

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

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

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

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

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

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

## Notice

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

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

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



用户手册

# V850ES/HF2

32 位单片机

硬件

---

**$\mu$ PD70F3702**

**$\mu$ PD70F3703**

**$\mu$ PD70F3704**

文档编号: U17719CA2V0UD00 (第二版)  
发布日期: 2008 年 1 月 N CP(K)

© NEC Electronics Corporation 2005  
日本出版

[备忘录]



## CMOS 设备注意事项

### ① 输入引脚处的电压适用波形

因输入噪声或反射波产生的波形失真可能造成故障。如果由于噪声等影响，CMOS 设备的输入停留在  $V_{IL}$ （最大）和  $V_{IH}$ （最小）之间，该设备可能产生故障。当输入电平固定时，以及在输入电平在  $V_{IL}$ （最大）和  $V_{IH}$ （最小）之间过渡阶段时，切记避免波动噪声进入设备。

### ② 未使用的 CMOS 输入引脚的处理

CMOS 设备输入端未连接可能会造成故障。如果输入引脚未连接，则由于噪声等原因可能会产生内部输入电平，从而导致故障。CMOS 设备的性能与双极型或 NMOS 型设备不同。必须借助上拉或下拉电路使 CMOS 设备的输入电平固定在高电平或低电平。如果每个未用的引脚可作为输出引脚的话，那么应将它们通过电阻与  $V_{DD}$  或  $GND$  连接。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说明。

### ③ 静电放电防护措施

如果 MOS 设备周围有强电场，将会击穿栅极氧化层，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，而且一旦有静电产生，必须立即将其释放。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。建立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在抗静电容器、静电屏蔽袋或导电材料中进行。所有的测试和测量工具，包括工作台和地板必须良好接地。操作员应当佩戴放电腕带以保证良好接地，不能用手直接解除半导体设备。对于装配有半导体设备的 PW 版，也应采取类似的静电防范措施。

### ④ 初始化前的状态

上电未必能确定 MOS 设备的初始状态。在刚打开电源后，具有复位功能的 MOS 设备并没有被初始化，因此，上电不能保证输出引脚的电平，I/O 设置和寄存器的内容。设备在收到复位信号后才能进行初始化。具有复位功能的设备在上电后必须立即进行复位操作。

### ⑤ 上电/断电顺序

当设备的内部操作和外部接口使用不同的电源时，一般来说，首先应接通内部电源，然后再接通外部电源。在切断电源时，通常先切断外部电源，然后切断内部电源。使用颠倒的上电/断电顺序会造成设备内部元件过压，从而由于异常电流的通过造成内部元件故障或性能降低。

正确的上电/断电顺序因设备而异，必须根据相关的设备规格说明单独判断。

### ⑥ 断电状态下的信号输入

切勿在设备未加电时输入信号或 I/O 上拉电源。由输入这样的信号或 I/O 上拉电源造成的电流引入会造成故障或异常电流通过设备，从而造成内部元件性能下降。在断电状态下的信号输入因设备而异，必须根据相关的设备规格说明单独判断。

**IECUBE** 是 NEC Electronics Corporation 在日本和德国的注册商标。

**MINICUBE** 是 NEC Electronics Corporation 在日本和德国的注册商标，或者在美国的商标。

**Applilet** 是 NEC Electronics 在日本，德国，香港，中国，韩国，英国和美国的注册商标。

**Windows**和**Windows NT**是Microsoft Corporation在美国和/或其他国家的注册商标或商标。

**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 The Real-Time Operating system Nucleus的缩写。

**ITRON** 是 Industrial TRON 的缩写。

- 本文档信息于 2006 年9 月开始使用。这些信息可能会更改，恕不另行通知。如果用户要进行实际的设计，请参阅最新出版的日电电子数据表或数据手册等，以获取日电电子产品的最新规格说明。并非在每个国家都能购买到所有的产品或型号。要了解产品的存货情况和其他相关信息，请联系日电电子销售代表。
- 未经日电电子的书面许可，不得以任何形式、任何手段复制本文档的任何部分。本文档出现的任何错误，日电电子不承担任何责任。
- 用户在使用本文档列出的日电电子产品时，产生的对第三方专利、版权以及其他知识产权的侵犯，或者由于使用这些产品产生的其他责任，日电电子的专利、版权或其他指示产权等不授权任何许可，无论是明示的或其他形式的许可。
- 文档中电路、软件和其他相关信息的说明，仅用于半导体产品操作和应用示例的说明之目的。客户在使用这些电路、软件和信息设计自己的设备时自己应承担全部责任。客户或第三方在使用这些电路、软件和信息时造成的损失，日电电子不承担任何责任。
- 尽管日电电子尽力提高日电电子产品质量、可靠性和安全性，但客户应该理解错误是不可能完全避免的。为了尽可能减少日电电子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、防火和抗失效等。
- 日电电子产品分为以下三个质量等级：“标准”、“专业”、“特级”

“特级”质量等级只适用基于客户指定的特定应用的“质量保证计划”开发的于日电电子产品。日电电子产品的推荐用途取决于下面介绍的产品的质量等级。客户在某个应用中使用日电电子产品时，必须核对其质量等级。

“标准”：计算机、办公设备、通信设备、测试测量设备、音频视频设备、家用电器、机床、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、防犯罪系统、安全设备和医疗设备（非专门用于生命救护的设备）。

“特级”：飞机、航天设备、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。

除非在日电电子数据表或数据手册中明确规定，一般的日电电子产品质量等级都是“标准”。如果客户希望在非日电电子要求的应用环境中使用日电电子产品，必须事先与日电电子销售代表联系，以确定日电电子是否支持该应用环境。

注：

- (1) “日电电子”在这里是指“日本电气电子公司”及其用于多数股权的子公司。
- (2) “日电电子产品”是指由日电电子或为日电电子（定义如上）开发或制造的产品。

# 前言

**读者对象** 本手册适用于那些希望了解 V850ES/HF2 的功能，并使用这些产品设计应用系统的用户。

**目的** 本手册用于帮助用户了解下面**组件**中介绍的 V850ES/HF2 的硬件功能。

**组件** 本手册分为两部分：硬件（本手册）和 架构（**V850ES 架构用户手册**）。

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

**手册使用方法** 在阅读本手册前，读者应掌握电气工程、逻辑电路和微处理器的一般知识。

要了解 V850ES/HF2 的全部功能  
→请阅读本手册**目录**。

要了解已知名称的寄存器的详细情况  
→请阅读**附录 B 寄存器索引**。

要了解指令功能的详细情况  
→请参考单本的 **V850ES 架构用户手册**

了解 V850ES/HF2 电气规范  
→ 请参阅《**V850ES 架构用户手册**》，该手册可单独购买。

在本手册中，“xxx 寄存器的 yyy 位”被描述为 “xxx.yyy 位”。请小心注意，如果在一个程序中描述“xxx.yyy”，则编译器/汇编器则不能正确识别它。

<R>标记显示重要的修订内容。将“<R>”复制到 PDF 文件中“查找”字段中，就很容易搜索到修订的内容。

约定

数据有效位: 高位在左，低位在右  
有效低电平表示:  $\overline{\text{xxx}}$  (在引脚或信号名称上有上划线)  
存储器映射地址: 高位地址在顶部，低位地址在底部

注: 正文中用**注**标记的脚注  
注意事项: 需要特别关注的信息  
备注: 补充信息  
数值表示: 二进制 ... xxxx 或 xxxxB  
十进制 ... xxxx  
十六进制 ... xxxxH

指示 2 的幂的前缀  
(地址空间、存储器  
容量):

K (千):  $2^{10} = 1024$   
M (兆):  $2^{20} = 1024^2$   
G (千兆):  $2^{30} = 1024^3$

## 相关文档

本手册中的相关文档可能包括前期版本。不过，前期版本不是这样标记的。

### V850ES/HF2 相关文档

文档名称	文档编号
V850ES 架构用户手册	U15943E
V850ES/HF2 硬件用户手册	本手册

### 开发工具相关文档

文档名称		文档编号
QB-V850MINI （片上调试仿真器）		U17638E
QB-MINI2 具有编程功能的片上调试仿真器		U18371E
CA850 3.00 版 C 编译包	操作	U17293E
	C 语言	U17291E
	汇编语言	U17292E
	连接指令	U17294E
PM+ 6.20 版项目管理器		U17990E
ID850QB 3.20 版集成调试器	操作	U17964E
SM850 2.50 版系统仿真器	操作	U16218E
SM850 2.00 或更高版本系统仿真器	外部零件用户开放接口规范	U14873E
SM+ 系统仿真器	操作	U17246E
	用户开放接口	U17247E
RX850 3.20 版实时操作系统	基础	U13430E
	安装	U17419E
	技术	U13431E
	任务调试器	U17420E
RX850 Pro 3.20 版实时操作系统	基础	U13773E
	安装	U17421E
	技术	U13772E
	任务调试器	U17422E
AZ850 3.30 版系统性能分析器		U17423E
PG-FP4 闪存编程器		U15260E

# 目 录

第一章 引言 .....	17
1.1 概述 .....	17
1.2 功能 .....	19
1.3 应用领域 .....	19
1.4 订购说明 .....	20
1.5 引脚配置(俯视图) .....	21
1.6 功能框图 .....	23
1.6.1 内部框图 .....	23
1.6.2 内部功能单元 .....	24
第二章 引脚功能 .....	25
2.1 引脚功能列表 .....	25
2.2 引脚功能描述 .....	32
2.3 引脚 I/O 电路类型及未用引脚建议连接方式 .....	38
2.4 引脚 I/O 电路 .....	40
2.5 注意事项 .....	40
第三章 CPU 功能 .....	41
3.1 功能 .....	41
3.2 CPU 寄存器组 .....	42
3.2.1 程序寄存器组 .....	43
3.2.2 系统寄存器组 .....	44
3.3 操作模式 .....	50
3.3.1 操作模式的指定 .....	50
3.4 地址空间 .....	51
3.4.1 CPU 地址空间 .....	51
3.4.2 CPU 地址空间的环绕 .....	52
3.4.3 存储空间映射图 .....	53
3.4.4 存储区域 .....	55
3.4.5 地址空间建议使用的方法 .....	58
3.4.6 外围 I/O 寄存器 .....	61
3.4.7 特殊功能寄存器 .....	68
3.4.8 注意事项 .....	72
第四章 端口功能 .....	75
4.1 功能 .....	75
4.2 基本端口配置 .....	75
4.3 端口功能 .....	78
4.3.1 端口功能操作 .....	78
4.3.2 设置端口引脚的注意事项 .....	79
4.3.3 端口 0 .....	80
4.3.4 端口 3 .....	84
4.3.5 端口 4 .....	89

4.3.6	端口 5	92
4.3.7	端口 7	98
4.3.8	端口 9	100
4.3.9	端口 CM	108
4.3.10	端口 CS	110
4.3.11	端口 CT	112
4.3.12	端口 DL	114
4.3.13	另外用作片上调试功能的端口引脚	116
4.3.14	要将端口引脚用作复用-功能引脚的寄存器设置	117
4.4	端口框图	120
4.5	注意事项	145
4.5.1	设置端口引脚的注意事项	145
第五章	时钟发生功能	146
5.1	概述	146
5.2	配置	147
5.3	寄存器	149
5.4	运行	154
5.4.1	各时钟的工作	154
5.4.2	时钟输出功能	154
5.5	PLL 功能	155
5.5.1	概述	155
5.5.2	寄存器	155
5.5.3	用法	159
第六章	16 位定时器/事件计数器 P (TMP)	160
6.1	概述	160
6.2	功能	160
6.3	配置	161
6.4	寄存器	163
6.5	操作	177
6.5.1	时间间隔定时器模式 (TPnMD2 至 TPnMD0 位=000)	178
6.5.2	外部事件计数模式 (TPnMD2 至 TPnMD0 位=001)	188
6.5.3	外部触发脉冲输出模式 (TPnMD2 至 TPnMD0 位=010)	196
6.5.4	单次脉冲输出模式 (TPnMD2 至 TPnMD0 位=011)	208
6.5.5	PWM 输出模式 (TPnMD2 至 TPnMD0 位=100)	215
6.5.6	独立定时器模式 (TPnMD2 至 TPnMD0 位=101)	224
6.5.7	脉冲宽度测量模式 (TPnMD2 至 TPnMD0 位=110)	241
6.5.8	定时器输出操作	247
6.6	定时器调谐工作功能	248
6.7	选择器功能	252
6.8	注意事项	254
第七章	16-位定时器/事件计数器 Q (TMQ)	255
7.1	概述	255
7.2	功能	255
7.3	配置	256



7.4	寄存器.....	259
7.5	操作 .....	277
7.5.1	计时器模式(TQ0MD2 到 TQ0MD0 位 = 000).....	278
7.5.2	外部事件计数模式(TQ0MD2 到 TQ0MD0 位 = 001) .....	287
7.5.3	外部触发脉冲输出模式(TQ0MD2 到 TQ0MD0 位 = 010) .....	296
7.5.4	单触发脉冲输出模式(TQ0MD2 到 TQ0MD0 位 = 011).....	309
7.5.5	PWM 输出模式(TQ0MD2 到 TQ0MD0 位 = 100) .....	318
7.5.6	自由定时器模式(TQ0MD2 到 TQ0MD0 位 = 101) .....	329
7.5.7	脉宽测量模式(TQ0MD2 到 TQ0MD0 位 = 110).....	349
7.5.8	PWM 模式(TQ0MD2 到 TQ0MD0 = 111) 的三角波 .....	355
7.5.9	定时器输出操作 .....	356
7.6	定时器调试操作功能 .....	357
7.7	注意事项 .....	361
第八章	16-位间隔定时器 M (TMM) .....	362
8.1	概述 .....	362
8.2	配置 .....	363
8.3	寄存器.....	
8.4	运行 .....	365
8.4.1	计时器模式.....	365
8.4.2	注意事项.....	
第九章	钟表定时器功能 .....	370
9.1	功能 .....	370
9.2	配置 .....	371
9.3	寄存器.....	373
9.4	操作 .....	377
9.4.1	作为钟表定时器操作.....	377
9.4.2	作为计时器的操作 .....	378
9.4.3	注意事项.....	379
第十章	看门狗定时器 2 的功能 .....	
10.1	功能.....	380
10.2	配置.....	381
10.3	寄存器.....	382
10.4	操作.....	385
第十一章	A/D 转换器.....	386
11.1	概述.....	386
11.2	功能.....	386
11.3	配置.....	387
11.4	寄存器.....	390
11.5	操作.....	398
11.5.1	基本操作.....	398
11.5.2	触发模式.....	399
11.5.3	操作模式.....	401
11.5.4	掉电比较模式.....	405
11.6	注意事项 .....	410

11.7 如何读取 A/D 转换器特性表 .....	414
第十二章 异步串行接口 (UARTA).....	418
12.1 特性 .....	418
12.2 配置 .....	419
12.3 寄存器 .....	421
12.4 中断请求信号 .....	427
12.5 操作 .....	428
12.5.1 数据格式 .....	428
12.5.2 SBF 传输/接收格式 .....	430
12.5.3 SBF 传输 .....	432
12.5.4 SBF 接收 .....	3
12.5.5 UART 传输 .....	434
12.5.6 连续传输程序 .....	435
12.5.7 UART 接收 .....	437
12.5.8 接收误差 .....	438
12.5.9 奇偶校验类型和操作 .....	440
12.5.10 接收数据噪音滤波器 .....	441
12.6 专用波特率发生器 .....	442
12.7 注意事项 .....	450
第十三章 3 线可变数据长度串行 I/O 接口 (CSIB).....	451
13.1 特性 .....	451
13.2 结构 .....	452
13.3 寄存器 .....	454
13.4 中断请求信号 .....	461
13.5 操作 .....	
13.5.1 单向传输模式 (主模式, 发送模式) .....	462
13.5.2 单向传输模式 (主模式, 接收模式) .....	464
13.5.3 单向传输模式 (主模式, 发送/接收模式) .....	466
13.5.4 单向传输模式 (从模式, 发送模式) .....	468
13.5.5 单向传输模式 (从模式, 接收模式) .....	470
13.5.6 单向传输模式 (从模式, 发送/接收模式) .....	472
13.5.7 连续传输模式 (主模式, 发送模式) .....	474
13.5.8 连续传输模式 (主模式, 接收模式) .....	476
13.5.9 连续传输模式 (主模式, 发送/接收模式) .....	479
13.5.10 连续传输模式 (从模式, 发送模式) .....	483
13.5.11 连续传输模式 (从模式, 接收模式) .....	485
13.5.12 连续传输模式 (从模式, 发送/接收模式) .....	488
13.5.13 接收错误 .....	492
13.5.14 时钟时序 .....	493
13.6 操作禁用时, 输出引脚的状态 .....	495
13.7 波特率发生器 .....	496
13.7.1 波特率的产生 .....	497
13.8 注意事项 .....	498
第十四章 中断/异常处理功能 .....	499

14.1	特性 .....	499
14.2	不可屏蔽中断 .....	502
14.2.1	操作过程 .....	504
14.2.2	恢复 .....	505
14.2.3	NP 标志 .....	506
14.3	可屏蔽中断 .....	507
14.3.1	操作过程 .....	507
14.3.2	恢复 .....	509
14.3.3	可屏蔽中断的优先权 .....	510
14.3.4	中断控制寄存器 (xxICn) .....	514
14.3.5	中断屏蔽寄存器 0 至 2 (IMR0 至 IMR2) .....	516
14.3.6	服务中优先级寄存器 (ISPR) .....	517
14.3.7	ID 标志 .....	518
14.3.8	看门狗定时器模式寄存器 2 (WDTM2) .....	518
14.4	软件异常 .....	519
14.4.1	操作过程 .....	519
14.4.2	恢复 .....	520
14.4.3	EP 标志 .....	521
14.5	异常陷阱 .....	522
14.5.1	非法操作码定义 .....	522
14.5.2	调试陷阱 .....	524
14.6	外部中断请求输入引脚 (NMI 以及 INTP0 至 INTP7) .....	526
14.6.1	噪音消除 .....	526
14.6.2	边沿检测 .....	526
14.7	CPU 的中断响应时间 .....	532
14.8	CPU 不响应中断的时间阶段 .....	533
14.9	注意事项 .....	533
第十五章	按键中断功能 .....	534
15.1	功能 .....	534
15.2	寄存器 .....	535
15.3	注意事项 .....	535
第十六章	待机功能 .....	536
16.1	概述 .....	536
16.2	寄存器 .....	538
16.3	HALT 模式 .....	541
16.3.1	设置和工作状态 .....	541
16.3.2	解除 HALT 模式 .....	541
16.4	IDLE1 模式 .....	543
16.4.1	设置和工作状态 .....	543
16.4.2	解除 IDLE1 模式 .....	543
16.5	IDLE2 模式 .....	545
16.5.1	设置及工作状态 .....	545
16.5.2	解除 IDLE2 模式 .....	545
16.5.3	在解除 IDLE2 模式时保证设立时间 .....	547

16.6	STOP 模式.....	548
16.6.1	设置及工作状态.....	548
16.6.2	解除 STOP 模式.....	548
16.6.3	在解除 STOP 模式时保证振荡稳定时间.....	550
16.7	副时钟工作模式.....	551
16.7.1	设置及工作状态.....	551
16.7.2	解除副时钟工作模式.....	551
16.8	副 IDLE 模式.....	553
16.8.1	设置及工作状态.....	553
16.8.2	解除副 IDLE 模式.....	553
第十七章	复位功能.....	556
17.1	概述.....	556
17.2	用于检验复位源的寄存器.....	557
17.3	操作.....	558
17.3.1	通过 RESET 引脚进行复位操作.....	558
17.3.2	通过看门狗定时器 2 进行复位操作.....	560
17.3.3	通过上电清零电路进行复位操作.....	561
17.3.4	通过低压检测器进行复位操作.....	561
17.3.5	通过时钟监视器进行复位操作.....	561
17.4	复位解除后的操作.....	562
第十八章	时钟监视器.....	564
18.1	功能.....	564
18.2	配置.....	564
18.3	寄存器.....	565
18.4	操作.....	566
第十九章	上电清零电路.....	569
19.1	功能.....	569
19.2	配置.....	569
19.3	操作.....	570
第二十章	低压检测器.....	571
20.1	功能.....	571
20.2	配置.....	571
20.3	寄存器.....	572
20.4	操作.....	574
20.4.1	要用于内部复位信号时.....	574
20.4.2	要用于中断时.....	576
20.5	RAM 保持电压检测操作.....	577
20.6	仿真功能.....	578
第二十一章	稳压器.....	579
21.1	概述.....	579
21.2	运行.....	580
第二十二章	闪存.....	581
22.1	功能.....	581
22.2	存储器配置.....	582

22.3	功能概述 .....	583
22.4	由专用闪存编程器重写 .....	586
22.4.1	编程环境 .....	586
22.4.2	通信模式 .....	587
22.4.3	闪存控制 .....	592
22.4.4	选择通信模式 .....	593
22.4.5	通信命令 .....	594
22.4.6	引脚连接 .....	595
22.5	通过自编程重写 .....	599
22.5.1	概述 .....	599
22.5.2	功能 .....	600
22.5.3	标准自编程流程 .....	601
22.5.4	闪存功能 .....	602
22.5.5	引脚处理 .....	602
22.5.6	使用的内部资源 .....	603
第二十三章	选项字节功能 .....	604
第二十四章	片上调试功能 .....	606
24.1	使用 DCU 调试 .....	607
24.1.1	连接电路示例 .....	607
24.1.2	接口信号 .....	607
24.1.3	可屏蔽功能 .....	609
24.1.4	寄存器 .....	609
24.1.5	操作 .....	611
24.1.6	注意事项 .....	612
24.2	不使用 DCU 调试 .....	613
24.2.1	电路连接示例 .....	613
24.2.2	可屏蔽功能 .....	614
24.2.3	用户资源的安全 .....	615
24.2.4	注意事项 .....	621
24.3	ROM 安全功能 .....	622
24.3.1	安全 ID .....	622
24.3.2	设置 .....	623
第二十五章	电气特性 .....	625
25.1	最大绝对级别 .....	625
25.2	电容 .....	627
25.3	运行条件 .....	627
25.4	振荡器特性 .....	628
25.4.1	主时钟振荡器特性 .....	628
25.4.2	副时钟振荡器特性 .....	629
25.4.3	PLL 特性 .....	630
25.4.4	内部振荡器特性 .....	630
25.5	电压调节器特性 .....	630
25.6	DC 特性 .....	631
25.6.1	I/O 电平 .....	631

25.6.2 引脚漏电流.....	632
25.6.3 电源电流.....	633
<b>25.7 数据保持特性 .....</b>	<b>634</b>
<b>25.8 AC 特性.....</b>	<b>635</b>
25.8.1 CLKOUT 输出时序.....	636
<b>25.9 基本操作 .....</b>	<b>637</b>
<b>25.10 闪存编程特性 .....</b>	<b>644</b>
<b>第二十六章 封装图.....</b>	<b>645</b>
<b>第二十七章 推荐焊接条件.....</b>	<b>646</b>
<b>附录 A 开发工具.....</b>	<b>647</b>
<b>A.1 软件包 .....</b>	<b>649</b>
<b>A.2 语言处理软件.....</b>	<b>649</b>
<b>A.3 控制软件.....</b>	<b>649</b>
<b>A.4 调试工具 (硬件) .....</b>	<b>650</b>
A.4.1 当使用在线仿真器 IECUBE QB-V850ESFX2 时 .....	650
A.4.2 当使用 MINICUBE QB-V850MINI 时.....	652
A.4.3 当时用 MINICUBE2 QB-MINI2 时 .....	653
<b>A.5 调试工具 (软件) .....</b>	<b>654</b>
<b>A.6 嵌入式软件 .....</b>	<b>655</b>
<b>A.7 闪存写入工具.....</b>	<b>656</b>
<b>附录 B 寄存器索引.....</b>	<b>657</b>
<b>附录 C 指令集列表.....</b>	<b>664</b>
<b>C.1 规则 .....</b>	<b>664</b>
<b>C.2 指令集(按字母先后顺序).....</b>	<b>667</b>
<b>附录 D 注意事项列表 .....</b>	<b>674</b>
<b>附录 E 修订历史.....</b>	<b>703</b>
<b>E.1 本版本中主要修订之处 .....</b>	<b>703</b>

## 第一章 引言

V850ES/HF2 日电电子 V850 系列单片机中的一种，用于实时控制应用中的低功耗运行。

### 1.1 概述

V850ES/HF2 是一款 32 位的单片机，片内集成了 V850ES CPU 内核以及相关外围功能器件，如 ROM/RAM，定时器/计数器，串行接口和一个 A/D 转换器。

除了具有高实时响应的特点及单时钟周期基本指令外，V850ES/HF2 还具有乘法指令，饱和运算指令，位操作指令等，这些指令是由一硬件乘法器来实现，是数字伺服控制应用最佳指令。

表 1-1 列出了 V850ES/HF2 系列产品。

表 1-1. V850ES/HF2 产品列表

产品型号		$\mu$ PD70F3702	$\mu$ PD70F3703	$\mu$ PD70F3704
内存	闪存	64 KB	128 KB	256 KB
	RAM	12 KB		
存储器空间	逻辑空间	64 MB		
通用寄存器		32 位 $\times$ 32 个寄存器		
主时钟 (振荡频率)		陶瓷振荡器/晶振/外部时钟 • 在 PLL 模式下操作时: $f_x = 4$ 到 5 MHz • 在时钟直通模式下操作时: $f_x = 4$ 到 5 MHz		
副时钟(振荡频率)		陶瓷振荡器/外部时钟: $f_{XT} = 32.768$ kHz RC 振荡器: 20 kHz		
内部振荡器		$f_R = 200$ kHz (TYP.)		
最小指令执行时间		50 ns (在主时钟 ( $f_{\text{ox}}$ ) = 20 MHz 操作时)		
DSP 功能		32 $\times$ 32 = 64: 200 到 250 ns (20 MHz 操作频率下) 32 $\times$ 32 + 32 = 32: 300 ns (20 MHz 操作频率下) 16 $\times$ 16 = 32: 50 到 100 ns (20 MHz 操作频率下) 16 $\times$ 16 + 32 = 32: 150 ns (20 MHz 操作频率下)		
I/O 口线		I/O: 67		
定时器		16 位定时器/事件计数器 P: 4 通道 16 位定时器/事件计数器 Q: 1 通道 16 位间隔定时器 M: 1 通道 看门狗定时器 2: 1 通道 钟表定时器: 1 通道		
A/D 转换器		10 位分辨率 $\times$ 12 通道		
串行接口		CSIB: 2 通道 UARTA (对于 LIN): 2 通道		
中断源		外部中断源: 9 (9) <sup>注</sup> , 内部中断源 I: 32		
省电功能		HALT/IDLE1/IDLE2/STOP/副时钟/副 IDLE 模式		
复位		RESET 引脚输入信号, 看门狗定时器 2 (WDT2), 时钟监视器 (CLM), POC 电路, 低电压检测器 (LVI)		
DCU		提供 (RUN/中断模式 下)		
工作电压		3.5 到 5.5 V (A/D 转换器: 4.0 到 5.5 V)		
工作环境温度		-40 到 +85°C		
封装形式		80 引脚塑封 TQFP (密脚距) (12 $\times$ 12 mm)		

注 括号内的数据指的是能够释放 STOP 模式的外部中断源数目。

## 1.2 功能

- 最小指令执行时间: 50 ns (在主系统时钟频率( $f_{xx}$ )为 20 MHz 下操作时)
- 通用寄存器: 32 位  $\times$  32 个寄存器
- CPU 功能: 有符号数乘法运算( $16 \times 16 \rightarrow 32$ ): 1 到 2 个时钟周期  
有符号数乘法运算( $32 \times 32 \rightarrow 64$ ): 1 到 5 个时钟周期  
饱和运算(包括上溢及下溢检测功能)  
32 位移位指令: 1 个时钟周期  
位操作指令  
长/短格式数据装载/存贮指令
- 存储空间: 64 MB 线性地址空间 (用于存储程序和数据)
  - 内部存储器:
    - RAM: 12 KB
    - 闪存: 64 KB/128 KB/256 KB (参见表 1-1)
- 中断和异常:
  - 不可屏蔽中断: 2 个中断源
  - 可屏蔽中断: 39 个中断源
  - 软件异常中断: 32 个中断源
  - 异常陷阱: 2 个中断源
- I/O 口线: I/O 端口数: 67
- 定时器功能:
  - 16 位间隔定时器 M (TMM): 1 通道
  - 16 位定时器/事件计数器 P (TMP): 4 通道
  - 16 位定时器/事件计数器 Q (TMQ): 1 通道
  - 钟表定时器: 1 通道
  - 看门狗定时器 2: 1 通道
- 串行接口: 异步串行接口 A (UARTA)  
3 线数据帧长度可变串行接口 B (CSIB)
  - UARTA (支持 LIN): 2 通道
  - CSIB: 2 通道
- A/D 转换器: 10 位分辨率: 12 通道
- DCU (调试控制单元): JTAG 接口
- 时钟发生器: 在主系统时钟或子系统时钟运行期间  
7 级 CPU 时钟( $f_{xx}$ ,  $f_{xx}/2$ ,  $f_{xx}/4$ ,  $f_{xx}/8$ ,  $f_{xx}/16$ ,  $f_{xx}/32$ ,  $f_{xt}$ )  
Clock-through 模式/PLL 模式可选
- 内部振荡时钟: 200 kHz (TYP.)
- 省电功能: HALT/IDLE1/IDLE2/STOP/副时钟/子 IDLE 模式
- 封装形式: 80 引脚塑封 TQFP (密脚距) ( $12 \times 12$ )

## 1.3 应用领域

消费设备



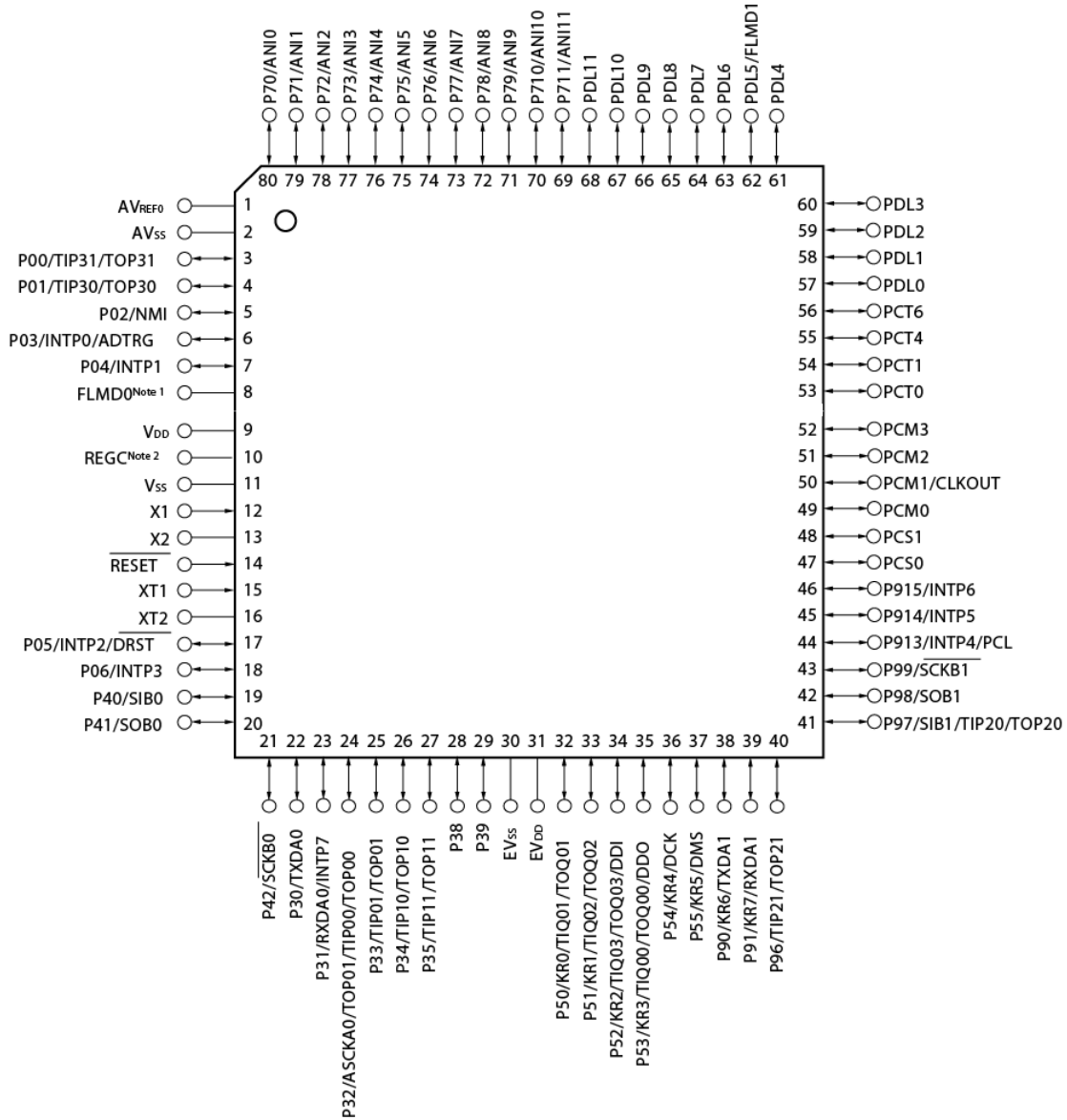
## 1.4 订购说明

产品型号	封装形式	片上闪存
$\mu$ PD70F3702GK-9EU-A	80 引脚塑料 TQFP 封装(密脚距) (12 × 12)	64 KB
$\mu$ PD70F3703GK-9EU-A	80 引脚塑料 TQFP 封装(密脚距) (12 × 12)	128 KB
$\mu$ PD70F3704GK-9EU-A	80 引脚塑料 TQFP 封装(密脚距) (12 × 12)	256 KB

**备注** 产品型号以-A 结尾产品为无铅产品。

## 1.5 引脚配置(俯视图)

80 引脚塑料 TQFP 封装(密脚距) (12 × 12)

 $\mu$ PD70F3702GK-9EU-A $\mu$ PD70F3704GK-9EU-A $\mu$ PD70F3703GK-9EU-A

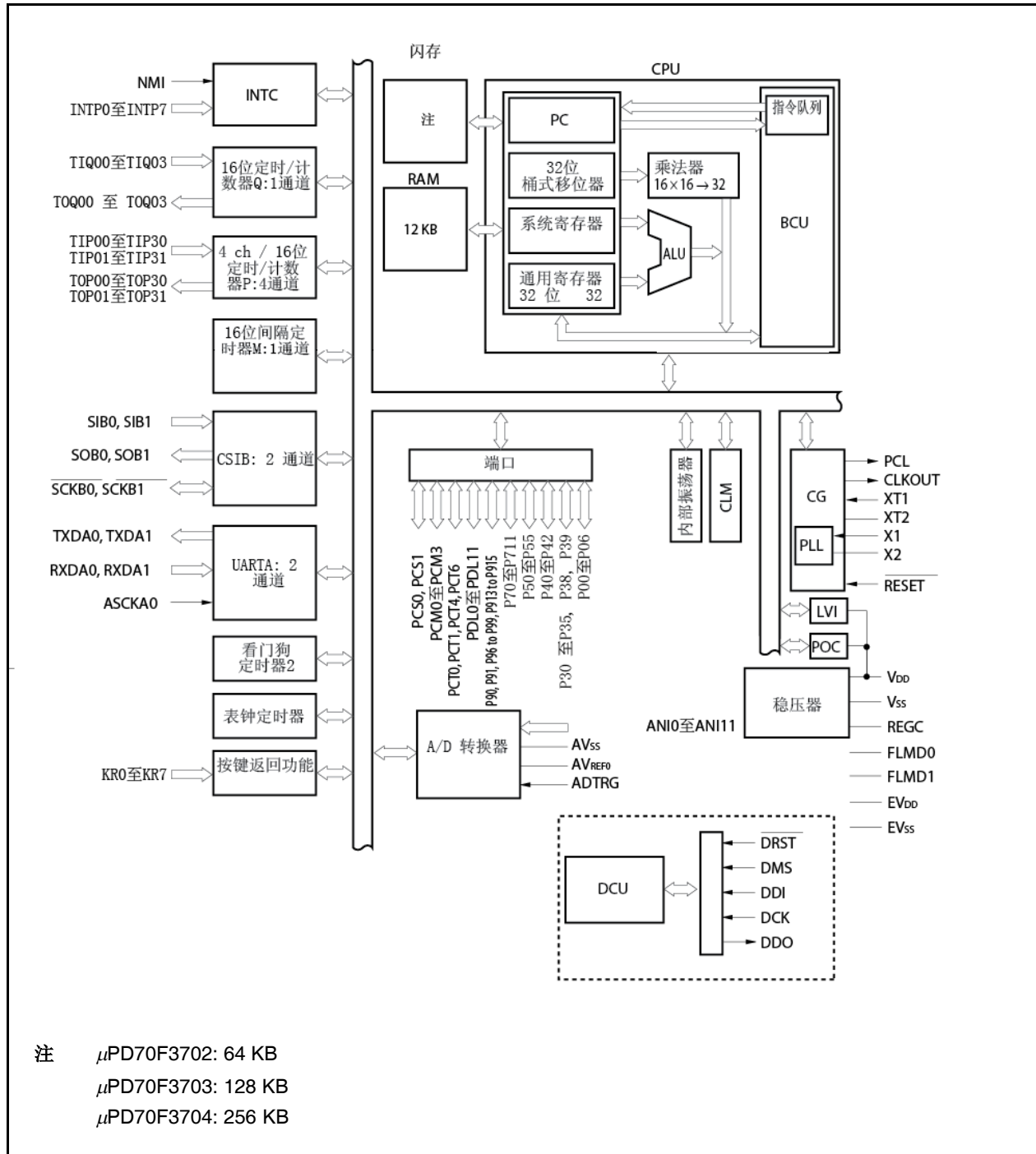
- 注
1. 正常模式下, 将该引脚连接到 VSS 引脚端。
  2. 通过一个 4.7  $\mu$ F(推荐值)电容将 REGC 引脚和 VSS 引脚连接在一起。

## 引脚定义

ADTRG:	A/D 触发器输入端	PCL:	可编程时钟输出端
ANI0 到 ANI11:	模拟输入端	PCM0 到 PCM3:	端口 CM
ASCKA0:	异步串行时钟	PCS0, PCS1:	端口 CS
AV <sub>REF0</sub> :	模拟参考电压	PCT0, PCT1,	
AV <sub>SS</sub> :	模拟电源 V <sub>SS</sub>	PCT4, PCT6:	端口 CT
CLKOUT:	时钟输出端	PDL0 到 PDL11:	端口 DL
DCK:	调试时钟	REGC:	稳压器控制
DDI:	调试数据输入端	RESET:	复位
DDO:	调试数据输出端	RXDA0, RXDA1:	接收数据
DMS:	调试模式选择端	SCKB0, SCKB1:	串行时钟
DRST:	调试复位端	SIB0, SIB1:	串行输入
EV <sub>DD</sub> :	端口电源	SOB0, SOB1:	串行输出
EV <sub>SS</sub> :	端口接地端	TIP00, TIP01,	
FLMD0, FLMD1:	闪存编程模式	TIP10, TIP11,	
INTP0 到 INTP7:	外部中断输入	TIP20, TIP21,	
KR0 到 KR7:	按键返回	TIP30, TIP31,	
NMI:	不可屏蔽中断请求	TIQ00 到 TIQ03:	定时器输入端
P00 到 P06:	端口 0	TOP00, TOP01,	
P30 到 P35,		TOP10, TOP11,	
P38, P39:	端口 3	TOP20, TOP21,	
P40 到 P42:	端口 4	TOP30, TOP31,	
P50 到 P55:	端口 5	TOQ00 to TOQ03:	定时器输出端
P70 到 P711:	端口 7	TXDA0, TXDA1:	发送数据端
P90, P91,		V <sub>DD</sub> :	电源端
P96 到 P99,		V <sub>SS</sub> :	接地端
P913 到 P915:	端口 9	X1, X2:	主系统时钟晶振
		XT1, XT2:	副系统时钟晶振

## 1.6 功能框图

## 1.6.1 内部框图



## 1.6.2 内部功能单元

### (1) CPU

该 CPU 使用五级管线化控制，确保了地址计算，算术逻辑运算，数据传送以及几乎所有其他指令的处理都可以在一个时钟周期内完成。

另外，片上其他一些专用硬件，比如乘法器(16 位 × 16 位 → 32 位)和桶式移位器(32 位)等，也提供了更快、更复杂的数据处理能力。

### (2) 总线控制单元 (BCU)

总线控制单元 BCU 执行内部总线控制任务。

### (3) ROM

该 ROM 区为一 256 KB/128 KB/64 KB 的闪存，对应的映射地址为 0000000H 到 003FFFFH /0000000H 到 001FFFFH /0000000H 到 000FFFF。在取指其间，CPU 可以在一个时钟内现实对该存储区的访问。

### (4) RAM

这是一 12 KB RAM，对应的映射地址为 3FFC000H 到 3FFEFFFH。在数据存取其间，CPU 可以在一个时钟内现实对该 RAM 区的访问。

### (5) 中断控制器(INTC)

该中断控制器用以处理来自片上外围硬件设备及外部硬件设备的中断请求(NMI，INTP0 到 INTP7)，并可以对这些中断请求指定 8 级中断优先权，而且可以执行多重服务控制。

### (6) 时钟发生器(CG)

配备有主系统时钟振荡器和副系统时钟振荡器，分别用于产生主系统时钟振荡频率( $f_x$ )和副系统时钟振荡频率( $f_{xt}$ )。有两种模式：在 clock-through 模式下， $f_x$  就用作主系统时钟频率( $f_{xx}$ )，而 PLL 在模式下， $f_x$  乘以 4 以后用作主系统时钟频率。

CPU 时钟频率( $f_{CPU}$ )可在  $f_{xx}$ ， $f_{xx}/2$ ， $f_{xx}/4$ ， $f_{xx}/8$ ， $f_{xx}/16$ ， $f_{xx}/32$  及  $f_{xt}$  这七种频率中选取。

### (7) 内部振荡器

配备一片上内部振荡器，振荡频率为 200 kHz (典型值)。该内部振荡器为看门狗定时器 2 及定时器 M 提供时钟源。

### (8) 定时器/计数器

配备片上 4 通道 16 位定时器/事件计数器 P (TMP)，1 通道 16 位定时器/事件计数器 Q (TMQ)，以及 1 通道 16 位间隔定时器 M (TMM)。

### (9) 钟表定时器

该定时器用于计数时钟(32.768 kHz 的副系统时钟或来自预分频率器 3 的  $f_{BRG}$  32.768 kHz 时钟)的参考时间周期(0.5 秒)的计数，也可用作主系统时钟的间隔定时器。

### (10) 看门狗定时器 2

配备一片上看门狗定时器，用以监测意外程序循环,系统异常等。

不论是内部振荡器时钟，还是主系统时钟都可以选作看门狗定时器 2 的时钟源。

在溢出产生后，看门狗定时器 2 会触发产生一个不可屏蔽中断请求信号(INTWDT2)或系统复位信号。

**(11) 串行接口**

V850ES/HF2 内部集成了三种类型串行接口：异步串行接口 A (UARTA)，3 线数据帧长度可变串行接口 B (CSIB)。

对于 UARTA，数据是通过 TXDA0, TXDA1, RXDA0, 以及 RXDA1 引脚进行传输的。

对于 CSIB，数据是通过 SOB0, SOB1, SIB0, SIB1, SCKB0, 以及 SCKBT 引脚进行传输的。

**(12) A/D 转换器**

这是一个 10 位，具有 12 个模拟输入引脚的 A/D 转换器，采用逐次逼近法进行模数转换。

**(13) 按键中断功能**

按键中断请求信号(INTKR)可通过在键输入（8 通道）引脚输入一个下降沿触发产生。

**(14) DCU (调试控制单元)**

配备片上调试功能，它是通过运用 JTAG(联合测试行动小组)通信说明来实现的。使用控制引脚输入电平和片上调试模式寄存器(OCDM)来实现通用端口功能与片上调试功能之间的切换。

**(15) 端口**

提供通用端口功能及控制引脚功能。要了解详细情况，请参阅**第四章 端口功能**。

## 第二章 引脚功能

本节说明 V850ES/HF2 的引脚名称及其功能。

### 2.1 引脚功能列表

提供两个 I/O 缓冲电源，AVREF0 和 EVDD。电源与引脚之间的对应关系如下所示。

表 2-1. 引脚 I/O 缓冲电源

电源	对应引脚
AVREF0	端口 7
EVDD	端口 0, 3 到 5, 9, CM, CS, CT, DL, RESET

#### (1) 端口引脚

表 2-2. 引脚列表 (端口引脚) (1/3)

引脚名称	引脚号	I/O	引脚功能	复用功能
P00	3	I/O	端口 0 7 位 I/O 端口 该端口各位输入/输出方式可以分别按位进行设定。	TIP31/TOP31
P01	4			TIP30/TOP30
P02	5			NMI
P03	6			INTP0/ADTRG
P04	7			INTP1
P05	17			INTP2/DRST
P06	18			INTP3
P30	22	I/O	端口 3 8 位 I/O 端口 该端口各位输入/输出方式可以分别按位进行设定。	TXDA0
P31	23			RXDA0/INTP7
P32	24			ASCKA0/TIP00/TOP00/TOP01
P33	25			TIP01/TOP01
P34	26			TIP10/TOP10
P35	27			TIP11/TOP11
P38	28			—
P39	29			—
P40	19	I/O	端口 4 3 位 I/O 端口 该端口各位输入/输出方式可以分别按位进行设定。	SIB0
P41	20			SOB0
P42	21			SCKB0

表 2-2. 引脚列表 (端口引脚) (2/3)

引脚名称	引脚号	I/O	引脚功能	复用功能
P50	32	I/O	端口 5 6 位 I/O 端口 该端口各位输入/输出方式可以分别按位进行设定。	KR0/TIQ01/TOQ01
P51	33			KR1/TIQ02/TOQ02
P52	34			KR2/TIQ03/TOQ03/DDI
P53	35			KR3/TIQ00/TOQ00/DDO
P54	36			KR4/DCK
P55	37			KR5/DMS
P70	80	I/O	端口 7 12 位 I/O 端口 该端口各位输入/输出方式可以分别按位进行设定。	ANI0
P71	79			ANI1
P72	78			ANI2
P73	77			ANI3
P74	76			ANI4
P75	75			ANI5
P76	74			ANI6
P77	73			ANI7
P78	72			ANI8
P79	71			ANI9
P710	70			ANI10
P711	69			ANI11
P90	38	I/O	端口 9 9 位 I/O 端口 该端口各位输入/输出方式可以分别按位进行设定。	KR6/TXDA1
P91	39			KR7/RXDA1
P96	40			TIP21/TOP21
P97	41			SIB1/TIP20/TOP20
P98	42			SOB1
P99	43			SCKB1
P913	44			INTP4/PCL
P914	45			INTP5
P915	46			INTP6
PCM0	49	I/O	端口 CM 4 位 I/O 端口 该端口各位输入/输出方式可以分别按位进行设定。	—
PCM1	50			CLKOUT
PCM2	51			—
PCM3	52			—
PCS0	47	I/O	端口 CS 2 位 I/O 端口 该端口各位输入/输出方式可以分别按位进行设定。	—
PCS1	48			—
PCT0	53	I/O	端口 CT 4 位 I/O 端口 该端口各位输入/输出方式可以分别按位进行设定。	—
PCT1	54			—
PCT4	55			—
PCT6	56			—



表 2-2. 引脚列表 (端口引脚) (3/3)

引脚名称	引脚号	I/O	引脚功能	复用功能
PDL0	57	I/O	端口 DL 12 位 I/O 端口 该端口各位输入/输出方式可以分别按位进行设定。	—
PDL1	58			—
PDL2	59			—
PDL3	60			—
PDL4	61			—
PDL5	62			FLMD1
PDL6	63			—
PDL7	64			—
PDL8	65			—
PDL9	66			—
PDL10	67			—
PDL11	68			—

## (2) 非端口引脚

表 2-3. 引脚列表 (非端口引脚) (1/3)

引脚名称	引脚号	I/O	引脚功能	复用功能
NMI <sup>※</sup>	5	输入	外部中断输入端。 (不可屏蔽, 并具有消除模拟噪声功能。)	P02
INTP0	6	输入	外部中断请求输入端。 (可屏蔽, 并具有消除模拟噪声功能。)	P03/ADTRG
INTP1	7			P04
INTP2	17			P05/ $\overline{\text{DRST}}$
INTP3	18			P06
INTP4	44			P913/PCL
INTP5	45			P914
INTP6	46			P915
INTP7	23			P31/RXDA0
TIP00	24	输入	外部事件/时钟信号输入端 (TMP00)	P32/ASCKA0/TOP00/TOP01
TIP01	25		外部事件输入端 (TMP01)	P33/TOP01
TIP10	26		外部事件/时钟信号输入端 (TMP10)	P34/TOP10
TIP11	27		外部事件输入端 (TMP11)	P35/TOP11
TIP20	41		外部事件/时钟信号输入端 (TMP20)	P97/SIB1/TOP20
TIP21	40		外部事件输入端 (TMP21)	P96/TOP21
TIP30	4		外部事件/时钟信号输入端 (TMP30)	P01/TOP30
TIP31	3		外部事件输入端 (TMP31)	P00/TOP31
TOP00	24	输出	定时器输出端 (TMP00)	P32/ASCKA0/TIP00/TOP01
TOP01	24		定时器输出端 (TMP01)	P32/ASCKA0/TIP00/TOP00
	25			P33/TIP01
TOP10	26		定时器输出端 (TMP10)	P34/TIP10
TOP11	27		定时器输出端 (TMP11)	P35/TIP11
TOP20	41		定时器输出端 (TMP20)	P97/SIB1/TIP20
TOP21	40		定时器输出端 (TMP21)	P96/TIP21
TOP30	4		定时器输出端 (TMP30)	P01/TIP30
TOP31	3		定时器输出端 (TMP31)	P00/TIP31
TIQ00	35	输入	外部事件/时钟信号输入端 (TMQ00)	P53/KR3/TOQ00/DDO
TIQ01	32		外部事件输入端 (TMQ01)	P50/KR0/TOQ01
TIQ02	33		外部事件输入端 (TMQ02)	P51/KR1/TOQ02
TIQ03	34		外部事件输入端 (TMQ03)	P52/KR2/TOQ03/DDI
TOQ00	35	输出	定时器输出端 (TMQ00)	P53/KR3/TIQ00/DDO
TOQ01	32		定时器输出端 (TMQ01)	P50/KR0/TIQ01
TOQ02	33		定时器输出端 (TMQ02)	P51/KR1/TIQ02
TOQ03	34		定时器输出端 (TMQ03)	P52/KR2/TIQ03/DDI

注 NMI 引脚另一复用功能是 P02 引脚。复位后该引脚为引脚功能, 要启用 NMI 引脚, 请将 PMC0.PMC02 位设置为 1。NMI 引脚的初始化设置为“无边沿检测”, NMI 引脚的有效边沿要使用 INTF0 和 INTR0 寄存器来设置。

表 2-3. 引脚列表 (非端口引脚) (2/3)

引脚名称	引脚号	I/O	引脚功能	复用功能
SIB0	19	输入	串行数据接收输入端 (CSIB0)	P40
SIB1	41		串行数据接收输入端 (CSIB1)	P97/TIP20/TOP20
SOB0	20	输出	串行发送数据输出端 (CSIB0)	P41
SOB1	42		串行发送数据输出端 (CSIB1)	P98
$\overline{\text{SCKB0}}$	21	I/O	串行时钟 I/O (CSIB0)	P42
$\overline{\text{SCKB1}}$	43		串行时钟 I/O (CSIB1)	P99
RXDA0	23	输入	串行数据接收输入端 (UARTA0)	P31/INTP7
RXDA1	39		串行数据接收输入端 (UARTA1)	P91/KR7
TXDA0	22	输出	串行发送数据输出端 (UARTA0)	P30
TXDA1	38		串行发送数据输出端 (UARTA1)	P90/KR6
ASCKA0	24	输入	波特率时钟输入至 UARTA0	P32/TIP00/TOP00/TOP01
ANI0	80	输入	模拟电压输入至 A/D 转换器	P70
ANI1	79			P71
ANI2	78			P72
ANI3	77			P73
ANI4	76			P74
ANI5	75			P75
ANI6	74			P76
ANI7	73			P77
ANI8	72			P78
ANI9	71			P79
ANI10	70			P710
ANI11	69			P711
$\text{AV}_{\text{REF0}}$	1	—	A/D 转换器参考电压输入端， 复用功能端口 7 的正电源	—
$\text{AV}_{\text{SS}}$	2	—	A/D 转换器接地端(与 $\text{V}_{\text{SS}}$ 等电位)	—
ADTRG	6	输入	A/D 转换器外部触发输入端	P03/INTP0
KR0	32	输入	按键中断输入端	P50/TIQ01/TOQ01
KR1	33			P51/TIQ02/TOQ02
KR2	34			P52/TIQ03/TOQ03/DDI
KR3	35			P53/TIQ00/TOQ00/DDO
KR4	36			P54/DCK
KR5	37			P55/DMS
KR6	38			P90/TXDA1
KR7	39			P91/RXDA1
DMS	37	输入	调试模式选择	P55/KR5
DDI	34	输入	调试数据输入端	P52/KR2/TIQ03/TOQ03
DDO	35	输出	调试数据输出端	P53/KR3/TIQ00/TOQ00
DCK	36	输入	调试时钟输入端	P54/KR4
$\overline{\text{DRST}}$	17	输入	调试复位信号输入端	P05/INTP2

表 2-3. 引脚列表 (非端口引脚) (3/3)

引脚名称	引脚号	I/O	引脚功能	复用功能
FLMD0	8	输入	闪存编程模式设定端	—
FLMD1	62			PDL5
CLKOUT	50	输出	内部系统时钟输出端	PCM1
PCL	44	输出	时钟输出 (X1 输入时钟和副时钟的定时输出端)	P913/INTP4
REGC	10	—	稳压器输出稳定电容连接端	—
RESET	14	输入	系统复位输入	—
X1	12	输入	主时钟振荡器连接	—
X2	13	—		—
XT1	15	输入	副时钟振荡器连接	—
XT2	16	—		—
V <sub>DD</sub>	9	—	内部电路的正电源端	—
V <sub>SS</sub>	11	—	内部电路接地端	—
EV <sub>DD</sub>	31	—	外部电路的正电源端 (与 V <sub>DD</sub> 等电位)	—
EV <sub>SS</sub>	30	—	外部电路接地端 (与 V <sub>SS</sub> 等电位)	—

## 2.2 引脚功能描述

### (1) P00 到 P06 (端口 0) ... 三态 I/O

P00 到 P06 用作 7 位 I/O 端口，其输入或输出方式可以位选。

该端口除了用作 I/O 功能端口以外，这些引脚还有复用功能，分别用作 NMI 输入端，外部中断请求信号输入端，定时器/计数器 I/O，A/D 转换器外部触发以及调试复位输入端。

其端口模式或控制模式可以进行位选，各引脚的有效沿(上升沿或下降沿)由寄存器 INTR0 及 INTF0 指定。

通过使用上拉电阻选择寄存器 0(PU0)，可以将片上上拉电阻连接至 P00 到 P06 引脚。

#### (a) 端口模式

P00 到 P06 的输入或输出模式，可以使用端口模式寄存器 0 (PM0)按位进行指定。

#### (b) 控制模式

##### (i) NMI (不可屏蔽中断请求) ... 输入

该引脚输入不可屏蔽中断请求信号。

##### (ii) INTP0 到 INTP3 (外部中断请求) ... 输入

这些引脚输入外部中断请求信号。

##### (iii) TIP30, TIP31 (定时器 输入) ... 输入

这些引脚向定时器 P3 (TMP3)输入外部计数时钟信号。

##### (iv) TOP30, TOP31 (定时器输出端) ... 输出

定时器 P3 (TMP3)向这些引脚输出脉冲信号。

##### (v) ADTRG (A/D 触发 输入) ... 输入

该引脚向 A/D 转换器输入外部触发信号。其(触发信号类型)由 A/D 转换器模式寄存器 0 (ADA0M0)进行控制。

##### (vi) DRST (调试复位) ... 输入

该引脚输入一个负逻辑调试复位信号，用于异步初始化调试控制单元(DCU)。维持该信号，可以实现对 DCU 的复位或使 DCU 无效。在不使用调试功能时，会一直保持该信号。

要了解详细情况，请参阅第二十四章 片上调试功能。

### (2) P30 到 P35, P38, P39 (端口 3) ... 三态 I/O

P30 到 P35, P38, 以及 P39 用作 8 位 I/O 端口，其输入或输出方式可以位选。

P30 到 P35 除了用作 I/O 功能端口以外，这些引脚还有复用功能，分别用作外部中断请求信号输入端，串行接口 I/O，以及定时器/计数器 I/O。其端口模式或控制模式可以进行位选，各引脚的有效沿(上升沿或下降沿)由寄存器 INTR3 及 INTF3 指定。

通过使用上拉电阻选择寄存器 3(PU3)，可以将片上上拉电阻连接至 P30 到 P35, P38, 以及 P39 引脚。

**(a) 端口模式**

P30 到 P35, P38, 以及 P39 的输入或输出模式, 可以使用端口模式寄存器 3 (PM3)按位进行指定。

**(b) 控制模式****(i) RXDA0 (接收数据) ... 输入**

该引脚输入 UARTA0 的串行接收数据。

**(ii) TXDA0 (发送数据) ... 输出**

该引脚输出 UARTA0 的串行发送数据。

**(iii) ASCKA0 (异步 串行时钟) ... 输入**

该引脚为 UARTA0 的异步串行时钟输入引脚。

**(iv) INTP7 (外部中断请求) ... 输入**

该引脚输入外部中断请求信号。

**(v) TIP00, TIP01, TIP10, TIP11 (定时器 输入) ... 输入**

这些引脚为定时器 P0 以及 P1 (TMP0 和 TMP1)P3 的输入引脚。

**(vi) TOP00, TOP01, TOP10, TOP11 (定时器输出端) ... 输出**

这些引脚为定时器 P0 以及 P1 (TMP0 和 TMP1)P3 的输出引脚。

**(3) P40 到 P42 (端口 4) ... 三态 I/O**

P40 到 P42 用作 3 位 I/O 端口, 其输入或输出方式可以位选。

该端口除了用作 I/O 功能端口以外, 这些引脚还用作串行接口 I/O。其端口模式或控制模式可以进行位选。

通过使用上拉电阻选择寄存器 4(PU4), 可以将片上上拉电阻连接至 P40 到 P42。

**(a) 端口模式**

P40 到 P42 的输入或输出模式, 可以使用端口模式寄存器 4 (PM4)按位进行指定。

**(b) 控制模式****(i) SIB0 (串行 输入) ... 输入**

该引脚输入 CSIB0 的串行接收数据。

**(ii) SOB0 (串行 输出) ... 输出**

该引脚输出 CSIB0 的串行发送数据。

**(iii) SCKB0 (串行时钟) ... 三态 I/O**

该引脚输入/输出 CSIB0 的串行时钟信号。

**(4) P50 到 P55 (端口 5) ... 三态 I/O**

P50 到 P55 用作 6 位 I/O 端口，其输入或输出方式可以位选。

该端口除了用作 I/O 功能端口以外，这些引脚还有复用功能，分别用作定时器/计数器 I/O，调试功能 I/O，以及按键中断输入。其端口模式或控制模式可以进行位选。

通过使用上拉电阻选择寄存器 5(PU5)，可以将片上上拉电阻连接至 P50 到 P55。

**(a) 端口模式**

P50 到 P55 的输入或输出模式，可以使用端口模式寄存器 5 (PM5)按位进行指定。

**(b) 控制模式****(i) KR0 到 KR5 (按键返回) ... 输入**

这些引脚输入一按键中断信号。其操作可以在输入端口模式下，由按键返回模式寄存器(KRM)进行指定。

**(ii) TIQ00, TIQ01, TIQ02, TIQ03 (定时器 输入) ... 输入**

这些引脚为定时器 Q0 (TMQ0)的输入引脚。

**(iii) TOQ00, TOQ01, TOQ02, TOQ03 (定时器 输出) ... 输出**

这些引脚为定时器 Q0 (TMQ0)的输出引脚。

**(iv) DDI (调试数据 输入) ... 输入**

该引脚向调试控制单元(DCU)输入调试数据。

要了解详细情况，请参阅第 24 章 片上调试功能。

**(v) DDO (调试数据 输出) ... 输出**

该引脚输出 DCU 调试数据。

要了解详细情况，请参阅第 24 章 片上调试功能。

**(vi) DCK (调试时钟 输入) ... 输入**

该引脚向调试控制单元(DCU)输入调试时钟信号。

要了解详细情况，请参阅第 24 章 片上调试功能。

**(vii) DMS (调试模式选择) ... 输入**

该引脚用以选择 DCU 的调试模式。

要了解详细情况，请参阅第 24 章 片上调试功能。

**(5) P70 到 P711 (端口 7) ... 三态 I/O**

P70 到 P711 用作 12 位 I/O 端口，其输入或输出方式可以位选。

该端口除了用作 I/O 功能端口以外，在控制模式下，这些引脚还用作 A/D 转换器的模拟量输入端。但是，在用作 A/D 转换器的模拟量输入引脚时，要设置该端口为输入模式，此时，不能读取该端口。

**(a) 端口模式**

P70 到 P711 的输入或输出模式，可以使用端口模式寄存器 7L 及 7H (PM7L 和 PM7H)按位进行指定。

**(b) 控制模式**

P70 到 P711 具有复用功能，可以分别用作 ANI0 到 ANI11 引脚。

**(i) ANI0 到 ANI11 (模拟量输入 0 到 11) ... 输入**

这些引脚向 A/D 转换器输入一模拟信号。

**(6) P90, P91, P96 到 P99, P913 到 P915 (端口 9) ... 三态 I/O**

P90, P91, P96 到 P99, 以及 P913 到 P915 用作 9 位 I/O 端口，其输入或输出方式可以位选。

该端口除了用作 I/O 功能端口以外，这些引脚还有复用功能，分别用作串行接口 I/O，定时器/计数器 I/O，时钟输出，外部中断请求信号输入端，以及按键中断输入。其端口模式或控制模式可以进行位选。P913 到 P915 引脚的有效沿(上升沿或下降沿)由寄存器 INTR9H 及 INTF9H 指定。

通过使用上拉电阻选择寄存器 9(PU9)，可以将片上上拉电阻连接至 P90, P91, P96 到 P99, 以及 P913 到 P915。

**(a) 端口模式**

P90, P91, P96 到 P99, 和 P913 到 P915 的输入或输出模式，可以使用端口模式寄存器 9 (PM9)按位进行指定。

**(b) 控制模式****(i) SIB1 (串行输入) ... 输入**

该引脚输入 CSIB1 的串行接收数据。

**(ii) SOB1 (串行输出) ... 输出**

该引脚输出 CSIB1 的串行发送数据。

**(iii)  $\overline{\text{SCKB1}}$  (串行时钟) ... 三态 I/O**

该引脚输入/输出 CSIB1 的串行时钟信号。

**(iv) RXDA1 (数据接收) ... 输入**

该引脚输入 UARTA1 的串行接收数据。

**(v) TXDA1 (数据发送) ... 输出**

该引脚输出 UARTA1 的串行发送数据。

**(vi) TIP20, TIP21 (定时器输入) ... 输入**

这些引脚为定时器 P2 (TMP2)的输入引脚。

**(vii) TOP20, TOP21 (定时器输出) ... 输出**

这些引脚为定时器 P2 (TMP2)的输出引脚。

**(viii) PCL (时钟输出) ... 输出**

该引脚输出时钟信号。



**(ix) INTP4 到 INTP6 (外部中断请求) ... 输入**

这些端口输入外部中断请求信号。

**(x) KR6, KR7 (按键返回) ... 输入**

这些引脚输入一按键中断信号。其操作可以在输入端口模式下，由按键返回模式寄存器(KRM)进行指定。

**(7) PCM0 到 PCM3 (端口 CM) ... 三态 I/O**

PCM0 到 PCM3 用作 4 位 I/O 端口，其输入或输出方式可以位选。

该端口除了用作 I/O 功能端口以外，在控制模式下，该引脚还用作总线时钟输出端。

**(a) 端口模式**

PCM0 到 PCM3 的输入或输出模式，可以使用端口模式寄存器 CM (PMCM)按位进行指定。

**(b) 控制模式****(i) CLKOUT (时钟 输出) ... 输出**

该引脚输出一由内部产生的总线时钟信号。

**(8) PCS0, PCS1 (端口 CS) ... 三态 I/O**

PCS0 和 PCS1 用作 2 位 I/O 端口，其输入或输出方式可以位选。

**(a) 端口模式**

PCS0 和 PCS1 的输入或输出模式，可以使用端口模式寄存器 CS (PMCS)按位进行指定。

**(9) PCT0, PCT1, PCT4, PCT6 (端口 CT) ... 三态 I/O**

PCT0, PCT1, PCT4, 和 PCT6 用作 4 位 I/O 端口，其输入或输出方式可以位选。

**(a) 端口模式**

PCT0, PCT1, PCT4, 和 PCT6 的输入或输出模式，可以使用端口模式寄存器 CT (PMCT)按位进行指定。

**(10) PDL0 到 PDL11 (端口 DL) ... 三态 I/O**

PDL0 到 PDL11 用作 12 位 I/O 端口，其输入或输出方式可以位选。

当对闪存编程 (当向 FLMD0 输入一高电平时) 时，PDL5 还用作 FLMD1 引脚。此时，请确保在 FLMD1 引脚输入一低电平。

**(a) 端口模式**

PDL0 到 PDL11 的输入或输出模式，可以使用端口模式寄存器 DL(PMDL)按位进行指定。

**(11)  $\overline{\text{RESET}}$  (复位) ... 输入**

$\overline{\text{RESET}}$  输入为异步输入。不论操作时钟处于怎样操作状态，当有一固定宽度的低电平信号输入到  $\overline{\text{RESET}}$  端时，在所有操作中，系统会优先执行复位操作。

该引脚用作释放待机模式(HALT, IDLE, 或 STOP)，以及正常的初始化/启动操作。

**(12) X1, X2 (主时钟晶振)**

这些引脚用于连接产生系统时钟的振荡器。

**(13) XT1, XT2 (副时钟晶振)**

这些引脚用于连接产生副时钟的振荡器。

**(14) AVss (模拟接地)**

该引脚为 A/D 转换器和复用功能端口的接地端。

**(15) AVREF0 (模拟参考电压) ... 输入**

该引脚为 A/D 转换器和复用功能端口提供正模拟电源。

同时, 该引脚也向 A/D 转换器提供参考电压。

**(16) EVDD (端口电源)**

该引脚为 I/O 端口和复用功能引脚提供正电源。

**(17) EVss (端口地)**

该引脚为 I/O 端口和复用功能引脚的接地端。

**(18) VDD (电源)**

该引脚提供正电源。要将所有 VDD 引脚端都连接到正电源。

**(19) Vss (接地)**

该引脚为接地端。要将所有 Vss 引脚端都连接到接地端。

**(20) FLMD0 (闪存编程模式) ... 输入**

该引脚为闪存编程模式的信号输入引脚。

在正常操作模式下, 要将该引脚连接至 Vss 端。

**(21) REGC (稳压器控制端) ... 输入**

该引脚连接稳压器电容。

## 2.3 引脚 I/O 电路类型及未用引脚建议连接方式

(1/2)

引脚	引脚号	I/O 电路类型	建议连接方式
P00/TIP31/TOP31	3	5-W	输入：通过一个电阻单独连接到 EV <sub>DD</sub> 或 EV <sub>SS</sub> 端 输出：开路
P01/TIP30/TOP30	4		
P02/NMI	5		
P03/INTP0/ADTRG	6		
P04/INTP1	7		
P05/INTP2/DRST	17	5-AF	输入：单独连接到 EV <sub>SS</sub> 输出：开路
P06/INTP3	18	5-W	输入：通过一个电阻单独连接到 EV <sub>DD</sub> 或 EV <sub>SS</sub> 端 输出：开路
P30/TXDA0	22	5-A	输入：通过一个电阻单独连接到 EV <sub>DD</sub> 或 EV <sub>SS</sub> 端 输出：开路
P31/RXDA0/INTP7	23	5-W	
P32/ASCKA0/TIP00/ TOP00/ TOP01	24		
P33/TIP01/TOP01	25		
P34/TIP10/TOP10	26		
P35/TIP11/TOP11	27		
P38	28	5-A	
P39	29		
P40/SIB0	19	5-W	输入：通过一个电阻单独连接到 EV <sub>DD</sub> 或 EV <sub>SS</sub> 端 输出：开路
P41/SOB0	20	5-A	
P42/SCKB0	21	5-W	
P50/KR0/TIQ01/TOQ01	32	5-W	输入：通过一个电阻单独连接到 EV <sub>DD</sub> 或 EV <sub>SS</sub> 端 输出：开路
P51/KR1/TIQ02/TOQ02	33		
P52/KR2/TIQ03/TOQ03/DDI	34		
P53/KR3/TIQ00/TOQ00/ DDO	35		
P54/KR4/DCK	36		
P55/KR5/DMS	37		
P70/ANI0 到 P711/ANI11	80 到 69	11-G	输入：通过一个电阻单独连接到 AV <sub>REF0</sub> 或 AV <sub>SS</sub> 端 输出：开路
P90/KR6/TXDA1	38	5-W	输入：通过一个电阻单独连接到 EV <sub>DD</sub> 或 EV <sub>SS</sub> 端 输出：开路
P91/KR7/RXDA1	39		
P96/TIP21/TOP21	40		
P97/SIB1/TIP20/TOP20	41		
P98/SOB1	42	5-A	
P99/SCKB1	43	5-W	
P913/INTP4/PCL	44		
P914/INTP5	45		
P915/INTP6	46		

(2/2)

引脚	引脚号	I/O 电路类型	建议连接方式
PCM0	49	5	输入： 通过一个电阻单独连接到 $EV_{DD}$ 或 $EV_{SS}$ 端 输出： 开路
PCM1/CLKOUT	50		
PCM2, PCM3	51, 52		
PCS0, PCS1	47, 48	5	输入： 通过一个电阻单独连接到 $EV_{DD}$ 或 $EV_{SS}$ 端 输出： 开路
PCT0, PCT1, PCT4, PCT6	53 到 56	5	输入： 通过一个电阻单独连接到 $EV_{DD}$ 或 $EV_{SS}$ 端 输出： 开路
PDL0 到 PDL4	57 到 61	5	输入： 通过一个电阻单独连接到 $EV_{DD}$ 或 $EV_{SS}$ 端 输出： 开路
PDL5/FLMD1	62		
PDL6 到 PDL11	63 到 68		
$AV_{REF0}$	1	—	直接连接到 $V_{DD}$
$AV_{SS}$	2	—	—
FLMD0 <sup>‡</sup>	8	—	直接连接到 $V_{SS}$
REGC	10	—	—
$\overline{RESET}$	14	2	—
X1	12	—	—
X2	13	—	—
XT1	15	16	通过一个电阻连接到 $V_{SS}$
XT2	16	16	开路
$V_{DD}$	9	—	—
$V_{SS}$	11	—	—
$EV_{DD}$	31	—	—
$EV_{SS}$	30	—	—

注 在自编程期间，如果有超出噪声消除宽度的噪声输入至  $\overline{RESET}$  端，当有电容连接至 FLMD0 引脚时，根据该电容的充电结束时间，可能会启动闪存在板模式。因此，不要将电容连接到 FLMD0 引脚端。

## 2.4 引脚 I/O 电路

图 2-1. 引脚 I/O 电路类型 (1/2)

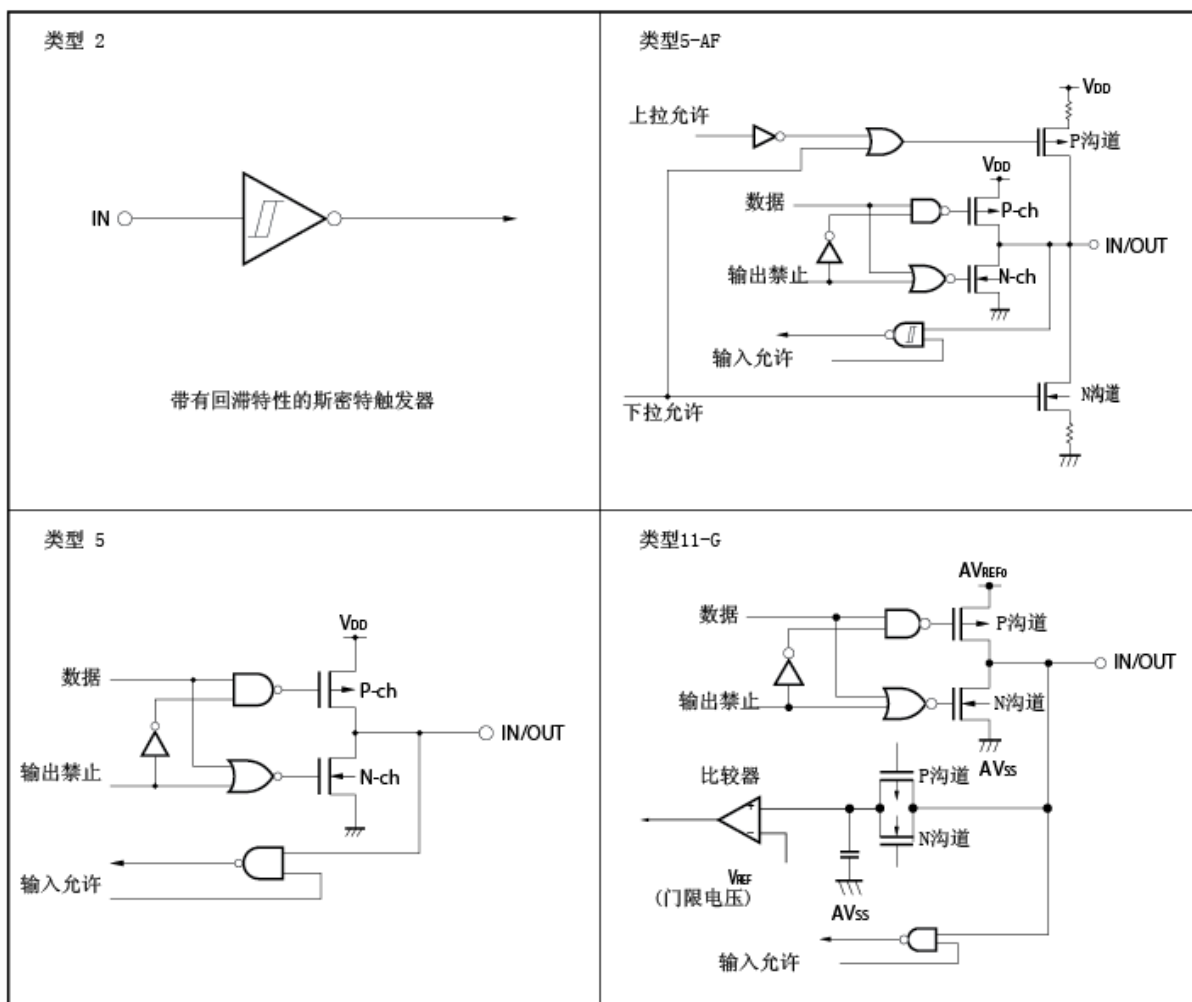
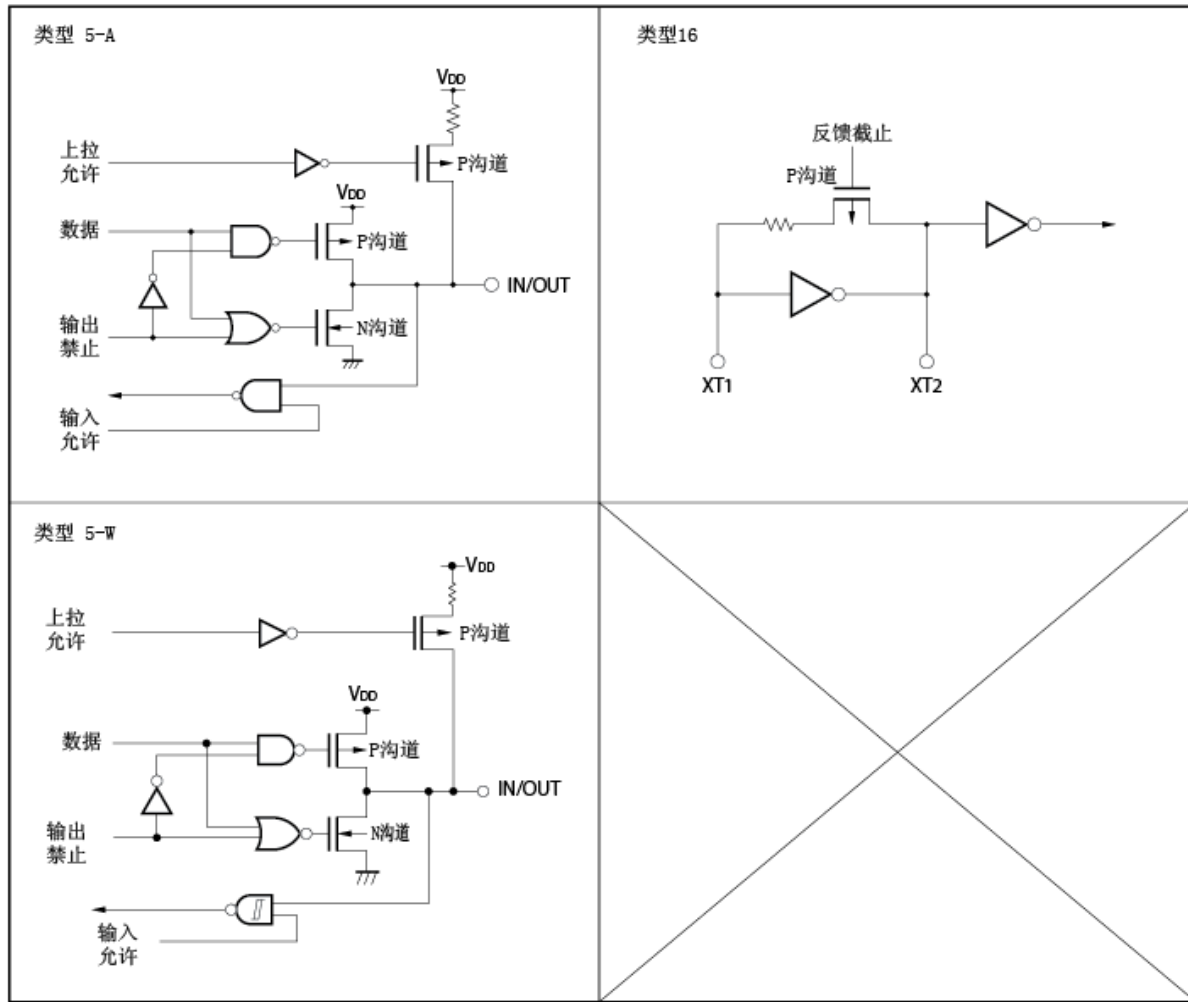


图 2-1. 引脚 I/O 电路类型 (2/2)



备注 VDD 与 EVDD 电位相同。同样，VSS 与 EVSS 电位相同。

## <R> 2.5 注意事项

注意，即使是在上电复位期间，以下引脚可能会临时输出一个未定义的电平信号。

P53/KR3/TIQ00/TOQ00/DDO 引脚

## 第三章 CPU 功能

V850ES/HF2 CPU 基于 RISC 体系结构构建, 通过使用 5 级管线控制, 对几乎所有指令都可以在一个时钟周期内完成执行。

### 3.1 特性

- 最小指令执行时间: 50 ns (在 20 MHz 时钟频率下运行时)
- 存储空间
  - 程序存储空间(物理地址): 64 MB 线性地址空间
  - 数据存储空间(逻辑地址): 4 GB 线性地址空间
- 通用寄存器个数: 32 位 × 32 个寄存器
- 内部 32 位体系结构
- 5 级管线控制
- 乘法/除法指令
- 饱和运算指令
- 32 位移位指令: 1 个时钟周期
- 长/短格式数据装载/存贮指令
- 四种类型位操作指令
  - SET1
  - CLR1
  - NOT1
  - TST1

3.2 CPU 寄存器组

V850ES/HF2 微控制器的寄存器分为两类：通用程序寄存器和专用系统寄存器。所有寄存器的数据宽度都是 32 位。需要了解详细内容，可参阅 **V850ES 架构用户手册**。

(1) 程序寄存器组		(2) 系统寄存器组	
31	0	31	0
r0	(零寄存器)	EIPC	(中断状态保存寄存器)
r1	(汇编保留寄存器)	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	(程序状态字)
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		DBPC	(异常/调试陷阱状态保存寄存器)
r14		DBPSW	(异常/调试陷阱状态保存寄存器)
r15			
r16			
r17		CTBP	(CALLT 基址指针)
r18			
r19			
r20			
r21			
r22			
r23			
r24			
r25			
r26			
r27			
r28			
r29	(元素指针 (EP))		
r30	(链接指针 (LP))		
r31			
31	0		
PC	(程序计数器)		



3.2.1 程序寄存器组

程序寄存器包括通用寄存器和程序计数器。

(1) 通用寄存器(r0 到 r31)

共有 32 个通用寄存器，r0 到 r31，可供使用。其中任何一个寄存器均可用来存放数据变量或者是地址变量。但是，必须注意寄存器 r0 和 r30 在指令中的隐含用法，在使用它们的时候要注意一些。寄存器 r0 中存放的数据始终为 0，用于使用数据 0 的操作或者 0 偏移量寻址。寄存器 r30 在 SLD 和 SST 指令访问存储空间时用作这些的基址指针。而寄存器 r1，r3 到 r5 以及 r31 隐含用于汇编器和 C 编译器。在使用这些寄存器时，为保护起见，应该先将其中内容保存起来，在使用完毕后再恢复其中内容。寄存器 r2 有时用于实时 OS，如果实时 OS 没有用到 r2，则该寄存器可以做变量寄存器之用。

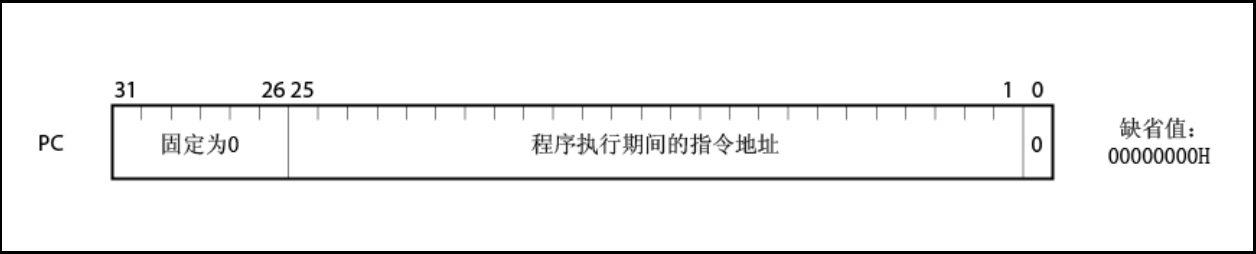
表 3-1. 程序寄存器

名称	用法	操作
r0	零寄存器	始终存放数据 0
r1	汇编程序保留寄存器	用作工作寄存器以创建 32 位立即数
r2	用作地址变量/数据变量寄存器(如果实时 OS 没有使用 寄存器 r2)	
r3	堆栈指针	在函数调用时用于创建堆栈
r4	全局指针	用于访问数据区中的全局变量
r5	文本指针	用作指示文本域起始地址寄存器(该区域为程序代码存放区域)
r6 到 r29	地址/数据变量寄存器	
r30	元素指针	用作访问存储器的基址指针
r31	链接指针	在编译器调用函数时用该寄存器
PC	程序计数器	在程序执行期间，存放指令地址

**备注** 需要更进一步了解有关寄存器 r1，r3 到 r5 以及 r31 在汇编器和 C 编译器中的应用，可以参阅 **CA850 (C 编译包) 汇编语言用户手册**。

(2) 程序计数器 (PC)

程序计数器存放指令执行期间的指令地址。该寄存器低 32 有效，其中第 31 到 26 位固定为 0，即使第 25 位向第 26 位有进位也将忽略不计。第 0 为固定为 0，这也就意味着程序执行不可能转移到奇地址处。



### 3.2.2 系统寄存器组

系统寄存器控制 CPU 的状态并存放中断信息。

这些寄存器可以由系统寄存器的装载/存储指令(LDSR 和 STSR)进行读写，使用的系统寄存器号列于下表。

表 3-2. 系统寄存器号

系统寄存器号	系统寄存器名称	操作数说明	
		LDSR 指令	STSR 指令
0	中断状态保存寄存器 (EIPC) <sup>注 1</sup>	√	√
1	中断状态保存寄存器 (EIPSW) <sup>注 1</sup>	√	√
2	NMI 状态保存寄存器 (FEPC) <sup>注 1</sup>	√	√
3	NMI 状态保存寄存器 (FEPSW) <sup>注 1</sup>	√	√
4	中断源寄存器 (ECR)	×	√
5	程序状态字 (PSW)	√	√
6 到 15	将来功能扩展保留寄存器(如果对这些寄存器进行访问，不能保证操作的有效性)	×	×
16	CALLT 指令执行状态保存寄存器 (CTPC)	√	√
17	CALLT 指令执行状态保存寄存器 (CTPSW)	√	√
18	异常/调试陷阱状态保存寄存器 (DBPC)	√ <sup>注 2</sup>	√ <sup>注 2</sup>
19	异常/调试陷阱状态保存寄存器 (DBPSW)	√ <sup>注 2</sup>	√ <sup>注 2</sup>
20	CALLT 指令基址指针 (CTBP)	√	√
21 到 31	将来功能扩展保留寄存器(如果对这些寄存器进行访问，不能保证操作的有效性)	×	×

- 注**
1. 由于只有一组这样中断状态保存寄存器可用，所以，如果允许多重中断，这些寄存器的内容必须在程序中通过编程加以保存。
  2. 对这些寄存器的访问，只有在 DBTRAP 指令或非法操作码与 DBRET 指令执行的间隔期间才能进行。

**注意事项** 即使 EIPC 或 FEPC 寄存器，或 CTPC 寄存器的第 0 位由 LDSR 指令设置为 1 了，在中断服务完成后，程序执行由 RETI 指令返回到主程序时，第 0 位仍然会被忽略(这是因为 PC 的第 0 位是固定为 0 的)。这样，对寄存器 EIPC，FEPC 以及 CTPC 就设置了一个偶数值(第 0 位 = 0)。

**备注**

√: 表示可以访问

×: 表示禁止访问

(1) 中断状态保存寄存器 (EIPC 和 EIPSW)

寄存器 EIPC 和 EIPSW 用于在产生中断时保存相关状态。

如果产生的是软件异常或者是可屏蔽中断，那么就将程序计数器(PC)中的内容保存到 EIPC 寄存器中，将程序状态字(PSW)中的内容保存到 EIPSW 寄存器中(如果产生的是不可屏蔽中断，那么就将这些相应寄存器内容保存到 NMI 状态保存寄存器(FEPC 及 FEPSW)中去。

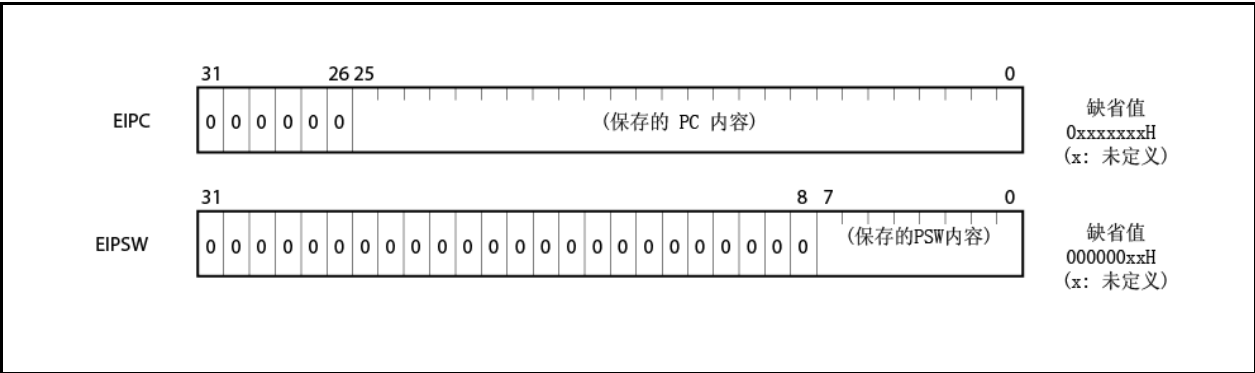
当有软件异常或者是可屏蔽中断产生时，就将正在执行指令的下一条指令地址保存到寄存器 EIPC 中。有些指令除外(可参阅 14.8 指令执行期间 CPU 不应答中断的指令)。

将 PSW 当前内容保存到 EIPSW 寄存器中去。

由于只有一组这样中断状态保存寄存器可用，所以，当允许多重中断，这些寄存器的内容必须在程序中通过编程加以保存。

EIPC 的第 31 位到第 26 位以及 EIPSW 的第 31 位到第 8 位作为保留位用于将来功能扩展(这些位始终固定为 0 值)。

通过 RETI 指令，将寄存器 EIPC 的值恢复到 PC 中，寄存器 EIPSW 的值恢复到 PSW 中。



(2) NMI 状态保存寄存器(FEPC 和 FEPSW)

寄存器 FEPC 和 FEPSW 用于在产生不可屏蔽中断(NMI)时保存相关状态。

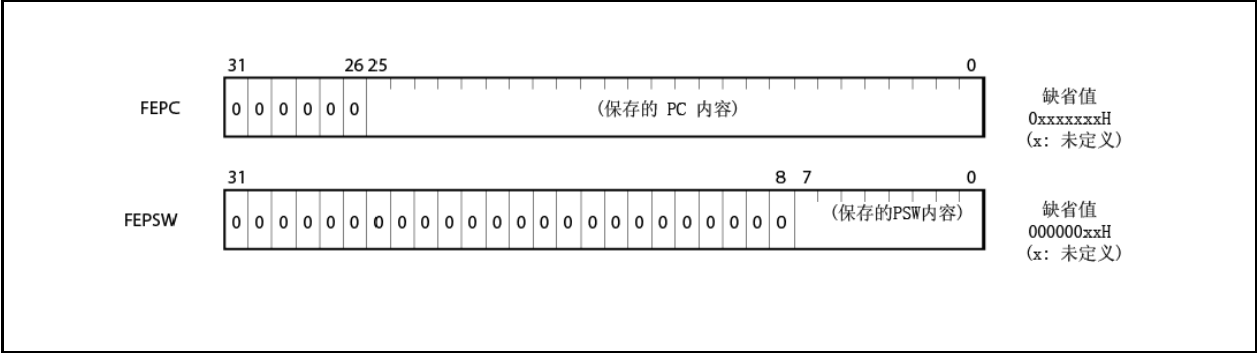
如果有不可屏蔽中断 NMI 产生时，那么就将程序计数器(PC)中的内容保存到 FEPC 寄存器中，将程序状态字(PSW)中的内容保存到 FEPSW 寄存器中。

当有不可屏蔽中断 NMI 产生时，就将正在执行指令的下一条指令地址保存到寄存器 FEPC 中。有些指令除外。将 PSW 当前内容保存到 FEPSW 寄存器中去。

由于只有一组这样 NMI 状态保存寄存器可用，所以，当允许多重中断，这些寄存器的内容必须在程序中通过编程加以保存。

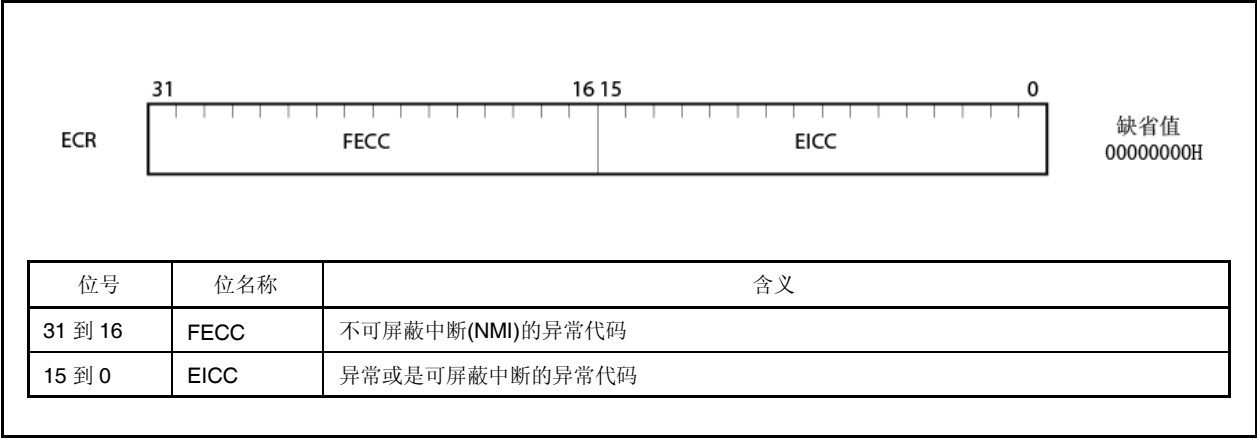
FEPC 的第 31 位到第 26 位以及 FEPSW 的第 31 位到第 8 位作为保留位用于将来功能扩展(这些位始终固定为 0 值)。

通过 RETI 指令，将寄存器 FEPC 的值恢复到 PC 中，寄存器 FEPSW 的值恢复到 PSW 中。



(3) 中断源寄存器 (ECR)

如果有异常或中断产生，那么，中断源寄存器(ECR)中就存放该异常或中断源信息。该寄存器存放的是每一个中断源的异常代码。由于该寄存器是只读寄存器，所以，不能通过 LDSR 指令将数据写入其中。



(4) 程序状态字 (PSW)

程序状态字(PSW)是一个指示程序状态(指令执行结果)和 CPU 状态这些标志位的集合体。  
如果通过 LDSR 指令更改了该寄存器某一位的内容,那么,在 LDSR 指令完成后,该新内容就会立即生效。但是,如果将 ID 标志位设为 1,那么在 LDSR 执行期间,中断请求将得不到应答。  
该寄存器的第 31 位到第 8 位作为保留位用于将来功能扩展(这些位始终固定为 0 值)。

(1/2)

PSW

31

8 7 6 5 4 3 2 1 0

RFU

NP EP ID SAT CY OV S Z

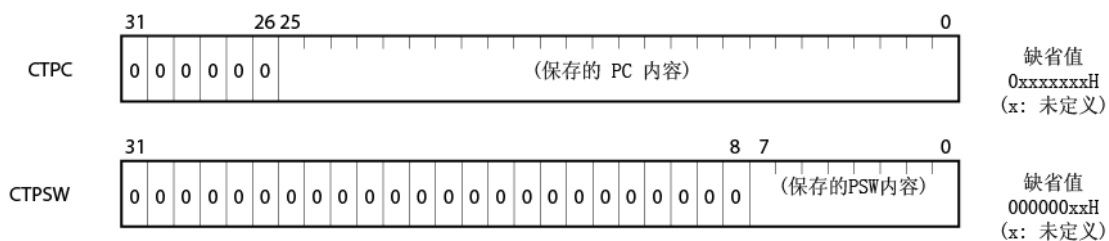
缺省值  
00000020H

位号	标志名称	含义
31 到 8	RFU	保留域: 固定为 0 值。
7	NP	标示正在服务一不可屏蔽中断(NMI)。当响应 NMI 中断请求时, 该位置 1, 禁止多重中断。 0: 没有服务 NMI 中断 1: 正在服务 NMI 中断
6	EP	标示正在处理一个异常, 当有异常产生时, 该位置 1, 即使该位置 1, 也可以响应中断请求。 0: 没有处理异常 1: 正在处理异常 I
5	ID	标示是否响应可屏蔽中断 0: 允许中断 1: 禁止中断
4	SAT <sup>注</sup>	标示饱和和运算指令结果溢出且饱和。由于该位为一累计标志位, 所以, 当饱和和运算指令结果饱和时, 该位便置 1, 即便是下一个运算结果没有饱和, 该位也不会清 0。请使用 LDSR 指令将该位清 0。在执行一算术运算时, 该位既不置 1, 也不清 0。 0: 没有饱和 1: 饱和
3	CY	标示运算结果有无进位或借位产生 0: 无进位或借位产生 1: 有进位或借位产生
2	OV <sup>注</sup>	标示在运算过程中有无溢出 0: 没有溢出出现。 1: 有溢出出现
1	S <sup>注</sup>	标示操作结果是否为负数 0: 结果为正数或零 1: 结果为负数
0	Z	标示运算结果是否为 0。 0: 结果不为 0。 1: 结果为 0

备注 也请参阅下页的注。

运算结果状态	标志位状态			饱和处理的运算结果
	SAT	OV	S	
超过最大正数值	1	1	0	7FFFFFFFH
超过最大负数值	1	1	1	80000000H
正 (没有超过最大值)	运算前保持该值	0	0	运算结果为其自身
负(没有超过最大值)			1	

CTPC 的第 31 位到第 26 位以及 CTPSW 的第 31 位到第 8 位作为保留位用于将来功能扩展(这些位始终固定为 0 值)。



### (6) 异常/调试陷阱状态保存寄存器 (DBPC 和 DBPSW)

寄存器 DBPC 和 DBPSW 是异常/调试陷阱状态保存寄存器。

如果有异常陷阱或者调试陷阱产生，那么就将程序计数器(PC)中的内容保存到 DBPC 寄存器中，将程序状态字 (PSW)中的内容保存到 DBPSW 寄存器中去。

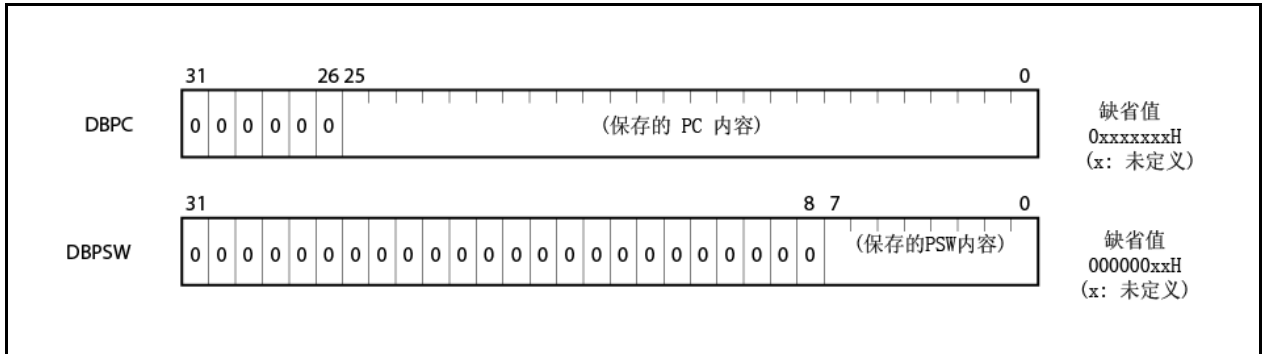
在异常陷阱或者调试陷阱产生时，保存到 DBPC 寄存器中的内容是正在执行指令的下一条指令的地址。

PSW 寄存器的当前内容保存到 DBPSW 寄存器中。

只有在 DBTRAP 指令或非法操作码与 DBRET 指令执行的间隔期间，才能对该寄存器进行读写。

DBPC 的第 31 位到第 26 位以及 DBPSW 的第 31 位到第 8 位作为保留位用于将来功能扩展(这些位始终固定为 0 值)。

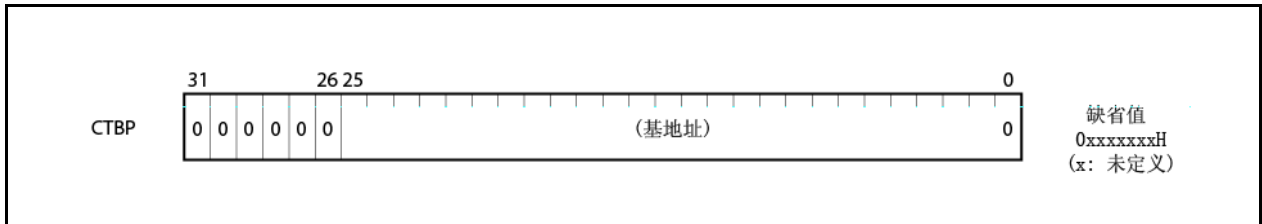
通过 DBRET 指令，将寄存器 DBPC 的值恢复到 PC 中，寄存器 DBPSW 的值恢复到 PSW 中。



### (7) CALLT 指令基址指针 (CTBP)

CALLT 指令基址指针(CTBP)用于指定表地址或者生成一个目标地址(第 0 位固定为 0 值)。

该寄存器第 31 位到第 26 位作为保留位用于将来功能扩展(这些位固定为 0 值)。



### 3.3 操作模式

V850ES/HF2 有如下几种操作模式。

#### (1) 正常操作模式

在这种操作模式下，系统复位释放后，程序执行转移到内部 ROM 中复位入口地址处，然后开始执行指令。

#### (2) 闪存编程模式

在这种操作模式下，通过使用闪存编程器可以对内部闪存进行编程。

#### (3) 片上调试模式

V850ES/HF2 提供了片上调试功能，它是通过片上调试仿真器运用 JTAG(联合测试行动小组)通信规程来实现的。

要了解详细内容，请参阅第二十四章 片上调试功能。

#### 3.3.1 操作模式的指定

通过使用 FLMD0 和 FLMD1 这 2 个引脚来指定操作模式。

在正常操作模式下，必须确保当复位释放后，在 FLMD0 引脚有一低电平信号输入。

在闪存编程模式下，如果连接了闪存编程器，则就有一高电平由闪存编程器输入到 FLMD0 引脚。但是，在自编程模式下，该高电平信号必需由外部电路输入。

复位释放后操作		复位后操作模式
FLMD0	FLMD1	
L	×	正常操作模式
H	L	闪存编程模式
H	H	禁止设置

**备注**

L: 表示低电平输入

H: 表示高电平输入

×: 无所谓高电平或低电平

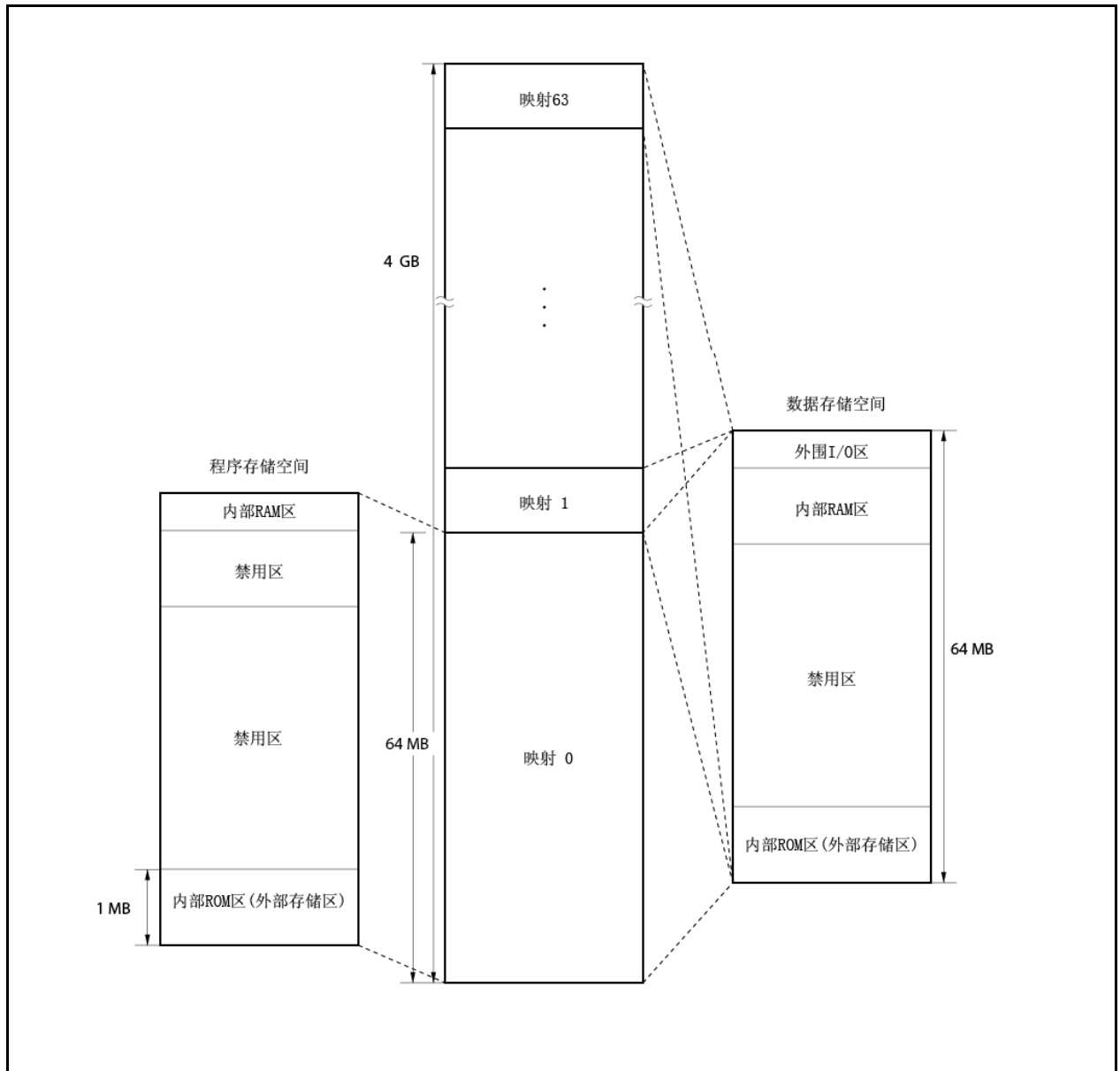


### 3.4 地址空间

#### 3.4.1 CPU 地址空间

用于支持寻址指令地址的存储空间包括，多达 1MB 内部 ROM 区域，以及最多可达 64MB 的线性地址空间(程序空间)区域的内部 RAM 区域。同时，有多达 4GB 的线性地址空间(数据空间)用于支持操作数寻址(数据访问)。但是，这 4GB 的地址空间，可以看作是 64 个 64 MB 的物理地址空间的映像。这意味着不管第 31 位到第 26 位的取值如何，均可以对同样 64 MB 物理地址空间进行访问。

图 3-1. 地址空间映像



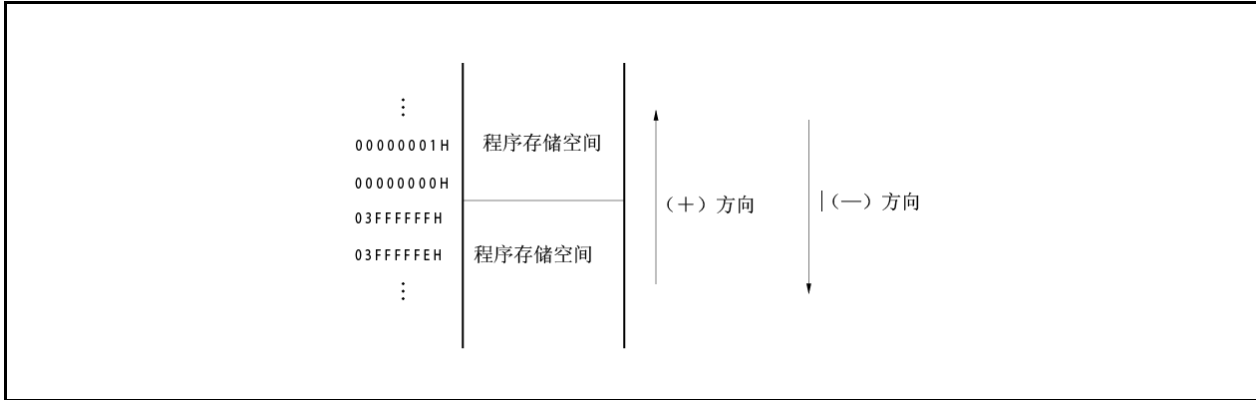
### 3.4.2 CPU 地址空间的环绕

#### (1) 程序空间

对于 32 位的 PC (程序计数器), 其高 6 位固定为 0, 而只有低 26 位是有效的。在计算程序转移地址时, 高 6 位会忽略第 25 位向第 26 位的进位或借位。

因此, 程序空间的最高地址, 03FFFFFFH, 与其最低地址 00000000H 是相邻而连续的。程序空间地址的最高地址和最低地址以这种方式连续起来, 称为地址环绕。

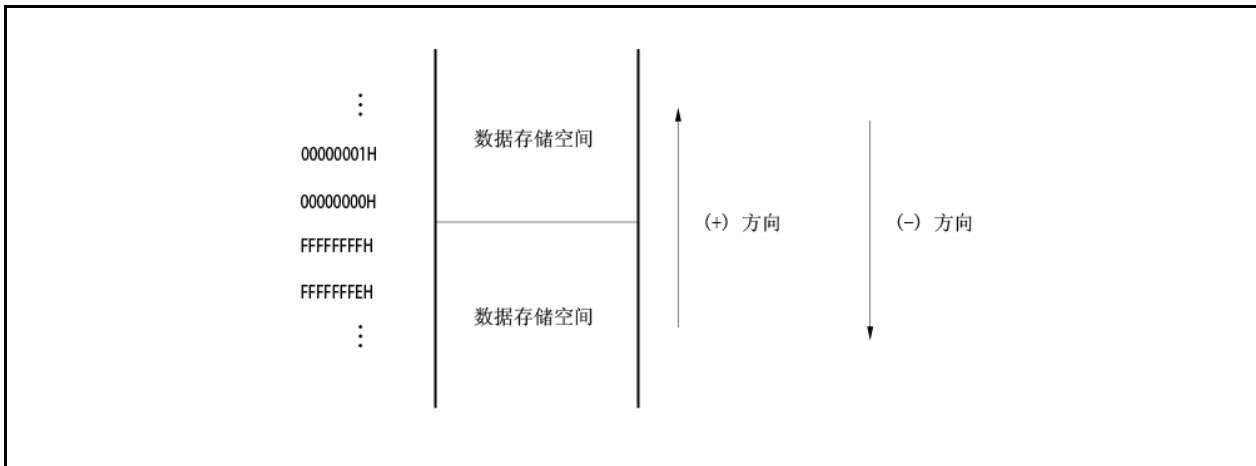
**注意事项** 由于从地址 03FFF000H 到 03FFFFFFH 的 4 KB 地址区域为片上外围 I/O 区域, 所以不能从该地址区域取指令。因此, 不允许执行那些转移地址计算结果影响该地址区域的操作。



#### (2) 数据地址空间

操作数地址计算操作结果超出 32 位时, 该结果地址将被忽略。

因此, 数据地址空间的最高地址, FFFFFFFFH, 与其最低地址 00000000H 是相邻而连续的, 而且在这两个地址边界处环绕起来。



3.4.3 存储空间映射图

如下所示的存储区域在 V850ES/HF2 内是保留的。

图 3-2. 数据存储空间映射图(物理地址)

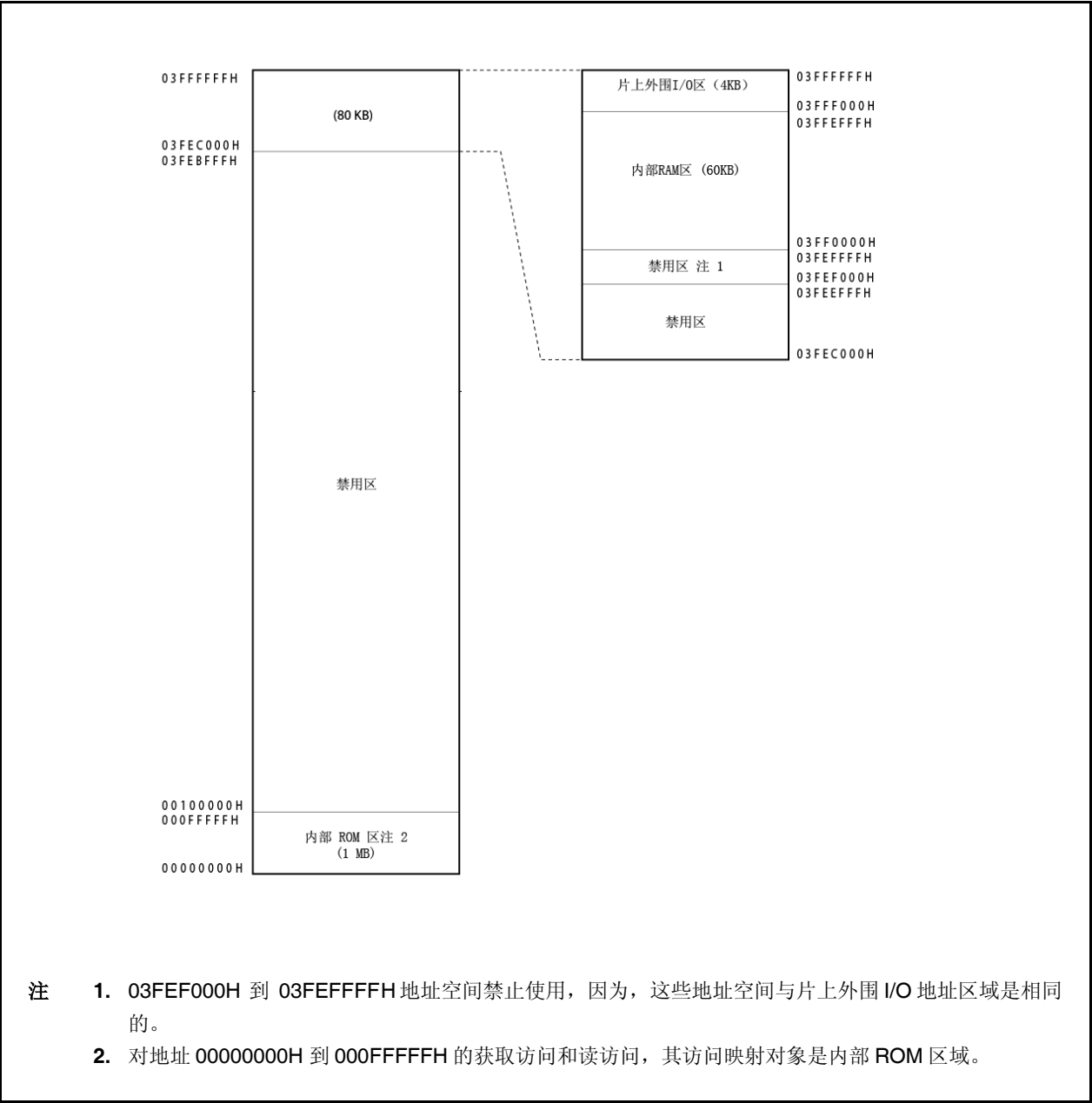
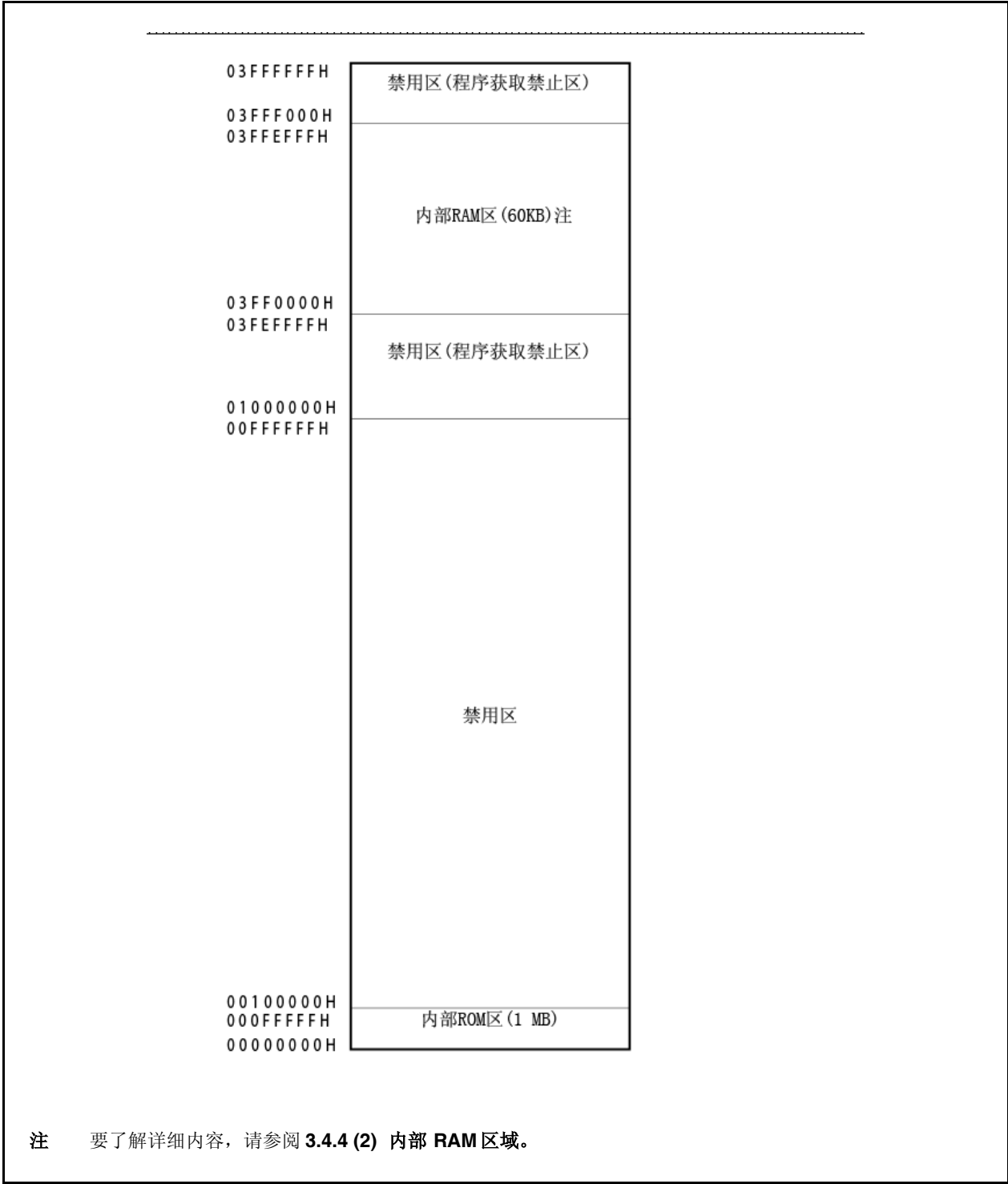


图 3-3. 程序存储器映射图



3.4.4 存储区域

(1) 内部 ROM 区域

最长达 1 MB 存储区域保留作为内部 ROM 区域。

(a) 内部 ROM (64 KB)

在  $\mu$ PD70F3702 类型控制器中，配备有 64 KB 存储区域，所分配的地址空间为 00000000H 到 0003FFFFH。

禁止访问 00100000H 到 00100000H 地址空间区域。

图 3-4. 内部 ROM 区域(64 KB)



(b) 内部 ROM (128 KB)

在  $\mu$ PD70F3703 类型控制器中，128 KB 存储区域分配给地址空间 00000000H 到 001FFFFFH。

禁止对 00200000H 到 00FFFFFFH 地址空间的访问。

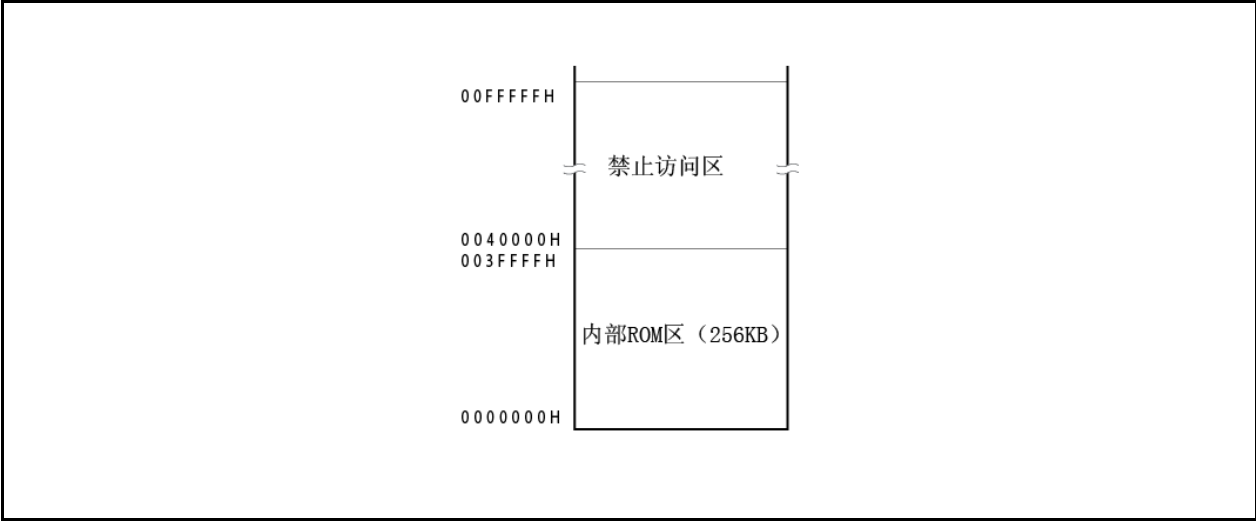
图 3-5. 内部 ROM 区域(128 KB)



(c) 内部 ROM (256 KB)

在μPD70F3704 类型控制器中，256 KB 存储区域分配给地址空间 00000000H 到 003FFFFFH。禁止对 0040000H 到 00FFFFFFH 地址空间的访问。

图 3-6. 内部 ROM 区域(256 KB)



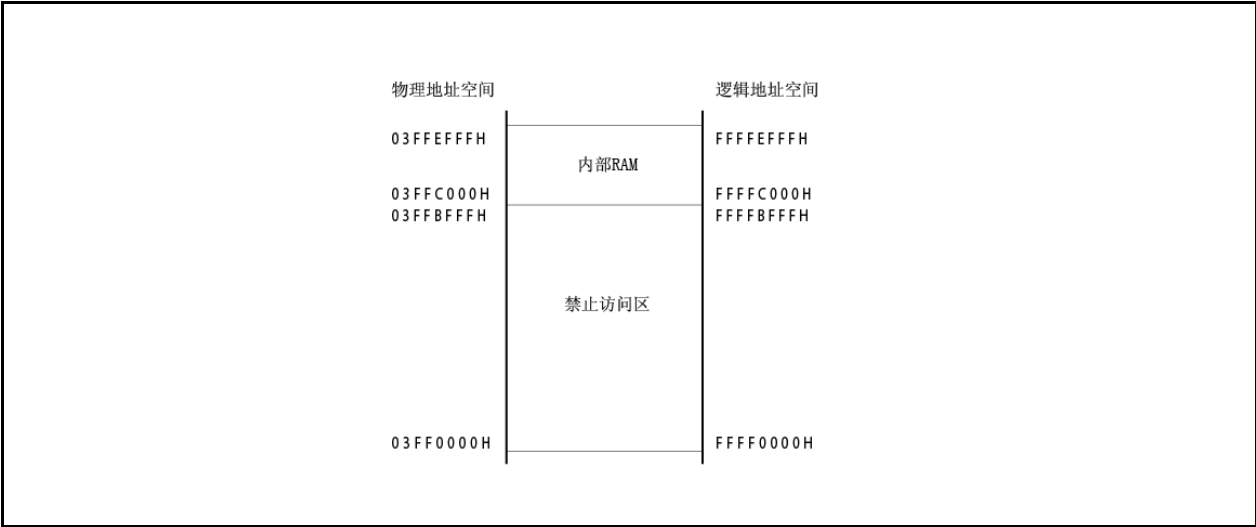
(2) 内部 RAM 区域

最多达 60 KB 存储区域保留作为内部 RAM 区域。

(a) 内部 RAM (12 KB)

在 V850ES/HF2 类型控制器中，12 KB 存储区域分配给地址空间 03FFC000H 到 03FFEFFFH。禁止对 03FF0000H 到 03FFBFFFFH 地址空间的访问。

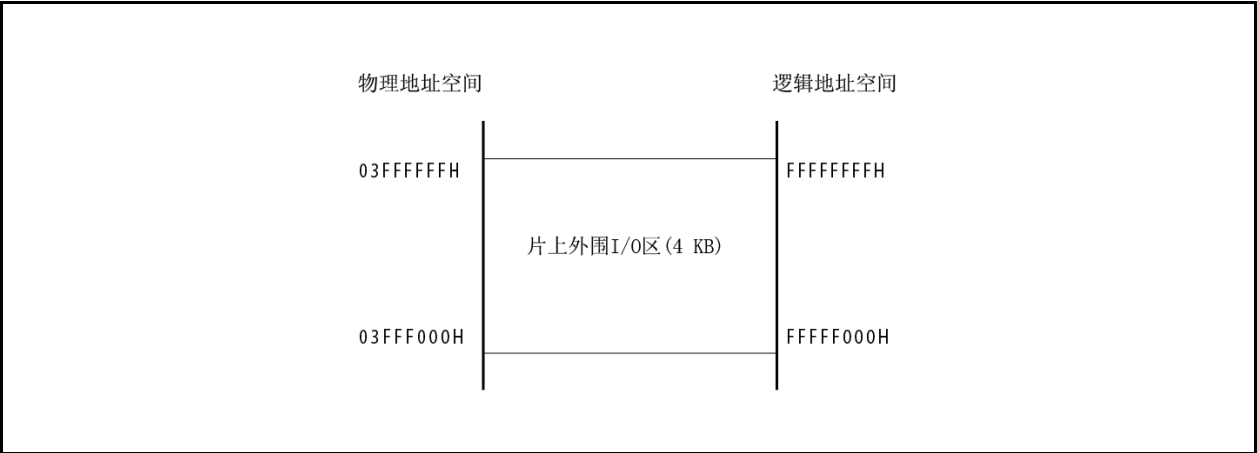
图 3-7. 内部 RAM 区域(12 KB)



(3) 片上外围 I/O 区

从 03FFF000H 到 03FFFFFFH 这 4 KB 地址空间保留用作片上外围 I/O 区。

图 3-8. 片上外围 I/O 区



为片上外围 I/O 区指定操作模式并监视其状态的这些外围 I/O 寄存器映射到片上外围 I/O 区。注意，不能从该区获取程序。

- 注意事项
- 1. 当以一个字长为单位访问这类寄存器时，则该字域要以半字为单位进行两次访问，并且以先低字节后高字节的顺序进行，同时该地址单元的低 2 位忽略不计。
  - 2. 对能够以字节为单位进行访问的寄存器，如果以半字为单位进行访问，那么，在读寄存器时，其高 8 位为未定义数据。写数据时，将数据写入低 8 位。
  - 3. 那些没有定义的地址区域保留为将来扩展之用。如果对这些地址进行访问，其操作没有定义而且也不保证操作的有效性。

### 3.4.5 地址空间建议使用方法

V850ES/HF2 体系结构要求：当访问数据空间的操作数时，必须确保有一个寄存器用作指针以生成地址。存储于该  $\pm 32$  KB 指针中的操作数地址可以直接由指令存取。因为可用作指针的通用寄存器的数目是有限的，通用寄存器多数为变量所用。但是，通过确保尽可能多的通用寄存器为变量所用，一方面可以防止在地址计算期间当指针值发生变化时而引起的系统性能的降低，同时也可以减小程序的大小。

#### (1) 程序空间

对于 32 位的 PC (程序计数器)，其高 6 位固定为 0，而只有低 26 位是有效的。因此，对于程序空间，从地址 00000000H 开始的 64 MB 连续空间与存储器映射是无条件对应的。

要将内部 RAM 用作程序空间，请访问地址空间 03FFC000H 到 03FFFEFFH。

**注意事项** 如果转移指令位于内部 RAM 区的上限地址处，那么这一跨越到片上外围 I/O 区的预取操作(无效获取)是不会发生的。



(2) 数据空间

对于 V850ES/HF2 来说，好像是在 4 GB CPU 地址空间内有 64 个 64 MB 的地址空间。因此，要将这 26 位地址的最低有效位(第 25 位)作为符号扩展位将该地址扩展为 32 位，并分配为一个地址。

(a) 地址环绕应用示例

如果 R = r0 (零寄存器) 指定为指令 LD/ST disp16 [R]指令所用，那么就可以通过符号扩展 disp16 对 00000000H ±32 KB 地址范围空间进行寻址。那么，通过一个指针就可以对所有资源进行寻址，包括内部硬件。

零寄存器(r0)是由硬件将其值固定为 0 的，实际上是不需要专门用作指针的寄存器。

图 3-9. 地址环绕 (μPD70F3704)

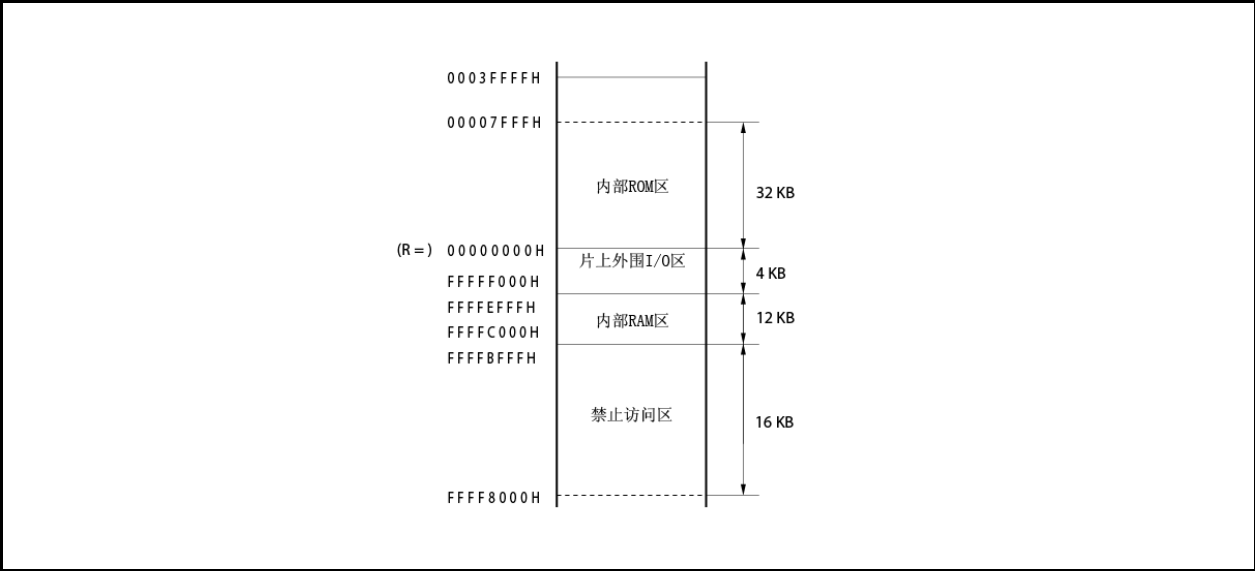
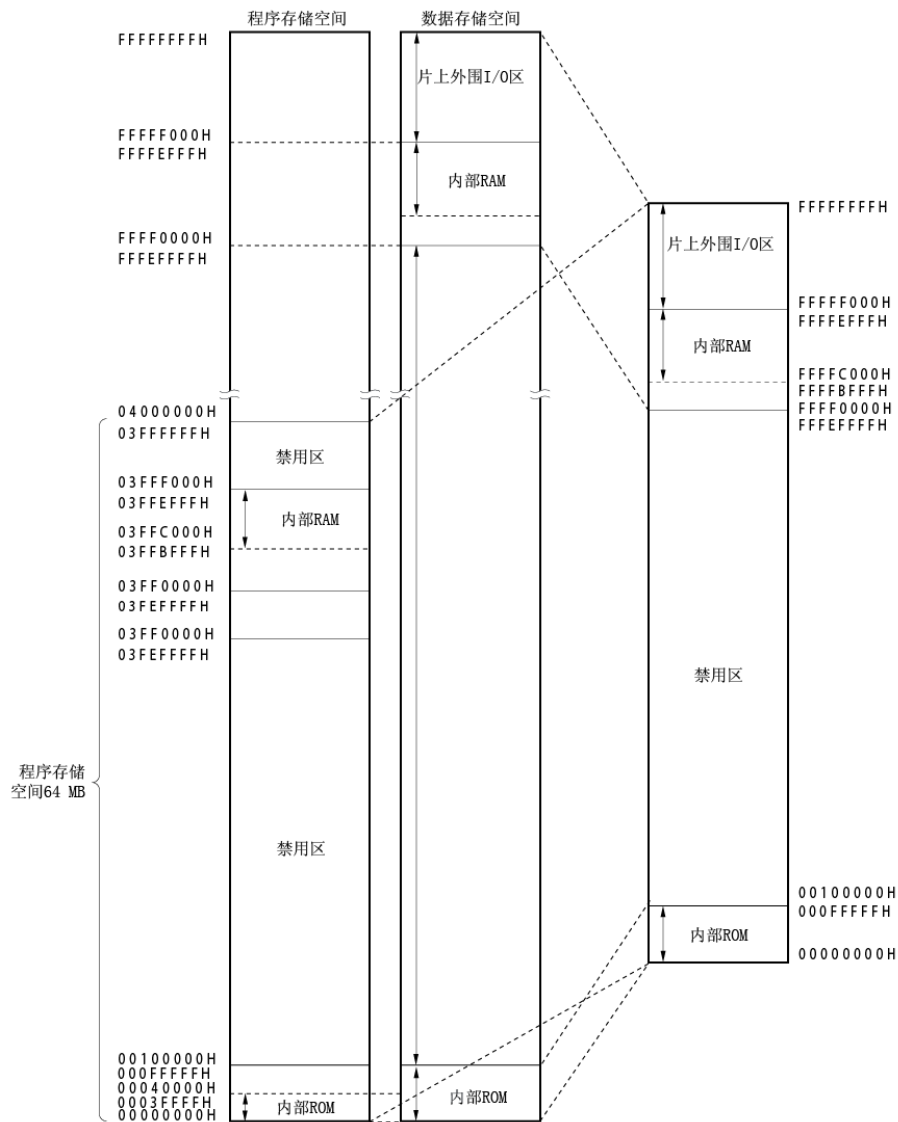


图 3-10. 建议的存储器映射图



备注

1. 表示建议区域。
2. 该图为 $\mu$ PD70F3704 的建议存储器映射图。

## 3.4.6 外围 I/O 寄存器

(1/7)

地址	功能寄存器名称	符号	R/W	可操作位			缺省值
				1	8	16	
FFFFF004H	端口 DL	PDL	R/W			√	未定义
FFFFF004H	端口 DLL	PDLL		√	√		未定义
FFFFF005H	端口 DLH	PDLH		√	√		未定义
FFFFF008H	端口 CS	PCS		√	√		未定义
FFFFF00AH	端口 CT	PCT		√	√		未定义
FFFFF00CH	端口 CM	PCM		√	√		未定义
FFFFF024H	端口模式寄存器 DL	PMDL				√	FFFFH
FFFFF024H	端口模式寄存器 DLL	PMDLL		√	√		FFH
FFFFF025H	端口模式寄存器 DLH	PMDLH		√	√		FFH
FFFFF028H	端口模式寄存器 CS	PMCS		√	√		FFH
FFFFF02AH	端口模式寄存器 CT	PMCT		√	√		FFH
FFFFF02CH	端口模式寄存器 CM	PMCM		√	√		FFH
FFFFF04CH	端口模式控制寄存器 CM	PMCCM		√	√		00H
FFFFF06EH	系统等待控制寄存器	VSWC			√		77H
FFFFF100H	中断屏蔽寄存器 0	IMR0				√	FFFFH
FFFFF100H	中断屏蔽寄存器 0L	IMR0L		√	√		FFH
FFFFF101H	中断屏蔽寄存器 0H	IMR0H		√	√		FFH
FFFFF102H	中断屏蔽寄存器 1	IMR1				√	FFFFH
FFFFF102H	中断屏蔽寄存器 1L	IMR1L		√	√		FFH
FFFFF103H	中断屏蔽寄存器 1H	IMR1H		√	√		FFH
FFFFF104H	中断屏蔽寄存器 2	IMR2				√	FFFFH
FFFFF104H	中断屏蔽寄存器 2L	IMR2L		√	√		FFH
FFFFF105H	中断屏蔽寄存器 2H	IMR2H		√	√		FFH
FFFFF110H	中断控制寄存器	LVIIIC		√	√		47H
FFFFF112H	中断控制寄存器	PIC0		√	√		47H
FFFFF114H	中断控制寄存器	PIC1		√	√		47H
FFFFF116H	中断控制寄存器	PIC2		√	√		47H
FFFFF118H	中断控制寄存器	PIC3		√	√		47H
FFFFF11AH	中断控制寄存器	PIC4		√	√		47H
FFFFF11CH	中断控制寄存器	PIC5		√	√		47H
FFFFF11EH	中断控制寄存器	PIC6		√	√		47H
FFFFF120H	中断控制寄存器	PIC7		√	√		47H
FFFFF122H	中断控制寄存器	TQ0OVIC		√	√		47H
FFFFF124H	中断控制寄存器	TQ0CCIC0		√	√		47H
FFFFF126H	中断控制寄存器	TQ0CCIC1		√	√		47H
FFFFF128H	中断控制寄存器	TQ0CCIC2		√	√		47H
FFFFF12AH	中断控制寄存器	TQ0CCIC3		√	√		47H
FFFFF12CH	中断控制寄存器	TP0OVIC		√	√		47H

(2/7)

地址	功能寄存器名称	符号	R/W	可操作位			缺省值
				1	8	16	
FFFFF12EH	中断控制寄存器	TP0CCIC0	R/W	√	√		47H
FFFFF130H	中断控制寄存器	TP0CCIC1		√	√		47H
FFFFF132H	中断控制寄存器	TP1OVIC		√	√		47H
FFFFF134H	中断控制寄存器	TP1CCIC0		√	√		47H
FFFFF136H	中断控制寄存器	TP1CCIC1		√	√		47H
FFFFF138H	中断控制寄存器	TP2OVIC		√	√		47H
FFFFF13AH	中断控制寄存器	TP2CCIC0		√	√		47H
FFFFF13CH	中断控制寄存器	TP2CCIC1		√	√		47H
FFFFF13EH	中断控制寄存器	TP3OVIC		√	√		47H
FFFFF140H	中断控制寄存器	TP3CCIC0		√	√		47H
FFFFF142H	中断控制寄存器	TP3CCIC1		√	√		47H
FFFFF144H	中断控制寄存器	TM0EQIC0		√	√		47H
FFFFF146H	中断控制寄存器	CB0RIC		√	√		47H
FFFFF148H	中断控制寄存器	CB0TIC		√	√		47H
FFFFF14AH	中断控制寄存器	CB1RIC		√	√		47H
FFFFF14CH	中断控制寄存器	CB1TIC		√	√		47H
FFFFF14EH	中断控制寄存器	UA0RIC		√	√		47H
FFFFF150H	中断控制寄存器	UA0TIC		√	√		47H
FFFFF152H	中断控制寄存器	UA1RIC		√	√		47H
FFFFF154H	中断控制寄存器	UA1TIC		√	√		47H
FFFFF156H	中断控制寄存器	ADIC		√	√		47H
FFFFF160H	中断控制寄存器	KRIC		√	√		47H
FFFFF162H	中断控制寄存器	WTIC		√	√		47H
FFFFF164H	中断控制寄存器	WTIC		√	√		47H
FFFFF1FAH	中断服务优先权寄存器	ISPR	R	√	√		00H
FFFFF1FCH	命令寄存器	PRCMD	W		√		未定义
FFFFF1FEH	省电控制寄存器	PSC	R/W	√	√		00H
FFFFF200H	A/D 转换器模式寄存器 0	ADA0M0		√	√		00H
FFFFF201H	A/D 转换器模式寄存器 1	ADA0M1		√	√		00H
FFFFF202H	A/D 转换器通道指定寄存器 0	ADA0S		√	√		00H
FFFFF203H	A/D 转换器模式寄存器 2	ADA0M2		√	√		00H
FFFFF204H	掉电比较模式寄存器	ADA0PFM		√	√		00H
FFFFF205H	掉电比较门限寄存器	ADA0PFT		√	√		00H
FFFFF210H	A/D 转换结果寄存器 0	ADA0CR0				√	未定义
FFFFF211H	A/D 转换结果寄存器 0H	ADA0CR0H	R		√		未定义
FFFFF212H	A/D 转换结果寄存器 1	ADA0CR1				√	未定义
FFFFF213H	A/D 转换结果寄存器 1H	ADA0CR1H			√		未定义
FFFFF214H	A/D 转换结果寄存器 2	ADA0CR2				√	未定义
FFFFF215H	A/D 转换结果寄存器 2H	ADA0CR2H			√		未定义

(3/7)

地址	功能寄存器名称	符号	R/W	可操作位			缺省值
				1	8	16	
FFFFF216H	A/D 转换结果寄存器 3	ADA0CR3	R			√	未定义
FFFFF217H	A/D 转换结果寄存器 3H	ADA0CR3H			√		未定义
FFFFF218H	A/D 转换结果寄存器 4	ADA0CR4				√	未定义
FFFFF219H	A/D 转换结果寄存器 4H	ADA0CR4H			√		未定义
FFFFF21AH	A/D 转换结果寄存器 5	ADA0CR5	R/W			√	未定义
FFFFF21BH	A/D 转换结果寄存器 5H	ADA0CR5H			√		未定义
FFFFF21CH	A/D 转换结果寄存器 6	ADA0CR6				√	未定义
FFFFF21DH	A/D 转换结果寄存器 6H	ADA0CR6H			√		未定义
FFFFF21EH	A/D 转换结果寄存器 7	ADA0CR7				√	未定义
FFFFF21FH	A/D 转换结果寄存器 7H	ADA0CR7H			√		未定义
FFFFF220H	A/D 转换结果寄存器 8	ADA0CR8				√	未定义
FFFFF221H	A/D 转换结果寄存器 8H	ADA0CR8H			√		未定义
FFFFF222H	A/D 转换结果寄存器 9	ADA0CR9				√	未定义
FFFFF223H	A/D 转换结果寄存器 9H	ADA0CR9H			√		未定义
FFFFF224H	A/D 转换结果寄存器 10	ADA0CR10				√	未定义
FFFFF225H	A/D 转换结果寄存器 10H	ADA0CR10H			√		未定义
FFFFF226H	A/D 转换结果寄存器 11	ADA0CR11				√	未定义
FFFFF227H	A/D 转换结果寄存器 11H	ADA0CR11H			√		未定义
FFFFF300H	按键返回模式寄存器	KRM		√	√		00H
FFFFF308H	选择器操作控制寄存器 0	SELCNT0		√	√		00H
FFFFF318H	噪音消除控制寄存器	NFC		√	√		00H
FFFFF400H	端口 0	P0		√	√		未定义
FFFFF406H	端口 3	P3				√	未定义
FFFFF406H	端口 3L	P3L		√	√		未定义
FFFFF407H	端口 3H	P3H		√	√		未定义
FFFFF408H	端口 4	P4		√	√		未定义
FFFFF40AH	端口 5	P5		√	√		未定义
FFFFF40EH	端口 7L	P7L		√	√		未定义
FFFFF40FH	端口 7H	P7H		√	√		未定义
FFFFF412H	端口 9	P9				√	未定义
FFFFF412H	端口 9L	P9L		√	√		未定义
FFFFF413H	端口 9H	P9H		√	√		未定义
FFFFF420H	端口模式寄存器 0	PM0		√	√		FFH
FFFFF426H	端口模式寄存器 3	PM3				√	FFFFH
FFFFF426H	端口模式寄存器 3L	PM3L		√	√		FFH
FFFFF427H	端口模式寄存器 3H	PM3H		√	√		FFH
FFFFF428H	端口模式寄存器 4	PM4		√	√		FFH
FFFFF42AH	端口模式寄存器 5	PM5		√	√		FFH
FFFFF42EH	端口模式寄存器 7L	PM7L		√	√		FFH
FFFFF42FH	端口模式寄存器 7H	PM7H		√	√		FFH

(4/7)

地址	功能寄存器名称	符号	R/W	可操作位			缺省值
				1	8	16	
FFFFF432H	端口模式寄存器 9	PM9	R/W			√	FFFFH
FFFFF432H	端口模式寄存器 9L	PM9L		√	√		FFH
FFFFF433H	端口模式寄存器 9H	PM9H		√	√		FFH
FFFFF440H	端口模式控制寄存器 0	PMC0		√	√		00H
FFFFF446H	端口模式控制寄存器 3L	PMC3L		√	√		00H
FFFFF448H	端口模式控制寄存器 4	PMC4		√	√		00H
FFFFF44AH	端口模式控制寄存器 5	PMC5		√	√		00H
FFFFF452H	端口模式控制寄存器 9	PMC9				√	0000H
FFFFF452H	端口模式控制寄存器 9L	PMC9L		√	√		00H
FFFFF453H	端口模式控制寄存器 9H	PMC9H		√	√		00H
FFFFF460H	端口功能控制寄存器 0	PFC0		√	√		00H
FFFFF466H	端口功能控制寄存器 3L	PFC3L		√	√		00H
FFFFF46AH	端口功能控制寄存器 5	PFC5		√	√		00H
FFFFF472H	端口功能控制寄存器 9	PFC9				√	0000H
FFFFF472H	端口功能控制寄存器 9L	PFC9L		√	√		00H
FFFFF473H	端口功能控制寄存器 9H	PFC9H		√	√		00H
FFFFF540H	TMQ0 控制寄存器 0	TQ0CTL0		√	√		00H
FFFFF541H	TMQ0 控制寄存器 1	TQ0CTL1		√	√		00H
FFFFF542H	TMQ0 I/O 控制寄存器 0	TQ0IOC0		√	√		00H
FFFFF543H	TMQ0 I/O 控制寄存器 1	TQ0IOC1		√	√		00H
FFFFF544H	TMQ0 I/O 控制寄存器 2	TQ0IOC2		√	√		00H
FFFFF545H	TMQ0 选择寄存器 0	TQ0OPT0		√	√		00H
FFFFF546H	TMQ0 捕获/比较寄存器 0	TQ0CCR0				√	0000H
FFFFF548H	TMQ0 捕获/比较寄存器 1	TQ0CCR1				√	0000H
FFFFF54AH	TMQ0 捕获/比较寄存器 2	TQ0CCR2				√	0000H
FFFFF54CH	TMQ0 捕获/比较寄存器 3	TQ0CCR3				√	0000H
FFFFF54EH	TMQ0 计数器读缓冲寄存器	TQ0CNT	R			√	0000H
FFFFF590H	TMP0 控制寄存器 0	TP0CTL0	R/W	√	√		00H
FFFFF591H	TMP0 控制寄存器 1	TP0CTL1		√	√		00H
FFFFF592H	TMP0 I/O 控制寄存器 0	TP0IOC0		√	√		00H
FFFFF593H	TMP0 I/O 控制寄存器 1	TP0IOC1		√	√		00H
FFFFF594H	TMP0 I/O 控制寄存器 2	TP0IOC2		√	√		00H
FFFFF595H	TMP0 选择寄存器 0	TP0OPT0		√	√		00H
FFFFF596H	TMP0 捕获/比较寄存器 0	TP0CCR0				√	0000H
FFFFF598H	TMP0 捕获/比较寄存器 1	TP0CCR1				√	0000H
FFFFF59AH	TMP0 计数器读缓冲寄存器	TP0CNT				√	0000H
FFFFF5A0H	TMP1 控制寄存器 0	TP1CTL0	R/W	√	√		00H
FFFFF5A1H	TMP1 控制寄存器 1	TP1CTL1		√	√		00H
FFFFF5A2H	TMP1 I/O 控制寄存器 0	TP1IOC0		√	√		00H
FFFFF5A3H	TMP1 I/O 控制寄存器 1	TP1IOC1		√	√		00H

(5/7)

地址	功能寄存器名称	符号	R/W	可操作位			缺省值
				1	8	16	
FFFFF5A4H	TMP1 I/O 控制寄存器 2	TP1IOC2	R/W	√	√		00H
FFFFF5A5H	TMP1 选择寄存器 0	TP1OPT0		√	√		00H
FFFFF5A6H	TMP1 捕获/比较寄存器 0	TP1CCR0				√	0000H
FFFFF5A8H	TMP1 捕获/比较寄存器 1	TP1CCR1				√	0000H
FFFFF5AAH	TMP1 计数器读缓冲寄存器	TP1CNT	R			√	0000H
FFFFF5B0H	TMP2 控制寄存器 0	TP2CTL0	R/W	√	√		00H
FFFFF5B1H	TMP2 控制寄存器 1	TP2CTL1		√	√		00H
FFFFF5B2H	TMP2 I/O 控制寄存器 0	TP2IOC0		√	√		00H
FFFFF5B3H	TMP2 I/O 控制寄存器 1	TP2IOC1		√	√		00H
FFFFF5B4H	TMP2 I/O 控制寄存器 2	TP2IOC2		√	√		00H
FFFFF5B5H	TMP2 选择寄存器 0	TP2OPT0		√	√		00H
FFFFF5B6H	TMP2 捕获/比较寄存器 0	TP2CCR0				√	0000H
FFFFF5B8H	TMP2 捕获/比较寄存器 1	TP2CCR1				√	0000H
FFFFF5BAH	TMP2 计数器读缓冲寄存器	TP2CNT				√	0000H
FFFFF5C0H	TMP3 控制寄存器 0	TP3CTL0	R/W	√	√		00H
FFFFF5C1H	TMP3 控制寄存器 1	TP3CTL1		√	√		00H
FFFFF5C2H	TMP3 I/O 控制寄存器 0	TP3IOC0		√	√		00H
FFFFF5C3H	TMP3 I/O 控制寄存器 1	TP3IOC1		√	√		00H
FFFFF5C4H	TMP3 I/O 控制寄存器 2	TP3IOC2		√	√		00H
FFFFF5C5H	TMP3 选择寄存器 0	TP3OPT0		√	√		00H
FFFFF5C6H	TMP3 捕获/比较寄存器 0	TP3CCR0				√	0000H
FFFFF5C8H	TMP3 捕获/比较寄存器 1	TP3CCR1				√	0000H
FFFFF5CAH	TMP3 计数器读缓冲寄存器	TP3CNT				√	0000H
FFFFF680H	钟表定时器操作模式寄存器	WTM	R/W	√	√		00H
FFFFF690H	TMM0 控制寄存器 0	TM0CTL0		√	√		00H
FFFFF694H	TMM0 比较寄存器 0	TM0CMP0				√	0000H
FFFFF6C0H	振荡稳定时间选择寄存器	OSTS			√		06H
FFFFF6C1H	PLL 锁定时间设定寄存器	PLLS			√		03H
FFFFF6D0H	看门狗定时器模式寄存器 2	WDTM2		√	√		67H
FFFFF6D1H	看门狗定时器启动寄存器	WDTE			√		9AH
FFFFF706H	端口功能控制扩展寄存器 3L	PFCE3L		√	√		00H
FFFFF70AH	端口功能控制扩展寄存器 5	PFCE5		√	√		00H
FFFFF712H	端口功能控制扩展寄存器 9	PFCE9				√	0000H
FFFFF712H	端口功能控制扩展寄存器 9L	PFCE9L		√	√		00H
FFFFF713H	端口功能控制扩展寄存器 9H	PFCE9H		√	√		00H
FFFFF802H	系统状态寄存器	SYS		√	√		00H
FFFFF80CH	内部振荡模式寄存器	RCM		√	√		00H
FFFFF820H	省电模式寄存器	PSMR		√	√		00H
FFFFF824H	锁定寄存器	LOCKR	R	√	√		00H

(6/7)

地址	功能寄存器名称	符号	R/W	可操作位			缺省值
				1	8	16	
FFFFF828H	处理器时钟控制寄存器	PCC	R/W	√	√		03H
FFFFF82CH	PLL 控制寄存器	PLLCTL		√	√		01H
FFFFF82EH	CPU 操作时钟状态寄存器	CCLS	R	√	√		00H
FFFFF82FH	可编程时钟模式寄存器	PCLM	R/W	√	√		00H
FFFFF870H	时钟监视器模式寄存器	CLM		√	√		00H
FFFFF888H	复位源标志寄存器	RESF		√	√		00H
FFFFF890H	低电压检测寄存器	LVIM		√	√		00H
FFFFF891H	低电压检测等级选择寄存器	LVIS			√		00H
FFFFF892H	内部RAM 数据状态寄存器	RAMS		√	√		01H
FFFFF8B0H	预分频器模式寄存器 0	PRSM0			√		00H
FFFFF8B1H	预分频比较寄存器 0	PRSCM0			√		00H
FFFFF9FCH	片上调试模式寄存器	OCDM		√	√		01H
FFFFF9FEH	外围仿真寄存器 1	PEMU1		√	√		00H
FFFFFA00H	UARTA0 控制寄存器 0	UA0CTL0		√	√		10H
FFFFFA01H	UARTA0 控制寄存器 1	UA0CTL1			√		00H
FFFFFA02H	UARTA0 控制寄存器 2	UA0CTL2			√		FFH
FFFFFA03H	UARTA0 选项 控制寄存器 0	UA0OPT0		√	√		14H
FFFFFA04H	UARTA0 状态寄存器	UA0STR		√	√		00H
FFFFFA06H	UARTA0接收数据寄存器	UA0RX	R		√		FFH
FFFFFA07H	UARTA0发送数据寄存器	UA0TX	R/W		√		FFH
FFFFFA10H	UARTA1 控制寄存器 0	UA1CTL0		√	√		10H
FFFFFA11H	UARTA1 控制寄存器 1	UA1CTL1			√		00H
FFFFFA12H	UARTA1 控制寄存器 2	UA1CTL2			√		FFH
FFFFFA13H	UARTA1 选项控制寄存器 0	UA1OPT0		√	√		14H
FFFFFA14H	UARTA1 状态寄存器	UA1STR		√	√		00H
FFFFFA16H	UARTA1接收数据寄存器	UA1RX	R		√		FFH
FFFFFA17H	UARTA1发送数据寄存器	UA1TX	R/W		√		FFH
FFFFFB00H	TIP00 引脚噪音消除控制寄存器	P00NFC		√	√		00H
FFFFFB04H	TIP01 引脚噪音消除控制寄存器	P01NFC		√	√		00H
FFFFFB08H	TIP10 引脚噪音消除控制寄存器	P10NFC		√	√		00H
FFFFFB0CH	TIP11 引脚噪音消除控制寄存器	P11NFC		√	√		00H
FFFFFB10H	TIP20 引脚噪音消除控制寄存器	P20NFC		√	√		00H
FFFFFB14H	TIP21 引脚噪音消除控制寄存器	P21NFC		√	√		00H
FFFFFB18H	TIP30 引脚噪音消除控制寄存器	P30NFC		√	√		00H
FFFFFB1CH	TIP31 引脚噪音消除控制寄存器	P31NFC		√	√		00H
FFFFFB50H	TIQ00 引脚噪音消除控制寄存器	Q00NFC		√	√		00H
FFFFFB54H	TIQ01 引脚噪音消除控制寄存器	Q01NFC		√	√		00H
FFFFFB58H	TIQ02 引脚噪音消除控制寄存器	Q02NFC		√	√		00H
FFFFFB5CH	TIQ03 引脚噪音消除控制寄存器	Q03NFC		√	√		00H

注意事项 要了解 OCDM 寄存器的详细情况，请参阅第 24 章 片上调试功能。



(7/7)

地址	功能寄存器名称	符号	R/W	可操作位			缺省值
				1	8	16	
FFFFFC00H	外部中断下降沿设定寄存器 0	INTF0	R/W	√	√		00H
FFFFFC06H	外部中断下降沿设定寄存器 3L	INTF3L		√	√		00H
FFFFFC13H	外部中断下降沿设定寄存器 9H	INTF9H		√	√		00H
FFFFFC20H	外部中断上升沿设定寄存器 0	INTR0		√	√		00H
FFFFFC26H	外部中断上升沿设定寄存器 3L	INTR3L		√	√		00H
FFFFFC33H	外部中断上升沿设定寄存器 9H	INTR9H		√	√		00H
FFFFFC40H	上拉电阻选择寄存器 0	PU0		√	√		00H
FFFFFC46H	上拉电阻选择寄存器 3	PU3				√	0000H
FFFFFC46H	上拉电阻选择寄存器 3L	PU3L		√	√		00H
FFFFFC47H	上拉电阻选择寄存器 3H	PU3H		√	√		00H
FFFFFC48H	上拉电阻选择寄存器 4	PU4		√	√		00H
FFFFFC4AH	上拉电阻选择寄存器 5	PU5		√	√		00H
FFFFFC52H	上拉电阻选择寄存器 9	PU9				√	0000H
FFFFFC52H	上拉电阻选择寄存器 9L	PU9L		√	√		00H
FFFFFC53H	上拉电阻选择寄存器 9H	PU9H		√	√		00H
FFFFFD00H	CSIB0 控制寄存器 0	CB0CTL0		√	√		01H
FFFFFD01H	CSIB0 控制寄存器 1	CB0CTL1		√	√		00H
FFFFFD02H	CSIB0 控制寄存器 2	CB0CTL2			√		00H
FFFFFD03H	CSIB0 状态寄存器	CB0STR		√	√		00H
FFFFFD04H	CSIB0 接收数据寄存器	CB0RX	R			√	0000H
FFFFFD04H	CSIB0 接收数据寄存器 L	CB0RXL			√		00H
FFFFFD06H	CSIB0 发送数据寄存器	CB0TX	R/W			√	0000H
FFFFFD06H	CSIB0 发送数据寄存器 L	CB0TXL			√		00H
FFFFFD10H	CSIB1 控制寄存器 0	CB1CTL0		√	√		01H
FFFFFD11H	CSIB1 控制寄存器 1	CB1CTL1		√	√		00H
FFFFFD12H	CSIB1 控制寄存器 2	CB1CTL2			√		00H
FFFFFD13H	CSIB1 状态寄存器	CB1STR		√	√		00H
FFFFFD14H	CSIB1 接收数据寄存器	CB1RX	R			√	0000H
FFFFFD14H	CSIB1 接收数据寄存器 L	CB1RXL			√		00H
FFFFFD16H	CSIB1 发送数据寄存器	CB1TX	R/W			√	0000H
FFFFFD16H	CSIB1 发送数据寄存器 L	CB1TXL			√		00H

### 3.4.7 特殊功能寄存器

特殊功能寄存器是那些在有意外程序循环发生时防止被非法数据写入的寄存器。V850ES/HF2 微控制器中有以下 7 个特殊功能寄存器。

- 省电控制寄存器 (PSC)
- 处理器时钟控制寄存器 (PCC)
- 时钟监测模式寄存器 (CLM)
- 复位源标志寄存器 (RESF)
- 低电压检测寄存器 (LVIM)
- 内部 RAM 数据状态寄存器 (RAMS)
- 片上调试模式寄存器 (OCDM)

另外，配备了 **PRCDM** 寄存器用以保护对特殊功能寄存器的写入访问，进而保证应用系统在有意外程序循环发生时不会意外停止。对特殊功能寄存器的写入访问是按照特定顺序进行的，而且非法存储操作会通过 **SYS** 寄存器进行报告。(即使对由于噪音、瞬时压降等因素引起的非法选择数据(地址：007AH)的读操作，也会进行报告。)。

**(1) 特殊功能寄存器数据的设置**

按照如下顺序对特殊功能寄存器进行数据设置。

- <1> 将向特殊寄存器设置的数据放入通用寄存器中。
- <2> 将在第<1>步准备好的数据写入 PRCMD 寄存器中。
- <3> 将设置数据写入特殊寄存器中(通过使用如下指令完成)。
  - 存储指令(ST/SST 指令)
  - 位操作指令 (SET1/CLR1/NOT1 指令)
- <4>到<8> 插入 NOP 指令 (5 条)<sup>※</sup>。

**[描述示例] 当使用 PSC 寄存器(设置待机模式)时**

```

      ST.B r11, PSMR[r0]      ; 设置 PSMR 寄存器 (设定 IDLE 及 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※                    ; 空操作指令
<5> NOP※                    ; 空操作指令
<6> NOP※                    ; 空操作指令
<7> NOP※                    ; 空操作指令
<8> NOP※                    ; 空操作指令
(下一条指令)
```

对特殊寄存器的读操作没有特殊的顺序要求。

**注** 当切换到 IDLE1 模式，IDLE2 模式，STOP 模式或子 IDLE 模式(通过设置 PSC.STP 位为 1 实现)时，在切换执行后，必需立即插入 5 条 NOP 指令。

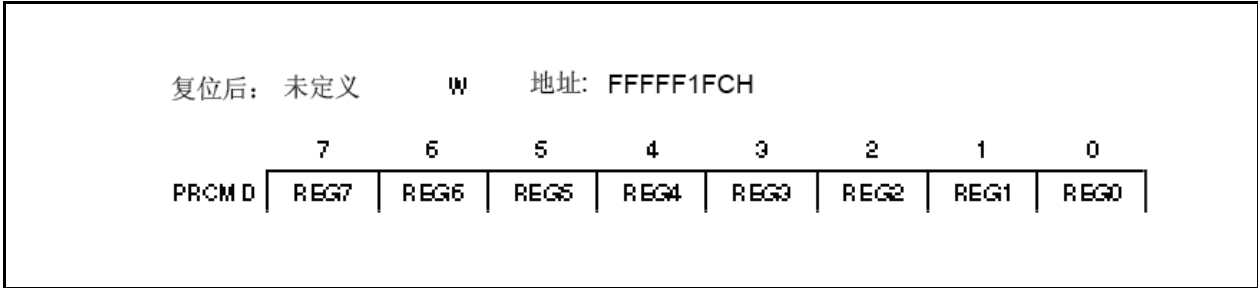
- 注意事项**
1. 当执行存储指令向命令寄存器存储数据的时候，中断会得不到应答。因为假定在以上的第<2> 和 <3>步执行的是连续的存储指令，如果在第<2> 和 <3>步之间有另外一条指令，并假设该指令对中断进行了应答，那么就无法建立上述顺序，从而导致故障出现。
  2. 虽然向 PRCMD 寄存器写入的是空数据，依然用在设置特殊寄存器时所用的那个通用寄存器(例中的第<3>步)来向 PRCMD 寄存器(例中的第<2>步)来写入数据。当通用寄存器用于寻址时，同样使用这种用法。

(2) 命令寄存器 (PRCMD)

PRCMD 寄存器是一个 8 位寄存器，用以保护那些可能对应用系统具有严重影响的寄存器免受非法写入，进而保证系统在有意外程序循环发生时不至于意外停止。注意，在数据提前写入 PRCMD 寄存器后，对特殊寄存器的第一个写访问是有效的。所以，可以通过这种方式，仅以一种特殊的顺序可以对该特殊寄存器进行重写，进而保护该寄存器不会被非法写入。

PRCMD 寄存器是只写寄存器，以 8 位进行写访问。(当对该寄存器执行读访问时，读出的是未定义的数据)。

系统复位后，该寄存器内容为未定义值。



(3) 系统状态寄存器 (SYS)

表示整个系统的操作状态的状态标志都分配给了该寄存器。  
该寄存器可以以 8 位或 1 位进行读或写操作。  
复位输入将该寄存器设置为 00H。

复位后: 00H R/		W	地址: FFFFF802H					
	7	6	5	4	3	2	1	0
SYS	0	0	0	0	0	0	0	PRERR
PRERR		保护错误检测						
0		未出现保护错误						
1		出现保护错误						

PRERR 标志的操作是按如下条件进行的。

- (a) 设置条件 (PRERR 标志 = 1)
- (i) 当向特殊寄存器写入数据而未向 PRCMD 寄存器写入任何数据时(在 3.4.7 (1) 设置特殊寄存器数据中当执行了第 <3>步而未执行第 <2>时)
  - (ii) 在 PRCMD 寄存器数据写入(如果在 3.4.7 (1) 设置特殊寄存器数据第 <3>步没有设置特殊寄存器)之后，当数据写入到片上外围 I/O 寄存器而不是写入特殊寄存器(包括位操作指令的执行)时

备注        即使在写 PRCMD 寄存器操作和写特殊寄存器操作之间，访问了内部 RAM，比如，读取了片上外围 I/O 寄存器(由位操作指令执行的读操作除外)，PRERR 标志也不置位，而且设置数据可以写入该特殊寄存器。

- (b) 清零条件 (PRERR 标志 = 0)
- (i) 当将 0 写入 PRERR 标志位时
  - (ii) 当系统复位时

- 注意事项 1. 如果将 0 写入 SYS 寄存器的 PRERR 位，但是该寄存器并不是特殊寄存器，那么，在对 PRCMD 寄存器进行写访问之后，PRERR 位就立即清为 0 (写访问优先)。
2. 如果将数据写入 PRCMD 寄存器，但是该寄存器并不是特殊寄存器，那么，在对 PRCMD 寄存器进行写访问之后，PRERR 位就立即设置为 1。

### 3.4.8 注意事项

#### (1) 需要先行设置的寄存器

当使用 V850ES/HF2 微控制器时，必需确保首先对下列寄存器进行设置。

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

在 VSWC, OCDM 和 WDTM2 寄存器设置后，如果有必要的话，才对其它寄存器进行设置。

当使用外部总线时，在上述寄存器设置完成之后，通过使用端口相关寄存器，将每个引脚设置为复用功能总线控制引脚模式。

#### (a) 系统等待控制寄存器 (VSWC)

VSWC 寄存器控制访问片上外围 I/O 寄存器的总线等待周期。

访问片上外围 I/O 寄存器需要三个时钟周期(没有等待周期)。根据操作频率的不同，V850ES/HF2 要求不同的等待周期。所以根据所使用的时钟频率，设置 VSWC 寄存器为以下各值。

VSWC 寄存器以 8 位单元进行读或写 (地址：FFFF06EH, 缺省值：77H)。

操作频率 (f <sub>CLK</sub> )	VSWC 的设置值	等待时钟周期数目
32 kHz ≤ f <sub>CLK</sub> < 16.6 MHz	00H	0 (没有等待时钟周期)
16.6 MHz ≤ f <sub>CLK</sub> ≤ 20 MHz	01H	1

#### (b) 片上调试模式寄存器 (OCDM)

要了解详细情况，请参见 第 24 章 片上调试功能。

#### (c) 看门狗定时器模式寄存器 2 (WDTM2)

WDTM2 寄存器用以设置溢出时间和看门狗定时器 2 的操作时钟。

在复位模式下，当复位释放后看门狗定时器 2 会自动启动。通过写 WDTM2 寄存器以激活此操作。

要了解详细情况，请参见 第十章 看门狗定时器 2 功能。

**(2) 访问特殊片上外围 I/O 寄存器**

本产品有两种类型的内部系统总线。

一种是 CPU 总线，另外一种是外围总线，用作与低速外围硬件的接口。

CPU 总线时钟与外围总线时钟是异步的，因此，如果访问 CPU 和访问外围硬件发生冲突的时候，就有可能传输无法预料的非法数据。所以，如果存在这种冲突的可能，当访问外围硬件时，就需要改变访问 CPU 的时钟周期数，以便传输正确的数据。这样，CPU 就不会启动处理下一条指令而是进入等待状态。当这种等待状态发生时，执行一条指令所需的时钟数就会增加如下所示的等待时钟数目。

在要求实时处理的应用过程中，必需要考虑这种情况。

当访问特殊片上外围 I/O 寄存器时，除了 VSWC 寄存器中设置的等待状态之外，还需要更多的等待状态。

访问条件以及如何计算所要插入的等待状态数目(CPU 时钟数)如下表所示。

外围功能	寄存器名称	访问类型	k
16 位定时器/事件计数器 P (TMP) (n = 0 到 3)	TPnCNT	读	1 或 2
	TPnCCR0, TPnCCR1	写	<ul style="list-style-type: none"> <li>第一次访问: 没有等待</li> <li>连续写: 3 或 4</li> </ul>
		读	1 或 2
16 位定时器/事件计数器 Q (TMQ)	TQ0CNT	读	1 或 2
	TQ0CCR0 到 TQ0CCR3	写	<ul style="list-style-type: none"> <li>第一次访问: 没有等待</li> <li>连续写: 3 或 4</li> </ul>
		读	1 或 2
看门狗定时器 2 (WDT2)	WDTM2	写 (当 WDT2 运行时)	3
A/D 转换器	ADA0M0	读	1 或 2
	ADA0CR0 到 ADA0CR11	读	1 或 2
	ADA0CR0H 到 ADA0CR11H	读	1 或 2

访问所需要的时钟数 =  $3 + i + j + (2 + j) \times k$

**注意事项** 在下面两种情况下，禁止访问上述寄存器。如果产生了等待周期，该等待周期只有通过复位来清除。

- CPU 在副时钟下运行且主时钟停止振荡。
- CPU 在内部振荡时钟下运行

**备注** i: VSWC 寄存器高 4 位之值(0 或 1)

j: VSWC 寄存器低 4 位之值(0 或 1)

**(3) sld 指令和中断请求冲突的限制****(a) 描述**

如果在 **sld** 指令之前就立即对<2>中所列指令（该指令紧接在<1>中所列指令之后）进行译码，同时在<1>中指令完成执行之前，有中断请求冲突发生，那么在<1>中的指令执行结果就有可能无法储存寄存器中去。

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**
- 乘法指令:                      **mul, mulh, mulhi, mulu**

指令 <2>

<b>mov reg1, reg2</b>	<b>not reg1, reg2</b>	<b>satsubr reg1, reg2</b>	<b>satsub reg1, reg2</b>
<b>satadd reg1, reg2</b>	<b>satadd imm5, reg2</b>	<b>or reg1, reg2</b>	<b>xor reg1, reg2</b>
<b>and reg1, reg2</b>	<b>tst reg1, reg2</b>	<b>subr reg1, reg2</b>	<b>sub reg1, reg2</b>
<b>add reg1, reg2</b>	<b>add imm5, reg2</b>	<b>cmp reg1, reg2</b>	<b>cmp imm5, reg2</b>
<b>mulh reg1, reg2</b>	<b>shr imm5, reg2</b>	<b>sar imm5, reg2</b>	<b>shl imm5, reg2</b>

<示例>

<i> **ld.w [r11], r10**    如果<iii>中 **sld** 指令之前就立即对<ii>中的 **mov** 指令进行译码，同时在<i>中 **ld** 指令执行完毕之前出现中断请求冲突，那么，<i>中指令执行的结果就有可能储存不到寄存器中去。

:

:

:

<ii> **mov r10, r28**

<iii> **sld.w 0x28, r10**

**(b) 对策**

<1> 当使用编译器 (CA850) 时

请使用 **CA850 Ver. 2.61** 或更新版本，因为相应指令序列的生成可以被自动挂起。

<2> 对于汇编编译器之对策

在指令<ii>之后就立即执行 **sld** 指令，可以采用以下任一种方法来避免上述操作。

- 立即在 **sld** 指令之前插入一 **nop** 指令。
- 不使用 **sld** 上一条指令<ii>指令中所用的寄存器来作为 **sld** 指令的目的寄存器。



## 第四章 端口功能

### 4.1 特性

- I/O 端口: 67
- 端口引脚另外用作其他外围功能 I/O 引脚
- 可用 1-位单元设置成输入或输出模式。

### 4.2 基本端口配置

V850ES/HF2 具有总共 67 个输入/输出端口，端口 0, 3 到 5, 7, 9, CM, CS, CT 和 DL.端口配置如下所示。

图 4-1.端口配置

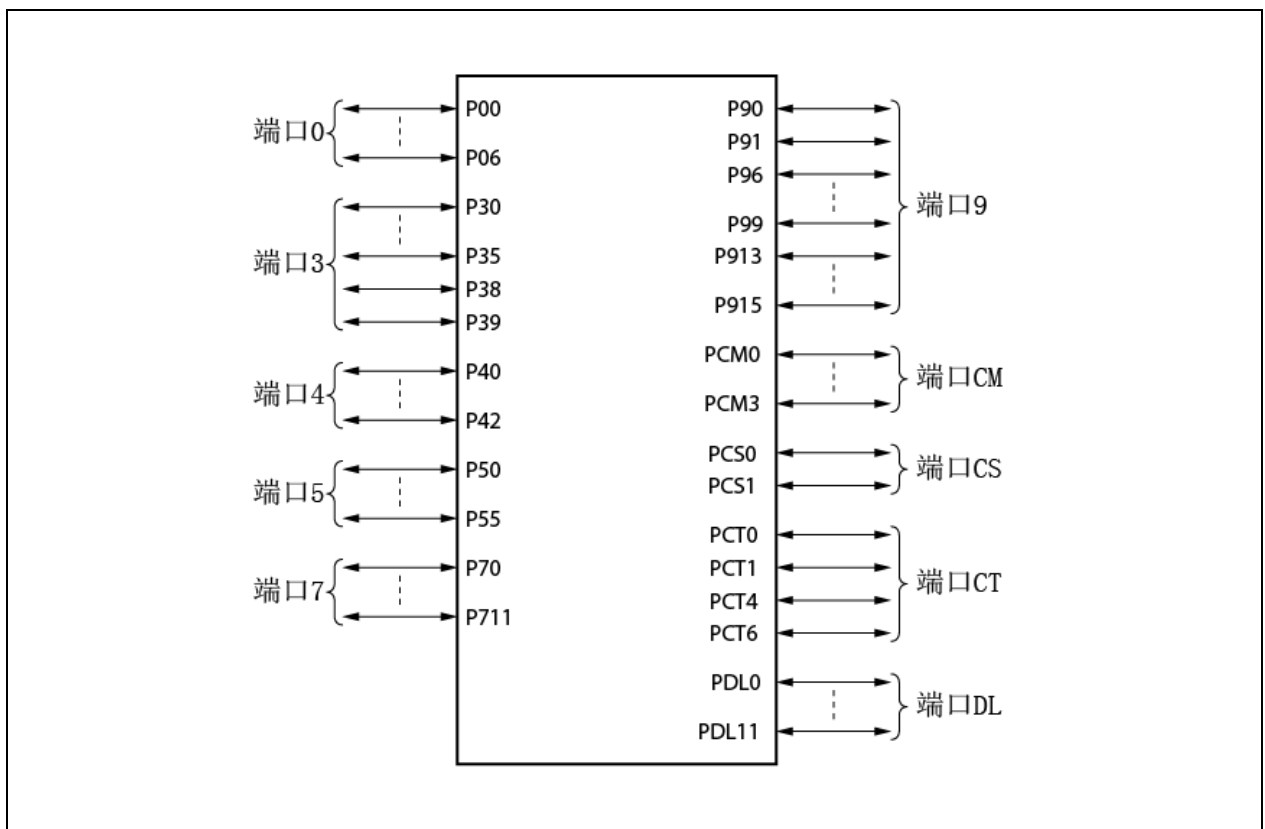


表 4-1. 端口配置

项目	配置
控制寄存器	端口模式寄存器(PMn: n = 0, 3 到 5, 7L, 7H, 9, CM, CS, CT 或 DL)
	端口模式控制寄存器(PMCn: n = 0, 3L, 4, 5, 9 或 CM)
	端口功能控制寄存器(PFCn: n = 0, 3L, 5 或 9)
	端口功能控制扩展寄存器(PFCEn: n = 3L, 5 或 9)
	上拉电阻选项寄存器(PUn: n = 0, 3 到 5 或 9)
端口	67

表 4-2. 引脚 I/O 缓冲器电源

电源	对应引脚
AV <sub>REF0</sub>	端口 7
EV <sub>DD</sub>	端口 0, 3 到 5, 9, CM, CS, CT, DL, RESET

## 4.3 端口功能

### 4.3.1 端口功能操作

端口操作根据输入或输出模式设置而有所不同，如下所示。

#### (1) 写入 I/O 端口

##### (a) 在输出模式下

值使用转移指令可写入输出锁存器。输出锁存器的内容从引脚输出。一旦数据写入输出锁存器，其保留到新数据写入输出锁存器。

##### (b) 在输入模式下

值使用转移指令可写入输出锁存器。然而，因为输出缓冲器处于关状态，所以引脚状态保持不变。一旦数据写入输出锁存器，其保留到新数据写入输出锁存器。

**注意事项** 尽管 1-位内存操作指令处理 1 位，其按 8-位单元访问端口。因此，如果端口包含输入和输出引脚，在输入模式下即使引脚未进行操作，也未定义输出锁存器引脚设置的内容。

#### (2) 从 I/O 端口读取

##### (a) 在输出模式下

输出锁存器的内容可使用转移指令读取。输出锁存器的内容保持不变。

##### (b) 在输入模式下

引脚状态可使用转移指令读取。输出锁存器的内容保持不变。

#### (3) I/O 端口操作

##### (a) 在输出模式下

对输出锁存器的内容进行操作且结果写入输出锁存器。输出锁存器的内容从引脚输出。一旦数据写入输出锁存器，其保留到新数据写入输出锁存器。

##### (b) 在输入模式下

输出锁存器的内容未定义。然而，因为输出缓冲器处于关状态，所以引脚状态保持不变。

**注意事项** 尽管 1-位内存操作指令处理 1 位，但是其按 8-位单元访问端口。如果端口包含输入和输出引脚，因此，即使引脚未进行操作，在输入模式下也未定义输出锁存器引脚设置的内容。

#### 4.3.2 设置端口引脚的注意事项

- (1) 端口号和复用功能根据产品而有所不同。将寄存器有关不可用端口和复用功能设置为复位后的值。
- (2) 使用以下程序设置寄存器端口。

- <1> 设置端口功能控制寄存器  $n(\text{PFCn})$  和端口功能控制扩展寄存器  $n(\text{PFCEn})$ 。
- <2> 设置端口模式控制寄存器  $n(\text{PMCn})$ 。
- <3> 设置外部中断下降沿规范寄存器  $n(\text{INTFn})$  和外部中断上升沿规范寄存器  $n(\text{INTRn})$ 。

如果设置  $\text{PMCn}$  寄存器之后设置  $\text{PFCn}$  和  $\text{PFCEn}$  寄存器，则当设置  $\text{PFCn}$  和  $\text{PFCEn}$  寄存器时可能设置其他外围功能引脚。

- (3)  $\text{PUn}$  寄存器的  $\text{PUnm}$  位(其连接片上上拉电阻)仅在输入模式下( $\text{PMn}$  寄存器的  $\text{PMnm}$  位= 1)有效。在输出模式下( $\text{PMn}$  寄存器的  $\text{PMnm}$  位= 0)，片上上拉寄存器由硬件断开。
- (4) 读取引脚电平和端口锁存器由端口模式寄存器( $\text{PMn}$ )控制。使用复用功能时方法相同。
- (5) 当在端口模式下读取时施密特( $\text{SHMT}$ )-触发输入缓冲器不作为  $\text{SHMT}$  缓冲器工作。

### 4.3.3 端口 0

端口 0 为 7-位端口(P00 到 P06)，I/O 设置可通过 1-位单元控制。

#### (1) 端口 0 功能

- 端口输入/输出数据可由 1-位单元指定。  
由端口寄存器 0(P0)指定
- 端口输入/输出模式可由 1-位单元指定。  
由端口模式寄存器 0(PM0)指定
- 端口模式或控制模式(复用功能)可由 1-位单元指定。  
由端口模式控制寄存器 0(PMC0)指定。
- 控制模式 1 或控制模式 2 可由 1-位单元指定。  
由端口功能控制寄存器 0(PFC0)指定
- 片上上拉电阻可由 1-位单元连接。  
由上拉电阻选项寄存器 0(PU0)指定

端口 0 另外用作以下引脚。

表 4-3. 端口 0 复用功能引脚

引脚名称	引脚编号	复用-功能引脚名称	I/O	备注	块类型
P00	3	TP31/TOP31	I/O	—	G-1
P01	4	TP30/TOP30			G-1
P02	5	NMI <sup>注 1</sup>			L-1
P03	6	INTP0/ADTRG			N-1
P04	7	INTP1			L-1
P05	17	INTP2/ $\overline{\text{DRST}}$ <sup>注 2</sup>			AA-1
P06	18	INTP3			L-2

- 注**
1. NMI 引脚另外用作 P02 引脚。其复位后用作 P02 引脚。  
要启用 NMI 引脚，将 PMC0.PMC02 位设置为 1。NMI 引脚的初始设置为“未检测到边缘”。使用 INTF0 和 INTR0 寄存器选择 NMI 引脚有效边缘。
  2. P05 引脚的复用功能为片上调试功能。外部复位后，P05/INTP2/ $\overline{\text{DRST}}$  引脚初始化为片上调试引脚( $\overline{\text{DRST}}$ )。要将 P05 引脚用作端口引脚，而不用作片上调试引脚，必须采取以下操作。

<1> OCDM.OCDM0 位(专用寄存器)清零。

<2> 将 P05/INTP2/ $\overline{\text{DRST}}$  引脚固定为低电平直到采取以下动作。

当未使用片上调试功能时，采取以上动作之前将高电平输入到  $\overline{\text{DRST}}$  引脚可能发生故障(CPU 死锁)。  
处理 P05 引脚时必须非常小心。

当未将高电平输入到 P05/INTP2/ $\overline{\text{DRST}}$  引脚(当此引脚固定为低电平)时，无需操作 OCDM.OCDM0 位。  
因为下拉电阻(30 k $\Omega$  TYP.)连接到缓冲器的 P05/INTP2/ $\overline{\text{DRST}}$  引脚，引脚无需通过外部电源固定为低电平。下拉电阻通过 OCDM0 位清零断开。

注意事项 P00 到 P06 引脚复用功能在输入模式下具有滞后特性，而在端口模式下不具有滞后特性。

(2) 寄存器

(a) 端口寄存器 0(P0)

端口寄存器 0(P0)为控制读取引脚电平和写入输出电平的 8-位寄存器。此寄存器可按 8-位或 1-位单元读取或写入。

复位后：未定义		R/W	地址：FFFFF400H					
	7	6	5	4	3	2	1	0
P0	0	P06	P05	P04	P03	P02	P01	P00

P0n	输出数据控制(在输出模式下)(n = 0 到 6)
0	输出 0.
1	输出 1.

(b) 端口模式寄存器 0(PM0)

这是指定输入或输出模式的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

复位后：FFH		R/W	地址：FFFFF420H					
	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	输入/输出模式控制(n = 0 到 6)	
0	输出模式	
1	输入模式	

## (c) 端口模式控制寄存器 0(PMC0)

这是指定端口模式或控制模式的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

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 引脚操作模式规范
0	I/O 端口
1	INTP3 输入

PMC05	P05 引脚操作模式规范
0	I/O 端口
1	INTP2/DRST 输入

PMC04	P04 引脚操作模式规范
0	I/O 端口
1	INTP1 输入

PMC03	P03 引脚操作模式规范
0	I/O 端口
1	INTP0/ADTRG 输入

PMC02	P02 引脚操作模式规范
0	I/O 端口
1	NMI 输入

PMC01	P01 引脚操作模式规范
0	I/O 端口
1	TIP30/TOP30 I/O

PMC00	P00 引脚操作模式规范
0	I/O 端口
1	TIP31/TOP31 I/O

**注意事项** 当 OCDM.OCDM0 位为 1 时 P05/INTP2/ $\overline{\text{DRST}}$  引脚用作  $\overline{\text{DRST}}$  引脚，而不管 PMC05 位值如何。

**(d) 端口功能控制寄存器 0(PFC0)**

这是指定控制模式 1 或控制模式 2 的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

复位后: 00H

R/W

地址: FFFFF460H

	7	6	5	4	3	2	1	0
PFC0	0	0	0	0	PFC03	0	PFC01	PFC00

PFC03	当 P03 引脚处于控制模式时的操作模式规范
0	INTP0 输入
1	ADTRG 输入

PFC01	当 P01 引脚处于控制模式时的操作模式规范
0	TIP30 输入
1	TOP30 输出

PFC00	当 P00 引脚处于控制模式时的操作模式规范
0	TIP31 输入
1	TOP31 输出

**(e) 上拉电阻选项寄存器 0(PU0)**

这是指定片上上拉电阻连接的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

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	片上上拉电阻连接控制(n = 0 到 6)
0	未连接
1	连接



#### 4.3.4 端口 3

端口 3 为 8-位端口(P30 到 P35, P38, P39), I/O 设置可由 1-位单元控制。

##### (1) 端口 3 的功能

- 端口输入/输出数据可由 1-位单元指定。  
由端口寄存器 3(P3)指定
- 端口输入/输出模式可由 1-位单元指定。  
由端口模式寄存器 3(PM3)指定
- 端口模式或控制模式(复用功能)可由 1-位单元指定。  
由端口模式控制寄存器 3L(PMC3L)指定
- 控制模式可由 1-位单元指定。  
由端口功能控制寄存器 3L(PFC3L)和端口功能控制扩展寄存器 3L(PFCE3L)指定
- 片上上拉电阻可由 1-位单元连接。  
由上拉电阻选项寄存器 3(PU3)指定

端口 3 另外用作以下引脚。

表 4-4. 端口 3 的复用-功能引脚

引脚名称	引脚编号	复用-功能引脚名称	I/O	备注	块类型
P30	22	TXDA0	I/O	—	E-2
P31	23	RXDA0/INTP7			L-2
P32	24	ASCKA0/TIP00/TOP00/TOP01			U-13
P33	25	TIP01/TOP01			G-1
P34	26	TIP10/TOP10			G-1
P35	27	TIP11/TOP11			G-1
P38	28	—			C-1
P39	29	—			C-1

**注意事项** P31 到 P35 引脚复用功能在输入模式下具有滞后特性, 而在端口模式下不具有滞后特性。

## (2) 寄存器

## (a) 端口寄存器 3(P3)

端口寄存器 3(P3)为控制读取引脚电平和写入输出电平的 16-位寄存器。此寄存器可读取或写入 16-位单元。然而，如果 P3 寄存器的高 8 位用作 P3H 寄存器且低 8 位用作 P3L 寄存器，则这些寄存器可按 8-位或 1-位单元读取或写入。

复位后：未定义		R/W	地址：FFFFF406H, FFFFF407H					
P3(P3H <sup>※</sup> )	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	输出数据控制(在输出模式下)(n = 0 到 5, 8, 9)							
0	输出 0.							
1	输出 1.							

注 要按 8-位或 1-位单元读取或写入 P3 寄存器的位 8 到 15，指定这些位为 P3H 寄存器的位 0 到 7。

## (b) 端口模式寄存器 3(PM3)

这是指定输入或输出模式的 16-位寄存器。其可读取或写入 16-位单元。

然而，如果 PM3 寄存器的高 8 位用作 PM3H 寄存器且低 8 位用作 PM3L 寄存器，则这些寄存器可按 8-位或 1-位单元读取或写入。

复位后：FFFFH		R/W	地址：FFFFF426H, FFFFF427H					
PM3(PM3H <sup>※</sup> )	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 模式控制(n = 0 到 5, 8, 9)							
0	输出模式							
1	输入模式							

注 要按 8-位或 1-位单元读取或写入 PM3 寄存器的位 8 到 15，指定这些位为 PM3H 寄存器的位 0 到 7。

**(c) 端口模式控制寄存器 3L(PMC3L)**

这是指定端口模式或控制模式的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

复位后: 00H

R/W

地址: FFFFF446H

7	6	5	4	3	2	1	0	
PMC3L	0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

PMC35	P35 引脚操作模式规范
0	I/O 端口
1	TIP11/TOP11 I/O

PMC34	P34 引脚操作模式规范
0	I/O 端口
1	TIP10/TOP10 I/O

PMC33	P33 引脚操作模式规范
0	I/O 端口
1	TIP01/TOP01 I/O

PMC32	P32 引脚操作模式规范
0	I/O 端口
1	ASCKA0/TIP00/TOP00/TOP01 I/O

PMC31	P31 引脚操作模式规范
0	I/O 端口
1	RXDA0/INTP7 输入 <sup>※</sup>

PMC30	P30 引脚操作模式规范
0	I/O 端口
1	TXDA0 输出

**注** INTP7 引脚另外用作 RXDA0 引脚。要用作 RXDA0 引脚，使复用-功能 INTP7 引脚的边缘检测功能无效(通过 INTF3.INTF31 和 INTR3.INTR31 位固定为 0)。要用作 INTP7 引脚，停止 UARTA0 的接收操作(通过 UA0CTL0.UA0RXE 位清零)。

**(d) 端口功能控制寄存器 3L(PFC3L)**

这是指定控制模式 1, 2, 3 或 4 的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

复位后: 00H		R/W		地址: FFFFF466H				
	7	6	5	4	3	2	1	0
PFC3L	0	0	PFC35	PFC34	PFC33	PFC32	0	0

**备注** 对于如何指定控制模式, 请参阅 **4.3.4(2)(f)P3 引脚控制模式设置**。

**(e) 端口功能控制扩展寄存器 3L(PFCE3L)**

这是指定控制模式 1, 2, 3 或 4 的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

复位后: 00H		R/W		地址: FFFFF706H				
	7	6	5	4	3	2	1	0
PFCE3L	0	0	0	0	0	PFCE32	0	0

**备注** 对于如何指定控制模式, 请参阅 **4.3.4(2)(f)P3 引脚控制模式设置**。

**(f) P3 引脚控制模式设置**

PFC35	P35 引脚控制模式规范
0	TIP11 输入
1	TOP11 输出

PFC34	P34 引脚控制模式规范
0	TIP10 输入
1	TOP10 输出

PFC33	P33 引脚控制模式规范
0	TIP01 输入
1	TOP01 输出

PFCE32	PFC32	P32 引脚控制模式规范
0	0	ASCKA0 输入
0	1	TOP01 输出
1	0	TIP00 输入
1	1	TOP00 输出

(g) 上拉电阻选项寄存器 3(PU3)

这是指定片上上拉电阻连接的 16-位寄存器。其可读取或写入 16-位单元。  
然而，如果 PU3 寄存器的高 8 位用作 PU3H 寄存器且低 8 位用作 PU3L 寄存器，则这些寄存器可按 8-位或 1-位单元读取或写入。

复位后：0000H		R/W	地址：FFFFFC46H, FFFFFC47H					
PU3(PU3H <sup>※</sup> )	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	PU39	PU38
(PU3L)	7	6	5	4	3	2	1	0
	0	0	PU35	PU34	PU33	PU32	PU31	PU30
PU3n		片上上拉电阻连接控制(n = 0 到 5, 8, 9)						
0		未连接						
1		连接						

注      要按 8-位或 1-位单元读取/写入 PU3 寄存器的位 8 到 15,指定这些位为 PU3H 寄存器的位 0 到 7。

4.3.5 端口 4

端口 4 为 3-位端口(P40 到 P42)，I/O 设置可由 1-位单元控制。

(1) 端口 4 的功能

- 端口输入/输出数据可由 1-位单元指定。  
由端口寄存器 4(P4)指定
- 端口输入/输出模式可由 1-位单元指定。  
由端口模式寄存器 4(PM4)指定
- 端口模式或控制模式(复用功能)可由 1-位单元指定。  
由端口模式控制寄存器 4(PMC4)指定
- 片上上拉电阻可由 1-位单元连接。  
由上拉电阻选项寄存器 4(PU4)指定

端口 4 另外用作以下引脚。

表 4-5. 端口 4 复用-功能引脚

引脚名称	引脚 编号	复用-功能引脚名称	I/O	备注	块类型
P40	19	SIB0	I/O	—	E-1
P41	20	SOB0			E-2
P42	21	$\overline{\text{SCKB0}}$			E-3

注意事项 P40 和 P42 引脚复用功能在输入模式下具有滞后特性，而在端口模式下不具有滞后特性。

**(2) 寄存器****(a) 端口寄存器 4(P4)**

端口寄存器 4(P4)为控制读取引脚电平和写入输出电平的 8-位寄存器。此寄存器可按 8-位或 1-位单元读取或写入。

复位后: 未定义		R/W		地址: FFFFF408H					
		7	6	5	4	3	2	1	0
P4		0	0	0	0	0	P42	P41	P40

P4n	输出数据控制(在输出模式下)(n = 0 到 2)
0	输出 0.
1	输出 1.

**(b) 端口模式寄存器 4(PM4)**

这是指定输入或输出模式的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

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	输入/输出模式控制(n = 0 到 2)
0	输出模式
1	输入模式

(c)端口模式控制寄存器 4(PMC4)

这是指定端口模式或控制模式的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

复位后: 00H		R/W		地址: FFFFF448H				
	7	6	5	4	3	2	1	0
PMC4	0	0	0	0	0	PMC42	PMC41	PMC40

PMC42	操作模式规范 P42 引脚	
0	I/O 端口	
1	$\overline{\text{SCKB0}}$ I/O	

PMC41	操作模式规范 P41 引脚	
0	I/O 端口	
1	SOB0 输出	

PMC40	操作模式规范 P40 引脚	
0	I/O 端口	
1	SIB0 输入	

(d) 上拉电阻选项寄存器 4(PU4)

这是 8-位寄存器指定片上上拉电阻连接。其可按 8-位或 1-位单元读取或写入。

复位后: 00H		R/W		地址: FFFFC48H					
		7	6	5	4	3	2	1	0
PU4		0	0	0	0	0	PU42	PU41	PU40

PU4n	片上上拉电阻连接控制(n = 0 到 2)
0	未连接
1	连接



### 4.3.6 端口 5

端口 5 为 6-位端口(P50 到 P55)，I/O 设置可由 1-位单元控制。

#### (1) 功能 of 端口 5

- 端口输入/输出数据可由 1-位单元指定。  
由端口寄存器 5(P5) 指定
- 端口输入/输出模式可由 1-位单元指定。  
由端口模式寄存器 5(PM5) 指定
- 端口模式或控制模式(复用功能)可由 1-位单元指定。  
由端口模式控制寄存器 5(PMC5) 指定
- 控制模式可由 1-位单元指定。  
由端口功能控制寄存器 5(PFC5)或端口功能控制扩展寄存器 5(PFCE5) 指定
- 片上上拉电阻可由 1-位单元连接。  
由上拉电阻选项寄存器 5(PU5) 指定

端口 5 另外用作以下引脚。

表 4-6. 端口 5 复用-功能引脚

引脚名称	引脚编号	复用-功能引脚名称	I/O	备注	块类型
P50	32	KR0/TIQ01/TOQ01	I/O	—	U-4
P51	33	KR1/TIQ02/TOQ02			U-4
P52	34	KR2/TIQ03/TOQ03/DDI <sup>※</sup>			U-5
P53	35	KR3/TIQ00/TOQ00/DDO <sup>※</sup>			U-6
P54	36	KR4/DCK <sup>※</sup>			G-2
P55	37	KR5/DMS <sup>※</sup>			G-2

**注** DDI, DDO, DCK 和 DMS 引脚用于片上调试功能。要将 DDI, DDO, DCK 和 DMS 引脚用作端口引脚，而不用作片上调试引脚，必须采取以下动作。

- <1> OCDM 寄存器(专用寄存器) OCDM0 位清零。
- <2> 将 P05/INTP2/ $\overline{\text{DRST}}$  引脚固定为低电平直到采取以上动作。

当未使用片上调试功能时，采取以上动作之前将高电平输入到  $\overline{\text{DRST}}$  引脚可能发生故障(CPU 死锁)。处理 P05 引脚需要非常小心。

当未将高电平输入到 P05/INTP2/ $\overline{\text{DRST}}$  引脚(当此引脚固定为低电平)时，无需操作 OCDM.OCDM0 位。因为下拉电阻(30 k $\Omega$  TYP.)连接到缓冲器的 P05/INTP2/ $\overline{\text{DRST}}$  引脚，引脚无需通过外部电源固定为低电平。下拉电阻通过 OCDM0 位清零断开。

**注意事项** P50 到 P55 引脚复用功能在输入模式下具有滞后特性，而在端口模式下不具有滞后特性。

## (2) 寄存器

## (a) 端口寄存器 5 (P5)

端口寄存器 5(P5)为控制读取引脚电平和写入输出电平的 8-位寄存器。此寄存器可按 8-位或 1-位单元读取或写入。

复位后: 未定义		R/W	地址: FFFFF40AH							
			7	6	5	4	3	2	1	0
P5			0	0	P55	P54	P53	P52	P51	P50

P5n	输出数据控制(在输出模式下)(n = 0 到 5)
0	输出 0.
1	输出 1.

## (b) 端口模式寄存器 5 (PM5)

这是指定输入或输出模式的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

复位后: FFH		R/W	地址: FFFFF42AH					
	7	6	5	4	3	2	1	0
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50

PM5n	I/O 模式控制(n = 0 到 5)
0	输出模式
1	输入模式

**(c) 端口模式控制寄存器 5(PMC5)**

这是指定端口模式或控制模式的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

**注意事项** 如果当 PFC5.PFC5n 和 PFCE5.PFCE5n 位为默认值(0)时控制模式使用 PMC5 寄存器指定，则未定义输出。

出于该原因，首先设置 PFC5.PFC5n 和 PFCE5.PFCE5n 位且接着将 PMC5n 位设置为 1 以设置控制模式。

复位后: 00H

R/W

地址: FFFFF44AH

	7	6	5	4	3	2	1	0
PMC5	0	0	PMC55	PMC54	PMC53	PMC52	PMC51	PMC50

PMC55	P55 引脚操作模式规范
0	I/O 端口
1	KR5 输入

PMC54	P54 引脚操作模式规范
0	I/O 端口
1	KR4 输入

PMC53	P53 引脚操作模式规范
0	I/O 端口
1	KR3/TIQ00/TOQ00 I/O

PMC52	P52 引脚操作模式规范
0	I/O 端口
1	KR2/TIQ03/TOQ03 I/O

PMC51	P51 引脚操作模式规范
0	I/O 端口
1	KR1/TIQ02/TOQ02 I/O

PMC50	P50 引脚操作模式规范
0	I/O 端口
1	KR0/TIQ01/TOQ01 I/O

(d) 端口功能控制寄存器 5(PFC5)

这是指定控制模式 1, 2, 3 或 4 的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

复位后: 00H		R/W	地址: FFFFF46AH					
	7	6	5	4	3	2	1	0
PFC5	0	0	PFC55	PFC54	PFC53	PFC52	PFC51	PFC50

**备注**      对于如何指定控制模式, 请参阅 **4.3.6(2)(f)P5 引脚的控制模式设置**。

(e) 端口功能控制扩展寄存器 5(PFCE5)

这是指定控制模式 1, 2, 3 或 4 的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

复位后: 00H		R/W	地址: FFFFF70AH					
	7	6	5	4	3	2	1	0
PFCE5	0	0	0	0	PFCE53	PFCE52	PFCE51	PFCE50

**备注**      对于如何指定控制模式, 请参阅 **4.3.6(2)(f)P5 引脚的控制模式设置**。

(f) P5 引脚的控制模式设置

注意事项 如果当 PFC5.PFC5n 和 PFCE5.PFCE5n 位为默认值(0)时控制模式使用 PMC5 寄存器指定, 则未定义输出。  
出于该原因, 首先设置 PFC5.PFC5n 和 PFCE5.PFCE5n 位且接着将 PMC5n 位设置为 1 以设置控制模式。

PFC55	P55 引脚控制模式规范
0	禁止设置
1	KR5 输入

PFC54	P54 引脚控制模式规范
0	禁止设置
1	KR4 输入

PFCE53	PFC53	P53 引脚控制模式规范
0	0	禁止设置
0	1	TIQ00/KR3 <sup>※</sup> 输入
1	0	TOQ00 输出
1	1	禁止设置

PFCE52	PFC52	P52 引脚控制模式规范
0	0	禁止设置
0	1	TIQ03/KR2 <sup>※</sup> 输入
1	0	TOQ03 输出
1	1	禁止设置

PFCE51	PFC51	P51 引脚控制模式规范
0	0	禁止设置
0	1	TIQ02/KR1 <sup>※</sup> 输入
1	0	TOQ02 输出
1	1	禁止设置

PFCE50	PFC50	P50 引脚控制模式规范
0	0	禁止设置
0	1	TIQ01/KR0 <sup>※</sup> 输入
1	0	TOQ01 输出
1	1	禁止设置

注 KRn 引脚另外用作 TIQ0m 引脚。要将此引脚用作 TIQ0m 引脚，使复用-功能 KRn 引脚的按键中断检测功能无效(通过 KRM.KRMn 位清零)。要将此引脚用作 KRn 引脚，使复用-功能 TIQ0m 引脚的边缘检测功能无效(n = 0 到 3, m = 0 到 3)。

引脚名称	用作 TIQ0m 引脚	用作 KRn 引脚
KR0/TIQ01	KRM 寄存器的 KRM0 位= 0	TQ0IOC1 寄存器的 TQ0TIG2, TQ0TIG3 位= 0
KR1/TIQ02	KRM 寄存器的 KRM1 位= 0	TQ0IOC1 寄存器的 TQ0TIG4, TQ0TIG5 位= 0
KR2/TIQ03	KRM 寄存器的 KRM2 位= 0	TQ0IOC1 寄存器的 TQ0TIG6, TQ0TIG7 位= 0
KR3/TIQ00	KRM 寄存器的 KRM3 位= 0	TQ0IOC1 寄存器的 TQ0TIG0, TQ0TIG1 位= 0 TQ0IOC1 寄存器的 TQ0EES0, TQ0EES1 位= 0 TQ0IOC1 寄存器的 TQ0ETS0, TQ0ETS1 位= 0

(g) 上拉电阻选项寄存器 5(PU5)

这是 8-位寄存器指定片上上拉电阻连接。其可按 8-位或 1-位单元读取或写入。

复位后: 00H		R/W	地址: FFFFFFFC4AH					
	7	6	5	4	3	2	1	0
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50

PU5n	片上上拉电阻连接控制(n = 0 到 5)
0	未连接
1	连接

### 4.3.7 端口 7

端口 7 为 12-位端口(P70 到 P711)，I/O 设置可由 1-位单元控制。

#### (1) 端口 7 的功能

- 端口输入/输出数据可由 1-位单元指定。  
由端口寄存器 7H，7L(P7H，P7L) 指定
- 端口输入/输出模式可由 1-位单元指定。  
由端口模式寄存器 7H，7L(PM7H，PM7L) 指定

端口 7 另外用作以下引脚。

表 4-7. 端口 7 复用-功能引脚

引脚名称	引脚编号	复用-功能引脚名称	I/O	备注	块类型
P70	80	ANI0	I/O	—	A-1
P71	79	ANI1			A-1
P72	78	ANI2			A-1
P73	77	ANI3			A-1
P74	76	ANI4			A-1
P75	75	ANI5			A-1
P76	74	ANI6			A-1
P77	73	ANI7			A-1
P78	72	ANI8			A-1
P79	71	ANI9			A-1
P710	70	ANI10			A-1
P711	69	ANI11			A-1

## (2) 寄存器

## (a) 端口寄存器 7H, 端口寄存器 7L(P7H, P7L)

端口寄存器 7H 和 7L(P7H 和 P7L)为控制读取引脚电平和写入输出电平的 8-位寄存器。这些寄存器可按 8-位或 1-位单元读取或写入。  
它们不能以 16-位单元访问。

复位后: 未定义		R/W	地址: FFFFF40FH, FFFFF40EH					
	7	6	5	4	3	2	1	0
P7H	0	0	0	0	P711	P710	P79	P78
	7	6	5	4	3	2	1	0
P7L	P77	P76	P75	P74	P73	P72	P71	P70
P7n	输出数据控制(在输出模式下)(n = 0 到 11)							
0	输出 0.							
1	输出 1.							

**注意事项** A/D 转换期间禁止读取 P7H 和 P7L 寄存器。

## (b) 端口模式寄存器 7H, 7L(PM7H, PM7L)

这些是指定输入或输出模式的 8-位寄存器。它们可按 8-位或 1-位单元读取或写入。  
这些寄存器不能以 16-位单元访问。

复位后: FFH		R/W	地址: FFFFF42FH, FFFFF42EH					
	7	6	5	4	3	2	1	0
PM7H	1	1	1	1	PM711	PM710	PM79	PM78
	7	6	5	4	3	2	1	0
PM7L	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70
PM7n	I/O 模式控制(n = 0 到 11)							
0	输出模式							
1	输入模式							

**注意事项** 要使用 P7n(ANIn)复用功能, 设置 PM7n 为 1.



### 4.3.8 端口 9

端口 9 为 9-位端口(P90, P91, P96 到 P99, P913 到 P915), I/O 设置可由 1-位单元控制。

#### (1) 功能 of 端口 9

- 端口输入/输出数据可由 1-位单元指定。  
由端口寄存器 9(P9) 指定
- 端口输入/输出模式可由 1-位单元指定。  
由端口模式寄存器 9(PM9) 指定
- 端口模式或控制模式(复用功能)可由 1-位单元指定。  
由端口模式控制寄存器 9(PMC9) 指定
- 控制模式可由 1-位单元指定。  
由端口功能控制寄存器 9(PFC9)和端口功能控制扩展寄存器 9(PFCE9) 指定
- 片上上拉电阻可由 1-位单元连接。  
由上拉电阻选项寄存器 9(PU9) 指定

端口 9 另外用作以下引脚。

表 4-8. 端口 9 复用-功能引脚

引脚名称	引脚编号	复用-功能引脚名称	I/O	备注	块类型
P90	38	KR6/TXDA1	I/O	—	U-12
P91	39	KR7/RXDA1			U-7
P96	40	TIP21/TOP21			U-9
P97	41	SIB1/TIP20/TOP20			U-8
P98	42	SOB1			G-3
P99	43	SCKB1			G-5
P913	44	INTP4/PCL			W-1
P914	45	INTP5			N-2
P915	46	INTP6			N-2

**注意事项** P90, P91, P96, P97, P99 和 P913 到 P915 引脚复用功能在输入模式下具有滞后特性, 而在端口模式下不具有滞后特性。

## (2) 寄存器

## (a) 端口寄存器 9(P9)

端口寄存器 9(P9)为控制读取引脚电平和写入输出电平的 16-位寄存器。此寄存器可读取或写入 16-位单元。然而，如果 P9 寄存器的高 8 位用作 P9H 寄存器且低 8 位用作 P9L 寄存器，则这些寄存器可按 8-位或 1-位单元读取或写入。

复位后：未定义		R/W	地址：FFFFF412H, FFFFF413H					
	15	14	13	12	11	10	9	8
P9(P9H <sup>※</sup> )	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	输出数据控制(在输出模式下)(n = 0, 1, 6 到 9, 13 到 15)							
0	输出 0。							
1	输出 1。							

注 要读取或写入 P9 寄存器的位 8 到 15 按 8-位或 1-位单元，指定这些位为 P9H 寄存器的位 0 到 7。

## (b) 端口模式寄存器 9(PM9)

这是指定输入或输出模式的 16-位寄存器。其可读取或写入 16-位单元。

然而，如果 PM9 寄存器的高 8 位用作 PM9H 寄存器且低 8 位用作 PM9L 寄存器，则这些寄存器可按 8-位或 1-位单元读取或写入。

复位后：FFFFH		R/W		地址：FFFFF432H, FFFFF433H					
		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 模式控制(n = 0 , 1, 6 到 9, 13 到 15)							
0		输出模式							
1		输入模式							

注 要按 8-位或 1-位单元读取或写入 PM9 寄存器的位 8 到 15，指定这些位为 PM9H 寄存器的位 0 到 7。

**(c) 端口模式控制寄存器 9(PMC9)**

这是指定端口模式或控制模式的 16-位寄存器。其可读取或写入 16-位单元。

然而，如果 PMC9 寄存器的高 8 位用作 PMC9H 寄存器且低 8 位用作 PMC9L 寄存器，则这些寄存器可按 8-位或 1-位单元读取或写入。

**注意事项** 如果当 PFC9.PFC9n 位和 PFCE9.PFCE9n 位为默认值(0)时控制模式使用 PMC9 寄存器指定，则未定义输出。  
出于这一原因，首先设置 PFC9.PFC9n 位和 PFCE9.PFCE9n 位为 1 且接着设置 PMC9n 位为 1 以设置控制模式。

(1/2)

复位后: 0000H

R/W

地址: FFFFF452H, FFFFF453H

	15	14	13	12	11	10	9	8
PMC9(PMC9H <sup>※</sup> )	PMC915	PMC914	PMC913	0	0	0	PMC99	PMC98
	7	6	5	4	3	2	1	0
(PMC9L)	PMC97	PMC96	0	0	0	0	PMC91	PMC90

PMC915	P915 引脚操作模式规范
0	I/O 端口
1	INTP6 输入

PMC914	P914 引脚操作模式规范
0	I/O 端口
1	INTP5 输入

PMC913	P913 引脚操作模式规范
0	I/O 端口
1	INTP4/PCL I/O

**注** 要按 8-位或 1-位单元读取或写入 PMC9 寄存器的位 8 到 15，指定这些位为 PMC9H 寄存器的位 0 到 7。

PMC99	P99 引脚操作模式规范
0	I/O 端口
1	SCKB1 I/O

PMC98	P98 引脚操作模式规范
0	I/O 端口
1	SOB1 输出

PMC97	P97 引脚操作模式规范
0	I/O 端口
1	SIB1/TIP20/TOP20 I/O

PMC96	P96 引脚操作模式规范
0	I/O 端口
1	TIP21/TOP21 I/O

PMC91	P91 引脚操作模式规范
0	I/O 端口
1	KR7/RXDA1 输入

PMC90	P90 引脚操作模式规范
0	I/O 端口
1	KR6/TXDA1 I/O

**(d) 端口功能控制寄存器 9(PFC9)**

这是指定控制模式 1, 2, 3 或 4 的 16-位寄存器。其可读取或写入 16-位单元。

然而, 如果 PFC9 寄存器的高 8 位用作 PFC9H 寄存器且低 8 位用作 PFC9L 寄存器, 则这些寄存器可按 8-位或 1-位单元读取或写入。

复位后: 0000H		R/W		地址: FFFFF472H, FFFFF473H				
	15	14	13	12	11	10	9	8
PFC9(PFC9H <sup>※</sup> )	PFC915	PFC914	PFC913	0	0	0	PFC99	PFC98
	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	0	0	0	0	PFC91	PFC90

**注** 要按 8-位或 1-位单元读取或写入 PFC9 寄存器的位 8 到 15, 指定这些位为 PFC9H 寄存器的位 0 到 7。

**备注** 对于如何指定控制模式, 请参阅 4.3.8(2)(f)P9 引脚控制模式设置。

**(e) 端口功能控制扩展寄存器 9(PFCE9)**

这是指定控制模式 1, 2, 3 或 4 的 16-位寄存器。其可读取或写入 16-位单元。

然而, 如果 PFC9 寄存器的高 8 位用作 PFC9H 寄存器且低 8 位用作 PFC9L 寄存器, 则这些寄存器可按 8-位或 1-位单元读取或写入。

复位后: 0000H		R/W	地址: FFFFF712H, FFFFF713H							
			15	14	13	12	11	10	9	8
PFCE9(PFCE9H <sup>※</sup> )			0	0	PFCE913	0	0	0	0	0
			7	6	5	4	3	2	1	0
(PFCE9L)			PFCE97	PFCE96	0	0	0	0	PFCE91	PFCE90

**注** 要按 8-位或 1-位单元读取或写入 PFCE9 寄存器的位 8 到 15, 指定这些位为 PFCE9H 寄存器的位 0 到 7。

**备注** 对于如何指定控制模式, 请参阅 4.3.8(2)(f)P9 引脚控制模式设置。

## (f) P9 引脚控制模式设置

**注意事项** 如果当 PFC9.PFC9n 和 PFCE9.PFCE9n 位为默认值(0)时控制模式使用 PMC9 寄存器指定, 则未定义输出。

出于这一原因, 首先设置 PFC9.PFC9n 和 PFCE9.PFCE9n 位且接着设置 PMC9n 位为 1 以设置控制模式。

PFC915	P915 引脚控制模式规范
0	禁止设置
1	INTP6 输入

PFC914	P914 引脚控制模式规范
0	禁止设置
1	INTP5 输入

PFCE913	PFC913	P913 引脚控制模式规范
0	0	禁止设置
0	1	INTP4 输入
1	0	PCL 输出
1	1	禁止设置

PFC99	P99 引脚控制模式规范
0	禁止设置
1	$\overline{\text{SCKB1}}$ I/O

PFC98	P98 引脚控制模式规范
0	禁止设置
1	SOB1 输出

PFCE97	PFC97	P97 引脚控制模式规范
0	0	禁止设置
0	1	SIB1 输入
1	0	TIP20 输入
1	1	TOP20 输出

PFCE96	PFC96	P96 引脚控制模式规范
0	0	禁止设置
0	1	禁止设置
1	0	TIP21 输入
1	1	TOP21 输出

PFCE91	PFC91	P91 引脚控制模式规范
0	0	禁止设置
0	1	KR7 输入
1	0	KR7/RXDA1 输入 <sup>※</sup>
1	1	禁止设置

PFCE90	PFC90	P90 引脚控制模式规范
0	0	禁止设置
0	1	KR6 输入
1	0	TXDA1 输出
1	1	禁止设置

注 KR7 引脚和 RXDA1 引脚为复用-功能引脚。

当将该引脚用作 RXDA1 引脚时，禁用 KR7 引脚按键中断检测。(KRM7 位 KRM 寄存器清零。) 同样，当将该引脚用作 KR7 引脚，建议设置 PFC91 位为 1 且 PFCE91 位清零。

(g) 上拉电阻选项寄存器 9(PU9)

这是指定片上上拉电阻连接的 16-位寄存器。其可读取或写入 16-位单元。  
然而，如果 PU9 寄存器的高 8 位用作 PU9H 寄存器且低 8 位用作 PU9L 寄存器，则这些寄存器可按 8-位或 1-位单元读取或写入。

复位后: 0000H		R/W	地址: FFFFC52H, FFFFC53H					
PU9(PU9H <sup>※</sup> )	15	14	13	12	11	10	9	8
	PU915	PU914	PU913	0	0	0	PU99	PU98
(PU9L)	7	6	5	4	3	2	1	0
	PU97	PU96	0	0	0	0	PU91	PU90
PU9n		片上上拉电阻连接控制(n = 0, 1, 6 到 9, 13 到 15)						
0		未连接						
1		连接						

注 要按 8-位或 1-位单元读取/写入 PU9 寄存器的位 8 到 15,指定这些位为 PU9H 寄存器的位 0 到 7。



#### 4.3.9 端口 CM

端口 CM 为 4-位端口(PCM0 到 PCM3)，I/O 设置可由 1-位单元控制。

##### (1) 端口 CM 的功能

- 端口输入/输出数据可由 1-位单元指定。  
由端口寄存器 CM(PCM) 指定
- 端口输入/输出模式可由 1-位单元指定。  
由端口模式寄存器 CM(PMCM) 指定
- 端口模式或控制模式(复用功能)可由 1-位单元指定。  
由端口模式控制寄存器 CM(PMCCM) 指定

端口 CM 另外用作以下引脚。

表 4-9. 端口 CM 复用-功能引脚

引脚名称	引脚编号	复用-功能引脚名称	I/O	备注	块类型
PCM0	49	—	I/O	—	B-1
PCM1	50	CLKOUT			D-2
PCM2	51	—			B-1
PCM3	52	—			B-1

## (2) 寄存器

## (a) 端口寄存器 CM(PCM)

端口寄存器 CM(PCM)为控制读取引脚电平和写入输出电平的 8-位寄存器。此寄存器可按 8-位或 1-位单元读取或写入。

复位后: 未定义		R/W		地址: FFFFF00CH				
	7	6	5	4	3	2	1	0
PCM	0	0	0	0	PCM3	PCM2	PCM1	PCM0

PCMn	输出数据控制(在输出模式下)(n = 0 到 3)
0	输出 0.
1	输出 1.

## (b) 端口模式寄存器 CM(PMCM)

这是指定输入或输出模式的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

复位后: FFH		R/W		地址: FFFFF02CH				
	7	6	5	4	3	2	1	0
PMCM	1	1	1	1	PMCM3	PMCM2	PMCM1	PMCM0

PMCMn	I/O 模式控制(n = 0 到 3)
0	输出模式
1	输入模式

## (c) 端口模式控制寄存器 CM(PMCCM)

这是指定端口模式或控制模式的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

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 引脚操作模式规范						
0	I/O 端口						
1	CLKOUT 输出						

**注意事项** 确保设置位 7 为 2 和 0 为“0”。

#### 4.3.10 端口 CS

端口 CS 为 2-位端口(PCS0, PCS1)，I/O 设置可由 1-位单元控制。

##### (1) 功能 of 端口 CS

- 端口输入/输出数据可由 1-位单元指定。  
由端口寄存器 CS(PCS) 指定
- 端口输入/输出模式可由 1-位单元指定。  
由端口模式寄存器 CS(PMCS) 指定

端口 CS 另外用作以下引脚。

表 4-10. 端口 CS 复用-功能引脚

引脚名称	引脚编号	复用-功能引脚名称	I/O	备注	块类型
PCS0	47	—	I/O	—	B-1
PCS1	48	—			B-1

## (2) 寄存器

## (a) 端口寄存器 CS(PCS)

端口寄存器 CS(PCS)为控制读取引脚电平和写入输出电平的 8-位寄存器。此寄存器可按 8-位或 1-位单元读取或写入。

复位后：未定义		R/W		地址：FFFFFF008H					
		7	6	5	4	3	2	1	0
PCS		0	0	0	0	0	0	PCS1	PCS0

PCS <sub>n</sub>	输出数据控制(在输出模式下)(n = 0, 1)
0	输出 0.
1	输出 1.

## (b) 端口模式寄存器 CS(PMCS)

这是指定输入或输出模式的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

复位后：FFH		R/W		地址：FFFFF028H				
	7	6	5	4	3	2	1	0
PMCS	0	0	0	0	0	0	PMCS1	PMCS0

PMCSn	I/O 模式控制(n = 0, 1)
0	输出模式
1	输入模式

#### 4.3.11 端口 CT

端口 CT 为 4-位端口(PCT0, PCT1, PCT4, PCT6), I/O 设置可由 1-位单元控制。

##### (1) 端口 CT 的功能

- 端口输入/输出数据可由 1-位单元指定。  
由端口寄存器 CT(PCT) 指定
- 端口输入/输出模式可由 1-位单元指定。  
由端口模式寄存器 CT(PMCT) 指定

端口 CT 另外用作以下引脚。

表 4-11. 端口 CT 复用-功能引脚

引脚名称	引脚编号	复用-功能引脚名称	I/O	备注	块类型
PCT0	53	—	I/O	—	B-1
PCT1	54	—			B-1
PCT4	55	—			B-1
PCT6	56	—			B-1

(2) 寄存器

(a) 端口寄存器 CT(PCT)

端口寄存器 CT(PCT)为控制读取引脚电平和写入输出电平的 8-位寄存器。此寄存器可按 8-位或 1-位单元读取或写入。

复位后：未定义		R/W	地址：FFFFF00AH					
PCT	7	6	5	4	3	2	1	0
	0	PCT6	0	PCT4	0	0	PCT1	PCT0
PCTn		输出数据控制(在输出模式下)(n = 0, 1, 4, 6)						
0		输出 0.						
1		输出 1.						

(b) 端口模式寄存器 CT(PMCT)

这是指定输入或输出模式的 8-位寄存器。其可按 8-位或 1-位单元读取或写入。

复位后：FFH		R/W	地址：FFFFF02AH					
PMCT	7	6	5	4	3	2	1	0
	1	PMCT6	1	PMCT4	1	1	PMCT1	PMCT0
PMCTn		I/O 模式控制(n = 0, 1, 4, 6)						
0		输出模式						
1		输入模式						

#### 4.3.12 端口 DL

端口 DL 为 12-位端口(PDL0 到 PDL11)，I/O 设置可由 1-位单元控制。

##### (1) 端口 DL 的功能

- 端口输入/输出数据可由 1-位单元指定。  
由端口寄存器 DL(PDL) 指定
- 端口输入/输出模式可由 1-位单元指定。  
由端口模式寄存器 DL(PMDL) 指定

端口 DL 另外用作以下引脚。

表 4-12. 端口 DL 复用-功能引脚

引脚名称	引脚编号	复用-功能引脚名称	I/O	备注	块类型
PDL0	57	—	I/O	—	B-1
PDL1	58	—			B-1
PDL2	59	—			B-1
PDL3	60	—			B-1
PDL4	61	(			B-1
PDL5	62	FLMD1 注			B-1
PDL6	63	(			B-1
PDL7	64	(			B-1
PDL8	65	(			B-1
PDL9	66	(			B-1
PDL10	67	(			B-1
PDL11	68	(			B-1

注 因为 FLMD1 引脚用于 flash 编程模式，其无需使用端口控制寄存器操作。如需详细信息，请参阅第二十二章 闪存。

## (2) 寄存器

## (a) 端口寄存器 DL(PDL)

端口寄存器 DL(PDL)为控制读取引脚电平和写入输出电平的 16-位寄存器。此寄存器可读取或写入 16-位单元。

然而，如果 PDL 寄存器的高 8 位用作 PDLH 寄存器且低 8 位用作 PDL 寄存器，则这些寄存器可按 8-位或 1-位单元读取或写入。

复位后：未定义		R/W		地址：FFFFF004H, FFFFF005H				
	15	14	13	12	11	10	9	8
PDL(PDLH <sup>※</sup> )	0	0	0	0	PDL11	PDL10	PDL9	PDL8
	7	6	5	4	3	2	1	0
(PDL)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0
PDLn	输出数据控制(在输出模式下)(n = 0 到 11)							
0	输出 0.							
1	输出 1.							

**注** 要读取或写入位 8 到 15 PDL 寄存器按 8-位或 1-位单元，指定这些位为位 0 到 7 PDLH 寄存器。

## (b) 端口模式寄存器 DL(PMDL)

这是指定输入或输出模式的 16-位寄存器。其可读取或写入 16-位单元。

然而，如果 PMDL 寄存器的高 8 位用作 PMDLH 寄存器且低 8 位用作 PMDL 寄存器，则这些寄存器可按 8-位或 1-位单元读取或写入。

复位后：FFFFH		R/W		地址：FFFFF024H, FFFFF025H				
	15	14	13	12	11	10	9	8
PMDL(PMDLH <sup>※</sup> )	1	1	1	1	PMDL11	PMDL10	PMDL9	PMDL8
	7	6	5	4	3	2	1	0
(PMDLL)	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0
PMDLn	I/O 模式控制(n = 0 到 11)							
0	输出模式							
1	输入模式							

**注** 要按 8-位或 1-位单元读取或写入 PMDL 寄存器的位 8 到 15，指定这些位为 PMDLH 寄存器的位 0 到 7。



#### 4.3.13 另外用作片上调试功能的端口引脚

表 4-13 所示引脚另外用作片上调试引脚。外部复位后，这些引脚初始化为片上调试引脚( $\overline{\text{DRST}}$ ，DDI，DDO，DCK 和 DMS)。

表 4-13 片上调试引脚

引脚名称	复用功能引脚
P05	INTP2/ $\overline{\text{DRST}}$
P52	KR2/TIQ03/TOQ03/DDI
P53	KR3/TIQ00/TOQ00/DDO
P54	KR4/DCK
P55	KR5/DMS

要将这些引脚用作端口引脚，而不用作片上调试引脚，外部复位后必须采取以下动作。

- <1> OCDM 寄存器(专用寄存器)的 OCDM0 位清零。
- <2> 将 P05/INTP2/ $\overline{\text{DRST}}$  引脚固定为低电平直到以上动作采取。

当未使用片上调试功能时，采取以上动作之前将高电平输入到  $\overline{\text{DRST}}$  引脚可能发生故障(CPU 死锁)。处理 P05 引脚需要非常小心。

当未将高电平输入到 P05/INTP2/ $\overline{\text{DRST}}$  引脚(当此引脚固定为低电平)时，无需操作 OCDM.OCDM0 位。

因为下拉电阻(30 k $\Omega$  TYP.)连接到缓冲器的 P05/INTP2/ $\overline{\text{DRST}}$  引脚，引脚无需由外部电源固定为低电平。下拉电阻通过 OCDM0 位清零断开。

如需详细信息，请参阅 第二十四章 片上调试功能。

## 4.3.14 要将端口引脚用作复用-功能引脚的寄存器设置

表 4-14. 端口引脚用作复用-功能引脚(1/4)

引脚名称	复用-功能引脚		PMn 寄存器	PMCn 寄存器	PFCm 寄存器	PFCEm 寄存器	其他位(寄存器)
	名称	I/O					
P00	TIP31	输入	无需设置	PMC00 = 1	PFC00 = 0	—	
	TOP31	输出	无需设置	PMC00 = 1	PFC00 = 1	—	
P01	TIP30	输入	无需设置	PMC01 = 1	PFC01 = 0	—	
	TOP30	输出	无需设置	PMC01 = 1	PFC01 = 1	—	
P02	NMI	输入	无需设置	PMC02 = 1	—	—	
P03	INTP0	输入	无需设置	PMC03 = 1	PFC03 = 0	—	INTx03(INTx0)
	ADTRG	输出	无需设置	PMC03 = 1	PFC03 = 1	—	
P04	INTP1	输入	无需设置	PMC04 = 1	—	—	INTx04(INTx0)
P05 <sup>注 1</sup>	INTP2	输入	无需设置	PMC05 = 1	—	—	INTx05(INTx0)
	DRST	输入	无需设置	无需设置	—	—	OCDM0(OCDM)= 1
P06	INTP3	输入	无需设置	PMC06 = 1	—	—	INTx06(INTx0)
P30	TXDA0	输出	无需设置	PMC30 = 1	—	—	
P31	RXDA0	输入	无需设置	PMC31 = 1	—	—	注 2
	INTP7	输入	无需设置	PMC31 = 1	—	—	注 2, INTx31(INTx3)
P32	ASCKA0	输入	无需设置	PMC32 = 1	PFC32 = 0	PFCE32 = 0	
	TOP01	输出	无需设置	PMC32 = 1	PFC32 = 1	PFCE32 = 0	
	TIP00	输入	无需设置	PMC32 = 1	PFC32 = 0	PFCE32 = 1	
	TOP00	输出	无需设置	PMC32 = 1	PFC32 = 1	PFCE32 = 1	
P33	TIP01	输入	无需设置	PMC33 = 1	PFC33 = 0	—	
	TOP01	输出	无需设置	PMC33 = 1	PFC33 = 1	—	
P34	TIP10	输入	无需设置	PMC34 = 1	PFC34 = 0	—	
	TOP10	输出	无需设置	PMC34 = 1	PFC34 = 1	—	
P35	TIP11	输入	无需设置	PMC35 = 1	PFC35 = 0	—	
	TOP11	输出	无需设置	PMC35 = 1	PFC35 = 1	—	
P40	SIB0	输入	无需设置	PMC40 = 1	—	—	
P41	SOB0	输出	无需设置	PMC41 = 1	—	—	
P42	SCKB0	I/O	无需设置	PMC42 = 1	—	—	

- 注
1. 外部复位后, P05/INTP2/DRST 引脚初始化为片上调试引脚(DRST)。禁止将 P05/INTP2/DRST 引脚用作片上调试引脚, 请参阅 第二十四章 片上调试功能。
  2. INTP7 引脚另外用作 RXDA0 引脚。要将此引脚用作 RXDA0 引脚, 使复用-功能 INTP7 引脚的边缘检测功能无效(通过 INTF3.INTF31 位和 INTR3.INTR31 位清零)。要将此引脚用作 INTP7 引脚, 停止 UARTA0 的接收操作(通过 UA0CTL0.UA0RXE 位清零)。

- 备注
1. 当使用复用功能时无需设置端口寄存器(Pn)。
  2. INTxn = INTFn, INTRn

表 4-14. 端口引脚用作复用-功能引脚(2/4)

引脚名称	复用-功能引脚		PMn 寄存器	PMCn 寄存器	PFCm 寄存器	PFCEm 寄存器	其他位(寄存器)
	名称	I/O					
P50	KR0	输入	无需设置	PMC50 = 1	PFC50 = 1	PFCE50 = 0	注 1
	TIQ01	输入	无需设置	PMC50 = 1	PFC50 = 1	PFCE50 = 0	注 1
	TOQ01	输出	无需设置	PMC50 = 1	PFC50 = 0	PFCE50 = 1	
P51	KR1	输入	无需设置	PMC51 = 1	PFC51 = 1	PFCE51 = 0	注 1
	TIQ02	输入	无需设置	PMC51 = 1	PFC51 = 1	PFCE51 = 0	注 1
	TOQ02	输出	无需设置	PMC51 = 1	PFC51 = 0	PFCE51 = 1	
P52	KR2	输入	无需设置	PMC52 = 1	PFC52 = 1	PFCE52 = 0	注 1
	TIQ03	输入	无需设置	PMC52 = 1	PFC52 = 1	PFCE52 = 0	注 1
	TOQ03	输出	无需设置	PMC52 = 1	PFC52 = 0	PFCE52 = 1	
	DDI <sup>注 2</sup>	输入	无需设置	无需设置	无需设置	无需设置	OCDM0(OCDM)= 1
P53	KR3	输入	无需设置	PMC53 = 1	PFC53 = 1	PFCE53 = 0	注 1
	TIQ00	输入	无需设置	PMC53 = 1	PFC53 = 1	PFCE53 = 0	注 1
	TOQ00	输出	无需设置	PMC53 = 1	PFC53 = 0	PFCE53 = 1	
	DDO <sup>注 2</sup>	输出	无需设置	无需设置	无需设置	无需设置	OCDM0(OCDM)= 1
P54	KR4	输入	无需设置	PMC54 = 1	PFC54 = 1	—	
	DCK <sup>注 2</sup>	输出	无需设置	无需设置	无需设置	—	OCDM0(OCDM)= 1
P55	KR5	输入	无需设置	PMC55 = 1	PFC55 = 1	—	
	DMS <sup>注 2</sup>	输出	无需设置	无需设置	无需设置	—	OCDM0(OCDM)= 1

注 1. KRn 引脚另外用作 TIQ0m 引脚。要将此引脚用作 TIQ0m 引脚，使复用-功能 KRn 引脚的按键中断检测功能无效(通过 KRM.KRMn 位清零)。要将此引脚用作 KRn 引脚，使复用-功能 TIQ0m 引脚的边缘检测功能无效(n = 0 到 3, m = 0 到 3)。

引脚名称	当用作 TIQ0m 引脚	当用作 KRn 引脚
KR0/TIQ01	KRM 寄存器的 KRM0 位= 0	TQ0IOC1 寄存器的 TQ0TIG2, TQ0TIG3 位= 0
KR1/TIQ02	KRM 寄存器的 KRM1 位= 0	TQ0IOC1 寄存器的 TQ0TIG4, TQ0TIG5 位= 0
KR2/TIQ03	KRM 寄存器的 KRM2 位= 0	TQ0IOC1 寄存器的 TQ0TIG6, TQ0TIG7 位= 0
KR3/TIQ00	KRM 寄存器的 KRM3 位= 0	TQ0IOC1 寄存器的 TQ0TIG0, TQ0TIG1 位= 0 TQ0IOC1 寄存器的 TQ0EES0, TQ0EES1 位= 0 TQ0IOC1 寄存器的 TQ0ETS0, TQ0ETS1 位= 0

2. DDI, DDO, DCK 和 DMS 引脚为片上调试引脚。外部复位后禁止将这些引脚用作片上调试引脚，请参阅 第二十四章 片上调试功能。

注意事项 如果当 PFC5.PFC5n 位和 PFCE5.PFCE5n 位为默认值(0)时控制模式指定使用 PMC5 寄存器，则未定义输出。出于这一原因，首先设置 PFC5.PFC5n 位和 PFCE5.PFCE5n 位且接着设置 PMC5n 位为 1 以设置控制模式。

备注 1. 当使用复用功能时无需设置端口寄存器(Pn)。  
2. INTxn = INTFn, INTRn

表 4-14. 端口引脚用作复用-功能引脚(3/4)

引脚名称	复用-功能引脚		PMn 寄存器	PMCn 寄存器	PFCm 寄存器	PFCEm 寄存器	其他位(寄存器)
	名称	I/O					
P70	ANI0	输入	PM70 = 1 <sup>注 1</sup>	—	—	—	
P71	ANI1	输入	PM71 = 1 <sup>注 1</sup>	—	—	—	
P72	ANI2	输入	PM72 = 1 <sup>注 1</sup>	—	—	—	
P73	ANI3	输入	PM73 = 1 <sup>注 1</sup>	—	—	—	
P74	ANI4	输入	PM74 = 1 <sup>注 1</sup>	—	—	—	
P75	ANI5	输入	PM75 = 1 <sup>注 1</sup>	—	—	—	
P76	ANI6	输入	PM76 = 1 <sup>注 1</sup>	—	—	—	
P77	ANI7	输入	PM77 = 1 <sup>注 1</sup>	—	—	—	
P78	ANI8	输入	PM78 = 1 <sup>注 1</sup>	—	—	—	
P79	ANI9	输入	PM79 = 1 <sup>注 1</sup>	—	—	—	
P710	ANI10	输入	PM710 = 1 <sup>注 1</sup>	—	—	—	
P711	ANI11	输入	PM711 = 1 <sup>注 1</sup>	—	—	—	
P90	KR6	输入	无需设置	PMC90 = 1	PFC90 = 1	PFCE90 = 0	
	TXDA1	输出	无需设置	PMC90 = 1	PFC90 = 0	PFCE90 = 1	
P91	KR7 <sup>注 2</sup>	输入	无需设置	PMC91 = 1	PFC91 = 1	PFCE91 = 0	
					PFC91 = 0	PFCE91 = 1	
P96	TIP21	输入	无需设置	PMC96 = 1	PFC96 = 0	PFCE96 = 1	
	TOP21	输出	无需设置	PMC96 = 1	PFC96 = 1	PFCE96 = 1	
P97	SIB1	输入	无需设置	PMC97 = 1	PFC97 = 1	PFCE97 = 0	
	TIP20	输入	无需设置	PMC97 = 1	PFC97 = 0	PFCE97 = 1	
	TOP20	输出	无需设置	PMC97 = 1	PFC97 = 1	PFCE97 = 1	
P98	SOB1	输出	无需设置	PMC98 = 1	PFC98 = 1	—	
P99	SCKB1	I/O	无需设置	PMC99 = 1	PFC99 = 1	—	
P913	INTP4	输入	无需设置	PMC913 = 1	PFC913 = 1	PFCE913 = 0	INTx913(INTx9H)
	PCL	输出	无需设置	PMC913 = 1	PFC913 = 0	PFCE913 = 1	
P914	INTP5	输入	无需设置	PMC914 = 1	PFC914 = 1	—	INTx914(INTx9H)
P915	INTP6	输入	无需设置	PMC915 = 1	PFC915 = 1	—	INTx915(INTx9H)

注 1. 设置 PM7n 为 1 以使用 P7n(ANIn)的复用功能。

2. KR7 引脚和 RXDA1 引脚为复用-功能引脚。

当将该引脚用作 RXDA1 引脚时，禁用 KR7 引脚按键中断检测。(KRM.KRM7 位清零。)

同样，当将该引脚用作 KR7 引脚时，建议设置 PFC91 位为 1 且 PFCE91 位清零。

注意事项 如果当 PFC9.PFC9n 位和 PFCE9.PFCE9n 位为默认值(0)时控制模式指定使用 PMC9 寄存器，则未定义输出。出于这一原因，首先设置 PFC9.PFC9n 位和 PFCE9.PFCE9n 位且接着设置 PMC9n 位为 1 以设置控制模式。

备注 1. 当使用复用功能时无需设置端口寄存器(Pn)。

2. INTxn = INTFn, INTRn

表 4-14. 端口引脚用作复用-功能引脚(4/4)

引脚名称	复用-功能引脚		PMn 寄存器	PMCn 寄存器	PFCm 寄存器	PFCEm 寄存器	其他位(寄存器)
	名称	I/O					
PCM1	CLKOUT	输出	无需设置	PMCCM1 = 1	—	—	
PDL5	FLMD1	输入	无需设置	无需设置	—	—	注

**注** FLMD1 引脚无需使用端口控制寄存器操作，因为其用于 flash 编程模式。如需详细信息，请参阅 第二十二章 闪存。

**备注** 当使用复用功能时无需设置端口寄存器(Pn)。

#### 4.4 端口框图

图 4-2. A-1 型框图

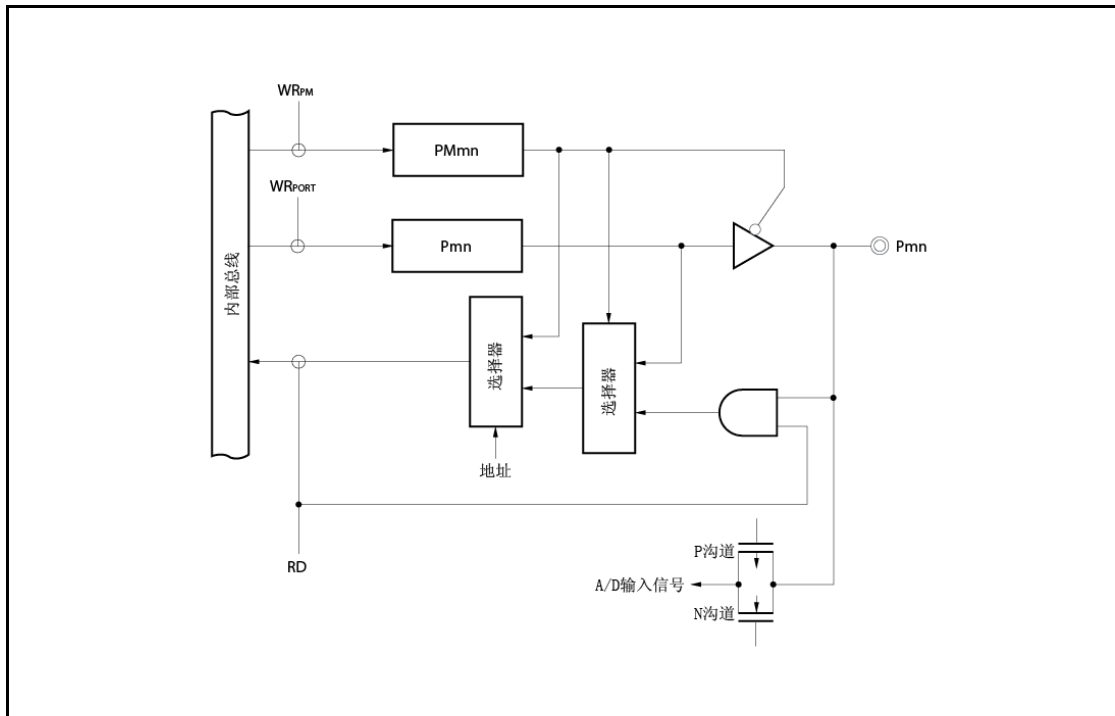


图 4-3. B-1 型框图

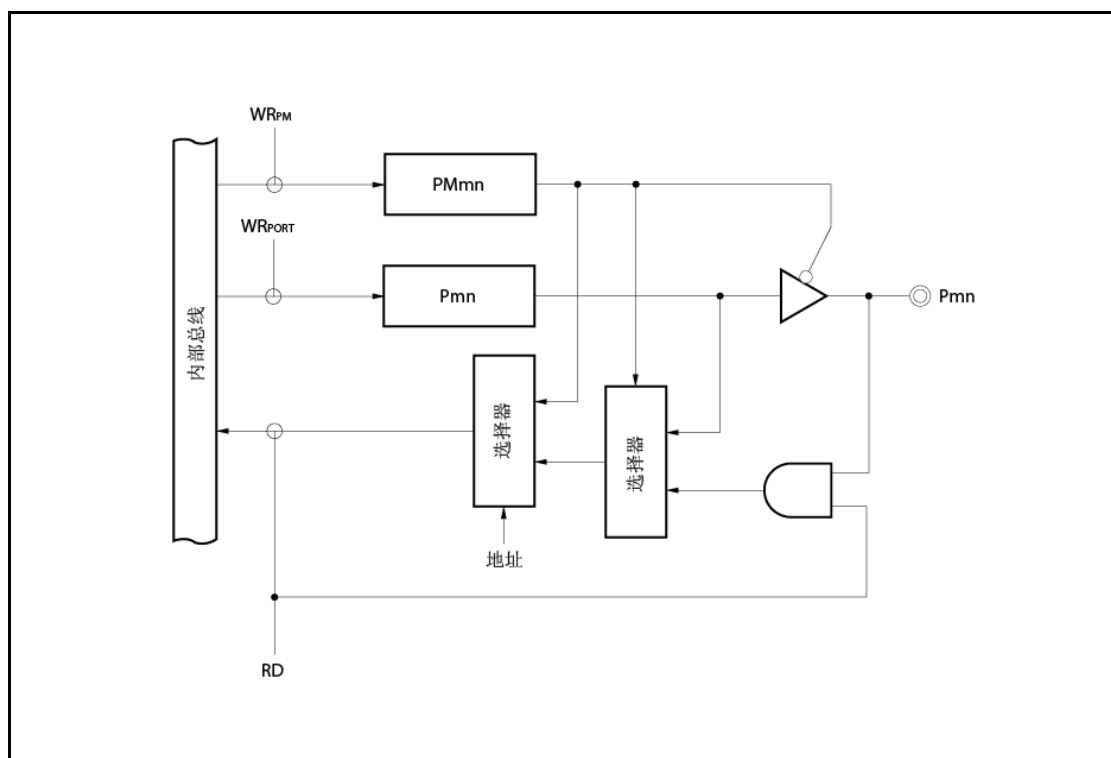


图 4-4. C-1 型框图

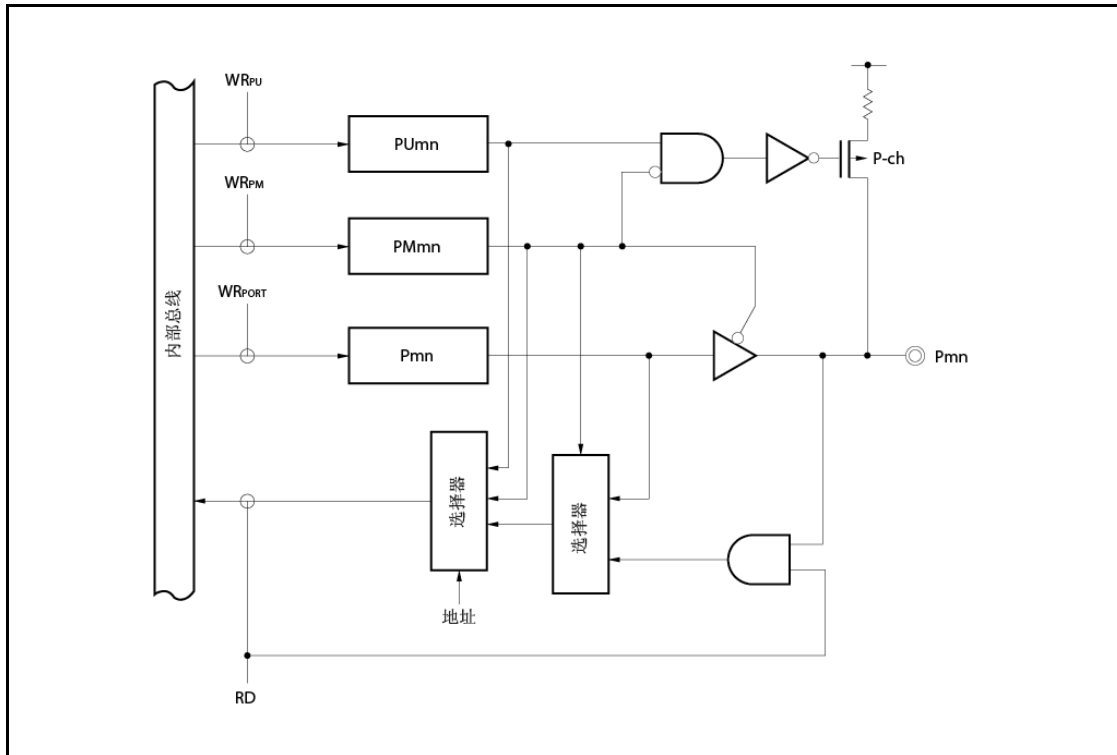






图 4-6. E-1 型框图

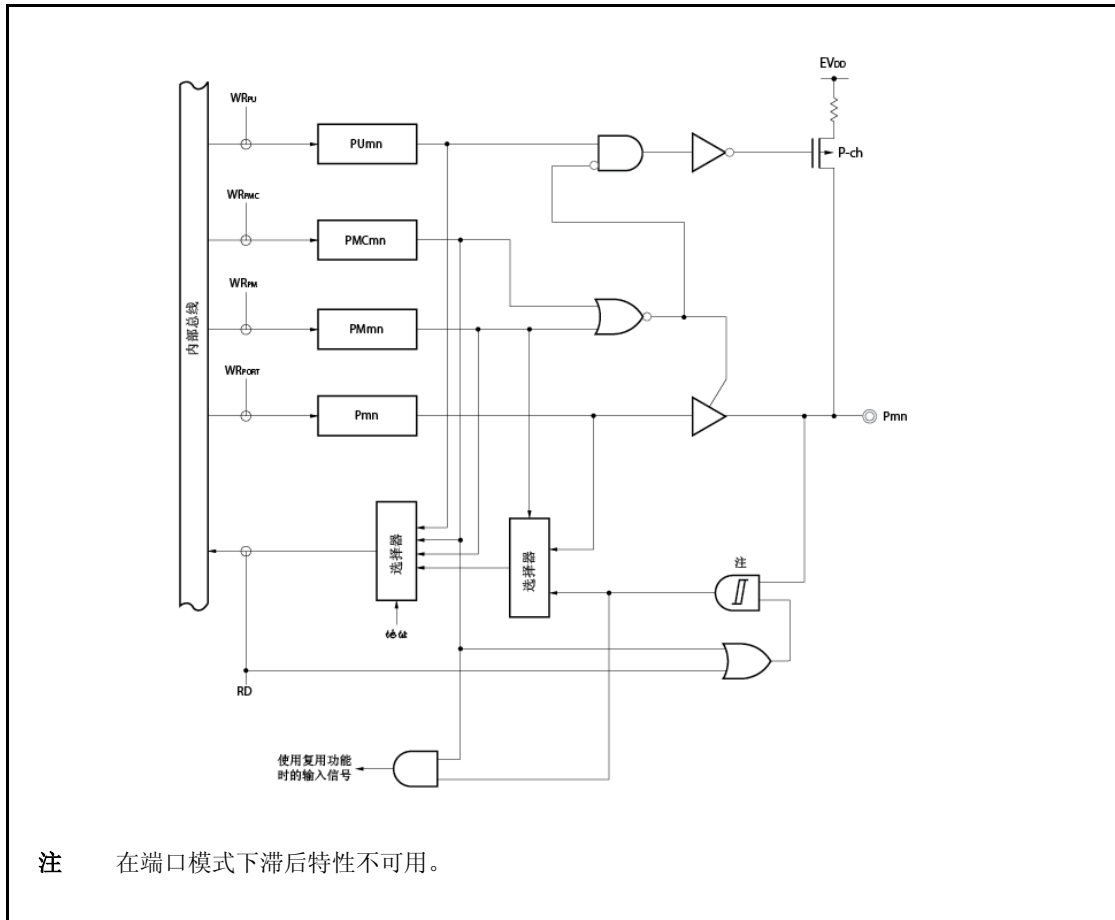


图 4-7. E-2 型框图

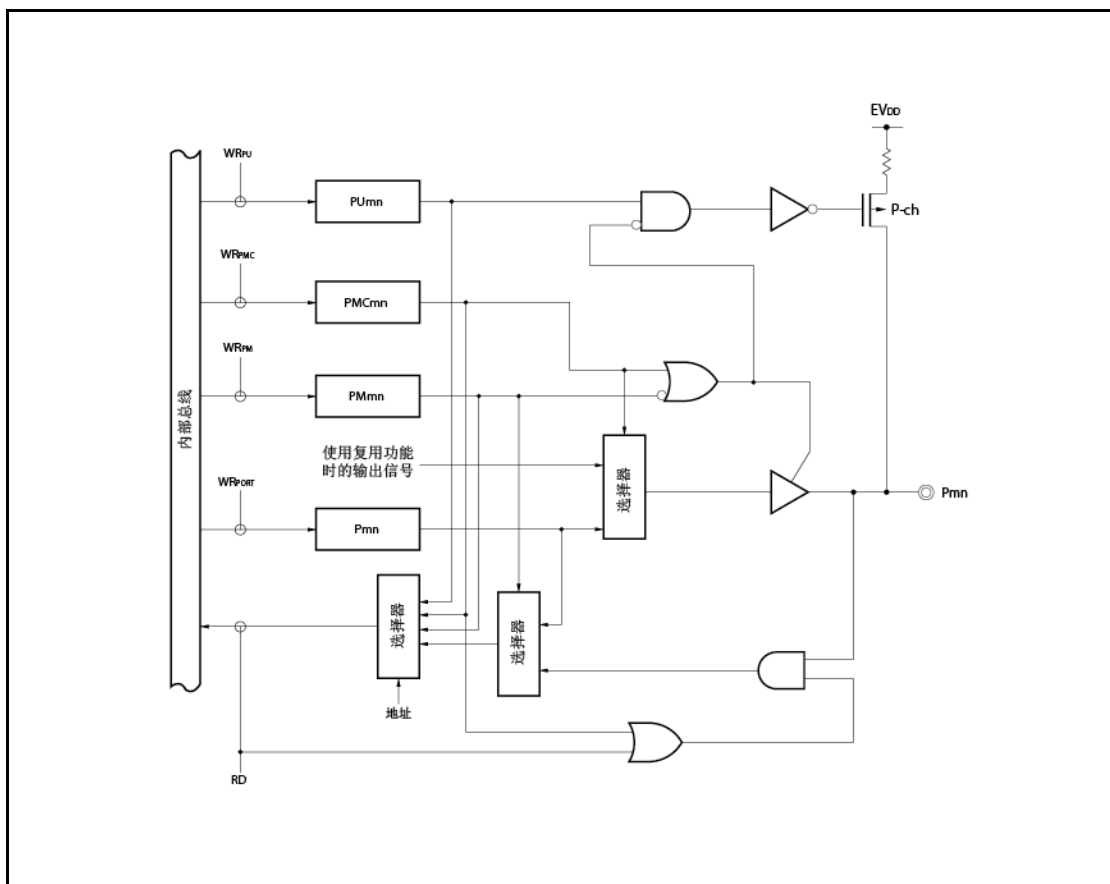


图 4-8. E-3 型框图

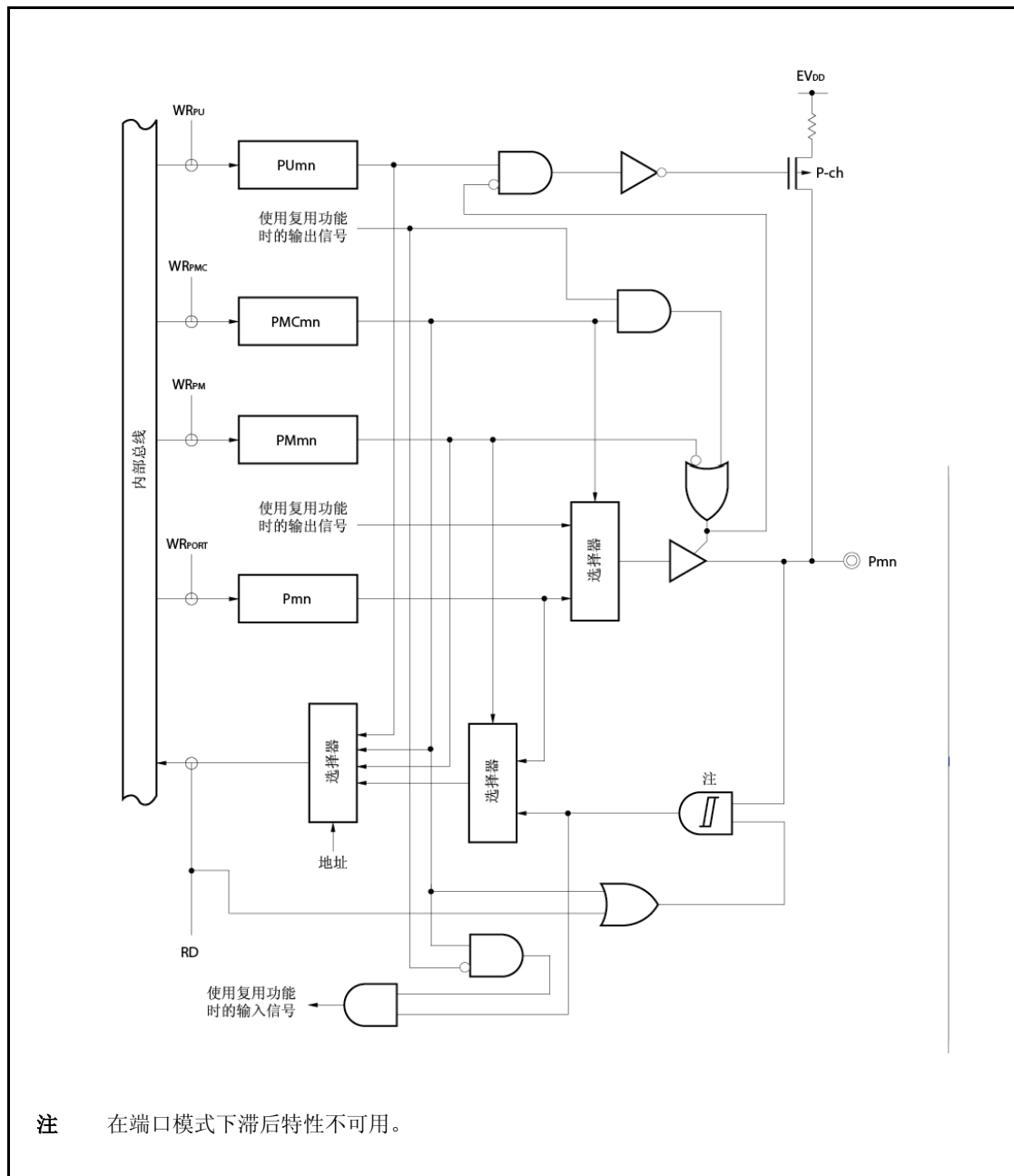


图 4-9. G-1 型框图

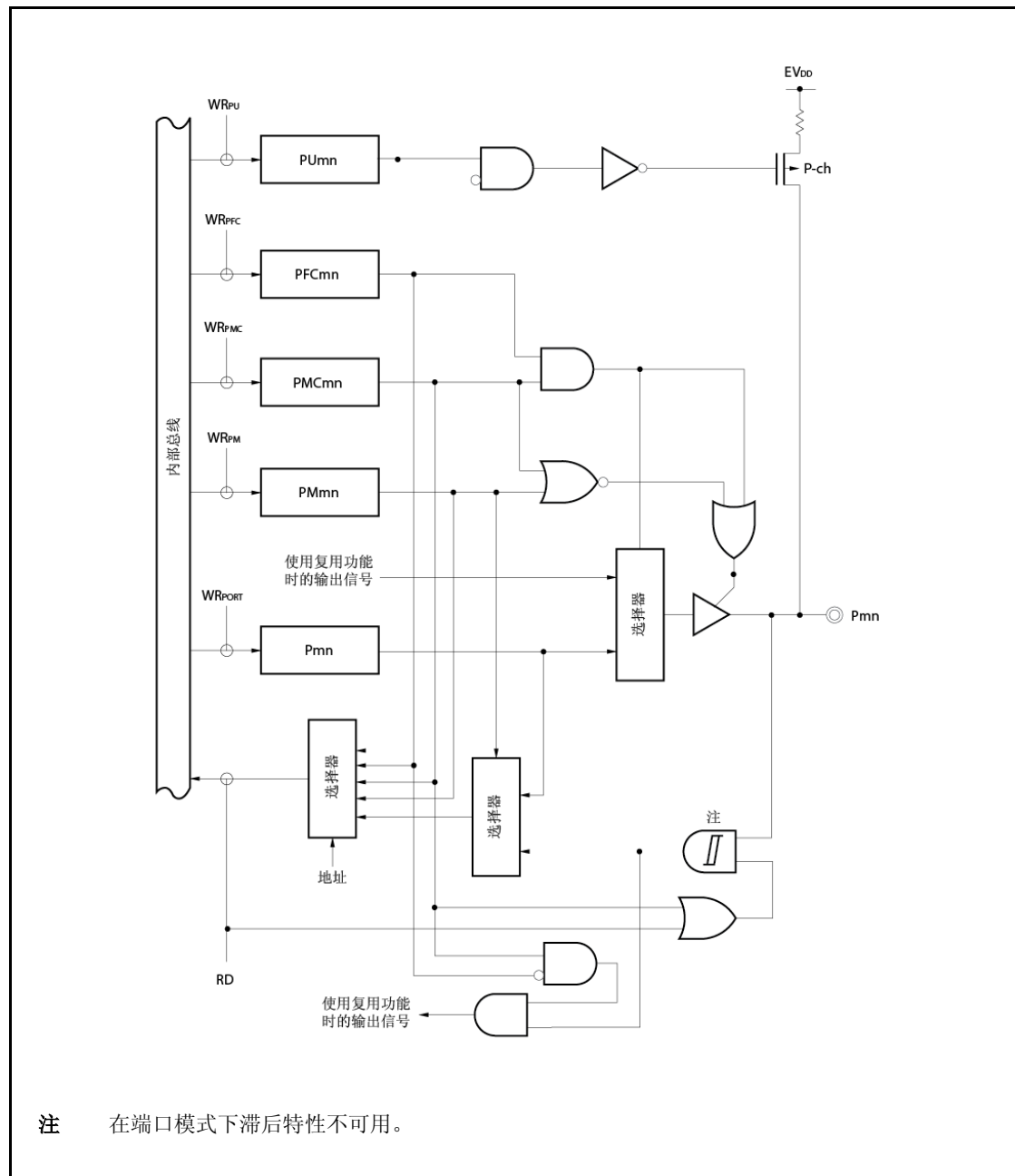


图 4-10. G-2 型框图

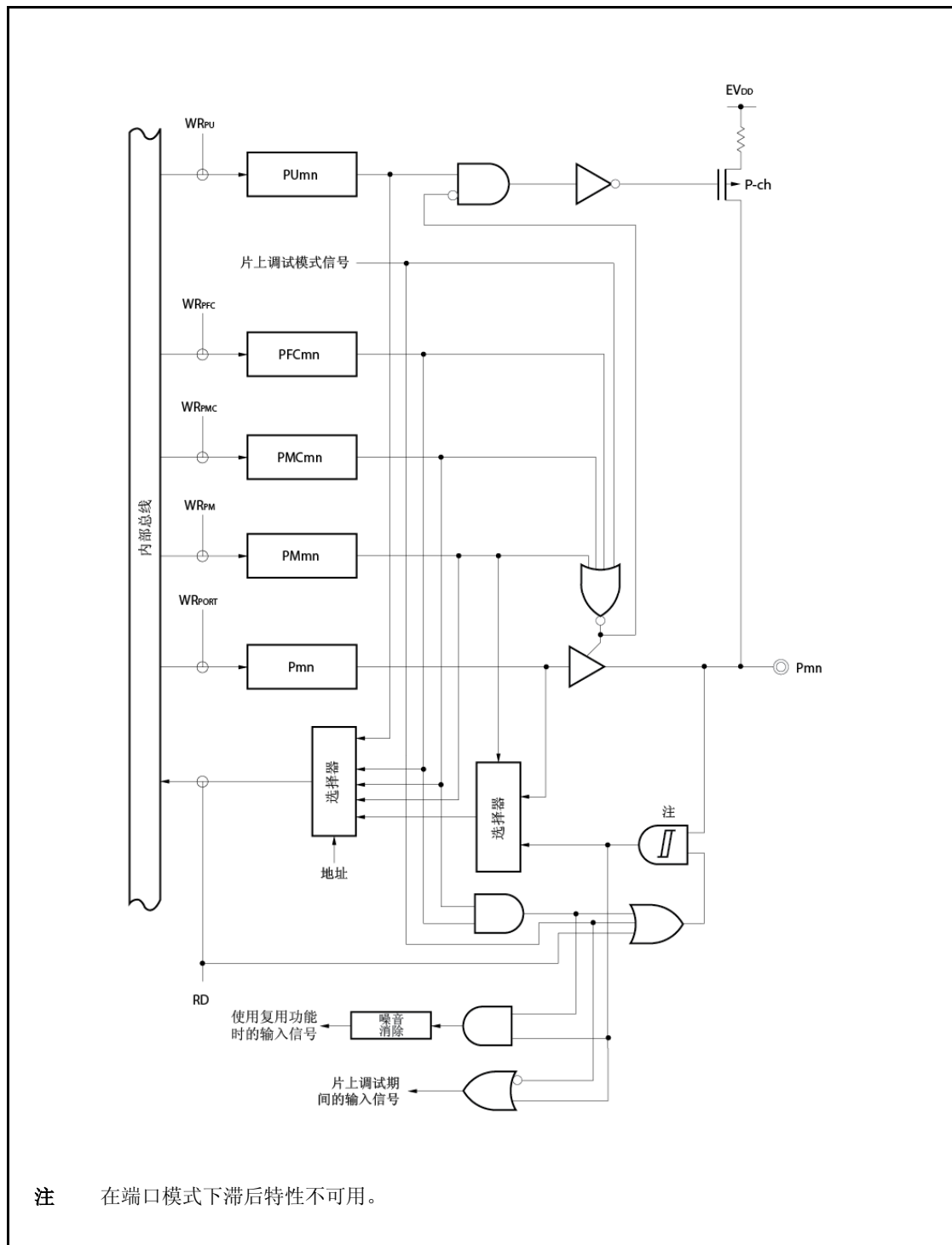


图 4-11. G-3 型框图

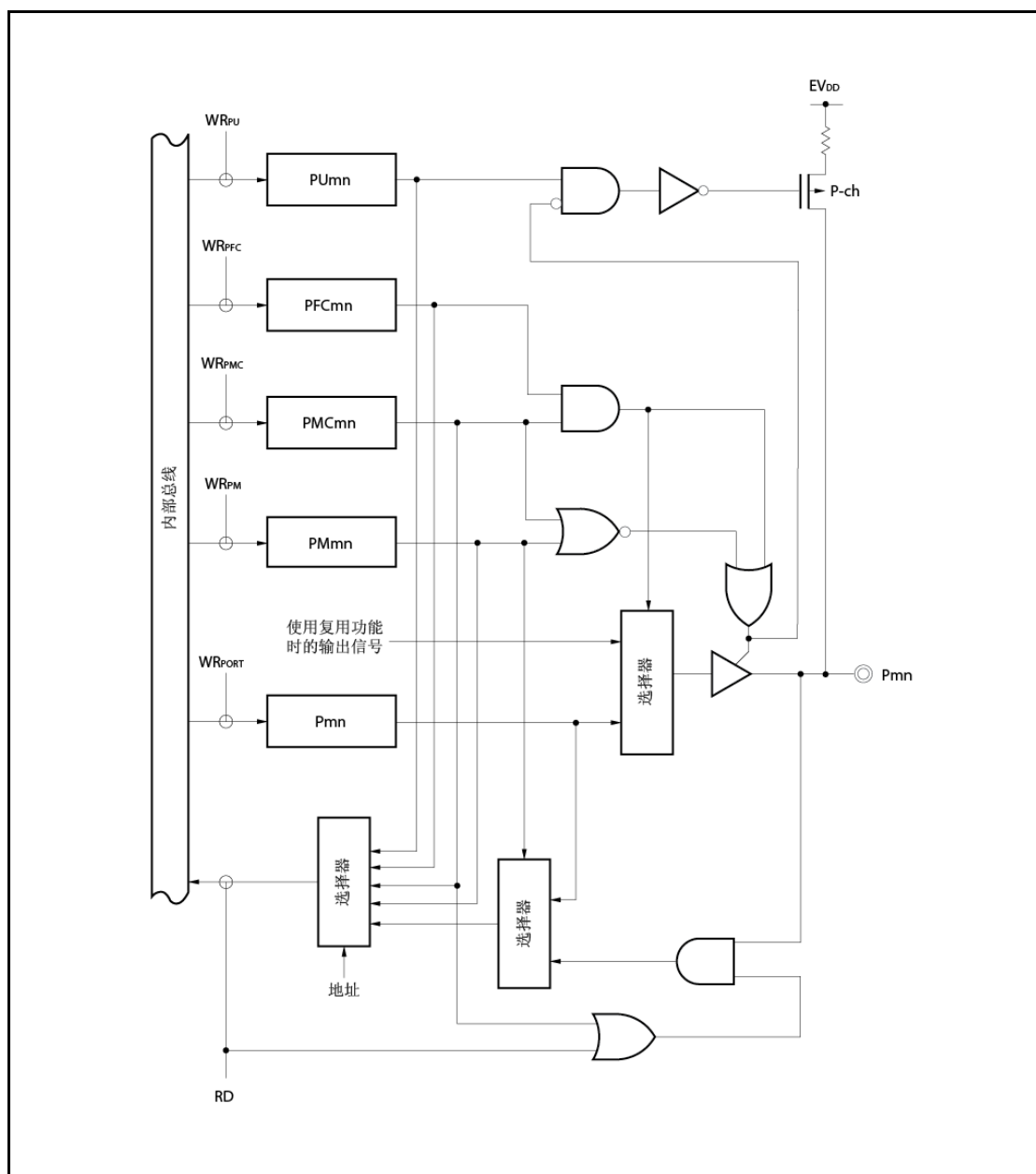


图 4-12. G-5 型框图

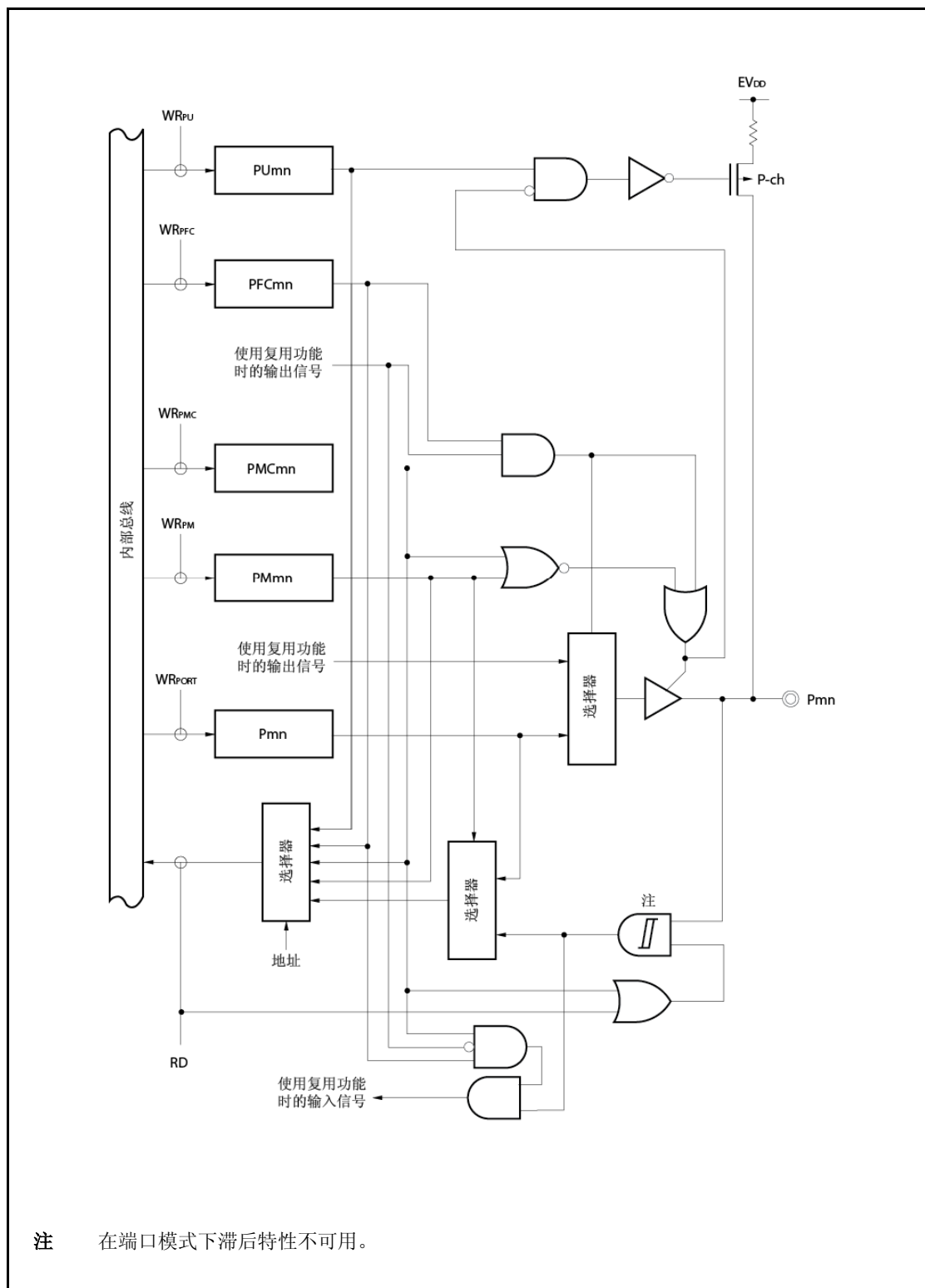




图 4-13. L-1 型框图

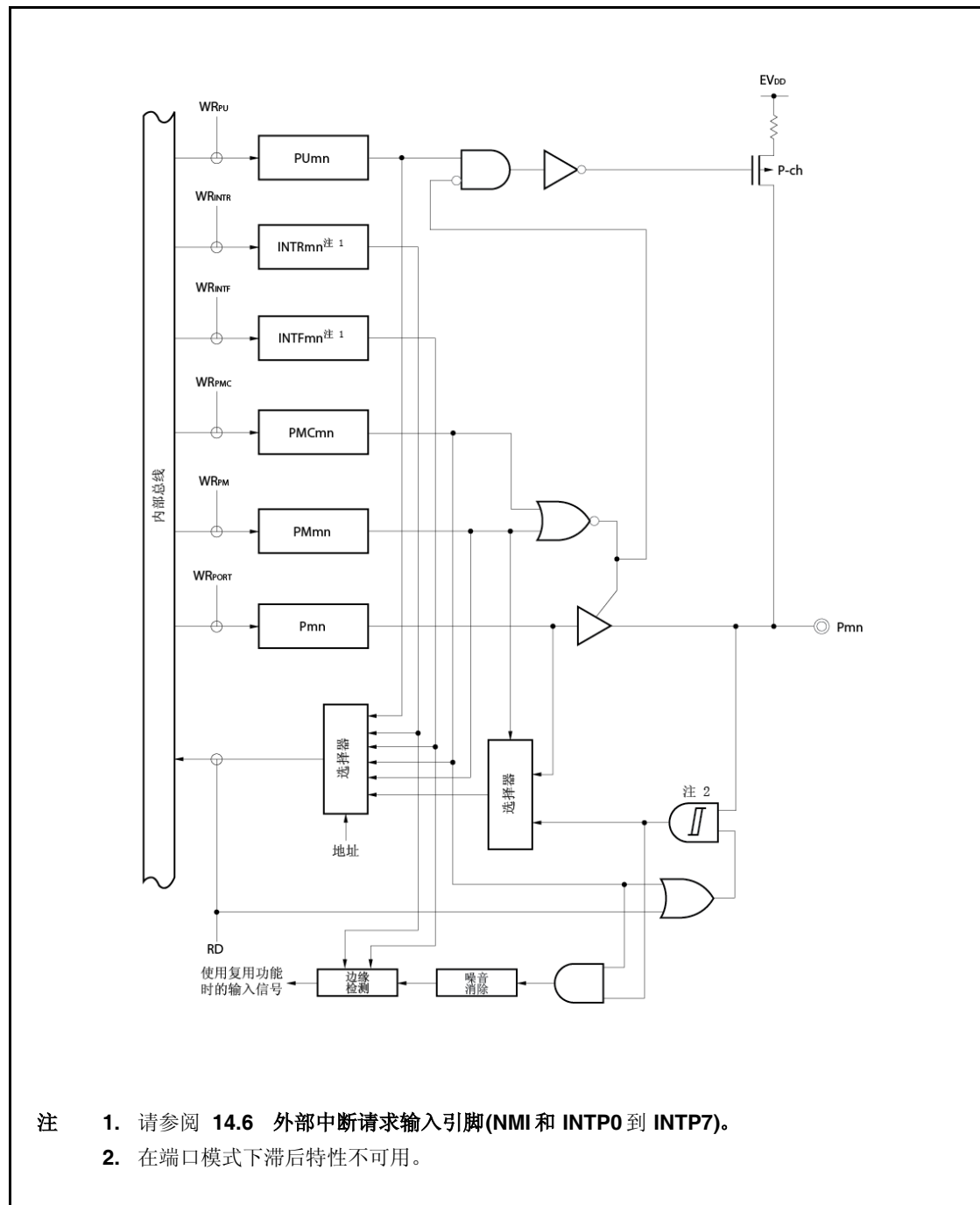


图 4-14. L-2 型框图

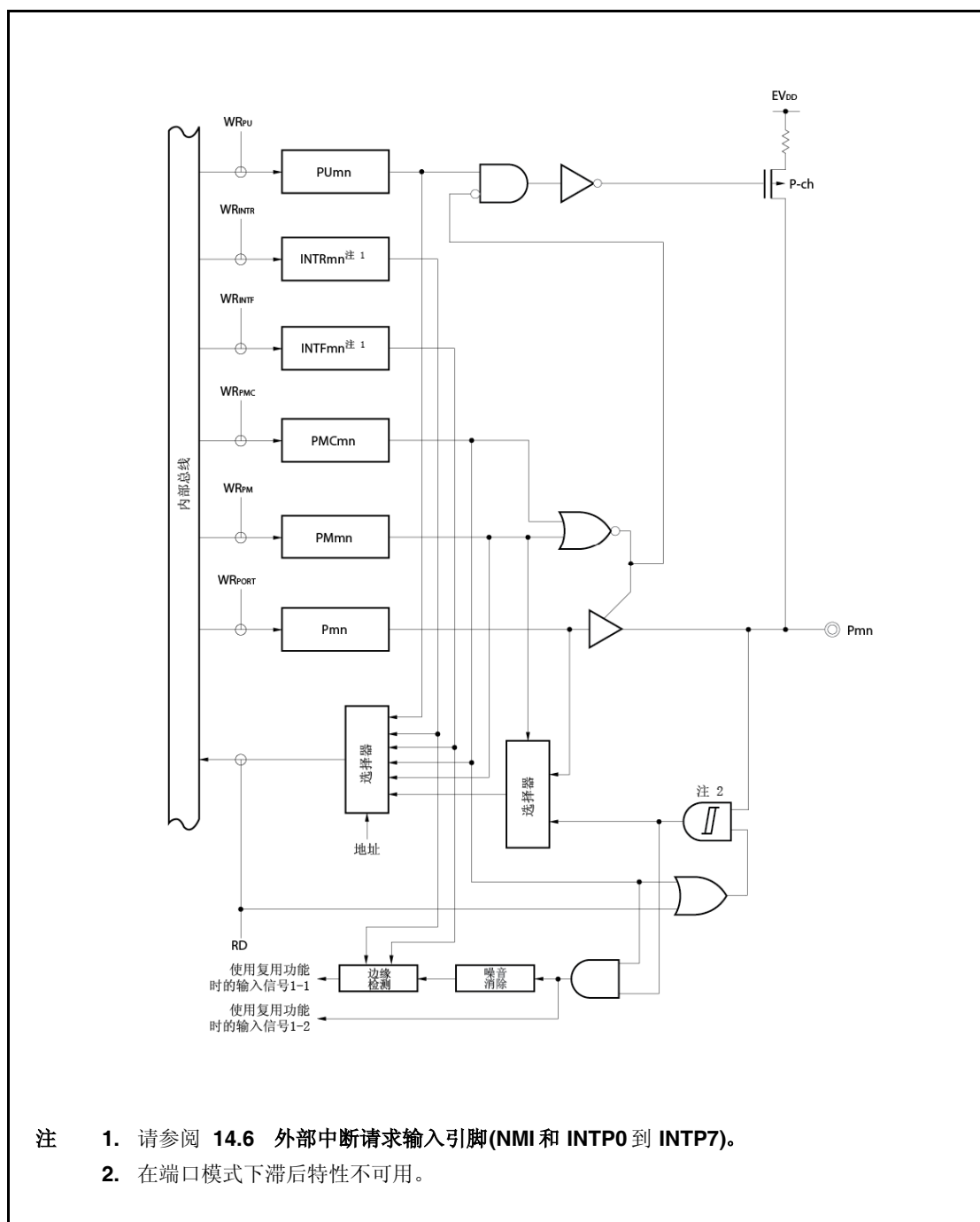


图 4-15. N-1 型框图

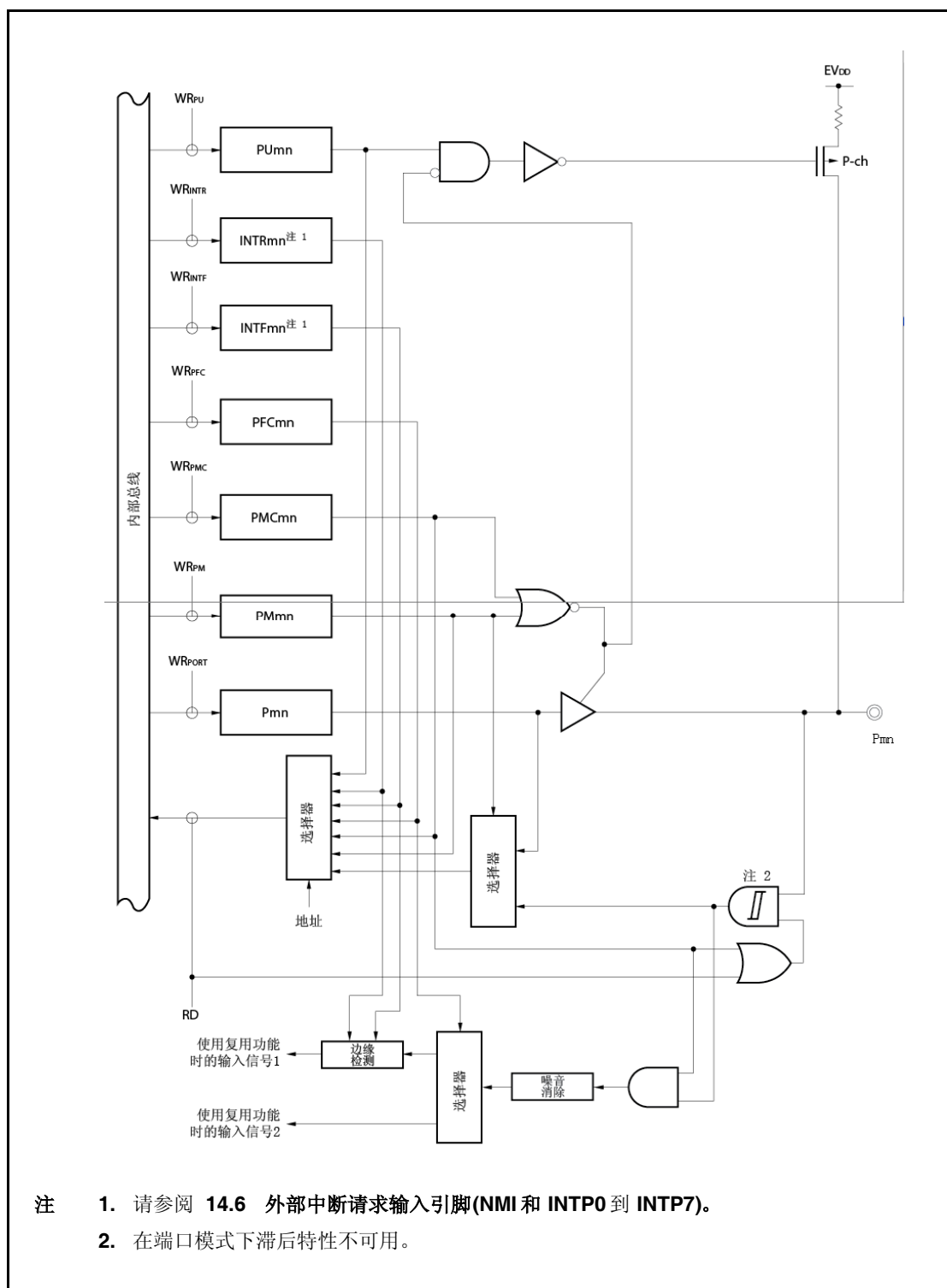


图 4-16. N-2 型框图

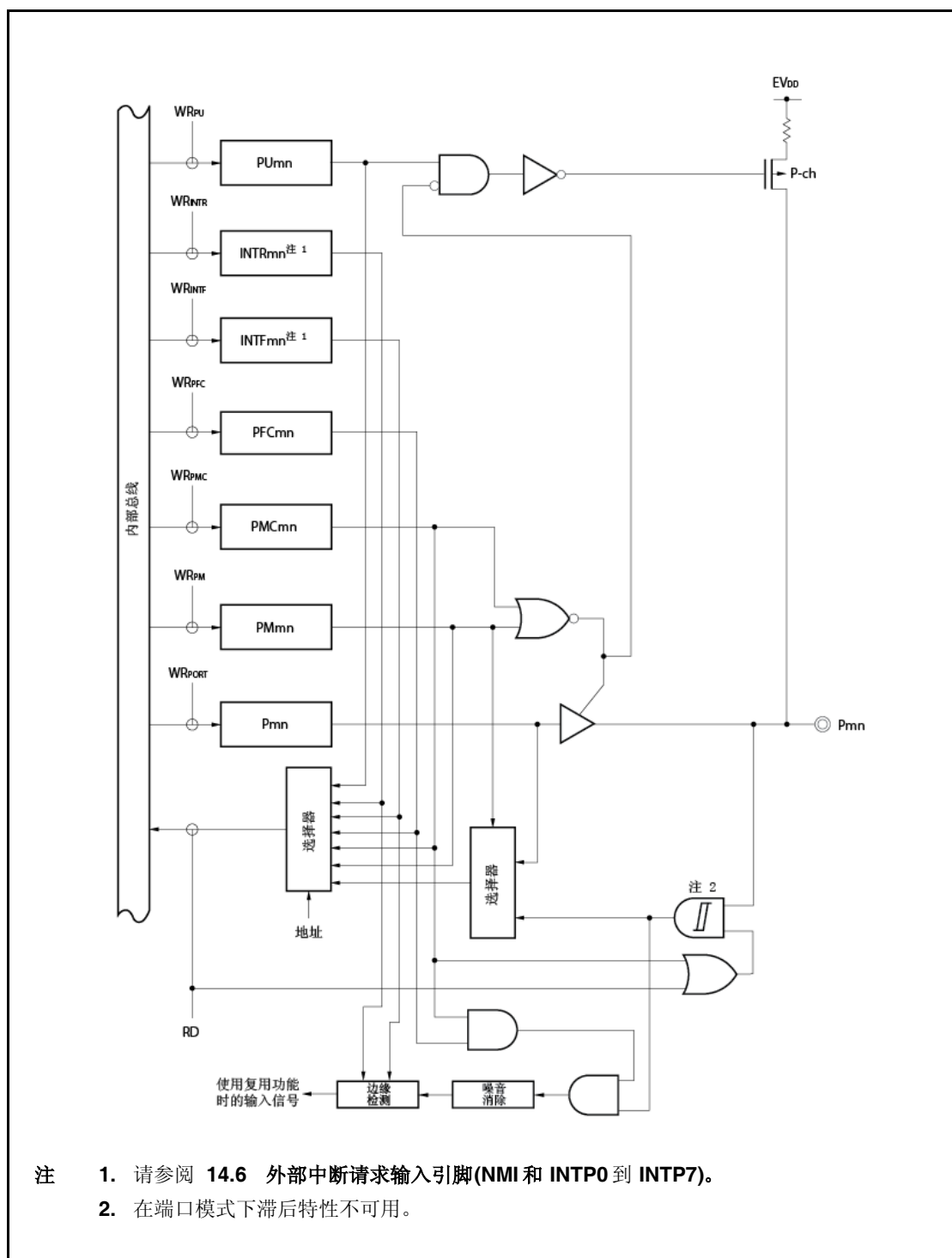


图 4-17. U-4 型框图

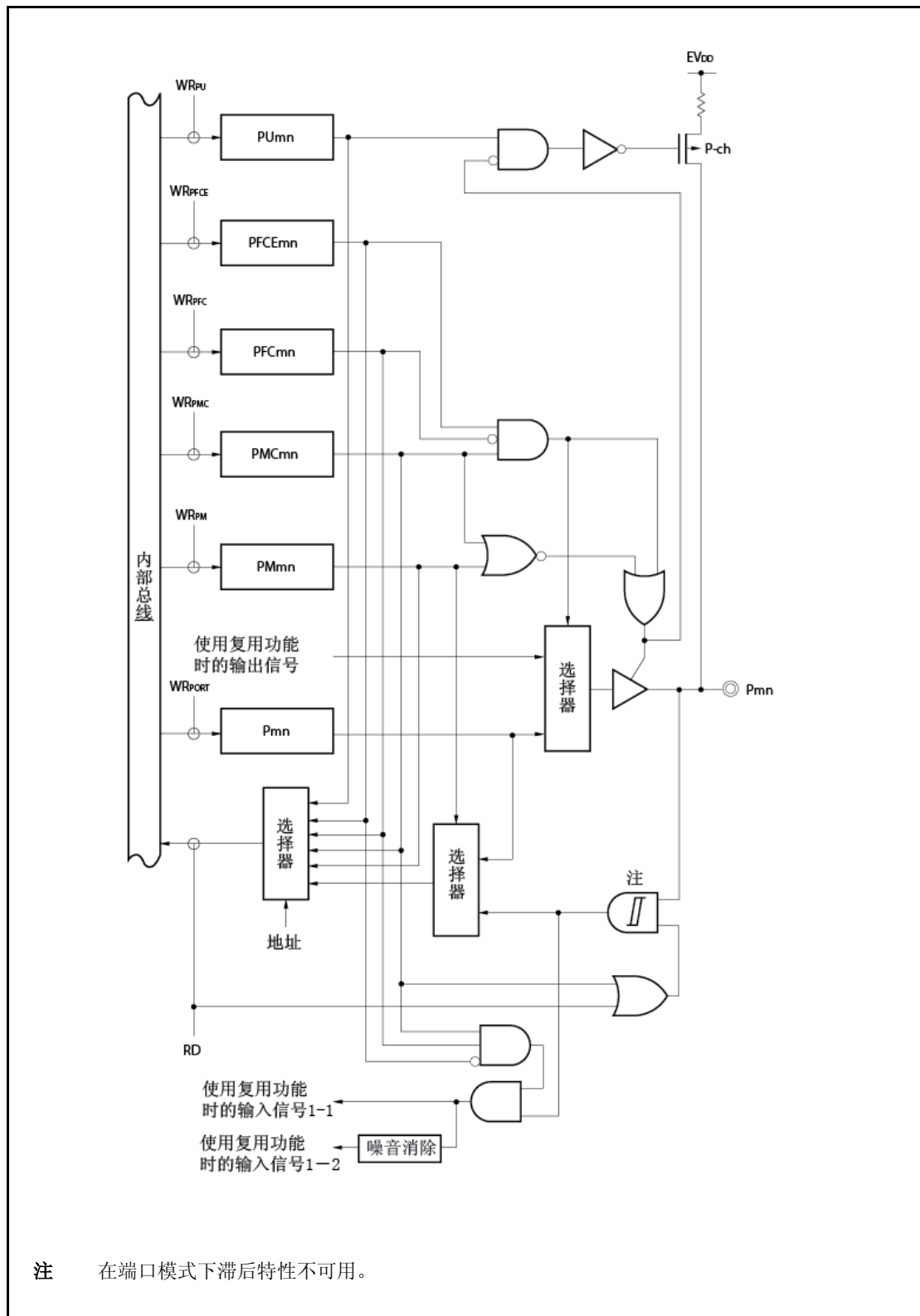


图 4-18. U-5 型框图

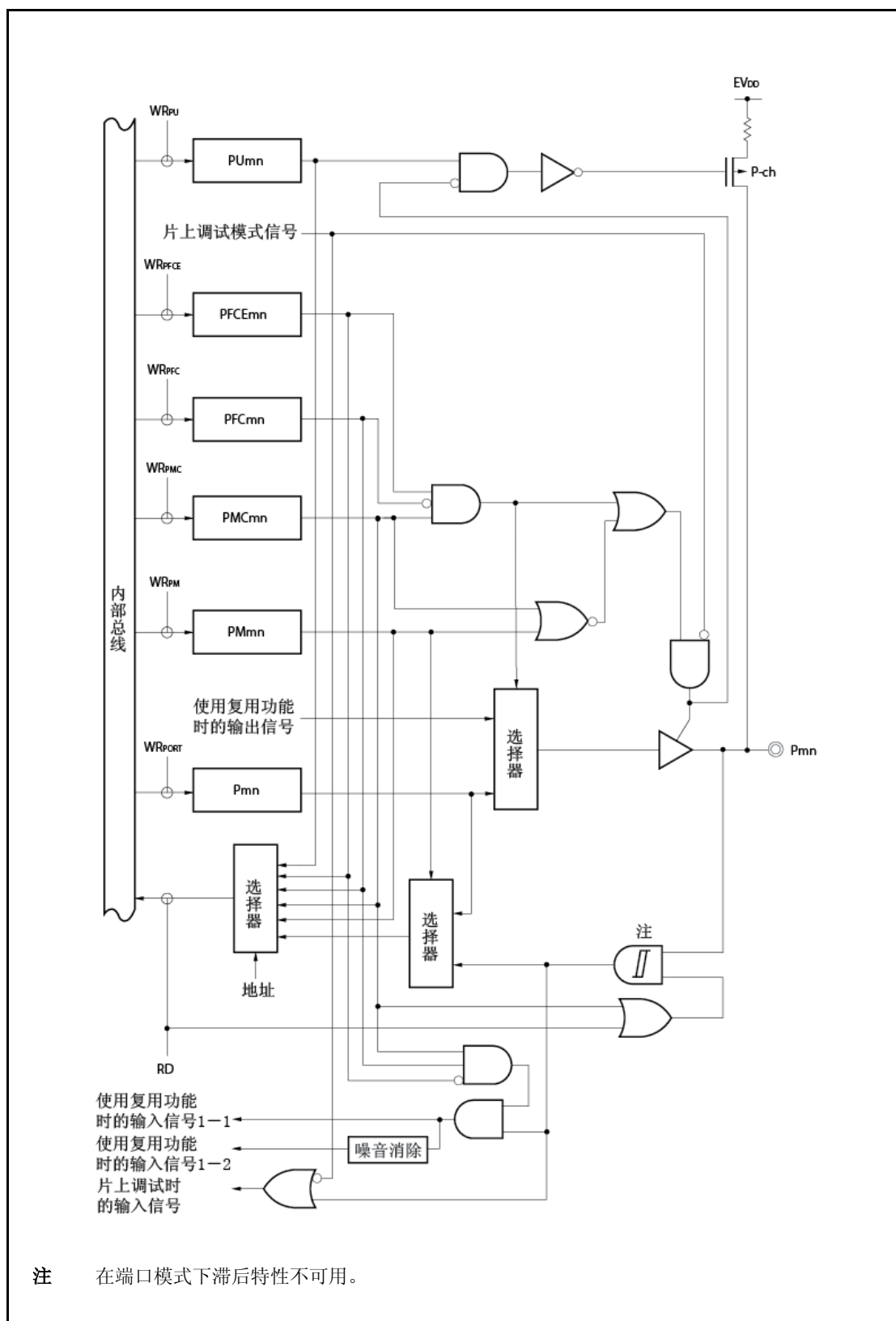


图 4-19. U-6 型框图

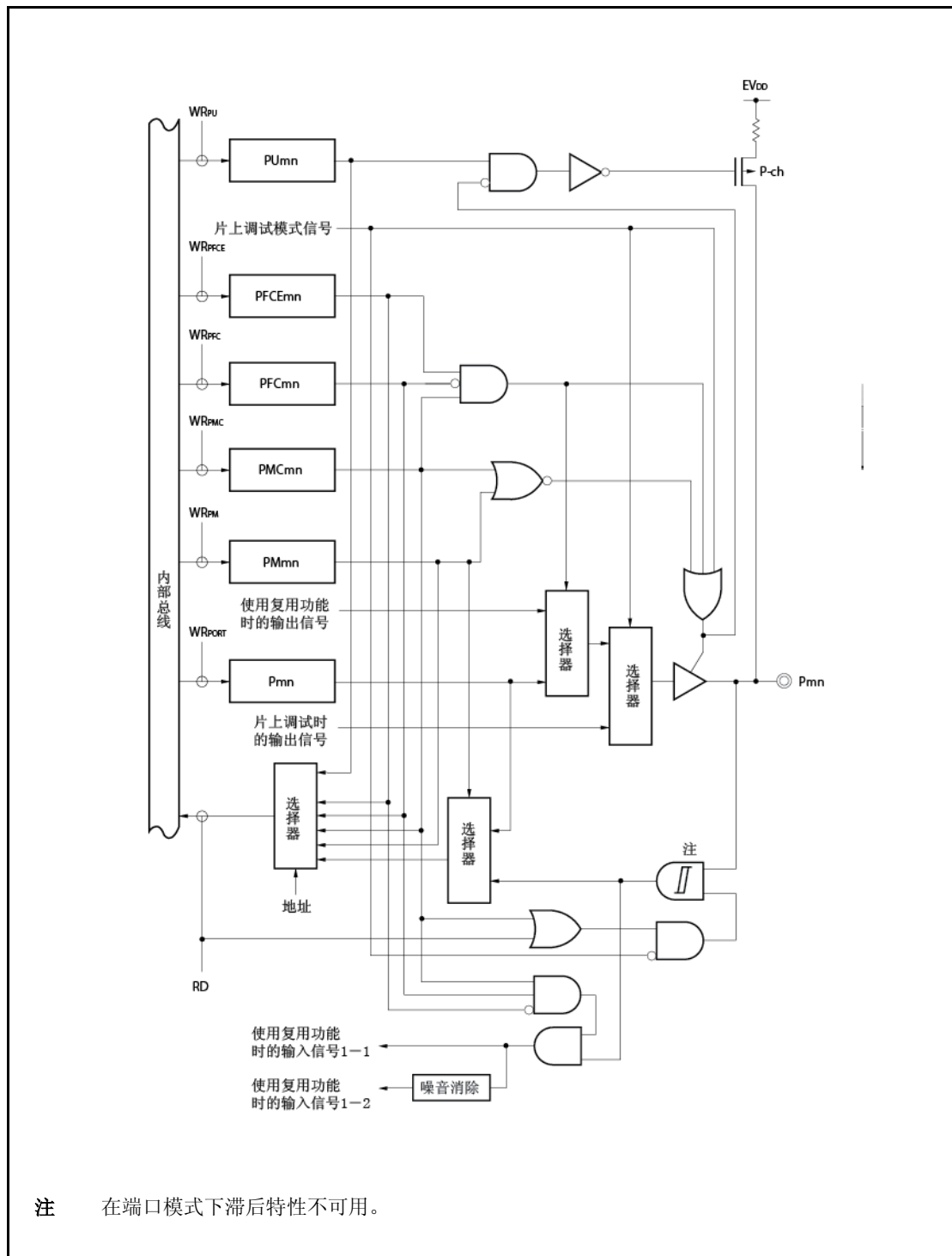
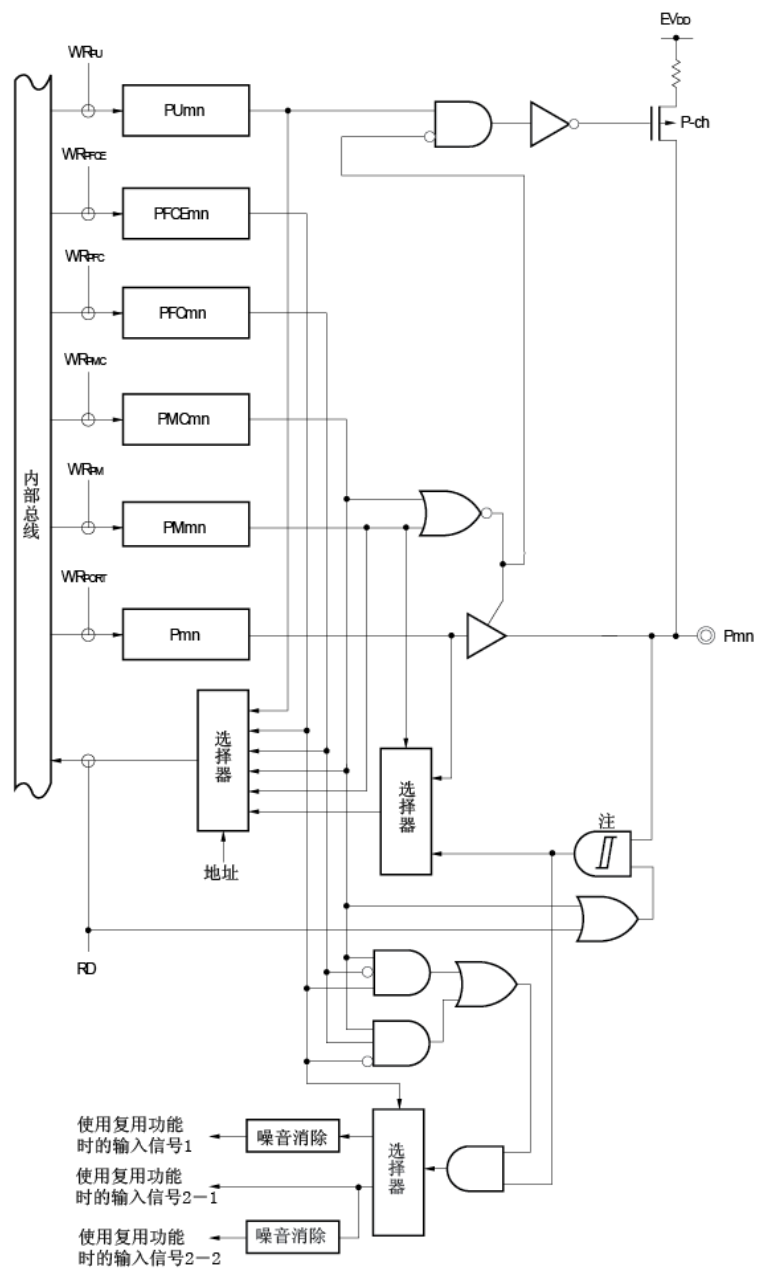


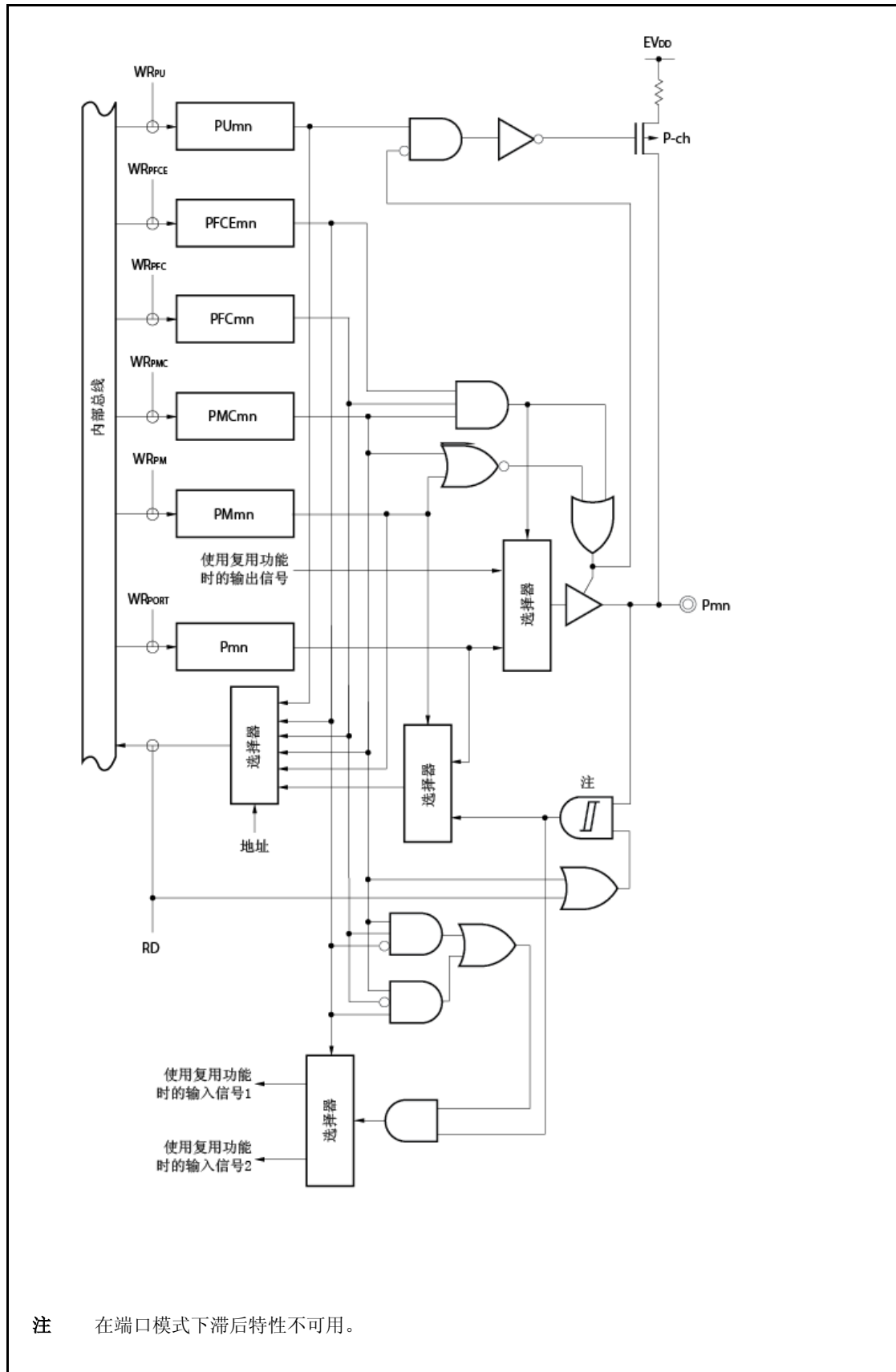
图 4-20. U-7 型框图



注 在端口模式下滞后特性不可用。

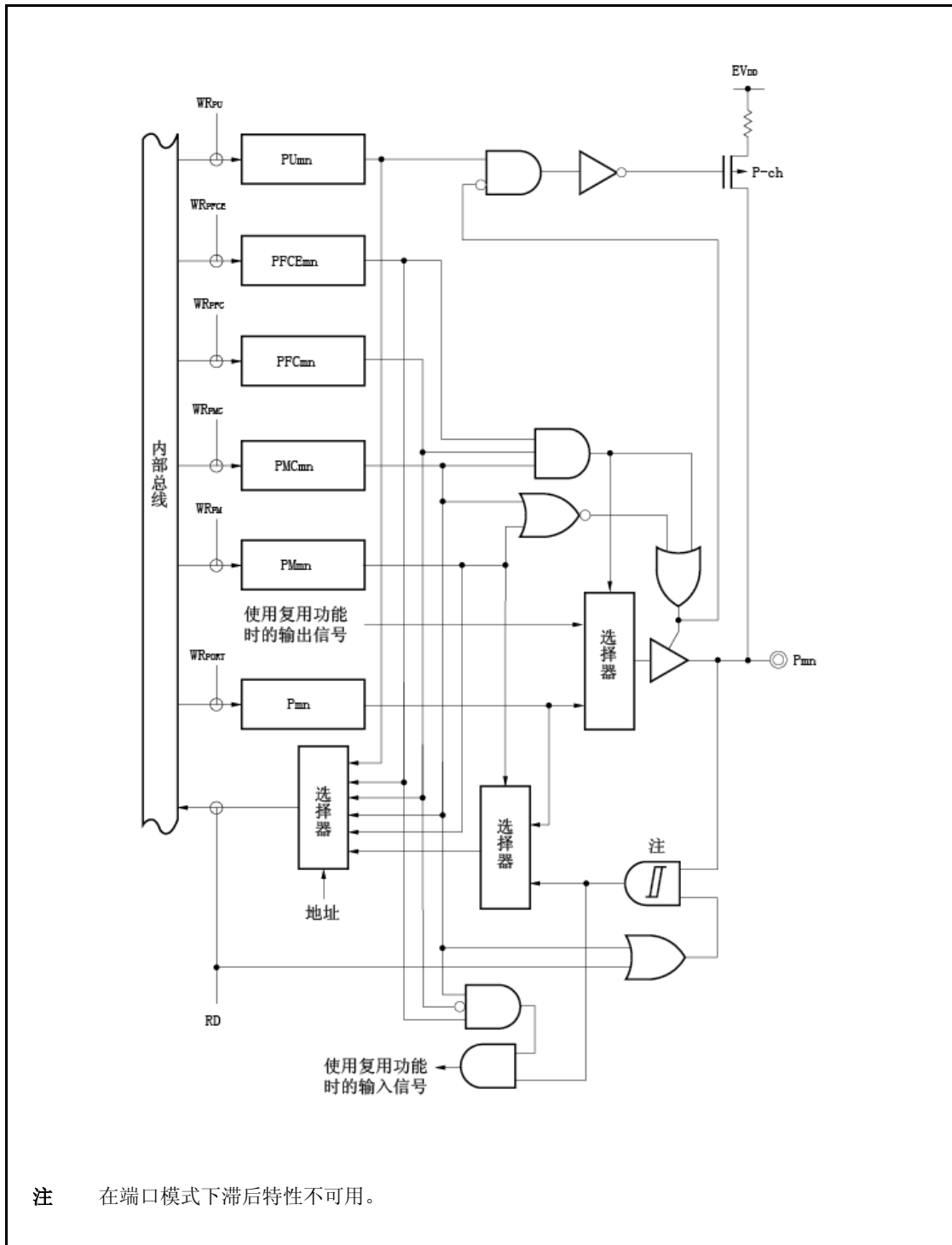


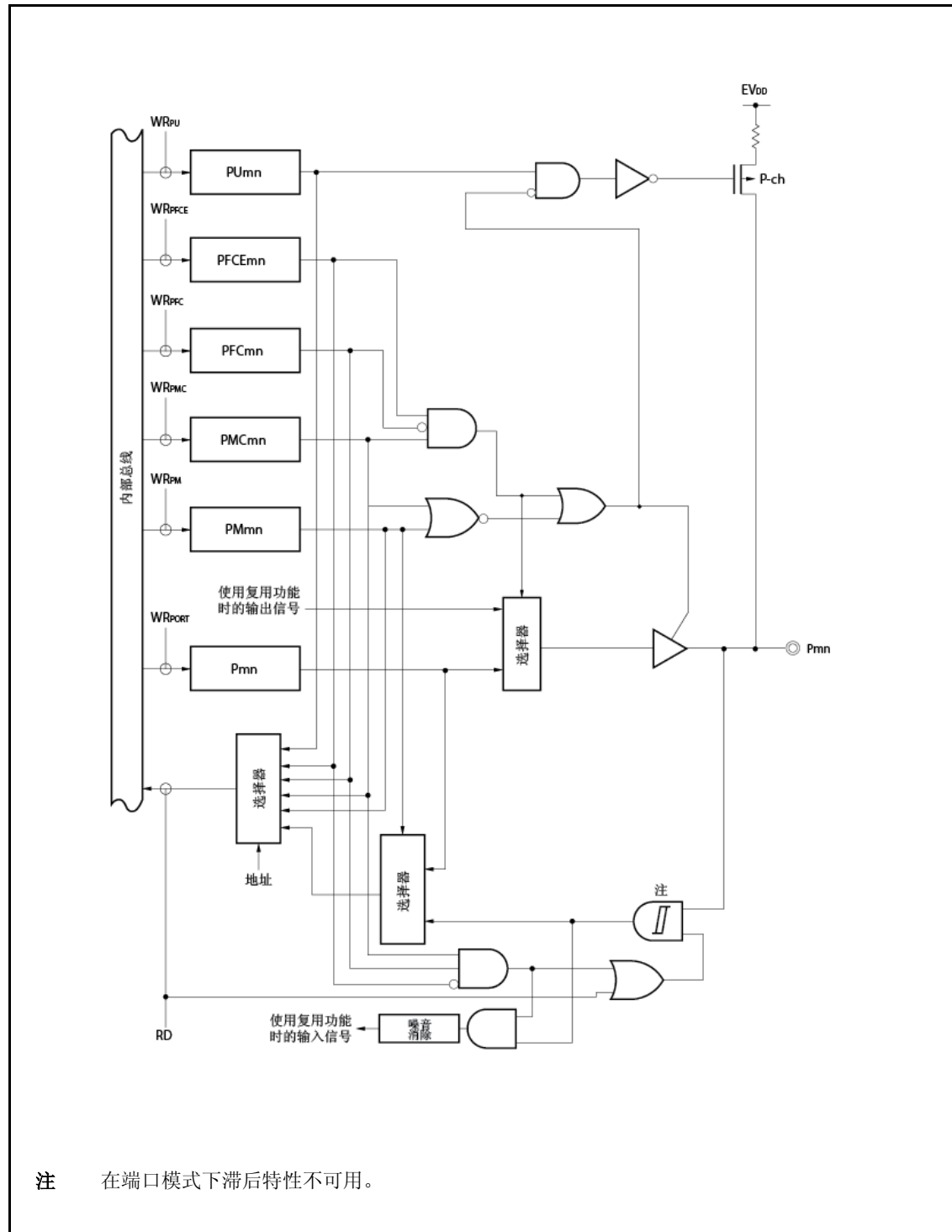
图 4-21. U-8 型框图



注 在端口模式下滞后特性不可用。

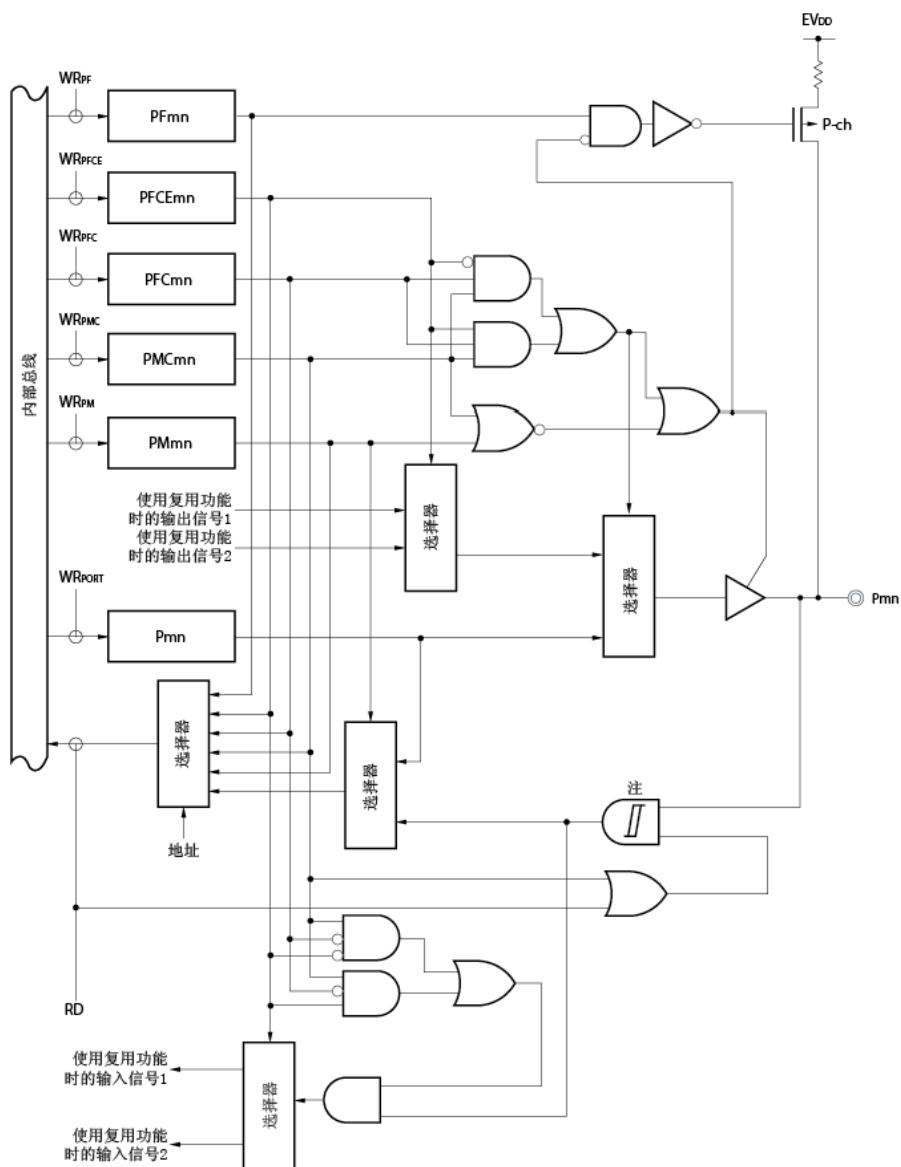
图 4-22. U-9 型框图





**注** 在端口模式下滞后特性不可用。

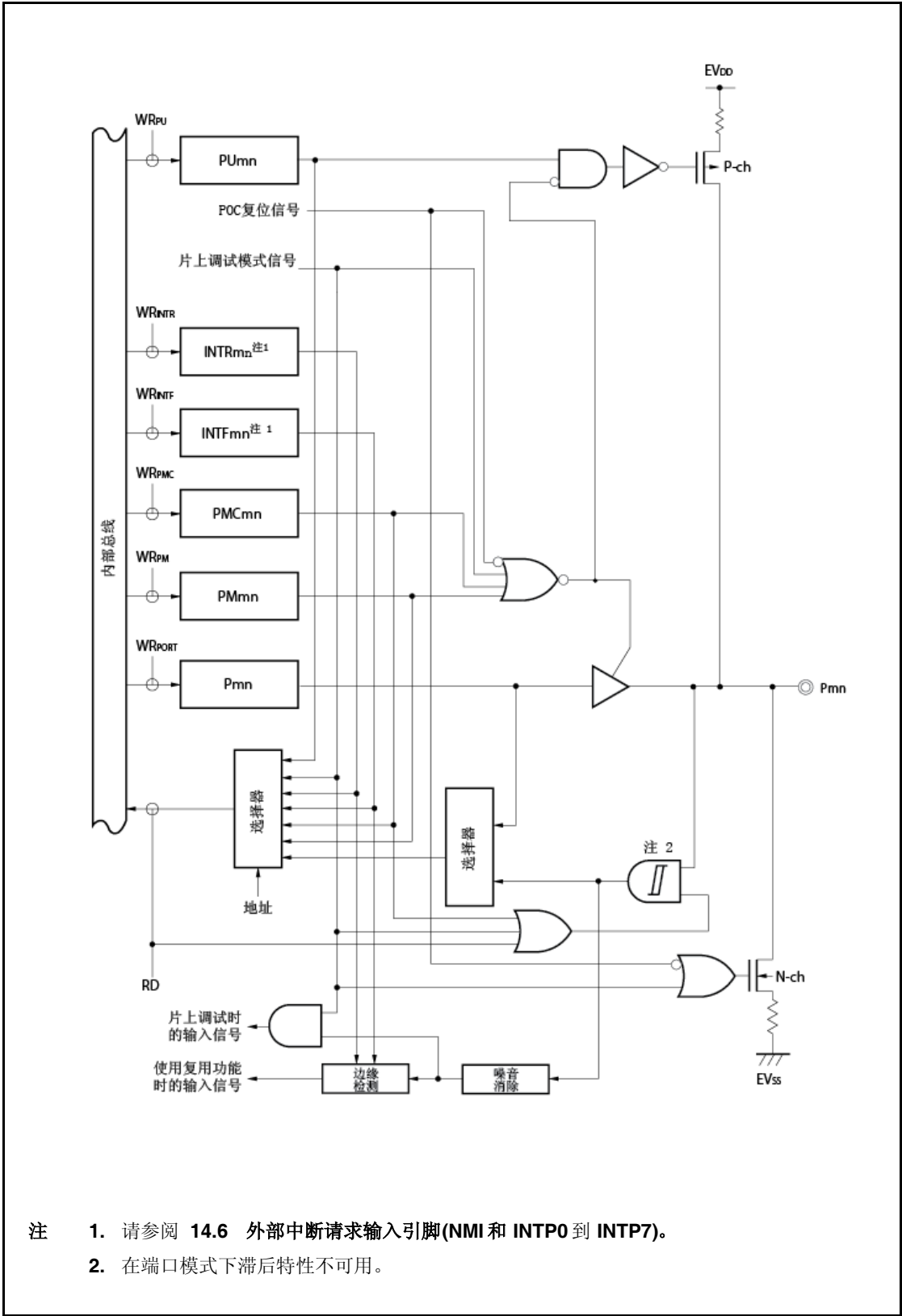
图 4-24. U-13 型框图



注 在端口模式下滞后特性不可用。



图 4-26. AA-1 型框图



## 4.5 注意事项

### 4.5.1 设置端口引脚的注意事项

- (1) 在 V850ES/HF2 中，常用端口功能和几个外围功能 I/O 引脚共用引脚。要在常用端口(端口模式)与外围功能 I/O 引脚(复用-功能模式)之间切换，请通过 PMCn 寄存器设置。对于此寄存器设置系列，请注意以下注意事项。

- (a) 从端口模式切换到复用-功能模式的注意事项  
按以下顺序从端口模式切换到复用-功能模式。

- |                          |            |
|--------------------------|------------|
| <1> 设置 PFCn 和 PFCEn 寄存器: | 复用-功能选择    |
| <2> 设置 PMCn 寄存器的对应位为 1:  | 切换到复用-功能模式 |

如果首先设置 PMCn 寄存器，则应注意以下注意事项，在此时或根据 PFCn 和 PFCEn 寄存器设置而进行引脚状态的改变，可能出现不当操作。

**注意事项** 无论端口模式/复用-功能模式，Pn 寄存器如下所示进行读取和写入。

- Pn 寄存器读取: 读取端口输出锁存器值(当 PMn.PMnm 位=0 时)或读取引脚状态(PMn.PMnm 位= 1)。
- Pn 寄存器写入: 写入端口输出锁存器

<R>

- (b) 复用-功能模式(输入)的注意事项

当 PMCn.PMCnm 位为 0 时由于 PMCn 寄存器设置值的 AND 输出和引脚电平，复用-功能块的输入信号为低电平。因此，根据端口设置和复用-功能操作启用时序，可能出现不当操作。因此，请按以下顺序在端口模式与复用-功能模式之间切换。

- 从端口模式切换到复用-功能模式(输入)  
使用 PMCn 寄存器设置引脚为复用-功能模式且接着启用复用-功能操作。
- 从复用-功能模式(输入)切换到端口模式  
停止复用-功能操作且接着将引脚切换到端口模式。

### 5.1 概述

可用下列时钟发生功能。

- 主时钟振荡器
  - 在时钟直通模式下  
 $f_x=4$  到 5MHz ( $f_{xx}=4$  到 5MHz)
  - 在 PLL 模式下  
 $f_x=4$  到 5MHz ( $f_{xx}=16$  到 20MHz)
- 副时钟振荡器 (由选项字节功能选择晶体振荡或 RC 振荡)
  - $f_{XT}=32.768\text{kHz}$  (晶体谐振器)
  - $f_{XT}=20\text{kHz}$  (RC 振荡器)
- 由锁相环 (PLL) 提供的乘法功能 ( $\times 4$ )
  - 可选时钟直通模式/PLL 模式
- 内部振荡器
  - $f_R=200\text{kHz}$  (典型)
- 内部系统时钟发生
  - 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
- 时钟输出功能
- 可编程时钟 (PCL) 输出功能

**备注**       $f_x$ : 主时钟振荡频率  
               $f_{xx}$ : 主时钟频率  
               $f_R$ : 内部振荡时钟频率  
               $f_{XT}$ : 副时钟频率





**(1) 主时钟振荡器**

主谐振器振荡频率 ( $f_x$ ) 如下。

- 在时钟直通模式下  
 $f_x=4$  到 5MHz
- 在 PLL 模式  
 $f_x=4$  到 5MHz ( $f_{xx}=16$  到 20MHz)

**(2) 副时钟振荡器**

子谐振器振荡频率 ( $f_{xt}$ ) 为 32.768kHz 或 20kHz。

**(3) 主时钟振荡器停止控制**

该电路产生控制信号来使主时钟振荡器的振荡停止。

当处于 STOP 模式下或当 PCC.MCK 位=1 (当 PCC.CLS 位=1 时有效)，主时钟振荡器振荡停止。

**(4) 内部振荡器**

振荡频率 ( $f_R$ ) 为 200kHz (典型)。

**(5) 预换算器 1**

该电路为下列片上外围功能提供时钟 ( $f_{xx}$  到  $f_{xx}/1$ 、024)：TMP0 到 TMP3、TMQ0、TMM0、CSIB0、CSIB1、UARTA0、UARTA1、ADC 和 WDT2。

**(6) 预换算器 2**

该电路对主时钟 ( $f_{xx}$ ) 进行分频。

由预换算器 2 产生的时钟 ( $f_{xx}$  到  $f_{xx}/32$ ) 供给选择器，生成 CPU 时钟 ( $f_{CPU}$ ) 和内部系统时钟 ( $f_{CLK}$ )。

$f_{CLK}$  为供给 INTC、ROM 和 RAM 模块的时钟，可从 CLKOUT 引脚输出。

**(7) 预换算器 3**

该电路将主时钟振荡器产生的时钟 ( $f_x$ ) 分频为特定频率 (32.768kHz) 并将此时钟供给监视定时器模块。

详情请见“第九章监视定时器功能”。

**(8) 预换算器 4**

<R> 该电路产生的时钟 ( $f_x$  到  $f_x/128$ ) 供给片上外围功能。

供给的模块仅 WDT2。

**(9) PLL**

该电路将主时钟振荡器产生的时钟 ( $f_x$ ) 倍频为 4 倍。

它工作在两种模式下：时钟直通模式照原样输出  $f_x$ ，PLL 模式输出倍频后的时钟。可以通过 PLLCTL.SELPLL 位进行模式选择。

## 5.3 寄存器

### (1) 处理器时钟控制寄存器 (PCC)

PCC 寄存器为专用寄存器。数据写入此寄存器时必须符合特定序列的组合（见 **3.4.7 专用寄存器**）。

该寄存器可进行 8 位元或 1 位元的读写。

复位将此寄存器设置为 03H。

复位后: 03H 读/写 地址: FFFFF828H

PCC

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

FRC	使用子时钟片上反馈电阻器
0	使用
1	未使用

MCK	主时钟振荡器控制
0	启用振荡
1	振荡停止

- 当系统用主时钟作为CPU时钟进行工作时，即使设置MCK位为(1)，主时钟的工作也不会停止。在CPU时钟更改为子时钟时停止。
- 在将MCK位从0设置为1之前，应停止在主时钟下工作的片上外围功能。
- 当主时钟停止，且设备工作在子时钟下时，应将MCK清零(0)并在将CPU时钟切换到主时钟前或实现片上外围功能之前，应通过软件保证振荡稳定时间。

MFRC	使用主时钟片上反馈电阻器
0	使用
1	未使用

CLS <sup>注</sup>	CPU 时钟(f <sub>cpu</sub> )
0	主时钟工作
1	子时钟工作

CK3	CK2	CK1	CK0	时钟选择 (f <sub>clk</sub> /f <sub>cpu</sub> )
0	0	0	0	f <sub>xx</sub>
0	0	0	1	f <sub>xx</sub> /2
0	0	1	0	f <sub>xx</sub> /4
0	0	1	1	f <sub>xx</sub> /8
0	1	0	0	f <sub>xx</sub> /16
0	1	0	1	f <sub>xx</sub> /32
0	1	1	?	禁止设定
1	?	?	?	f <sub>xt</sub>

注 CLS 位为只读位。

- 注意事项
1. 当 CLKOUT 正在输出时不要改变 CPU 时钟（利用 CK3 到 CK0 位）。
  2. 利用位操作指导对 CK3 位进行操作。在使用 8 位操作指令时，不要改变 CK2 位到 CK0 位设定的值。

备注 ×: 无关

## (a) 主时钟工作→副时钟工作设置示例

- <1> CK3 位←1: 建议使用位操作指令。不要改变 CK2 到 CK0 位。
- <2> 副时钟工作: 读取 CLS 位检验副时钟是否开始工作。在 CK3 位设定好之后到副时钟工作开始之前, 需要经过以下时间。

最大:  $1/f_{XT}$  (1/副时钟频率)

- <3> MCK 位←1: 仅在停止主时钟时才将 MCK 位设置为 1。

- 注意事项 1. 在停止主时钟时, 停止 PLL。还应停止在主时钟下工作的片上外围功能的运行。
2. 若下列条件不满足, 应改变 CK2 到 CK0 位以使条件满足, 再改换到副时钟工作模式。

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

备注 内部系统时钟 ( $f_{CLK}$ ): 通过设置 CK2 到 CK0 位, 由主时钟 ( $f_{XX}$ ) 产生的时钟

[说明性示例]

```
<1> _SET_SUB_RUN :
    st.b      r0, PRCMD[r0]
    set1      3, PCC[r0]          -- CK3 位 ← 1

<2> _CHECK_CLS :
    tst1      4, PCC[r0]          -- 等待直到副时钟开始工作。
    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 上面是举例说明。要小心注意的是第<2>步中 CLS 位是在闭环中读取的。

## (b) 副时钟工作→主时钟工作设置示例

- <1> MCK 位←0:                   主时钟开始振荡
- <2> 通过程序插入等待，一直等到主时钟的振荡稳定时间过去。
- <3> CK3 位←0:                   建议使用位操作指令。不要更改 CK2 到 CK0 位。
- <4> 主时钟工作:                   在 CK3 位设定好之后到主时钟工作开始之前，需要经过以下时间。  
                                   最大:  $1/f_{XT}$  (1/副时钟频率)  
                                   因此，在设定 CK3 位为 0 后应紧接着插入一条 NOP 指令，或读取 CLS 位来  
                                   检查主时钟工作是否已经开始。

**注意事项** 只有在主时钟振荡稳定后才能启用在主时钟下工作的片上外围功能的运行。若在振荡稳定时间过去之前启用这些功能的运行，可能会导致故障。

[说明性示例]

```

<1> _START_MAIN_OSC :
    st.b      r0, PRCMD[r0]      -- 解除对专用寄存器的保护
    clr1      6, PCC[r0]         -- 主时钟开始振荡
<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]         -- 等待直到主时钟开始工作。
    bnz       _CHECK_CLS

```

**备注**       以上为举例说明。要小心注意的是在第<4>步中 CLS 位是在闭环中读取的。

**(2) 内部振荡模式寄存器 (RCM)**

RCM 寄存器为 8 位寄存器，设定内部振荡器的工作模式。

该寄存器可进行 8 位元或 1 位元的读写。

复位将此寄存器设置为 00H。

复位后: 00H 读/写 地址: FFFF80CH

	7	6	5	4	3	2	1	0
RCM	0	0	0	0	0	0	0	RSTOP

RSTOP	内部振荡器振荡/停止
0	内部振荡器振荡
1	内部振荡器停止

- 注意事项**
1. 通过设置选项字节可使 RCM 寄存器的设置生效。详情请见“第二十三章 选项字节功能”。
  2. 当 CPU 工作在内部振荡时钟下 (CCLS.CCLS F 位=1) 时无法停止内部振荡器。不要将 RSTOP 位设置为 1。
  3. 即便 RSTOP 位设置为 1，只要 CCLS.CCLS F 位设置为 1 (在振荡稳定过程中出现 WDT 溢出时)，内部振荡器就会振荡。此时，RSTOP 位保持为 1。

**(3) CPU 工作时钟状态寄存器 (CCLS)**

CCLS 寄存器指示 CPU 工作时钟的状态。

该寄存器只读，以 8 位或 1 位为单位。

复位将此寄存器设置为 00H。

复位后: 00H<sup>注</sup> 读 地址: FFFF82EH

	7	6	5	4	3	2	1	0
CCLS	0	0	0	0	0	0	0	CCLS F

CCLS F	CPU 工作时钟状态
0	工作在主时钟 (fx) 或子时钟 (fxr) 下。
1	工作在内部振荡时钟 (fxr) 下。

**注** 在复位解除后，若在振荡稳定过程中出现了 WDT 溢出，则 CCLS F 位被置为 1 且复位值为 01H。

## 5.4 运行

### 5.4.1 各时钟的工作

下表说明了各时钟的工作状态。

表 5-1 各时钟的工作状态

<R>

寄存器设定及工作状态   目标时钟	PCC 寄存器								
	CLK 位=0、MCK 位=0					CLS 位=1、 MCK 位=0		CLS 位=1、 MCK 位=1	
	复位时	振荡稳定时 间计数时	HALT 模 式	IDLE1、 IDLE2 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 (f <sub>x</sub> )	×	○	○	○	×	○	○	×	×
主系统时钟 (f <sub>xx</sub> )	×	×	○	×	×	○	×	×	×
副时钟振荡器 (f <sub>XT</sub> )	○	○	○	○	○	○	○	○	○
CPU 时钟 (f <sub>cpu</sub> )	×	×	×	×	×	○	×	○	×
内部系统时钟 (f <sub>clk</sub> )	×	×	○	×	×	○	×	○	×
主时钟 (在 PLL 模式下, f <sub>xx</sub> )	×	注 1	○	注 2	×	○	○	×	×
外围时钟 (f <sub>xx</sub> 到 f <sub>xx</sub> /1024)	×	×	○	×	×	○	×	×	×
WT 时钟 (主)	×	×	○	○	×	○	○	×	×
WT 时钟 (副)	○	○	○	○	○	○	○	○	○
WDT2 时钟 (内部振荡)	×	○	○	○	○	○	○	○	○
WDT2 时钟 (主)	×	×	○	×	×	○	×	×	×

注 1. 振荡稳定时间过去 1/2 后振荡开始, 在锁闭时间之后供给稳定的时钟。

2. 可在 IDLE1 模式下工作。在 IDLE2 模式下停止

备注 ○: 可工作

×: 停止

### 5.4.2 时钟输出功能

时钟输出功能用于从 CLKOUT 引脚输出内部系统时钟 (f<sub>clk</sub>)。

内部系统时钟 (f<sub>clk</sub>) 利用 PCC.CK3 到 PCC.CK0 位进行选择。

CLKOUT 引脚另外也可作为 PCM1 引脚使用, 若在端口 CM 的控制寄存器中指定, 也可实现时钟输出引脚的功能。

CLKOUT 引脚的状态与表 5-1 中内部系统时钟的相同, 在可工作状态下能够输出时钟。在停止状态下, 它输出低电平。但是, 复位后直到设置为输出模式之前, CLKOUT 引脚都处在端口模式 (PCM1 引脚: 输入模式) 下。因此, 引脚的状态为高阻 (Hi-Z)。



5.5 PLL 功能

5.5.1 概述

在 V850ES/HF2 中，可选择由 PLL 功能输出的振荡频率 4 倍的工作时钟或者选择时钟直通模式作为 CPU 或片上外围功能的工作时钟。

使用 PLL 功能时：

输入时钟=4 到 5MHz（输出：16 到 20MHz）

时钟直通模式：

输入时钟=4 到 5MHz（输出：4 到 5MHz）

5.5.2 寄存器

(1) PLL 控制寄存器 (PLLCTL)

PLLCTL 寄存器为 8 位寄存器，控制 PLL 功能。  
该寄存器可进行 8 位元或 1 位元的读写。  
复位将此寄存器设置为 01H。

复位后: 01H    读/写    地址: FFFFF82CH							
PLLCTL	7	6	5	4	3	2	1
	0	0	0	0	0	0	SELPLL   PLLON
PLLON		PLL操作停止寄存器					
0		PLL 停止					
1		PLL operating (在PLL 工作开始后，需要经过频率稳定的锁闭时间)					
SELPLL		CPU工作时钟选择寄存器					
0		Clock-through模式					
1		PLL模式					

- 注意事项
1. 当 PLLON 位清为 0 时，SELPLL 位自动清为 0（时钟直通模式）。
2. 仅当 PLL 时钟频率稳定时，SELPLL 位才可以设为 1。若未稳定（未锁定），则在写入数据时对 SELPLL 位写入“0”。

(2) 锁定寄存器 (LOCKR)

在加电后或 STOP 模式解除瞬间，相位锁定会以给定的频率发生，其稳定所需的时间为锁闭时间（频率稳定时间）。此稳定前的状态称为锁闭状态，而稳定后的状态称为锁定状态。

LOCKR 寄存器包含一个 LOCK 位，反映 PLL 频率稳定的状态。

该寄存器只读，以 8 位或 1 位为单位。

复位将此寄存器设置为 00H。

复位后: 00H    读    地址: FFFF824H							
LOCKR	7	6	5	4	3	2	1
	0	0	0	0	0	0	0
LOCK							
PLL锁定状态检查							
0    锁定状态							
1    未锁定状态							

注意事项 Lock 寄存器不会实时反映 PLL 的锁定状态。设置/清零的条件如下。

[设置条件]

- 系统复位时<sup>※</sup>
- 在 IDLE2 或 STOP 模式下
- 设置 PLL 停止时（PLLCTL.PLLON 位清为 0）
- 停止主时钟并在副时钟下使用 CPU（设置 PCC.CK3 位为 1 及 PCC.MCK 位为 1）时

注     寄存器经过复位会被设为 01H，在复位解除且振荡稳定时间过去之后，会清零为 00H。

[清零条件]

- 复位解除后振荡稳定时间溢出（OSTS 寄存器默认时间（见 16.2（3）振荡稳定时间选择寄存器（OSTS））后
- 当 STOP 模式在 PLL 工作状态下得到设置时，在 STOP 模式解除后振荡稳定定时器溢出（由 OSTS 寄存器设置的时间）后
- 当 PLLCTL.PLLON 位从 0 变为 1 时，PLL 锁闭时间定时器溢出（由 PLLS 寄存器设置的时间）后
- 当 IDLE2 模式在 PLL 工作过程中得到设置时，IDLE2 模式解除后插入的设立时间（由 OSTS 寄存器设置的时间）解除后

(3) PLL 锁闭时间指定寄存器 (PLLS)

PLLS 寄存器为 8 位寄存器，用来在 PLLCTL.PLLON 位从 0 变为 1 时选择 PLL 锁闭时间。  
该寄存器可进行 8 位读写。  
复位将此寄存器设置为 03H。

复位后：03H 读/写 地址： FFFFF6C1H

	7	6	5	4	3	2	1	0
PLLS	0	0	0	0	0	0	PLLS1	PLLS0

PLLS1	PLLS0	PLL锁闭时间的选择
0	0	$2^{10}/f_x$
0	1	$2^{11}/f_x$
1	0	$2^{12}/f_x$
1	1	$2^{13}/f_x$ (默认值)

注意事项 1. 通过设置使锁闭时间为 800 s 或以上。  
2. 在锁闭期内不要改变 PLLS 寄存器的设置。

备注 fxx: 主时钟振荡频率

(4) 可编程时钟模式寄存器 (PCLM)

PCLM 寄存器为 8 位寄存器，用来控制 PCL 输出。  
该寄存器可进行 8 位元或 1 位元的读取或写入。

复位后: 00H    读/写    地址: FFFF82FH

	7	6	5	4	3	2	1	0
PCLM	0	0	0	PCLE	0	0	PCK1	PCK0

PCLE	PCL引脚输出操作的选择
0	禁用PCL引脚输出 (PCL引脚固定为低电平)
1	启用PCL引脚输出

注意事项 首先设置端口相关控制寄存器 (PM、PMC、PFC、PFCE 寄存器等等)，再将 PCLE 位设置为 1。

<R>

PCK1	PCK0	PLL输出时钟的选择
0	0	f <sub>PCL</sub> /2
0	1	f <sub>PCL</sub> /4
1	0	f <sub>PCL</sub> /8
1	1	f <sub>PCL</sub> /16

注意事项 仅在 PLL 工作过程中才能设置 PCLE 位。要停止 PLL，应将 PCLE 位清为 0。

备注        f<sub>PCL</sub>: 可编程频率

### 5.5.3 用法

#### (1) 当使用 PLL 时

- 在复位信号解除后，PLL 工作（PLLCTL.PLLON 位=1），但因为默认模式为时钟直通模式（PLLCTL.SELPLL 位=0），所以应选择为 PLL 模式（SELPLL 位=1）。
- 要启用 PLL 的工作，首先应将 PLLON 位设置为 1，然后在 LOCKR.LOCK 位=0 后将 SELPLL 位设置为 1。要停止 PLL，首先应选择时钟直通模式（SELPLL 位=0），等待 8 个时钟或以上，再停止 PLL（PLLON 位=0）。
- 在转换到 IDLE2 或 STOP 模式时，PLL 停止（与设定无关），并从 IDLE2 或 STOP 模式恢复到转换前的状态。恢复所需时间如下。

##### (a) 在从时钟直通模式转换到 IDLE2 或 STOP 模式时

- STOP 模式：设置 OSTS 寄存器使振荡稳定时间为 1ms（最小）或以上。
- IDLE2 模式：设置 OSTS 寄存器使设立时间为 350 $\mu$ s（最小）或以上。

##### (b) 在保持 PLL 工作模式情况下变换到 IDLE2 或 STOP 模式

- STOP 模式：设置 OSTS 寄存器使振荡稳定时间为 1ms（最小）或以上。
- IDLE2 模式：设置 OSTS 寄存器使设立时间为 800 $\mu$ s（最小）或以上。

当变换到 IDLE1 模式时，PLL 不停止。若有必要，应停止 PLL。

#### (2) 当未使用 PLL 时

- 在复位信号解除后，时钟直通模式（SELPLL 位=0）被选中，但 PLL 正在工作（PLLON 位=1），所以必须停止 PLL（PLLON 位=0）。

## 第六章 16 位定时器/事件计数器 P (TMP)

定时器 P (TMP) 为 16 位定时器/事件计数器。

V850ES/HF2 有四个定时器/事件计数器通道，为 TMP0 到 TMP3。

### 6.1 概述

TMPn 的大概情况如下所示。

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

备注      n=0 到 3

### 6.2 功能

TMPn 具有下列功能。

- 时间间隔定时器
- 外部事件计数器
- 外部触发脉冲输出
- 单次脉冲输出
- PWM 输出
- 独立定时器
- 脉冲宽度测量

备注      n=0 到 3

6.3 配置

TMPn 包含下列硬件。

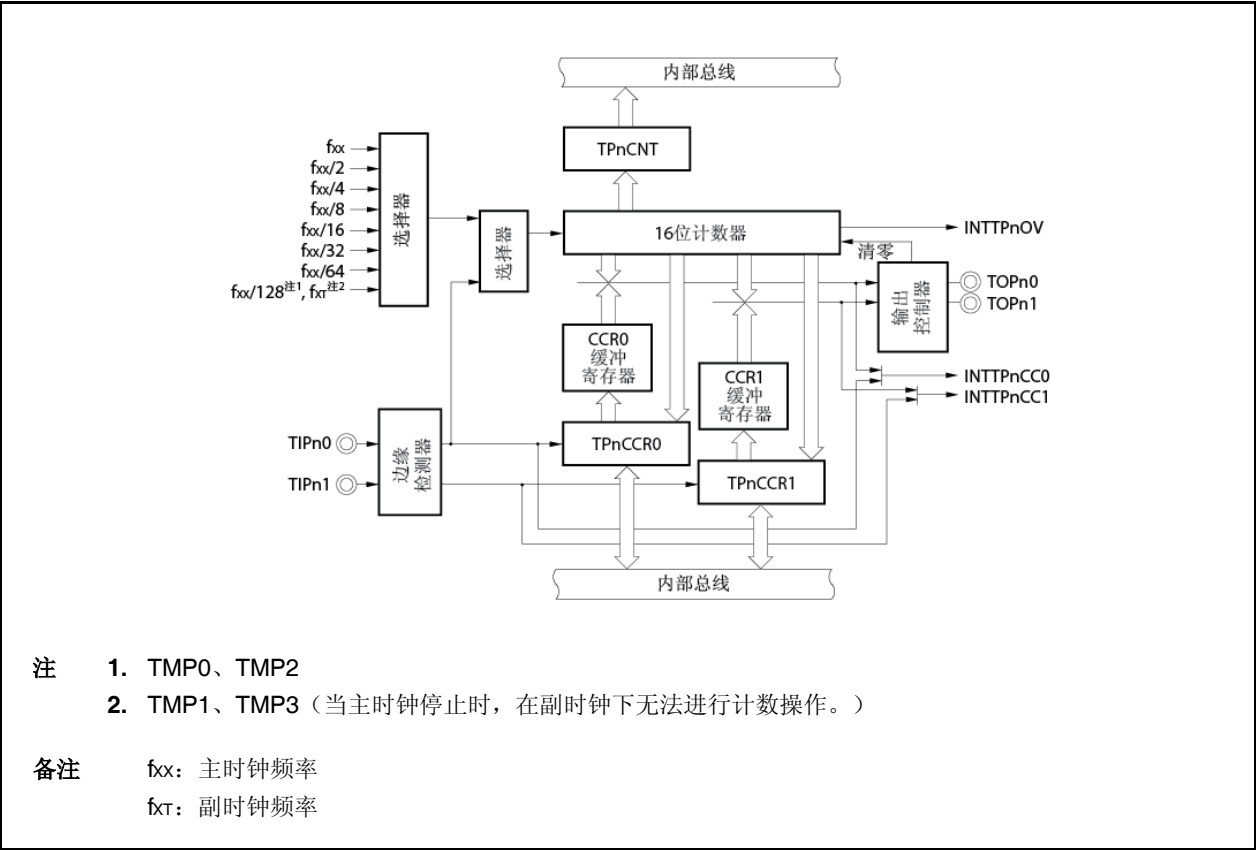
表 6-1 TMPn 的配置

项目	配置
定时器寄存器	16 位计数器
寄存器	TMPn 捕捉/比较寄存器 0、1 (TPnCCR0、TPnCCR1) TMPn 计数器读取缓冲寄存器 (TPnCNT) CCR0、CCR1 缓冲寄存器
定时器输入	2 (TIPn0 <sup>注1</sup> 、TIPn1 引脚)
定时器输出	2 (TOPn0、TOPn1 引脚)
控制寄存器 <sup>注2</sup>	TMPn 控制寄存器 0、1 (TPnCTL0、TPnCTL1) TMPn I/O 控制寄存器 0 到 2 (TPnIOC0 到 TPnIOC2) TMPn 选项寄存器 0 (TPnOPT0)

- 注
- 1. Tipn0 引脚交替实现下列功能：捕捉触发输入信号、外部事件计数输入信号、外部触发输入信号。
  - 2. 当使用 TIPn0、TIPn1、TOPn0、TOPn1 引脚的功能时，应参见“表 4-14 用端口引脚作为复用功能引脚”。

备注 n=0 到 3

图 6-1 TMPn 框图



**(1) 16 位计数器**

此 16 位计数器可以给内部时钟或外部事件计数。

此计数器的计数数值可通过 TPnCNT 寄存器读取。

当 TPnCTL0.TpnCE 位=0 时，16 位计数器的值为 FFFFH。若此时读取 TPnCNT 寄存器，则会读取 0000H。

复位将 TPnCE 位设置为 0。因此，16 位计数器设置为 FFFFH。

**(2) CCR0 缓冲寄存器**

该 16 位比较寄存器用来比较 16 位计数器的计数值。

当使用 TPnCCR0 寄存器作为比较寄存器时，写入 TPnCCR0 寄存器的值会传送到 CCR0 缓冲寄存器中。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号（INTTPnCC0）。

Ccr0 缓冲寄存器无法直接读写。

Ccr0 缓冲寄存器在复位后清零为 0000H，因为 TPnCCR0 寄存器清零为 0000H。

**(3) CCR1 缓冲寄存器**

该 16 位比较寄存器用来比较 16 位计数器的计数值。

当 TPnCCR1 寄存器用作比较寄存器时，写入 TPnCCR1 寄存器的值会传送到 CCR1 缓冲寄存器。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号（INTTPnCC1）。

Ccr1 缓冲寄存器不能直接读写。

Ccr1 缓冲寄存器在复位后清零为 0000H，因为 TPnCCR1 寄存器清零为 0000H。

**(4) 边缘检测器**

此电路检测输入到 TIPn0 和 TIPn1 引脚的有效边缘。可以用 TPnIOC1 和 TPnIOC2 寄存器选择无边缘、上升边缘、下降边缘或者上升下降边缘之一作为有效边缘。

**(5) 输出控制器**

此电路控制 TOPn0 和 TOPn1 引脚的输出。输出控制器由 TPnIOC0 寄存器控制。

**(6) 选择器**

此选择器选择 16 位计数器的计数时钟。可选择八种内部时钟或外部事件作为计数时钟。



## 6.4 寄存器

控制 TMPn 的寄存器如下所示。

- TMPn 控制寄存器 0 (TPnCTL0)
- TMPn 控制寄存器 1 (TPnCTL1)
- TMPn I/O 控制寄存器 0 (TPnIOC0)
- TMPn I/O 控制寄存器 1 (TPnIOC1)
- TMPn I/O 控制寄存器 2 (TPnIOC2)
- TMPn 选项寄存器 0 (TPnOPT0)
- TMPn 捕捉/比较寄存器 0 (TPnCCR0)
- TMPn 捕捉/比较寄存器 1 (TPnCCR1)
- TMPn 计数器读取缓冲寄存器 (TPnCNT)

- 备注
1. 当使用 TIPn0、TIPn1、TOPn0、TOPn1 引脚的功能时，应参见“表 4-14 用端口引脚作为复用功能引脚”。
  2. n=0 到 3

(1) TMPn 控制寄存器 0 (TPnCTL0)

TPnctl0 寄存器为 8 位寄存器，控制 TMPn 的工作。

该寄存器可进行 8 位元或 1 位元的读写。

复位将此寄存器设置为 00H。

总是可以用软件将相同值写入 TPnCTL0 寄存器。

复位后: 00H 读/写 地址: TP0CTL0 FFFFF590H, TP1CTL0 FFFFF5A0H,  
TP2CTL0 FFFFF5B0H, TP3CTL0 FFFFF5C0H

	7	6	5	4	3	2	1	0
TPnCTL0	TPnCE	0	0	0	0	TPnCKS2	TPnCKS1	TPnCKS0

(n = 0 至 3)

TPnCE	TMPn操作控制
0	禁用TMPn操作 (TMPn异步复位 <sup>注1</sup> )
1	启用TMPn操作。TMPn操作开始。

TPnCKS2	TPnCKS1	TPnCKS0	内部振荡时钟的选择	
			n = 0, 2	n = 1, 3
0	0	0	f <sub>xx</sub>	
0	0	1	f <sub>xx</sub> /2	
0	1	0	f <sub>xx</sub> /4	
0	1	1	f <sub>xx</sub> /8	
1	0	0	f <sub>xx</sub> /16	
1	0	1	f <sub>xx</sub> /32	
1	1	0	f <sub>xx</sub> /64	
1	1	1	f <sub>xx</sub> /128	f <sub>XT</sub> <sup>注2</sup>

- 注 1. TPnOPT0.TPnOVF 位、16 位计数器、定时器输出 (TOPn0、TOPn1 引脚)  
2. 当主时钟停止时，在副时钟下无法进行计数操作。

- 注意事项 1. 在 TPnCE 位=0 时设定 TPnCKS2 到 TPnCKS0 位。  
当 TPnCE 位的值从 0 变为 1 时，TPnCKS2 到 TPnCKS0 位可以同时设定。  
2. 一定要将 3 到 6 位清为“0”。

备注 f<sub>xx</sub>: 主时钟频率  
f<sub>XT</sub>: 副时钟频率

(2) TMPn 控制寄存器 1（TPnCTL1）

Tpnctl1 寄存器是控制 TMPn 工作的 8 位寄存器。  
该寄存器可进行 8 位元或 1 位元的读取或写入。  
复位将此寄存器设置为 00H。

(1/2)

复位后: 00H      读/写      地址:

TP0CTL1 FFFFF591H, TP1CTL1 FFFFF5A1H,  
TP2CTL1 FFFFF5B1H, TP3CTL1 FFFFF5C1H

7	6	5	4	3	2	1	0
TPnSYE	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0

  
(n = 0 至 3)

TPnSYE	调谐工作模式启用控制																											
0	独立工作模式（异步工作模式）																											
1	调谐工作模式（指定从属的操作） 此模式下，定时器P可在主定时器同步下工作。 <table><tr><td>主定时器</td><td colspan="6">从定时器</td></tr><tr><td>TMP0</td><td>TMP1</td><td colspan="5">—</td></tr><tr><td>TMP2</td><td>TMP3</td><td colspan="5">TMQ0</td></tr></table>							主定时器	从定时器						TMP0	TMP1	—					TMP2	TMP3	TMQ0				
主定时器	从定时器																											
TMP0	TMP1	—																										
TMP2	TMP3	TMQ0																										
关于调谐工作模式，请参见“6. 6定时器调谐工作功能”																												
注意事项 一定要将TP0SYE、TP2SYE位清为0。																												

TPnEST	软件触发控制						
0	—						
1	生成外部触发输入的有效信号。 ●在 单次脉冲输出模式：通过向TPnEST标志写入1作为 触发，可输出单个脉冲。 ●在外部触发脉冲输出模式下：通过向TPnEST位写入1作为触发 来输出PWM波形。						

- 注意事项 1. TPnEST 位仅在外部触发脉冲输出模式或单次脉冲输出模式下有效。在其它模式下，忽略对该位写入 1 的动作。
2. 一定要将 3 到 4 位清为“0”。

TPnEEE	计数时钟的选择
0	禁用外部事件计数输入的工作。 (根据由TPnCTL0.TPnCK0 至 TPnCK2位选择的计数时钟进行计数。)
1	启用外部事件计数输入的工作。 (在外部事件计数输入信号的有效边缘进行计数)
TPnEEE位选择进行计数时是按照内部计数时钟 还是按照外部事件计数输入的有效边缘。	

TPnMD2	TPnMD1	TPnMD0	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	单次脉冲输出模式
1	0	0	PWM输出模式
1	0	1	独立定时器模式
1	1	0	脉冲宽度测量模式
1	1	1	禁止设置

- 注意事项
1. 在外部事件计数模式下，外部事件计数输入被选中且与 TPnEEE 位的值无关。
  2. 应在 TPnCTL0.TPnCE 位=0 时设置 TPnEEE、TPnMD2 到 TPnMD0 位。  
(TPnCE 位=1 时可写入相同的值。)当 TPnCE 位=1 时，无法保证重写操作的进行。如果进行了错误的重写操作，应先将 TPnCE 位清为 0 再重新设置这些位。

&lt;R&gt;

**(3) TMPnI/O 控制寄存器 0 (TPnIOC0)**

TPnIOC0 寄存器为 8 位寄存器，控制定时器输出 (TOPn0、TOPn1 引脚)。

该寄存器可进行 8 位元或 1 位元的读取或写入。

复位将此寄存器设置为 00H。

复位后: 00H

读/写

地址: TP0IOC0 FFFF592H, TP1IOC0 FFFF5A2H,  
TP2IOC0 FFFF5B2H, TP3IOC0 FFFF5C2H

TPnIOC0	7	6	5	4	3	2	1	0
(n = 0 至 3)	0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0

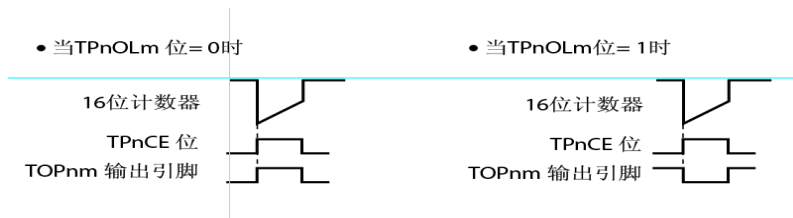
TPnOL1	TOPn1 引脚输出电平设定 <sup>注</sup>
0	TOPn1 引脚输出从高水平开始。
1	TOPn1 引脚输出从低电平开始。

TPnOE1	TOPn1 引脚输出设定
0	禁止定时器输出 当TPnOL1位=0时: 从TOPn1引脚输出低电平。 • 当TPnOL1 位 = 1时, 从TOPn1 引脚输出高电平
1	启用定时器输出 (从TOPn1引脚输出方波)。

TPnOL0	TOPn0 引脚输出电平设定 <sup>注</sup>
0	TOPn0引脚输出从高水平开始。
1	TOPn0引脚输出从低电平开始。

TPnOE0	TOPn0 引脚输出设定
0	禁用定时器输出 • 当TPnOL0 位 = 0时, 从TOPn0 引脚输出低电平 • 当TPnOL0 位 = 1时, 从TOPn0 引脚输出高电平
1	启用定时器输出 (从TOPn0引脚输出方波)。

**注** 由 TPnOLm 位指定的定时器输出引脚 (TOPnm) 的输出电平如下所示。



- 注意事项**
1. 应在 TPnCTL0.TPnCE 位 = 0 时重写 TPnOL1、TPnOE1、TPnOL0、TPnOE0 位。(TPnCE 位 = 1 时可写入相同的值。) 如果进行了错误的重写操作, 应先将 TPnCE 位清为 0 再重新设置这些位。
  2. 当 TPnCE 和 TPnOEm 位为 0 时, 即便对 TPnOLm 位进行人为操作, TOPnm 引脚输出电平也会变化。

**备注** n=0 到 3、m=0、1

(4) TMPn I/O 控制寄存器 1 (TPnIOC1)

Tpnioc1 寄存器为 8 位寄存器，用来控制捕捉触发输入信号的有效边缘 (TIPn0、TIPn1 引脚)。  
该寄存器可进行 8 位元或 1 位元的读取或写入。  
复位将此寄存器设置为 00H。

复位后: 00H

读/写

地址: TP0IOC1 FFFFF593H, TP1IOC1 FFFFF5A3H,  
TP2IOC1 FFFFF5B3H, TP3IOC1 FFFFF5C3H

	7	6	5	4	3	2	1	0
TPnIOC1	0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0

(n = 0 to 3)

TPnIS3	TPnIS2	捕捉触发输入信号(TIPn1引脚)有效边缘的设置
0	0	无边缘检测 (捕捉操作无效)
0	1	检测上升边缘
1	0	检测下降边缘
1	1	检测上升下降边缘

TPnIS1	TPnIS0	捕捉触发输入信号(TIPn0引脚)有效边缘的设置
0	0	无边缘检测 (捕捉操作无效)
0	1	检测上升边缘
1	0	检测下降边缘
1	1	检测上升下降边缘

- 注意事项
1. 应在 TPnCTL0.TPnCE 位=0 时重写 TPnIS3 到 TPnIS0 位。  
(TPnCE 位=1 时可写入相同的值。) 如果进行了错误的重写操作，应先将 TPnCE 位清为 0 再重新设置这些位。
  2. Tpnis3 到 TPnIS0 位仅在独立定时器模式和脉宽测量模式下有效。在所有其它模式下，无法进行捕捉操作。

## (5) TMPn I/O 控制寄存器 2 (TPnIOC2)

TPnIOC2 寄存器为 8 位寄存器，用来控制外部事件计数输入信号 (TIPn0 引脚) 和外部触发输入信号 (TIPn0 引脚) 的有效边缘。

该寄存器可进行 8 位元或 1 位元的读取或写入。

复位将此寄存器设置为 00H。

复位后: 00H

读/写

地址: TP0IOC2 FFFFFFF594H, TP1IOC2 FFFFFFF5A4H,  
TP2IOC2 FFFFFFF5B4H, TP3IOC2 FFFFFFF5C4H

	7	6	5	4	3	2	1	0
TPnIOC2	0	0	0	0	TPnEES1	TPnEES0	TPnETS1	TPnETS0

(n = 0 至3)

TPnEES1	TPnEES0	外部事件计数输入信号 (TIPn0引脚) 有效边缘设置
0	0	无边缘检测 (外部事件计数无效)
0	1	检测上升边缘
1	0	检测下降边缘
1	1	检测上升下降边缘

TPnETS1	TPnETS0	捕捉触发输入信号 (TIPn0引脚) 有效边缘的设置
0	0	无边缘检测 (外部触发无效)
0	1	上升边缘的检测
1	0	下降边缘的检测
1	1	上升下降边缘的检测

- 注意事项**
1. 应在 TPnCTL0.TPnCE 位=0 时重写 TPnEES1、TPnEES0、TPnETS1、TPnETS0 位。(TPnCE 位=1 时可写入相同的值。) 如果进行了错误的重写操作，应先将 TPnCE 位清为 0 再重新设置这些位。
  2. 仅当 TPnCTL1.TPnEEE 位=1 或已设置为外部事件计数模式 (TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0 位=001) 时 TPnEES1 和 TPnEES0 位才有效。
  3. 仅当设置为外部触发脉冲输出模式 (TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0 位=010) 或单次脉冲输出模式 (TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0=011) 时 TPnETS1 和 TPnETS0 位才有效。

(6) TMPn 选项寄存器 0 (TPnOPT0)

Tpnopt0 寄存器为 8 位寄存器，用来设定捕捉/比较操作并对溢出进行检测。  
该寄存器可进行 8 位元或 1 位元的读取或写入。  
复位将此寄存器设置为 00H。

复位后: 00H      读/写      地址: TP0OPT0 FFFF595H, TP1OPT0 FFFF5A5H,  
TP2OPT0 FFFF5B5H, TP3OPT0 FFFF5C5H

	7	6	5	4	3	2	1	0
TPnOPT0 (n = 0 至 3)	0	0	TPnCCS1	TPnCCS0	0	0	0	TPnOVF

TPnCCS1	TPnCCR1 寄存器捕捉/比较选择
0	选择比较寄存器
1	选择捕捉寄存器
只有在独立定时器模式下 TPnCCS1 位的设置才有效。	

TPnCCS0	TPnCCR0 寄存器捕捉/比较选择
0	选择比较寄存器
1	选择捕捉寄存器
TPnCCS0 位的设定仅在独立定时器模式下有效。	

TPnOVF	TMPn 溢出检测标志
设置 (1)	出现溢出
复位 (0)	TPnOVF 位写入 0 或 TPnCTL0.TPnCE 位 = 0
<ul style="list-style-type: none"> <li>在独立定时器模式或脉冲宽度测量模式下，16 位计数器计数值从 FFFFH 溢出到 0000H 时，TPnOVF 位设置为 1。</li> <li>当 TPnOVF 位设置为 1 时，同时产生中断请求信号 (INTTPNOV)。INTTPNOV 信号不会在独立定时器模式、脉宽测量模式之外的模式下生成。</li> <li>当 TPnOVF 位 = 1 时，即使在读取 TPnOVF 位或 TPnOPT0 寄存器时，TPnOVF 位也不会清零。</li> </ul> TPnOVF 位既可读又可写，但 TPnOVF 位无法由软件设置为 1。 写入 1 对 TMPn 的工作无影响。	

- 注意事项
1. 应在 TPnCE 位 = 0 时重写 TPnCCS1 和 TPnCCS0 位。  
(TPnCE 位 = 1 时可写入相同的值。) 如果进行了错误的重写操作，应先将 TPnCE 位清为 0 再重新设置这些位。
  2. 一定要将 1 到 3 位、6 位、7 位清为“0”。



(7) TMPn 捕捉/比较寄存器 0 (TPnCCR0)

根据模式的不同，TPnCCR0 寄存器可用作捕捉寄存器或比较寄存器。  
根据 TPnOPT0.TPnCCS0 位的设定，此寄存器仅可在独立定时器模式下用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TPnCCR0 寄存器只能用作捕捉寄存器。在其它模式下，此寄存器只能用作比较寄存器。  
Tpnccr0 寄存器可在工作过程中进行读写。  
此寄存器可进行 16 位读写。  
复位将此寄存器设置为 0000H。

**注意事项** 下列情况下禁止对 TPnCCR0 寄存器的访问。详情请见“3.4.8 (2) 访问专用片上外围 I/O 寄存器”。

- CPU 在副时钟下工作，主时钟振荡停止
- CPU 在内部振荡时钟下工作



(a) 用作比较寄存器

即便 TPnCTL0.TPnCE 位=1，TPnCCR0 寄存器也可以重写。

Tpnccr0 寄存器的设置值会传送到 CCR0 缓冲寄存器。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号（INTTPnCC0）。若此时启用了 TOPn0 引脚输出，TOPn0 引脚的输出会反相。

在时间间隔定时器模式、外部事件计数模式、外部触发脉冲输出模式、单次脉冲输出模式或 PWM 输出模式下，当 TPnCCR0 寄存器作为循环寄存器使用时，如果 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配，则 16 位计数器的值会清零（0000H）。

(b) 用作捕捉寄存器

当 TPnCCR0 寄存器在独立定时器模式下用作捕捉寄存器时，如果检测到捕捉触发输入引脚（TIPn0 引脚）的有效边缘，16 位计数器的计数值会存储在 TPnCCR0 寄存器中。在脉宽测量模式下，若检测到捕捉触发输入引脚（TIPn0）的有效边缘，则 16 位计数器的计数值存储在 TPnCCR0 寄存器中且 16 位计数器清零（0000H）。

即便捕捉操作与对 TPnCCR0 寄存器的读取相冲突，仍然能读取 TPnCCR0 寄存器的正确值。

下表展示了各模式下的捕捉/比较寄存器的功能及如何向比较寄存器写入数据。

表 6-2 各模式下捕捉/比较寄存器的功能、如何向比较寄存器写入数据

工作模式	捕捉/比较寄存器	如何写入比较寄存器
时间间隔定时器	比较寄存器	随时写
外部事件计数器	比较寄存器	随时写
外部触发脉冲输出	比较寄存器	批量写
单次脉冲输出	比较寄存器	随时写
PWM 输出	比较寄存器	批量写
独立定时器	捕捉/比较寄存器	随时写
脉冲宽度测量	捕捉寄存器	—

(8) TMPn 捕捉/比较寄存器 1 (TPnCCR1)

根据模式的不同，TPnCCR1 寄存器可用作捕捉寄存器或比较寄存器。

根据 TPnOPT0.TPnCCS1 位的设定，此寄存器仅可在独立定时器模式下用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TPnCCR1 寄存器只能用作捕捉寄存器。在其它模式下，此寄存器只能用作比较寄存器。

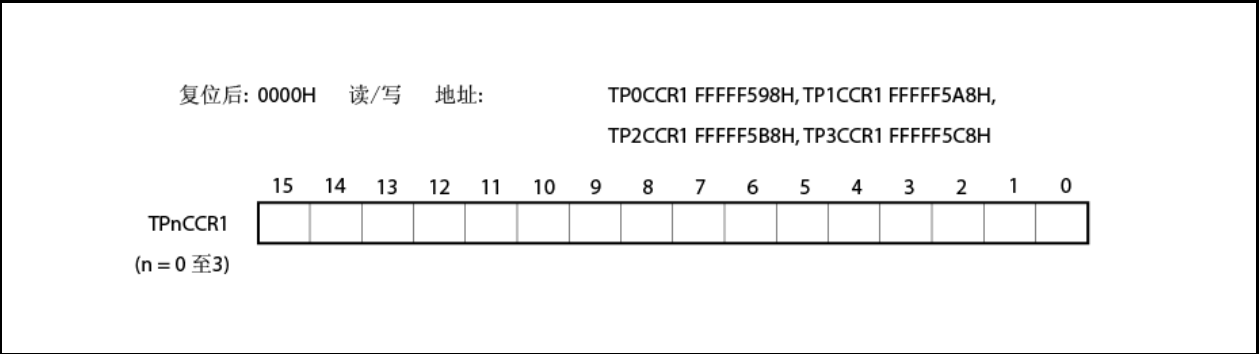
TPnCCR1 寄存器可在工作过程中进行读写。

此寄存器可进行 16 位读写。

复位将此寄存器设置为 0000H。

**注意事项** 下列情况下禁止对 TPnCCR1 寄存器的访问。详情请见“3.4.8 (2) 访问专用片上外围 I/O 寄存器”。

- CPU 在副时钟下工作，主时钟振荡停止
- CPU 在内部振荡时钟下工作



(a) 用作比较寄存器

TPnCCR1 寄存器即使在 TPnCTL0.TPnCE 位=1 时也可以重写。

TPnCCR1 寄存器的设定值传送给 CCR1 缓冲寄存器。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号（INTTPnCC1）。若此时启用了 TOPn1 引脚输出，TOPn1 引脚的输出会反相。

(b) 用作捕捉寄存器

当 TPnCCR1 寄存器在独立定时器模式下用作捕捉寄存器时，如果检测到捕捉触发输入引脚（TIPn1 引脚）的有效边缘，16 位计数器的计数值会存储在 TPnCCR1 寄存器中。在脉宽测量模式下，若检测到捕捉触发输入引脚（TIPn1）的有效边缘，则 16 位计数器的计数值存储在 TPnCCR1 寄存器中且 16 位计数器清零（0000H）。

即便捕捉操作与对 TPnCCR1 寄存器的读取相冲突，仍然能读取 TPnCCR1 寄存器的正确值。

下表展示了各模式下的捕捉/比较寄存器的功能及如何向比较寄存器写入数据。

表 6-3 各模式下捕捉/比较寄存器的功能、如何向比较寄存器写入数据

工作模式	捕捉/比较寄存器	如何写入比较寄存器
时间间隔定时器	比较寄存器	随时写
外部事件计数器	比较寄存器	随时写
外部触发脉冲输出	比较寄存器	批量写
单次脉冲输出	比较寄存器	随时写
PWM 输出	比较寄存器	批量写
独立定时器	捕捉/比较寄存器	随时写
脉冲宽度测量	捕捉寄存器	—

(9) TMPn 计数器读取缓冲寄存器 (TPnCNT)

TPnCNT 寄存器为读取缓冲寄存器，可以读取 16 位计数器的计数值。  
若在 TPnCTL0.TPnCE 位=1 时读取此寄存器，则可以读取 16 位计数器的计数值。  
此寄存器只读，16 位。  
当 TPnCE 位=0 时，TPnCNT 寄存器的值清为 0000H。若此时读取 TPnCNT 寄存器，则不会读取 16 位计数器的值 (FFFFH)，而会读取 0000H。  
TPnCNT 寄存器的值在复位后清零为 0000H，因为 TPnCE 位清为 0。

注意事项 下列情况下禁止对 TPnCNT 寄存器的访问。详情请见“3.4.8 (2) 访问专用片上外围 I/O 寄存器”。

- CPU 在副时钟下工作，主时钟振荡停止
- CPU 在内部振荡时钟下工作



(10) **TIPnm 引脚消噪控制寄存器 (PnmNFC)**

PnmNFC 寄存器为 8 位寄存器，用来设置定时器 P 输入引脚的数字静噪滤波器进行消噪。  
该寄存器可进行 8 位元或 1 位元的读取或写入。  
复位将此寄存器设置为 00H。

复位后：00H      读/写    地址： P00NFCFFFFFFB00H (TIP00 引脚)  
P01NFCFFFFFFB04H (TIP01 引脚)  
P10NFCFFFFFFB08H (TIP10 引脚)  
P11NFCFFFFFFB0CH (TIP11 引脚)  
P20NFCFFFFFFB10H (TIP20 引脚)  
P21NFCFFFFFFB14H (TIP21 引脚)  
P30NFCFFFFFFB18H (TIP30 引脚)  
P31NFCFFFFFFB1CH (TIP31 引脚)

	7	6	5	4	3	2	1	0
PnmNFC	0	NFSTS	0	0	0	NFC2	NFC1	NFC0

(n=0 到 3、m=0、1)

NFSTS	数字静噪滤波器采样次数的设定
0	3 次
1	2 次

NFC2	NFC1	NFC0	采样时钟	
			n=0、2	n=1、3
0	0	0	$f_{xx}$	
0	0	1	$f_{xx}/2$	
0	1	0	$f_{xx}/4$	
0	1	1	$f_{xx}/16$	$f_{xx}/8$
1	0	0	$f_{xx}/32$	$f_{xx}/16$
1	0	1	$f_{xx}/64$	$f_{XT}$
其它情况			禁止设定	

- 注意事项**
- 一定要将 3 到 5 位、7 位清为“0”。
  - 在设定 PnmNFC 寄存器前输入定时器输入引脚 (TIPnm) 的信号输出时会消除数字噪声。  
因此，应通过 PnmNFC 寄存器设置采样时钟 (NFC2 到 NFC0) 和采样次数 (NFSTS)，等待初始化时间 = (采样时钟) × (采样次数)，再启用定时器。

**备注**      可准确消除的噪声宽度为 (采样时钟) × (采样次数-1)。宽度小于该值的噪声如果与采样时钟同步也可能导致计数错误。

## 6.5 操作

TMPn 可进行下列操作。

操作	TPnCTL1.TPnEST 位 (软件触发位)	TIPn0 引脚 (外部触发输入)	捕捉/比较寄存器的设定	写入比较寄存器
时间间隔定时器模式	无效	无效	仅比较	随时写
外部事件计数模式 <sup>注1</sup>	无效	无效	仅比较	随时写
外部触发脉冲输出模式 <sup>注2</sup>	有效	有效	仅比较	批量写
单次脉冲输出模式 <sup>注2</sup>	有效	有效	仅比较	随时写
PWM 输出模式	无效	无效	仅比较	批量写
独立定时器模式	无效	无效	启用切换	随时写
脉宽测量模式 <sup>注2</sup>	无效	无效	仅捕捉	不适用

- 注**
1. 要使用外部事件计数模式，应指定不检测 TIPn0 引脚捕捉触发输入的有效边缘（通过将 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位清零为“00”）。
  2. 当使用外部触发脉冲输出模式、单次脉冲输出模式、脉宽测量模式时，应选择内部时钟作为计数时钟（通过将 TPnCTL1.TPnEEE 位清为 0）。

**备注**     n=0 到 3

6.5.1 时间间隔定时器模式 (TPnMD2 到 TPnMD0 位=000)

在时间间隔定时器模式下，若将 TPnCTL0.TPnCE 位设置为 1，则会生成指定时间间隔的中断请求信号 (INTTPnCC0)。可从 TOPn0 引脚输出半周期等于该时间间隔的方波。

通常，在时间间隔定时器模式下不使用 TPnCCR1 寄存器。

图 6-2 时间间隔定时器的配置

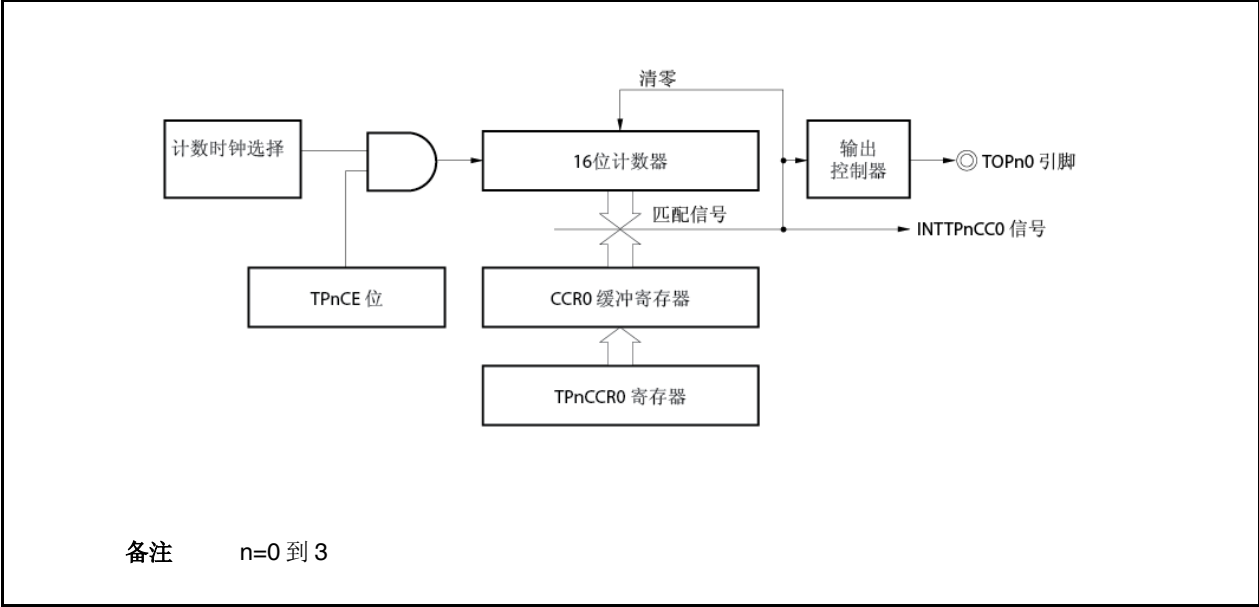
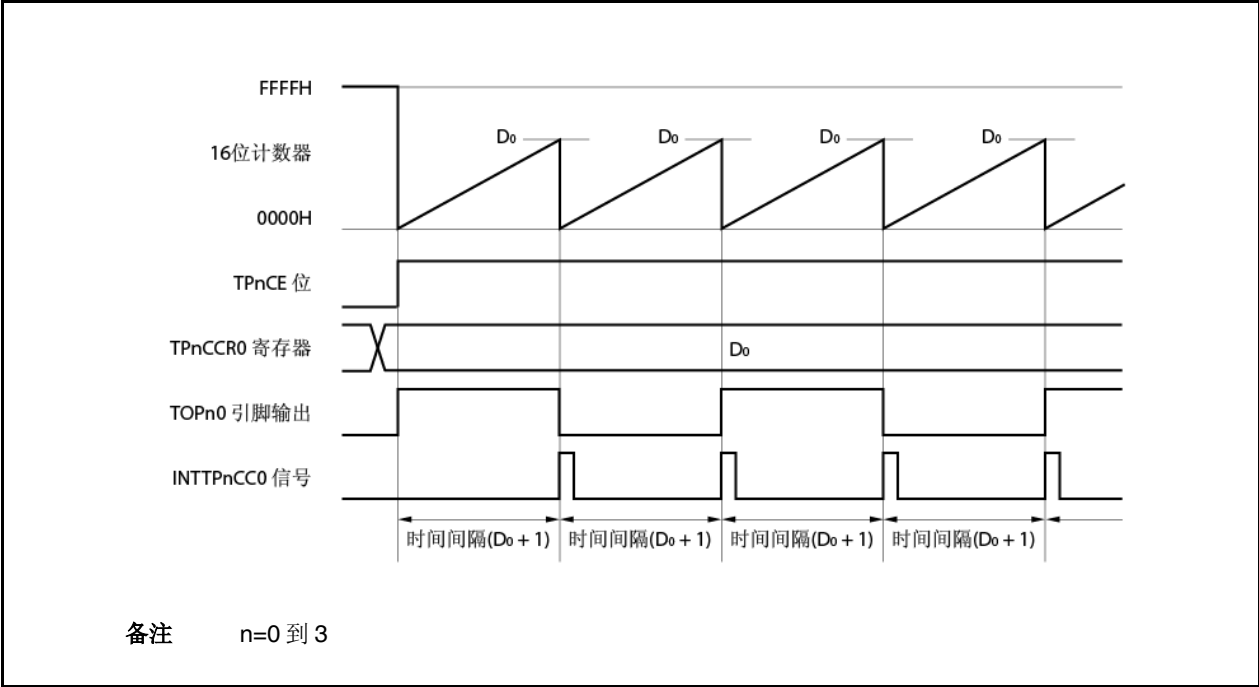


图 6-3 时间间隔定时器模式下操作的基本定时





当 TPnCE 位设为 1 时，16 位计数器在与计数时钟同步时其值从 FFFFH 清零为 0000H，且计数器开始计数。此时，TOPn0 引脚的输出反相。此外，TPnCCR0 寄存器的设置值会传送到 CCR0 缓冲寄存器。

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配时，16 位计数器清零为 0000H，TOPn0 引脚的输出反相，并生成比较匹配中断请求信号（INTTPnCC0）。

时间间隔可用下式计算。

$$\text{时间间隔} = (\text{TPnCCR0 寄存器的设置值} + 1) \times \text{计数时钟周期}$$

备注 n=0 到 3

图 6-4 时间间隔定时器模式下操作的寄存器设置（1/2）

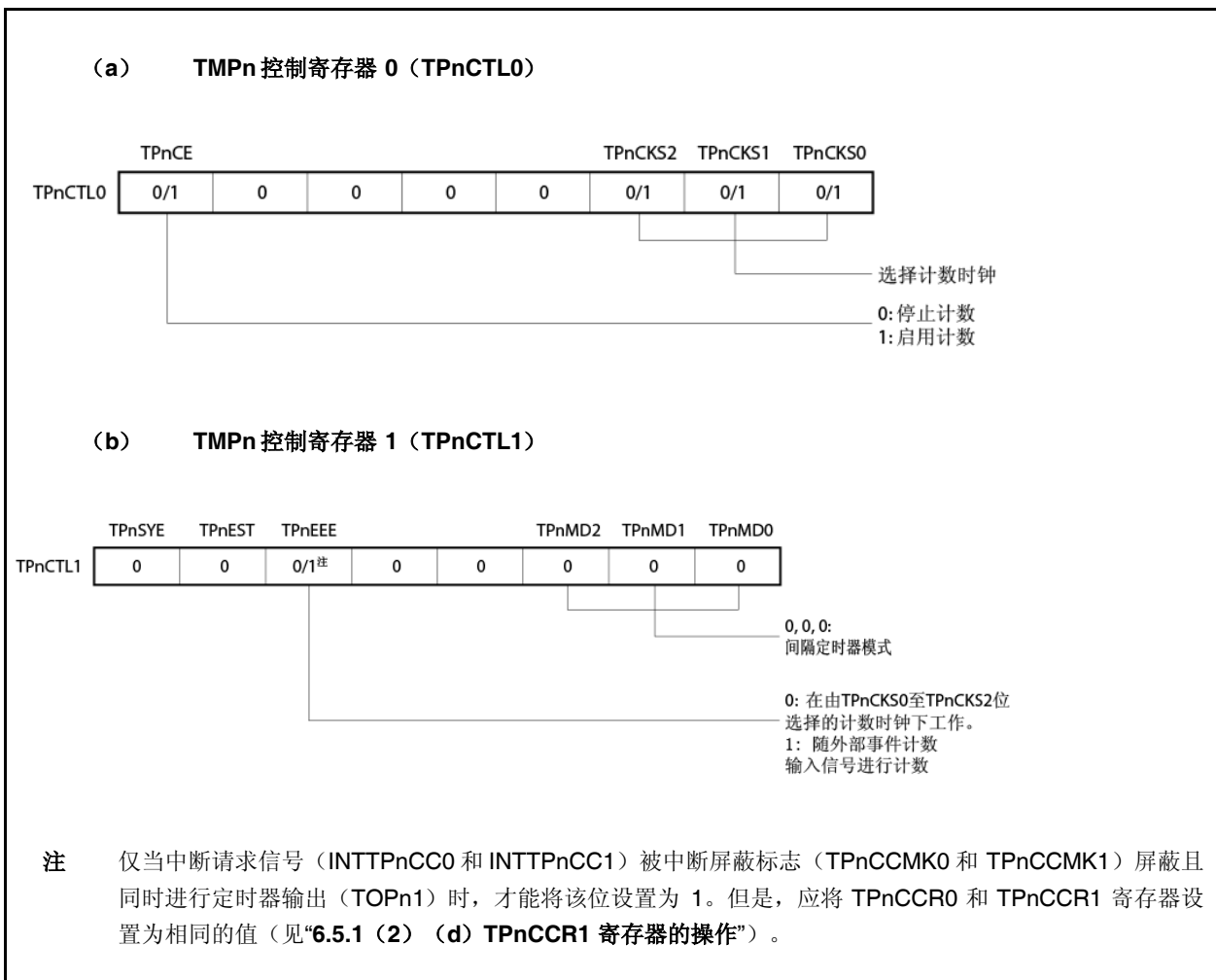
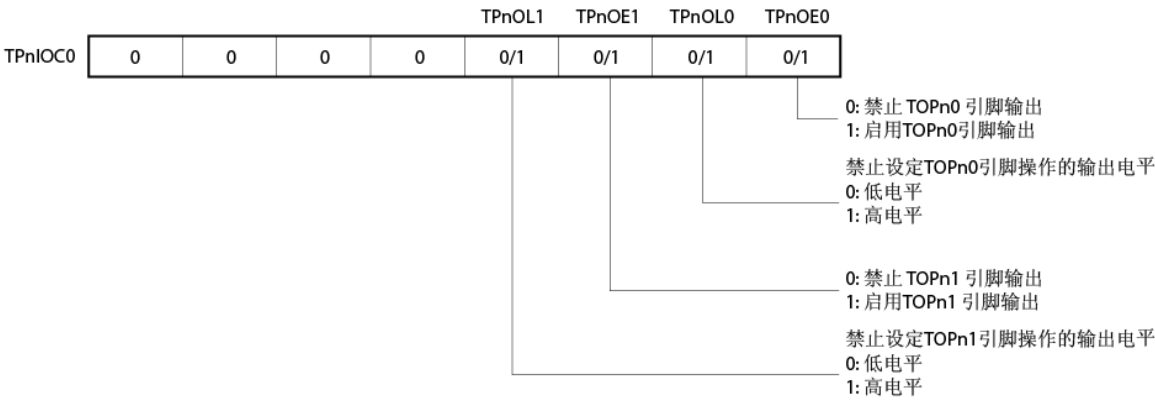


图 6-4 时间间隔定时器模式下操作的寄存器设置 (2/2)

(c) TMPn I/O 控制寄存器 0 (TPnIOC0)



(d) TMPn 计数器读取缓冲寄存器 (TPnCNT)

通过读取 TPnCNT 寄存器可以读取 16 位计数器的计数值。

(e) TMPn 捕捉/比较寄存器 0 (TPnCCR0)

若 TPnCCR0 寄存器设置为  $D_0$ ，则时间间隔如下。

时间间隔 =  $(D_0 + 1) \times$  计数时钟周期

(f) TMPn 捕捉/比较寄存器 1 (TPnCCR1)

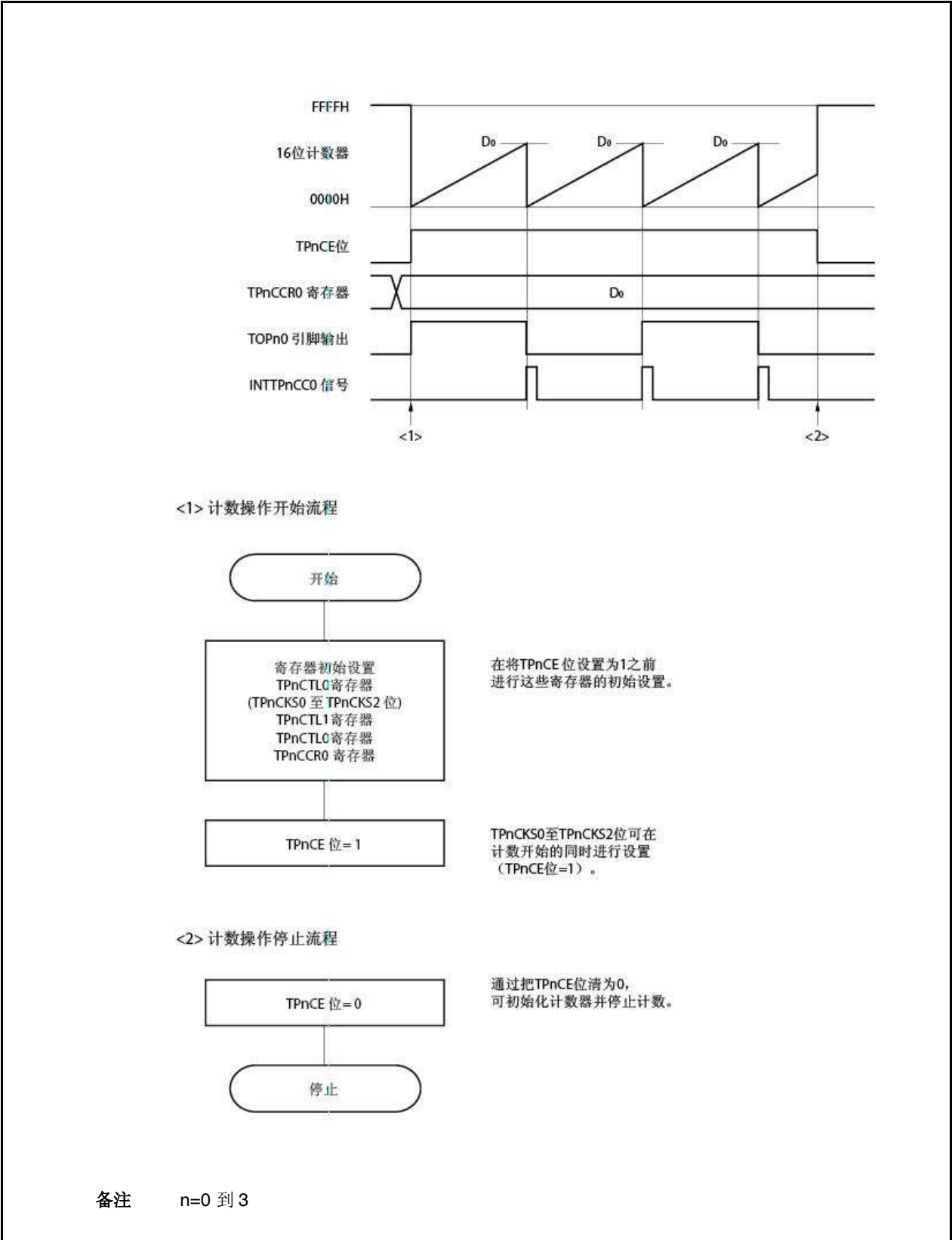
通常，在时间间隔定时器模式下不使用 TPnCCR1 寄存器。但是，TPnCCR1 寄存器的设置值会传送到 CCR1 缓冲寄存器。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 (INTTPnCC1)。

因此，应利用相应的中断屏蔽标志 (TPnCCMK1) 屏蔽该中断请求。

- 备注
1. TMPn I/O 控制寄存器 1 (TPnIOC1)、TMPn I/O 控制寄存器 2 (TPnIOC2)、TMPn 选项寄存器 0 (TPnOPT0) 在时间间隔定时器模式下不使用。
  2. n=0 到 3

(1) 时间间隔定时器模式工作流程

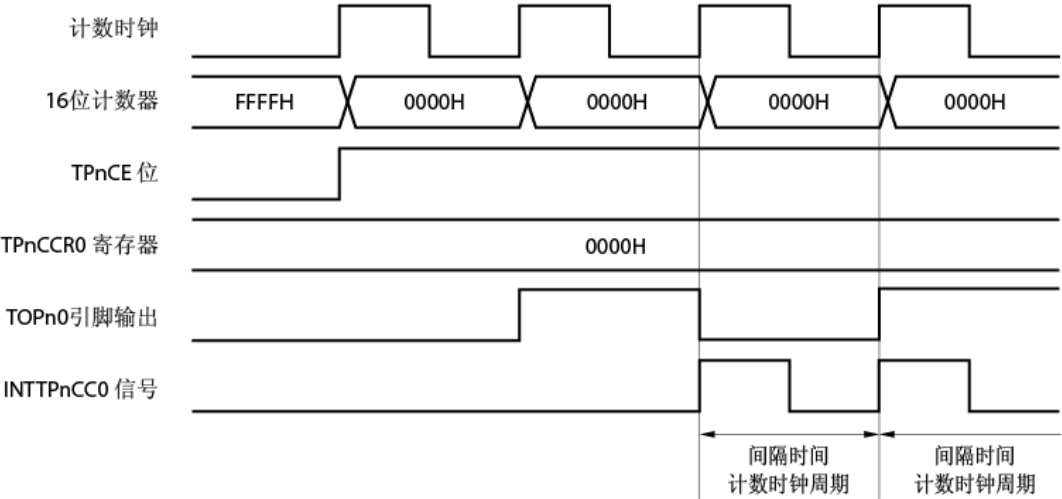
图 6-5 时间间隔定时器模式下的软件处理流程



(2) 时间间隔定时器模式的操作定时

(a) 在 TPnCCR0 寄存器设为 0000H 时的操作

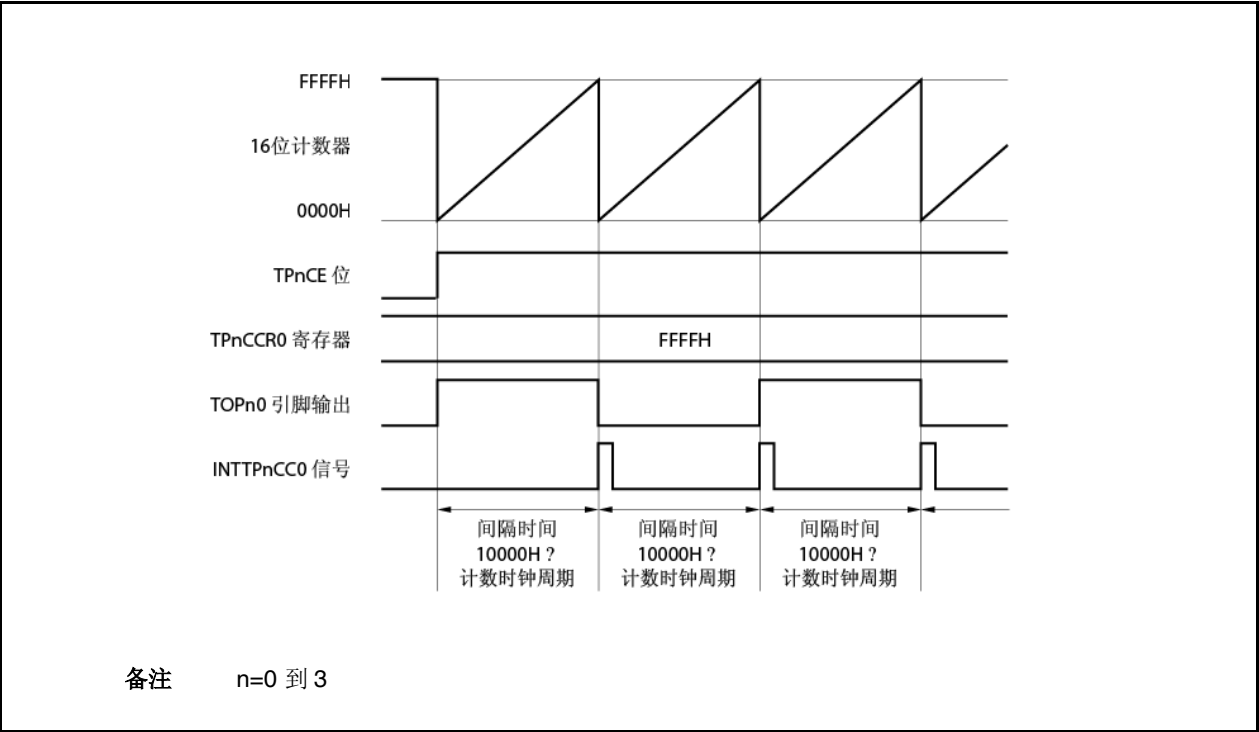
若 TPnCCR0 寄存器设置为 0000H，则在第一个计数时钟后每个计数时钟都会生成 INTTPnCC0 信号，且 TOPn0 引脚的输出反相。  
16 位计数器的值总是 0000H。



备注 n=0 到 3

(b) 在 TPnCCR0 寄存器设为 FFFFH 时的操作

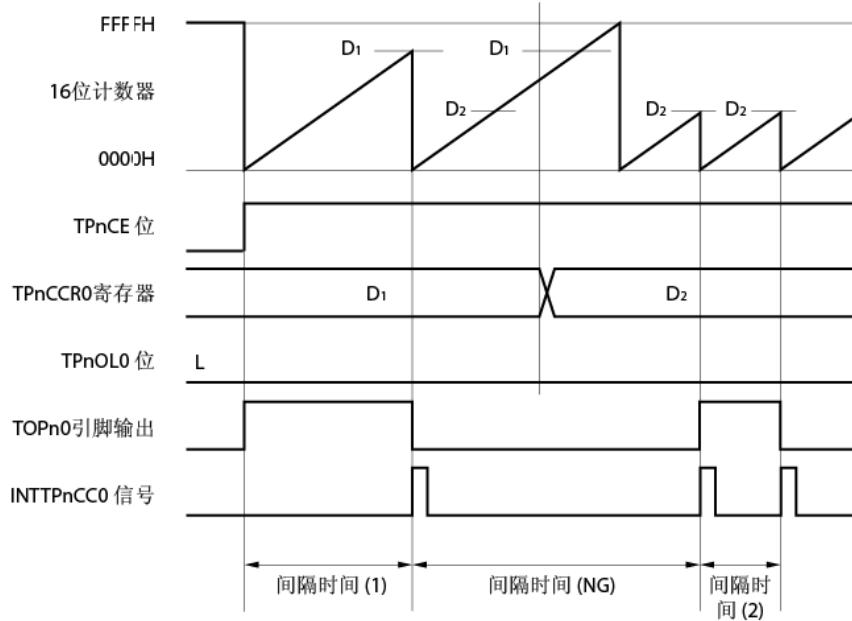
若 TPnCCR0 寄存器设为 FFFFH，则 16 位计数器计数到 FFFFH 为止。在下一个向上计数定时的同步下，计数器清零为 0000H。生成 INTTPnCC0 信号，且 TOPn0 引脚输出反相。此时，不会产生溢出中断请求信号（INTTPnOV），也不会将溢出标志（TPnOPT0.TPnOVF 位）设置为 1。



## (c) 关于重写 TPnCCR0 寄存器的注意点

要把 TPnCCR0 寄存器的值变为一个较小的值，应停止计数一次，再改变设定值。

若 TPnCCR0 寄存器的值在计数过程中重写为较小的值，则 16 位计数器可能会溢出。



- 备注**
1. 间隔时间 (1) :  $(D_1+1) \times \text{计数时钟周期}$   
 间隔时间 (NG) :  $(10000H+D_2+1) \times \text{计数时钟周期}$   
 间隔时间 (2) :  $(D_2+1) \times \text{计数时钟周期}$
  2. n=0 到 3

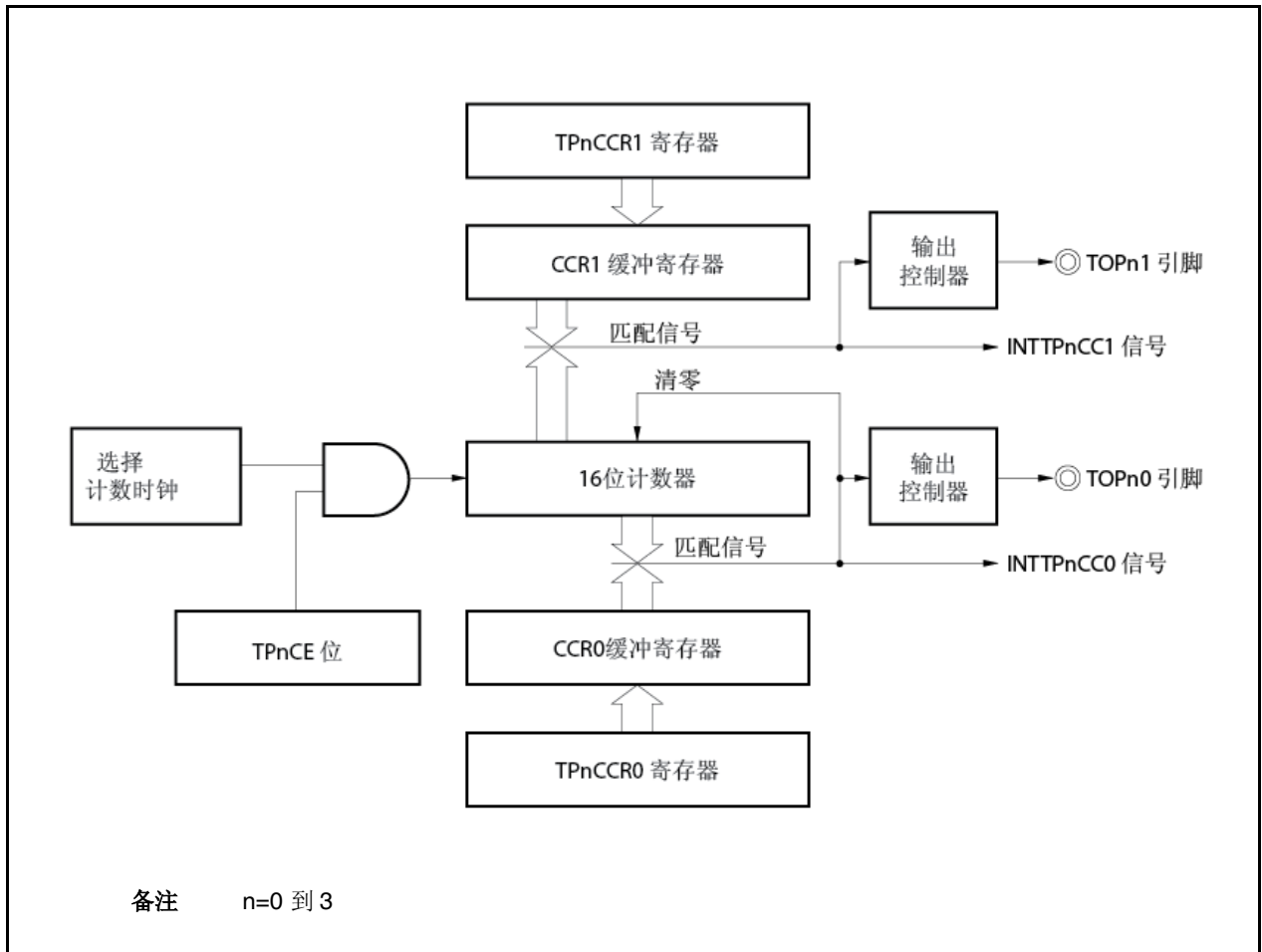
当计数值大于  $D_2$  而小于  $D_1$  时，若 TPnCCR0 寄存器的值从  $D_1$  变为  $D_2$ ，则在重写 TPnCCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。结果，用来比较的 16 位计数器的值为  $D_2$ 。

但是，因为计数值已经超过了  $D_2$ ，所以 16 位计数器会向上计数到 FFFFH、溢出，然后从 0000H 开始重新向上计数。当计数值和  $D_2$  相符时，会生成 INTTPnCC0 信号且 TOPn0 引脚输出反相。

因此，INTTPnCC0 信号可能不会以原先预计的间隔时间“ $(D_1+1) \times \text{计数时钟周期}$ ”或“ $(D_2+1) \times \text{计数时钟周期}$ ”来产生，而会以“ $(10000H+D_2+1) \times \text{计数时钟周期}$ ”的时间间隔生成。

## (d) TPnCCR1 寄存器的操作

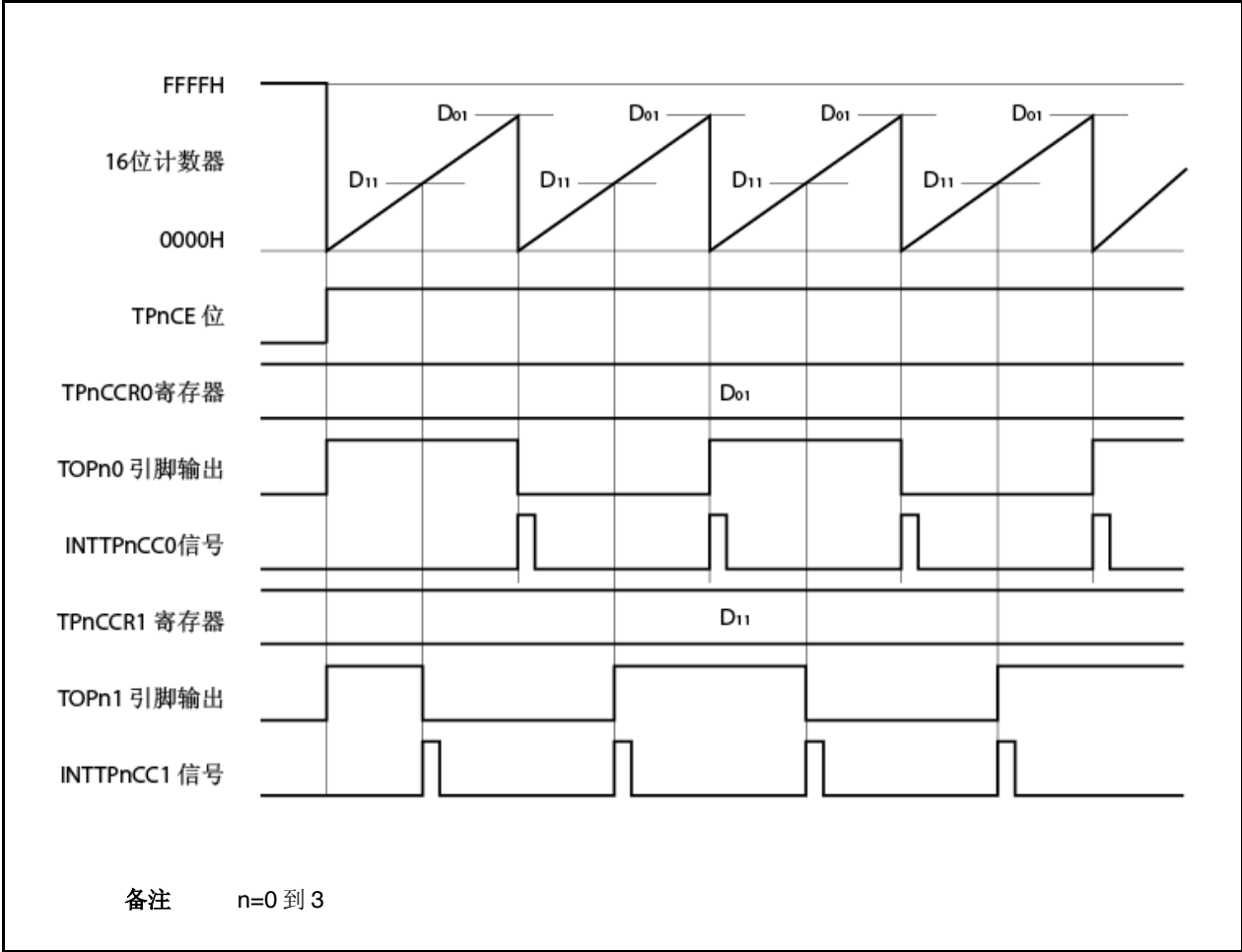
图 6-6 TPnCCR1 寄存器的配置



若 TPnCCR1 寄存器的设定值小于 TPnCCR0 寄存器的设定值，则 INTTPnCC1 信号每周期生成一次。同时，TOPn1 引脚的输出反相。

Topn1 引脚输出一个方波，其周期与 TOPn0 引脚的输出相同。

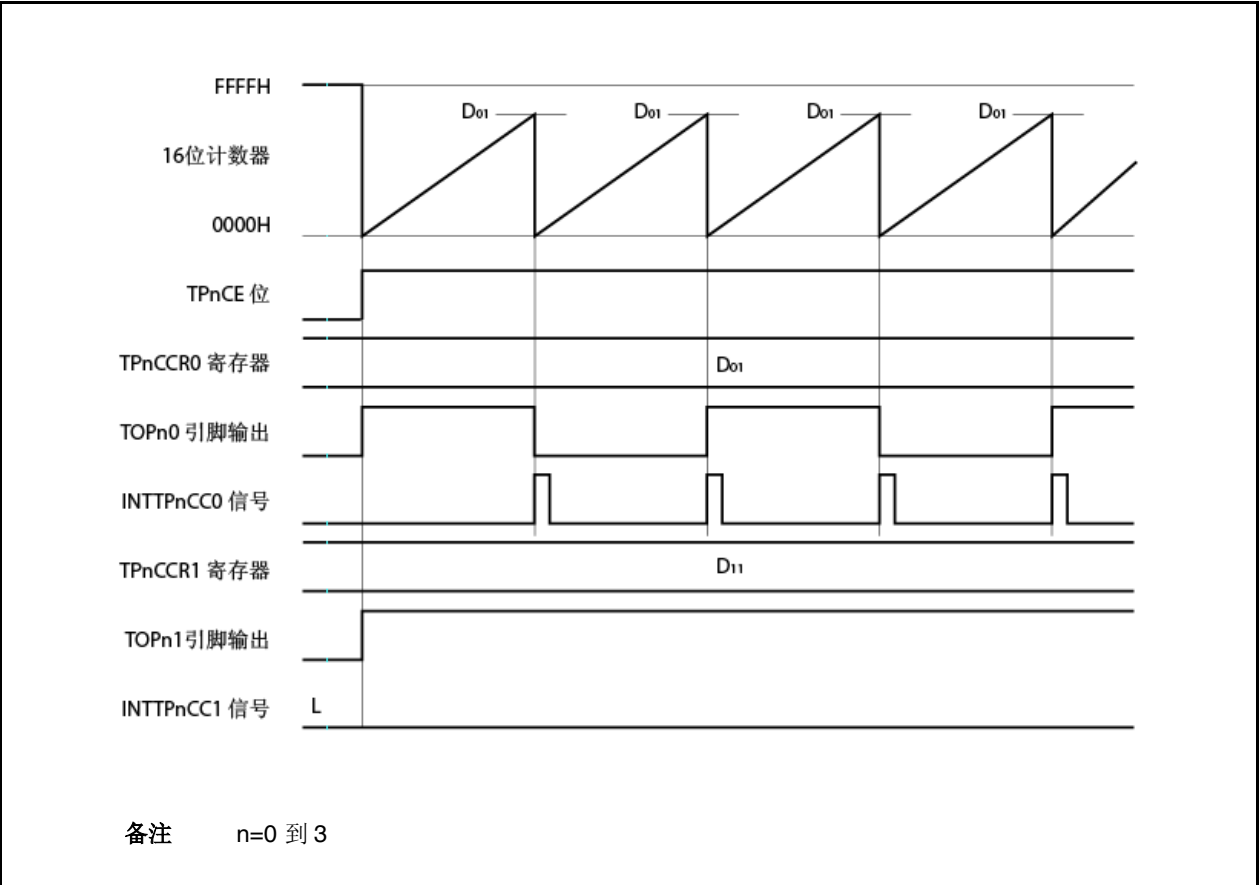
图 6-7 当  $D_{01} \geq D_{11}$  时的时序图





若 TPnCCR1 寄存器的设定值大于 TPnCCR0 寄存器的设定值，则 16 位计数器的计数值不匹配 TPnCCR1 寄存器的值。结果，不会生成 INTTPnCC1 信号，也不会改变 TOPn1 引脚的输出。

图 6-8 当 D01<D11 时的时序图



6.5.2 外部事件计数模式 (TPnMD2 到 TPnMD0 位=001)

在外部事件计数模式下，当 TPnCTL0.TPnCE 位设置为 1 时会对外部事件计数输入的有效边缘进行计数，且每次计数到指定个数的边缘时都会生成中断请求信号 (INTTPnCC0)。不能使用 TOPn0 引脚。

通常，在外部事件计数模式下不使用 TPnCCR1 寄存器。

图 6-9 外部事件计数模式下的配置

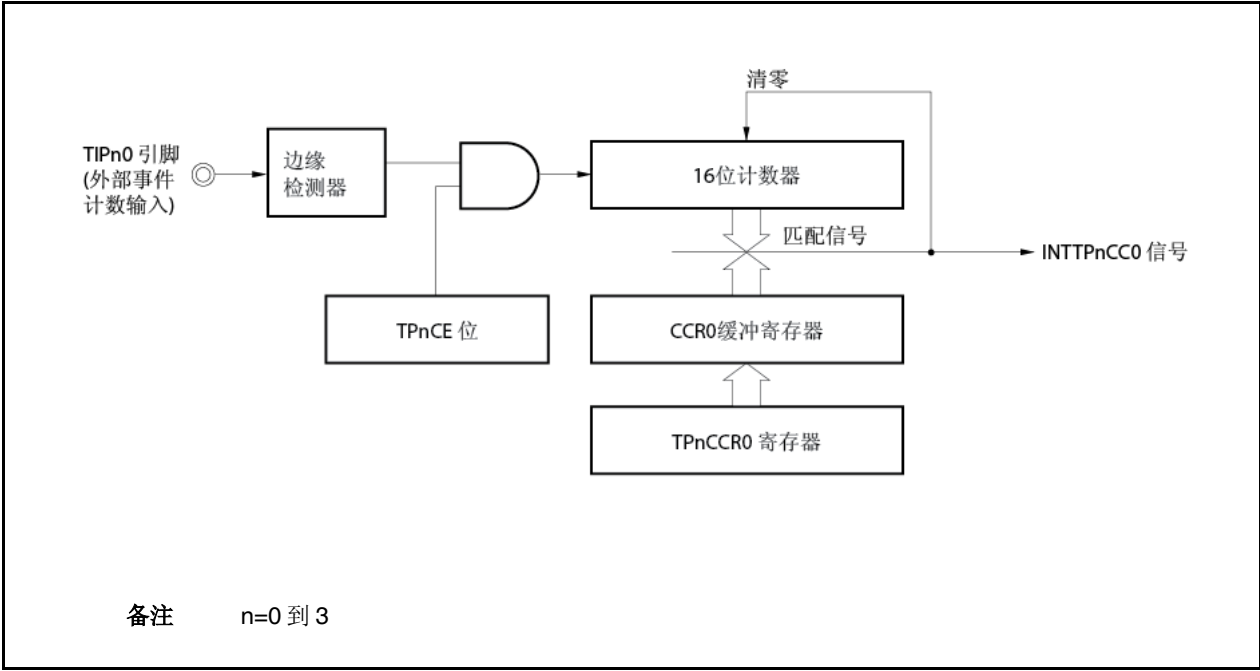
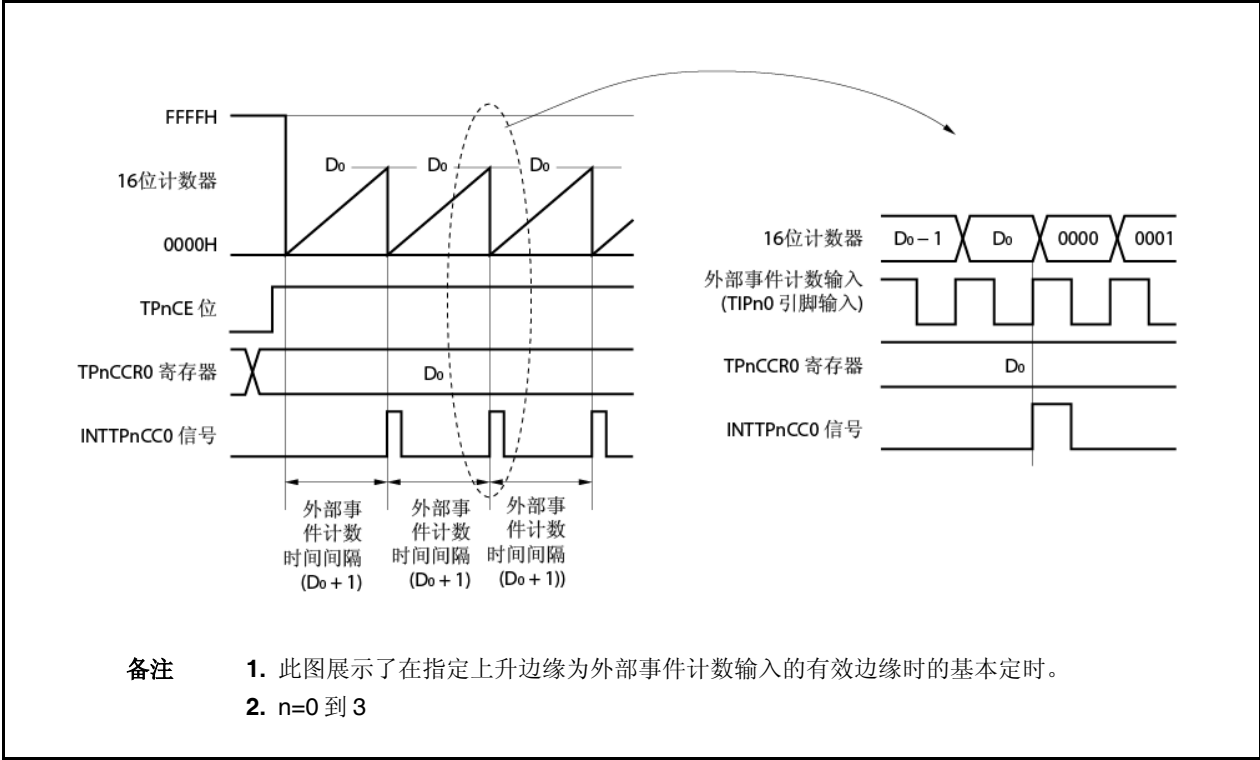


图 6-10 外部事件计数模式下的基本定时



当 TPnCE 位设为 1 时，16 位计数器的值从 FFFFH 清零为 0000H 计数器在每次检测到外部事件计数输入的有效边缘时进行计数。此外，TPnCCR0 寄存器的设置值会传送到 CCR0 缓冲寄存器。

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配时，16 位计数器清零为 0000H 并生成比较匹配中断请求信号 (INTTPnCC0)。

每检测到外部事件计数输入的有效边缘 (TPnCCR0 寄存器的设置值+1) 次时都会生成 INTTPnCC0 信号。

图 6-11 外部事件计数模式下操作的寄存器设置 (1/2)

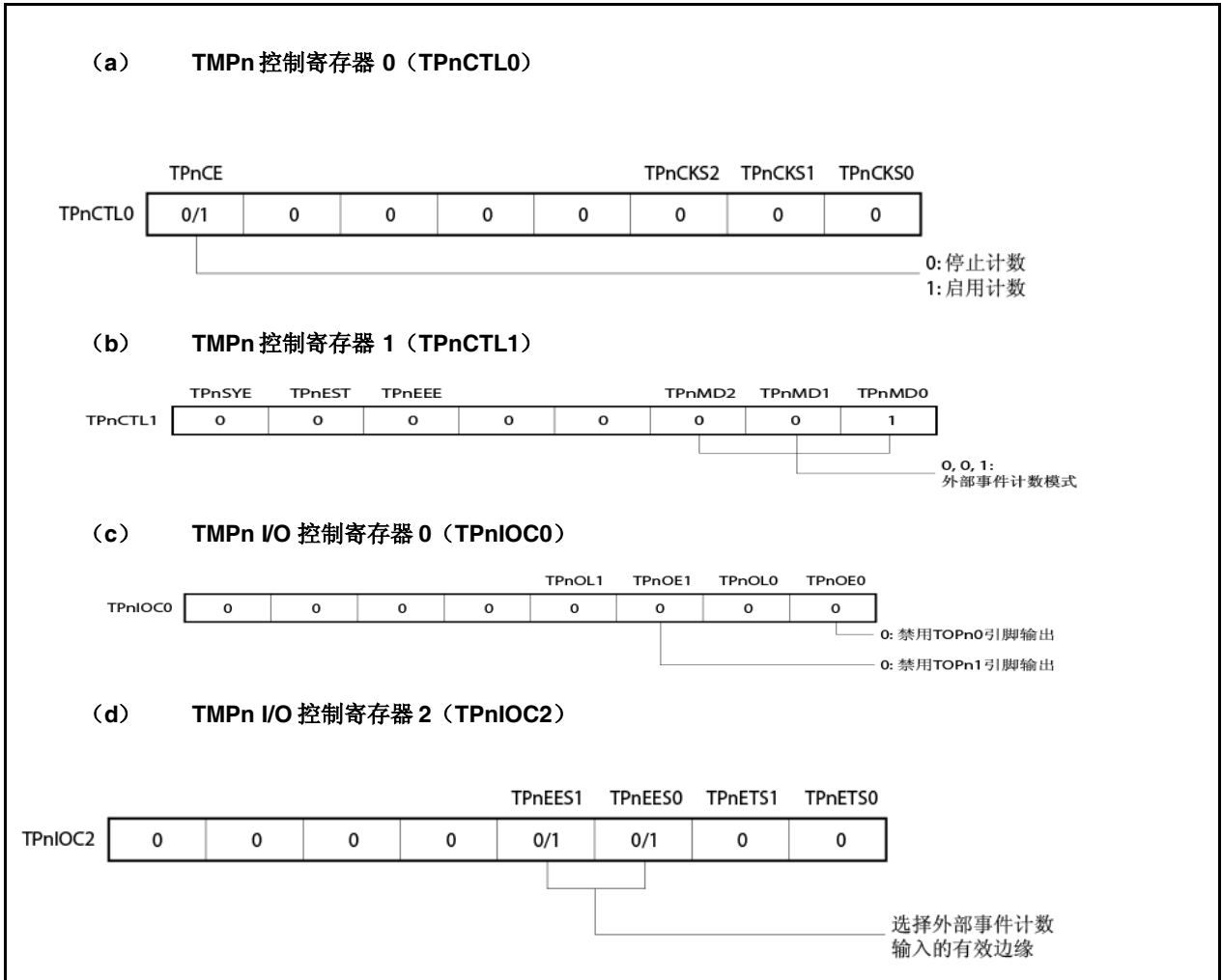


图 6-11 外部事件计数模式下操作的寄存器设置 (2/2)

**(e) TMPn 计数器读取缓冲寄存器 (TPnCNT)**

通过读 TPnCNT 寄存器可以读取 16 位计数器的计数值。

**(f) TMPn 捕捉/比较寄存器 0 (TPnCCR0)**

若将 D<sub>0</sub> 设置给 TPnCCR0 寄存器，当外部事件计数的个数达到 (D<sub>0</sub>+1) 时，计数器清零且生成比较匹配中断请求信号 (INTTPnCC0)。

**(g) TMPn 捕捉/比较寄存器 1 (TPnCCR1)**

通常，在外部事件计数模式下不使用 TPnCCR1 寄存器。但是，TPnCCR1 寄存器的设置值还是会传送到 CCR1 缓冲寄存器。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 (INTTPnCC1)。

因此，应利用中断屏蔽标志 (TPnCCMK1) 屏蔽该中断信号。

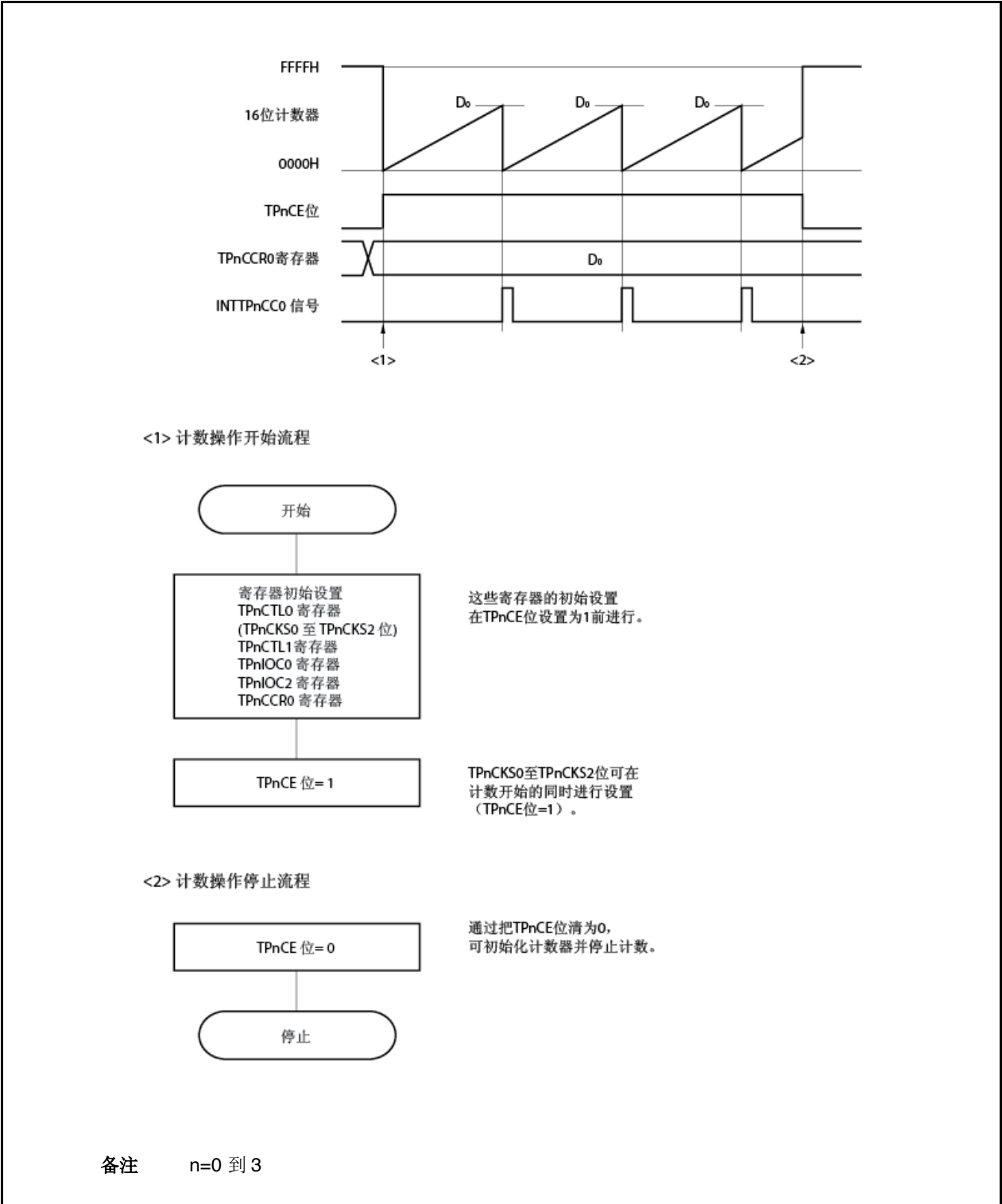
**注意事项** 当使用外部时钟作为计数时钟时，外部时钟只能从 TIPn0 引脚输入。此时，应将 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位设置为 00 (捕捉触发输入 (TIPn0 引脚)：无边缘检测)。

**备注**

1. TMPn I/O 控制寄存器 1 (TPnIOC1) 和 TMPn 选项寄存器 0 (TPnOPT0) 在外部事件计数模式下不使用。
2. n=0 到 3

(1) 外部事件计数模式工作流程

图 6-12 外部事件计数模式下的软件处理流程

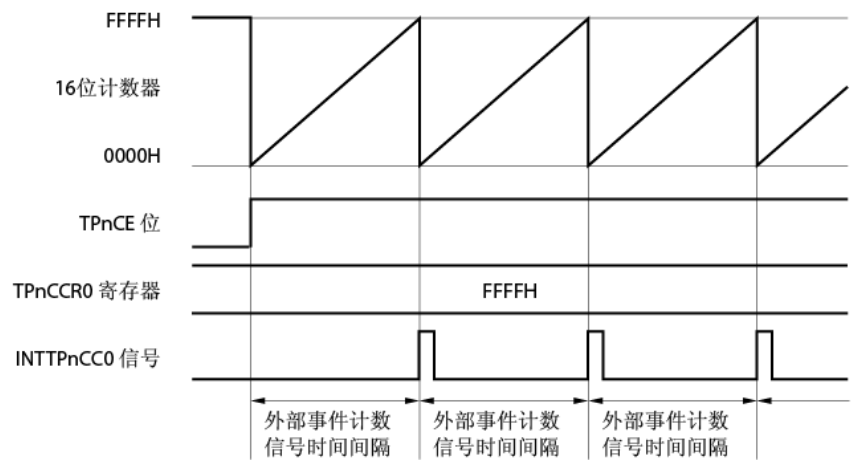


(2) 外部事件计数模式下的操作定时

- 注意事项
- 1. 在外部事件计数模式下，不要将 TPnCCR0 寄存器设为 0000H。
  - 2. 在外部事件计数模式下，无法使用定时器输出。如果利用外部事件计数输入进行定时器输出，应设置为时间间隔定时器模式，为计数时钟选择由外部事件计数输入启用的操作（TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0 位=000、TPnCTL1.TPnEEE 位=1）。

(a) 在 TPnCCR0 寄存器设为 FFFFH 时的操作

若 TPnCCR0 寄存器设为 FFFFH，则每次检测到外部事件计数信号的有效边缘时 16 位计数器都会计数到 FFFFH。在下一个向上计数定时同步下 16 位计数器会清零为 0000H，并生成 INTTPnCC0 信号。此时，TPnOPT0.TPnOVF 位未设置。

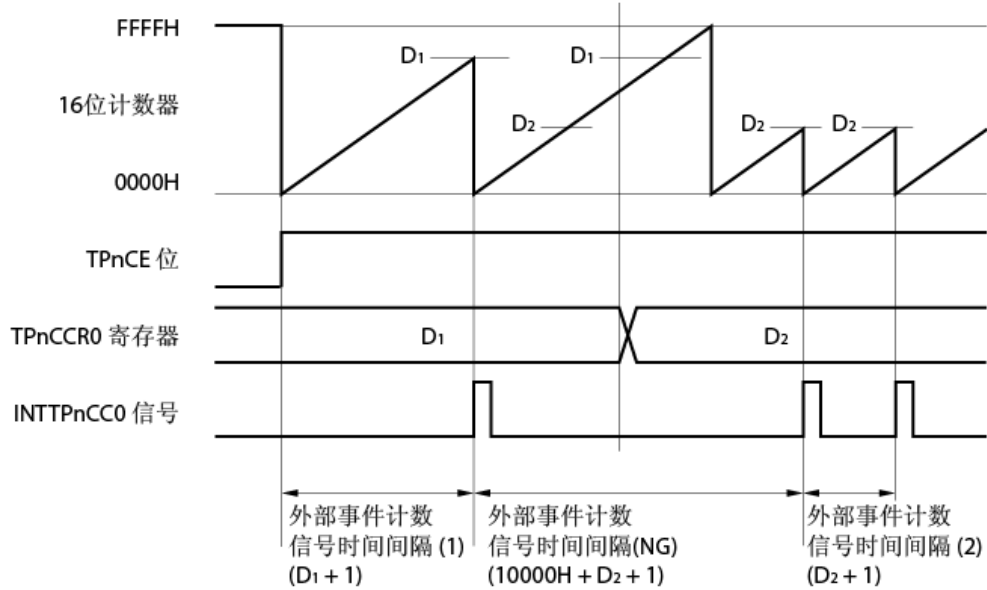


备注      n=0 到 3

## (b) 重写 TPnCCR0 寄存器的注意点

要把 TPnCCR0 寄存器的值变为一个较小的值，应停止计数一次，再改变设定值。

若 TPnCCR0 寄存器的值在计数过程中重写为较小的值，则 16 位计数器可能会溢出。



备注 n=0 到 3

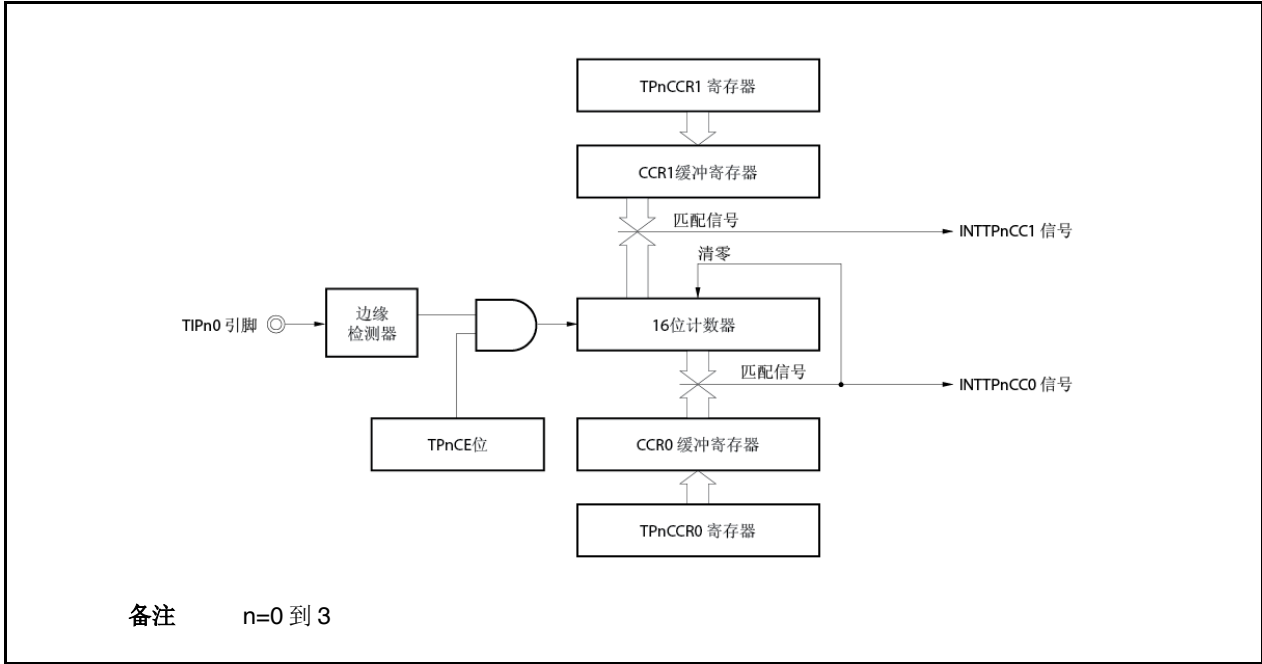
当计数值大于  $D_2$  而小于  $D_1$  时，若 TPnCCR0 寄存器的值从  $D_1$  变为  $D_2$ ，则在重写 TPnCCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。结果，用来与 16 位计数器进行比较的值为  $D_2$ 。

但是，因为计数值已经超过了  $D_2$ ，所以 16 位计数器会向上计数到 FFFFH、溢出，然后从 0000H 开始重新向上计数。当计数值匹配  $D_2$  时，会生成 INTTPnCC0 信号。

因此，INTTPnCC0 信号可能不会以原先预计的有效边缘计数“ $(D_1+1)$  次”或“ $(D_2+1)$  次”来产生，而会在“ $(10000H+D_2+1)$  次”有效边缘计数时生成。

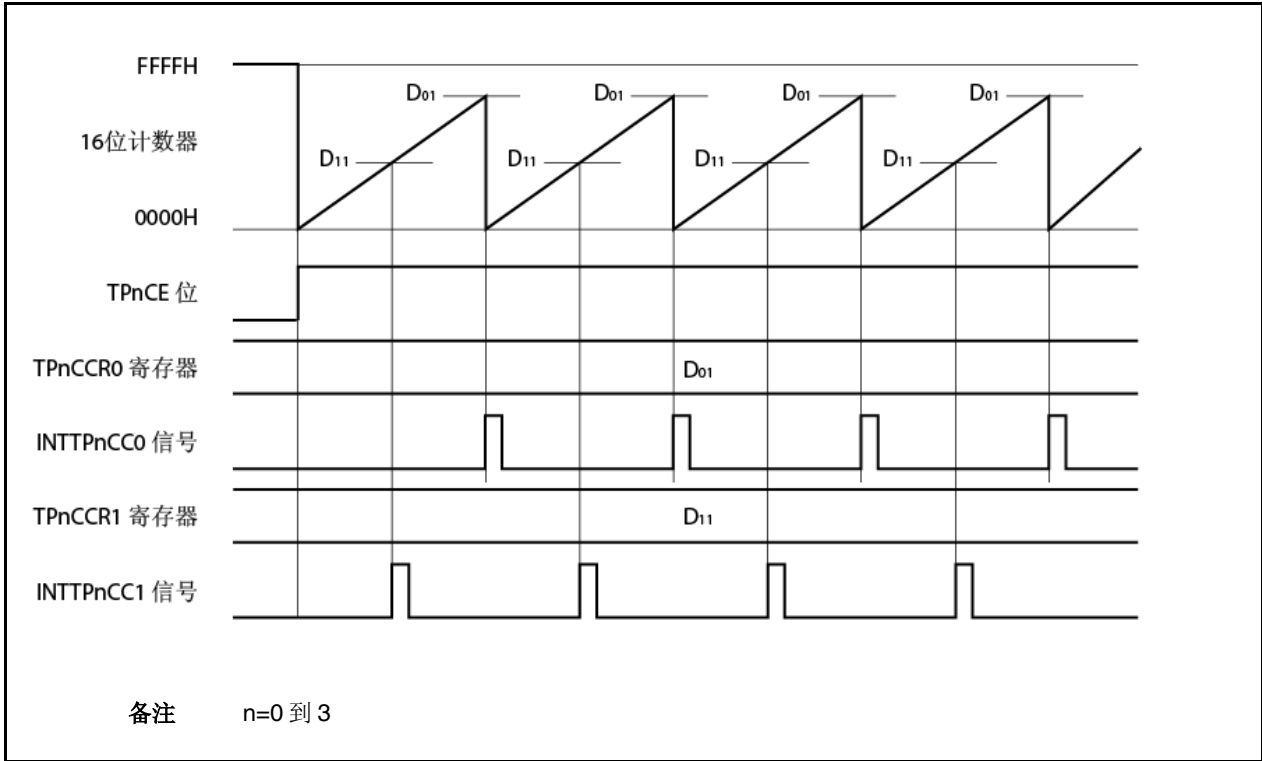
(c) Tpnccr1 寄存器的操作

图 6-13 TPnCCR1 寄存器的配置



若 TPnCCR1 寄存器的设定值小于 TPnCCR0 寄存器的设定值，则 INTTPnCC1 信号每周期生成一次。

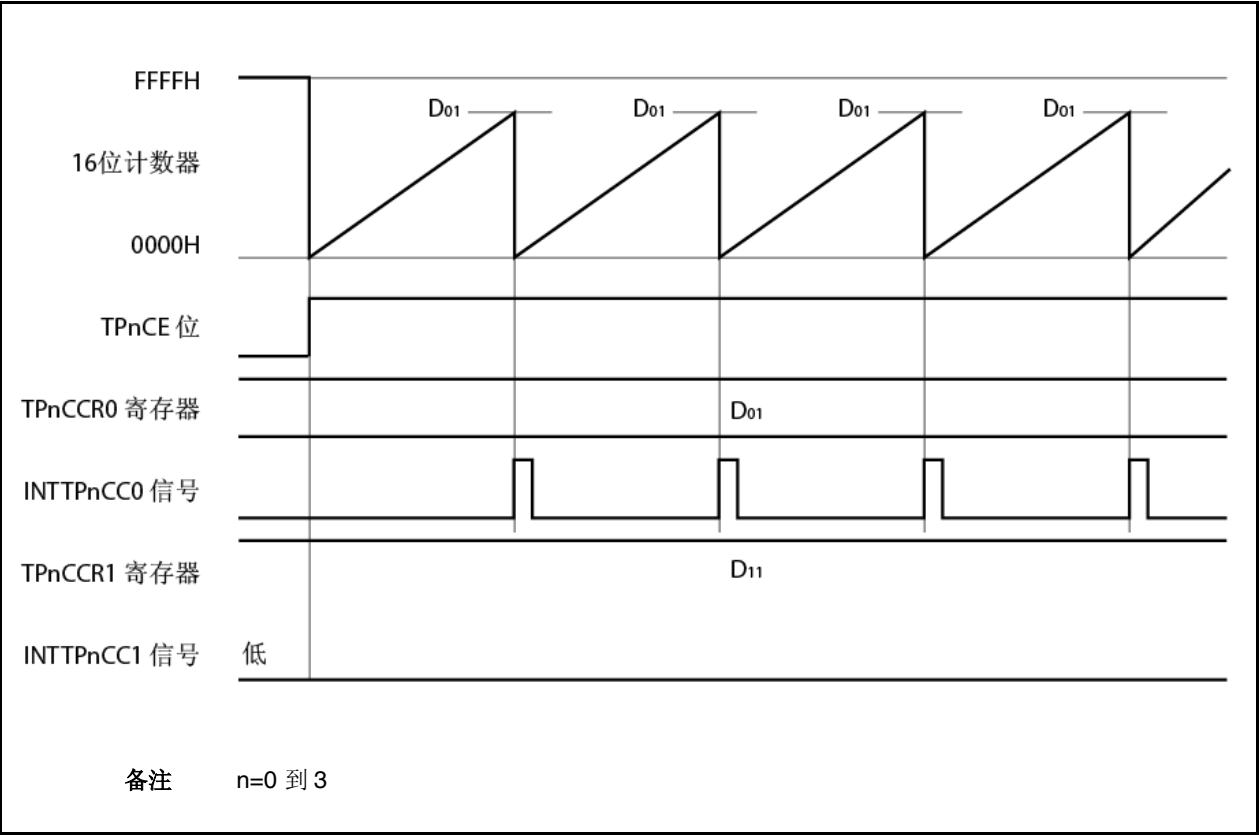
图 6-14 当 D01≥D11 时的时序图





若 TPnCCR1 寄存器的设定值大于 TPnCCR0 寄存器的设定值，则因为 16 位计数器的计数值与 TPnCCR1 寄存器的值不匹配，故不会产生 INTTPnCC1 信号。

图 6-15 当 D01<D11 时的时序图



### 6.5.3 外部触发脉冲输出模式 (TPnMD2 到 TPnMD0 位=010)

在外部触发脉冲输出模式下，当 **TPnCTL0.TPnCE** 位设为 1 时，16 位定时器/事件计数器 **P** 等待触发。当检测到外部触发输入信号的有效边缘时，16 位定时器/事件计数器 **P** 开始计数，且从 **TOPn1** 引脚输出一个 PWM 波形。

脉冲也可以通过生成软件触发来输出，而不是利用外部触发。在使用软件触发时，以 **PWM** 波形的一个周期作为其半周期的方波也可从 **TOPn0** 引脚输出。

图 6-16 外部触发脉冲输出模式下的配置

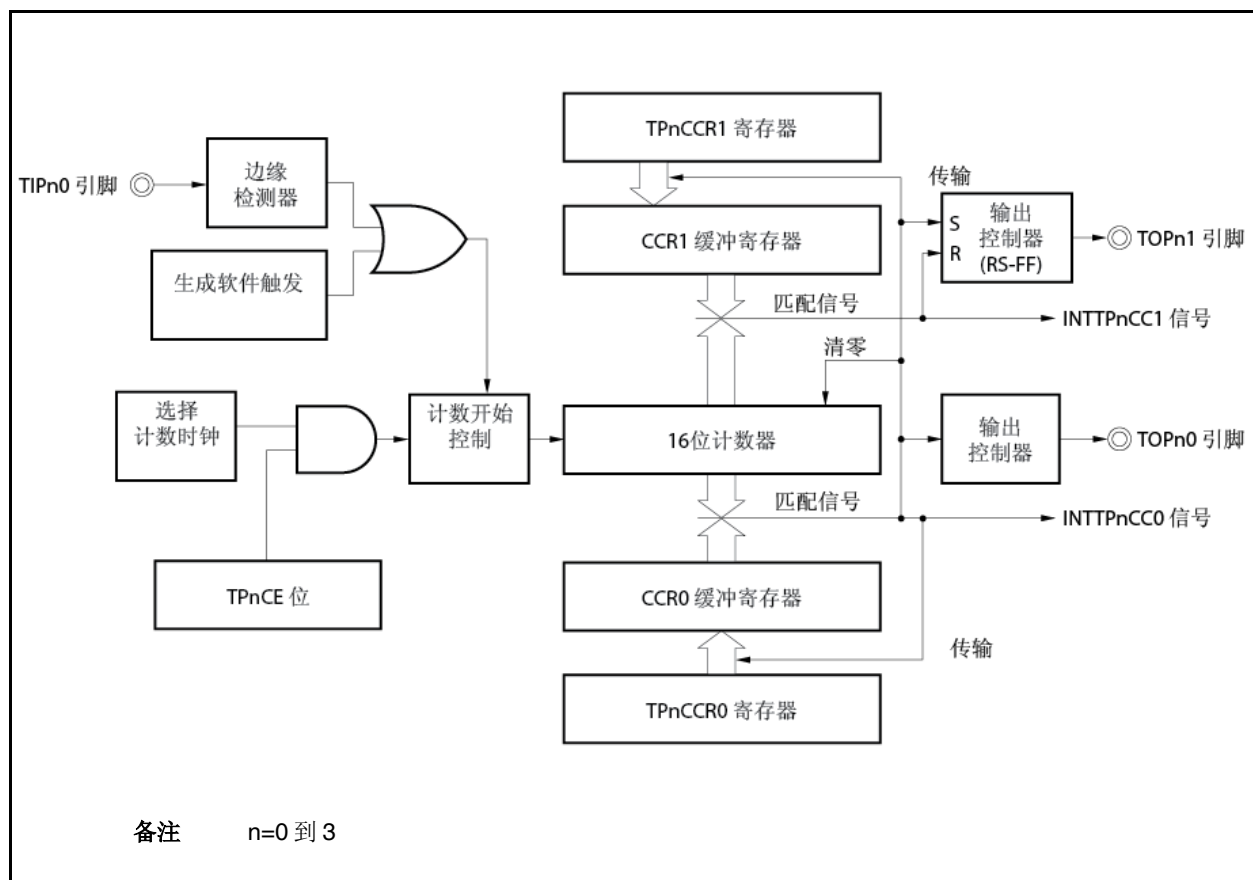
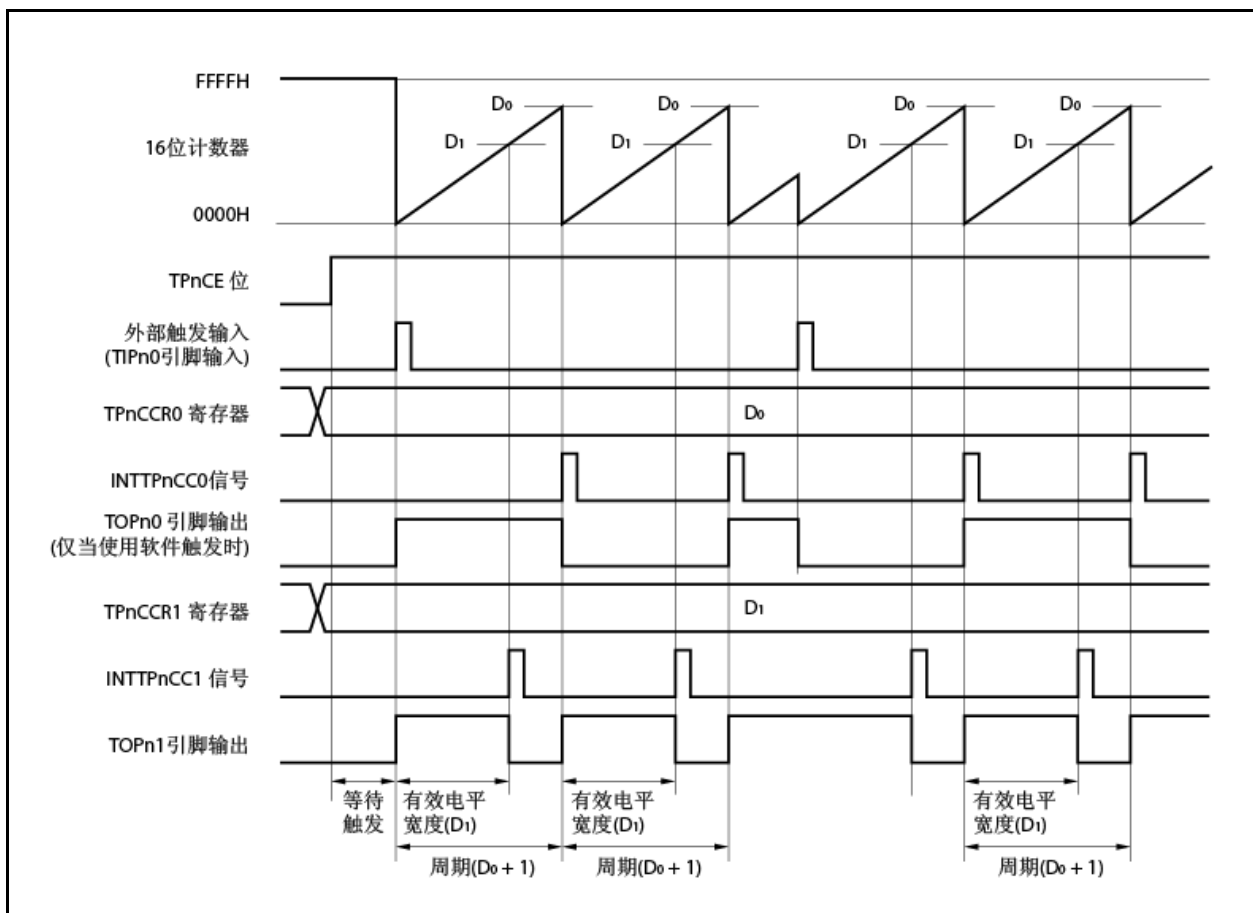


图 6-17 外部触发脉冲输出模式下的基本定时



当 TPnCE 位设为 1 时，16 位定时器/事件计数器 P 等待触发。当触发产生时，16 位计数器从 FFFFH 清零为 0000H，同时开始计数，并从 TOPn1 引脚输出 PWM 波形。若正在计数时再次产生触发，则计数器清零为 0000H 并重新开始。（TOPn0 引脚的输出反相。当触发产生时，不管状态如何（高/低）TOPn1 引脚都会输出高电平。）

PWM 波形的有效电平宽度、周期、占空比可计算如下。

有效电平宽度= (TPnCCR1 寄存器的设置值) × 计数时钟周期

周期= (TPnCCR0 寄存器的设置值+1) × 计数时钟周期

占空比= (TPnCCR1 寄存器的设置值) / (TPnCCR0 寄存器的设置值+1)

当 16 位计数器在其计数值与 CCR0 缓冲寄存器的值匹配后进行下一次计数时，会生成比较匹配请求信号 INTTPnCC0，且 16 位计数器清零为 0000H。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值相匹配时，就会生成比较匹配中断请求信号 INTTPnCC1。

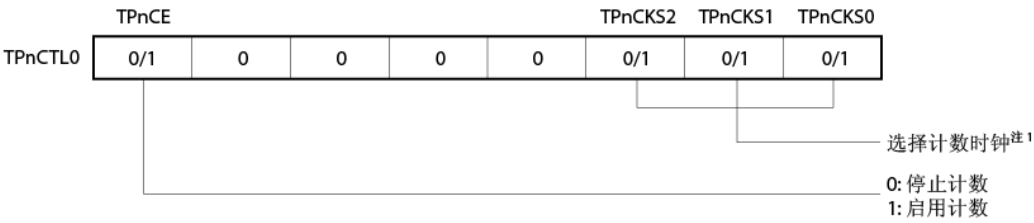
当 16 位计数器的计数值与 CCRm 缓冲寄存器的值相匹配时，TPnCCRm 寄存器设定的值就传送到 CCRm 缓冲寄存器，且 16 位计数器清零为 0000H。

外部触发输入信号的有效边缘或者将软件触发 (TPnCTL1.TPnEST 位) 设置为 1 都可以作为触发。

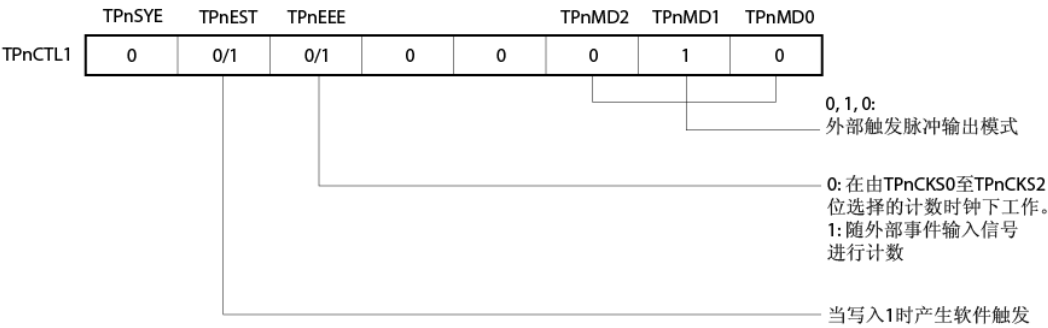
备注 n=0 到 3、m=0、1

图 6-18 外部触发脉冲输出模式下寄存器的设置 (1/2)

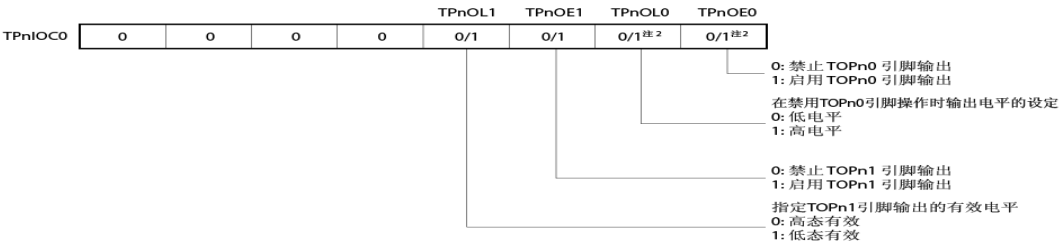
(a) TMPn 控制寄存器 0 (TPnCTL0)



(b) TMPn 控制寄存器 1 (TPnCTL1)



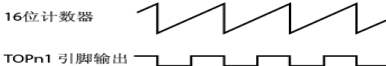
(c) TMPn I/O 控制寄存器 0 (TPnIOC0)



• 当TPnOL1 位= 0时

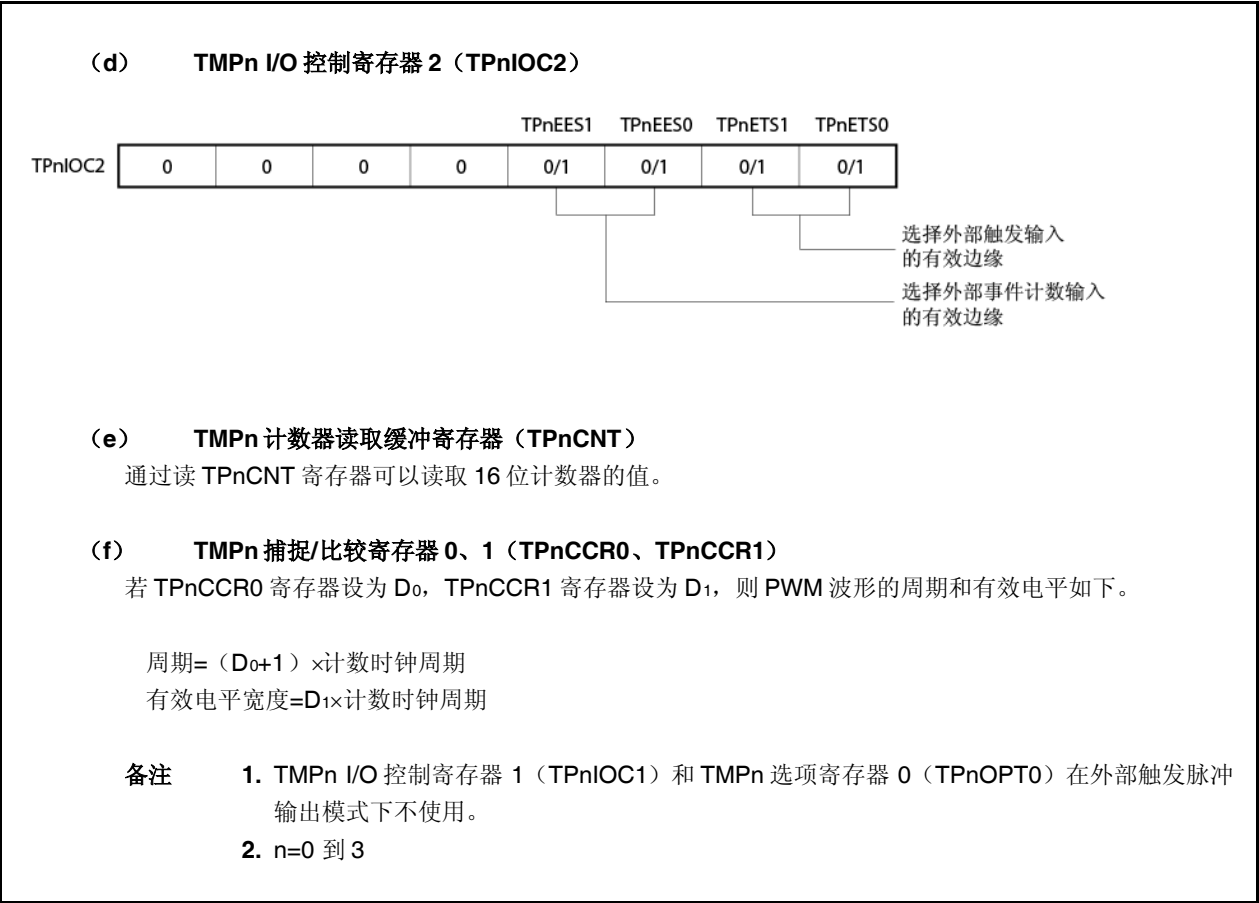


• 当TPnOL1 位= 1时



- 注
1. 当 TPnCTL1.TPnEEE 位=1 时，设定无效。
  2. 在外部触发脉冲输出模式下不使用 TOPn0 引脚时，应将此位清为 0。

图 6-18 外部触发脉冲输出模式下寄存器的设置 (2/2)



(1) 外部触发脉冲输出模式的流程图

图 6-19 外部触发脉冲输出模式下的软件处理流程 (1/2)

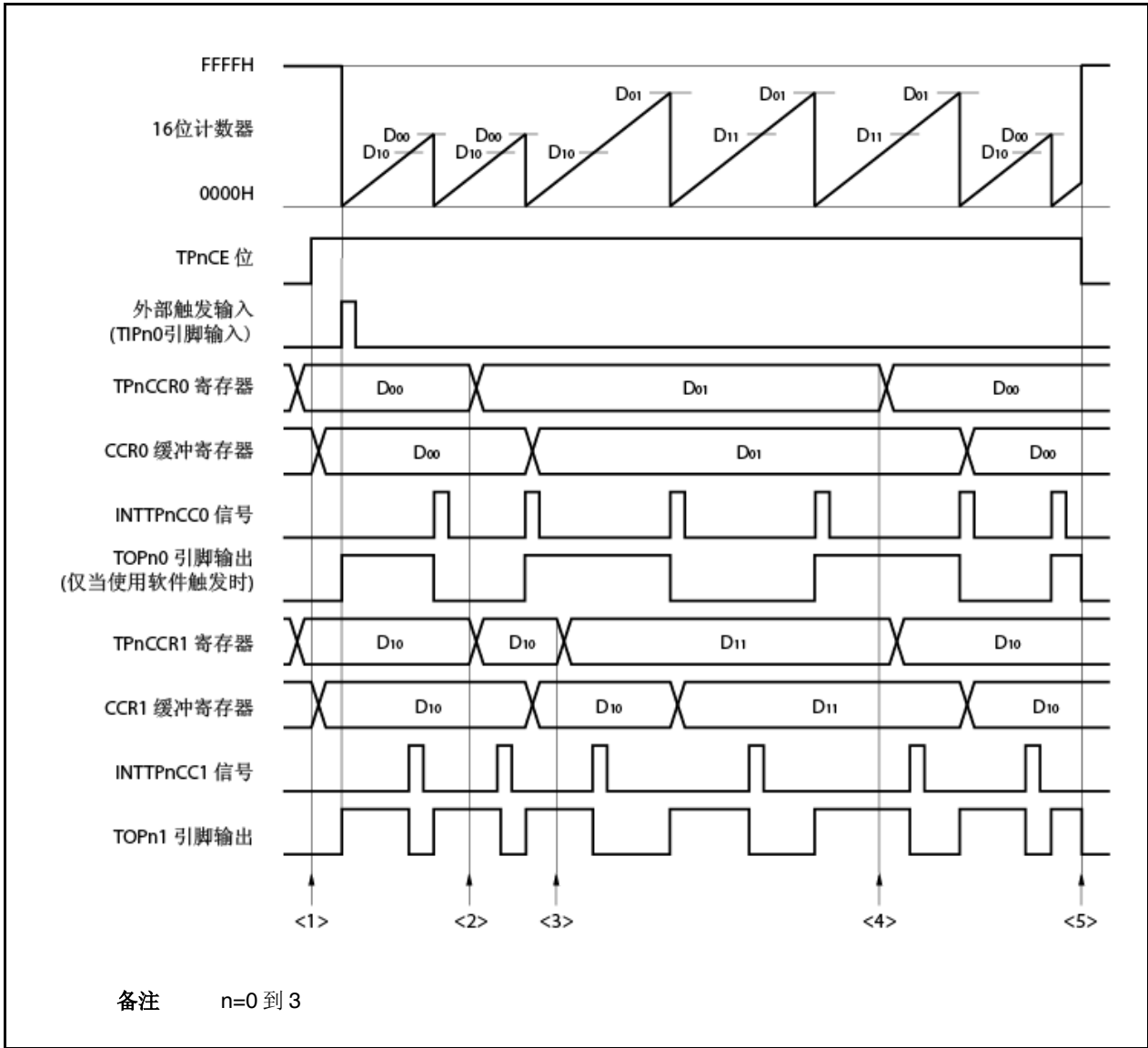
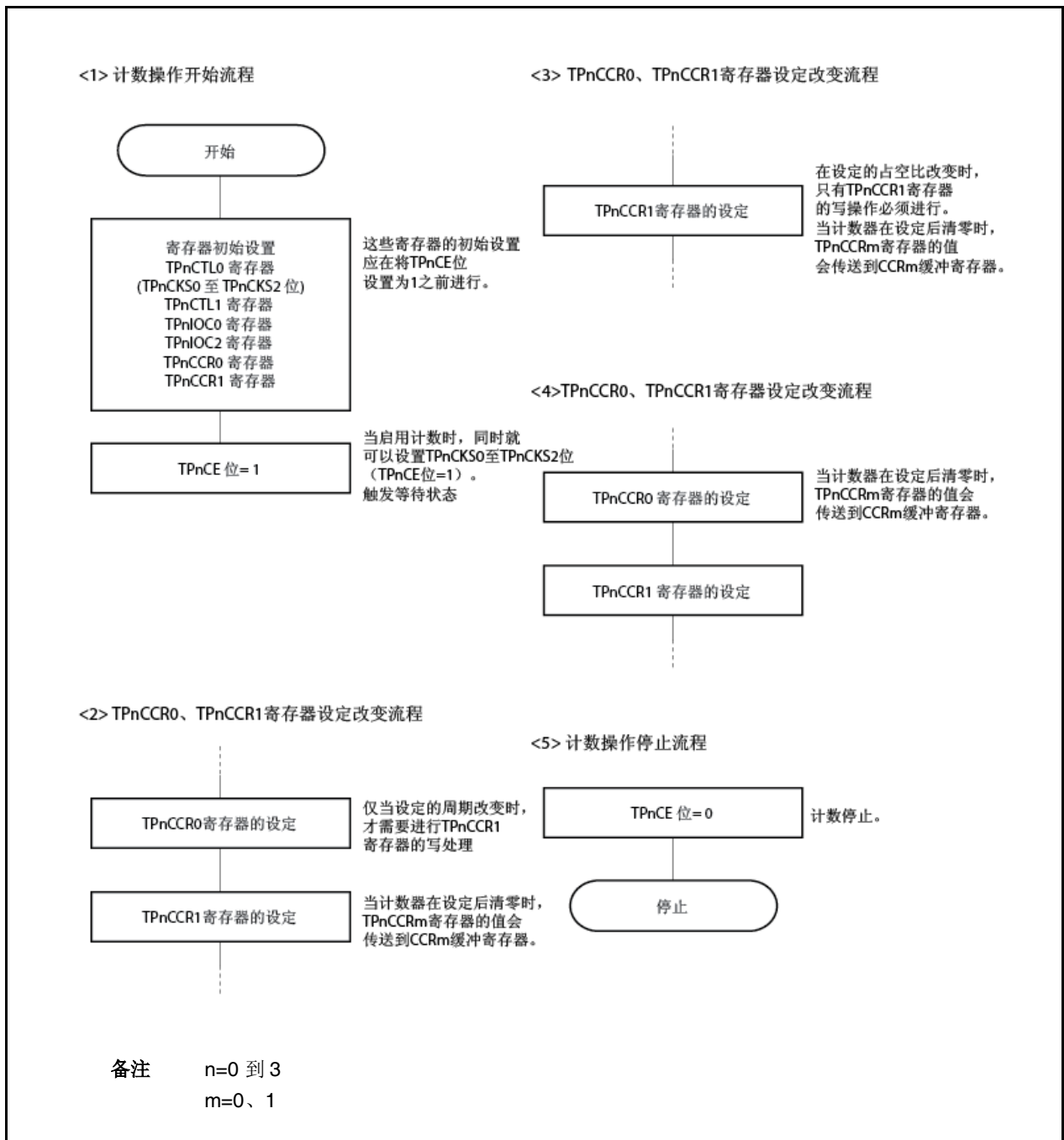


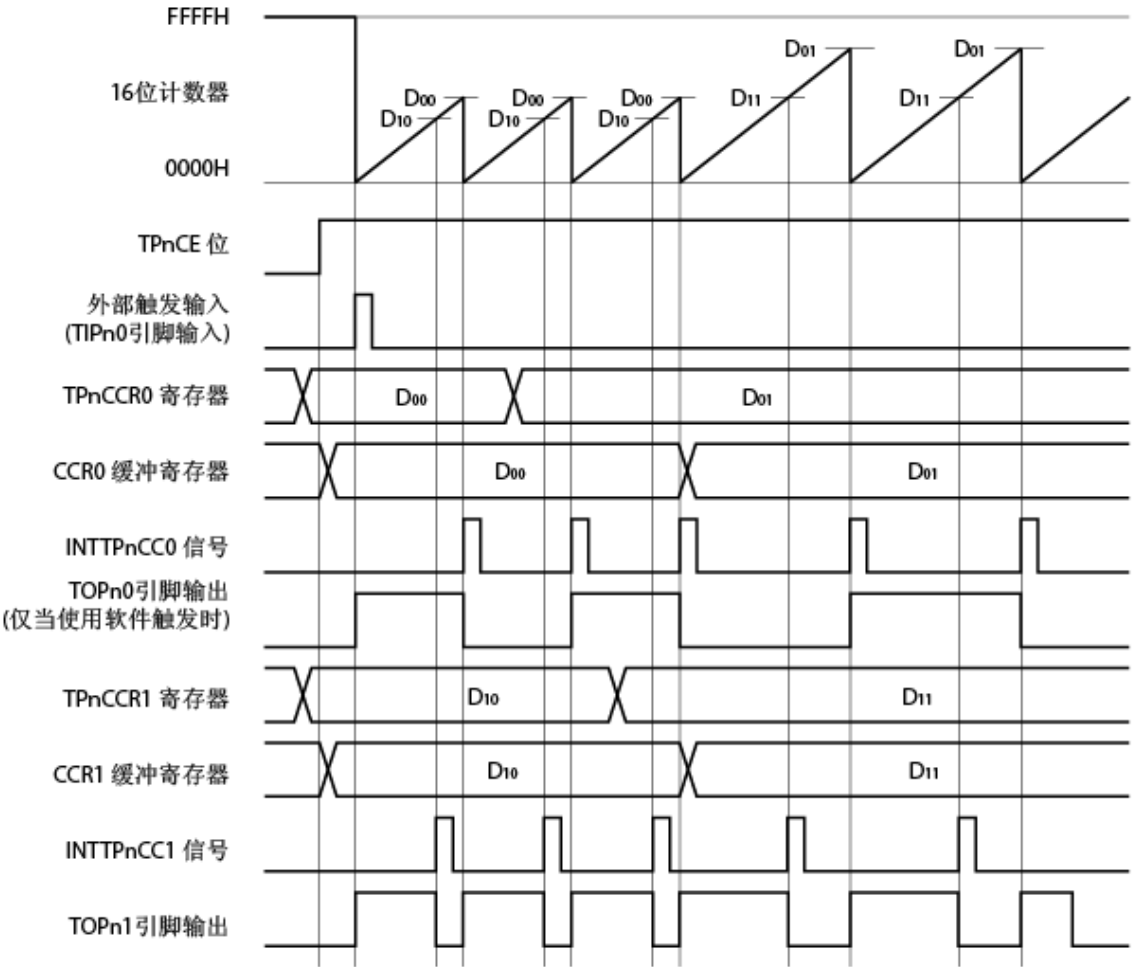
图 6-19 外部触发脉冲输出模式下的软件处理流程 (2/2)



(2) 外部触发脉冲输出模式操作定时

(a) 在工作过程中改变脉冲宽度的注意点

要在计数器工作时改变 PWM 波形，应最后写入 TPnCCR1 寄存器。  
应在检测到 INTTPnCC0 信号之后写 TPnCCR1 寄存器，之后重写 TPnCCRm 寄存器。





为了将数据从 TPnCCRm 寄存器传送到 CCRm 缓冲寄存器，必须写 TPnCCR1 寄存器。

此时要改变 PWM 波形的周期和有效电平宽度，应首先将周期设定给 TPnCCR0 寄存器，然后将有效电平宽度设定给 TPnCCR1 寄存器。

要只改变 PWM 波形的周期，应首先把周期设定给 TPnCCR0 寄存器，然后将相同值写入 TPnCCR1 寄存器。

仅要改变 PWM 波形的有效电平宽度（占空比），只有 TPnCCR1 寄存器必须设置。

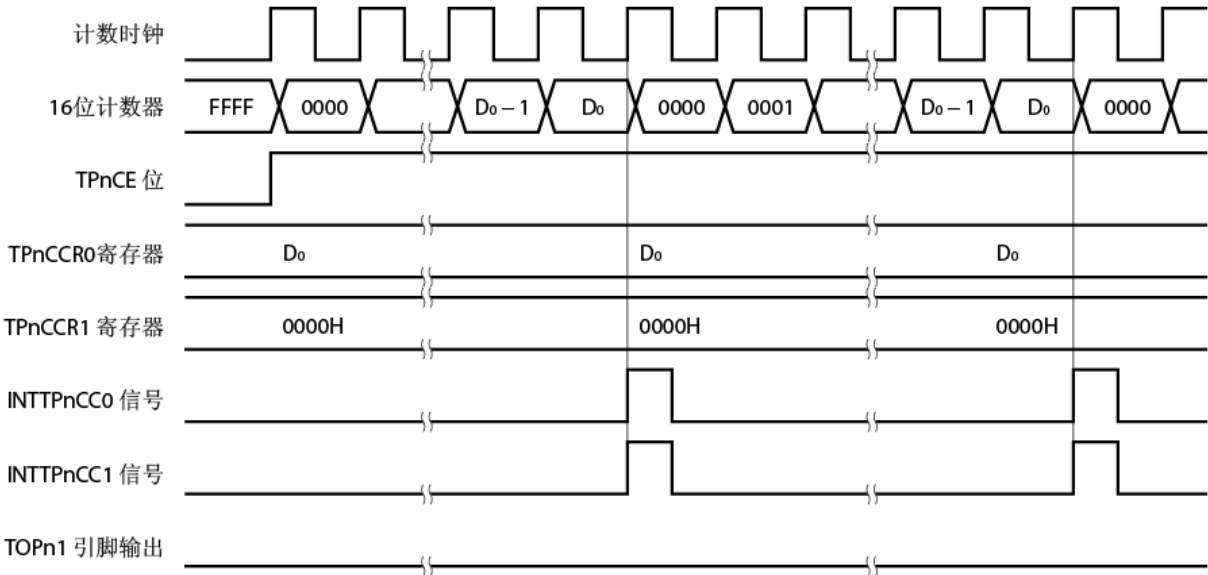
在将数据写入 TPnCCR1 寄存器后，写入 TPnCCRm 寄存器的值在 16 位计数器清零同步下传送到 CCRm 缓冲寄存器并用作与 16 位计数器进行比较的值。

要在写入 TPnCCR1 寄存器一次之后再次写 TPnCCR0 或 TPnCCR1 寄存器，应在 INTTPnCC0 信号产生之后进行。否则，因为把数据从 TPnCCRm 寄存器传送到 CCRm 缓冲寄存器的时间与写 TPnCCRm 寄存器相冲突，CCRm 缓冲寄存器的值可能变得无定义。

备注      n=0 到 3  
             m=0、1

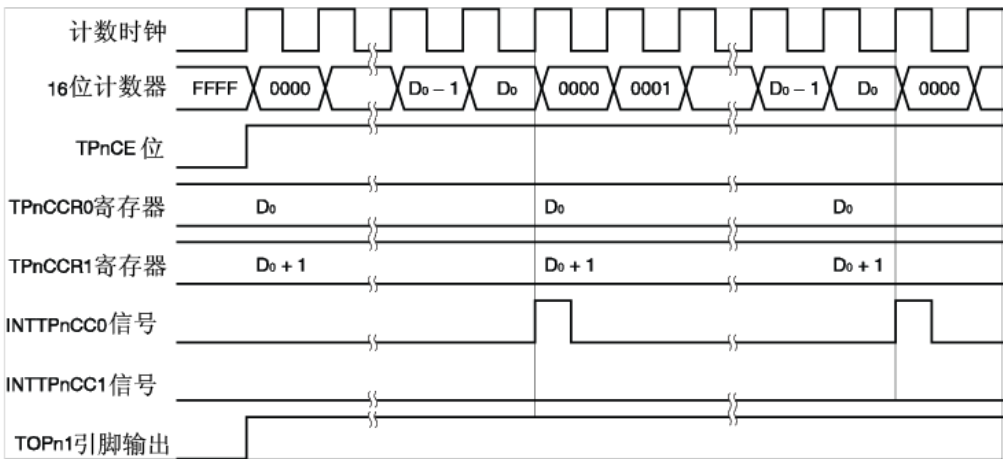
(b) PWM 波形的 0%/100%输出

要输出 0% 波形，应将 TPnCCR1 寄存器设为 0000H。若 TPnCCR0 寄存器的设定值为 FFFFH，则会周期性产生 INTTPnCC1 信号。



备注 n=0 到 3

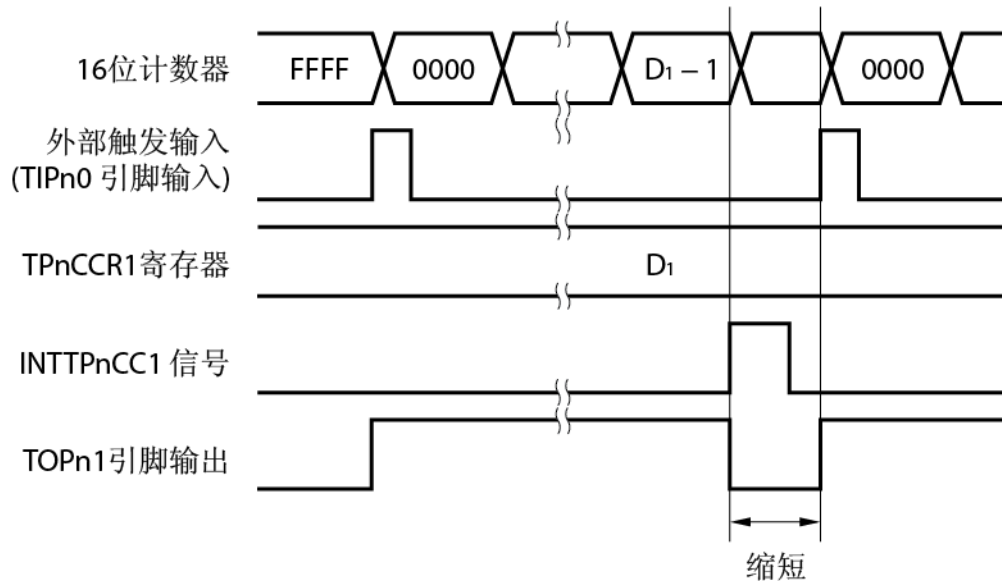
要输出 100% 波形，应将 TPnCCR1 寄存器设为 (TPnCCR0 寄存器的设置值+1)。若 TPnCCR0 寄存器的设定值是 FFFFH，则无法产生 100% 输出。



备注 n=0 到 3

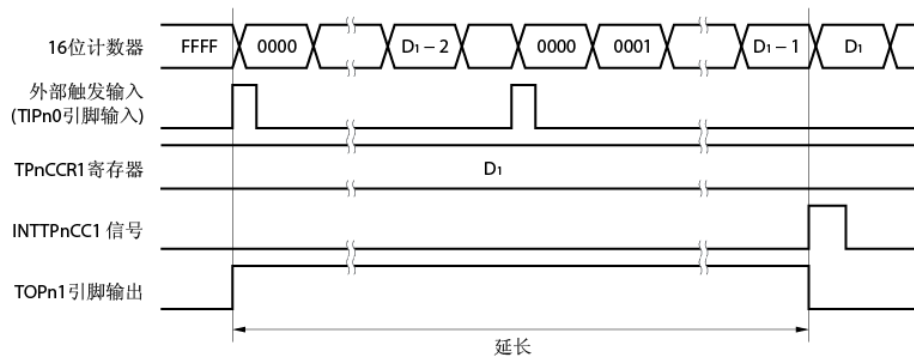
## (c) 触发检测与匹配 TPnCCR1 寄存器之间的冲突

若在 INTTPnCC1 信号生成后立刻检测到触发，则 16 位计数器立刻清零为 0000H，TOPn1 引脚的输出信号断定，计数器继续计数。结果，PWM 波形的无效时段缩短了。



备注 n=0 到 3

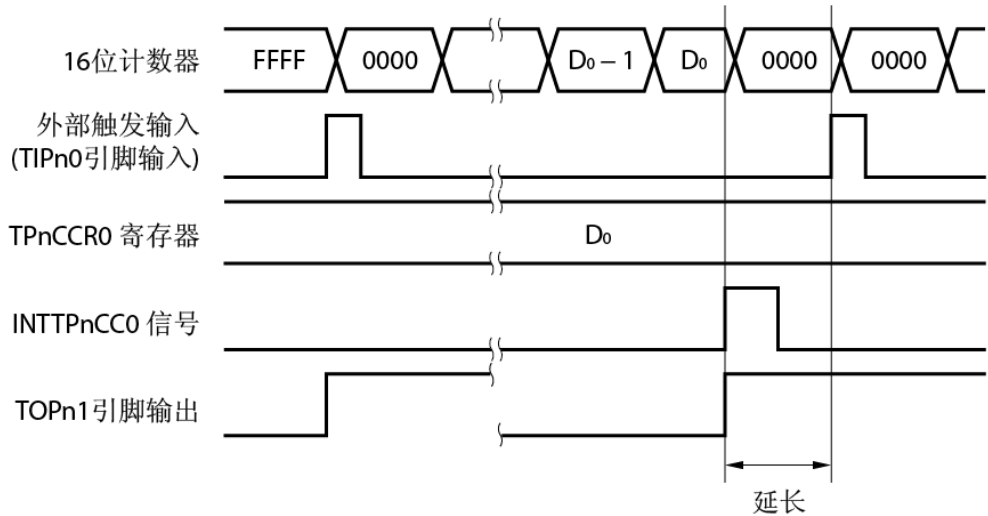
若在 INTTPnCC1 信号产生前瞬间检测到触发，则不会产生 INTTPnCC1 信号，且 16 位计数器清零为 0000H，继续计数。TOPn1 引脚的输出信号仍然有效。结果，PWM 波形的有效时段延长了。



备注 n=0 到 3

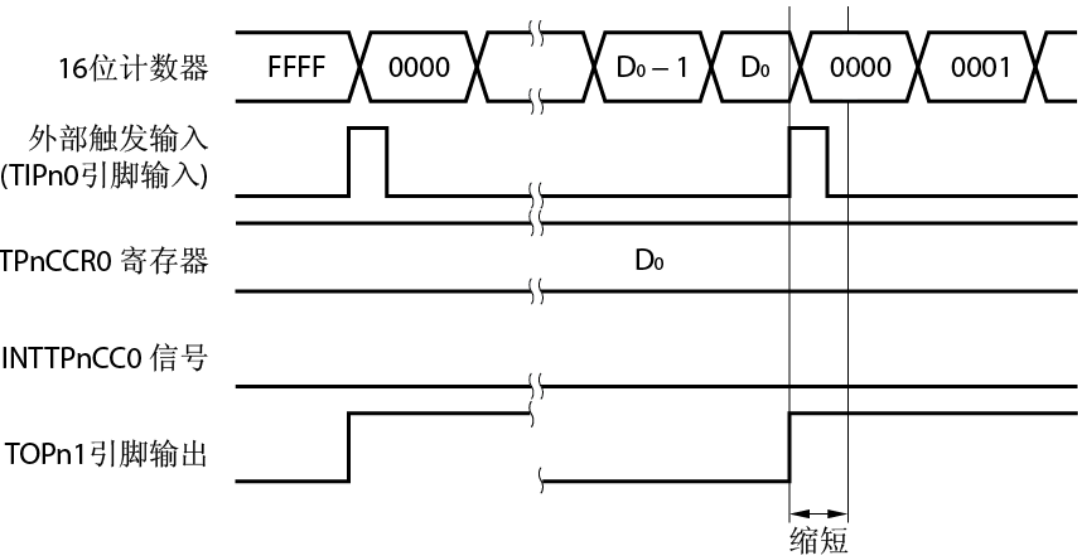
(d) 触发检测与匹配 TPnCCR0 寄存器之间的冲突

若在 INTTPnCC0 信号产生后瞬间检测到触发，则 16 位计数器清零为 0000H 且继续向上计数。因此，TOPn1 引脚的有效时段延长的部分为从产生 INTTPnCC0 信号到检测到触发的时间。



备注 n=0 到 3

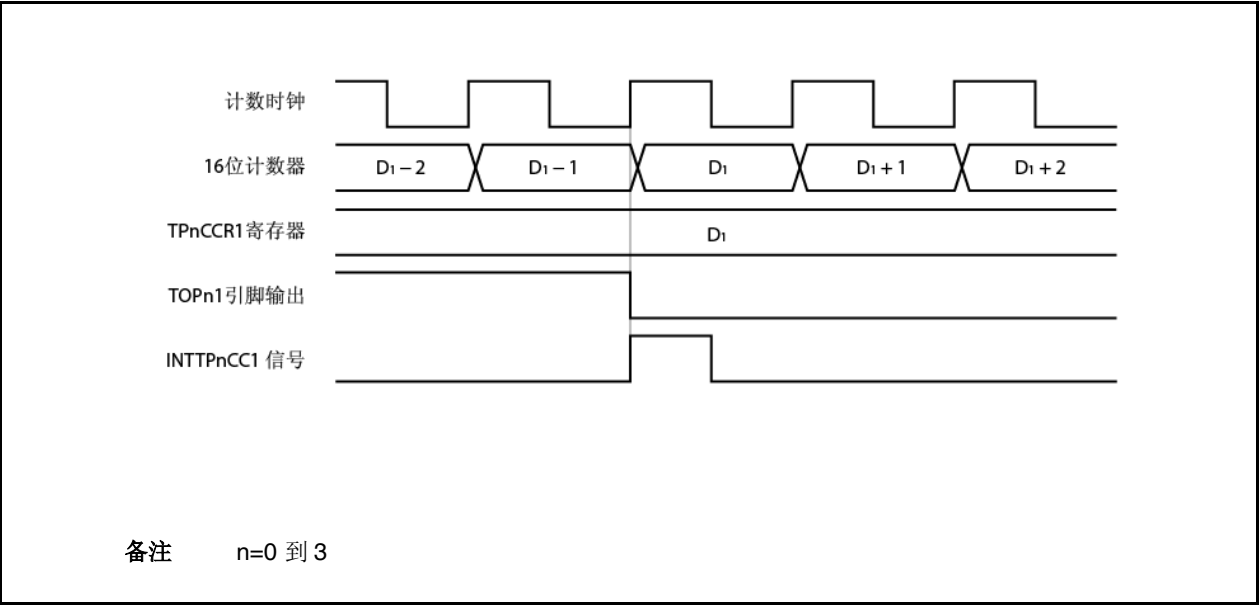
若在产生 INTTPnCC0 信号之前瞬间检测到触发，则 INTTPnCC0 信号不会产生。16 位计数器清零为 0000H，TOPn1 引脚输出断定，计数器继续计数。结果，PWM 波形的无效时段缩短了。



备注 n=0 到 3

(e) 比较匹配中断请求信号 (INTTPnCC1) 的产生定时

Inttpncc1 信号在外部触发脉冲输出模式下的产生定时与其它 INTTPnCC1 信号的定时不同：当 16 位计数器的计数值与 TPnCCR1 寄存器的值匹配时，产生 INTTPnCC1 信号。



通常，INTTPnCC1 信号的产生与下一次向上计数同步，且在 16 位计数器的计数值与 TPnCCR1 寄存器的值匹配之后。

但是，在外部触发脉冲输出模式下，其产生会提前一个时钟。这是因为为了匹配改变 TOPn1 引脚输出信号的定时，改变了该定时。

#### 6.5.4 单次脉冲输出模式 (TPnMD2 到 TPnMD0 位=011)

在单次脉冲输出模式下, 当 TPnCTL0.TPnCE 位设置为 1 时 16 位定时器/事件计数器 P 等待触发。当检测到外部触发的有效边缘时, 16 位定时器/事件计数器 P 开始计数并从 TOPn1 引脚输出单次脉冲。

除外部触发外, 也可生成软件触发来输出脉冲。当使用软件触发时, TOPn0 引脚在 16 位计数器计数时输出有效电平, 当计数器停止 (等待触发) 时输出无效电平。

图 6-20 单次脉冲输出模式下的配置

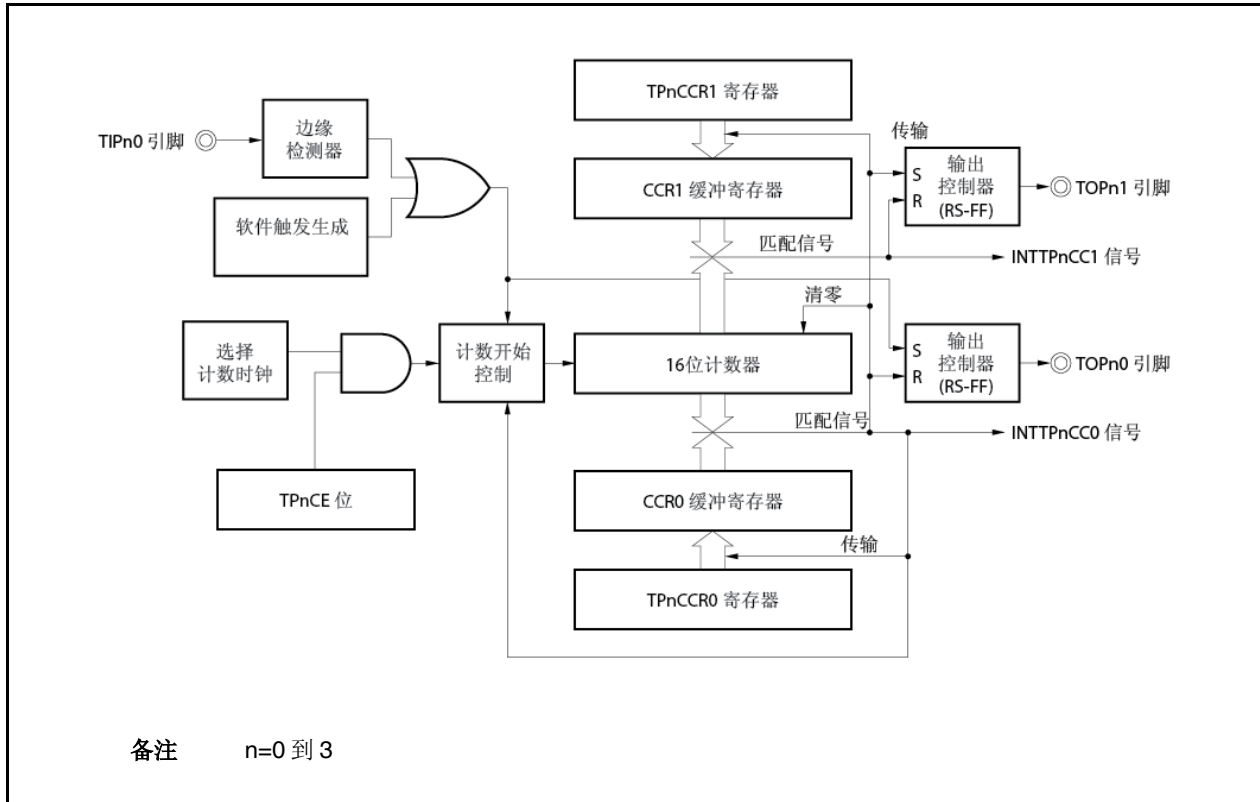
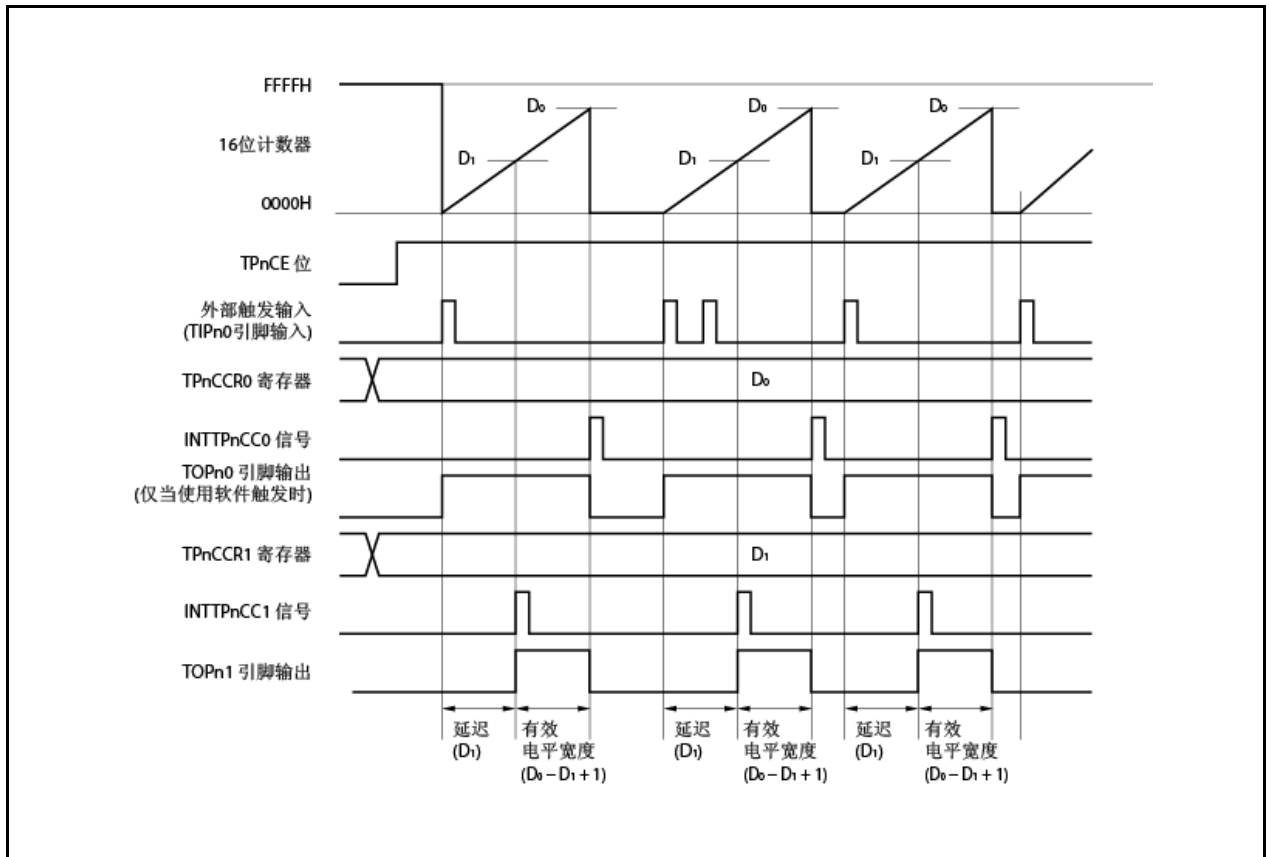


图 6-21 单次脉冲输出模式下的基本定时



当 TPnCE 位设置为 1 时 16 位定时器/事件计数器 P 等待触发。当触发生成时，16 位计数器从 FFFFH 清零为 0000H，开始计数，并从 TOPn1 引脚输出一个单次脉冲。在输出单次脉冲后，16 位计数器设置为 FFFFH，停止计数，等待触发。如果当单次脉冲正在输出时再次生成触发，则该触发会被忽略。

单次脉冲的延迟时段和有效电平宽度可计算如下。

输出延迟时段 = (TPnCCR1 寄存器的设置值) × 计数时钟周期

有效电平宽度 = (TPnCCR0 寄存器的设置值 - TPnCCR1 寄存器的设置值 + 1) × 计数时钟周期

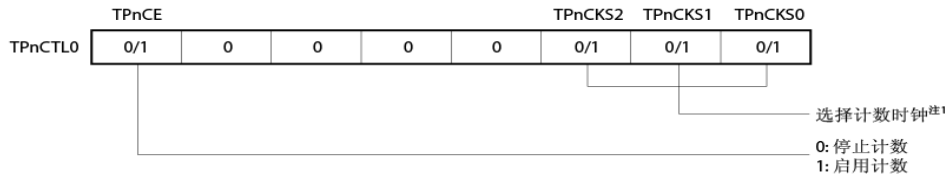
当 16 位计数器在其计数值与 CCR0 缓冲寄存器的值匹配之后进行计数时，会产生比较匹配中断请求信号 INTTPnCC0。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值相符时，就会生成比较匹配中断请求信号 INTTPnCC1。

外部触发输入的有效边缘或将软件触发 (TPnCTL1.TPnEST 位) 设置为 1 均可用来作为触发。

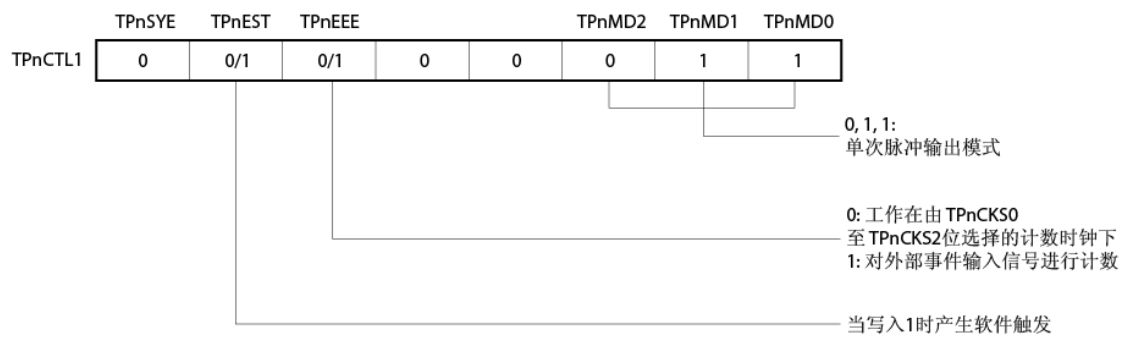
备注     n=0 到 3  
           m=0、1

图 6-22 单次脉冲输出模式下寄存器的设定 (1/2)

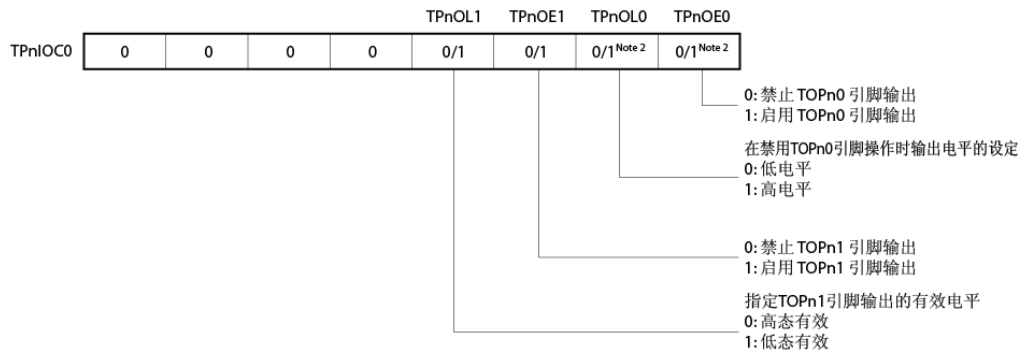
## (a) TMPn 控制寄存器 0 (TPnCTL0)



## (b) TMPn 控制寄存器 1 (TPnCTL1)



## (c) TMPn I/O 控制寄存器 0 (TPnIOC0)



• 当 TPnOL1 位= 0 时



• 当 TPnOL1 位= 1 时

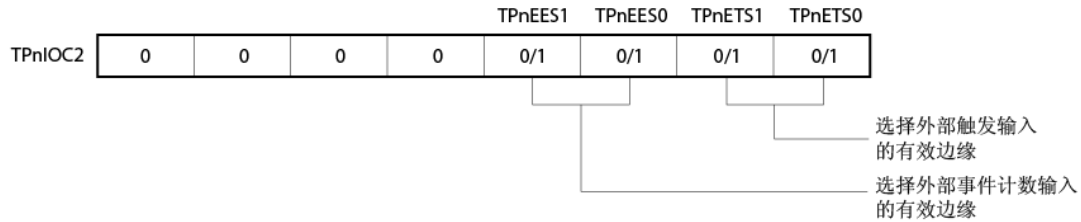


- 注 1. 当 TPnCTL1.TPnEEE 位=1 时, 设置无效。  
2. 在单次脉冲输出模式下不使用 TOPn0 引脚时, 应将此位清为 0。



图 6-22 单次脉冲输出模式下寄存器的设定 (2/2)

## (d) TMPn I/O 控制寄存器 2 (TPnIOC2)



## (e) TMPn 计数器读取缓冲寄存器 (TPnCNT)

通过读 TPnCNT 寄存器，可以读取 16 位计数器的值。

## (f) TMPn 捕捉/比较寄存器 0、1 (TPnCCR0、TPnCCR1)

若将 D<sub>0</sub> 设置为 TPnCCR0 寄存器，将 D<sub>1</sub> 设置给 TPnCCR1 寄存器，则单次脉冲的有效电平宽度和输出延迟时段如下所示。

有效电平宽度 = (D<sub>0</sub> - D<sub>1</sub> + 1) × 计数时钟周期

输出延迟时段 = (D<sub>1</sub>) × 计数时钟周期

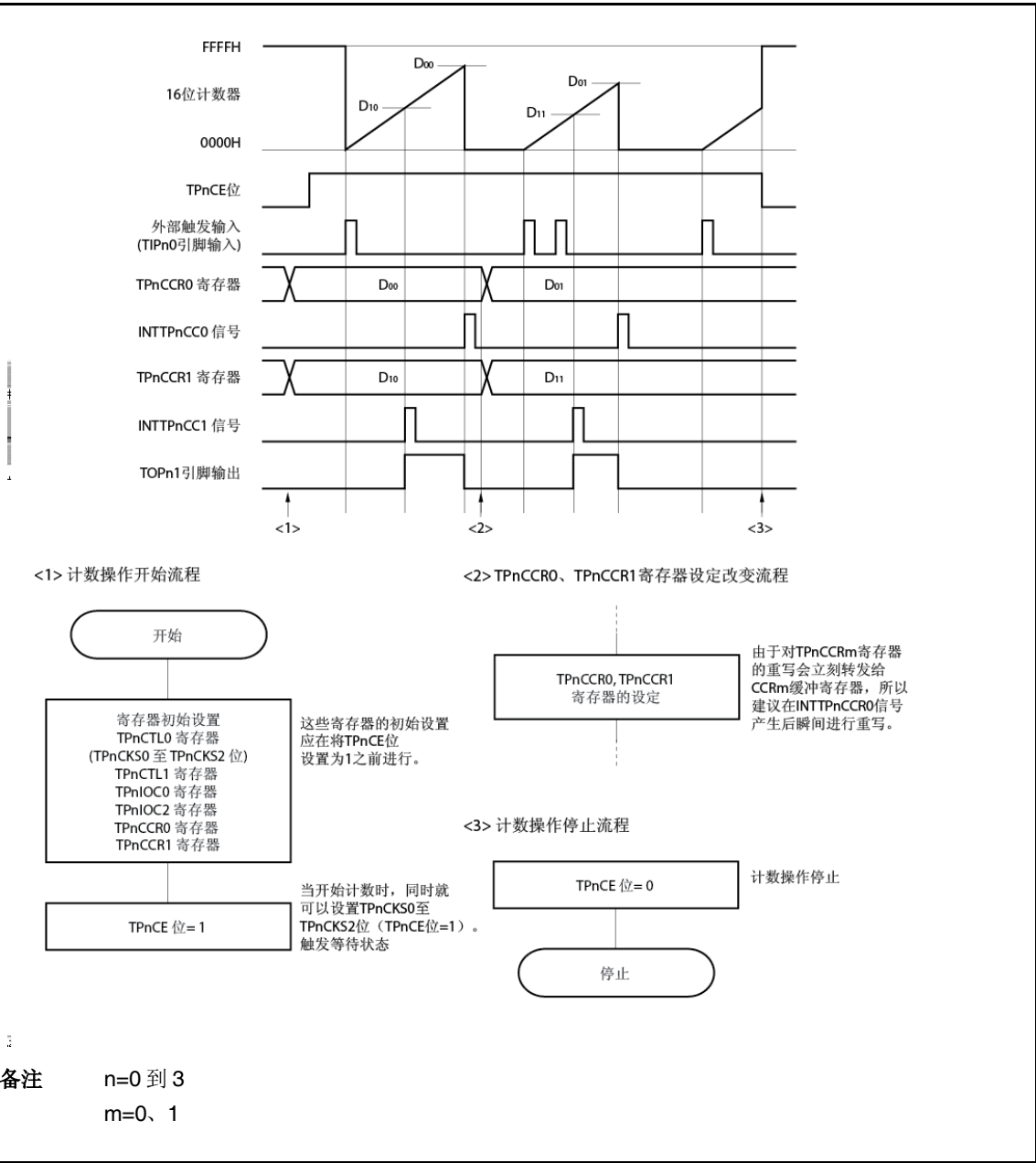
**注意事项** 如果 TPnCCR1 寄存器的设置值大于 TPnCCR0 寄存器的值，则即便在单次脉冲输出模式下也不会输出单次脉冲。

**备注** 1. TMPn I/O 控制寄存器 1 (TPnIOC1) 和 TMPn 选项寄存器 0 (TPnOPT0) 在单次脉冲输出模式下不使用。

2. n=0 到 3

(1) 单次脉冲输出模式下的工作流程

图 6-23 单次脉冲输出模式下的工作流程

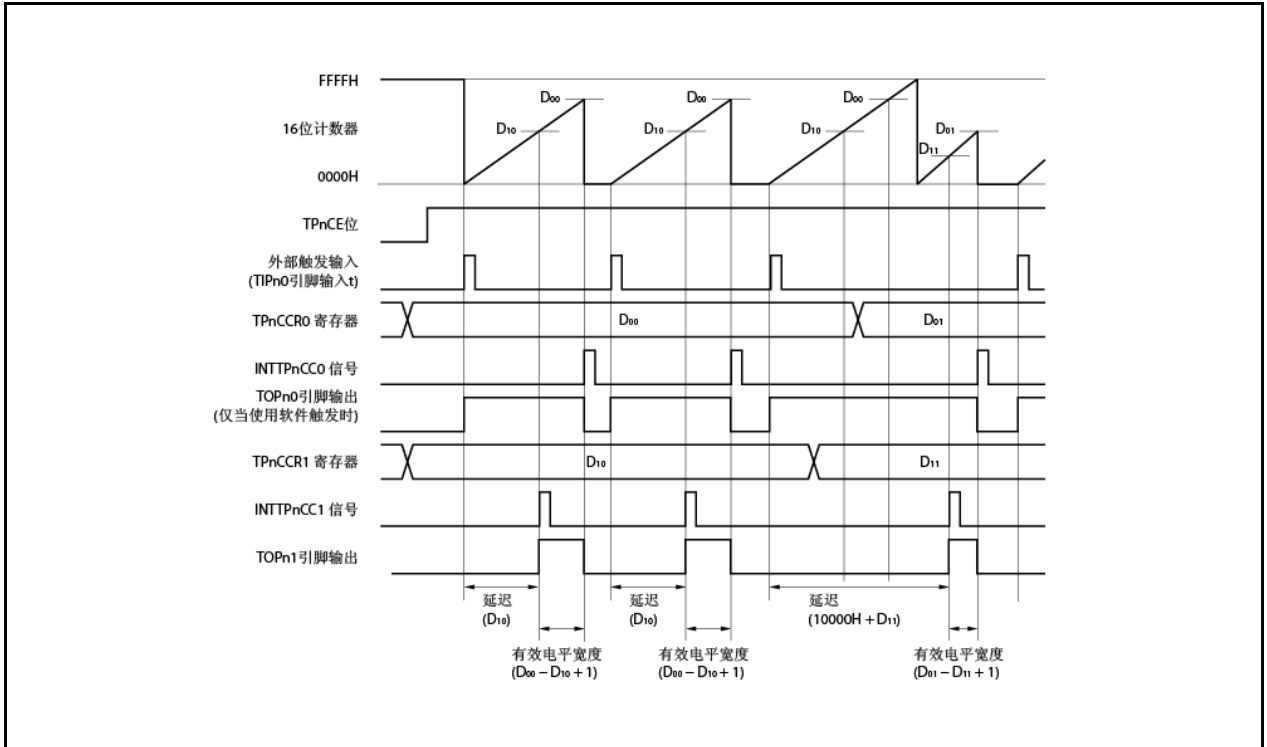


## (2) 单次脉冲输出模式下的操作定时

## (a) 关于重写 TPnCCRm 寄存器的注意点

要将 TPnCCRm 寄存器的设置值更改为较小的值，应停止计数一次，再改变设置值。

如果在计数过程中 TPnCCRm 寄存器的值被重写为较小的值，则 16 位计数器会溢出。



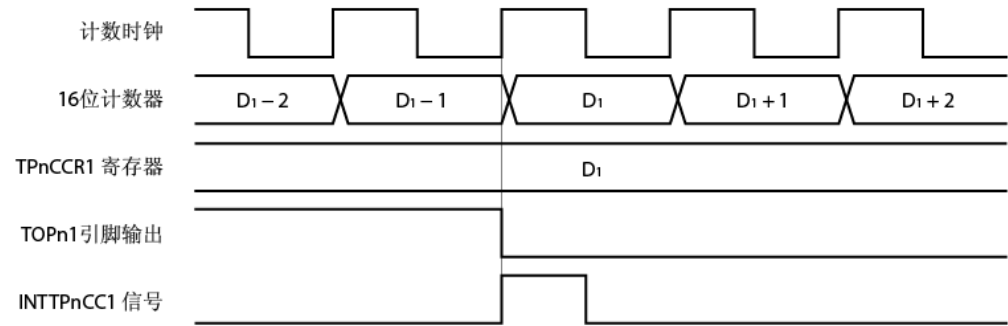
当 TPnCCR0 寄存器从  $D_{00}$  重写为  $D_{01}$ 、TPnCCR1 寄存器从  $D_{10}$  重写为  $D_{11}$ （其中  $D_{00} > D_{01}$  且  $D_{10} > D_{11}$ ）时，如果重写 TPnCCR1 寄存器时 16 位计数器的计数值大于  $D_{11}$  且小于  $D_{10}$ 、重写 TPnCCR0 寄存器时计数值大于  $D_{01}$  且小于  $D_{00}$ ，则在重写寄存器并与计数值进行比较后，各设置值会尽快反映出来。计数器向上计数到 FFFFH，然后从 0000H 开始重新向上计数。当计数值与  $D_{11}$  匹配时，计数器会产生 INTTPnCC1 信号并断定 TOPn1 引脚。当计数值匹配  $D_{01}$  时，计数器会产生 INTTPnCC0 信号，对 TOPn1 引脚解除断定，停止计数。

因此，计数器输出的脉冲的延迟时段或有效时段可能与原先预计的单次脉冲不同。

备注       $n=0$  到 3  
              $m=0, 1$

(b) 比较匹配中断请求信号 (INTTPnCC1) 的产生定时

在单次脉冲输出模式下 INTTPnCC1 信号的产生定时与 INTTPnCC1 信号不同；当 16 位计数器的计数值与 TPnCCR1 寄存器的值匹配时产生 INTTPnCC1 信号。



备注 n=0 到 3

通常，在 16 位计数器的计数值匹配 TPnCCR1 寄存器的值之后的下次向上计数时产生 INTTPnCC1 信号。但是，在单次脉冲输出模式下会提前一个时钟产生。这是因为为了匹配 TOPn1 引脚的定时，该定时被改变了。

备注 n=0 到 3

### 6.5.5 PWM 输出模式 (TPnMD2 到 TPnMD0 位=100)

在 PWM 输出模式，当 TPnCTL0.TPnCE 位设置为 1 时会从 TOPn1 引脚输出 PWM 波形。

此外，会从 TOPn0 引脚输出一个脉冲，其半周期为 PWM 波形的一个周期。

图 6-24 PWM 输出模式下的配置

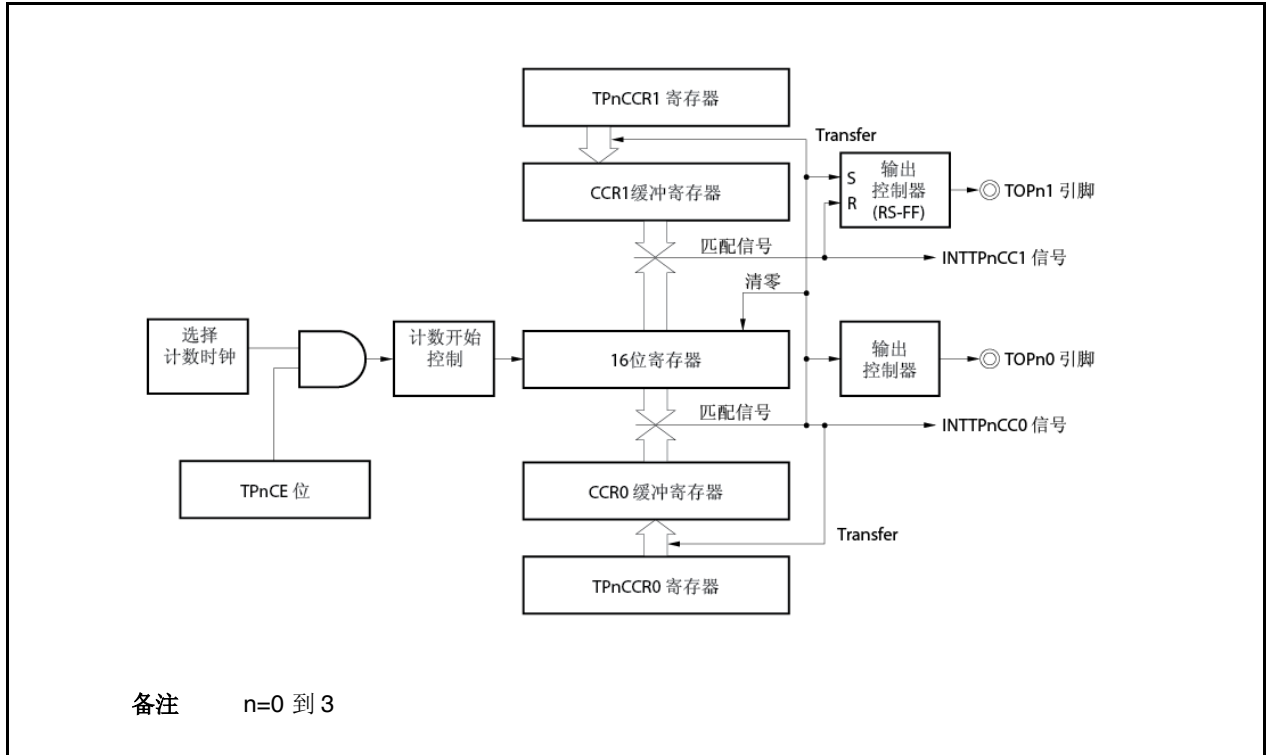
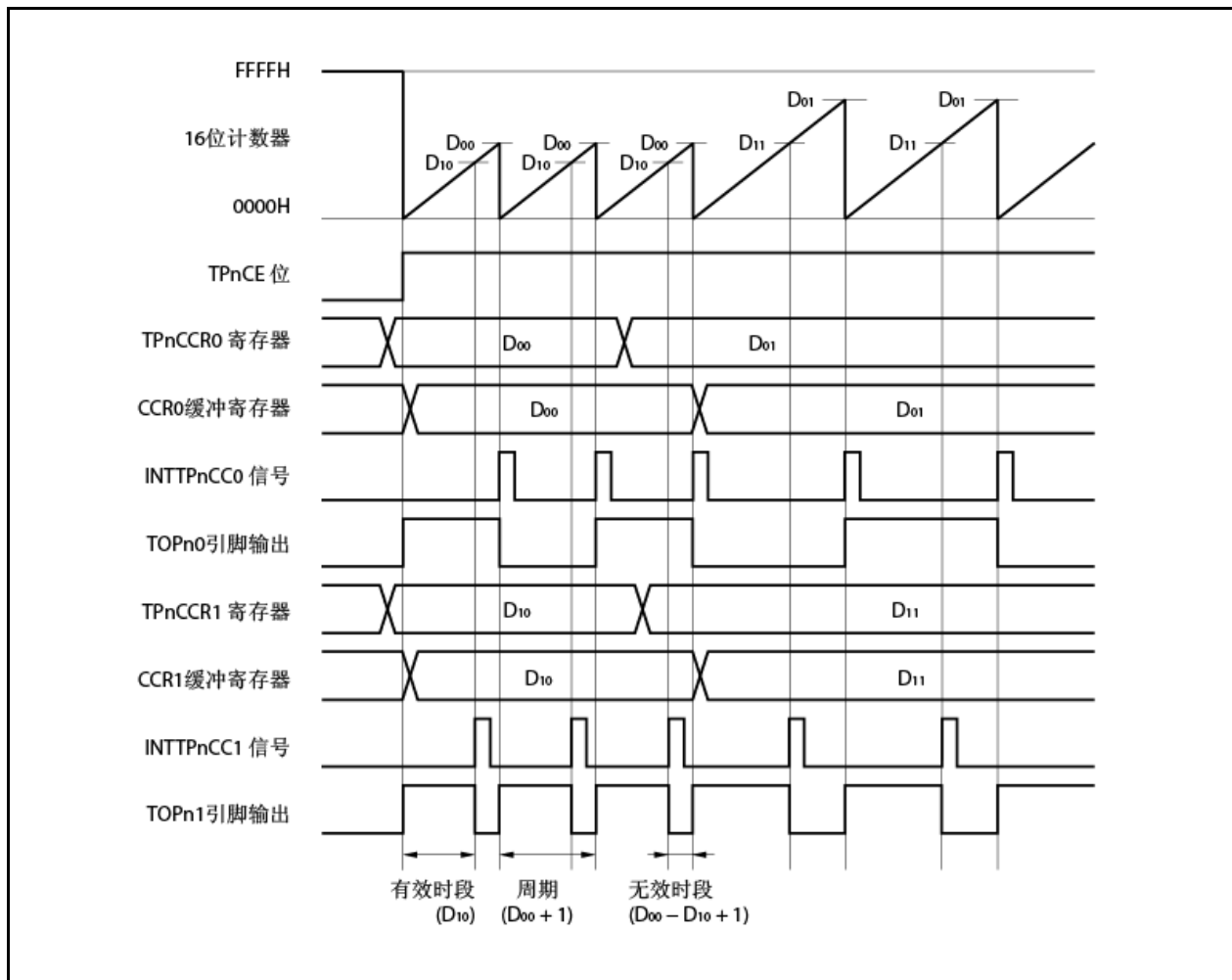


图 6-25 PWM 输出模式下的基本定时



当 TPnCE 位设置为 1 时，16 位计数器从 FFFFH 清零为 0000H，开始计数，从 TOPn1 引脚输出 PWM 波形。PWM 波形的有效电平宽度、周期、占空比可计算如下。

有效电平宽度 = (TPnCCR1 寄存器的设置值) × 计数时钟周期

周期 = (TPnCCR0 寄存器的设置值 + 1) × 计数时钟周期

占空比 = (TPnCCR1 寄存器的设置值) / (TPnCCR0 寄存器的设置值 + 1)

在计数正在进行时，可通过重写 TPnCCRm 寄存器改变 PWM 波形。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配时，新写入的值会得到反映，且 16 位计数器清零为 0000H。

16 位计数器的计数值匹配 CCR0 缓冲寄存器的值之后的下次向上计数时会产生比较匹配中断请求信号 INTTPnCC0，且 16 位计数器清零为 0000H。16 位计数器的计数值匹配 CCR1 缓冲寄存器的值时，会产生比较匹配中断请求信号 INTTPnCC1。

当 16 位计数器的计数值与 CCRm 缓冲寄存器的值相匹配时，TPnCCRm 寄存器设定的值会传送到 CCRm 缓冲寄存器，且 16 位计数器清零为 0000H。

备注 n=0 到 3、m=0、1

图 6-26 PWM 输出模式下寄存器的设置 (1/2)

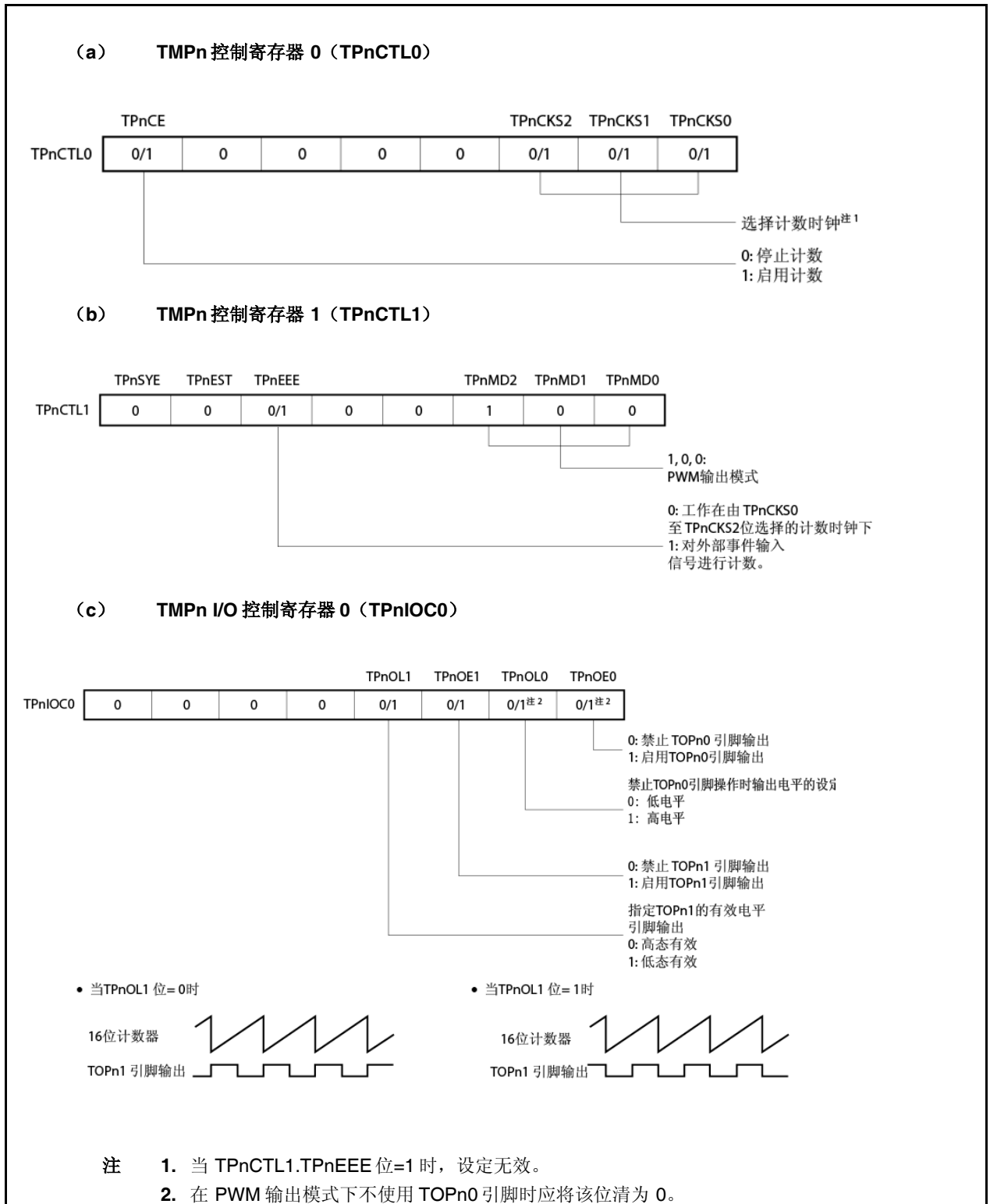
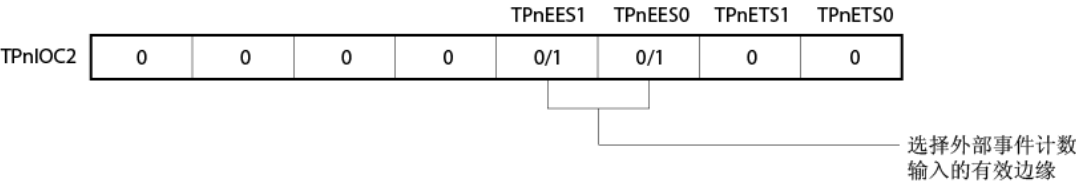


图 6-26 PWM 输出模式下寄存器的设置 (2/2)

(d) TMPn I/O 控制寄存器 2 (TPnIOC2)



(e) TMPn 计数器读取缓冲寄存器 (TPnCNT)

通过读取 TPnCNT 寄存器可读取 16 位计数器的值。

(f) TMPn 捕捉/比较寄存器 0、1 (TPnCCR0、TPnCCR1)

若 TPnCCR0 寄存器设为  $D_0$ ，TPnCCR1 寄存器设为  $D_1$ ，则 PWM 波形的周期和有效电平如下。

周期 =  $(D_0 + 1) \times$  计数时钟周期

有效电平宽度 =  $D_1 \times$  计数时钟周期

- 备注
- 1. TMPn I/O 控制寄存器 1 (TPnIOC1) 和 TMPn 选项寄存器 0 (TPnOPT0) 在 PWM 输出模式下不使用。
  - 2. n=0 到 3



(1) PWM 输出模式的操作流程

图 6-27 PWM 输出模式下的软件处理流程 (1/2)

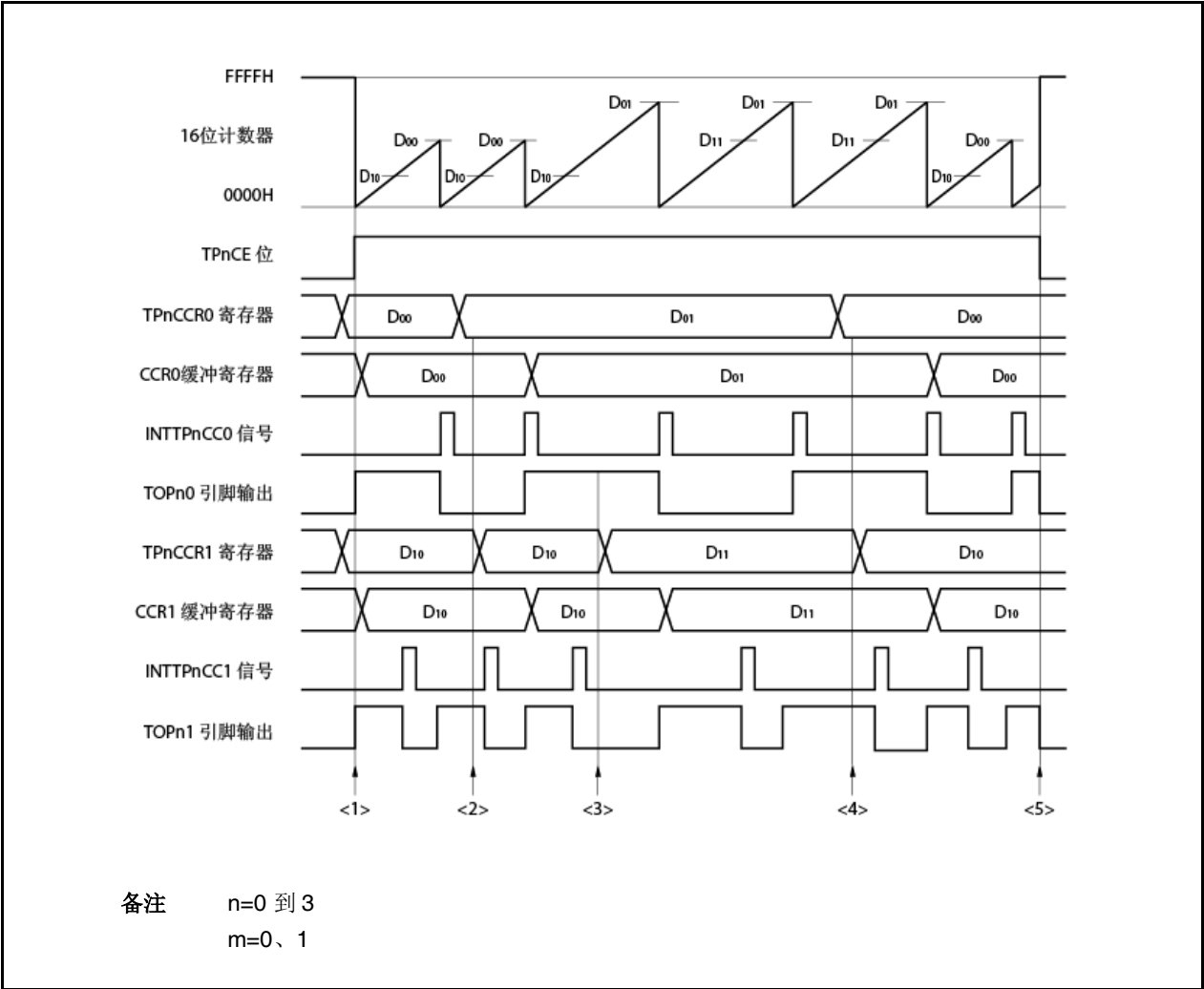
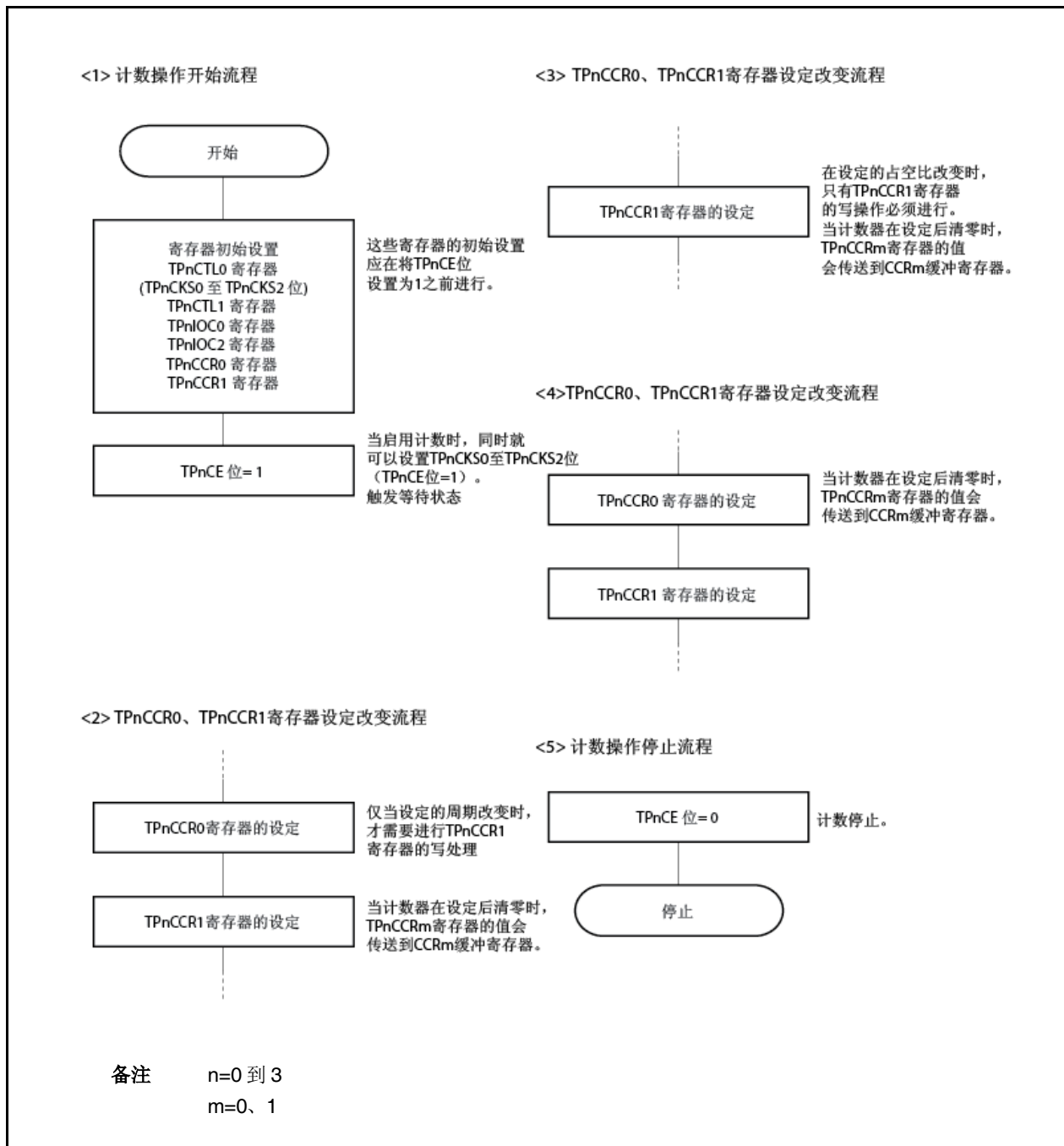


图 6-27 PWM 输出模式下的软件处理流程 (2/2)

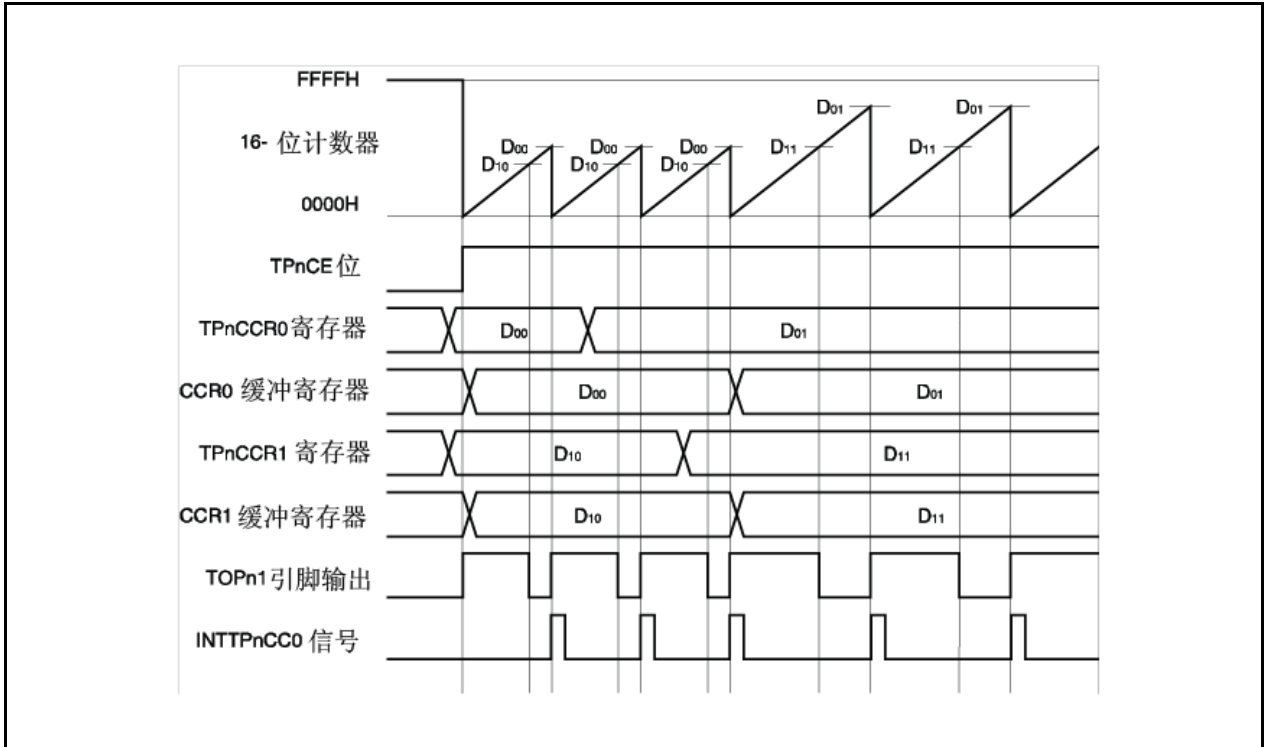


## (2) PWM 输出模式操作定时

## (a) 在操作过程中改变脉冲宽度

要在计数器工作时改变 PWM 波形，应最后写 TPnCCR1 寄存器。

应在检测到 INTTPnCC1 信号后写入 TPnCCR1 寄存器，之后再重写 TPnCCRm 寄存器。



要将数据从 TPnCCRm 寄存器传送到 CCRm 缓冲寄存器中，必须写入 TPnCCR1 寄存器。

此时要改变 PWM 波形的周期和有效电平，应首先将周期设置给 TPnCCR0 寄存器，然后将有效电平设置给 TPnCCR1 寄存器。

要只改变 PWM 波形的周期，应首先把周期设定给 TPnCCR0 寄存器，然后将相同值写入 TPnCCR1 寄存器。

要只改变 PWM 波形的有效电平宽度（占空比），只需要设置 TPnCCR1 寄存器。

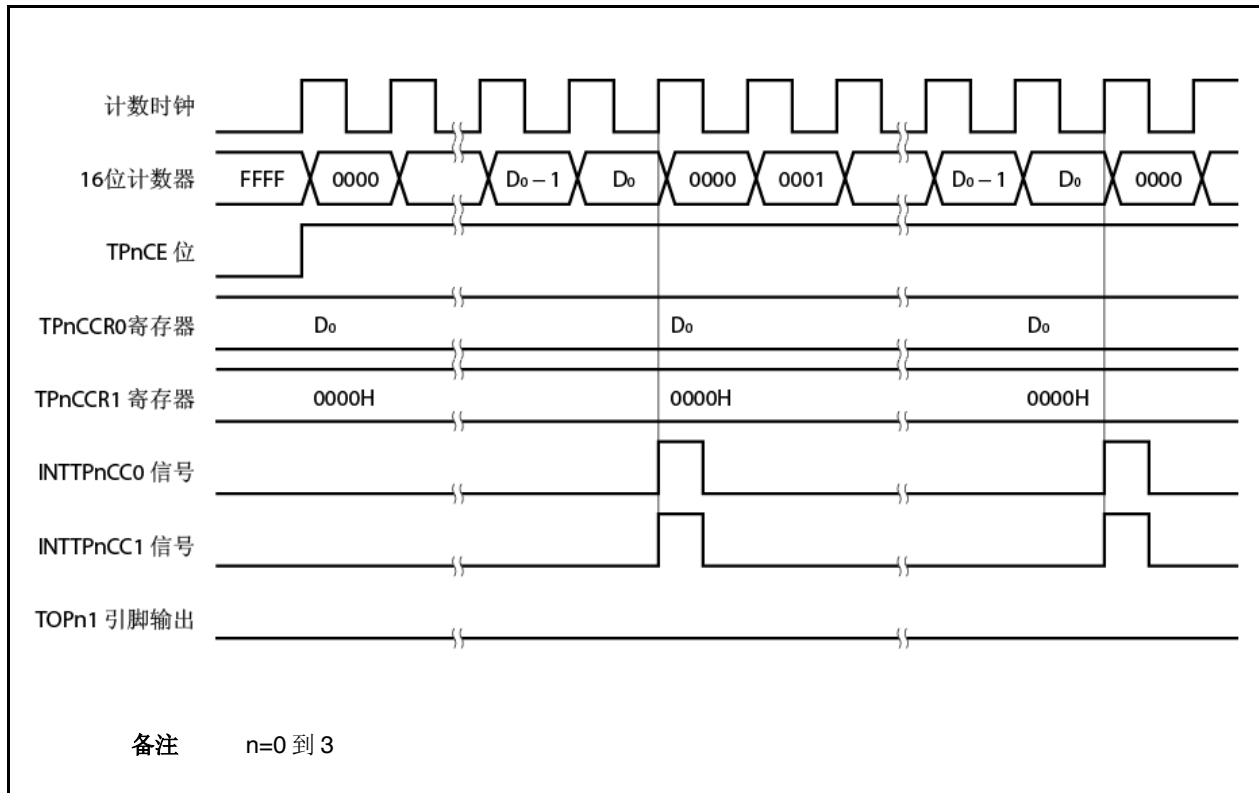
在数据写入 TPnCCR1 寄存器后，写入 TPnCCRm 寄存器的数据在 16 位计数器清零的同步下传送到 CCRm 缓冲寄存器中并被当作与 16 位计数器进行比较的值。

要在写入 TPnCCR1 寄存器一次之后再次写 TPnCCR0 或 TPnCCR1 寄存器，应在 INTTPnCC0 信号产生之后进行。否则，因为把数据从 TPnCCRm 寄存器传送到 CCRm 缓冲寄存器的时间与写 TPnCCRm 寄存器相冲突，CCRm 缓冲寄存器的值可能变得无定义。

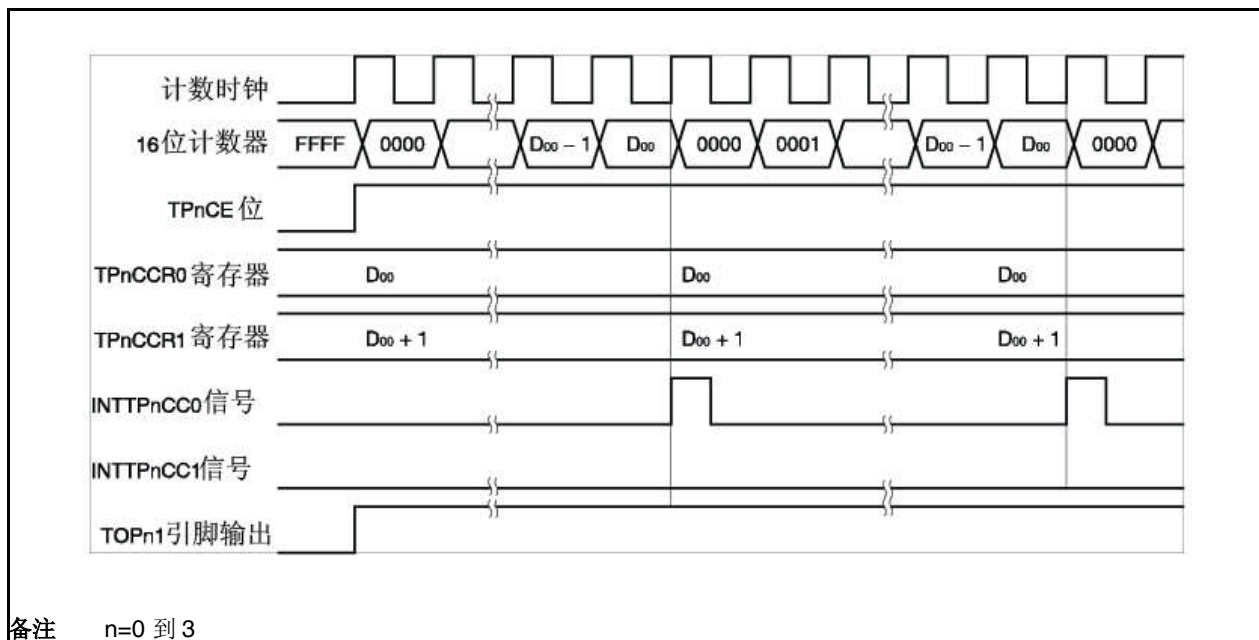
备注 n=0 到 3、m=0、1

## (b) PWM 波形的 0%/100%输出

要输出 0%波形，应将 TPnCCR1 寄存器设置为 0000H。如果 TPnCCR0 寄存器的设置值为 FFFFH，则 INTTPnCC1 信号会周期性产生。

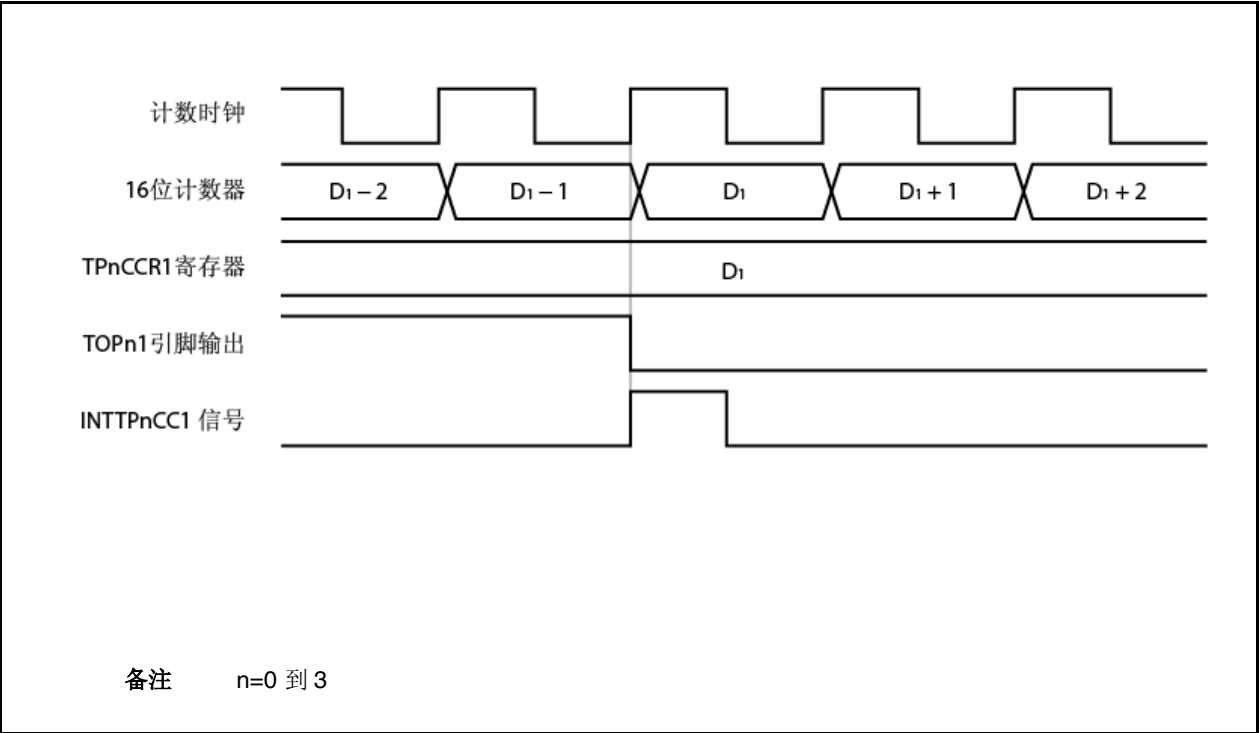


要输出 100%波形，应将 (TPnCCR0 寄存器设置值+1) 的值设置给 TPnCCR1 寄存器。如果 TPnCCR0 寄存器的设置值为 FFFFH，则无法产生 100%输出。



(c) 比较匹配中断请求信号 (INTTPnCC1) 产生定时

在 PWM 输出模式下 INTTPnCC1 信号的产生定时与其它 INTTPnCC1 信号的定时不同；当 16 位计数器的计数值匹配 TPnCCR1 寄存器的值时会产生 INTTPnCC1 信号。



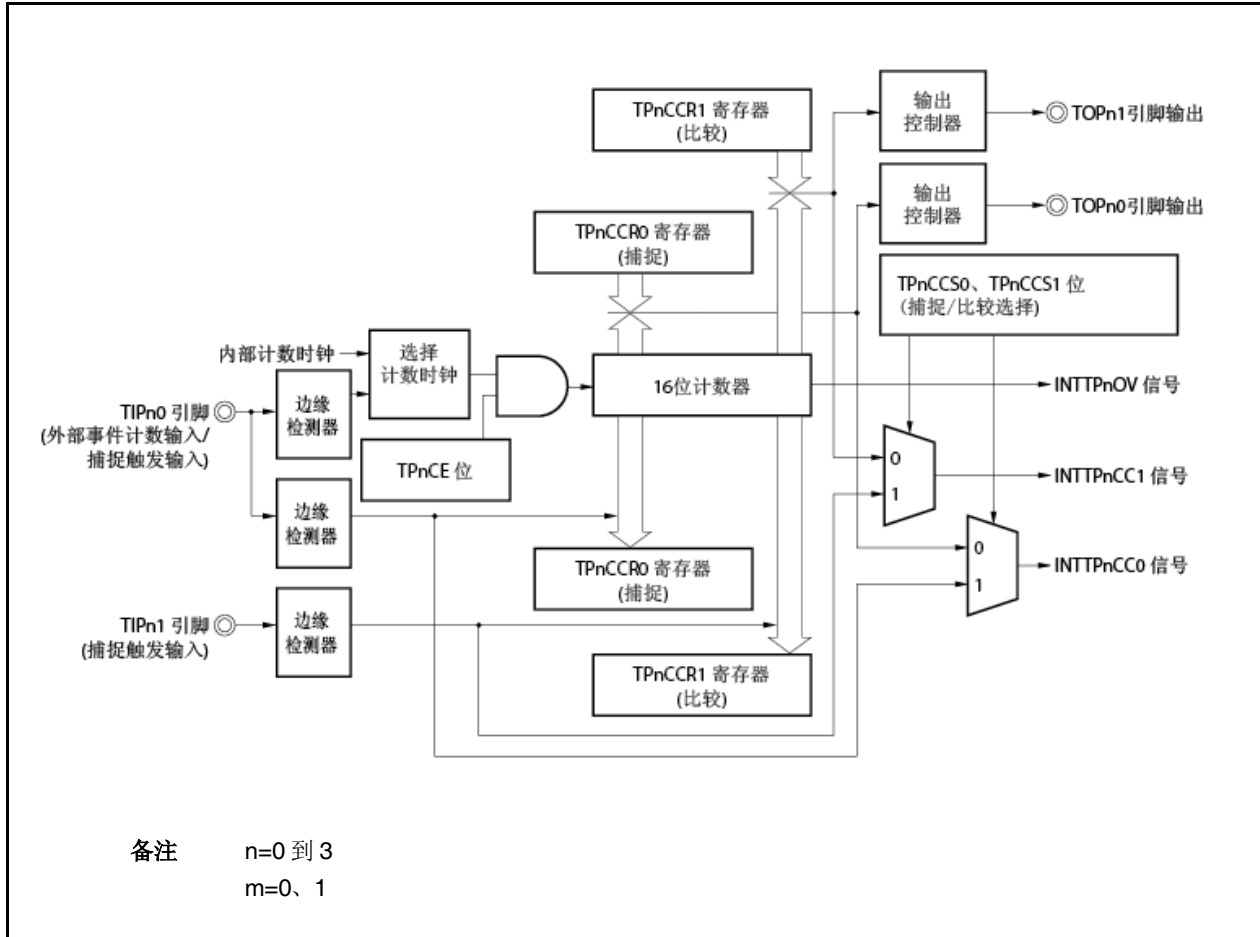
通常，在 16 位计数器的计数值匹配 TPnCCR1 寄存器的值的下次向上计数的同步下会产生 INTTPnCC1 信号。

但是，在 PWM 输出模式下，会提前一个时钟产生。这是因为为了匹配 TOPn1 引脚输出信号的改变定时，该定时也被改变了。

## 6.5.6 独立定时器模式 (TPnMD2 到 TPnMD0 位=101)

在独立定时器模式下，当 TPnCTL0.TPnCE 位设置为 1 时 16 位定时器/事件计数器 P 开始计数。此时，根据 TPnOPT0.TPnCCS0 和 TPnOPT0.TPnCCS1 位的设置，TPnCCRm 寄存器可作为比较寄存器或捕捉寄存器使用。

图 6-28 独立定时器模式的配置

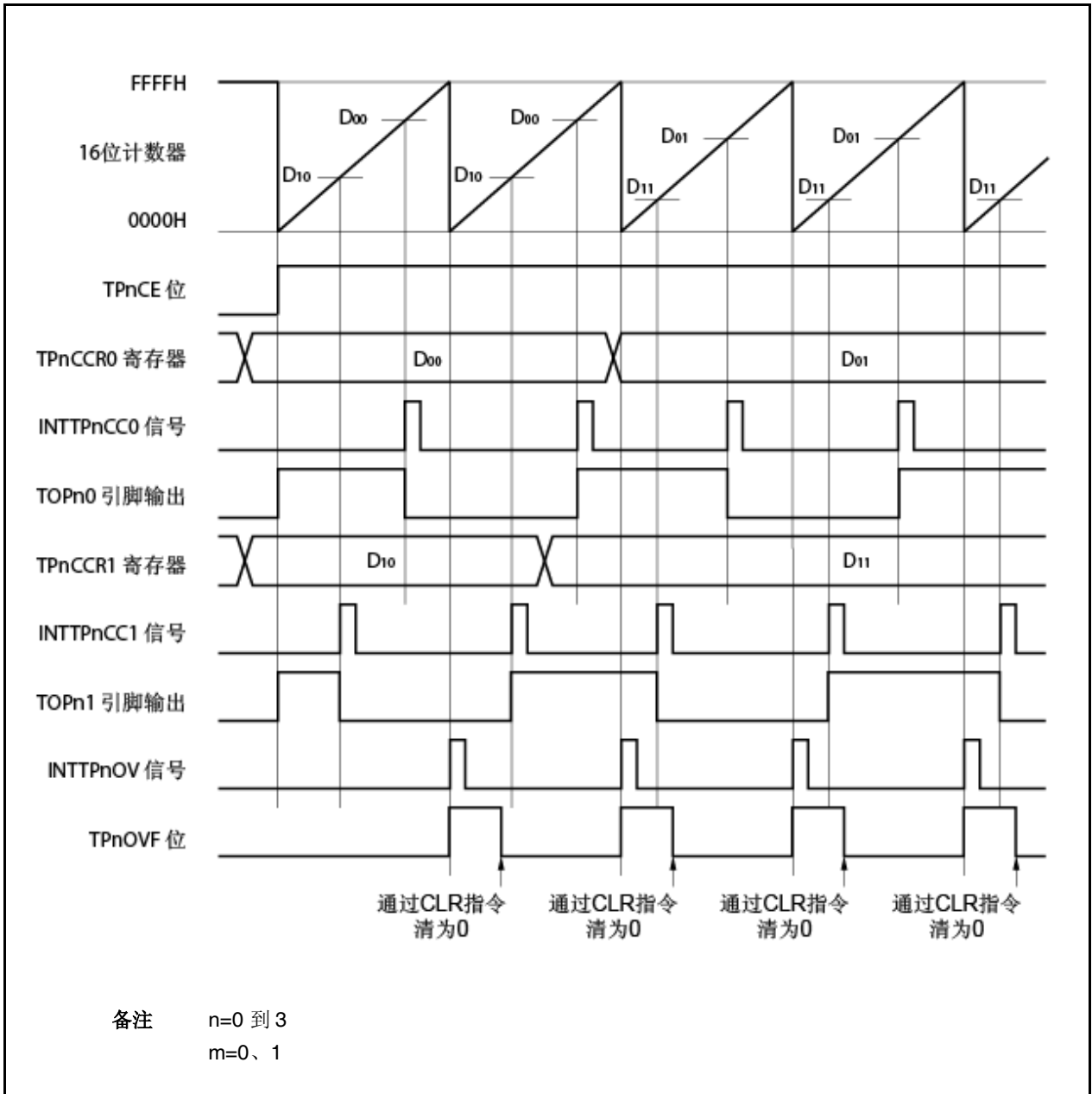


当 TPnCE 位设置为 1 时，16 位定时器/事件计数器 P 开始计数，TOPn0 和 TOPn1 引脚的输出信号反相。当 16 位计数器的计数值在此之后匹配 TPnCCRm 寄存器的设置值时，会产生比较匹配中断请求信号 (INTTPnCCm)，TOPnm 引脚的输出信号反相。

16 位计数器在计数时钟的同步下继续计数。当向上计数到 FFFFH 时，会在下个时钟产生溢出中断请求信号 (INTTPnOV)，清零为 0000H，继续计数。此时，溢出标志 (TPnOPT0.TPnOVF 位) 也设置为 1。应通过软件执行 CLR 指令将溢出标志清为 0。

当计数器正在工作时可重写 TPnCCRm 寄存器。如果进行重写，新值会在当时得到反映并与计数值进行比较。

图 6-29 独立定时器模式（比较功能）的基本定时



当 TPnCE 位设置为 1 时，16 位计数器开始计数。当检测到输入 TIPnm 引脚的有效边缘时，16 位计数器的计数值存储在 TPnCCRm 寄存器中，并产生一个捕捉中断请求信号 (INTTPnCCm)。

16 位计数器在计数时钟下继续计数。当向上计数至 FFFFH 时，会在下个时钟产生溢出中断请求信号 (INTTPnOV)，清零为 0000H，继续计数。此时，溢出标志 (TPnOPT0.TPnOVF 位) 也会设置为 1。应通过软件执行 CLR 指令将溢出标志清为 0。

图 6-30 独立定时器模式（捕捉功能）的基本定时

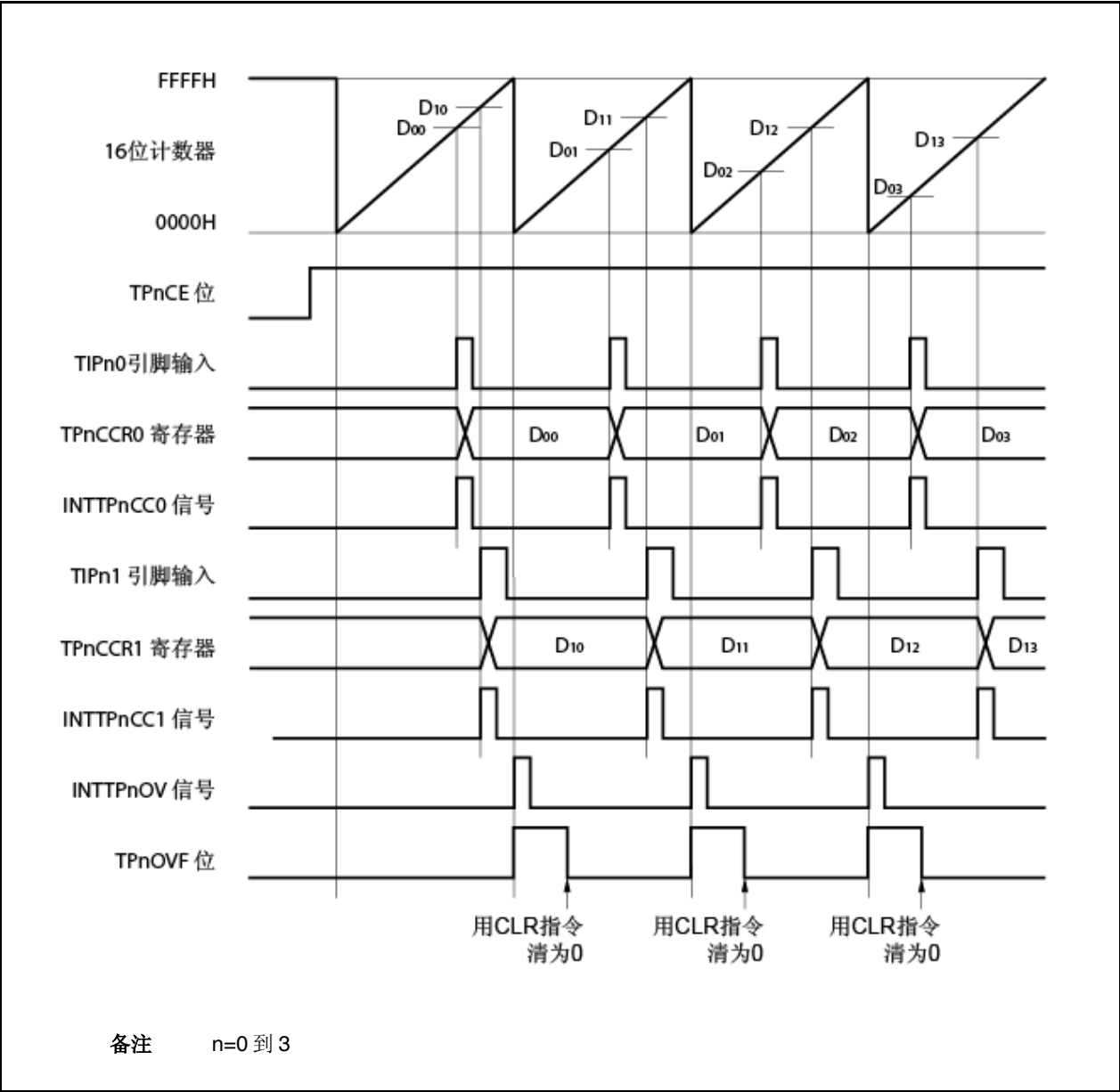




图 6-31 独立定时器模式的寄存器设置 (1/2)

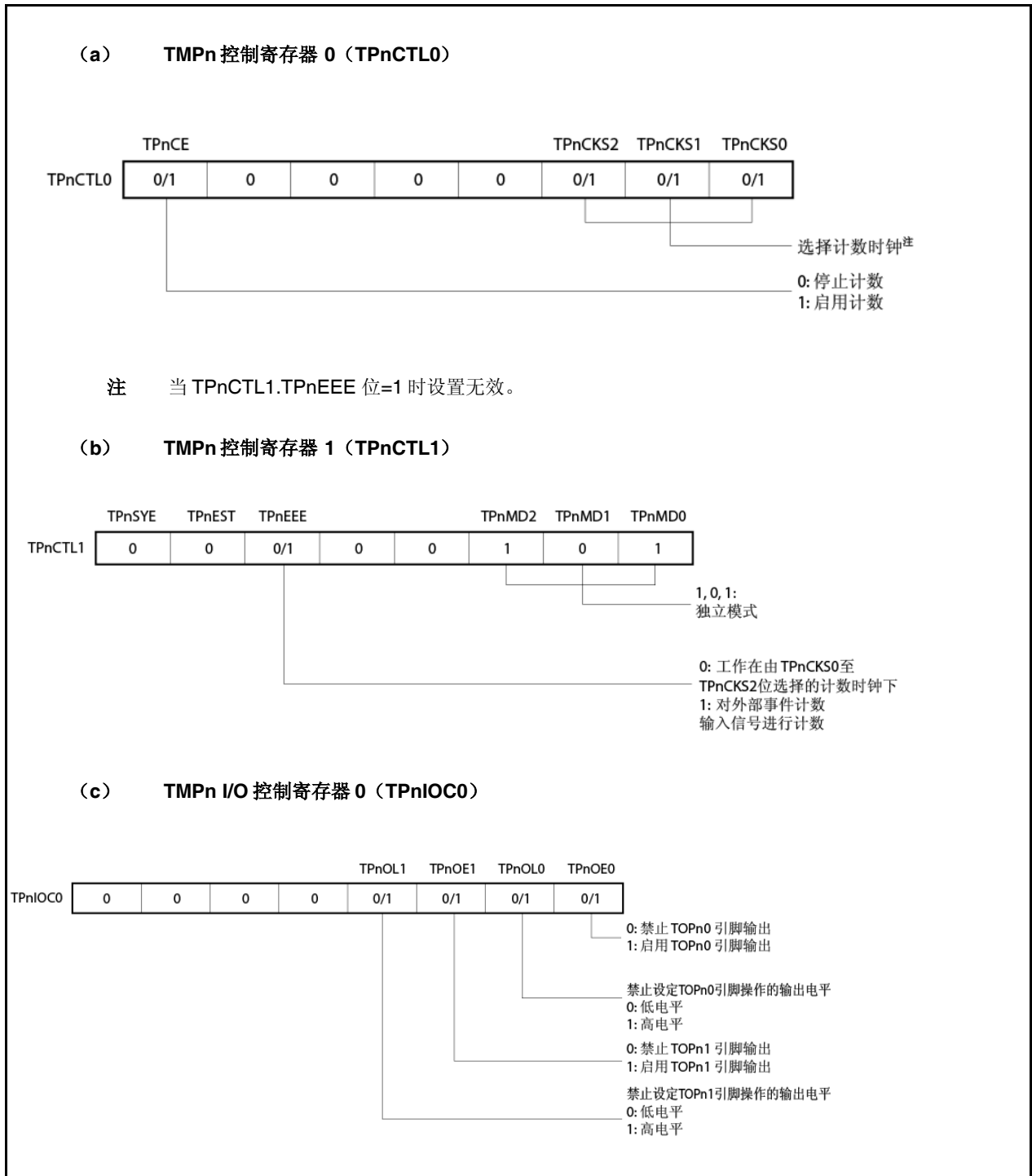
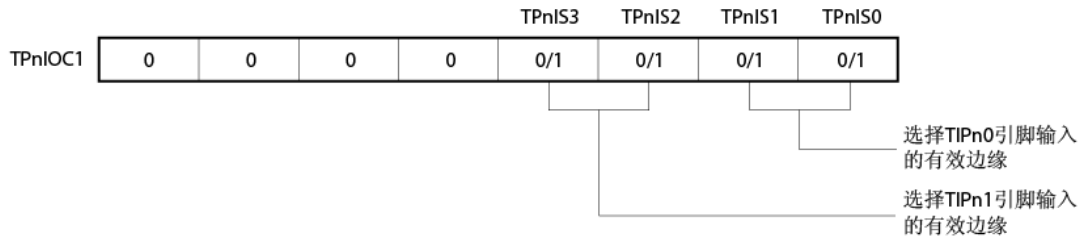
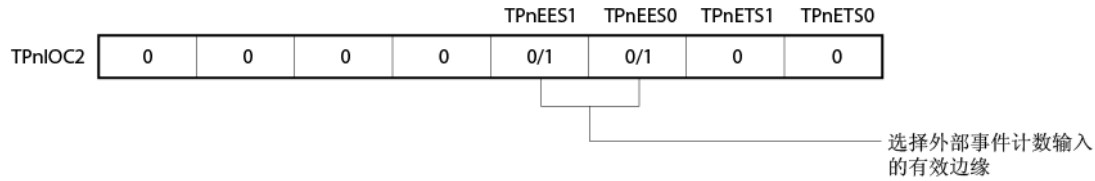


图 6-31 独立定时器模式的寄存器设置 (2/2)

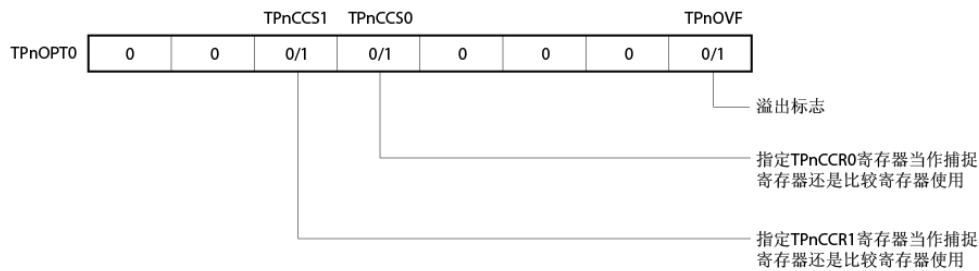
## (d) TMPn I/O 控制寄存器 1 (TPnIOC1)



## (e) TMPn I/O 控制寄存器 2 (TPnIOC2)



## (f) TMPn 选项寄存器 0 (TPnOPT0)



## (g) TMPn 计数器读取缓冲寄存器 (TPnCNT)

通过读取 TPnCNT 寄存器可读取 16 位计数器的值。

## (h) TMPn 捕捉/比较寄存器 0、1 (TPnCCR0、TPnCCR1)

根据 TPnOPT0.TPnCCSm 位的设定，这些寄存器用作捕捉寄存器或比较寄存器。

当用作捕捉寄存器时，这些寄存器在检测到有效边缘输入到 TIPnm 引脚时存储 16 位计数器的计数值。

当这些寄存器用作比较寄存器且将 D<sub>m</sub> 设置给 TPnCCRm 寄存器时，INTTPnCCm 信号在计数器达到 (D<sub>m</sub>+1) 时生成，且 TOPnm 引脚的输出信号反相。

备注      n=0 到 3  
m=0、1

（1）独立定时器模式的工作流程

（a） 将捕捉/比较寄存器用作比较寄存器时

图 6-32 独立定时器模式（比较功能）软件处理流程（1/2）

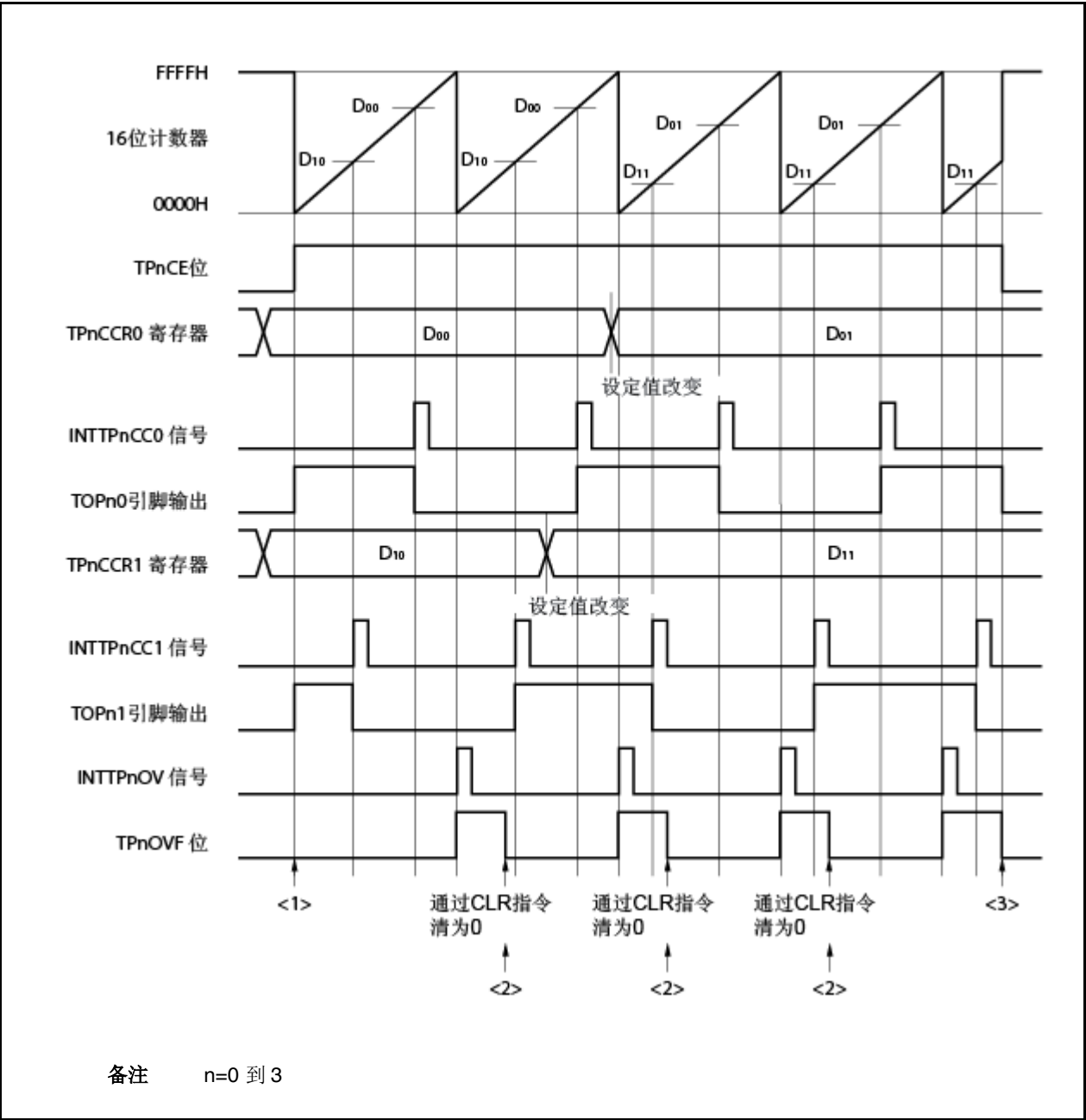
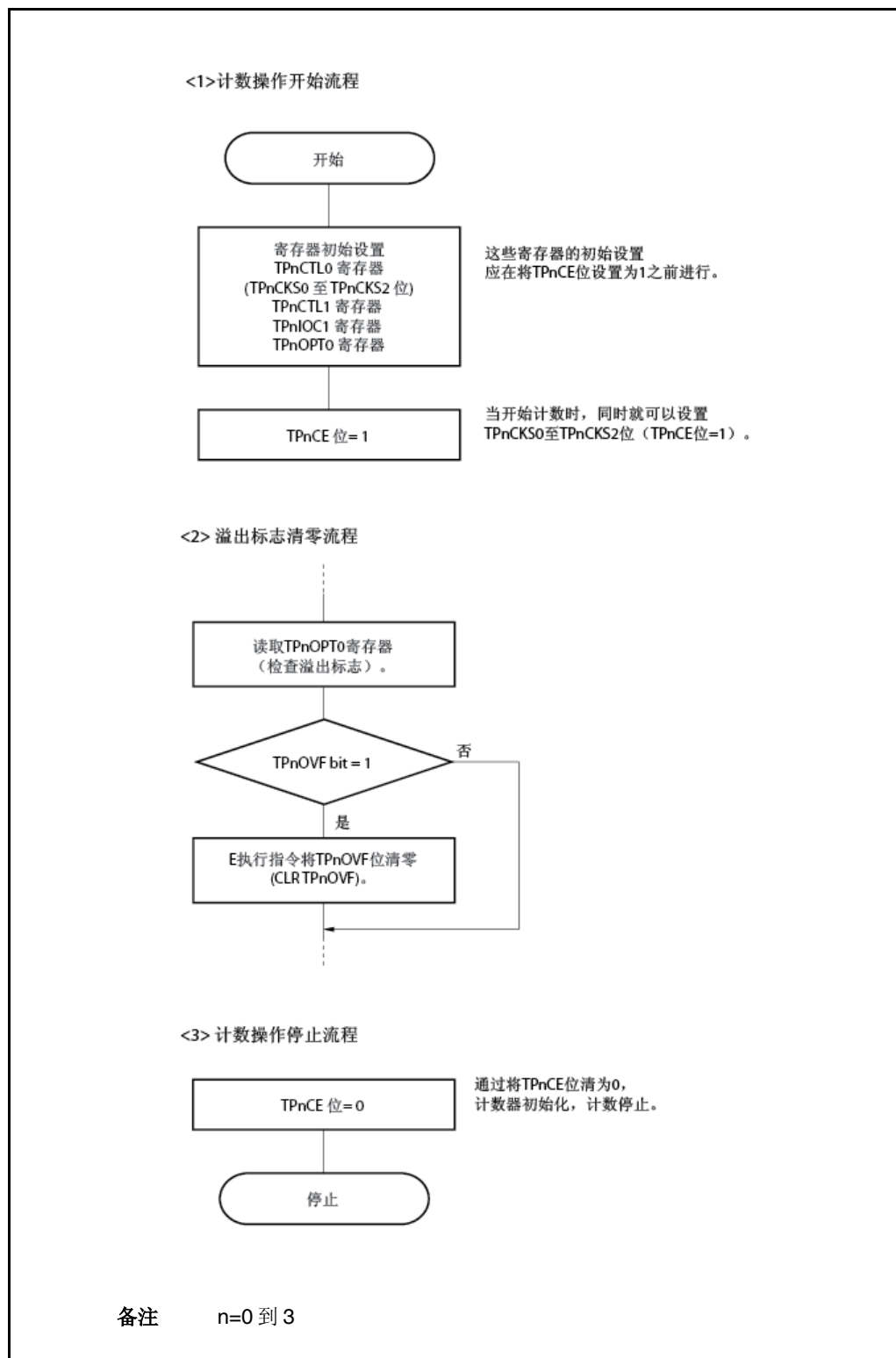


图 6-32 独立定时器模式（比较功能）软件处理流程（2/2）



(b) 将捕捉/比较寄存器用作捕捉寄存器时

图 6-33 独立定时器模式（捕捉功能）软件处理流程（1/2）

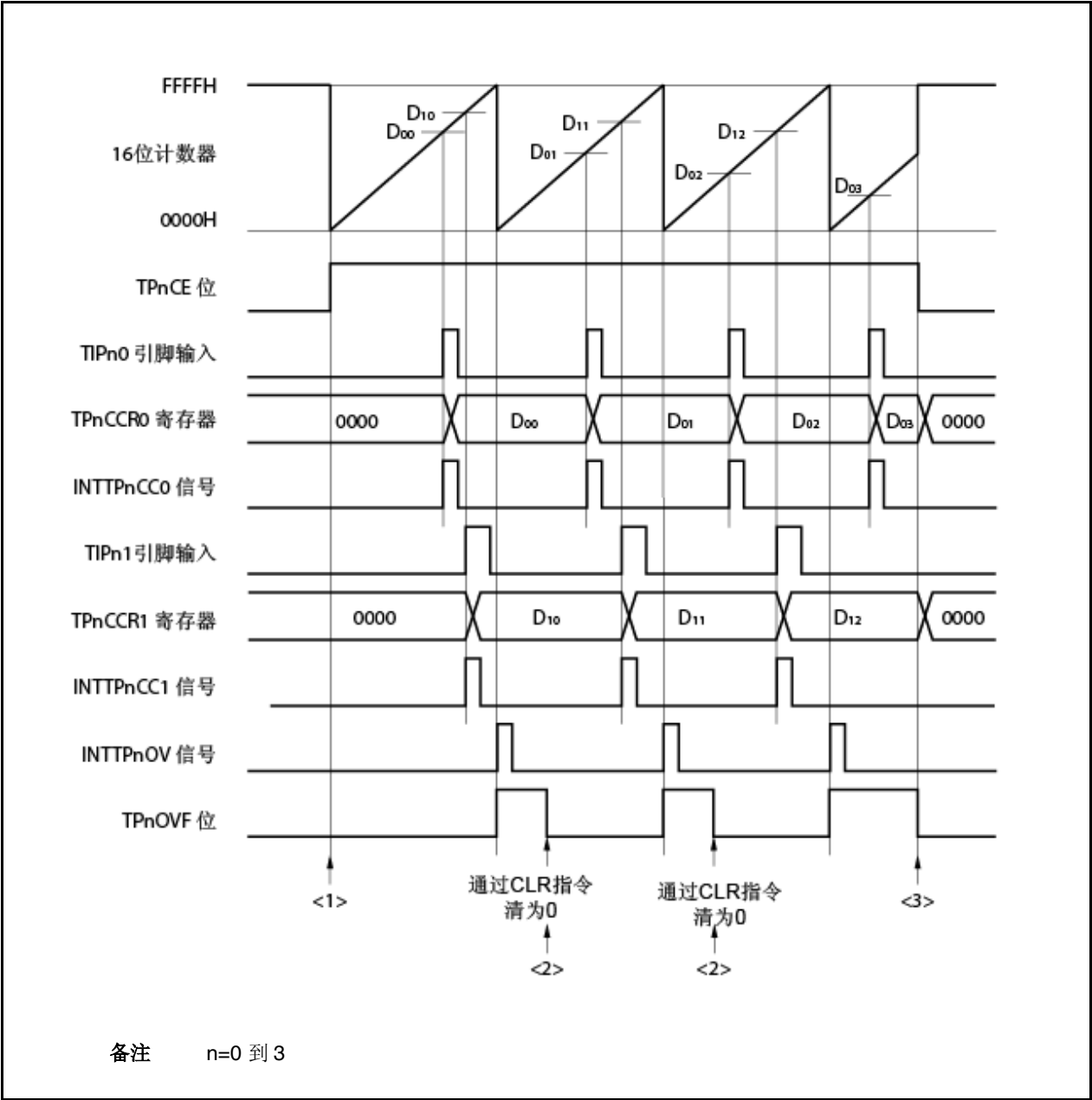
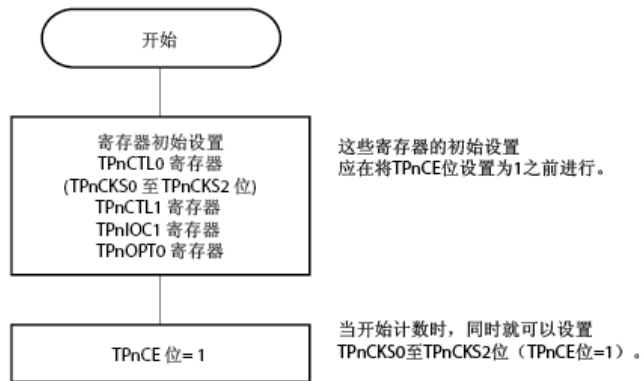
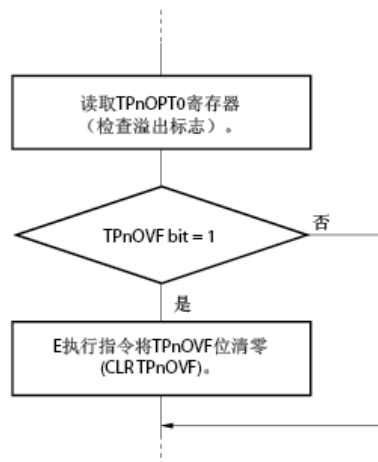


图 6-33 独立定时器模式（捕捉功能）软件处理流程（2/2）

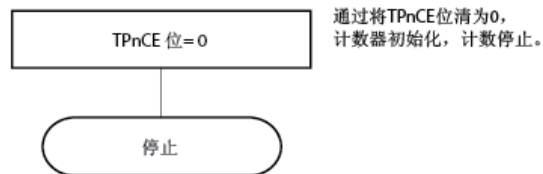
## &lt;1&gt; 计数操作开始流程



## &lt;2&gt; 溢出标志清零流程



## &lt;3&gt; 计数操作停止流程

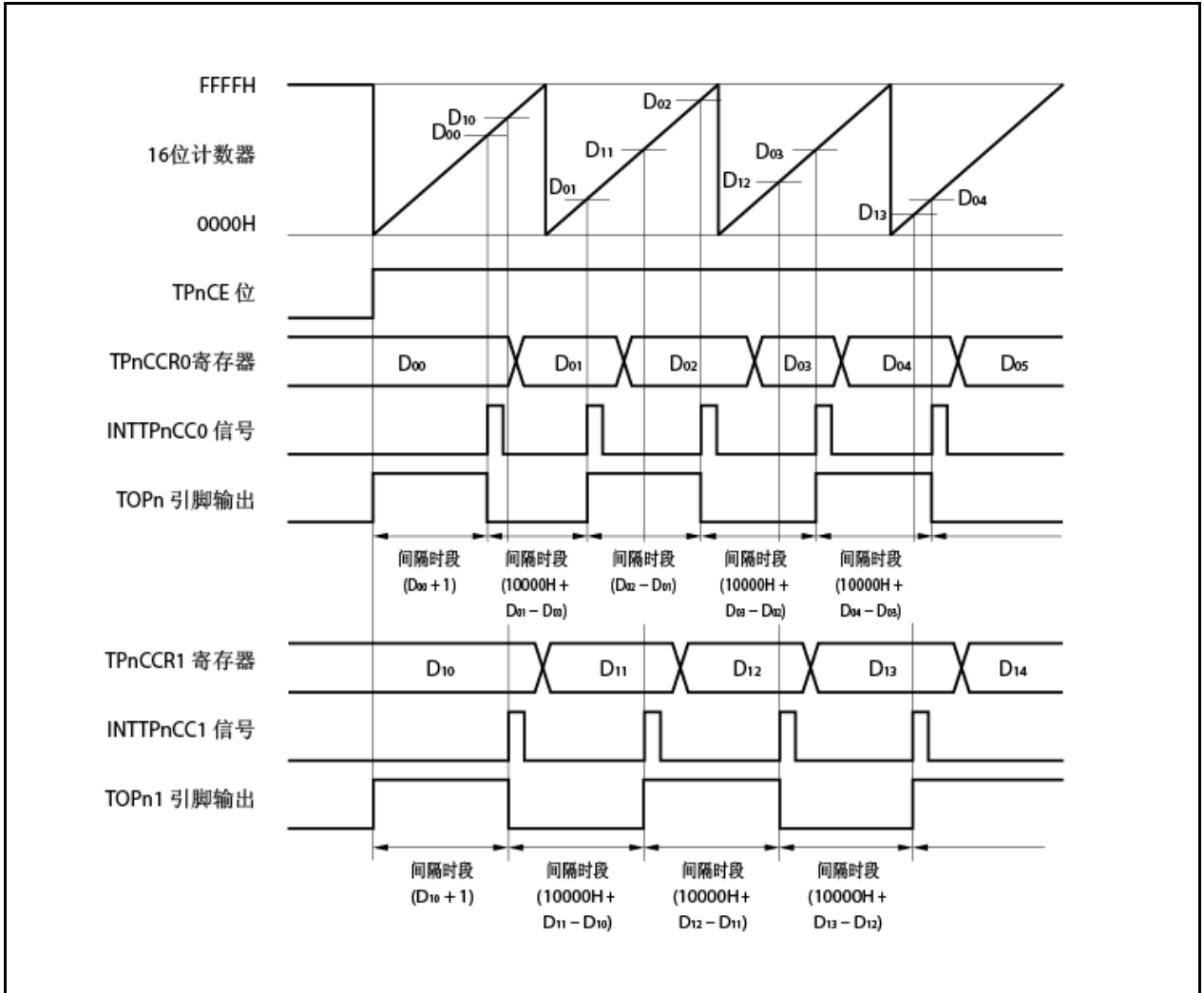


备注 n=0 到 3

## (2) 独立定时器模式的操作定时

## (a) 比较寄存器的时间间隔操作

在将 16 位定时器/事件计数器 P 用作时间间隔定时器、将 TPnCCRm 寄存器用作比较寄存器时，每次检测到 INTTPnCCm 信号时都必须用软件处理设置比较值才能产生下一个中断请求信号。



在独立定时器模式下进行时间间隔操作时，一个通道内可以设置两个时间间隔。

要进行时间间隔操作，在检测到 INTTPnCCm 信号时执行的中断服务中必须重新设置对应的 TPnCCRm 寄存器的值。

重新设置 TPnCCRm 寄存器的设置值可用下式计算，其中“D<sub>m</sub>”为间隔时段。

比较寄存器默认值：D<sub>m</sub>-1

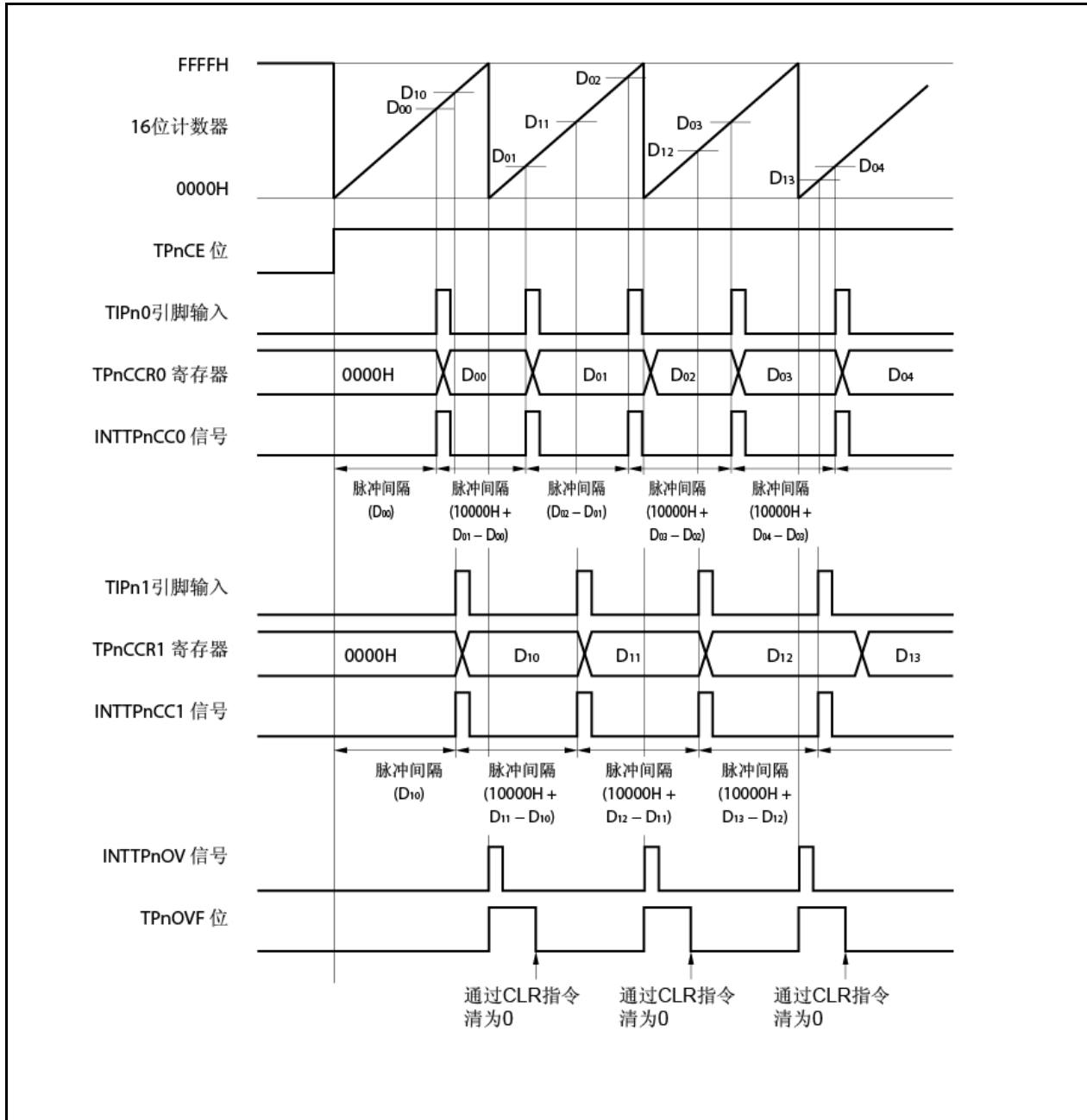
第二次及之后设置给比较寄存器的值：先前设置值+D<sub>m</sub>

（如果计算结果大于 FFFFH，应从结果中减去 10000H 再将该值设置给寄存器。）

备注      n=0 到 3  
             m=0、1

## (b) 捕捉寄存器的脉冲宽度测量

用 TPnCCRm 寄存器作为捕捉寄存器进行脉冲宽度测量时，每次检测到 INTTPnCCm 信号进行捕捉寄存器读取及计算时间间隔时必须进行软件处理。



在独立定时器模式下执行脉冲宽度测量时，一个通道内可以测量两个脉冲宽度。

要测量脉冲宽度时，其计算方法是先在 INTTPnCCm 信号同步下读取 TPnCCRm 寄存器的值，再计算读取值与先前读取值之前的差。

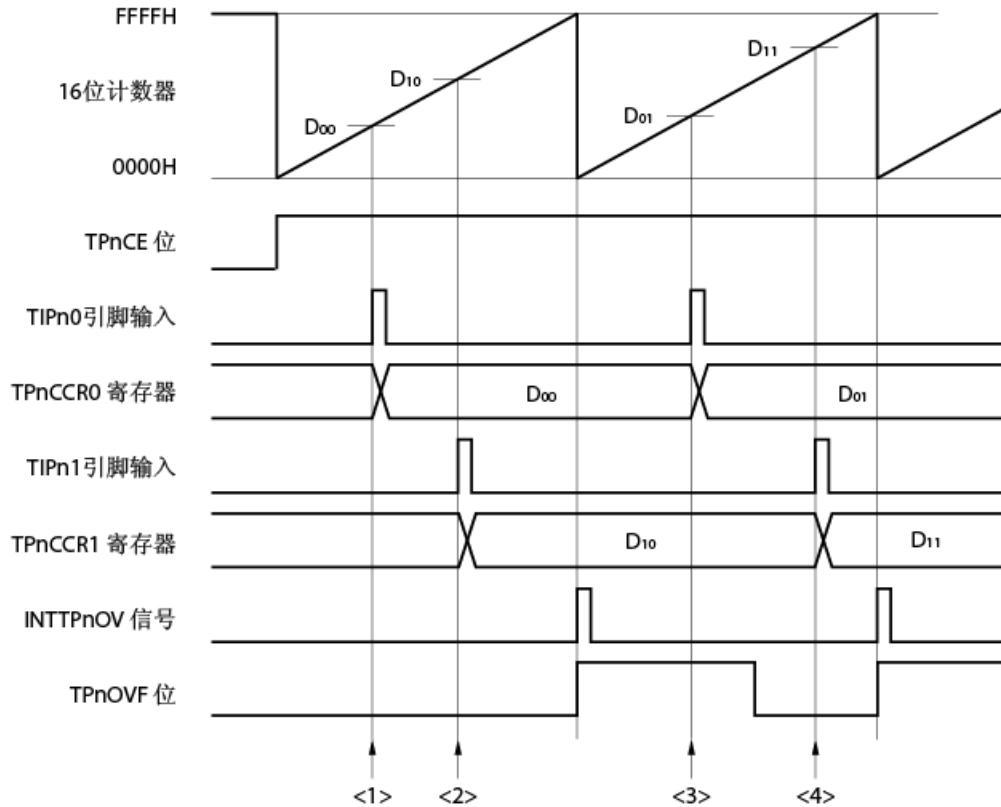
备注 n=0 到 3  
m=0、1



## (c) 使用两个捕捉寄存器时溢出的处理

在使用两个捕捉寄存器时，必须小心处理溢出标志。首先，不正确处理的例子如下所示。

在使用两个捕捉寄存器时不正确处理示例



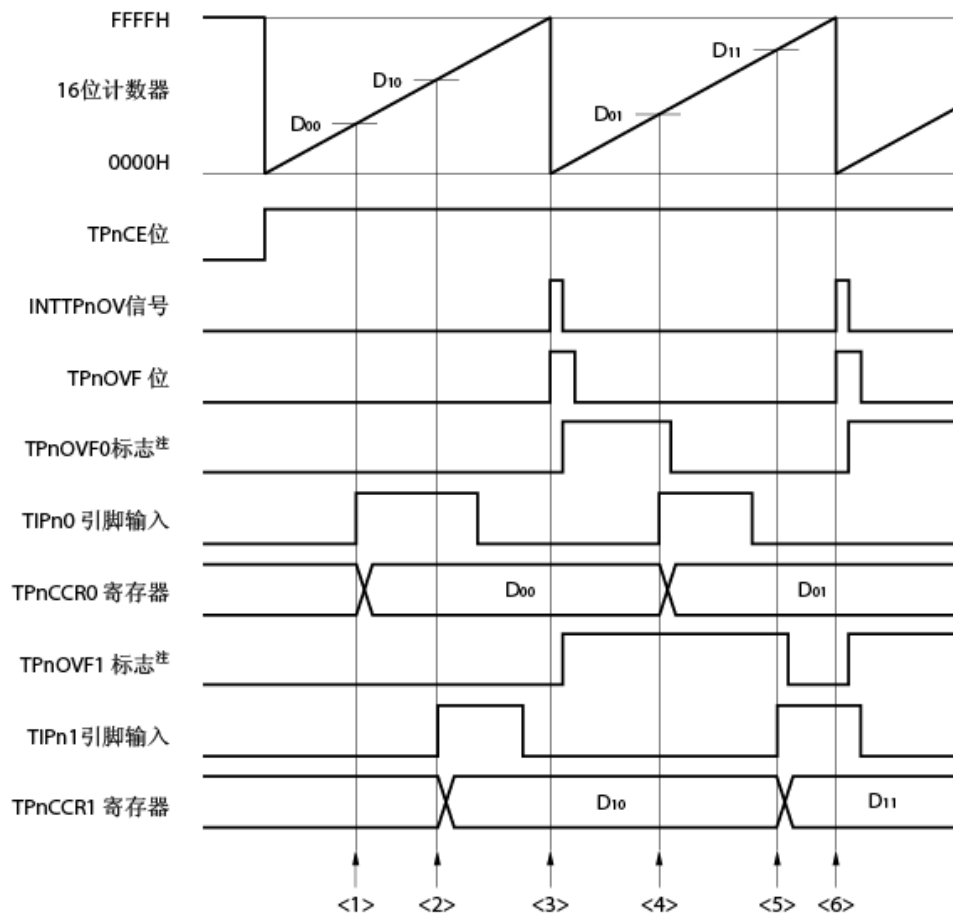
在独立定时器模式下测量两个脉冲宽度时可能会出现下列问题。

- <1> 读取 TPnCCR0 寄存器（设置 TIPn0 引脚输入的默认值）。
- <2> 读取 TPnCCR1 寄存器（设置 TIPn1 引脚输入的默认值）。
- <3> 读取 TPnCCR0 寄存器。  
读取溢出标志。如果溢出标志为 1，将其清为 0。  
因为溢出标志为 1，所以脉冲宽度可用  $(10000H + D_{01} - D_{00})$  计算。
- <4> 读取 TPnCCR1 寄存器。  
读取溢出标志。因为标志在<3>中清零，所以读取 0。  
因为溢出标志为 0，所以脉冲宽度可用  $(D_{11} - D_{10})$  计算（不正确）。

在使用两个捕捉寄存器时，如果溢出标志由一个捕捉寄存器清为 0，则另一个捕捉寄存器就可能不会得到正确的脉冲宽度。

在使用两个捕捉寄存器时应使用软件。关于如何使用软件的例子如下所示。

使用两个捕捉寄存器的示例（使用溢出中断）



注 TPnovf0 和 TPnOVF1 标志由软件在内部 RAM 中设置。

<1> 读取 TPnCCR0 寄存器（设置 TIPn0 引脚输入的默认值）。

<2> 读取 TPnCCR1 寄存器（设置 TIPn1 引脚输入的默认值）。

<3> 出现溢出。在溢出中断服务中将 TPnOVF0 和 TPnOVF1 标志设置为 1，并将溢出标志清为 0。

<4> 读取 TPnCCR0 寄存器。

读取 TPnOVF0 标志。如果 TPnOVF0 标志为 1，应将其清为 0。

因为 TPnOVF0 标志为 1，所以脉冲宽度可用  $(10000H + D_{01} - D_{00})$  计算。

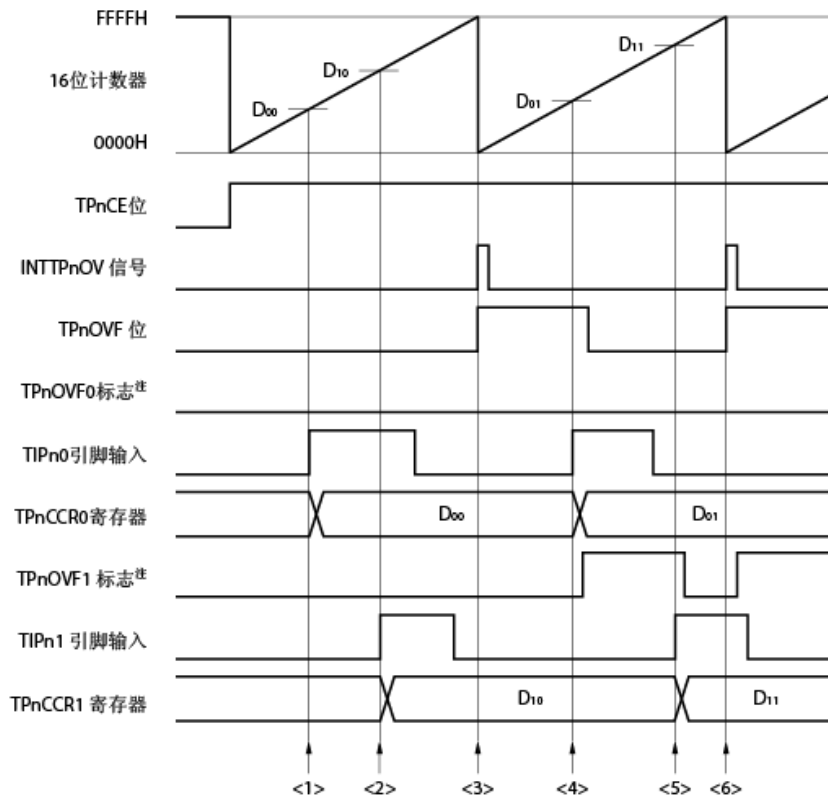
<5> 读取 TPnCCR1 寄存器。

读取 TPnOVF1 标志。如果 TPnOVF1 标志为 1，应将其清为 0（TPnOVF0 标志在<4>中清零，TPnOVF1 标志仍然是 1）。

因为 TPnOVF1 标志为 1，所以脉冲宽度可用  $(10000H + D_{11} - D_{10})$  计算（正确）。

<6> 同<3>

使用两个捕捉寄存器的示例（不使用溢出中断）



注 TPnovf0 和 TPnOVF1 标志由软件在内部 RAM 中设置。

<1> 读取 TPnCCR0 寄存器（设置 TIPn0 引脚输入的默认值）。

<2> 读取 TPnCCR1 寄存器（设置 TIPn1 引脚输入的默认值）。

<3> 出现溢出。软件无操作。

<4> 读取 TPnCCR0 寄存器。

读取溢出标志。如果溢出标志为 1，应只将 TPnOVF1 标志设置为 1 并将溢出标志清为 0。

因为溢出标志为 1，所以脉冲宽度可用  $(10000H + D_{01} - D_{00})$  计算。

<5> 读取 TPnCCR1 寄存器。

读取溢出标志。因为溢出标志在<4>中清零，所以读取 0。

读取 TPnOVF1 标志。如果 TPnOVF1 标志为 1，应将其清为 0。

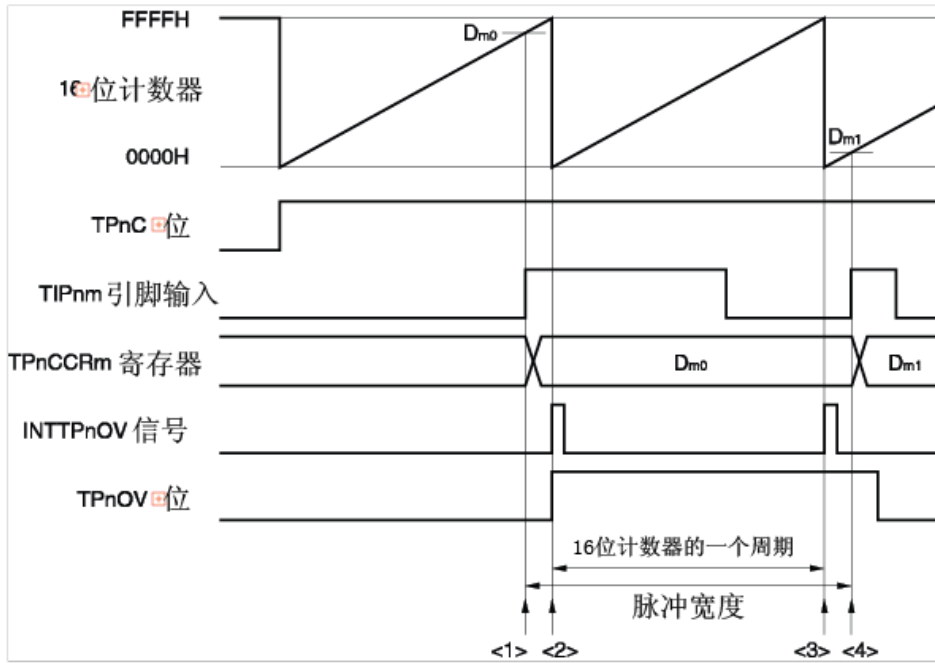
因为 TPnOVF1 标志为 1，所以脉冲宽度可用  $(10000H + D_{11} - D_{10})$  计算（正确）。

<6> 同<3>

## (d) 长捕捉触发间隔下的溢出处理

当脉冲宽度大于 16 位计数器的一个周期时必须小心处理，因为从第一个到下一个捕捉触发之间可能会出现不止一次溢出。首先，不正确处理的例子如下所示。

长捕捉触发间隔下不正确处理示例



在独立定时器模式下测量长脉冲宽度时可能会出现下面的问题。

<1> 读取 TPnCCRM 寄存器（设置 TIPnm 引脚输入的默认值）。

<2> 出现溢出。软件无操作。

<3> 再次出现溢出。软件无操作。

<4> 读取 TPnCCRM 寄存器。

读取溢出标志。如果溢出标志为 1，将其清为 0。

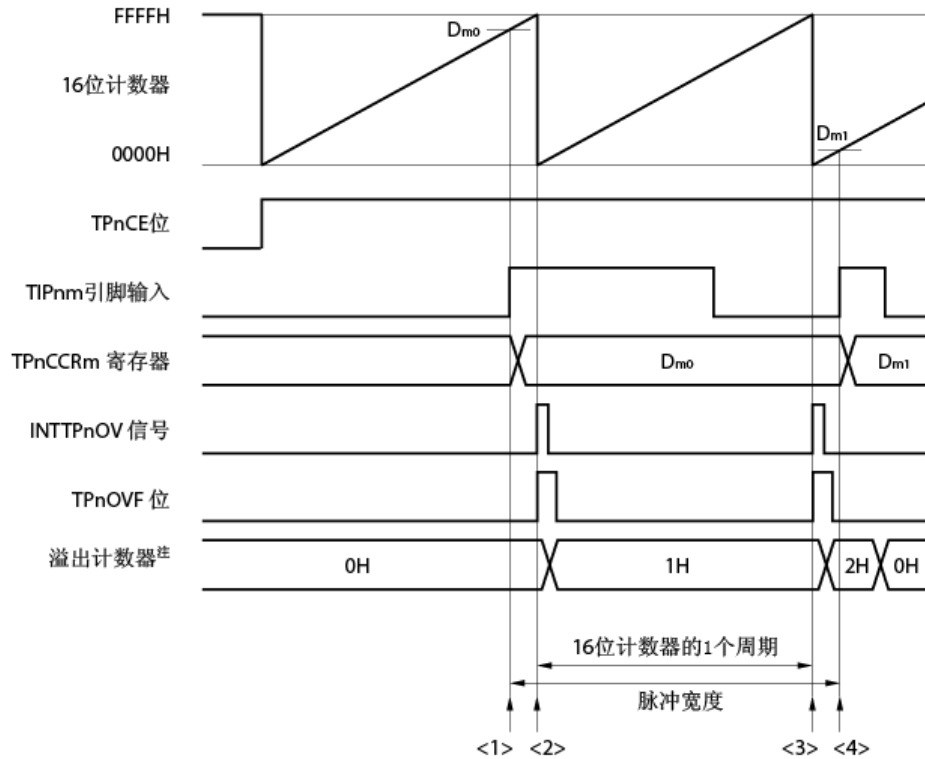
因为溢出标志为 1，所以脉冲宽度可用  $(10000H + D_{m1} - D_{m0})$  计算（不正确）。

实际上，因为溢出出现了两次，所以脉冲宽度一定是  $(20000H + D_{m1} - D_{m0})$ 。

在长捕捉触发间隔下如果出现两次（或更多）溢出，应可能得不到正确的脉冲宽度。

如果是长捕捉触发间隔，应减慢计数时钟以延长 16 位计数器的一个周期，或者使用软件。关于如何使用软件例子如下所示。

长捕捉触发间隔的示例



注 溢出计数器通过软件在内部 RAM 中强制设置。

<1> 读取 TPnCCRm 寄存器（设置 TIPnm 引脚输入的默认值）。

<2> 出现溢出。在溢出中断服务中增加溢出计数器的值并将溢出标志清为 0。

<3> 再次出现溢出。在溢出中断服务中增加 (+1) 溢出计数器的值并将溢出标志清为 0。

<4> 读取 TPnCCRm 寄存器。

读取溢出计数器。

→当溢出计数器为“N”时，脉冲宽度可用  $(N \times 10000H + D_{m1} - D_{m0})$  计算。

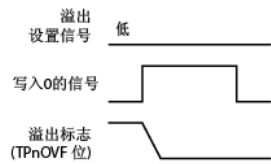
在本例中，因为溢出出现了两次，所以脉冲宽度为  $(20000H + D_{m1} - D_{m0})$ 。

将溢出计数器清零 (0H)。

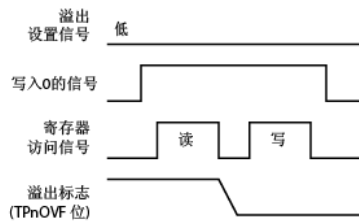
(e) 将溢出标志清零

将溢出标志清为 0 的方法是先使用 CLR 指令将 TPnOVF 位清为 0，再将 8 位数据（0 位是 0）写入 TPnOPT0 寄存器。要准确检测溢出，应在 TPnOVF 位为 1 时读取该位，然后通过操作指令将溢出标志清零。

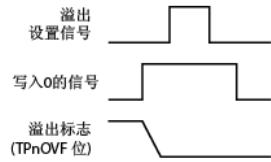
(i) 写入 0 的操作 (与设置无冲突)



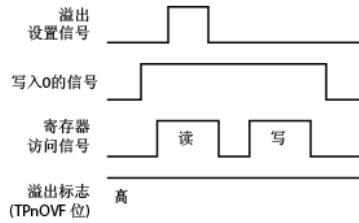
(iii) 清为 0 的操作 (与设置无冲突)



(ii) 写入 0 的操作 (与设置冲突)



(iv) 清为 0 的操作 (与设置冲突)



备注 n=0 到 3

要将溢出标志清为 0，应读取溢出标志来检验是否设置为 1 并用 CLR 指令将其清零。如果在没有检验溢出标志是否为 1 的情况下将 0 写入该标志，则溢出的设置信息可能会因为写入 0 而删除（上图 (ii)）。因此，即便实际出现了溢出，软件也可能判断为无溢出。

如果在利用 CLR 指令将溢出标志清为 0 时 CLR 指令的执行与溢出产生相冲突，则即使在执行清零指令后溢出标志仍然保持设置状态。

### 6.5.7 脉冲宽度测量模式 (TPnMD2 到 TPnMD0 位=110)

在脉冲宽度测量模式下，当TPnCTL0.TPnCE 位设置为 1 时 16 位定时器/事件计数器 P 开始计数。每次检测到输入TIPnm引脚的有效边缘时，16 位计数器的计数值就会存储在 TPnCCRm 寄存器中且 16 位计数器清零为 0000H。

通过在出现捕捉中断请求信号（INTTPnCCm）后读取 TPnCCRm 寄存器，可以测量有效边缘的时间间隔。

可将 TIPn0 或 TIPn1 引脚选为捕捉触发输入引脚。对于未使用的引脚，应用 TPnIOC1 寄存器指定为“无边缘检测”。

当使用外部时钟作为计数时钟时，因为外部时钟固定到 **TIPn0** 引脚，所以应测量 **TIPn1** 引脚的脉冲宽度。此时，应将 **TPnIOC1.TPnIS1** 和 **TPnIOC1.TPnIS0** 位清零为 00（捕捉触发输入（**TIPn0** 引脚）：无边缘检测）。

图 6-34 脉冲宽度测量模式的配置

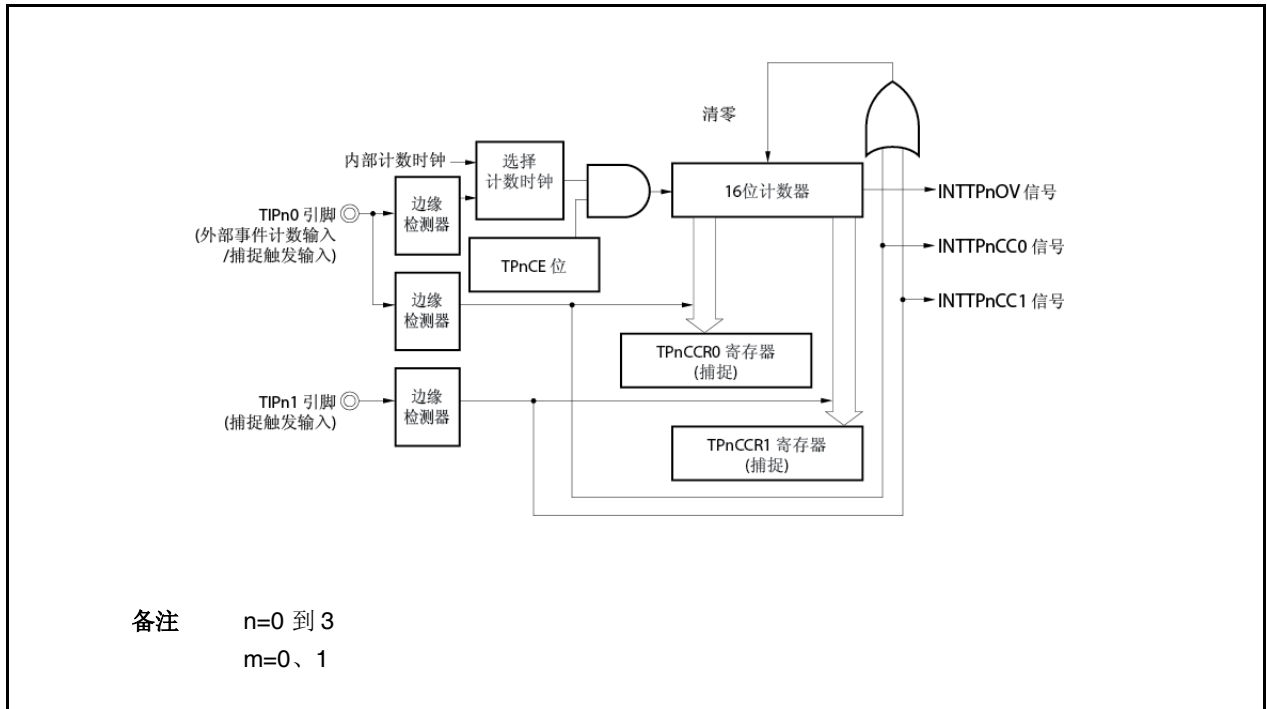
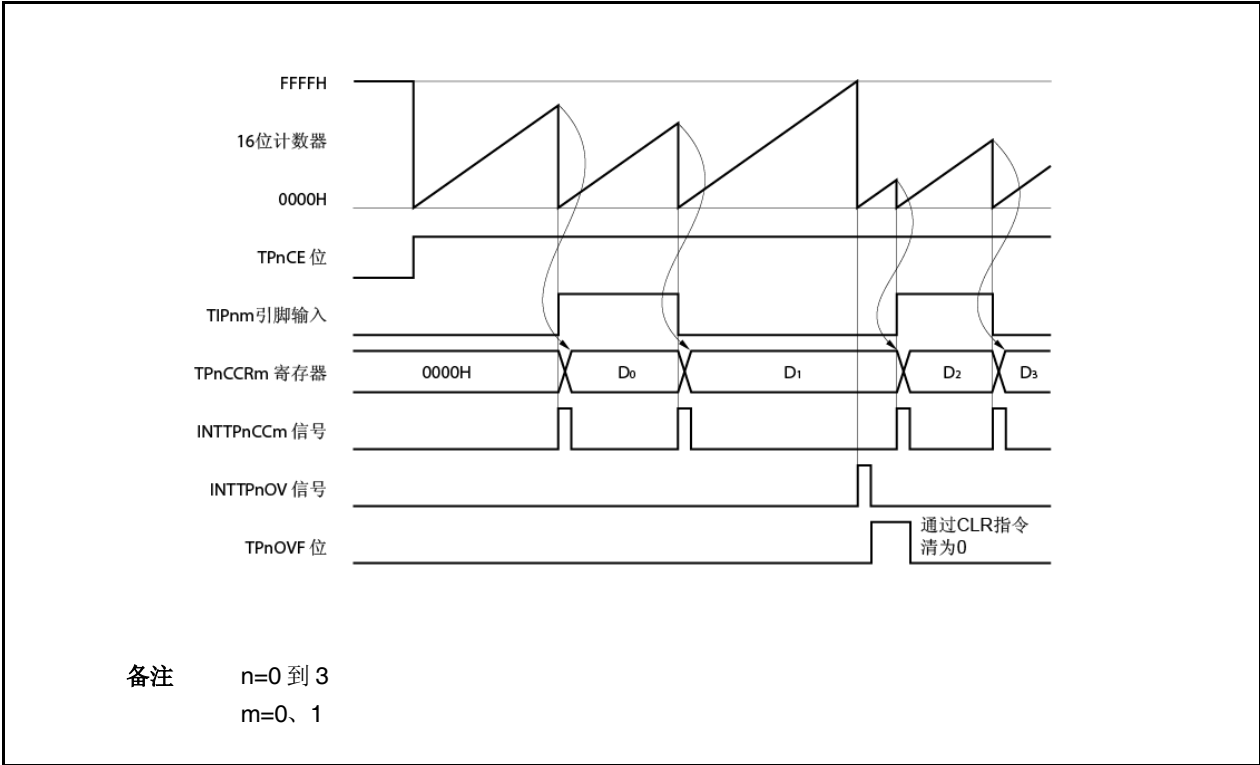


图 6-35 脉冲宽度测量模式的基本定时



当 TPnCE 位设置为 1 时，16 位计数器开始计数。当在此之后检测到输入到 TIPnm 引脚的有效边缘时，16 位计数器的计数值会存储在 TPnCCRm 寄存器中，16 位计数器清零为 0000H，生成捕捉中断请求信号（INTTPnCCm）。脉冲宽度计算如下。

脉冲宽度=捕捉值×计数时钟周期

如果当 16 位计数器向上计数到 FFFFH 时有效边缘还没有输入 TIPnm 引脚，则下一个计数时钟会产生溢出中断请求信号（INTTPnOV），且计数器清零为 0000H 并继续计数。此时，溢出标志（TPnOPT0.TPnOVF 位）也会设置为 1。应通过软件执行 CLR 指令将溢出标志清为 0。

如果溢出标志设置为 1，脉冲宽度可计算如下。

脉冲宽度=（10000H×TPnOVF 位设置（1）计数+捕捉值）×计数时钟周期

**备注** n=0 到 3  
m=0、1



图 6-36 脉冲宽度测量模式下的寄存器设置 (1/2)

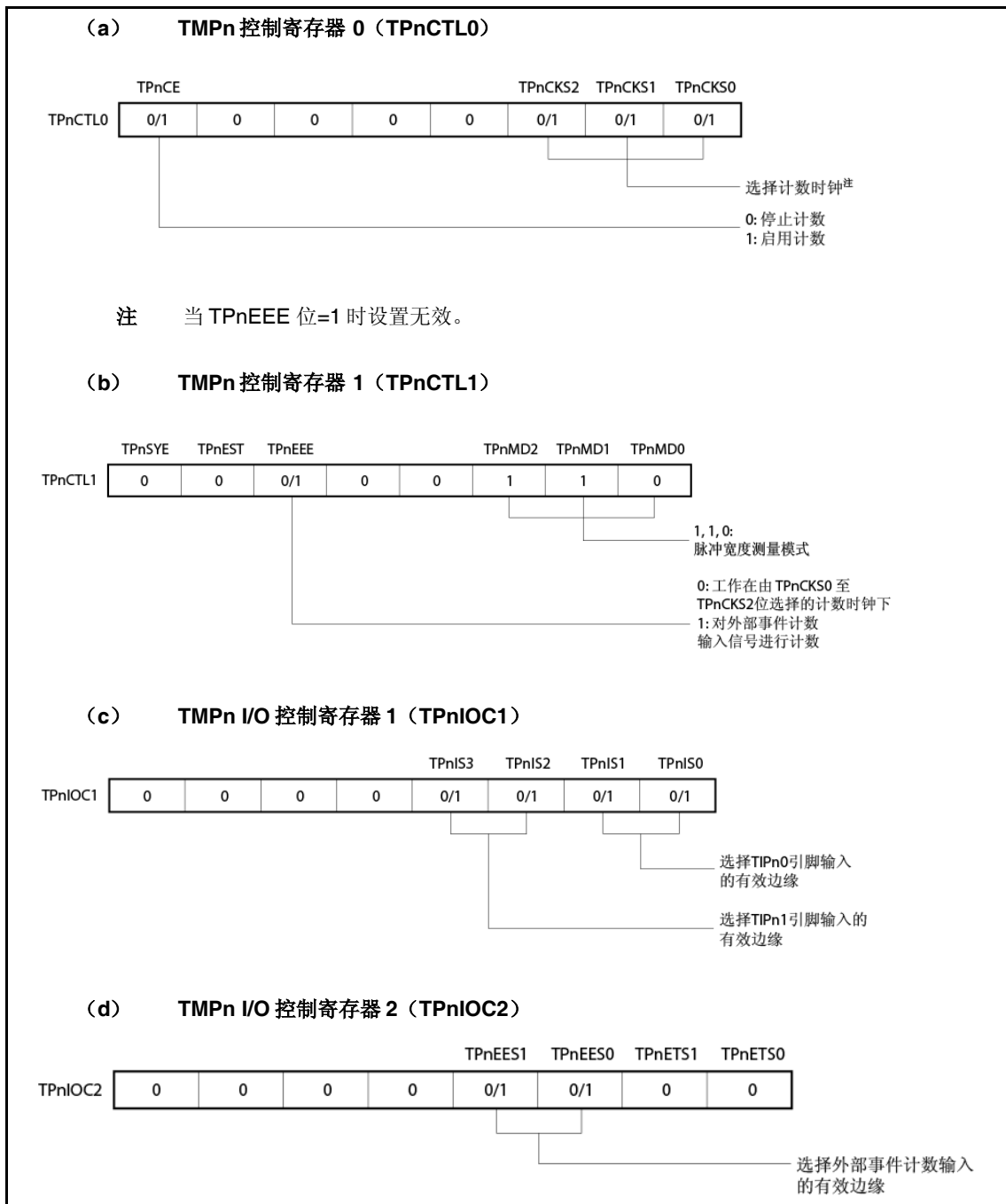
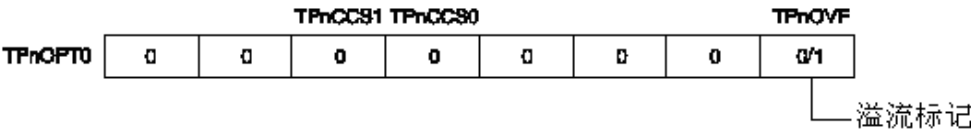


图 6-36 脉冲宽度测量模式下的寄存器设置 (2/2)

(e)      **TMPn 选项寄存器 0 (TPnOPT0)**



(f)      **TMPn 计数器读取缓冲寄存器 (TPnCNT)**

通过读取 TPnCNT 寄存器可读取 16 位计数器的值。

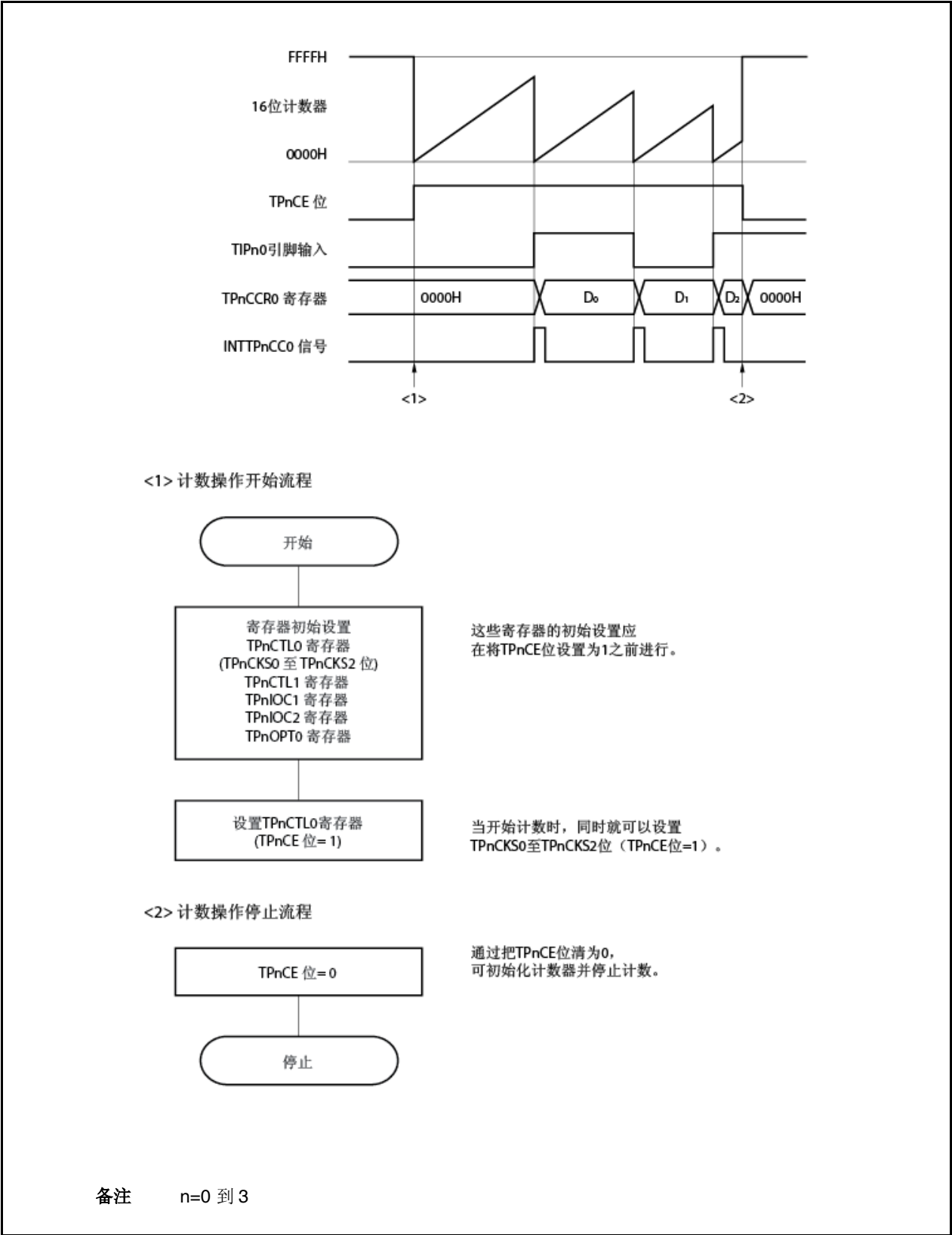
(g)      **TMPn 捕捉/比较寄存器 0、1 (TPnCCR0、TPnCCR1)**

当检测到输入 TIPnm 引脚的有效边缘时，这些寄存器存储 16 位计数器的计数值。

- 备注**
- 1. TMPn I/O 控制寄存器 0 (TPnIOC0) 在脉冲宽度测量模式下不使用。
  - 2. n=0 到 3  
m=0、1

（1）脉冲宽度测量模式下的工作流程

图 6-37 脉冲宽度测量模式下的软件处理流程

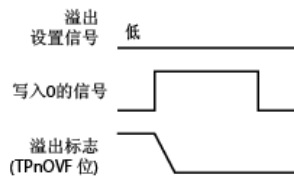


## (2) 脉冲宽度测量模式下的操作定时

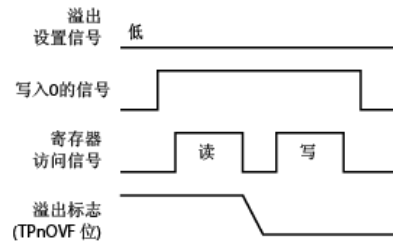
## (a) 将溢出标志清零

将溢出标志清为 0 的方法是先使用 CLR 指令将 TPnOVF 位清为 0，再将 8 位数据（0 位是 0）写入 TPnOPT0 寄存器。要准确检测溢出，应在 TPnOVF 位为 1 时读取该位，然后通过操作指令将溢出标志清零。

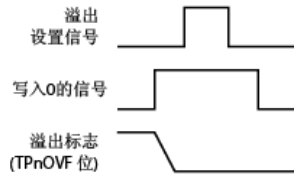
(i) 写入 0 的操作 (与设置无冲突)



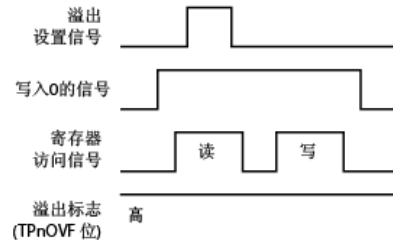
(iii) 清为 0 的操作 (与设置无冲突)



(ii) 写入 0 的操作 (与设置冲突)



(iv) 清为 0 的操作 (与设置冲突)



备注 n=0 到 3

要将溢出标志清为 0，应读取溢出标志来检验是否设置为 1 并用 CLR 指令将其清零。如果在没有检验溢出标志是否为 1 的情况下将 0 写入该标志，则溢出的设置信息可能会因为写入 0 而删除（上图 (ii)）。因此，即便实际出现了溢出，软件也可能判断为无溢出。

如果在利用 CLR 指令将溢出标志清为 0 时 CLR 指令的执行与溢出产生相冲突，则即使在执行清零指令后溢出标志仍然保持设置状态。

### 6.5.8 定时器输出操作

下表展示了 TOPn0 和 TOPn1 引脚的操作和输出电平。

表 6-4 各模式下的定时器输出控制

工作模式	TOPn1 引脚	TOPn0 引脚
时间间隔定时器模式	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
单次脉冲输出模式	单次脉冲输出	
PWM 输出模式	PWM 输出	
独立定时器模式	方波输出（仅当使用比较功能时）	
脉冲宽度测量模式	—	

备注 n=0 到 3

表 6-5 定时器输出控制位控制下 TOPn0 和 TOPn1 引脚的真值表

TPnIOC0.TPnOLm 位	TPnIOC0.TPnOEm 位	TPnCTL0.TPnCE 位	TOPnm 引脚的电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前瞬间为低电平，计数开始后为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前瞬间为高电平，计数开始后为低电平

备注 n=0 到 3  
m=0、1

## 6.6 定时器调谐工作功能

定时器 P 和定时器 Q 具有定时器调谐工作功能。

可进行同步的定时器列于表 6-6 中。

表 6-6 定时器的调谐工作模式

主定时器	从定时器	
TMP0	TMP1	—
TMP2	TMP3	TMQ0

**注意事项** 1. 调谐工作模式通过 TPmCTL1.TPmSYE 和 TQ0CTL1.TQ0SYE 位启用或停用。对于 TMP2 来说，可指定 TMP3 或 TMQ0 或两者作为从定时器。

2. 应按照下列步骤设置调谐工作模式。

<1> 设置从定时器的 TPmCTL1.TPmSYE 和 TQ0CTL1.TQ0SYE 位来启用调谐工作。

将从定时器的 TPmCTL1.TPmMD2 到 TPmCTL1.TPmMD0 和 TQ0CTL1.TQ0MD2 到 TQ0CTL1.TQ0MD 位设置为独立模式。

<2> 通过 TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0 位设置定时器模式。

此时，不要设置主定时器的 TPnCTL1.TPnSYE 位。

<3> 设置主从定时器的比较寄存器。

<4> 设置从定时器的 TPmCTL0.TPmCE 和 TQ0CTL0.TQ0CE 位以启用内部工作时钟下的运行。

<5> 设置主定时器的 TPnCTL0.TPnCE 位以启用内部工作时钟下的运行。

**备注** m=1、3  
n=0、2

表 6-7 和 6-8 展示了可在调谐工作模式下使用的定时器模式（√：可设置、×：不可设置）。

表 6-7 在调谐工作模式下可用的定时器模式

主定时器	独立模式	PWM 模式	三角波 PWM 模式
TMP0	√	√	×
TMP2	√	√	×

表 6-8 定时器输出功能

调谐通道	定时器	引脚	独立模式		PWM 模式		三角波 PWM 模式	
			调谐关	调谐开	调谐关	调谐开	调谐关	调谐开
Ch0	TMP0（主）	TOP00	PPG		切换		N/A	
		TOP01	PPG		PWM		N/A	
	TMP1（从）	TOP10	PGP		切换	PWM	N/A	
		TOP11	PPG		PWM		N/A	
Ch1	TMP2（主）	TOP20	PPG		切换		N/A	
		TOP21	PPG		PWM		N/A	
	TMP3（从）	TOP30	PPG		切换	PWM	N/A	
		TOP31	PPG		PWM		N/A	
	TMQ0（从）	TOQ00	PPG		切换	PWM	切换	N/A
		TOQ01 到 TOQ03	PPG		PWM		三角波 PWM	N/A

备注

从主定时器比较寄存器传输数据到从定时器比较寄存器的定时如下。

PPG:CPU 写定时

切换、PWM、三角波 PWM:定时器计数器和比较寄存器匹配 TOPn0 和 TOQ00 的定时（n=0 到 3）

图 6-38 调谐工作图 (TMP2、TMP3、TMQ0)

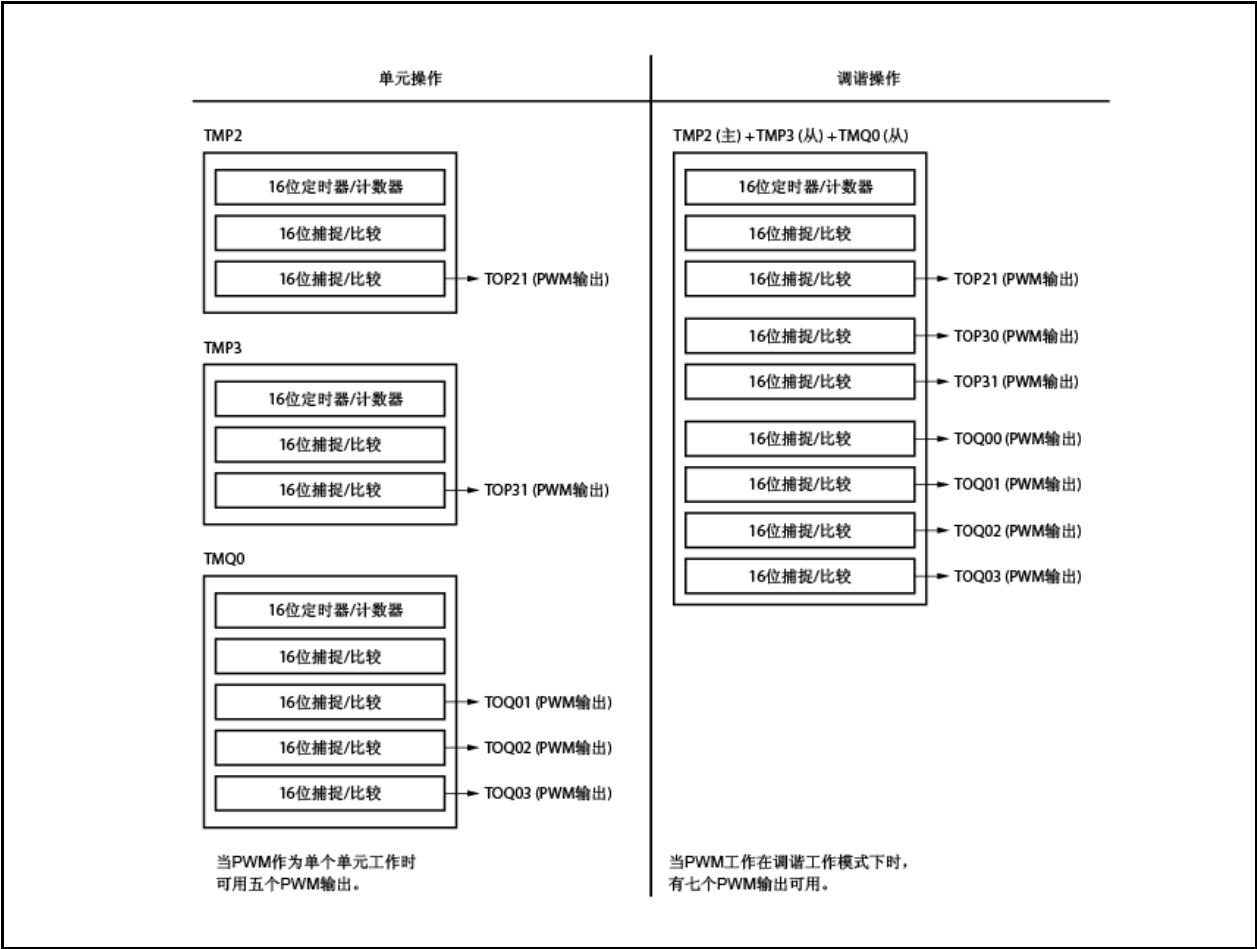
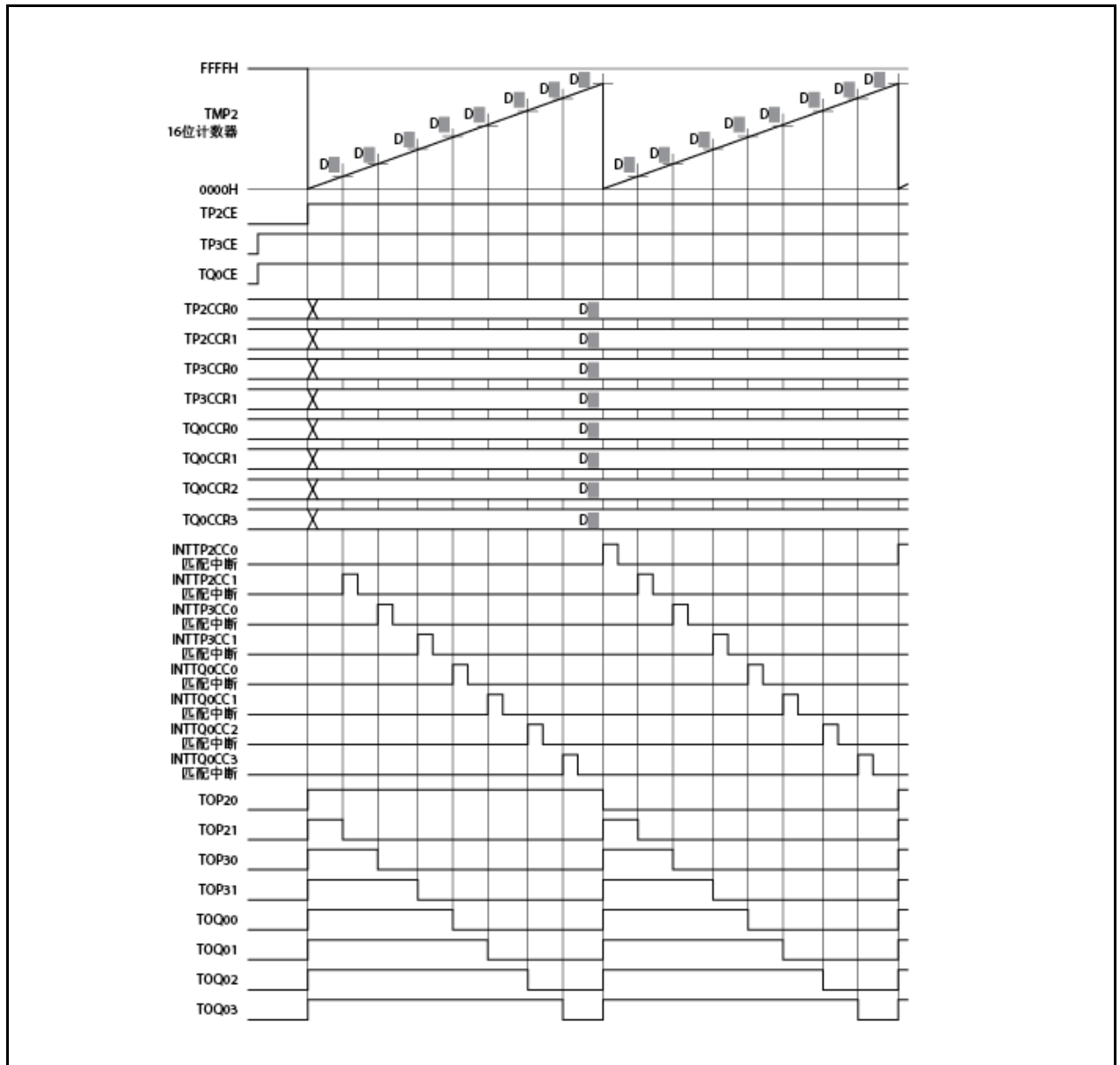




图 6-39 调谐 PWM 功能的基本操作定时 (TMP2、TMP3、TMQ0)



## 6.7 选择器功能

在 V850ES/HF2 中，端口的替换功能引脚和外围 I/O (TMP、TMM0 或 UARTA) 可用来选择 TMP 的捕捉触发输入。

利用此功能，可实现下列内容。

- 可从端口/定时器替换功能引脚 (TIP10 和 TIP11 引脚) 和 UARTA 接收替换功能引脚 (RXDA0 和 RXDA1) 选择 TMP1 的 TIP10、TIP11 输入信号。  
→ 当选择 UARTA0 或 UARTA1 的 RXDA0 或 RXDA1 信号时，可计算 UARTA LIN 接收传输的波特率误差。
- 可从端口/定时器替换功能引脚 (TIP01 引脚) 选择 TMP0 的 TIP01 输入信号，及 TMM0 的 INTTMM0EQ0 信号。

**注意事项 1.** 在使用选择器功能时，应在连接定时器前连接 TMP 的捕捉触发输入。

**2.** 在设置选择器功能时，应首先禁用要连接的外围 I/O (TMP、TMM0 或 UARTA)。

选择器功能的捕捉输入由下面的寄存器指定。

(1) 选择器操作控制寄存器 0（SELCNT0）

Selcnt0 寄存器为 8 位寄存器，用来为 TMP0 和 TMP1 选择捕捉触发。  
该寄存器可进行 8 位元或 1 位元的读取或写入。  
复位将此寄存器设置为 00H。

复位后: 00H    读/写    地址: FFFFF308H

	7	6	5	4	3	2	1	0
SELCNT0	0	0	0	ISEL04	ISEL03	ISEL02	0	0

ISEL04	选择TIP11 输入信号(TMP1)
0	TIP11引脚输入
1	RXDA1引脚输入

ISEL03	选择TIP10输入信号(TMP1)
0	TIP10引脚输入
1	RXDA0引脚输入

ISEL02 <sup>注</sup>	选择TIP01输入信号 (TMP0)
0	TIP01引脚输入
1	TMM0的INTTMM0EQ0中断

注     在下列情况下应用 INTTMM0EQ0 中断信号作为 TIP01 输入信号。  
TMM0 工作时钟≥TMP0 工作时钟×4

- 注意事项
1. 要将 ISEL02 到 ISEL04 位设置为 1，应在捕捉输入模式下设置对应的引脚。
  2. 应在目标（TMP0、TMP1、TMM0、UARTA0 或 UARTA1）停止工作时设置 ISEL02 到 ISEL06 位。

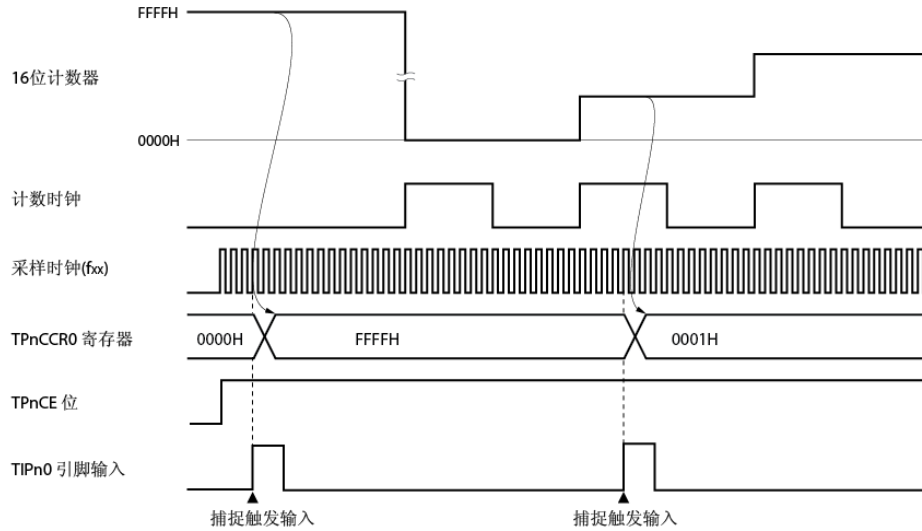
<R>

## 6.8 注意事项

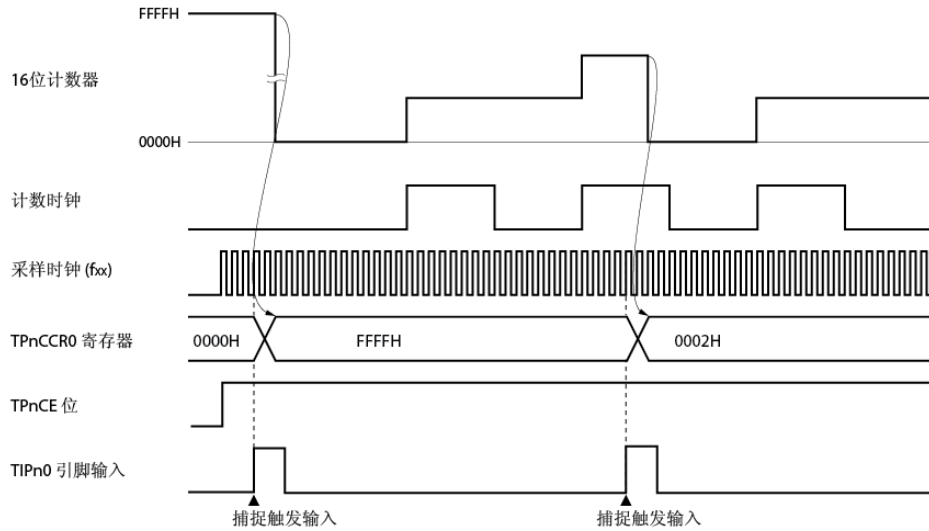
## (1) 捕捉操作

当使用捕捉操作且选择慢时钟作为计数时钟时，如果在 TPnCE 位设置为 1 后立刻输入捕捉触发，则在 TPnCCR0 和 TPnCCR1 寄存器中可能会捕捉到 FFFFH 而非 0000H。

## (a) 独立定时器模式



## (b) 脉冲宽度测量模式



## 第七章 16-位定时器/事件计数器 Q (TMQ)

定时器 Q(TMQ)为 16-位定时器/事件计数器。

V850ES/HF2 组合 TMQ0。

### 7.1 概述

TMQ0 的要点如下所示。

- 时钟选择：8 种
- 捕获/触发输入引脚：4
- 外部事件计数输入引脚：1
- 外部触发输入引脚：1
- 定时器/计数器：1
- 捕获/比较寄存器：4
- 捕获/比较匹配中断请求信号：4
- 定时器输出引脚：4

### 7.2 功能

TMQ0 具有以下功能。

- 计时器
- 外部事件计数器
- 外部触发脉冲输出
- 单触发脉冲输出
- PWM 输出
- 自由定时器
- 脉宽测量
- 三角形波 PWM 输出
- 定时器调试操作功能

7.3 配置

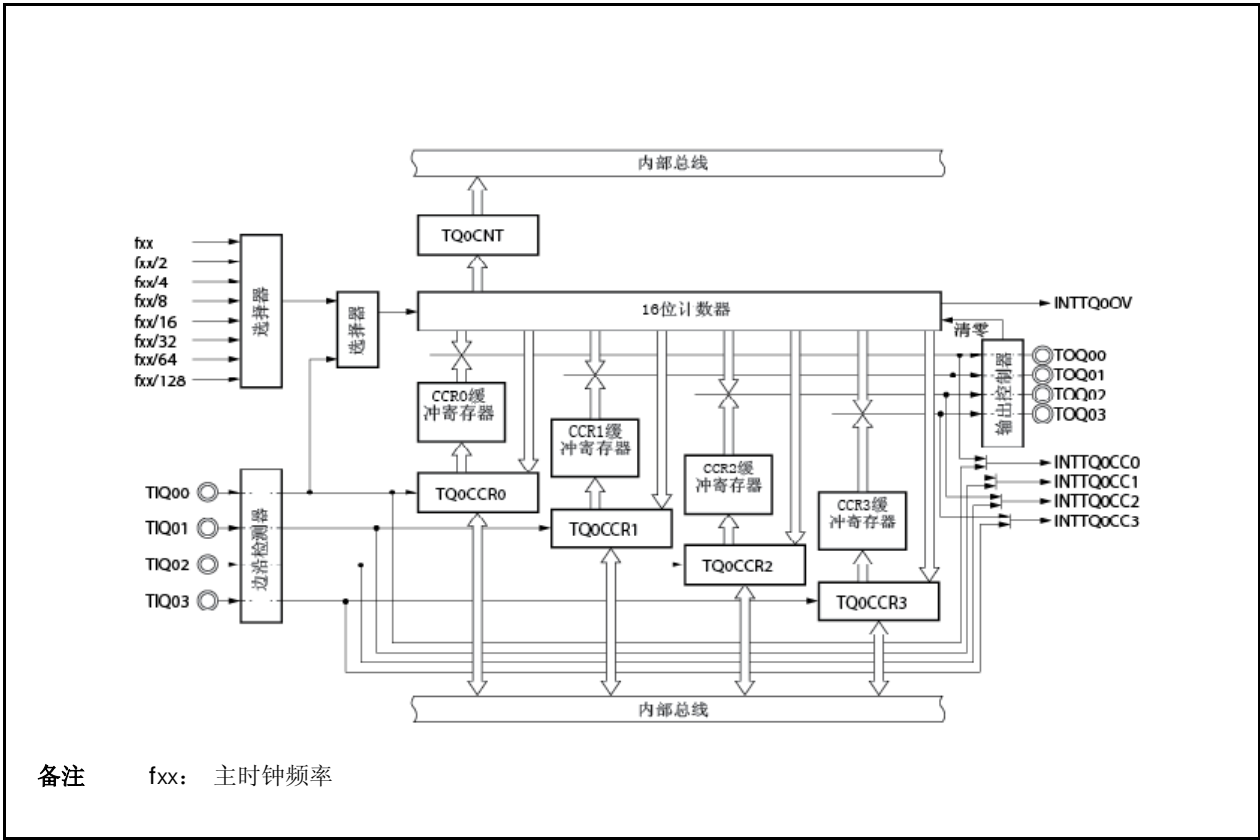
TMQ0 包括以下硬件。

表 7-1. TMQ0 配置

项目	配置
定时器寄存器	16-位计数器
寄存器	TMQ0 捕获/比较寄存器 0 到 3(TQ0CCR0 到 TQ0CCR3) TMQ0 计数器读取缓冲寄存器(TQ0CNT) CCR0 到 CCR3 缓冲寄存器
定时器输入	4(TIQ00 <sup>※1</sup> 到 TIQ03 引脚)
定时器输出	4(TOQ00 到 TOQ03 引脚)
控制寄存器 <sup>※2</sup>	TMQ0 控制寄存器 0, 1(TQ0CTL0, TQ0CTL1) TMQ0 I/O 控制寄存器 0 到 2(TQ0IOC0 到 TQ0IOC2) TMQ0 选项寄存器 0(TQ0OPT0)

- 注 1. TIQ00 引脚另外用作捕获触发输入信号，外部事件计数输入信号及外部触发输入信号。
2. 当使用 TIQ00 到 TIQ03 和 TOQ00 到 TOQ03 引脚的功能时，请参阅 表 4-14 端口引脚用作复用-功能引脚。

图 7-1. TMQ0 框图



**(1) 16-位计数器**

此 16-位计数器可对内部时钟或外部事件计数。

此计数器的计数值可使用 TQ0CNT 寄存器读取。

当 TQ0CTL0.TQ0CE 位= 0 时, 16-位计数器的值为 FFFFH。如果此时读取 TQ0CNT 寄存器, 则读取 0000H。

TQ0CE 位设为零。因此, 16-位计数器设为 FFFFH。

**(2) CCR0 缓冲寄存器**

这是比较 16-位计数器的计数值的 16-位比较寄存器。

当 TQ0CCR0 寄存器用作比较寄存器时, 写入 TQ0CCR0 寄存器的值转移到 CCR0 缓冲寄存器。

当 16-位计数器的计数值匹配 CCR0 缓冲寄存器的值时, 产生比较匹配中断请求信号(INTTQ0CC0)。

CCR0 缓冲寄存器不能直接读取或写入。

复位后 CCR0 缓冲寄存器清为 0000H, 而 TQ0CCR0 寄存器清为 0000H。

**(3) CCR1 缓冲寄存器**

这是比较 16-位计数器的计数值的 16-位比较寄存器。

当 TQ0CCR1 寄存器用作比较寄存器时, 写入 TQ0CCR1 寄存器的值转移到 CCR1 缓冲寄存器。

当 16-位计数器的计数值匹配 CCR1 缓冲寄存器的值时, 产生比较匹配中断请求信号(INTTQ0CC1)。

CCR1 缓冲寄存器不能直接读取或写入。

复位后 CCR1 缓冲寄存器清为 0000H, 而 TQ0CCR1 寄存器清为 0000H。

**(4) CCR2 缓冲寄存器**

这是比较 16-位计数器的计数值的 16-位比较寄存器。

当 TQ0CCR2 寄存器用作比较寄存器时, 写入 TQ0CCR2 寄存器的值转移到 CCR2 缓冲寄存器。

当 16-位计数器的计数值匹配 CCR2 缓冲寄存器的值时, 产生比较匹配中断请求信号(INTTQ0CC2)。

CCR2 缓冲寄存器不能直接读取或写入。

复位后 CCR2 缓冲寄存器清为 0000H, 而 TQ0CCR2 寄存器清为 0000H。

**(5) CCR3 缓冲寄存器**

这是比较 16-位计数器的计数值的 16-位比较寄存器。

当 TQ0CCR3 寄存器用作比较寄存器时, 写入 TQ0CCR3 寄存器的值转移到 CCR3 缓冲寄存器。

当 16-位计数器的计数值匹配 CCR3 缓冲寄存器的值时, 产生比较匹配中断请求信号(INTTQ0CC3)。

CCR3 缓冲寄存器不能直接读取或写入。

复位后 CCR3 缓冲寄存器清为 0000H, 而 TQ0CCR3 寄存器清为 0000H。

**(6) 边沿监测器**

此电路检测输入到 TIQ00 和 TIQ03 引脚的有效沿。无沿, 上升沿, 下降沿, 或上升沿和下降沿均可使用 TQ0IOC1 和 TQ0IOC2 寄存器选作有效沿。

**(7) 输出控制器**

此电路控制 TOQ00 到 TOQ03 引脚的输出。输出控制器由 TQ0IOC0 寄存器控制。

**(8) 选择器**

此选择器选择 16-位计数器的计数时钟。八类内部时钟或外部事件均可选作计数时钟。



## 7.4 寄存器

控制 TMQ0 的寄存器如下所示。

- TMQ0 控制寄存器 0(TQ0CTL0)
- TMQ0 控制寄存器 1(TQ0CTL1)
- TMQ0 I/O 控制寄存器 0(TQ0IOC0)
- TMQ0 I/O 控制寄存器 1(TQ0IOC1)
- TMQ0 I/O 控制寄存器 2(TQ0IOC2)
- TMQ0 选项寄存器 0(TQ0OPT0)
- TMQ0 捕获/比较寄存器 0(TQ0CCR0)
- TMQ0 捕获/比较寄存器 1(TQ0CCR1)
- TMQ0 捕获/比较寄存器 2(TQ0CCR2)
- TMQ0 捕获/比较寄存器 3(TQ0CCR3)
- TMQ0 计数器读取缓冲寄存器(TQ0CNT)

备注      当使用 TIQ00 到 TIQ03 和 TOQ00 到 TOQ03 引脚的功能时，请参阅 **表 4-14 端口引脚用作复用-功能引脚。**

### (1) TMQ0 控制寄存器 0(TQ0CTL0)

TQ0CTL0 寄存器为控制 TMQ0 运行的 8-位寄存器。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

相同值始终可通过软件写入 TQ0CTL0 寄存器。

复位后: 00H		R/W		地址: FFFFF540H						
		7	6	5	4	3	2	1	0	
TQ0CTL0	TQ0CE	0	0	0	0	0	TQ0CKS2	TQ0CKS1	TQ0CKS0	

TQ0CE	TMQ0操作控制
0	TMQ0操作禁用(TMQ0异步复位 <sup>注</sup> )
1	TMQ0操作禁用(TMQ0操作启动)

TQ0CKS2	TQ0CKS1	TQ0CKS0	内部计数时钟选择
0*	0*	0*	f <sub>xx</sub>
0	0	1	f <sub>xx</sub> /2
0	1	0*	f <sub>xx</sub> /4
0	1*	1	f <sub>xx</sub> /8
1	0*	0	f <sub>xx</sub> /16
1	0	1	f <sub>xx</sub> /32
1*	1	0	f <sub>xx</sub> /64
1	1	1	f <sub>xx</sub> /128

注 TQ0OPT0.TQ0OVF 位, 16-位计数器, 定时器输出(TOQ00 到 TOQ03 引脚)

注意事项 1. 当 TQ0CE 位= 0 时设置 TQ0CKS2 到 TQ0CKS0 位。

当 TQ0CE 位的值从 0 改变为 1 时, 可同时设置 TQ0CKS2 到 TQ0CKS0 位。

2. 确保位 3 到 6 清零。

备注 f<sub>xx</sub>: 主时钟频率

(2)
 TMQ0 控制寄存器 1(TQ0CTL1)

TQ0CTL1 寄存器为控制 TMQ0 运行的 8-位寄存器。  
 此寄存器可读取或写入 8-位或 1-位单元。  
 此寄存器重设为 00H。

(1/2)

复位后: 00H      R/W      地址: FFFFF541H

7	6	5	4	3	2	1	0	
TQ0CTL1	TQ0SYE	TQ0EST	TQ0EEE	0	0	TQ0MD2	TQ0MD1	TQ0MD0

TQ0SYE	调试操作模式启用控制	
0	独立操作模式（异步操作模式）	
1	调试操作模式（从属操作规格） 此模式下，定时器P可以与主定时器同步操作	
	主定时器	从定时器
	TMP2	TMP3      TMQ0
	如需调试操作模式的有关信息，请参阅7.6定时器调试操作功能。	

TQ0EST	软件触发控制
0	-
1	为外部触发输入产生有效信号。 <ul style="list-style-type: none"><li>在单触发脉冲输出模式下：输出单触发脉冲并将1写入TQ0EST位作为触发器</li><li>在外部触发脉冲模式下：输出PWM波形并将1写入TPOEST位作为触发器</li></ul>

- 注意事项
 1. 仅在外部触发脉冲输出模式或单触发脉冲输出模式下 TQ0EST 位有效。在任何其他模式下，忽略此位写入 1。
2. 确保位 3 到 4 清零。

TQ0EEE	计数时钟选择
0	外部事件计数输入的禁用操作 (通过TQ0CTL0.TQ0CK0 到TQ0CK2位选择的计数时钟进行计数)
1	外部事件计数输入的启用操作 (在外部事件计数输入信号的有效沿进行计数)
位选择是否通过外部事件计数输入的有效沿的内部计数时钟进行计数。	

TQ0MD2	TQ0MD1	TQ0MD0	定时器模式选择
0	0	0	计时器模式
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	单触发脉冲输出模式
1	0	0	输出模式
1	0	1	自由定时器模式
1	1	0	脉宽测量模式
1	1	1	三角波PWM模式

- 注意事项**
1. 在外部事件计数模式下选择外部事件计数输入而不考虑 TQ0EEE 位的值。
  2. 当 TQ0CTL0.TQ0CE 位= 0 时设置 TQ0EEE 和 TQ0MD2 到 TQ0MD0 位。(当 TQ0CE 位= 1 时可写入相同值。) 当 TQ0CE 位= 1 时进行重写不保证此操作。如果重写错误, 则 TQ0CE 位清零且接着再次设置这些位。

<R>

(3) TMQ0 I/O 控制寄存器 0(TQ0IOC0)

TQ0IOC0 寄存器为控制定时器输出(TOQ00 到 TOQ03 引脚)的 8-位寄存器。

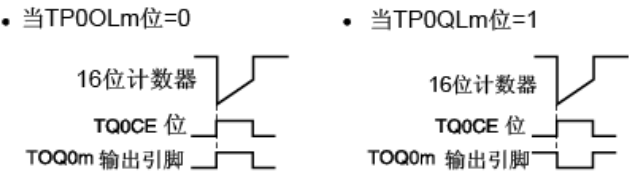
此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

复位后 : 00H      R/W      地址 : FFFF542H

TQ0IOC0	7	6	5	4	3	2	1	0
	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0OLm		TQ0m引脚输出电平设置 (m = 0 to 3) 注						
0		TQ0m引脚高电平启动						
1		TQ0m引脚低电平启动						
TQ0OEm		TQ0m引脚输出设置 (m = 0 to 3)						
0		定时器输出禁用 • 当TP0OLm 位=0: 从TQ0m引脚输出低电平 • 当TP0OLm 位=1: 从TQ0m引脚输出高电平						
1		定时器输出启用 (从TQ0m引脚输出方波)						

注      由 TQ0OLm 位指定的定时器输出引脚(TOQ0m)的输出电平如下所示。



- 注意事项
1. 当 TQ0CTL0.TQ0CE 位 = 0 时重写 TQ0OLm 和 TQ0OEm 位。(当 TQ0CE 位= 1 时可写入相同值。)如果重写错误, 则 TQ0CE 位清零且接着再次设置这些位。
  2. 即使当 TQ0CE 和 TQ0OEm 位为 0 时操作 TQ0OLm 位, TOQ0m 引脚输出电平也发生变化。

备注      m = 0 到 3

#### (4) TMQ0 I/O 控制寄存器 1(TQ0IOC1)

TQ0IOC1 寄存器为控制捕获触发输入信号(TIQ00 到 TIQ03 引脚) 有效沿的 8-位寄存器。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

复位后: 00H      R/W      地址: FFFFF543H

	7	6	5	4	3	2	1	0
TQ0IOC1	TQ0IS7	TQ0IS6	TQ0IS5	TQ0IS4	TQ0IS3	TQ0IS2	TQ0IS1	TQ0IS0

TQ0IS7	TQ0IS6	捕获触发输出信号（TIQ03引脚）有效沿设置
0	0	未检测到有效沿（捕获操作无效）
0	1	检测到上升沿
1	0	检测到下降沿
1	1	检测到两边沿

TQ0IS5	TQ0IS4	捕获触发输出信号（TIQ03引脚）有效沿检测
0	0	未检测到有效沿（捕获操作无效）
0	1	检测到上升沿
1	0	检测到下降沿
1	1	检测到两边沿

TQ0IS3	TQ0IS2	捕获触发输出信号（TIQ01引脚）有效沿设置
0	0	未检测到有效沿（捕获操作无效）
0	1	检测到上升沿
1	0	检测到下降沿
1	1	检测到两边沿

TQ0IS1	TQ0IS0	捕获触发输出信号（TIQ00引脚）有效沿设置
0	0	未检测到有效沿（捕获操作无效）
0	1	检测到上升沿
1	0	检测到下降沿
1	1	检测到两边沿

- 注意事项
1. 当 TQ0CTL0.TQ0CE 位= 0 时重写 TQ0IS7 到 TQ0IS0 位。(当 TQ0CE 位= 1 时可写入相同值。) 如果重写错误, 则 TQ0CE 位清零且接着再次设置这些位。
  2. 仅在自由定时器模式和脉宽测量模式下 TQ0IS7 到 TQ0IS0 位有效。在所有其他模式下, 捕获操作不可用。

### (5) TMQ0 I/O 控制寄存器 2(TQ0IOC2)

TQ0IOC2 寄存器为控制外部事件计数输入信号(TIQ00 引脚)和外部触发输入信号(TIQ00 引脚)的有效沿的 8-位寄存器。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

复位后: 00H      R/W      地址: FFFFF544H

	7	6	5	4	3	2	1	0
TQ0IOC2	0	0	0	0	TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0

TQ0EES1	TQ0EES0	外部事件计数输入信号 (TIQ00 引脚)有效沿设置
0	0	未检测到有效沿 (外部事件计数无效)
0	1	检测到上升沿
1	0	检测到下降沿
1	1	检测到两边沿

TQ0ETS1	TQ0ETS0	外部触发输入信号 (TIQ00 引脚)有效沿设置
0	0	未检测到有效沿 (外部触发无效)
0	1	检测到上升沿
1	0	检测到下降沿
1	1	检测到两边沿

- 注意事项**
1. 当 TQ0CTL0.TQ0CE 位= 0 时重写 TQ0EES1, TQ0EES0, TQ0ETS1 及 TQ0ETS0 位。(当 TQ0CE 位= 1 时可写入相同值。)如果重写错误, 则 TQ0CE 位清零且接着再次设置这些位。
  2. 仅当 TQ0CTL1.TQ0EEE 位= 1 时或当已设置外部事件计数模式 (TQ0CTL1.TQ0MD2 到 TQ0CTL1.TQ0MD0 位 = 001)时 TQ0EES1 和 TQ0EES0 位有效。
  3. 仅当设置外部触发脉冲输出模式 (TQ0CTL1.TQ0MD2 到 TQ0CTL1.TQ0MD0 位 = 010)或单触发脉冲输出模式(TQ0CTL1.TQ0MD2 到 TQ0CTL1.TQ0MD0 = 011) 时 TQ0ETS1 和 TQ0ETS0 位有效。

## (6) TMQ0 选项寄存器 0(TQ0OPT0)

TQ0OPT0 寄存器为 8-位寄存器用于设置捕获/比较操作和检测溢出。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

复位后: 00H      R/W      地址: FFFFF545H

	7	6	5	4	3	2	1	0
TQ0OPT0	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0	0	0	0	TQ0OVF

TQ0CCSm	TQ0CCRm寄存器捕获/比较选择
0	选择比较寄存器
1	选择捕获寄存器
尽在自由定时器模式下TQ0CCSm位设置有效	

TQ0OVF	TMQ0溢出检测
设置 (1)	发生溢出
复位 (0)	TQ0OVF位写入 0 或TQ0CTL0.TQ0CE位=0
<ul style="list-style-type: none"> <li>当自由定时器模式或脉宽测量模式下16位计数器的计数值从FFFFH溢出到0000H时TQ0OVF位设为1。</li> <li>TQ0OVF位设为1的同时产生中断请求信号 (INTTQ0OV)。 在自由定时器模式及脉宽测量模式之外不产生INTTQ0OV信号。</li> <li>即使当TQ0OVF 位=1时读取TQ0OVF位TQ0OPT0寄存器时TQ0OVF 位也不清零。</li> <li>TQ0OVF位可以读取和写入，而TQ0OVF位不能由软件设为1。 写入1不影响TMQ0的操作。</li> </ul>	

**注意事项** 1. 当 TQ0CTL0.TQ0CE 位= 0 时重写 TQ0CCS3 到 TQ0CCS0 位。(当 TQ0CE 位= 1 时可写入相同值。)  
如果重写错误，TQ0CE 位清零且接着再次设置这些位。

2. 确保位 1 到 3 清零。

**备注** m = 0 到 3



(7) TMQ0 捕获/比较寄存器 0(TQ0CCR0)

根据模式 TQ0CCR0 寄存器可用作捕获寄存器或比较寄存器。

根据 TQ0OPT0.TQ0CCS0 位的设置，仅在自由定时器模式下此寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式下，TQ0CCR0 寄存器仅可用作捕获寄存器。在任何其他模式下，此寄存器仅可用作比较寄存器。

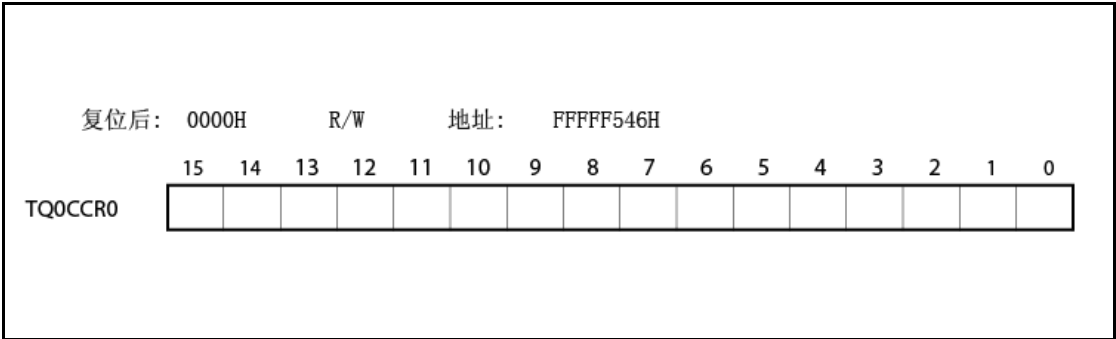
运行期间可读取或写入 TQ0CCR0 寄存器。

此寄存器可读取或写入 16-位单元。

此寄存器重设为 0000H。

**注意事项** 在以下状态禁止访问 TQ0CCR0 寄存器。如需详细信息，请参阅 3.4.8(2)访问专用片上外围 I/O 寄存器。

- 当按副时钟 CPU 运行且主时钟振荡停止时
- 当 CPU 按内部振荡时钟运行时



**(a) 用作比较寄存器**

即使当 TQ0CTL0.TQ0CE 位= 1 时也可重写 TQ0CCR0 寄存器。

TQ0CCR0 寄存器的设置值转移到 CCR0 缓冲寄存器。当 16-位计数器的值匹配 CCR0 缓冲寄存器的值时，产生比较匹配中断请求信号(INTTQ0CC0)。如果此时启用 TOQ00 引脚输出，则 TOQ00 引脚的输出反相。

在计时器模式，外部事件计数模式，外部触发脉冲输出模式，单触发脉冲输出模式，PWM 输出模式，或三角形波 PWM 模式下，当 TQ0CCR0 寄存器用作周期寄存器时，如果其计数值匹配 CCR0 缓冲寄存器的值则 16-位计数器的值清零(0000H)。

**(b) 用作捕获寄存器**

在自由定时器模式下当 TQ0CCR0 寄存器用作捕获寄存器时，如果检测到捕获触发输入引脚(TIQ00 引脚)的有效沿则 16-位计数器的计数值存储在 TQ0CCR0 寄存器。在脉冲-宽度测量模式下，如果检测到捕获触发输入引脚(TIQ00 引脚)的有效沿则 16-位计数器的计数值存储在 TQ0CCR0 寄存器且 16-位计数器清零(0000H)。

即使捕获操作和读取 TQ0CCR0 寄存器冲突，可读取 TQ0CCR0 寄存器的校正值。

下表给出了每种模式下捕获/比较寄存器的功能以及如何将数据写入比较寄存器。

**表 7-2. 每种模式下捕获/比较寄存器的功能以及如何写入比较寄存器**

操作模式	捕获/比较寄存器	如何写入比较寄存器
计时器	比较寄存器	任意时刻写入
外部事件计数器	比较寄存器	任意时刻写入
外部触发脉冲输出	比较寄存器	成批写入
单触发脉冲输出	比较寄存器	任意时刻写入
PWM 输出	比较寄存器	成批写入
自由定时器	捕获/比较寄存器	任意时刻写入
脉宽测量	捕获寄存器	—
三角形波 PWM 模式	比较寄存器	成批写入

(8) TMQ0 捕获/比较寄存器 1(TQ0CCR1)

根据模式 TQ0CCR1 寄存器可用作捕获寄存器或比较寄存器。

根据 TQ0OPT0.TQ0CCS1 位的设置，仅在自由定时器模式下此寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式下，TQ0CCR1 寄存器仅可用作捕获寄存器。在任何其他模式下，此寄存器仅可用作比较寄存器。

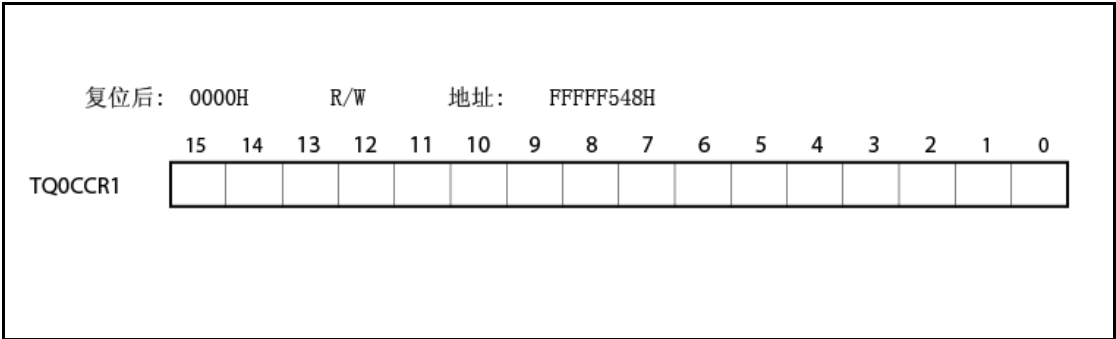
TQ0CCR1 寄存器可读取或写入运行期间。

此寄存器可读取或写入 16-位单元。

此寄存器重设为 0000H。

**注意事项** 在以下状态禁止访问 TQ0CCR1 寄存器。如需详细信息，请参阅 3.4.8(2)访问专用片上外围 I/O 寄存器。

- 当按副时钟 CPU 运行且主时钟振荡停止时
- 当 CPU 按内部振荡时钟运行时



**(a) 用作比较寄存器**

即使当 TQ0CTL0.TQ0CE 位= 1 时也可重写 TQ0CCR1 寄存器。

TQ0CCR1 寄存器的设置值转移到 CCR1 缓冲寄存器。当 16-位计数器的值匹配 CCR1 缓冲寄存器的值时，产生比较匹配中断请求信号(INTTQ0CC1)。如果此时启用 TOQ01 引脚输出，则 TOQ01 引脚的输出反相。

**(b) 用作捕获寄存器**

在自由定时器模式下当 TQ0CCR1 寄存器用作捕获寄存器时，如果检测到捕获触发输入引脚(TIQ01 引脚)的有效沿则 16-位计数器的计数值存储在 TQ0CCR1 寄存器。在脉冲-宽度测量模式下，如果检测到捕获触发输入引脚(TIQ01 引脚)的有效沿则 16-位计数器的计数值存储在 TQ0CCR1 寄存器且 16-位计数器清零(0000H)。

即使捕获操作和读取 TQ0CCR1 寄存器冲突，也可读取 TQ0CCR1 寄存器的校正值。

下表给出了每种模式下捕获/比较寄存器的功能以及如何将数据写入比较寄存器。

**表 7-3. 每种模式下捕获/比较寄存器的功能和如何写入比较寄存器**

操作模式	捕获/比较寄存器	如何写入比较寄存器
计时器	比较寄存器	任意时刻写入
外部事件计数器	比较寄存器	任意时刻写入
外部触发脉冲输出	比较寄存器	成批写入
单触发脉冲输出	比较寄存器	任意时刻写入
PWM 输出	比较寄存器	成批写入
自由定时器	捕获/比较寄存器	任意时刻写入
脉宽测量	捕获寄存器	—
三角形波 PWM 模式	比较寄存器	成批写入

(9) TMQ0 捕获/比较寄存器 2(TQ0CCR2)

根据模式 TQ0CCR2 寄存器可用作捕获寄存器或比较寄存器。

根据 TQ0OPT0.TQ0CCS2 位的设置，仅在自由定时器模式下此寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式下，TQ0CCR2 寄存器仅可用作捕获寄存器。在任何其他模式下，此寄存器仅可用作比较寄存器。

运行期间可读取或写入 TQ0CCR2 寄存器。

此寄存器可读取或写入 16-位单元。

此寄存器重设为 0000H。

**注意事项** 在以下状态禁止访问 TQ0CCR2 寄存器。如需详细信息，请参阅 3.4.8(2)访问专用片上外围 I/O 寄存器。

- 当按副时钟 CPU 运行且主时钟振荡停止时
- 当 CPU 按内部振荡时钟运行时

复位后: 0000H	R/W	地址: FFFF54AH																
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CCR2																		

**(a) 用作比较寄存器**

即使当 TQ0CTL0.TQ0CE 位= 1 时也可重写 TQ0CCR2 寄存器。

TQ0CCR2 寄存器的设置值转移到 CCR2 缓冲寄存器。当 16-位计数器的值匹配 CCR2 缓冲寄存器的值时，产生比较匹配中断请求信号(INTTQ0CC2)。如果此时启用 TOQ02 引脚输出，则 TOQ02 引脚的输出反相。

**(b) 用作捕获寄存器**

在自由定时器模式下当 TQ0CCR2 寄存器用作捕获寄存器时，如果检测到捕获触发输入引脚(TIQ02 引脚)的有效沿则 16-位计数器的计数值存储在 TQ0CCR2 寄存器。在脉冲-宽度测量模式下，如果检测到捕获触发输入引脚(TIQ02 引脚)的有效沿则 16-位计数器的计数值存储在 TQ0CCR2 寄存器且 16-位计数器清零(0000H)。

即使捕获操作和读取 TQ0CCR2 寄存器冲突，也可读取 TQ0CCR2 寄存器的校正值。

下表给出了每种模捕获/比较寄存器的功能式以及如何将数据写入比较寄存器。

**表 7-4. 每种模式下捕获/比较寄存器的功能和如何写入比较寄存器**

操作模式	捕获/比较寄存器	如何写入比较寄存器
计时器	比较寄存器	任意时刻写入
外部事件计数器	比较寄存器	任意时刻写入
外部触发脉冲输出	比较寄存器	成批写入
单触发脉冲输出	比较寄存器	任意时刻写入
PWM 输出	比较寄存器	成批写入
自由定时器	捕获/比较寄存器	任意时刻写入
脉宽测量	捕获寄存器	—
三角形波 PWM 模式	比较寄存器	成批写入

(10)TMQ0 捕获/比较寄存器 3(TQ0CCR3)

根据模式 TQ0CCR3 寄存器可用作捕获寄存器或比较寄存器。

根据 TQ0OPT0.TQ0CCS3 位的设置，仅在自由定时器模式下此寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式下，TQ0CCR3 寄存器仅可用作捕获寄存器。在任何其他模式下，此寄存器仅可用作比较寄存器。

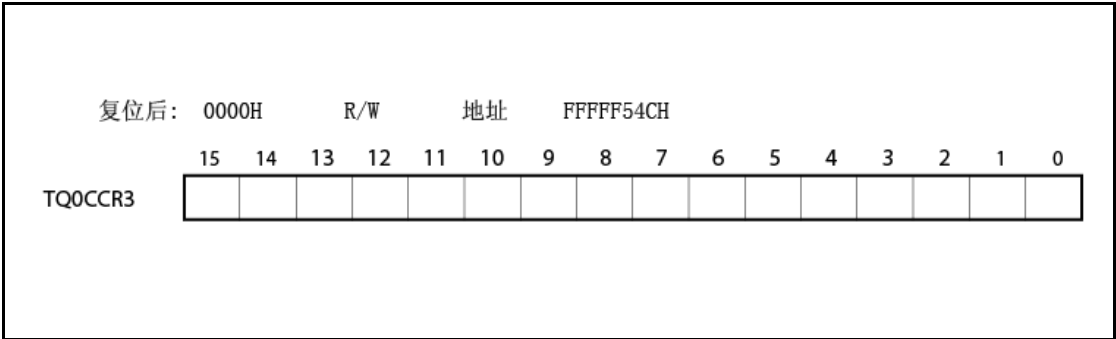
TQ0CCR3 寄存器可读取或写入运行期间。

此寄存器可读取或写入 16-位单元。

此寄存器重设为 0000H。

**注意事项** 在以下状态禁止访问 TQ0CCR3 寄存器。如需详细信息，请参阅 3.4.8(2)访问专用片上外围 I/O 寄存器。

- 当按副时钟 CPU 运行且主时钟振荡停止时
- 当 CPU 按内部振荡时钟运行时



**(a) 用作比较寄存器**

即使当 TQ0CTL0.TQ0CE 位= 1 时也可重写 TQ0CCR3 寄存器。

TQ0CCR3 寄存器的设置值转移到 CCR3 缓冲寄存器。当 16-位计数器的值匹配 CCR3 缓冲寄存器的值时，产生比较匹配中断请求信号(INTTQ0CC3)。如果此时启用 TOQ03 引脚输出，则 TOQ03 引脚的输出反相。

**(b) 用作捕获寄存器**

在自由定时器模式下当 TQ0CCR3 寄存器用作捕获寄存器时，如果检测到捕获触发输入引脚(TIQ03 引脚)的有效沿则 16-位计数器的计数值存储在 TQ0CCR3 寄存器。在脉冲-宽度测量模式下，如果检测到捕获触发输入引脚(TIQ03 引脚)的有效沿则 16-位计数器的计数值存储在 TQ0CCR3 寄存器且 16-位计数器清零(0000H)。

即使捕获操作和读取 TQ0CCR3 寄存器冲突，也可读取 TQ0CCR3 寄存器的校正值。

下表给出了每种模式下捕获/比较寄存器的功能以及如何将数据写入比较寄存器。

**表 7-5. 每种模式下捕获/比较寄存器的功能和如何写入比较寄存器**

操作模式	捕获/比较寄存器	如何写入比较寄存器
计时器	比较寄存器	任意时刻写入
外部事件计数器	比较寄存器	任意时刻写入
外部触发脉冲输出	比较寄存器	成批写入
单触发脉冲输出	比较寄存器	任意时刻写入
PWM 输出	比较寄存器	成批写入
自由定时器	捕获/比较寄存器	任意时刻写入
脉宽测量	捕获寄存器	—
三角形波 PWM 模式	比较寄存器	成批写入



(11)TMQ0 计数器读取缓冲寄存器(TQ0CNT)

TQ0CNT 寄存器为可读取 16-位计数器的计数值的读取缓冲寄存器。

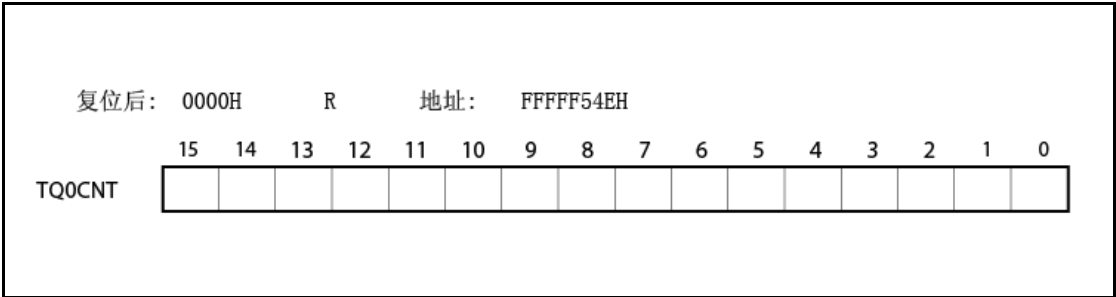
如果当 TQ0CTL0.TQ0CE 位= 1 时读取此寄存器，则可读取 16-位定时器的计数值。

此寄存器为 16-位单元只读寄存器。

当 TQ0CE 位= 0 时 TQ0CNT 寄存器的值清为 0000H。如果此时读取 TQ0CNT 寄存器，则不读取 16-位计数器的值(FFFFH)，而读取 0000H。

复位后 TQ0CNT 寄存器的值清为 0000H，而 TQ0CE 位清为 0。

- 注意事项 在以下状态禁止访问 TQ0CNT 寄存器。如需详细信息，请参阅 3.4.8(2)访问专用片上外围 I/O 寄存器。
- 当按副时钟 CPU 运行且主时钟振荡停止时
  - 当 CPU 按内部振荡时钟运行时



### (12) TIQ0m 引脚噪音消除控制寄存器(Q0mNFC)

**Q0mNFC** 寄存器为设置用于消除定时器 Q 输入引脚噪音的数字噪音滤波器的 8-位寄存器。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

复位后: 00H      R/W    Address:      Q00NFC: FFFFFB50H(TIQ00 引脚)

Q01NFC: FFFFB54H(TIQ01 引脚)

Q02NFC: FFFFB58H(TIQ02 引脚)

Q03NFC: FFFFB5CH(TIQ03 引脚)

	7	6	5	4	3	2	1	0
Q0mNFC	0	NFSTS	0	0	0	NFC2	NFC1	NFC0

( $m = 0$  到 3)

NFSTS	数字噪音滤波器采样的次数设置
0	3 次
1	2 次

NFC2	NFC1	NFC0	采样时钟
0	0	0	$f_{xx}$
0	0	1	$f_{xx}/2$
0	1	0	$f_{xx}/4$
0	1	1	$f_{xx}/16$
1	0	0	$f_{xx}/32$
1	0	1	$f_{xx}/64$
以上之外			禁止设置

**注意事项** 1. 确保位 3 到 5 和 7 清零。

2. 输出设置 Q0mNFC 寄存器之前输入到定时器输入引脚(TIQ0m)的信号且消除数字噪音。

因此，使用 Q0mNFC 寄存器设置采样时钟(NFC2 到 NFC0)和采样次数(NFSTS)，等待初始化时间=(采样时钟)×(采样次数)，且启动定时器运行。

备注 可准确消除的噪音宽度为(采样时钟) × (采样次数- 1)。如果与采样时钟同步, 则甚至宽度窄于此的噪音可能导致错误计数。

## 7.5 操作

TMQ0 可执行以下操作。

运行	TQ0CTL1.TQ0EST 位 (软件触发位)	TIQ00 引脚 (外部触发输入)	捕获/比较寄存器设置	比较寄存器写入
计时器模式	无效	无效	仅比较寄存器	任意时刻写入
外部事件计数模式 <sup>注 1</sup>	无效	无效	仅比较寄存器	任意时刻写入
外部触发脉冲输出模式 <sup>注 2</sup>	有效	有效	仅比较寄存器	成批写入
单触发脉冲输出模式 <sup>注 2</sup>	有效	有效	仅比较寄存器	任意时刻写入
PWM 输出模式	无效	无效	仅比较寄存器	成批写入
自由定时器模式	无效	无效	启用切换	任意时刻写入
脉宽测量模式 <sup>注 2</sup>	无效	无效	仅捕获寄存器	不可用
三角形波 PWM 模式	无效	无效	仅比较寄存器	成批写入

- 注**
1. 要使用外部事件计数模式，指定未检测到 TIQ00 引脚捕获触发输入的有效沿(通过 TQ0IOC1.TQ0IS1 且 TQ0IOC1.TQ0IS0 位清零)。
  2. 当使用外部触发脉冲输出模式，单触发脉冲输出模式以及脉宽测量模式时，选择内部时钟作为计数时钟(通过 TQ0CTL1.TQ0EEE 位清零)。

7.5.1 计时器模式(TQ0MD2 到 TQ0MD0 位 = 000)

在计时器模式下，如果 TQ0CTL0.TQ0CE 位设为 1 则以指定间隔产生中断请求信号(INTTQ0CC0)。可从 TQ0Q0 引脚输出半周期等于此间隔的方波。

通常，在计时器模式下不使用 TQ0CCR1 到 TQ0CCR3 寄存器。

图 7-2. 计时器配置

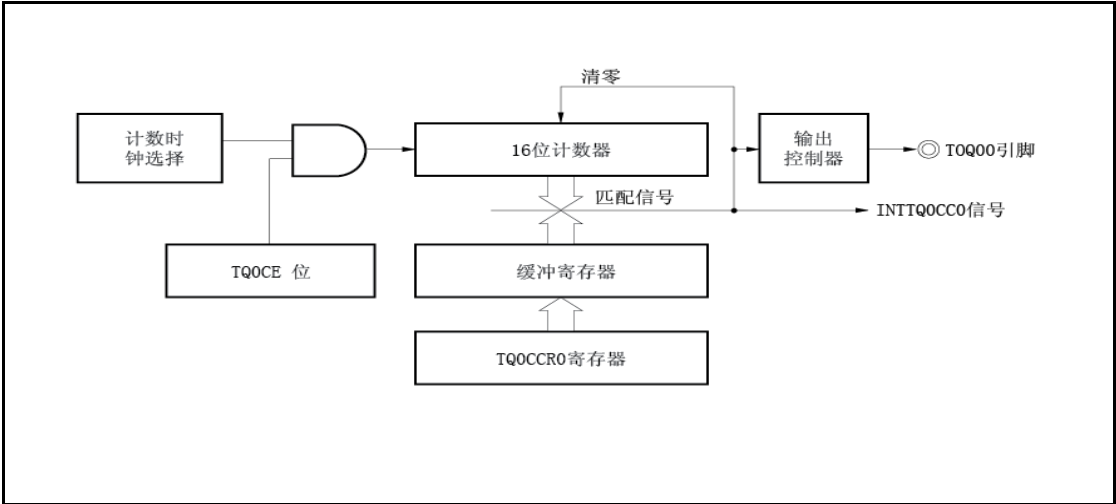
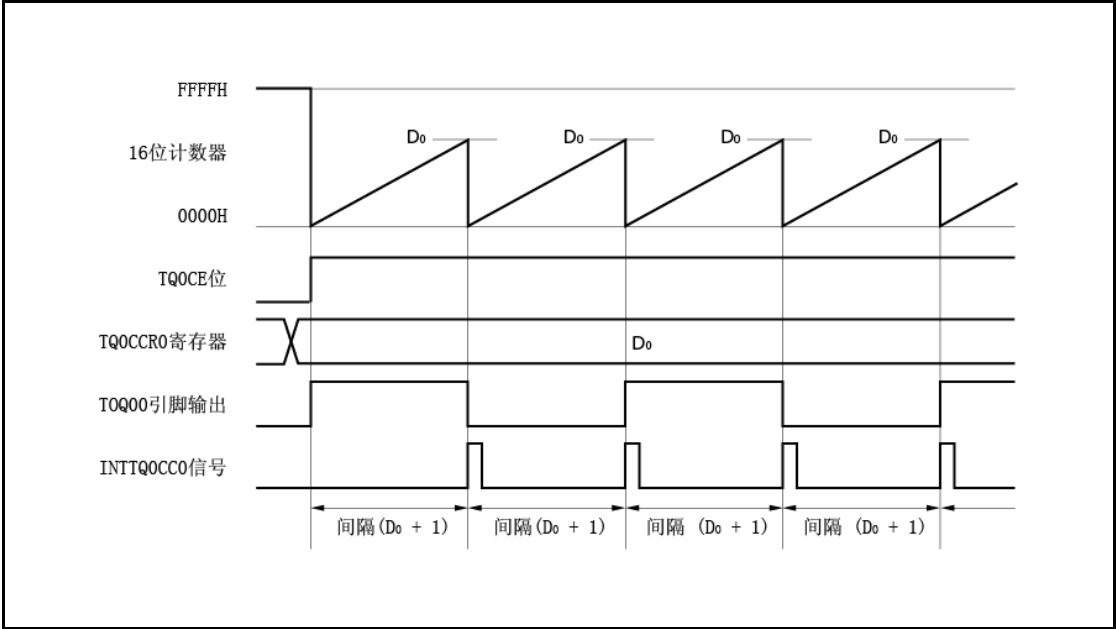


图 7-3. 在计时器模式下运行基本时序



当 TQ0CE 位设为 1 时，16-位计数器的值与计数时钟同步从 FFFFH 清为 0000H，且计数器开始计数。此时，TOQ00 引脚的输出反相。另外，TQ0CCR0 寄存器的设置值转移到 CCR0 缓冲寄存器。

当 16-位计数器的计数值匹配 CCR0 缓冲寄存器的值，16-位计数器清为 0000H，TOQ00 引脚的输出反相，且产生比较匹配中断请求信号(INTTQ0CC0)。

间隔可由以下表达式计算。

$$\text{间隔} = (\text{TQ0CCR0 寄存器的设置值} + 1) \times \text{计数时钟周期}$$

图 7-4. 计时器模式运行的寄存器设置(1/2)

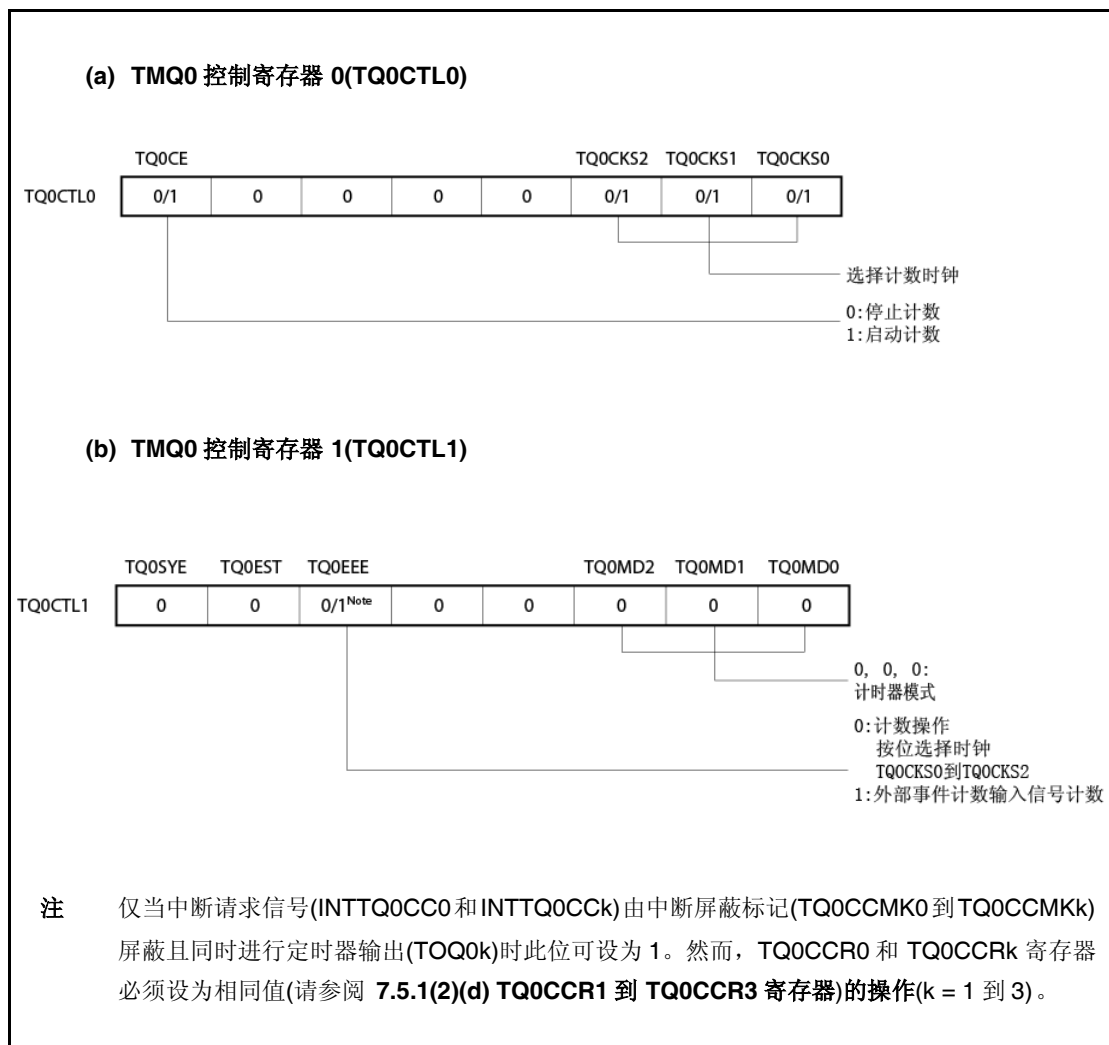
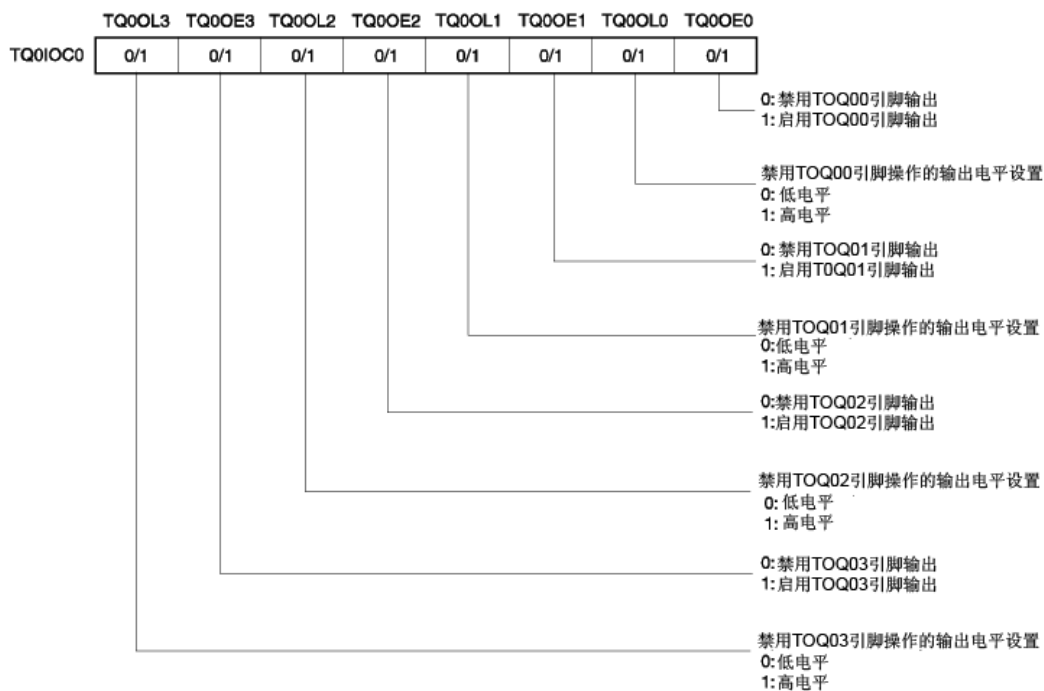


图 7-4. 计时器模式运行的寄存器设置(2/2)

(c) TMQ0 I/O 控制寄存器 0(TQ0IOC0)



(d) TMQ0 计数器读取缓冲寄存器(TQ0CNT)

通过读取 TQ0CNT 寄存器，即可读取 16-位计数器的计数值。

(e) TMQ0 捕获/比较寄存器 0(TQ0CCR0)

如果 TQ0CCR0 寄存器设为  $D_0$ ，则间隔如下所示。

$$\text{间隔} = (D_0 + 1) \times \text{计数时钟周期}$$

(f) TMQ0 捕获/比较寄存器 1 到 3(TQ0CCR1 到 TQ0CCR3)

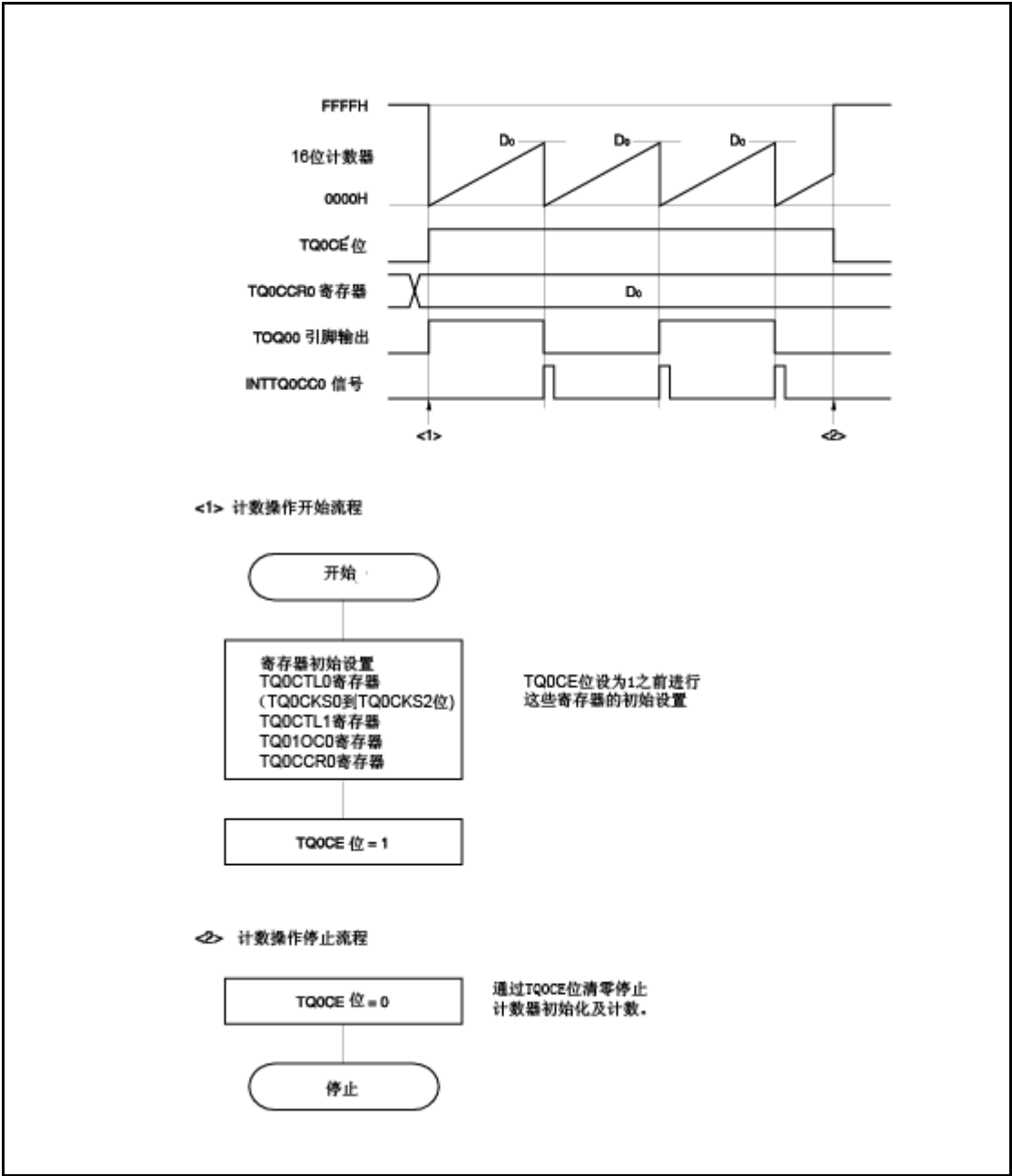
通常，在计时器模式下不使用 TQ0CCR1 到 TQ0CCR3 寄存器。然而，TQ0CCR1 到 TQ0CCR3 寄存器的设置值转移到 CCR1 到 CCR3 缓冲寄存器。当 16-位计数器的计数值匹配 CCR1 到 CCR3 缓冲寄存器的值时产生比较匹配中断请求信号(INTTQ0CC1 到 INTTQ0CCR3)。

因此，使用对应中断屏蔽标记(TQ0CCMK1 到 TQ0CCMK3) 屏蔽中断请求。

**备注** 在计时器模式下不使用 TMQ0 I/O 控制寄存器 1(TQ0IOC1)，TMQ0 I/O 控制寄存器 2(TQ0IOC2)及 TMQ0 选项寄存器 0(TQ0OPT0)。

(1) 计时器模式操作流程

图 7-5. 计时器模式下的软件处理流程

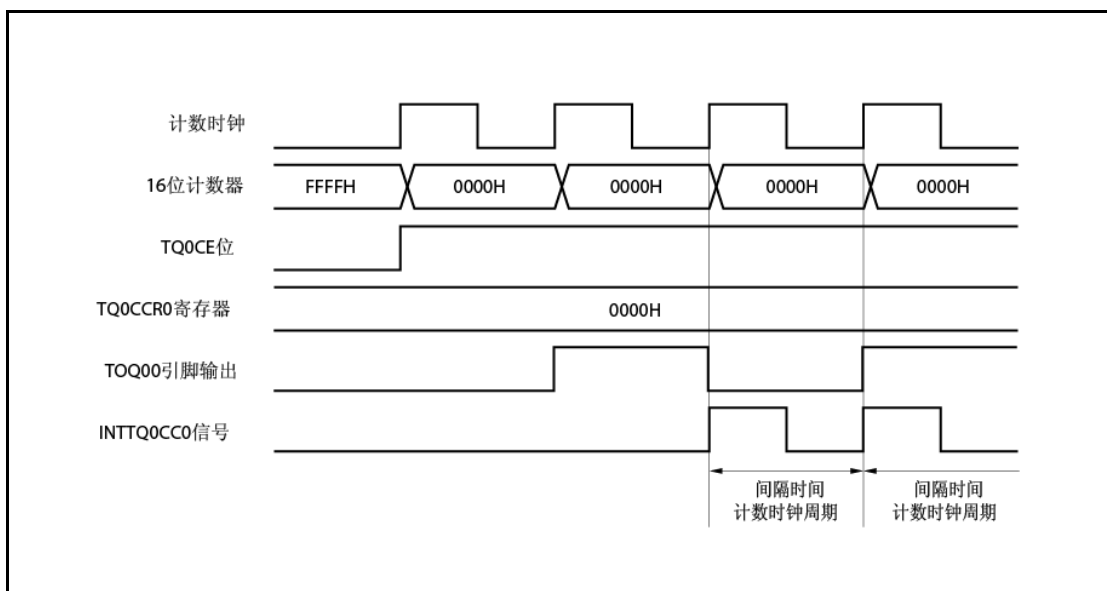


## (2) 计时器模式运行时序

### (a) 如果 TQ0CCR0 寄存器设为 0000H 时的操作

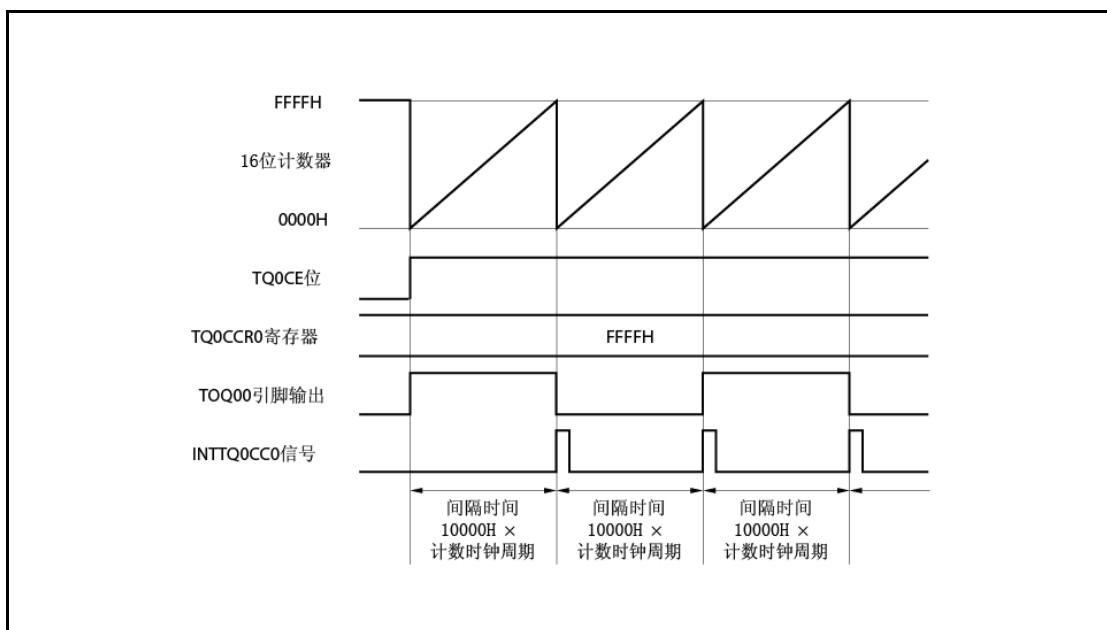
如果 TQ0CCR0 寄存器设为 0000H，则在第一个计数时钟脉冲之后的每个计数时钟脉冲产生 INTTQ0CC0 信号，且 TOQ00 引脚的输出反相。

16-位计数器的值始终为 0000H。



### (b) 如果 TQ0CCR0 寄存器设为 FFFFH 时的操作

如果 TQ0CCR0 寄存器设为 FFFFH，则 16-位计数器计数上至 FFFFH。计数器与下一数完时序同步清为 0000H。INTTQ0CC0 信号产生且 TOQ00 引脚的输出反相。此时，不产生溢出中断请求信号(INTTQ0OV)，溢出标记(TQ0OPT0.TQ0OVF 位)也不设为 1。

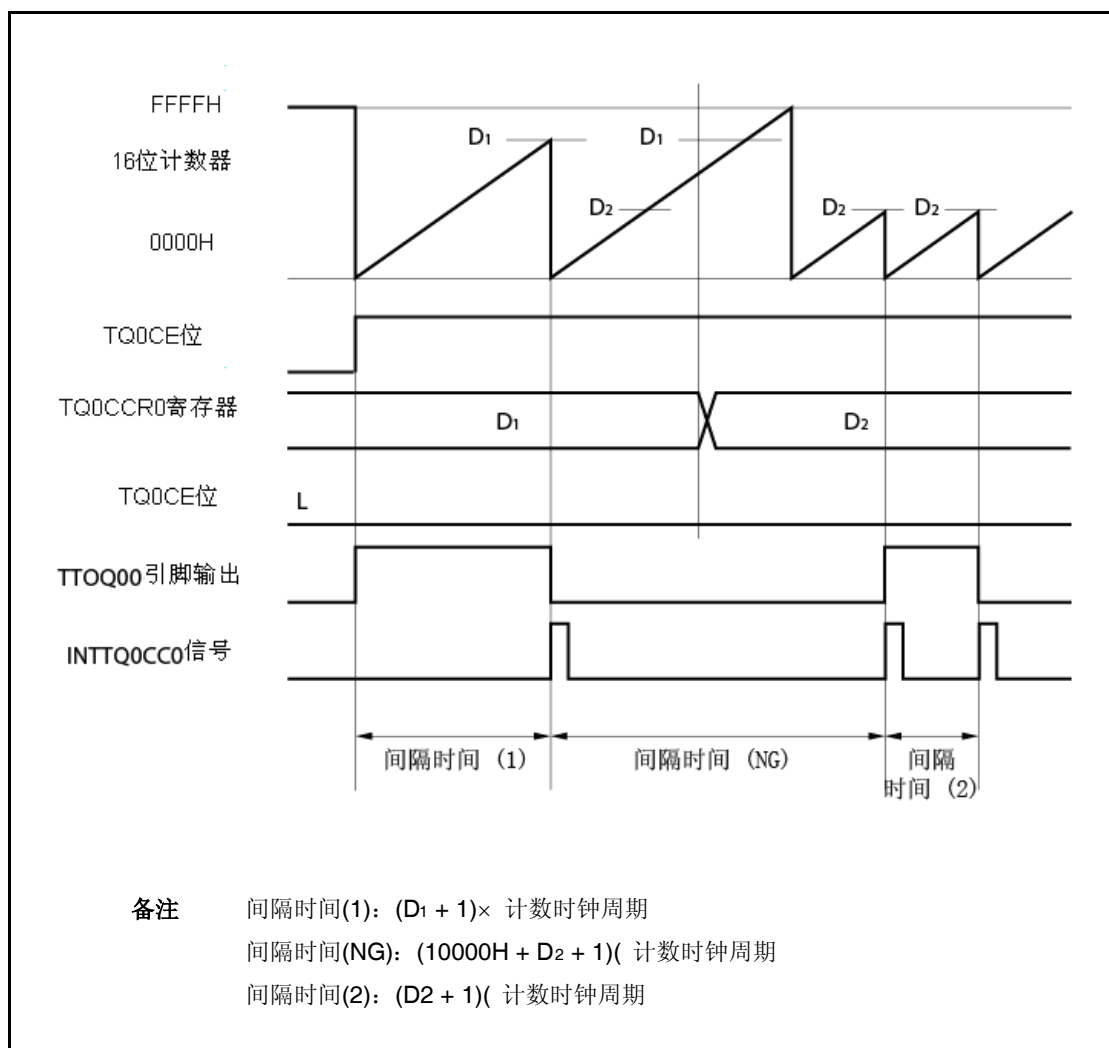




### (c) 重写 TQ0CCR0 寄存器的注意事项

要将 TQ0CCR0 寄存器的值改为更小的值，则停止计数一次且接着改变设置值。

如果计数期间 TQ0CCR0 寄存器的值重写为更小的值，则 16-位计数器可能溢出。



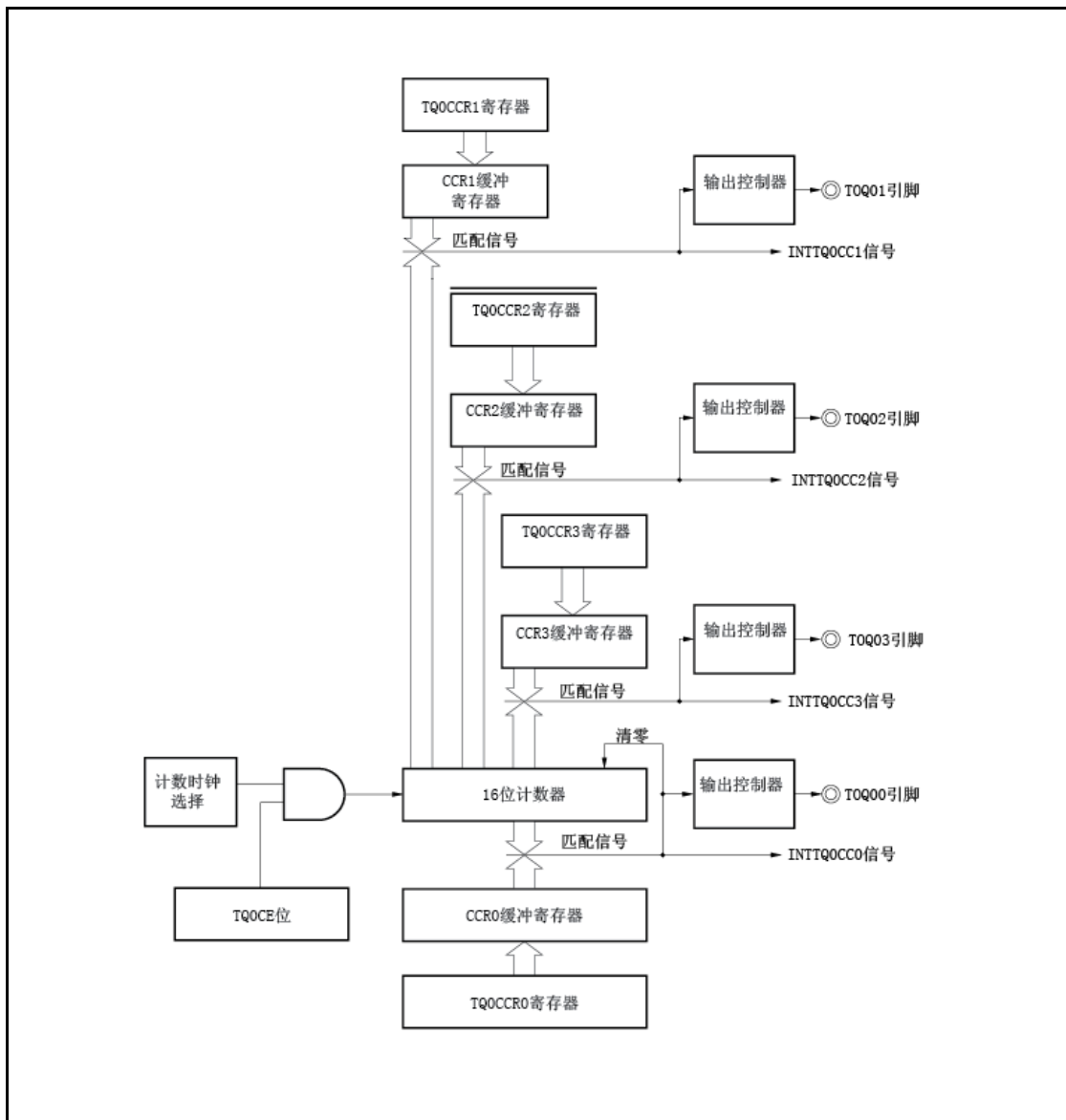
如果当计数值大于 D2 而小于 D1 时 TQ0CCR0 寄存器的值从 D1 改为 D2，则重写 TQ0CCR0 寄存器同时计数值转移到 CCR0 缓冲寄存器。结果，比较的 16-位计数器的值为 D2。

然而，因为计数值已超过 D2，16-位计数器计数上至 FFFFH，溢出，且接着再次从 0000H 计数。当计数值匹配 D2 时，产生 INTTQ0CCR0 信号且 TQ0Q0 引脚的输出反相。

因此，不能在开始预计的间隔时间 “ $(D_1 + 1) \times \text{计数时钟周期}$ ” 或 “ $(D_2 + 1) \times \text{计数时钟周期}$ ” 产生 INTTQ0CCR0 信号，而可能 “ $(10000H + D_2 + 1) \times \text{计数时钟周期}$ ” 间隔产生。

(d) TQ0CCR1 到 TQ0CCR3 寄存器的操作

图 7-6. TQ0CCR1 到 TQ0CCR3 寄存器的配置

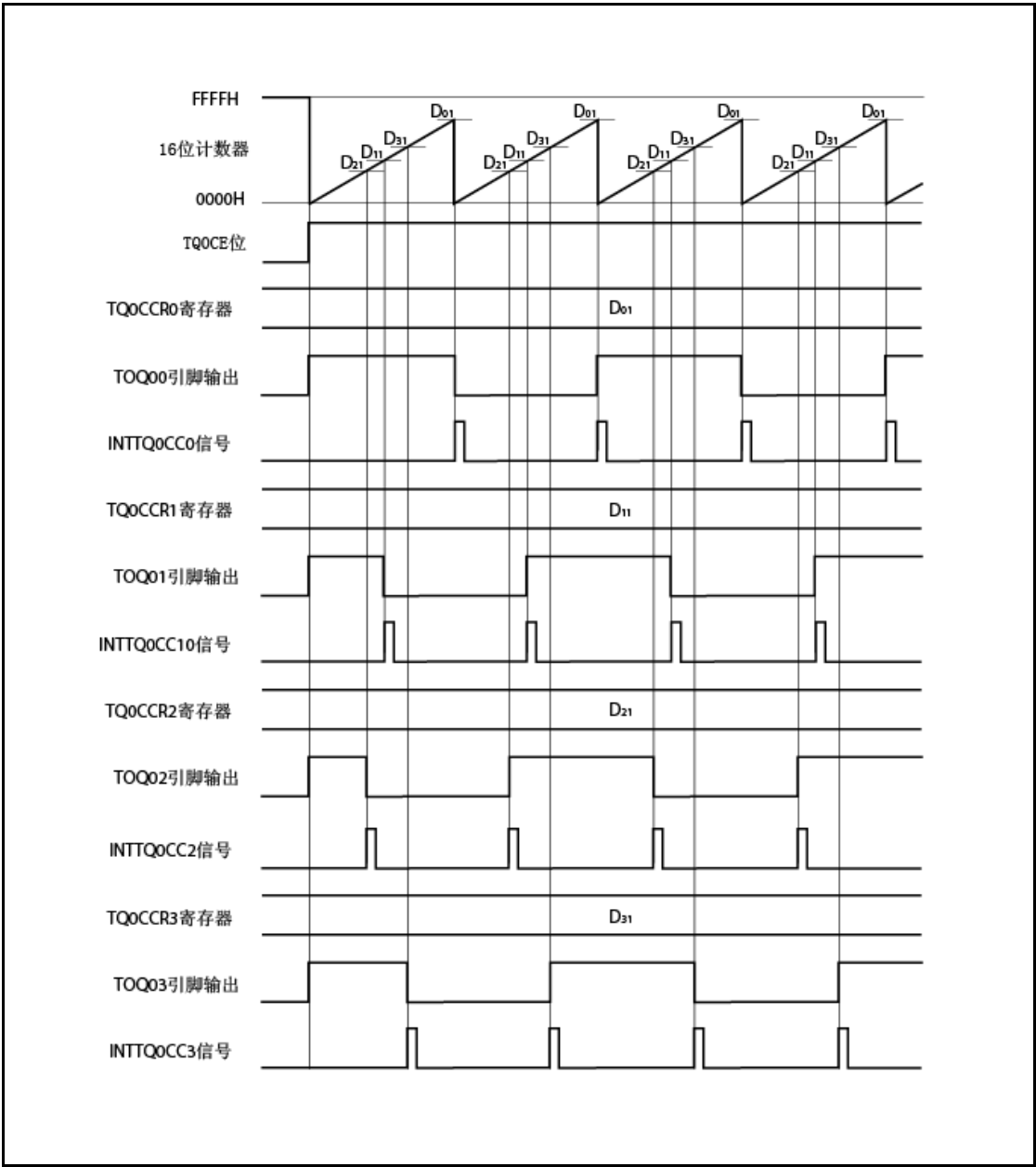


如果 TQ0CCRk 寄存器的设置值小于 TQ0CCR0 寄存器的设置值，INTTQ0CCk 信号每个周期产生一次。同时，TOPQ0k 引脚的输出反相。

TQ00k 引脚输出具有与 TQ000 引脚输出周期相同的方波。

备注 k = 1 到 3

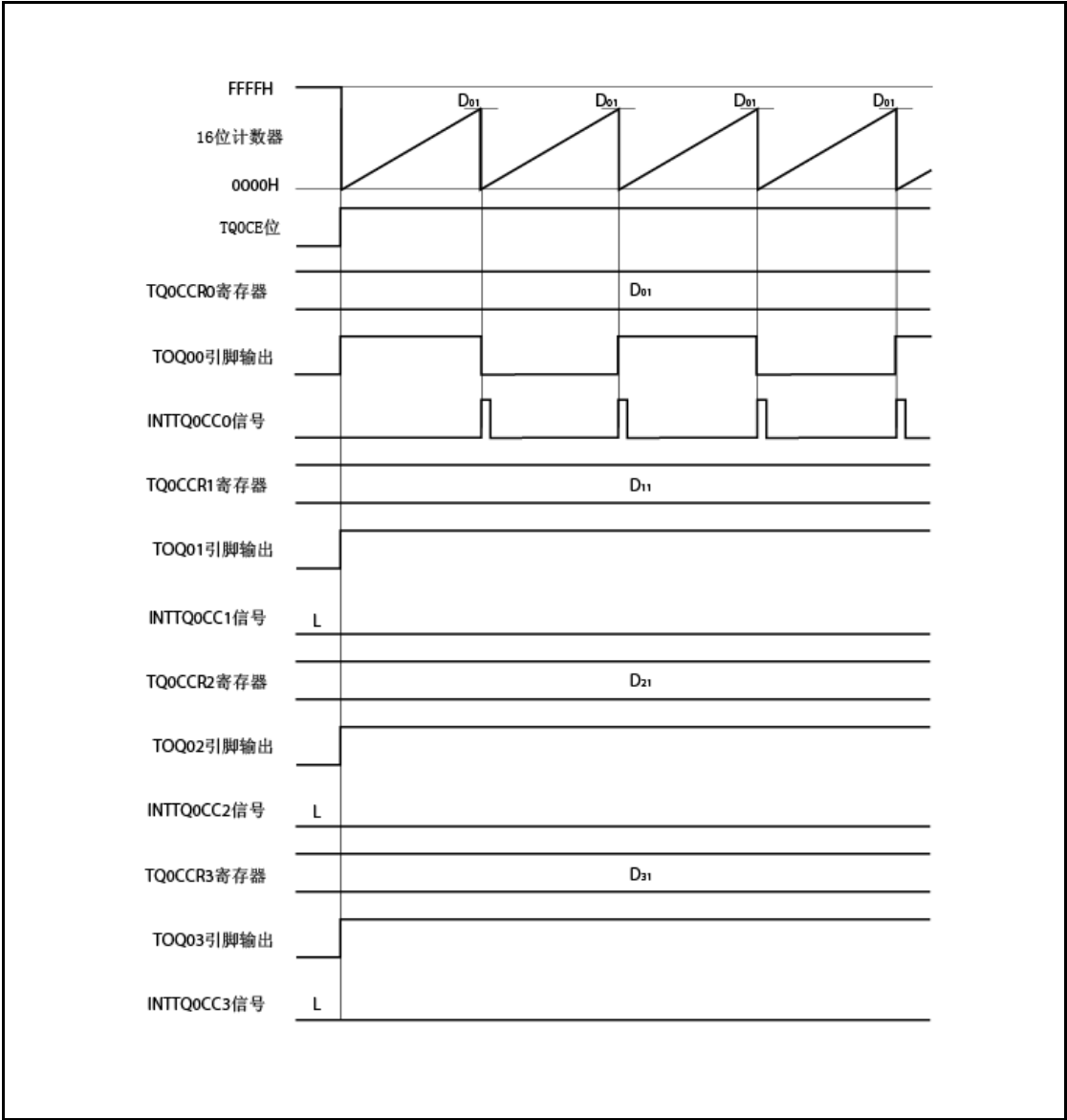
图 7-7. 当 D01 ≥ Dk1 时的时序图



如果 TQ0CCRk 寄存器的设置值大于 TQ0CCR0 寄存器的设置值，则 16-位计数器的计数值不匹配 TQ0CCRk 寄存器的值。结果，不产生 INTTQ0CCk 信号，也不改变 TOQ0k 引脚的输出。

备注      k = 1 到 3

图 7-8. 当 D<sub>01</sub> < D<sub>k1</sub> 时的时序图



7.5.2 外部事件计数模式(TQ0MD2 到 TQ0MD0 位 = 001)

在外部事件计数模式下，当 TQ0CTL0.TQ0CE 位设为 1 时对外部事件计数输入的有效沿计数，且每当指定已计数的边沿数时产生中断请求信号(INTTQ0CC0)。不能使用 TOQ00 引脚。  
通常，在外部事件计数模式下不使用 TQ0CCR1 到 TQ0CCR3 寄存器。

图 7-9. 在外部事件计数模式下的配置

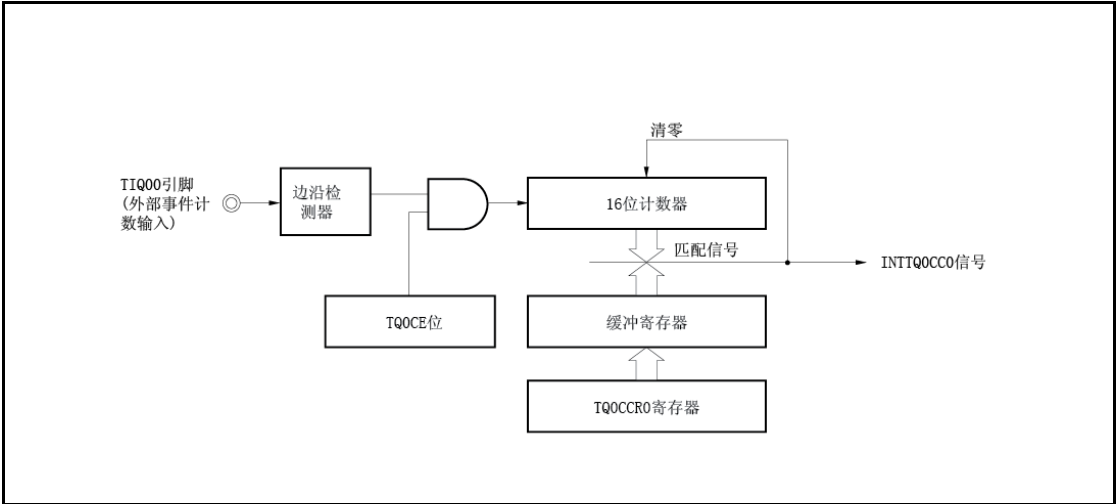
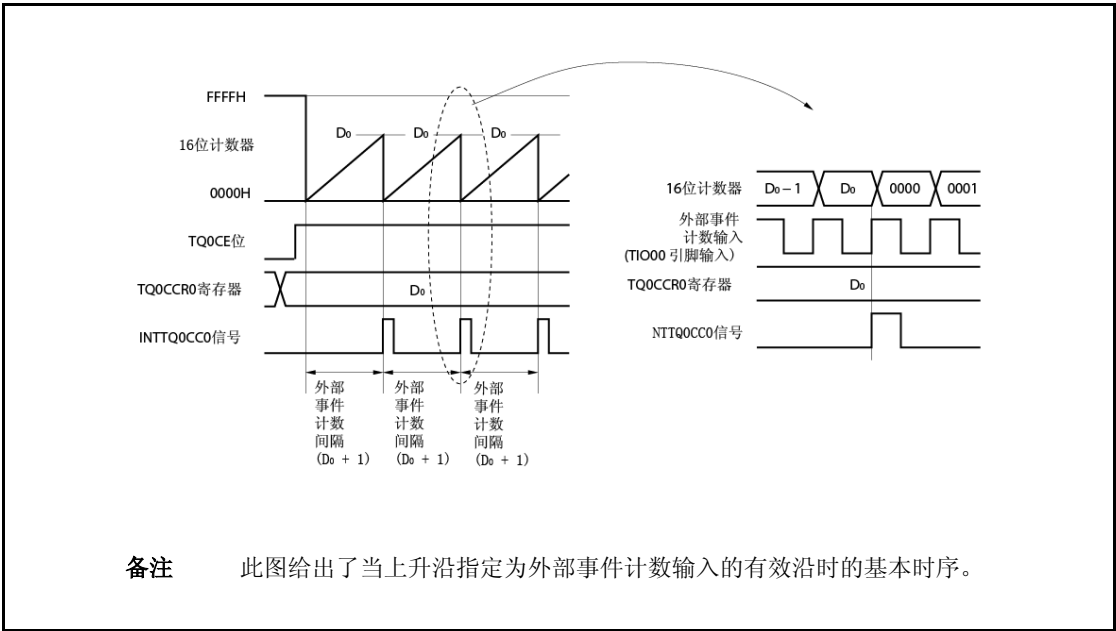


图 7-10. 在外部事件计数模式下的基本时序



**备注** 此图给出了当上升沿指定为外部事件计数输入的有效沿时的基本时序。

当 TQ0CE 位设为 1 时, 16-位计数器的值从 FFFFH 清为 0000H。每当检测到外部事件计数输入的有效沿时计数器计数。另外, TQ0CCR0 寄存器的设置值转移到 CCR0 缓冲寄存器。

当 16-位计数器的计数值匹配 CCR0 缓冲寄存器的值时, 16-位计数器清为 0000H, 且产生比较匹配中断请求信号(INTTQ0CC0)。

每当检测到外部事件计数输入的有效沿(TQ0CCR0 寄存器的设置值+ 1)次时产生 INTTQ0CC0 信号。

图 7-11. 在外部事件计数模式下操作的寄存器设置(1/2)

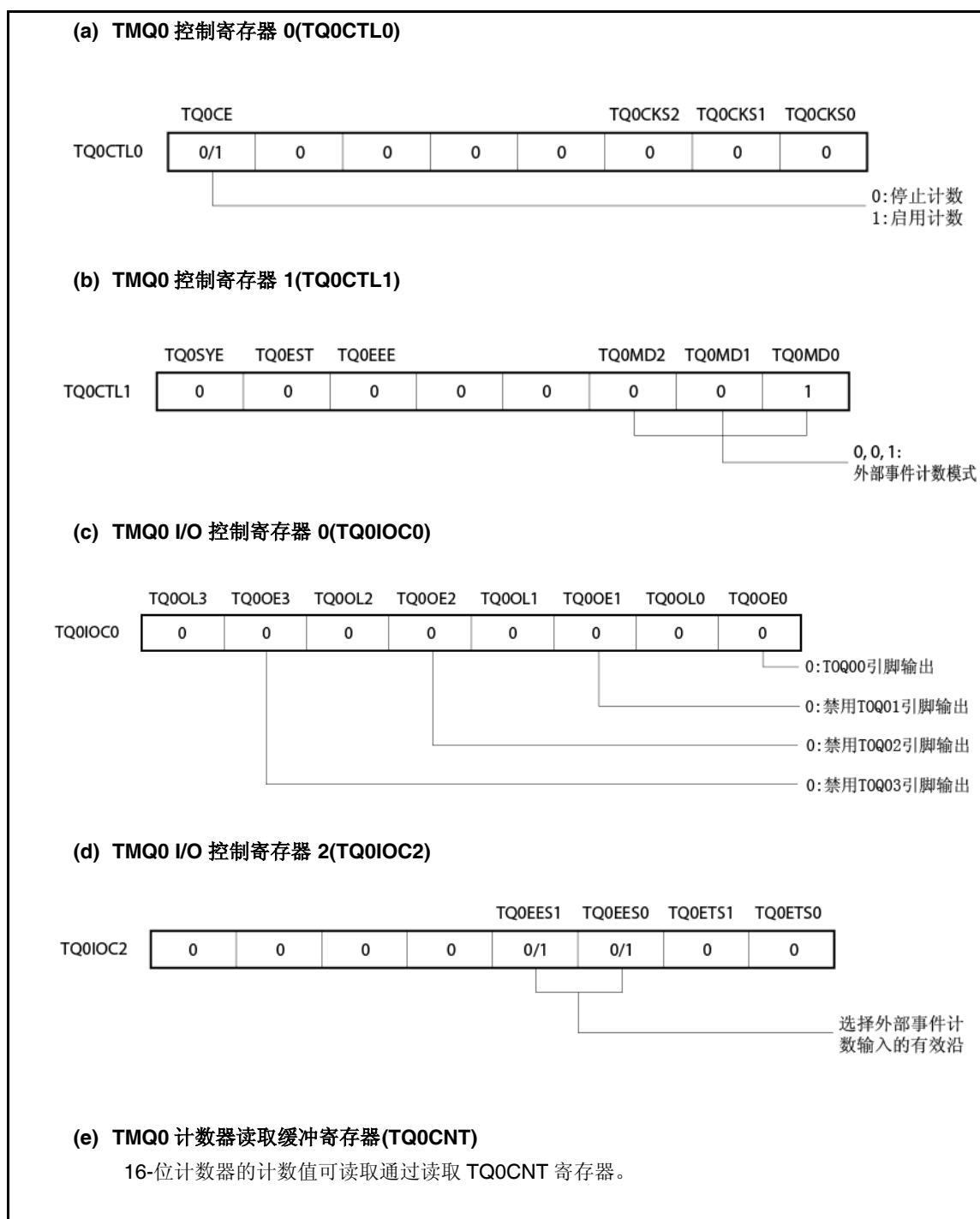


图 7-11. 在外部事件计数模式下操作的寄存器设置(2/2)

**(f) TMQ0 捕获/比较寄存器 0(TQ0CCR0)**

如果  $D_0$  设为 TQ0CCR0 寄存器, 则计数器清零且当外部事件计数达到  $(D_0 + 1)$  时产生比较匹配中断请求信号(INTTQ0CC0)。

**(g) TMQ0 捕获/比较寄存器 1 到 3(TQ0CCR1 到 TQ0CCR3)**

通常, 在外部事件计数模式下不使用 TQ0CCR1 到 TQ0CCR3 寄存器。然而, TQ0CCR1 到 TQ0CCR3 寄存器的设置值转移到 CCR1 到 CCR3 缓冲寄存器。当 16-位计数器的计数值匹配 CCR1 到 CCR3 缓冲寄存器的值时, 产生比较匹配中断请求信号(INTTQ0CC1 到 INTTQ0CC3)。

因此, 使用中断屏蔽标记(TQ0CCMK1 到 TQ0CCMK3) 屏蔽中断信号。

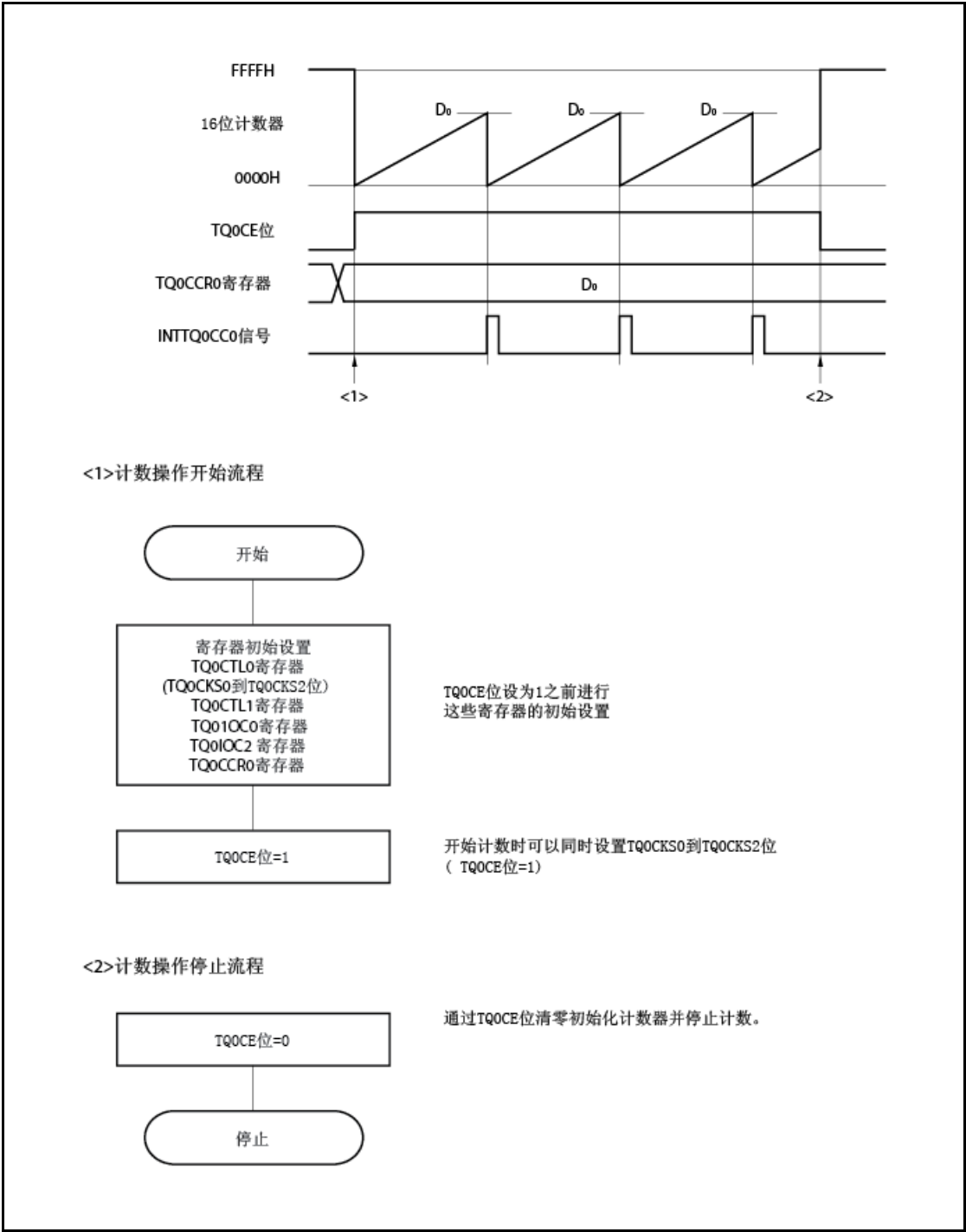
**注意事项** 当外部时钟用作计数时钟时, 外部时钟仅可从 TIQ00 引脚输入。此时, 设置 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位为 00(捕获触发输入(TIQ00 引脚): 未检测到边沿)。

**备注** 在外部事件计数模式下不使用 TMQ0 I/O 控制寄存器 1(TQ0IOC1)和 TMQ0 选项寄存器 0(TQ0OPT0)。

<R>

(1) 外部事件计数模式操作流程

图 7-12. 在外部事件计数模式下的软件处理流程



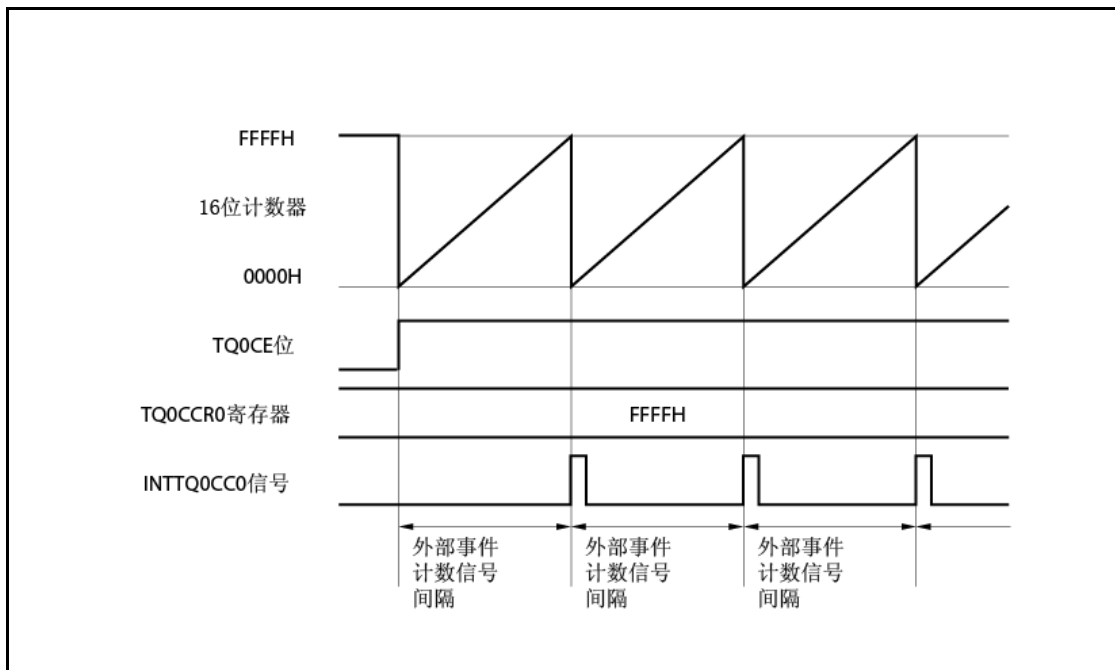


## (2) 在外部事件计数模式下的操作时序

- 注意事项
1. 在外部事件计数模式下，TQ0CCR0 寄存器禁止设为 0000H。
  2. 在外部事件计数模式下，禁止使用定时器输出。如果使用外部事件计数输入进行定时器输出，设置计时器模式，且选择通过计数时钟的外部事件计数输入启用的操作 (TQ0CTL1.TQ0MD2 到 TQ0CTL1.TQ0MD0 位 = 000，TQ0CTL1.TQ0EEE 位 = 1)。

### (a) 如果 TQ0CCR0 寄存器设为 FFFFH 时的操作

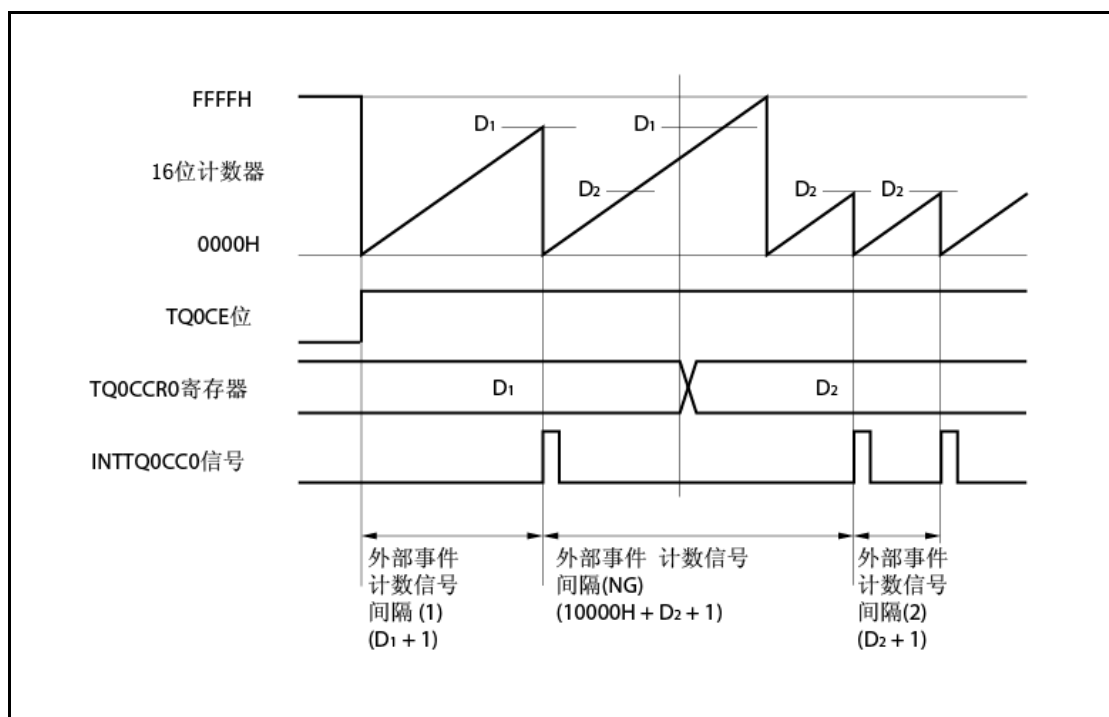
如果 TQ0CCR0 寄存器设为 FFFFH，每当检测到外部事件计数信号的有效沿时 16-位计数器计数到 FFFFH。16-位计数器与下一数完时序同步清为 0000H，且产生 INTTQ0CC0 信号。此时，不设置 TQ0OPT0.TQ0OVF 位。



### (b) 重写 TQ0CCR0 寄存器的注意事项

要将 TQ0CCR0 寄存器的值改为更小的值，停止计数一次且接着改变设置值。

如果计数期间 TQ0CCR0 寄存器的值重写为更小的值，则 16-位计数器可能溢出。



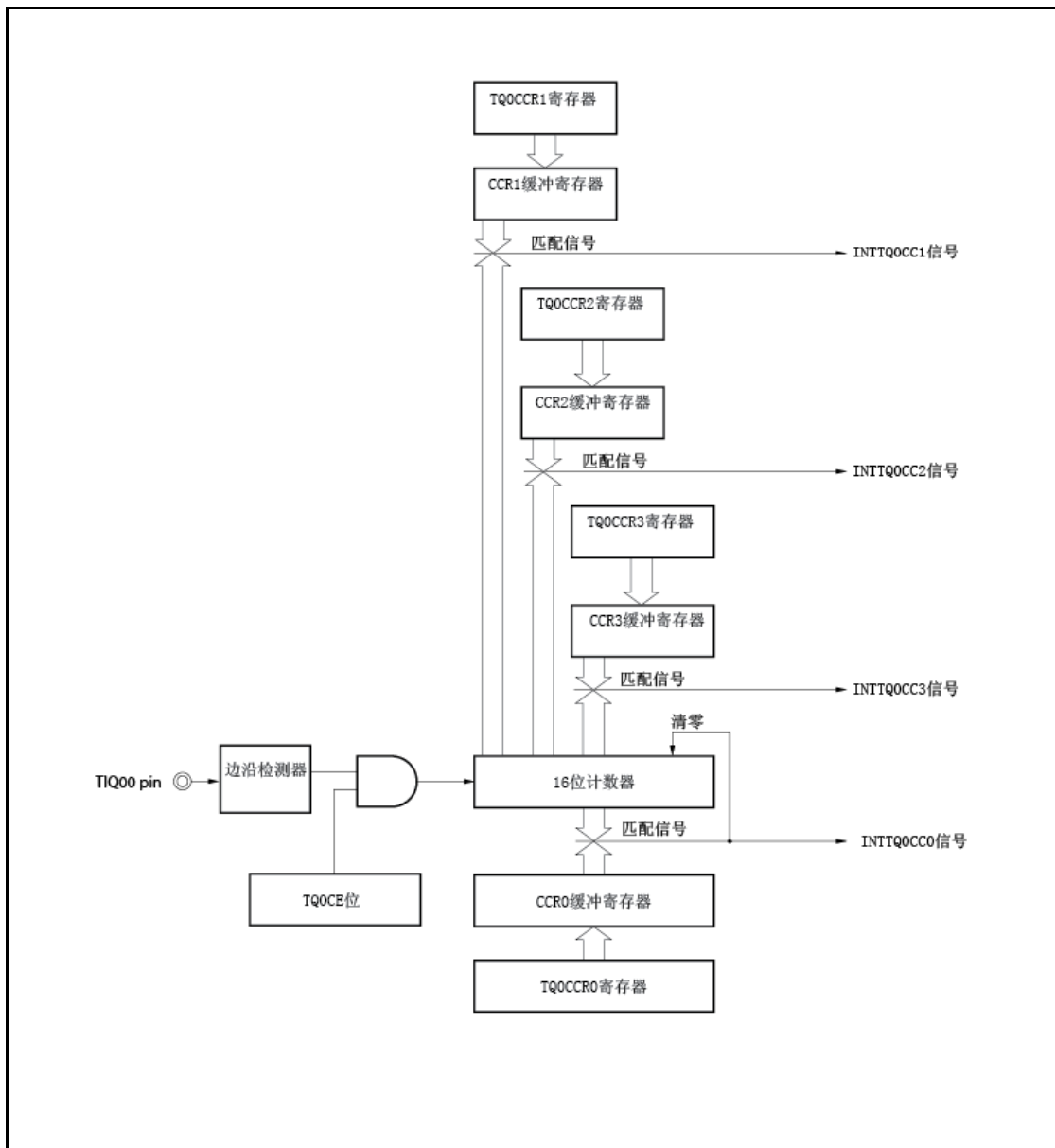
如果当计数值大于  $D_2$  而小于  $D_1$  时 TQ0CCR0 寄存器的值从  $D_1$  改为  $D_2$ ，则重写 TQ0CCR0 寄存器的同时计数值转移到 CCR0 缓冲寄存器。结果，与 16-位计数器比较的值为  $D_2$ 。

然而，因为计数值已超过  $D_2$ ，16-位计数器计数上至 FFFFH，溢出，且接着再次从 0000H 计数。当计数值匹配  $D_2$  时，产生 INTTQ0CC0 信号。

因此，在开始预计的有效沿计数“ $(D_1 + 1)$ 次”或“ $(D_2 + 1)$ 次”不能产生 INTTQ0CC0 信号，而可能在有效沿计数“ $(10000H + D_2 + 1)$ 次”产生。

(c) TQ0CCR1 到 TQ0CCR3 寄存器的操作

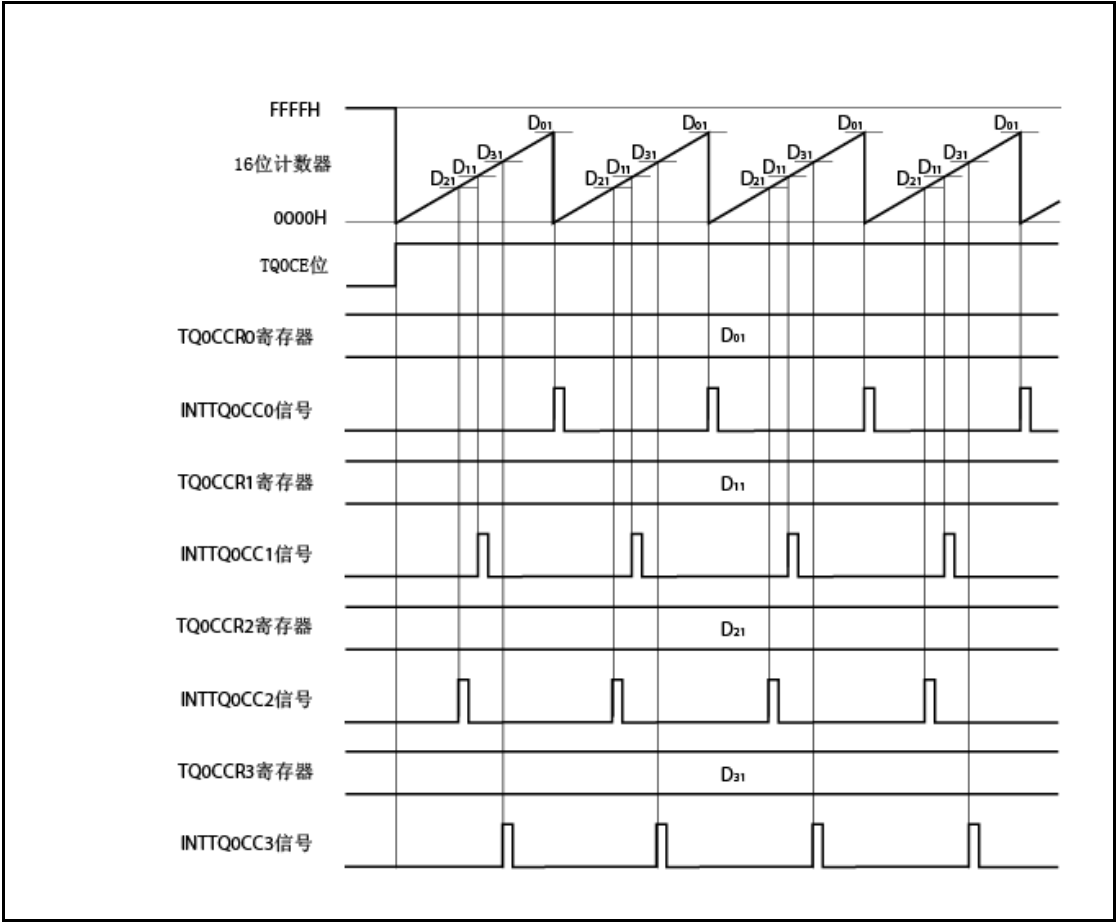
图 7-13. TQ0CCR1 到 TQ0CCR3 寄存器的配置



如果 TQ0CCRk 寄存器的设置值小于 TQ0CCR0 寄存器的设置值，则每个周期产生 INTTQ0CCk 信号一次。

备注      k = 1 到 3

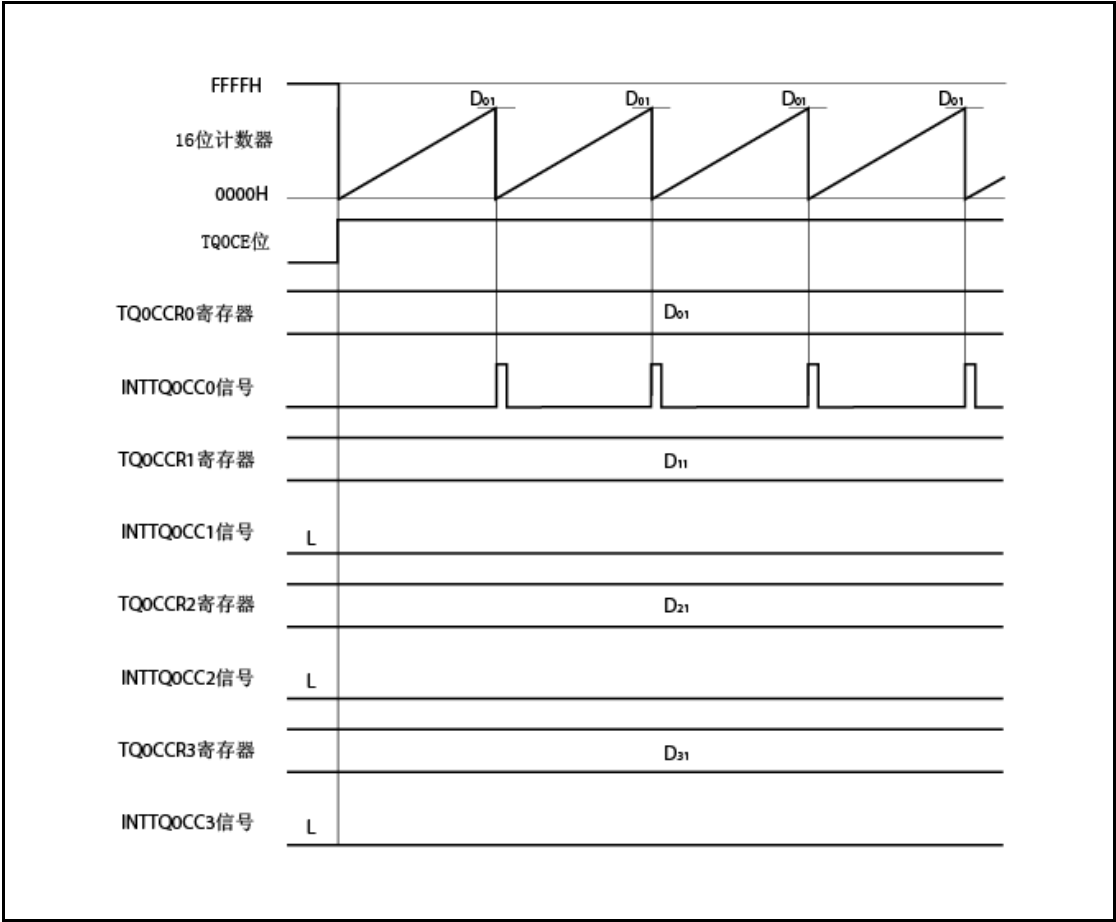
图 7-14. 当  $D_{01} \geq D_{k1}$  时的时序图



如果 TQ0CCRk 寄存器的设置值大于 TQ0CCR0 寄存器的设置值，则因为 16-位计数器的计数值与 TQ0CCRk 寄存器的值不匹配不产生 INTTQ0CCk 信号。

备注      k = 1 到 3

图 7-15. 当  $D_{01} < D_{k1}$  时的时序图



## 7.5.3 外部触发脉冲输出模式(TQ0MD2 到 TQ0MD0 位 = 010)

在外部触发脉冲输出模式下，当 TQ0CTL0.TQ0CE 位设为 1 时 16-位定时器/事件计数器 Q 等待触发。当检测到外部触发输入信号的有效沿时，16-位定时器/事件计数器 Q 开始计数，且从 TOQ01 到 TOQ03 引脚输出 PWM 波形。

脉冲还可通过产生软件触发而不是使用外部触发输出。当使用软件触发时，具有作为其周期一半的一周期 PWM 波形的方波还可从 TOQ00 引脚输出。

图 7-16. 在外部触发脉冲输出模式下的配置

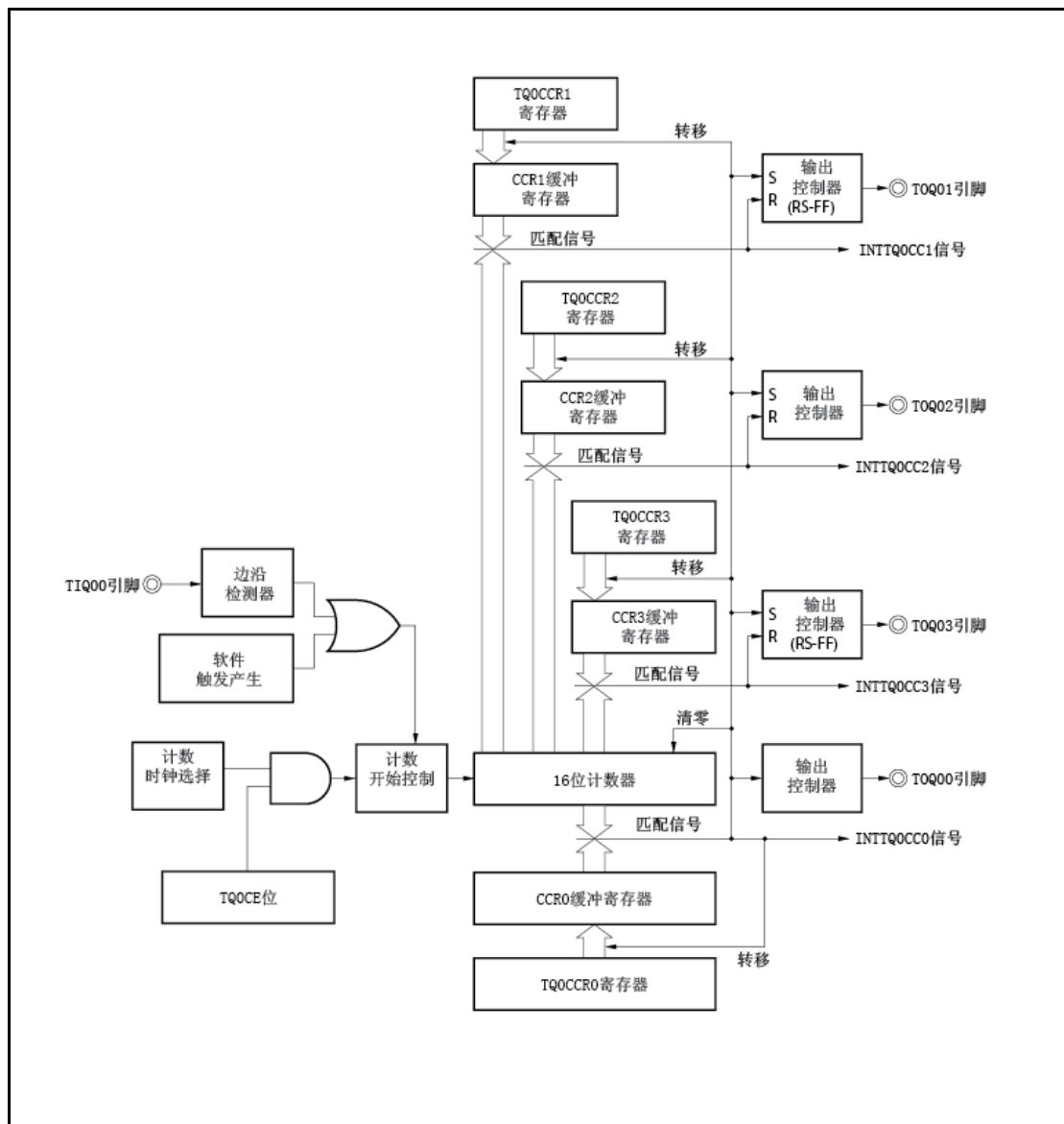
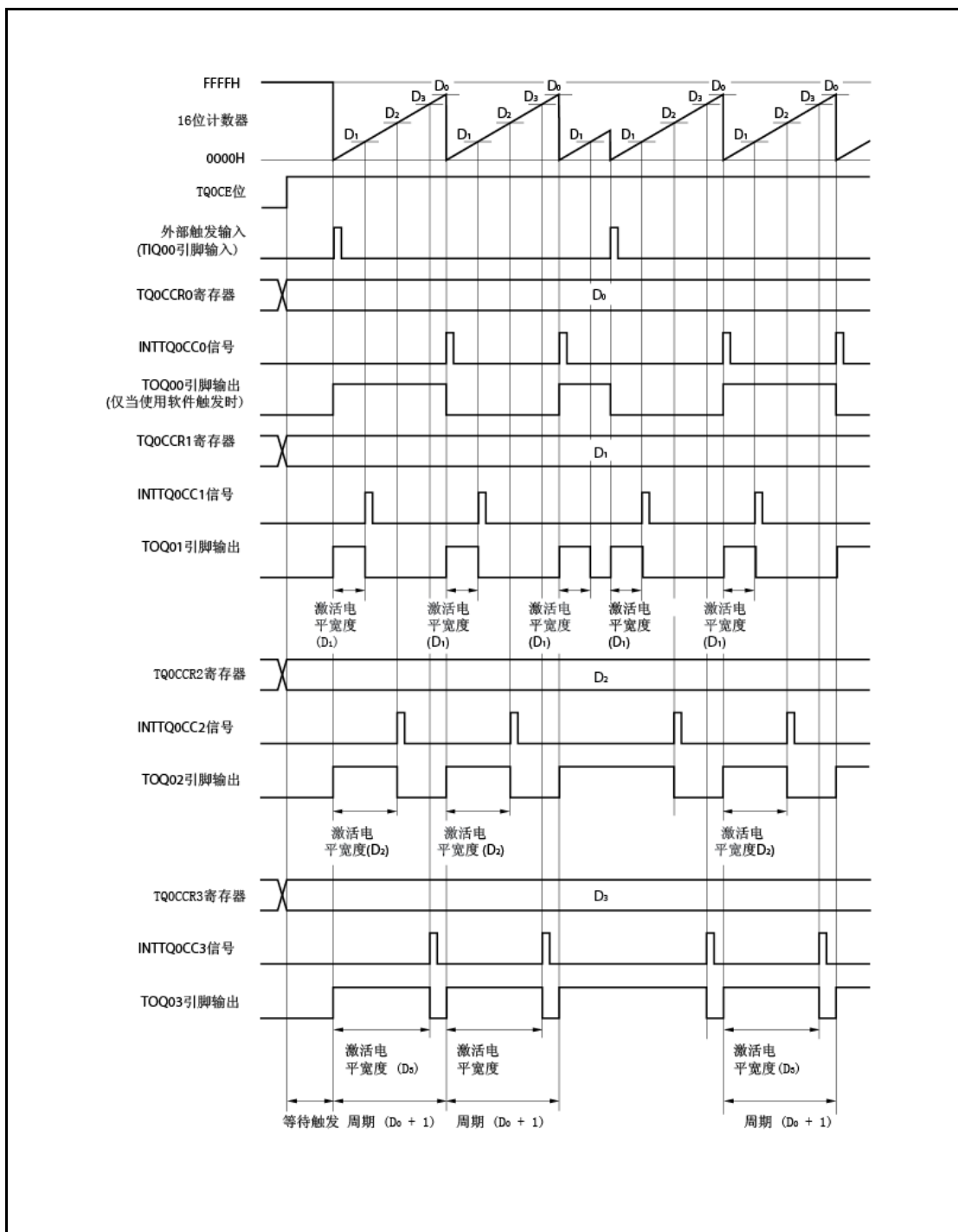


图 7-17. 在外部触发脉冲输出模式下的基本时序



当 TQ0CE 位设为 1 时 16-位定时器/事件计数器 Q 等待触发。发生触发时，16-位计数器从 FFFFH 清为 0000H，同时开始计数，且从 TOQ0k 引脚输出 PWM 波形。当计数器运行时如果再次发生触发，则计数器清为 0000H 并重新开始。( TOQ00 引脚的输出反相。TOQ0k 引脚输出高-电平而不考虑当发生触发时的状态(高/低)。)

PWM 波形的激活电平宽度，周期，及 占空因数可如下所示计算。

激活电平宽度=(TQ0CCRk 寄存器的设置值) ×计数时钟周期

周期 =(TQ0CCR0 寄存器的设置值+ 1) ×计数时钟周期

占空因数=(TQ0CCRk 寄存器的设置值)/(TQ0CCR0 寄存器的设置值+ 1)

当 16-位计数器的计数值匹配 CCR0 缓冲寄存器的值之后其进行下一次计数时产生比较匹配请求信号 INTTQ0CC0，且 16-位计数器清为 0000H。当 16-位计数器的计数值匹配 CCRk 缓冲寄存器的值时产生比较匹配中断请求信号 INTTQ0CCk。

当 16-位计数器的计数值匹配 CCR0 缓冲寄存器的值时，设至 TQ0CCRm 寄存器的值转移到 CCRm 缓冲寄存器且 16-位计数器清为 0000H。

外部触发输入信号的有效沿或将软件触发(TQ0CTL1.TQ0EST 位)设为 1 用作触发器。

备注 k = 1 到 3, m = 0 到 3

图 7-18. 在外部触发脉冲输出模式下寄存器的设置(1/3)





图 7-18. 在外部触发脉冲输出模式下寄存器的设置(2/3)

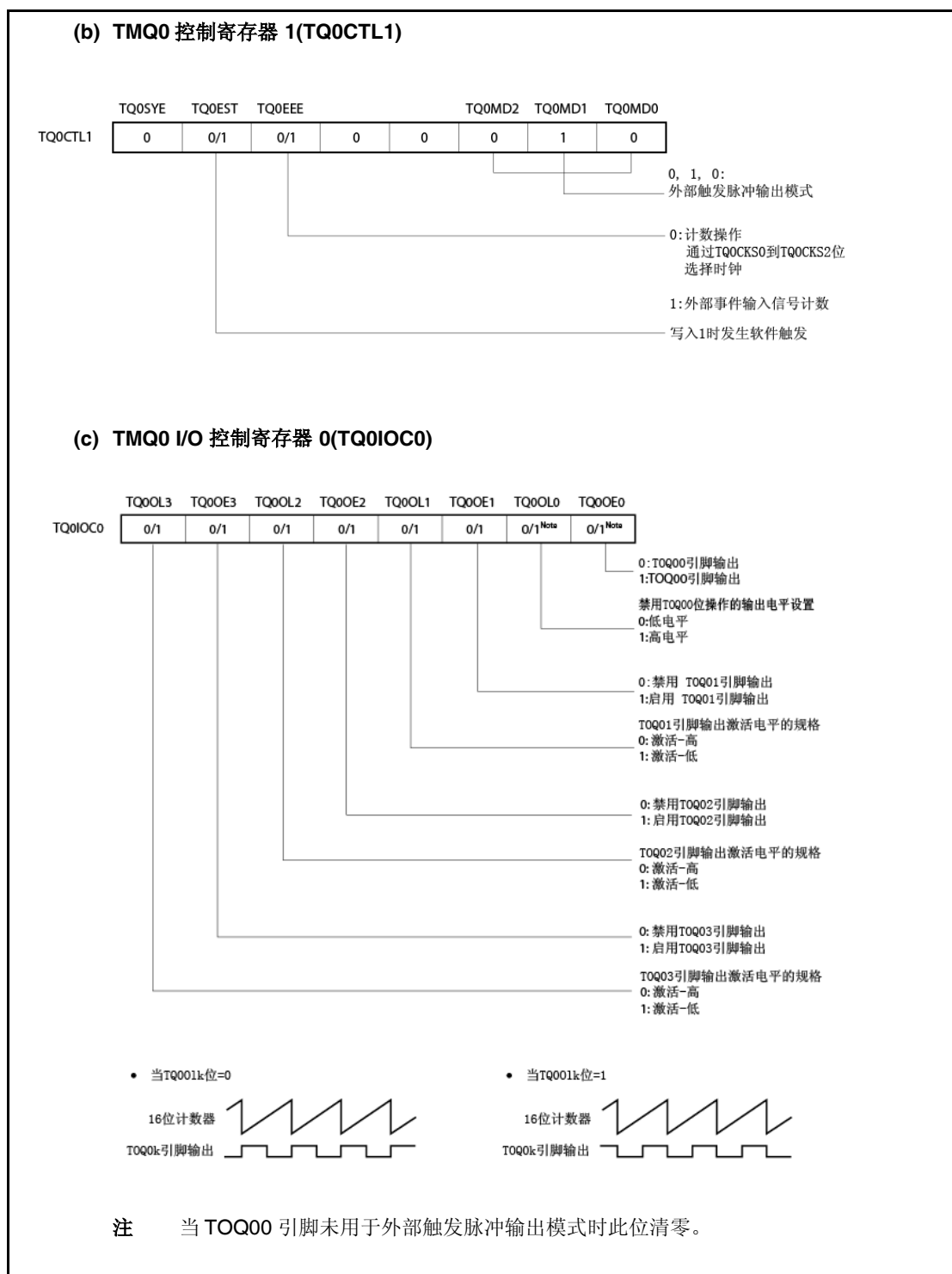
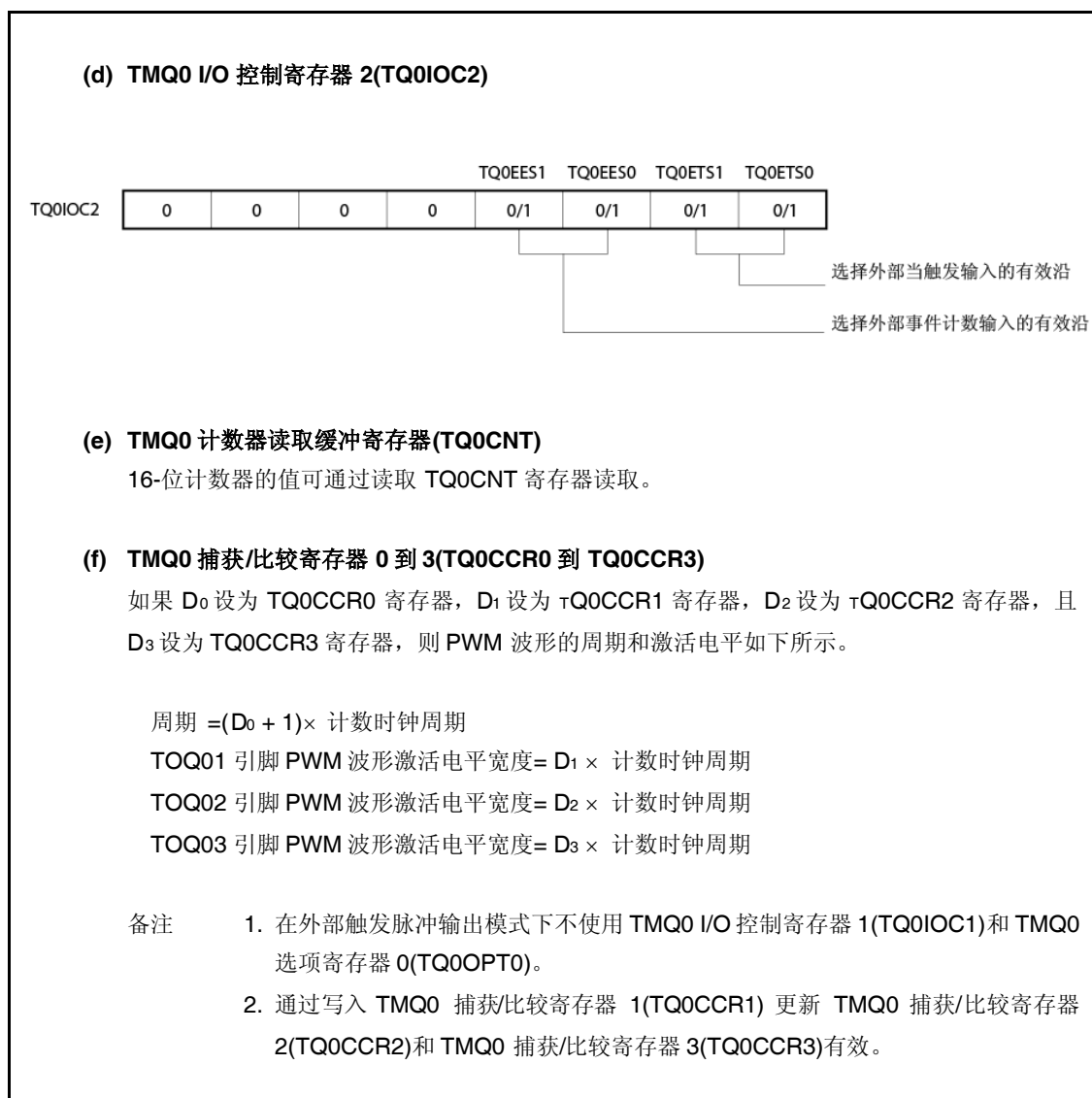
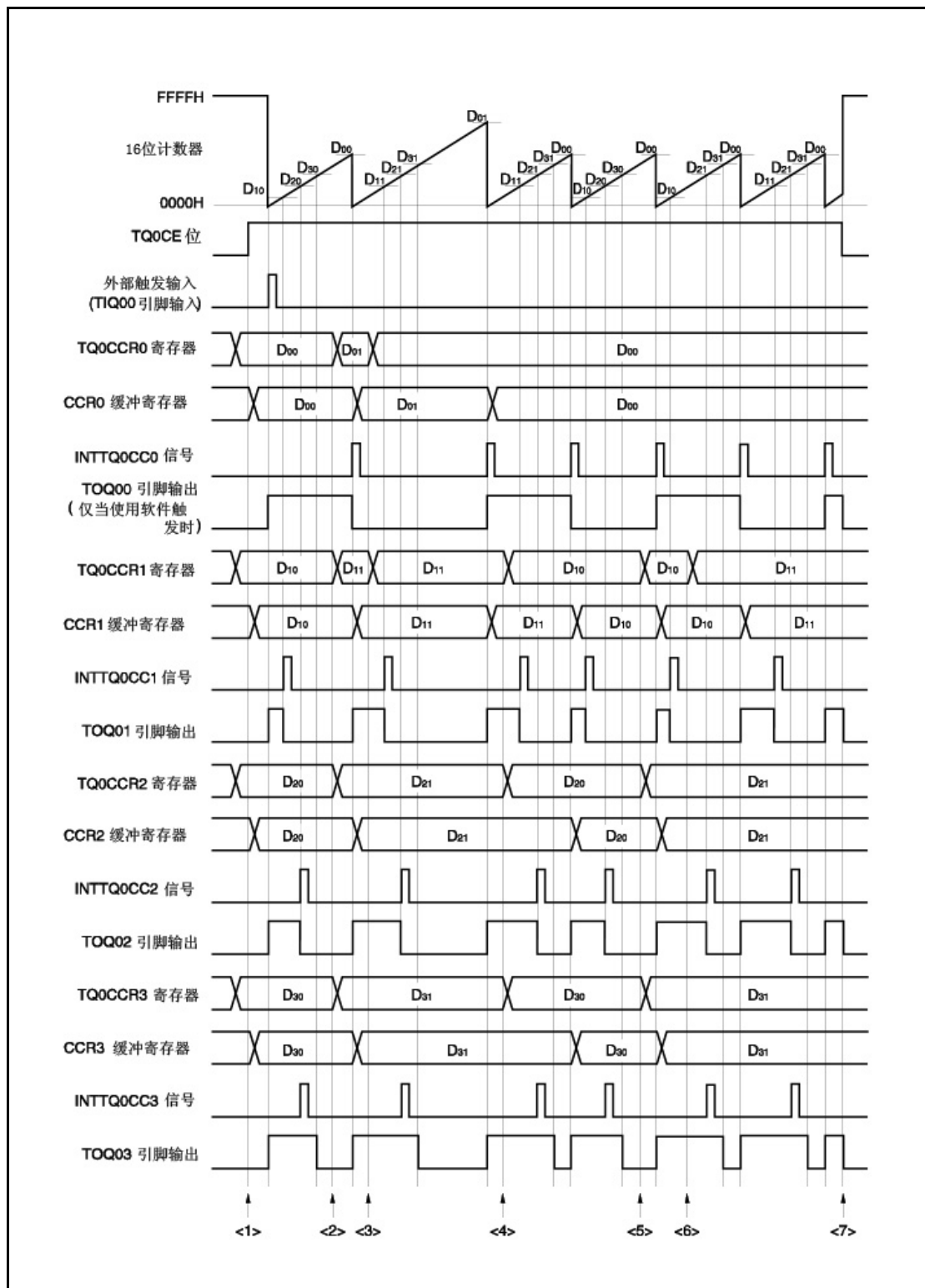
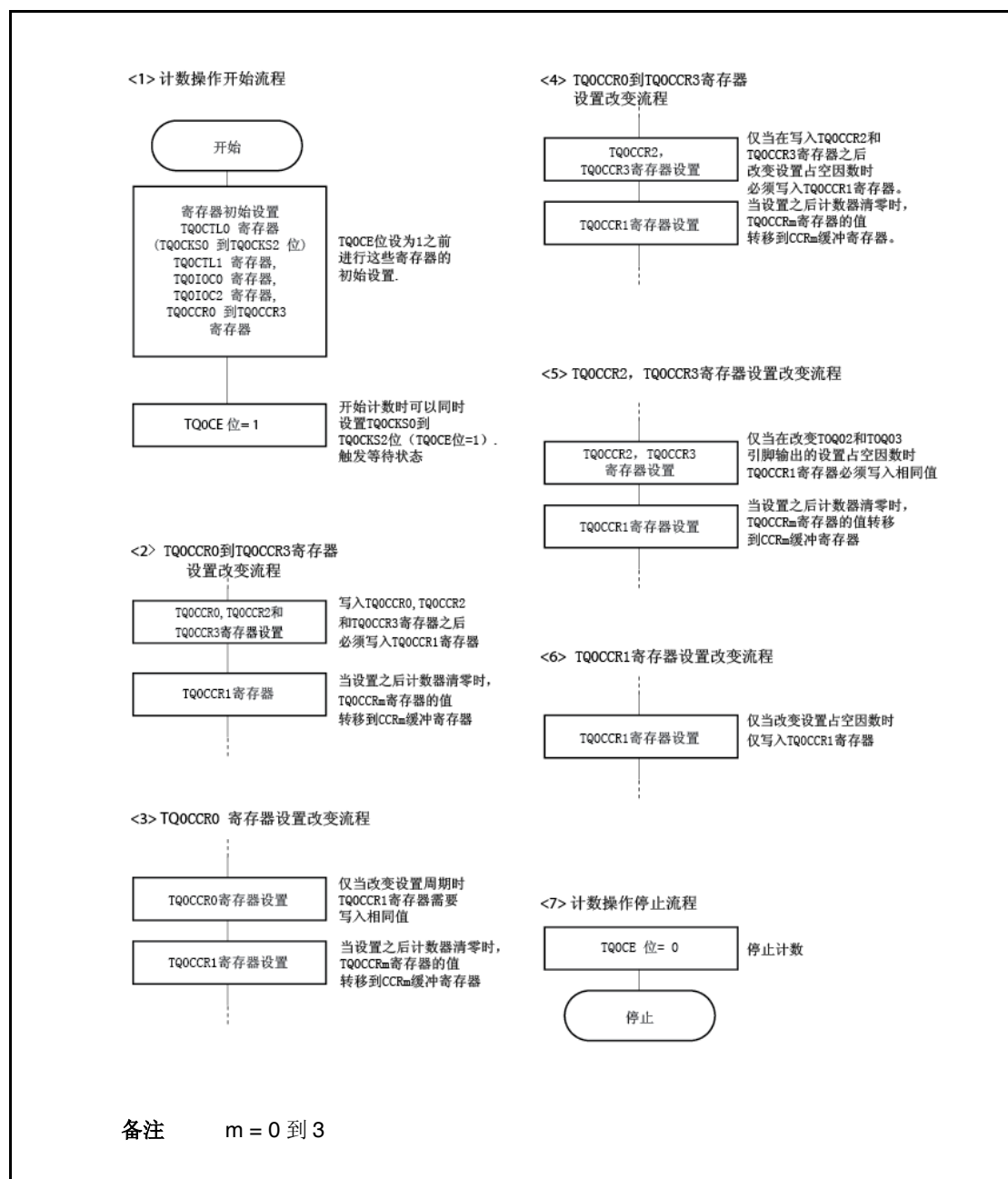


图 7-18. 在外部触发脉冲输出模式下寄存器的设置(3/3)





用户手册 U17719CA2V0UD

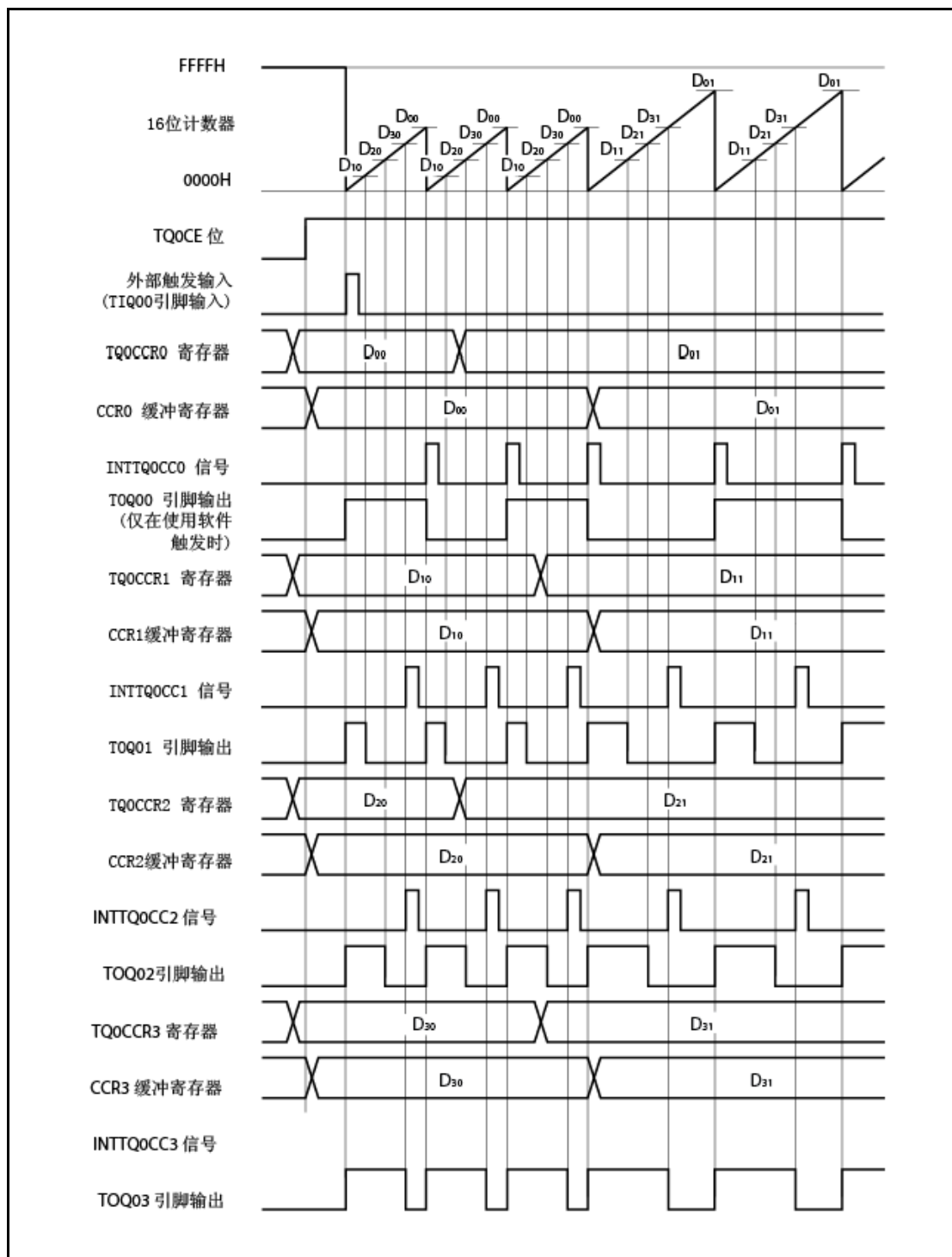


## (2) 外部触发脉冲输出模式运行时序

### (a) 运行期间改变脉宽的注意事项

当计数器运行时要改变 PWM 波形，最后请写入 TQ0CCR1 寄存器。

在检测到 INTTQ0CC0 信号之后写入 TQ0CCR1 寄存器之后重写 TQ0CCRk 寄存器。



为了从 TQ0CCRm 寄存器将数据传输到 CCRm 缓冲寄存器，必须写入 TQ0CCR1 寄存器。  
此时要改变 PWM 波形的周期和激活电平宽度，首先设置 TQ0CCR0 寄存器的周期，设置 TQ0CCR2 和 TQ0CCR3 寄存器的激活电平宽度，且接着设置 TQ0CCR1 寄存器的激活电平。  
仅要改变 PWM 波形周期，首先设置 TQ0CCR0 寄存器的周期，且接着 TQ0CCR1 寄存器写入相同值。

仅要改变 PWM 波形的激活电平宽度(占空因数)，首先设置 TQ0CCR2 和 TQ0CCR3 寄存器的激活电平且接着设置 TQ0CCR1 寄存器的激活电平。

仅要改变 TOQ01 引脚 PWM 波形输出的激活电平宽度(占空因数)，仅设置 TQ0CCR1 寄存器。

仅要改变 TOQ02 和 TOQ03 引脚 PWM 波形输出的激活电平宽度(占空因数)，首先设置 TQ0CCR2 和 TQ0CCR3 寄存器的激活电平宽度，且接着 TQ0CCR1 寄存器写入相同值。

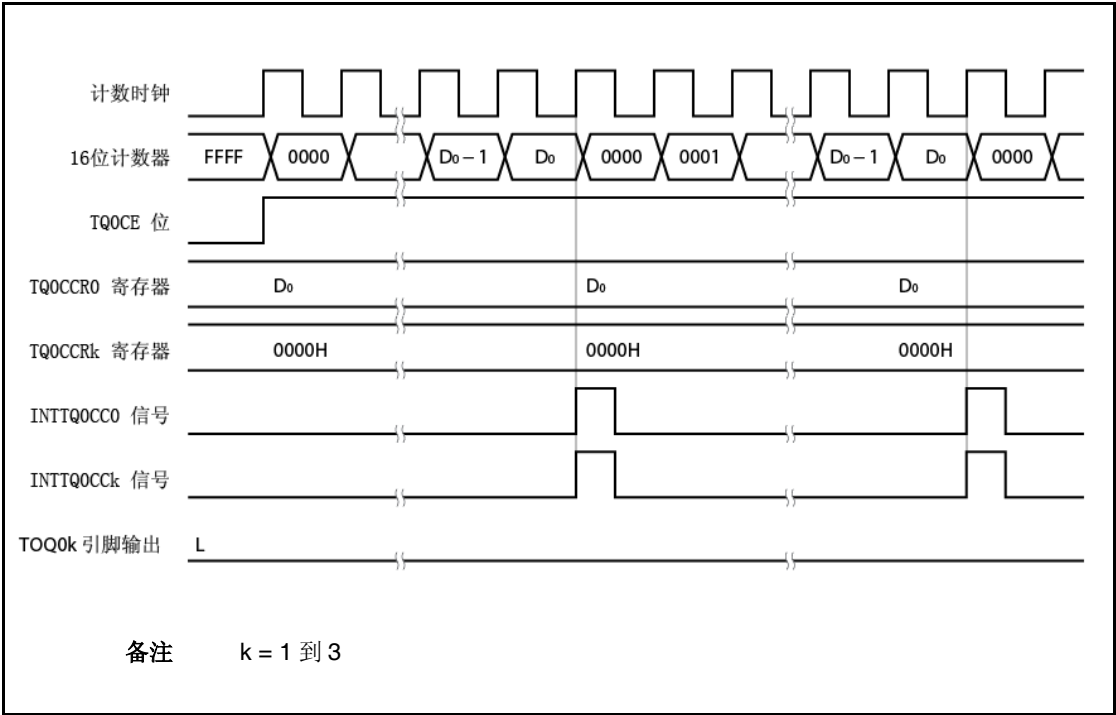
数据写入 TQ0CCR1 寄存器之后，写入 TQ0CCRm 寄存器的值与 16-位计数器清零同步转移到 CCRm 缓冲寄存器，且用作与 16-位计数器比较的值。

写入 TQ0CCR1 寄存器一次之后要再次写入 TQ0CCR0 到 TQ0CCR3 寄存器，在产生 INTTQ0CC0 信号之后进行。否则，因为数据从 TQ0CCRm 寄存器传输到 CCRm 缓冲寄存器的时序与写入 TQ0CCRm 寄存器冲突，所以可能未定义 CCRm 缓冲寄存器的值。

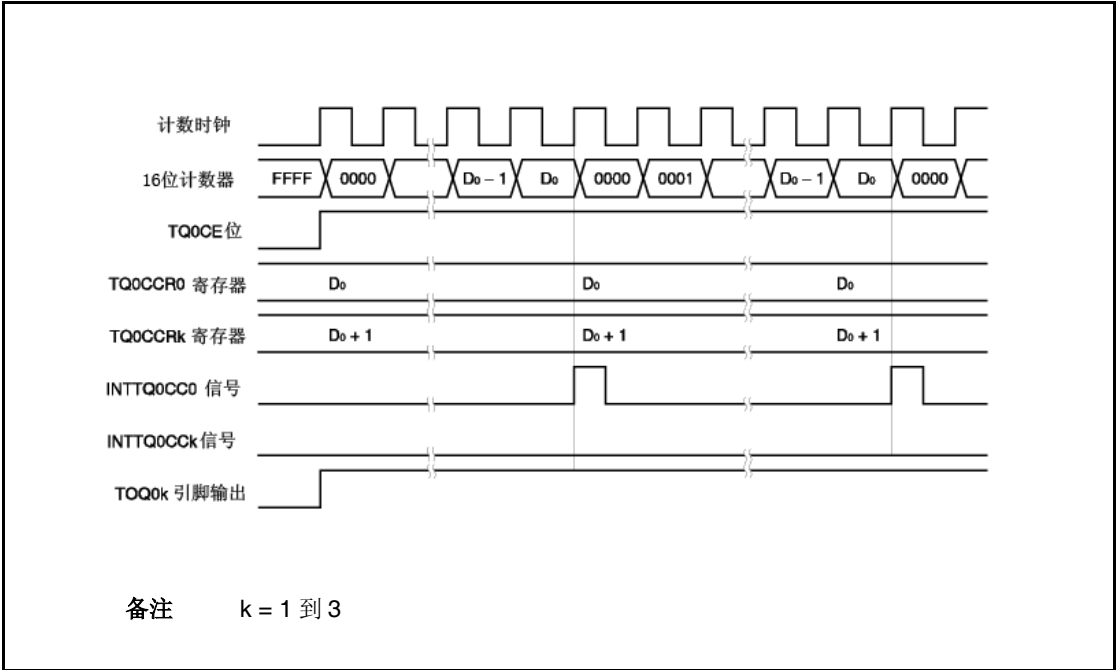
**备注**      m = 0 到 3

(b) PWM 波形的 0%/100% 输出

要输出 0% 波形，将 TQ0CCRk 寄存器设置为 0000H。如果 TQ0CCR0 寄存器的设置值为 FFFFH，则定期产生 INTTQ0CCk 信号。

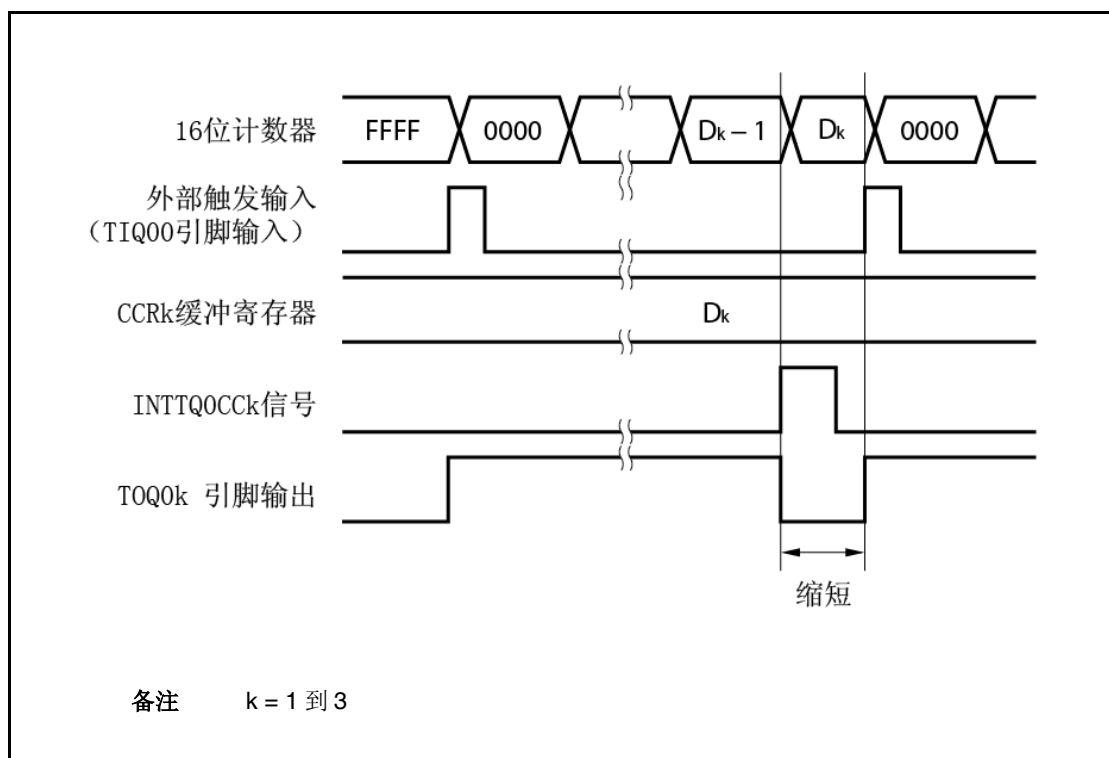


要输出 100% 波形，设置 TQ0CCRk 寄存器为值(TQ0CCR0 寄存器的设置值+ 1)。如果 TQ0CCR0 寄存器的设置值为 FFFFH，则不能产生 100% 输出。

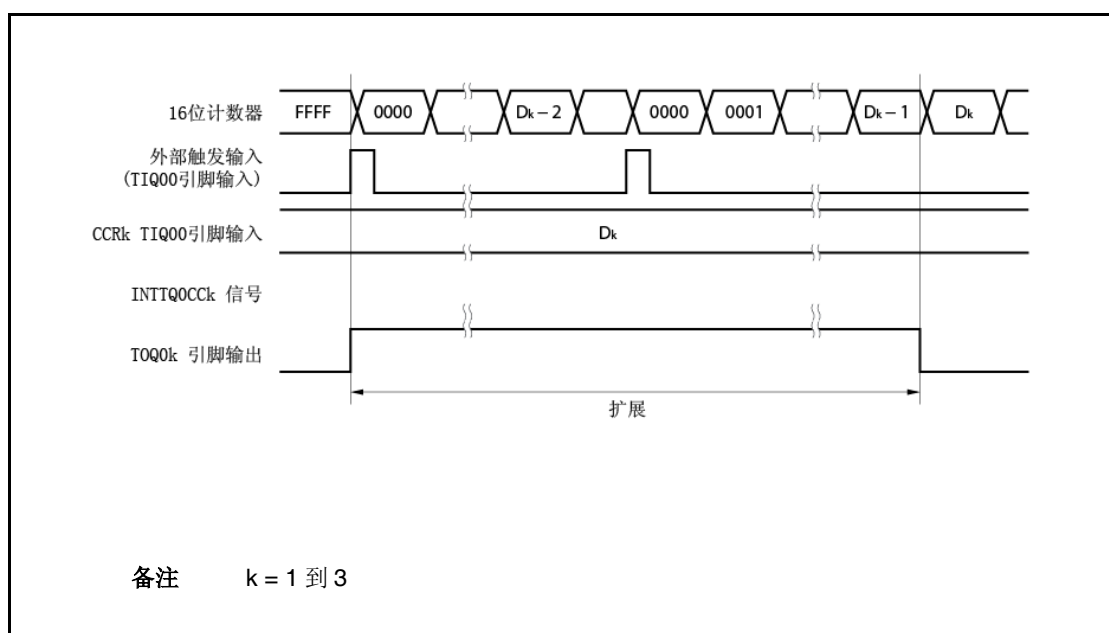


(c) 触发检测和与 CCRk 缓冲寄存器匹配之间的冲突

如果 INTTQ0CCk 信号产生之后立即检测触发器，则 16-位计数器立即清为 0000H，断定 TOQ0k 引脚的输出信号，且计数器继续计数。结果，PWM 波形的非活动周期缩短。



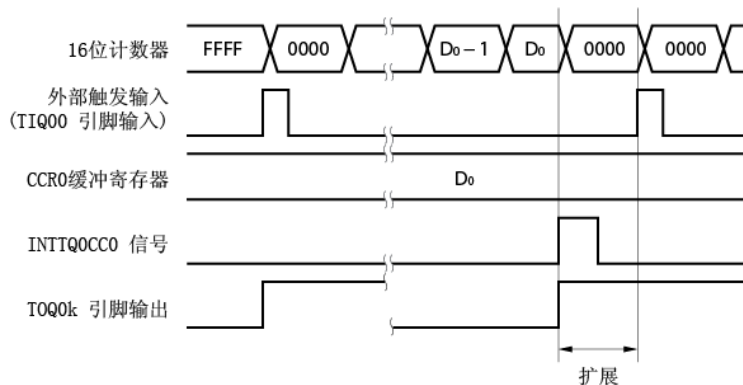
如果 INTTQ0CCk 信号产生之前立即检测触发器，则不产生 INTTQ0CCk 信号，且 16-位计数器清为 0000H 并继续计数。TOQ0k 引脚的输出信号保持激活。结果，PWM 波形的激活周期扩展。





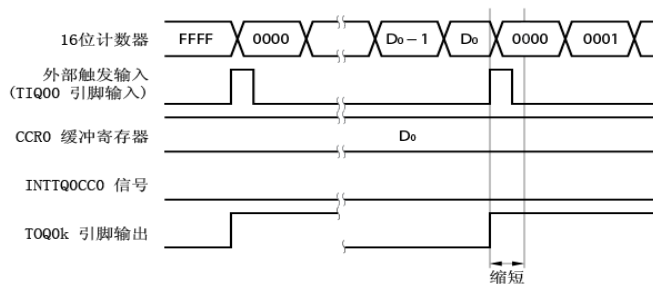
(d) 触发检测和与 CCR0 缓冲寄存器匹配之间的冲突

如果 INTTQ0CC0 信号产生之后立即检测触发器，则 16-位计数器清为 0000H 并继续计数。因此，TOQ0k 引脚的激活周期按从 INTTQ0CC0 信号产生到触发检测的时间扩展。



备注 k = 1 到 3

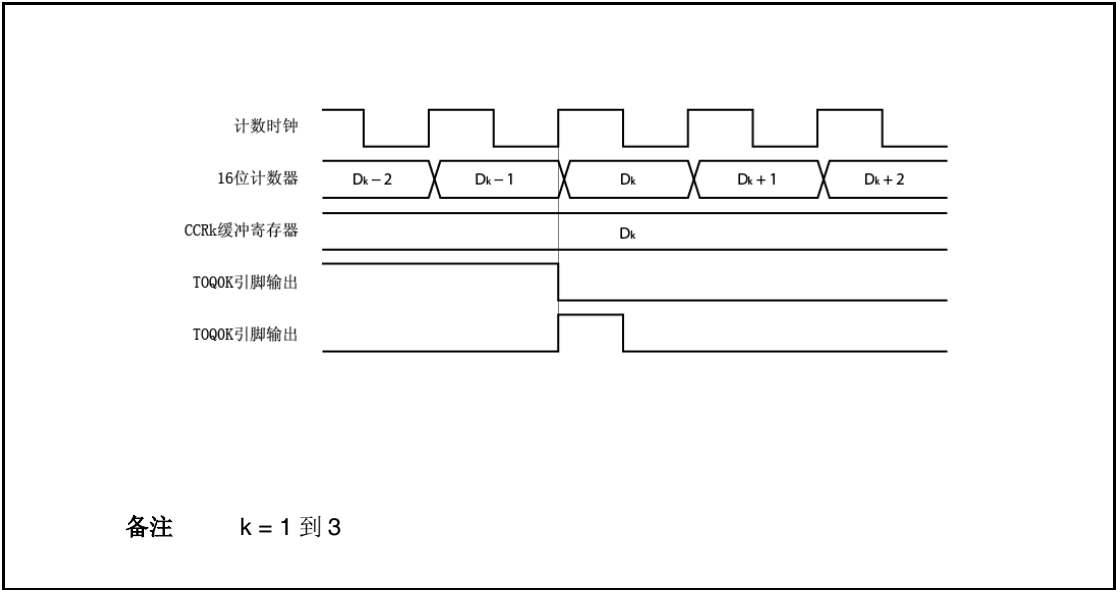
如果 INTTQ0CC0 信号产生之前立即检测触发器，则不产生 INTTQ0CC0 信号。16-位计数器清为 0000H，断定 TOQ0k 引脚，且计数器继续计数。结果，PWM 波形的非活动周期缩短。



备注 k = 1 到 3

(e) 比较匹配中断请求信号(INTTQ0CCK)的产生时序

在外部触发脉冲输出模式下 INTTQ0CCK 信号的产生时序与其他 INTTQ0CCK 信号时序不同；当 16-位计数器的计数值匹配 CCRk 缓冲寄存器的值时产生 INTTQ0CCK 信号。



通常，16-位计数器的计数值匹配 CCRk 缓冲寄存器的值之后 INTTQ0CCK 信号与下一次计数同步产生。

然而，在外部触发脉冲输出模式下，其较早产生一时钟脉冲。这是因为时序改为匹配改变 TOQ0k 引脚的输出信号的时序。

#### 7.5.4 单触发脉冲输出模式(TQ0MD2 到 TQ0MD0 位 = 011)

在单触发脉冲输出模式下，当 TQ0CTL0.TQ0CE 位设为 1 时 16-位定时器/事件计数器 Q 等待触发。当检测到外部触发输入的有效沿时，16-位定时器/事件计数器 Q 开始计数，且从 TOQ01 到 TOQ03 引脚输出单触发脉冲。

代替外部触发，还可产生软件触发以输出脉冲。当使用软件触发时，TOQ00 引脚输出激活电平同时 16-位计数器计数，且当计数器停止(等待触发)时输出非活动电平。

图 7-20. 在单触发脉冲输出模式下的配置

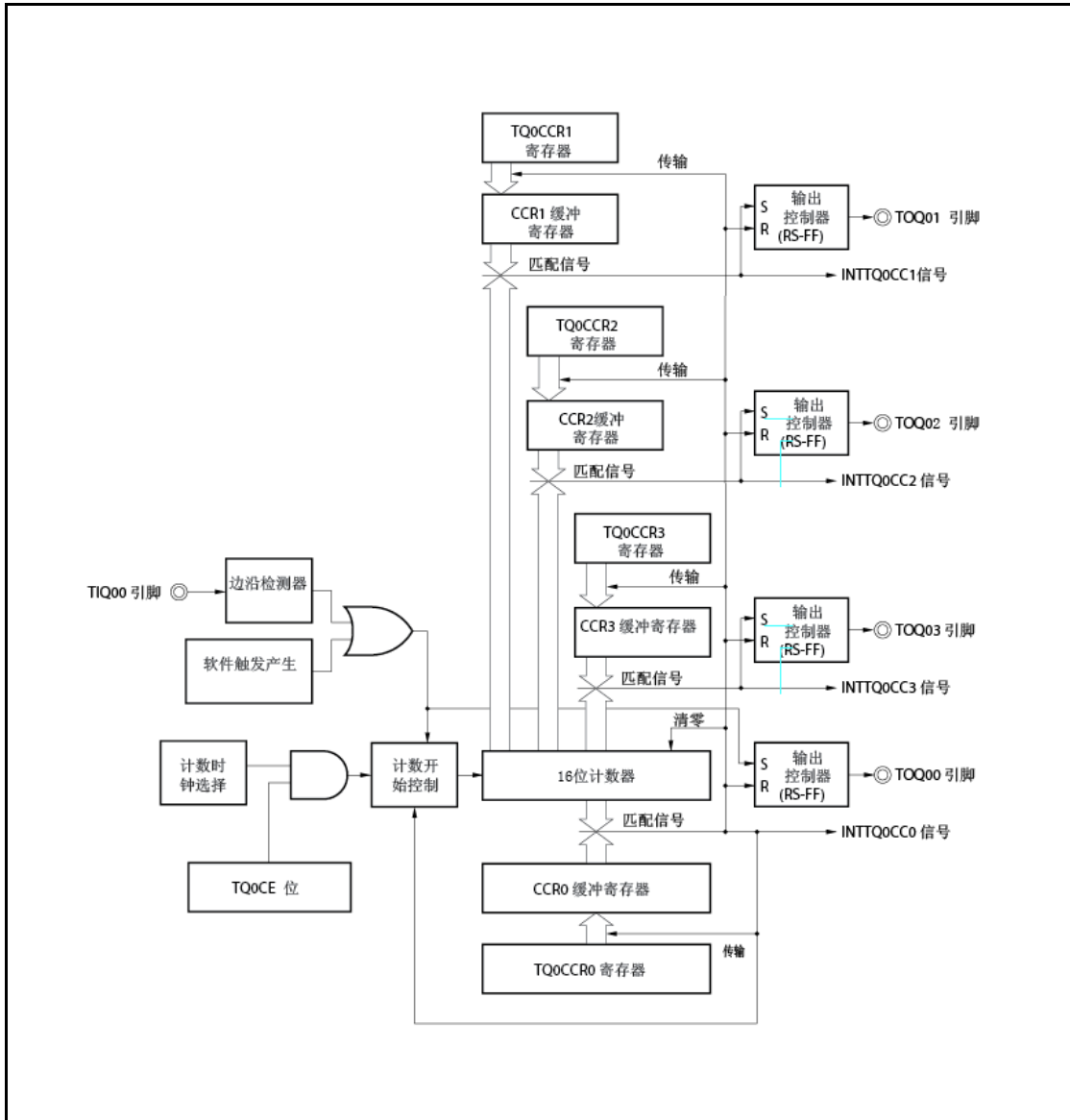
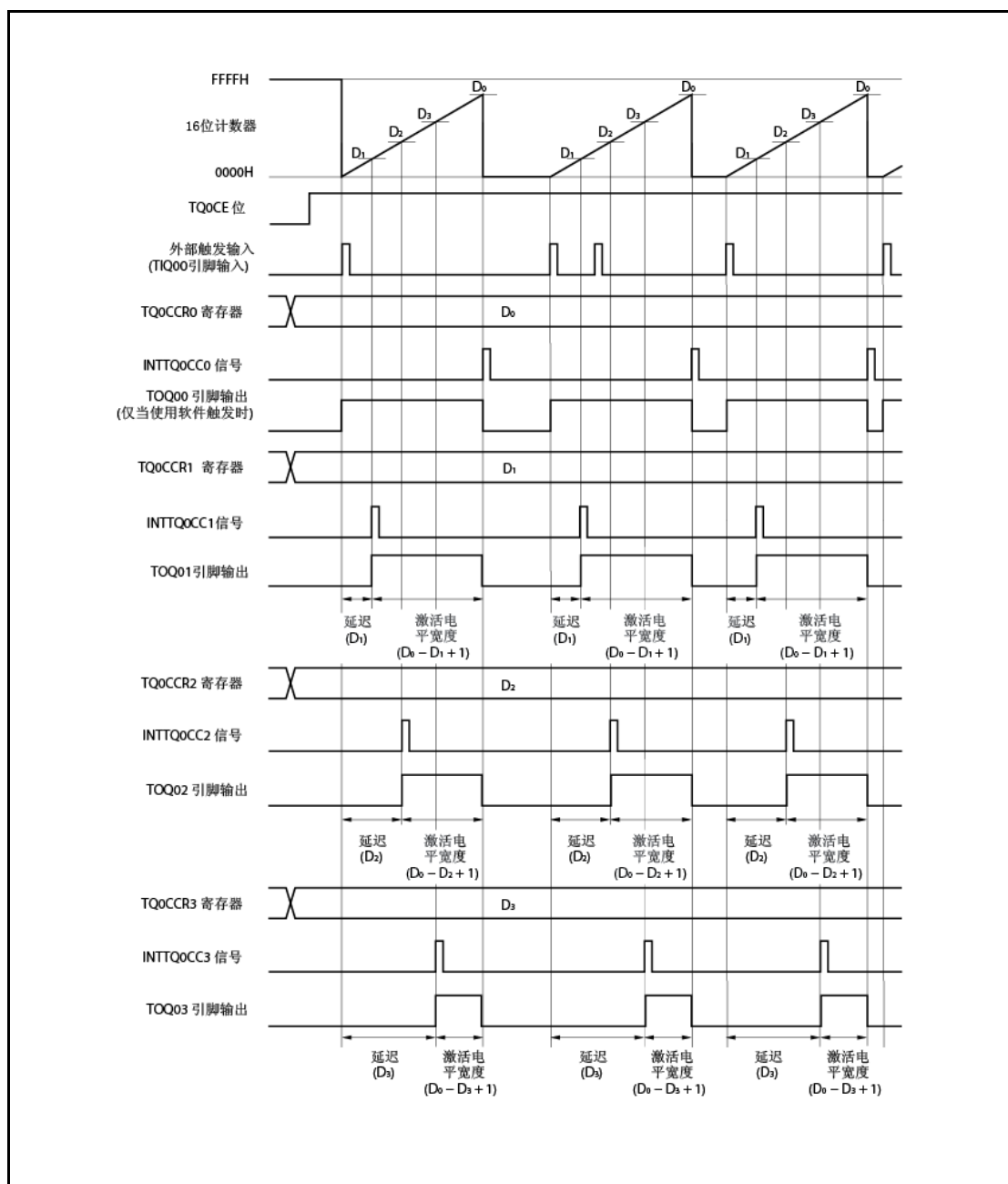


图 7-21. 在单触发脉冲输出模式下的基本时序



当 TQ0CE 位设为 1，16-位定时器/事件计数器 Q 等待触发。当产生触发时，16-位计数器从 FFFFH 清零 000H，开始计数，且从 TOQ0k 引脚输出单触发脉冲。输出单触发脉冲之后，16-位计数器设为 FFFFH，停止计数，且等待触发。当输出单触发脉冲时如果再次产生触发，则忽略。

单触发脉冲的输出延迟周期和激活电平宽度可如下所示进行计算。

输出延迟周期=(TQ0CCRk 寄存器的设置值) ×计数时钟周期

激活电平宽度=(TQ0CCR0 寄存器的设置值-TQ0CCRk 寄存器的设置值+ 1) × 计数时钟周期

16- 位计数器计数值匹配 CCR0 缓冲寄存器的值之后当其计数时产生比较匹配中断请求信号 INTTQ0CC0。当 16-位计数器的计数值匹配 CCRk 缓冲寄存器的值时产生比较匹配中断请求信号 INTTQ0CCK。

外部触发输入的有效沿或设置软件触发(TQ0CTL1.TQ0EST 位)为 1 用作触发。

备注      k = 1 到 3

图 7-22. 在单触发脉冲输出模式下寄存器的设置(1/3)

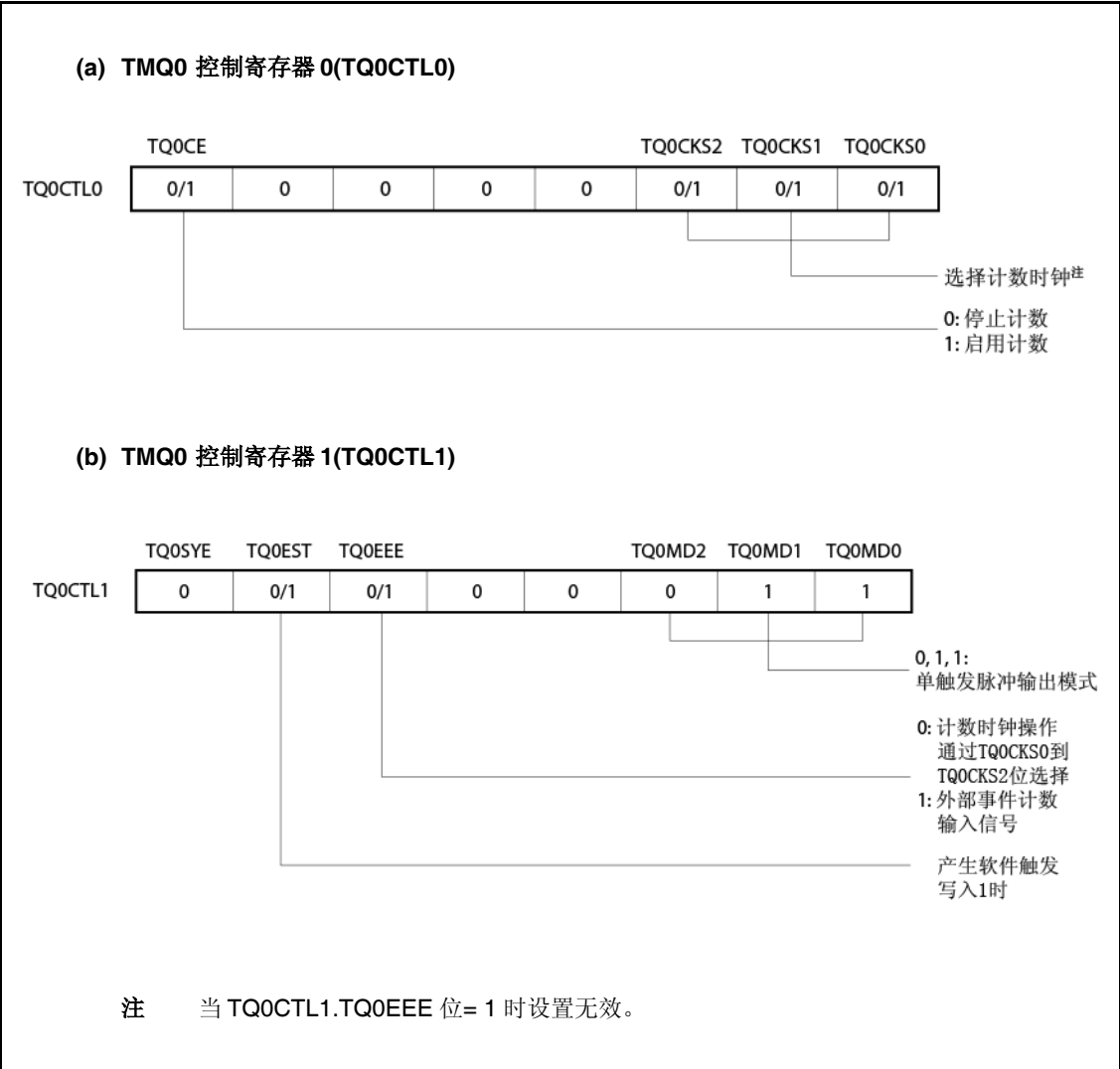


图 7-22. 在单触发脉冲输出模式下寄存器的设置(2/3)

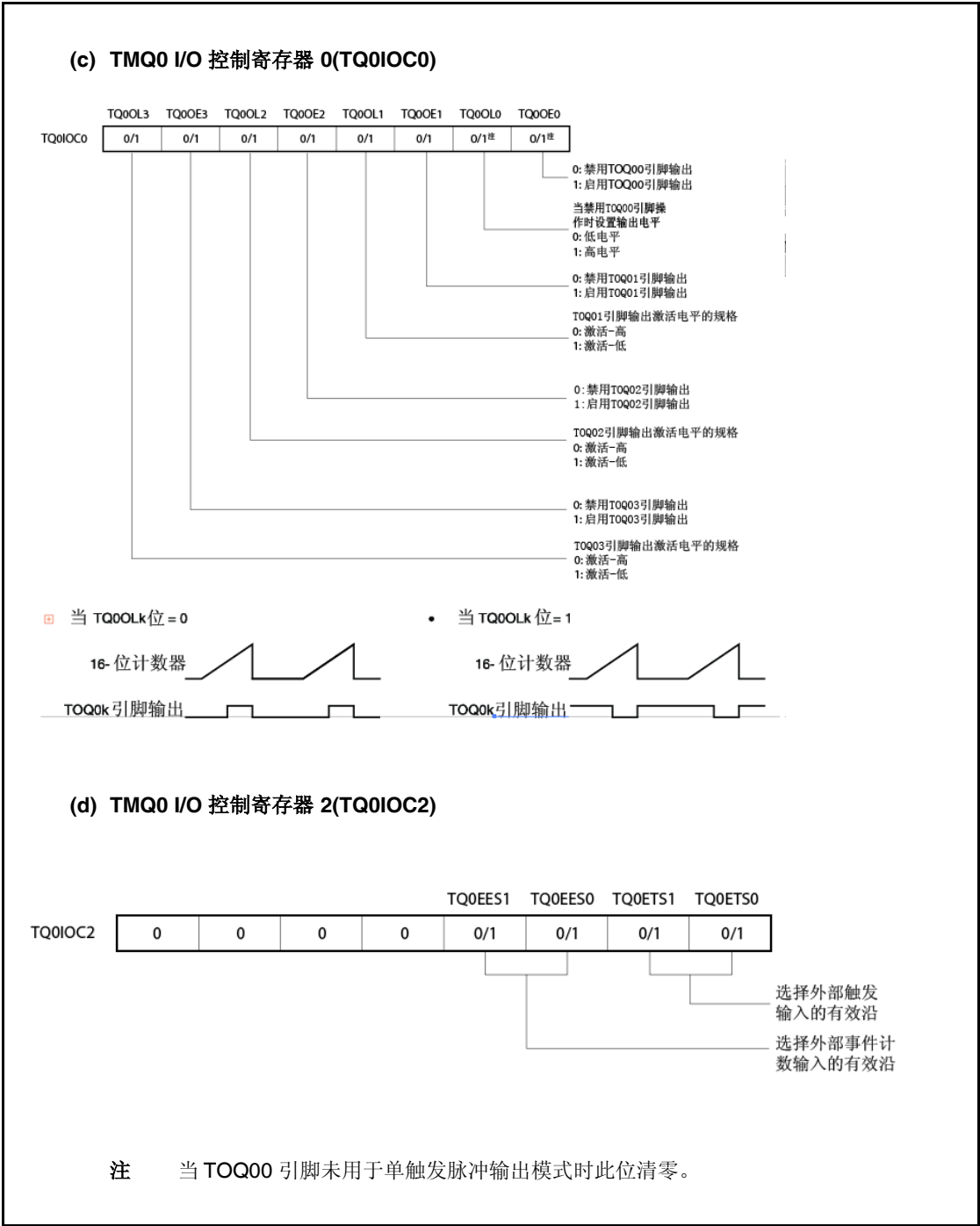


图 7-22.寄存器设置在单触发脉冲输出模式下(3/3)

(e) **TMQ0 计数器读取缓冲寄存器(TQ0CNT)**

16-位计数器的值可通过读取 TQ0CNT 寄存器来读取。

(f) **TMQ0 捕获/比较寄存器 0 到 3(TQ0CCR0 到 TQ0CCR3)**

如果 D0 设为 TQ0CCR0 寄存器且 Dk 设为 TQ0CCRk 寄存器，单触发脉冲的激活电平宽度和输出延迟周期如下所示。

激活电平宽度= $(D0 - Dk + 1) \times$  计数时钟周期

输出延迟周期= $(Dk) \times$ 计数时钟周期

**注意事项** 如果 **TQ0CCRk** 寄存器的设置值大于 **TQ0CCR0** 寄存器的值，即使在单触发脉冲输出模式下也不输出单触发脉冲。

- 备注**
- 1. 在单触发脉冲输出模式下不使用 **TMQ0 I/O 控制寄存器 1(TQ0IOC1)**和 **TMQ0 选项寄存器 0(TQ0OPT0)**。
  - 2.  $k = 1$  到  $3$

<R>

(1) 在单触发脉冲输出模式下的操作流程

图 7-23. 在单触发脉冲输出模式下的软件处理流程(1/2)

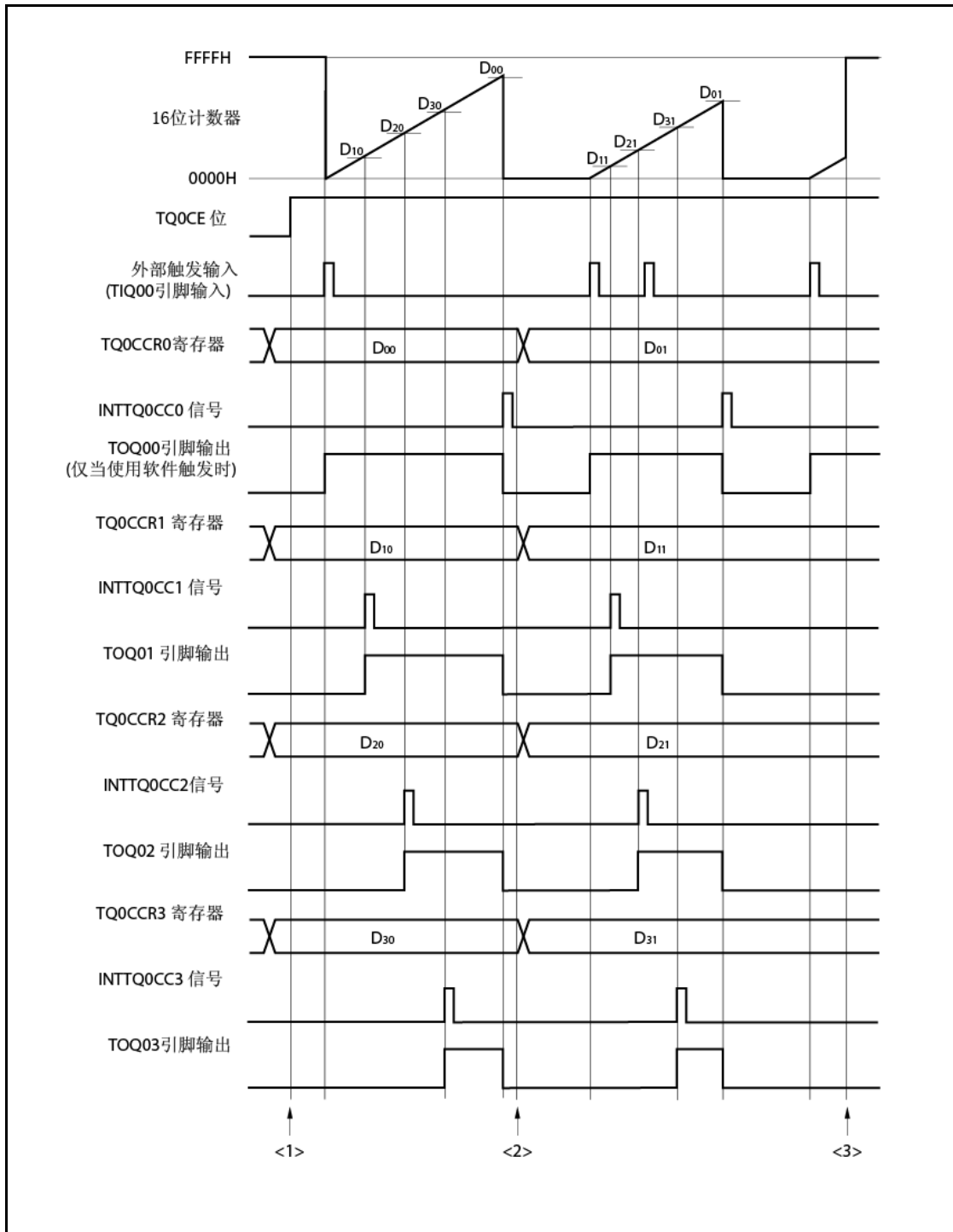
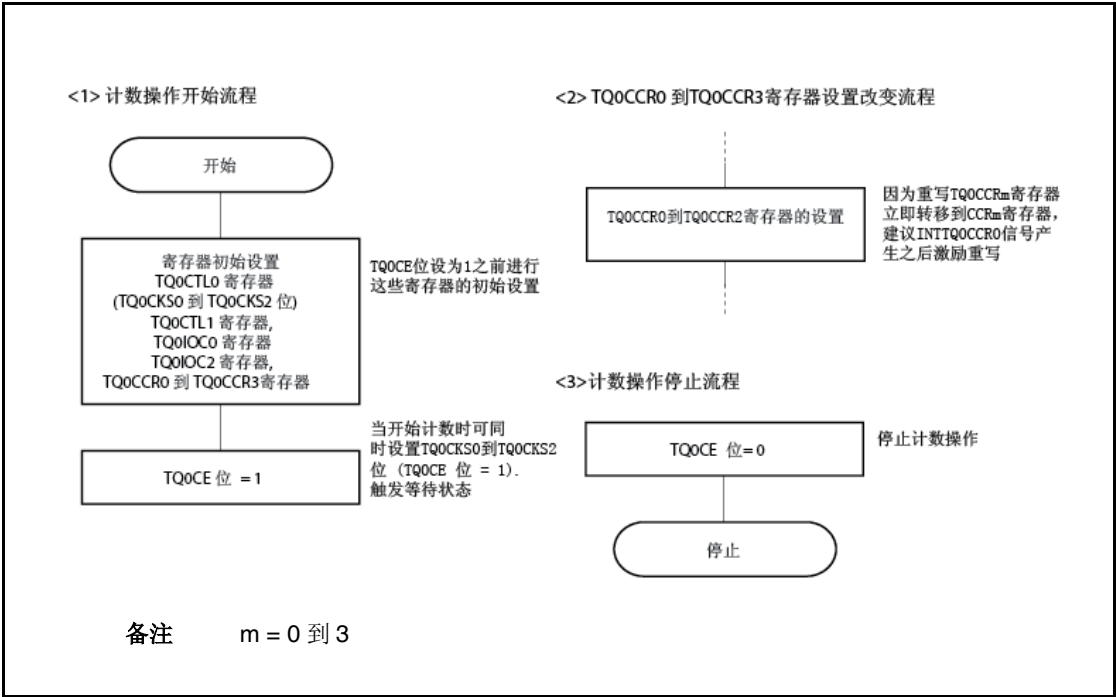




图 7-23. 在单触发脉冲输出模式下的软件处理流程(2/2)

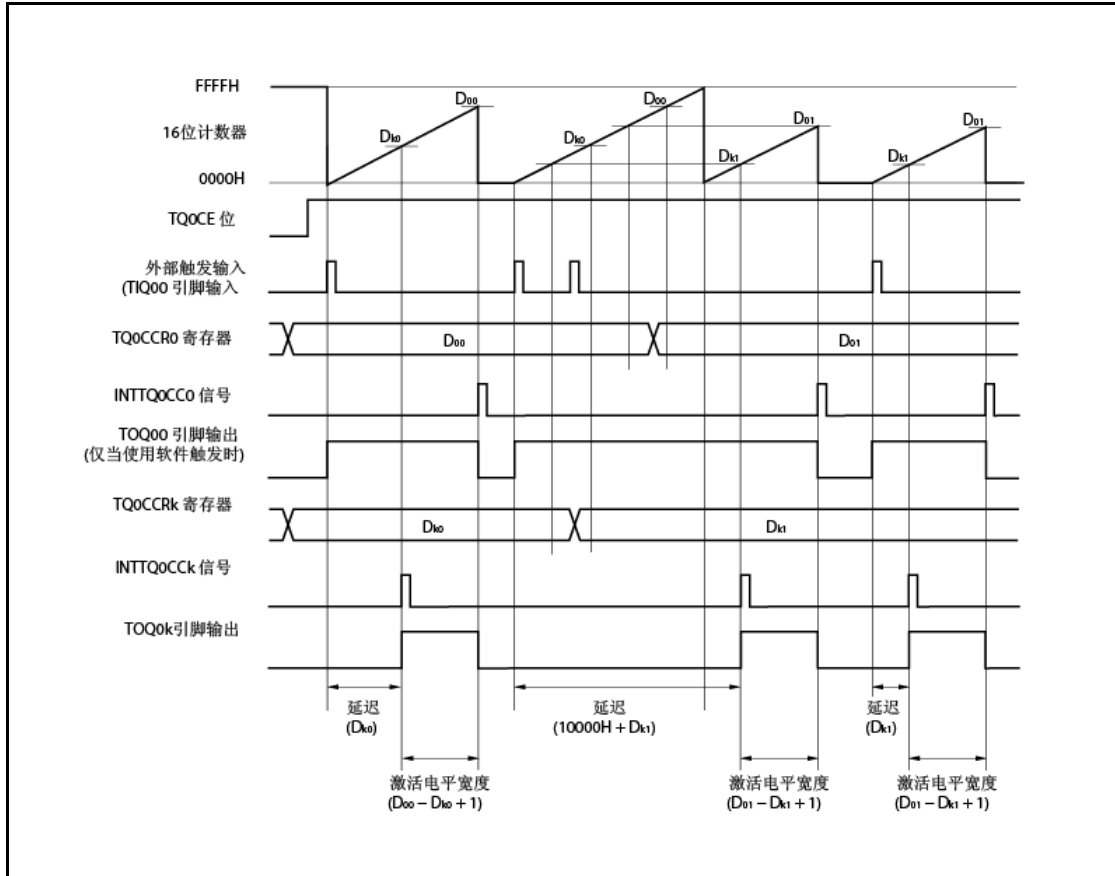


(2) 在单触发脉冲输出模式下的操作时序

(a) 重写 TQ0CCRm 寄存器的注意事项

要 TQ0CCRm 寄存器的设置值改为更小的值，停止计数一次，且接着改变设置值。

如果计数期间 TQ0CCR0 寄存器的值重写为更小的值，则 16-位计数器可能溢出。



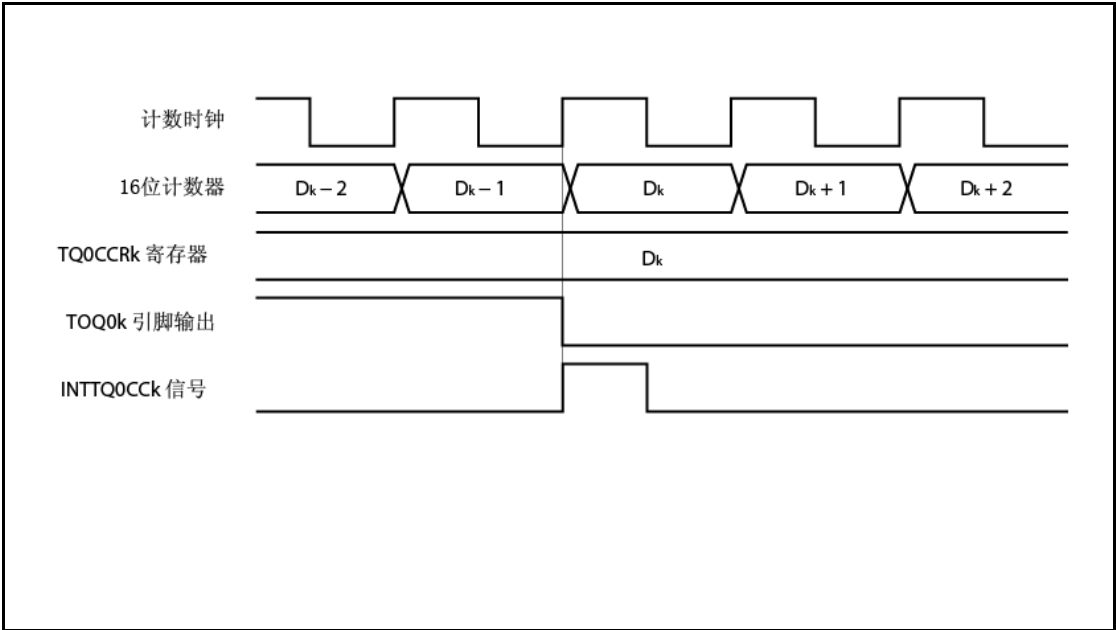
当 TQ0CCR0 寄存器从 D00 到 D01 重写且 TQ0CCRk 寄存器从 Dk0 到 Dk1 重写时,其中  $D00 > D01$  和  $Dk0 > Dk1$ ，如果当 16-位计数器的计数值大于 Dk1 且小于 Dk0 时 TQ0CCRk 寄存器重写且如果当计数值大于 D01 且小于 D00 时 TQ0CCR0 寄存器重写，则寄存器一重写即可反映各设置值且与计数值比较。计数器计数上至 FFFFH 且接着从 0000H 再次计数。当计数值匹配 Dk1 时，计数器产生 INTTQ0Ck 信号且断定 TOQ0k 引脚。当计数值匹配 D01 时，计数器产生 INTTQ0CC0 信号，解除断定 TOQ0k 引脚，且停止计数。

因此，计数器可能输出脉冲具有不同于开始预计的单触发脉冲的延迟周期或激活周期。

备注 k = 1 到 3

(b) 比较匹配中断请求信号(INTTQ0CCK)的产生时序

在单触发脉冲输出模式下 INTTQ0CCK 信号的产生时序不同于其他 INTTQ0CCK 信号；当 16-位计数器的计数值匹配 TQ0CCRk 寄存器的值时产生 INTTQ0CCK 信号。



通常，16-位计数器计数值匹配 TQ0CCRk 寄存器的值之后当其下一次计数时产生 INTTQ0CCK 信号。

然而，在单触发脉冲输出模式下，其产生较早一时钟脉冲。这是因为时序改为匹配 TOQ0k 引脚的改变时序。

备注      k = 1 到 3

### 7.5.5 PWM 输出模式(TQ0MD2 到 TQ0MD0 位 = 100)

在 PWM 输出模式下，当 TQ0CTL0.TQ0CE 位设为 1 时 PWM 波形从 TOQ01 到 TOQ03 引脚输出。此外，具有作为其半周期的 PWM 波形一周期的脉冲从 TOQ00 引脚输出。

图 7-24. 在 PWM 输出模式下的配置

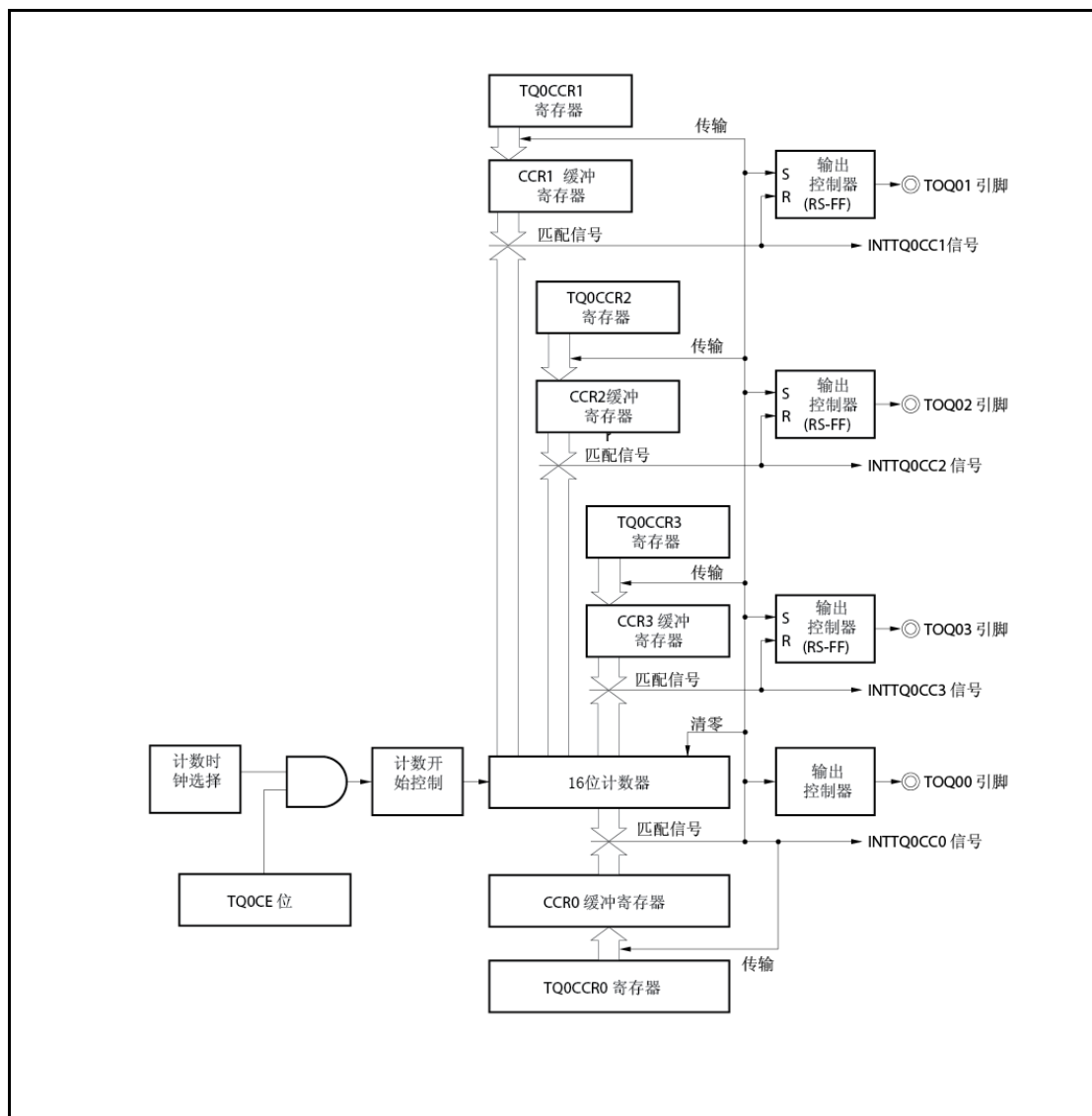
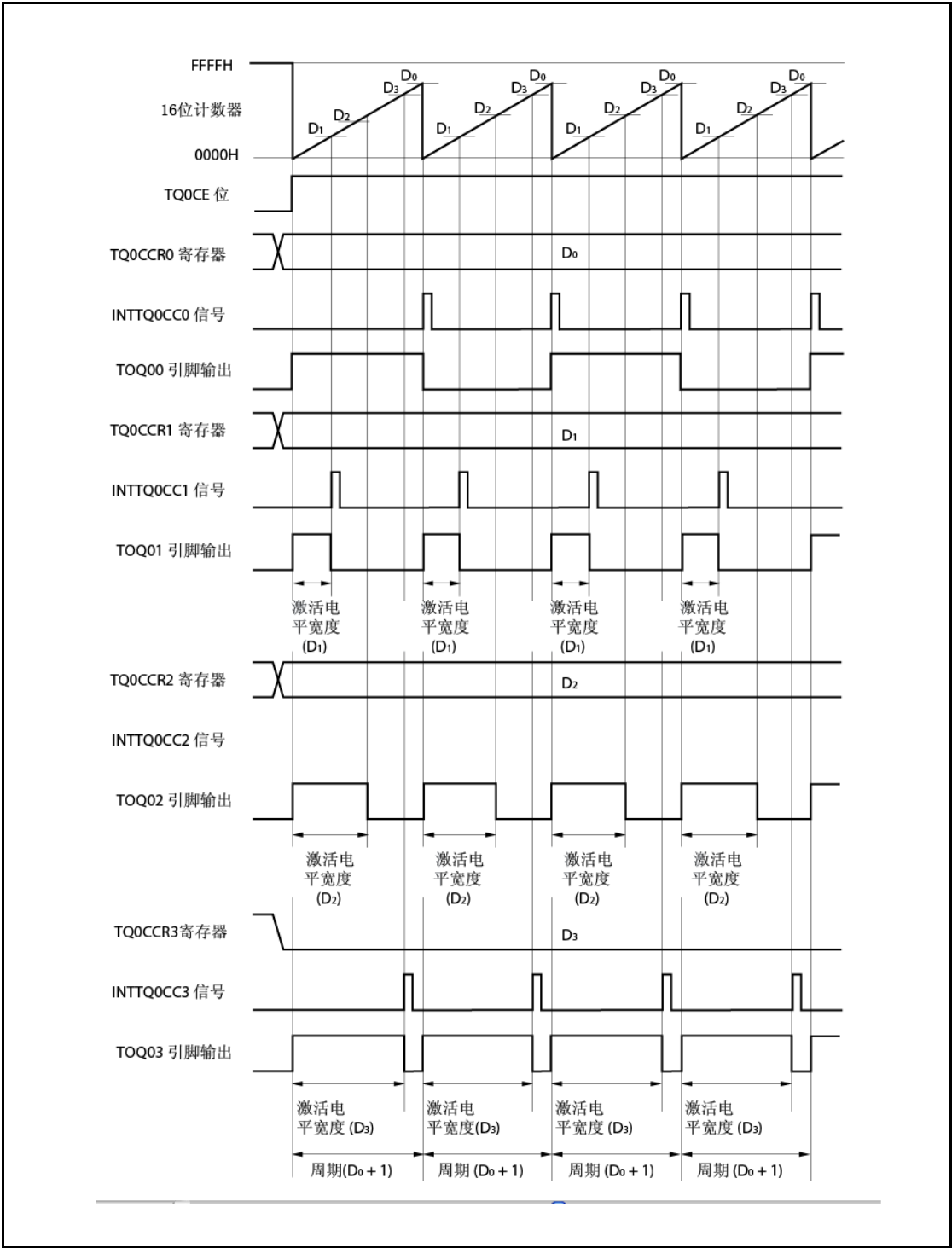


图 7-25. 在 PWM 输出模式下的基本时序



当 TQ0CE 位设为 1 时，16-位计数器从 FFFFH 清零 000H，开始计数，且从 TQ0Qk 引脚输出 PWM 波形。

PWM 波形的 激活电平宽度，周期及占空因数可如下所示进行计算。

激活电平宽度=(TQ0CCRk 寄存器的设置值) × 计数时钟周期  
周期 =(TQ0CCR0 寄存器的设置值+ 1) ×计数时钟周期  
占空因数=(TQ0CCRk 寄存器的设置值)/(TQ0CCR0 寄存器的设置值+ 1)

当计数器运行时 PWM 波形可通过重写 TQ0CCRm 寄存器改变。当 16-位计数器的计数值匹配 CCR0 缓冲寄存器的值时反映新写入值且 16-位计数器清零 000H。

16-位计数器的计数值匹配 CCR0 缓冲寄存器的值之后当 16-位计数器下一次计数时产生比较匹配中断请求信号 INTTQ0CC0，且 16-位计数器清零 000H。当 16-位计数器的计数值匹配 CCRk 缓冲寄存器的值时产生比较匹配中断请求信号 INTTQ0CCk。

备注 k = 1 到 3, m = 0 到 3

图 7-26. 在 PWM 输出模式下寄存器的设置(1/3)

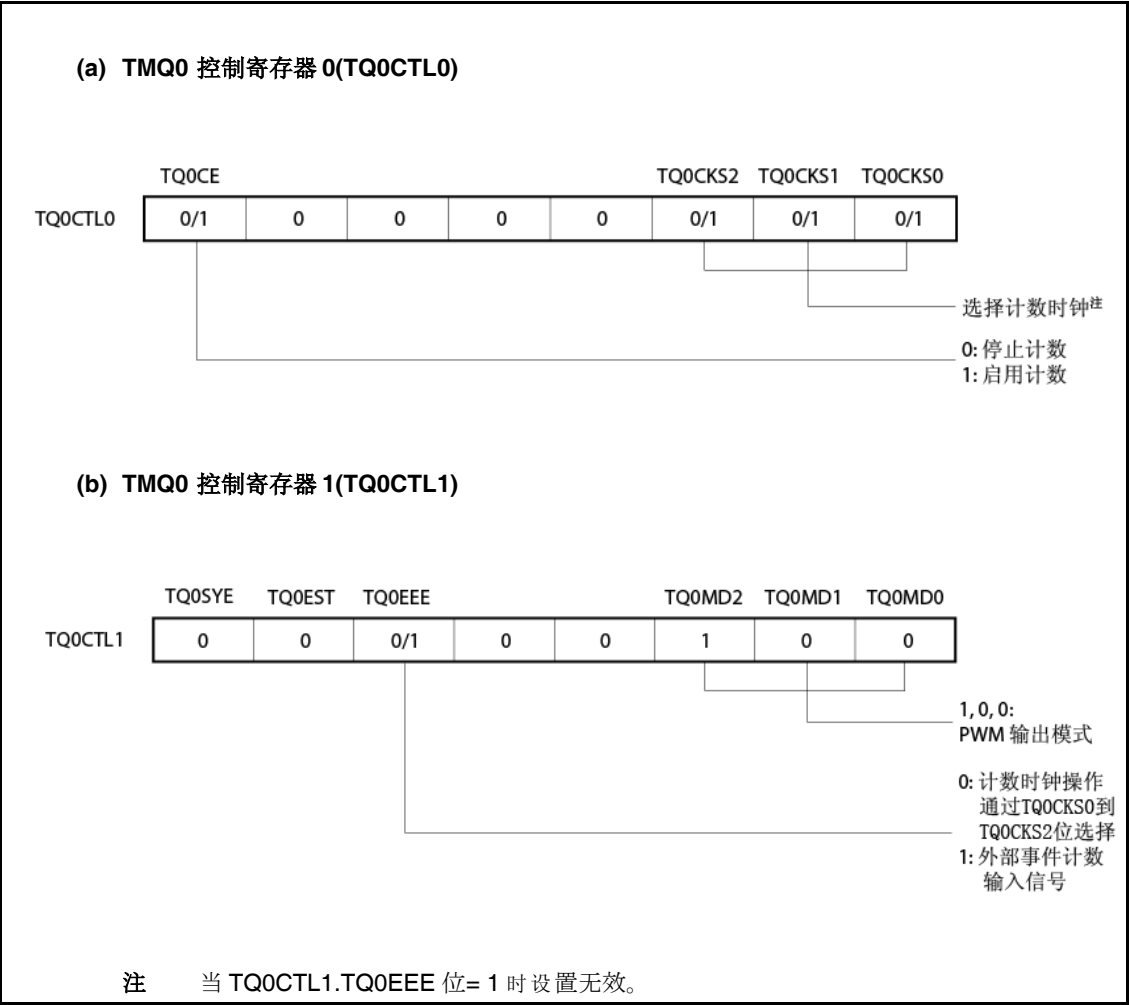


图 7-26. 在 PWM 输出模式下寄存器的设置(2/3)

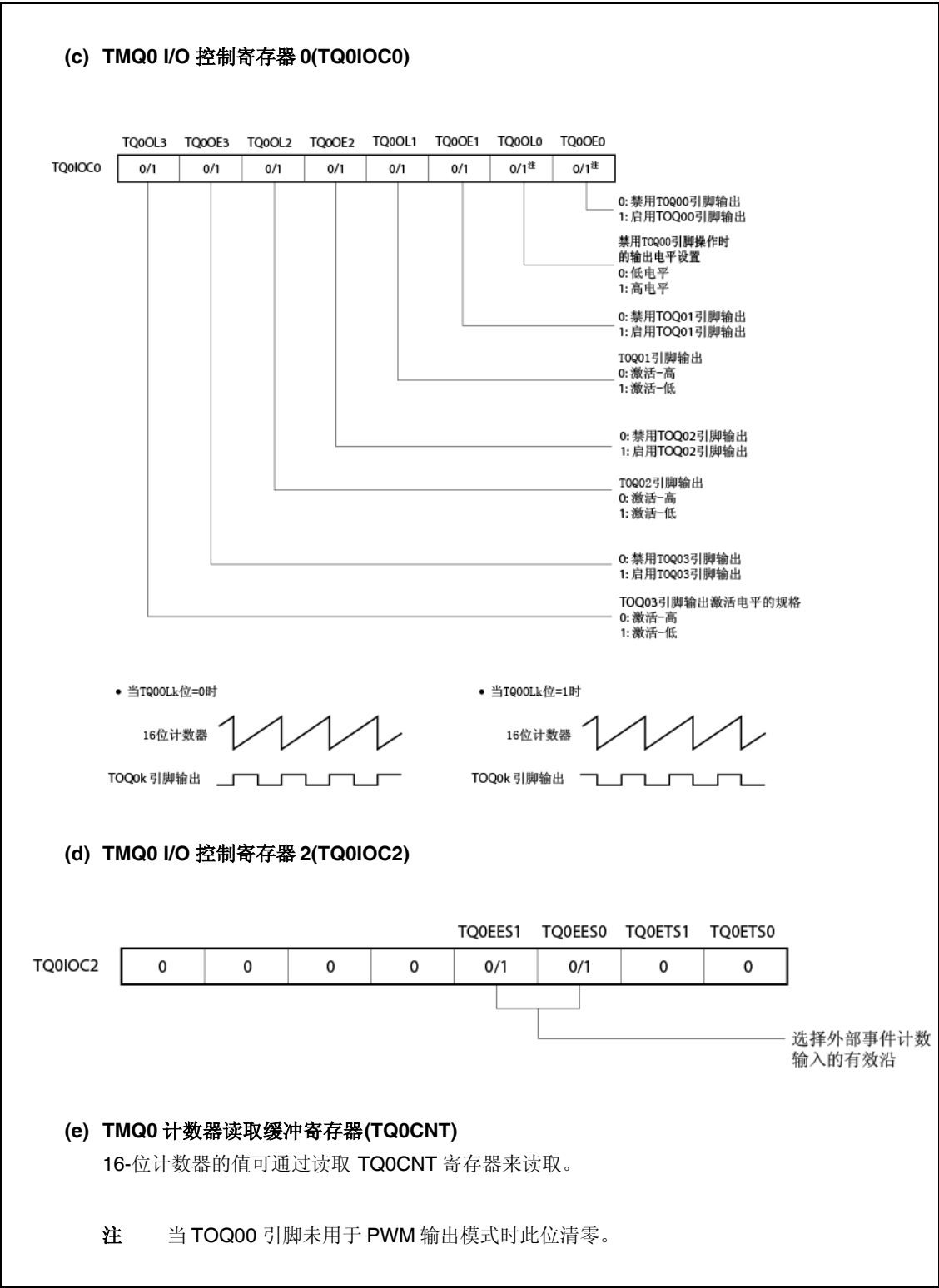


图 7-26. 在 PWM 输出模式下寄存器的设置(3/3)

**(f) TMQ0 捕获/比较寄存器 0 到 3(TQ0CCR0 到 TQ0CCR3)**

如果 D0 设为 TQ0CCR0 寄存器且 Dk 设为 TQ0CCK 寄存器，则 PWM 波形的周期和激活电平如下所示。

周期  $= (D0 + 1) \times \text{计数时钟周期}$

激活电平宽度  $= Dk \times \text{计数时钟周期}$

- 备注
1. 在 PWM 输出模式下不使用 TMQ0 I/O 控制寄存器 1(TQ0IOC1)和 TMQ0 选项寄存器 0(TQ0OPT0)。
  2. 通过写入 TMQ0 捕获/比较寄存器 1(TQ0CCR1)更新 TMQ0 捕获/比较寄存器 2(TQ0CCR2)和 TMQ0 捕获/比较寄存器 3(TQ0CCR3)有效。



(1) 在 PWM 输出模式下的操作流程

图 7-27. 在 PWM 输出模式下的软件处理流程(1/2)

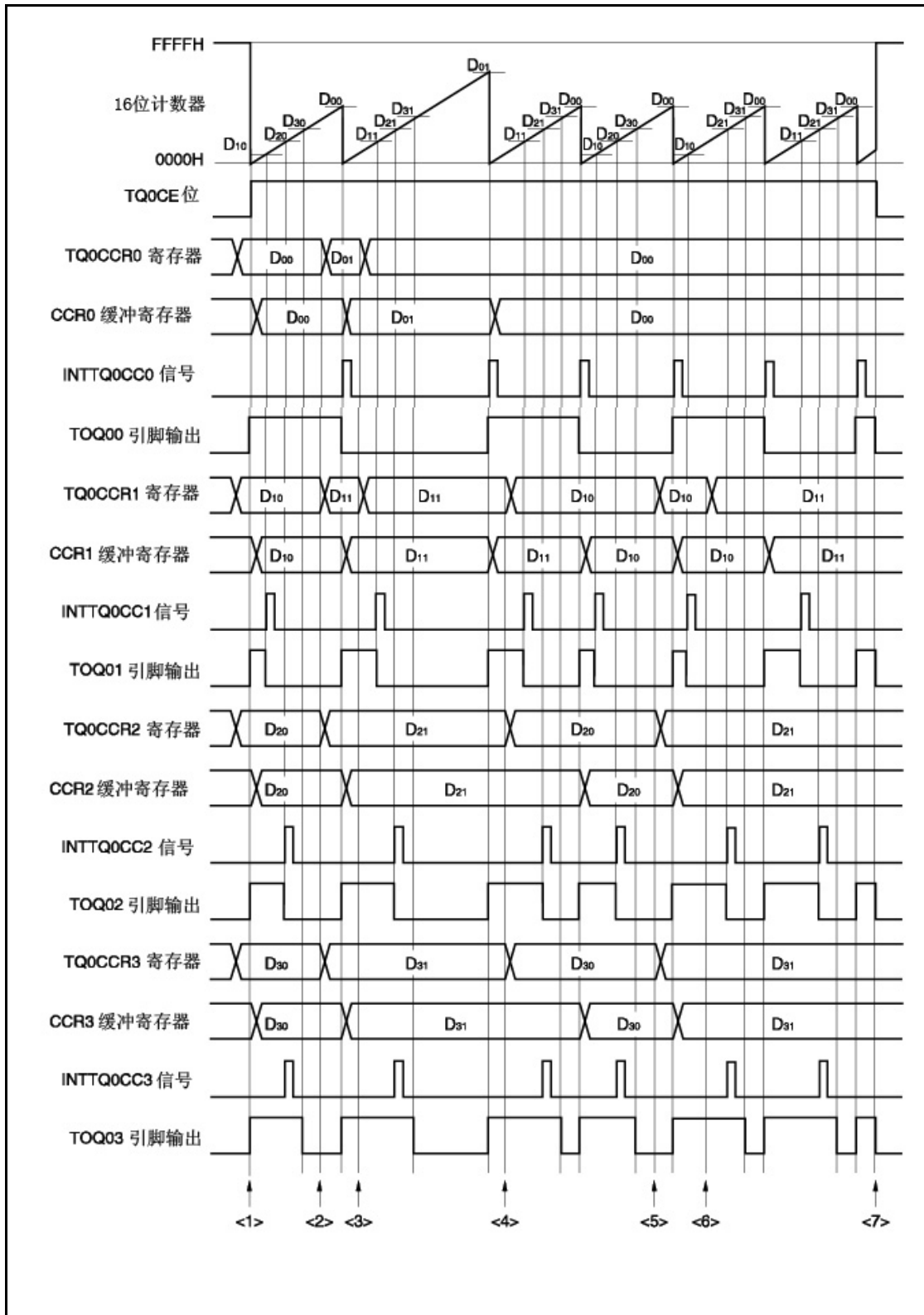
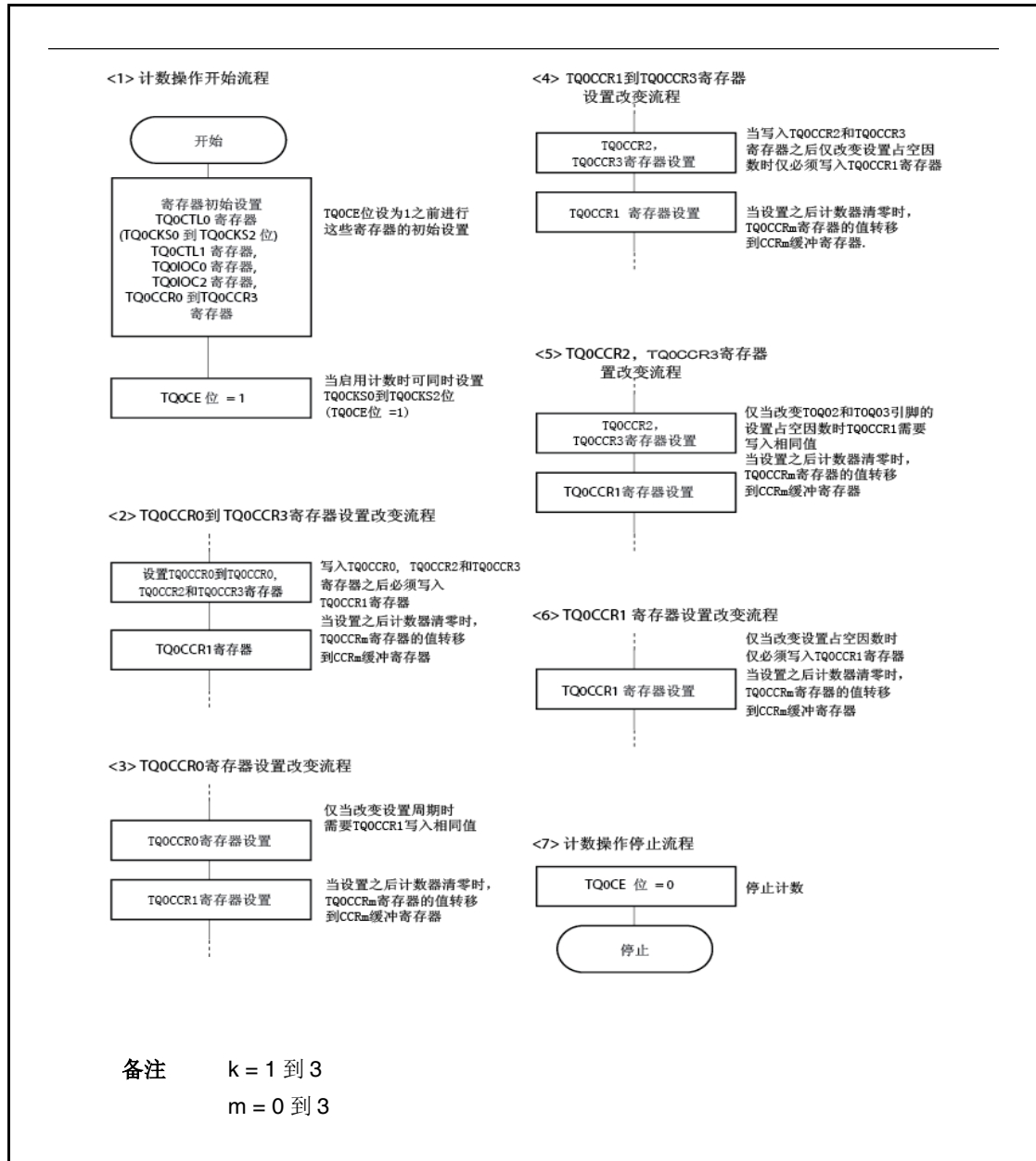


图 7-27. 在 PWM 输出模式下的软件处理流程(2/2)

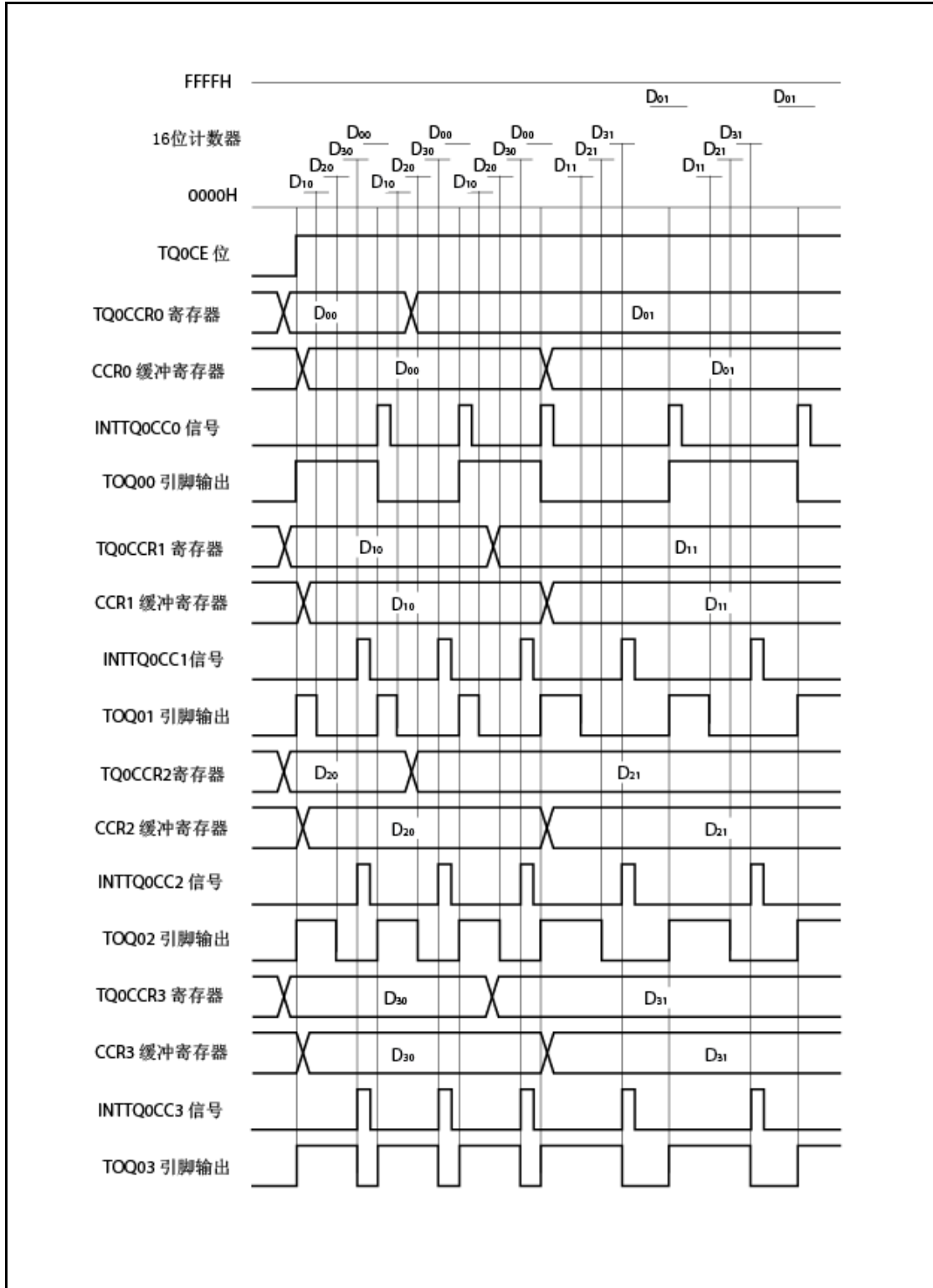


## (2) PWM 输出模式运行时序

### (a) 运行期间改变脉宽

当计数器运行时要改变 PWM 波形，最后写入 TQ0CCR1 寄存器。

在检测到 INTTQ0CC1 信号之后写入 TQ0CCR1 寄存器之后，重写 TQ0CCRk 寄存器。



数据要从 TQ0CCRM 寄存器传输到 CCRm 缓冲寄存器，必须写入 TQ0CCR1 寄存器。

此时要改变 PWM 波形的周期和激活电平，首先设置 TQ0CCR0 寄存器的周期，设置 TQ0CCR2 和 TQ0CCR3 寄存器的激活电平宽度，且接着设置 TQ0CCR1 寄存器的激活电平宽度。

仅要改变 PWM 波的激活电平宽度(占空因数)，首先设置 TQ0CCR2 和 TQ0CCR3 寄存器的激活电平，且接着设置 TQ0CCR1 寄存器的激活电平。

仅要改变 TOQ01 引脚的 PWM 波形输出的激活电平宽度(占空因数)，则仅要设置 TQ0CCR1 寄存器。

仅要改变 TOQ02 和 TOQ03 引脚的 PWM 波形输出的激活电平宽度(占空因数)，首先设置 TQ0CCR2 和 TQ0CCR3 寄存器的激活电平宽度，且接着 TQ0CCR1 寄存器写入相同值。

写入 TQ0CCR1 寄存器之后，写入 TQ0CCRM 寄存器的值与时序 16-位计数器清零同步转移到 CCRm 缓冲寄存器，且用作与 16-位计数器的值比较的值。

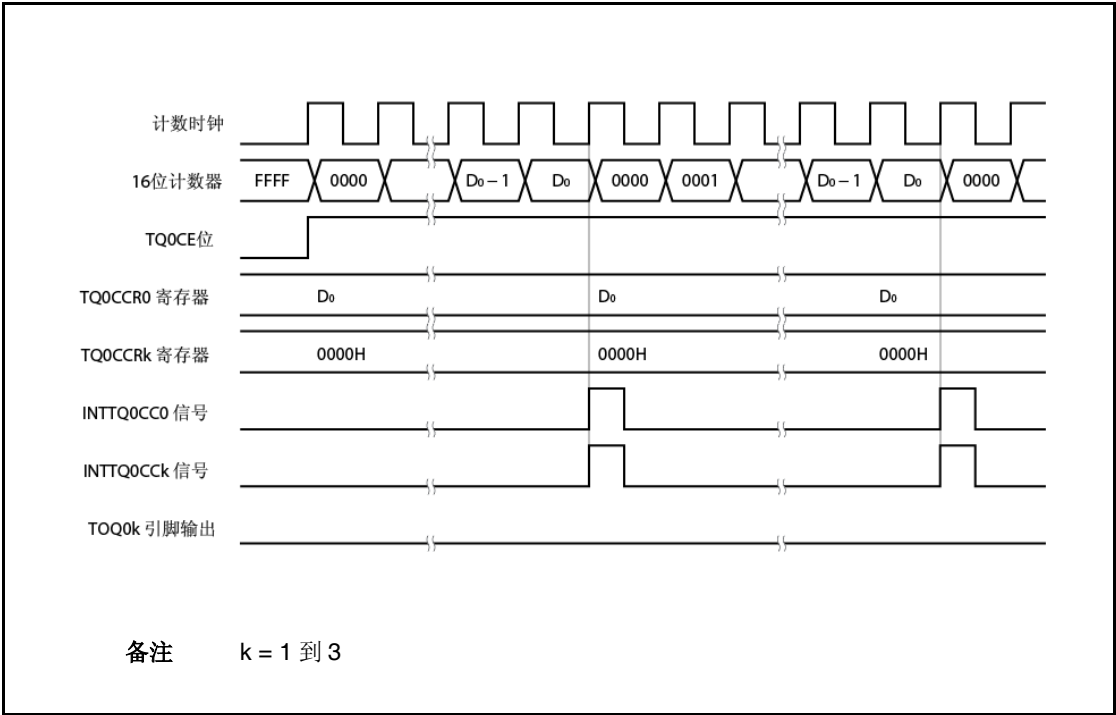
仅要改变 PWM 波形的周期，首先设置 TQ0CCR0 寄存器的周期，且接着 TQ0CCR1 寄存器写入相同值。

写入 TQ0CCR1 寄存器一次之后要再次写入 TQ0CCR0 到 TQ0CCR3 寄存器，INTTQ0CC0 信号产生之后进行。否则，因为数据从 TQ0CCRM 寄存器传输到 CCRm 缓冲寄存器的时序与写入 TQ0CCRM 寄存器冲突，所以可能未定义 CCRm 缓冲寄存器的值。

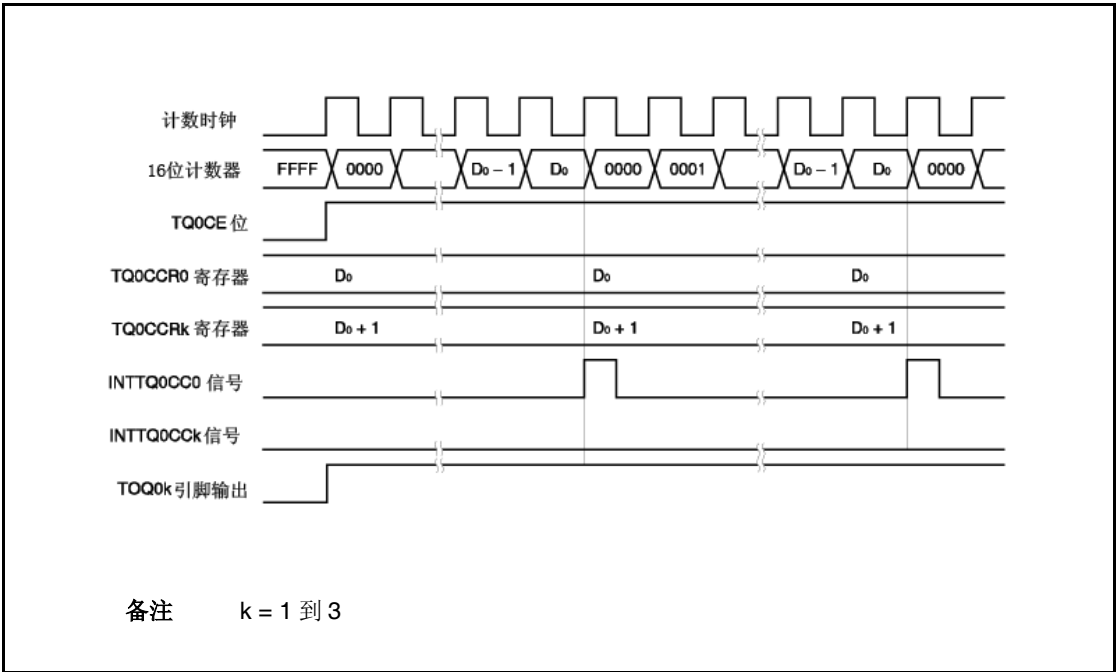
**备注**      m = 0 到 3

(b) PWM 波形的 0%/100%输出

要输出 0% 波形，将 TQ0CCRk 寄存器设置为 0000H。如果 TQ0CCR0 寄存器的设置值为 FFFFH，则定期产生 INTTQ0CCK 信号。

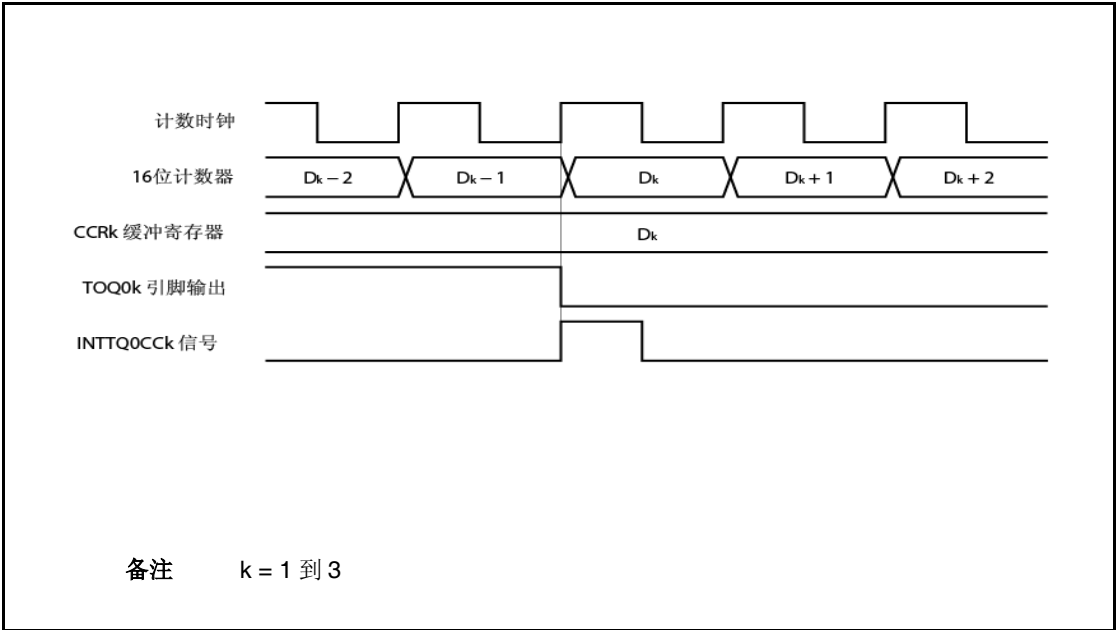


要输出 100% 波形，设置 TQ0CCRk 寄存器为值(TQ0CCR0 寄存器的设置值+1)。如果 TQ0CCR0 寄存器的设置值为 FFFFH，则不能产生 100%输出。



(c) 比较匹配中断请求信号(INTTQ0CCK)的产生时序

在外部触发脉冲输出模式下 INTTQ0CCK 信号的产生时序与其他 INTTQ0CCK 信号时序不同；  
当 16-位计数器的计数值匹配 CCRk 缓冲寄存器的值时产生 INTTQ0CCK 信号。



通常，16-位计数器的计数值匹配 CCRk 缓冲寄存器的值之后 INTTQ0CCK 信号与下一次计数同步产生。

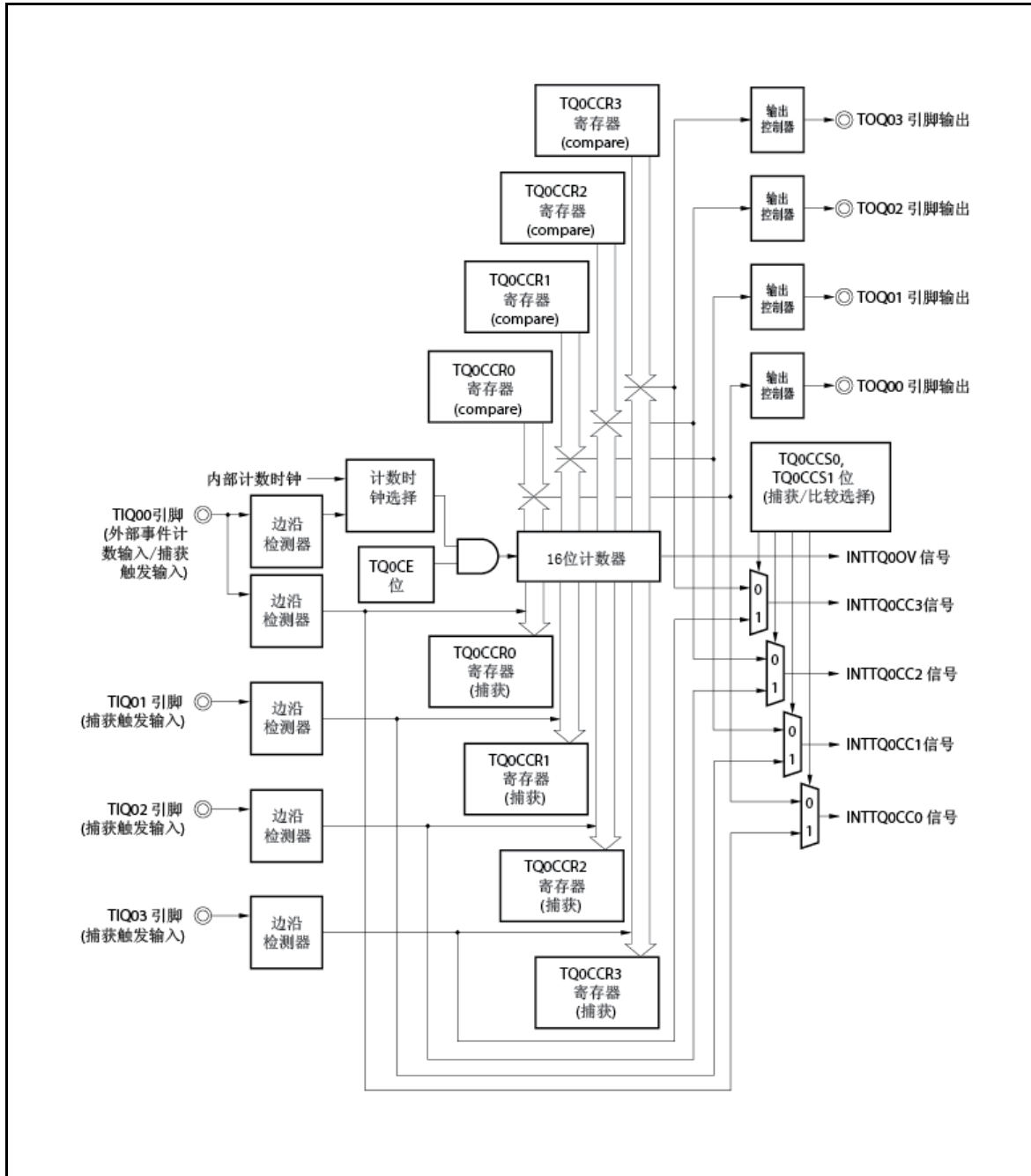
然而，在外部触发脉冲输出模式下，其较早产生一时钟脉冲。这是因为时序改为匹配改变 TOQ0k 引脚的输出信号的时序。

### 7.5.6 自由定时器模式(TQ0MD2 到 TQ0MD0 位 = 101)

在自由定时器模式下，当 TQ0CTL0.TQ0CE 位设为 1 时 16-位定时器/事件计数器 Q 开始计数。此时根据 TQ0OPT0.TQ0CCS0 和 TQ0OPT0.TQ0CCS1 位的设置，TQ0CCRm 寄存器可用作比较寄存器或捕获寄存器。

备注 m = 0 到 3

图 7-28. 自由定时器模式下的配置

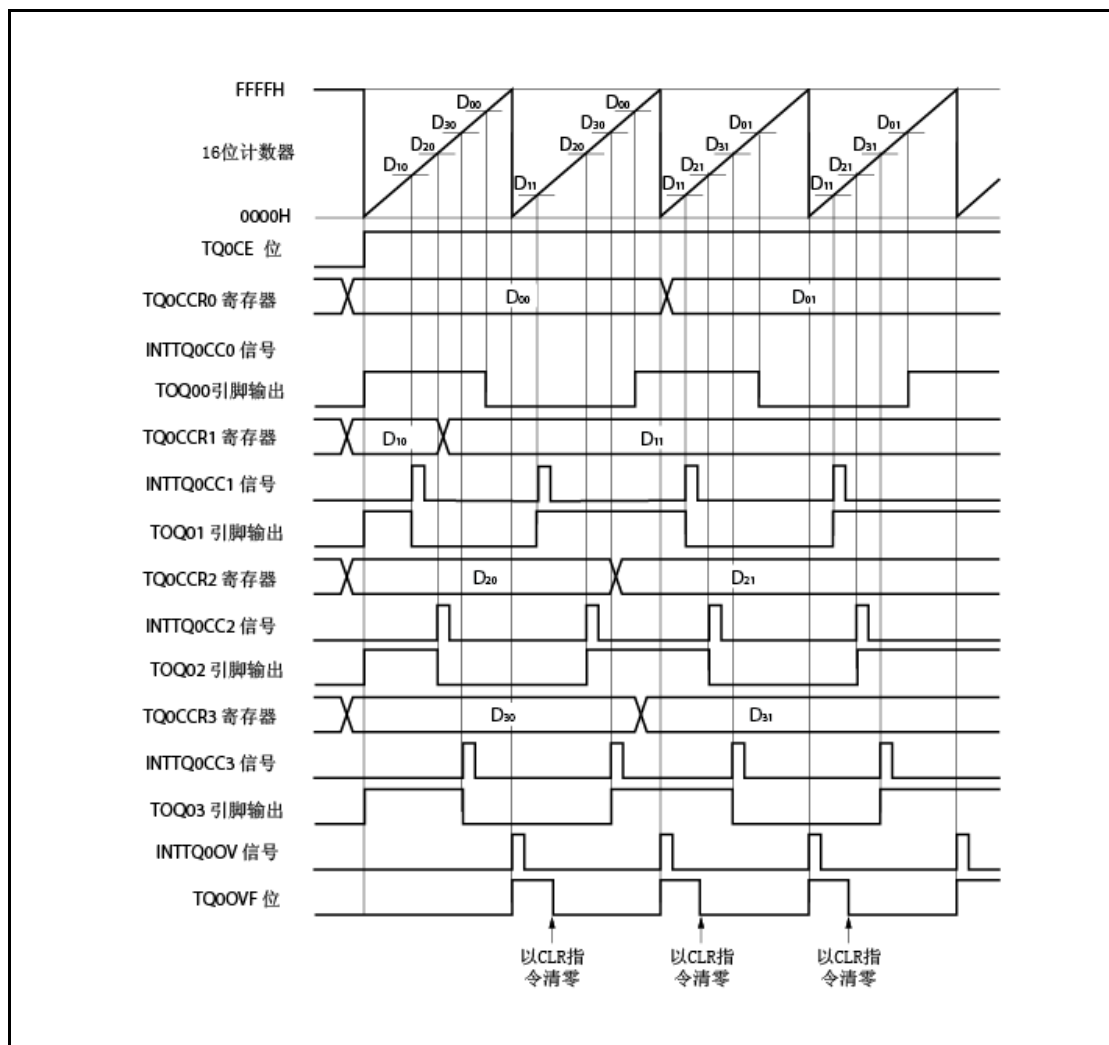


当 TQ0CE 位设为 1 时, 16-位定时器/事件计数器 Q 开始计数, 且 TOQ00 到 TOQ03 引脚的输出信号反相。当 16-位计数器的计数值稍后匹配 TQ0CCRm 寄存器的设置值时, 产生比较匹配中断请求信号 (INTTQ0CCm), 且 TOQ0m 引脚的输出信号反相。

16-位计数器与计数时钟同步继续计数。当其计数上至 FFFFH 时, 在下一时钟脉冲产生溢出中断请求信号 (INTTQ0OV), 清零 0000H, 且继续计数。此时, 溢出标记 (TQ0OPT0.TQ0OVF 位) 也设为 1。通过软件执行 CLR 指令溢出标记清零。

当计数器运行时可重写 TQ0CCRm 寄存器。如果重写, 此时反映新值, 且与计数值比较。

图 7-29. 自由定时器模式下的基本时序(比较功能)





当 TQ0CE 位设为 1 时, 16-位计数器开始计数。当检测到输入到 TIQ0m 引脚的有效沿时, 16-位计数器的计数值存储在 TQ0CCRm 寄存器, 且产生捕获中断请求信号(INTTQ0CCm)。

16-位计数器与计数时钟同步继续计数。当其计数上至 FFFFH, 在下一时钟脉冲处产生溢出中断请求信号(INTTQ0OV), 清零 000H, 且继续计数。此时, 溢出标记(TQ0OVF 位)也设为 1。通过软件执行 CLR 指令溢出标记清零。

图 7-30. 自由定时器模式下的基本时序(捕获功能)

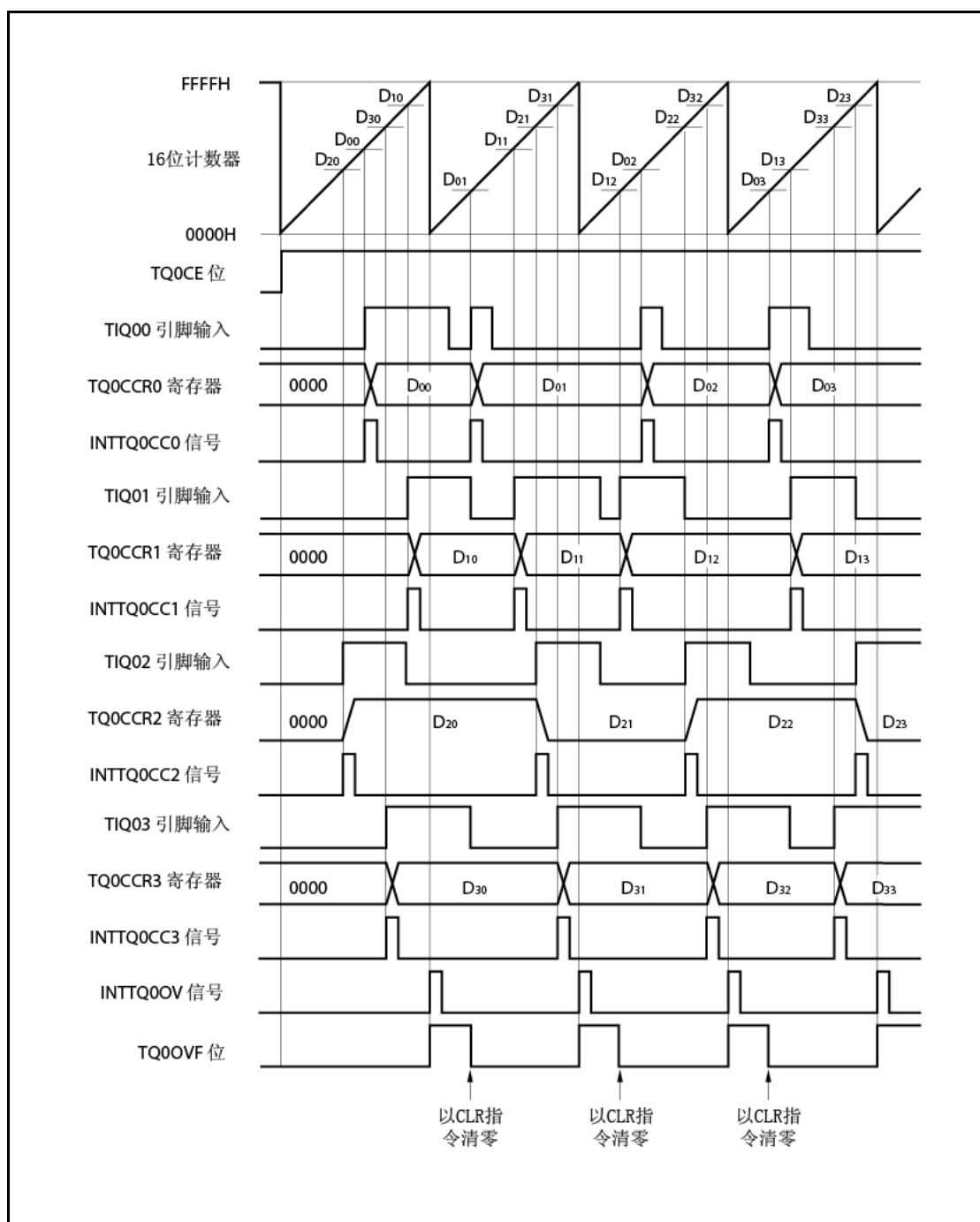


图 7-31. 自由定时器模式下的寄存器设置(1/3)

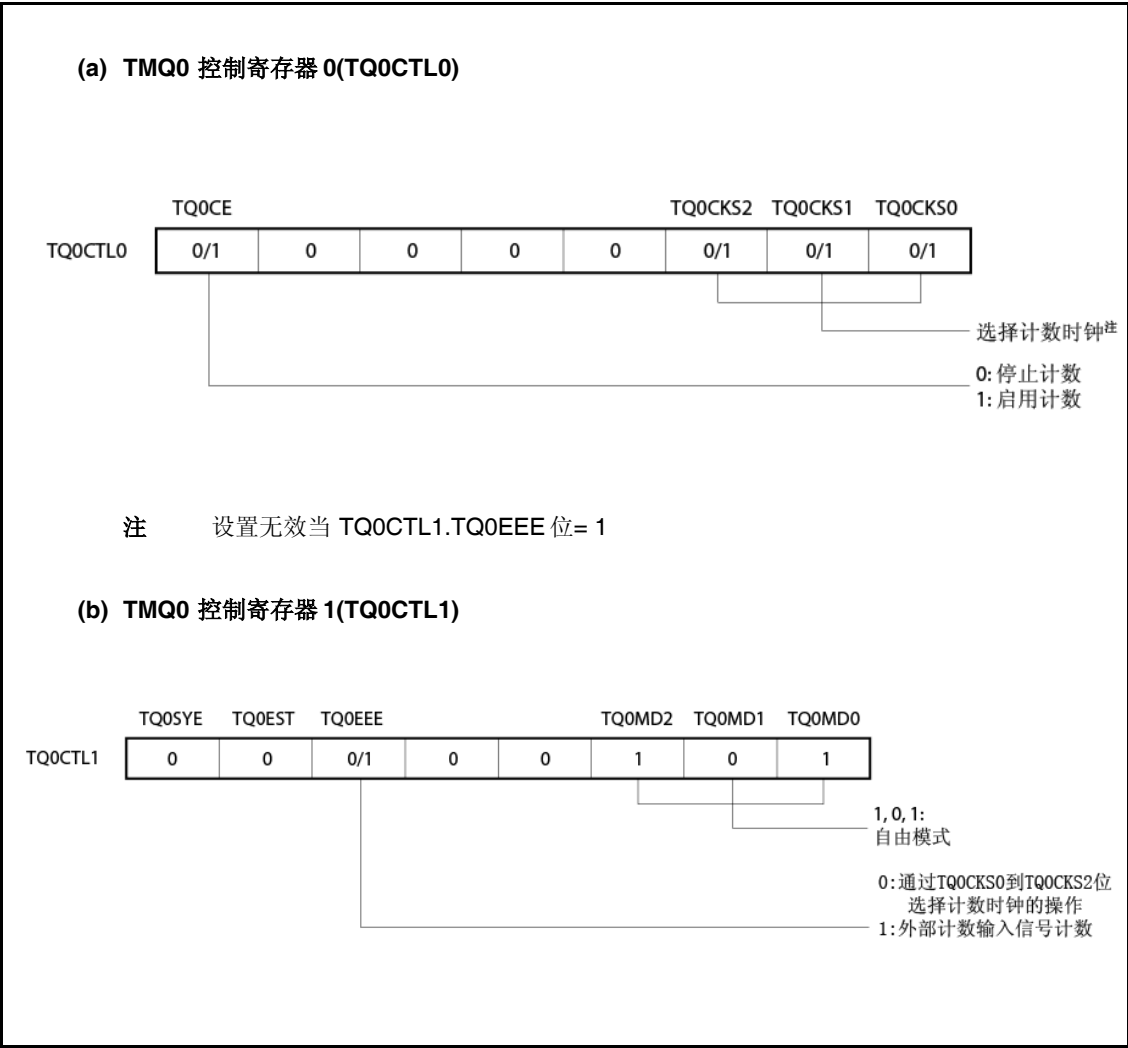
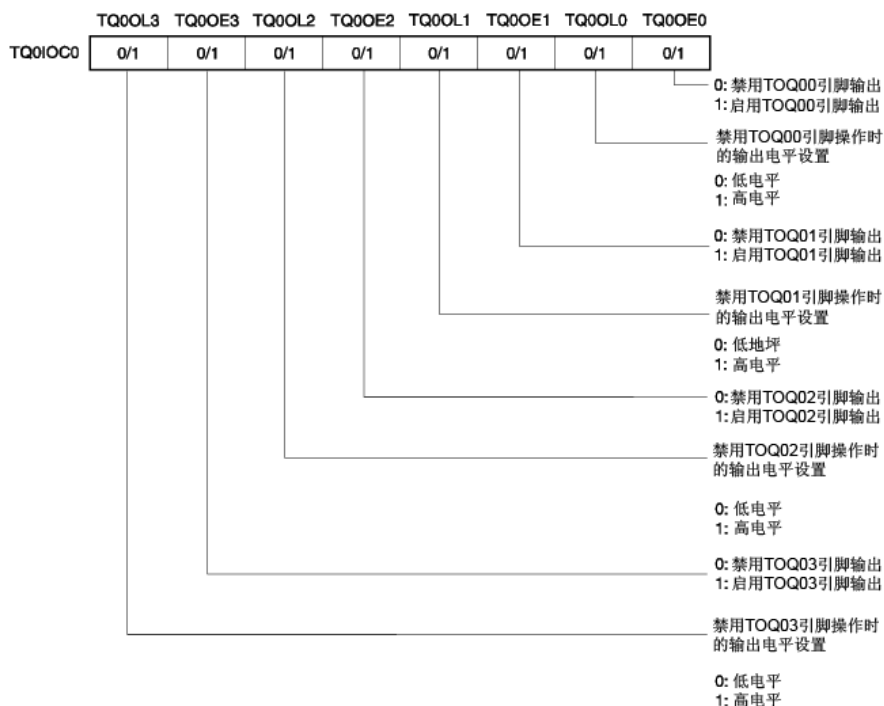


图 7-31. 自由定时器模式下的寄存器设置(2/3)

(c) TMQ0 I/O 控制寄存器 0(TQ0IOC0)



(d) TMQ0 I/O 控制寄存器 1(TQ0IOC1)

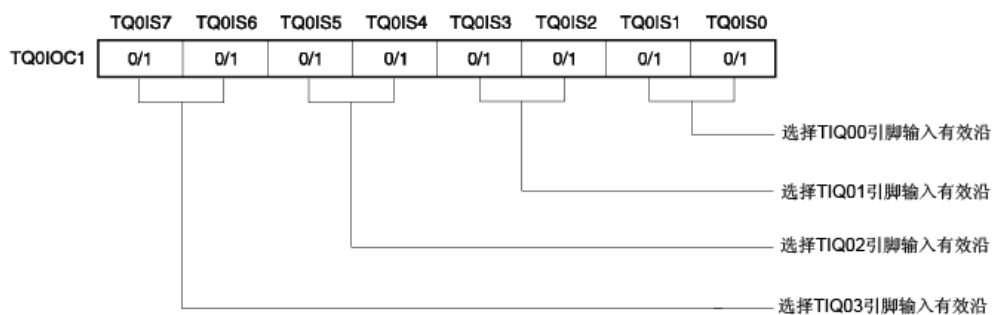
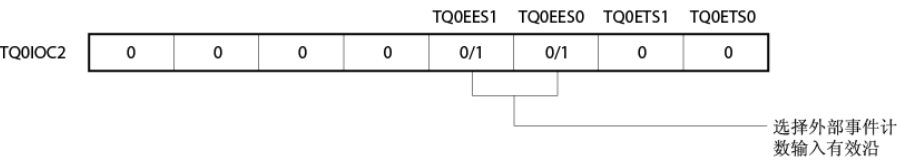
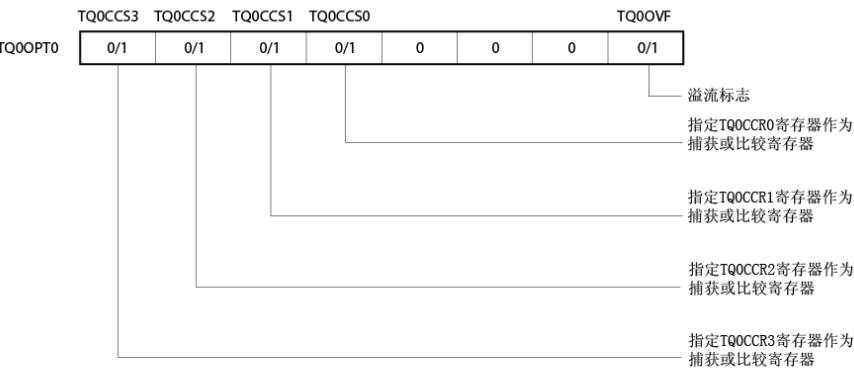


图 7-31. 自由定时器模式下的寄存器设置(3/3)

(e) TMQ0 I/O 控制寄存器 2(TQ0IOC2)



(f) TMQ0 选项寄存器 0(TQ0OPT0)



(g) TMQ0 计数器读取缓冲寄存器(TQ0CNT)

16-位计数器的值可通过读取 TQ0CNT 寄存器来读取。

(h) TMQ0 捕获/比较寄存器 0 到 3(TQ0CCR0 到 TQ0CCR3)

根据 TQ0OPT0.TQ0CCSm 位设置这些寄存器用作捕获寄存器或比较寄存器。

当寄存器用作捕获寄存器时，当检测到输入到 TIQ0m 引脚的有效沿时其存储 16-位计数器的计数值。

当寄存器用作比较寄存器时且当 Dm 设为 TQ0CCRm 寄存器时，当计数器达到(Dm+1)时产生 INTTQ0CCm 信号，且 TOQ0m 引脚的输出信号反相。

备注 m = 0 到 3

(1) 自由定时器模式下的操作流程

(a) 当捕获/比较寄存器用作比较寄存器时

图 7-32. 自由定时器模式下的软件处理流程(比较功能)(1/2)

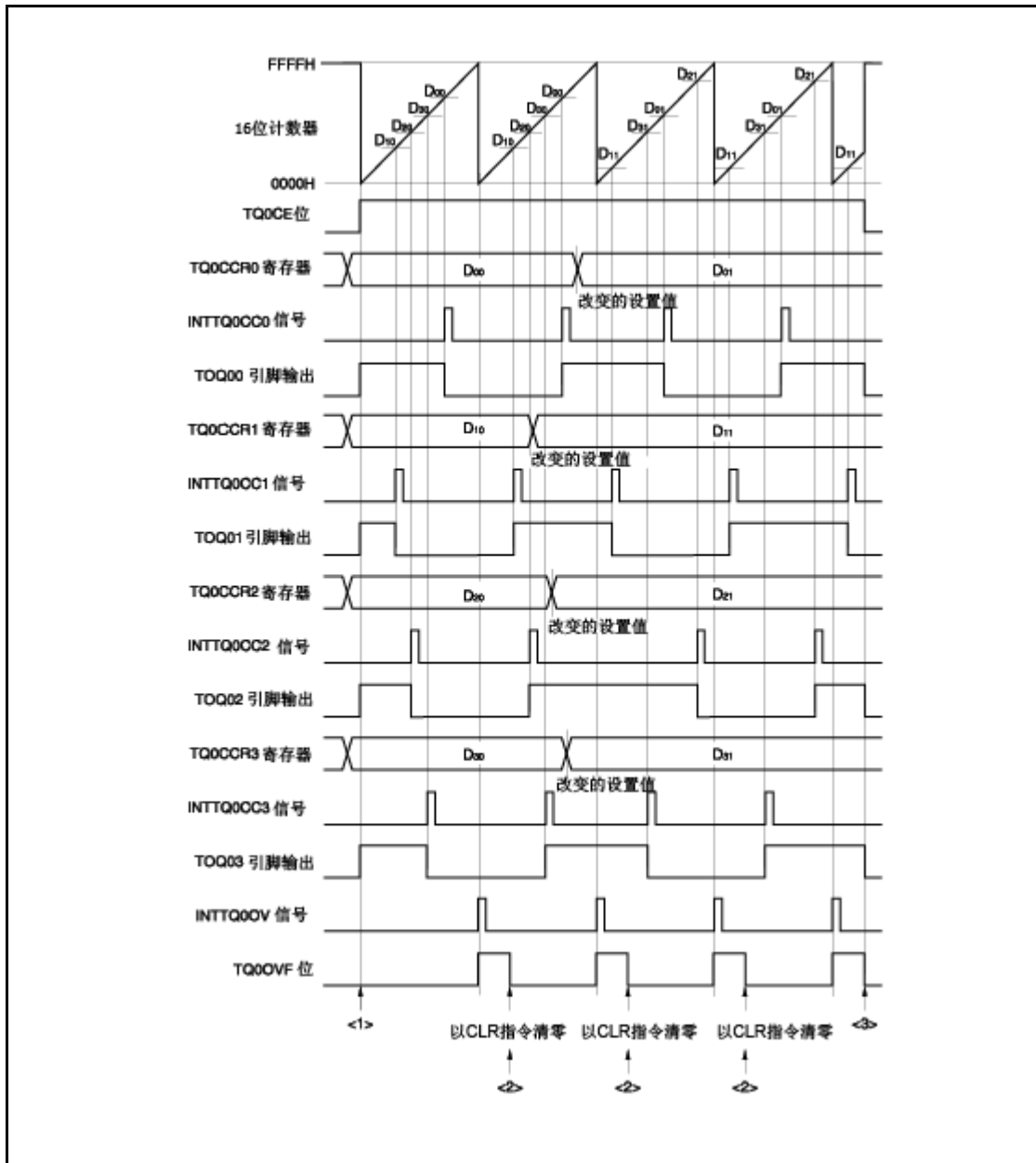
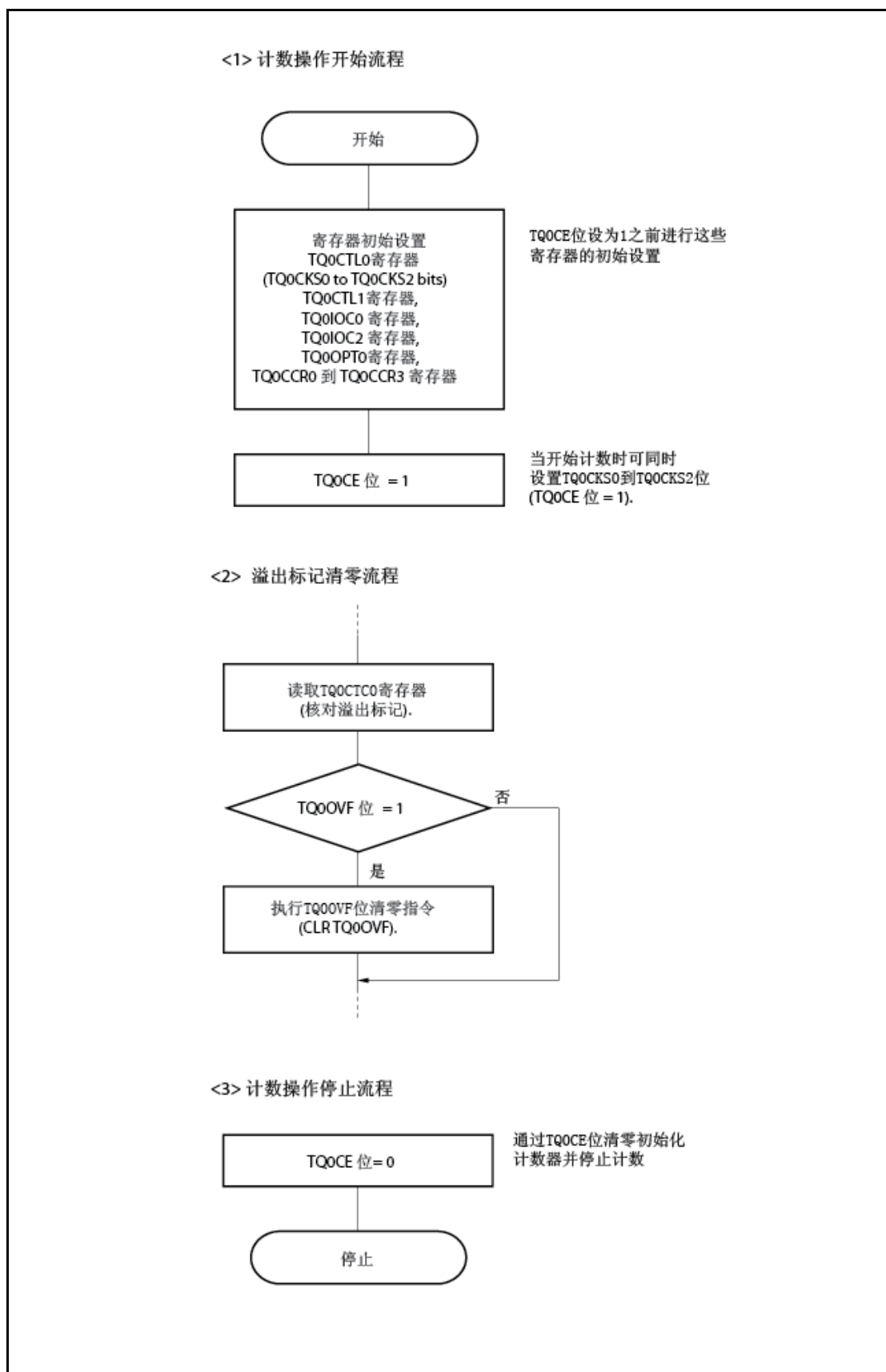


图 7-32. 自由定时器模式下的软件处理流程(比较功能)(2/2)



(b) 当捕获/比较寄存器用作捕获寄存器

图 7-33. 自由定时器模式下的软件处理流程(捕获功能)(1/2)

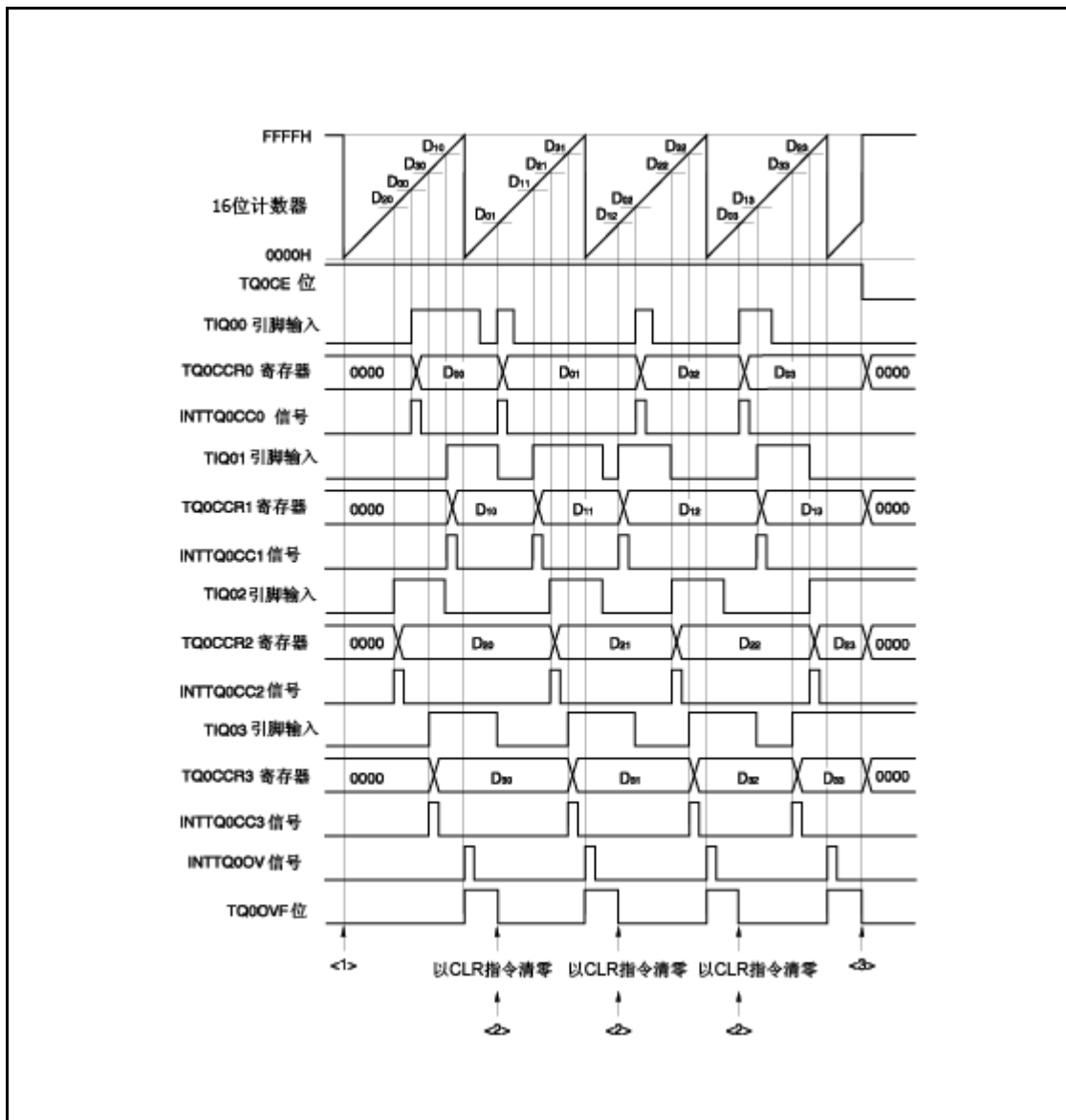
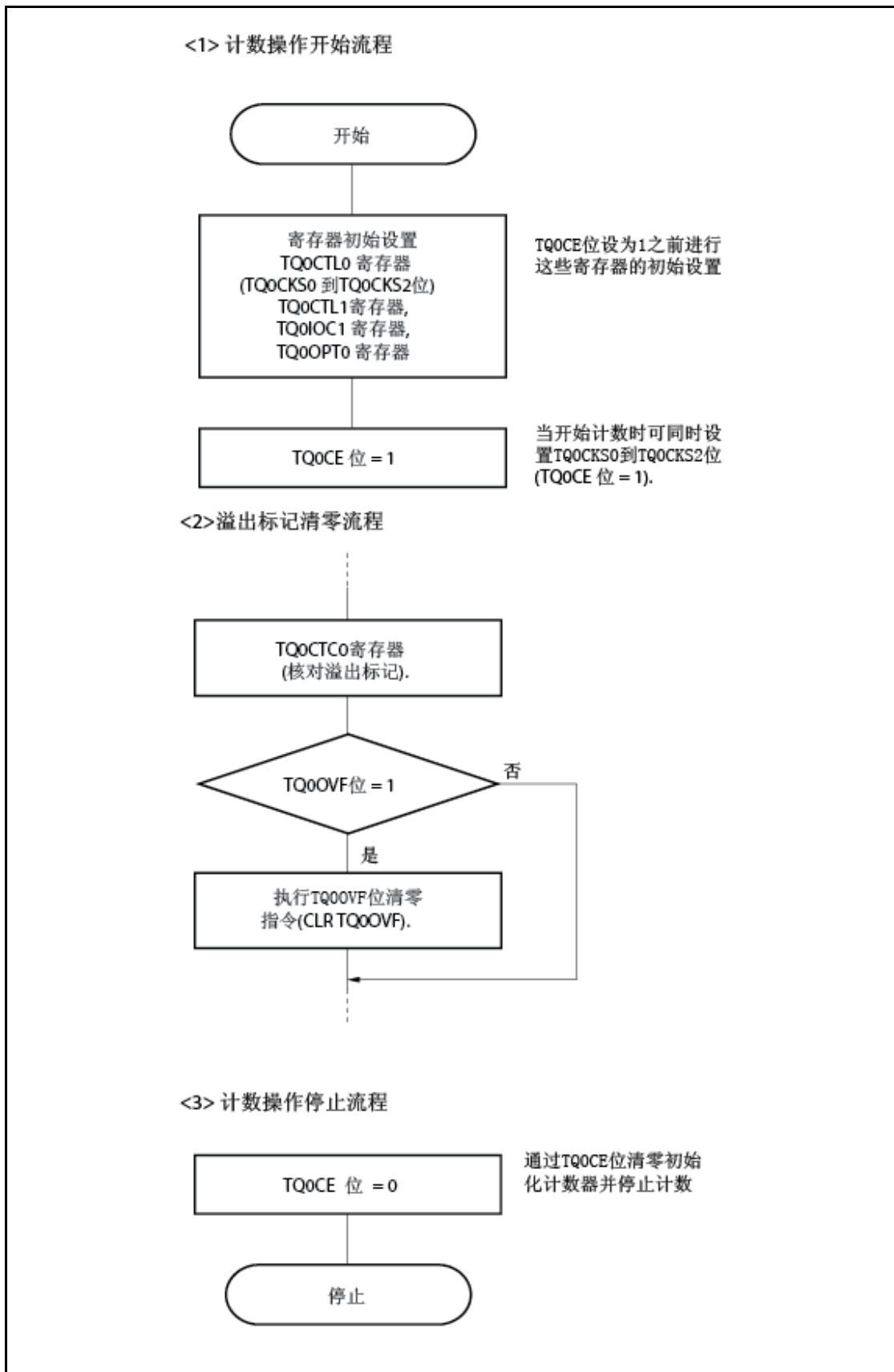


图 7-33. 自由定时器模式下的软件处理流程(捕获功能)(2/2)

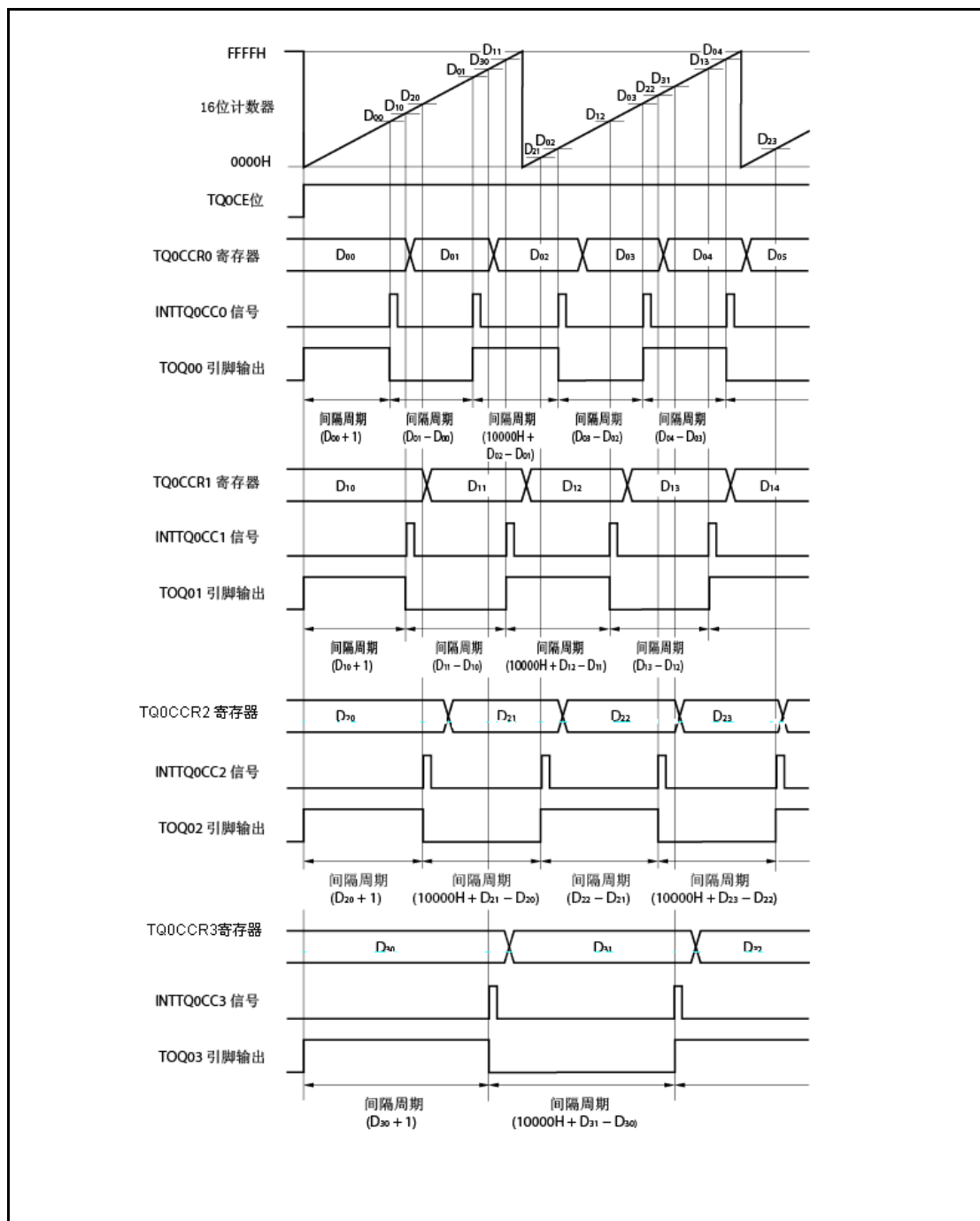




## (2) 自由定时器模式下的操作时序

### (a) 比较寄存器的时间间隔操作

当 16-位定时器/事件计数器 Q 用作计时器且 TQ0CCRm 寄存器用作比较寄存器时，需要设置比较值的软件处理以每当检测到 INTTQ0CCm 信号时产生下一中断请求信号。



当自由定时器模式下执行时间间隔操作，一个通道可设置两个间隔。

要执行间隔操作，当检测到 INTTQ0CCm 信号时必须重设进行中断服务中相应 TQ0CCRm 寄存器的值。

复位 TQ0CCRm 寄存器的设置值可通过以下表达式计算，其中“Dm”为间隔周期。

比较寄存器默认值：Dm - 1

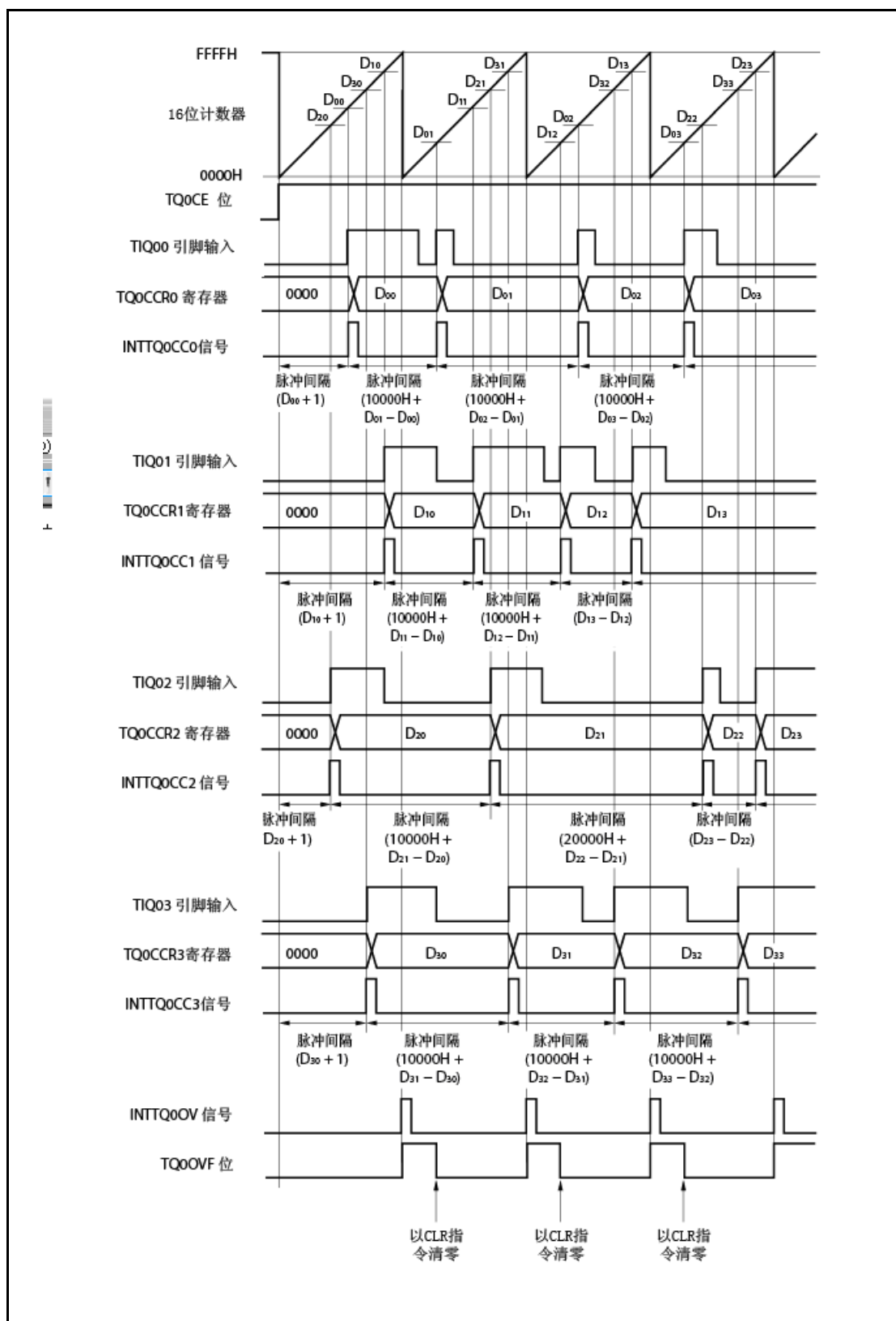
第二次及随后设为比较寄存器的值：先前设置值+ Dm

(如果计算 结果大于 FFFFH，从结果减去 10000H 并将此值设置到寄存器。)

**备注**      m = 0 到 3

## (b) 捕获寄存器的脉宽测量

当 TQ0CCRm 寄存器用作捕获寄存器时进行脉宽测量，每当检测到 INTTQ0CCm 信号时需要读取捕获寄存器的软件处理且用于计算间隔。



自由定时器模式下当进行脉宽测量时，一个通道可测量四脉冲宽度。

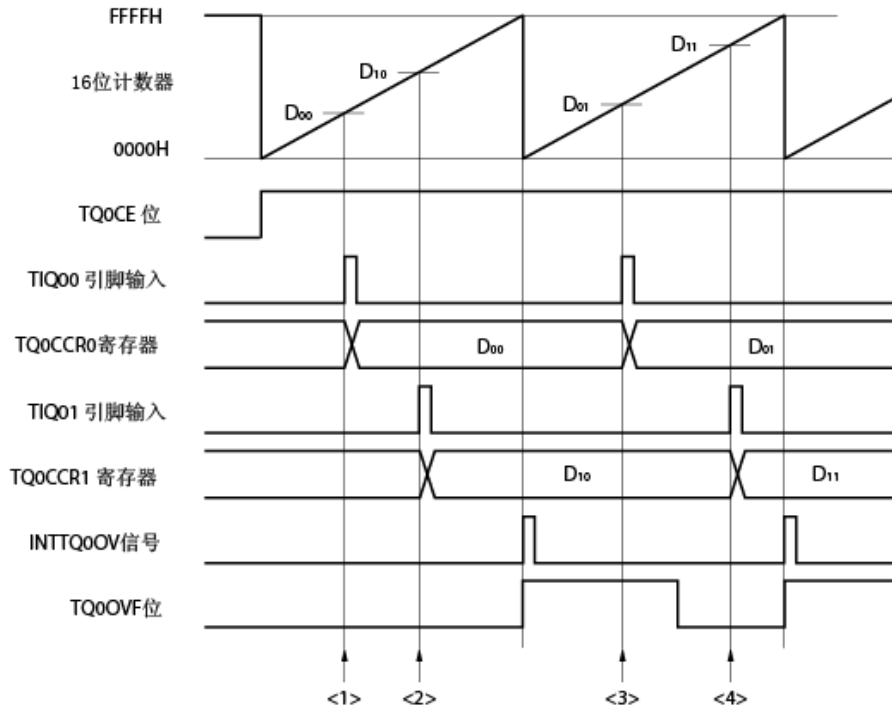
要测量脉冲宽度，脉宽可通过与 INTTQ0CCm 信号同步读取 TQ0CCRm 寄存器的值计算，并计算读取值与先前读取值之间的差值。

备注      m = 0 到 3

(c) 当使用两个或两个以上捕获寄存器时的溢出处理

当使用两个捕获寄存器时处理溢出标记必须小心。首先，错误处理的示例如下所示。

当使用两个或两个以上捕获寄存器时的错误处理示例



当自由定时器模式下测量两脉冲宽度时可能出现以下问题。

<1> 读取 TQ0CCR0 寄存器(TIQ00 引脚输入的默认值设置)。

<2> 读取 TQ0CCR1 寄存器(TIQ01 引脚输入的默认值设置)。

<3> 读取 TQ0CCR0 寄存器。

读取溢出标记。如果溢出标记为 1，则其清零。

因为溢出标记为 1，所以脉宽可通过 $(10000H + D_{01} - D_{00})$  计算。

<4> 读取 TQ0CCR1 寄存器。

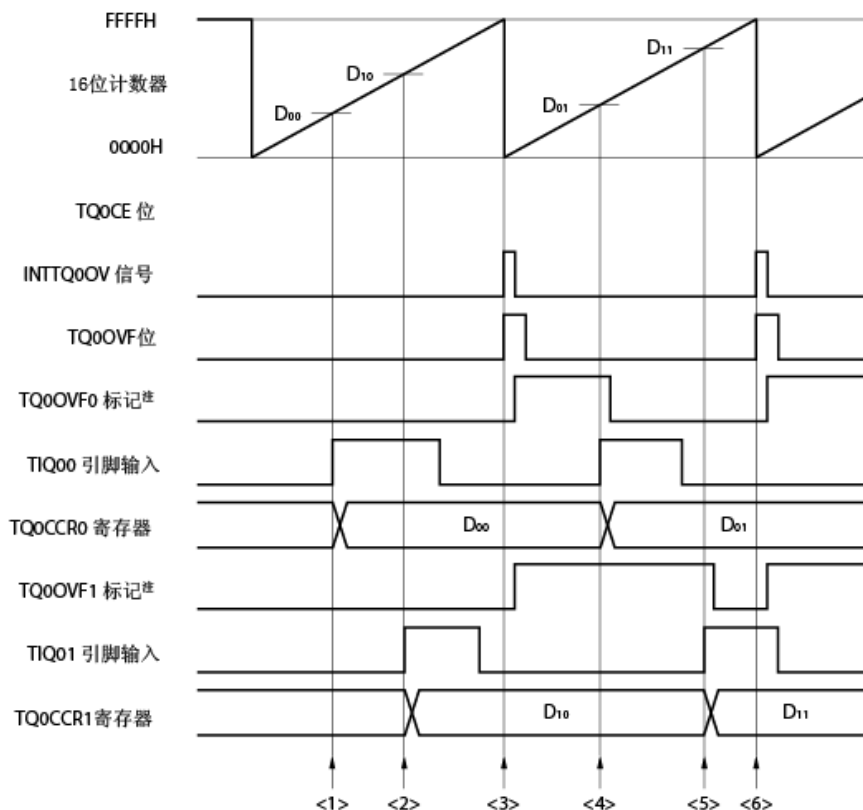
读取溢出标记。因为在<3>标记清零，所以读取 0。

因为溢出标记为 0，脉宽可通过 $(D_{11} - D_{10})$ (错误) 计算。

当使用两个捕获寄存器时，且如果溢出标记通过一个捕获寄存器清零，则其他捕获寄存器可能不能获得校正脉冲宽度。

当使用两个捕获寄存器时使用软件。如何使用软件的示例如下所示。

使用两个捕获寄存器时的示例(使用溢出中断)



注 TQ0OVF0 和 TQ0OVF1 标记通过软件在内部 RAM 设置。

<1> 读取 TQ0CCR0 寄存器(TIQ00 引脚输入的默认值设置)。

<2> 读取 TQ0CCR1 寄存器(TIQ01 引脚输入的默认值设置)。

<3> 发生溢出。溢出中断服务过程中 TQ0OVF0 和 TQ0OVF1 标记设为 1，且溢出标记清零。

<4> 读取 TQ0CCR0 寄存器。

读取 TQ0OVF0 标记。如果 TQ0OVF0 标记为 1，则其清零。

因为 TQ0OVF0 标记为 1，所以脉宽可通过 $(10000H + D01 - D00)$  计算。

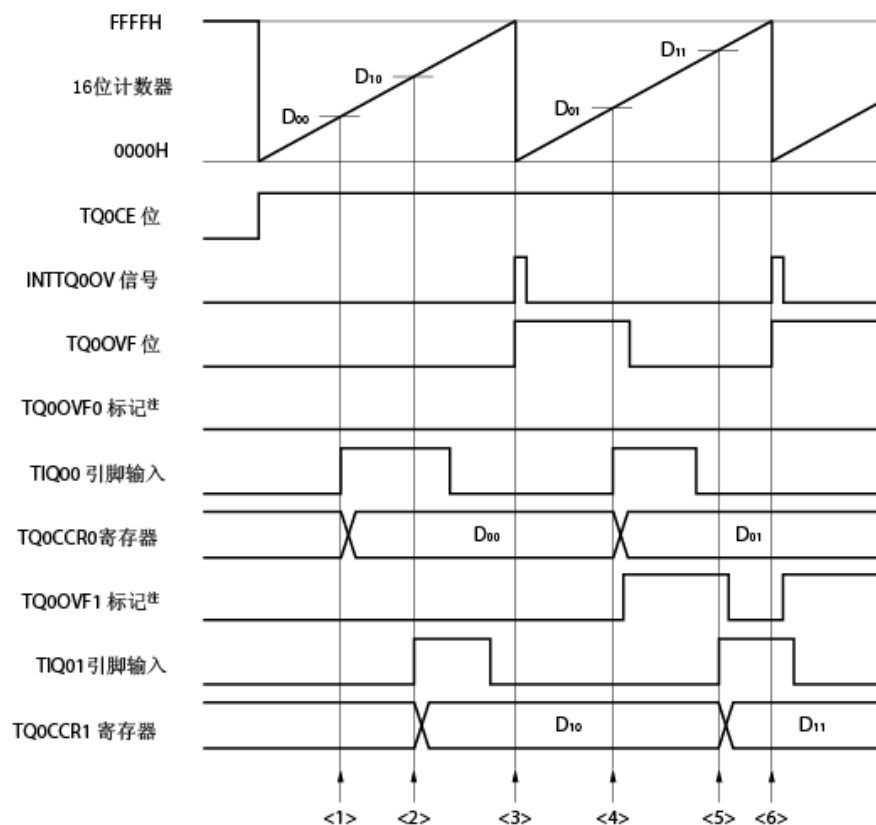
<5> 读取 TQ0CCR1 寄存器。

读取 TQ0OVF1 标记。如果 TQ0OVF1 标记为 1，则其清零(在<4>TQ0OVF0 标记清零，且 TQ0OVF1 标记保持 1)。

因为 TQ0OVF1 标记为 1，脉宽可通过 $(10000H + D11 - D10)$ (校正) 计算。

<6> 与<3>相同

使用两个捕获寄存器时的示例(不使用溢出中断)



注 TQ0OVF0 和 TQ0OVF1 标记通过软件在内部 RAM 设置。

<1> 读取 TQ0CCR0 寄存器(TIQ00 引脚输入的默认值设置)。

<2> 读取 TQ0CCR1 寄存器(TIQ01 引脚输入的默认值设置)。

<3> 发生溢出。不能通过软件进行操作。

<4> 读取 TQ0CCR0 寄存器。

读取溢出标记。如果溢出标记为 1，仅 TQ0OVF1 标记设为 1，且溢出标记清零。

因为溢出标记为 1，脉宽可通过 $(10000H + D01 - D00)$  计算。

<5> 读取 TQ0CCR1 寄存器。

读取溢出标记。因为在<4>溢出标记清零，所以读取 0。

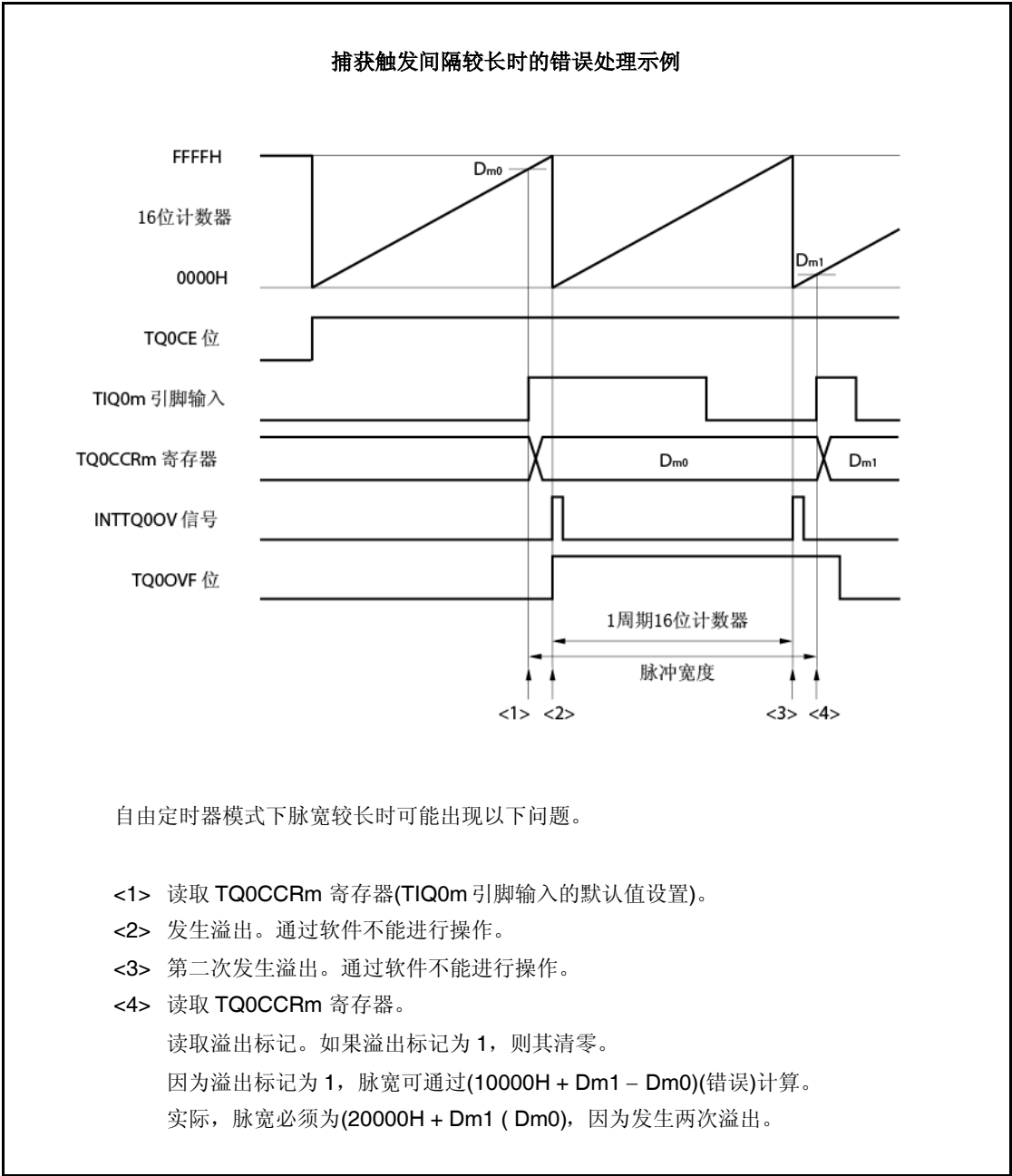
读取 TQ0OVF1 标记。如果 TQ0OVF1 标记为 1，则其清零。

因为 TQ0OVF1 标记为 1，脉宽可通过 $(10000H + D11 - D10)$ (校正) 计算。

<6> 与<3>相同

(d) 捕获触发间隔较长时的溢出处理

如果脉宽大于 16-位计数器的一周期，必须小心，因为从第一次捕获触发到下一次可能出现一次以上溢出。首先，错误处理的示例如下所示。

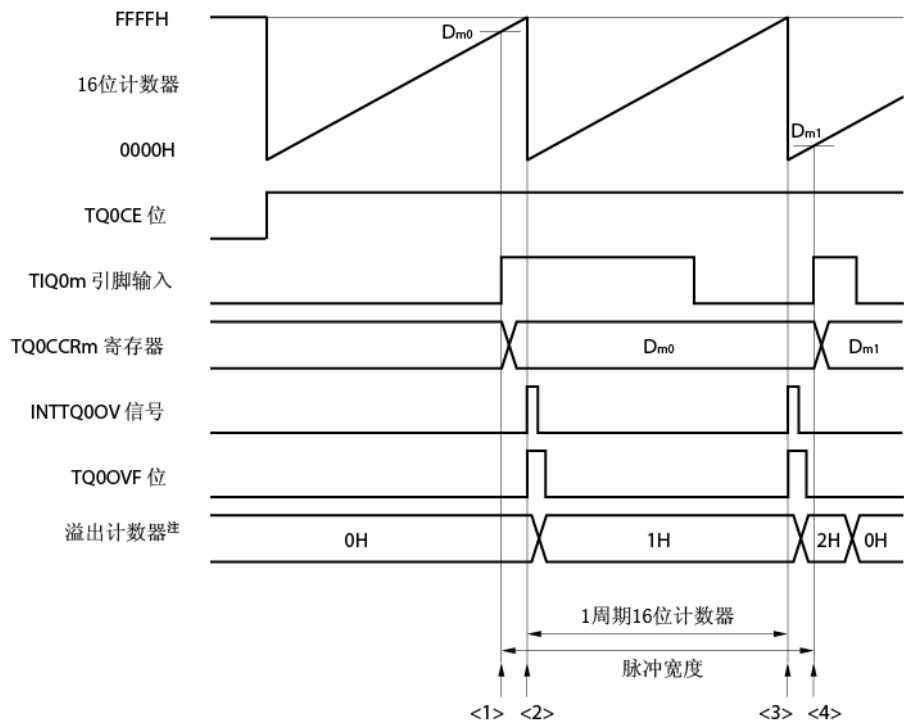


如果当捕获触发间隔较长时发生两次或两次以上溢出，则不能获得校正脉宽。

如果捕获触发间隔较长，减慢计数时钟以延长 16-位计数器的一周期或使用软件。如何使用软件的示例如下所示。



当捕获触发间隔较长时的示例

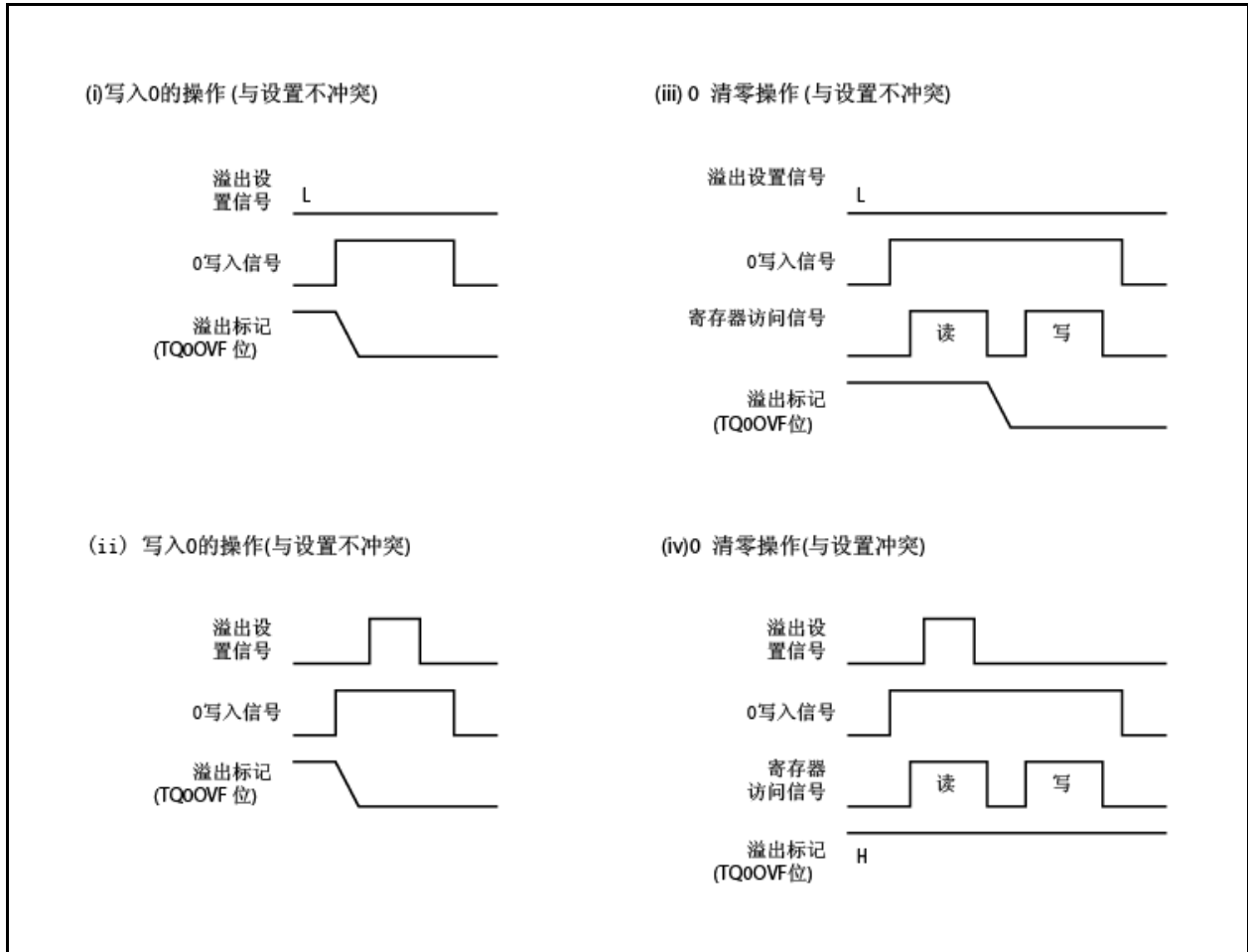


注 通过内部 RAM 的软件任意设置溢出计数器。

- <1> 读取 TQ0CCRm 寄存器(TIQ0m 引脚输入默认值设置)。
- <2> 发生溢出。溢出中断服务中溢出计数器加一溢出标记清零。
- <3> 第二次发生溢出。溢出中断服务中溢出计数器加一(+1) 且溢出标记清零。
- <4> 读取 TQ0CCRm 寄存器。  
读取溢出计数器。  
→ 当溢出计数器为“N”时，脉宽可通过 $(N \times 10000H + D_{m1} - D_{m0})$  计算。  
在此示例中，脉宽为 $(20000H + D_{m1} - D_{m0})$ ，因为发生两次溢出。  
溢出计数器清零(0H)。

(e) 溢出标记清零

通过 TQ0OVF 位清零以 CLR 指令和通过将 8-位数据(位 0 为 0) 写入 TQ0OPT0 寄存器溢出标记可清零。要准确检测溢出，当为 1 时请读取 TQ0OVF 位，且接着使用位操作指令使溢出标记清零。



要使溢出标记清零，请读取溢出标记以核对其是否设为 1，且以 CLR 指令清零。如果 0 写入溢出标记而不核对标记是否为 1，则溢出设置信息可通过写入 0((ii)以上图擦除。因此，即使当实际发生溢出时，软件可能判断未发生溢出。

如果执行 CLR 指令与当溢出标记以 CLR 指令清零时的溢出发生时冲突，即使执行清零指令之后溢出标记也保持设置。

### 7.5.7 脉宽测量模式(TQ0MD2 到 TQ0MD0 位 = 110)

脉宽测量模式下，当 TQ0CTL0.TQ0CE 位设为 1 时 16-位定时器/事件计数器 Q 开始计数。每当检测到输入到 TIQ0m 引脚的有效沿时，16-位计数器的计数值存储在 TQ0CCRm 寄存器，且 16-位计数器清为 000H。

捕获中断请求信号(INTTQ0CCm)出现之后有效沿的间隔可通过读取 TQ0CCRm 寄存器测量。

选择 TIQ00 到 TIQ03 引脚作为捕获触发输入引脚。通过使用未用引脚的 TQ0IOC1 寄存器指定“未检测到边沿”。

当外部时钟用作计数时钟时，因为外部时钟固定为 TIQ00 引脚，所以测量 TIQ0k 引脚的脉宽。此时，TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位清零 (捕获触发输入(TIQ00 引脚)：未检测到边沿)。

备注      m = 0 到 3  
              k = 1 到 3

图 7-34. 脉宽测量模式下的配置

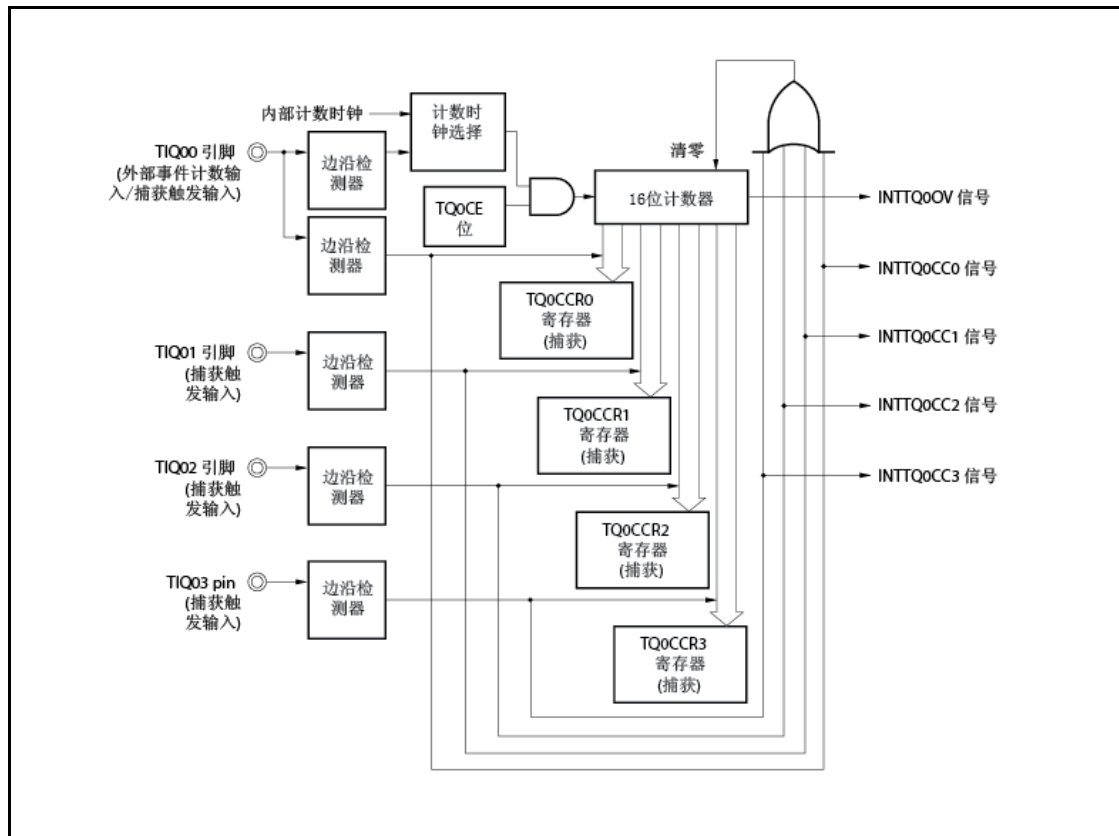
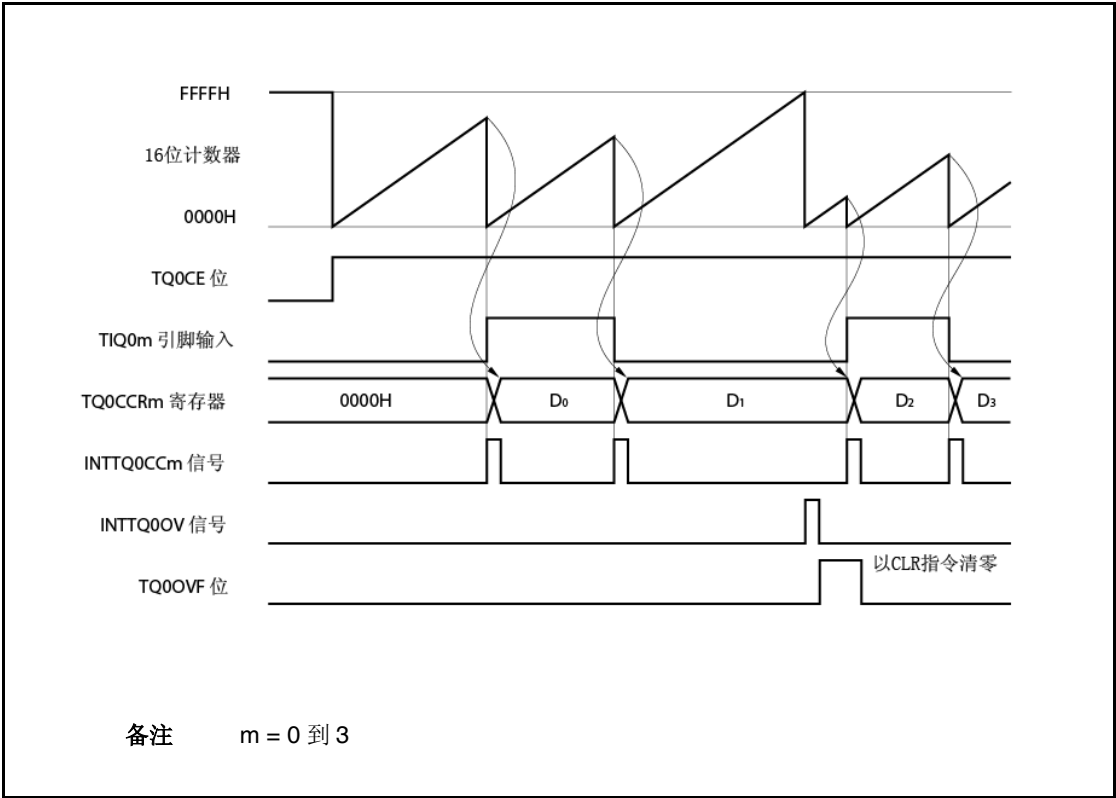


图 7-35. 脉宽测量模式下的基本时序



当 TQ0CE 位设为 1 时，16-位计数器开始计数。当稍后检测到输入到 TIQ0m 引脚的有效沿时，16-位计数器的计数值存储在 TQ0CCRm 寄存器，16-位计数器清为 000H，且产生捕获中断请求信号 (INTTQ0CCm)。

脉宽计算如下所示。

脉宽=捕获值×计数时钟周期

如果即使当 16-位计数器计数上至 FFFFH 时有效沿也未输入到 TIQ0m 引脚，则在下一计数时钟脉冲产生溢出中断请求信号 (INTTQ0OV)，且计数器清零并继续计数。此时，溢出标记 (TQ0OPT0.TQ0OVF 位) 也设为 1。溢出标记通过经软件执行 CLR 指令清零。

如果溢出标记设为 1，脉宽可如下所示进行计算。

脉宽=(10000H × TQ0OVF 位设置(1) 计数+捕获值) × 计数时钟周期

备注 m = 0 到 3

图 7-36. 脉宽测量模式下寄存器的设置(1/2)

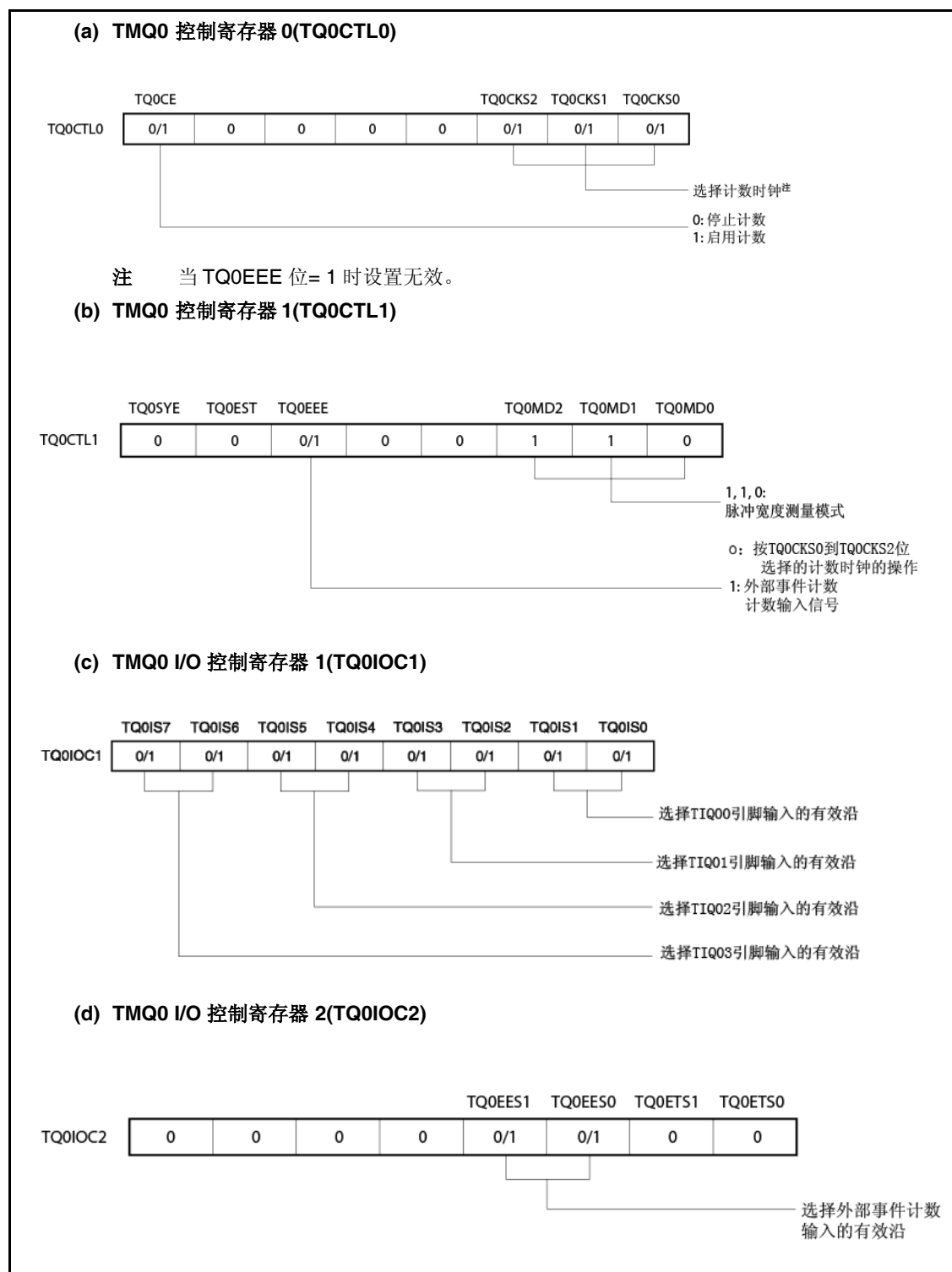
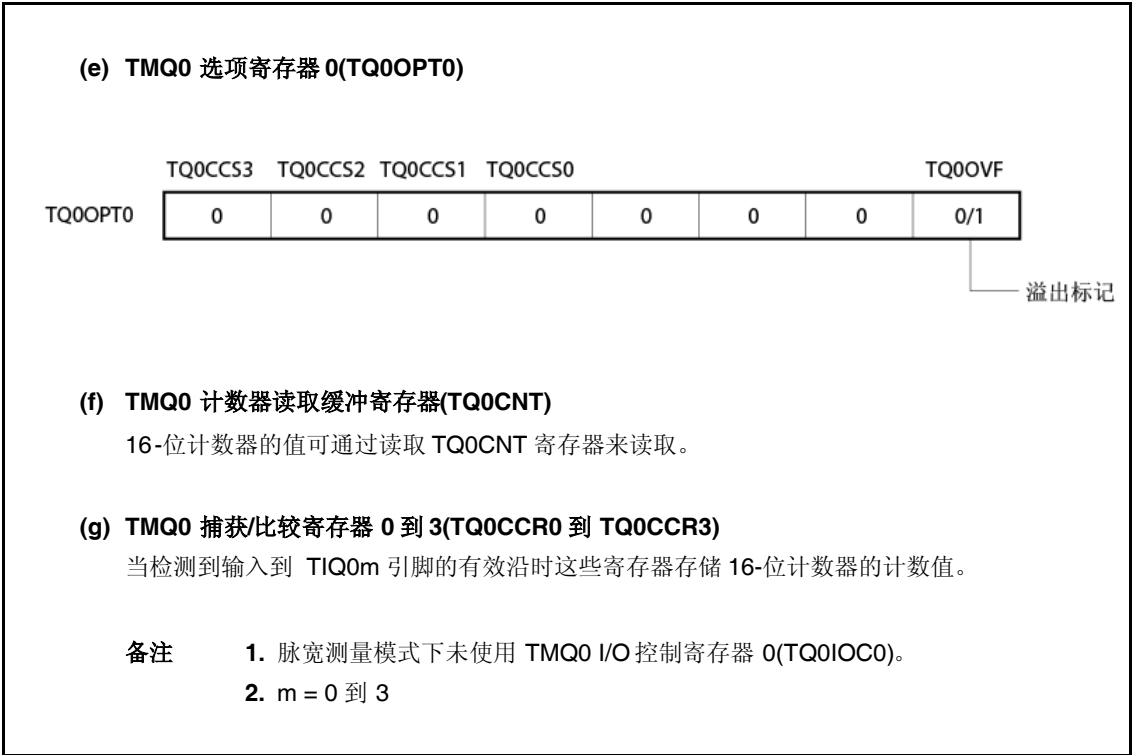
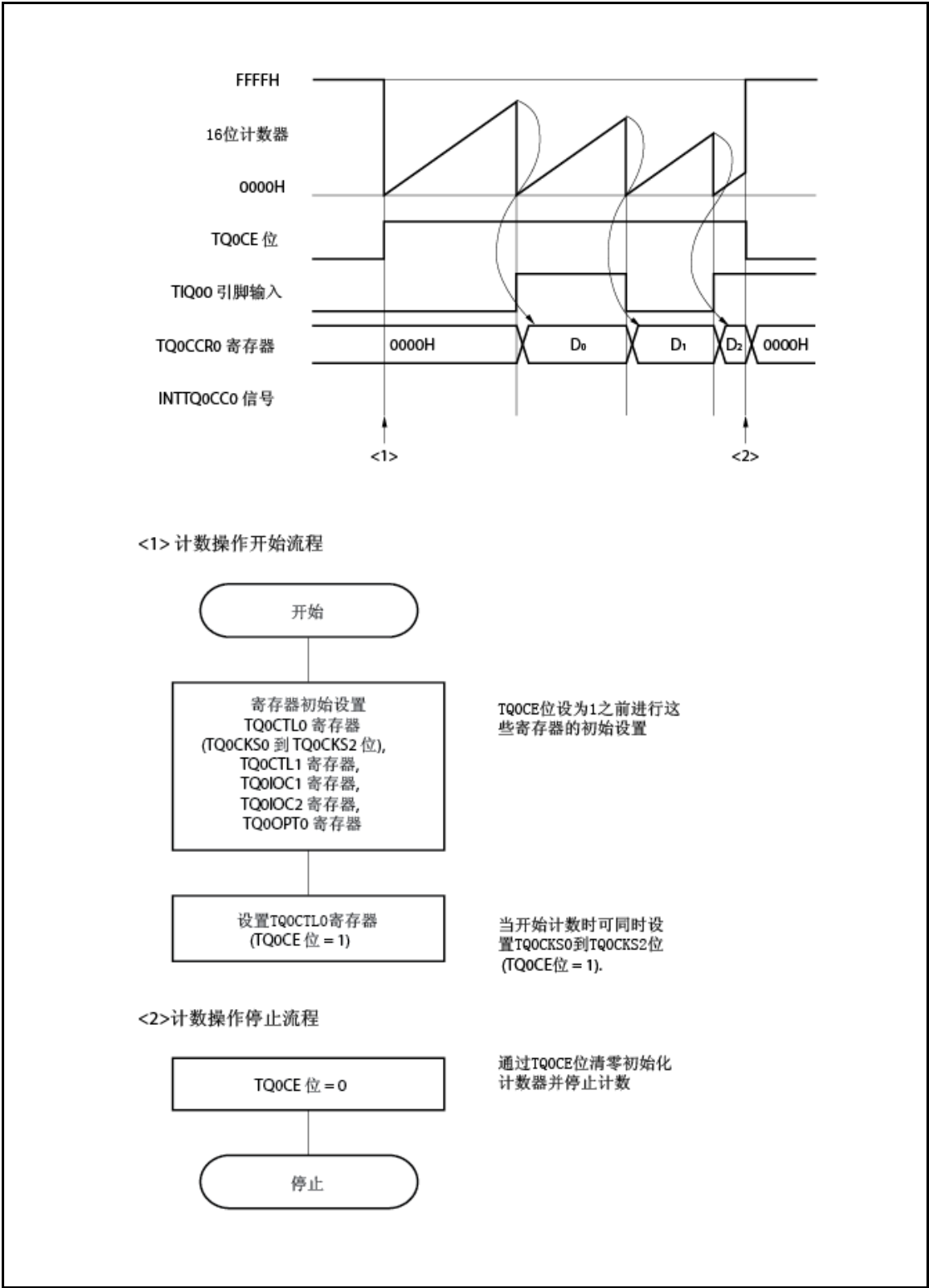


图 7-36. 脉宽测量模式下寄存器的设置(2/2)



(1) 脉宽测量模式下的操作流程

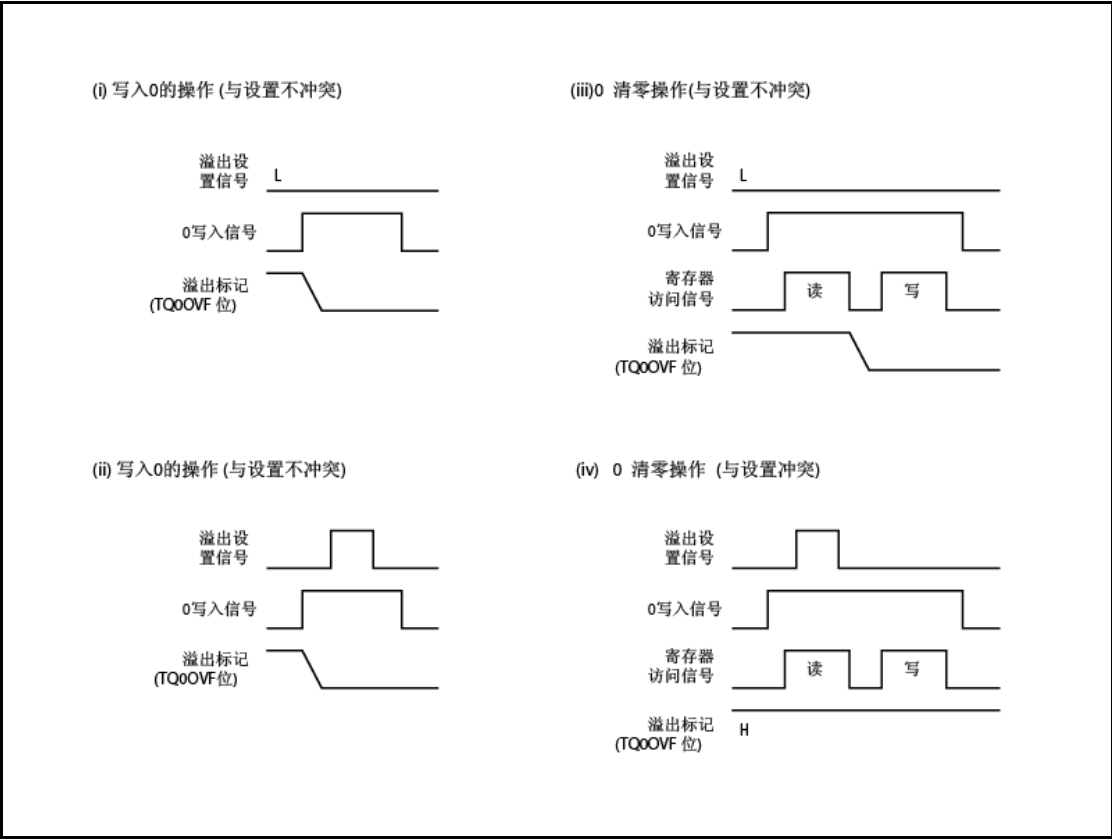
图 7-37. 脉宽测量模式下的软件处理流程



(2) 脉宽测量模式下的操作时序

(a) 溢出标记清零

溢出标记可通过 TQ0OVF 位清零以 CLR 指令和通过将 8-位数据(位 0 is 0)写入 TQ0OPT0 寄存器清零。要准确检测溢出，当为 1 时请读取 TQ0OVF 位，且接着使用位操作指令将溢出标记清零。



要使溢出标记清零，请读取溢出标记以核对其是否设为 1，且以 CLR 指令清零。如果 0 写入溢出标记而不核对标记是否为 1，则溢出设置信息可通过写入 0((ii)以上图擦除。因此，即使当实际发生溢出时，软件可能判断未发生溢出。

如果执行 CLR 指令与当溢出标记以 CLR 指令清零时的溢出发生时冲突，即使执行清零指令之后溢出标记也保持设置。



7.5.8 PWM 模式(TQ0MD2 到 TQ0MD0 = 111) 的三角波

在三角波 PWM 模式下，使用 TMQ0 捕获/比较寄存器k(TQ0CCRk)设置占空因数，且使用 TMQ0 捕获/比较寄存器 0(TQ0CCR0)设置周期。

使用这些四个寄存器且运行定时器，输出具有可变周期的三角波 PWM。

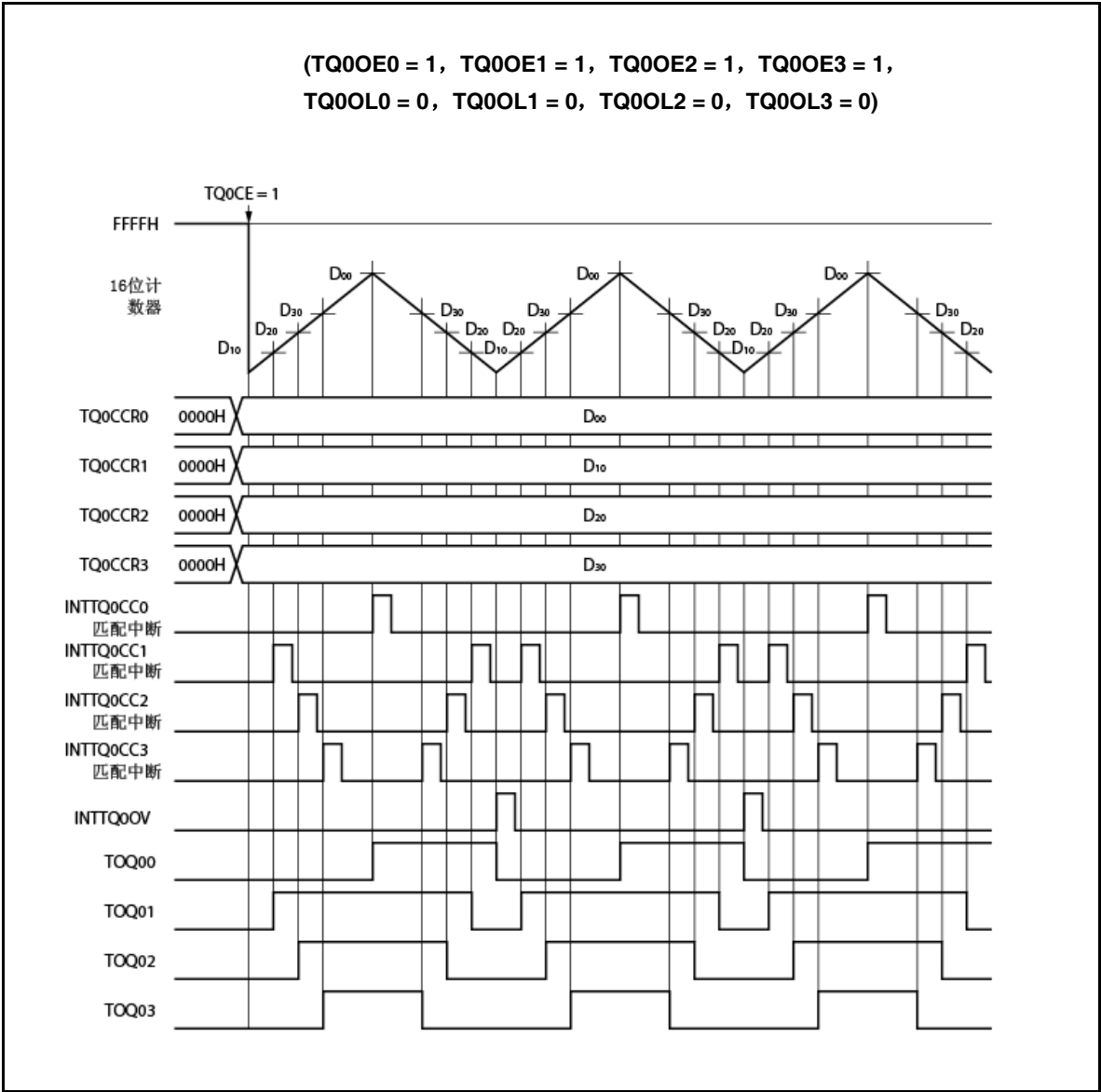
当 TQ0CE = 1 时可重写 TQ0CCRm 寄存器的值。

要停止定时器 Q，请将 TQ0CE 清零。PWM 的波形从 TOQ0k 引脚输出。当 16-位计数器的值匹配 TQ0CCR0 寄存器的值且当计数器下溢时 TOQ00 引脚产生触发输出。

**注意事项** 在 PWM 模式下，TQ0CCRm 寄存器的捕获功能不能使用，因为此寄存器仅可用作比较寄存器。

**备注** m = 0 到 3, k = 1 到 3

图 7-38. 在三角波 PWM 模式下的基本操作时序



### 7.5.9 定时器输出操作

下表给出了操作和输出 TOQ00 到 TOQ03 引脚的电平。

表 7-6. 每种模式下的定时器输出控制

操作模式	TOQ00 引脚	TOQ01 引脚	TOQ02 引脚	TOQ03 引脚
计时器模式	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	方波输出	外部触发脉冲输出	外部触发脉冲输出	外部触发脉冲输出
单触发脉冲输出模式		单触发脉冲输出	单触发脉冲输出	单触发脉冲输出
PWM 输出模式		PWM 输出	PWM 输出	PWM 输出
自由定时器模式	方波输出(仅当使用比较功能时)			
脉宽测量模式	—			
三角波 PWM 输出模式	方波输出	三角波 PWM 输出	三角波 PWM 输出	三角波 PWM 输出

表 7-7. 定时器输出控制位控制下的 TOQ00 到 TOQ03 引脚真值表

TQ0IOC0.TQ0OLm 位	TQ0IOC0.TQ0OEm 位	TQ0CTL0.TQ0CE 位	TOQ0m 引脚电平
0	0	×	低-电平输出
	1	0	低-电平输出
		1	计数之前即为低电平, 计数开始之后即为高电平
1	0	×	高-电平输出
	1	0	高-电平输出
		1	计数之前即为高电平, 开始计数之后即为低电平

备注 m = 0 到 3

### 7.6 定时器调试操作功能

定时器 P 和定时器 Q 具有定时器调试操作功能。  
可同步的定时器在表 7-8 给出。

表 7-8. 定时器调整操作模式

主定时器	从定时器	
TMP0	TMP1	—
TMP2	TMP3	TMQ0

- 注意事项
1. 通过 TPmCTL1.TPmSYE 和 TQ0CTL1.TQ0SYE 位启用或禁用调试操作模式。对于 TMQ2, TMQ3 和 TMQ0 的任一者或两者可指定为从属。
  2. 使用以下程序设置调试操作模式。
    - <1> 设置从定时器的 TPmCTL1.TPmSYE 和 TQ0CTL1.TQ0SYE 位以启用调试运行。  
从定时器的 TPmCTL1.TPmMD2 到 TPmCTL1.TPmMD0 和 TQ0CTL1.TQ0MD2 到 TQ0CTL1.TQ0MD0 位设为自由模式。
    - <2> 使用 TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0 位设置定时器模式。  
此时, 禁止设置主定时器的 TPnCTL1.TPnSYE 位。
    - <3> 设置主、从定时器的比较寄存器值。
    - <4> 设置从定时器的 TPmCTL0.TPmCE 和 TQ0CTL0.TQ0CE 位以启用内部运行 时钟操作。
    - <5> 设置主定时器的 TPnCTL0.TPnCE 位以启用内部运行时钟操作。

备注 m = 1, 3

表 7-9 和 7-10 给出了可用于调试操作模式的定时器模式(√: 可设置, ×: 不可设置)。

表 7-9. 可用于调试操作模式的定时器模式

主定时器	自由模式	PWM 模式	三角波 PWM 模式
TMP0	√	√	×
TMP2	√	√	×

表 7-10. 定时器输出功能

调试通道	定时器	引脚	自由模式		PWM 模式		三角波 PWM 模式	
			调试关	调试开	调试关	调试开	调试关	调试开
Ch0	TMP0 (主)	TOP00	PPG	←	触发	←	N/A	←
		TOP01	PPG	←	PWM	←	N/A	←
	TMP1 (从)	TOP10	PPG	←	触发	PWM	N/A	←
		TOP11	PPG	←	PWM	←	N/A	←
Ch1	TMP2 (主)	TOP20	PPG	←	触发	←	N/A	←
		TOP21	PPG	←	PWM	←	N/A	←
	TMP3 (从)	TOP30	PPG	←	触发	PWM	N/A	←
		TOP31	PPG	←	PWM	←	N/A	← (
	TMQ0(从)	TOQ00	PPG	←	触发	PWM	触发	N/A
		TOQ01 到 TOQ03	PPG	←	PWM	←	三角波 PWM	N/A

备注        数据从主定时器的比较寄存器传输到从定时器的比较寄存器的时序如下所示。

PPG:

触发, PWM, 三角波 PWM:

CPU 写入时序

定时器计数器和比较寄存器匹配 TOPn0 和 TOQ00 的时序(n = 0 到 3)

图 7-39. 调试操作图(TMP2, TMP3, TMQ0)

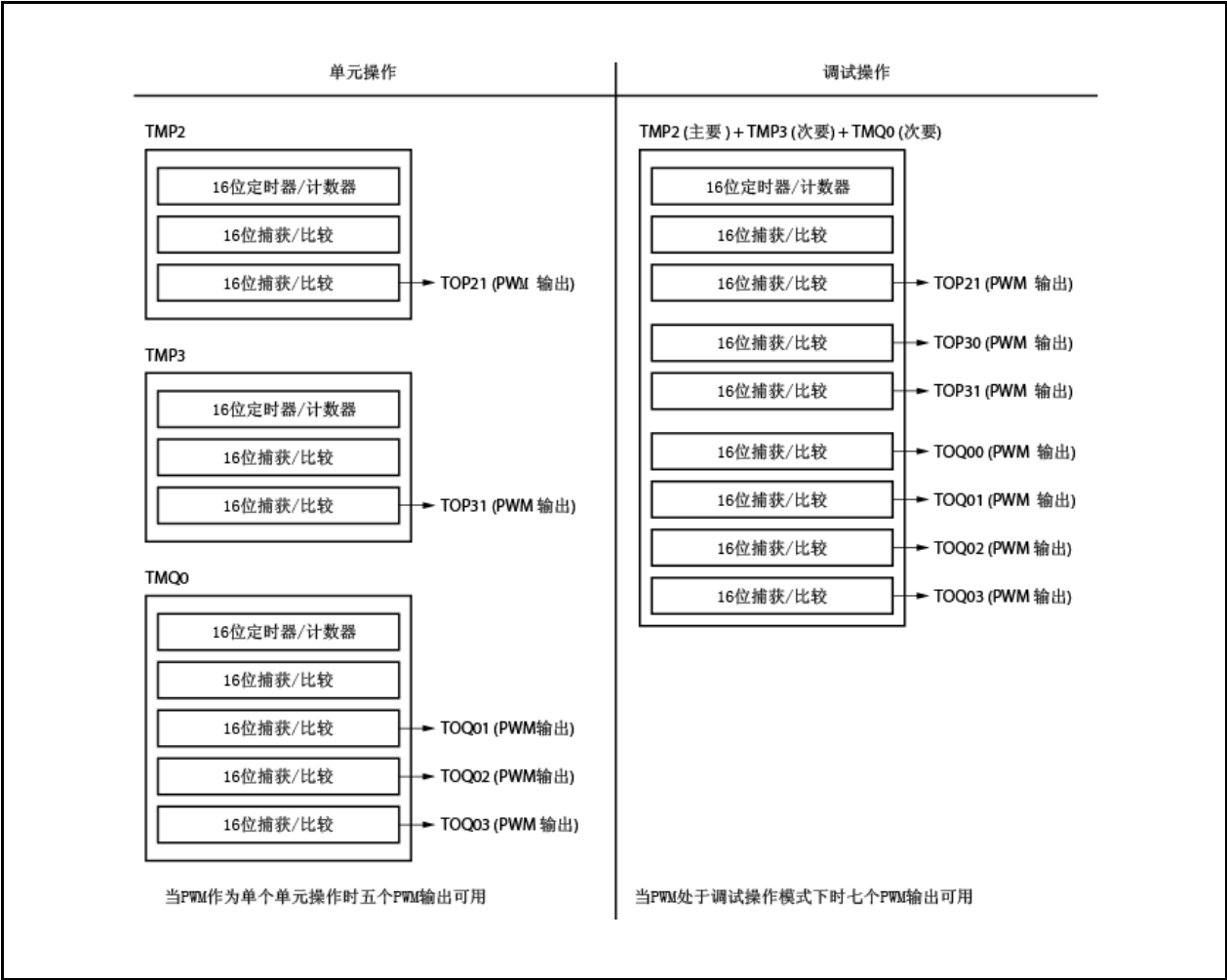
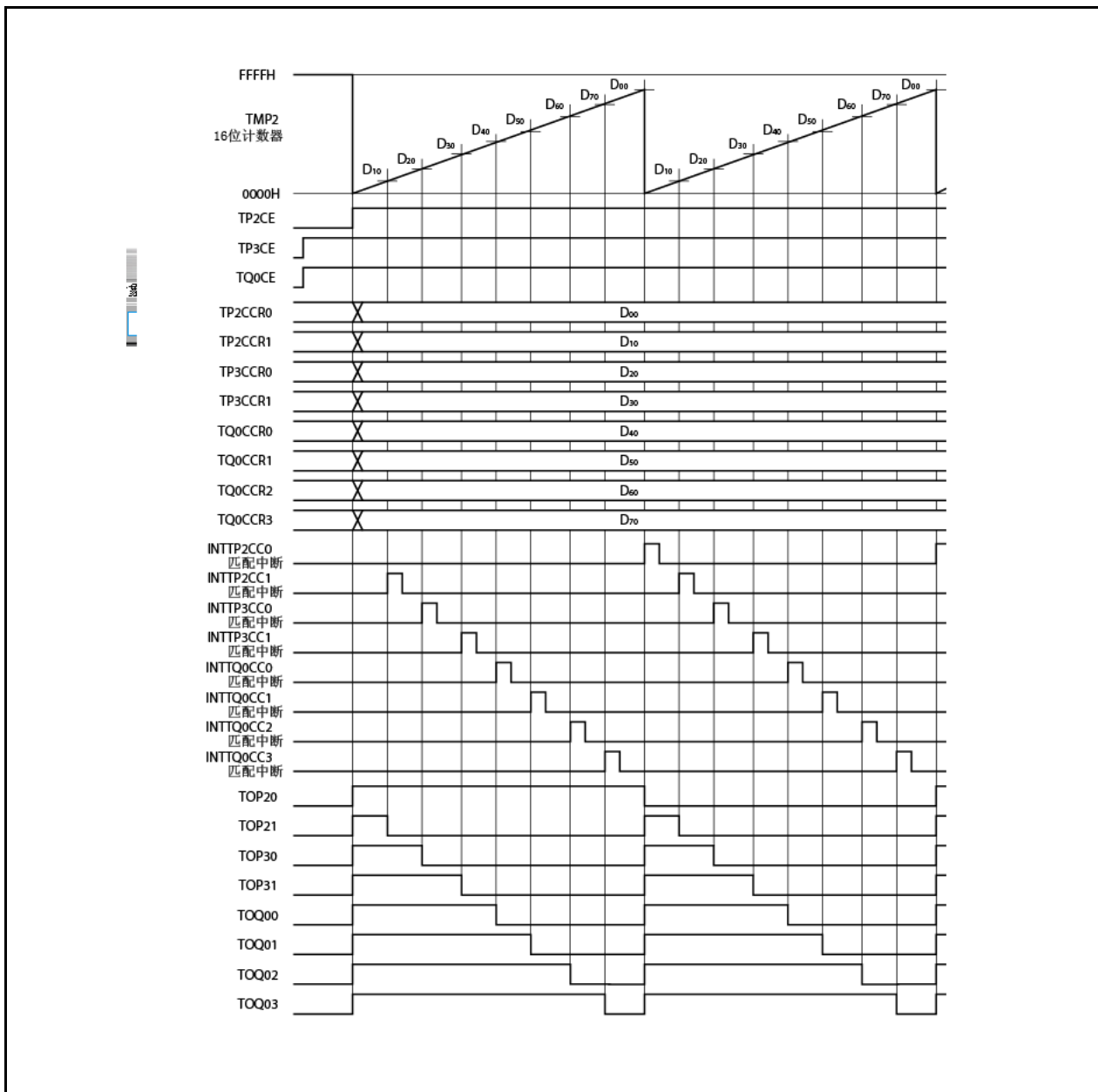


图 7-40. 调试 PWM 功能(TMP2, TMP3, TMQ0)的基本操作时序

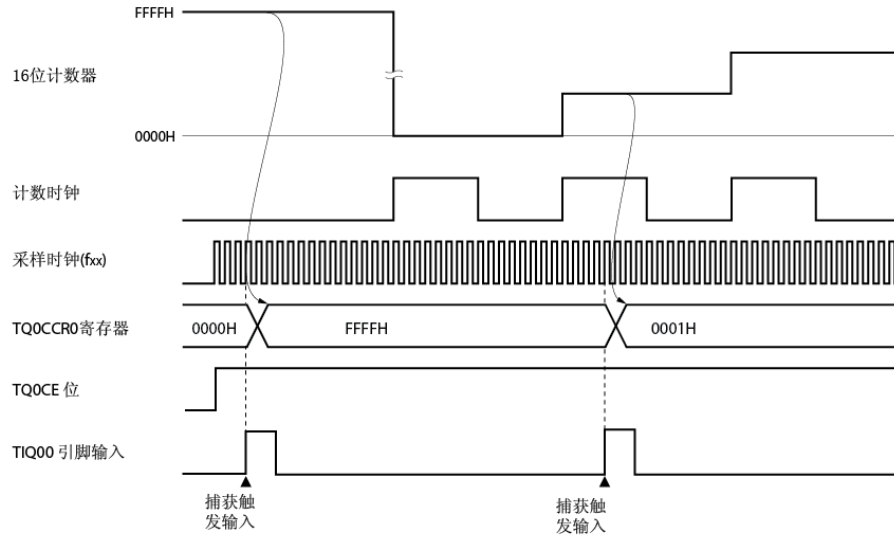


## 7.7 注意事项

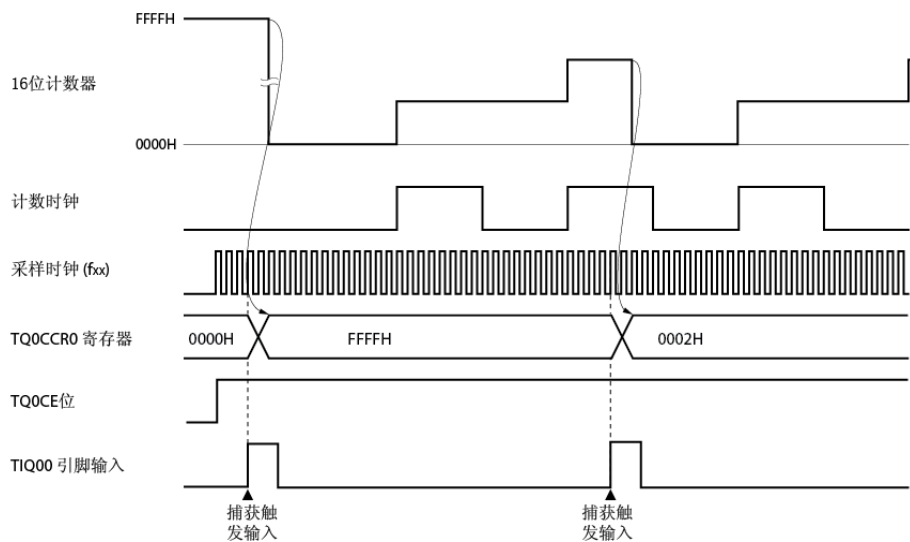
### (1) 捕获操作

当使用捕获操作且较慢时钟选作计数时钟时，如果 TQ0CE 位设为 1 之后立即输入捕获触发，则在 TQ0CCR0, TQ0CCR1, TQ0CCR2 及 TQ0CCR3 寄存器可捕获 FFFFH 而非 0000H。

#### (a) 自由定时器模式



#### (b) 脉宽测量模式



## 第八章 16-位间隔定时器 M (TMM)

### 8.1 概述

- 间隔功能
- 可选 8 时钟
- 16-位计数器× 1  
(定时器计数运行期间不能读取 16-位计数器。)
- 比较寄存器×1  
(定时器计数运行期间不能写入比较寄存器。)
- 比较匹配中断× 1

定时器 M 仅支持清零&开始模式。不支持自由定时器模式。



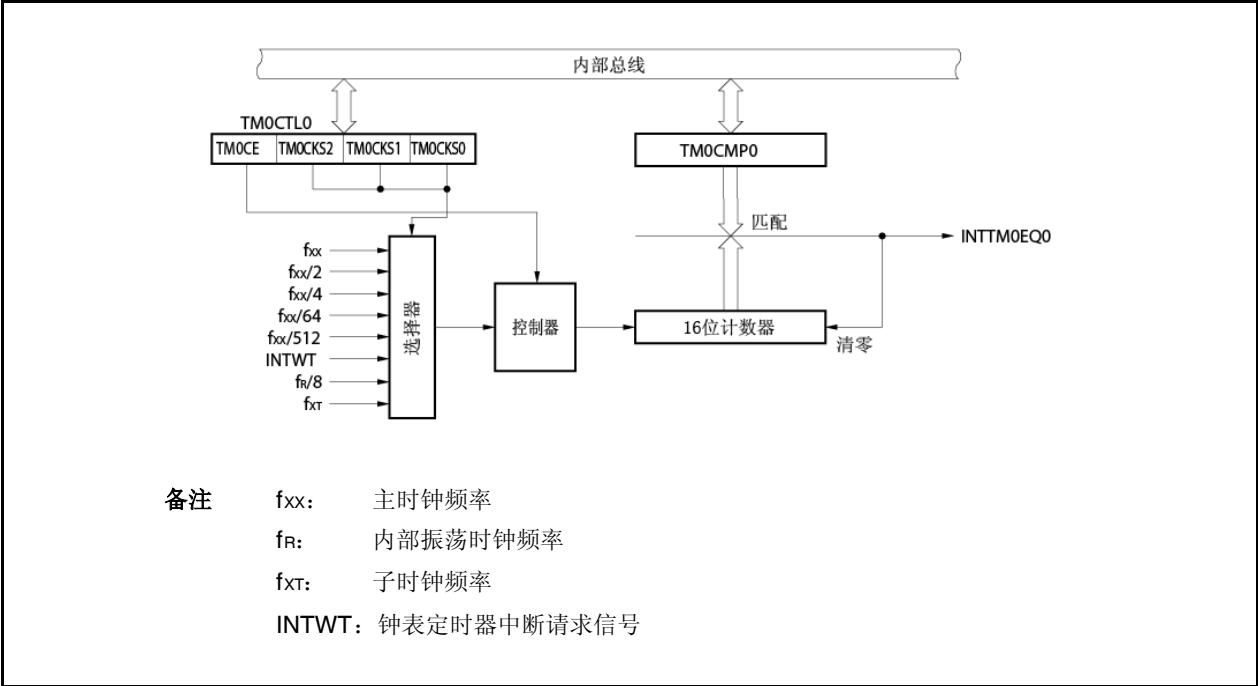
8.2 配置

TMM0 包括以下硬件。

表 8-1. TMM0 配置

项目	配置
定时器寄存器	16-位计数器
寄存器	TMM0 比较寄存器 0(TM0CMP0)
控制寄存器	TMM0 控制寄存器 0(TM0CTL0)

图 8-1. TMM0 框图

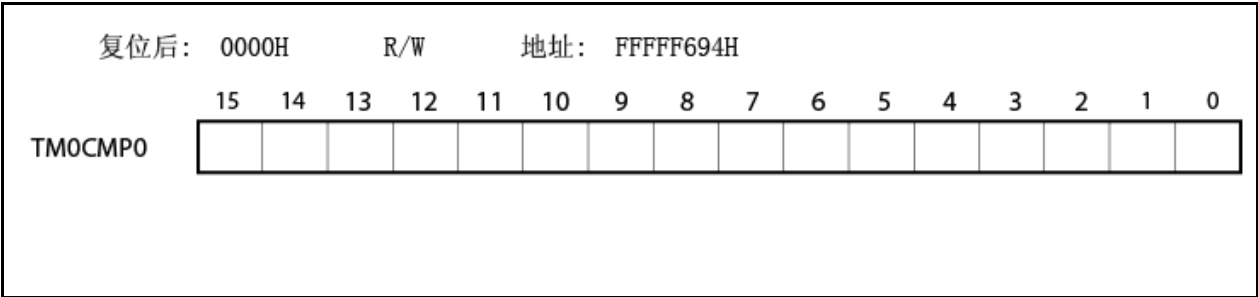


(1) 16-位计数器

这是内部时钟计数的 16-位计数器。  
不能读取或写入 16-位计数器。

(2) TMM0 比较寄存器 0(TM0CMP0)

TM0CMP0 寄存器为 16-位比较寄存器。  
此寄存器可读取或写入 16-位单元。  
此寄存器重设为 0000H。  
通过软件 TM0CMP0 寄存器始终可写入相同值。  
当 TM0CTL0.TM0CE 位= 1 时禁止 TM0CMP0 寄存器重写。



8.3 寄存器

(1) TMM0 控制寄存器(TMOCTL0)

TMOCTL0 寄存器为控制 TMM0 操作的 8-位寄存器。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

通过软件 TMOCTL0 寄存器始终可写入相同值。重写此寄存器， TMOCE 位除外，当定时器运行时禁止重写。

复位后: 00H

R/W

地址: FFFFF690H

7	6	5	4	3	2	1	0	
TMOCTL0	TMOCE	0	0	0	0	TMOCKS2	TMOCKS1	TMOCKS0

TMOCE	内部时钟运行启用/禁用规格
0	TMM0操作禁用（16-位计数器异步复位）。 运行时钟应用程序停止。
1	TMM0操作启用。运行时钟应用程序开始。TMM0操作开始。
TMM0的内部时钟控制和内部电路复位与TMOCE位异步进行。 当TMOCE位清零时，TMM0的内部时钟禁用（固定为低电平） 且16位计数器异步复位。	

TMOCKS2	TMOCKS1	TMOCKS0	计数时钟选择
0	0	0	f <sub>xx</sub>
0	0	1	f <sub>xx</sub> /2
0	1	0	f <sub>xx</sub> /4
0	1	1	f <sub>xx</sub> /64
1	0	0	f <sub>xx</sub> /512
1	0	1	INTWT
1	1	0	f <sub>in</sub> /8
1	1	1	f <sub>xt</sub>

- 注意事项 1. 当 TMOCE 位= 0 时设置 TMOCKS2 到 TMOCKS0 位。  
当值 TMOCE 从 0 改为 1 时，不能同时设置 TMOCKS2 到 TMOCKS0 位的值。
2. 确保位 3 到 6 清零。

备注 f<sub>xx</sub>: 主时钟频率  
f<sub>in</sub>: 内部振荡时钟频率  
f<sub>xt</sub>: 子时钟频率

8.4 运行

注意事项 禁止将 TM0CMP0 寄存器设为 FFFFH.

8.4.1 计时器模式

在计时器模式下，如果 TM0CTL0.TM0CE 位设为 1 则以指定间隔产生中断请求信号(INTTM0EQ0)。

图 8-2 计时器配置

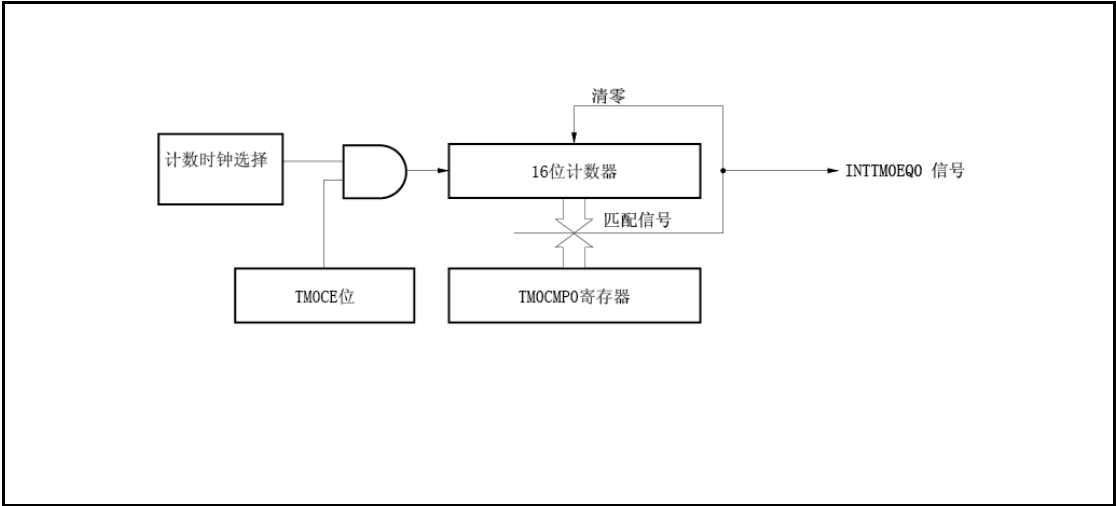
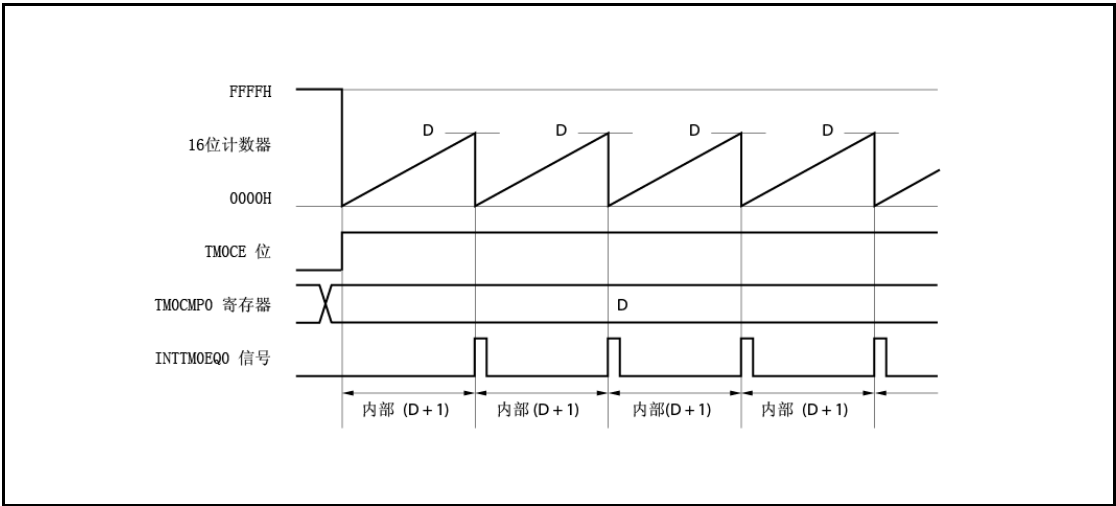


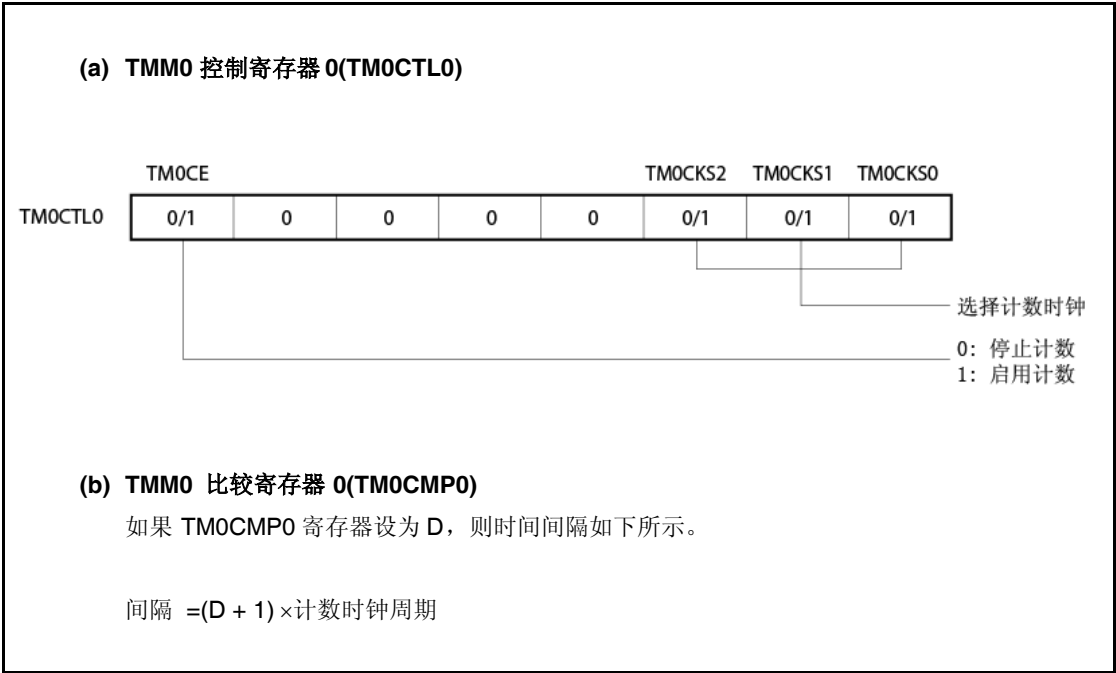
图 8-3. 在计时器模式下操作基本时序



当 TM0CE 位设为 1 时，16-位计数器的值与计数时钟同步从 FFFFH 清为 0000H，且计数器开始计数。  
当 16-位计数器的计数值匹配 TM0CMP0 寄存器的值时，16-位计数器清为 000H 且产生比较匹配中断请求信号(INTTM0EQ0)。  
间隔可通过以下表达式计算。

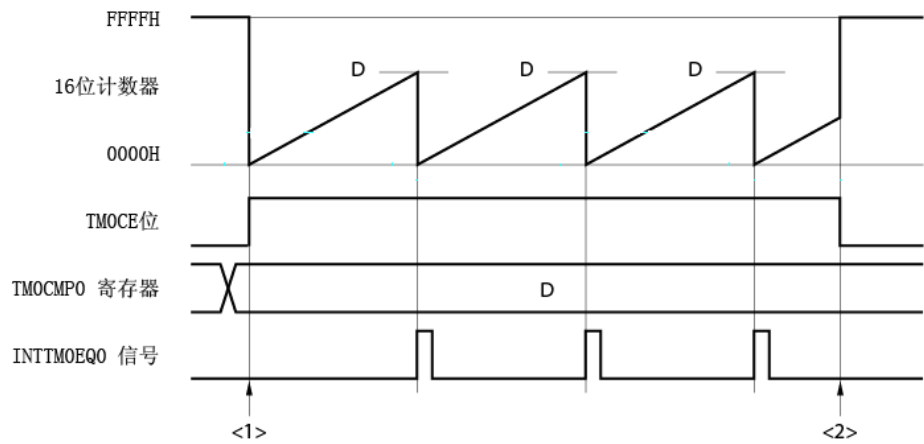
间隔 = (设置 TM0CMP0 寄存器的值+ 1)× 计数时钟周期

图 8-4. 计时器模式操作的寄存器设置

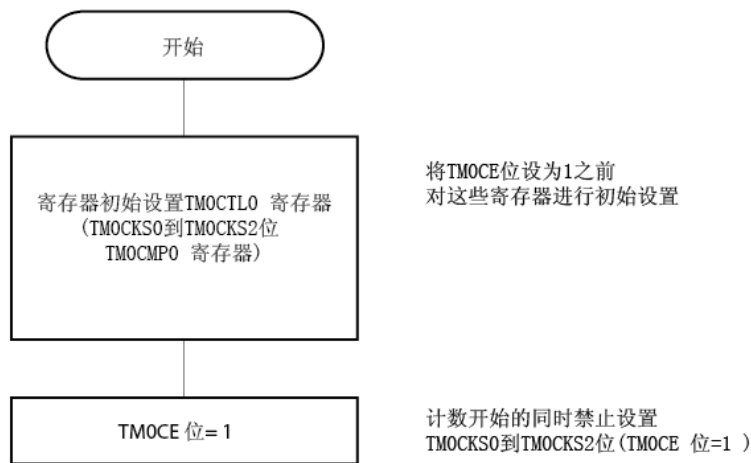


(1) 计时器模式操作流程

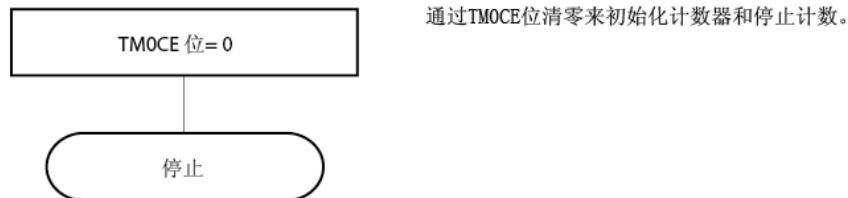
图 8-5. 在计时器模式下的软件处理流程



<1> 计数操作开始流程



<2>计数操作停止流程

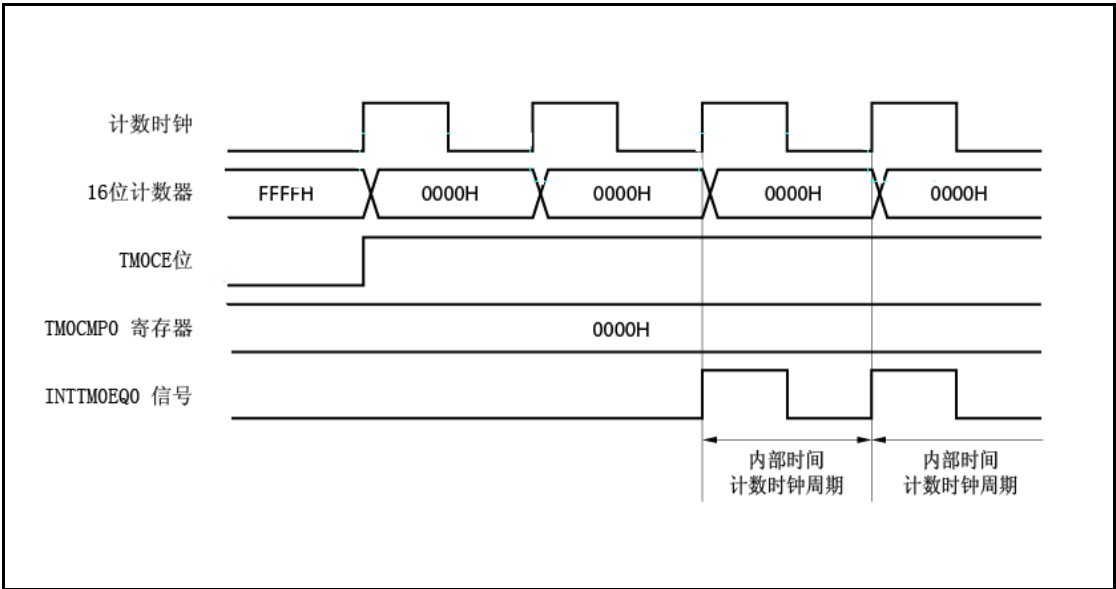


(2) 计时器模式运行时序

注意事项 禁止将 **TM0CMP0** 寄存器设为 **FFFFH**。

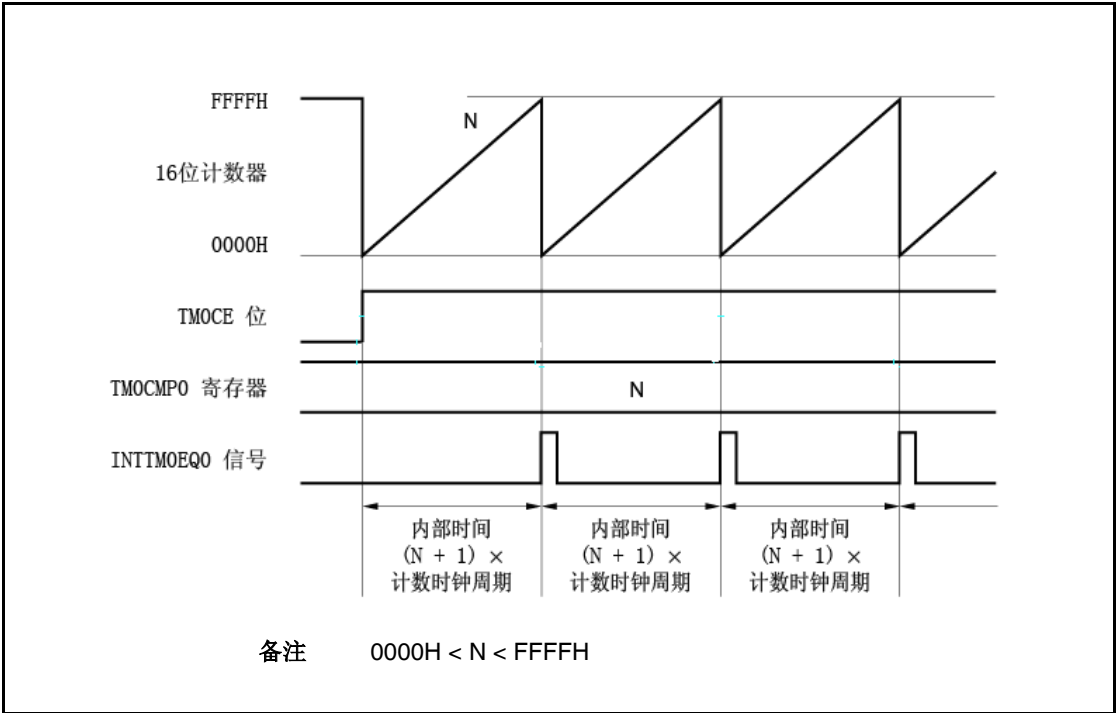
(a) **TM0CMP0** 寄存器设为 **0000H** 时的操作

如果 **TM0CMP0** 寄存器设为 **0000H**，则在每一计数时钟脉冲处产生 **INTTM0EQ0** 信号。  
16-位计数器的值始终为 **0000H**。



(b) **TM0CMP0** 寄存器设为 **N** 时的操作

如果 **TM0CMP0** 寄存器设为 **N**，则 16-位计数器计数上至 **N**。计数器与下一计数时序同步清零且产生 **INTTM0EQ0** 信号。



8.4.2 注意事项

(1) TM0CTL0.TM0CE 位设为 1 之后，根据计数时钟选择，需花费 16-位计数器以下时间开始计数。

选择计数时钟	最大时间之前计数开始
$f_{xx}$	$2/f_{xx}$
$f_{xx}/2$	$6/f_{xx}$
$f_{xx}/4$	$24/f_{xx}$
$f_{xx}/64$	$128/f_{xx}$
$f_{xx}/512$	$1024/f_{xx}$
INTWT	INTWT 信号的第二个上升沿
$f_R/8$	$16/f_R$
$f_{XT}$	$2/f_{XT}$

(2) 当 TMM0 运行时禁止重写 TM0CMP0 和 TM0CTL0 寄存器。  
如果当 TM0CE 位为 1 时重写这些寄存器，则不能保证操作。  
如果其误重写，则 TM0CTL0.TM0CE 位清零，且寄存器复位。

## 第九章 钟表定时器功能

### 9.1 功能

钟表定时器具有以下功能。

- 钟表定时器： 中断请求信号(INTWT)使用主时钟或副时钟以 0.5 或 0.25 秒间隔产生。
- 计时器： 中断请求信号(INTWTI)以设置间隔产生。

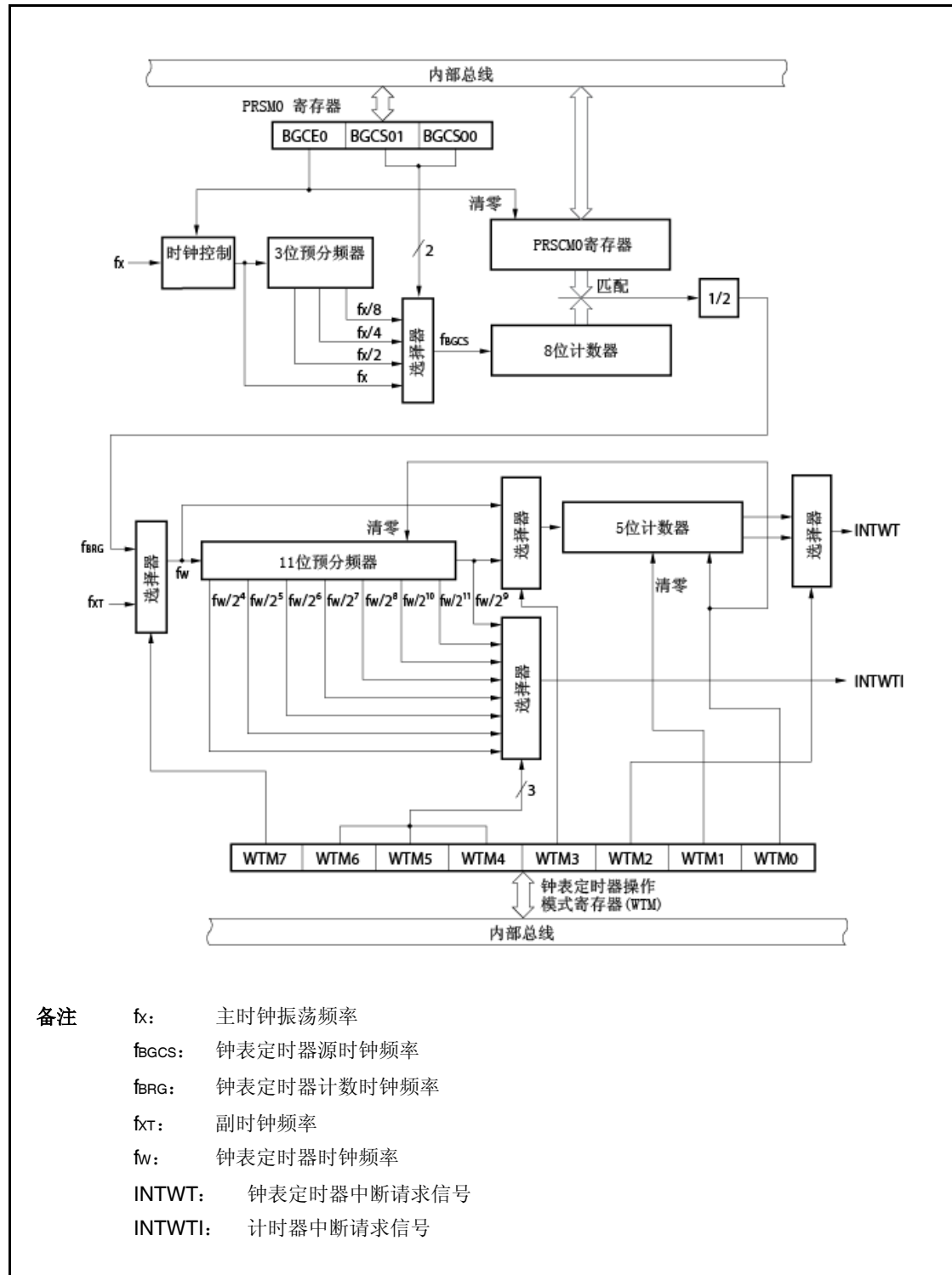
钟表定时器和计时器功能可同时使用。



## 9.2 配置

钟表定时器的框图如下所示。

图 9-1. 钟表定时器的框图



**(1) 时钟控制**

当钟表定时器对主时钟进行操作时此块控制供应和停止操作时钟( $f_x$ )。

**(2) 3-位预分频器**

此预分频器分割  $f_x$  以产生  $f_x/2$ ,  $f_x/4$  或  $f_x/8$ 。

**(3) 8-位计数器**

此 8-位计数器对源时钟计数( $f_{BGS}$ )。

**(4) 11-位预分频器**

此预分频器分割  $f_w$  以产生  $f_w/2^4$  到  $f_w/2^{11}$  时钟。

**(5) 5-位计数器**

此计数器计数  $f_w$  或  $f_w/2^9$ , 且以  $2^4/f_w$ ,  $2^5/f_w$ ,  $2^{12}/f_w$  或  $2^{14}/f_w$  间隔产生钟表定时器中断请求信号。

**(6) 选择器**

钟表定时器具有以下五个选择器。

- 选择  $f_x$ ,  $f_x/2$ ,  $f_x/4$  或  $f_x/8$  之一作为钟表定时器的源时钟的选择器
- 选择主时钟( $f_x$ )或副时钟( $f_{XT}$ )作为钟表定时器时钟的选择器
- 选择  $f_w$  或  $f_w/2^9$  作为计数 5-位计数器时钟频率的选择器
- 选择  $2^4/f_w$ ,  $2^{13}/f_w$ ,  $2^5/f_w$  或  $2^{14}/f_w$  作为 INTWT 信号产生时间间隔的选择器
- 选择  $2^4/f_w$  到  $2^{11}/f_w$  作为计时器中断请求信号(INTWTI)产生时间间隔的选择器

**(7) PRSCM 寄存器**

这是设置间隔时间的 8-位比较寄存器。

**(8) PRSM 寄存器**

此寄存器控制钟表定时器的时钟供应。

**(9) WTM 寄存器**

这是钟表定时器/计时器控制操作的 8-位寄存器, 且设置中断请求信号产生间隔。

9.3 寄存器

以下寄存器提供用于钟表定时器。

- 预分频器模式寄存器 0(PRSM0)
- 预分频器比较寄存器 0(PRSCM0)
- 钟表定时器操作模式寄存器(WTM)

(1) 预分频器模式寄存器 0(PRSM0)

PRSM0 寄存器控制钟表定时器计数时钟产生。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

复位后：00H

R/W

地址：FFFF8B0H

PRSM0

7	6	5	4	3	2	1	0
0	0	0	BGCE0	0	0	BGCS01	BGCS00

BGCE0	主时钟操作启用
0	禁用
1	启用

BGCS01	BGCS00	选择钟表定时器源时钟 (f <sub>BGCS</sub> )		
			5 MHz	4 MHz
0	0	f <sub>x</sub>	200 ns	250 ns
0	1	f <sub>x</sub> /2	400 ns	500 ns
1	0	f <sub>x</sub> /4	800 ns	1 μs
1	1	f <sub>x</sub> /8	1.6 μs	2 μs

注意事项

1.

钟表定时器操作期间禁止改变 BGCS00 和 BGCS01 位的值。

2.

BGCE0 位设为 1 之前设置 PRSM0 寄存器。

3.

根据使用的主时钟频率设置 PRSM0 和 PRSCM0 寄存器以便获得 32.768 kHz f<sub>BRG</sub> 频率。

(2) 预分频器比较寄存器 0(PRSCM0)

PRSCM0 寄存器为 8-位比较寄存器。  
此寄存器可读取或写入 8-位单元。  
此寄存器重设为 00H。

复位后： 00H      R/W      地址： FFFFF8B1H

	7	6	5	4	3	2	1	0
PRSCM0	PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00

注意事项

1. 钟表定时器操作期间禁止重写 PRSCM0 寄存器。
2. PRSM0.BGCE0 位设为 1 之前设置 PRSCM0 寄存器。
3. 根据使用的主时钟频率设置 PRSM0 和 PRSCM0 寄存器以便获得 32.768 kHz fBRG 频率。

fBRG 计算如下所示。

$f_{BRG} = f_{BGCS}/2N$

备注      fBGCS: 通过 PRSM0 寄存器设置的钟表定时器源时钟  
N:      PRSCM0 寄存器的设置值= 1 到 256  
        然而, 仅当 PRSCM0 寄存器设为 00H 时 N = 256。

**(3) 钟表定时器操作模式寄存器(WTM)**

WTM 寄存器启用或禁用计数时钟和钟表定时器操作时，设置预分频器的间隔时间，5-位计数器的控制操作，且设置钟表标记的设置时间。

设置 WTM 寄存器之前设置 PRSM0 寄存器。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

(1/2)

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	预分频器的内部时间选择
0	0	0	0	$2^4/f_w$ (488 $\mu$ s: $f_w = f_{XT}$ )
0	0	0	1	$2^5/f_w$ (977 $\mu$ s: $f_w = f_{XT}$ )
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 $\mu$ s: $f_w = f_{BAG}$ )
1	0	0	1	$2^5/f_w$ (977 $\mu$ s: $f_w = f_{BAG}$ )
1	0	1	0	$2^6/f_w$ (1.95 ms: $f_w = f_{BAG}$ )
1	0	1	1	$2^7/f_w$ (3.90 ms: $f_w = f_{BAG}$ )
1	1	0	0	$2^8/f_w$ (7.81 ms: $f_w = f_{BAG}$ )
1	1	0	1	$2^9/f_w$ (15.6 ms: $f_w = f_{BAG}$ )
1	1	1	0	$2^{10}/f_w$ (31.2 ms: $f_w = f_{BAG}$ )
1	1	1	1	$2^{11}/f_w$ (62.5 ms: $f_w = f_{BAG}$ )

WTM7	WTM3	WTM2	看门狗的设置时间选择
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 $\mu$ s: $f_w = f_{XT}$ )
0	1	1	$2^4/f_w$ (488 $\mu$ s: $f_w = f_{XT}$ )
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 $\mu$ s: $f_w = f_{BRG}$ )
1	1	1	$2^4/f_w$ (488 $\mu$ s: $f_w = f_{BRG}$ )

WTM1	5位计数器操作控制
0	操作停止之后清零
1	开始

WTM0	钟表定时器操作启用
0	停止操作（预分频器和5位计数器清零）
1	启用操作

**注意事项** 当 WTM0 和 WTM1 位均为 0 时重写 WTM2 到 WTM7 位。

- 备注**
1.  $f_w$ : 钟表定时器时钟频率
  2. 圆括号中的值应用于  $f_w = 32.768$  kHz 的操作
  3.  $f_{XT}$ : 副时钟频率
  4.  $f_{BRG}$ : 钟表定时器计数时钟频率

## 9.4 操作

### 9.4.1 作为钟表定时器操作

钟表定时器以固定时间间隔产生中断请求信号(INTWT)。钟表定时器操作以副时钟(32.768 kHz)或主时钟使用 0.25 或 0.5 秒时间间隔。

当 WTM.WTM1 和 WTM.WTM0 位设为 11 时计数操作开始。当 WTM0 位清零时, 11-位预分频器和 5-位计数器清零且计数操作停止。

钟表定时器的时间可通过 WTM1 位清零且接着 5-位计数器当同时作为计时器操作时调整。此时, 钟表定时器可能出现上至 15.6 ms 的误差, 而不影响计时器。

如果主时钟用作钟表定时器的计数时钟, 则设置使用 PRSM0.BGCS01 和 BGCS00 位的计数时钟, 使用 PRSCM0 寄存器的 8-位比较值, 且钟表定时器的计数时钟频率( $f_{BRG}$ )为 32.768 kHz。

当 PRSM0.BGCE0 位设置(1)时,  $f_{BRG}$  供给钟表定时器。

$f_{BRG}$  可通过以下表达式计算。

$$f_{BRG} = f_x / (2^{m+1} \times N)$$

要将  $f_{BRG}$  设为 32.768 kHz, 请执行以下计算并设置 BGCS01 和 BGCS00 位和 PRSCM0 寄存器。

<1> 设置  $N = f_x / 65,536$ 。设置  $m = 0$ 。

<2> 当四舍五入所得的值  $N$  的第一小数位为偶数时,  $N/2$  四舍五入且  $m$  作为  $m + 1$  之前设置  $N$ 。

<3> 重复 <2>直到  $N$  为奇数或  $m = 3$ 。

<4> 设置四舍五入所得的值  $N$  的第一小数位为 PRSCM0 寄存器且  $m$  为 BGCS01 和 BGCS00 位。

实例: 当  $f_x = 4.00 \text{ MHz}$

<1>  $N = 4,000,000 / 65,536 = 61.03\dots$ ,  $m = 0$

<2>, <3> 因为  $N$ (四舍五入第一小数位)为奇数,  $N = 61$ ,  $m = 0$ 。

<4> PRSCM0 寄存器的设置值: 3DH(61), BGCS01 和 BGCS00 位的设置值: 00

此时, 实际  $f_{BRG}$  频率如下所示。

$$\begin{aligned} f_{BRG} &= f_x / (2^{m+1} \times N) = 4,000,000 / (2 \times 61) \\ &= 32.787 \text{ kHz} \end{aligned}$$

备注 m: 除法值(BGCS01 和 BGCS00 位的设置值)= 0 到 3

N: PRSCM0 寄存器的设置值= 1 到 256

然而, 仅当 PRSCM0 寄存器设为 00H 时  $N = 256$ 。

$f_x$ : 主时钟振荡频率

## 9.4.2 作为计时器的操作

钟表定时器还可用作计时器，其以按预先设置计数值指定的间隔重复产生中断请求信号(INTWTI)。  
间隔时间可通过 WTM 寄存器的 WTM4 到 WTM7 位选择。

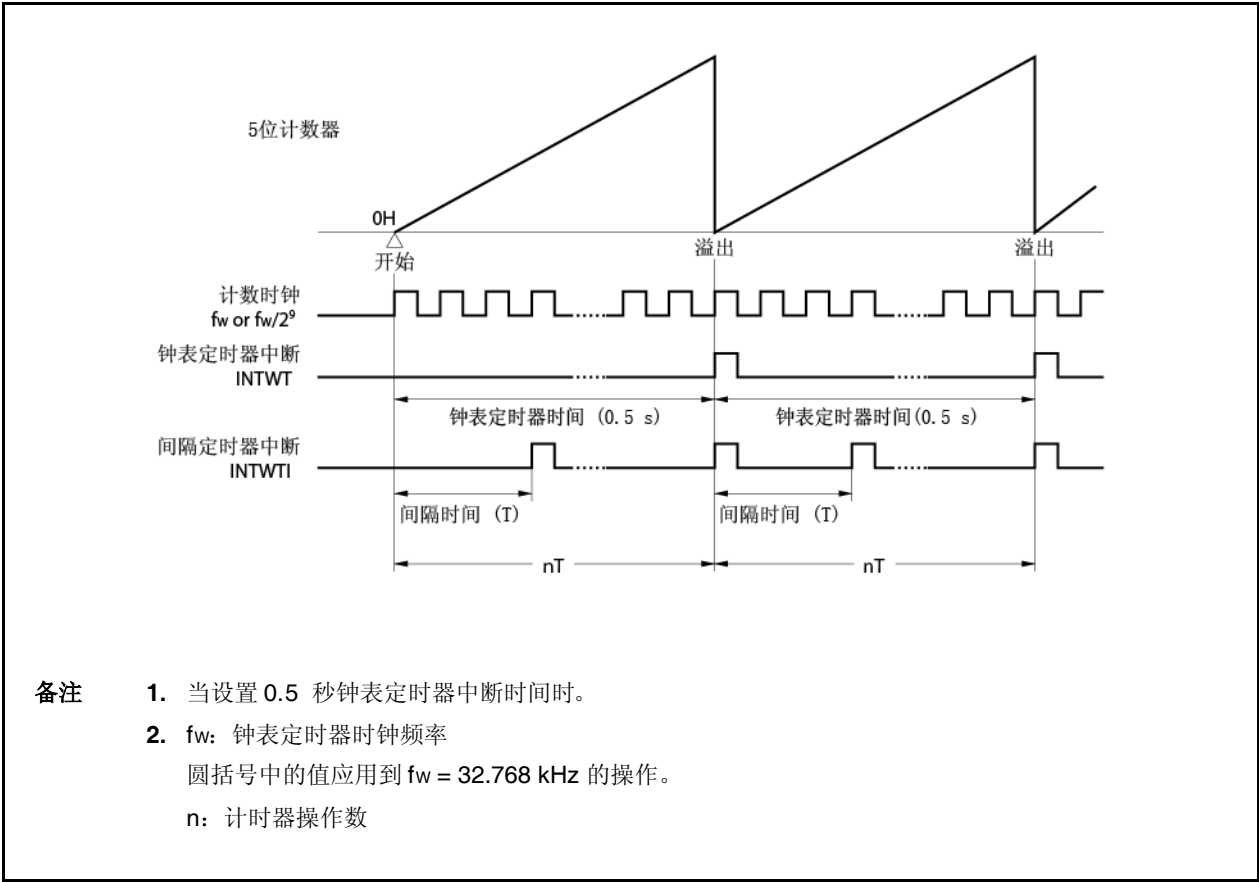
表 9-1. 计时器间隔时间

WTM7	WTM6	WTM5	WTM4	间隔时间	
0	0	0	0	$2^4 \times 1/f_w$	488 $\mu s$ (以 $f_w = f_{XT} = 32.768$ kHz 操作)
0	0	0	1	$2^5 \times 1/f_w$	977 $\mu s$ (以 $f_w = f_{XT} = 32.768$ kHz 操作)
0	0	1	0	$2^6 \times 1/f_w$	1.95 ms(以 $f_w = f_{XT} = 32.768$ kHz 操作)
0	0	1	1	$2^7 \times 1/f_w$	3.91 ms(以 $f_w = f_{XT} = 32.768$ kHz 操作)
0	1	0	0	$2^8 \times 1/f_w$	7.81 ms(以 $f_w = f_{XT} = 32.768$ kHz 操作)
0	1	0	1	$2^9 \times 1/f_w$	15.6 ms(以 $f_w = f_{XT} = 32.768$ kHz 操作)
0	1	1	0	$2^{10} \times 1/f_w$	31.3 ms(以 $f_w = f_{XT} = 32.768$ kHz 操作)
0	1	1	1	$2^{11} \times 1/f_w$	62.5 ms(以 $f_w = f_{XT} = 32.768$ kHz 操作)
1	0	0	0	$2^4 \times 1/f_w$	488 $\mu s$ (以 $f_w = f_{BRG} = 32.768$ kHz 操作)
1	0	0	1	$2^5 \times 1/f_w$	977 $\mu s$ (以 $f_w = f_{BRG} = 32.768$ kHz 操作)
1	0	1	0	$2^6 \times 1/f_w$	1.95 ms(以 $f_w = f_{BRG} = 32.768$ kHz 操作)
1	0	1	1	$2^7 \times 1/f_w$	3.91 ms(以 $f_w = f_{BRG} = 32.768$ kHz 操作)
1	1	0	0	$2^8 \times 1/f_w$	7.81 ms(以 $f_w = f_{BRG} = 32.768$ kHz 操作)
1	1	0	1	$2^9 \times 1/f_w$	15.6 ms(以 $f_w = f_{BRG} = 32.768$ kHz 操作)
1	1	1	0	$2^{10} \times 1/f_w$	31.3 ms(以 $f_w = f_{BRG} = 32.768$ kHz 操作)
1	1	1	1	$2^{11} \times 1/f_w$	62.5 ms(以 $f_w = f_{BRG} = 32.768$ kHz 操作)

备注       $f_w$ : 钟表定时器时钟频率



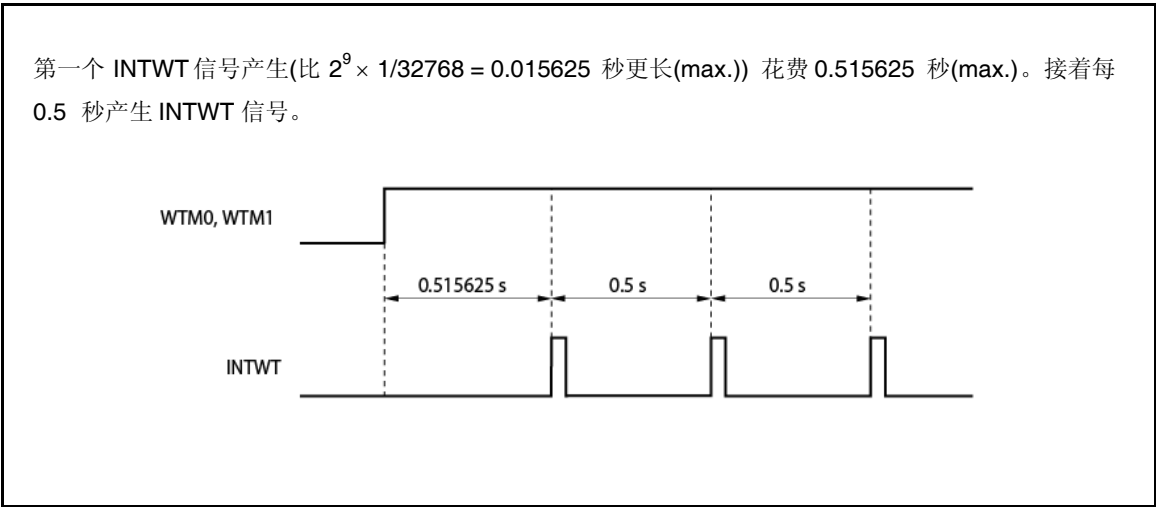
图 9-2. 钟表定时器/计时器操作时序



9.4.3 注意事项

操作启用之后第一个钟表定时器中断请求信号 (INTWT) 产生之前需要一些时间 (WTM.WTM1 和 WTM.WTM0 位 = 1)。

图 9-3. 钟表定时器中断请求信号 (INTWT) 产生实例  
(当中断周期 = 0.5 s)



## 第十章 看门狗定时器 2 的功能

### 10.1 功能

看门狗定时器 2 具有以下功能。

- 默认-开始 看门狗定时器<sup>注 1</sup>
  - 复位模式：当看门狗定时器 2 溢出(产生 WDT2RES 信号)时复位操作
  - 不可屏蔽中断请求模式：当看门狗定时器 2 溢出(产生 of INTWDT2 信号)<sup>注 2</sup>时 NMI 操作
- 可选择从主时钟和为源时钟的内部振荡时钟作输入

**注 1.** 看门狗定时器 2 在复位模式下自动开始以下复位释放。

当不使用看门狗定时器 2 时，进行复位之前经此功能或看门狗定时器 2 清零一次停止其操作且在下一间隔时间内停止。

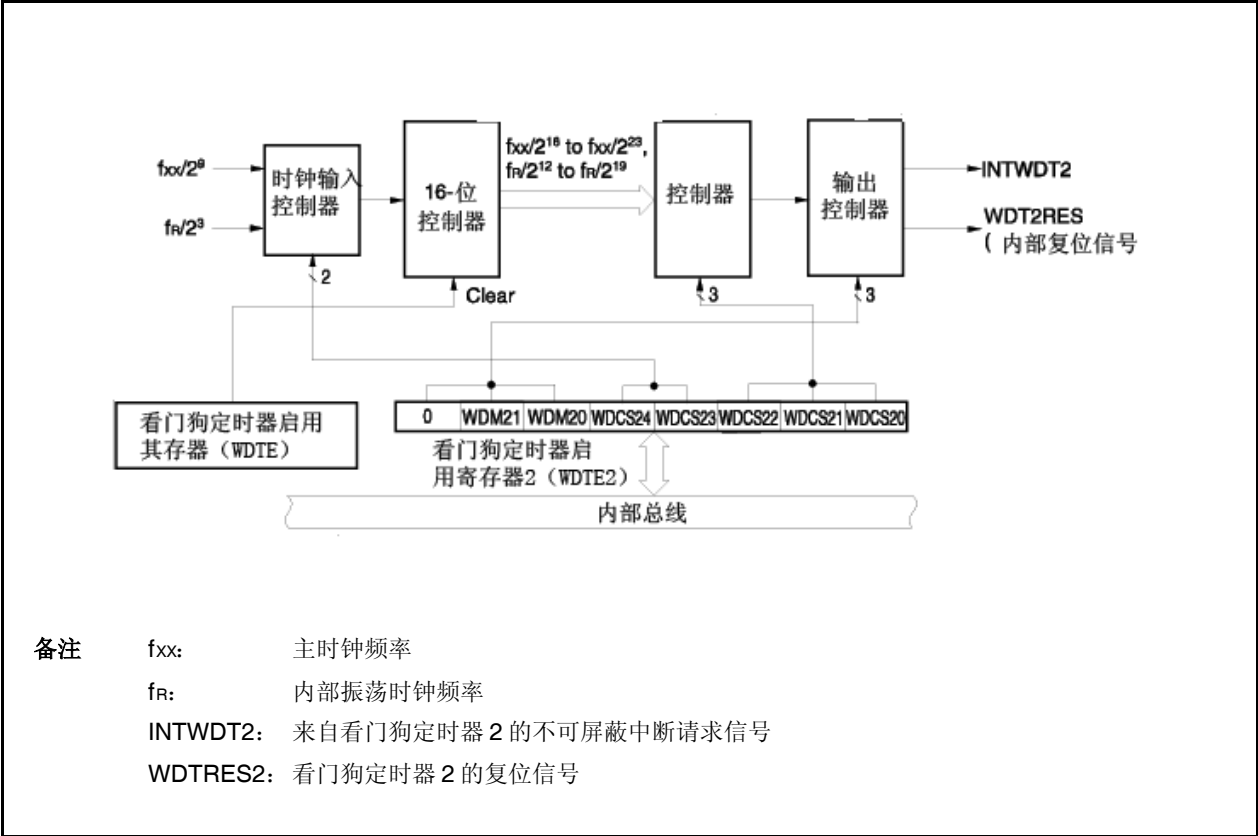
同样，即使默认设置(复位模式，间隔时间： $f_{R/2}^{19}$ )无需改变，出于验证目的仅写入 WDTM2 寄存器一次。

2. 对于由于不可屏蔽中断请求信号(INTWDT2)而产生的不可屏蔽中断服务，请参阅 14.2.2(2) 来自 INTWDT2 信号。

10.2 配置

以下给出了看门狗定时器 2 的方块图。

图 10-1. 看门狗定时器 2 的方块图



看门狗定时器 2 包括以下硬件。

表 10-1. 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) 看门狗定时器启用寄存器(WDTE)

## 10.3 寄存器

## (1) 看门狗定时器模式寄存器 2(WDTM2)

WDTM2 寄存器设置看门狗定时器 2 的溢出时间和操作时钟。

此寄存器可读取或写入 8-位单元。此寄存器可读取任意次数，而其复位释放后仅可写入一次。

此寄存器重设为 67H。

**注意事项** 在以下状态禁止访问 WDTM2 寄存器。如需详细信息，请参阅 3.4.8(2)访问专用片上外围 I/O 寄存器。

- 当 CPU 按副时钟运行且主时钟振荡停止时
- 当 CPU 按内部振荡时钟运行时

复位后:		67H	R/W	地址: FFFFF6D0H						
		7	6	5	4	3	2	1	0	
WDTM2		0	WDM21	WDM20	WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	

WDM21	WDM20	看门狗定时器操作模式选择2注
0	0	停止运行
0	1	非可屏蔽中断请求模式（INTWDT2信号产生） 复位模式（WDT2RE信号产生）
1	-	

**注** 如果 OPB1 位使用选项字节功能设为 1 (请参阅 第二十三章)，复位模式固定。

**注意事项** 1. 如需 WDCS20 到 WDCS24 位的详细信息，请参阅 表 10-2 看门狗定时器 2 时钟选择。

2. 如果复位后 WDTM2 寄存器重写两次，强制产生溢出信号且计数器复位。

3. 要有意产生溢出信号，请仅写入 WDTM2 寄存器两次或将 ACH 之外的值写入 WDTE 寄存器一次。然而，当看门狗定时器 2 设为停止运行时，即使数据仅写入 WDTM2 寄存器两次或“ACH”之外的值仅写入 WDTE 寄存器一次，也不产生溢出信号。

4. 要停止看门狗定时器 2 运行，请将 RCM.R 停止位设为 1(内部振荡器停止)，且将 1FH 写入 WDTM2 寄存器。然而，如果 OPB1 位使用选项字节功能设为 1 (请参阅 第 23 章)，则看门狗定时器 2 不能通过复位以外的任何方式停止。

<R>

表 10-2. 看门狗定时器 2 时钟选择

WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	选择时钟	100 kHz(MIN.)	200 kHz(TYP.)	400 kHz(MAX.)
0	0	0	0	0	$2^{12}/f_R$	41.0 ms	20.5 ms	10.2 ms
0	0	0	0	1	$2^{13}/f_R$	81.9 ms	41.0 ms	20.5 ms
0	0	0	1	0	$2^{14}/f_R$	163.8 ms	81.9 ms	41.0 ms
0	0	0	1	1	$2^{15}/f_R$	327.7 ms	163.8 ms	81.9 ms
0	0	1	0	0	$2^{16}/f_R$	655.4 ms	327.7 ms	163.8 ms
0	0	1	0	1	$2^{17}/f_R$	1, 310.7 ms	655.4 ms	327.7 ms
0	0	1	1	0	$2^{18}/f_R$	2, 621.4 ms	1, 310.7 ms	655.4 ms
0	0	1	1	1	$2^{19}/f_R$	5, 242.9 ms	2, 621.4 ms	1, 310.7 ms
						f <sub>xx</sub> = 4 MHz		f <sub>xx</sub> = 5 MHz
0	1	0	0	0	$2^{16}/f_{xx}$	16.4 ms	13.1 ms	
0	1	0	0	1	$2^{17}/f_{xx}$	32.8 ms	26.2 ms	
0	1	0	1	0	$2^{18}/f_{xx}$	65.5 ms	52.4 ms	
0	1	0	1	1	$2^{19}/f_{xx}$	131.1 ms	104.9 ms	
0	1	1	0	0	$2^{20}/f_{xx}$	262.1 ms	209.7 ms	
0	1	1	0	1	$2^{21}/f_{xx}$	524.3 ms	419.4 ms	
0	1	1	1	0	$2^{22}/f_{xx}$	1, 048.6 ms	838.9 ms	
0	1	1	1	1	$2^{23}/f_{xx}$	2, 097.2 ms	1, 677.7 ms	
1	1	1	1	1	运行停止			

**注意事项** 如果 **OPB1** 位使用选项字节功能设为 1，则时钟固定为内部振荡时钟( $f_R$ ) (可选择  $2^{12}/f_R$  到  $2^{19}/f_R$ )。如需详细信息，请参阅 第二十三章 选项字节功能。

(2) 看门狗定时器启用寄存器(WDTE)

看门狗定时器 2 的计数器清零且通过将“ACH”写入 WDTE 寄存器重新开始计数。

WDTE 寄存器可读取或写入 8-位单元。

此寄存器重设为 9AH。



- 注意事项
1. 当“ACH”以外的值写入 WDTE 寄存器时，强制输出溢出信号。
  2. 当执行 WDTE 寄存器的 1-位内存操作指令时，强制输出溢出信号。
  3. 要有意产生溢出信号，请仅写入 WDTM2 寄存器两次或将 ACH 之外的值写入 WDTE 寄存器一次。  
然而，当看门狗定时器 2 设为停止运行时，即使数据仅写入 WDTM2 寄存器两次或“ACH”之外的值仅写入 WDTE 寄存器一次，也不产生溢出信号。
  4. WDTE 寄存器的读取值为“9AH”(其不同与写入值“ACH”)。

<R>

## 10.4 操作

复位释放后看门狗定时器 2 在复位模式下自动开始。

WDTM2 寄存器仅可复位后使用字节访问写入一次。要使用看门狗定时器 2，请使用 8-位内存操作指令将操作模式和间隔时间写入 WDTM2 寄存器。此后，不能停止看门狗定时器 2 运行。

WDTM2 寄存器的 WDCS24 到 WDCS20 位用于选择看门狗定时器 2 循环检测时间间隔。

将 ACH 写入 WDTM2 寄存器清零看门狗定时器 2 的计数器并再次开始计数操作。计数操作开始之后，在循环检测时间间隔内将 ACH 写入 WDTM2。

如果时间间隔到期而 ACH 未写入 WDTM2 寄存器，则根据 WDM21 和 WDTM2.WDM20 位的设置值产生复位信号(WDT2RES)或不可屏蔽中断请求信号(INTWDT2)。

当 WDTM2.WDM21 位设为 1(复位模式)时，如果释放复位或备用后振荡稳定期间 WDT 发生溢出，将不出现内部复位且 CPU 时钟将切换到内部振荡时钟。

要不使用看门狗定时器 2，请将 1FH 写入 WDTM2 寄存器。

如需当设置不可屏蔽中断请求模式时的不可屏蔽中断服务的有关信息，请参阅 14.2.2(2) 来自 INTWDT2 信号。

## 第十一章 A/D 转换器

### 11.1 概述

A/D 转换器将模拟输入信号转换为数字值，具有 10 位分辨率，且可操作 12 模拟输入信号通道(ANIO 到 ANI11)。

A/D 转换器具有以下特性。

- 10-位分辨率
- 12 通道
- 逐次逼近法
- 工作电压： $AV_{REF0} = 4.0$  到  $5.5\text{ V}$
- 模拟输入电压：0 V 到  $AV_{REF0}$
- 按操作模式提供以下功能。
  - 连续选择模式
  - 连续扫描模式
  - 单触发扫描模式
- 按触发模式提供以下功能。
  - 软件触发模式
  - 外部触发模式(外部，1)
  - 定时器触发模式
- 掉电监控功能(转换结果比较功能)

### 11.2 功能

#### (1) 10-位分辨率 A/D 转换

从 ANIO 到 ANI11 选择模拟输入通道，且以 10 位分辨率重复 A/D 转换操作。每当 A/D 转换完成时，产生中断请求信号(INTAD)。

#### (2) 掉电检测功能

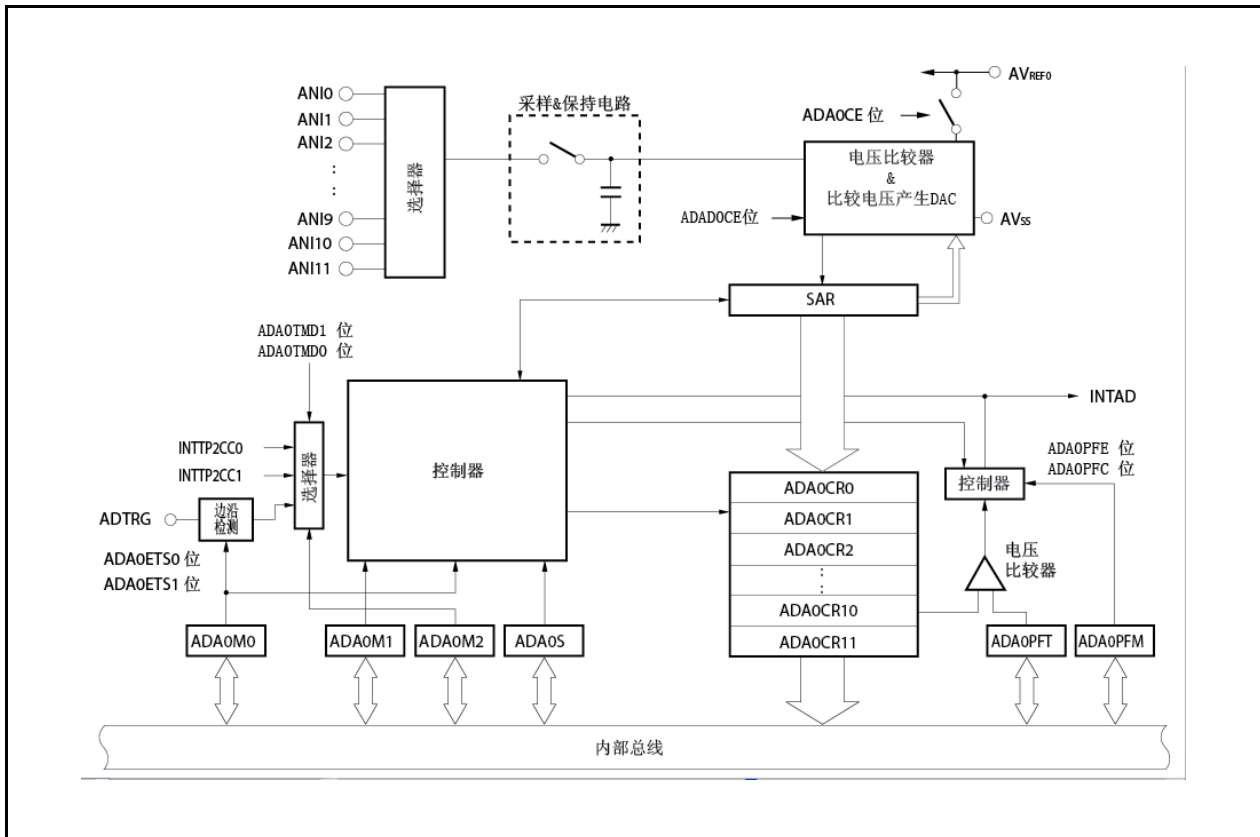
此功能用于检测电池电压降落。将 A/D 转换结果(ADA0CRnH 寄存器的值)与 ADA0PFT 寄存器的值比较，且仅当满足指定比较条件 ( $n = 0$  到  $11$ )时产生 INTAD 信号。



### 11.3 配置

A/D 转换器的框图如下所示。

图 11-1. A/D 转换器的框图



A/D 转换器包括以下硬件。

表 11-1. A/D 转换器的配置

项目	配置
模拟输入	12 通道(ANI0 到 ANI11 引脚)
寄存器	逐次逼近寄存器(SAR) A/D 转换结果寄存器 0 到 11(ADA0CR0 到 ADA0CR11) A/D 转换结果寄存器 0H 到 11H(ADCR0H 到 ADCR11H): 仅可读取高 8 位
控制寄存器	A/D 转换器模式寄存器 0 到 2(ADA0M0 到 ADA0M2) A/D 转换器通道规范寄存器 0(ADA0S) 掉电比较模式寄存器(ADA0PFM) 掉电比较阈值寄存器(ADA0PFT)

**(1) 逐次逼近寄存器(SAR)**

SAR 寄存器将模拟输入信号的电压值与比较电压产生 DAC 的输出电压 (比较电压) 比较, 且保持从最高位(MSB) 开始的比较结果。

当比较结果保持到最低位(LSB)(亦即, 当 A/D 转换完成), SAR 寄存器的内容转移到 ADA0CRn 寄存器。

备注 n = 0 到 11

**(2) A/D 转换结果寄存器 n(ADA0CRn), A/D 转换结果寄存器 nH(ADA0CRnH)**

ADA0CRn 寄存器为存储 A/D 转换结果的 16-位寄存器。ADA0ARn 包含 12 寄存器且 A/D 转换结果存储在对应于模拟输入的 AD0CRn 寄存器的 10 高位。(低 6 位固定为 0。)

**(3) A/D 转换器模式寄存器 0(ADA0M0)**

此寄存器指定操作模式并通过 A/D 转换器控制转换操作。

**(4) A/D 转换器模式寄存器 1(ADA0M1)**

此寄存器设置要转换的模拟输入信号的转换时间。

**(5) A/D 转换器模式寄存器 2(ADA0M2)**

此寄存器设置硬件触发模式。

**(6) A/D 转换器通道规范寄存器(ADA0S)**

此寄存器设置输入要转换的模拟电压的输入端口。

**(7) 掉电比较模式寄存器(ADA0PFM)**

此寄存器设置掉电监控模式。

**(8) 掉电比较阈值寄存器(ADA0PFT)**

ADA0PFT 寄存器设置与 A/D 转换结果寄存器 nH(ADA0CRnH) 的值比较的阈值。将设为 ADA0PFT 寄存器的 8-位数据与 A/D 转换结果寄存器(ADA0CRnH)的高 8 位比较。

**(9) 控制器**

当 A/D 转换完成时或当使用掉电检测功能时控制器将 A/D 转换结果(ADA0CRnH 寄存器的值)与 ADA0PFT 寄存器的值比较, 且仅当满足指定比较条件时产生 INTAD 信号。

**(10) 采样&保持电路**

采样&保持电路采样由输入电路选择的每一模拟输入信号并将采样数据发送到电压比较器。A/D 转换期间此电路同样保持采样模拟输入信号电压。

**(11) 电压比较器**

电压比较器将采样和保持的电压值与比较电压产生 DAC 的电压值比较。

**(12) 比较电压产生 DAC**

此比较电压产生 DAC 连接在  $AV_{REF0}$  与  $AV_{SS}$  之间并产生与模拟输入信号比较的电压。

**(13) ANI0 到 ANI11 引脚**

这些是 12 A/D 转换器通道的模拟输入引脚且用于将模拟信号输入转换成数字信号。由 ADA0S 寄存器选作模拟输入的一引脚之外的引脚可用作输入端口引脚。

- 注意事项**
1. 确保输入到 ANI0 到 ANI11 引脚的电压不超过额定值。尤其如果通道输入  $V_{REF0}$  电压或更高电压，则未定义通道转换值，且还可能影响其他通道的转换值。
  2. 模拟输入引脚(ANI0 到 ANI11)另外用作输入端口引脚(P70 到 P711)。如果选择 ANI0 到 ANI11 的任一者执行 A/D 转换，则转换期间禁止执行到端口 7 的输入指令。如果执行，则转换分辨率可能降级。

**(14)  $AV_{REF0}$  引脚**

这是引脚用于输入 A/D 转换器的参考电压。即使当未使用 A/D 转换器时始终使此引脚的电压始与  $V_{DD}$  引脚相同。输入到 ANI0 到 ANI11 引脚的信号基于施加在  $AV_{REF0}$  与  $AV_{SS}$  引脚之间的电压转换位数字信号。

**(15)  $AV_{SS}$  引脚**

这是 A/D 转换器的接地引脚。即使当未使用 A/D 转换器时始终使此引脚的电压始与  $V_{SS}$  引脚相同。

## 11.4 寄存器

A/D 转换器由以下寄存器控制。

- A/D 转换器模式寄存器 0, 1, 2(ADA0M0, ADA0M1, ADA0M2)
- A/D 转换器通道寄存器规范 0(ADA0S)
- 掉电比较模式寄存器(ADA0PFM)

还使用以下寄存器。

- A/D 转换结果寄存器 n(ADA0CRn)
- A/D 转换结果寄存器 nH(ADA0CRnH)
- 掉电比较阈值寄存器(ADA0PFT)

### (1) A/D 转换器模式寄存器 0(ADA0M0)

ADA0M0 寄存器为指定操作模式和控制转换操作的 8-位寄存器。

此寄存器可读取或写入 8-位或 1-位单元。然而，ADA0EF 位只读。

此寄存器重设为 00H。

**注意事项** 在以下状态禁止访问 ADA0M0 寄存器。如需详细信息，请参阅 3.4.8(2) 访问专用片上外围 I/O 寄存器。

- 当 CPU 按副时钟运行且主时钟振荡停止时
- 当 CPU 按内部振荡时钟运行时

复位后: 00H R/W 地址: FFFFF200H

	7	6	5	4	3	2	1	0
ADA0M0	ADA0CE	0	ADA0MD1	ADA0MD0	ADA0ETS1	ADA0ETS0	ADA0TMD	ADA0EF

ADA0CE	A/D转换控制
0	停止A/D转换
1	启用A/D转换

ADA0MD1	ADA0MD0	A/D转换器操作模式规格
0	0	继续选择模式
0	1	继续扫描模式
1	0	禁止的设置
1	1	单触发扫描模式

ADA0ETS1	ADA0ETS0	外部触发（ADTRG引脚）输入有效沿规格
0	0	无边沿检测
0	1	下降沿检测
1	0	上升沿检测
1	1	上升及下降沿检测

ADA0TMD	触发模式规格
0	软件触发模式
1	外部触发模式/定时器触发模式

ADA0EF	A/D转换器状态显示
0	停止的A/D转换
1	进行中的A/D转换

- 注意事项
1. 忽略写入位 0 的操作。
  2. 当 A/D 转换启用时(ADA0CE 位= 1)禁止改变 ADA0M1 寄存器的值。
  3. 如果 A/D 转换期间写入 ADA0M0, ADA0M2, ADA0S, ADA0PFM 及 ADA0PFT 寄存器(ADA0EF 位= 1), 则按照模式执行以下操作。
    - 在软件触发模式下  
A/D 转换停止且从开始处再次开始。
    - 在硬件触发模式下  
A/D 转换停止, 且设置触发备用状态。
  4. 当不使用 A/D 转换器时, 通过设置 ADA0CE 位停止操作以减少耗电。
  5. A/D 转换开始时的输入引脚数据的第一次转换的分辨率可能降级。如需详细信息, 请参阅 11.6(7) AV<sub>REF0</sub> 引脚。

### (2) A/D 转换器模式寄存器 1(ADA0M1)

ADA0M1 寄存器为控制转换时间规范的 8-位寄存器。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

复位后: 00H    R/W    地址: FFFFF201H								
	7	6	5	4	3	2	1	0
ADA0M1	ADA0HS1	0	0	0	ADA0FR3	ADA0FR2	ADA0FR1	ADA0FR0

- 注意事项
1. 确保位 6 到 4 清零。
  2. 确保 ADA0HS1 位设为“1”。

备注      有关 A/D 转换时间设置的示例, 请参阅 表 11-2.

表11-2. 转换模式设置示例

ADA0HS1	ADA0FR3 到 ADA0FR0				A/D 转换时间	f <sub>xx</sub> = 20 MHz	f <sub>xx</sub> = 16 MHz	f <sub>xx</sub> = 4 MHz	A/D 稳定时间 <sup>注</sup>
	3	2	1	0					
1	0	0	0	0	31/f <sub>xx</sub>	禁止设置	禁止设置	7.75 μs	16/f <sub>xx</sub>
	0	0	0	1	62/f <sub>xx</sub>	3.10 μs	3.88 μs	15.50 μs	31/f <sub>xx</sub>
	0	0	1	0	93/f <sub>xx</sub>	4.65 μs	5.81 μs	禁止设置	47/f <sub>xx</sub>
	0	0	1	1	124/f <sub>xx</sub>	6.20 μs	7.75 μs	禁止设置	50/f <sub>xx</sub>
	0	1	0	0	155/f <sub>xx</sub>	7.75 μs	9.69 μs	禁止设置	50/f <sub>xx</sub>
	0	1	0	1	186/f <sub>xx</sub>	9.30 μs	11.63 μs	禁止设置	50/f <sub>xx</sub>
	0	1	1	0	217/f <sub>xx</sub>	10.85 μs	13.56 μs	禁止设置	50/f <sub>xx</sub>
	0	1	1	1	248/f <sub>xx</sub>	12.40 μs	15.50 μs	禁止设置	50/f <sub>xx</sub>
	1	0	0	0	279/f <sub>xx</sub>	13.95 μs	禁止设置	禁止设置	50/f <sub>xx</sub>
	1	0	0	1	310/f <sub>xx</sub>	15.50 μs	禁止设置	禁止设置	50/f <sub>xx</sub>
	1	0	1	0	341/f <sub>xx</sub>	禁止设置	禁止设置	禁止设置	50/f <sub>xx</sub>
	1	0	1	1	372/f <sub>xx</sub>	禁止设置	禁止设置	禁止设置	50/f <sub>xx</sub>
	1	1	0	0	403/f <sub>xx</sub>	禁止设置	禁止设置	禁止设置	50/f <sub>xx</sub>
	1	1	0	1	434/f <sub>xx</sub>	禁止设置	禁止设置	禁止设置	50/f <sub>xx</sub>
	1	1	1	0	465/f <sub>xx</sub>	禁止设置	禁止设置	禁止设置	50/f <sub>xx</sub>
	1	1	1	1	496/f <sub>xx</sub>	禁止设置	禁止设置	禁止设置	50/f <sub>xx</sub>

注      当 ADA0M0 寄存器的 ADA0CE 位从 0 改为 1 以确保 A/D 转换器稳定时间, 输入以上时钟值之一以后开始第一次 A/D 转换。

- <R>      注意事项
1. 设为 3.1 μs ≤ 转换时间 ≤ 15.5 μs。
  2. 稳定时间期间禁止 ADA0M0, ADA0M2, ADA0S, ADA0PFM 以及 ADA0PFT 寄存器的重写和触发输入。
- <R>

(3) A/D 转换器模式寄存器 2(ADA0M2)

ADA0M2 寄存器指定硬件触发模式。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

复位后: 00H	R/W	地址: FFFFF203H						
ADA0M2	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	ADA0TMD1	ADA0TMD0
		ADA0TMD1	ADA0TMD0	硬件触发模式规格				
		0	0	外部触发模式 (当检测到ADTRG引脚有效沿时)				
		0	1	定时器触发模式0 (当产生INTTP2CC0中断请求时)				
		1	0	定时器触发模式1 (当产生INTTP2CC1中断请求时)				
		1	1	禁止的设置				

注意事项 确保位 7 到 2 清零。

#### (4) A/D 转换器通道规范寄存器 0(ADA0S)

ADA0S 寄存器指定输入转换成数字信号的模拟电压的引脚。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

复位后: 00H R/W 地址: FFFFF202H

	7	6	5	4	3	2	1	0
ADA0S	0	0	0	0	ADA0S3	ADA0S2	ADA0S1	ADA0S0

ADA0S3	ADA0S2	ADA0S1	ADA0S0	选择模式	扫描模式
0	0	0	0	ANI0	ANI0
0	0	0	1	ANI1	ANI0, ANI1
0	0	1	0	ANI2	ANI0 到 ANI2
0	0	1	1	ANI3	ANI0 到 ANI3
0	1	0	0	ANI4	ANI0 到 ANI4
0	1	0	1	ANI5	ANI0 到 ANI5
0	1	1	0	ANI6	ANI0 到 ANI6
0	1	1	1	ANI7	ANI0 到 ANI7
1	0	0	0	ANI8	ANI0 到 ANI8
1	0	0	1	ANI9	ANI0 到 ANI9
1	0	1	0	ANI10	ANI0 到 ANI10
1	0	1	1	ANI11	ANI0 到 ANI11
以上值除外				禁止设置	



(5) A/D 转换结果寄存器 n, nH(ADA0CRn, ADA0CRnH)

ADA0CRn 和 ADA0CRnH 寄存器存储 A/D 转换结果。

这些寄存器按 16-位或 8-位单元只读。然而, 指定 16-位访问的 ADA0CRn 寄存器和 8-位访问的 ADA0CRnH 寄存器。转换结果的 10 位从 ADA0CRn 寄存器的高 10 位读取, 且 0 从低 6 位读取。转换结果的高 8 位从 ADA0CRnH 寄存器读取。

**注意事项** 在以下状态禁止访问 ADA0CRn 和 ADA0CRnH 寄存器。如需详细信息, 请参阅 3.4.8(2) 访问专用片上外围 I/O 寄存器。

- 当 CPU 按副时钟运行且主时钟振荡停止时
- 当 CPU 按内部振荡时钟运行时

复位后: 未定义 R 地址: ADA0CR0 FFFFF210H, ADA0CR1 FFFFF212H,  
ADA0CR2 FFFFF214H, ADA0CR3 FFFFF216H,  
ADA0CR4 FFFFF218H, ADA0CR5 FFFFF21AH,  
ADA0CR6 FFFFF21CH, ADA0CR7 FFFFF21EH,  
ADA0CR8 FFFFF220H, ADA0CR9 FFFFF222H,  
ADA0CR10 FFFFF224H, ADA0CR11 FFFFF226H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADA0CRn	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

复位后: 未定义 R 地址: ADA0CR0H FFFFF211H, ADA0CR1H FFFFF213H,  
ADA0CR2H FFFFF215H, ADA0CR3H FFFFF217H,  
ADA0CR4H FFFFF219H, ADA0CR5H FFFFF21BH,  
ADA0CR6H FFFFF21DH, ADA0CR7H FFFFF21FH,  
ADA0CR8H FFFFF221H, ADA0CR9H FFFFF223H,  
ADA0CR10H FFFFF225H, ADA0CR11H FFFFF227H

	7	6	5	4	3	2	1	0
ADA0CRn H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

备注 n = 0 到 11

**注意事项** 写入 ADA0M0 和 ADA0S 寄存器的操作可能导致 ADA0CRn 寄存器的内容未定义。转换之后, 写入 ADA0M0 和 ADA0S 寄存器之前读取转换结果。如果使用以上系列之外的系列则可能不读取校正转换结果。

模拟输入引脚(ANI0 到 ANI11)的模拟电压输入与 A/D 转换结果(ADA0CRn 寄存器)之间的关系如下所示。

$$SAR = \text{INT}\left(\frac{V_{IN}}{AVREF0} \times 1,024 + 0.5\right)$$

$$ADA0CR^{\text{注}} = SAR \times 64$$

或,

$$(SAR - 0.5) \times \frac{AVREF}{1,024} \leq V_{IN} < (SAR + 0.5) \times \frac{AVREF}{1,024}$$

INT(): 返回整值的函数 ()

V<sub>IN</sub>: 模拟输入电压

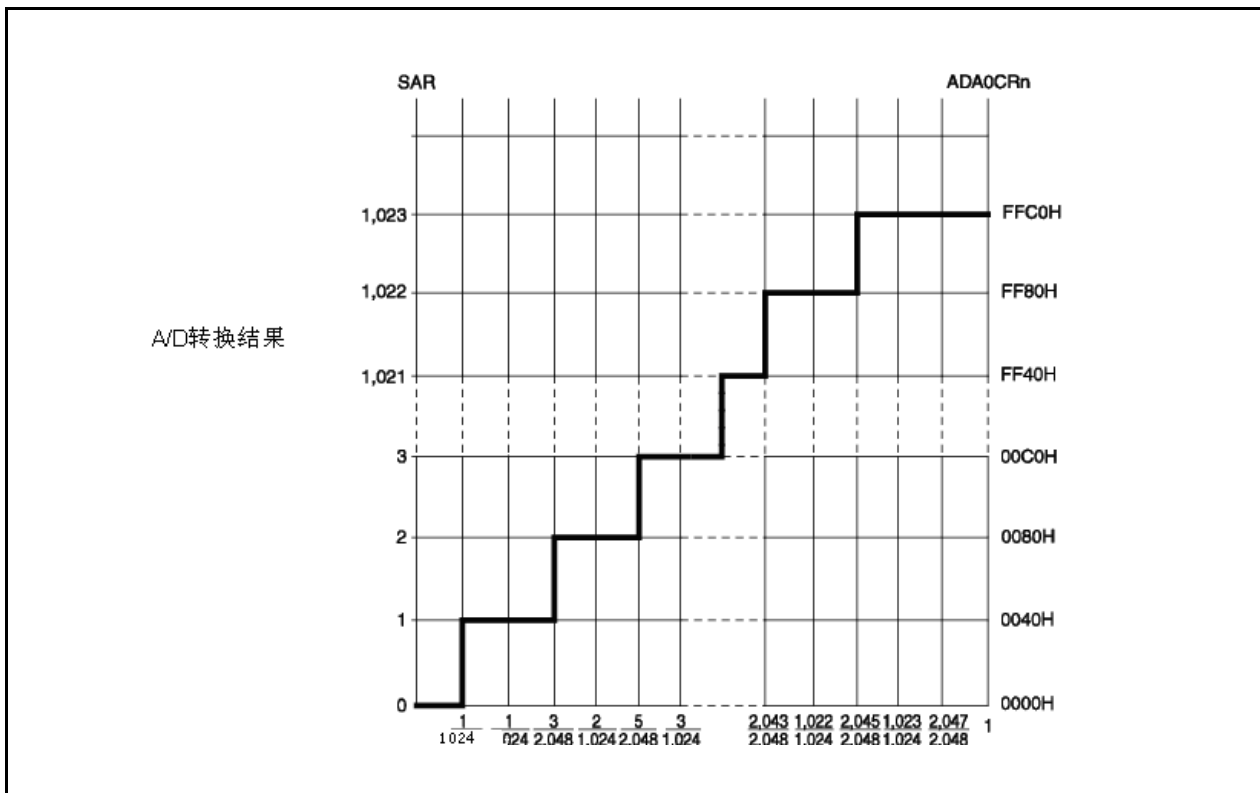
AV<sub>REF0</sub>: AV<sub>REF0</sub> 引脚电压

ADA0CR: ADA0CRn 寄存器的值

注 ADA0CRn 寄存器的低 6 位固定为 0。

以下给出了模拟输入电压与 A/D 转换结果之间的关系。

图 11-2. 模拟输入电压与 A/D 转换结果的关系



(6) 掉电比较模式寄存器(ADA0PFM)

ADA0PFM 寄存器为设置掉电比较模式的 8-位寄存器。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

复位后: 00H		R/W	地址: FFFF204H							
			7	6	5	4	3	2	1	0
ADA0PFM			ADA0PFE	ADA0PFC	0	0	0	0	0	0
ADA0PFE		掉电比较启用/禁用选择								
0		禁用的掉电比较								
1		启用的掉电比较								
ADA0PFC		掉电比较模式选择								
0		当ADA0CRnH≥ADA0PFT时产生中断请求信号(INTAD)								
1		当ADA0CRnH<ADA0PFT时产生中断请求信号(INTAD)								

- 注意事项
1. 在选择模式下，将设为 ADA0PFT 寄存器的 8-位数据与由 ADA0S 寄存器指定的 ADA0CRnH 寄存器的值比较。如果结果匹配 ADA0PFC 位指定的条件，则转换结果存储在 ADA0CRn 寄存器且 INTAD 信号产生。然而，如果不匹配，则不产生中断信号。
  2. 在扫描模式下，将设为 ADA0PFT 寄存器的 8-位数据与 ADA0CR0H 寄存器的内容比较。如果结果匹配由 ADA0PFC 位指定的条件，则转换结果存储在 ADA0CR0 寄存器且产生 INTAD 信号。然而，如果不匹配，则不产生 INTAD 信号。而不考虑比较结果，扫描操作继续进行且转换结果存储在 ADA0CRn 寄存器直到扫描操作完成。然而，扫描操作完成之后不产生 INTAD 信号。

(7) 掉电比较阈值寄存器(ADA0PFT)

<R>

ADA0PFT 寄存器当与 A/D 转换结果寄存器 nH(ADA0CRnH)比较时设置阈值。

将 ADA0PFT 寄存器中的 8-位数据设置与 ADA0CRnH 寄存器的值比较。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 00H。

复位后: 00H		R/W	地址: FFFFF205H																	
			7	6	5	4	3	2	1	0										
ADA0PFT	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																			

## 11.5 操作

### 11.5.1 基本操作

- <1> 通过使用 ADA0M0, ADA0M1, ADA0M2 及 ADA0S 寄存器设置操作模式, 触发模式及执行 A/D 转换的转换时间。当设置 ADA0M0 寄存器的 ADA0CE 位时, 在软件触发模式下转换开始且在外  
部或定时器触发模式 A/D 转换器等待触发。
- <2> 当 A/D 转换开始时, 输入到选择模拟输入通道的电压通过采样&保持电路采样。
- <3> 当采样&保持电路在特定时间采样输入通道, 其输入保持状态, 且保持输入模拟电压直到 A/D 转  
换完成。
- <4> 设置逐次逼近寄存器(SAR)的位 9 以将比较电压产生 DAC 设为  $(1/2)AV_{REF0}$ 。
- <5> 比较电压产生 DAC 与模拟输入电压之间的电压差值通过电压比较器比较。如果模拟输入电压高于  
 $(1/2)AV_{REF0}$ , 则 SAR 寄存器的 MSB 保持设置。如果其低于  $(1/2)AV_{REF0}$ , 则 MSB 复位。
- <6> 接着, 自动设置 SAR 寄存器的位 8 并开始下一次比较。根据位 9 的值, 已设置其结果, 比较电  
压产生 DAC 的选择如下所示。
  - 位 9 = 1:  $(3/4)AV_{REF0}$
  - 位 9 = 0:  $(1/4)AV_{REF0}$比较此比较电压和模拟输入电压且, 根据结果, 位 8 的操作如下所示。  
模拟输入电压  $\geq$  比较电压: 位 8 = 1  
模拟输入电压  $\leq$  比较电压: 位 8 = 0
- <7> 继续对 SAR 寄存器的位 0 进行此比较。
- <8> 当完成 10 位比较时, 有效数字结果存储在 SAR 寄存器中, 其接着转移并存储在 ADA0CRn 寄存  
器中。其后, 产生 A/D 转换结束中断请求信号(INTAD)。

### 11.5.2 触发模式

开始转换操作的时序通过设置触发模式指定。触发模式包括软件触发模式和硬件触发模式。硬件触发模式包括定时器触发模式 0 和 1 以及外部触发模式。使用 ADA0M0.ADA0TMD 位设置触发模式。硬件触发模式通过 ADA0M2.ADA0TMD1 和 ADA0M2.ADA0TMD0 位设置。

#### (1) 软件触发模式

当 ADA0M0.ADA0CE 位设为 1 时，转换由 ADA0S 寄存器指定的模拟输入引脚(ANI0 到 ANI11)的信号。当转换完成时，结果存储在 ADA0CRn 寄存器。同时，产生 A/D 转换结束中断请求信号(INTAD)。如果通过 ADA0M0.ADA0MD1 和 ADA0M0.ADA0MD0 位指定的操作模式为连续选择/扫描模式，则开始下一次转换，除非第一次转换完成之后 ADA0CE 位清零。如果操作模式为单触发选择/扫描模式则转换执行一次并结束。

当转换开始时，ADA0M0.ADA0EF 位设为 1(指示转换进行中)。

如果转换期间写入 ADA0M0，ADA0M2，ADA0S，ADA0PFM 或 ADA0PFT 寄存器，则转换异常中止且从开始处再次开始。

#### (2) 外部触发模式

在此模式下，当外部触发输入(到 ADTRG 引脚)时开始转换由 ADA0S 寄存器指定的模拟输入引脚(ANI0 到 ANI11)的信号。检测到的外部触发边沿(亦即，上升沿，下降沿或上升沿和下降沿)可使用 ADA0M0.ADA0ETS1 和 ADA0M0.ATA0ETS0 位指定。当 ADA0CE 位设为 1 时，A/D 转换器等待触发，且外部触发输入之后开始转换。

当转换完成时，转换结果存储在 ADA0CRn 寄存器中，而不考虑连续选择，连续扫描或单触发扫描模式是否通过 ADA0MD1 和 ADA0MD0 位设为操作模式。同时，产生 INTAD 信号，且 A/D 转换器等待再次触发。

当转换开始时，ADA0EF 位设为 1(指示转换进行中)。然而，当 A/D 转换器等待触发时，ADA0EF 位清零(指示转换停止)。如果转换操作期间输入有效触发，则转换异常中止且从开始处再次开始。如果转换操作期间写入 ADA0M0，ADA0M2，ADA0S，ADA0PFM 或 ADA0PFT 寄存器，转换不异常中止，且 A/D 转换器等待再次触发。

### (3) 定时器触发模式

在此模式下，通过连接到定时器的捕获/比较寄存器的比较匹配中断请求信号(INTTP2CC0 或 INTTP2CC1)开始转换由 ADA0S 寄存器指定的模拟输入引脚(ANI0 到 ANI11)的信号。INTTP2CC0 或 INTTP2CC1 信号通过 ADA0TMD1 和 ADA0TMD0 位选择，且转换在指定比较匹配中断请求信号的上升沿开始。当 ADA0CE 位设为 1 时，A/D 转换器等待触发，且当输入定时器比较匹配中断请求信号时开始转换。

当转换完成时，不考虑连续选择，连续扫描或单触发扫描模式是否通过 ADA0MD1 和 ADA0MD0 位设为操作模式，转换结果存储在 ADA0CRn 寄存器。同时，产生 INTAD 信号，且 A/D 转换器等待再次触发。

当转换开始时，ADA0EF 位设为 1(指示转换进行中)。然而，当 A/D 转换器等待触发时，ADA0EF 位清零(指示转换停止)。如果转换操作期间输入有效触发，则转换异常中止且从开始处再次开始。

如果转换操作期间写入 ADA0M0, ADA0M2, ADA0S, ADA0PFM 或 ADA0PFT 寄存器，则转换停止，且 A/D 转换器等待再次触发。

### 11.5.3 操作模式

三种操作模式可用作设置 ANI0 到 ANI11 引脚的模式：连续选择模式，连续扫描模式以及单触发扫描模式。

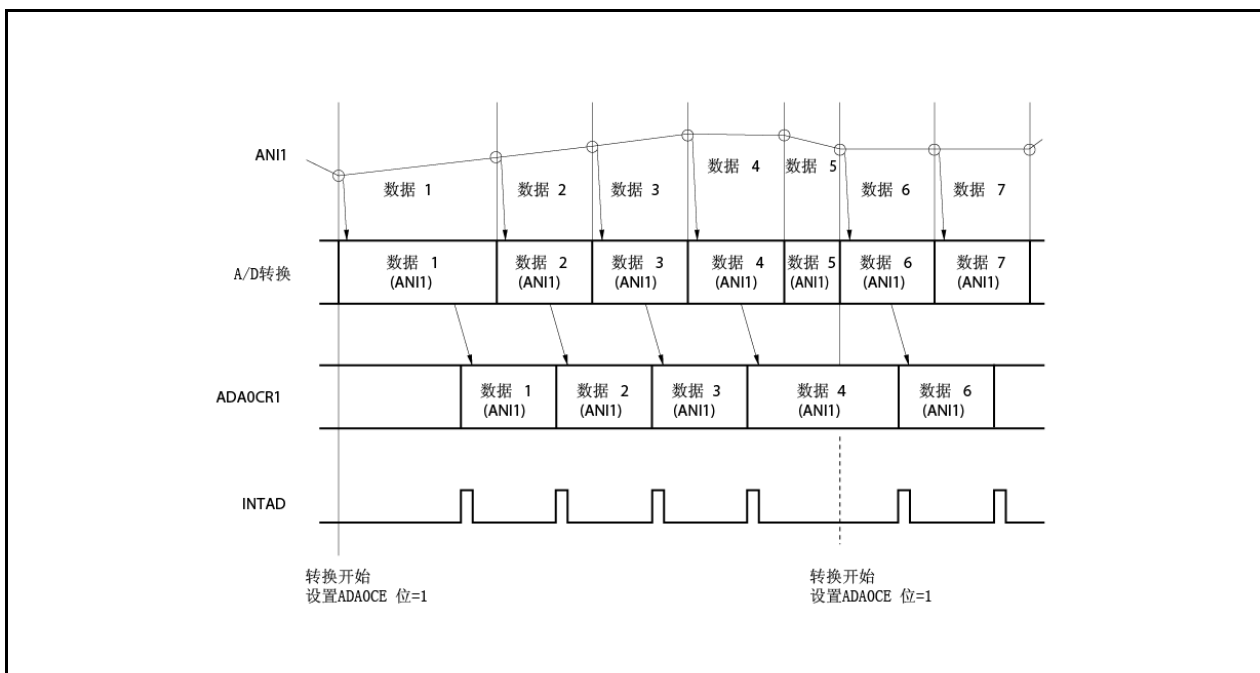
操作模式通过 ADA0M0.ADA0MD1 和 ADA0M0.ADA0MD0 位选择。

#### (1) 连续选择模式

在此模式下，通过 ADA0S 寄存器选择的一模拟输入引脚电压连续转换成数字值。

转换结果存储在对应于模拟输入引脚的 ADA0CRn 寄存器。在此模式下，模拟输入引脚对应于基于一对一的 ADA0CRn 寄存器。每当 A/D 转换完成时，产生 A/D 转换结束中断请求信号(INTAD)。完成转换之后，开始下一次转换，除非 ADA0M0.ADA0CE 位清零(n = 0 到 11)。

图 11-3. 连续选择模式操作时序的示例(ADA0S 寄存器= 01H)



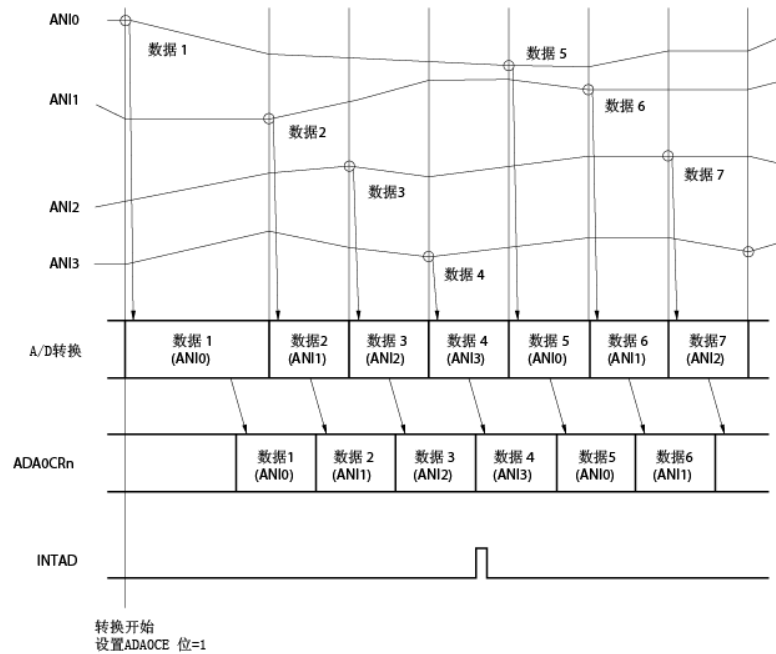
#### (2) 连续扫描模式

在此模式下，继续选择模拟输入引脚，从 ANI0 引脚到由 ADA0S 寄存器指定的引脚，且其值转换成数字值。

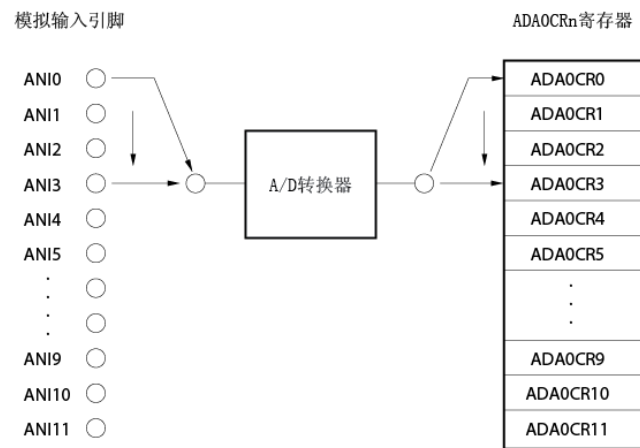
每次转换结果存储在对应于模拟输入引脚的 ADA0CRn 寄存器。当完成由 ADA0S 寄存器指定的模拟输入引脚转换时，产生 INTAD 信号，且 A/D 转换再次从 ANI0 引脚开始，除非 ADA0CE 位清零(n = 0 到 11)。

图 11-4. 连续扫描模式操作时序的示例(ADA0S 寄存器= 03H)

(a) 时序示例



(b) 框图





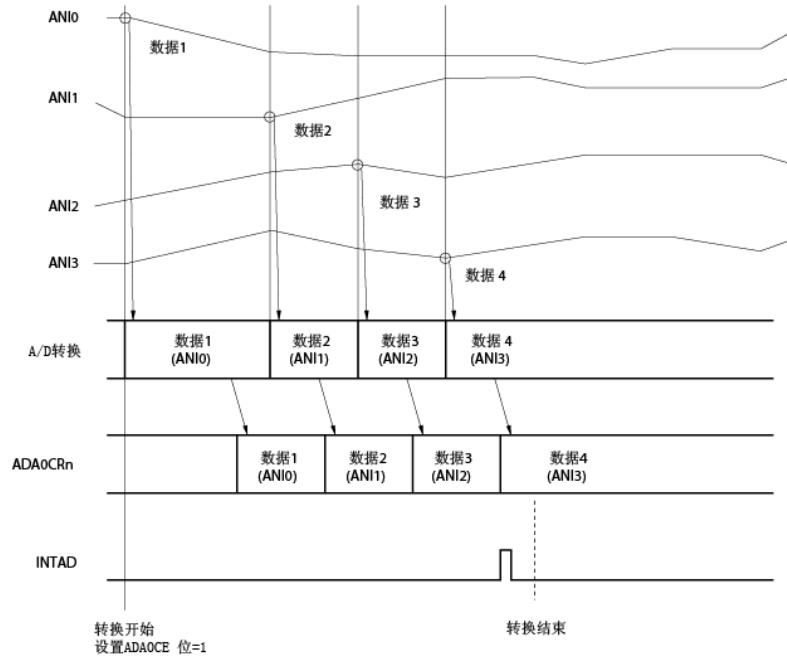
**(3) 单触发扫描模式**

在此模式下，继续选择模拟输入引脚，从 ANI0 引脚到由 ADA0S 寄存器指定的引脚，且其值转换成数字值。

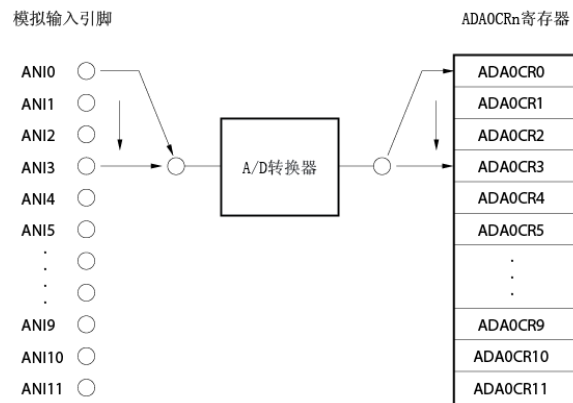
每次转换结果存储在对应于模拟输入引脚的 ADA0CRn 寄存器。当完成由 ADA0S 寄存器指定的模拟输入引脚转换时，产生 INTAD 信号。其完成之后 A/D 转换停止 (n = 0 到 11)。

图 11-5. 单触发扫描模式操作时序的示例 (ADA0S 寄存器= 03H)

(a) 时序示例



(b) 框图



#### 11.5.4 掉电比较模式

A/D 转换结束中断请求信号(INTAD)可如下通过 ADA0PFM 和 ADA0PFT 寄存器控制。

- 当 ADA0PFM.ADA0PFE 位= 0 时，每当转换完成时产生 INTAD 信号(正常使用 A/D 转换器)。
- 当 ADA0PFE 位= 1 时且当 ADA0PFM.ADA0PFC 位= 0 时，当转换完成时将 ADA0CRnH 寄存器的值与 ADA0PFT 寄存器的值进行比较，且仅在  $ADA0CRnH \geq ADA0PFT$  时产生 INTAD 信号。
- 当 ADA0PFE 位= 1 时且当 ADA0PFC 位= 1 时，当转换完成时将 ADA0CRnH 寄存器的值与 ADA0PFT 寄存器的值进行比较，且仅在  $ADA0CRnH < ADA0PFT$  时产生 INTAD 信号。

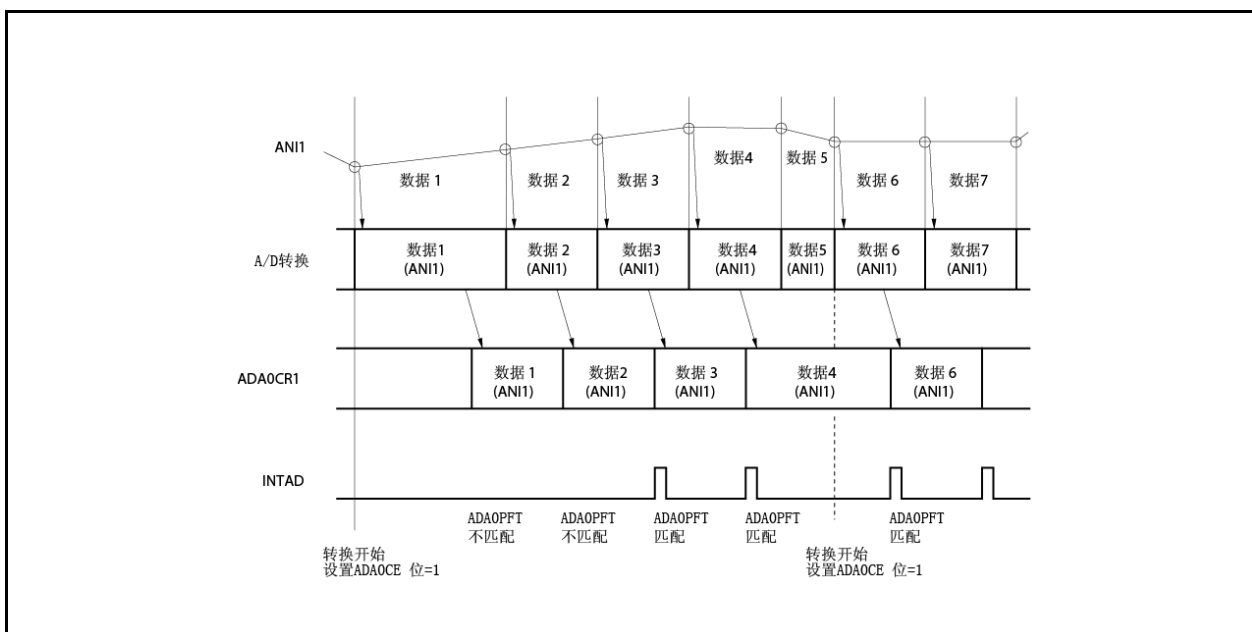
**备注**      n = 0 到 11

在掉电比较模式下，三种模式可用作设置 ANI0 到 ANI11 引脚的模式：连续选择模式，连续扫描模式以及单触发扫描模式。

### (1) 连续选择模式

在此模式下，将由 ADA0S 寄存器指定的模拟输入引脚电压的转换结果与 ADA0PFT 寄存器的设置值进行比较。如果掉电比较结果匹配 ADA0PFC 位设置的条件，则转换结果存储在 ADA0CRn 寄存器，且产生 INTAD 信号。如果不匹配，则转换结果存储在 ADA0CRn 寄存器，且不产生 INTAD 信号。完成第一次转换之后，下一次转换开始，除非 ADA0M0.ADA0CE 位清零(n = 0 到 11)。

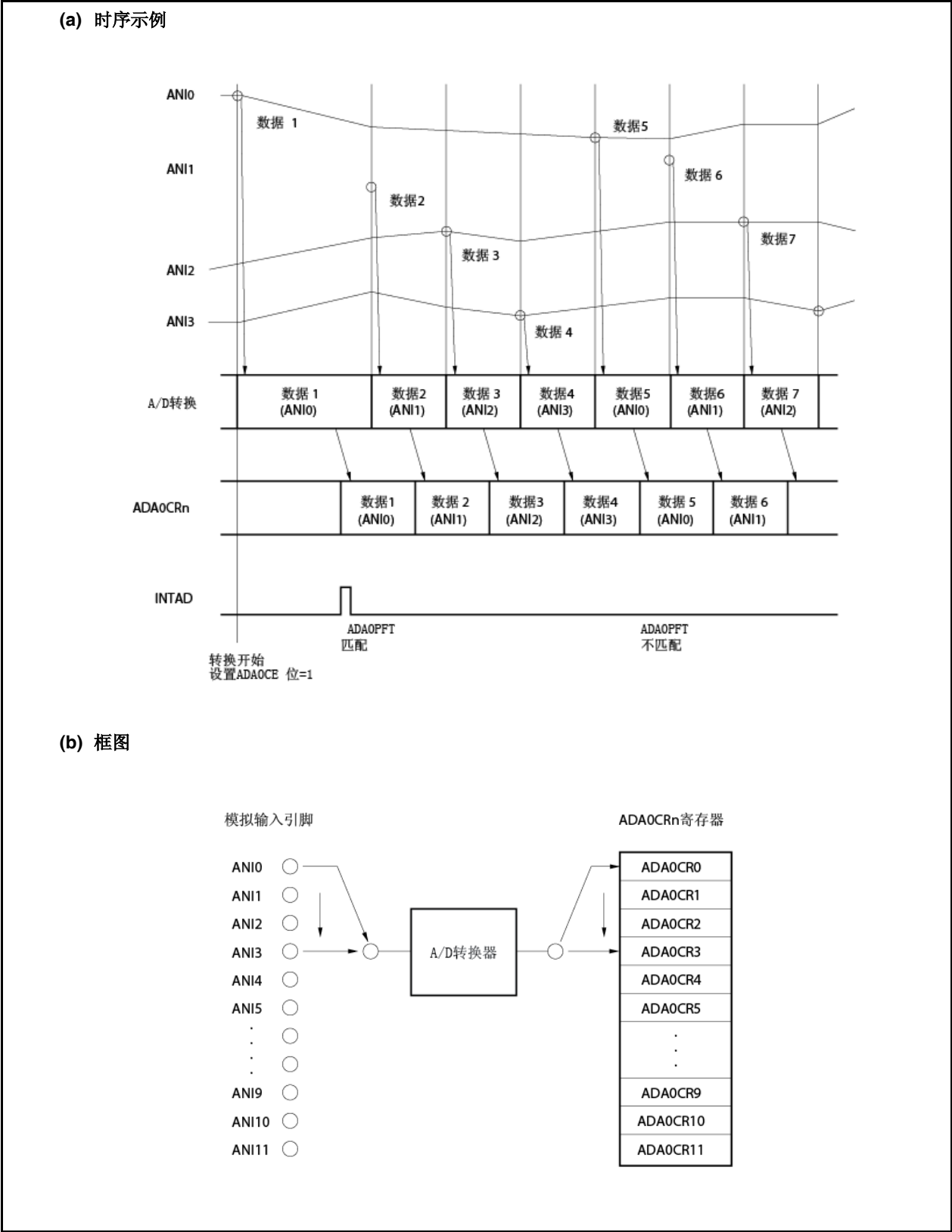
图 11-6.连续选择模式操作时序的示例  
(当进行掉电比较时：ADA0S 寄存器 = 01H)



### (2) 连续扫描模式

在此模式下，存储从 ANI0 引脚到由 ADA0S 寄存器指定的引脚继续选择的模拟输入引脚电压的转换结果，且将通道 0 的 ADA0CR0H 寄存器的设置值与 ADA0PFT 寄存器的值进行比较。如果掉电比较结果匹配由 ADA0PFC 位设置的条件，则转换结果存储在 ADA0CR0 寄存器，且产生 INTAD 信号。如果不匹配，则转换结果存储在 ADA0CR0 寄存器，且不产生 INTAD 信号。第一次转换结果存储在 ADA0CR0 寄存器之后，连续存储继续转换模拟输入引脚上至由 ADA0S 寄存器指定的引脚电压的结果。完成转换之后，下一次转换再次从 ANI0 引脚开始，除非 ADA0CE 位清零。

图 11-7. 连续扫描模式操作时序的示例  
(当进行掉电比较时：ADA0S 寄存器 = 03H)

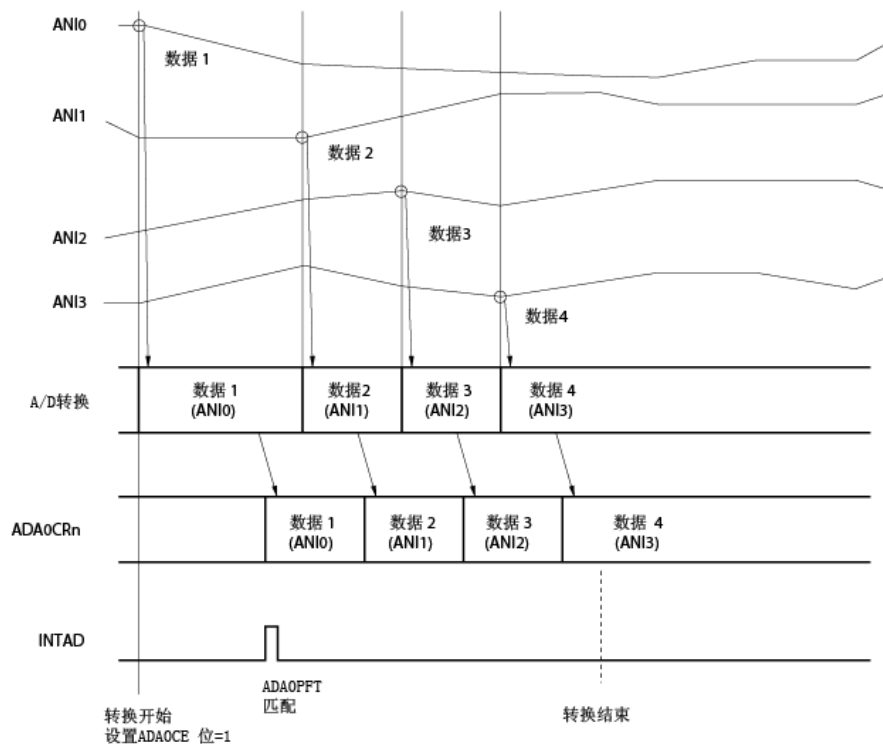


**(3) 单触发扫描模式**

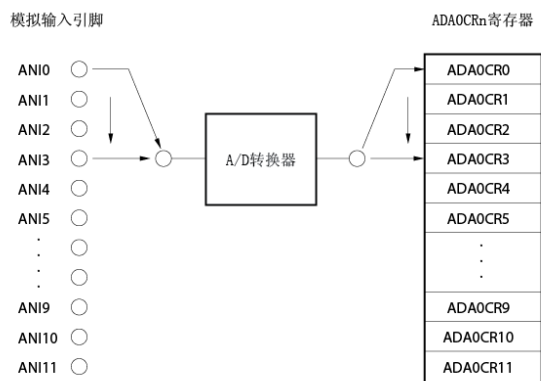
在此模式下，存储继续选择从 ANI0 引脚到由 ADA0S 寄存器指定的引脚的模拟输入引脚电压的转换结果，且将通道 0 的 ADA0CR0H 寄存器的设置值与 ADA0PFT 寄存器的设置值比较。如果掉电比较结果 匹配 ADA0PFC 位的设置条件，则转换结果存储在 ADA0CR0 寄存器且产生 INTAD 信号。如果不匹配，则转换结果存储在 ADA0CR0 寄存器，且不产生 INTAD0 信号。第一次转换结果存储在 ADA0CR0 寄存器之后，结果继续存储转换由 ADA0S 寄存器指定的模拟输入引脚信号的。其完成之后转换停止。

图 11-8 单触发扫描模式操作.时序的示例  
(当进行掉电比较时: ADA0S 寄存器 = 03H)

(a) 时序示例



(b) 框图



## 11.6 注意事项

## (1) 当未使用 A/D 转换器时

当未使用 A/D 转换器时，通过 ADA0M0.ADA0CE 位清零可减少耗电。

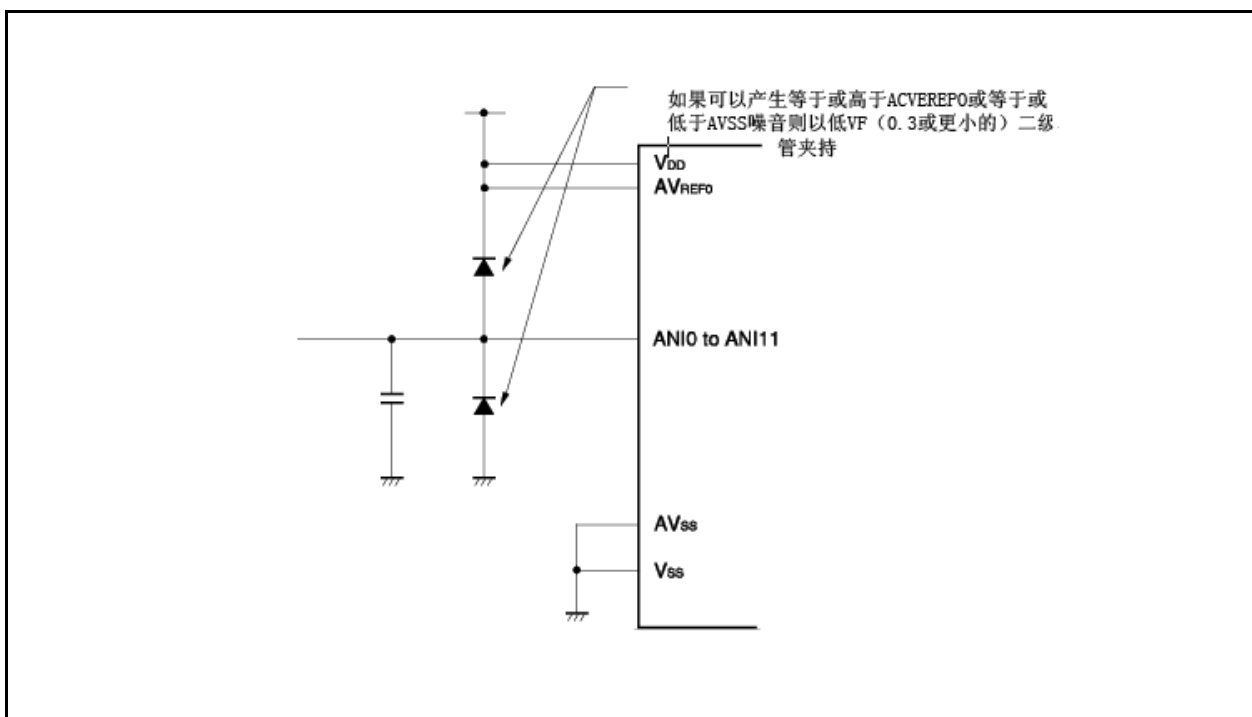
## (2) ANI0 到 ANI11 引脚的输入范围

输入在 ANI0 到 ANI11 引脚指定范围内的电压。如果输入到任何引脚的电压等于或高于  $AV_{REF0}$  或等于或低于  $AV_{SS}$  (即使在绝对最大值范围内)，则未定义通道转换值，且还可能影响其他通道的转换值。

## (3) 噪音对策

要保持 10-位分辨率，必须有效保护 ANI0 到 ANI11 引脚免受噪音干扰。噪音影响随输出模拟输入源的电阻变高而增加。要降低噪音，建议如图 11-9 所示连接外部电容。

图 11-9. 模拟输入引脚处理



## (4) 备用 I/O

模拟输入引脚(ANI0 到 ANI11)另外用作端口引脚。当选择 ANI0 到 ANI11 引脚之一进行 A/D 转换时，转换期间禁止执行读取输入端口或写入输出端口的指令因为转换分辨率可能降低。

同样，如果由于连接到端口引脚的外部电路的影响电流流动，则 A/D 转换期间转换分辨率可能在设为输出端口引脚的引脚处降落。

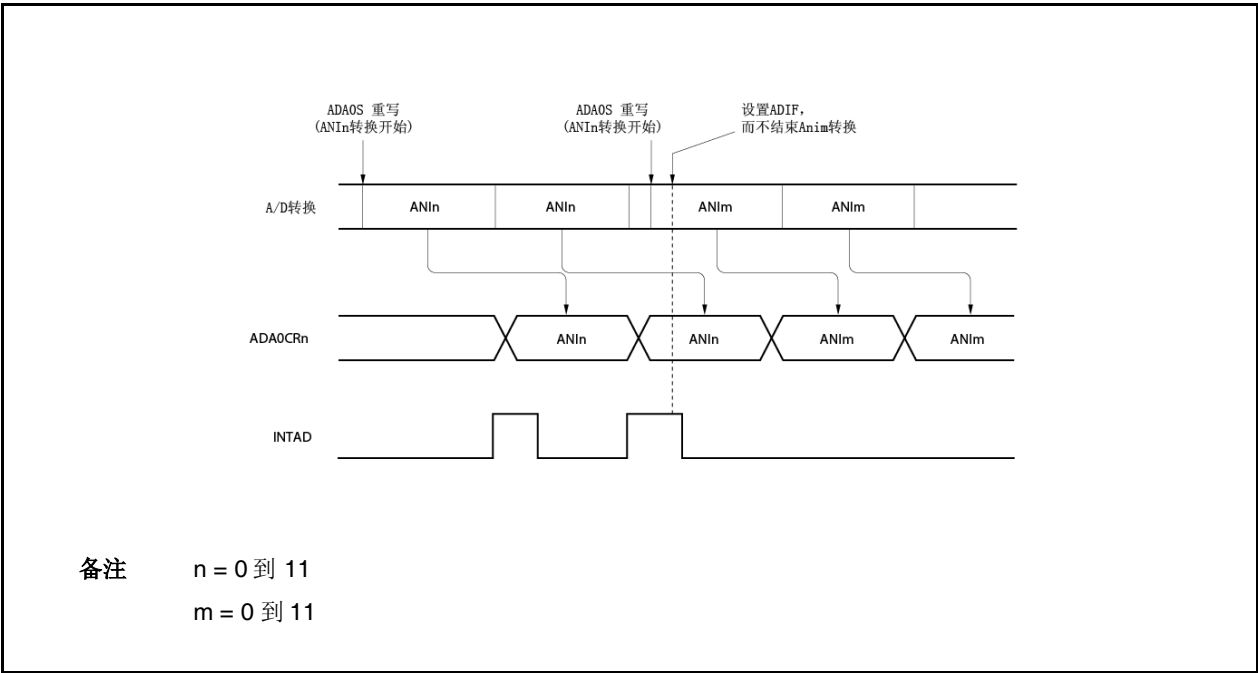
如果数字脉冲施加到靠近转换输入信号引脚的引脚，则由于耦合噪音的影响，A/D 转换值不能作为期望值。因此，禁止将脉冲施加到靠近进行 A/D 转换引脚的引脚。



(5) 中断请求标记(ADIF)

即使 ADA0S 寄存器的内容改变中断请求标记(ADIF)也不清零。因此，如果 A/D 转换期间模拟输入引脚改变，则可存储先前选定模拟输入信号的转换结果且 ADA0S 寄存器重写之前可立即设置转换结束中断请求标记。如果 ADA0S 寄存器重写之后立即读取 ADIF 标记，则即使不完成新选择模拟输入引脚的 A/D 转换也可设置 ADIF 标记。当 A/D 转换停止时，恢复转换之前 ADIF 标记清零。

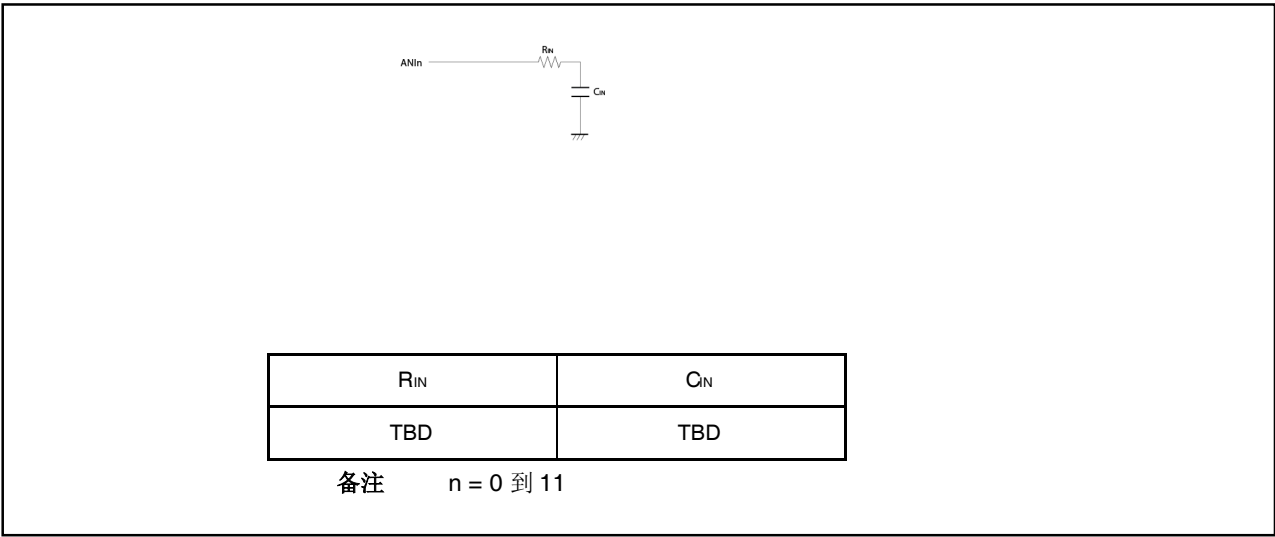
图 11-10. A/D 转换结束中断请求产生的时序



(6) 内部等效电路

以下给出了模拟输入块的等效电路。

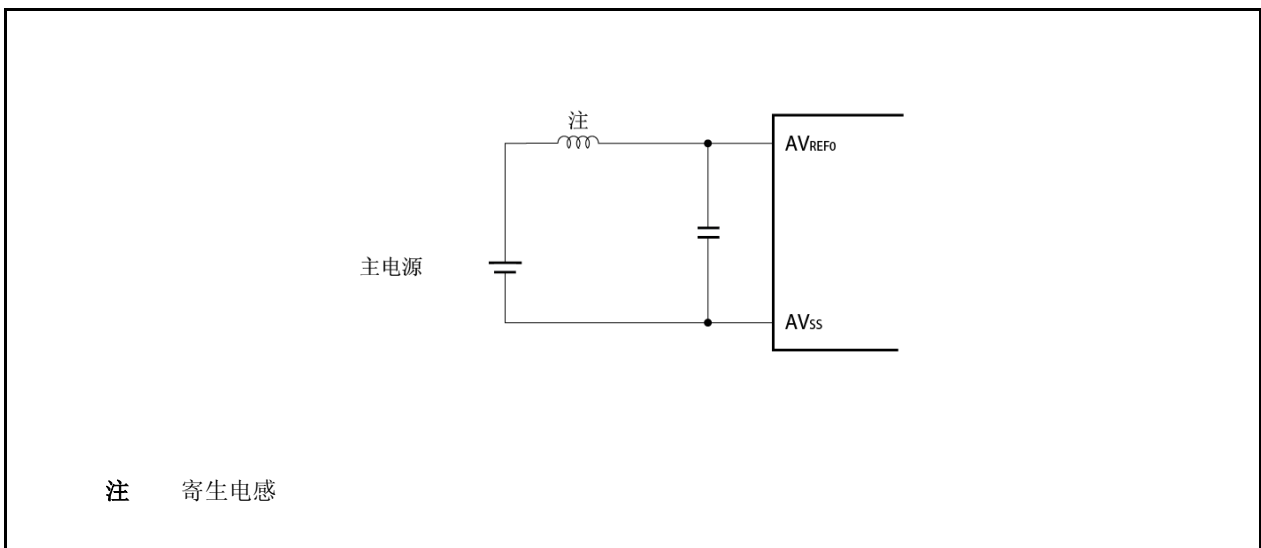
图 11-11. ANIn 引脚内部等效电路



### (7) AV<sub>REF0</sub> 引脚

- (a) AV<sub>REF0</sub> 引脚用作 A/D 转换器的电源引脚且同样向备用-功能端口供电。在其中使用备用电源的应用中，确保供应与 V<sub>DD</sub> 到 AV<sub>REF0</sub> 引脚相同的电压，如图 11-12 所示。
- (b) AV<sub>REF0</sub> 引脚还用作 A/D 转换器的参考电压引脚。如果向 AV<sub>REF0</sub> 引脚供电的源具有高阻抗或如果电源具有低 电流供应能力，参考电压可能发生波动，这是由于转换期间电流流动（尤其，启用转换操作之后位 ADA0CE 即设为 1）。结果，转换精度可能下降。要避免此情况，建议将电容跨 AV<sub>REF0</sub> 和 AV<sub>SS</sub> 引脚连接以抑制参考电压波动，如图 11-12 所示。
- (c) 如果向 AV<sub>REF0</sub> 引脚供电的源具有高 DC 阻抗(例如，因为嵌入二极管)，启用转换时的电压可能低于转换停止时的电压，因为电压降落由 A/D 转换电流所导致。

图 11-12. AV<sub>REF0</sub> 引脚处理示例



<R>

### (8) 读取 ADA0CRn 结果

当写入 ADA0M0 到 ADA0M2, ADA0S, ADA0PFM 或 ADA0PFT 寄存器时，可能未定义 ADA0CRn 寄存器的内容。完成转换之后和写入 ADA0M0 到 ADA0M2, ADA0S, ADA0PFM 或 ADA0PFT 寄存器之前读取转换结果。同样，当确认外部/定时器触发时，可能未定义 ADA0CRn 寄存器的内容。完成转换之后和确认下一外部/定时器触发之前读取转换结果。以上时序之外可能读取不到校正转换结果。

### (9) A/D 转换结果

如果模拟输入引脚及参考电压输入引脚存在噪音，则该噪音可能产生无效的转换结果。需要软件处理以避免此无效的转换结果对系统的负面影响。此软件处理示例如下所示。

- 利用大量 A/D 转换并使其作为 A/D 转换结果。
- 连续执行 A/D 转换数次且使用这些结果，删除可能获得的任何例外的结果。
- 如果获得判断产生系统故障的 A/D 转换结果，则进行故障处理之前确保重复核对系统故障。

&lt;R&gt;

**(10) 备用模式**

因为 A/D 转换器在停止模式下停止操作，所以转换结果无效，因此可减少耗电。操作恢复之后释放停止模式，而释放停止模式之后的 A/D 转换结果无效。当释放停止模式之后使 A/D 转换器，设置 TOP 模式或释放停止模式之前，释放停止模式之后 ADA0M0.ADA0CE 位清零接着 ADA0CE 位设为 1。

在 IDLE1, IDLE2 或副时钟操作模式下，操作继续。因此，要降低耗电，ADA0M0.ADA0CE 位清零。在 IDLE1 和 IDLE2 模式下，因为不能保留模拟输入电压值，所以释放 IDLE1 和 IDLE2 模式之后 A/D 转换结果无效。设置 IDLE1 和 IDLE2 模式之前转换结果有效。

&lt;R&gt;

**(11) 稳定时间期间重写寄存器和触发输入**

稳定时间期间禁止重写 ADA0M0, ADA0M2, ADA0S, ADA0PFM 以及 ADA0PFT 寄存器和触发输入。

**(12) A/D 转换结果的变化**

A/D 转换结果可能根据供应电压波动而变化或可能受噪音影响。要减少变化，请采取措施，例如平均 A/D 转换结果。

**(13) A/D 转换结果滞后特性**

连续比较类型 A/D 转换器在内部采样&保持电容内保持模拟输入电压且接着进行 A/D 转换。A/D 转换完成之后，模拟输入电压保持在内部采样&保持电容内。结果，可能出现以下现象。

- 当 A/D 转换使用相同通道时，如果电压高于或低于先前 A/D 转换，则可能出现滞后特性可能，其中转换结果受先前值影响。因此，即使在相同电压下进行转换，结果也可能不同。
- 当切换模拟输入通道时，可能出现滞后特性，其中转换结果受先前通道值影响。这是因为 A/D 转换器用于 A/D 转换。因此，即使在相同电压下进行转换，结果也可能不同。

&lt;R&gt;

因此，要获得更准确的转换结果，请对相同通道连续执行两次 A/D 转换，且舍弃第一次转换结果。

### 11.7 如何读取 A/D 转换器特性表

本节介绍了有关 A/D 转换器的术语。

#### (1) 分辨率

可确认的最小模拟输入电压，亦即，模拟输入电压与 1 位数字输出的比率称作 1 LSB(最低位)。满刻度的 1 LSB 比率表示为 %FSR(满刻度范围)。%FSR 为表示为百分比的可转换模拟输入电压范围的比率，且可如下表示，而与分辨率无关。

$$\begin{aligned} 1\%FSR &= (\text{可转换模拟输入电压的最大值} - \text{可转换模拟输入电压的最小值}) / 100 \\ &= (AV_{REF0} - 0) / 100 \\ &= AV_{REF0} / 100 \end{aligned}$$

当分辨率为 10 位，1 LSB 如下所示：

$$\begin{aligned} 1 \text{ LSB} &= 1/2^{10} = 1/1,024 \\ &= 0.098\%FSR \end{aligned}$$

精度由总误差确定，而与分辨率无关。

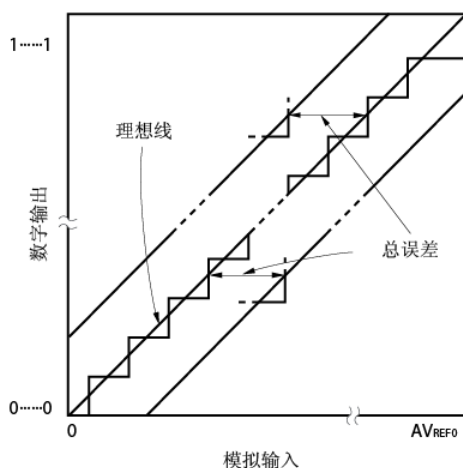
#### (2) 总误差

这是实际测量值与理论值之间的最大差值。

其为零刻度误差，满刻度误差，线性误差以及这些误差组合的总和。

特性表的总误差不包括量化误差。

图 11-13. 总误差

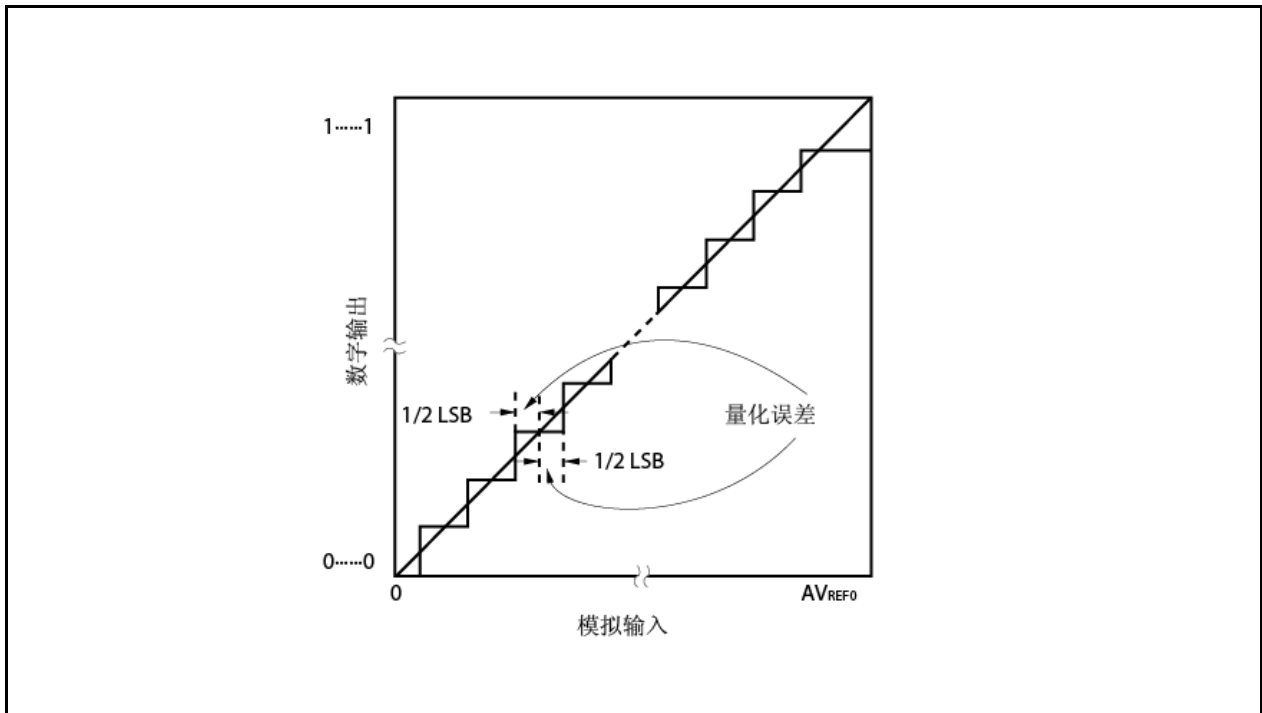


### (3) 量化误差

这是当模拟值转换成数字值时不可避免出现的 $\pm 1/2 \text{ LSB}$  误差。因为 A/D 转换器将 $\pm 1/2 \text{ LSB}$  范围内的模拟输入电压转换为相同数字代码，量化误差不可避免。

此误差不包括在特性表中的总误差，零刻度误差，满刻度误差，积分线性误差或微分线性误差。

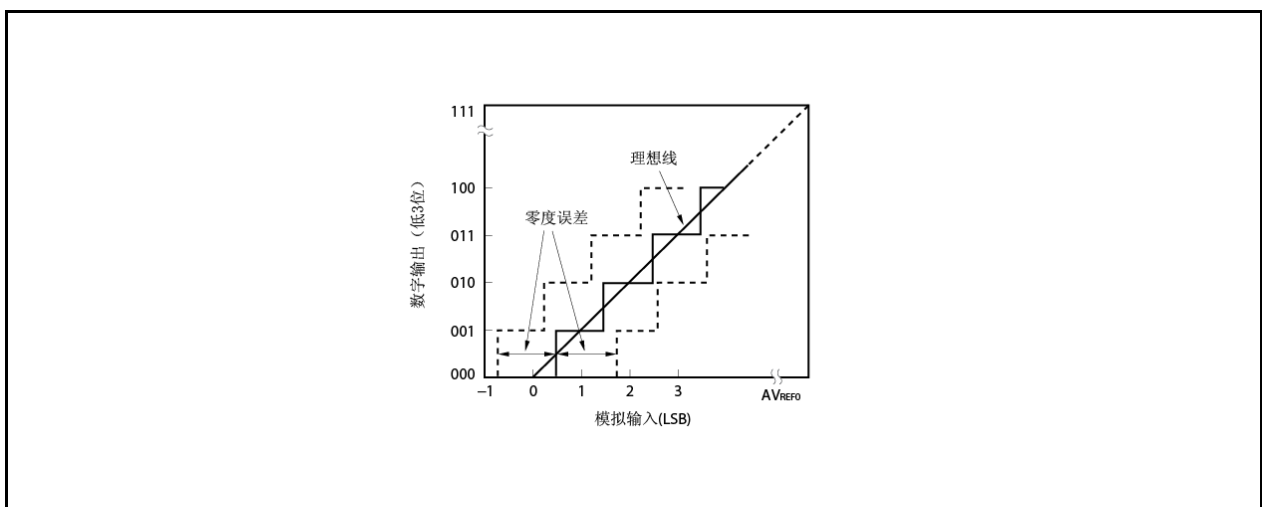
图 11-14. 量化误差



### (4) 零刻度误差

这是当数字输出从 0...000 改为 0...001( $1/2 \text{ LSB}$ )时实际测量模拟输入电压与其理论值之间的差值。

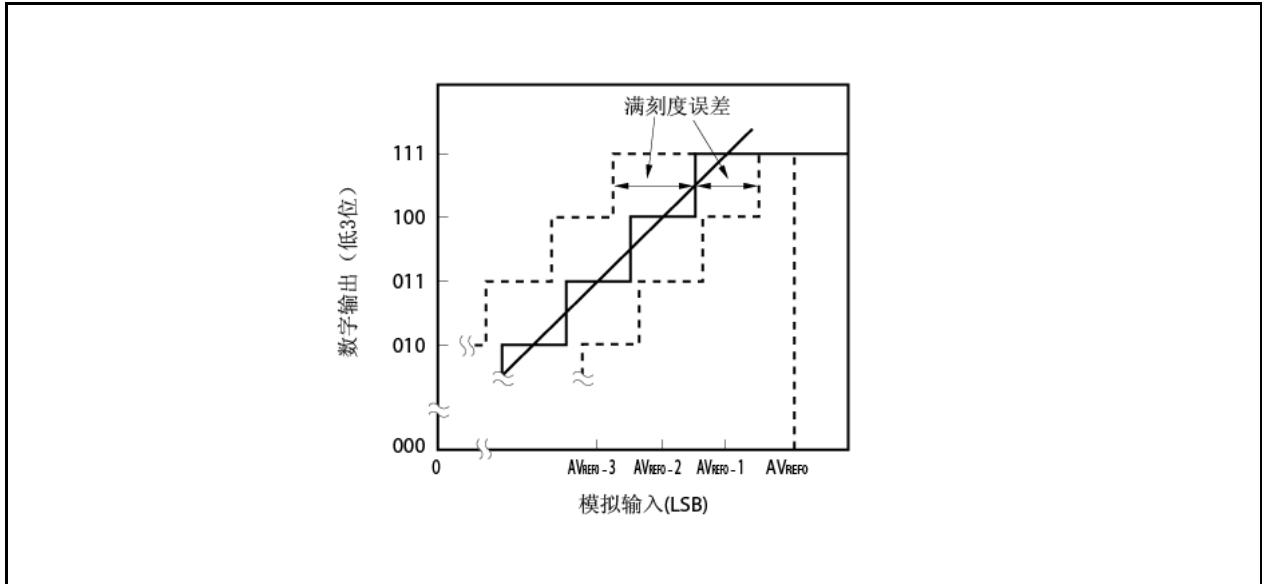
图 11-15. 零刻度误差



### (5) 满刻度误差

这是当数字输出从 1...110 改为 1...111(满刻度(3/2 LSB))时实际测量模拟输入电压与其理论值之间的差值。

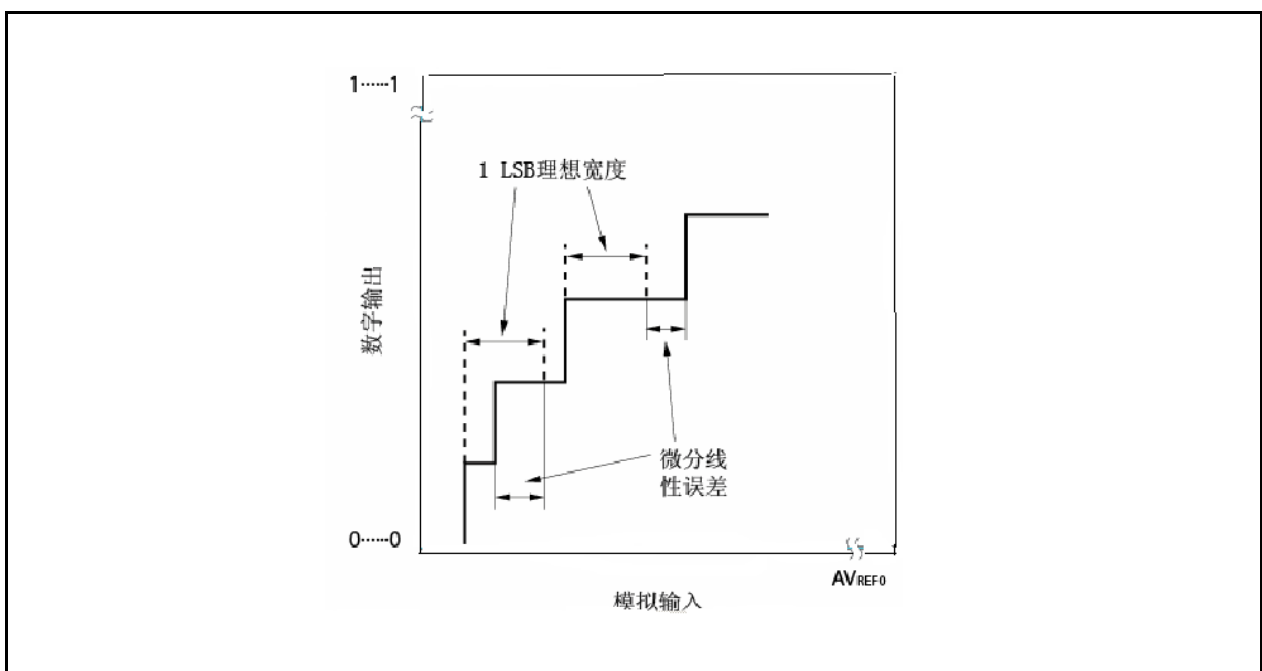
图 11-16. 满刻度误差



### (6) 微分线性误差

理想地，输出绝对代码的宽度为 1 LSB。此误差指示当绝对代码输出时实际测量值与其理论值之间的差值。此指示当应用于相同通道模拟输入引脚的电压从  $AV_{SS}$  到  $AV_{REF0}$  始终增加逐位时 A/D 转换的基本特性。当输入电压增加或降低时或当使用两个或两个以上通道时，请参阅 11.7(2) 总误差。

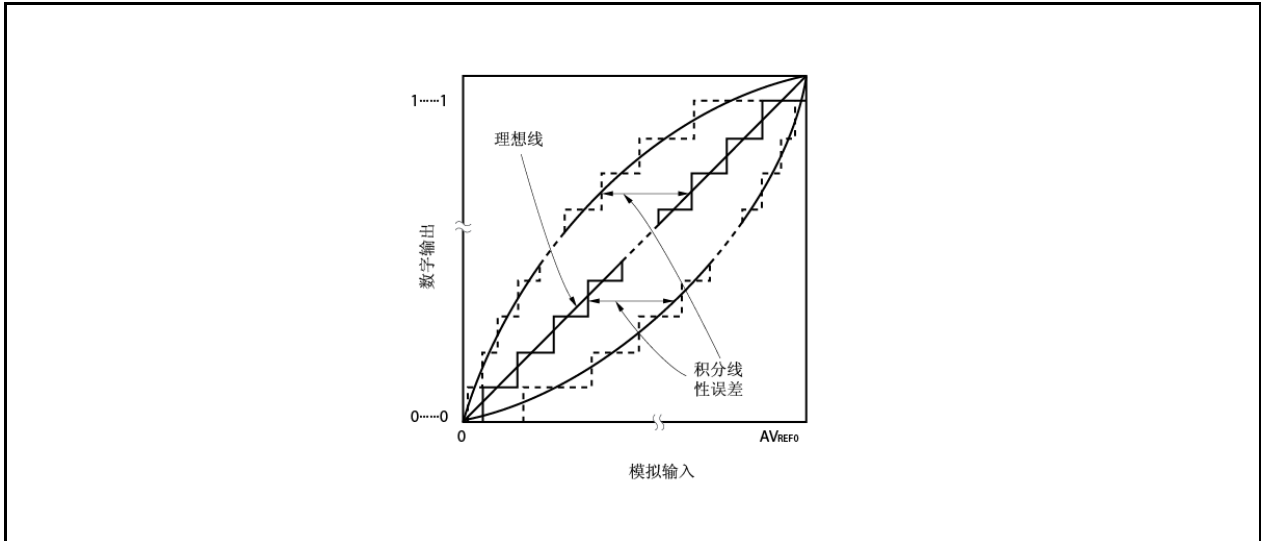
图 11-17. 微分线性误差



### (7) 积分线性误差

此误差指示其转换特性与从理想线性背离程度的关系。其指示实际测量值与其理论值之间的最大差值，其中零刻度误差和满刻度误差为 0。

图 11-18. 积分线性误差



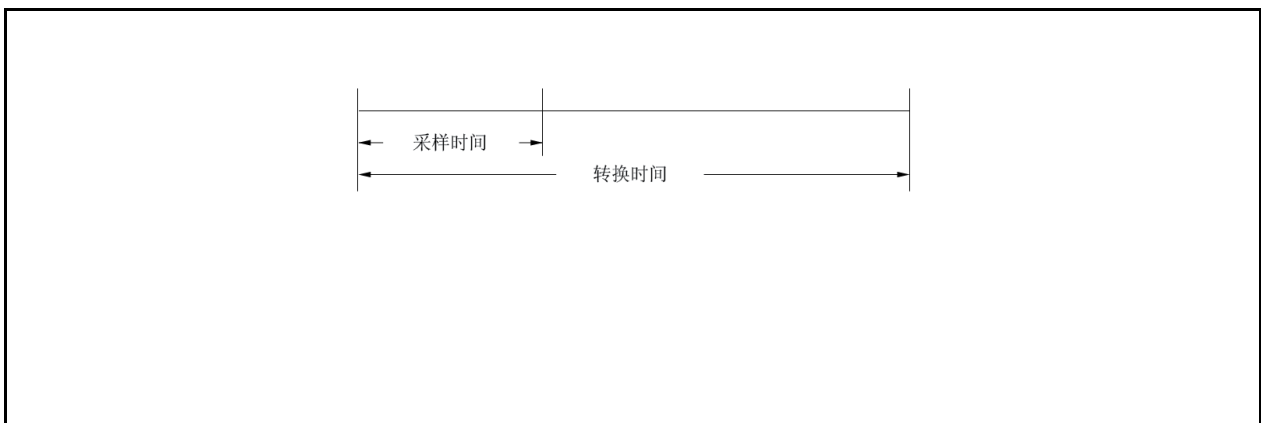
### (8) 转换时间

这是产生每次触发之后获得数字输出需要的时间。  
特性表中的转换时间包括采样时间。

### (9) 采样时间

这是模拟切换开以将模拟电压加载到采样&保持电路的时间。

图 11-19. 采样时间



## 第十二章 异步串行接口 (UARTA)

V850ES/HF2 包括两通道异步串行接口(UARTA)。

### 12.1 特性

- 传输率: 300 bps 到 312.5 kbps(使用内部系统时钟 of 20 MHz 和专用波特率发生器)
- 全双工通信:
  - 内部 UARTAn 接收数据寄存器(UAnRX)
  - 内部 UARTAn 传输数据寄存器(UAnTX)
- 2-引脚配置:
  - TXDAn: 传输数据输出引脚
  - RXDAn: 接收数据输入引脚
- 接收误差输出功能
  - 奇偶校验误差
  - 帧误差
  - 溢出误差
- 中断源: 2
  - 接收完成中断(INTUAnR):

在接收启用状态下通过 ORing 三类接收误差产生中断。其还在传输完成之后当接收数据从接收移位寄存器转移到接收数据寄存器时串口时产生。
  - 传输启用中断(INTUAnT):

在传输启用状态下此中断当传输数据从传输数据寄存器传输到传输移位寄存器时发生。
- 字符长度: 7, 8 位
- 奇偶校验功能: 奇数, 偶数, 0, 无
- 传输停止位: 1, 2 位
- 片上专用波特率发生器
- 可选择 MSB-/LSB-第一次传输
- 可能传输/接收数据反相输入/输出
- SBF(同步中断字段) 可能以 LIN(局部互连网)通信格式传输/接收
  - SBF 传输可选择 13 到 20 位
  - 确认 SBF 接收可能 11 位或 11 位以上
  - 提供的 SBF 接收标记

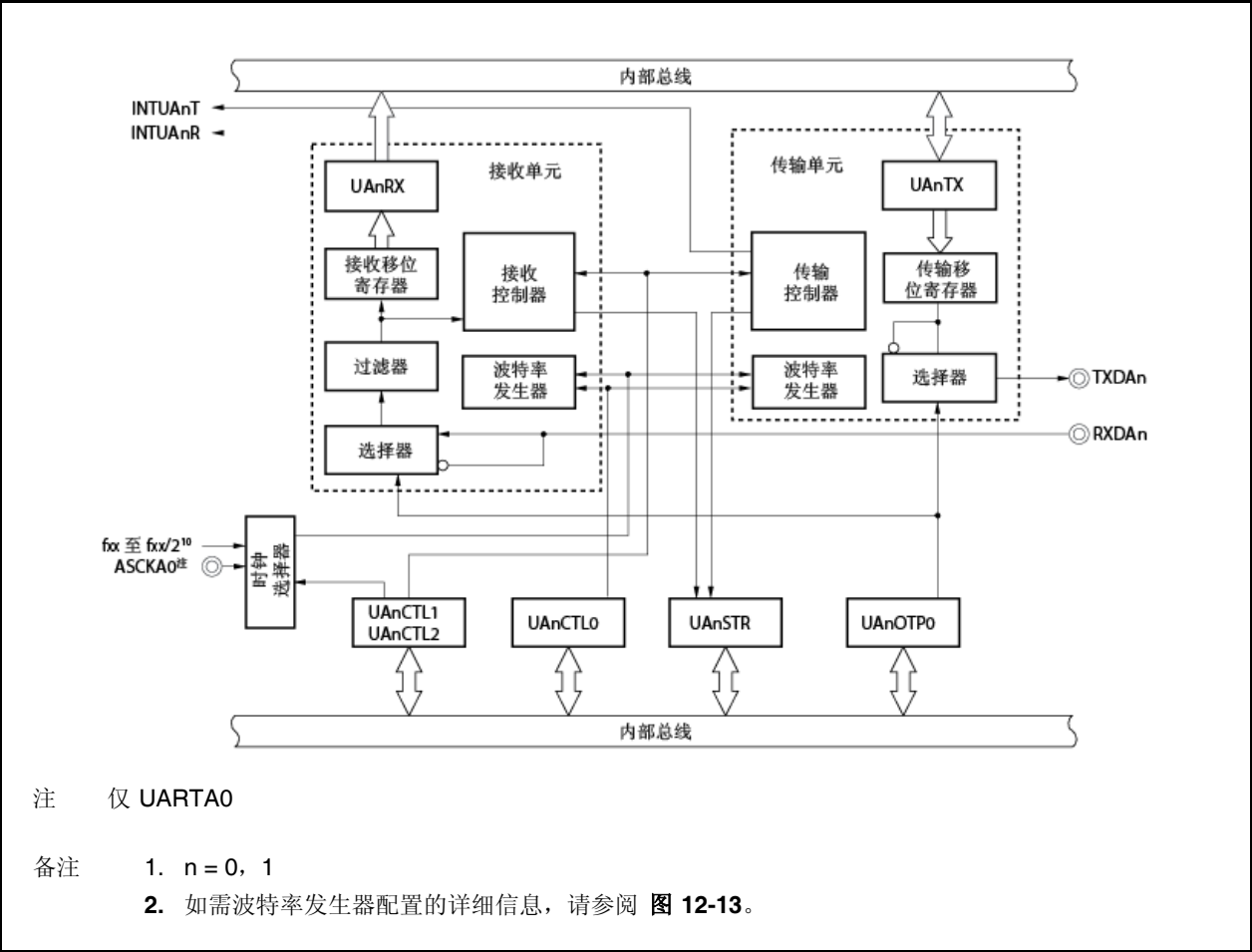
备注      n = 0, 1



12.2 配置

UARTAn 框图如下所示。

图 12-1. 异步串行接口框图



UARTAn 包括以下硬件单元。

表 12-1. UARTAn 配置

项目	配置
寄存器	UARTAn 控制寄存器 0(UAnCTL0) UARTAn 控制寄存器 1(UAnCTL1) UARTAn 控制寄存器 2(UAnCTL2) UARTAn 选项控制寄存器 0(UAnOPT0) UARTAn 状态寄存器(UAnSTR) UARTAn 接收移位寄存器 UARTAn 接收数据寄存器(UAnRX) UARTAn 传输移位寄存器 UARTAn 传输数据寄存器(UAnTX)

**(1) UARTAn 控制寄存器 0(UAnCTL0)**

UAnCTL0 寄存器为用于指定 UARTAn 操作的 8-位寄存器。

**(2) UARTAn 控制寄存器 1(UAnCTL1)**

UAnCTL1 寄存器为用于选择 UARTAn 输入时钟的 8-位寄存器。

**(3) UARTAn 控制寄存器 2(UAnCTL2)**

UAnCTL2 寄存器为用于控制 UARTAn 波特率的 8-位寄存器。

**(4) UARTAn 选项控制寄存器 0(UAnOPT0)**

UAnOPT0 寄存器为用于控制 UARTAn 串口传输的 8-位寄存器。

**(5) UARTAn 状态寄存器(UAnSTR)**

UAnSTRn 寄存器包含当接收误差出现时指示误差内容的标记。当出现接收误差时设置每一接收误差标记 (为 1) 且通过读取 UAnSTR 寄存器复位(为 0)。

**(6) UARTAn 接收移位寄存器**

这是用于将输入到 RXDAn 引脚的串行数据转换为并行数据的移位寄存器。当接收 1 字节数据且检测到停止位时，接收数据转移到 UAnRX 寄存器。

此寄存器不能直接操作。

**(7) UARTAn 接收数据寄存器(UAnRX)**

UAnRX 寄存器为保持接收数据的 8-位寄存器。当接收 7 个字符时，0 存储在最高位(当接收数据 LSB 在先时。在接收启用状态下，接收数据与 1 帧移入处理完成同步从 UARTAn 接收移位寄存器转移到 UAnRX 寄存器。传输到 UAnRX 寄存器还导致输出接收完成中断请求信号(INTUAnR)。

**(8) UARTAn 传输移位寄存器**

传输移位寄存器为用于将从 UAnTX 寄存器传输的并行数据转换成串行数据的移位寄存器。

当从 UAnTX 寄存器转移 1 字节数据时，移位寄存器数据从 TXDAn 引脚输出。

此寄存器不能直接操作。

**(9) UARTAn 传输数据寄存器(UAnTX)**

UAnTX 寄存器为 8-位传输数据缓冲器。当传输数据写入 UAnTX 寄存器时传输开始。当数据可写入 UAnTX 寄存器(当一帧数据从 UAnTX 寄存器转移到 UARTAn 传输移位寄存器)时，产生传输启用中断请求信号(INTUAnT)。

12.3 寄存器

(1) UARTAn 控制寄存器 0(UAnCTL0)

UAnCTL0 寄存器为控制 UARTAn 串口传输操作的 8-位寄存器。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 10H。

(1/2)

复位后: 10H    R/W    地址:    UA0CTL0 FFFFFFFA00H, UA1CTL0 FFFFFFFA10H

	7	6	5	4	3	2	1	0
UAnCTL0	UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL

(n = 0, 1)

UAnPWR	UARTAn操作控制
0	禁用UARTAn操作 (UARTAn异步复位)
1	启用UARTAn操作
通过UAnPWR控制UARTAn操作。TXDAn引脚输出固定为高电平以使UAnPWR位清零 (如果UAnOPT0. UAnTDL位=1则固定为低电平)	

UAnTXE	传输操作启用
0	禁用传输操作
1	启用传输操作
<ul style="list-style-type: none"> <li>要开始传输, 请将UAnPWR位设为1并接着将UAnTXE位设为1。</li> <li>要停止传输, 请将UAnTXE清零, 并接着将UAnTXE位清零。</li> <li>要初始化传输单元, 请将UAnTXE位清零, 等待两个基本时钟周期, 并接着再次将UAnTXE位设为1, 否则将不能进行初始化 (如需基本时钟的详细信息, 请参阅12.6(1)(a)基本时钟)</li> </ul>	

UAnRXE	接收操作启用
0	禁用接收操作
1	启用接收操作
<ul style="list-style-type: none"> <li>要开始接收, 请将UAnPWR位设为1并接着将UAnRXE位设为1</li> <li>要停止接收, 请将UAnPWR位清零并接着将UAnRXE位清零。</li> <li>要初始化接收单元, 请将UAnRXE位清零, 等待两个基本时钟周期, 并接着再次将UAnTXE位设为1, 否则将不能进行初始化 (如需基本时钟的详细信息, 请参阅12.6(1)(a)基本时钟)</li> </ul>	

<R>

UAnDIR	传输方向选择
0	MSB-第一次传输
1	LSB-第一次传输
<ul style="list-style-type: none"><li>• 仅当UAnPWR位=0或UAnTXE位= UAnRXE位=0时此寄存器可重写</li><li>• 当在LIN模式下进行传输和接收时，将UAnDIR位设为1。</li></ul>	

UAnPS1	UAnPS0	传输期间的奇偶校验选择	接收期间的奇偶校验选择
0	0	无奇偶校验输出	无奇偶校验接收
0	1	0 奇偶校验接收	无奇偶校验接收
1	0	奇校验输出	奇校验核对
1	1	偶校验输出	偶校验核对
<ul style="list-style-type: none"><li>• 仅当UAnPWR位=0或UAnTXE位= UAnRXE位=0时此寄存器可重写</li><li>• 如果在接收期间选择“0奇偶校验接收”，则不进行奇偶校验。从而不设置UAnSTR. UanPE位</li><li>• 当在LIN模式下进行传输和接收时，UanPS1和UAnPS0位清零。</li></ul>			

<R>

UAnCL	传输/接收1帧数据的数据字符长度规格
0	7位
1	8位
<ul style="list-style-type: none"><li>• 仅当UAnPWR位=0或UAnTXE位= UAnRXE位=0时此寄存器可重写</li><li>• 当在LIN模式下进行传输和接收时，UanCL位设为1。</li></ul>	

UAnSL	当在LIN模式下进行传输和接收时，UanCL位设为1。
0	1位
1	2位
仅当UAnPWR位=0或UAnTXE位= UAnRXE位=0时此寄存器可重写0.	

备注      如需奇偶校验的详细信息，请参阅 **12.5.9 奇偶校验类型和操作**.

(2) UARTAn 控制寄存器 1(UAnCTL1)

如需详细信息，请参阅 12.6(2) UARTAn 控制寄存器 1(UAnCTL1)。

(3) UARTAn 控制寄存器 2(UAnCTL2)

如需详细信息，请参阅 12.6(3)UARTAn 控制寄存器 2(UAnCTL2)。

(4) UARTAn 选项控制寄存器 0(UAnOPT0)

UAnOPT0 寄存器为控制 UARTAn 寄存器串口传输操作的 8-位寄存器。

此寄存器可读取或写入 8-位或 1-位单元。

此寄存器重设为 14H。

(1/2)

复位后: 14H      R/W      地址:      UA0OPT0 FFFFA03H, UA1OPT0 FFFFA13H

7	6	5	4	3	2	1	0
UAnSRF	UAnSRT	UAnSTT	UAnSLS2	UAnSLS1	UAnSLS0	UAnTDL	UAnRDL

(n = 0, 1)

UAnSRF	SBF接收标记
0	当设置UAnCTL0. UAnPWR位=UAnCTL0. UAnRXE位=0时 同样当SBF接收正常结束时
1	在SBF接收时
•在LIN通信期间判断SBF（同步间隔字段） •当出现SBF接收错误时UAnSRF保持为1，且接着SBF接收再次开始 •UAnSRF为只读位	

UAnSRT	SBF接收触发
0	—
1	SBF接收触发
•在LIN通信期间此为SBF接收触发位，且当读取时，始终读取“0”。 对于接收，设置UAnSRT位(为1)以启用SBF接收 •在设置UAnPWR位=UAnRXE位=1之后设置UAnSRT位	

UAnSTT	SBF接收触发
0	—
1	SBF接收触发
•在LIN通信期间此为SBF接收触发位，且当读取时，始终读取“0”。 •在设置UAnPWR位=UAnRXE位=1之后设置UAnSRT位	

注意事项 SBF 接收(UAnSRF 位= 1)期间禁止设置 UAnSRT 和 UAnSTT 位(为 1)。

<R>

UAnSLS2	UAnSLS1	UAnSLS0	SBF传输长度选择
1	0	1	13-位输出（复位值）
1	1	0	14-位输出
1	1	1	15-位输出
0	0	0	16-位输出
0	0	1	17-位输出
0	1	0	18-位输出
0	1	1	19-位输出
1	0	0	20-位输出

当UAnPWR位=0 或当UAnTXE位=0时可设置此寄存器

UAnTDL	传输电平位
0	传输数据的正常输出
1	传输数据的反相输出

- 使用UAnTDL位可反相TXDAn引脚的输出电平
- 当UAnPWR位=0 或当UAnTXE位=0时可设置此寄存器

UAnRDL	接收数据电平位
0	传输数据的正常输入
1	传输数据的反相输入

- 使用UAnRDL位可反相RXDAn引脚的输出电平
- 当UAnPWR位=0 或当UAnTXE位=0时可设置此寄存器

#### (5) UARTAn 状态寄存器(UAnSTR)

UAnSTR 寄存器为显示 UARTAn 传输状态并接收误差内容的 8-位寄存器。

当 UAnPE, UAnFE 以及 UAnOVE 位可读写时，此寄存器可读取或写入 8-位或 1-位单元，而 UAnTSF 位为只读位。然而，这些位仅可通过写入 0 清零；其不能通过写入 1 设置(即使写入 1，也保留该值)。

初始化条件如下所示。

寄存器/位	初始化条件
UAnSTR 寄存器	<ul style="list-style-type: none"> <li>• 复位</li> <li>• UAnCTL0.UAnPWR = 0</li> </ul>
UAnTSF 位	<ul style="list-style-type: none"> <li>• UAnCTL0.UAnTXE = 0</li> </ul>
UAnPE, UAnFE, UAnOVE 位	<ul style="list-style-type: none"> <li>• 写入 0</li> <li>• UAnCTL0.UAnRXE = 0</li> </ul>

复位后: 00H R/W 地址: UA0STR FFFFFFFA04H, UA1STR FFFFFFFA14H

	7	6	5	4	3	2	1	0
UAnSTR	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

(n = 0, 1)

UAnTSF	传输状态标记
0	<ul style="list-style-type: none"> <li>• 当已设置UAnPWR位=0 或UAnTXE位=0时</li> <li>• 当以下传输完成后, UAnTX寄存器就不再进行后续的数据输出</li> </ul>
1	写入UAnTX寄存器
<ul style="list-style-type: none"> <li>• 在进行连续传输时, UAnTSF 位始终为1。</li> <li>• 在启用传输单元时, 检查UAnTSF位是否为0。</li> <li>• 否则如果UAnTSF位= 1, 传输数据将不受保障</li> </ul>	

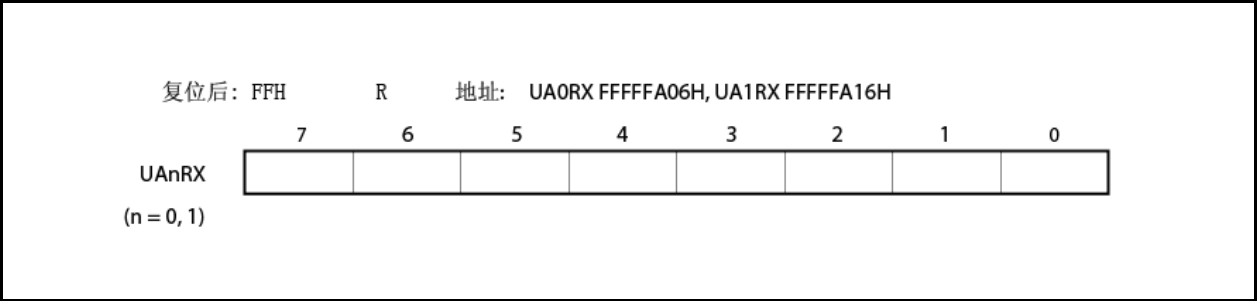
UAnPE	奇偶检验错误标记
0	<ul style="list-style-type: none"> <li>• 当已设置UAnPWR位=0 或UAnTXE位=0时</li> <li>• 当已写入0时</li> </ul>
1	当接收期间数据及奇偶校验位的奇偶校验不匹配时
<ul style="list-style-type: none"> <li>• UAnPE位的操作由UAnCTL0. UAnPS1和UAnCTL0. UAnPS0位的设置来控制。</li> <li>• 可读写UAnPE位, 且仅可通过写入0来将其清零, 且其不可通过写入1来设置, 当此位写入1时, 该值保留。</li> </ul>	

UAnFE	帧错误标记
0	<ul style="list-style-type: none"> <li>• 当已设置UAnPWR位=0 或UAnTXE位=0时</li> <li>• 当已写入0时</li> </ul>
1	当接收期间未检测到停止位时
<ul style="list-style-type: none"> <li>• 仅核对接收数据停止位的第一位, 而不管UAnCTL0, UanSL位的值。</li> <li>• 可以读写UanFE位, 且仅可通过写入0来将其清零, 且其不可通过写入1来设置, 当此位写入1时, 该值保留。</li> </ul>	

UAnOVE	溢出错误标记
0	<ul style="list-style-type: none"> <li>• 当已设置UAnPWR位=0 或UAnTXE位=0时</li> <li>• 当已写入0时</li> </ul>
1	当接收数据已设为UanRX寄存器, 且在已读取接收数据之前完成下一接收操作时。
<ul style="list-style-type: none"> <li>• 当发生溢出错误时, 舍弃此数据而不将下一接收数据写入接收缓冲器。</li> <li>• 可以读写UAnFE位, 且仅可通过写入0来将其清零, 且其不可通过写入1来设置, 当此位写入1时, 该值保留。</li> </ul>	

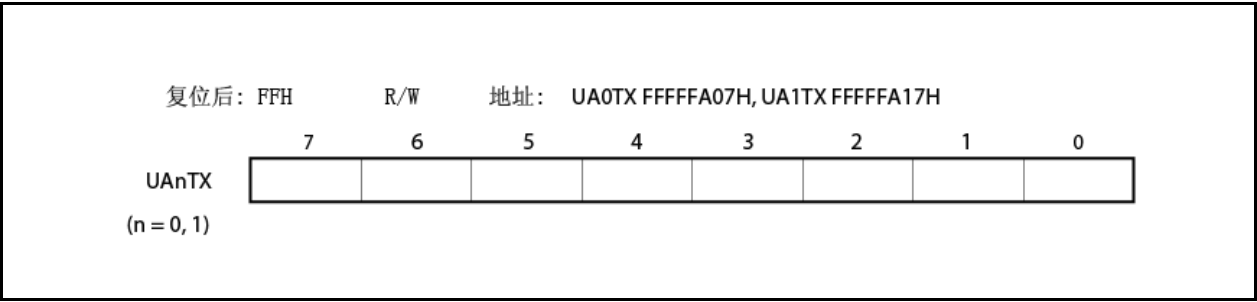
(6) UARTAn 接收数据寄存器(UAnRX)

UAnRX 寄存器为存储由接收移位寄存器转换的并行数据的 8-位缓冲寄存器。  
当完成 1 字节数据接收时存储在接收移位寄存器中的数据转移到 UAnRX 寄存器。  
LSB-在先接收期间当数据长度指定为 7 位时,接收数据转移到 UAnRX 寄存器的位 6到 0 且 MSB 始终为 0。MSB-在先接收期间,接收数据转移到 UAnRX 寄存器的位 7 到 1 且 LSB 始终为 0。  
当发生溢出误差(UAnOVE)时,此时接收数据不转移到 UAnRX 寄存器,并将其舍弃。  
此寄存器为 8-位单元的只读寄存器。  
除复位输入之外,通过 UAnCTL0.UAnPWR 位清零 UAnRX 寄存器可设为 FFH。



(7) UARTAn 传输数据寄存器(UAnTX)

UAnTX 寄存器为用于设置传输数据的 8-位寄存器。  
此寄存器可读取或写入 8-位单元。  
此寄存器重设为 FFH。





12.4 中断请求信号

从 UARTAn 产生以下两个中断请求信号。

- 接收完成中断请求信号(INTUAnR)
- 传输启用中断请求信号(INTUAnT)

这两个中断请求信号的默认优先级为接收完成中断请求信号之后是传输启用中断请求信号。

表 12-2. 中断和其默认优先级

中断	优先级
接收完成	高
传输启用	低

(1) 接收完成中断请求信号(INTUAnR)

在接收启用状态当数据转移到接收移位寄存器时且传输到 UAnRX 寄存器时输出接收完成中断请求信号。  
当出现接收误差时还输出接收完成中断请求信号。因此，当接收接收完成中断请求信号并读取数据时，读取 UAnSTR 寄存器且核对接收结果是否为误差。  
在接收禁用状态下产生未完成接收中断请求信号。

(2) 传输启用中断请求信号(INTUAnT)

如果传输启用时传输数据从 UAnTX 寄存器转移到 UARTAn 传输移位寄存器，则产生传输启用中断请求信号。

<R>

## 12.5 操作

### 12.5.1 数据格式

执行全双工串行数据接收和传输。

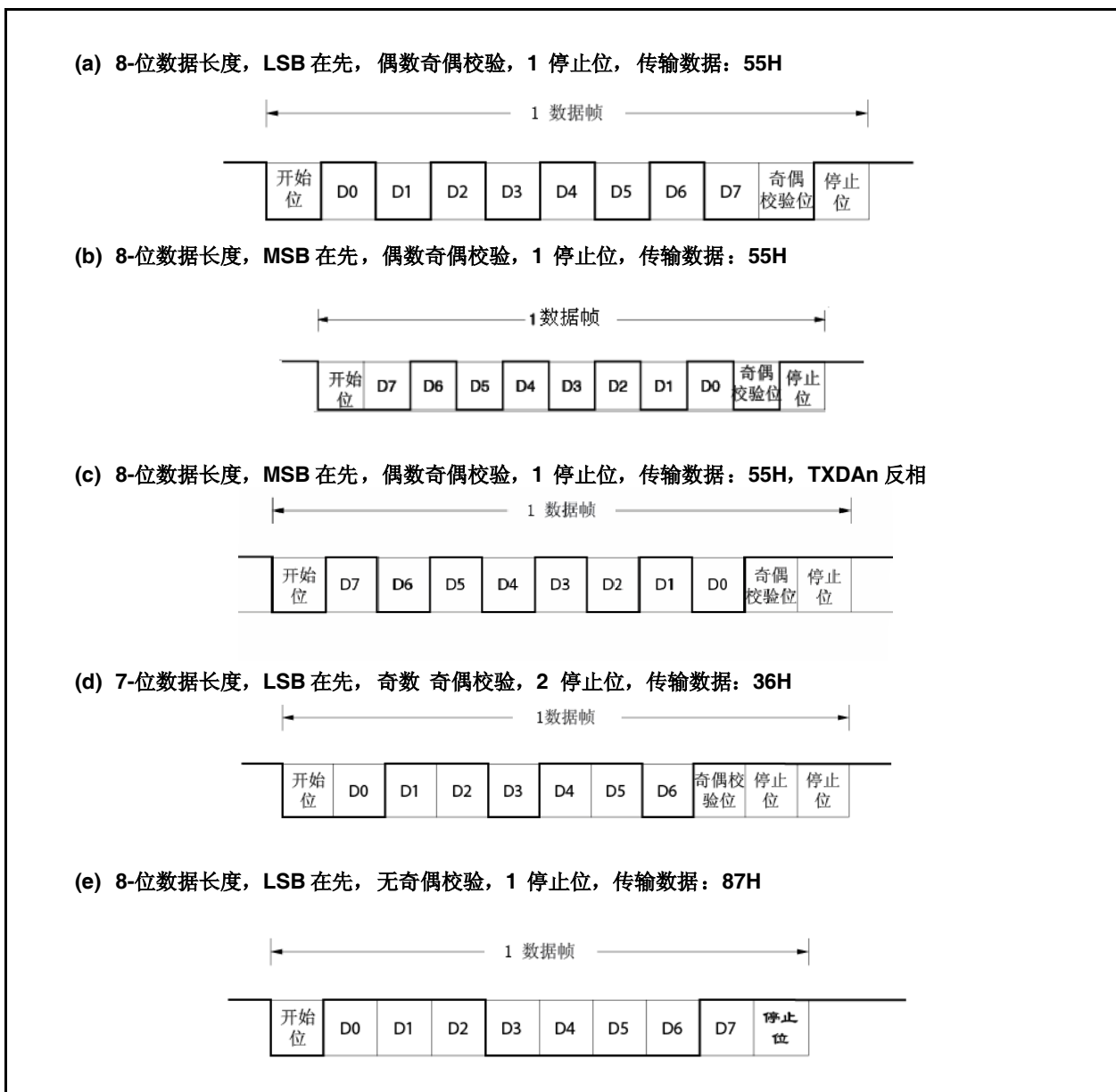
如图所示 12-2，一数据帧传输/接收数据包括起始位，字符位，奇偶校验位以及停止位。

1 数据帧内的字符位长度规格，奇偶校验选择，停止位长度规格以及 MSB/LSB-在先传输规格使用 UAnCTL0 寄存器执行。

此外， TXDAn 位的 UART 输出/反相输出控制使用 UAnOPT0.UAnTDL 位进行。

- 起始位 ..... 1 位
- 字符位 ..... 7 位/8 位
- 奇偶校验位 ..... 偶数奇偶校验/奇数奇偶校验/0 奇偶校验/无奇偶校验
- 停止位 ..... 1 位/2 位

图 12-2. UART 传输/接收数据格式



### 12.5.2 SBF 传输/接收格式

V850ES/HF2 具有 SBF(同步中断字段)传输/接收控制功能以启用 LIN 功能的使用。

**备注** LIN 表示局部互连网且为意欲辅助汽车网络降低成本的低-速(1 到 20 kbps)串口通信协议。  
LIN 通信为单-主机通信，且一台主机可连接最多 15 个分机。  
LIN 分机用于控制开关，制动器及传感器，且这些经 LIN 网络连接到 LIN 主机。  
通常，LIN 主机连接到网络，例如 CAN(控制器局域网)。  
此外，LIN 总线使用单-线方法且经符合 ISO9141 的收发器连接到节点。  
在 LIN 协议中，主机传输具有波特率信息的帧且分机接收该帧并校正波特率误差。因此，当分机的波特率误差处于 $\pm 15\%$ 或更少时可能进行通信。

图 12-3 和 12-4 概述 LIN 的传输和接收操作。

图 12-3. LIN 传输操作概述

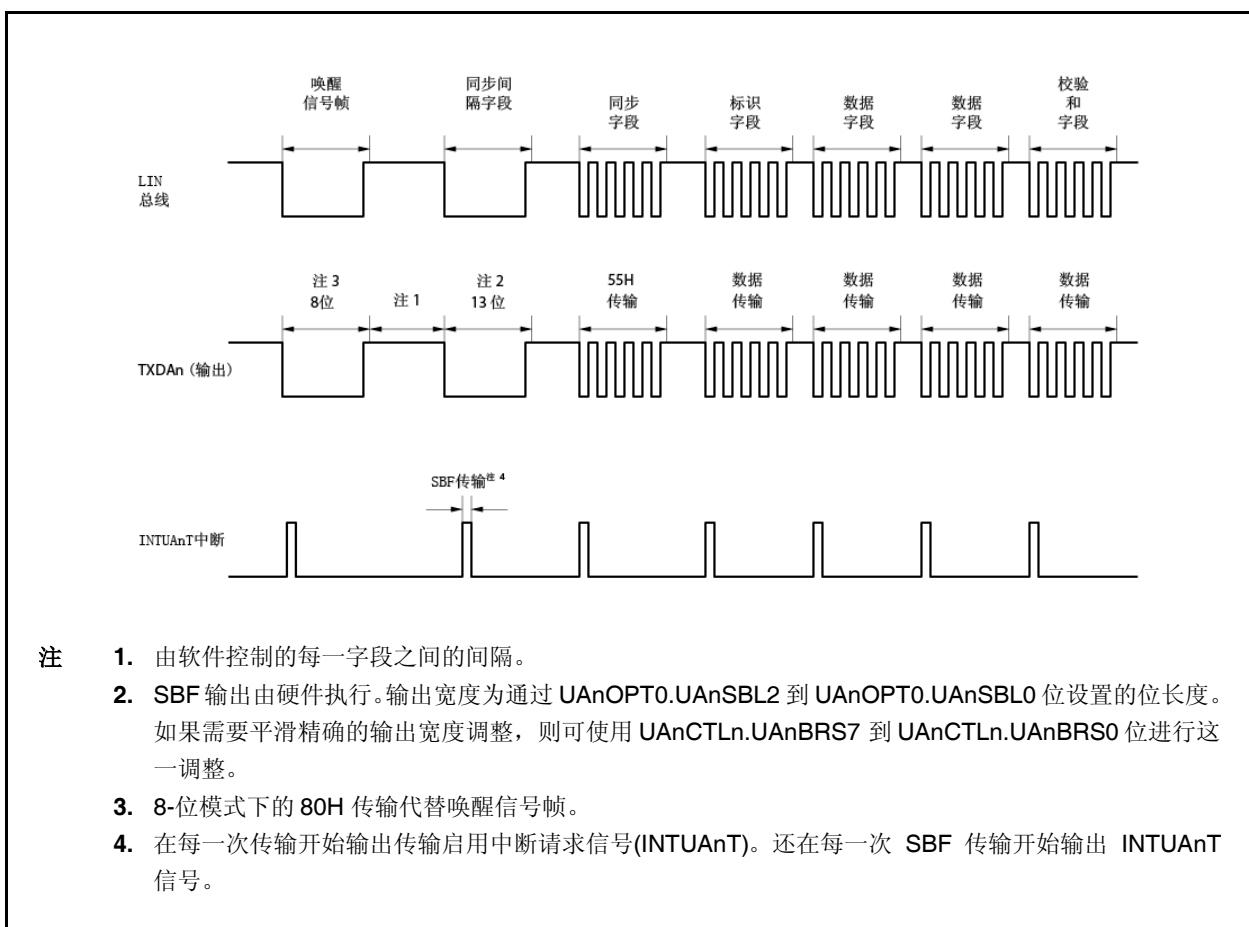
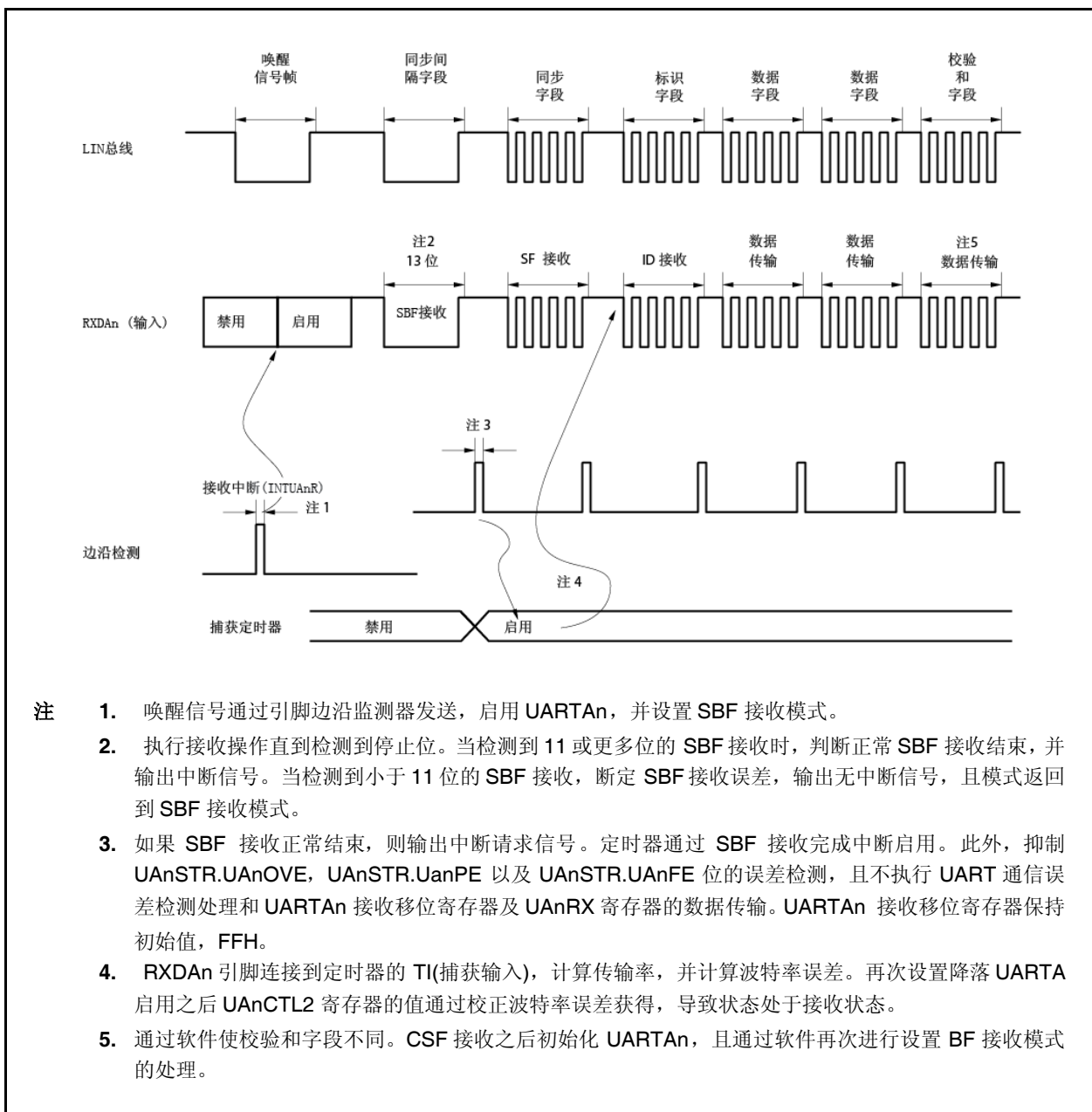


图 12-4. LIN 接收操作概述



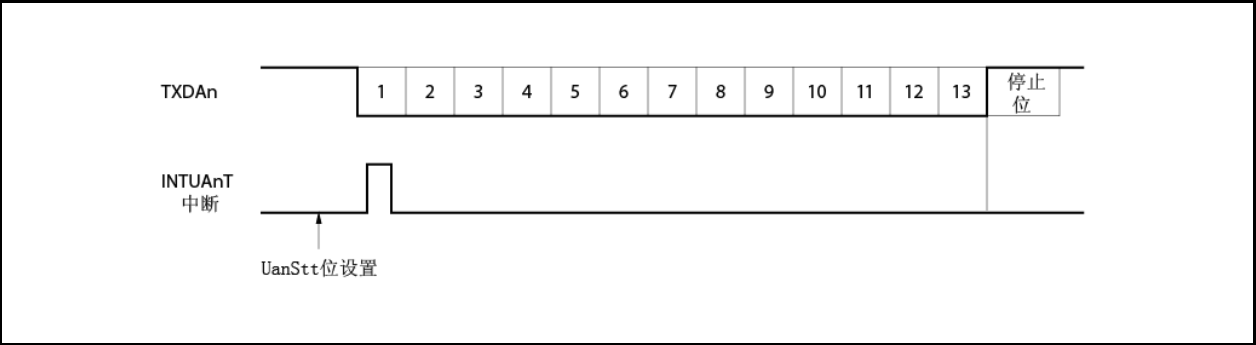
12.5.3 SBF 传输

当 UAnCTL0.UAnPWR 位=UAnCTL0.UAnTXE 位=1 时，进入传输启用状态，且通过设置(为 1)SBF 传输触发(UAnOPT0.UAnSTT 位) 开始 SBF 传输。

此后，输出由 UAnOPT0.UAnSLS2 到 UAnOPT0.UAnSLS0 位指定的位 13 到 20 的低电平宽度。SBF 传输开始时产生传输启用中断请求信号(INTUAnT)。SBF 传输结束之后，UAnSTT 位自动清零。此后，恢复 UART 传输模式。

停止传输，直到接着要传输的数据写入 UAnTX 寄存器或直到设置 SBF 传输触发(UAnSTT 位)。

图 12-5. SBF 传输



### 12.5.4 SBF 接收

通过 UAnCTL0.UAnPWR 位设为 1 且接着 UAnCTL0.UAnRXE 位设为 1 达到接收启用状态。

通过将 BF 接收触发(UAnOPT0.UAnSTR 位)设为 1 来设置 SBF 接收等待状态。

在 SBF 接收等待状态下, 类似于 UART 接收等待状态, 监控 RXDAn 引脚并进行起始位检测。

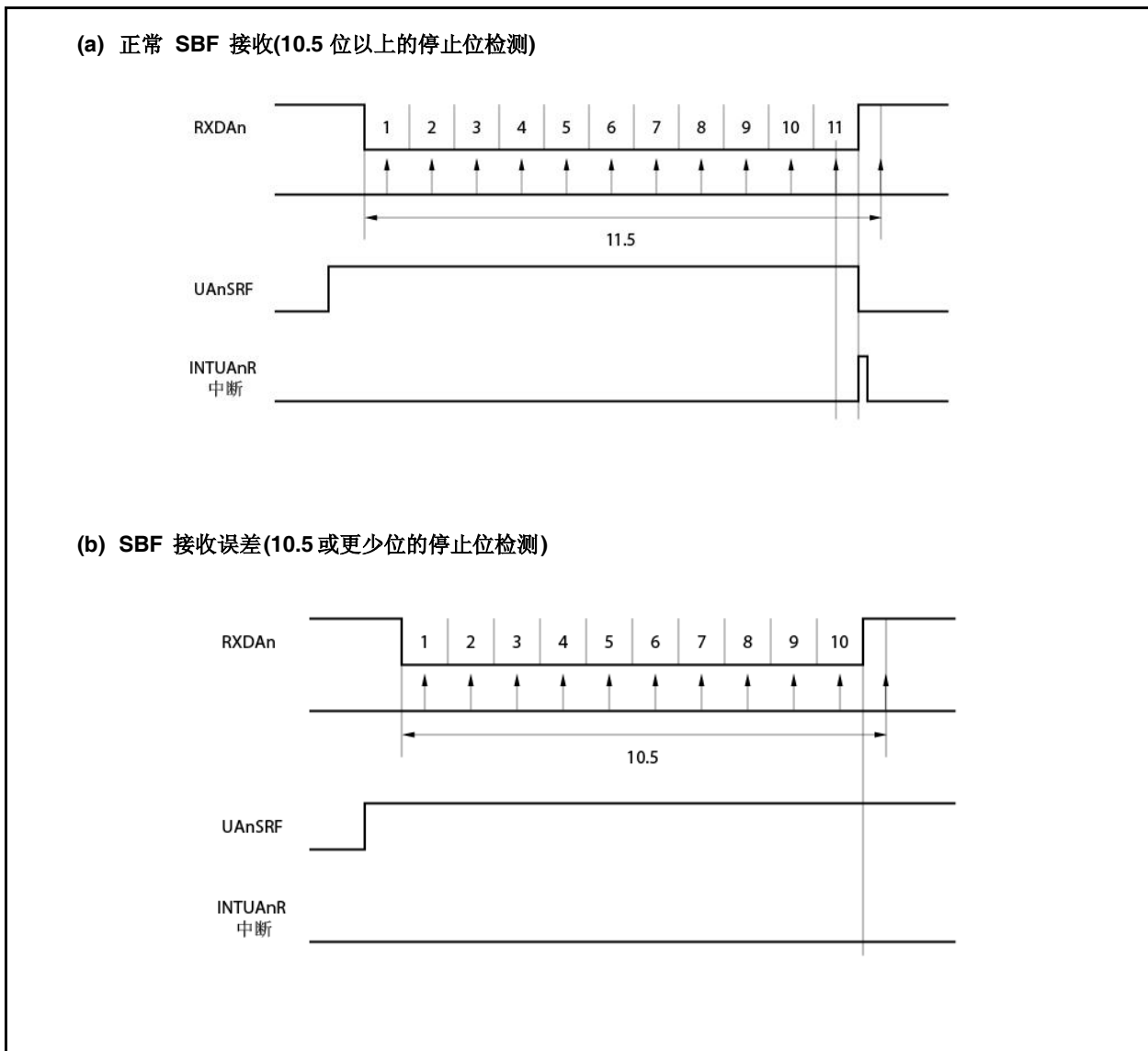
检测起始位以后, 开始接收和内部计数器根据波特率设置计数。

当接收停止位时, 如果 SBF 宽度为 11 或更多位, 判断正常处理且输出接收完成中断请求信号(INTUAnR)。

UAnOPT0.UAnSRF 位自动清零且 SBF 接收结束。抑制 UAnSTR.UAnOVE, UAnSTR.UanPE 以及 UAnSTR.UAnFE 位的误差检测且不执行 UART 通信误差检测处理。此外, 不执行 UARTAn 接收移位寄存器及 UAnRX 寄存器的数据传输且保留 FFH, 初始值。如果 SBF 宽度为 10 或更少的位, 接收中止为误差处理而不输出中断, 且返回 SBF 接收模式。此时 UAnSRF 位不清零。

- <R> 注意事项 1. 如果数据接收期间传输 SBF, 则出现组帧误差。
- <R> 2. 禁止 SBF 接收(UAnSRF = 1)期间 BF 接收触发位(UAnSRT)及 SBF 传输触发位(UAnSTT)设为 1。

图 12-6. SBF 接收



12.5.5 UART 传输

通过 UAnCTL0.UAnPWR 位设为 1 向 TXDAn 引脚输出高电平。

下一次，通过 UAnCTL0.UAnTXE 位设为 1 来设置传输启用状态，且通过将传输数据写入 UAnTX 寄存器开始传输。

自动添加起始位，奇偶校验位及停止位。

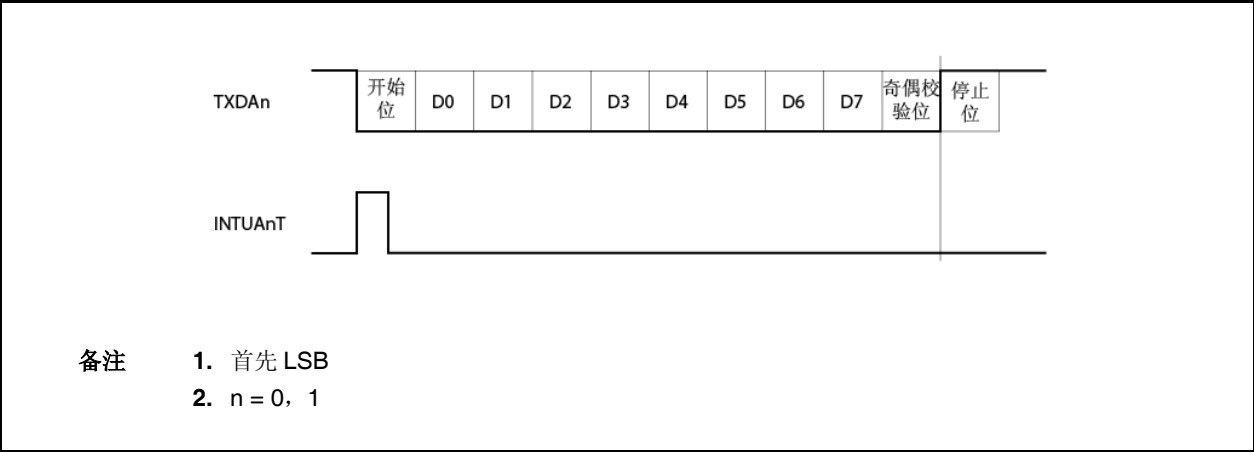
因为 UARTAn 不提供 CTS(传输启用信号)输入引脚，所以使用该端口核对传输目的地是否启用接收。

传输操作开始时 UAnTX 寄存器的数据转移到 UARTAn 传输移位寄存器。

完成 UAnTX 寄存器到 UARTAn 传输移位寄存器的数据传输时产生传输启用中断请求信号(INTUAnT)，且此后 UARTAn 传输移位寄存器的内容输出到 TXDAn 引脚。

产生 INTUAnT 信号之后启用下一传输数据写入 UAnTX 寄存器。

图 12-7. UART 传输





### 12.5.6 连续传输程序

当 UARTAn 传输移位寄存器开始移位操作时 UARTAn 可将下一传输数据写入 UAnTX 寄存器。可从传输启用中断请求信号(INTUAnT)判断 UARTAn 传输移位寄存器的传输时序。

传输期间通过将下一次要传输的数据写入 UAnTX 寄存器实现有效传输率。

连续传输期间，传输数据写入 UAnTX 寄存器且转移到 UARTAn 传输移位寄存器之后，传输请求中断信号(INTUAnT)产生之前严禁将下一次传输数据写入 UAnTX 寄存器。产生传输请求中断信号之前如果值写入 UAnTX 寄存器，则以前的设置传输数据由最近的传输数据覆盖。

**注意事项** 当进行连续传输期间初始化传输时，确保 UAnSTR.UAnTSF 位为 0，接着进行初始化。当不能保证 UAnTSF 位为 1 时传输初始化的数据。

在连续传输情况下，从停止位到下一数据起始位的传输率通过两个操作时钟脉冲自正常比率扩展。

图 12-8. 连续传输处理流程

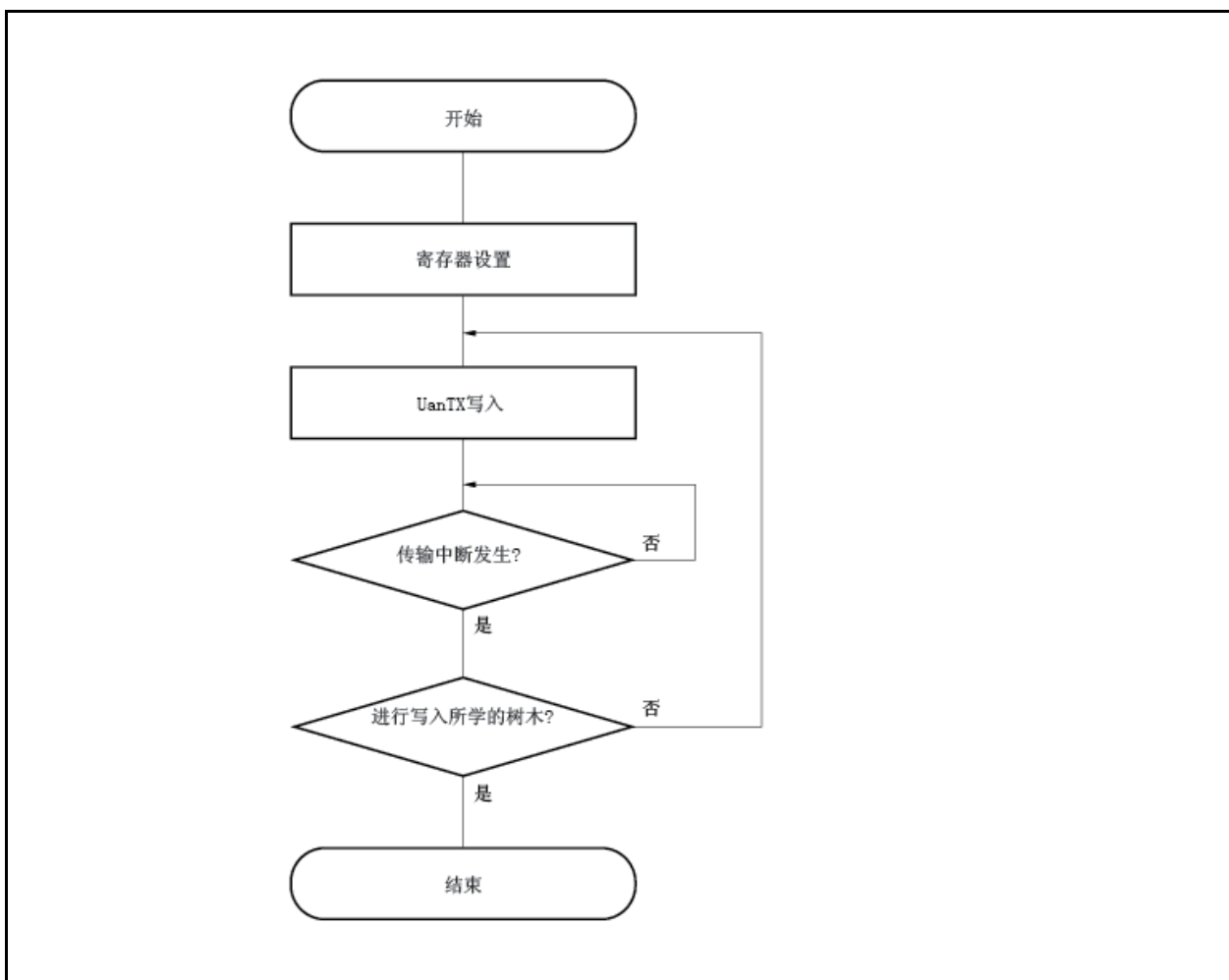
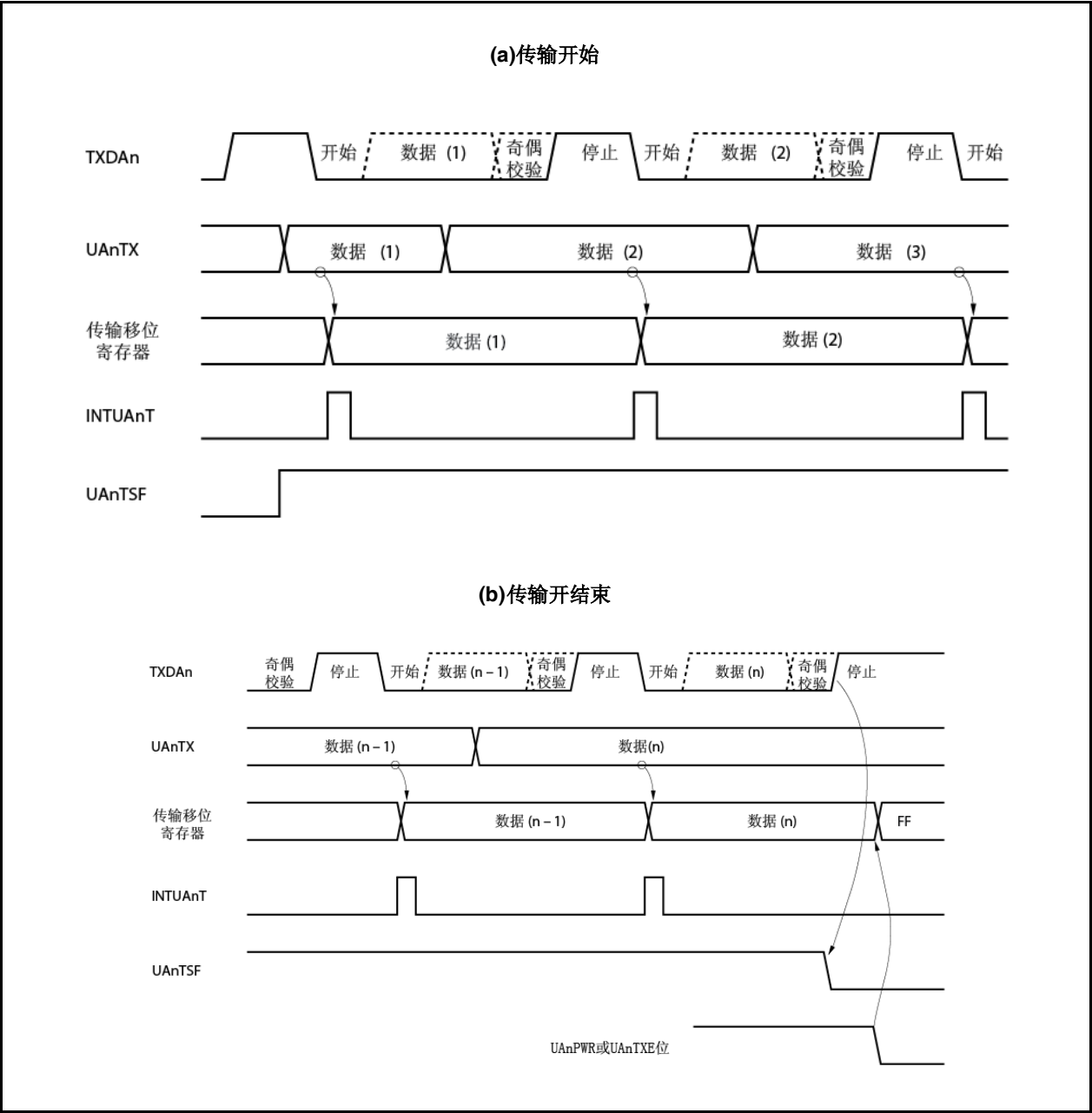


图 12-9. 连续传输操作时序



### 12.5.7 UART 接收

接收等待状态通过 UAnCTL0.UAnPWR 位设为 1 且接着 UAnCTL0.UAnRXE 位设为 1 来设置。在接收等待状态下，监控 RXDAn 引脚且进行起始位检测。

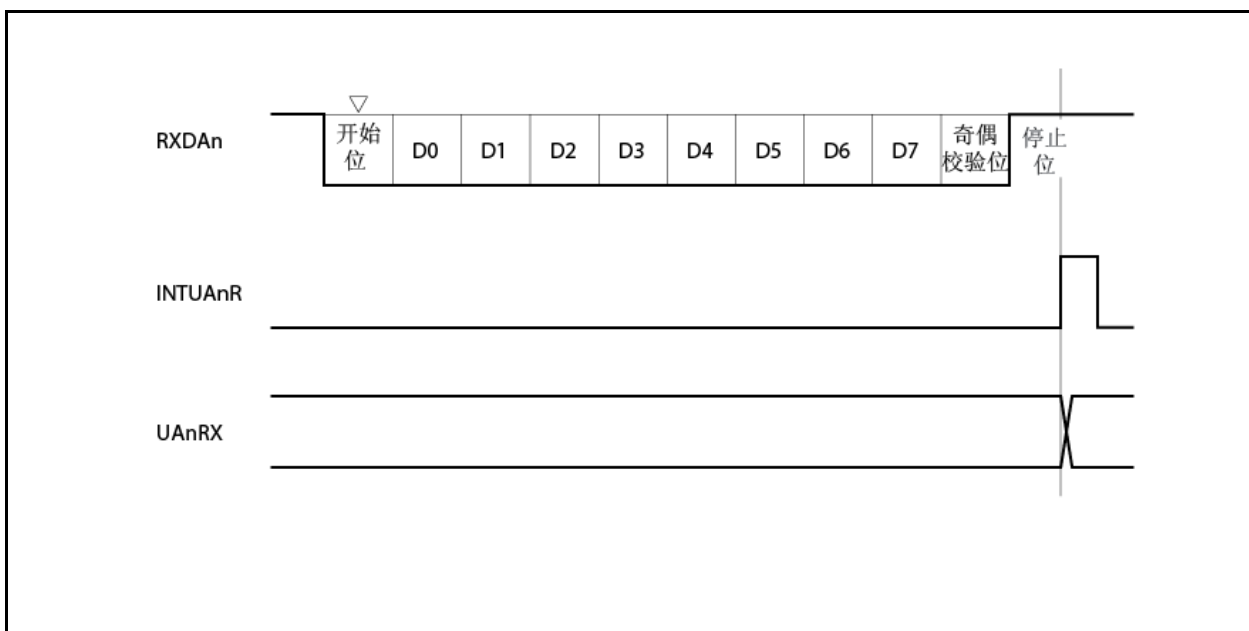
使用两步检测程序进行起始位检测。

首先检测到 RXDAn 引脚的上升沿并在下降沿开始采样。如果在起始位采样点 RXDAn 引脚为低电平则确认起始位。起始位确认之后，接收操作开始，且串行数据根据波特率设置保存到 UARTAn 接收移位寄存器。

当接收到停止位时输出接收完成中断请求信号(INTUAnR)时，UARTAn 接收移位寄存器的数据写入 UAnRX 寄存器。然而，如果出现溢出误差(UAnSTR.UAnOVE 位)，此时接收数据不写入 UAnRX 寄存器，并将其舍弃。

接收期间出现偶数奇偶校验误差(UAnSTR.UAnPE 位)或组帧误差(UAnSTR.UAnFE 位)，继续接收直到第一停止位的接收位置，且接收完成以后输出 INTUAnR。

图 12-10. UART 接收



- 注意事项**
1. 即使当出现接收误差时也要确保读取 UAnRX 寄存器。如果不读取 UAnRX 寄存器，则下一次数据接收期间出现溢出误差，且接收误差连续不确定出现。
  2. 假定仅有一个停止位执行接收期间的操作。忽略第二个停止位。
  3. 当接收完成时，读取 UAnRX 寄存器之后产生接收完成中断请求信号(INTUAnR)，且 UAnPWR 或 UAnRXE 位清零。如果 INTUAnR 信号产生之前 UAnPWR 或 UAnRXE 位清零，则不能保证读取 UAnRX 寄存器的值。
  4. 如果接收 UARTAn 完成处理(INTUAnR 信号产生)且 UAnPWR 位= 0 或 UAnRXE 位= 0 冲突，则可能产生 INTUAnR 信号而不管这些存储在 UAnRX 寄存器的数据。  
要完成接收而不等待 INTUAnR 信号产生，设置(1)中断控制寄存器(UAnRIC)的中断屏蔽标记(UAnRMK)且接着设置(1)UAnPWR 位= 0 或 UAnRXE 位= 0 之后，请确保 UAnRIC 寄存器的中断请求标记(UAnRIF)清零(0)。

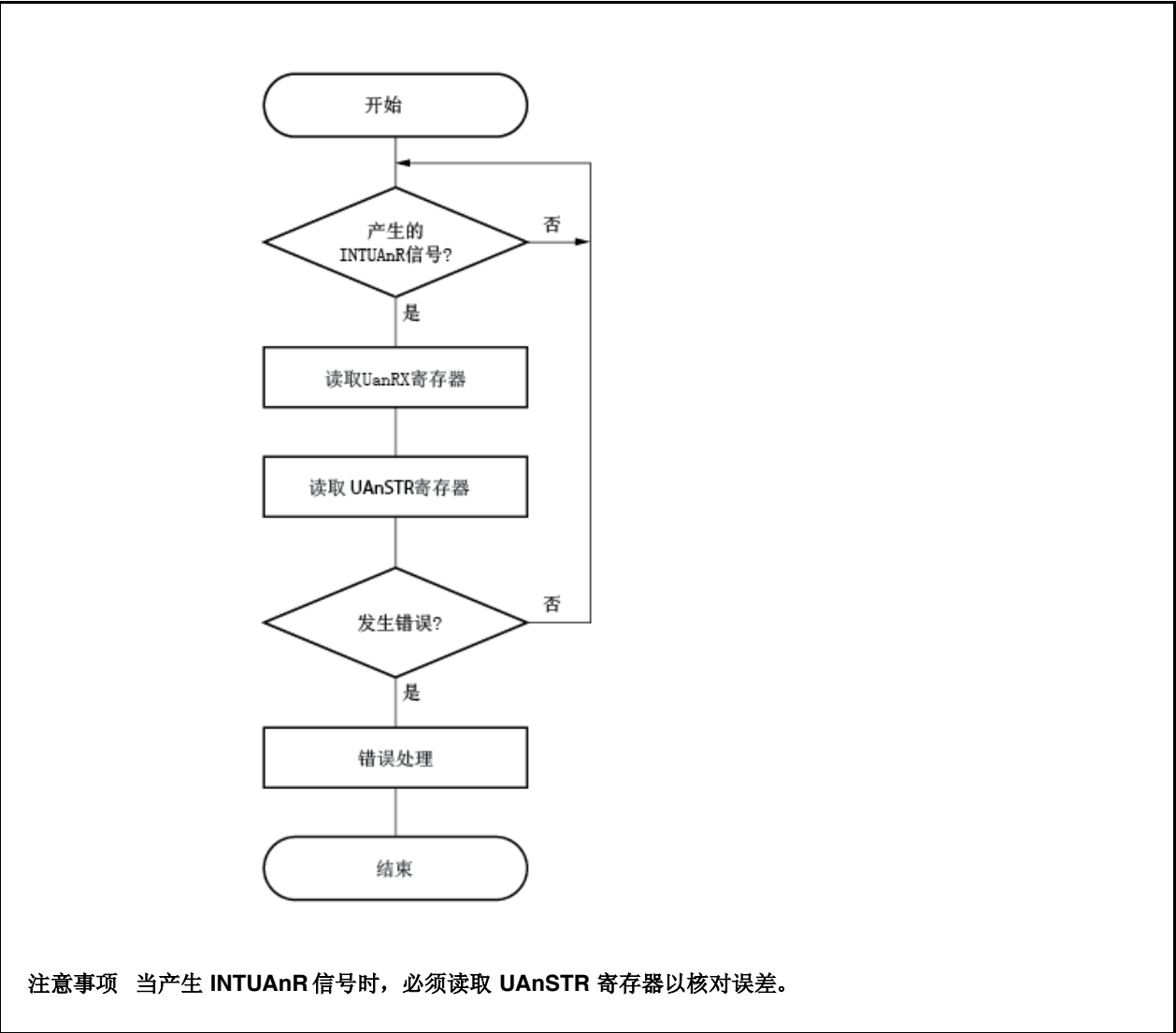
12.5.8 接收误差

接收操作期间的误差有三类：奇偶校验误差，组帧误差以及溢出误差。数据接收结果误差标记设置在 UAnSTR 寄存器中，且当出现误差时输出接收完成中断请求信号(INTUAnR)。

通过读取 UAnSTR 寄存器的内容可能确定接收期间其误差出现。

读取之后通过写入 0 接收误差标记清零。

- 接收数据读取流程



- 接收误差导致

误差标记	接收误差	原因
UAnPE	奇偶校验误差	接收奇偶校验位不匹配该设置
UAnFE	组帧误差	检测不到停止位
UAnOVE	溢出误差	从接收缓冲器读取数据之前完成下一次数据接收

当出现接收误差时，取决于误差种类执行以下程序。

- 奇偶校验误差

如果由于接线端的噪音等问题而接收假数据，则舍弃接收数据并重新传输。

- 组帧误差

波特率误差可能出现在接收端与传输端之间或可能误检测起始位。因为这是通信格式的重大误差，所以传输端核对操作停止，彼此执行初始化处理，且接着开始再次通信。

- 溢出误差

因为读取接收数据之前完成下一次接收，所以舍弃 1 帧数据。如果需要此数据，请重新传输。

**注意事项** 如果连续接收期间发生接收误差中断，则读取 **UAnSTR** 寄存器的内容必须在完成下一次接收之前读取，接着执行误差处理。

12.5.9 奇偶校验类型和操作

**注意事项** 当使用 LIN 功能时，将 UAnCTL0.UAnPS1 和 UAnCTL0.UAnPS0 位固定为 00。

奇偶校验位用于检测通信数据的位误差。通常传输端和接收端使用相同奇偶校验。

在偶数奇偶校验和奇数奇偶校验情况下，可能检测到奇数-计数位误差。在 0 奇偶校验和无奇偶校验情况下，不能检测到误差。

(a) 偶数奇偶校验

(i) 传输期间

控制传输数据中值为“1”的位数，包括奇偶校验位，以便其为偶数。奇偶校验位值如下所示。

- 传输数据中值为“1”的奇数位数: 1
- 传输数据中值为“1”的偶数位数: 0

(ii) 接收期间

对传输数据中值为“1”的位数计数，包括奇偶校验位，且如果其为奇数，则输出奇偶校验误差。

(b) 奇数奇偶校验

(i) 传输期间

控制传输数据中值为“1”的位数，包括奇偶校验位，以便其为奇数。奇偶校验位值如下所示。

- 传输数据中值为“1”的奇数位数: 0
- 传输数据中值为“1”的偶数位数: 1

(ii) 接收期间

对传输数据中值为“1”的位数计数，包括奇偶校验位，且如果其为偶数，则输出奇偶校验误差。

(c) 0 奇偶校验

传输期间，奇偶校验位始终为 0，而不考虑传输数据。

接收期间，不执行奇偶校验位核对。因此，不出现奇偶校验误差，而不考虑奇偶校验位为 0 还是 1。

(d) 无奇偶校验

无奇偶校验位添加到传输数据。

假定进行接收时无奇偶校验位。因为无奇偶校验位所以无奇偶校验误差出现。

12.5.10 接收数据噪音滤波器

此滤波器使用预分频器输出的基本时钟采样 **RXDAn** 引脚。  
当两次读取相同采样值时，匹配监测器输出改变且采样 **RXDAn** 信号作为输入数据。因此，不超过 2 时钟宽度的数据判断为噪音且不传给内部电路(请参阅 图 12-12)。如需基本时钟的信息，请参阅 12.6(1)(a) 基本时钟。  
此外，因为电路如图 12-11 所示，所以接收操作进行的处理相对于外部信号状态延迟 3 个时钟脉冲。

图 12-11. 噪音滤波器电路

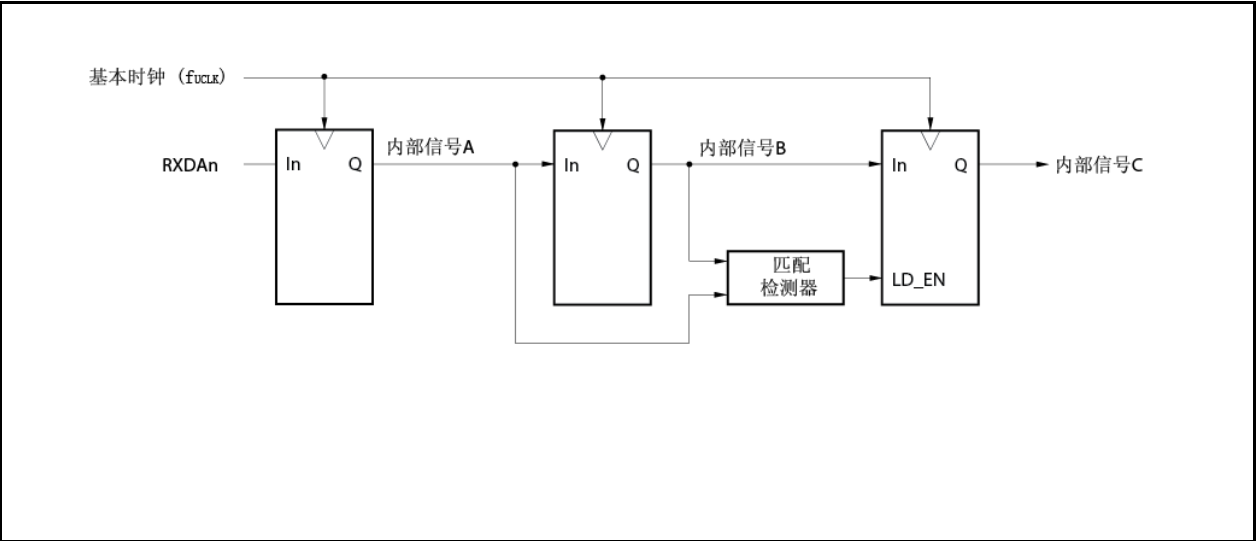
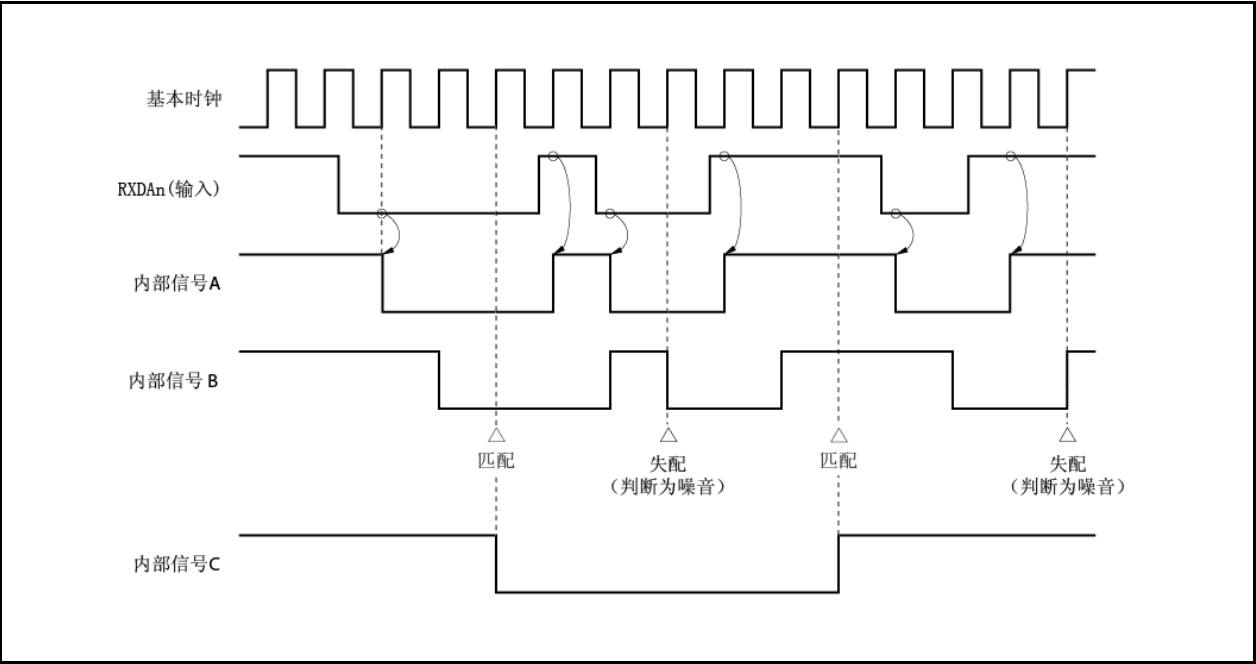


图 12-12. 判断为噪音的 **RXDAn** 信号时序



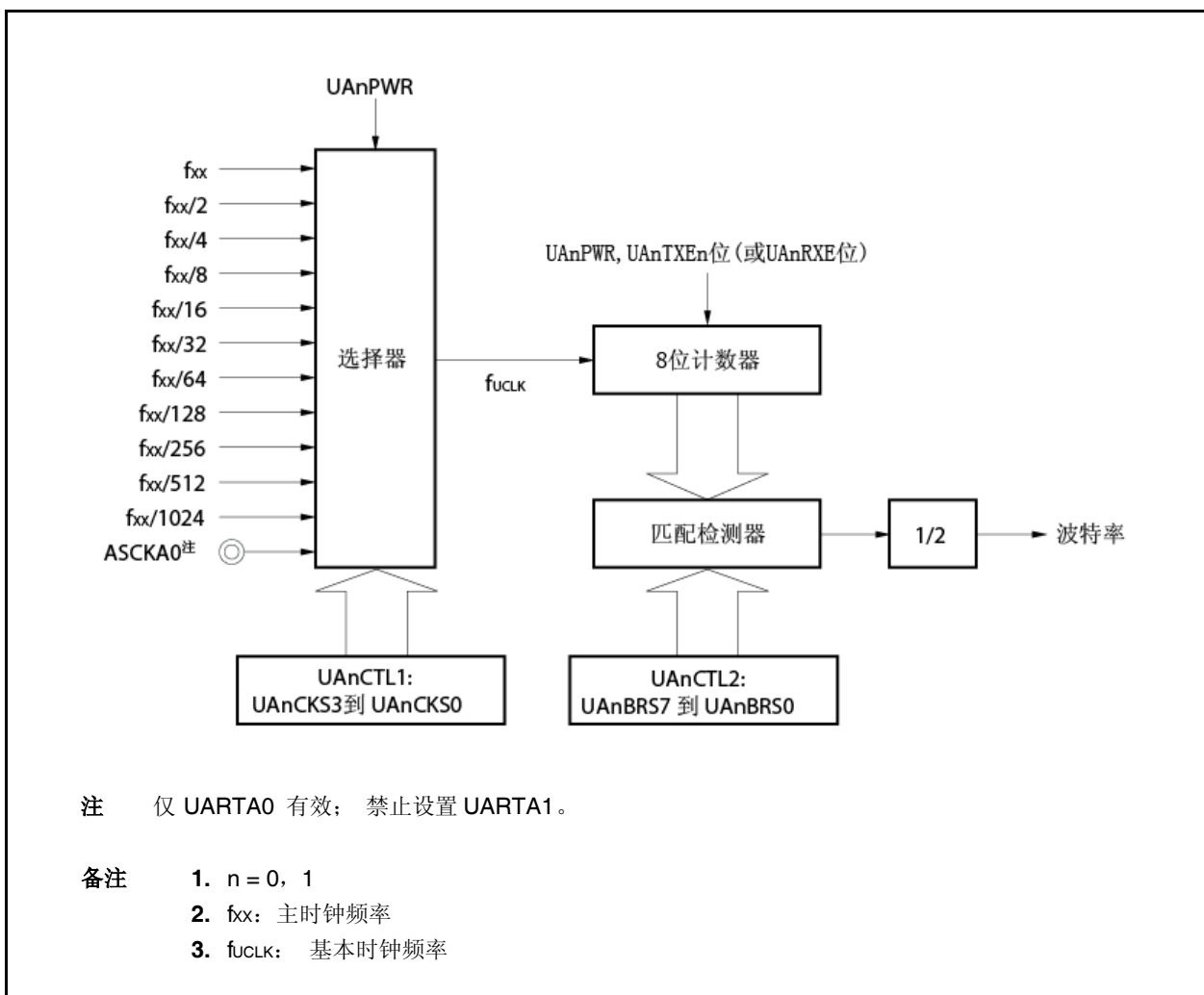
## 12.6 专用波特率发生器

专用波特率发生器包括源时钟选择器块和 8-位可编程计数器，且传输期间产生串行时钟并以 UARTAn 接收。对于串行时钟，可为每个通道选择专用波特率发生器输出。

存在用于传输的 8-位计数器奇用于接收的另一 8-位计数器。

### (1) 波特率发生器配置

图 12-13. 波特率发生器的配置



#### (a) 基本时钟

当 UAnCTL0.UAnPWR 位为 1 时，通过 UAnCTL1.UAnCKS3 到 UAnCTL1.UAnCKS0 位选择的时钟供给 8-位计数器。此时钟称作基本时钟( $f_{UCLK}$ )。

#### (b) 串行时钟产生

串行时钟可通过设置 UAnCTL1 寄存器及 UAnCTL2 寄存器( $n = 0, 1$ ) 产生。

基本时钟通过 UAnCTL1.UAnCKS3 到 UAnCTL1.UAnCKS0 位选择。

8-位计数器的分频值可使用 UAnCTL2.UAnBRS7 到 UAnCTL2.UAnBRS0 位设置。



(2) UARTAn 控制寄存器 1(UAnCTL1)

UAnCTL1 寄存器为选择 UARTAn 基本时钟的 8-位寄存器。  
此寄存器可读取或写入 8-位单元。  
此寄存器重设为 00H。

注意事项 UAnCTL0.UAnPWR 位清零之前重写 UAnCTL1 寄存器。

复位后: 00H      R/W      地址: UA0CTL1 FFFFA01H, UA1CTL1 FFFFA11H

UAnCTL1  
(n = 0, 1)

7	6	5	4	3	2	1	0
0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	基本时钟 (fuc1k) 选择
0	0	0	0	fxx
0	0	0	1	fxx/2
0	0	1	0	fxx/4
0	0	1	1	fxx/8
0	1	0	0	fxx/16
0	1	0	1	fxx/32
0	1	1	0	fxx/64
0	1	1	1	fxx/128
1	0	0	0	fxx/256
1	0	0	1	fxx/512
1	0	1	0	fxx/1,024
1	0	1	1	外部时钟 <sup>注</sup> (ASCKA0 引脚)
以上之外				禁用的设置

注      仅 UARTA0 有效; 禁止设置 UARTA1。

备注      fxx: 主时钟频率

(3) UARTAn 控制寄存器 2(UAnCTL2)

UAnCTL2 寄存器为选择 UARTAn 波特率(串口传输速度)时钟的 8-位寄存器。  
此寄存器可读取或写入 8-位单元。  
此寄存器重设为 FFH。

注意事项 重写 UAnCTL2 寄存器之前 UAnCTL0.UAnPWR 位清零或 UAnTXE 和 UAnRXE 位清零。

复位后：FFH

R/W

地址：UA0CTL2 FFFFFFFA02H, UA1CTL2 FFFFFFFA12H

7	6	5	4	3	2	1	0
UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0

UAnCTL2

(n = 0, 1)

UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0	默认(k)	串行时钟
0	0	0	0	0	0	×	×	×	禁用的设置
0	0	0	0	0	1	0	0	4	f <sub>CLK</sub> /4
0	0	0	0	0	1	0	1	5	f <sub>CLK</sub> /5
0	0	0	0	0	1	1	0	6	f <sub>CLK</sub> /6
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	f <sub>CLK</sub> /252
1	1	1	1	1	1	0	1	253	f <sub>CLK</sub> /253
1	1	1	1	1	1	1	0	254	f <sub>CLK</sub> /254
1	1	1	1	1	1	1	1	255	f <sub>CLK</sub> /255

备注 fUCLK: 时钟频率通过 UAnCTL1.UAnCKS3 到 UAnCTL1.UAnCKS0 位选择

#### (4) 波特率

波特率由以下等式获得。

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

当使用内部时钟时，等式如下所示(当 CKA0 引脚用作 UARTA0 时钟时，使用以上等式计算)。

$$\text{波特率} = \frac{f_{\text{xx}}}{2^{m+1} \times k} \quad [\text{bps}]$$

**备注**  $f_{\text{CLK}}$  = 通过 UAnCTL1.UAnCKS3 到 UAnCTL1.UAnCKS0 位选择的基本时钟频率  
 $f_{\text{xx}}$ : 主时钟频率  
 $m$  = 使用 UAnCTL1.UAnCKS3 到 UAnCTL1.UAnCKS0 位设置的值( $m = 0$  到  $10$ )  
 $k$  = 使用 UAnCTL2.UAnBRS7 到 UAnCTL2.UAnBRS0 位设置的值( $k = 4$  到  $255$ )

波特率误差由以下等式获得。

$$\begin{aligned} \text{误差}(\%) &= \left( \frac{\text{实际波特率(带误差的波特率)}}{\text{目标波特率(校正波特率)}} - 1 \right) \times 100 [\%] \\ &= \left( \frac{f_{\text{CLK}}}{2 \times k \times \text{目标波特率}} - 1 \right) \times 100 [\%] \end{aligned}$$

当使用内部时钟时，等式如下所示(当 CKA0 引脚用作 UARTA0 时钟时，使用以上等式计算波特率误差)。

$$\text{误差}(\%) = \left( \frac{f_{\text{xx}}}{2^{m+1} \times k \times \text{目标波特率}} - 1 \right) \times 100 [\%]$$

- 注意事项**
1. 传输期间的波特率误差必须在接收端的误差公差范围内。
  2. 接收期间的波特率误差必须满足在(5)接收期间可允许波特率范围中指示的范围。

要设置波特率，执行以下计算并设置 UAnCTL1 和 UAnCTL2 寄存器(当使用内部时钟)。

<1> 设置  $k = f_{xx} / (2 \times \text{目标波特率})$ 。设置  $m = 0$ 。

<2> 设置  $k = k/2$  和  $m = m + 1$  其中  $k \geq 256$ 。

<3> 重复 <2>直到  $k < 256$ 。

<4> 四舍五入 k 的第一小数位。

如果通过四舍五入  $k = 256$ ，再次执行 <2> (k 将为 128)。

<5> 将 m 设为 UAnCTL1 寄存器且 k 设为 UAnCTL2 寄存器。

实例： 当  $f_{XX} = 20 \text{ MHz}$  且目标波特率= 153,600 bps 时

<1>  $k = 20,000,000 / (2 \times 153,600) = 65.10\dots$ ，  $m = 0$

<2>， <3>  $k = 65.10\dots < 256$ ，  $m = 0$

<4> UAnCTL2 寄存器的设置值:  $k = 65 = 41\text{H}$ ， UAnCTL1 寄存器的设置值:  $m = 0$

实际波特率  $= 20,000,000 / (2 \times 65)$   
 $= 153,846 \text{ [bps]}$

波特率误差  $= \{20,000,000 / (2 \times 65 \times 153,600) \times 100\}$   
 $= 0.160 \text{ [%]}$

波特率设置的代表实例如下所示。

表 12-3. 波特率发生器设置数据

波特率 (bps)	$f_{XX} = 20 \text{ MHz}$			$f_{XX} = 16 \text{ MHz}$			$f_{XX} = 10 \text{ MHz}$		
	UAnCTL1	UAnCTL2	ERR(%)	UAnCTL1	UAnCTL2	ERR(%)	UAnCTL1	UAnCTL2	ERR(%)
300	08H	82H	0.16	0AH	1AH	0.16	07H	82H	0.16
600	07H	82H	0.16	0AH	0DH	0.16	06H	82H	0.16
1, 200	06H	82H	0.16	09H	0DH	0.16	05H	82H	0.16
2, 400	05H	82H	0.16	08H	0DH	0.16	04H	82H	0.16
4, 800	04H	82H	0.16	07H	0DH	0.16	03H	82H	0.16
9, 600	03H	82H	0.16	06H	0DH	0.16	02H	82H	0.16
19, 200	02H	82H	0.16	05H	0DH	0.16	01H	82H	0.16
31, 250	01H	A0H	0.00	01H	80H	0.00	00H	A0H	0.00
38, 400	01H	82H	0.16	00H	D0H	0.16	00H	82H	0.16
76, 800	00H	82H	0.16	03H	0DH	0.16	00H	41H	0.16
153, 600	00H	41H	0.16	02H	0DH	0.16	00H	21H	(1.36)
312, 500	00H	20H	0.00	00H	1AH	-1.54	00H	10H	0.00

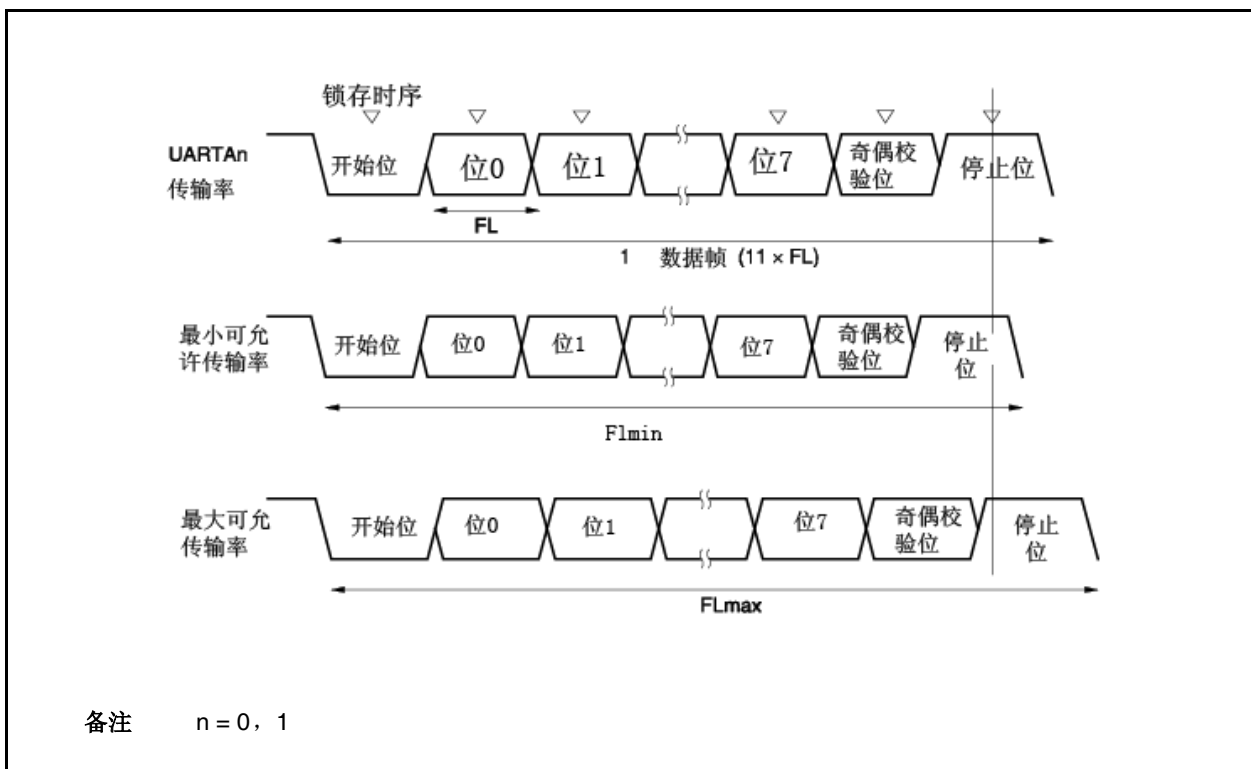
备注  $f_{xx}$ : 主时钟频率  
ERR: 波特率误差(%)

### (5) 接收期间可允许的波特率范围

接收期间可允许的波特的波特率误差范围如下所示。

**注意事项** 接收期间的波特率误差必须使用以下等式设置在可允许误差范围内。

图 12-14. 接收期间可允许的波特率范围



如图 12-14 所示，通过使用开始位检测以后的 UAnCTL2 寄存器设置的计数器确定接收数据锁存器时序。如果可及时接收此锁存器时序的达到最后一位数据(停止位)，则传输数据可正常接收。  
当这应用于 11-位接收时，以下为理论结果。

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

Brate: UARTAn 波特率(n = 0, 1)

k: UAnCTL2.UAnBRS7 到 UAnCTL2.UAnBRS0 位的设置值(n = 0, 1)

FL: 1-位数据长度

锁存器时序边限: 2 时钟

$$\text{最小可允许传输率: } FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

因此，可由目的地接收的最大波特率如下所示。

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

同样地，得到以下最大可允许传输率符合以下等式。

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

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

因此，可由目的地接收的最小波特率如下所示。

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

为了得到最小和最大波特率值从上述等式得到的 UARTAn 和目的地的可允许的波特率误差符合以下规格。

表 12-4. 最大/最小可允许波特率误差

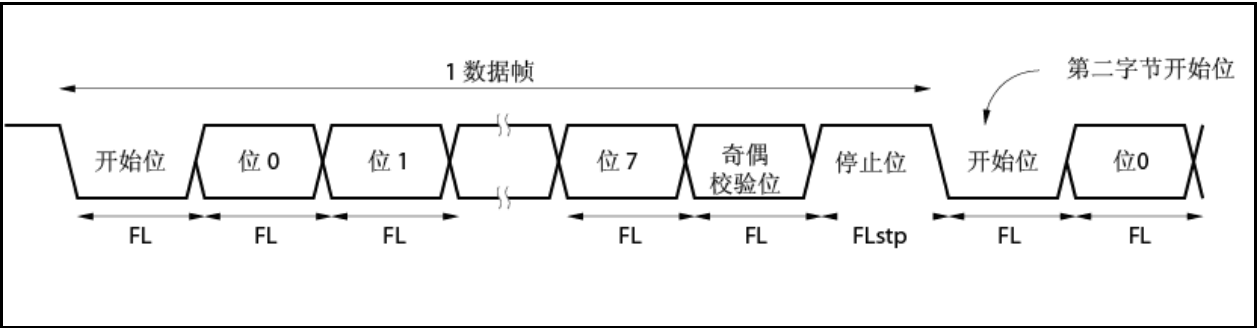
分频比(k)	最大可允许波特率误差	最小可允许波特率误差
4	+2.32%	-2.43%
8	+3.52%	-3.61%
20	+4.26%	-4.30%
50	+4.56%	-4.58%
100	+4.66%	( 4.67%
255	+4.72%	( 4.72%

- 备注
1. 接收精度取决于 1 帧的位计数，输入时钟频率以及分频比(k)。输入时钟频率和较大分频比(k) 高，则精度高。
  2. k: UAnCTL2.UAnBRS7 到 UAnCTL2.UAnBRS0 位的设置值(n = 0, 1)

(6) 波特率连续传输期间

连续传输期间，从停止位到下一起始位的传输率通常长于 2 基本时钟。然而，经由接收端检测的起始位进行时序初始化，因此这不影响传输结果。

图 12-15. 连续传输期间的传输率



假定 1 位数据长度：FL； 停止位长度：FLstp 和基本时钟频率：f<sub>UCLK</sub>，我们得到以下等式。

$$FLstp = FL + 2/f_{UCLK}$$

因此，连续传输期间的传输率如下所示。

$$\text{传输率} = 11 ( FL + (2/f_{UCLK}) )$$

## 12.7 注意事项

- (1) 当供应到 UARTAn 的时钟停止时(例如, 在 IDLE1, IDLE2 或停止模式下), 在时钟供应停止之前立即停止保持值的每一寄存器的操作。TXDAn 引脚输出在时钟供应停止之前还立即保持和输出该值。然而, 时钟供应恢复之后不保证此操作。因此, 时钟供应恢复之后, 电路应通过将 UAnCTL0.UAnPWR, UAnCTL0.UAnRXEn 以及 UAnCTL0.UAnTXEn 位设为 000 初始化。
- (2) RXDA1 和 KR7 引脚必须不同时使用。要使用 RXDA1 引脚, 则禁止使用 KR7 引脚。要使用 KR7 引脚, 则禁止使用 RXDA1 引脚(建议 PFC91 位设为 1 且 PFCE91 位清零)。
- (3) 在 UARTAn 中, 不出现由通信误差导致的中断。当执行接收数据传输时, 即使传输期间出现误差(奇偶校验, 溢出, 组帧)也不能进行误差处理。通信期间读取 UAnSTR 寄存器以核对误差。
- (4) 以下系列中开始 UARTAn。
  - <1> UAnCTL0.UAnPWR 位设为 1。
  - <2> 设置端口。
  - <3> UAnCTL0.UAnTXE 位设为 1, UAnCTL0.UAnRXE 位设为 1。
- (5) 以下系列中停止 UARTAn。
  - <1> 设置 UAnCTL0.UAnTXE 位, UAnCTL0.UAnRXE 位。
  - <2> 设置端口并设置 UAnCTL0.UAnPWR 位(如果端口设置不变, 则没有问题)。
- (6) 在传输模式下(UAnCTL0.UAnPWR 位= 1 和 UAnCTL0.UAnTXE 位= 1), 通过软件禁止 UAnTX 寄存器重新写入相同值, 因为通过写入此寄存器传输开始。要连续传输相同值, 覆盖相同值。
- (7) 在连续传输过程, 从停止位到下一起始位的传输率将通常更多扩展 2 个基本时钟脉冲。然而, 接收端通过检测起始位初始化时序以便不影响接收结果。
- (8) 如果在片上调试(OCD)模式下执行中断命令且如果 UART 接收数据, 则出现溢出误差。



## 第十三章 3 线可变数据长度串行 I/O 接口 (CSIB)

V850ES/HF2 具有两个通道的 3 线串行接口(CSIB)。

### 13.1 特性

- 传输率: 最大 8 Mbps. (在  $f_{xx} = 20 \text{ MHz}$  频率下, 使用内部时钟)
- 主模式和从模式可选
- 8 位至 16 位传输, 3 线串行接口
- 中断请求信号 (INTCBnT, INTCBnR)  $\times 2$
- 可切换串行时钟和数据相位
- 传输数据长度以 1 位为单位, 在 8 至 16 之间可选
- 传输数据 MSB 起始位/LSB 起始位 可选
- 3 线传输
  - SOBn: 串行数据输出
  - SIBn: 串行数据输入
  - $\overline{\text{SCKBn}}$ : 串行时钟 I/O

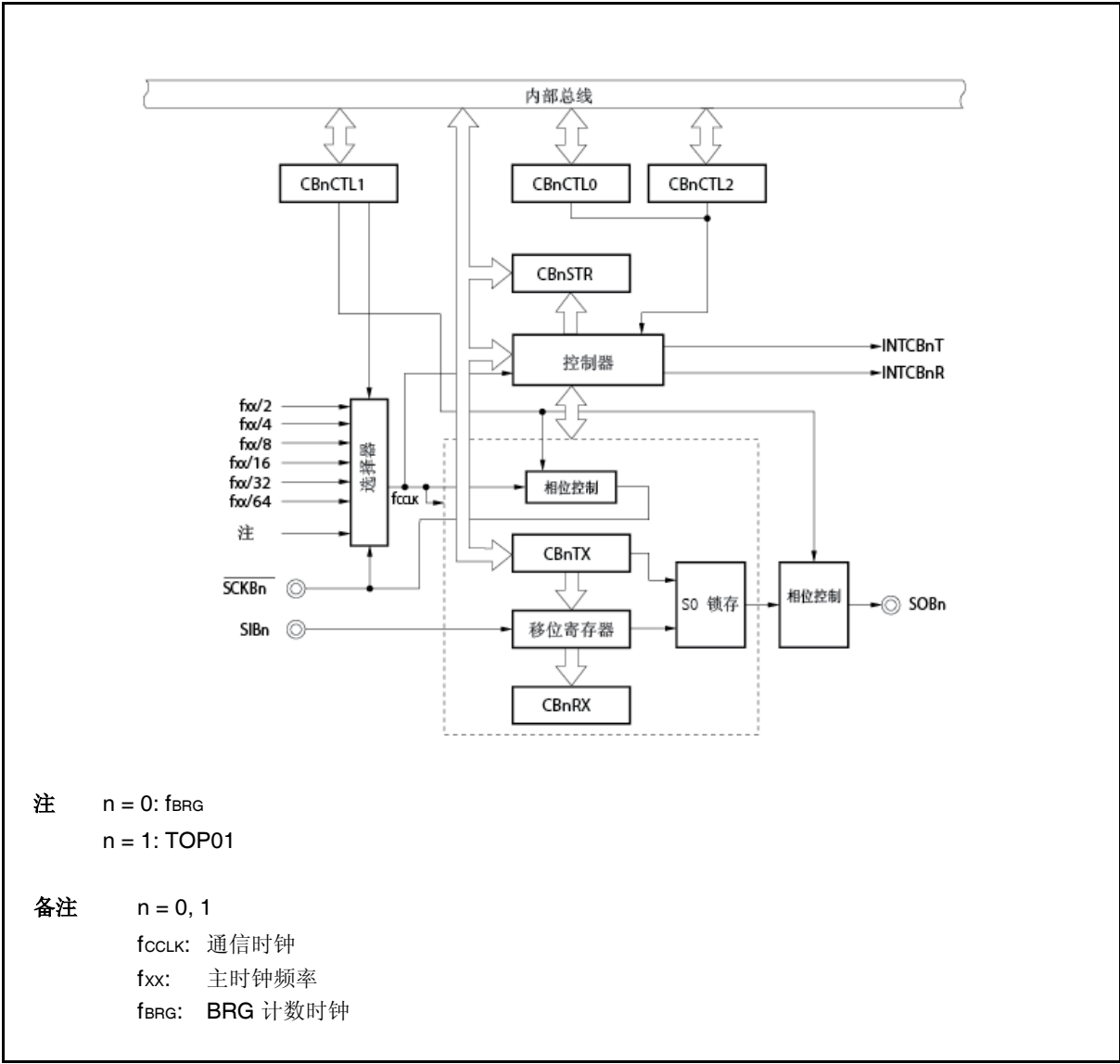
可指定传输模式, 接收模式, 以及传输/接收模式

备注       $n = 0, 1$

13.2 结构

下图显示了 CSIBn 框图。

图 13-1. CSIBn 框图



CSIBn 包括如下一些硬件。

表 13-1. CSIBn 配置

项目	配置
寄存器	CSIBn 接收数据寄存器 (CBnRX) CSIBn 发送数据寄存器 (CBnTX)
控制寄存器	CSIBn 控制寄存器 0 (CBnCTL0) CSIBn 控制寄存器 1 (CBnCTL1) CSIBn 控制寄存器 2 (CBnCTL2) CSIBn 状态寄存器 (CBnSTR)

**(1) CSIBn 接收数据寄存器 (CBnRX)**

CBnRX 寄存器是一个 16 位缓冲寄存器，用以保存接收数据。

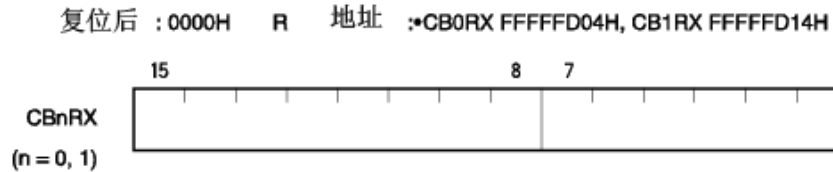
该寄存器为只读寄存器，以 16 位为单位进行操作。

在接收允许状态下，通过读 CBnRX 寄存器而启动数据接收操作过程。

如果所传输的数据长度为 8 位，则该寄存器的低 8 位便作为 CBnRXL 寄存器以 8 位为单位进行只读操作。

复位后，该寄存器被设置为 0000H。

除了复位输入以外，也可以通过将 CBnCTL0 寄存器的 CBnPWR 位清 0 的方式，初始化 CBnRX 寄存器。

**(2) CSIB 发送数据寄存器 (CBnTX)**

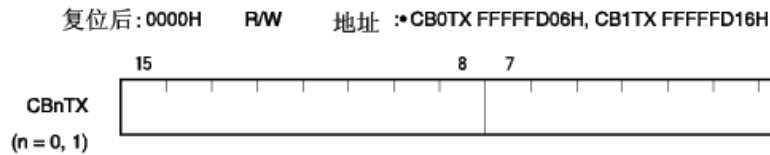
CBnTX 寄存器是一个 16 位缓冲寄存器，用以写入 CSIBn 的传输数据。

该寄存器可以以 16 位为单位进行读或写操作。

在发送允许状态下，通过将数据写入 CBnTX 寄存器而启动数据发送操作过程。

如果所传输的数据长度为 8 位，则该寄存器的低 8 位便作为 CBnTXL 寄存器以 8 位为单位进行只读操作。

复位后，该寄存器被设置为 0000H。



**备注**    通信启动条件如下所示。

发送模式 (CBnTXE 位 = 1, CBnRXE 位 = 0) :

发送/接收模式 (CBnTXE 位 = 1, CBnRXE 位 = 1):

接收模式 (CBnTXE 位 = 0, CBnRXE 位 = 1):

写入 CBnTX 寄存器

写入 CBnTX 寄存器

从 CBnRX 寄存器中读出数据

13.3 寄存器

使用如下一些寄存器以控制 CSIBn 操作。

- CSIBn 控制寄存器 0 (CBnCTL0)
- CSIBn 控制寄存器 1 (CBnCTL1)
- CSIBn 控制寄存器 2 (CBnCTL2)
- CSIBn 状态寄存器 (CBnSTR)

(1) CSIBn 控制寄存器 0 (CBnCTL0)

CBnCTL0 I 寄存器用以控制 CSIBn 串行传输操作过程。

该寄存器可以以 8 位或 1 位为单位进行读或写操作。

复位后，该寄存器被设置为 01H。

(1/3)

复位后: 01H      R/W      地址:      CB0CTL0 FFFFFFFD00H, CB1CTL0 FFFFFFFD10H

	7	6	5	4	3	2	1	0
CBnCTL0	CBnPWR	CBnTXE <sup>注</sup>	CBnRXE <sup>注</sup>	CBnDIR <sup>注</sup>	0	0	CBnTMS <sup>注</sup>	CBnSCE

(n = 0, 1)

CBnPWR	CSIBn操作的禁用/启用设定
0	禁用CSIBn操作并使CBnSTR复位寄存器
1	启用CSIBn操作
• CBnPWR位控制CSIBn操作并复位内部电路	

CBnTXE <sup>注</sup>	发送操作的禁用/启用设定
0	禁止发送操作
1	允许发送操作
• 当CBnTXE位为0时，S0Bn输出低电平信号	

CBnRXE <sup>注</sup>	接收操作的禁用/启用设定
0	禁止发送操作
1	允许发送操作
• 当CBnRXE位清为0时，就不会有接收完成中断输出，即使在为了禁止接受操作而传输了设定数据的情况下，也不会有接收完成中断输出，而且也不会更新接收数据(CBnRX 寄存器)。	

注 当 CBnPWR 位 = 0 时，这些位才能被重写。但是，在对这些位进行重写的同时，也可以对 CBnPWR 进行置位 CBnPWR 位= 1。

<R> 注意事项 要强行挂起发送/接收操作过程，应将 CBnPWR 位清为 0，而不是将 CBnRXE 或 CBnTXE 位清 0。同时，时钟输出停止。

CBnDIR <sup>注</sup>	传输方向模式设定 (MSB/LSB)
0	从MSB位开始传送
1	从LSB位开始传送

CBnTMS <sup>注</sup>	传输模式设定
0	单向传输模式
1	连续传输模式

[在单向传输模式下]

产生接收完成中断请求信号 (INTCBnR)。

即使在传输允许 (CBnTXE位 = 1) 状态下，

也不会产生传输允许中断请求信号 (INTCBnT)。

如果在通信 (CBnSTR. CBnTSF 位 = 1) 过程期间，写入了下一个发送数据，

则该数据会被忽略，而且也不会启动下一次通信。同时，

如果设定了只进行接收的通信模式 (CBnTXE位= 0, CBnRXE位= 1)，

那么在通信过程中，即使读取了接收数据，也不会启动下一次通信。

[在连续传输模式下]

在通信 (CBnSTR. CBnTSF 位 = 1) 过程期间，通过写入下一个发送数据

而启动连续传输模式。在传输允许中断 (INTCBnT) 产生后，便写入下一个传输数据。

如果在连续传输模式下，设定了只进行接收的通信模式 (CBnTXE位= 0, CBnRXE位= 1)，

那么，不管CBnRX寄存器的读操作是怎样的状态，在接收完成中断 (INTCBnR) 产生后，

便启动了下一数据连续接收过程。

因此，会立即从CBnRX寄存器读取接收数据。如果该读操作有延迟，

就会产生overrun错误。

**注** 当 CBnPWR 位 = 0 时，这些位才能被重写。但是，在对这些位进行重写的同时，也可以对 CBnPWR 进行置位 CBnPWR 位= 1。

CBnSCE	启动传输的禁止/允许设定位
0	通信启动触发无效
1	通信启动触发有效
<ul style="list-style-type: none"> <li>在主模式下 该位用来启动或禁止通信启动触发。                             <ul style="list-style-type: none"> <li>(a) 在单向传输或发送/接收传输模式下，或连续发送模式，或连续发送/接收模式下，CBnSCE位的设置情况对通信操作没有影响。</li> <li>(b)在单向接收模式下 在读取最后一个接收数据之前，要将CBnSCE位清为0以禁止启动接收，这是因为在读取接收数据的时候，会启动数据接受。</li> <li>(c)在连续接收模式下 在最后一个数据接收完成之前的一个通信时钟，要将CBnSCE位清为0以禁止在最后一个数据接收之后又启动数据接收。</li> </ul> </li> <li>在从模式下 该位用来启动或禁止通信启动触发。 将CBnSCE位设置为1。</li> </ul> <p>[CBnSCE位的用法]</p> <ul style="list-style-type: none"> <li>在单向接收模式下                             <ul style="list-style-type: none"> <li>&lt;1&gt;当通过INTCBnR中断服务，将最后一个数据接收完成的时候，在读取CBnRX寄存器之前，将CBnSCE位清零。</li> <li>&lt;2&gt;在确认CBnSTR.CBnTSF位= 0之后，将CBnRXE位清零以禁止接收。如果要继续接收数据，可以通过虚读CBnRX寄存器，将CBnSCE位置为1，便可启动下一次接受过程</li> </ul> </li> <li>在连续接收模式下                             <ul style="list-style-type: none"> <li>&lt;1&gt;在通过INTCBnR中断服务，接收最后一个数据期间，将CBnSCE位清零。</li> <li>&lt;2&gt; 度取CBnRX寄存器</li> <li>&lt;3&gt;在中断应答后，通过读CBnRX寄存器，来读取最后一个接收数据。</li> <li>&lt;4&gt;在确认CBnSTR.CBnTSF位= 0之后，将CBnRXE位清零以禁止接收。如果要继续接收数据，可以通过虚读CBnRX寄存器，将CBnSCE位置为1，以等待下一次接受过程</li> </ul> </li> </ul>	

- 注**
1. 如果在 CBnSCE 位为 1 时对该位进行读取，则会启动下一次通信操作。
  2. 如果没有在最后一个数据接收完成的前一个时钟周期将 CBnSCE 位清为 0，则会自动启动下一次通信操作。

**注意事项** 确保要将位 3 和位 2 清至“0”。

## (2) CSIBn 控制寄存器 1 (CBnCTL1)

CBnCTL1 是一个 8 位寄存器，用以控制 CSIBn 串行传输操作过程。

该寄存器可以以 8 位或 1 位为单位进行读或写操作。

复位后，该寄存器被设置为 00H。

&lt;R&gt;

**注意事项** 只有当 CBnCTL0.CBnPWR 位 = 0，或 CBnCTL0.CBnTXE 及 CBnRXE 位 = 0 时，CBnCTL1 寄存器才能被重写。

复位后: 00H R/W 地址: CB0CTL1 FFFFFFFD01H, CB1CTL1 FFFFFFFD11H

	7	6	5	4	3	2	1	0
CBnCTL1	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

(n = 0, 1)

	CBnCKP	CBnDAP	基于SCKBn的数据发送/接收时序的设定
通信类型1	0	0	
通信类型2	0	1	
通信类型3	1	0	
通信类型4	1	1	

CBnCKS2	CBnCKS1	CBnCKS0	通信时钟(f <sub>CCLK</sub> ) <sup>注1</sup>		模式
			n = 0	n = 1	
0	0	0	f <sub>xx</sub> /2		主模式
0	0	1	f <sub>xx</sub> /4		主模式
0	1	0	f <sub>xx</sub> /8		主模式
0	1	1	f <sub>xx</sub> /16		主模式
1	0	0	f <sub>xx</sub> /32		主模式
1	0	1	f <sub>xx</sub> /64		主模式
1	1	0	f <sub>BRG</sub> <sup>注2</sup>	TMPO (TOP01)	主模式
1	1	1	外部时钟(SCKBn)		从模式

&lt;R&gt;

- 注**
1. 设置通信时钟 (f<sub>CCLK</sub>) 为 8 MHz 或更低。
  2. 要了解详细情况，请参阅 13.7 波特率发生器。

(3) CSIBn 控制寄存器 2 (CBnCTL2)

CBnCTL2 是一个 8 位寄存器，用以控制 CSIBn 串行传输数据位数。

该寄存器以 8 位为单位进行读或写操作。

复位后，该寄存器被设置为 00H。

**注意事项** 只有当 CBnCTL0.CBnPWR 位 = 0, 或 CBnCTL0.CBnTXE 及 CBnRXE 位= 0 时，CBnCTL2 寄存器才能被重写。

复位后: 00H    R/W    地址:    CB0CTL2 FFFFFFFD02H. CB1CTL2 FFFFFFFD12H

7	6	5	4	3	2	1	0
0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

(n = 0, 1)

CBnCL3	CBnCL2	CBnCL1	CBnCL0	串行寄存器位长度
0	0	0	0	8 位
0	0	0	1	9 位
0	0	1	0	10 位
0	0	1	1	11 位
0	1	0	0	12 位
0	1	0	1	13 位
0	1	1	0	14 位
0	1	1	1	15 位
1	×	×	×	16 位

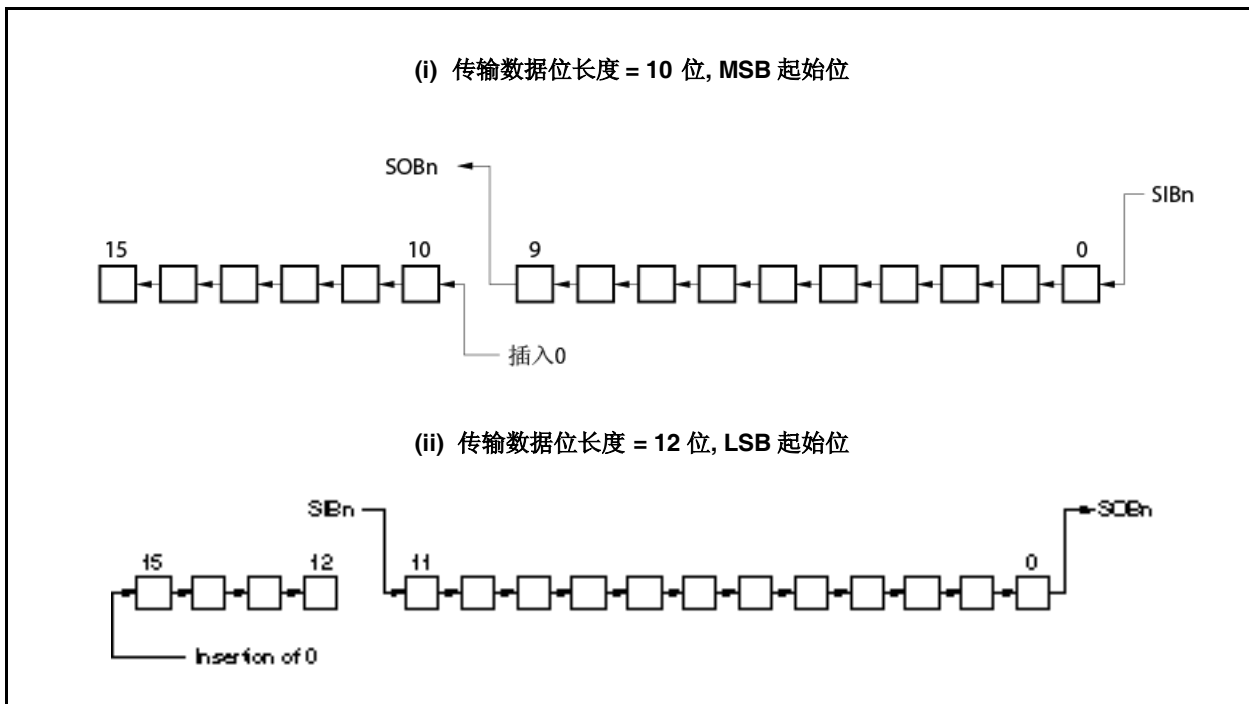
- 备注**
1. 如果传输数据不是 8 位或 16 位,请用 CBnTX 寄存器和 CBnRX 寄存器的 LSB 位进行补位。
  2. ×: 不关心



**(a) 传输数据长度改变功能**

CSIBn 传输的数据长度可以以 1 位为单位进行设定, 通过使用 CBnCTL2 寄存器的 CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 这四位, 在 8 至 16 位设置传输数据长度。

当设置的传输数据长度不是 16 位时, 则不论传输数据的起始位是 MSB 位还是 LSB 位, 都要从 LSB 起将数据设置到 CBnTX 或 CBnRX 寄存器中去。对没有使用的较高位, 可以设置为任何数据, 但是, 在串行传输后接收数据变为 0。



(4) CSIBn 状态寄存器 (CBnSTR)

CBnSTR 是一个 8 位寄存器，用以显示 CSIBn 的状态。  
该寄存器可以以 8 位或 1 位为单位进行读或写操作。但是 CBnTSF 标志位为只读标志位。  
复位后，该寄存器被设置为 00H。  
除了复位输入以外，也可以通过将 CBnCTL0 寄存器的 CBnPWR 位清 0 的方式，初始化 CBnSTR 寄存器。

复位后: 00H      R/W      地址:      CB0STR FFFFFFFD03H, CB1STR FFFFFFFD13H

	7	6	5	4	3	2	1	0
CBnSTR	CBnTSF	0	0	0	0	0	0	CBnOVE

(n = 0, 1)

CBnTSF	通信状态标志位
0	通信已停止
1	通信进行中
<ul style="list-style-type: none"><li>在传输过程中，如果数据储备在CBnTX寄存器内，则对该寄存器进行设置，在接收过程中，CBnTX寄存器发生虚读时进行设置。</li><li>传输结束后，该标志在计时器终端被清零。</li></ul>	

CBnOVE	Overrun错误标志位
0	无overrun错误
1	Overrun
<ul style="list-style-type: none"><li>在接收操作完成时，当下一数据接收结束而CPU没有读取接收缓冲器中数据的时候，便会产生overrun错误。</li><li>在这种情况下，CBnOVE标志位显示overrun错误发生状态标志</li><li>在单向传输模式下，位也是有效的。因此，在只使用发送操作时，要注意以下几点</li><li>不检测CBnOVE标志位</li><li>即使不要求读取接收数据，也要读取该位的值。</li><li>通过向该位写入数据0而将该标志位清零，但是，即使向该标志位写入数据1，也不能将其置位。</li></ul>	

13.4 中断请求信号

CSIBn 可以产生以下两种类型的中断请求信号。

- 接收完成中断请求信号 (INTCBnR)
- 发送允许中断请求信号 (INTCBnT)

在这两个中断请求信号中，默认情况下，接收完成中断请求信号的中断优先级较高，而发送允许中断请求信号的中断优先级较低。

表 13-2. 中断请求信号及其默认优先级

中断类型	优先级
接收完成中断	高
发送允许中	低

(1) 接收完成中断请求信号 (INTCBnR)

在接收允许状态下，当接收据传送到 CBnRX 寄存器的时候，便会触发产生接收完成中断请求信号。  
如果有溢出覆盖（overrun）错误发生，也会产生这种中断请求信号。  
当 CPU 响应接收完成中断请求信号并读取数据时，要读取 CBnSTR 寄存器值以确认接收结果无误。  
在单向传输模式下，即使仅仅执行发送，当发送完成时，会产生 INTCBnR 中断请求信号。

(2) 发送允许中断请求信号 (INTCBnT)

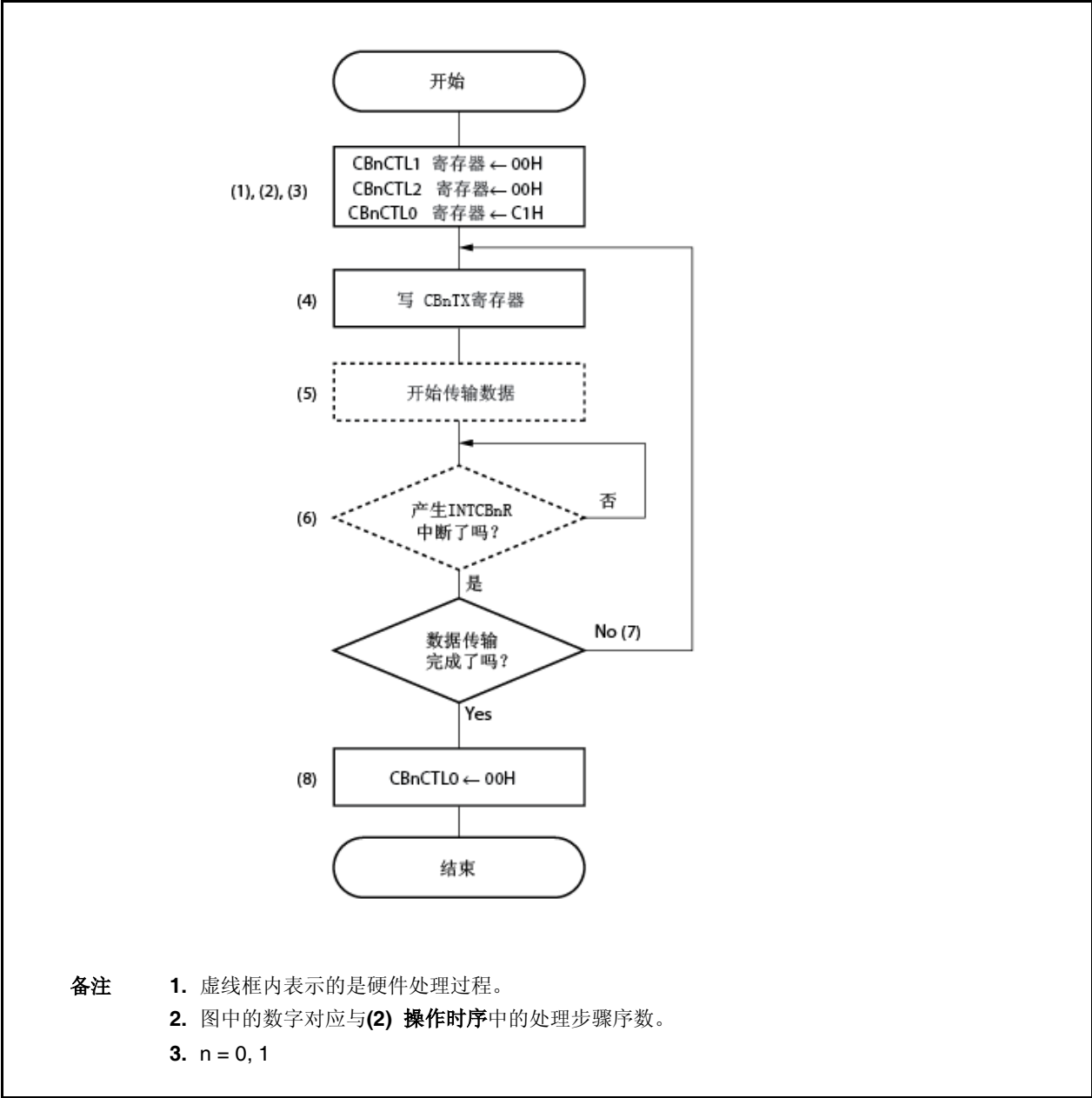
在连续发送模式或连续发送/接收模式下，从 CBnTX 寄存器传输发送数据，而且一旦 CBnTX 寄存器写操作允许，就会产生发送允许中断请求信号。  
在单向发送模式及在单向发送/接收模式下，不会产生 INTCBnT 中断信号。

<R> 13.5 操作

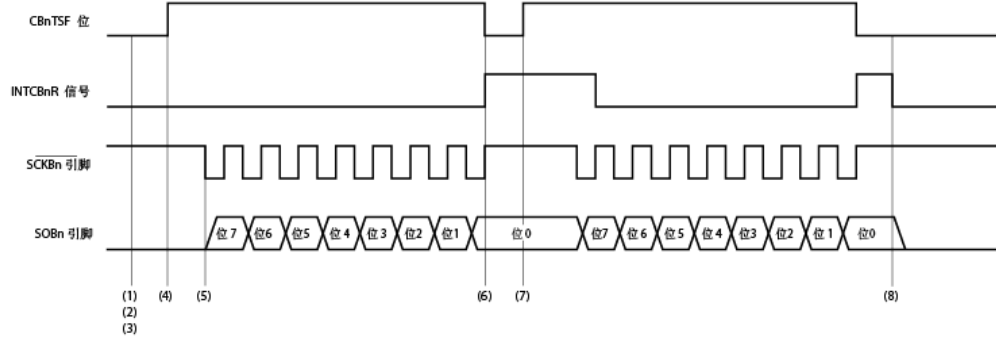
13.5.1 单向传输模式 (主模式，发送模式)

MSB 起始位 (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位 = 00), 通信时钟 ( $f_{\text{CCLK}}$ ) =  $f_x/2$  (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000), 传输数据位长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



## (2) 操作时序



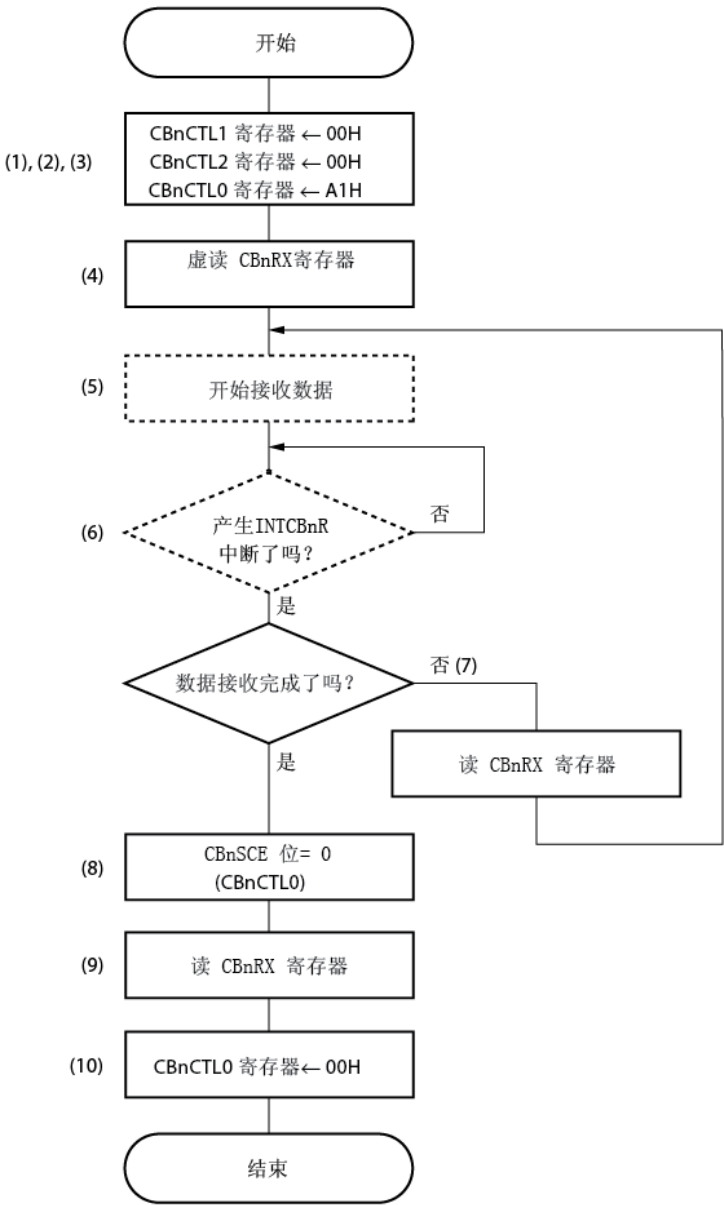
- (1) 将 00H 写入 CBnCTL1 寄存器, 并选择通信类型 1, 通信时钟( $f_{CCLK} = f_{XX}/2$ ), 以及主模式。
- (2) 将 00H 写入 CBnCTL2 寄存器, 并且设置传输数据长度为 8 位。
- (3) 将 C1H 写入 CBnCTL0 寄存器, 并且在通信时钟( $f_{CCLK}$ )操作允许设定的同时, 选择 MSB 作为数据传输的起始位。
- (4) 通过将发送数据写入 CBnTX 寄存器将 CBnSTR.CBnTSF 位设置为 1, 启动发送操作过程。
- (5) 当数据发送启动后, 向 SCKBn 引脚输出串行时钟信号, 同时使用串行时钟从 SOBn 引脚同步输出发送数据。
- (6) 当将由 CBnCTL2 寄存器所设定的一定长度传输数据发送完毕的时候, 便停止串行时钟输出和发送数据输出, 并在串行时钟的最后一个边沿产生接收完成中断请求信号(INTCBnR), 同时将 CBnTSF 位清 0。
- (7) 如果要继续进行数据发送, 请在 INTCBnR 信号产生后, 再次通过将发送数据写入 CBnTX 寄存器而启动下一次发送过程。
- (8) 要结束数据发送, 请写入 CBnCTL0.CBnPWR 位 = 0 以及 CBnCTL0.CBnTXE 位 = 0 即可。

备注      n = 0, 1

13.5.2 单向传输模式 (主模式, 接收模式)

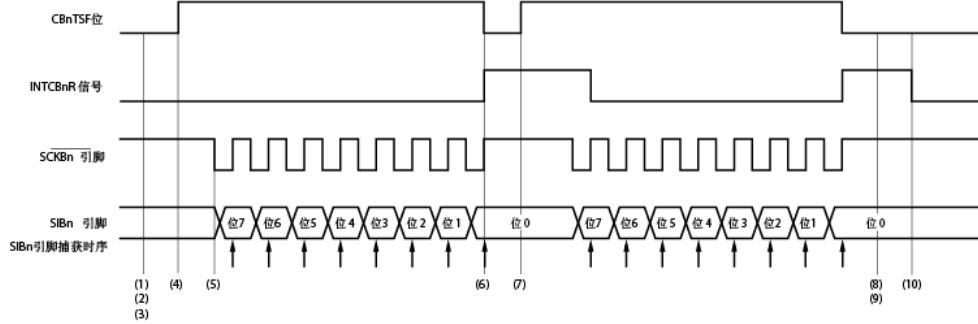
MSB 起始位 (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位 = 00), 通信时钟 ( $f_{\text{CCLK}}$ ) =  $f_x/2$  (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000), 传输数据位长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



- 备注
- 1. 虚线框内表示的是硬件处理过程。
  - 2. 图中的数字对应与(2) 操作时序中的处理步骤序号。
  - 3.  $n = 0, 1$

## (2) 操作时序



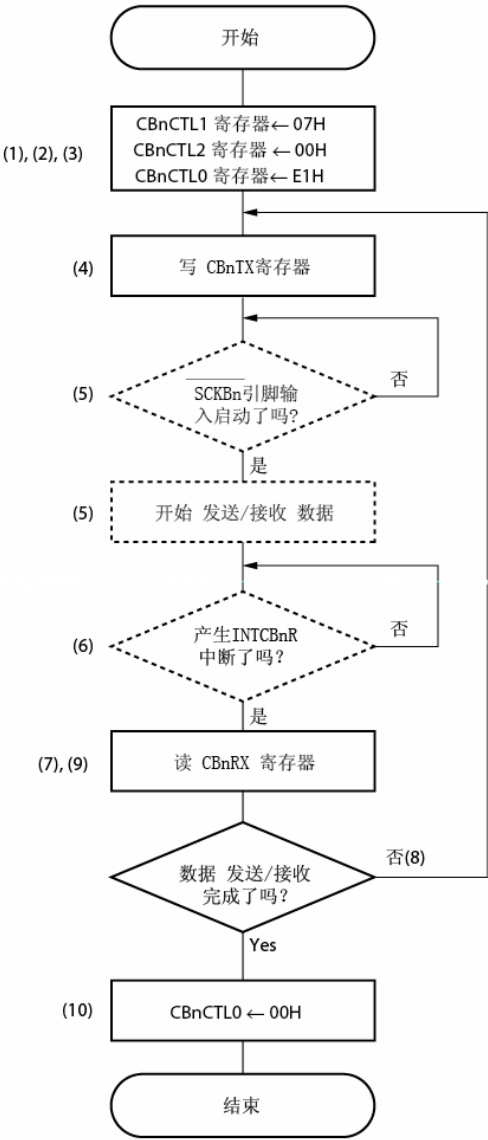
- (1) 将 00H 写入 CBnCTL1 寄存器, 并选择通信类型 1, 通信时钟( $f_{CCLK}$ ) =  $f_{XX}/2$ , 以及主模式。
- (2) 将 00H 写入 CBnCTL2 寄存器, 并且设置传输数据长度为 8 位。
- (3) 将 A1H 写入 CBnCTL0 寄存器, 选择接收模式, 并且在通信时钟( $f_{CCLK}$ )操作允许设定的同时, 选择 MSB 作为数据传输的起始位。
- (4) 通过对 CBnRX 寄存器执行一个虚读操作, 将 CBnSTR.CBnTSF 位设置为 1, 启动数据接收过程。
- (5) 当数据接收启动后, 向 SCKBn 引脚输出串行时钟信号, 同时使用串行时钟同步捕获 SIBn 引脚的接收数据。
- (6) 当将由 CBnCTL2 寄存器所设定的一定长度传输数据接收完毕的时候, 便停止串行时钟输出和数据捕获, 并在串行时钟的最后一个边沿产生接收完成中断请求信号(INTCBnR), 同时将 CBnTSF 位清 0。
- (7) 如果要继续进行数据接收, 请在 INTCBnR 信号产生后, 读取 CBnCTL0.CBnSCE 位保持为 1 的 CBnRX 寄存器值, 从而启动下一次接收过程。
- (8) 设置 CBnSCE 位为 0, 以读取没有启动下一次接收操作的 CBnRX 寄存器值。
- (9) 读取 CBnRX 寄存器值。
- (10) 要结束数据接收, 请写入 CBnCTL0.CBnPWR 位 = 0 以及 CBnCTL0.CBnRXE 位 = 0 即可。

备注       $n = 0, 1$

13.5.3 单向传输模式 (主模式，发送/接收模式)

MSB 起始位 (CBnCTL0.CBnDIR 位= 0), 通信类型 1 (CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位 = 00), 通信时钟 ( $f_{\text{CLK}}$ ) =  $f_{\text{xx}}/2$  (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000), 传输数据位长度= 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

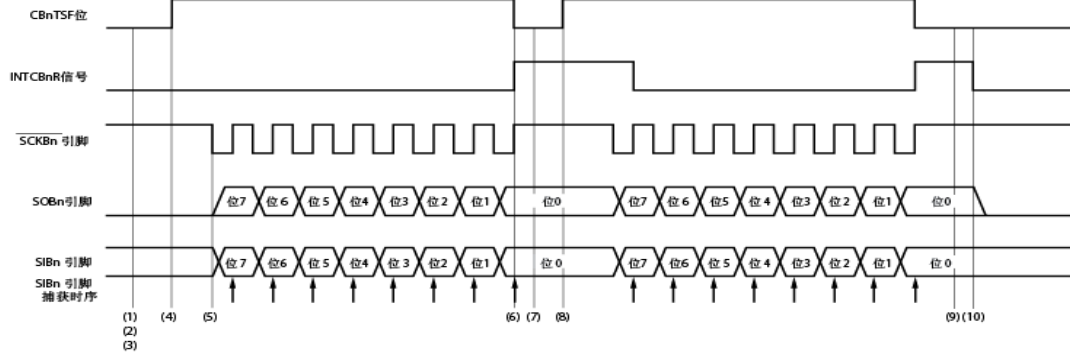
(1) 操作流程



- 备注
- 1. 虚线框内表示的是硬件处理过程。
  - 2. 图中的数字对应与(2) 操作时序中的处理步骤序数。
  - 3. n = 0, 1



## (2) 操作时序



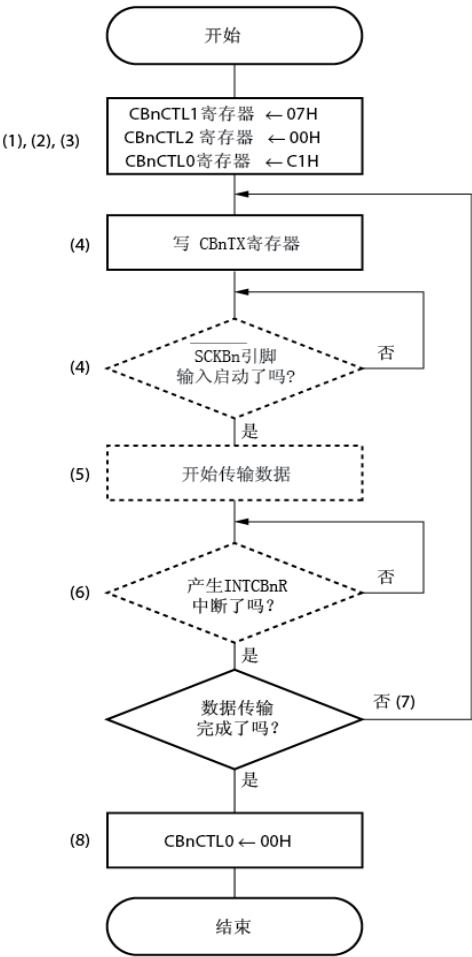
- (1) 将 00H 写入 CBnCTL1 寄存器, 并选择通信类型 1, 通信时钟( $f_{CCLK}$ ) =  $f_{XX}/2$ , 以及主模式。
- (2) 将 00H 写入 CBnCTL2 寄存器, 并且设置传输数据长度为 8 位。
- (3) 将 E1H 写入 CBnCTL0 寄存器, 选择发送/接收模式, 并且在通信时钟 ( $f_{CCLK}$ ) 操作允许设定的同时, 选择 MSB 作为数据传输的起始位。
- (4) 通过将发送数据写入 CBnTX 寄存器将 CBnSTR.CBnTSF 位设置为 1, 启动发送/接收操作过程。
- (5) 当数据发送/接收启动后, 向 SCKBn 引脚输出串行时钟信号, 使用串行时钟向 SOBn 引脚同步输出发送数据, 同时捕获 SIBn 引脚的接收数据。
- (6) 当将由 CBnCTL2 寄存器所设定的一定长度传输数据发送/接收完毕的时候, 便停止串行时钟输出, 发送数据输出和数据捕获, 并在串行时钟的最后一个边沿产生接收完成中断请求信号 (INTCBnR), 同时将 CBnTSF 位清 0。
- (7) 读取 CBnRX 寄存器值。
- (8) 如果要继续进行数据发送/接收, 请再次向 CBnTX 寄存器写入发送数据。
- (9) 读取 CBnRX 寄存器值。
- (10) 要结束数据发送/接收, 请写入 CBnCTL0.CBnPWR 位 = 0, CBnCTL0.CBnTXE = 0 以及 CBnCTL0.CBnRXE 位 = 0 即可。

备注       $n = 0, 1$

13.5.4 单向传输模式 (从模式，发送模式)

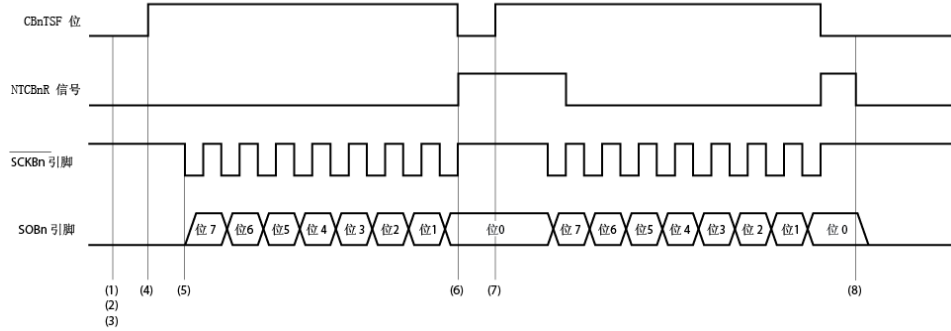
MSB 起始位 (CBnCTL0.CBnDIR 位= 0), 通信类型 1 (CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f<sub>CCLK</sub>) = 外部时钟 ( $\overline{\text{SCKBn}}$ ) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111), 传输数据位长度= 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



- 备注
- 1. 虚线框内表示的是硬件处理过程。
  - 2. 图中的数字对应与(2) 操作时序中的处理步骤序号。
  - 3. n = 0, 1

## (2) 操作时序



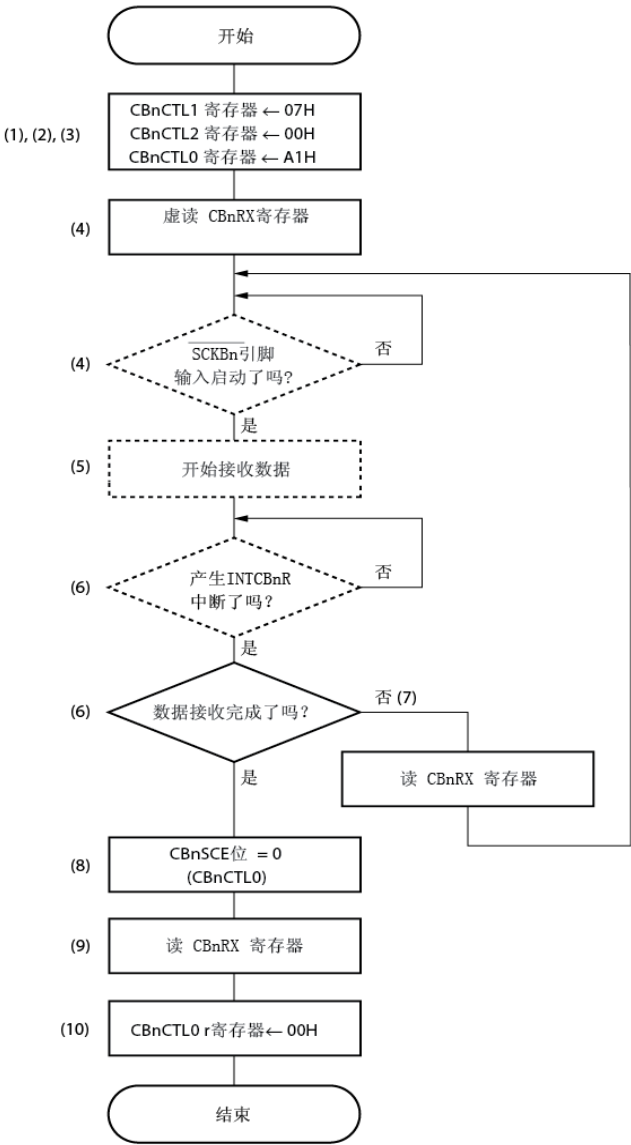
- (1) 将 07H 写入 **CBnCTL1** 寄存器, 并选择通信类型 1, 通信时钟 ( $f_{\text{CLK}}$ ) = 外部时钟 (SCKBn), 以及从模式。
- (2) 将 00H 写入 **CBnCTL2** 寄存器, 并且设置传输数据长度为 8 位。
- (3) 将 C1H 写入 **CBnCTL0** 寄存器, 选择发送模式, 并且在通信时钟( $f_{\text{CLK}}$ )操作允许设定的同时, 选择 MSB 作为数据传输的起始位。
- (4) 通过将发送数据写入 **CBnTX** 寄存器将 **CBnSTR.CBnTSF** 位设置为 1, 从而使得设备处于等待串行时钟信号输入状态。
- (5) 当有串行时钟信号输入时, 使用串行时钟, 便从 **SOBn** 引脚同步输出发送数据。
- (6) 当将由 **CBnCTL2** 寄存器所设定的一定长度传输数据发送完毕的时候, 便停止串行时钟输出和发送数据输出, 并在串行时钟的最后一个边沿产生接收完成中断请求信号(**INTCBnR**), 同时将 **CBnTSF** 位清 0。
- (7) 如果要继续进行数据发送, 请在 **INTCBnR** 信号产生后, 再次向 **CBnTX** 寄存器写入发送数据, 等待串行时钟输入。
- (8) 要结束数据发送, 请写入 **CBnCTL0.CBnPWR** 位 = 0 以及 **CBnCTL0.CBnTXE** 位 = 0 即可。

备注       $n = 0, 1$

13.5.5 单向传输模式 (从模式, 接收模式)

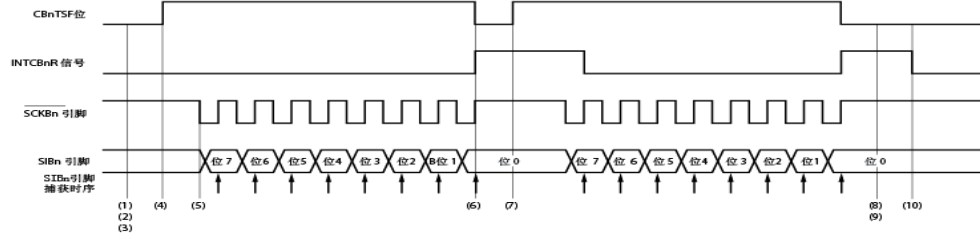
MSB 起始位 (CBnCTL0.CBnDIR 位= 0), 通信类型 1 (CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f<sub>CLK</sub>) = 外部时钟 ( $\overline{\text{SCKBn}}$ ) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111), 传输数据位长度= 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



- 备注
- 1. 虚线框内表示的是硬件处理过程。
  - 2. 图中的数字对应与**(2) 操作时序**中的处理步骤序数。
  - 3. n = 0, 1

## (2) 操作时序



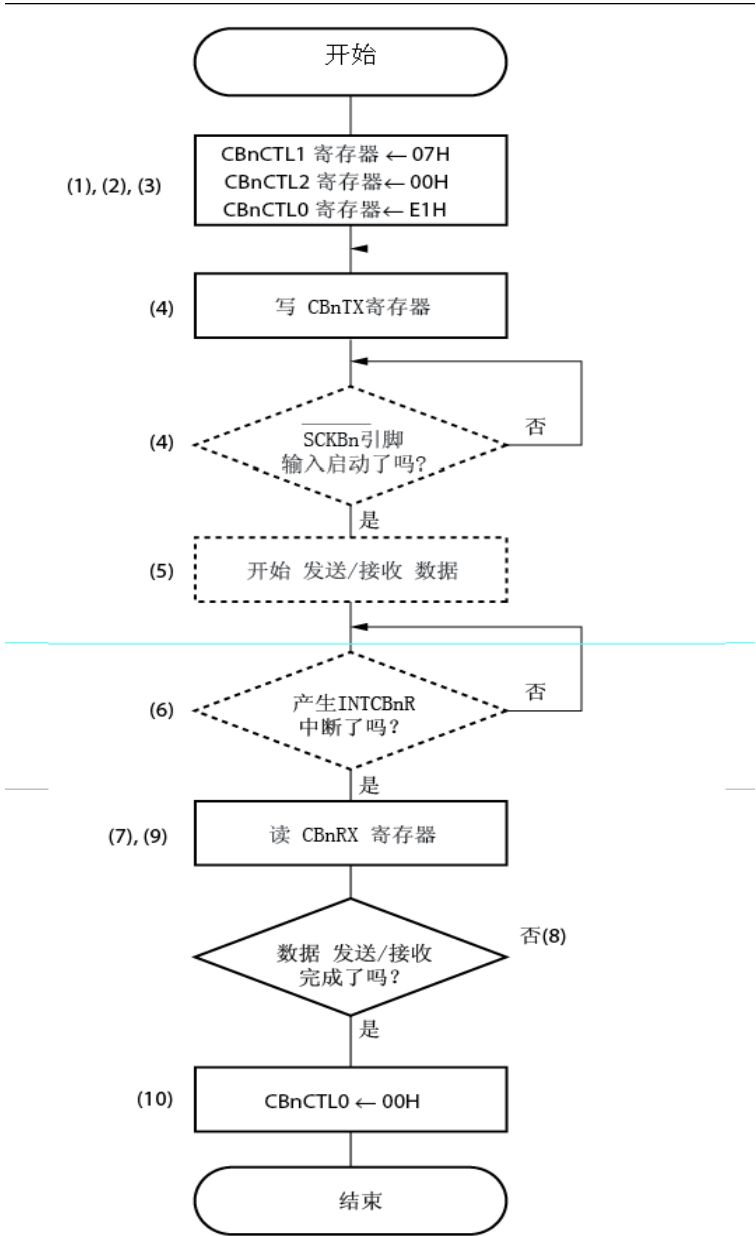
- (1) 将 07H 写入 **CBnCTL1** 寄存器, 并选择通信类型 1, 通信时钟 ( $f_{\text{CCLK}}$ ) = 外部时钟 (**SCKBn**), 以及从模式。
- (2) 将 00H 写入 **CBnCTL2** 寄存器, 并且设置传输数据长度为 8 位。
- (3) 将 A1H 写入 **CBnCTL0** 寄存器, 选择接收模式, 并且在通信时钟操作允许设定的同时, 选择 **MSB** 作为数据传输的起始位。( $f_{\text{CCLK}}$ )。
- (4) 通过对 **CBnRX** 寄存器执行一个虚读操作, 将 **CBnSTR.CBnTSF** 位设置为 1, 从而使得设备处于等待串行时钟信号输入状态。
- (5) 当有串行时钟信号输入时, 使用串行时钟同步捕获 **SIBn** 引脚的接收数据。
- (6) 当将由 **CBnCTL2** 寄存器所设定的一定长度传输数据接收完毕的时候, 便停止串行时钟输出和数据捕获, 并在串行时钟的最后一个边沿产生接收完成中断请求信号(**INTCBnR**), 同时将 **CBnTSF** 位清 0。
- (7) 如果要继续进行数据接收, 请在 **INTCBnR** 信号产生后, 读取 **CBnCTL0.CBnSCE** 位保持为 1 的 **CBnRX** 寄存器值, 等待串行时钟输入。
- (8) 要结束数据接收, 请写入 **CBnSCE** 位 = 0 即可。
- (9) 读取 **CBnRX** 寄存器值。
- (10) 要结束数据接收, 请写入 **CBnCTL0.CBnPWR** 位 = 0 以及 **CBnCTL0.CBnRXE** 位 = 0 即可。

备注       $n = 0, 1$

13.5.6 单向传输模式 (从模式，发送/接收模式)

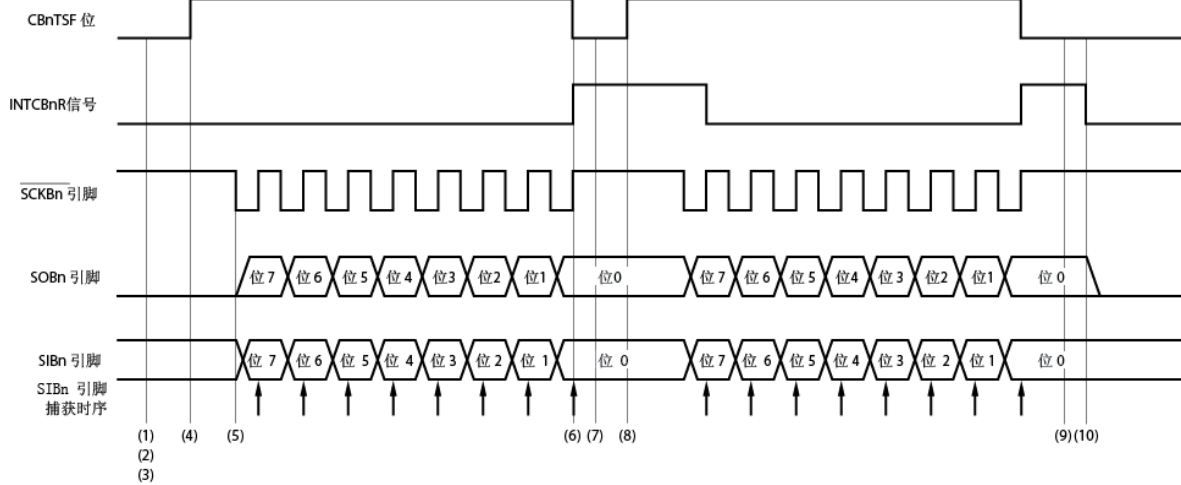
MSB 起始位 (CBnCTL0.CBnDIR 位= 0), 通信类型 1 (CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f<sub>CCLK</sub>) = 外部时钟 ( $\overline{\text{SCKBn}}$ ) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111), 传输数据位长度= 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



- 备注
- 1. 虚线框内表示的是硬件处理过程。
  - 2. 图中的数字对应与(2) 操作时序中的处理步骤序数。
  - 3. n = 0, 1

## (2) 操作时序



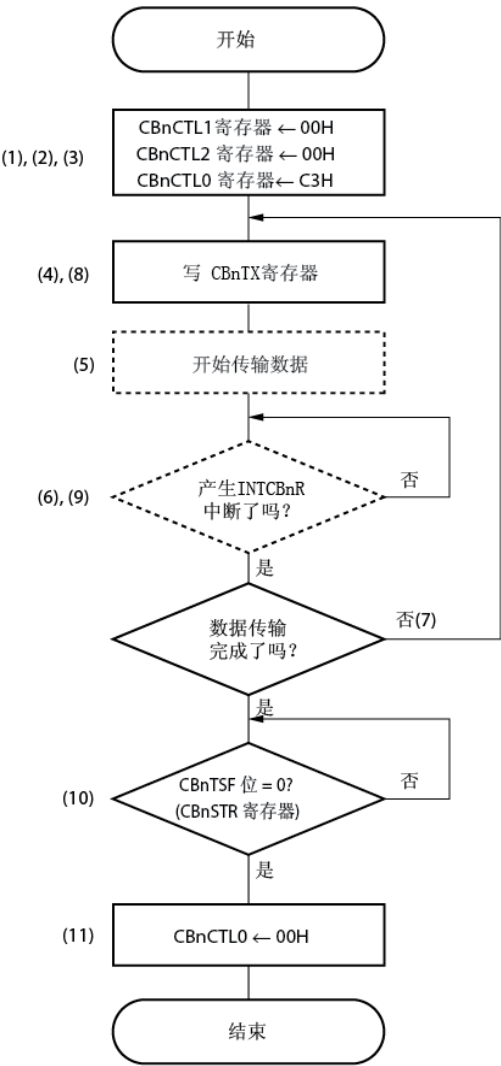
- (1) 将 07H 写入 CBnCTL1 寄存器, 并选择通信类型 1, 通信时钟 (f<sub>cCLK</sub>) = 外部时钟 (SCKBn), 以及从模式。
- (2) 将 00H 写入 CBnCTL2 寄存器, 并且设置传输数据长度为 8 位。
- (3) 将 E1H 写入 CBnCTL0 寄存器, 选择发送/接收模式, 并且在通信时钟(f<sub>cCLK</sub>)操作允许设定的同时, 选择 MSB 作为数据传输的起始位。.
- (4) 通过将发送数据写入 CBnTX 寄存器将 CBnSTR.CBnTSF 位设置为 1, 从而使得设备处于等待串行时钟信号输入状态。
- (5) 当有串行时钟信号输入时, 使用串行时钟向 SOBn 引脚同步输出发送数据, 同时捕获 SIBn 引脚的接收数据。
- (6) 当将由 CBnCTL2 寄存器所设定的一定长度传输数据发送/接收完毕的时候, 便停止串行时钟输出, 发送数据输出和数据捕获, 并在串行时钟的最后一个边沿产生接收完成中断请求信号 (INTCBnR), 同时将 CBnTSF 位清 0。
- (7) 读取 CBnRX 寄存器值。
- (8) 如果要继续进行数据发送/接收, 请再次向 CBnTX 寄存器写入发送数据。等待串行时钟输入。
- (9) 读取 CBnRX 寄存器值。
- (10) 要结束数据发送/接收, 请写入 CBnCTL0.CBnPWR 位 = 0, CBnCTL0.CBnTXE = 0 以及 CBnCTL0.CBnRXE 位 = 0 即可。

备注      n = 0, 1

13.5.7 连续传输模式 (主模式，发送模式)

MSB 起始位 (CBnCTL0.CBnDIR 位= 0), 通信类型 1 (CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f<sub>CLK</sub>) = f<sub>xx</sub>/2 (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000), 传输数据位长度= 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

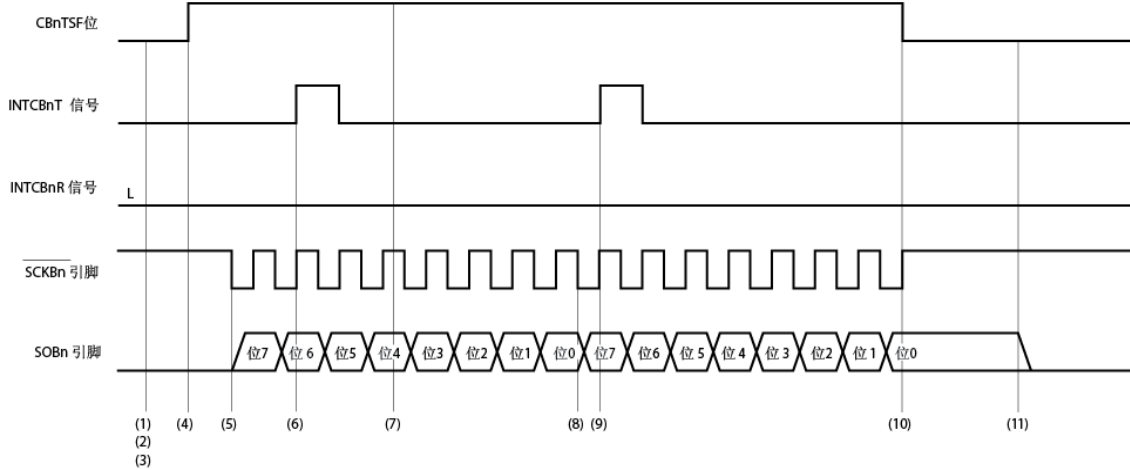
(1) 操作流程



- 备注
- 1. 虚线框内表示的是硬件处理过程。
  - 2. 图中的数字对应与(2) 操作时序中的处理步骤序号。
  - 3. n = 0, 1



## (2) 操作时序



- (1) 将 00H 写入 CBnCTL1 寄存器, 并选择通信类型 1, 通信时钟( $f_{CCLK}$ ) =  $f_{XX}/2$ , 以及主模式。
- (2) 将 00H 写入 CBnCTL2 寄存器, 并且设置传输数据长度为 8 位。
- (3) 将 C3H 写入 CBnCTL0 寄存器, 选择发送模式, MSB 作为数据传送起始位, 以及在通信时钟( $f_{CCLK}$ )操作允许设置的同时, 选择连续传输模式。
- (4) 通过将发送数据写入 CBnTX 寄存器将 CBnSTR.CBnTSF 位设置为 1, 启动发送操作过程。
- (5) 当数据发送启动后, 向 SCKBn 引脚输出串行时钟信号, 同时使用串行时钟从 SOBn 引脚同步输出发送数据。
- (6) 当发送数据由 CBnTX 寄存器向移位寄存器传送完毕, 同时写 CBnTX 寄存器操作允许时, 便会产生发送允许中断请求信号(INTCBnT)。
- (7) 如果要继续进行数据发送, 请在 INTCBnT 信号产生后, 再次向 CBnTX 寄存器写入发送数据。
- (8) 当在通信完成之前, 又向 CBnTX 寄存器写入了新的发送数据时, 那么在本次通信完成之后, 便启动了下一次通信过程。
- (9) 当发送数据由 CBnTX 寄存器向移位寄存器传送完毕时, 产生 INTCBnT 信号。要结束含有当前数据发送的连续发送过程, 请不要向 CBnTX 寄存器写入数据。
- (10) 当在数据传输完成之前, 没有向 CBnTX 寄存器写入下一个要发送的数据时, 便在传输完成之后, 停止向 SCKBn 引脚输出串行时钟, 同时将 CBnTSF 位清 0。
- (11) 要释放发送允许状态, 请在确认 CBnTSF 位为 0 之后, 写入 CBnCTL0.CBnPWR 位 = 0 以及 CBnCTL0.CBnTXE 位 = 0 即可。

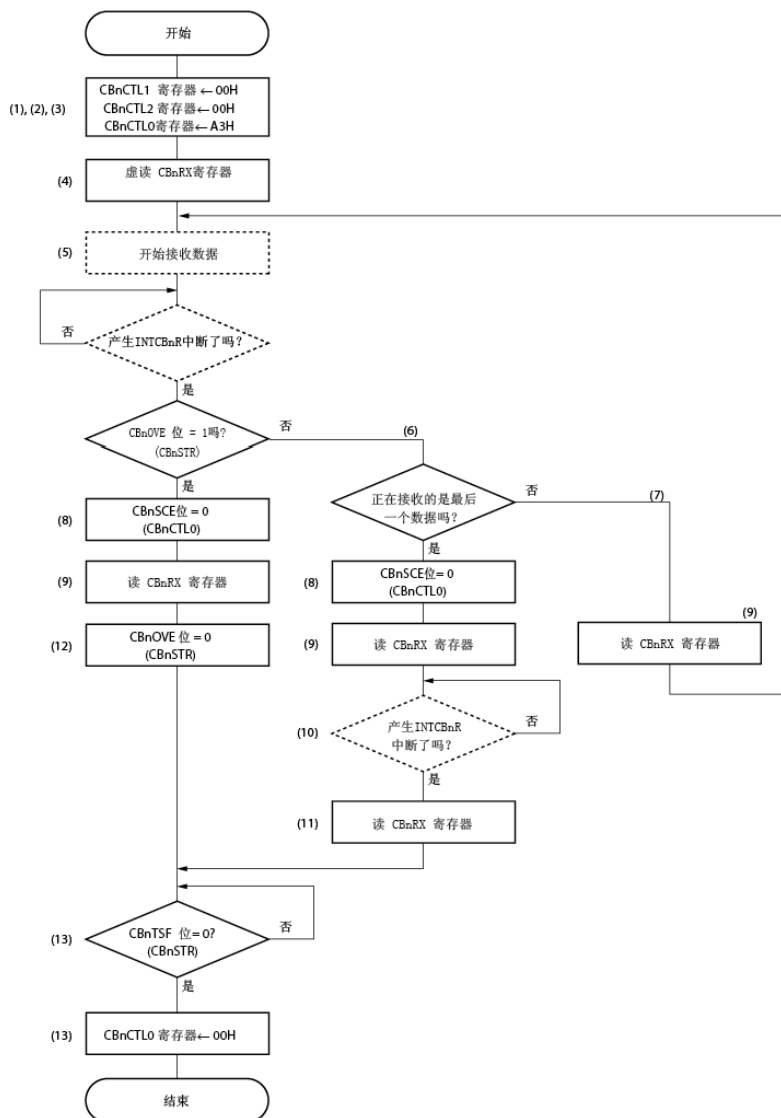
**注意事项** 在连续发送模式下, 不产生接收完成中断请求信号(INTCBnR)。

**备注** n = 0, 1

**13.5.8 连续传输模式 (主模式, 接收模式)**

MSB 起始位 (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位 = 00), 通信时钟 ( $f_{\text{CCLK}}$ ) =  $f_{\text{xx}}/2$  (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000), 传输数据位长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

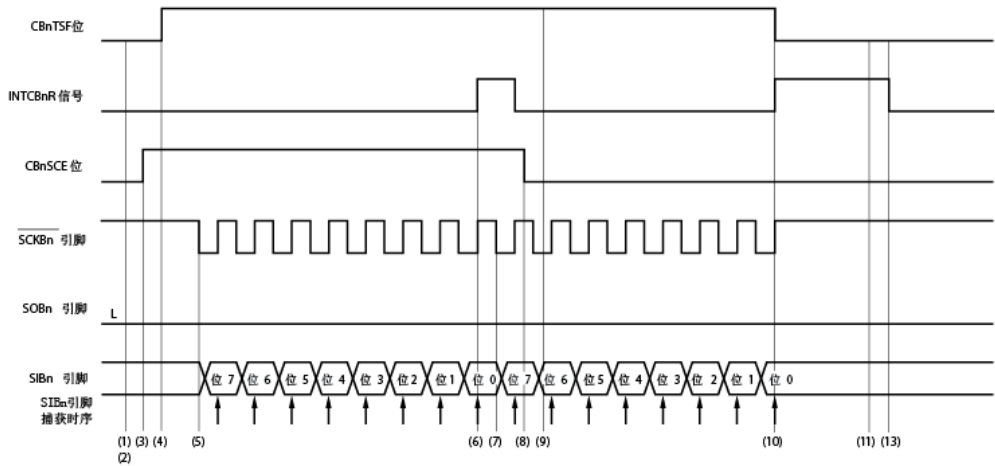
## (1) 操作流程



## 备注

1. 虚线框内表示的是硬件处理过程。
2. 图中的数字对应与(2)操作时序中的处理步骤序号。
3.  $n = 0, 1$

(2) 操作时序



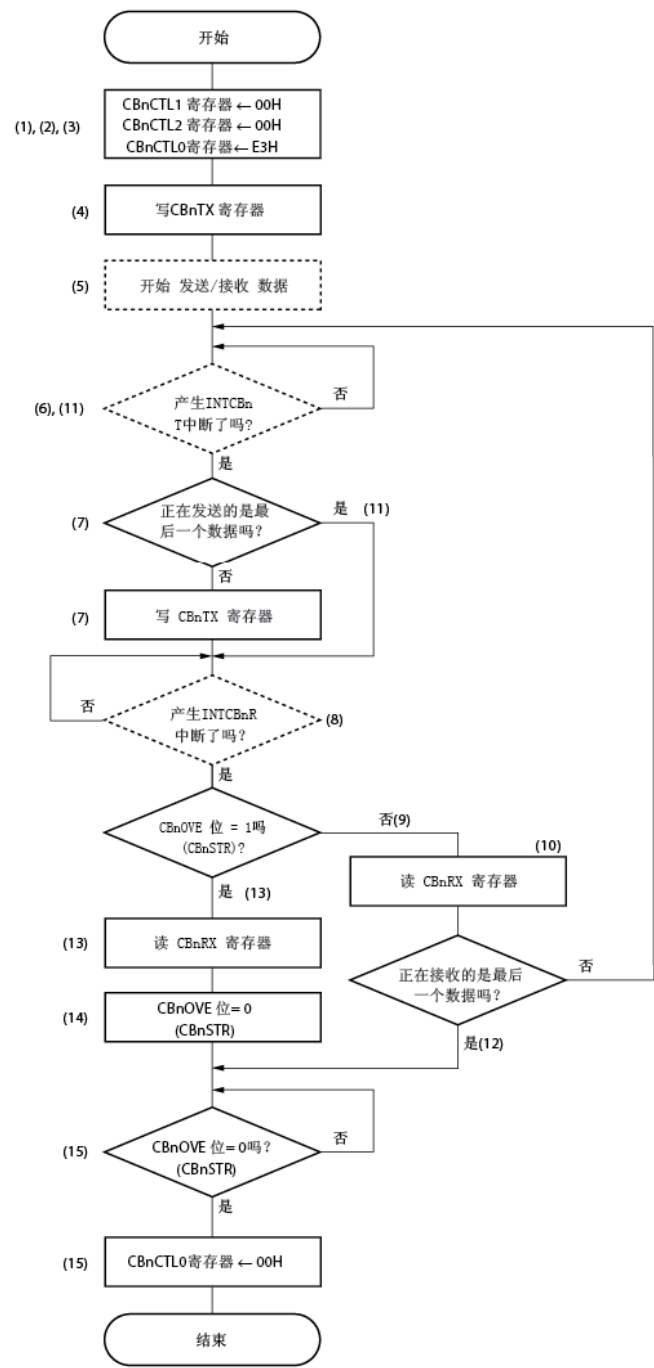
- (1) 将 00H 写入 CBnCTL1 寄存器, 并选择通信类型 1, 通信时钟( $f_{CCLK}$ ) =  $f_{XX}/2$ , 以及主模式。
- (2) 将 00H 写入 CBnCTL2 寄存器, 并且设置传输数据长度为 8 位。
- (3) 将 A3H 写入 CBnCTL0 寄存器, 选择接收模式, MSB 作为数据传送起始位, 以及在通信时钟( $f_{CCLK}$ )操作允许设置的同时, 选择连续传输模式。
- (4) 通过对 CBnRX 寄存器执行一个虚读操作, 将 CBnSTR.CBnTSF 位设置为 1, 启动数据接收过程。
- (5) 当数据接收启动后, 向 SCKBn 引脚输出串行时钟信号, 同时使用串行时钟同步捕获 SIBn 引脚的接收数据。
- (6) 当数据接收完成时, 便会产生接收完成中断请求信号(INTCBnR), 同时 CBnRX 寄存器的读操作允许。
- (7) 当在通信完成时, 设置 CBnCTL0.CBnSCE 位为 1, 那么在本次通信完成之后, 便启动了下一次通信过程。
- (8) 要结束含有当前数据接收操作的连续数据接收过程, 请写入 CBnSCE 位 = 0 即可。
- (9) 读取 CBnRX 寄存器值。
- (10) 当数据接收完成时, 便会产生接收完成中断请求信号(INTCBnR), 同时 CBnRX 寄存器的读操作允许。当在通信完成之前, 将 CBnSCE 位设置为 0, 则会停止向 SCKBn 引脚输出串行时钟, 同时将 CBnTSF 位清 0, 以结束该接收操作。
- (11) 读取 CBnRX 寄存器值。
- (12) 如果有溢出覆盖 (overrun) 错误发生, 则将 CBnSTR.CBnOVE 位清 0, 并清除错误标志。
- (13) 要释放接收允许状态, 请在确认 CBnTSF 位为 0 之后, 写入 CBnCTL0.CBnPWR 位 = 0 以及 CBnCTL0.CBnTXE 位 = 0 即可。

备注 n = 0, 1

**13.5.9 连续传输模式 (主模式, 发送/接收模式)**

MSB 起始位 (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位 = 00), 通信时钟 ( $f_{CCLK}$ ) =  $f_{XX}/2$  (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000), 传输数据位长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程

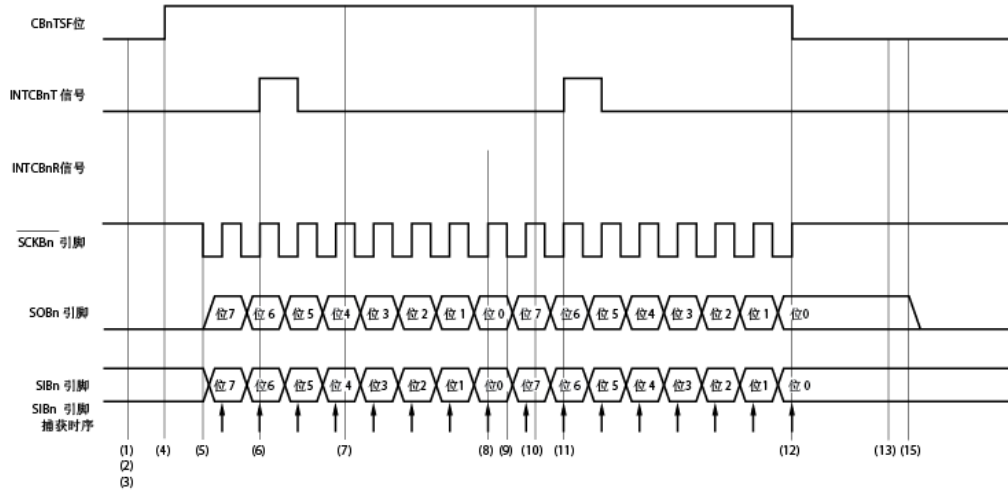


备注

- 1. 虚线框内表示的是硬件处理过程。
- 2. 图中的数字对应与(2) 操作时序 中的处理步骤序数。
- 3. n = 0, 1

## (2) 操作时序

(1/2)



- (1) 将 00H 写入 **CBnCTL1** 寄存器, 并选择通信类型 1, 通信时钟( $f_{CCLK}$ ) =  $f_{XX}/2$ , 以及主模式。
- (2) 将 00H 写入 **CBnCTL2** 寄存器, 并且设置传输数据长度为 8 位。
- (3) 将 E3H 写入 **CBnCTL0** 寄存器, 选择发送/接收模式,, **MSB** 作为数据传送起始位, 以及在通信时钟( $f_{CCLK}$ )操作允许设置的同时, 选择连续传输模式。
- (4) 通过将发送数据写入 **CBnTX** 寄存器将 **CBnSTR.CBnTSF** 位设置为 1, 启动发送/接收操作过程。
- (5) 当数据发送/接收启动后, 向 **SCKBn** 引脚输出串行时钟信号, 使用串行时钟向 **SOBn** 引脚同步输出发送数据, 同时捕获 **SIBn** 引脚的接收数据。
- (6) 当发送数据由 **CBnTX** 寄存器向移位寄存器传送完毕, 同时写 **CBnTX** 寄存器操作允许时, 便会产生发送允许中断请求信号(**INTCBnT**)。
- (7) 如果要继续进行数据发送/接收, 请在 **INTCBnT** 信号产生后, 再次向 **CBnTX** 寄存器写入发送数据。
- (8) 当数据发送/接收完成时, 便会产生接收完成中断请求信号(**INTCBnR**), 同时 **CBnRX** 寄存器的读操作允许。
- (9) 当在通信完成之前, 又向 **CBnTX** 寄存器写入了新的发送数据时, 那么在本次通信完成之后, 便启动了下一次通信过程。
- (10) 读取 **CBnRX** 寄存器值。

备注       $n = 0, 1$

- (11) 当发送数据由 CBnTX 寄存器向移位寄存器传送完毕时, 产生 INTCBnT 信号。要结束含有当前数据发送/接收操作的连续数据发送/接收过程, 请不要写 CBnTX 寄存器。
- (12) 当在数据传输完成之前, 没有向 CBnTX 寄存器写入下一个要发送的数据时, 便在传输完成之后, 停止向 SCKBn 引脚输出串行时钟, 同时将 CBnTSF 位清 0。
- (13) 当产生接收错误中断请求信号 (INTCBnR), 便读取 CBnRX 寄存器值。
- (14) 如果有溢出覆盖 (overrun) 错误发生, 则将 CBnSTR.CBnOVE 位清 0, 并清除错误标志。
- (15) 要释放发送/接收允许状态, 请在确认 CBnTSF 位为 0 之后, 写入 CBnCTL0.CBnPWR 位 = 0 以及 CBnCTL0.CBnTXE 位 = 0 即可。

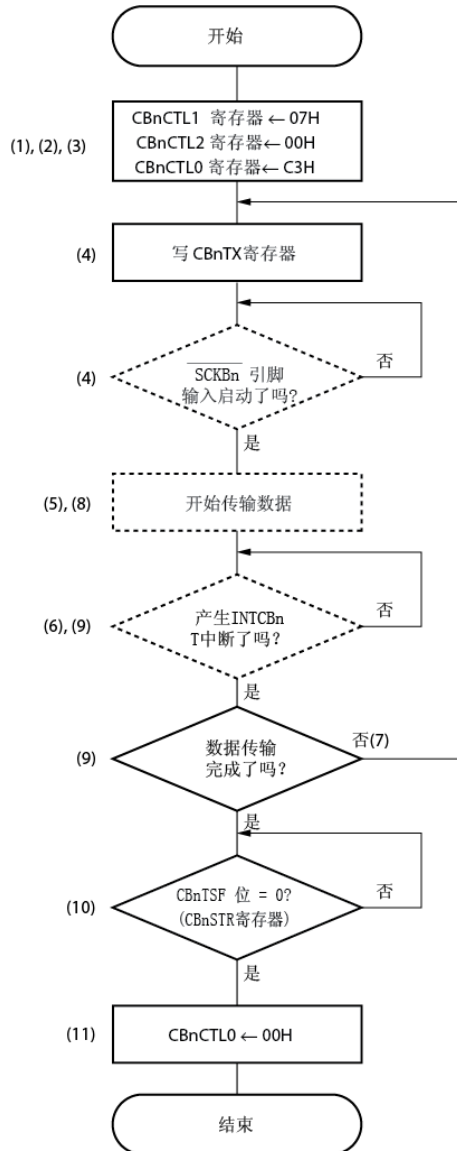
备注      n = 0, 1



## 13.5.10 连续传输模式 (从模式, 发送模式)

MSB 起始位 (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (fCCLK) = 外部时钟 ( $\overline{\text{SCKBn}}$ ) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111), 传输数据位长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

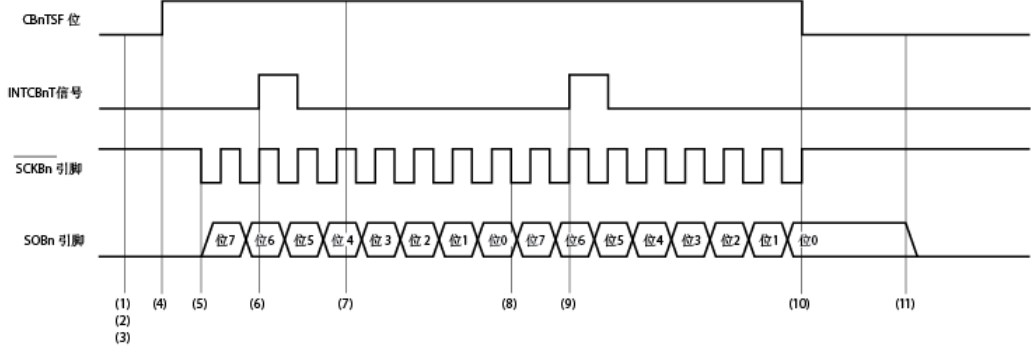
## (1) 操作流程



## 备注

1. 虚线框内表示的是硬件处理过程。
2. 图中的数字对应与(2)操作时序中的处理步骤序号。
3. n = 0, 1

(2) 操作时序



- (1) 将 07H 写入 CBnCTL1 寄存器, 并选择通信类型 1, 通信时钟 (fCCLK) = 外部时钟 (SCKBn), 以及从模式。
- (2) 将 00H 写入 CBnCTL2 寄存器, 并且设置传输数据长度为 8 位。
- (3) 将 C3H 写入 CBnCTL0 寄存器, 选择发送模式, MSB 作为数据传送起始位, 以及在通信时钟(fCCLK)操作允许设置的同时, 选择连续传输模式。
- (4) 通过将发送数据写入 CBnTX 寄存器将 CBnSTR.CBnTSF 位设置为 1, 从而使得设备处于等待串行时钟信号输入状态。
- (5) 当有串行时钟信号输入时, 使用串行时钟, 便从 SOBn 引脚同步输出发送数据。
- (6) 当发送数据由 CBnTX 寄存器向移位寄存器传送完毕, 同时写 CBnTX 寄存器操作允许时, 便会产生发送允许中断请求信号(INTCBnT)。
- (7) 如果要继续进行数据发送, 请在 INTCBnT 信号产生后, 再次向 CBnTX 寄存器写入发送数据。
- (8) 当在由 CBnCTL2 寄存器所设定的一定长度传输数据发送完毕后, 有串行时钟信号输入时, 便启动了连续数据发送过程。
- (9) 当发送数据由 CBnTX 寄存器向移位寄存器传送完毕, 同时写 CBnTX 寄存器操作允许时, 便产生 INTCBnT 信号。要结束含有当前数据发送的连续发送过程, 请不要向 CBnTX 寄存器写入数据。
- (10) 当 CBnCTL2 寄存器设定传输数据长度的时钟输入时, 没有向 CBnTX 寄存器写入数据, 这时便将 CBnTSF 位清为 0 以结束数据发送操作。
- (11) 要释放发送允许状态, 请在确认 CBnTSF 位为 0 之后, 写入 CBnCTL0.CBnPWR 位 = 0 以及 CBnCTL0.CBnTXE 位 = 0 即可。

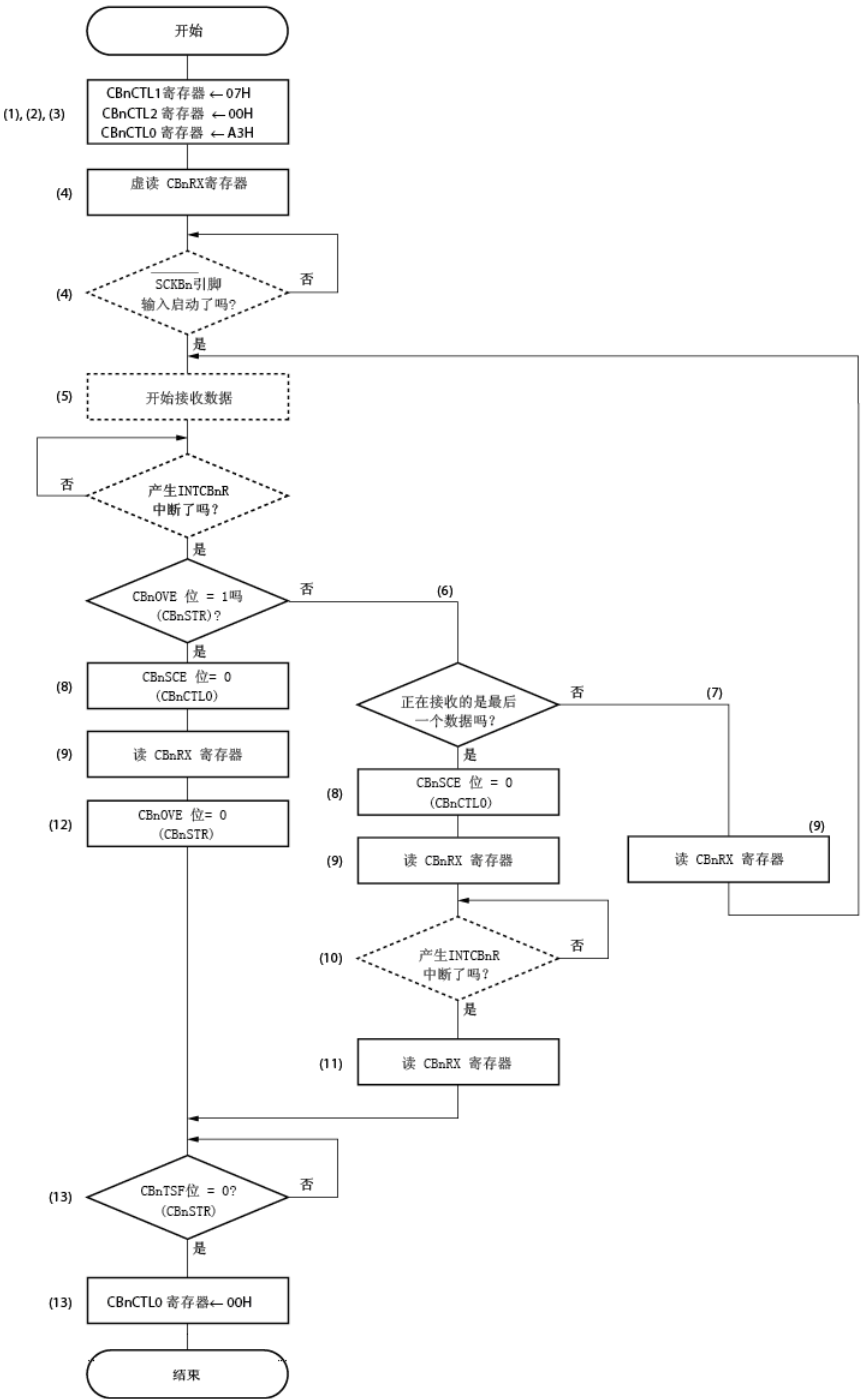
**注意事项** 在连续发送模式下, 不产生接收完成中断请求信号(INTCBnR)。

**备注** n = 0, 1

**13.5.11 连续传输模式 (从模式, 接收模式)**

MSB 起始位 (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f<sub>CCLK</sub>) = 外部时钟 ( $\overline{\text{SCKBn}}$ ) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111), 传输数据位长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

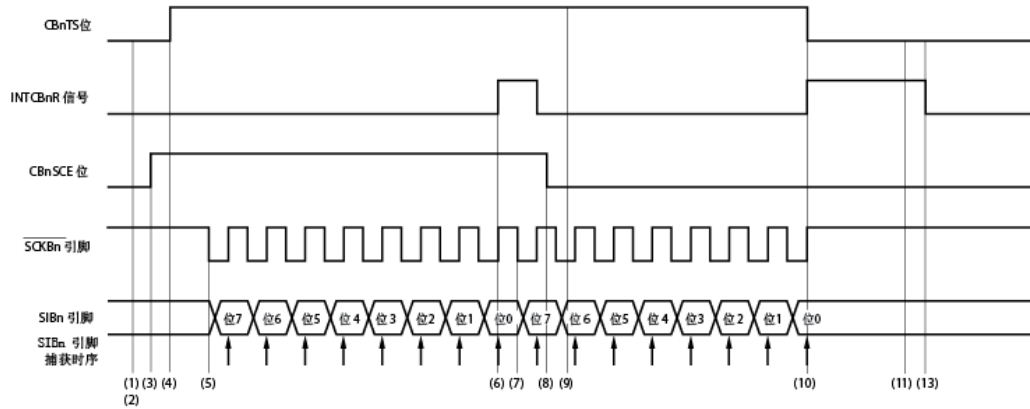
(1) 操作流程



备注

- 1. 虚线框内表示的是硬件处理过程。
- 2. 图中的数字对应与(2)操作时序中的处理步骤序数。
- 3. n = 0, 1

## (2) 操作时序



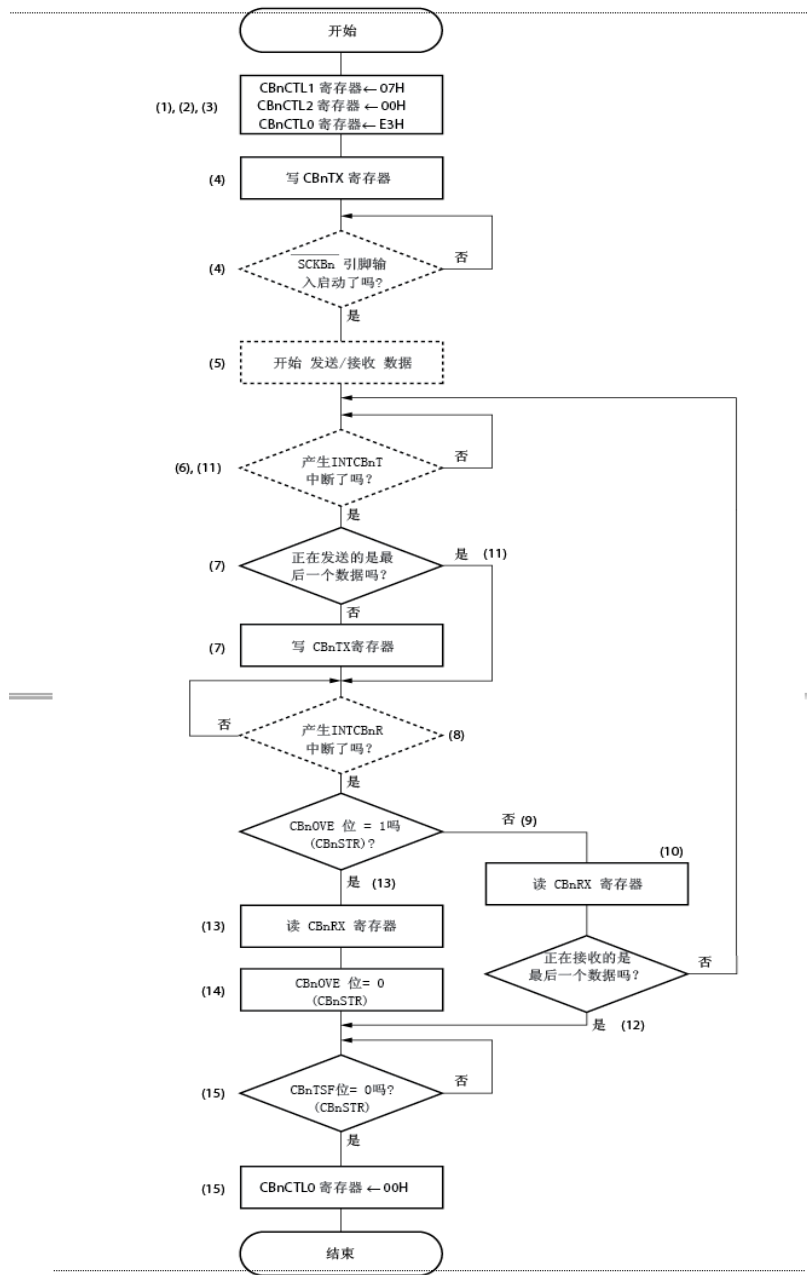
- (1) 将 07H 写入 CBnCTL1 寄存器, 并选择通信类型 1, 通信时钟 (fcCLK) = 外部时钟 (SCKBn), 以及从模式。
- (2) 将 00H 写入 CBnCTL2 寄存器, 并且设置传输数据长度为 8 位。
- (3) 将 A3H 写入 CBnCTL0 寄存器, 选择接收模式,, MSB 作为数据传送起始位, 以及在通信时钟(fcCLK)操作允许设置的同时, 选择连续传输模式。
- (4) 通过对 CBnRX 寄存器执行一个虚读操作, 将 CBnSTR.CBnTSF 位设置为 1, 从而使得设备处于等待串行时钟信号输入状态。
- (5) 当有串行时钟信号输入时, 使用串行时钟同步捕获 SIBn 引脚的接收数据。
- (6) 当数据接收完成时, 便会产生接收完成中断请求信号(INTCBnR), 同时 CBnRX 寄存器的读操作允许。
- (7) 当在 CBnCTL0.CBnSCE 位= 1 状态下,有串行时钟输入时,便启动了连续接收过程。
- (8) 要结束含有当前数据接收操作的连续数据接收过程, 请写入 CBnSCE 位 = 0 即可。
- (9) 读取 CBnRX 寄存器值。
- (10) 当数据接收完成时, 便会产生接收完成中断请求信号(INTCBnR), 同时 CBnRX 寄存器的读操作允许。  
当在通信完成之前, 设置 CBnSCE 位= 0,则将 CBnTSF 位清 0, 以结束该接收操作。
- (11) 读取 CBnRX 寄存器值。
- (12) 如果有溢出覆盖 (overrun) 错误发生, 则将 CBnSTR.CBnOVE 位清 0, 并清除错误标志。
- (13) 要释放接收允许状态, 请在确认 CBnTSF 位为 0 之后, 写入 CBnCTL0.CBnPWR 位 = 0 以及 CBnCTL0.CBnTXE 位 = 0 即可。

备注      n = 0, 1

**13.5.12 连续传输模式 (从模式, 发送/接收模式)**

MSB 起始位 (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位 = 00), 通信时钟 ( $f_{\text{CCLK}}$ ) = 外部时钟 ( $\overline{\text{SCKBn}}$ ) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111), 传输数据位长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

## (1) 操作流程

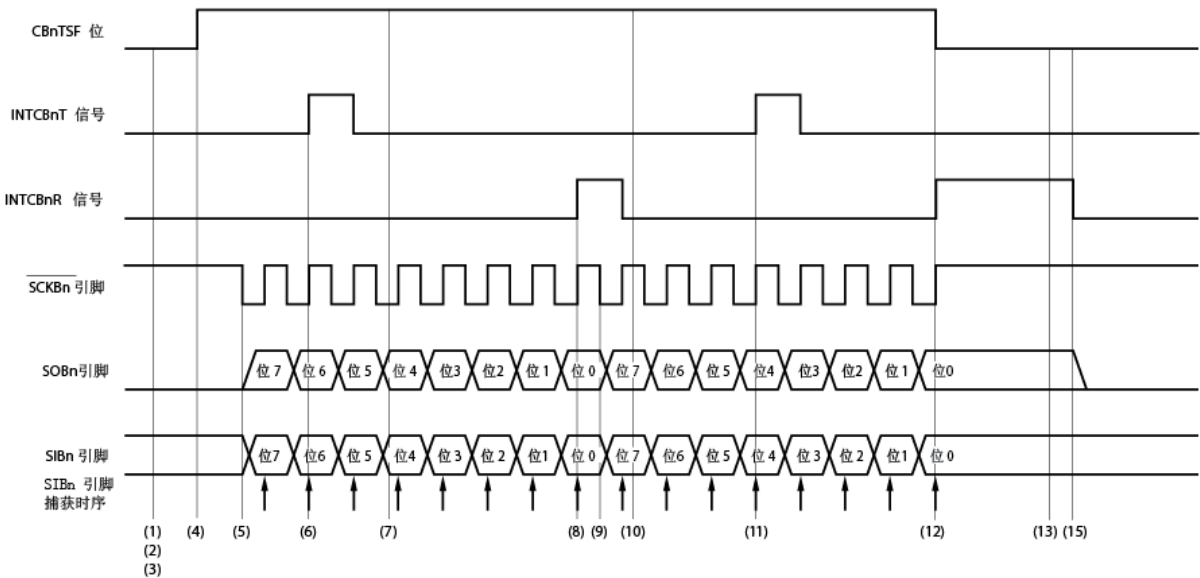


## 备注

1. 虚线框内表示的是硬件处理过程。
2. 图中的数字对应与**(2) 操作时序**中的处理步骤序号。
3.  $n = 0, 1$

(2) 操作时序

(1/2)



- (1) 将 07H 写入 **CBnCTL1** 寄存器, 并选择通信类型 1, 通信时钟 ( $f_{CLK}$ ) = 外部时钟 (**SCKBn**), 以及从模式。
- (2) 将 00H 写入 **CBnCTL2** 寄存器, 并且设置传输数据长度为 8 位。
- (3) 将 E3H 写入 **CBnCTL0** 寄存器, 选择发送/接收模式, **MSB** 作为数据传送起始位, 以及在通信时( $f_{CLK}$ )钟操作允许设置的同时, 选择连续传输模式。
- (4) 通过将发送数据写入 **CBnTX** 寄存器将 **CBnSTR.CBnTSF** 位设置为 1, 从而使得设备处于等待串行时钟信号输入状态。
- (5) 当有串行时钟信号输入时, 使用串行时钟向 **SOBn** 引脚同步输出发送数据, 同时捕获 **SIBn** 引脚的接收数据。
- (6) 当发送数据由 **CBnTX** 寄存器向移位寄存器传送完毕, 同时写 **CBnTX** 寄存器操作允许时, 便会产生发送允许中断请求信号(**INTCBnT**)。
- (7) 如果要继续进行数据发送, 请在 **INTCBnT** 信号产生后, 再次向 **CBnTX** 寄存器写入发送数据。
- (8) 当将由 **CBnCTL2** 寄存器所设定的一定长度传输数据接收完毕的时候, 便会产生接收完成中断请求信号(**INTCBnR**), 同时 **CBnRX** 寄存器的读操作允许。
- (9) 当连续输入串行时钟信号时, 便启动了连续发送/接收数据过程。
- (10) 读取 **CBnRX** 寄存器值。
- (11) 当发送数据由 **CBnTX** 寄存器向移位寄存器传送完毕, 同时写 **CBnTX** 寄存器操作允许时, 便产生 **INTCBnT** 信号。 要结束含有当前数据发送/接收操作的连续数据发送/接收过程, 请不要写 **CBnTX** 寄存器。

备注  $n = 0, 1$



- (12) 当 CBnCTL2 寄存器设定传输数据长度的时钟信号输入时, 没有向 CBnTX 寄存器写入数据, 便产生 INTCBnR 信号。这时要将 CBnTSF 位清为 0 以结束数据发送/接收操作。
- (13) 当 INTCBnR 信号产生时, 读取 CBnRX 寄存器值。
- (14) 如果有溢出覆盖 (overrun) 错误发生, 则将 CBnSTR.CBnOVE 位清 0, 并清除错误标志。
- (15) 要释放发送/接收允许状态, 请在确认 CBnTSF 位为 0 之后, 写入 CBnCTL0.CBnPWR 位 = 0 以及 CBnCTL0.CBnTXE 位 = 0 即可。

备注      n = 0, 1

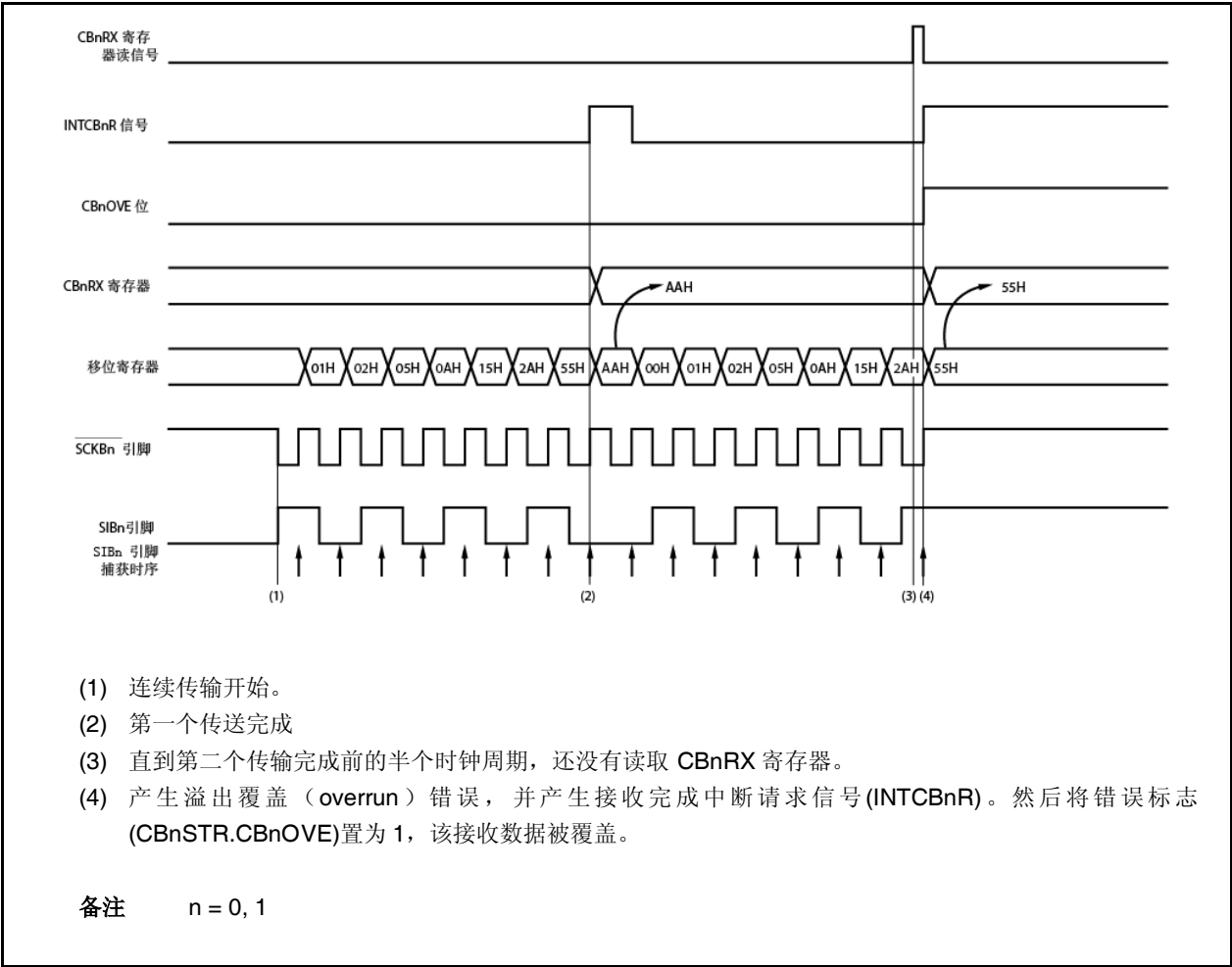
13.5.13 接收错误

连续传输模式下，在接收允许状态(CBnCTL0.CBnRXE 位 = 1)下执行数据传输时，如果在 INTCBnR 信号产生后，CBnRX 寄存器读取之前，完成了下一个接收操作，那么便又会产生一个接收完成中断请求信号(INTCBnR)。并且将溢出覆盖(overrun)错误标志(CBnSTR.CBnOVE)置为 1。

即使产生了溢出覆盖(overrun)错误，由于 CBnRX 寄存器中的数据已经更新，先前的接收数据也会被丢失掉。即使产生了接收错误，如果没有对 CBnRX 寄存器进行读取，当下一个接收过程完成时，也会再产生一个 INTCBnR 信号。

为了避免 overrun 错误的产生，请在 INTCBnR 信号产生以后，在对下一个接收数据的最后一位进行采样之前的半个时钟周期内，完成对 CBnRX 寄存器的读操作。

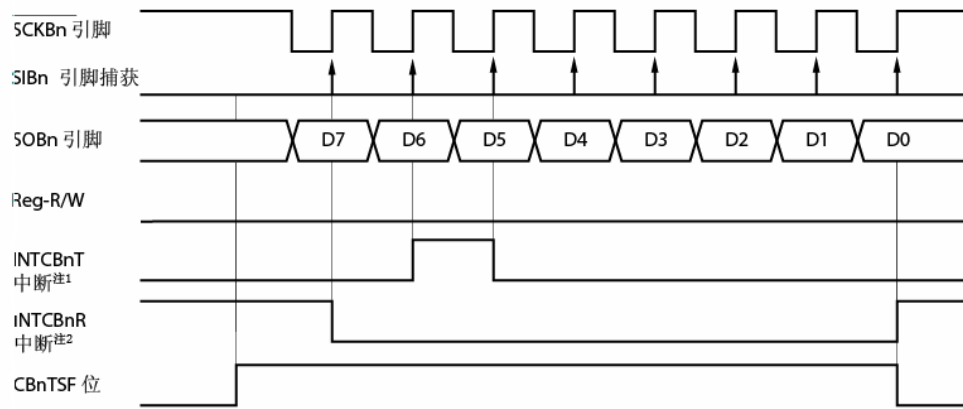
(1) 操作时序



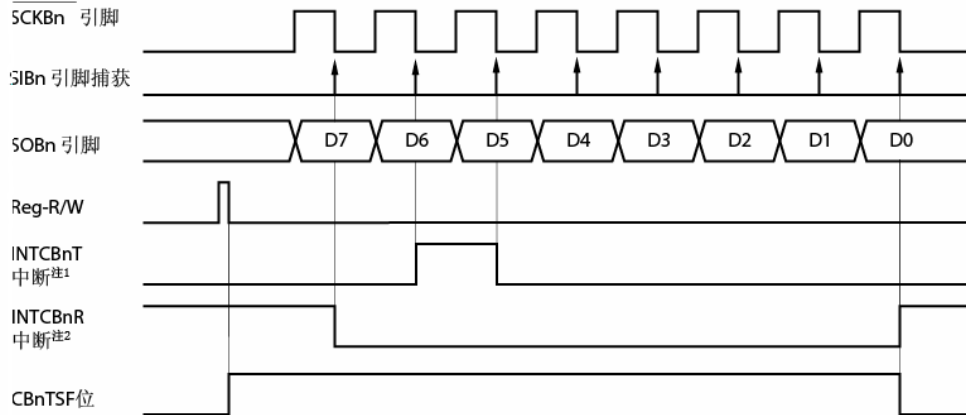
## 13.5.14 时钟时序

(1/2)

(i) 通信类型 1 (CBnCKP 和 CBnDAP 位 = 00)



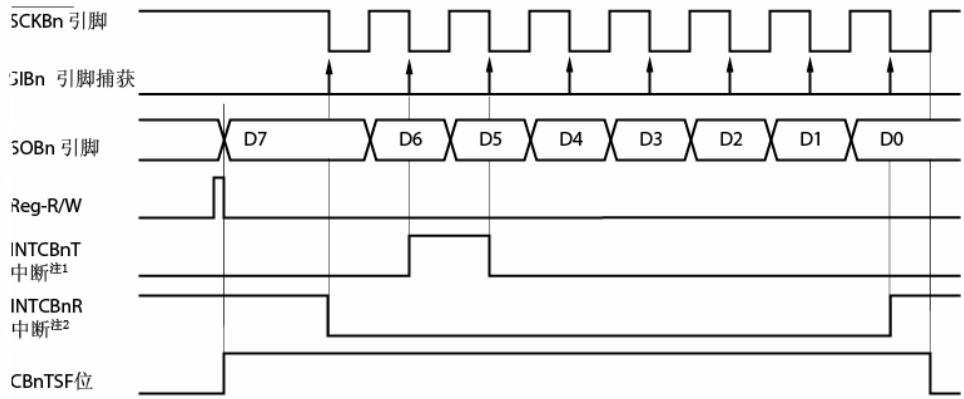
(ii) 通信类型 3 (CBnCKP 和 CBnDAP 位 = 10)



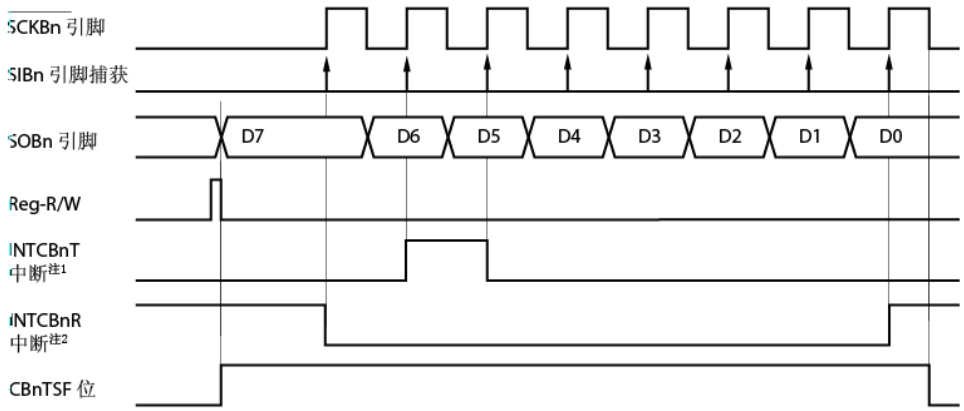
- 注 1. 在连续发送模式或连续发送/接收模式下，当写入 CBnTX 寄存器的数据传输到移位寄存器中去的时候，便会设置 INTCBnT 中断。在单向发送模式或单向发送/接收模式下，则不会产生 INTCBnT 中断请求信号，但是当通信结束时，会产生 INTCBnR 中断请求信号。
2. 如果数据接收能够正确地结束，而且在接收允许状态下，读取了 CBnRX 寄存器中的数据，这时便会产生 INTCBnR 中断。在单向传输模式下，即便是单向发送模式，当通信结束时，也会产生 INTCBnR 中断请求信号。

注意事项 在单向传输模式下，在 CBnTSF 位为 1 的情况下，CBnTX 寄存器写操作会被忽略。但是，这对传输期间的操作并没有影响。

(iii) 通信类型 2 (CBnCKP 和 CBnDAP 位 = 01)



(iv) 通信类型 4 (CBnCKP 和 CBnDAP 位 = 11)



- 注
1. 在连续发送模式或连续发送/接收模式下，当写入 CBnTX 寄存器的数据传输到移位寄存器中去的时候，便会设置 INTCBnT 中断。在单向发送模式或单向发送/接收模式下，则不会产生 INTCBnT 中断请求信号，但是当通信结束时，会产生 INTCBnR 中断请求信号。
  2. 如果数据接收能够正确地结束，而且在接收允许状态下，读取了 CBnRX 寄存器中的数据，这时便会产生 INTCBnR 中断。在单向传输模式下，即便是单向发送模式，当通信结束时，也会产生 INTCBnR 中断请求信号。

注意事项 在单向传输模式下，在 CBnTSF 位为 1 的情况下，CBnTX 寄存器写操作会被忽略。但是，这对传输期间的操作并没有影响。

## 13.6 操作禁用时,输出引脚的状态

(1)  $\overline{\text{SCKBn}}$  引脚

当 CSIBn 操作禁用(CBnCTL0.CBnPWR 位 = 0)时,  $\overline{\text{SCKBn}}$  引脚 的输出状态如下。

CBnCKS2	CBnCKS1	CBnCKS0	CBnCKP	$\overline{\text{SCKBn}}$ 引脚 的输出
1	1	1	x	高阻态
以上数值以外的其它值			0	固定为高电平
			1	固定为低电平

- 备注**
1. 如果 CBnCTL1.CBnCKP 以及 CBnCKS2 至 CBnCKS0 中的任何一位被重写, 则  $\overline{\text{SCKBn}}$  引脚的输出电平会发生改变。
  2. n = 0, 1
  3. x: 不必考虑

(2)  $\text{SOBn}$  引脚

当 CSIBn 操作禁用(CBnPWR 位 = 0)时,  $\text{SOBn}$  引脚的输出状态如下。

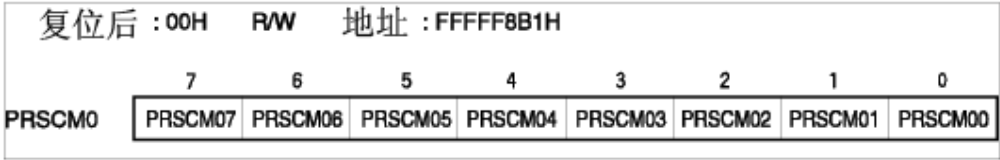
CBnTXE	CBnDAP	CBnDIR	$\text{SOBn}$ 引脚 输出
0	x	x	固定为低电平
1	0	x	$\text{SOBn}$ 锁存值 (低电平)
		0	CBnTX 寄存器值 (MSB)
		1	CBnTX 寄存器值 (LSB)

- 备注**
1. 如果 CBnCTL0.CBnTXE, CBnCTL0.CBnDIR 位, 以及 CBnCTL1.CBnDAP 中的任何一位被重写,  $\text{SOBn}$  引脚的输出会发生改变。
  2. n = 0, 1
  3. x: 不必考虑



(2) 预分频比较寄存器 0 (PRSCM0)

PRSCM0 寄存器是一个 8 位比较寄存器。  
该寄存器可以以 8 位为单位进行读或写操作。  
复位后，该寄存器被设置为 00H。



- 注意事项
- 1. 当钟表定时器和 CSIB0 正在运行时，请不要重写 PRSCM0 寄存器。
  - 2. 在 PRSM0.BGCE0 位设置为 1 之前，对 PRSCM0 寄存器进行设置。

13.7.1 波特率的产生

通过对主时钟分频来生成发送/接收时钟，按照如下公式计算由主时钟所产生的波特率值。

$$f_{BRG} = \frac{f_{XX}}{2^{k+1} \times N}$$

<R>      注意事项      将 f<sub>BRG</sub> 设置为 8 MHz 或更低一些。

备注

f<sub>BRG</sub>:    BRG 计数时钟

f<sub>XX</sub>:    主时钟振荡频率

k:        PRSM0 寄存器设定值 = 0 至 3

N:        PRSCM0 寄存器设定值= 1 至 256

但是, 仅当 PRSCM0 寄存器设置为 00H 时, N = 256。

### 13.8 注意事项

- (1) 关于在操作期间(CBnCTL0.CBnPWR 位为 1)禁止重写的寄存器，如果在操作期间，错误地执行了重写操作，那么请立即将 CBnCTL0.CBnPWR 设置为 0，然后初始化 CSIBn。

操作期间禁止重写的寄存器如下所示。

- CBnCTL0 寄存器: CBnTXE, CBnRXE, CBnDIR, CBnTMS 位
- CBnCTL1 寄存器: CBnCKP, CBnDAP, CBnCKS2 to CBnCKS0 位
- CBnCTL2 寄存器: CBnCL3 to CBnCL0 位

- (2) 在通信类型 2 和 4 (CBnCTL1.CBnDAP 位 = 1)中，要在接收完成中断请求(INTCBnR)产生后的半个  $\overline{\text{SCKBn}}$  时钟期内，将 CBnSTR.CBnTSF 位清 0。

在单向传输模式下，通信期间(CBnTSF 位 = 1)的下一个发送数据写操作会被忽略，也不会启动下一次通信。同样，如果设置了只接收的通信方式(CBnCTL0.CBnTXE 位 = 0, CBnCTL0.CBnRXE 位 = 1)，在通信期间(CBnTSF 位 = 1)，即使读取了接收数据，也不会启动下一次通信。

因此，当在通信类型 2 或 4 下(CBnDAP 位 = 1)，使用单向传输模式，要特别注意以下几点。

- 如果要启动下一次数据发送，首先请确认 CBnTSF 位 = 0，然后再将发送数据写入 CBnTX 寄存器中去。
- 如果要在只接收的通信方式(CBnCTL0.CBnTXE 位 = 0, CBnCTL0.CBnRXE 位 = 1)连续执行下一个接收操作，请先确认 CBnTSF 位 = 0，然后再读取 CBnRX 寄存器数据。

或者，使用连续传输模式，而不使用单向传输模式。

备注      n = 0, 1



## 第十四章 中断/异常处理功能

V850ES/HF2 单片机配备了一个专用中断控制器(INTC)，用以处理中断服务。V850ES/HF2 一共可以处理 41 个中断请求。

中断是指在程序执行过程中产生的不依赖于执行程序的事件，异常则是在程序执行过程中产生的与执行程序有一定关系的事件。

V850ES/HF2 单片机可以处理来自片上外围硬件和外部中断源的中断请求信号。同时，V850ES/HF2 通过指令 TRAP(软件异常)或通过异常事件的产生(即，非法操作码的拾取或捕获) (异常陷阱)，可以启动异常处理过程。

### 14.1 特性

#### ○ 中断

- 不可屏蔽中断: 2 个中断源
- 可屏蔽中断: 外部: 8 中断源个, 内部: 31 个中断源
- 8 级可编程中断优先权 (可屏蔽中断)
- 根据优先级进行多个中断控制
- 可对每个可屏蔽中断请求进行屏蔽设定。
- 噪声消除, 边沿检测, 以及对外部中断请求信号的有效触发方式的设定。

#### ○ 异常

- 软件异常: 32 个异常源
- 异常陷阱: 2 个异常源(非法操作码异常, 调试陷阱)

表 14-1 列出了所有中断/异常源。

表 14-1. 中断源列表 (1/2)

类型	分类	默认优先级	名称	触发方式	产生的部件	异常代码	处理程序地址	PC 恢复值	中断控制寄存器
复位	中断	–	RESET	RESET 引脚输入 内部复位输入	RESET	0000H	00000000H	不确定	–
不可屏蔽	中断	–	NMI	NMI 引脚有效边沿输入	引脚	0010H	00000010H	nextPC	–
		–	INTWDT2	WDT2 溢出	WDT2	0020H	00000020H	注 1	–
软件异常	异常	–	TRAP0n <sup>注 2</sup>	TRAP 指令	–	004nH <sup>注 2</sup>	00000040H	nextPC	–
		–	TRAP1n <sup>注 2</sup>	TRAP 指令	–	005nH <sup>注 2</sup>	00000050H	nextPC	–
异常陷阱	异常	–	ILGOP/ DBG0	非法操作码/ DBTRAP 指令	–	0060H	00000060H	nextPC	–
可屏蔽	中断	0	INTLVI	低电压检测	POCLVI	0080H	00000080H	nextPC	LVIIC
		1	INTP0	外部中断引脚输入边沿检测 (INTP0)	引脚	0090H	00000090H	nextPC	PIC0
		2	INTP1	外部中断引脚输入边沿检测 (INTP1)	引脚	00A0H	000000A0H	nextPC	PIC1
		3	INTP2	外部中断引脚输入边沿检测 (INTP2)	引脚	00B0H	000000B0H	nextPC	PIC2
		4	INTP3	外部中断引脚输入边沿检测 (INTP3)	引脚	00C0H	000000C0H	nextPC	PIC3
		5	INTP4	外部中断引脚输入边沿检测 (INTP4)	引脚	00D0H	000000D0H	nextPC	PIC4
		6	INTP5	外部中断引脚输入边沿检测 (INTP5)	引脚	00E0H	000000E0H	nextPC	PIC5
		7	INTP6	外部中断引脚输入边沿检测 (INTP6)	引脚	00F0H	000000F0H	nextPC	PIC6
		8	INTP7	外部中断引脚输入边沿检测 (INTP7)	引脚	0100H	00000100H	nextPC	PIC7
		9	INTTQ0OV	TMQ0 溢出	TMQ0	0110H	00000110H	nextPC	TQ0OVIC
		10	INTTQ0CC0	TMQ0 捕获 0/比较 0 匹配	TMQ0	0120H	00000120H	nextPC	TQ0CCIC0
		11	INTTQ0CC1	TMQ0 捕获 1/比较 1 匹配	TMQ0	0130H	00000130H	nextPC	TQ0CCIC1
		12	INTTQ0CC2	TMQ0 捕获 2/比较 2 匹配	TMQ0	0140H	00000140H	nextPC	TQ0CCIC2
		13	INTTQ0CC3	TMQ0 捕获 3/比较 3 匹配	TMQ0	0150H	00000150H	nextPC	TQ0CCIC3
		14	INTTP0OV	TMP0 溢出	TMP0	0160H	00000160H	nextPC	TP0OVIC
		15	INTTP0CC0	TMP0 捕获 0/比较 0 匹配	TMP0	0170H	00000170H	nextPC	TP0CCIC0
		16	INTTP0CC1	TMP0 捕获 1/比较 1 匹配	TMP0	0180H	00000180H	nextPC	TP0CCIC1
		17	INTTP1OV	TMP1 溢出	TMP1	0190H	00000190H	nextPC	TP1OVIC
		18	INTTP1CC0	TMP1 捕获 0/比较 0 匹配	TMP1	01A0H	000001AH	nextPC	TP1CCIC0
		19	INTTP1CC1	TMP1 捕获 1/比较 1 匹配	TMP1	01B0H	000001B0H	nextPC	TP1CCIC1
		20	INTTP2OV	TMP2 溢出	TMP2	01C0H	000001C0H	nextPC	TP2OVIC
		21	INTTP2CC0	TMP2 捕获 0/比较 0 匹配	TMP2	01D0H	000001D0H	nextPC	TP2CCIC0

注 1. 要了解 INTWDT2 中断时的恢复情况, 请参阅 14.2.2 (2) 由 INTWDT2 中断请求信号恢复。

2. n = 0H 至 FH

表 14-1. 中断源列表 (2/2)

类型	分类	默认优先级	名称	触发方式	产生的部件	异常代码	处理程序地址	PC 恢复值	中断控制寄存器
可屏蔽	中断	22	INTTP2CC1	TMP2 捕获 1/比较 1 匹配	TMP2	01E0H	000001E0H	nextPC	TP2CCIC1
		23	INTTP3OV	TMP3 溢出	TMP3	01F0H	000001F0H	nextPC	TP3OVIC
		24	INTTP3CC0	TMP3 捕获 0/比较 0 匹配	TMP3	0200H	00000200H	nextPC	TP3CCIC0
		25	INTTP3CC1	TMP3 捕获 1/比较 1 匹配	TMP3	0210H	00000210H	nextPC	TP3CCIC1
		26	INTTM0EQ0	TMM0 比较 匹配	TMM0	0220H	00000220H	nextPC	TM0EQIC0
		27	INTCB0R	CSIB0 接收完成	CSIB0	0230H	00000230H	nextPC	CB0RIC
		28	INTCB0T	CSIB0 连续发送写允许	CSIB0	0240H	00000240H	nextPC	CB0TIC
		29	INTCB1R	CSIB1 接收完成	CSIB1	0250H	00000250H	nextPC	CB1RIC
		30	INTCB1T	CSIB1 连续发送写允许	CSIB1	0260H	00000260H	nextPC	CB1TIC
		31	INTUA0R	UARTA0 接收完成	UARTA0	0270H	00000280H	nextPC	UA0RIC
		32	INTUA0T	UARTA0 发送允许	UARTA0	0280H	00000280H	nextPC	UA0TIC
		33	INTUA1R	UARTA1 接收完成/UARTA1 接收错误	UARTA1	0290H	00000290H	nextPC	UA1RIC
		34	INTUA1T	UARTA1 发送允许	UARTA1	02A0H	000002A0H	nextPC	UA1TIC
		35	INTAD	A/D 转换完成	A/D	02BH	000002B0H	nextPC	ADIC
		36	INTKR	按键返回中断请求	KR	0300H	00000300H	nextPC	KRIC
		37	INTWTI	钟表定时器间隔	WT	0310H	00000310H	nextPC	WTIIC
		38	INTWT	钟表定时器参考时间	WT	0320H	00000320H	nextPC	WTIC

- 备注**
- 默认优先级: 在两个或两个以上可屏蔽中断请求同时发生情况时采用默认优先级, 中断优先级 0 的优先级最高。  
不可屏蔽中断优先权顺序为 INTWDT2 > NMI.
  - PC 恢复值: 当中断服务开始时, 程序计数器(PC)的值保存到 EIPC, FEPC, 或 DBPC 中。但是, 要注意的是, 在不可屏蔽中断或可屏蔽中断响应期间, 如果执行了下面任何一条指令, 恢复的 PC 值并不是 nextPC 值(如果在中断执行, 中断停止期间, 进行了中断响应, 并且在该中断服务完成之后又进行了 PC 恢复)。
    - 装载指令 (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
    - 除法指令 (DIV, DIVH, DIVU, DIVHU)
    - PREPARE, DISPOSE 指令 (只有在堆栈指针更新之前有中断产生的情况下)
  - nextPC: 该 PC 值是中断/异常处理过程下一个处理过程的起始地址值。
  - 当一非法代码异常发生时, 该非法指令的地址为(恢复的 PC - 4)。

## 14.2 不可屏蔽中断

即使在中断禁止状态(DI)下,对不可屏蔽中断也是无条件进行响应的。不可屏蔽中断 NMI 与优先级控制无关,与其它所有中断请求信号相比,具有优先权。

该型号单片机具有以下两种类型的不可屏蔽中断请求信号。

- NMI 引脚输入 (NMI)
- 由看门狗定时器(INTWDT2)溢出产生的不可屏蔽中断请求信号。

NMI 引脚的有效边沿信号可以从以下四种类型中进行选择:“上升沿”,“下降沿”,“上升沿和下降沿”,以及“无边沿检测”。

通过将 PMC0.PMC02 位设置为 1,启用 NMI 引脚功能,同时将 INTF0.INTF02 位和 INTR0.INTR02 位设置为期望的数值,以指定所期望的有效触发边沿。

当设置 WDTM2.WDM21 位和位 WDTM2.WDM20 为“01”时,便可通过看门狗定时器 2 的功能溢出触发产生这种类型的不可屏蔽中断请求信号。

如果有两个或两个以上不可屏蔽中断请求同时发生时,则优先级高的中断请求得到响应服务(优先级低的中断请求信号将被忽略),如下所示。

### INTWDT2 > NMI

如果正在对一个 NMI 中断请求进行服务的同时,又有新的 NMI 或 INTWDT2 请求信号产生,则按照如下方式对这些中断请求进行服务。

#### (1) 如果正在对一个 NMI 进行服务的同时,又有新的 NMI 请求信号产生

不论 PSW.NP 位的值如何,该新的 NMI 中断请求信号将保持等待状态。在当前正在服务的 NMI 执行完成(即在 RETI 指令执行之后)之后,才响应该等待的 NMI 请求信号。

#### (2) 如果正在对一个 NMI 进行服务的同时,又有新的 INTWDT2 请求信号产生

如果在 NMI 服务执行期间, NP 位保持设置为(1),则该新的 INTWDT2 请求信号将保持等待状态。在当前正在服务的 NMI 执行完成(即在 RETI 指令执行之后)之后,才响应该等待的 INTWDT2 请求信号。

如果在 NMI 服务执行期间, NP 位清为(0),则响应执行该新近产生的 INTWDT2 请求信号(停止当前正在服务的 NMI)。

**注意事项** 要了解不可屏蔽中断服务执行不可屏蔽中断请求信号 INTWDT2 的相关情况,请参阅 14.2.2 (2) 从 INTWDT2 信号返回。

图 14-1. 不可屏蔽中断请求信号应答操作 (1/2)

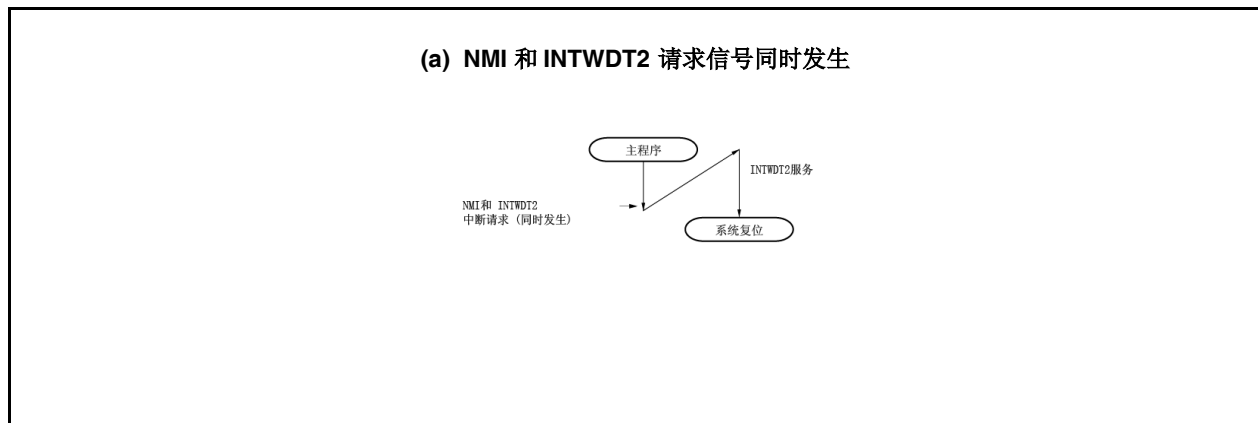
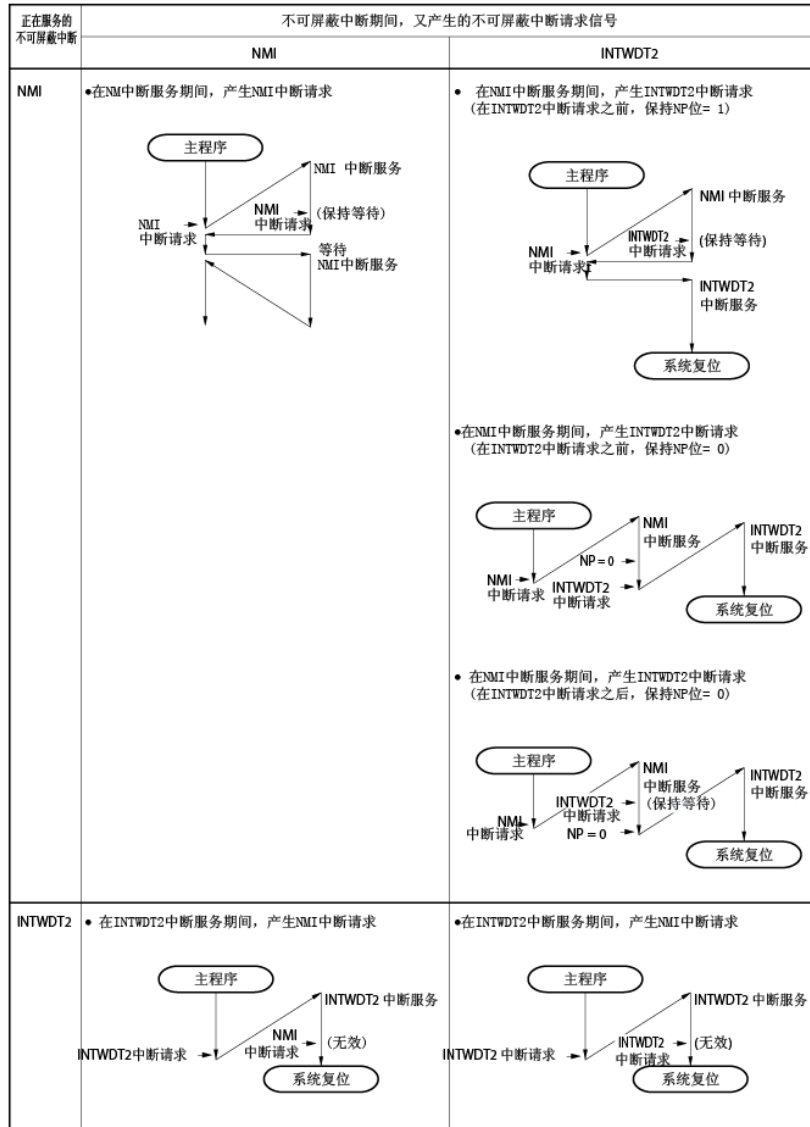


图 14-1. 不可屏蔽中断请求信号应答操作 (2/2)

## (b) 在不可屏蔽中断服务期间，又有不可屏蔽中断请求信号产生



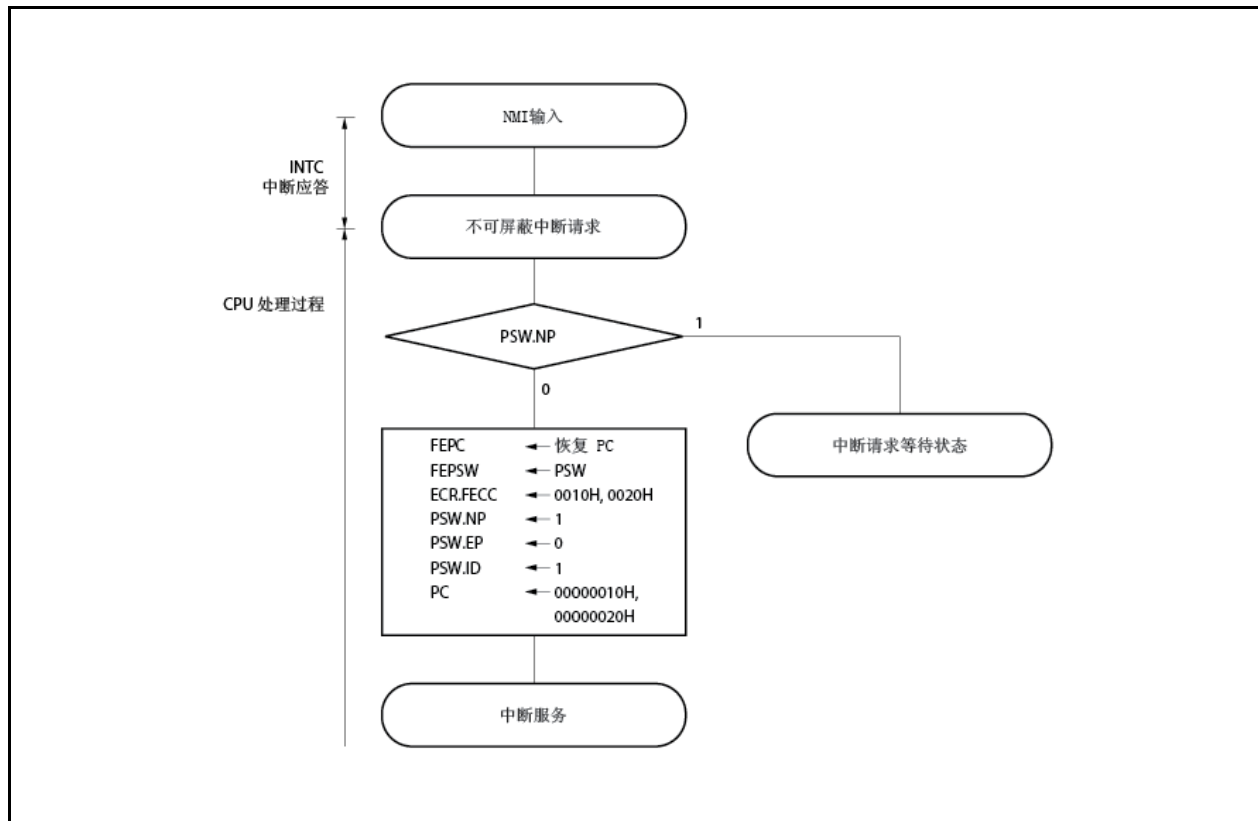
### 14.2.1 操作过程

如果有不可屏蔽中断请求信号产生，CPU 将执行如下处理过程，并将控制权交给该中断服务处理程序。

- <1> 将回存 PC 保存到 FEPC 中去。
- <2> 将当前 PSW 保存到 FEPSW 中去。
- <3> 将异常代码(0010H, 0020H)写入 ECR 的高半字(FECC)中去。
- <4> 将 PSW.NP 位和 PSW.ID 位设置为 1 并将 PSW.EP 位清为 0。
- <5> 将 PC 值设置成为与该不可屏蔽中断相对应的中断处理程序地址(00000010H, 00000020H)。

图 14-2 显示了不可屏蔽中断的服务流程。

图 14-2. 不可屏蔽中断服务配置



14.2.2 恢复

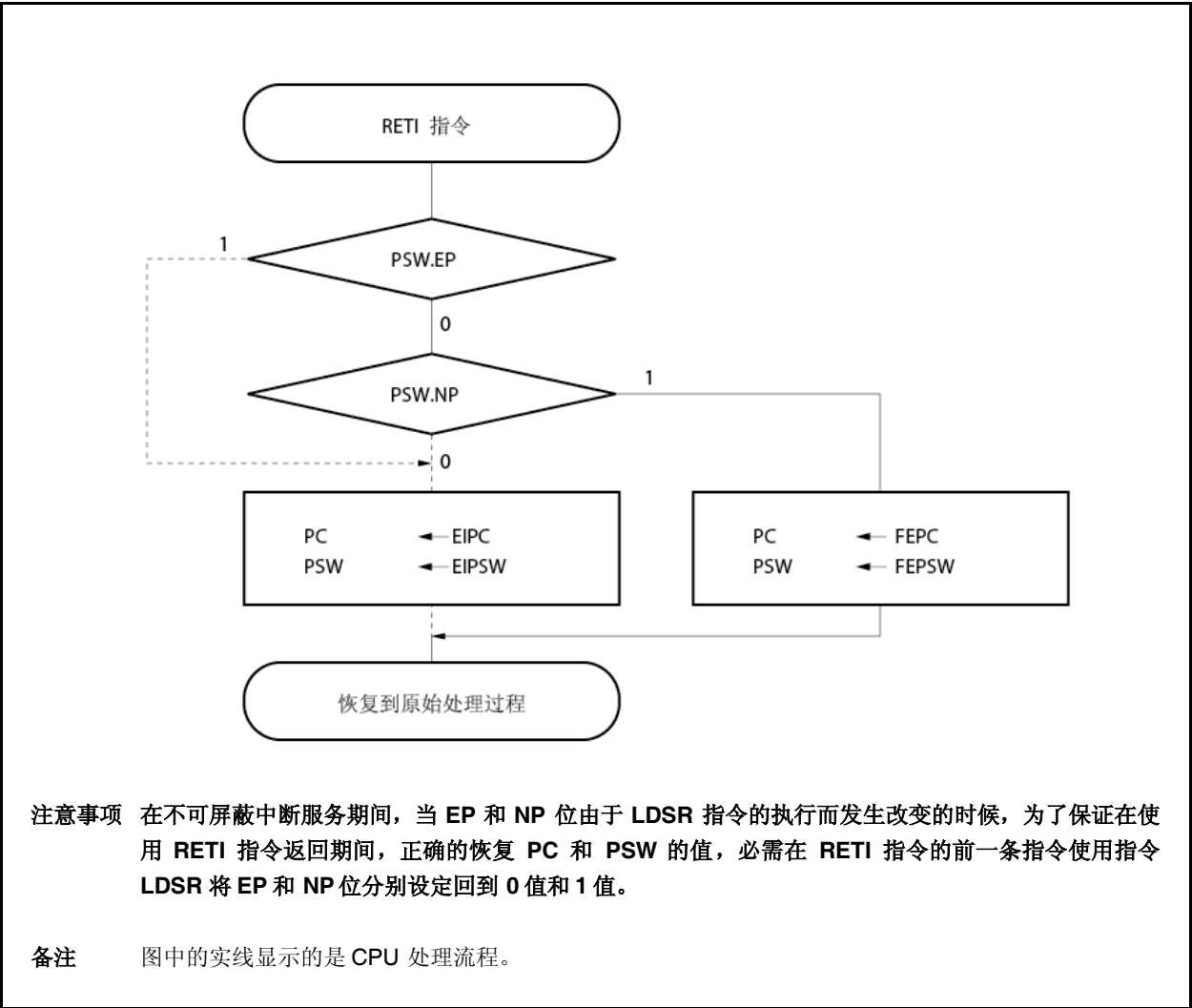
(1) 从 NMI 引脚输入的中断返回

通过 RETI 指令，从 NMI 中断服务返回程序执行。  
当执行 RETI 指令时，CPU 将执行如下处理过程，并将控制转到回存 PC 的地址处。

- <1>分别从 FEPC 和 FEPSW 装载恢复回存 PC 和 PSW，因为 PSW.EP 位为 0，PSW.NP 位为 1。
- <2> 将程序控制返回到回存 PC 和 PSW 所保存的地址处。

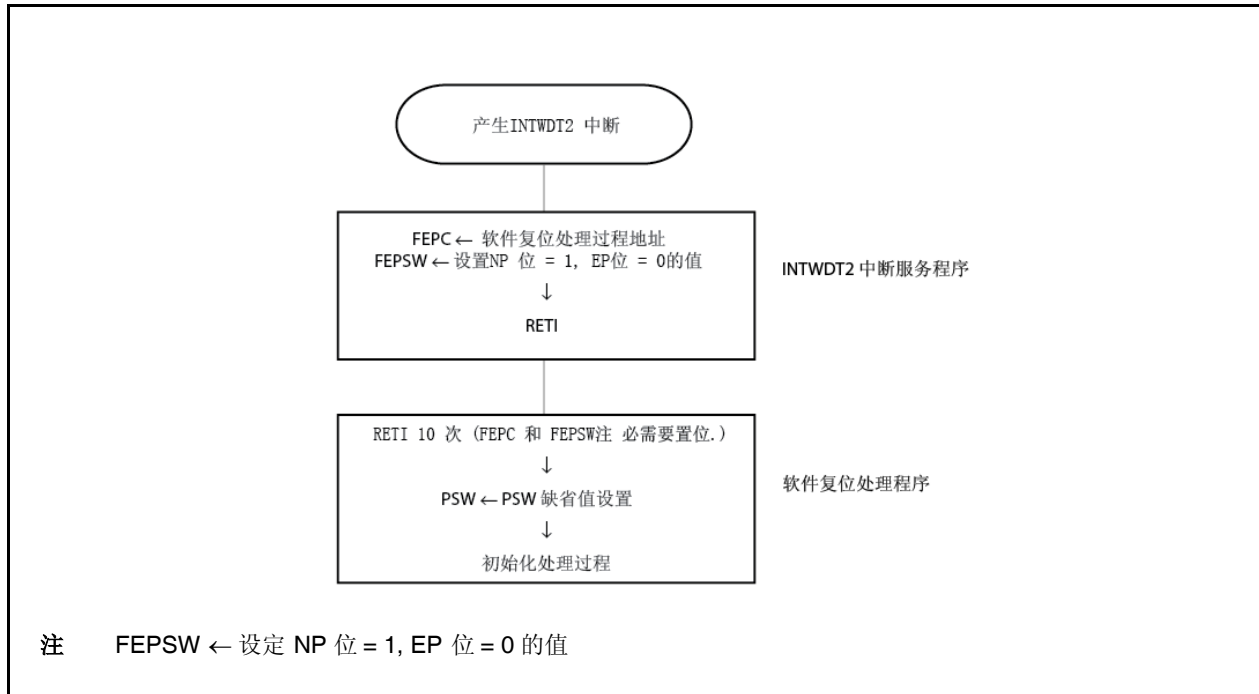
图 14-3 说明了 RETI 指令的处理过程。

图 14-3. RETI 指令处理过程



**(2) 从 INTWDT2 信号的中断返回**

禁止使用 RETI 指令，将由不可屏蔽中断请求(INTWDT2)产生的不可屏蔽中断服务，从中断服务程序执行中返回。所以，必需执行如下的软件复位处理过程进行中断返回。

**图 14-4. 软件复位处理过程****14.2.3 NP 标志**

该 NP 标志显示正在执行不可屏蔽中断服务。

当响应不可屏蔽中断时，该位被置位，并屏蔽不可屏蔽中断以禁止响应多个中断请求。

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	不可屏蔽中断服务状态
0	没有不可屏蔽中断服务过程
1	当前正在服务不可屏蔽中断



### 14.3 可屏蔽中断

可屏蔽中断请求信号可由中断控制寄存器加以屏蔽。V850ES/HF2 单片机共有 39 个可屏蔽中断源。

如果有两个或两个以上可屏蔽中断请求同时发生，那么将按照默认优先级进行响应。另外，除了该默认优先级以外，还可以通过使用中断控制寄存器来指定八级中断优先权(可编程中断优先权控制)。

当 CPU 已经对一个中断请求信号进行响应时，就禁止响应其他可屏蔽中断请求信号，同时将中断禁止状态位(DI)置位。

在中断服务程序中，当执行了 EI 指令时，中断允许状态位(EI)也就被置位。这样就允许对比正在处理的中断请求信号具有更高优先级的中断进行中断服务(由中断控制寄存器进行设定)。要注意的是，只有高优先级的中断才有这种能力，同级中断不能嵌套。

然而，为了允许多重中断，必需在执行 EI 指令之前，要将 EIPC 和 EIPSW 中的内容保存到存储器或通用寄存器中去，同时，在指令 RETI 之前，应执行 DI 指令，以恢复 EIPC 和 EIPSW 的原始数值。

#### 14.3.1 操作过程

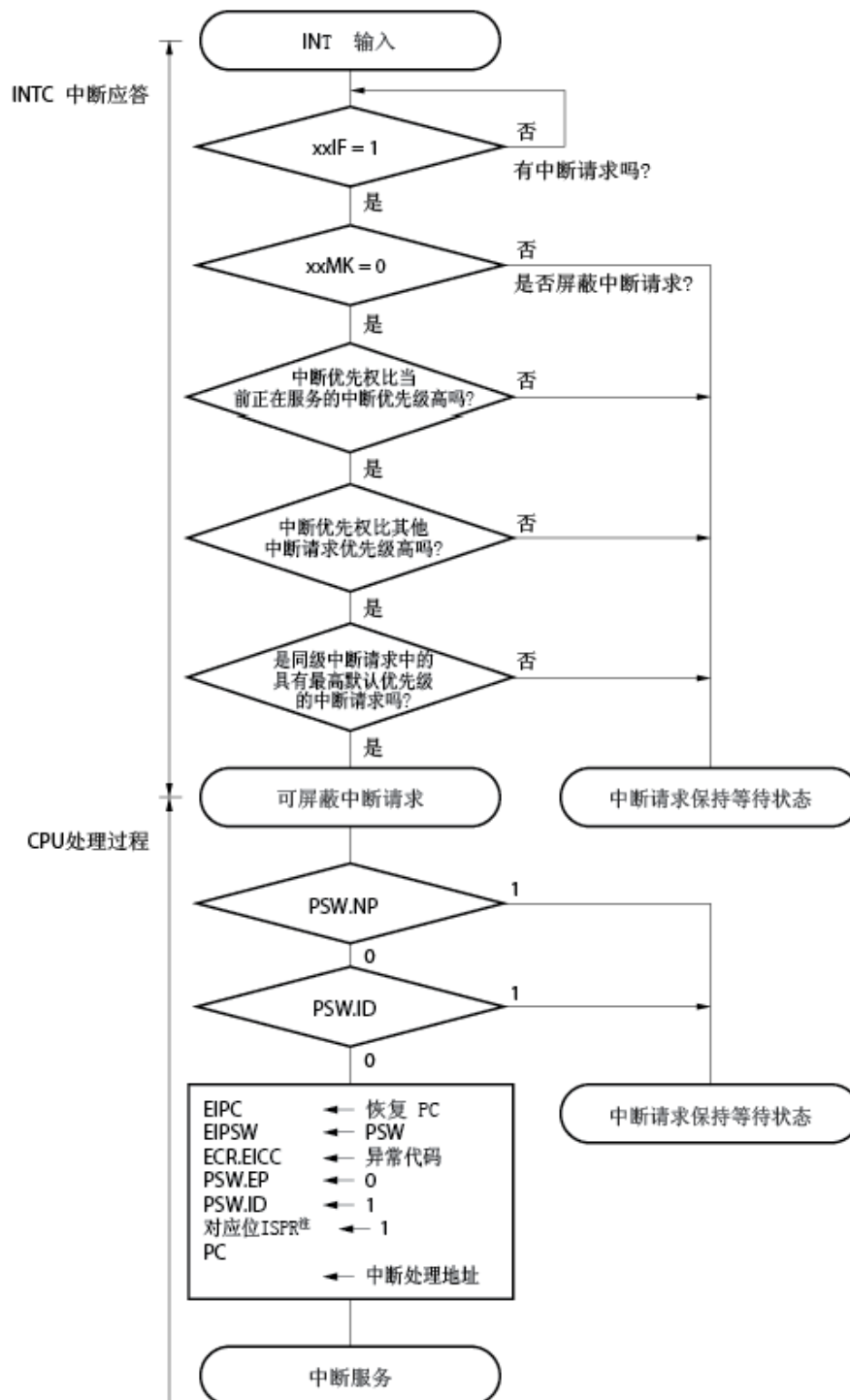
如果有可屏蔽中断请求产生，CPU 将执行如下处理过程，并将控制权交给该中断服务处理程序。

- <1> 将回存 PC 保存到 EIPC 中去。
- <2> 将当前 PSW 保存到 EIPSW 中去。
- <3> 将异常代码(0010H, 0020H)写入 ECR 的低半字(EICC)中去。
- <4> 将 PSW.ID 位设置为 1，将 PSW.EP 位清为 0。
- <5> 将与每个中断相对应的中断处理程序地址设置到 PC 中去，传递程序执行控制权。

通过 INTC 屏蔽的可屏蔽中断请求信号以及在另外一个中断服务期间产生的可屏蔽中断请求信号(当 PSW.NP 位 = 1 或 PSW.ID 位 = 1 时)，均在 INTC 内处于保持等待状态。在这种情况下，如果可屏蔽中断请求信号取消屏蔽，或者通过使用 RETI 或 LDSR 指令将 NP 位和 ID 位清 0，便可根据这些等待的可屏蔽中断请求信号的优先级，开始对一个新的可屏蔽中断请求信号执行中断服务。

下图显示了可屏蔽中断的服务过程。

图 14-5. 可屏蔽中断服务过程



注 要了解 ISPR 寄存器相关内容，请参阅 14.3.6 服务优先权寄存器 (ISPR)。

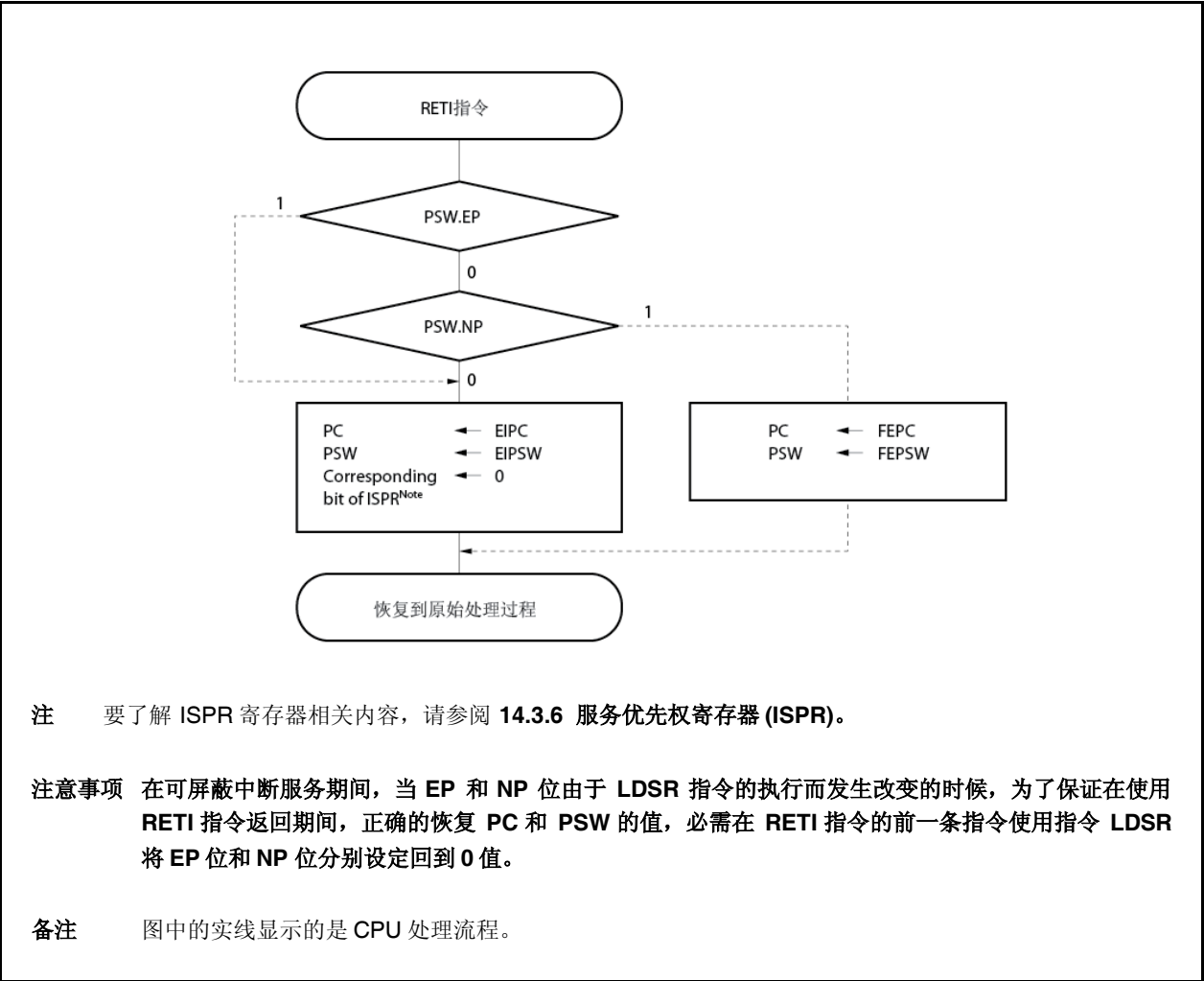
14.3.2 恢复

通过 RETI 指令来执行可屏蔽中断服务的恢复。  
当执行 RETI 指令时，CPU 将执行如下几个步骤，并将控制转到回存 PC 的地址处。

- <1> 分别从 EIPC 和 EIPSW 装载回存 PC 和 PSW，因为 PSW.EP 位为 0，PSW.NP 位为 0。
- <2> 将程序控制返回到回存 PC 和 PSW 所保存的地址处。

图 14-6 说明了 RETI 指令的处理过程。

图 14-6. RETI 指令处理过程



14.3.3 可屏蔽中断的优先权

在另一个中断正在服务期间，又进行中断响应的多重中断服务，由 INTC 控制执行。可由中断优先级来控制多重中断。

中断优先级控制有两种类型：一种是基于默认优先级的控制，一种是可编程优先级控制，这种优先级控制是通过设定中断控制寄存器(XXICn)的中断优先级设定位(XXPRn)来进行优先级指定的。当有两个或两个以上具有同样优先级（由 XXPRn 位设定）的中断请求信号产生时，那么将会根据每个中断请求信号事先所分配好的中断请求类型(默认优先级)，按顺序对这些中断请求信号进行中断服务。要了解更多信息，请参阅表 14-1 中断/异常源列表。通过对优先级设定标志的设置，可编程优先控制将中断请求信号设定为八级优先等级。

注意，当对一个中断请求信号进行响应时，PSW.ID 标志自动被设置为 1。因此，当使用多重中断时，必需预先将 ID 标志清为 0(比如，通过在中断服务程序中放置一条 EI 指令来实现)，以设置中断允许模式。

备注      xx: 各外围设备的标识名 (请参见 表 14-2 中断控制寄存器 (XXICn))  
            n: 外围设备号 (请参见 表 14-2 中断控制寄存器 (XXICn))。

图 14-7. 在对一个中断服务期间，又有另外一个中断请求信号产生时的处理过程示例 (1/2)

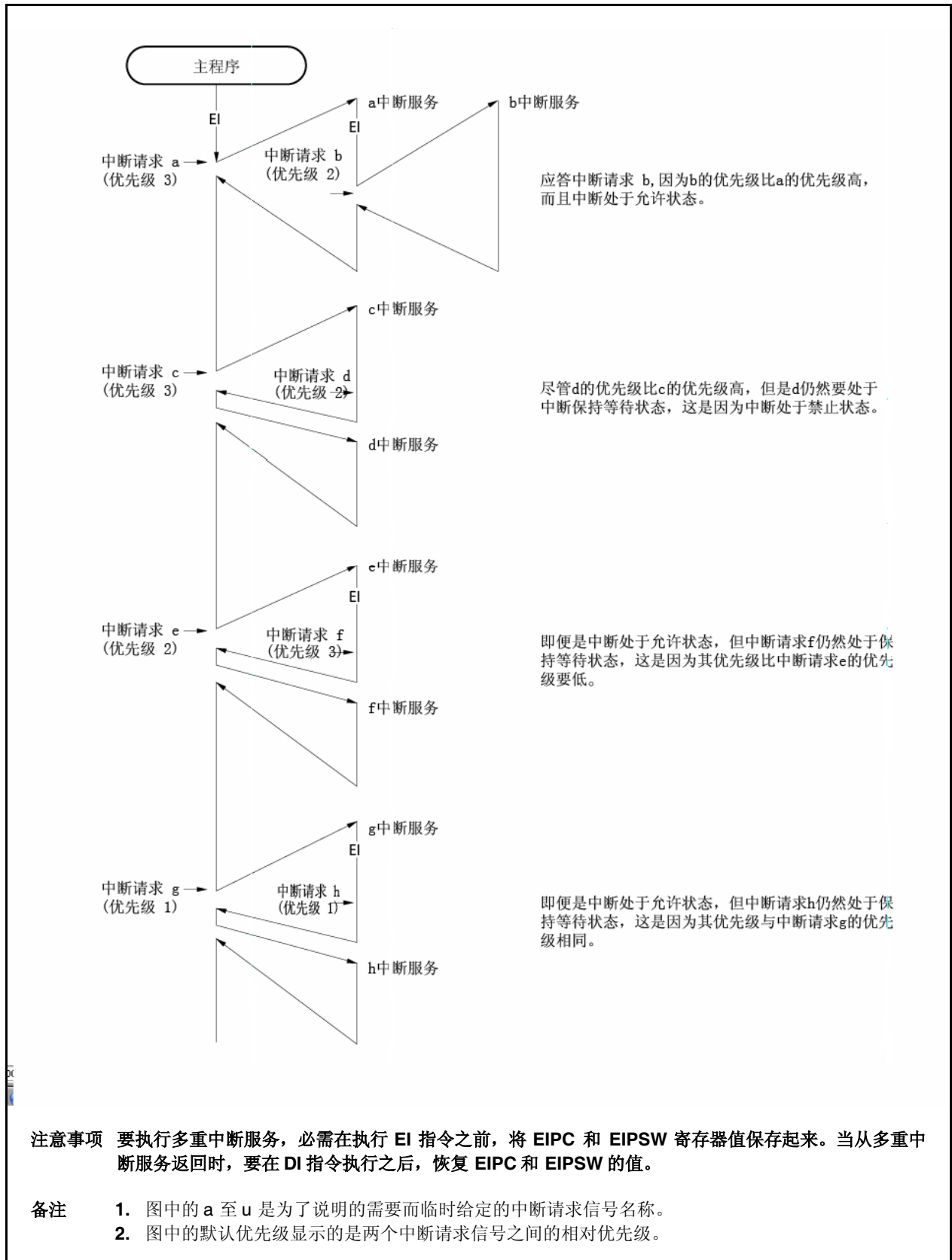


图 14-7. 在对一个中断服务期间，又有另外一个中断请求信号产生时的处理过程示例 (2/2)

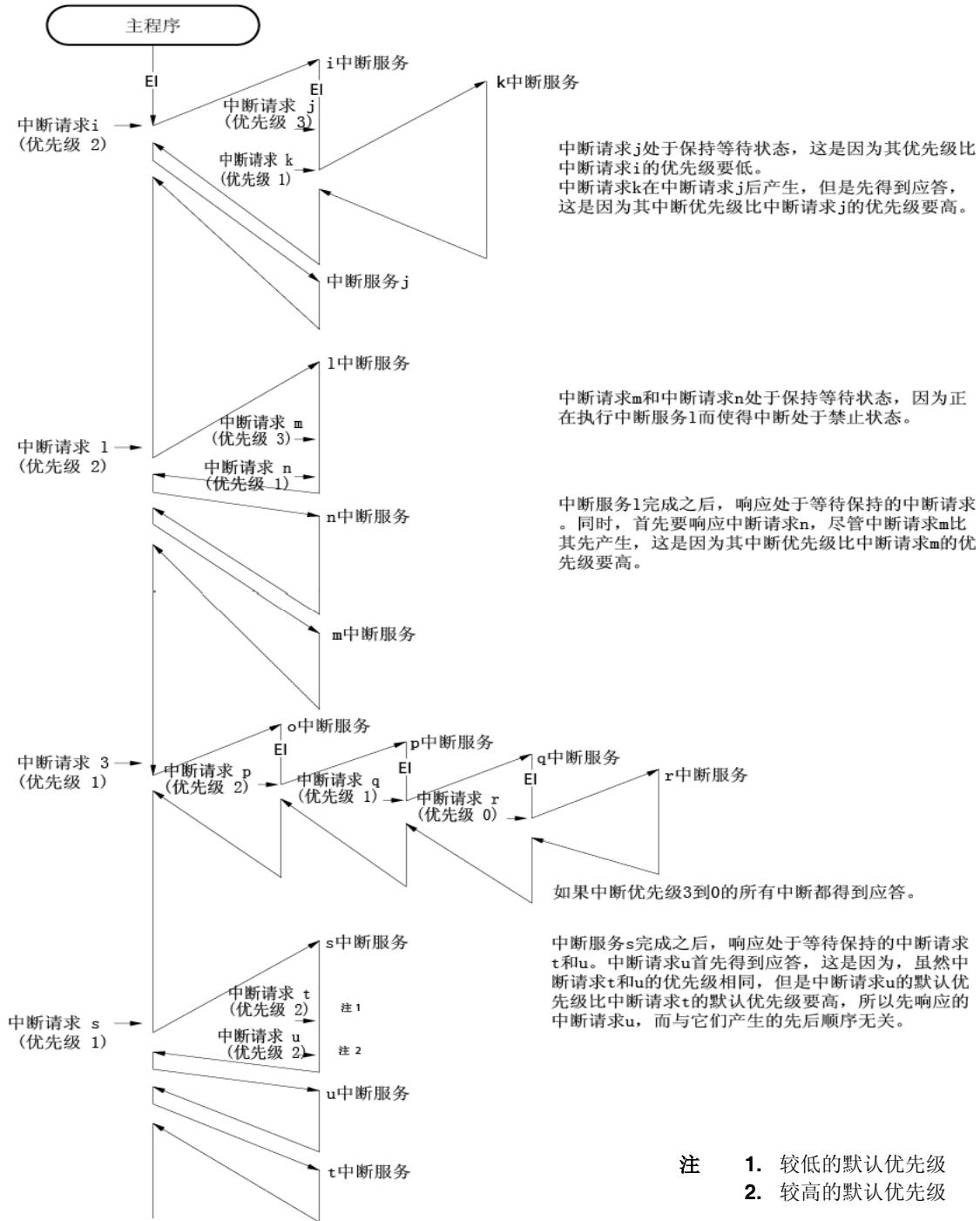
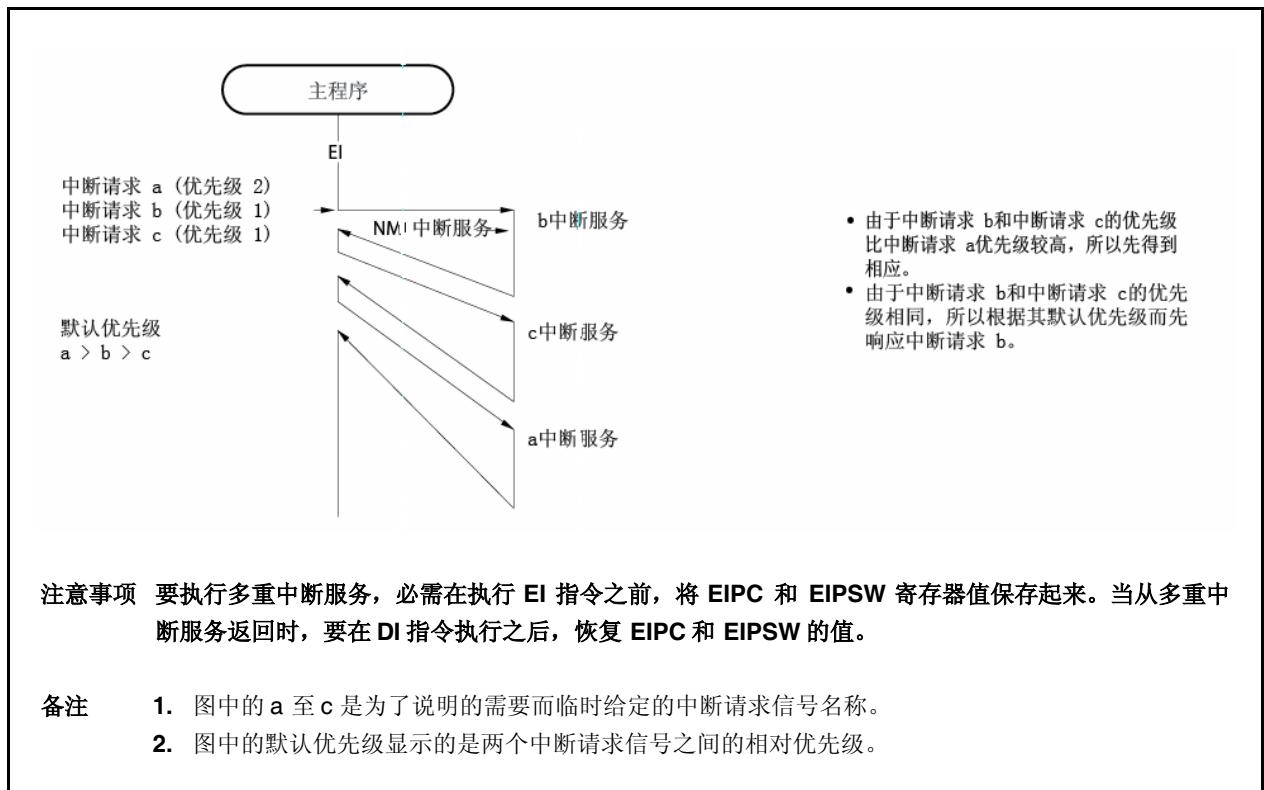


图 14-8. 多个中断请求信号同时产生时的中断服务处理过程示例



#### 14.3.4 中断控制寄存器 (xxICn)

xxICn 寄存器分配给个中断请求信号(可屏蔽中断)，并设置各可屏蔽中断请求的控制条件。

该寄存器可以以 8 位或 1 位进行读或写操作。

复位时，该寄存器被设置为 47H。

**注意事项** 要对 xxICn.xxIFn 位进行读取，请禁止中断(DI)或将中断屏蔽。如果在中断允许(EI)期间或未屏蔽对应中断而读取 xxIFn 位，当中断响应和读取该 xxIFn 位发生冲突时，可能读不到正确的数值。

复位后：47H R/W 地址：FFFFF110H to FFFFF164H

	7	6	5	4	3	2	1	0
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	中断请求标志位 <sup>注</sup>
0	没有中断请求发生
1	有中断请求发生

xxMKn	中断屏蔽标志位
0	中断服务允许
1	中断服务禁止(等待)

xxPRn2	xxPRn1	xxPRn0	中断优先级指定位
0	0	0	指定优先级为0级(最高)
0	0	1	指定优先级为1级
0	1	0	指定优先级为2级
0	1	1	指定优先级为3级
1	0	0	指定优先级为4级
1	0	1	指定优先级为5级
1	1	0	指定优先级为6级
1	1	1	指定优先级为7级(最低)

**注** 如果响应了一个中断请求信号，xxIFn 标志便由硬件自动复位。

**备注** xx: 各外围设备的标识名 (请参见 表 14-2 中断控制寄存器 (xxICn))

n: 外围设备号 (请参见 表 14-2 中断控制寄存器 (xxICn))。

中断控制寄存器的地址和位含义如下所示。



表 14-2. 中断控制寄存器 (xxICn)

地址	寄存器	位							
		7	6	5	4	3	2	1	0
FFFFF110H	LVIIIC	LVIIIF	LVIMK	0	0	0	LVIPR2	LVIPR1	LVIPR0
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
FFFFF120H	PIC7	PIF7	PMK7	0	0	0	PPR72	PPR71	PPR70
FFFFF122H	TQ0OVIC	TQ0OVIF	TQ0OVMK	0	0	0	TQ0OVPR2	TQ0OVPR1	TQ0OVPR0
FFFFF124H	TQ0CCIC0	TQ0CCIF0	TQ0CCMK0	0	0	0	TQ0CCPR02	TQ0CCPR01	TQ0CCPR00
FFFFF126H	TQ0CCIC1	TQ0CCIF1	TQ0CCMK1	0	0	0	TQ0CCPR12	TQ0CCPR11	TQ0CCPR10
FFFFF128H	TQ0CCIC2	TQ0CCIF2	TQ0CCMK2	0	0	0	TQ0CCPR22	TQ0CCPR21	TQ0CCPR20
FFFFF12AH	TQ0CCIC3	TQ0CCIF3	TQ0CCMK3	0	0	0	TQ0CCPR32	TQ0CCPR31	TQ0CCPR30
FFFFF12CH	TP0OVIC	TP0OVIF	TP0OVMK	0	0	0	TP0OVPR2	TP0OVPR1	TP0OVPR0
FFFFF12EH	TP0CCIC0	TP0CCIF0	TP0CCMK0	0	0	0	TP0CCPR02	TP0CCPR01	TP0CCPR00
FFFFF130H	TP0CCIC1	TP0CCIF1	TP0CCMK1	0	0	0	TP0CCPR12	TP0CCPR11	TP0CCPR10
FFFFF132H	TP1OVIC	TP1OVIF	TP1OVMK	0	0	0	TP1OVPR2	TP1OVPR1	TP1OVPR0
FFFFF134H	TP1CCIC0	TP1CCIF0	TP1CCMK0	0	0	0	TP1CCPR02	TP1CCPR01	TP1CCPR00
FFFFF136H	TP1CCIC1	TP1CCIF1	TP1CCMK1	0	0	0	TP1CCPR12	TP1CCPR11	TP1CCPR10
FFFFF138H	TP2OVIC	TP2OVIF	TP2OVMK	0	0	0	TP2OVPR2	TP2OVPR1	TP2OVPR0
FFFFF13AH	TP2CCIC0	TP2CCIF0	TP2CCMK0	0	0	0	TP2CCPR02	TP2CCPR01	TP2CCPR00
FFFFF13CH	TP2CCIC1	TP2CCIF1	TP2CCMK1	0	0	0	TP2CCPR12	TP2CCPR11	TP2CCPR10
FFFFF13EH	TP3OVIC	TP3OVIF	TP3OVMK	0	0	0	TP3OVPR2	TP3OVPR1	TP3OVPR0
FFFFF140H	TP3CCIC0	TP3CCIF0	TP3CCMK0	0	0	0	TP3CCPR02	TP3CCPR01	TP3CCPR00
FFFFF142H	TP3CCIC1	TP3CCIF1	TP3CCMK1	0	0	0	TP3CCPR12	TP3CCPR11	TP3CCPR10
FFFFF144H	TM0EQIC0	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR02	TM0EQPR01	TM0EQPR00
FFFFF146H	CB0RIC	CB0RIF	CB0RMK	0	0	0	CB0RPR2	CB0RPR1	CB0RPR0
FFFFF148H	CB0TIC	CB0TIF	CB0TMK	0	0	0	CB0TPR2	CB0TPR1	CB0TPR0
FFFFF14AH	CB1RIC	CB1RIF	CB1RMK	0	0	0	CB1RPR2	CB1RPR1	CB1RPR0
FFFFF14CH	CB1TIC	CB1TIF	CB1TMK	0	0	0	CB1TPR2	CB1TPR1	CB1TPR0
FFFFF14EH	UA0RIC	UA0RIF	UA0RMK	0	0	0	UA0RPR2	UA0RPR1	UA0RPR0
FFFFF150H	UA0TIC	UA0TIF	UA0TMK	0	0	0	UA0TPR2	UA0TPR1	UA0TPR0
FFFFF152H	UA1RIC	UA1RIF	UA1RMK	0	0	0	UA1RPR2	UA1RPR1	UA1RPR0
FFFFF154H	UA1TIC	UA1TIF	UA1TMK	0	0	0	UA1TPR2	UA1TPR1	UA1TPR0
FFFFF156H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF160H	KRIC	KRIF	KRMK	0	0	0	KRPR2	KRPR1	KRPR0
FFFFF162H	WTIIC	WTIIF	WTIMK	0	0	0	WTIPR2	WTIPR1	WTIPR0
FFFFF164H	WTIC	WTIF	WTMK	0	0	0	WTPR2	WTPR1	WTPR0

### 14.3.5 中断屏蔽寄存器 0 至 2 (IMR0 至 IMR2)

寄存器 IMR0 至 IMR2 为可屏蔽中断设置中断屏蔽状态。寄存器 IMR0 至 IMR2 的 xxMKn 位与 xxICn.xxMKn 位是等效的。

寄存器 IMRm 可以以 16 位进行读或写操作(m = 0 至 2)。

如果 IMRm 寄存器的高 8 位用作 IMRmH 寄存器, 低 8 位用作 IMRmL 寄存器, 那么这些寄存器便可以以 8 位或 1 位进行读或写操作(m = 0 至 2)。

复位时, 该寄存器被设置为 FFFFH。

**注意事项** 设备文件将 xxICn.xxMKn 位定义为保留字。如果使用 xxMKn 名对寄存器某一位进行位操作, 那么会对 xxICn 寄存器内容进行重写, 而不是 IMRm 寄存器 (作为结果, IMRm 寄存器内容也被重写了)。

复位后: FFFFH		R/W	地址:		IMR2 FFFFF104H, IMR2L FFFFF104H, IMR2H FFFFF105H				
15		14		13	12	11	10	9	8
IMR2 (IMR2H <sup>注</sup> )	1	1	1	1	1	WTMK	WTIMK	KRMK	
7		6		5	4	3	2	1	0
IMR2L	1	1	1	1	ADMK	UA1TMK	UA1RMK	UA0TMK	
复位后: FFFFH		R/W	地址:		IMR1 FFFFF102H, IMR1L FFFFF102H, IMR1H FFFFF103H				
15		14		13	12	11	10	9	8
IMR1 (IMR1H <sup>注</sup> )	UA0RMK	CB1TMK	CB1RMK	CB0TMK	CB0RMK	TM0EQMK0	TP3CCMK1	TP3CCMK0	
7		6		5	4	3	2	1	0
IMR1L	TP3OVMK	TP2CCMK1	TP2CCMK0	TP2OVMK	TP1CCMK1	TP1CCMK0	TP1OVMK	TP0CCMK1	
复位后: FFFFH		R/W	地址:		IMR0 FFFFF100H, IMR0L FFFFF100H, IMR0H FFFFF101H				
15		14		13	12	11	10	9	8
IMR0 (IMR0H <sup>注</sup> )	TP0CCMK0	TP0OVMK	TQ0CCMK3	TQ0CCMK2	TQ0CCMK1	TQ0CCMK0	TQ0OVMK	PMK7	
7		6		5	4	3	2	1	0
IMR0L	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	
xxMKn		设置中断屏蔽标志							
0		中断服务允许							
1		中断服务禁止							

**注** 要以 8 位或 1 位读取 IMR0 至 IMR2 寄存器的 8 至 15 位, 请将这些位设置为 IMR0H 至 IMR2H 寄存器的 0 至 7 位。

**注意事项** 请将 IMR2 寄存器的 15 至 11 位 以及 7 至 4 位设置为“1”。如果更改了对这些位的设置, 并不保证该操作的有效性。

**备注** xx: 各外围设备的标识名 (请参见 表 14-2 中断控制寄存器 (xxICn)).

n: 外围设备号 (请参见 表 14-2 中断控制寄存器 (xxICn))

14.3.6 服务中优先级寄存器 (ISPR)

ISPR 寄存器保存当前正在响应的可屏蔽中断的优先级。当有中断请求信号响应时，该寄存器中与该中断请求信号所对应的优先级位就置为 1，并且在中断服务期间，保持该设置。

当执行 RETI 指令时，该寄存器中与最高优先级中断请求信号所对应的位便会由硬件自动设置为 0。但是，当由不可屏蔽中断服务或异常处理返回时，该位并不复位至 0。

该寄存器为只读寄存器，可以以 8 位或 1 位进行读操作。

复位时，该寄存器被设置为 00H。

**注意事项** 如果在中断允许(EI)状态下读取 ISPR 寄存器的时候，响应了一个中断请求，那么对 ISPR 寄存器的读取值，就有可能的是被中断响应所设置之后的值。因此，为了在中断响应前，准确地读取 ISPR 寄存器值，请在中断禁止(DI)状态下，对该寄存器进行读取。

复位后:00H    R    地址:FFFF1FAH

	7	6	5	4	3	2	1	0
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	当前响应的中断优先级标志
0	中断优先级为n的中断请求信号没有得到响应
1	中断优先级为n的中断请求信号得到响应

备注      n = 0 至 7 (优先级)



14.4 软件异常

软件异常是 CPU 在执行 TRAP 指令时候而产生的，而且始终能够得到响应。

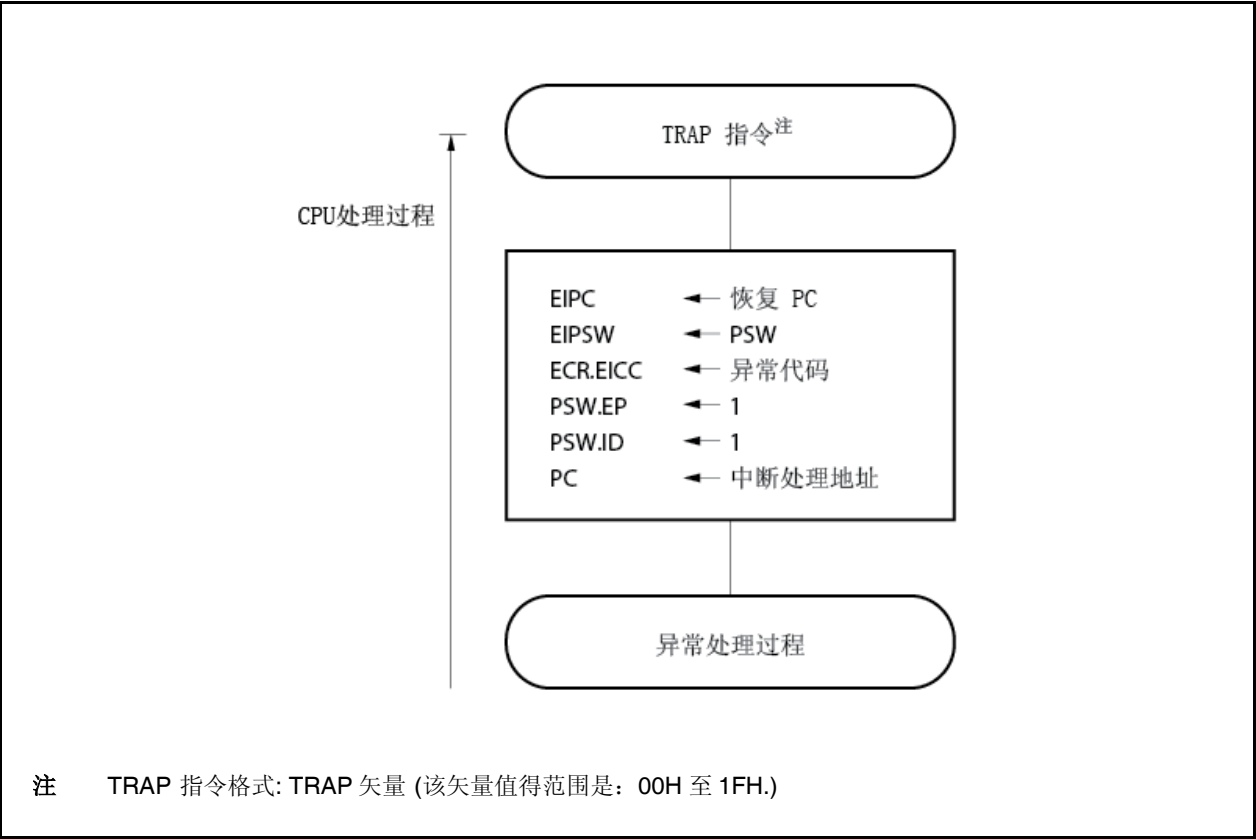
14.4.1 操作过程

如果有软件异常产生，CPU 将执行如下处理过程，并将控制权交给该异常处理程序。

- <1> 将回存 PC 保存到 EIPC 中去。
- <2> 将当前 PSW 保存到 EIPSW 中去。
- <3> 将异常代码(0010H, 0020H)写入 ECR(中断源)的低 16 位(EICC)中去。
- <4> 将 PSW. EP 位和 PSW.ID 位设置为 1。
- <5> 将 PC 值设置成为与该软件异常相对应的处理程序地址(00000040H 或 00000050H)，传递程序执行控制权。

图 14-9 说明了软件异常的处理过程。

图 14-9. 软件异常处理过程



处理程序地址由 TRAP 指令的操作码(矢量)确定。如果该矢量为 00H 至 0FH，地址为 00000040H，如果该矢量为 10H 至 1FH，地址为 00000050H。

### 14.4.2 恢复

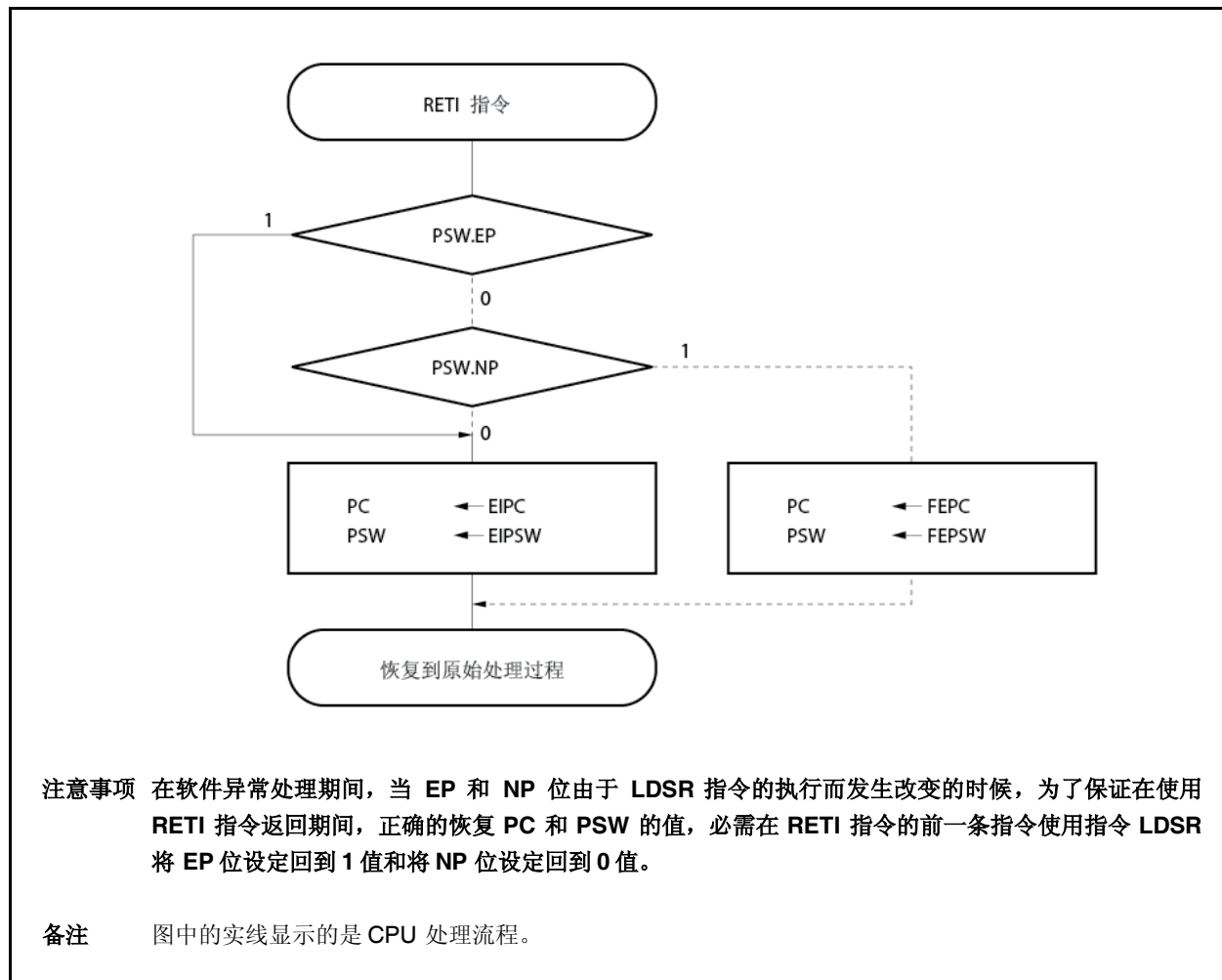
通过 **RETI** 指令来执行软件异常的恢复。

当执行 **RETI** 指令时，CPU 将执行如下处理过程，并将控制转到回存 PC 的地址处。

- <1> 分别从 **EIPC** 和 **EIPSW** 装载回存 PC 和 PSW，因为 **PSW.EP** 位为 1。
- <2> 将程序控制返回到回存 PC 和 PSW 所保存的地址处。

图 14-10 说明了 **RETI** 指令的处理过程。

图 14-10. RETI 指令处理过程



14.4.3 EP 标志

该 EP 标志用来显示正在执行异常处理的状态标志。当异常发生时，该位被置位。

复位后:00000020H

PSW

31876543210

0

NP

EP

ID

SAT

CY

OV

S

Z

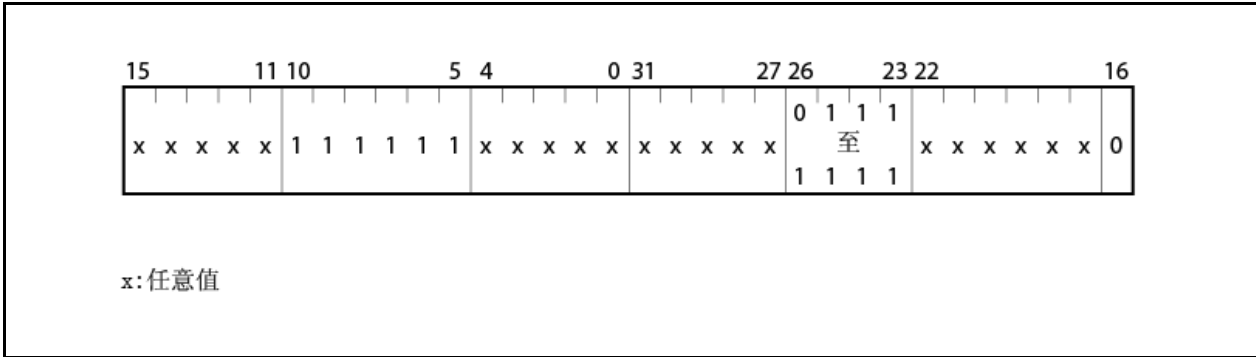
EP	异常处理状态
0	没有进行异常处理
1	异常处理正在进行中

14.5 异常陷阱

异常陷阱是在非法执行指令时，所要求产生的中断。在 V850ES/HF2 单片机中,将非法代码异常(ILGOP: 非法代码陷阱)认为是一个异常陷阱。

14.5.1 非法操作码 定义

非法指令带有一个 111111B 的操作码(10 至 5 位), 一个 0111B 至 1111B 的 sub-opcode (26 至 23 位), 以及一个 0B 的 sub-opcode (位 16)。当执行该非法指令时，便产生一异常陷阱。



**注意事项** 由于有可能在后来将该指令分配给一非法操作代码，所以建议不要使用该指令。

(1) 操作过程

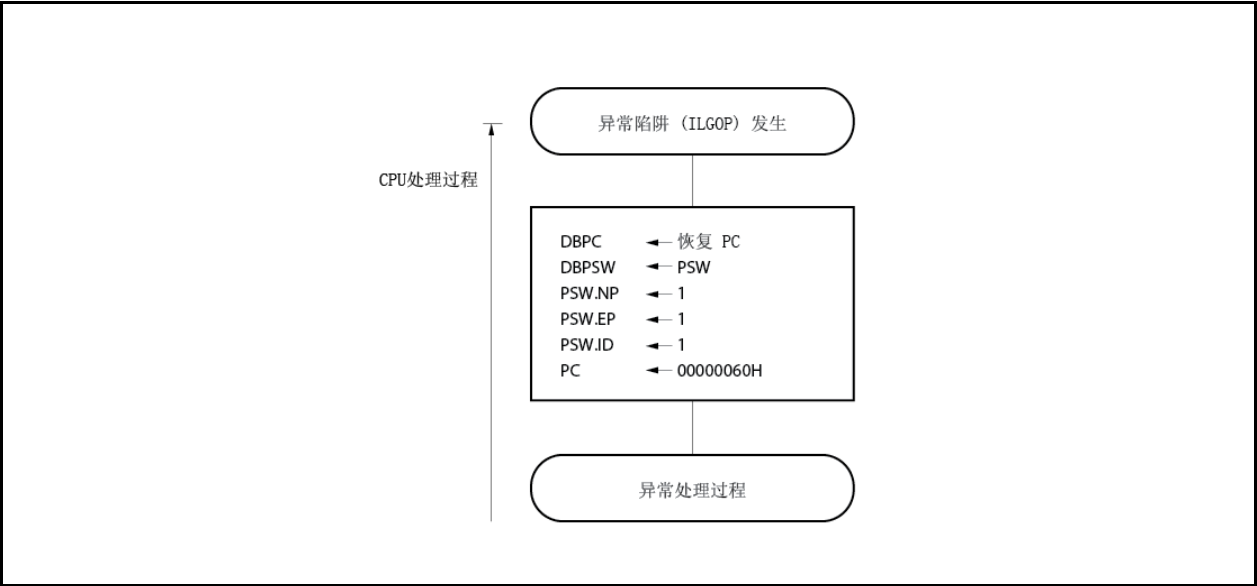
如果有异常陷阱产生，CPU 将执行如下处理过程，并将控制权交给该异常处理程序。

- <1> 将回存 PC 保存到 DBPC 中去。
- <2> 将当前 PSW 保存到 DBPSW 中去。
- <3> 将 PSW.NP, PSW.EP, 以及 PSW.ID 位设置为 1。
- <4> 将 PC 值设置成为与该异常陷阱相对应的处理程序地址(00000060H)，传递程序执行控制权。

图 14-11 说明了异常陷阱的处理过程。



图 14-11. 异常陷阱处理过程



(2) 返回

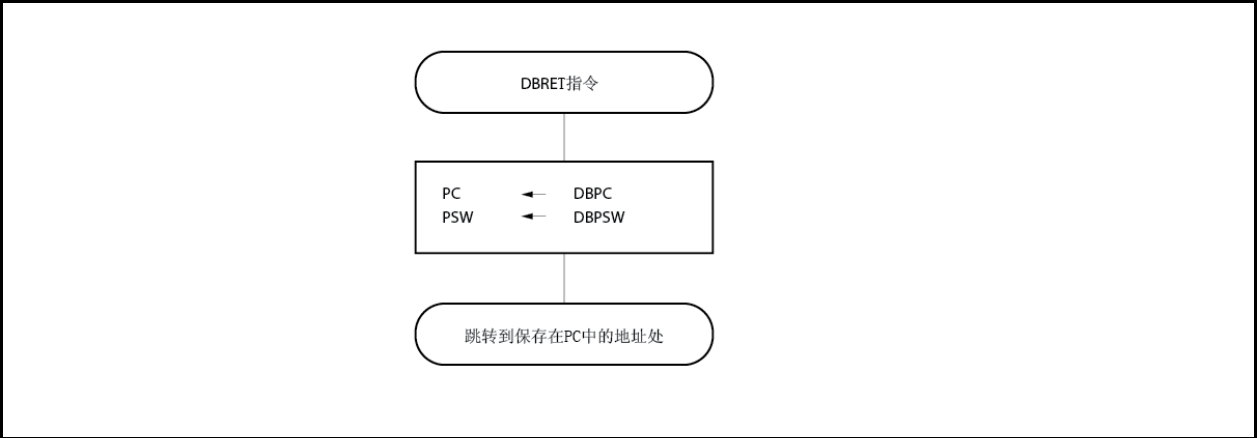
通过 DBRET 指令来执行异常陷阱的返回。通过执行 DBRET 指令，CPU 将执行如下处理过程，并将控制转到回存 PC 的地址处。

- <1>从 DBPC 和 DBPSW 装载回存 PC 和 PSW。
- <2>将程序控制返回到回存 PC 和 PSW 所保存的地址处。

**注意事项** 只有在非法代码和 DBRET 指令执行的间隔期间，才能访问 DBPC 和 DBPSW 寄存器。

图 14-12 说明了由异常陷阱返回的处理过程。

图 14-12. 从异常陷阱恢复的处理过程



14.5.2 调试陷阱

调试陷阱是在执行 DBTRAP 指令时所产生的一个异常，而且始终能够得到响应。

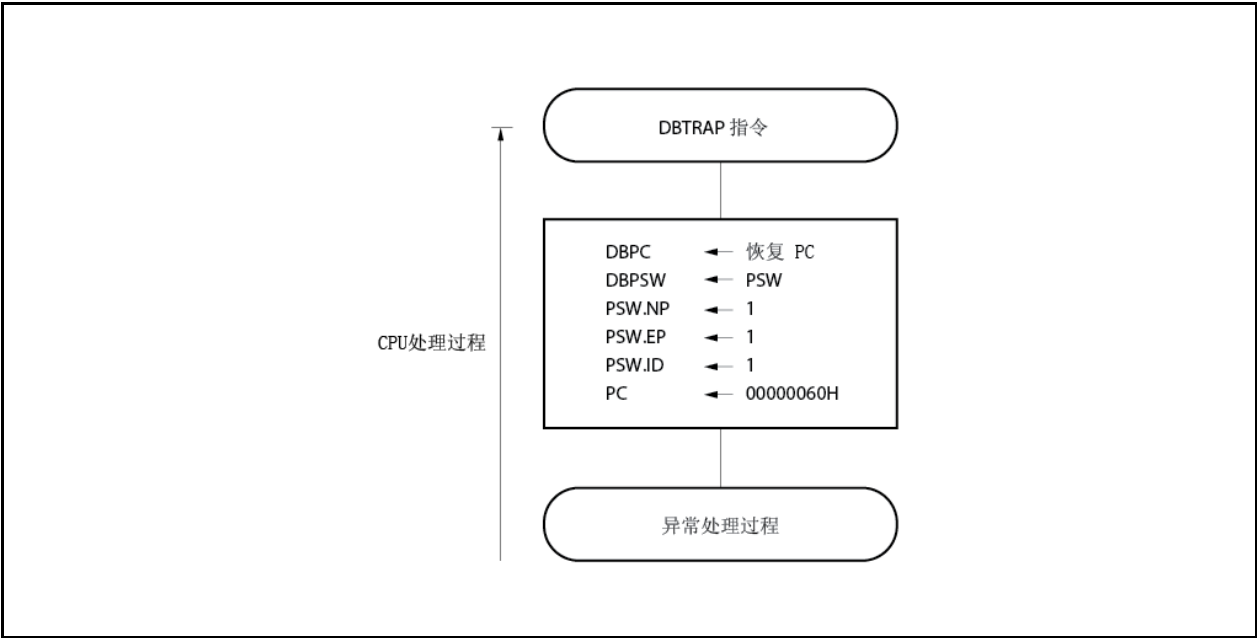
(1) 操作过程

当有调试陷阱产生时，CPU 将执行如下处理过程。

- <1> 将回存 PC 保存到 DBPC 中去。
- <2> 将当前 PSW 保存到 DBPSW 中去。
- <3> 将 PSW.NP, PSW.EP, 以及 PSW.ID 位设置为 1。
- <4> 将 PC 值设置成为与该调试陷阱相对应的处理程序地址(00000060H)，并传递程序执行控制权。

图 14-13 显示了该调试陷阱处理方式。

图 14-13. 调试陷阱处理方式



## (2) 恢复

通过 DBRET 指令来执行调试陷阱的恢复。

通过执行 DBRET 指令，CPU 将执行如下处理步骤，并将控制转到回存 PC 的地址处。

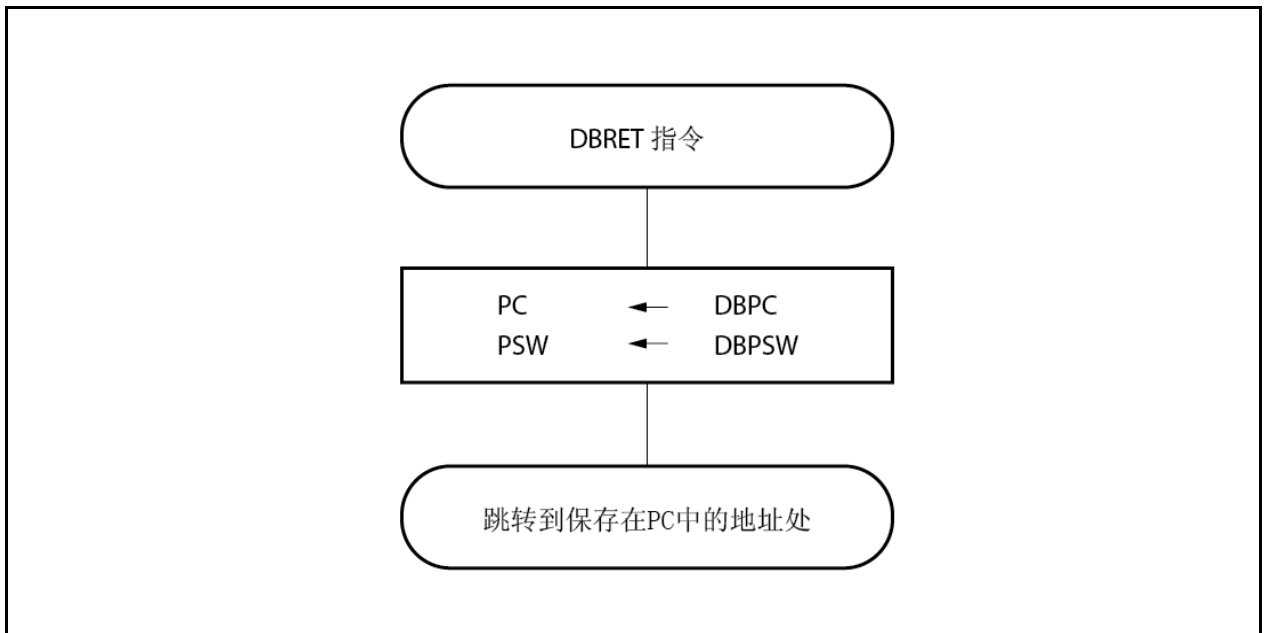
<1> 从 DBPC 和 DBPSW 装载回存 PC 和 PSW。

<2> 将程序控制返回到回存 PC 和 PSW 所保存的地址处。

**注意事项** 只有在 DBTRAP 指令和 DBRET 指令执行的间隔期间，才能访问 DBPC 和 DBPSW 寄存器。

图 14-14 显示了从调试陷阱返回的处理方式。

图 14-14. 从调试陷阱恢复的处理方式



## 14.6 外部中断请求输入引脚 (NMI 以及 INTP0 至 INTP7)

### 14.6.1 噪音消除

#### (1) 有关 NMI 引脚的噪音消除

NMI 引脚带有一个使用模拟延时的内部噪音消除电路。因此，NMI 引脚的输入电平信号不会作为边沿信号而被检测到，除非该输入电平信号保持指定时间或更长时间。因此，在指定时间之后，检测到边沿信号。

NMI 引脚可以用来释放 STOP 模式。在 STOP 模式下，通过使用系统时钟不能消除噪音，这是因为内部系统时钟停止了。

#### (2) 有关 INTP0 至 INTP7 引脚的噪音消除

INTP0 至 INTP7 各引脚均带有一个使用模拟延时的内部噪音消除电路。因此，NMI 引脚的输入电平信号不会作为边沿信号而被检测到，除非该输入电平信号保持指定时间或更长时间。因此，在指定时间之后，检测到边沿信号。

### 14.6.2 边沿检测

NMI 以及 INTP0 至 INTP7 各引脚的有效边沿，可以从以下四种类型中进行选择。

- 上升沿
- 下降沿
- 既有上升沿又有下降沿
- 无边沿检测

复位后，不检测引脚的边沿信号。因此，除非使用 INTF0 和 INTR0 寄存器来启用有效边沿(NMI 引脚功能为正常的端口引脚)，否则，不会对中断请求信号进行响应。

**(1) 外部中断下降沿、上升沿设定寄存器 0 (INTF0, INTR0)**

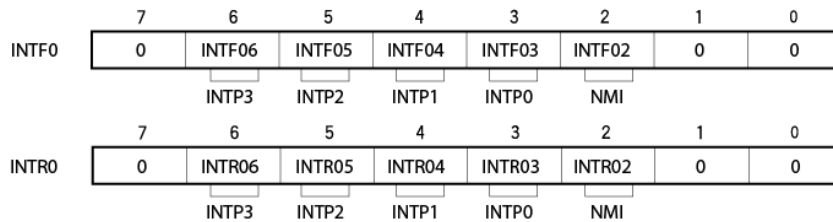
INTF0 和 INTR0 是两个 8 位寄存器，通过其 2 来指定 NMI 引脚的下降沿和上升沿检测，通过其 3 至 6 位来指定外部中断引脚(INTP0 至 INTP3)的下降沿和上升沿检测。

这些寄存器可以以 8 位为单位或 1 位进行读或写操作。

复位时，这些寄存器被设置为 00H。

**注意事项** 当功能由外部中断功能(复用功能)改变成端口功能时，可能会检测到边沿信号。因此，应先将 INTF0n 和 INTR0n 位清为 00，然后再设置为端口功能。

复位后: 00H R/W 地址: INTF0 FFFFC00H, INTR0 FFFFC20H



**备注** 要了解有效边沿设定组合情况，请参阅表 14-3。

**表 14-3. 有效触发边沿的指定**

INTF0n	INTR0n	有效触发边沿的指定 (n = 2 至 6)
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	既有上升沿又有下降沿

**注意事项** 如果相应引脚不用作 NMI 引脚或 INTP0 至 INTP3 引脚时，请确保要将 INTF0n 位和 INTR0n 位清为 00。

**备注** n = 2: NMI 引脚控制  
n = 3 至 6: INTP0 至 INTP3 引脚控制

(2) 外部中断上升沿、下降沿设定寄存器 3L (INTR3L, INTF3L)

INTR3L 和 INTF3L 是两个 8 位寄存器，用来指定 INTP7 引脚的上升沿和下降沿检测。  
这些寄存器可以以 8 位为单位或 1 位进行读或写操作。  
复位时，这些寄存器被设置为 00H。

**注意事项** 当功能由外部中断功能(复用功能)改变成端口功能时，可能会检测到边沿信号。因此，应先将 INTF31 和 INTR31 位清为 00，然后再设置为端口模式。



**备注**        要了解有效边沿设定组合情况，请参阅 表 14-4.

表 14-4. 有效触发边沿的指定

INTF31	INTR31	有效触发边沿的指定
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	既有上升沿又有下降沿

**注意事项** 如果相应引脚不用作 INTP7 引脚时，请确保要将 INTF31 和 INTR31 位清为 00。

### (3) 外部中断下降沿、上升沿设定寄存器 9H (INTF9H, INTR9H)

INTF9H 和 INTR9H 是两个 8 位寄存器，用来指定外部中断引脚(INTP4 至 INTP6)的下降沿和上升沿检测。这些寄存器可以以 8 位为单位或 1 位进行读或写操作。  
复位时，这些寄存器被设置为 00H。

**注意事项** 当功能由外部中断功能(复用功能)改变成端口功能时，可能会检测到边沿信号。因此，应先将 INTF9n 和 INTR9n 位清为 00，然后再设置为端口模式。

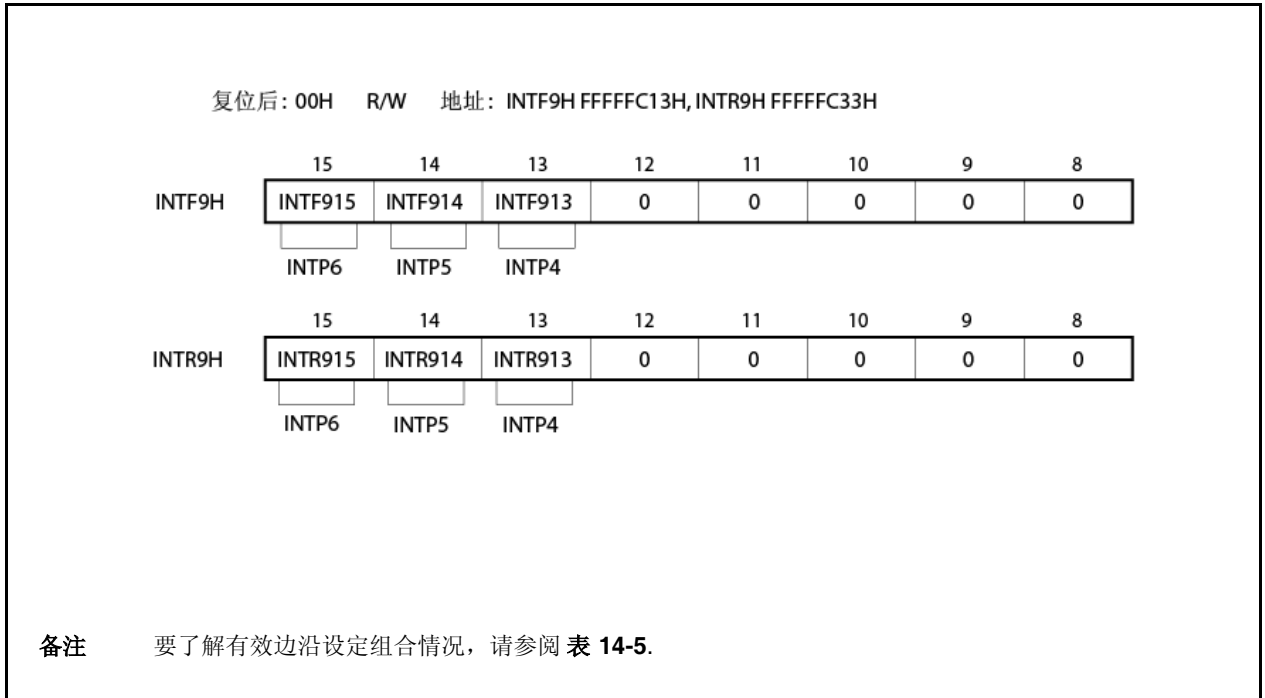


表 14-5. 有效触发边沿的指定

INTF9n	INTR9n	有效触发边沿的指定 (n = 13 to 15)
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	既有上升沿又有下降沿

**注意事项** 如果相应引脚不用作 INTP4 至 INTP6 引脚时，请确保要将 INTF9n 和 INTR9n 位清为 00。

**备注**      n = 13 至 15: INTP4 至 INTP6 引脚控制

**(4) 噪音消除控制寄存器 (NFC)**

可以为 INTP3 引脚选择数字噪音消除功能。通过使用 NFC 寄存器来进行噪音消除的设定。

当选择了数字噪音消除时，数字采样的采样频率可以从以下这些频率中选择： $f_{xx}/64$ ,  $f_{xx}/128$ ,  $f_{xx}/256$ ,  $f_{xx}/512$ ,  $f_{xx}/1,024$ , 以及  $f_{xt}$ 。采样时间由 NFC.NFSTS 位进行设定。

当选择了数字噪音消除时，如果在待机模式下执行采样的时钟停止了，于是，就不能使用 INTP3 中断请求信号来释放该待机模式。当  $f_{xt}$  采样时钟时，INTP3 中断请求信号可以用于释放副时钟操作模式或 IDLE1/IDLE2/STOP/sub-IDLE 模式。

这些寄存器可以以 8 位为单位进行读或写操作。

复位时，该寄存器被设置为 00H。

**注意事项** 采样时钟改变后，到数字噪音消除器初始化，所需要的时间为：采样时钟周期 × 由 NFSTS 位所设定的次数。如果在采样时钟改变后，同时，在“采样时钟周期 × 由 NFSTS 位所设定的次数”时间经过之前，输入了 INTP3 有效边沿触发信号，于是就可能会产生中断请求信号。因此，当使用中断功能时，要注意以下几点。

- 当使用中断功能时，在经过“采样时钟周期 × 由 NFSTS 位所设定的次数”时间之后，在中断请求标志 (PIC3.PIF3 位) 已经清 0 后，要设置中断允许。



复位后：00H R/W 地址：FFFF318H

	7	6	5	4	3	2	1	0
NFC	NFEN	NFSTS	0	0	0	NFC2	NFC1	NFC0

NFEN	设置INTP3引脚噪音消除
0	模拟噪音消除
1	数字噪音消除

NFSTS	数字噪音消除采样次数设置
0	采样次数×3
1	采样次数×2

NFC2	NFC1	NFC0	数字采样时钟
0	0	0	fxx/64
0	0	1	fxx/128
0	1	0	fxx/256
0	1	1	fxx/512
1	0	0	fxx/1,024
1	0	1	fxt (副时钟)
其它值			禁止设置

- 备注
- 1. 由于执行了 3 次采样，所以可靠地消除噪音的时间宽度为 2 个时钟周期。
  - 2. 在噪音宽度小于 2 个采样时钟的情况下，如果噪音与采样时钟同时输入，就会产生中断请求信号。

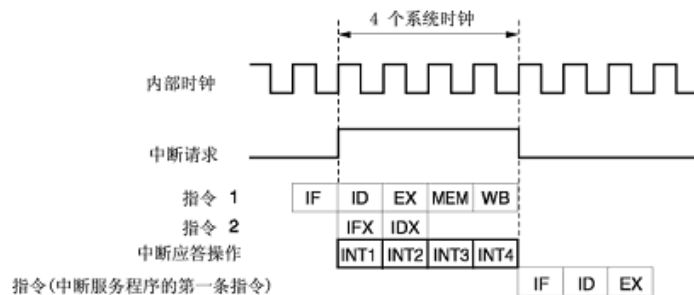
## 14.7 CPU 的中断响应时间

除了以下几种情况之外，CPU 的中断响应时间最小为 4 个时钟周期。要连续输入中断请求信号，前后两次中断请求信号输入时间间隔应至少为 5 个时钟周期。

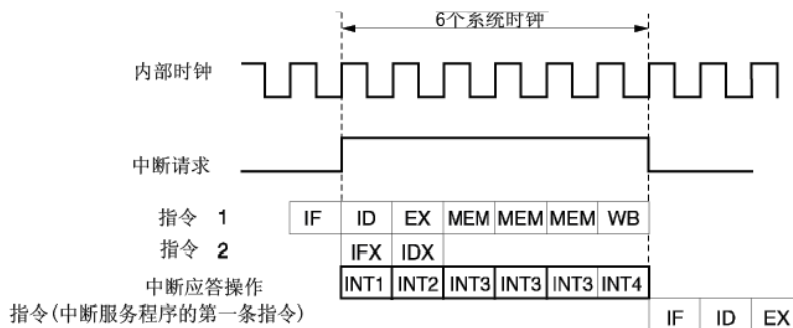
- 在 IDLE1/IDLE2/STOP 模式下
- 当访问外部总线时
- 当连续执行中断请求非采样指令时(请参阅 14.8 CPU 不响应中断的时间阶段。)
- 当访问中断控制寄存器时

图 14-15. 中断请求信号响应时的管线控制 (概要)

### (1) 最小中断响应时间



### (2) 最大中断响应时间



**备注** INT1 至 INT4: 中断响应处理过程  
 IFX: 无效的指令拾取  
 IDX: 无效的指令译码

	中断响应时间 (内部系统时钟)		条件
	内部中断	外部中断	
最小	4	4 + 模拟延迟时间	以下几种情况除外. • 在 IDLE1/IDLE2/STOP 模式下 • 当访问外部总线时 • 当连续执行两个或更多个中断请求非采样指令时 • 访问外围 I/O 寄存器时
最大	6	6 + 模拟延迟时间	

## 14.8 CPU 不响应中断的时间阶段

指令执行时，CPU 响应中断。但是在中断请求非采样指令和下一条指令之间(中断保持等待状态)，CPU 不响应任何中断。

中断请求非采样指令如下。

- EI 指令
- DI 指令
- LDSR reg2, 0x5 指令 (用于 PSW)
- 用于 PRCMD 寄存器的存储指令
- 用于 以下 寄存器的存储, SET1, NOT1, 或 CLR1 指令。
  - 中断相关寄存器:
    - 中断控制寄存器 (xxICn), 中断屏蔽寄存器 0 至 2 (IMR0 至 IMR2)
  - 省电控制寄存器 (PSC)
  - 片上调试模式寄存器 (OCDM)
  - 外围仿真寄存器 1 (PEMU1):

**备注**     xx: 各外围设备的标识名 (请参见 表 14-2 中断控制寄存器 (xxICn))  
          n: 外围设备号 (请参见 表 14-2 中断控制寄存器 (xxICn)).

## 14.9 注意事项

NMI 引脚是 P02 引脚的复用功能引脚。该引脚复位后为一通常端口引脚。因此，要启用 NMI 引脚，请使用 PMC0 寄存器使 NMI 引脚有效。NMI 引脚的初始化设置为“无边沿检测”，请通过使用 INTF0 和 INTRO 寄存器来选择 NMI 引脚的有效触发边沿。

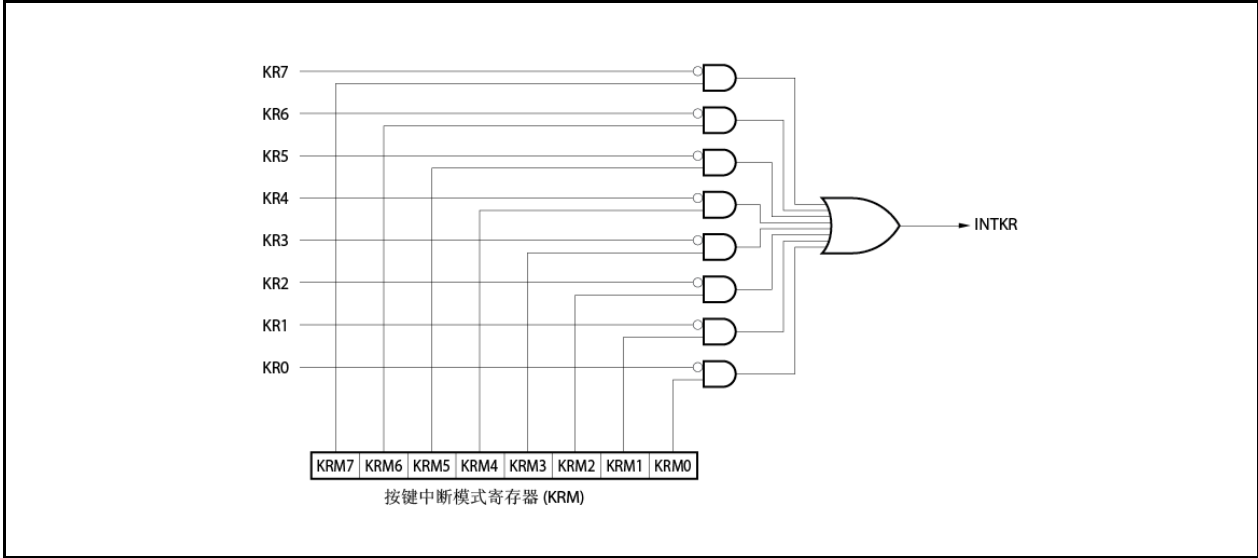
15.1 功能

通过设置 **KRM** 寄存器向八个按键输入引脚（**KR0** 至 **KR7**）输入下降边缘，可产生按键中断请求信号（**INTKR**）。

表 15-1 按键中断检测引脚的分配

标志	引脚描述
KRM0	以 1 位元控制 KR0 信号
KRM1	以 1 位元控制 KR1 信号
KRM2	以 1 位元控制 KR2 信号
KRM3	以 1 位元控制 KR3 信号
KRM4	以 1 位元控制 KR4 信号
KRM5	以 1 位元控制 KR5 信号
KRM6	以 1 位元控制 KR6 信号
KRM7	以 1 位元控制 KR7 信号

图 15-1 按键中断框图



15.2 寄存器

(1) 按键中断模式寄存器（KRM）

KRM 寄存器利用 KR0 至 KR7 信号控制 KRM0 至 KRM7 位。  
该寄存器可进行 8 位或 1 位读写。  
复位将该寄存器清零为 00H。

复位后: 00H    读/写    地址: FFFF300H

	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	按键中断模式的控制
0	不检测按键中断信号
1	检测按键中断信号

**注意事项** 在将 KRM 寄存器清零为 00H 一次后，应重写 KRM 寄存器。

**备注**        关于替换功能引脚的设置，见“表 4-14 用端口引脚作为替换功能引脚”。

15.3 注意事项

- (1) 如果向 KR0 至 KR7 中任何一个引脚输入低电平，则即输入另一个引脚的下降边缘也不会产生 INTKR 信号。
- (2) RXDA1 和 KR7 引脚一定不能同时使用。使用 RXDA1 引脚时，不要使用 KR7 引脚。使用 KR7 引脚时，不要使用 RXDA1 引脚（建议将 PFC91 位设置为 1 并将 PFCE91 位清为 0）。
- (3) 如果 KRM 寄存器被改变，则可能会产生中断请求信号（INTKR）。要防止这种情况的发生，应在禁用中断（DI）或在屏蔽后改变 KRM 寄存器，然后将中断请求标志（KRIC.KRIF 位）清为 0，并启用中断（EI）或清除屏蔽。
- (4) 要使用按键中断功能，一定要将端口引脚设置为按键中断引脚，再启用 KRM 寄存器。要从按键中断引脚切换为端口引脚，应禁用 KRM 寄存器的工作再设置端口引脚。

16.1 概述

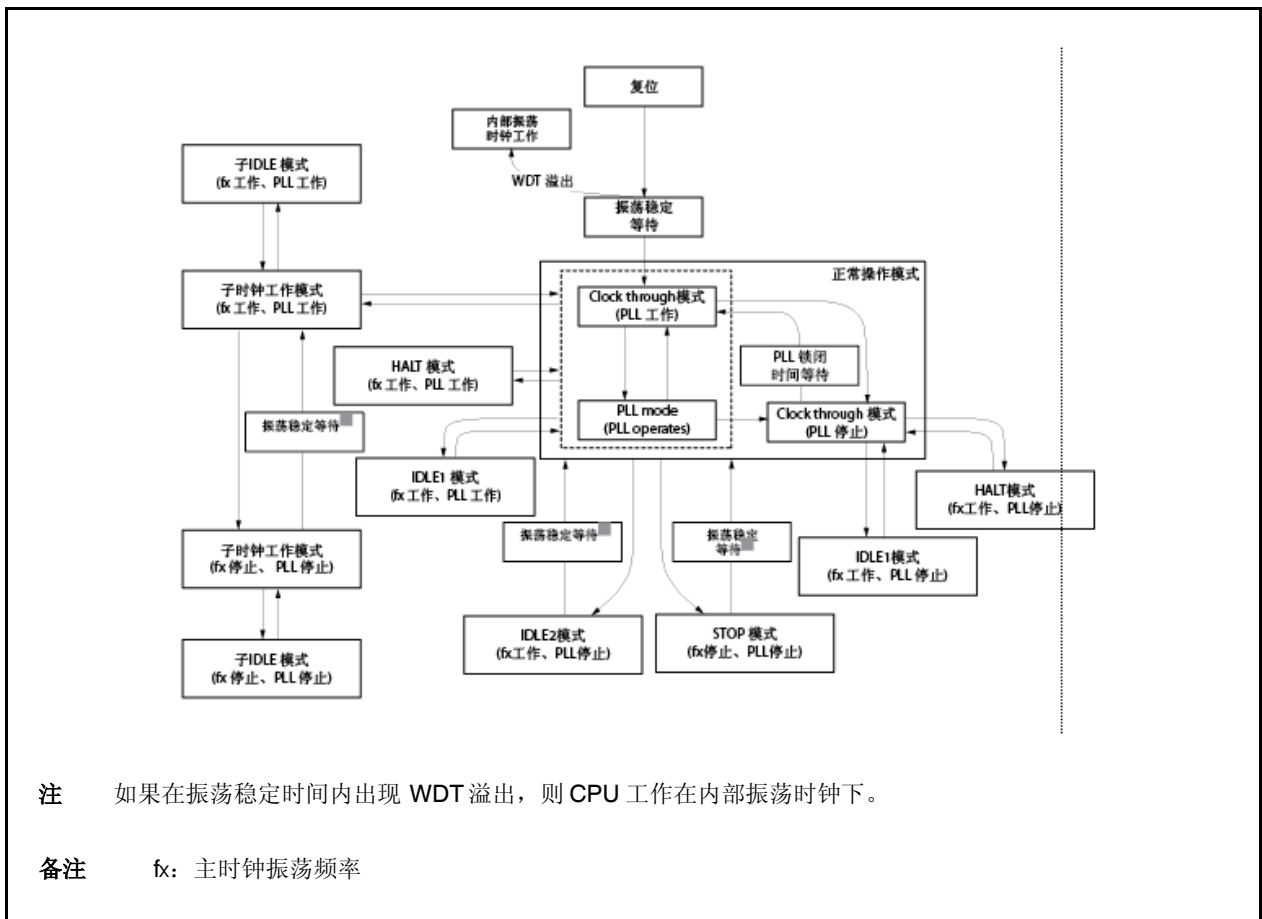
通过使用待机模式组合并选择合适的应用模式，系统的功耗可有效降低。可用的待机模式列于表 16-1 中。

表 16-1 待机模式

模式	功能简述
HALT 模式	此模式下，只停止 CPU 的工作时钟
IDLE1 模式	此模式下，除振荡器、PLL 和闪存外其余内部电路的工作停止
IDLE2 模式	此模式下，除振荡器外芯片的所有其它内部工作停止
STOP 模式	此模式下，除副时钟振荡器外芯片的所有其它内部工作停止
副时钟工作模式	此模式下，用副时钟作为内部系统时钟
子 IDLE 模式	此模式下，除振荡器工作在副时钟工作模式外，芯片的所有其它内部工作停止。

注 PLL 保持先前的工作状态。

图 16-1 状态转变



16.2 寄存器

(1) 节电控制寄存器 (PSC)

PSC 寄存器为 8 位寄存器，用来控制待机功能。该寄存器的 STP 位用来指定 STOP 模式。该寄存器为专用寄存器，只能由特殊序列组合进行写入（见“3.4.7 专用寄存器”）。

该寄存器可进行 8 位或 1 位读写。

复位将该寄存器清零为 00H。

复位后: 00H		R/W		地址: FFFFF1FEH				
PSC	7	6	5	4	3	2	1	0
	0	NMI1M	NMI0M	INTM	0	0	STP	0

NMI1M•	在出现INTWDT2信号时的待机模式接触控制
0•	启用INTWDT2信号解除待机模式
1	禁用INTWDT2信号解除待机模式

NMI0M•	通过NM1引脚输入进行待机模式解除控制
0•	通过启用NM1引脚输入解除待机模式
1	通过禁用NM1引脚输入解除待机模式

INTM•	通过可屏蔽中断请求信号进行待机模式解除控制
0•	通过启用可屏蔽中断请求信号解除待机模式
1	通过禁用可屏蔽中断请求信号解除待机模式

STP•	待机模式 <sup>注</sup> 设置
0•	正常模式
1	待机模式

注     待机模式由 STP 位设置：IDLE1、IDLE2、STOP 或子 IDLE 模式

- 注意事项
- 1. 在设置 IDLE1、IDLE2、STOP 或子 IDLE 模式前，应先设置 PSMR.PSM1 和 PSMR.PSM0 位再设置 STP 位。
  - 2. 当 HALT 模式解除时，NMI1M、NMI0M、INTM 位的设置无效。
  - 3. 如果 NMI1M、NMI0M 或 INTM 位在 STP 位设置为 1 的同时设置为 1，则 NMI1M、NMI0M 或 INTM 位的设置变为无效。如果在设置 IDLE1/IDLE2/STOP 模式时有未屏蔽的中断请求信号正在保留待定，应对应该中断请求信号的位（NMI1M、NMI0M 或 INTM）设置为 1，再将 STP 位设置为 1。

<R>



(2) 节电模式寄存器 (PSMR)

PSMR 寄存器为 8 位寄存器，用来控制节电模式中的工作状态及时钟的工作。  
该寄存器可进行 8 位或 1 位读写。  
复位将该寄存器清零为 00H。

复位后: 00H    读/写    地址: FFFF820H							
PSMR	7	6	5	4	3	2	1    0
	0	0	0	0	0	0	PSM1   PSM0
PSM1		PSM0		软件待机模式下操作的指定			
0		0		IDLE1			
0		1		STOP 模式			
1		0		IDLE2、子IDLE 模式			
1		1		STOP 模式			

- 注意事项
- 1. 一定要将 2 至 7 位清为“0”。
  - 2. 仅当 PSC.STP 位为 1 时 PSM0 和 PSM1 位才有效。

备注

IDLE1:    此模式下，除振荡器工作及一些其它电路（闪存和 PLL）外，所有其它工作停止。  
在 IDLE1 模式解除后，无需保证振荡稳定时间就可以恢复工作模式，类似于 HALT 模式。

IDLE2:    此模式下，除振荡器工作外所有其它工作停止。  
在 IDLE2 模式解除后，在由 OSTS 寄存器（闪存和 PLL）指定的设立时间过去之后才能恢复正常工作模式。

STOP:    此模式下，除副时钟振荡器工作外所有其它工作停止。  
在 STOP 模式解除后，在由 OSTS 寄存器指定的振荡稳定时间过去之后才能恢复正常工作模式。

子 IDLE:    此模式下，除振荡器外，所有其它工作停止。在通过中断请求信号解除 IDLE 模式后，在保证 12 个副时钟周期过去之后才能恢复为副时钟工作模式。

**(3) 振荡稳定时间选择寄存器 (OSTS)**

在 STOP 模式解除后振荡稳定之前的等待时间或在 IDLE2 模式解除后片上闪存稳定之前的等待时间均由 OSTS 寄存器控制。

OSTS 寄存器可进行 8 位读写。

复位将该寄存器清零为 06H。

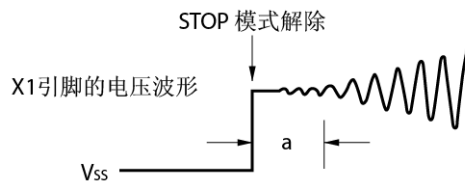
复位后: 06H R/W 地址: FFFFF6C0H

	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/设立时间的选择 <sup>注</sup>	fx	
				4 MHz	5 MHz
0	0	0	$2^{10}/f_x$	0.256 ms	0.205 ms
0	0	1	$2^{11}/f_x$	0.512 ms	0.410 ms
0	1	0	$2^{12}/f_x$	1.024 ms	0.819 ms
0	1	1	$2^{13}/f_x$	2.048 ms	1.638 ms
1	0	0	$2^{14}/f_x$	4.096 ms	3.277 ms
1	0	1	$2^{15}/f_x$	8.192 ms	6.554 ms
1	1	0	$2^{16}/f_x$	16.38 ms	13.107 ms
1	1	1	禁止设置		

**注** 在解除 STOP 模式和 IDLE2 模式时，分别需要等待振荡稳定时间和设立时间。

**注意事项 1.** Stop 模式解除后的等待时间不包括在 STOP 模式解除后时钟振荡开始（下图的“a”）之前的时间，且与 STOP 模式是由复位解除还是由于出现中断请求信号而解除无关。



**2.** 一定要将 3 至 7 位清为“0”。

**3.** 复位解除后的振荡稳定时间为  $2^{16}/f_x$ （因为 OSTS 寄存器的初始值=06H）。

**备注**  $f_x$ =主时钟振荡频率

## 16.3 HALT 模式

### 16.3.1 设置和工作状态

在正常工作模式下执行专用指令（HALT）时，设置为 HALT 模式。

在 HALT 模式下，时钟振荡器继续工作。只停止 CPU 的时钟供给；其它片上外围功能的时钟继续供给。

结果，程序执行停止，内部 RAM 保留了设置为 HALT 模式之前的内容。独立于 CPU 指令处理的片上外围功能继续工作。

表 16-3 展示了 HALT 模式的工作状态。

对于间歇性工作，利用 HALT 模式与正常工作模式的组合可降低系统的平均电流消耗。

**注意事项 1.** 应在 HALT 指令后插入五个（或更多）NOP 指令。

**2.** 如果当未屏蔽中断请求信号正在保留待定时执行 HALT 指令，状态会切换到 HALT 模式，但随后 HALT 模式会立即被待定的中断请求解除。

### 16.3.2 解除 HALT 模式

通过在 HALT 模式下可工作的外围功能发出的不可屏蔽中断请求信号（NMI 引脚输入、INTWDT2 信号）、未屏蔽的外部中断请求信号（INTP0 至 INTP7 引脚输入）、未屏蔽的内部中断请求信号，或通过复位信号（来自于 RESET 引脚输入、WDT2RES 信号、加电清零电路（POC）、低压检测器（LVI）或时钟监视器（CLM）），可解除 HALT 模式。

在 HALT 模式解除后，恢复为正常工作模式。

#### （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解除 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，可解除 IDLE 模式，且与中断请求信号的优先级无关。但是，若 HALT 模式是在中断服务例程中设置的，则之后发出的中断请求信号会进行下列处理。

- （a） 如果发出的中断请求信号的优先级低于当前正在服务中的中断请求，则 HALT 模式解除，但该中断请求信号不会得到肯定应答。中断请求信号本身被保留。
- （b） 如果发出的中断请求信号的优先级高于当前正在服务中的中断请求（包括不可屏蔽中断请求信号），则 HALT 模式解除且该中断请求信号得到肯定应答。

**表 16-2 用中断请求信号解除 HALT 模式后的操作**

解除来源	中断启用（EI）状态	中断禁用（DI）状态
不可屏蔽中断请求信号	转到处理程序地址处执行。	
可屏蔽中断请求信号	转到处理程序地址处执行或执行下个指令	执行下个指令。

## (2) 通过复位解除 HALT 模式

进行与正常的复位操作相同的操作。

表 16-3Halt 模式的工作状态

Halt 模式的设置		工作状态	
项		不使用副时钟时	使用副时钟时
主时钟振荡器		启用振荡	
副时钟振荡器		—	启用振荡
内部振荡器		启用振荡	
PLL		可操作	
CPU		停止操作	
中断控制器		可操作	
定时器 P (TMP0 至 TMP3)		可操作	
定时器 Q (TMQ0)		可操作	
定时器 M (TMM0)		当选择非 f <sub>XT</sub> 时钟作为计数时钟时可操作	可操作
监视定时器		当选择 f <sub>x</sub> (BRG 分频) 作为计数时钟时可操作	可操作
看门狗定时器 2		可操作	
串行接口	CSIB0、CSIB1	可操作	
	UARTA0、UARTA1	可操作	
模数转换器		可操作	
按键中断功能 (KR)		可操作	
端口功能		保持设置为 HALT 模式之前的状态	
内部数据		CPU 寄存器、状态、数据及其它内部数据，如内部 RAM 的内容等，均保持设置为 HALT 模式之前的状态。	

## 16.4 IDLE1 模式

### 16.4.1 设置和工作状态

通过在正常工作模式下将 PSMR.PSM1 和 PSMR.PSM0 位清零为 00 并将 PSC.STP 位设置为 1 可设置为 IDLE1 模式。

在 IDLE1 模式下，时钟振荡器、PLL 和闪存继续工作但对 CPU 及其它片上外围功能的时钟供给停止。

结果，程序停止执行，内部 RAM 保持设置为 IDLE1 模式之前的内容。CPU 及其它片上外围功能停止工作。但是，可在副时钟或外部时钟下工作的片上外围功能继续工作。

表 16-5 展示了 IDLE1 模式的操作状态。

因为 IDLE1 模式下停止了片上外围功能的工作，所以它比 HALT 模式更能降低功耗。主时钟振荡器不停止，所以在 IDLE1 模式解除后无需等待振荡稳定时间就能恢复为正常工作模式，其方式类似于解除 HALT 模式时。

- 注意事项**
1. 应在 PSC 寄存器存储数据指令后插入五个或更多 NOP 指令来设置为 IDLE1 模式。
  2. 若当未屏蔽中断请求信号正在保留待定时设置 IDLE1 模式，则 IDLE1 模式会因特定中断请求而立即解除。

### 16.4.2 解除 IDLE1 模式

通过在 IDLE1 模式下可工作的外围功能发出的不可屏蔽中断请求信号（NMI 引脚输入、INTWDT2 信号）、未屏蔽的外部中断请求信号（INTP0 至 INTP7 引脚输入）、未屏蔽的内部中断请求信号，或通过复位信号（来自于 RESET 引脚输入、WDT2RES 信号、加电清零电路（POC）、低压检测器（LVI）或时钟监视器（CLM）），可解除 IDLE1 模式。

在 IDLE1 模式解除后，恢复为正常工作模式。

#### （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解除 IDLE1 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，可解除 IDLE1 模式，且与中断请求信号的优先级无关。但是，若 IDLE1 模式是在中断服务例程中设置的，则之后发出的中断请求信号会进行如下的处理。

- 注意事项**
1. 通过将 PSC.NMI1M、PSC.NMI0M、PSC.INTM 位设置为 1 禁用的中断请求信号变为无效，IDLE1 模式不解除。
  2. 如果通过 NFC 寄存器选择消除数字噪声，如果从 fxx/64、fxx/128、fxx/256、fxx/512、fxx/1024 中选择采样时钟，则无法通过 INTP3 引脚的中断请求信号解除 IDLE1 模式。详情请见“14.6.2（4）消噪控制寄存器（NFC）”。

- （a） 如果发出的中断请求信号的优先级低于当前正在服务中的中断请求，则 IDLE1 模式解除，但该中断请求信号不会得到肯定应答。中断请求信号本身被保留。
- （b） 如果发出的中断请求信号的优先级高于当前正在服务中的中断请求（包括不可屏蔽中断请求信号），则 IDLE1 模式解除且该中断请求信号得到肯定应答。

表 16-4 用中断请求信号解除 IDLE1 模式后的操作

解除来源	中断启用 (EI) 状态	中断禁用 (DI) 状态
不可屏蔽中断请求信号	转到处理程序地址处执行。	
可屏蔽中断请求信号	转到处理程序地址处执行或执行下个指令	执行下个指令。

## (2) 用复位解除 IDLE1 模式

进行与正常的复位操作相同的操作。

表 16-5 Idle1 模式的操作状态

Idle1 模式的设置		工作状态	
项		不使用副时钟时	使用副时钟时
主时钟振荡器		启用振荡	
副时钟振荡器		—	启用振荡
内部振荡器		启用振荡	
PLL		可操作	
CPU		停止操作	
中断控制器		停止操作 (但可以解除待机模式)	
定时器 P (TMP0 至 TMP3)		停止操作	
定时器 Q (TMQ0)		停止操作	
定时器 M (TMM0)		当选择 $f_{n/8}$ 为计数时钟时可操作	当选择 $f_{n/8}$ 或 $f_{xt}$ 为计数时钟时可操作
监视定时器		当选择 $f_x$ (BRG 分频) 作为计数时钟时可操作	可操作
看门狗定时器 2		可操作	
串行接口	CSIB0、CSIB1	当选择 SCKBn 输入时钟作为计数时钟时可操作 (n=0、1)	
	UARTA0、UARTA1	停止操作 (但当选择 ASCKA0 输入时钟时 UARTA0 可操作)	
模数转换器		保持操作 (保持转换结果) <sup>※</sup>	
按键中断功能 (KR)		可操作	
端口功能		保持设置为 IDLE1 模式之前的状态	
内部数据		CPU 寄存器、状态、数据及其它内部数据, 如内部 RAM 的内容等, 均保持设置为 IDLE1 模式之前的状态。	

注 要实现低功耗, 应在切换为 IDLE1 模式前停止模数转换器。

## 16.5 IDLE2 模式

### 16.5.1 设置及工作状态

通过在正常工作模式下将 PSMR.PSM1 和 PSMR.PSM0 位设置为 10 并将 PSC.STP 位设置为 1 可设置为 IDLE2 模式。

在 IDLE2 模式下，时钟振荡器继续工作但对 CPU、PLL、闪存及其它片上外围功能的时钟供给停止。

结果，程序停止执行，内部 RAM 保持设置为 IDLE2 模式之前的内容。CPU、PLL 及其它片上外围功能停止工作。但是，可在副时钟或外部时钟下工作的片上外围功能继续工作。

表 16-7 展示了 IDLE2 模式的操作状态。

因为 IDLE2 模式下停止了片上外围功能、PLL 和闪存的工作，所以它比 IDLE1 模式更能降低功耗。但是，因为停止了 PLL 和闪存，所以在解除 IDLE2 模式时需要等待 PLL 和闪存的设立时间。

- 注意事项**
1. 应在 PSC 寄存器存储数据指令后插入五个或更多 NOP 指令来设置为 IDLE2 模式。
  2. 若当未屏蔽中断请求信号正在保留待定时设置 IDLE2 模式，则 IDLE2 模式会因特定中断请求而立即解除。

### 16.5.2 解除 IDLE2 模式

通过在 IDLE2 模式下可工作的外围功能发出的不可屏蔽中断请求信号（NMI 引脚输入、INTWDT2 信号）、未屏蔽的外部中断请求信号（INTP0 至 INTP7 引脚输入）、未屏蔽的内部中断请求信号，或通过复位信号（来自于 RESET 引脚输入、WDT2RES 信号、加电清零电路（POC）、低压检测器（LVI）或时钟监视器（CLM）），可解除 IDLE2 模式。PLL 回到在设置为 IDLE2 模式之前的工作状态。

在 IDLE2 模式解除后，恢复为正常工作模式。

#### （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解除 IDLE2 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，可解除 IDLE2 模式，且与中断请求信号的优先级无关。但是，若 IDLE2 模式是在中断服务例程中设置的，则之后发出的中断请求信号会进行如下的处理。

- 注意事项**
1. 通过将 PSC.NMI1M、PSC.NMI0M、PSC.INTM 位设置为 1 禁用的中断请求信号变为无效，IDLE2 模式不解除。
  2. 如果通过 NFC 寄存器选择消除数字噪声且从 fxx/64、fxx/128、fxx/256、fxx/512、fxx/1024 中选择采样时钟，则无法通过 INTP3 引脚的中断请求信号解除 IDLE2 模式。详情请见“14.6.2（4）消噪控制寄存器（NFC）”。

- （a） 如果发出的中断请求信号的优先级低于当前正在服务中的中断请求，则 IDLE2 模式解除，但该中断请求信号不会得到肯定应答。中断请求信号本身被保留。
- （b） 如果发出的中断请求信号的优先级高于当前正在服务中的中断请求（包括不可屏蔽中断请求信号），则 IDLE2 模式解除且该中断请求信号得到肯定应答。

表 16-6 用中断请求信号解除 IDLE2 模式后的操作

解除来源	中断启用 (EI) 状态	中断禁用 (DI) 状态
不可屏蔽中断请求信号	在保证规定的设立时间后，转到处理程序地址处执行。	
可屏蔽中断请求信号	在保证规定的设立时间后，转到处理程序地址处执行或执行下个指令。	在保证规定的设立时间后执行下个指令。

## (2) 用复位解除 IDLE2 模式

进行与正常的复位操作相同的操作。

表 16-7 Idle2 模式的操作状态

Idle2 模式的设置		工作状态	
项		不使用副时钟时	使用副时钟时
主时钟振荡器		启用振荡	
副时钟振荡器		—	启用振荡
内部振荡器		启用振荡	
PLL		停止操作	
CPU		停止操作	
中断控制器		停止操作（但可以解除待机模式）	
定时器 P (TMP0 至 TMP3)		停止操作	
定时器 Q (TMQ0)		停止操作	
定时器 M (TMM0)		当选择 $f_R/8$ 为计数时钟时可操作	当选择 $f_R/8$ 或 $f_{XT}$ 为计数时钟时可操作
监视定时器		当选择 $f_X$ (BRG 分频) 作为计数时钟时可操作	可操作
看门狗定时器 2		可操作	
串行接口	CSIB0、CSIB1	当选择 SCKBn 输入时钟作为计数时钟时可操作 (n=0、1)	
	UARTA0、UARTA1	停止操作（但当选择 ASCKA0 输入时钟时 UARTA0 可操作）	
模数转换器		保持操作（保持转换结果）注	
按键中断功能 (KR)		可操作	
端口功能		保持设置为 IDLE2 模式之前的状态	
内部数据		CPU 寄存器、状态、数据及其它内部数据，如内部 RAM 的内容等，均保持设置为 IDLE2 模式之前的状态。	

注 要实现低功耗，应在切换为 IDLE2 模式前停止模数转换器。



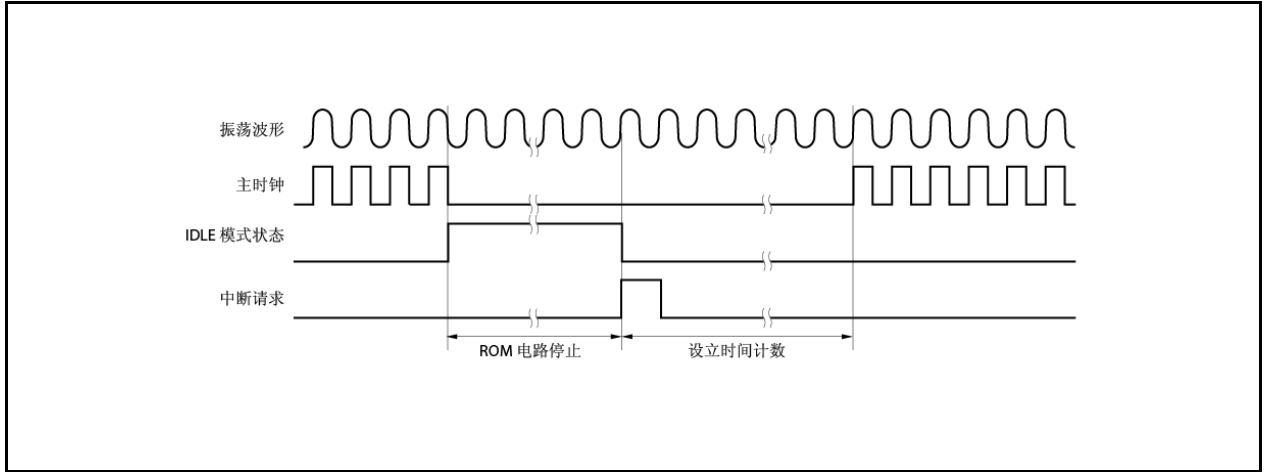
### 16.5.3 在解除 IDLE2 模式时保证设立时间

因为在设置为 IDLE2 模式后除主时钟振荡器外其它模块停止工作，所以应在 IDLE2 模式解除后保证 ROM（闪存）的设立时间。

#### （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解除 IDLE2 模式

通过设置 OSTS 寄存器保证规定的设立时间。

当解除源产生时，专用内部定时器根据 OSTS 寄存器的设置开始计数。当它溢出时，恢复为正常操作模式。



#### （2）通过复位解除（RESET 引脚输入、WDT2RES 生成）

此操作与正常复位相同。

振荡稳定时间为 OSTS 寄存器的初始值  $2^{16}/f_x$ 。

## 16.6 STOP 模式

### 16.6.1 设置及工作状态

通过在正常工作模式下将 PSMR.PSM1 和 PSMR.PSM0 位设置为 01 或 11 并将 PSC.STP 位设置为 1 可设置为 STOP 模式。

在 STOP 模式下，副时钟振荡器继续工作，但主时钟振荡器停止。对 CPU 和片上外围功能的时钟供给停止。

结果，程序停止执行，内部 RAM 保持设置为 STOP 模式之前的内容。在由副时钟振荡器振荡产生的时钟或在外部时钟下工作的片上外围功能继续工作。

表 16-9 展示了 STOP 模式的工作状态。

因为 STOP 模式停止了主时钟振荡器的工作，所以它能将功耗降到比 IDLE2 模式更低的水平。如果未使用副时钟振荡器、内部振荡器、外部时钟，则只有漏电流，功耗最小化。

- 注意事项**
1. 应在 PSC 寄存器存储数据指令后插入五个或更多 NOP 指令来设置为 STOP 模式。
  2. 若当未屏蔽中断请求信号正在保留待定时设置为 STOP 模式，则 STOP 模式会因待定中断请求而立即解除。

### 16.6.2 解除 STOP 模式

通过在 STOP 模式下可工作的外围功能发出的不可屏蔽中断请求信号（NMI 引脚输入、INTWDT2 信号）、未屏蔽的外部中断请求信号（INTP0 至 INTP7 引脚输入）、未屏蔽的内部中断请求信号，或通过复位信号（来自于 RESET 引脚输入、WDT2RES 信号、加电清零电路（POC）或低压检测器（LVI）），可解除 STOP 模式。

在 STOP 模式解除后，必须保证振荡稳定时间才能恢复为正常工作模式。

- 注意事项**
1. 通过将 PSC.NMI1M、PSC.NMI0M、PSC.INTM 位设置为 1 禁用的中断请求信号变为无效，STOP 模式不解除。
  2. 如果通过 NFC 寄存器选择消除数字噪声且从 fxx/64、fxx/128、fxx/256、fxx/512、fxx/1024 中选择采样时钟，则无法通过 INTP3 引脚的中断请求信号解除 STOP 模式。详情请见“14.6.2（4）消噪控制寄存器（NFC）”。

#### （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解除 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，可解除 STOP 模式，且与中断请求信号的优先级无关。但是，若 STOP 模式是在中断服务例程中设置的，则之后发出的中断请求信号进行如下的处理。

- （a）如果发出的中断请求信号的优先级低于当前正在服务中的中断请求，则 STOP 模式解除，但该中断请求信号不会得到肯定应答。中断请求信号本身被保留。
- （b）如果发出的中断请求信号的优先级高于当前正在服务中的中断请求（包括不可屏蔽中断请求信号），则 STOP 模式解除且该中断请求信号得到肯定应答。

表 16-8 用中断请求信号解除 STOP 模式后的操作

解除来源	中断启用 (EI) 状态	中断禁用 (DI) 状态
不可屏蔽中断请求信号	在保证振荡稳定时间后转到处理程序地址处执行。	
可屏蔽中断请求信号	在保证振荡稳定时间后转到处理程序地址处执行或执行下个指令。	在保证振荡稳定时间后执行下个指令。

## (2) 用复位解除 STOP 模式

进行与正常的复位操作相同的操作。

表 16-9 Stop 模式的工作状态

Stop 模式的设置		工作状态	
项		不使用副时钟时	使用副时钟时
主时钟振荡器		停止振荡	
副时钟振荡器		—	启用振荡
内部振荡器		启用振荡	
PLL		停止操作	
CPU		停止操作	
中断控制器		停止操作（但可以解除待机模式）	
定时器 P (TMP0 至 TMP3)		停止操作	
定时器 Q (TMQ0)		停止操作	
定时器 M (TMM0)		当选择 $f_{W/8}$ 为计数时钟时可操作	当选择 $f_{W/8}$ 或 $f_{XT}$ 为计数时钟时可操作
监视定时器		停止操作	当选择 $f_{XT}$ 为计数时钟时可操作
看门狗定时器 2		当选择 $f_W$ 为计数时钟时可操作	
串行接口	CSIB0、CSIB1	当选择 $SCKB_n$ 输入时钟作为计数时钟时可操作 ( $n=0, 1$ )	
	UARTA0、UARTA1	停止操作（但当选择 $ASCKA0$ 输入时钟时 $UARTA0$ 可操作）	
模数转换器		停止操作（转换结果未定义） <sup>1, 2</sup>	
按键中断功能 (KR)		可操作	
端口功能		保持设置为 STOP 模式之前的状态	
内部数据		CPU 寄存器、状态、数据及其它内部数据，如内部 RAM 的内容等，均保持设置为 STOP 模式之前的状态。	

- 注
1. 如果在模数转换器正在工作时设置为 STOP 模式，则模数转换器自动停止，在 STOP 模式解除后重新开始工作。但是，在这种情况下，STOP 模式解除后的模数转换结果是无效的。在设置为 STOP 模式前的所有模数转换结果均无效。
  2. 即使在模数转换器正在工作时设置为 STOP 模式，其功耗的减低等同于在设置为 STOP 模式前停止模数转换器。

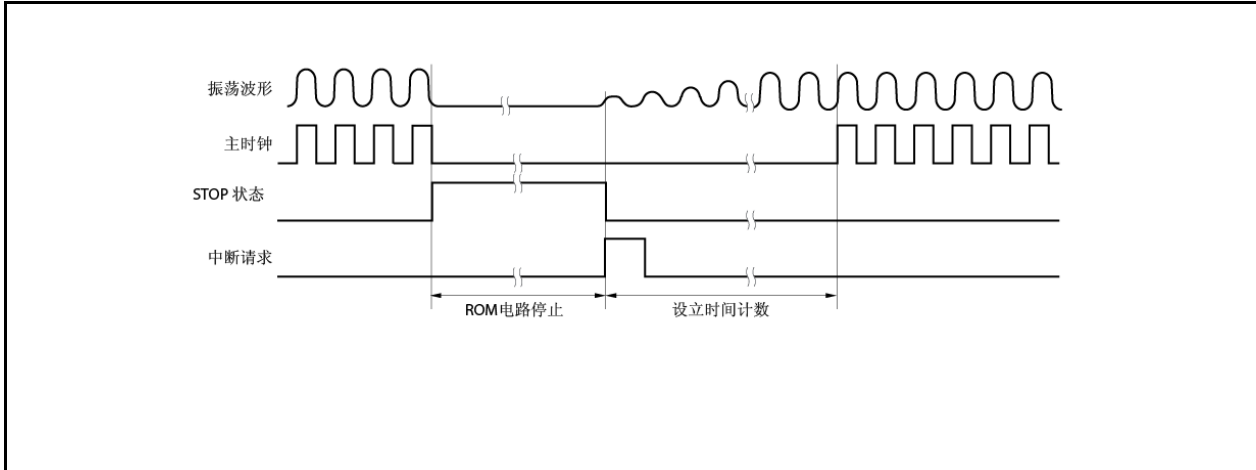
### 16.6.3 在解除 STOP 模式时保证振荡稳定时间

因为设置为 STOP 模式后主时钟振荡器停止工作，所以在 STOP 模式解除后要保证主时钟振荡器的振荡稳定时间。

#### (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解除 STOP 模式

通过设置 OSTS 寄存器来保证振荡稳定时间。

当解除源产生时，专用内部定时器根据 OSTS 寄存器的设置开始计数。当它溢出时，恢复为正常操作模式。



#### (2) 通过复位解除

此操作与正常复位相同。

振荡稳定时间为 OSTS 寄存器的初始值  $2^{16}/f_x$ 。

## 16.7 副时钟工作模式

### 16.7.1 设置及工作状态

通过在正常工作模式下将 **PCC.CK3** 位设置为 **1** 可设置为副时钟工作模式。

在设置为副时钟工作模式时，内部系统时钟从主时钟变为副时钟。利用 **PCC.CLS** 检验时钟是否已切换。

当 **PCC.MCK** 设置为 **1** 时，主时钟振荡器停止工作。结果，系统只工作在副时钟下。

在副时钟工作模式下，因为用副时钟作为内部系统时钟，所以功耗能够下降到低于正常工作模式的水平。此外，通过停止主时钟振荡器的工作可进一步将功耗降低到 **STOP** 模式的水平。

表 16-10 展示了副时钟工作模式下的工作状态。

- 注意事项**
1. 在操作 **CK3** 位时，不要改变 **PCC.CK2** 至 **PCC.CK0** 位的设置值（建议使用位操作指令来对各位进行操作）。关于 **PCC** 寄存器的详情请参见“5.3（1）处理器时钟控制寄存器（PCC）”。
  2. 如果下列条件不满足，应改变 **CK2** 至 **CK0** 位使得条件满足，并设置为副时钟工作模式。  
内部系统时钟 ( $f_{CLK}$ ) > 副时钟 ( $f_{XT}$ ) × 4

**备注**      内部系统时钟 ( $f_{CLK}$ ):      根据 **CK2** 至 **CK0** 位的设置从主时钟 ( $f_{xx}$ ) 产生的时钟

### 16.7.2 解除副时钟工作模式

当 **CK3** 位清为 **0** 时，通过复位信号（来自于 **RESET** 引脚输入、**WDT2RES** 信号、加电清零电路（**POC**）或低压检测器（**LVI**））可解除副时钟工作模式。

如果主时钟停止（**MCK** 位=1），应将 **MCK** 位设置为 **1**，用软件保证主时钟的振荡稳定时间，再将 **CK3** 位清为 **0**。

当副时钟工作模式解除时，恢复为正常操作模式。

- 注意事项** 在操作 **CK3** 位时，不要改变 **CK2** 至 **CK0** 位的设置值（建议使用位操作指令来对各位进行操作）。  
关于 **PCC** 寄存器的详情请参见“5.3（1）处理器时钟控制寄存器（PCC）”。

表 16-10 副时钟工作模式下的工作状态

副时钟工作模式的设置		工作状态	
项		主时钟振荡时	主时钟停止时
副时钟振荡器		启用振荡	
内部振荡器		启用振荡	
PLL		可操作	停止工作 <sup>注</sup>
CPU		可操作	
中断控制器		可操作	
定时器 P (TMP0 至 TMP3)		可操作	停止操作
定时器 Q (TMQ0)		可操作	停止操作
定时器 M (TMM0)		可操作	当选择 $f_R/8$ 或 $f_{XT}$ 为计数时钟时可操作
监视定时器		可操作	当选择 $f_{XT}$ 为计数时钟时可操作
看门狗定时器 2		可操作	当选择 $f_R$ 为计数时钟时可操作
串行接口	CSIB0、CSIB1	可操作	当选择 SCKBn 输入时钟作为计数时钟时可操作 (n=0、1)
	UARTA0、UARTA1	可操作	停止操作 (但当选择 ASCKA0 输入时钟时 UARTA0 可操作)
模数转换器		可操作	停止操作
按键中断功能 (KR)		可操作	
端口功能		可设置	
内部数据		可设置	

注 一定要在停止主时钟前停止 PLL (PLLCTL.PLLON=0)。

注意事项 当 CPU 工作在副时钟下且主时钟振荡停止时，访问出现等待的寄存器是被禁止的。如果产生了等待，只能通过复位进行解除 (见 3.4.8 (2))。

## 16.8 子 IDLE 模式

### 16.8.1 设置及工作状态

通过在副时钟工作模式下将 PSMR.PSM1 和 PSMR.PSM0 位设置为 10 并将 PSC.STP 位设置为 1 可设置为子 IDLE 模式。

在此模式下，时钟振荡器继续工作但对 CPU、闪存及其它片上外围功能的时钟供给停止。

结果，程序停止执行，内部 RAM 保持设置为子 IDLE 模式之前的内容。CPU 及其它片上外围功能停止。但是，可在副时钟或外部时钟下工作的片上外围功能继续工作。

因为子 IDLE 模式下 CPU、闪存及其它片上外围功能停止工作，所以它比副时钟工作模式更能降低功耗。如果在主时钟已停止后设置子 IDLE 模式，则当前的功耗可降低到 STOP 模式的水平。

表 16-12 展示了子 IDLE 模式的操作状态。

- 注意事项**
1. 在将设置为子 IDLE 模式的指令存储到 PSC 寄存器中后，应插入五个（或更多）NOP 指令。
  2. 若当未屏蔽中断请求信号正在保留待定时设置为子 IDLE 模式，则子 IDLE 模式会因待定时中断请求而立即解除。

### 16.8.2 解除子 IDLE 模式

通过在子 IDLE 模式下可工作的外围功能发出的不可屏蔽中断请求信号（NMI 引脚输入、INTWDT2 信号）、未屏蔽的外部中断请求信号（INTP0 至 INTP7 引脚输入）、未屏蔽的内部中断请求信号，或通过复位信号（来自于 RESET 引脚输入、WDT2RES 信号、加电清零电路（POC）、低压检测器（LVI）或时钟监视器（CLM）），可解除子 IDLE 模式。PLL 回到在设置为子 IDLE 模式之前的工作状态。

当通过中断请求信号解除子 IDLE 模式时，会设置为副时钟工作模式。

#### （1）通过不可屏蔽中断请求信号或未屏蔽可屏蔽中断请求信号解除子 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，可解除子 IDLE 模式，且与中断请求信号的优先级无关。

但是，若子 IDLE 模式是在中断服务例程中设置的，则之后发出的中断请求信号将进行如下的处理。

**注意事项** 1. 通过将 PSC.NMI1M、PSC.NMI0M、PSC.INTM 位设置为 1 禁用的中断请求信号变为无效，子 IDLE 模式不解除。

2. 当子 IDLE 模式解除时，从解除子 IDLE 模式的中断请求信号生成到模式解除要经过 12 个副时钟周期（约 366 $\mu$ s）。

3. 如果通过 NFC 寄存器选择消除数字噪声且从 fxx/64、fxx/128、fxx/256、fxx/512、fxx/1024 中选择采样时钟，则无法通过 INTP3 引脚的中断请求信号解除子 IDLE 模式。详情请见“14.6.2（4）消噪控制寄存器（NFC）”。

（a） 如果发出的中断请求信号的优先级低于当前正在服务中的中断请求，则子 IDLE 模式解除，但该中断请求信号不会得到肯定应答。中断请求信号本身被保留。

（b） 如果发出的中断请求信号的优先级高于当前正在服务中的中断请求（包换不可屏蔽中断请求信号），则子 IDLE 模式解除且该中断请求信号得到肯定应答。

**表 16-11 用中断请求信号解除子 IDLE 模式后的操作**

解除来源	中断启用（EI）状态	中断禁用（DI）状态
不可屏蔽中断请求信号	转到处理程序地址处执行。	
可屏蔽中断请求信号	转到处理程序地址处执行或执行下个指令	执行下个指令。



## (2) 通过复位解除子 IDLE 模式

进行与正常的复位操作相同的操作。

表 16-12 子 IDLE 模式的工作状态

子 IDLE 模式的设置		工作状态	
		主时钟振荡时	主时钟停止时
副时钟振荡器		启用振荡	
内部振荡器		启用振荡	
PLL		可操作	停止工作 <sup>※1</sup>
CPU		停止操作	
中断控制器		停止操作（但可以解除待机模式）	
定时器 P（TMP0 至 TMP3）		停止操作	
定时器 Q（TMQ0）		停止操作	
定时器 M（TMM0）		当选择 $f_R/8$ 或 $f_{XT}$ 为计数时钟时可操作	
监视定时器		停止操作	当选择 $f_{XT}$ 为计数时钟时可操作
看门狗定时器 2		当选择 $f_R$ 为计数时钟时可操作	
串行接口	CSIB0、CSIB1	当选择 SCKBn 输入时钟作为计数时钟时可操作（n=0、1）	
	UARTA0、UARTA1	停止操作（但当选择 ASCKA0 输入时钟时 UARTA0 可操作）	
模数转换器		保持操作（保持转换结果） <sup>※2</sup>	
按键中断功能（KR）		可操作	
端口功能		保持设置为子 IDLE 模式之前的状态	
内部数据		CPU 寄存器、状态、数据及其它内部数据，如内部 RAM 的内容等，均保持设置为子 IDLE 模式之前的状态。	

- 注
1. 一定要在停止主时钟前停止 PLL（PLLCTL.PLLON 位=0）。
  2. 要实现低功耗，应在切换为子 IDLE 模式前停止模数转换器。

### 17.1 概述

可用下面的复位功能。

#### (1) 四种复位源

- 通过 **RESET** 引脚输入外部复位
- 通过看门狗定时器 2 (WDT2) 溢出 (WDT2RES) 进行复位
- 通过比较低压检测器 (LVI) 的供电电压和检测到的电压进行系统复位
- 通过检测时钟监视器 (CLM) 振荡停止进行系统复位
- 通过加电清零电路进行系统复位

当复位解除后，复位来源可通过复位源标志寄存器 (RESF) 进行确认。

#### (2) 应急工作模式

如果在复位后插入的主时钟振荡稳定时间内 WDT2 出现溢出，则会判断为主时钟振荡异常，CPU 开始在内部振荡时钟下工作。

**注意事项** 当 CPU 正在内部振荡时钟下工作时，访问产生了等待状态的寄存器是被禁止的。关于产生等待状态的寄存器，请见“3.4.8 (2) 访问特定片上外围 I/O 寄存器”。

17.2 用于检验复位源的寄存器

V850ES/HF2 有四种复位源。在复位解除后，出现的复位来源可通过复位源标志寄存器（RESF）进行检查。

(1) 复位源标志寄存器（RESF）

RESF 寄存器为专用寄存器，只能通过特定序列的组合写入（见“3.4.7 专用寄存器”）。

RESF 寄存器指示产生复位信号的来源。

该寄存器可进行 8 位或 1 位读写。

**RESET** 引脚输入或 POC 复位将该寄存器设置为 00H。如果复位来源不是 **RESET** 引脚信号，则默认值会发生变化。

复位后: 00H<sup>注</sup>    R/W    地址: FFFF888H

	7	6	5	4	3	2	1	0
RESF	0	0	0	WDT2RF	0	0	CLMRF	LVIRF

WDT2RF	来自WDT2的复位信号	
0	不产生	
1	产生	

CLMRF	来自 CLM的复位信号	
0	不产生	
1	产生	

LVIRF	来自LVI的的复位信号	
0	不产生	
1	产生	

注     当通过 **RESET** 引脚执行复位时，RESF 寄存器的值清零为 00H。在通过看门狗定时器 2（WDT2）、低压检测器（LVI）或时钟监视器（CLM）执行复位时，该寄存器的复位标志（WDT2RF 位、CLMRF 位、LVIRF 位）会被设置。但是，其它来源会保留。

注意事项   只能向该寄存器的各位写入“0”。如果写入“0”与设置标志（出现复位）冲突，应首先设置标志。

## 17.3 操作

### 17.3.1 通过 **RESET** 引脚进行复位操作

当低电平输入到 **RESET** 引脚时，系统复位，各硬件设备初始化。

当 **RESET** 引脚的电平从低变高时，复位状态解除。

表 17-1 **Reset** 引脚输入后的硬件状态

项		复位中	复位后
主时钟振荡器 (fx)		振荡停止	振荡开始
子时钟振荡器 (fxT)	晶体振荡	振荡继续	
	RC 振荡	振荡停止	振荡开始
内部振荡器		振荡停止	振荡开始
外围时钟 (fx 至 fx/1、024)		操作停止	在保证振荡稳定时间后操作开始
内部系统时钟 (fCLK)、CPU 时钟 (fCPU)		操作停止	在保证振荡稳定时间（初始化为 fx/8）后操作开始
CPU		初始化	在保证振荡稳定时间后程序开始执行
看门狗定时器 2		操作停止（初始化为 0）	操作开始
内部 RAM		如果加电复位或 CPU 访问与复位输入冲突，则为未定义（数据损坏）。 否则保持复位后瞬间输入的值 <sup>1</sup>	
I/O 线路（端口/替换功能引脚）		高阻 <sup>2</sup>	
片上外围 I/O 寄存器		初始化为指定状态，设置 OCDM 寄存器（01H）。	
其它片上外围功能		操作停止	在保证振荡稳定时间后可以开始工作

- <R> 注
1. 由于支持引导交换功能，V850ES/HF2 在内部系统复位状态解除后其固件会使用一部分内部 RAM。因此，在加电复位后一些 RAM 区的内容无法保留。详情请见“17.4 复位解除后的操作”。
  2. 电源打开时，下面的引脚即使在复位过程中也可能会临时输出未定义的电平。

•P53/KR3/TIQ00/TOQ00/DDO 引脚

**注意事项** OCDM 寄存器通过 **RESET** 引脚输入进行初始化。因此要小心注意的是，如果在复位解除后 OCDM.OCDM0 位清零前有高电平输入到 P05/DRST 引脚，则会进入片上调试模式。详情请见“第四章 端口功能”。

图 17-1 通过 RESET 引脚输入进行复位操作的定时

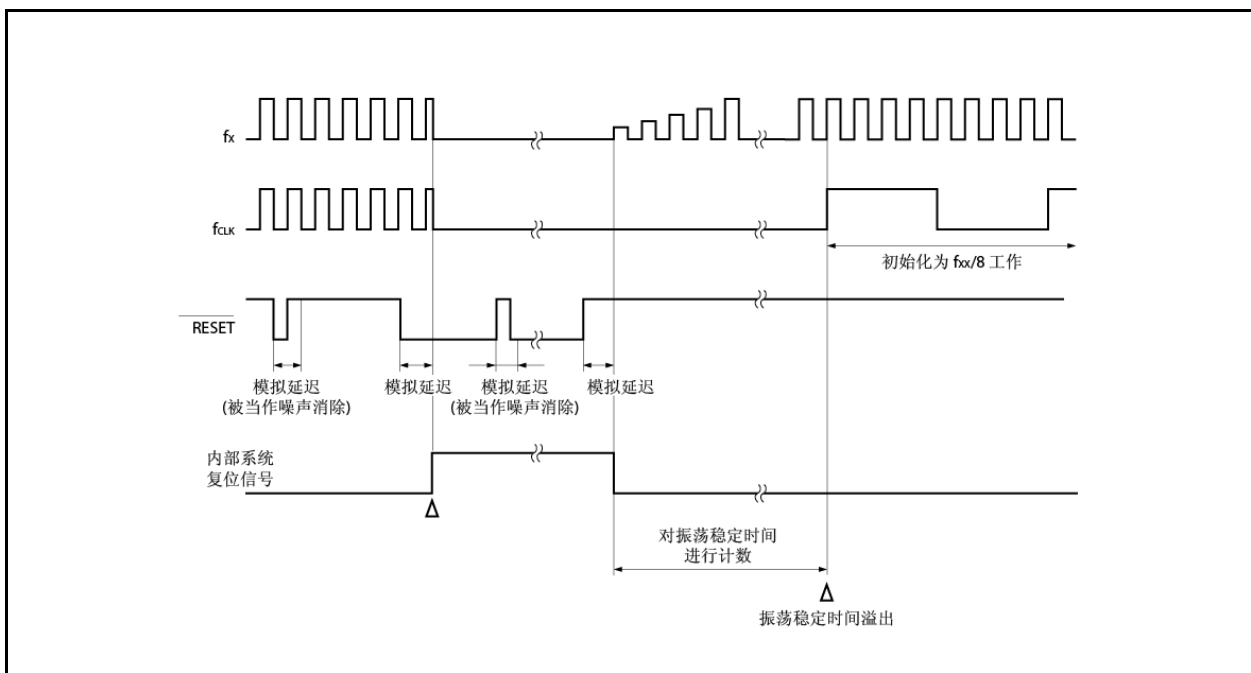
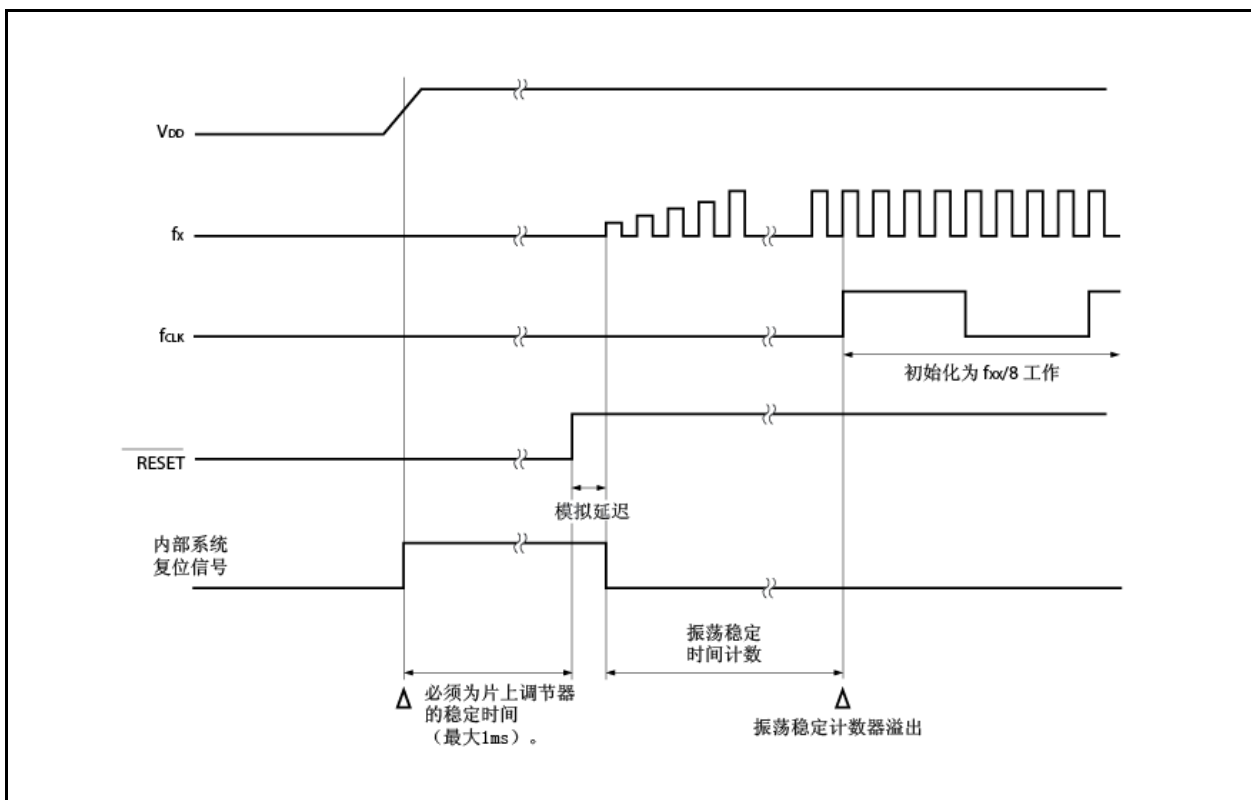


图 17-2 上电复位操作的定时



### 17.3.2 通过看门狗定时器 2 进行复位操作

当看门狗定时器 2 由于溢出而设置为复位操作时，在看门狗定时器 2 溢出（产生 WDT2RES 信号）时，会执行系统复位，硬件初始化为初始状态。

在看门狗定时器 2 溢出后，进入复位状态并持续预先确定的时间（模拟延迟）然后复位状态自动解除。

在复位期间，主时钟振荡器停止。

表 17-2 在看门狗定时器 2 复位操作过程中的硬件状态

项		复位中	复位后
主时钟振荡器 (fX)		振荡停止	振荡开始
子时钟振荡器 (fXT)	晶体振荡	振荡继续	
	RC 振荡	振荡停止	振荡开始
内部振荡器		振荡停止	振荡开始
外围时钟 (fxx 至 fxx/1、024)		操作停止	在保证振荡稳定时间后操作开始
内部系统时钟 (fxx)、 CPU 时钟 (fCPU)		操作停止	在保证振荡稳定时间（初始化为 fxx/8） 后操作开始
CPU		初始化	在保证振荡稳定时间后执行程序
看门狗定时器 2		操作停止（初始化为 0）	操作开始
内部 RAM		如果加电复位或 CPU 访问与复位输入冲突，则为未定义（数据损坏）。 否则保持复位后瞬间输入的值 <sup>注</sup> 。	
I/O 线路（端口/替换功能引脚）		高阻	
片上外围 I/O 寄存器		初始化为指定状态，OCDM 寄存器保持原值。	
上述之外的片上外围功能		操作停止	在保证振荡稳定时间后可以开始工作。

<R> 注 由于支持引导变换功能，V850ES/HF2 在内部系统复位状态解除后其固件会使用一部分内部 RAM。因此，在加电复位后一些 RAM 区的内容无法保留。详情请见“17.4 复位解除后的操作”。

### 17.3.3 通过加电清零电路进行复位操作

当启用加电清零操作时会比较供电电压与检测电压。如果供电电压下降到检测电压以下（包括加电时），系统复位，各硬件设备初始化为默认状态。

复位状态从检测到电压下降开始持续到供电电压上升至检测电压以上，然后自动清零。在复位状态清零后，经过主时钟振荡器的振荡稳定时间（OSTS 寄存器的默认值： $2^{16}/f_x$ ），然后 CPU 开始程序的执行。详情请见“[第十九章加电清零电路](#)”。

### 17.3.4 通过低压检测器进行复位操作

当启用 LVI 操作且当 LVIM.LVIMD 位设置为“1”时，会对供电电压和检测电压进行比较。如果供电电压下降到检测电压以下，则系统复位，各硬件设备初始化为默认状态。

复位状态从检测到电压下降开始持续到供电电压上升至检测电压以上，然后自动清零。在复位状态清零后，经过主时钟振荡器的振荡稳定时间（OSTS 寄存器的默认值： $2^{16}/f_x$ ），然后 CPU 开始程序的执行。

详情请见“[第二十章 低压检测器](#)”。

### 17.3.5 通过时钟监视器进行复位操作

当启用时钟监视器操作时，通过采样时钟（内部振荡器）对主时钟进行监视。如果检测到主时钟停止，则系统复位，各硬件设备初始化默认状态。

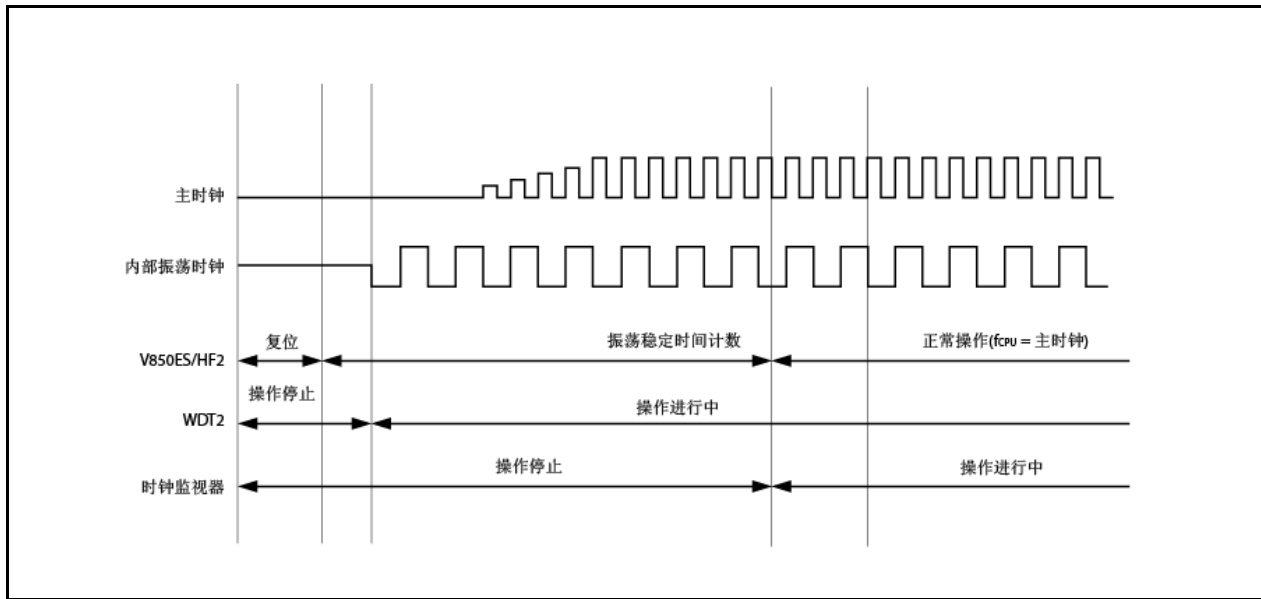
详情请见“[第十八章 时钟监视器](#)”。

#### <R> 17.4 复位解除后的操作

在复位解除后，主时钟开始振荡，在保证了振荡稳定时间（OSTS 寄存器的初始值： $2^{16}/f_x$ ）后，CPU 开始执行程序。

复位解除后，WDT2 立即用内部振荡时钟作为源时钟开始工作。

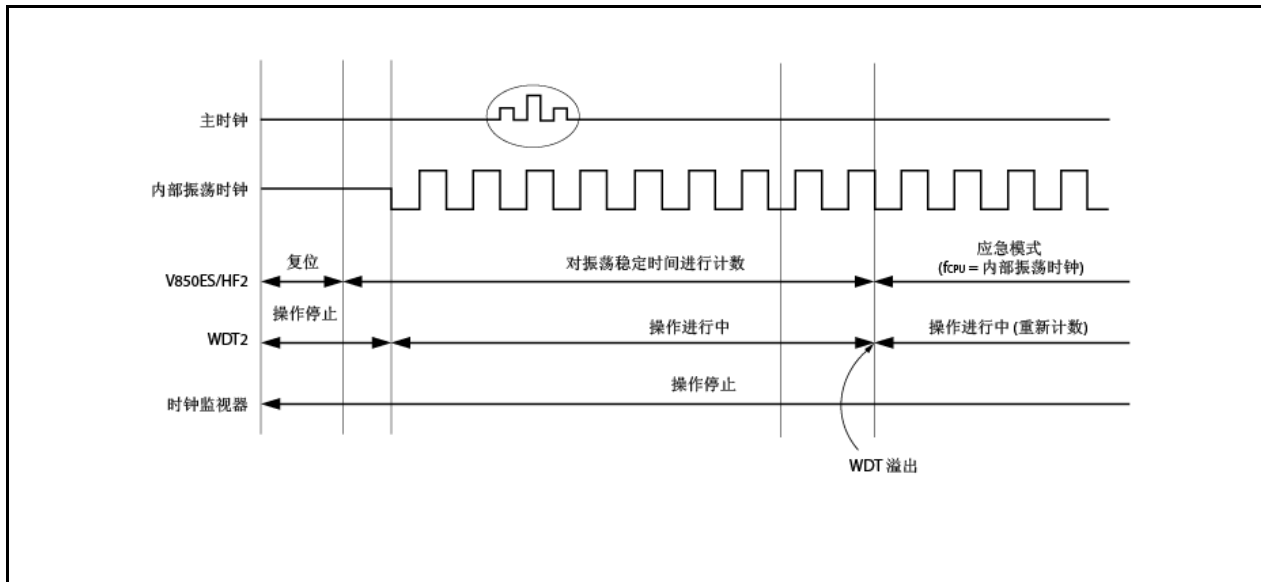
图 17-3 复位解除后的操作



##### (1) 应急工作模式

如果在振荡稳定时间得到保证前主时钟出现异常，则 WDT2 会在执行 CPU 程序前溢出。此时，CPU 开始用内部振荡时钟作为源时钟执行程序。

图 17-4 复位解除后的操作



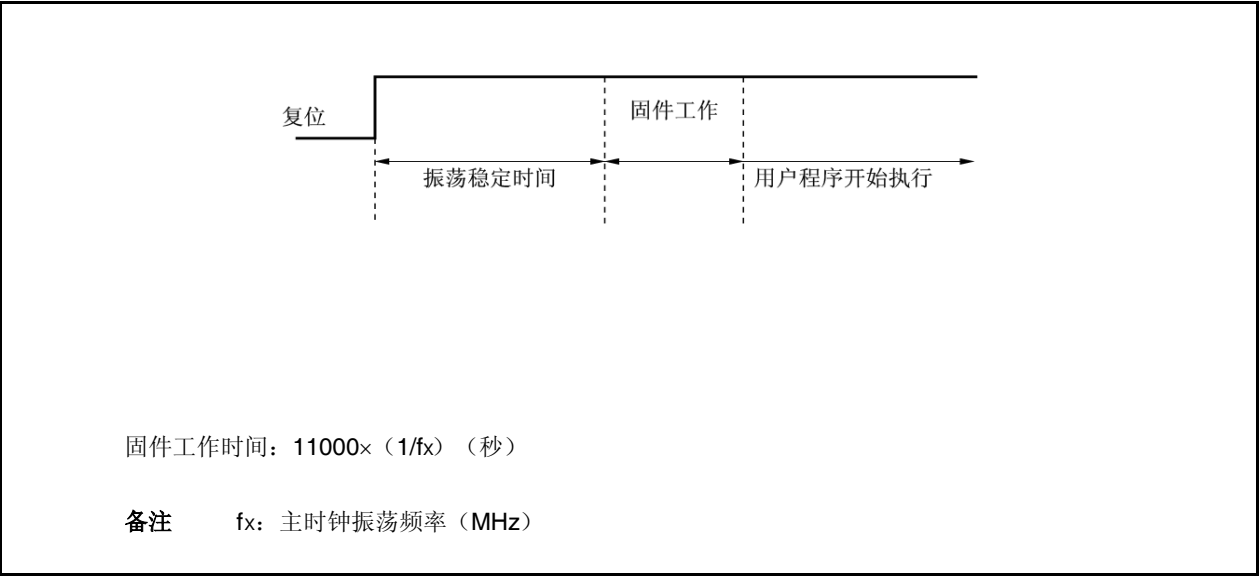
CPU 工作时钟的状态可以通过 CPU 工作时钟状态寄存器（CCLS）进行检查。



(2) 固件的工作

在 V850ES/HF2 中，在复位解除后，片上固件在用户程序开始前工作以支持引导切换功能。

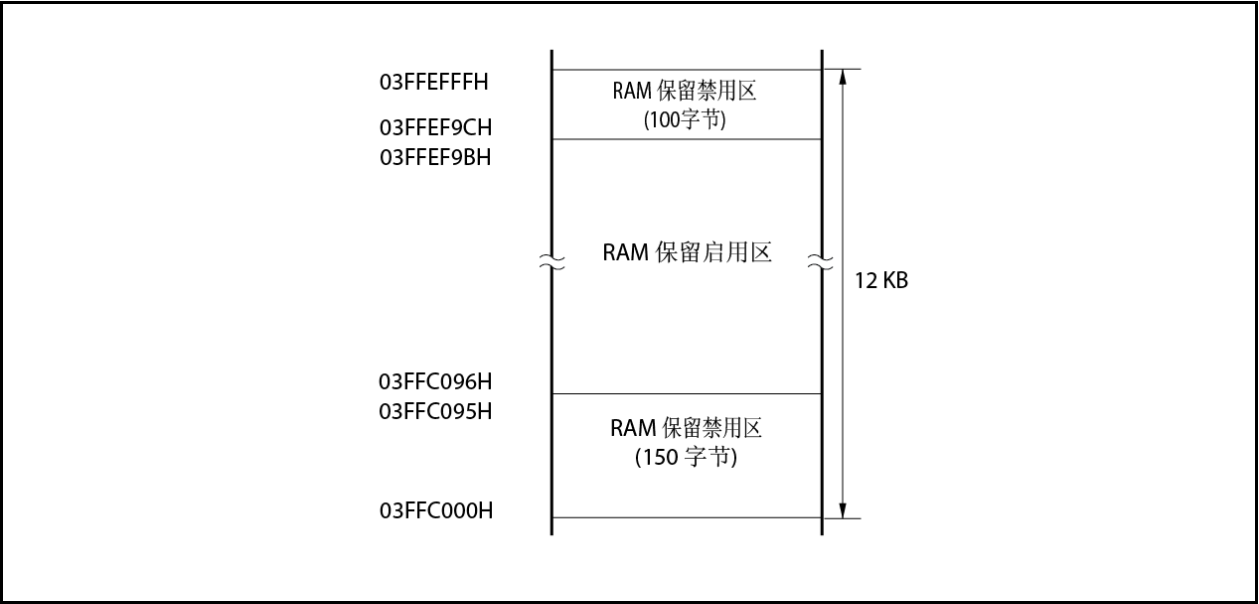
图 17-5 固件的工作



由于固件使用了一部分内部 RAM，所以下面的 RAM 区的内容即使在加电状态下也不会在复位中保留。

•12KB RAM 的版本：03FFC000H 至 03FFC095H、03FFEF9CH 至 03FFFFFFFFH

图 17-6 RAM 保留启用区



第十八章 时钟监视器

18.1 功能

时钟监视器利用内部振荡时钟对主时钟进行采样，并在主时钟停止振荡时产生复位请求信号。

一旦时钟监视器的操作通过操作启用标志启用，就不能通过非复位的方式清为 0。

当出现时钟监视器的复位时，RESF.CLMRF 位被设置。关于 RESF 寄存器的详情，请参见“17.2 用于检验复位源的寄存器”。

在下列情况下时钟监视器自动停止。

- 在 STOP 模式解除后的振荡稳定时间内
- 当主时钟停止（从子时钟工作过程中 PCC.MCK 位=1 到主时钟工作过程中 PCC.CLS 位=0）时
- 当采样时钟（内部振荡时钟）停止时
- 当 CPU 在内部振荡时钟下工作时

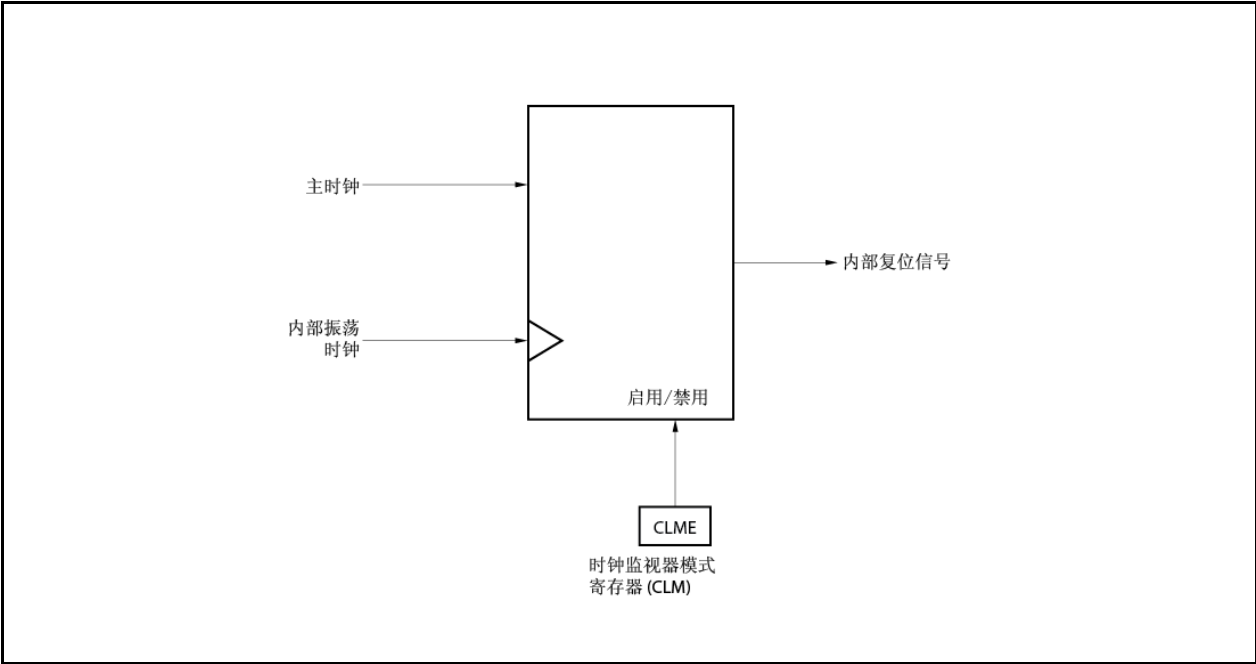
18.2 配置

时钟监视器包含下列硬件。

表 18-1 时钟监视器的配置

项	配置
控制寄存器	时钟监视器模式寄存器（CLM）

图 18-1 时钟监视器的框图



18.3 寄存器

时钟监视器由时钟监视器模式寄存器（CLM）控制。

(1) 时钟监视器模式寄存器（CLM）

CLM 寄存器为专用寄存器。只能通过特殊序列组合进行写入（见“3.4.7 专用寄存器”）。

该寄存器用于设置时钟监视器的工作模式。

该寄存器可进行 8 位元或 1 位元的读取或写入。

复位将此寄存器设置为 00H。

复位后：00H      读/写      地址：FFFFFF870H								
	7	6	5	4	3	2	1	0
CLM	0	0	0	0	0	0	0	CLME
CLME		启用或禁用时钟监视器的工作						
0		禁用时钟监视器的工作。						
1		启用时钟监视器的工作。						

- 注意事项
- 1. 一旦 CLME 位设置为 1，就无法通过除复位之外的其它方式清为 0。
  - 2. 当出现时钟监视器的复位时，CLME 位清为 0，RESF.CLMRF 位设置为 1。

18.4 操作

本节说明时钟监视器的功能。启动及停止条件如下。

<启动条件>

通过将 CLM.CLME 位设置为 1 启用操作

<停止条件>

- 当 STOP 模式解除后正在对振荡稳定时间计数时
- 当主时钟停止（从子时钟工作过程中 PCC.MCK 位=1 到主时钟工作过程中 PCC.CLS 位=0）时
- 当采样时钟（内部振荡时钟）停止时
- 当 CPU 在内部振荡时钟下工作时

表 18-2 时钟监视器的工作状态  
(当 CLM.CLME 位=1 且在内部振荡时钟工作过程中)

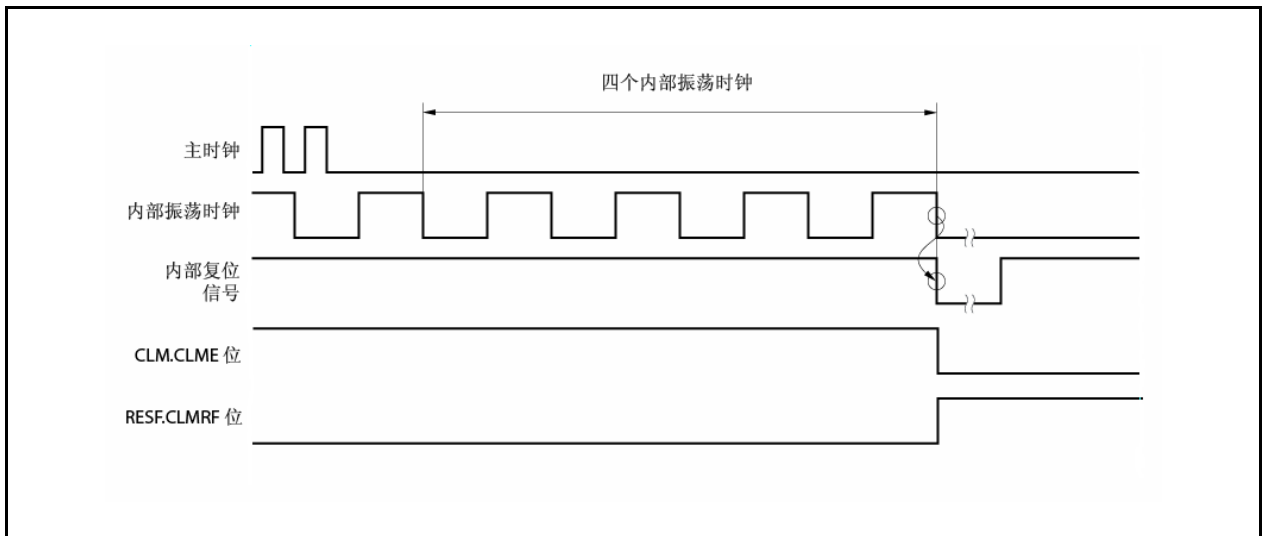
CPU 工作时钟	工作模式	主时钟状态	内部振荡时钟的状态	时钟监视器的状态
主时钟	HALT 模式	振荡	振荡 <sup>注1</sup>	工作 <sup>注2</sup>
	IDLE1、IDLE2 模式	振荡	振荡 <sup>注1</sup>	工作 <sup>注2</sup>
	STOP 模式	停止	振荡 <sup>注1</sup>	停止
子时钟（PCC 寄存器的 MCK 位=0）	子 IDLE 模式	振荡	振荡 <sup>注1</sup>	工作 <sup>注2</sup>
子时钟（PCC 寄存器的 MCK 位=1）	子 IDLE 模式	停止	振荡 <sup>注1</sup>	停止
内部振荡时钟	—	停止	振荡 <sup>注1</sup>	停止
复位中	—	停止	停止	停止

- 注
1. 利用选项字节功能（见二十三章）启用内部振荡器停止并将 RCM.RSTOP 位设置为 1，可停止内部振荡器。
  2. 当内部振荡器停止时，时钟监视器停止。

### (1) 当主时钟振荡停止时的操作 (CLME 位=1)

如果当 CLME 位=1 时主时钟的振荡停止，则会产生如图 18-2 所示的内部复位信号。

图 18-2 由于主时钟停止振荡产生的复位时段

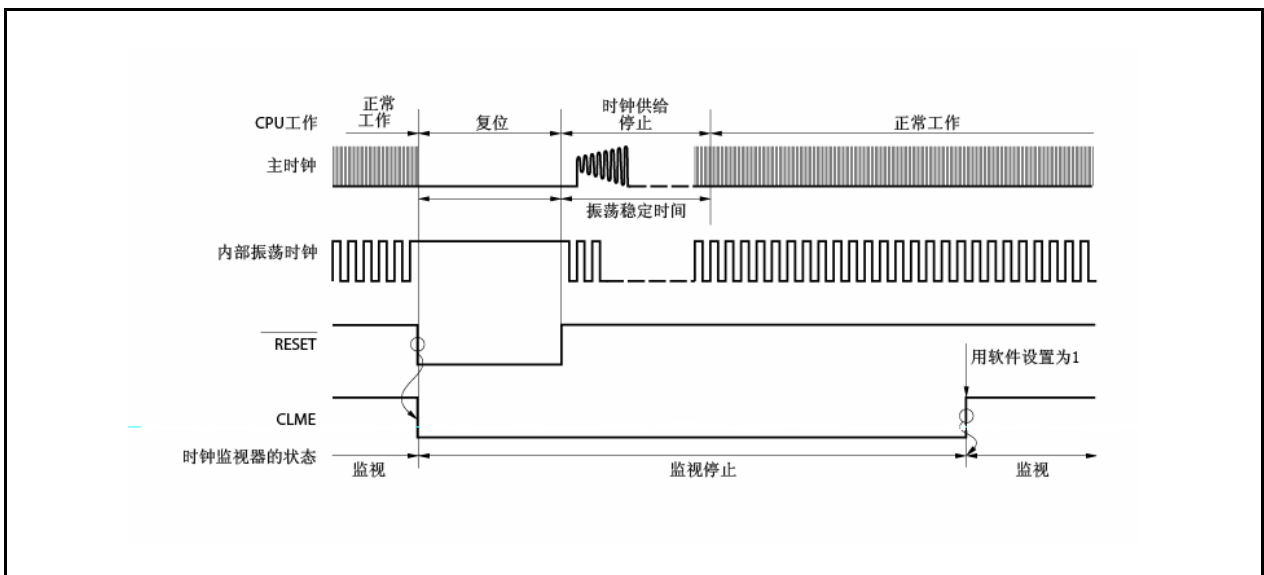


### (2) Reset 输入后的时钟监视器状态

RESET 输入将 CLM.CLME 位清为 0 并停止时钟监视器的工作。在主时钟振荡稳定时间结束后用软件将 CLME 位设置为 1 时，监视开始。

图 18-3 Reset 输入后的时钟监视器状态

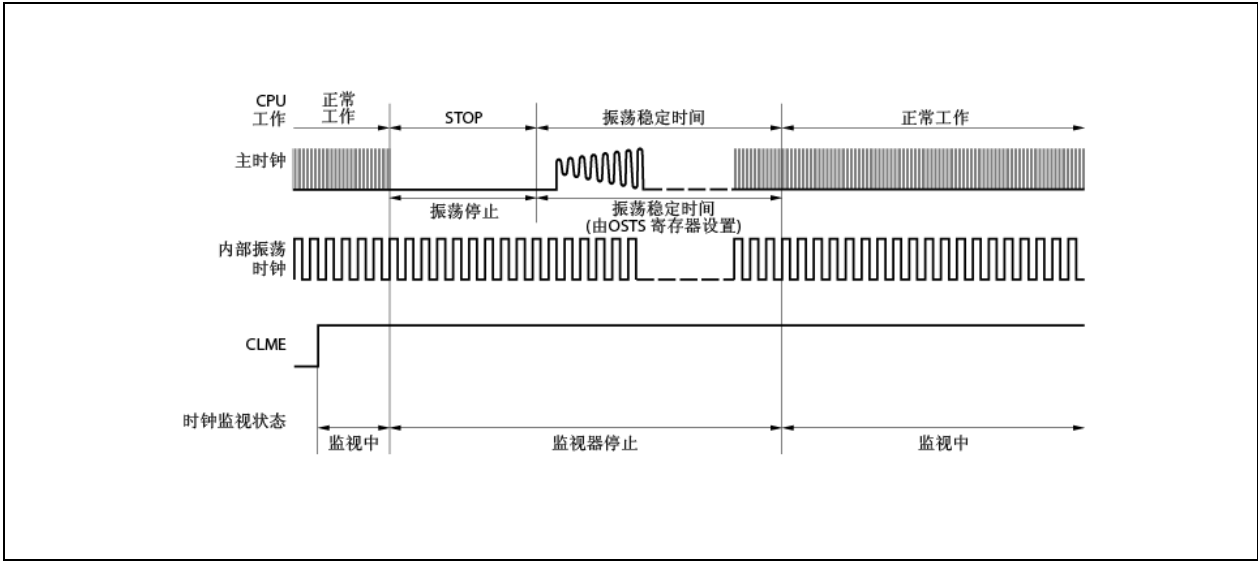
(在 RESET 输入后及主时钟振荡稳定时间结束时将设置 CLM.CLME 位=1)



(3) 在 STOP 模式下或 STOP 模式解除后的操作

如果在 CLM.CLME 位=1 时设置为 STOP 模式，则在 STOP 模式下正在对振荡稳定时间进行计数时监视操作停止。在振荡稳定时间后，监视操作自动开始。

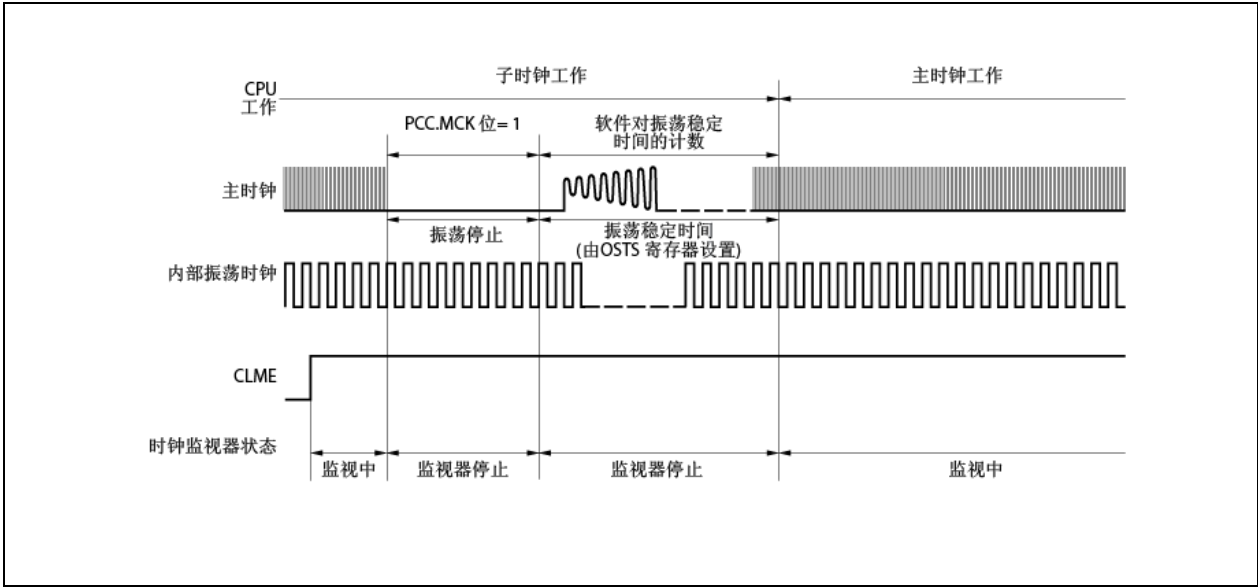
图 18-4 在 STOP 模式下或 STOP 模式解除后的操作



(4) 主时钟停止时的操作（强制）

在子时钟工作过程中（PCC.CLS 位=1）或通过将 PCC.MCK 位设置为 1 停止主时钟时，监视操作会一直停止到主时钟开始工作（PCC.CLS 位=0）。当主时钟开始工作时监视操作自动开始。

图 18-5 主时钟停止时的操作（强制）



(5) 当 CPU 工作在内部振荡时钟下时的操作（CCLS.CCLSF 位=1）

当 CCLSF 位为 1 时，即使 CLME 位设置为 1 监视操作也不会停止。

## 19.1 功能

上电清零（POC）电路的功能如下所示。

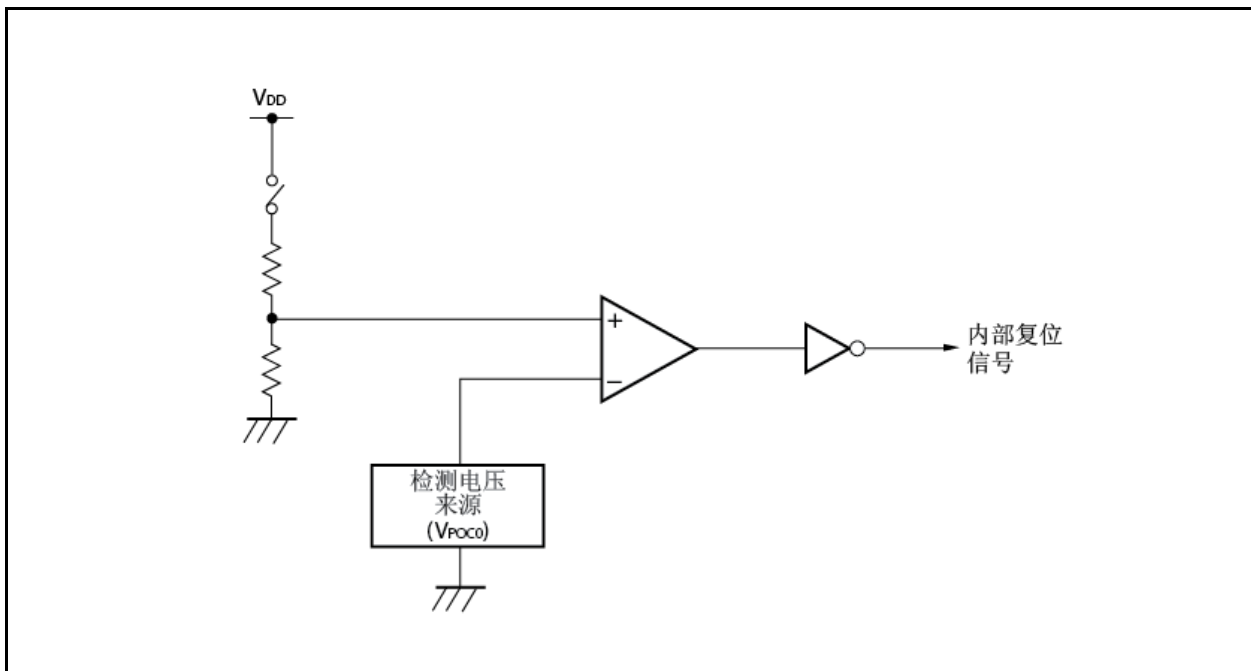
- 在上电时生成复位信号。
- 比较供电电压（ $V_{DD}$ ）和检测电压（ $V_{POC0}$ ），当  $V_{DD} < V_{POC0}$  时生成复位信号（检测电压（ $V_{POC0}$ ）： $3.7V \pm 0.2V$ ）。

- 备注**
1. V850es/HF2 有多个用来产生内部复位信号的内部硬件设备。当系统由看门狗定时器 2（WDT2RES）、低压检测器（LVI）或时钟监视器（CLM）复位时，对应于复位源的标志将被分配到复位源标志寄存器（RESF）中。  
当内部复位信号由 WDT2RES、LVI 或时钟监视器生成时，RESF 寄存器不会清零，其对应于复位源的标志设置为 1。关于 RESF 寄存器的详情请参见“**第十七章 复位功能**”。
  2. 如果外接谐振器的工作频率为 5MHz，则从上电到程序开始执行的时间为“从上电到解除复位的时间+16ms”。但是，该时间会根据外部来源的不同而不同（例如微处理器供电电压的状态、谐振器的稳定时间等）。

## 19.2 配置

框图如下所示。

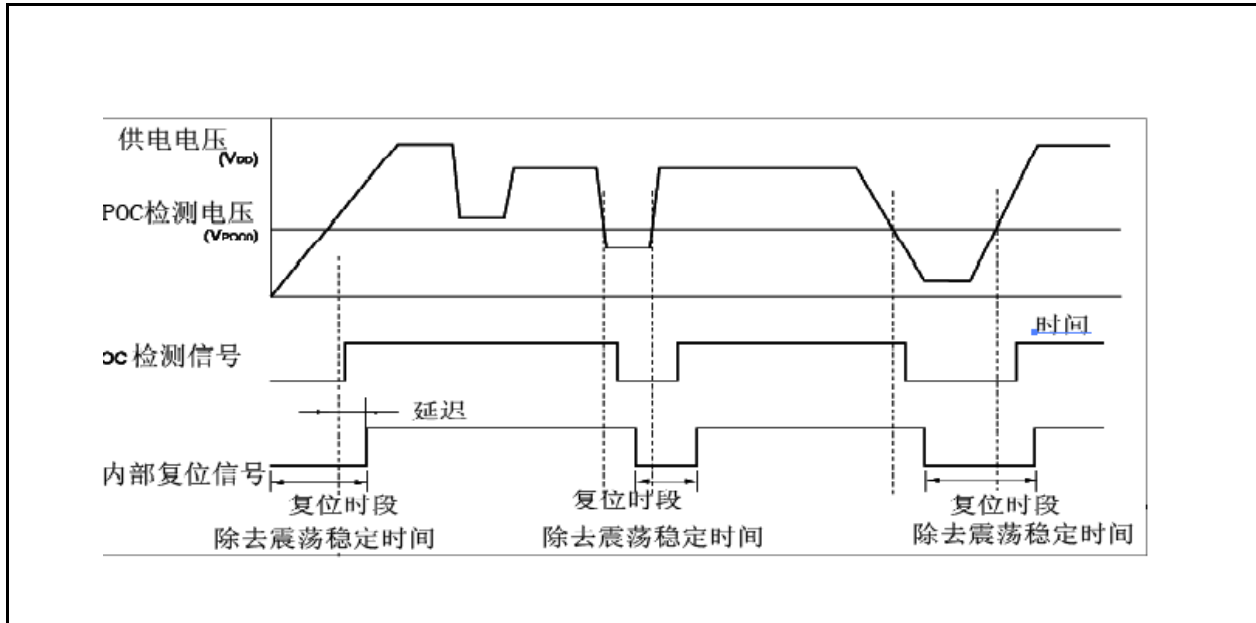
图 19-1 上电清零电路的框图



### 19.3 操作

在对供电电压和检测电压进行比较时，如果供电电压低于检测电压（包括在上电时），系统复位，各硬件回到特定的状态。

图 19-2 上电清零电路产生复位信号的定时





## 第二十章 低电压检测器

### 20.1 功能

低电压检测器（LVI）具有下列功能。

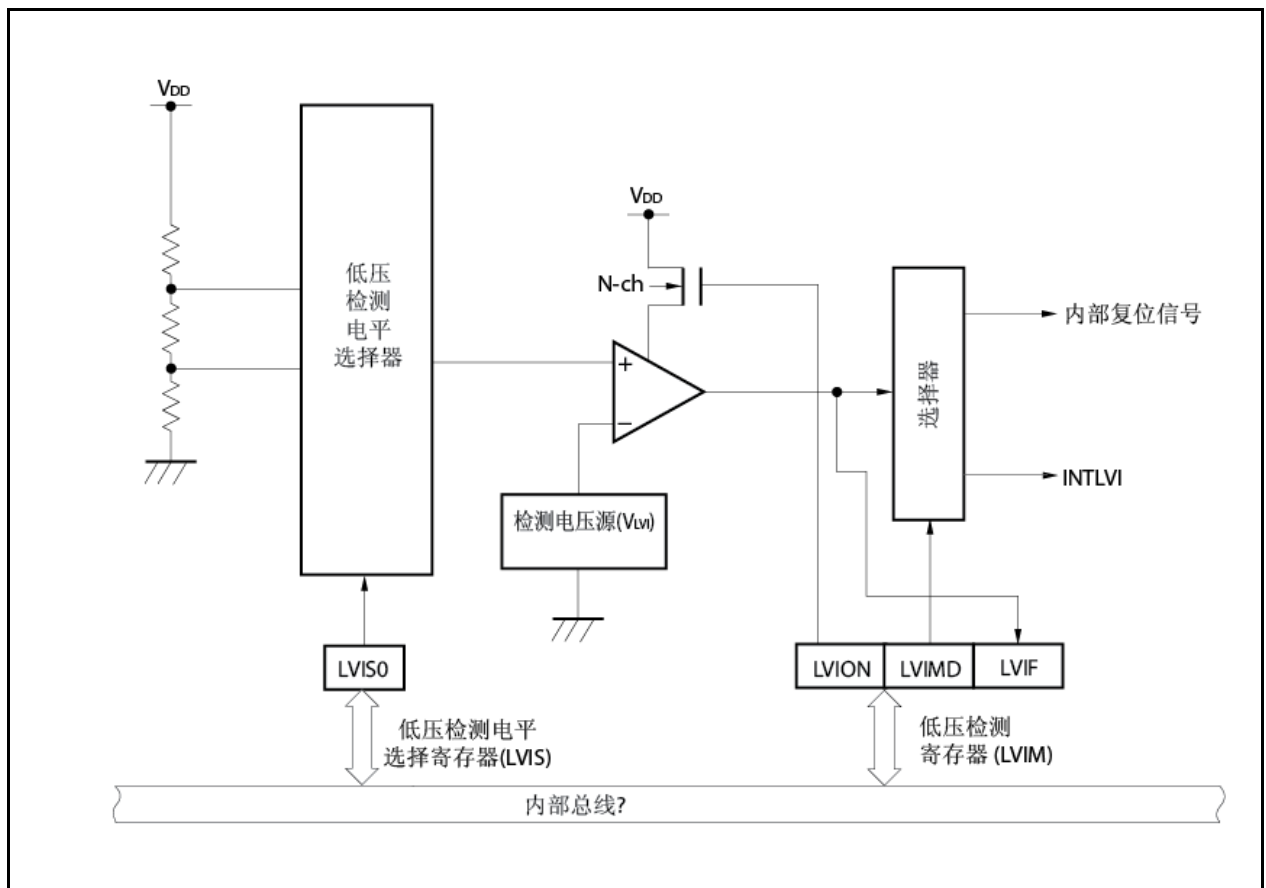
- 比较供电电压（ $V_{DD}$ ）和检测电压（ $V_{LVI}$ ），并在  $V_{DD} < V_{LVI}$  时产生中断请求信号或内部复位信号。
- 要检测的供电电压的高低可通过软件更改（两步）。
- 可选择中断请求信号或内部复位信号。
- 可工作在 STOP 模式下。
- 可通过软件停止操作。

如果用低电压检测器产生复位信号，则当复位信号产生时 RESF.LVIRF 位设置为 1。关于 RESF 寄存器的详情请参见“第十七章复位功能”。

### 20.2 配置

框图如下所示。

图 20-1 低电压检测器的框图



20.3 寄存器

(1) 低压检测寄存器 (LVIM)

LVIM 寄存器用于启用或禁用低压检测，以及用于设置低电压检测器的工作模式。LVIM 寄存器为专用寄存器。它只能通过特定序列的组合进行写入（见“3.4.7 专用寄存器”）。该寄存器可进行 8 位元或 1 位元的读取或写入。但是，0 位是只读的。

复位后: 00H

读/写

地址: FFFFF890H

	7	6	5	4	3	2	1	0
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF

LVION	启用或禁用低压检测的工作
0	禁用工作。
1	启用工作。

LVIMD	选择低压检测的工作模式
0	当供电电压 <检测电压时产生中断请求信号 INTLVI。
1	当供电电压 <检测电压时产生内部复位信号 LVIREs。

LVIF	低压检测标志
0	当供电电压 >检测电压时，或当禁用操作时
1	供电电压 <检测电压

- 注意事项
- 在将 LVION 位设置为 1 后，在用 LVIF 位检查电压之前应等待 0.2ms（最多）。
  - 当 LVION 位=1、LVIMD 位=0 时，LVIF 标志的值作为输出信号 INTLVI 输出。
  - 一定要将 2 至 6 位清为“0”。
  - 在将 LVIM.LVION、LVIM.LVIMD 位设置为 1 后，在由于非低压检测导致的复位请求出现之前低电压检测器无法停止。

<R>

(2) 低压检测电平选择寄存器 (LVIS)

LVIS 寄存器用于选择要检测的低压的电平。  
该寄存器可进行 8 位读写。

复位后: 00H		读/写		地址: FFFF891H				
	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	0	0	LVIS0

LVIS0	检测电平
0	4.4V±0.2V
1	4.2V±0.2V

- 注意事项
1. 在将 LVIM.LVION、LVIM.LVIMD 位设置为 1 后，在由于非低压检测导致的复位请求出现之前该寄存器无法写入。
  2. 一定要将 1 至 7 位清为“0”。

(3) 内部 RAM 数据状态寄存器 (RAMS)

RAMS 寄存器为标志寄存器，用于指示内部 RAM 是否有效。RAMS 寄存器为专用寄存器。它只能通过特定序列的组合进行写入（见“3.4.7 专用寄存器”）。  
关于 RAMS 寄存器请见“20.5RAM 保持电压检测操作”。  
该寄存器可进行 8 位元或 1 位元的读取或写入。

注意事项 下面展示了复位后的特定序列。

复位后: 01H		读/写		地址: FFFFF892H				
	7	6	5	4	3	2	1	0
RAMS	0	0	0	0	0	0	0	RAMF

RAMF	内部 RAM 数据有效/无效
0	有效
1	无效

## 20.4 操作

根据 LVIM.LVIMD 位的设置，会产生中断请求信号（INTLVI）或内部复位信号。

### 20.4.1 要用于内部复位信号时

<启动操作>

<1>屏蔽 LVI 的中断。

<2>利用 LVIS.LVIS0 位选择要检测的电平。

<3>将 LVIM.LVION 位设置为 1（启用操作）。

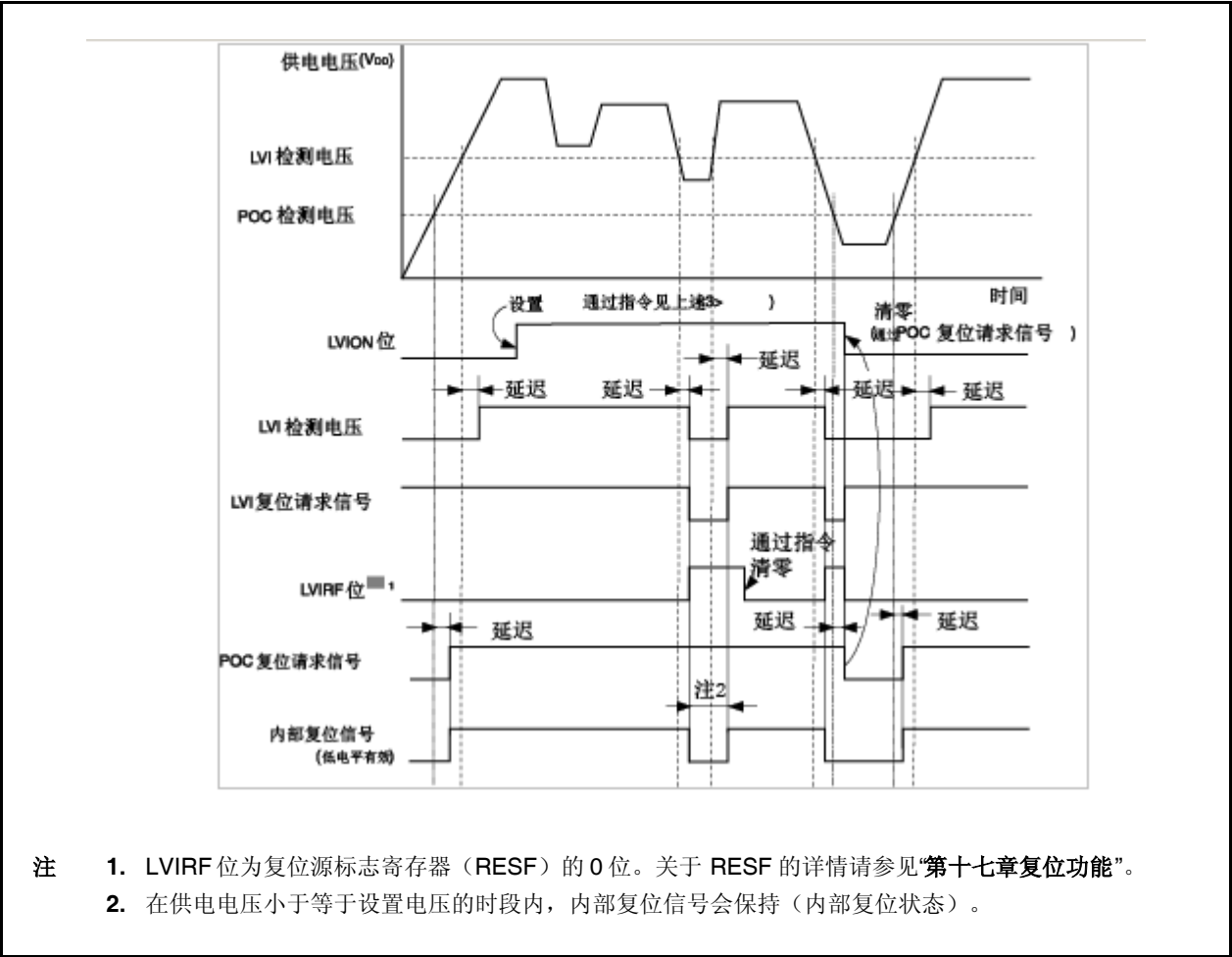
<4>用软件插入最大 0.2ms 的等待周期。

<5>利用 LVIM.LVIF 位检验是否供电电压>检测电压。

<6>将 LVIM.LVIMD 位设置为 1（以生成内部复位信号）。

**注意事项** 如果 LVIMD 位设置为 1，在非 LVI 的复位请求产生之前 LVIM 和 LVIS 寄存器的内容不会发生改变。

图 20-2 低电压检测器的操作定时 (LVIMD 位=1)



<R>

## 20.4.2 用于中断时

<启动操作>

<1>屏蔽 LVI 的中断。

<2>利用 LVIS.LVIS0 位选择要检测的电平。

<3>将 LVIM.LVION 位设置为 1（启用操作）。

<4>用软件插入最大 0.2ms 的等待周期。

<5>利用 LVIM.LVIF 位检验是否供电电压>检测电压。

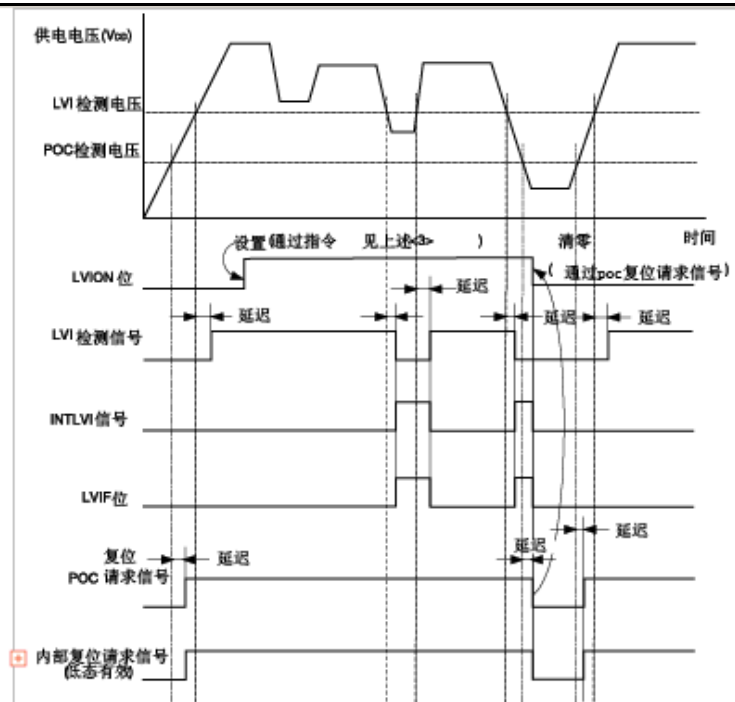
<6>将 LVI 的中断请求标志清零。

<7>去除对 LVI 中断的屏蔽。

<停止操作>

将 LVION 位清为 0。

图 20-3 低电压检测器的操作定时（LVIM 位=0）

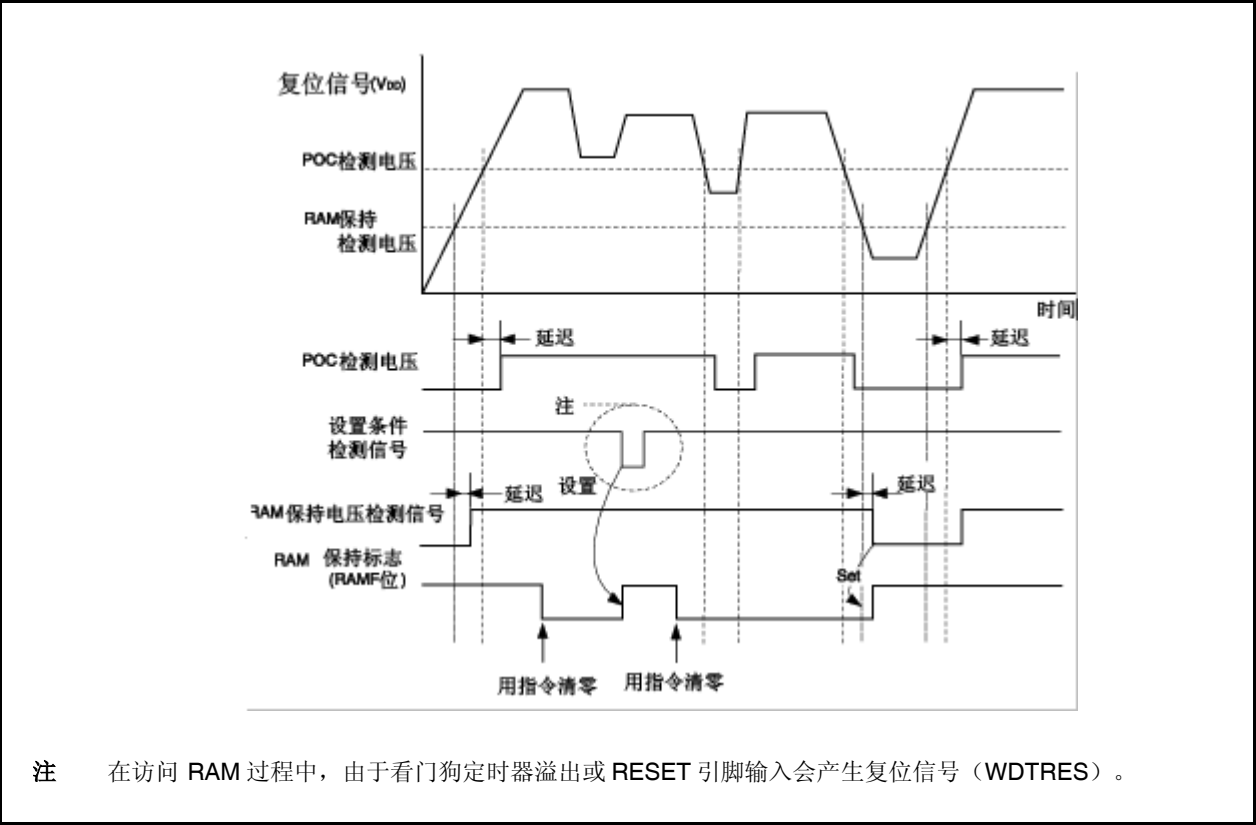


20.5 RAM 保持电压检测操作

对供电电压和检测电压进行比较。当供电电压下降到检测电压以下时（包括加电时），RAMS.RAMF 位被设置（1）。

当未使用 POC 功能且使用 RAM 保持电压检测功能时，如果检测到的电压下降到工作电压以下，就一定要输入外部复位信号。

图 20-4 Ram 保持电压检测功能的操作定时



20.6 仿真功能

当使用内电路仿真器时，RAM 保持标志（RAMS.RAMF 位）的操作可通过在调试器中操作 PEMU1 寄存器进行伪控制和仿真。

该寄存器仅在仿真模式下有效。在正常模式下无效。

（1）外围仿真寄存器 1（PEMU1）

复位后: 00H	读/写	地址: FFFFF9FEH						
	7	6	5	4	3	2	1	0
PEMU1	0	0	0	0	0	EVARAMIN	0	0
EVARAMIN	对 RAM 保持电压检测信号的伪指定							
0	不检测低于 RAM 保持电压的电压。							
1	检测低于 RAM 保持电压的电压（设置 RAMF 标志）。							

注意事项 该位不会自动清零。

[用法]

在使用内电路仿真器时，通过在调试器中重写该寄存器实现 RAMF 的伪仿真。

- <1> CPU 暂停（CPU 停止工作）
- <2> 利用寄存器写命令将 EVARAMIN 位设置为 1。  
 通过将 EVARAMIN 位设置为 1，在硬件上将 RAMF 位设置为 1（内部 RAM 数据无效）。
- <3> 利用寄存器写命令将 EVARAMIN 位设置为 0。  
 除非进行该操作（将 EVARAMIN 位清为 0），否则无法通过 CPU 操作指令将 RAMF 位清为 0。
- <4> 运行 CPU，继续仿真。

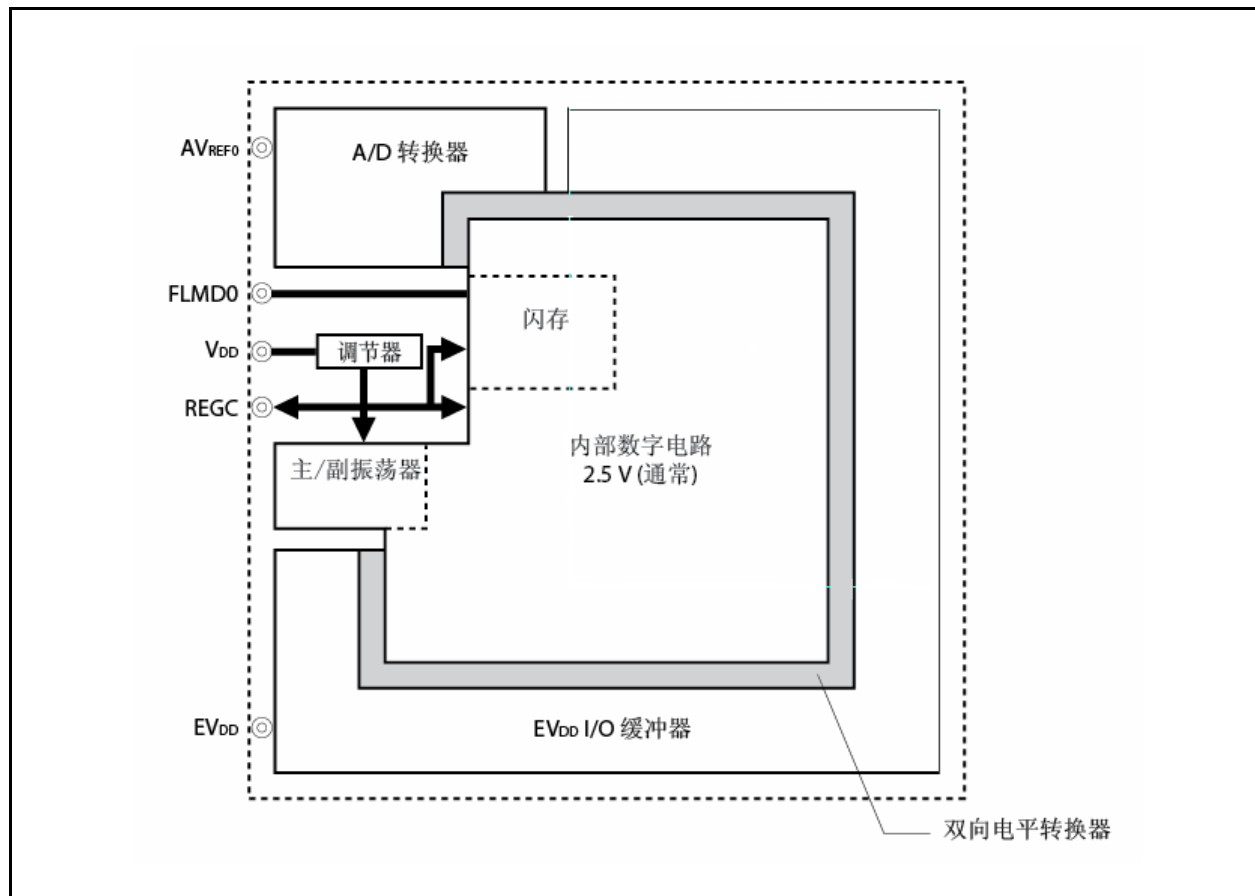


## 21.1 概述

V850ES/HF2 集成了一个稳压器以降低功耗和噪音。

该稳压器为振荡器和内部逻辑电路(A/D 转换器, D/A 转换器以及输出缓冲器除外)提供一低跃变电源供电电压  $V_{DD}$ 。其输出电压设置为 2.5 V (典型值)。

图 21-1. 稳压器



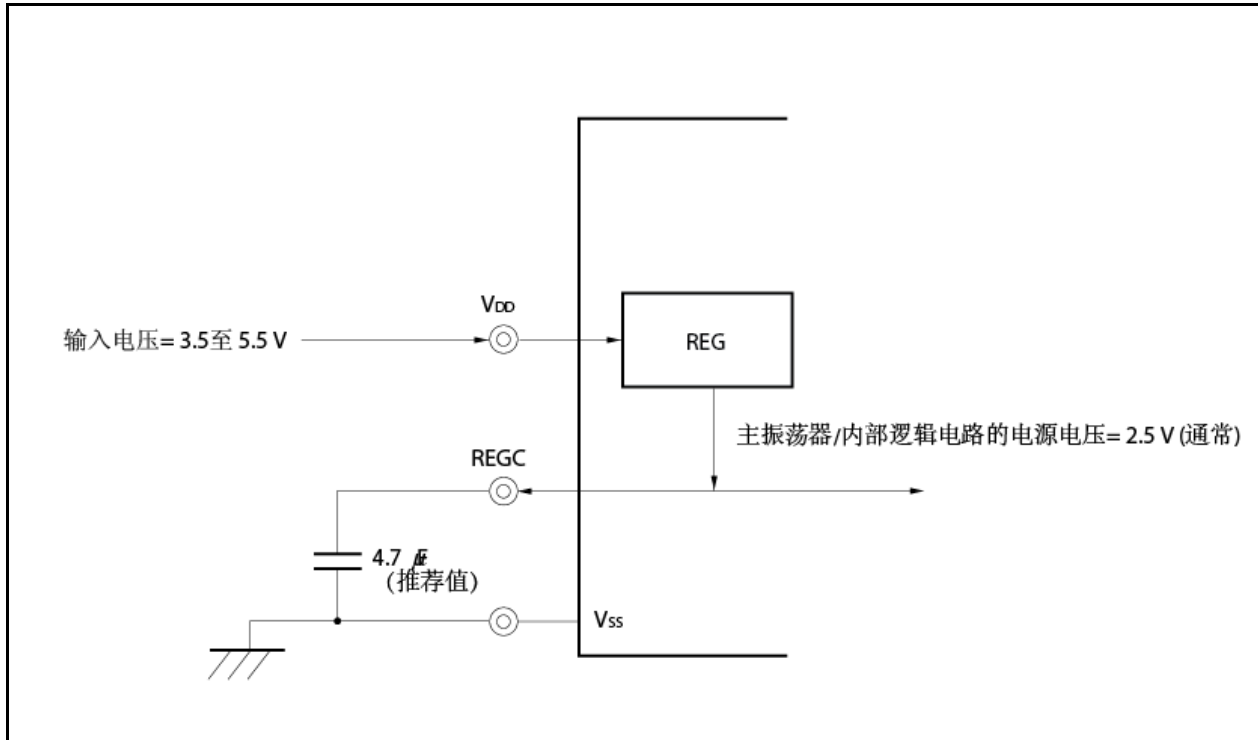
## 21.2 运行

这种产品稳压器可以运行在任何模式下(通常操作模式, HALT 模式, IDLE1 模式, IDLE2 模式, STOP 模式或在复位期间)。

确保在 REGC 引脚连接一个电容(4.7  $\mu\text{F}$ ) (推荐值) 以稳定稳压器输出电压。

稳压器引脚连接方法如下所示:

图 21-2. REGC 引脚连接



## 第二十二章 闪存

闪存形式为开发环境和大量生产应用提供了以下优点。

- 用于在V850ES/HF2焊接到目标系统上之后更改软件。
- 用于在开始大量生产时的数据调整。
- 用于根据各种型号小型生产中的规格区分软件。
- 便于库存管理。
- 用于在装运之后更新软件。

### 22.1 特性

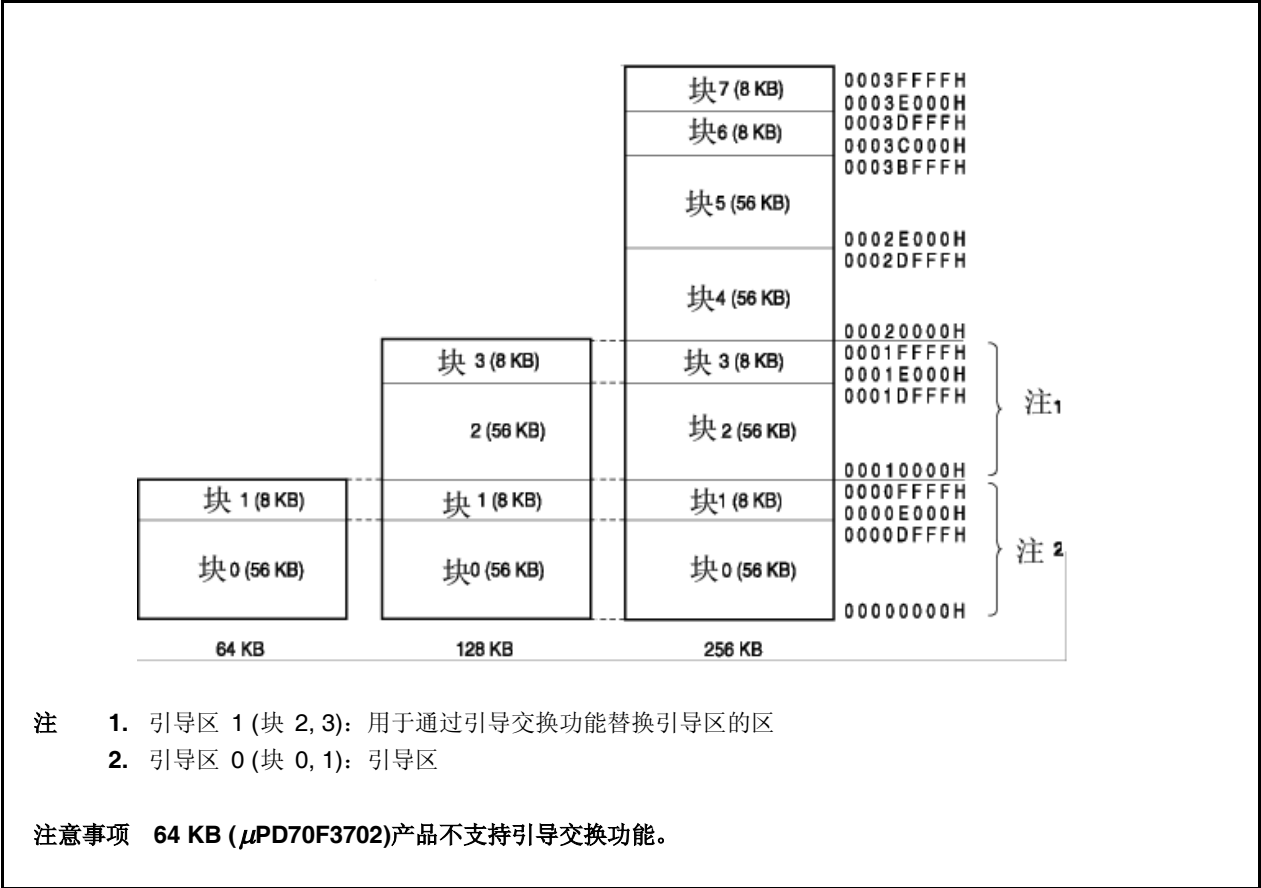
- 4-字节/1-时钟访问 (取指令时)
- 容量: 256 KB/128 KB/64 KB
- 写电压:单电源擦/写
- 重写方法
  - 经串行接口通过与专用闪存编程器通信重写 (主存储器/扩展存储器编程)
  - 通过用户程序重写闪存(自编程)
- 支持闪存写禁用功能(安全功能)
- 通过使用引导交换功能的自编程安全重写整个闪存区
- 自编程期间可以确认中断。

22.2 存储器配置

256K，128K和64K内部闪存区划分为8，4和2块,而且可以按块单元进行编程/擦除。还可以一次擦除所有块。

当使用引导交换功能时，位于0引导区的地址的物理存储器（块0，1）由位于1引导区的地址的物理存储器（块2，3）替换。有关引导交换功能的详细信息，请参阅22.5节的通过自编程重写。

图 22-1. 闪存映射



22.3 功能概述

V850ES/HF2的内部闪存可以使用专用闪存编程器的重写功能进行重写，而不管V850ES/HF2是否已经安装在目标系统上 (扩展存储器/主存储器编程)。

另外，还支持禁止将用户程序重写入内部闪存的安全功能，使未授权人不能改变程序。

使用用户程序的重写功能(自编程)对于假定目标系统生产/装运之后改变程序的应用程序比较理想。还支持安全地重写整个闪存区的引导交换功能。另外，自编程期间支持中断服务，以便在各种情况下可以重写闪存，例如在与外部设备连接时。

表 22-1. 重写方法

重写方法	功能概述	运行模式
主存储器编程	闪存可以在设备安装在目标系统上之后使用专用闪存编程器重写。	闪存编程模式
扩展存储器编程	闪存可以在设备安装在目标系统上之前使用专用闪存编程器和专用程序适配板（FA 系列）重写。	
自编程	闪存可以通过执行经过主存储器/扩展存储器编程已经提前写入存储器的用户程序重写。(自编程期间，不能从内部闪存区获取指令或对内部闪存区进行数据访问。因此，重写程序必须提前传输到内部RAM或外部存储器)。	正常运行模式

备注      FA 系列是Naito Densei Machida Mfg. Co., Ltd.公司的产品。

表 22-2. 基本功能

功能	功能概述	支持(○：支持， ×：不支持)	
		主存储器/扩展存储器编程	自编程
块擦除	擦除指定存储器块的内容。	○	○
芯片擦除	立即擦除整个存储区的内容。	○	×
写	写入指定地址，并进行校验检查以查看写电平是否安全。	○	○
校验/校验和	将从闪存中读出的数据与从闪存编程器中传输的数据进行比较。	○	× (可以由用户程序读取)
空检查	检查整个存储器的擦除状态。	○	○
安全设置	禁止使用块擦除命令，芯片擦除命令，编程命令和读命令	○	× (仅在设置从可用变成禁用时支持)

下表列出了安全功能。装运之后默认启用块擦除命令禁用，芯片擦除命令禁用，编程命令禁用，而且经过主存储器/扩展存储器编程通过重写可以设置安全性。每种安全功能可以结合其他安全功能同时使用。

表22-3. 安全功能

功能	功能概述
块擦除命令禁用	禁止对所有块执行块擦除命令。可以通过执行芯片擦除命令初始化禁用设置。
芯片擦除命令禁用	禁止对所有块执行块擦除和芯片擦除命令。一旦设置禁用，则因为不能执行芯片擦除命令，所以所有禁用设置不能初始化。
编程命令禁用	禁止对所有块执行编程和块擦除命令。可以通过执行芯片擦除命令初始化禁用设置。
读命令禁用	禁止对所有块执行读命令。可以通过执行芯片擦除命令初始化禁用设置。
重写引导区禁用	不支持

表 22-4. 安全设置

功能	设置每个安全功能时，擦除，写，读操作 (√: 可执行, ×: 不可执行, —: 不支持)		有关安全设置的注释	
	主存储器/ 扩展存储器编程	自编程	主存储器/ 扩展存储器编程	自编程
块擦除命令禁用	块擦除命令: × 芯片擦除命令: √ 编程命令: √ 读取命令: √	块擦除 (FlashBlockErase): √ 芯片擦除: — 写 (FlashWordWrite): √ 读 (FlashWordRead): √	可以通过芯片擦除命令初始化禁用设置。	仅在设置从启用变为禁用时支持
芯片擦除命令禁用	块擦除命令: × 芯片擦除命令: × 编程命令: √ <sup>注</sup> 读命令: √	块擦除 (FlashBlockErase): √ 芯片擦除: — 写 (FlashWordWrite): √ 读 (FlashWordRead): √	不能初始化禁用设置。	
编程命令禁用	块擦除命令: × 芯片擦除命令: √ 编程命令: × 读命令: √	块擦除 (FlashBlockErase): √ 芯片擦除: — 写 (FlashWordWrite): √ 读 (FlashWordRead): √	可以通过芯片擦除命令初始化禁用设置。	
读命令禁用	块擦除命令: √ 芯片擦除命令: √ 编程命令: √ 读命令: —	块擦除 (FlashBlockErase): √ 芯片擦除: — 写 (FlashWordWrite): √ 读 (FlashWordRead): √	可以通过芯片擦除命令初始化禁用设置。	

**注** 在这种情况下，由于擦除命令无效，所以不能写入不同于已经写入闪存中数据的数据。

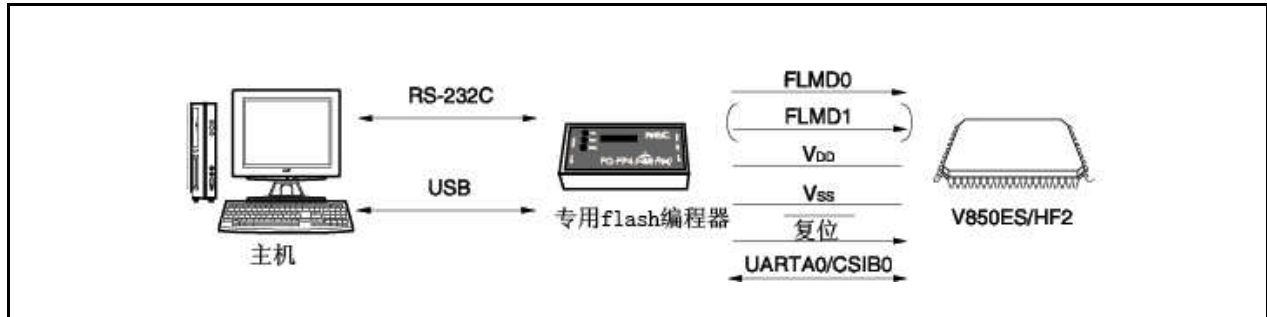
## 22.4 由专用闪存编程器重写

闪存可以在 V850ES/HF2 安装在目标系统(主存储器编程)上之后使用专用闪存编程器重写。闪存还可以在设备安装在目标系统(扩展存储器编程)上之前使用专用程序适配器 (FA 系列)重写。

### 22.4.1 编程环境

以下给出了将程序写入 V850ES/HF2 的闪存所需的环境。

图 22-2. 将程序写入闪存所需要的环境



需要主机控制专用闪存编程器。

UARTA0 或 CSIB0 用于专用闪存编程器与 V850ES/HF2 之间的接口以执行写，擦除等操作。需要用于扩张存储器写入的专用程序适配器 (FA 系列)。

- FA-70F3704GK-9EU-MX (已连线)
- FA-80GK-9EU-A (未连线:需要连线)

**备注** FA 系列是 Naito Densei Machida Mfg. Co., Ltd. 公司的产品。



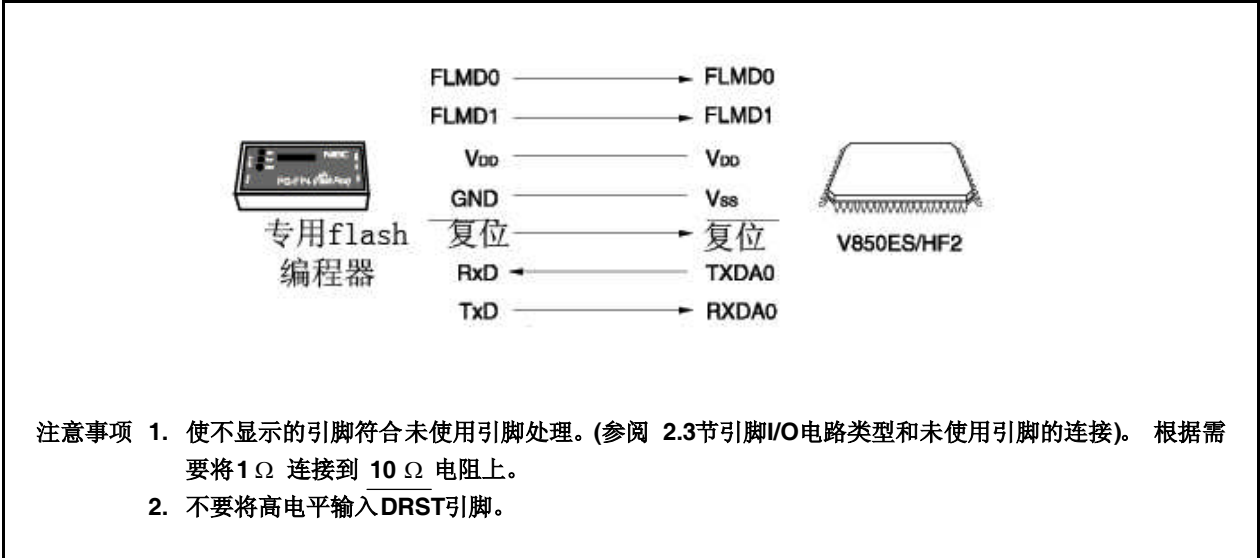
22.4.2 通信模式

在专用闪存编程器和V850ES/HF2之间的通信使用V850ES/HF2的UARTA0 或CSIB0接口通过串口通信进行。

(1) UARTA0

传输速度: 9,600, 19,200, 31,250, 38,400, 76,800, 153,600 bps  
(不支持57,600, 115,200, 和128,000 bps)

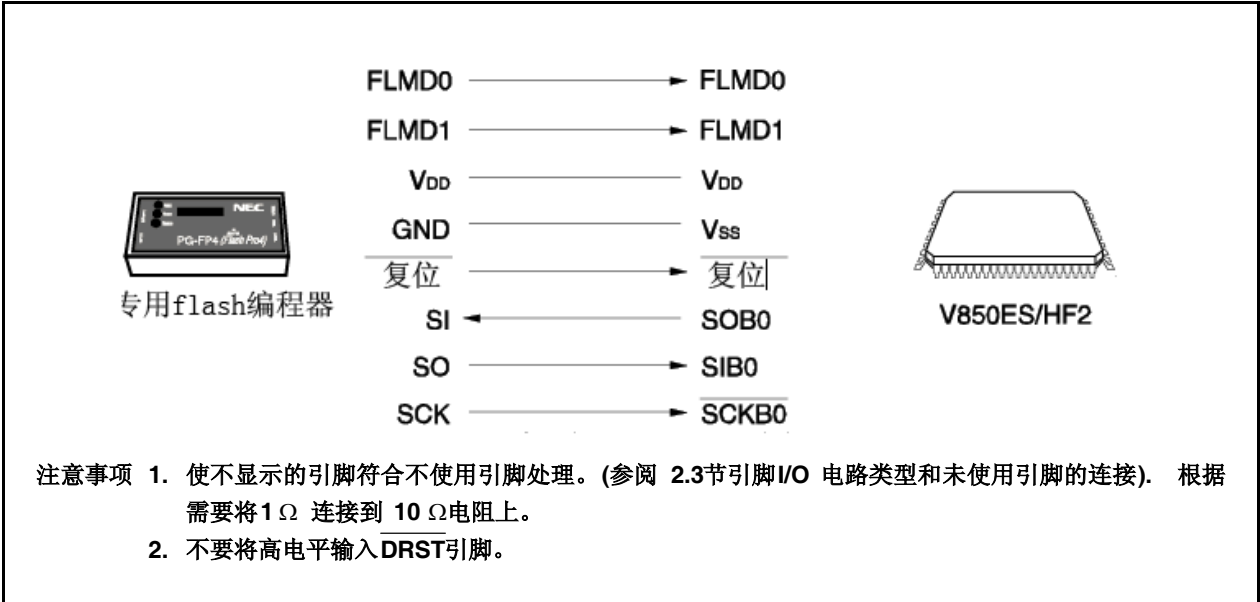
图 22-3. 与专用闪存编程器(UARTA0)的通信



(2) CSIB0

串行时钟: 2.4 kHz 至 2.5 MHz (首先MSB)

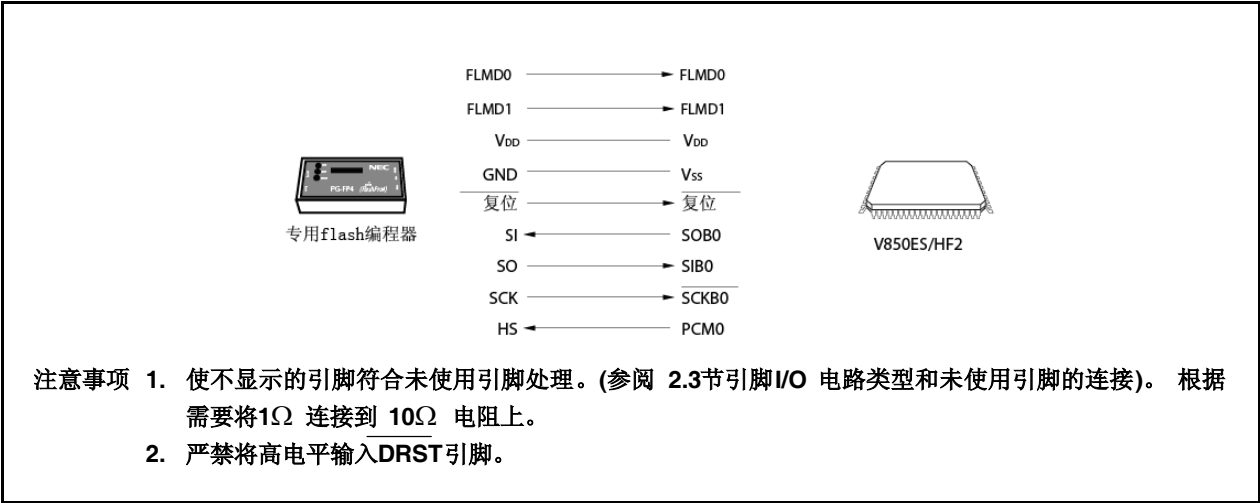
图 22-4. 与专用闪存编程器(CSIB0) 的通信



(3) CSIB0 + HS

串口时钟: 2.4 kHz 至 2.5 MHz (首先MSB )

图 22-5. 与专用闪存编程器 (CSIB0 + HS)的通信



专用闪存编程器输出传输时钟且V850ES/HF2作为从属设备操作。

当PG-FP4用作专用闪存编程器时，其产生到V850ES/HF2的以下信号。有关详细信息，请参阅PG-FP4 用户手册 (U15260E)。

表 22-5. 专用闪存编程器 (PG-FP4)的信号连接

PG-FP4			V850ES/HF2	连接的处理		
信号名称	I/O	引脚功能	引脚名称	UARTA0	CSIB0	CSIB0 + HS
FLMD0	输出	写启用/禁用	FLMD0	⊙	⊙	⊙
FLMD1	输出	写启用/禁用	FLMD1	⊙ <sup>注 1</sup>	⊙ <sup>注 1</sup>	⊙ <sup>注 1</sup>
VDD	—	VDD电压生成/电压监视器	VDD	⊙	⊙	⊙
GND	—	接地	VSS	⊙	⊙	⊙
CLK	输出	时钟输出到 V850ES/HF2	X1, X2	× <sup>注 2</sup>	× <sup>注 2</sup>	× <sup>注 2</sup>
RESET	输出	复位信号	RESET	⊙	⊙	⊙
SI/RxD	输入	接受信号	SOB0, TXDA0	⊙	⊙	⊙
SO/TxD	输出	发送信号	SIB0, RXDA0	⊙	⊙	⊙
SCK	输出	传输时钟	SCKB0	×	⊙	⊙
HS	输入	用于CSIB0 + HS 通信的交换信号	PCM0	×	×	⊙

- 注 1. 如图22-6所示连接这些引脚，或通过板上的下拉电阻连接至地。
2. 时钟不能通过闪存编程器的CLK引脚提供。在板上创建振荡器并提供时钟。

备注 ⊙: 必须连接。

×: 不一定连接。

表 22-6. V850ES/HF2 闪存配线适配器(FA-80GK-9EU-A)的配线

闪存编程器 (PG-FP4) 连接引脚			FA板引脚 的名称	使用的CSIB0 + HS		使用的CSIB0		使用的UARTA0	
信号 名称	I/O	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	接受信号	SI	P41/SOB0	20	P41/SOB0	20	P30/TXDA0	22
SO/TxD	输出	发送信号	SO	P40/SIB0	19	P40/SIB0	19	P31/RXDA0/INTP7	23
SCK	输出	传输时钟	SCK	P42/SCKB0	21	P42/SCKB0	21	不需要	—
CLK	输出	到 V850ES/HF2 的时钟	X1	不需要	—	不需要	—	不需要	—
			X2	不需要	—	不需要	—	不需要	—
/RESET	输出	复位信号	/RESET	RESET	14	RESET	14	RESET	14
FLMD0	输出	写电压	FLMD0	FLMD0	8	FLMD0	8	FLMD0	8
FLMD1	输出	写电压	FLMD1	PDL5/FLMD1	62	PDL5/FLMD1	62	PDL5/FLMD1	62
HS	输入	用于CSIB0 + HS 通信的交换信 号	RESERVE /HS	PCM0	49	不需要	—	不需要	—
VDD	—	VDD电压生成/ 电压监视器	VDD	V <sub>DD</sub>	9	V <sub>DD</sub>	9	V <sub>DD</sub>	9
				EVDD	31	EV <sub>DD</sub>	31	EV <sub>DD</sub>	31
				AV <sub>REF0</sub>	1	AV <sub>REF0</sub>	1	AV <sub>REF0</sub>	1
GND	—	接地	GND	V <sub>SS</sub>	11	V <sub>SS</sub>	11	V <sub>SS</sub>	11
				AV <sub>SS</sub>	2	AV <sub>SS</sub>	2	AV <sub>SS</sub>	2
				EV <sub>SS</sub>	30	EV <sub>SS</sub>	30	EV <sub>SS</sub>	30

- 注意事项
1. 确保将 REGC 引脚通过 4.7  $\mu$ F（推荐值）电容器连接到 GND。
  2. 时钟不能从闪存编程器的 CLK 引脚提供。在板上创建振荡器并提供时钟。

图 22-6. V850ES/HF2 闪存配线适配器(FA-80GK-9EU-A) 的配线示例  
(在 CSIB0 + HS 模式下) (1/2)

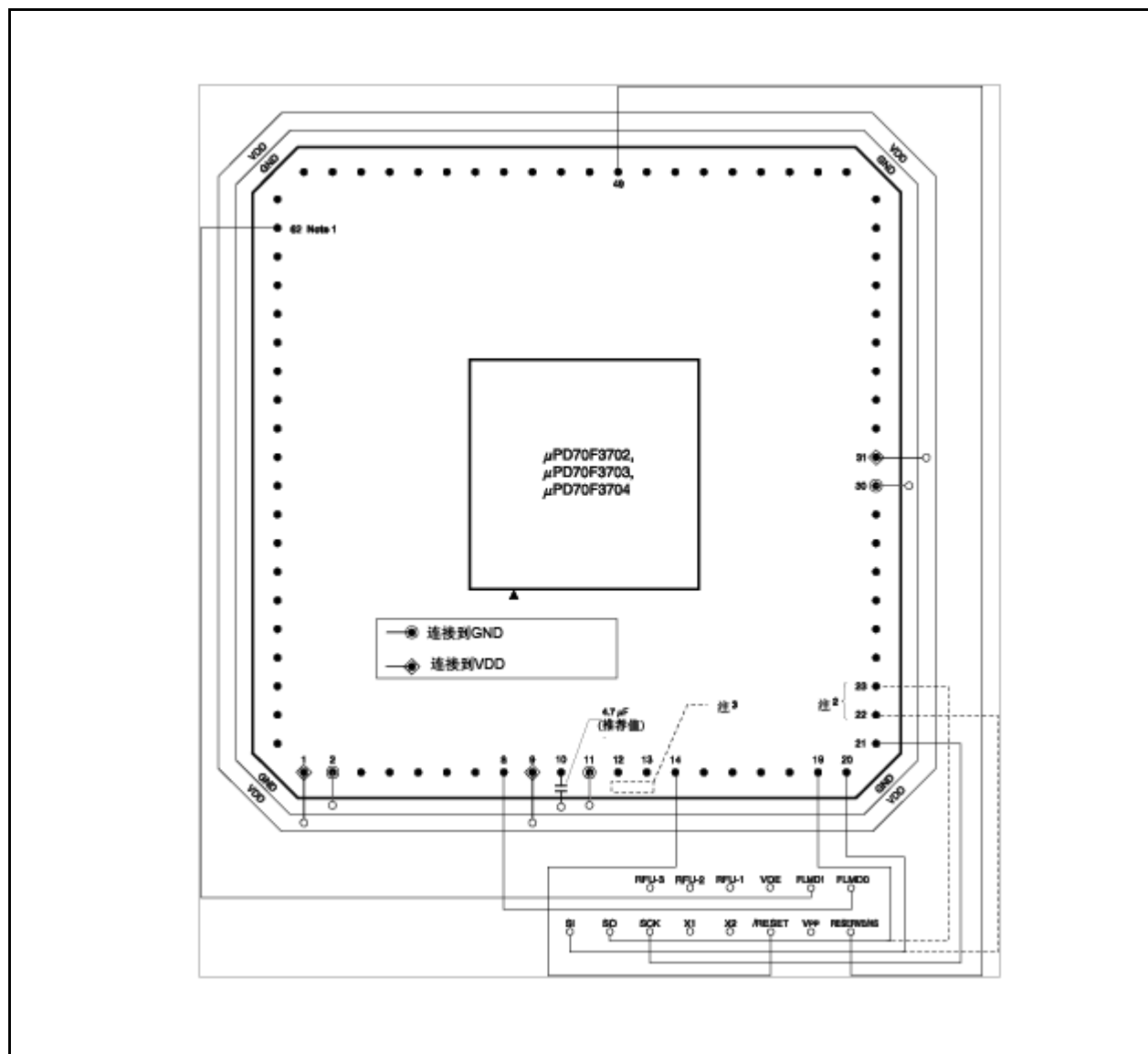
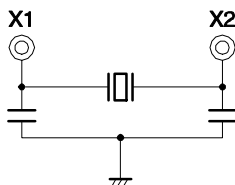


图 22-6. V850ES/HF2 闪存配线适配器 (FA-80GK-9EU-A)的配线示例  
(在 CSIB0 + HS 模式下) (2/2)

- 注
1. 如下所示连接FLMD1引脚，或通过下拉电阻将其连接至板上的GND端口。
  2. 使用UARTA0时使用的引脚。
  3. 在闪存配线适配器创建振荡器（如虚线所示）并提供时钟。此处为振荡器的示例。

示例



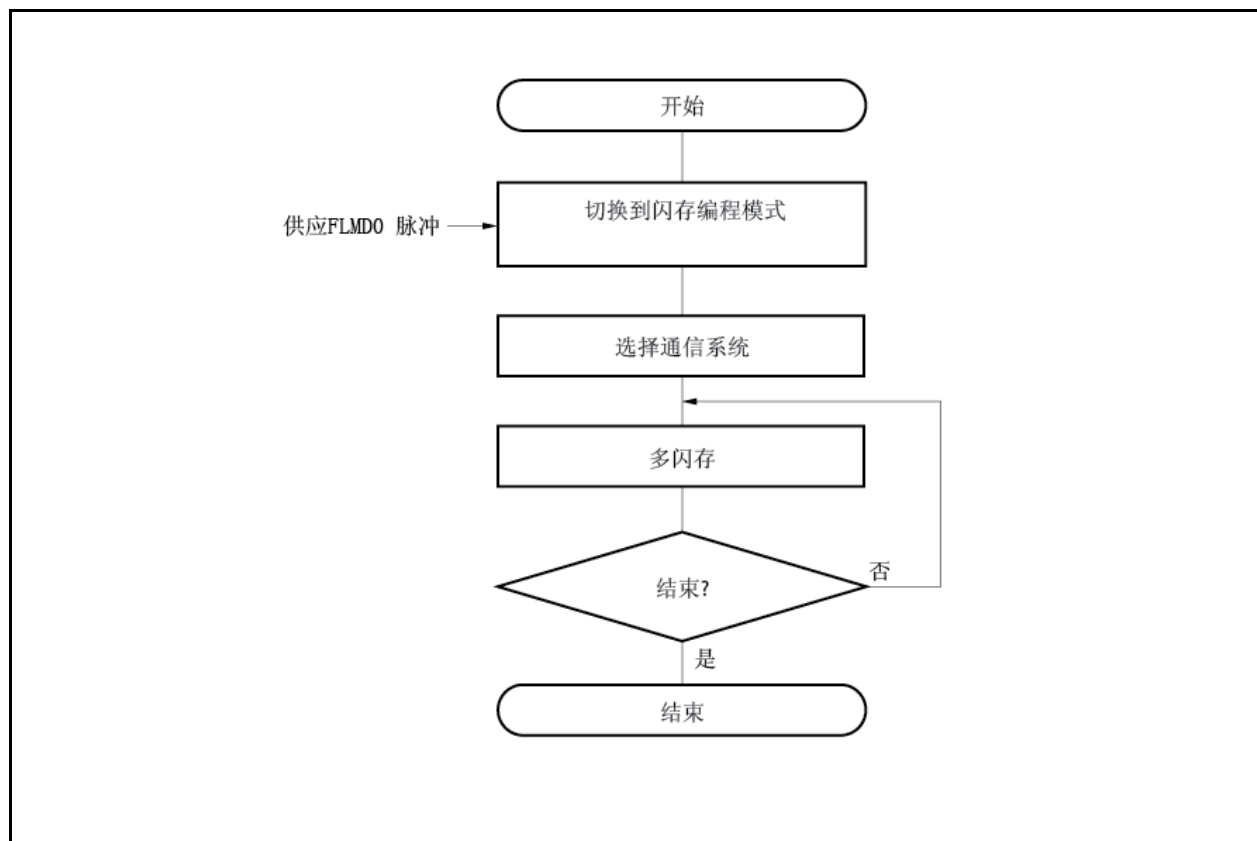
注意事项 严禁将高电平输入到  $\overline{\text{DRST}}$  引脚。

- 备注
1. 根据未使用的引脚的处理处理未显示的引脚。(参阅 2.3节引脚I/O 电路类型和未使用引脚的连接)。
  2. 该适配器用于80-引脚塑料TQFP 封装。

### 22.4.3 闪存控制

下图给出了操作闪存的步骤。

图 22-7. 操作闪存的步骤

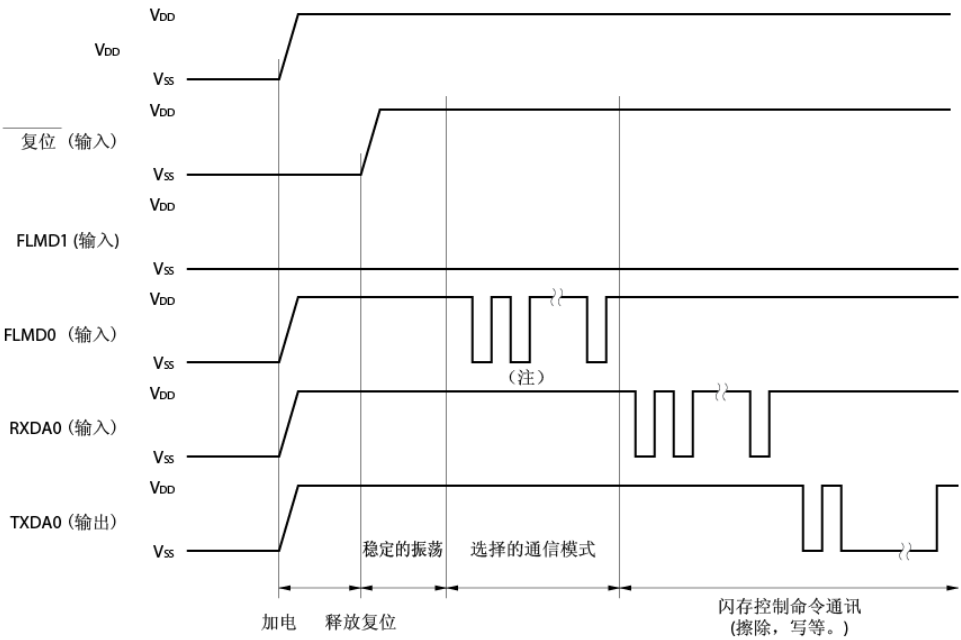


22.4.4 选择通信模式

在 V850ES/HF2 中，切换至闪存编程模式之后，通过将脉冲（最大 11 个脉冲）输入至 FLMD0 引脚选择通信模式。FLMD0 脉冲由专用闪存编程器产生。

下图给出了脉冲数与通信模式之间的关系。

图 22-8. 选择通信模式



注 时钟数如下取决于通信模式。

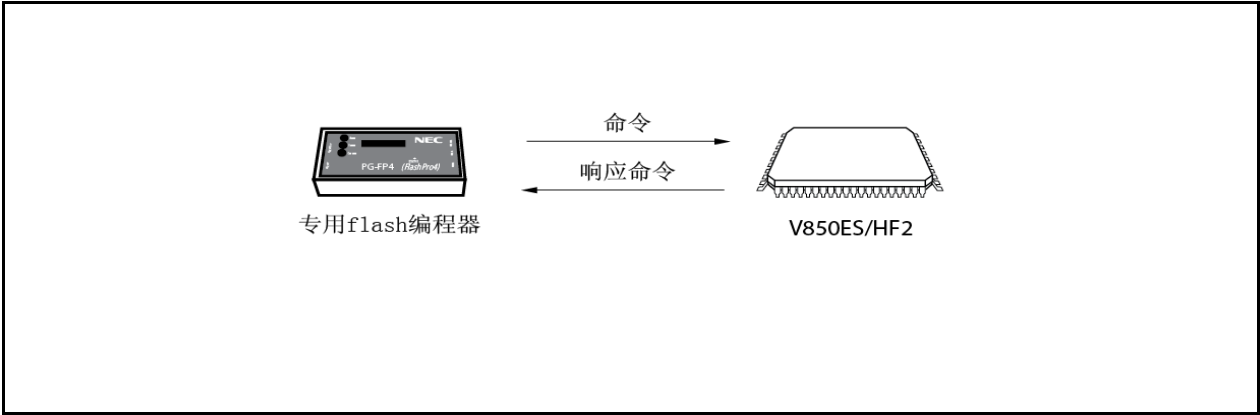
FLMD0 Pulse	通信模式	备注
0	UARTA0	通信速率: 9,600 bps (复位后), 首先从LSB开始
8	CSIB0	V850ES/HF2进行从属设备操作, 首先从MSB开始
11	CSIB0 HS	V850ES/HF2进行从属设备操作, 首先从MSB开始
其他	RFU	禁用设定

注意事项 选择 UARTA0 时，在接收FLMD0脉冲之后，接收时钟根据从专用闪存编程器发出的复位命令计算。

22.4.5 通信命令

V850ES/HF2通过命令与专用闪存编程器通信。从专用闪存编程器发送到V850ES/HF2的信号称作“命令”。这些从V850ES/HF2发送到专用闪存编程器的响应信号称作“响应命令”。

图 22-9. 通信命令



下表给出了 V850ES/HF2 中用于闪存控制的命令。所有这些命令从专用闪存编程器发出，且 V850ES/HF2 执行对应于这些命令的处理操作。

表 22-7. 闪存控制命令

分类	命令名称	支持			功能
		CSIB0	CSIB0 + HS	UARTA0	
空检查	块空检查命令	√	√	√	如果已经恰当擦除指定块中的存储器内容，则进行检查。
擦除	芯片擦除命令	√	√	√	擦除整个存储器。
	块擦除命令	√	√	√	擦除指定块的存储器内容。
写	编程命令	√	√	√	写入指定地址范围并执行内容校验检查。
校验	校验命令	√	√	√	将指定地址范围内的存储器内容与从闪存编程器传输的数据进行比较。
	校验和命令	√	√	√	读取指定地址范围中的校验和。
系统设置控制	硅签名命令	√	√	√	读取硅签名信息。
	安全设置命令	√	√	√	禁用芯片擦除命令，块擦除命令，编程命令，读命令。



### 22.4.6 引脚连接

执行主存储器写操作时，将连接器安装在连接至专用闪存编程器的目标系统。同样，结合板上的功能从正常运行模式切换至闪存编程模式。

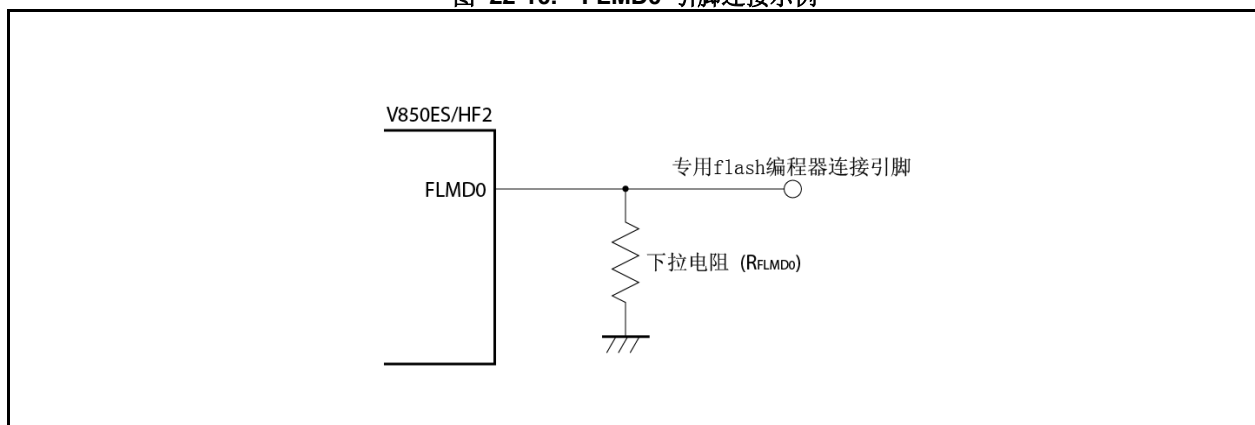
在闪存编程模式下，所有未用于闪存编程的引脚变成与立即复位后相同的状态。因此，在外部设备不承认复位后的当时状态时需要引脚操作。

#### (1) FLMD0 引脚

在正常运行模式下，将 VSS 电平的电压输入到 FLMD0 引脚。在闪存编程模式下，将 VDD 电平的写电压输入到 FLMD0 引脚。

因为在自编程模式下 FLMD0 引脚充当写保护引脚，所以在写入闪存之前，VSS 电平的电压必须通过端口控制等提供到 FLMD0 引脚。有关详细信息，请参阅 22.5.5 (1) FLMD0 引脚。

图 22-10. FLMD0 引脚连接示例



#### (2) FLMD1 引脚

将 0 V 输入到 FLMD0 引脚时，FLMD1 引脚不起作用。当将 VDD 提供到 FLMD0 引脚时，进入闪存编程模式，因此必须将 0 V 输入到 FLMD1 引脚。下图给出了 FLMD1 引脚连接的示例。

图 22-11. FLMD1 引脚连接示例

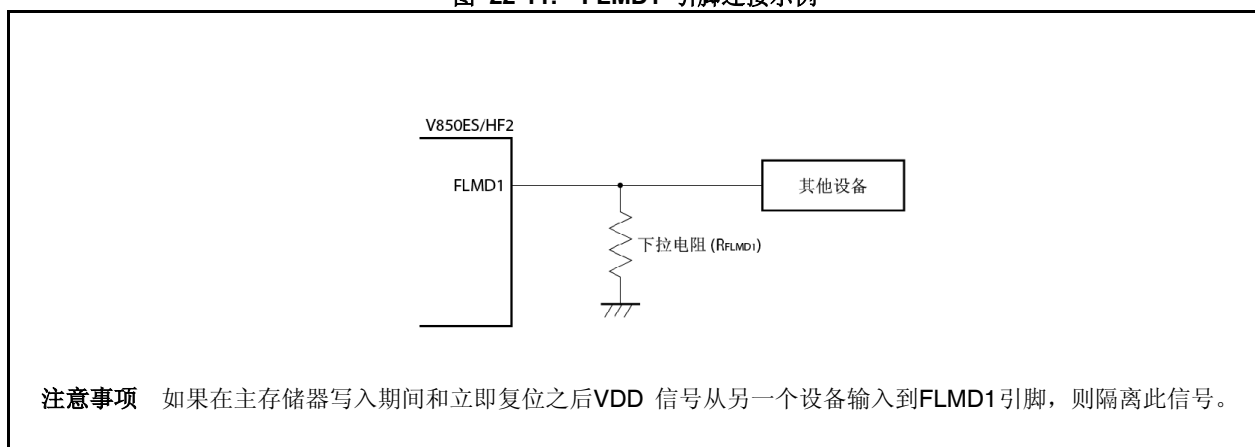


表 22-8. 释放复位功能时，FLMD0 与 FLMD1 引脚和运行模式之间的关系

FLMD0	FLMD1	运行模式
0	无关	正常运行模式
VDD	0	闪存编程模式
VDD	VDD	禁用设置

(3) 串口引脚

下表给出了每个串口使用的引脚。

表 22-9. 串口使用的引脚

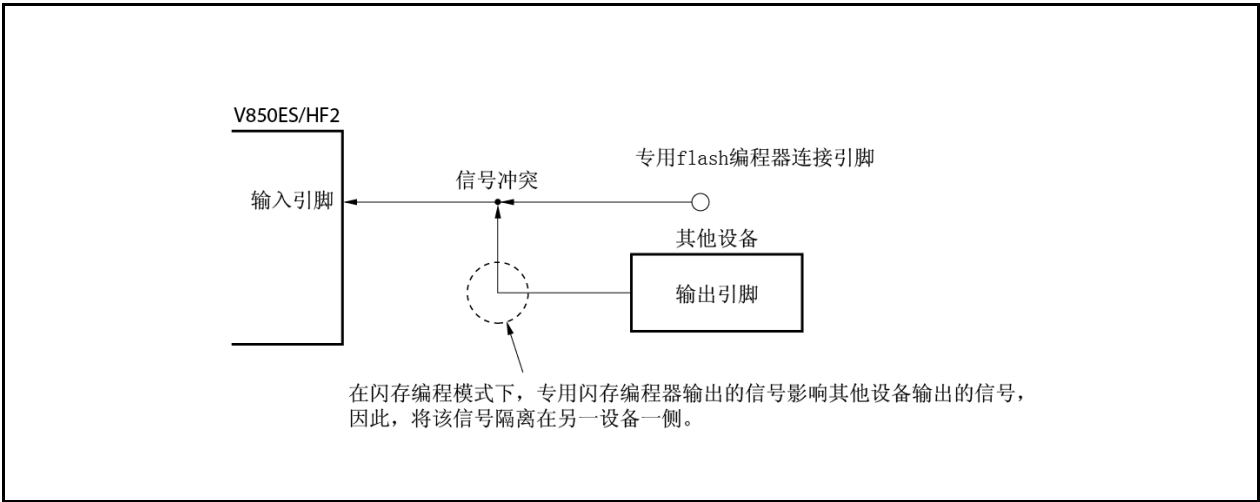
串口	使用的引脚
UARTA0	TXDA0, RXDA0
CSIB0	SOB0, SIB0, SCKB0
CSIB0 HS	SOB0, SIB0, SCKB0, PCM0

将专用闪存编程器连接到已经连接到其他板上设备的串口引脚时，小心以避免信号冲突和其他设备故障。

(a) 信号冲突

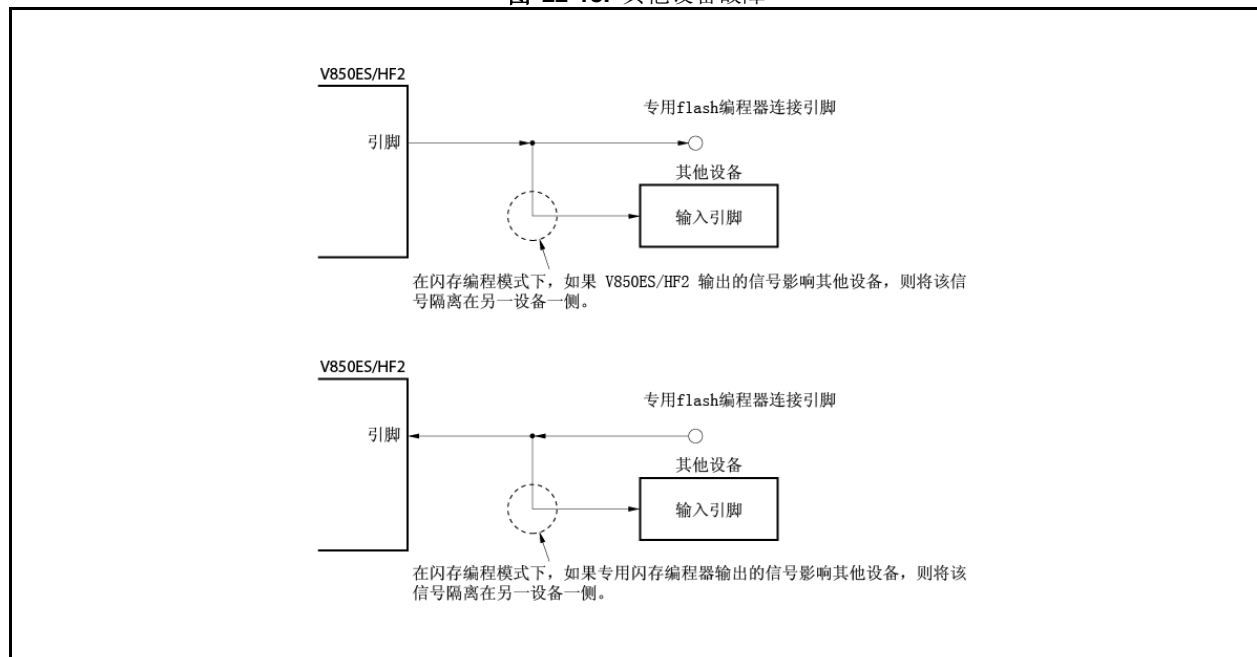
将专用闪存编程器（输出）连接到已经连接到其他设备(输出) 的串口引脚（输入)时，出现信号冲突。 要避免信号冲突，请隔离到其他设备的连接或将其他设备设定为输出高电阻状态。

图 22-12. 信号冲突（串口输入引脚）



**(b) 其他设备故障**

专用闪存编程器 (输出或输入) 连接到已经连接到其他设备(输出) 的串口引脚(输入或输出)时, 信号输出到其他设备, 导致设备故障。要避免这种情况, 请隔离到其他设备的连接。

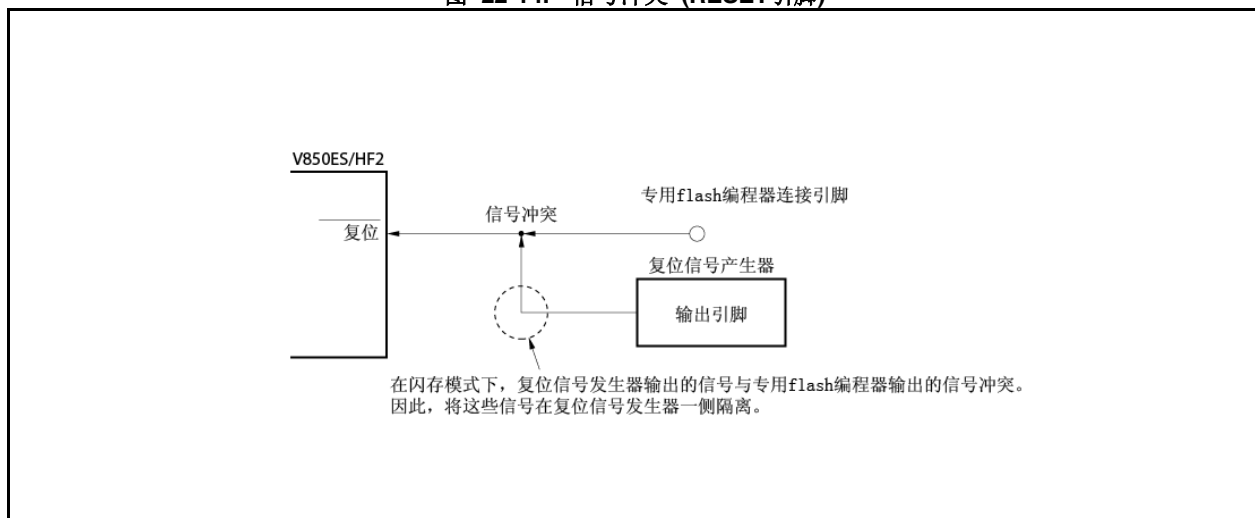
**图 22-13. 其他设备故障**

**(4) RESET 引脚**

专用闪存编程器的复位信号连接到已连接至板上复位信号生成器的 **RESET** 引脚复位时,出现信号冲突。要避免信号冲突,请隔离到复位信号生成器的连接。

在闪存编程模式下从用户系统输入复位信号时,不能正确执行编程操作。因此,不输入来自专用闪存编程器的复位信号之外的信号。

图 22-14. 信号冲突 (RESET引脚)

**(5) 端口引脚(包括 NMI)**

系统切换到闪存编程模式下时,所有未用于闪存编程的引脚处于与刚复位之后相同的状态。如果连接到每个端口的设备不能识别刚复位后的端口状态,则引脚需要进行适当的操作,例如通过电阻连接到 VDD 或通过电阻连接到 VSS。

**(6) 其他信号引脚**

以和正常运行模式下相同的状态连接 X1, X2, XT1, 和 XT2。

闪存编程期间,将低电平输入到DRST引脚或将引脚保持断开。严禁输入高电平。

**(7) 电源**

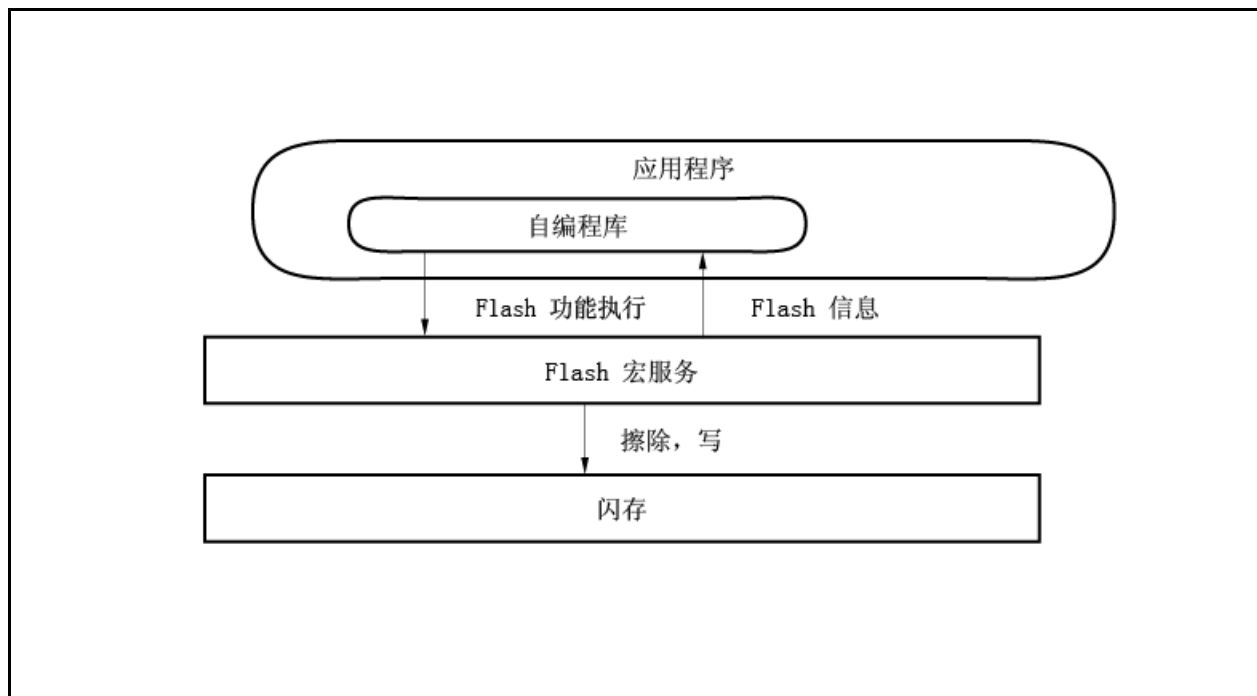
提供与正常运行模式下相同的功率(VDD, VSS, EVDD, EVSS, AVREF0, AVSS, REGC)。

## 22.5 通过自编程重写

### 22.5.1 概述

V850ES/HF2支持闪存宏服务，其允用户程序自己重写内部闪存。通过使用此接口和用于重写自编程库，其具有用户应用程序的闪存，可以通过提前传输到内部RAM或外部存储器的用户应用程序重写闪存。从而，可以升级用户程序且在该领域中可以重写恒定数据。

图 22-15. 自编程概念

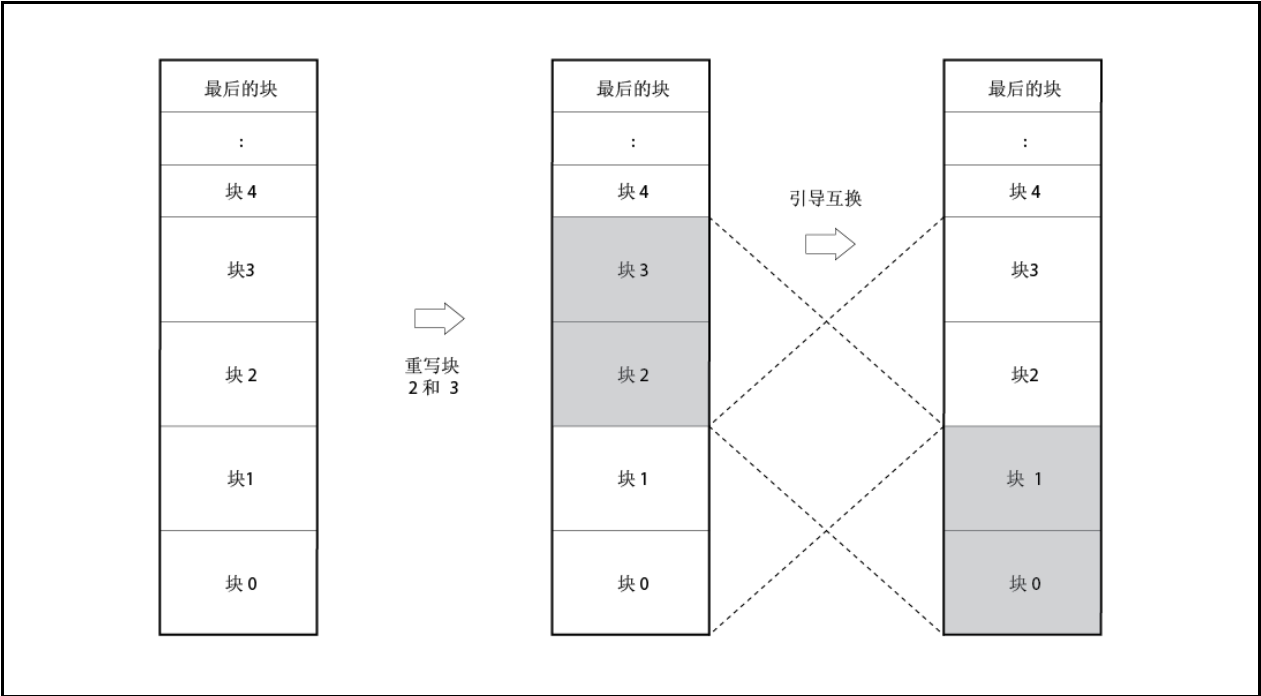


22.5.2 功能

(1) 安全自编程 (引导交换功能)

PD70F3703 和 70F3704支持引导交换功能，其可以将块0至1的物理存储器与块2至3的物理存储器交换。通过将启动程序提前写入块2至3并接着互换引脚物理存储器，因为正确的用户程序始终存处于块0至1中，所以即使在重写期间出现电源故障，也可以安全地重写整个区。

注意事项      安全自编程 (引导交换功能)  
图 22-16. 重写 整个存储区(引导交换)



(2) 中断支持

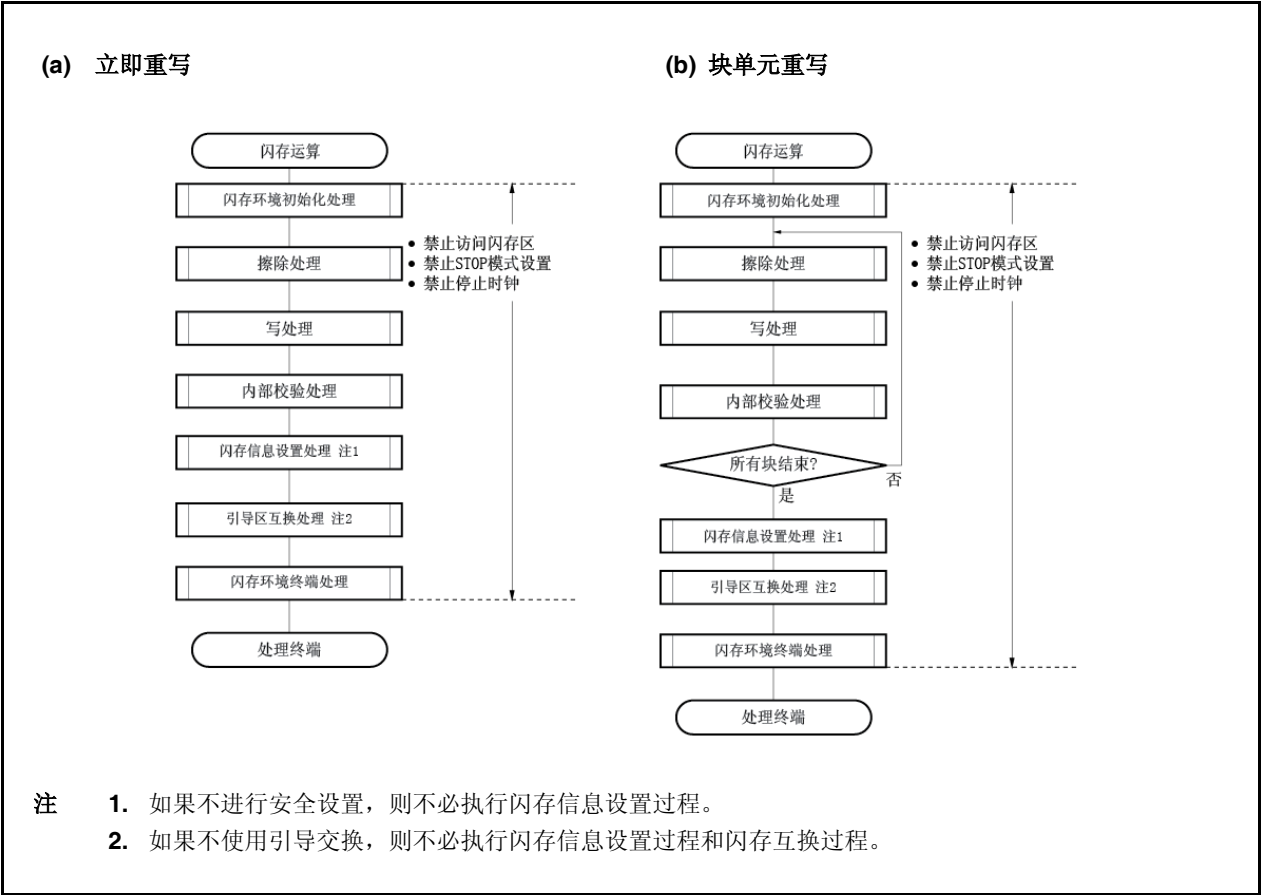
自编程期间不能从闪存获取指令。从而，即使发生中断，也不能使用写入闪存的用户句柄。因此，在 V850ES/ HF2 中，要在自编程期间使用中断，处理程序传输到内部 RAM 的专用地址<sup>※</sup>。分配跳转指令，其将处理程序传输到内部 RAM 的专用地址<sup>※</sup>处的用户中断服务。

注      NMI 中断:            内部 RAM的开始地址  
可屏蔽中断:            内部RAM + 4 地址的开始地址

22.5.3 标准自编程流程

通过闪存自编程重写闪存的整个处理过程如下所示。

图 22-17. 标准自编程程序流程



22.5.4 闪存功能

表 22-10. 闪存功能表

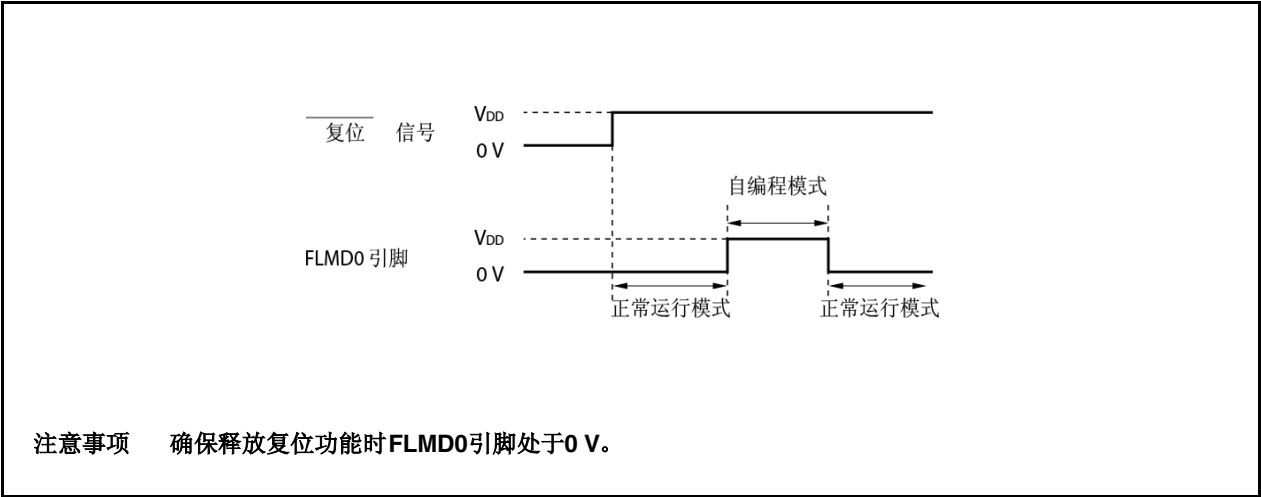
函数名称	概述	支持
FlashEnv	闪存控制宏初始化	√
FlashBlockErase	指定的块的擦除	√
FlashWordWrite	从指定地址写入	√
FlashBlockIVerify	指定的一个块的内部校验	√
FlashBlockBlankCheck	指定的一个块的空检查	√
FlashFLMDCheck	FLMD 引脚检查	√
FlashStatusCheck	立即指定运行之前的状态检查	√
FlashGetInfo	闪存信息读取	√
FlashSetInfo	闪存信息设置	√
FlashBootSwap	引导区互换	√
FlashWordRead	从指定地址读取数据	√
FlashSetUserHandler	用户中断句柄注册功能	√

22.5.5 引脚处理

(1) FLMD0 引脚

释放复位功能时，FLMD0 引脚用于设定运行模式且用于在自重写期间使闪存免于写入。从而释放复位功能时，需要将施加到 FLMD0 引脚的电压保持在 0 V 且执行正常操作。在自编程模式周期期间，重写存储器之前，还需要将 VDD 电平的电压通过端口控制施加到 FLMD0 引脚。完成自编程之后，FLMD0 引脚上的电压必须返回到 0 V。

图 22-18. 模式改变时序





### 22.5.6 使用的内部资源

下表列出了自编程使用的内部资源。这些内部资源还可以自由地用于自编程以外的用途。

表 22-11. 使用的内部资源

资源名称	说明
堆栈区(用户栈 + 300 字节)	由用户使用的堆栈的扩展由库使用 (可以用于内部 RAM 和外部 RAM)。
库代码 (约 2,500 字节)	库的程序实体(可以用于操作闪存块以外的任何位置)。
应用程序	作为用户应用程序执行。 调用闪存功能。
可屏蔽中断	可以用于用户应用程序执行状态或自编程状态。要在自编程状态下使用这个中断, 因为处理程序传输到内部 RAM 开始地址+4 地址的地址, 所以分配跳转指令, 该指令将处理程序提前传输到内部 RAM 开始地址+ 4 地址的地址处的用户中断服务。
NMI 中断	可以用于用户应用程序执行状态或自编程状态。要在自编程状态下使用这个中断, 因为处理程序传输到内部 RAM 开始地址的地址, 所以分配跳转指令, 该指令将处理程序提前传输到内部 RAM 开始地址的地址处的用户中断服务。

第二十三章 选项字节功能

作为 8 位数据，选项字节存储在内存（内 ROM 区）的地址 000007AH 中。  
当向 V850ES/HF2 写程序时，确保设置选项数据与作为默认值在地址 000007AH 中的下列选项相一致。  
在程序执行期间，不能重写该区域数据。

地址:000007AH

7	6	5	4	3	2	1	0
OPB7	OPB6	—	—	—	—	OPB1	OPB0

OPB7	OPB6	副时钟运行模式设置
0	0	晶体振荡器模式
1	1	RC 振荡器模式

OPB1	看门狗定时器2模式设置
0	运行可选时钟 (fx/f <sub>R</sub> ) 可选INTWDT2 模式/WDTRES 模式
1	固定到内振荡器时钟 (f <sub>R</sub> ) 固定到WDTRES 模式

OPB0	启用/禁用停止内部振荡器
0	启用停止
1	禁用停止

&lt;R&gt;

使用的 CA850 的程序样本如下所示。

**[程序样本]**

```
#-----  
# OPTION_BYTES  
#-----  
.section "OPTION_BYTES"  
.byte 0b00000001 -- 0x7a  
.byte 0b00000000 -- 0x7b  
.byte 0b00000000 -- 0x7c  
.byte 0b00000000 -- 0x7d  
.byte 0b00000000 -- 0x7e  
.byte 0b00000000 -- 0x7f
```

**注意事项** 确保在本节中写入 **6** 字节。如果不足 **6** 字节，连接器运行时，产生错误。

**错误信息: F4112:**不合法“OPTION\_BYTES”节范围。

**备注** 设置 0x00 地址为 007BH 至 007FH.

V850ES/HF2 片上调试功能可以通过下列两种方法实现。

- 使用 DCU (调试控制单元)  
在 V850ES/HF2 中，片上调试功能通过使用  $\overline{\text{DRST}}$ ，DCK， DMS， DDI 和 DDO 引脚作为调试接口引脚由片上 DCU 实现。
- 不使用 DCU  
片上调试功能通过使用用户资源替代 DCU，由 MINICUBE2 或类似实现

下表列出了两个片上调试功能的特性。

表 26-1. 片上调试功能特性

		使用 DCU 调试	未使用 DCU 调试
调试接口引脚		$\overline{\text{DRST}}$ , DCK, DMS, DDI, DDO	<ul style="list-style-type: none"><li>使用 UARTA0 时 RXD0, TXD0</li><li>使用 CSIB0 I 时 SIB0, SOB0, <math>\overline{\text{SCKB0}}</math>, HS (PCM0)</li></ul>
用户资源保证		不需要	需要
硬件断点功能		2 点	2 点
软件断点功能	内 ROM 区	4 点	4 点
	内 RAM 区	2000 点	2000 点
实时 RAM 监控功能 <sup>注 1</sup>		可用	可用
动态存储器修改功能(DMM) <sup>注 2</sup>		可用	可用
屏蔽功能		复位, NMI, INTWDT2	$\overline{\text{RESET}}$ 引脚
ROM 安全功能		10-字节 ID 码验证	10-字节 ID 码验证
使用硬件		NINICUBE <sup>®</sup> ,等	NINICUBE2, 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

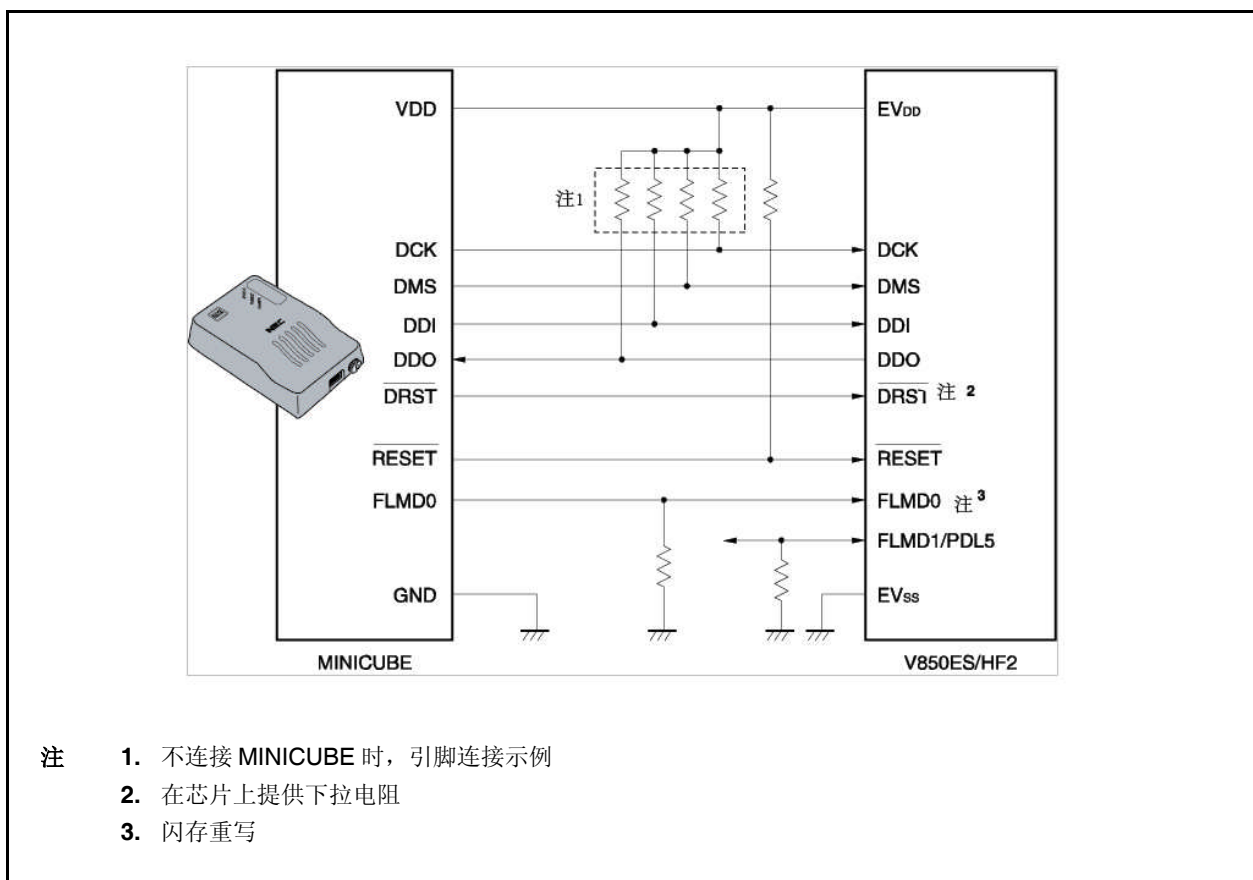
注     1. 在程序执行期间，读出存储器内容的功能。  
       2. 在程序执行期间，重写 RAM 内容的功能。

## 24.1 使用 DCU 调试

使用调试接口引脚(DRST, DCK, DMS, DDI, 和 DDO)连接到片上调试仿真器(MINICUBE)可以调试程序。

### 24.1.1 连接电路示例

图 24-1. 当通信接口使用调试接口引脚时电路连接示例



### 24.1.2 接口信号

接口信号如下所述。

(1) DRST

这是片上调试单元的复位输入信号。这是一个负逻辑信号，用于异步初始化调试控制单元。

MINICUBE 在启动集成调试器之后检测目标系统的  $V_{DD}$  时升高 DRST 信号并启动设备的片上调试单元。

**DRST** 信号处于高位时, 在 **CPU** 中还产生复位信号。

当通过启动集成调试器而开始调试时，始终会产生 CPU 复位。

**(2) DCK**

这是一个时钟输入信号。它从 MINICUBE 供应 20 MHz 的时钟。在片上调试单元，在 DCK 信号的上升沿对 DMS 和 DDI 信号取样，且在其下降沿输出数据 DDO。

**(3) DMS**

这是传输模式选择信号。调试单元中的传输状态随 DMS 信号的电平改变而改变。

**(4) DDI**

这是数据输入信号。在 DCK 的上升沿处在片上调试单元中对信号取样。

**(5) DDO**

这是数据输出信号。在 DCK 的下降沿处从片上调试单元输出信号。

**(6) EV<sub>DD</sub>**

这个信号用于检测目标系统的 VDD。如果检测不到来自目标系统的 VDD，则来自 MINICUBE ( $\overline{\text{DRST}}$ , DCK, DMS, DDI, FLMD0, 和  $\overline{\text{RESET}}$ ) 的信号输出处于高阻抗状态。

**(7) FLMD0**

闪存自身编程功能用于将数据经过集成调试器下载到闪存的功能。在闪存自编程期间，FLMD0 引脚必须保持高电平。另外，将下拉电阻连接到 FLMD0 引脚。

FLMD0 引脚可以用以下两种方法的任一种来控制。

**<1> 要从 MINICUBE 控制**

将来自 MINICUBE 的 FLMD0 信号连接到 FLMD0 引脚。

在正常模式下，MINICUBE 不驱动任何元件(高阻抗状态)。

在中断期间，执行集成调试器的下载功能时，MINICUBE 使 FLMD0 引脚上升到高电平。

**<2> 要从端口控制**

请将设备的任一个端口连接到 FLMD0 引脚。

可以使用与用于用户程序实现闪存自编程功能的一个端口相同的端口。

在集成调试器控制台上，进行设定以在执行下载功能之前将该端口引脚上升到高电平或在执行下载功能之后使该端口引脚电平下降。

有关详细信息，请参阅 ID850QB 3.10 版集成调试器操作用户手册(U17435E)。

**(8)  $\overline{\text{RESET}}$** 

这是系统复位输入引脚。如果通过用户编程设定 OCDM 寄存器的 OCDM0 位值使  $\overline{\text{DRST}}$  引脚无效，则不进行片上调试。因此，复位受 MINICUBE 影响，使用  $\overline{\text{RESET}}$  引脚，使  $\overline{\text{DRST}}$  使引脚无效(初始化)。

24.1.3 可屏蔽功能

可屏蔽 Reset, NMI, 和 INTWDT2 信号。  
调试器 (ID850QB) 和相应的 V850ES/HF2 的可屏蔽功能如下所示。

表 24-2. 可屏蔽功能

ID850QB 可屏蔽功能	相应的 V850ES/HF2 功能
NMI0	NMI 引脚输入
NMI2	不可屏蔽中断请求信号(INTWDT2)生成
STOP	×
HOLD	×
RESET	由 RESET 引脚输入, 低电压探测器, 时钟监视器, 上电清零电路 或看门狗时钟 (WDT2) 溢出产生复位信号
WAIT	×

24.1.4 寄存器

(1) 片上调试模式寄存器 (OCDM)

OCDM 寄存器用于选择正常操作模式或片上调试模式。此寄存器为专用寄存器, 且仅可以结合特定系列写入(请参阅 3.4.7 专用寄存器)。  
此寄存器还用于指定具有片上调试功能的引脚用作片上调试引脚还是用作常用端口/外围功能引脚。其还可用于断开 P05/INTP2/ $\overline{\text{DRST}}$  引脚的内部下拉电阻。  
OCDM 寄存器仅可以在  $\overline{\text{DRST}}$  在引脚输入低电平时写入。  
此寄存器可以读或写 8-位或 1-位单元。

复位后:01H<sup>Note</sup> R/W 地址:FFFF9FCH

	7	6	5	4	3	2	1	0
OCDM	0	0	0	0	0	0	0	OCDM0

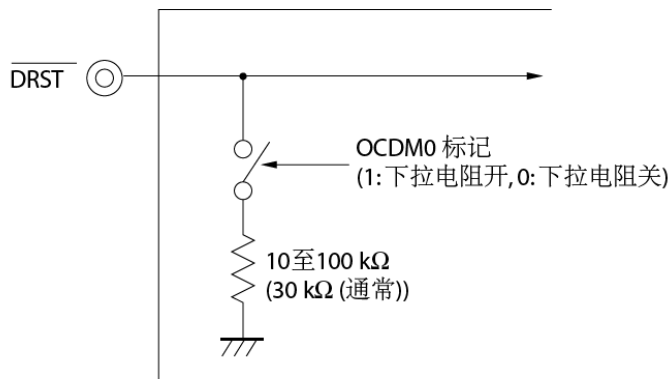
OCDM0	运行模式
0	选择正常运行模式(其中另外充当片上调试功能引脚的引脚用作端口/外围功能引脚)并断开 P05/INTP2/DRST 引脚的片上下拉电阻。
1	当 $\overline{\text{DRST}}$ 引脚处于低电平时:正常运行模式 (其中另外充当片上调试功能引脚的引脚用作端口/外围功能引脚) 当 $\overline{\text{DRST}}$ 引脚处于高电平时:片上调试模式 (其中另外充当片上调试功能引脚的引脚用作片上调试模式引脚)

**注**  $\overline{\text{RESET}}$  输入将此寄存器设定为 01H。然而,在通过 WDT2RES 信号、时钟监视器(CLM)或低压探测器 (LVI), 上电清零电路 (POC) 复位之后, 保留 OCDM0 寄存器的值。

**注意事项** 1. 当外部复位之后, DDI, DDO, DCK 和 DMS 引脚不用作片上调试引脚而用作端口引脚时, 必须采取以下措施的一种。

- 将 P05/INTP2/ $\overline{\text{DRST}}$  引脚输入低电平。
- 设定 OCDM0 位。在这种情况下, 请采取以下措施。
  - <1>将 OCDM0 位清零。
  - <2> 将 P05/INTP2/ $\overline{\text{DRST}}$  将引脚固定在低电平, 直到完成<1>。

2.  $\overline{\text{DRST}}$  引脚有一个片上下拉电阻。此电阻在 OCDM0 标志位清零时断开。





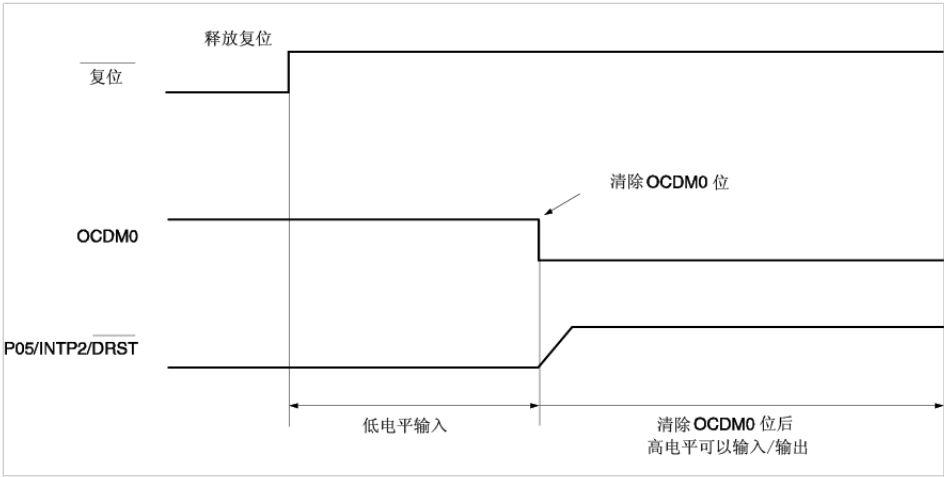
24.1.5 操作

在下表所示情况下，使片上调试功能无效。  
不使用此功能时，使  $\overline{\text{DRST}}$  使引脚保持低电平，直到  $\text{OCDM.OCDM0}$  标志位清零。

<div>OCDM0 Flag</div> <div><math>\overline{\text{DRST}}</math> 引脚</div>	0	1
L	无效	无效
H	无效	无效

备注     L: 低电平输入  
           H: 高电平输入

图 24-2. 不使用片上调试功能时的时序



### 24.1.6 注意事项

- (1) 在 RUN（程序执行）期间，如果(从目标系统或从内部复位资源)输入复位信号，此断点功能可能会出现故障。
- (2) 即使屏蔽功能屏蔽了该复位信号，如果从引脚输入一个复位信号，I/O 缓冲器（端口引脚）可能被复位。
- (3) 因为看门狗时钟 2 生成的目标复位或内复位使内闪存软件断点设置暂时无效。当发生硬件中断或被迫中断时，断点再次生效，但直至此时，也不会发生软件中断现象。
- (4) 中断期间引脚复位被屏蔽，且 CPU 和外围设备 I/O 不能复位。当执行用户程序时，一旦闪存被 DMM 重写或被 RAM 监测功能读取时，如果生成引脚复位或内复位，则 CPU 和外围设备 I/O 不能正确复位。
- (5) 当满足下列条件（a）和（b）且由于中断等仿真器（IECUBE<sup>®</sup>, MINICUBE）操作停止时，看门狗时钟 2 不能停止，且发生复位或非屏蔽中断。当发生复位时，调试器挂断。
  - (a) 主时钟或副时钟被看门狗时钟 2 用作源时钟。
  - (b) 内振荡器时钟停止(RCM.RSTOP 位 = 1)。

为了避免这种情况，执行下列操作之一。

- 当使用仿真器时，使用内振荡器时钟作为源时钟。
  - 当使用仿真器时，不停止内振荡器。
- (6) 当满足下列条件(a)和（b），且由于中断等仿真器(IECUBE, MINICUBE)操作停止时候，即使外围设备中断功能设置到“Break”，TMM 不停止。
    - (a) INTWT，内振荡器时钟(f<sub>R</sub>/8)或副时钟被选做 TMM 源时钟。
    - (b) 停止主时钟。

为了避免这种情况，执行下列操作之一。

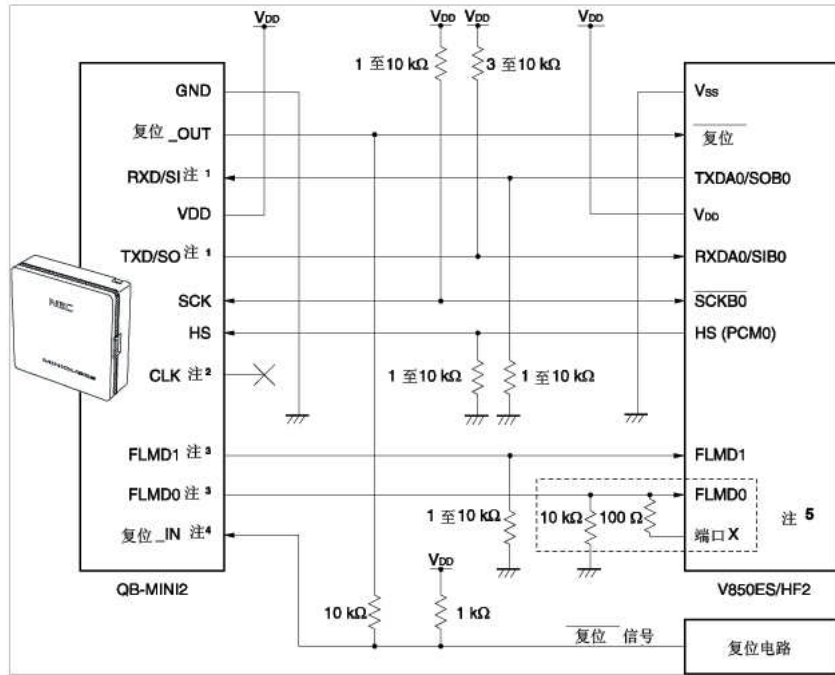
- 当使用仿真器时，主时钟(f<sub>xx</sub>, f<sub>xx</sub>/2, f<sub>xx</sub>/4, f<sub>xx</sub>/64, f<sub>xx</sub>/512)被用作源时钟。
  - 当使用仿真器时，禁用主时钟振荡器。
- (7) 在片上调试模式中，DDD 引脚被迫设为高电平输出。

## 24.2 不使用 DCU 调试

以下给出了如何使用 MINICUBE2 的 UARTA0 (RXDA0 和 TXDA0)引脚, 或 CSIB0 (SIB0, SOB0, SCKB0, 和 HS (PMC0))引脚作为调试接口, 不使用 DCU, 执行片上调试功能。

### 24.2.1 电路连接示例

图 24-3. 当 UARTA0/CSIB0 用作通信接口时, 电路连接示例



- 注 1. 将 V850ES/HF2 的 TXDA0/SOB0 (传输端) 连接到目标连接器的 RXD/SI (接收端), 并将目标连接器的 TXD/SO (传输端) 连接到 RXDA0/SIB0 V850ES/HF2 的 (接收端)。
2. 闪存编程期间, 该引脚用来从 MINICUBE2 提供时钟。有关详细情况, 请参阅第二十二章 闪存。
3. 已连接到 (FLMD0, FLMD1) 引脚的 V850ES/HF2-端引脚可用作复用功能引脚, 除了调试中断期间, 闪存重写。因为该引脚处于 Hi-Z 状态。
4. 该连接设计用于 RESET 信号从 N-ch 开漏缓冲器输出的功能(输出电阻: 100 Ω 或更少)。
5. 虚线标出的电路为闪存自编程, 其功能通过端口控制 FLMD0 实现。使用端口输入或输出高电平。当闪存自编程不执行时, FLMD0 引脚下拉电阻位于 1 至 10 kΩ 之间。

备注 UARTA0 或 CSIB0 用作通信接口时, 关于已使用引脚请参阅表 24-3。

表 24-3. V850ES/HF2 和 MINICUBE2 间配线

MINICUBE2 (QB-MINI2)引脚配置			与 CSIB0-HS		与 UARTA0	
信号名称	I/O	引脚功能	引脚名称	引脚编号	引脚名称	引脚编号
SI/RxD	输入	引脚从 V850ES/HF2 接收命令和数据。	P41/SOB0	20	P30/TXD0	22
SO/TxD	输出	引脚传输命令和数据到 V850ES/HF2	P40/SIB0	19	P31/RXD0	23
SCK	输出	3-线 串口通信时钟输出引脚	P42/SCKB0	21	Not needed	–
CLK <sup>Note</sup>	输出	V850ES/HF2 时钟输出引脚	不需要 <sup>註</sup>	–	不需要 <sup>註</sup>	–
			不需要 <sup>註</sup>	–	不需要 <sup>註</sup>	–
RESET_OUT	输出	V850ES/HF2 复位输出引脚	RESET	14	RESET	14
FLMD0	输出	输出引脚用来设 V850ES/HF2 到调试模式或编程模式	FLMD0	8	FLMD0	8
FLMD1	输出	输出引脚用来设编程模式	PDL5/FLMD1	62	PDL5/FLMD1	62
HS	输入	用于 CSIO + HS 通信的交换信号	PCM0	49	不需要	–
GND	–	接地	V <sub>ss</sub>	11	V <sub>ss</sub>	11
			AV <sub>ss</sub>	2	AV <sub>ss</sub>	2
			EV <sub>ss</sub>	30	EV <sub>ss</sub>	30
RESET_IN	输入	目标系统的复位输入引脚				

注 作为用于 MINICUBE2 的 flash 编程器的时钟输出。有关详细情况，请参阅第 22 章闪存。

24.2.2 可屏蔽功能

仅复位信号可以被屏蔽。  
调试器(ID850QB)和相应的 V850ES/HF2 功能的屏蔽功能如下所示。

表 24-4. 可屏蔽功能

ID850QB 可屏蔽功能	相应的 V850ES/HF2 功能
NMI0	–
NMI1	–
NMI2	–
STOP	–
HOLD	–
RESET	RESET 引脚输入生成复位信号
WAIT	–

### 24.2.3 用户资源的安全

用户必须准备以下条件来进行 MINICUBE2 和目标设备的通信及履行各个调试功能。该项需要在用户程序中或使用编译器选项时进行设置。

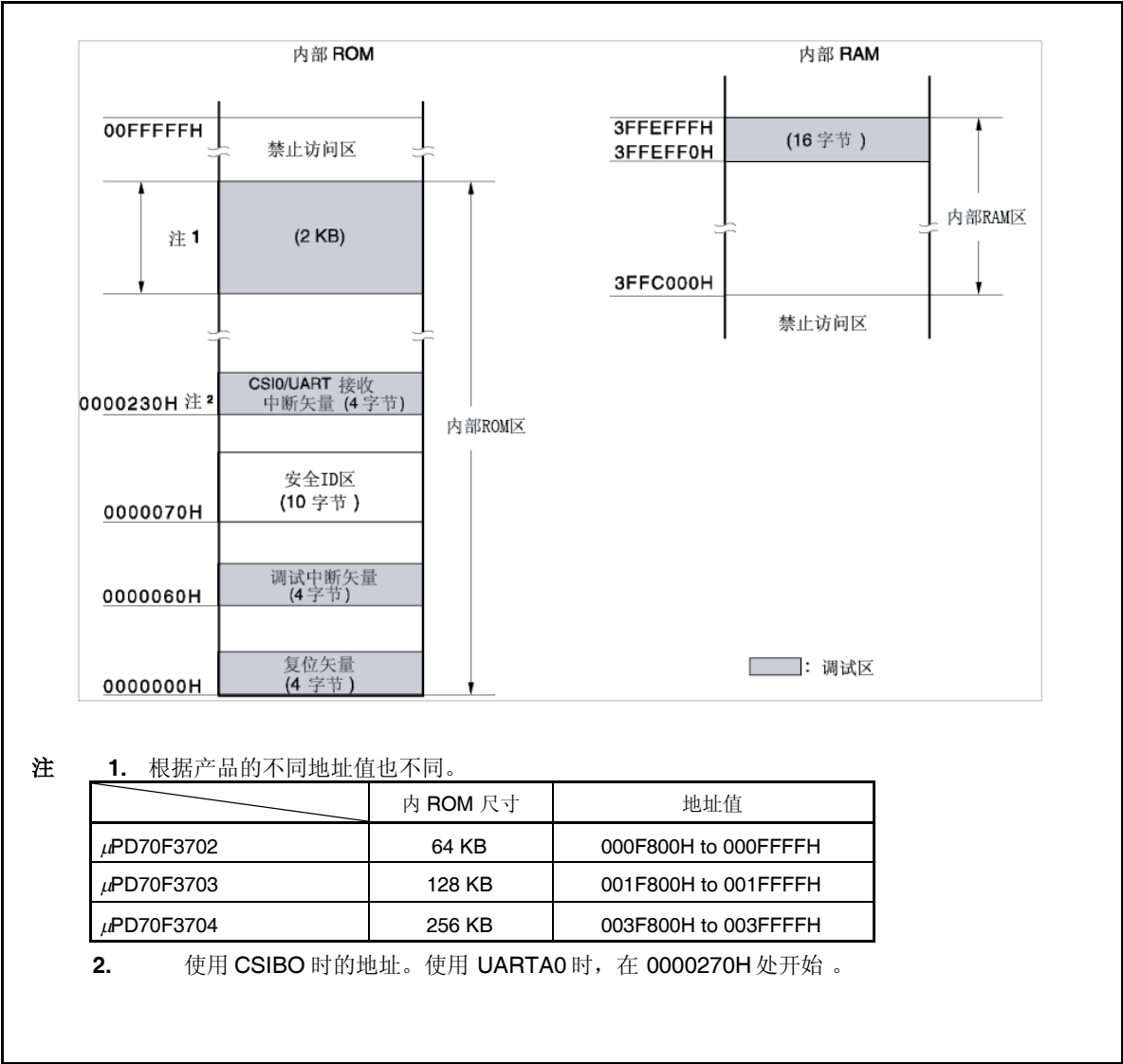
#### (1) 存储空间的安全

图 24-4 阴影部分为保留放置调试监控器程序的区域，因此这部分空间不能分配用户程序和数据。这部分空间由用户程序保证安全免得被使用。

#### (2) 安全 ID 设置

在图 24-4 中，ID 码必须被镶嵌入 0000070H 和 0000079H 之间的区域中，以阻止记忆被未被授权人读取。有关详细情况，请参阅 **24.3 ROM 安全功能**。

表 24-4. 当分配调试监视器程序时存储空间



**(3) 复位矢量**

复位矢量包括调试监视程序的转跳指令。

[如何保证区域安全]

不需要有意的去保证该区的安全。但下载程序时，调试器根据下列情况重写复位矢量。如果重写形态与下列情况不匹配，调试器产生错误（当使用 ID850Q 时 F0C34）。

**(a) 当从地址 0 开始的连续放两个 nop 指令时。**

重写前		重写后
0x0 nop	→	在 0x0 转跳至调试监视程序
0x2 nop		0x4 xxxx
0x4 xxxx		

**(b) 当从地址 0 开始连续放两个 0xFFFF 时(已擦除过设备)**

重写前		重写后
0x0 0xFFFF	→	在 0x0 转跳至调试监视程序
0x2 0xFFFF		0x4 xxxx
0x4 xxxx		

**(c) 在地址 0 处放置 jr 指令 (使用 CA850 时)**

重写前		重写后
0x0 jr disp22	→	在 0x0 转跳至调试监视程序
		0x4 jr disp22 - 4

**(d) 从地址 0 处连续放置 mov32 和 jmp (当使用 IAR 编译器 ICCV850 时)**

重写前		重写后
0x0 mov imm32,reg1	→	在 0x0 处转跳至调试监视程序
0x6 jmp [reg1]		0x4 mov imm32,reg1
		0xa jmp [reg1]

**(e) 在地址 0 处放置调试监视程序转跳指令**

重写前		重写后
在 0x0 处转跳至调试监视程序	→	不改变

#### (4) 调试监视程序的区域安全

图 24-4 中阴影部分为分配调试监视程序的区域。监视程序为调试通信接口执行初始化处理，并对 CPU 进行运行或中断处理。内 ROM 区必须用 0xFF 填充。该区不得由用户程序重写。

##### [如何确保区域安全]

如果用户程序不使用该区域，不必必须保证区域安全。

为了避免在调试器启动期间可能发生的问题，推荐使用编译器提前保证该区域安全。

下面给出了保证该区域安全的示例，使用 NEC Electronics CA850 编译器。添加汇编源文件和连接指令代码，如下所示。

- 汇编源 (作为汇编源文件添加下列代码)

```
-- 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 "INTCB0R"
.space 4, 0xff

-- Secures 16-byte space for monitor RAM section
.section "MonitorRAM", bss
.lcomm monitorramsym, 16, 4 -- defines symbol monitorramsym
```

- 连接指令 (将下列代码添加到连接指令文件)

当内 ROM 为 256KB (末端地址为 003FFFFH) 且当内 RAM 为 24KB (末端地址为 3FFEFFFH) 时，示例如下。

```
MROMSEG      : !LOAD ?R V0x03f800{
               MonitorROM    = $PROGBITS    ?A MonitorROM;
};

MRAMSEG       : !LOAD ?RW V0x03ffeff0{
               MonitorRAM    = $NOBITS      ?AW MonitorRAM;
};
```



(5) 通信串口安全

UARTA0 和 CSIB0 用于 MINICUBE2 和目标设备之间的通信。调试监视程序执行有关串口模式的设置，但如果设置被用户程序改变，可能会发生通信错误。  
为了阻止发生该问题，必须在用户程序中保证通信串口的安全。

[如何保证通信串口安全]

- 片上调试模式寄存器 (OCDM)  
对于片上调试功能使用 UARTA0 或 CSIB0F，设置 OCDM 寄存器功能为正常模式，确保如下所示。
  - 输入低电平到 P05/INTP2/DRST 引脚。
  - 设置 OCDM0 位如下：
    - <1> 将 OCDM0 位清除到 0。
    - <2> 将 P05/INTP2/DRST 引脚输入固定到低电平，直到<1>处理完成。
- 串口寄存器  
禁止在用户程序中设置与 CSIB0 或 UARTA0 有关的寄存器。
- 中断屏蔽寄存器  
使用 CSIB0 时，不屏蔽传输末端中断(INTCB0R)。使用 UARTA0 时，不屏蔽接收末端中断 (INTUA0R)。

(a) 当使用 CSIB0 时

76543210

CB0RIC

×

0

×

×

×

×

×

×

(b) 当使用 UARTA0 时

76543210

UA0RIC

×

0

×

×

×

×

×

×

备注

×

不必考虑

- 使用 **UARTA0** 时，端口寄存器  
使用 **UARTA0** 时，通过调试监视程序设置端口寄存器使 **TXDA0** 和 **RXDA0** 引脚有效。在调试期间禁止使用用户程序改变下列寄存器设置。（同值可以被覆盖。）

	7	6	5	4	3	2	1	0
PMC3L	x	x	x	x	x	x	1	1

备注      x: 不必考虑

- 端口寄存器使用 **CSIB0** 时  
当使用 **CSIB0** 时，调试监视程序设置端口寄存器使 **SIB0**，**SOB0**，**SCKB0** 和 **HS (PMC0)**引脚有效。在调试期间禁止使用用户程序更改下列寄存器设置。（同值可以被覆盖。）

(a) <b>SIB0, SOB0 和 SCKB0</b> 设置								
	7	6	5	4	3	2	1	0
PMC4	x	x	x	x	x	1	1	1

(b) <b>HS (PMC0 引脚)</b> 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。  
相应的 **HS** 引脚端口值根据调试器状态由监视程序更改。为了在 **8** 位单元执行端口寄存器设置，用户程序通常可以使用读取—修改—写入。如果写入前发生调试中断，意外操作可能被执行。

备注      x: 不必考虑

#### 24.2.4 注意事项

##### (1) 作调试用的设备的处理

禁止安装曾在批量生产的产品上用作调试的设备，因为在调试期间，闪存被重写且闪存重写次数不能被保证。此外，禁止将调试监视程序嵌入批量生产的产品中。

##### (2) 当不能执行断点时

如果满足下列条件之一，则不能执行被迫断点。

- 禁用中断(DI)
- 用来在 MINICUBE2 和目标设备间进行通信的串行接口中断被屏蔽。
- 当禁止可屏蔽中断备用释放时，进入待机模式。
- MINICUBE2 和目标设备间的通信模式为 UARTA0，主时钟已停止。

##### (3) 当伪实时 RAM 监视器功能 (RRM) 和 DMM 功能无法运行时

如果满足下列条件之一，伪 RRM 功能和 DMM 功能无法运行

- 中断禁止 (DI)
- 用于 MINICUBE2 和目标设备间进行通信的串行接口中断被屏蔽。
- 当禁止可屏蔽中断备用释放时，进入待机模式。
- MINICUBE2 和目标设备间的通信模式为 UARTA0，主时钟已停止。
- MINICUBE2 和目标设备间的通信模式为 UARTA0，用来通信的时钟与调试器指定不同。

##### (4) 启用备用释放伪 RRM 和 DMM 功能。

如果符合下列条件之一，伪 RRM 功能和 DMM 功能释放备用模式。

- MINICUBE2 和目标设备的通信模式为 CSIB0。
- MINICUBE2 和目标设备的通信模式为 UARTA0，提供主时钟。

##### (5) 使用 DMM 功能，需要一个明确的顺序写入外围 I/O 寄存器。

需要一个明确的顺序的外围 I/O 寄存器不能用 DMM 功能写入。

##### (6) 使调试器启动变慢的设备。

当调试器启动时，调试监视程序的芯片擦除和写入运行，但该操作需要花费大约十几秒的时间。

##### (7) 调试监视程序的写入

当 CPU 操作时钟设置用调试器更改时，调试器重写到监视程序。所需要的时间和 (6) 提到的一样。对于集成调试器 ID850QB，应用于对配置对话框的设置进行更改时。

##### (8) 闪存自编程序

如果分配到调试监视程序的空间被闪存自编程序重写，则调试器运行不再正常。

## 24.3 ROM 安全功能

### 24.3.1 安全 ID

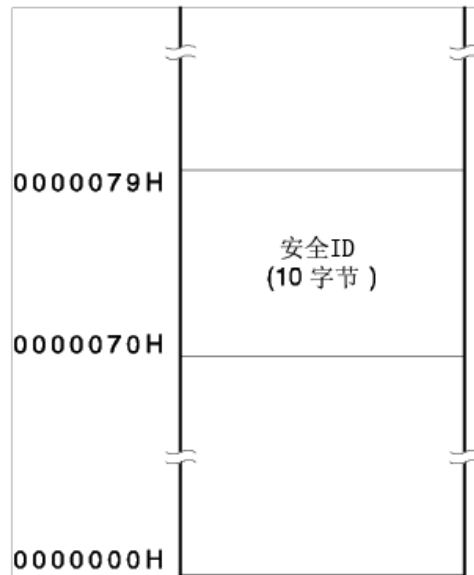
V850ES/HF2 的闪存版本执行验证使用 10 字节 ID 码被片上调试仿真器阻止在片上调试期间未授权人读出闪存内容。

在 0000070H 和 0000079H 之间的 10 一字节片上闪存区设置 ID 码，以使调试器执行 ID 验证。

如果 ID 匹配，安全释放，启用读取闪存并使用片上调试仿真器。

- 设置 10-字节 ID 码到 0000070H 和 0000079H 之间。
- 0000079H 位 7 为片上调试仿真器的启用标记(0: 禁用; 1: 启用)。
- 当启动片上调试仿真器时，调试器需要输入 ID。当输入 ID 代码到调试器中，且 ID 代码 0000070H 至 0000079H 设置匹配，调试器启动。
- 如果片上仿真启用标记为 0，即使 ID 代码匹配，也不能进行调试。

图 24-5. 安全 ID 区



**注意事项** 在擦除闪存之后，1 被写入到整个区域。

### 24.3.2 设置

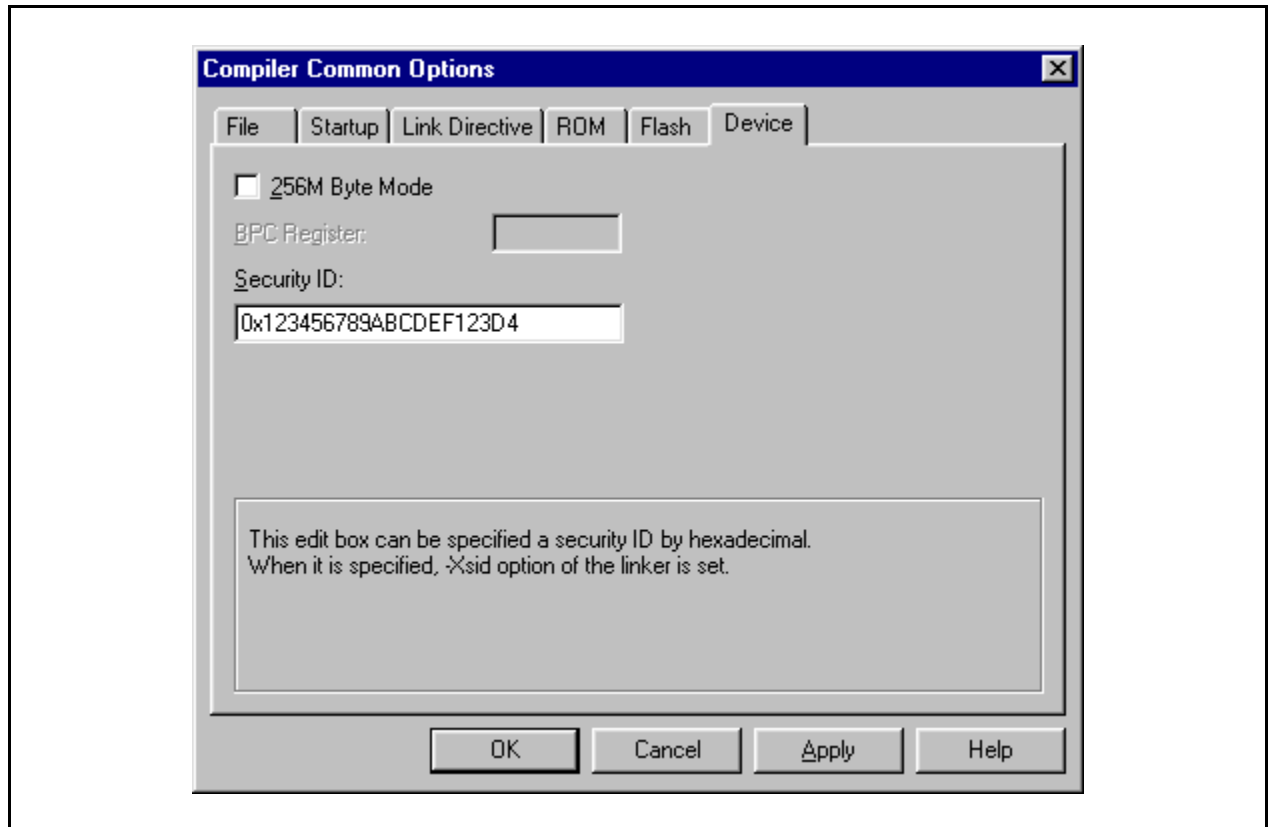
下表列出了如何设置 ID 代码，如表 24-5 所示。

当 ID 代码设置如表 24-5 所示，在 ID850QB 的配置对话框中 ID 代码输入为“123456789ABCDEF123D4”（ID 代码大小写不敏感）。

表 24-5. ID 代码

地址	值
0x70	0x12
0x71	0x34
0x72	0x56
0x73	0x78
0x74	0x9A
0x75	0xBC
0x76	0xDE
0x77	0XF1
0x78	0x23
0x79	0xD4

ID 代码可以指定设备文件，该文件用于支持 CA850 3.10 或更高版本和使用 PM+ 编译器命令选项设置的安全 ID。



[程序示例(当使用 CA850 3.10 或更高版本)时]

```
#-----  
#      SECURITYID  
#-----  
      .section    "SECURITY_ID"    --Interrupt handler address 0x70  
      .word       0x78563412      --0-3 byte code  
      .word       0xF1DEBC9A      --4-7 byte code  
      .hword      0xD423           --8-9 byte code
```

**备注** 将上述程序示例添加到启动文件。

## 第二十五章 电气特性

### 25.1 最大绝对级别

#### 最大绝对级别( $T_A = 25^\circ\text{C}$ ) (1/2)

参数	符号	条件	级别	单位
电源电压	$V_{DD}$	$V_{DD} = EV_{DD}$	-0.5 至+6.5	V
	$EV_{DD}$	$V_{DD} = EV_{DD}$	-0.5 至+6.5	V
	$AV_{REF0}$		-0.5 至+6.5	V
	$V_{SS}$	$V_{SS} = EV_{SS} = AV_{SS}$	-0.5 至+0.5	V
	$AV_{SS}$	$V_{SS} = EV_{SS} = AV_{SS}$	-0.5 至+0.5	V
	$EV_{SS}$	$V_{SS} = EV_{SS} = AV_{SS}$	-0.5 至+0.5	V
输入电压	$V_{I1}$	P00 至 P06, P30 至 P35, P38, P39, P40 至 P42, P50 至 P55, P90, P91, P96 至 P99, P913 至 P915, PCM0 至 PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 至 PDL11, RESET, FLMD0	-0.5 至 $EV_{DD} + 0.5$ <sup>※</sup>	V
	$V_{I3}$	X1, X2, XT1, XT2	-0.5 至 $V_{RO} + 0.5$ <sup>※</sup>	V
模拟输入电压	$V_{IAN}$	P70 至 P711	-0.5 至 $AV_{REF0} + 0.5$ <sup>※</sup>	V

注 确保不超过每个电源电压的最大绝对级别（最大值）。

- 注意事项 1. 严禁 IC 设备的输出(或 I/O)引脚彼此直接连接，或连接到 VDD, VCC 和 GND。
2. 如果任何参数即使短时超过最大绝对级别，则可能导致产品质量问题。也就是说，最大绝对级别为产品处于遭受物理性损坏的边缘时的级别值，且因此产品必须在确保不超过最大绝对级别的条件下使用。显示 DC 特性和 AC 特性的级别和条件表示正常运行期间的质量保证范围。
3. 当将外部电路直接连接到逐渐升高的高阻抗状态下的引脚时，设计定时器已避免外部电路的输出冲突。

备注 除非另有规定，复用功能引脚的特性与端口引脚的特性相同。

最大绝对级别 ( $T_A = 25^{\circ}\text{C}$ ) (2/2)

参数	符号	条件		级别	单位
输出电流, 低	I <sub>OL</sub>	P00 至 P06, P30 至 P35, P38, P39, P40 至 P42, P50 至 P55, P90, P91, P96 至 P99, P913 至 P915, PCM0 至 PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 至 PDL11	每个引脚	4	mA
			所有引脚总数	50	mA
		P70 至 P711	每个引脚	4	mA
			所有引脚总数	20	mA
输出电流, 高	I <sub>OH</sub>	P00 至 P06, P30 至 P35, P38, P39, P40 至 P42, P50 至 P55, P90, P91, P96 至 P99, P913 至 P915, PCM0 至 PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 至 PDL11	每个引脚	−4	mA
			所有引脚总数	−50	mA
		P70 至 P711	每个引脚	−4	mA
			所有引脚总数	−20	mA
运行环境温度	T <sub>A</sub>	正常操作模式		−40 至 +85	°C
		闪存编程模式			
存储温度	T <sub>stg</sub>			−40 至+125	°C

- 注意事项**
1. 严禁 IC 产品的输出(或 I/O)引脚彼此直接连接, 或连接到  $V_{DD}$ ,  $V_{CC}$  和  $GND$ 。
  2. 如果任何参数即使短时超过最大绝对级别, 则可能导致产品质量问题。也就是说, 最大绝对级别为产品处于遭受实际损坏的边缘时的级别值, 且因此产品必须在确保不超过最大绝对级别的条件下使用。  
显示 DC 特性和 AC 特性的级别和条件表示正常运行期间的质量保证范围。
  3. 当将外部电路直接连接到逐渐升高的高阻抗状态下的引脚时, 设计定时器已避免外部电路的输出冲突。

**备注** 除非另有规定, 复用功能引脚的特性与端口引脚的特性相同。



## 25.2 电容

(TA = 25°C, VDD = EVDD = AVREF0 = VSS = EVSS = AVSS = 0 V)

参数	符号	条件	最小	通常	最大	单位
I/O 电容	C <sub>IO</sub>	f <sub>x</sub> = 1 MHz, 未测量的引脚返回到 0 V.			10	pF

## 25.3 运行条件

(TA = 40 至 +85°C, VDD = EVDD = 3.5 V 至 5.5 V, 4.0 V ≤ AVREF0 ≤ 5.5 V, VSS = EVSS = AVSS = 0 V)

参数	符号	条件	最小	通常	最大	单位
内部系统时钟频率	f <sub>CLK</sub>	REGC = 4.7 μF, 主系统时钟运行	4		20	MHz
		REGC = 4.7 μF, 副系统时钟运行 (晶体振荡器)	32		35	kHz
		REGC = 4.7 μF, 副系统时钟运行 (RC 振荡器)	12.5 <sup>注</sup>		27.5 <sup>注</sup>	kHz

注 内部系统时钟频率为振荡器频率的一半。

## 25.4 振荡器特性

### 25.4.1 主时钟振荡器特性

( $T_A = 40$  至  $+85^\circ\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ )

振荡器	推荐电路	参数	条件	最小	通常	最大	单位
陶瓷振荡器		振荡频率 ( $f_x$ ) <sup>注 1</sup>		4		5	MHz
		振荡稳定时间 <sup>注 2</sup>	释放复位功能之后		$2^{16}/f_x$		s
			释放停止模式之后	$0.5$ <sup>注 3</sup>	注 4		ms
			释放 IDLE2 模式之后	$0.35$ <sup>注 3</sup>	注 4		ms
晶体振荡器		振荡频率 ( $f_x$ ) <sup>注 1</sup>		4		5	MHz
		振荡稳定时间 <sup>注 2</sup>	释放复位功能之后		$2^{16}/f_x$		s
			释放停止模式之后	$0.5$ <sup>注 3</sup>	注 4		ms
			释放 IDLE2 模式之后	$0.35$ <sup>注 3</sup>	注 4		ms

- 注
1. 以上振荡频率仅表示振荡器的特性。
  2. 释放复位或停止模式后稳定振荡所需的时间。
  3. 稳定访问内部闪存所需的时间。
  4. 该值根据 OSTS 寄存器设定不同而有所变化。

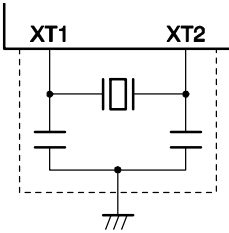
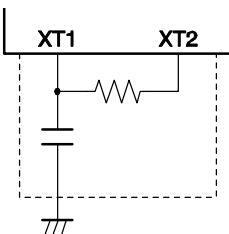
注意事项 1. 使用主时钟振荡器时，如下在上图虚线封闭的区域接线以避免接入电容的不利影响。

- 使接线长度尽量短。
- 严禁将接线与其他信号线交叉。
- 严禁在高波动电流流过的信号线附近布线。
- 时钟使振荡器 电容器的接地点电位与  $V_{SS}$  相同。
- 严禁将电容器接入高电流流过的接地模式。
- 严禁从振荡器获取信号。

2. 主时钟停止且设备按副时钟运行时，切换回主时钟之前，等待到振荡稳定时间已经由程序保障为止。

## 25.4.2 副时钟振荡器特性

(T<sub>A</sub> = 40 至 +85°C, V<sub>DD</sub> = E<sub>VDD</sub> = 3.5 V 至 5.5 V, 4.0 V ≤ A<sub>VREF0</sub> ≤ 5.5 V, V<sub>SS</sub> = E<sub>VSS</sub> = A<sub>VSS</sub> = 0 V)

振荡器	推荐电路	参数	条件	最小	通常	最大	单位
晶体振荡器		振荡频率 (f <sub>XT</sub> ) <sup>註 1</sup>		32	32.768	35	kHz
		振荡稳定时间 <sup>註 2</sup>				10	s
RC 振荡器		振荡频率 <sup>註 1, 4</sup>	R = 390 kΩ ±5% <sup>註 3</sup> C = 47 pF ±10% <sup>註 3</sup>	25	40	55	kHz
		振荡稳定时间 <sup>註 2</sup>				100	μs

- 注
1. 以上所示振荡频率仅表示振荡器特性。CPU 运行时钟， 见 25.8 AC 特性。
  2. 从 V<sub>DD</sub> 达到振荡器电压范围 (最小 3.5 V) 时到振荡器稳定时所需的时间。
  3. 使接线长度尽量短，以避免接入电容的不利影响。
  4. RC 振荡器频率为 40 kHz (通常)。 This clock is internally divided by 2. In the case of the RC resonator, 内部系统时钟频率为振荡器频率的一半。最小= 12.5 kHz, 通常 = 20 kHz, 最大= 27.5 kHz。

注意事项 1. 使用副时钟振荡器时，如下在上图虚线封闭的区域接线以避免接入电容的不利影响。

- 使接线长度尽量短。
  - 严禁将接线与其他信号线交叉。
  - 严禁在高波动电流流过的信号线附近布线。
  - 时钟使振荡器 电容器的接地点电位与 V<sub>SS</sub> 相同。
  - 严禁将电容器接入高电流流过的接地模式。
  - 严禁从振荡器获取信号。
2. 副时钟振荡器设计为低幅度电路以降低电流消耗，而且由于大于主时钟振荡器的噪声，所以更容易出故障。因此使用副时钟时，更应注意配线方法。

25.4.3 PLL 特性

(T<sub>A</sub> = 40 至 +85°C, V<sub>DD</sub> = EV<sub>DD</sub> = 3.5 V 到 5.5 V, 4.0 V ≤ AV<sub>REF0</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS</sub> = AV<sub>SS</sub> = 0 V)

参数	符号	条件	最小	通常	最大	单位
输入频率	f <sub>x</sub>		4		5	MHz
输出频率	f <sub>xx</sub>		16		20	MHz
锁定时间	t <sub>PLL</sub>	V <sub>DD</sub> 达到.: 3.5 V（最小）之后			800	μs

25.4.4 内部振荡器特性

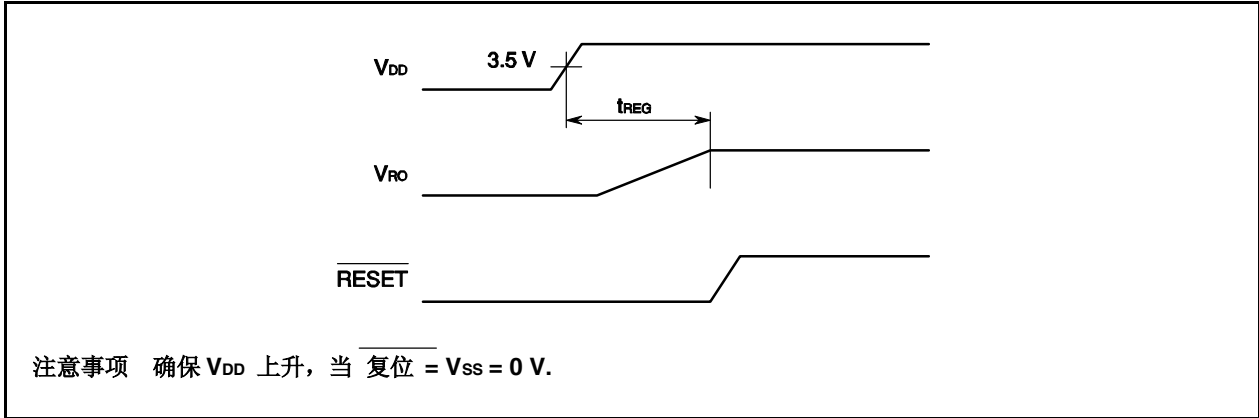
(T<sub>A</sub> = 40 至 +85°C, V<sub>DD</sub> = EV<sub>DD</sub> = 3.5 V 至 5.5 V, 4.0 V ≤ AV<sub>REF0</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS</sub> = AV<sub>SS</sub> = 0 V)

参数	符号	条件	最小	通常	最大	单位
输出频率	f <sub>R</sub>		100	200	400	kHz

25.5 电压调节器特性

(T<sub>A</sub> = 40 至 +85°C, V<sub>DD</sub> = EV<sub>DD</sub>, V<sub>SS</sub> = EV<sub>SS</sub> = AV<sub>SS</sub> = 0 V)

参数	符号	条件	最小	通常	最大	单位
输入频率	V <sub>DD</sub>		3.5		5.5	V
输出频率	V <sub>RO</sub>			2.5		V
锁定时间	t <sub>REG</sub>	V <sub>DD</sub> 达到 3.5 V（最小）之后 C = 4.7 μF ±20% 连接到 REGC 引脚			1	ms



## 25.6 DC 特性

## 25.6.1 I/O 电平

( $T_A = 40$  至  $+85^\circ\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ )

(1/2)

参数	符号	条件	最小	通常	最大	单位
输入电压, 高	$V_{IH1}$	P30, P34, P38, P41, P98, PCM0 至 PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 至 PDL11	$0.7EV_{DD}$		$EV_{DD}$	V
	$V_{IH2}$	P00 至 P06, P31 至 P33, P35, P39, P40, P42, P50 至 P55, P90, P91, P96, P97, P99, P913 至 P915	$0.8EV_{DD}$		$EV_{DD}$	V
	$V_{IH4}$	P70 至 P711	$0.7AV_{REF0}$		$AV_{REF0}$	V
	$V_{IH5}$	$\overline{RESET}$ , FLMD0	$0.8EV_{DD}$		$EV_{DD}$	V
输入电压, 低	$V_{IL1}$	P30, P34, P38, P41, P98, PCM0 至 PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 至 PDL11	$EV_{SS}$		$0.3EV_{DD}$	V
	$V_{IL2}$	P00 至 P06, P31 至 P33, P35, P39, P40, P42, P50 至 P55, P90, P91, P96, P97, P99, P913 至 P915	$EV_{SS}$		$0.2EV_{DD}$	V
	$V_{IL4}$	P70 至 P711	$AV_{SS}$		$0.3AV_{REF0}$	V
	$V_{IL5}$	$\overline{RESET}$ , FLMD0	$EV_{SS}$		$0.2EV_{DD}$	V

**备注** 除非另有规定, 复用功能引脚的特性与端口引脚的特性相同。

( $T_A = 40$  至  $+85^\circ\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ )

(2/2)

参数	符号	条件	最小	通常	最大	单位
输出电压, 高 <sup>1</sup>	$V_{OH1}$	P00 至 P06, P30 至 P35, P38, P39, P40 至 P42, P50 至 P55, P90, P91, P96 至 P99, P913 至 P915, PCM0 至 PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 至 PDL11	$I_{OH} = -1.0\text{ mA}$	$EV_{DD} - 1.0$		$EV_{DD}$ V
			$I_{OH} = -0.1\text{ mA}$	$EV_{DD} - 0.5$		$EV_{DD}$ V
	$V_{OH3}$	P70 至 P711	$I_{OH} = -1.0\text{ mA}$	$AV_{REF0} - 1.0$		$AV_{REF0}$ V
			$I_{OH} = -0.1\text{ mA}$	$AV_{REF0} - 0.5$		$AV_{REF0}$ V
输出电压, 低 <sup>1</sup>	$V_{OL1}$	P00 至 P06, P30 至 P35, P38, P39, P40 至 P42, P50 至 P55, P90, P91, P96 至 P99, P913 至 P915, PCM0 至 PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 至 PDL11	$I_{OL} = 1.0\text{ mA}$	0	0.4	V
	$V_{OL3}$	P70 至 P711	$I_{OL} = 1.0\text{ mA}$	0	0.4	V
软件上拉电阻	$R_1$	$V_I = 0\text{ V}$	10	30	100	k $\Omega$
软件下拉电阻 <sup>2</sup>	$R_2$	$V_I = V_{DD}$	10	30	100	k $\Omega$

- 注
1.  $I_{OH}/I_{OL}$  的各电源电压的最大总值为 20 mA/-20 mA ( $EV_{DD}$ ,  $AV_{REF0}$ )。
  2. 仅  $\overline{DRST}$  引脚

备注 除非另有规定, 复用功能引脚的特性与端口引脚的特性相同。

## 25.6.2 引脚漏电流

( $T_A = 40$  至  $+85^\circ\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ )

参数	符号	条件	最小	通常	最大	单位
输入漏电流, 高 <sup>1</sup>	$I_{LH1}$	$V_{IN} = V_{DD}$	模拟引脚		+0.2	$\mu\text{A}$
			模拟引脚除外		+0.5	
输入漏电流, 低 <sup>1</sup>	$I_{LH1}$	$V_{IN} = 0\text{ V}$	模拟引脚		-0.2	$\mu\text{A}$
			模拟引脚除外		-0.5	
输出漏电流, 高	$I_{LOH1}$	$V_O = V_{DD}$	模拟引脚		+0.2	$\mu\text{A}$
			模拟引脚除外		+0.5	
输出漏电流, 低	$I_{LOH1}$	$V_O = 0\text{ V}$	模拟引脚		-0.2	$\mu\text{A}$
			模拟引脚除外		-0.5	

注 FLMD0 引脚值如下:

- 输入漏电流, 高: 2  $\mu\text{A}$  (最大)
- 输入漏电流, 低: -2  $\mu\text{A}$  (最大)

## 25.6.3 电源电流

(T<sub>A</sub> = 40 至 +85°C, V<sub>DD</sub> = EV<sub>DD</sub> = 3.5 V 至 5.5 V, 4.0 V ≤ AV<sub>REF0</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS</sub> = AV<sub>SS</sub> = 0 V)

参数	符号	条件		最小	通常	最大	单位	
电源电流 <sup>註 1</sup>	I <sub>DD1</sub>	正常运行模式	f <sub>xx</sub> = 20 MHz (f <sub>x</sub> = 5 MHz)	所有外围设备功能运行		25	40	mA
				所有外围设备功能停止		20		mA
	I <sub>DD2</sub>	HALT 模式	f <sub>xx</sub> = 20 MHz (f <sub>x</sub> = 5 MHz)	所有外围设备功能运行		14	24	mA
				所有外围设备功能停止		9		mA
	I <sub>DD3</sub>	IDLE1 模式	f <sub>xx</sub> = 5 MHz (f <sub>x</sub> = 5 MHz), PLL off			0.6	0.9	mA
	I <sub>DD4</sub>	IDLE2 模式	f <sub>xx</sub> = 5 MHz (f <sub>x</sub> = 5 MHz), PLL off			0.25	0.7	mA
	I <sub>DD5</sub>	副时钟运行模式 <sup>註 2, 3</sup>	晶体振荡器 (f <sub>XT</sub> = 32.768 kHz)			200	400	μA
			RC 振荡器(f <sub>XT</sub> = 40 kHz <sup>註 4</sup> )			200	400	μA
	I <sub>DD6</sub>	副空闲模式 <sup>註 2, 3</sup>	晶体振荡器 (f <sub>XT</sub> = 32.768 kHz)			20	120	μA
			RC 振荡器 (f <sub>XT</sub> = 40 kHz <sup>註 4</sup> )			35	140	μA
	I <sub>DD7</sub>	停止模式 <sup>註 2, 5</sup>	POC 停止, 内部振荡器停止			7	50	μA
			POC 运行, 内部振荡器停止			10	55	μA
			POC 停止, 内部振荡器运行			15	65	μA
			POC 运行, 内部振荡器运行			18	70	μA

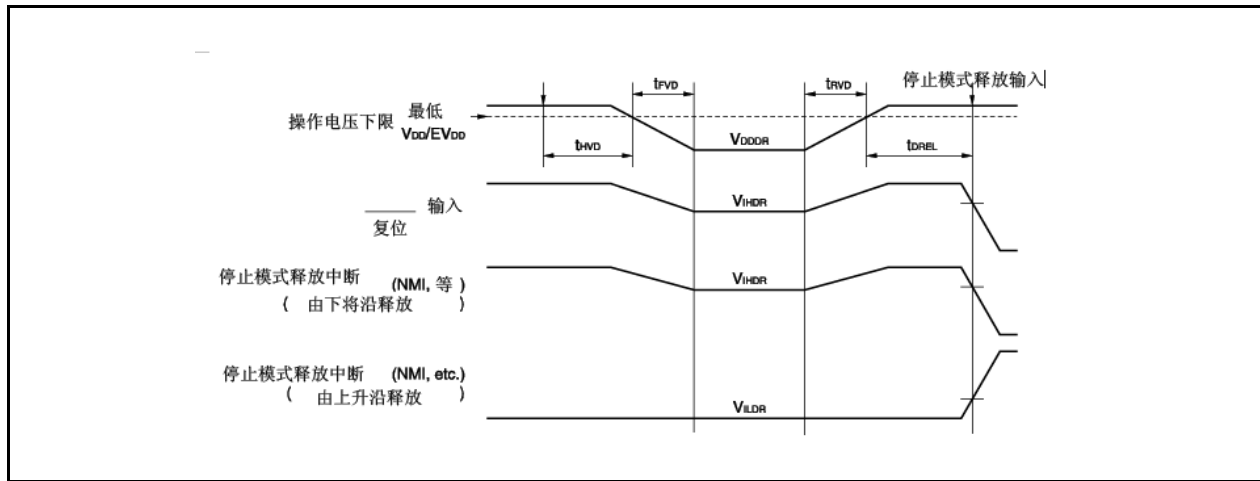
- 注
1. V<sub>DD</sub> 和 EV<sub>DD</sub> 的总电流 (所有端口停止)。流过 AV<sub>REF0</sub> 电流和端口缓冲器电流 (包括流过片上上拉/下拉电阻的电流) 不含在内。
  2. 当主时钟振荡器停止。
  3. POC 运行, 内部振荡器运行。
  4. RC 振荡器频率为 40 kHz (TYP.)。该时钟可以被 2 整除。
  5. 当不使用副时钟振荡器时。

## 25.7 数据保持特性

停止模式下 ( $T_A = 40$  至  $+85^{\circ}\text{C}$ ,  $V_{DD} = EV_{DD} = 1.9\text{ V}$  至  $5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ )

参数	符号	条件	最小	通常	最大	单位
数据保持电压	$V_{DDDR}$	停止模式 (所有功能停止)	1.9		5.5	V
数据保持电流	$I_{DDDR}$	$V_{DDDR} = 2.0\text{ V}$ (所有功能停止)		6	45	$\mu\text{A}$
电源电压上升时间	$t_{rVD}$		1			$\mu\text{s}$
电源电压下降时间	$t_{fVD}$		1			$\mu\text{s}$
电源电压保持时间	$t_{HVD}$	释放停止模式后	0			ms
停止释放信号输入时间	$t_{DREL}$	$V_{DD}$ 达到 $3.5\text{ V}$ (最小) 后	0			$\mu\text{s}$
数据保持输入电压, 高	$V_{IHDR}$	所有输入端口	$0.9V_{DDDR}$		$V_{DDDR}$	V
数据保持输入电压, 低	$V_{ILDR}$	所有输入端口	0		$0.1V_{DDDR}$	V

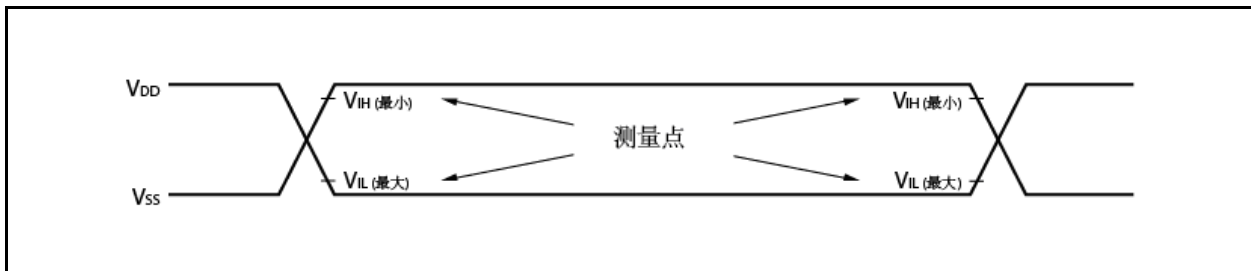
**注意事项** 切换到停止模式且从停止模式恢复必须在分级运行范围内进行。



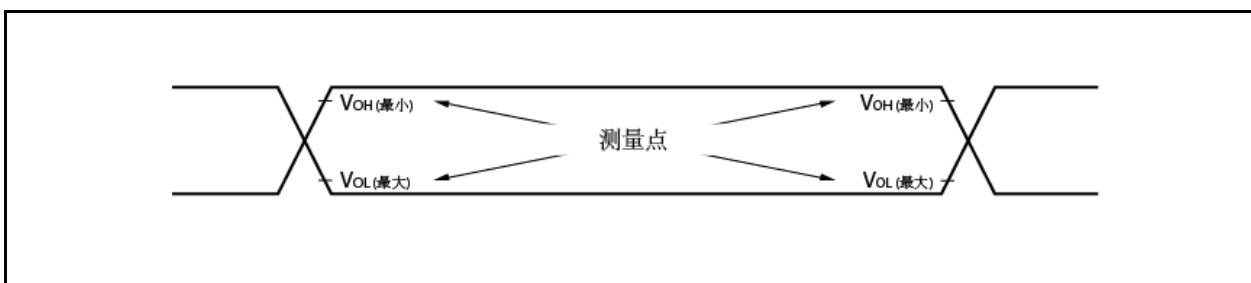


## 25.8 AC 特性

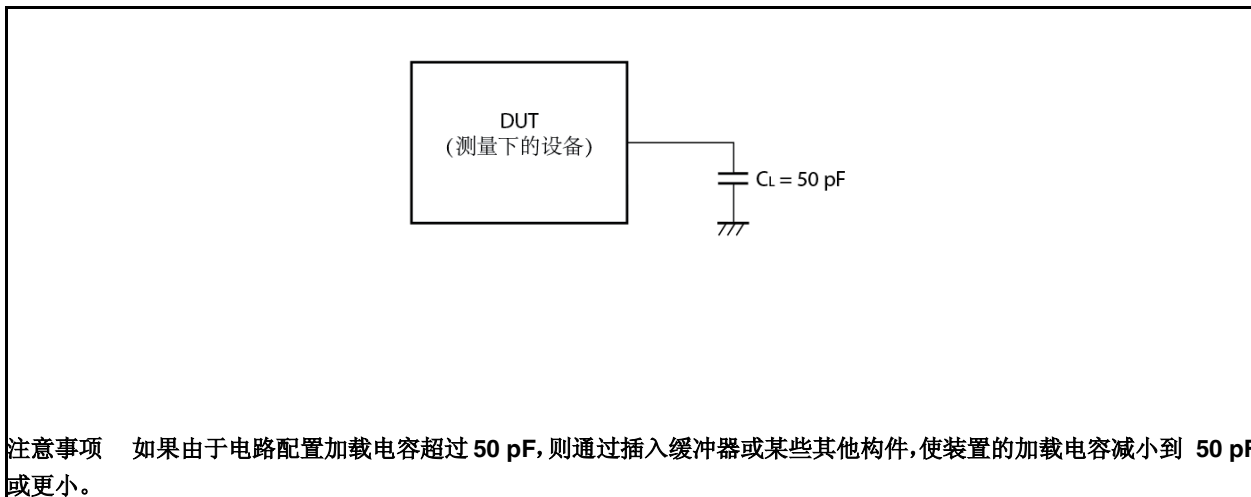
### (1) AC 测试输入测量点 ( $V_{DD}$ , $AV_{REF0}$ , $EV_{DD}$ )



### (2) AC 测试输出测量点



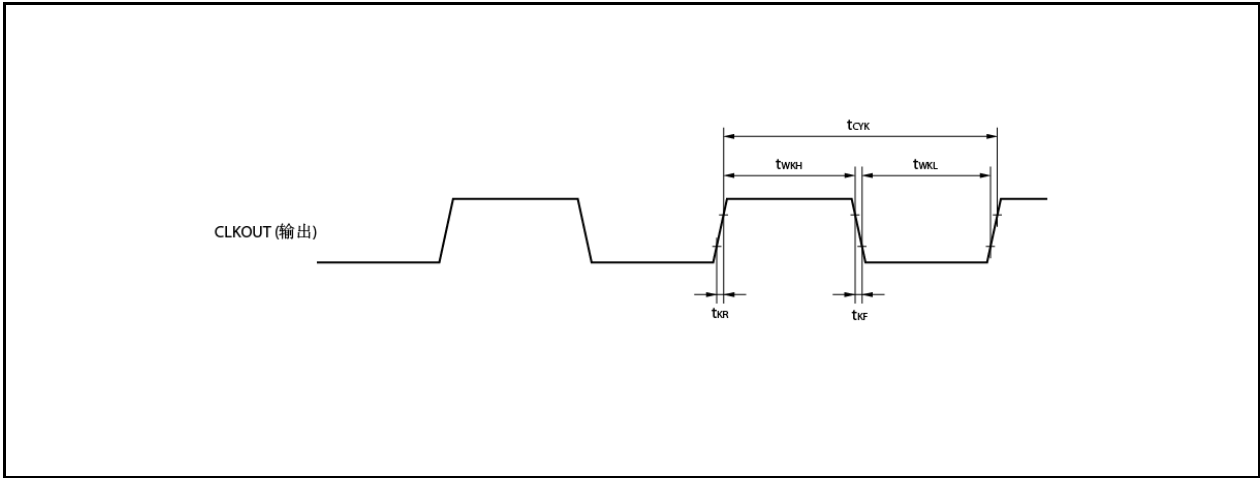
### (3) 加载条件



25.8.1 CLKOUT 输出时序

(T<sub>A</sub> = 40 至 +85°C, V<sub>DD</sub> = EV<sub>DD</sub> = 3.5 V 至 5.5 V, 4.0 V ≤ AV<sub>REF0</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS</sub> = AV<sub>SS</sub> = 0 V, C<sub>L</sub> = 50 pF)

参数	符号	条件	最小	最大	单位
输出周期	t <sub>CYK</sub>		50 ns	80 μs	
高电平宽度	t <sub>WKH</sub>		t <sub>CYK</sub> /2 - 15		ns
低电平宽度	t <sub>WKL</sub>		t <sub>CYK</sub> /2 - 15		ns
上升时间	t <sub>KR</sub>			15	ns
下降时间	t <sub>KF</sub>			15	ns



25.9 基本操作

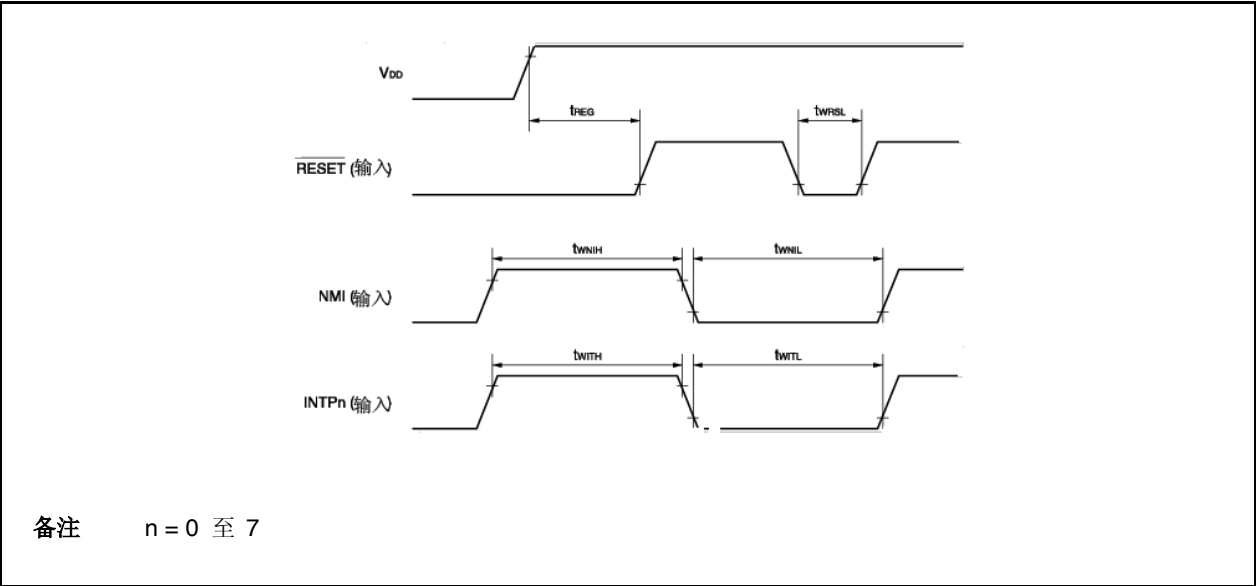
(1) 复位， 中断时序

( $T_A = 40$  至  $+85^{\circ}\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ ,  $C_L = 50\text{ pF}$ )

参数	符号	条件	最小	最大	单位
RESET 低电平宽度	$t_{WRS\overline{L}}$		500		ns
NMI 高电平宽度	$t_{WNIH}$	模拟消噪	500		ns
NMI 低电平宽度	$t_{WNIL}$	模拟消噪	500		ns
INTP <sub>n</sub> <sup>※1</sup> 高电平宽度	$t_{WITH}$	模拟消噪 ( $n = 0$ 至 $7$ )	500		ns
		数字消噪 ( $n = 3$ )	注 2		ns
INTP <sub>n</sub> <sup>※1</sup> 低电平宽度	$t_{WITL}$	模拟消噪( $n = 0$ 至 $7$ )	500		ns
		数字消噪 ( $n = 3$ )	注 2		ns

注 1. INTP0/P03 引脚与 ADTRG 引脚情况下应用相同值。INTP2/P05 引脚在  $\overline{DRST}$  引脚情况下应用相同值。  
2.  $2T$  采样+ 20 或  $3T$  采样 + 20  
T 采样：样时钟用于噪音消除。

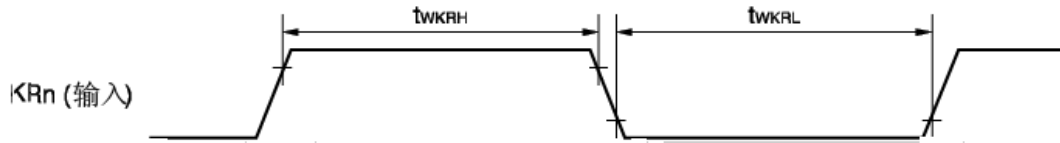
复位/中断



## (2) 按键中断时序

( $T_A = 40$  至  $+85^\circ\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ ,  $C_L = 50\text{ pF}$ )

参数	符号	条件	最小	最大	单位
KRn 输入高电平宽度	$t_{WKRH}$	模拟消噪 ( $n = 0$ 至 $7$ )	500		ns
KRn 输入低电平宽度	$t_{WKRL}$		500		ns



备注  $n = 0$  至  $7$

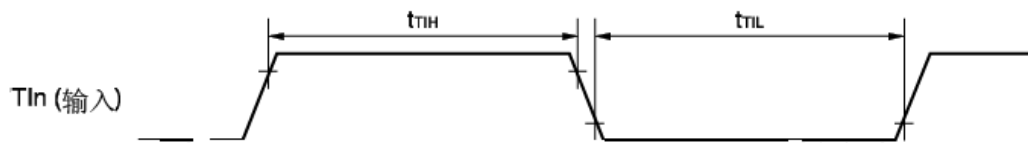
## (3) 定时器输入时序

( $T_A = 40$  至  $+85^\circ\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ ,  $C_L = 50\text{ pF}$ )

参数	符号	条件	最小	最大	单位
高电平宽度	$t_{TIH}$	TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, TIP31, TIQ00 至 TIQ03 <sup>※1</sup>	注 2		ns
低电平宽度	$t_{TIL}$		注 2		ns

注 1. 仅输入捕捉信号时，可以消除 TIP00, TIP10, TIP20, TIP30 和 TIQ00 引脚噪音。  
当输入外在触发器信号或外时间计数器信号，不能消除噪音。

2.  $2T_{\text{采样}} + 20$  or  $3T_{\text{采样}} + 20$   
 $T_{\text{采样}}$ : 噪音消除采样时钟



备注  $TIn$ : TIP00, TIP01, TIP10 TIP11, TIP20, TIP21, TIP30, TIP31, TIQ00 至 TIQ03

## (4) CSIB 时序

## (a) 主模式

(TA = 40 至 +85°C, VDD = EVDD = 3.5 V 至 5.5 V, 4.0 V ≤ AVREF0 ≤ 5.5 V, VSS = EVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小	最大	单位
SCKBn 周期	tKCYn		125		ns
SCKBn 高电平宽度	tKHn		tKCYn/2 - 15		ns
SCKBn 低电平宽度	tKLn		tKCYn/2 - 15		ns
SIBn 加载时间 (至 SCKBn↑)	tSIKn		30		ns
SIBn 保持时间 (从 SCKBn↑)	tSIn		25		ns
输出延迟时间 SCKBn↓至 SOBn	tKSO n			25	ns

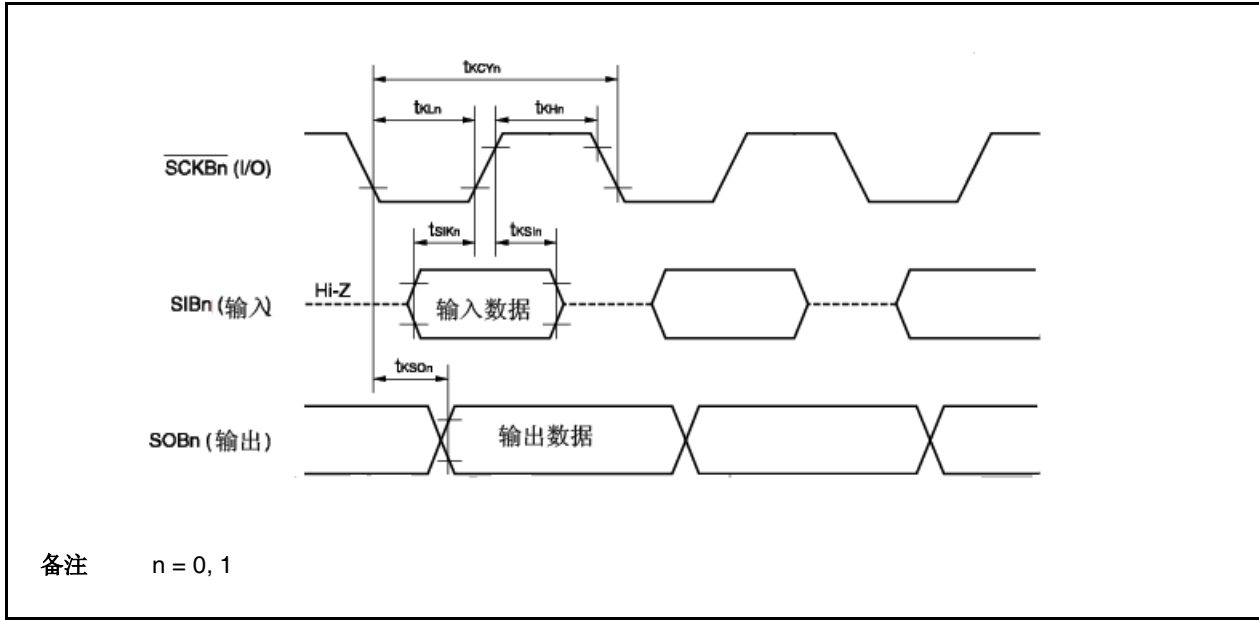
备注 n = 0, 1

## (b) 从属模式

(TA = 40 至 +85°C, VDD = EVDD = 3.5 V 至 5.5 V, 4.0 V ≤ AVREF0 ≤ 5.5 V, VSS = EVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小	最大	单位
SCKBn 周期时间	tKCYn		200		ns
SCKBn 高电平宽度	tKHn		90		ns
SCKBn 低电平宽度	tKLn		90		ns
SIBn 加载时间 (至 SCKBn↑)	tSIKn		50		ns
SIBn 保持时间 (从 SCKBn↑)	tSIn		50		ns
输出延迟时间从 SCKBn↓至 SOBn	tKSO n			50	ns

备注 n = 0, 1



#### (5) UARTA 时序

( $T_A = 40$  至  $+85^{\circ}\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ ,  $C_L = 50\text{ pF}$ )

参数	符号	条件	最小	最大	单位
通讯率				312.5	kbps
ASCK0 周期时间				10	MHz

#### (6) A/D 转换器

( $T_A = 40$  至  $+85^{\circ}\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ ,  $C_L = 50\text{ pF}$ )

参数	符号	条件	最小	通常	最大	单位
分辨率					10	bit
整体误差 <sup>注</sup>		$4.0 \leq AV_{REF0} \leq 5.5\text{ V}$		$\pm 0.15$	$\pm 0.3$	%FSR
转化时间	$t_{CONV}$		3.1		16	$\mu\text{s}$
模拟输入电压	$V_{IAN}$		$AV_{SS}$		$AV_{REF0}$	V
AVREF0 电流	$I_{AREF0}$	当使用 A/D 转换器		5	10	mA
		当不使用 A/D 转换器		1	10	$\mu\text{A}$

注 不包括量化误差( $\pm 0.05\%$ FSR)。表示满标度值的比率(%FSR)。

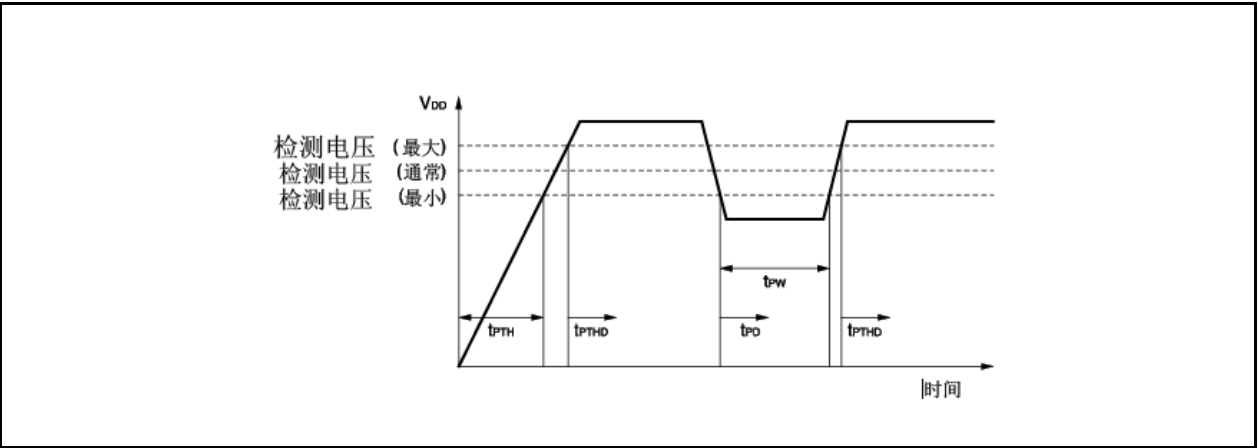
备注 FSR: 满标度量程

(7) POC 电路特性

( $T_A = 40$  至  $+85^{\circ}\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ ,  $C_L = 50\text{ pF}$ )

参数	符号	条件	最小	通常	最大	单位
检测电压	$V_{POC0}$		3.5	3.7	3.9	V
电源电压启动时间	$t_{PTH}$	$V_{DD} = 0\text{ V} \rightarrow 3.5\text{ V}$	0.002			ms
响应推迟时间 1 <sup>注1</sup>	$t_{PTH0}$	电源应用程序 $V_{DD}$ 达到 $3.9\text{ V}$ 之后			3.0	ms
响应推迟时间 2 <sup>注2</sup>	$t_{PD}$	电源下降 $V_{DD}$ 降落到 $3.5\text{ V}$ 之后			1	ms
最小 $V_{DD}$ 宽度	$t_{PW}$		0.2			ms

- 注
- 1. 检测检测电压后释放复位所需的时间。
  - 2. 检测检测电压后输出复位所需的时间。

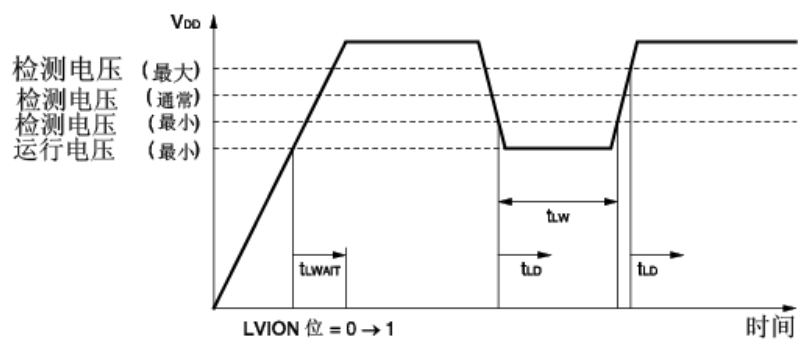


## (8) LVI 电路特性

( $T_A = 40$  至  $+85^{\circ}\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ ,  $C_L = 50\text{ pF}$ )

参数	符号	条件	最小	通常	最大	单位
检测电压	$V_{LV10}$		4.2	4.4	4.6	V
	$V_{LV11}$		4.0	4.2	4.4	V
响应时间 <sup>注 1</sup>	$t_{LD}$	在 $V_{DD}$ 达到 $V_{LV10}/V_{LV11}$ (最大) 之后或 降落到 $V_{LV10}/V_{LV11}$ (最小) 之后		0.2	2	ms
最小 $V_{DD}$ 宽度	$t_{LW}$		0.2			ms
参考电压稳定等待时间 <sup>注 2</sup>	$t_{LWAIT}$	在 $V_{DD}$ 达到 $3.5\text{ V}$ 之后或 $LVION$ 位(LVIM.位 7) 从 0 到 1 变化		0.1	0.2	ms

- 注 1. 检测检测电压后输出中断/复位信号所需的时间。  
 2. 使用 POC 功能时不可用。



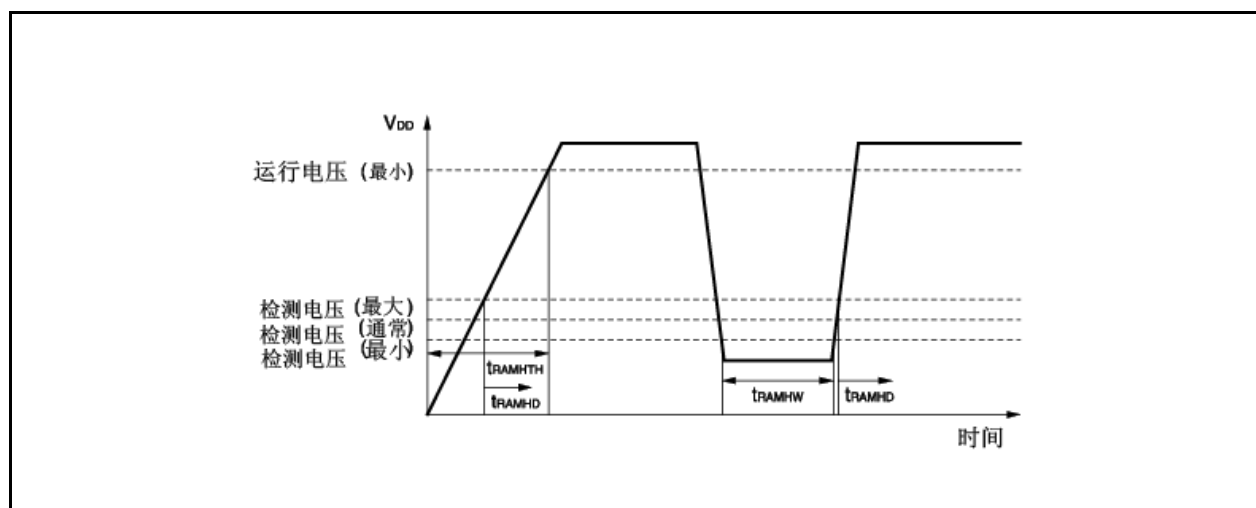


## (9) RAM 保持标记特性

( $T_A = 40$  至  $+85^{\circ}\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ ,  $C_L = 50\text{ pF}$ )

参数	符号	条件	最小	通常	最大	单位
检测电压	$V_{RAMH}$		1.9	2.0	2.1	V
电源电压上升时间	$t_{RAMHTH}$	$V_{DD} = 0\text{ V} \rightarrow 3.5\text{ V}$	0.002		1800	ms
响应时间 <sup>注</sup>	$t_{RAMHD}$	电源电压达到检测电压后 (MAX)		0.2	2.0	ms
最小 $V_{DD}$ 宽度	$t_{RAMHW}$		0.2			ms

注 检测检测电压后设置 RAMF 位所需的时间。



## 25.10 闪存编程特性

## (1) 基本特性

( $T_A = -40$  至  $+85^{\circ}\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ ,  $C_L = 50\text{ pF}$ )

参数	符号	条件	最小	通常	最大	单位
运行频率	$f_{CPU}$		4		20	MHz
电源电压	$V_{DD}$		3.5		5.5	V
写入次数	$C_{WRT}^{\#}$				100	次数
输入电压, 高	$V_{IH}$	FLMD0	$0.8EV_{DD}$		$EV_{DD}$	V
输入电压, 低	$V_{IL}$	FLMD0	$EV_{SS}$		$0.2EV_{SS}$	V
写入时间 + 擦除时间	$t_{WRT} + t_{ERASE}$				TBD	s
编程温度	$t_{PRG}$		-40		+85	$^{\circ}\text{C}$

注 当开始写入到出货产品时, “擦除待写入”与“仅写入”都记做重写方式。

示例 (P: 写入, E: 擦除)

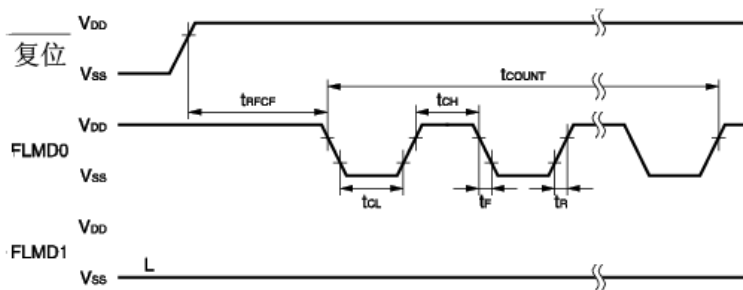
出货产品  $\rightarrow P \rightarrow E \rightarrow P \rightarrow E \rightarrow P$ : 3 重写

出货产品  $\rightarrow E \rightarrow P \rightarrow E \rightarrow P \rightarrow E \rightarrow P$ : 3 重写

## (2) 串口写入运行特性

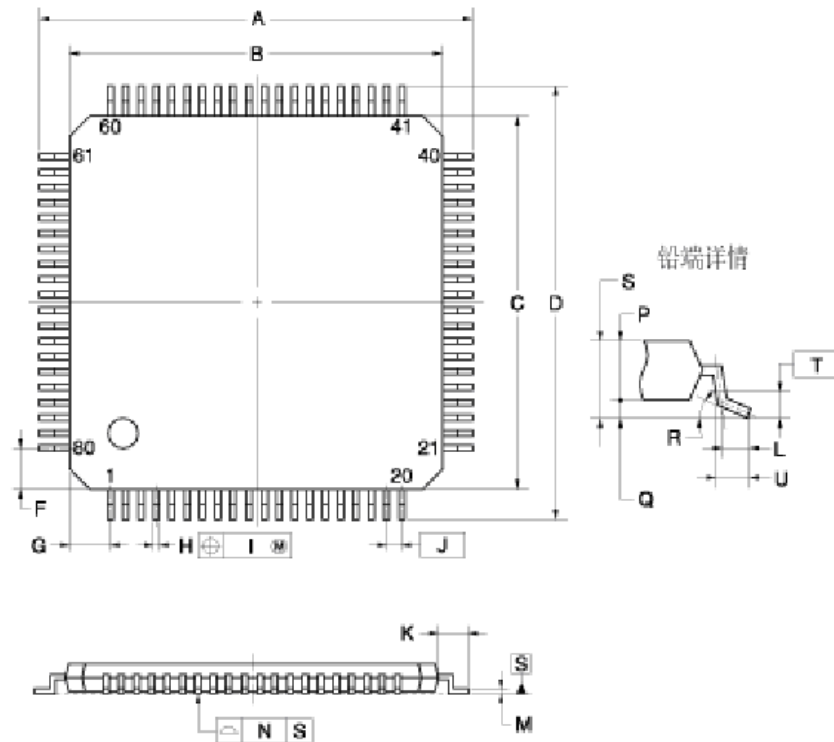
( $T_A = -40$  至  $+85^{\circ}\text{C}$ ,  $V_{DD} = EV_{DD} = 3.5\text{ V}$  至  $5.5\text{ V}$ ,  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$ ,  $C_L = 50\text{ pF}$ )

参数	符号	条件	最小	通常	最大	单位
FLMD0 从 RESET $\uparrow$ 设置时间	$t_{RFCF}$		$70536/f_x$			s
计数执行时间	$t_{COUNT}$				3	ms
FLMD0 高电平宽度	$t_{CH}$		10		100	$\mu\text{s}$
FLMD0 低电平宽度	$t_{CL}$		10		100	$\mu\text{s}$
FLMD0 上升时间	$t_r$				50	ns
FLMD0 下降时间	$t_f$				50	ns



## 第二十六章 封装图

80-引脚塑料TQFP封装（精密间距）（12×12）



注:

每条铅中心线位于最大材料条件下其真实位置的0.08mm范围内。

物品尺寸	
A	14.0±0.2
B	12.0±0.2
C	12.0±0.2
D	14.0±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	0.145±0.05
N	0.08
P	1.0
Q	0.1±0.05
R	3°~4°
S	1.1±0.1
T	0.25
U	0.6±0.15
P80GK-50-9EU-1	

V850ES/HF2 应该在以下推荐条件下焊接和安装。

有关技术信息，请参阅以下网站。

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

表 27-1. 表面安装类型焊接条件

**μPD70F3702GK-9EU-A: 80-引脚塑料 TQFP 封装 (密脚距) (12 × 12)**

**μPD70F3703GK-9EU-A: 80-引脚塑料 TQFP 封装 (密脚距) (12 × 12)**

**μPD70F3704GK-9EU-A: 80-引脚塑料 TQFP 封装 (密脚距) (12 × 12)**

焊接方法	焊接条件	推荐条件符号
红外再流	封装峰值温度: 260° C, 时间: 最大 60 秒 (在 220° C 或更高时), 计数: 三次或更少, 暴露限度: 7 天 <sup>※</sup> (此后, 在 125° C 下预烘干 20 到 72 小时)	IR60-207-3
局部加热	引脚温度: 最大 350° C, 时间: 最大 3 秒 (每行引脚)	—

**注** 打开干燥包封装后, 在可允许的存储周期内在 25°C 或更低于 65%RH 或更低的条件下存储。

**注意事项** 禁止不同焊接方法一起使用 (局部加热除外)。

**备注**

1. 零件号码末端处带 -A 的产品为无铅产品。
2. 对于以上推荐之外的焊接方法和条件, 请联系 NEC Electronics 销售代表。

V850ES/HF2 的系统开发中使用以下开发工具。

图 A-1 显示了开发工具的组成。

- **支持 PC98-NX 系列**

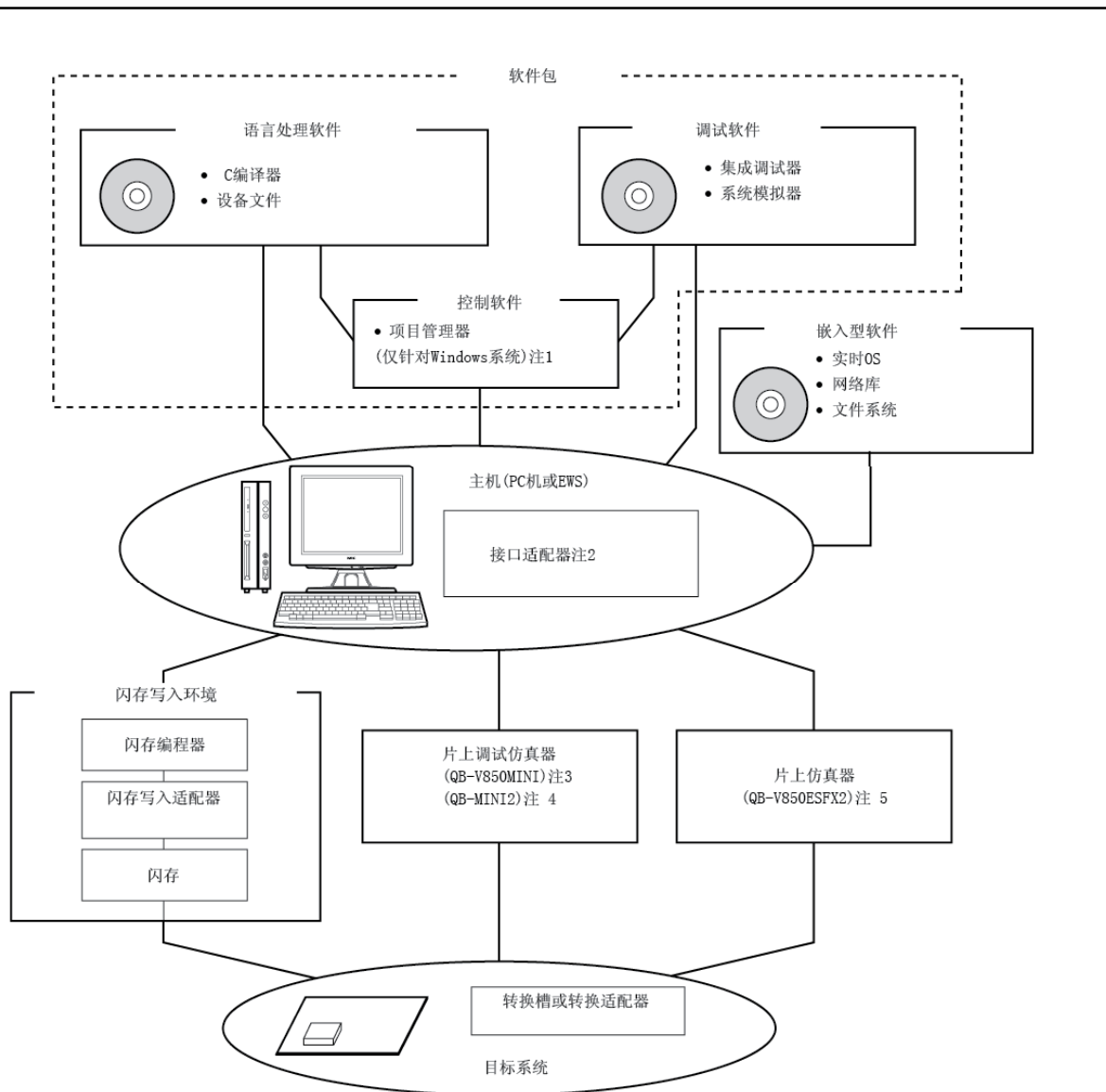
除非特别说明，IBM PC/AT™ 兼容机支持的产品与PC98-NX 系列计算机是兼容的。当使用PC98-NX 系列计算机时，参阅 IBM PC/AT 兼容机的使用说明。

- **Windows™**

除非特别说明，“Windows”是指以下几种操作系统。

- Windows 98, 2000
- Windows Me
- Windows XP
- Windows NT™ . 4.0 版本

图 A-1. 开发工具的组成



- 注
1. C 编译器中包含有项目管理器 PM+ 。  
PM+ 只能在 Windows 操作系统中使用。
  2. QB-V850MINI, QB-MINI2, 以及 QB-V850ESFX2 至支持 USB 接口。
  3. 购买 QB-V850MINI 时还提供 ID850QB, USB 接口电缆、OCD 电缆外、自检板, KEL 适配器和 KEL 连接器。其他均为可选产品。
  4. 购买 QB-MINI2 时还提供 USB 接口电缆、16 针目标电缆、10 针目标电缆以及 78K0-OCD 板卡 (不提供集成调试器。), 其他均为可选产品。
  5. 购买 QB-V850ESFX2 时还提供 ID850QB、闪存编程器 PG-FPL、供电单元以及 USB 接口适配器。其他均为可选产品。

## A.1 软件包

SP850 V850 系列软件包 微控制器	通常用于 V850 微控制器的开发工具(软件)都包含在此软件包中。.
	产品号: $\mu$ SxxxxSP850

**备注** 产品号中的xxxx 随主机和使用的 OS 的不同而有所不同。

$\mu$ SxxxxSP850

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	该编译器将 C 程序转换成微控制器可执行的目标代码。 由项目管理器启动该编译器。
	产品号: $\mu$ SxxxxCA703000
DF703724 设备文件	该文件包含设备特定信息。 该设备文件应当结合工具 (CA850, 用于 V850ES/Hx2 的 SM+, 或 ID850QB)一起使用。). 相应的 OS 和主机随所使用的工具不同而不同。

**备注** 产品号中的xxxx随主机和使用的 OS 的不同而有所不同。

$\mu$ SxxxxCA703000

xxxx	主机	OS	存储介质
AB17	PC-9800 系列, IBM PC/AT 兼容机	Windows (日文版)	CD-ROM
BB17		Windows (英文版)	
3K17	SPARC 主站™	SunOS™ (Rel. 4.1.4), Solaris™ (Rel. 2.5.1)	

## A.3 控制软件

PM+ 项目管理器	这是一套为用户设计的控制软件, 使用该软件, 用户可以在 Windows 环境下有效地进行软件开发。用户程序开发所进行的所有操作, 如启动编辑器、构建程序和启动调试器, 都可以由项目管理器 PM+执行。 <注意事项> PM+包含在 C 编译器软件包 (CA850) 中。 仅在 Windows 中使用。
--------------	--

## A.4 调试工具 (硬件)

## A.4.1 当使用在线仿真器 IECUBE QB-V850ESFX2 时

下面系统结构显示了将 QB-V850ESFX2 连接到到主机(PC-9800 系列, PC/AT 兼容机)上去的情形。即便是没有选件产品, 也可以进行连接。

图 A-2. 系统组成 (使用 QB-V850ESFX2 时) (1/2)

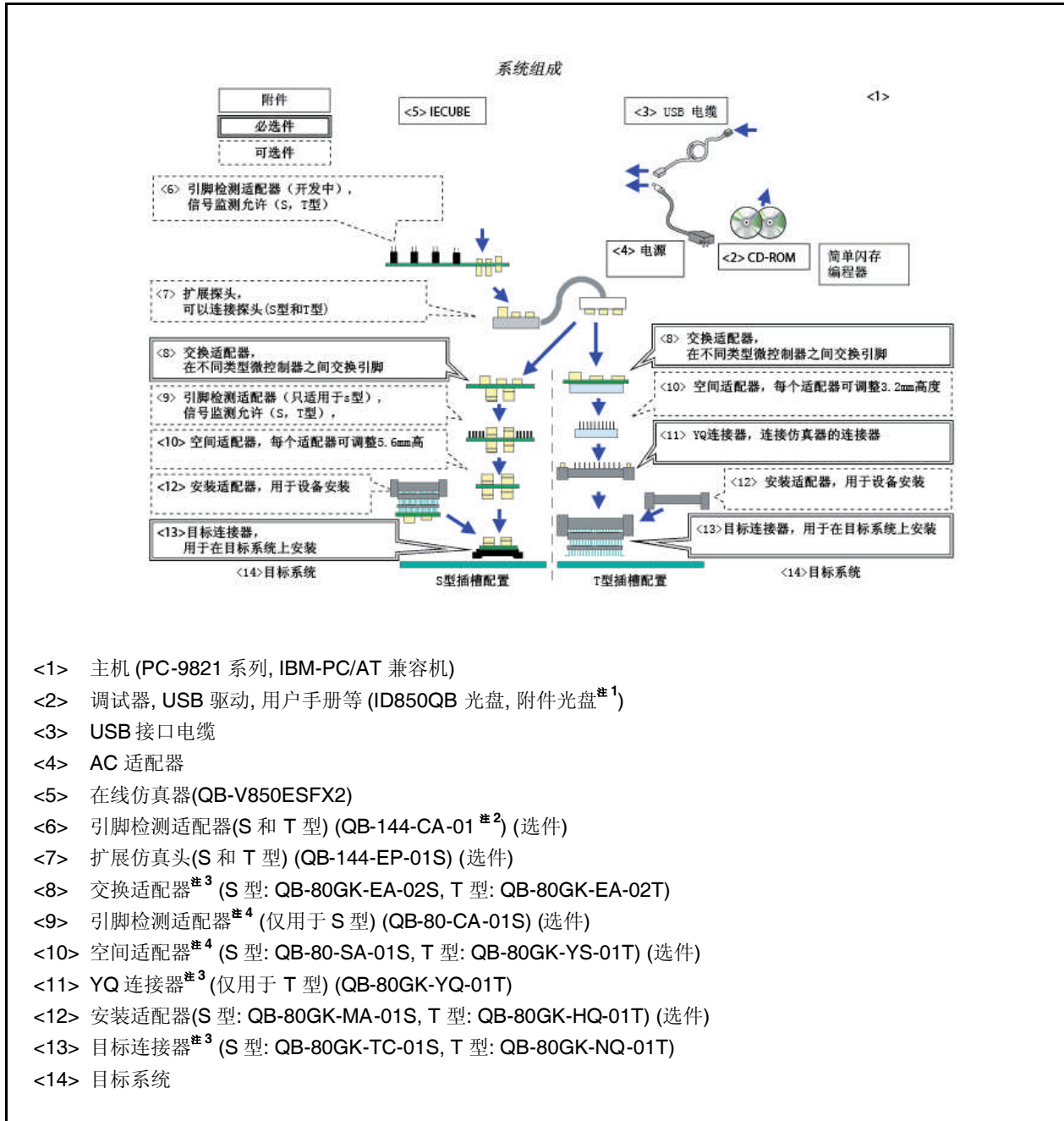




图 A-2. 系统组成 (使用 QB-V850ESFX2 时) (2/2)

- 注
1. 由日电电子网站下载设备文件。  
<http://www.necel.com/micro/ods/eng/>
  2. 正在开发中
  3. 根据订货号的不同 提供不同设备。
    - 当订购的是 QB-V850ESFX2-ZZZ 时  
不提供交换适配器和目标连接器。
    - 当订购的是 QB-V850ESFX2-S80GK 时  
提供 QB-80GK-EA-02S 和 QB-80GK-TC-01S 。
    - 当订购的是 QB-V850ESFX2-T80GK 时  
提供 QB-80GK-EA-02T, QB-80GK-YQ-01T, 和 QB-80GK-NQ-01T 。
  4. 当同时使用设备 <9> 和 <10>时, 可不必考虑其先后顺序。

<5> QB-V850ESFX2 <sup>注</sup> 在线仿真器	在使用 V850ES/HF2 产品开发应用系统时, 该在线仿真器用于调试硬件和软件。它支持集成调试器 ID850QB。该仿真器应当与供电单元、仿真头结合使用, 并使用 USB 接口与连接主机。
<3> USB 接口电缆	该电缆用以连接主机和在线仿真器 QB-V850ESFX2。
<4> AC 适配器	通过更换 AC 插头, 可支持 100 到 240 V 电压。
<8> QB-80GK-EA-02S QB-80GK-EA-02T 交换适配器	该适配器用以执行引脚转换
<9> QB-80-CA-01S 引脚检测适配器	该适配器用于示波器的波形检测等。
<10> QB-80-SA-01S QB-80GK-YS-01T 空间适配器	该适配器用于调整高度。
<11> QB-80GK-YQ-01T YQ 连接器	用于连接目标连接器和交换适配器的转换适配器。
<12> QB-80GK-MA-01S QB-80GK-HQ-01T 安装适配器	该适配器用于将 V850ES/HF2 安装到目标板上(使用插槽)。
<13> QB-80GK-TC-01S QB-80GK-NQ-01T 目标连接器	该连接器用于焊接到目标系统。

注 购买 QB-V850ESFX2 时, 还提供供电单元, USB 接口电缆以及简单编程器 PG-FPL。同时也提供控制软件集成调试器 ID850QB。

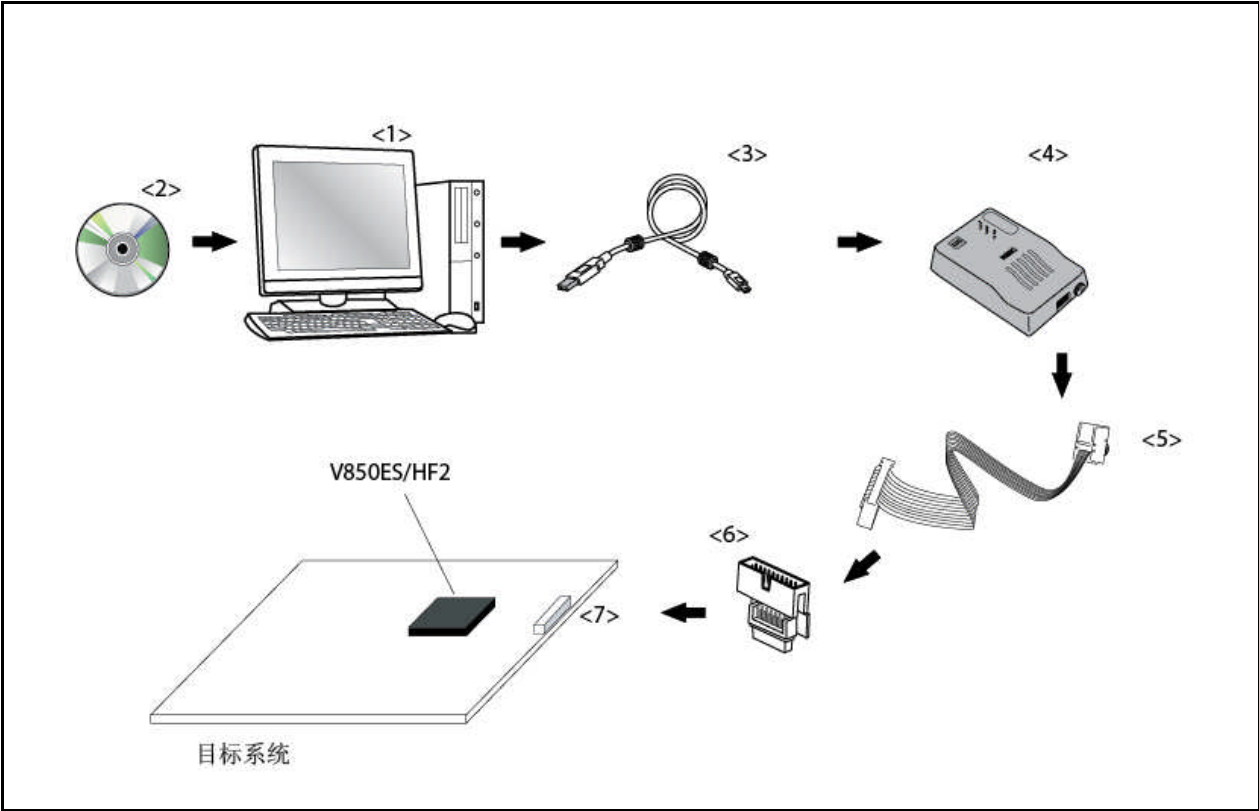
备注 尖括号中的数字与图 A-2 中的数字相对应。

A.4.2 当使用 MINICUBE QB-V850MINI 时

(1) 使用 MINICUBE 的片上仿真器

下面系统结构显示了将 MINICUBE 连接到主机(PC-9821 系列, PC/AT 兼容机)上去的情形。

图 A-3. 片上仿真器系统结构



<1> 主机	带有 USB 口的 PC 机。
<2> CD-ROM <sup>※1</sup>	该 CD-ROM 盘中包含有集成调试器 ID850QB, N 线检测器, 设备驱动软件, 文档等等。该光盘随 MINICUBE 一起提供。
<3> USB 接口电缆	该 USB 电缆用以连接主机和 MINICUBE, 随 MINICUBE 一起提供。该电缆长约 2 米。
<4> MINICUBE 片上调试仿真器	在使用 V850ES/HF2 产品开发应用系统时, 该片上调试仿真器用于调试硬件和软件。它支持集成调试器 ID850QB。
<5> OCD 电缆	该电缆用以连接 MINICUBE 和目标系统, 随 MINICUBE 一起提供。该电缆长约 20 厘米。
<6> 连接器转换板卡 KEL 适配器	该转换板卡随 MINICUBE 一起提供。
<7> MINICUBE 连接器 KEL 连接器 <sup>※2</sup>	8830E-026-170S (随 MINICUBE 一起提供) 8830E-026-170L (独立销售)

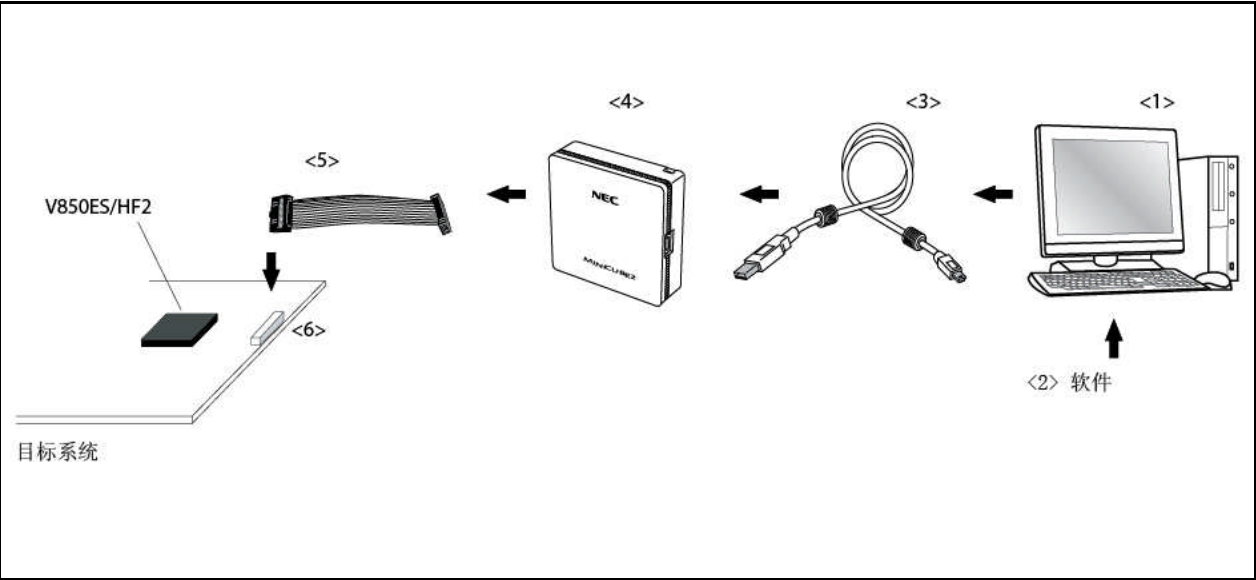
注 1. 由日电电子网站下载设备文件。  
<http://www.necel.com/micro/ods/eng/index.html>  
2. 为 KEL Corporation 公司产品。

备注 尖括号中的数字与图 A-3 中的数字相对应。

A.4.3 当时用 MINICUBE2 QB-MINI2 时

下面系统结构显示了将 MINICUBE2 连接到主机(PC-9821 系列, PC/AT 兼容机)上去的情形。

图 A-4. 片上仿真系统的系统结构



<1> 主机	带有 USB 口的 PC 机。
<2> 软件	集成调试器 ID850QB, 设备文件,等等。 由日电电子网站下载设备文件。 <a href="http://www.necel.com/micro/ods/eng/">http://www.necel.com/micro/ods/eng/</a>
<3> USB 接口电缆	该 USB 电缆用以连接主机和 MINICUBE, 随 MINICUBE 一起提供。该电缆长约 2 米。
<4> MINICUBE2 在线调试仿真器	在使用 V850ES/HF2 产品开发应用系统时, 该片上调试仿真器用于调试硬件和软件。 它支持集成调试器 ID850QB。
<5> 16 针目标电缆	该电缆用以连接 MINICUBE2 和目标系统, 随 MINICUBE 一起提供。该电缆长约 15 厘米。
<6> 目标连接器 (独立销售)	使用 2.54 mm 脚距 16-针 通用连接器。

备注 尖括号中的数字与图 A-4 中的数字相对应。

A.5 调试工具 (软件)

用于 V850ES/Hx2 的 SM+ (开发中) 系统模拟器	该模拟器适用于 V850 系列微控器产品，用于 V850ES/Hx2 的 SM+是基于 Windows 的软件。 在主机上模拟目标系统操作时，该模拟器可以对 C 源程序或汇编文件进行调试。 使用 V850ES/Hx2 的 SM+，可以在不依赖硬件开发的基础上进行应用逻辑测试和性能测试，从而提供了较高的开发效率和软件质量。 该 SM+应当结合设备文件一起使用。
	产品号：     μS××××SM703712-B
ID850QB 集成调试器	该调试器支持V850 系列微控器的在线仿真器。ID850QB 是基于Windows 的软件。它改善了C 兼容的调试功能，并使用集成窗口功能(结合源程序、分开显示和跟踪结果的存储区显示)来显示源程序的跟踪结果。 该调试器应当结合设备文件一起使用。
	产品号：     μS×××× ID703000-QB (ID850QB)

备注       产品号中的××××随主机和使用的 OS 的不同而有所不同。

μS××××ID703000-QB

××××	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.6 嵌入式软件

RX850, RX850 Pro 实时 OS	RX850 和 RX850 Pro 是符合 $\mu$ ITRON 3.0 标准规范的实时操作系统。 提供一个用于生成多个信息表的工具(配置器)。 RX850 Pro 比 RX850 具有更多功能。
	产品号: $\mu$ SxxxxRX703000- $\Delta\Delta\Delta\Delta$ (RX850) $\mu$ SxxxxRX703100- $\Delta\Delta\Delta\Delta$ (RX850 Pro)
Applilet <sup>®</sup>	这是一个驱动配置器, 它可以自动生成 V850ES/HF2 的示例程序。
RX-FS850 (系统文件)	这是一 FAT 文件系统功能。 支持 CD-ROM 系统文件功能。 该系统文件与实时 OS RX850 Pro 一起使用。

注 预了解如何获得 Applilet, 请向日电电子销售代表咨询。

注意事项 要购买 RX850 或 RX850 Pro 产品, 请先填写购买申请表并签署许可协议。

备注 产品号中的xxxx随主机和使用的 OS 的不同而有所不同。

$\mu$ SxxxxRX703000- $\Delta\Delta\Delta\Delta$

$\mu$ SxxxxRX703100- $\Delta\Delta\Delta\Delta$

$\Delta\Delta\Delta\Delta$	产品概述	批量产品中使用的最大数量
001	评估版对象	不用于批量产品
100K	批量产品对象	十万个
001M		一百万个
010M		一千万个
S01	源程序	批量产品的对象源程序

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 主站	Solaris (Rel. 2.5.1)	

**A.7 闪存写入工具**

Flashpro IV (产品号: PG-FP4) 闪存编程器	闪存编程器专门为具有片上闪存的微控制器所用。
QB-MINI2 (MINICUBE2)	带有编程功能的片上调试仿真器。
FA-80GK-9EU-A 闪存写入适配器	闪存写入适配器用于连接 Flashpro IV 等. (未连线)。
FA-70F3704GK-9EU-MX 闪存写入适配器	闪存写入适配器用于连接 Flashpro IV 等. (已连线)。

**备注** FA-80GK-9EU-A 和 FA-70F3704GK-9EU-MX 为 Naito Densei Machida Mfg. Co., Ltd 产品。  
电话: +81-42-750-4172

## 附录 B 寄存器索引

(1/7)

符号	名称	设备	页码
ADA0CR0	A/D 转换结果寄存器 0	ADC	395
ADA0CR0H	A/D 转换结果寄存器 0H	ADC	395
ADA0CR1	A/D 转换结果寄存器 1	ADC	395
ADA0CR1H	A/D 转换结果寄存器 1H	ADC	395
ADA0CR10	A/D 转换结果寄存器 10	ADC	395
ADA0CR10H	A/D 转换结果寄存器 10H	ADC	395
ADA0CR11	A/D 转换结果寄存器 11	ADC	395
ADA0CR11H	A/D 转换结果寄存器 11H	ADC	395
ADA0CR2	A/D 转换结果寄存器 2	ADC	395
ADA0CR2H	A/D 转换结果寄存器 2H	ADC	395
ADA0CR3	A/D 转换结果寄存器 3	ADC	395
ADA0CR3H	A/D 转换结果寄存器 3H	ADC	395
ADA0CR4	A/D 转换结果寄存器 4	ADC	395
ADA0CR4H	A/D 转换结果寄存器 4H	ADC	395
ADA0CR5	A/D 转换结果寄存器 5	ADC	395
ADA0CR5H	A/D 转换结果寄存器 5H	ADC	395
ADA0CR6	A/D 转换结果寄存器 6	ADC	395
ADA0CR6H	A/D 转换结果寄存器 6H	ADC	395
ADA0CR7	A/D 转换结果寄存器 7	ADC	395
ADA0CR7H	A/D 转换结果寄存器 7H	ADC	395
ADA0CR8	A/D 转换结果寄存器 8	ADC	395
ADA0CR8H	A/D 转换结果寄存器 8H	ADC	395
ADA0CR9	A/D 转换结果寄存器 9	ADC	395
ADA0CR9H	A/D 转换结果寄存器 9H	ADC	395
ADA0M0	A/D 转换模式寄存器 0	ADC	390
ADA0M1	A/D 转换模式寄存器 1	ADC	392
ADA0M2	A/D 转换模式寄存器 2	ADC	393
ADA0PFM	掉电比较模式寄存器	ADC	397
ADA0PFT	掉电比较阈值寄存器	ADC	397
ADA0S	A/D 转换通道选择寄存器 0	ADC	394
ADIC	中断控制寄存器	INTC	514
CB0CTL0	CSIB0 控制寄存器 0	CSI	454
CB0CTL1	CSIB0 控制寄存器 1	CSI	457
CB0CTL2	CSIB0 控制寄存器 2	CSI	458
CB0RIC	中断控制寄存器	INTC	514
CB0RX	CSIB0 接收数据寄存器	CSI	453
CB0RXL	CSIB0 接收数据寄存器 L	CSI	453
CB0STR	CSIB0 状态寄存器	CSI	460
CB0TIC	中断控制寄存器	INTC	514
CB0TX	CSIB0 传输数据寄存器	CSI	453
CB0TXL	CSIB0 传输数据寄存器 L	CSI	453

(2/7)

符号	名称	设备	页码
CB1CTL0	CSIB1 控制寄存器 0	CSI	454
CB1CTL1	CSIB1 控制寄存器 1	CSI	457
CB1CTL2	CSIB1 控制寄存器 2	CSI	458
CB1RIC	中断控制寄存器	INTC	514
CB1RX	CSIB1 接收数据寄存器	CSI	453
CB1RXL	CSIB1 接收数据寄存器 L	CSI	453
CB1STR	CSIB1 状态寄存器	CSI	460
CB1TIC	中断控制寄存器	INTC	514
CB1TX	CSIB1 传输数据寄存器	CSI	453
CB1TXL	CSIB1 传输数据寄存器 L	CSI	453
CCLS	CPU 操作时钟状态寄存器	CG	153
CLM	时钟 监视器模式寄存器	CLM	465
CTBP	CALLT 基址指针	CPU	49
CTPC	CALLT 执行状态保存寄存器	CPU	48
CTPSW	CALLT 执行状态保存寄存器	CPU	48
DBPC	异常/调试陷阱状态保存寄存器	CPU	49
DBPSW	异常/调试陷阱状态保存寄存器	CPU	49
ECR	中断源寄存器	CPU	46
EIPC	中断状态保存寄存器	CPU	45
EIPSW	中断状态保存寄存器	CPU	45
FEPC	NMI 状态保存寄存器	CPU	46
FEPSW	NMI 状态保存寄存器	CPU	46
IMR0	中断屏蔽寄存器 0	INTC	516
IMR0H	中断屏蔽寄存器 0H	INTC	516
IMR0L	中断屏蔽寄存器 0L	INTC	516
IMR1	中断屏蔽寄存器 1	INTC	516
IMR1H	中断屏蔽寄存器 1H	INTC	516
IMR1L	中断屏蔽寄存器 1L	INTC	516
IMR2	中断屏蔽寄存器 2	INTC	516
IMR2H	中断屏蔽寄存器 2H	INTC	516
IMR2L	中断屏蔽寄存器 2L	INTC	516
INTF0	外部中断下降沿设定寄存器 0	INTC	527
INTF3L	外部中断下降沿设定寄存器 3L	INTC	528
INTF9H	外部中断下降沿设定寄存器 9H	INTC	529
INTR0	外部中断上升沿设定寄存器 0	INTC	527
INTR3L	外部中断上升沿设定寄存器 3L	INTC	528
INTR9H	外部中断上升沿设定寄存器 9H	INTC	529
ISPR	服务优先权寄存器	INTC	517
KRIC	中断控制寄存器	INTC	514
KRM	按键返回模式寄存器	KR	535
LOCKR	锁定寄存器	CG	156
LVIIC	中断控制寄存器	INTC	514
LVIM	低电压检测寄存器	LVI	572
LVIS	低电压检测等级选择寄存器	LVI	573



符号	名称	设备	页码
NFC	噪音消除控制寄存器	INTC	530
OCDM	片上调试模式寄存器	DCU	609
OSTS	振荡稳定时间选择寄存器	待机	540
P0	端口 0	端口	80
P00NFC	TIP00 引脚噪音消除控制寄存器	定时器	176
P01NFC	TIP01 引脚噪音消除控制寄存器	定时器	176
P10NFC	TIP10 引脚噪音消除控制寄存器	定时器	176
P11NFC	TIP11 引脚噪音消除控制寄存器	定时器	176
P20NFC	TIP20 引脚噪音消除控制寄存器	定时器	176
P21NFC	TIP21 引脚噪音消除控制寄存器	定时器	176
P3	端口 3	端口	84
P30NFC	TIP30 引脚噪音消除控制寄存器	定时器	176
P31NFC	TIP31 引脚噪音消除控制寄存器	定时器	176
P3H	端口 3H	端口	84
P3L	端口 3L	端口	84
P4	端口 4	端口	89
P5	端口 5	端口	92
P7H	端口 7H	端口	98
P7L	端口 7L	端口	98
P9	端口 9	端口	100
P9H	端口 9H	端口	100
P9L	端口 9L	端口	100
PC	程序计数器	CPU	43
PCC	处理器时钟控制寄存器	CG	149
PCLM	可编程时钟模式寄存器	CG	158
PCM	端口 CM	端口	108
PCS	端口 CS	端口	110
PCT	端口 CT	端口	112
PDL	端口 DL	端口	114
PDLH	端口 DLH	端口	114
PDLL	端口 DLL	端口	114
PEMU1	外围仿真寄存器 1	LVI	578
PFC0	端口功能控制寄存器 0	端口	82
PFC3L	端口功能控制寄存器 3L	端口	86
PFC5	端口功能控制寄存器 5	端口	94
PFC9	端口功能控制寄存器 9	端口	102
PFC9H	端口功能控制寄存器 9H	端口	102
PFC9L	端口功能控制寄存器 9L	端口	102
PFCE3L	端口功能控制扩展寄存器 3L	端口	86
PFCE5	端口功能控制扩展寄存器 5	端口	94
PFCE9	端口功能控制扩展寄存器 9	端口	103
PFCE9H	端口功能控制扩展寄存器 9H	端口	103
PFCE9L	端口功能控制扩展寄存器 9L	端口	103
PIC0	中断控制寄存器	INTC	514

(4/7)

符号	名称	设备	页码
PIC1	中断控制寄存器	INTC	514
PIC2	中断控制寄存器	INTC	514
PIC3	中断控制寄存器	INTC	514
PIC4	中断控制寄存器	INTC	514
PIC5	中断控制寄存器	INTC	514
PIC6	中断控制寄存器	INTC	514
PIC7	中断控制寄存器	INTC	514
PLLCTL	PLL 控制寄存器	CG	155
PLLS	PLL 锁定时间设定寄存器	CG	157
PM0	端口模式寄存器 0	端口	80
PM3	端口模式寄存器 3	端口	84
PM3H	端口模式寄存器 3H	端口	84
PM3L	端口模式寄存器 3L	端口	84
PM4	端口模式寄存器 4	端口	89
PM5	端口模式寄存器 5	端口	92
PM7H	端口模式寄存器 7H	端口	98
PM7L	端口模式寄存器 7L	端口	98
PM9	端口模式寄存器 9	端口	100
PM9H	端口模式寄存器 9H	端口	100
PM9L	端口模式寄存器 9L	端口	100
PMC0	端口模式控制寄存器 0	端口	81
PMC3L	端口模式控制寄存器 3L	端口	85
PMC4	端口模式控制寄存器 4	端口	90
PMC5	端口模式控制寄存器 5	端口	93
PMC9	端口模式控制寄存器 9	端口	101
PMC9H	端口模式控制寄存器 9H	端口	101
PMC9L	端口模式控制寄存器 9L	端口	101
PMCCM	端口模式控制寄存器 CM	端口	108
PMCM	端口模式寄存器 CM	端口	108
PMCS	端口模式寄存器 CS	端口	110
PMCT	端口模式寄存器 CT	端口	112
PMDL	端口模式寄存器 DL	端口	114
PMDLH	端口模式寄存器 DLH	端口	114
PMDLL	端口模式寄存器 DLL	端口	114
PRCMD	命令寄存器	CPU	70
PRSCM0	预分频比较寄存器 0	WT	374, 497
PRSM0	预分频模式寄存器 0	WT	374, 497
PSC	省电控制寄存器	待机	538
PSMR	省电模式寄存器	待机	539
PSW	程序状态字	CPU	47
PU0	上拉电阻选项寄存器 0	端口	82
PU3	上拉电阻选项寄存器 3	端口	87
PU3H	上拉电阻选项寄存器 3H	端口	87
PU3L	上拉电阻选项寄存器 3L	端口	87

符号	名称	设备	页码
PU4	上拉电阻选项寄存器 4	端口	90
PU5	上拉电阻选项寄存器 5	端口	96
PU9	上拉电阻选项寄存器 9	端口	106
PU9H	上拉电阻选项寄存器 9H	端口	106
PU9L	上拉电阻选项寄存器 9L	端口	106
Q00NFC	TIQ00 引脚噪音消除控制寄存器	定时器	276
Q01NFC	TIQ01 引脚噪音消除控制寄存器	定时器	276
Q02NFC	TIQ02 引脚噪音消除控制寄存器	定时器	276
Q03NFC	TIQ03 引脚噪音消除控制寄存器	定时器	276
r0 to r31	通用寄存器	CPU	43
RAMS	内部 RAM 数据状态寄存器	LVI	573
RCM	内部振荡模式寄存器	CG	153
RESF	复位源标志寄存器	复位	557
SELCNT0	选择器操作控制寄存器 0	定时器	253
SYS	系统状态寄存器	CPU	71
TM0CMP0	TMM0 比较寄存器 0	定时器	363
TM0CTL0	TMM0 控制寄存器 0	定时器	364
TM0EQIC0	中断控制寄存器	INTC	514
TP0CCIC0	中断控制寄存器	INTC	514
TP0CCIC1	中断控制寄存器	INTC	514
TP0CCR0	TMP0 捕获/比较寄存器 0	定时器	171
TP0CCR1	TMP0 捕获/比较寄存器 1	定时器	173
TP0CNT	TMP0 计数器读缓冲寄存器	定时器	175
TP0CTL0	TMP0 控制寄存器 0	定时器	164
TP0CTL1	TMP0 控制寄存器 1	定时器	165
TP0IOC0	TMP0 I/O 控制寄存器 0	定时器	167
TP0IOC1	TMP0 I/O 控制寄存器 1	定时器	168
TP0IOC2	TMP0 I/O 控制寄存器 2	定时器	169
TP0OPT0	TMP0 选项寄存器 0	定时器	170
TP0OVIC	中断控制寄存器	INTC	514
TP1CCIC0	中断控制寄存器	INTC	514
TP1CCIC1	中断控制寄存器	INTC	514
TP1CCR0	TMP1 捕获/比较寄存器 0	定时器	171
TP1CCR1	TMP1 捕获/比较寄存器 1	定时器	173
TP1CNT	TMP1 计数器读缓冲寄存器	定时器	175
TP1CTL0	TMP1 控制寄存器 0	定时器	164
TP1CTL1	TMP1 控制寄存器 1	定时器	165
TP1IOC0	TMP1 I/O 控制寄存器 0	定时器	167
TP1IOC1	TMP1 I/O 控制寄存器 1	定时器	168
TP1IOC2	TMP1 I/O 控制寄存器 2	定时器	169
TP1OPT0	TMP1 选项寄存器 0	定时器	170
TP1OVIC	中断控制寄存器	INTC	514
TP2CCIC0	中断控制寄存器	INTC	514
TP2CCIC1	中断控制寄存器	INTC	514

(6/7)

符号	名称	设备	页码
TP2CCR0	TMP2 捕获/比较寄存器 0	定时器	171
TP2CCR1	TMP2 捕获/比较寄存器 1	定时器	173
TP2CNT	TMP2 计数器读缓冲寄存器	定时器	175
TP2CTL0	TMP2 控制寄存器 0	定时器	164
TP2CTL1	TMP2 控制寄存器 1	定时器	165
TP2IOC0	TMP2 I/O 控制寄存器 0	定时器	167
TP2IOC1	TMP2 I/O 控制寄存器 1	定时器	168
TP2IOC2	TMP2 I/O 控制寄存器 2	定时器	169
TP2OPT0	TMP2 选项寄存器 0	定时器	170
TP2OVIC	中断控制寄存器	INTC	514
TP3CCIC0	中断控制寄存器	INTC	514
TP3CCIC1	中断控制寄存器	INTC	514
TP3CCR0	TMP3 捕获/比较寄存器 0	定时器	171
TP3CCR1	TMP3 捕获/比较寄存器 1	定时器	173
TP3CNT	TMP3 计数器读缓冲寄存器	定时器	175
TP3CTL0	TMP3 控制寄存器 0	定时器	164
TP3CTL1	TMP3 控制寄存器 1	定时器	165
TP3IOC0	TMP3 I/O 控制寄存器 0	定时器	167
TP3IOC1	TMP3 I/O 控制寄存器 1	定时器	168
TP3IOC2	TMP3 I/O 控制寄存器 2	定时器	169
TP3OPT0	TMP3 选项寄存器 0	定时器	170
TP3OVIC	中断控制寄存器	INTC	514
TQ0CCIC0	中断控制寄存器	INTC	514
TQ0CCIC1	中断控制寄存器	INTC	514
TQ0CCIC2	中断控制寄存器	INTC	514
TQ0CCIC3	中断控制寄存器	INTC	514
TQ0CCR0	TMQ0 捕获/比较寄存器 0	定时器	267
TQ0CCR1	TMQ0 捕获/比较寄存器 1	定时器	269
TQ0CCR2	TMQ0 捕获/比较寄存器 2	定时器	271
TQ0CCR3	TMQ0 捕获/比较寄存器 3	定时器	273
TQ0CNT	TMQ0 计数器读缓冲寄存器	定时器	275
TQ0CTL0	TMQ0 控制寄存器 0	定时器	260
TQ0CTL1	TMQ0 控制寄存器 1	定时器	261
TQ0IOC0	TMQ0 I/O 控制寄存器 0	定时器	263
TQ0IOC1	TMQ0 I/O 控制寄存器 1	定时器	264
TQ0IOC2	TMQ0 I/O 控制寄存器 2	定时器	265
TQ0OPT0	TMQ0 选项寄存器 0	定时器	266
TQ0OVIC	中断控制寄存器	INTC	514
UA0CTL0	UARTA0 控制寄存器 0	UART	421
UA0CTL1	UARTA0 控制寄存器 1	UART	443
UA0CTL2	UARTA0 控制寄存器 2	UART	444
UA0OPT0	UARTA0 选项控制寄存器 0	UART	423
UA0RIC	中断控制寄存器	INTC	514
UA0RX	UARTA0 接收数据寄存器	UART	426

(7/7)

符号	名称	设备	页码
UA0STR	UARTA0 状态寄存器	UART	424
UA0TIC	中断控制寄存器	INTC	514
UA0TX	UARTA0 传输数据寄存器	UART	426
UA1CTL0	UARTA1 控制寄存器 0	UART	421
UA1CTL1	UARTA1 控制寄存器 1	UART	443
UA1CTL2	UARTA1 控制寄存器 2	UART	444
UA1OPT0	UARTA1 选项控制寄存器 0	UART	423
UA1RIC	中断控制寄存器	INTC	514
UA1RX	UARTA1 接收数据寄存器	UART	426
UA1STR	UARTA1 状态寄存器	UART	424
UA1TIC	中断控制寄存器	INTC	514
UA1TX	UARTA1 传输数据寄存器	UART	426
VSWC	系统等待控制寄存器	CPU	72
WDTE	看门狗定时器使能寄存器	WDT	384
WDTM2	看门狗定时器模式寄存器 2	WDT	382, 518
WTIC	中断控制寄存器	INTC	514
WTIIC	中断控制寄存器	INTC	514
WTM	表钟定时器操作模式寄存器	WT	375

## 附录 C 指令集列表

### C.1 规则

#### (1) 用于描述操作数的寄存器符号

寄存器符号	说明
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: 主要用于存储除法运算结果的余数和乘法运算结果的高 32 位数据。
bit#3	用于指定位数的 3 位数据。
immX	X 位立即数
dispX	X 位偏移量数据
regID	系统寄存器号
vector	指定陷阱矢量的 5 位数据(00H 至 1FH)
cccc	显示条件代码的 4 位数据
sp	堆栈指针(r3)
ep	元素指针(r30)
listX	X 项寄存器列表

#### (2) 用于描述操作码的寄存器符号

寄存器符号	说明
R	指定 reg1 或 regID 的 1 位代码数据
r	指定 reg2 的 1 位代码数据
w	指定 reg3 的 1 位代码数据
d	1 位偏移量数据
l	1 位立即数 (指示立即数的高位)
i	1 位立即数
cccc	显示条件代码的 4 位数据
CCCC	显示 Bcond 指令的条件代码之 4 位数据
bbb	指定位数的 3 位数据
L	指定寄存器列表中的程序寄存器的 1 位数据

## (3) 用于描述操作的寄存器符号

寄存器符号	说明
←	数据输入
GR [ ]	通用寄存器
SR [ ]	系统寄存器
zero-extend (n)	用 n 个零扩展至字数据 (零扩展)。
sign-extend (n)	用 n 个符号扩展至字 (符号扩展)
load-memory (a, b)	从地址 a 读大小为 b 的数据
store-memory (a, b, c)	将大小为 c 的数据 b 写入地址 a 中
load-memory-bit (a, b)	读地址 a 的第 b 位数据。
store-memory-bit (a, b, c)	将数据 c 写入地址 a 的第 b 位
saturated (n)	对 n 执行饱和和处理 (n 为二进制补码). l 计算结果, 如果, $n \geq 7FFFFFFFH$ , 则将结果设置为 7FFFFFFFH. $n \leq 80000000H$ , 则将结果设置为 80000000H.
result	反映标志位的结果
Byte	字节(8 位)
Halfword	半字(16 位)
Word	字 (32 位)
+	加
-	减
ll	串位
×	乘
÷	除
%	除法运算结果的余数
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) 用于执行时钟寄存器符号

寄存器符号	说明
i	在第一条指令执行后, 如果就立即执行另外一条指令(结束)。
r	在第一条指令执行后, 如果就立即重复执行该指令(重复)。
l	在第一条指令执行后, 如果就立即使用该指令的执行结果(等待)。

## (5) 用于标志操作的寄存器符号

标识符	说明
(Blank)	无变化
0	清零
X	按结果设置或清零
R	预存值恢复

## (6) 条件代码

条件代码 (cccc)	条件公式	说明
0 0 0 0	$OV = 1$	溢出
1 0 0 0	$OV = 0$	没有溢出
0 0 0 1	$CY = 1$	有进位 低于 (小于)
1 0 0 1	$CY = 0$	无进位 不低于 (大于或等于)
0 0 1 0	$Z = 1$	结果为 0
1 0 1 0	$Z = 0$	结果不为 0
0 0 1 1	$(CY \text{ or } Z) = 1$	不高于 (小于或等于)
1 0 1 1	$(CY \text{ or } Z) = 0$	高于 (大于)
0 1 0 0	$S = 1$	结果为负数
1 1 0 0	$S = 0$	结果为正数
0 1 0 1	—	一直 (无条件的)
1 1 0 1	$SAT = 1$	饱和
0 1 1 0	$(S \text{ xor } OV) = 1$	小于有符号数
1 1 1 0	$(S \text{ xor } OV) = 0$	大于或等于有符号数
0 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 1$	小于或等于有符号数
1 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 0$	大于有符号数



## C.2 指令集(按字母先后顺序)

(1/6)

助记符	操作数	操作码	操作	执行时钟			标志位				
				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	dddd1011dddcccc 注 1	如果满足条件 则 PC←PC+sign-extend(disp9)	2	2	2					
			当满足条件时	注 2	注 2	注 2					
			当不满足条件时	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwwww01101000010	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 wwwwww01101000000	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 dddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)	3 注 3	3 注 3	3 注 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)	3 注 3	3 注 3	3 注 3				×	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiii wwwwww01100cccc0	if conditions are satisfied then GR[reg3]←sign-extended(imm5) else GR[reg3]←GR[reg2]	1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRRR wwwwww011001cccc0	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		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW	3	3	3	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW	3	3	3	R	R	R	R	R

(2/6)

助记符	操作数	操作码	操作	执行时钟			标志位				
				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		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5,list12	00000110011111L LLLLLLLLLLLL00000	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]	00000110011111L LLLLLLLLLLLLRRRRR 注 5	sp←sp+zero-extend(imm5 逻辑左移 2 位 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 重复以上 2 步,直到列表 12 所有寄存器都装载为止 PC←GR[reg1]	n+3 注 4	n+3 注 4	n+3 注 4					
DIV	reg1,reg2,reg3	rrrrr11111RRRRR wwwwww01011000000	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] <sup>注 6</sup>	35	35	35		×	×	×	
	reg1,reg2,reg3	rrrrr11111RRRRR wwwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1] <sup>注 6</sup> GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	rrrrr11111RRRRR wwwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1] <sup>注 6</sup> GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	rrrrr11111RRRRR wwwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	Stop	1	1	1					
HSW	reg2,reg3	rrrrr11111100000 wwwwww01101000100	GR[reg3]←GR[reg2](15:0)    GR[reg2] (31:16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110ddddd dddddddddddddd0 注 7	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110ddddd dddddddddddddd0 注 7	PC←PC+sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adrs,Byte))	1	1	注 11					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR dddddddddddddd1 注 8, 10	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adrs,Byte))	1	1	注 11					

(3/6)

助记符	操作数	操作码	操作	执行时钟			标志位				
				i	r	l	CY	OV	S	Z	SAT
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR dddddddddddddd0 注 8	adr←GR[reg1]+sign-extend(displ6) GR[reg2]←sign-extend(Load-memory(adrr,Halfword))	1	1	注 11					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 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(displ6) GR[reg2]←zero-extend(Load-memory(adrr,Halfword))	1	1	注 11					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR dddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(displ6) GR[reg2]←Load-memory(adrr,Word)	1	1	注 11					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]	1	1	1					
	imm5,reg2	rrrrr010000iiii	GR[reg2]←sign-extend(imm5)	1	1	1					
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii iiiiiiiiiiiiiiii	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]+(imm16ll0 <sup>16</sup> )	1	1	1					
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3]llGR[reg2]←GR[reg2]xGR[reg1] 注 14	1	4	5					
	imm9,reg2,reg3	rrrrr111111iiii wwwww01001111100 注 13	GR[reg3]llGR[reg2]←GR[reg2]xsign-extend(imm9)	1	4	5					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2] <sup>6</sup> xGR[reg1] <sup>6</sup>	1	1	2					
	imm5,reg2	rrrrr010111iiii	GR[reg2]←GR[reg2] <sup>6</sup> xsign-extend(imm5)	1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] <sup>6</sup> ximm16	1	1	2					
MULU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3]llGR[reg2]←GR[reg2]xGR[reg1] 注 14	1	4	5					
	imm9,reg2,reg3	rrrrr111111iiii wwwww0100111110 注 13	GR[reg3]llGR[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]	01bbb11110RRRRR dddddddddddddd	adr←GR[reg1]+sign-extend(displ6) Z flag←Not(Load-memory-bit(adrr,bit#3)) Store-memory-bit(adrr,bit#3,Z flag)	3 注 3	3 注 3	3 注 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adrr,reg2)) Store-memory-bit(adrr,reg2,Z flag)	3 注 3	3 注 3	3 注 3				×	

(4/6)

助记符	操作数	操作码	操作	执行时钟			标志位				
				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 iiiiiiiiiiiiiiii	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 <sup>注15</sup>	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17	n+2 注4 注17	n+2 注4 注17					
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	rrrrr111111RRRRR 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	rrrrr1111110cccc 0000001000000000	如果满足条件, 则 GR[reg2]←(GR[reg2]Logically shift left by 1) 或 00000001H 且 GR[reg2]←(GR[reg2]Logically shift left by 1) 或 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 iiiiiiiiiiiiiiii	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	rrrrr1111110cccc 0000000000000000	如果满足条件, 则 GR[reg2]←00000001H 且 GR[reg2]←00000000H	1	1	1					

助记符	操作数	操作码	操作	执行时钟			标志位				
				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] 逻辑左移 GR[reg1]位	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiii	GR[reg2]←GR[reg2] 逻辑左移 by zero-extend(imm5)位	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr11111RRRRR 0000000010000000	GR[reg2]←GR[reg2] 逻辑右移 GR[reg1]位	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiii	GR[reg2]←GR[reg2] 逻辑右移 by zero-extend(imm5)位	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	rrrrr0000110dddd 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	rrrrr0000111dddd 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					

(6/6)

助记符	操作数	操作码	操作	执行时钟			标志位				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]-GR[reg1]	1	1	1	x	x	x	x	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]-GR[reg2]	1	1	1	x	x	x	x	
SWITCH	reg1	0000000010RRRRR	adr←(PC+2) + (GR [reg1] 逻辑左移 1 位) PC←(PC+2) + (sign-extend (Load-memory (adr,Halfword)) 逻辑左移 1 位	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	vector	000001111111iiii 0000000100000000	EIPC ←PC+4 (Restored PC) EIPSW ←PSW ECR.EICC ←中断代码 PSW.EP ←1 PSW.ID ←1 PC ←00000040H (当中断矢量由 00H 变为 0FH 时) 00000050H (当中断矢量由 10H 变为 1FH 时)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	x	x	
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				x	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3 注3	3 注3	3 注3				x	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	x	x	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	x	x	
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 位数据。
  2. 3 如果先前有一指令立即重写了 PSW 寄存器的内容。.
  3. 如果没有等待状态(3 + 读访问等待状态数)。
  4. n 为列表 12 装载寄存器的总数 (根据等待状态数。 同样,如果没有等待状态, n 为列表 12 装载寄存器的总数,如果 n = 0,执行与 n = 1 情况一样的操作。)
  5. RRRRR: 00000 以外的数。
  6. 仅低半字数据有效。
  7. dddddddddddddddddddd: disp22 的高 21 位数据。
  8. dddddddddddddddd: disp16 的高 15 位数据。
  9. 取决于等待状态数(如果没有等待状态则为 1)。
  10. b: disp16 的第 0 位。
  11. 取决于等待状态数(如果没有等待状态则为 2)。

- 注 12. 在该指令中，出于助记符描述的方便起见，将 **reg2** 用作源寄存器，而寄存器 **reg1** 域用作操作码。因此，该指令的助记符描述和操作码中的寄存器配置的含义与其他指令不同。
- rrrrr =寄存器 **reg1D** 配置  
RRRRR =寄存器 **reg2** 配置
13. **iiii**: **imm9** 的低 5 位数据。  
**IIII**: **imm9** 的高 4 位数据。
14. 请不要将通用寄存器 **reg1** 和 **reg3** 指定为同一个寄存器。
15. **sp/imm**: 指定子操作码的 19 位和 20 位。
16. **ff = 00**: 将 **sp** 装载至 **ep** 中。  
01: 将符号扩展的 16 位立即数(第 47 到 32 位)装载至 **ep** 中。  
10: 将逻辑左移 16 位的 16 位立即数(第 47 到 32 位)装载至 **ep** 中。  
11: 将 32 位立即数(第 63 到 32 位)装载至 **ep** 中。
17. 如果 **imm = imm32**, **n + 3** 个时钟。
18. **rrrrr**: 00000 以外的数。
19. **ddddddd**: **disp8** 的高 7 位数据。
20. **dddd**: **disp5** 的高 4 位数据。
21. **dddddd**: **disp8** 的高 6 位数据。

本附录列出本文档中的注意事项。

表中“类别（硬件/软件）”定义如下。

硬件: 微控制器内部/外部硬件的注意事项

软件: 寄存器设置或编程的注意事项

(1/29)

章节	类别	功能	功能详解	注意事项	页码
第一章	硬件	引言	FLMD0	正常模式下, 将该引脚连接到 VSS 引脚端。	p. 20 <input type="checkbox"/>
			REGC	通过一个 4.7 $\mu$ F (推荐值) 电容将 REGC 引脚和 VSS 引脚连接在一起。	p. 20 <input type="checkbox"/>
第二章	软件	引脚 功能	NMI	NMI 引脚另一复用功能是 P02 引脚。复位后该引脚为引脚功能, 要启用 NMI 引脚, 请将 PMC0.PMC02 位设置为 1。NMI 引脚的初始化设置为“无边沿检测”, NMI 引脚的有效边沿要使用 INTF0 和 INTRO 寄存器来设置。	p. 28 <input type="checkbox"/>
			FLMD0	在自编程期间, 如果有超出噪声消除宽度的噪声输入至 RESET 端, 当有电容连接至 FLMD0 引脚时, 根据该电容的充电结束时间, 可能会启动闪存板模式。因此, 不要将电容连接到 FLMD0 引脚端。	p. 38 <input type="checkbox"/>
			当电源上电时	注意, 即使是在上电复位期间, 以下引脚可能会临时输出一个不确定的电平信号。 P53/KR3/TIQ00/TOQ00/DDO 引脚	p. 40 <input type="checkbox"/>
第三章	软件	CPU 功能	EIPC 寄存器 EIPSW 寄存器 FEPC 寄存器 FEPSW 寄存器	由于只有一组这样中断状态保存寄存器可用, 所以, 如果允许多重中断, 这些寄存器的内容必须在程序中通过编程加以保存。	p. 44 <input type="checkbox"/>
			EIPC, FEPC	即使 EIPC 或 FEPC 寄存器, 或 CTPC 寄存器的第 0 位由 LDSR 指令设置为 1 了, 在中断服务完成后, 程序执行由 RETI 指令返回到主程序时, 第 0 位仍然会被忽略(这是因为 PC 的第 0 位是固定为 0 的)。这样, 对寄存器 EIPC, FEPC 以及 CTPC 就设置了一个偶数值(第 0 位 = 0)。	p. 44 <input type="checkbox"/>
			程序存储空间	由于从地址 03FFF000H 到 03FFFFFFH 的 4 KB 地址区域为片上外围 I/O 区域, 所以不能从该地址区域取指令。因此, 不允许执行那些转移地址计算结果影响该地址区域的操作。	p. 52 <input type="checkbox"/>
			片上外围 I/O 区	当以一个字长为单位访问这类寄存器时, 则该字域要以半字为单位进行两次访问, 并且以先低字节后高字节的顺序进行, 同时该地址单元的低 2 位忽略不计。	p. 57 <input type="checkbox"/>
				对能够以字节为单位进行访问的寄存器, 如果以半字为单位进行访问, 那么, 在读寄存器时, 其高 8 位为不确定数据。写数据时, 将数据写入低 8 位。	p. 57 <input type="checkbox"/>
				那些没有定义的地址区域保留为将来扩展之用。如果对这些地址进行访问, 其操作没有定义而且也不保证操作的有效性。	p. 57 <input type="checkbox"/>
			内部 RAM 区	如果转移指令位于内部 RAM 区的上限地址处, 那么这一跨越到片上外围 I/O 区的预取操作(无效获取)是不会发生的。	p. 58 <input type="checkbox"/>



章节	类别	功能	功能详解	注意事项	页码
第三章	软件	CPU 功能	特殊功能寄存器数据的设置	当切换到IDLE1模式，IDLE2模式，STOP模式或sub-IDLE模式(通过设置PSC.STP位为1实现)时，在切换执行后，必需立即插入5条NOP指令。	p. 69 <input type="checkbox"/>
				当执行存储指令向命令寄存器存储数据的时候，中断会得不到应答。因为假定在以上的第<2>和<3>步执行的是连续的存储指令，如果在第<2>和<3>步之间有另外一条指令，并假设该指令对中断进行了应答，那么就无法建立上述顺序，从而导致故障出现。	p. 69 <input type="checkbox"/>
				虽然向PRCMD寄存器写入的是空数据，依然用在设置特殊寄存器时所用的那个通用寄存器(例中的第<3>步)来向PRCMD寄存器(例中的第<2>步)来写入数据。当通用寄存器用于寻址时，同样使用这种用法。	p. 69 <input type="checkbox"/>
			SYS 寄存器	如果将0写入SYS寄存器的PRERR位，但是该寄存器并不是特殊寄存器，那么，在对PRCMD寄存器进行写访问之后，PRERR位就立即清为0(写访问优先)。	p. 71 <input type="checkbox"/>
				如果将数据写入PRCMD寄存器，但是该寄存器并不是特殊寄存器，那么，在对PRCMD寄存器进行写访问之后，PRERR位就立即设置为1。	p. 71 <input type="checkbox"/>
			需要先行设置的寄存器	当使用V850ES/HF2微控制器时，必需确保首先对下列寄存器进行设置。 ●系统等待控制寄存器(VSWC) ●片上调试模式寄存器(OCDM) ●门狗定时器模式寄存器2(WDTM2)	p. 72 <input type="checkbox"/>
			VSWC 寄存器	访问片上外围I/O寄存器需要三个时钟周期(没有等待周期)。根据操作频率的不同，V850ES/HF2要求不同的等待周期。所以根据所使用的时钟频率，设置VSWC寄存器为以下各值。	p. 72 <input type="checkbox"/>
				访问特殊片上外围I/O寄存器	在下面两种情况下，禁止访问上述寄存器。如果产生了等待周期，该等待周期只有通过复位来清除。 ●CPU在副时钟下运行且主时钟停止振荡。 ●CPU在内部振荡时钟下运行
第四章	硬件	端口 功能	端口功能	尽管1-位内存操作指令处理1位，其按8-位单元访问端口。因此，如果端口包含输入和输出引脚，在输入模式下即使引脚未进行操作，也未定义输出锁存器引脚设置的内容。	p. 77 <input type="checkbox"/>
	软件		端口 0	NMI 引脚另外用作 P02 引脚。其复位后用作 P02 引脚。 要启用NMI引脚，将PMC0.PMC02位设置为1。NMI引脚的初始设置为“未检测到边缘”。使用INTF0和INTR0寄存器选择NMI引脚有效边缘。	p. 79 <input type="checkbox"/>
	硬件, 软件			P05 引脚的复用功能为片上调试功能。外部复位后，P05/INTP2/DRST 引脚初始化为片上调试引脚(DRST)。要将 P05 引脚用作端口引脚，而不用作片上调试引脚，必须采取以下动作。 <1> OCDM.OCDM0 位(专用寄存器)清零。 <2> 将 P05/INTP2/DRST 引脚固定为低电平直到采取以下动作。 未使用片上调试功能时，采取以上动作之前将高电平输入到 DRST 引脚可能发生故障(CPU死锁)。处理 P05 引脚时必须非常小心。 当未将高电平输入到 P05/INTP2/DRST 引脚(当此引脚固定为低电平时，无需操作 OCDM.OCDM0 位。 因为下拉电阻(30 kΩ TYP.)连接到缓冲器的 P05/INTP2/DRST 引脚，引脚无需通过外部电源固定为低电平。下拉电阻通过 OCDM0 位清零断开。	p. 79 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第四章	硬件	端口功能	端口 0	P00到P06引脚复用功能在输入模式下具有滞后特性，而在端口模式下不具有滞后特性。	p. 80 <input type="checkbox"/>
	软件		PMC0 寄存器	当OCDM.OCDM0位为1时P05/INTP2/DRST引脚用作DRST引脚，而不管PMC05位值如何。	p. 81 <input type="checkbox"/>
	硬件		端口 3	P31到P35引脚复用功能在输入模式下具有滞后特性，而在端口模式下不具有滞后特性。	p. 83 <input type="checkbox"/>
	软件		P3 寄存器	要按8-位或1-位单元读取或写入P3寄存器的位8到15，指定这些位为P3H寄存器的位0到7。	p. 84 <input type="checkbox"/>
	软件		PM3 寄存器	要按8-位或1-位单元读取或写入PM3寄存器的位8到15，指定这些位为PM3H寄存器的位0到7。	p. 84 <input type="checkbox"/>
			PMC3L 寄存器	INTP7引脚另外用作RXDA0引脚。要用作RXDA0引脚，使复用-功能INTP7引脚的边缘检测功能无效(通过INTF3.INTF31和INTR3.INTR31位固定为0)。要用作INTP7引脚，停止UARTA0的接收操作(通过UA0CTL0.UA0RXE位清零)。	p. 85 <input type="checkbox"/>
			PU3 寄存器	要按8-位或1-位单元读取/写入PU3寄存器的位8到15，指定这些位为PU3H寄存器的位0到7。	p. 87 <input type="checkbox"/>
			端口 4	P40和P42引脚复用功能在输入模式下具有滞后特性，而在端口模式下不具有滞后特性。	p. 88 <input type="checkbox"/>
	硬件, 软件		端口 5	DDI, DDO, DCK 和 DMS 引脚用于片上调试功能。要将 DDI, DDO, DCK 和 DMS 引脚用作端口引脚，而不用作片上调试引脚，必须采取以下动作。 <1> OCDM 寄存器(专用寄存器) OCDM0 位清零。 <2> 将 P05/INTP2/DRST 引脚固定为低电平直到采取以上动作。 当未使用片上调试功能时，采取以上动作之前将高电平输入到 DRST 引脚可能发生故障(CPU 死锁)。处理 P05 引脚需要非常小心。 当未将高电平输入到 P05/INTP2/DRST 引脚(当此引脚固定为低电平时，无需操作 OCDM.OCDM0 位。 因为下拉电阻(30 kΩ TYP.)连接到缓冲器的P05/INTP2/DRST引脚，引脚无需通过外部电源固定为低电平。下拉电阻通过OCDM0位清零断开。	p. 91 <input type="checkbox"/>
	硬件		端口 5 PMC5 寄存器	P50到P55引脚复用功能在输入模式下具有滞后特性，而在端口模式下不具有滞后特性。	p. 91 <input type="checkbox"/>
	软件			如果当PFC5.PFC5n和PFCE5.PFCE5n位为默认值(0)时控制模式使用PMC5寄存器指定，则未定义输出。出于该原因，首先设置PFC5.PFC5n和PFCE5.PFCE5n位且接着将PMC5n位设置为1以设置控制模式。	pp. 93, 94 <input type="checkbox"/>
	软件		端口5复用功能规范	KRn引脚另外用作TIQ0m引脚。要将此引脚用作TIQ0m引脚，使复用-功能KRn引脚的按键中断检测功能无效(通过KRM.KRMn位清零)。要将此引脚用作KRn引脚，使复用-功能TIQ0m引脚的边缘检测功能无效(n = 0到3, m = 0到3)。	p. 95 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第四章	软件	端口功能	P7H 寄存器, P7L 寄存器	A/D转换期间禁止读取P7H和P7L寄存器。	p. 98 <input type="checkbox"/>
			PM7H 寄存器, PM7L 寄存器	要使用P7n(ANIn)复用功能, 设置PM7n 到 1.	p. 98 <input type="checkbox"/>
	硬件	端口 9	P90, P91, P96, P97, P99和P913到P915引脚复用功能在输入模式下具有滞后特性, 而在端口模式下不具有滞后特性。	p. 99 <input type="checkbox"/>	
	软件	P9 寄存器	要读取或写入位8到15 of P9寄存器按8-位或1-位单元, 指定这些位为位0到7 of P9H寄存器。	p. 100 <input type="checkbox"/>	
		PM9 寄存器	要按8-位或1-位单元读取或写入PM9寄存器的位8到15, 指定这些位为PM9H寄存器的位0到7。	p. 100 <input type="checkbox"/>	
		PMC9 寄存器	如果当PFC9.PFC9n位和PFCE9.PFCE9n位为默认值(0)时控制模式使用PMC9寄存器指定, 则未定义输出。出于这一原因, 首先设置PFC9.PFC9n位和PFCE9.PFCE9n位为1且接着设置PMC9n位为1以设置控制模式。	p. 101 <input type="checkbox"/>	
			要按8-位或1-位单元读取或写入PMC9寄存器的位8到15, 指定这些位为PMC9H寄存器的位0到7。	p. 101 <input type="checkbox"/>	
		PFC9 寄存器	要按8-位或1-位单元读取或写入PFC9寄存器的位8到15, 指定这些位为PFC9H寄存器的位0到7。	p. 102 <input type="checkbox"/>	
		PFCE9 寄存器	要按8-位或1-位单元读取或写入PFCE9寄存器的位8到15, 指定这些位为PFCE9H寄存器的位0到7。	p. 103 <input type="checkbox"/>	
		P9引脚控制模式设置	如果当PFC9.PFC9n和PFCE9.PFCE9n位为默认值(0)时控制模式使用PMC9寄存器指定, 则未定义输出。出于这一原因, 首先设置PFC9.PFC9n和PFCE9.PFCE9n位且接着设置PMC9n位为1以设置控制模式。	p. 104 <input type="checkbox"/>	
			KR7引脚和RXDA1引脚为复用-功能引脚。 当将该引脚用作RXDA1引脚时, 禁用KR7引脚按键中断检测。(KRM7位KRM寄存器清零。) 同样, 当将该引脚用作KR7引脚, 建议设置PFC91位为1且PFCE91位清零。	p. 105 <input type="checkbox"/>	
		PU9 寄存器	要按8-位或1-位单元读取/写入PU9寄存器的位8到15, 指定这些位为PU9H寄存器的位0到7。	p. 106 <input type="checkbox"/>	
		PMCCM 寄存器	确保设置位7为2和0为“0”。	p. 108 <input type="checkbox"/>	
		端口 DL的功能	因为FLMD1引脚用于flash编程模式, 其无需使用端口控制寄存器操作。如需详细信息, 请参阅 <b>第二十二章 闪存</b> 。	p. 113 <input type="checkbox"/>	
		PDL 寄存器	要读取或写入位8到15 PDL寄存器按8-位或1-位单元, 指定这些位为位0到7 PDLH寄存器。	p. 114 <input type="checkbox"/>	
		PMDL 寄存器	要按8-位或1-位单元读取或写入PMDL寄存器的位8到15, 指定这些位为PMDLH寄存器的位0到7	p. 114 <input type="checkbox"/>	
	要将端口引脚用作复用-功能引脚的寄存器设置	外部复位后, P05/INTP2/DRST引脚初始化为片上调试引脚(DRST)。禁止将P05/INTP2/DRST引脚用作片上调试引脚, 请参阅 <b>第二十四章 片上调试功能</b> 。	p. 116 <input type="checkbox"/>		

章节	类别	功能	功能详解	注意事项	页码
第四章	软件	端口功能	要将端口引脚用作复用-功能引脚的寄存器设置	INTP7引脚另外用作RXDA0引脚。要将此引脚用作RXDA0引脚，使复用-功能INTP7引脚的边缘检测功能无效(通过INTF3.INTF31位和INTR3.INTR31位清零)。要将此引脚用作INTP7引脚，停止UARTA0的接收操作(通过UA0CTL0.UA0RXE位清零)。	p. 116 <input type="checkbox"/>
				KRn引脚另外用作TIQ0m引脚。要将此引脚用作TIQ0m引脚，使复用-功能KRn引脚的按键中断检测功能无效(通过KRM.KRMn位清零)。要将此引脚用作KRn引脚，使复用-功能TIQ0m引脚的边缘检测功能无效。	p. 117 <input type="checkbox"/>
				DDI, DDO, DCK和DMS引脚为片上调试引脚。外部复位后禁止将这些引脚用作片上调试引脚，请参阅 <b>第二十四章 片上调试功能</b> 。	p. 117 <input type="checkbox"/>
				如果当PFC5.PFC5n位和PFCE5.PFCE5n位为默认值(0)时控制模式指定使用PMC5寄存器，则未定义输出。 出于这一原因，首先设置PFC5.PFC5n位和PFCE5.PFCE5n位且接着设置PMC5n位为1以设置控制模式。	p. 117 <input type="checkbox"/>
				设置PM7n为1以使用P7n(ANIn)的复用功能。	p. 118 <input type="checkbox"/>
				KR7引脚和RXDA1引脚为复用-功能引脚。 当将该引脚用作RXDA1引脚时，禁用KR7引脚按键中断检测。(KRM.KRM7位清零。) 同样，当将该引脚用作KR7引脚时，建议设置PFC91位为1且PFCE91位清零。	p. 118 <input type="checkbox"/>
				如果当PFC9.PFC9n位和PFCE9.PFCE9n位为默认值(0)时控制模式指定使用PMC9寄存器，则未定义输出。 出于这一原因，首先设置PFC9.PFC9n位和PFCE9.PFCE9n位且接着设置PMC9n位为1以设置控制模式。	p. 118 <input type="checkbox"/>
				FLMD1引脚无需使用端口控制寄存器操作，因为其用于flash编程模式。如需详细信息，请参阅 <b>第二十二章 闪存</b> 。	p. 119 <input type="checkbox"/>
			从端口模式切换到复用-功能模式	按以下顺序从端口模式切换到复用-功能模式。 <1> 设置PFCn和PFCEn寄存器： 复用-功能选择 <2> 设置PMCn寄存器的对应位为1： 切换到复用-功能模式 如果首先设置PMCn寄存器，则应注意以下注意事项，在此时或根据PFCn和PFCEn寄存器设置而进行引脚状态的改变，可能出现不当操作。	p. 145 <input type="checkbox"/>
				无论端口模式/复用-功能模式，Pn寄存器如下所示进行读取和写入。 • Pn寄存器读取： 读取端口输出锁存器值(当PMn.PMnm位=0时)或读取引脚状态(PMn.PMnm位=1)。 • Pn寄存器写入： 写入端口输出锁存器	p. 145 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第四章	软件	端口功能	复用-功能模式(输入)的注意事项	<p>当PMCn.PMCnm位为0 时由于PMCn寄存器设置值的AND输出和引脚电平，复用功能块的输入信号为低电平。因此，根据端口设置和复用-功能操作启用时序，可能出现不当操作。因此，请按以下顺序在端口模式与复用-功能模式之间切换。</p> <ul style="list-style-type: none"> <li>从端口模式切换到复用-功能模式(输入)</li> </ul> <p>使用 PMCn 寄存器设置引脚为复用-功能模式且接着启用复用-功能操作。</p> <ul style="list-style-type: none"> <li>从复用-功能模式(输入)切换到端口模式</li> </ul> <p>停止复用-功能操作且接着将引脚切换到端口模式。</p>	p. 145 <input type="checkbox"/>
第五章	软件	时钟发生功能	时钟发生	在振荡稳定时间内当看门狗定时器2溢出时，选择内部振荡时钟。	p. 147 <input type="checkbox"/>
			PCC 寄存器	当CLKOUT正在输出时不要改变CPU时钟（利用CK3至CK0位）。	p. 150 <input type="checkbox"/>
				利用位操作指导对CK3位进行操作。在使用8位操作指令时，不要改变CK2位至CK0位设定的值。	p. 150 <input type="checkbox"/>
				在停止主时钟时，停止PLL。还应停止在主时钟下工作的片上外围功能的运行。	p. 151 <input type="checkbox"/>
				若下列条件不满足，应改变 CK2 至 CK0 位以使条件满足，再改换到副时钟工作模式。 内部系统时钟 (fCLK) > 副时钟 (fXT) × 4	p. 151 <input type="checkbox"/>
				只有在主时钟振荡稳定后才能启用在主时钟下工作的片上外围功能的运行。若在振荡稳定时间过去之前启用这些功能的运行，可能会导致故障。	p. 152 <input type="checkbox"/>
			RCM 寄存器	通过设置选项字节可使RCM寄存器的设置生效。详情请见“第二十三章选项字节功能”。	p. 153 <input type="checkbox"/>
				当CPU工作在内部振荡时钟下（CCLS.CCLSf位=1）时无法停止内部振荡器。不要将RSTOP位设置为1。	p. 153 <input type="checkbox"/>
				即便RSTOP位设置为1，只要CCLS.CCLSf位设置为1（在振荡稳定过程中出现WDT溢出时），内部振荡器就会振荡。此时，RSTOP位保持为1。	p. 153 <input type="checkbox"/>
			CCLS 寄存器	在复位解除后，若在振荡稳定过程中出现了WDT溢出，则CCLSf位被置为1且复位值为01H。	p. 153 <input type="checkbox"/>
			PLLCTL 寄存器	当PLLON位清为0时，SELPLL位自动清为0（clock-through模式）。	p. 155 <input type="checkbox"/>
				仅当PLL时钟频率稳定时，SELPLL位才可以设为1。若未稳定（未锁定），则在写入数据时对SELPLL位写入“0”。	p. 155 <input type="checkbox"/>
			LOCKR 寄存器	Lock寄存器不会实时反映PLL的锁定状态。设置/清零的条件如下。	p. 156 <input type="checkbox"/>
			PLLS 寄存器	通过设置使锁闭时间为800 s或以上。	p. 157 <input type="checkbox"/>
				在锁闭期内不要改变PLLS寄存器的设置。	p. 157 <input type="checkbox"/>
			PLCM 寄存器	首先设置端口相关控制寄存器（PM、PMC、PFC、PFCE 寄存器等等），再将PCLE位设置为1。	p. 158 <input type="checkbox"/>
				仅在PLL工作过程中才能设置PCLE位。要停止PLL，应将PCLE位清为0。	p. 158 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第六章	软件	16位定时器/事件计数器P (TMP)	TPnCTL0 寄存器	在TPnCE位=0时设定TPnCKS2至TPnCKS0位。 当TPnCE位的值从0变为1时，TPnCKS2至TPnCKS0位可以同时设定。	p. 164 <input type="checkbox"/>
				一定要将3至6位清为“0”。	p. 164 <input type="checkbox"/>
			TPnCTL1 寄存器	一定要将 TP0SYE 和TP2SYE为清为 0。	p. 165 <input type="checkbox"/>
				TPnEST 位仅在外触发脉冲输出模式或单次脉冲输出模式下有效。在其它模式下，忽略对该位写入1的动作。	p. 165 <input type="checkbox"/>
				一定要将3至4位清为“0”。	p. 165 <input type="checkbox"/>
				在外部事件计数模式下，外部事件计数输入被选中且与TPnEEE位的值无关。	p. 166 <input type="checkbox"/>
				应在TPnCTL0.TPnCE位=0时设置TPnEEE、TPnMD2至TPnMD0位。（TPnCE位=1时可写入相同的值。）当TPnCE位=1时，无法保证重写操作的进行。如果进行了错误的重写操作，应先将TPnCE位清为0再重新设置这些位。	p. 166 <input type="checkbox"/>
			TPnIOC0 寄存器	应在TPnCTL0.TPnCE位=0时重写TPnOL1、TPnOE1、TPnOL0、TPnOE0位。（TPnCE位=1时可写入相同的值。）如果进行了错误的重写操作，应先将TPnCE位清为0再重新设置这些位。	p. 167 <input type="checkbox"/>
				当TPnCE和TPnOEm位为0时，即便对TPnOLm位进行人为操作，TOPnm引脚输出电平也会变化。	p. 167 <input type="checkbox"/>
			TPnIOC1 寄存器	应在TPnCTL0.TPnCE位=0时重写TPnIS3至TPnIS0位。（TPnCE位=1时可写入相同的值。）如果进行了错误的重写操作，应先将TPnCE位清为0再重新设置这些位。	p. 168 <input type="checkbox"/>
				TPnis3至TPnIS0位仅在独立定时器模式和脉宽测量模式下有效。在所有其它模式下，无法进行捕捉操作。	p. 168 <input type="checkbox"/>
			TPnIOC2 寄存器	应在TPnCTL0.TPnCE位=0时重写TPnEES1、TPnEES0、TPnETS1、TPnETS0位。（TPnCE位=1时可写入相同的值。）如果进行了错误的重写操作，应先将TPnCE位清为0再重新设置这些位。	p. 169 <input type="checkbox"/>
				仅当TPnCTL1.TPnEEE位=1或已设置为外部事件计数模式（TPnCTL1.TPnMD2至TPnCTL1.TPnMD0位=001）时TPnEES1和TPnEES0位才有效。	p. 169 <input type="checkbox"/>
				仅当设置为外部触发脉冲输出模式（TPnCTL1.TPnMD2至TPnCTL1.TPnMD0位=010）或单次脉冲输出模式（TPnCTL1.TPnMD2至TPnCTL1.TPnMD0=011）时TPnETS1和TPnETS0位才有效。	p. 169 <input type="checkbox"/>
			TPnOPT0 寄存器	应在TPnCE位=0时重写TPnCCS1和TPnCCS0位。（TPnCE位=1时可写入相同的值。）如果进行了错误的重写操作，应先将TPnCE位清为0再重新设置这些位。	p. 170 <input type="checkbox"/>
				一定要将1至3位、6位、7位清为“0”。	p. 170 <input type="checkbox"/>
			TPnCCR0 寄存器	下列情况下禁止对TPnCCR0寄存器的访问。详情请见“3.4.8（2）访问特定片上外围I/O寄存器”。 •CPU在副时钟下工作，主时钟振荡停止 •CPU在内部振荡时钟下工作	p. 171 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第六章	软件	16位定时器/事件计数器P (TMP)	TPnCCR1 寄存器	<p>下列情况下禁止对TPnCCR1寄存器的访问。详情请见“3.4.8 (2) 访问特定片上外围I/O寄存器”。</p> <ul style="list-style-type: none"> <li>●CPU在副时钟下工作，主时钟振荡停止</li> <li>●CPU在内部振荡时钟下工作</li> </ul>	p. 173 <input type="checkbox"/>
			TPnCNT 寄存器	<p>下列情况下禁止对 TPnCNT 寄存器的访问。详情请见“3.4.8 (2) 访问特定片上外围 I/O 寄存器”。</p> <ul style="list-style-type: none"> <li>●CPU在副时钟下工作，主时钟振荡停止</li> <li>●CPU在内部振荡时钟下工作</li> </ul>	p. 175 <input type="checkbox"/>
			PnmNFC 寄存器	一定要将3至5位、7位清为“0”。	p. 176 <input type="checkbox"/>
				<p>在设定PnmNFC寄存器前输入定时器输入引脚 (TIPnm) 的信号输出时会消除数字噪声。</p> <p>因此，应通过PnmNFC寄存器设置采样时钟 (NFC2至NFC0) 和采样次数 (NFSTS)，等待初始化时间= (采样时钟) × (采样次数)，再启用定时器。</p>	p. 176 <input type="checkbox"/>
			操作	要使用外部事件计数模式，应指定不检测TIPn0引脚捕捉触发输入的有效边缘 (通过将TPnIOC1.TPnIS1和TPnIOC1.TPnIS0位清零为“00”)。	p. 177 <input type="checkbox"/>
				当使用外部触发脉冲输出模式、单次脉冲输出模式、脉宽测量模式时，应选择内部时钟作为计数时钟 (通过将TPnCTL1.TPnEEE位清为0)。	p. 177 <input type="checkbox"/>
			时间间隔定时器模式 (TPnMD2 至 TPnMD0 位=000)	仅当中断请求信号 (INTTPnCC0和INTTPnCC1) 被中断屏蔽标志 (TPnCCMK0和TPnCCMK1) 屏蔽且同时进行定时器输出 (TOPn1) 时，才能将该位设置为1。但是，应将TPnCCR0和TPnCCR1寄存器设置为相同的值 (见“6.5.1 (2) (d) TPnCCR1寄存器的操作”)。	p. 179 <input type="checkbox"/>
			关于重写TPnCCR0寄存器的注意点	<p>要把 TPnCCR0 寄存器的值变为一个较小的值，应停止计数一次，再改变设定值。</p> <p>若TPnCCR0寄存器的值在计数过程中重写为较小的值，则16位计数器可能会溢出。</p>	p. 184 <input type="checkbox"/>
			外部事件计数模式下操作的寄存器设置 (	当使用外部时钟作为计数时钟时，外部时钟只能从TIPn0引脚输入。此时，应将TPnIOC1.TPnIS1和TPnIOC1.TPnIS0位设置为00 (捕捉触发输入 (TIPn0引脚)：无边缘检测)。	p. 190 <input type="checkbox"/>
			外部事件计数模式下的操作定时	在外部事件计数模式下，不要将TPnCCR0寄存器设为0000H。	p. 192 <input type="checkbox"/>
				在外部事件计数模式下，无法使用定时器输出。如果利用外部事件计数输入进行定时器输出，应设置为时间间隔定时器模式，为计数时钟选择由外部事件计数输入启用的操作 (TPnCTL1.TPnMD2至TPnCTL1.TPnMD0位=000、TPnCTL1.TPnEEE位=1)。	p. 192 <input type="checkbox"/>
			重写TPnCCR0寄存器的注意点	<p>要把 TPnCCR0 寄存器的值变为一个较小的值，应停止计数一次，再改变设定值。</p> <p>若TPnCCR0寄存器的值在计数过程中重写为较小的值，则16位计数器可能会溢出。</p>	p. 193 <input type="checkbox"/>
			TPnIOC0, TPnCE0, TPnOL0 位	在外部触发脉冲输出模式下不使用TOPn0引脚时，应将此位清为0。	p. 198 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第六章	软件	16位定时器/事件计数器 P (TMP)	在工作过程中改变脉冲宽度的注意点	要在计数器工作时改变 PWM 波形, 应最后写入 TPnCCR1 寄存器。应在检测到INTTPnCC0信号之后写TPnCCR1寄存器, 之后重写TPnCCRm寄存器。	p. 202 <input type="checkbox"/>
			TPnIOC0, TPnOE0, TPnOL0位	在单次脉冲输出模式下不使用TOPn0引脚时, 应将此位清为0。	p. 210 <input type="checkbox"/>
			单次脉冲输出模式下寄存器的设定	如果TPnCCR1寄存器的设置值大于TPnCCR0寄存器的值, 则即使在单次脉冲输出模式下也不会输出单次脉冲。	p. 211 <input type="checkbox"/>
			关于重写TPnCCRm寄存器的注意点	要将 TPnCCRm 寄存器的设置值更改为较小的值, 应停止计数一次, 再改变设置值。 如果在计数过程中 TPnCCRm寄存器的值被重写为较小的值, 则16位计数器会溢出。	p. 213 <input type="checkbox"/>
			TPnIOC0, TPnOE0, TPnOL0 位	在PWM输出模式下不使用TOPn0引脚时应将该位清为0。	p. 217 <input type="checkbox"/>
			定时器调谐工作功能	调谐工作模式通过TPmCTL1.TPmSYE和TQ0CTL1.TQ0SYE位启用或停用。对于TMP2来说, 可指定TMP3或TMQ0或两者作为从定时器。	p. 248 <input type="checkbox"/>
				应按下下列步骤设置调谐工作模式。 <1> 设置从定时器的 TPmCTL1.TPmSYE 和 TQ0CTL1.TQ0SYE 位来启用调谐工作。 将从定时器的 TPmCTL1.TPmMD2 至 TPmCTL1.TPmMD0 和 TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD 位设置为独立模式。 <2> 通过 TPnCTL1.TPnMD2 至 TPnCTL1.TPnMD0 位设置定时器模式。此时, 不要设置主定时器的 TPnCTL1.TPnSYE 位。 <3> 设置主从定时器的比较寄存器。 <4> 设置从定时器的 TPmCTL0.TPmCE 和 TQ0CTL0.TQ0CE 位以启用内部工作时钟下的运行。 <5> 设置主定时器的TPnCTL0.TPnCE位以启用内部工作时钟下的运行。	p. 248 <input type="checkbox"/>
			选择器功能	在使用选择器功能时, 应在连接定时器前连接TMP的捕捉触发输入。	p. 252 <input type="checkbox"/>
				在设置选择器功能时, 应首先禁用要连接的外围I/O (TMP、TMM0或UARTA)。	p. 252 <input type="checkbox"/>
			SELCNT0 寄存器	在下列情况下应用 INTTM0EQ0 中断信号作为 TIP01 输入信号。 TMM0工作时钟≥TMP0工作时钟×4	p. 253 <input type="checkbox"/>
				要将ISEL02至ISEL04位设置为1, 应在捕捉输入模式下设置对应的引脚。	p. 253 <input type="checkbox"/>
				应在目标 (TMP0、TMP1、TMM0、UARTA0或UARTA1) 停止工作时设置ISEL02至ISEL06位。	p. 253 <input type="checkbox"/>
			捕捉操作	当使用捕捉操作且选择慢时钟作为计数时钟时, 如果在TPnCE位设置为1后立刻输入捕捉触发, 则在TPnCCR0和TPnCCR1寄存器中可能会捕捉到FFFFH而非0000H。	p. 254 <input type="checkbox"/>



章节	类别	功能	功能详解	注意事项	页码
第七章	软件	16位定时器/事件计数器Q (TMQ)	TQ0CTL0 寄存器	当TQ0CE位= 0时设置TQ0CKS2到TQ0CKS0位。 当TQ0CE位的值从0改变为1时, 可同时设置TQ0CKS2到TQ0CKS0位。	p. 260 <input type="checkbox"/>
				确保位3到6清零。	p. 260 <input type="checkbox"/>
			TQ0CTL1 寄存器	仅在外部触发脉冲输出模式或单触发脉冲输出模式下TQ0EST位有效。在任何其他模式下, 忽略此位写入1。	p. 261 <input type="checkbox"/>
				确保位3到4清零。	p. 261 <input type="checkbox"/>
				在外部事件计数模式下选择外部事件计数输入而不考虑TQ0EEE位的值。	p. 262 <input type="checkbox"/>
				当TQ0CTL0.TQ0CE位= 0时设置TQ0EEE和TQ0MD2到TQ0MD0位。(当TQ0CE位= 1时可写入相同值。) 当TQ0CE位= 1时进行重写不保证此操作。如果重写错误, 则TQ0CE位清零且接着再次设置这些位。	p. 262 <input type="checkbox"/>
			TQ0IOC0 寄存器	当TQ0CTL0.TQ0CE位= 0时重写TQ0OLm和TQ0OEm位。(当TQ0CE位= 1时可写入相同值。)如果重写错误, 则TQ0CE位清零且接着再次设置这些位。	p. 263 <input type="checkbox"/>
				即使当TQ0CE和TQ0OEm位为0时操作TQ0OLm位, TQ0Om引脚输出电平也发生变化。	p. 263 <input type="checkbox"/>
			TQ0IOC1 寄存器	当TQ0CTL0.TQ0CE位= 0时重写TQ0IS7到TQ0IS0位。(当TQ0CE位= 1时可写入相同值。)如果重写错误, 则TQ0CE位清零且接着再次设置这些位。	p. 264 <input type="checkbox"/>
				仅在自由定时器模式和脉宽测量模式下TQ0IS7到TQ0IS0位有效。在所有其他模式下, 捕获操作不可用。	p. 264 <input type="checkbox"/>
			TQ0IOC2 寄存器	当TQ0CTL0.TQ0CE位= 0时重写TQ0EES1, TQ0EES0, TQ0ETS1及TQ0ETS0位。(当TQ0CE位= 1时可写入相同值。)如果重写错误, 则TQ0CE位清零且接着再次设置这些位。	p. 265 <input type="checkbox"/>
				仅当TQ0CTL1.TQ0EEE位= 1时或当已设置外部事件计数模式(TQ0CTL1.TQ0MD2到TQ0CTL1.TQ0MD0位 = 001)时TQ0EES1和TQ0EES0位有效。	p. 265 <input type="checkbox"/>
				仅当设置外部触发脉冲输出模式(TQ0CTL1.TQ0MD2到TQ0CTL1.TQ0MD0位 = 010)或单触发脉冲输出模式(TQ0CTL1.TQ0MD2到TQ0CTL1.TQ0MD0 = 011)时TQ0ETS1和TQ0ETS0位有效。	p. 265 <input type="checkbox"/>
			TQ0OPT0 寄存器	当TQ0CTL0.TQ0CE位= 0时重写TQ0CCS3到TQ0CCS0位。(当TQ0CE位= 1时可写入相同值。)如果重写错误, TQ0CE位清零且接着再次设置这些位。	p. 266 <input type="checkbox"/>
				确保位1到3清零。	p. 266 <input type="checkbox"/>
			TQ0CCR0 寄存器	在以下状态禁止访问TQ0CCR0寄存器。如需详细信息, 请参阅 3.4.8(2)访问专用片上外围 I/O 寄存器。 • 当按子时钟CPU运行且主时钟振荡停止时 • 当CPU按内部振荡时钟运行时	p. 267 <input type="checkbox"/>
			TQ0CCR1 寄存器	在以下状态禁止访问TQ0CCR1寄存器。如需详细信息, 请参阅 3.4.8(2)访问专用片上外围 I/O 寄存器。 • 当按子时钟 CPU 运行且主时钟振荡停止时 • 当CPU按内部振荡时钟运行时	p. 269 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第七章	软件	16位定时器/事件计数器 Q (TMQ)	TQ0CCR2 寄存器	在以下状态禁止访问 TQ0CCR2 寄存器。如需详细信息，请参阅 3.4.8(2)访问专用片上外围 I/O 寄存器。 • 当按副时钟 CPU 运行且主时钟振荡停止时 • 当CPU按内部振荡时钟运行时	p. 271 <input type="checkbox"/>
			TQ0CCR3 寄存器	在以下状态禁止访问 TQ0CCR3 寄存器。如需详细信息，请参阅 3.4.8(2)访问专用片上外围 I/O 寄存器。 • 当按副时钟 CPU 运行且主时钟振荡停止时 • 当CPU按内部振荡时钟运行时	p. 273 <input type="checkbox"/>
			TQ0CNT 寄存器	在以下状态禁止访问 TQ0CNT 寄存器。如需详细信息，请参阅 3.4.8(2)访问专用片上外围 I/O 寄存器。 • 当按副时钟 CPU 运行且主时钟振荡停止时 • 当CPU按内部振荡时钟运行时	p. 275 <input type="checkbox"/>
			Q0mNFC 寄存器	确保位3到5和7清零。	p. 276 <input type="checkbox"/>
				输出设置Q0mNFC寄存器之前输入到定时器输入引脚(TIQ0m)的信号且消除数字噪音。因此，使用Q0mNFC寄存器设置采样时钟(NFC2到NFC0)和采样次数(NFSTS)，等待初始化时间=(采样时钟) × (采样次数)，且启动定时器运行。	p. 276 <input type="checkbox"/>
			外部事件计数模式	要使用外部事件计数模式，指定未检测到TIQ00引脚捕获触发输入的有效沿(通过TQ0IOC1.TQ0IS1且TQ0IOC1.TQ0IS0位清零)。	p. 277 <input type="checkbox"/>
			外部触发脉冲输出模式，单触发脉冲输出模式以及脉宽测量模式	当使用外部触发脉冲输出模式，单触发脉冲输出模式以及脉宽测量模式时，选择内部时钟作为计数时钟(通过TQ0CTL1.TQ0EEE位清零)。	p. 277 <input type="checkbox"/>
			TQ0CTL1.TQ0EEE 位	仅当中断请求信号(INTTQ0CC0和INTTQ0CCK)由中断屏蔽标记(TQ0CCMK0到TQ0CCMKk)屏蔽且同时进行定时器输出(TQ0Qk)时此位可设为1。然而，TQ0CCR0和TQ0CCRk寄存器必须设为相同值(请参阅 7.5.1(2)(d) TQ0CCR1到TQ0CCR3 寄存器)的操作(k = 1到3)。	p. 279 <input type="checkbox"/>
			重写TQ0CCR0寄存器的注意事项	要将 TQ0CCR0 寄存器的值改为更小的值，则停止计数一次且接着改变设置值。 如果计数期间TQ0CCR0寄存器的值重写为更小的值，则16-位计数器可能溢出。	pp. 283, 292 <input type="checkbox"/>
			在外部事件计数模式下操作的寄存器设置	当外部时钟用作计数时钟时，外部时钟仅可从TIQ00引脚输入。此时，设置TQ0IOC1.TQ0IS1和TQ0IOC1.TQ0IS0位为 00(捕获触发输入(TIQ00引脚)：未检测到边沿)。	p. 289 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第七章	软件	16位定时器/事件计数器Q (TMQ)	在外部事件计数模式下的操作时序	在外部事件计数模式下，TQ0CCR0寄存器禁止设为0000H。	p. 291 <input type="checkbox"/>
				在外部事件计数模式下，禁止使用定时器输出。如果使用外部事件计数输入进行定时器输出，设置计时器模式，且选择通过计数时钟的外部事件计数输入启用的操作 (TQ0CTL1.TQ0MD2到TQ0CTL1.TQ0MD0位 = 000，TQ0CTL1.TQ0EEE位 = 1)。	p. 291 <input type="checkbox"/>
			TQ0IOC0.TQ0OE0, TQ0OL0 位	当TQ000引脚未用于外部触发脉冲输出模式时此位清零。	p. 299 <input type="checkbox"/>
			运行期间改变脉宽的注意事项	当计数器运行时要改变PWM波形，最后请写入TQ0CCR1寄存器。 在检测到INTTQ0CC0信号之后写入TQ0CCR1寄存器之后重写TQ0CCRk寄存器。	p. 303 <input type="checkbox"/>
			TQ0IOC0.TQ0OE0, TQ0OL0 位	当TQ000引脚未用于单触发脉冲输出模式时此位清零。	p. 312 <input type="checkbox"/>
			寄存器设置在单触发脉冲输出模式下	如果TQ0CCRk寄存器的设置值大于TQ0CCR0寄存器的值，即使在单触发脉冲输出模式下也不输出单触发脉冲。	p. 313 <input type="checkbox"/>
			重写TQ0CCRm寄存器的注意事项	要 TQ0CCRm 寄存器的设置值改为更小的值，停止计数一次，且接着改变设置值。 如果计数期间TQ0CCR0寄存器的值重写为更小的值，则16-位计数器可能溢出。	p. 316 <input type="checkbox"/>
			TQ0IOC0.TQ0OE0, TQ0OL0 位	当TQ000引脚未用于PWM输出模式时此位清零。	p. 321 <input type="checkbox"/>
			PWM模式 (TQ0MD2到TQ0MD0 = 111) 的三角波	在PWM模式下，TQ0CCRm寄存器的捕获功能不能使用，因为此寄存器仅可用作比较寄存器。	p. 355 <input type="checkbox"/>
			定时器调整操作模式	通过TPmCTL1.TPmSYE和TQ0CTL1.TQ0SYE位启用或禁用调试操作模式。对于TMQ2，TMQ3和TMQ0的任一者或两者可指定为从属。 使用以下程序设置调试操作模式。 <1> 设置从定时器的 TPmCTL1.TPmSYE 和 TQ0CTL1.TQ0SYE 位以启用调试运行。 从定时器的TPmCTL1.TPmMD2到TPmCTL1.TPmMD0和TQ0CTL1.TQ0MD2到TQ0CTL1.TQ0MD0位设为自由模式。 <2> 使用 TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0 位设置定时器模式。 此时，禁止设置主定时器的 TPnCTL1.TPnSYE 位。 <3> 设置主、从定时器的比较寄存器值。 <4> 设置从定时器的 TPmCTL0.TPmCE 和 TQ0CTL0.TQ0CE 位以启用内部运行 时钟操作。 <5> 设置主定时器的TPnCTL0.TPnCE位以启用内部运行时钟操作。	p. 357 <input type="checkbox"/> p. 357 <input type="checkbox"/>
			捕获操作	当使用捕获操作且较慢时钟选作计数时钟时，如果TQ0CE位设为1之后立即输入捕获触发，则在TQ0CCR0，TQ0CCR1，TQ0CCR2及TQ0CCR3寄存器可捕获FFFFH而非0000H。	p. 361 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第八章	软件	16-位间隔定时器M (TMM)	TM0CTL0 寄存器	当 TM0CE 位= 0 时设置 TM0CKS2 到 TM0CKS0 位。 当值TM0CE 从0改为1时, 不能同时设置TM0CKS2到TM0CKS0位的值。 确保位 3到6清零。	p. 364 <input type="checkbox"/> p. 364 <input type="checkbox"/>
			间隔计时器模式	禁止将TM0CMP0寄存器设为 FFFFH.	pp. 365, 368 <input type="checkbox"/>
			开始计数	TM0CTL0.TM0CE位设为 1之后, 根据计数时钟选择, 需花费 16-位计数器以下时间开始计数。	p. 369 <input type="checkbox"/>
			TM0CMP0, TM0CTL0 寄存器s	当 TMM0 运行时禁止重写 TM0CMP0 和 TM0CTL0 寄存器。 如果当 TM0CE 位为 1 时重写这些寄存器, 则不能保证操作。 如果其误重写, 则 TM0CTL0.TM0CE 位清零, 且寄存器复位。	p. 369 <input type="checkbox"/>
第九章	软件	钟表定时器功能	PRSM0 寄存器	钟表定时器操作期间禁止改变BGCS00和BGCS01位的值。 BGCE0位设为 1之前设置PRSM0寄存器。 根据使用的主时钟频率设置 PRSM0和PRSCM0 寄存器以便获得32.768 kHz fBRG 频率。	p. 373 <input type="checkbox"/> p. 373 <input type="checkbox"/> p. 373 <input type="checkbox"/>
			PRSCM0 寄存器	钟表定时器操作期间禁止重写 PRSCM0寄存器。 PRSM0.BGCE0位设为 1之前设置PRSCM0寄存器。 根据使用的主时钟频率设置 PRSM0和PRSCM0 寄存器以便获得32.768 kHz fBRG 频率。	p. 374 <input type="checkbox"/> p. 374 <input type="checkbox"/> p. 374 <input type="checkbox"/>
			WTM 寄存器	当WTM0和WTM1位均为0时重写WTM2到WTM7位。	p. 376 <input type="checkbox"/>
			注意事项	操作启用之后第一个钟表定时器中断请求信号(INTWT)产生之前需要一些时间 (WTM.WTM1和WTM.WTM0位 = 1)。 第一个INTWT 信号产生(比 $2^9 \times 1/32768 = 0.015625$ 秒更长(max.)) 花费0.515625 秒(max.)。接着每0.5 秒产生INTWT信号。	p. 379 <input type="checkbox"/> p. 379 <input type="checkbox"/>
	硬件				
第十章	软件	看门狗定时器2的功能	功能	看门狗定时器2在复位模式下自动开始以下复位释放。 当不使用看门狗定时器2时, 进行复位之前经此功能或看门狗定时器2清零一次停止其操且在下一间隔时间内停止。 同样, 即使默认设置(复位模式, 间隔时间: $f_{\text{W}}/2^{19}$ )无需改变, 出于验证目的仅写入WDTM2寄存器一次。 对于由于不可屏蔽中断请求信号(INTWDT2)而产生的不可屏蔽中断服务, 请参阅 14.2.2(2) 来自INTWDT2信号。	p. 380 <input type="checkbox"/> p. 380 <input type="checkbox"/>
			WDTM2 寄存器	在以下状态禁止访问 WDTM2 寄存器。如需详细信息, 请参阅 3.4.8(2)访问专用片上外围 I/O 寄存器。 •当 CPU 按子时钟运行且主时钟振荡停止时 • 当CPU按内部振荡时钟运行时 如果OPB1位使用选项字节功能设为1 (请参阅 第二十三章), 复位模式固定。 如需WDCS20到WDCS24位的详细信息, 请参阅 表10-2 看门狗定时器2时钟选择。	p. 382 <input type="checkbox"/> p. 382 <input type="checkbox"/> p. 382 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第十章	软件	看门狗定时器2的功能	WDTM2 寄存器	如果复位后WDTM2寄存器重写两次，强制产生溢出信号且计数器复位。	p. 382 <input type="checkbox"/>
				要有意产生溢出信号，请仅写入WDTM2寄存器两次或将ACH之外的值写入WDTE寄存器一次。	p. 382 <input type="checkbox"/>
				<R> 然而，当看门狗定时器2设为停止运行时，即使数据仅写入WDTM2寄存器两次或“ACH”之外的值仅写入WDTE寄存器一次，也不产生溢出信号。	
				要停止看门狗定时器2运行，请将RCM.R停止位设为1(内部振荡器停止)，且将1FH写入WDTM2寄存器。然而，如果OPB1位使用选项字节功能设为1(请参阅第23章)，则看门狗定时器2不能通过复位以外的任何方式停止。	p. 382 <input type="checkbox"/>
				如果OPB1位使用选项字节功能设为1，则时钟固定为内部振荡时钟( $f_R$ )(可选择 $2^{12}/f_R$ 到 $2^{19}/f_R$ )。如需详细信息，请参阅第23章 选项字节功能。	p. 383 <input type="checkbox"/>
			WDTE 寄存器	当“ACH”以外的值写入WDTE寄存器时，强制输出溢出信号。	p. 384 <input type="checkbox"/>
				当执行WDTE寄存器的1-位内存操作指令时，强制输出溢出信号。	p. 384 <input type="checkbox"/>
				要有意产生溢出信号，请仅写入WDTM2寄存器两次或将ACH之外的值写入WDTE寄存器一次。	p. 384 <input type="checkbox"/>
				<R> 然而，当看门狗定时器2设为停止运行时，即使数据仅写入WDTM2寄存器两次或“ACH”之外的值仅写入WDTE寄存器一次，也不产生溢出信号。	
				WDTE寄存器的读取值为“9AH”(其不同与写入值“ACH”)。	p. 384 <input type="checkbox"/>
第十一章	硬件	A/D转换器	ANIO到ANI11引脚	确保输入到ANIO到ANI11引脚的电压不超过额定值。尤其如果通道输入VREF0电压或更高电压，则未定义通道转换值，且还可能影响其他通道的转换值。	p. 389 <input type="checkbox"/>
				模拟输入引脚(ANIO到ANI11)另外用作输入端口引脚(P70到P711)。如果选择ANIO到ANI11的任一者执行A/D转换，则转换期间禁止执行到端口7的输入指令。如果执行，则转换分辨率可能降级。	p. 389 <input type="checkbox"/>
	软件		ADA0M0 寄存器	在以下状态禁止访问ADA0M0寄存器。如需详细信息，请参阅 3.4.8(2) 访问专用片上外围 I/O寄存器。	p. 390 <input type="checkbox"/>
				<ul style="list-style-type: none"> <li>当 CPU 按副时钟运行且主时钟振荡停止时</li> <li>当CPU按内部振荡时钟运行时</li> </ul>	
				忽略写入位 0 的操作。	p. 391 <input type="checkbox"/>
				当A/D转换启用时(ADA0CE位= 1)禁止改变ADA0M1寄存器的值。	p. 391 <input type="checkbox"/>
				如果 A/D 转换期间写入 ADA0M0, ADA0M2, ADA0S, ADA0PFM 及 ADA0PFT 寄存器(ADA0EF位= 1)，则按照模式执行以下操作。	p. 391 <input type="checkbox"/>
				<ul style="list-style-type: none"> <li>在软件触发模式下</li> </ul> A/D 转换停止且从开始处再次开始。 <ul style="list-style-type: none"> <li>在硬件触发模式下</li> </ul> A/D转换停止，且设置触发备用状态。	

章节	类别	功能	功能详解	注意事项	页码
第十一章	软件	A/D 转换器	ADA0M0 寄存器	当不使用A/D转换器时，通过设置ADA0CE位停止操作以减少耗电。	p. 391 <input type="checkbox"/>
				A/D转换开始时的输入引脚数据的第一次转换的分辨率可能降级。如需详细信息，请参阅 11.6(7) AVREF0引脚。	p. 391 <input type="checkbox"/>
			ADA0M1 寄存器	确保位 6到4清零。	p. 392 <input type="checkbox"/>
				确保ADA0HS1位设为“1”	p. 392 <input type="checkbox"/>
			转换模式设置示例	设为 $3.1\mu s \leq \text{转换时间} \leq 15.5\mu s$ 。	p. 392 <input type="checkbox"/>
				稳定时间期间禁止ADA0M0, ADA0M2, ADA0S, ADA0PFM以及ADA0PFT寄存器的重写和触发输入。	p. 392 <input type="checkbox"/>
			ADA0M2 寄存器	确保位 7到2清零。	p. 393 <input type="checkbox"/>
			ADA0CRn, ADA0CRnH 寄存器	在以下状态禁止访问 ADA0CRn 和 ADA0CRnH 寄存器。如需详细信息，请参阅 3.4.8(2) 访问专用片上外围 I/O 寄存器。 ●当 CPU 按副时钟运行且主时钟振荡停止时 ●当CPU按内部振荡时钟运行时	p. 395 <input type="checkbox"/>
				写入ADA0M0和ADA0S寄存器的操作可能导致ADA0CRn寄存器的内容未定义。转换之后，写入ADA0M0和ADA0S寄存器之前读取转换结果。如果使用以上系列之外的系列则可能不读取校正转换结果。	p. 395 <input type="checkbox"/>
			ADA0PFM 寄存器	在选择模式下，将设为ADA0PFT寄存器的8-位数据与由ADA0S寄存器指定的ADA0CRnH寄存器的值比较。如果结果匹配ADA0PFC位指定的条件，则转换结果存储在ADA0CRn寄存器且INTAD信号产生。然而，如果不匹配，则不产生中断信号。	p. 397 <input type="checkbox"/>
				在扫描模式下，将设为ADA0PFT寄存器的8-位数据与ADA0CR0H寄存器的内容比较。如果结果匹配由ADA0PFC位指定的条件，则转换结果存储在ADA0CR0寄存器且产生INTAD信号。然而，如果不匹配，则不产生INTAD信号。而不考虑比较结果，扫描操作继续进行且转换结果存储在ADA0CRn寄存器直到扫描操作完成。然而，扫描操作完成之后不产生INTAD信号。	p. 397 <input type="checkbox"/>
			当未使用A/D转换器时	当未使用A/D转换器时，通过ADA0M0.ADA0CE位清零可减少耗电。	p. 410 <input type="checkbox"/>
			ANIO到ANI11引脚的输入范围	输入在ANIO到ANI11引脚指定范围内的电压。如果输入到任何引脚的电压等于或高于AVREF0或等于或低于AVSS(即使在绝对最大值范围内)，则未定义通道转换值，且还可能影响其他通道的转换值。	p. 410 <input type="checkbox"/>
			噪音对策	要保持10-位分辨率，必须有效保护ANIO到ANI11引脚免受噪音干扰。噪音影响随输出模拟输入源的电阻变高而增加。要降低噪音，建议如图11-9所示连接外部电容。	p. 410 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第十一章	软件	A/D 转换器	备用 I/O	模拟输入引脚(ANIO 到 ANI11)另外用作端口引脚。当选择 ANIO 到 ANI11 引脚之一进行 A/D 转换时, 转换期间禁止执行读取输入端口或写入输出端口的指令因为转换分辨率可能降低。  同样, 如果由于连接到端口引脚的外部电路的影响电流流动, 则 A/D 转换期间转换分辨率可能在设为输出端口引脚的引脚处降落。  如果数字脉冲施加到靠近转换输入信号引脚的引脚, 则由于耦合噪音的影响, A/D 转换值不能作为期望值。因此, 禁止将脉冲施加到靠近进行 A/D 转换引脚的引脚。	p. 410
			中断请求标记 (ADIF)	即使 ADA0S寄存器的内容改变中断请求标记(ADIF)也不清零。因此, 如果A/D转换期间模拟输入引脚改变, 则可存储先前选定模拟输入信号的转换结果且ADA0S寄存器重写之前可立即设置转换结束中断请求标记。如果ADA0S寄存器重写之后立即读取ADIF标记, 则即使不完成新选择模拟输入引脚的A/D转换也可设置ADIF标记。当A/D转换停止时, 恢复转换之前ADIF标记清零。	p. 411
			AVREF0引脚	(a) AVREF0引脚用作A/D转换器的电源引脚且同样向备用-功能端口供电。在其中使用备用电源的应用中, 确保供应与VDD到AVREF0引脚相同的电压, 如图11-12所示。  (b) AVREF0 引脚还用作 A/D 转换器的参考电压引脚。如果向 AVREF0 引脚供电的源具有高阻抗或如果电源具有低 电流供应能力, 参考电压可能发生波动, 这是由于转换期间电流流动 (尤其, 启用转换操作之后位 ADA0CE 即设为 1)。结果, 转换精度可能下降。要避免此情况, 建议将电容跨 AVREF0 和 AVSS 引脚连接以抑制参考电压波动, 如图 11-12 所示。  (c) 如果向AVREF0引脚供电的源具有高DC阻抗(例如, 因为嵌入二极管), 启用转换时的电压可能低于转换停止时的电压, 因为电压降落由A/D转换电流所导致。	p. 412
	软件	读取ADA0CRn结果	当写入ADA0M0到ADA0M2, ADA0S, ADA0PFM或ADA0PFT寄存器时, 可能未定义ADA0CRn寄存器的内容。完成转换之后和写入ADA0M0到ADA0M2, ADA0S, ADA0PFM或ADA0PFT寄存器之前读取转换结果。同样, 当确认外部/定时器触发时, 可能未定义ADA0CRn寄存器的内容。完成转换之后和确认下一外部/定时器触发之前读取转换结果。以上时序之外可能读取不到校正转换结果。	p. 412	
		A/D 转换结果	如果模拟输入引脚及参考电压输入引脚存在噪音, 则该噪音可能产生无效的转换结果。需要软件处理以避免此无效的转换结果对系统的负面影响。此软件处理示例如下所示。 <ul style="list-style-type: none"><li>●利用大量 A/D 转换并使其作为 A/D 转换结果。</li><li>●连续执行 A/D 转换数次且使用这些结果, 删除可能获得的任何例外的结果。</li><li>●如果获得判断产生系统故障的A/D转换结果, 则进行故障处理之前确保重复核对系统故障。</li></ul>	p. 412	

章节	类别	功能	功能详解	注意事项	页码
第十一章	软件	A/D 转换器		<R>	
			备用模式	因为 A/D 转换器在停止模式下停止操作，所以转换结果无效，因此可减少耗电。操作恢复之后释放停止模式，而释放停止模式之后的 A/D 转换结果无效。当释放停止模式之后使 A/D 转换器，设置 TOP 模式或释放停止模式之前，释放停止模式之后 ADA0M0.ADA0CE 位清零接着 ADA0CE 位设为 1。 在 IDLE1，IDLE2 或副时钟操作模式下，操作继续。因此，要降低耗电，ADA0M0.ADA0CE 位清零。在 IDLE1 和 IDLE2 模式下，因为不能保留模拟输入电压值，所以释放 IDLE1 和 IDLE2 模式之后 A/D 转换结果无效。设置 IDLE1 和 IDLE2 模式之前转换结果有效。	p. 413 <input type="checkbox"/>
			稳定时间期间重写寄存器和触发输入	稳定时间期间禁止重写 ADA0M0，ADA0M2，ADA0S，ADA0PFM 以及 ADA0PFT 寄存器和触发输入。	p. 413 <input type="checkbox"/>
			A/D 转换结果的变化	A/D 转换结果可能根据供应电压波动而变化或可能受噪音影响。要减少变化，请采取措施，例如平均 A/D 转换结果。	p. 413 <input type="checkbox"/>
			A/D 转换结果滞后特性	连续比较类型 A/D 转换器在内部采样 & 保持电容内保持模拟输入电压且接着进行 A/D 转换。A/D 转换完成之后，模拟输入电压保持在内部采样 & 保持电容内。结果，可能出现以下现象。 ●当 A/D 转换使用相同通道时，如果电压高于或低于先前 A/D 转换，则可能出现滞后特性可能，其中转换结果受先前值影响。因此，即使在相同电压下进行转换，结果也可能不同。 ●当切换模拟输入通道时，可能出现滞后特性，其中转换结果受先前通道值影响。这是因为 A/D 转换器用于 A/D 转换。因此，即使在相同电压下进行转换，结果也可能不同。因此，要获得更准确的转换结果，请对相同通道连续执行两次 A/D 转换，且舍弃第一次转换结果。	p. 413 <input type="checkbox"/>
第十二章	软件	异步串行接口 A (UARTA)	UAnOPT0 寄存器	SBF 接收 (UAnSRF 位 = 1) 期间禁止设置 UAnSRT 和 UAnSTT 位 (为 1)。	p. 423 <input type="checkbox"/>
			SBF 接收	如果数据接收期间传输 SBF，则出现组帧误差。	p. 433 <input type="checkbox"/>
				禁止 SBF 接收 (UAnSRF = 1) 期间 BF 接收触发位 (UAnSRT) 及 SBF 传输触发位 (UAnSTT) 设为 1。	p. 433 <input type="checkbox"/>
			连续传输程序	当进行连续传输期间初始化传输时，确保 UAnSTR.UAnTSF 位为 0，接着进行初始化。当不能保证 UAnTSF 位为 1 时传输初始化的数据。 在连续传输情况下，从停止位到下一数据起始位的传输率通过两个操作时钟脉冲自正常比率扩展。	p. 435 <input type="checkbox"/>



章节	类别	功能	功能详解	注意事项	页码	
第十二章	软件	异步串行接口 A (UARTA)	UART接收	即使当出现接收误差时也要确保读取UAnRX寄存器。如果不读取UAnRX寄存器，则下一次数据接收期间出现溢出误差，且接收误差连续不确定出现。	p. 437	<input type="checkbox"/>
				假定仅有一个停止位执行接收期间的操作。忽略第二个停止位。	p. 437	<input type="checkbox"/>
				当接收完成时，读取UAnRX寄存器之后产生接收完成中断请求信号(INTUAnR)，且UAnPWR或UAnRXE位清零。如果INTUAnR信号产生之前UAnPWR或UAnRXE位清零，则不能保证读取UAnRX寄存器的值。	p. 437	<input type="checkbox"/>
				如果接收UARTAn完成处理 (INTUAnR信号产生)且UAnPWR位= 0或UAnRXE位= 0冲突，则可能产生INTUAnR信号而不管这些存储在 UAnRX寄存器的数据。 要完成接收而不等待INTUAnR信号产生，设置 (1)中断控制寄存器 (UAnRIC)的中断屏蔽标记(UAnRMK)且接着设置 (1)UAnPWR位= 0或UAnRXE位= 0之后，请确保 UAnRIC寄存器的中断请求标记(UAnRIF)清零(0)。	p. 437	<input type="checkbox"/>
			接收误差	当产生INTUAnR信号时，必须读取UAnSTR寄存器以核对误差。	p. 438	<input type="checkbox"/>
				如果连续接收期间发生接收误差中断，则读取UAnSTR寄存器的内容必须在完成下一次接收之前读取，接着执行误差处理。	p. 439	<input type="checkbox"/>
			LIN 功能	当使用LIN功能时，将UAnCTL0.UAnPS1和UAnCTL0.UAnPS0位固定为00。	p. 440	<input type="checkbox"/>
			UAnCTL1 寄存器	UAnCTL0.UAnPWR位清零之前重写UAnCTL1寄存器。	p. 443	<input type="checkbox"/>
			UAnCTL2 寄存器	重写UAnCTL2寄存器之前UAnCTL0.UAnPWR位清零或UAnTXE和UAnRXE位清零。	p. 444	<input type="checkbox"/>
			波特率误差	传输期间的波特率误差必须在接收端的误差公差范围内。	p. 445	<input type="checkbox"/>
				接收期间的波特率误差必须满足在(5)接收期间可允许波特率范围中指示的范围。	p. 445	<input type="checkbox"/>
				接收期间可允许的波特率范围	接收期间的波特率误差必须使用以下等式设置在可允许误差范围内。	p. 447
	当供应到UARTAn的时钟停止时	当供应到UARTAn的时钟停止时(例如，在IDLE1，IDLE2或停止模式下)，在时钟供应停止之前立即停止保持值的每一寄存器的操作。TXDAn引脚输出在时钟供应停止之前还立即保持和输出该值。然而，时钟供应恢复之后不保证此操作。因此，时钟供应恢复之后，电路应通过将UAnCTL0.UAnPWR，UAnCTL0.UAnRXEn以及UAnCTL0.UAnTXEn位设为000初始化。	p. 450	<input type="checkbox"/>		
	RXDA1 引脚 KR7 引脚	RXDA1和KR7引脚必须不同时使用。要使用RXDA1引脚，则禁止使用KR7引脚。要使用KR7引脚，则禁止使用RXDA1引脚(建议PFC91位设为1且PFCE91位清零)。	p. 450	<input type="checkbox"/>		

章节	类别	功能	功能详解	注意事项	页码
第十二章	软件	异步串行接口A (UARTA)	当执行接收数据传输时	在UARTAn中, 不出现由通信误差导致的中断。当执行接收数据传输时, 即使传输期间出现误差(奇偶校验, 溢出, 组帧)也不能进行误差处理。通信期间读取UAnSTR寄存器以核对误差。	p. 450 <input type="checkbox"/>
			开始UARTAn	以下系列中开始 UARTAn。 <1> UAnCTL0.UAnPWR 位设为 1。 <2> 设置端口。 <3> UAnCTL0.UAnTXE位设为1, UAnCTL0.UAnRXE位设为1。	p. 450 <input type="checkbox"/>
			停止UARTAn	以下系列中停止 UARTAn。 <1> 设置 UAnCTL0.UAnTXE 位, UAnCTL0.UAnRXE 位。 <2> 设置端口并设置UAnCTL0.UAnPWR位(如果端口设置不变, 则没有问题)。	p. 450 <input type="checkbox"/>
			在传输模式下	在传输模式下(UAnCTL0.UAnPWR位= 1和UAnCTL0.UAnTXE位= 1), 通过软件禁止UAnTX寄存器重新写入相同值, 因为通过写入此寄存器传输开始。要连续传输相同值, 覆盖相同值。	p. 450 <input type="checkbox"/>
			在连续传输过程	在连续传输过程, 从停止位到下一起始位的传输率将通常更多扩展 2个基本时钟脉冲。然而, 接收端通过检测起始位初始化时序以便不影响接收结果。	p. 450 <input type="checkbox"/>
			片上调试模式	如果在片上调试(OCD)模式下执行中断命令且如果UART 接收数据, 则出现溢出误差。	p. 450 <input type="checkbox"/>
第十三章	软件	3线可变数据长度串行I/O接口 (CSIB)	CBnCTL0 寄存器	要强行挂起发送/接收操作过程, 应将CBnPWR位清为0, 而不是将CBnRXE或CBnTXE位清0。同时, 时钟输出停止。	p. 454 <input type="checkbox"/>
				确保要将位3 和位 2清至“0”。	p. 456 <input type="checkbox"/>
			CBnCTL1 寄存器	只有当CBnCTL0.CBnPWR 位 = 0, 或 CBnCTL0.CBnTXE 及 CBnRXE 位= 0时, CBnCTL1寄存器才能被重写。	p. 457 <input type="checkbox"/>
				设置通信时钟 (fCCLK)为 8 MHz 或更低。	p. 457 <input type="checkbox"/>
			CBnCTL2 寄存器	只有当CBnCTL0.CBnPWR 位 = 0, 或 CBnCTL0.CBnTXE 及 CBnRXE 位= 0时, CBnCTL2寄存器才能被重写。	p. 458 <input type="checkbox"/>
			连续传输模式 (主模式, 发送模式)	在连续发送模式下, 不产生接收完成中断请求信号(INTCBnR)。	p. 475 <input type="checkbox"/>
			连续传输模式 (主模式, 接收模式)	在连续发送模式下, 不产生接收完成中断请求信号(INTCBnR)。	p. 484 <input type="checkbox"/>
			时钟时序	在单向传输模式下, 在CBnTSF位为1的情况下, CBnTX 寄存器写操作会被忽略。但是, 这对传输期间的操作并没有影响。	pp. 493, 494 <input type="checkbox"/>
			PRSM0 寄存器	当钟表定器和CSIB0正在运行时, 请不要重写PRSM0寄存器。	p. 496 <input type="checkbox"/>
				在BGCE0位设置为1之前, 对PRSM0寄存器进行设置。	p. 496 <input type="checkbox"/>
			PRSCM0 寄存器	当钟表定器和CSIB0正在运行时, 请不要重写PRSCM0寄存器。	p. 497 <input type="checkbox"/>
				在PRSM0.BGCE0位设置为1之前, 对PRSCM0寄存器进行设置。	p. 497 <input type="checkbox"/>
			波特率的产生	将 fBRG 设置为 8 MHz 或更低一些。	p. 497 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第十三章	软件	3线可变数据长度串行I/O接口(CSIB)	CBnCTL0 寄存器 CBnCTL1 寄存器 CBnCTL2 寄存器	关于在操作期间(CBnCTL0.CBnPWR 位为 1)禁止重写的寄存器，如果在操作期间，错误地执行了重写操作，那么请立即将 CBnCTL0.CBnPWR 设置为 0，然后初始化 CSIBn。 操作期间禁止重写的寄存器如下所示。 <ul style="list-style-type: none"> <li>• CBnCTL0 寄存器: CBnTXE, CBnRXE, CBnDIR, CBnTMS 位</li> <li>• CBnCTL1 寄存器: CBnCKP, CBnDAP, CBnCKS2 to CBnCKS0 位</li> <li>• CBnCTL2 寄存器: CBnCL3 to CBnCL0 位</li> </ul>	p. 498 <input type="checkbox"/>
			通信类型 2 和 4	在 通信类型 2 和 4 (CBnCTL1.CBnDAP 位 = 1)中, 要在接收完成中断请求(INTCBnR)产生后的半个 SCKBn 时钟期内, 将 CBnSTR.CBnTSF 位清 0。 在单向传输模式下, 通信期间(CBnTSF 位 = 1)的下一个发送数据写操作会被忽略, 也不会启动下一次通信。同样, 如果设置了只接收的通信方式(CBnCTL0.CBnTXE 位= 0, CBnCTL0.CBnRXE 位= 1), 在通信期间(CBnTSF 位 = 1), 即使读取了接收数据, 也不会启动下一次通信。 因此, 当在 通信类型 2 或 4 下(CBnDAP 位 = 1), 使用单向传输模式, 要特别注意以下几点。 <ul style="list-style-type: none"> <li>• 如果要启动下一次数据发送, 首先请确认 CBnTSF 位 = 0, 然后再将发送数据写入 CBnTX 寄存器中去。</li> <li>• 如果要在只接收的通信方式 (CBnCTL0.CBnTXE 位 = 0, CBnCTL0.CBnRXE 位= 1) 连续执行下一个接收操作, 请先确认 CBnTSF 位 = 0, 然后再读取 CBnRX 寄存器数据。</li> </ul> 或者, 使用连续传输模式, 而不使用单向传输模式。	p. 498 <input type="checkbox"/>
第十四章	软件	中断/异常处理功能	不可屏蔽中断	要了解不可屏蔽中断服务执行不可屏蔽中断请求信号INTWDT2的相关情况, 请参阅14.2.2 (2) 从 INTWDT2 信号返回。 在不可屏蔽中断服务期间, 当EP和NP位由于LDSR指令的执行而发生改变的时候, 为了保证在使用RETI指令返回期间, 正确的恢复PC和PSW的值, 必需在RETI指令的前一条指令使用指令LDSR 将EP和NP位分别设定回到0值和1值。	p. 502 <input type="checkbox"/>
			不可屏蔽中断	在可屏蔽中断服务期间, 当EP和NP位由于LDSR指令的执行而发生改变的时候, 为了保证在使用RETI指令返回期间, 正确的恢复PC和PSW的值, 必需在RETI指令的前一条指令使用指令LDSR 将EP位和NP位分别设定回到0值。	p. 509 <input type="checkbox"/>
			多重中断	要执行多重中断服务, 必需在执行EI指令之前, 将EIPC 和EIPSW寄存器值保存起来。当从多重中断服务返回时, 要在DI 指令执行之后, 恢复EIPC 和 EIPSW的值。	pp. 511 to 513 <input type="checkbox"/>
			中断控制寄存器	要对xxICn.xxIFn位进行读取, 请禁止中断(DI)或将中断屏蔽。如果在中断允许(EI)期间或未屏蔽对应中断而读取xxIFn 位, 当中断响应和读取该xxIFn位发生冲突时, 可能读不到正确的数值。 如果响应了一个中断请求信号, xxIFn 标志便由硬件自动复位。	p. 514 <input type="checkbox"/>
			IMR0 to IMR2 寄存器	设备文件将xxICn.xxMKn位定义为保留字。如果使用xxMKn 名对寄存器某一位进行位操作, 那么会对xxICn寄存器内容进行重写, 而不是IMRm寄存器 (作为结果, IMRm寄存器内容也被重写了)。 请将IMR2寄存器的 15至 11位 以及 7 至 4位设置为“1”。如果更改了对这些位的设置, 并不保证该操作的有效性。	p. 516 <input type="checkbox"/>
					p. 516 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第十四章	软件	中断/异常处理功能	IMP0 to IMR2 寄存器	请将IMR2寄存器的 15至 11位 以及 7至 4位设置为“1”。如果更改了对这些位的设置，并不保证该操作的有效性。	p. 516 <input type="checkbox"/>
			ISPR 寄存器	如果在中断允许(EI)状态下读取ISPR寄存器的时候，响应了一个中断请求，那么对ISPR寄存器的读取值，就有可能的是被中断响应所设置之后的值。因此，为了在中断响应前，准确地读取ISPR寄存器值，请在中断禁止(DI)状态下，对该寄存器进行读取。	p. 517 <input type="checkbox"/>
			从软件异常处理过程中恢复	在软件异常处理期间，当EP和NP位由于LDSR指令的执行而发生改变的时候，为了保证在使用RETI指令返回期间，正确的恢复PC和PSW的值，必需在RETI指令的前一条指令使用指令LDSR将EP位设定回到1值和将NP位设定回到0值。	p. 520 <input type="checkbox"/>
			非法操作码 定义	由于有可能在后来将该指令分配给一非法操作代码，所以建议不要使用该指令。	p. 522 <input type="checkbox"/>
			从非法操作码恢复	只有在非法代码和DBRET指令执行的间隔期间，才能访问DBPC 和 DBPSW 寄存器。	p. 523 <input type="checkbox"/>
			从调试陷阱恢复	只有在DBTRAP指令和DBRET指令执行的间隔期间，才能访问DBPC 和 DBPSW 寄存器。	p. 525 <input type="checkbox"/>
			INTF0, INTR0 寄存器s	当功能由外部中断功能(复用功能)改变成端口功能时，可能会检测到边沿信号。因此，应先将INTF0n和INTR0n位清为00，然后再设置为端口功能。	p. 527 <input type="checkbox"/>
				如果相应引脚不用作NMI引脚或INTP0 至 INTP3引脚时，请确保要将INTF0n位和INTR0n位清为 00。	p. 527 <input type="checkbox"/>
			INTR3L, INTF3L 寄存器s	当功能由外部中断功能(复用功能)改变成端口功能时，可能会检测到边沿信号。因此，应先将INTF31和INTR31位清为00，然后再设置为端口模式。	p. 528 <input type="checkbox"/>
				如果相应引脚不用作INTP7引脚时，请确保要将INTF31和INTR31位清为 00。	p. 528 <input type="checkbox"/>
			INTF9H, INTR9H 寄存器s	当功能由外部中断功能(复用功能)改变成端口功能时，可能会检测到边沿信号。因此，应先将INTF9n和INTR9n位清为00，然后再设置为端口模式。	p. 529 <input type="checkbox"/>
				如果相应引脚不用作INTP4 至 INTP6引脚时，请确保要将INTF9n和INTR9n位清为 00。	p. 529 <input type="checkbox"/>
			NFC 寄存器	采样时钟改变后，到数字噪音消除器初始化，所需要的时间为:采样时钟周期 × 由 NFSTS 位所设定的次数。如果在采样时钟改变后，同时，在“采样时钟周期 × 由 NFSTS 位所设定的次数”时间经过之前，输入了 INTP3 有效边沿触发信号，于是就可能会产生中断请求信号。因此，当使用中断功能时，要注意以下几点。 • 当使用中断功能时，在经过“采样时钟周期 × 由NFSTS位所设定的次数”时间之后，在中断请求标志(PIC3.PIF3位)已经清0后，要设置中断允许。	p. 530 <input type="checkbox"/>
			NMI 引脚	NMI 引脚是P02引脚的复用功能引脚。该引脚复位后为一通常端口引脚。因此，要启用NMI引脚，请使用PMC0寄存器使NMI引脚有效。NMI引脚的初始化设置为“无边沿检测”，请通过使用INTF0和INTR0寄存器来选择NMI 引脚的有效触发边沿。	p. 533 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第十五章	软件	按键中断功能	KRM 寄存器	在将KRM寄存器清零为00H一次后，应重写KRM寄存器。	p. 535 <input type="checkbox"/>
				如果KRM寄存器被改变，则可能会产生中断请求信号（INTKR）。要防止这种情况的发生，应在禁用中断（DI）或在屏蔽后改变KRM寄存器，然后将中断请求标志（KRIC.KRIF位）清为0，并启用中断（EI）或清除屏蔽。	p. 535 <input type="checkbox"/>
			KR0 至 KR7 引脚	如果向KR0至KR7中任何一个引脚输入低电平，则即输入另一个引脚的下降边缘也不会产生INTKR信号。	p. 535 <input type="checkbox"/>
			RXDA1引脚 KR7引脚	RXDA1和KR7引脚一定不能同时使用。使用RXDA1引脚时，不要使用KR7引脚。使用KR7引脚时，不要使用RXDA1引脚（建议将PFC91位设置为1并将PFCE91位清为0）。	p. 535 <input type="checkbox"/>
			使用按键中断功能	要使用按键中断功能，一定要将端口引脚设置为按键中断引脚，再启用KRM寄存器。要从按键中断引脚切换为端口引脚，应禁用KRM寄存器的工作再设置端口引脚。	p. 535 <input type="checkbox"/>
第十六章	软件	待机功能	PSC 寄存器	在设置IDLE1、IDLE2、STOP或子IDLE模式前，应先设置PSMR.PSM1和PSMR.PSM0位再设置STP位。	p. 538 <input type="checkbox"/>
				当HALT模式解除时，NMI1M、NMI0M、INTM位的设置无效。	p. 538 <input type="checkbox"/>
				如果NMI1M、NMI0M或INTM位在STP位设置为1的同时设置为1，则NMI1M、NMI0M或INTM位的设置变为无效。如果在设置IDLE1/IDLE2/STOP模式时有未屏蔽的中断请求信号正在保留待定，应对应该中断请求信号的位（NMI1M、NMI0M或INTM）设置为1，再将STP位设置为1。	p. 538 <input type="checkbox"/>
			PSMR 寄存器	一定要将2至7位清为“0”。	p. 539 <input type="checkbox"/>
				仅当PSC.STP位为1时PSM0和PSM1位才有效。	p. 539 <input type="checkbox"/>
			OSTS 寄存器	Stop模式解除后的等待时间不包括在STOP模式解除后时钟振荡开始（下图的“a”）之前的时间，且与STOP模式是由复位解除还是由于出现中断请求信号而解除无关。	p. 540 <input type="checkbox"/>
				一定要将3至7位清为“0”。	p. 540 <input type="checkbox"/>
				复位解除后的振荡稳定时间为 $2^{16}/f_x$ （因为OSTS寄存器的初始值=06H）。	p. 540 <input type="checkbox"/>
			HALT 模式	应在HALT指令后插入五个（或更多）NOP指令。	p. 541 <input type="checkbox"/>
				如果当未屏蔽中断请求信号正在保留待定时执行HALT指令，状态会切换到HALT模式，但随后HALT模式会立即被待定的中断请求解除。	p. 541 <input type="checkbox"/>
			IDLE1 模式	应在PSC寄存器存储数据指令后插入五个或更多NOP指令来设置为IDLE1模式。	p. 543 <input type="checkbox"/>
				若当未屏蔽中断请求信号正在保留待定时设置IDLE1模式，则IDLE1模式会因待定中断请求而立即解除。	p. 543 <input type="checkbox"/>
			解除IDLE1 模式	通过将PSC.NMI1M、PSC.NMI0M、PSC.INTM位设置为1禁用的中断请求信号变为无效，IDLE1模式不解除。	p. 543 <input type="checkbox"/>
				如果通过NFC寄存器选择消除数字噪声，如果从fxx/64、fxx/128、fxx/256、fxx/512、fxx/1024中选择采样时钟，则无法通过INTP3引脚的中断请求信号解除IDLE1模式。详情请见“14.6.2（4）消噪控制寄存器（NFC）”。	p. 543 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第十六章	软件	待机功能	IDLE2模式	应在PSC寄存器存储数据指令后插入五个或更多NOP指令来设置为IDLE2模式。	p. 545 <input type="checkbox"/>
				若当未屏蔽中断请求信号正在保留待定时设置IDLE2模式，则IDLE2模式会因待定时中断请求而立即解除。	p. 545 <input type="checkbox"/>
			解除IDLE2模式	通过将PSC.NMI1M、PSC.NMI0M、PSC.INTM位设置为1禁用的中断请求信号变为无效，IDLE2模式不解除。	p. 545 <input type="checkbox"/>
				如果通过NFC寄存器选择消除数字噪声且从fxx/64、fxx/128、fxx/256、fxx/512、fxx/1024中选择采样时钟，则无法通过INTP3引脚的中断请求信号解除IDLE2模式。详情请见“14.6.2（4）消噪控制寄存器（NFC）”。	p. 545 <input type="checkbox"/>
			Stop 模式	应在PSC寄存器存储数据指令后插入五个或更多NOP指令来设置为STOP模式。	p. 548 <input type="checkbox"/>
				若当未屏蔽中断请求信号正在保留待定时设置为STOP模式，则STOP模式会因待定时中断请求而立即解除。	p. 548 <input type="checkbox"/>
			解除STOP模式	通过将PSC.NMI1M、PSC.NMI0M、PSC.INTM位设置为1禁用的中断请求信号变为无效，STOP模式不解除。	p. 548 <input type="checkbox"/>
				如果通过NFC寄存器选择消除数字噪声且从fxx/64、fxx/128、fxx/256、fxx/512、fxx/1024中选择采样时钟，则无法通过INTP3引脚的中断请求信号解除STOP模式。详情请见“14.6.2（4）消噪控制寄存器（NFC）”。	p. 548 <input type="checkbox"/>
			Stop模式的工作状态	如果在模数转换器正在工作时设置为STOP模式，则模数转换器自动停止，在STOP模式解除后重新开始工作。但是，在这种情况下，STOP模式解除后的模数转换结果是无效的。在设置为STOP模式前的所有模数转换结果均无效。	p. 549 <input type="checkbox"/>
				即使在模数转换器正在工作时设置为STOP模式，其功耗的减低等同于在设置为STOP模式前停止模数转换器。	p. 549 <input type="checkbox"/>
			副时钟工作模式	在操作CK3位时，不要改变CK2至CK0位的设置值（建议使用位操作指令来对各位进行操作）。 于PCC寄存器的详情请参见“5.3（1）处理器时钟控制寄存器（PCC）”。	p. 551 <input type="checkbox"/>
				如果下列条件不满足，应改变CK2至CK0位使得条件满足，并设置为副时钟工作模式。 内部系统时钟（fCLK）>副时钟（fXT）× 4	p. 551 <input type="checkbox"/>
			解除副时钟工作模式	在操作CK3位时，不要改变CK2至CK0位的设置值（建议使用位操作指令来对各位进行操作）。关于PCC寄存器的详情请参见“5.3（1）处理器时钟控制寄存器（PCC）”。	p. 551 <input type="checkbox"/>
			副时钟工作模式下的工作状态	一定要在停止主时钟前停止PLL（PLLCTL.PLLON=0）。	p. 552 <input type="checkbox"/>
				当CPU工作在副时钟下且主时钟振荡停止时，访问出现等待的寄存器是被禁止的。如果产生了等待，只能通过复位进行解除（见3.4.8（2））。	p. 552 <input type="checkbox"/>
			子IDLE模式	在将设置为子IDLE模式的指令存储到PSC寄存器中后，应插入五个（或更多）NOP指令。	p. 553 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第十六章	软件	待机功能	子IDLE模式	若当未屏蔽中断请求信号正在保留待定时设置为子IDLE模式，则子IDLE模式会因待定时中断请求而立即解除	p. 553 <input type="checkbox"/>
			解除子IDLE模式	通过将PSC.NMI1M、PSC.NMI0M、PSC.INTM位设置为1禁用的中断请求信号变为无效，子IDLE模式不解除。	p. 554 <input type="checkbox"/>
				当子IDLE模式解除时，从解除子IDLE模式的中断请求信号生成到模式解除要经过12个副时钟周期（约366μs）。	p. 554 <input type="checkbox"/>
				如果通过NFC寄存器选择消除数字噪声且从fxx/64、fxx/128、fxx/256、fxx/512、fxx/1024中选择采样时钟，则无法通过INTP3引脚的中断请求信号解除子IDLE模式。详情请见“14.6.2（4）消噪控制寄存器（NFC）”。	p. 554 <input type="checkbox"/>
			子IDLE模式的工作状态	一定要在停止主时钟前停止PLL（PLLCTL.PLLON位=0）	p. 555 <input type="checkbox"/>
				要实现低功耗，应在切换为子IDLE模式前停止模数转换器。	p. 555 <input type="checkbox"/>
第十七章	软件	复位功能	应急工作模式	当CPU正在内部振荡时钟下工作时，访问产生了等待状态的寄存器是被禁止的。关于产生等待状态的寄存器，请见“3.4.8（2）访问特定片上外围I/O寄存器”。	p. 556 <input type="checkbox"/>
			RESF 寄存器	只能向该寄存器的各位写入“0”。如果写入“0”与设置标志（出现复位）冲突，应首先设置标志。	p. 557 <input type="checkbox"/>
	硬件		复位后内部RAM 状态	由于支持引导互换功能，V850ES/HF2在内部系统复位状态解除后其固件会使用一部分内部RAM。因此，在加电复位后一些RAM区的内容无法保留。详情请见“17.4 复位解除后的操作”。	pp. 558, 560 <input type="checkbox"/>
			Reset引脚输入后的硬件状态	电源打开时，下面的引脚即使在复位过程中也可能会临时输出未定义的电平。 ● P53/KR3/TIQ00/TOQ00/DDO引脚	p. 558 <input type="checkbox"/>
	硬件,			OCDM寄存器通过RESET引脚输入进行初始化。因此要小心注意的是，如果在复位解除后OCDM.OCDM0位清零前有高电平输入到P05/DRST引脚，则会进入片上调试模式。详情请见“第四章端口功能”。	p. 558 <input type="checkbox"/>
	第十八章	软件	时钟监视器	CLM 寄存器	一旦CLME位设置为1，就无法通过除复位之外的其它方式清为0。
当出现时钟监视器的复位时，CLME位清为0，RESF.CLMRF位设置为1。					p. 565 <input type="checkbox"/>
内部振荡器				利用选项字节功能（见二十三章）启用内部振荡器停止并将RCM.RSTOP位设置为1，可停止内部振荡器。	p. 566 <input type="checkbox"/>
				当内部振荡器停止时，时钟监视器停止。	p. 566 <input type="checkbox"/>
第二十章	软件	低压检测器	LVIM 寄存器	在将LVION位设置为1后，在用LVIF位检查电压之前应等待0.2ms（最多）。	p. 572 <input type="checkbox"/>
				当LVION位=1、LVIMD位=0时，LVIF标志的值作为输出信号INTLVI输出。	p. 572 <input type="checkbox"/>
				一定要将2至6位清为“0”。	p. 572 <input type="checkbox"/>
				在将LVIM.LVION、LVIM.LVIMD位设置为1后，在由于非低压检测导致的复位请求出现之前低压检测器无法停止。	p. 572 <input type="checkbox"/>

(25/29)

章节	类别	功能	功能详解	注意事项	页码
第二十章	软件	低压检测器	LVIS 寄存器	在将LVIM.LVION、LVIM.LVIMD位设置为1后，在由于非低压检测导致的复位请求出现之前该寄存器无法写入。 一定要将1至7位清为“0”。	p. 573 <input type="checkbox"/>
			RAMS 寄存器	下面展示了复位后的特定序列。 ●设置条件： 检测到低于检测电平的电压 通过指令进行设置 通过看门狗定时器溢出生成复位信号 当正在访问 RAM时产生复位信号 通过时钟监视器产生复位信号 ●清零条件： 以特定顺序写入0	p. 573 <input type="checkbox"/>
			要用于内部复位信号时	如果LVIMD位设置为1，在非LVI的复位请求产生之前LVIM和LVIS寄存器的内容不会发生改变。	p. 574 <input type="checkbox"/>
			PEMU1 寄存器	EVARAMIN位不会自动清零。	p. 578 <input type="checkbox"/>
第二十一章	硬件	闪存	闪存映射	仅“0”可以被写到该寄存器的各位。如果写入“0”与设置标记冲突（复位重复），设置标记优先。	p. 582 <input type="checkbox"/>
			通信模式	使不显示的引脚符合不使用引脚处理。(参阅 2.3节引脚I/O 电路类型和未使用引脚的连接)。 根据需要1 Ω 连接到 10 Ω 电阻上。	pp. 587, 588 <input type="checkbox"/>
				不要将高电平输入DRST引脚。	pp. 587, 588 <input type="checkbox"/>
			PG-FP4	如图22-6所示连接这些引脚，或通过板上的下拉电阻连接至GND。	p. 588 <input type="checkbox"/>
				时钟不能通过闪存编程器的CLK引脚提供。在板上创建振荡器并提供时钟。	p. 588 <input type="checkbox"/>
			FA-80GK-9EU-A	确保将REGC引脚通过4.7 μF（推荐值）电容器连接到GND。	p. 589 <input type="checkbox"/>
				时钟不能从闪存编程器的CLK引脚提供。在板上创建振荡器并提供时钟。	p. 589 <input type="checkbox"/>
			FA-80GK-9EU-A (在 CSIB0 + HS 模式下)	如下所示连接FLMD1 引脚，或通过下拉电阻将其连接至板上的GND端口。	p. 591 <input type="checkbox"/>
				在闪存配线适配器创建振荡器（如虚线所示）并提供时钟。此处为振荡器的示例。	p. 591 <input type="checkbox"/>
				严禁将高电平输入到 DRST 引脚。	p. 591 <input type="checkbox"/>
			选择通信模式	选择 UARTA0 时，在接收FLMD0脉冲之后，接收时钟根据从专用闪存编程器发出的复位命令计算。	p. 593 <input type="checkbox"/>
			FLMD1引脚连接	如果在主存储器写入期间和立即复位之后VDD 信号从另一个设备输入到FLMD1引脚，则隔离此信号。	p. 595 <input type="checkbox"/>
			安全自编程 (引导互换功能)	安全自编程 (引导互换功能)	p. 600 <input type="checkbox"/>
			FLMD0引脚处理	确保释放复位功能时FLMD0引脚处于0 V。	p. 602 <input type="checkbox"/>



章节	类别	功能	功能详解	注意事项	页码
第二章	硬件	选项字节功能	CA850 采样程序	确保在本节中写入 6 字节。如果不足 6 字节，连接器运行时，产生错误。 错误信息: F4112:不合法“OPTION_BYTES”节范围。	p. 605 <input type="checkbox"/>
第二十四章	硬件, 软件	片上调试功能	OCDM 寄存器	<p>当外部复位之后，DDI, DDO, DCK 和 DMS 引脚不用作片上调试引脚而用作端口引脚时，必须采取以下措施的一种。</p> <ul style="list-style-type: none"> <li>将 P05/INTP2/DRST 引脚输入低电平。</li> <li>设定 ODCM0 位。在这种情况下，请采取以下措施。</li> </ul> <p>&lt;1&gt; 将 ODCM0 位清零。</p> <p>&lt;2&gt; 将 P05/INTP2/DRST 将引脚固定在低电平，直到完成&lt;1&gt;。</p>	p. 610 <input type="checkbox"/>
				DRST引脚有一个片上下拉电阻。此电阻在OCDM0标志位清零时断开。	p. 610 <input type="checkbox"/>
	硬件		注意事项 (有DCU)	在RUN（程序执行）期间，如果(从目标系统或从内部复位资源)输入复位信号，此断点功能可能会出现故障。	p. 612 <input type="checkbox"/>
				即使屏蔽功能屏蔽了该复位信号，如果从引脚输入一个复位信号，I/O 缓冲器（端口引脚）可能被复位。	p. 612 <input type="checkbox"/>
				因为看门狗时钟 2 生成的目标复位或内复位使内闪存的软件断点设置暂时无效。当发生硬件中断或被迫中断时，断点再次生效，但直至此时，也不会发生软件中断现象。	p. 612 <input type="checkbox"/>
				中断期间引脚复位被屏蔽，且 CPU 和外围设备 I/O 不能复位。当执行用户程序时，一旦闪存被 DMM 重写或被 RAM 监测功能读取时，如果生成引脚复位或内复位，则 CPU 和外围设备 I/O 不能正确复位。	p. 612 <input type="checkbox"/>
				<p>当满足下列条件（a）和（b）且由于中断等仿真器（IECUBE®, MINICUBE）操作停止时，看门狗时钟 2 不能停止，且发生复位或非屏蔽中断。当发生复位时，调试器挂断。</p> <p>主时钟或副时钟被看门狗时钟 2 用作源时钟。</p> <p>内振荡器时钟停止(RCM.RSTOP 位 = 1)。</p> <p>为了避免这种情况，执行下列操作之一。</p> <ul style="list-style-type: none"> <li>当使用仿真器时，使用内振荡器时钟作为源时钟。</li> <li>当使用仿真器时，不停止内振荡器。</li> </ul>	p. 612 <input type="checkbox"/>
				<p>当满足下列条件(a)和（b），且由于中断等仿真器(IECUBE, MINICUBE)操作停止时候，即使外围设备中断功能设置到“Break”，TMM 不停止。</p> <p>(a) INTWT，内振荡器时钟 (fR/8)或副时钟被选做 TMM 源时钟。</p> <p>(b) 停止主时钟。</p> <p>为了避免这种情况，执行下列操作之一。</p> <ul style="list-style-type: none"> <li>当使用仿真器时，主时钟(fXX, fXX/2, fXX/4, fXX/64, fXX/512)被用作源时钟。</li> <li>当使用仿真器时，禁用主时钟振荡器。</li> </ul>	p. 612 <input type="checkbox"/>
	硬件			在片上调试模式中，DDD引脚被迫设为高电平输出。	p. 612 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第二十四章	硬件	片上调试功能	注 意 事 项 (无 DCU)	禁止安装曾在批量生产的产品上用作调试的设备，因为在调试期间，闪存被重写且闪存重写次数不能被保证。此外，禁止将调试监视程序嵌入批量生产的产品中。	p. 621 <input type="checkbox"/>
	软件			如果满足下列条件之一，则不能执行被迫断点。 ●禁用中断(DI) ●用来在 MINICUBE2 和目标设备间进行通信的串行接口中断被屏蔽。 ●当禁止可屏蔽中断备用释放时，进入待机模式。 ●MINICUBE2 和目标设备间的通信模式为 UARTA0，主时钟已停止。	p. 621 <input type="checkbox"/>
				当伪实时 RAM 监视器功能（RRM）和 DMM 功能无法运行时 如果满足下列条件之一，伪 RRM 功能和 DMM 功能无法运行 ●中断禁止 (DI) ●用于 MINICUBE2 和目标设备间进行通信的串行接口中断被屏蔽。 ●当禁止可屏蔽中断备用释放时，进入待机模式。 ●MINICUBE2 和目标设备间的通信模式为 UARTA0，主时钟已停止。 ●MINICUBE2 和目标设备间的通信模式为 UARTA0，用来通信的时钟与调试器指定不同。	p. 621 <input type="checkbox"/>
				如果符合下列条件之一，伪 RRM 功能和 DMM 功能释放备用模式。 ●MINICUBE2 和目标设备的通信模式为 CSIB0。 ●MINICUBE2 和目标设备的通信模式为 UARTA0，提供主时钟。	p. 621 <input type="checkbox"/>
				需要一个明确的顺序的外围 I/O 寄存器不能用 DMM 功能写入。	p. 621 <input type="checkbox"/>
				当调试器启动时，调试监视程序的芯片擦除和写入运行，但该操作需要花费大约十几秒的时间。	p. 621 <input type="checkbox"/>
				当 CPU 操作时钟设置用调试器更改时，调试器重写到监视程序。所需要的时间和 (6) 提到的一样。对于集成调试器 ID850QB，应用于对配置对话框的设置进行更改时。	p. 621 <input type="checkbox"/>
				如果分配到调试监视程序的空间被闪存自编程序重写，则调试器运行不再正常。	p. 621 <input type="checkbox"/>
			安全 ID	在擦除闪存之后，1 被写入到整个区域。	p. 622 <input type="checkbox"/>
第二十五章	硬件	电气规格	最大绝对级别	确保不超过每个电源电压的最大绝对级别（最大值）。	p. 625 <input type="checkbox"/>
				严禁 IC 产品的输出 (或 I/O) 引脚彼此直接连接，或连接到 V <sub>DD</sub> , V <sub>CC</sub> 和 GND。	pp. 625, 626 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第二十五章	硬件	电气规格	最大绝对级别	如果任何参数即使短时超过最大绝对级别，则可能导致产品质量问题。也就是说，最大绝对级别为产品处于遭受实际损坏的边缘时的级别值，且因此产品必须在确保不超过最大绝对级别的条件下使用。 显示DC特性和AC特性的级别和条件表示正常运行期间的质量保证范围。	pp. 625, 626 <input type="checkbox"/>
				当将外部电路直接连接到逐渐升高的高阻抗状态下的引脚时，设计定时器已避免外部电路的输出冲突。	pp. 625, 626 <input type="checkbox"/>
			主时钟振荡器特性	使用主时钟振荡器时，如下在上图虚线封闭的区域接线以避免接入电容的不利影响。 <ul style="list-style-type: none"> <li>使接线长度尽量短。</li> <li>严禁将接线与其他信号线交叉。</li> <li>严禁在高波动电流流过的信号线附近布线。</li> <li>时钟使振荡器 电容器的接地点电位与 VSS 相同。</li> <li>严禁将电容器接入高电流流过的接地模式。</li> <li>严禁从振荡器获取信号。</li> </ul>	p. 628 <input type="checkbox"/>
	软件			主时钟停止且设备按副时钟运行时，切换回主时钟之前，等待到振荡稳定时间已经由程序保障为止。	p. 628 <input type="checkbox"/>
	硬件	副时钟振荡器特性		使用副时钟振荡器时，如下在上图虚线封闭的区域接线以避免接入电容的不利影响。 <ul style="list-style-type: none"> <li>使接线长度尽量短。</li> <li>严禁将接线与其他信号线交叉。</li> <li>严禁在高波动电流流过的信号线附近布线。</li> <li>时钟使振荡器 电容器的接地点电位与 VSS 相同。</li> <li>严禁将电容器接入高电流流过的接地模式。</li> <li>严禁从振荡器获取信号。</li> </ul>	p. 629 <input type="checkbox"/>
				副时钟振荡器设计为低幅度电路以降低电流消耗，而且由于大于主时钟振荡器的噪声，所以更容易出故障。因此使用副时钟时，更应注意配线方法。	p. 629 <input type="checkbox"/>
		电压调节器特性	确保VDD 上升，当 复位 = Vss = 0 V。		p. 630 <input type="checkbox"/>
		引脚泄漏电流	FLMD0 引脚值如下： <ul style="list-style-type: none"> <li>输入泄漏电流，高：2 μA (最大)</li> <li>输入泄漏电流，低-2 μA (最大)</li> </ul>		p. 632 <input type="checkbox"/>
		数据保持特性	切换到停止模式且从停止模式恢复必须在分级运行范围内进行。		p. 634 <input type="checkbox"/>

章节	类别	功能	功能详解	注意事项	页码
第二十五章	硬件	电气规范	AC 特性	如果由于电路配置加载电容超过50 pF，则通过插入缓冲器或某些其他构件，使装置的加载电容减小到 50 pF或更小。	p. 635 <input type="checkbox"/>
	软件		编程特性	当开始写入到出货产品时，“擦除待写入”与“仅写入”都记做重写方式。 示例 (P: 写入, E: 擦除) 出货产品 → P → E → P → E → P: 3 重写 出货产品 → E → P → E → P → E → P: 3 重写	p. 644 <input type="checkbox"/>
第二十章	硬件	推荐焊接条件	推荐焊接条件	严禁不同焊接方法一起使用 (局部加热除外)。	p. 646 <input type="checkbox"/>
附录A	软件	开发工具	RX850, RX850 Pro	要购买RX850 或 RX850 Pro产品，请先填写购买申请表并签署许可协议。	p. 655 <input type="checkbox"/>
附录C	软件	指令组列表	指令组	对于通用的寄存器reg1 和reg2， 不指定同一个寄存器，	p. 673 <input type="checkbox"/>

## E.1 本版本中主要修订之处

(1/2)

页码	描述
p. 40	添加了 2.5 注意事项
p. 68	在中 3.4.7 特殊寄存器添加了描述
p. 145	添加了 4.5.1 (b) 关于复用功能模式(输入)的注意事项
p. 147	修改了图 5-1 时钟发生器
p. 148	修改了 5.2 (8) 预分频器 4 中的描述
p. 154	修改了 表 5-1 各时钟操作状态
p. 158	修改了 5.5.2 (4) 可编程时钟模式寄存器 (PCLM)
p. 167	修改了 6.4 (3) TMPn I/O 控制寄存器 0 (TPnIOC0)
p. 189	在图 6-11 外部事件计数模式下操作的寄存器设置中添加注意事项
p. 210	在图 6-22 单发脉冲输出模式下的寄存器设置中添加注意事项
p. 253	在 26.7 (1) 选择器操作控制寄存器 0 (SELCNT0)中添加注意事项
p. 263	在 7.4 (3) TMQ0 I/O 控制寄存器 0 (TQ0IOC0)中添加了注
p. 289	在图 7-11 外部事件计数模式下操作的寄存器设置中添加注意事项
p. 313	在图 7-22 单发脉冲输出模式下的寄存器设置中添加注意事项
p. 382	修改了 10.3 (1) 看门狗定时器模式寄存器 2 (WDTM2)中的注意事项 3 , 4
p. 384	修改了 10.3 (2) 看门狗定时器使能寄存器 (WDTE)中的注意事项 3
p. 392	在表 11-2 转换模式设置示例中添加注意事项
p. 397	修改了 11.4 (7)掉电比较阈值寄存器 (ADA0PFT)中的描述
p. 412	修改了 11.6 (8) 读 ADA0CRn 结果中的描述
p. 413	添加了 11.6 (10) 待机模式
p. 413	添加了 11.6 (11) 在时间标准化期间重写寄存器和触发器输入
p. 413	修改了 11.6 (13) A/D 转化结果回滞特性中的描述
p. 421	在 12.3 (1) UARTAn 控制寄存器 0 (UAnCTL0)中添加了描述
p. 423	在 12.3 (4) UARTAn 选项控制寄存器 0 (UAnOPT0)中添加了描述
p. 427	在 12.4 (1) 接收完整中断请求信号中添加了描述(INTUAnR)
p. 433	在 12.5.4 SBF 接收中添加了注意事项
p. 454	修改了 13.3 (1) CSIBn 控制寄存器 0 (CBnCTL0)中的注意事项
p. 457	修改了 13.3 (2) CSIBn 控制寄存器 1 (CBnCTL1)中的注意事项并添加了注 1
p. 462	修改了 13.5 操作
p. 497	在 13.7.1 波特率生成 (器) 中添加了注意事项
p. 530	修改了 14.6.2 (4) 噪声消除控制寄存器 (NFC)中的描述
p. 538	在 16.2 (1) 省电控制寄存器 (PSC)中添加了注意事项 3
p. 558	修改了 表 17-1 有关 RESET 引脚输入的硬件状态中的 注 1 的描述
p. 560	修改了表 17-2 看门狗定时器 2 复位操作期间的硬件状态
p. 562	添加了 17.4 复位释放后的操作
p. 572	在 20.3 (1) 低电压检测寄存器(LVIM) 中添加了注意事项 4
p. 575	修改了 图 20-2 低电压检测器的操作时序 (LVIMD Bit = 1)

(2/2)

页码	描述
p. 582	修改了 22.2 存储器结构
p. 583	添加了 22.3 功能概述
p. 587	修改了 22.4 2 (1) UARTA0 中的传输速率
p. 594	修改了 表 22-7 闪存控制命令
p. 601	修改了 图 22-17 标准自编程流程
p. 603	修改了表 22-11 所使用的内部资源
p. 604	在第 23 章 选项字节功能中添加了描述
p. 606	修改了 第 24 章片上调试功能
p. 630	在 25.5 稳压器特性 中添加了注意事项
p. 646	添加了第 27 章推荐焊接条件
p. 647	添加了附录 A 开发工具
p. 674	添加了附录 D 注意事项列表
p. 703	添加了附录 E 修订历史

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