 n, nL (SOTBFn, SOTBFnL)

SOTBFn 寄存器是存储连续传输模式中初始传输数据的 16 位缓冲寄存器。

即使写入数据到 SOTBFn 寄存器也不会开始传输操作。

该寄存器支持 16 位只读方式。但当下 8 位用于 SOTBFnL 寄存器时，该寄存器支持 8 位只读方式。

该寄存器在初始化后复位。

注意事项 只有当设置了 16 位数据长度 (CSIM0n.CCLn 位 = 1) 时，设置了 8 位数据长度 (CCLn 位 = 0) 时，或者只有在空闲状态 (CSIM0n.CSOTn 位 = 0) 才访问 SOTBFn 和 SOTBFnL 寄存器。如果在数据传输过程中访问了 SOTBFn 和 SOTBFnL 寄存器，则不能保证数据。

(a) SOTBFn 寄存器

复位后: 0000H R/W 地址: SOTBF0 FFFFFFFD08H, SOTBF1 FFFFFFFD18H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOTBFn	SOTBFn	SOTBFn	SOTBFn	SOTBFn	SOTBFn	SOTBFn	SOTBFn	SOTBFn	SOTBFn	SOTBFn	SOTBFn	SOTBFn	SOTBFn	SOTBFn	SOTBFn	SOTBFn
(n = 0, 1)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

(b) SOTBFnL 寄存器

复位后: 00H R/W 地址: SOTBF0L FFFFFFFD08H, SOTBF1L FFFFFFFD18H

	7	6	5	4	3	2	1	0
SOTBFnL	SOTBFn7	SOTBFn6	SOTBFn5	SOTBFn4	SOTBFn3	SOTBFn2	SOTBFn1	SOTBFn0
(n = 0, 1)								

(7) 串行输入/输出移位寄存器 n, nL (SIO0n, SIO0nL)

SIO0n 寄存器是将并行数据转换为串行数据的 16 位移位寄存器。

即使读取了 SIO0n 寄存器也不会开始传输操作。

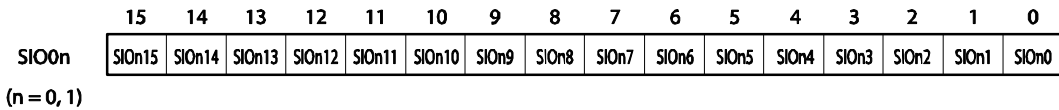
该寄存器支持 16 位只读方式。但当下 8 位用于 SIO0n 寄存器时，该寄存器支持 8 位只读方式。

除了复位输入外，可以通过清零 CSIM0n.CSIOEn 位初始化该寄存器。

注意事项 只有当设置了 16 位数据长度 (CSIM0n.CCLn 位 = 1) 时，设置了 8 位数据长度 (CCLn 位 = 0) 时，或者只有在空闲状态 (CSIM0n.CSOTn 位 = 0) 才读取 SIO0n 寄存器和 SIO0nL 寄存器。如果在数据传输过程中读取了 SIO0n 寄存器和 SIO0nL 寄存器，则不能保证数据。

(a) SIO0n 寄存器

复位后: 0000H R 地址: SIO00 FFFFFFFD0AH, SIO01 FFFFFFFD1AH



(b) SIO0nL 寄存器

复位后: 00H R 地址: SIO00L FFFFFFFD0AH, SIO01L FFFFFFFD1AH

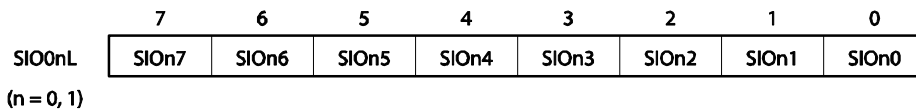


表 15-1. 每个缓冲寄存器的使用

寄存器名称	R/W	单传输		连续传输 ^{注1}	
		传输/接收模式 ^{注2}	仅接收模式	传输/接收模式	仅接收模式
SIRBn (SIRBnL)	读	储存接收到的数据 ^{注2}	<ul style="list-style-type: none"> 读取开始接收 储存接收到的数据 	储存到 (N-1) 个为止的接收到的数据 ^{注2} (除了最后两个数据以外的数据)	<ul style="list-style-type: none"> 读取开始接收 储存到 (N-2) 个为止的接收到的数据^{注2} (除了最后两个数据以外的数据)
		当传输和接收完成时, 从该寄存器读取接收到的数据。	<ul style="list-style-type: none"> 先读取虚拟数据, 然后开始传输。 要在接收完成后接收下一个数据, 从该寄存器读取接收到的数据。 	当接收完成后, 从该寄存器读取接收到的数据。重复此操作直至接收了第 (N-2) 个数据。 (补充) 不要从该寄存器读取第 (N-1) 个数据。如果读取了则会开始接收操作而无法完成连续传输。	
SIRBEn (SIRBEnL)	读	-	储存接收到的数据 ^{注2}	-	储存第 (N-1) 个接收到的数据 ^{注2}
		未使用。	如果在接收完成后不再接收下一个数据, 从该寄存器读取接收到的数据。	未使用	当收到第 (N-1) 或 N (最后一个) 个数据后, 从该寄存器读取第 N 个 (最后一个) 接收到的数据。
SIO0n (SIO0nL)	读	-	-	储存第 N 个 (最后一个) 接收到的数据 ^{注2}	储存第 N 个 (最后一个) 接收到的数据 ^{注2}
		未使用。	未使用。	当第 N 次 (最后一次) 传输/接收完成时, 读取第 N 次 (最后一次) 的数据。	当收到第 N 个 (最后一个) 数据时, 读取第 N 个 (最后一个) 数据。
SOTBn (SOTBnL)	写	<ul style="list-style-type: none"> 在写入时开始传输/接收。 储存要传输的数据。 	-	<ul style="list-style-type: none"> 在写入时开始传输/接收 储存要传输的第二个及以后的数据 	-
		<ul style="list-style-type: none"> 当传输和接收完成时, 写入下次要传输的数据。 	未使用	当传输和接收完成时, 写入下次要传输到该寄存器的数据开始下一次传输/接收。	未使用
SOTBFn (SOTBFnL)	写	-	-	保存最先要传输的数据 ^{注3}	-
		未使用	未使用	在开始传输/接收 (写入到 SOTBFn) 时, 写入最先要传输的数据。	未使用

注 1. 假设要传输的数据个数是 N。
 2. 读取和写入都不会启动通讯。

备注 在 16 位模式下, 使用括号外的寄存器; 在 8 位模式下, 使用括号内的寄存器。

15.4 操作

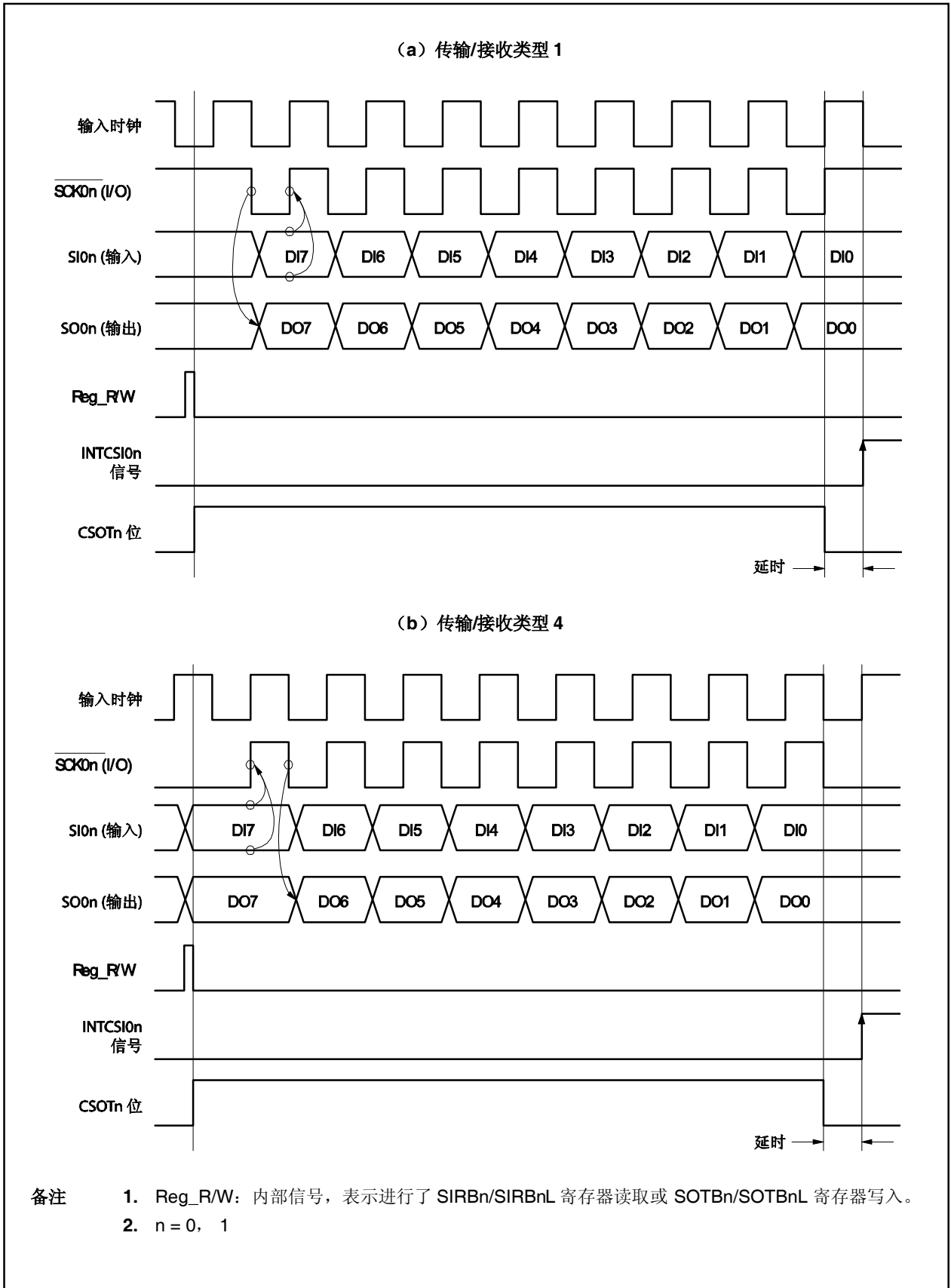
15.4.1 传输/接收完成中断请求信号 (INTCSI0n)

在数据传输/接收完成后 INTCSI0n 信号被设为 (1)。

写入到 CSIM0n 寄存器将 INTCSI0n 信号清零。

注意事项 只有在主模式 (CSICn.CKS0n2 到 CSICn.CKS0n0 位不是 111B) 下延迟模式 (CSIM0n.CSITn 位=1) 才有效。在从模式 (CKS0n2 到 CKS0n0 位是 111B) 下不要设置延迟模式。

图 15-2. 在延迟模式下 INTCSI0n 信号输出的时序图



15.4.2 单传输模式

(1) 用法

在仅接收模式下 (CSIM0n.TRMDn 位 = 0)，通过读取 SIRBn/SIRBnL 寄存器启动通讯。

在传输/接收模式下 (TRMDn 位 = 1)，通过写入到 SOTBn/SOTBnL 寄存器启动通讯。

在从模式下，必须事先允许操作 (CSIM0n.CSI0En 位 = 1)。

当通讯启动时，CSIM0n.CSOTn 位的值变为 1 (通讯执行状态)。

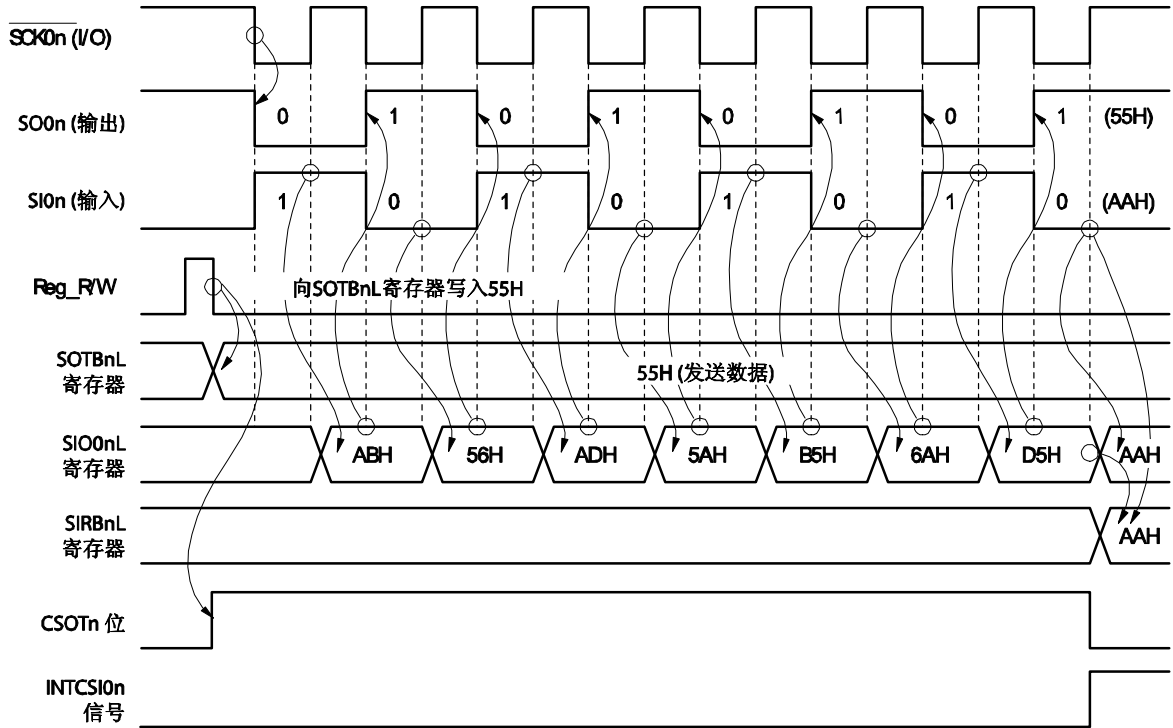
在通讯完成时，生成传输/接收完成中断请求信号 (INTCSI0n)，CSOTn 位被清零。然后等待下一个数据通讯请求。

注意事项 当 CSOTn 位 = 1 时，不要操作 CSI0n 寄存器。

备注 n = 0, 1

图 15-3. 单传输模式下的时序图 (1/2)

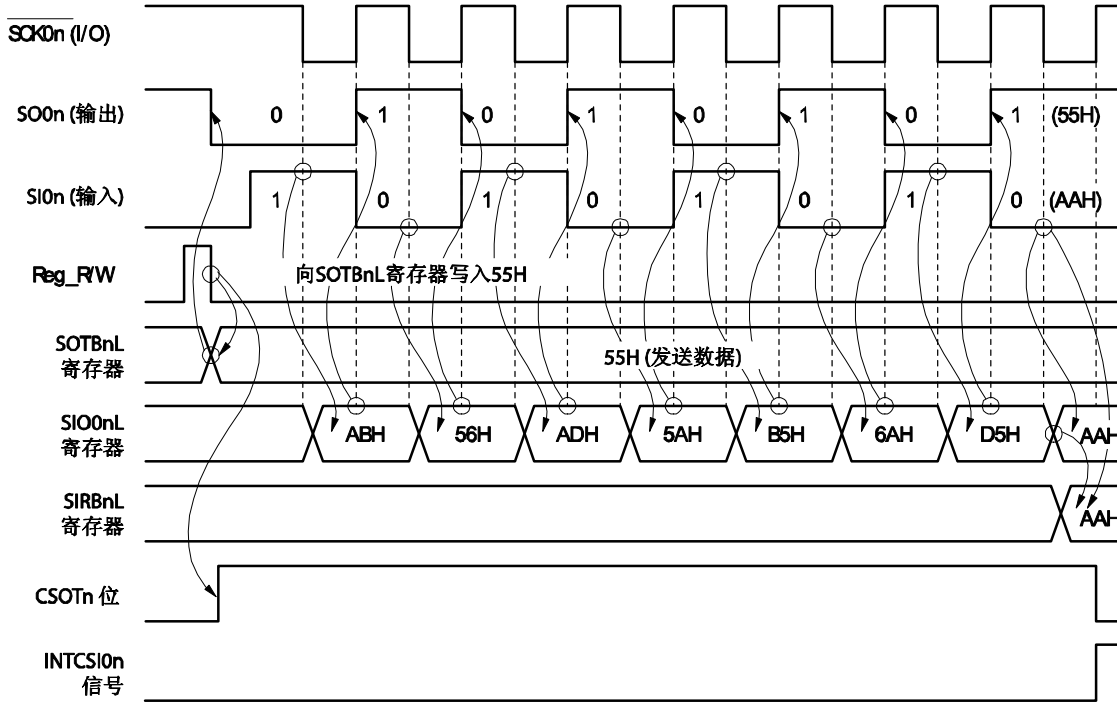
(a) 在传输/接收模式下，数据长度：8 位，传输方向：先是最高有效位，无中断延迟，单独传输模式，当接收到 AAH 和传输了 55H 后，传输/接收类型 1



- 备注
1. Reg_R/W: 内部信号，表示进行了 SIRBn/SIRBnL 寄存器读取或 SOTBn/SOTBnL 寄存器写入。
 2. 关于传输/接收类型，请参见 15.3 (2) 同步串行接口时钟选择寄存器 n (CSICn)。
 3. n = 0, 1

图 15-3. 单传输模式下的时序图 (2/2)

(b) 在传输/接收模式下，数据长度：8 位，传输方向：先是最高有效位，无中断延迟，单独传输模式，当接收到 AAH 和传输了 55H 后，传输/接收类型 2



- 备注
1. Reg_R/W: 内部信号，表示进行了 SIRBn/SIRBnL 寄存器读取或 SOTBn/SOTBnL 寄存器写入。
 2. 关于传输/接收类型，敬请参阅 15.3 (2) 同步串行接口时钟选择寄存器 n (CSICn)。
 3. n = 0, 1

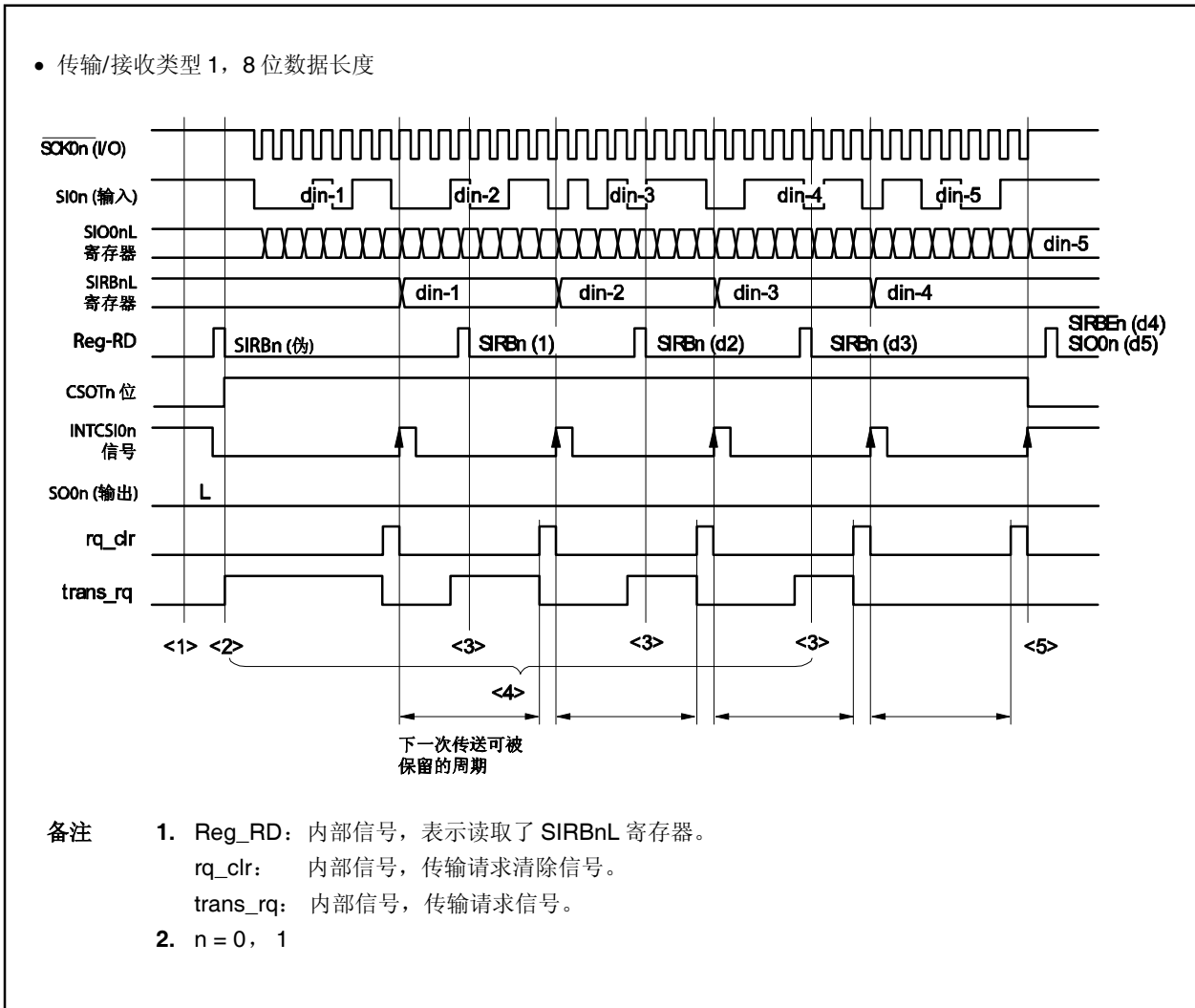
15.4.3 连续传输模式

(1) 用法 (仅接收: 8 位数据长度)

- <1> 设置连续传输模式 (CSIM0n.AUTOn 位 = 1) 和仅接收模式 (CSIM0n.TRMDn 位 = 0)。
- <2> 读取 SIRBnL 寄存器 (通过模拟读取开始传输)。
- <3> 当传输/接收完成中断请求信号 (INTCSI0n) 生成后, 读取 SIRBnL 寄存器^注 (预定下一次传输)。
- <4> 重复第<3>步 (N - 2) 次 (N: 传输数据的次数)。忽略第 (N - 1) 个数据的接收触发的中断 (此时可以读取 SIRBEnL 寄存器)。
- <5> 在生成最后一个 INTCSI0n 信号后, 读取 SIRBEnL 寄存器和 SIO0nL 寄存器^注。

注 当传输 N 个数据时, 接收数据是通过读取 SIRBnL 寄存器中的第一个数据到第 (N - 2) 个数据加载的; 第 (N - 1) 个数据是通过读取 SIRBEnL 寄存器加载的; 第 N 个 (最后一个) 数据是通过读取 SIO0nL 寄存器加载的。(参见表 15-1 每个缓冲寄存器的使用。)

图 15-4. 连续传输（仅接收）时序图



在连续传输模式的情况下, 在第一次传输的开始时设置两个传输请求。在 INTCSIO0n 信号后, 如果 SIRBnL 寄存器可以在下个传输预定期内被读取则传输继续。如果不能读取, 传输结束, SIRBnL 寄存器不接收 SIO0nL 寄存器的新值。

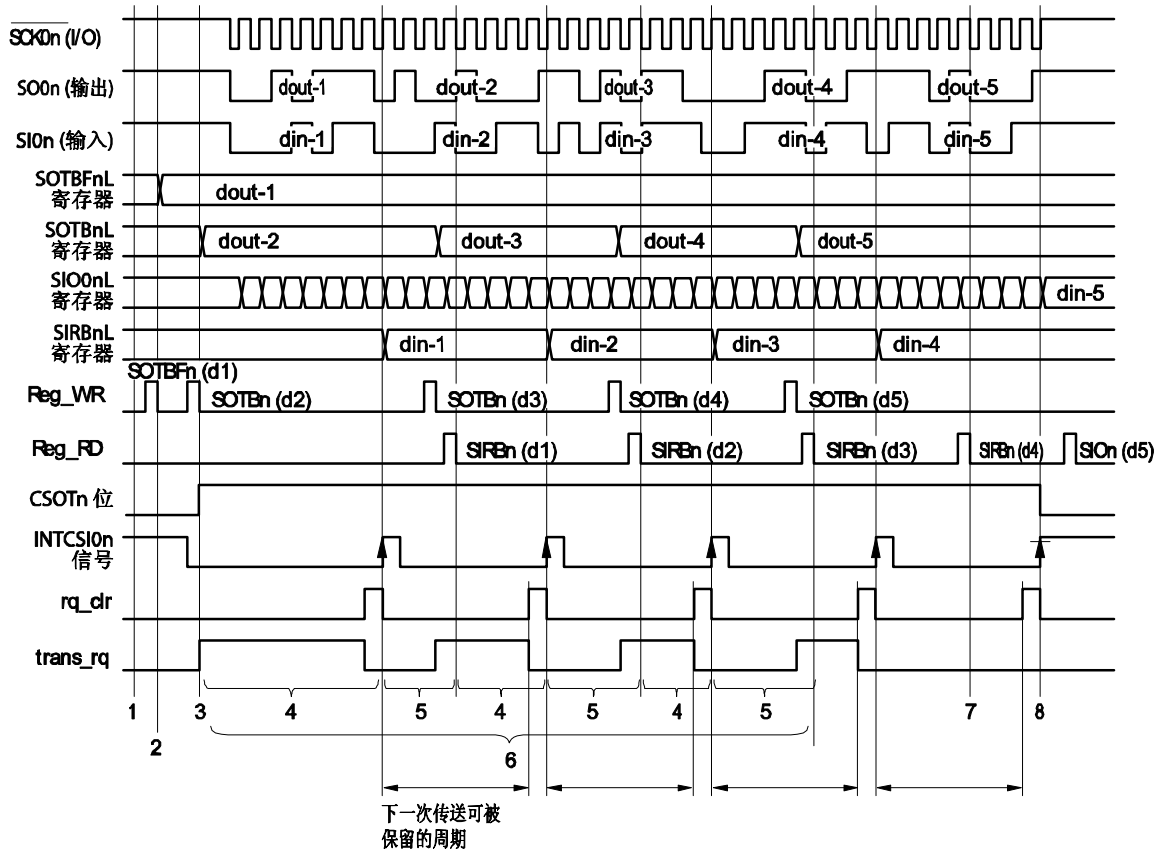
最后的数据可以通过在传输完成后读取 SIO0nL 寄存器获得。

(2) 用法 (传输/接收: 8 位数据长度)

- <1> 设置连续传输模式 (CSIM0n.AUTOn 位 = 1) 和传输/接收模式 (CSIM0n.TRMDn 位 = 1)。
- <2> 写入第 1 个数据到 SOTBFnL 寄存器。
- <3> 写入第 2 个数据到 SOTBnL 寄存器 (开始传输)。
- <4> 当传输/接收完成中断请求信号 (INTCSI0n) 生成后, 写入下一个数据到 SOTBnL 寄存器 (预定下一次传输)。读取 SIRBnL 寄存器以加载接收数据。
- <5> 只要还有要发送的数据, 不断重复第<4>步。
- <6> 当生成 INTCSI0n 信号时, 读取 SIRBnL 寄存器以加载 (N-1) 次的接收数据 (N: 传输数据的次数)。
- <7> 在最后一个 INTCSI0n 信号后, 读取 SIO0nL 寄存器以加载第 N 个 (最后一个) 接收数据。

图 15-5. 连续传输 (传输/接收) 时序图

- 传输/接收类型 1, 8 位数据长度



- 备注
1. Reg_WR: 内部信号, 表示写入了 SOTBnL 寄存器。
Reg_RD: 内部信号, 表示读取了 SIRBnL 寄存器。
rq_clr: 内部信号, 传输请求清除信号。
trans_rq: 内部信号, 传输请求信号。
 2. n = 0, 1

在连续传输模式的情况下, 在第一次传输的开始时设置两个传输请求。在 INTCSI0n 信号后, 如果可以在下个传输预定期内写入 SOTBnL 寄存器则传输继续。如果不能写入, 传输结束, SIRBnL 寄存器不接收 SIO0nL 寄存器的新值。

最后接收的数据可以通过在传输完成后读取 SIO0nL 寄存器获得。

(3) 下一个传输预定期

在连续传输模式下，下一次传输必须在图 15-6 中显示的时间期内准备好。

图 15-6. 下一个传输预定期的时序图 (1/2)

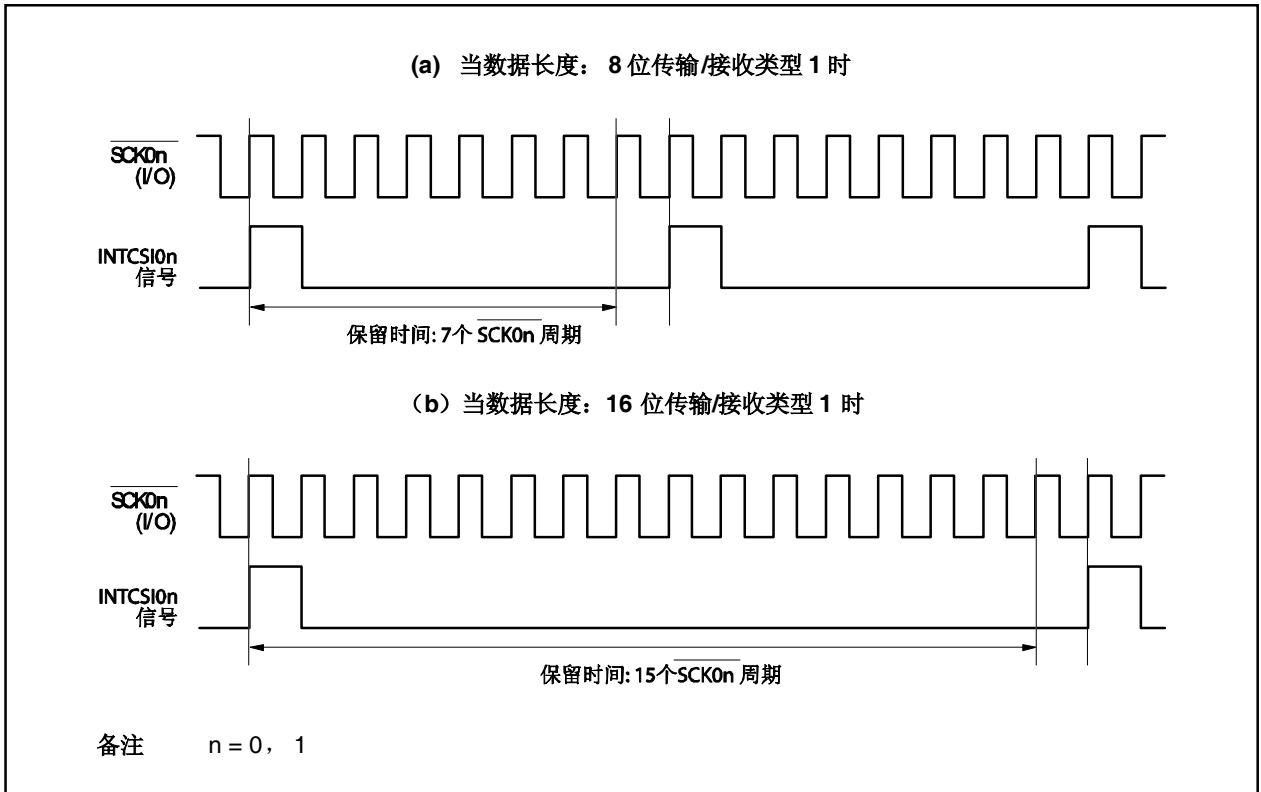
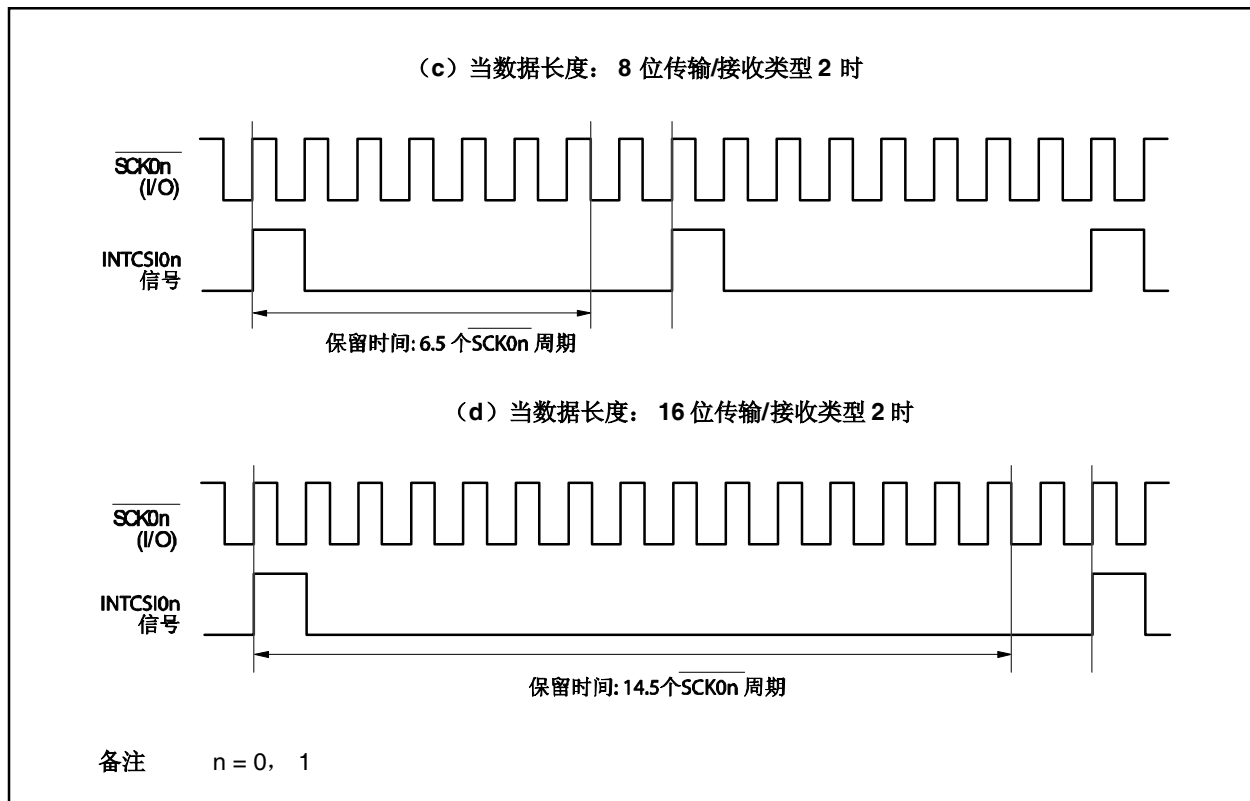


图 15-6. 下一个传输预定期的时序图 (2/2)



(4) 注意事项

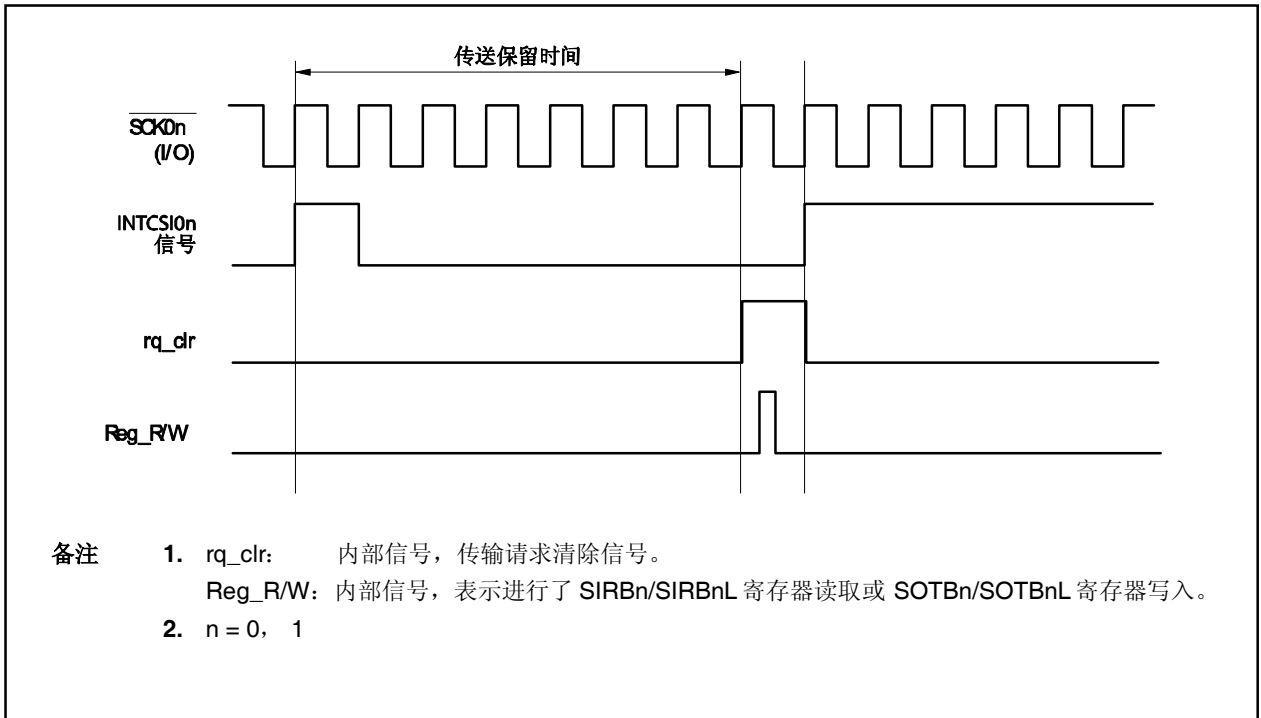
要继续连续传输，必须在传输预定期内读取 SIRBn 寄存器或写入到 SOTBn 寄存器。

如果在传输预定期结束后访问 SIRBn 寄存器或 SOTBn 寄存器，则出现以下情况：

(i) 在发生传输请求清除和寄存器访问之间的冲突的情况下

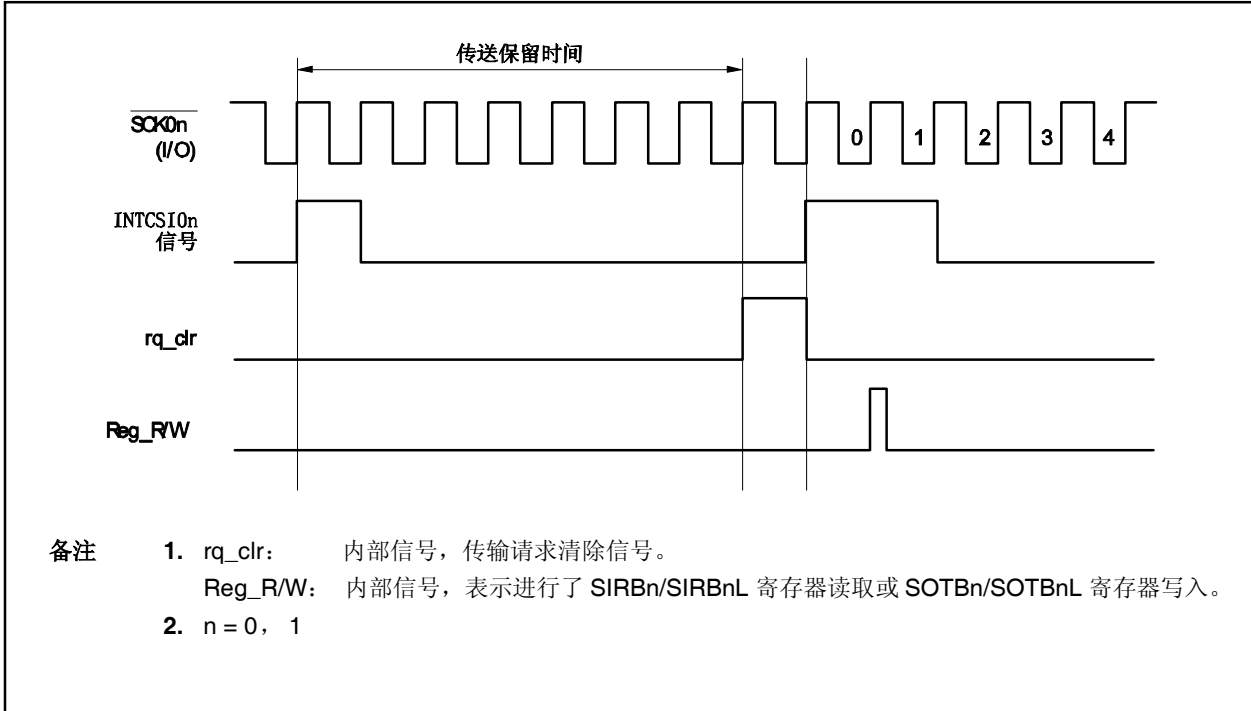
由于传输请求优先级较高，下一次传输请求被忽略。因此传输中断，不能进行正常的数据传输。

图 15-7. 传输请求清除和寄存器访问之间的冲突



- (ii) 在发送传输/接收完成中断请求信号 (INTCSIO_n) 和寄存器访问之间的冲突的情况下
 由于连续传输立即停止，作为新的连续传输执行。
 在从模式下，造成位相误差 (参见图 15-8)。
 在传输/接收模式下，重新传输 SOTBF_n 寄存器的值，发送非法的数据。

图 15-8. 中断请求和寄存器访问之间的冲突



15.5 输出引脚

下面讲述输出引脚。关于每个引脚的设置，敬请参阅表 4-12 当端口引脚用作复用功能时的设置。

(1) $\overline{\text{SCK0n}}$ 引脚

当禁用 CSI0n 操作 (CSI0En 位 = 0) 时， $\overline{\text{SCK0n}}$ 引脚的输出状态如下：

表 15-2. $\overline{\text{SCK0n}}$ 引脚输出状态

CKPn	CKS0n2	CKS0n1	CKS0n0	$\overline{\text{SCK0n}}$ 引脚输出
0	忽略	忽略	忽略	固定为高电平
1	1	1	1	高阻抗
	其他			固定为低电平

备注 n = 0, 1

(2) SO0n 引脚

当禁用 CSI0n 操作 (CSI0En 位 = 0) 时，SO0n 引脚的输出状态如下：

表 15-3. SO0n 引脚输出状态

TRMDn	DAPn	AUTO0n	CCLn	DIRn	SO0n 引脚输出
0	忽略	忽略	忽略	忽略	固定为低电平
1	0	忽略	忽略	忽略	SO 锁值 (低电平)
				1	0
	1	SOTBn0 位值			
	1	0	0		SOTBn15 位值
			1		SOTBn0 位值
	1	0	0	0	SOTBFn7 位值
				1	SOTBFn0 位值
		1	0	1	0
1					SOTBFn0 位值

备注 n = 0, 1

第十六章 IIC 总线

如果要使用内部 IIC 总线功能，将 P38/SDA0 和 P39/SCL0 引脚分别用作串行发送/接收数据输入输出引脚（SDA0）和串行时钟输入输出引脚（SCL0），并将它们设为 N 通道漏极开路输出。

在 V850ES/KE2 中，提供 IIC 总线的一个通道。

16.1 特点功能

IIC0 有以下两种模式。

- 操作停止模式
- IIC 总线模式（支持多主）

(1) 操作停止模式

此模式和在没有进行串行发送时使用，因此可用于降低功率消耗。

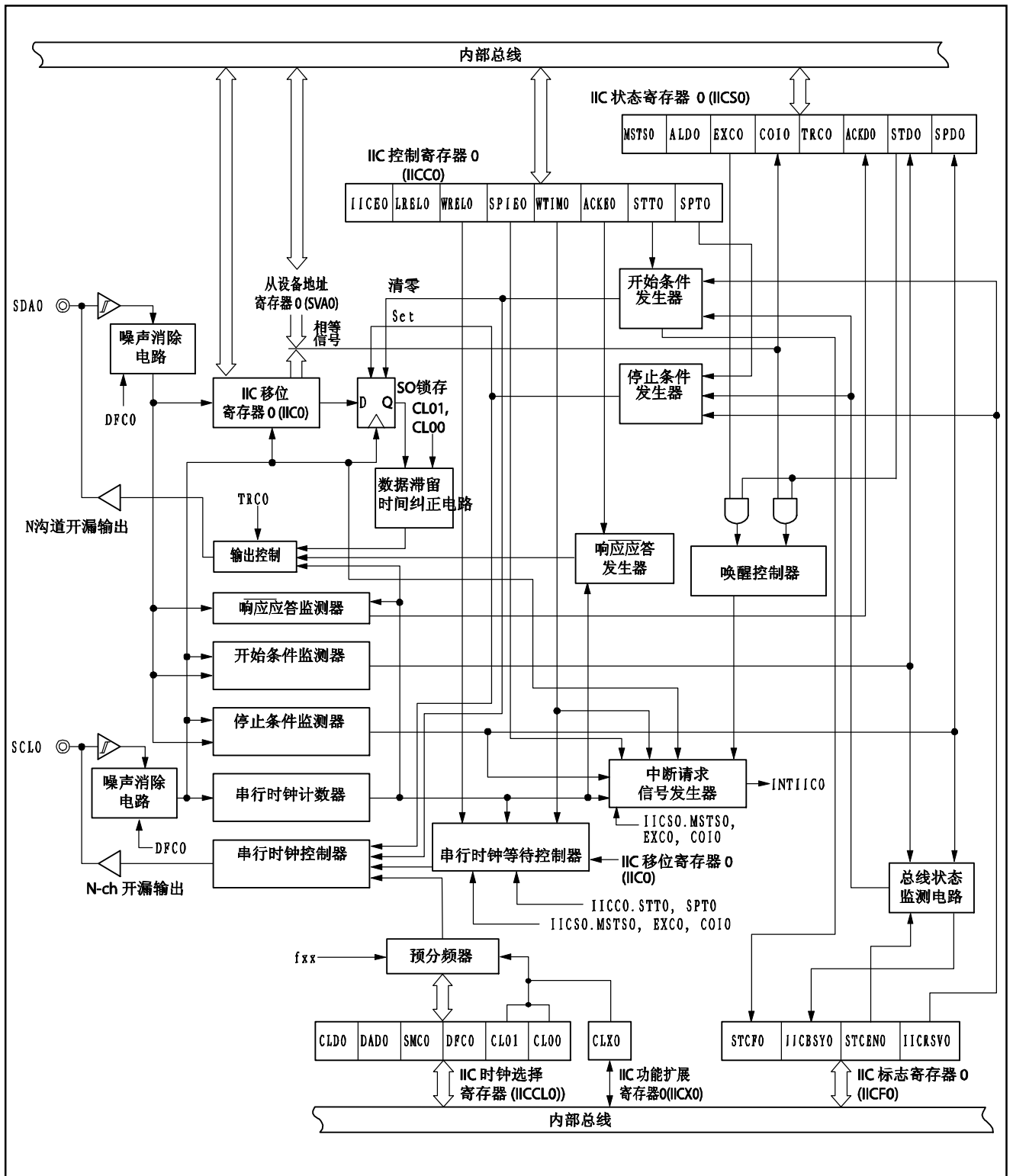
(2) IIC 总线模式（支持多主）

此模式用于通过两条线路（串行时钟（SCL0）线和串行数据总线（SDA0））与多个设备进行 8 位数据发送。

此模式符合 IIC 总线格式，主设备能够生成“开始条件”、“地址”、“发送方向指定”、“数据”和“停止条件”数据通过串行数据总线发送到从设备。从设备自动通过检测接收的状态。本功能可以简化控制 IIC 总线的应用程序部分。

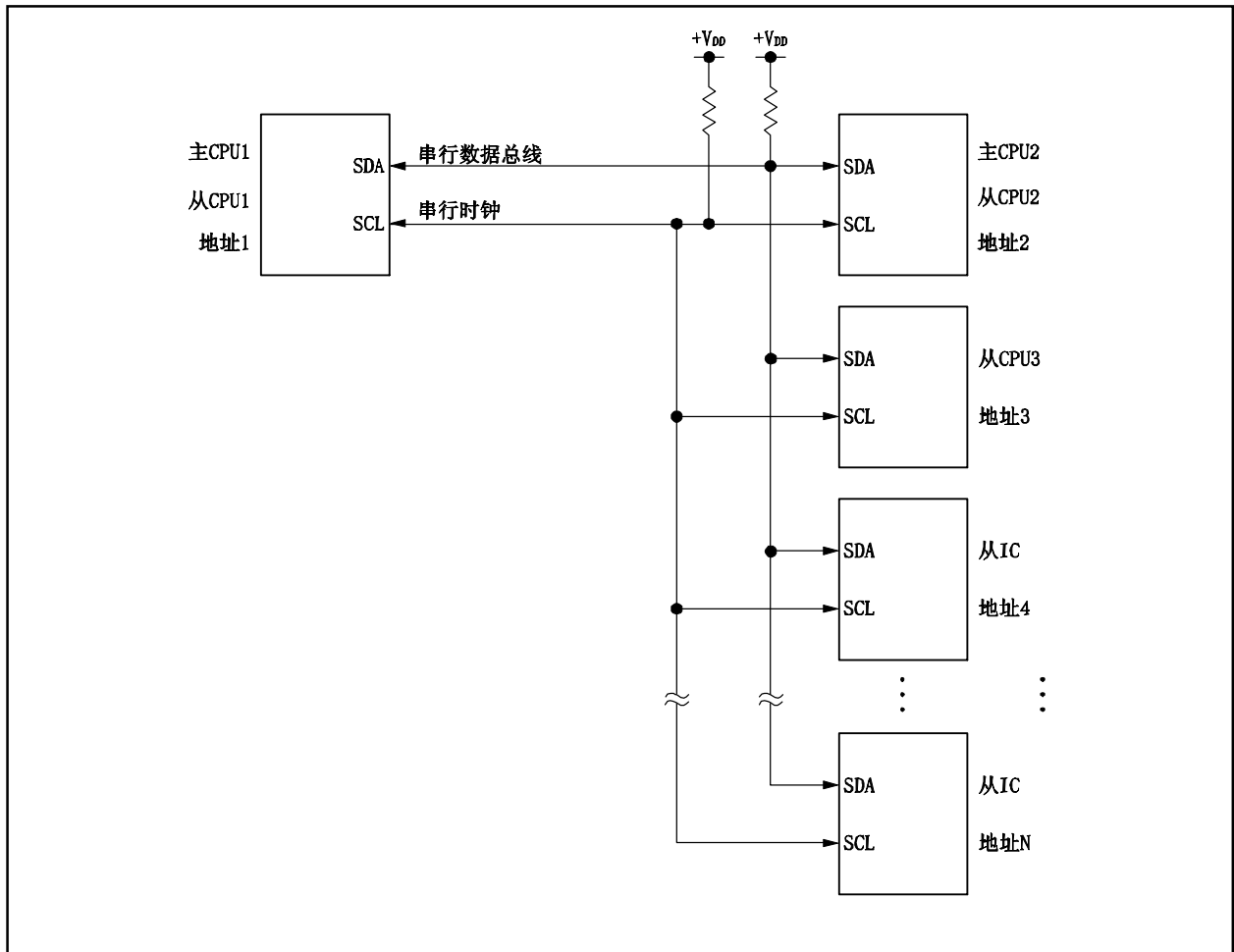
由于将 SCL0 和 SDA0 引脚用于 N-通道漏极开路输出，内部 I²C0 要求上拉电阻用于串行时钟线和串行数据总线。

图 16-1. I²C0 框图



串行总线的配置实例如下。

图 16-2. 使用 I²C 总线的串行总线的配置实例



16.2 配置

I²C0 包括以下硬件。

表 16-1. I²C0 的配置

项目	配置
寄存器	IIC 移位寄存器 0 (IIC0) 从地址寄存器 0 (SVA0)
控制寄存器	IIC 控制寄存器 0 (IICC0) IIC 状态寄存器 0 (IICS0) IIC 标志寄存器 0 (IICF0) IIC 时钟选择寄存器 0 (IICCL0) IIC 功能扩展寄存器 0 (IICX0)

(1) IIC 移位寄存器 0 (IIC0)

IIC0 寄存器用于将 8 位串行数据转化为 8 位并行数据以及将 8 位并行数据转化为 8 位串行数据。IIC0 寄存器可以用于数据发送和接收。

对 IIC0 寄存器的读和写操作用于控制实际的发送和接收操作。

IIC0 寄存器支持 8 位读写方式。

IIC0 寄存器复位后的值为 00H。

(2) 从地址寄存器 0 (SVA0)

SVA0 寄存器在从模式下设置本地地址。

SVA0 寄存器支持 8 位读写方式。

SVA0 寄存器复位后的值为 00H。

(3) SO 锁存器

SO 锁存器用于保持 SDA0 引脚的输出电平。

(4) 唤醒控制器

当此寄存器收到的地址相等 SVA0 寄存器的地址值时或当收到一个扩展代码时，此电路产生一个中断请求信号 (INTIIC0)。

(5) 预分频器

此项选择要使用的采样时钟。

(6) 串行时钟计数器

此计数器计数发送/接收操作中输入和输出的串行时钟以及用于检验发送或接收的 8 位数据。

(7) 中断请求信号发生器

此电路控制中断请求信号 (INTIIC0) 的生成。

在以下触发情况之一发生时，产生一个 IIC 中断。

- 串行时钟的第八或第九个时钟下降 (IICC0.WTIM0 位设置)
- 当检测到停止条件时产生中断请求 (IICC0.SPIE0 位设置)

- (8) **串行时钟控制器**
在主模式中，此电路通过 SCL0 引脚从采样时钟生成时钟输出。
- (9) **串行时钟等待控制器**
此项控制等待时间。
- (10) **$\overline{\text{ACK}}$ 发生器、停止条件检测器、开始条件检测器和 $\overline{\text{ACK}}$ 检测器**
此电路用于生成和探测各种状态。
- (11) **数据保持时间校正电路**
此电路产生与串行时钟下降沿对应的数据保留时间。
- (12) **开始条件生成器**
此电路在 IICC0.STT0 设时序产生一个开始条件。
但在通讯预定禁用状态下 (IICF0.IICRSV0 位= 1)，当总线没有释放时 (IICF0.IICBSY0 位= 1)，开始条件请求被忽略，IICF0.STCF0 位被设为 1。
- (13) **停止条件生成器**
当 IIC0.SPT0 位设为 (1) 时生成停止条件。
- (14) **总线状态检测器**
此电路检测总线是否通过检测开始条件和停止条件释放。
但由于在操作后不能立即检测总线状态，最初的状态由 IICF0.STCEN0 位设置。

16.3 寄存器

I²C0 由以下寄存器控制。

- IIC 控制寄存器 0 (IICC0)
- IIC 状态寄存器 0 (IICS0)
- IIC 标志寄存器 0 (IICF0)
- IIC 时钟选择寄存器 0 (IICCL0)
- IIC 功能扩展寄存器 0 (IICX0)

同时也使用以下寄存器。

- IIC 移位寄存器 0 (IIC0)
- 从地址寄存器 0 (SVA0)

备注 关于复用功能的引脚设置，敬请参阅表 4-12 当端口引脚用于复用功能时的设置。

(1) IIC 控制寄存器 0 (IICC0)

IICC0 寄存器用于启用/停止 I²C0 的操作，设置等待时间以及设置其他的 I²C 操作。

IICC0 寄存器支持 8 位或 1 位读写方式。但在 IICE0 位是 0 或等待期间设置 SPIE0、WTIM0 和 ACKE0 位。当将 IICE0 位从“0”设置为“1”时，可同时设置这些位。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: IICC0 FFFFFFFD82H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICC0	IICE0	LRELO	WRELO	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I ² C0 运行启用/禁用指定
0	停止运行。复位 IICS0 寄存器 ^{注1} 。停止内部运行。
1	启用运行。

确保当 SCL0 和 SDA0 先在高电平时设置位为 1。

清除的条件 (IICE0 位= 0)	设置的条件 (IICE0 位= 1)
<ul style="list-style-type: none"> • 通过指令清除 • 复位 	<ul style="list-style-type: none"> • 通过指令设置

LRELO ^{注2}	从通信中退出
0	正常运行
1	此从当前的通讯退出并设置待机模式。在执行后本设置自动清零。其运用包括在收到本地无关的扩展代码的情况。SCL0 和 SDA0 线设为高阻抗。STT0、SPT0、IICS0.MSTS0、IICS0.EXC0、IICS0.COI0、IICS0.TRC0、IICS0.ACKD0 和 IICS0.STD0 位被清零。

退出通信后的待机模式将保持有效直到满足以下通信进入条件。

- 在检测到停止条件后，在主模式中重新启动。
- 在开始条件后，出现地址相等或扩展码接收。

清除的条件 (LRELO 位= 0)	设置的条件 (LRELO 位= 1)
<ul style="list-style-type: none"> • 在执行后自动清除 • 复位 	<ul style="list-style-type: none"> • 通过指令设置

WRELO ^{注2}	等待取消控制
0	不要取消等待
1	取消等待。在等待取消后本设置自动清零。

清除的条件 (WRELO 位= 0)	设置的条件 (WRELO 位= 1)
<ul style="list-style-type: none"> • 在执行后自动清除 • 复位 	<ul style="list-style-type: none"> • 通过指令设置

- 注
1. IICS0 寄存器和 IICF0.STCF0、IICF0.IICBSY0、IICCL0.CLD0 和 IICCL0.DAD0 位复位。
 2. 当 IICE0 位= 0 时，此标记的信号无效。

注意事项 如果在 SCL0 线处于高电平且 SDA0 线处于低电平的情况下启用 I²C0 的运行 (IICE0 位=1)，则立即检测开始条件。为了避免此情况，在启用 I²C0 的运行后，立即将位操作指令设置为 LRELO 位=1。

SPIE0 ^注		启用/禁用检测到停止条件时生成中断请求	
0	禁用		
1	启用		
清除的条件 (SPIE0 位= 0)		设置的条件 (SPIE0 位= 1)	
<ul style="list-style-type: none"> • 通过指令清除 • 复位 		<ul style="list-style-type: none"> • 通过指令设置 	
WTIMO ^注		等待和中断请求生成的控制	
0	在第八个时钟的下降沿生成中断请求。 主模式：在第八个时钟输出后，时钟输出设置为低电平并设置等待。 从模式：在第八个时钟输出后，时钟输出设置为低电平设置主设备为等待。		
1	在第九个时钟的下降沿生成中断请求。 主模式：在第九个时钟输出后，时钟输出设置为低电平并设置等待。 从模式：在第九个时钟输出后，时钟输出设置为低电平并设置主设备为等待。		
在独立于此位设置的地址传送过程中，在第九个时钟的下降处生成中断。在地址传送完成后此位的设置有效。在主模式下，在地址传送的过程中在第九个时钟的下降处插入一个等待。对于已经收到本地地址的从设备，在发出 ACK 后在第九个时钟的下降处插入一个等待。但当从设备已经收到一个扩展代码时，在第八个时钟的下降处插入一个等待。			
清除的条件 (WTIMO 位= 0)		设置的条件 (WTIMO 位= 1)	
<ul style="list-style-type: none"> • 通过指令清除 • 复位 		<ul style="list-style-type: none"> • 通过指令设置 	
ACKE0 ^注		确认控制	
0	禁用确认。		
1	启用确认。在第九个时钟周期中，设置 SDA0 线低电平。		
ACKE0 位的设置对地址接收无效，但对扩展代码的地址接收有效。			
清除的条件 (ACKE0 位= 0)		设置的条件 (ACKE0 位= 1)	
<ul style="list-style-type: none"> • 通过指令清除 • 复位 		<ul style="list-style-type: none"> • 通过指令设置 	
<p>注 当 IICE0 位= 0 时，此标记信号无效。</p>			

STT0	开始条件触发器	
0	不生成开始条件。	
1	<p>当总线释放时（停止模式）： 生成一个开始条件（作为主设备的开始）。当 SCL0 线为高电平时，SDA0 线从高电平变为低电平，然后生成开始条件。在经过规定的时间后，当 SCL0 线变为低电平（等待状态）。</p> <p>当第三方在通讯时</p> <ul style="list-style-type: none"> • 当启用通讯预定功能时（IICF0.IICRSV0 位 = 0） 与开始条件预定标记作用相同。当设置为 1 时，在总线释放后自动生成一个开始条件。 • 当禁用通讯预定功能时（IICF0.IICRSV0 位 = 1） IICF0.STCF0 位设为 1，清除 STT0 位设置的信息。不生成开始条件。 <p>在等待状态下（主设备时）： 在发出等待后生成一个重新启动条件。</p>	
<p>关于设置时序的注意事项</p> <p>对于主设备接收：不能在传送中设置为 1。只有在 ACKE0 位被清零且通知从设备最后的接收时才能设置为 1。</p> <p>对于主设备传送：在 ACK 期间无法正常生成一个开始条件。在第九个时钟的输出后的等待期间设置为 1。</p> <ul style="list-style-type: none"> • 无法和 SPT0 位同时设为 1。 • 当 STT0 位当设为 1 时，禁止将 STT0 位设置为 1 直到设置清零。 		
清除的条件（STT0 位 = 0）		设置的条件（STT0 位 = 1）
<ul style="list-style-type: none"> • 当在通讯预定禁用状态下 STT0 位设置为 1 时 • 通过仲裁中失败清除 • 当由主设备设备产生开始条件时清除 • 当 LRELO 位 = 1 时（从通讯中退出） • 当 IICE0 位从 1 变为 0 时（运行停止） • 复位 		<ul style="list-style-type: none"> • 通过指令设置
<p>备注 如果在数据设置后读取，STT0 位为 0。</p>		

SPT0	停止条件触发器	
0	不生成停止条件。	
1	生成一个停止条件（作为主设备发送的终止）。 当 SDA0 线为低电平时，将 SCL0 线设为高电平或者等待直到 SCL0 引脚变为高电平。在经过规定的 时间后，当 SDA0 线从低电平变为高电平并生成一个停止条件。	
<p>关于设置时序的注意事项</p> <p>对于主设备接收：不能在传送中设置为 1。只有在 ACKE0 位被清零且只有通知从设备最后的接收后的等待 期间才能设置为 1。</p> <p>对于主设备传送：在 ACK 期间无法正常生成一个停止条件。在第九个时钟的输出后的等待期间设置为 1。</p> <ul style="list-style-type: none"> • 无法和 STT0 位同时设为 1。 • 只有在主模式^注中才能将 SPT0 位设置为 1。 • 当 WTIMO 位被清零时，如果在第八个时钟输出后的等待期间 SPT0 位被设为 1，注意在第九个时钟的高电平期 间会产生一个停止条件。 <p>在第八个时钟后的等待期间，WTIMO 位应从 0 变为 1；在第九个时钟后的等待期间，SPT0 位应设置为 1。</p> <ul style="list-style-type: none"> • 当 SPT0 位当设为 1 时，禁止将 SPT0 位设置为 1 直到设置清零。 		
清除的条件（SPT0 位= 0）		设置的条件（SPT0 位= 1）
<ul style="list-style-type: none"> • 通过仲裁中失败清除 • 当检测到停止条件后自动清除 • 当 LRELO 位= 1 时（从通讯中退出） • 当 IICE0 位从 1 变为 0 时（运行停止） • 复位 		<ul style="list-style-type: none"> • 通过指令设置
<p>注 仅在主模式中设置 SPT0 位为 1。但在切换到运行启用状态后检测到第一个停止条件前必须 将设置 SPT0 位置为 1 并生成一个停止条件。关于更多细节，敬请参阅 16.14 注意事项。</p> <p>注意事项 当 IICS0.TRC0 设置为 1 时，第九个时钟期间的 WRELO 位设置 1，等待取消；之后 TRC0 位被清零，SDA0 线被设为高阻抗。</p> <p>备注 如果在数据设置后读取，SPT0 位为 0。</p>		

(2) IIC 状态寄存器 0 (IICS0)

IICS0 寄存器显示 IIC0 总线的状态。

IICS0 寄存器支持 8 位或 1 位只读方式。

只有当 IICC0.STT0 位为 1 时或者在等待期间，IICS0 寄存器只能被读取。

该寄存器复位后的值为 00H。

注意事项 主时钟停止，CPU 运行在副时钟上时，不要访问 IICS0 寄存器。关于更多细节，
敬请参阅 3.4.8 (2)。

(1/3)

复位后: 00H	R	地址: IICS0 FFFFFFFD86H															
IICS0	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;"><7></td> <td style="text-align: center;"><6></td> <td style="text-align: center;"><5></td> <td style="text-align: center;"><4></td> <td style="text-align: center;"><3></td> <td style="text-align: center;"><2></td> <td style="text-align: center;"><1></td> <td style="text-align: center;"><0></td> </tr> <tr> <td style="text-align: center;">MSTS0</td> <td style="text-align: center;">ALD0</td> <td style="text-align: center;">EXC0</td> <td style="text-align: center;">COI0</td> <td style="text-align: center;">TRC0</td> <td style="text-align: center;">ACKD0</td> <td style="text-align: center;">STD0</td> <td style="text-align: center;">SPD0</td> </tr> </table>	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0
<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>										
MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0										
MSTS0	主设备状态																
0	从设备状态或通讯等待状态																
1	主设备通讯状态																
清除的条件 (MSTS0 位= 0)				设置的条件 (MSTS0 位= 1)													
<ul style="list-style-type: none"> • 当检测到停止条件时 • 当 ALD0 位=1 时 (仲裁失败) • 通过 IICC0.LRELO 位= 1 清除 (从通讯中退出) • 当 IICC0.IICE0 位从 1 变为 0 时 (运行停止) • 复位 				<ul style="list-style-type: none"> • 当生成一个开始条件时 													
ALD0	仲裁失败的探测																
0	此状态指的是没有仲裁或仲裁结果是“胜”。																
1	此状态表示仲裁结果是“败”。MSTS0 位被清零。																
清除的条件 (ALD0 位= 0)				设置的条件 (ALD0 位= 1)													
<ul style="list-style-type: none"> • 在 IICS0 寄存器读取后自动清除^注 • 当 IICE0 位从 1 变为 0 时 (运行停止) • 复位 				<ul style="list-style-type: none"> • 当仲裁结果是“败”。 													

注 当为 IICS0 寄存器中的另一个位执行位操作指示时 ALD0 位也会被清除。

EXC0		扩展代码接收的检测	
0	没有收到扩展代码。		
1	收到扩展代码。		
清除的条件 (EXC0 位= 0)		设置的条件 (EXC0 位= 1)	
<ul style="list-style-type: none"> • 当检测到开始条件时 • 当检测到停止条件时 • 通过 LRELO 位= 1 清除 (从通讯中退出) • 当 IICE0 位从 1 变为 0 时 (运行停止) • 复位 		<ul style="list-style-type: none"> • 当收到的地址数据的上四位时“0000”或“1111”时 (设在第八个时钟的上升沿)。 	
COI0		相等地址的检测	
0	地址不相等。		
1	地址相等。		
清除的条件 (COI0 位= 0)		设置的条件 (COI0 位= 1)	
<ul style="list-style-type: none"> • 当检测到开始条件时 • 当检测到停止条件时 • 通过 LRELO 位= 1 清除 (从通讯中退出) • 当 IICE0 位从 1 变为 0 时 • 复位 		<ul style="list-style-type: none"> • 当收到的地址相等本地地址时 (SVA0 寄存器) (设在第八个时钟的上升沿)。 	
TRC0		传送/接收状态的检测	
0	接收状态 (非传送状态)。设置 SDA0 线为高阻抗。		
1	传送状态。启用 SO 锁中的值用于到 SDA0 线的输出 (在第一个字节的第九个时钟的上升沿有效开始)。		
清除的条件 (TRC0 位= 0)		设置的条件 (TRC0 位= 1)	
<ul style="list-style-type: none"> • 当检测到停止条件时 • 通过 LRELO 位= 1 清除 (从通讯中退出) • 当 IICE0 位从 1 变为 0 时 (运行停止) • 通过 IICC0.WRELO 位= 1 清除^注 (等待释放) • 当 ALD0 位从 0 变为 1 时 (仲裁失败) • 复位 主设备 <ul style="list-style-type: none"> • 当输出“1”到第一个字节的最低有效位时 (发送方向指定位) 从设备 <ul style="list-style-type: none"> • 当检测到开始条件时 当未用于通信时		主设备 <ul style="list-style-type: none"> • 当生成一个开始条件时 • 当输出“0”到第一个字节的最低有效位时 (发送方向指定位) 从设备 <ul style="list-style-type: none"> • 当输出“1”到第一个字节的最低有效位时 (发送方向指定位) 	
<p>注 当 IICC0.WRELO 位设为 1, 在第九个时钟释放等待状态且 TRC0 位= 1 时, IICS0.TRCS0 位被清零, SDA0 变为高阻抗。</p>			

ACKD0	ACK 的检测	
0	没有检测到 $\overline{\text{ACK}}$ 。	
1	检测到 $\overline{\text{ACK}}$ 。	
清除的条件 (ACKD0 位= 0)		设置的条件 (ACKD0 位= 1)
<ul style="list-style-type: none"> 当检测到停止条件时 在第一个时钟下一字节的上升沿 通过 LREL0 位= 1 清除 (从通讯中退出) 当 IICE0 位从 1 变为 0 时 (运行停止) 复位 		<ul style="list-style-type: none"> 在 SCL0 引脚第九个时钟的上升沿将 SDA0 引脚设为低电平后
STD0	开始条件的检测	
0	没有检测到开始条件。	
1	检测到开始条件。此显示地址传送期有效。	
清除的条件 (STD0 位= 0)		设置的条件 (STD0 位= 1)
<ul style="list-style-type: none"> 当检测到停止条件时 在地址传送后下一个字节的第一个时钟的上升沿 通过 LREL0 位= 1 清除 (从通讯中退出) 当 IICE0 位从 1 变为 0 时 (运行停止) 复位 		<ul style="list-style-type: none"> 当检测到开始条件时
SPD0	停止条件的检测	
0	没有检测到开始条件。	
1	检测到停止条件。主设备通讯终止，总线释放。	
清除的条件 (SPD0 位= 0)		设置的条件 (SPD0 位= 1)
<ul style="list-style-type: none"> 在设置此位和检测到开始条件后地址传送字节的第一个时钟的上升沿 当 IICE0 位从 1 变为 0 时 (运行停止) 复位 		<ul style="list-style-type: none"> 当检测到停止条件时

(3) IIC 标志寄存器 0 (IICF0)

IICF0 是一个设置 I²C0 允许模式以及显示 IIC 总线状态的寄存器。

该寄存器支持 8 位或 1 位读写方式。但 STCF0 和 IICBSY0 位是只读的。

IICRSV0 位可以用于启用/禁用通讯预定功能（参见 **16.13 通讯预定**）。

STCEN0 位可用于设置 IICBSY0 位的初始值（参见 **16.14 注意事项**）。

只有当 I²C0 的运行被禁止时（IICC0.IICE0 位= 0），IICRSV0 和 STCEN0 才是可写的。当运行被禁止时，IICF0 寄存器可读。

该寄存器复位后的值为 00H。

复位后: 00H	R/W*	地址: IICF0 FFFFFFFD8AH						
	<7>	<6>	5	4	3	2	<1>	<0>
IICS0	STCF0	IICBSY0	0	0	0	0	STCEN0	IICRSV0

STCF0	IIC0.STT0 清除标记
0	生成开始条件
1	开始条件生成失败: 清除 STT0 标记
清除的条件 (STCF0 位= 0)	
<ul style="list-style-type: none"> 通过设置 STT0 位=1 清除 当 IICE0 位从 1 变为 0 时 (运行停止) 复位 	
设置的条件 (STCF0 位= 1)	
<ul style="list-style-type: none"> 当通讯预定禁用时 (IICRSV0 位=1), 生成开始条件不成功, STT0 位清零。 	

IICBSY0	IIC 总线状态标记
0	总线释放状态 (当 STCEN0 位= 1 时的初始通讯状态)。
1	总线通讯状态 (当 STCEN0 位= 0 时的初始通讯状态)。
清除的条件 (IICBSY0 位= 0)	
<ul style="list-style-type: none"> 停止条件的检测 当 IICE0 位从 1 变为 0 时 (运行停止) 复位 	
设置的条件 (IICBSY0 位= 1)	
<ul style="list-style-type: none"> 开始条件的检测 当 STCEN0 位= 0 时 IICE0 的设置 	

STCEN0	初始开始启用触发器
0	在运行启用后 (IICE0 位= 1), 一旦检测到停止条件后启用开始条件的生成。
1	在运行启用后 (IICE0 位= 1), 没有检测到停止条件就启用开始条件的生成。
清除的条件 (STCEN0 位= 0)	
<ul style="list-style-type: none"> 停止条件的检测 复位 	
设置的条件 (STCEN0 位=1)	
<ul style="list-style-type: none"> 通过指令设置 	

IICRSV0	通讯预定功能禁用位
0	启用通讯预定
1	禁用通讯预定
清除的条件 (IICRSV0 位= 0)	
<ul style="list-style-type: none"> 通过指令清除 复位 	
设置的条件 (IICRSV0 位= 1)	
<ul style="list-style-type: none"> 通过指令设置 	

注 第 6 位和 7 位是只读位。

- 注意事项
1. 只有在运行停止时 (IICE0 位= 0) 写入到 STCEN0 位。
 2. 由于当 STCEN0 位 = 1 时不论实际总线的状态如何而识别位总线释放状态 (IICBSY0 位=0), 当生成第一个开始条件 (STT0 位= 1) 时, 必须核实没有第三方的通讯在进行中, 从而避免通讯受到破坏。
 3. 只有在运行停止时 (IICE0 位= 0) 写入到 IICRSV0 位。

(4) IIC 时钟选择寄存器 0 (IICCL0)

IICCL0 寄存器用于设置 IIC0 总线的传送时钟。

IICCL0 寄存器支持 8 位或 1 位读写方式。但 CLD0 和 DAD0 位是只读的。SMC0、CL01 和 CL00 位与 IICX0.CLX0 位组合设置。（参见 16.3 (6) IIC0 传送时钟设置方法）。

当 IICC0.IICE0 位= 0 时设置 IICCL0 寄存器。

该寄存器复位后的值为 00H。

复位后: 00H R/W ^注 地址: IICCL0 FFFFD84H

	7	6	<5>	<4>	3	2	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	CL01	CL00

CLD0	SCL0 引脚电平的检测（只有在 IICC0.IICE0 位= 1 时有效）	
0	检测到 SCL0 引脚处于低电平。	
1	检测到 SCL0 引脚处于高电平。	
清除的条件（CLD0 位= 0）		设置的条件（CLD0 位= 1）
<ul style="list-style-type: none"> 当 SCL0 引脚处于低电平时 当 IICE0 位从 1 变为 0 时（运行停止） 复位 		<ul style="list-style-type: none"> 当 SCL0 引脚处于高电平时

DAD0	SDA0 引脚电平的检测（只有在 IICE0 位= 1 时有效）	
0	检测到 SDA0 引脚处于低电平。	
1	检测到 SDA0 引脚处于高电平。	
清除的条件（DAD0 位= 0）		设置的条件（DAD0 位= 1）
<ul style="list-style-type: none"> 当 SDA0 引脚处于低电平时。 当 IICE0 位从 1 变为 0 时（运行停止） 复位 		<ul style="list-style-type: none"> 当 SDA0 引脚处于高电平时

SMC0	运行模式切换	
0	运行在标准模式。	
1	运行在高速模式。	

DFC0	数字滤波器运行控制	
0	数字滤波器关闭。	
1	数字滤波器开启。	
数字滤波器只能在高速模式下使用。 在高速模式下，不论 DFC0 位设置/清除，传送时钟不变。 数字滤波器用于在高速模式下消除噪声。		

注 第 4 位和 5 位是只读位。

(5) IIC 功能扩展寄存器 0 (IICX0)

这些寄存器设置 I²C0 的功能扩展（只有在高速模式下有效）。

IICX0 寄存器支持 8 位或 1 位读写方式。CLX0 位与 IICCL0.SMC0、IICCL0.CL01 和 IICCL0.CL00 位组合设置。（参见 6.3 (6) I²C0 传送时钟设置方法）。

当 IICC0.IICE0 位=0 时设置 IICX0 寄存器。

该寄存器复位后的值为 00H。

复位后: 00H	R/W	地址: IICX0 FFFFFFFD85H							
		7	6	5	4	3	2	1	<0>
IICX0		0	0	0	0	0	0	0	CLX0

(6) I²C0 传送时钟设置方法

I²C0 传送时钟频率 (f_{SCL}) 是通过以下表达式计算的。

$$f_{SCL} = 1 / (m \times T + t_R + t_F)$$

m = 12, 24, 48, 54, 86, 88, 172, 198 (参见表 16-2 选择时钟设置。)

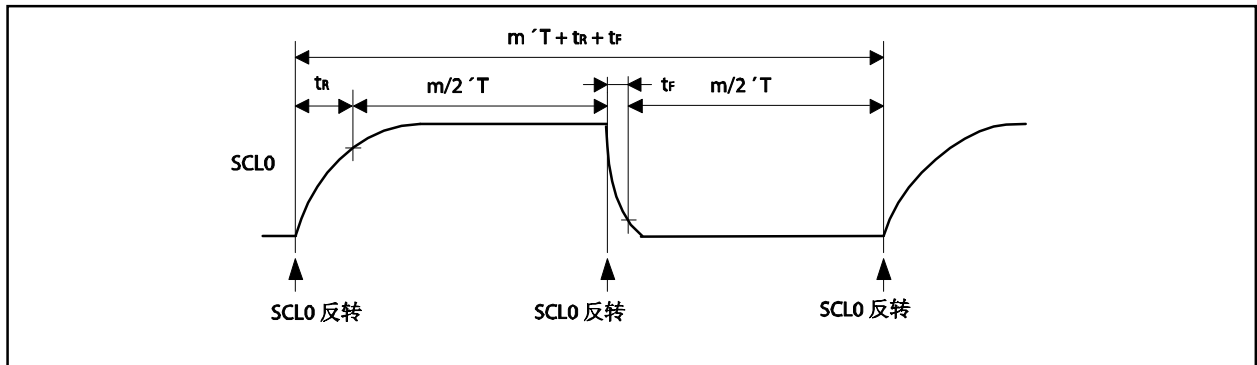
T: 1/f_{XX}

t_R: SCL0 上升时间

t_F: SCL0 下降时间

例如, 当 f_{XX} = 20 MHz, m = 54, t_R = 200 ns, t_F = 50 ns 时, I²C0 传送时钟频率 (f_{SCL}) 通过以下表达式计算。

$$f_{SCL} = 1 / (54 \times 50 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 339 \text{ kHz}$$



通过组合 IICCL0.SMC0、IICCL0.CL01、IICCL0.CL00 和 IICX0.CLX0 位设置选择时钟。

表 16-2. 时钟设置

IICX0	IICCL0			选择时钟	传送时钟 (fxx/m)	可设置的内部系统时钟频率 (fxx) 范围	运行模式	
	0 位	3 位	1 位					0 位
CLX0	SMC0	CL01	CL00					
0	0	0	0	fxx/2	fxx/88	4.0 MHz~8.38 MHz	正常模式 (SMC0 位= 0)	
0	0	0	1	fxx/2	fxx/172	8.38 MHz~16.76 MHz		
0	0	1	0	fxx	fxx/86	4.19 MHz~8.38 MHz		
0	0	1	1	fxx/3	fxx/198	16.0 MHz~19.8 MHz		
0	1	0	x	fxx/2	fxx/48	8 MHz~16.76 MHz	高速模式 (SMC0 位= 1)	
0	1	1	0	fxx	fxx/24	4 MHz~8.38 MHz		
0	1	1	1	fxx/3	fxx/54	16 MHz~20 MHz		
1	0	x	x	禁止设置				
1	1	0	x	fxx/2	fxx/24	8.00 MHz~8.38 MHz	高速模式 (SMC0 位= 1)	
1	1	1	0	fxx	fxx/12	4.00 MHz~4.19 MHz		
1	1	1	1	禁止设置				

备注 x: 忽略

(7) IIC 移位寄存器 0 (IIC0)

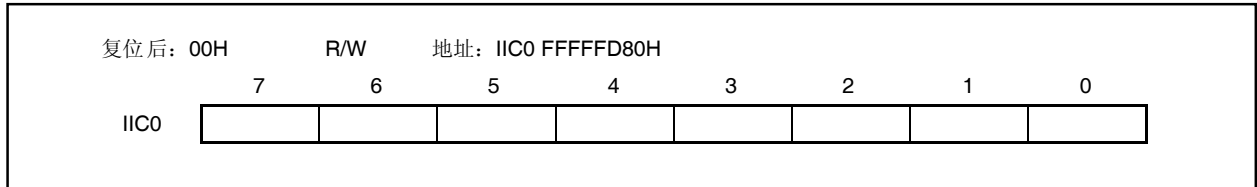
IIC0 移位寄存器用于与串行时钟同步的串行发送/接收（移位操作）。

IIC0 寄存器支持 8 位读写方式，但在数据发送过程中不能写入数据到 IIC0 移位寄存器。

只有在等待期间才能访问（读写）IIC0 移位寄存器。在通讯状态下禁止访问此寄存器。但对于主设备，只有在发送触发器位（IICC0.STT0 位）已经设置为 1 后才能写入 IIC0 移位寄存器。

当在等待期间写入 IIC0 移位寄存器时，等待取消，数据发送开始。

该寄存器复位后的值为 00H。



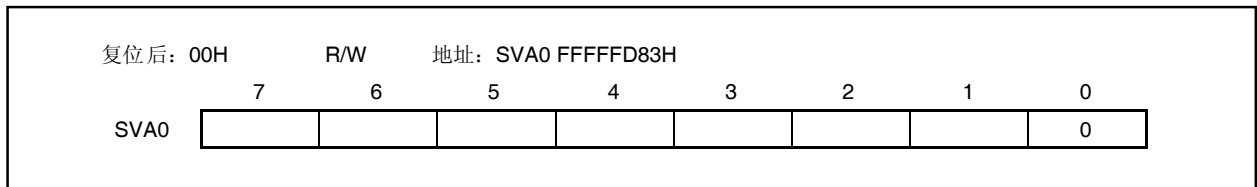
(8) 从地址寄存器 0 (SVA0)

SVA0 寄存器保存 IIC 总线从设备的地址。

但在 IICS0.STD0 位= 1（开始条件检测）时禁止重写此寄存器。

SVA0 寄存器支持 8 位读写方式，但 0 位固定为 0。

该寄存器复位后的值为 00H。



16.4 功能

16.4.1 引脚配置

串行时钟引脚（SCL0）和串行数据总线引脚（SDA0）配置如下。

SCL0 此引脚用于串行时钟输入和输出。

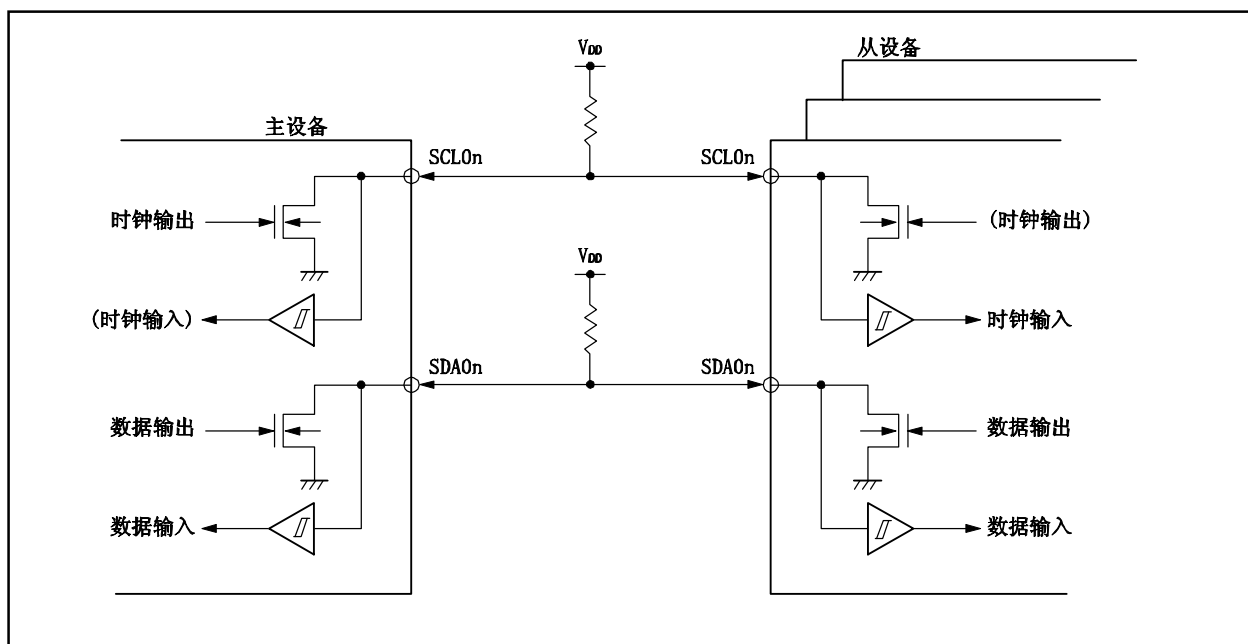
此引脚是主设备和从设备的 N-通道漏极开路输出。输入是施密特输入。

SDA0 此引脚用于串行数据输入和输出。

此引脚是主设备和从设备的 N-通道漏极开路输出。输入是施密特输入。

由于串行时钟线和串行数据总线是 N 通道漏极开路输出，要求外部提升电阻。

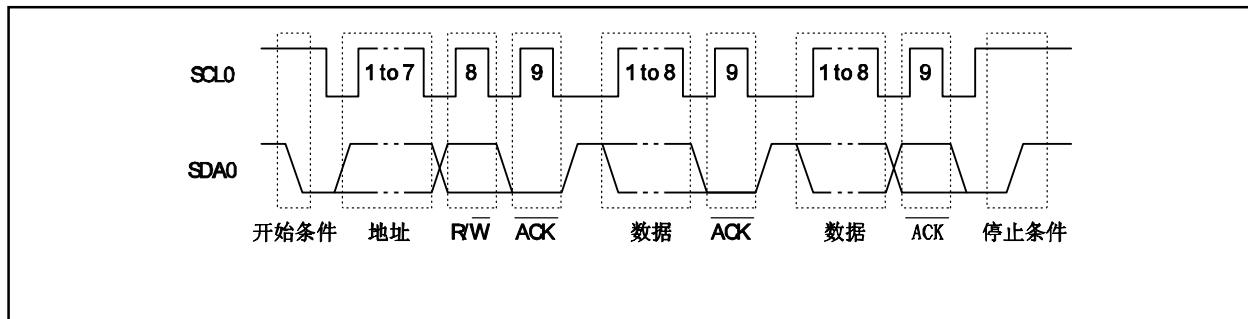
图 16-3. 引脚配置图



16.5 IIC 总线的定义和控制方法

以下部分描述 IIC 总线的串行数据通讯格式和 IIC 总线生成的状态。通过 IIC 总线的串行数据总线生成的“开始条件”、“地址”、“发送方向指定”、“数据”和“停止条件”的传送时序如下所示。

图 16-4. 内部 IC 总线的串行数据时序



主设备生成开始条件、从设备地址和停止条件。

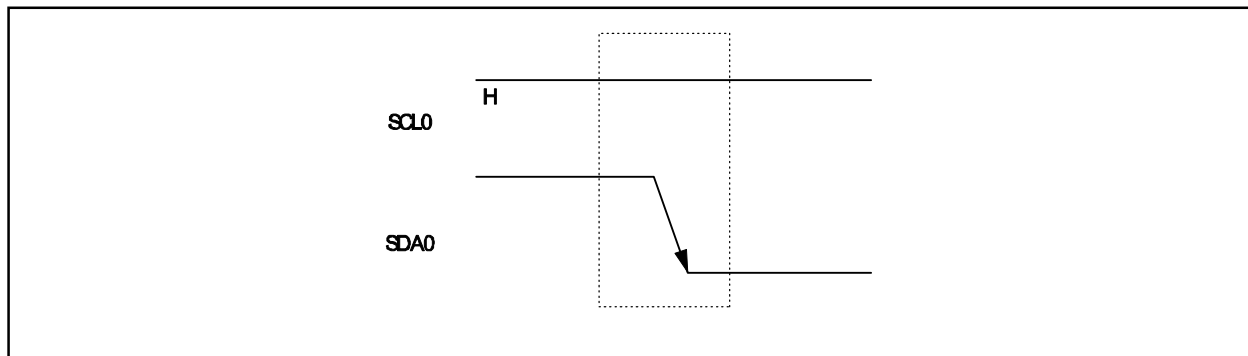
可以使用主设备或从设备生成 ACK。（一般情况下，由接收 8 位数据的设备生成）。

串行时钟（SCL0）由主设备连续输出。但在从设备中，可以延长 SCL0 的低电平期以及插入等待。

16.5.1 开始条件

当 SCL0 引脚在高电平，SDA0 引脚从高电平变为低电平时，达到开始条件。当主设备开始到从设备的串行发送时，生成 SCL0 引脚和 SDA0 引脚的开始条件。当设备作为从设备使用时，检测到开始条件。

图 16-5. 开始条件



当在检测到停止条件（IICS0.SPD0 位= 1）后设置 IICC0.STT0 位为 1 时生成开始条件。当检测到开始条件时，IICS0.STD0 位为 1。

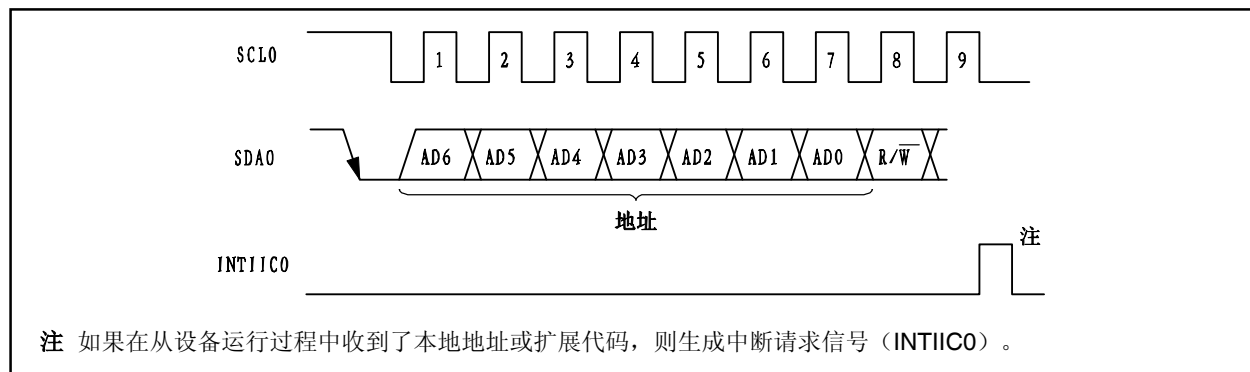
16.5.2 地址

开始条件后的 7 位数据定义为地址。

地址是一个 7 位数据段，输出用于选择一个通过总线与主设备连接的从设备。因此，通过总线连接的从设备必须有一个唯一的地址。

从设备包括检测开始条件的硬件并检查 7 位数据相等 SVA0 寄存器中储存的数据值。如果地址数据相等 SVA0 值，则选择从设备，并与主设备通讯直到主设备生成一个开始条件或停止条件。

图 16-6. 地址



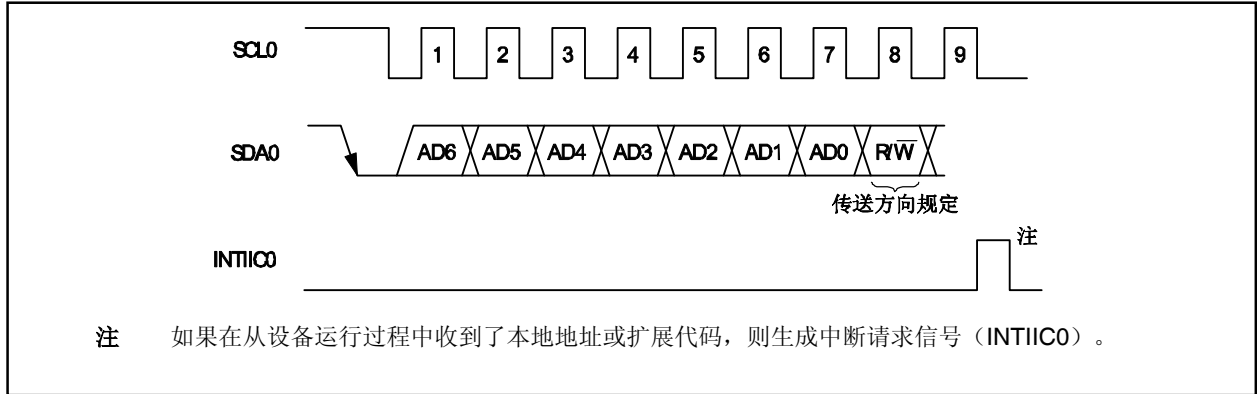
从设备地址和第八位（如下面的 **16.5.3 传送方向指定** 所述指定传送方向），同时写入到 IIC0 寄存器然后输出。接收到的地址被写入到 IIC0 寄存器。

从设备地址被分配到 IIC0 寄存器更高的 7 位。

16.5.3 传送方向指定

除了 7 位地址数据外，主设备还发送指定传送方向的一个位。当传送方向指定位的值是 0 时，它表示主设备在向从设备传送数据。当传送方向指定位的值是 1 时，它表示主设备在向从设备接收数据。

图 16-7. 传送方向指定



16.5.4 $\overline{\text{ACK}}$

$\overline{\text{ACK}}$ 用于确认传送和接收设备的串行数据状态。

接收设备每收到 8 个位的数据后返回 $\overline{\text{ACK}}$ 。

传送设备一般每传送 8 个位的数据后收到 $\overline{\text{ACK}}$ 。当从接收设备返回 $\overline{\text{ACK}}$ 时，认为接收正常，然后处理继续进行。 $\overline{\text{ACK}}$ 的检测根据 IIC0.ACKD0 位确认。

当主设备是接收设备时，在接收最后的数据后不返回 $\overline{\text{ACK}}$ 而是生成停止条件。当从设备是接收设备而没有返回 $\overline{\text{ACK}}$ 时，主设备会生成停止条件或重新启动条件，然后停止当前的传送。没有返回 $\overline{\text{ACK}}$ 可能由以下原因造成。

- (a) 接收不正常。
- (b) 接收了最后的数据。
- (c) 接收设备（从设备）没有指定的地址。

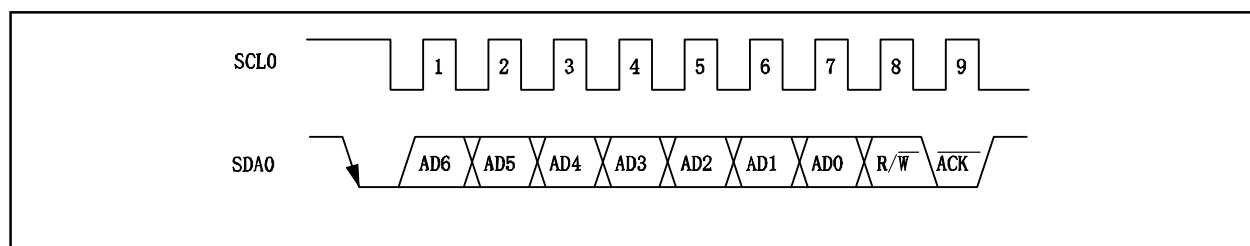
当接收设备在第九个时钟内设置 SDA0 线为低电平时，生成 $\overline{\text{ACK}}$ （正常接收）。

当 IIC0.ACKE0 位设为 1 时，启用自动 $\overline{\text{ACK}}$ 生成。在第 7 个地址位后第 8 个位的发送造成设置了 IIC0.TRC0 位。正常情况下，设置 ACKE0 位为 1 进行接收（TRC0 位=0）。

当从设备在接收时（TRC0 位=0 时），如果从设备不能接收数据或不需要接收更多的数据，将 ACKE0 位清零以向主设备表示不能再收更多的数据了。

类似的，当主设备在接收时（TRC0 位=0 时），如果不需要接收更多的数据了，将 ACKE0 位清零以阻止生成 $\overline{\text{ACK}}$ 。这样会通知从设备（传送设备）数据传送结束（发送停止）。

图 16-8. $\overline{\text{ACK}}$



当收到本地地址时，不论 ACKE0 位的值如何都会自动生成 $\overline{\text{ACK}}$ 。如果接收的地址不是本地地址（NACK）则不会生成 $\overline{\text{ACK}}$ 。

在接收扩展代码时，在生成 $\overline{\text{ACK}}$ 前将 ACKE0 位设为 1。

在数据接收过程中的 $\overline{\text{ACK}}$ 生成方法是基于如下所述的时序设定的。

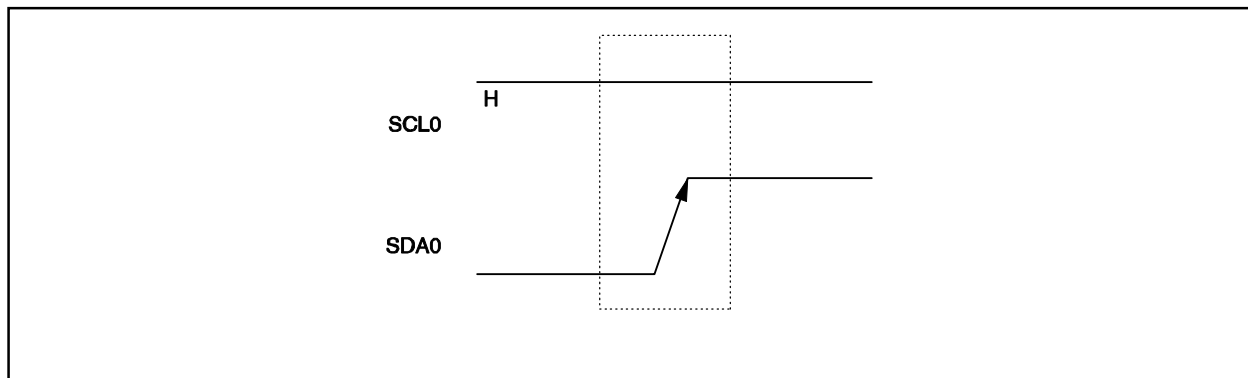
- 当选择 8 时钟等待时（IIC0.WTIM0 位=0）：
如果在等待状态取消前将 ACKE0 位设为 1，则在 SCL0n 引脚 8 时钟的下降沿生成 $\overline{\text{ACK}}$ 。
- 当选择 9 时钟等待时（IIC0.WTIM0 位=1）：
如果提前将 ACKE0 位设为 1，则生成 $\overline{\text{ACK}}$ 。

16.5.5 停止条件

当 SCL0 引脚处于高电平时，将 SDA0 引脚从低电平改至高电平会生成一个停止条件。

在完成了从主设备到从设备的串行传送后生成停止条件。当设备用作从设备时可以检测停止条件。

图 16-9. 停止条件



当 IICC0.SPT0 位设为 1 时会生成停止条件。当检测到停止条件时，IICS0.SPD0 位被设为 1，在 IICC0.SPIE0 位被设为 1 时则生成中断请求信号 (INTIIC0)。

16.5.6 等待状态

等待状态用于通知通讯主设备或从设备准备发送或接收数据，即处于等待状态中。

将 SCL0 引脚设为低电平则向通讯方通知等待状态。当取消了主设备和从设备的等待状态时，可以开始下一次数据发送。

图 16-10. 等待状态 (1/2)

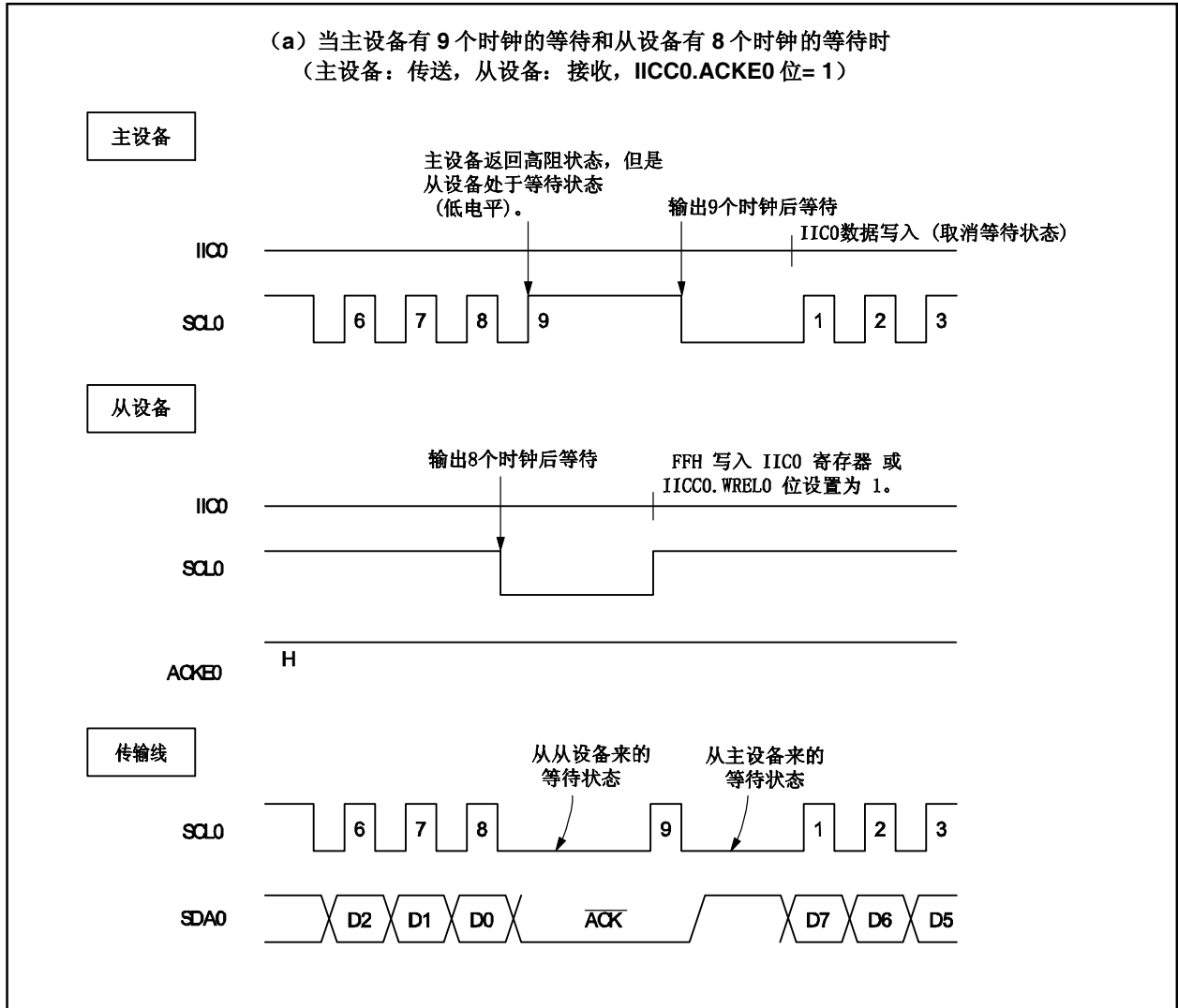
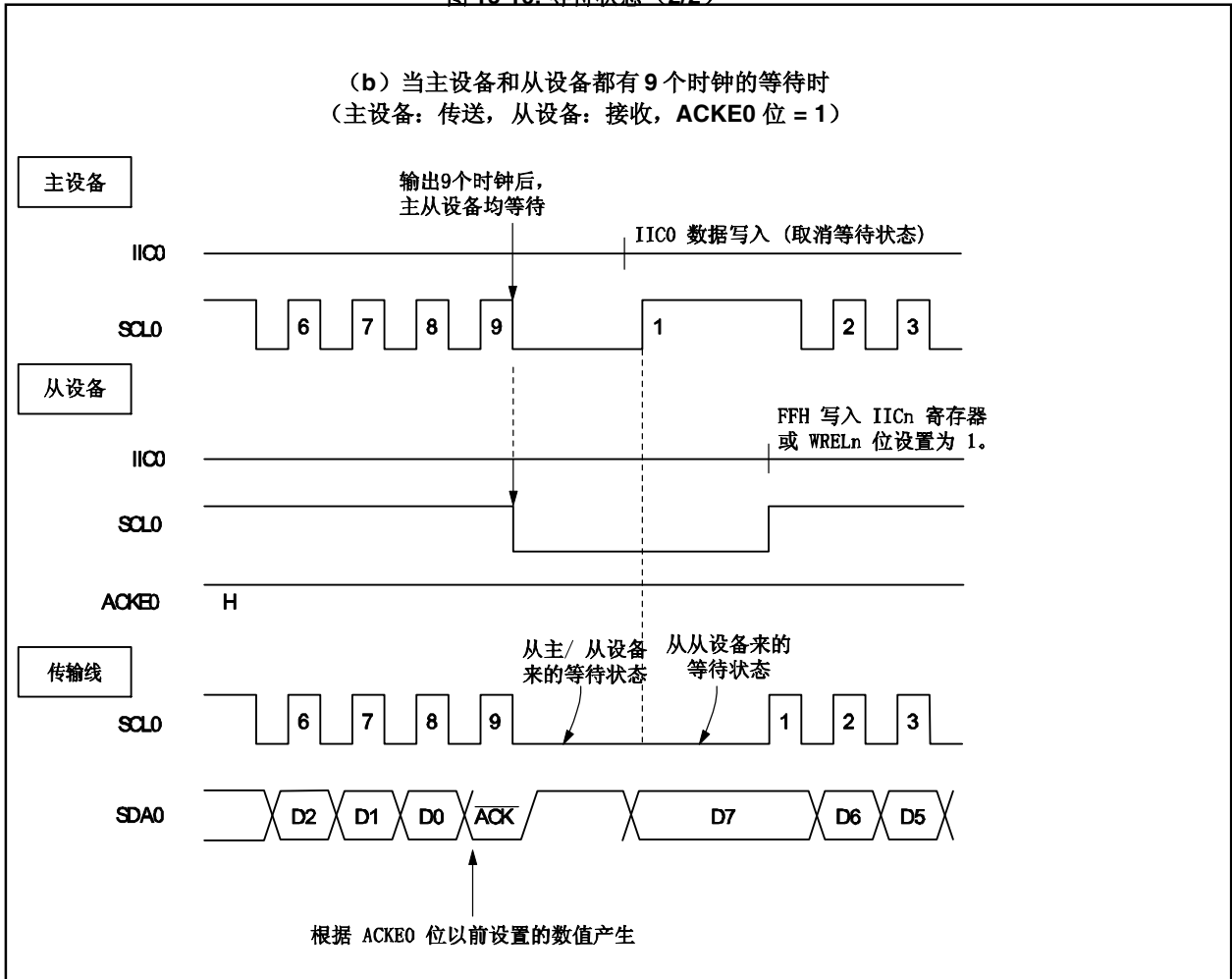


图 16-10. 等待状态 (2/2)



在生成开始条件后会自动生成等待状态。此外，等待状态会根据 IICC0.WTIMO 位的设置而自动生成。

一般情况下，当 IICC0.WRELO 位设为 1 或 FFH 写入到 IIC0 寄存器时，等待状态会被取消；传送方写入数据到 IIC0 寄存器会取消等待状态。

主设备也可以通过以下方法取消等待状态。

- 将 IICC0.STT0 位设为 1
- 将 IICC0.SPT0 位设为 1

16.5.7 等待状态取消方法

在 I^2C 的情况下，一般可以通过以下方法取消等待状态。

- 写入数据到 IIC0 寄存器
- 将 IICC0.WRELO 位设为 1（取消等待状态）
- 将 IICC0.STT0 位设为 1（生成开始条件）^注
- 将 IICC0.SPT0 位设为 1（生成停止条件）^注

注 仅限于主设备。

如果执行了等待状态取消的操作，则 I^2C 将取消等待状态并重新开始通讯。

当取消等待状态和发送数据（包括地址）时，写入数据到 IIC0 寄存器。

要在取消等待状态后接收数据或者要完成数据发送，则将 WRELO 位设为 1。

要在取消等待状态后生成重新启动条件，则将 STT0 位设为 1。

要在取消等待状态后生成停止条件，则将 SPT0 位设为 1。

对每个等待状态只能执行一次取消。

比如在通过将 WRELO 设为 1 取消等待状态后写入数据到 IIC0 寄存器，SDA0 线改变时序和 IIC0 寄存器写入时序之间的冲突可能会造成输出到 SDA0 线的的数据不正确。

即使在其他操作中，如果中途停止通讯，将 IICC0.IICE0 位清零会停止通讯，使得等待状态取消。

如果由于噪声等造成 IIC 总线死锁，将 IICC0.LRELO 位设为 1 会引起通讯操作退出，使得等待状态取消。

16.6 I²C 中断请求信号 (INTIIC0)

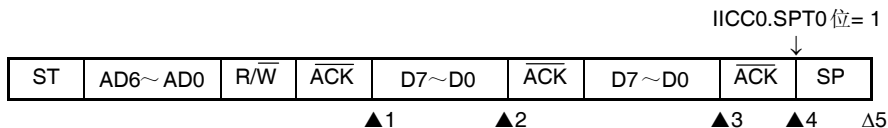
以下符号显示在 INTIIC0 中断请求信号生成时序和 INTIIC0 信号时序 IICS0 寄存器的值。

备注	ST:	开始条件
	AD6 到 AD0:	地址
	R/W:	发送方向指定
	$\overline{\text{ACK}}$:	确认
	D7 到 D0:	数据
	SP:	停止条件

16.6.1 主设备运行

(1) 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止 (正常传送/接收)

<1> 当 IICC0.WTIMO 位= 0 时

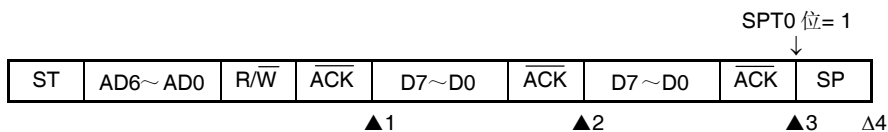


- ▲1: IICS0 寄存器 = 1000X110B
- ▲2: IICS0 寄存器 = 1000X000B
- ▲3: IICS0 寄存器 = 1000X000B (WTIMO 位 = 1^{*})
- ▲4: IICS0 寄存器 = 1000XX00B
- Δ 5: IICS0 寄存器 = 00000001B

注 要生成一个停止条件，将 WTIMO 位设为 1 并改变中断请求信号 (INTIIC0) 的生成。

- 备注
- ▲: 总是生成
 - Δ: 只有在 IICC0.SPIE0 位 = 1 时生成
 - X: 忽略

<2> 当 WTIMO 位= 1 时

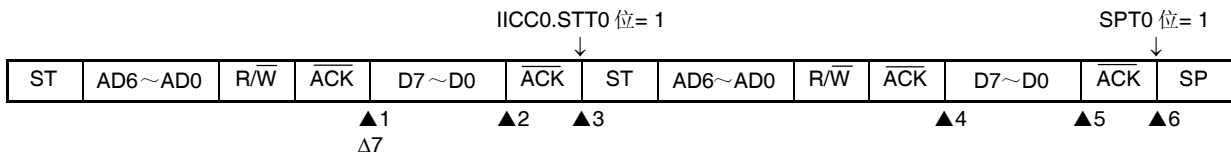


- ▲1: IICS0 寄存器 = 1000X110B
- ▲2: IICS0 寄存器 = 1000X100B
- ▲3: IICS0 寄存器 = 1000XX00B
- Δ 4: IICS0 寄存器 = 00000001B

- 备注
- ▲: 总是生成
 - Δ: 只有在 SPIE0 位 = 1 时生成
 - X: 忽略

(2) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止 (重新开始)

<1> 当 WTIMO 位= 0 时

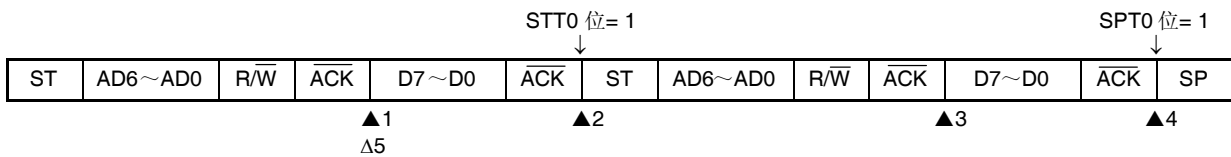


- ▲1: IICS0 寄存器 = 1000X110B
- ▲2: IICS0 寄存器 = 1000X000B (WTIMO 位= 1^{#1})
- ▲3: IICS0 寄存器 = 1000XX00B (WTIMO 位= 0^{#2})
- ▲4: IICS0 寄存器 = 1000X110B
- ▲5: IICS0 寄存器 = 1000X000B (WTIMO 位= 1^{#3})
- ▲6: IICS0 寄存器 = 1000XX00B
- Δ 7: IICS0 寄存器 = 00000001B

注: 1. 要生成一个开始条件, 将 WTIMO 位设为 1 并改变中断请求信号 (INTIIC0) 的生成。
 2. 将 WTIMO 位清零以复原设置。
 3. 要生成一个停止条件, 将 WTIMO 位设为 1 并改变中断请求信号 (INTIIC0) 的生成。

备注: ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

<2> 当 WTIMO 位= 1 时

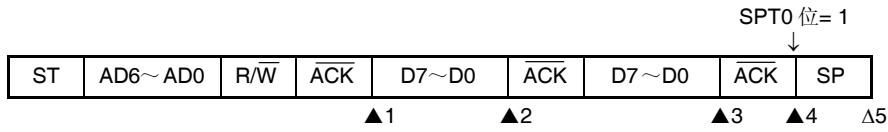


- ▲1: IICS0 寄存器 = 1000X110B
- ▲2: IICS0 寄存器 = 1000XX00B
- ▲3: IICS0 寄存器 = 1000X110B
- ▲4: IICS0 寄存器 = 1000XX00B
- Δ 5: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

(3) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止 (扩展代码传送)

<1> 当 WTIM0 位= 0 时

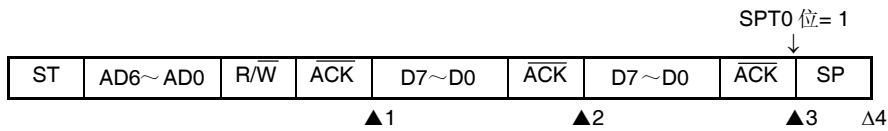


- ▲1: IICS0 寄存器 = 1010X110B
- ▲2: IICS0 寄存器 = 1010X000B
- ▲3: IICS0 寄存器 = 1010X000B (WTIM0 位= 1[※])
- ▲4: IICS0 寄存器 = 1010XX00B
- Δ 5: IICS0 寄存器 = 00000001B

注: 要生成一个停止条件, 将 WTIM0 位设为 1 并改变中断请求信号 (INTIIC0) 的生成。

备注: ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

<2> 当 WTIM0 位= 1 时



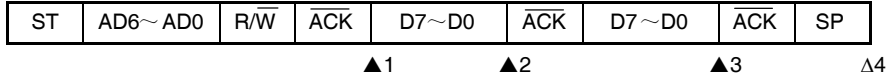
- ▲1: IICS0 寄存器 = 1010X110B
- ▲2: IICS0 寄存器 = 1010X100B
- ▲3: IICS0 寄存器 = 1010XX00B
- Δ 4: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

16.6.2 从设备运行（当接收从设备地址数据（地址相等）时）

(1) 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止

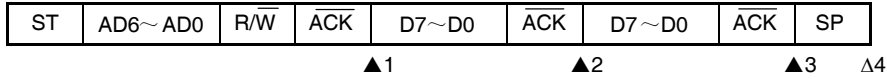
<1> 当 IICC0.WTIM0 位= 0 时



- ▲1: IICS0 寄存器 = 0001X110B
- ▲2: IICS0 寄存器 = 0001X000B
- ▲3: IICS0 寄存器 = 0001X000B
- Δ 4: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 IICC0.SPIE0 位= 1 时生成
 X: 忽略

<2> 当 WTIM0 位= 1 时

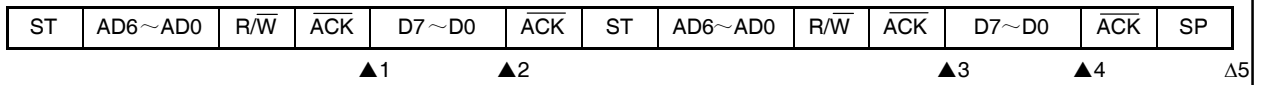


- ▲1: IICS0 寄存器 = 0001X110B
- ▲2: IICS0 寄存器 = 0001X100B
- ▲3: IICS0 寄存器 = 0001XX00B
- Δ 4: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

(2) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 开始

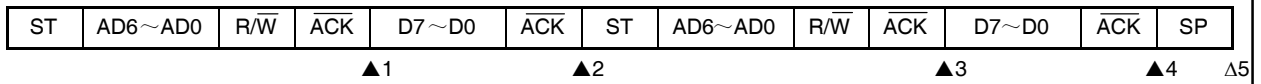
<1> 当 **WTIM0** 位= 0 时 (重新启动后, 地址相等)



- ▲1: IICS0 寄存器 = 0001X110B
- ▲2: IICS0 寄存器 = 0001X000B
- ▲3: IICS0 寄存器 = 0001X110B
- ▲4: IICS0 寄存器 = 0001X000B
- Δ 5: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 **SPIE0** 位= 1 时生成
 X: 忽略

<2> 当 **WTIM0** 位= 1 时 (重新启动后, 地址相等)

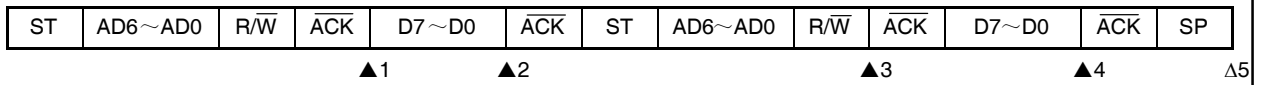


- ▲1: IICS0 寄存器 = 0001X110B
- ▲2: IICS0 寄存器 = 0001XX00B
- ▲3: IICS0 寄存器 = 0001X110B
- ▲4: IICS0 寄存器 = 0001XX00B
- Δ 5: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 **SPIE0** 位= 1 时生成
 X: 忽略

(3) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 代码 ~ 数据 ~ 停止

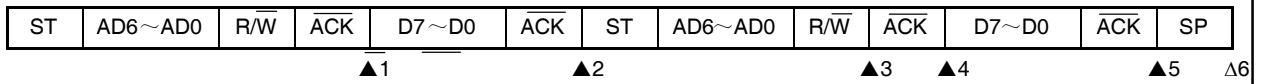
<1> 当 **WTIM0** 位= 0 时 (重新启动后, 扩展代码接收)



- ▲1: IICS0 寄存器 = 0001X110B
- ▲2: IICS0 寄存器 = 0001X000B
- ▲3: IICS0 寄存器 = 0010X010B
- ▲4: IICS0 寄存器 = 0010X000B
- Δ 5: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

<2> 当 **WTIM0** 位= 1 时 (重新启动后, 扩展代码接收)



- ▲1: IICS0 寄存器 = 0001X110B
- ▲2: IICS0 寄存器 = 0001XX00B
- ▲3: IICS0 寄存器 = 0010X010B
- ▲4: IICS0 寄存器 = 0010X110B
- ▲5: IICS0 寄存器 = 0010XX00B
- Δ 6: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

(4) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 WTIMO 位= 0 时 (重新启动后, 地址不相等 (= 非扩展代码))

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	ST	AD6~AD0	R/W	ACK	D7~D0	ACK	SP
----	---------	-----	-----	-------	-----	----	---------	-----	-----	-------	-----	----

▲1 ▲2 ▲3
 Δ4

- ▲1: IICS0 寄存器 = 0001X110B
- ▲2: IICS0 寄存器 = 0001X000B
- ▲3: IICS0 寄存器 = 00000110B
- Δ 4: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

<2> 当 WTIMO 位= 1 时 (重新启动后, 地址不相等 (=非扩展代码))

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	ST	AD6~AD0	R/W	ACK	D7~D0	ACK	SP
----	---------	-----	-----	-------	-----	----	---------	-----	-----	-------	-----	----

▲1 ▲2 ▲3
 Δ4

- ▲1: IICS0 寄存器 = 0001X110B
- ▲2: IICS0 寄存器 = 0001XX00B
- ▲3: IICS0 寄存器 = 00000110B
- Δ 4: IICS0 寄存器 = 00000001B

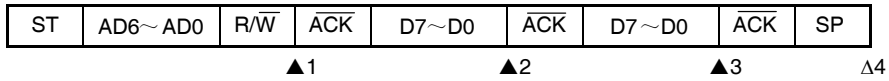
备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

16.6.3 从设备运行（当接收扩展代码时）

当接收扩展代码时总是在通讯状态下。

(1) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止

<1> 当 IICC0.WTIMO 位= 0 时



▲1: IICS0 寄存器 = 0010X010B

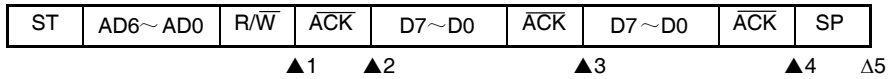
▲2: IICS0 寄存器 = 0010X000B

▲3: IICS0 寄存器 = 0010X000B

Δ 4: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 IICC0.SPIE0 位= 1 时生成
 X: 忽略

<2> 当 WTIMO 位= 1 时



▲1: IICS0 寄存器 = 0010X010B

▲2: IICS0 寄存器 = 0010X110B

▲3: IICS0 寄存器 = 0010X100B

▲4: IICS0 寄存器 = 0010XX00B

Δ 5: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

(2) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 **WTIMO** 位= 0 时 (重新启动后, 地址相等)

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	ST	AD6~AD0	R/W	ACK	D7~D0	ACK	SP
			▲1		▲2					▲3	▲4	
			Δ5									

▲1: IICS0 寄存器 = 0010X010B
 ▲2: IICS0 寄存器 = 0010X000B
 ▲3: IICS0 寄存器 = 0001X110B
 ▲4: IICS0 寄存器 = 0001X000B
 Δ 5: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

<2> 当 **WTIMO** 位= 1 时 (重新启动后, 地址相等)

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	ST	AD6~AD0	R/W	ACK	D7~D0	ACK	SP
			▲1	▲2		▲3				▲4		▲5
			Δ6									

▲1: IICS0 寄存器 = 0010X010B
 ▲2: IICS0 寄存器 = 0010X110B
 ▲3: IICS0 寄存器 = 0010XX00B
 ▲4: IICS0 寄存器 = 0001X110B
 ▲5: IICS0 寄存器 = 0001XX00B
 Δ 6: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

(3) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 代码 ~ 数据 ~ 停止

<1> 当 **WTIMO** 位= 0 时 (重新启动后, 扩展代码接收)

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	ST	AD6~AD0	R/W	ACK	D7~D0	ACK	SP
			▲1		▲2				▲3		▲4	
			△5									

▲1: IICS0 寄存器 = 0010X010B
 ▲2: IICS0 寄存器 = 0010X000B
 ▲3: IICS0 寄存器 = 0010X010B
 ▲4: IICS0 寄存器 = 0010X000B
 △5: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 △: 只有在 **SPIE0** 位= 1 时生成
 X: 忽略

<2> 当 **WTIMO** 位= 1 时 (重新启动后, 扩展代码接收)

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	ST	AD6~AD0	R/W	ACK	D7~D0	ACK	SP
			▲1	▲2		▲3			▲4	▲5		▲6
												△7

▲1: IICS0 寄存器 = 0010X010B
 ▲2: IICS0 寄存器 = 0010X110B
 ▲3: IICS0 寄存器 = 0010XX00B
 ▲4: IICS0 寄存器 = 0010X010B
 ▲5: IICS0 寄存器 = 0010X110B
 ▲6: IICS0 寄存器 = 0010XX00B
 △7: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 △: 只有在 **SPIE0** 位= 1 时生成
 X: 忽略

(4) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 **WTIM0** 位= 0 时 (重新启动后, 地址不相等 (= 非扩展代码))

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	ST	AD6~AD0	R/W	ACK	D7~D0	ACK	SP
			▲1		▲2					▲3		Δ4

▲1: IICS0 寄存器 = 0010X010B
 ▲2: IICS0 寄存器 = 0010X000B
 ▲3: IICS0 寄存器 = 00000110B
 Δ 4: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

<2> 当 **WTIM0** 位= 1 时 (重新启动后, 地址不相等 (= 非扩展代码))

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	ST	AD6~AD0	R/W	ACK	D7~D0	ACK	SP
			▲1	▲2		▲3				▲4		Δ5

▲1: IICS0 寄存器 = 0010X010B
 ▲2: IICS0 寄存器 = 0010X110B
 ▲3: IICS0 寄存器 = 0010XX00B
 ▲4: IICS0 寄存器 = 00000110B
 Δ 5: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

16.6.4 无通讯运行

(1) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止



16.6.5 仲裁失败运行（在仲裁失败后作为从设备运行）

当作为多主系统中的主设备使用时，通过读取 IICS0.MSTS0 位查看仲裁结果以查看每个 INTIIC0 中断出现时的仲裁结果。

(1) 在传送从地址数据的过程中出现仲裁失败

<1> 当 IICC0.WTIMO 位= 0 时

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	D7~D0	ACK	SP
			▲1		▲2		▲3	Δ4

▲1: IICS0 寄存器 = 0101X110B
 ▲2: IICS0 寄存器 = 0001X000B
 ▲3: IICS0 寄存器 = 0001X000B
 Δ4: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 IICC0.SPIE0 位= 1 时生成
 X: 忽略

<2> 当 WTIMO 位= 1 时

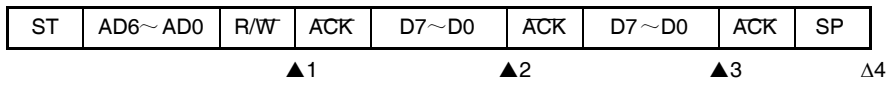
ST	AD6~AD0	R/W	ACK	D7~D0	ACK	D7~D0	ACK	SP
			▲1		▲2		▲3	Δ4

▲1: IICS0 寄存器 = 0101X110B
 ▲2: IICS0 寄存器 = 0001X100B
 ▲3: IICS0 寄存器 = 0001XX00B
 Δ4: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

(2) 在传送扩展代码的过程中出现仲裁失败

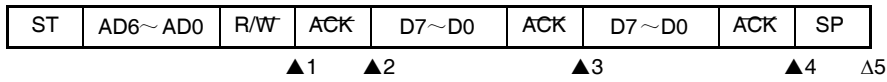
<1> 当 WTIMO 位= 0 时



- ▲1: IICS0 寄存器 = 0110X010B
- ▲2: IICS0 寄存器 = 0010X000B
- ▲3: IICS0 寄存器 = 0010X000B
- Δ 4: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

<2> 当 WTIMO 位= 1 时



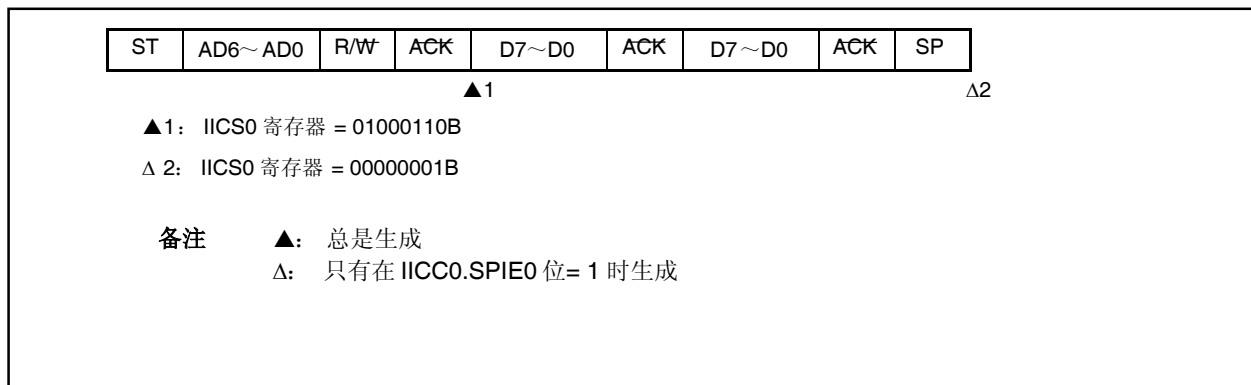
- ▲1: IICS0 寄存器 = 0110X010B
- ▲2: IICS0 寄存器 = 0010X110B
- ▲3: IICS0 寄存器 = 0010X100B
- ▲4: IICS0 寄存器 = 0010XX00B
- Δ 5: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

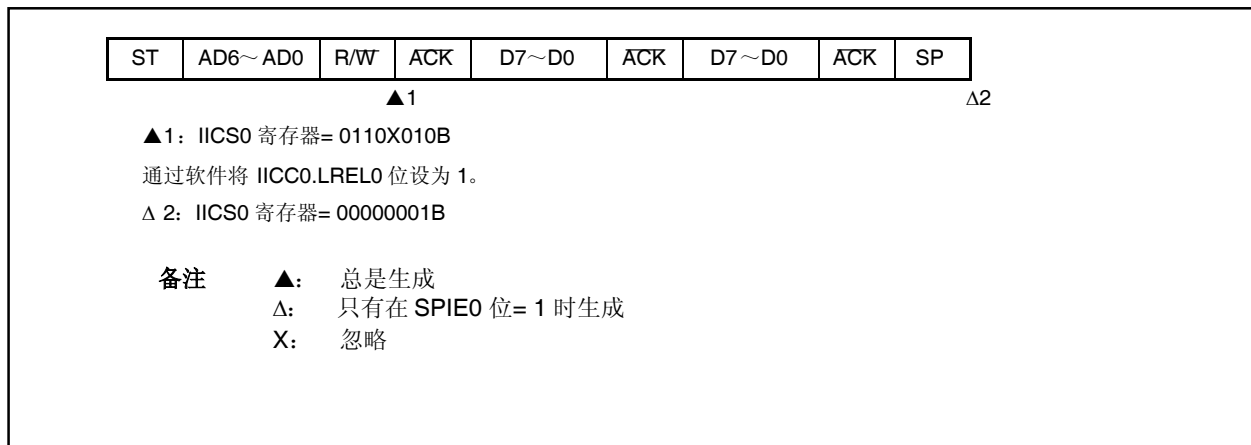
16.6.6 当出现仲裁失败时的运行（在仲裁失败后无通讯）

当作为多主系统中的主设备使用时，通过读取 IICS0.MSTS0 位查看仲裁结果以查看每个 INTIIC0 中断出现时的仲裁结果。

(1) 在传送从地址数据的过程中出现仲裁失败

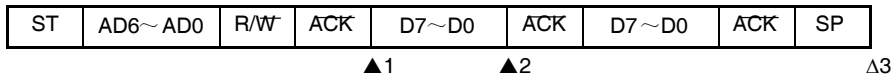


(2) 在传送扩展代码的过程中出现仲裁失败



(3) 在传送数据的过程中出现仲裁失败

<1> 当 IIC0.WTIM0 位= 0 时



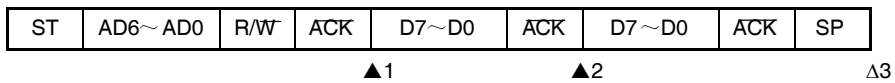
▲1: IICS0 寄存器 = 10001110B

▲2: IICS0 寄存器 = 01000000B

Δ 3: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成

<2> 当 WTIM0 位= 1 时



▲1: IICS0 寄存器 = 10001110B

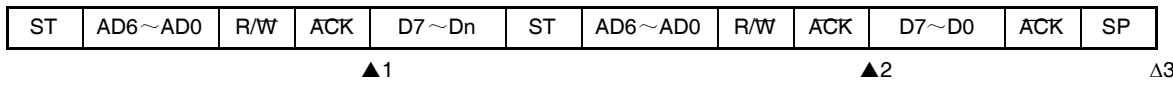
▲2: IICS0 寄存器 = 01000100B

Δ 3: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成

(4) 由于在数据传送过程中的重启条件而出现仲裁失败时

<1> 非扩展代码 (例如: 地址不相等)



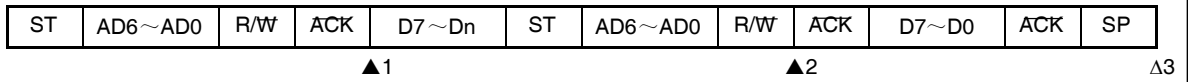
▲1: IICS0 寄存器 = 1000X110B

▲2: IICS0 寄存器 = 01000110B

Δ 3: IICS0 寄存器 = 00000001B

- 备注**
1. ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略
 2. Dn = D6~D0

<2> 扩展代码



▲1: IICS0 寄存器 = 1000X110B

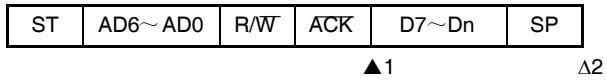
▲2: IICS0 寄存器 = 0110X010B

通过软件将 IICC0.LRELO 位设为 1。

Δ 3: IICS0 寄存器 = 00000001B

- 备注**
1. ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略
 2. Dn = D6~D0

(5) 由于在数据传送过程中的停止条件而出现仲裁失败时



▲1: IICS0 寄存器 = 1000X110B

Δ 2: IICS0 寄存器 = 01000001B

- 备注 s
1. ▲: 总是生成
 Δ: 只有在 IICC0.SPIE0 位= 1 时生成
 X: 忽略
 2. Dn = D6~D0

(6) 由于当尝试生成开始条件时 SDA0n 引脚的低电平而出现仲裁损失时

<1> 当 WTIM0 位= 0 时

IICC0.STT0 位= 1
↓

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	D7~D0	ACK	D7~D0	ACK	SP
			▲1		▲2	▲3	▲4			Δ5

▲1: IICS0 寄存器 = 1000X110B
 ▲2: IICS0 寄存器 = 1000X000B (WTIM0 位= 1)
 ▲3: IICS0 寄存器 = 1000X100B (WTIM0 位= 0)
 ▲4: IICS0 寄存器 = 01000000B
 Δ 5: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

<2> 当 WTIM0 位= 1 时

IICC0.STT0 位= 1
↓

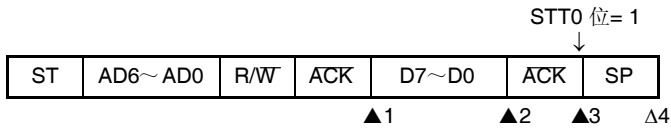
ST	AD6~AD0	R/W	ACK	D7~D0	ACK	D7~D0	ACK	D7~D0	ACK	SP
			▲1		▲2		▲3			Δ4

▲1: IICS0 寄存器 = 1000X110B
 ▲2: IICS0 寄存器 = 1000X100B
 ▲3: IICS0 寄存器 = 01000100B
 Δ 4: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

(7) 由于当尝试生成重新开始条件时的停止条件而出现仲裁损失时

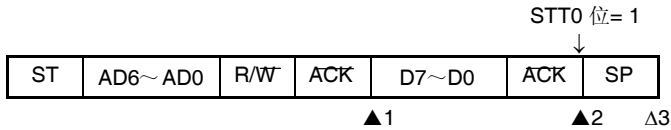
<1> 当 WTIMO 位= 0 时



- ▲1: IICS0 寄存器 = 1000X110B
- ▲2: IICS0 寄存器 = 1000X000B (WTIMO 位= 1)
- ▲3: IICS0 寄存器 = 1000XX00B
- Δ 4: IICS0 寄存器 = 01000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

<2> 当 WTIMO 位= 1 时

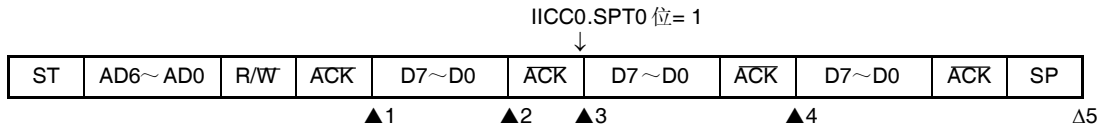


- ▲1: IICS0 寄存器 = 1000X110B
- ▲2: IICS0 寄存器 = 1000XX00B
- Δ 3: IICS0 寄存器 = 01000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

(8) 由于当尝试生成停止条件时 SDA0n 引脚的低电平而出现仲裁损失时

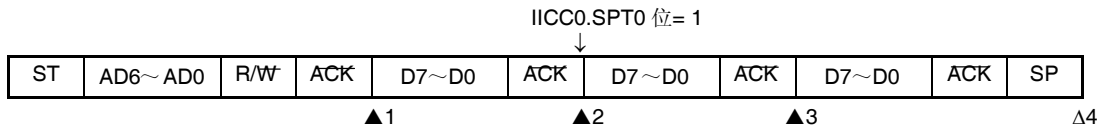
<1> 当 WTIM0 位= 0 时



- ▲1: IICS0 寄存器 = 1000X110B
- ▲2: IICS0 寄存器 = 1000X000B (WTIM0 位= 1)
- ▲3: IICS0 寄存器 = 1000X100B (WTIM0 位= 0)
- ▲4: IICS0 寄存器 = 01000100B
- Δ 5: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

<2> 当 WTIM0 位= 1 时



- ▲1: IICS0 寄存器 = 1000X110B
- ▲2: IICS0 寄存器 = 1000X100B
- ▲3: IICS0 寄存器 = 01000100B
- Δ 4: IICS0 寄存器 = 00000001B

备注 ▲: 总是生成
 Δ: 只有在 SPIE0 位= 1 时生成
 X: 忽略

16.7 中断请求信号 (INTIIC0) 生成时序和等待控制

IICC0.WTIM0 位的设置确定 INTIIC0 信号生成以及相应的等待控制的时序，如下所示。

表 16-3. INTIIC0 信号生成时序和等待控制

WTIM0 位	在从设备运行中			在主设备运行中		
	地址	数据接收	数据传送	地址	数据接收	数据传送
0	9 ^{#1, 2}	8 ^{#2}	8 ^{#2}	9	8	8
1	9 ^{#1, 2}	9 ^{#2}	9 ^{#2}	9	9	9

- 注**
1. 当相等设置到 SVA0 寄存器的地址时，从设备的 INTIIC0 信号和等待期出现在第九个时钟的下降沿。此时，不论设置到 IICC0.ACKE0 位的值如何，生成 ACK。对于已经收到扩展代码的从设备，在第八个时钟的下降沿出现 INTIIC0 信号。当在重新启动后地址不相等时，在第九个时钟的下降沿产生 INTIIC0 信号，但不出现等待。
 2. 如果收到的地址不相等 SVA0 寄存器的内容，而没有收到扩展代码，不出现 INTIIC0 信号或等待。

备注 表中的数字表示串行时钟信号的数量。中断请求和等待控制在这些时钟信号的下降沿同步。

(1) 在地址传送/接收过程中

- 从设备的运行：中断和等待时序根据注 1 和注 2 中条件确定而不论 WTIM0 位。
- 主设备的运行：中断和等待时序出现在在第九个时钟的下降沿而不论 WTIM0 位。

(2) 在数据接收过程中

- 主/从设备的运行：根据 WTIM0 位确定中断和等待时序。

(3) 在数据传送过程中

- 主/从设备的运行：根据 WTIM0 位确定中断和等待时序。

(4) 等待取消的方法

可以通过以下方法取消等待状态。

- 写入数据到 IIC0 寄存器
- 设置 IICC0.WREL0 位（取消等待状态）
- 设置 IICC0.STT0 位（生成开始条件）^注
- 设置 IICC0.SPT0 位（生成停止条件）^注

注 仅限于主设备。

当选择了 8 个时钟的等待（WTIM0 位=0）时，必须在等待取消前确定是否生成 \overline{ACK} 。

(5) 停止条件的检测

当检测到停止条件时生成 INTIIC0 信号。

16.8 地址相等检测方法

在 IIC 总线模式下，主设备可以通过传送相应的从设备地址选择一个特定的从设备。

地址相等检测由硬件自动完成。当本地地址设为 SVA0 寄存器且设为 SVA0 寄存器的地址与主设备发送的从设备地址，或者当收到扩展代码时，出现 INTIIC0 中断请求信号。

16.9 错误检测

在 IIC 总线模式下，由发送设备的 IIC0 寄存器捕捉数据传送过程中串行数据总线（SDA0）的状态，这样就可以比较数据发送前的 IIC0 寄存器数据和传送后的 IIC0 寄存器数据从而检测发送错误。当比较的数据值不相等时则认为出现了发送错误。

16.10 扩展代码

- (1) 当接收数据的上四位是 0000 或 1111，为扩展代码接收设置扩展代码标记 (EXC0)，并在第 8 个时钟的下降沿发出中断请求信号 (INTIIC0)。
保存在 SVA0 寄存器中的本地地址不收影响。
- (2) 如果通过 10 位地址发送设置 11110xx0 到 SVA0 寄存器，并且从主设备发送 11110xx0，则结果如下。注意在第 8 个时钟的下降沿出现的 INTIIC0 信号。
- 高 4 位数据相等：IICS0.EXC0 位= 1
 - 7 位数据相等：IICS0.COI0 位= 1
- (3) 由于 INTIIC0 信号出现后的处理根据扩展代码后的数据而不同，此项处理由软件进行。即使地址不相等，接收扩展代码的从设备总是在通讯状态中。
比如，在收到扩展代码后不希望作为从设备运行，可将 IICC0.LRELO 位设为 1，CPU 将进入下一个通讯等待状态。

表 16-4. 扩展代码位定义

从设备地址	R/W 位	描述
0000 000	0	一般呼叫地址
0000 000	1	开始字节
0000 001	X	CBUS 地址
0000 010	X	为不同总线格式预定的地址
1111 0xx	X	10 位从设备地址指定

16.11 仲裁

当主设备同时生成一个开始条件时（在 IICS0.STD0 位设为 1 前将 IICC0.STT0 位设为 1 时），主设备之间通讯的随着时钟数量的调整进行直到数据出现不同。这种操作称为仲裁。

当一个主设备在仲裁中失败时，则通过仲裁失败出现的时序将仲裁失败标记（IICS0.ALD0 位）设为（1），并设置 SCL0 和 SDA0 线为高阻抗，从而释放总线。

仲裁失败是基于下一个中断请求信号（INTIIC0）（第 8 或 9 个时钟，当检测到停止条件时等）以及通过软件设置的 ALD0 位= 1 检测的。

关于中断请求时序的更多细节，敬请参阅 16.6 IIC 中断请求信号（INTIIC0）。

图 16-11. 仲裁时序实例

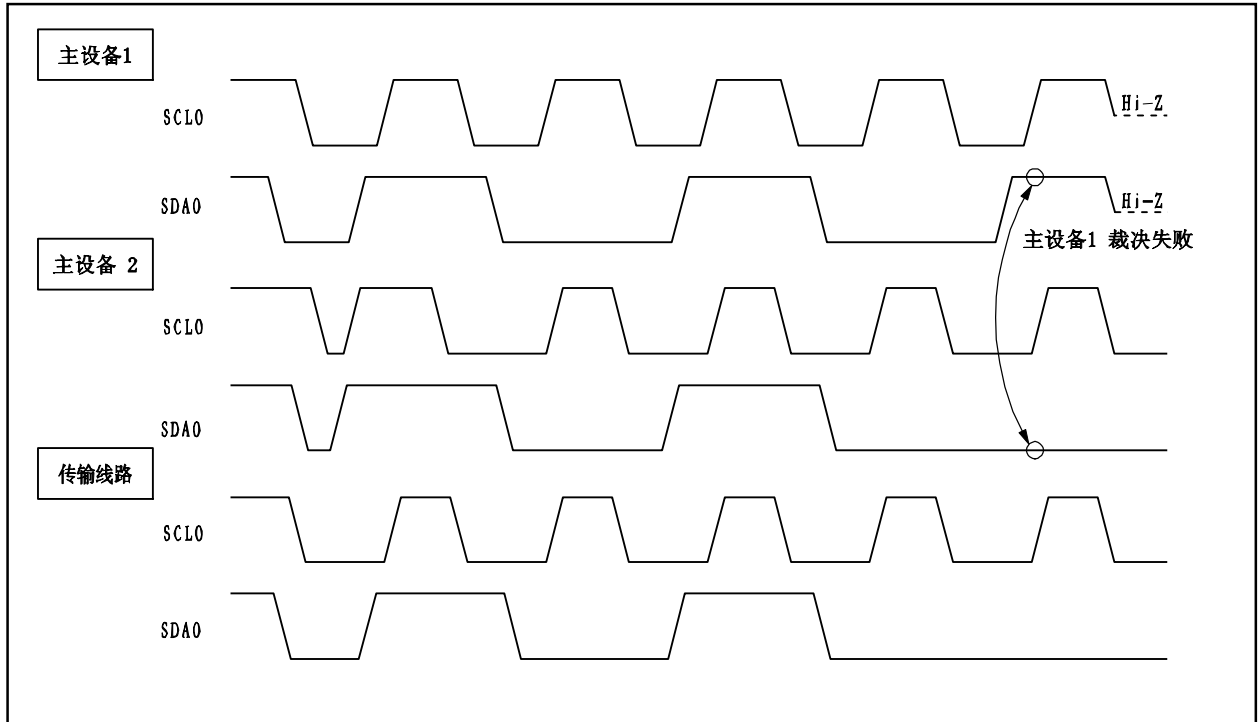


表 16-5: 仲裁过程中的状态和中断请求生成时序

仲裁过程中的状态	中断请求生成时序
在地址传送过程中	在字节传送后的第八或第九个时钟的下降沿 ^{注1}
在地址传送后读写数据	
在扩展代码传送过程中	
在扩展代码传送后读写数据	
在数据发送过程中	
在数据接收后 ACK 传送期间内	
在数据发送过程中检测到重新启动条件时	
在数据发送过程中检测到停止条件时	当生成停止条件时 (当 IICC0.SPIE0= 1 时) ^{注2}
在尝试生成一个重新开始条件过程中 SDA0 引脚在低电平时	在字节传送后的第八或第九个时钟的下降沿 ^{注1}
在尝试生成一个重新开始条件过程中检测到停止条件时	当生成停止条件时 (当 SPIE0 位= 1 时) ^{注2}
在尝试生成一个停止条件过程中 SDA0 引脚在低电平时	在字节传送后的第八或第九个时钟的下降沿 ^{注1}
在尝试生成一个停止条件过程中 SCL0 引脚在低电平时	

- 注
1. 当 IICC0.WTIM0 位= 1 时, 中断请求在第九个时钟的下降沿出现。当 WTIM0 位= 0 并收到扩展代码的从设备地址时, 中断请求在第八个时钟的下降沿出现。
 2. 当有可能出现仲裁时, 为主设备运行的运行设置 SPIE0 位= 1。

16.12 唤醒功能

IIC 总线从功能在收到本地地址或扩展代码时生成一个中断请求信号 (INTIIC0)。

此功能通过防止地址不相等时出现不必要的中断请求, 从而提高处理的有效性。

当检测到开始条件时, 设定为唤醒等待模式。当由于仲裁失败可能改变主设备 (生成开始条件) 而向从设备发送地址时, 唤醒等待模式生效。

但当检测到停止条件时, 会忽略唤醒功能而设置 IICC0.SPIE0 位, 从而确定是否启用或禁用中断请求。

16.13 通讯预定

16.13.1 当启用通讯预定功能时 (IICF0.IICRSV0 位= 0)

要在当前没有使用总线是启动主设备通讯，可以在总线释放的情况下通过通讯预定启用开始条件的发送。当没有使用总线时有两种模式。

- 当仲裁造成主设备或从设备都不运行时
- 当接收到扩展代码且从设备运行禁止时 (当 IICC0.LRELO 位设为“1”时没有返回 $\overline{\text{ACK}}$ 而总线释放)。

如果当没有使用总线时设置 IICC0.STT0 位为 (1)，则在总线释放后 (检测到停止条件后) 自动生成开始条件并设置等待状态。

通过将 IICC0.SPIE0 位设为 1，检测由于中断请求 (INTIIC0) 的出现而造成的总线释放 (检测停止条件)，然后写入地址到 IIC0 寄存器，从而自动作为主设备启用通讯。在检测到一个停止条件之前，写入数据到 IIC0 寄存器的被设为无效。

当已经将 STT0 位设为 (1) 时，则根据总线状态确定运行模式 (作为开始条件或通讯预定)。

如果总线已经释放 生成一个开始条件

如果总线没有释放 (等待模式) 通讯预定

要检测为 STT0 位确定的运行模式，设置 STT0 位 (1)，等待等待周期，然后查看 IICS0.MSTS0 位。

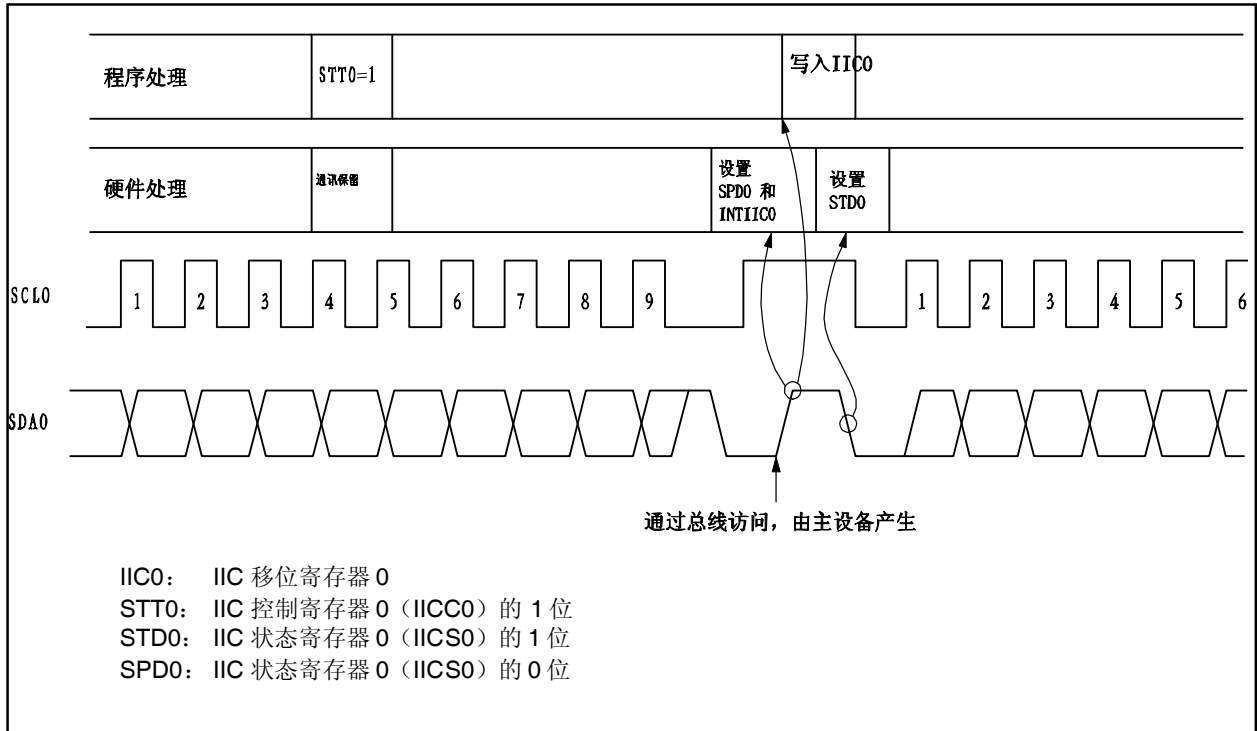
应通过软件设置的等待周期如表 16-6 中所列。可以通过 IICX0.CLX0、IICCL0.SMC0、IICCL0.CL01 和 IICCL0.CL00 位的设定设置等待周期。

表 16-6. 等待周期

CLX0	SMC0	CL01	CL00	选择的时钟	等待周期
0	0	0	0	$f_{xx}/2$	46 个时钟
0	0	0	1	$f_{xx}/2$	86 个时钟
0	0	1	0	f_{xx}	43 个时钟
0	0	1	1	$f_{xx}/3$	102 个时钟
0	1	0	1/0	$f_{xx}/2$	30 个时钟
0	1	1	0	f_{xx}	15 个时钟
0	1	1	1	$f_{xx}/3$	36 个时钟
1	1	0	1/0	$f_{xx}/2$	18 个时钟
1	1	1	0	f_{xx}	9 个时钟

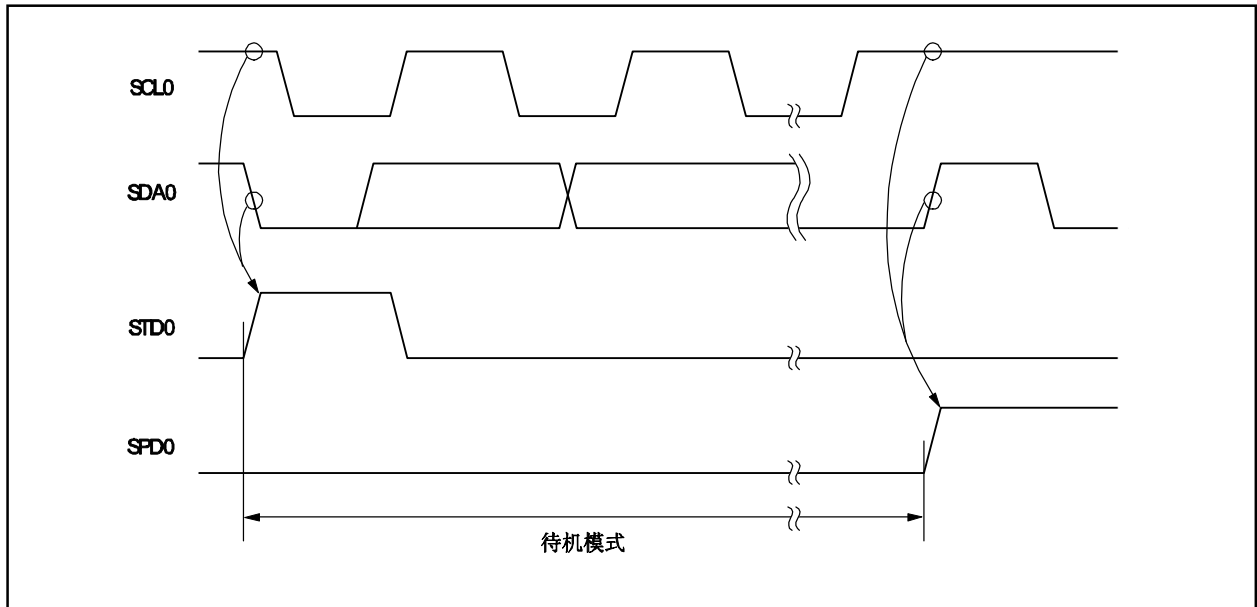
通讯预定时序如下所示。

图 16-12. 通讯预定时序



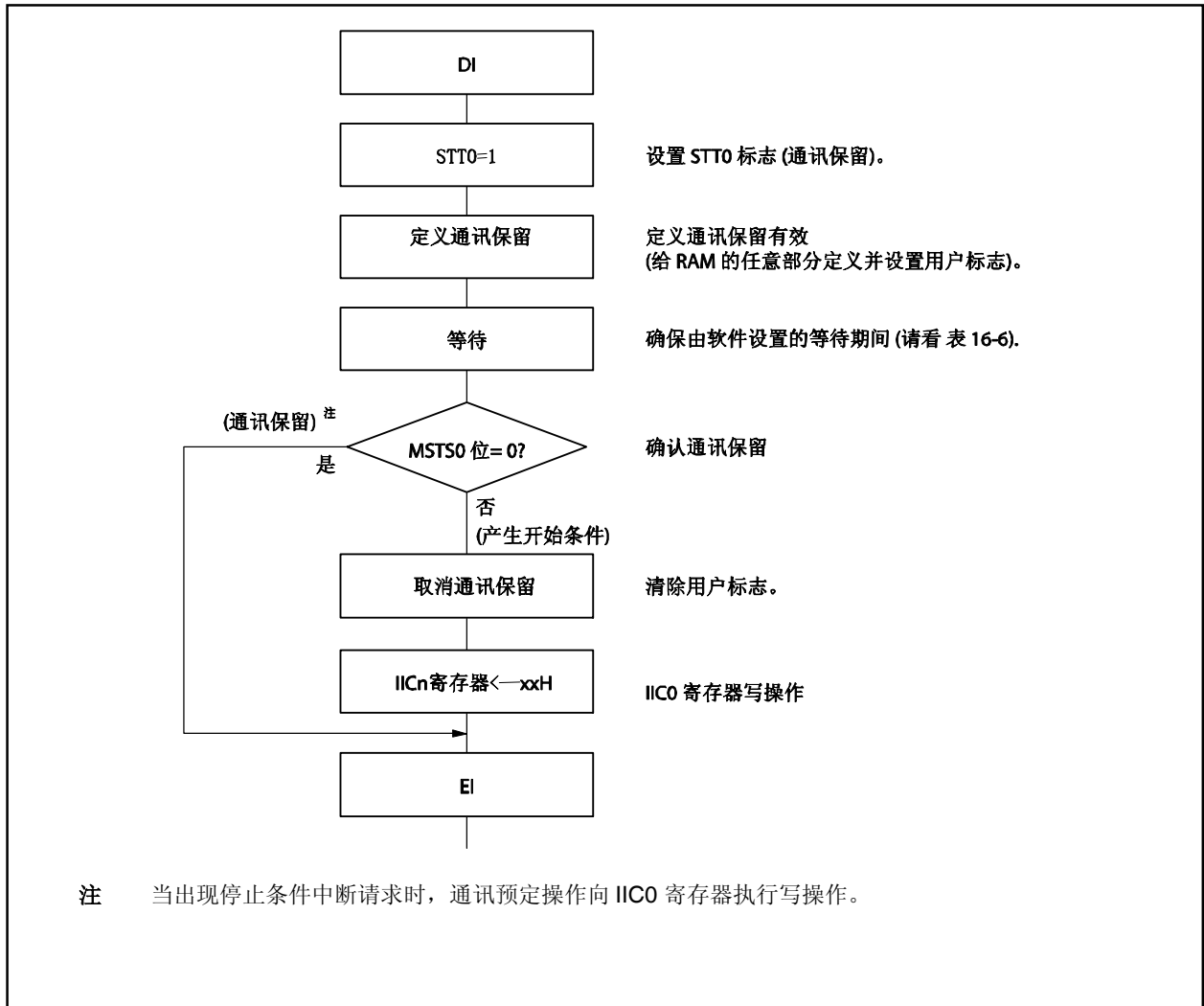
通讯预定通过以下时序接收。在 IICS0.STD0 位设为 1 后，在检测到停止条件前可以通过设定 IICC0.STT0 位设置通讯预定。

图 16-13. 接收通讯预定的时序



通讯预定流程图如下所示。

图 16-14. 通讯预定流程图



16.13.2 当通讯预定功能被禁用时 (IICF0.IICRSV0 位 = 1)

在当总线通讯过程中的通讯中没有使用总线时设置 IICC0.STT0 位的情况下，此请求被拒绝而不生成开始条件。在不使用总线时包括以下两个状态。

- 当仲裁造成主设备或从设备都不运行时
- 当接收到扩展代码且从设备运行禁止时 (当 IICC0.LREL0 位设为 1 时没有返回 \overline{ACK} 而总线释放)

要确认是否生成开始条件或拒绝请求，检查 IICF0.STCF0 标记。表 16-7 中显示的时间是必须的，直到在设置 STT0 位= 1 后设定 STCF0 标记。因此通过软件确保软件。

表 16-7. 等待周期

<R>

CL01	CL00	选择的时钟	等待周期
0	0	$f_{xx}/2$	10 个时钟
0	1	$f_{xx}/2$	10 个时钟
1	0	f_{xx}	5 个时钟
1	1	$f_{xx}/3$	15 个时钟

16.14 注意事项

- (1) 当 IICF0.STCEN0 位 = 0 时
在启用 I²C0 后，通讯状态 (IICF0.IICBSY0 位 = 1) 立即被识别而不论实际的总线状态。要在没有检测到停止条件的状态下执行主设备通讯，在开始主设备通讯前生成停止条件然后释放总线。
通过以下顺序生成停止条件：
 - <1> 设置 IICCL0 寄存器。
 - <2> 设置 IICC0.IICE0 位。
 - <3> 设置 IICC0.SPT0 位。
- (2) 当 IICF0.STCEN0 位 = 1 时
在启用 I²C0 后，总线释放状态 (IICBSY0 位 = 0) 立即被识别而不论实际的总线状态。要生成第一个开始条件 (IICC0.STT0 位 = 1)，必须确认总线已经释放从而避免干扰其他通讯。
- (3) 当 V850ES/KE2 的 IICC0.IICE0 位设为 1 且与其他设备的通讯在进行中时，可以根据通讯线的状态检测开始条件。确保在 SCL0 和 SDA0 线设为高电平时将 IICC0.IICE0 位设为 1。
- (4) 在启动运行 (IICC0.IICE0 位 = 1) 前通过 IICCL0 和 IICX0 寄存器确定运行的时钟频率。要改变运行的时钟频率，立即将 IICC0.IICE0 位清零。
- (5) 在将 IICC0.STT0 和 IICC0.SPT0 设为 1 后，必须首先将两者清零才能复位。
- (6) 如果已经预定了通讯，将 IICC0.SPIE0 位设为 1，从而通过检测停止条件而生成中断请求。在生成中断请求后，通过向 I²C0 写入通讯数据可释放等待状态，然后发送开始。如果检测到一个停止条件不产生一个中断，由于没有生成中断请求，发送在等待状态中将会停止。但是没有必要将软件的 SPIE0 位设为 1 以检测 IICS0.MSTS0 位。

16.15 通讯操作

以下通过流程图显示了三个运行步骤。

(1) 单主系统中的主设备运行

以下流程图所示的单主系统使用 V850ES/KE2 作为主设备。

流程图大致分为初始设置和通讯处理。初始设置在启动时执行。如果要求与从设备通讯，准备好通讯然后执行通讯处理。

(2) 多主系统中的主设备运行

在 IIC 总线多主系统中，当总线参与通讯时无法通过 IIC 总线的规格判定总线是处于释放或使用状态的。在这里，当数据和时钟在某一时期（1 帧）处于高电平时，则 V850ES/KE2 参与通讯而总线处于释放状态。

此流程图大致分为初始设置、通讯等待和通讯处理。在此省略了 V850ES/KE2 在仲裁中失败而被指定为从设备时的处理。在启动时处理初始设置以参加通讯。然后在等待作为主设备的通讯请求或等待指定为从设备。

实际的通讯在通讯处理中进行，其支持与从设备的发送和接收以及与其他主设备的仲裁。

(3) 从设备运行

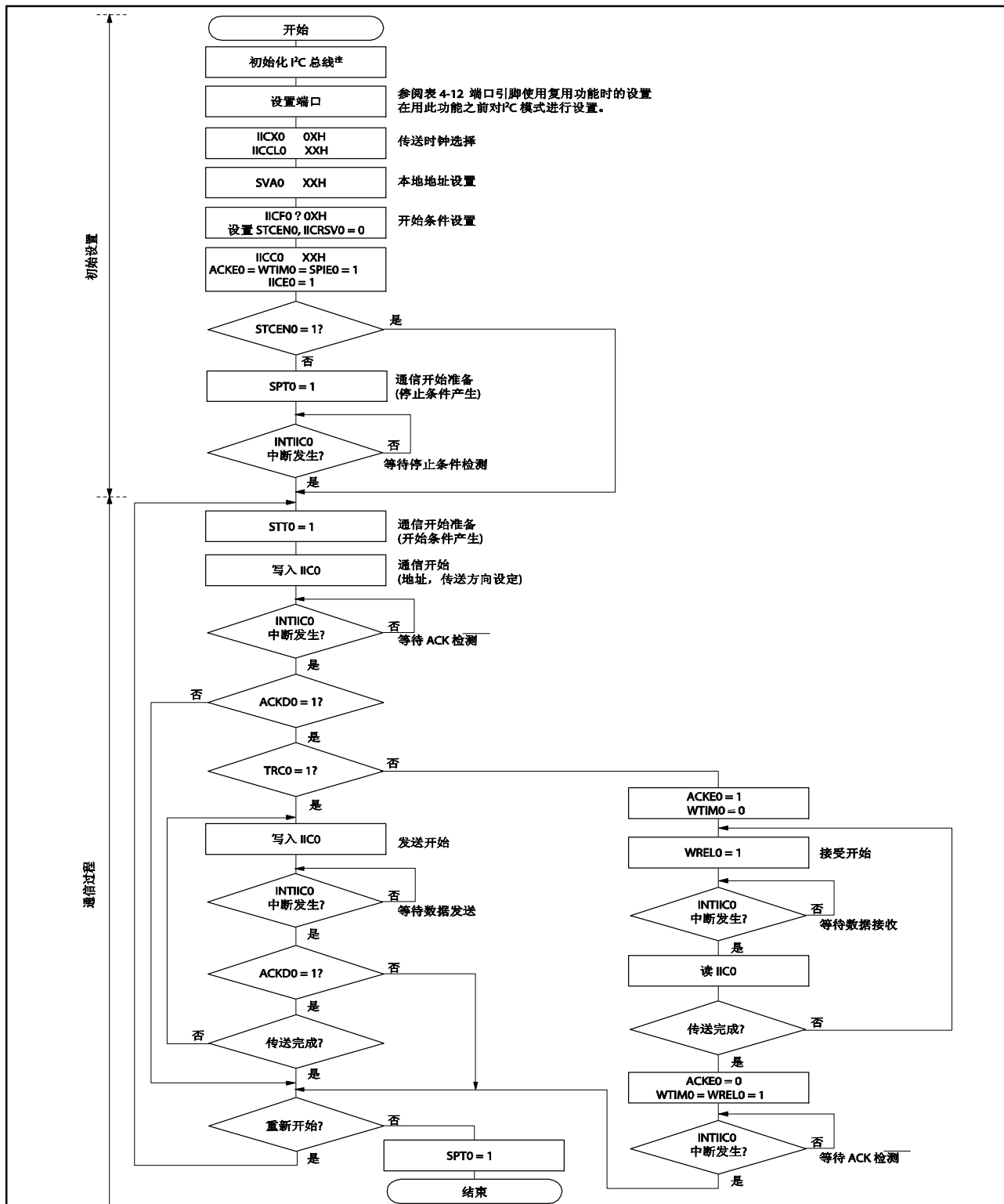
以下显示的是 V850ES/KE2 用于 IIC 总线的从设备的实例。

当作为从设备使用时，运行是通过一个中断启动的。在启动时执行初始设置，然后等待 INTIIC0 中断出现（通讯等待）。当 INTIIC0 中断出现时，会对对通讯状态进行判定，判定结果会被以标记的形式传送到主处理程序中。

通过检查标记，执行必要的通讯处理。

16.15.1 单主系统中的主设备运行

图 16-15. 单主系统中的主设备运行



注 请根据通讯中产品的规格释放 IIC0 总线 (SCL0, SDA0 引脚 = 高电平)。

例如, 当 EEPROM™ 输出一个低电平到 SDA0 引脚时, 设置 SCL0 引脚到输出端口, 并从该输出端口输出时钟脉冲直到 SDA0 引脚保持高电平。

备注 对于传送和接收格式, 必须符合通讯中产品的规格。

16.15.2 多主系统中的主设备运行

图 16-16. 多主系统中的主设备运行 (1/3)

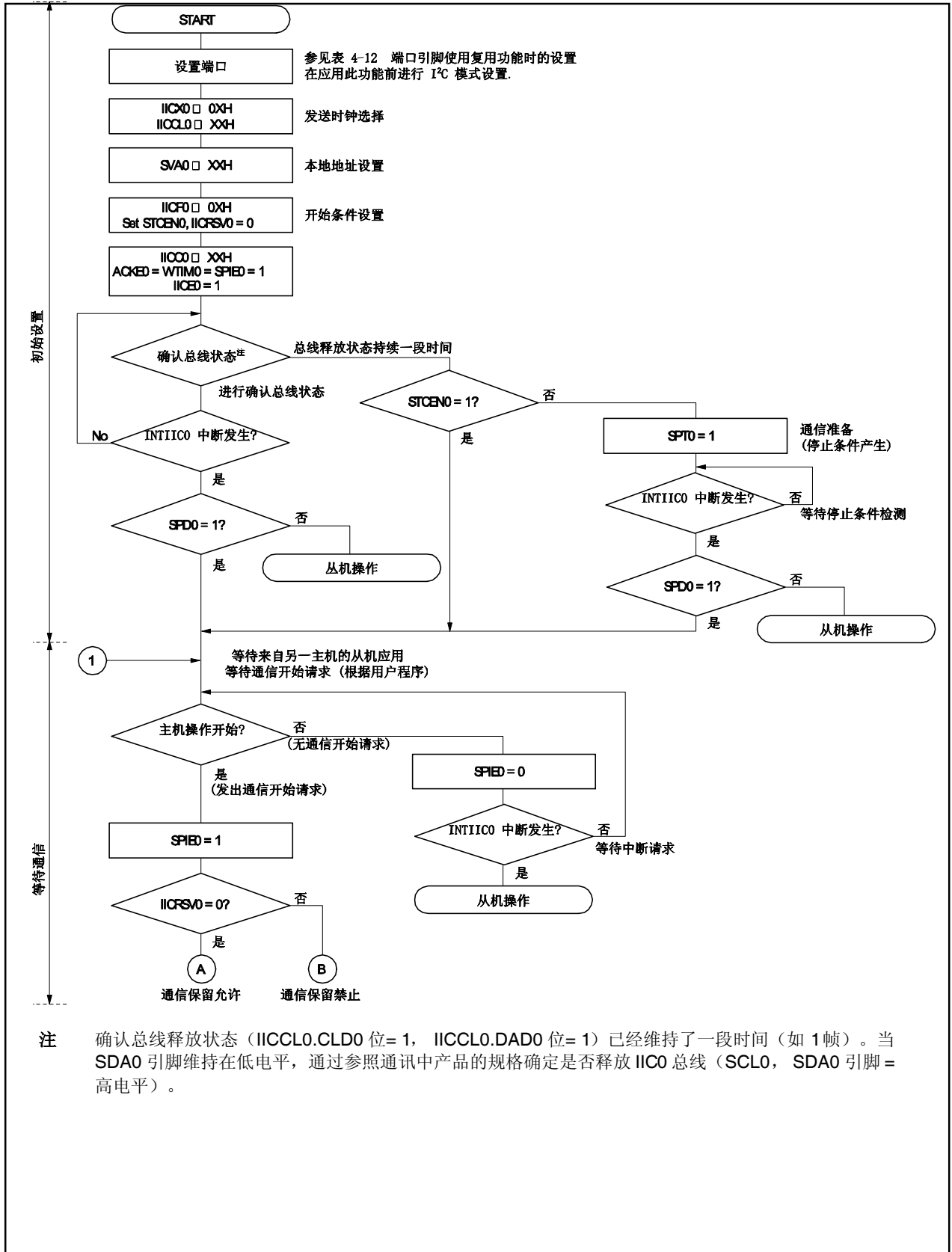


图 16-16. 多主系统中的主设备运行 (2/3)

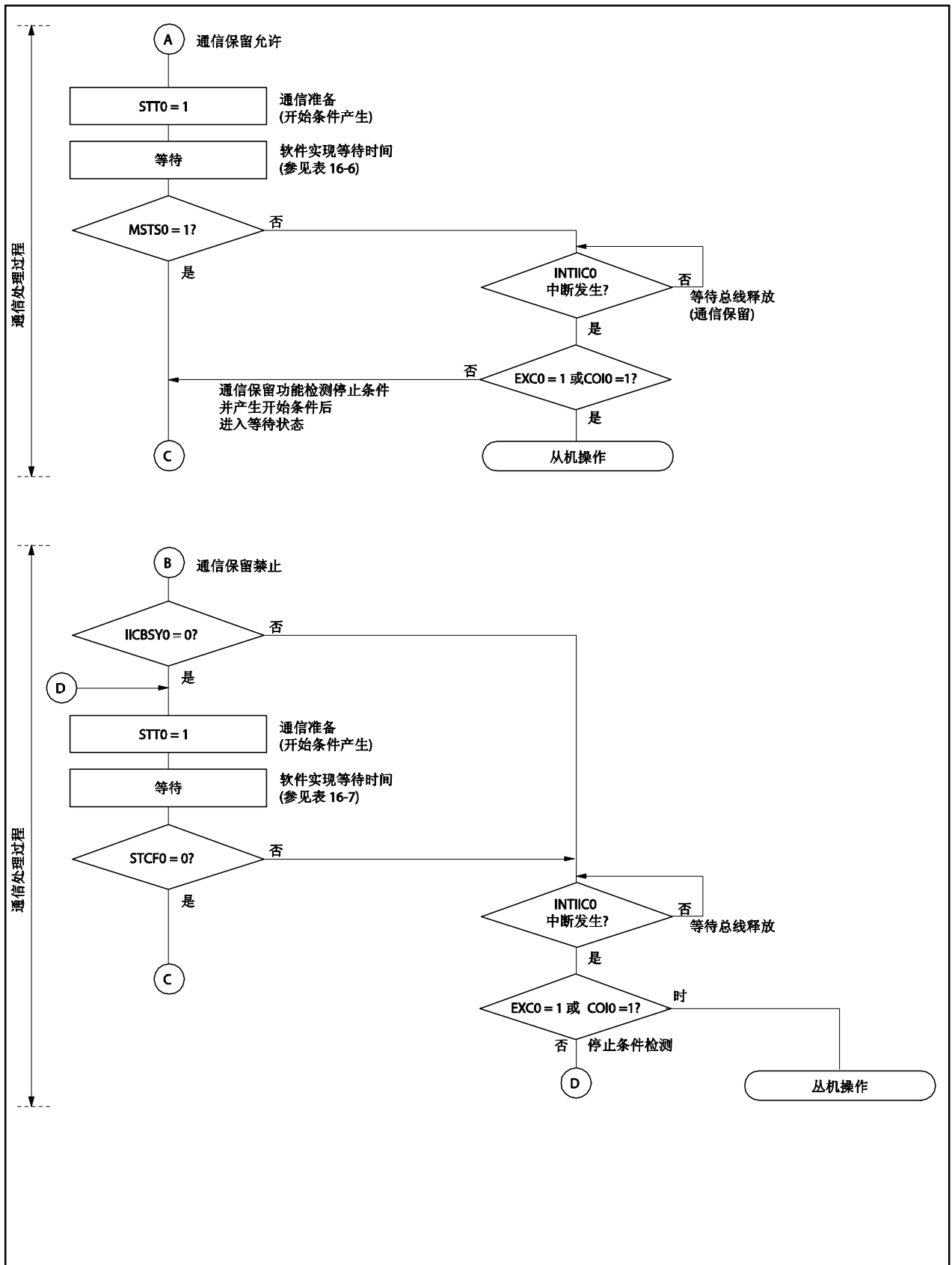
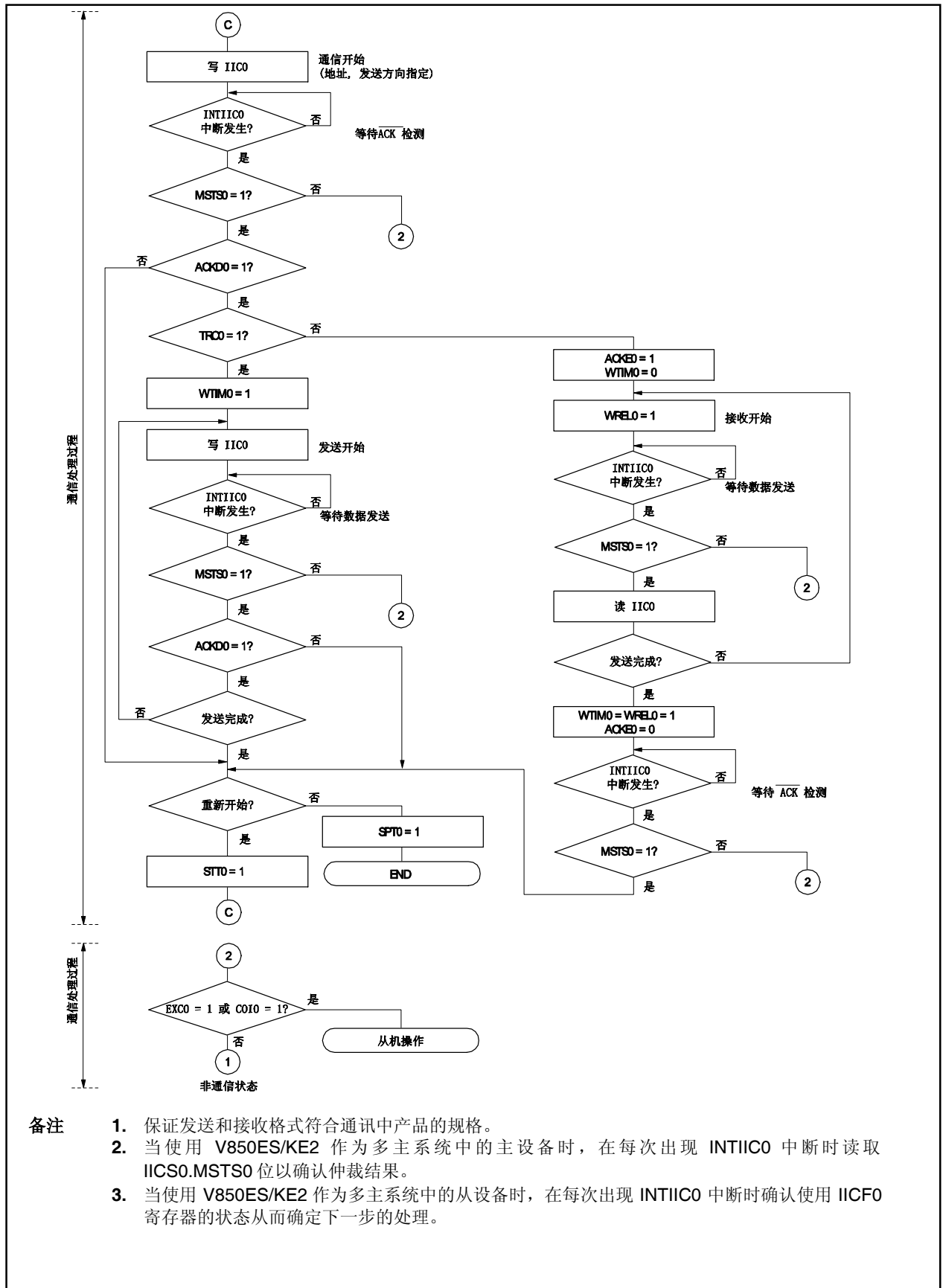


图 16-16. 多主系统中的主设备运行 (3/3)



- 备注
1. 保证发送和接收格式符合通讯中产品的规格。
 2. 当使用 V850ES/KE2 作为多主系统中的主设备时, 在每次出现 INTIIC0 中断时读取 IICS0.MSTSD0 位以确认仲裁结果。
 3. 当使用 V850ES/KE2 作为多主系统中的从设备时, 在每次出现 INTIIC0 中断时确认使用 IICF0 寄存器的状态从而确定下一步的处理。

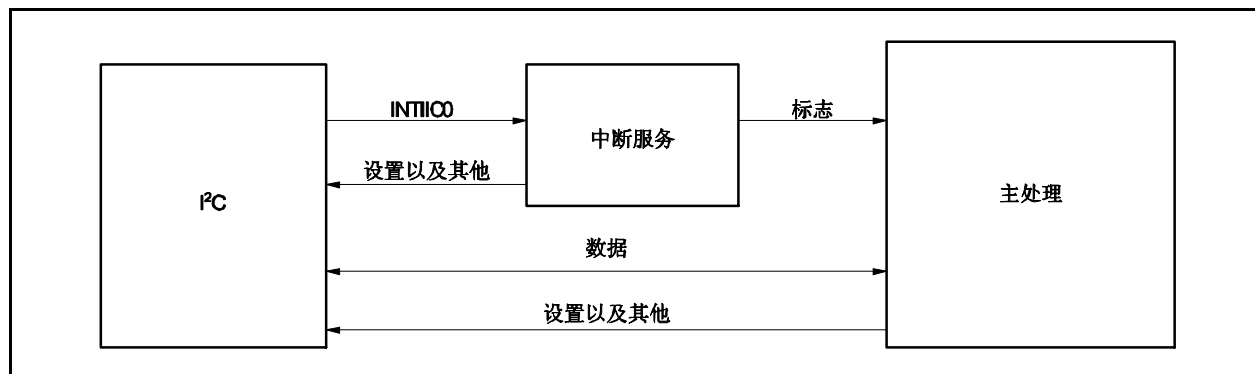
16.15.3 从设备运行

以下显示从设备运行的处理步骤。

从设备的运行基本上是事件驱动的。因此有必要通过 INTIIC0 中断进行处理（处理要求对运行状态作重大改变，如在通讯中检测停止条件）。

以下描述假设数据通讯不支持扩展代码。同时还假设 INTIIC0 中断服务只进行状态改变处理，在主处理的过程中进行实际数据通讯。

图 16-17：从设备运行过程中的软件概览



因此，准备了以下三个标记，因而可以通过将这些标记替代 INTIIC0 信号传送到主处理程序，从而完成数据发送处理。

(1) 通讯模式标记

此标记显示以下通讯状态。

清除模式：数据通讯不在进行中

通讯模式：数据通讯进行中（有效地址检测，停止条件检测，未检测到来自主设备的 $\overline{\text{ACK}}$ ，地址不相等）

(2) 准备就绪标记

此标记显示启用了数据通讯。这与正常数据发送过程中的 INTIIC0 中断状态相同。此标记在中断处理块中设置，在主处理块中清除。首次处理数据的准备就绪标记在中断处理块中设置，因此首次数据不经过清除处理就发送。（地址相等被认为是对下一数据的请求。）

(3) 通讯方向标记

此标记显示通讯的方向，与 IICS0.TRC0 位的值相同。

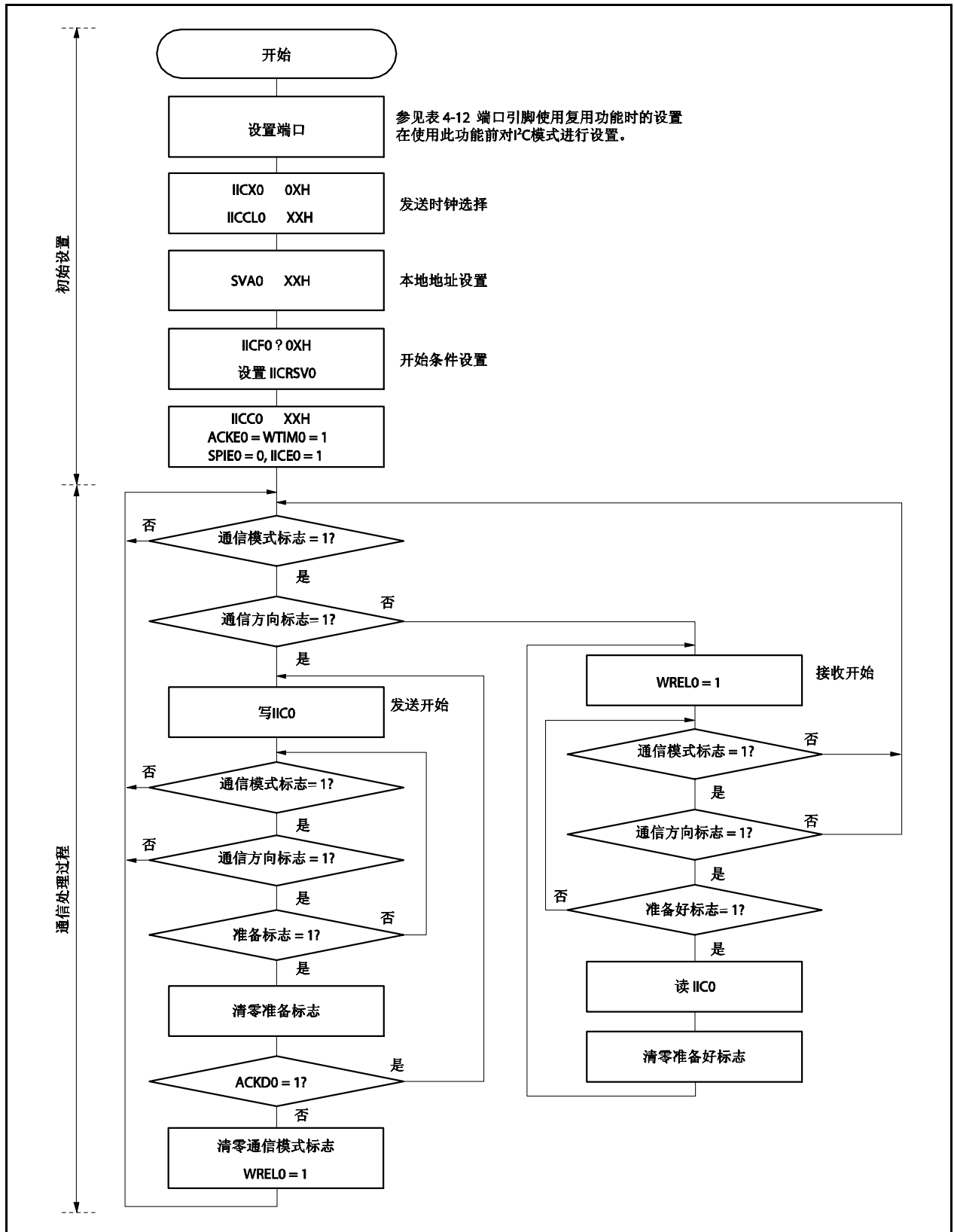
以下显示从设备运行过程中主处理块的运行。

开启 I^2C0 ，等待通讯启用状态。在通讯启用后，通过使用通讯模式标记和准备就绪状态（停止条件和开始条件的处理由中断进行，条件由标记确认）进行发送。

对于发送，重复发送操作做到主设备停止返回 $\overline{\text{ACK}}$ 。当主设备停止返回 $\overline{\text{ACK}}$ 时，发送完成。

对于接收，接收要求数量的数据，在发送完成后不立即返回 $\overline{\text{ACK}}$ 请求下一数据。发送完成后，主设备生成停止条件或重新启动条件。这样会退出通讯。

图 16-18. 从设备运行流程图 (1)

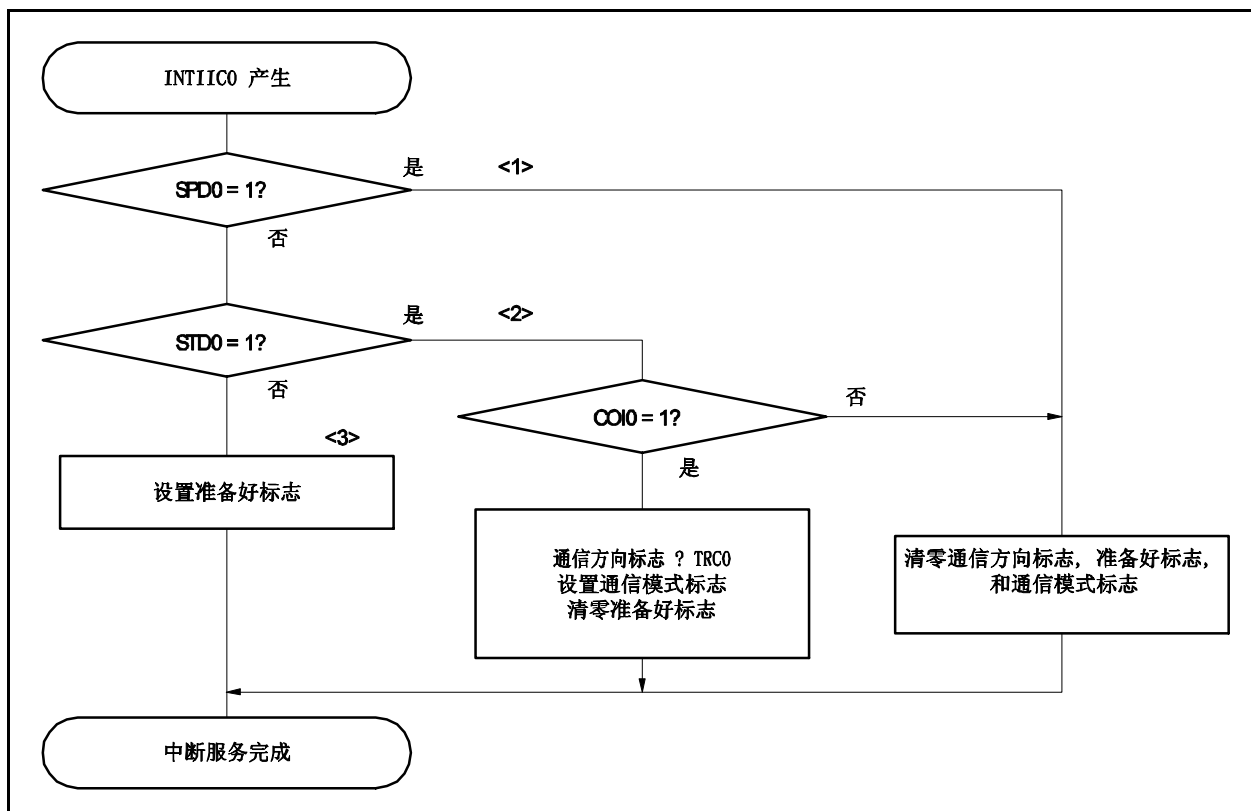


以下是通过 INTIIC0 中断处理从设备的一个实例（这里假设不使用扩展代码）。在 INTIIC0 中断的过程中，确认状态并执行以下步骤。

- <1> 当检测到停止条件时，通讯终止。
- <2> 当检测到开始条件时，地址确认。如果地址不相等，通讯终止。如果地址相等，则设置通讯模式并释放等待，运行返回冲突（准备就绪标记被清除）。
- <3> 对于数据发送/接收，当设置准备就绪标志后，运行从中断中返回而 IIC0 总线保持等待状态。

备注 以上的<1>到<3>条对应图 16-19 从设备运行流程图（2）中的<1>到<3>条。

图 16-19. 从设备运行流程图（2）



16.16 数据通讯的时序

当使用 IIC 总线模式时，主设备通过串行总线生成一个地址选择数个从设备之一作为通讯伙伴。

在输出从设备地址后，主设备传送指定数据发送方向的 **IICS0.TRC0** 位，然后开始与从设备的串行通讯。

IIC0 寄存器的移位操作与串行时钟的下降沿同步（**SCL0** 引脚）。发送的数据被传送到 **SO** 锁中并通过 **SDA0** 引脚输出（最高有效位先输出）。

通过 **SDA0** 引脚输入的数据在 **SCL0** 引脚的上升沿被 **IIC0** 寄存器捕获。

数据通讯的时序如下图所示。

图 16-20. 主设备到从设备通讯实例
(当为主设备和从设备选择 9 时钟等待的情况下) (1/3)

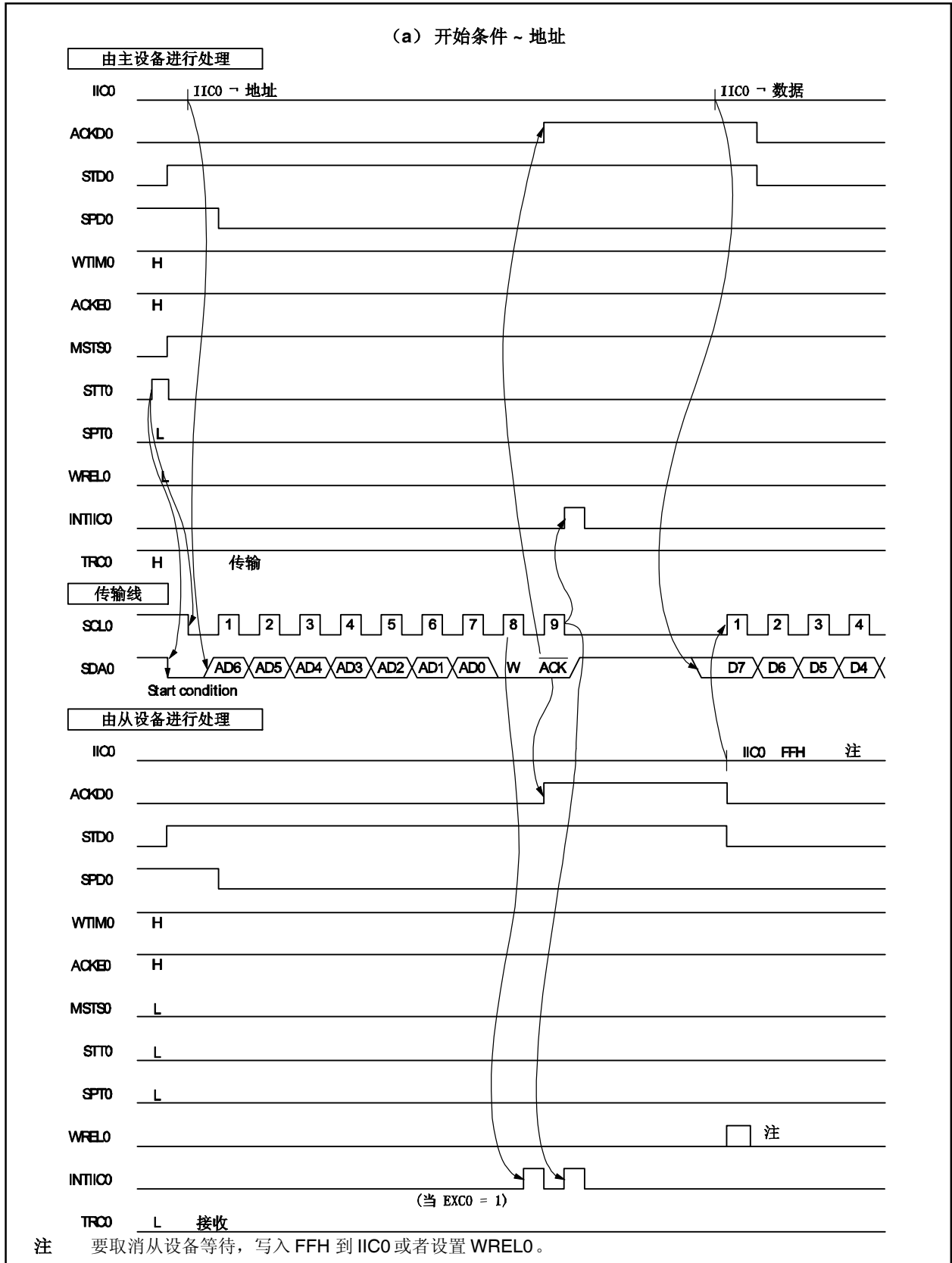


图 16-20. 主设备到从设备通讯实例
(当为主设备和从设备选择 9 时钟等待的情况下) (2/3)

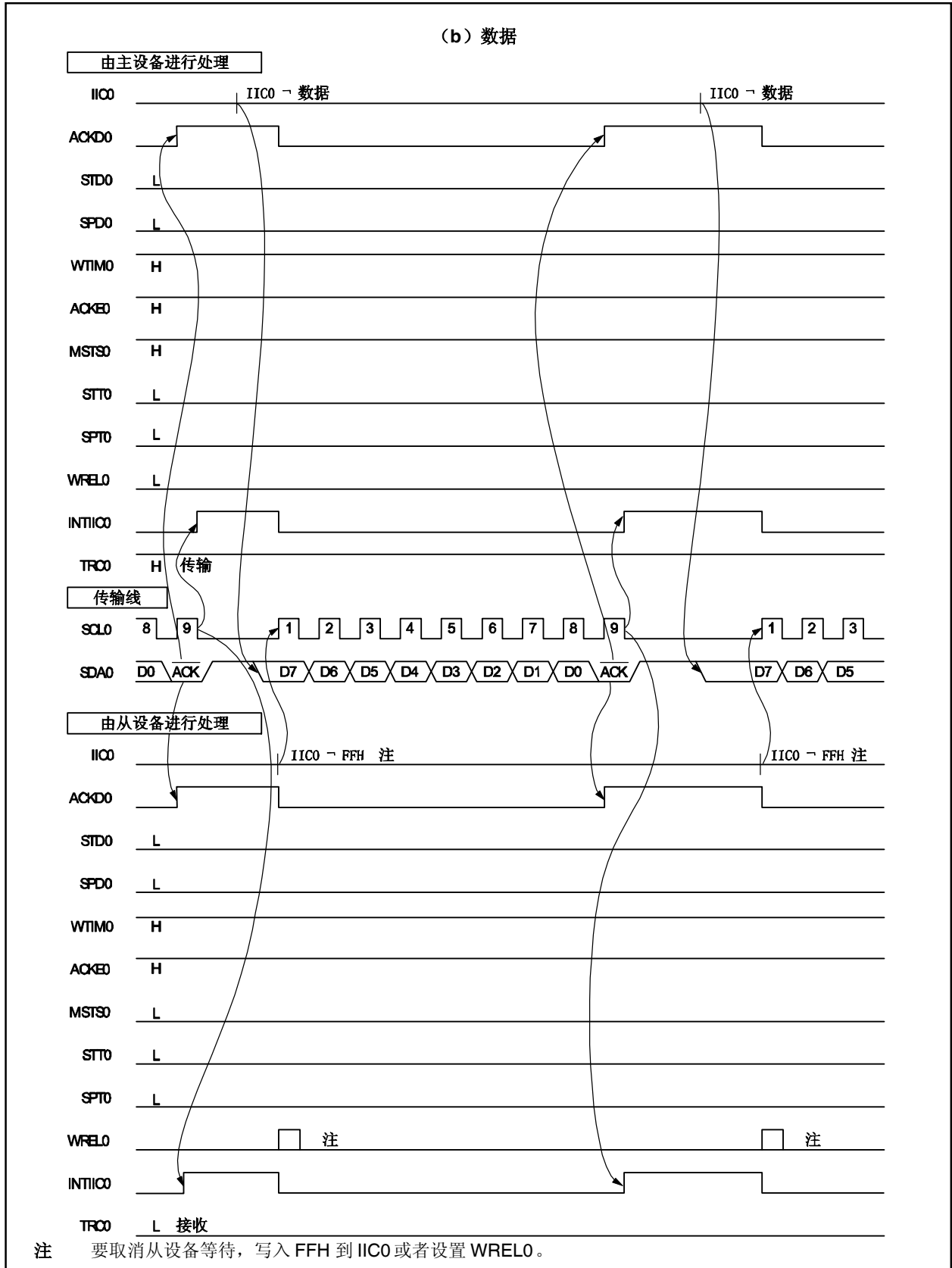


图 16-20. 主设备到从设备通讯实例
(当为主设备和从设备选择 9 时钟等待的情况下) (3/3)

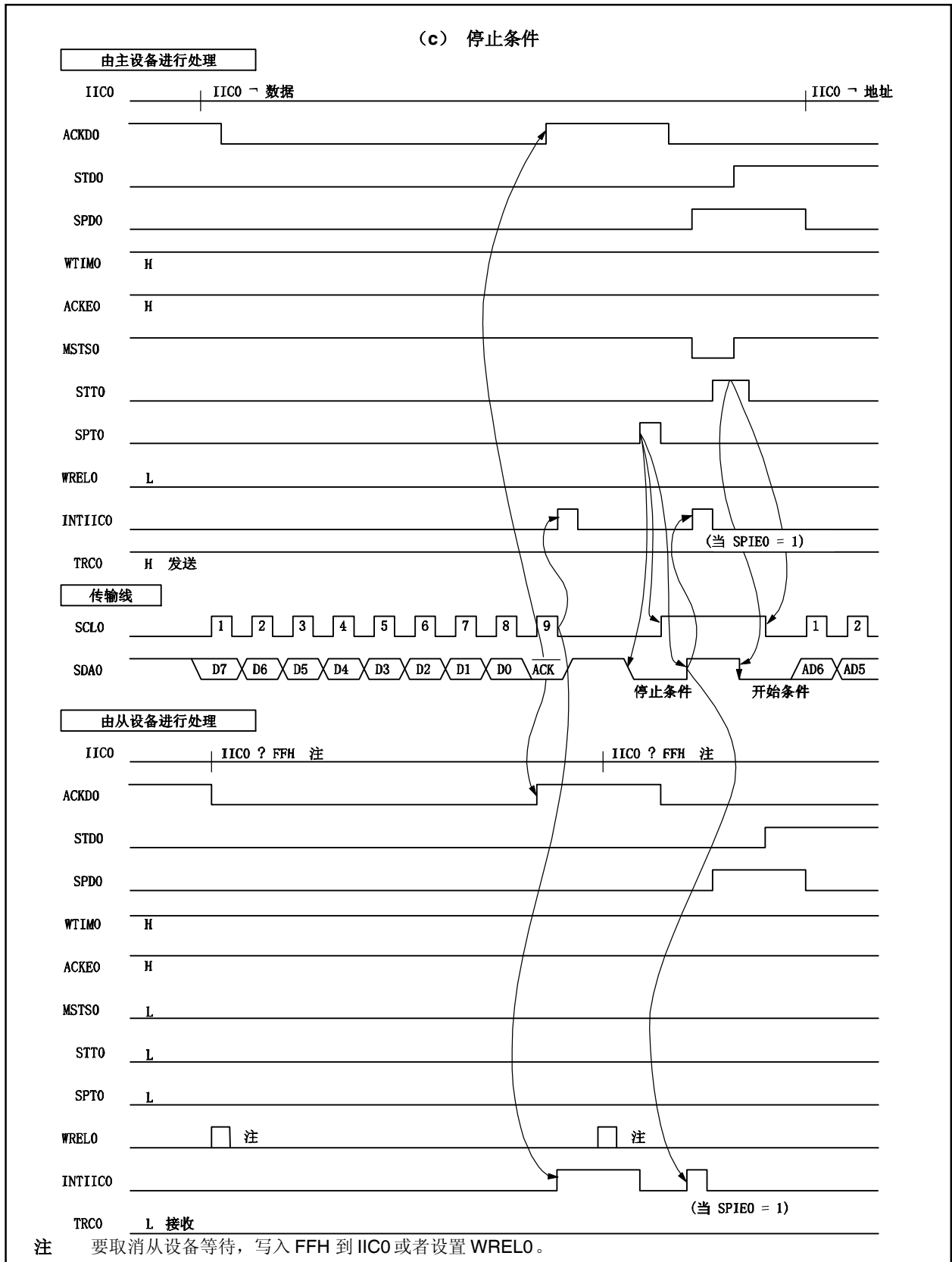


图 16-21. 从设备到主设备通讯实例
 (当为主设备选择 8 时钟等待和从设备选择 9 时钟等待的情况下) (1/3)

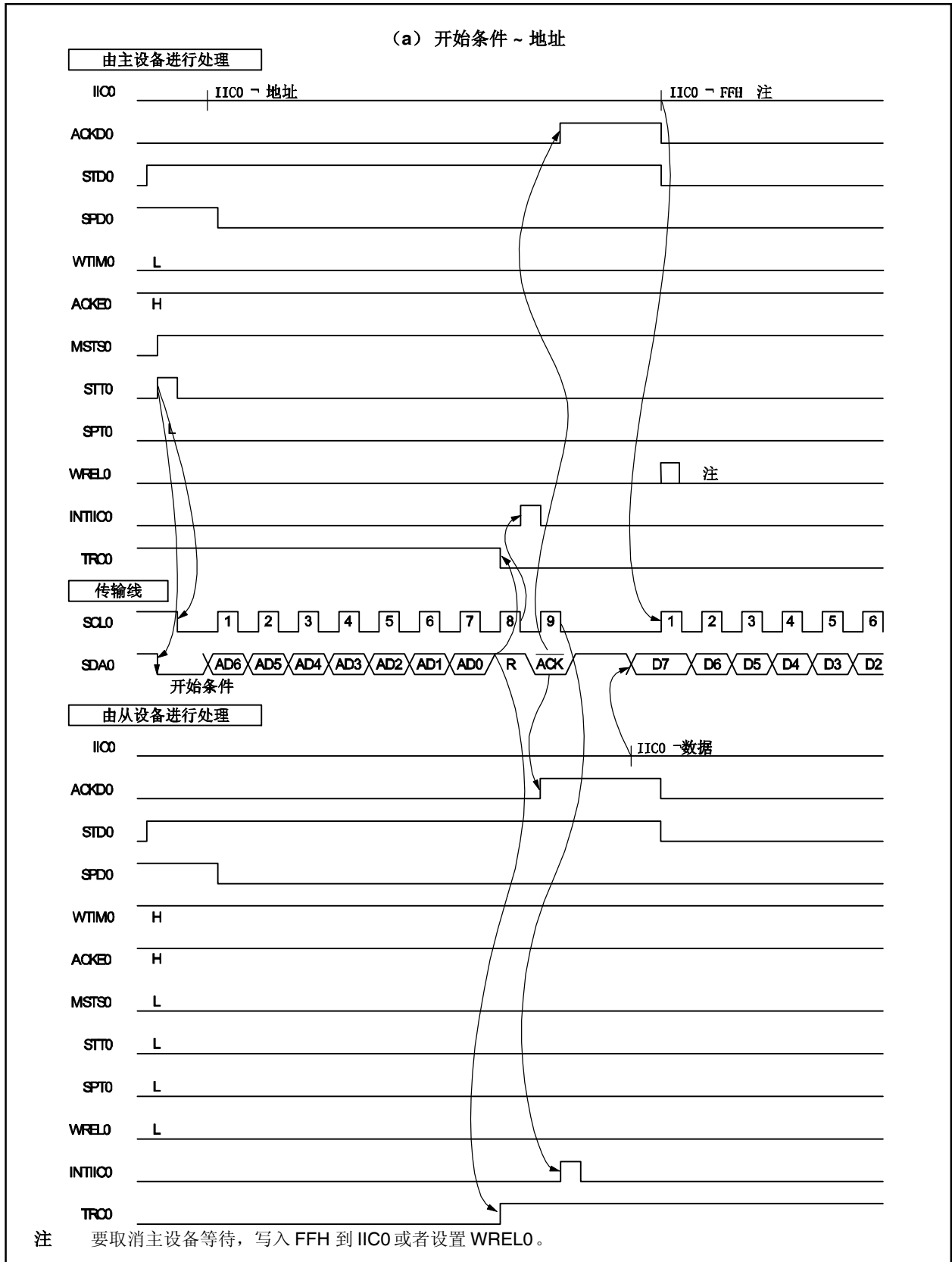


图 16-21. 从设备到主设备通讯实例
(当为主设备选择 8 时钟等待和从设备选择 9 时钟等待的情况下) (2/3)

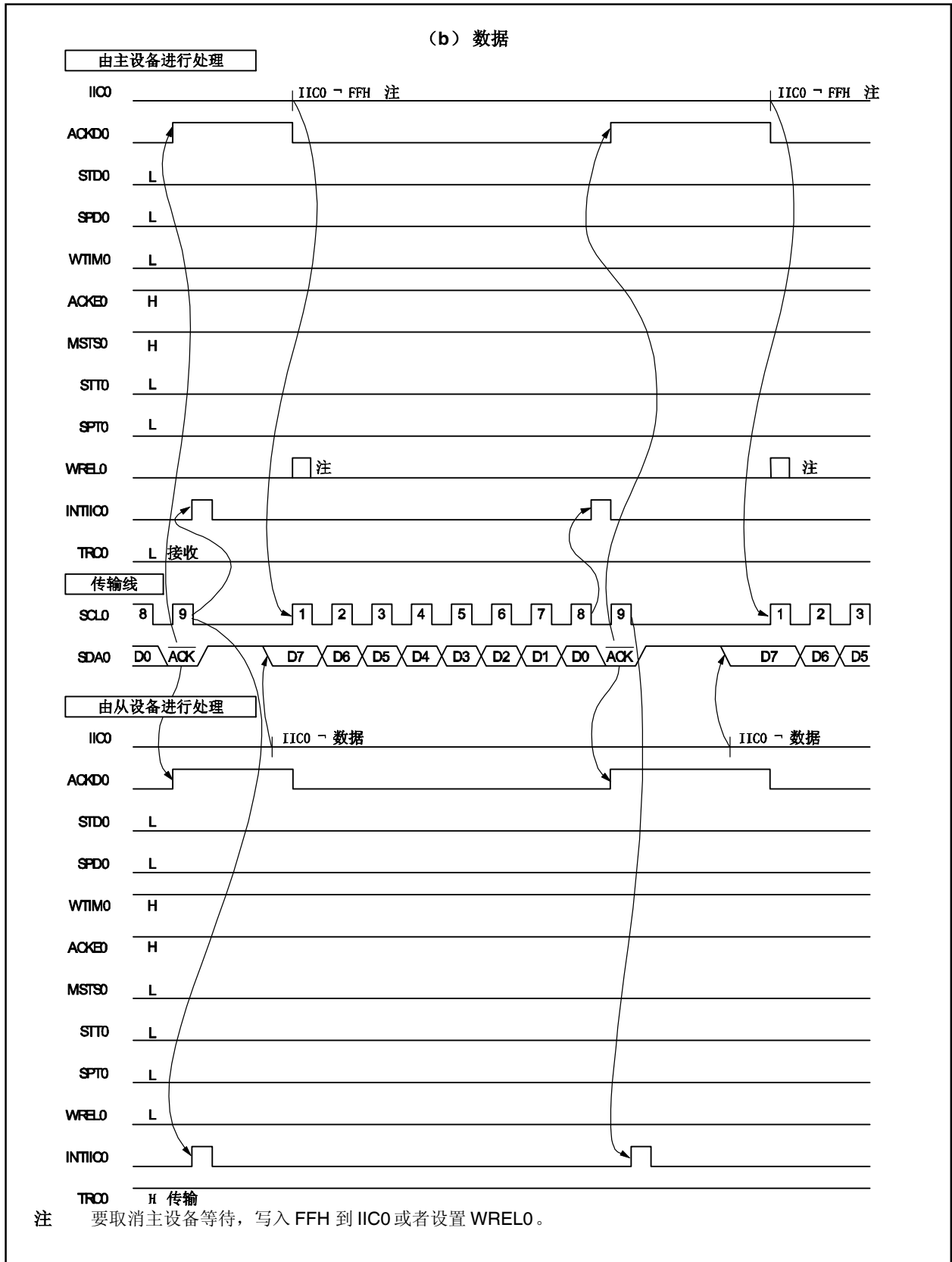
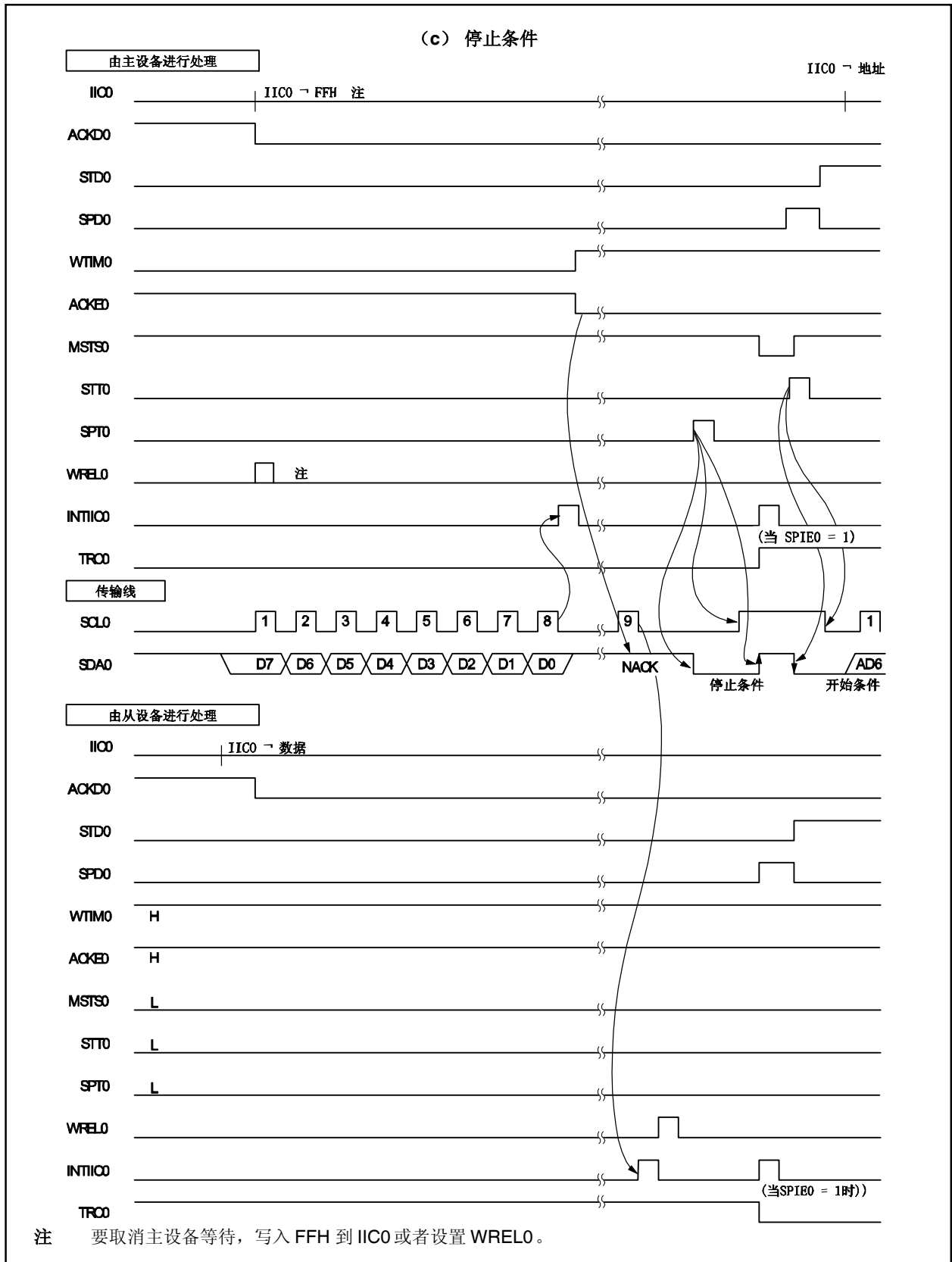


图 16-21. 从设备到主设备通讯实例
(当为主设备选择 8 时钟等待和从设备选择 9 时钟等待的情况下) (3/3)



第十七章 中断/异常处理功能

17.1 概述

V850ES/KE2 为中断服务器提供一个专门的可处理总共 35 个中断请求的中断控制器（INTC）。

中断是指不受程序执行影响的事件发生，异常是指依赖于程序执行的事件发生。

V850ES/KE2 可以处理由片上外围硬件和外部事件源引起的中断请求信号。此外，异常处理可以通过 TRAP 指令（软件异常）或者产生一个异常事件（例如：存在非法操作码）（异常陷阱）开始。

17.1.1 特点

中断源			V850ES/KE2	
中断功能	非屏蔽中断	外部	1 个通道（NMI 引脚）	
		内部	2 个通道（WDT1, WDT2）	
	屏蔽中断	外部	8 个通道（均为边沿检测中断）	
		内部	WDT1	1 个通道
			TMP	3 个通道
			TM0	2 个通道
			TMH	2 个通道
			TM5	2 个通道
			WT	2 个通道
			BRG	1 个通道
			UART	6 个通道
			CSI0	2 个通道
			IIC	1 个通道
			KR	1 个通道
AD	1 个通道			
全部	24 个通道			
异常功能	软件异常		16 个通道（TRAP00H~TRAP0FH）	
			16 个通道（TRAP10H~TRAP1FH）	
	异常陷阱		2 个通道（ILGOP/DBG0）	

表 17-1 列出了中断/异常源。

表 17-1. 中断源列表 (1/2)

类型	级别	缺省优先级	名称	触发	中断源	异常代码	处理程序地址	恢复 PC	中断控制寄存器
复位	中断	-		RESET 引脚输入	引脚	0000H	00000000H	未定义	-
				来自 WDT1, WDT2 的内部复位输入	WDT1 WDT2				
非屏蔽	中断	-	NMI	NMI 引脚有效边沿输入	引脚	0010H	00000010H	下一个 PC	-
		-	INTWDT1	WDT1 溢出 (选择非屏蔽中断时)	WDT1	0020H	00000020H	注 1	-
		-	INTWDT2	WDT2 溢出 (选择非屏蔽中断时)	WDT2	0030H	00000030H	注 1	-
软件异常	异常	-	TRAP0n ^{‡2}	TRAP 指令	-	004nH ^{‡2}	00000040H	下一个 PC	-
		-	TRAP1n ^{‡2}	TRAP 指令	-	005nH ^{‡2}	00000050H	下一个 PC	-
异常陷阱	异常	-	ILGOP/ DBG0	非法操作码/ DBTRAP 指令	-	0060H	00000060H	下一个 PC	-
可屏蔽	中断	0	INTWDTM1	WDT1 溢出 (选择间隔计数器时)	WDT1	0080H	00000080H	下一个 PC	WDT1IC
		1	INTP0	INTP0 引脚有效边沿输入	引脚	0090H	00000090H	下一个 PC	PIC0
		2	INTP1	INTP1 引脚有效边沿输入	引脚	00A0H	000000A0H	下一个 PC	PIC1
		3	INTP2	INTP2 引脚有效边沿输入	引脚	00B0H	000000B0H	下一个 PC	PIC2
		4	INTP3	INTP3 引脚有效边沿输入	引脚	00C0H	000000C0H	下一个 PC	PIC3
		5	INTP4	INTP4 引脚有效边沿输入	引脚	00D0H	000000D0H	下一个 PC	PIC4
		6	INTP5	INTP5 引脚有效边沿输入	引脚	00E0H	000000E0H	下一个 PC	PIC5
		7	INTP6	INTP6 引脚有效边沿输入	引脚	00F0H	000000F0H	下一个 PC	PIC6
		10	INTTM010	TM01 和 CR010 比较匹配	TM01	0120H	00000120H	下一个 PC	TM01C10
		11	INTTM011	TM01 和 CR011 比较匹配	TM01	0130H	00000130H	下一个 PC	TM01C11
		12	INTTM50	TM50 和 CR50 比较匹配	TM50	0140H	00000140H	下一个 PC	TM51C0
		13	INTTM51	TM51 和 CR51 比较匹配	TM51	0150H	00000150H	下一个 PC	TM51C1
		14	INTCSI00	CSI00 传送完成	CSI00	0160H	00000160H	下一个 PC	CSI0IC0
		15	INTCSI01	CSI01 传送完成	CSI01	0170H	00000170H	下一个 PC	CSI0IC1
		16	INTSRE0	UART0 接收发生错误	UART0	0180H	00000180H	下一个 PC	SREIC0
		17	INTSR0	UART0 接收完成	UART0	0190H	00000190H	下一个 PC	SRIC0
		18	INTST0	UART0 发送完成	UART0	01A0H	000001AH	下一个 PC	STIC0
		19	INTSRE1	UART1 接收发生错误	UART1	01B0H	000001B0H	下一个 PC	SREIC1
		20	INTSR1	UART1 接收完成	UART1	01C0H	000001C0H	下一个 PC	SRIC1
		21	INTST1	UART1 发送完成	UART1	01D0H	000001D0H	下一个 PC	STIC1

- 注
1. INTWDT1 和 INTWDT2 的恢复, 参见 17.10 注意事项。
 2. n 的取值范围是 0 到 FH。

表 17-1. 中断源列表 (2/2)

类型	级别	缺省优先级	名称	触发	中断源	异常代码	处理程序地址	恢复 PC	中断控制寄存器
可屏蔽	中断	22	INTTMH0	TMH0 和 CMP00/CMP01 比较匹配	TMH0	01E0H	000001E0H	下一个 PC	TMHIC0
		23	INTTMH1	TMH1 和 CMP00/CMP01 比较匹配	TMH1	01F0H	000001F0H	下一个 PC	TMHIC1
		25	INTIIC0	I ² C0 传送完成	I ² C0	0210H	00000210H	下一个 PC	IICIC0
		26	INTAD	A/D 转换完成	A/D	0220H	00000220H	下一个 PC	ADIC
		27	INTKR	按键返回中断	KR	0230H	00000230H	下一个 PC	KRIC
		28	INTWTI	钟表定时器间隔	WT	0240H	00000240H	下一个 PC	WTIIC
		29	INTWT	钟表定时器参考时间	WT	0250H	00000250H	下一个 PC	WTIC
		30	INTBRG	预分频器 3 的 8 位计数器与 PRSCM 比较匹配	Prescaler 3	0260H	00000260H	下一个 PC	BRGIC
		45	INTP7	INTP7 引脚有效边沿输入	Pin	0390H	00000390H	下一个 PC	PIC7
		46	INTTP0OV	TMP0 溢出	TMP	03A0H	000003A0H	下一个 PC	TP0OVIC
		47	INTTP0CC0	TMP0 捕获 0/比较 0 匹配	TMP	03B0H	000003B0H	下一个 PC	TP0CCIC0
		48	INTTP0CC1	TMP1 捕获 1/比较 1 匹配	TMP	03C0H	000003C0H	下一个 PC	TP0CCIC1

备注 1. 缺省优先级: 表示当两个或更多的可屏蔽中断请求同时发生时, 它们的有限顺序。最高值为 0。

非屏蔽中断的优先级顺序如下所示。

INTWDT2 > INTWDT1 > NMI

恢复 PC: 当中断服务产生时被保存到 EIPC, FEPC, 或是 DBPC 的程序计数器 (PC) 的值。但是, 当非屏蔽中断或者可屏蔽中断被确认, 且下列指令之一正在被执行时, 恢复 PC 不会成为下一个 PC (如果在中断执行期间确认中断, 则该操作停止并在中断服务结束后恢复)。

- 载入指令 (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
- 除法指令 (DIV, DIVH, DIVU, DIVHU)
- 预备, 处理指令 (仅限中断在堆栈指针更新之前产生的情况)

下一个 PC: 在中断/异常处理之后的开始处理的 PC 的值。

2. 当一个非法的操作代码异常发生时, 非法指令的执行地址通过保存的 PC 值减去 4 得出。

17.2 非屏蔽中断

非屏蔽中断请求会被无条件的确认，即使是在禁止中断的条件下（DI）。一个非屏蔽中断不会受到优先级的控制，并且相对于其它的中断请求而言具有绝对的优先权。

在 V850ES/KE2 中有下列 3 种非屏蔽中断请求信号。

- NMI 引脚输入（NMI）
- 非屏蔽中断请求信号可以通过看门狗定时器 1 溢出产生（INTWDT1）
- 非屏蔽中断请求信号可以通过看门狗定时器溢出产生（INTWDT2）

NMI 引脚的有效边沿可以选择以下四种方式：“上升沿”，“下降沿”，“双边沿”和“无边沿检测”。

当 WDTM1.WDTM14 位和 WDTM1.WDTM13 位设为 10 时，由看门狗定时器 1 溢出而产生的非屏蔽中断请求信号（INTWDT1）生效。

当 WDTM2.WDM21 位和 WDTM2.WDM20 位设为 01 时，由看门狗定时器 2 溢出而产生的非屏蔽中断请求信号（INTWDT2）生效。

如果两个或是更多的非屏蔽中断请求同时产生，具有较高优先级的中断首先生效，如下所示（具有较低优先级的中断请求信号将被忽略）。

INTWDT2 > INTWDT1 > NMI

在 NMI 过程中，产生了新的 NMI，INTWDT1，或 INTWDT2 请求信号，执行如下处理过程。

(1) 如果在非屏蔽中断正在进行的同时，发出一个新的 NMI 请求信号

新的 NMI 请求信号被保持，忽略 PSW.NP 位的值。而新的 NMI 请求信号会在当前的 NMI 运行完毕后被确认（在 RETI 指令执行后）。

(2) 如果在非屏蔽中断正在进行的同时，发出一个 INTWDT1 请求信号

当存在正在进行的 NMI 时，如果 NP 位的值是（1），则 INTWDT1 请求信号被保持。被保持的 INTWDT1 请求信号会在当前的 NMI 运行完毕后被确认（在 RETI 指令执行后）。

当存在正在进行的 NMI 时，如果 NP 位为（0），则立即执行新产生的 INTWDT1 请求信号（NMI 服务停止）。

(3) 如果在非屏蔽中断正在进行的同时，发出一个 INTWDT2 请求信号

新产生的 INTWDT2 请求信号将被执行，而无论 NP 值为多少（NMI 过程被执行）。

注意事项 由非屏蔽中断请求信号（INTWDT1，INTWDT2）执行的非屏蔽中断服务，参考 17.10 注意事项。

图 17-1. 非屏蔽中断请求信号确认操作 (1/2)

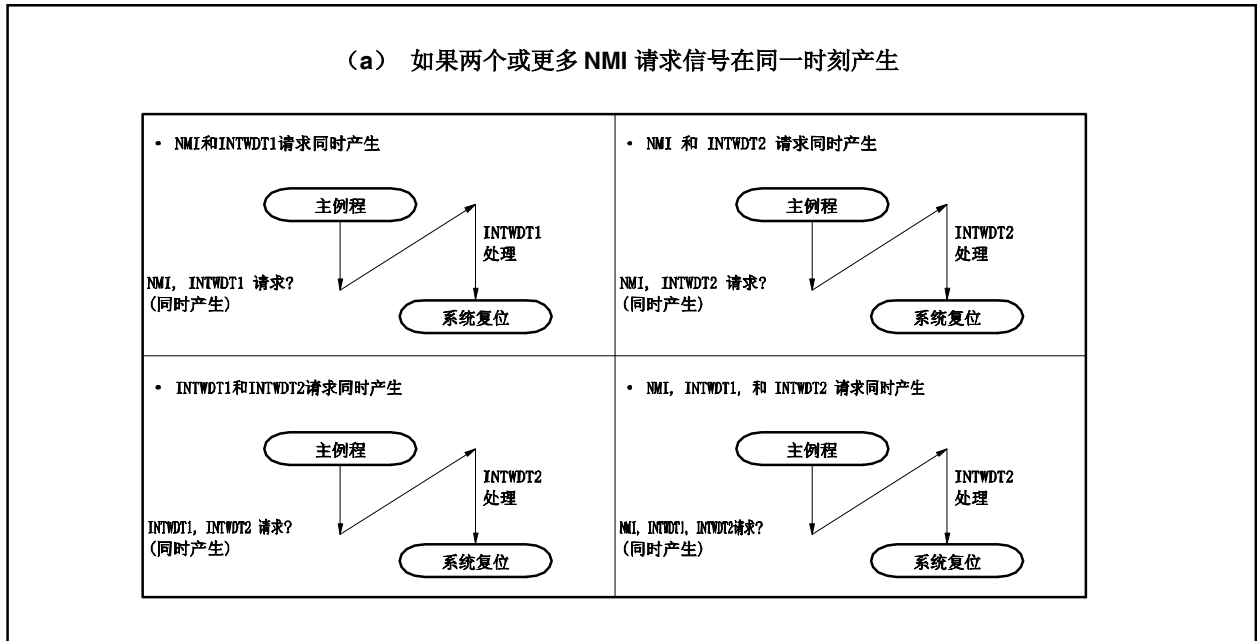
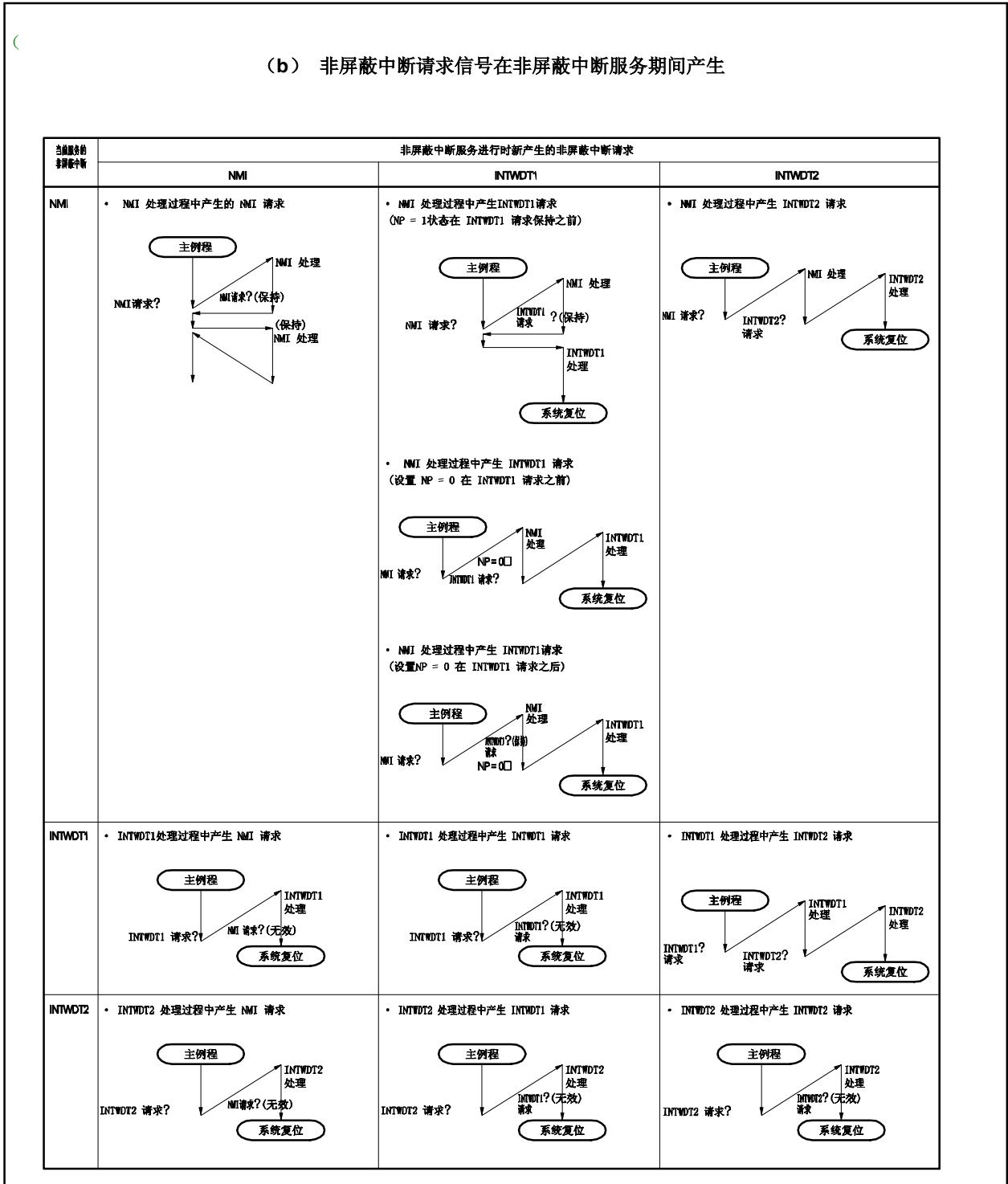


图 17-1. 非屏蔽中断请求信号确认操作 (2/2)

(b) 非屏蔽中断请求信号在非屏蔽中断服务期间产生



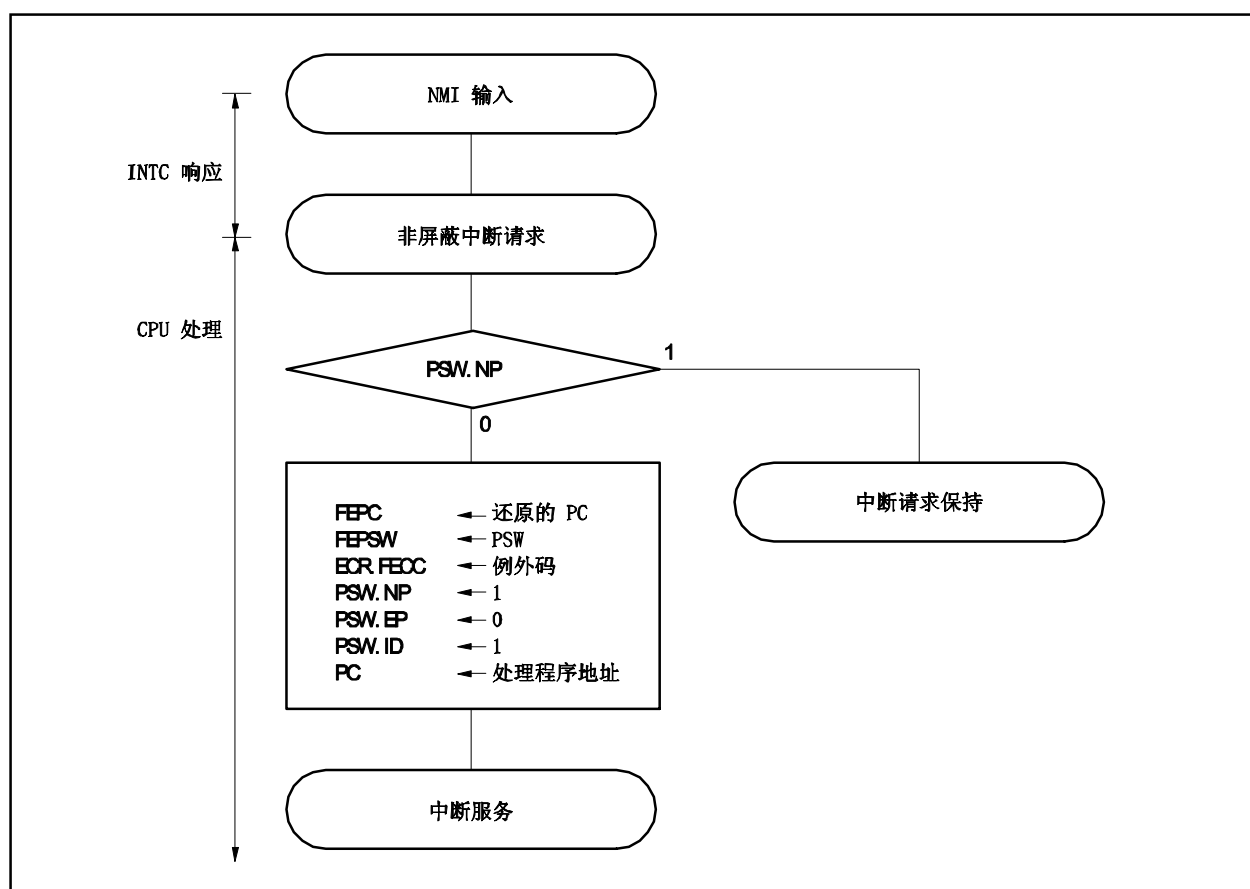
17.2.1 操作

如果产生一个非屏蔽中断请求信号，CPU 执行下列处理，并且将操作权转移给处理例程。

- <1> 将恢复 PC 保存的 FEPC。
- <2> 将当前的 PSW 保存到 FEPSW。
- <3> 向 ECR 的高半字 (FECC) 写入异常代码 (0010H, 0020H, 0030H)。
- <4> 将 PSW.NP 位和 PSW.ID 位置 1 并将 PSW.EP 位清零。
- <5> 在 PC 中设置相应的非屏蔽中断的处理地址 (00000010H, 00000020H, 00000030H)，然后转移操作权。

非屏蔽中断的服务过程如图 17-2 所示。

图 17-2. 非屏蔽中断的服务过程



17.2.2 恢复

通过 RETI 指令从非屏蔽中断服务中执行恢复。

(1) 通过 NMI 引脚输入

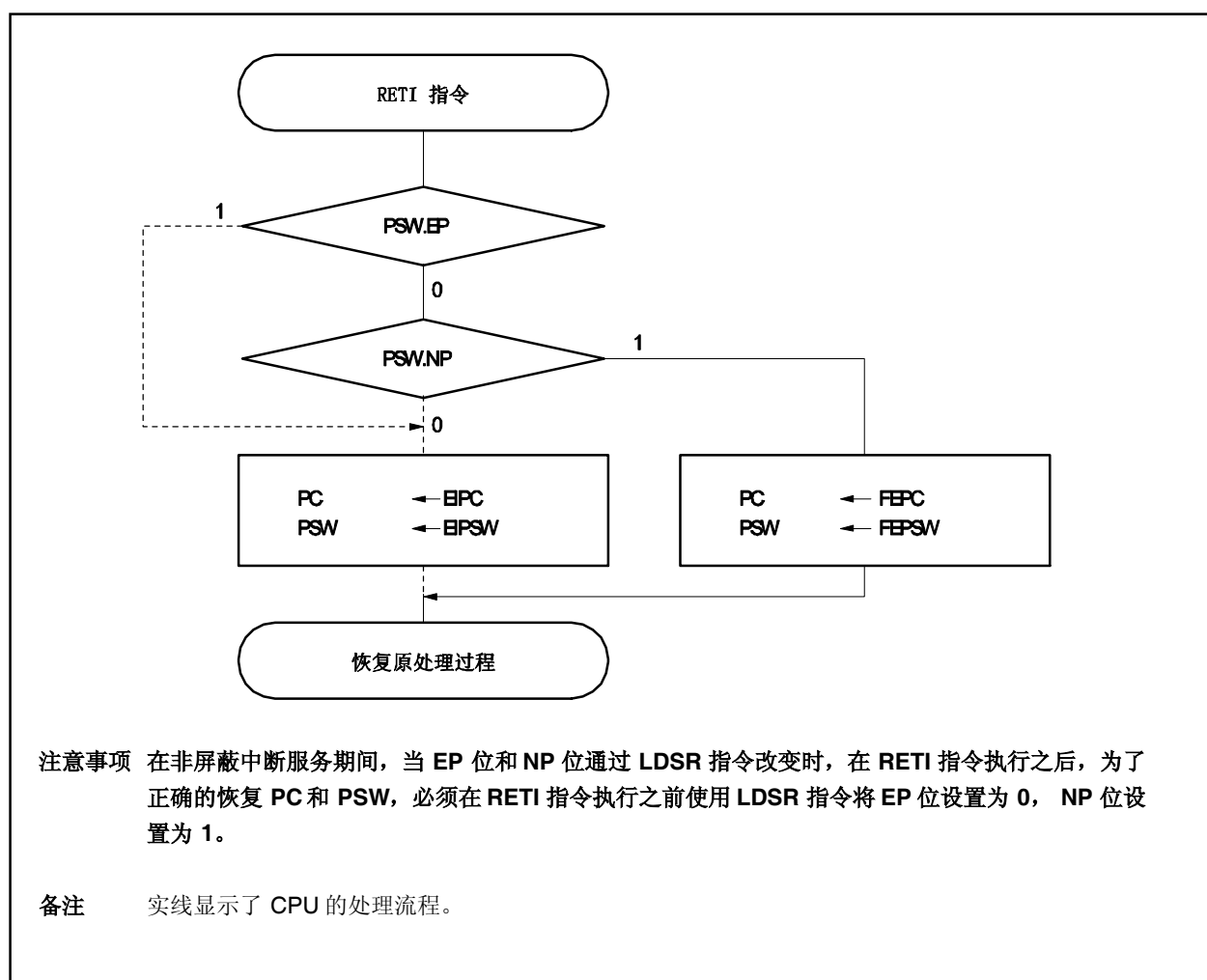
通过 RETI 指令从 NMI 服务中执行恢复。

当执行 RETI 指令时，CPU 执行下列处理，并且将操作权转移给 PC 存储的地址。

- (i) 分别从 FEPC 和 FEPSW 中加载被保存的 PC 和 PSW 的值，因为 PSW.EP 位的值是 0 且 PSW.NP 的值为 1。
- (ii) 操作权转移给恢复 PC 和 PSW 的地址。

图 17-3 说明了 RETI 指令的处理过程。

图 17-3. RETI 指令处理过程



(2) 通过 INTWDT1, INTWDT2 信号

对于非屏蔽中断请求信号（INTWDT1, INTWDT2）引入的非屏蔽中断服务，参见 17.10 注意事项。

17.2.3 NP 标志

NP 标志是一个状态标志，它指示了正在执行非屏蔽中断。当非屏蔽中断请求被确认时会设置该标志，并且屏蔽所有非屏蔽中断以防止多重中断。

复位后: 00000020H

	31		8	7	6	5	4	3	2	1	0
PSW	0			NP	EP	ID	SAT	CY	OV	S	Z

NP	NMI 启用状态
0	没有启用不可屏蔽中断
1	正在启用不可屏蔽中断

17.3 可屏蔽中断

可屏蔽中断请求信号可以通过中断控制寄存器屏蔽。V850ES/KE2 拥有 33 个可屏蔽中断源（参见 17.1.1 特点）。

如果有两个或者更多的可屏蔽中断请求信号同时产生，它们将依照缺省优先级的顺序被确认。除了缺省优先级之外，还有八个优先级等级可以使用，这需要使用中断控制寄存器可变成优先级控制。

如果一个中断请求信号被确认，将禁止对其它可屏蔽中断请求信号的确认，从而进入了中断禁止状态（DI）。

当中断服务程序中执行了 EI 指令时，进入中断允许状态（EI），这将允许具有比正在进行处理的当前中断请求信号更高优先级（通过中断控制寄存器设定）的中断。注意只有优先级更高的中断有这种功能，具有相同优先级的中断不能实现嵌套。

但是为了允许多重中断，在执行 EI 指令之前需要将 EIPC 和 EIPSW 保存到内存或是通用目的寄存器当中，并且在 RETI 指令恢复 EIPC 和 EIPSW 的原始值之前执行 DI 指令。

当 WDTM1.WDTM14 位清零时，看门狗定时器 1 溢出中断为可屏蔽中断（INTWDTM1）。

17.3.1 操作

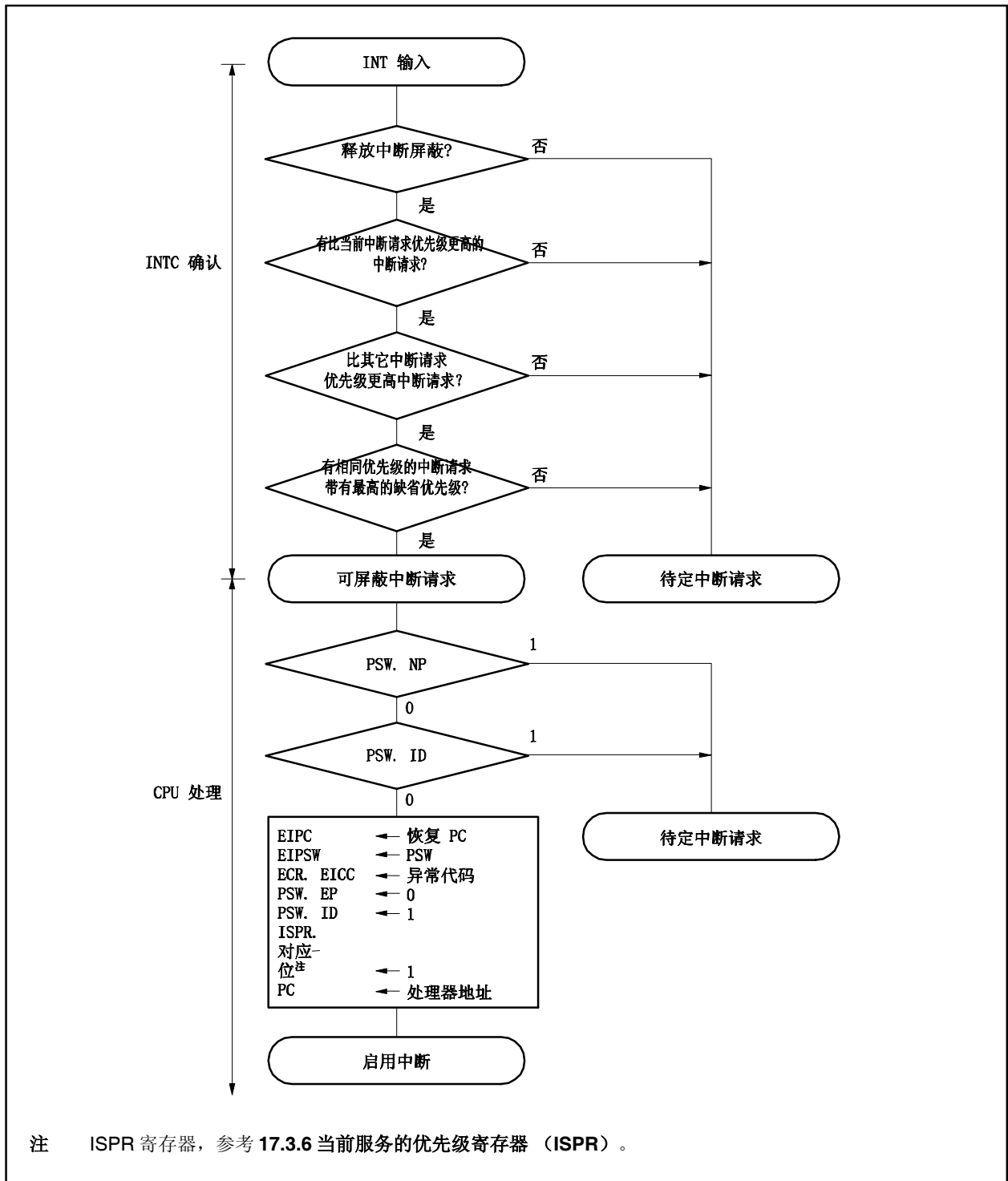
如果产生了可屏蔽中断，CPU 将执行下列处理，并且将操作权转移给处理例程。

- <1> 将恢复的 PC 保存到 EIPC。
- <2> 将当前的 PSW 保存到 EIPSW。
- <3> 向 ECR 的低半字（EICC）写入一个异常代码。
- <4> 将 PSW.ID 位置 1 且将 PSW.EP 位清零。
- <5> 将相应中断的处理地址设置给 PC，并且转移控制权。

通过 INTC 屏蔽的可屏蔽中断请求信号和在另一个中断正在执行过程中（当 PSW.NP 位=1 或是 PSW.ID 位=1）产生的可屏蔽中断请求信号被内部保持。如果可屏蔽中断未被屏蔽，或者通过 RETI 指令或是 LDSR 指令将 NP 位和 ID 位清零，一个新的可屏蔽中断服务将按照可屏蔽中断请求的优先级顺序执行。

图 17-4 说明了可屏蔽中断的执行过程。

图 17-4. 可屏蔽中断服务



17.3.2 恢复

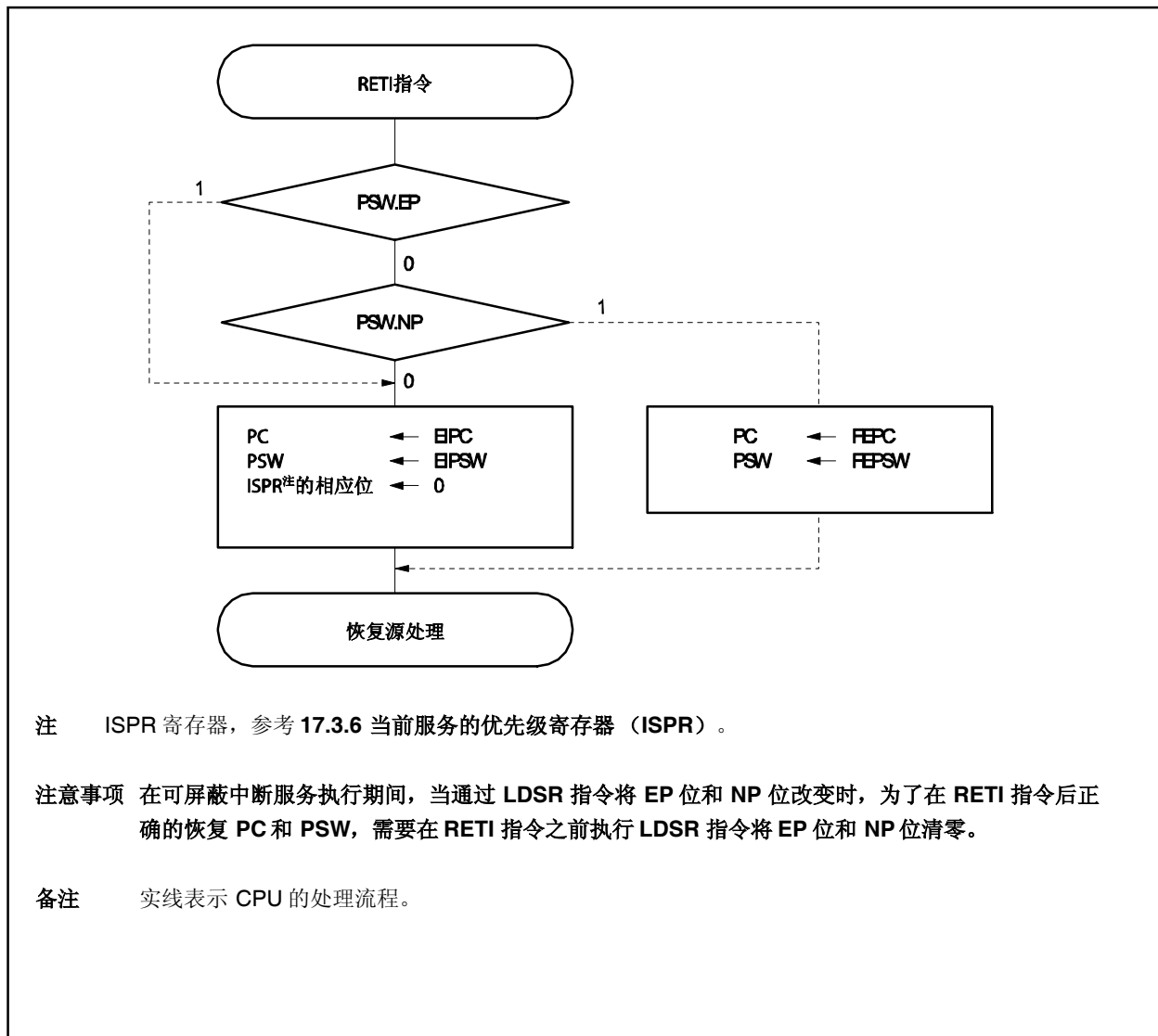
通过执行 RETI 指令能够从可屏蔽中断服务中恢复。

当 RETI 指令执行后，CPU 将执行下列步骤，且将操作权转移给恢复的 PC 地址。

- (1) 从 EIPC 和 EIPSW 中加载恢复的 PC 和 PSW，这是因为 PSW.EP 位为 0 并且 PSW.NP 位为 0。
- (2) 将操作权转移给恢复 PC 和 PSW 的地址。

图 17-5 举例说明 RETI 指令的处理过程。

图 17-5. RETI 指令处理过程



17.3.3 可屏蔽中断的优先级

INTC 执行多重中断服务，这是指当一个中断服务正在进行的时候确认另外一个中断。多重中断可以通过优先级来控制。

一共有两种类型的优先级别控制：基于缺省优先级别的控制和基于可编程优先级别的控制，后者是通过中断优先级指定位（`xxICn.xxPRn` 位）的设置来实现的。当两个具有通过 `xxPRn` 指定相同优先级的中断同时产生时，中断请求信号将依照预先指定给每一种中断类型的优先级别（缺省优先级别）顺序进行中断服务。如需更多的信息，参考表 17-1 中断源列表。可编程的优先级通过设定优先级别制定标志分八个等级对用户化的中断请求信号进行控制。

注意当一个中断请求信号被确认后，`PSW.ID` 标志将自动置 1。因此，当使用复合中断时，需要预先将 `ID` 标志清零来设定中断允许模式（例如，在中断服务程序中设置 `EI` 指令）。

备注 `xx`: 表示任意的外围单元的名称（参考表 17-2 中断控制寄存器（`xxICn`））
 `n`: 外围单元编号（参考表 17-2 中断控制寄存器（`xxICn`））。

图 17-6. 中断嵌套举例 (1/2)

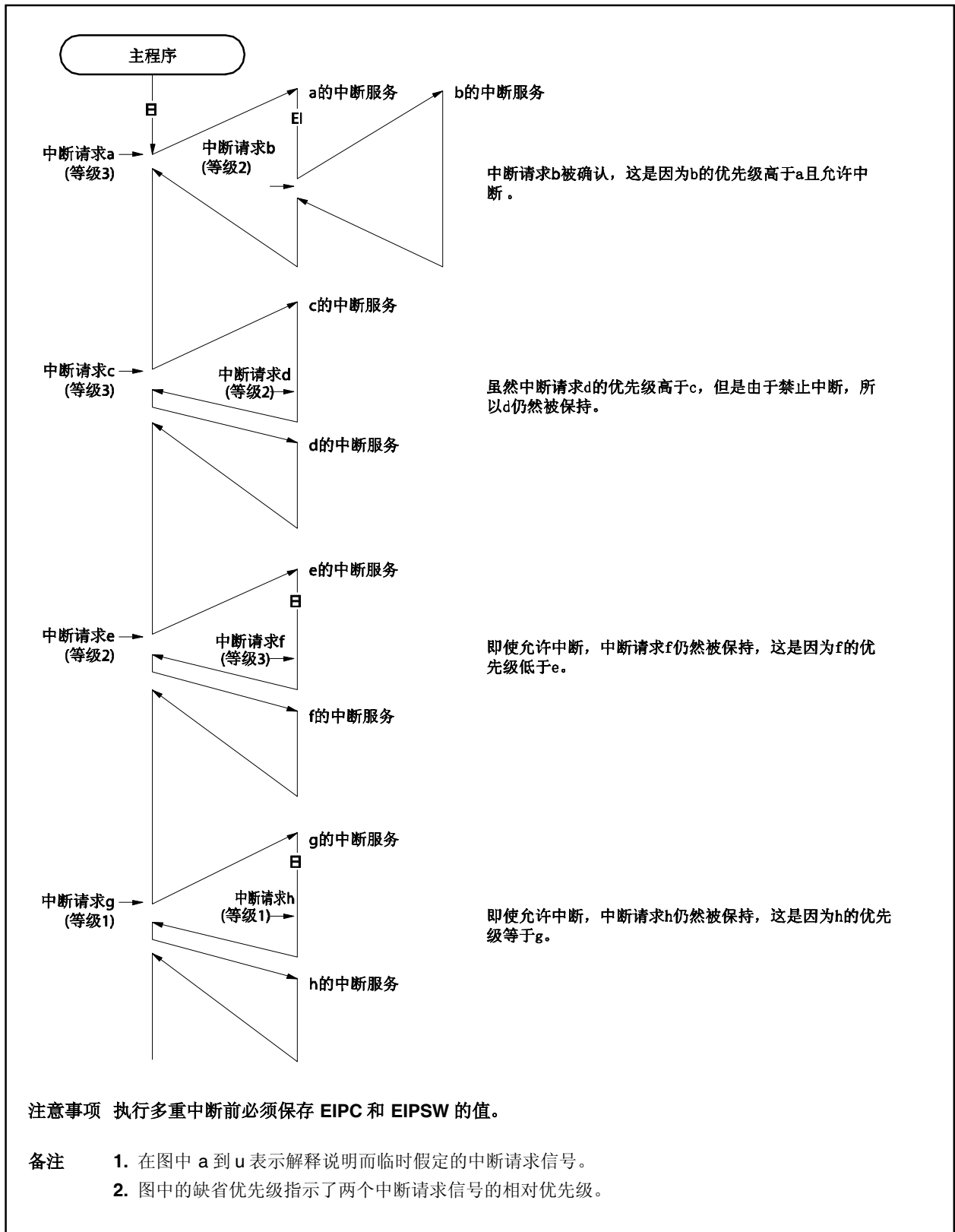


图 17-6. 中断嵌套举例 (2/2)

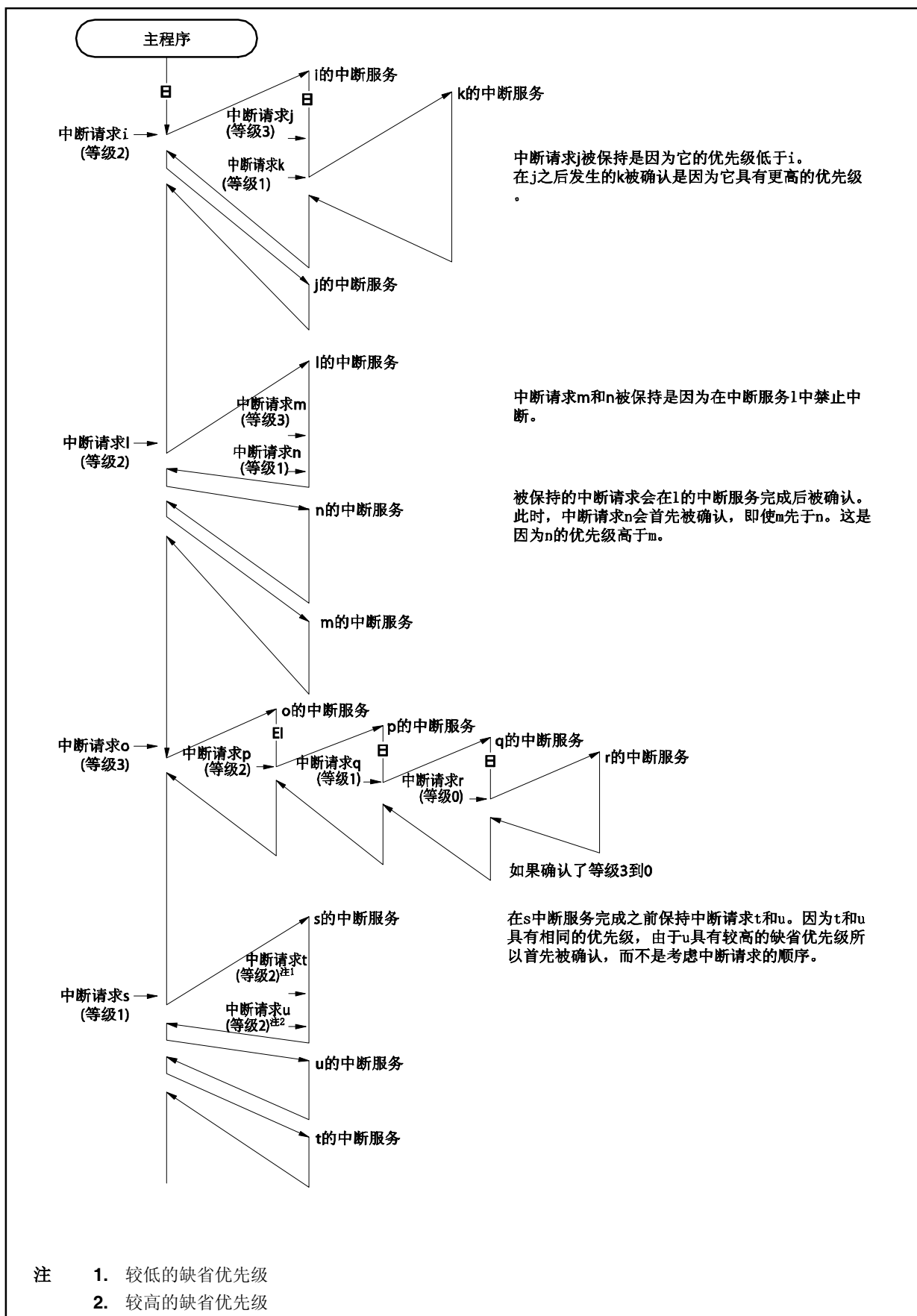
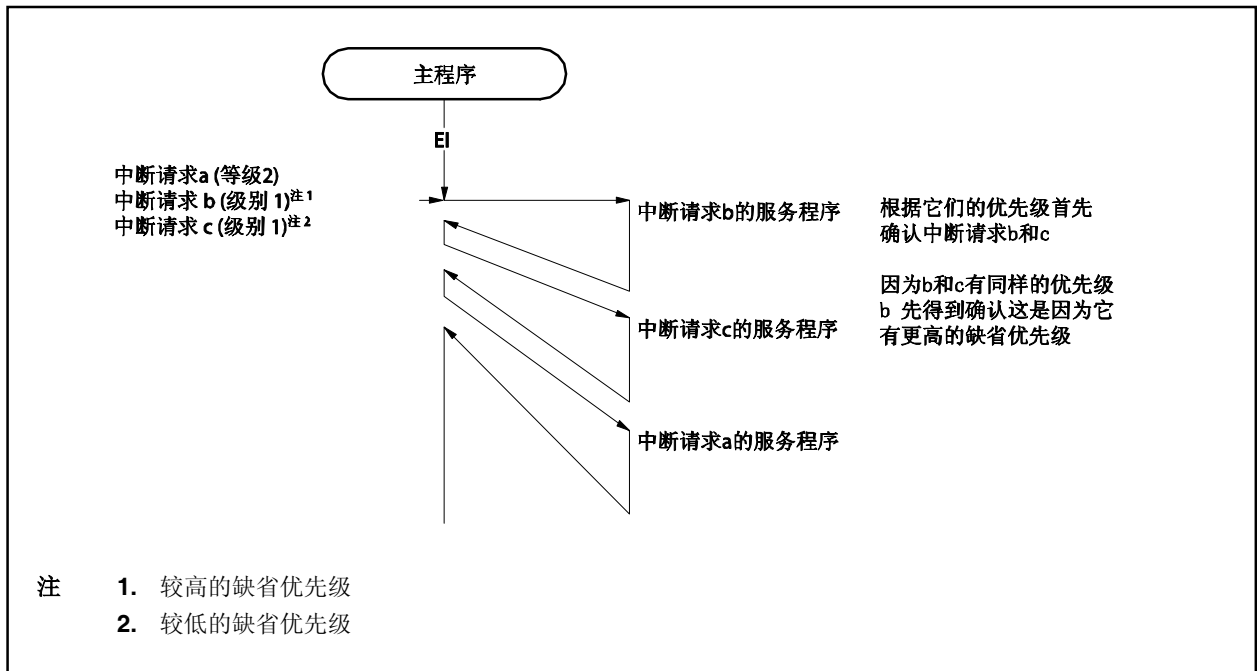


图 17-7. 服务中断请求信号同时发生举例



17.3.4 中断控制寄存器 (xxICn)

xxICn 寄存器被分配给每一个中断请求信号 (可屏蔽中断) 并且设置每一个可屏蔽中断请求控制条件。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 47H。

注意事项 禁止中断 (DI) 或者屏蔽中断来读取 xxICn.xxIFn 位。如果在允许中断 (EI) 或是中断未被屏蔽的时候, 读取 xxIFn 位, 确认中断与读取该位冲突时数据可能有误。

复位后: 47H R/W 地址: FFFFF110H ~ FFFFF168H

	<7>	<6>	5	4	3	2	1	0
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	中断请求标志 ^注
0	没有生成中断请求
1	生成中断请求

xxMKn	中断屏蔽标志
0	容许进行中断
1	禁止进行中断 (待定)

xxPRn2	xxPRn1	xxPRn0	中断优先级规格位
0	0	0	规定 级别0 (最高)
0	0	1	规定 级别 1
0	1	0	规定 级别 2
0	1	1	规定 级别 3
1	0	0	规定 级别 4
1	0	1	规定 级别 5
1	1	0	规定 级别 6
1	1	1	规定 级别 7 (最低)

注 如果确认中断信号后, xxIFn 标志通过硬件自动复位。

备注 xx: 表示任意的外围单元的名称 (参考表 17-2 中断控制寄存器 (xxICn))
n: 外围单元编号 (参考表 17-2 中断控制寄存器 (xxICn))

中断控制寄存器的地址和位如下表所示。

表 17-2. 中断控制寄存器 (xxICn)

地址	寄存器	位							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	WDT1IC	WDT1IF	WDT1MK	0	0	0	WDT1PR2	WDT1PR1	WDT1PR0
FFFFF112H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF114H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF116H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF118H	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
FFFFF11AH	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
FFFFF11CH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
FFFFF11EH	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60
FFFFF124H	TM0IC10	TM0IF10	TM0MK10	0	0	0	TM0PR102	TM0PR101	TM0PR100
FFFFF126H	TM0IC11	TM0IF11	TM0MK11	0	0	0	TM0PR112	TM0PR111	TM0PR110
FFFFF128H	TM5IC0	TM5IF0	TM5MK0	0	0	0	TM5PR02	TM5PR01	TM5PR00
FFFFF12AH	TM5IC1	TM5IF1	TM5MK1	0	0	0	TM5PR12	TM5PR11	TM5PR10
FFFFF12CH	CSI0IC0	CSI0IF0	CSI0MK0	0	0	0	CSI0PR02	CSI0PR01	CSI0PR00
FFFFF12EH	CSI0IC1	CSI0IF1	CSI0MK1	0	0	0	CSI0PR12	CSI0PR11	CSI0PR10
FFFFF130H	SREIC0	SREIF0	SREMK0	0	0	0	SREPR02	SREPR01	SREPR00
FFFFF132H	SRIC0	SRIF0	SRMK0	0	0	0	SRPR02	SRPR01	SRPR00
FFFFF134H	STIC0	STIF0	STMK0	0	0	0	STPR02	STPR01	STPR00
FFFFF136H	SREIC1	SREIF1	SREMK1	0	0	0	SREPR12	SREPR11	SREPR10
FFFFF138H	SRIC1	SRIF1	SRMK1	0	0	0	SRPR12	SRPR11	SRPR10
FFFFF13AH	STIC1	STIF1	STMK1	0	0	0	STPR12	STPR11	STPR10
FFFFF13CH	TMHIC0	TMHIF0	TMHMK0	0	0	0	TMHPR02	TMHPR01	TMHPR00
FFFFF13EH	TMHIC1	TMHIF1	TMHMK1	0	0	0	TMHPR12	TMHPR11	TMHPR10
FFFFF142H	IICIC0	IICIF0	IICMK0	0	0	0	IICPR02	IICPR01	IICPR00
FFFFF144H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF146H	KRIC	KRIF	KRMK	0	0	0	KRPR2	KRPR1	KRPR0
FFFFF148H	WTIIC	WTIIF	WTIMK	0	0	0	WTIPR2	WTIPR1	WTIPR0
FFFFF14AH	WTIC	WTIF	WTMK	0	0	0	WTPR2	WTPR1	WTPR0
FFFFF14CH	BRGIC	BRGIF	BRGMK	0	0	0	BRGPR2	BRGPR1	BRGPR0
FFFFF172H	PIC7	PIF7	PMK7	0	0	0	PPR72	PPR71	PPR70
FFFFF174H	TP0OVIC	TP0OVIF	TP0OVMK	0	0	0	TP0OVPR2	TP0OVPR1	TP0OVPR0
FFFFF176H	TP0CCIC0	TP0CCIF0	TP0CCMK0	0	0	0	TP0CCPR02	TP0CCPR01	TP0CCPR00
FFFFF178H	TP0CCIC1	TP0CCIF1	TP0CCMK1	0	0	0	TP0CCPR12	TP0CCPR11	TP0CCPR10

17.3.5 中断屏蔽寄存器 0, 1, 3 (IMR0, IMR1, IMR3)

这些寄存器为可屏蔽中断设定中断屏蔽状态。IMR0, IMR1 和 IMR3 寄存器的 xxMKn 位与 xxCn 寄存器的 xxMKn 位一一对应。

IMRm 寄存器支持 16 位读写方式 (m = 0, 1, 3)。

如果 IMRk 寄存器的高 8 位用作 IMRkH 寄存器, 低 8 位用作 IMRkL 寄存器, 该寄存器支持 8 位或 1 位读写方式 (k = 0, 1)。

注意事项 在此设备文件中, xxCn 寄存器的 xxMKn 位定义为一个保留字。因此, 如果以 xxMKn 名称执行位操作, 则 xxCn 寄存器而不是 IMRm 寄存器被复写 (结果 IMRm 寄存器也被复写)。

复位后: FFFFH		R/W	地址:		IMR0 FFFF100H, IMR0L FFFF100H, IMR0H FFFF101H																																							
			15	14	13	12	11	10	9	8																																		
IMR0 (IMR0H ^注)	CS10MK1	CS10MK0	TM5MK1	TM5MK0	TM0MK11	TM0MK10	1	1																																				
	7	6	5	4	3	2	1	0																																				
(IMR0L)	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	WDT1MK																																				
复位后: FFFFH		R/W	地址:		IMR1 FFFF102H, IMR1L FFFF102H, IMR1H FFFF103H																																							
			15	14	13	12	11	10	9	8																																		
IMR1 (IMR1H ^注)	1	BRGMK	WTMK	WTIMK	KRMK	ADMK	IICMK0	1																																				
	7	6	5	4	3	2	1	0																																				
(IMR1L)	TMHMK1	TMHMK0	STMK1	SRMK1	SREMK1	STMK0	SRMK0	SREMK0																																				
复位后: FFFFH		R/W	地址:		IMR3, IMR3L FFFF106H																																							
			15	14	13	12	11	10	9	8																																		
IMR3	1	1	1	1	1	1	1	1	1																																			
	7	6	5	4	3	2	1	0																																				
(IMR3L)	1	1	1	TPOCCMK1	TPOCCMK0	TPOVMK	PMK7	1																																				
		<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td>xxMKn</td> <td colspan="10">中断屏蔽标志设置</td> </tr> <tr> <td>0</td> <td colspan="10">容许中断进行</td> </tr> <tr> <td>1</td> <td colspan="10">禁止中断进行</td> </tr> </table>										xxMKn	中断屏蔽标志设置										0	容许中断进行										1	禁止中断进行									
xxMKn	中断屏蔽标志设置																																											
0	容许中断进行																																											
1	禁止中断进行																																											

注 为了以 8 位或 1 位为单元读取或写入 IMR0 和 IMR1 寄存器的位 8 到位 15, 需要将它们设定为 IMR0H 和 IMR1H 寄存器的位 0 到位 7。

注意事项 将 IMR0 寄存器的位 9 和 8, IMR1 寄存器的位 15 和 8, 以及 IMR3 寄存器的位 15 到 5 和位 0 置 1。如果这些位的设定值改变, 操作将不会得到保证。

备注 xx: 表示任意的外围单元的名称 (参考表 17-2 中断控制寄存器 (xxICn))
n: 外围单元编号 (参考表 17-2 中断控制寄存器 (xxICn))

17.3.6 当前服务的优先级寄存器 (ISPR)

这个寄存器保存当前正在应答的可屏蔽中断的优先级。当中断请求信号被应答，与该中断请求信号的优先级相应的位置 1，并始终保持此设置直到执行该中断的服务。

当执行 RETI 指令后，对应具有最高优先级的中断请求信号的位通过硬件自动复位为 0。但是，当从非屏蔽中断服务或是异常处理中返回时，该位不会复位为 0。

该寄存器支持 8 位或 1 位只读方式。

ISPR 寄存器复位后的值为 00H。

注意事项 如果在中断允许状态 (EI) 对 ISPR 寄存器进行读取时确认一个中断，在寄存器的位通过确认中断的方式被设置之后，可以读取 ISPR 寄存器的值。为了在确认中断之前正确的读取 ISPR 寄存器的值，需要禁止中断 (DI)。

复位后: 00H R 地址: FFFF1FAH								
ISPR	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0
	ISPRn	当前确认的中断优先级						
	0	带优先级的中断请求没得到确认						
	1	带优先级n的中断请求得到确认						
备注	n = 0~7 (优先级别)							

17.3.8 看门狗定时器模式寄存器 1 (WDTM1)

这是一个特殊的寄存器，只能在特定的顺序下写入。要产生一个可屏蔽中断 (INTWDT1)，将 WDTM14 位清零。

该寄存器支持 8 位或 1 位读写方式 (关于更多细节，敬请参阅第 11 章 看门狗定时器的功能)。

复位后: 00H R/W Address: FFFFF6C2H

	<7>	6	5	4	3	2	1	0
WDTM1	RUN1	0	0	WDTM14	WDTM13	0	0	0

RUN1	看门狗定时器操作模式选择 ^{注 1}
0	停止计数操作
1	清0计数器，开始计数操作

WDTM14	WDTM13	看门狗定时器操作模式选择 ^{注 2}
0	0	间隔定时器模式 (在溢出时生成可屏蔽中断INTWDTM1)
0	1	
1	0	看门狗定时器模式 1 ^{注 3} (在溢出时生成不可屏蔽中断INTWDT1)
1	1	看门狗定时器模式 2 (在溢出时开始WDTRES2复位操作)

- 注**
- 一旦 RUN1 位被置 1，不能由软件将其清零。因此，一旦计数开始，只有复位才能使其停止。
 - 一旦 WDTM14 和 WDTM13 位被置 1，不能由软件将其清零。只有复位才能使这些位清零。
 - 对于非屏蔽中断请求信号 (INTWDT1) 的非屏蔽中断服务，参见 17.10 注意事项。

17.4 外部中断请求输入引脚（NMI 和 INTP0~INTP7）

17.4.1 噪声消除

（1）NMI 引脚的噪声消除

NMI 引脚拥有内部噪声消除电路，它使用模拟延迟。因此，NMI 引脚的输入信号不会作为边沿被检测，除非输入电平保持一定的时间。所以只有特定的时间过去后才会检测此边沿。

NMI 引脚可以用来释放 STOP 模式。在 STOP 模式中，使用系统时钟不会消除噪声，因为内部系统时钟是停止的。

（2）INTP0~INTP2 及 INTP4~INTP7 引脚的噪声消除

INTP0 到 INTP2 及 INTP4 到 INTP7 引脚拥有内部噪声消除电路，它使用模拟延迟。因此，这些引脚的输入信号不会作为边沿被检测，除非输入电平保持一定的时间。所以只有特定的时间过去后才会检测此边沿。

（3）INTP3 引脚的噪声消除

INTP3 拥有一个数字/模拟噪声消除电路，可由 NFC.NFEN 位选择。

数字噪声消除电路的采样数可由 NFC.NFSTS 位选择，可选值为 2 和 3。采样时钟可由 NFC.NFC2 到 NFC.NFC0 位在 $f_{xx}/64$ ， $f_{xx}/128$ ， $f_{xx}/256$ ， $f_{xx}/512$ ， $f_{xx}/1024$ 和 f_{XT} 中选择。如果采样时钟设为 $f_{xx}/64$ ， $f_{xx}/128$ ， $f_{xx}/256$ ， $f_{xx}/512$ 或 $f_{xx}/1024$ ，则采样时钟在 IDLE/STOP 模式停止。因此不能用它释放待机模式。要释放待机模式，选择 f_{XT} 为采样时钟或选择模拟噪声消除。

(a) 数字噪声消除控制寄存器 (NFC)

NFC 寄存其控制 INTP3 引脚的噪声消除。如果噪声消除时钟为 f_{XT} ，则 INTP3 引脚的外部中断功能可用，即使是 IDLE/STOP 模式。

该寄存器支持 8 位或 1 位读写方式。

NFC 寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFFF318H

	7	6	5	4	3	2	1	0
NFC	NFEN	NFSTS	0	0	0	NFC2	NFC1	NFC0

NFEN	INTP3引脚噪声清除的设置
0	模拟噪声清除
1	设置噪声清除

NFSTS	数字噪声清除抽样次数的设置
0	抽样次数 = 3次
1	抽样次数 = 2 次

NFC2	NFC1	NFC0	抽样设置选择
0	0	0	$f_{XX}/64$
0	0	1	$f_{XX}/128$
0	1	0	$f_{XX}/256$
0	1	1	$f_{XX}/512$
1	0	0	$f_{XX}/1024$
1	0	1	f_{XT}
以上除外			禁止设置

备注 f_{XX} : 主时钟频率
 f_{XT} : 副时钟频率

<噪声消除宽度>

噪声消除宽度 (t_{WIT3}) 如下, 其中 T 为采样时钟周期, M 为采样数。

- $t_{WIT3} < (M - 1) T$: 确定为噪声被消除
- $(M - 1) T \leq t_{WIT3} < MT$: 可能被当作噪声消除, 可能被当作有效边沿检测
- $t_{WIT3} \geq MT$: 确定为有效边沿被检测

为了使得输入到 $INTP3$ 引脚的有效边沿被准确检测, 需要输入宽度大于 MT 的脉冲信号。

NFSTS	NFC2	NFC1	NFC0	采样时钟	最小消除噪声宽度		
					$f_{xx} = 20 \text{ MHz}$	$f_{xx} = 10 \text{ MHz}$	$f_{xx} = 8 \text{ MHz}$
0	0	0	0	$f_{xx}/64$	$6.4 \mu\text{s}$	$12.8 \mu\text{s}$	$16 \mu\text{s}$
0	0	0	1	$f_{xx}/128$	$12.8 \mu\text{s}$	$25.6 \mu\text{s}$	$32 \mu\text{s}$
0	0	1	0	$f_{xx}/256$	$25.6 \mu\text{s}$	$51.2 \mu\text{s}$	$64 \mu\text{s}$
0	0	1	1	$f_{xx}/512$	$51.2 \mu\text{s}$	$102.4 \mu\text{s}$	$128 \mu\text{s}$
0	1	0	0	$f_{xx}/1024$	$102.4 \mu\text{s}$	$204.8 \mu\text{s}$	$256 \mu\text{s}$
0	1	0	1	$f_{xt} (32.768 \text{ kHz})$	$61.04 \mu\text{s}$		
1	0	0	0	$f_{xx}/64$	$3.2 \mu\text{s}$	$6.4 \mu\text{s}$	$8 \mu\text{s}$
1	0	0	1	$f_{xx}/128$	$6.4 \mu\text{s}$	$12.8 \mu\text{s}$	$16 \mu\text{s}$
1	0	1	0	$f_{xx}/256$	$12.8 \mu\text{s}$	$25.6 \mu\text{s}$	$32 \mu\text{s}$
1	0	1	1	$f_{xx}/512$	$25.6 \mu\text{s}$	$51.2 \mu\text{s}$	$64 \mu\text{s}$
1	1	0	0	$f_{xx}/1024$	$51.2 \mu\text{s}$	$102.4 \mu\text{s}$	$128 \mu\text{s}$
1	1	0	1	$f_{xt} (32.768 \text{ kHz})$	$30.52 \mu\text{s}$		
以上除外				禁止设置			

17.4.2 边沿检测

每一个 NMI 和 $INTP0 \sim INTP7$ 引脚的有效边沿可以在以下四种方式中选择。

- 上升沿
- 下降沿
- 双边沿
- 无边沿检测

在复位后, NMI 引脚不会检测边沿。因此, 中断请求信号不会被确认 (NMI 引脚功能作为一个普通端口引脚), 除非使用 $INTFO$ 和 $INTRO$ 寄存器允许有效边沿。

若 $P02$ 用于输出端口, 应将 NMI 引脚设置为“无边沿检测”。

(1) 外部中断下降，上升沿指定寄存器 0 (INTF0, INTR0)

INTF0 和 INTR0 寄存器是 8 位寄存器，指定了 NMI 引脚的上升沿和下降沿的检测，外部中断引脚 (INTP0 to INTP3) 的上升沿和下降沿的检测。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

注意事项 在引脚的功能从外部中断功能（复用功能）转变到端口功能的过程中，可能会执行边沿检测。因此，首先要将 INTF0n 位和 INTR0n 位设定为 00，然后再设定端口模式。

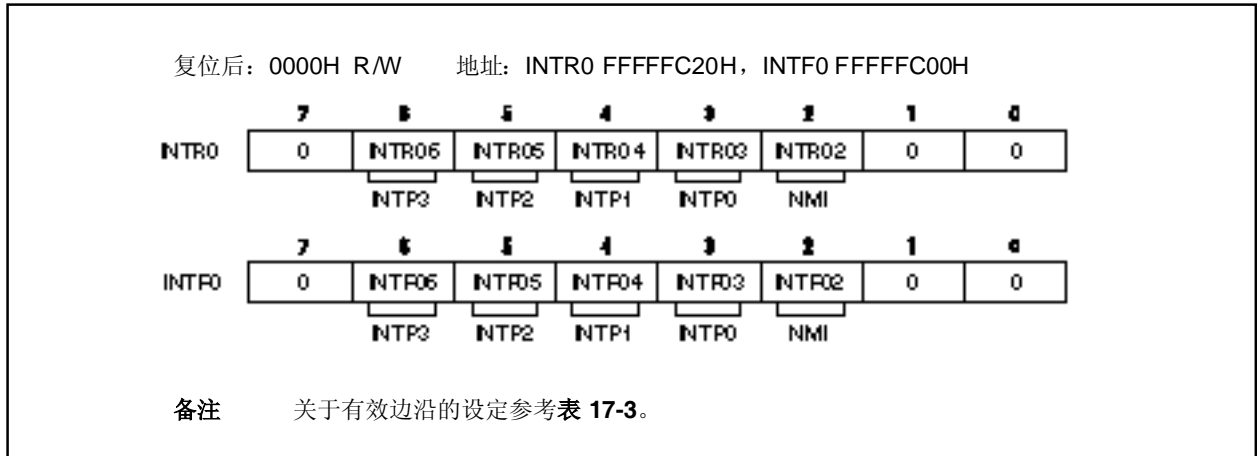


表 17-3. NMI 和 INTP0~INTP3 引脚的有效边沿设定

INTF0n	INTR0n	有效边沿设定 (n = 2~6)
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	双边沿

备注 n=2: 控制 NMI 引脚
 n=3~6: 控制 INTP0~INTP3 引脚

(2) 外部中断下降，上升沿指定寄存器 3 (INTF3, INTR3)

INTF3 和 INTR3 寄存器是 8 位寄存器，它指定了外部中断引脚 (INTP7) 的上升沿和下降沿的检测。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

注意事项 在引脚的功能从外部中断功能 (复用功能) 转变到端口功能的过程中，可能会执行边沿检测。因此，首先要将 INTF31 和 INTR31 位设定为 00，然后再设定端口模式。

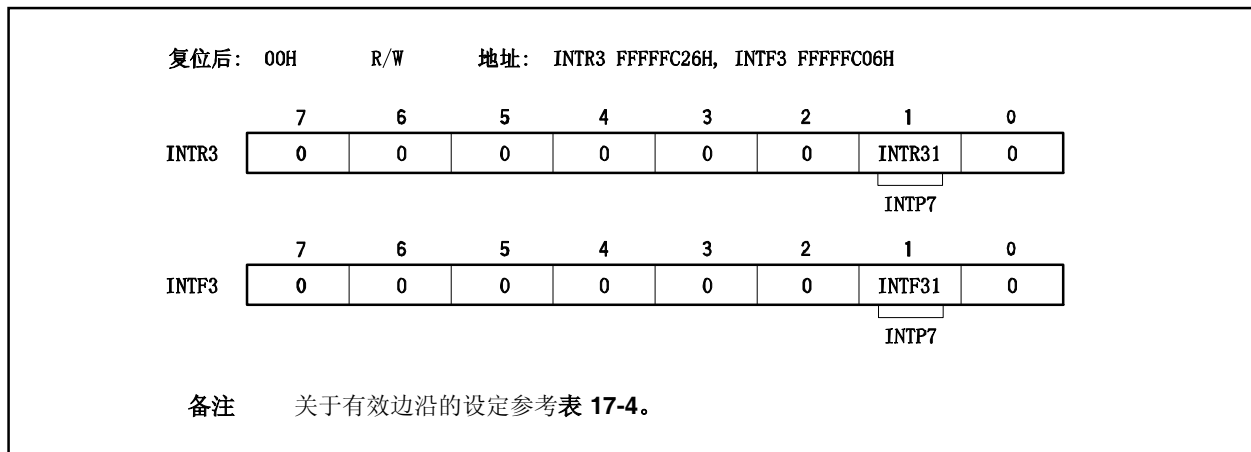


表 17-4. INTP7 引脚有效边沿设定

INTF31	INTR31	有效边沿设定
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	双边沿

(3) 外部中断下降，上升沿指定寄存器 9H (INTF9H, INTR9H)

INTF9H 和 INTR9H 寄存器是 8 位寄存器，它们指定了外部中断引脚 (INTP4 到 INTP6) 的上升沿和下降沿的检测。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

注意事项 当引脚的功能从外部中断功能 (复用功能) 转变到端口功能时，可以进行边沿检测。因此，首先要将 INTF9n 位和 INTR9n 位设定为 00，然后再设定端口模式。

复位后: 00H R/W 地址: INTR9H FFFFC33H, INTF9H FFFFC13H

	7	6	5	4	3	2	1	0
INTR9H	INTR915	INTR914	INTR913	0	0	0	0	0
	INTP6	INTP5	INTP4					

	7	6	5	4	3	2	1	0
INTF9H	INTF915	INTF914	INTF913	0	0	0	0	0
	INTP6	INTP5	INTP4					

备注 关于有效边沿的设定参考表 17-5。

表 17-5. INTP4 到 INTP6 引脚有效边沿设定

INTF9n	INTR9n	有效边沿设定 (n = 13~15)
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	双边沿

备注 n = 13~15: 控制 INTP4~INTP6 引脚

17.5 软件异常

当 CPU 执行 TRAP 指令的时候会产生一个软件的异常，且总是会被确认。

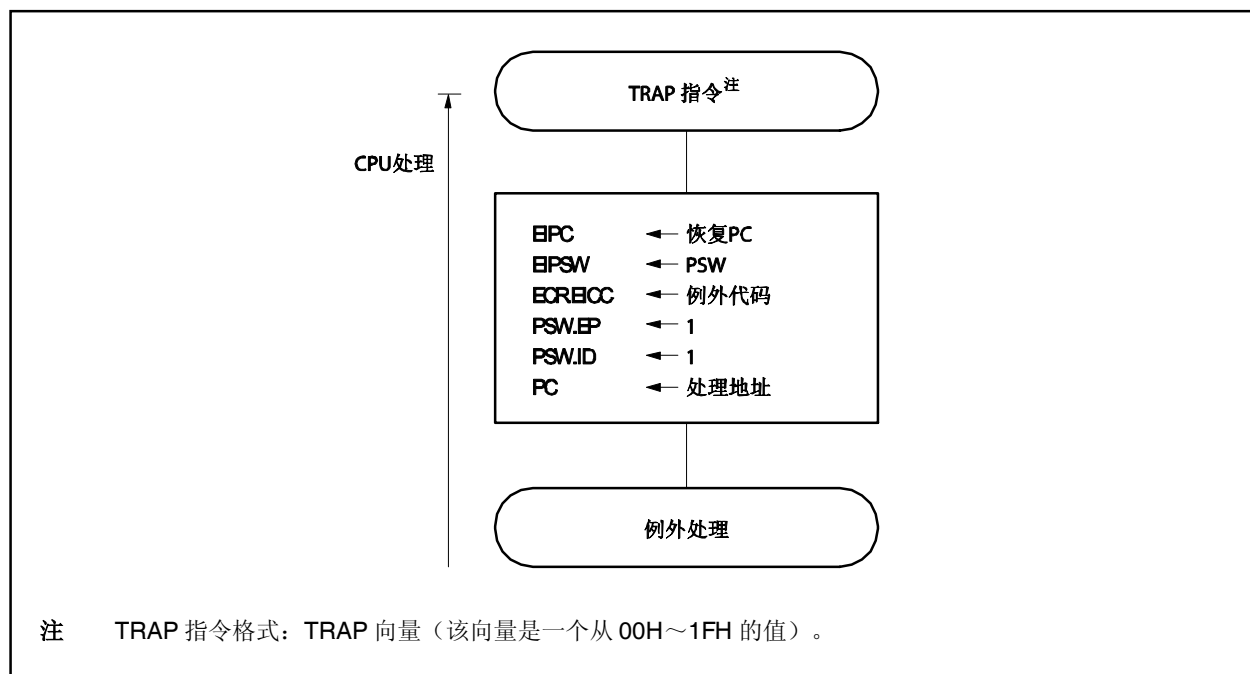
17.5.1 操作

如果一个软件异常产生，CPU 将执行下列处理过程，并且将操作权转移给处理例程。

- <1> 将恢复的 PC 保存在 EIPC 中。
- <2> 将当前的 PSW 保存在 EIPSW 中。
- <3> 在 ECR（中断源）的低 16 位（EICC）中写入一个异常代码。
- <4> 将 PSW.EP 位和 PSW.ID 位置 1。
- <5> 在 PC 中设置附和软件异常的处理地址（00000040H 或 00000050H），并且转移控制权。

图 17-8 说明软件异常的处理过程。

图 17-8. 软件异常处理过程



处理地址是通过 TRAP 指令的操作数（向量）决定的。如果该向量是 00H~0FH，它将变为 00000040H；而如果向量的值是 10H~1FH，它将变为 00000050H。

17.5.2 恢复

通过执行 RETI 指令可以从软件异常进程中恢复。

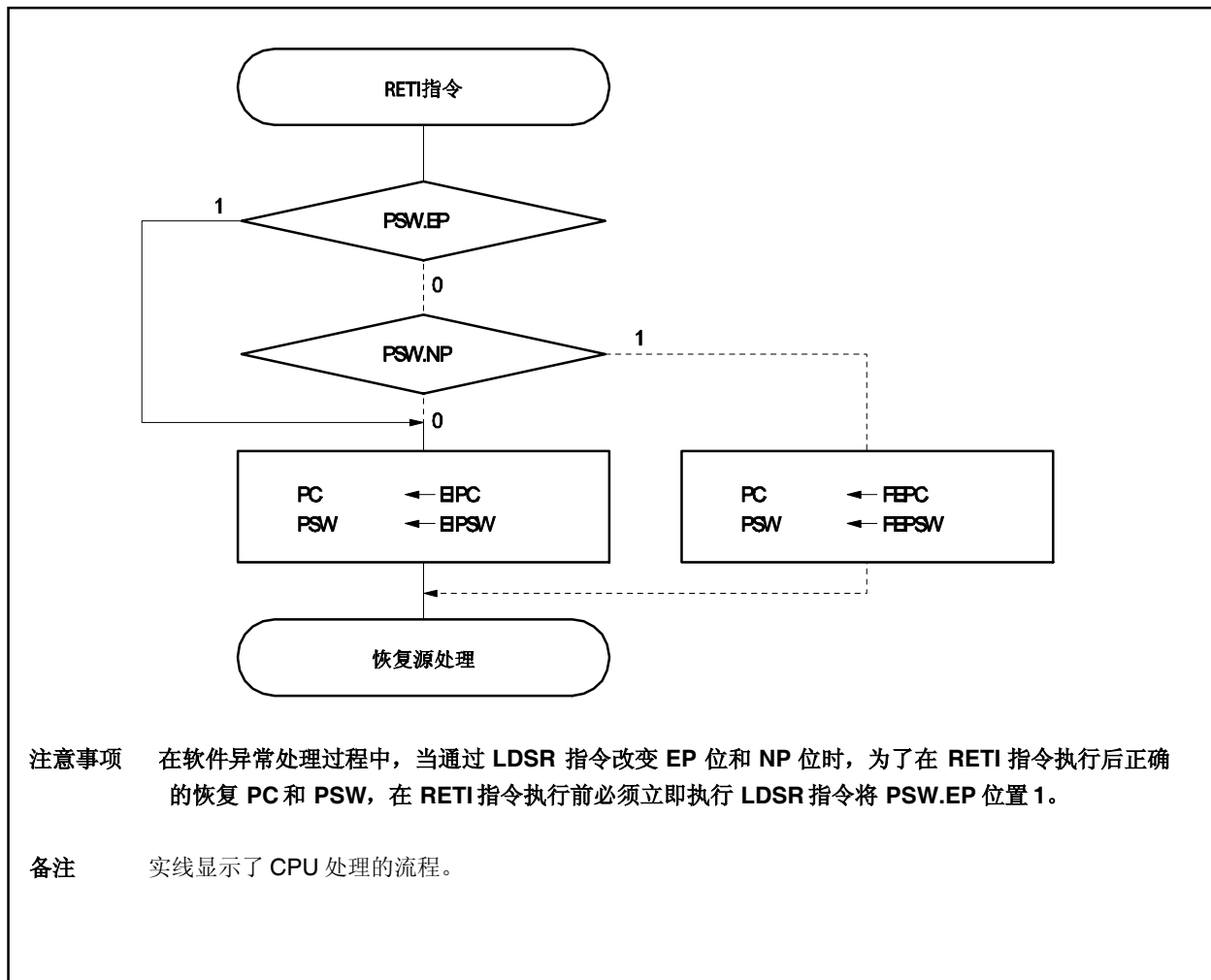
通过执行 RETI 指令，CPU 执行下列处理过程，并且将控制权转移到恢复 PC 的地址。

<1> 由于 PSW.EP 位为 1，从 EIPC 和 EIPSW 中加载恢复的 PC 和 PSW。

<2> 将控制权转移到恢复 PC 和 PSW 的地址。

图 17-9 说明了 RETI 指令的处理过程。

图 17-9. RETI 指令处理过程

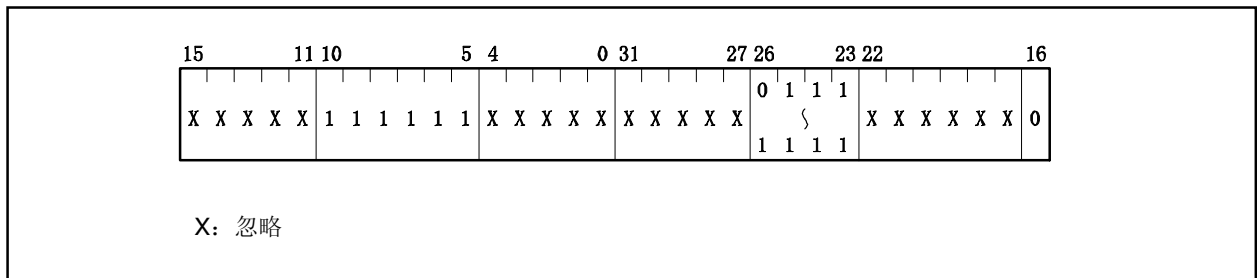


17.6 异常陷阱

异常陷阱是一个中断，在指令的非法操作发生时需要使用。在 V850ES/KE2，一个非法的操作代码陷阱（ILGOP：非法的操作代码陷阱）被认为是一个异常陷阱。

17.6.1 非法的操作代码定义

非法指令位 10~位 5 的操作代码是 111111B，副操作代码（位 26~位 23）是 0111B~1111B，副操作代码（位 16）是 0B。当这个非法指令执行后，产生一个异常陷阱。



注意事项 由于将来有可能分配指令到一个非法的操作代码，所以建议不要使用非法操作代码。

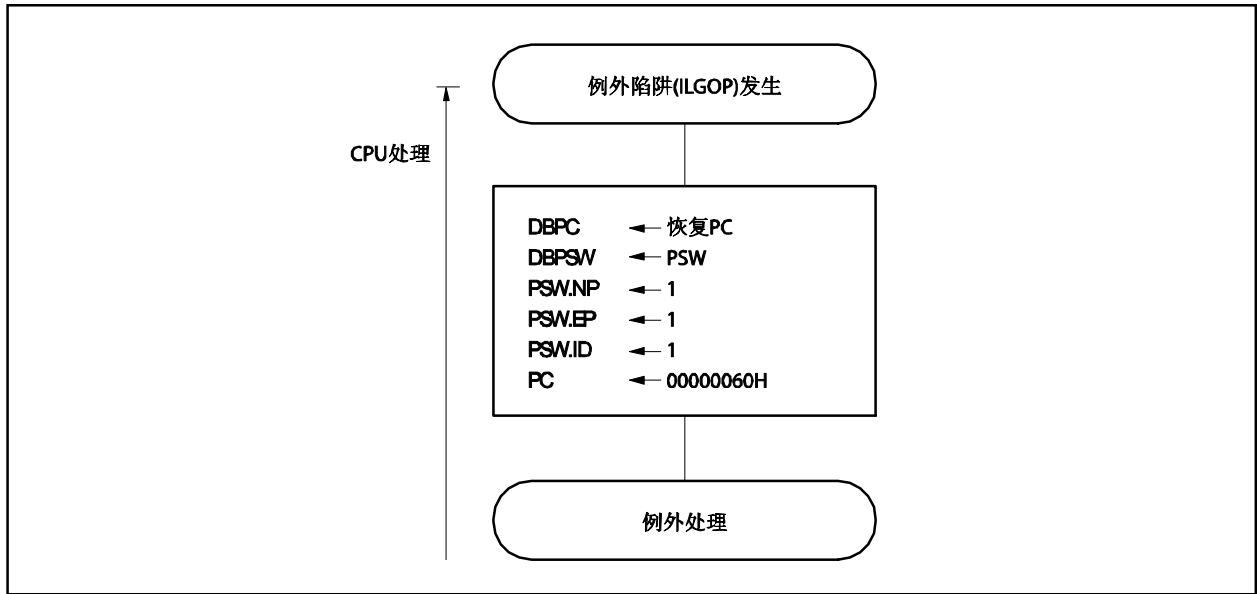
(1) 操作

如果产生一个异常陷阱，CPU 执行下列处理过程，并且将操作权转移给处理例程。

- <1> 将恢复的 PC 保存到 DBPC。
- <2> 将当前的 PSW 保存到 DBPSW。
- <3> 将 PSW.NP 位，PSW.EP 位，和 PSW.ID 位置 1。
- <4> 在 PC 中设置相应异常陷阱的处理地址（00000060H），并且转移控制权。

图 17-10 说明异常陷阱的处理过程。

图 17-10. 异常陷阱的处理过程



(2) 恢复

通过执行 DBRET 指令实现从异常陷阱中恢复。通过执行 DBRET 指令，CPU 执行下列处理过程并且控制着恢复 PC 的地址。

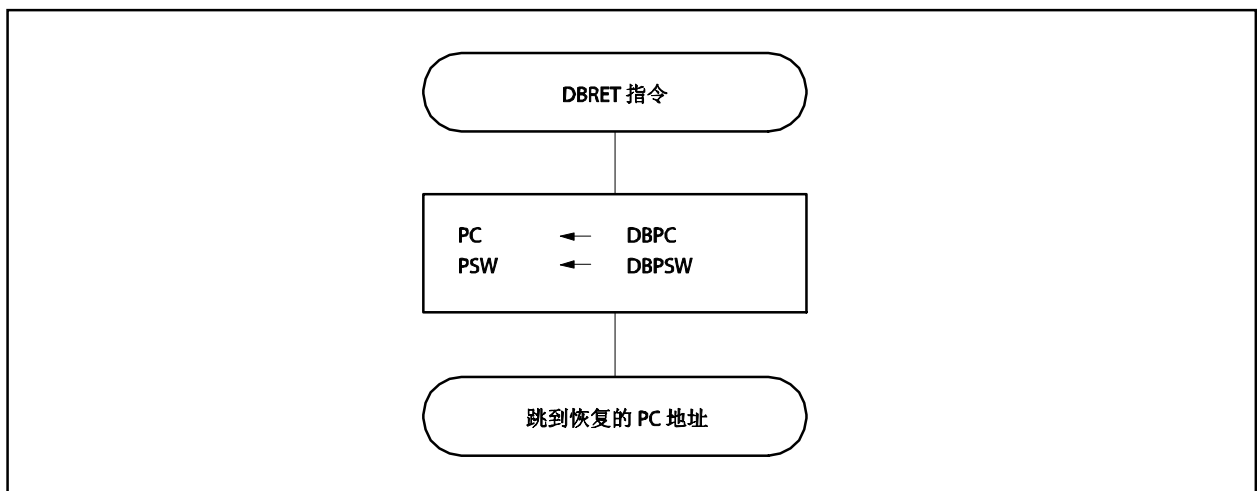
<1> 从 DBPC 和 DBPSW 中加载恢复的 PC 和 PSW。

<2> 向恢复的 PC 和 PSW 代表的地址转移控制权。

<R> 注意事项 只有在执行一条非法代码和执行 DBRET 指令之间的间隔时才可以访问 DBPC 和 DBPSW。

图 17-11 说明从异常陷阱中恢复的处理过程。

图 17-11. 从异常陷阱中恢复的处理流程



17.6.2 调试陷阱

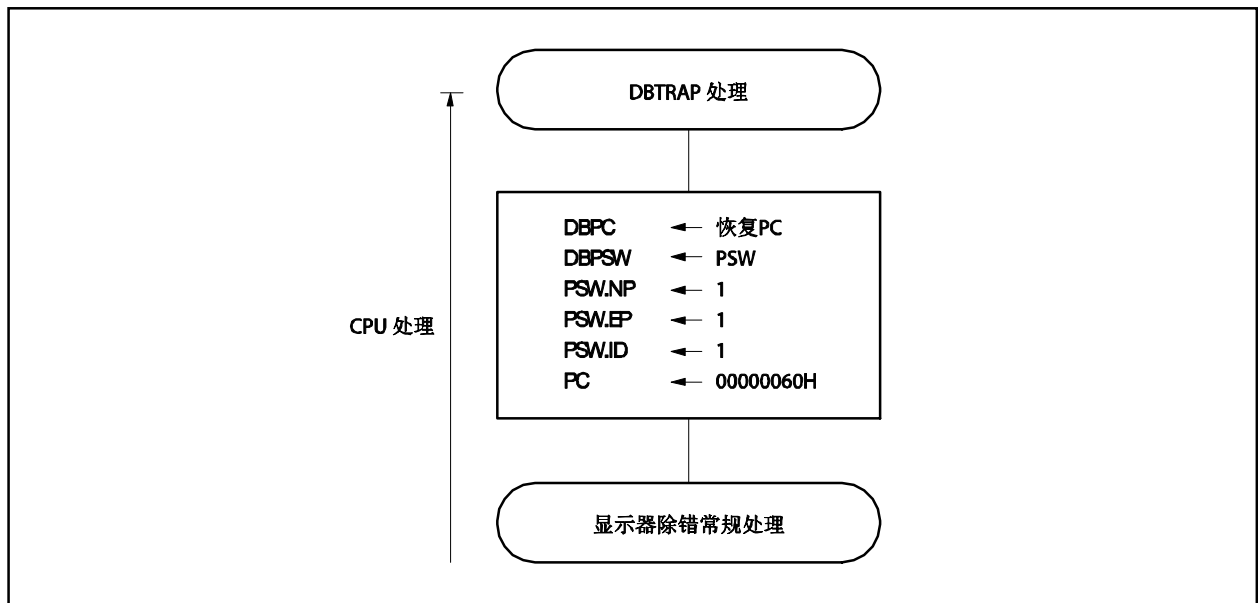
调试陷阱是一个异常，它在 DBTRAP 指令执行时产生，且总是会被确认。
在出现调试陷阱之后，CPU 将执行如下处理过程。

(1) 操作

- <1> 将恢复的 PC 保存到 DBPC。
- <2> 将当前的 PSW 保存到 DBPSW。
- <3> 将 PSW.NP 位，PSW.EP 位，和 PSW.ID 位置 1。
- <4> 在 PC 中设置相应调试陷阱的处理地址（00000060H），并且转移控制权。

图 17-12 显示了调试陷阱的处理格式。

图 17-12. 调试陷阱的处理



(2) 恢复

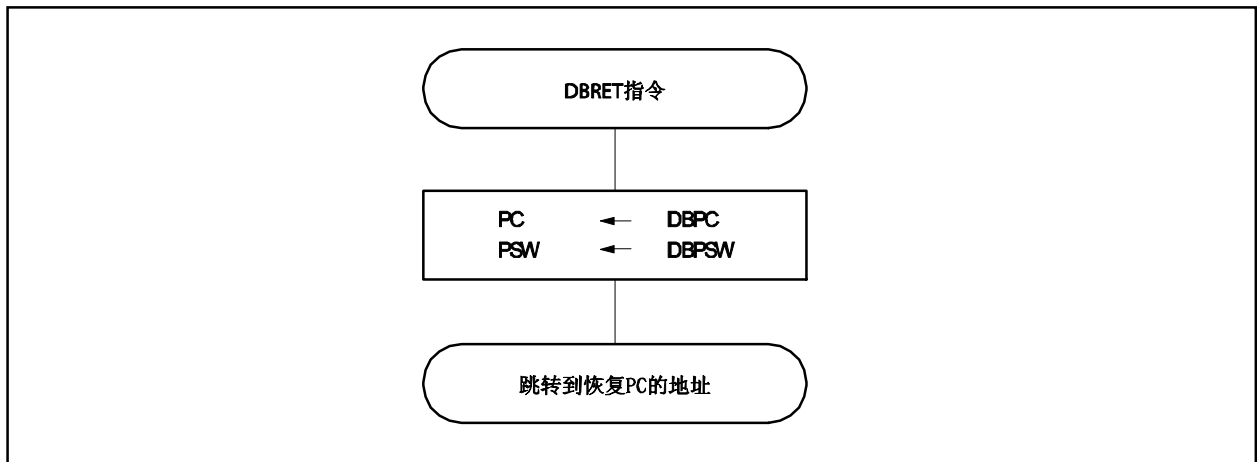
通过执行 DBRET 指令可以从调试陷阱中恢复。执行 DBRET 指令，CPU 将执行以下步骤，并将控制权转移到恢复 PC 的地址。

- <1> 从 DBPC 和 DBPSW 中加载恢复的 PC 和 PSW。
- <2> 向恢复的 PC 和 PSW 代表的地址转移控制权。

<R> **注意事项** 在 DBTRAP 指令执行后 DBRET 指令执行之前，可以访问 DBPC 和 DBPSW。

图 17-13 显示了从调试陷阱过程中恢复的处理流程。

图 17-13. 从调试陷阱过程中恢复的处理流程



17.7 多中断服务控制

如果产生了更高优先级的中断请求信号，多中断服务控制功能可以将当前执行的中断服务例程停止，并且应答更高优先级的中断请求信号。

如果正在执行中断的过程中产生了较低优先级或同等优先级的中断请求信号，则较迟的中断请求信号将被保留。

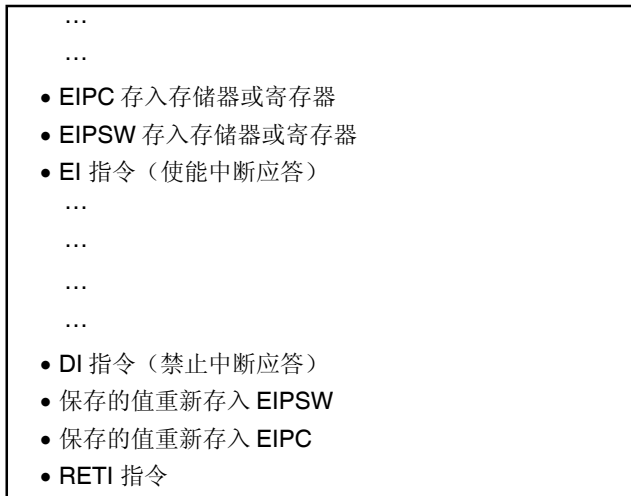
多中断服务控制将在使能中断（PSW.ID 位= 0）时执行。即使在中断服务例程中，多中断控制也必须在使能中断（ID 位 = 0）时执行。

如果可屏蔽中断或软件异常服务程序中产生了可屏蔽中断或软件异常，必须保存 EIPC 和 EIPSW。

下面举例说明该过程。

(1) 服务程序中应答可屏蔽中断请求信号

可屏蔽中断或异常服务程序



←应答可屏蔽中断

(2) 在服务程序中产生异常

可屏蔽中断或异常的服务程序



← 应答异常，如 TRAP 指令。

可在多中断服务控制中设置每个可屏蔽中断请求的优先级 0 到 7（0 最高）。要设置优先级，根据每个可屏蔽中断请求向 `xxICn.xxPRn0~xxICn.xxPRn2` 位写入数值。复位后，终端请求被 `xxICn.xxMKn` 位屏蔽，`xxPRn0~xxPRn2` 位设置优先级为 7。

可屏蔽中断优先级如下。

（高） 0 级 > 1 级 > 2 级 > 3 级 > 4 级 > 5 级 > 6 级 > 7 级（低）

由于多中断服务控制而被保留的中断服务将在更高优先级的中断服务完成且 `RETI` 指令执行后继续。保留的中断请求信号在当前中断服务完成且 `RETI` 指令执行后应答。

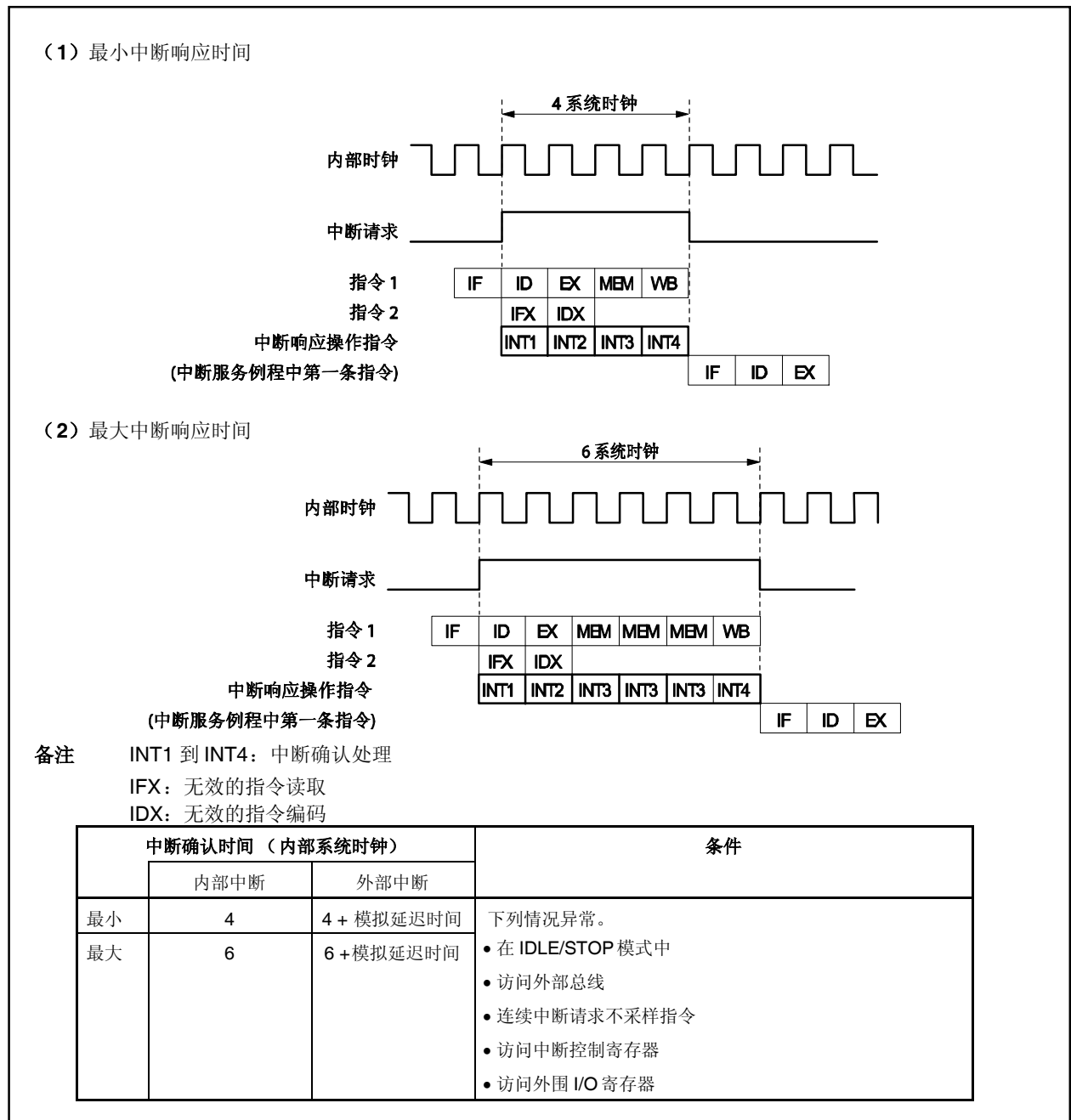
注意事项 在非屏蔽中断服务例程中（到 `RETI` 指令执行为止）可屏蔽中断不被应答并始终保持。

17.8 中断响应时间

除非出现下列情况，CPU 的中断响应时间最少需要 4 个时钟周期。如果连续输入中断请求信号，中断请求信号最少应该间隔 4 个时钟。

- IDLE/STOP 模式
- 当访问外部总线时
- 连续中断请求不采样指令（参考 17.9 CPU 不对中断进行确认的时期）
- 当访问中断控制寄存器时
- 当访问外围 I/O 寄存器时

图 17-14. 中断请求信号应答的流水线操作（概图）



17.9 CPU 不对中断进行确认的时期

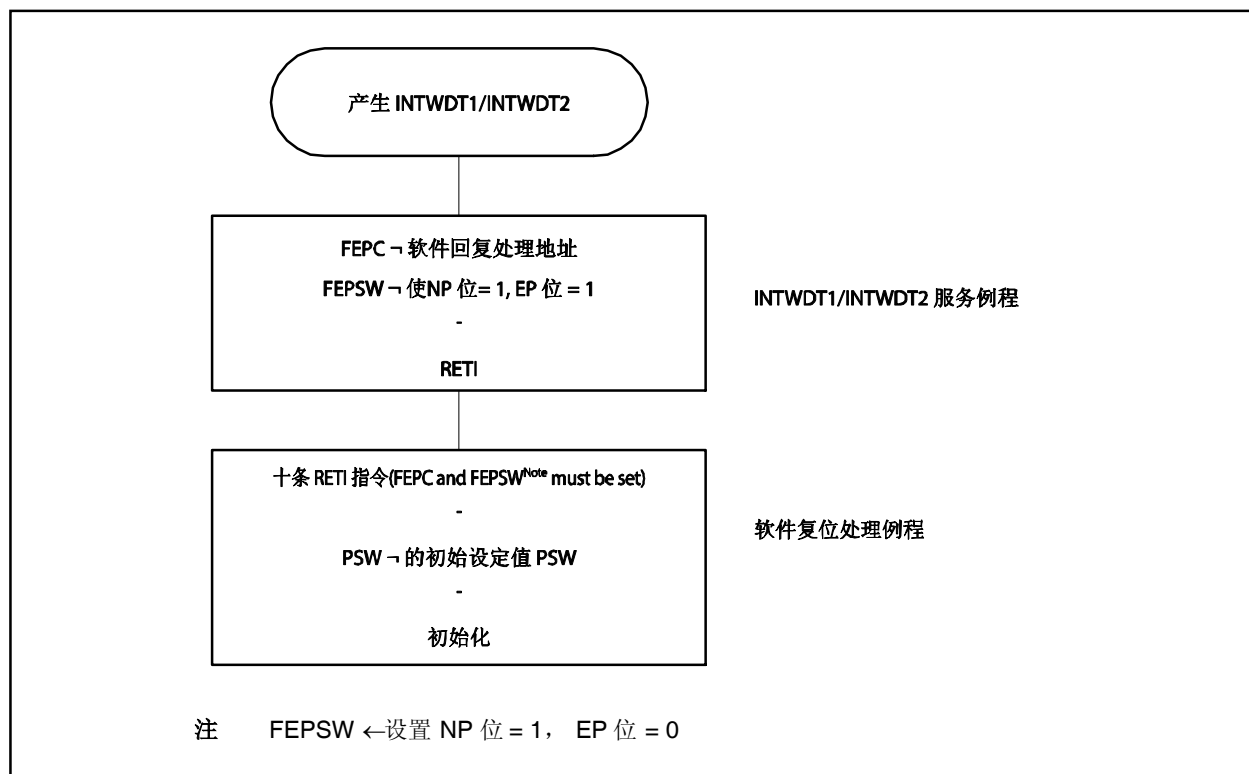
当执行指令时，中断会通过 CPU 确认。但是，在两个中断请求不采样指令之间（中断被保持）。中断不采样信号指令如下。

- EI 指令
- DI 指令
- PSW 的 LDSR reg2, 0x5 指令
- PRCMD 寄存器的存储指令
- 以下寄存器的存储，SET1, NOT1 和 CLR1 指令。
 - 有关中断的寄存器：
 - 中断控制寄存器 (xxICn)，中断屏蔽寄存器 0, 1, 3 (IMR0, IMR1, IMR3)
 - 节能控制寄存器 (PSC)

17.10 注意事项

设计这个系统，从而使非屏蔽中断请求信号 (INTWDT1/INTWDT2) 触发的非屏蔽中断在服务后，能够通过 RETI 指令恢复。

图 17-15. 通过 RETI 指令恢复



第十八章 按键中断功能

18.1 功能

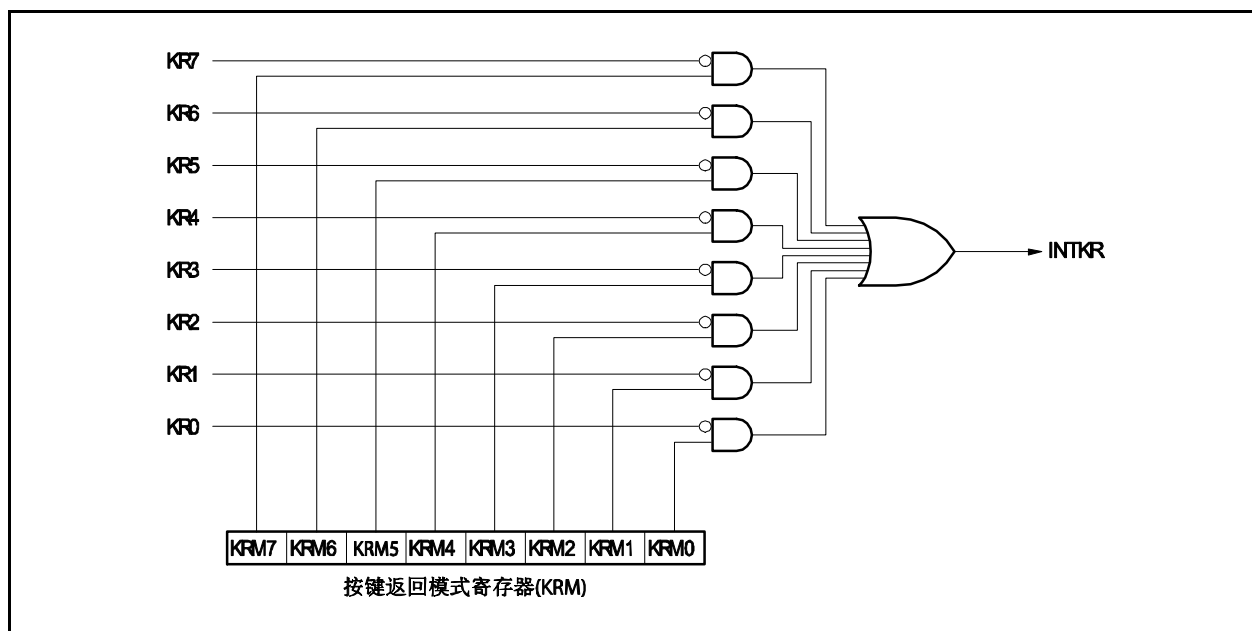
通过设置 KRM 寄存器，给八个按键输入引脚（KR0 ~KR7）输入下降沿，可以产生按键中断请求信号（INTKR）。

注意事项 如果低电平输入给 KR0 到 KR7 中的任何一个，那么即使有下降沿输入到另一引脚时，INTKR 信号也将不会产生。

表 18-1. 按键返回检测引脚的任务

标志	引脚描述
KRM0	位选控制 KR0 信号
KRM1	位选控制 KR1 信号
KRM2	位选控制 KR2 信号
KRM3	位选控制 KR3 信号
KRM4	位选控制 KR4 信号
KRM5	位选控制 KR5 信号
KRM6	位选控制 KR6 信号
KRM7	位选控制 KR7 信号

图 18-1. 按键返回框图



18.2 寄存器

(1) 按键返回模式寄存器 (KRM)

KRM 寄存器使用 KR0 ~ KR7 信号控制 KRM0 ~ KRM7 位。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFFF300H

	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	按键返回模式的控制
0	不检测按键返回
1	检测按键返回信号

注意事项 如果 KRM 寄存器被改变了, 一个中断请求信号(INTKR)就将可能产生。为避免此种情况发生, 在屏蔽中断 (DI) 后改变 KRM 寄存器, 然后在清零中断请求标志后 (KRIC.KRIF 位)使中断起效。

备注 对于复用功能引脚设置, 参见表 4-12 使用端口引脚作为复用功能引脚。

第十九章 待机功能

19.1 概述

通过结合使用待机模式并选择合适的应用模式，可以有效的降低系统功耗。

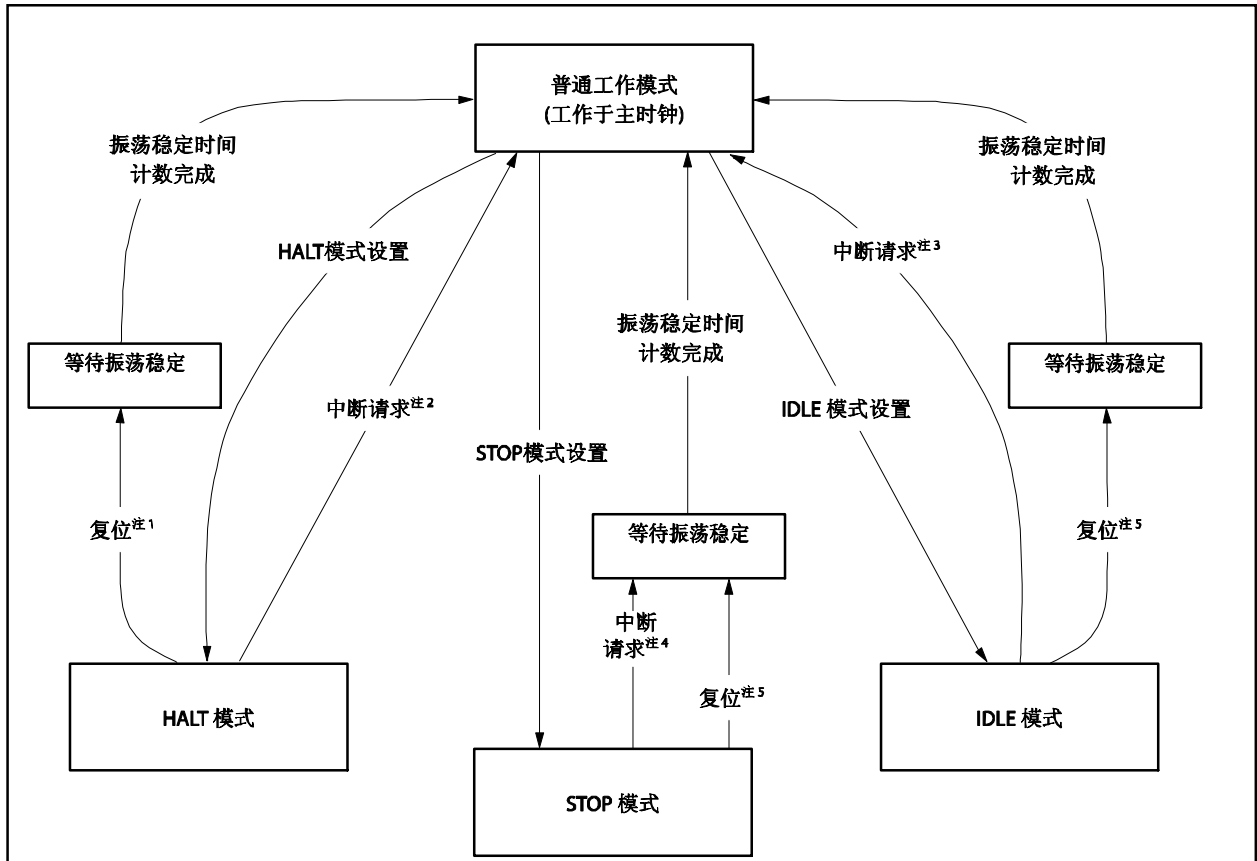
可用到的待机模式在表 19-1 中列出。

表 19-1. 待机模式

模式	功能概要
HALT 模式	此模式仅停止 CPU 的操作时钟
IDLE 模式	此模式停止除振荡器外所有内部电路的操作 ^{注 1}
STOP 模式	此模式停止除副时钟振荡器外所有内部电路的操作 ^{注 2}
副时钟操作模式	此模式副时钟被用作内部系统时钟
副 IDLE 模式	此模式停止除了振荡器之外的全部片内操作，工作在副时钟操作模式下

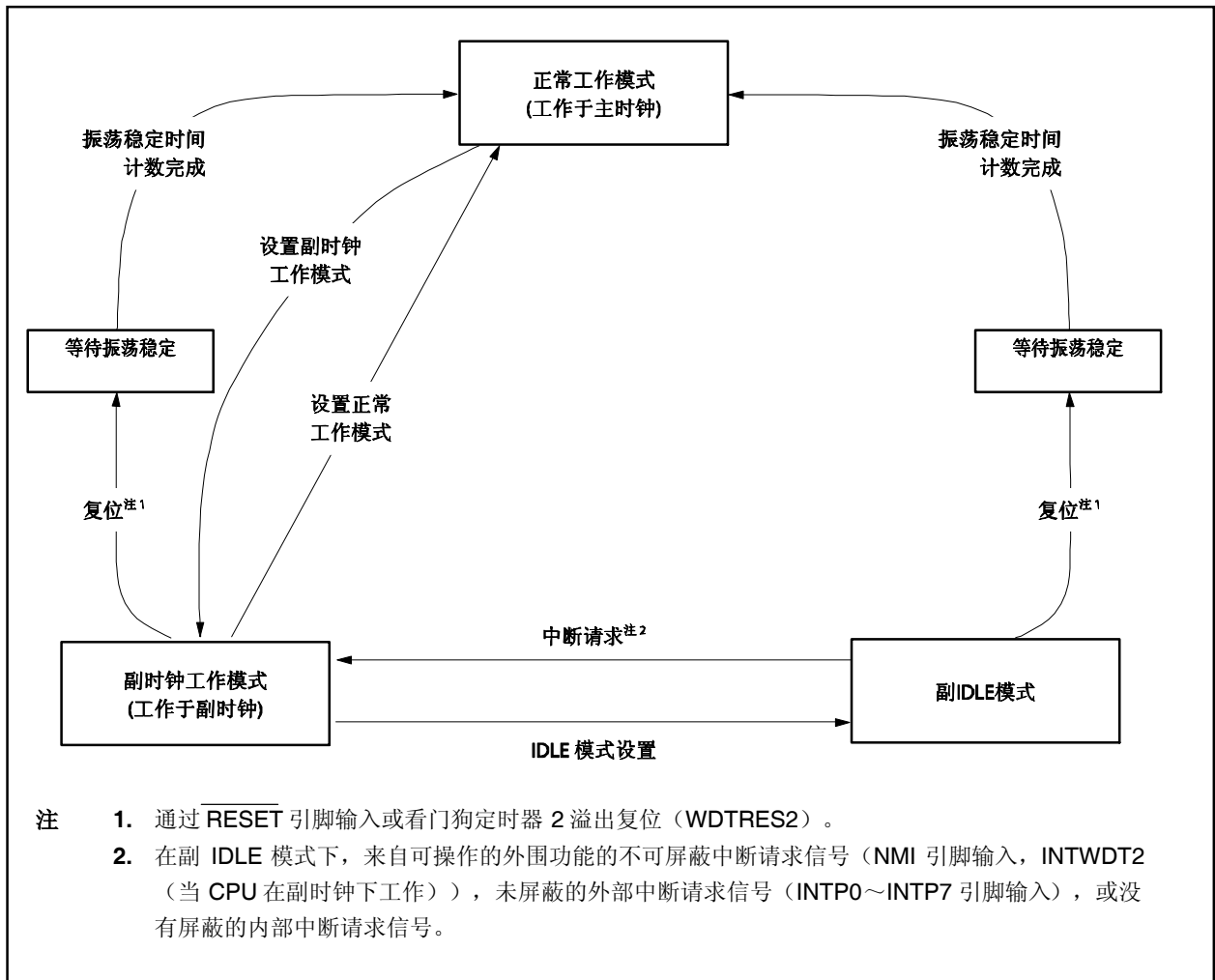
- 注
1. PLL 不停止。停止 PLL，转换到 IDLE 模式以减少能耗。
 2. 转入时钟直连模式，停止 PLL，然后再转入 STOP 模式。关于更多细节，敬请参阅第五章 时钟生成功能。

图 19-1. 状态转换 (1/2)



- 注
1. 通过 RESET 引脚输入进行复位，看门狗定时器 1 溢出 (WDTRES1)，或看门狗定时器 2 溢出 (WDTRES2)。
 2. 不可屏蔽中断请求信号 (NMI, INTWDT1, INTWDT2 或没有屏蔽的可屏蔽中断请求信号)。
 3. 在 IDLE 模式下，来自可操作的外围功能的不可屏蔽中断请求信号 (NMI 引脚输入, INTWDT2 (当 CPU 在副时钟下工作))，未屏蔽的外部中断请求信号 (INTP0~INTP7 引脚输入)，或没有屏蔽的内部中断请求信号。
 4. 在 STOP 模式下，来自可操作的外围功能的不可屏蔽中断请求信号 (NMI 引脚输入, INTWDT2 (当 CPU 在副时钟下工作))，未屏蔽的外部中断请求信号 (INTP0~INTP7 引脚输入)，或没有屏蔽的内部中断请求信号。
 5. 通过 RESET 引脚输入或看门狗定时器 2 (当 CPU 在副时钟下工作) 溢出复位 (WDTRES2)。

图 19-1. 状态转换 (2/2)



19.2 寄存器

(1) 节电控制寄存器 (PSC)

PSC 寄存器是控制待机功能的 8 位寄存器。此寄存器的 STP 位被用于指定 STOP 模式。此寄存器是仅能使用特殊顺序组合写入的特殊寄存器（参见 3.4.7 特殊寄存器）。

该寄存器支持 8 位或 1 位读写方式。

PSC 寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFFF1FBH

	< 7 >	6	< 5 >	< 4 >	3	2	< 1 >	0
PSC	NMI2M	0	NMI0M	INTM	0	0	STP	0

NMI2M	通过INTWDT2信号来控制待机模式 ^注 的释放
0	容许通过INTWDT2信号来释放待机模式 ^注
1	禁止通过INTWDT2信号来释放待机模式 ^注

NMI0M	通过 NMI 引脚输入来控制待机模式 ^注 的释放
0	容许通过NMI引脚来释放待机模式 ^注
1	禁止通过NMI引脚来释放待机模式 ^注

INTM	通过可屏蔽中断请求信号来控制 待机模式 ^注 的释放
0	容许通过可屏蔽中断请求信号来释放待机模式 ^注
1	禁止通过可屏蔽中断请求信号来释放待机模式 ^注

STP	待机模式 ^注 设置
0	正常模式
1	待机模式 ^注

注 在该情况下，待机模式包括 IDLE/STOP 模式，不包括 HALT 模式。

- 注意事项**
1. 如果 NMI2M, NMI0M 或 INTM 位设置为 1，而 STP 位同时也设置为 1，NMI2M, NMI0M, 或 INTM 位则变为无效。在 IDLE/STOP 模式下，如有一个未屏蔽的中断请求信号等待处理，将该中断请求信号（NMI2M, NMI0M, 或 INTM）所对应的位设置为 1，然后将 STP 位也设置为 1。
 2. 在 IDLE/STOP 模式设置下，先设置 PSMR.PSM 位然后设置 STP 位。

(2) 节电模式寄存器 (PSMR)

PSMR 寄存器是一个 8 位寄存器，其控制待机模式中的操作状态和时钟操作。
 该寄存器支持 8 位或 1 位读写方式。
 PSMR 该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF820H

	7	6	5	4	3	2	1	<0>
PSMR	XTSTP	0	0	0	0	0	0	PSM

XTSTP	使用副时钟振荡器的规格
0	使用副时钟振荡器
1	不使用副时钟振荡器

PSM	待机模式操作的规格
0	IDLE 模式
1	STOP 模式

- 注意事项
1. 在连接副时钟振荡器时，一定要将 XTSTP 位清零。
 2. 一定要将 PSMR 寄存器的位 1~6 清零。
 3. 仅当 PSC.STP 位为 1 时 PSM 位有效。

(3) 振荡稳定时间选择寄存器 (OSTS)

通过 OSTS 寄存器控制 STOP 模式释放后到振荡稳定的等待时间。

OSTS 寄存器支持 8 位读写方式。

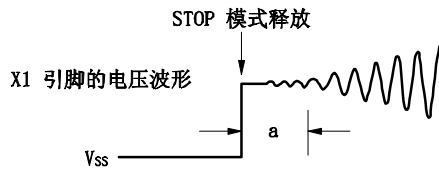
OSTS 寄存器复位后的值为 01H。

复位后: 01H R/W 地址: FFFFF6C0H

	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		
				fx		
				4 MHz	5 MHz	10 MHz
0	0	0	$2^{13}/f_x$	2.048 ms	1.638 ms	0.819 ms
0	0	1	$2^{16}/f_x$	8.192 ms	6.554 ms	3.277 ms
0	1	0	$2^{16}/f_x$	16.38 ms	13.11 ms	6.554 ms
0	1	1	$2^{17}/f_x$	32.77 ms	26.21 ms	13.11 ms
1	0	0	$2^{18}/f_x$	65.54 ms	52.43 ms	26.21 ms
1	0	1	$2^{19}/f_x$	131.1 ms	104.9 ms	52.43 ms
1	1	0	$2^{20}/f_x$	262.1 ms	209.7 ms	104.9 ms
1	1	1	$2^{21}/f_x$	524.3 ms	419.4 ms	209.7 ms

注意事项 1. STOP 模式释放后的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），不管通过复位输入或中断请求信号的产生是否释放了 STOP 模式。



2. 确定将第 3 位到 第 7 位清零。
3. 复位释放后的振荡稳定时间是 $2^{15}/f_x$ （因为 OSTS 寄存器的初始值= 01H）。
4. 振荡稳定时间也是在外部时钟输入时写入。

备注 fx: 主时钟振荡频率

19.3 HALT 模式

19.3.1 设置和操作状态

当在正常操作模式中执行专用指令（HALT）时，HALT 模式被设置。

在 HALT 模式中，时钟振荡器继续操作。仅停止对 CPU 的时钟供应，继续提供给其它内置外围设备功能的时钟供应。

结果，程序停止执行，并且内部 RAM 保持 HALT 模式设置之前的内容。独立于 CPU 指令处理的内置外围设备功能可继续操作。

表 19-3 显示了 HALT 模式中的操作状态。

通过使用 HALT 模式并结合正常操作模式的间歇操作可降低系统的平均电流消耗。

- 注意事项**
1. 在 HALT 指令之后插入五个或更多的 NOP 指令。
 2. 如果在执行 HALT 指令时有一个未屏蔽的中断请求被保持等待响应，则此状态被转移到 HALT 模式中，然后通过这个等待响应的中断请求，HALT 模式被立即释放。

19.3.2 释放 HALT 模式

通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT1，INTWDT2 信号），未屏蔽的可屏蔽中断请求信号，及复位信号（RESET 引脚输入，WDTRES1，WDTRES2 信号）释放 HALT 模式。

在 HALT 模式释放后，恢复正常操作模式。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 HALT 模式。

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 HALT 模式，而不管中断请求信号的优先级。如果在某个中断服务程序中设置 HALT 模式，则之后产生的中断请求信号按如下所述来执行。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 HALT 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 HALT 模式，并且响应此中断请求信号。

表 19-2. 通过中断请求信号释放 HALT 模式后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	执行对处理地址的跳转	
可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令	执行下一条指令

(2) 通过复位释放 HALT 模式

执行与正常复位操作相同的操作。

表 19-3. HALT 模式中的操作状态

HALT 模式设置		当 CPU 在主时钟下工作	
		当副时钟不用时	使用副时钟时
CPU		停止操作	
主时钟振荡器		允许振荡	
副时钟振荡器		—	允许振荡
中断控制器		可操作	
定时器 P (TMP0)		可操作	
16 位定时器 (TM01)		可操作	
8 位定时器 (TM50, TM51)		可操作	
定时器 H (TMH0, TMH1)		可操作	
钟表定时器		当主时钟输出被选作计数时钟时可操作	可操作
看门狗定时器 1		可操作	
看门狗定时器 2		当主时钟输出被选作计数时钟时可操作	可操作
串行接口	CSI00, CSI01	可操作	
	I ² C0	可操作	
	UART0, UART1	可操作	
按键中断功能		可操作	
A/D 转换器		可操作	
实时输出		可操作	
端口功能		在 HALT 模式被设置之前保持状态。	
内部数据		CPU 寄存器, 状态, 数据和全部其它的内部数据, 例如内部 RAM 的内容, 被保持为 HALT 模式设置之前的状态。	

19.4 IDLE 模式

19.4.1 设置和操作状态

通过清 0 PSMR.PSM 位设置 IDLE 模式，然后在正常操作模式下把 PSC.STP 位设置为 1。

在 IDLE 模式下，时钟振荡器继续工作但是向 CPU 或其他片上外围功能供时的时钟停止。

结果，停止程序执行，并且保持在设置 IDLE 模式之前的内部 RAM 的内容。CPU 和其它内置外围设备功能停止操作。但是，使用副时钟或外部时钟的内置外围设备功能继续操作。

表 19-5 显示了在 IDLE 模式中的操作状态。

因为 IDLE 模式停止了内置外围设备功能的操作，所以 IDLE 模式比 HALT 模式更能降低功耗。因为不停止主时钟振荡器，所以当用与释放 HALT 模式相同的方法释放 IDLE 模式时，无须等待振荡稳定时间就能恢复到正常操作模式。

注意事项 在执行往 PSC 寄存器中存入数据来设置 IDLE 模式的指令之后，插入五个或更多的 NOP 指令。

19.4.2 释放 IDLE 模式

通过来自在 IDLE 模式下可操作的外围功能的不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号（当 CPU 在副时钟下操作）），未屏蔽外部中断请求信号（INTP0 到 INTP7 引脚输入），未屏蔽内部中断请求信号，或复位（RESE 引脚输入，WDTRES2 信号（当 CPU 在副时钟下操作））释放 IDLE 模式。

在 IDLE 模式释放后，恢复正常操作模式。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE 模式，而不管中断请求信号的优先级。如果在某个中断服务程序中设置 IDLE 模式，则之后产生的中断请求信号按如下所述来执行。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 IDLE 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 IDLE 模式，并且响应此中断请求信号。

表 19-4. 通过中断请求信号释放 IDLE 模式后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	执行对处理地址的跳转	
可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令	执行下一条指令

注意事项 通过设置 PSC.NMI2M, PSC.NMI0M, 或 PSC.INTM 位为 1 来禁止中断请求信号，这样来禁止中断信号是无效的，IDLE2 模式也不会释放。

(2) 通过复位释放 IDLE 模式。

执行与正常复位操作相同的操作。

表 19-5. IDLE 模式中的操作状态

IDLE 模式设置		CPU 在主时钟下操作	
		当不使用副时钟时	当使用副时钟时
CPU		停止操作	
主时钟振荡器		允许振荡	
副时钟振荡器		-	允许振荡
中断控制器		停止操作	
定时器 P (TMP0)		停止操作	
16 位定时器 (TM01)		当 INTWT 被选作计数钟表, 及 f _{BRG} 被选作 WT 的计数时钟可以操作	当 INTWT 被选作计数钟表可操作
8 位定时器 (TM50, TM51)		<ul style="list-style-type: none"> • 当 T15m 被选作计数钟表时可操作 • 当 INTTM010 被选作计数钟表及 TM01 在 IDLE 模式下可操作 	
定时器 H (TMH0)		停止操作	
定时器 H (TMH1)		停止操作	当 f _{XT} 选作计数时钟时可操作
钟表定时器		选择主时钟为计数时钟时可操作	可操作
看门狗定时器 1		停止操作	
看门狗定时器 2		停止操作	当 f _{XT} 选作计数时钟时可操作
串行接口	CSI00, CSI01	当 SCK0m 输入时钟被选为操作时钟时可操作	
	I ² C0	停止操作	
	UART0	当 ASCK0 被选作计数时钟是可操作	
	UART1	停止操作	
按键中断功能		可操作	
A/D 转换器		停止操作 ^注	
调整器		操作继续	
实时输出		当 INTTM5m 被选作输出触发器, TM5m 在 IDLE 模式下可用时可操作。然而 RTBH0 及 RTBL0 寄存器不可升级, 因为 CPU 已停止。	
端口功能		保持设置 IDLE 模式之前的状态。	
内部数据		CPU 寄存器, 状态, 数据和全部其它的内部数据, 例如内部 RAM 的内容, 被保持为 IDLE 模式设置之前的状态。	

注 设置 ADM.ADCS 和 ADM.ADCS2 位为 00B。

备注 m = 0, 1

19.5 STOP 模式

19.5.1 设置和操作状态

在正常模式下当 PSMR.PSM 位和 PSC.STP 位设置为 1 时 STOP 模式便可设定。

在 STOP 模式中，副时钟振荡器继续操作，主时钟振荡器停止。停止给 CPU 和内置外围设备功能的时钟供应。

结果，停止程序执行，并且保持在设置 STOP 模式之前的内部 RAM 的内容。使用副时钟振荡器或外部时钟的内置外围设备功能继续操作。

表 19-7 显示了在 STOP 模式中的操作状态。

因为 STOP 模式停止了主时钟振荡器的操作，所以 STOP 模式比 IDLE 模式更能降低功耗。如果不使用副时钟振荡器，内部振荡器和外部时钟，则功耗最低仅为漏电流消耗的功耗。

注意事项 在执行往 PSC 寄存器中存入数据来设置 STOP 模式的指令之后，插入五个或更多的 NOP 指令。

19.5.2 释放 STOP 模式

可通过不可屏蔽中断请求信号（NMI 引脚输入，NTWDT2 信号（当 CPU 在副时钟下操作时）），未屏蔽的外部中断请求信号（INTP0 到 INTP7 引脚输入），为屏蔽的内部中断请求信号，来自 STOP 模式中可操作的外围设备功能的未中断请求信号，或复位信号（RESET 引脚输入，WDTRES2 信号（当 CPU 在副时钟下操作时））来释放 STOP 模式。

在 STOP 模式释放后，在振荡稳定时间被保证之后，恢复正常操作模式。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 STOP 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 STOP 模式，而不管中断请求信号的优先级。如果在某个中断服务程序中设置 STOP 模式，则之后产生的中断请求信号按如下所述来执行。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 STOP 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 STOP 模式，并且响应此中断请求信号。

表 19-6. 通过中断请求信号释放 STOP 模式后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	执行对处理地址的跳转	
可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令	执行下一条指令

注意事项 通过设置 PSC.NMI2M, PSC.NMI0M 位 或 PSC.INTM 位为 1 来禁止中断请求信号，这样来禁止中断信号是无效的，IDLE2 模式也不会释放。

(2) 通过复位释放 STOP 模式

执行与正常复位操作相同的操作。

表 19-7. STOP 模式中的操作状态

STOP 模式设置		当 CPU 在主时钟下操作	
		当不使用副时钟时	当使用副时钟时
CPU		停止振荡	
主时钟振荡器		振荡停止	
副时钟振荡器		-	允许振荡
中断控制器		停止操作	
定时器 P (TMP0)		停止操作	
16 位定时器 (TM01)		停止操作	当 INTWT 被选作计数钟表, 及 f _{XT} 被选作 WT 的计数时钟可以操作
8 位定时器 (TM50, TM51)		当 TI5m 被选作计数时钟时可操作	当 TI5m 被选作计数时钟时或 TM010 被选作计数时钟及 TM01 在 STOP 模式下可使用时可操作
定时器 H (TMH0)		停止操作	
定时器 H (TMH1)		停止操作	当 f _{XT} 被选作计数时钟时可操作
钟表定时器		停止操作	当 f _{XT} 被选作计数时钟时可操作
看门狗定时器 1		停止操作	
看门狗定时器 2		停止操作	当 f _{XT} 被选作计数时钟时可操作
串行接口	CSI00, CSI01	当 SCK0m 输入时钟被选作操作时钟时可操作	
	I ² C0	停止操作	
	UART0	当 ASCK0 被选作计数时钟是可操作	
	UART1	停止振荡	
按键中断功能		可操作	
A/D 转换器		停止操作 ^注	
实时输出		当 NTTM5m 被选作实时输出触发器和 TM5m 在 STOP 模式下可用时可操作。然而, RTBH0 和 RTBL0 不可被升级因为 CPU 已停止。	
端口功能		保持 STOP 模式设置之前的状态。	
内部数据		CPU 寄存器, 状态, 数据和全部其它的内部数据, 例如内部 RAM 的内容, 被保持为 STOP 模式设置之前的状态。	

注 设置 ADM.ADCS 位和 ADM.ADCS2 位为 00B。

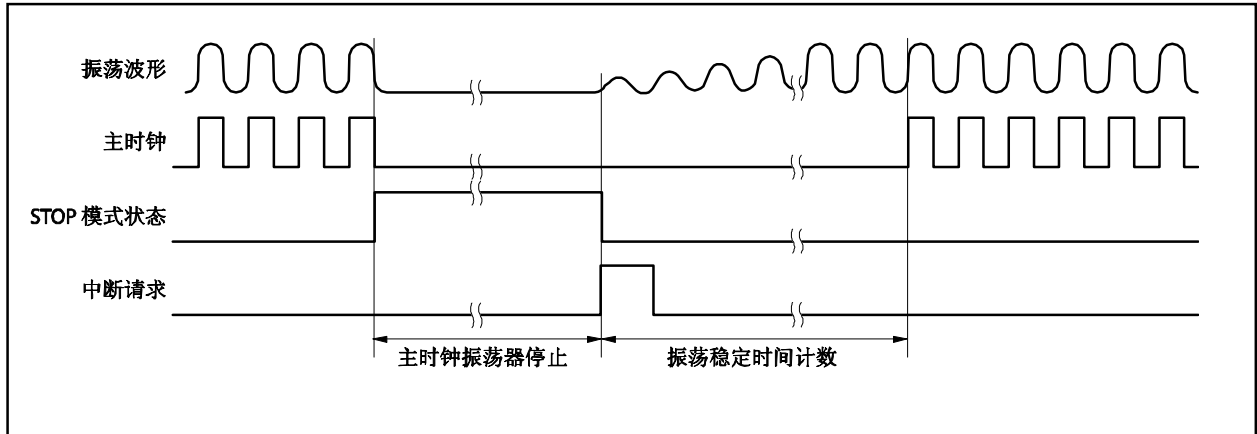
备注 m = 0, 1

19.5.3 当释放 STOP 模式时保证振荡稳定时间

当 STOP 模式释放时，只有由 OSTS 寄存器设置的振荡稳定器时间变小。如果 STOP 模式由复位释放，然而，OSTS 寄存器的复位值， $2^{15}/f_x$ ($8.192 \text{ ms } f_x = 4 \text{ MHz}$) 变小。

当 STOP 模式由下面所示的振荡请求信号释放时操作可进行。

图 19-2. 振荡稳定器时间



备注 关于 OSTS 寄存器的更多细节，敬请参阅 19.2 (3) 振荡稳定器时间选作寄存器 (OSTS)。

19.6 副时钟操作模式

19.6.1 设置和操作状态

通过在正常操作模式中设置 **PCC.CK3** 位为 **1** 来设置副时钟操作模式。

当副时钟操作模式被设置时，内部系统时钟从主时钟转换为副时钟。

当 **PCC.MCK** 位被设置为 **1** 时，主时钟振荡器的操作被停止。这样，系统仅以副时钟来操作。

表 19-8 显示了副时钟操作模式的状态。

在副时钟操作模式中，与正常操作模式相比，功耗降到一个较低的水平上，这是因为副时钟被用作内部系统时钟。除此之外，通过停止主时钟振荡器的操作，功耗能更进一步降低到 **STOP** 模式的等级。

- 注意事项**
1. 当操作 **CK3** 位时，不要改变 **PCC.CK2** 到 **PCC.CK0** 位（推荐使用位操作指令来操作）的设置值。对于 **PCC** 寄存器的细节描述，参见 5.3（1）处理器时钟控制寄存器（**PCC**）。
 2. 如果以下的条件不满足，改变 **CK2** 到 **CK0** 位的设置以便满足条件并且设置副时钟操作模式。
内部系统时钟 (f_{CLK}) > 副时钟 (f_{XT} : 32.768 kHz) × 4

备注 内部系统时钟 (f_{CLK})：通过设置位 **CK2** 到 **CK0**，可从主时钟生成时钟

19.6.2 释放副时钟操作模式

当 **CK3** 位清零或复位（**RESET** 引脚输入，**WDTRES1**，**WDTRES2** 信号）时，副时钟操作模式释放。如果主时钟停止（**MCK** 位 = **1**），设置 **MCK** 位到 **1**，通过软件保证主时钟的振荡稳定时间，将 **CK3** 位清零。

当副时钟操作模式释放后，恢复为正常操作模式。

- 注意事项** 当操作 **CK3** 位时，不要改变 **CK2** 到 **CK0** 位（推荐使用位操作指令来操作）的设置值。关于更多细节，敬请参阅 5.3（1）处理器时钟控制寄存器（**PCC**）。

表 19-8. 副时钟操作模式中的操作状态

副时钟操作模式设置 项目		操作状态	
		当主时钟振荡时	当主时钟停止时
CPU		可操作	
副时钟振荡器		允许振荡	
中断控制器		可操作	
定时器 P (TMP0)		可操作	停止操作
16 位定时器 (TM01)		可操作	当 INTWT 被选作计数钟表, 及 f _{XT} 被选作 WT 的计数时钟可以操作
8 位定时器 (TM50, TM51)		可操作	<ul style="list-style-type: none"> • 当 T15m 被选作计数时钟时可操作 • 当 INTTM010 被选作计数时钟时及 TM01 在副时钟操作模式下可使用时可操作
定时器 H (TMH0)		可操作	停止操作
定时器 H (TMH1)		可操作	当 f _{XT} 被选作计数时钟时可操作
钟表定时器		可操作	当 f _{XT} 被选作计数时钟时可操作
看门狗定时器 1		停止操作	
看门狗定时器 2		可操作	当 f _{XT} 被选作计数时钟时可操作
串行接口	CSI00, CSI01	可操作	当 SCK0m 输入时钟被选作操作时钟时可操作
	I ² C0	可操作	停止操作
	UART0	可操作	当 ASCK0 被选作计数时钟是可操作
	UART1	可操作	停止操作
按键中断功能		可操作	
A/D 转换器		可操作	停止操作
实时输出		可操作	TM5m 当 INTTM5m 被选作实时输出触发器和 T15m 被选作计数时钟。。
端口功能		可设置	
内部数据		可设置	

备注 m = 0, 1

19.7 副 IDLE 模式

19.7.1 设置和操作状态

当在副时钟模式下 PSMR.PSM 位清 0 及 PSC.STP 设置为 1 时，副 IDLE 模式可被设置。

在此模式中，时钟振荡器继续操作，停止给 CPU，flash 存储器和其它内置外围设备功能的时钟供应。

结果，停止程序执行，并且保持设置副 IDLE 模式之前的内部 RAM 的内容。停止 CPU 和其它内置外围设备功能。

然而，可使用副时钟或外部时钟操作的内置外围设备功能继续操作。

表 19-10 显示了副 IDLE 模式中的操作状态。

因为副 IDLE 模式停止了 CPU 和其它外围设备功能的操作，所以它比副时钟操作模式更能降低功耗。如果在主时钟停止后设置副 IDLE 模式，当前功耗能降低到 STOP 模式的等级。

注意事项 在执行往 PSC 寄存器中存入数据来设置副 IDLE 模式的指令之后，插入五个或更多的 NOP 指令。

19.7.2 释放副 IDLE 模式

可通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号（当 CPU 在副时钟下操作时）），未屏蔽的外部中断请求信号（INTP0 到 INTP7 引脚输入），为屏蔽的内部中断请求信号，来自副 IDLE 模式中可操作的外围设备功能的未中断请求信号，或复位信号（RESET 引脚输入，WDTRRES2 信号（当 CPU 在副时钟下操作时））来释放副 IDLE 模式。

当副 IDLE 模式由中断请求信号释放，副时钟操作模式设定，如果由复位释放，恢复正常操作模式。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放副 IDLE 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放副 IDLE 模式，而不管中断请求信号的优先级。如果在某个中断服务程序中设置副 IDLE 模式，则之后产生的中断请求信号按如下所述来执行。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放副 IDLE 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放副 IDLE 模式，并且响应此中断请求信号。

表 19-9. 通过中断请求信号释放副 IDLE 模式之后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	执行对处理地址的跳转	
可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令	执行下一条指令

注意事项 通过设置 PSC.NMI2M，PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样来禁止中断信号是无效的，IDLE2 模式也不会释放。

(2) 通过复位释放副 IDLE 模式

执行与正常复位操作相同的操作

表 19-10. 在副 IDLE 模式中的操作状态

项目	副 IDLE 模式设置	操作状态	
		当主时钟振荡时	当主时钟停止时
CPU		停止操作	
副时钟振荡器		允许振荡	
中断控制器		停止操作	
定时器 P (TMP0)		停止操作	
16 位定时器 (TM01)		当 INTWT 被选作计数钟表可操作	当 INTWT 被选作计数钟表, 及 f _{XT} 被选作 WT 的计数时钟可以操作
8 位定时器 (TM50, TM51)		<ul style="list-style-type: none"> • 当 T15m 被选作计数钟表可操作 • INTTM010 被选作计数时钟时及 TM01 在副 IDLE 模式下可使用时可操作 	
定时器 H (TMH0)		停止操作	
定时器 H (TMH1)		当选择 f _{XT} 作为计数时钟时可操作	
钟表定时器		可操作	当选择 f _{XT} 作为计数时钟时可操作
看门狗定时器 1		停止操作	
看门狗定时器 2		当选择 f _{XT} 作为计数时钟时可操作	
串行接口	CSI00, CSI01	当 SCK0m 输入时钟被选作操作时钟时可操作	
	I ² C0	停止操作	
	UART0	当 ASCK0 被选作计数时钟是可操作	
	UART1	停止操作	
按键中断功能		可操作	
A/D 转换器		停止操作 ^注	
实时输出		当 INTTM5m 被选作实时输出触发器和 TM5m 被设为副 IDLE 模式的可操作条件	
端口功能		保持设置副 IDLE 模式之前的状态。	
内部数据		CPU 寄存器, 状态, 数据和全部其它的内部数据, 例如内部 RAM 的内容, 被保持为副 IDLE 模式设置之前的状态。	

注 设置 ADM.ADCS 和 ADM.ADCS2 位为 00B。

备注 m = 0, 1

第二十章 复位功能

20.1 概述

如下复位功能可用：

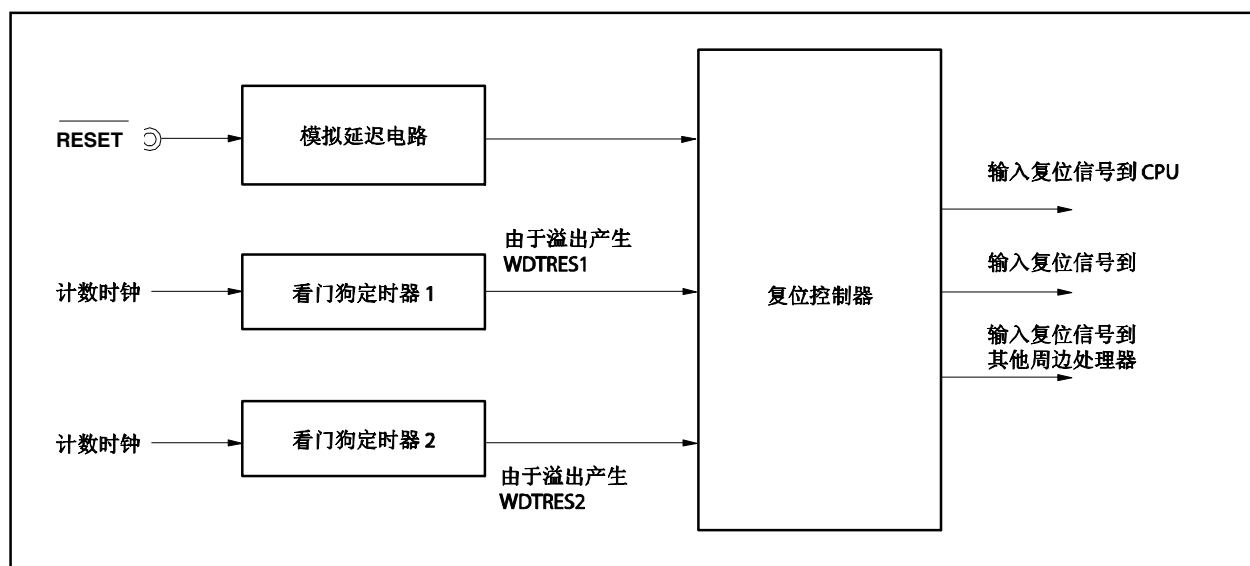
- 由 $\overline{\text{RESET}}$ 引脚输入引起的复位
- 由看门狗定时器 1 (WDTRES1) 溢出引起的复位
- 由看门狗定时器 2 (WDTRES2) 溢出引起的复位

当 $\overline{\text{RESET}}$ 引脚的电平由低变为高电平，复位状态释放，CPU 开始执行程序。在程序中有必要的对每个寄存器的内容初始化。

$\overline{\text{RESET}}$ 引脚含有一个噪声消除模块，通过模拟延时防止噪声引起的故障。

20.2 配置

图 20-1. 复位框图



20.3 操作

当 **RESET** 引脚输入一个低电平或看门狗定时器 1 或 2 溢出(WDTRES1 或 WDTRES2) 系统复位，并且所有硬件初始化。

当 **RESET** 引脚输入一个低电平，主时钟停止振荡。因此，系统总功耗可降低。

当 **RESET** 引脚的电平由低变为高或接收到 WDTRES1 或 WDTRES2 信号，复位状态释放。

如果复位状态由 **RESET** 引脚输入或 WDTRES2 信号释放，经过振荡稳定时间（OSTS 寄存器的复位值： $2^{15}/f_{xx}$ ），CPU 开始执行程序。

如果复位状态由 WDTRES1 信号释放，无法写入振荡稳定时间，因为主系统时钟振荡器不停止。

表 20-1. RESET 引脚输入后或 WDTRES2 信号产生后的硬件状态

项目	复位期间	复位后
主时器(f_x)钟振荡	停止振荡	开始振荡
副时钟振荡器(f_{XT})	振荡继续	
外围时钟(f_{xx} to $f_{xx}/1024$)	停止操作	安全振荡稳定时间之后开始振荡
内部系统时钟(f_{CLK})	停止操作	安全振荡稳定时间之后开始操作(初始化为 $f_{xx}/8$)
CPU 时钟(f_{CPU})	停止操作	安全振荡稳定时间之后开始操作(初始化为 $f_{xx}/8$)
看门狗定时器 1 时钟(f_{XW})	停止操作	开始工作
CPU	初始化	安全振荡稳定时间之后开始执行程序
内部 RAM	如果上电复位或者 CPU 通路河复位输入冲突(数据被破坏)则不确定, 其余值在复位后则立即保留	
I/O 引脚	高阻抗	
片上外围 I/O 寄存器	初始化为指定状态	
看门狗定时器 2	停止操作	安全振荡稳定时间之后开始操作
其他片上外围功能	停止操作	安全振荡稳定时间之后可以开始操作

表 20-2. WDTRES1 信号产生后的硬件状态

项目	复位期间	复位后
主时钟振荡器(f_x)	振荡继续	
副时钟振荡器 (f_{XT})	振荡继续	
外围时钟(f_{xx} to $f_{xx}/1024$)	操作停止	操作开始
内部系统时钟(f_{CLK})	振荡继续(初始化为 $f_{xx}/8$)	
CPU 时钟(f_{CPU})	振荡继续(初始化为 $f_{xx}/8$)	
看门狗定时器 1 时钟(f_{XW})	操作继续	
内部 RAM	如果 CPU 向 RAM 写数据与复位输入冲突 (数据损坏) 则不确定, 否则复位前的最终数值被保留。	
I/O 线	高阻抗	
片上外围 I/O 寄存器	初始化为指定状态	
看门狗定时器 2	操作停止	操作开始
其他片上外围功能	操作停止	操作可以开始

图 20-2. RESET 引脚输入后的硬件状态

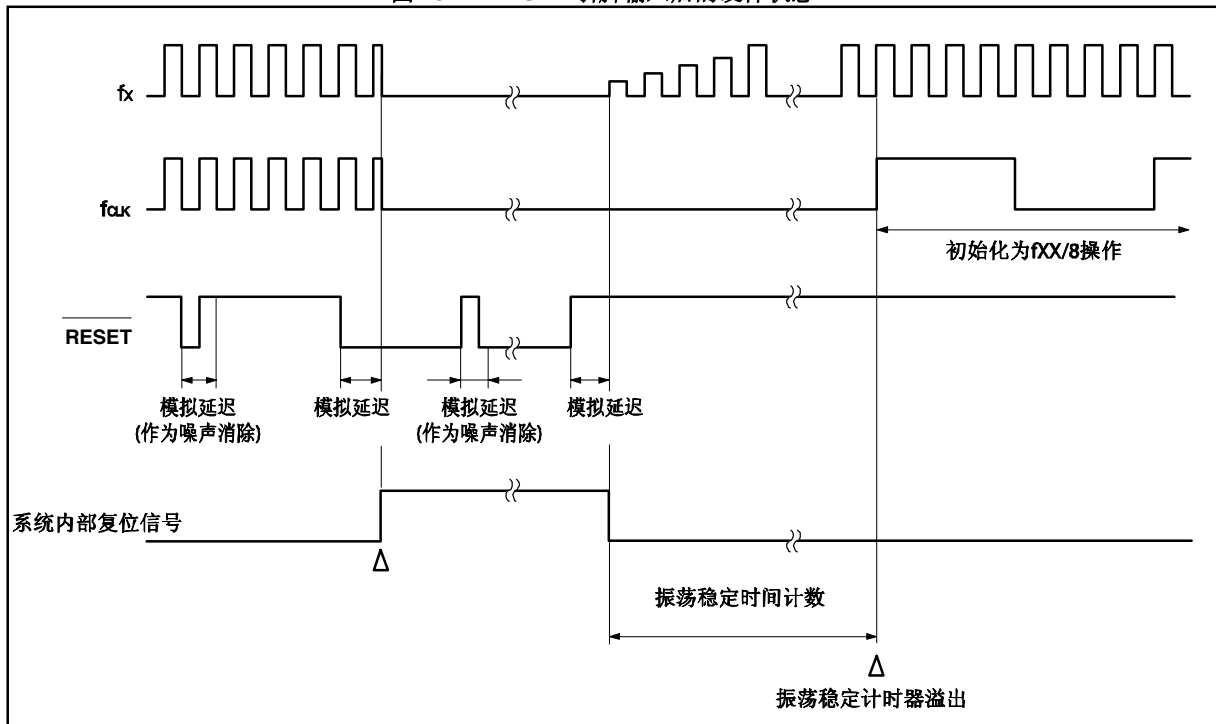


图 20-3. 电源应用操作

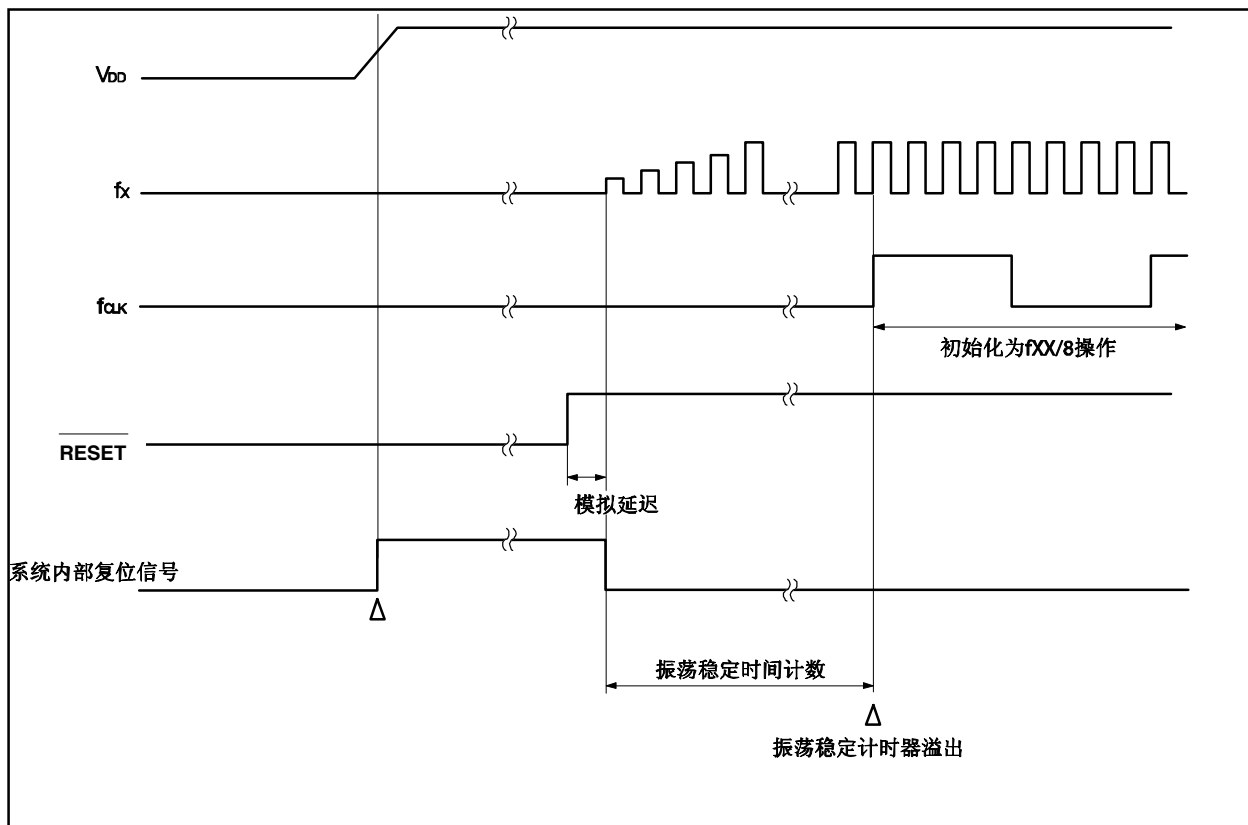


图 20-4. 由看门狗定时器 1 引起的复位操作时序

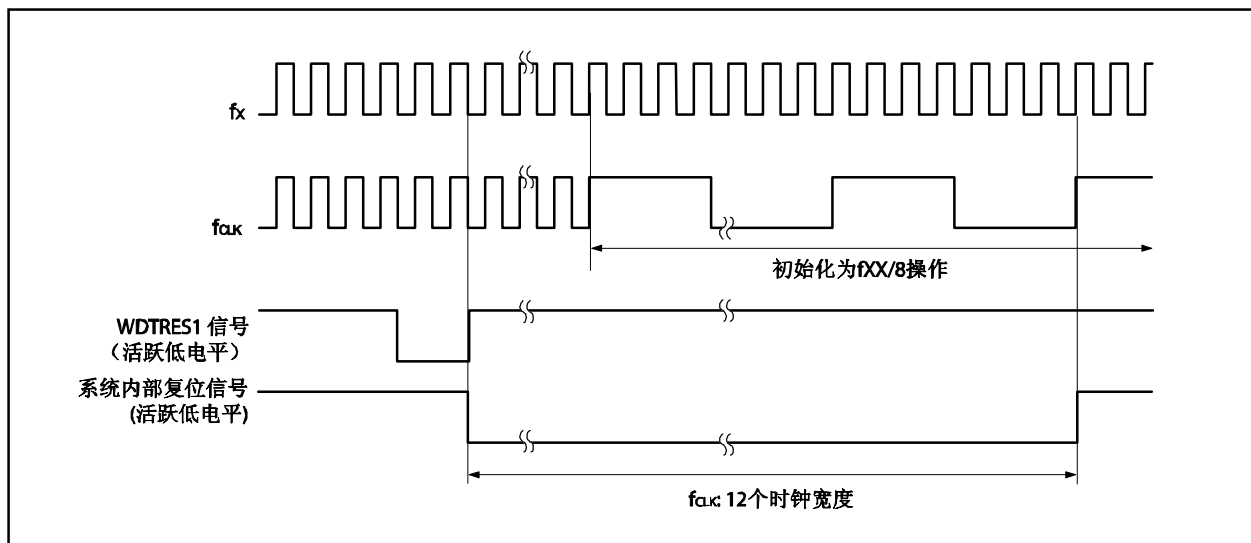
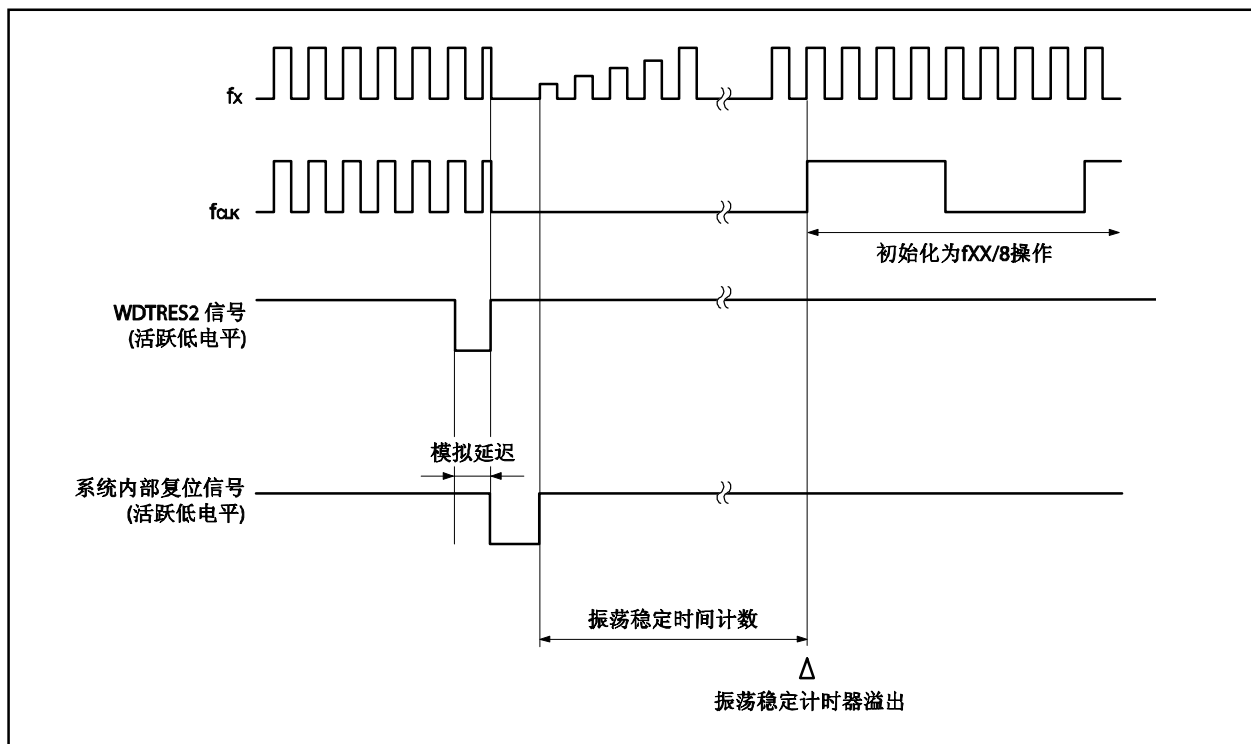


图 20-5. 由看门狗定时器 2 引起的复位操作时序



第二十一章 flash 存储器

注意事项 要获得关于flash存储器重写入的电气特性，敬请参阅第二十三章 电气特性。

flash存储器的版本一般在以下开发环境中以及大批量生产应用时使用。

- 在将V850ES/KE2焊接到目标系统上改写软件。
- 在开始大批量生产时的数据调整。
- 根据各种型号的小规模生产中的规格细分软件。
- 辅助存货管理。
- 在装运后升级软件。

21.1 特点

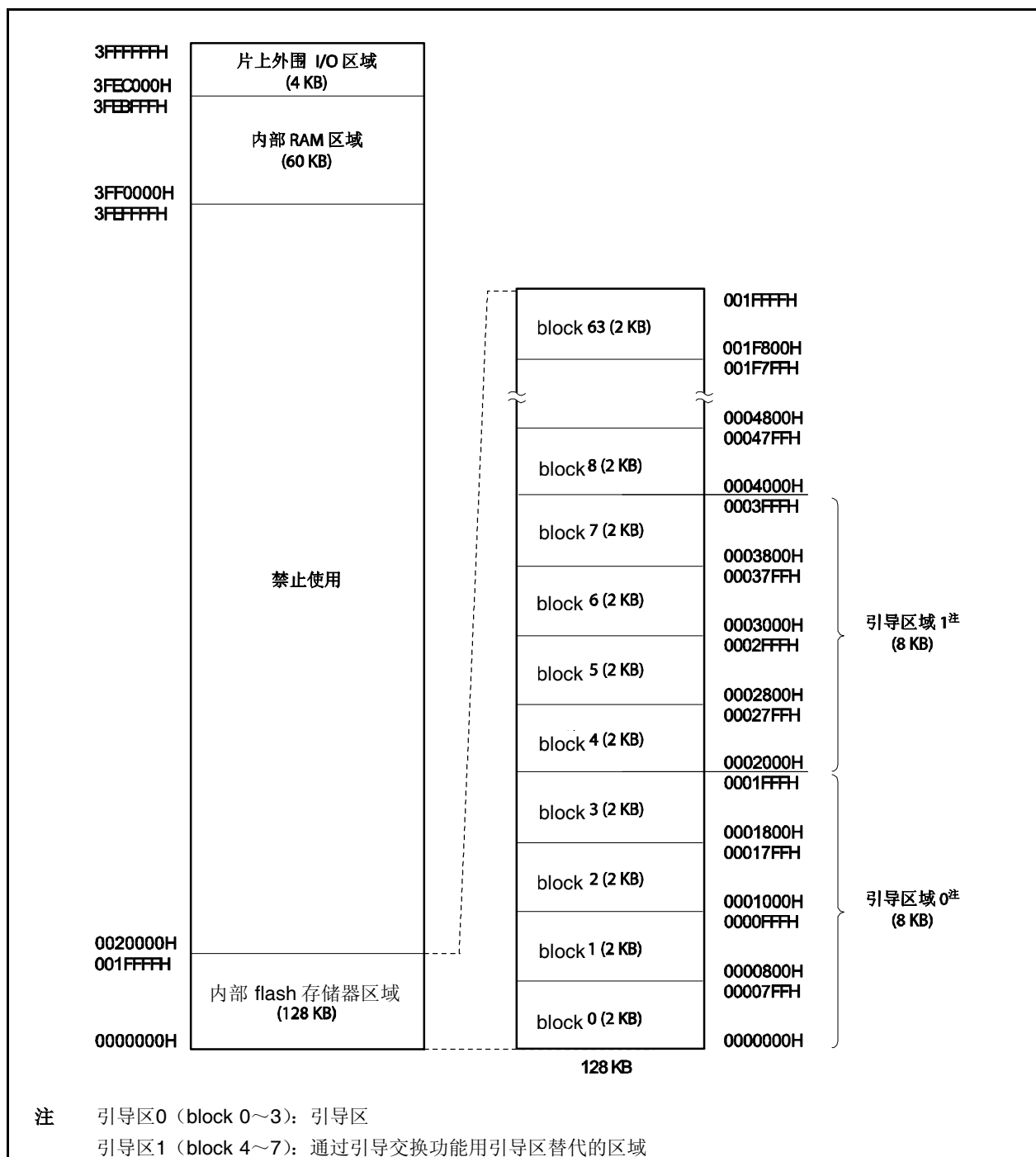
- 4字节/1时钟的访问（当取出指令时）
- 容量：128 KB
- 写入电压：用单独的电源擦除/写入
- 重写方法
 - 通过串行接口与专用flash存储器编程器通讯来进行重写入（片上/片下编程）
 - 通过用户程序重新写入flash存储器（安全功能）
- 支持flash存储器写禁止功能（安全功能）
- 使用引导交换功能通过自编程安全地重新写入到整个flash存储器区域
- 可以在自编程过程中确认中断

21.2 存储器配置

128 KB 内部flash存储器区被分为64个block，可以block为单位编程/擦除。也可以一次擦除所有的block。

当使用引导交换功能时，位于引导区0地址的物理存储器（block 0~3）被引导区1地址的物理存储器（block 4~7）替代。关于引导交换功能的更多细节，敬请参阅21.5 通过自编程重写。

图21-1. flash存储器映射



21.3 功能概述

不论V850ES/KE2是否已经被安装到目标系统（片上编程/片下编程），可以用专用flash存储器编程器的重写功能重新写入V850ES/KE2的内部flash存储器。

同时还支持禁止重写已经写入到内部flash存储器的用户程序的安全功能，这样未经授权的人就无法改动程序了。

使用用户程序（自编程）进行重写的功能适用于假设程序在目标系统生产/运输后改变的情况。同时还支持重新安全地重写整个flash存储器区域的引导交换功能。此外，在自编程过程中还支持中断服务，因此可以在各种条件下重写flash存储器，如在与外部设备通讯过程中。

表21-1. 重写方法

重写方法	功能概述	操作方法
片上编程	通过使用专用flash存储器编程器，可以在设备安装到目标系统后重写flash存储器。	Flash存储器编程模式
片下编程	通过使用专用flash存储器编程器和专用程序适配板（FA系列），可以在设备安装到目标系统前重写flash存储器。	
自编程	可以执行事先通过片上/片下编程写入到flash存储器的用户程序重写flash存储器。（在自编程的过程中，无法从内部flash存储器区域取出指令或无法访问内部flash存储器的数据。因此，必须提前将重写程序传输到内部RAM或外部存储器）。	正常操作模式

备注 FA系列是Naito Densei Machida Mfg. Co., Ltd.的产品。

表21-2. 基本功能

功能	功能概述	支持 (○: 支持, ×: 不支持)	
		片上/片下编程	自编程
block擦除	擦除指定存储器block的内容。	○	○
片擦除	一次擦除整个存储器block的内容。	○	×
写	写入到指定的地址并检查写入级是否可靠。	○	○
校验/校验和	将从存储器读取的数据与flash存储器编程器传来的数据比较。	○	× (无法通过用户程序读取)
空白检查	检查整个存储器的擦除状态。	○	○
安全设置	可以禁止使用block擦除命令、片擦除命令、编程命令、读取命令和重写引导区。	○	× (只有设置从启用改为禁用时才支持)

下表列出了安全功能。在出厂时默认启用block擦除命令禁止、片擦除命令禁止和编程命令禁止，可以通过片上/片下编程重写来设置安全。每个安全功能可以同时与其他安全功能组合使用。

表21-3. 安全功能

功能	功能概述
block擦除命令禁止	禁止在所有的block上执行block擦除命令。禁止的设置可以通过执行一个片擦除命令初始化。
片擦除命令禁止	禁止在所有的block上执行片擦除命令。在禁止设置后，由于不能执行片擦除命令，禁止的设置无法初始化。
编程命令禁止	禁止在所有的block上执行程序程序和block擦除命令。禁止的设置可以通过执行一个片擦除命令初始化。
读取命令禁止	禁止在所有的block上执行读取命令。禁止的设置可以通过执行一个片擦除命令初始化。
引导区重写禁止	可以保护从block0到指定的最后block的引导区。被保护的引导区无法重新被写入（擦除再写入）。禁止的设置可以通过执行一个片擦除命令初始化。

表21-4. 安全设置

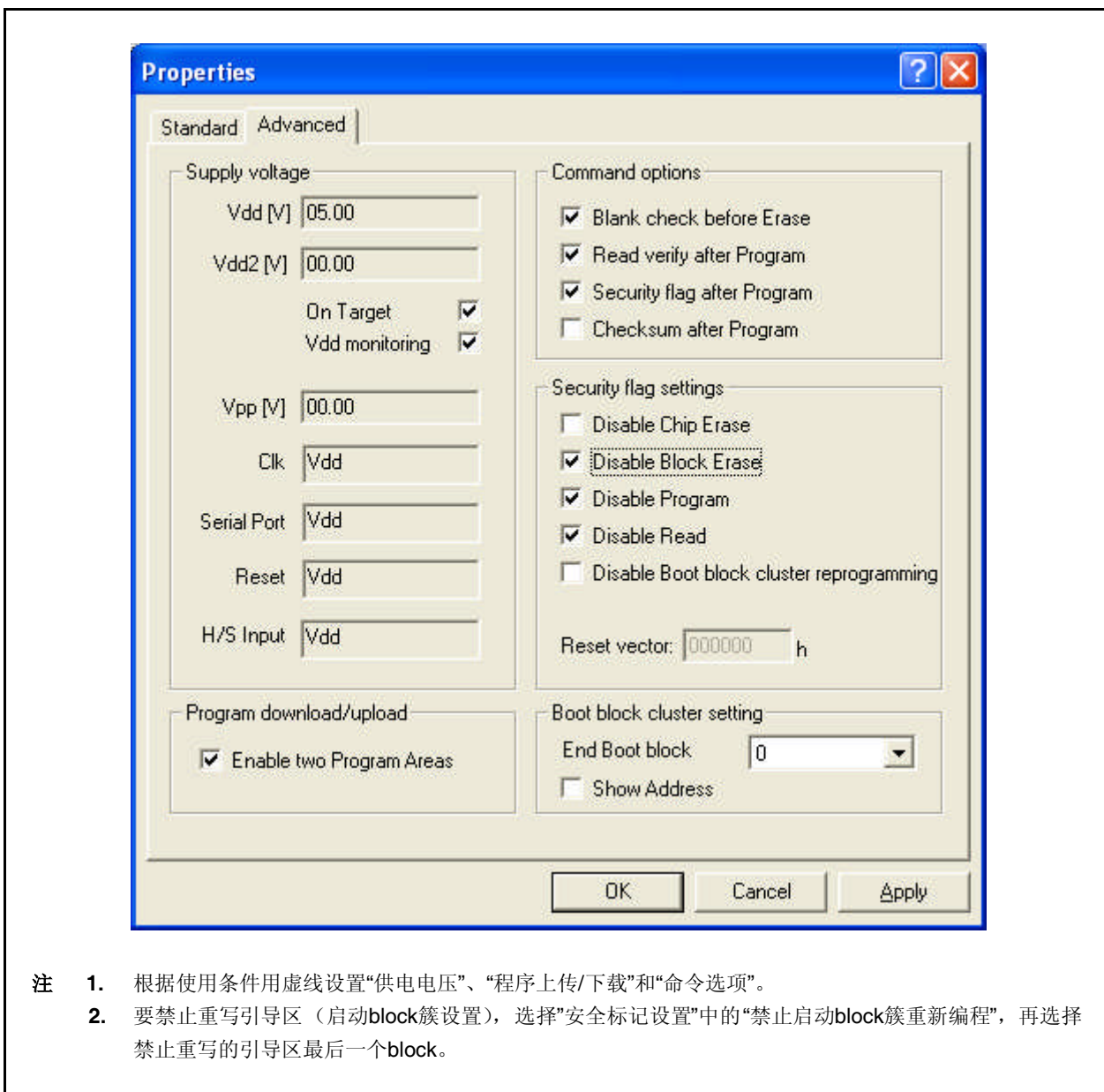
功能	当设置每个安全时的擦除、读、写操作 (√ : 可执行, × : 不可执行, - : 不支持)		关于安全设定的说明	
	片上/片下编程	自编程	片上/片下编程	自编程
block擦除命令禁止	block擦除命令: × 片擦除命令: √ 编程命令: √ 读取命令: √	block擦除 (FlashBlockErase): √ 片擦除: - 写 (FlashWordWrite): √ 读 (FlashWordRead): √	可以用片擦除命令初始化禁止的设置。	仅支持设置从允许到禁止时
片擦除命令禁止	block擦除命令: × 片擦除命令: × 编程命令: √ ^{# 1} 读取命令: √	block擦除 (FlashBlockErase): √ 片擦除: - 写 (FlashWordWrite): √ 读 (FlashWordRead): √	无法初始化禁止的设置。	
编程命令禁止	block擦除命令: × 片擦除命令: √ 编程命令: × 读取命令: √	block擦除 (FlashBlockErase): √ 片擦除: - 写 (FlashWordWrite): √ 读 (FlashWordRead): √	可以用片擦除命令初始化禁止的设置。	
读取命令禁止	block擦除命令: √ 片擦除命令: √ 编程命令: √ 读取命令: ×	block擦除 (FlashBlockErase): √ 片擦除: - 写 (FlashWordWrite): √ 读 (FlashWordRead): √	可以用片擦除命令初始化禁止的设置。	
引导区重写禁止	block擦除命令: √ ^{# 2} 片擦除命令: × 编程命令: √ ^{# 2} 读取命令: √	block擦除 (FlashBlockErase): √ 片擦除: - 写 (FlashWordWrite): √ 读 (FlashWordRead): √	无法初始化禁止的设置。	

- 注
1. 在此情况想, 由于擦除命令是无效的, 无法写入与已经写入到flash存储器的数据不同的数据。
 2. 重写被禁止的引导区是无效的。

(1) 通过PG-FP4进行安全设置（安全标记设置）

当禁止读取命令（禁止读取）时，要提高安全级别，建议同时禁止block擦除命令（禁止block擦除）和编程命令（禁止编程）。

此外，当重写程序与光罩式ROM版本不同时，还要禁止片擦除命令（禁止片擦除）。



- 注
1. 根据使用条件用虚线设置“供电电压”、“程序上传/下载”和“命令选项”。
 2. 要禁止重写引导区（启动block簇设置），选择“安全标记设置”中的“禁止启动block簇重新编程”，再选择禁止重写的引导区最后一个block。

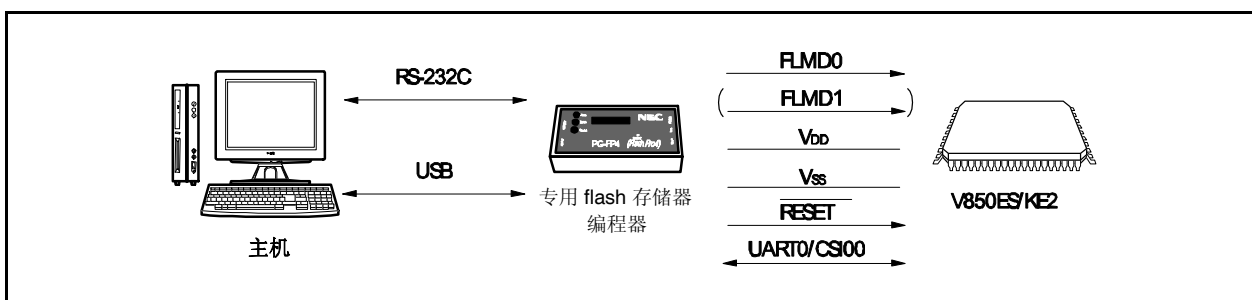
21.4 通过专用 flash 存储器编程器重写

在安装V850ES/KE2到目标系统后用专用flash存储器编程器重写flash存储器（片上编程）。也可以在安装设备到目标系统前（片下编程）用专用编程适配器（FA系列）重写flash存储器。

21.4.1 编程环境

下面显示的是写入程序到V850ES/KE2的flash存储器所要求的环境。

图21-2. 写入程序到flash存储器所要求的环境



要求用一台主机控制专用flash存储器编程器。

UART0或CSI00用于专用flash存储器编程器和执行写入、读取等操作的V850ES/KE2之间的接口。要进行片下写入，要求专用编程适配器（FA系列）。

- FA-70F3726GB-8EU-MX（已经配线）
- FA-64GB-8EU-A（没有配线：要求配线）

备注 FA系列是Naito Densai Machida Mfg. Co., Ltd.的产品。

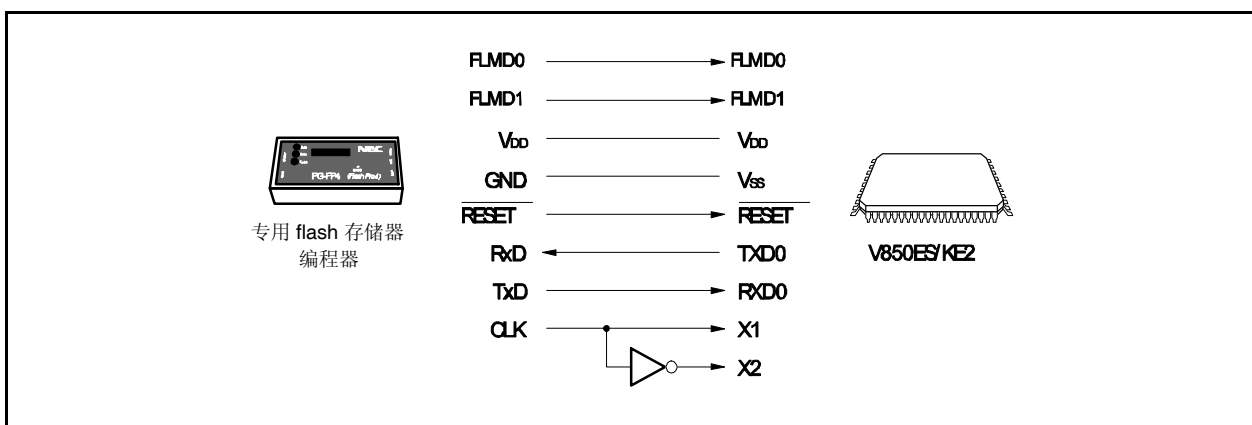
21.4.2 通讯模式

专用flash存储器编程器和V850ES/KE2之间的通讯是通过V850ES/KE2的UART0或CSI00接口的串行通讯进行的。

(1) UART0

传输速率： 9, 600, 19, 200, 31, 250, 38, 400, 76, 800, 153, 600 bps
 (不支持57, 600、115, 200或128, 000 bps的设置。)

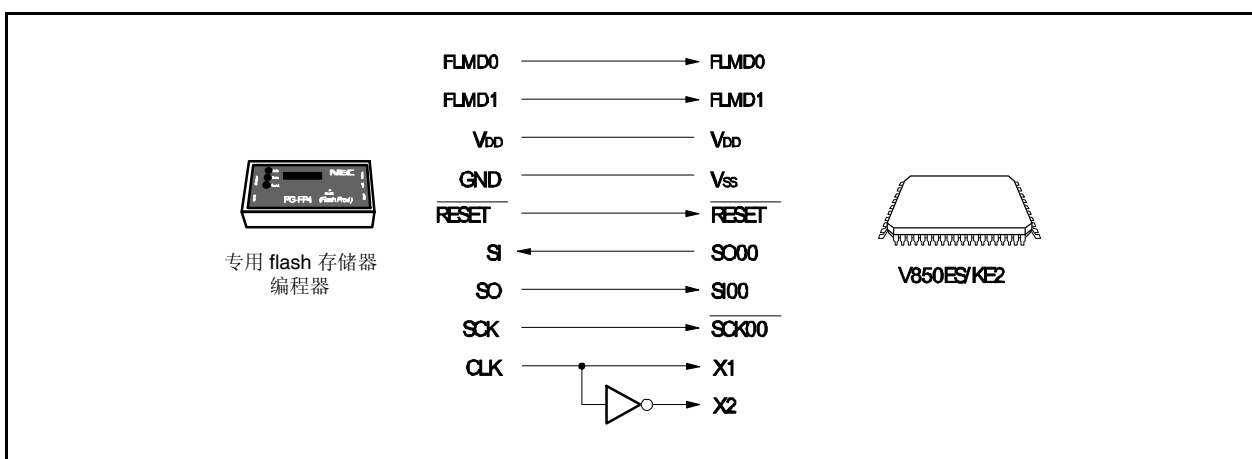
图21-3. 与专用flash存储器编程器 (UART0) 通讯



(2) CSI00

串行时钟： 2.4 kHz到2.5 MHz (最高有效位优先)

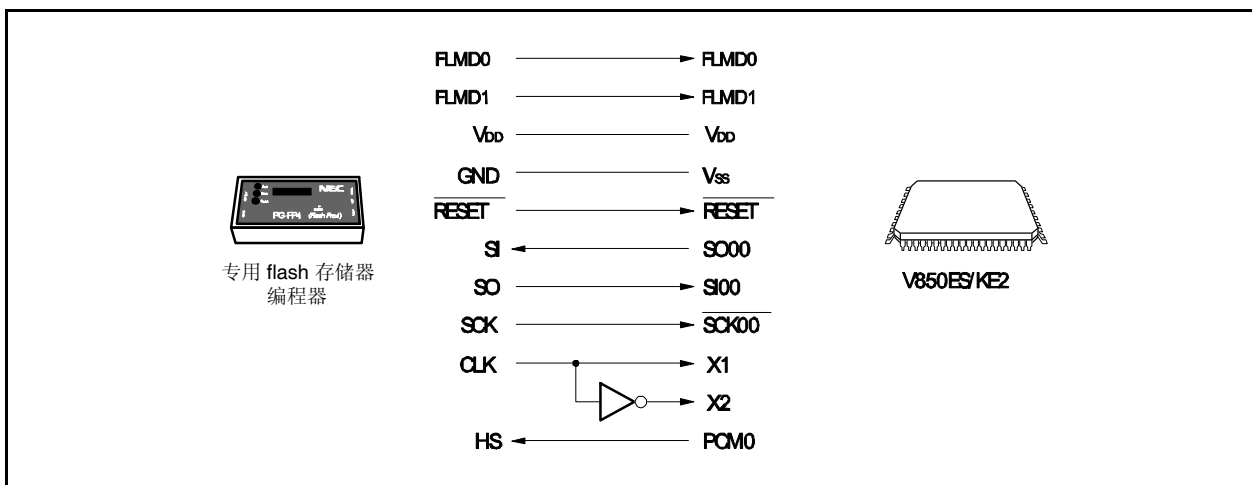
图 21-4. 与专用flash存储器编程器 (CSI00) 通讯



(3) CSI00 + HS

串行时钟： 2.4 kHz到2.5 MHz（最高有效位优先）

图21-5. 与专用flash存储器编程器（CSI00 + HS）通讯



专用flash存储器编程器输出传输时钟，V850ES/KE2作为从设备运行。

当PG-FP4用作专用flash存储器编程器时，向V850ES/KE2生成以下信号。关于更多细节，敬请参阅PG-FP4 用户手册（U15260E）。

表21-5. 专用flash存储器编程器（PG-FP4）的信号连接

PG-FP4			V850ES/KE2	连接的处理		
信号名称	输入/输出	引脚功能	引脚名称	UART0	CSI00	CSI00 + HS
FLMD0	输出	写入启用/禁止	FLMD0	◎	◎	◎
FLMD1	输出	写入启用/禁止	FLMD1	◎ ^{注1}	◎ ^{注1}	◎ ^{注1}
VDD	—	VDD 电压生成/电压监控	VDD	◎	◎	◎
GND	—	接地	VSS	◎	◎	◎
CLK	输出	输出到V850ES/KE2的时钟	X1, X2	× ^{注2}	× ^{注2}	× ^{注2}
RESET	输出	复位信号	RESET	◎	◎	◎
SI/RxD	输入	接收信号	SO00	◎	◎	◎
SO/TxD	输出	传输信号	SI00	◎	◎	◎
SCK	输出	传输时钟	SCK00	×	◎	◎
HS	输入	CSI00 + HS通讯的握手信号	PCM0	×	×	◎

- 注
1. 如图21-6，连接引脚或者通过下拉电阻连接到板上的GND。
 2. 连接这些引脚来提供来自PG-FP4的时钟（如图21-6，接线或在板上创建一个振荡器并提供时钟）。

备注

◎：必须连接。
 ×：不需要连接。

表21-6. V850ES/KE2和PG-FP4之间的接线

flash存储器编程器的引脚配置 (PG-FP4)			FA 片上的引 脚名称	与CSI00-HS通讯		与CSI00通讯		与UART0通讯	
信号名称	输入/输出	引脚功能		引脚名称	引脚编号	引脚名称	引脚编号	引脚名称	引脚编号
SI/RxD	输入	接收信号	SI	P41/SO00	20	P41/SO00	20	P30/TXD0	22
SO/TxD	输出	传输信号	SO	P40/SI00	19	P40/SI00	19	P31/RXD0/ INTP7	23
SCK	输出	传输时钟	SCK	P42/SCK00	21	P42/SCK00	21	不需要	不需要
CLK	输出	设定时钟到 V850ES/KE2	X1	X1	7	X1	7	X1	7
			X2	X2 [*]	8	X2 [*]	8	X2 [*]	8
/RESET	输出	复位信号	/RESET	RESET	9	RESET	9	RESET	9
FLMD0	输出	写入电压	FLMD0	FLMD0	3	FLMD0	3	FLMD0	3
FLMD1	输出	写入电压	FLMD1	PDL5/ FLMD1	52	PDL5/ FLMD1	52	PDL5/ FLMD1	52
HS	输入	CSI00 + HS通讯的握手信号	RESERVE /HS	PCM0	45	不需要	不需要	不需要	不需要
VDD	-	V _{DD} 电压生成/电压监控	VDD	V _{DD}	4	V _{DD}	4	V _{DD}	4
				EV _{DD}	33	EV _{DD}	33	EV _{DD}	33
				AV _{REF0}	1	AV _{REF0}	1	AV _{REF0}	1
GND	-	地	GND	V _{SS}	6	V _{SS}	6	V _{SS}	6
				AV _{SS}	2	AV _{SS}	2	AV _{SS}	2
				EV _{SS}	32	EV _{SS}	32	EV _{SS}	32

注 当使用flash存储器编程器的时钟输出时，将编程器的CLK连接到X1，然后连接其反向信号到X2。

图21-6. V850ES/KE2 flash存储器写入适配器 (FA-64GB-8EU-A) 的接线示例 (1/2)

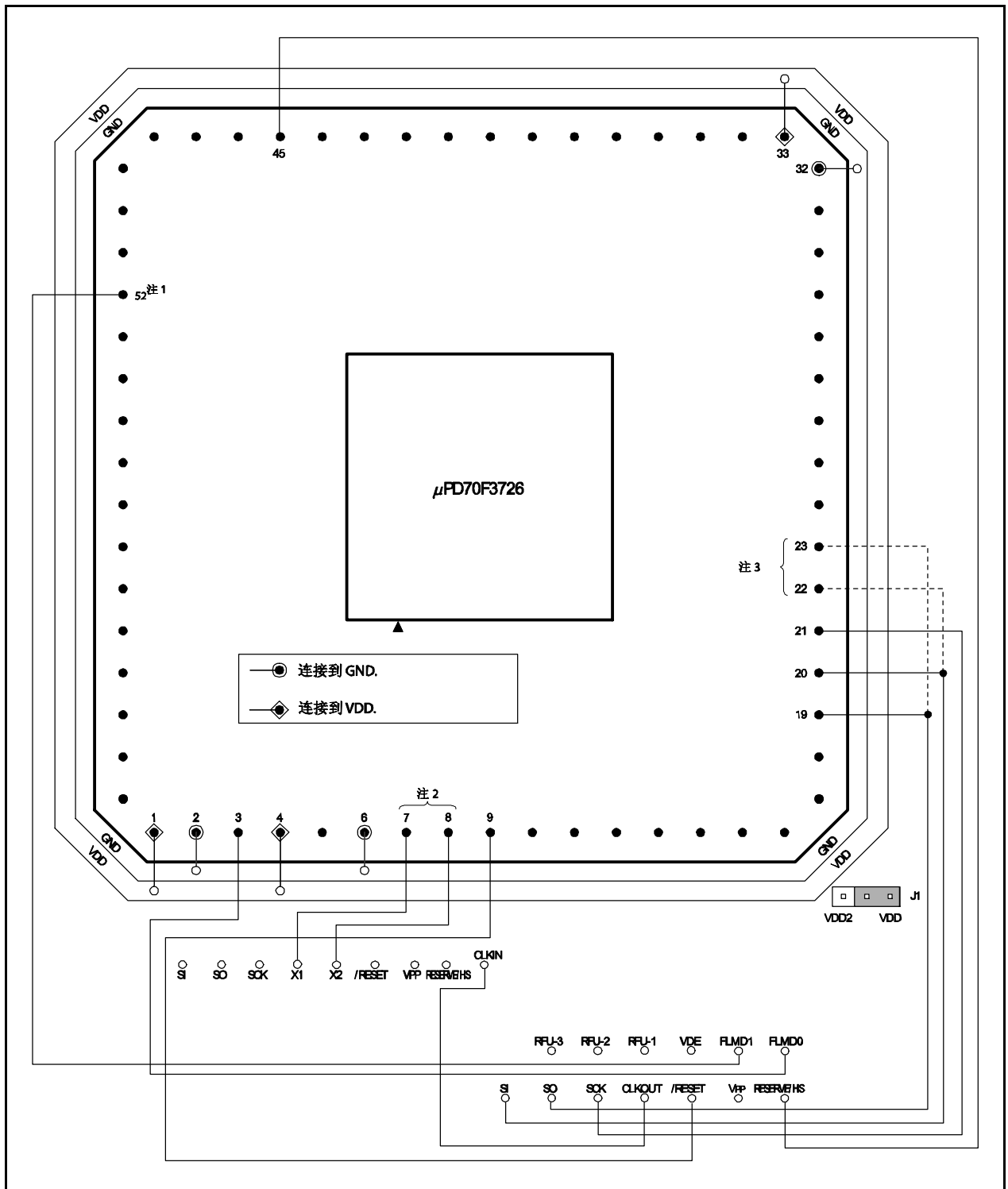
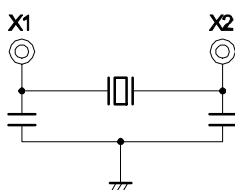


图21-6. V850ES/KE2 flash存储器写入适配器（FA-64GB-8EU-A）的接线示例（2/2）

- 注**
1. 如图所示连接FLMD1引脚或者通过下拉电阻连接该引脚到板上的GND。
 2. 上图显示的是PG-FP4提供时钟的接线示例。
确保在PG-FP4提供时钟时如下图所示设置和连接。

- 将flash存储器适配器（FA系列）的J1设到V_{DD}侧。
- 连接FA的CLKOUT到FA的CLKIN。
- 连接FA的X1到设备的X1。
- 连接FA的X2到设备的X2。

如果在flash存储器适配器上创建了一个振荡器并提供时钟，则不必进行以上的设置和连接。下面是一个电路的例子。



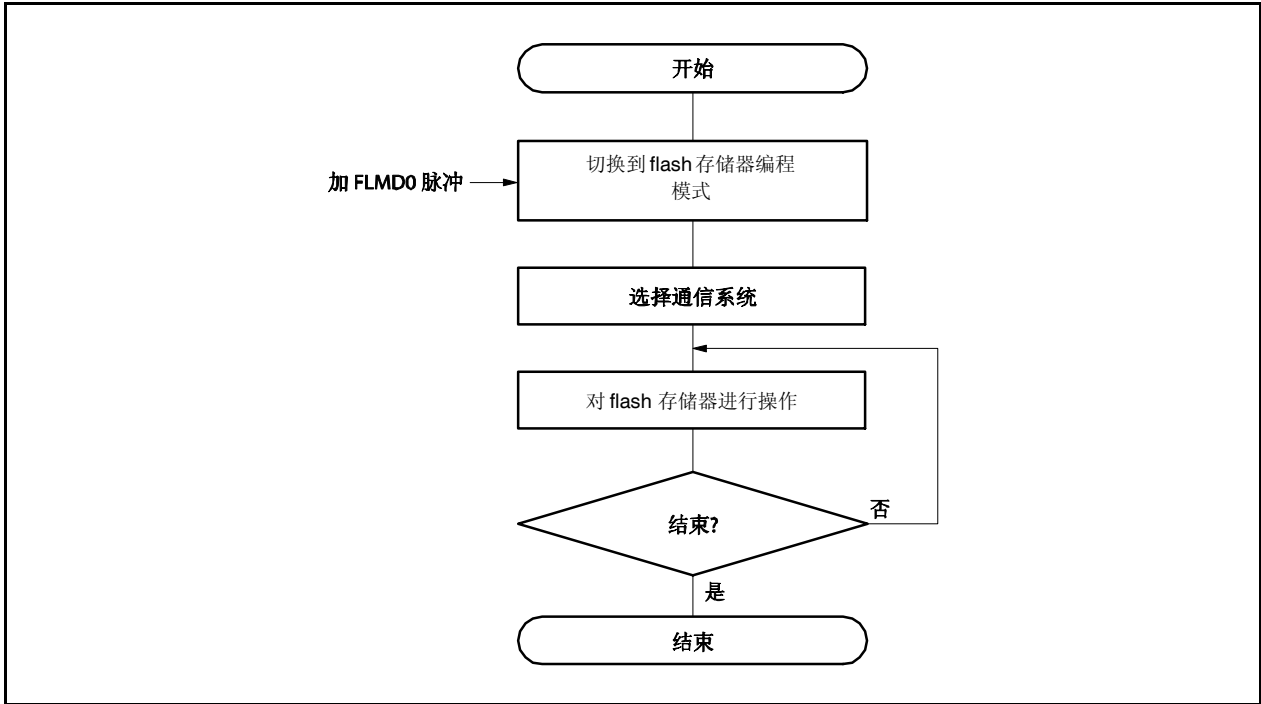
3. 当使用UART0时的通讯引脚。

- 备注**
1. 对于以上没有提到的引脚，根据规定的不使用引脚的处理进行处理。（参见2.2 I/O引脚电路和不使用的引脚的建议连接）。
当通过电阻连接到V_{DD}时，建议使用1 kΩ到10 kΩ的电阻。
 2. 此适配器用于64引脚塑料LQFP（窄节距）包。
 3. 此图显示的是使用支持握手CSI时的配线。

21.4.3 flash 存储器控制

下图显示了flash存储器控制的步骤。

图21-7. flash存储器控制的步骤

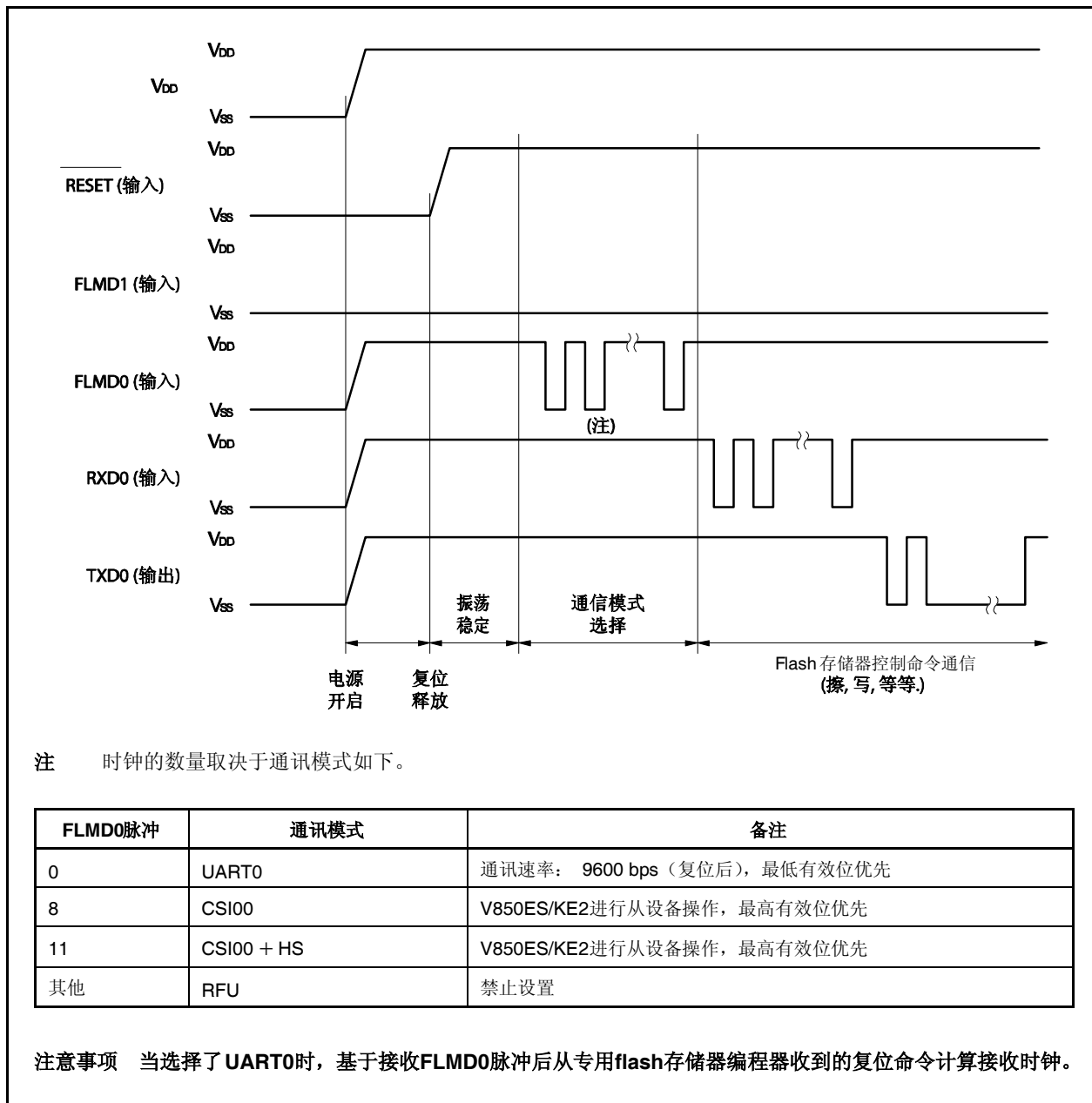


21.4.4 通讯模式的选择

在V850ES/KE2中，通讯模式是通过切换到flash存储器编程模式后输入脉冲（最多可输入11个脉冲）到FLMD0引脚。FLMD0脉冲可以由专用flash存储器编程器生成。

下图显示的是脉冲的次数和通讯模式之间的关系。

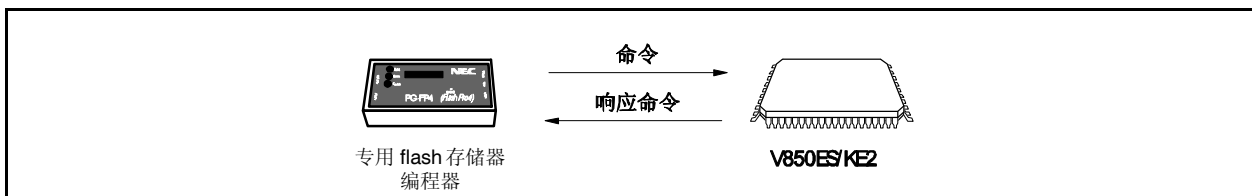
图21-8. 通讯模式的选择



21.4.5 通讯命令

V850ES/KE2通过命令的方式与专用flash存储器编程器通讯。从专用flash存储器编程器发往V850ES/KE2的信号称为信号。从V850ES/KE2发出到专用flash存储器编程器的响应信号称为“响应命令”。

图21-9. 通讯命令



下表中显示了 V850ES/KE2中的flash 存储器控制命令。所有的这些命令从专用flash 存储器编程器发出，V850ES/KE2根据命令进行处理。

表21-7. flash存储器控制命令

分类	命令名称	支持			功能
		CSI00	CSI00 + HS	UART0	
空白检查	block空白检查命令	○	○	○	检查指定block中的内容是否被正确擦除。
擦除	片擦除命令	○	○	○	擦除整个存储器的内容。
	block擦除命令	○	○	○	擦除指定存储器block的内容。
写入	编程命令	○	○	○	写入到指定的地址范围，执行检查的内容。
校验	校验命令	○	○	○	比较指定地址范围中的存储器内容与从flash存储器编程器中传输来的数据。
	校验和命令	○	○	○	读取指定地址范围中的校验和。
系统设置控制	硅信号命令	○	○	○	读取硅信号信息。
	安全设置命令	○	○	○	禁止使用block擦除命令、片擦除命令、编程命令、读取命令和重写引导区。

21.4.6 引脚连接

当进行片上写入时，将连接器安装到目标系统以连接到专用flash存储器编程器。同时，在板上提供从正常模式切换到flash存储器编程模式。

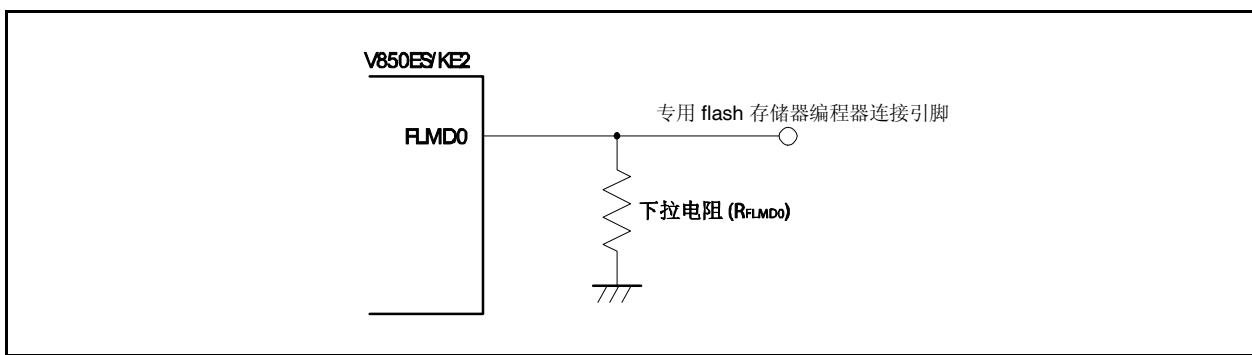
在flash存储器编程模式中，所有的不用于flash存储器编程的引脚变成与复位后相同的状态。因此，当外部设备在复位后没有立即确认状态时应要求引脚处理。

(1) FLMD0引脚

在正常运行模式下，输入电压 V_{SS} 到FLMD0引脚。在flash存储器编程模式下，向FLMD0引脚提供一个写入电压 V_{DD} 。

由于FLMD0引脚作为自编程模式下的写保护引脚，在写入到flash存储器前必须通过端口控制向FLMD0引脚提供电压 V_{DD} 。关于更多细节，敬请参阅21.5.5 (1) FLMD0引脚。

图21-10. FLMD0 引脚连接示例



(2) FLMD1引脚

当输入到FLMD0引脚的电压为0V时，FLMD0引脚不工作。当向FLMD0引脚提供V_{DD}电压时，进入flash存储器编程模式，因此必须向FLMD1引脚输入0V电压。以下显示的是FLMD1引脚连接的示例。

图21-11. FLMD1引脚连接示例

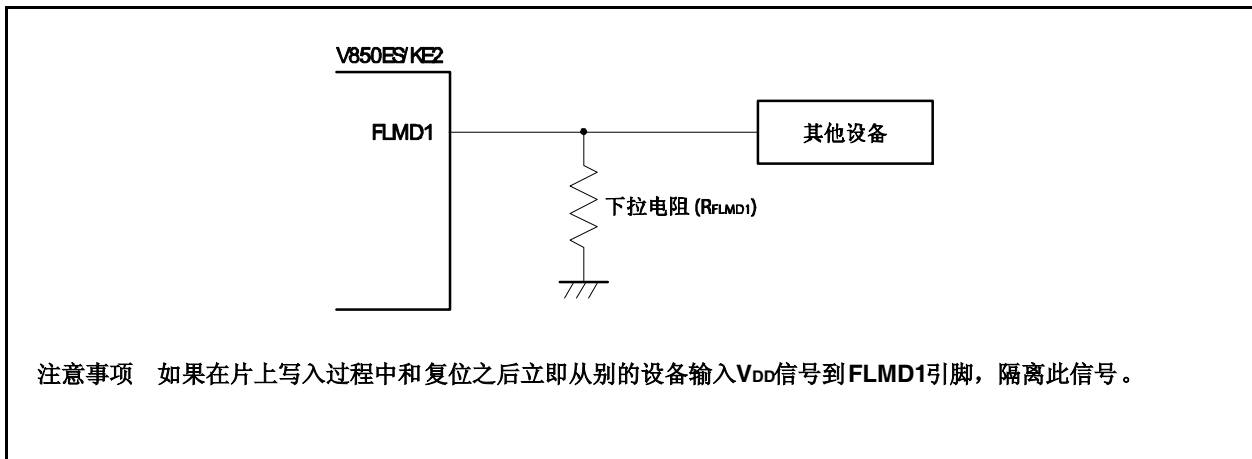


表21-8. 当放开复位时FLMD0和FLMD1引脚与操作模式之间的关系

FLMD0	FLMD1	操作模式
0	忽略	正常操作模式
V _{DD}	0	flash存储器编程模式
V _{DD}	V _{DD}	禁止设置

(3) 串行接口引脚

下表显示的是每个串行接口使用的引脚。

表21-9. 串行接口使用的引脚

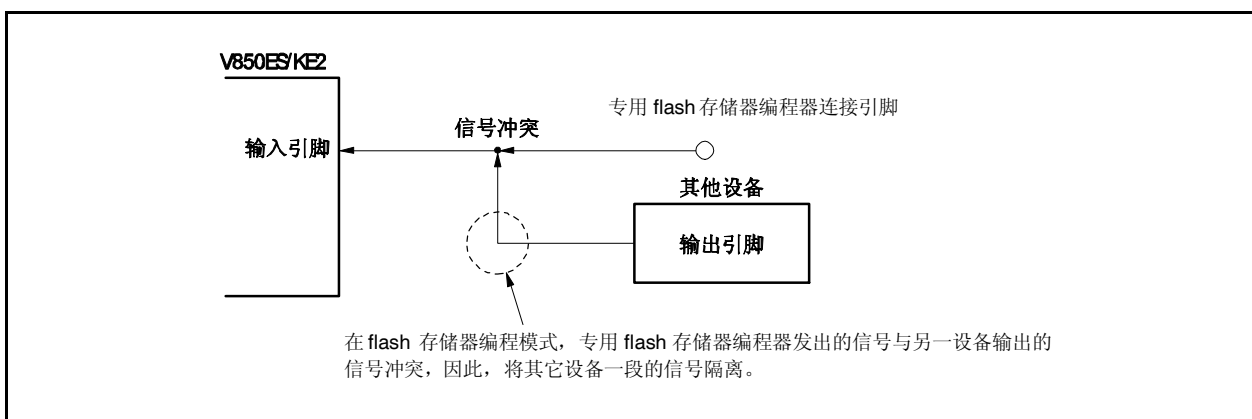
串行接口	使用的引脚
UART0	TXD0, RXD0
CSI00	SO00, SI00, $\overline{\text{SCK00}}$
CSI00 HS	SO00, SI00, $\overline{\text{SCK00}}$, PCM0

当连接专用flash存储器编程器到与另一个板上设备连接的串行接口引脚时，应小心避免信号的冲突和另一个设备的故障。

(a) 信号的冲突

当连接专用flash存储器编程器（输出）到与另一个板上设备（输出）连接的串行接口引脚（输入）时，会出现信号冲突。要避免信号冲突，应隔离到其他设备的连接或这只其他设备为输出高阻抗状态。

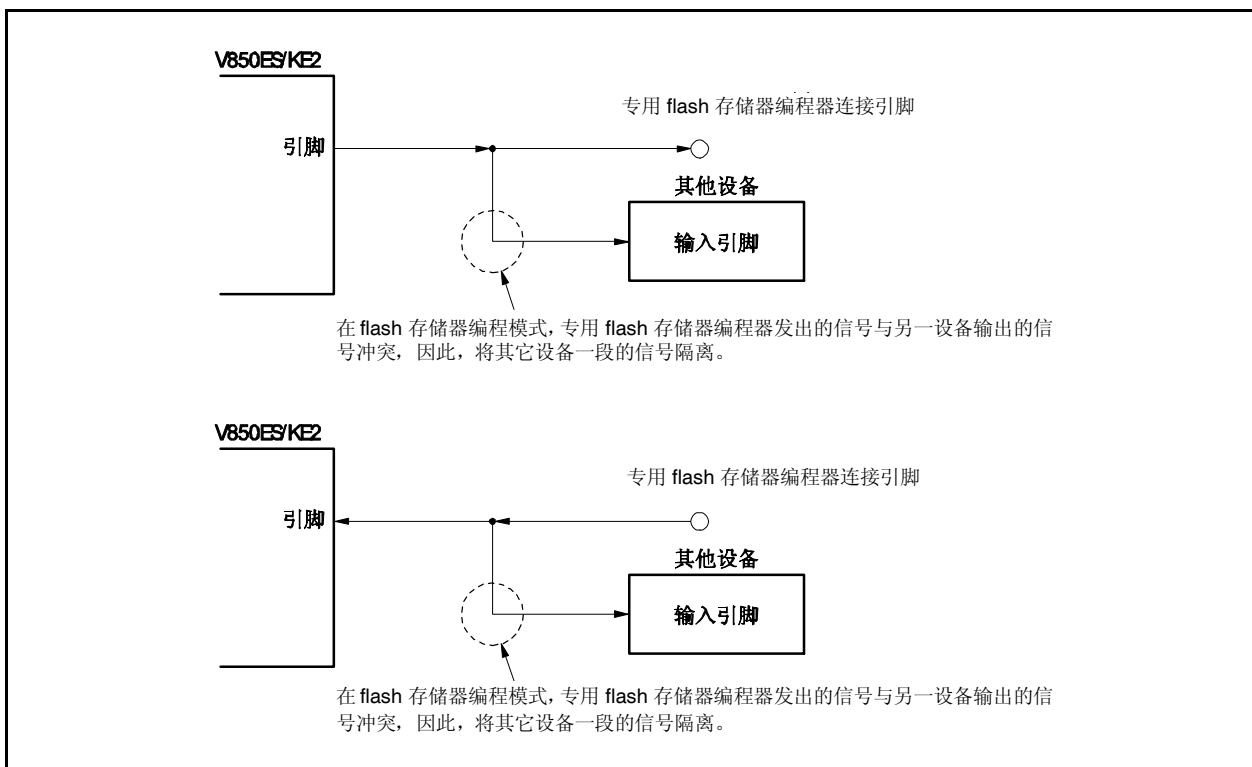
图21-12. 信号的冲突（串行接口输入引脚）



(b) 其他设备的故障

当连接专用flash存储器编程器（输出或输入）到与另一个板上设备（输入）连接的串行接口针脚（输入或输出）时，信号输出到其他设备，造成其他设备发生故障。要避免这种情况，应隔离到其他设备的连接。

图21-13. 其他设备的故障

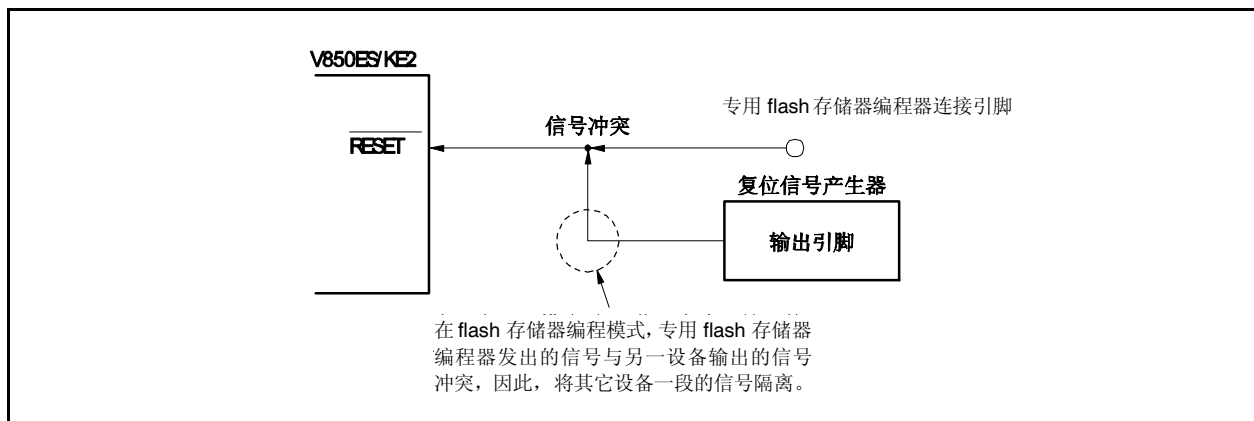


(4) RESET引脚

当专用flash存储器编程器的复位信号连接到与板上复位信号发生器的RESET引脚时，会发生信号冲突。要避免信号冲突，必须合理到复位信号发生器的连接。

当在flash存储器编程模式下从用户系统输入复位信号时，将无法正确地进行编程操作。因此，不要输入来自专用flash存储器编程器的复位信号以外的信号。

图21-14. 信号的冲突 (RESET引脚)

**(5) 端口引脚 (包括NMI)**

当系统切换到flash存储器编程模式时，所有不用于flash存储器编程的引脚与复位后的状态相同。如果与此端口连接的外部设备不识别复位后端口的状态，需要对引脚进行适当的处理，如通过一个电阻连接到V_{DD}或者V_{SS}。

(6) 其他信号引脚

与正常操作模式下的状态一样地连接X1、X2、XT1和XT2。

(7) 电源

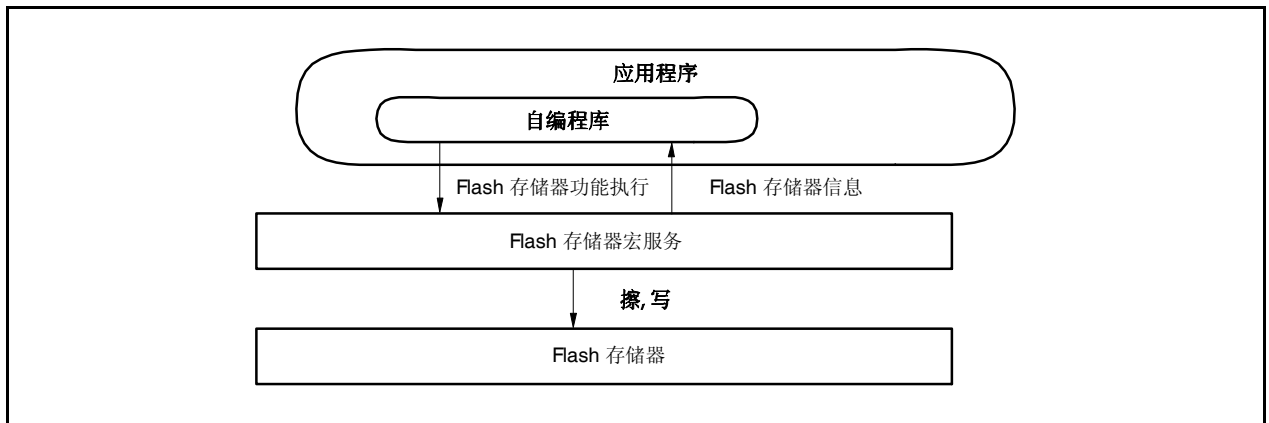
提供与正常操作模式下一样的电源 (V_{DD}、V_{SS}、EV_{DD}、EV_{SS}、AV_{SS}和AV_{REF0})。

21.5 通过自编程重写

21.5.1 概述

V850ES/KE2支持允许用户程序自己重写内部flash存储器的flash存储器宏服务。通过使用此接口和用于使用用户应用程序重写flash存储器的自编程库,可以通过事先传输到内部RAM或外部存储器的用于应用程序重写flash存储器。因此可以升级用户程序和不断重写数据。

图21-15. 自编程的概念

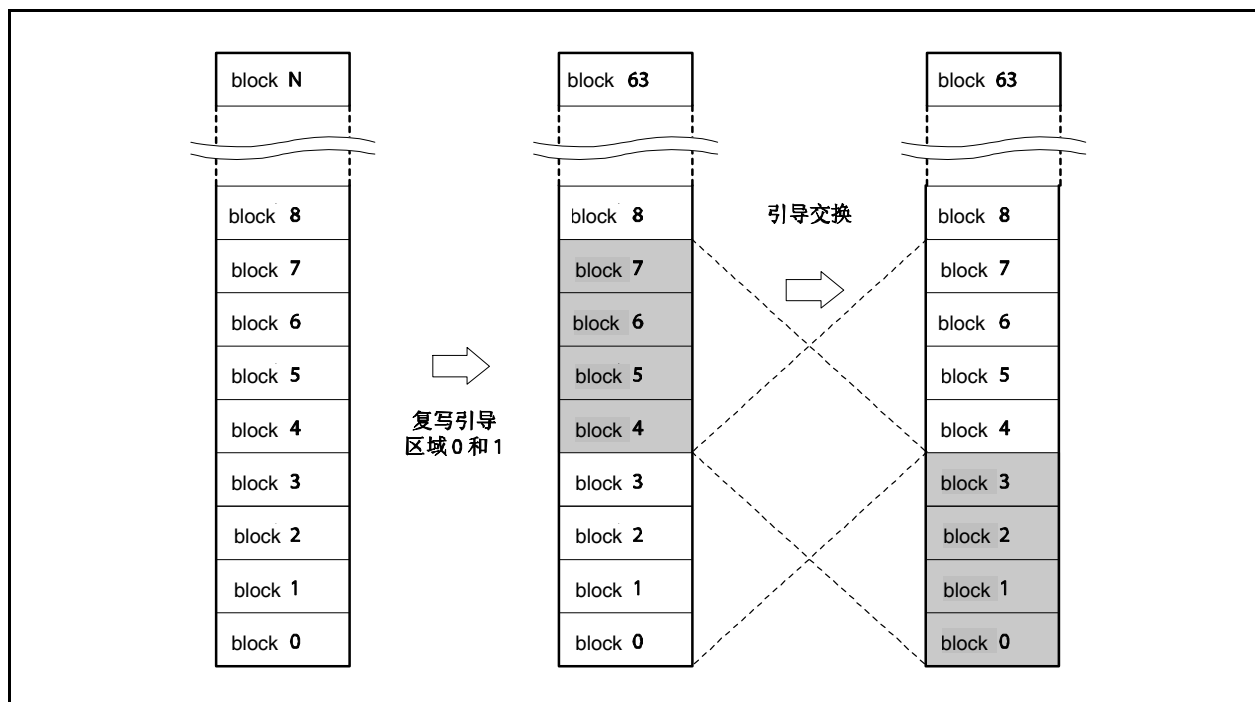


21.5.2 特点

(1) 安全的自编程（引导交换功能）

V850ES/KE2支持引导交换功能，即将引导区0的物理存储器（block0到3）与引导区1的物理存储器（block4到7）互换。通过事先写入要被重写到引导区1中的启动程序然后交换物理存储器，即使在重写过程中出现断电也可以安全地重写整个区域，因为正确的用户程序始终存在于引导区0。

图21-16. 重写整个存储器区域（引导交换）



(2) 中断支持

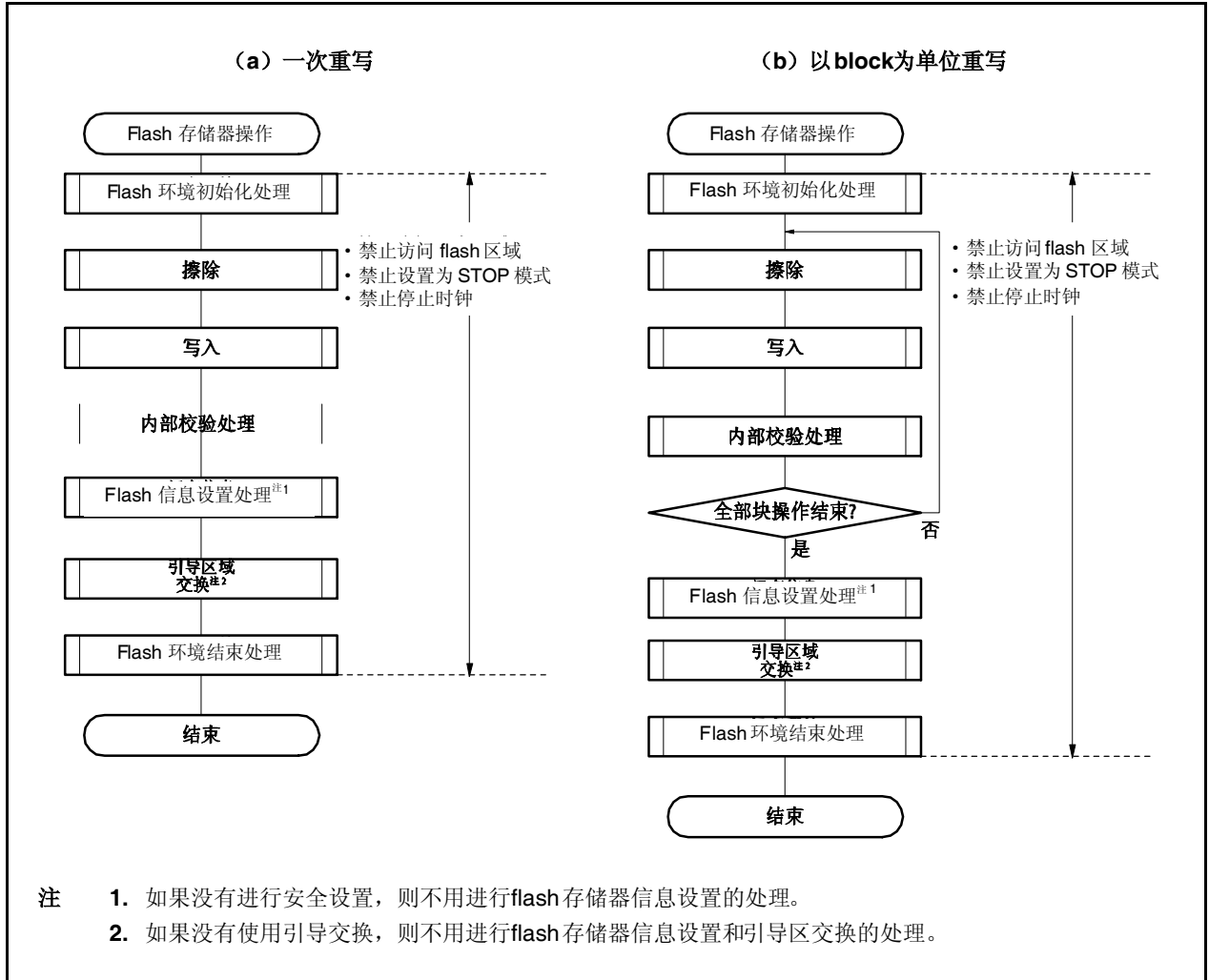
在自编程过程中不能从flash存储器获取指令。按照常规做法，即使出现中断也不能使用写入到flash存储器的用户句柄。因此在V850ES/KE2中，如果要在自编程过程中使用中断，必须在内部RAM中处理到指定地址*的转移。将转移处理的转移指令分配到内部RAM中指定地址*的用户中断服务。

注 NMI中断： 内部RAM的开始地址
可屏蔽的中断： 内部RAM的开始地址+ 4个地址

21.5.3 标准自编程流程

通过flash存储器自编程重写flash存储器的整个程序如下所示。

图21-17. 标准自编程流程



21.5.4 flash 存储器功能

表21-10. 主要flash存储器功能清单

功能名称	概述	支持
FlashEnv	flash存储器控制宏的初始化	√
Flashblock擦除	只擦除指定的一个block	√
FlashWordWrite	从指定的地址写入	√
FlashBlockIVerify	指定block的内部核查	√
FlashBlockBlankCheck	指定block的空白检查	√
FlashFLMDCheck	检查FLMD引脚	√
FlashGetInfo	阅读flash存储器信息	√
FlashSetInfo	设置flash存储器信息	√
FlashBootSwap	引导区交换	√
FlashWordRead	从指定的地址读取数据	√

备注 关于更多细节，敬请参阅V850系列flash存储器自编程（单电源flash存储器）用户手册。

请联系日电电子的销售代理获得上述手册。

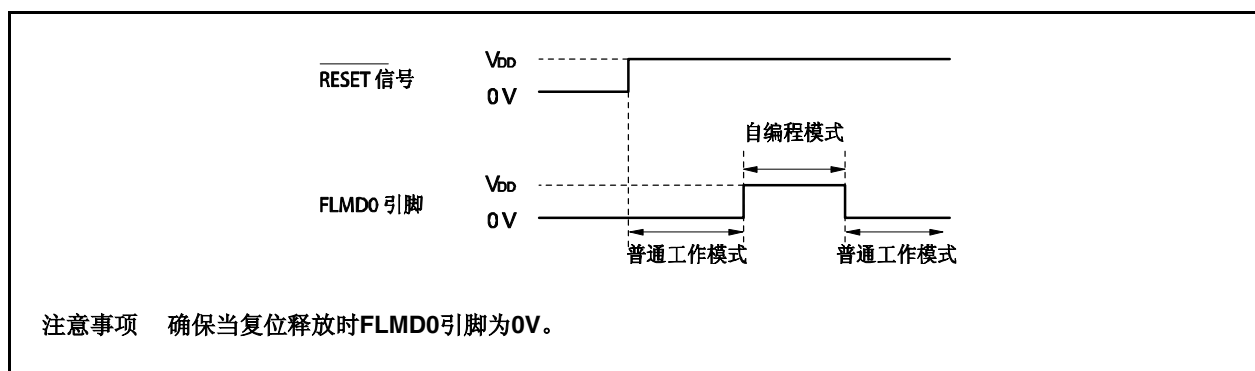
21.5.5 引脚处理

(1) FLMD0引脚

FLMD0引脚用于在释放复位时设置操作模式，从而保护flash存储器在自重写过程中不被写入。因此必须在释放复位并执行正常操作时保持供到FLMD0引脚的电压为0。在重写存储器前通过引脚控制进行自编程过程中，必须向FLMD0引脚供应 V_{DD} 的电压。

当自编程完成时，FLMD0引脚的电压必须返回0V。

图21-18. 模式变换配时



21.5.6 使用的内部资源

下表列出了用于自编程的内部资源。这些内部资源也可以自由地用于自编程以外的目的。

表21-11. 使用的内部资源

资源名称	描述
登录RAM区 (内部RAM/外部RAM大小: 136 字节)	用于flash存储器宏服务的程序和参数位于此区域。登录程序和默认的参数通过调用库初始化函数复制。
堆栈区(堆栈大小: 600字节)	用户可以通过库使用堆栈的扩展(可以在内部RAM和RAM中使用)。
库代码(代码大小: 约1600字节)	库程序实体(可以在要操作的flash存储器block以外的任何地方使用)。
应用程序	作为用户应用程序执行。 调用flash存储器函数。
可屏蔽的中断	可以在用户应用程序执行状态或自编程状态中使用。要在自编程状态中使用此中断, 由于处理将转移到内部RAM开始地址+4个地址(3FFE004H)的地址, 必须事先将转移处理的转移指令分配到内部RAM开始地址+4个地址(3FFE004H)的用户中断服务。
NMI中断	可以在用户应用程序执行状态或自编程状态中使用。要在自编程状态中使用此中断, 由于处理将转移到内部RAM开始地址(3FFE000H), 必须事先将转移处理的转移指令分配到内部RAM开始地址(3FFE000H)的用户中断服务。
TM50, TM51	由于TM50和TM51在flash存储器宏服务中使用, 不要在自编程状态中使用它们。 在自编程之后使用TM50和TM51时, 应重新设置。

备注 关于更多细节, 敬请参阅**V850系列flash存储器自编程(单电源flash存储器)用户手册**。
请联系日电电子的销售代理获得上述手册。

V850ES/KE2 应用用户资源，通过 MINICUBE2 实现片上调试功能。

V850ES/KE2 没有提供 DCU(调试控制单元)。但是通过使用片上调试仿真器(MINICUBE[®])和调试适配器(QB-V850ESKX1H-DA)，V850ES/KE2 可作为简易内部电路仿真器。关于连接，参见附录 A 开发工具。

备注 DCU(调试控制单元)电路通过使用 $\overline{\text{DRST}}$ ， DCK ， DMS ， DDI ，和 DDO 引脚作为调试接口引脚，可以实现片上调试功能。

下表显示了两个片上调试功能的特性。

表 22-1. 片上调试功能特性

		使用 DCU 调试	不使用 DCU 调试
调试接口引脚		不支持	<ul style="list-style-type: none"> 使用 UART0 时 RXD0, TXD0 使用 CSI00 时 SI00, SO00, SCK00, HS (PCM0)
保护用户资源		不支持	需要
硬件断点功能		不支持	不支持
软件断点功能	内部 ROM 区域	不支持	4 点
	RAM 区域	不支持	2000 点
实时 RAM 监视功能 ^{注1}		不支持	可用
动态存储器修改功能 (DMM 功能) ^{注2}		不支持	可用
屏蔽功能		不支持	$\overline{\text{RESET}}$ 引脚
安全功能		不支持	10 字节 ID 码加密
使用的硬件		不支持	NINICUBE2 等等
追踪功能		不支持	不支持
调试中断接口功能(DBINT)		不支持	不支持

注 1. 此功能在程序执行时读出存储器内容。
2. 此功能在程序执行时改写 RAM 的内容。

22.1 不使用 DCU 调试

下面描述如何用 MINICUBE2 实现片上调试功能。UART0 引脚(RXD0, TXD0)或 CSI00 引脚(SI00, SO00, SCK00, HS (PMC0))作为调试引脚, 不使用 DCU。

22.1.1 电路连接示例

图 22-1. 使用 UART0/CSI00 作为通信接口的电路连接示例

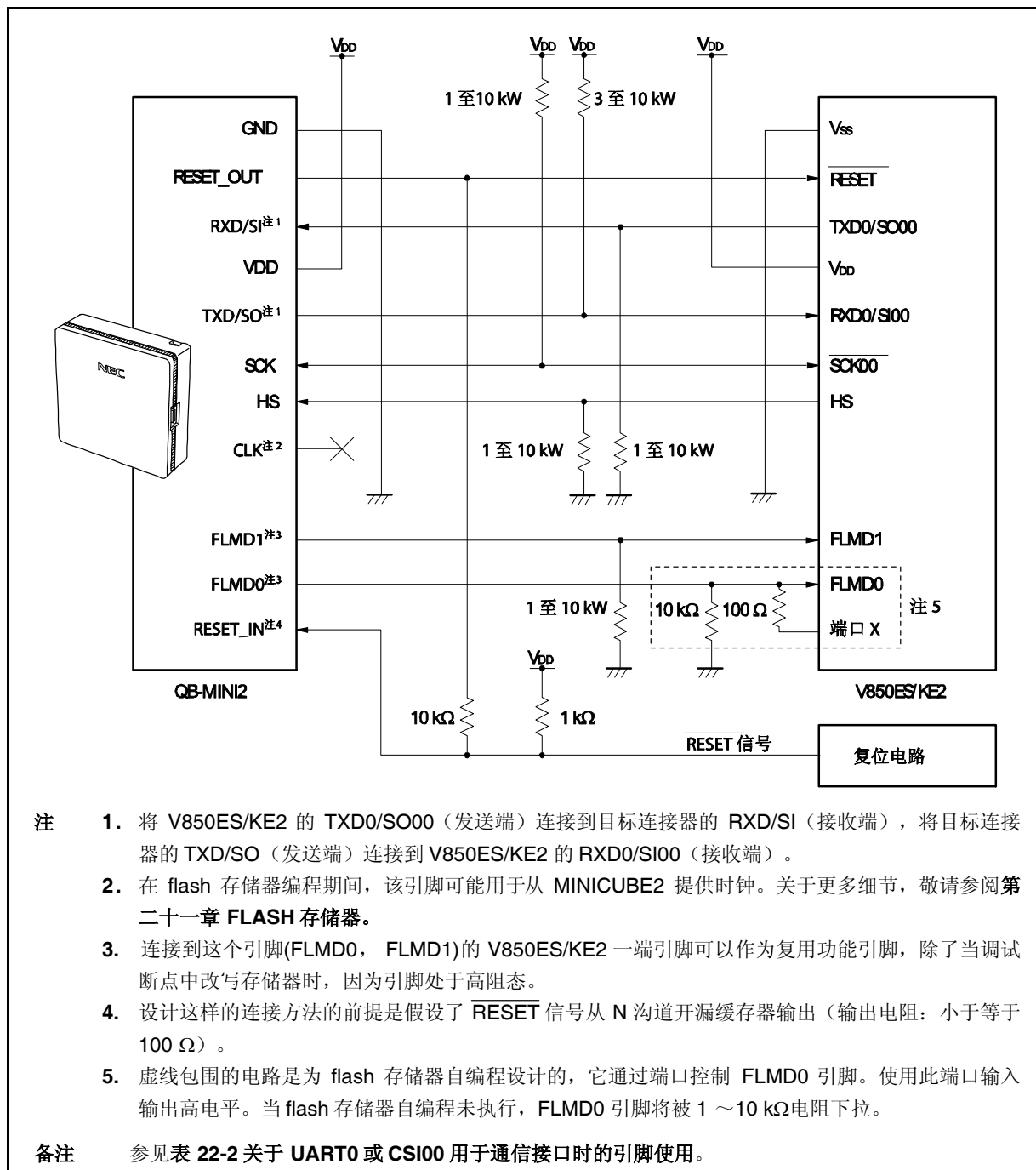


表 22-2. V850ES/KE2 与 MINICUBE2 之间的连线

MINICUBE2 (QB-MINI2)的引脚配置			通过 CSI00-HS		通过 UART0	
信号名称	输入/输出	引脚功能	引脚名称	引脚号	引脚名称	引脚号
SI/RxD	输入	从 V850ES/KE2 接收命令和数据引脚	P41/SO00	20	P30/TXD0	22
SO/TxD	输出	向 V850ES/KE2 发送命令和数据引脚	P40/SI00	19	P31/RXD0	23
SCK	输出	3 线串行通信时钟输出引脚	P42/SCK00	21	不需要	不需要
CLK [‡]	输出	向 V850ES/KE2 时钟输出引脚	不需要 [‡]	不需要 [‡]	不需要 [‡]	不需要 [‡]
			不需要 [‡]	不需要 [‡]	不需要 [‡]	不需要 [‡]
RESET_OUT	输出	向 V850ES/KE2 复位输出引脚	RESET	9	RESET	9
FLMD0	输出	设置 V850ES/KE2 调试模式或可编程模式的输出引脚	FLMD0	3	FLMD0	3
FLMD1	输出	设置可编程模式的输出引脚	PDL5/FLMD1	52	PDL5/FLMD1	52
HS	输入	CSI00 + HS 通信的握手信号	PCM0	45	不需要	不需要
VDD	-	VDD 电压产生	VDD	4	VDD	4
			EVDD	33	BVDD	33
			AVREF0	1	AVREF0	1
GND	-	地	VSS	6	VSS	6
			AVSS	2	AVSS	2
			EVSS	32	EVSS	32
RESET_IN	输入	目标系统的复位输入引脚				

注 作为 MINICUBE2 flash 存储器编程器的时钟输出。关于更多细节，敬请参阅 第二十一章 FLASH 存储器。

22.1.2 可屏蔽功能

只有复位信号可被屏蔽。

调试器(ID850QB)的可屏蔽功能以及相应的 V850ES/KE2 功能如下。

表 22-3. 可屏蔽功能

ID850QB 的可屏蔽功能	相应的 V850ES/KE2 功能
NMI0	-
NMI1	-
NMI2	-
STOP	-
HOLD	-
RESET	RESET 引脚输入产生的 RESET 信号
WAIT	-

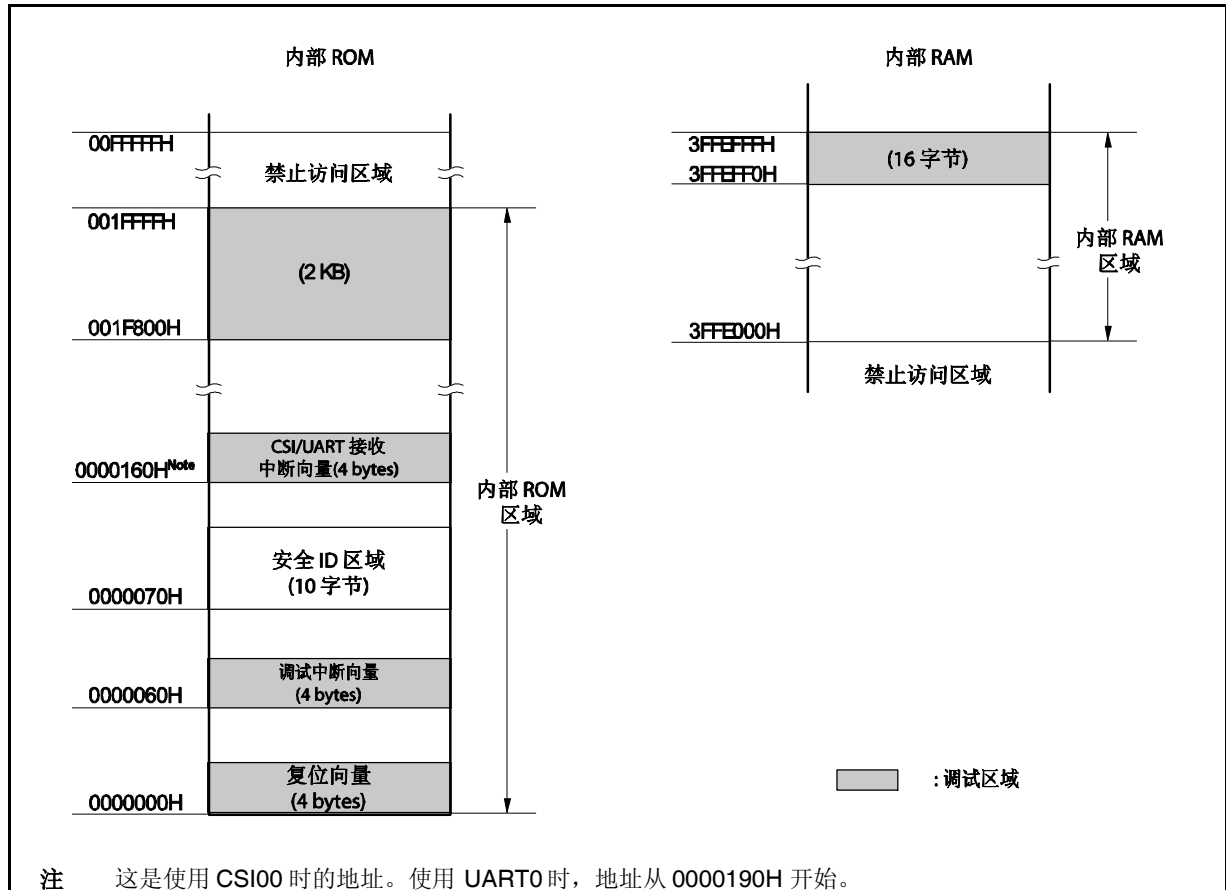
22.1.3 保护用户资源

用户必须做如下准备，以执行 MINICUBE2 和目标设备的通信，并实现每种调试功能。这些项目需要在用户程序中设置或使用编译器选项。

(1) 保护内存空间

图 22-2 阴影部分是放置调试监测程序的保留区域，因此用户程序和数据不能分配到这些空间。这些空间必须受到保护，不能被用户程序使用。

图 22-2. 调试监测程序分配的内存空间



- 安全 ID 设置

ID 码必须嵌入 0000070H ~ 0000079H 之间的区域，如图 22-2，以防止内存被没有授权的人独取。关于更多细节，敬请参阅 22.2 ROM 安全功能。

(2) 复位向量

一个复位向量包含调试监测程序的跳转指令。

[如何保护区域]

无需有意保护此区域。但当下载程序时，调试器将复位向量按如下情况改写。如果改写类型与如下情况不匹配，调试器产生错误(F0C34 当使用 ID850QB 时)。

(a) 0 从地址 0 开始连续放置 2 条 nop 指令

改写前		改写后
0x0 nop	→	跳转到 0x0 处调试监测程序
0x2 nop		0x4 xxxx
0x4 xxxx		

(b) 从地址 0 连续放置 2 个 0xFFFF (已擦除器件)

改写前		改写后
0x0 0xFFFF	→	跳转到 0x0 处调试监测程序
0x2 0xFFFF		0x4 xxxx
0x4 xxxx		

(c) jr 指令放在地址 0 处(使用 CA850 时)

改写前		改写后
0x0 jr disp22	→	跳转到 0x0 处调试监测程序
		0x4 jr disp22 - 4

(d) mov32 和 jmp 从地址 0 连续放置(使用 IAR 编译器 ICCV850)

改写前		改写后
0x0 mov imm32, reg1	→	跳转到 0x0 处调试监测程序
0x6 jmp [reg1]		0x4 mov imm32, reg1
		0xa jmp [reg1]

(e) 地址 0 处放置调试监测程序的跳转指令

改写前		改写后
跳转到 0x0 处调试监测程序	→	不改变

(3) 调试监测程序区域的保护

图 22-2 阴影部分是分配调试监测程序的区域。监测程序对调试通信接口执行初始化处理，对 CPU 执行断点处理。

[如何保护区域]

如果用户程序不用此区域则无需保护此区域。

为避免可能在调试启动时发生的问题，建议用编译器预先保护此区域。

下面为保护此区域的示例，应用 NEC 电子编译器 CA850。添加汇编源文件和链接指示码如下所示。

- 汇编源(添加如下代码作为汇编源文件)

```
-- Secures 2 KB space for monitor ROM section
.section "MonitorROM", const
.space 0x800, 0xff

-- Secures interrupt vector for debugging
.section "DBG0"
.space 4, 0xff

-- Secures interrupt vector for serial communication
-- Change the section name according to the serial communication mode used
.section "INTCSI00"
.space 4, 0xff

-- Secures 16-byte space for monitor RAM section
.section "MonitorRAM", bss
.lcomm monitorramsym, 16, 4 -- defines symbol monitorramsym
```

- 链接指示(添加如下码至链接指示文件.)

如下为一示例，内部 ROM 为 128 KB(末尾地址为 003FFFFH)内部 RAM 为 4 KB(末尾地址为 1FFEFFFH)

```
MROMSEG      : !LOAD ?R V0x01f800{
              MonitorROM = $PROGBITS ?A MonitorROM;
};

MRAMSEG      : !LOAD ?RW V0x03ffeff0{
              MonitorRAM  = $NOBITS ?AW MonitorRAM;
};
```

(4) 通信串行接口的保护

UART0 或 CSI00 用于在 MINICUBE2 和目标系统之间通信。与串行接口模式相关的设置由调试监测程序执行，但是如果设置被用户程序改变，将可能发生通信错误。

为防止这样的问题发生，通信串行接口需要在用户程序中被保护。

[如何保护通信串行接口]

- 串行接口寄存器

不要再用户程序中设置与 UART0 和 CSI00 相关的寄存器。

- 中断屏蔽寄存器

当使用 UART0 时，不要屏蔽接收结束中断(INTSR0)。当使用 CSI00 时，不要屏蔽发送结束中断(INTCSI00)。

(a) 使用 CSI00 时								
	7	6	5	4	3	2	1	0
CSI0IC0	x	0	x	x	x	x	x	x
(b) 使用 UART0 时								
	7	6	5	4	3	2	1	0
SRIC0	x	0	x	x	x	x	x	x
备注	x: 不必考虑							

- 使用 UART0 时的端口寄存器

当使用 UART0 时，端口寄存器通过调试监测程序设置为使得 TXD0 和 RXD0 引脚有效。当用户程序调试时，不要改变如下寄存器设置。（同样的值可以多次写入）

	7	6	5	4	3	2	1	0
PMC3L	x	x	x	x	x	x	1	1

备注 x: 不必考虑

- 使用 CSI00 时的端口设置。

使用 CSI00 时，端口寄存器通过调试监测程序设置为使得 SI00，SO00， $\overline{\text{SCK00}}$ 和 HS (PMC0) 引脚有效。调试时不要在用户程序中改变如下寄存器设置。（同样的值可以多次写入）

(a) SI00, SO00, 和 $\overline{\text{SCK00}}$ 设置

	7	6	5	4	3	2	1	0
PMC4	x	x	x	x	x	1	1	1

(b) HS (PMC0 引脚) 设置

	7	6	5	4	3	2	1	0
PMCM	x	x	x	x	x	x	x	0

	7	6	5	4	3	2	1	0
PCM	x	x	x	x	x	x	x	注

注 禁止写入该位。
与 HS 引脚对应的端口值根据调试器状态被监测程序改变。要以 8 位单元执行端口寄存器设置，用户程序通常可以使用读-修改-写。如果调试中断在写入前发生，将可能执行一个不可预料的操作。

备注 x: 不必考虑

22.1.4 注意事项

(1) 用于调试的设备的处理

不要将用于调试的设备安装在大批量生产的产品上，因为调试过程中 flash 存储器会被改写，flash 存储器改写的次数不能保证。此外，不要把调试监测程序嵌入大批量生产的产品中。

(2) 断点不能执行时

如果满足下列条件之一，强制断点不能执行。

- 中断禁止(DI)
- 发给串行接口并用于 MINICUBE2 和目标设备之前通信的中断被屏蔽
- 禁止由可屏蔽中断释放待机模式时进入待机模式
- MINICUBE2 与目标设备之间通信模式为 UART0，且主时钟被 MINICUBE2 停止

(3) 当伪实时 RAM 监测器(RRM)功能和 DMM 功能不工作时

满足如下条件之一时，伪实时 RAM 监测器(RRM)功能和 DMM 功能不工作。

- 中断被禁止(DI)
- 发给串行接口用于 MINICUBE2 和目标设备之间通信的中断被屏蔽
- 由于禁止可屏蔽中断释放待机而进入待机模式
- MINICUBE2 与目标设备之间通信模式为 UART0，且主时钟被 MINICUBE2 停止
- MINICUBE2 与目标设备之间通信模式为 UART0。且用于通信的时钟与调试器指定的时钟不同

(4) 允许伪 RRM 和 DMM 功能时的待机释放

如果如下条件之一满足，待机模式由伪 RRM 和 DMM 功能释放。

- MINICUBE2 和目标设备之间的通信模式为 CSI00
- MINICUBE2 和目标设备之间的通信模式为 UART0，且主时钟已加入

(5) 应用 DMM 功能写需要特定顺序的外围 I/O 寄存器

需要特定顺序的外围 I/O 寄存器不能用 DMM 功能写入。

(6) flash 存储器自编程

如果分配了调试监测程序的空间被 flash 存储器自编程改写，调试器不能再正常工作。

22.2 ROM 安全功能

22.2.1 安全 ID

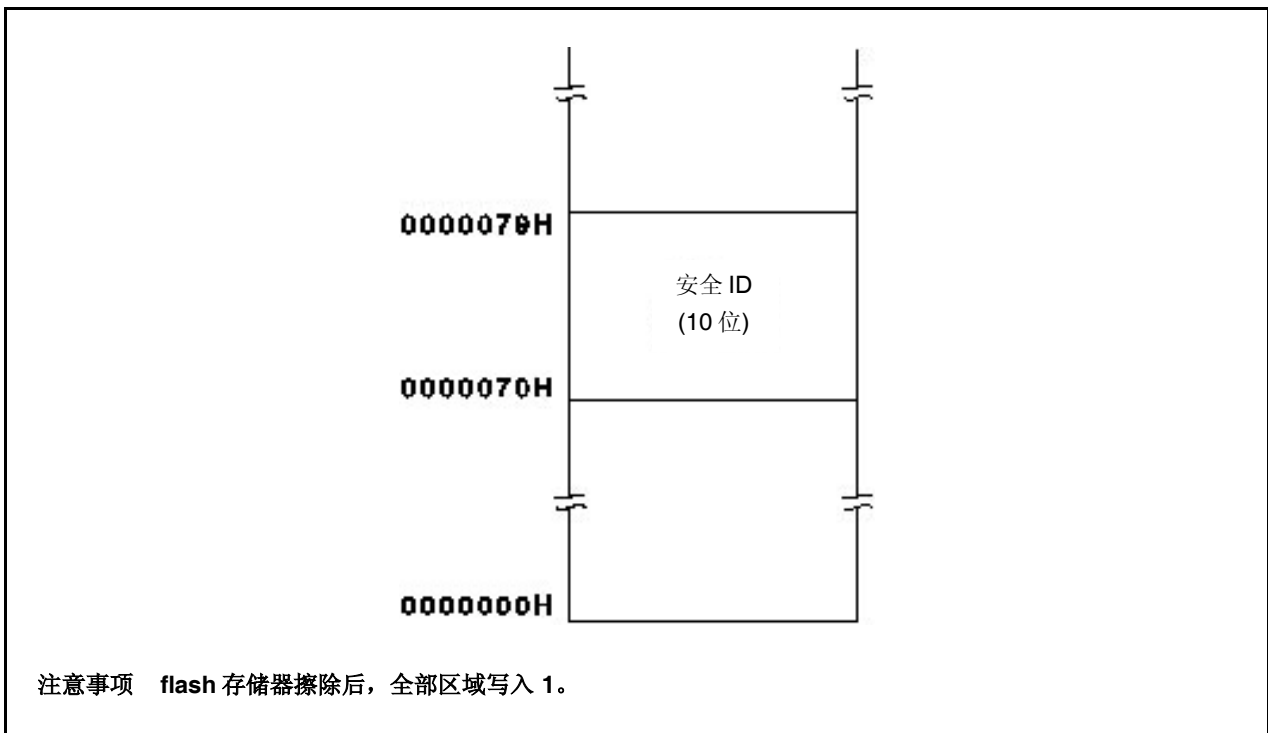
V850ES/KE2 的 flash 存储器版使用 10 个字节的 ID 码来进行加密，以防在片上调试期间 flash 存储器的内容被没有授权的人通过片上调试仿真器读取。

在 10 个字节的片上 flash 从 0000070H ~ 0000079H 的存储区域进行设置，对调试器进行 ID 加密。

如果 IDs 一致，那么就释放安全保密功能，读取和使用片上调试仿真器均可。

- 在 0000070H ~ 0000079H 之间设置 10 个字节的 ID 号码。
- 0000079H 的位 7 是 片上调试仿真器的允许标志
(0: 禁止, 1: 允许)
- 当片上调试仿真器启动时，调试器要求输入 ID 号码。当输入的 ID 号码与 0000070H ~ 0000079H 中设置的 ID 号码一致时，调试器开始工作。
- 如果片上调试仿真器允许标志为 0，即使 ID 号码匹配，也不能进行调试。

图 22-3. 安全 ID 区域



22.2.2 设置

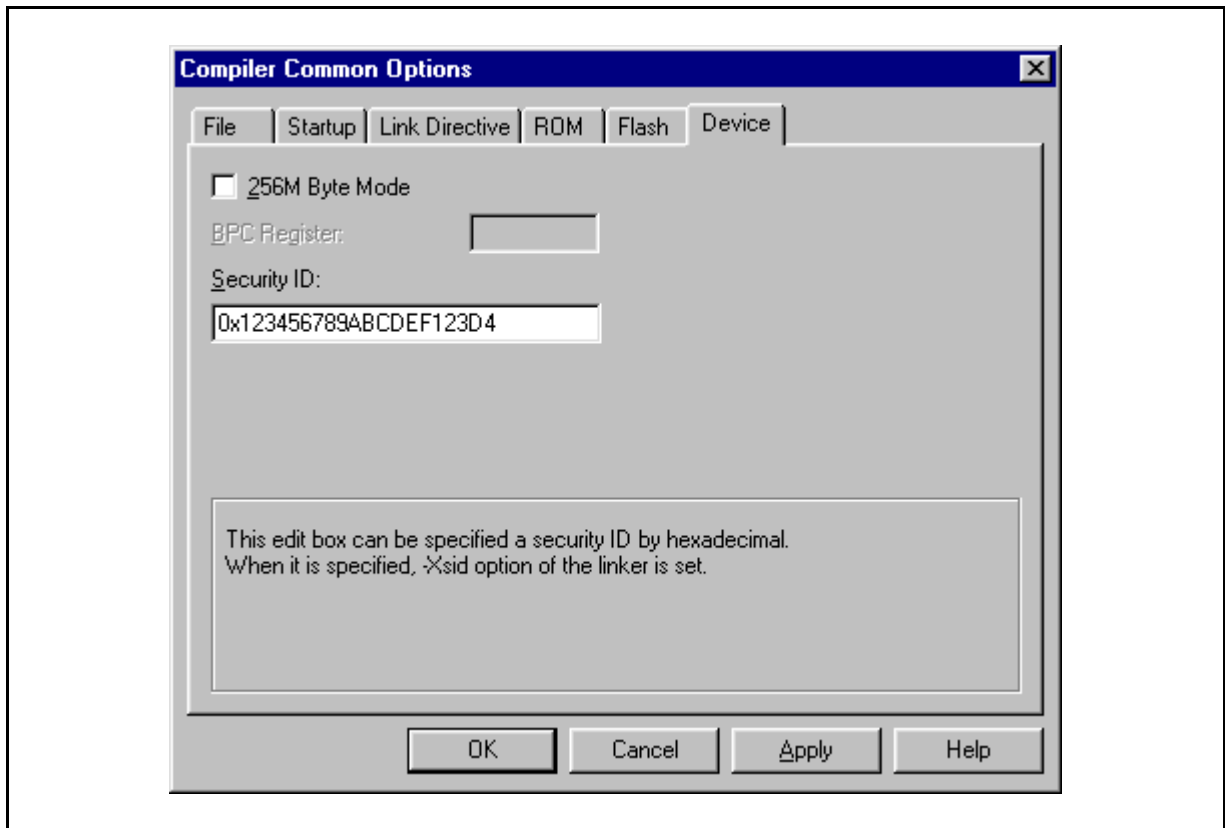
表 22-4 显示如何设置 ID 码。

当 ID 码如表 22-4 所示设置，ID850QB 的配置对话框中的 ID 码输入为“123456789ABCDEF123D4”(不区分 ID 码大小写)。

表 22-4. ID 码

地址	值
0x70	0x12
0x71	0x34
0x72	0x56
0x73	0x78
0x74	0x9A
0x75	0xBC
0x76	0xDE
0x77	0xF1
0x78	0x23
0x79	0xD4

对于支持 CA850 Ver.3.10 或以上版本以及可以应用 PM+编译器普通选项设置的安全 ID 的设备文件，可以指定 ID 码。



[程序示例使用 CA850 Ver.3.10 或以上版本]

```
#-----  
#      安全 ID  
#-----  
      .section    "SECURITY_ID"    --中断处理地址 0x70  
      .word       0x78563412       --0-3 字节码  
      .word       0xF1DEBC9A       --4-7 字节码  
      .hword      0xD423           --8-9 字节码
```

备注 将以上程序添加到启动文件中。

第二十三章 电气特性

最大额定值 (TA = 25°C) (1/2)

参数	符号	条件	额定值	单位
电源电压	V _{DD}	V _{DD} = EV _{DD} = AV _{REF0}	-0.3 ~ +6.5	V
	AV _{REF0}	V _{DD} = EV _{DD} = AV _{REF0}	-0.3 ~ +6.5	V
	EV _{DD}	V _{DD} = EV _{DD} = AV _{REF0}	-0.3 ~ +6.5	V
	V _{SS}	V _{SS} = EV _{SS} = AV _{SS}	-0.3 ~ +0.3	V
	AV _{SS}	V _{SS} = EV _{SS} = AV _{SS}	-0.3 ~ +0.3	V
	EV _{SS}	V _{SS} = EV _{SS} = AV _{SS}	-0.3 ~ +0.3	V
输入电压	V _{I1}	P00 ~ P06, P30 ~ P35, P38, P39, P40 ~ P42, P50 ~ P55, P90, P91, P96 ~ P99, P913 ~ P915, PCM0, PCM1, PDL0 ~ PDL7, RESET, FLMD0	-0.3 ~ EV _{DD} + 0.3 ^注	V
	V _{I2}	X1, X2, XT1, XT2	-0.3 ~ V _{DD} + 0.3 ^注	V
模拟输入电压	V _{IAN}	P70 ~ P77	-0.3 ~ AV _{REF0} + 0.3 ^注	V

注 确保不要超过每个电源电压的最大额定值（最大值）。

- 注意事项**
1. 不要直接将 IC 产品的输出（或 I/O）引脚连接起来，或连接到 V_{DD}，V_{CC} 和 GND。漏极开路引脚或集电极开路引脚，可以直接互相连接。
如果输出引脚设置为高阻抗状态并且避免与外部电路的输出时序冲突，可以直接将 IC 产品的输出引脚与外部电路连接起来。
 2. DC 特性和 AC 特性中指出的额定值和条件是正常工作的质量保证。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 (TA = 25°C) (2/2)

参数	符号	条件		额定值	单位
输出电流, 低	I _{OL}	注	每个引脚	20	mA
		P38, P39		30	mA
		P00 ~ P06, P30 ~ P35, P38, P39, P40 ~ P42	所有引脚总和: 70 mA	35	mA
		P50 ~ P55, P90, P91, P96 ~ P99, P913 ~ P915, PCM0, PCM1, PDL0 ~ PDL7		35	mA
输出电流, 高	I _{OH}	注	每个引脚	-10	mA
		P00 ~ P06, P30 ~ P35, P40 ~ P42	所有引脚总和: -60 mA	-30	mA
		P50 ~ P55, P90, P91, P96 ~ P99, P913 ~ P915, PCM0, PCM1, PDL0 ~ PDL7		-30	mA
输出电流, 高	T _A	正常操作模式		-40 ~ +85	°C
		Flash 存储器编程模式		-40 ~ +85	°C
存储温度	T _{stg}			-40 ~ +125	°C

注 P00 ~ P06, P30~P35, P40~P42, P50~P55, P90, P91, P96~P99, P913~P915, PCM0, PCM1, PDL0~PDL7

- 注意事项 1. 不要直接将 IC 产品的输出 (或 I/O) 引脚连接起来, 或连接到 V_{DD}, V_{CC} 和 GND。漏极开路引脚或集电极开路引脚, 可以直接互相连接。如果输出引脚设置为高阻抗状态并且避免与外部电路的输出时序冲突, 可以直接将 IC 产品的输出引脚与外部电路连接起来。
2. DC 特性和 AC 特性中指出的额定值和条件是正常工作的质量保证。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。

电容 (TA = 25°C, V_{DD} = EV_{DD} = AV_{REF0} = V_{SS} = EV_{SS} = AV_{SS} = 0 V)

参数	符号	条件		MIN.	TYP.	MAX.	单位
输入电容	C _I	f _X = 1 MHz	P70 ~ P77			15	pF
I/O 电容	C _{IO}	不可测量的引脚返回 0 V	注			15	pF
			P38, P39			20	pF

注 P00~P06, P30~P35, P40~P42, P50~P55, P90, P91, P96~P99, P913~P915, PCM0, PCM1, PDL0~PDL7

备注 f_x: 主时钟振荡频率

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

参数	符号	条件	MIN.	TYP.	MAX.	单位
输入频率	f_X		2		5	MHz
输出频率	f_{XX}		8		20	MHz
锁时间	t_{PLL}	V_{DD} 达到 2.7 V 之后 (最小值)			200	μs

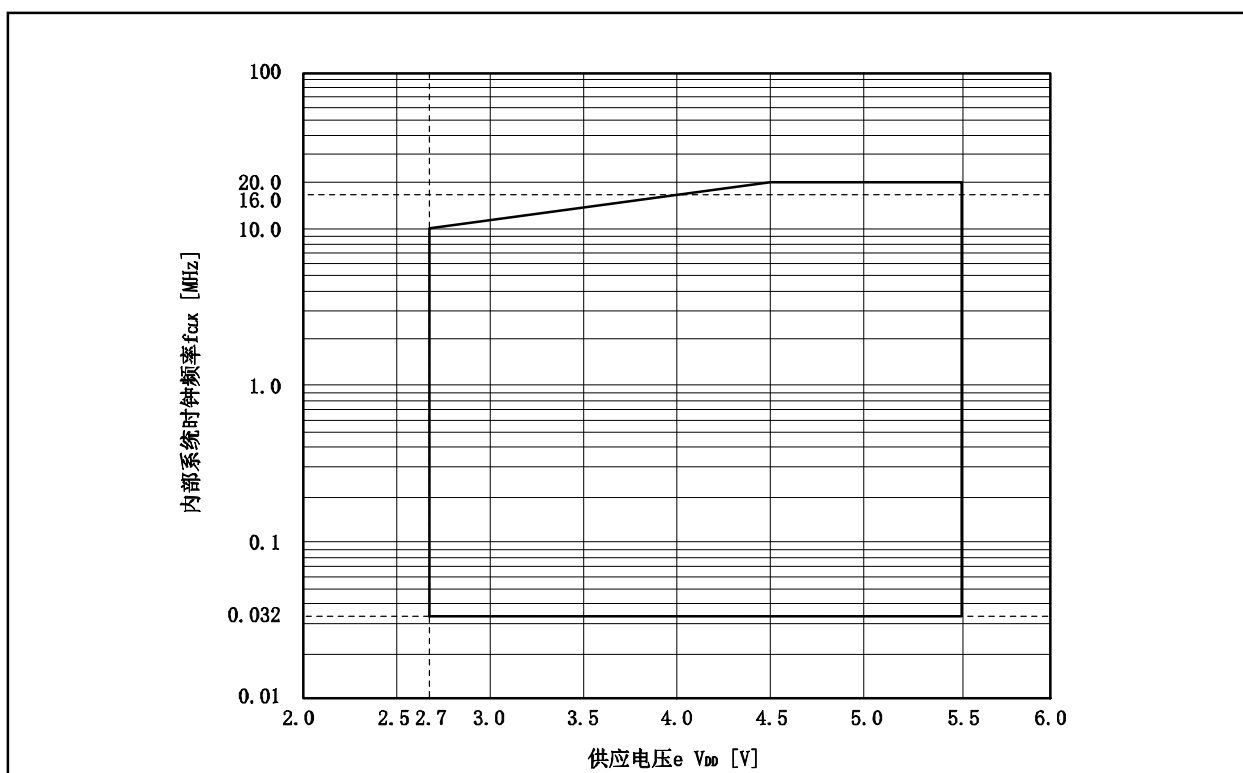
工作条件

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.7 \sim 5.5 \text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0 \text{ V}$, $C_L = 50 \text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位	
内部系统时钟频率	f_{CLK}	PLL 模式	$V_{DD} = 4.5 \sim 5.5 \text{ V}$	0.25		20	MHz
			$V_{DD} = 4.0 \sim 5.5 \text{ V}$	0.25		16	MHz
			$V_{DD} = 2.7 \sim 5.5 \text{ V}$	0.25		10	MHz
		时钟直连模式	$V_{DD} = 2.7 \sim 5.5 \text{ V}$	0.0625		10	MHz
		在副时钟下工作	注		32.768		kHz

注 $V_{DD} = 2.7 \sim 5.5 \text{ V}$

内部系统时钟频率与电源电压关系



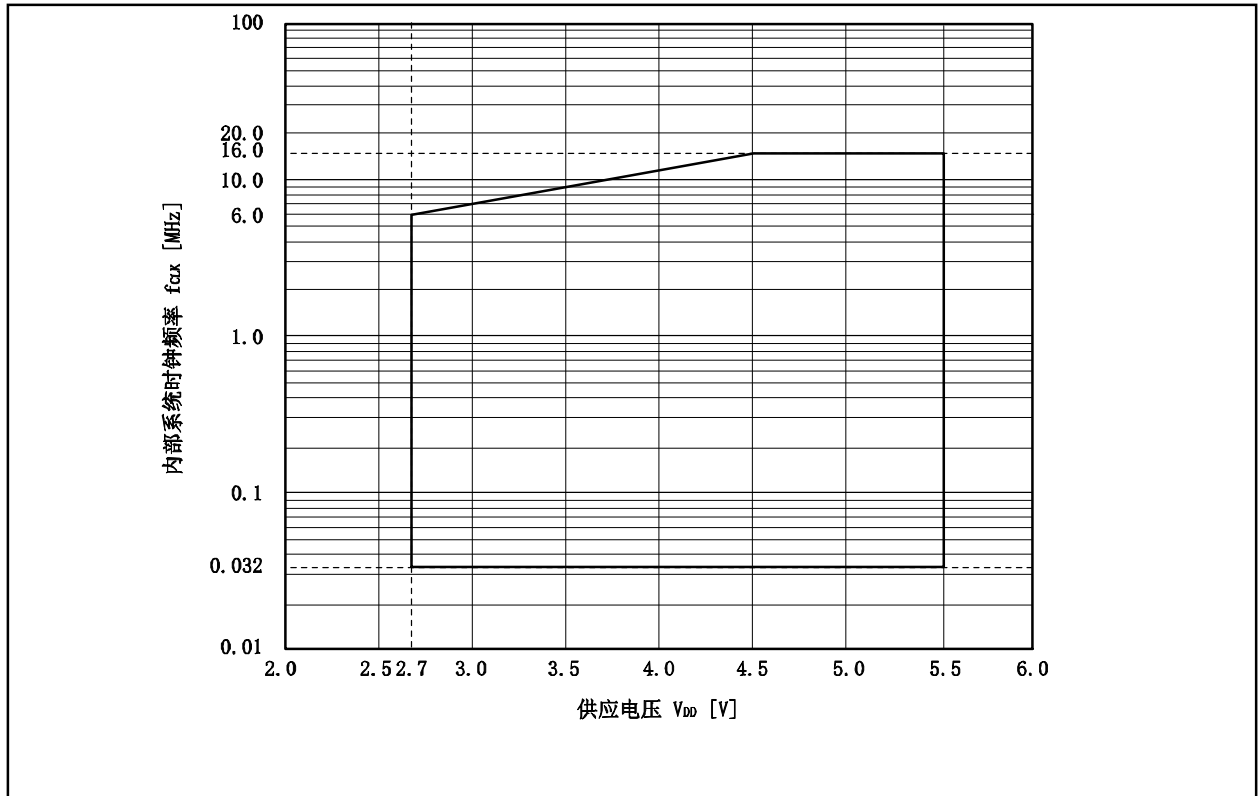
EEPROM 仿真时的操作条件

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.7 \sim 5.5 \text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0 \text{ V}$, $C_L = 50 \text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位	
内部系统时钟频率	f_{CLK}	PLL 模式	$V_{DD} = 4.5 \sim 5.5 \text{ V}$	0.25		16	MHz
			$V_{DD} = 4.0 \sim 5.5 \text{ V}$	0.25		12	MHz
			$V_{DD} = 2.7 \sim 5.5 \text{ V}$	0.25		6	MHz
		时钟直连模式	$V_{DD} = 4.0 \sim 5.5 \text{ V}$	0.0625		10	MHz
			$V_{DD} = 2.7 \sim 5.5 \text{ V}$	0.0625		6	MHz
		在副时钟下工作	注 1, 2		32.768		kHz

- 注 1. $V_{DD} = 2.7 \sim 5.5 \text{ V}$
 2. 不要停止主时钟。

内部系统时钟频率与电源电压关系



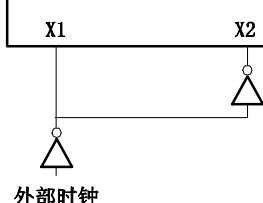
主时钟振荡器特性

(1) 晶体谐振器, 陶瓷谐振器 ($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = 2.7 \sim 5.5 \text{ V}$, $V_{SS} = 0 \text{ V}$)

电路示例	参数	条件		MIN.	TYP.	MAX.	单位	
	振荡频率 (f_x) ^{注1}	PLL 模式	$V_{DD} = 4.5 \sim 5.5 \text{ V}$	2		5	MHz	
			$V_{DD} = 4.0 \sim 5.5 \text{ V}$	2		4	MHz	
			$V_{DD} = 2.7 \sim 5.5 \text{ V}$	2		2.5	MHz	
	振荡稳定时间 ^{注2}	复位释放后	时钟直连模式	$V_{DD} = 2.7 \sim 5.5 \text{ V}$	2		10	MHz
			STOP 模式释放后	OSTS0 = 1			$2^{15}/f_x$	s
						注 3	s	

- 注
1. 仅指振荡器特性。
 2. 复位操作或 STOP 模式释放后谐振稳定的时间。
 3. 数值根据 OSTS 寄存器的设置有所变化。

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

电路示例	参数	条件		MIN.	TYP.	MAX.	单位
	X1, X2 输入频率 (f_x) ^注	PLL 模式	$V_{DD} = 4.5 \sim 5.5 \text{ V}$	2		5	MHz
			$V_{DD} = 4.0 \sim 5.5 \text{ V}$	2		4	MHz
			$V_{DD} = 2.7 \sim 5.5 \text{ V}$	2		2.5	MHz
		时钟直连模式	$V_{DD} = 2.7 \sim 5.5 \text{ V}$	2		10	MHz

注 输入波形的占空比必须在 $50\% \pm 5\%$ 范围内。

- 注意事项
1. 当使用主时钟振荡器, 上图中被虚线包围的部分的配线应按照如下布线方法布线, 以防止连接线电容产生不利影响。
 - 连接线越短越好。
 - 连接线不应与其他信号线交叉。
 - 流经的电流变化较大的信号线不要在振荡器周围布线。
 - 要保持振荡器电容器的接地点电压与 V_{SS} 相同。
 - 不要将电容的地信号接入大电流地。
 - 不要从振荡器获取信号。
 2. 当主时钟停止副时钟工作时, 在转换回主时钟前等待直到振荡稳定时间, 由程序保证。

副时钟振荡器特性

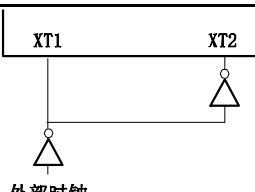
(1) 晶体谐振器 ($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = 2.7 \sim 5.5 \text{ V}$, $V_{SS} = 0 \text{ V}$)

电路示例	参数	条件	MIN.	TYP.	MAX.	单位
	振荡频率 (f_{XT}) ^注		32	32.768	35	kHz
	振荡稳定时间 ^{注2}			10		s

注 1. 仅指振荡器特性。

2. 从 V_{DD} 达到振荡电压范围 (2.7 V (最小值)) 到晶体谐振器稳定的时间。

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

电路示例	参数	条件	MIN.	TYP.	MAX.	单位
 外部时钟	输入频率 (f_X) ^注	$V_{DD} = 2.7 \sim 5.5 \text{ V}$	32		35	kHz

注 输入波形的占空比必须在 $50\% \pm 5\%$ 范围内。

- 注意事项
- 当使用副时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。
 - 连接线越短越好。
 - 连接线不应与其他信号线交叉。
 - 流经的电流变化较大的信号线不要在振荡器周围布线。
 - 要保持振荡器电容器的接地点电压与 V_{SS} 相同。
 - 不要将电容的地信号接入大电流地。
 - 不要从振荡器获取信号。
 - 为了缩减能耗，副时钟振荡器被设计为低幅电路，并且它比主时钟振荡器更容易受到噪声的影响，此在使用副时钟时更要注意布线方法。

DC 特性

(TA = -40 ~ +85°C, VDD = EVDD = AVREF0 = 2.7 ~ 5.5 V, VSS = EVSS = AVSS = 0 V) (1/3)

参数	符号	条件	MIN.	TYP.	MAX.	单位	
输出电流, 高	IOH1	P00 ~ P06, P30 ~ P35, P40 ~ P42, P50 ~ P55, P90, P91, P96 ~ P99, P913 ~ P915, PCM0, PCM1, PDL0 ~ PDL7 的每个引脚			-5.0	mA	
		P00 ~ P06, P30 ~ P35, P40 ~ P42 的总和	EVDD = 4.0 ~ 5.5 V		-30	mA	
			EVDD = 2.7 ~ 5.5 V		-15	mA	
		P50 ~ P55, P90, P91, P96 ~ P99, P913 ~ P915, PCM0, PCM1, PDL0 ~ PDL7 的总和	EVDD = 4.0 ~ 5.5 V		-30	mA	
EVDD = 2.7 ~ 5.5 V			-15	mA			
输出电流, 低	IOL1	P00 ~ P06, P30 ~ P35, P40 ~ P42, P50 ~ P55, P90, P91, P96 ~ P99, P913 ~ P915, PCM0, PCM1, PDL0 ~ PDL7 的每个引脚			10	mA	
		P38, P39 每个引脚	EVDD = 4.0 ~ 5.5 V		15	mA	
			EVDD = 2.7 ~ 5.5 V		8	mA	
		P00 ~ P06, P30 ~ P35, P40 ~ P42 的总和				30	mA
		P38, P39, P50 ~ P55, P90, P91, P96 ~ P99, P913 ~ P915, PCM0, PCM1, PDL0 ~ PDL7 的总和				30	mA
输入电压, 高	VIH1	注 1	0.7 EVDD		EVDD	V	
	VIH2	注 2	0.8 EVDD		EVDD	V	
	VIH3	P70 ~ P77	0.7 AVREF0		AVREF0	V	
	VIH4 ^{注3}	X1, X2, XT1, XT2	VDD - 0.5		VDD	V	
输入电压, 低	VIL1	注 1	EVSS		0.3 EVDD	V	
	VIL2	注 2	EVSS		0.2 EVDD	V	
	VIL3	P70 ~ P77	AVSS		0.3 AVREF0	V	
	VIL4 ^{注3}	X1, X2, XT1, XT2	VSS		0.4	V	

- 注
1. P00, P01, P30, P41, P98, PCM0, PCM1, PDL0 ~ PDL7 以及它们的复用功能引脚。
 2. RESET, FLMD0, P02 ~ P06, P31 ~ P35, P38, P39, P40, P42, P50 ~ P55, P90, P91, P96, P97, P99, P913 ~ P915 以及它们的复用功能引脚。
 3. 使用外部时钟时。

<R>

备注 除非特别指出, 复用功能引脚的特性与其他端口引脚相同。

DC 特性

(TA = -40 ~ +85°C, VDD = EVDD = AVREF0 = 2.7 ~ 5.5 V, VSS = EVSS = AVSS = 0 V) (2/3)

参数	符号	条件		MIN.	TYP.	MAX.	单位
输出电压, 高	VOH1	注 1	IOH = -2.0 mA, EVDD = 4.0 ~ 5.5 V	EVDD - 1.0		EVDD	V
		注 2	IOH = -0.1 mA, EVDD = 2.7 ~ 5.5 V	EVDD - 0.5		EVDD	V
输出电压, 低	VOL1	注 3	IOl = 2.0 mA ^{注4}	0		0.8	V
	VOL2	P38, P39	IOl = 15 mA, EVDD = 4.0 ~ 5.5 V	0		2.0	V
			IOl = 8 mA, EVDD = 3.0 ~ 5.5 V	0		1.0	V
			IOl = 5 mA, EVDD = 2.7 ~ 5.5 V	0		1.0	V
输入漏电流, 高	ILIH	VIN = VDD				3.0	μA
输入漏电流, 低	ILIL	VIN = 0 V				-3.0	μA
输出漏电流, 高	ILOH	VO = VDD				3.0	μA
输出漏电流, 低	ILOL	VO = 0 V				-3.0	μA
上拉电阻	RL	VIN = 0 V		10	30	100	kΩ

- 注 1. P00 ~ P06, P30 ~ P35, P40 ~ P42 以及它们的复用功能引脚总和: IOH = -30 mA, P50 ~ P55, P90, P91, P96 ~ P99, P913 ~ P915, PCM0, PCM1, PDL0 ~ PDL7 它们的复用功能引脚总和: IOH = -30 mA.
2. P00 ~ P06, P30 ~ P35, P40 ~ P42 以及它们的复用功能引脚总和: IOH = -15 mA, P50 ~ P55, P90, P91, P96 ~ P99, P913 ~ P915, PCM0, PCM1, PDL0 ~ PDL7 以及它们的复用功能引脚总和: IOH = -15 mA.
3. P00 ~ P06, P30 ~ P35, P40 ~ P42 以及它们的复用功能引脚总和: IOl = 30 mA, P38, P39, P50 ~ P55, P90, P91, P96 ~ P99, P913 ~ P915, PCM0, PCM1, PDL0 ~ PDL7 以及它们的复用功能引脚总和: IOl = 30 mA.
4. 对于 P38 和 P39 的 IOl 参考 IOl1。

DC 特性

(TA = -40 ~ +85°C, VDD = EVDD = AVREF0 = 2.7 ~ 5.5 V, VSS = EVSS = AVSS = 0 V) (3/3)

参数	符号	条件	MIN.	TYP. ^{注2}	MAX.	单位	
电源电压 ^{注1}	IDD1	正常工作模式（所有外围功能工作）					
		f _{XX} = 20 MHz (f _X = 5 MHz) (PLL 模式) V _{DD} = 5 V ±10%		51	70	mA	
		f _{XX} = 10 MHz (时钟直连模式) V _{DD} = 3 V ±10%		17	34	mA	
	IDD2	HALT 模式（所有外围功能工作）					
		f _{XX} = 20 MHz (f _X = 5 MHz) (PLL 模式) V _{DD} = 5 V ±10%		25	38	mA	
		f _{XX} = 10 MHz (时钟直连模式) V _{DD} = 3 V ±10%		9	15	mA	
	IDD3	IDLE 模式（钟表定时器工作）					
		f _X = 5 MHz (PLL 模式关闭) V _{DD} = 5 V ±10%		1.8	2.9	mA	
		f _X = 10 MHz (时钟直连模式) V _{DD} = 3 V ±10%		1.4	2.4	mA	
	IDD4	副时钟运行模式 (f _{XT} = 32.768 kHz) 主振荡停止		240	400	μA	
	IDD5	副 IDLE 模式 (f _{XT} = 32.768 kHz) 钟表定时器运行, 主振荡停止		20	75	μA	
	IDD6	STOP 模式					
		副时钟振荡		15	60	μA	
		副时钟停止 (XT1 = V _{SS} , PSMR.XTSTP 位= 1)		0.1	30	μA	
IDD7	Flash 存储器擦/写 (TA = -40 ~ +85°C)						
	f _{XX} = 20 MHz (f _X = 5 MHz) (PLL 模式) V _{DD} = 5 V ±10%		51	70	mA		
	f _{XX} = 10 MHz (时钟直连模式) V _{DD} = 3 V ±10%		17	34	mA		

- 注
1. V_{DD} 和 EV_{DD} 电流总和（所有端口关闭）。AV_{REF0} 不包括在内。
 2. V_{DD} 典型值如下
V_{DD} = 5.0 V 当 VDD = 5 V ±10% 时
V_{DD} = 3.0 V 当 VDD = 3 V ±10% 时

备注

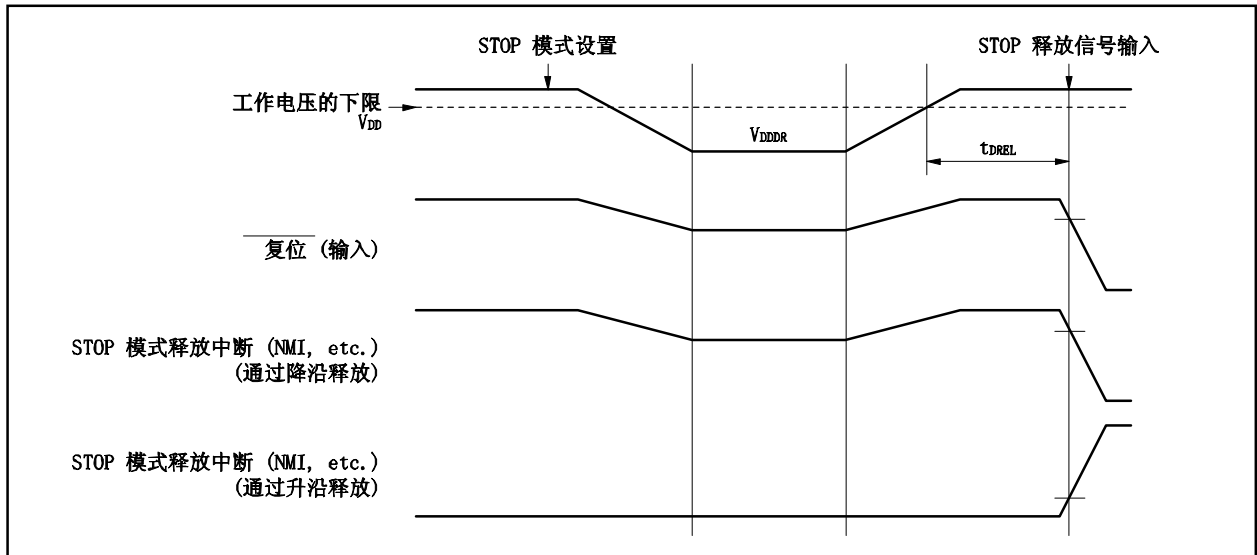
f_{XX}: 主时钟频率
f_X: 主时钟振荡频率
f_{XT}: 副时钟频率

数据保持特性

STOP 模式 (Ta = -40 ~ +85°C)

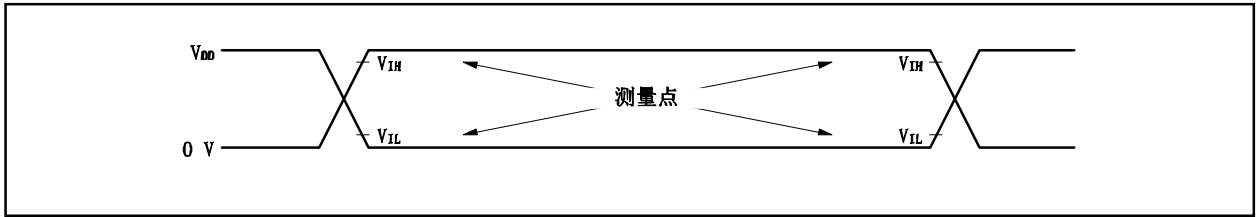
参数	符号	条件	MIN.	TYP.	MAX.	单位
数据保持电压	V _{DDDR}	STOP 模式	2.0		5.5	V
STOP 释放信号输入时间	t _{DREL}		0			μs

注意事项 转换到 STOP 模式和从 STOP 模式还原必须在额定的工作范围完成。

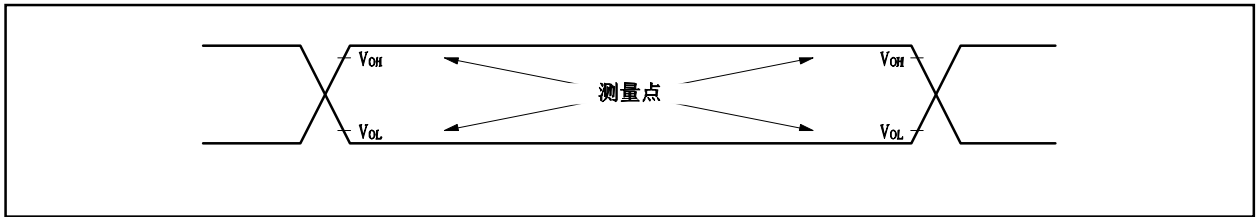


AC 特性

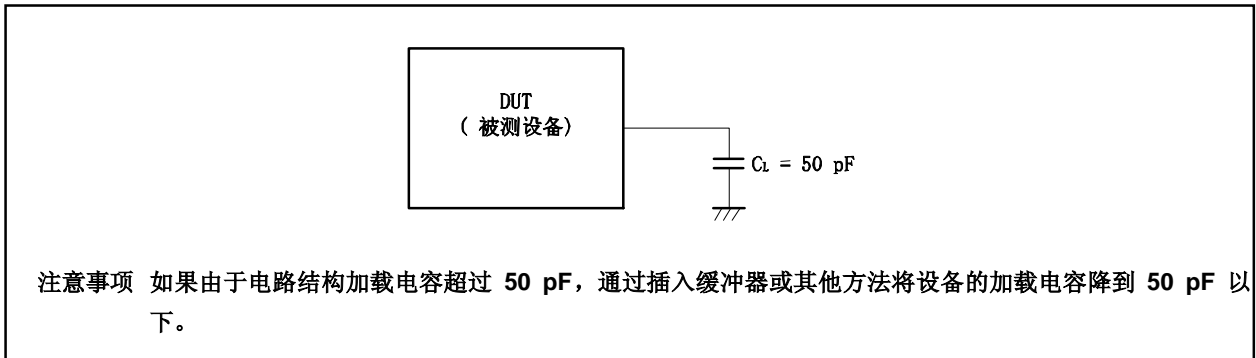
AC 输入测试点 (V_{DD} , AV_{REF0} , EV_{DD})



AC 输出测试点



加载条件

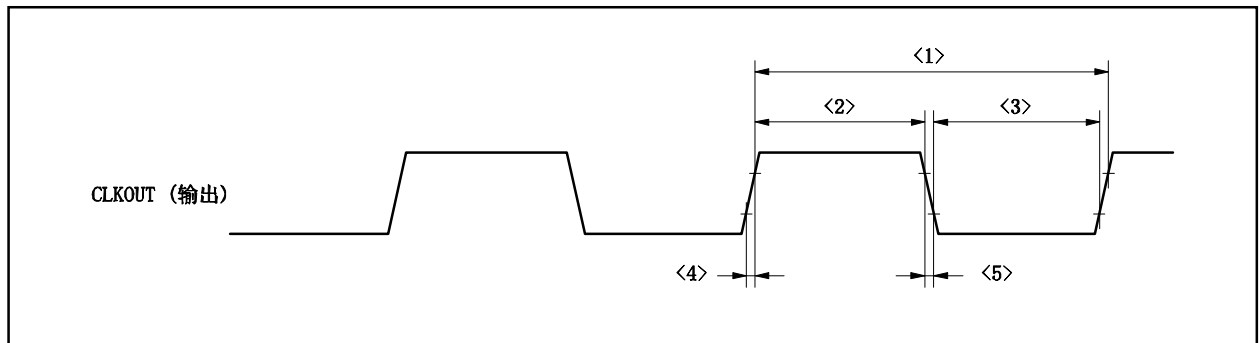


CLKOUT 输出时序

($T_A = -40 \sim +85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.7 \sim 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号		条件	MIN.	MAX.	单位
输出周期	t_{CYK}	<1>		50 ns	30.6 μs	
高电平宽度	t_{WKH}	<2>	$V_{DD} = 4.0 \sim 5.5\text{ V}$	$t_{CYK} / 2 - 17$		ns
			$V_{DD} = 2.7 \sim 5.5\text{ V}$	$t_{CYK} / 2 - 26$		ns
低电平宽度	t_{WKL}	<3>	$V_{DD} = 4.0 \sim 5.5\text{ V}$	$t_{CYK} / 2 - 17$		ns
			$V_{DD} = 2.7 \sim 5.5\text{ V}$	$t_{CYK} / 2 - 26$		ns
上升时间	t_{KR}	<4>	$V_{DD} = 4.0 \sim 5.5\text{ V}$		17	ns
			$V_{DD} = 2.7 \sim 5.5\text{ V}$		26	ns
下降时间	t_{KF}	<5>	$V_{DD} = 4.0 \sim 5.5\text{ V}$		17	ns
			$V_{DD} = 2.7 \sim 5.5\text{ V}$		26	ns

时钟时序



基本操作

(1) 复位/外部中断时序

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.7 \sim 5.5 \text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0 \text{ V}$, $C_L = 50 \text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
RESET 低电平宽度	t_{WRSL1}	<87> 开启状态下的复位	2		μs
	t_{WRSL2}	<88> 开启复位	2		μs
NMI 高电平宽度	t_{WNIH}	<89> 模拟噪声消除	1		μs
NMI 低电平宽度	t_{WNIL}	<90> 模拟噪声消除	1		μs
INTPn 高电平宽度	t_{WITH}	<91> $n = 0 \sim 7$ (模拟噪声消除)	600		ns
		$n = 3$ (选择数字噪声消除时)	$N_i \times t_{ISMP} + 200$		ns
INTPn 低电平宽度	t_{WITL}	<92> $n = 0 \sim 7$ (模拟噪声消除)	600		ns
		$n = 3$ (选择数字噪声消除时)	$N_i \times t_{ISMP} + 200$		ns
ADTRG 高电平脉宽	t_{WADH}	<93> $V_{DD} = 4.0 \sim 5.5 \text{ V}$	$T + 50$		ns
		$V_{DD} = 2.7 \sim 5.5 \text{ V}$	$T + 100$		ns
ADTRG 低电平宽度	t_{WADL}	<94> $V_{DD} = 4.0 \sim 5.5 \text{ V}$	$T + 50$		ns
		$V_{DD} = 2.7 \sim 5.5 \text{ V}$	$T + 100$		ns

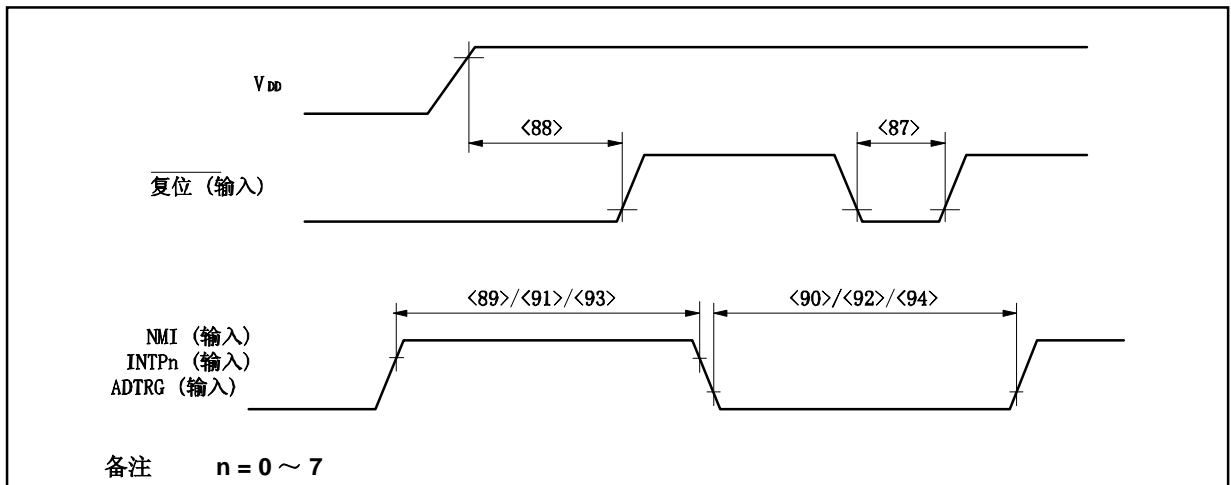
备注 1. N_i : 采样数由 NFC.NFSTS 位设定

t_{ISMP} : INTP3 引脚的数字噪声消除采样时钟周期

T : A/D 基准时钟周期 (f_{AD})

2. 以上指出了可被精确检测为有效边沿的脉冲宽度。因此如果脉宽小于以上值，仍然可能被检测为有效边沿。

复位/中断



定时器定时

$T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.7 \sim 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位	
TI01 高电平宽度	t_{TI0H}	<95>	$V_{DD} = 4.5 \sim 5.5\text{ V}$	$2T_{\text{smp}0} + 100$ ^{注1}		ns
			$V_{DD} = 2.7 \sim 5.5\text{ V}$	$2T_{\text{smp}0} + 200$ ^{注1}		ns
TI01 低电平宽度	t_{TI0L}	<96>	$V_{DD} = 4.5 \sim 5.5\text{ V}$	$2T_{\text{smp}0} + 100$ ^{注1}		ns
			$V_{DD} = 2.7 \sim 5.5\text{ V}$	$2T_{\text{smp}0} + 200$ ^{注1}		ns
TI5m 高电平宽度	t_{TI5H}	<97>	$V_{DD} = 4.5 \sim 5.5\text{ V}$	50		ns
			$V_{DD} = 2.7 \sim 5.5\text{ V}$	100		ns
TI5m 低电平宽度	t_{TI5L}	<98>	$V_{DD} = 4.5 \sim 5.5\text{ V}$	50		ns
			$V_{DD} = 2.7 \sim 5.5\text{ V}$	100		ns
TIP0m 高电平宽度	t_{TIPH}	<99>	$V_{DD} = 4.5 \sim 5.5\text{ V}$	$n_p \times T_{\text{smp}p} + 100$ ^{注2}		ns
			$V_{DD} = 2.7 \sim 5.5\text{ V}$	$n_p \times T_{\text{smp}p} + 200$ ^{注2}		ns
TIP0m 低电平宽度	t_{TIPL}	<100>	$V_{DD} = 4.5 \sim 5.5\text{ V}$	$n_p \times T_{\text{smp}p} + 100$ ^{注2}		ns
			$V_{DD} = 2.7 \sim 5.5\text{ V}$	$n_p \times T_{\text{smp}p} + 200$ ^{注2}		ns

注 1. $T_{\text{smp}0}$: 定时器 0 计数时钟周期

然而, 当 TI0n 配置为外部时间计数输入时, $T_{\text{smp}0} = 4/f_{XX}$ 。

2. n_p : 采样时钟个数, 由 PmNFC.PmNFSTS 位设定。

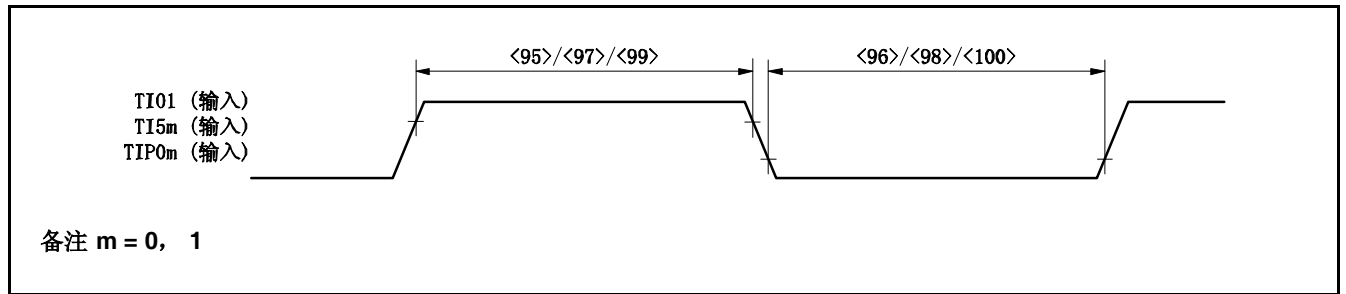
$T_{\text{smp}p}$: TIP0m 引脚数字噪声消除采样时钟周期。

然而如果 TIP00 配置为外部时间计数输入或外部触发输入, 则 $T_{\text{smp}p} = 0$ (数字噪声不消除)。

备注 1. $m = 0, 1$

2. 以上指出了可被精确检测为有效边沿的脉冲宽度。因此如果脉宽小于以上值, 仍然可能被检测为有效边沿。

定时器输入



UART 时序

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.7 \sim 5.5 \text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0 \text{ V}$, $C_L = 50 \text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
发送速率				312.5	kbps
ASCK0 频率		$V_{DD} = 4.5 \sim 5.5 \text{ V}$		12	MHz
		$V_{DD} = 2.7 \sim 5.5 \text{ V}$		6	MHz

CSIO 时序

(1) 主设备模式

(TA = -40 ~ +85°C, VDD = EVDD = AVREF0 = 2.7 ~ 5.5 V, VSS = EVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	MIN.	MAX.	单位	
SCK0n 周期时间	t _{KCY1}	<101>	V _{DD} = 4.0 ~ 5.5 V	200		ns
			V _{DD} = 2.7 ~ 5.5 V	400		ns
SCK0n 高/低电平宽度	t _{KH1} , t _{KL1}	<102>		t _{KCY1} /2 - 30		ns
SI0n 建立时间 (到 SCKBn)	t _{SIK1}	<103>	V _{DD} = 4.0 ~ 5.5 V	30		ns
			V _{DD} = 2.7 ~ 5.5 V	50		ns
SI0n 保持时间 (从 SCK0n)	t _{KSI1}	<104>	V _{DD} = 4.0 ~ 5.5 V	30		ns
			V _{DD} = 2.7 ~ 5.5 V	50		ns
从 SCK0n 到 SO0n 输出的延迟时间	t _{KSO1}	<105>	V _{DD} = 4.0 ~ 5.5 V		30	ns
			V _{DD} = 2.7 ~ 5.5 V		60	ns

备注 n = 0, 1

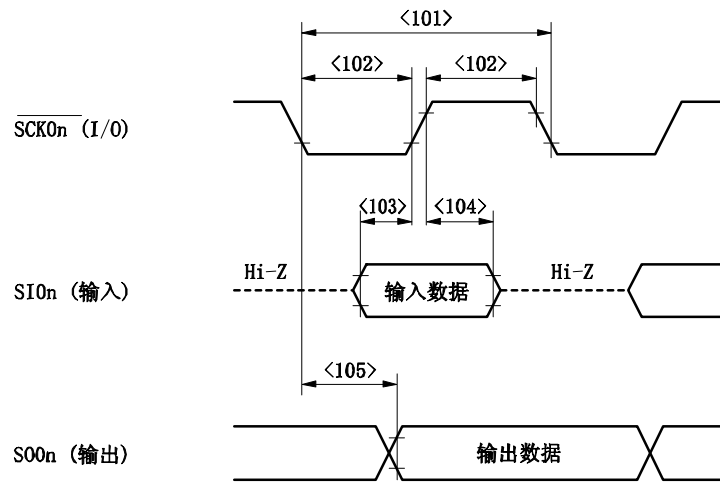
(2) 从设备模式

(TA = -40 ~ +85°C, VDD = EVDD = AVREF0 = 2.7 ~ 5.5 V, VSS = EVSS = AVSS = 0 V, CL = 50 pF)

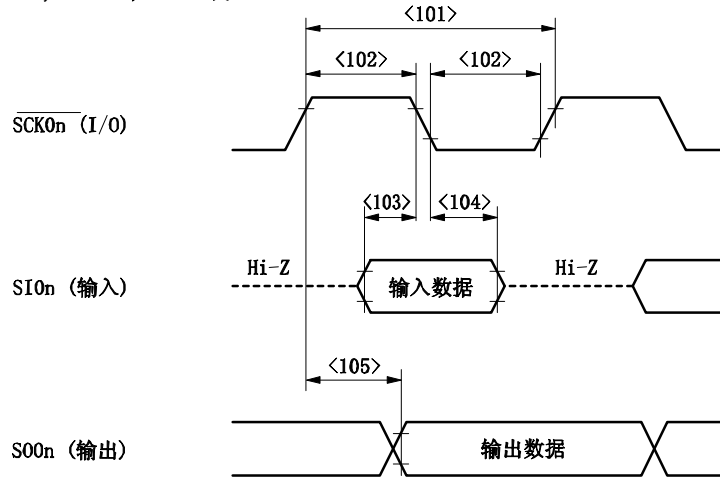
参数	符号	条件	MIN.	MAX.	单位	
SCK0n 周期时间	t _{KCY2}	<101>	V _{DD} = 4.0 ~ 5.5 V	200		ns
			V _{DD} = 2.7 ~ 5.5 V	400		ns
SCK0n 高/低电平宽度	t _{KH2} , t _{KL2}	<102>	V _{DD} = 4.0 ~ 5.5 V	45		ns
			V _{DD} = 2.7 ~ 5.5 V	90		ns
SI0n 建立时间 (到 SCK0n)	t _{SIK2}	<103>	V _{DD} = 4.0 ~ 5.5 V	30		ns
			V _{DD} = 2.7 ~ 5.5 V	60		ns
SI0n 保持时间 (从 SCK0n)	t _{KSI2}	<104>	V _{DD} = 4.0 ~ 5.5 V	30		ns
			V _{DD} = 2.7 ~ 5.5 V	60		ns
从 SCK0n 到 SO0n 输出的延迟时间	t _{KSO2}	<105>	V _{DD} = 4.0 ~ 5.5 V		50	ns
			V _{DD} = 2.7 ~ 5.5 V		100	ns

备注 n = 0, 1

(a) CSICn.CKPn, DAPn 位= 00 或 11



(b) CSICn.CKPn, DAPn 位= 01 或 10



备注 n = 0, 1

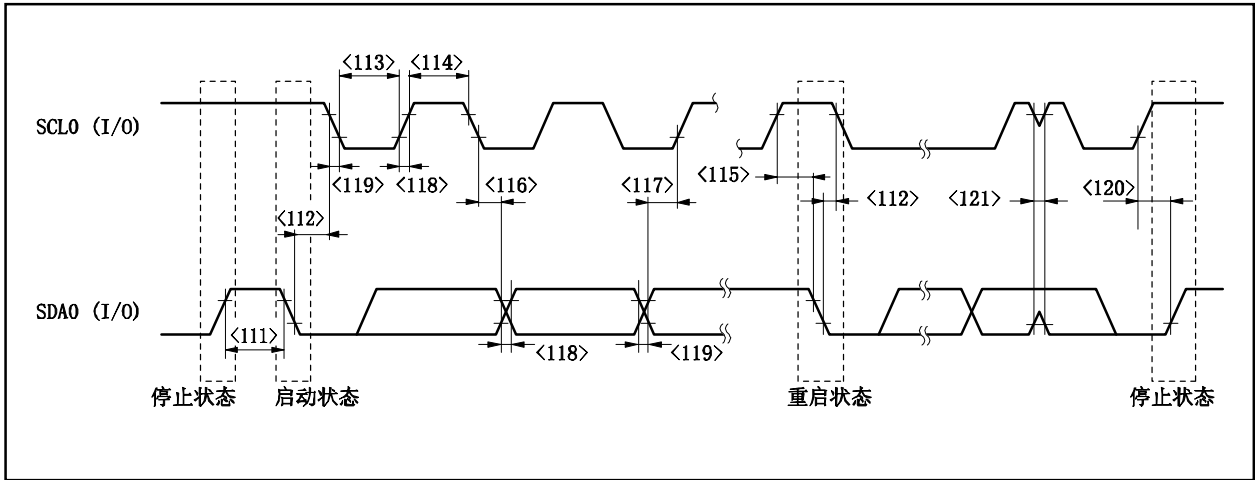
I²C 总线模式(T_A = -40 ~ +85°C, V_{DD} = EV_{DD} = AV_{REF0} = 2.7~5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数		符号		正常模式		高速模式		单位
				MIN.	MAX.	MIN.	MAX.	
SCL0 时钟频率		f _{CLK}		0	100	0	400	kHz
总线自由时间 (从启动到停止的条件)		t _{BUF}	<111>	4.7	-	1.3	-	μs
保持时间 ^{注1}		t _{HD: STA}	<112>	4.0	-	0.6	-	μs
SCL0 时钟低电平宽度		t _{LOW}	<113>	4.7	-	1.3	-	μs
SCL0 时钟高电平宽度		t _{HIGH}	<114>	4.0	-	0.6	-	μs
启动/重启条件的建立时间		t _{SU: STA}	<115>	4.7	-	0.6	-	μs
数据保持时间	CBUS 兼容主设备	t _{HD: DAT}	<116>	5.0	-	-	-	μs
	I ² C 模式			0 ^{注2}	-	0 ^{注2}	0.9 ^{注3}	μs
数据建立时间		t _{SU: DAT}	<117>	250	-	100 ^{注4}	-	ns
SDA0 和 SCL0 信号上升时间		t _r	<118>	-	1000	20 + 0.1Cb ^{注5}	300	ns
SDA0 和 SCL0 信号下降时间		t _f	<119>	-	300	20 + 0.1Cb ^{注5}	300	ns
停止条件建立时间		t _{SU: STO}	<120>	4.0	-	0.6	-	μs
输入滤波器尖峰抑制的脉冲宽度		t _{SP}	<121>	-	-	0	50	ns
每个总线的加载电容		Cb		-	400	-	400	pF

注

- 在起始条件中，第一个时钟脉冲产生在保持时间后。
- 系统内 SDA0n (在 SCL0n 信号的 V_{IHmin.}) 信号需要一个最少 300ns 的保持时间，以使在 SCL0n 的下降沿时占用未定义区域。
- 如果系统不延长 SCL0n 信号低的保持时间 (t_{LOW})，仅需要满足最大数据保持时间 (t_{HD: DAT})。
- 高速模式 I²C 总线可以用于正常模式 I²C 总线系统。在这种情况下，设定高速模式 I²C 总线，因此会遇到如下条件。
 - 如果系统不延长 SCL0n 信号低的保持时间：
t_{SU: DAT} ≥ 250 ns
 - 如果系统延长 SCL0n 信号低的保持时间：
发送如下数据位到 SDA0n 优先于 SCL0n 释放 (t_{rmax.} + t_{SU: DAT} = 1, 000 + 250 = 1, 250 ns: 正常模式 I²C 总线规范)。
- Cb: 一个总线的电容总和 (单位: pF)

I²C 总线模式



A/D 转换器

(TA = -40 ~ +85°C, VDD = EVDD = AVREF0 = 2.7 ~ 5.5 V, VSS = EVSS = AVSS = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位	
分辨率			10	10	10	位	
总误差 ^{注1}	AINL	4.0 ≤ AVREF0 ≤ 5.5 V		±0.2	±0.4	%FSR	
		2.7 ≤ AVREF0 ≤ 4.0 V		±0.3	±0.6	%FSR	
转换时间	tCONV	4.5 ≤ AVREF0 ≤ 5.5 V	高速模式	3.0		100	μs
			正常模式	14.0		100	μs
		4.0 ≤ AVREF0 ≤ 4.5 V	高速模式	4.8		100	μs
			正常模式	14.0		100	μs
		2.85 ≤ AVREF0 ≤ 4.0 V	高速模式	6.0		100	μs
			正常模式	17.0		100	μs
		2.7 ≤ AVREF0 ≤ 2.85 V	高速模式	14.0		100	μs
			正常模式	17.0		100	μs
零度误差 ^{注1}	Ezs	4.0 ≤ AVREF0 ≤ 5.5 V			±0.4	%FSR	
		2.7 ≤ AVREF0 ≤ 4.0 V			±0.6	%FSR	
满度误差 ^{注1}	Efs	4.0 ≤ AVREF0 ≤ 5.5 V			±0.4	%FSR	
		2.7 ≤ AVREF0 ≤ 4.0 V			±0.6	%FSR	
非线性误差 ^{注2}	ILE	4.0 ≤ AVREF0 ≤ 5.5 V			±2.5	LSB	
		2.7 ≤ AVREF0 ≤ 4.0 V			±4.5	LSB	
微分线性误差 ^{注2}	DLE	4.0 ≤ AVREF0 ≤ 5.5 V			±1.5	LSB	
		2.7 ≤ AVREF0 ≤ 4.0 V			±2.0	LSB	
模拟输入电压	VIAN		0		AVREF0	V	
AVREF0 电流	IAREF0	使用 A/D 转换时		1.3	2.5	mA	
		不用 A/D 转换时注 3		1.0	10	μA	

- 注 1. 不包括量化误差 (±0.05 %FSR)。
 2. 不包括量化误差 (±0.5 LSB)。
 3. ADM.ADCS 位 = 0, ADM.ADCS2 位 = 0

备注 LSB: 最小有效位
 FSR: 满度范围

Flash 存储器编程特性

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.7 \sim 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

(1) 基本特性

参数	符号	条件	MIN.	TYP.	MAX.	单位
编程工作频率		$V_{DD} = 4.5 \sim 5.5\text{ V}$	2		20	MHz
		$V_{DD} = 4.0 \sim 5.5\text{ V}$	2		16	MHz
		$V_{DD} = 2.7 \sim 5.5\text{ V}$	2		10	MHz
电源电压	V_{DD}		2.7		5.5	V
重写次数	C_{ERWR}	注	100			次
编程温度	t_{PRG}		-40		+85	$^\circ\text{C}$

注 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。

示例 (P: 写入, E: 擦除)

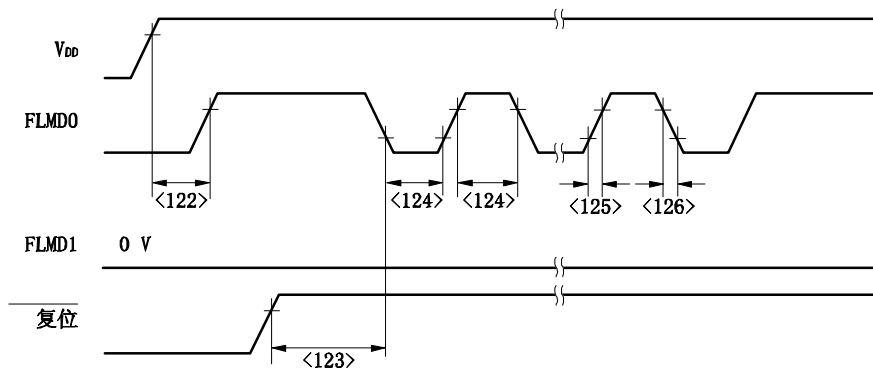
出库产品 → P → E → P → E → P: 3 次重写

出库产品 → E → P → E → P → E → P: 3 次重写

(2) 串行写入操作特性

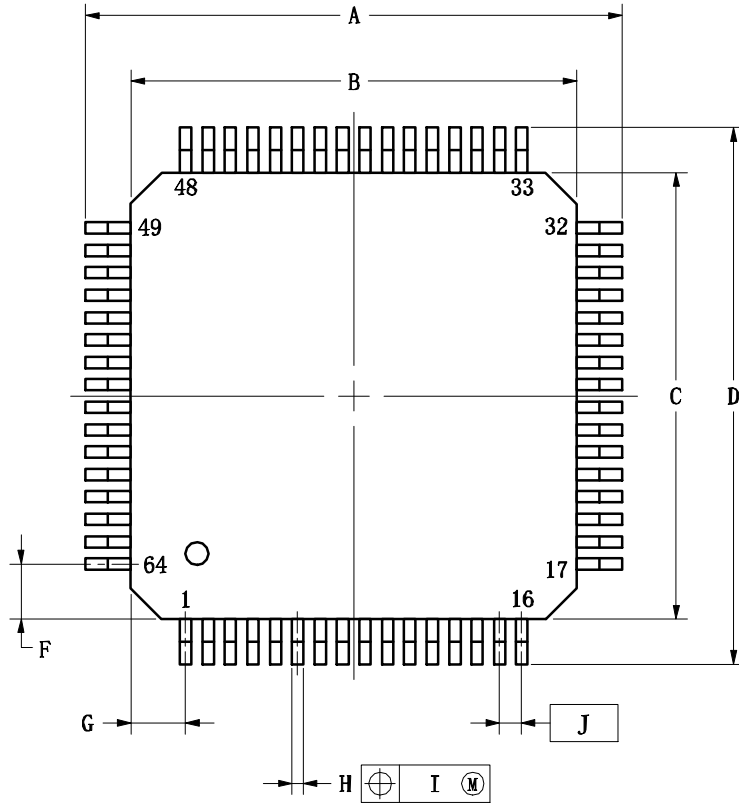
参数	符号		条件	MIN.	TYP.	MAX.	单位
建立时间从 $V_{DD}\uparrow$ 到 $FLMD0\uparrow$	t_{DP}	<122>		10 ms		3 s	
从 $RESET\uparrow$ 到 $FLMD0$ 脉冲输入开始时间	t_{RP}	<123>		$66611.2/f_X$			s
$FLMD0$ 脉冲高电平宽度/低电平宽度	t_{PW}	<124>		10		100	μs
$FLMD0$ 脉冲上升时间	t_R	<125>				50	ns
$FLMD0$ 脉冲下降时间	t_F	<126>				50	ns

串行写入操作时序

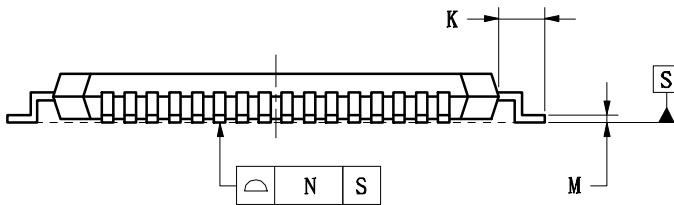
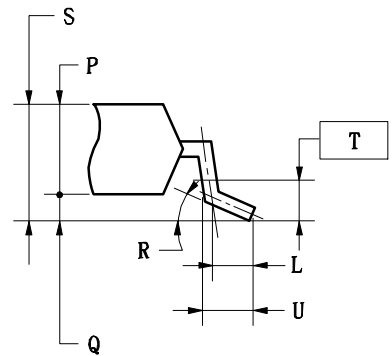


第二十四章 封装图

64- 引脚 塑封 (10x10)



引脚详端图



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0.13mm(T.P.)内。

项目	尺寸
A	12.0±0.2
B	10.0±0.2
C	10.0±0.2
D	12.0±0.2
F	1.25
G	1.25
H	0.22±0.05
I	0.08
J	0.5 (T.P.)
K	1.0±0.2
L	0.5
M	0.17 ^{+0.03} _{-0.07}
N	0.08
P	1.4
Q	0.1±0.05
R	3° ^{+4°} _{-3°}
S	1.5±0.10
T	0.25
U	0.6±0.15

S64GB-50-8EU-2

第二十五章 推荐焊接条件

V850ES/KE2 应在如下推荐条件下焊接和装配。

有关技术信息的更多细节，敬请参阅如下网站。

半导体设备装配手册 (<http://www.necel.com/pkg/en/mount/index.html>)

表 25-1. 贴装焊接条件

μPD70F3726GB-8EU-A: 64 引脚塑封 LQFP (窄节距) (10 × 10)

焊接方法	焊接条件	推荐型号
红外线	最高温度: 260°C, 时间: 最多 60 秒 (220°C 或更高)。 次数: 小于等于 3 次, 暴露限制: 7 天 ^注 (之后在 125°C 预烘 20 ~ 72 小时)	IR60-207-3
局部加热	引脚温度: 最大 350°C, 时间: 最长 3 秒/行针	—

注 干燥下打开后, 存储在低于 25°C 和 65% RH 下, 或减少存储时间。

注意事项 不要一起使用不同的焊接方式 (除局部加热外)。

备注

1. 零件号以 A 结尾的产品是无铅产品。
2. 关于软焊方法和状况的更多细节, 请联系 NEC 电子客服代理。

附录 A 开发工具

在使用 V850ES/KE2 的系统开发中可应用如下开发工具。

图 A-1 所示为开发工具的组成。

- **支持 PC98-NX 系列**

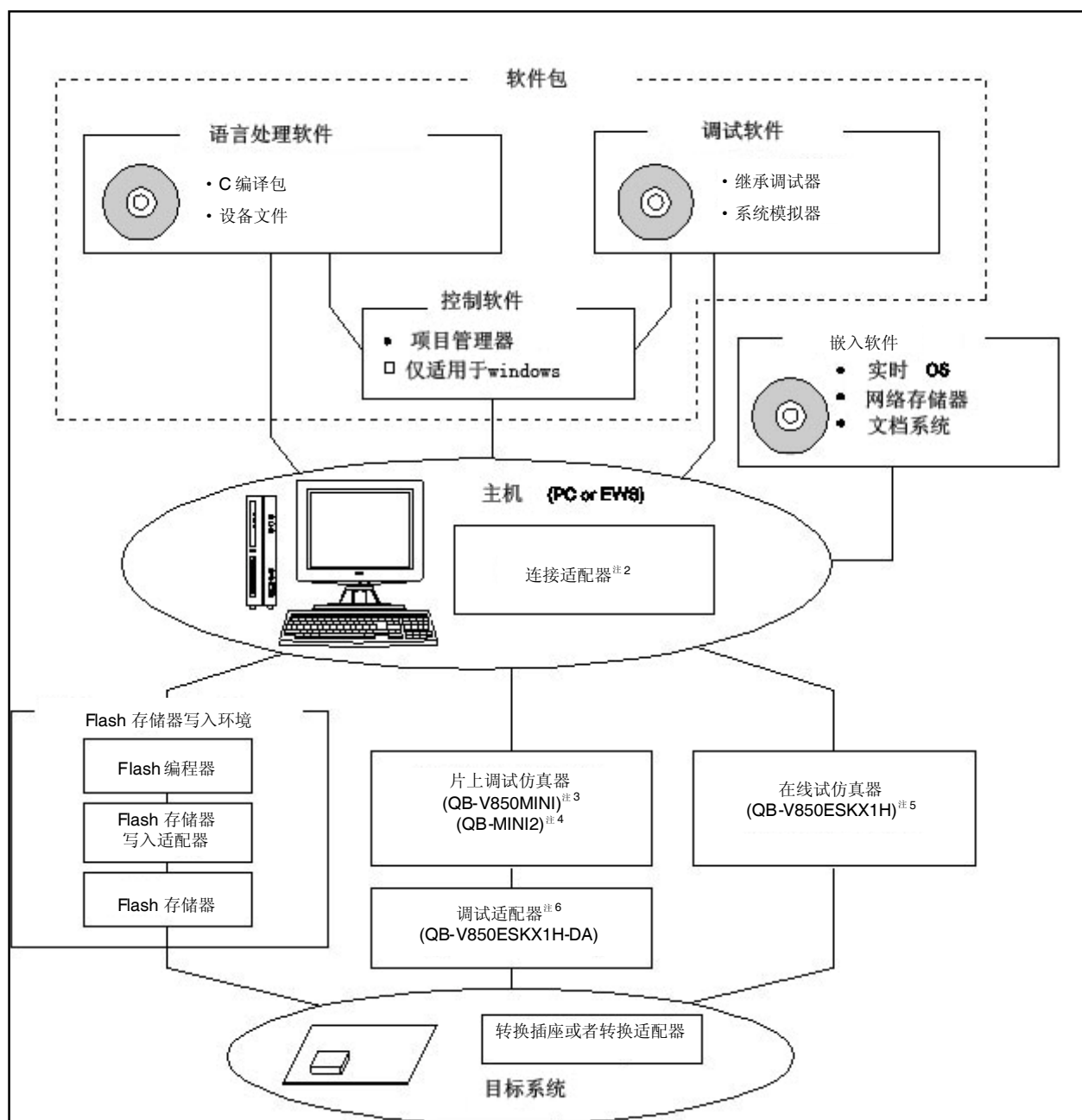
除非特别说明，IBM PC/AT™ 兼容机支持的产品与 PC98-NX 系列计算机是兼容的。当使用 PC98-NX 系列计算机时，参阅 IBM PC/AT 兼容机的使用说明。

- **Windows™**

除非特别说明，“Windows”指以下几种操作系统。

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

图 A-1. 开发工具的组成



- 注
1. 程序管理器 PM+包含在 C 编译包里。
PM+只能在 windows 中使用。
 2. QB-V850MINI, QB-MINI2, 和 QB-V850ESKX1H 只支持 USB 接口。
 3. QB-V850MINI 提供了 ID850QB, USB 接口线, OCD 线, 自检测板, KEL 适配器, 以及 KEL 连接器其他产品均为可选产品。
 4. QB-MINI2 提供了 USB 接口线, 16 引脚目标线, 10 引脚目标线, 以及 78K0-OCD 板(不提供集成调试器) 其他产品均为可选产品。
 5. QB-V850ESKX1H 提供了 ID850QB, 闪存编写器 PG-FPL, 电源单元和 USB 接口适配器。其他产品均为可选产品。
 6. 仅当使用 MINICUBE (QB-V850MINI)时有此需要。

A.1 软件包

SP850 V850 微控制器系列软件包	本软件包包含适用于 V850 微控制器 系列开发工具（软件） 产品型号： $\mu S_{xxxx}SP850$
-------------------------	---

备注 产品型号中的 xxxx 随主机和使用的 OS 而变化。

$\mu S_{xxxx}SP850$

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

A.2 语言处理软件

CA850 C 编译器包	此编译器将 C 语言程序转换为微控制器可执行的目标代码。此编译器由项目管理 PM+ 启动。 产品型号： $\mu S_{xxxx}CA703000$
DF703734 设备文件	该文件包含设备特有的信息。 该设备文件应结合工具(CA850, V850ES/Kx2 的 SM+, 和 ID850 QB)一起使用。 相应的 OS 和主机随使用工具而变化。

备注 产品型号中的 xxxx 随主机和使用的 OS 而变化。

$\mu S_{xxxx}CA703000$

xxxx	主机	OS	存储介质
AB17	PC-9800 系列，	Windows (日文版)	CD-ROM
BB17	IBM PC/AT 兼容机	Windows (英文版)	
3K17	SPARC 站™	SunOS™ (参考 4.1.4), Solaris™ (参考 2.5.1)	

A.3 控制软件

PM+ 项目管理	这是一个控制软件，可使用户在 Windows 环境下有效地进行软件开发。用户程序开发中进行的任何操作，如启动编辑器、构造程序和启动调试器，都可以由 PM+ 执行。 <注意事项> PM+ 包含在 C 编译器安装包 CA850。 仅在 Windows 下使用。
-------------	---

A.4 调试工具 (硬件)

A.4.1 当使用 IECUBE® QB-V850ESKX1H 时

当连接 QB-V850ESKX1H 到主机(PC-9800 系列, PC/AT 兼容机)时的系统配置如下所示。即使没有准备可选产品, 也可进行连接。

图 A-2. 系统配置 (使用 QB-V850ESKX1H) (1/2)

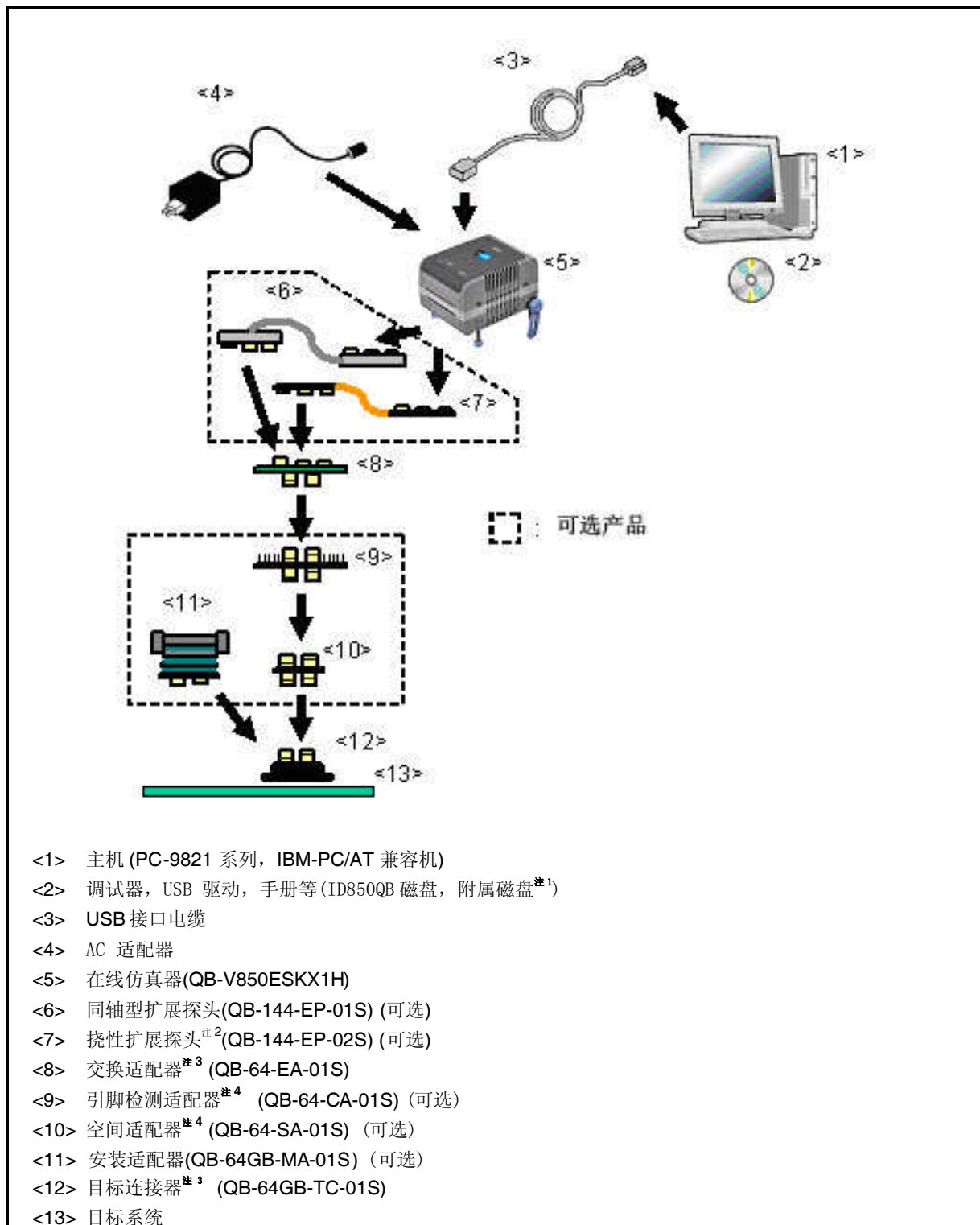


图 A-2. 系统配置 (用于 QB-V850ESKX1H) (2/2)

注	<ol style="list-style-type: none"> 1. 从 NEC Electronics 网站下载设备文件。 http: //www.necel.com/micro/ods/eng/index.html 2. 开发中 3. 依照订购的数字提供器件。 <ul style="list-style-type: none"> • 订购 QB-V850ESKX1H-ZZZ 时 不提供交换适配器和目标连接器。 • 订购 QB-V850ESKX1H-S64GB 时 提供 QB-64GB-EA-01S 和 QB-64GB-TC-01S。 4. 当同时使用<9>和<10>时, 不必关心<9>和<10>之间的顺序。
----------	---

<5> QB-V850ESKX1H [■] 在线仿真器	该在线仿真器用于使用 V850 ES/KE2 开发的应用系统的硬件和软件调试。它支持集成调试 ID850 QB。该仿真器应当与电源和仿真头结合使用。使用 USB 接口电缆连接该仿真器和主机。
<3> USB 接口电缆	连接主机与 QB-V850ESKX1H 的电缆。
<4> AC 适配器	通过更换 AC 插座可支持 100 到 240V 电压。
<8> QB-64-EA-01S 交换适配器	适配器进行引脚转换。
<9> QB-64-CA-01S 检查引脚适配器	使用示波镜等用于波形检测的适配器。
<10>QB-64-SA-01S 空间适配器	适配器调整高度。
<11>QB-64GB-MA-01S 安装适配器	通过插座安装 V850ES/KE2 的适配器。
<12>QB-64GB-TC-01S 目标连接器	到目标系统上焊料的接头。

注 QB-V850ESKX1H 提供了电源单元, USB 接口电缆和闪存烧写器 PG-FPL。它还提供了集成调试器 ID850QB 作为控制软件。

备注 角括号中的数字对应图 A-2 中的数字。

■

A.4.2 当使用 MINICUBE QB-V850MINI 时

(1) 使用 MINICUBE 和 QB-V850ESKX1H-DA 的调试模拟器。

当连接调试适配器 QB-V850ESKX1H-DA 及 MINICUBE 到主机(PC-9821 系列, PC/AT 兼容机)时的系统配置如下所示。即使没有准备可选产品, 仍可连接。

图 A-3. 系统配置 (用于 QB-V850ESKX1H-DA) (当使用可选产品时)

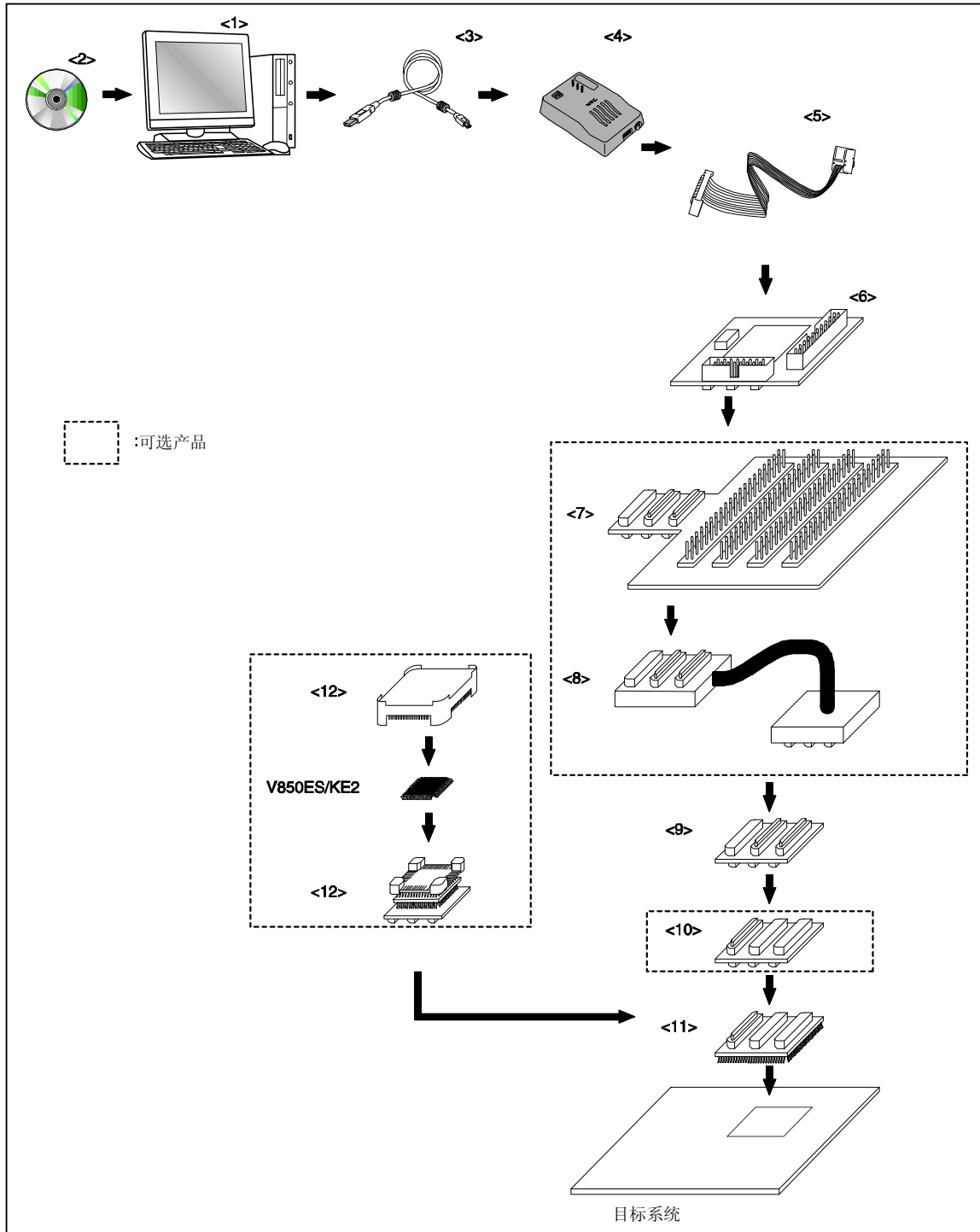
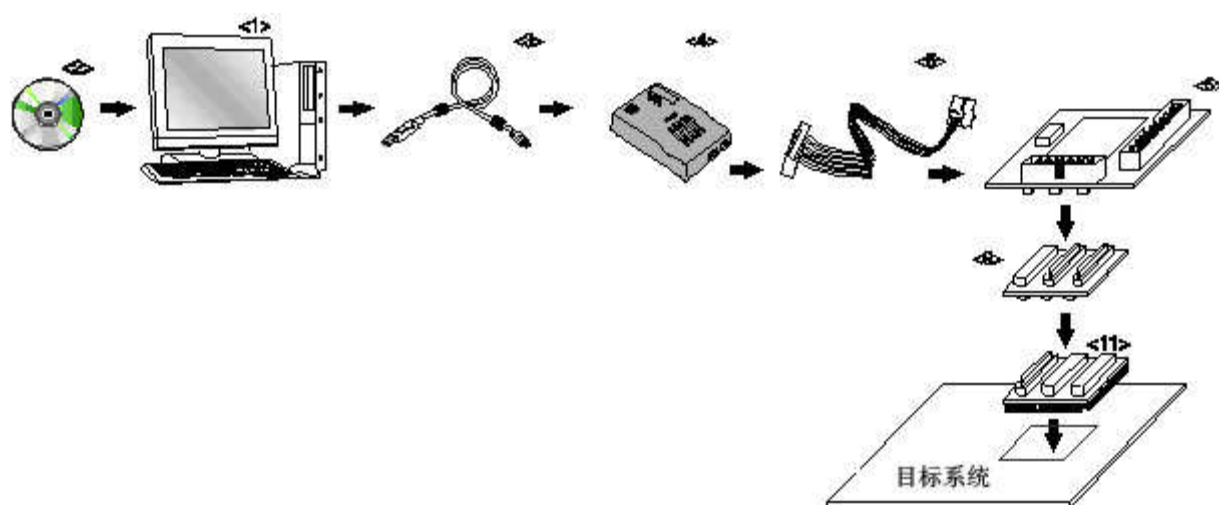


图 A-4. 系统配置（用于 QB-V850ESKX1H-DA）（当使用可选产品时）



<1> 主机	带有 USB 端口的 PC
<2> CD-ROM ^注	CD-ROM 包括集成调试器 ID850QB, N 道线检测器, 设备驱动, 以及文件, 并自带 MINICUBE。
<3> USB 接口电缆	USB 线连接主机及 MINICUBE, USB 线自带 MINICUBE。线长大约为 2m。
<4> MINICUBE 片上调试模拟器	当使用 V850ES/KE2 开发应用系统时, 该片上调试模拟器用于调试软件。其自带集成调试器 ID850QB。
<5> OCD 电缆	用于连接 MINICUBE 和目标系统的电缆 与 MINICUBE 一起提供。电缆的长度大约为 20 厘米。
<6> QB-V850ESKX1H-DA 调试适配器	通过与 MINICUBE 一起使用, 该适配器可用作电路模拟器, 其自带 MINICUBE。
<7> QB-64-CA-01S (可选) 检查引脚适配器	使用示波镜等用于波形检测的适配器。
<8> QB-144-EP-01S (可选) 同轴型扩展探头	用探头连接 QB-V850ESKX1H-DA 和交换适配器。线长约 40cm。
<9> QB-64-EA-01S 交换适配器	适配器进行引脚转换。
<10> QB-64-SA-01S (可选) 空间适配器	适配器调节高度。
<11> QB-64GB-TC-01S 目标连接器	到目标系统上焊料的接头。
<12> QB-64GB-MA-01S (可选) 安装适配器	通过插座安装 V850ES/KE2 的适配器。

注 从 NEC Electronics 网站下载设备文件。

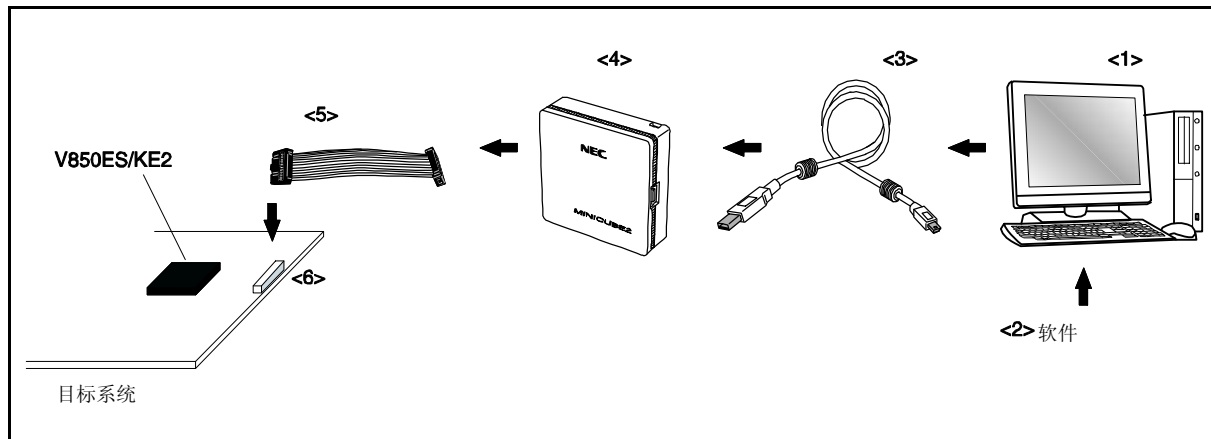
<http://www.necel.com/micro/ods/eng/index.html>

备注 角括号中的数字对应图 A-3 和 A-4 中的数字。

■ A.4.3 当使用 MINICUBE2 QB-MINI2 时

当把 MINICUBE2 和主机(PC-9821 系列, PC/AT 兼容机)连接时, 系统配置显示如下。

图 A-5. 片上仿真系统的系统配置



<1> 主机	使用 USB 端口的 PC 机
<2> 软件	集成调试器 ID850QB, 设备文件, 等等。 从 NEC Electronics 网站下载设备文件。 http://www.necel.com/micro/ods/eng/
<3> USB 接口电缆	使用 USB 电缆连接主机和 MINICUBE2。它提供了 MINICUBE2。电缆长度接近 2m。
<4> MINICUBE2 片上调试仿真器	当使用 V850ES/KE2 开发应用系统时, 该片上调试仿真器提供调试硬件和软件。它支持集成调试器 ID850QB。
<5> 16 引脚目标电缆	使用电缆连接 MINICUBE2 和目标系统。 它提供了 MINICUBE2。电缆长度接近 15 cm。
<6> 目标连接器(独立销售)	使用脚距为 2.54mm 的 16 引脚的多用连接器。

备注 角括号中的数字对应图 A-5 中的数字。

A.5 调试工具 (软件)

V850ES/Kx2 的 SM+ [®] 系统模拟器 (开发中)	该系统模拟器支持 V850 系列。V850ES/Kx2 和 SM+ 是基于 Windows 的软件。当模拟目标系统操作时用于在主机上调试 C 源程序或汇编程序。使用 V850ES/Kx2 和 SM+ 允许执行逻辑测试应用和基于硬件开发的独立测试，因此提供了很高的开发效率和软件质量。V850ES/Kx2 和 SM+ 应与设备文件一起使用。 产品型号：μSxxxxSM703734-B
ID850QB 集成调试器	此调试器支持 V850 系列在线仿真器。ID850 和 ID850QB 是基于 Windows 的软件。它拥有改良的 C 语言编译调试功能，并且能够通过结合源程序的集成窗口功能，分解显示，和存储器显示来显示追踪源程序的结果。改功能需要使用器件文件(独立销售)。 产品型号：μSxxxx ID703000-QB (ID850QB)

备注 产品型号中的 xxxx 使用的 OS 而变化。

μSxxxxID703000-QB

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

A.6 嵌入式软件

RX850, RX850 Pro 实时 OS	RX850 和 RX850 Pro 是依照 μ TRON 3.0 规范的实时操作系统。 提供一个为了产生多个信息表格的工具 (配置器)。 RX850 Pro 比 RX850 具有更多的功能。
	产品型号: μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$ (RX850) μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$ (RX850 Pro)
Applilet ^注	这是一个驱动配置, 它可自动为 V850ES/KE2 生成采样程序。
RX-FS850 (文件系统)	这是 FAT 文件系统功能。 此文件系统支持 CD-ROM 文件系统功能。 此文件系统用在实时 OS RX850 Pro。

注 关于如何获得 Applilet, 请咨询 NEC 电子销售代理。

注意事项 为了购买 RX850 或 RX850 Pro, 首先需要填写购买申请表并且签署用户协议。

备注 产品型号中的 xxxxx 和 $\Delta\Delta\Delta\Delta$ 随主机和使用的 OS 而变化。

μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$

μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$

$\Delta\Delta\Delta\Delta$	产品概述	用于大规模生产的上限
001	评价目标	不用于批量生产
100K	大规模生产目标	10 万单元
001M		100 万单元
010M		1000 万单元
S01	源程序	量产的目标源程序

xxxx	主机	OS	存储介质
AB17	PC-9800 系列,	Windows (日文版)	CD-ROM
BB17	IBM PC/AT 兼容机	Windows (英文版)	
3K17	SPARC 站	Solaris (参考 2.5.1)	

A.7 Flash 存储器写入工具

Flashpro IV (产品型号: PG-FP4) Flash 编程器	Flash 编程器专用于有片上 flash 存储器的微控制器。
QB-MINI2 (MINICUBE2)	通过插座安装 V850ES/KE2 的适配器。
FA-64GB-8EU-A Flash 存储器写入适配器	Flash 存储器写入适配器用于连接 Flashpro IV, 等。(没有配线)
FA-70F3726GB-8EU-MX Flash 存储器写入适配器	Flash 存储器写入适配器用于连接 Flashpro IV, 等。(已经配线)

备注 FA-64GB-8EU-A 和 FA-70F3726GB-8EU-MX 是 Naito Densai Machida Mfg. Co., Ltd 的产品。
电话: +81-42-750-4172

附录 B 指令集列表

B.1 常规指令

(1) 用于描述操作数的寄存器符号

寄存器符号	解释
reg1	通用寄存器：用作源寄存器。
reg2	通用寄存器：主要用作目的寄存器。在有些指令中也用作源寄存器。
reg3	通用寄存器：主要用作存放除法运算结果的余数或乘法运算结果的高 32 位。
bit#3	用于指定位编号的 3 位数据
immX	X 位立即数据
dispX	X 位偏移量数据
regID	系统寄存器编号
vector	用于指定陷阱向量的 5 位数据(00H ~ 1FH)
cccc	表示条件代码的 4 位数据
sp	堆栈指针 (r3)
ep	元素指针 (r30)
listX	X 个寄存器列表

(2) 用于描述操作码的寄存器符号

寄存器符号	解释
R	指定 reg1 或 regID 的 1 位数据码
r	指定 reg2 的 1 位数据码
w	指定 reg3 的 1 位数据码
d	1 位位移数据
l	1 位立即数据 (指示立即数据的高位)
i	1 位立即数据
cccc	表示条件代码的 4 位数据
CCCC	表示 Bcond 指令的条件代码的 4 位数据
bbb	用于指定位编号的 3 位数据
L	用于指定在寄存器列表中的程序寄存器的 1 位数据

(3) 用于描述操作数的寄存器符号

寄存器符号	解释
←	输入到
GR []	通用寄存器
SR []	系统寄存器
zero-extend (n)	扩展 n 个 0 直到字长度
sign-extend (n)	扩展 n 个符号直到字长度
load-memory (a, b)	从地址 a 读数据 b。
store-memory (a, b, c)	写长度为 c 的数据 b 到地址 a。
load-memory-bit (a, b)	读地址 a 的 b 位。
store-memory-bit (a, b, c)	写入 c 到地址 a 的 b 位。
saturated (n)	执行 n 的饱和处理(n 是 2 的反码) 计算结果如下 n ≥ 7FFFFFFFH, 它的值为 7FFFFFFFH. n ≤ 80000000H, 它的值为 80000000H.
result	映射结果在一个标志中
Byte	字节 (8 位)
Halfword	半个字 (16 位)
Word	字 (32 位)
+	加
-	减
	位连接
×	乘法
÷	除法
%	除法结果余数
AND	逻辑与
OR	逻辑或
XOR	异或
NOT	逻辑非
logically shift left by	逻辑左移
logically shift right by	逻辑右移
arithmetically shift right by	算术右移

(4) 用于表示操作时钟的寄存器符号

寄存器符号	解释
i	在指令执行后立即执行另一条指令 (issue)。
r	在指令执行后立即重复同一指令 (repeat)。
l	在指令执行后在指令中立即使用执行结果 (latency)。

(5) 用于描述标志操作的寄存器符号

标识符	解释
(Blank)	不变
0	清零
X	依据结果设置或清零
R	恢复预先保存的值。

(6) 条件代码

条件代码 (cccc)	条件公式	解释
0 0 0 0	$OV = 1$	溢出
1 0 0 0	$OV = 0$	没有溢出
0 0 0 1	$CY = 1$	进位 小于(小于)
1 0 0 1	$CY = 0$	没有进位 不低于(大于或等于)
0 0 1 0	$Z = 1$	零
1 0 1 0	$Z = 0$	非零
0 0 1 1	$(CY \text{ or } Z) = 1$	不大于(小于或等于)
1 0 1 1	$(CY \text{ or } Z) = 0$	大于(大于)
0 1 0 0	$S = 1$	负
1 1 0 0	$S = 0$	正
0 1 0 1	-	一直(无论什么情况下)
1 1 0 1	$SAT = 1$	饱和
0 1 1 0	$(S \text{ xor } OV) = 1$	小于符号
1 1 1 0	$(S \text{ xor } OV) = 0$	大于等于符号
0 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 1$	小于等于符号
1 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 0$	大于符号

B.2 指令集(按字母顺序)

(1/6)

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]	1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiii	GR[reg2]←GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]	1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)	1	1	1		0	×	×	
Bcond	disp9	dddd1011ddcccc 注1	如果条件满足 则 PC←PC+sign-extend(disp9)	当条件满足时	2	2	2				
			当条件不满足时	注2	注2	注2					
BSH	reg2,reg3	rrrrr11111100000 www01101000010	GR[reg3]←GR[reg2] (23 : 16) GR[reg2] (31 : 24) GR[reg2] (7 : 0) GR[reg2] (15 : 8)	1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 www01101000000	GR[reg3]←GR[reg2] (7 : 0) GR[reg2] (15 : 8) GR [reg2] (23 : 16) GR[reg2] (31 : 24)	1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Halfword))	4	4	4					
CLR1	bit#3,disp16[reg1]	10bbb111110RRRRR dddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)	3	3	3				×	
	reg2,[reg1]	rrrrr111111RRRRR 000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)	3	3	3				×	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiii www011000cccc0	if conditions are satisfied then GR[reg3]←sign-extended(imm5) else GR[reg3]←GR[reg2]	1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRR www011001cccc0	if conditions are satisfied then GR[reg3]←GR[reg1] else GR[reg3]←GR[reg2]	1	1	1					
CMP	reg1,reg2	rrrrr001111RRRRR	result←GR[reg2]-GR[reg1]	1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010011iiii	result←GR[reg2]-sign-extend(imm5)	1	1	1	×	×	×	×	
CTRET		000001111100000 0000000101000100	PC←CTPC PSW←CTPSW	3	3	3	R	R	R	R	R
DBRET		000001111100000 0000000101000110	PC←DBPC PSW←DBPSW	3	3	3	R	R	R	R	R

助记符	操作数	操作码	操作	执行时钟			标志					
				i	r	l	CY	OV	S	Z	SAT	
DBTRAP		1111100001000000	DBPC←PC+2 (restored PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3						
DI		000001111100000 0000000101100000	PSW.ID←1	1	1	1						
DISPOSE	imm5,list12	0000011001iiiiL LLLLLLLLLLLL000000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded	n+1 注4	n+1 注4	n+1 注4						
	imm5,list12,[reg1]	0000011001iiiiL LLLLLLLLLLLLRRRRR 注5	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded PC←GR[reg1]	n+3 注4	n+3 注4	n+3 注4						
DIV	reg1,reg2,reg3	rrrrr11111RRRRR wwwww01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×		
DIVH	reg1,reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1] ^{#6}	35	35	35		×	×	×		
	reg1,reg2,reg3	rrrrr11111RRRRR wwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1] ^{#6} GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×		
DIVHU	reg1,reg2,reg3	rrrrr11111RRRRR wwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1] ^{#6} GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×		
DIVU	reg1,reg2,reg3	rrrrr11111RRRRR wwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×		
EI		100001111100000 0000000101100000	PSW.ID←0	1	1	1						
HALT		000001111100000 0000000100100000	Stop	1	1	1						
HSW	reg2,reg3	rrrrr1111100000 wwwww01101000100	GR[reg3]←GR[reg2](15:0) GR[reg2] (31:16)	1	1	1	×	0	×	×		
JARL	disp22,reg2	rrrrr11110dddd dddddddddddddd0 注7	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2						
JMP	[reg1]	0000000011RRRRR	PC←GR[reg1]	3	3	3						
JR	disp22	0000011110dddd dddddddddddddd0 注7	PC←PC+sign-extend(disp22)	2	2	2						
LD.B	disp16[reg1],reg2	rrrrr11100RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	注11						
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR dddddddddddddd1 注s 8, 10	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	注11						

助记符	操作数	操作码	操作	执行时钟			标志					
				i	r	l	CY	OV	S	Z	SAT	
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR dddddddddddddd0 注 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	注 11						
LDSR	reg2,regID	rrrrr111111RRRRR 000000000100000 注 12	SR[regID]←GR[reg2]	1	1	1						
			Other than regID = PSW regID = PSW	1	1	1	×	×	×	×	×	
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR dddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	注 11						
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR dddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)	1	1	注 11						
MOV	reg1,reg2	rrrrr00000RRRRR	GR[reg2]←GR[reg1]	1	1	1						
	imm5,reg2	rrrrr010000iiii	GR[reg2]←sign-extend(imm5)	1	1	1						
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii llllllllllllllllll	GR[reg1]←imm32	2	2	2						
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1						
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 0 ¹⁶)	1	1	1						
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] GR[reg2]←GR[reg2]xGR[reg1] 注 14	1	4	5						
	imm9,reg2,reg3	rrrrr111111iiii wwwww01001111100 注 13	GR[reg3] GR[reg2]←GR[reg2]xsign-extend(imm9)	1	4	5						
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2] ^{#6} xGR[reg1] ^{#6}	1	1	2						
	imm5,reg2	rrrrr010111iiii	GR[reg2]←GR[reg2] ^{#6} xsign-extend(imm5)	1	1	2						
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] ^{#6} ximm16	1	1	2						
MULU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] GR[reg2]←GR[reg2]xGR[reg1] 注 14	1	4	5						
	imm9,reg2,reg3	rrrrr111111iiii wwwww0100111110 注 13	GR[reg3] GR[reg2]←GR[reg2]xzero-extend(imm9)	1	4	5						
NOP		0000000000000000	Pass at least one clock cycle doing nothing.	1	1	1						
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])	1	1	1	0	×	×			
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR dddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)	3	3	3					×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)	3	3	3					×	

助记符	操作数	操作码	操作	执行时钟			标志					
				i	r	l	CY	OV	S	Z	SAT	
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×		
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×		
PREPARE	list12,imm5	0000011110iiiiL LLLLLLLLLLLL00001	Store-memory(sp-4,GR[reg in list12],Word) sp←sp-4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 注4	n+1 注4	n+1 注4						
	list12,imm5, sp/imm ^{#15}	0000011110iiiiL LLLLLLLLLLLLff011 imm16/imm32 注 16	Store-memory(sp-4,GR[reg in list12],Word) sp←sp+4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5) ep←sp/imm	n+2 注4 注7	n+2 注4 注7	n+2 注4 注7						
RETI		000001111100000 0000000101000000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R	
SAR	reg1,reg2	rrrrr11111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×		
	imm5,reg2	rrrrr010101iiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend(imm5)	1	1	1	×	0	×	×		
SASF	cccc,reg2	rrrrr111110cccc 0000001000000000	if conditions are satisfied then GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H else GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1						
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×	
	imm5,reg2	rrrrr010001iiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5))	1	1	1	×	×	×	×	×	
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]-GR[reg1])	1	1	1	×	×	×	×	×	
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]-sign-extend(imm16))	1	1	1	×	×	×	×	×	
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]-GR[reg2])	1	1	1	×	×	×	×	×	
SETF	cccc,reg2	rrrrr111110cccc 0000000000000000	If conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1						

助记符	操作数	操作码	操作	执行时钟			标志					
				i	r	l	CY	OV	S	Z	SAT	
SET1	bit#3,disp16[reg1]	00bbb11110RRRRR dddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 注3	3 注3	3 注3					×	
	reg2,[reg1]	rrrrr11111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 注3	3 注3	3 注3					×	
SHL	reg1,reg2	rrrrr11111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×		
	imm5,reg2	rrrrr010110iiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×		
SHR	reg1,reg2	rrrrr11111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×		
	imm5,reg2	rrrrr010100iiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×		
SLD.B	disp7[ep],reg2	rrrrr0110dddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	注9						
SLD.BU	disp4[ep],reg2	rrrrr0000110ddd 注 18	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	注9						
SLD.H	disp8[ep],reg2	rrrrr1000dddddd 注 19	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	注9						
SLD.HU	disp5[ep],reg2	rrrrr0000111ddd 注 s 18, 20	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	注9						
SLD.W	disp8[ep],reg2	rrrrr1010dddddd0 注 21	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	注9						
SST.B	reg2,disp7[ep]	rrrrr0111dddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1						
SST.H	reg2,disp8[ep]	rrrrr1001dddddd 注 19	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Halfword)	1	1	1						
SST.W	reg2,disp8[ep]	rrrrr1010dddddd1 注 21	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1						
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR dddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1						
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR dddddddddddddd0 注 8	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Halfword)	1	1	1						
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR dddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Word)	1	1	1						
STSR	regID,reg2	rrrrr11111RRRRR 0000000010000000	GR[reg2]←SR[regID]	1	1	1						

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]-GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]-GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	0000000010RRRRR	adr←(PC+2) + (GR [reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr,Halfword)) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15 : 0))	1	1	1					
TRAP	向量	000001111111iiii 0000000100000000	EIPC ←PC+4 (Restored PC) EIPSW ←PSW ECR.EICC ←Interrupt code PSW.EP ←-1 PSW.ID ←-1 PC ←00000040H (当向量是 00H ~ 0FH) 00000050H (当向量是 10H ~ 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr,bit#3))	3 注3	3 注3	3 注3					×
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3 注3	3 注3	3 注3					×
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15 : 0))	1	1	1					

- 注
1. dddddddd: disp9 的高 8 位。
 2. 如果有重写即刻的 PSW 内容的指令，那么时钟数为 3。
 3. 如果没有等待状态(3 + 读取访问状态的数量)。
 4. n 是 list12 加载寄存器的总数。(和等待状态数量一致。同样，如果没有等待状态，n 是 list12 寄存器的总数。如果 n = 0，和 n = 1 时的操作一样)
 5. RRRRR: 00000 除外。
 6. 仅低半字数据有效。
 7. dddddddddddddddddddd: disp22 的高 21 位。
 8. dddddddddddddddd: disp16 的高 15 位。
 9. 根据等待状态的数目(如果没有等待状态为 1)。
 10. b: disp16 的第 0 位。
 11. 根据等待状态的数目(如果没有等待状态为 2)。

- 注** 12. 在这个指令中，为了方便记忆码的描述，源寄存器设为 **reg2**，但是 **reg1** 字段用于操作码。因此，这个寄存器在记忆码描述中的详细意思和在操作码和其它指令不同。
- rrrrr = regID 规范
RRRRR = reg2 规范
13. **iiii**: **imm9** 的低 5 位。
IIII: **imm9** 的高 4 位。
14. 对于通用寄存器 **reg1** 和 **reg3**，不要指定相同的寄存器。
15. **sp/imm**: 由副操作码的位 19 和位 20 指定。
16. **ff = 00**: 在 **ep** 中加载 **sp**。
01: 在 **ep** 中加载符号扩展 16 位立即数(位 47 ~ 32)。
10: 在 **ep** 中加载符号扩展 16 位逻辑左移 16 位立即数(位 47 ~ 32)。
11: 在 **ep** 中加载 32 位立即数(位 63 ~ 32)。
17. 如果 **imm = imm32**, **n + 3** 个时钟。
18. **rrrrr**: 除 00000 外。
19. **ddddddd**: **disp8** 的高 7 位。
20. **dddd**: **disp5** 的高 4 位。
21. **dddddd**: **disp8** 的高 6 位。

附录 C 寄存器索引

(1/6)

符号	名称	单元	页码
ADCR	A/D 转换结果寄存器	ADC	367
ADCRH	A/D 转换结果寄存器 H	ADC	367
ADIC	中断控制寄存器	INTC	535
ADM	A/D 转换模式寄存器	ADC	362
ADS	模拟输入通道指定寄存器	ADC	366
ASIF0	异步串行接口发送状态寄存器 0	UART	393
ASIF1	异步串行接口发送状态寄存器 1	UART	393
ASIM0	异步串行接口模式寄存器 0	UART	390
ASIM1	异步串行接口模式寄存器 1	UART	390
ASIS0	异步串行接口状态寄存器 0	UART	392
ASIS1	异步串行接口状态寄存器 1	UART	392
BRGC0	波特率生成器控制寄存器 0	UART	411
BRGC1	波特率生成器控制寄存器 1	UART	411
BRGIC	中断控制寄存器	INTC	535
CKSR0	时钟选择寄存器 0	UART	410
CKSR1	时钟选择寄存器 1	UART	410
CMP00	8 位定时器 H 比较寄存器 00	TMH	308
CMP01	8 位定时器 H 比较寄存器 01	TMH	309
CMP10	8 位定时器 H 比较寄存器 10	TMH	308
CMP11	8 位定时器 H 比较寄存器 11	TMH	309
CR010	16 位定时器捕捉/比较寄存器 010	TM0	220
CR011	16 位定时器捕捉/比较寄存器 011	TM0	221
CR5	16 位定时器比较寄存器 5	TM5	290
CR50	8 位定时器比较寄存器 50	TM5	290
CR51	8 位定时器比较寄存器 51	TM5	290
CRC01	捕捉/比较控制寄存器 01	TM0	226
CSI0IC0	中断控制寄存器	INTC	535
CSI0IC1	中断控制寄存器	INTC	535
CSIC0	时钟控制的串行接口时钟选择寄存器 0	CSI0	423
CSIC1	时钟控制的串行接口时钟选择寄存器 1	CSI0	423
CSIM00	时钟控制的串行接口模式寄存器 00	CSI0	421
CSIM01	时钟控制的串行接口模式寄存器 01	CSI0	421
CTBP	CALLT 基础指针	CPU	42
CTPC	CALLT 执行状态保存寄存器	CPU	41
CTPSW	CALLT 执行状态保存寄存器	CPU	41
DBPC	异常/调试陷阱状态保存寄存器	CPU	42
DBPSW	异常/调试陷阱状态保存寄存器	CPU	42
ECR	中断源寄存器	CPU	39
EIPC	中断状态保存寄存器	CPU	38
EIPSW	中断状态保存寄存器	CPU	38

符号	名称	单元	页码
FEPC	NMI 状态保存寄存器	CPU	39
FEPSW	NMI 状态保存寄存器	CPU	39
IIC0	IIC 移位寄存器 0	I ² C	462
IICC0	IIC 控制寄存器 0	I ² C	449
IICCL0	IIC 时钟选择寄存器 0	I ² C	459
IICF0	IIC 标志寄存器 0	I ² C	457
IICIC0	中断控制寄存器	INTC	535
IICSO	IIC 状态寄存器 0	I ² C	454
IICX0	IIC 功能扩展寄存器 0	I ² C	460
IMR0	中断屏蔽寄存器 0	INTC	537
IMR0H	中断屏蔽寄存器 0H	INTC	537
IMR0L	中断屏蔽寄存器 0L	INTC	537
IMR1	中断屏蔽寄存器 1	INTC	537
IMR1H	中断屏蔽寄存器 1H	INTC	537
IMR1L	中断屏蔽寄存器 1L	INTC	537
IMR3	中断屏蔽寄存器 3	INTC	537
IMR3L	中断屏蔽寄存器 3L	INTC	537
INTF0	外部中断下降沿指定寄存器 0	INTC	544
INTF3	外部中断下降沿指定寄存器 3	INTC	545
INTF9H	外部中断下降沿指定寄存器 9H	INTC	546
INTR0	外部中断上升沿指定寄存器 0	INTC	544
INTR3	外部中断上升沿指定寄存器 3	INTC	545
INTR9H	外部中断上升沿指定寄存器 9H	INTC	546
ISPR	当前服务的优先级寄存器	INTC	538
KRIC	中断控制寄存器	INTC	535
KRM	按键返回模式寄存器	KR	559
NFC	数字噪声消除控制寄存器	INTC	542
OSTS	振荡稳定时间选择寄存器	待机	565
P0	端口 0 寄存器	端口	72
P0NFC	TIP00 噪声消除控制寄存器	TMP	215
P1NFC	TIP01 噪声消除控制寄存器	TMP	215
P3	端口 3 寄存器	端口	75
P3H	端口 3 寄存器 H	端口	75
P3L	端口 3 寄存器 L	端口	75
P4	端口 4 寄存器	端口	80
P5	端口 5 寄存器	端口	82
P7	端口 7 寄存器	端口	85
P9	端口 9 寄存器	端口	87
P9H	端口 9 寄存器 H	端口	87
P9L	端口 9 寄存器 L	端口	87
PC	程序计数器	CPU	36
PCC	处理器时钟控制寄存器	CG	123
PCM	端口 CM 寄存器	端口	92

符号	名称	单元	页码
PDL	端口 DL 寄存器	端口	95
PF3H	端口 3 功能寄存器 H	端口	77
PF4	端口 4 功能寄存器	端口	81
PF9H	端口 9 功能寄存器 H	端口	89
PFC3	端口 3 功能控制寄存器	端口	77
PFC5	端口 5 功能控制寄存器	端口	84
PFC9	端口 9 功能控制寄存器	端口	90
PFC9H	端口 9 功能控制寄存器 H	端口	90
PFC9L	端口 9 功能控制寄存器 L	端口	90
PFCE3	端口 3 功能控制扩展寄存器	端口	77
PFM	掉电比较模式寄存器	ADC	369
PFT	掉电比较阈值寄存器	ADC	369
PIC0	中断控制寄存器	INTC	535
PIC1	中断控制寄存器	INTC	535
PIC2	中断控制寄存器	INTC	535
PIC3	中断控制寄存器	INTC	535
PIC4	中断控制寄存器	INTC	535
PIC5	中断控制寄存器	INTC	535
PIC6	中断控制寄存器	INTC	535
PIC7	中断控制寄存器	INTC	535
PLLCTL	PLL 控制寄存器	CG	128, 358
PM0	端口 0 模式寄存器	端口	72
PM3	端口 3 模式寄存器	端口	75
PM3H	端口 3 模式寄存器 H	端口	75
PM3L	端口 3 模式寄存器 L	端口	75
PM4	端口 4 模式寄存器	端口	80
PM5	端口 5 模式寄存器	端口	82
PM9	端口 9 模式寄存器	端口	87
PM9H	端口 9 模式寄存器 H	端口	87
PM9L	端口 9 模式寄存器 L	端口	87
PMC0	端口 0 模式控制寄存器	端口	73
PMC3	端口 3 模式控制寄存器	端口	76
PMC3H	端口 3 模式控制寄存器 H	端口	76
PMC3L	端口 3 模式控制寄存器 L	端口	76
PMC4	端口 4 模式控制寄存器	端口	81
PMC5	端口 5 模式控制寄存器	端口	83
PMC9	端口 9 模式控制寄存器	端口	88
PMC9H	端口 9 模式控制寄存器 H	端口	88
PMC9L	端口 9 模式控制寄存器 L	端口	88
PMCCM	端口 CM 模式控制寄存器	端口	92
PMCM	端口 CM 模式寄存器	端口	92
PMDL	端口 DL 模式寄存器	端口	95
PRCMD	命令寄存器	CPU	59

符号	名称	单元	页码
PRM01	预分频模式寄存器 01	TM0	229
PRSCM	间隔定时器 BRG 比较寄存器	CG	333
PRSM	间隔定时器 BRG 模式寄存器	CG	332
PSC	功率节省控制寄存器	待机	563
PSMR	功率节省模式寄存器	待机	564
PSW	程序状态字	CPU	40
PU0	上拉电阻选项寄存器 0	端口	73
PU3	上拉电阻选项寄存器 3	端口	79
PU4	上拉电阻选项寄存器 4	端口	81
PU5	上拉电阻选项寄存器 5	端口	84
PU9	上拉电阻选项寄存器 9	端口	91
PU9H	上拉电阻选项寄存器 9H	端口	91
PU9L	上拉电阻选项寄存器 9L	端口	91
PUCM	上拉电阻选项寄存器 CM	端口	93
PUDL	上拉电阻选项寄存器 DL	端口	95
r0 到 r31	通用寄存器	CPU	36
RTBH0	实时输出缓冲寄存器 H0	RTP	352
RTBL0	实时输出缓冲寄存器 L0	RTP	352
RTPC0	实时输出端口控制寄存器 0	RTP	354
RTPM0	实时输出端口模式寄存器 0	RTP	353
RXB0	接收缓冲寄存器 0	UART	394
RXB1	接收缓冲寄存器 1	UART	394
SELCNT1	选择器操作控制寄存器 1	TM0	230
SIO00	串行 I/O 移位寄存器 0	CSIO	428
SIO00L	串行 I/O 移位寄存器 0L	CSIO	428
SIO01	串行 I/O 移位寄存器 1	CSIO	428
SIO01L	串行 I/O 移位寄存器 1L	CSIO	428
SIRB0	时钟控制的串行接口接收缓冲寄存器 0	CSIO	424
SIRB0L	时钟控制的串行接口接收缓冲寄存器 0L	CSIO	424
SIRB1	时钟控制的串行接口接收缓冲寄存器 1	CSIO	424
SIRB1L	时钟控制的串行接口接收缓冲寄存器 1L	CSIO	424
SIRBE0	时钟控制的串行接口只读接收缓冲寄存器 0	CSIO	425
SIRBE0L	时钟控制的串行接口只读接收缓冲寄存器 0L	CSIO	425
SIRBE1	时钟控制的串行接口只读接收缓冲寄存器 1	CSIO	425
SIRBE1L	时钟控制的串行接口只读接收缓冲寄存器 1L	CSIO	425
SOTB0	时钟控制的串行接口发送缓冲寄存器 0	CSIO	426
SOTB0L	时钟控制的串行接口发送缓冲寄存器 0L	CSIO	426
SOTB1	时钟控制的串行接口发送缓冲寄存器 1	CSIO	426
SOTB1L	时钟控制的串行接口发送缓冲寄存器 1L	CSIO	426
SOTBF0	时钟控制的串行接口初始发送缓冲寄存器 0	CSIO	427
SOTBF0L	时钟控制的串行接口初始发送缓冲寄存器 0L	CSIO	427
SOTBF1	时钟控制的串行接口初始发送缓冲寄存器 1	CSIO	427
SOTBF1L	时钟控制的串行接口初始发送缓冲寄存器 1L	CSIO	427

符号	名称	单元	页码
SREIC0	中断控制寄存器	INTC	535
SREIC1	中断控制寄存器	INTC	535
SRIC0	中断控制寄存器	INTC	535
SRIC1	中断控制寄存器	INTC	535
STIC0	中断控制寄存器	INTC	535
STIC1	中断控制寄存器	INTC	535
SVA0	地址寄存器 0	I ² C	462
SYS	系统状态寄存器	CPU	59
TCL50	定时器时钟选择寄存器 50	TM5	291
TCL51	定时器时钟选择寄存器 51	TM5	291
TM01	16 位定时器计数器 01	TM0	220
TM0IC10	中断控制寄存器	INTC	535
TM0IC11	中断控制寄存器	INTC	535
TM5	16 位定时器计数器 5	TM5	289
TM50	8 位定时器计数器 50	TM5	289
TM51	8 位定时器计数器 51	TM5	289
TM5IC0	中断控制寄存器	INTC	535
TM5IC1	中断控制寄存器	INTC	535
TMC01	16 位定时器模式控制寄存器 01	TM0	224
TMC50	8 位定时器模式控制寄存器 50	TM5	292
TMC51	8 位定时器模式控制寄存器 51	TM5	292
TMCYC0	8 位定时器 H 载波控制寄存器 0	TMH	313
TMCYC1	8 位定时器 H 载波控制寄存器 1	TMH	313
TMHIC0	中断控制寄存器	INTC	535
TMHIC1	中断控制寄存器	INTC	535
TMHMD0	8 位定时器 H 模式寄存器 0	TMH	311
TMHMD1	8 位定时器 H 模式寄存器 1	TMH	312
TOC01	16 位定时器输出控制寄存器 01	TM0	227
TP0CCIC0	中断控制寄存器	INTC	535
TP0CCIC1	中断控制寄存器	INTC	535
TP0CCR0	TMP0 捕捉/比较寄存器 0	TMP	139
TP0CCR1	TMP0 捕捉/比较寄存器 1	TMP	141
TP0CNT	TMP0 计数器读取缓冲寄存器	TMP	143
TP0CTL0	TMP0 控制寄存器 0	TMP	133
TP0CTL1	TMP0 控制寄存器 1	TMP	134
TP0IOC0	TMP0 I/O 控制寄存器 0	TMP	135
TP0IOC1	TMP0 I/O 控制寄存器 1	TMP	136
TP0IOC2	TMP0 I/O 控制寄存器 2	TMP	137
TP0OPT0	TMP0 选择寄存器 0	TMP	138
TP0OVIC	中断控制寄存器	INTC	535
TXB0	缓冲寄存器 0	UART	395
TXB1	缓冲寄存器 1	UART	395
VSWC	系统等待控制寄存器	CPU	61

符号	名称	单元	页码
WDCS	看门狗定时钟选择寄存器	WDT	343
WDT1IC	中断控制寄存器	INTC	535
WDTE	看门狗定时器使能寄存器	WDT	349
WDTM1	看门狗定时器模式寄存器 1	WDT	344, 540
WDTM2	看门狗定时器模式寄存器 2	WDT	348
WTIC	中断控制寄存器	INTC	535
WTIIC	中断控制寄存器	INTC	535
WTM	钟表定时器操作模式寄存器	WT	336

附录 D 注意事项列表

该附录列举了文件中所述的注意事项。

如下表分类（硬件/软件）。

硬件： 微控制器内/外硬件的注意事项

软件： 软件的注意事项，例如寄存器设置或程序

(1/27)

章节	分类	功能	功能细节	注意事项	页码
第一章	硬件	引脚功能	FLMD0	正常模式下，将该引脚与V _{SS} 相连接。	p. 20 <input type="checkbox"/>
			NC	断开NC引脚。	p. 20 <input type="checkbox"/>
			EV _{DD}	使EV _{DD} 和V _{DD} 电势相同。	p. 20 <input type="checkbox"/>
第二章	硬件	引脚功能	XT1	当引脚不用时，务必将PSMR.XTSTP位设置为1。	p. 31 <input type="checkbox"/>
第三章	软件	CPU 功能	EIPC, EIPSW, FEPC, FEPSW	由于系统寄存器只有一组，因此，要实现多重中断需用程序对这组寄存器进行保存。	p. 37 <input type="checkbox"/>
			EIPC, FEPC, CTPC	即使EIPC, FEPC或CTPC的第0位被LDSR指令置1，在中断处理程序执行后程序由RETI指令返回时，第0位仍然被忽略(这是因为PC的第0位固定为0)。因此请使用偶数数值(第0位为0)来设置EIPC, FEPC或CTPC。	p. 37 <input type="checkbox"/>
			程序空间	由于从地址03FFF000H~03FFFFFFH的4KB范围是片上外围I/O设备区，这个区域不能进行取指令操作。因此，要避免执行可能使分支地址计算结果为此4KB中值的操作。	p. 45 <input type="checkbox"/>
			片上外围 I/O 区域	当寄存器被一个字操作指令读写时，一个字的空间被分为低16位和高16位分两次被字节操作先后读写，其中最低2位地址被忽略。	p. 49 <input type="checkbox"/>
				若对能进行字节操作的寄存器进行半字操作，那么读取时，高8位数据无效，写入时，低8位数据被写入寄存器。	p. 49 <input type="checkbox"/>
				未被指定为寄存器的地址为未来扩展保留。对这些地址操作结果的有效性是无法保证的。	p. 49 <input type="checkbox"/>
			向特殊寄存器中写入数据	设置IDLE模式 或 STOP 模式(通过将PSC.STP位置1)后，要立即插入5条NOP指令。	p. 58 <input type="checkbox"/>
对于PRCMD寄存器的存储指令，系统将不响应中断。存储指令应该是按上述的第<2>和<3>步骤连续执行的。若在步骤<2>和<3>之间执行了其他指令，而这个指令又进行了中断的响应，则将打乱上述的顺序导致设备误操作。	p. 58 <input type="checkbox"/>				
			尽管对PRCMD写入的数据为虚数据，但还是请使用与设置特殊寄存器(例子中的<3>)的通用寄存器相同的寄存器来向PRCMD寄存器(例子中的<2>)写入虚数据。这个过程也同样可以用于使用通用寄存器进行寻址的操作中。	p. 58 <input type="checkbox"/>	

章节	分类	功能	功能细节	注意事项	页码
第三章	软件	CPU 功能	系统状态寄存器	如果在对PRCMD寄存器进行写入之后, 向SYS寄存器(并非特殊寄存器)的PRERR位写入0, 那么PRERR位将被清零(写入命令优先)。	p. 60 <input type="checkbox"/>
				如果在对PRCMD寄存器进行写入之后, 又执行了对PRCMD寄存器(并非特殊寄存器)的写入操作, 那么PRERR位将被置1。	p. 60 <input type="checkbox"/>
			寄存器访问的等待	在使用V850ES/KE2器件时, 需要首先设置的寄存器。 • 系统等待控制寄存器 (VSWC) • 看门狗定时器模式寄存器 2 (WDTM2)	p. 61 <input type="checkbox"/>
			系统等待控制寄存器	访问一个片上外围I/O寄存器需要3个时钟周期的时间(不包含等待周期)。V850ES/KE2 需要根据不同的操作频率插入不同的等待周期数。因此, 请根据所使用的内部系统时钟, 将以下对应的数值设置到VSWC寄存器中。	p. 61 <input type="checkbox"/>
			访问特定的片上外围I/O寄存器	如果取自内部 ROM或内部 RAM, 等待数如下所示。 如果取自外部存储器, 等待数则可能减少。 外部存储器访问周期的影响根据等待设置等的变化有所不同。但是上述等待数是最大值, 因此不会产生更大的值。 当CPU工作在副时钟且X1引脚上没有时钟输入时, 不要访问发生等待的寄存器。如果等待发生, 则只能由复位释放。	p. 63 <input type="checkbox"/> p. 63 <input type="checkbox"/>
第四章	硬件	端口功能	端口n功能寄存器	无论PMcN寄存器的设置如何, 只有在PMn.PMnM位为0(输出模式)时, PFnm位才有效。当PFnm位为1时, (输入模式), 在PFn寄存器的设置值是无效的。	p. 69 <input type="checkbox"/>
			端口 0	P02 ~ P06 引脚在复用功能的输入模式具有滞后性, 但在端口模式没有滞后性。	p. 72 <input type="checkbox"/>
			端口 3	P31 ~ P35, P38和 P39在复用功能期间有滞后性, 但在端口模式下不会。	p. 74 <input type="checkbox"/>
	软件	P3 寄存器	要以8位或1位为单位读写P3寄存器的位8~15, 需要将他们指定为P3H寄存器的位0~7。	p. 75 <input type="checkbox"/>	
		PM3 寄存器	要以8位或1位为单位读写PM3寄存器的位8~15, 需要将他们指定为PM3H寄存器的位0~7。	p. 75 <input type="checkbox"/>	
		PMC3 寄存器	要以8位或1位为单位读写PMC3寄存器的位8~15, 需要将他们指定为PMC3H寄存器的位0~7。	p. 76 <input type="checkbox"/>	
			INTP7引脚和RXD0引脚为复用功能引脚。当使用引脚的RXD0功能, 需要禁止复用功能INTP7引脚的边沿检测(将INTF3.INTF31和INTR3.INTR31清0)。当使用引脚的INTP7功能, 停止UART0接收操作(将ASIM0.RXE0位清零)。	p. 76 <input type="checkbox"/>	
	PF3H 寄存器	当P38和P39作为N沟道开漏输出复用功能引脚使用时, 请按下列顺序设置。一定要先将接口锁存器设置为1, 再将引脚设置N沟道开漏输出。 P3n 位 = 1 → PF3n 位 = 1 → PMC3n 位 = 1	p. 77 <input type="checkbox"/>		
硬件	指定端口3的复用功能引脚	ASCK0和 ADTRG引脚是复用功能引脚。当使用引脚的ASCK0功能时, 禁止复用功能ADTRG引脚的触发输入(将ADS.TRG位清零或将 ADS.ADTMD 位设置为1)。当使用引脚的ADTRG功能时, 不要将UART0操作时钟设置为外部输入(设置CKSR0.TPS03到CKSR0.TPS00位为1011以外的数)。	p. 78 <input type="checkbox"/>		

章节	分类	功能	功能细节	注意事项	页码	
第四章	硬件	端口功能	指定端口3的复用功能引脚	在PFC3n 和 PFCE3n位保持初始值0时, 通过PMC3.PMC3n位指定P3n引脚的复用功能时, 输出变得不确定。因此, 要设置P3n引脚的复用功能, 首先将PFC3n和PFCE3n位设置为1, 再将PMC3n位设置为1 (n = 3, 4)。	p. 78 <input type="checkbox"/>	
			端口 4	P40 和 P42 引脚在复用功能的输入模式具有滞后性, 但在端口模式没有滞后性。	p. 80 <input type="checkbox"/>	
			PF4 寄存器	当P41和P42作为N沟道开漏输出复用功能引脚, 按如下顺序设置。一定要先将端口锁存器设置为1, 然后再将引脚设置为N沟道开漏输出。 P4n 位 = 1 → PF4n 位 = 1 → PMC4n 位 = 1	p. 81 <input type="checkbox"/>	
	软件	硬件	端口 9	PFC5 寄存器	在PFC5n位保持初始值0时, 通过PMC5.PMC5n位指定P5n引脚的复用功能时, 输出变得不确定。因此, 要设置 P5n引脚的复用功能2, 首先将PFC5n 位设置为1, 再将PMC5n引脚设置为1 (n = 3 ~5)。	p. 84 <input type="checkbox"/>
				P9 寄存器	P97, P99和P913~P915在复用功能下具有滞后性, 在端口模式下则无。	p. 86 <input type="checkbox"/>
		软件	P9 寄存器	要以8位或1位读写P9寄存器的位8~15, 需要将它们指定为P9H寄存器的位0~7。	p. 87 <input type="checkbox"/>	
			PM9 寄存器	要以8位或1位读写PM9寄存器的位8~15, 需要将它们指定为PM9H寄存器的位0~7。	p. 87 <input type="checkbox"/>	
			PMC9 寄存器	要以8位或1位读写PMC9寄存器的位8~15, 需要将它们指定为PMC9H寄存器的位0~7。	p. 88 <input type="checkbox"/>	
			PF9H 寄存器	当将P98和P99作为N沟道开漏复用功能引脚时, 按以下顺序设置N沟道开漏。一定要先设置端口锁存器为1, 后设置N沟道开漏输出。 P9n 位 = 11 → PFC9n 位 = 0/11 → PF9n 位 = 11 → PMC9n 位 = 1	p. 89 <input type="checkbox"/>	
			PFC9 寄存器	在PFC9n位保持初始值0时, 通过PMC9.PMC9n位指定端口9的复用功能时, 输出变得不确定。因此, 要设置端口 9的复用功能2时, 首先设置PFC9n位为1, 再设置PMC9n位为1 (n = 0, 1, 6~9, 13~15)。	p. 90 <input type="checkbox"/>	
				要以8位或1位读写PFC9寄存器的位8~15, 需要将它们指定为PFC9H寄存器的位0~7。	p. 90 <input type="checkbox"/>	
			PU9 寄存器	以8位或1位读写PU9寄存器的位8~15, 需要将它们指定为PU9H寄存器的位0~7。	p. 91 <input type="checkbox"/>	
			RXD0, INTP7	INTP7 和 RXD0引脚为复用功能引脚。当把引脚作为RXD0引脚时, 禁止复用功能引脚INTP7的边沿检测 (将 INTF3.INTF31 位以及INTR3 .INTR31 位清零)。当将引脚作为INTP7引脚使用时, 停止UART0接收操作 (将ASIM0.RXE0位清零)。	p. 115 <input type="checkbox"/>	
			ASCK0, ADTRG	ASCK0和 ADTRG引脚为复用功能引脚。当把引脚作为ASCK0引脚时, 禁用复用功能引脚ADTRG的触发输入 (将 ADS.TRG位清零或将ADS.ADTMD 位置1)。当将引脚作为ADTRG引脚使用时, 勿将UART0操作时钟设置为外部输入 (将CKSR0.TPSR03到CKSR0.TPS00位设置为除了1011以外的数)。	p. 115 <input type="checkbox"/>	
			端口n寄存器 (Pn) 位操作指令的注意事项	当位操作指令执行在一个同时提供输入和输出功能的端口时, 不属于操作对象目标的输入端口的输出锁存的值也可能被写入。因此, 当端口从输入模式切换到输出模式时, 推荐重写输出锁存。	p. 118 <input type="checkbox"/>	
			滞后性	在端口模式下, 以下端口没有滞后性。 P02 ~ P06 P31 ~ P35, P38, P39 P40, P42 P97, P99, P913 ~ P915	p. 119 <input type="checkbox"/>	

章节	分类	功能	功能细节	注意事项	页码
第五章	软件	时钟生成功能	PCC 寄存器	在通过CLKOUT 向外输出时钟信号时，不要改变CPU时钟的频率（通过CK3~CK0位）。	p. 124 <input type="checkbox"/>
				对CK3位操作时，请使用位操作指令。如果使用8位操作指令，不要改变CK2~CK0位的值。	p. 124 <input type="checkbox"/>
				当CPU在副时钟下工作且X1引脚没有时钟输入时，不要访问寄存器，否则将发生等待。（访问方法的更多细节，敬请参阅3.4.8（2）特殊片上外围I/O寄存器的访问）如果等待发生，则只能由复位释放。	p. 124 <input type="checkbox"/>
				当停止主时钟工作时，也要同时停止PLL。	p. 125 <input type="checkbox"/>
				如果不满足下面的条件，那么请改变CK2~CK0位的设置以满足该条件，然后再切换到副时钟工作模式。 内部系统时钟 (fCLK) > 副时钟 (fXT: 32.768 kHz) × 4	p. 125 <input type="checkbox"/>
			PLLCTL 寄存器	请务必将3~7位清零。改变第3位的值并不影响操作。	p. 128 <input type="checkbox"/>
第六章	软件	16位定时器/事件计数器P (TMP)	TP0CTL0 寄存器	在TP0CE位 = 0时才可对TP0CKS2 ~ TP0CKS0 位进行设置。当将TP0CE 位的值从0改变为1时，可同时对 TP0CKS2 ~ TP0CKS0位进行设置。	p. 133 <input type="checkbox"/>
				请务必将第3~6位清零。	p. 133 <input type="checkbox"/>
			TP0CTL1 寄存器	对TP0EST位的设置只有在外部触发脉冲输出模式或单脉冲输出模式才有效。在其它模式下，对该位的置1操作将被忽略。	p. 134 <input type="checkbox"/>
				在外部事件计数模式下，无论TP0EEE位如何设置，外部事件计数输入都是被选中的。	p. 134 <input type="checkbox"/>
				在TP0CTL0.TP0CE位 = 0时才可对TP0EEE和TP0MD2 ~ TP0MD0 位进行设置 (TP0CE 位 = 1时只可以向这些位写入相同的值)。如果在TP0CE位 = 1时对这些位进行了改写，那么操作结果将不能被保证。若由于误操作引起了改写，则要将TP0CE位清零然后再次设置这些寄存器位。	p. 134 <input type="checkbox"/>
				请务必将第3, 4和7位清零。	p. 134 <input type="checkbox"/>
			TP0IOC0 寄存器	在TP0CTL0.TP0CE 位 = 0时才可对 TP0OL1, TP0OE1, TP0OL0 和 TP0OE0 位进行改写 (TP0CE 位 = 1时只可以向这些位写入相同的值)。若在TP0CE位 = 1由于误操作引起了改写，则要将TP0CE位清零然后再次设置这些寄存器位。	p. 135 <input type="checkbox"/>
				在TP0CE位和TP0OEm位为0时，即使对TP0OLm位进行了操作，TOP0m引脚的输出电平也不能确定。(a = 0, 1)。	p. 135 <input type="checkbox"/>
			TP0IOC1 寄存器	在TP0CTL0.TP0CE 位 = 0时才可对 TP0IS3 ~ TP0IS0位进行改写 (TP0CE 位 = 1时只可以向这些位写入相同的值)。若在TP0CE位 = 1由于误操作引起了改写，则要将TP0CE位清零然后再次设置这些寄存器位。	p. 136 <input type="checkbox"/>
				TP0IS3 ~ TP0IS0 位只有在自由运行定时器模式和脉宽测量模式下才有效。其它模式下，无法进行捕捉操作。	p. 136 <input type="checkbox"/>
			TP0IOC2 寄存器	在TP0CTL0.TP0CE 位 = 0时才可对 TP0EES1, TP0EES0, TP0ETS1和 TP0ETS0位进行改写 (TP0CE 位 = 1时只可以向这些位写入相同的值)。若在TP0CE位 = 1由于误操作引起了改写，则要将TP0CE位清零然后再次设置这些寄存器位。	p. 137 <input type="checkbox"/>
				TP0EES1 和 TP0EES0 位只有在TP0CTL1.TP0EEE 位 = 1 或设置了外部事件计数模式(TP0CTL1.TP0MD2 ~ TP0CTL1.TP0MD0 位 = 001)时才有效。	p. 137 <input type="checkbox"/>
TP0ETS1 和 TP0ETS0 位只有在外部触发脉冲输出模式(TP0CTL1.TP0MD2 ~ TP0CTL1.TP0MD0 位 = 010) 或单脉冲输出模式(TP0CTL1.TP0MD2 ~ TP0CTL1.TP0MD0 = 011) 下才有效。	p. 137 <input type="checkbox"/>				

第 六 章	软件	功能	功能细节	注意事项	页码
第六章	软件	16位定时器/事件计数器P (TMP)	TP0OPT0 寄存器	在TP0CTL0.TP0CE 位 = 0时才可对 TP0CCS1和TP0CCS0位进行改写 (TP0CE 位 = 1时只可以向这些位写入相同的值)。若在TP0CE位 = 1由于误操作引起了改写, 则要将TP0CE位清零然后再次设置这些寄存器位。 请务必将第1~3位, 第6和第7位清零。	p. 138 <input type="checkbox"/>
			TP0CCR0 寄存器	当CPU工作于副时钟且主时钟振荡停止时禁止访问TP0CCR0寄存器。的更多细节, 敬请参阅 3.4.8 (2)。	p. 139 <input type="checkbox"/>
			TP0CCR1 寄存器	当CPU工作于副时钟且主时钟振荡停止时禁止访问TP0CCR1寄存器。的更多细节, 敬请参阅 3.4.8 (2)。	p. 141 <input type="checkbox"/>
			TP0CNT 寄存器	以下情况下禁止访问TP0CNT寄存器。的更多细节, 敬请参阅 3.4.8 (2) 访问特定的片上外围I/O寄存器。	p. 143 <input type="checkbox"/>
			操作	使用外部事件计数模式时, 请关闭对 TIP00引脚捕捉触发输入的有效边沿的检测(通过将TPIOIC1.TP0IS1 和 TP0IOC1.TP0IS0 位清零)。	p. 144 <input type="checkbox"/>
				使用外部触发脉冲输出模式, 单脉冲输出模式和脉宽测量模式时, 请选择内部时钟作为计数时钟(通过将TP0CTL1.TP0EEE位清零)。	p. 144 <input type="checkbox"/>
			TP0CTL1.TP0EEE 位	该寄存器位只有在中断请求信号(INTTP0CC0 和 INTTP0CC1)被中断屏蔽标志 (TP0CCMK0 和 TP0CCMK1)屏蔽且同时允许了定时器输出 (TOP01)时, 才可以被置1。此时, 请将相同的值设置到TP0CCR0 和 TP0CCR1 寄存器中。(, 敬请参阅 6.5.1 (2) (d) TP0CCR1 寄存器的操作)。	p. 146 <input type="checkbox"/>
			外部事件计数模式的操作时序	在外部事件计数模式中, 不要将TP0CCR0 和TP0CCR1 寄存器设置为0000H。	p. 159 <input type="checkbox"/>
				在外部事件计数模式中, 禁止使用定时器输出功能。如果要将外部事件计数输入作为定时器输出使用, 那么请将工作模式设置为间隔定时器模式并将外部事件计数输入设定为计数时钟(TP0CTL1.TP0MD2 ~ TP0CTL1.TP0MD0 位 = 000, TP0CTL1.TP0EEE 位 = 1)。	p. 159 <input type="checkbox"/>
			单脉冲输出模式的寄存器设置	如果TP0CCR1寄存器设定的数值大于TP0CCR0设定的数值, 则单脉冲输出模式下必不输出单脉冲。	p. 178 <input type="checkbox"/>
PaNFC 寄存器	经过采样时钟周期×采样次数的时间后, 再将TMP0的16位计数器使能 (TP0CTL.TP0CE 位 = 1)。 务必将位7, 5~ 3全部清零。	p. 215 <input type="checkbox"/>			
捕捉操作	当使用捕捉操作, 且选择fXX/8, fXX/16, fXX/32, fXX/64, fXX/128, 或外部事件计数器(TP0CLT1.TP0EEE位 = 1)为计数时钟时, 如果TP0CE 位置1后马上输入捕捉触发源, TP0CCRn 寄存器可能会捕捉到FFFFH而不是0000H。	p. 217 <input type="checkbox"/>			
第七章	软件	16位定时器/事件计数器0	CR010 和 CR011 寄存器	当P35引脚用作TI010的有效边沿且使用定时器输出功能, 将P32引脚设置为定时器输出引脚 (TO01)。 如清零TMC013 和 TMC012位与捕捉触发器输入冲突, 则捕捉数据不确定。	p. 221 <input type="checkbox"/>
				将捕捉模式转至对比模式时, 先将 TMC013 和 TMC012 位清至 00, 然后在修改设置。 除非设备复位, 捕捉的数值将保存在 CR010 和 CR011 计数器内。如果转换到对比模式, 要先设置对比值。	p. 221 <input type="checkbox"/>
					p. 221 <input type="checkbox"/>

第 七 章	软件	功能	功能细节	注意事项	页码
第 七 章	软件	16位定时器/事件计数器 0	CR010 和 CR011寄存器的捕捉操作	如果用TI010引脚输入信号的反相信号来捕捉TM01寄存器的计数值并存入CR010寄存器，则捕捉完数值后，不产生中断请求信号INTTM010。如果TI010引脚输入信号的反相信号作为触发源时检测到TI011引脚的有效边沿，不执行捕捉，但是产生INTTM010信号，作为外部中断信号。如果不用外部中断，需要屏蔽INTTM010信号。	p. 223 <input type="checkbox"/>
			TMC01 寄存器	16位定时器/事件计数器01在TMC012 和TMC013不再为00（停止工作模式）的时刻开始工作。将TMC012 和TMC013置为00将使工作停止。 当主时钟停止副时钟工作时，不要访问TMC01寄存器。 详细情况，敬请参阅3.4.8（2）。 当交替使用TO01引脚和TI010引脚时一定要将TMC011位清零。	p. 224 <input type="checkbox"/> p. 224 <input type="checkbox"/> p. 225 <input type="checkbox"/>
			CRC01 寄存器	为确保捕捉操作正确进行，捕捉触发器需要一个比PRM01或SELCNT1寄存器选择的计数时钟长两个周期的脉冲。	p. 226 <input type="checkbox"/>
			TOC01 寄存器	以下步骤中一定要设置 TOC01 寄存器。 <1> TOC014 和 TOC011 位设置为 1。 <2> 仅将 TOE01 位设置为 1。 <3> 将LVS01位或LVR01位设置为1。	p. 227 <input type="checkbox"/>
			PRM01 寄存器	当设置PRM011和 PRM010位为11（指定TI010引脚的有效边沿为计数时钟）时，不要应用如下设置。 • 从TI010引脚有效边沿输入清零和开始模式 • 设置TI010引脚为捕捉触发引脚 如果在TI010 或TI011引脚为高电平，且TI010 或TI011引脚的上升沿或双边沿为有效边沿时使能16位定时器/事件计数器01，则TI010 或TI011引脚的高电平会被视为一个上升沿。当TI010 或TI011引脚上拉时注意这点。但在定时器工作停止后再重新使能的情况下，不会检测到上升沿。 当 P35引脚作为 TI010的有效边沿且使用定时器输出功能时，设置P32引脚为定时器输出引脚（TO01）。	p. 229 <input type="checkbox"/> p. 229 <input type="checkbox"/> p. 229 <input type="checkbox"/>
			计数时钟设置	选择内部时钟时，请按此设置以满足如下要求： VDD = 4.0~5.5 V: 计数时钟 ≤ 10 MHz VDD = 2.7 ~4.0 V: 计数时钟 ≤ 5 MHz 外部时钟脉冲要大于内部时钟（fXX/4）周期的2倍。	p. 230 <input type="checkbox"/> p. 230 <input type="checkbox"/>
			通过TI010引脚有效边沿的输入进入清零和开始模式的操作	不要设置计数时钟为TI010引脚的有效边沿（RPM01.PR011和 RPM01.PR010位 = 11）。PRM011和 PRM010位为11时，TM01寄存器清零。	p. 240 <input type="checkbox"/>
			PPG 输出操作	要在操作期间改变占空比因子（CR011寄存器值），参见7.5.1 TM01操作期间改写CR011寄存器。	p. 265 <input type="checkbox"/>
			PPG 输出操作的寄存器设置	设置CR010 和 CR011寄存器的值，使满足0000H ≤ CR011 < CR010 ≤ FFFFH。	p. 266 <input type="checkbox"/>
			单脉冲输出操作	在单脉冲正在输出时，不要再一次输入触发（设置 OSPT01为 1 或检测TI010 引脚的有效边沿）。要再次输出单脉冲，需要在当前单脉冲输出完成后再产生新的触发。	p. 268 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码
第七章	软件	16 位定时器/事件计数器 0	LVS01 和 LVR01 位	确保按照上面的步骤 <1>、<2> 和 <3> 设置 LVS01 和 LVR01 位。步骤 <2> 可以在 <1> 后和 <3> 前执行。	p. 282 <input type="checkbox"/>
			TI010/TO01 引脚的复用功能	在检测 TI010 引脚作为触发源的有效边沿时执行单脉冲输出，需用 TO01 输出，复用为 P32。 当使用 TO01 引脚输出且复用为 P35 时，如果 TI010 引脚同样复用为 P35 则不可使用。 当仅使用软件触发源（置 (1) TOC01.OSPT01 位）作为单脉冲输出的启动触发源时，P32 和 P35 任意一个可以作为 TO01 引脚的输出。	p. 283 <input type="checkbox"/>
				通过检测 TI010 引脚输入的有效边沿执行 TO01 引脚输出反转操作时，用 TO01 输出，复用为 P32。 当使用 TO01 引脚输出且复用为 P35 时，如果 TI010 引脚同样复用为 P35 则不可使用。使用 TO01 引脚且复用为 P35 时，将 TMC01.TMC011 位清零。	p. 283 <input type="checkbox"/>
	硬件		启动定时器的误差	定时器启动后，匹配信号产生前，最高可产生 1 个时钟周期的误差。这是因为 TM01 寄存器的计数操作的启动是与计数脉冲同步的。	p. 283 <input type="checkbox"/>
	软件		CR0n0 和 CR0n1 寄存器的设置	设置 CR010 和 CR011 寄存器（TM01 寄存器和 CR010 寄存器匹配时清零并开始的模式下） 设置 CR010 和 CR011 寄存器为 0000H 以外的值（当这些寄存器作为外部事件计数器时，单脉冲计数操作是不能实现的）。	p. 283 <input type="checkbox"/>
	硬件		捕捉寄存器的数据保持时间	如果在读 CR010/CR011 寄存器时，TI011/TI010 引脚输入有效边沿，则 CR010/CR011 执行捕捉操作，但此时所读值正确性不受保证。但是由于检测到有效边沿，产生中断请求信（INTTM010/INTTM011）。	p. 284 <input type="checkbox"/>
		软件		在 16 位定时器/事件计数器 0 已经停止后，CR010 和 CR011 寄存器的值不能得到保证。	p. 284 <input type="checkbox"/>
	硬件		设置有效边沿	在定时器操作停止时（TMC01.TMC013 和 TMC01.TMC012 位 = 00）设置 TI010 引脚的有效边沿。通过 PRM01.ES100 和 PRM01.ES101 位设置有效边沿。	p. 284 <input type="checkbox"/>
			重触发单脉冲	保证在单脉冲输出模式正在输出有效电平时不产生触发。 保证在当前有效电平输出完毕后输入下一个触发源。	p. 284 <input type="checkbox"/>
			OVF01 标志	除了 TM01 寄存器溢出的情况以外，以下情况也使 TMC01.OVF01 标志设置为 1。 选择如下模式：TM01 寄存器和 CR010 寄存器匹配时清零并开始。 → 将 CR010 寄存器设置为 FFFFH → 当 TM01 寄存器从 FFFFH（与 CR010 寄存器匹配）清为 0000H 时	p. 285 <input type="checkbox"/>
				TM01 寄存器溢出后，清零 OVF01 标志无效，即使清零是在下一个计数时钟被计数操作前进行（在 TM01 寄存器变为 0001H 之前进行）。	p. 285 <input type="checkbox"/>
	硬件		单脉冲输出	单脉冲输出通常在如下两种模式之一，自由运行定时器模式和清零并开始发生在 TI010 引脚有效边沿时的模式。在如下模式单脉冲输出无法实现，清零并开始发生在 TM01 寄存器和 CR010 寄存器匹配时的模式。	p. 285 <input type="checkbox"/>

第 七 章	功 能	功 能 细 节	注 意 事 项	页 码	
第七 章	硬 件	16位定时器/事件计数器0	捕捉操作	如果指定TI010引脚的有效边沿为计数时钟，指定TI010引脚为触发源的捕捉寄存器工作不正常。	p. 286 <input type="checkbox"/>
				为了精确的捕捉计数值，TI010和TI011引脚输入的捕捉触发源脉冲宽度必须大于两个计数时钟周期，计数时钟由PRM01和SELCNT1寄存器选择。	p. 286 <input type="checkbox"/>
				尽管捕捉操作在计数时钟的下降沿执行，中断请求信号（INTTM010，INTTM011）在下个计数时钟的上升沿产生。	p. 286 <input type="checkbox"/>
	软 件	边沿检测	如果复位后16位定时器/事件计数器0的操作被禁止且TI010或TI011引脚处于高电平状态且上升沿或者双边沿指定为TI010或TI011引脚的有效边沿，则TI010或TI011引脚的高电平被检测为上升沿。注意到这点当TI010或TI011引脚上拉时。但是当操作先停止再重新使能时，上升沿不被检测。	当TM01寄存器的计数值在TI010引脚输入信号的反相相位被捕捉到CR010寄存器，计数值被捕捉后不产生中断信号（INTTM010）。如果在此操作中TI011引脚上检测到有效边沿，捕捉操作不执行，但产生INTTM010信号，作为外部中断信号。	p. 286 <input type="checkbox"/>
				根据TI010引脚的有效边沿用于计数时钟或者捕捉触发源，噪声消除的采样时钟有所不同。在前者的情况，采样以 $f_{xx}/4$ 执行，在后者的情况，采样计数时钟是由PRM01和SELCNT1寄存器选择的。	p. 286 <input type="checkbox"/>
				当输入到TI010引脚的信号被采样，且连续两次检测到有效边沿，则认为检测到有效边沿。因此，脉冲宽度较短的噪声可以被消除。	
第八 章	软 件	8位定时器/事件计数器5	TM5n 寄存器	当以级联方式连接时，甚至最低定时器（TM50）被清除时，这些寄存器成为000H。	p. 289 <input type="checkbox"/>
			CR5n 寄存器	当TM5n寄存器和CR5n寄存器（TMC5n.TMC5n6位=0）匹配并且进行清除&启动时，在计数操作中不要写不同值到CR5n寄存器中。	p. 290 <input type="checkbox"/>
				在PWM模式中，设置CR5n寄存器重写间隔为三个或更多计数时钟（用TCL5n寄存器选择时钟）。	p. 290 <input type="checkbox"/>
				当使用级联连接方式时，在改变CR5n寄存器值之间，确保停止定时器。	p. 290 <input type="checkbox"/>
	硬 件	TCL5n 寄存器	当选择间隔时钟时，设置此值以满足以下条件。 VDD = 4.0 ~ 5.5 V: 技术时钟 ≤ 10 MHz VDD = 2.7 ~ 4.0 V: 技术时钟 ≤ 5 MHz	p. 291 <input type="checkbox"/>	
			在用不同数据重写TCL5n寄存器时，停止定时器操作。	p. 291 <input type="checkbox"/>	
			TMC5n 寄存器	因为TO51和TI51是相同引脚的交换功能，在一个时刻仅仅一个引脚可以使用。	p. 293 <input type="checkbox"/>
			在非PWM模式下，LVS5n和LVR5n位设置有效。	p. 293 <input type="checkbox"/>	
软 件	TMC5n 寄存器	不要同时设置以下的<1> ~ <4>项，设置如下。 <1> 设置TMC5n1, TMC5n6, 和TMC514 ^注 位: 运行模式的设置 <2> 为定时器输出使能设置TOE5n位。: 定时器输出使能 <3> 设置LVS5n和LVR5n位（小于2）: 定时器输出F/F的设置 <4> 设置TCE5n位	p. 293 <input type="checkbox"/>		

	分类	功能	功能细节	注意事项	页码
第八章	软件	8位定时器/事件计数器5	作为间隔定时器运行	在间隔定时器操作期间，不要重写CR5n寄存器的值。	p. 294 <input type="checkbox"/>
			作为外部事件计数器运行	当外部事件计数器操作期间，不要重写CR5n寄存器的值。	p. 296 <input type="checkbox"/>
			方波输出操作	在方波输出期间，不要重写CR5n寄存器的值。	p. 297 <input type="checkbox"/>
			8位PWM输出操作	CR5n寄存器重写时间间隔必须为三个或更多操作时钟（由TCL5n寄存器进行设置）。	p. 299 <input type="checkbox"/>
			基于CR5n寄存器转换的操作	在步骤<1>和<2>之间，当从CR5n寄存器重加载时，实际上使用的值不同（已读值：M；CR5n寄存器的真实值：N）。	p. 301 <input type="checkbox"/>
			作为间隔定时器运行（16位）	为了在级联连接中使用8位访问进行写入，在操作开始时设置TCE51位为1，然后设置TCE50位为1。当操作停止时，清除TCE50位为0，然后清除TCE51位为0。	p. 302 <input type="checkbox"/>
				在级联连接中，TI50输入，TO50输出，并且使用INTTM50信号。不要使用TI51输出，TO51输出和INTTM51信号，而是将他们掩膜（关于细节，参见第17章 中断/异常处理功能）。将LVS51，LVR51，TMC511，和TOE51位清零。	p. 302 <input type="checkbox"/>
				在定时器操作中，不要改变CR5寄存器的值。	p. 302 <input type="checkbox"/>
			作为外部事件计数器运行（16位）	在外部事件计数器操作期间，不要重写CR5n寄存器的值。	p. 304 <input type="checkbox"/>
				在级联连接模式中，要使用8位访问进行写入，设置TCE51位为1，然后设置TCE50位为1。当操作停止时，清除TCE50位为0，然后清除TCE51位为0（n = 0, 1）。	p. 304 <input type="checkbox"/>
				在级联连接模式中，使用TI50输出和INTTM50信号。不要使用TI51输入，TO51输出和INTTM51信号，而是将他们掩膜（关于细节，敬请参阅第17章 中断/异常处理功能）。将LVS51，LVR51，TMC511，和TOE51位清除为0。	p. 304 <input type="checkbox"/>
				在外部事件计数器操作期间，不要改变CR5寄存器的值。	p. 304 <input type="checkbox"/>
			方波输出操作（16位分辨率）	在操作期间，不要将不同的值写入到CR5寄存器中。	p. 305 <input type="checkbox"/>
启动定时器错误	在定时器启动之后且在产生匹配信号之间，1个时钟发生错误。这是因为TM5n寄存器与计数脉冲没有同步启动。	p. 306 <input type="checkbox"/>			
第九章	软件	8位定时器H	CMPn0 寄存器	在定时器计数操作被禁止期间对CMPn0寄存器进行数据重写。	p. 308 <input type="checkbox"/>
			CMPn1 寄存器	在PWM输出模式和载波发生器模式下，定时器计数操作被终止之后（TMHEn位 = 0），当启动定时器计数操作（TMHMDn.TMHEn位 = 1）时要对CMPn1寄存器进行设置。（即使对CMPn1寄存器设定的是相同的值也要再次设置）。	p. 309 <input type="checkbox"/>
			TMHMD0 寄存器	设置时满足以下条件。 VDD = 4.0 ~ 5.5 V: 技术时钟 ≤ 10 MHz VDD = 2.7 ~ 4.0 V: 技术时钟 ≤ 5 MHz	p. 311 <input type="checkbox"/>
				当TMHE0位=1时，设置位（除了TMHMD0寄存器中的设置位）被禁止。	p. 311 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码	
第九章	软件	8位定时器H	TMHMD0 寄存器	在PWM输出模式下，定时器计数操作被终止之后（TMHE0 位 = 0），当启动定时器计数操作（TMHE0 位 = 1）时要对CMP01寄存器进行设置。（即使对CMP01寄存器设定的是相同的值也要再次设置）。	p. 311 <input type="checkbox"/>	
				当使用的是载波发生器模式，则要将8位定时器H0计数时钟频率设置为8位定时器/事件计数器50计数时钟频率的6倍或更高。	p. 311 <input type="checkbox"/>	
			TMHMD1 寄存器	设置时满足以下条件。 VDD = 4.0 ~ 5.5 V: 技术时钟 ≤ 10 MHz VDD = 2.7 ~ 4.0 V: 技术时钟 ≤ 5 MHz	p. 312 <input type="checkbox"/>	
				当TMHE1位=1时，设置位（除了TMHMD1寄存器中的设置位）被禁止。	p. 312 <input type="checkbox"/>	
				在PWM输出模式和载波发生器下，定时器计数操作被终止之后（TMHE1 位 = 0），当启动定时器计数操作（TMHE1 位 = 1）时要对CMP11寄存器进行设置。（即使对CMP11寄存器设定的是相同的值也要再次设置）。	p. 312 <input type="checkbox"/>	
			PWM 输出模式操作	当定时器计数器正在运行时，可以改变CMPn1寄存器中设定的值。但是这至少需要3个运行时钟（此信号由TMHMDn寄存器的CKSHn2~CKSHn0位来选择）。直到这个值被传输给寄存器时CMPn1寄存器的值才被改变。	p. 318 <input type="checkbox"/>	
				定时器计数操作被终止（TMHEn 位 = 0）之后再启动计数器计数操作时（TMHEn 位 = 1）一定要对寄存器进行设置。（即使对寄存器设置的是相同的值也要再次设置）。	p. 318 <input type="checkbox"/>	
				要确保CMPn1寄存器中设定的值（M）和CMPn0寄存器中设定的值（N）在下面这个范围之内。 $00H \leq \text{CMPn1 (M)} < \text{CMPn0 (N)} \leq \text{FFH}$	p. 318 <input type="checkbox"/>	
			传输时间	对NRZBn位进行重写之后，至少应在第二个时钟才能对其进行再次重写，否则不能保证从NRZBn位到NRZn位的传输。	p. 324 <input type="checkbox"/>	
				当在载波发生器模式下使用8位定时器/事件计数器5n时，则在<1>时间中产生一个中断。当使用其他模式（除了载波发生器模式）时可以在不同时间内产生中断。	p. 324 <input type="checkbox"/>	
			在载波发生器模式下设置寄存器	定时器计数操作被终止之后（TMHEn 位 = 0）再次启动定时器计数操作时（TMHEn 位 = 1）一定要设置CMPn1寄存器。（即使对CMPn1寄存器设置相同的值也要再次设置）。	p. 326 <input type="checkbox"/>	
				将CMPn0 和 CMPn1 寄存器的值设置在01H ~ FFH范围之内。	p. 326 <input type="checkbox"/>	
				在载波发生器模式下，CMPn1寄存器被重写之后，将新值传输给此寄存器需要3个操作时钟（通过TMHMDn.CKSHn0 到 TMHMDn.CKSHn2位选择的信号）。	p. 326 <input type="checkbox"/>	
				启动计数操作之前要对TMCYn.RMCn位进行设置。	p. 326 <input type="checkbox"/>	
					当使用载波发生器模式时，将8位Hn计数时钟频率设置为8位定时器/事件计数器5n时钟频率的6倍或更高。	p. 326 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码	
第十章	软件	间隔定时器, 钟表定时器	PRSM 寄存器	对这些位进行设置, 使其满足以下条件。 VDD = 4.0 ~ 5.5 V: fBGCS ≤ 10 MHz VDD = 2.7 ~ 4.0 V: fBGCS ≤ 5 MHz	p. 332 <input type="checkbox"/>	
				当间隔定时器BRG正在运行时 (BGCE位=1), 不要改变TODIS、BGCS1和BGCS0位的值。在设置BGCE位之前要先对TODIS、BGCS1和BGCS0位进行设置。	p. 332 <input type="checkbox"/>	
				当BGCE位被清除时 (清为0), 则8位计数器即被清除。	p. 332 <input type="checkbox"/>	
			PRSCM 寄存器	当间隔定时器BRG正在运行时不要对PRSCM寄存器进行重写 (PRSM.BGCE位=1)。对BGCE位进行设置(1)之前要先设置PRSCM寄存器。	p. 333 <input type="checkbox"/>	
			WTM 寄存器	当WTM0和WTM1位设置为0时, 将WTM2重新写入到WTM7位中。	p. 337 <input type="checkbox"/>	
	硬件		作为钟表定时器的操作	启动之后, 产生第一个钟表定时器中断请求 (INTWT)之前需要一段时间 (WTM.WTM1 和 WTM.WTM0 位 = 11)。	p. 339 <input type="checkbox"/>	
				产生第一次INTWT需要0.515625 (最大) 秒 ($2^9 \times 1/32768 = 0.015625$ (最大) 秒)。然后每隔0.5秒产生INTWT。	p. 339 <input type="checkbox"/>	
	软件		当钟表定时器和间隔定时器BRG同时运行时	钟表定时器将副时钟设置为计数时钟时, 间隔定时器 BRG 的时间间隔可以设置为任意值。改变此间隔时间并不会对钟表定时器造成影响 (改变间隔时间前要终止其运行)。 当对钟表定时器使用主时钟作为计数时钟时, 要将间隔定时器BRG的时间间隔设置为65.536 kHz (近似值)。不要更改此数值。	p. 340 <input type="checkbox"/>	
				当间隔定时器 BRG和间隔定时器WT同时运行时	间隔定时器 WT 使用副时钟作为计数时钟时, 间隔定时器 BRG 和 WT 的时间间隔可以被设置为任意值。也可以在以后的过程中对其进行更改 (改变它的值之前要终止其运行)。 当对间隔定时器WT使用主时钟作为计数时钟时, 间隔定时器BRG的时间间隔可以设置为任意值, 但是以后不能再对其进行改变 (只有当间隔定时器WT终止运行时才能对其进行更改)。间隔定时器WT的时间间隔可以设置成 $2^5 \sim 2^{12}$ (间隔定时器BRG中的集合值)。它也可以在以后的过程中进行更改。	p. 340 <input type="checkbox"/>
				当钟表定时器和间隔定时器WT同时运行时	定时器 WT 的时间间隔可以设置为 488 μs 和 62.5 ms之间的值。以后不能对其进行更改。 当钟表定时器正在运行时不要终止间隔定时器WT (清除(0) WTM.WTM0位)。如果在清除(0)之后对WTM0位进行设置(1), 则钟表定时器将会产生0.5或0.25秒的偏差。	p. 340 <input type="checkbox"/>
当钟表定时器、间隔定时器BRG和间隔定时器WT同时运行时				钟表定时器使用副时钟作为计数时钟时, 间隔定时器 BRG 和 WT 的时间间隔可以设置为任意值。在以后的过程中可以对间隔定时器 BRG 的时间间隔进行改变 (改变前要终止其运行)。 钟表定时器使用主时钟作为计数时钟时, 将间隔定时器 BRG 的时间间隔设置约为 65.536 kHz。以后不能对其进行改变。间隔定时器 WT 的时间间隔可以设置为 488 μs 和 62.5 ms之间的值。以后也不能对其进行改变。 当钟表定时器正在运行时, 不要终止间隔定时器 BRG (清除(0) PRSM.BGCE位) 或间隔定时器WT (清除(0) WTM.WTM0位)。	p. 340 <input type="checkbox"/>	
第十一章	软件	看门狗定时器功能	WDTM1 寄存器	当主时钟被停止且CPU运行在次时钟状态下时, 请勿访问WDTM1寄存器。 关于更多细节, 敬请参阅3.4.8 (2)。	p. 344 <input type="checkbox"/>	
				一旦RUN1位被设定 (为1) 时, 它将被软件清零。 因此, 当计时开始时, 它只能被复位操作停止。	p. 344 <input type="checkbox"/>	
				一旦 WDTM13 和 WDTM14 位被复位 (为 1), 它们就不能被软件清除 (为0), 仅可以通过复位清除。	p. 344 <input type="checkbox"/>	

分类	功能	功能细节	注意事项	页码
第十一章 软件	看门狗定时器功能	看门狗定时器1的操作	当选择副时钟作为CPU时钟时，看门狗定时器1计数停止(看门狗定时器1数值保持)。	p. 345 <input type="checkbox"/>
			INTWDT1信号的非屏蔽中断服务，参见17.10注意事项。	p. 345 <input type="checkbox"/>
		用于间隔定时器的操作	一旦设置WDTM14位为1（即选择看门狗定时器1模式），只有复位才能使其进入间隔定时器模式。	p. 346 <input type="checkbox"/>
			当选择副时钟作为CPU时钟时，看门狗定时器1计数停止（看门狗定时器1数值保持）。	p. 346 <input type="checkbox"/>
		看门狗定时器功能2	复位释放后，看门狗定时器2会自动启动。 当不使用看门狗定时器2，在复位前通过该功能使其停止工作，或清除看门狗定时器2并在下一次间隔开始前使其停止工作。 为了确认操作的正确性，即使在不需要改变默认设置（复位模式，间隔时间： $f_{xx}/2^{25}$ ）的情况下，也要对寄存器WDTM2执行一次写操作。	p. 347 <input type="checkbox"/>
			由不可屏蔽中断请求信号（INTWDT2）引发的不可屏蔽中断服务，可参见17.10注意事项。	p. 347 <input type="checkbox"/>
		WDTM2寄存器	当主时钟停止，CPU工作在副时钟时，不要访问WDTM2寄存器。 的更多细节，请参见3.4.8（2）。	p. 348 <input type="checkbox"/>
			若要停止看门狗定时器2，向WDTM2寄存器写入“1FH”。	p. 348 <input type="checkbox"/>
			如果复位后2次写WDTM2寄存器，则会强制输出溢出信号。	p. 348 <input type="checkbox"/>
			若要有意产生一个溢出信号，向WDTM2寄存器仅写入两次数据，或向WDTE寄存器仅写入一次不等于“ACH”的数据。 但看门狗定时器2停止工作时不产生溢出信号，即使向WDTM2寄存器仅写入两次数据，或向WDTE寄存器仅写入一次不等于“ACH”的数据。	p. 348 <input type="checkbox"/>
		WDTE寄存器	如果“ACH”以外的数值被写入寄存器WDTE，必定产生溢出信号。	p. 349 <input type="checkbox"/>
			当向寄存器WDTE执行1位存储器操作指令，必定产生溢出信号。	p. 349 <input type="checkbox"/>
			读取寄存器WDTE的值为“9AH”（与写入值“ACH”不同）。	p. 349 <input type="checkbox"/>
			若要有意产生一个溢出信号，向WDTM2寄存器仅写入两次数据，或向WDTE寄存器仅写入一次不等于“ACH”的数据。 但看门狗定时器2停止工作时不产生溢出信号，即使向WDTM2寄存器仅写入两次数据，或向WDTE寄存器仅写入一次不等于“ACH”的数据。	p. 349 <input type="checkbox"/>
第十二章 软件	实时输出功能 (RTO)	RTBLO和RTBH0寄存器	当向寄存器RTBH0的第6、7位进行写操作时，写入的数据恒为0。	p. 352 <input type="checkbox"/>
			当主时钟停止,CPU使用副时钟进行操作时，禁止访问寄存器RTBLO和RTBH0。详细情况，参见3.4.8（2）。	p. 352 <input type="checkbox"/>
		寄存器RTBLO和RTBH0的操作	设置实时输出端口后，要在产生实时输出触发之前，将欲输出的数据设置到寄存器RTBLO和RTBH0。	p. 352 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码
第十二章	软件	实时输出功能 (RTO)	RTPM0 寄存器	为了将实时输出信号 (RTPOUT00 ~ RTPOUT05) 从引脚 (RTP00 ~ RTP05) 输出, 需要通过PMC5和PFC5寄存器将这些引脚设置为实时输出口。	p. 353 <input type="checkbox"/>
				通过使能实时输出操作 (RTPC0.RTPOE0 位 = 1), 允许实时输出的那些位将执行实时输出操作, 禁止实时输出的那些位将输出0。	p. 353 <input type="checkbox"/>
				如果禁止实时输出 (RTPOE0 位 = 0), 实时输出信号 (RTPOUT00 ~ RTPOUT05) 全部输出0, 无论RTPM0寄存器如何设置。	p. 353 <input type="checkbox"/>
			RTPC0 寄存器	当禁止实时输出操作 (RTPOE0位 = 0) 时, 实时输出信号的所有位 (RTPOUT00 ~ RTPOUT05) 输出“0”。	p. 354 <input type="checkbox"/>
				只有当RTPOE0 = 0时, 设置BYTE0和EXTR0位。	p. 354 <input type="checkbox"/>
			RTPOE0 位	如果当RTPOE0 = 0时写寄存器RTBH0 和RTBL0, 数据被分别发送到实时输出锁存OH 和OL。	p. 356 <input type="checkbox"/>
				即使当RTPOE0 = 1时写寄存器RTBH0 和RTBL0, 数据也不被发送到实时输出锁存OH 和OL。	p. 356 <input type="checkbox"/>
			实时输出信号	为了将实时输出信号 (RTPOUT00 ~ RTPOUT05) 从引脚输出, 需要通过PMC5和PFC5寄存器设置实时输出口 (RTP00 ~ RTP05)。	p. 356 <input type="checkbox"/>
			冲突	避免下述软件冲突。 • 实时输出允许/禁止转换 (RTPOE0) 和所选择的实时输出触发之间的冲突。 • 实时输出允许状态下写寄存器RTBH0 和RTBL0, 与所选择的实时输出触发之间的冲突。	p. 356 <input type="checkbox"/>
			初始化	在操作初始化前, 停止实时输出 (RTPOE0 位 = 0)。	p. 356 <input type="checkbox"/>
	复位	一旦实时输出被禁止 (RTPOE0 = 0), 确保在再次允许实时输出 (RTPOE0 = 0 → 1) 前初始化寄存器RTBH0 和RTBL0。	p. 356 <input type="checkbox"/>		
	硬件	安全功能	无论端口设置如何, P50 ~ P55引脚将全部通过INTP0引脚置为高阻。	p. 357 <input type="checkbox"/>	
			以下寄存器中与P50 ~ P55引脚相关的所有位初始化。 • P5 寄存器 • PM5 寄存器 • PMC5 寄存器 • PU5 寄存器 • PFC5 寄存器	p. 357 <input type="checkbox"/>	
			PLLCTL 寄存器	从实时输出口 (RTP00 ~ RTP05) 输出数值前, 选择INTP0引脚中断边沿检测, 并设置RTOST0位。	p. 358 <input type="checkbox"/>
			通过INTP0引脚将P50 ~ P55引脚置为高阻之后, 需要先取消安全功能, 再重新将端口 (P50 ~ P55 引脚) 设置为实时输出口。 [重新设置端口步骤] <1> 取消安全功能, 并通过将RTOST0位清零禁止端口设置。 <2> 将RTOST0位置1 (仅在必要时)。 <3> 重新设置为实时输出口。	p. 358 <input type="checkbox"/>	
			务必将4~7位清零。改变第3位的值不影响操作。	p. 358 <input type="checkbox"/>	

章节	分类	功能	功能细节	注意事项	页码
第十三章	硬件	A/D 转换器	A/D 转换器	使用A/D转换器时，AVREF0需要与VDD 和EVDD处于同一电势。	p. 359 <input type="checkbox"/>
			ADM 寄存器	正常模式下(ADHS1, ADHS0 位 = 00)，在 A/D 转换操作 (ADCS 位 = 1) 中，禁止写入 ADM 寄存器。	p. 363 <input type="checkbox"/>
				高速模式下 (ADHS1, ADHS0 bits = 10 或 01)，在A/D转换操作时，如在 ADM中写入同值，转换中止并重新启动。在A/D操作中严禁写入FR2 ~ FR0, ADHS1, 和 ADHS0位。	
				禁止设置ADHS1 和 ADHS0位为11。	p. 363 <input type="checkbox"/>
				在主时钟停止，副时钟操作时，禁止访问 ADM寄存器。详情参见3.4.8 (2) 访问特殊片上外围I/O寄存器。	p. 363 <input type="checkbox"/>
			设置ADCS 位和 ADCS2 位	如果ADCS 和 ADCS2位从00B变为10B用于提升,参考电压生成器自动开启。如果 ADCS位清零而ADCS2为0，电压生成器自动关闭。在软件触发模式下 (ADS.TRG 位 = 0)，严禁使用第一个A/D转换结果。 在硬件触发模式下 (TRG=1)，只有在用于提升电压的参考电压生成器的振稳定时间结束后启动A/D转换时，才可使用A/D转换结果。	p. 365 <input type="checkbox"/>
				如果ADCS 和 ADCS2位从00B变为11B,用于提升电压的参考电压生成器自动开启。如果ADCS位清零而ADCS2为1，电压生成器保持开启。在软件触发模式下 (TRG 位 = 0)，禁止使用A/D转换结果。 在硬件触发模式下(TRG 位 = 1)，只有在用于提升电压的参考电压生成器的振稳定时间结束后启动A/D转换时，才可使用A/D转换结果。	p. 365 <input type="checkbox"/>
			操作顺序	当ADCS位设置为 (1) 和ADCS2位设置为 (1) 时，操作用于提升电压的参考电压生成器需要1 μ s (高速模式) 或 14 μ s (正常模式)。ADCS2位设置为 (1) 后，用于提升电压的参考电压生成器需要工作1 μ s (高速模式) 或14 μ s (正常模式) 或更长时间，才能将ADCS2位设置为 (1)。	p. 365 <input type="checkbox"/>
			ADS 寄存器	EGA1和 EGA0位仅在硬件触发模式 (TRG 位 = 1) 和外部触发模式下 (ADTRG 引脚输入 ADTMD 位 = 1) 有效。	p. 366 <input type="checkbox"/>
				ADTMD位仅在硬件触发模式下有效 (TRG 位 = 1)。	p. 366 <input type="checkbox"/>
				正常模式下 (ADM.ADHS1, ADM.ADHS0 位 = 00) 在A/D转换操作中 (ADM.ADCS 位 = 1)，禁止写入ADS寄存器。	p. 366 <input type="checkbox"/>
				正常模式下ADHS1, ADHS0 位 = 00) 在A/D转换操作中 (ADCS 位 = 1)，禁止冗余输入软/硬件触发。	p. 366 <input type="checkbox"/>
				在主时钟停止副时钟运行时禁止访问ADS寄存器。关于更多细节，敬请参考3.4.8 (2) 访问特殊片上外围I/O寄存器。 一定要将位3清至“0”。	p. 366 <input type="checkbox"/>
			ADCR 和 ADCRH 寄存器	在主时钟停止副时钟运行时禁止访问 ADCR和ADCRH寄存器。关于更多细节，敬请参考3.4.8 (2) 访问特殊片上外围I/O寄存器。	p. 367 <input type="checkbox"/>
			PFM 寄存器	正常模式下 (ADM.ADHS1, ADM.ADHS0 位 = 00) 在A/D转换操作中 (ADM.ADCS 位 = 1)，禁止写入PFM寄存器。	p. 369 <input type="checkbox"/>
				在主时钟停止副时钟运行时禁止访问 PFM寄存器。关于更多细节，敬请参考3.4.8 (2) 访问特殊片上外围I/O寄存器。	p. 369 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码
第十三章	软件	A/D 转换器	PFT 寄存器	正常模式下 (ADM.ADHS1, ADM.ADHS0 位 = 00) 在 A/D 转换操作中 (ADM.ADCS 位 = 1), 禁止写入 PFM 寄存器。	p. 369 <input type="checkbox"/>
				在主时钟停止副时钟运行时禁止访问 PFM 寄存器。关于更多细节, 敬请参考 3.4.8 (2) 访问特殊片上外围 I/O 寄存器。	p. 369 <input type="checkbox"/>
			在 A/D 转换中使用 A/D 转换器	<1> ~ <3> 之间的时间间隔必须为 1 μ s (高速模式) 或 1 4 μ s (正常模式) 或更多。	p. 376 <input type="checkbox"/>
				步骤 <1> 和 <2> 可调换。	p. 376 <input type="checkbox"/>
				步骤 <1> 可忽略。然而, 如忽略步骤 <1>, 在步骤 <3> 后不要使用首次转换结果。	p. 376 <input type="checkbox"/>
	待机模式时的耗电量	步骤 <4> ~ <7> 之间的时间间隔与 ADHS1, ADHS0, 和 FR2 ~ FR0 位设置的转换时间不同。	p. 376 <input type="checkbox"/>		
		步骤 <6> ~ <7> 之间的时间间隔与 ADHS1, ADHS0, 和 FR2 ~ FR0 位设置的转换时间相同。	p. 376 <input type="checkbox"/>		
	在待机模式时 A/D 转换器操作停止。这时, 可通过停止转换模式 (ADM.ADCS 位 = 0) 和参考电压生成器 (ADM.ADCS2 位 = 0) 来减少耗电量。	p. 377 <input type="checkbox"/>			
	硬件	ANI0 ~ ANI7 引脚的输入范围	输入指定范围的电压值到引脚 ANI0 ~ ANI17。如果大于等于 AVREF0 或小于等于 AVSS (即使在最大绝对额定值范围内) 的电压被输入到这些引脚, 则那个通道的转换值不确定, 其它通道的转换值也会受到影响。	p. 377 <input type="checkbox"/>	
	软件	冲突的操作	在转换结束时, 写入 ADCR 寄存器与读取 ADCR 寄存器冲突。	p. 377 <input type="checkbox"/>	
			读取 ADCR 优先。读取之后, 新的转换结果写入 ADCR 寄存器。	p. 377 <input type="checkbox"/>	
	硬件	抑制噪声的方法	在转换结束时, 写入 ADCR 寄存器与写入 ADM 寄存器或写入 ADS 寄存器冲突。写入 ADM 寄存器或写入 ADS 寄存器优先。没写入 ADCR 或生成转换结束中断请求信号 (INTAD)。	p. 377 <input type="checkbox"/>	
			为了确保 10 位分辨率, 注意 AVREF0 和 ANI0 ~ ANI7 引脚上的噪声。噪声影响随着模拟输入源的输出阻抗的增加而提高。为了降低噪声, 推荐连接一个外部电容, 如图 13-8 所示。	p. 378 <input type="checkbox"/>	
如果可能产生大于等于 AVREF0 或小于等于 AVSS 的噪声, 则用一个 VF 较小 (小于等于 0.3V) 的二极管钳位。		p. 378 <input type="checkbox"/>			
ANI0/P70 ~ ANI7/P77 引脚		模拟输入引脚 (ANI0 ~ ANI11) 可功能复用为输入端口引脚 (P70 ~ P77)。 当从引脚 ANI0 ~ ANI11 中选择一个执行 A/D 转换时, 不要在转换期间对端口 7 执行输入指令, 否则转换分辨率将下降。 如果某引脚正在进行 A/D 转换, 其临近引脚输入一个数字脉冲, 则由于耦合噪声的影响, A/D 转换值可能与预期值有出入。因此, 在 A/D 转换过程中, 确保不使用临近引脚传输脉冲信号。	p. 378 <input type="checkbox"/>		
AVREF0 引脚的输入阻抗	AVREF0 引脚和 AVSS 引脚之间串联着数十个 k Ω 的电阻。 因此, 如果参考电压源的输出阻抗较高, 输出阻抗将与 AVREF0 引脚和 AVSS 引脚之间的串联电阻串联, 导致产生较大的参考电压误差。	p. 378 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第十三章	软件	A/D 转换器	中断请求标志 (ADIC.ADIF 位)	即使寄存器 ADS 的内容改变, 中 ADIF 位也不清零。 因此, 如果在 A/D 转换过程中模拟输入引脚改变, ADIF 位可能设置为 (1), 因为在上一次模拟输入引脚的 A/D 转换结束后 ADS 寄存器才立即被复写。在该情况下, 如果寄存器 ADS 复写后立即读取标志 ADIF, 标志 ADIF 可能置 1, 即使新选择的模拟输入信号的转换还未完成。 当 A/D 转换停止, 在再次转换前使标志 ADIF 清零。	p. 379 <input type="checkbox"/>
			A/D 转换后的首次转换值	在 ADM.ADCS2 位设置为 1 后如果 ADM.ADCS 位在 1 μ s (高速模式) 或 1 4 μ s (正常模式) 内设置为 1, 或如果 ADCS2 清零后设置 ADCS 位为 1, A/D 转换操作启动后的首次转换值可能不会符合标准。采取相关措施如发出 A/D 转换结束中断请求信号 (INTAD) 和丢弃首次转换结果。	p. 379 <input type="checkbox"/>
			读取 A/D 转换结果寄存器 (ADCR)	当执行寄存器 ADM 或 ADS 的写指令, 寄存器的 ADCR 内容可能不确定。当转换操作完成后, 在写入 ADM 或 ADS 寄存器前读取转换值。如果不按照以上时序读取, 可能无法读取正确的转换值。 当主时钟 (f_x) 振荡停止, CPU 在副时钟下操作时, 禁止访问 ADCR 和 ADCRH 寄存器。关于更多细节, 敬请参考 3.4.8 (2) 访问特殊片上外围 I/O 寄存器。	p. 379 <input type="checkbox"/>
			A/D 转换器采样时间和 A/D 转换启动延迟时间	A/D 转换采样时间因 ADM 的设置值不同有所变化。A/D 转换器操作使能后, 经过一定延迟时间后, 才开始采样。 在进行设置时需注意查看 A/D 转换时间, 并注意表 13-10 和表 13-4 所示的内容。	p. 380 <input type="checkbox"/>
			寄存器写入反应时间, 触发反应时间	反应时间是指等待期之后的时间。关于等待功能, 参考 3.4.8 (2) 访问特殊片上外围 I/O 寄存器。	p. 381 <input type="checkbox"/>
			A/D 转换结果的浮动	由于供电电压的波动, A/D 转换结果可能产生变化, 或者可能受到噪声影响。为了减小浮动, 使用多次测量求平均值的方法。	p. 382 <input type="checkbox"/>
			A/D 转换结果滞后特性	逐次逼近 A/D 转换器维持内部采样和保持电容的模拟输入电压, 同时进行 A/D 转换。A/D 转换结束后模拟输入电压保留在内部采样和保持电容中。结果, 可能产生下述现象。 • 当同一个通道用于 A/D 转换, 如果电压高于或低于先前 A/D 转换, 则出现滞后特征, 转换结果受到上次转换值影响。因此, 即使同一个模拟输入电压也会产生不同结果。 • 当改变模拟输入通道, 可能出现滞后特征, 转换结果受到上次转换通道影响。这是因为只有一个 A/D 转换器用于 A/D 转换。因此, 即使同一个模拟输入电压也会产生不同结果。 因此, 为获得更为精确的转换值, 在同一通道下连续两次操作 A/D 转换过程, 丢弃首次转换结果。	p. 382 <input type="checkbox"/>
正常模式下的 A/D 转换操作	<ul style="list-style-type: none"> 软件触发模式: 正常模式下在转换过程中, 禁止写入 ADM, ADS, PFM, 或 PFT 寄存器。(ADM.ADHS1, ADM.ADHS0 位 = 00)。 硬件触发模式 (外部触发/定时器触发): 在正常模式下无法使用该模式 (ADHS1, ADHS0 位 = 00), 可在高速模式下使用 (ADHS1, ADHS0 位 = 10 或 01)。 	p. 382 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第十四章	软件	异步串行接口 (UART)	ASIMn 寄存器	在使用UARTn时, 确保在设置CKSRn和BRGCn 寄存器前设置与UARTn功能相关的外部引脚到控制模式, 然后设置UARTEn位到1, 在开始设置其他位。	p. 390 <input type="checkbox"/>
				当输入高电平到RXDn引脚时将UARTEn和RXEn位设为1。当输入低电平到RXDn引脚将这些位设置为1, 接收将会开始。	p. 390 <input type="checkbox"/>
				当禁止接收时, 接收移位寄存器不检测开始位。不对RXBn 寄存器进行移入处理或传输处理, 同时保留RXBn 寄存器的内容。 当允许接收时, 接收移位操作开始, 并与开始位的检测同步。当一个帧的接收完成时, 接收移位寄存器的内容被传送到RXBn 寄存器。与此同时, 产生一个接收完成中断请求信号 (INTSRn)。	p. 391 <input type="checkbox"/>
			ASISn 寄存器	当在ASIMn.UARTEn位或ASIMn.RXEn位清零或者读取ASISn 寄存器时, PEn、FEn和OVerEn位清零。	p. 392 <input type="checkbox"/>
				禁止使用位操作指令进行操作。	p. 392 <input type="checkbox"/>
				当停止主时钟, CPU运行在副时钟上时, 不要访问ASISn 寄存器。关于更多细节, 敬请参阅3.4.8 (2)。	p. 392 <input type="checkbox"/>
			传输中断	正常情况下, 当传输移位寄存器边变空时, 则生成INTSTn信号。但如果传输移位寄存器是由于复位而变空的, 则不生成INTSTn信号。	p. 398 <input type="checkbox"/>
			TXBFn 位	在连续传输中, ASIF.TXBFn和ASIF.TXSFn位的值变化如下10 → 11 → 01。因此, 不要基于TXBFn和TXSFn位的组合确认状态。在连续传输的过程中仅读取TXBFn位。	p. 400 <input type="checkbox"/>
				在连续进行传输时, 将第一个传输数据 (第一个字节) 写入到TXBn 寄存器并确认TXBFn位是0, 然后将第二个的传输数据 (第二个字节) 写入到TXBn 寄存器。如果当TXBFn位是1时写入到TXBn 寄存器, 则无法保证传输数据。	p. 400
			TXSFn 位	在连续传输中, ASIF.TXBFn和ASIF.TXSFn位的值变化如下10 → 11 → 01。因此, 不要基于TXBFn和TXSFn位的组合确认状态。在连续传输的过程中仅读取TXBFn位。	p. 400 <input type="checkbox"/>
				在连续传输完成的情况下初始化传输单元时, 确认在出现传输完成中断后TXSFn位是0, 然后执行初始化。如果在TXSFn位是1时进行初始化, 则无法保证传输数据。	p. 400 <input type="checkbox"/>
				在连续进行传输时, 如果在执行一个数据帧的传输后的INTSTn中断服务前完成了下一次传输, 则可能出现超运转误差。可以通过嵌入可以计数传输数据和参照TXSFn位次数的程序检测超运转误差。	p. 400 <input type="checkbox"/>
			UARTn接收完成中断配时	即使出现接收误差的情况下, 必须读取RXBn 寄存器。如果没有读取RXBn 寄存器, 在下次接收数据时将出现超运转误差, 且接收误差状态经无限期延续。	p. 405 <input type="checkbox"/>
				接收操作总是在假设停止位长度为1的情况下进行。 第二个停止位会被忽略。	p. 405 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码
第十四章	软件	异步串行接口 (UART)	波特率生成器 n (BRGn)	设置f _{UCLK} 满足以下条件。 • V _{DD} = 4.5 ~ 5.5 V: f _{UCLK} ≤ 12 MHz • V _{DD} = 2.7 ~ 4.5 V: f _{UCLK} ≤ 6 MHz	p. 409 <input type="checkbox"/>
				ASCK0引脚的输出只能由UART0使用。	p. 409 <input type="checkbox"/>
			CKSRn 寄存器	在重新写入到TPSn3到TPSn0位前将ASIMn.UARTEn位清零。	p. 410 <input type="checkbox"/>
				设置f _{UCLK} 满足以下条件。 • V _{DD} = 4.5 ~ 5.5 V: f _{UCLK} ≤ 12 MHz • V _{DD} = 2.7 ~ 4.5 V: f _{UCLK} ≤ 6 MHz	p. 410 <input type="checkbox"/>
				ASCK0引脚的输入时钟只能由UART0使用。 禁止设置UART1和UART2。	p. 410 <input type="checkbox"/>
			BRGCn 寄存器	如果要覆盖MDLn7到MDLn0位, 应首先将ASIMn.TXEn和ASIMn.RXEn位清零。	p. 411 <input type="checkbox"/>
			波特率误差	确保传输过程中的波特率误差不超过接收目标的允许误差。	p. 412 <input type="checkbox"/>
				确保接收过程中的波特率误差在接收过程中允许的波特率范围内, 如“14.6.4.接收过程中允许的波特率范围”中所述。	p. 412 <input type="checkbox"/>
			波特率生成器设置	基础时钟 (f _{UCLK}) 的允许频率如下。 • V _{DD} = 4.5 ~ 5.5 V: f _{UCLK} ≤ 12 MHz • V _{DD} = 2.7 ~ 4.5 V: f _{UCLK} ≤ 6 MHz	p. 413 <input type="checkbox"/>
			接收过程中允许的波特率范围	下述的等式应用于设定接收过程的波特率, 使得其始终在允许的误差范围内。	p. 414 <input type="checkbox"/>
UARTn 的注意事项	当向UARTn供应时钟停止时 (比如在空闲或停止模式), 操作停止, 每个寄存器保留在停止供应时钟前的值。TXDn引脚输出同样保留和输出在停止提供时钟前的值。在时钟供应重新启动后, 则不能保证操作。因此, 在供应重新启动后, 应通过将ASIMn.UARTEn、ASIMn.RXEn和ASIMn.TXEn位清为000。	p. 416 <input type="checkbox"/>			
	UARTn有两个阶段的缓冲配置, 包括TXBn寄存器和传输移位寄存器以及显示每个缓冲器状态的状态标记 (ASIFn.TXBFn和ASIFn.TXSFn位)。如果TXBFn和TXSFn位在连续传输中读取, 值的变化如下: 10 → 11 → 01。对于写入下一个数据到TXBn寄存器的配时, 在连续传输过程中仅读取TXBFn位。	p. 416 <input type="checkbox"/>			
第十五章	软件	时钟控制的串行接口 (CSI0)	CSIM0n 寄存器	只有当CSOTn位 = 0时才能覆盖CSIM0n.TRMDn、CSIM0n.CCLn、CSIM0n.DIRn、CSIM0n.CSITn和CSIM0n.AUTOn位。如果在其他时间覆盖了这些位, 则不能保证运行。	p. 421 <input type="checkbox"/>
			CSICn 寄存器	只有在CSIM0n.CSI0En位=0时才能覆盖CSICn寄存器。	p. 423 <input type="checkbox"/>
				设置系列时钟以满足以下条件。 • V _{DD} = 4.0 ~ 5.5 V: 串行时钟 ≤ 5 MHz • V _{DD} = 2.7 ~ 4.0 V: 串行时钟 ≤ 2.5 MHz	p. 423 <input type="checkbox"/>
			SIRBn 和 SIRBnL 寄存器	只有当设置了16位数据长度 (CSIM0n.CCLn位 = 1) 时才读取SIRBn寄存器。	p. 424 <input type="checkbox"/>
只有当设置了8位数据长度 (CCLn位 = 0) 时才读取SIRBnL寄存器。 当设置了单传输模式时 (CSIM0n.AUTOn位 = 0), 只有在空闲状态IM0n.CSOTn位 = 0) 才进行读取操作。如果在数据传输中读取了SIRBn或SIRBnL寄存器, 则不能保证数据。	p. 424 <input type="checkbox"/>				

章节	分类	功能	功能细节	注意事项	页码
第十五章	软件	时钟控制的串行接口 (CSI0)	SIRBEn 和 SIRBEnL 寄存器	即使从SIRBEn和SIRBEnL寄存器读取了数据也不开始接收操作。	p. 425 <input type="checkbox"/>
				只有当设置了16位数据长度 (CSIM0n.CCLn位 = 1) 时才读取SIRBEn寄存器。 只有当设置了8位数据长度 (CCLn位 = 0) 时才读取SIRBEnL寄存器。	p. 425 <input type="checkbox"/>
			SOTBn 和 SOTBnL 寄存器	只有当设置了16位数据长度 (CSIM0n.CCLn位 = 1) 时才读取SOTBn寄存器。 只有当设置了8位数据长度 (CCLn位 = 0) 时才读取SOTBnL寄存器。	p. 426 <input type="checkbox"/>
				当设置了单传输模式时 (CSIM0n.AUTOn位 = 0)，只有在空闲状态IM0n.CSOTn位 = 0) 才进行访问。如果在数据传输中访问了SOTBn和SOTBnL寄存器，则不能保证数据。	p. 426 <input type="checkbox"/>
			SOTBFn 和 SOTBFnL 寄存器	只有当设置了16位数据长度 (CSIM0n.CCLn位 = 1) 时，设置了8位数据长度 (CCLn位 = 0) 时，或者只有在空闲状态 (CSIM0n.CSOTn位 = 0) 才访问SOTBFn和SOTBFnL寄存器。如果在数据传输过程中访问了SOTBFn和SOTBFnL寄存器，则不能保证数据。	p. 427 <input type="checkbox"/>
			SIO0n 和 SIO0nL 寄存器	只有当设置了16位数据长度 (CSIM0n.CCLn位 = 1) 时，设置了8位数据长度 (CCLn位 = 0) 时，或者只有在空闲状态 (CSIM0n.CSOTn位 = 0) 才读取SIO0n寄存器和SIO0nL寄存器。如果在数据传输过程中读取了SIO0n寄存器和SIO0nL寄存器，则不能保证数据。	p. 428 <input type="checkbox"/>
			CSIM0n.CSITn位 = 1	只有在主模式 (CSICn.CKS0n2到CSICn.CKS0n0位不是111B) 下延迟模式 (CSIM0n.CSITn位 = 1) 才有效。在从模式 (CKS0n2到CKS0n0位是111B) 下不要设置延迟模式。	p. 430 <input type="checkbox"/>
			CSIM0n.CSOTn位 = 1	当CSOTn位 = 1时，不要操作CSI0n寄存器。	p. 432 <input type="checkbox"/>
连续传输模式的注意事项	要继续连续传输，必须在传输预定期内读取SIRBn寄存器或写入到SOTBn寄存器。	p. 441 <input type="checkbox"/>			
	在发生传输请求清除和寄存器访问之间的冲突的情况下。由于传输请求优先级较高，下一次传输请求被忽略。因此传输中断，不能进行正常的数据传输。	p. 441 <input type="checkbox"/>			
	在发送传输/接收完成中断请求信号 (INTCSI0n) 生成和寄存器访问之间的冲突的情况下。 由于连续传输立即停止，作为新的连续传输执行。 在从模式下，造成位相误差 (参见图 15-8)。 在传输/接收模式下，重新传输SOTBFn寄存器的值，发送非法的数据。	p. 442 <input type="checkbox"/>			
第十六章	软件	内部IC总线	引脚设置	如果要使用内部IC总线功能，将P38/SDA0和P39/SCL0引脚分别用作串行传输/接收数据输入/输出引脚 (SDA0) 和串行时钟输入/输出引脚 (SCL0)，并将它们设为N通道漏极开路输出。	p. 444 <input type="checkbox"/>
			IICC0 寄存器	如果在SCL0线处于高电平且SDA0线处于低电平的情况下启用I ² C0的运行 (IICE0位=1)，则立即检测开始条件。为了避免此情况，在启用I ² C0的运行后，立即将位操作指令设置为LREL0位=1。	p. 450 <input type="checkbox"/>
			IICC0.SPT0 位	仅在主模式中设置SPT0位为1。但在切换到运行启用状态后检测到第一个停止条件前必须将设置SPT0位置为1并生成一个停止条件。关于更多细节，敬请参阅 16.14 注意事项。	p. 453 <input type="checkbox"/>

	分类	功能	功能细节	注意事项	页码
第十六章	软件	内部IC总线	IICC0.SPT0 位	当IICS0.TRC0设置为1时，第九个时钟期间的WRELO位设置1，等待取消；之后TRC0位被清零，SDA0线被设为高阻抗。	p. 453 <input type="checkbox"/>
			IICS0 寄存器	时钟停止，CPU 运行在副时钟上时，不要访问 IICS0 寄存器。关于详细信息，请参见 3.4.8 (2)。	p. 454 <input type="checkbox"/>
			IICF0 寄存器	只有在运行停止时 (IICE0位= 0) 写入到STCEN0位。	p. 458 <input type="checkbox"/>
				由于当STCEN0位 = 1时不论实际总线的状态如何而识别位总线释放状态 (IICBSY0位=0)，当生成第一个开始条件 (STT0位= 1) 时，必须核实没有第三方的通讯在进行中，从而避免通讯受到破坏。	p. 458 <input type="checkbox"/>
				只有在运行停止时 (IICE0 位= 0) 写入到 IICRSV0 位。	p. 458 <input type="checkbox"/>
			I ² C 中断请求信号	要生成一个停止条件，将WTIMO位设为1 并改变中断请求信号 (INTIIC0) 的生成。	pp. 473 to 475 <input type="checkbox"/>
				要生成一个开始条件，将WTIMO位设为1 并改变中断请求信号 (INTIIC0) 的生成。	p. 474 <input type="checkbox"/>
				将WTIMO位清零以复原设置。	p. 474 <input type="checkbox"/>
			中断请求信号 (INTIIC0) 生成配时和等待控制	当匹配设置到 SVA0 寄存器的地址时，从设备的 INTIIC0 信号和等待期出现在第九个时钟的下降边。此时，不论设置到 IICC0.ACKE0 位的值如何，生成 ACK。对于已经收到扩展代码的从设备，在第八个时钟的下降边出现 INTIIC0 信号。当在重新启动后地址不匹配时，在第九个时钟的下降边产生INTIIC0信号，但不出现等待。	p. 494 <input type="checkbox"/>
				如果收到的地址不匹配SVA0寄存器的内容，而没有收到扩展代码，不出现INTIIC0信号或等待。	p. 494 <input type="checkbox"/>
			判优器	当IICC0.WTIM0位= 1时，中断请求在第九个时钟的下降边出现。当WTIMO位= 0 并收到扩展代码的从设备地址时，中断请求在第八个时钟的下降边出现。当有可能出现判优时，为主设备运行的运行设置SPIE0位= 1。	p. 498 <input type="checkbox"/>
			当 IICF0.STCEN0 位 = 0	当 IICF0.STCEN0 位 = 0 时在启用 I ² C0 后，通讯状态 (IICF0.IICBSY0 位 = 1) 立即被识别而不论实际的总线状态。要在没有检测到停止条件的状态下执行主设备通讯，在开始主设备通讯前生成停止条件然后释放总线。通过以下顺序生成停止条件： <1> 设置 IICCL0 寄存器。 <2> 设置 IICC0.IICE0 位。 <3> 设置IICC0.SPT0位。	p. 503 <input type="checkbox"/>
当 IICF0.STCEN0 位 = 1	在启用I ² C0后，总线释放状态 (IICBSY0位= 0) 立即被识别而不论实际的总线状态。要生成第一个开始条件 (IICC0.STT0位= 1)，必须确认总线已经释放从而避免干扰其他通讯。	p. 503 <input type="checkbox"/>			
IICC0.IICE0 位 =1	当V850ES/KE2的IICC0.IICE0位设为1且与其他设备的通讯在进行中时，可以根据通讯线的状态检测开始条件。确保在 SCL0和SDA0线设为高电平时将IICC0.IICE0位设为1。	p. 503 <input type="checkbox"/>			
IICC0.IICE0 位 = 1	在启动运行 (IICC0.IICE0位= 1) 前通过IICCL0和IICX0寄存器确定运行的时钟频率。要改变运行的时钟频率，立即将 IICC0.IICE0位 清零。	p. 503 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第十六章	软件	内部IC总线	IICC0.STT0 和 IICC0.SPT0 位	在将IICC0.STT0和IICC0.SPT0设为1后，必须首先将两者清零才能复位。	p. 503 <input type="checkbox"/>
			通讯预定	如果已经预定了通讯，将IICC0.SPIE0位设为1，从而通过检测停止条件而生成中断请求。在生成中断请求后，通过向I ² C0写入通讯数据可释放等待状态，然后传输开始。如果检测到一个停止条件不产生一个中断，由于没有生成中断请求，传输在等待状态中将会停止。但是没有必要将软件的SPIE0位设为1以检测IICS0.MSTS0位。	p. 503 <input type="checkbox"/>
			单主系统中的主设备运行	请根据通讯中产品的规格释放内部 IC 总线（SCL0， SDA0 引脚 +高电平）。 例如，当EEPROM TM 输出一个低电平到SDA0引脚时，设置SCL0引脚到输出口，并从该输出口输出时钟脉冲直到SDA0引脚保持高电平。	p. 505 <input type="checkbox"/>
			多主系统中的主设备运行	确认总线释放状态（IICCL0.CLD0位= 1， IICCL0.DAD0位= 1）已经维持了一段时间（如1帧）。当SDA0引脚维持在低水平，通过参照通讯中产品的规格确定是否释放内部IC总线（SCL0， SDA0引脚 = 高电平）。	p. 506 <input type="checkbox"/>
				保证传输和接收格式符合通讯中产品的规格。	p. 508 <input type="checkbox"/>
				当使用V850ES/KE2作为多主系统中的主设备时，在每次出现INTIIC0中断时读取IICS0.MSTS0位以确认判优结果	p. 508 <input type="checkbox"/>
				当使用V850ES/KE2作为多主系统中的从设备时，在每次出现INTIIC0中断时确认使用IICF0寄存器的状态从而确定下一步的处理。	p. 508 <input type="checkbox"/>
			从设备等待取消	要取消从设备等待，写入FFH到IIC0或者设置WRELO。	pp. 513 ~ 515 <input type="checkbox"/>
			主设备等待取消	要取消主设备等待，写入FFH到IIC0或者设置WRELO。	pp. 516 ~ 518 <input type="checkbox"/>
			第十七章	软件	中断/例外处理功能
在非屏蔽中断服务期间，当EP位和NP位通过LDSR指令改变时，在RETI指令执行之后，为了正确的还原PC和PSW，必须在RETI指令执行之前使用LDSR指令将EP位设置为0， NP位设置为1。	p. 526 <input type="checkbox"/>				
可屏蔽中断	在可屏蔽中断服务执行期间，当通过LDSR指令将EP位和NP位改变时，为了在RETI指令后正确的恢复PC和PSW，需要在RETI指令之前执行LDSR指令将EP位和NP位清零。	p. 530 <input type="checkbox"/>			
多重中断	执行多重中断前必须保存EIPC 和EIPSW的值。	p. 532 <input type="checkbox"/>			
中断控制 寄存器	禁止中断（DI）或者屏蔽中断来读取xxlCn.xxIFn位。如果在允许中断（EI）或是中断未被屏蔽的时候，读取xxlFn位，确认中断与读取该位冲突时数据可能有误。	p. 535 <input type="checkbox"/>			
	如果确认中断信号后，xxlFn标志通过硬件自动复位。	p. 535 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码	
第十七章	软件	中断/例外处理功能	IMR0, IMR1, IMR3 寄存器	在此设备文件中, xxICn 寄存器的xxMKn 位定义为一个保留字。因此, 如果以 xxMKn 名称执行位操作, 则xxICn寄存器而不是IMRm寄存器被复写 (结果IMRm寄存器也被复写)。	p. 537 <input type="checkbox"/>	
				为了以8位或1位为单元读取或写入IMR0和IMR1寄存器的位8~位15, 需要将它们设定为IMR0H到IMR3H寄存器的位0~位7。	p. 537 <input type="checkbox"/>	
				将IMR0寄存器的位9和8, IMR1寄存器的位15~8, 以及IMR3寄存器的位15~5和位0置1。如果这些位的设定值改变, 操作将不会得到保证。	p. 537 <input type="checkbox"/>	
				ISPR 寄存器	如果在中断允许状态 (EI) 对ISPR寄存器进行读取时确认一个中断, 在寄存器的位通过确认中断的方式被设置之后, 可以读取 ISPR寄存器的值。为了在确认中断之前正确的读取ISPR寄存器的值, 需要禁止中断 (DI)。	p. 538 <input type="checkbox"/>
				WDTM1 寄存器	一旦RUN1位被置1, 不能由软件将其清零。因此, 一旦计数开始, 只有复位才能使其停止。	p. 540 <input type="checkbox"/>
					一旦WDTM14 和 WDTM13位被置1, 不能由软件将其清零。只有复位才能使这些位清零。	p. 540 <input type="checkbox"/>
				INTR0 和 INTF0 寄存器	在引脚的功能从外部中断功能 (复用功能) 转变到端口功能的过程中, 可能会执行边沿检测。因此, 首先要将INTF0n位和INTR0n位设定为00, 然后再设定端口模式。	p. 544 <input type="checkbox"/>
				INTR3 和 INTF3 寄存器	在引脚的功能从外部中断功能 (复用功能) 转变到端口功能的过程中, 可能会执行边沿检测。因此, 首先要将INTF31 和 INTR31位设定为00, 然后再设定端口模式。	p. 545 <input type="checkbox"/>
				INTR9H 和 INTF9H 寄存器	当引脚的功能从外部中断功能 (复用功能) 转变到端口功能时, 可以进行边沿检测。因此, 首先要将INTF9n位和INTR9n位设定为00, 然后再设定端口模式。	p. 546 <input type="checkbox"/>
				软件例外进程中恢复	在软件例外处理过程中, 当通过 LDSR 指令改变 EP 位和 NP 位时为了在 RETI 指令执行后正确的恢 PC 和 PSW, 在 RETI 指令执行前必须立即执行 LDSR 指令将 EP 位置 1, NP 位清零。	p. 548 <input type="checkbox"/>
				非法的操作代码	由于将来有可能分配指令到一个非法的操作代码, 所以建议不要使用非法操作代码。	p. 550 <input type="checkbox"/>
				从非法的操作代码中恢复	只有在执行一条非法代码和执行DBRET指令之间的间隔时才可以访问DBPC and DBPSW。.	p. 551 <input type="checkbox"/>
				从调试陷阱的处理中恢复	在DBTRAP指令执行后DBRET指令执行之前, 可以访问DBPC和DBPSW。	p. 553 <input type="checkbox"/>
		在服务程序中产生例外	在非屏蔽中断服务例程中 (到RETI指令执行为止) 可屏蔽中断不被应答并始终保持。	p. 555 <input type="checkbox"/>		

章节	分类	功能	功能细节	注意事项	页码		
第十七章	软件	中断/例外处理功能	从NMI恢复	<p>设计这个系统，从而使非屏蔽中断请求信号(INTWDT1/INTWDT2)触发的非屏蔽中断在服务后，能够通过RETI指令恢复。</p> <p><1> 产生 INTWDT1/INTWDT2</p> <p><2> FEPC ← 软件复位处理地址 FEPSW ← 使NP位 = 1, EP位 = 1的值</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">RETI</p> <p style="text-align: center;">↓</p> <p><3> 十条 RETI 指令 (FEPC 和 FEPSW注 必须设置)</p> <p style="text-align: center;">↓</p> <p>PSW ← PSW的初始设定值</p> <p style="text-align: center;">↓</p> <p style="text-align: center;">初始化</p>	p. 557	<input type="checkbox"/>	
第十八章	硬件	按键中断功能	KR0 ~ KR7 引脚	如果低电平输入给KR0 ~ KR7中的任何一个，那么即使有下降沿输入到另一引脚时，INTKR信号也将不会产生。	p. 558	<input type="checkbox"/>	
				KRM 寄存器	如果KRM寄存器被改变了，一个中断请求信号(INTKR)就将可能产生。为避免此种情况发生，在屏蔽中断(DI)后改变KRM寄存器，然后在清零中断请求标志后(KRIC.KRIF位)使中断起效。	p. 559	<input type="checkbox"/>
第十九章	软件	待机功能	IDLE 模式	PLL不停止。停止PLL，转换到IDLE模式以减少能耗。	p. 560	<input type="checkbox"/>	
				STOP 模式	转入时钟直连模式，停止PLL，然后再转入STOP模式。关于更多细节，敬请参考第5章 时钟生成功能。	p. 560	<input type="checkbox"/>
				PSC 寄存器	如果NMI2M, NMI0M或INTM位设置为1，而STP位同时也设置为1，NMI2M, NMI0M, 或INTM位则变为无效。在IDLE/STOP模式下，如有一个未屏蔽的中断请求信号等待处理，将该中断请求信号所对应的位设置为(NMI2M, NMI0M, 或INTM) 1，然后将STP位也设置为1。	p. 563	<input type="checkbox"/>
					在IDLE/STOP模式设置下，先设置PSMR.PSM位然后设置STP位。	p. 563	<input type="checkbox"/>
				PSMR 寄存器	在连接副时钟共振器时，一定要将XTSTP位清零。	p. 564	<input type="checkbox"/>
					一定要将PSMR寄存器的位1~6清零。	p. 564	<input type="checkbox"/>
					仅当PSC.STP位为1时PSM位有效。	p. 564	<input type="checkbox"/>
				OSTS 寄存器	STOP模式释放后的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)，不管通过复位输入或中断请求信号的产生是否释放了STOP模式。	p. 565	<input type="checkbox"/>
					确定将第3位~第7位清零。	p. 565	<input type="checkbox"/>
					复位释放后的振荡稳定时间是 $2^{15}/f_X$ (因为OSTS寄存器的初始值=01H)。	p. 565	<input type="checkbox"/>
HALT 指令	振荡稳定时间也是在外时钟输入时写入。	p. 565	<input type="checkbox"/>				
	在HALT指令之后插入五个或更多的NOP指令。	p. 566	<input type="checkbox"/>				
IDLE 模式	如果在执行HALT指令时有一个未屏蔽的中断请求被保持等待响应，则此状态被转移到HALT模式中，然后通过这个等待响应的中断请求，HALT模式被立即释放。	p. 566	<input type="checkbox"/>				
	在执行往PSC寄存器中存入数据来设置IDLE模式的指令之后，插入五个或更多的NOP指令。	p. 568	<input type="checkbox"/>				
释放IDLE 模式	通过设置PSC.NMI2M, PSC.NMI0M, 或 PSC.INTM位为1来禁止中断请求信号，这样来禁止中断信号是无效的，IDLE2模式也不会释放。	p. 569	<input type="checkbox"/>				

章节	分类	功能	功能细节	注意事项	页码
第十九章	软件	待机功能	STOP 模式	在执行往PSC寄存器中存入数据来设置STOP模式的指令之后，插入五个或更多的NOP指令。	p. 571 <input type="checkbox"/>
			释放 STOP 模式	通过设置PSC.NMI2M, PSC.NMI0M位 或 PSC.INTM位为1来禁止中断请求信号，这样来禁止中断信号是无效的，IDLE2模式也不会释放。	p. 572 <input type="checkbox"/>
			副时钟模式	当操作CK3位时，不要改变PCC.CK2~ PCC.CK0位（推荐使用位操作指令来操作）的设置值。对于PCC寄存器的细节描述，参见5.3（1）处理器时钟控制寄存器（PCC）。	p. 575 <input type="checkbox"/>
				如果以下的条件不满足，改变CK2~ CK0位的设置以便满足条件并且设置副时钟操作模式。 内部系统时钟（fCLK）>副时钟（fXT: 32.768 kHz）× 4	p. 575 <input type="checkbox"/>
			释放副时钟操作模式	当操作CK3位时，不要改变CK2~ CK0位（推荐使用位操作指令来操作）的设置值。对于PCC寄存器的细节描述，参见5.3（1）处理器时钟控制寄存器（PCC）。	p. 575 <input type="checkbox"/>
			副-IDLE 模式	在执行往PSC寄存器中存入数据来设置副IDLE模式的指令之后，插入五个或更多的NOP指令。	p. 577 <input type="checkbox"/>
			释放副-IDLE 模式	通过设置PSC.NMI2M, PSC.NMI0M, 或 PSC.INTM位为 1来禁止中断请求信号，这样来禁止中断信号是无效的，IDLE2模式也不会释放。	p. 578 <input type="checkbox"/>
第二十一章	硬件	flash存储器	flash存储器	要获得关于flash存储器重写入的电气规范，请参见 第23章 电气规范。	p. 585 <input type="checkbox"/>
			PG-FP4	如图21-6，连接引脚或者通过下拉电阻连接到板上的GND。	p. 593 <input type="checkbox"/>
				连接这些引脚来提供来自PG-FP4的时钟（如图21-6，接线或在板上创建一个振荡器并提供时钟）。	p. 593 <input type="checkbox"/>
				当使用flash存储器编程器的时钟输出时，将编程器的CLK连接到X1，然后连接其反向信号到X2。	p. 594 <input type="checkbox"/>
	FA-80GC-8BT-A	如图所示连接FLMD1引脚或者通过下拉电阻连接该引脚到板上的GND。	p. 596 <input type="checkbox"/>		
		确保在PG-FP4提供时钟时如下图所示设置和连接。 <ul style="list-style-type: none"> •将flash存储器适配器（FA系列）的J1设到VDD侧。 •连接FA的CLKOUT到FA的CLKIN。 •连接FA的X1到设备的X1。 •连接FA的X2到设备的X2。 如果在flash存储器适配器上创建了一个振荡器并提供时钟，则不必进行以上的设置和连接。下面是一个电路的例子。	p. 596 <input type="checkbox"/>		
		通讯模式的选择	当选择了UART0时，基于接收FLMD0脉冲后从专门的flash存储器编程器收到的复位命令计算接收时钟。	p. 598 <input type="checkbox"/>	
硬件	硬件	FLMD1 引脚	如果在片上写入过程中和复位之后立即从别的设备输入VDD信号到FLMD1引脚，隔离此信号。	p. 601 <input type="checkbox"/>	
		FLMD0 引脚	确保当复位释放时FLMD0引脚为0V。	p. 608 <input type="checkbox"/>	
第二十二章	硬件	片上调试功能	注意事项	不要将用于调试的设备安装在大批量生产的产品上，因为调试过程中flash存储器会被改写，flash存储器改写的次数不能保证。此外，不要把调试监测程序嵌入大批量生产的产品中。	p. 618 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码
第二十三章	软件	片上调试功能	注意事项	如果满足下列条件之一，强制断点不能执行。 <ul style="list-style-type: none"> • 中断禁止(DI) • 发给串行接口并用于MINICUBE2和目标设备之前通信的中断被屏蔽 • 禁止由可屏蔽中断释放待机模式时进入待机模式 • MINICUBE2与目标设备之间通信模式为UART0，且主时钟被MINICUBE2停止 	p. 618 <input type="checkbox"/>
				满足如下条件之一时，伪实时RAM监测器(RRM)功能和DMM功能不工作。 <ul style="list-style-type: none"> • 中断被禁止(DI) • 发给串行接口用于MINICUBE2和目标设备之间通信的中断被屏蔽 • 由于禁止可屏蔽中断释放待机而进入待机模式 • MINICUBE2与目标设备之间通信模式为UART0，且主时钟被MINICUBE2停止 • MINICUBE2与目标设备之间通信模式为UART0。且用于通信的时钟与调试器指定的时钟不同 	p. 618 <input type="checkbox"/>
				如果如下条件之一满足，待机模式由伪RRM和DMM功能释放。 <ul style="list-style-type: none"> • MINICUBE2和目标设备之间的通信模式为CSI00 • MINICUBE2和目标设备之间的通信模式为UART0，且主时钟已加入 	p. 618 <input type="checkbox"/>
				需要特定顺序的外围I/O寄存器不能用DMM功能写入。	p. 618 <input type="checkbox"/>
				如果分配了调试监测程序的空间被flash存储器自编程改写，调试器不能再正常工作。	p. 618 <input type="checkbox"/>
		安全 ID	flash存储器擦除后，全部区域写入1。	p. 619 <input type="checkbox"/>	
第二十三章	硬件	电气特性	最大额定值	确保不要超过每个电源电压的最大额定值 (MAX.值)。	p. 622 <input type="checkbox"/>
				不要直接将IC产品的输出 (或I/O) 引脚连接起来，或连接到V _{DD} ，V _{CC} 和 GND。漏极开路引脚或集电极开路引脚，可以直接互相连接。 如果输出引脚设置为高阻抗状态并且避免与外部电路的输出时序冲突，可以直接将IC产品的输出引脚与外部电路连接起来。	pp. 622, 623 <input type="checkbox"/>
				DC特性和AC特性中指出的额定值和条件是正常工作的质量保证。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。	pp. 622, 623 <input type="checkbox"/>
		EEPROM 仿真	不要停止主时钟。	p. 625 <input type="checkbox"/>	

章节	分类	功能	功能细节	注意事项	页码	
第二十三章	硬件	电气特性	主时钟振荡器特性	输入波形的占空比必须在50% ±5%范围内。	p. 626 <input type="checkbox"/>	
				当使用主时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。 <ul style="list-style-type: none"> • 连接线越短越好。 • 连接线不应与其他信号线交叉。 • 流经的电流变化较大的信号线不要在振荡器周围布线。 • 要保持振荡器电容器的接地点电压与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。 	p. 626 <input type="checkbox"/>	
	软件			当主时钟停止副时钟工作时，在转换回主时钟前等待直到振荡稳定时间，由程序保证。	p. 626 <input type="checkbox"/>	
				副时钟振荡器特性	输入波形的占空比必须在50% ±5%范围内。	p. 627 <input type="checkbox"/>
	硬件				当使用副时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。 <ul style="list-style-type: none"> • 连接线越短越好。 • 连接线不应与其他信号线交叉。 • 流经的电流变化较大的信号线不要在振荡器周围布线。 • 要保持振荡器电容器的接地点电压与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。 	p. 627 <input type="checkbox"/>
				软件		
	DC特性	对于P38 和 P39的 IOL参考 I _{OL1} 。	p. 629 <input type="checkbox"/>			
	软件			数据保持特性	转换到STOP模式和从 STOP模式还原必须在额定的工作范围完成。	p. 631 <input type="checkbox"/>
				AC特性	如果由于电路结构加载电容超过50 pF，通过插入缓冲器或其他方法将设备的加载电容降到50 pF以下。	p. 632 <input type="checkbox"/>
	软件		内部IC 总线模式		系统内SDA0n（在SCL0n信号的VIHmin.）信号需要一个最少300ns的保持时间，以使在SCL0n 的下降沿时占用未定义区域。	p. 640 <input type="checkbox"/>
如果系统不延长SCL0n信号低的保持时间（t _{Low} ），仅需要满足最大数据保持时间（t _{HD: DAT} ）。					p. 640 <input type="checkbox"/>	

章节	分类	功能	功能细节	注意事项	页码
第二十三章	软件	电气特性	内部IC总线模式	<p>高速模式I²C总线可以用于正常模式I²C总线系统。在这种情况下，设定高速模式I²C总线，因此会遇到如下条件。</p> <ul style="list-style-type: none"> 如果系统不延长SCL0n信号低的保持时间： tSU: DAT ≥ 250 ns 如果系统延长SCL0n信号低的保持时间： 发送如下数据位到SDA0n优先于SCL0n释放 (tRmax. + tSU: DAT = 1000 + 250 = 1250 ns: 正常模式I²C总线规范)。 	p. 640 <input type="checkbox"/>
	硬件		Flash存储器编程特性	<p>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。</p> <p>示例(P: 写, E: 擦出)</p> <p>出库产品 → P → E → P → E → P: 3次重写</p> <p>出库产品 → E → P → E → P → E → P: 3次重写</p>	p. 643 <input type="checkbox"/>
第二十五章	硬件	推荐焊接条件	推荐焊接条件	不要一起使用不同的焊接方式（除局部加热外）。	p. 645 <input type="checkbox"/>
附录 A	软件	开发工具	RX850 或 RX850 Pro	为了购买RX850 或 RX850 Pro，首先需要填写购买申请表并且签署用户协议。	p. 655 <input type="checkbox"/>
附录 B	软件	指令集列表	指令集	对于通用寄存器reg1 和 reg3，不要指定相同的寄存器。	p. 665 <input type="checkbox"/>

附录 E 修订履历

E.1 在此版本中的主要修订

页码	描述
p. 42	增加 3.2.2 (6) 异常/调试陷阱状态保存寄存器 (DBPC, DBPSW) 的描述
p. 49	增加 3.4.4 (4) 访问所需的时钟数
p. 135	修改 6.4 (3) TMP0 I/O 控制寄存器 0 (TP0IOC0)
p. 178	增加图 6-22 单脉冲输出模式下的寄存器设置的注意事项
p. 288	修改图 8-1 8 位定时器/事件计数器 5n 的框图
p. 316	增加图 9-3 间隔定时器的时序/方波输出操作的注
p. 334	修改 10.1.4 (1) 间隔定时器 BRG 的操作
p. 348	修改 11.2.3 (1) 看门狗定时器模式寄存器 2 (WDTM2) 的注意事项
p. 349	修改 11.2.3 (2) 看门狗定时器使能寄存器 (WDTE) 的注意事项
p. 363	修改 13.4 (1) A/D 转换模式寄存器 (ADM) 的注意事项
p. 366	修改 13.4 (2) 模拟输入通道指定寄存器 (ADS) 的注意事项
p. 369	修改 13.4 (4) 掉电比较模式寄存器 (PFM) 的注意事项
p. 369	修改 13.4 (5) 掉电比较阈值寄存器 (PFT) 的注意事项
p. 371	修改 13.5.2 触发模式
p. 372	修改 13.5.3 (1) 选择模式
p. 373	修改 13.5.3 (2) 扫描模式
p. 374	修改图 13-5 扫描模式操作时序 (ADS.ADS2 到 ADS.ADS0 位 = 011B) 的举例
p. 377	修改图 13-7 如何在待机模式中缩减能耗的举例
p. 382	增加 13.6 (14) 正常模式下的 A/D 转换操作
p. 502	修改表 16-7 等待周期中的等待周期
p. 551	增加 17.6.1 (2) 恢复的注意事项
p. 553	增加 17.6.2 (2) 恢复的注意事项
p. 588	修改表 21-2 基本功能
p. 588	修改表 21-3 安全功能
p. 589	增加表 21-4 安全设置
p. 590	增加 21.3 (1) PG-FP4 的安全设置 (安全标记设置)
p. 592	修改 21.4.2 (1) UART0 的发送速率
p. 599	修改表 21-7 Flash 存储器控制命令
p. 610	修改第 22 章 片上调试功能
p. 628	增加第 23 章 电器特性中的注 3 DC 特性
p. 645	增加第 25 章 推荐焊接条件
p. 646	增加附录 A 开发工具
p. 672	增加附录 D 注意事项列表
p. 699	增加附录 E 修订履历

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

[深圳]

日电电子(中国)有限公司深圳分公司
深圳市福田区益田路卓越时代广场大厦 39 楼
3901, 3902, 3909 室
电话: (+86) 755-8282-9800
传真: (+86) 755-8282-9899

[上海]

日电电子(中国)有限公司上海分公司
中国上海市浦东新区银城中路200号
中银大厦2409-2412和2509-2510室
电话: (+86) 21-5888-5400
传真: (+86) 21-5888-5230

[香港]

香港日电电子有限公司
香港九龙旺角太子道西193号新世纪广场
第2座16楼1601-1613室
电话: (+852) 2886-9318
传真: (+852) 2886-9022
2886-9044

上海恩益禧电子国际贸易有限公司
中国上海市浦东新区银城中路200号
中银大厦2511-2512室
电话: (+86) 21-5888-5400
传真: (+86) 21-5888-5230