

To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

用户手册

RENESAS

V850ES/HG2

32 位单片微控制器

硬件

μ PD70F3706

μ PD70F3707

文档编号 U17718CA2V0UD00 (第二版)

发行日期 2008 年 1 月 CP(K)

© NEC Electronics Corporation 2007

日本印刷

[备忘录]

① 输入引脚处的电压波形

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

② 未使用的输入引脚的处理

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

CMOS设备的输入电平必须借助上拉或下拉电路固定在高电平或低电平。每一个未使用引脚都应该通过附加电阻连接到 V_{DD} 或GND。如果有可能尽量定义为输出引脚。对未使用引脚的处理因设备而异，必须遵循与设备相关的规定和说明。

③ ESD 防护措施

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

④ 初始化之前的状态

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

⑤ 电源开关顺序

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

对于每个设备电源的正确开关顺序必须依据设备的规范说明分别进行判断。

⑥ 电源关闭状态下的输入信号

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

每个设备电源关闭时的信号输入必须依据设备的规范说明分别进行判断。

Applilet 是日电电子有限公司在日本, 德国, 香港, 中国, 韩国, 英国和美国的注册商标。

IECUBE 是日电电子有限公司在日本和德国的注册商标。

MINICUBE 是日电电子有限公司在日本和德国的注册商标或在美国的商标。

EEPROM 是日电电子有限公司的商标。

Windows 和 **Windows NT** 是微软公司在美国和/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是实时操作系统核心的缩写。

ITRON是工业实时操作系统核心的缩写。

- 本档中的信息为2006年9月更新。今后档中的信息被更新时，不会有特别的提示，因此，如阁下正使用本档中所描述的器件进行设计开发，请您留意日电电子有限公司最新发布的数据表单(data sheet)、数据文本(data book)或其他相关文档，以获取最新公布的产品信息。并非在每个国家都可获得全部型号的产品。请联系**NEC Electronics** 销售代理，了解供货和其他相关信息。
- 未经**NEC Electronics**的书面许可，不能对本档复制。本档出现的任何错误，**NEC Electronics**不承担责任。
- 如果用户在使用本档列出的**NEC Electronics** 半导体产品或通过其他途径使用这些产品时，产生侵犯专利、版权以及其他侵犯第三方知识产权的行为，**NEC Electronics** 不承担责任。对于**NEC Electronics** 及其他子公司的任何专利、版权以及其他知识产权，**NEC Electronics** 没有以许可、明示、暗示以及其他任何方式授权。
- 档中电路、软件和其他相关信息的描述，用来说明半导体产品操作和应用的例子。客户在使用这些电路、软件和信息时负全责。客户或第三方在使用这些电路、软件和信息时造成的损失，**NEC Electronics**不承担责任。
- **NEC Electronics**尽力提高半导体产品的质量、可靠性和安全性，但请客户理解错误是不可能完全避免的。为了尽可能减少由于**NEC Electronics** 半导体产品所带来的个人财产及人身安全（包括死亡）的风险，客户在设计过程中应加强安全措施，如容错、耐火性和自检等。
- **NEC Electronics**产品分为以下三个质量等级：
“标准”、“专业”、“特级”。“特级”质量等级仅用于客户定制的半导体产品。一种半导体产品的应用主要依据它的质量等级。客户在使用某种半导体产品之前应先了解它的质量等级。
“标准”：计算机，办公设备，通信设备，测试设备，视频音频设备，家用电子产品，机械工具，个人电子设备和工业机器人
“专业”：运输设备（汽车，火车，轮船等），交通控制系统，防灾系统，反犯罪系统，安全设备和医疗设备（不是专用于生命救护的设备）
“特级”：飞机，航空设备，水下中继器，核反应堆控制系统，生命救护系统和用于生命救护的医疗设备等。
除非在**NEC Electronics**规格书或规格手册中特别规定，一般的**NEC Electronics**产品的质量等级都是“标准”的。如果客户希望在不受**NEC Electronics**产品应用限制的条件使用**NEC Electronics**半导体产品，必须事先与**NEC Electronics**销售代理联系，以确定**NEC Electronics**是否支持该应用环境。
(注)
(1) “**NEC Electronics**”在这里是指**NEC Electronics Corporation**和它的主要子公司。
(2) “**NEC Electronics**半导体产品”是指由**NEC Electronics**或为**NEC Electronics**开发和制造的半导体产品（如上述定义）。

M8E 02.11-1

引言

读者对象	本手册适用于那些希望了解 V850ES/HG2 产品的功能并准备利用其进行应用系统开发的用户。
目的	本手册用于帮助用户了解下面 组件 中描述的 V850ES/HG2 硬件功能。
组件	V850ES/HG2 产品手册主要分为两个部分： 硬件（本手册）和架构（ V850ES 架构用户手册 ）。

硬件

- 引脚功能
- CPU 功能
- 片上外围功能
- Flash 存储器编程
- 电气特性

架构

- 数据类型
- 寄存器设置
- 指令格式和指令集
- 中断和异常
- 流水线操作

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

如何获悉 V850ES/HG2 的全部功能
→ 请依照**目录**阅读本手册。

如何获取某寄存器的详细信息
→ 请参考**附录 B 寄存器索引**。

如何了解指令的详细功能
→ 请参阅**V850ES 架构用户手册**。

如何了解 V850ES/HG2 产品的电气指标
→ 请参考**第二十六章 电器特性**。

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

文档较前一版本的主要修改之处用<R> 符号标注。在 PDF 文件中的“查找栏”中复制“<R>”，就可以很容易地找到修改之处。

规定

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

相关文档

本手册中指出的相关文档包括了最初的版本，但未注明。

V850ES/HG2 产品的相关文档

文档名称	文档编号
V850ES 架构用户手册	U15943E
V850ES/HG2 硬件用户手册	本手册

开发工具相关的文档

文档名称	文档编号	
QB-V850MINI 片上调试仿真器	U17638E	
QB-MINI2 带编程功能的片上调试仿真器	U18371E	
CA850 Ver. 3.00 C 编译器安装包	操作	U17293E
	C 语言	U17291E
	汇编语言	U17292E
	链接伪指令	U17294E
PM+ Ver. 6.20 工程管理器	U17990E	
ID850QB Ver. 3.20 集成调试器	操作	U17964E
SM850 Ver. 2.50 系统仿真器	操作	U16218E
SM850 Ver. 2.00 或更高版系统仿真器	开放式外部用户接口规范	U14873E
SM+系统仿真器	操作	U17246E
	用户开放接口	U17247E
RX850 Ver. 3.20 实时操作系统	基础	U13430E
	安装	U17419E
	技术	U13431E
	任务调试器	U17420E
RX850 Pro Ver. 3.20 实时操作系统	基础	U13773E
	安装	U17421E
	技术	U13772E
	任务调试器	U17422E
AZ850 Ver. 3.30 系统性能分析器	U17423E	
PG-FP4 Flash 存储器编程器	U15260E	

目录

第一章	绪论	17
1.1	概要	17
1.2	性能	19
1.3	应用领域	19
1.4	订货信息	20
1.5	引脚图（俯视图）	21
1.6	功能模块配置	23
1.6.1	内部结构框图	23
1.6.2	内部单元	24
第二章	引脚功能	26
2.1	引脚功能列表	26
2.2	引脚功能描述	32
2.3	I/O 引脚电路和不使用引脚的推荐连接法	39
2.4	引脚 I/O 电路	41
2.5	注意事项	42
第三章	CPU 功能	43
3.1	特点	43
3.2	CPU 寄存器组	44
3.2.1	程序寄存器组	45
3.2.2	系统寄存器组	46
3.3	操作模式	52
3.3.1	指定操作模式	52
3.4	地址空间	53
3.4.1	CPU 地址空间	53
3.4.2	CPU 地址空间的回绕	54
3.4.3	存储器映射	55
3.4.4	区域	57
3.4.5	地址空间的推荐使用方法	59
3.4.6	外围 I/O 寄存器	62
3.4.7	特殊寄存器	71
3.4.8	注意事项	75
第四章	端口功能	78
4.1	特点	78
4.2	端口的基本配置	78
4.3	端口功能	80
4.3.1	端口功能的操作	80
4.3.2	关于设置端口引脚的注释	81
4.3.3	端口 0	82
4.3.4	端口 1	86
4.3.5	端口 3	89
4.3.6	端口 4	95
4.3.7	端口 5	98
4.3.8	端口 7	104

4.3.9	端口 9	106
4.3.10	端口 CM	114
4.3.11	端口 CS	116
4.3.12	端口 CT	118
4.3.13	端口 DL	120
4.3.14	复用为片上调试功能的端口引脚	122
4.3.15	当使用端口引脚作为复用功能引脚时的寄存器设置	123
4.4	端口框图	128
4.5	注意事项	154
4.5.1	设置端口引脚的注意事项	154
第五章	时钟发生器功能	155
5.1	概述	155
5.2	结构图	156
5.3	寄存器	158
5.4	操作	163
5.4.1	各时钟的操作	163
5.4.2	时钟输出功能	163
5.5	PLL 功能	164
5.5.1	概述	164
5.5.2	寄存器	164
5.5.3	使用方法	168
第六章	16 位定时器/事件计数器 P (TMP)	169
6.1	概述	169
6.2	功能	169
6.3	配置	170
6.4	寄存器	172
6.5	操作	186
6.5.1	间隔定时器模式 (TPnMD2 ~ TPnMD0 位 = 000)	187
6.5.2	外部事件计数模式 (TPnMD2 ~ TPnMD0 位 = 001)	197
6.5.3	外部触发脉冲输出模式 (TPnMD2 ~ TPnMD0 位 = 010)	205
6.5.4	单脉冲输出模式 (TPnMD2 ~ TPnMD0 位 = 011)	217
6.5.5	PWM 输出模式 (TPnMD2 ~ TPnMD0 位 = 100)	224
6.5.6	自由运行定时器模式 (TPnMD2 ~ TPnMD0 位 = 101)	233
6.5.7	脉宽测量模式 (TPnMD2 ~ TPnMD0 位 = 110)	250
6.5.8	定时器输出操作	256
6.6	定时器调谐操作功能	257
6.7	选择器功能	261
6.8	注意事项	263
第七章	16 位定时器/事件计数器 Q (TMQ)	264
7.1	概述	264
7.2	功能	264
7.3	配置	265
7.4	寄存器	268
7.5	操作	286
7.5.1	间隔定时器模式 (TQnMD2 ~ TQnMD0 位 = 000)	287
7.5.2	外部事件计数模式 (TQnMD2 ~ TQnMD0 位 = 001)	296

7.5.3	外部触发脉冲输出模式 (TQnMD2 ~ TQnMD0 位 = 010)	305
7.5.4	单次脉冲输出模式 (TQnMD2 ~ TQnMD0 位 = 011)	318
7.5.5	PWM 输出模式 (TQnMD2 ~ TQnMD0 位 = 100)	327
7.5.6	自由运行定时器模式 (TQnMD2 ~ TQnMD0 位 = 101)	338
7.5.7	脉宽测量模式 (TQnMD2 ~ TQnMD0 位 = 110)	358
7.5.8	三角波 PWM 模式 (TQnMD2 ~ TQnMD0 位 = 111)	364
7.5.9	定时器输出操作	365
7.6	定时器调谐操作功能	366
7.7	注意事项	370
第八章	16 位间隔定时器 M (TMM)	371
8.1	概述	371
8.2	配置	372
8.3	寄存器	373
8.4	操作	374
8.4.1	间隔定时器模式	374
8.4.2	注意事项	378
第九章	钟表定时器功能	379
9.1	功能	379
9.2	配置	380
9.3	寄存器	382
9.4	操作	386
9.4.1	钟表定时器的操作	386
9.4.2	间隔定时器的操作	387
9.4.3	注意事项	388
第十章	看门狗定时器 2 的功能	389
10.1	功能	389
10.2	配置	390
10.3	寄存器	391
10.4	操作	394
第十一章	A/D 转换器	395
11.1	概述	395
11.2	功能	395
11.3	配置	396
11.4	寄存器	399
11.5	操作	407
11.5.1	基本操作	407
11.5.2	触发模式	408
11.5.3	操作模式	410
11.5.4	掉电比较模式	414
11.6	注意事项	419
11.7	A/D 转换器特征表的阅读方法	423
第十二章	异步串行接口 A (UARTA)	427
12.1	特点	427
12.2	配置	428

12.3	寄存器.....	430
12.4	中断请求信号	436
12.5	操作.....	437
12.5.1	数据格式	437
12.5.2	SBF 发送/接收格式	439
12.5.3	SBF 发送	441
12.5.4	SBF 接收	442
12.5.5	UART 发送.....	443
12.5.6	连续发送.....	444
12.5.7	UART 接收.....	446
12.5.8	接收错误	447
12.5.9	校验位的类型和操作	449
12.5.10	接收数据的噪声过滤器.....	450
12.6	专用波特率发生器	451
12.7	注意事项.....	459
第十三章	3 线可变长度串行 I/O (CSIB).....	460
13.1	特点.....	460
13.2	配置.....	461
13.3	寄存器.....	463
13.4	中断请求信号	470
13.5	操作.....	471
13.5.1	单传输模式（主模式, 发送模式）	471
13.5.2	单传输模式（主模式, 接收模式）	473
13.5.3	单传输模式（主模式, 发送/接收模式）	475
13.5.4	单传输模式（从模式, 发送模式）	477
13.5.5	单传输模式（从模式, 接收模式）	479
13.5.6	单传输模式（从模式, 发送/接收模式）	481
13.5.7	连续传输模式（主模式, 发送模式）	483
13.5.8	连续传输模式（主模式, 接收模式）	485
13.5.9	连续传输模式（主模式, 发送/接收模式）	488
13.5.10	连续传输模式（从模式, 发送模式）	492
13.5.11	连续传输模式（从模式, 接收模式）	494
13.5.12	连续传输模式（从模式, 发送/接收模式）	497
13.5.13	接收错误	501
13.5.14	时钟时序.....	502
13.6	操作禁用状态输出引脚.....	504
13.7	波特率发生器	505
13.7.1	波特率发生器	506
13.8	注意事项.....	507
第十四章	DMA 功能（DMA 控制器）	508
14.1	特点.....	508
14.2	配置.....	509
14.3	寄存器.....	510
14.4	传送目标.....	518
14.5	传送模式.....	518
14.6	传送类型.....	519
14.7	DMA 通道优先级	520

14.8	DMA 传送的相关时间.....	520
14.9	DMA 传送启动系数.....	521
14.10	DMA 中止因素.....	522
14.11	DMA 传送结束.....	522
14.12	操作时序.....	522
14.13	注意事项.....	527
第十五章	中断/异常处理功能	531
15.1	特点.....	531
15.2	非屏蔽中断.....	535
15.2.1	操作.....	537
15.2.2	恢复.....	538
15.2.3	NP 标志	539
15.3	可屏蔽中断.....	540
15.3.1	操作.....	540
15.3.2	恢复.....	542
15.3.3	可屏蔽中断的优先级	543
15.3.4	中断控制寄存器 (xxICn).....	547
15.3.5	中断屏蔽寄存器 0~3 (IMR0~ IMR3).....	549
15.3.6	当前服务的优先级寄存器(ISPR)	551
15.3.7	ID 标志.....	552
15.3.8	看门狗定时器模式寄存器 2 (WDTM2).....	552
15.4	软件异常.....	553
15.4.1	操作.....	553
15.4.2	恢复.....	554
15.4.3	标志.....	555
15.5	异常陷阱.....	556
15.5.1	非法操作数定义	556
15.5.2	调试陷阱	558
15.6	外部中断请求输入引脚(NMI 和 INTPO ~ INTPI0).....	560
15.6.1	噪声消除.....	560
15.6.2	边沿检测.....	560
15.7	CPU 的中断确认时间	567
15.8	CPU 不对中断进行确认的时期.....	568
15.9	注意事项.....	568
第十六章	按键中断功能.....	569
16.1	功能.....	569
16.2	寄 存 器	57
0		
16.3	注意事项.....	570
第十七章	待机功能.....	571
17.1	概述	571
17.2	寄存器.....	573
17.3	HALT 模式	576
17.3.1	设置和操作状态	576
17.3.2	释放 HALT 模式	576

17.4	IDLE1 模式	578
17.4.1	设置和操作状态	578
17.4.2	释放 IDLE1 模式	578
17.5	IDLE2 模式	580
17.5.1	设置和操作状态	580
17.5.2	释放 IDLE2 模式	580
17.5.3	当释放 IDLE2 模式时保证设置时间	582
17.6	STOP 模式	583
17.6.1	设置和操作状态	583
17.6.2	释放 STOP 模式.....	583
17.6.3	当释放 STOP 模式时保证振荡稳定时间	585
17.7	副时钟操作模式	586
17.7.1	设置和操作状态	586
17.7.2	释放副时钟操作模式.....	586
17.8	副 IDLE 模式	588
17.8.1	设置和操作状态	588
17.8.2	释放副 IDLE 模式	589
第十八章	复位功能	591
18.1	概述.....	591
18.2	检测复位源的寄存器	592
18.3	操作.....	593
18.3.1	通过 RESET 引脚的复位操作.....	593
18.3.2	由看门狗定时器 2 引起的复位操作.....	595
18.3.3	上电清零电路的复位操作	596
18.3.4	低电压检测器引起的复位操作	596
18.3.5	时钟监视器的复位操作.....	596
<R> 18.4	复位释放后的操作	597
第十九章	时钟监视器	599
19.1	功能.....	599
19.2	配置.....	599
19.3	寄存器.....	600
19.4	操作.....	601
第二十章	开机清除电路	604
20.1	功能.....	604
20.2	配置.....	604
20.3	运行.....	605
第二十一章	低电压检测器	606
21.1	功能.....	606
21.2	配置.....	606
21.3	寄存器.....	607
21.4	操作.....	609
21.4.1	使用内部复位信号.....	609
21.4.2	用作中断	611
21.5	RAM 保持电压检测操作	612
21.6	仿真功能.....	613

	第二十二章 稳压器	614
	22.1 概述	614
	22.2 操作	615
	第二十三章 FLASH 存储器	616
	23.1 特点	616
<R>	23.2 内存配置	617
<R>	23.3 功能概述	618
	23.4 通过专门的 Flash 存储器编程器重写	621
	23.4.1 编程环境	621
	23.4.2 通讯模式	622
	23.4.3 Flash 存储器控制	627
	23.4.4 通讯模式的选择	628
	23.4.5 通讯命令	629
	23.4.6 引脚连接	630
	23.5 通过自编程重写	634
	23.5.1 概述	634
	23.5.2 特点	635
	23.5.3 标准自编程流程	636
	23.5.4 Flash 存储器功能	637
	23.5.5 引脚处理	637
	23.5.6 使用的内部资源	638
	第二十四章 选项字节功能	639
	第二十五章 片上调试功能	641
	25.1 使用 DCU 调试	642
	25.1.1 电路连接示例	642
	25.1.2 接口信号	642
	25.1.3 可屏蔽功能	644
	25.1.4 寄存器	644
	25.1.5 操作	646
	25.1.6 注意事项	647
	25.2 不使用 DCU 调试	648
	25.2.1 电路连接示例	648
	25.2.2 可屏蔽功能	649
	25.2.3 保护用户资源	650
	25.2.4 注意事项	656
	25.3 ROM 安全功能	657
	25.3.1 安全 ID	657
	25.3.2 设置	658
	第二十六章 电气特性	660
	26.1 最大额定值	660
	26.2 电容	662
	26.3 操作条件	662
	26.4 振荡器特性	663
	26.4.1 主时钟振荡器特性	663

26.4.2	副时钟振荡器特性.....	664
26.4.3	PLL 特性.....	665
26.4.4	内部振荡器特性.....	665
26.5	电压调节器特性.....	665
26.6	DC 特性.....	666
26.6.1	I/O 电平.....	666
26.6.2	引脚漏电流.....	667
26.6.3	电源电流.....	668
26.7	数据保持特性.....	669
26.8	AC 特性.....	670
26.8.1	CLKOUT 输出时序.....	671
26.9	基本操作.....	672
26.10	Flash 存储器编程特性.....	679
第二十七章	封装图.....	680
<R>	第二十八章 推荐焊接条件.....	681
<R>	附录 A 开发工具.....	682
A.1	软件包.....	684
A.2	语言处理软件.....	684
A.3	控制软件.....	684
A.4	调试工具 (硬件).....	685
A.4.1	当使用 IECUBE QB-V850ESFX2 时.....	685
A.4.2	当使用 MINICUBE QB-V850MINI 时.....	687
A.4.3	当使用 MINICUBE2 QB-MINI2 时.....	688
A.5	调试工具 (软件).....	689
A.6	嵌入式软件.....	690
A.7	Flash 存储器写入工具.....	691
附录 B	寄存器索引.....	692
附录 C	指令集列表.....	701
C.1	常规指令.....	701
C.2	指令集 (按字母顺序).....	704
<R>	附录 D 注意事项列表.....	711
<R>	附录 E 修订履历.....	744
E.1	在此版本中的主要修订.....	744

第一章 绪论

V850ES/HG2 是日电电子公司(NEC Electronics)的 V850 系列单片微控制器产品之一，主要应用在实时控制应用中的低功耗操作。

1.1 概要

V850ES/HG2 是一款 32 位单片微控制器，它包含了 V850ES CPU 内核以及诸如 ROM/RAM、定时器/计数器、串行接口和 A/D 转换器等的外围设备功能。

除了实时响应特性和基本指令单时钟执行特点之外，V850ES/HG2 还具有由硬件乘法器来实现的乘法指令、饱和运算指令、位操作指令等为数字伺服控制应用而优化的指令。

表 1-1 列出了 V850ES/HG2 系列的所有产品。

表 1-1. V850ES/HG2 产品列表

产品代号		μ PD70F3706	μ PD70F3707
内存	Flash 存储器	128 KB	256 KB
	RAM	12 KB	
存储空间	逻辑空间	64 MB	
通用寄存器		32 位 \times 32 寄存器	
主时钟 (振荡频率)		陶瓷/晶体/外部时钟 • PLL 模式: $f_x = 4 \sim 5$ MHz • 时钟直通模式: $f_x = 4 \sim 5$ MHz	
副时钟 (振荡频率)		晶体/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.)	
最短指令执行时间		50 ns (工作在主时钟 (f_{XX}) = 20 MHz 下)	
DSP 功能		32 \times 32 = 64: 200 ~ 250 ns (20 MHz) 32 \times 32 + 32 = 32: 300 ns (20 MHz) 16 \times 16 = 32: 50 ~ 100 ns (20 MHz) 16 \times 16 + 32 = 32: 150 ns (20 MHz)	
I/O 端口		I/O: 84	
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: 2 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 16 通道	
串行接口		CSIB: 2 通道 UARTA (支持 LIN): 3 通道	
DMA 控制器		4 通道 (传送目标: 片上外围 I/O, 内部 RAM)	
中断源		外部: 12 (12) [#] , 内部: 43	
节电功能		HALT/IDLE1/IDLE2/STOP/副时钟/副 IDLE 模式	
复位		$\overline{\text{RESET}}$ 引脚输入, 看门狗定时器 2 (WDT2), 时钟监视器 (CLM), 上电清零电路, 低电压探测器 (LVI)	
DCU		提供 (运行/中断)	
操作电源电压		3.5 ~ 5.5 V (A/D 转换器: 4.0 ~ 5.5 V)	
工作环境温度		-40 ~ +85°C	
封装		100 引脚塑料 LQFP (密脚距) (14 \times 14 mm)	

注 括号中的数据表示可以释放 STOP 模式的外部中断的个数。

1.2 性能

- 最短指令执行时间: 50 ns (工作在主时钟(f_{xx}) 为 20 MHz 下)
- 通用寄存器: 32 位 × 32 寄存器
- CPU 特性:
 - 带符号乘法 ($16 \times 16 \rightarrow 32$): 1 ~ 2 时钟周期
 - 带符号乘法($32 \times 32 \rightarrow 64$): 1 ~ 5 时钟周期
 - 饱和操作 (包含上溢和下溢的检测功能)
 - 32 位移位指令: 1 时钟周期
 - 位操作指令
 - 区分长/短格式的载入/存储指令
- 存储器空间: 为程序和数据提供 64 MB 的线性地址空间
 - 内部存储器:
 - RAM: 12 KB
 - Flash 存储器: 128 KB/256 KB (参见 表 1-1)
- 中断和异常:
 - 不可屏蔽中断: 2 个
 - 可屏蔽中断: 53 个
 - 软件异常: 32 个
 - 异常陷阱: 2 个
- I/O 线: I/O 端口: 84
- 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 (TMP): 4 通道
 - 16 位定时器/事件计数器 Q (TMQ): 2 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 线可变长度串行接口 B (CSIB)
 - UARTA (支持 LIN): 3 通道
 - CSIB: 2 通道
- A/D 转换器: 10 位分辨率: 16 通道
- DMA 控制器: 4 通道
- DCU (调试控制单元): JTAG 接口
- 时钟发生器:
 - 主时钟或副时钟操作期间
 - 可选 7 个级别的 CPU 时钟 (f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/8$, $f_{xx}/16$, $f_{xx}/32$, f_{xt})
 - 时钟直通模式/PLL 模式
- 内部振荡时钟: 200 kHz (TYP.)
- 节电功能: HALT/IDLE1/IDLE2/STOP/副时钟/副 IDLE 模式
- 封装: 100 引脚塑料 LQFP (密脚距) (14×14)

1.3 应用领域

消费电子设备

1.4 订货信息

产品型号	封装	片上 Flash 存储器
μ PD70F3706GC-8EA-A	100 引脚塑料 LQFP (密脚距) (14 × 14)	128 KB
μ PD70F3707GC-8EA-A	100 引脚塑料 LQFP (密脚距) (14 × 14)	256 KB

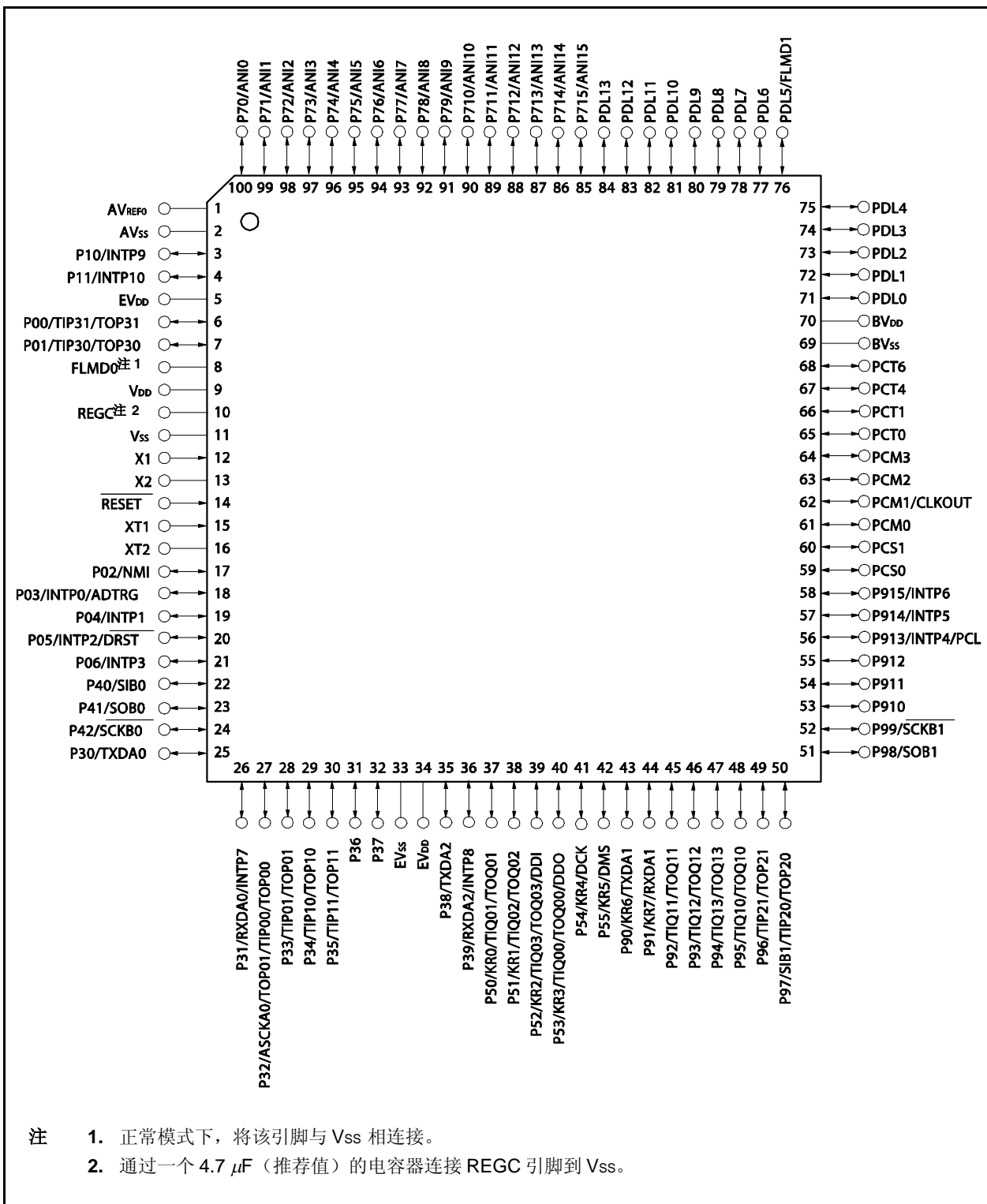
备注 代号末尾带 A 的产品属于无铅产品。

1.5 引脚图 (俯视图)

100 引脚塑料 LQFP (密脚距) (14 × 14)

μPD70F3706GC-8EA-A

μPD70F3707GC-8EA-A

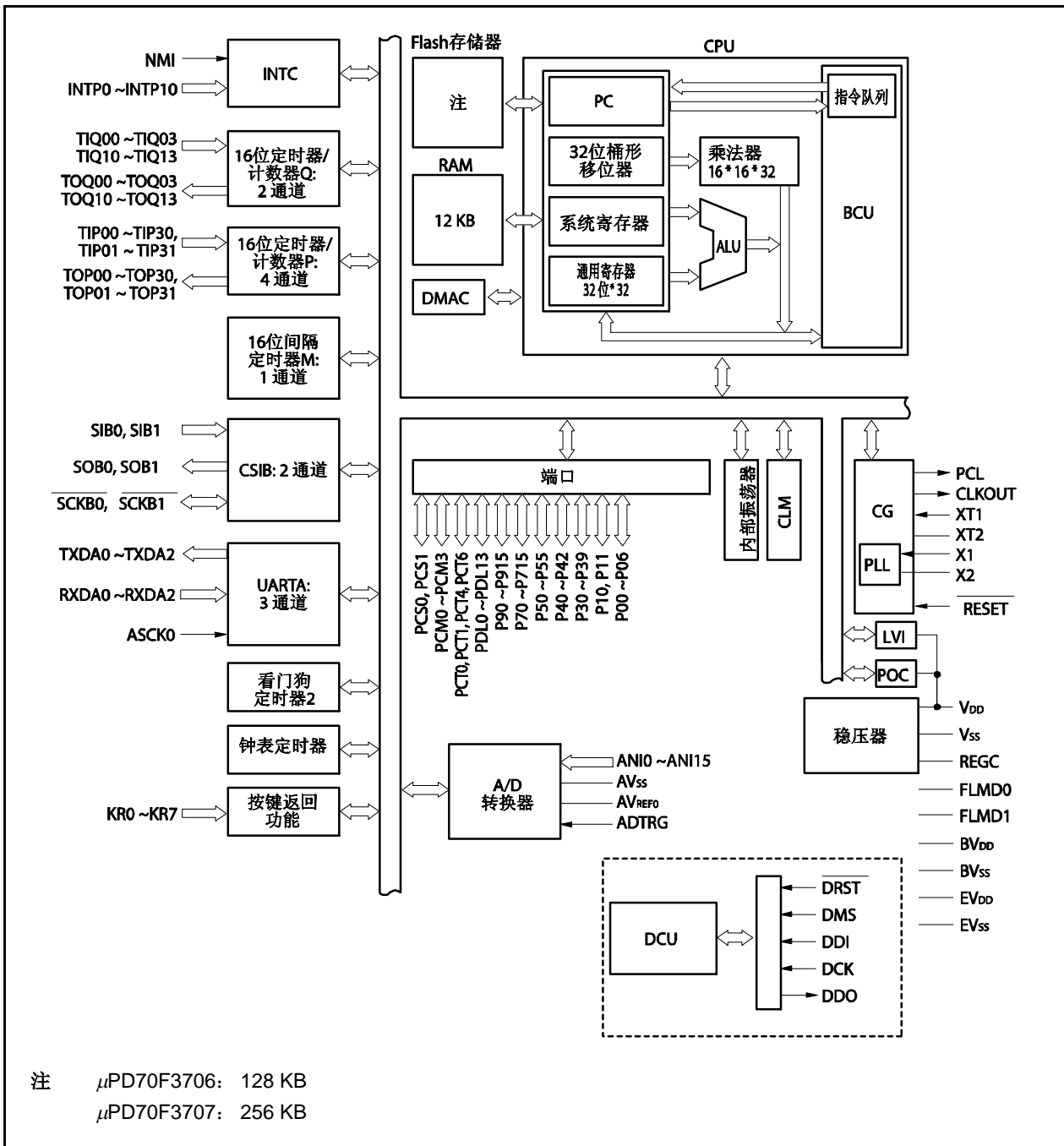


引脚名称

ADTRG:	A/D 触发输入	PCS0, PCS1:	端口 CS
ANI0 ~ ANI15:	模拟输入	PCT0, PCT1,	
ASCKA0:	异步串行时钟	PCT4, PCT6:	端口 CT
AVREF0:	模拟参考电压	PDL0 ~ PDL13:	端口 DL
AVSS:	模拟 Vss	REGC:	稳压器控制
BVDD:	总线接口电源	RESET:	复位
BVSS:	总线接口地	RXDA0 ~ RXDA2:	接收数据
CLKOUT:	时钟输出	SCKB0, SCKB1:	串行时钟
DCK:	调试时钟	SIB0, SIB1:	串行输入
DDI:	调试数据输入	SOB0, SOB1:	串行输出
DDO:	调试数据输出	TIP00, TIP01,	
DMS:	调试模式选择	TIP10, TIP11,	
DRST:	调试复位 t	TIP20, TIP21,	
EVDD:	端口电源	TIP30, TIP31,	
EVSS:	端口地	TIQ00 ~ TIQ03,	
FLMD0, FLMD1:	Flash 程序模式	TIQ10 ~ TIQ13:	定时器输入
INTP0 ~ INTP10:	外部中断输入	TOP00, TOP01,	
KR0 ~ KR7:	按键返回	TOP10, TOP11,	
NMI:	不可屏蔽中断请求	TOP20, TOP21,	
P00 ~ P06:	端口 0	TOP30, TOP31,	
P10, P11:	端口 1	TOQ00 ~ TOQ03,	
P30 ~ P39:	端口 3	TOQ10 ~ TOQ13:	定时器输出
P40 ~ P42:	端口 4	TXDA0 ~ TXDA2:	发送数据
P50 ~ P55:	端口 5	VDD:	电源
P70 ~ P715:	端口 7	VSS:	地
P90 ~ P915:	端口 9	X1, X2:	主时钟晶振
PCL:	可编程时钟输出	XT1, XT2:	副时钟晶振
PCM0 ~ PCM3:	端口 CM		

1.6 功能模块配置

1.6.1 内部结构框图



1.6.2 内部单元

(1) CPU

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

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

(2) 总线控制单元 (BCU)

BCU 控制内部总线。

(3) ROM

V850ES/HG2 产品提供了 256 KB 或 128 KB 的 flash 存储器，映射地址为 0000000H ~ 003FFFFH 或 0000000H ~ 001FFFFH。在取指令过程中，该存储器可由 CPU 在一个时钟周期内访问。

(4) RAM

V850ES/HG2 产品提供了 12 KB 的 RAM，映射地址为 3FFC000H ~ 3FFEFFFH。在数据访问过程中，该 RAM 可由 CPU 在一个时钟周期内访问。

(5) 中断控制器 (INTC)

该控制器用来处理由片上外围硬件和外部硬件发出的硬件中断请求（NMI，INTP0 ~ INTP10）。可以对这些中断请求指定 8 个级别的中断优先级，同时也可进行多重中断服务的处理。

(6) 时钟发生器 (CG)

V850ES/HG2 产品提供了一个产生主时钟振荡频率（ f_x ）的主时钟振荡器及一个产生副时钟振荡频率（ f_{XT} ）的副时钟振荡器，作为主时钟频率（ f_{xx} ）， f_x 用于时钟直通模式并且在 PLL 模式中其值要乘以 4。

CPU 时钟频率（ f_{CPU} ）可从这七种类型中选择： f_{xx} ， $f_{xx}/2$ ， $f_{xx}/4$ ， $f_{xx}/8$ ， $f_{xx}/16$ ， $f_{xx}/32$ 和 f_{XT} 。

(7) 内部振荡器

片上集成了一个内部振荡器。该振荡器的频率是 200 kHz (TYP.)。内部振荡器的振荡频率将提供给看门狗定时器 2 和定时器 M 作为时钟信号。

(8) 定时器与计数器

V850ES/HG2 产品提供了片上四通道的 16 位定时器/事件计数器 P (TMP)，双通道的 16 位定时器/事件计数器 Q (TMQ)，和单通道 16 位间隔定时器 M (TMM)。

(9) 钟表定时器

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

(10) 看门狗定时器 2

V850ES/HG2 产品提供了片上看门狗定时器，该定时器用于检测意外的程序循环，系统异常等错误状态。内部振荡器或主时钟可以被选作为时钟源。

在发生溢出后，看门狗定时器 2 将生成一个不可屏蔽的中断请求信号 (INTWDT2)或系统复位信号 (WDT2RES)。

(11) 串行接口

V850ES/HG2 包括两种形式的串行接口：异步串行接口 A (UARTA) 和 3 线可变长度串行接口 B (CSIB)。

使用 UARTA 时，数据通过 TXDA0 ~ TXDA2 和 RXDA0 ~ RXDA2 引脚传输。

使用 CSIB 时，数据通过 SOB0, SOB1, SIB0, SIB1, $\overline{\text{SCKB0}}$ 和 $\overline{\text{SCKB1}}$ 引脚传输。

(12) A/D 转换器

该 10 位 A/D 转换器包括 16 个模拟输入引脚。采用逐次逼近法进行转换。

(13) DMA 控制器

V850ES/HG2 产品提供了片上 4 通道 DMA 控制器。为了响应通过片上外围 I/O 发送的中断请求，该寄存器在内部 RAM 和片上外围 I/O 设备之间传输数据。

(14) 按键中断功能

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

(15) DCU（调试控制单元）

V850ES/HG2 提供了可以使用 JTAG（联合测试行动小组）通讯规范的片上调试功能，使用控制引脚输入电平及片上调试模式寄存器 (OCDM) 完成正常端口功能及片上调试功能之间的切换。

(16) 端口

V850ES/HG2 产品提供了通用端口功能和控制引脚功能。关于更多细节，敬请参阅第四章 端口功能。

第二章 引脚功能

该部分讲解了 V850ES/HG2 的引脚名称和功能。

2.1 引脚功能列表

可用引脚 I/O 缓冲式供电电源有三种：AVREF0，BVDD 和 EVDD。这些电源与引脚的关系如下所示。

表 2-1. 引脚 I/O 缓冲式供电电源

供电电源	对应引脚
AVREF0	端口 7
BVDD	端口 CM，端口 CS，端口 CT，端口 DL
EVDD	端口 0，端口 1，端口 3，端口 4，端口 5，端口 9，RESET

(1) 端口引脚

表 2-2. 引脚列表（端口引脚）(1/3)

引脚名称	引脚编号	I/O	功能	复用功能
P00	6	I/O	端口 0 7 位 I/O 端口 可以位选输入/输出模式。	TIP31/TOP31
P01	7			TIP30/TOP30
P02	17			NMI
P03	18			INTP0/ADTRG
P04	19			INTP1
P05	20			INTP2/DRST
P06	21			INTP3
P10	3	I/O	端口 1 2 位 I/O 端口 可以位选输入/输出模式。	INTP9
P11	4			INTP10
P30	25	I/O	端口 3 10 位 I/O 端口 可以位选输入/输出模式。	TXDA0
P31	26			RXDA0/INTP7
P32	27			ASCKA0/TIP00/TOP00/TOP01
P33	28			TIP01/TOP01
P34	29			TIP10/TOP10
P35	30			TIP11/TOP11
P36	31			-
P37	32			-
P38	35			TXDA2
P39	36			RXDA2/INTP8

表 2-2. 引脚列表（端口引脚）(2/3)

引脚名称	引脚编号	I/O	功能	复用功能
P40	22	I/O	端口 4 3 位 I/O 端口 可以位选输入/输出模式。	SIB0
P41	23			SOB0
P42	24			SCKB0
P50	37	I/O	端口 5 6 位 I/O 端口 可以位选输入/输出模式。	KR0/TIQ01/TOQ01
P51	38			KR1/TIQ02/TOQ02
P52	39			KR2/TIQ03/TOQ03/DDI
P53	40			KR3/TIQ00/TOQ00/DDO
P54	41			KR4/DCK
P55	42			KR5/DMS
P70	100			I/O
P71	99	ANI1		
P72	98	ANI2		
P73	97	ANI3		
P74	96	ANI4		
P75	95	ANI5		
P76	94	ANI6		
P77	93	ANI7		
P78	92	ANI8		
P79	91	ANI9		
P710	90	ANI10		
P711	89	ANI11		
P712	88	ANI12		
P713	87	ANI13		
P714	86	ANI14		
P715	85	ANI15		
P90	43	I/O	端口 9 16 位 I/O 端口 可以位选输入/输出模式。	KR6/TXDA1
P91	44			KR7/RXDA1
P92	45			TIQ11/TOQ11
P93	46			TIQ12/TOQ12
P94	47			TIQ13/TOQ13
P95	48			TIQ10/TOQ10
P96	49			TIP21/TOP21
P97	50			SIB1/TIP20/TOP20
P98	51			SOB1
P99	52			SCKB1
P910	53			-
P911	54			-
P912	55			-
P913	56			INTP4/PCL
P914	57			INTP5
P915	58	INTP6		

表 2-2. 引脚列表（端口引脚）(3/3)

引脚名称	引脚编号	I/O	功能	复用功能
PCM0	61	I/O	端口 CM 4 位 I/O 端口 可以位选输入/输出模式。	-
PCM1	62			CLKOUT
PCM2	63			-
PCM3	64			-
PCS0	59	I/O	端口 CS 2 位 I/O 端口 可以位选输入/输出模式。	-
PCS1	60			-
PCT0	65	I/O	端口 CT 4 位 I/O 端口 可以位选输入/输出模式。	-
PCT1	66			-
PCT4	67			-
PCT6	68			-
PDL0	71	I/O	端口 DL 14 位 I/O 端口 可以位选输入/输出模式。	-
PDL1	72			-
PDL2	73			-
PDL3	74			-
PDL4	75			-
PDL5	76			FLMD1
PDL6	77			-
PDL7	78			-
PDL8	79			-
PDL9	80			-
PDL10	81			-
PDL11	82			-
PDL12	83			-
PDL13	84			-

(2) 非端口引脚

表 2-3. 引脚列表 (非端口引脚) (1/3)

引脚名称	引脚编号	I/O	功能	复用功能	
NMI [※]	17	输入	外部中断输入 (不可屏蔽, 模拟噪声消除电路)	P02	
INTP0	18	输入	外部中断请求输入 (可屏蔽, 模拟噪声消除电路)	P03/ADTRG	
INTP1	19			P04	
INTP2	20			P05/DRST	
INTP3	21			P06	
INTP4	56			P913/PCL	
INTP5	57			P914	
INTP6	58			P915	
INTP7	26			P31/RXDA0	
INTP8	36			P39/RXDA2	
INTP9	3			P10	
INTP10	4			P11	
TIP00	27	输入	外部事件/时钟输入 (TMP00)	P32/ASCKA0/TOP00/TOP01	
TIP01	28		外部事件输入 (TMP01)	P33/TOP01	
TIP10	29		外部事件/时钟输入 (TMP10)	P34/TOP10	
TIP11	30		外部事件输入 (TMP11)	P35/TOP11	
TIP20	50		外部事件/时钟输入 (TMP20)	P97/SIB1/TOP20	
TIP21	49		外部事件输入 (TMP21)	P96/TOP21	
TIP30	7		外部事件/时钟输入 (TMP30)	P01/TOP30	
TIP31	6		外部事件输入 (TMP31)	P00/TOP31	
TOP00	27		输出	定时器输出(TMP00)	P32/ASCKA0/TIP00/TOP01
TOP01	27	定时器输出(TMP01)		P32/ASCKA0/TIP00/TOP00	
	28			P33/TIP01	
TOP10	29	定时器输出(TMP10)		P34/TIP10	
TOP11	30	定时器输出(TMP11)		P35/TIP11	
TOP20	50	定时器输出(TMP20)		P97/SIB1/TIP20	
TOP21	49	定时器输出(TMP21)		P96/TIP21	
TOP30	7	定时器输出(TMP30)		P01/TIP30	
TOP31	6	定时器输出(TMP31)		P00/TIP31	
TIQ00	40	输入		外部事件/时钟输入 (TMQ00)	P53/KR3/TOQ00/DDO
TIQ01	37			外部事件输入 (TMQ01)	P50/KR0/TOQ01
TIQ02	38		外部事件输入 (TMQ02)	P51/KR1/TOQ02	
TIQ03	39		外部事件输入 (TMQ03)	P52/KR2/TOQ03/DDI	
TIQ10	48		外部事件/时钟输入 (TMQ10)	P95/TOQ10	

注 NMI 引脚可功能复用为 P02 引脚。复位后, 引脚功能为 P02 引脚。要允许 NMI 引脚, 设置 PMC0.PMC02 位为 1。NMI 引脚的初始设置为“无边沿检测”。使用 INTF0 和 INTR0 寄存器选择 NMI 引脚的有效沿。

表 2-3. 引脚列表 (非端口引脚) (2/3)

引脚名称	引脚编号	I/O	功能	复用功能	
TIQ11	45	输入	外部事件输入 (TMQ11)	P92/TOQ11	
TIQ12	46		外部事件输入 (TMQ12)	P93/TOQ12	
TIQ13	47		外部事件输入 (TMQ13)	P94/TOQ13	
TOQ00	40	输出	定时器输出(TMQ00)	P53/KR3/TIQ00/DDO	
TOQ01	37		定时器输出(TMQ01)	P50/KR0/TIQ01	
TOQ02	38		定时器输出(TMQ02)	P51/KR1/TIQ02	
TOQ03	39		定时器输出(TMQ03)	P52/KR2/TIQ03/DDI	
TOQ10	48		定时器输出(TMQ10)	P95/TIQ10	
TOQ11	45		定时器输出(TMQ11)	P92/TIQ11	
TOQ12	46		定时器输出(TMQ12)	P93/TIQ12	
TOQ13	47		定时器输出(TMQ13)	P94/TIQ13	
SIB0	22		输入	串行接收数据输入 (CSIB0)	P40
SIB1	50			串行接收数据输入 (CSIB1)	P97/TIP20/TOP20
SOB0	23		输出	串行发送数据输出 (CSIB0)	P41
SOB1	51	串行发送数据输出 (CSIB1)		P98	
$\overline{\text{SCKB0}}$	24	I/O	串行时钟 I/O (CSIB0)	P42	
$\overline{\text{SCKB1}}$	52		串行时钟 I/O (CSIB1)	P99	
RXDA0	26	输入	串行接收数据输入 (UARTA0)	P31/INTP7	
RXDA1	44		串行接收数据输入 (UARTA1)	P91/KR7	
RXDA2	36		串行接收数据输入 (UARTA2)	P39/INTP8	
TXDA0	25	输出	串行发送数据输出 (UARTA0)	P30	
TXDA1	43		串行发送数据输出 (UARTA1)	P90/KR6	
TXDA2	35		串行发送数据输出 (UARTA2)	P38	
ASCKA0	27	输入	波特率时钟输入到 UARTA0	P32/TIP00/TOP00/TOP01	
ANI0	100	输入	模拟电压输入到 A/D 转换器	P70	
ANI1	99			P71	
ANI2	98			P72	
ANI3	97			P73	
ANI4	96			P74	
ANI5	95			P75	
ANI6	94			P76	
ANI7	93			P77	
ANI8	92			P78	
ANI9	91			P79	
ANI10	90			P710	
ANI11	89			P711	
ANI12	88			P712	
ANI13	87			P713	
ANI14	86			P714	
ANI15	85	P715			

表 2-3. 引脚列表 (非端口引脚) (3/3)

引脚名称	引脚编号	I/O	功能	复用功能
AV _{REF0}	1	-	参考电压输入到 A/D 转换器, 端口 7 的正电源	-
AV _{SS}	2	-	A/D 转换器的地电位 (与 V _{SS} 电位相同)	-
ADTRG	18	输入	A/D 转换器外部触发输入	P03/INTP0
KR0	37	输入	按键中断输入	P50/TIQ01/TOQ01
KR1	38			P51/TIQ02/TOQ02
KR2	39			P52/TIQ03/TOQ03/DDI
KR3	40			P53/TIQ00/TOQ00/DDO
KR4	41			P54/DCK
KR5	42			P55/DMS
KR6	43			P90/TXDA1
KR7	44			P91/RXDA1
DMS	42			输入
DDI	39	输入	调试数据输入	P52/KR2/TIQ03/TOQ03
DDO	40	输出	调试数据 输出	P53/KR3/TIQ00/TOQ00
DCK	41	输入	调试时钟输入	P54/KR4
DRST	20	输入	调试复位输入	P05/INTP2
FLMD0	8	输入	Flash 编程模式设置引脚	-
FLMD1	76			PDL5
CLKOUT	62	输出	内部系统时钟输出	PCM1
PCL	56	输出	时钟输出 (X1 输入时钟和副时钟的时序输出)	P913/INTP4
REGC	10	-	为稳定稳压器输出连接电容器	-
RESET	14	输入	系统复位输入	-
X1	12	输入	连接主时钟振荡器	-
X2	13	-		-
XT1	15	输入	连接副时钟振荡器	-
XT2	16	-		-
V _{DD}	9	-	内部电路的正电源引脚	-
V _{SS}	11	-	内部电路的地电位	-
BV _{DD}	70	-	总线接口和复用功能端口的正电源引脚	-
BV _{SS}	69	-	总线接口和复用功能端口的地电位	-
EV _{DD}	5	-	外部电路的正电源引脚 (与 V _{DD} 的电位相同)	-
	34	-		-
EV _{SS}	33	-	外部电路的地电位 (与 V _{SS} 的电位相同)	-

2.2 引脚功能描述

(1) P00 ~ P06 (端口 0) ... 3 状态 I/O

P00~P06 用作一个 7 位 I/O 端口，可以以位为单元中设置为输入或输出。

除了作为 I/O 端口外，这些引脚也用作 NMI 输入，外部中断请求信号输入，定时器/计数器 I/O，A/D 转换器的外部触发器和调试复位输入。

在端口模式或控制模式下，该端口可以以位为单位进行设置。每个引脚的有效沿由 INTR0 和 INTF0 寄存器指定。

通过使用上拉电阻选项寄存器 0 (PU0)，可以将片上上拉电阻连接到 P00 ~ P06。

(a) 端口模式

通过使用端口模式寄存器 0 (PM0)，可以在输入或输出模式下以位为单位设置 P00~P06。

(b) 控制模式

(i) NMI (不可屏蔽中断 请求) ...输入

这些引脚输入不可屏蔽中断请求信号。

(ii) INTP0 ~ INTP3 (外部中断输入) ...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

这些引脚输入一个外部计数时钟到定时器 P3 (TMP3)。

(iv) TOP30, TOP31 (定时器输出) ... 输出

这些引脚输出一个来自定时器 P3 (TMP3)的脉冲信号。

(v) ADTRG (A/D 触发输入) ...输入

这些引脚输入一个外部触发器到 A/D 转换器。通过使用 A/D 转换器模式寄存器 0 (ADA0M0)可以对输入进行控制。

(vi) $\overline{\text{DRST}}$ (调试复位) ...输入

该引脚输入一个调试复位信号，一个可以同步初始化调试控制单元 (DCU) 的负逻辑信号。为了解除该信号，复位或使 DCU 无效，当不使用调试功能时，不解除该信号。

关于更多细节，敬请参阅第二十五章 片上调试功能。

(2) P10, P11 (端口 1) ... 3 状态 I/O

P10 和 P11 用作一个 2 位 I/O 端口，可以以位为单元中设置为输入或输出。

除了用作 I/O 端口外，这些引脚也用作在控制模式下的外部中断请求信号输入引脚。在端口模式或控制模式下，该端口可以以位为单位进行设置。每个引脚的有效沿由 INTR1 和 INTF1 寄存器指定。

通过使用上拉电阻选项寄存器 1 (PU1)，可以将片上上拉电阻连接到 P10 和 P11。

(a) 端口模式

通过使用端口模式寄存器 1 (PM1)，可以在输入或输出模式下以位为单位设置 P10 和 P11。

(b) 控制模式**(i) INTP9, INTP10 (外部中断输入) ... 输入**

这些引脚输入外部中断请求信号。

(3) P30 ~ P39 (端口 3) ... 3 状态 I/O

P30~P39 用作一个 10 位 I/O 端口，可以以位为单元中设置为输入或输出。

除了作为一个 I/O 端口外，这些引脚也用作外部中断请求信号输入，串行接口 I/O 和定时器/计数器 I/O。在端口模式或控制模式下，该端口可以以位为单位进行设置。每个引脚的有效沿由 INTR3 和 INTF3 寄存器指定。

通过使用上拉电阻选项寄存器 3 (PU3)，可以将片上上拉电阻连接到 P30 ~ P39。

(a) 端口模式

通过使用端口模式寄存器 3 (PM3)，可以在输入或输出模式下以位为单位设置 P30 ~ P39。

(b) 控制模式**(i) RXDA0, RXDA2 (接收数据) ... 输入**

这些引脚输入 UARTA0 和 UARTA2 的串行接收数据。

(ii) TXDA0, TXDA2 (发送数据) ... 输出

这些引脚输出 UARTA0 和 UARTA2 的串行发送数据。

(iii) ASCKA0 (异步串行时钟) ... 输入

该引脚是 UARTA0 的输入引脚。

(iv) INTP7, INTP8 (外部中断输入) ... 输入

这些引脚输入外部中断请求信号。

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

这些引脚是定时器 P0 和 P1 (TMP0 和 TMP1) 的输入引脚。

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

这些引脚是定时器 P0 和 P1 (TMP0 和 TMP1) 的输出引脚。

(4) P40 ~ P42 (端口 4) ... 3 状态 I/O

P40~P42 用作一个 3 位 I/O 端口，可以以位为单元中设置为输入或输出。

除了作为一个 I/O 端口外，这些引脚也用作串行接口 I/O。在端口模式或控制模式下，该端口可以以位为单位进行设置。

通过使用上拉电阻选项寄存器 4 (PU4)，可以将片上上拉电阻连接到 P40 ~ P42。

(a) 端口模式

通过使用端口模式寄存器 4 (PM4)，可以在输入或输出模式下以位为单位设置 P40 ~ P42。

(b) 控制模式**(i) SIB0 (串行输入) ... 输入**

这些引脚输入 CSIB0 的串行接收数据。

(ii) SOB0 (串行输出) ... 输出

这些引脚输出 CSIB0 的串行发送数据。

(iii) $\overline{\text{SCKB0}}$ (串行时钟) ... 3 状态 I/O

这些引脚输入/输出 CSIB0 的串行时钟。

(5) P50 ~ P55 (端口 5) ... 3 状态 I/O

P50~P55 用作一个 6 位 I/O 端口，可以以位为单元中设置为输入或输出。

除了作为一个 I/O 端口外，这些引脚也用作定时器/计数器 I/O，调试功能 I/O 和按键中断输入。在端口模式或控制模式下，该端口可以以位为单位进行设置。

通过使用上拉电阻选项寄存器 5 (PU5)，可以将片上上拉电阻连接到 P50 ~ P55。

(a) 端口模式

通过使用端口模式寄存器 5 (PM5)，可以在输入或输出模式下以位为单位设置 P50~P55。

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

这些引脚输入按键中断。在输入端口模式下，该操作可通过使用按键返回模式寄存器(KRM)指定。

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

这些引脚是定时器 Q0 (TMQ0)的输入引脚。

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

这些引脚是定时器 Q0 (TMQ0)的输出引脚。

(iv) DDI (调试数据输入) ... 输入

这些引脚输入调试数据到调试控制单元 (DCU)。

关于更多细节，敬请参阅第二十五章 片上调试功能。

(v) DDO (调试数据输出) ... 输出

该引脚输出来自 DCU 调试数据。

关于更多细节，敬请参阅第二十五章 片上调试功能。

(vi) DCK (调试时钟输入) ...输入

这些引脚输入调试时钟到 DCU。

关于更多细节，敬请参阅**第二十五章 片上调试功能**。

(vii) DMS (调试模式选择) ...输入

该引脚选择 DCU 的调试模式。

关于更多细节，敬请参阅**第二十五章 片上调试功能**。

(6) P70 ~ P715 (端口 7) ... 3 状态 I/O

P70~715 用作一个 16 位 I/O 端口，可以以位为单元中设置为输入或输出。

除了作为一个 I/O 端口外，这些引脚也用于在控制模式下到 A/D 转换器的模拟输入。当使用模拟输入引脚时，在输入模式下设置该端口。同时，不要读取该端口。

(a) 端口模式

通过使用端口模式寄存器 7L 和 7H (PM7L 和 PM7H)，可以在输入或输出模式下以位为单位设置 P70~P715。

(b) 控制模式

P70 ~ P715 用作 ANI0 ~ ANI15 引脚的复用引脚。

(i) ANI0 ~ ANI15 (模拟输入 0 ~ 15) ...输入

这些引脚输入模拟信号到 A/D 转换器。

(7) P90 ~ P915 (端口 9) ... 3 状态 I/O

P90~P915 用作一个 16 位 I/O 端口，可以以位为单元中设置为输入或输出。

除了作为一个 I/O 端口外，这些引脚也用于串行接口 I/O，定时器/计数器 I/O，时钟输出，外部中断请求信号输入，和 按键中断输入。在端口模式或控制模式下，该端口可以以位为单位进行设置。P913 ~ P915 的有效沿由 INTR9H 和 INTF9H 寄存器指定。

通过使用上拉电阻选项寄存器 9 (PU9)，可以将片上上拉电阻连接到 P90 ~ P915。

(a) 端口模式

通过使用端口模式寄存器 9 (PM9)，可以在输入或输出模式下以位为单位设置 P90~P915。

(b) 控制模式**(i) SIB1 (串行输入) ...输入**

这些引脚输入 CSIB1 的串行接收数据。

(ii) SOB1 (串行输出) ... 输出

这些引脚输出 CSIB1 的串行发送数据。

(iii) SCKB1 (串行时钟) ... 3 状态 I/O

这些引脚输入/输出 CSIB1 的串行时钟。

(iv) RXDA1 (接收数据) ..输入

这些引脚输入 UARTA1 的串行接收数据。

(v) TXDA1 (发送数据) ... 输出

该引脚输出 UARTA1 串行发送数据。

(vi) TIP20, TIP21 (定时器输入) ..输入

这些引脚是定时器 P2 (TMP2)的输入引脚。

(vii) TOP20, TOP21 (定时器输出) ... 输出

这些引脚是定时器 P2 (TMP2)的输出引脚。

(viii) TIQ10, TIQ11, TIQ12, TIQ13 (定时器输入) ...输入

这些引脚是定时器 Q1 (TMQ1)的输入引脚。

(ix) TOQ10, TOQ11, TOQ12, TOQ13 (定时器输出) ... 输出

这些引脚是定时器 Q1 (TMQ1)的输出引脚。

(x) PCL (时钟输出) ... 输出

该引脚输出时钟。

(xi) INTP4 ~ INTP6 (外部中断输入) ...输入

这些引脚输入外部中断请求信号。

(xii) KR6, KR7 (按键返回) ..输入

这些引脚输入按键中断。在输入端口模式下，该操作通过使用按键返回模式寄存器(KRM)指定。

(8) PCM0 ~ PCM3 (端口 CM) ... 3 状态 I/O

PCM0 ~ PCM3 用作一个 4 位 I/O 端口，可以以位为单元中设置为输入或输出。

除了作为一个 I/O 端口外，这些引脚也用于控制模式下的总线时钟输出。

(a) 端口模式

通过使用端口模式寄存器 CM (PMCM)，可以在输入或输出模式下以位为单位设置 PCM0 ~ PCM3。

(b) 控制模式**(i) CLKOUT (时钟输出) ... 输出**

该引脚输出内部生成总线时钟。

(9) PCS0, PCS1 (端口 CS) ... 3 状态 I/O

PCS0 和 PCS1 用作一个 2 位 I/O 端口，可以以位为单元中设置为输入或输出。

(a) 端口模式

通过使用端口模式寄存器 CS (PMCS)，可以在输入或输出模式下以位为单位设置 PCS0 和 PCS1。

(10) PCT0, PCT1, PCT4, PCT6 (端口 CT) ... 3 状态 I/O

PCT0, PCT1, PCT4 和 PCT6 用作一个 4 位 I/O 端口, 可以以位为单元中设置为输入或输出。

(a) 端口模式

通过使用端口模式寄存器 CT (PMCT), 可以在输入或输出模式下以位为单位设置 PCT0, PCT1, PCT4 和 PCT6。

(11) PDL0 ~ PDL13 (端口 DL) ... 3 状态 I/O

PDL0 ~ PDL13 用作一个 14 位 I/O 端口, 可以以位为单元中设置为输入或输出。

当 flash 存储器在编程时 (输入高电平到 FLMD0), PDL5 也用作 FLMD1 引脚。同时, 确保输入低电平到 FLMD1 引脚。

(a) 端口模式

通过使用端口模式寄存器 DL (PMDL), 可以在输入或输出模式下以位为单位设置 PDL0 ~ PDL13。

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

$\overline{\text{RESET}}$ 输入为异步输入。不论运行时钟如何, 当向 $\overline{\text{RESET}}$ 引脚输入固定的低电平宽度的信号时, 系统被复位, 且此操作优先于所有其他的操作。

该引脚用于释放待机模式 (HALT, IDLE, 或 STOP), 也用于正常的初始化和开始。

(13) X1, X2 (主时钟晶振)

该引脚用于连接生成系统时钟的振荡器。

(14) XT1, XT2 (副时钟晶振)

该引脚用于连接生成副时钟的振荡器。

(15) AV_{SS} (模拟地)

该引脚为 A/D 转换器和复用功能端口的地引脚。

(16) AV_{REF0} (模拟参考电压) ... 输入

该引脚提供正模拟电源到 A/D 转换器 和复用功能端口。

它也提供参考电压到 A/D 转换器。

(17) EV_{DD} (端口电源)

该引脚提供正电源到 I/O 端口和复用功能引脚。

(18) EV_{SS} (端口地)

该引脚是 I/O 端口和复用功能引脚的地引脚。

(19) V_{DD} (电源)

该引脚提供正电源, 将所有 V_{DD} 引脚连接到一个正电源。

(20) V_{SS} (地)

该引脚为地引脚。连接 V_{SS} 引脚到地。

(21) FLMD0 (Flash 编程模式) ...输入

该引脚为 flash 存储器编程模式下的信号输入引脚。

在正常操作模式下，连接该引脚到 V_{SS}。

(22) BV_{DD} (端口电源)

该引脚提供正电源到 I/O 端口和复用功能引脚。

(23) BV_{SS} (端口地)

该引脚为 I/O 端口和复用功能引脚的地引脚。

(24) REGC (稳压器控制) ...输入

引脚与稳压器的电容相连接。

2.3 I/O 引脚电路和不使用引脚的推荐连接法

(1/2)

引脚	引脚编号	I/O 电路类型	推荐连接
P00/TIP31/TOP31	6	5-W	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P01/TIP30/TOP30	7		
P02/NMI	17		
P03/INTP0/ADTRG	18		
P04/INTP1	19		
P05/INTP2/DRST	20	5-AF	输入： 单独连接到 EV _{SS} 输出： 保持开路
P06/INTP3	21	5-W	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P10/INTP9	3	5-W	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P11/INTP10	4		
P30/TXDA0	25	5-A	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P31/RXDA0/INTP7	26	5-W	
P32/ASCKA0/TIP00/TOP00/TOP01	27	5-W	
P33/TIP01/TOP01	28		
P34/TIP10/TOP10	29		
P35/TIP11/TOP11	30		
P36	31		
P37	32	5-W	
P38/TXDA2	35		
P39/RXDA2/INTP8	36		
P40/SIB0	22	5-W	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P41/SOB0	23	5-A	
P42/SCKB0	24	5-W	
P50/KR0/TIQ01/TOQ01	37	5-W	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P51/KR1/TIQ02/TOQ02	38		
P52/KR2/TIQ03/TOQ03/DDI	39		
P53/KR3/TIQ00/TOQ00/DDO	40		
P54/KR4/DCK	41		
P55/KR5/DMS	42		
P70/ANI0 ~ P715/ANI15	85 ~ 100	11-G	输入： 通过一个电阻单独连接到 AV _{REF0} 或 AV _{SS} 输出： 保持开路
P90/KR6/TXDA1	43	5-W	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P91/KR7/RXDA1	44		
P92/TIQ11/TOQ11	45		
P93/TIQ12/TOQ12	46		
P94/TIQ13/TOQ13	47		
P95/TIQ10/TOQ10	48		

引脚	引脚编号	I/O 电路类型	推荐连接
P96/TIP21/TOP21	49	5-W	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P97/SIB1/TIP20/TOP20	50		
P98/SOB1	51		
P99/SCKB1	52		
P910 ~ P912	53 ~ 55		
P913/INTP4/PCL	56	5-W	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P914/INTP5	57		
P915/INTP6	58		
PCM0	61	5	输入： 通过一个电阻单独连接到 BV _{DD} 或 BV _{SS} 输出： 保持开路
PCM1/CLKOUT	62		
PCM2	63		
PCM3	64		
PCS0	59	5	输入： 通过一个电阻单独连接到 BV _{DD} 或 BV _{SS} 输出： 保持开路
PCS1	60		
PCT0	65	5	输入： 通过一个电阻单独连接到 BV _{DD} 或 BV _{SS} 输出： 保持开路
PCT1	66		
PCT4	67		
PCT6	68		
PDL0 ~ PDL4	71 ~ 75	5	输入： 通过一个电阻单独连接到 BV _{DD} 或 BV _{SS} 输出： 保持开路
PDL5/FLMD1	76		
PDL6 ~ PDL13	77 ~ 84		
AV _{REF0}	1	-	直接连接到 V _{DD}
AV _{SS}	2	-	-
FLMD0 [*]	8	-	直接连接到 V _{SS}
REGC	10	-	-
RESET	14	2	-
X1	12	-	-
X2	13	-	-
XT1	15	16	通过一个电阻连接到 V _{SS}
XT2	16	16	保持开路
V _{DD}	9	-	-
V _{SS}	11	-	-
BV _{DD}	70	-	-
BV _{SS}	69	-	-
EV _{DD}	5, 34	-	-
EV _{SS}	33	-	-

注 如果在自编程期间，输入到 **RESET** 引脚的噪声超过噪声消除宽度，当一个电容器连接到 **FLMD0** 引脚时，根据电容电荷端定时，可能进入 flash 存储器 on-board 模式。因此，不要将电容器连接到 **FLMD0** 引脚。

2.4 引脚 I/O 电路

图 2-1. 引脚 I/O 电路类型(1/2)

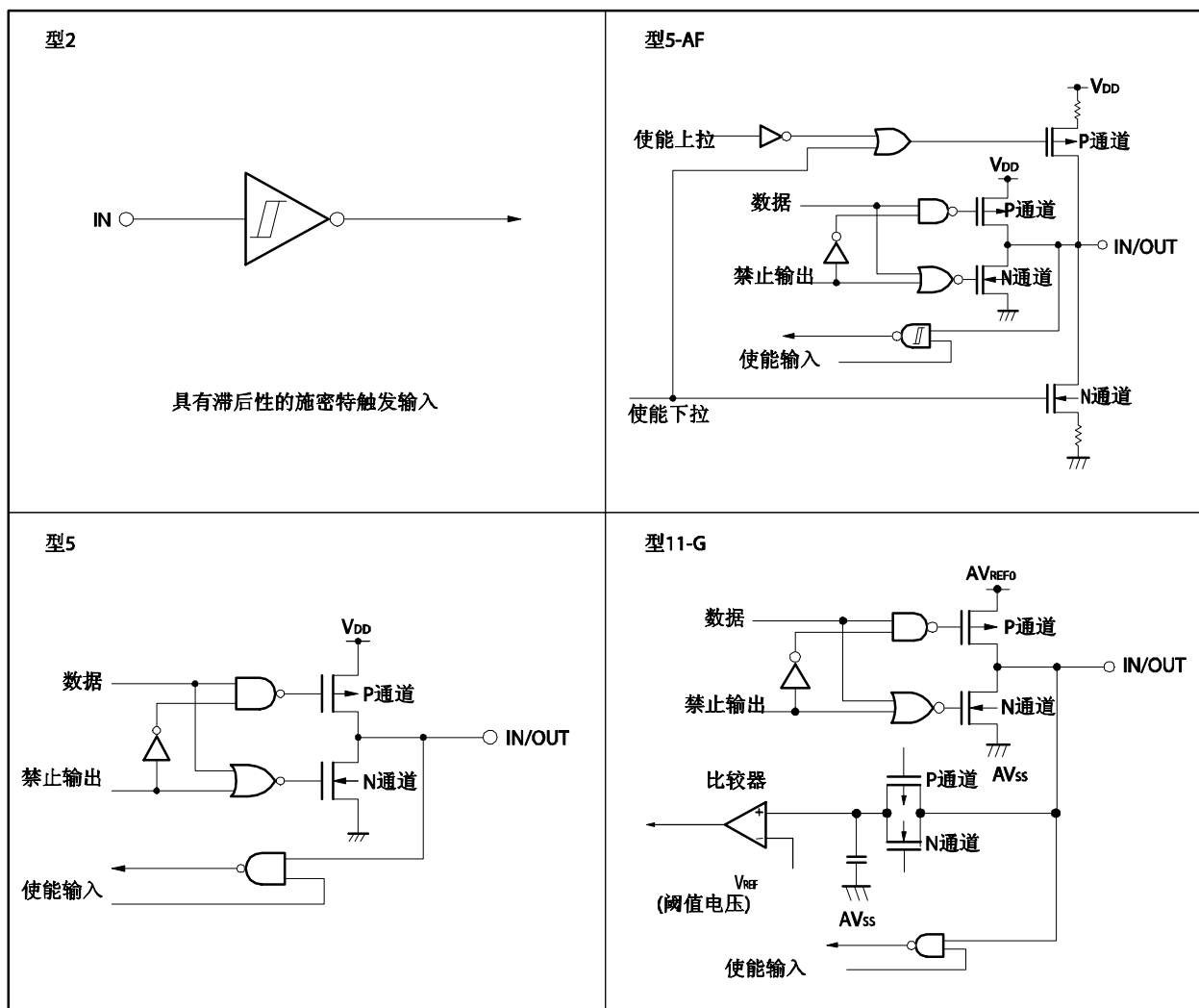
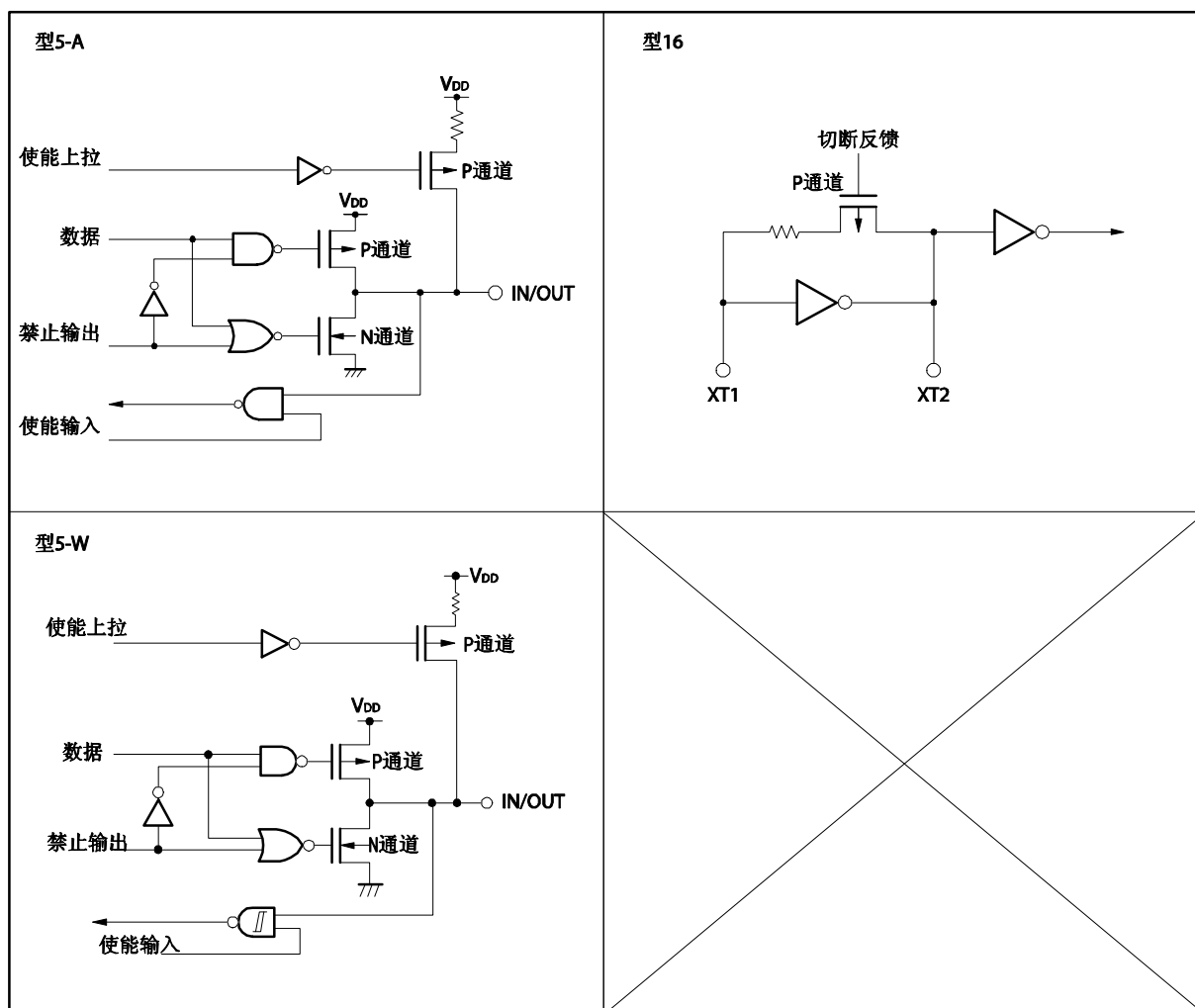


图 2-1. 引脚 I/O 电路类型 (2/2)



备注 视 V_{DD} 为 EV_{DD} 或 BV_{DD}。视 V_{SS} 为 EV_{SS} 或 BV_{SS}。

2.5 注意事项

注意以下引脚可能暂时输出一个未定义电平，即使在使用电源复位期间。
P53/KR3/TIQ00/TOQ00/DDO 引脚

第三章 CPU 功能

V850ES/HG2 系列产品的 CPU 是基于 RISC 结构设计的，具备 5 级流水线控制，几乎所有指令均可在一个时钟周期内完成。

3.1 特点

- 最小指令执行时间：50 ns（20 MHz 操作）
- 存储器空间 程序（物理地址）空间： 64 MB 线性
 数据（逻辑空间）空间： 4 GB 线性
- 通用寄存器：32 位 × 32 寄存器
- 内部 32 位架构
- 5 级流水线控制
- 乘除法指令
- 饱和运算指令
- 32 位移位指令：1 时钟
- 长/短格式的载入/存储指令
- 4 种位操作指令
 - SET1
 - CLR1
 - NOT1
 - TST1

3.2 CPU 寄存器组

V850ES/HG2 产品的寄存器可分为两组：通用程序寄存器和专用系统寄存器。所有寄存器的宽度均为 32 位。关于更多细节，敬请参阅 **V850ES 架构用户手册**。

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

3.2.1 程序寄存器组

程序寄存器组包含通用寄存器和程序计数器。

(1) 通用寄存器(r0 ~ r31)

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

但是, r0 和 r30 是被指令隐含使用的, 因此要慎用这两个寄存器。寄存器 r0 始终保持 0 值, 用于使用了 0 值的操作和偏移量为 0 的寻址操作。当指令 SLD 和 SST 访问存储器时, 寄存器 r30 被作为基指针使用。r1, r3~r5 以及 r31 被汇编器和 C 编译器隐含使用。当使用这些寄存器时, 要对它们的内容进行保存, 使用之后, 要对寄存器的值进行恢复。寄存器 r2 有时会被实时操作系统所使用。如果实时系统不使用 r2, 它可作为变量寄存器使用。

表 3-1. 程序寄存器

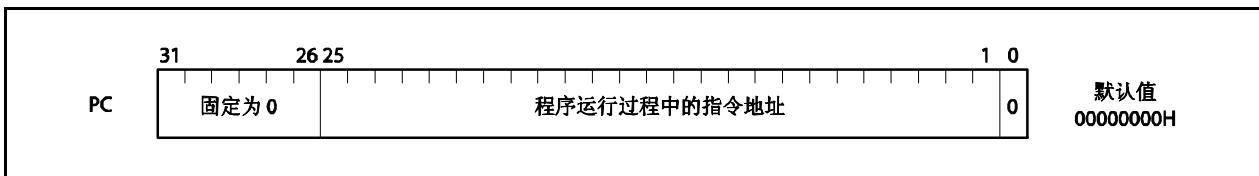
名称	用途	操作
r0	零寄存器	始终保持为零
r1	汇编器保留寄存器	用于生成 32 位立即数据的工作寄存器
r2	地址/数据变量寄存器 (如果实时操作系统不使用 r2)	
r3	堆栈指针	当函数被调用时用于生成堆栈架构
r4	全局指针	用于访问数据区域内的全局变量
r5	文本指针	用于表示文本区域 (放置程序程序代码的区域) 的开始部分的寄存器
r6 ~ r29	地址/数据变量寄存器	
r30	元素指针	用于访问存储器时基指针
r31	连接指针	用于编译器调用函数时
PC	程序计数器	程序执行时保持指令地址

备注 更多关于 r1, r3 ~ r5 和 r31 用于汇编程序和 C 编译器的细节, 敬请参阅 CA850 (C 编译器安装包) 汇编语言用户手册。

(2) 程序计数器(PC)

在程序执行时, 该程序计数器保持指令地址。该寄存器的低 26 位有效。第 31 ~ 26 位恒为 0。若第 25 位到第 26 位产生进位, 那么这个进位将被忽略。

第 0 位固定为 0。因此, 向奇地址的跳转将不会被执行。



3.2.2 系统寄存器组

系统寄存器控制 CPU 的状态并且保持中断信息。

通过下表所列的寄存器编号，可以使用系统寄存器载入/存储指令（LDSR 和 STSR）读写这些寄存器。

表 3-2. 系统寄存器编号

系统寄存器编号	系统寄存器名称	操作规范	
		LDSR 指令	STSR 指令
0	中断状态保存寄存器(EIPC) ^{#1}	√	√
1	中断状态保存寄存器(EIPSW) ^{#1}	√	√
2	NMI 状态保存寄存器 (FEPC) ^{#1}	√	√
3	NMI 状态保存寄存器 (FEPSW) ^{#1}	√	√
4	中断源寄存器(ECR)	×	√
5	程序状态字(PSW)	√	√
6 ~ 15	为将来功能扩展预留（如果访问这些寄存器，则操作不能保证。）	×	×
16	CALLT 执行状态保存寄存器 (CTPC)	√	√
17	CALLT 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)	√ ^{注2}	√ ^{注2}
19	异常/调试陷阱状态保存寄存器(DBPSW)	√ ^{注2}	√ ^{注2}
20	CALLT 基址指针 (CTBP)	√	√
21 ~ 31	为将来功能扩展预留（如果访问这些寄存器，则操作不能保证。）	×	×

- 注**
1. 由于系统寄存器只有一组，如果要实现多重中断服务，就必须用程序对这组寄存器的内容进行保存。
 2. 这些寄存器只能在 DBTRAP 指令或非法操作代码和 DBRET 指令之间被访问。

注意事项 即使 EIPC 或 FEPC，或 CTPC 的第 0 位被 LDSR 指令置为 1，在中断服务程序执行后由 RETI 指令返回到主程序时，第 0 位仍然被忽略（这是因为 PC 的第 0 位固定为 0）。将 EIPC，FEPC 和 CTPC 设置为偶数值（第 0 位=0）。

备注

√: 允许访问
 ×: 禁止访问

(1) 中断状态保存寄存器(EIPC 和 EIPSW)

当中断发生时，EIPC 和 EIPSW 用于保存状态。

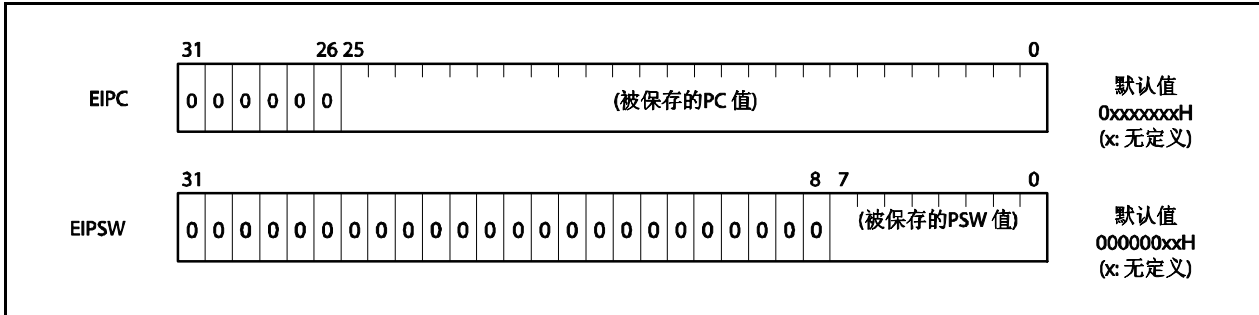
如果发生软件异常或可屏蔽中断，程序计数器 PC 的值被保存在 EIPC 中，程序状态字 PSW 的值被保存在 EIPSW 中（如果发生不可屏蔽中断，这些值会被保存在 NMI 状态保存寄存器 FEPC 和 FEPSW 中）。软件异常或可屏蔽中断发生时，当前执行指令的下一个指令（除了某些指令之外）的地址被保存到 EIPC 中，（参见 15.8 CPU 不对中断进行响应的时期）。

当前的 PSW 值被保存在 EIPSW 中。

由于只有一组中断状态保存寄存器可用，因此当允许多重中断时，这些寄存器的值需要由程序自行保存。

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

通过执行 RETI 指令，将 EIPC 的值恢复到 PC 且将 EIPSW 的值恢复到 PSW。



(2) 不可屏蔽中断状态保存寄存器(FEPC 和 FEPSW)

当不可屏蔽中断（NMI）发生时，FEPC 和 FEPSW 用于保存状态。

如果发生不可屏蔽中断（NMI），程序计数器 PC 的值被保存在 FEPC 中，程序状态字 PSW 的值被保存在 FEPSW 中。

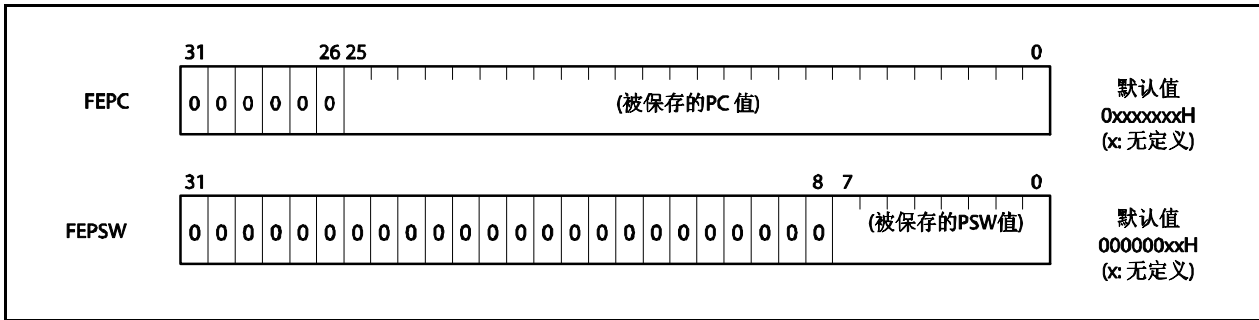
当不可屏蔽中断发生时，当前执行指令的下一条指令（除了某些指令之外）的地址会被保存到 FEPC 中。

当前的 PSW 值被保存在 FEPSW 中。

由于只有一组 NMI 状态保存寄存器可用，因此当多重中断发生时，这些寄存器的值需要由程序自行保存。

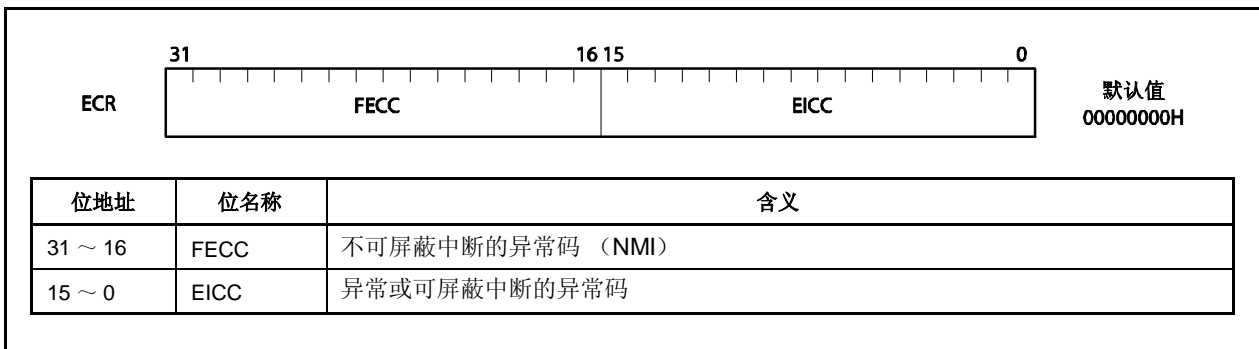
FEPC 的第 26 到 31 位和 FEPSW 的第 8 到 31 位为扩展功能保留（这些位始终为 0）。

通过执行 RETI 指令，将 FEPC 的值恢复到 PC 且将 FEPSW 的值恢复到 PSW。



(3) 中断源寄存器(ECR)

如果异常或中断发生时，中断源寄存器（ECR）用于保持异常或中断源。该寄存器保持每一个中断源的异常代码。因为该寄存器是只读寄存器，因此当使用 LDSR 指令时不能将数据写入该寄存器。



(4) 程序状态字(PSW)

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

如果使用 LDSR 指令更改其中一位标志, 那么新的状态将在 LDSR 指令执行后立即生效。但如果 ID 标志设为 1, 当 LDSR 指令执行时, 中断请求将不会被响应。

该寄存器的第 8 到 31 位为扩展功能保留 (这些位始终为 0)。

(1/2)

PSW	<div style="display: flex; justify-content: space-between; align-items: center;"> 31 8 7 6 5 4 3 2 1 0 </div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> RFU <table border="1" style="float: right; border-collapse: collapse;"> <tr> <td style="text-align: center;">NP</td> <td style="text-align: center;">EP</td> <td style="text-align: center;">ID</td> <td style="text-align: center;">SAT</td> <td style="text-align: center;">CY</td> <td style="text-align: center;">OV</td> <td style="text-align: center;">S</td> <td style="text-align: center;">Z</td> </tr> </table> </div> <div style="text-align: right; margin-top: 5px;"> 默认值 00000020H </div>	NP	EP	ID	SAT	CY	OV	S	Z
NP	EP	ID	SAT	CY	OV	S	Z		
位地址	标志名称	含义							
31 ~ 8	RFU	保留区域。固定为零。							
7	NP	表示正在进行非可屏蔽中断 (NMI) 服务。响应 NMI 后该位被设置为 1, 同时禁止多重中断的响应。 0: 未执行 NMI 服务 1: 正在执行 NMI 服务							
6	EP	表示正在进行异常处理。异常产生后该位被设置为 1。即使标志被设置, 仍可以对中断请求进行响应。 0: 未执行异常处理。 1: 正在执行异常处理。							
5	ID	显示可屏蔽中断能否被响应。 0: 允许中断 1: 禁止中断							
4	SAT ^注	显示由于产生了溢出, 饱和运算处理指令的结果是饱和的。因为该标志是累加性标志, 所以运算结果为饱和的饱和运算指令将把该标志置 1, 而即使下一条指令的运算结果不为饱和, 该标志也不会被清零。使用 LDSR 指令将该标志清零。通过执行算术运算指令, 该标志既不会被置 1, 也不会被清零。 0: 不饱和 1: 饱和							
3	CY	运算产生进位或借位时, 该标志被置 1。 0: 未产生进位或借位。 1: 产生进位或借位。							
2	OV ^注	显示运算过程中产生溢出时, 该标志被置 1。 0: 未产生溢出。 1: 产生溢出。							
1	S ^注	显示运算结果为负值时, 该标志被置 1。 0: 运算结果为正值或零。 1: 运算结果为负值。							
0	Z	显示运算结果为零时, 该标志被设置 0。 0: 运算结果不为 0。 1: 运算结果为 0。							
备注 请阅读下页中的注。									

注 执行饱和和处理运算的操作结果是由 OV 和 S 标志的内容决定的。在饱和运算过程中只有当 OV 标志被置为 1 时，SAT 标志才会被置为 1。

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

(5) CALLT 执行状态保存寄存器 (CTPC 和 CTPSW)

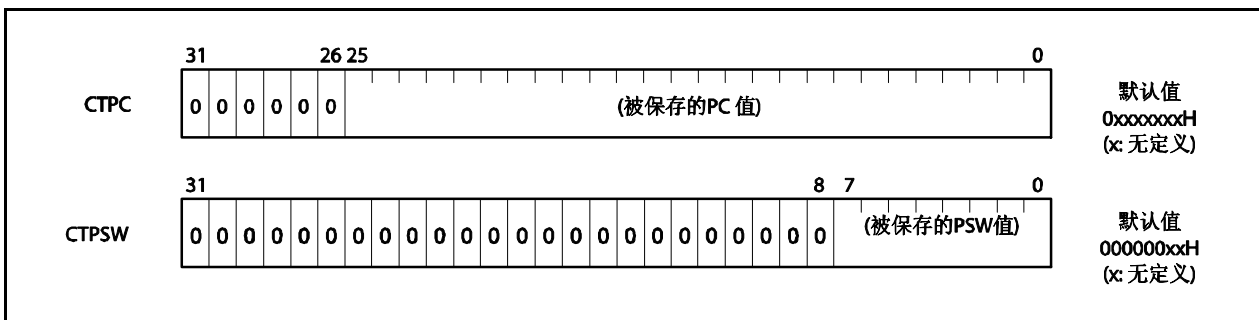
CTPC 和 CTPSW 是 CALLT 执行状态保存寄存器。

当 CALLT 指令执行时，程序计数器 PC 值被保存在 CTPC 中，程序状态字 PSW 值被保存在 CTPSW 中。

保存在 CTPC 中的值是 CALLT 下一条指令的地址。

保存在 CTPSW 中的值是当前的 PSW 值。

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



3.3 操作模式

V850ES/HG2 具有以下几种操作模式。

(1) 普通操作模式

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

(2) Flash 存储器编程模式

在这个模式下，内部 flash 存储器可以用 flash 编程器进行编程。

(3) 片上调试模式

The V850ES/HG2 提供了一个使用 JTAG（联合测试行动小组）通讯规范的片上调试功能。更多细节，敬请参阅**第二十五章 片上调试功能**。

3.3.1 指定操作模式

通过使用 FLMD0 和 FLMD1 引脚指定操作模式。

普通模式下，要确保 FLMD0 引脚在复位释放期间输入低电平。

Flash 存储器编程模式下，flash 编程器应该输入 FLMD0 引脚高电平信号，但是在自编程模式下这个信号必须通过外部电路输入。

复位被释放时的操作		复位后的操作模式
FLMD0	FLMD1	
L	×	普通操作模式
H	L	Flash 存储器编程模式
H	H	禁止设置

备注

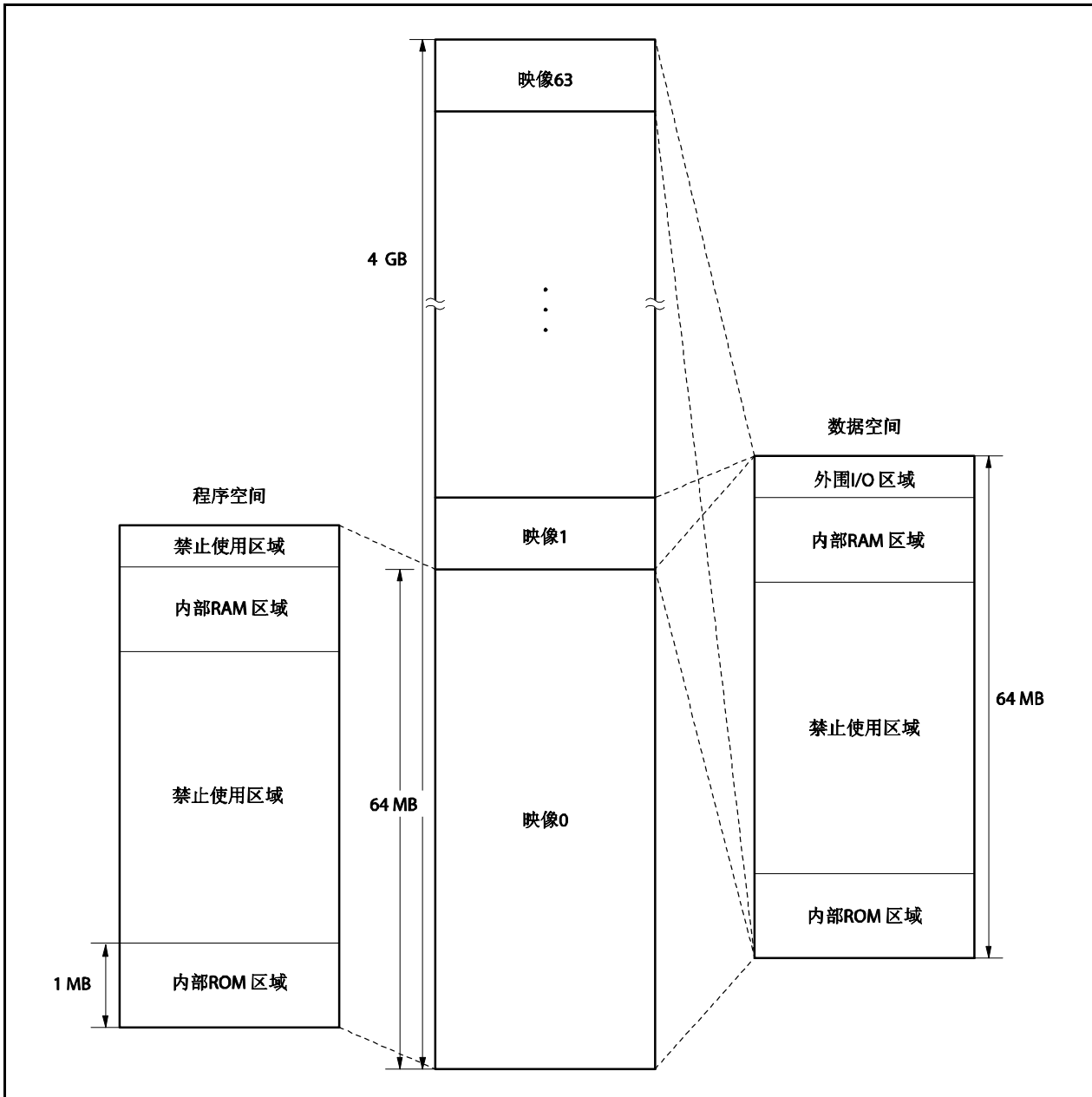
L: 低电平输入
H: 高电平输入
×: 忽略

3.4 地址空间

3.4.1 CPU 地址空间

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

图 3-1. 地址空间图



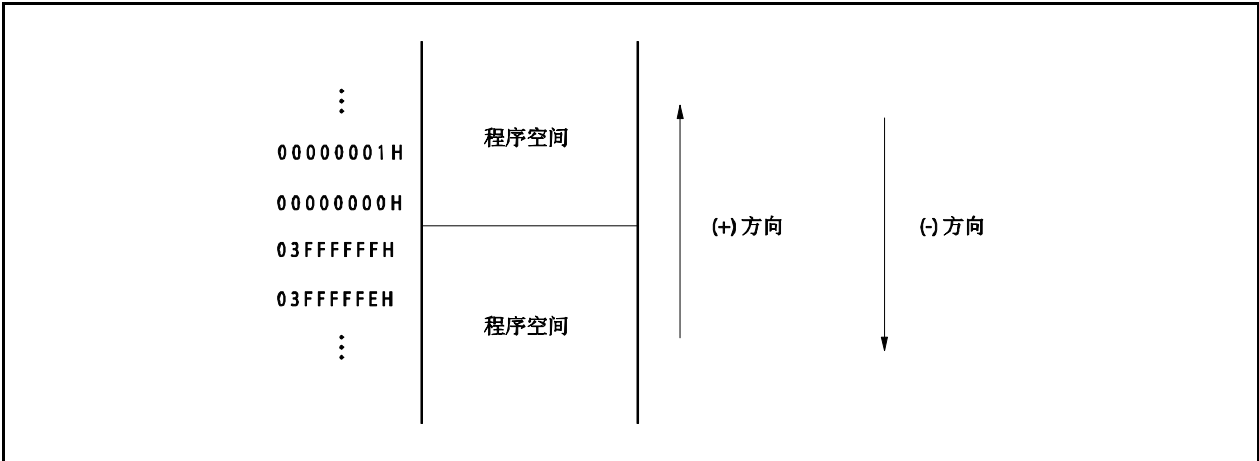
3.4.2 CPU 地址空间的回绕

(1) 程序空间

PC（程序计数器）的 32 位中，高 6 位固定为 0，只有低 26 位有效。在分支地址计算过程中，高 6 位忽略第 25 到 26 位的进位或借位。

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

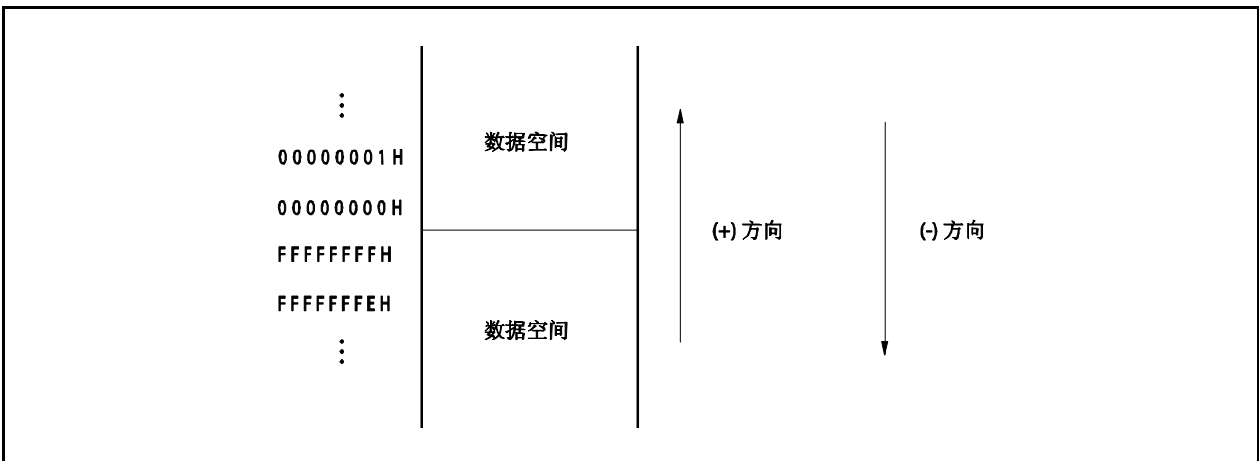
注意事项 由于地址 03FFF000H ~ 03FFFFFFH 的 4 KB 范围是片上外围 I/O 区域，因此指令不能从这个区域取址。因此，要避免执行可能使分支地址计算结果为该范围中值的操作。



(2) 数据空间

结果超过 32 位的操作数地址计算操作将被忽略。

因此，数据空间的最高地址 FFFFFFFFH 和最低地址 00000000H 之间是相连的，回绕现象会发生在这些地址附近。



3.4.3 存储器映射

V850ES/HG2 存储器的保留区域如下所示。

图 3-2. 数据存储映射（物理地址）

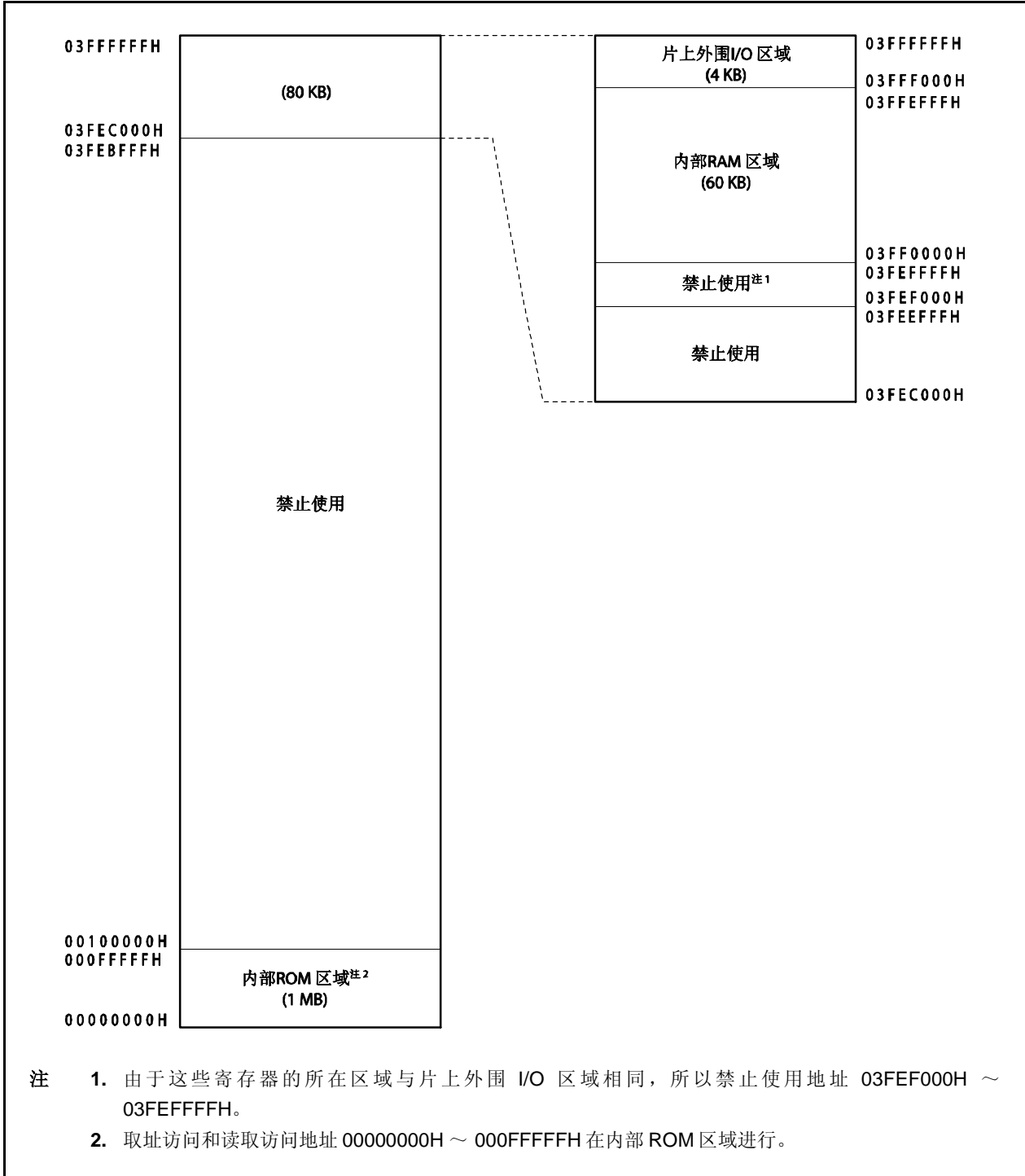
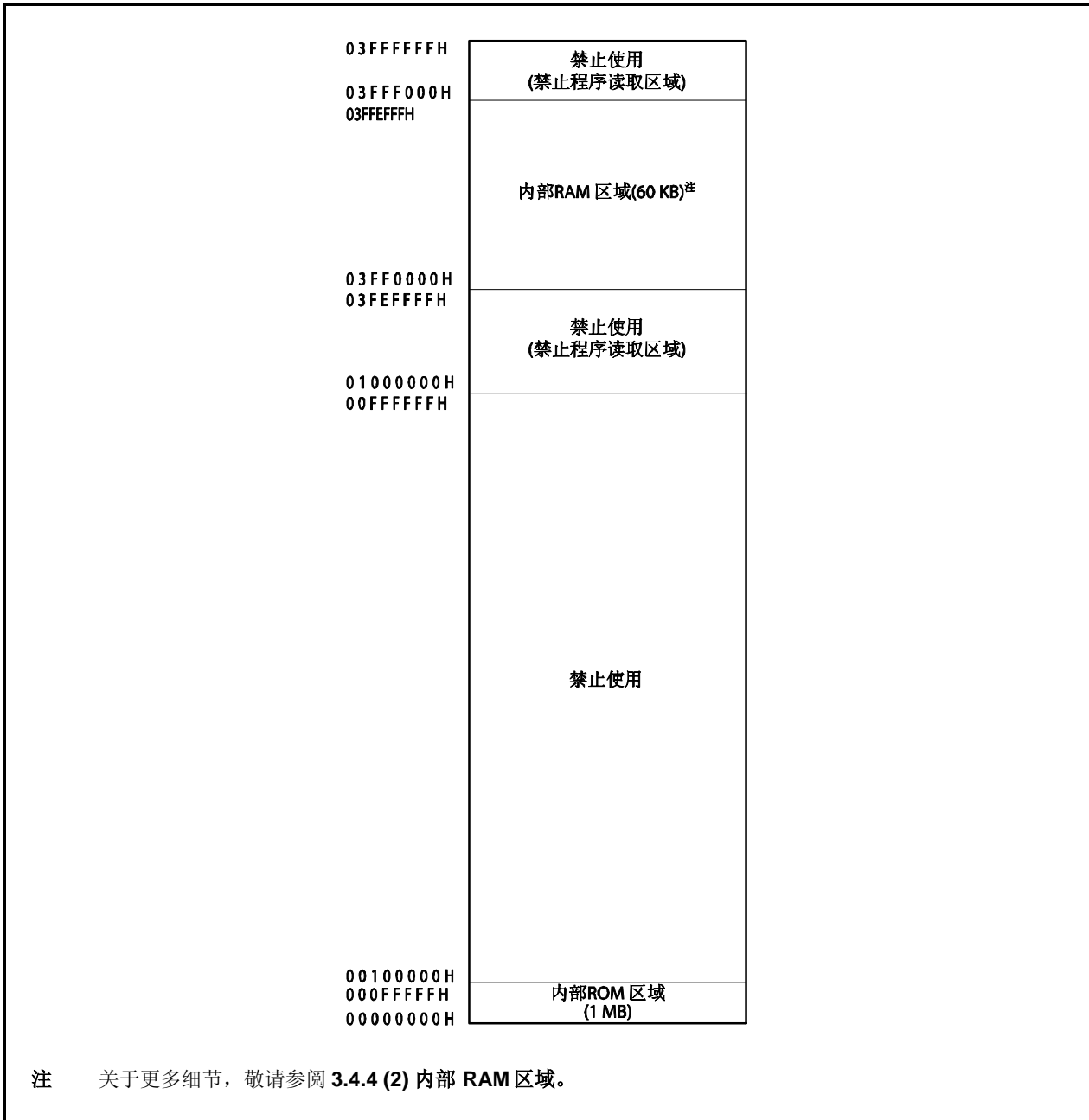


图 3-3. 程序存储器映射



3.4.4 区域

(1) 内部 ROM 区域

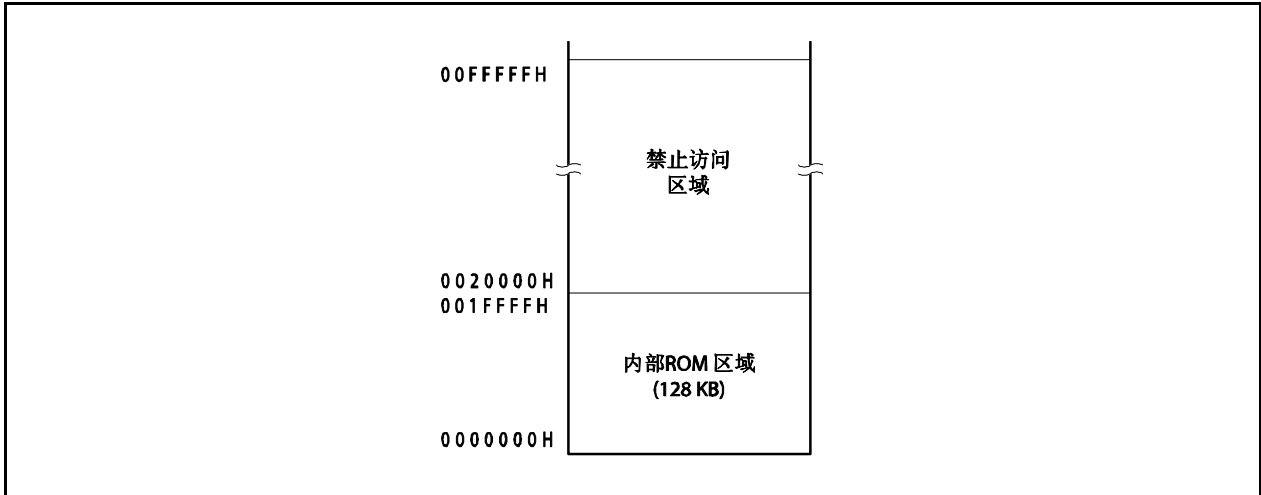
最大 1 MB 的空间被保留为内部 ROM 区域。

(a) 内部 ROM (128 KB)

μ PD70F3706 芯片的内部 ROM 区域为地址 0000000H ~ 001FFFFH 的 128 KB 区域。

对地址 0020000H ~ 00FFFFFFH 的访问是被禁止的。

图 3-4. 内部 ROM 区域 (128 KB)

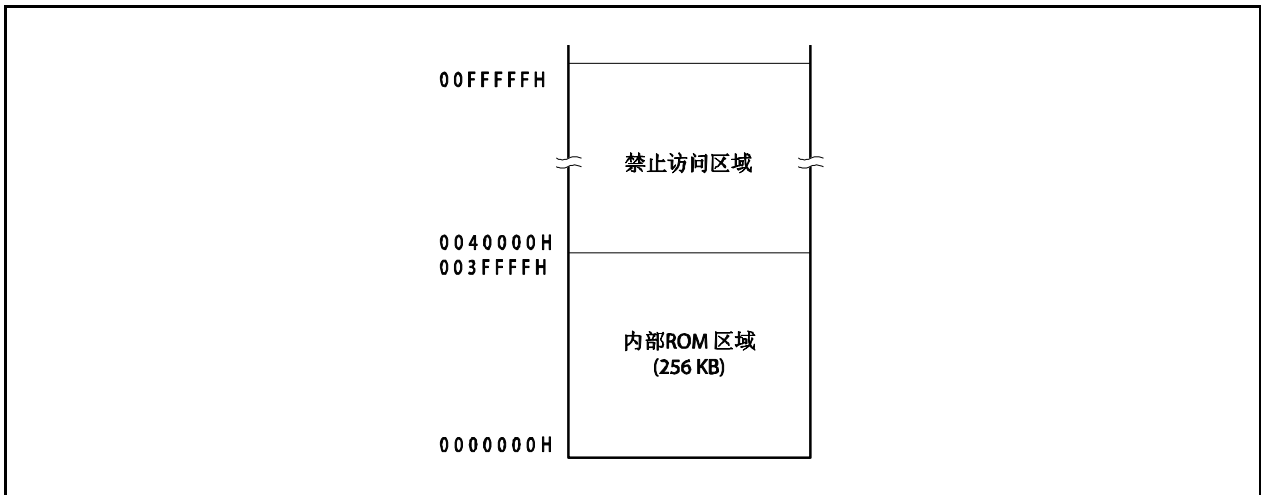


(b) 内部 ROM (256 KB)

μ PD70F3707 芯片的内部 ROM 区域为地址 0000000H ~ 003FFFFH 的 256 KB 区域。

对地址 0040000H ~ 00FFFFFFH 的访问是被禁止的。

图 3-5. 内部 ROM 区域 (256 KB)



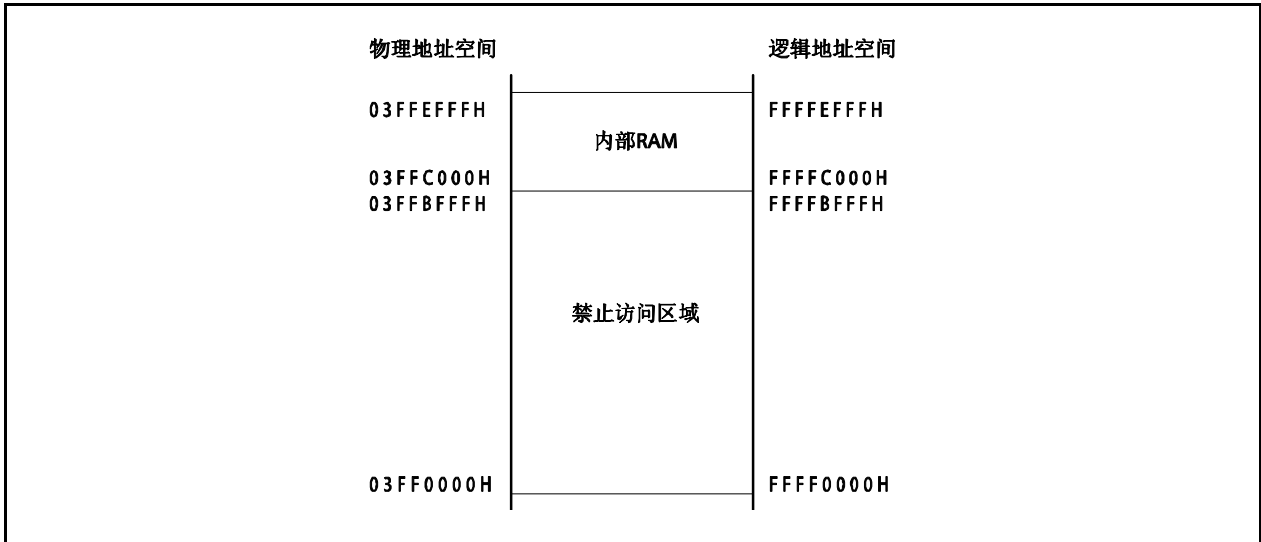
(2) 内部 RAM 区域

最大 60 KB 的空间被保留为内部 RAM 区域。

(a) 内部 RAM (12 KB)

V850ES/HG2 芯片的内部 RAM 区域为地址 03FFC000H ~ 03FFFEFFFH 的 12 KB 区域。
对地址 03FF0000H ~ 03FFBFFFH 的访问是被禁止的。

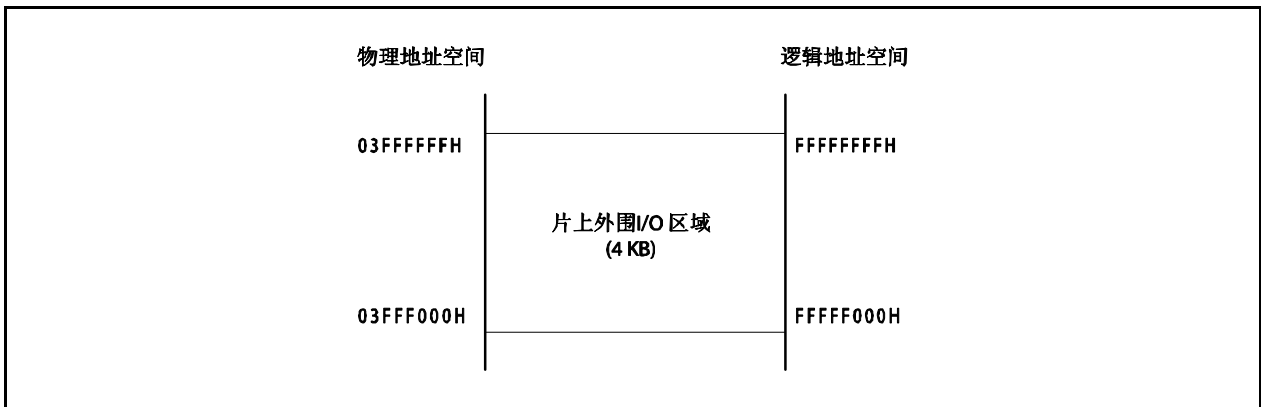
图 3-6. 内部 RAM 区域 (12 KB)



(3) 片上外围 I/O 区域

地址 03FFF000H ~ 03FFFFFFFH 的 4 KB 区域被保留为片上外围 I/O 区域。

图 3-7. 片上外围 I/O 区域



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

- 注意事项**
1. 当寄存器被一个字操作指令读写时，一个字的空间被分为低 16 位和高 16 位，分两次被字节操作先后读写，其中最低 2 位地址被忽略。
 2. 若对能进行字节操作的寄存器进行半字长操作，那么读取该寄存器时，高 8 位数据未被指定，将数据写入低 8 位数据。
 3. 未被指定为寄存器的地址为将来扩展保留。当这些地址被访问时，该操作未被指定且操作结果的有效性无法保证。

3.4.5 地址空间的推荐使用方法

V850ES/HG2 的架构要求在对数据区域中的操作数进行读写时，要确保有一个寄存器可以作为地址指针来使用。这个指针中的地址值的 $\pm 32\text{KB}$ 的地址范围内的操作数可以被指令直接访问。由于能够用来充当指针的通用寄存器数量有限，当指针值改变引起地址计算时，为了不影响系统的性能，要指定尽可能多的通用寄存器来保存变量值，这样也会减少程序所占的空间。

(1) 程序空间

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

若使用内部 RAM 区域作为程序空间，可访问地址 03FFC000H ~ 03FFEFFFH (12 KB)。

注意事项 如果分支指令在内部 RAM 区域的上限处发生，那么将不会产生跨越片上外围 I/O 区域的预取址（无效取址）。

(2) 数据空间

在 V850ES/HG2 系列器件中，4GB 的 CPU 地址空间可被看作 64 个 64MB 的地址空间，因此 26 位地址的最低有效位（位 25）是作为符号扩展位被扩展至 32 位而被用作地址值的。

(a) 回绕式内存空间的应用举例

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

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

图 3-8. 回绕 (μPD70F3707)

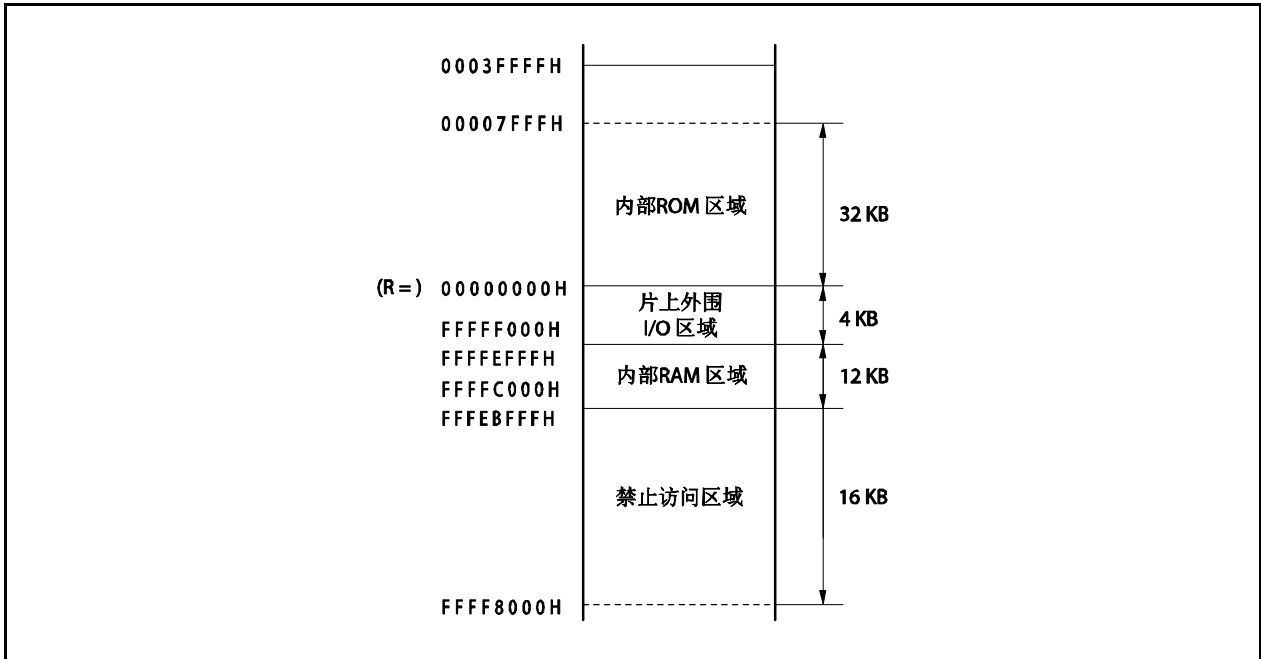
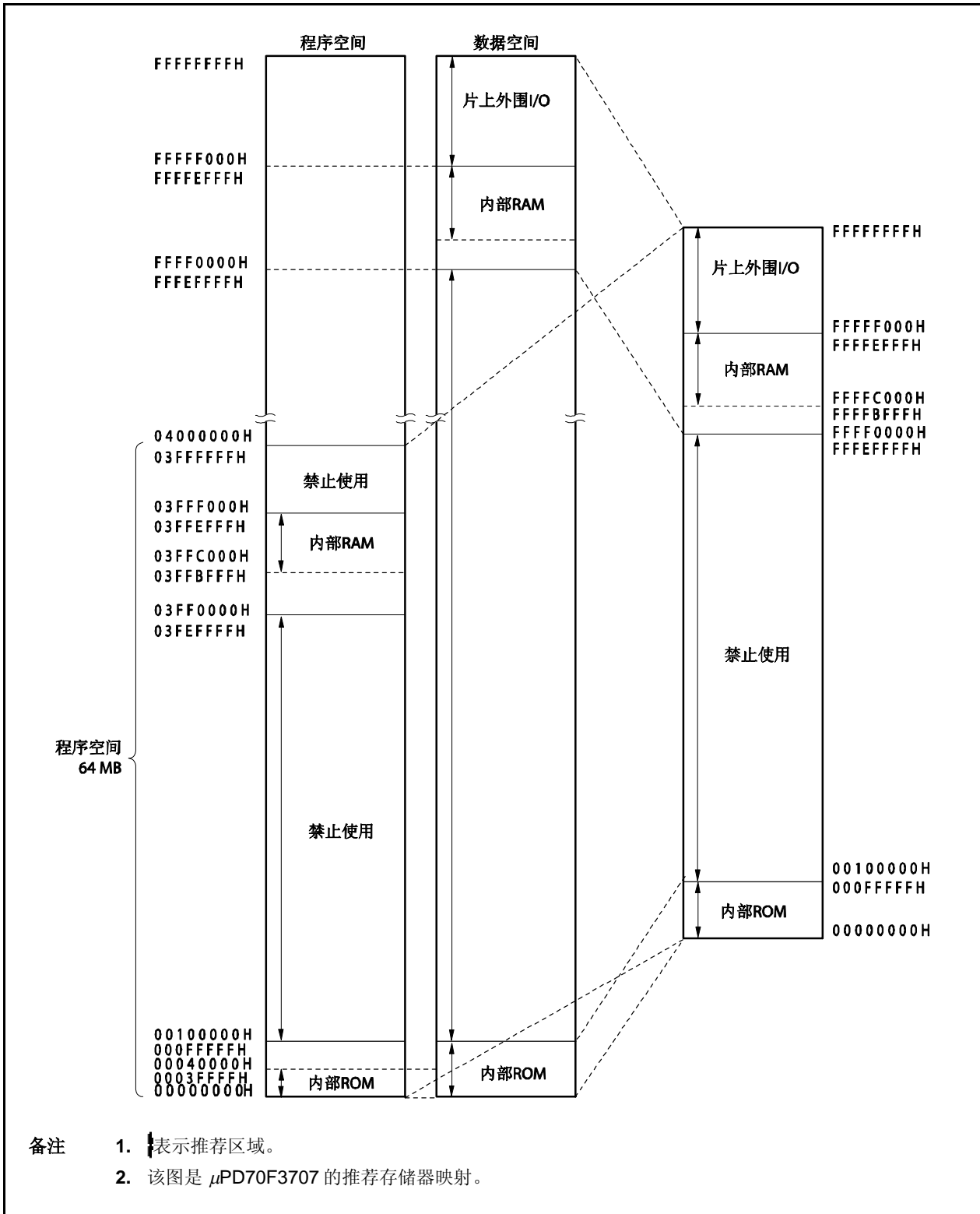


图 3-9. 推荐的存储器映射



3.4.6 外围 I/O 寄存器

(1/9)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
FFFFF004H	端口 DL	PDL	R/W			√	无定义
FFFFF004H	端口 DLL	PDLL		√	√		无定义
FFFFF005H	端口 DLH	PDLH		√	√		无定义
FFFFF008H	端口 CS	PCS		√	√		无定义
FFFFF00AH	端口 CT	PCT		√	√		无定义
FFFFF00CH	端口 CM	PCM		√	√		无定义
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
FFFFF080H	DMA 源地址寄存器 0L	DSA0L				√	无定义
FFFFF082H	DMA 源地址寄存器 0H	DSA0H				√	无定义
FFFFF084H	DMA 目的地址寄存器 0L	DDA0L				√	无定义
FFFFF086H	DMA 目的地址寄存器 0H	DDA0H				√	无定义
FFFFF088H	DMA 源地址寄存器 1L	DSA1L				√	无定义
FFFFF08AH	DMA 源地址寄存器 1H	DSA1H				√	无定义
FFFFF08CH	DMA 目的地址寄存器 1L	DDA1L				√	无定义
FFFFF08EH	DMA 目的地址寄存器 1H	DDA1H				√	无定义
FFFFF090H	DMA 源地址寄存器 2L	DSA2L				√	无定义
FFFFF092H	DMA 源地址寄存器 2H	DSA2H				√	无定义
FFFFF094H	DMA 目的地址寄存器 2L	DDA2L				√	无定义
FFFFF096H	DMA 目的地址寄存器 2H	DDA2H				√	无定义
FFFFF098H	DMA 源地址寄存器 3L	DSA3L				√	无定义
FFFFF09AH	DMA 源地址寄存器 3H	DSA3H				√	无定义
FFFFF09CH	DMA 目的地址寄存器 3L	DDA3L				√	无定义
FFFFF09EH	DMA 目的地址寄存器 3H	DDA3H				√	无定义
FFFFF0C0H	DMA 传输计数寄存器 0	DBC0				√	无定义
FFFFF0C2H	DMA 传输计数寄存器 1	DBC1				√	无定义
FFFFF0C4H	DMA 传输计数寄存器 2	DBC2				√	无定义
FFFFF0C6H	DMA 传输计数寄存器 3	DBC3			√	无定义	
FFFFF0D0H	DMA 寻址控制寄存器 0	DADC0			√	0000H	
FFFFF0D2H	DMA 寻址控制寄存器 1	DADC1			√	0000H	
FFFFF0D4H	DMA 寻址控制寄存器 2	DADC2			√	0000H	
FFFFF0D6H	DMA 寻址控制寄存器 3	DADC3			√	0000H	
FFFFF0E0H	DMA 通道控制寄存器 0	DCHC0	√	√		00H	
FFFFF0E2H	DMA 通道控制寄存器 1	DCHC1	√	√		00H	
FFFFF0E4H	DMA 通道控制寄存器 2	DCHC2	√	√		00H	
FFFFF0E6H	DMA 通道控制寄存器 3	DCHC3	√	√		00H	

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
FFFFFF100H	中断屏蔽寄存器 0	IMR0	R/W			√	FFFFH
FFFFFF100H	中断屏蔽寄存器 0L	IMR0L		√	√		FFH
FFFFFF101H	中断屏蔽寄存器 0H	IMR0H		√	√		FFH
FFFFFF102H	中断屏蔽寄存器 1	IMR1				√	FFFFH
FFFFFF102H	中断屏蔽寄存器 1L	IMR1L		√	√		FFH
FFFFFF103H	中断屏蔽寄存器 1H	IMR1H		√	√		FFH
FFFFFF104H	中断屏蔽寄存器 2	IMR2				√	FFFFH
FFFFFF104H	中断屏蔽寄存器 2L	IMR2L		√	√		FFH
FFFFFF105H	中断屏蔽寄存器 2H	IMR2H		√	√		FFH
FFFFFF106H	中断屏蔽寄存器 3	IMR3				√	FFFFH
FFFFFF106H	中断屏蔽寄存器 3L	IMR3L		√	√		FFH
FFFFFF107H	中断屏蔽寄存器 3H	IMR3H		√	√		FFH
FFFFFF110H	中断控制寄存器	LVVIC		√	√		47H
FFFFFF112H	中断控制寄存器	PIC0		√	√		47H
FFFFFF114H	中断控制寄存器	PIC1		√	√		47H
FFFFFF116H	中断控制寄存器	PIC2		√	√		47H
FFFFFF118H	中断控制寄存器	PIC3		√	√		47H
FFFFFF11AH	中断控制寄存器	PIC4		√	√		47H
FFFFFF11CH	中断控制寄存器	PIC5		√	√		47H
FFFFFF11EH	中断控制寄存器	PIC6		√	√		47H
FFFFFF120H	中断控制寄存器	PIC7		√	√		47H
FFFFFF122H	中断控制寄存器	TQ0OVIC		√	√		47H
FFFFFF124H	中断控制寄存器	TQ0CCIC0		√	√		47H
FFFFFF126H	中断控制寄存器	TQ0CCIC1		√	√		47H
FFFFFF128H	中断控制寄存器	TQ0CCIC2		√	√		47H
FFFFFF12AH	中断控制寄存器	TQ0CCIC3		√	√		47H
FFFFFF12CH	中断控制寄存器	TP0OVIC		√	√		47H
FFFFFF12EH	中断控制寄存器	TP0CCIC0		√	√		47H
FFFFFF130H	中断控制寄存器	TP0CCIC1		√	√		47H
FFFFFF132H	中断控制寄存器	TP1OVIC		√	√		47H
FFFFFF134H	中断控制寄存器	TP1CCIC0		√	√		47H
FFFFFF136H	中断控制寄存器	TP1CCIC1		√	√		47H
FFFFFF138H	中断控制寄存器	TP2OVIC		√	√		47H
FFFFFF13AH	中断控制寄存器	TP2CCIC0		√	√		47H
FFFFFF13CH	中断控制寄存器	TP2CCIC1		√	√		47H
FFFFFF13EH	中断控制寄存器	TP3OVIC		√	√		47H
FFFFFF140H	中断控制寄存器	TP3CCIC0		√	√		47H
FFFFFF142H	中断控制寄存器	TP3CCIC1		√	√		47H
FFFFFF144H	中断控制寄存器	TM0EQIC0		√	√		47H
FFFFFF146H	中断控制寄存器	CB0RIC		√	√		47H
FFFFFF148H	中断控制寄存器	CB0TIC		√	√		47H
FFFFFF14AH	中断控制寄存器	CB1RIC		√	√		47H

地址	功能寄存器名称	符号	R/W	可操作位数			默认值	
				1	8	16		
FFFF14CH	中断控制寄存器	CB1TIC	R/W	√	√		47H	
FFFF14EH	中断控制寄存器	UA0RIC		√	√		47H	
FFFF150H	中断控制寄存器	UA0TIC		√	√		47H	
FFFF152H	中断控制寄存器	UA1RIC		√	√		47H	
FFFF154H	中断控制寄存器	UA1TIC		√	√		47H	
FFFF156H	中断控制寄存器	ADIC		√	√		47H	
FFFF160H	中断控制寄存器	KRIC		√	√		47H	
FFFF162H	中断控制寄存器	WTIC		√	√		47H	
FFFF164H	中断控制寄存器	WTIC		√	√		47H	
FFFF166H	中断控制寄存器	PIC8		√	√		47H	
FFFF168H	中断控制寄存器	PIC9		√	√		47H	
FFFF16AH	中断控制寄存器	PIC10		√	√		47H	
FFFF16CH	中断控制寄存器	TQ1OVIC		√	√		47H	
FFFF16EH	中断控制寄存器	TQ1CCIC0		√	√		47H	
FFFF170H	中断控制寄存器	TQ1CCIC1		√	√		47H	
FFFF172H	中断控制寄存器	TQ1CCIC2		√	√		47H	
FFFF174H	中断控制寄存器	TQ1CCIC3		√	√		47H	
FFFF176H	中断控制寄存器	UA2RIC		√	√		47H	
FFFF178H	中断控制寄存器	UA2TIC		√	√		47H	
FFFF182H	中断控制寄存器	DMAIC0		√	√		47H	
FFFF184H	中断控制寄存器	DMAIC1		√	√		47H	
FFFF186H	中断控制寄存器	DMAIC2		√	√		47H	
FFFF188H	中断控制寄存器	DMAIC3		√	√		47H	
FFFF1FAH	当前服务优先级寄存器	ISPR		R	√	√		00H
FFFF1FCH	命令寄存器	PRCMD	W		√		无定义	
FFFF1FEH	节电控制寄存器	PSC	R/W	√	√		00H	
FFFF200H	A/D 转换器模式寄存器 0	ADA0M0		√	√		00H	
FFFF201H	A/D 转换器模式寄存器 1	ADA0M1		√	√		00H	
FFFF202H	A/D 转换器通道指定寄存器	ADA0S		√	√		00H	
FFFF203H	A/D 转换器模式寄存器 2	ADA0M2		√	√		00H	
FFFF204H	掉电比较模式寄存器	ADA0PFM		√	√		00H	
FFFF205H	掉电比较临界值寄存器	ADA0PFT		√	√		00H	
FFFF210H	A/D 转换器结果寄存器 0	ADA0CR0		R			√	无定义
FFFF211H	A/D 转换器结果寄存器 0H	ADA0CR0H				√		无定义
FFFF212H	A/D 转换器结果寄存器 1	ADA0CR1						√
FFFF213H	A/D 转换器结果寄存器 1H	ADA0CR1H			√		无定义	
FFFF214H	A/D 转换器结果寄存器 2	ADA0CR2					√	无定义
FFFF215H	A/D 转换器结果寄存器 2H	ADA0CR2H			√		无定义	
FFFF216H	A/D 转换器结果寄存器 3	ADA0CR3					√	无定义
FFFF217H	A/D 转换器结果寄存器 3H	ADA0CR3H			√		无定义	
FFFF218H	A/D 转换器结果寄存器 4	ADA0CR4					√	无定义
FFFF219H	A/D 转换器结果寄存器 4H	ADA0CR4H			√		无定义	

地址	功能寄存器名称	符号	R/W	可操作位数			默认值	
				1	8	16		
FFFFF21AH	A/D 转换器结果寄存器 5	ADA0CR5	R			√	无定义	
FFFFF21BH	A/D 转换器结果寄存器 5H	ADA0CR5H			√		无定义	
FFFFF21CH	A/D 转换器结果寄存器 6	ADA0CR6				√	无定义	
FFFFF21DH	A/D 转换器结果寄存器 6H	ADA0CR6H			√		无定义	
FFFFF21EH	A/D 转换器结果寄存器 7	ADA0CR7				√	无定义	
FFFFF21FH	A/D 转换器结果寄存器 7H	ADA0CR7H			√		无定义	
FFFFF220H	A/D 转换器结果寄存器 8	ADA0CR8				√	无定义	
FFFFF221H	A/D 转换器结果寄存器 8H	ADA0CR8H			√		无定义	
FFFFF222H	A/D 转换器结果寄存器 9	ADA0CR9				√	无定义	
FFFFF223H	A/D 转换器结果寄存器 9H	ADA0CR9H			√		无定义	
FFFFF224H	A/D 转换器结果寄存器 10	ADA0CR10				√	无定义	
FFFFF225H	A/D 转换器结果寄存器 10H	ADA0CR10H			√		无定义	
FFFFF226H	A/D 转换器结果寄存器 11	ADA0CR11				√	无定义	
FFFFF227H	A/D 转换器结果寄存器 11H	ADA0CR11H			√		无定义	
FFFFF228H	A/D 转换器结果寄存器 12	ADA0CR12				√	无定义	
FFFFF229H	A/D 转换器结果寄存器 12H	ADA0CR12H			√		无定义	
FFFFF22AH	A/D 转换器结果寄存器 13	ADA0CR13				√	无定义	
FFFFF22BH	A/D 转换器结果寄存器 13H	ADA0CR13H			√		无定义	
FFFFF22CH	A/D 转换器结果寄存器 14	ADA0CR14				√	无定义	
FFFFF22DH	A/D 转换器结果寄存器 14H	ADA0CR14H			√		无定义	
FFFFF22EH	A/D 转换器结果寄存器 15	ADA0CR15				√	无定义	
FFFFF22FH	A/D 转换器结果寄存器 15H	ADA0CR15H			√		无定义	
FFFFF300H	按键返回模式寄存器	KRM		R/W	√	√		00H
FFFFF308H	选择器操作控制寄存器 0	SELCNT0			√	√		00H
FFFFF318H	噪声消除控制寄存器	NFC		√	√		00H	
FFFFF400H	端口 0	P0		√	√		无定义	
FFFFF402H	端口 1	P1		√	√		无定义	
FFFFF406H	端口 3	P3				√	无定义	
FFFFF406H	端口 3L	P3L		√	√		无定义	
FFFFF407H	端口 3H	P3H		√	√		无定义	
FFFFF408H	端口 4	P4		√	√		无定义	
FFFFF40AH	端口 5	P5		√	√		无定义	
FFFFF40EH	端口 7L	P7L		√	√		无定义	
FFFFF40FH	端口 7H	P7H		√	√		无定义	
FFFFF412H	端口 9	P9				√	无定义	
FFFFF412H	端口 9L	P9L		√	√		无定义	
FFFFF413H	端口 9H	P9H		√	√		无定义	
FFFFF420H	端口模式寄存器 0	PM0		√	√		FFH	
FFFFF422H	端口模式寄存器 1	PM1		√	√		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	

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
FFFFF428H	端口模式寄存器 4	PM4	R/W	√	√		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
FFFFF432H	端口模式寄存器 9	PM9				√	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
FFFFF442H	端口模式控制寄存器 1	PMC1		√	√		00H
FFFFF446H	端口模式控制寄存器 3	PMC3				√	0000H
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 选项寄存器 0	TQ0OPT0	√	√		00H	
FFFFF546H	TMQ0 捕捉/比较寄存器 0	TQ0CCR0			√	0000H	
FFFFF548H	TMQ0 捕捉/比较寄存器 1	TQ0CCR1			√	0000H	
FFFFF54AH	TMQ0 捕捉/比较寄存器 2	TQ0CCR2			√	0000H	
FFFFF54CH	TMQ0 捕捉/比较寄存器 3	TQ0CCR3			√	0000H	
FFFFF54EH	TMQ0 计数器读取缓存寄存器	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 选项寄存器 0	TP0OPT0		√	√		00H
FFFFF596H	TMP0 捕捉/比较寄存器 0	TP0CCR0				√	0000H
FFFFF598H	TMP0 捕捉/比较寄存器 1	TP0CCR1				√	0000H
FFFFF59AH	TMP0 计数器读取缓存寄存器	TP0CNT		R		√	0000H

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
FFFF5A0H	TMP1 控制寄存器 0	TP1CTL0	R/W	√	√		00H
FFFF5A1H	TMP1 控制寄存器 1	TP1CTL1		√	√		00H
FFFF5A2H	TMP1 I/O 控制寄存器 0	TP1IOC0		√	√		00H
FFFF5A3H	TMP1 I/O 控制寄存器 1	TP1IOC1		√	√		00H
FFFF5A4H	TMP1 I/O 控制寄存器 2	TP1IOC2		√	√		00H
FFFF5A5H	TMP1 选项寄存器 0	TP1OPT0		√	√		00H
FFFF5A6H	TMP1 捕捉/比较寄存器 0	TP1CCR0				√	0000H
FFFF5A8H	TMP1 捕捉/比较寄存器 1	TP1CCR1				√	0000H
FFFF5AAH	TMP1 计数器读取缓存寄存器	TP1CNT		R			√
FFFF5B0H	TMP2 控制寄存器 0	TP2CTL0	R/W	√	√		00H
FFFF5B1H	TMP2 控制寄存器 1	TP2CTL1		√	√		00H
FFFF5B2H	TMP2 I/O 控制寄存器 0	TP2IOC0		√	√		00H
FFFF5B3H	TMP2 I/O 控制寄存器 1	TP2IOC1		√	√		00H
FFFF5B4H	TMP2 I/O 控制寄存器 2	TP2IOC2		√	√		00H
FFFF5B5H	TMP2 选项寄存器 0	TP2OPT0		√	√		00H
FFFF5B6H	TMP2 捕捉/比较寄存器 0	TP2CCR0				√	0000H
FFFF5B8H	TMP2 捕捉/比较寄存器 1	TP2CCR1				√	0000H
FFFF5BAH	TMP2 计数器读取缓存寄存器	TP2CNT		R			√
FFFF5C0H	TMP3 控制寄存器 0	TP3CTL0	R/W	√	√		00H
FFFF5C1H	TMP3 控制寄存器 1	TP3CTL1		√	√		00H
FFFF5C2H	TMP3 I/O 控制寄存器 0	TP3IOC0		√	√		00H
FFFF5C3H	TMP3 I/O 控制寄存器 1	TP3IOC1			√		00H
FFFF5C4H	TMP3 I/O 控制寄存器 2	TP3IOC2		√	√		00H
FFFF5C5H	TMP3 选项寄存器 0	TP3OPT0		√	√		00H
FFFF5C6H	TMP3 捕捉/比较寄存器 0	TP3CCR0				√	0000H
FFFF5C8H	TMP3 捕捉/比较寄存器 1	TP3CCR1				√	0000H
FFFF5CAH	TMP3 计数器读取缓存寄存器	TP3CNT		R			√
FFFF610H	TMQ1 控制寄存器 0	TQ1CTL0	R/W	√	√		00H
FFFF611H	TMQ1 控制寄存器 1	TQ1CTL1		√	√		00H
FFFF612H	TMQ1 I/O 控制寄存器 0	TQ1IOC0		√	√		00H
FFFF613H	TMQ1 I/O 控制寄存器 1	TQ1IOC1		√	√		00H
FFFF614H	TMQ1 I/O 控制寄存器 2	TQ1IOC2		√	√		00H
FFFF615H	TMQ1 定时器选项寄存器 0	TQ1OPT0		√	√		00H
FFFF616H	TMQ1 捕捉/比较寄存器 0	TQ1CCR0				√	0000H
FFFF618H	TMQ1 捕捉/比较寄存器 1	TQ1CCR1				√	0000H
FFFF61AH	TMQ1 捕捉/比较寄存器 2	TQ1CCR2				√	0000H
FFFF61CH	TMQ1 捕捉/比较寄存器 3	TQ1CCR3			√	0000H	
FFFF61EH	TMQ1 计数器读取缓存寄存器	TQ1CNT	R			√	0000H
FFFF680H	钟表定时器操作模式寄存器	WTM	R/W	√	√		00H
FFFF690H	TMM0 控制寄存器 0	TM0CTL0		√	√		00H
FFFF694H	TMM0 比较寄存器 0	TM0CMP0				√	0000H
FFFF6C0H	振荡稳定时间选择寄存器	OSTS			√		06H

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
FFFFF6C1H	PLL 锁止时间指定寄存器	PLLS	R/W		√		03H
FFFFF6D0H	看门狗定时器模式寄存器2	WDTM2		√	√		67H
FFFFF6D1H	看门狗定时器使能寄存器r	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
FFFFF810H	DMA 触发源寄存器 0	DTFR0		√	√		00H
FFFFF812H	DMA 触发源寄存器 1	DTFR1		√	√		00H
FFFFF814H	DMA 触发源寄存器 2	DTFR2		√	√		00H
FFFFF816H	DMA 触发源寄存器 3	DTFR3		√	√		00H
FFFFF820H	节电模式寄存器	PSMR		√	√		00H
FFFFF824H	锁寄存器	LOCKR		R	√	√	
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	低电压检测电平选择寄存器	LVIS			√		00H
FFFFF892H	内部RAM数据状态寄存器	RAMS		√	√		01H
FFFFF8B0H	预分频模式寄存器 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		√	
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

注意事项 更多关于 OCDM 寄存器的细节，敬请参阅第二十五章 片上调试功能。

地址	功能寄存器名称	符号	R/W	可操作位数			默认值	
				1	8	16		
FFFFFFA16H	UARTA1 接收数据寄存器	UA1RX	R		√		FFH	
FFFFFFA17H	UARTA1 发送数据寄存器	UA1TX	R/W		√		FFH	
FFFFFFA20H	UARTA2 控制寄存器 0	UA2CTL0		√	√		10H	
FFFFFFA21H	UARTA2 控制寄存器 1	UA2CTL1			√		00H	
FFFFFFA22H	UARTA2 控制寄存器 2	UA2CTL2			√		FFH	
FFFFFFA23H	UARTA2 选项控制寄存器 0	UA2OPT0		√	√		14H	
FFFFFFA24H	UARTA2 状态寄存器	UA2STR		√	√		00H	
FFFFFFA26H	UARTA2 接收数据寄存器	UA2RX		R		√		FFH
FFFFFFA27H	UARTA2 发送数据寄存器	UA2TX		R/W		√		FFH
FFFFFFB00H	TIP00 引脚噪声消除控制寄存器	P00NFC		√	√		00H	
FFFFFFB04H	TIP01 引脚噪声消除控制寄存器	P01NFC		√	√		00H	
FFFFFFB08H	TIP10 引脚噪声消除控制寄存器	P10NFC		√	√		00H	
FFFFFFB0CH	TIP11 引脚噪声消除控制寄存器	P11NFC		√	√		00H	
FFFFFFB10H	TIP20 引脚噪声消除控制寄存器	P20NFC		√	√		00H	
FFFFFFB14H	TIP21 引脚噪声消除控制寄存器	P21NFC		√	√		00H	
FFFFFFB18H	TIP30 引脚噪声消除控制寄存器	P30NFC		√	√		00H	
FFFFFFB1CH	TIP31 引脚噪声消除控制寄存器	P31NFC		√	√		00H	
FFFFFFB50H	TIQ00 引脚噪声消除控制寄存器	Q00NFC		√	√		00H	
FFFFFFB54H	TIQ01 引脚噪声消除控制寄存器	Q01NFC		√	√		00H	
FFFFFFB58H	TIQ02 引脚噪声消除控制寄存器	Q02NFC		√	√		00H	
FFFFFFB5CH	TIQ03 引脚噪声消除控制寄存器	Q03NFC		√	√		00H	
FFFFFFB60H	TIQ10 引脚噪声消除控制寄存器	Q10NFC		√	√		00H	
FFFFFFB64H	TIQ11 引脚噪声消除控制寄存器	Q11NFC		√	√		00H	
FFFFFFB68H	TIQ12 引脚噪声消除控制寄存器	Q12NFC		√	√		00H	
FFFFFFB6CH	TIQ13 引脚噪声消除控制寄存器	Q13NFC		√	√		00H	
FFFFFFC00H	外部中断下降沿指定寄存器 0	INTF0		√	√		00H	
FFFFFFC02H	外部中断下降沿指定寄存器 1	INTF1		√	√		00H	
FFFFFFC06H	外部中断下降沿指定寄存器 3	INTF3				√	0000H	
FFFFFFC06H	外部中断下降沿指定寄存器 3L	INTF3L		√	√		00H	
FFFFFFC13H	外部中断下降沿指定寄存器 9H	INTF9H		√	√		00H	
FFFFFFC20H	外部中断上升沿指定寄存器 0	INTR0		√	√		00H	
FFFFFFC22H	外部中断上升沿指定寄存器 1	INTR1		√	√		00H	
FFFFFFC26H	外部中断上升沿指定寄存器 3	INTR3				√	0000H	
FFFFFFC26H	外部中断上升沿指定寄存器 3L	INTR3L		√	√		00H	
FFFFFFC33H	外部中断上升沿指定寄存器 9H	INTR9H		√	√		00H	
FFFFFFC40H	上拉电阻选项寄存器 0	PU0		√	√		00H	
FFFFFFC42H	上拉电阻选项寄存器 1	PU1	√	√		00H		
FFFFFFC46H	上拉电阻选项寄存器 3	PU3			√	0000H		
FFFFFFC46H	上拉电阻选项寄存器 3L	PU3L	√	√		00H		
FFFFFFC47H	上拉电阻选项寄存器 3H	PU3H	√	√		00H		
FFFFFFC48H	上拉电阻选项寄存器 4	PU4	√	√		00H		
FFFFFFC4AH	上拉电阻选项寄存器 5	PU5	√	√		00H		

地址	功能寄存器名称	符号	R/W	可操作位数			默认值	
				1	8	16		
FFFFFC52H	上拉电阻选项寄存器 9	PU9	R/W			√	0000H	
FFFFFC52H	上拉电阻选项寄存器 9L	PU9L		√	√		00H	
FFFFFC53H	上拉电阻选项寄存器 9H	PU9H		√	√		00H	
FFFFFD00H	CSIB0 控制寄存器 0	CB0CTL0	R/W	√	√		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	R/W	√	√		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 特殊寄存器

特殊寄存器是为了防止由意外的程序循环引起的数据非法写入而受到保护的寄存器。V850ES/HG2 中共有如下七种特殊寄存器。

- 节电控制寄存器 (PSC)
- 处理器时钟控制寄存器 (PCC)
- 时钟监视器模式寄存器 (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)

此外，器件还提供了一个命令寄存器 (PRCDM) 来阻截对特殊寄存器的非法写操作，这样应用系统就不会因为意外的程序循环而停止工作了。对特殊寄存器的正常写操作是通过特定的流程实现的，非法的写操作将被记录在系统状态寄存器 (SYS) 中（由于噪声，瞬时电压降等原因，甚至当选项数据的读取操作（地址：007AH）非法时也报告）。

(1) 向特殊寄存器中写入数据

请按照以下流程向特殊寄存器中写入数据。

- <1> 禁止 DMA 操作
- <2> 将要向特殊寄存器中写入的数据储存在通用寄存器中
- <3> 将步骤<2>中存储的数据写入 PRCMD 寄存器
- <4> 向特殊寄存器写入数据(使用以下指令):
 - 存指令(ST/STT 指令)
 - 位操作指令(SET1/CLR1/NOT1 指令)
- (<5> ~ <9>插入 NOP 指令 (5 条))^註
- <10> 按需要允许 DMA 操作

[例] 写 PSC 寄存器 (设置待机模式)

- ST.B r11, PSMD[r0] ; 设置 PSMD 寄存器 (设置 IDLE1, IDLE2 和 STOP 模式)。
- <1>CLR1 0, DCHCn[r0] ; 禁止 DMA 操作。 n = 0 ~ 3
- <2>MOV0x02, r10
- <3>ST.B r10, PRCMD[r0] ; 写 PRCMD 寄存器。
- <4>ST.B r10, PSC[r0] ; 设置 PSC 寄存器。
- <5>NOP^註 ; 哑指令
- <6>NOP^註 ; 哑指令
- <7>NOP^註 ; 哑指令
- <8>NOP^註 ; 哑指令
- <9>NOP^註 ; 哑指令
- <10>SET1 0, DCHCn[r0] ; 允许 DMA 操作。 n = 0 ~ 3
(下一条指令)

读取特殊寄存器时不需经过特定流程。

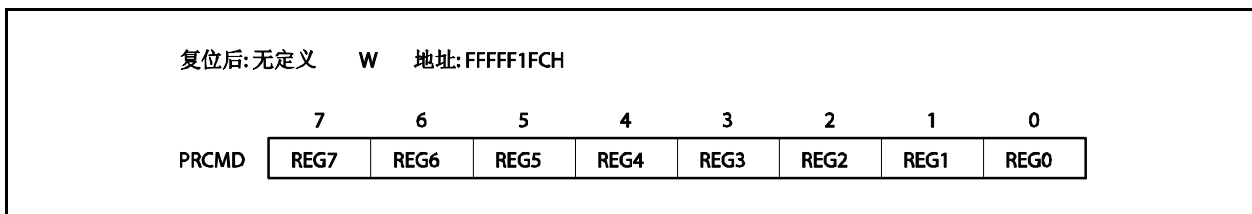
注 设置 IDLE 1 模式, IDLE 2 模式, STOP 模式, 或副 STOP 模式 (通过将 PSC.STP 位置 1) 后, 要立即插入 5 条 NOP 指令。

- 注意事项**
1. 当对命令寄存器进行写操作时, 系统将不响应中断。存储指令应该是按上述的第<3>和<4>步骤连续执行的。若在步骤<3>和<4>之间执行了其他指令, 而这个指令又进行了中断的响应, 则将打乱上述的顺序导致设备误操作
 2. 尽管对 PRCMD 写入的数据为虚数据, 但还是请使用与设置特殊寄存器(例子中的<4>)的通用寄存器相同的寄存器来向 PRCMD 寄存器(例子中的<3>)写入虚数据。这个过程也同样可以用于使用通用寄存器进行寻址的操作中。

(2) 命令寄存器 (PRCMD)

命令寄存器(PRCMD)用于保护那些会严重影响系统运行状态的寄存器不被轻易写入，从而使应用系统不会因为意外的程序循环而停止工作。除非数据先被写入 PRCMD，否则向特殊寄存器中写入的数据是无效的。数据只有通过特定的流程才可能被写入特殊寄存器，这样就能够防止对特殊寄存器的非法写操作。

只能对 PRCMD 进行 8 位写操作（若对其进行读取，则所读数据不确定）。



(3) 系统状态寄存器 (SYS)

该寄存器中包含了用于指定整个系统的操作状态的状态标志。
 该寄存器支持 8 位或 1 位读写方式。
 将寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF802H							
	7	6	5	4	3	2	1 0
SYS	0	0	0	0	0	0	PRERR
	PRERR		保护性错误				
	0	保护性错误未发生					
	1	保护性错误发生					

PRERR 标志会在以下情况下改变。

(a) 设置条件 (PRERR 标志 = 1)

- (i) 当向特殊寄存器写入数据之前没有对 PRCMD 寄存器进行写入时（在执行 **3.4.7 (1) 向特殊寄存器中写入数据** 中的步骤<4>之前没有执行步骤<3>）
- (ii) 当对 PRCMD 寄存器进行写入操作后，没有进行对特殊寄存器的写入，而是写入了片上外围 I/O 寄存器时（如果 **3.4.7 (1) 向特殊寄存器中写入数据** 中的步骤<3>不是对特殊寄存器的写入操作）。

备注 在写入 PRCMD 寄存器和特殊寄存器的两个操作之间，若插入了对片上外围 I/O 寄存器的读取操作（位操作除外），PRERR 标志位不会被置 1，数据也可以正常写入特殊寄存器。

(b) 清零条件 (PRERR 标志 = 0)

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

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

3.4.8 注意事项

(1) 需要首先设置的寄存器

在使用 V850ES/HG2 器件时，请务必首先对以下寄存器进行设置。

- 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)

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

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

(a) 系统等待控制寄存器 (VSWC)

VSWC 寄存器用于控制总线访问片上外围 I/O 寄存器的等待时间。

访问一个片上外围 I/O 寄存器需要 3 个时钟周期的时间(不包含等待周期)。V850ES/HG2 需要根据不同的操作频率插入不同的等待周期数。因此，请根据所使用的时钟，将以下对应的数值设置到 VSWC 寄存器中。

VSWC 寄存器支持 8 位读写方式（地址：FFFFF06EH，默认值：77H）。

操作频率 (f _{CLK})	VSWC 的设置值	等待周期数
32 kHz ≤ f _{CLK} < 16.6 MHz	00H	0（无等待周期数）
16.6 MHz ≤ f _{CLK} ≤ 20 MHz	01H	1

(b) 片上调试模式寄存器 (OCDM)

关于更多细节，敬请参阅第二十五章 片上调试功能。

(c) 看门狗模式寄存器 2 (WDTM2)

WDTM2 寄存器用于设置看门狗定时器 2 的溢出时间和操作时钟。

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

关于更多细节，敬请参阅第十章 看门狗定时器 2 的功能。

(2) 访问特定的片上外围 I/O 寄存器

本产品具有两种内部系统总线。

一种是 CPU 总线，另一种是用于与低速外围硬件进行通信的外围总线。

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

对于那些对实时处理要求较高的应用来说，一定要考虑上述的情况。

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

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

外围功能	寄存器名称	访问	k
16 位定时器/事件计数器 P (TMP) (n = 0 ~ 3)	TPnCNT	读	1 或 2
	TPnCCR0, TPnCCR1	写	<ul style="list-style-type: none"> • 第一次访问: 无等待 • 连续写入: 3 或 4
		读	1 或 2
16 位定时器/事件计数器 Q (TMQ) (m = 0, 1)	TQmCNT	读	1 或 2
	TQmCCR0 ~ TQmCCR3	写	<ul style="list-style-type: none"> • 第一次访问: 无等待 • 连续写入: 3 或 4
		读	1 或 2
看门狗定时器 2 (WDT2)	WDTM2	写 (当 WDT2 工作时)	3
A/D 转换器	ADA0M0	读	1 或 2
	ADA0CR0 ~ ADA0CR15	读	1 或 2
	ADA0CR0H ~ ADA0CR15H	读	1 或 2

$$\text{访问所需的时钟周期数} = 3 + i + j + (2 + j) \times k$$

注意事项 以下状态中，禁止访问上述的寄存器。如果这时产生了等待状态，那么只有复位才可退出等待状态。

- 当主时钟振荡停止，系统工作于副时钟时
- 当 CPU 工作于内部振荡器时钟时

备注 i: VSWC 寄存器高 4 位的值 (0 或 1)
j: VSWC 寄存器低 4 位的值 (0 或 1)

第四章 端口功能

4.1 特点

- I/O 端口： 84
- 端口引脚复用作其它外围功能 I/O 引脚
- 输入或输出模式可以以位为单位设置

4.2 端口的基本配置

V850ES/HG2 由端口 0, 1, 3 ~ 5, 7, 9, CM, CS, CT 和 DL 共 84 个输入/输出端口组成。端口配置情况如下图所示。

图 4-1. 端口配置

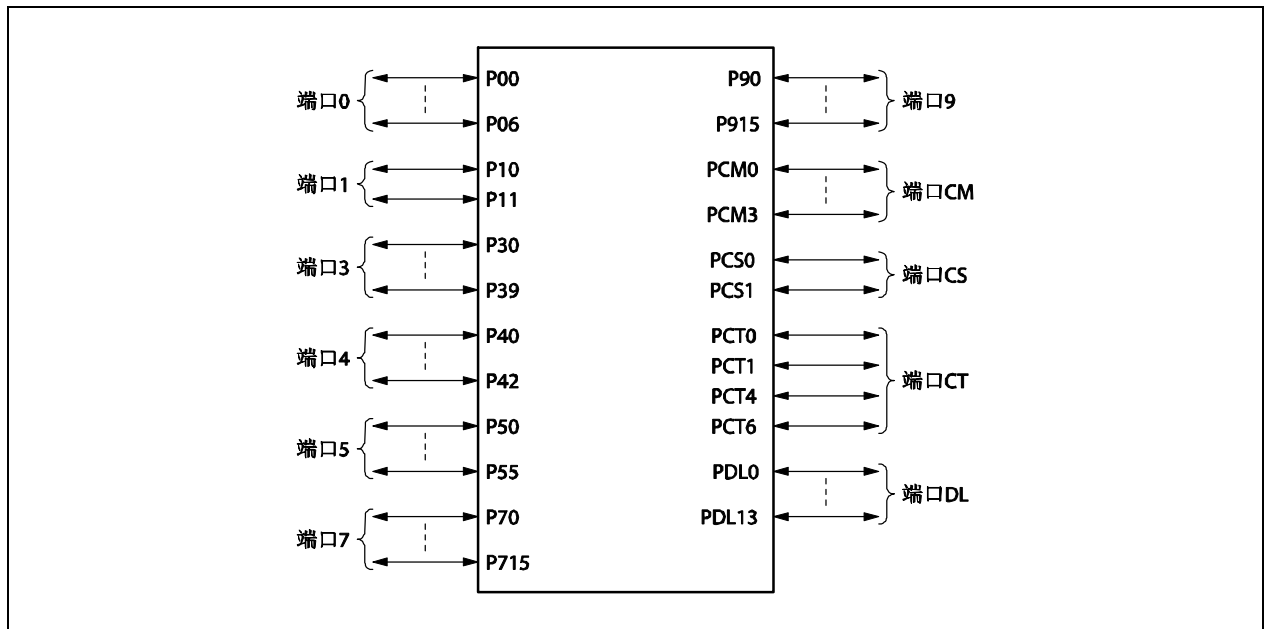


表 4-1. 端口配置

项目	配置
控制寄存器	端口模式寄存器(PMn: n = 0, 1, 3, 4, 5, 7L, 7H, 9, CM, CS, CT 或 DL)
	端口模式控制寄存器(PMCn: n = 0, 1, 3, 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, 1, 3, 4, 5 或 9)
端口	84

表 4-2. 引脚 I/O 缓冲器电源

电源	对应引脚
AVREF0	端口 7
BVDD	端口 CM, 端口 CS, 端口 CT, 端口 DL
EVDD	端口 0, 端口 1, 端口 3, 端口 4, 端口 5, 端口 9, RESET

4.3 端口功能

4.3.1 端口功能的操作

端口操作依据如下所述的输入或输出模式的设置而不同。

(1) 写入 I/O 端口

(a) 输出模式

通过使用传输指令可以将值写入输出锁存器。输出锁存器的内容从引脚输出。一旦数据被写入输出锁存器，数据将一直保留直到新数据写入到输出锁存器。

(b) 输入模式

通过使用传输指令可以将值写入输出锁存器。然而，因为输出缓冲器关闭，引脚状态保持不变。一旦数据被写入输出锁存器，数据将一直保留直到新数据写入到输出锁存器。

注意事项 尽管 1 位存储器操作指令操作了 1 位，该存储器访问 8 位单元的端口，如果该端口具有输入和输出引脚，则在输入模式下设置的引脚输出锁存器的内容没有定义，甚至引脚不能进行操作。

(2) 从 I/O 端口读取

(a) 输出模式

通过传输指令可以读取输出锁存器的内容。而输出锁存器的内容不会改变。

(b) 输入模式

通过传输指令可以读取引脚状态。而输出锁存器的内容不会改变。

(3) I/O 端口的操作

(a) 输出模式

对输出锁存器的内容执行操作，并将结果写入输出锁存器。输出锁存器的内容从引脚输出。一旦数据被写入输出锁存器，数据将一直保留直到新数据写入到输出锁存器。

(b) 输入模式

输出锁存器的内容没有定义。然而，因为输出缓冲器关闭，引脚状态保持不变。

注意事项 尽管 1 位存储器操作指令操作了 1 位，该存储器访问 8 位单元的端口，如果该端口具有输入和输出引脚，则在输入模式下设置的引脚输出锁存器的内容没有定义，甚至引脚不能进行操作。

4.3.2 关于设置端口引脚的注释

- (1) 根据产品的不同，端口数量和复用功能也会不同，在复位后，设置与不可用端口和复用功能相关的寄存器为一个值。
- (2) 采用如下步骤设置端口寄存器。

- <1> 设置端口功能控制寄存器 n (PFCn) 和端口功能控制扩展寄存器 n (PFCEn)。
- <2> 设置端口模式控制寄存器 n (PMcn)。
- <3> 设置外部中断下降沿指定寄存器 n (INTFn) 和外部中断上升沿指定寄存器 n (INTRn)。

如果在设置 PMcn 寄存器之后设置 PFCn 和 PFCEn 寄存器，则当 PFCn 和 PFCEn 寄存器被设置时，意外的外围功能引脚也可能被设置。

- (3) 仅在输入模式下 (PMn 寄存器的 PMnm 位 = 1)，Pun 寄存器的 PUnm 位 (该位连接一个片上上拉电阻) 有效。在输出模式下 (PMn 寄存器的 PMnm 位 = 0)，硬件将片上上拉电阻断开。
- (4) 读取引脚电平和端口锁存器由端口模式寄存器(PMn)控制。当使用复用功能时，同样也是这样。
- (5) 当施密特缓冲器在端口模式下被读取时，施密特(SHMT)触发输入缓冲器不作为 SHMT 缓冲器而工作。

4.3.3 端口 0

端口 0 是一个可以以位为单位控制 I/O 设置的 7 位端口 (P00 ~ P06)。

(1) 端口 0 的功能

- 可以以位为单位指定端口的输入/输出数据。
由端口寄存器 0 (P0) 指定
- 可以以位为单位指定端口的输入/输出模式。
由端口模式寄存器指定 0 (PM0)
- 可以以位为单位指定端口模式或控制模式 (复用功能)。
由端口模式控制寄存器指定 0 (PMC0)
- 可以以位为单位指定控制模式 1 或控制模式 2。
由端口功能控制寄存器指定 0 (PFC0)
- 可以以位为单位连接片上上拉电阻。
由上拉电阻选项寄存器指定 0 (PU0)

端口 0 包含以下复用功能引脚。

表 4-3. 端口 0 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	模块类型
P00	6	TP31/TOP31	I/O	-	G-1
P01	7	TP30/TOP30			G-1
P02	17	NMI ^{※1}			L-1
P03	18	INTP0/ADTRG			N-1
P04	19	INTP1			L-1
P05	20	INTP2/DRST ^{※2}			AA-1
P06	21	INTP3			L-2

注 1. NMI 引脚用作 P02 引脚的复用引脚，在复位后其用作 P02 引脚。要启用 NMI 引脚，设置 PMC0.PMC02 位为 1。NMI 引脚的初始设置为“无边沿检测”。使用 INTF0 和 INTR0 寄存器选择 NMI 引脚的有效沿。

2. P05 引脚的可功能复用为片上调试功能。外部复位后，P05/INTP2/DRST 引脚的初始设置为片上调试引脚 (DRST)。若使用 P05 引脚作为端口引脚，而不是片上调试引脚，应采取以下措施。

<1> 将 OCDM.OCDM0 位 (特殊寄存器) 清零。

<2> 采取上述措施后，将 P05/INTP2/DRST 引脚固定到低电平。

当不使用片上调试功能，在采取上述措施之前，输入高电平到 DRST 引脚会引起故障 (CPU 死锁)。在处理 P05 引脚时要十分小心。

当没有输入高电平到 P05/INTP2/DRST 引脚时 (当该引脚固定为低电平)，不需要操作 OCDM.OCDM0 位。

因为下拉电阻 (30 kΩ TYP.) 连接到 P05/INTP2/DRST 引脚的缓冲区时，该引脚不必通过外部源固定为低电平。通过将 OCDM0 位清零，可以断开下拉电阻。

注意事项 P00 ~ P06 引脚在复用功能的输入模式下具有滞后性，但在端口模式下不具有滞后性。

(2) 寄存器

(a) 端口寄存器 0 (P0)

端口寄存器 0 (P0) 是控制读取引脚电平和写入输出电平的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后：无定义		R/W	地址： FFFFF400H					
	7	6	5	4	3	2	1	0
P0	0	P06	P05	P04	P03	P02	P01	P00
	P0n	输出数据的控制（输出模式）(n = 0 ~ 6)						
	0	输出 0。						
	1	输出 1。						

(b) 端口模式寄存器 0 (PM0)

该寄存器是指定输入或输出模式的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

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/输出模式的控制(n = 0 ~ 6)						
	0	输出模式						
	1	输入模式						

(c) 端口模式控制寄存器 0 (PMC0)

该寄存器是指定端口模式或控制模式的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: 00H R/W 地址: FFFF440H

	7	6	5	4	3	2	1	0
PMC0	0	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	PMC00

PMC06	P06 引脚操作模式的指定
0	I/O 端口
1	INTP3 输入

PMC05	P05 引脚操作模式的指定
0	I/O 端口
1	INTP2/DRST 输入

PMC04	P04 引脚操作模式的指定
0	I/O 端口
1	INTP1 输入

PMC03	P03 引脚操作模式的指定
0	I/O 端口
1	INTP0/ADTRG 输入

PMC02	P02 引脚操作模式的指定
0	I/O 端口
1	NMI 输入

PMC01	P01 引脚操作模式的指定
0	I/O 端口
1	TIP30/TOP30 I/O

PMC00	P00 引脚操作模式的指定
0	I/O 端口
1	TIP31/TOP31 I/O

注意事项 无论 PMC05 位的值如何, 当 OCDM.OCDM0 位为 1 时, P05/INTP2/DRST 引脚都会变为 DRST 引脚。

(d) 端口功能控制寄存器 0 (PFC0)

该寄存器是指定控制模式 1 或控制模式 2 的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

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 P03 引脚处于控制模式下的操作模式的指定							
	0	INTP0 输入						
	1	ADTRG 输入						
	PFC01 P01 引脚处于控制模式下的操作模式的指定							
	0	TIP30 输入						
	1	TOP30 输出						
	PFC00 P00 引脚处于控制模式下的操作模式的指定							
	0	TIP31 输入						
	1	TOP31 输出						

(e) 上拉电阻选项寄存器 0 (PU0)

该寄存器是指定片上上拉电阻连接的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: 00H		R/W		地址: FFFFC40H				
	7	6	5	4	3	2	1	0
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00
	PU0n 片上上拉电阻连接的控制(n = 0 ~ 6)							
	0	不连接						
	1	连接						

4.3.4 端口 1

端口 1 是一个可以以位为单位控制 I/O 设置的 2 位端口 (P10, P11)。

(1) 端口 1 功能

- 可以以位为单位指定端口的输入/输出数据。
由端口寄存器 1 (P1) 指定
- 可以以位为单位指定端口的输入/输出模式。
由端口模式寄存器 1 (PM1) 指定
- 可以以位为单位指定端口模式或控制模式 (复用功能)。
由端口模式控制寄存器 1 (PMC1) 指定
- 可以以位为单位连接片上上拉电阻。
由上拉电阻选项寄存器 1 (PU1) 指定

端口 1 包含以下复用功能引脚。

表 4-4. 端口 1 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	模块类型
P10	3	INTP9	I/O	-	L-1
P11	4	INTP10			L-1

注意事项 P10 和 P11 引脚在复用功能的输入模式下具有滞后性，但在端口模式下不具有滞后性。

(2) 寄存器**(a) 端口寄存器 1 (P1)**

端口寄存器 1 (P1) 是控制读取引脚电平和写入输出电平的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: 无定义		R/W	地址: FFFFF402H					
	7	6	5	4	3	2	1	0
P1	0	0	0	0	0	0	P11	P10
P1n	输出数据的控制 (输出模式) (n = 0, 1)							
0	输出 0。							
1	输出 1。							

(b) 端口模式寄存器 1 (PM1)

该寄存器是指定输入或输出模式的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: FFH		R/W	地址: FFFFF422H					
	7	6	5	4	3	2	1	0
PM1	1	1	1	1	1	1	PM11	PM10
PM1n	输入/输出模式的控制 (n = 0, 1)							
0	输出模式							
1	输入模式							

(c) 端口模式控制寄存器 1 (PMC1)

该寄存器是指定端口模式或控制模式的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: 00H		R/W		地址: FFFFF442H				
	7	6	5	4	3	2	1	0
PMC1	0	0	0	0	0	0	PMC11	PMC10
	PMC11		P11 引脚操作模式的指定					
	0	I/O 端口						
	1	INTP10 输入						
	PMC10		P10 引脚操作模式的指定					
	0	I/O 端口						
	1	INTP9 输入						

(d) 上拉电阻选项寄存器 1 (PU1)

该寄存器是指定片上上拉电阻连接的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: 00H		R/W		地址: FFFFFC42H				
	7	6	5	4	3	2	1	0
PU1	0	0	0	0	0	0	PU11	PU10
	PU1n		片上上拉电阻连接的控制 (n = 0, 1)					
	0	不连接						
	1	连接						

4.3.5 端口 3

端口 3 是一个可以以位为单位控制 I/O 设置的 10 位端口（P30 ~ P39）。

(1) 端口 3 的功能

- 可以以位为单位指定端口的输入/输出数据。
由端口寄存器 3 (P3) 指定
- 可以以位为单位指定端口的输入/输出模式。
由端口模式寄存器 3 (PM3) 指定
- 可以以位为单位指定端口模式或控制模式（复用功能）。
由端口模式控制寄存器 3 (PMC3) 指定
- 可以以位为单位指定控制模式。
由端口功能控制寄存器 3 (PFC3) 和端口功能控制扩展寄存器 3L (PFCE3L) 指定
- 可以以位为单位连接片上上拉电阻。
由上拉电阻选项寄存器 3 (PU3) 指定

端口 3 包含以下复用功能引脚。

表 4-5. 端口 3 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	模块类型
P30	25	TXDA0	I/O	-	E-2
P31	26	RXDA0/INTP7			L-2
P32	27	ASCKA0/TIP00/TOP00/TOP01			U-13
P33	28	TIP01/TOP01			G-1
P34	29	TIP10/TOP10			G-1
P35	30	TIP11/TOP11			G-1
P36	31	-			C-1
P37	32	-			C-1
P38	35	TXDA2			E-2
P39	36	RXDA2/INTP8			L-2

注意事项 P31 ~ P35 和 P39 引脚在复用功能的输入模式下具有滞后性，但在端口模式下不具有滞后性。

(2) 寄存器**(a) 端口寄存器 3 (P3)**

端口寄存器 3 (P3) 是控制读取引脚电平和写入输出电平的 16 位寄存器。该寄存器支持 16 位读写方式。如果使用 P3 寄存器的高 8 位作为 P3H 寄存器，低 8 位作为 P3L 寄存器，这些寄存器支持 8 位或 1 位读写方式。

复位后: 无定义		R/W	地址: FFFFF406H, FFFFF407H					
	15	14	13	12	11	10	9	8
P3 (P3H [Ⓢ])	0	0	0	0	0	0	P39	P38
	7	6	5	4	3	2	1	0
(P3L)	P37	P36	P35	P34	P33	P32	P31	P30
P3n	输出数据的控制 (输出模式) (n = 0 ~ 9)							
0	输出 0。							
1	输出 1。							

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

(b) 端口模式寄存器 3 (PM3)

该寄存器是指定输入或输出模式的 16 位寄存器。该寄存器支持 16 位读写方式。如果使用 PM3 寄存器的高 8 位作为 PM3H 寄存器，低 8 位作为 PM3L 寄存器，这些寄存器支持 8 位或 1 位读写方式。

复位后: FFFFH		R/W	地址: FFFFF426H, FFFFF427H					
	15	14	13	12	11	10	9	8
PM3 (PM3H [Ⓢ])	1	1	1	1	1	1	PM39	PM38
	7	6	5	4	3	2	1	0
(PM3L)	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30
PM3n	I/O 模式的控制 (n = 0 ~ 9)							
0	输出模式							
1	输入模式							

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

(c) 端口模式控制寄存器 3 (PMC3)

该寄存器是指定端口模式或控制模式的 16 位寄存器。该寄存器支持 16 位读写方式。

如果使用 PMC3 寄存器的高 8 位作为 PMC3H 寄存器，低 8 位作为 PMC3L 寄存器，这些寄存器支持 8 位或 1 位读写方式。

(1/2)

复位后: 0000H	R/W	地址: FFFFF446H, FFFFF447H							
		15	14	13	12	11	10	9	8
PMC3 (PMC3H ^{※1})		0	0	0	0	0	0	PMC39	PMC38
		7	6	5	4	3	2	1	0
(PMC3L)		0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

PMC39	P39 引脚操作模式的指定
0	I/O 端口
1	RXDA2/INTP8 输入 ^{※2}

PMC38	P38 引脚操作模式的指定
0	I/O 端口
1	TXDA2 输出

PMC35	P35 引脚操作模式的指定
0	I/O 端口
1	TIP11/TOP11 I/O

PMC34	P34 引脚操作模式的指定
0	I/O 端口
1	TIP10/TOP10 I/O

- 注**
1. 要以 8 位 或 1 位为单位读写 PMC3 寄存器的位 8 ~ 15，需要将他们指定为 PMC3H 寄存器的位 0 ~ 7。
 2. INTP8 引脚功能复用作 RXDA2 引脚。为了作为 RXDA2 引脚使用，使复用功能 INTP8 引脚的边沿检测功能无效（通过固定 INTF3.INTF39 位和 INTR3.INTR39 位为 0）。为了作为 INTP8 引脚使用，停止 UARTA2 的接收操作（通过将 UA2CTL0.UA2RXE 位清零）。

PMC33	P33 引脚操作模式的指定
0	I/O 端口
1	TIP01/TOP01 I/O

PMC32	P32 引脚操作模式的指定
0	I/O 端口
1	ASCKA0/TIP00/TOP00/TOP01 I/O

PMC31	P31 引脚操作模式的指定
0	I/O 端口
1	RXDA0/INTP7 输入 ^注

PMC30	P30 引脚操作模式的指定
0	I/O 端口
1	TXDA0 输出

注 INTP7 引脚功能复用作 RXDA0 引脚。为了作为 RXDA0 引脚使用，使复用功能 INTP7 引脚的边沿检测功能无效（通过固定 INTF3.INTF31 位和 INTR3.INTR31 位为 0）。为了作为 INTP7 引脚使用，停止 UARTA0 的接收操作（通过将 UA0CTL0.UA0RXE 位清零）。

(d) 端口功能控制寄存器 3L (PFC3L)

该寄存器是指定控制模式 1, 2, 3 或 4 的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: 00H	R/W	地址: FFFF466H						
	7	6	5	4	3	2	1	0
PFC3L	0	0	PFC35	PFC34	PFC33	PFC32	0	0

备注 关于如何指定控制模式，敬请参阅 4.3.5 (2) (f) P3 引脚控制模式的设置。

(e) 端口功能控制扩展寄存器 3L (PFCE3L)

该寄存器是指定控制模式 1, 2, 3 或 4 的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: 00H	R/W	地址: FFFF706H						
	7	6	5	4	3	2	1	0
PFCE3L	0	0	0	0	0	PFCE32	0	0
<p>备注 关于如何指定控制模式, 敬请参阅 4.3.5 (2) (f) P3 引脚控制模式的设置。</p>								

(f) P3 引脚控制模式的设置

PFC35	P35 引脚控制模式的指定	
0	TIP11 输入	
1	TOP11 输出	

PFC34	P34 引脚控制模式的指定	
0	TIP10 输入	
1	TOP10 输出	

PFC33	P33 引脚控制模式的指定	
0	TIP01 输入	
1	TOP01 输出	

PFCE32	PFC32	P32 引脚控制模式的指定	
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)

该寄存器是指定片上上拉电阻连接的 16 位寄存器。该寄存器支持 16 位或 1 位读写方式。

如果使用 PU3 寄存器的高 8 位作为 PU3H 寄存器，低 8 位作为 PU3L 寄存器，这些寄存器支持 8 位或 1 位读写方式。

复位后: 0000H		R/W	地址: FFFFC46H, FFFFC47H					
	15	14	13	12	11	10	9	8
PU3 (PU3H [※])	0	0	0	0	0	0	PU39	PU38
	7	6	5	4	3	2	1	0
(PU3L)	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30
	片上上拉电阻连接的控制 (n = 0 ~ 9)							
PU3n								
0	不连接							
1	连接							

注 要以 8 位 或 1 位为单位读写 PU3 寄存器的位 8 ~ 15，需要将他们指定为 PU3H 寄存器的位 0 ~ 7。

4.3.6 端口 4

端口 4 是一个可以以位为单位控制 I/O 设置的 3 位端口（P40 ~ P42）。

(1) 端口 4 的功能

- 可以以位为单位指定端口的输入/输出数据。
由端口寄存器 4 (P4) 指定
- 可以以位为单位指定端口的输入/输出模式。
由端口模式寄存器 4 (PM4) 指定
- 可以以位为单位指定端口模式或控制模式（复用功能）。
由端口模式控制寄存器 4 (PMC4) 指定
- 可以以位为单位连接片上上拉电阻。
由上拉电阻选项寄存器 4 (PU4) 指定

端口 4 包含以下复用功能引脚。

表 4-6. 端口 4 复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	I/O	备注	模块类型
P40	22	SIB0	I/O	-	E-1
P41	23	SOB0			E-2
P42	24	SCKB0			E-3

注意事项 P40 和 P42 引脚在复用功能的输入模式下具有滞后性，但在端口模式下不具有滞后性。

(2) 寄存器**(a) 端口寄存器 4 (P4)**

端口寄存器 4 (P4) 是控制读取引脚电平和写入输出电平的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

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

(b) 端口模式寄存器 4 (PM4)

该寄存器是指定输入或输出模式的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

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/输出模式的控制 (n = 0 ~ 2)							
0	输出模式							
1	输入模式							

(c) 端口模式控制寄存器 4 (PMC4)

该寄存器是指定端口模式或控制模式的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

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		P42 引脚操作模式的指定					
	0	I/O 端口						
	1	SCKB0 输入/输出						
	PMC41		P41 引脚操作模式的指定					
	0	I/O 端口						
	1	SOB0 输出						
	PMC40		P40 引脚操作模式的指定					
	0	I/O 端口						
	1	SIB0 输入						

(d) 上拉电阻选项寄存器 4 (PU4)

该寄存器是指定片上上拉电阻连接的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

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		片上上拉电阻连接的控制 (n = 0 ~ 2)					
	0	不连接						
	1	连接						

4.3.7 端口 5

端口 5 是一个可以以位为单位控制 I/O 设置的 6 位端口 (P50 ~ P55)。

(1) 端口 5 的功能

- 可以以位为单位指定端口的输入/输出数据。
由端口寄存器 5 (P5) 指定
- 可以以位为单位指定端口的输入/输出模式。
由端口模式寄存器 5 (PM5) 指定
- 可以以位为单位指定端口模式或控制模式。
由端口模式控制寄存器 5 (PMC5) 指定
- 可以以位为单位指定控制模式。
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
- 可以以位为单位连接片上上拉电阻。
由上拉电阻选项寄存器 5 (PU5) 指定

端口 5 包含以下复用功能引脚。

表 4-7. 端口 5 复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	I/O	备注	模块类型
P50	37	KR0/TIQ01/TOQ01	I/O	-	U-4
P51	38	KR1/TIQ02/TOQ02			U-4
P52	39	KR2/TIQ03/TOQ03/DDI [※]			U-5
P53	40	KR3/TIQ00/TOQ00/DDO [※]			U-6
P54	41	KR4/DCK [※]			G-2
P55	42	KR5/DMS [※]			G-2

注 DDI, DDO, DCK 和 DMS 引脚用于片上调试功能。采用如下步骤将 DDI, DDO, DCK 和 DMS 引脚作为端口引脚, 而不是片上调试引脚使用。

- <1> 将 OCDM 寄存器 (特殊寄存器) 的 OCDM0 位清零。
<2> 采取上述措施后, 将 P05/INTP2/ $\overline{\text{DRST}}$ 引脚固定为低电平。

当不使用片上调试功能, 在采取上述措施之前, 输入高电平到 $\overline{\text{DRST}}$ 引脚会引起故障 (CPU 死锁)。在处理 P05 引脚时要十分小心。

当没有输入高电平到 P05/INTP2/ $\overline{\text{DRST}}$ 引脚时 (当该引脚固定为低电平), 不需要操作 OCDM.OCDM0 位。

因为下拉电阻 (30 k Ω TYP.) 连接到 P05/INTP2/ $\overline{\text{DRST}}$ 引脚的缓冲区时, 该引脚不必通过外部源固定为低电平。通过将 OCDM0 位清零, 可以断开下拉电阻。

注意事项 P50 ~ P55 引脚在复用功能的输入模式下具有滞后性, 但在端口模式下不具有滞后性。

(2) 寄存器**(a) 端口寄存器 5 (P5)**

端口寄存器 5 (P5) 是控制读取引脚电平和写入输出电平的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: 无定义	R/W	地址: FFFFF40AH						
	7	6	5	4	3	2	1	0
P5	0	0	P55	P54	P53	P52	P51	P50
P5n	输出数据的控制 (输出模式) (n = 0 ~ 5)							
0	输出 0。							
1	输出 1。							

(b) 端口模式寄存器 5 (PM5)

该寄存器是指定输入或输出模式的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

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 模式的控制 (n = 0 ~ 5)							
0	输出模式							
1	输入模式							

(c) 端口模式控制寄存器 5 (PMC5)

该寄存器是指定端口模式或控制模式的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

注意事项 当 **PFC5.PFC5n** 和 **PFCE5.PFCE5n** 位是默认值 (0) 时，如果由 **PMC5** 寄存器指定控制模式，则输出变成无定义。

因此，首先设置 **PFC5.PFC5n** 和 **PFCE5.PFCE5n** 位，然后将 **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 引脚操作模式的指定
0	I/O 端口
1	KR5 输入

PMC54	P54 引脚操作模式的指定
0	I/O 端口
1	KR4 输入

PMC53	P53 引脚操作模式的指定
0	I/O 端口
1	KR3/TIQ00/TOQ00 I/O

PMC52	P52 引脚操作模式的指定
0	I/O 端口
1	KR2/TIQ03/TOQ03 I/O

PMC51	P51 引脚操作模式的指定
0	I/O 端口
1	KR1/TIQ02/TOQ02 I/O

PMC50	P50 引脚操作模式的指定
0	I/O 端口
1	KR0/TIQ01/TOQ01 I/O

(d) 端口功能控制寄存器 5 (PFC5)

该寄存器是指定控制模式 1, 2, 3 或 4 的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

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	关于如何指定控制模式, 敬请参阅 4.3.7 (2) (f) P5 引脚控制模式的设置。							

(e) 端口功能控制扩展寄存器 5 (PFCE5)

该寄存器是指定控制模式 1, 2, 3 或 4 的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

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	关于如何指定控制模式, 敬请参阅 4.3.7 (2) (f) P5 引脚控制模式的设置。							

(f) P5 引脚控制模式的设置

注意事项 当 PFC5.PFC5n 和 PFCE5.PFCE5n 位是默认值 (0) 时, 如果由 PMC5 寄存器指定控制模式, 则输出不确定。

因此, 首先设置 PFC5.PFC5n 和 PFCE5.PFCE5n 位, 然后将 PMC5n 位设置为 1 以设置控制模式。

PFC55	P55 引脚控制模式的指定
0	禁止设置
1	KR5 输入

PFC54	P54 引脚控制模式的指定
0	禁止设置
1	KR4 输入

PFCE53	PFC53	P53 引脚控制模式的指定
0	0	禁止设置
0	1	TIQ00/KR3 [#] 输入
1	0	TOQ00 输出
1	1	禁止设置

PFCE52	PFC52	P52 引脚控制模式的指定
0	0	禁止设置
0	1	TIQ03/KR2 [#] 输入
1	0	TOQ03 输出
1	1	禁止设置

PFCE51	PFC51	P51 引脚控制模式的指定
0	0	禁止设置
0	1	TIQ02/KR1 [#] 输入
1	0	TOQ02 输出
1	1	禁止设置

PFCE50	PFC50	P50 引脚控制模式的指定
0	0	禁止设置
0	1	TIQ01/KR0 [#] 输入
1	0	TOQ01 输出
1	1	禁止设置

注 KRn 引脚用作 TIQ0m 引脚的复用引脚，为了作为 TIQ0m 引脚使用，使复用功能 KRn 引脚的按键返回检测功能无效（通过将 KRM.KRm_n 位为 0）。为了作为 KRn 引脚使用，使复用功能 TIQ0m 引脚的边沿检测功能无效(n = 0 ~ 3, m = 0 ~ 3)。

引脚名称	作为 TIQ0m 引脚使用	作为 KRn 引脚使用
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 TQ0IOC2 寄存器的 TQ0EES0, TQ0EES1 位 = 0 TQ0IOC2 寄存器的 TQ0ETS0, TQ0ETS1 位 = 0

(g) 上拉电阻选项寄存器 5 (PU5)

该寄存器是指定片上上拉电阻连接的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: 00H	R/W	地址: FFFFC4AH						
	7	6	5	4	3	2	1	0
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50
	片上上拉电阻连接的控制 (n = 0 ~ 5)							
	PU5n							
	0	不连接						
	1	连接						

4.3.8 端口 7

端口 7 是一个可以以位为单位控制 I/O 设置的 16 位端口 (P70 ~ P15)。

(1) 端口 7 的功能

- 可以以位为单位指定端口的输入/输出数据。
由端口寄存器 7H, 7L (P7H, P7L) 指定
- 可以以位为单位指定端口的输入/输出模式。
由端口模式寄存器 7H, 7L (PM7H, PM7L) 指定

端口 7 包含以下复用功能引脚。

表 4-8. 端口 7 复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	I/O	备注	模块类型
P70	100	ANI0	I/O	-	A-1
P71	99	ANI1			A-1
P72	98	ANI2			A-1
P73	97	ANI3			A-1
P74	96	ANI4			A-1
P75	95	ANI5			A-1
P76	94	ANI6			A-1
P77	93	ANI7			A-1
P78	92	ANI8			A-1
P79	91	ANI9			A-1
P710	90	ANI10			A-1
P711	89	ANI11			A-1
P712	88	ANI12			A-1
P713	87	ANI13			A-1
P714	86	ANI14			A-1
P715	85	ANI15			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) 是控制读取引脚电平和写入输出电平的 8 位寄存器。这些寄存器支持 8 位或 1 位读写方式。

这些寄存器不支持 16 位访问方式。

复位后: 无定义		R/W	地址: FFFFF40FH, FFFFF40EH					
	7	6	5	4	3	2	1	0
P7H	P715	P714	P713	P712	P711	P710	P79	P78
	7	6	5	4	3	2	1	0
P7L	P77	P76	P75	P74	P73	P72	P71	P70
P7n	输出数据的控制 (输出模式) (n = 0 ~ 15)							
0	输出 0。							
1	输出 1。							

注意事项 在 A/D 转换期间不要读取 P7H 和 P7L 寄存器。

(b) 端口模式寄存器 7H, 7L (PM7H, PM7L)

这些寄存器是指定输入或输出模式的 8 位寄存器。这些寄存器支持 8 位或 1 位读写方式。

这些寄存器不支持 16 位访问方式。

复位后: FFH		R/W	地址: FFFFF42FH, FFFFF42EH					
	7	6	5	4	3	2	1	0
PM7H	PM715	PM714	PM713	PM712	PM711	PM710	PM79	PM78
	7	6	5	4	3	2	1	0
PM7L	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70
PM7n	I/O 模式的控制 (n = 0 ~ 15)							
0	输出模式							
1	输入模式							

注意事项 当使用 P7n 引脚的复用功能时, 需要将 PM7n 位置 1。

4.3.9 端口 9

端口 9 是一个可以以位为单位控制 I/O 设置的 9 位或 16 位端口（P90 ~ P915）。

(1) 端口 9 的功能

- 可以以位为单位指定端口的输入/输出数据。
由端口寄存器 9 (P9) 指定
- 可以以位为单位指定端口的输入/输出模式。
由端口模式寄存器 9 (PM9) 指定
- 可以以位为单位指定端口模式或控制模式（复用功能）。
由端口模式控制寄存器 9 (PMC9) 指定
- 可以以位为单位指定控制模式。
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
- 可以以位为单位连接片上上拉电阻。
由上拉电阻选项寄存器 9 (PU9) 指定

端口 9 包含以下复用功能引脚。

表 4-9. 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	模块类型
P90	43	KR6/TXDA1	I/O	-	U-12
P91	44	KR7/RXDA1			U-7
P92	45	TIQ11/TOQ11			U-11
P93	46	TIQ12/TOQ12			U-11
P94	47	TIQ13/TOQ13			U-11
P95	48	TIQ10/TOQ10			U-11
P96	49	TIP21/TOP21			U-9
P97	50	SIB1/TIP20/TOP20			U-8
P98	51	SOB1			G-3
P99	52	SCKB1			G-5
P910	53	-			C-1
P911	54	-			C-1
P912	55	-			C-1
P913	56	INTP4/PCL			W-1
P914	57	INTP5			N-2
P915	58	INTP6			N-2

注意事项 P90 ~ P97, P99 和 P913 ~ P915 引脚在复用功能的输入模式下具有滞后性，但在端口模式下不具有滞后性。

(2) 寄存器**(a) 端口寄存器 9 (P9)**

端口寄存器 9 (P9) 是控制读取引脚电平和写入输出电平的 16 位寄存器。该寄存器支持 16 位读写方式。如果使用 P9 寄存器的高 8 位作为 P9H 寄存器，低 8 位作为 P9L 寄存器，这些寄存器支持 8 位或 1 位读写方式。

复位后: 无定义		R/W	地址: FFFFF412H, FFFFF413H					
	15	14	13	12	11	10	9	8
P9 (P9H [Ⓓ])	P915	P914	P913	P912	P911	P910	P99	P98
	7	6	5	4	3	2	1	0
(P9L)	P97	P96	P95	P94	P93	P92	P91	P90
P9n	输出数据的控制 (输出模式) (n = 0 ~ 15)							
0	输出 0。							
1	输出 1。							

注 要以 8 位 或 1 位为单位读写 P9 寄存器的位 8 ~ 15，需要将他们指定为 P9H 寄存器的位 0 ~ 7。

(b) 端口模式寄存器 9 (PM9)

该寄存器是指定输入或输出模式的 16 位寄存器。该寄存器支持 16 位读写方式。如果使用 PM9 寄存器的高 8 位作为 PM9H 寄存器，低 8 位作为 PM9L 寄存器，这些寄存器支持 8 位或 1 位读写方式。

复位后: FFFFH		R/W	地址: FFFFF432H, FFFFF433H					
	15	14	13	12	11	10	9	8
PM9 (PM9H [Ⓓ])	PM915	PM914	PM913	PM912	PM911	PM910	PM99	PM98
	7	6	5	4	3	2	1	0
(PM9L)	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90
PM9n	I/O 模式的控制 (n = 0 ~ 15)							
0	输出模式							
1	输入模式							

注 要以 8 位 或 1 位为单位读写 PM9 寄存器的位 8 ~ 15，需要将他们指定为 PM9H 寄存器的位 0 ~ 7。

(c) 端口模式控制寄存器 9 (PMC9)

该寄存器是指定端口模式或控制模式的 16 位寄存器。该寄存器支持 16 位读写方式。

如果使用 PMC9 寄存器的高 8 位作为 PMC9H 寄存器，低 8 位作为 PMC9L 寄存器，这些寄存器支持 8 位或 1 位读写方式。

注意事项 当 PFC9.PFC9n 位和 PFCE9.PFCE9n 位是默认值 (0) 时，如果由 PMC9 寄存器指定控制模式，则输出不确定。

因此，首先设置 PFC9.PFC9n 和 PFCE9.PFCE9n 位为 1，然后将 PMC9n 位设置为 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 ^注)	PMC915	PMC914	PMC913	0	0	0	PMC99	PMC98
	7	6	5	4	3	2	1	0
(PMC9L)	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90
	PMC915	P915 引脚操作模式的指定						
	0	I/O 端口						
	1	INTP6 输入						
	PMC914	P914 引脚操作模式的指定						
	0	I/O 端口						
	1	INTP5 输入						
	PMC913	P913 引脚操作模式的指定						
	0	I/O 端口						
	1	INTP4/PCL I/O						
	PMC99	P99 引脚操作模式的指定						
	0	I/O 端口						
	1	SCKB1 I/O						
注	要以 8 位或 1 位为单位读写 PMC9 寄存器的位 8 ~ 15，需要将他们指定为 PMC9H 寄存器的位 0 ~ 7。							

PMC98	P98 引脚操作模式的指定
0	I/O 端口
1	SOB1 输出

PMC97	P97 引脚操作模式的指定
0	I/O 端口
1	SIB1/TIP20/TOP20 I/O

PMC96	P96 引脚操作模式的指定
0	I/O 端口
1	TIP21/TOP21 I/O

PMC95	P95 引脚操作模式的指定
0	I/O 端口
1	TIQ10/TOQ10 I/O

PMC94	P94 引脚操作模式的指定
0	I/O 端口
1	TIQ13/TOQ13 I/O

PMC93	P93 引脚操作模式的指定
0	I/O 端口
1	TIQ12/TOQ12 I/O

PMC92	P92 引脚操作模式的指定
0	I/O 端口
1	TIQ11/TOQ11 I/O

PMC91	P91 引脚操作模式的指定
0	I/O 端口
1	KR7/RXDA1 输入

PMC90	P90 引脚操作模式的指定
0	I/O 端口
1	KR6/TXDA1 I/O

(d) 端口功能控制寄存器 9 (PFC9)

该寄存器是指定控制模式 1, 2, 3 或 4 的 16 位寄存器。该寄存器支持 16 位读写方式。

如果使用 PFC9 寄存器的高 8 位作为 PFC9H 寄存器, 低 8 位作为 PFC9L 寄存器, 这些寄存器支持 8 位或 1 位读写方式。

复位后: 0000H		R/W		地址: FFFFF472H, FFFFF473H				
	15	14	13	12	11	10	9	8
PFC9 (PFC9H [※])	PFC915	PFC914	PFC913	0	0	0	PFC99	PFC98
	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

注 要以 8 位 或 1 位为单位读写 PFC9 寄存器的位 8 ~ 15, 需要将他们指定为 PFC9H 寄存器的位 0 ~ 7。

备注 关于如何指定控制模式, 敬请参阅 4.3.9 (2) (f) P9 引脚控制模式的设置。

(e) 端口功能控制扩展寄存器 9 (PFCE9)

该寄存器是指定控制模式 1, 2, 3 或 4 的 16 位寄存器。该寄存器支持 16 位读写方式。

如果使用 PFC9 寄存器的高 8 位作为 PFC9H 寄存器, 低 8 位作为 PFC9L 寄存器, 这些寄存器支持 8 位或 1 位读写方式。

复位后: 0000H		R/W		地址: FFFFF712H, FFFFF713H				
	15	14	13	12	11	10	9	8
PFCE9 (PFCE9H [※])	0	0	PFCE913	0	0	0	0	0
	7	6	5	4	3	2	1	0
(PFCE9L)	PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

注 要以 8 位 或 1 位为单位读写 PFCE9 寄存器的位 8 ~ 15, 需要将他们指定为 PFCE9H 寄存器的位 0 ~ 7。

备注 关于如何指定控制模式, 敬请参阅 4.3.9 (2) (f) P9 引脚控制模式的设置。

(f) P9 引脚控制模式的设置

注意事项 当 PFC9.PFC9n 和 PFCE9.PFCE9n 位是默认值 (0) 时, 如果由 PMC9 寄存器指定控制模式, 则输出不确定。

因此, 首先设置 PFC9.PFC9n 和 PFCE9.PFCE9n 位, 然后将 PMC9n 位设置为 1 以设置控制模式。

PFC915	P915 引脚控制模式的指定	
0	禁止设置	
1	INTP6 输入	

PFC914	P914 引脚控制模式的指定	
0	禁止设置	
1	INTP5 输入	

PFCE913	PFC913	P913 引脚控制模式的指定
0	0	禁止设置
0	1	INTP4 输入
1	0	PCL 输出
1	1	禁止设置

PFC99	P99 引脚控制模式的指定	
0	禁止设置	
1	SCKB1 I/O	

PFC98	P98 引脚控制模式的指定	
0	禁止设置	
1	SOB1 输出	

PFCE97	PFC97	P97 引脚控制模式的指定
0	0	禁止设置
0	1	SIB1 输入
1	0	TIP20 输入
1	1	TOP20 输出

PFCE96	PFC96	P96 引脚控制模式的指定
0	0	禁止设置
0	1	禁止设置
1	0	TIP21 输入
1	1	TOP21 输出

PFCE95	PFC95	P95 引脚控制模式的指定
0	0	禁止设置
0	1	TIQ10 输入
1	0	TOQ10 输出
1	1	禁止设置

PFCE94	PFC94	P94 引脚控制模式的指定
0	0	禁止设置
0	1	TIQ13 输入
1	0	TOQ13 输出
1	1	禁止设置

PFCE93	PFC93	P93 引脚控制模式的指定
0	0	禁止设置
0	1	TIQ12 输入
1	0	TOQ12 输出
1	1	禁止设置

PFCE92	PFC92	P92 引脚控制模式的指定
0	0	禁止设置
0	1	TIQ11 输入
1	0	TOQ11 输出
1	1	禁止设置

PFCE91	PFC91	P91 引脚控制模式的指定
0	0	禁止设置
0	1	KR7 输入
1	0	KR7/RXDA1 输入 ^注
1	1	禁止设置

PFCE90	PFC90	P90 引脚控制模式的指定
0	0	禁止设置
0	1	KR6 输入
1	0	TXDA1 输出
1	1	禁止设置

注 KR7 和 RXDA1 为一组复用功能引脚。

当用作 RXDA1 引脚时，需要禁止 KR7 引脚的按键返回检测。（将 KRM 寄存器的 KRM7 位清零。）
同时，当用作 KR7 引脚时，推荐将 PFC91 位置 1 并且将 PFCE91 位清零。

(g) 上拉电阻选项寄存器 9 (PU9)

该寄存器是指定片上上拉电阻连接的 16 位寄存器。该寄存器支持 16 位读写方式。

如果使用 PU9 寄存器的高 8 位作为 PU9H 寄存器，低 8 位作为 PU9L 寄存器，这些寄存器支持 8 位或 1 位读写方式。

复位后: 0000H	R/W	地址: FFFFC52H, FFFFC53H						
	15	14	13	12	11	10	9	8
PU9 (PU9H [※])	PU915	PU914	PU913	PU912	PU911	PU910	PU99	PU98
	7	6	5	4	3	2	1	0
(PU9L)	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90
	PU9n	片上上拉电阻连接的控制 (n = 0 ~ 15)						
	0	不连接						
	1	连接						

注 要以 8 位 或 1 位为单位读写 PU9 寄存器的位 8 ~ 15，需要将他们指定为 PU9H 寄存器的位 0 ~ 7。

4.3.10 端口 CM

端口 CM 是一个可以以位为单位控制 I/O 设置的 4 位端口（PCM0 ~ PCM3）。

(1) 端口 CM 的功能

- 可以以位为单位指定端口的输入/输出数据。
由端口寄存器 CM (PCM) 指定
- 可以以位为单位指定端口的输入/输出模式。
由端口模式寄存器 CM (PMCM) 指定
- 可以以位为单位指定端口模式或控制模式（复用功能）。
由端口模式控制寄存器 CM (PMCCM) 指定

端口 CM 包含以下复用功能引脚。

表 4-10. 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	模块类型
PCM0	61	—	I/O	—	B-1
PCM1	62	CLKOUT			D-2
PCM2	63	—			B-1
PCM3	64	—			B-1

(2) 寄存器**(a) 端口寄存器 CM (PCM)**

端口寄存器 CM (PCM) 是控制读取引脚电平和写入输出电平的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: 无定义		R/W		地址: FFFFF00CH				
	7	6	5	4	3	2	1	0
PCM	0	0	0	0	PCM3	PCM2	PCM1	PCM0
	PCMN	输出数据的控制 (输出模式) (n = 0 ~ 3)						
	0	输出 0。						
	1	输出 1。						

(b) 端口模式寄存器 CM (PMCM)

该寄存器是指定输入或输出模式的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

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 模式的控制 (n = 0 ~ 3)						
	0	输出模式						
	1	输入模式						

(c) 端口模式控制寄存器 CM (PMCCM)

该寄存器是指定端口模式或控制模式的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: 00H		R/W		地址: FFFFF04CH				
	7	6	5	4	3	2	1	0
PMCCM	0	0	0	0	0	0	PMCCM1	0
	PMCCM1	PCM1 引脚操作模式的规格						
	0	I/O 端口						
	1	CLKOUT 输出						

4.3.11 端口 CS

端口 CS 是一个可以以位为单位控制 I/O 设置的 2 位端口（PCS0, PCS1）。

(1) 端口 CS 的功能

- 可以以位为单位指定端口的输入/输出数据。
由端口寄存器 CS (PCS) 指定
- 可以以位为单位指定端口的输入/输出模式。
由端口模式寄存器 CS (PMCS) 指定

端口 CS 包含以下复用功能引脚。

表 4-11. 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	模块类型
PCS0	59	-	I/O	-	B-1
PCS1	60	-			B-1

(2) 寄存器**(a) 端口寄存器 CS (PCS)**

端口寄存器 CS (PCS) 是控制读取引脚电平和写入输出电平的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: 无定义	R/W	地址: FFFFF008H							
		7	6	5	4	3	2	1	0
PCS		0	0	0	0	0	0	PCS1	PCS0
	PCSn	输出数据的控制 (输出模式) (n = 0, 1)							
	0	输出 0。							
	1	输出 1。							

(b) 端口模式寄存器 CS (PMCS)

该寄存器是指定输入或输出模式的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: FFH	R/W	地址: FFFFF028H							
		7	6	5	4	3	2	1	0
PMCS		1	1	1	1	1	1	PMCS1	PMCS0
	PMCSn	I/O 模式的控制 (n = 0, 1)							
	0	输出模式							
	1	输入模式							

4.3.12 端口 CT

端口 CT 是一个可以以位为单位控制 I/O 设置的 4 位端口（PCT0, PCT1, PCT4, PCT6）。

(1) 端口 CT 的功能

- 可以以位为单位指定端口的输入/输出数据。
由端口寄存器 CT (PCT) 指定
- 可以以位为单位指定端口的输入/输出模式。
由端口模式寄存器 CT (PMCT) 指定

端口 CT 包含以下复用功能引脚。

表 4-12. 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	模块类型
PCT0	65	-	I/O	-	B-1
PCT1	66	-			B-1
PCT4	67	-			B-1
PCT6	68	-			B-1

(2) 寄存器**(a) 端口寄存器 CT (PCT)**

端口寄存器 CT (PCT) 是控制读取引脚电平和写入输出电平的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

复位后: 无定义	R/W	地址: FFFFF00AH						
	7	6	5	4	3	2	1	0
PCT	0	PCT6	0	PCT4	0	0	PCT1	PCT0
PCTn	输出数据的控制 (输出模式) (n = 0, 1, 4, 6)							
0	输出 0。							
1	输出 1。							

(b) 端口模式寄存器 CT (PMCT)

该寄存器是指定输入或输出模式的 8 位寄存器。该寄存器支持 8 位或 1 位的读写方式。

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

4.3.13 端口 DL

端口 DL 是一个可以以位为单位控制 I/O 设置的 14 位端口 (PDL0 ~ PDL13)。

(1) 端口 DL 的功能

- 可以以位为单位指定端口的输入/输出数据。
由端口寄存器 DL (PDL) 指定
- 可以以位为单位指定端口的输入/输出模式。
由端口模式寄存器 DL (PMDL) 指定

端口 DL 包含以下复用功能引脚。

表 4-13. 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	模块类型
PDL0	71	-	I/O	-	B-1
PDL1	72	-			B-1
PDL2	73	-			B-1
PDL3	74	-			B-1
PDL4	75	-			B-1
PDL5	76	FLMD1 ^注			B-1
PDL6	77	-			B-1
PDL7	78	-			B-1
PDL8	79	-			B-1
PDL9	80	-			B-1
PDL10	81	-			B-1
PDL11	82	-			B-1
PDL12	83	-			B-1
PDL13	84	-			B-1

注 由于需要在 flash 编程模式下使用 FLMD1 引脚，因此不需要操作端口控制寄存器。关于更多细节，敬请参阅第二十三章 Flash 存储器。

(2) 寄存器**(a) 端口寄存器 DL (PDL)**

端口寄存器 DL (PDL) 是控制读取引脚电平和写入输出电平的 16 位寄存器。该寄存器支持 16 位读写方式。

如果使用 PDL 寄存器的高 8 位作为 PDL H 寄存器，低 8 位作为 PDL L 寄存器，这些寄存器支持 8 位或 1 位读写方式。

复位后: 无定义		R/W	地址: FFFFF004H, FFFFF005H					
	15	14	13	12	11	10	9	8
PDL (PDLH 8)	0	0	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8
	7	6	5	4	3	2	1	0
(PDL L)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0
PDLn	输出数据的控制 (输出模式) (n = 0 ~ 13)							
0	输出 0。							
1	输出 1。							

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

(b) 端口模式寄存器 DL (PMDL)

该寄存器是指定输入或输出模式的 16 位寄存器。该寄存器支持 16 位读写方式。

如果使用 PMDL 寄存器的高 8 位作为 PMDL H 寄存器，低 8 位作为 PMDL L 寄存器，这些寄存器支持 8 位或 1 位读写方式。

复位后: FFFFH		R/W	地址: FFFFF024H, FFFFF025H					
	15	14	13	12	11	10	9	8
PMDL (PMDLH 8)	1	1	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8
	7	6	5	4	3	2	1	0
(PMDL L)	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0
PMDLn	I/O 模式的控制 (n = 0 ~ 13)							
0	输出模式							
1	输入模式							

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

4.3.14 复用为片上调试功能的端口引脚

表 4-14 中所示的引脚复用为片上调试引脚。外部复位后，这些引脚的初始设置为片上调试引脚 ($\overline{\text{DRST}}$, DDI, DDO, DCK 和 DMS)。

表 4-14. 片上调试引脚

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

外部复位后，采取以下措施将这些引脚用作端口引脚，而不是片上调试引脚。

- <1> 将 OCDM 寄存器（特殊寄存器）的 OCDM0 位清零。
- <2> 采取上述措施后，将 $\overline{\text{P05/INTP2/DRST}}$ 引脚固定为低电平。

当不使用片上调试功能，在采取上述措施之前，输入高电平到 $\overline{\text{DRST}}$ 引脚会引起故障（CPU 死锁）。在处理 P05 引脚时要十分小心。

当没有输入高电平到 $\overline{\text{P05/INTP2/DRST}}$ 引脚时（当该引脚固定为低电平），不需要操作 OCDM.OCDM0 位。

因为下拉电阻 (30 k Ω TYP.) 连接到 $\overline{\text{P05/INTP2/DRST}}$ 引脚的缓冲区时，该引脚不必通过外部源固定为低电平。通过将 OCDM0 位清零，可以断开下拉电阻。

关于更多细节，敬请参阅**第二十五章 片上调试功能**。

4.3.15 当使用端口引脚作为复用功能引脚时的寄存器设置

表 4-15. 端口引脚用作复用功能引脚(1/5)

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	其它位 (寄存器)
	名称	I/O					
P00	TIP31	输入	设置无要求	PMC00 = 1	PFC00 = 0	–	
	TOP31	输出	设置无要求	PMC00 = 1	PFC00 = 1	–	
P01	TIP30	输入	设置无要求	PMC01 = 1	PFC01 = 0	–	
	TOP30	输出	设置无要求	PMC01 = 1	PFC01 = 1	–	
P02	NMI	输入	设置无要求	PMC02 = 1	–	–	
P03	INTP0	输入	设置无要求	PMC03 = 1	PFC03 = 0	–	INTx03 (INTx0)
	ADTRG	输出	设置无要求	PMC03 = 1	PFC03 = 1	–	
P04	INTP1	输入	设置无要求	PMC04 = 1	–	–	INTx04 (INTx0)
P05 ^注	INTP2	输入	设置无要求	PMC05 = 1	–	–	INTx05 (INTx0)
	DRST	输入	设置无要求	设置无要求	–	–	OCDM0 (OCDM) = 1
P06	INTP3	输入	设置无要求	PMC06 = 1	–	–	INTx06 (INTx0)
P10	INTP9	输入	设置无要求	PMC10 = 1	–	–	INTx10 (INTx1)
P11	INTP10	输入	设置无要求	PMC11 = 1	–	–	INTx11 (INTx1)

注 外部复位后，P05/INTP2/DRST 引脚的初始设置为片上调试引脚(DRST)。为了不将 P05/INTP2/DRST 引脚用作片上调试引脚，敬请参阅第二十五章 片上调试功能。

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

表 4-15. 端口引脚用作复用功能引脚 (2/5)

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	其它位 (寄存器)
	名称	I/O					
P30	TXDA0	输出	设置无要求	PMC30 = 1	–	–	
P31	RXDA0	输入	设置无要求	PMC31 = 1	–	–	注 1
	INTP7	输入	设置无要求	PMC31 = 1	–	–	注 1, INTx31 (INTx3)
P32	ASCKA0	输入	设置无要求	PMC32 = 1	PFC32 = 0	PFCE32 = 0	
	TOP01	输出	设置无要求	PMC32 = 1	PFC32 = 1	PFCE32 = 0	
	TIP00	输入	设置无要求	PMC32 = 1	PFC32 = 0	PFCE32 = 1	
	TOP00	输出	设置无要求	PMC32 = 1	PFC32 = 1	PFCE32 = 1	
P33	TIP01	输入	设置无要求	PMC33 = 1	PFC33 = 0	–	
	TOP01	输出	设置无要求	PMC33 = 1	PFC33 = 1	–	
P34	TIP10	输入	设置无要求	PMC34 = 1	PFC34 = 0	–	
	TOP10	输出	设置无要求	PMC34 = 1	PFC34 = 1	–	
P35	TIP11	输入	设置无要求	PMC35 = 1	PFC35 = 0	–	
	TOP11	输出	设置无要求	PMC35 = 1	PFC35 = 1	–	
P38	TXDA2	输出	设置无要求	PMC38 = 1	–	–	
P39	RXDA2	输入	设置无要求	PMC39 = 1	–	–	注 2
	INTP8	输入	设置无要求	PMC39 = 1	–	–	注 2, INTx39 (INTx3)
P40	SIB0	输入	设置无要求	PMC40 = 1	–	–	
P41	SOB0	输出	设置无要求	PMC41 = 1	–	–	
P42	SCKB0	I/O	设置无要求	PMC42 = 1	–	–	

- 注**
1. INTP7 引脚用作 RXDA0 引脚的复用引脚，为了作为 RXDA0 引脚使用，使复用功能 INTP7 引脚的边沿检测功能无效（通过将 INTF3.INTF31 位和 INTR3.INTR31 位清零）。为了作为 INTP7 引脚使用，停止 UARTA0 的接收操作（通过将 UA0CTL0.UA0RXE 位清零）。
 2. INTP8 引脚用作 RXDA2 引脚的复用引脚，为了作为 RXDA2 引脚使用，使复用功能 INTP8 引脚的边沿检测功能无效（通过固定 INTF3.INTF39 位和 INTR3.INTR39 位为 0）。为了作为 INTP8 引脚使用，停止 UARTA2 的接收操作（通过将 UA2CTL0.UA2RXE 位清零）。

- 备注**
1. 当使用复用功能时，不需要设置端口寄存器 (Pn)。
 2. INTxn = INTFn, INTRn

表 4-15. 端口引脚用作复用功能引脚 (3/5)

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	其它位 (寄存器)
	名称	I/O					
P50	KR0	输入	设置无要求	PMC50 = 1	PFC50 = 1	PFCE50 = 0	注 1
	TIQ01	输入	设置无要求	PMC50 = 1	PFC50 = 1	PFCE50 = 0	注 1
	TOQ01	输出	设置无要求	PMC50 = 1	PFC50 = 0	PFCE50 = 1	
P51	KR1	输入	设置无要求	PMC51 = 1	PFC51 = 1	PFCE54 = 0	注 1
	TIQ02	输入	设置无要求	PMC51 = 1	PFC51 = 1	PFCE51 = 0	注 1
	TOQ02	输出	设置无要求	PMC51 = 1	PFC51 = 0	PFCE51 = 1	
P52	KR2	输入	设置无要求	PMC52 = 1	PFC52 = 1	PFCE52 = 0	注 1
	TIQ03	输入	设置无要求	PMC52 = 1	PFC52 = 1	PFCE52 = 0	注 1
	TOQ03	输出	设置无要求	PMC52 = 1	PFC52 = 0	PFCE52 = 1	
	DDI ^{注 2}	输入	设置无要求	设置无要求	设置无要求	设置无要求	OCDM0 (OCDM) = 1
P53	KR3	输入	设置无要求	PMC53 = 1	PFC53 = 1	PFCE53 = 0	注 1
	TIQ00	输入	设置无要求	PMC53 = 1	PFC53 = 1	PFCE53 = 0	注 1
	TOQ00	输出	设置无要求	PMC53 = 1	PFC53 = 0	PFCE53 = 1	
	DDO ^{注 2}	输出	设置无要求	设置无要求	设置无要求	设置无要求	OCDM0 (OCDM) = 1
P54	KR4	输入	设置无要求	PMC54 = 1	PFC54 = 1	—	
	DCK ^{注 2}	输出	设置无要求	设置无要求	设置无要求	—	OCDM0 (OCDM) = 1
P55	KR5	输入	设置无要求	PMC55 = 1	PFC55 = 1	—	
	DMS ^{注 2}	输出	设置无要求	设置无要求	设置无要求	—	OCDM0 (OCDM) = 1

注 1. KRn 引脚复用作 TIQ0m 引脚，为了作为 TIQ0m 引脚使用。使复用功能 KRn 引脚的按键返回检测功能无效（通过将 KRM.KRMn 位为 0）。为了作为 KRn 引脚使用，使复用功能 TIQ0m 引脚的边沿检测功能无效(n = 0 ~ 3, m = 0 ~ 3)。

引脚名称	用作 TIQ0m 引脚时	用作 KRn 引脚时
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 KRM3 位的 = 0	TQ0IOC1 寄存器的 TQ0TIG0, TQ0TIG1 位= 0 TQ0IOC2 寄存器的 TQ0EES0, TQ0EES1 位= 0 TQ0IOC2 寄存器的 TQ0ETS0, TQ0ETS1 位= 0

2. DDI, DDO, DCK 和 DMS 引脚是一组片上调试引脚。外部复位后，为了不将这些引脚用作片上调试引脚，敬请参阅第二十五章 片上调试功能。

注意事项 当 PFC5.PFC5n 位和 PFCE5.PFCE5n 位是默认值 (0) 时，如果由 PMC5 寄存器指定控制模式，则输出变成无定义。

因此，首先设置 PFC5.PFC5n 位和 PFCE5.PFCE5n 位，然后将 PMC5n 位设置为 1 以设置控制模式。

备注 1. 当使用复用功能时，不需要设置端口寄存器 (Pn)。

2. INTxn = INTFn, INTRn

表 4-15. 端口引脚用作复用功能引脚 (4/5)

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	其它位 (寄存器)
	名称	I/O					
P70	ANI0	输入	PM70 = 1 ^注	-	-	-	
P71	ANI1	输入	PM71 = 1 ^注	-	-	-	
P72	ANI2	输入	PM72 = 1 ^注	-	-	-	
P73	ANI3	输入	PM73 = 1 ^注	-	-	-	
P74	ANI4	输入	PM74 = 1 ^注	-	-	-	
P75	ANI5	输入	PM75 = 1 ^注	-	-	-	
P76	ANI6	输入	PM76 = 1 ^注	-	-	-	
P77	ANI7	输入	PM77 = 1 ^注	-	-	-	
P78	ANI8	输入	PM78 = 1 ^注	-	-	-	
P79	ANI9	输入	PM79 = 1 ^注	-	-	-	
P710	ANI10	输入	PM710 = 1 ^注	-	-	-	
P711	ANI11	输入	PM711 = 1 ^注	-	-	-	
P712	ANI12	输入	PM712 = 1 ^注	-	-	-	
P713	ANI13	输入	PM713 = 1 ^注	-	-	-	
P714	ANI14	输入	PM714 = 1 ^注	-	-	-	
P715	ANI15	输入	PM715 = 1 ^注	-	-	-	

注 使用 P7n (ANIn)的复用功能将 PM7n 设置为 1。

注意事项 当 PFC6.PFC6n 位(n = 0 ~ 8)是默认值 (0) 时, 如果由 PMC6 寄存器指定控制模式, 则输出变成无定义。

因此, 首先设置 PFC6.PFC6n 位, 然后将 PMC6n 位设置为 1 以设置控制模式。

备注 1. 当使用复用功能时, 不需要设置端口寄存器 (Pn)。

2. INTxn = INTFn, INTRn

表 4-15. 端口引脚用作复用功能引脚 (5/5)

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	其它位 (寄存器)
	名称	I/O					
P90	KR6	输入	设置无要求	PMC90 = 1	PFC90 = 1	PFCE90 = 0	
	TXDA1	输出	设置无要求	PMC90 = 1	PFC90 = 0	PFCE90 = 1	
P91	KR7 ^{注1}	输入	设置无要求	PMC91 = 1	PFC91 = 1	PFCE91 = 0	
					PFC91 = 0	PFCE91 = 1	
P91	RXDA1	输入	设置无要求	PMC91 = 1	PFC91 = 0	PFCE91 = 1	
P92	TIQ11	输入	设置无要求	PMC92 = 1	PFC92 = 1	PFCE92 = 0	
	TOQ11	输出	设置无要求	PMC92 = 1	PFC92 = 0	PFCE92 = 1	
P93	TIQ12	输入	设置无要求	PMC93 = 1	PFC93 = 1	PFCE93 = 0	
	TOQ12	输出	设置无要求	PMC93 = 1	PFC93 = 0	PFCE93 = 1	
P94	TIQ13	输入	设置无要求	PMC94 = 1	PFC94 = 1	PFCE94 = 0	
	TOQ13	输出	设置无要求	PMC94 = 1	PFC94 = 0	PFCE94 = 1	
P95	TIQ10	输入	设置无要求	PMC95 = 1	PFC95 = 1	PFCE95 = 0	
	TOQ10	输出	设置无要求	PMC95 = 1	PFC95 = 0	PFCE95 = 1	
P96	TIP21	输入	设置无要求	PMC96 = 1	PFC96 = 0	PFCE96 = 1	
	TOP21	输出	设置无要求	PMC96 = 1	PFC96 = 1	PFCE96 = 1	
P97	SIB1	输入	设置无要求	PMC97 = 1	PFC97 = 1	PFCE97 = 0	
	TIP20	输入	设置无要求	PMC97 = 1	PFC97 = 0	PFCE97 = 1	
	TOP20	输出	设置无要求	PMC97 = 1	PFC97 = 1	PFCE97 = 1	
P98	SOB1	输出	设置无要求	PMC98 = 1	PFC98 = 1	–	
P99	SCKB1	I/O	设置无要求	PMC99 = 1	PFC99 = 1	–	
P913	INTP4	输入	设置无要求	PMC913 = 1	PFC913 = 1	PFCE913 = 0	INTx913 (INTx9H)
	PCL	输出	设置无要求	PMC913 = 1	PFC913 = 0	PFCE913 = 1	
P914	INTP5	输入	设置无要求	PMC914 = 1	PFC914 = 1	–	INTx914 (INTx9H)
P915	INTP6	输入	设置无要求	PMC915 = 1	PFC915 = 1	–	INTx915 (INTx9H)
PCM1	CLKOUT	输出	设置无要求	PMCCM1 = 1	–	–	
PDL5	FLMD1	输入	设置无要求	设置无要求	–	–	注 2

注 1. KR7 和 RXDA1 为一组复用功能引脚。

当用作 RXDA1 引脚时，需要禁止 KR7 引脚的按键返回检测。（将 KRM.KRM7 位清零。）

同时，当用作 KR7 引脚时，推荐将 PFC91 位置 1 并且将 PFCE91 位清零。

2. 因为在 flash 编程模式下使用 FLMD1 引脚，因此不必通过端口控制寄存器来操作 FLMD1。关于更多细节，敬请参阅第二十三章 Flash 存储器。

注意事项 当 PFC9.PFC9n 位和 PFCE9.PFCE9n 位是默认值 (0) 时，如果由 PMC9 寄存器指定控制模式，则输出变成无定义。

因此，首先设置 PFC9.PFC9n 位和 PFCE9.PFCE9n 位，然后将 PMC9n 位设置为 1 以设置控制模式。

备注 1. 当使用复用功能时，不需要设置端口寄存器 (Pn)。

2. INTxn = INTFn, INTRn

4.4 端口框图

图 4-2. A-1 类型的框图

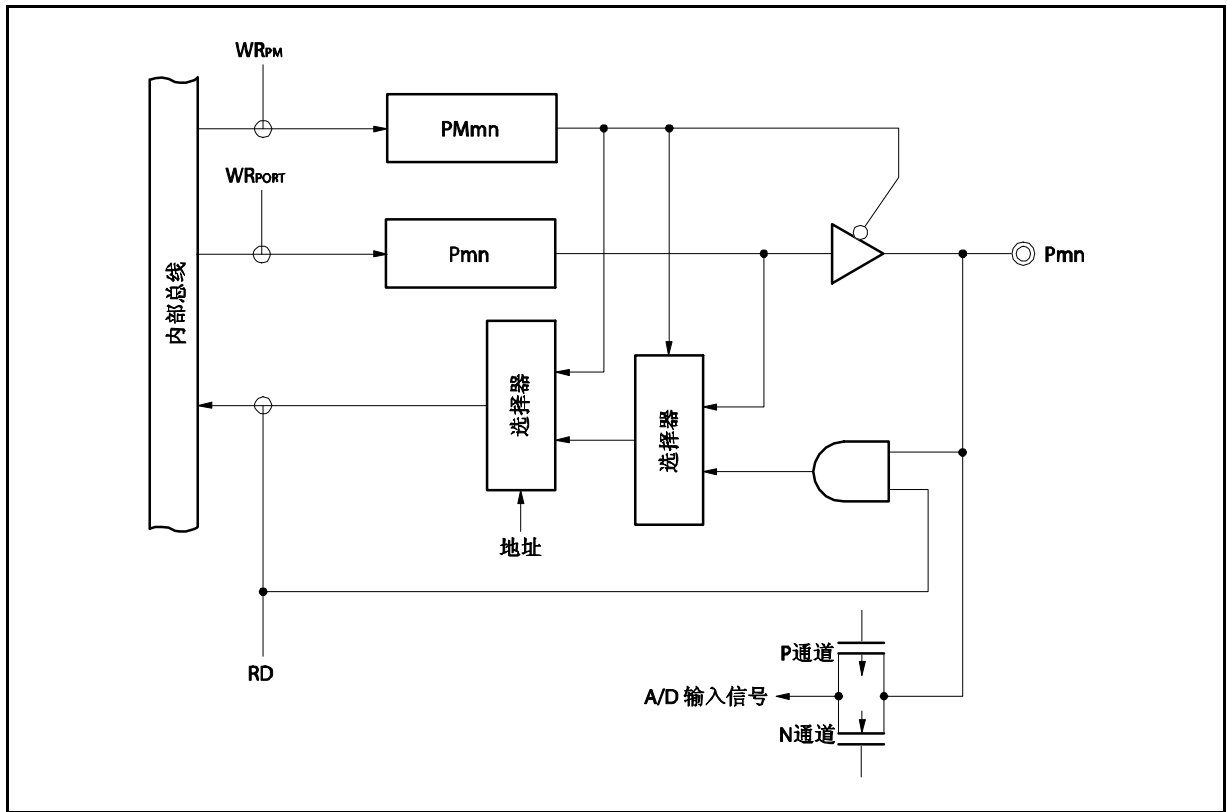


图 4-3. B-1 类型的框图

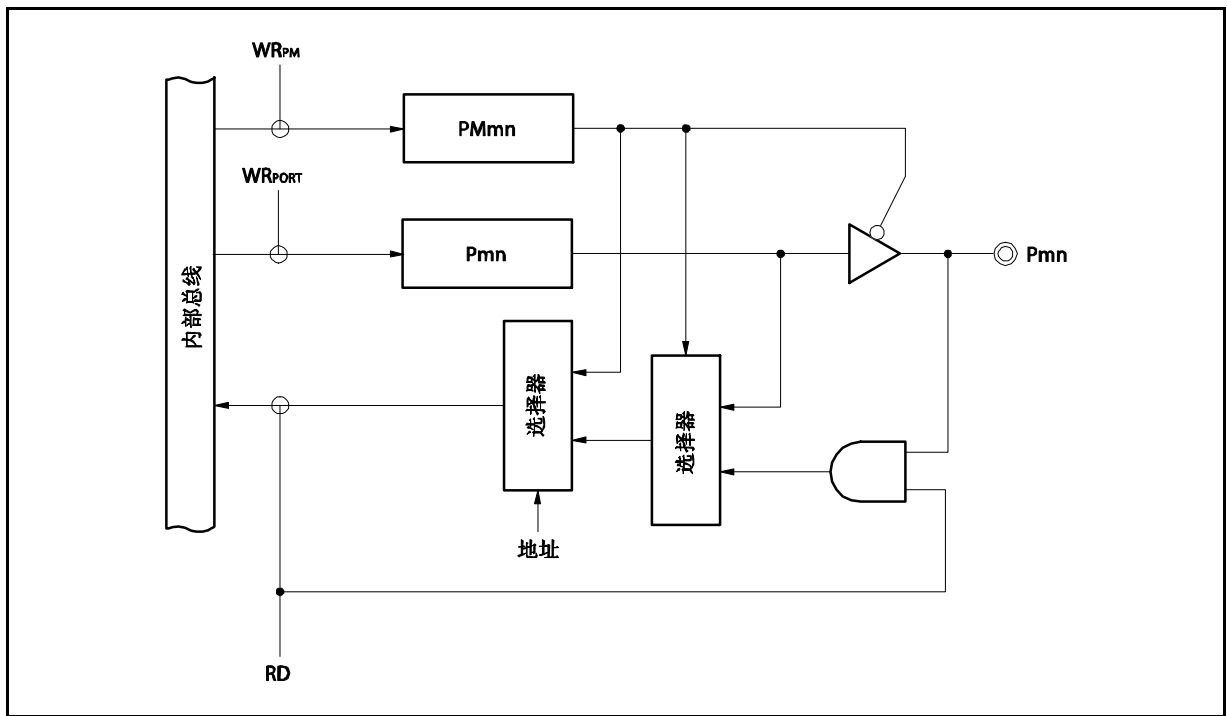


图 4-4. C-1 类型的框图

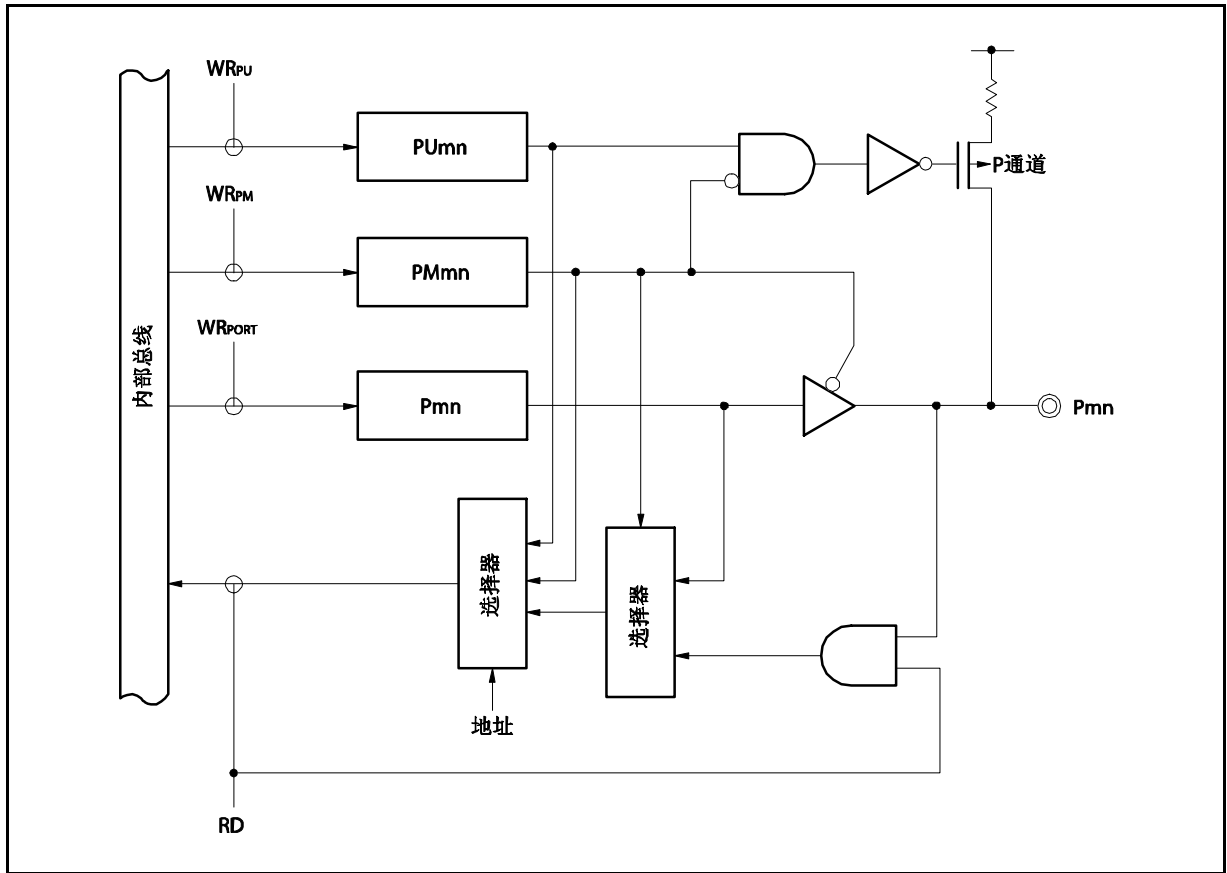


图 4-5. D-2 类型的框图

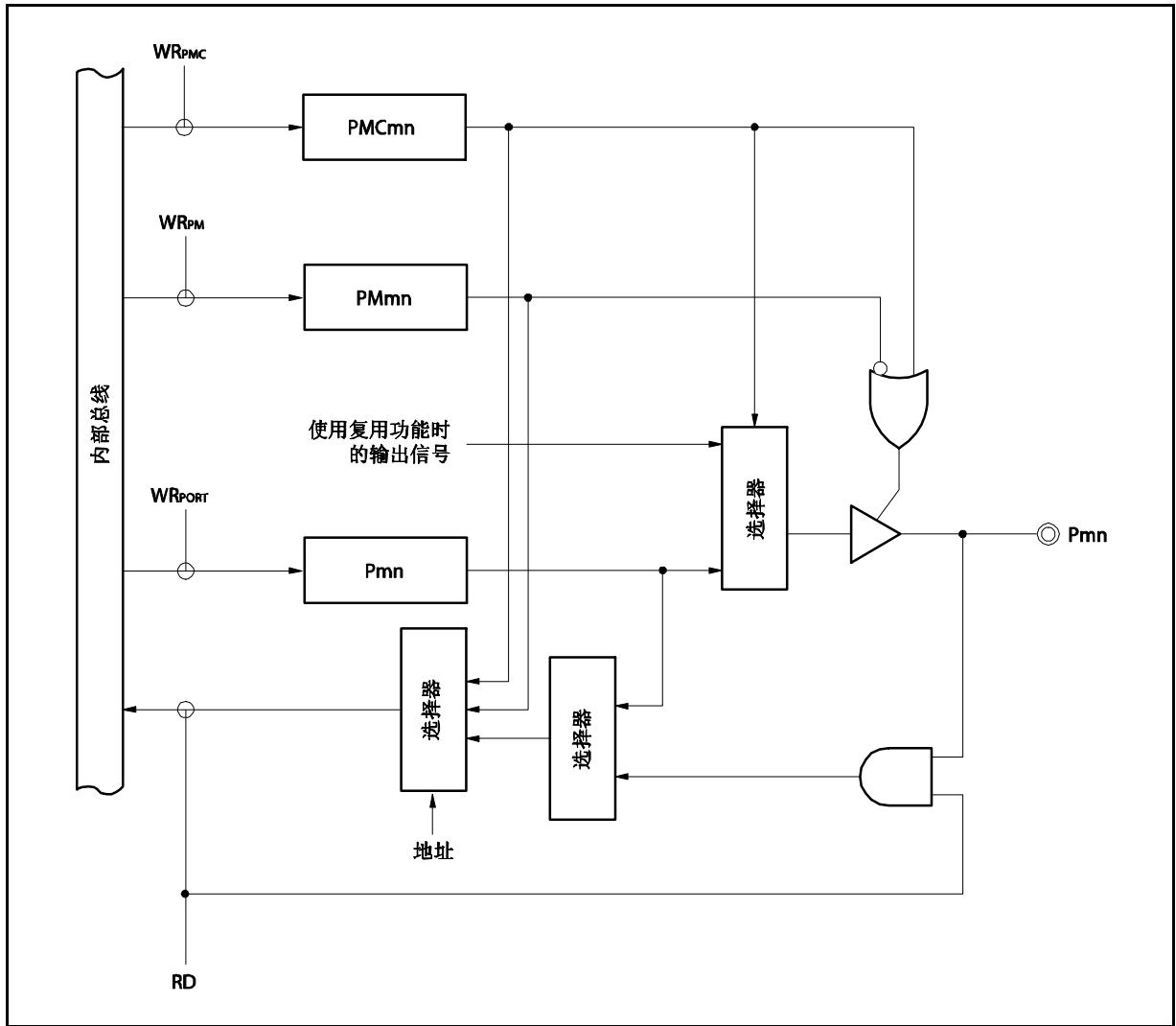


图 4-6. E-1 类型的框图

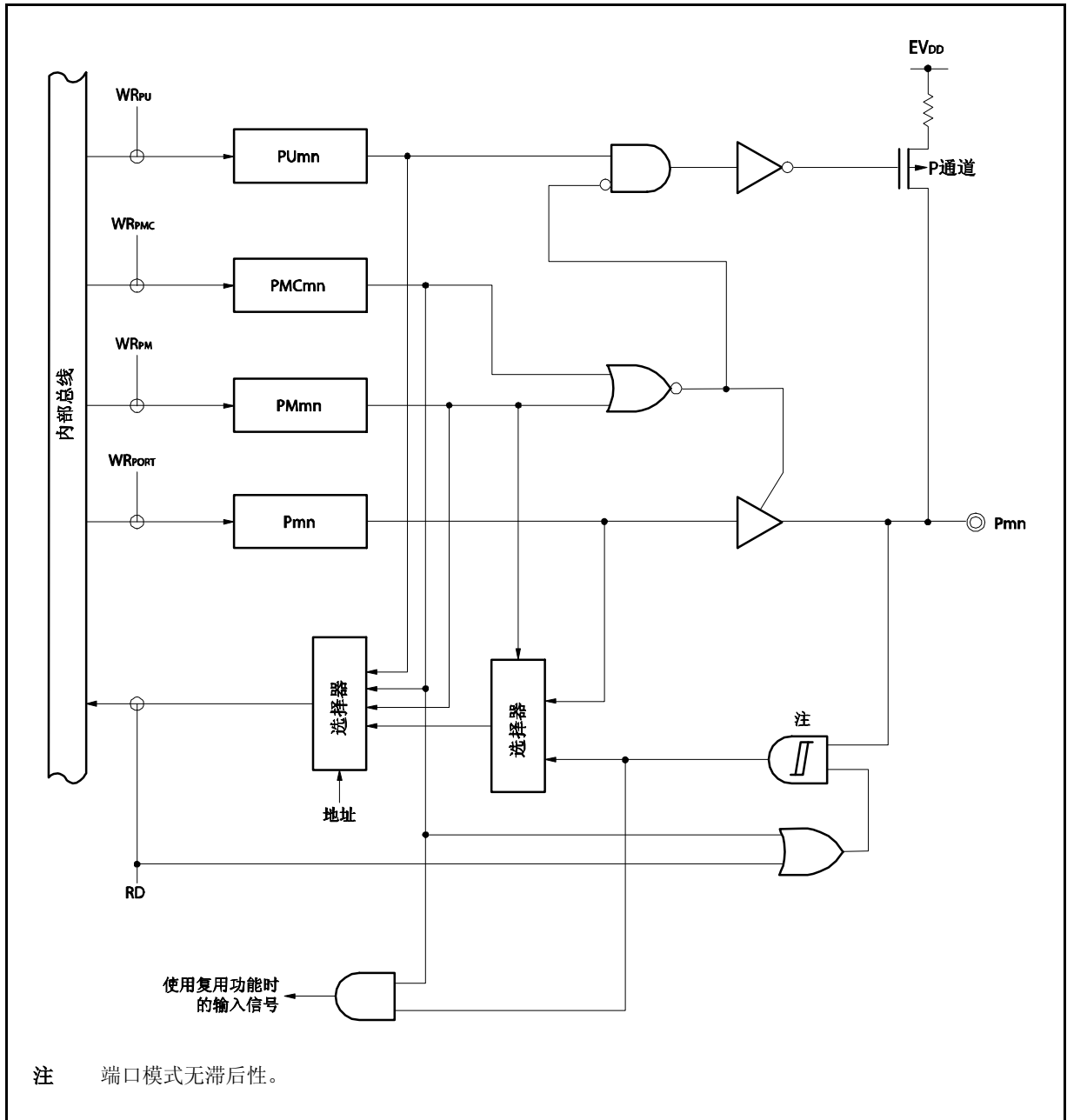


图 4-7. E-2 类型的框图

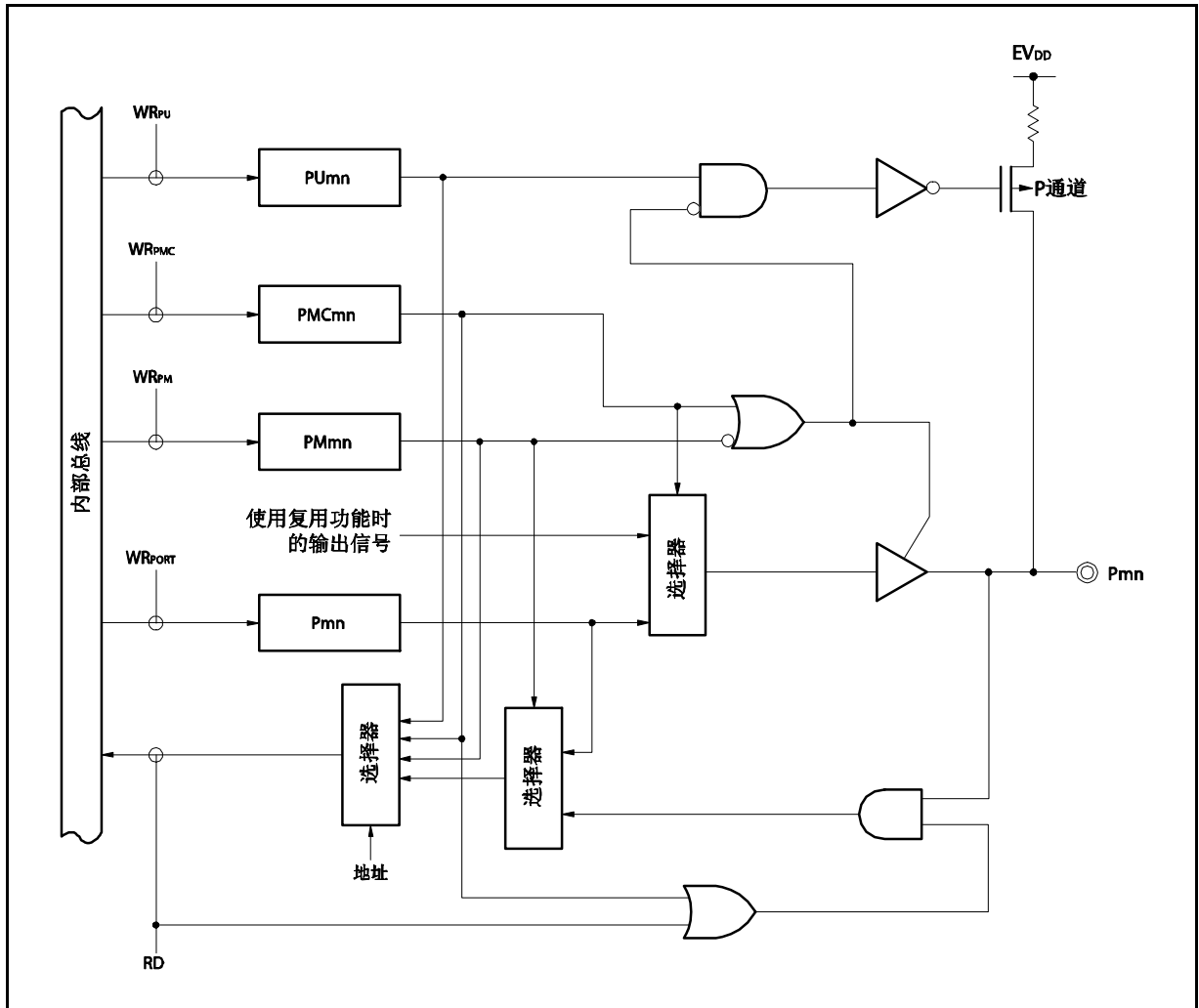


图 4-8. E-3 类型的框图

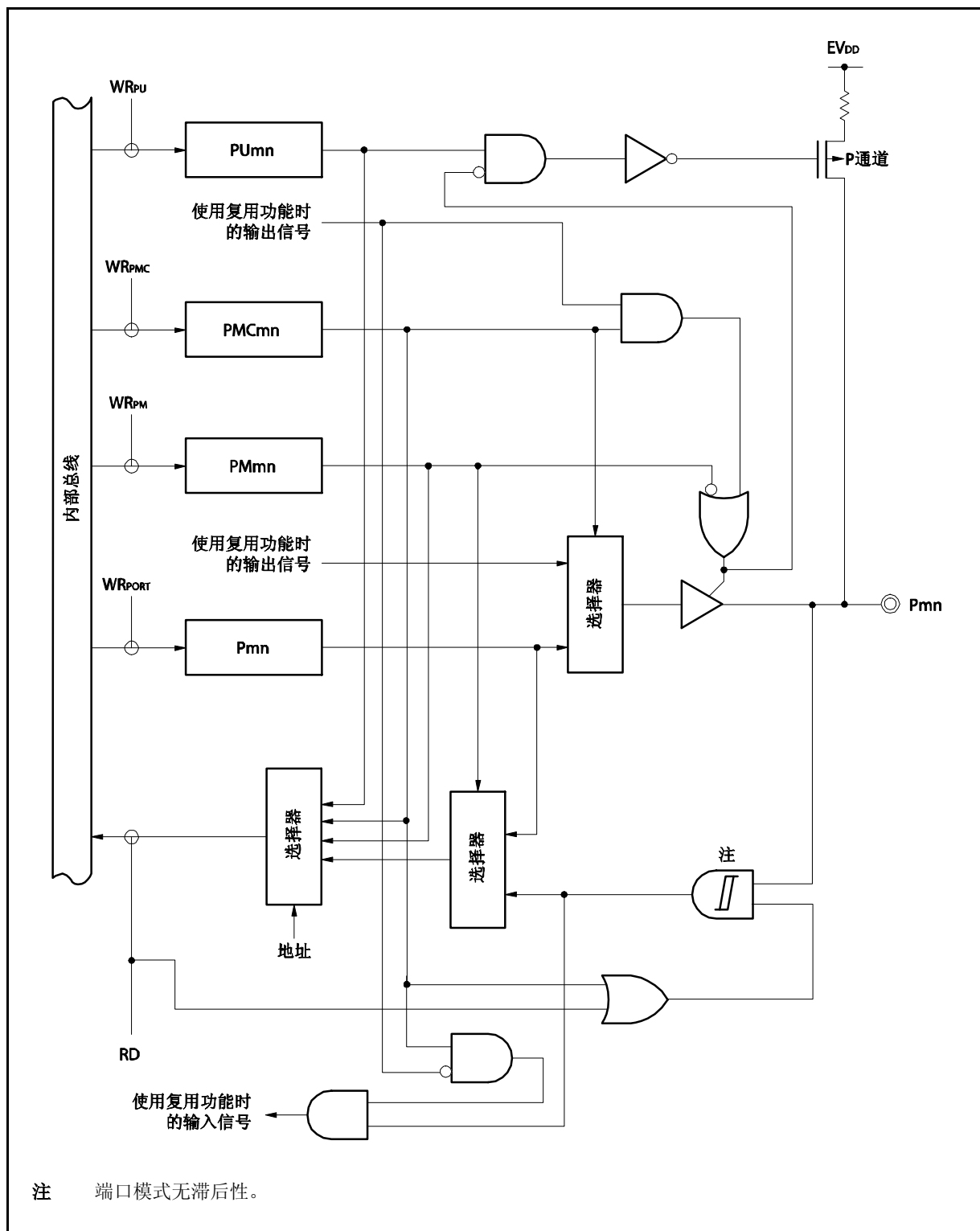


图 4-9. G-1 类型的框图

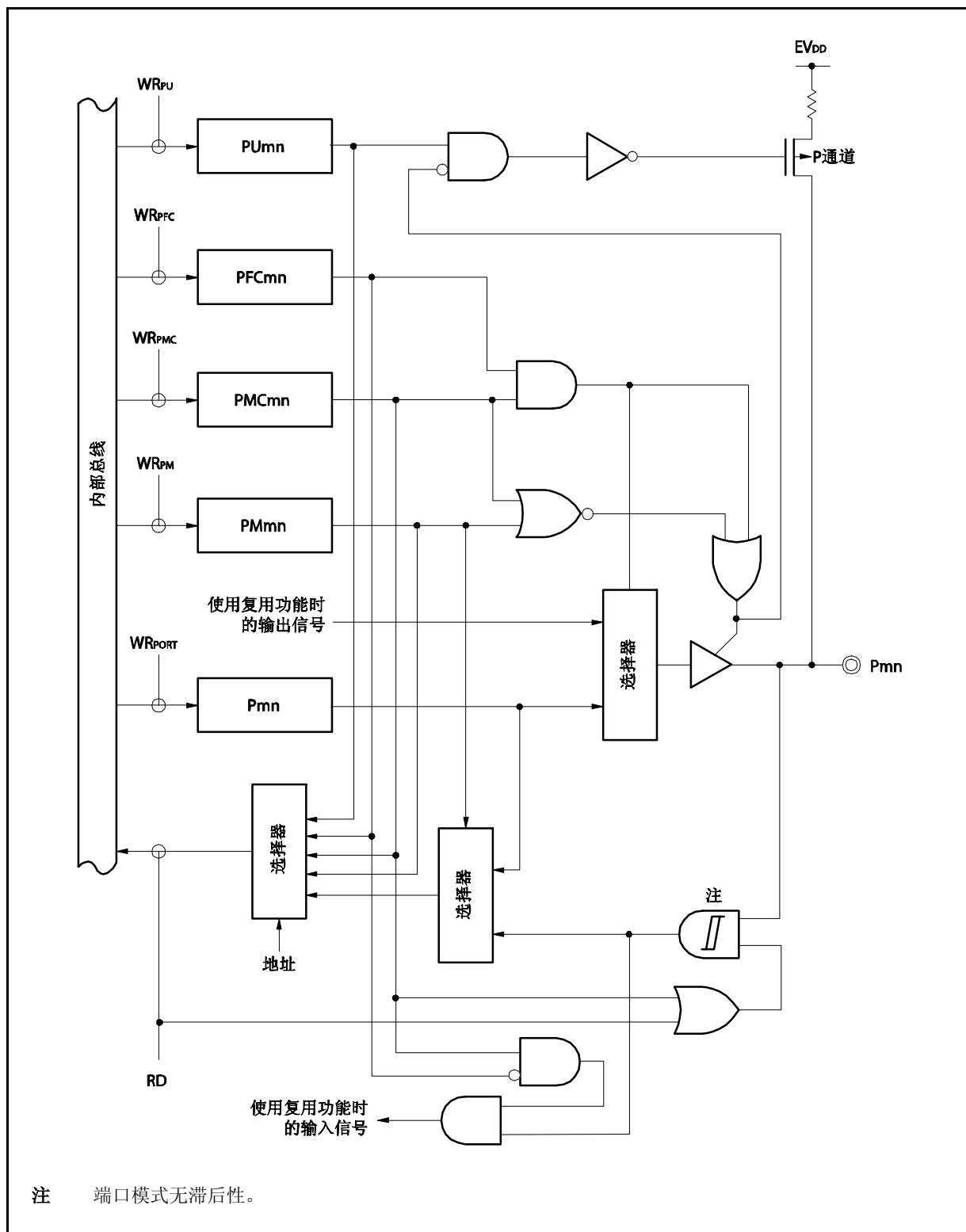


图 4-10. G-2 类型的框图

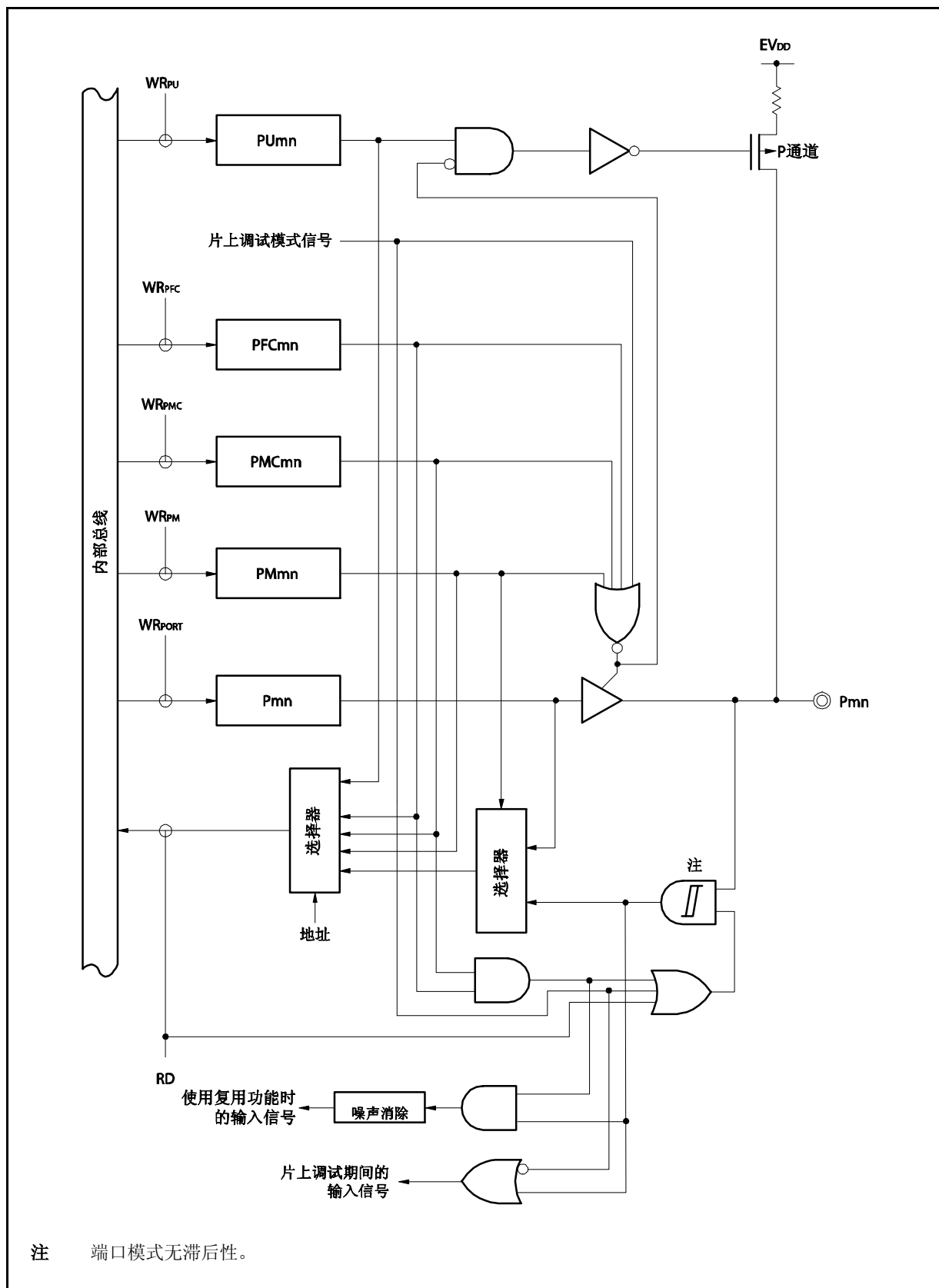


图 4-11. G-3 类型的框图

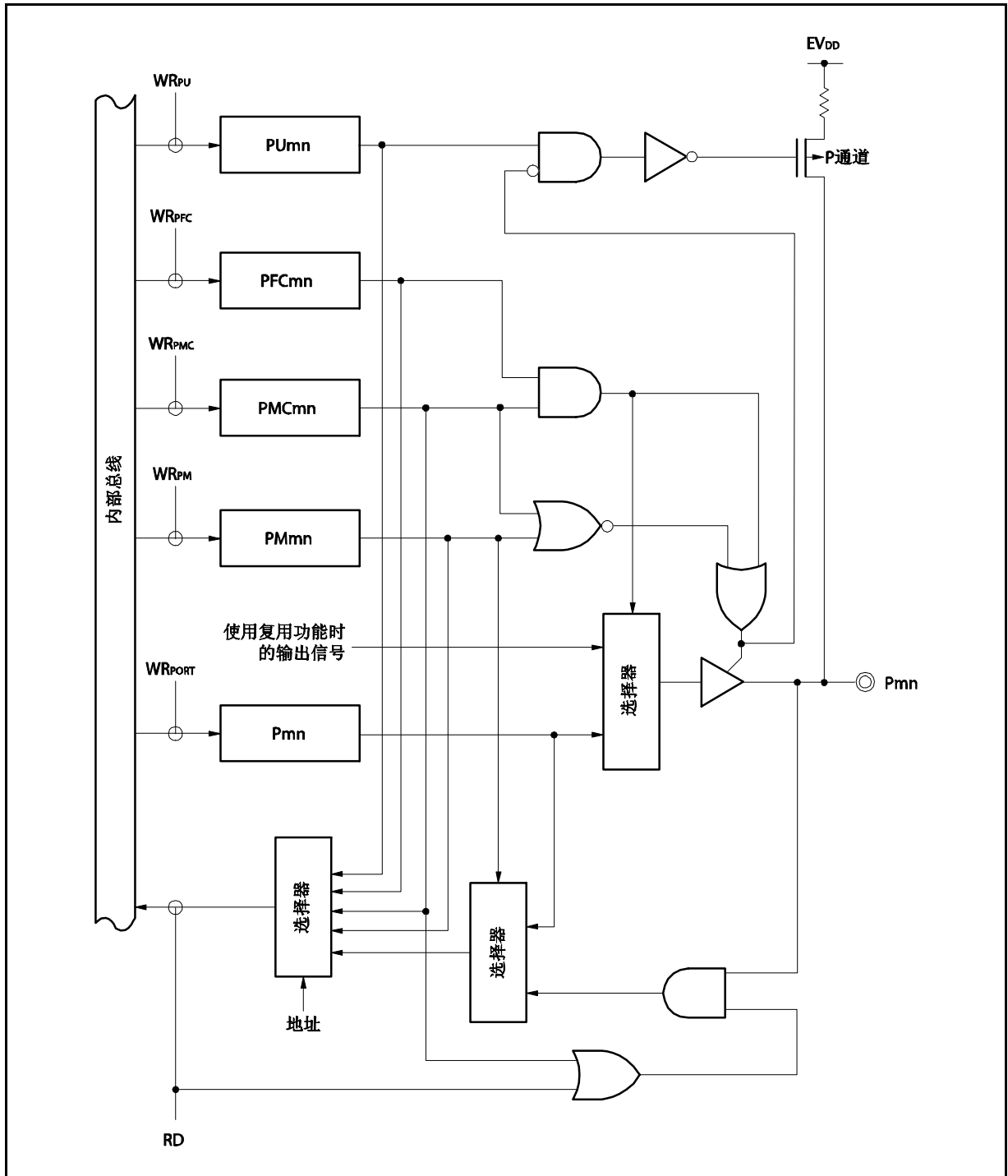


图 4-12. G-5 类型的框图

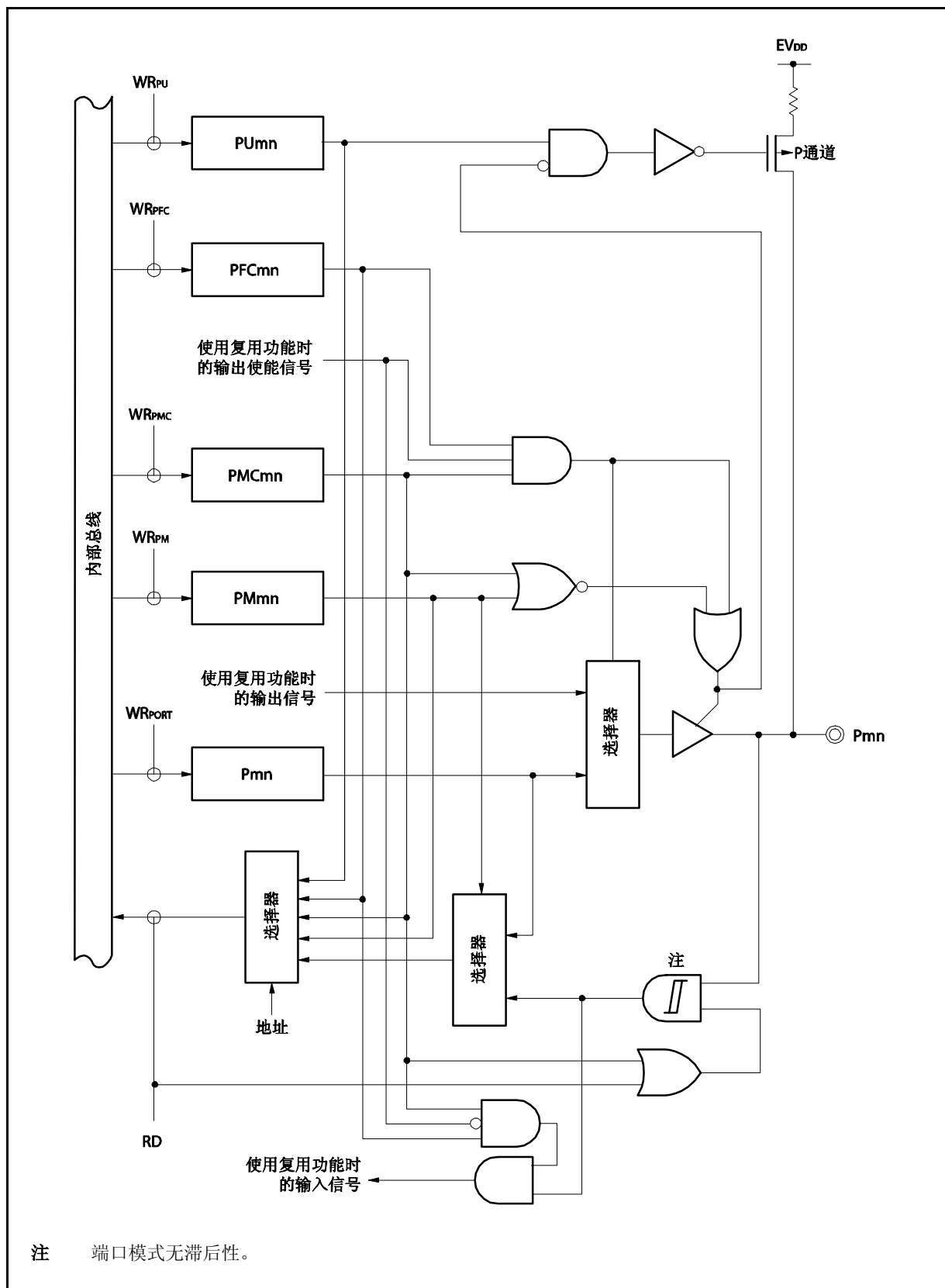


图 4-14. L-2 类型的框图

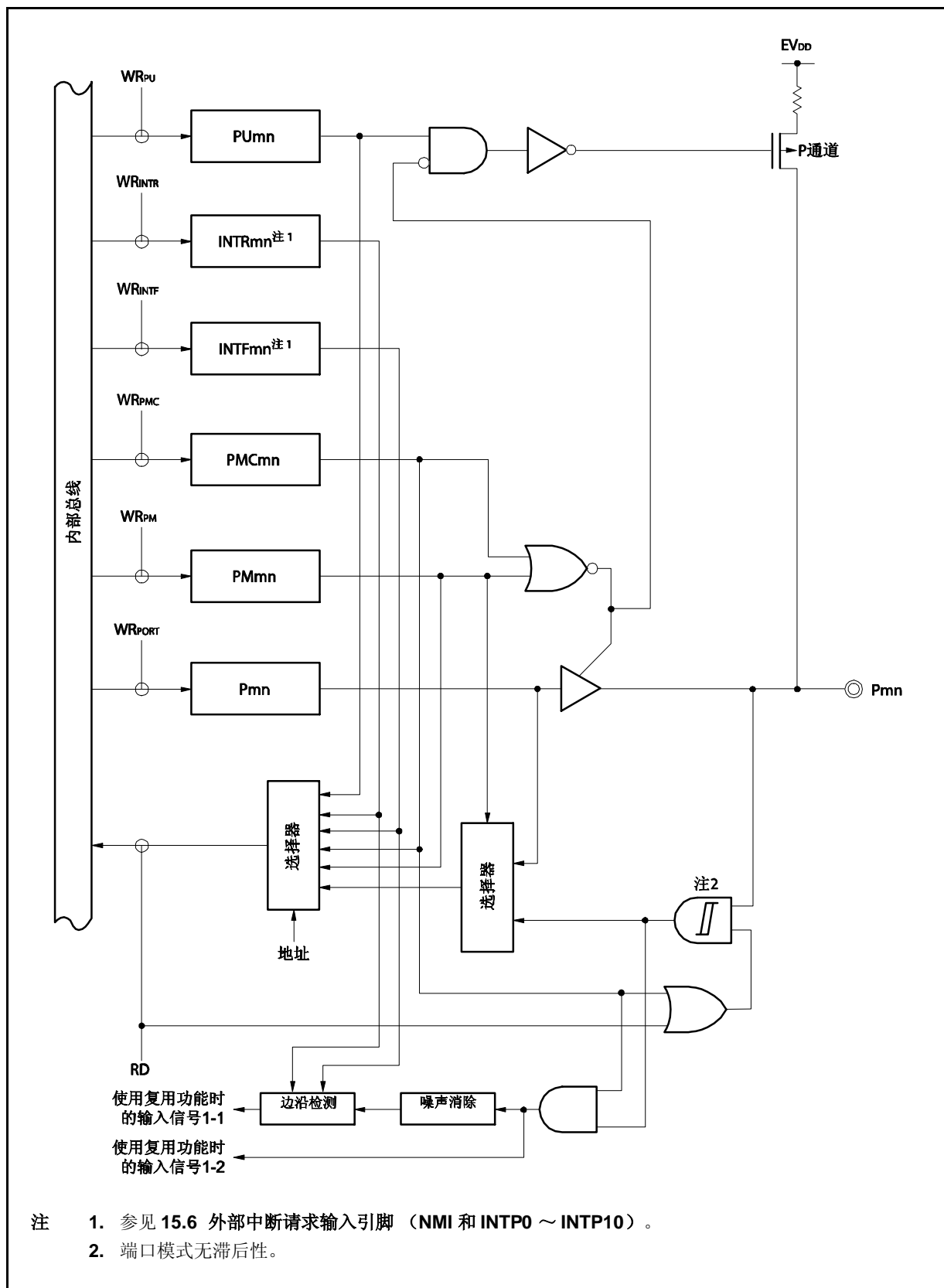


图 4-15. N-1 类型的框图

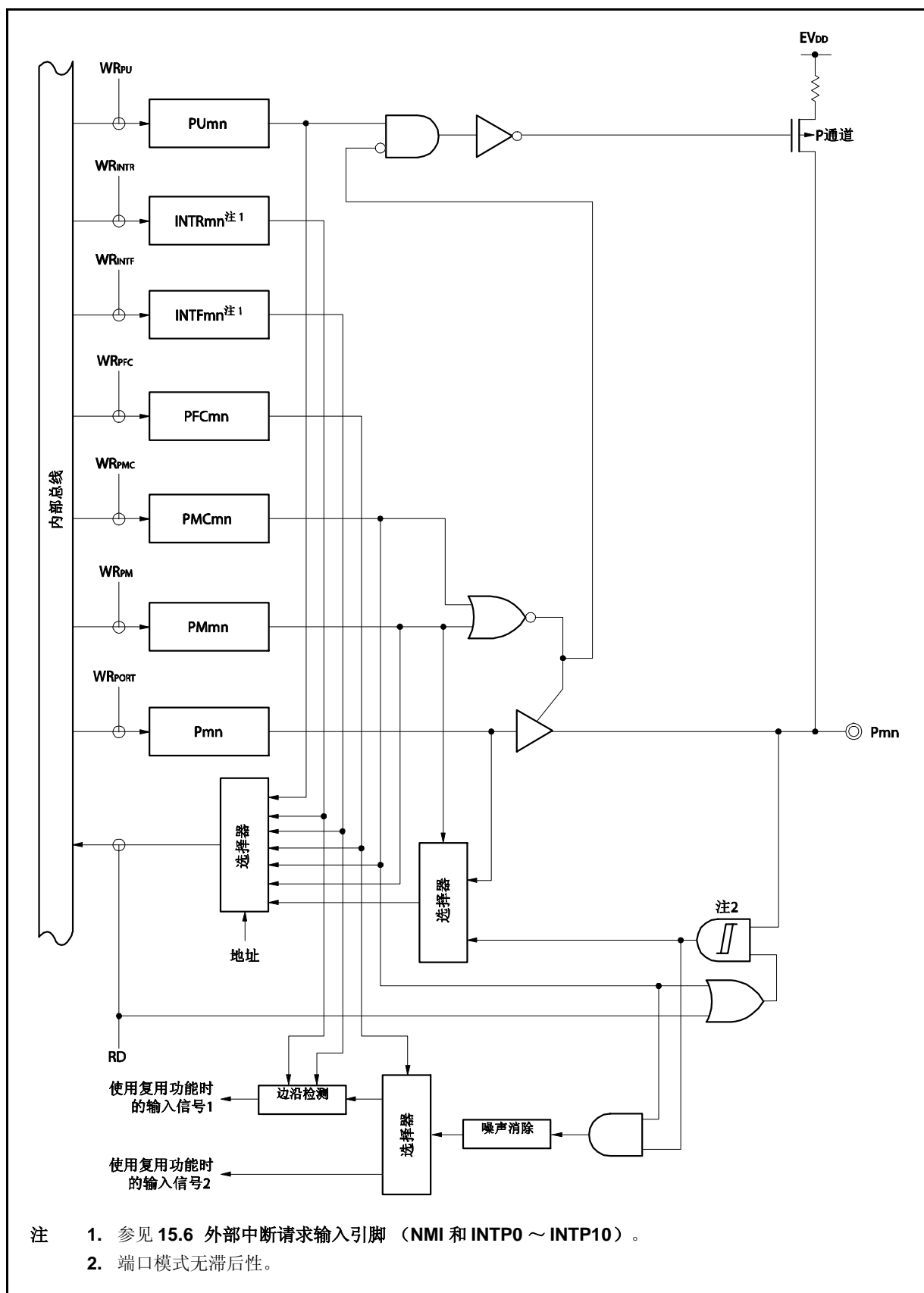


图 4-16. N-2 类型的框图

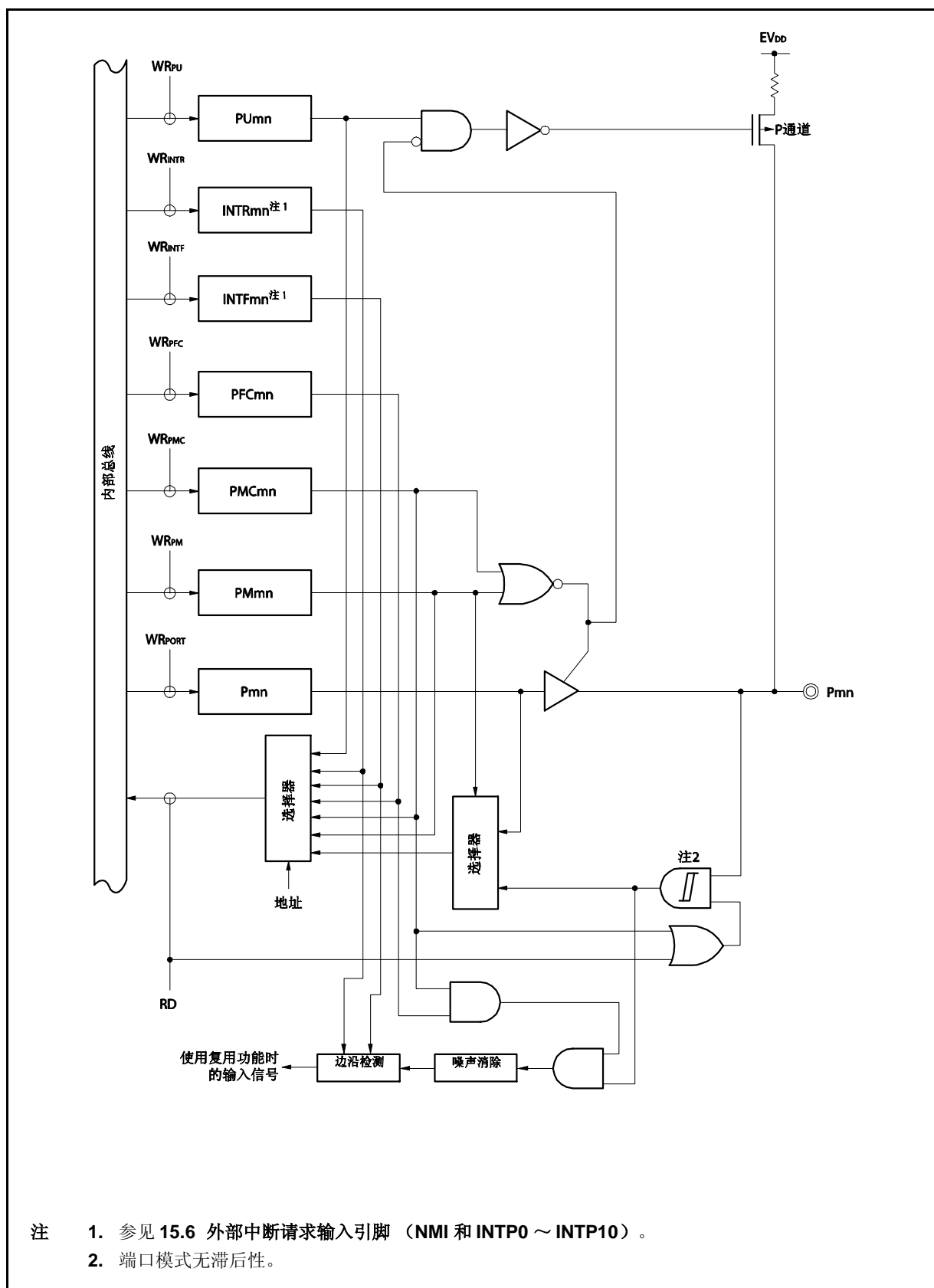
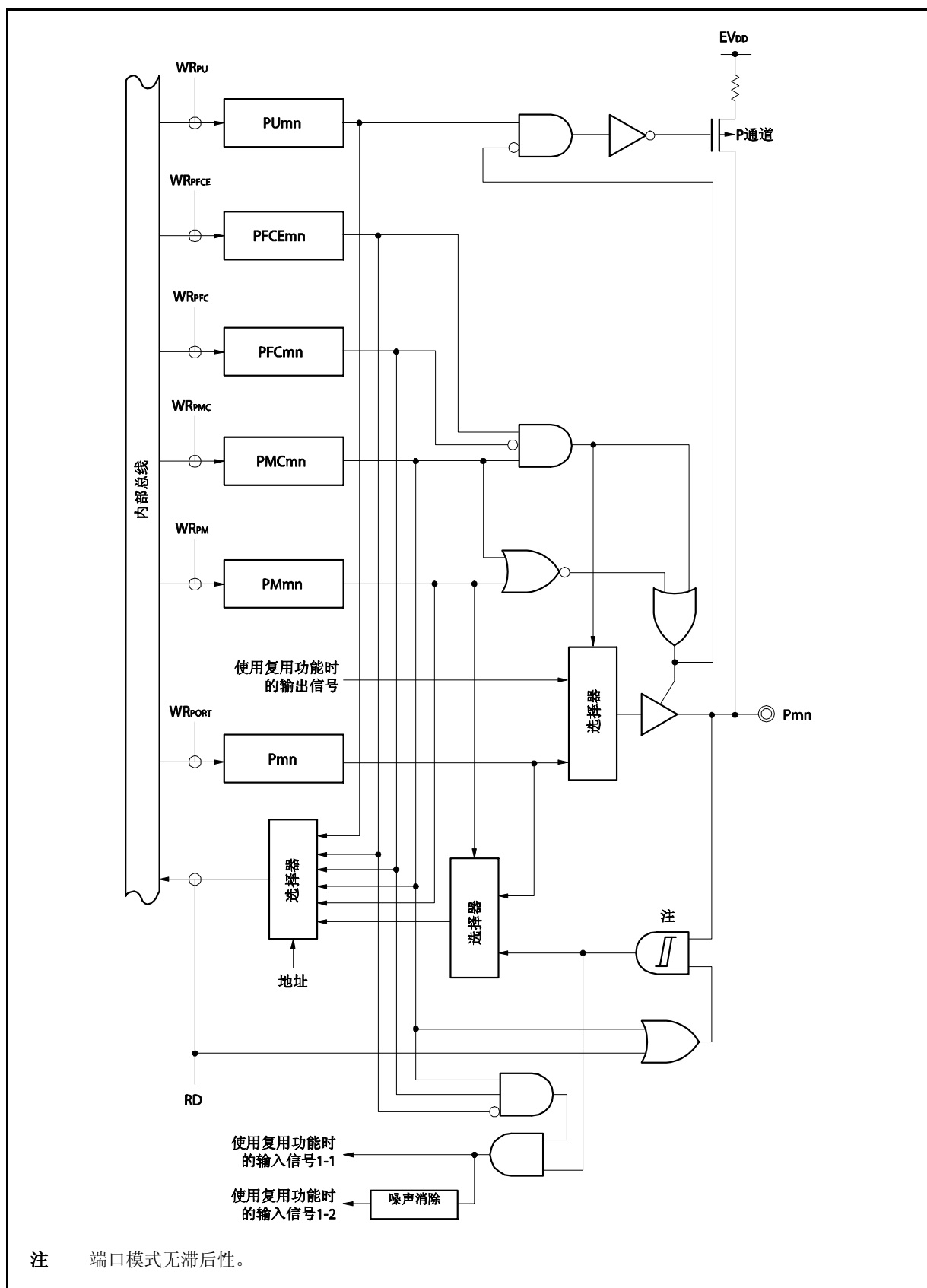


图 4-17. U-4 类型的框图



注 端口模式无滞后性。

图 4-18. U-5 类型的框图

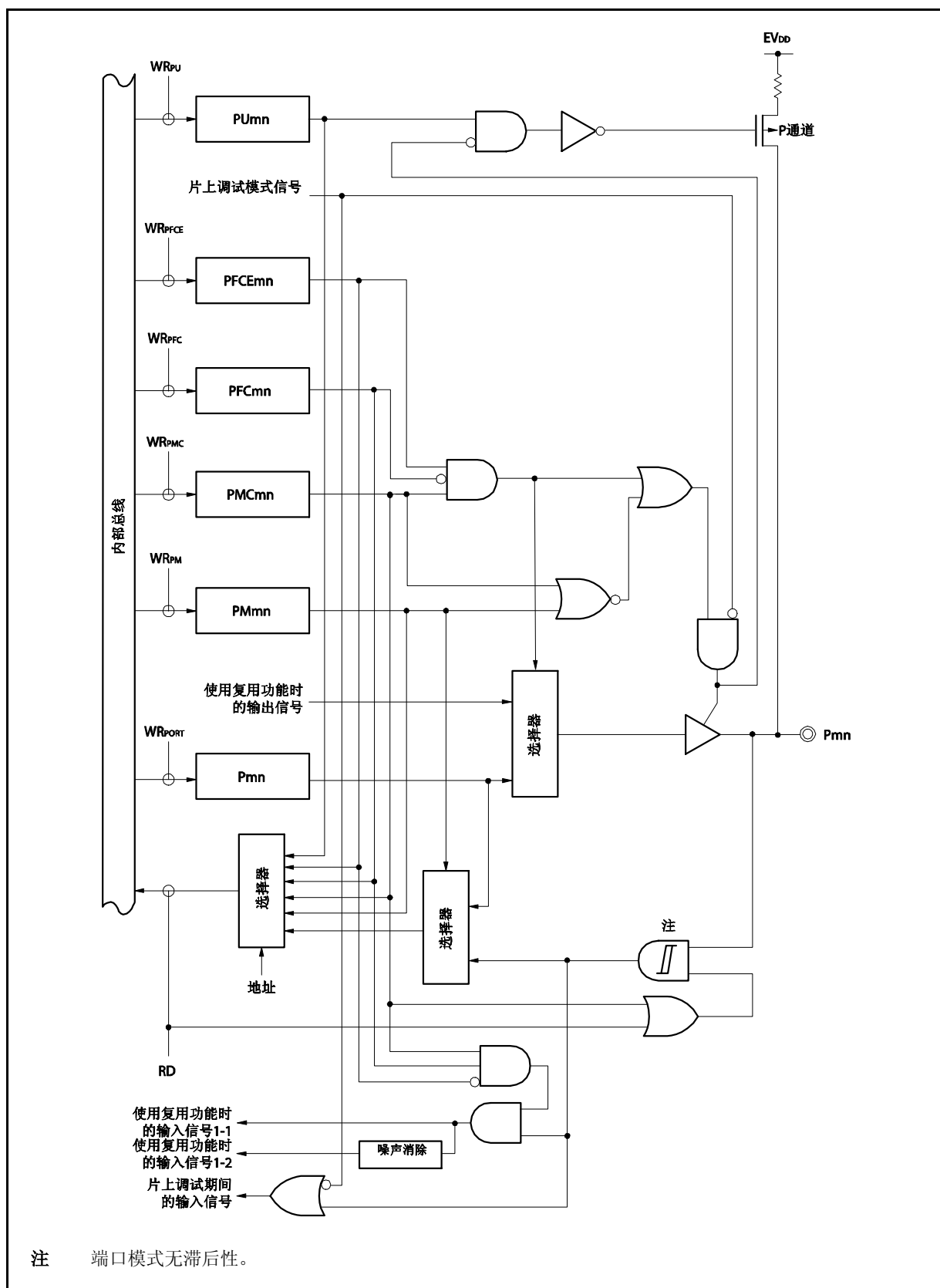


图 4-20. U-7 类型的框图

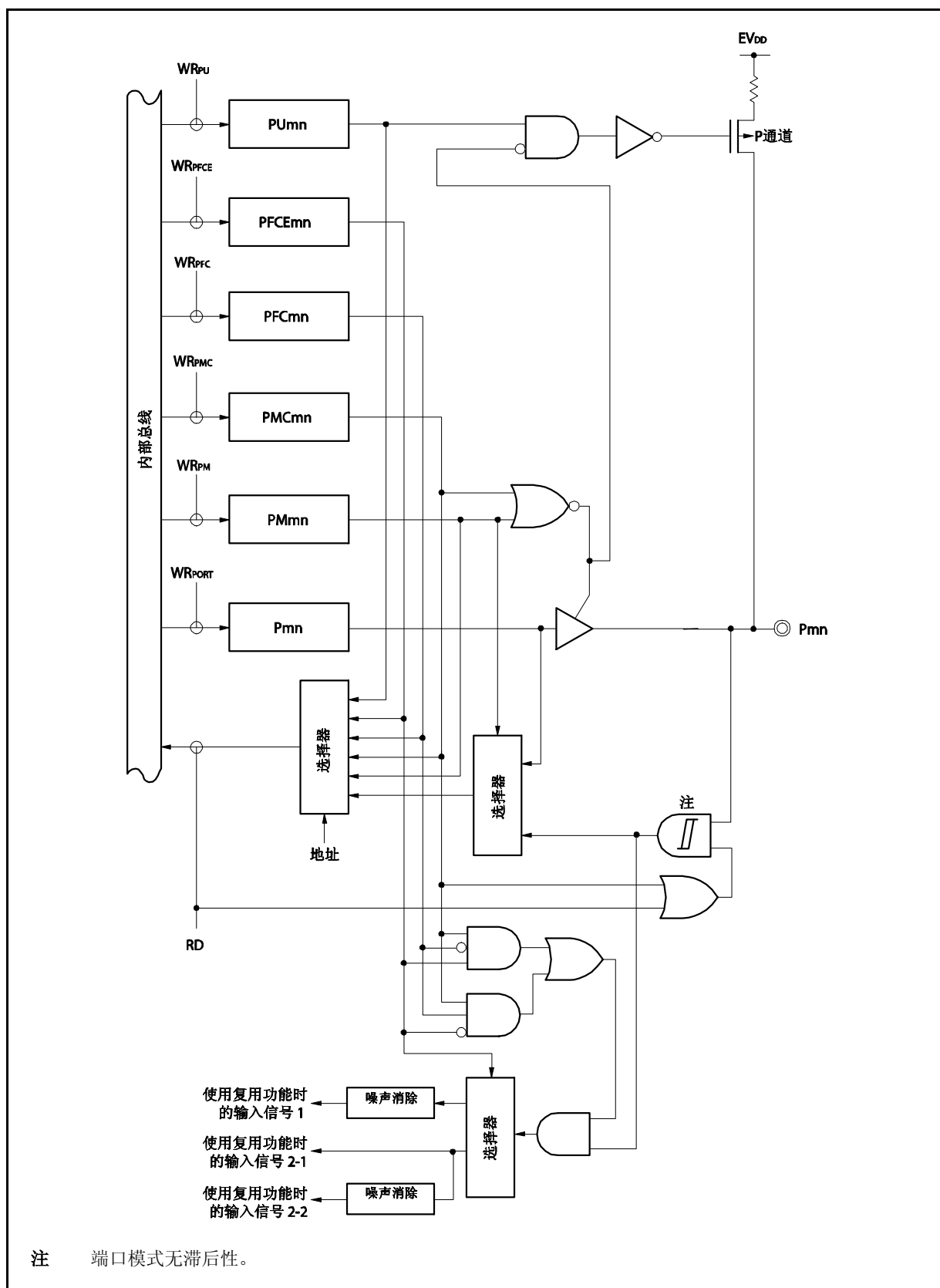


图 4-21. U-8 类型的框图

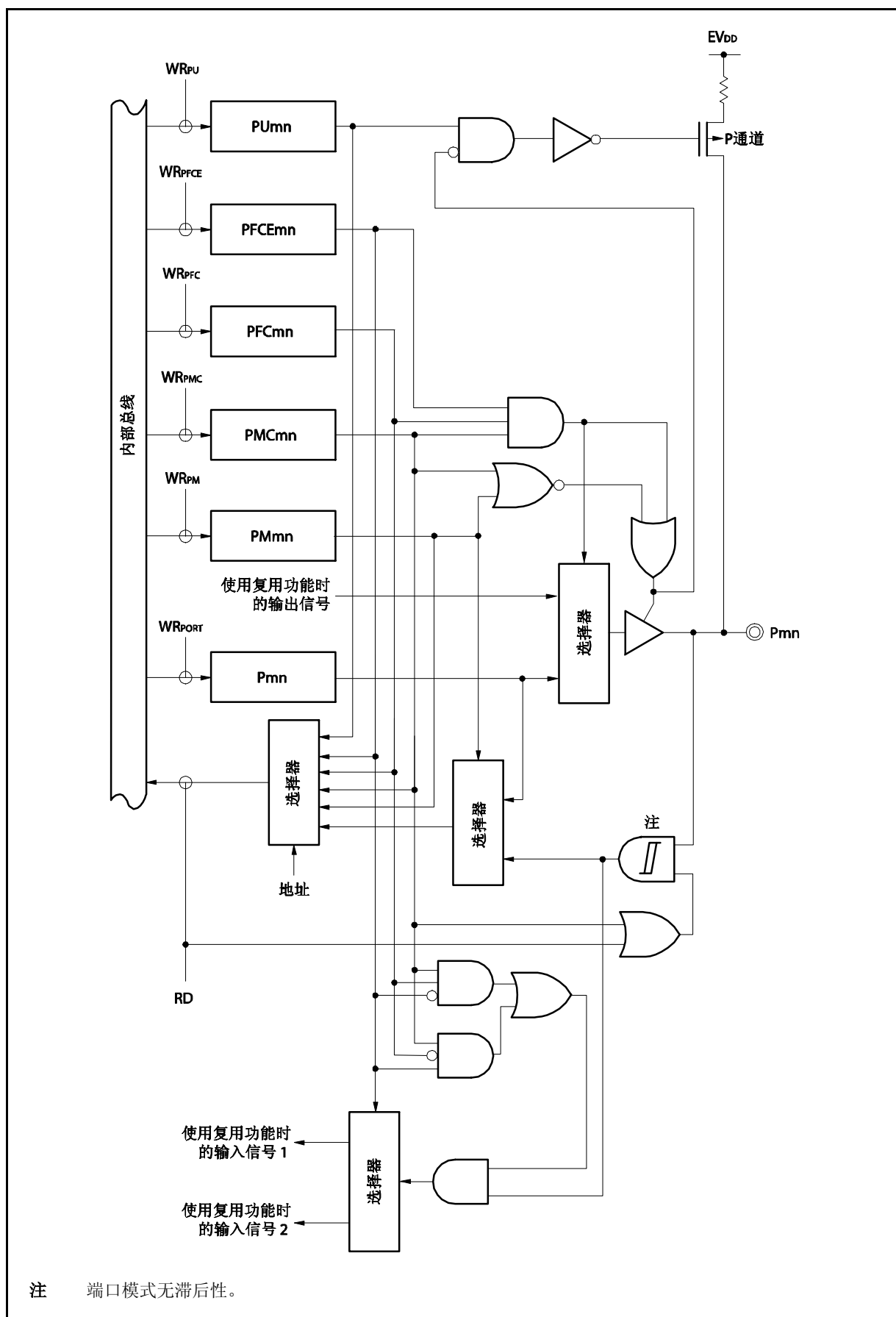
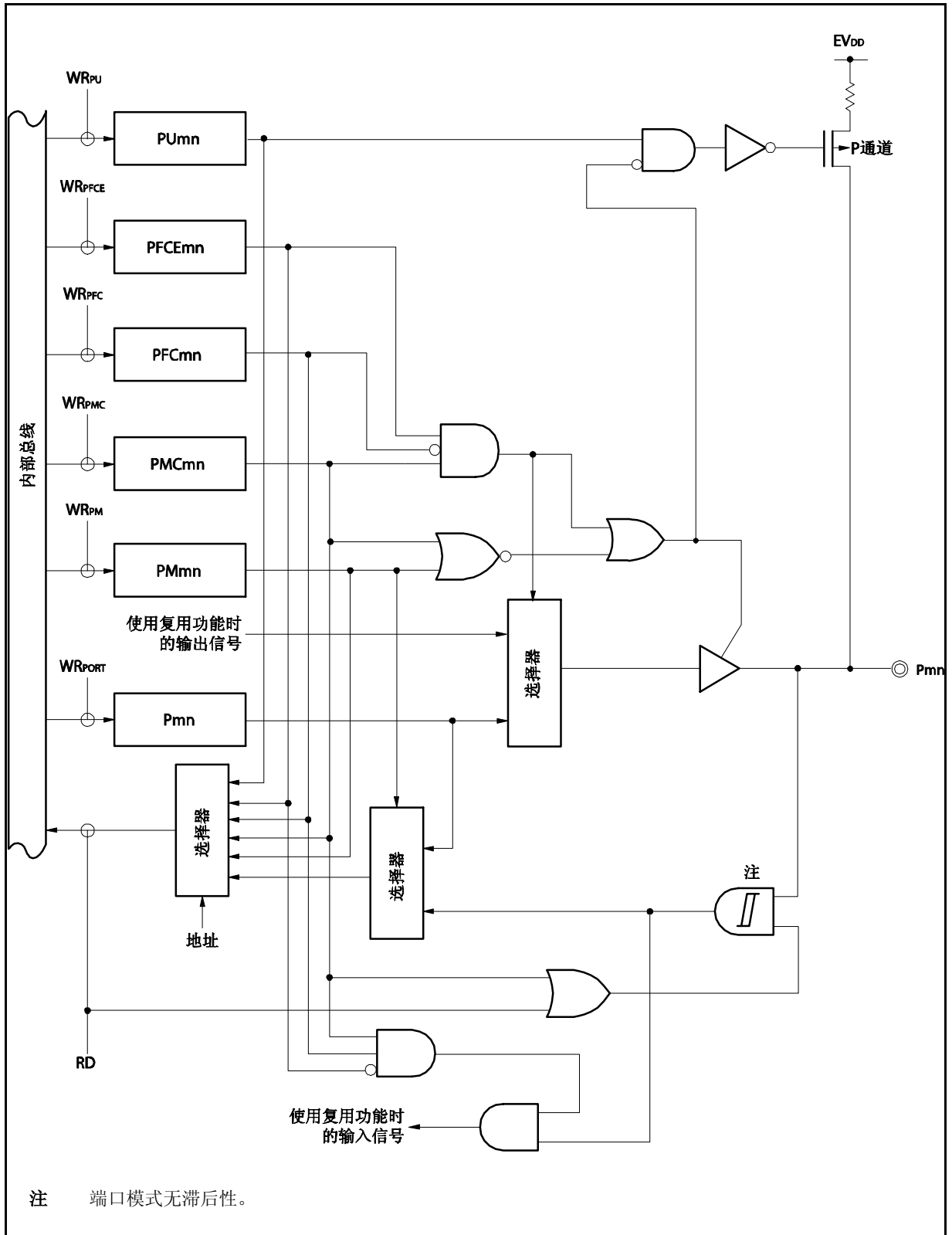
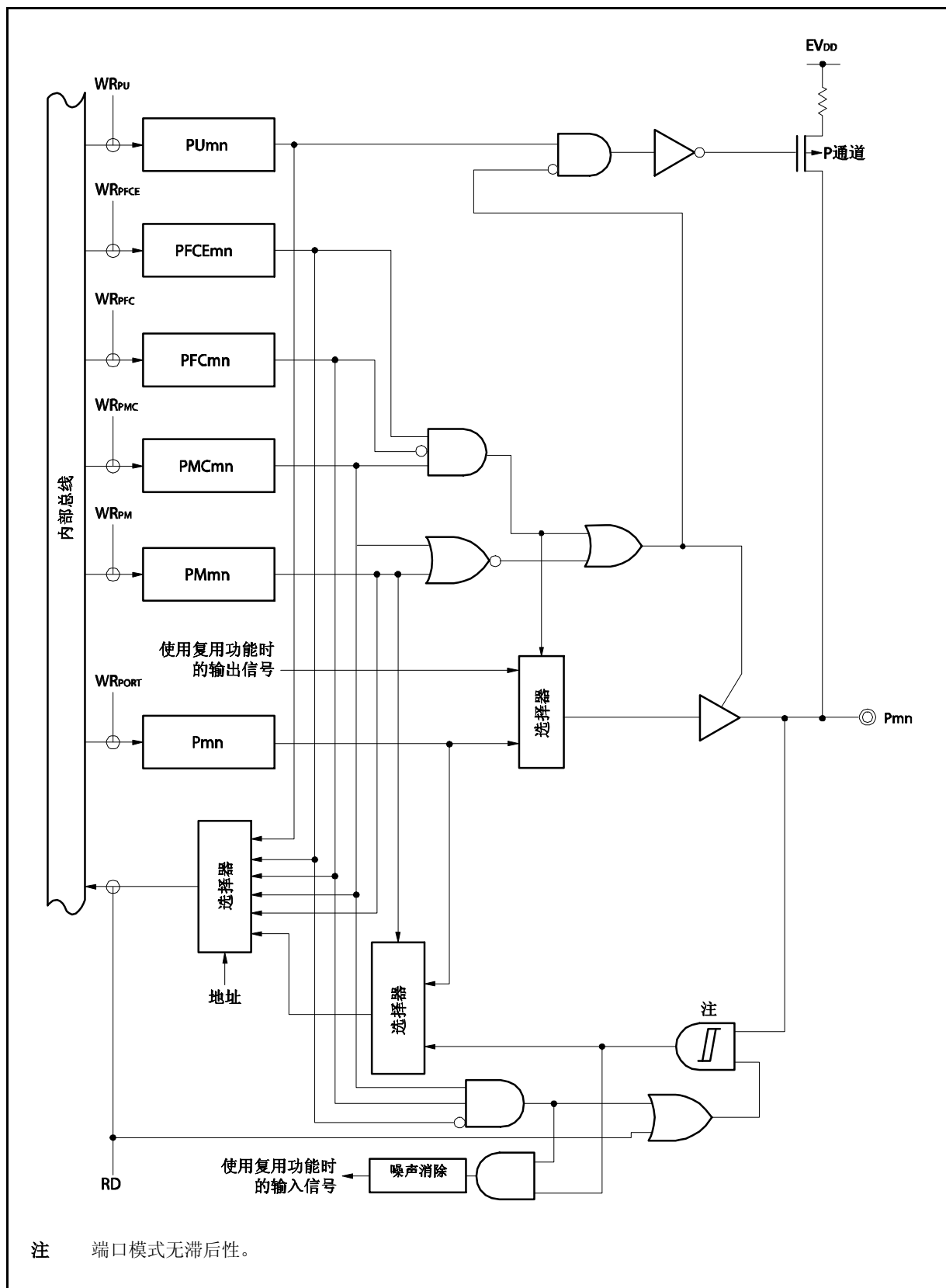


图 4-23. U-11 类型的框图



注 端口模式无滞后性。

图 4-24. U-12 类型的框图



注 端口模式无滞后性。

图 4-25. U-13 类型的框图

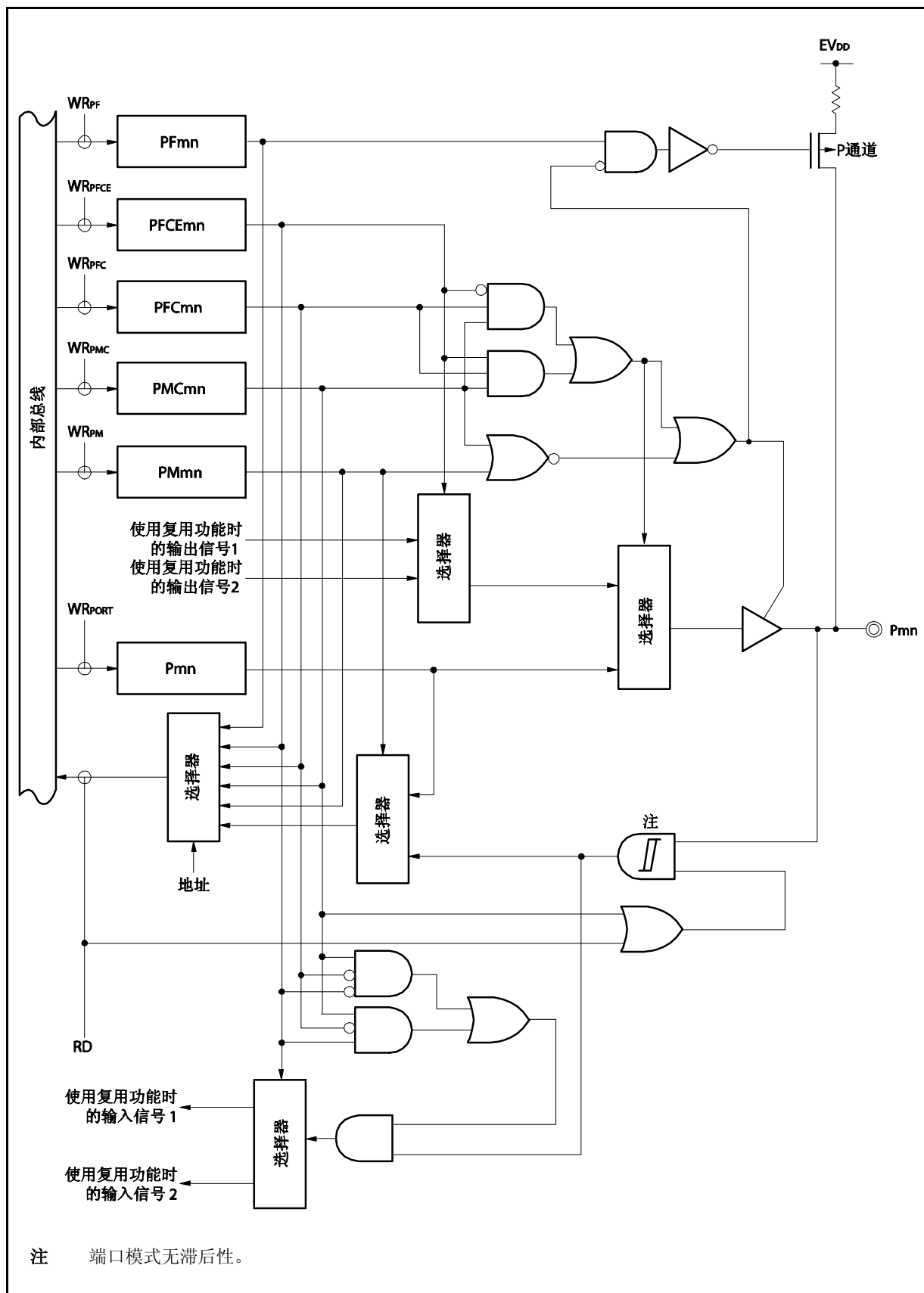
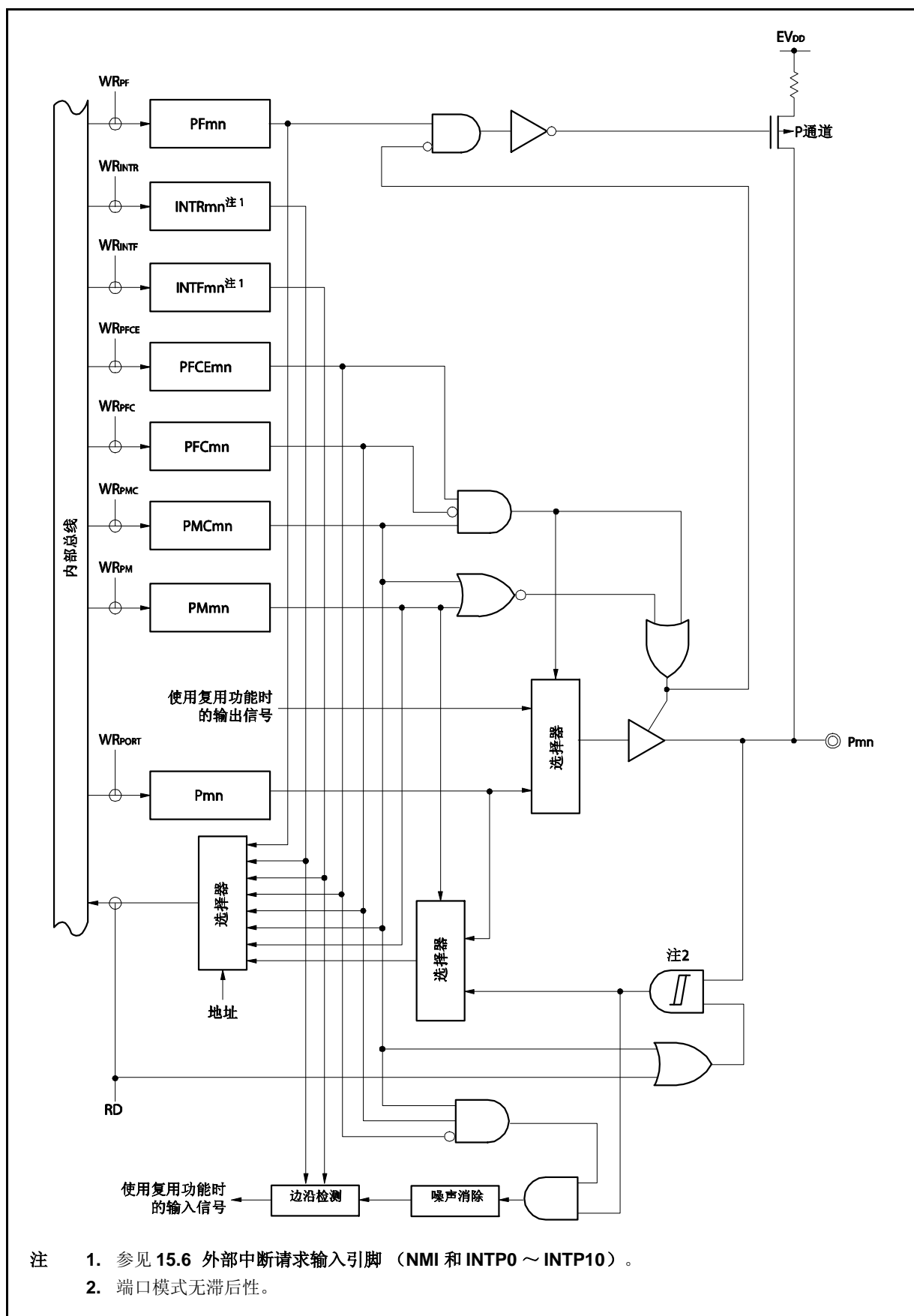
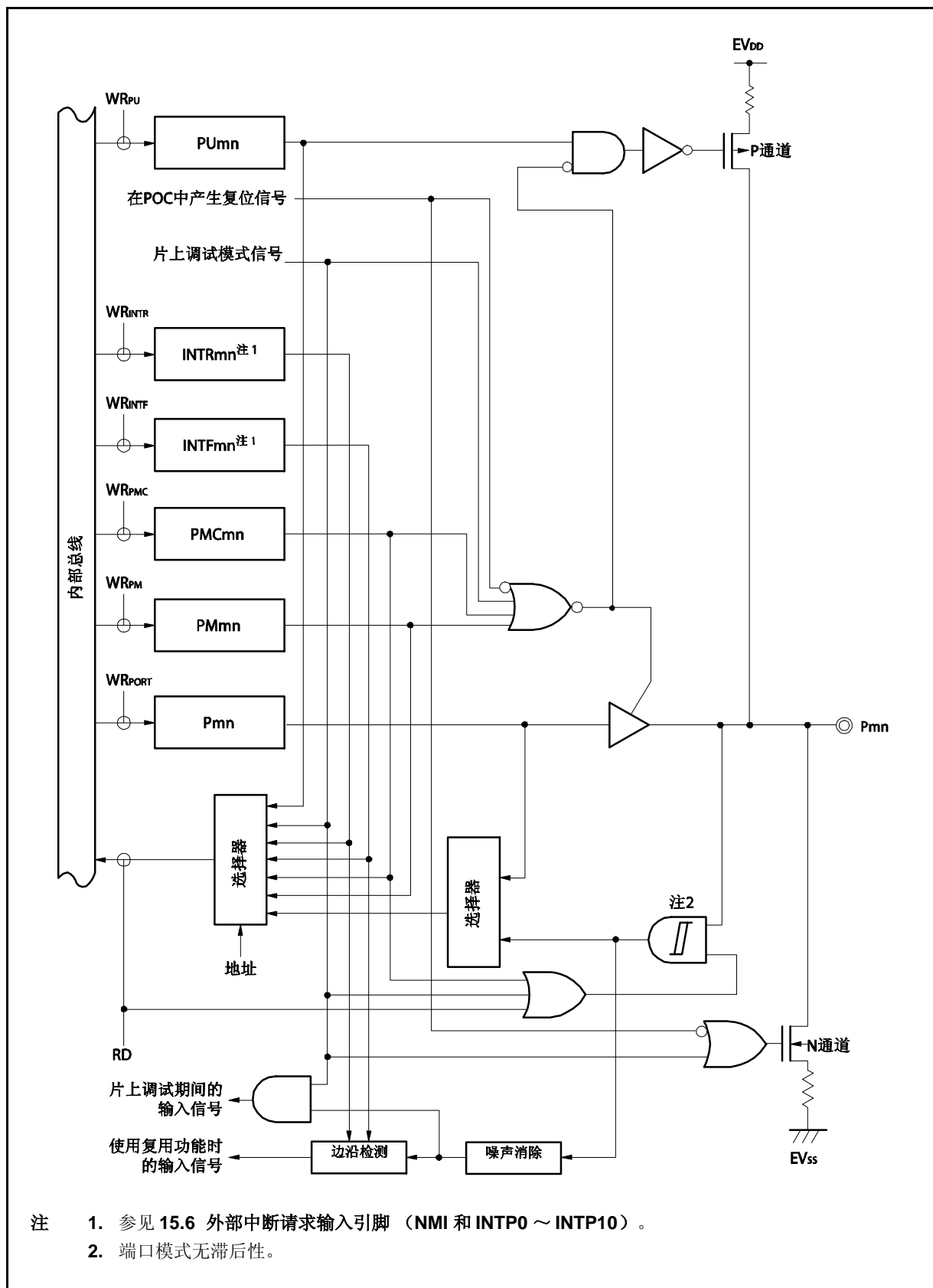


图 4-26. W-1 类型的框图



- 注 1. 参见 15.6 外部中断请求输入引脚 (NMI 和 INTP0 ~ INTP10)。
 2. 端口模式无滞后性。

图 4-27. AA-1 类型的框图



4.5 注意事项

4.5.1 设置端口引脚的注意事项

(1) 在 V850ES/HG2 系列中，通用端口功能和一些外围功能 I/O 引脚共用一个引脚。通过设置 PMCn 寄存器来切换通用端口 (端口模式) 和外围功能 I/O 引脚 (复用功能模式)。关于此寄存器的设置顺序，需要注意以下几点问题。

(a) 从端口模式切换到复用功能模式的注意事项

要从端口模式切换到复用功能模式需要按照以下顺序执行。

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

如果首先设置了 PMCn 寄存器，需要注意，再按照 PFCn 和 PFCEn 寄存器的设置时或基于这些设置，有可能发生意外的操作。

注意事项 无论是端口模式/复用功能模式，读写 Pn 寄存器如下。

- 读 Pn 寄存器: 读取端口输出锁存器的值 (当 PMn.PMnm 位= 0)，或读取引脚状态 (PMn.PMnm 位= 1)。
- 写 Pn 寄存器: 写入端口输出锁存器

<R>

(b) 复用功能模式的注意事项 (输入)

当 PMCn.PMCnm 位为 0 时，由于 PMCn 寄存器设置值以及引脚电平 AND 输出，因此输入到复用功能模块的信号为低电平。因而，由于端口设置和复用功能操作使能时序，可能会发生意外的操作。因此，要按照以下顺序从端口模式切换到复用功能模式。

- 要从端口模式切换到复用功能模式 (输入)
使用 PMCn 寄存器将引脚设置为复用功能模式，然后允许复用功能操作。
- 要从复用功能模式 (输入) 切换到端口模式
停止复用功能操作然后将引脚切换到端口模式。

第五章 时钟发生器功能

5.1 概述

V850ES/HG2 产品提供了以下时钟发生器功能。

- 主时钟振荡器
 - 时钟直通模式
 $f_x = 4 \sim 5 \text{ MHz}$ ($f_{xx} = 4 \sim 5 \text{ MHz}$)
 - PLL 模式
 $f_x = 4 \sim 5 \text{ MHz}$ ($f_{xx} = 16 \sim 20 \text{ MHz}$)
- 副时钟振荡器（由选项字节功能选择的晶体振荡器或 RC 振荡器）
 - $f_{xT} = 32.768 \text{ kHz}$ （晶体振荡器）
 - $f_{xT} = 20 \text{ kHz}$ （RC 振荡器）
- PLL（锁相环）倍频（ $\times 4$ ）功能
 - 可选择时钟直通模式/PLL 模式
- 内部振荡器
 - $f_R = 200 \text{ kHz}$ (TYP.)
- 内部系统时钟发生器
 - 7 级 (f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/8$, $f_{xx}/16$, $f_{xx}/32$, f_{xT})
- 外围时钟发生器功能
- 时钟输出功能
- 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 : 内部振荡器时钟频率
- f_{xT} : 副时钟频率

(1) 主时钟振荡器

主振荡器产生以下振荡频率 (f_x)。

- 时钟直通模式
 $f_x = 4 \sim 5 \text{ MHz}$
- PLL 模式
 $f_x = 4 \sim 5 \text{ MHz}$ ($f_{xx} = 16 \sim 20 \text{ MHz}$)

(2) 副时钟振荡器

副振荡器产生频率为 32.768 kHz 或 20 kHz 的振荡信号。

(3) 主时钟振荡器停止控制电路

该电路生成一个可停止主时钟振荡器振荡的控制信号。

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

(4) 内部振荡器

产生一个频率为 200 kHz (TYP.) 的振荡信号 (f_R)。

(5) 预分频器 1

该电路生成的时钟信号($f_{xx} \sim f_{xx}/1, 024$)将提供给以下片上外围功能: TMP0 ~ TMP3, TMQ0, TMQ1, TMM0, CSIB0, CSIB1, UARTA0 ~ UARTA2, ADC 和 WDT2。

(6) 预分频器 2

该电路对主时钟 (f_{xx}) 进行分频。

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

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

(7) 预分频器 3

该电路将主时钟振荡器产生的时钟信号(f_x)分频为一个指定的频率(32.768 kHz)并将这个时钟频率提供给钟表定时器模块。

关于更多细节, 敬请参阅第九章 钟表定时器功能。

(8) 预分频器 4

<R> 该电路生成的时钟信号 ($f_x \sim f_x/128$)将提供给片上外围功能。

提供的模块仅是 WDT2。

(9) PLL

该电路对主时钟振荡器产生的时钟信号(f_x)进行 4 倍频。

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

5.3 寄存器

(1) 处理器时钟控制寄存器 (PCC)

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

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 03H。

复位后: 03H R/W 地址: FFFF828H

	7	6	5	4	3	2	1	0
PCC	FRC	MCK	MFRC	CLS ^注	CK3	CK2	CK1	CK0
FRC	副时钟片上反馈电阻的使用							
0	使用							
1	不使用							
MCK	主时钟振荡器的控制							
0	使能振荡							
1	停止振荡							
<ul style="list-style-type: none"> 当系统在主时钟作为CPU时钟操作时，即使MCK位设为1，主时钟的操作也不停止。它在CPU时钟转变为副时钟后停止。 在设置MCK位从0到1之前，停止主时钟的片上外围功能操作。 当主时钟停止，设备在副时钟下操作时，将MCK位清零并且在将CPU时钟转变为主时钟或执行片上外围功能之前，通过软件来确定振荡稳定时间。 								
MFRC	主时钟片上反馈电阻的使用							
0	使用							
1	不使用							
CLS ^注	CPU 时钟状态 (f _{cpu})							
0	主时钟操作							
1	副时钟操作							
CK3	CK2	CK1	CK0	时钟选择(f _{clk} /f _{cpu})				
0	0	0	0	f _{xx}				
0	0	0	1	f _{xx} /2				
0	0	1	0	f _{xx} /4				
0	0	1	1	f _{xx} /8				
0	1	0	0	f _{xx} /16				
0	1	0	1	f _{xx} /32				
0	1	1	'	禁止设置				
1	'	'	'	f _{xt}				

注 CLS 位是只读位。

- 注意事项
1. 在通过 CLKOUT 向外输出时钟信号时，不要改变 CPU 时钟的频率（通过 CK3~CK0 位）。
 2. 对 CK3 位操作时，请使用位操作指令。如果使用 8 位操作指令，不要改变 CK2~CK0 位的值。

备注 x: 忽略

(a) 从副时钟操作切换至主时钟操作的举例

- <1> CK3 位 ← 1: 推荐使用位操作指令。不要改变 CK2~CK0 位的值。
- <2> 副时钟操作: 读取 CLS 位的值以检查副时钟是否已经启动。设置 CK3 位后要经过以下的时间, 副时钟操作才能启动。
最大: $1/f_{XT}$ (1/副时钟频率)
- <3> MCK 位 ← 1: 只有在停止主时钟工作的时候才可将 MCK 位置 1。

注意事项 1. 当停止主时钟工作时, 也要同时停止 PLL。另外, 工作于主时钟频率的片上外围功能也被停止。

- 2.** 如果不满足下面的条件, 那么请改变 CK2~CK0 位的设置以满足该条件, 然后再切换到副时钟工作模式。

$$\text{内部系统时钟 (f}_{CLK}) > \text{副时钟 (f}_{XT}) \times 4$$

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

[示例]

```

_DMA_DISABLE:
ctrl      0, DCHCn[r0]      --禁止 DMA 操作。 n = 0 ~ 3
<1> _SET_SUB_RUN :
st.b     r0, PRCMD[r0]
set1     3, PCC[r0]        -- CK3 位 ← 1
<2> _CHECK_CLS :
tst1     4, PCC[r0]        --等待副时钟工作的开始。
bz       _CHECK_CLS
<3> _STOP_MAIN_CLOCK :
st.b     r0, PRCMD[r0]
set1     6, PCC[r0]        -- MCK 位 ← 1, 主时钟停止。
_DMA_ENABLE:
set1     0, DCHCn[r0]      --允许 DMA 操作。 n = 0 ~ 3

```

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

(b) 从副时钟操作切换至主时钟操作的举例

- <1> MCK 位 \leftarrow 0: 主时钟开始振荡
- <2> 通过软件插入等待时间, 等待主时钟振荡稳定。
- <3> CK3 位 \leftarrow 0: 推荐使用位操作指令。不要改变 CK2~CK0 位的值。
- <4> 主时钟操作: 设置 CK3 位后要经过以下的时间, 主时钟操作才能启动。
 最大: $1/f_{XT}$ (1/副时钟频率)
 因此, 请在将 CK3 位清零后立即插入 NOP 指令或通过读取 CLS 位的方法检查主时钟是否开始工作。

注意事项 只有在主时钟振荡稳定之后才可以使工作于主时钟频率的片上外围功能开始工作。如果它们在振荡稳定之前就被启动, 则可能引起误操作。

[示例]

```

_DMA_DISABLE:
  clrl      0, DCHCn[r0]          --禁止 DMA 操作。 n = 0 ~ 3
<1> _START_MAIN_OSC :
  st.b      r0, PRCMD[r0]        --解除对特殊寄存器的保护。
  clr1      6, PCC[r0]           --启动主时钟振荡。
<2> movea   0x55, r0, r11        --等待振荡稳定时间。
  _WAIT_OST :
  nop
  nop
  nop
  addi      -1, r11, r11
  cmp       r0, r11
  bne       _WAIT_OST
<3> st.b    r0, PRCMD[r0]
  clr1      3, PCC[r0]           -- CK3  $\leftarrow$  0
<4> _CHECK_CLS :
  tst1      4, PCC[r0]           --等待主时钟开始工作。
  bnz       _CHECK_CLS
  _DMA_ENABLE:
  setl      0, DCHCn[r0]        --允许 DMA 操作。 n = 0 ~ 3

```

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

(2) 内部振荡模式寄存器 (RCM)

RCM 寄存器是用于设置内部振荡器工作模式的 8 位寄存器。

该寄存器支持 8 位或 1 位的读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF80CH

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

RSTOP	内部振荡器的振荡/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，使 RCM 寄存器的设置是有效的。
关于更多细节，敬请参阅第二十四章 选项字节功能。
 2. CPU 工作于内部振荡器时钟时(CCLS.CCLS_F 位 = 1)，不可以停止内部振荡器的工作。不要将 RSTOP 位置 1。
 3. 即使 RSTOP 位为 1，如果 CCLS.CCLS_F 位被置 1（振荡稳定时间中产生 WDT 溢出时），内部振荡器也会开始振荡。此时，RSTOP 位仍为 1。

(3) CPU 操作时钟状态寄存器 (CCLS)

CCLS 寄存器是用于指示 CPU 操作时钟状态的寄存器。

该寄存器支持 8 位或 1 位只读方式。

该寄存器复位后的值为 00H。

复位后: 00H^产 R 地址: FFFF82EH

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

CCLS _F	CPU 操作时钟状态
0	主时钟 (f _x) 或副时钟 (f _{xn}) 的操作。
1	内部振荡时钟 (f _{in}) 的操作。

注 如果复位释放后在振荡稳定时间中产生 WDT 的溢出，那么 CCLS_F 位将被置 1，并且 CCLS 寄存器的复位值变为 01H。

5.4 操作

5.4.1 各时钟的操作

下表显示了各时钟的操作状态。

表 5-1. 各时钟的操作状态

寄存器设置 以及操作状态	PCC 寄存器								
	CLK 位 = 0, MCK 位 = 0					CLS 位 = 1, MCK 位 = 0		CLS 位 = 1, MCK 位 = 1	
	复位 期间	振荡稳定 时间计数器 期间	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
目标时钟									
主时钟振荡器 (fx)	×	○	○	○	×	○	○	×	×
主系统时钟 (fxx)	×	×	○	×	×	○	×	×	×
副时钟振荡器 (fxT)	○	○	○	○	○	○	○	○	○
CPU 时钟 (fCPU)	×	×	×	×	×	○	×	○	×
内部系统时钟 (fCLK)	×	×	○	×	×	○	×	○	×
主时钟 (PLL 模式, fxx)	×	注 1	○	注 2	×	○	○	×	×
外围时钟 (fxx ~ fxx/1, 024)	×	×	○	×	×	○	×	×	×
WT 时钟 (主)	×	×	○	○	×	○	○	×	×
WT 时钟 (副)	○	○	○	○	○	○	○	○	○
WDT2 时钟 (内部振荡)	×	○	○	○	○	○	○	○	○
WDT2 时钟 (主)	×	×	○	×	×	○	×	×	×

- 注 1. 在 1/2 振荡稳定时间之后振荡开始, 并且在锁止时间后提供稳定时钟。
2. 在 IDLE1 模式下可操作。在 IDLE2 模式下停止。

备注 ○: 可操作
×: 停止

5.4.2 时钟输出功能

时钟输出功能用于将内部系统时钟(fCLK)从 CLKOUT 引脚输出。

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

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

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

5.5 PLL 功能

5.5.1 概述

在 V850ES/HG2 系列产品中，CPU 和片上外围功能的时钟信号可以选择频率为晶振频率的 4 倍的 PLL 模式或时钟直通模式。

使用 PLL 功能时： 输入时钟 = 4 ~ 5 MHz（输出：16 ~ 20 MHz）
时钟直通模式： 输入时钟 = 4 ~ 5 MHz（输出：4 ~ 5 MHz）

5.5.2 寄存器

(1) PLL 控制寄存器 (PLLCTL)

PLLCTL 寄存器是用来控制 PLL 功能的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 01H。

复位后: 01H R/W 地址: FFFF82CH							
PLLCTL	7	6	5	4	3	2	1 0
	0	0	0	0	0	0	SELPLL PLLON
PLLON	PLL 操作停止寄存器						
0	PLL 停止						
1	PLL 操作 (PLL 操作开始后, 振荡频率所需的锁止时间)						
SELPLL	CPU 操作时钟选择寄存器						
0	时钟直通模式						
1	PLL 模式						

注意事项

1. 当 PLLON 位被清零时，SELPLL 位也会被自动清零（进入时钟直通模式）。
2. 只有在 PLL 时钟频率稳定后，SELPLL 位才能够被置 1。如果时钟频率没有稳定（未入锁），那么无论向该位写入的数据为何值，SELPLL 位都会被写入"0"。

(2) 锁定寄存器 (LOCKR)

相位锁定将在上电后或 STOP 模式释放后按照给定的频率实现，入锁时间（频率稳定时间）是指这个过程所需要的稳定时间。直到频率稳定之前的状态，被叫做入锁状态，频率稳定后的状态叫做锁定状态。

包含 LOCK 位的 LOCKR 寄存器用于反映 PLL 频率稳定状态。

该寄存器支持 8 位或 1 位只读方式。

该寄存器复位后的值为 00H。

复位后: 00H R 地址: FFFF824H

	7	6	5	4	3	2	1	0
LOCKR	0	0	0	0	0	0	0	LOCK

LOCK	PLL 锁定状态 检测
0	锁定状态
1	未锁定状态

注意事项 LOCK 寄存器不能实时反映 PLL 的锁定状态。置位和清零条件如下。

[置位条件]

- 系统复位^注
- 在 IDLE2 或 STOP 模式下
- 设置 PLL 为停止时（将 PLLCTL.PLLON 位清零）
- 主时钟停止且 CPU 工作于副时钟时（设置 PCC.CK3 位为 1 并且设置 PCC.MCK 位为 1）

注 该寄存器会被复位信号置为 01H，而在复位释放并经过振荡稳定时间之后，该寄存器会被设置为 00H。

[清零条件]

- 复位释放后振荡稳定时间（OSTS 寄存器的默认时间（参见 17.2 (3) 振荡稳定时间选择寄存器（OSTS）））溢出时。
- 在 PLL 工作状态下设置了 STOP 模式，STOP 模式释放后，振荡稳定定时器溢出时（时间由 OSTS 寄存器设置）。
- 当 PLLCTL.PLLON 位的设置从 0 改变为 1，PLL 入锁时间定时器产生溢出时（时间由 PLLS 寄存器设置）。
- 在 PLL 工作状态下设置了 IDLE2 模式，IDLE2 模式解除时，所插入的设立时间（由 OSTS 寄存器设置的时间）结束后。

(3) PLL 入锁时间指定寄存器(PLLS)

PLLS 寄存器是用来选择在将 PLLCTL.PLLON 位从 0 变为 1 时的 PLL 入锁时间的 8 位寄存器。

该寄存器支持 8 位读写方式。

该寄存器复位后的值为 03H。

复位后: 03H R/W 地址: FFFF6C1H

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

PLLS1	PLLS0	PLL 锁止时间的选择
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. 将入锁时间设置为 800 μ s 或更长。
 2. 入锁期间不要改变 PLLS 寄存器的设置。

备注 f_x : 主时钟振荡频率

(4) 可编程时钟模式寄存器 (PCLM)

PCLM 是用来控制 PCL 输出的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。

复位后: 00H R/W 地址: FFFFF82FH

	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	禁止PCL 引脚输出 (PCL引脚固定在低电平)
1	使能PCL 引脚输出

注意事项 首先设置端口相关的控制寄存器（PM，PMC，PFC 和 PFCE 寄存器等），然后将 PCLE 位置 1。

PCK1	PCK0	PLL 输出时钟的选择
0	0	$f_{PCL}/2$
0	1	$f_{PCL}/4$
1	0	$f_{PCL}/8$
1	1	$f_{PCL}/16$

注意事项 仅在 PLL 工作期间将 PCLE 位置 1。为了停止 PLL，可将 PCLE 位清零。

备注 f_{PCL} : 可编程频率

<R>

5.5.3 使用方法

(1) 使用 PLL 时

- 复位信号被释放之后，PLL 处于工作状态(PLLCTL.PLLON 位 = 1)，但由于默认模式为时钟直通模式(PLLCTL.SELPLL 位= 0)，为使 PLL 有效，请选择 PLL 模式(SELPLL 位 = 1)。
- 允许 PLL 工作，首先要将 PLLON 位置 1，然后在 LOCKR.LOCK 位 = 0 之后将 SELPLL 位置 1。停止 PLL 工作，首先选择时钟直通模式(SELPLL 位 = 0)，等待 8 个或更多时钟周期，然后停止 PLL 工作(PLLON 位 = 0)。
- 系统模式切换至 IDLE2 或 STOP 模式时，PLL 将无条件停止工作，系统模式从 IDLE2 或 STOP 切换回之前的模式时，PLL 将恢复工作。恢复时间如下所示。

(a) 从时钟直通模式切换至 IDLE2 或 STOP 模式时

- STOP 模式： 设置 OSTS 寄存器以便使振荡稳定时间为 1 ms (min.) 或更长。
- IDLE2 模式： 设置 OSTS 寄存器以便使设立时间为 350 μ s (min.) 或更长。

(b) PLL 工作模式中，切换至 IDLE 2 或 STOP 模式时

- STOP 模式： 设置 OSTS 寄存器以便使振荡稳定时间为 1 ms (min.) 或更长。
- IDLE2 模式： 设置 OSTS 寄存器以便使设立时间为 800 μ s (min.) 或更长。

向 IDLE1 模式切换时，PLL 不会自动停止。请根据需要停止 PLL 的工作。

(2) 不使用 PLL 时

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

第六章 16 位定时器/事件计数器 P (TMP)

定时器 P (TMP) 是一个 16 位的定时器/事件计数器。

V850ES/HG2 具有 4 个定时器/事件计数器通道，TMP0 ~ TMP3。

6.1 概述

以下是 TMPn 的概况。

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

备注 n = 0 ~ 3

6.2 功能

TMPn 具有以下功能。

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

备注 n = 0 ~ 3

6.3 配置

TMPn 包含以下硬件。

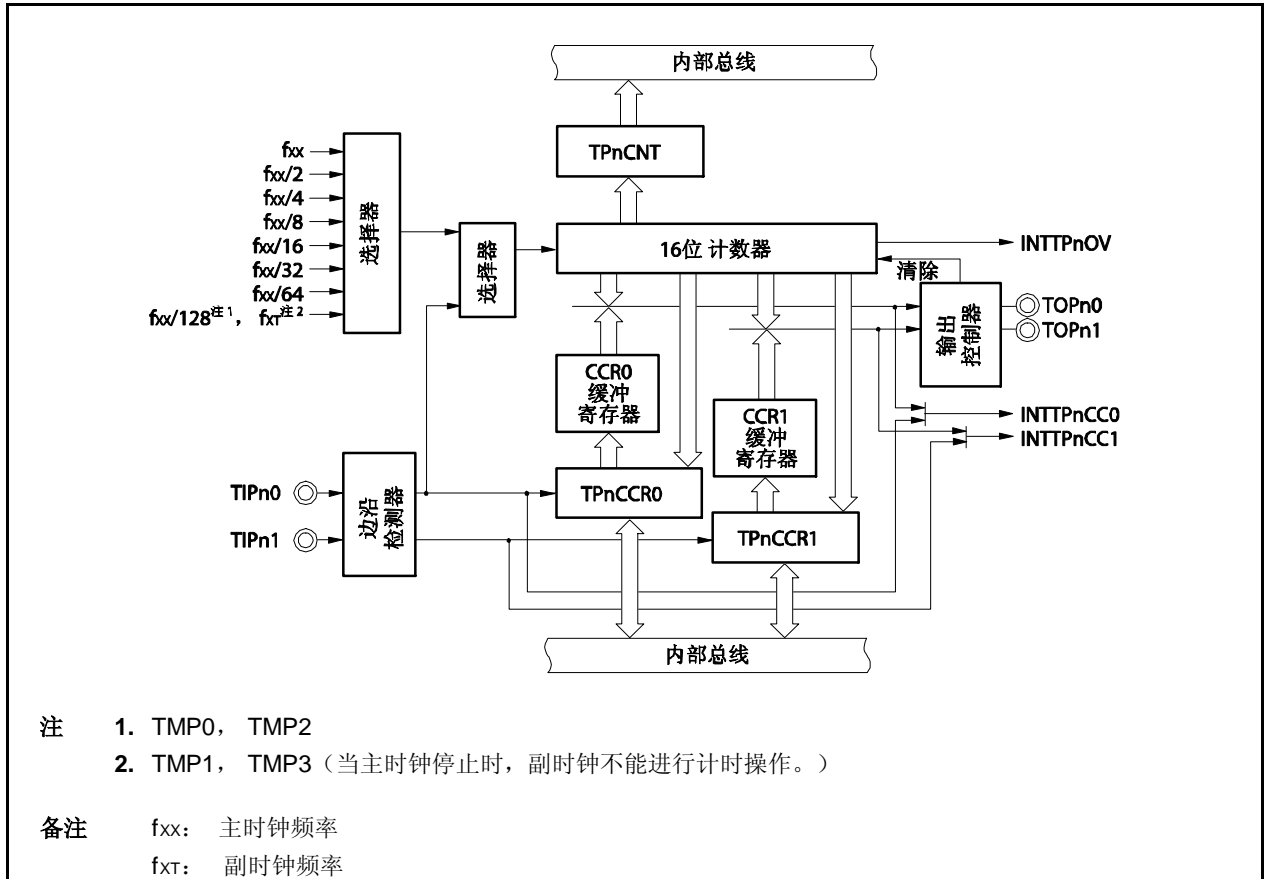
表 6-1. TMPn 的配置

器件	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 ^{注 1} , TIPn1 引脚)
定时器输出	2 (TOPn0, TOPn1 引脚)
控制寄存器 ^{注 2}	TMPn 控制寄存器 0, 1 (TPnCTL0, TPnCTL1) TMPn I/O 控制寄存器 0 ~ 2 (TPnIOC0 ~ TPnIOC2) TMPn 选项寄存器 0 (TPnOPT0)

- 注
1. TIPn0 引脚的复用功能包括捕捉触发输入信号，外部事件计数输入信号和外部触发输入信号。
 2. 使用 TIPn0, TIPn1, TOPn0 和 TOPn1 引脚功能时，请参照表 4-15 端口引脚作为复用引脚使用。

备注 n = 0 ~ 3

图 6-1. TMPn 的框图



(1) 16 位计数器

该 16 位计数器可以对内部时钟或外部事件进行计数。

可以使用 TPnCNT 寄存器读取该计数器的计数值。

当 TPnCTL0.TPnCE 位 = 0 时，16 位计数器的值为 FFFFH。如果此时对 TPnCNT 寄存器进行读取，则读取值将为 0000H。

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

(2) CCR0 缓冲寄存器

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

当 TPnCCR0 寄存器被作为比较寄存器使用时，向 TPnCCR0 寄存器写入的数据将被传送至 CCR0 缓冲寄存器。如果 16 位计数器的计数值与 CCR0 缓冲寄存器的值相等，会产生一个比较相等中断请求信号 (INTTPnCC0)。

CCR0 缓冲寄存器不能被读/写。

复位后 CCR0 缓冲寄存器和 TPnCCR0 寄存器都将被清零。

(3) CCR1 缓冲寄存器

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

当 TPnCCR1 寄存器被作为比较寄存器使用时，向 TPnCCR1 寄存器写入的数据将被传送至 CCR1 缓冲寄存器。如果 16 位计数器的计数值与 CCR1 缓冲寄存器的值相等，会产生一个比较相等中断请求信号 (INTTPnCC1)。

CCR1 缓冲寄存器不能被读/写。

复位后 CCR1 缓冲寄存器和 TPnCCR1 寄存器都将被清零。

(4) 边沿检测器

该电路用于检测 TIPn0 和 TIPn1 引脚输入的有效边沿。通过 TPnIOC1 和 TPnIOC2 寄存器选择无有效沿，上升沿，下降沿或双沿有效。

(5) 输出控制器

该电路用于控制 TOPn0 和 TOPn1 引脚的输出。输出控制器是由 TPnIOC0 寄存器控制的。

(6) 选择器

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

6.4 寄存器

用于控制 TMPn 的寄存器如下所述。

- 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. 使用 TIPn0, TIPn1, TOPn0 和 TOPn1 引脚功能时, 请参照表 4-15 端口引脚作为复用引脚使用。
 2. n = 0 ~ 3

(1) TMPn 控制寄存器 0 (TPnCTL0)

TPnCTL0 寄存器是用于控制 TMPn 操作的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

可使用软件向 TPnCTL0 寄存器写入相同的数值。

复位后: 00H R/W 地址: TP0CTL0 FFFF590H, TP1CTL0 FFFF5A0H,
TP2CTL0 FFFF5B0H, TP3CTL0 FFFF5C0H

	7	6	5	4	3	2	1	0
TPnCTL0 (n = 0 ~ 3)	TPnCE	0	0	0	0	TPnCKS2	TPnCKS1	TPnCKS0

TPnCE	TMPn 操作控制
0	禁用 TMPn 操作 (TMPn 异步复位 ^{注1})。
1	启用 TMPn 操作。TMPn 操作开始。

TPnCKS2	TPnCKS1	TPnCKS0	内部计数时钟选择	
			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} /8	
1	0	0	f _{xx} /16	
1	0	1	f _{xx} /32	
1	1	0	f _{xx} /64	
1	1	1	f _{xx} /128	f _{xT} ^{注2}

- 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, 副时钟不能进行计数操作。

- 注意事项 1. 在 TPnCE 位 = 0 时才可对 TPnCKS2 ~ TPnCKS0 位进行设置。
当将 TPnCE 位的值从 0 改变为 1 时, 可同时对 TPnCKS2 ~ TPnCKS0 位进行设置。
2. 请务必将第 3 到第 6 位清零。

备注 f_{xx}: 主时钟频率
f_{xT}: 副时钟频率

(2) TMPn 控制寄存器 1 (TPnCTL1)

TPnCTL1 寄存器是用于控制 TMPn 操作的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

该寄存器复位后的值为 00H。

(1/2)

复位后: 00H R/W 地址: TP0CTL1 FFFFF591H, TP1CTL1 FFFFF5A1H,
TP2CTL1 FFFFF5B1H, TP3CTL1 FFFFF5C1H

	7	6	5	4	3	2	1	0
TPnCTL1 (n = 0 ~ 3)	TPnSYE	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0

TPnSYE	调谐操作模式启用控制	
0	独立操作模式(异步操作模式)	
1	调谐操作模式(从操作规格) 在该模式下, 定时器P 能够与主定时器同步操作。	
	主定时器	从定时器
	TMP0	TMP1 -
	TMP2	TMP3 TMQ0
对于 调谐操作模式, 参见6.6 定时器调谐操作功能。		
注意事项 一定要清零 TP0SYE及TP2SYE位。		

TPnEST	软件触发控制
0	-
1	为外部触发输入发生一有效信号。 · 单脉冲输出模式: 作为触发写入1 到 TPnEST 位时, 输出单脉冲。 · 外部触发脉冲输出模式: 作为触发写入1 到 TPnEST 位时, 输出一 PWM 波形。

- 注意事项 1. 对 TPnEST 位的设置只有在外部触发脉冲输出模式或单脉冲输出模式才有效。在其它模式下, 对该位的置 1 操作将被忽略。
2. 请务必将第 3, 4 位清零。

TPnEEE	计数时钟选择
0	禁用外部事件计数输入操作。 (采用由 TPnCTL0.TPnCK0~ TPnCK2 位选择的计数时钟进行计数。)
1	启用外部事件计数输入操作。 (在外部事件计数输入信号的有效边沿进行计数。)
TPnEEE 位选择内部计数时钟计数操作或外部事件计数输入的有效边沿。	

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	单脉冲输出模式
1	0	0	PWM 输出模式
1	0	1	自由运行定时器模式
1	1	0	脉冲宽度测量模式
1	1	1	禁止时钟

- 注意事项**
1. 在外部事件模式下，无论 TPnEEE 位如何设置，外部事件计数输入都是被选中的。
 2. 只有在 TPnCTL0.TPnCE 位 = 0 时才可以对 TPnEEE 和 TQnMD2 ~ TPnMD0 位进行设置 (TPnCE 位 = 1 时，可以对这些位写入相同的值)。如果在 TPnCE 位 = 1 时对这些位进行了改写，那么操作结果将不能被保证。若由于误操作引起了改写，则要将 TQnCE 位清零然后再次设置这些寄存器位。

<R> (3) TMPn I/O 控制寄存器 0 (TPnIOC0)

TPnIOC0 寄存器是用于控制定时器输出的 8 位寄存器 (TOPn0, TOPn1 引脚)。

可对该寄存器进行 8 位或 1 位的读写操作。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: TP0IOC0 FFFFF592H, TP1IOC0 FFFFF5A2H,
TP2IOC0 FFFFF5B2H, TP3IOC0 FFFFF5C2H

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

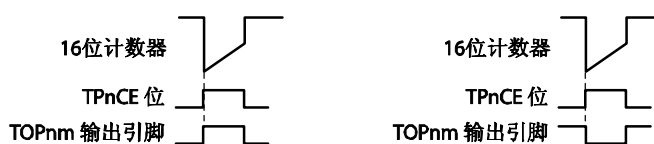
TPnOL1	TPn1 引脚输出电平设置 ^注
0	TPn1 引脚输出在高电平开始
1	TPn1 引脚输出在低电平开始

TPnOE1	TPn1 引脚输出设置
0	禁用定时器输出 · 当 TPnOL1 位=0: 由 TOPn1 引脚输出低电平 · 当 TPnOL1 位=1: 由 TOPn1 引脚输出高电平
1	启用定时器输出 (由 TOPn1 引脚输出一方波)。

TPnOL0	TPn0 引脚输出电平设置 ^注
0	TPn0 引脚输出在高电平开始
1	TPn0 引脚输出在低电平开始

TPnOE0	TPn0 引脚输出设置
0	禁用定时器输出 · 当 TPnOL0 位=0: 由 TOPn0 引脚输出低电平 · 当 TPnOL0 位=1: 由 TOPn0 引脚输出高电平
1	启用定时器输出 (由 TOPn0 引脚输出一方波)。

注 定时器输出引脚 (TOPnm) 的输出电平由以下 TPnOLm 位指定。
· 当 TPnOLm 位=0 · 当 TPnOLm 位=1



- 注意事项
1. 在 TPnCTL0.TPnCE 位 = 0 时才可对 TPnOL1, TPnOE1, TPnOL0 和 TPnOE0 位进行改写 (TPnCE 位 = 1 时只可以向这些位写入相同的值)。若在 TPnCE 位 = 1 由于误操作引起了改写, 则要将 TPnCE 位清零然后再次设置这些寄存器位。
 2. 在 TPnCE 位和 TPnOEm 位为 0 时, 即便对 TPnOLm 位进行了操作, TOPnm 引脚的输出电平也不能确定。

备注 n=0~3, m=0, 1

(4) TMPn I/O 控制寄存器 1 (TPnIOC1)

TPnIOC1 寄存器是用于控制捕捉触发输入信号 (TIPn0, TIPn1 引脚) 有效沿的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: TP0IOC1 FFFF593H, TP1IOC1 FFFF5A3H,
TP2IOC1 FFFF5B3H, TP3IOC1 FFFF5C3H

	7	6	5	4	3	2	1	0
TPnIOC1 (n = 0 ~ 3)	0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0

TPnIS3	TPnIS2	捕捉触发输入信号(TIPn1 引脚)有效边沿设置
0	0	无边沿检测(捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升和下降沿

TPnIS1	TPnIS0	捕捉触发输入信号(TIPn0 引脚)有效边沿设置
0	0	无边沿检测(捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升和下降沿

- 注意事项**
1. 在 TPnCTL0.TPnCE 位 = 0 时才可对 TPnIS3 ~ TPnIS0 位进行改写 (TPnCE 位 = 1 时只可以向这些位写入相同的值)。若在 TPnCE 位 = 1 由于误操作引起了改写, 则要将 TPnCE 位清零然后再次设置这些寄存器位。
 2. TPnIS3 ~ TPnIS0 位只有在自由运行定时器模式和脉宽测量模式下才有效。其它模式下, 无法进行捕捉操作。

(5) TMPn I/O 控制寄存器 2 (TPnIOC2)

TPnIOC2 寄存器是用于控制外部事件计数输入信号 (TIPn0 引脚) 有效沿和外部触发输入信号有效沿的 8 位寄存器 (TIPn0 引脚)。

可对该寄存器进行 8 位或 1 位的读写操作。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: TP0IOC2 FFFF594H, TP1IOC2 FFFF5A4H,
TP2IOC2 FFFF5B4H, TP3IOC2 FFFF5C4H

	7	6	5	4	3	2	1	0
TPnIOC2 (n = 0 ~ 3)	0	0	0	0	TPnEES1	TPnEES0	TPnETS1	TPnETS0

TPnEES1	TPnEES0	外部事件计数输入信号(TIPn0 引脚)有效边沿设置
0	0	无边沿检测(外部事件计数无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升和下降沿

TPnETS1	TPnETS0	外部触发输入信号(TIPn0 引脚)有效边沿设置
0	0	无边沿检测(外部触发无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升和下降沿

- 注意事项**
1. 在 TPnCTL0.TPnCE 位 = 0 时才可对 TPnEES1, TPnEES0, TPnETS1 和 TPnETS0 位进行改写 (TPnCE 位 = 1 时只可以向这些位写入相同的值)。若在 TPnCE 位 = 1 由于误操作引起了改写, 则要将 TPnCE 位清零然后再次设置这些寄存器位。
 2. TPnEES1 和 TPnEES0 位只有在 TPnCTL1.TPnEEE 位 = 1 或设置了外部事件计数模式 (TPnCTL1.TPnMD2 ~ TPnCTL1.TPnMD0 位 = 001) 时才有效。
 3. TPnETS1 和 TPnETS0 位只有在外部触发脉冲输出模式 (TPnCTL1.TPnMD2 ~ TPnCTL1.TPnMD0 位 = 010) 或单脉冲输出模式 (TPnCTL1.TPnMD2 ~ TPnCTL1.TPnMD0 = 011) 下才有效。

(6) TMPn 选项寄存器 0 (TPnOPT0)

TPnOPT0 寄存器是用于设置捕捉/比较操作和检测溢出的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

该寄存器复位后的值为 00H。

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

	7	6	5	4	3	2	1	0
TPnOPT0 (n = 0 ~ 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
TPnCCS1 位仅在自由运行定时器模式下设置有效。	

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 位仅在自由运行定时器模式下设置有效。	

TPnOVF	TMPn 溢出检测标志
设置(1)	发生溢出
复位(0)	TPnOVF 位写 0 或 TPnCTL0.TPnCE 位 = 0
<ul style="list-style-type: none"> • The TPnOVF bit is set to 1 当 16 位计数器的计数值在自由运行定时器模式下或脉冲宽度测量模式下溢出 FFFFH ~ 0000H。 • 在 TPnOVF 位设置为 1 的同时发生一中断请求信号 (INTTPnOV)。INTTPnOV 信号只在自由运行定时器模式下或脉冲宽度测量模式下发生。 • 即使当 TPnOVF 位 = 1 时读取 TPnOVF 位或 TPnOPT0, TPnOVF 位也不被清零。 • TPnOVF 位可被读取或写入, 但 TPnOVF 位不能通过软件设置为 1。 写 1 操作对 TMPn 操作没有影响。 	

注意事项 1. 在 TPnCE 位 = 0 时才可对 TPnCCS1 和 TPnCCS0 位进行改写 (TPnCE 位 = 1 时只可以向这些位写入相同的值)。若在 TPnCE 位 = 1 由于误操作引起了改写, 则要将 TPnCE 位清零然后再次设置这些寄存器位。

2. 请务必将第 1~3 位, 第 6 和第 7 位清零。

(7) TMPn 捕捉/比较寄存器 0 (TPnCCR0)

根据选择模式的不同，TPnCCR0 可用作捕捉寄存器或比较寄存器。

该寄存器只有在自由运行模式下根据 TPnOPT0.TPnCCS0 位设置的不同，可以用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TPnCCR0 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。

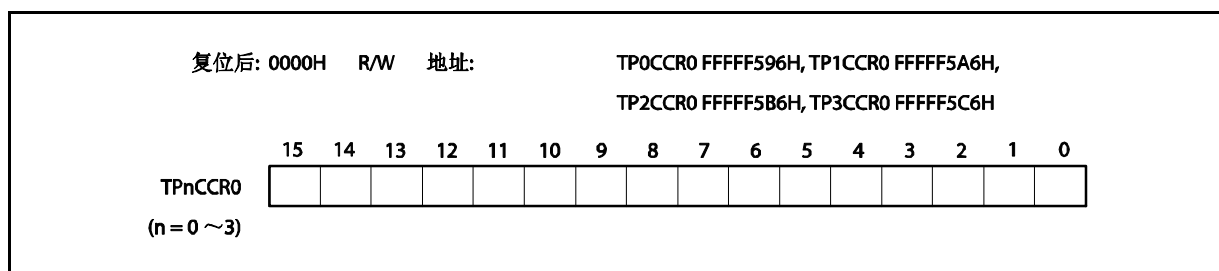
TPnCCR0 寄存器可以在操作过程中进行读写。

可对该寄存器进行 16 位的读写操作。

该寄存器复位后的值为 0000H。

注意事项 以下情况下禁止访问 TPnCCR0 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上外围 I/O 寄存器。

- 当 CPU 工作于副时钟且主时钟振荡停止时
- 当 CPU 工作于内部振荡时钟时



(a) 作为比较寄存器时的功能

TPnCCR0 寄存器可以在 TPnCTL0.TPnCE 位 = 1 时被改写。

向 TPnCCR0 寄存器中设置的数值将被传送到 CCR0 缓冲寄存器中。当 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相等时，会产生一个比较相等中断请求信号(INTTPnCC0)。如果允许了 TOPn0 引脚输出，那么 TOPn0 引脚输出的电平将被反转。

当 TPnCCR0 在间隔定时器模式，外部事件计数模式，外部触发脉冲输出模式，单脉冲输出模式或 PWM 输出模式下被用作周期寄存器时，如果 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相等，那么 16 位计数器将被清零(0000H)。

(b) 作为捕捉寄存器时的功能

当 TPnCCR0 寄存器在自由运行定时器模式下被用作捕捉寄存器时，如果检测到了捕捉触发输入引脚 (TIPn0 引脚) 的有效边沿，那么 16 位计数器的计数值将被保存到 TPnCCR0 寄存器中。在脉宽测量模式下，如果检测到了捕捉触发输入引脚 (TIPn0 引脚) 的有效边沿，那么 16 位计数器的计数值将被保存到 TPnCCR0 寄存器中并且同时 16 位计数器被清零(0000H)。

即使捕捉操作和读取 TPnCCR0 寄存器操作冲突，仍可正确读出 TPnCCR0 寄存器中的值。

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 7-2. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

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

(8) TMPn 捕捉/比较寄存器 1 (TPnCCR1)

根据选择模式的不同，TPnCCR1 可用作捕捉寄存器或比较寄存器。

该寄存器只有在自由运行模式下根据 TPnOPT0.TPnCCS1 位设置的不同，可以用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TPnCCR1 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。

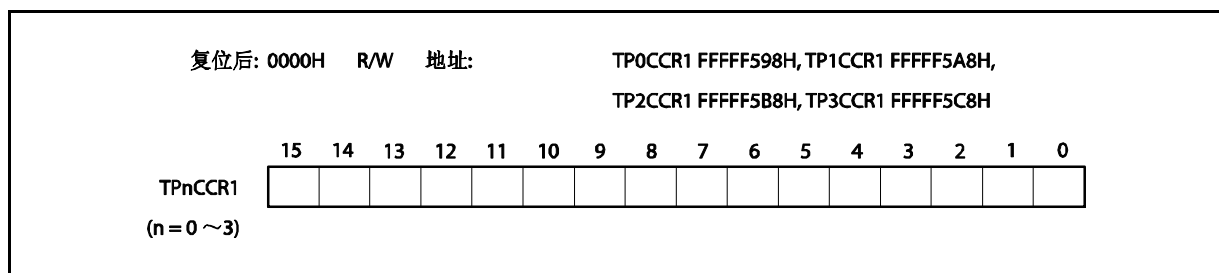
TPnCCR1 寄存器可以在操作过程中进行读写。

可对该寄存器进行 16 位的读写操作。

该寄存器复位后的值为 0000H。

注意事项 以下情况下禁止访问 TPnCCR1 寄存器。详细内容请参考 3.4.8(2) 访问特定的片上外围 I/O 寄存器。

- 当 CPU 工作于副时钟且主时钟振荡停止时
- 当 CPU 工作于内部振荡时钟时



(a) 作为比较寄存器时的功能

TPnCCR1 寄存器可以在 TPnCTL0.TPnCE 位 = 1 时被改写。

向 TPnCCR1 寄存器中设置的数值将被传送到 CCR1 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 缓冲寄存器中的数值相等时，会产生一个比较相等中断请求信号(INTTPnCC1)。如果允许了 TOPn1 引脚输出功能，那么 TOPn1 引脚输出的电平将被反转。

(b) 作为捕捉寄存器时的功能

当 TPnCCR1 寄存器在自由运行定时器模式下被用作捕捉寄存器时，如果检测到了捕捉触发输入引脚 (TIPn1 引脚) 的有效边沿，那么 16 位计数器的计数值将被保存到 TPnCCR1 寄存器中。在脉宽测量模式下，如果检测到了捕捉触发输入引脚 (TIPn1 引脚) 的有效边沿，那么 16 位计数器的计数值将被保存到 TPnCCR1 寄存器中并且同时 16 位计数器被清零(0000H)。

即使捕捉操作和读取 TPnCCR1 寄存器的操作产生冲突，仍可正确读出 TPnCCR1 寄存器中的值。

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 6-3. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

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

6.5 操作

TMPn 可进行以下操作。

操作	TPnCTL1.TPnEST 位 (软件触发位)	TIPn0 引脚 (外部触发输入)	捕捉/比较寄存器设置	比较寄存器写入
间隔定时器模式	无效	无效	仅比较	可随时写入
外部事件计数模式 ^{注 1}	无效	无效	仅比较	可随时写入
外部触发脉冲输出模式 ^{注 2}	有效	有效	仅比较	同时写入
单脉冲输出模式 ^{注 2}	有效	有效	仅比较	可随时写入
PWM 输出模式	无效	无效	仅比较	同时写入
自由运行定时器模式	无效	无效	可选	可随时写入
脉宽测量模式 ^{注 2}	无效	无效	仅捕捉	不可操作

- 注**
1. 使用外部事件计数模式时，请关闭对 TIPn0 引脚捕捉触发输入的有效边沿的检测（通过将 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位清为“00”）。
 2. 使用外部触发脉冲输出模式，单脉冲输出模式和脉宽测量模式时，请选择内部时钟作为计数时钟（通过将 TPnCTL1.TPnEEE 位清零）。

备注 n = 0 ~ 3

6.5.1 间隔定时器模式 (TPnMD2 ~ TPnMD0 位 = 000)

在间隔定时器模式下，当 TPnCTL0.TPnCE 位 = 1 时，每隔指定间隔时间都会产生一个中断请求信号 (INTTPnCC0)，同时可以从 TOPn0 引脚输出一个周期为 2 倍间隔时间的方波。

通常情况下，间隔定时器模式中不使用 TPnCCR1 寄存器。

图 6-2. 间隔定时器的配置图

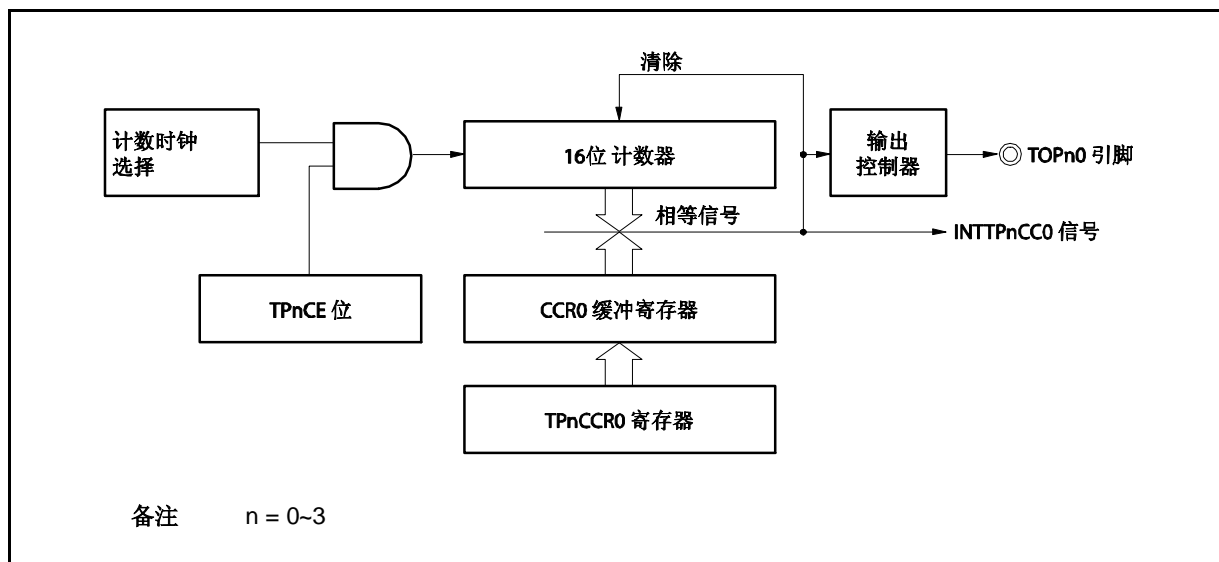
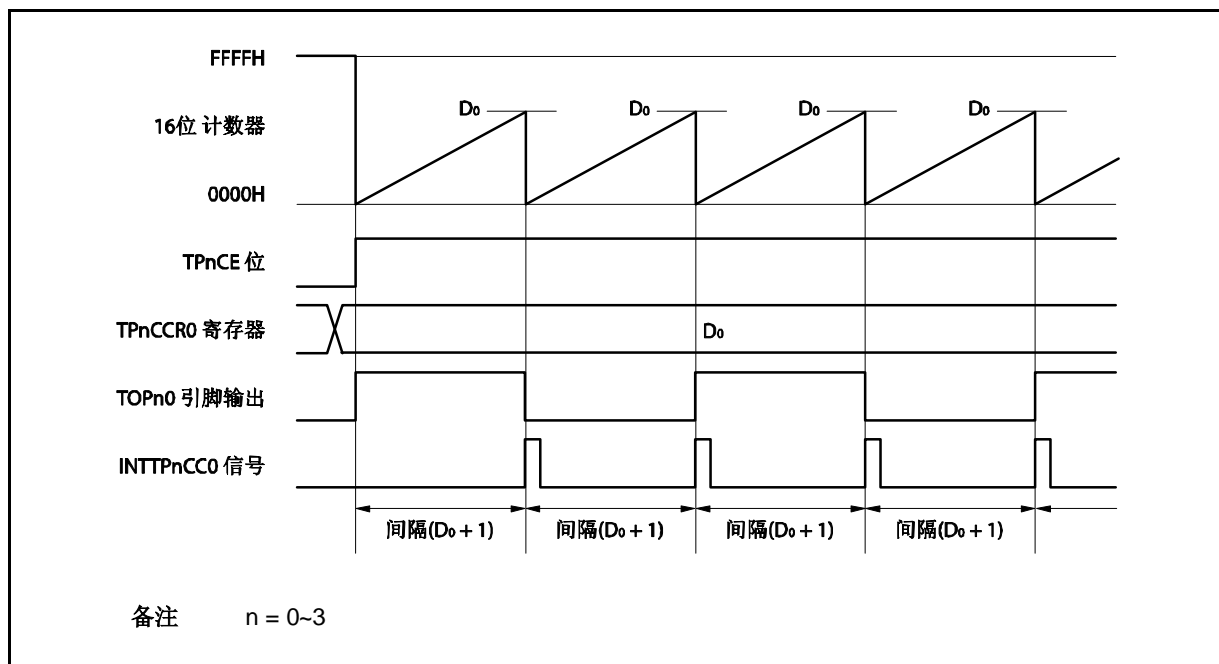


图 6-3. 间隔定时器模式操作的基本时序



16 位计数器将在 TPnCE 位被置 1 时，与计数时钟同步从 FFFFH 清零为 0000H 并开始计数。此时，TOPn0 引脚的输出电平被反转。另外，TPnCCR0 寄存器的设置值也会在此时被传送到 CCR0 缓冲寄存器。

当 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相等时，16 位计数器清零，TOPn0 引脚输出电平反转并且产生一个比较相等中断请求信号 (INTTPnCC0)。

间隔时间可由以下算式求得。

$$\text{间隔时间} = (\text{TPnCCR0 寄存器的设定值} + 1) \times \text{计数时钟周期}$$

备注 n = 0 ~ 3

图 6-4. 间隔定时器模式操作的寄存器设置(1/2)

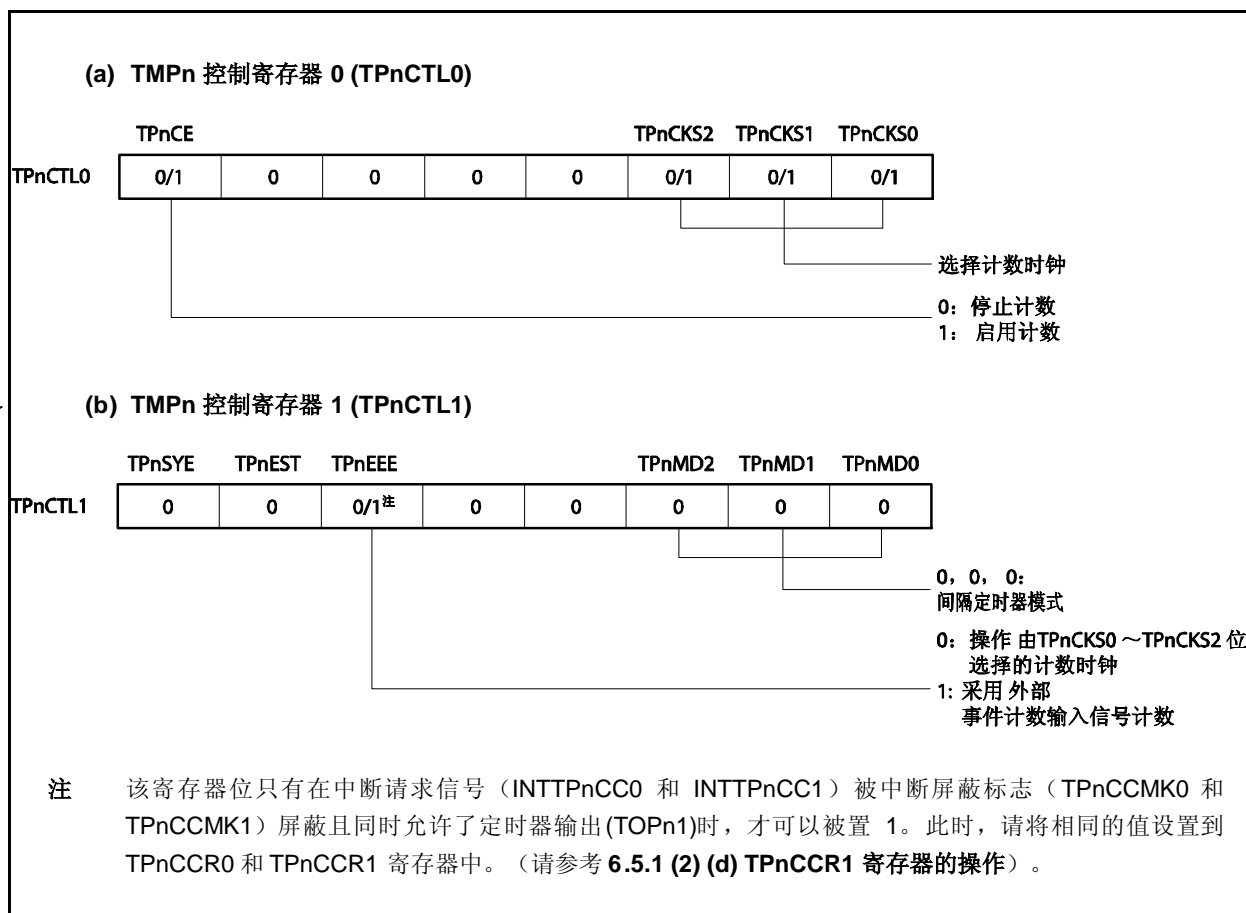
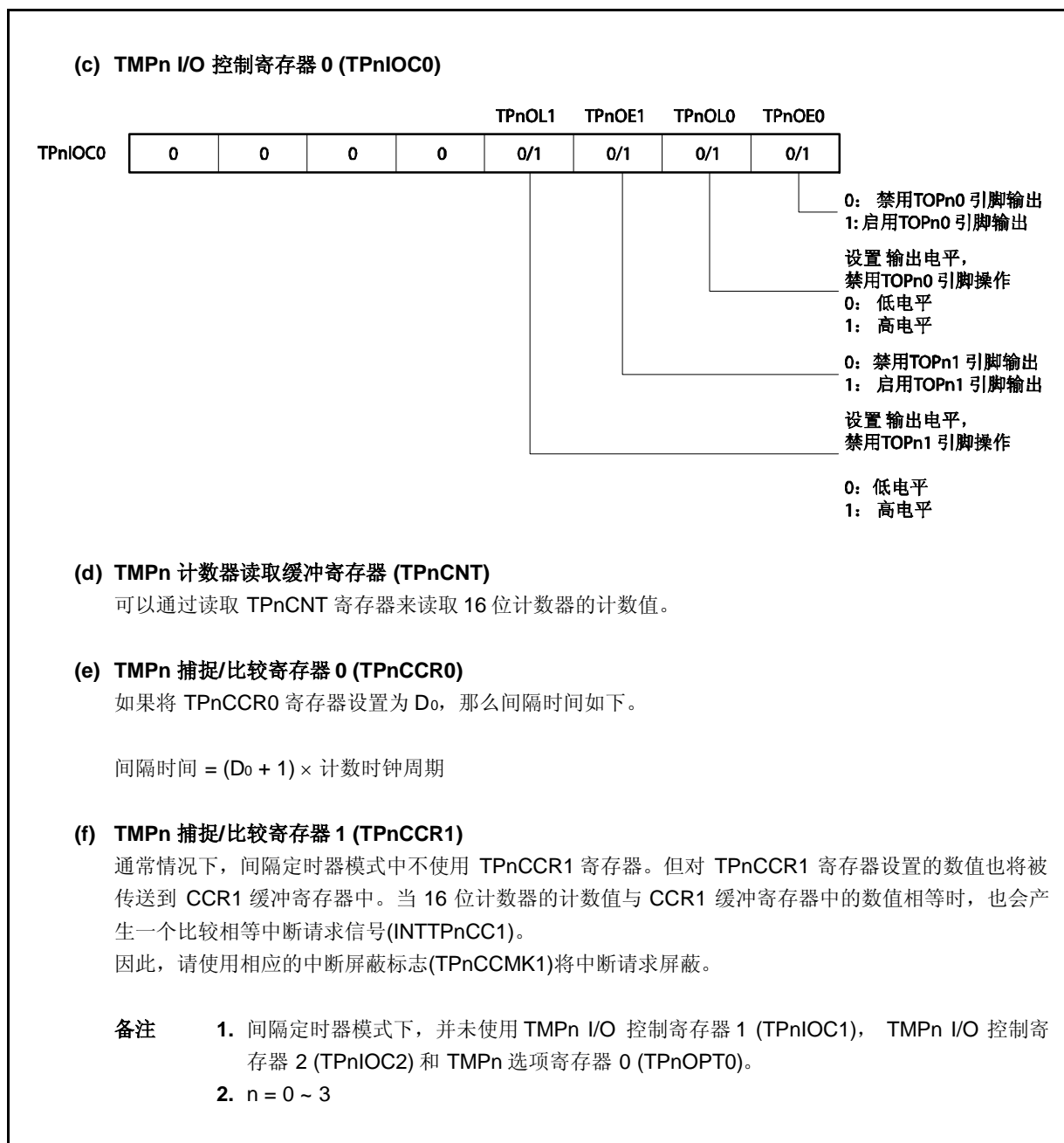
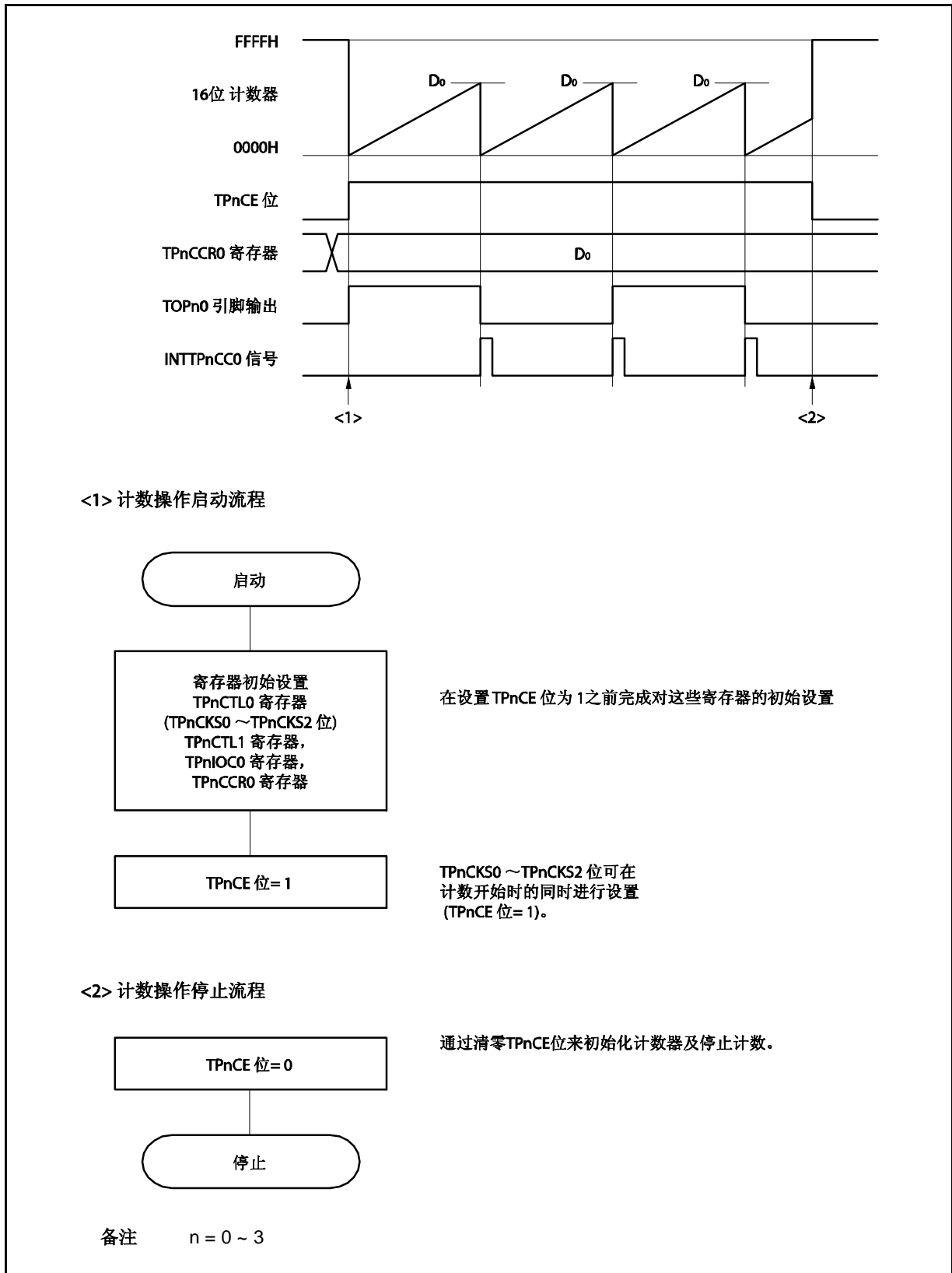


图 6-4. 间隔定时器模式操作的寄存器设置 (2/2)



(1) 间隔定时器模式操作流程

图 6-5. 间隔定时器模式下的软件处理流程

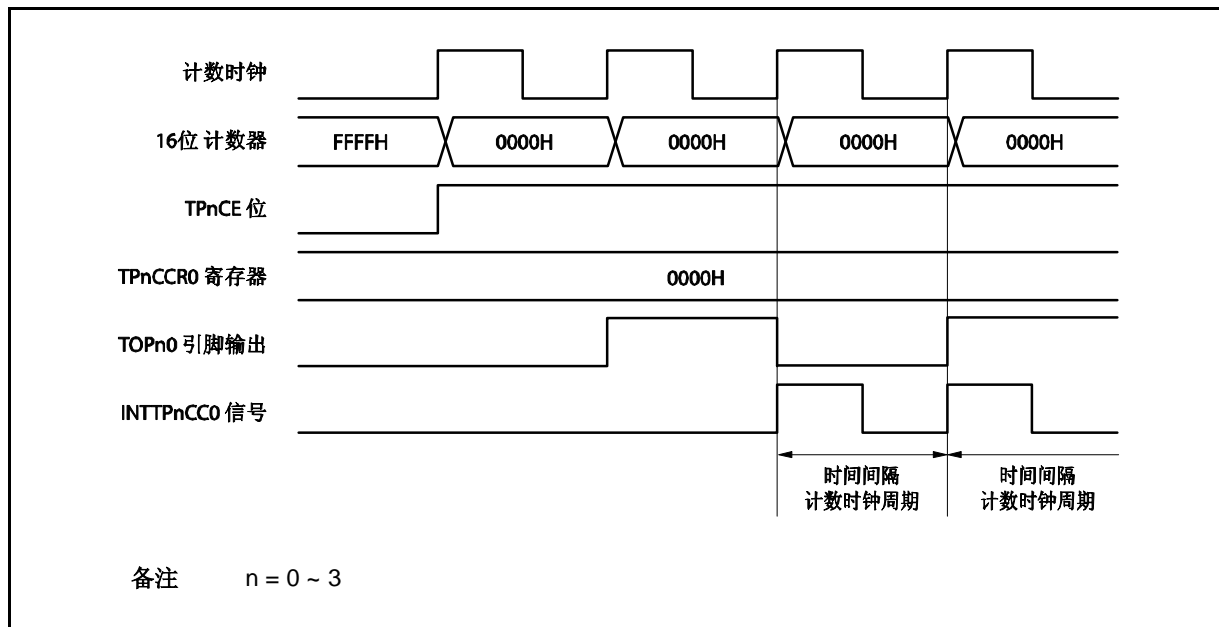


(2) 间隔定时器模式操作时序

(a) TPnCCR0 寄存器设置为 0000H 时的操作

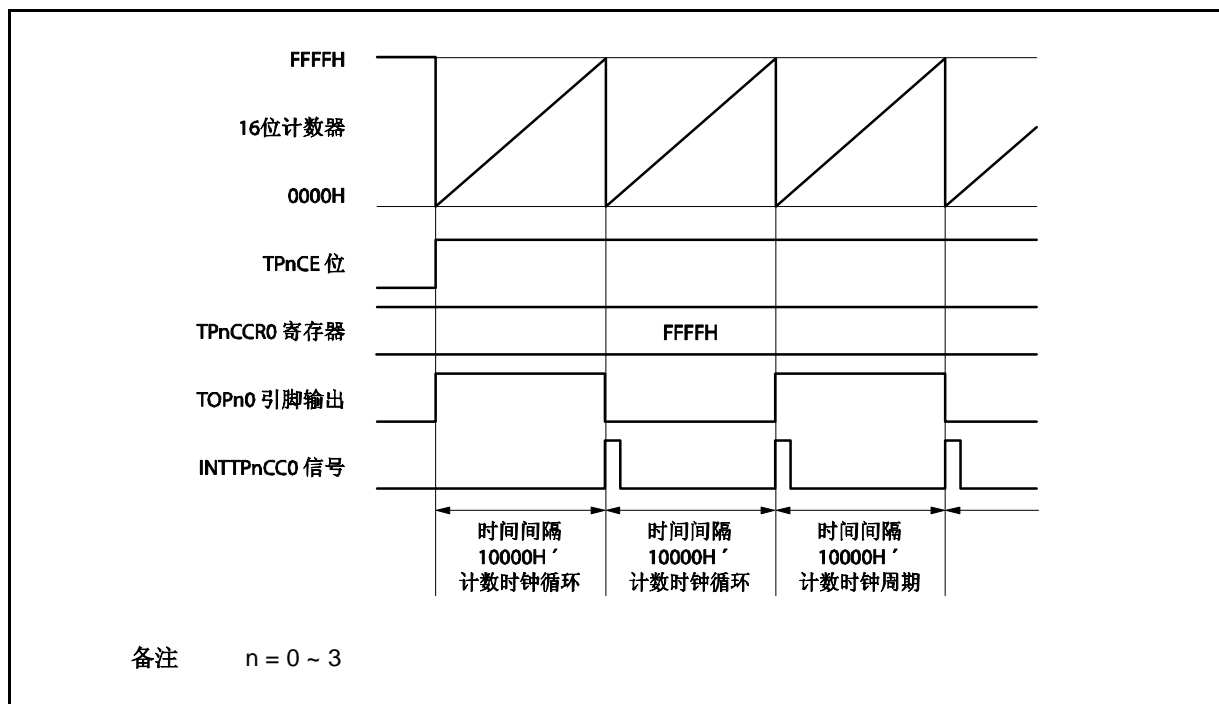
如果 TPnCCR0 寄存器被设置为 0000H，那么在第一个计数时钟之后每当计数时钟来临，都会产生 INTTPnCC0 信号，并且 TOPn0 引脚的输出都会反转。

16 位计数器的计数值始终保持为 0000H。



(b) TPnCCR0 寄存器被设置为 FFFFH 时的操作

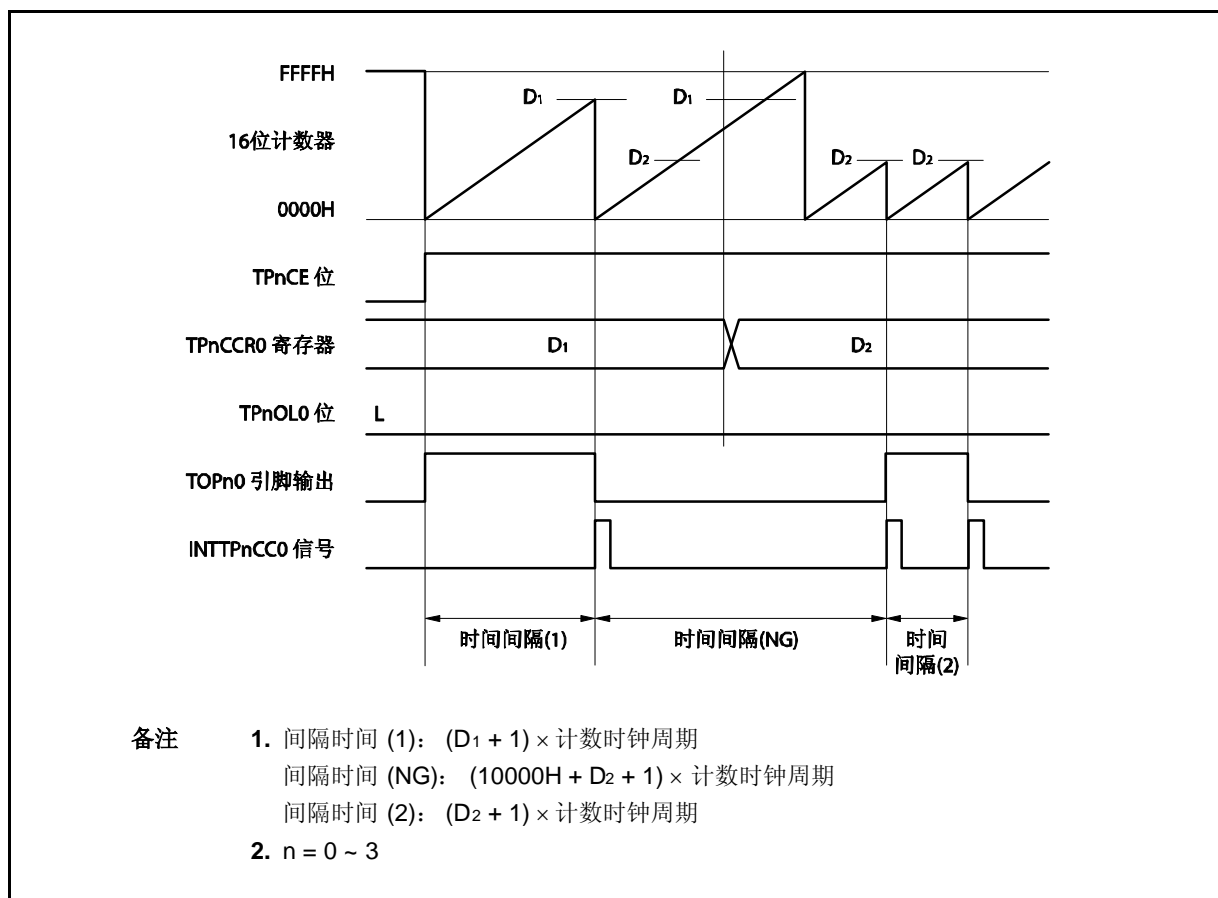
如果 TPnCCR0 寄存器被设置为 FFFFH，那么 16 位计数器将计数到 FFFFH。并在下一个计数时钟时刻同步被清零。同时产生 INTTPnCC0 信号，TOPn0 引脚输出的电平反转。此时，不会产生溢出中断请求信号(INTTPnOV)，溢出标志位(TPnOPT0.TPnOVF 位)也不会被置 1。



(c) 改写 TPnCCR0 寄存器时需要注意的事项

将 TPnCCR0 寄存器的值减小时，要先停止计数，再对设定值进行改变。

如果在计数过程中将 TPnCCR0 寄存器的值减小，那么 16 位计数器可能产生溢出。

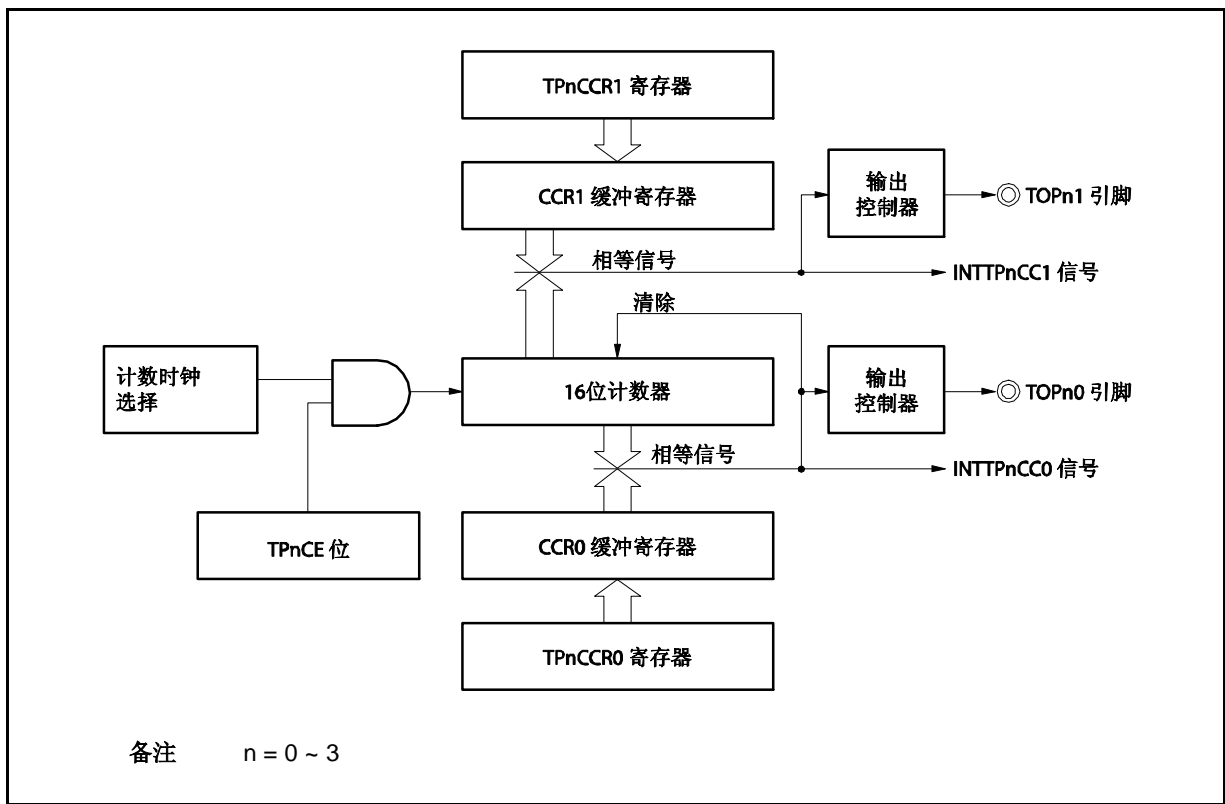


如果当前计数值大于 D_2 但小于 D_1 ，将 TPnCCR0 寄存器的值从 D_1 改变为 D_2 ，新的设定值将在改写后立即从 TPnCCR0 寄存器传送到 CCR0 缓冲寄存器中，那么，16 位计数器的计数值将与 D_2 进行比较。由于计数值已经超过了 D_2 ，因此 16 位计数器会继续计数到 FFFFH 并产生溢出，之后重新从 0000H 开始计数。在重新计数过程中，计数值与 D_2 相等时，才会产生 INTTPnCC0 信号，同时 TOPn0 引脚输出电平反转。

因此可见，INTTPnCC0 信号不会在预期的间隔时间 “ $(D_1 + 1) \times$ 计数时钟周期” 或 “ $(D_2 + 1) \times$ 计数时钟周期” 时产生，而是在一个 “ $(10000H + D_2 + 1) \times$ 计数时钟周期” 的间隔时间产生。

(d) TPnCCR1 寄存器的操作

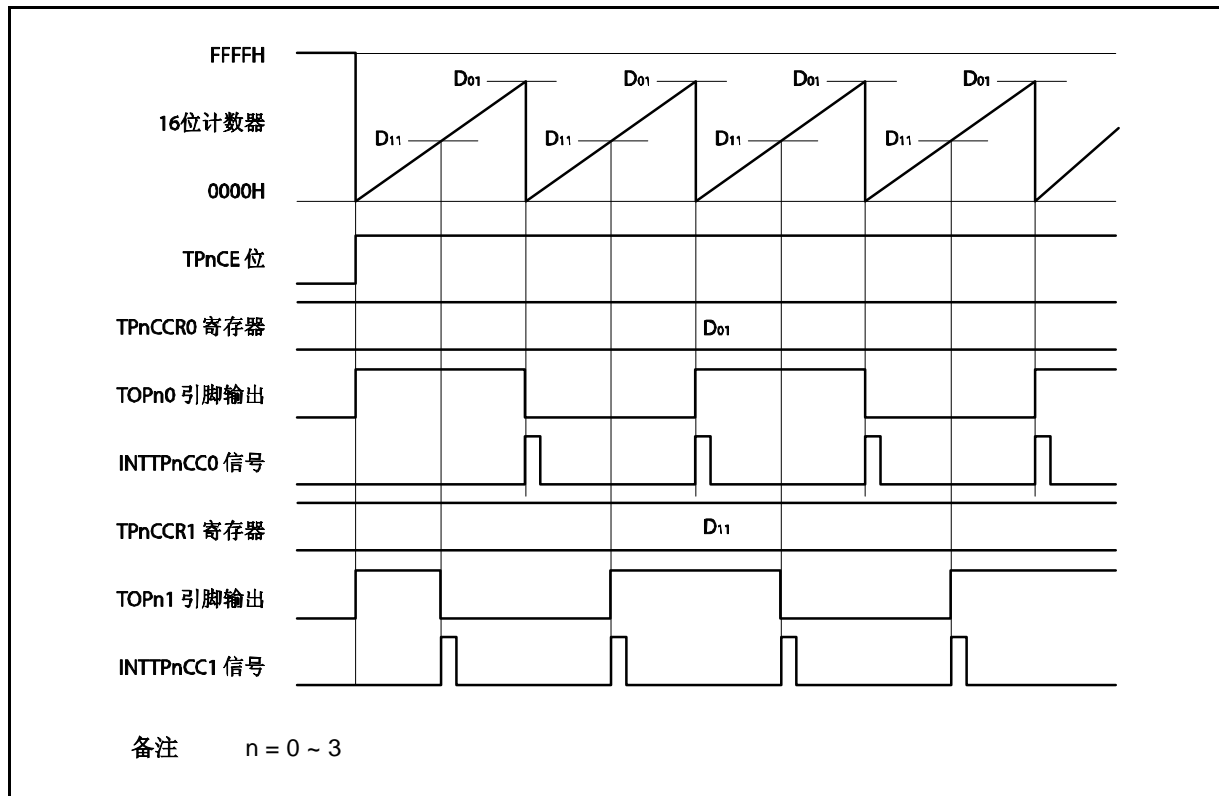
图 6-6. TPnCCR1 寄存器的配置图



如果 TPnCCR1 寄存器的值小于 TPnCCR0 寄存器的值，那么 INTTPnCC1 信号将每周期产生一次。与此同时，TOPn1 引脚的输出电平也会反转。

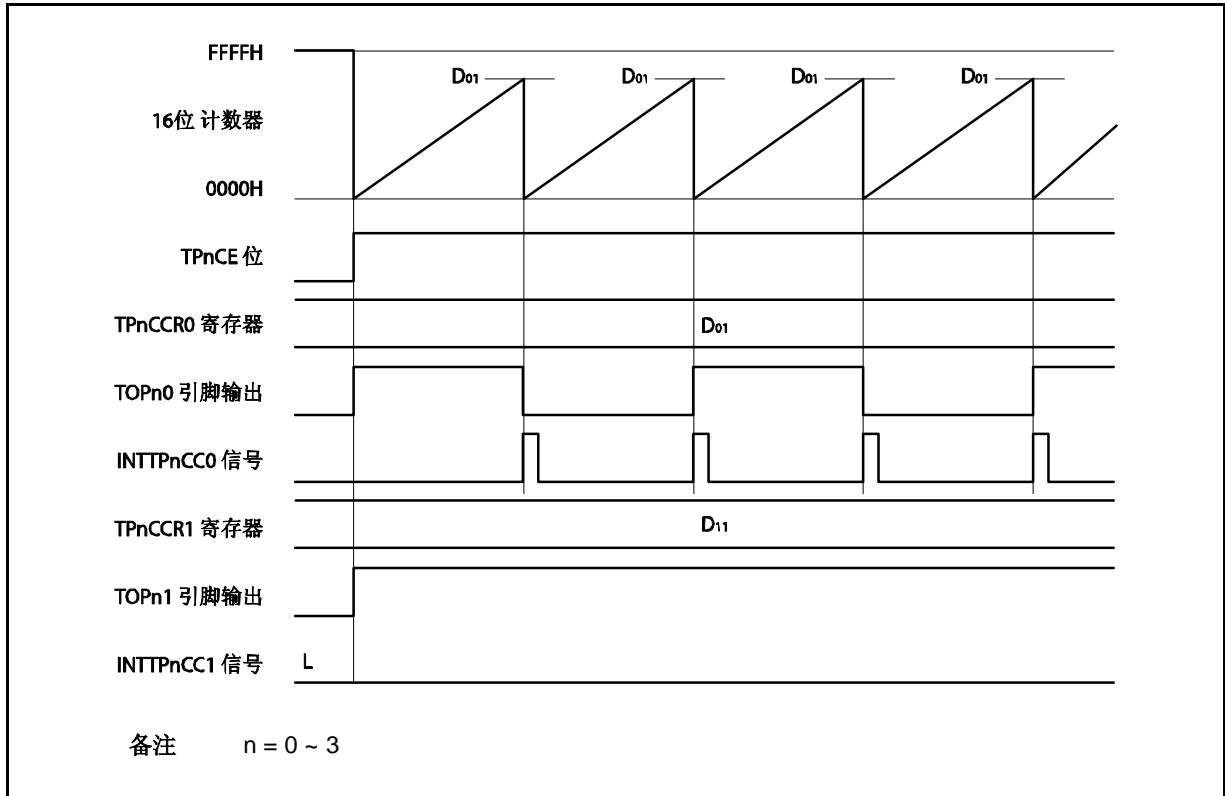
TOPn1 引脚输出的方波周期与 TOPn0 引脚的方波周期相同。

图 6-7. $D_{01} \geq D_{11}$ 时的时序图



如果 TPnCCR1 寄存器的值大于 TPnCCR0 寄存器的值，那么 16 位计数器的计数值不会与 TPnCCR0 寄存器中的值相等。因此，不会产生 INTTPnCC1 信号，TOPn1 引脚的输出电平也不会反转。

图 6-8. $D_{01} < D_{11}$ 时的时序图



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

在外部事件计数模式下, 当 TPnCTL0.TPnCE 位被设置为 1 时, 外部事件的有效沿将被计数, 并在指定数量的有效沿被计数时产生一个中断请求信号(INTTPnCC0)。该模式下不可使用 TOPn0 引脚。

通常情况下, 外部事件计数模式下不使用 TPnCCR1 寄存器。

图 6-9. 外部事件计数模式的配置图

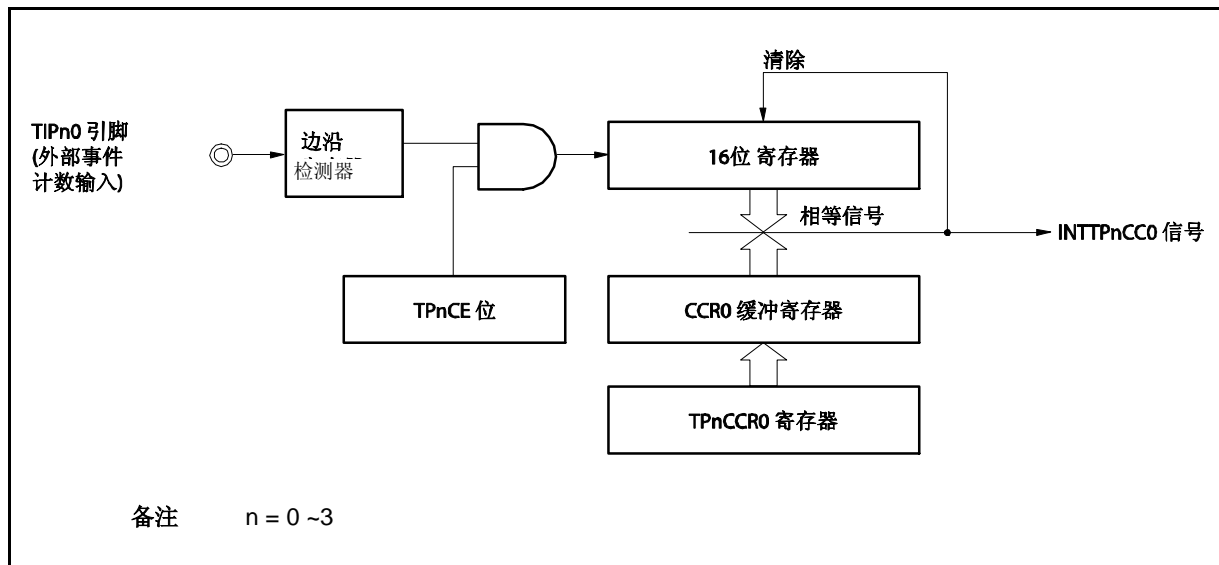
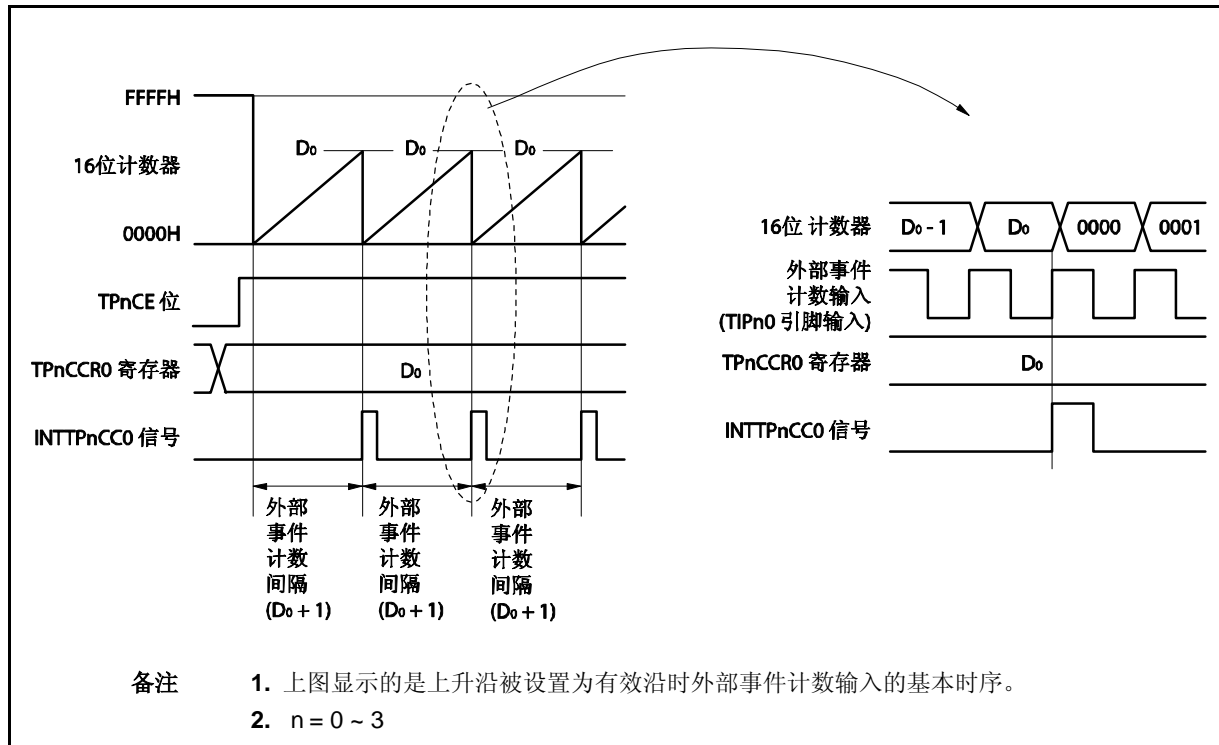


图 6-10. 外部事件计数模式的基本时序



当 TPnCE 位被设置为 1 时，16 位计数器的计数值从 FFFFH 清零为 0000H。每次外部事件计数输入的有效沿被检测时，计数器计数一次。此外，TPnCCR0 寄存器的设定值会被传送到 CCR0 缓冲寄存器中。

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值相等时，16 位计数器将被清零，并且产生一个比较相等中断请求信号(INTTPnCC0)。

每当检测到 (TPnCCR0 寄存器设置值 + 1) 次外部事件计数输入的有效沿时，都会产生 INTTPnCC0 信号。

图 6-11. 外部事件计数模式操作的寄存器设置 (1/2)

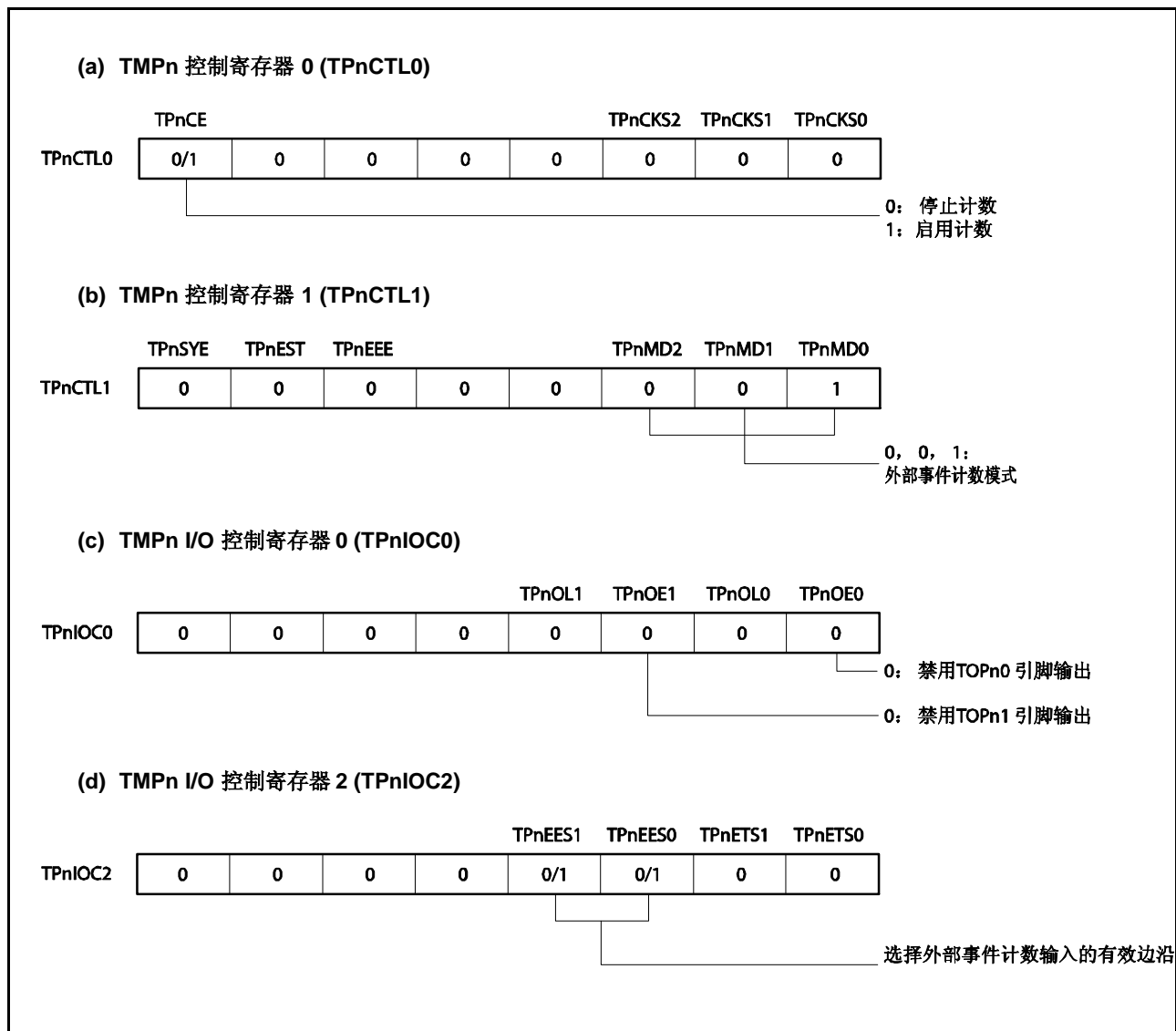


图 6-11. 外部事件计数模式操作的寄存器设置 (2/2)

(e) TMPn 计数器读取缓冲寄存器 (TPnCNT)

可以通过读取 TPnCNT 寄存器来读取 16 位计数器的计数值。

(f) TMPn 捕捉/比较寄存器 0 (TPnCCR0)

如果向 TPnCCR0 寄存器写入 D_0 ，则计数器被清零，并且将在外部事件计数达到 $(D_0 + 1)$ 时产生比较相等中断请求信号 (INTTPnCC0)。

(g) TMPn 捕捉/比较寄存器 1 (TPnCCR1)

通常情况下，外部事件计数模式中不使用 TPnCCR1 寄存器。但对 TPnCCR1 寄存器设置的数值也将被传送到 CCR1 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 缓冲寄存器中的数值相等时，也会产生一个比较相等中断请求信号 (INTTPnCC1)。

因此，请使用相应得中断屏蔽标志 (TPnCCMK1) 将中断请求屏蔽。

注意事项 当外部时钟用于计数时钟时，外部时钟可以仅从 TIPn0 引脚输入。此时，设置 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位为 00。（捕捉触发器输入 (TIPn0 引脚)：无边沿检测）。

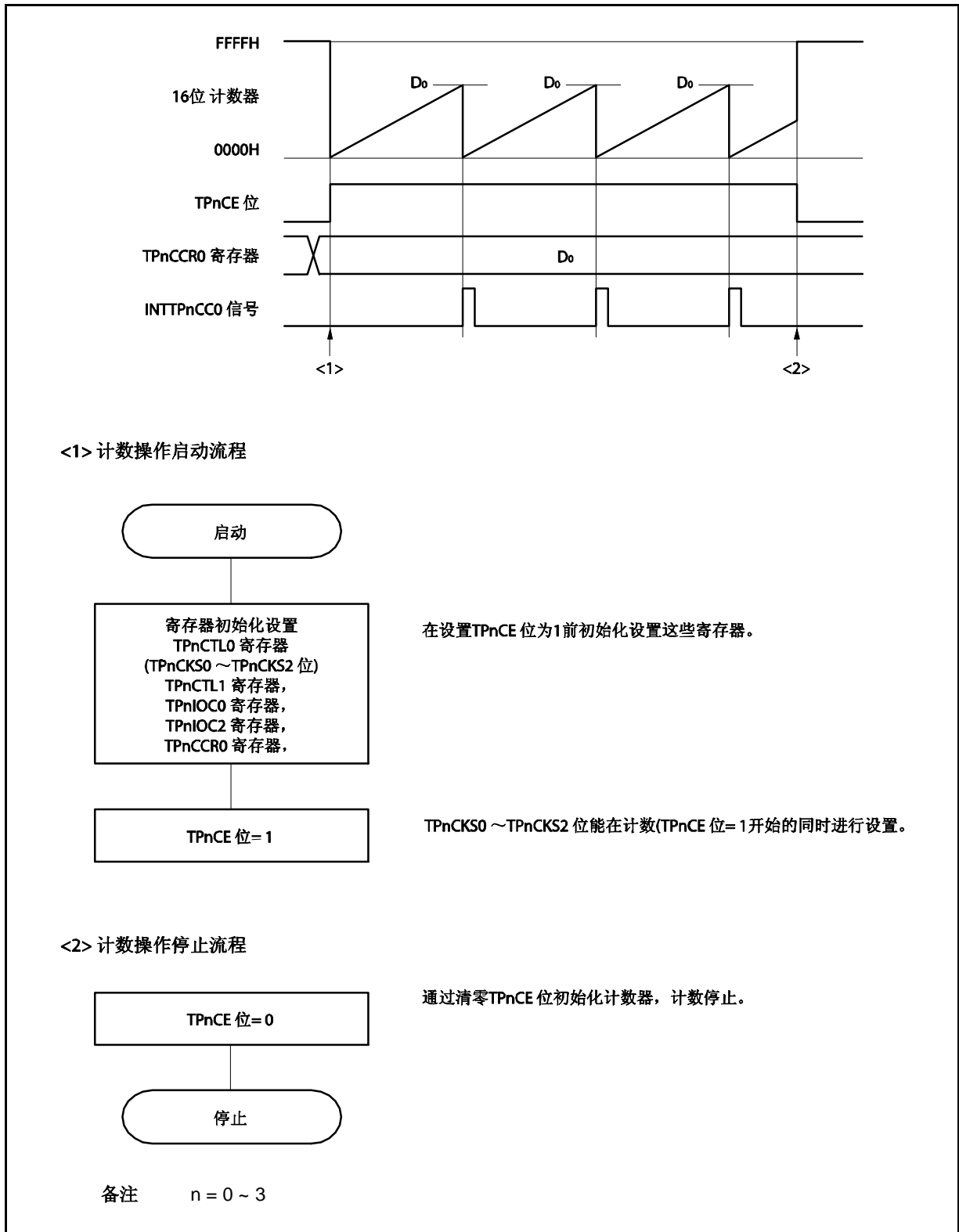
备注 1. 外部事件计数模式下，并未使用 TMPn I/O 控制寄存器 1 (TPnIOC1) 和 TMPn 选项寄存器 0 (TPnOPT0)。

2. $n = 0 \sim 3$

<R>

(1) 外部事件计数模式操作流程

图 6-12. 外部事件计数模式的软件处理流程

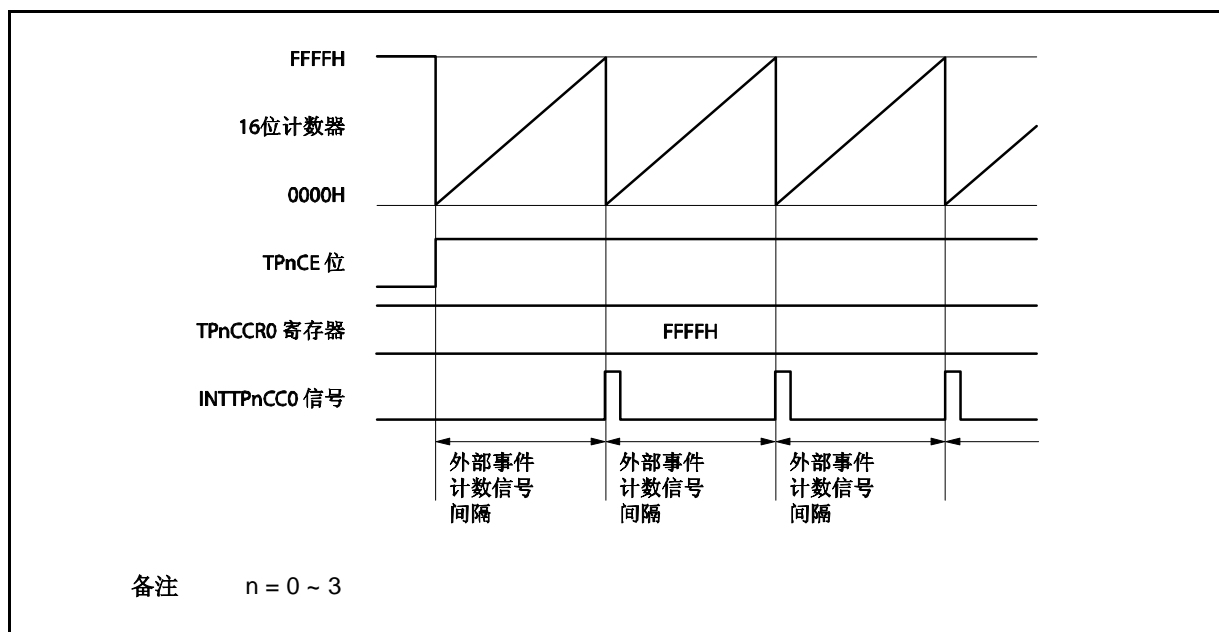


(2) 外部事件计数模式的操作时序

- 注意事项
1. 在外部事件计数模式中，不要将 TPnCCR0 寄存器设置为 0000H。
 2. 在外部事件计数模式中，禁止使用定时器输出功能。如果要将外部事件计数输入作为定时器输出使用，那么请将工作模式设置为间隔定时器模式并将外部事件计数输入设定为计数时钟 (TPnCTL1.TPnMD2 ~ TPnCTL1.TPnMD0 位 = 000, TPnCTL1.TPnEEE 位 = 1)。

(a) TPnCCR0 寄存器被设置为 FFFFH 时的操作

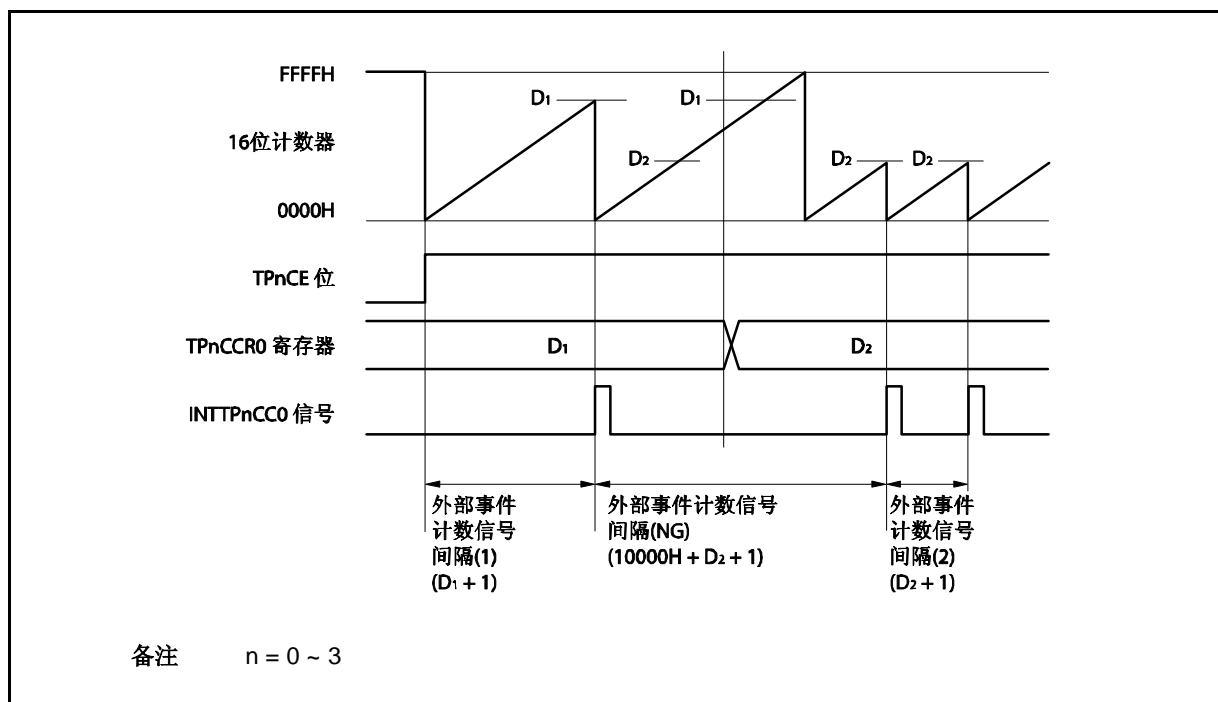
如果 TPnCCR0 寄存器被设置为 FFFFH，那么 16 位计数器将在每次检测到外部事件计数信号的有效沿处计数到 FFFFH。并在下一个计数事件时刻被同步清零，同时产生 INTTPnCC0 信号。此时，溢出标志位 (TPnOPT0.TPnOVF 位) 不会被置 1。



(b) 改写 TPnCCR0 寄存器时需要注意的事项

将 TPnCCR0 寄存器的值减小时，要先停止计数，再对设定值进行改变。

如果在计数过程中将 TPnCCR0 寄存器的值减小，那么 16 位计数器可能产生溢出。



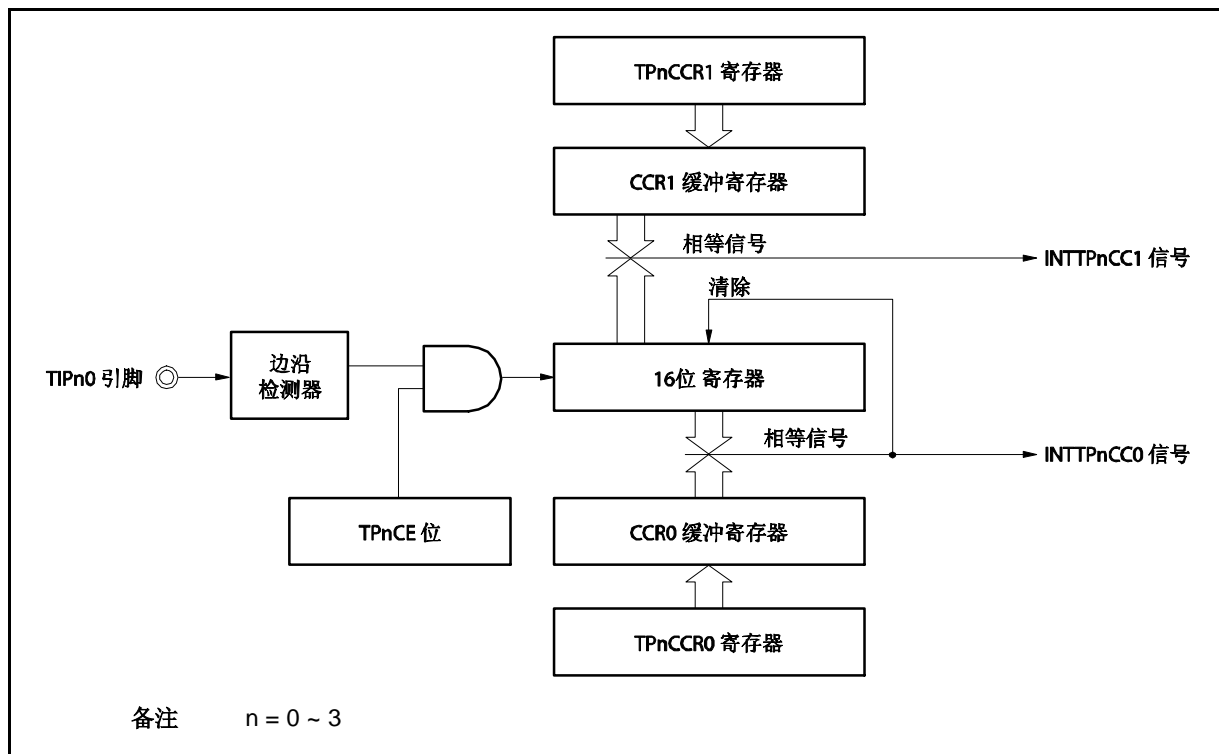
如果当前计数值大于 D₂ 但小于 D₁，将 TPnCCR0 寄存器的值从 D₁ 改变为 D₂，新的设定值将在改写后立即从 TPnCCR0 寄存器传送到 CCR0 缓冲寄存器中，那么，16 位计数器的计数值将与 D₂ 进行比较。

由于计数值已经超过了 D₂，因此 16 位计数器会继续计数到 FFFFH 并产生溢出，之后重新从 0000H 开始计数。在重新计数过程中，计数值与 D₂ 相等时，才会产生 INTTPnCC0 信号。

由此可见，INTTPnCC0 信号不会在预期的事件计数次数“(D₁ + 1) 次”或“(D₂ + 1) 次”时产生，而是在一个“(10000H + D₂ + 1) 次”的计数次数时产生。

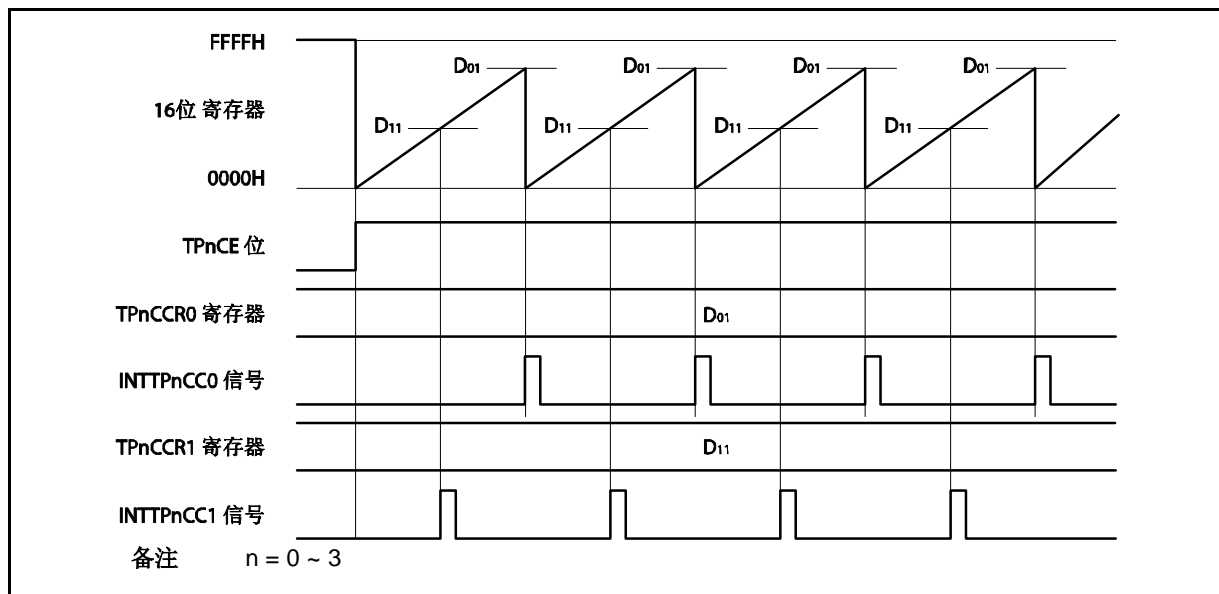
(c) TPnCCR1 寄存器的操作

图 6-13. TPnCCR1 寄存器的配置



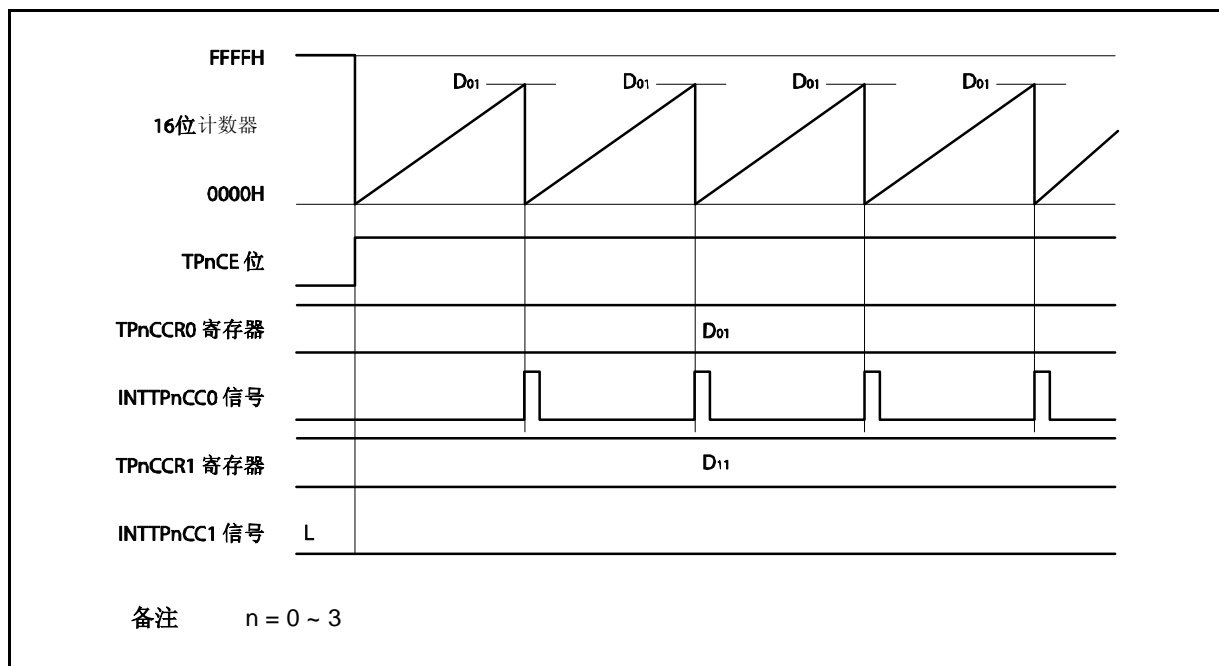
如果 TPnCCR1 寄存器的值小于 TPnCCR0 寄存器的值，那么 INTTPnCC1 信号将每周期产生一次。

图 6-14. $D_{01} \geq D_{11}$ 时的时序图



如果 TPnCCR1 寄存器的值大于 TPnCCR0 寄存器的值，那么 16 位计数器的计数值不会与 TPnCCR0 寄存器中的值相等。因此，不会产生 INTTPnCC1 信号。

图 6-15. $D_{01} < D_{11}$ 时的时序图



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

在外部触发脉冲输出模式中, TPnCTL0.TPnCE 位被置 1 后, 16 位定时器/事件计数器 P 进入等待触发信号的状态。当检测到一个外部触发输入信号的有效沿时, 16 位定时器/事件计数器 P 开始计数, 并由 TOPn1 引脚输出一个 PWM 波形。

也可以使用软件触发代替外部触发来开始脉冲的输出。当使用软件触发时, 还可由 TOPn0 引脚输出一个以 PWM 波形周期的 2 倍为周期的方波。

图 6-16. 外部触发脉冲输出模式的配置

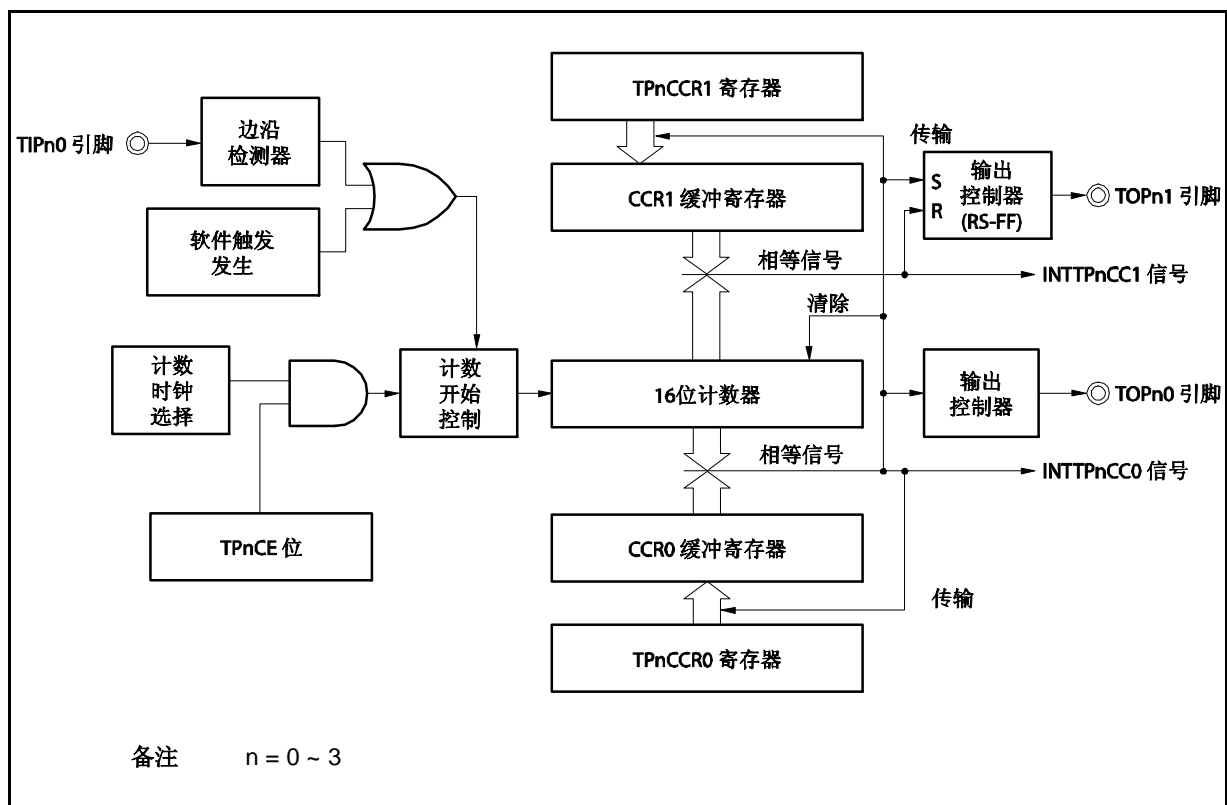
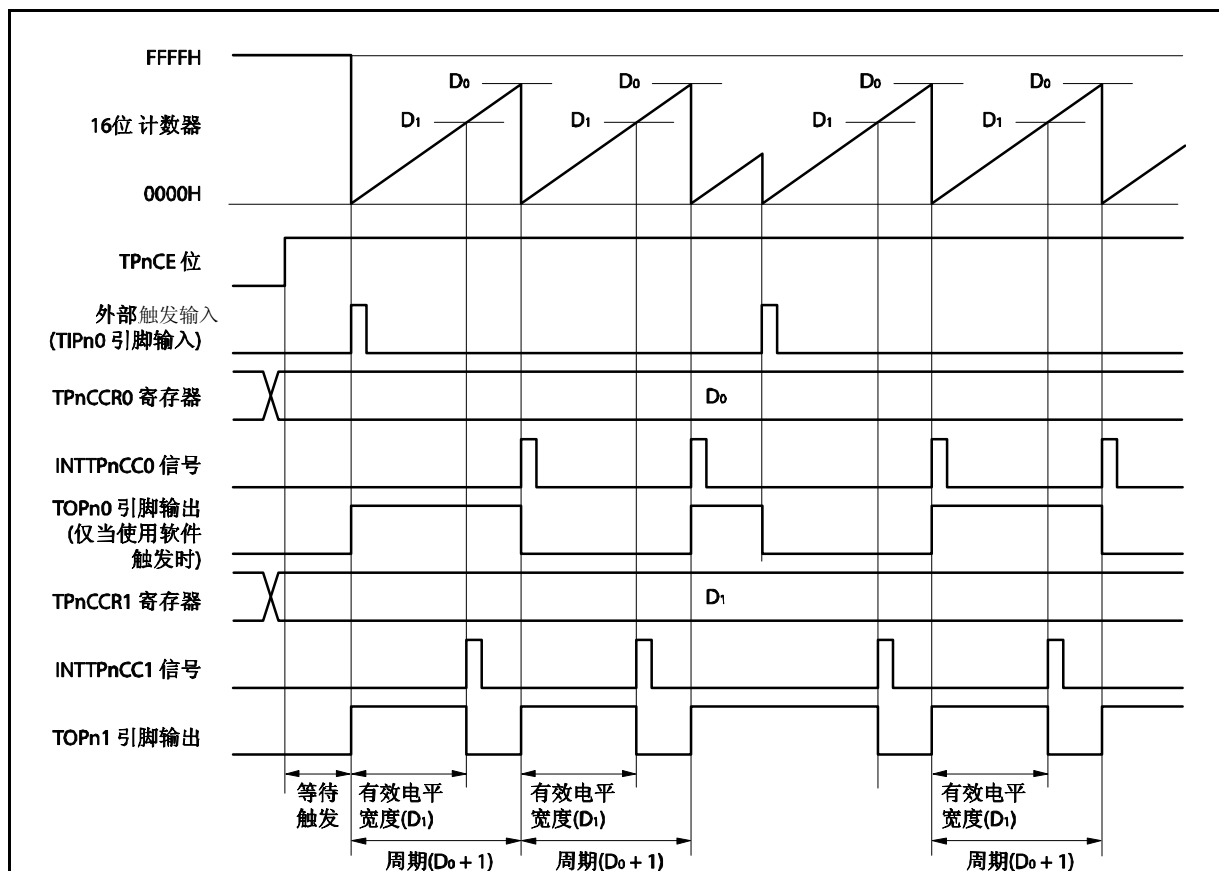


图 6-17. 外部触发脉冲输出模式的基本时序



TPnCE 位被置 1 后，16 位定时器/事件计数器 P 进入等待触发信号的状态。触发产生后，16 位计数器从 FFFFH 清零为 0000H，同时开始计数并从 TOPn1 引脚输出一个 PWM 波形。如果在计数器计数过程中，再次检测到触发信号，那么计数器将被清零并重新开始计数。（TOPn0 引脚的输出电平反转。TOPn1 在触发产生时输出高电平（无论当前状态为高/低）。）

PWM 波形的有效电平宽度，周期以及占空比系数可由以下的算式得出。

$$\text{有效电平宽度} = (\text{TPnCCR1 寄存器的设置值}) \times \text{计数时钟周期}$$

$$\text{周期} = (\text{TPnCCR0 寄存器的值} + 1) \times \text{计数时钟周期}$$

$$\text{占空比系数} = (\text{TPnCCR1 寄存器的值}) / (\text{TPnCCR0 寄存器的值} + 1)$$

比较相等中断请求信号 INTTPnCC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器中的值相等后产生，同时 16 位计数器也会被清零。比较相等中断请求信号 INTTPnCC1 会在 16 位计数器的计数值与 CCR1 缓冲寄存器中的值相等时产生。

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

触发源可以从外部触发输入信号的有效沿或设置软件触发 (TPnCTL1.TPnEST 位 = 1) 的方式中进行选择。

备注 n = 0 ~ 3, m = 0, 1

图 6-18. 外部触发脉冲输出模式的寄存器设置 (1/2)

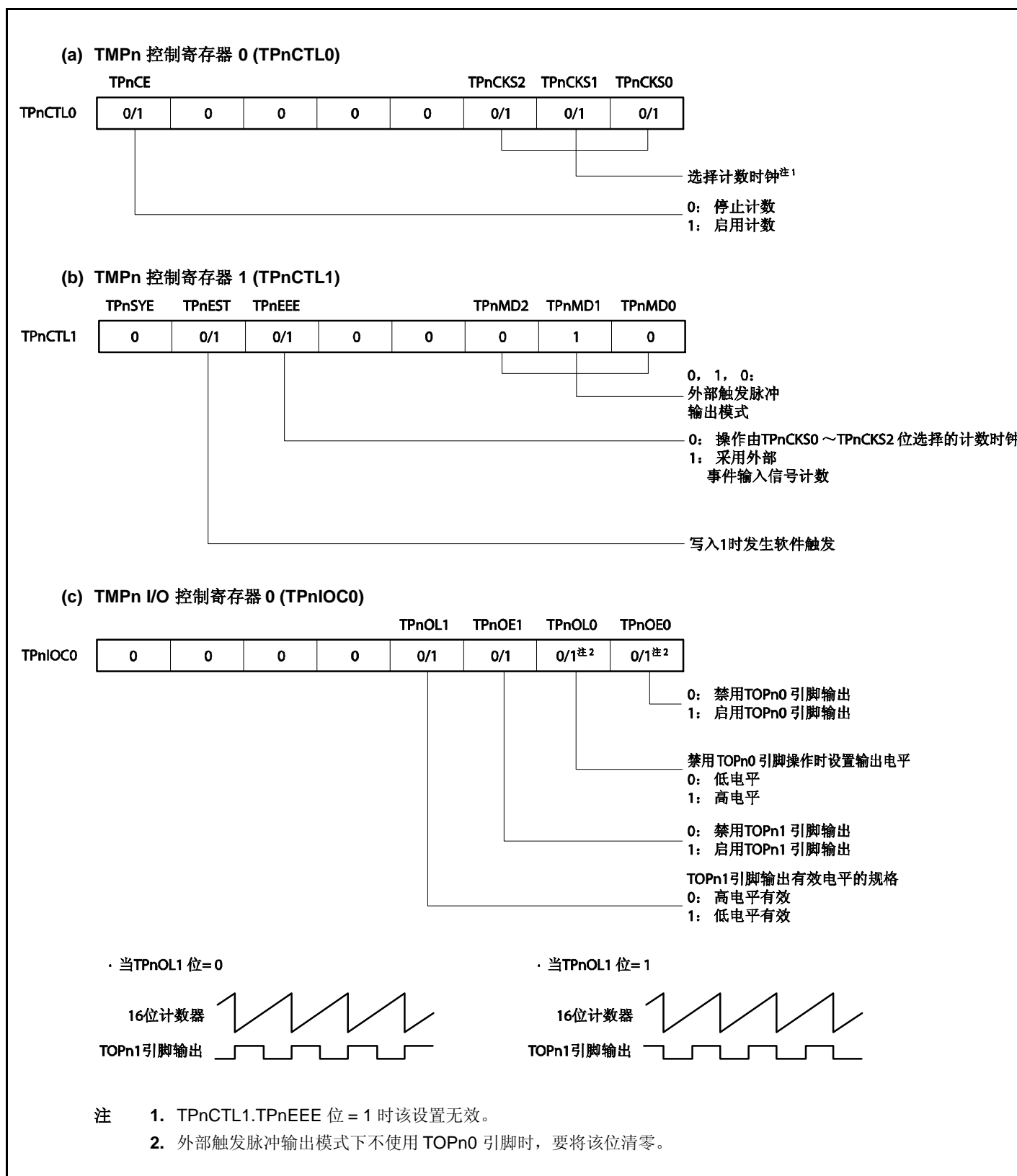
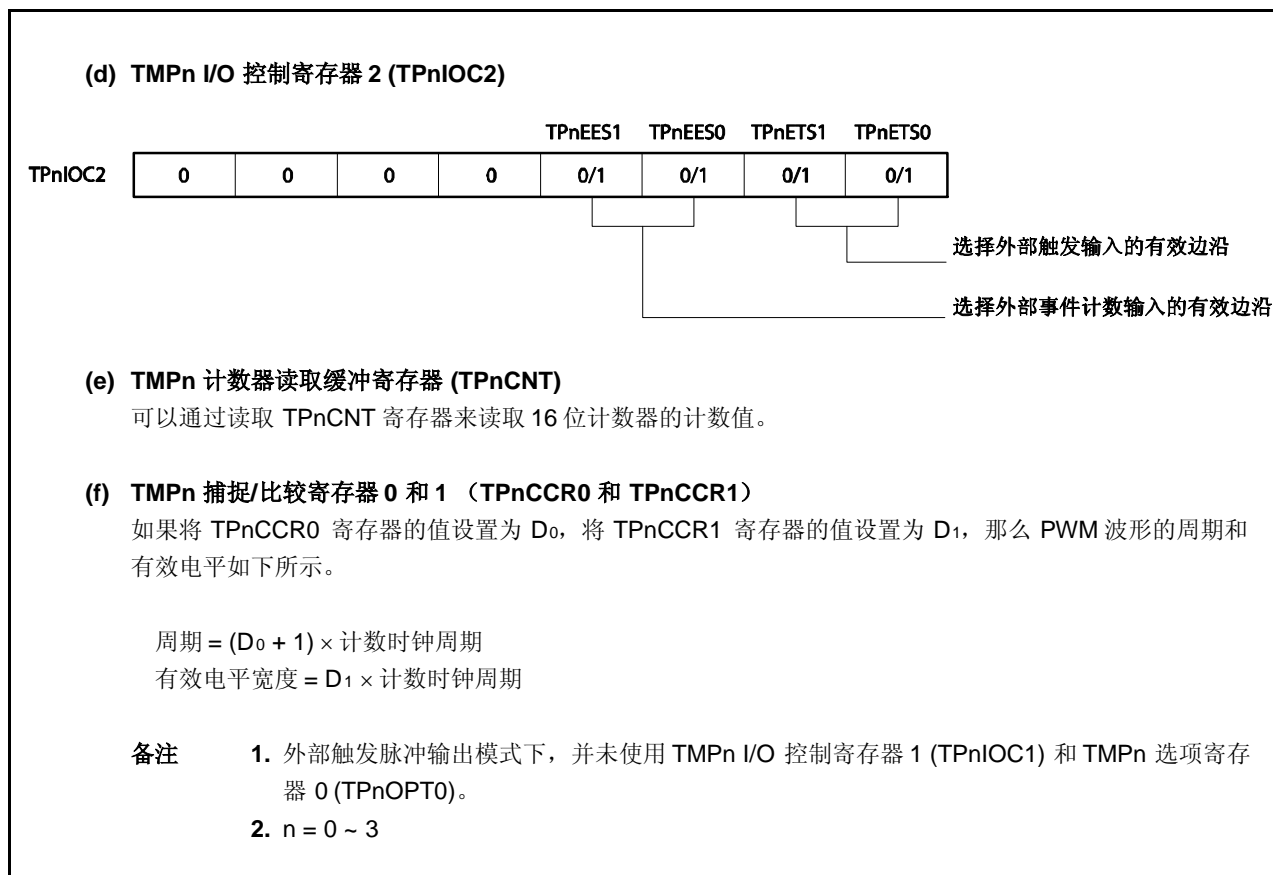


图 6-18. 外部触发脉冲输出模式的寄存器设置 (2/2)



(1) 外部触发脉冲输出模式的操作流程

图 6-19. 外部触发脉冲输出模式的软件处理流程 (1/2)

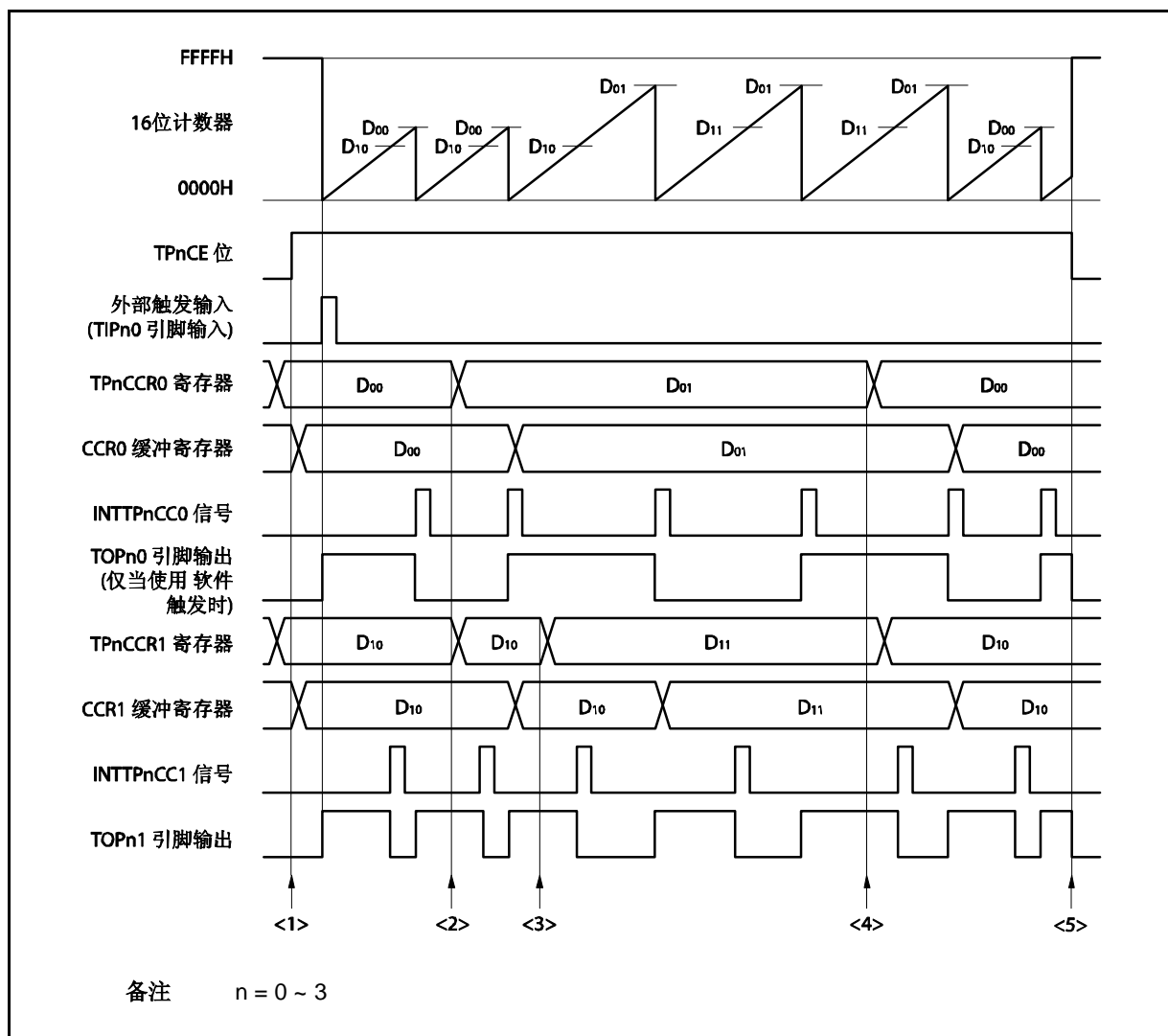
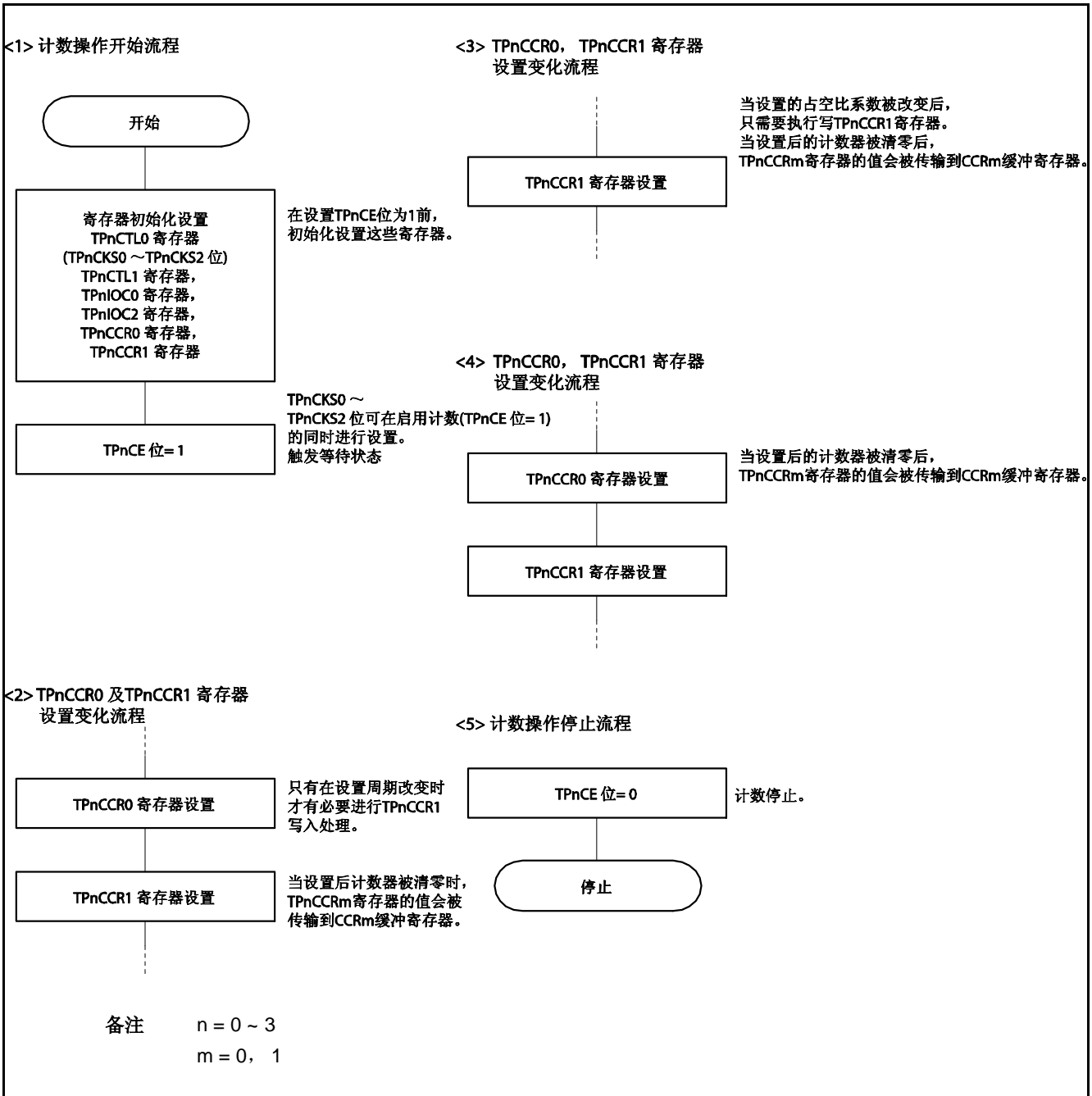


图 6-19. 外部触发脉冲输出模式的软件处理流程 (2/2)

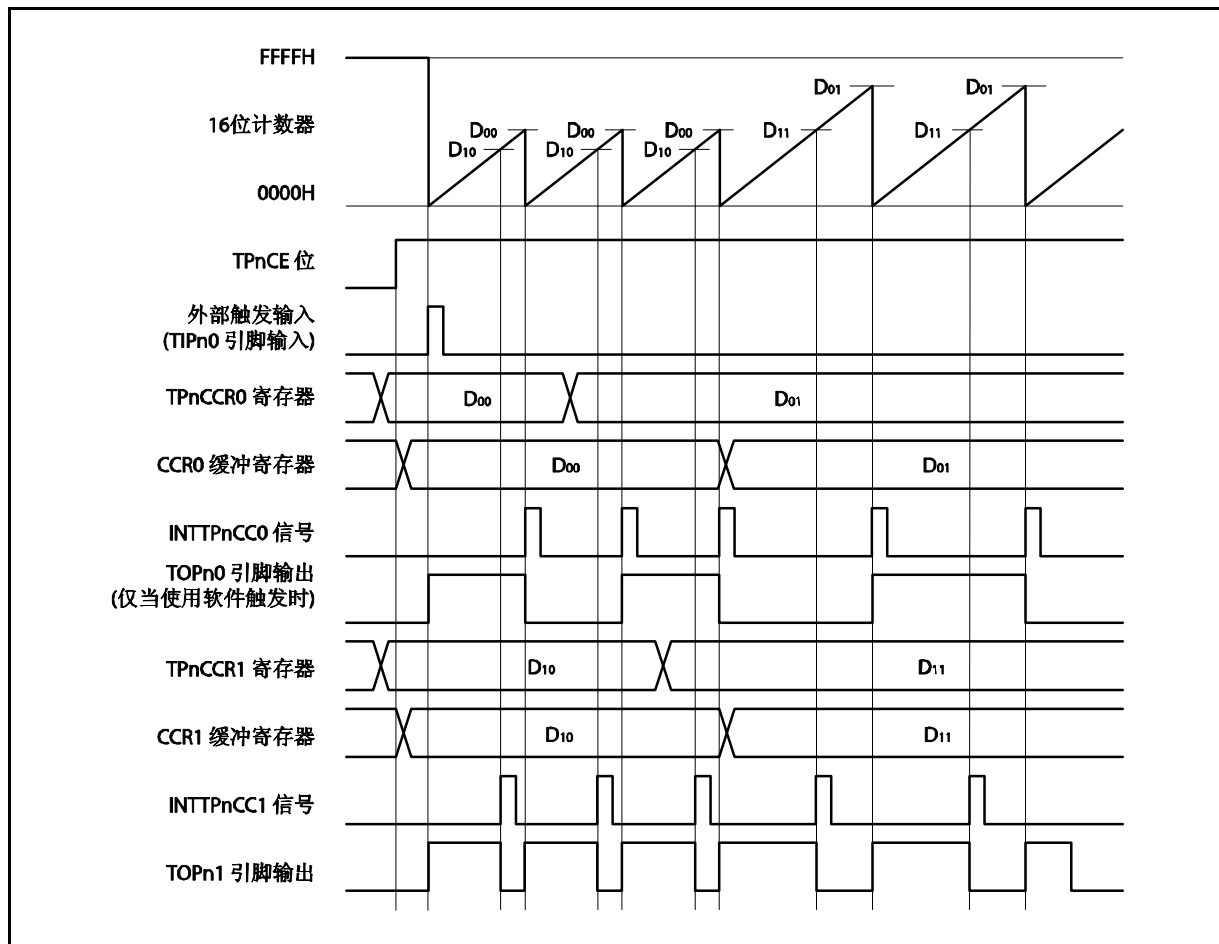


(2) 外部触发脉冲输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

计数器计数过程中，若改变 PWM 波形，则要在最后进行对 TPnCCR1 寄存器的写入操作。

对 TPnCCR1 寄存器进行写入操作之后，若需再次更改 TPnCCRm 寄存器的值，须等待下一个 INTTPnCC0 信号被检测到后，再进行相关操作。



若要将 TPnCCRm 寄存器中设置的数据传送至 CCRm 缓冲寄存器，必须对 TPnCCR1 寄存器进行写入操作。

同时改变 PWM 波形的周期和有效电平时，先将周期值写入 TPnCCR0 寄存器，再将有效电平宽度值写入 TPnCCR1 寄存器。

只改变 PWM 波形的周期时，先将周期值写入 TPnCCR0 寄存器，再向 TPnCCR1 寄存器写入与原来相同的值。

只改变 PWM 波形的有效电平宽度（占空比系数）时，只需重新设置 TPnCCR1 寄存器的值即可。

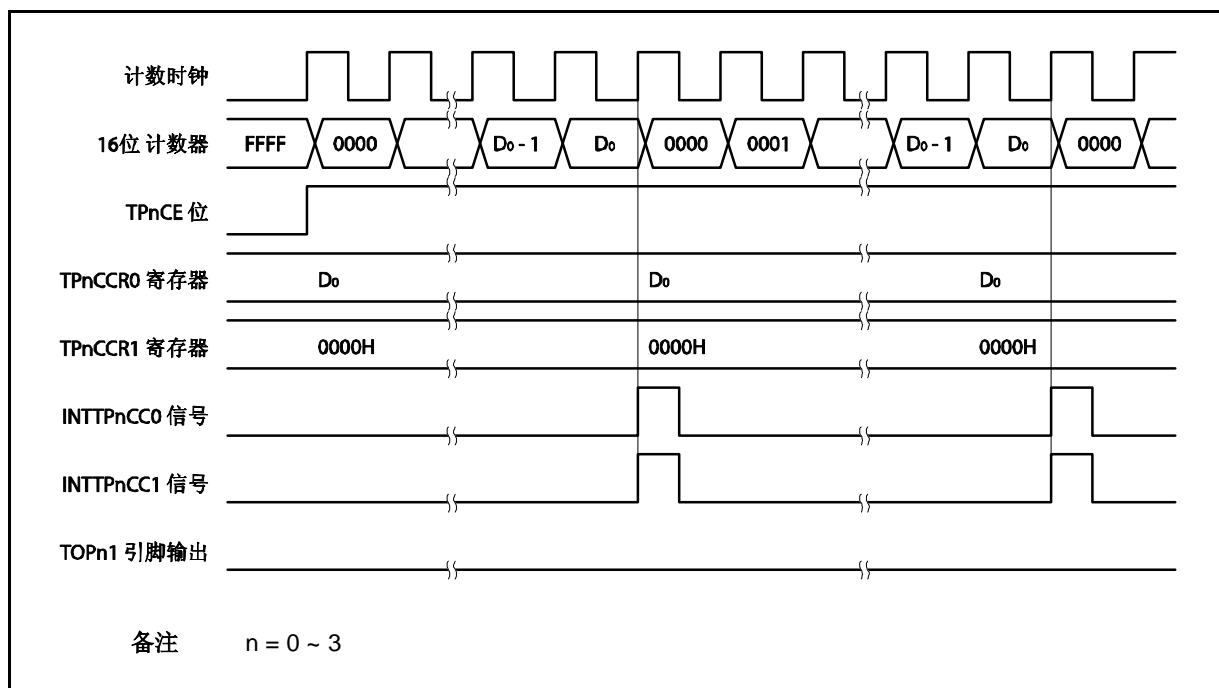
数据被写入 TPnCCR1 寄存器之后，当 16 位计数器被清零时，TPnCCRm 寄存器中的数值才会被传送至 CCRm 缓冲寄存器中。此时 CCRm 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

向 TPnCCR1 寄存器的写入操作完成后，若需再次改写 TPnCCR0 或 TPnCCR1 寄存器的值，则要在产生 INTTPnCC0 信号之后再改写。否则，由于数据从 TPnCCRm 寄存器传送至 CCRm 缓冲寄存器的时序与 TPnCCRm 寄存器的写入时序相冲突，CCRm 缓冲寄存器中所设置的数值可能无法确定。

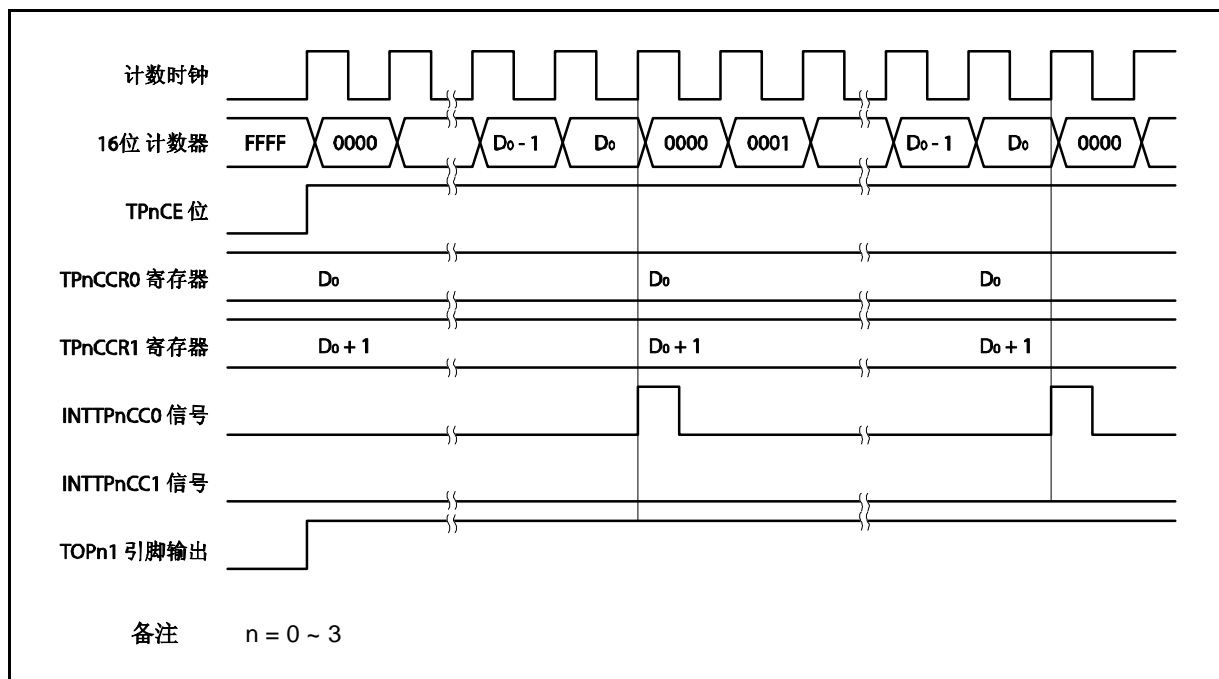
备注 n = 0 ~ 3
 m = 0, 1

(b) PWM 波形的 0%/100% 输出

若要输出 0% 的波形，则需将 TPnCCR1 寄存器设置为 0000H。如果 TPnCCR0 寄存器被设置为 FFFFH，那么 INTTPnCC1 信号将周期性产生。

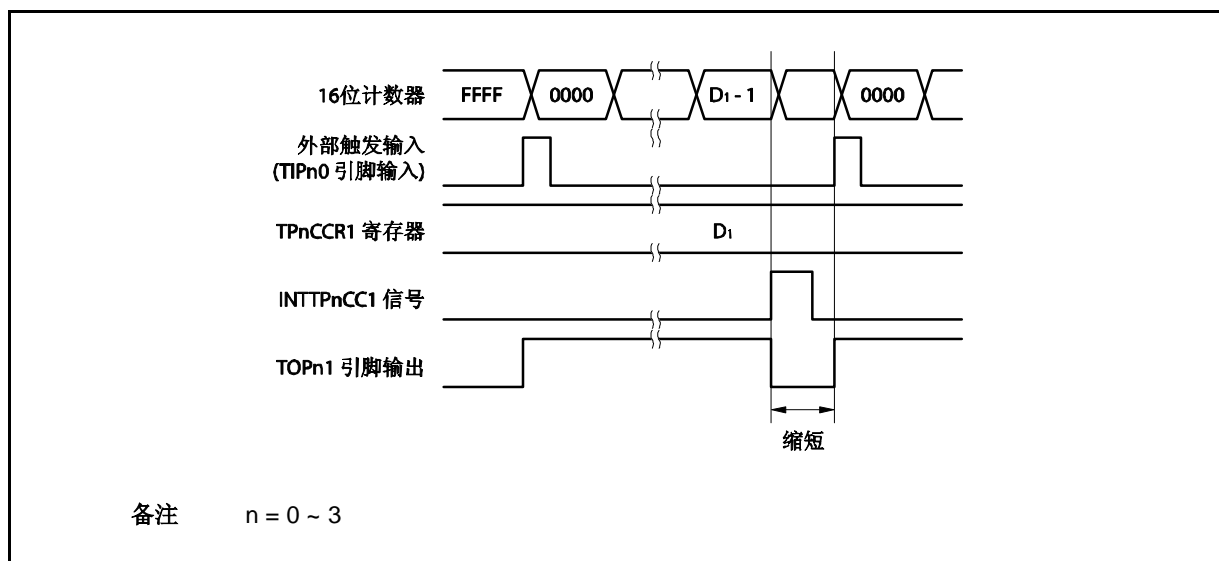


若要输出 100% 波形，则需将 TPnCCR1 寄存器设置为 (TPnCCR0 寄存器值 + 1)。TPnCCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。

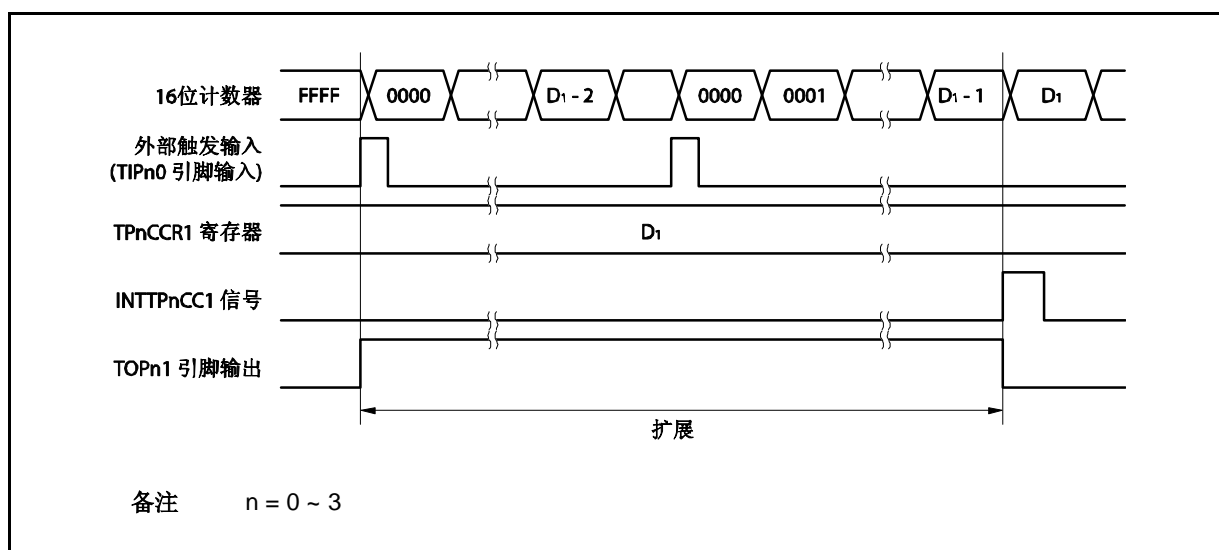


(c) 触发检测和 TPnCCR1 寄存器相等之间的冲突

如果触发信号紧随 INTTPnCC1 中断信号的产生被检测，那么 16 位计数器将被立即清零并重新开始计数，TOPn1 引脚输出也会改变为有效电平。这种情况下，PWM 波形的非有效电平时间就会被缩短。

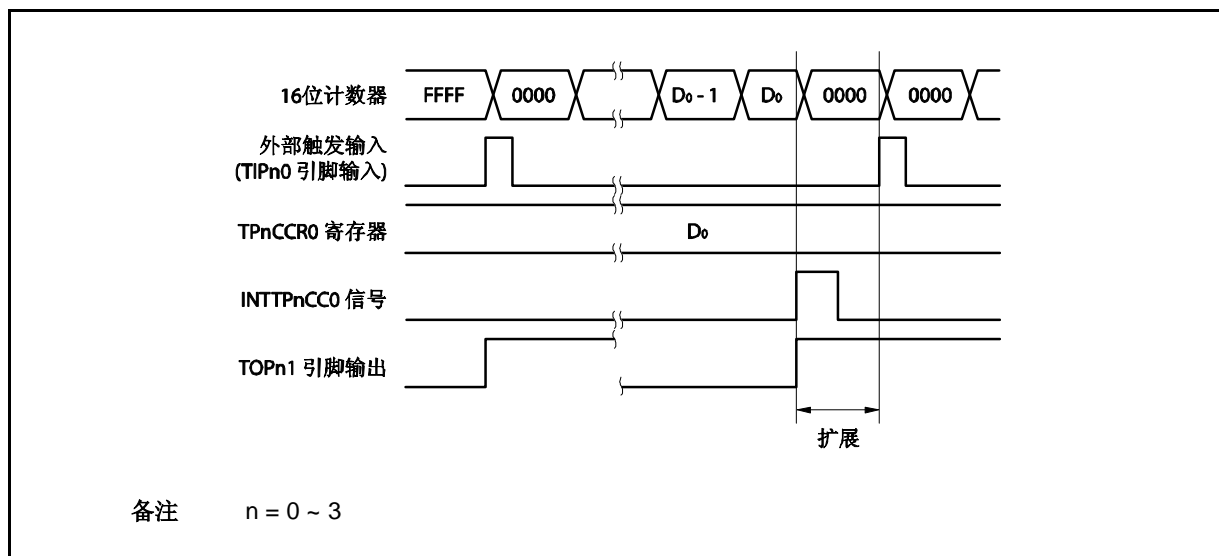


如果在 INTTPnCC1 信号产生之前检测到触发信号，那么 INTTPnCC1 信号将不会产生，此时，16 位计数器被清零并重新开始计数。TOPn1 引脚的输出也会继续保持为有效电平。这种情况下，PWM 波形的有效电平时间就会被延长。

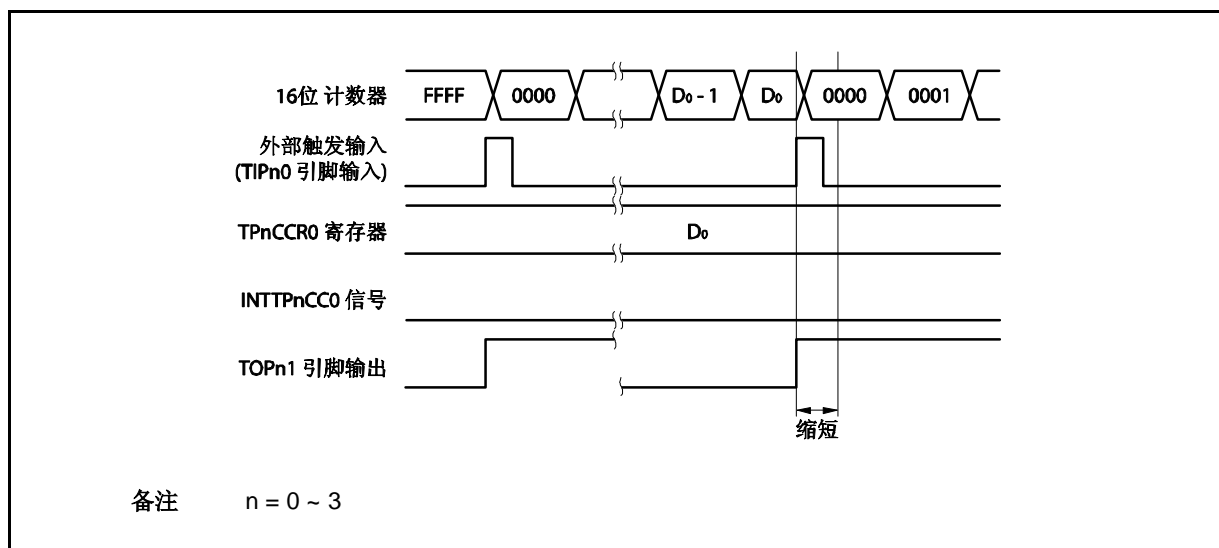


(d) 触发检测和 TPnCCR0 寄存器相等之间的冲突

如果触发信号紧随 INTTPnCC0 中断信号的产生被检测，那么 16 位计数器将被清零并重新开始计数。因此，TOPn1 引脚输出的有效电平时间就会被延长（从 INTTPnCC0 信号产生到触发信号被检测的时间）。

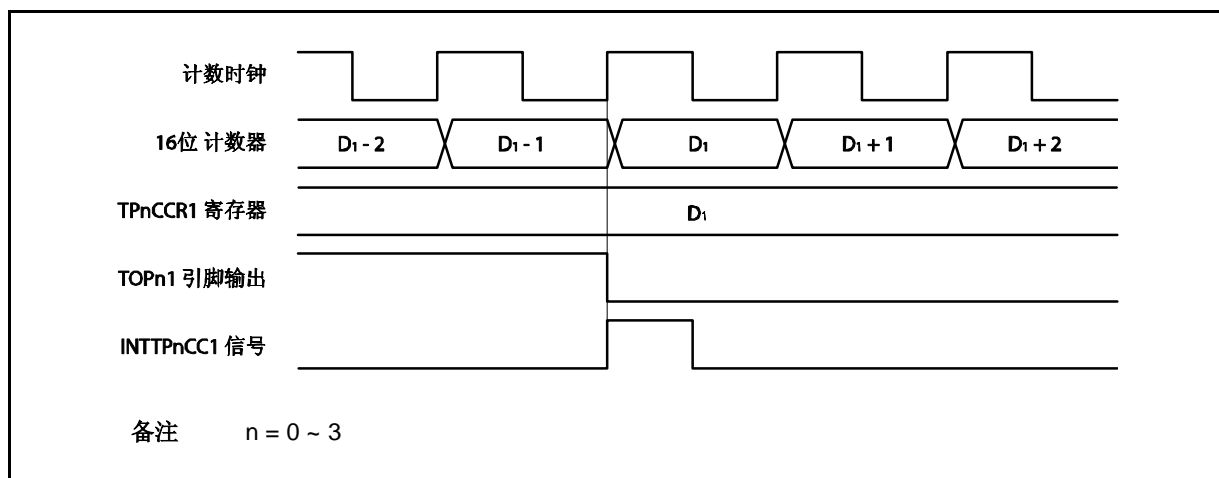


如果在 INTTPnCC0 信号产生之前检测到触发信号，那么 INTTPnCC0 信号将不会产生。16 位计数器将被清零并重新开始计数，TOPn1 引脚输出也会改变为有效电平。这种情况下，PWM 波形的非有效电平时间就会被缩短。



(e) 比较相等中断请求信号的产生时序 (INTTPnCC1)

外部触发脉冲输出模式下，INTTPnCC1 信号的产生时序与其它模式下 INTTPnCC1 信号的产生时序有所不同；INTTPnCC1 信号将在 16 位计数器的计数值与 TPnCCR1 寄存器的值相等时产生。



通常情况下，INTTPnCC1 信号会在 16 位计数器计数值与 TPnCCR1 寄存器的值相等之后的下一个计数时钟同步产生。

但在外部触发脉冲输出模式下，INTTPnCC1 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOPn1 引脚输出信号变化的时序而改变的。

6.5.4 单脉冲输出模式 (TPnMD2 ~ TPnMD0 位 = 011)

在单脉冲输出模式中, TPnCTL0.TPnCE 位被置 1 后, 16 位定时器/事件计数器 P 进入等待触发信号的状态。当检测到一个外部触发输入信号的有效沿时, 16 位定时器/事件计数器 P 开始计数, 并由 TOPn1 引脚输出一个单脉冲波形。

也可以使用软件触发代替外部触发来开始脉冲的输出。当使用软件触发时, TOPn0 引脚会在 16 位计数器计数时输出有效电平, 在计数器停止计数时 (等待触发时) 输出非有效电平。

图 6-20. 单脉冲输出模式的配置图

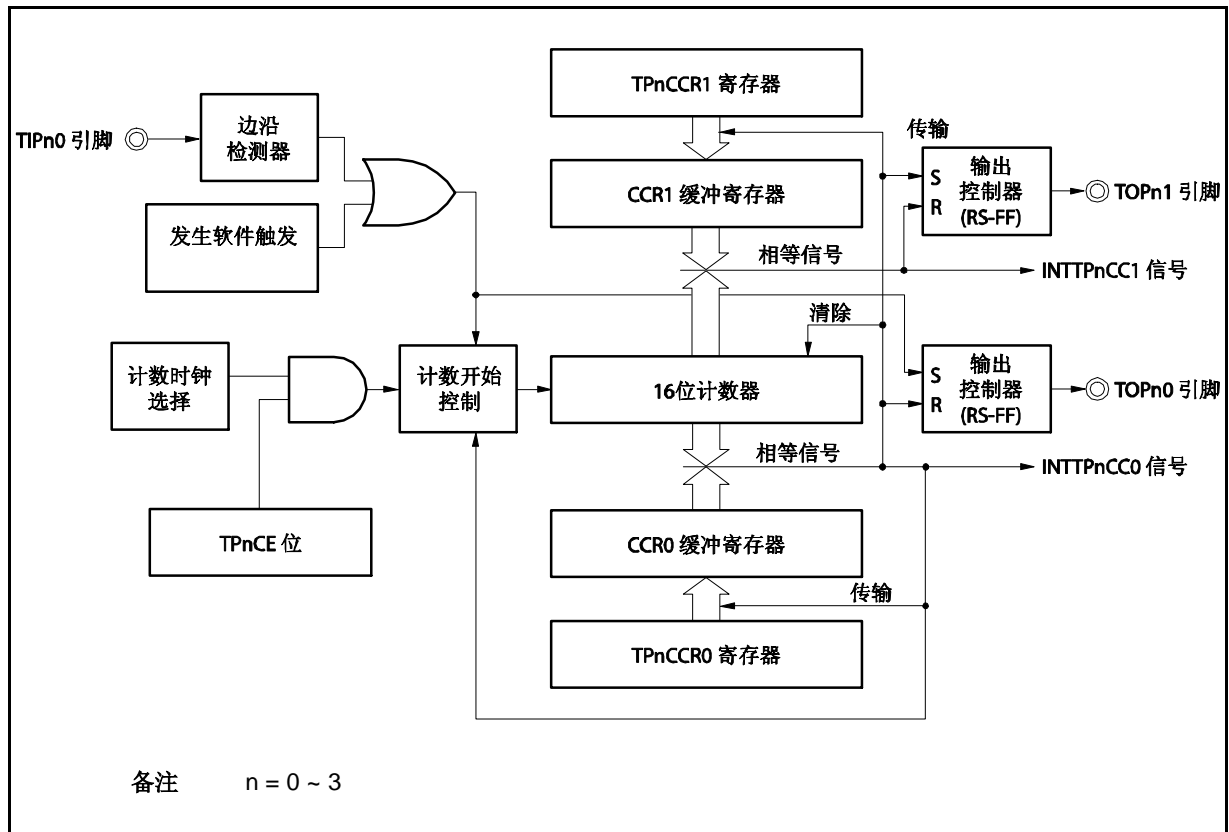
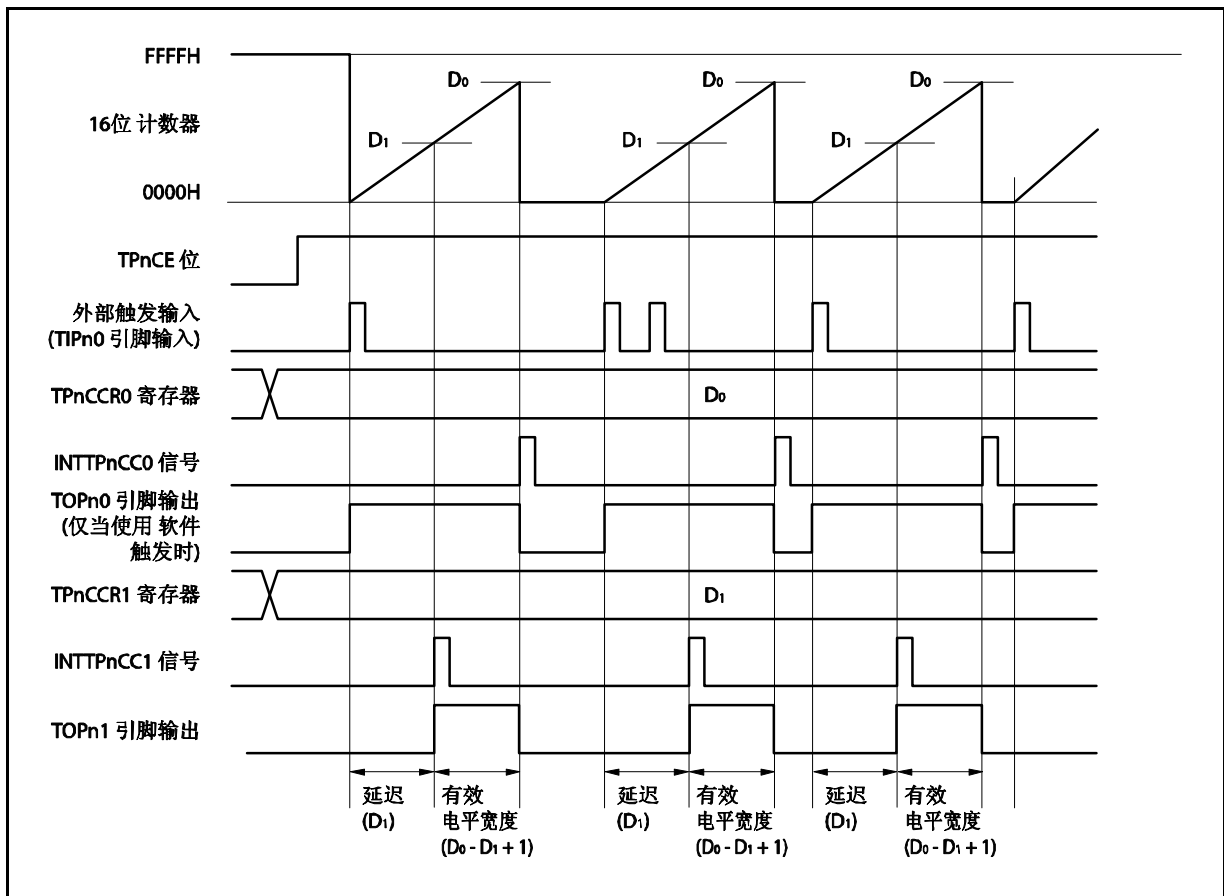


图 6-21. 单脉冲输出模式的基本时序



TPnCE 位被置 1 后，16 位定时器/事件计数器 P 进入等待触发信号的状态。触发被检测后，16 位计数器从 FFFFH 清零为 0000H，同时开始计数并从 TOPn1 引脚输出一个单脉冲波形。单脉冲波形输出后，16 位计数器将被重置为 FFFFH 并停止计数，16 位定时器/时间计数器 P 重新进入等待触发信号的状态。如果在单脉冲波形输出过程中再次检测到触发信号，那么这个触发信号将被忽略。

单脉冲波形的输出延迟时间以及有效电平宽度可由以下算式得出。

输出延迟时间 = (TPnCCR1 寄存器的设置值) × 计数时钟周期

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

比较相等中断请求信号 INTTPnCC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器中的值相等时的下一个计数时钟时产生。比较相等中断请求信号 INTTPnCC1 会在 16 位计数器的计数值与 CCR1 缓冲寄存器中的值相等时产生。

触发源可以从外部触发输入信号的有效沿和设置软件触发 (TPnCTL1.TPnEST 位 = 1) 的方式中进行选择。

备注 n = 0 ~ 3
 m = 0, 1

图 6-22. 单脉冲输出模式的寄存器设置 (1/2)

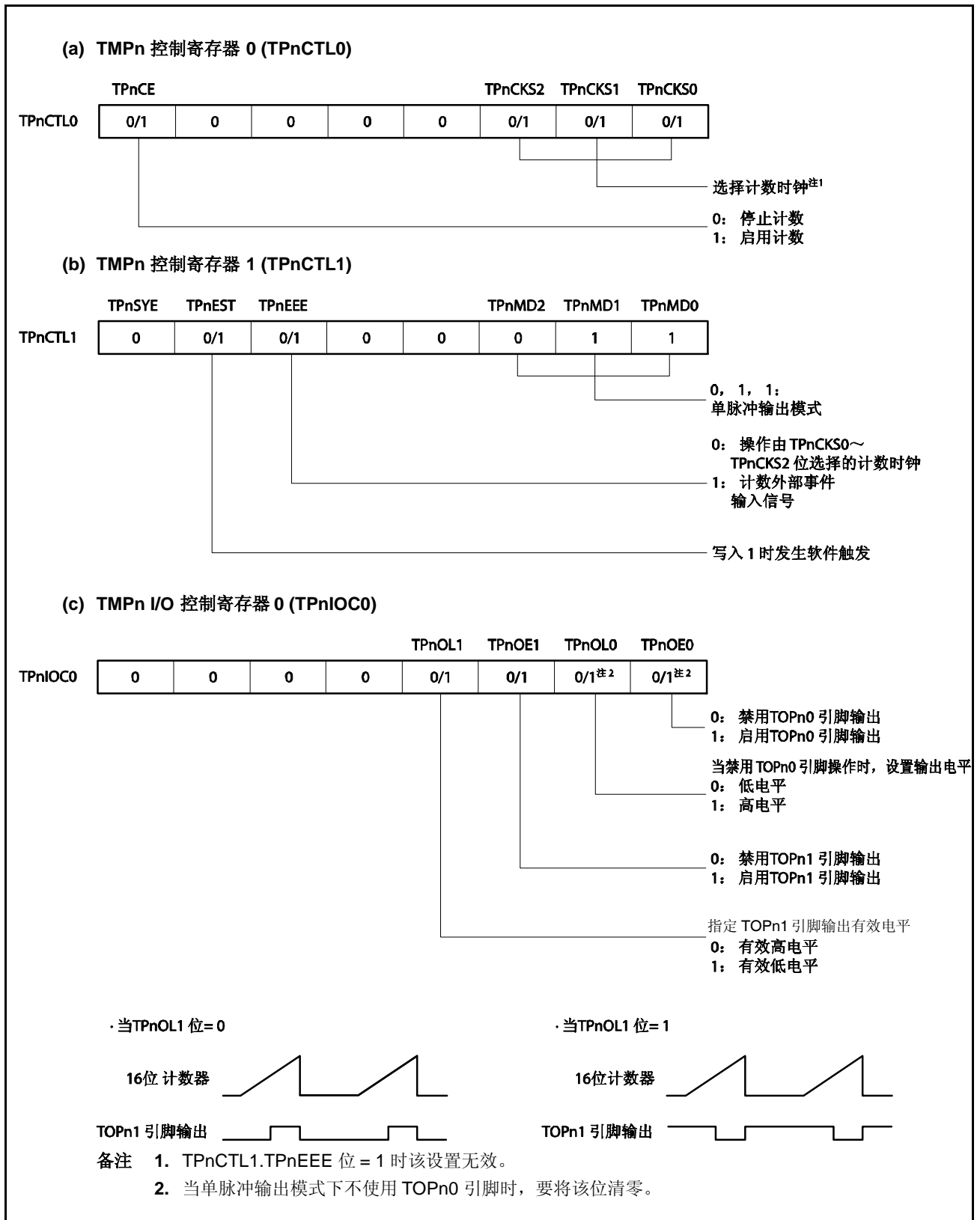


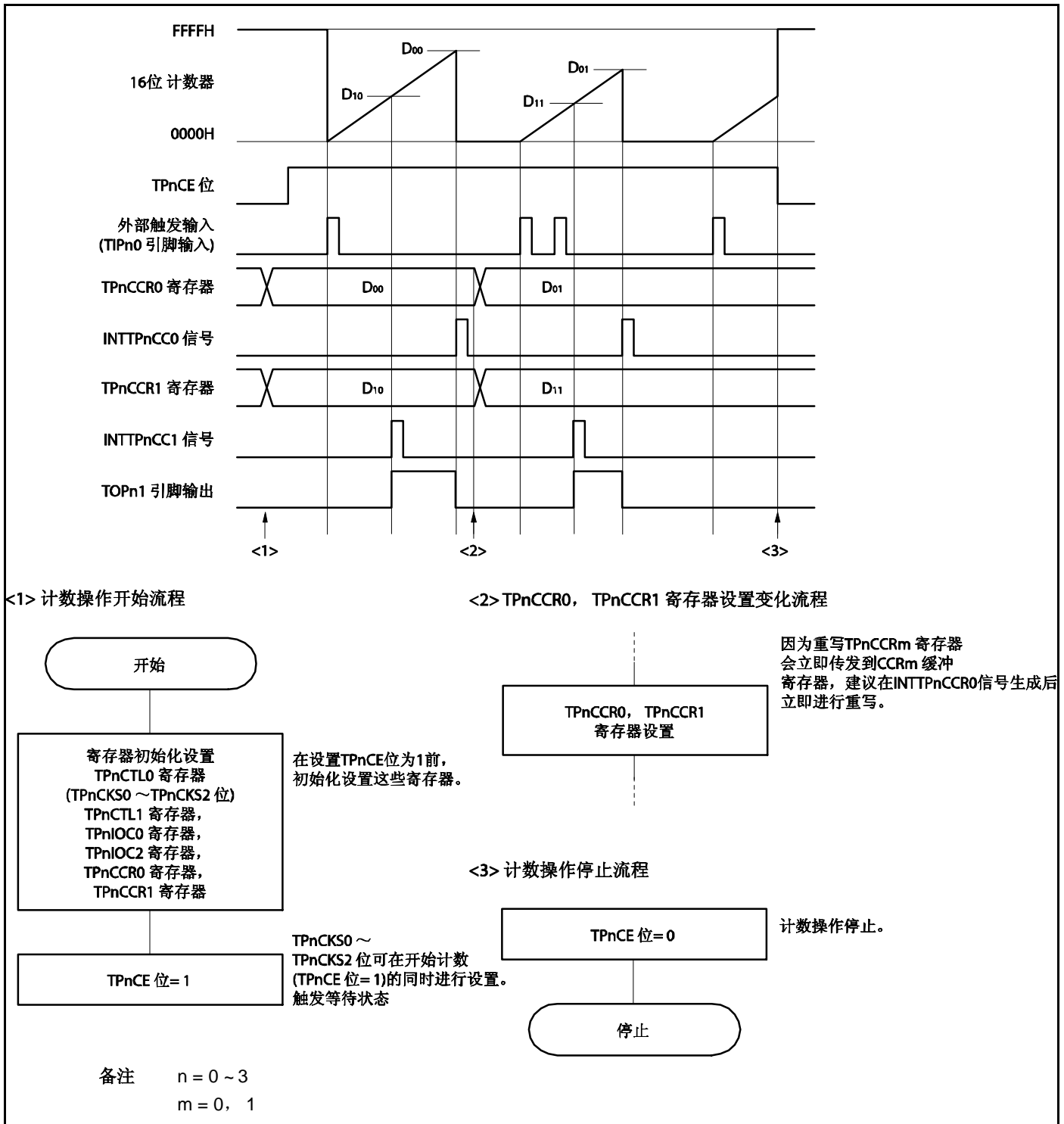
图 6-22. 单脉冲输出模式的寄存器设置 (2/2)



<R>

(1) 单脉冲输出模式的操作流程

图 6-23. 单脉冲输出模式的软件操作流程

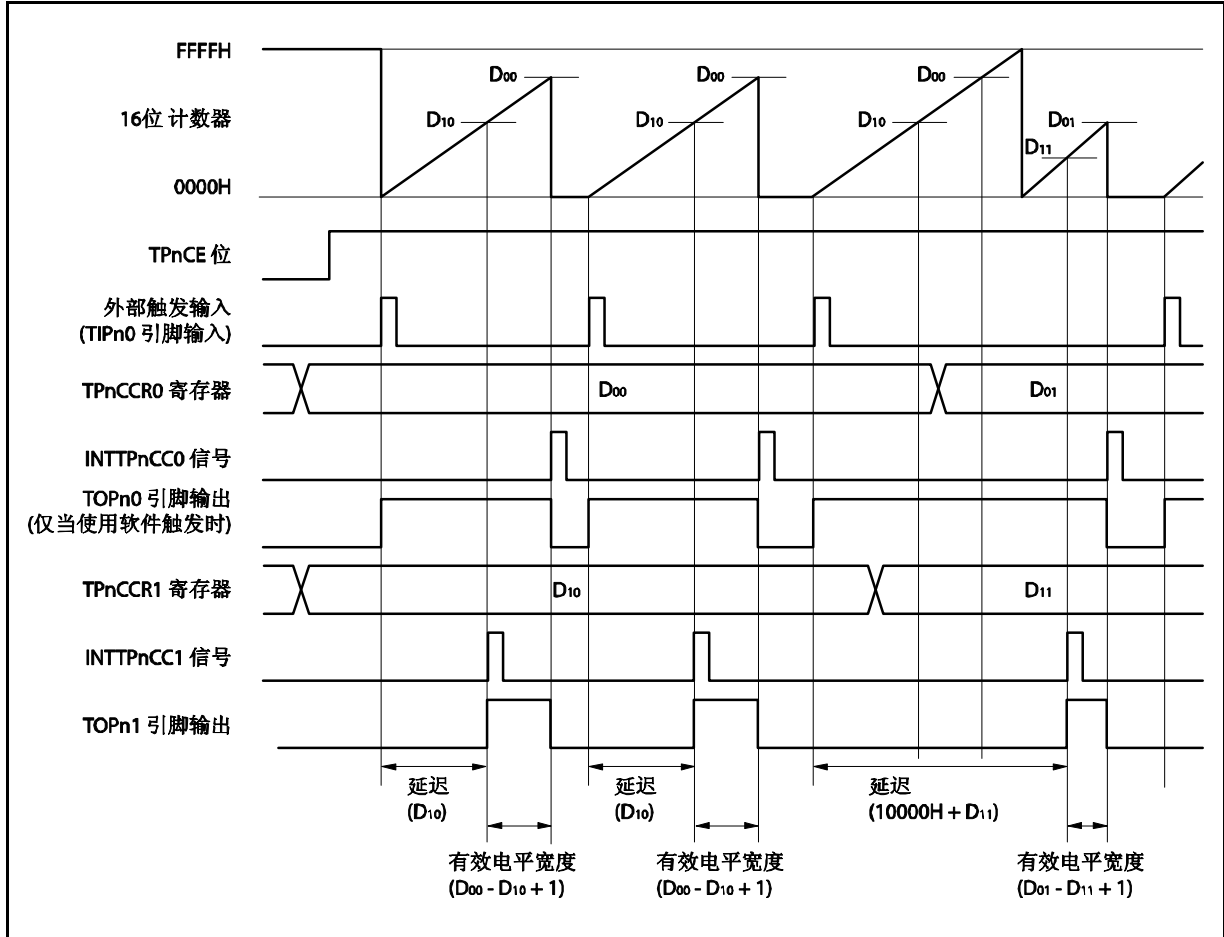


(2) 单脉冲输出模式的操作时序

(a) 改写 TPnCCRm 寄存器时需要注意的事项

将 TPnCCRm 寄存器的值减小时，要先停止计数，再对设定值进行改变。

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

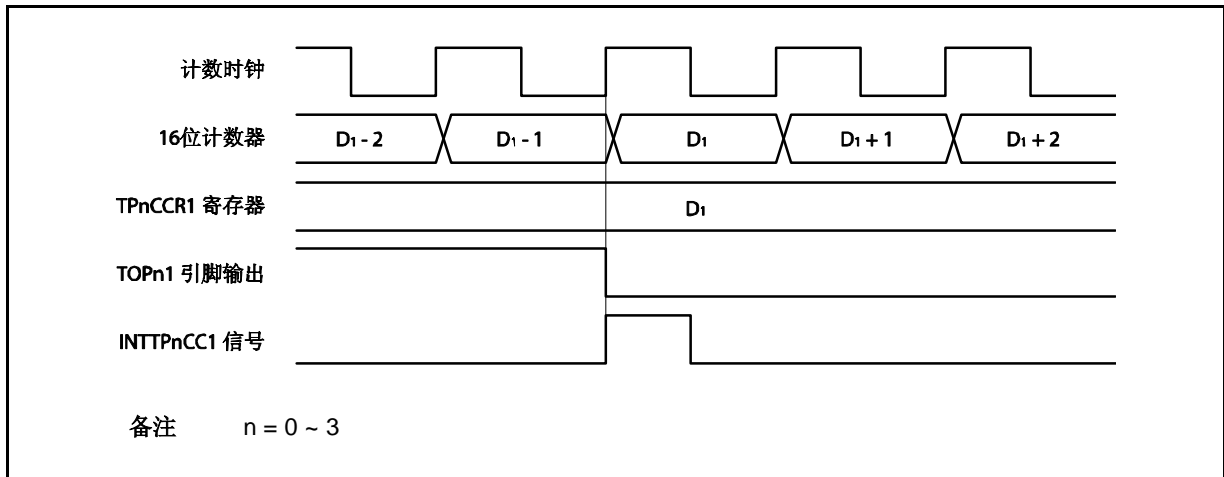


当将 TPnCCR0 寄存器的值从 D00 改写为 D01，TPnCCR1 寄存器的值从 D10 改写为 D11 时，这里假设 D00 > D01 且 D10 > D11，如果 TPnCCR1 寄存器被改写时，16 位计数器的计数值大于 D11 且小于 D10，并且 TPnCCR0 寄存器被改写时，计数器的计数值大于 D01 且小于 D00，两个设定值都将在改写后立即生效并与计数值进行比较。此时计数器会计数到 FFFFH 然后从 0000H 重新开始计数。当计数值与 D11 相等时，将产生 INTTPnCC1 信号，同时 TOPn1 引脚电平变为有效电平。当计数值与 D01 相等时，将产生 INTTPnCC0 信号，同时 TOPn1 引脚电平变为非有效电平并且计数器停止计数。因此可见，此时定时器输出的单脉冲波形的延迟时间或有效电平宽度会与预期数值有所差异。

备注 n = 0 ~ 3
m = 0, 1

(b) 比较相等中断请求信号的产生时序 (INTTPnCC1)

单脉冲输出模式下，INTTPnCC1 信号的产生时序与其它模式下 INTTPnCC1 信号的产生时序有所不同。这里，INTTPnCC1 信号将在 16 位计数器的计数值与 TPnCCR1 寄存器的值相等时产生。



通常情况下，INTTPnCC1 信号会在 16 位计数器计数值与 TPnCCR1 寄存器的值相等之后的下一个计数时钟同步产生。

但在单脉冲输出模式下，INTTPnCC1 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOPn1 引脚输出信号变化的时序而改变的。

备注 $n = 0 \sim 3$

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

PWM 输出模式下, TPnCTL0.TPnCE 位被置 1 后, TOPn1 引脚将输出一个 PWM 波形。
另外, TOPn0 引脚将输出一个以 PWM 波形周期的 2 倍为周期的脉冲信号。

图 6-24. PWM 输出模式的配置图

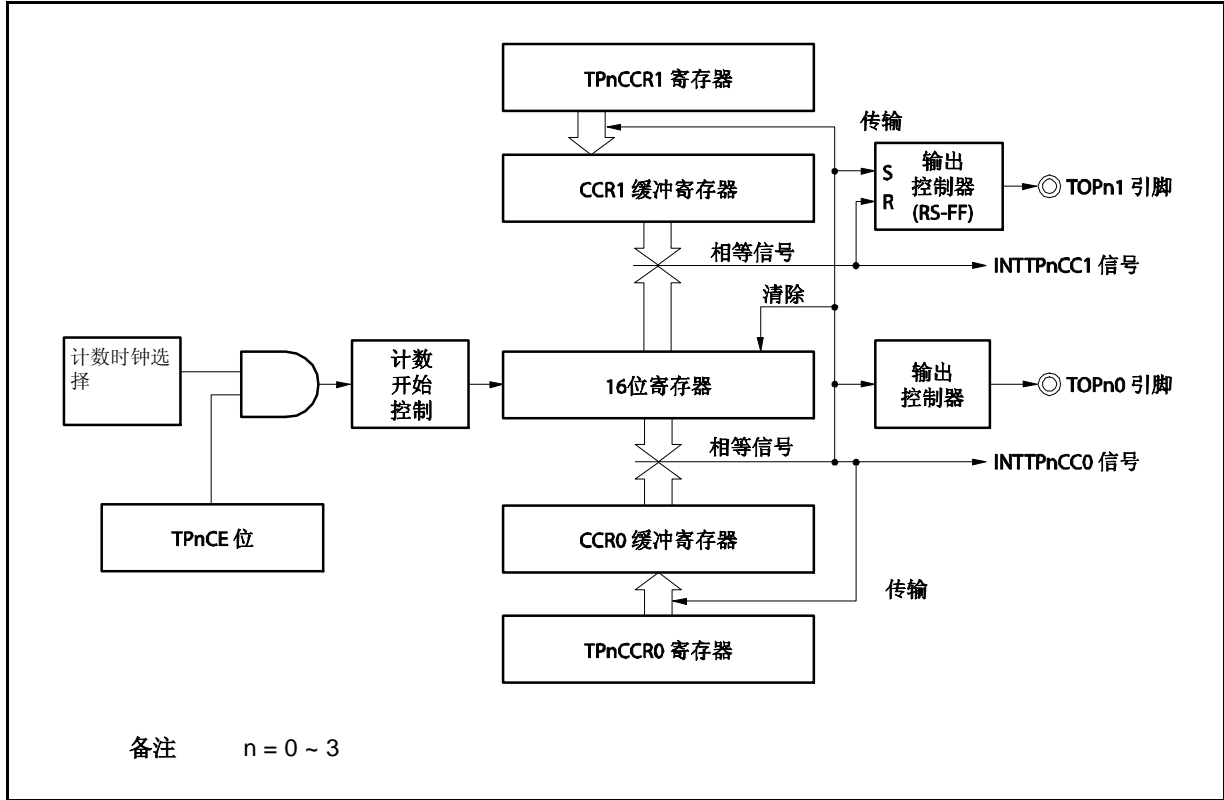
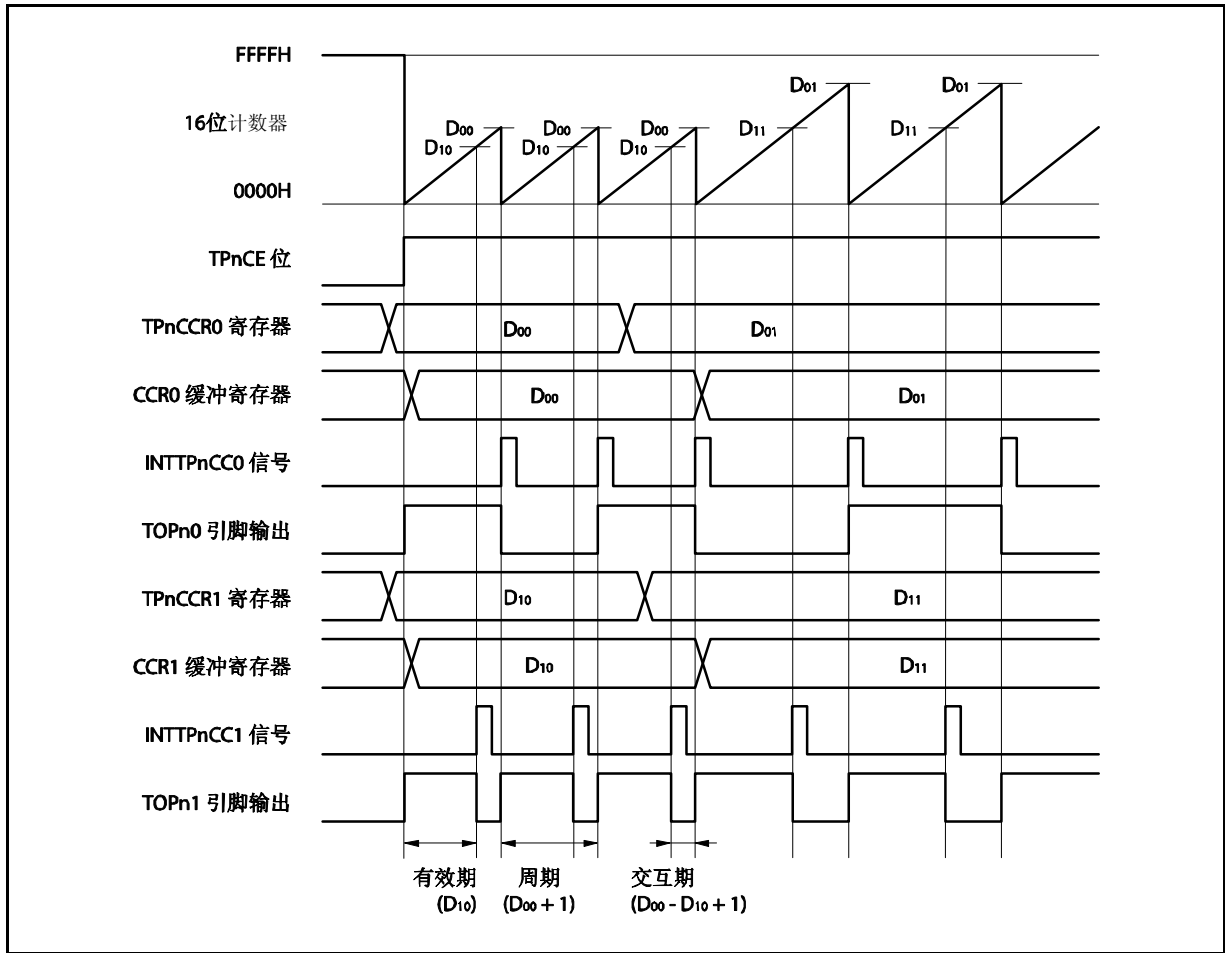


图 6-25. PWM 输出模式的基本时序



TPnCE 位被置 1 后，16 位计数器从 FFFFH 清零为 0000H 并开始计数，同时从 TOPn1 引脚输出一个 PWM 波形。

PWM 波形的有效电平宽度，周期以及占空比系数可由以下算式得出。

$$\begin{aligned} \text{有效电平宽度} &= (\text{TPnCCR1 寄存器的值}) \times \text{计数时钟周期} \\ \text{周期} &= (\text{TPnCCR0 寄存器的值} + 1) \times \text{计数时钟周期} \\ \text{占空比系数} &= (\text{TPnCCR1 寄存器的值}) / (\text{TPnCCR0 寄存器的值} + 1) \end{aligned}$$

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

比较相等中断请求信号 INTTPnCC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器的值相等后的下一个计数时钟时产生，同时 16 位计数器将被清零。比较相等中断请求信号 INTTPnCC1 会在 16 位计数器的计数值与 CCR1 缓冲寄存器的值相等时产生。

当 16 位计数器的计数值与 CCRm 缓冲寄存器中的值相等时，TPnCCRm 寄存器中设置的值会被传送至 CCRm 缓冲寄存器，同时 16 位计数器被清零。

备注 n = 0 ~ 3, m = 0, 1

图 6-26. PWM 输出模式的寄存器设置 (1/2)

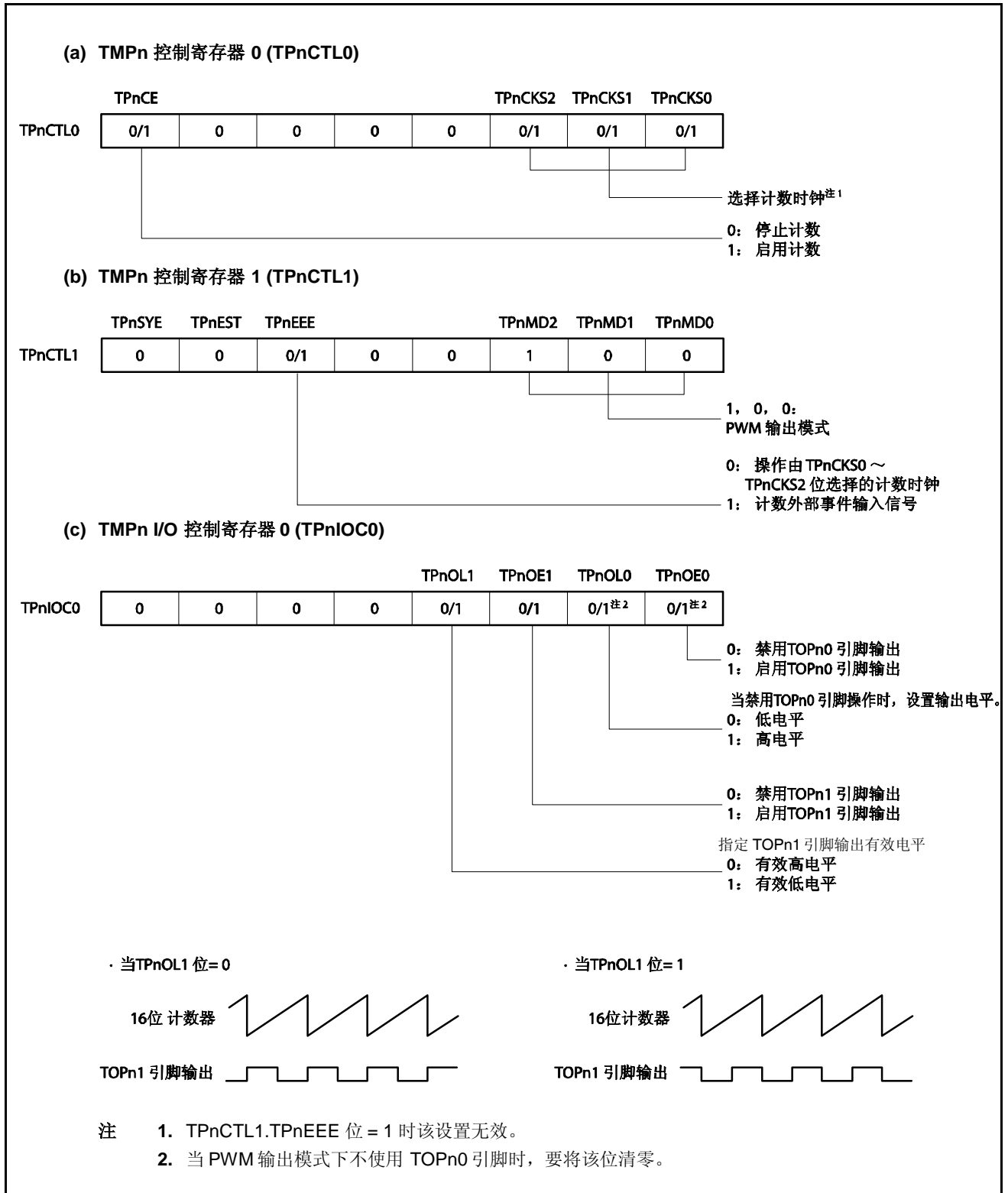
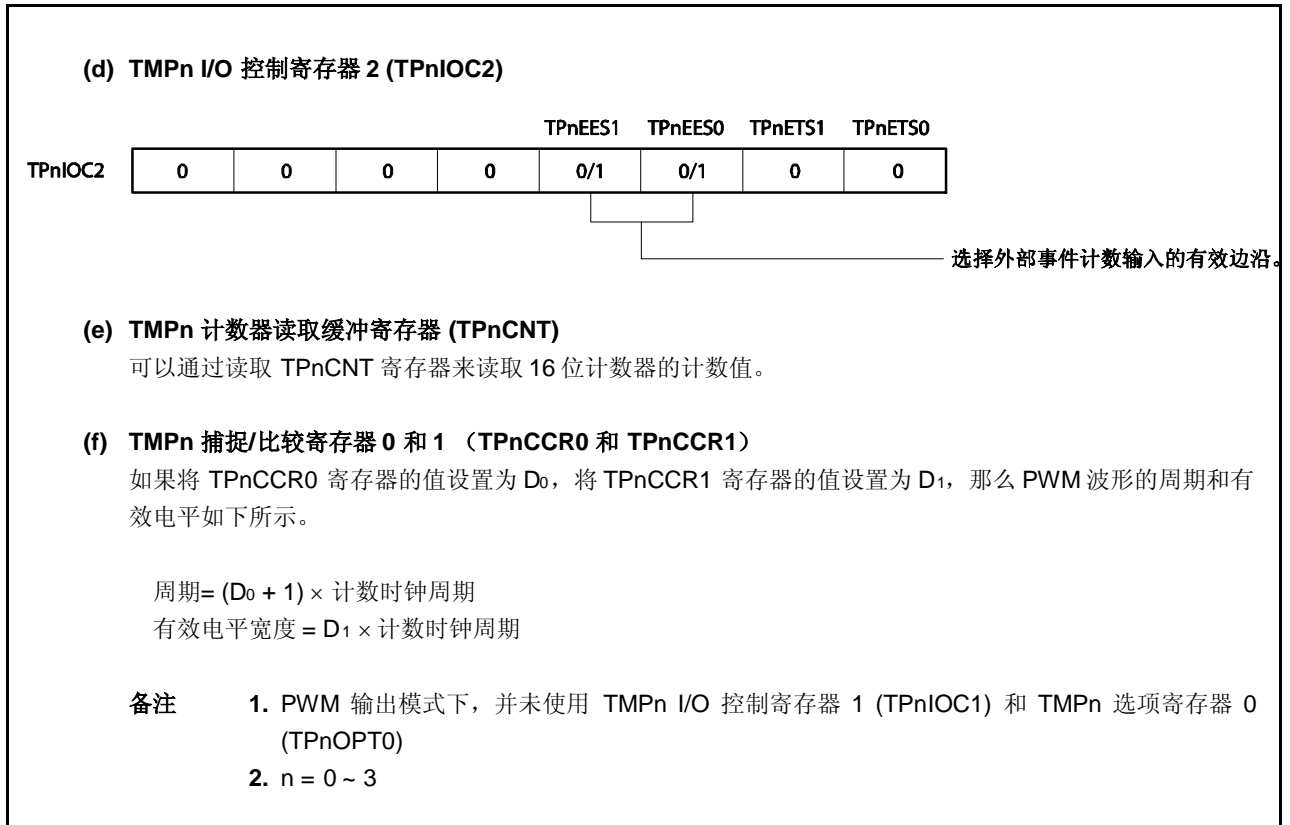


图 6-26. PWM 输出模式的寄存器设置 (2/2)



(1) PWM 输出模式的操作流程

图 6-27. PWM 输出模式的软件处理流程 (1/2)

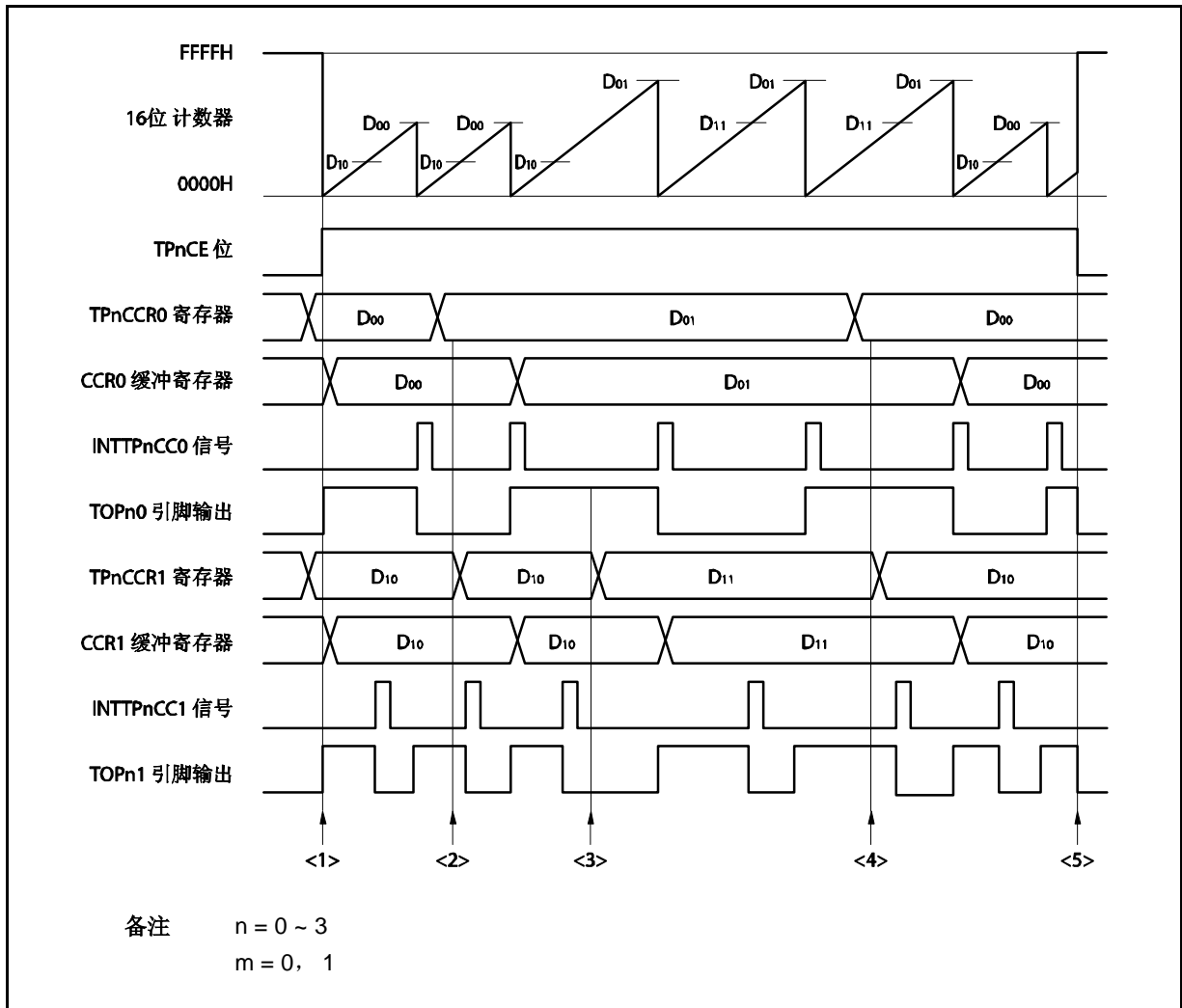
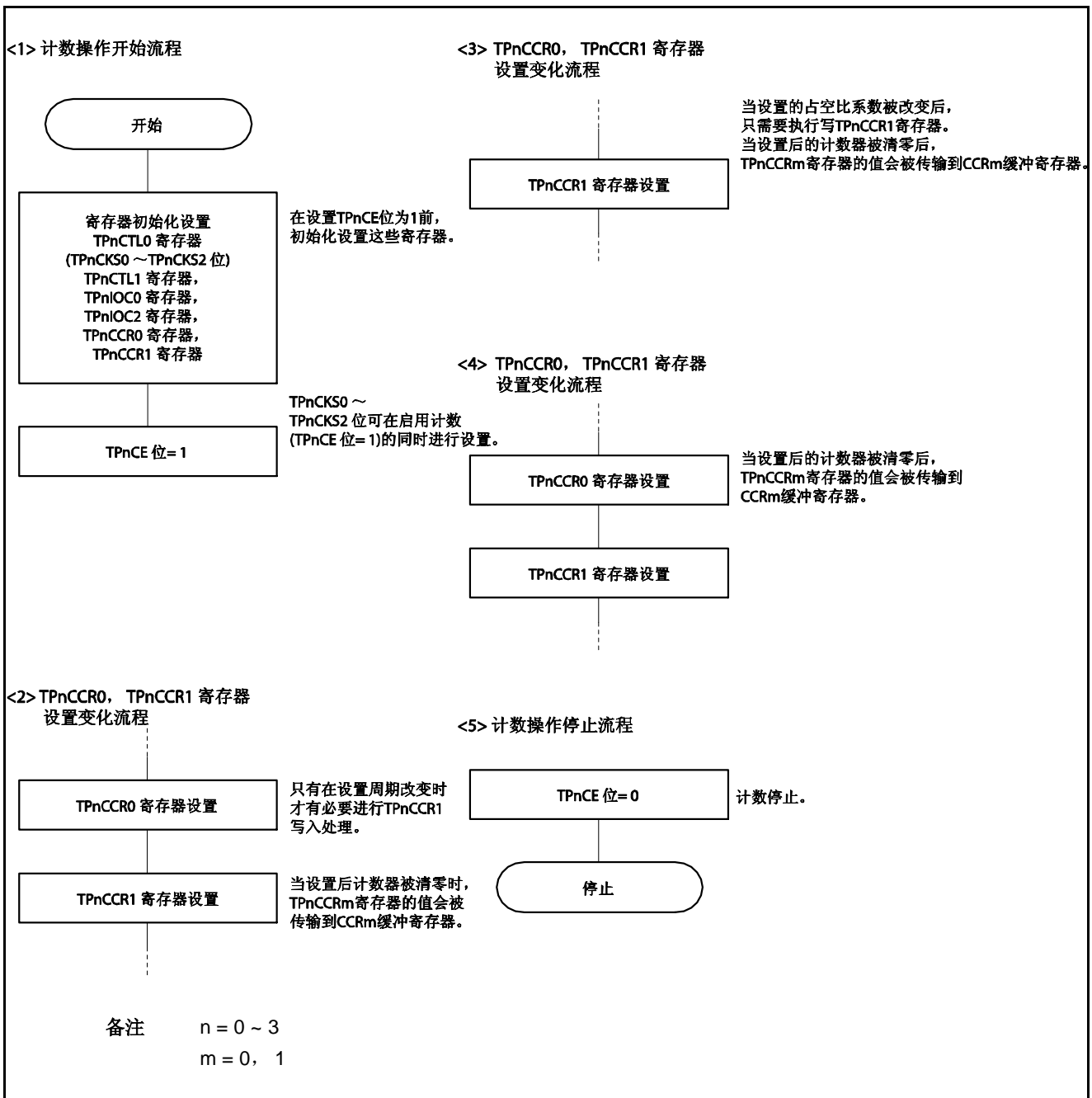


图 6-27. PWM 输出模式的软件处理流程 (2/2)

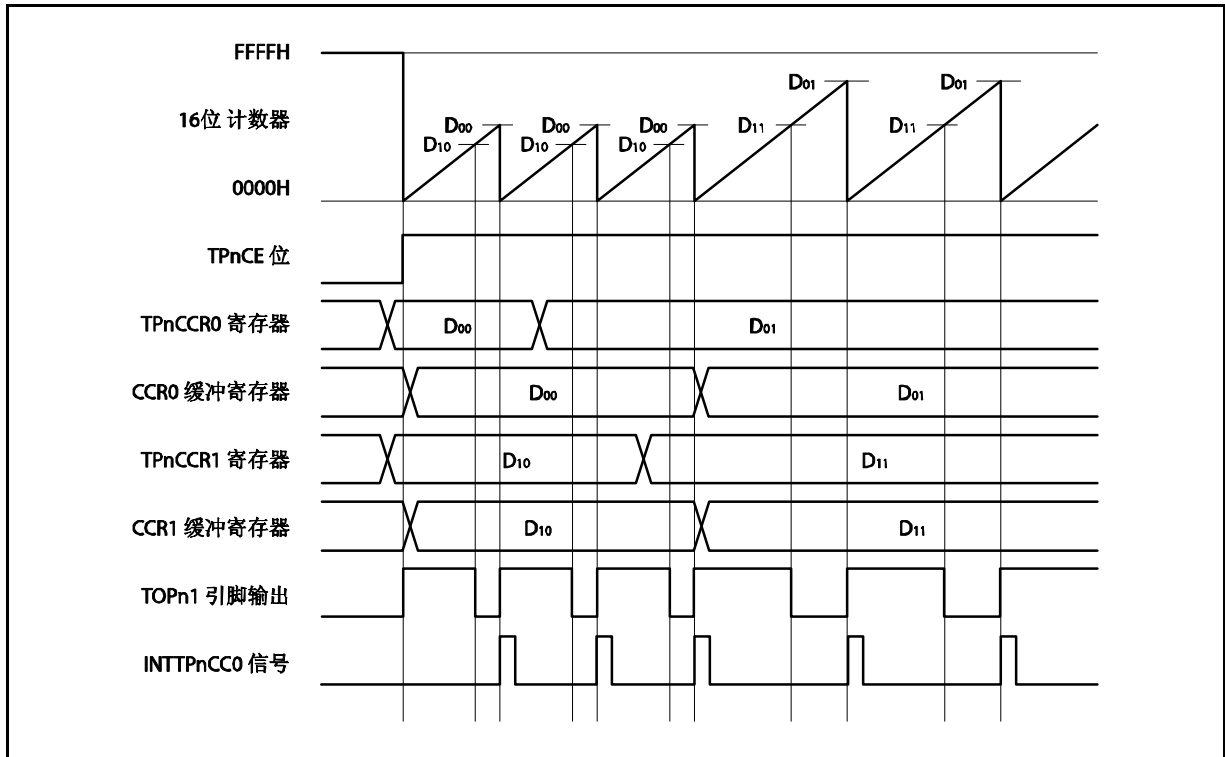


(2) PWM 输出模式的操作时序

(a) 操作过程中改变脉宽度

计数器计数过程中，若改变 PWM 波形，则要在最后进行对 TPnCCR1 寄存器的写入操作。

对 TPnCCR1 寄存器进行写入操作之后，若需再次更改 TPnCCRm 寄存器的值，须等待下一个 INTTPnCC1 信号被检测到。



若要将 TPnCCRm 寄存器中设置的数据传送至 CCRm 缓冲寄存器，必须对 TPnCCR1 寄存器进行写入操作。

同时改变 PWM 波形的周期和有效电平时，先将周期值写入 TPnCCR0 寄存器，再将有效电平宽度值写入 TPnCCR1 寄存器。

只改变 PWM 波形的周期时，先将周期值写入 TPnCCR0 寄存器，再向 TPnCCR1 寄存器写入与原来相同的值。

只改变 PWM 波形的有效电平宽度（占空比系数）时，只需重新设置 TPnCCR1 寄存器的值即可。

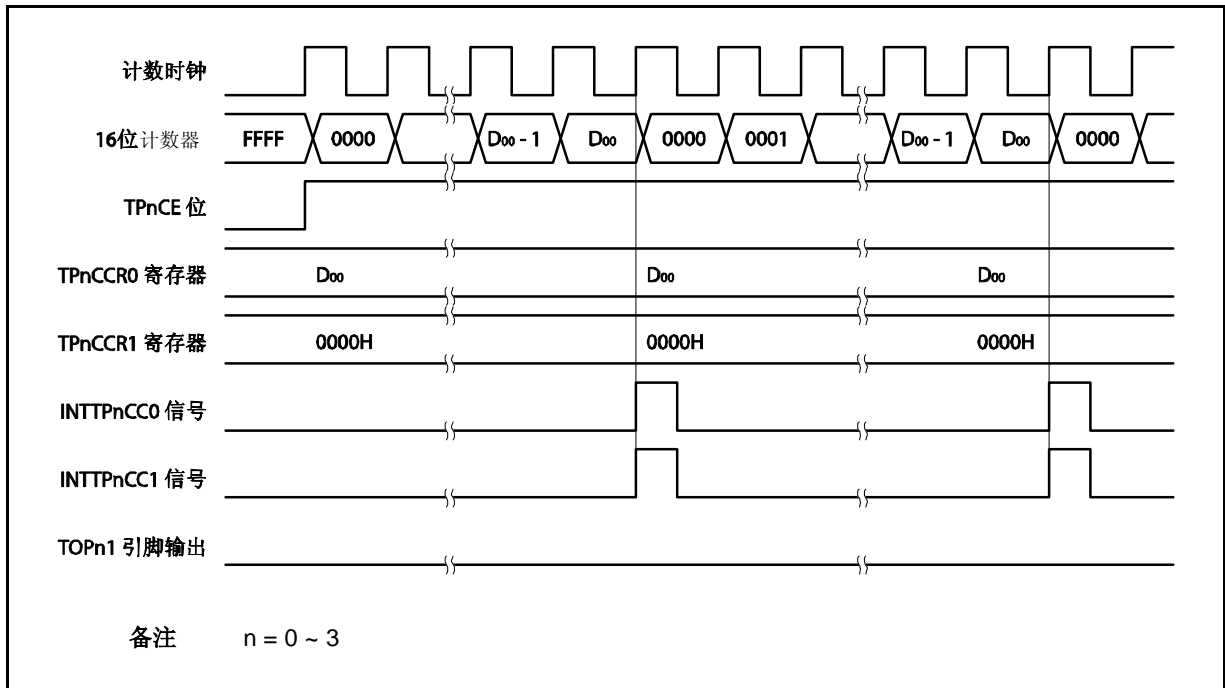
数据被写入 TPnCCR1 寄存器之后，16 位计数器被清零时，TPnCCRm 寄存器中的数值才会被同步传送至 CCRm 缓冲寄存器中，此时 CCRm 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

向 TPnCCR1 寄存器的写入操作完成后，若需再次改写 TPnCCR0 或 TPnCCR1 寄存器的值，则要在产生 INTTPnCC0 信号之后再进行改写。否则，由于数据从 TPnCCRm 寄存器传送至 CCRm 缓冲寄存器的时刻与 TPnCCRm 寄存器的写入时刻相冲突，CCRM 缓冲寄存器中所设置的数值可能无法确定。

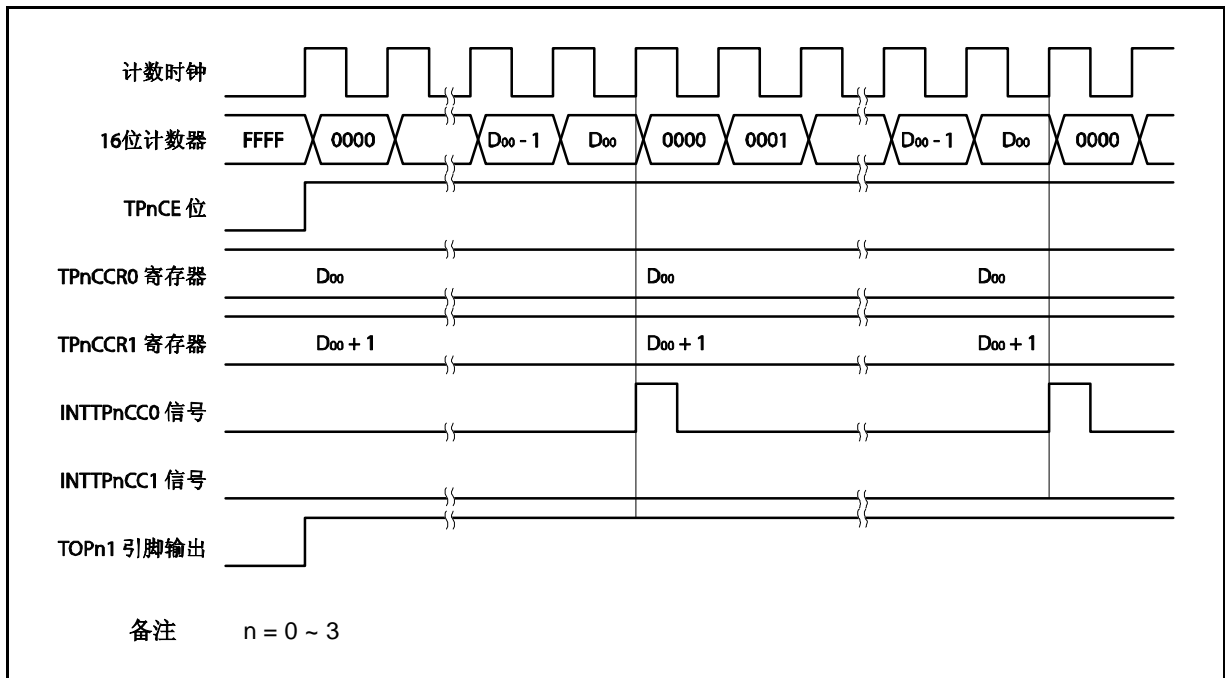
备注 n = 0 ~ 3, m = 0, 1

(b) PWM 波形的 0%/100% 输出

若要输出 0% 的波形，则需将 TPnCCR1 寄存器设置为 0000H。如果 TPnCCR0 寄存器被设置为 FFFFH，那么 INTTPnCC1 信号将周期性产生。

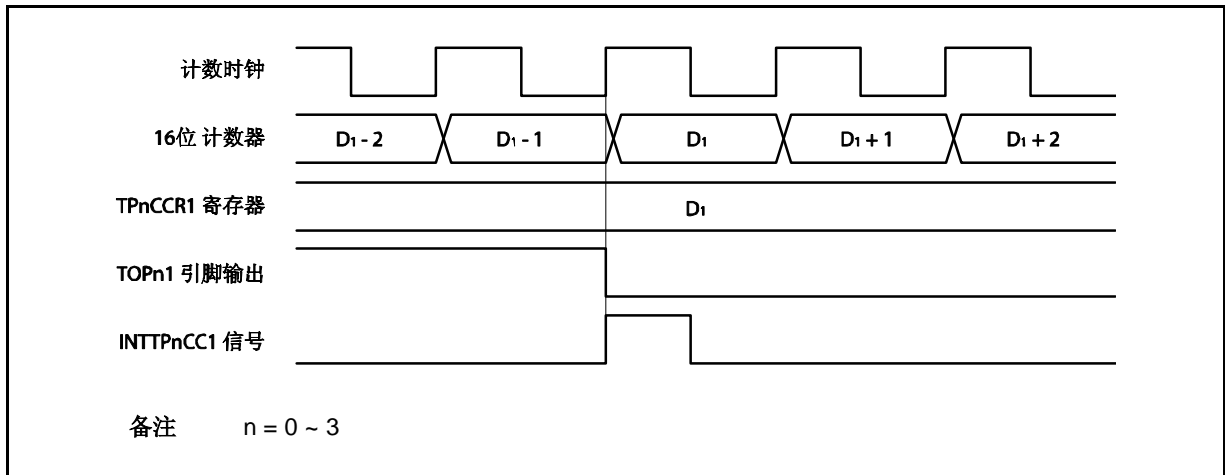


若要输出 100% 波形，则需将 TPnCCR1 寄存器设置为 (TPnCCR0 寄存器值 + 1)。TPnCCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。



(c) 比较相等中断请求信号的产生时序 (INTTPnCC1)

PWM 输出模式下，INTTPnCC1 信号的产生时序与其它模式下 INTTPnCC1 信号的产生时序有所不同，这里，INTTPnCC1 信号将在 16 位计数器的计数值与 TPnCCR1 寄存器的值相等时产生。



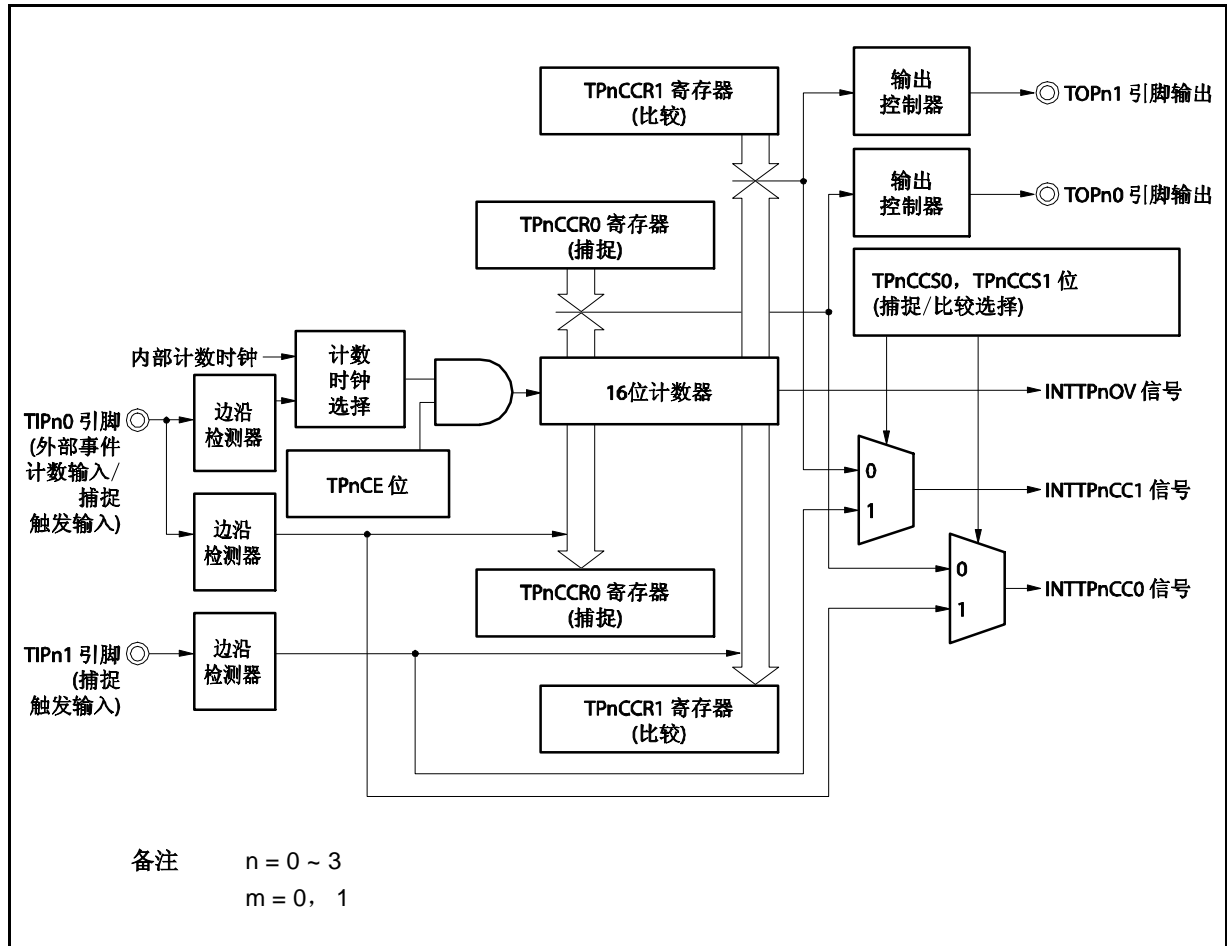
通常情况下，INTTPnCC1 信号会在 16 位计数器计数值与 TPnCCR1 寄存器的值相等之后的下一个计数时钟同步产生。

但在 PWM 输出模式下，INTTPnCC1 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOPn1 引脚输出信号变化的时序而改变的。

6.5.6 自由运行定时器模式 (TPnMD2 ~ TPnMD0 位 = 101)

在自由运行定时器模式中，TPnCTL0.TPnCE 位被置 1 后，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 寄存器的功能可以在比较寄存器和捕捉寄存器之间进行选择。

图 6-28. 自由运行定时器模式的配置图

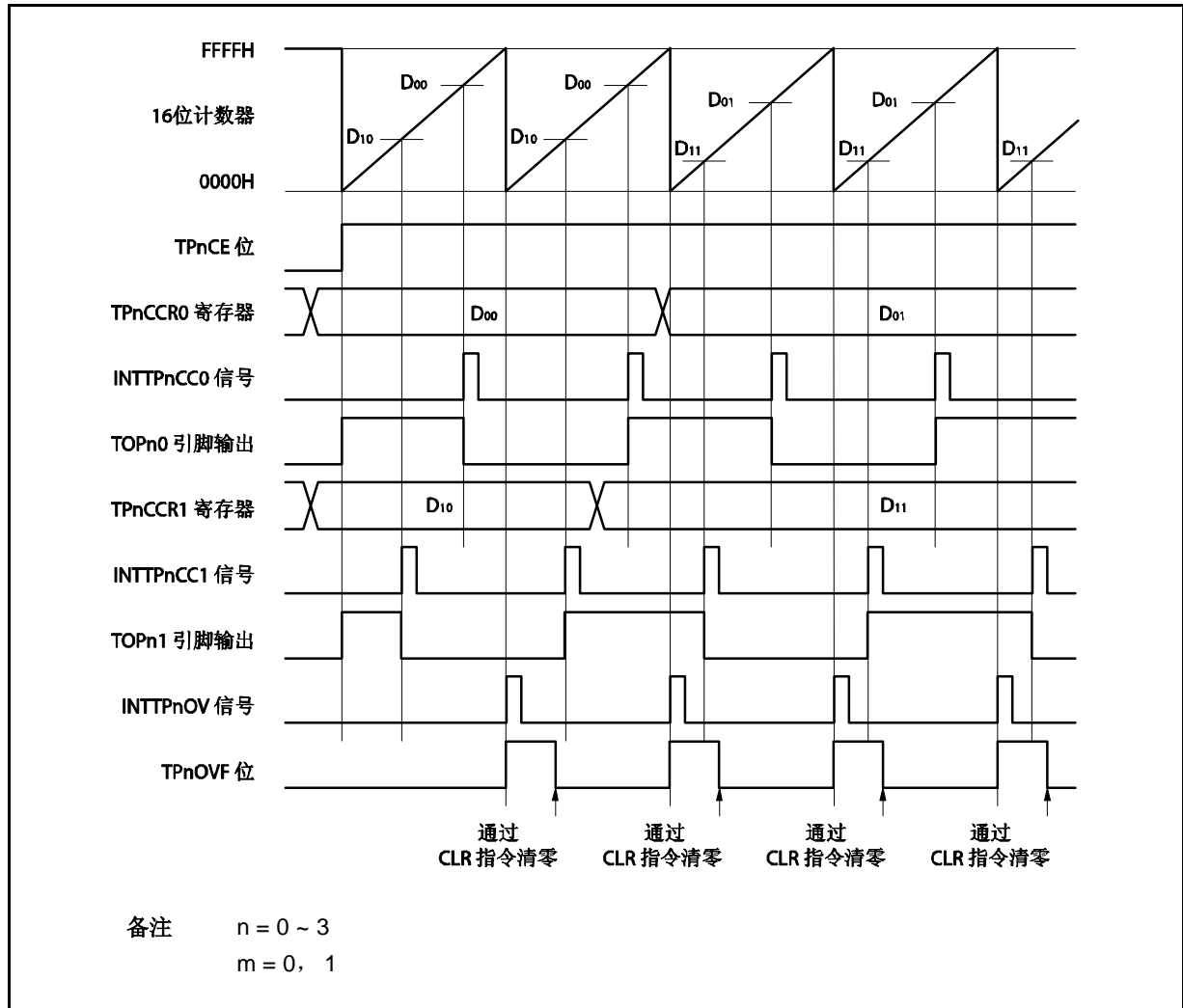


TPnCE 位被置 1 后，16 位定时器/事件 P 计数器开始计数，同时 TOPn0 和 TOPn1 引脚输出信号的电平反转。当 16 位计数器的计数值与 TPnCCRm 寄存器的值相等时将产生一个比较相等中断请求信号(INTTPnCCm)，同时 TOPnm 引脚的输出电平将反转。

此后，16 位计数器将继续与计数时钟同步计数。当它计数到 FFFFH 时，将在下一个计数时钟时产生一个溢出中断请求信号(INTTPnOV)，同时计数器被清零并重新开始计数。此时，溢出标志(TPnOPT0.TPnOVF)也会被置 1。请通过软件使用 CLR 指令将溢出标志位清零。

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

图 6-29. 自由运行定时器模式的基本时序（比较功能）



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

此后，16 位计数器将继续与计数时钟同步计数。当它计数到 FFFFH 时，将在下一个计数时钟时产生一个溢出中断请求信号(INTTPnOV)，同时计数器被清零并重新开始计数。此时，溢出标志(TPnOPT0.TPnOVF)也会被置 1。请通过软件使用 CLR 指令将溢出标志位清零。

图 6-30. 自由运行定时器模式的基本时序（捕捉功能）

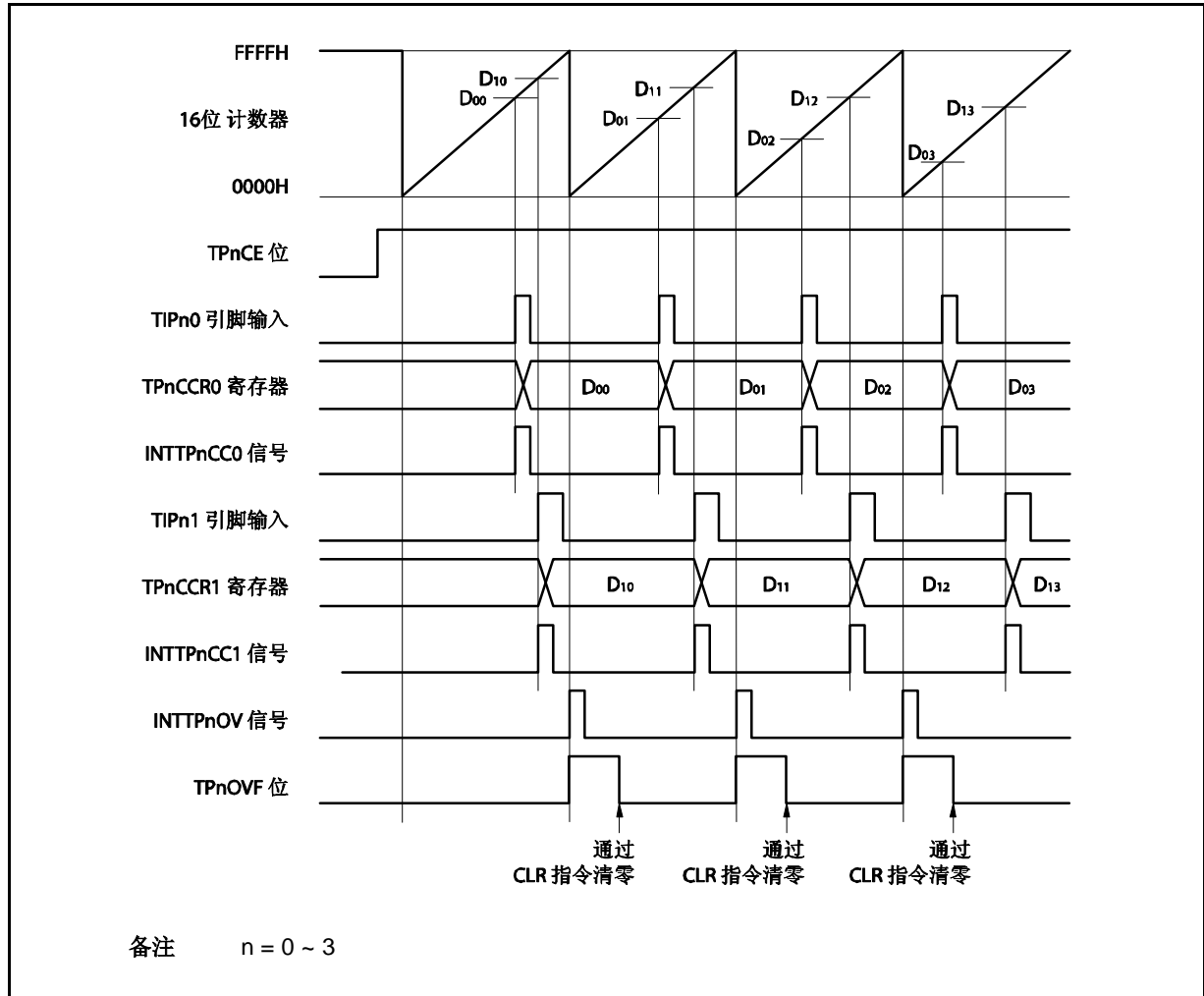


图 6-31. 自由运行定时器模式的寄存器设置 (1/2)

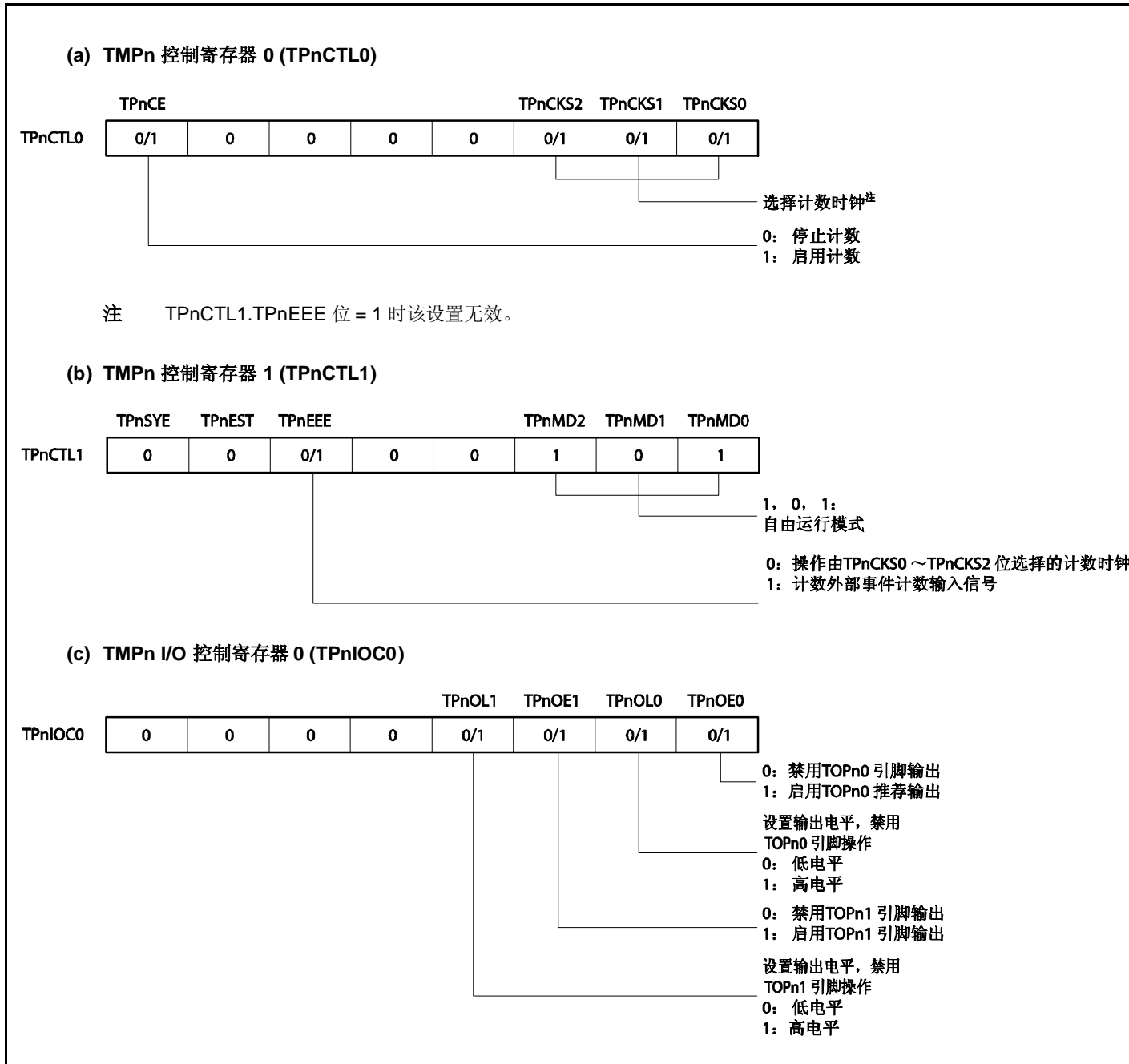
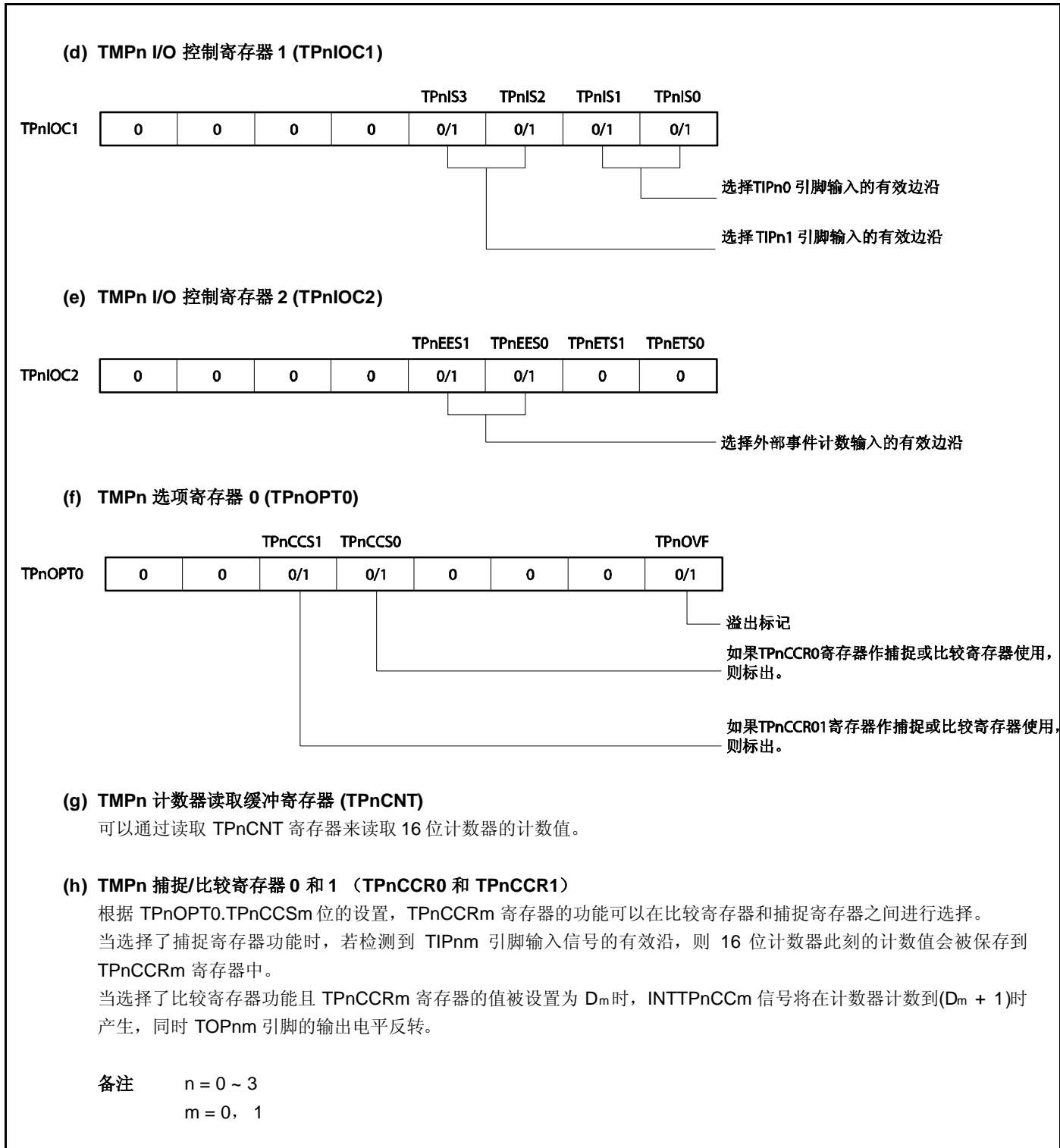


图 6-31. 自由运行定时器模式的寄存器设置 (2/2)



(1) 自由运行定时器模式的操作系统

(a) 当捕捉/比较寄存器用作比较寄存器时

图 6-32. 自由运行定时器模式的软件处理流程（比较功能）(1/2)

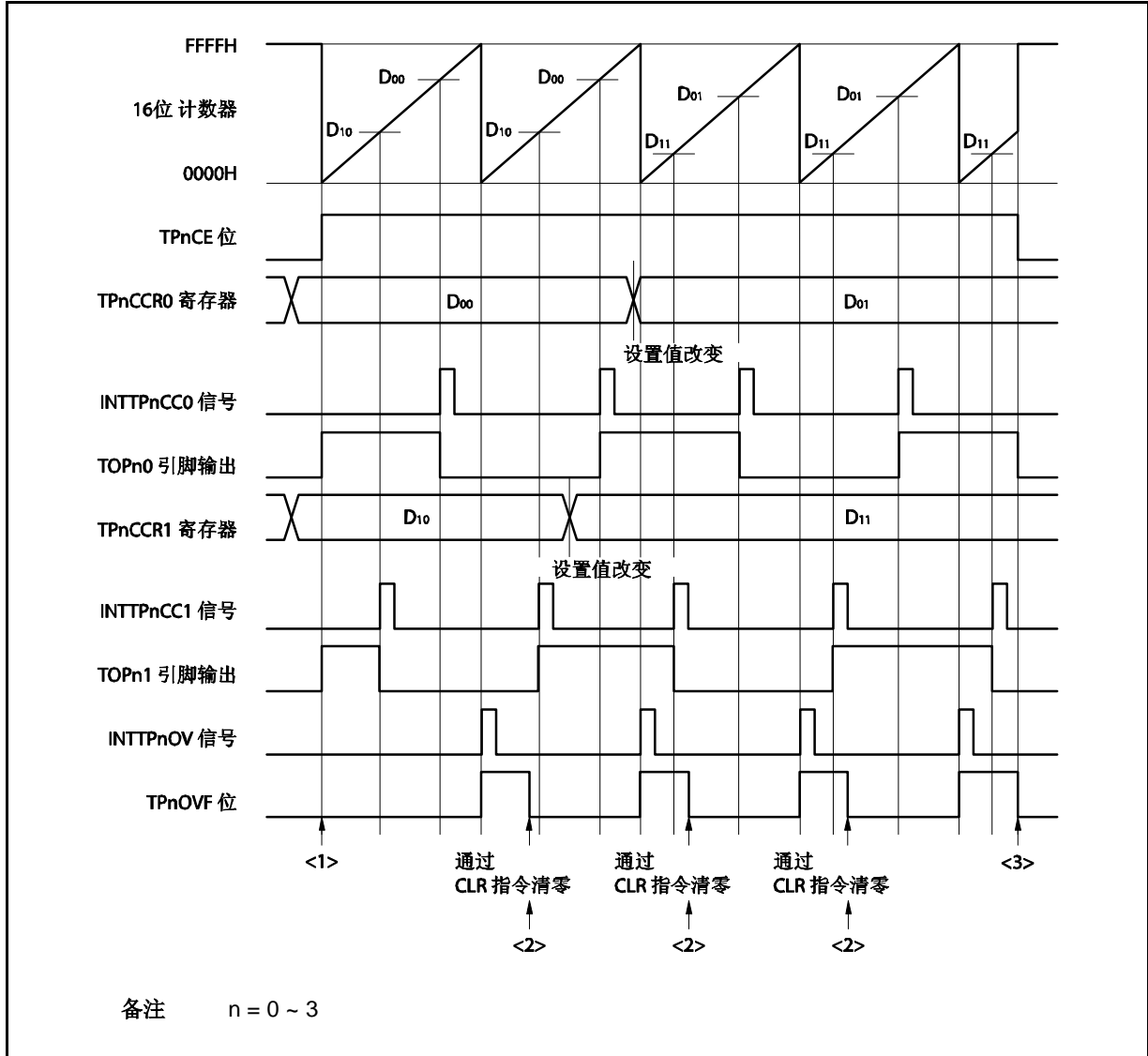
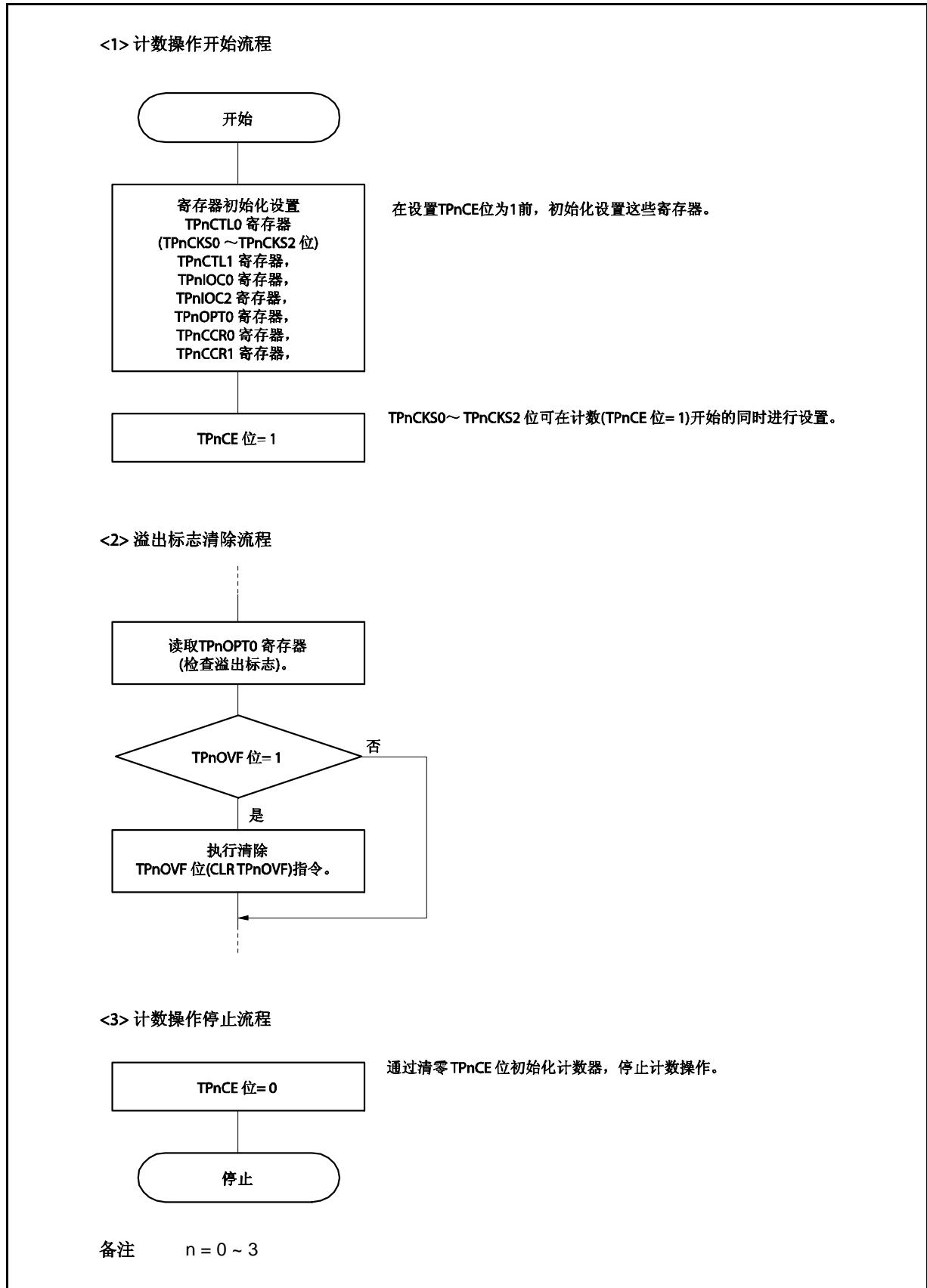


图 6-32. 自由运行定时器模式的软件处理流程（比较功能）(2/2)



(b) 当捕捉/比较寄存器用作捕捉寄存器时

图 6-33. 自由运行定时器模式的软件处理流程（捕捉功能）(1/2)

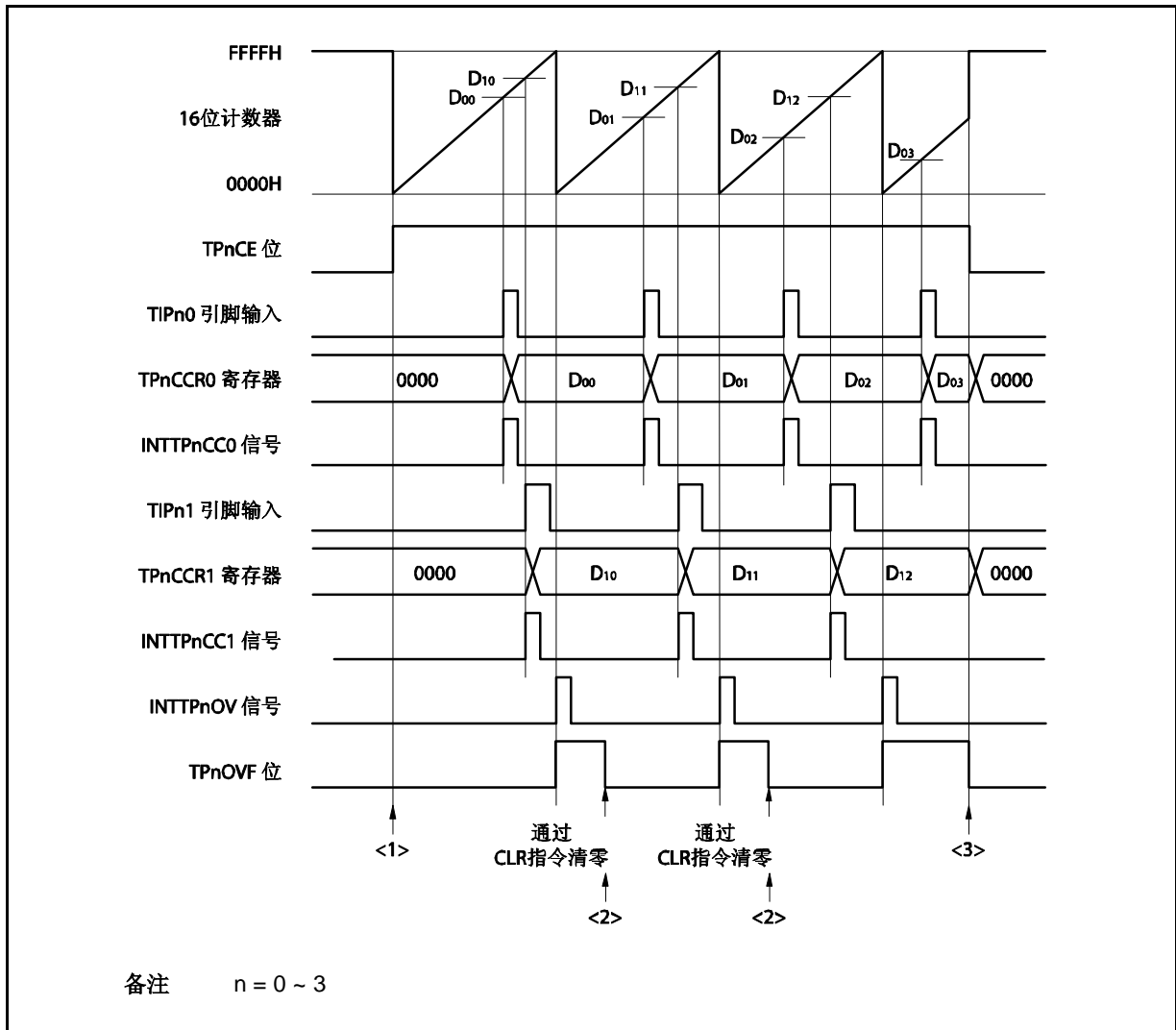
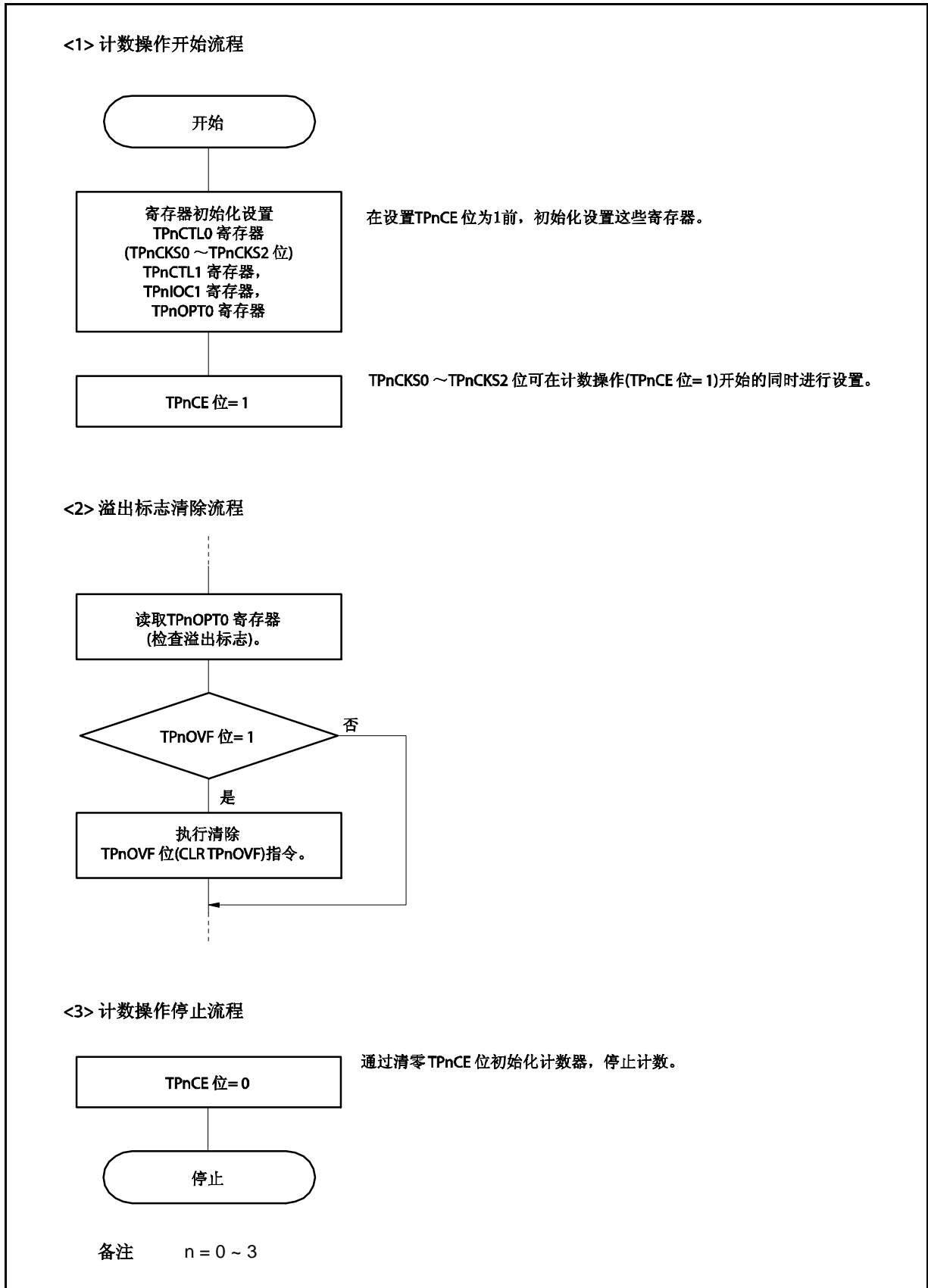


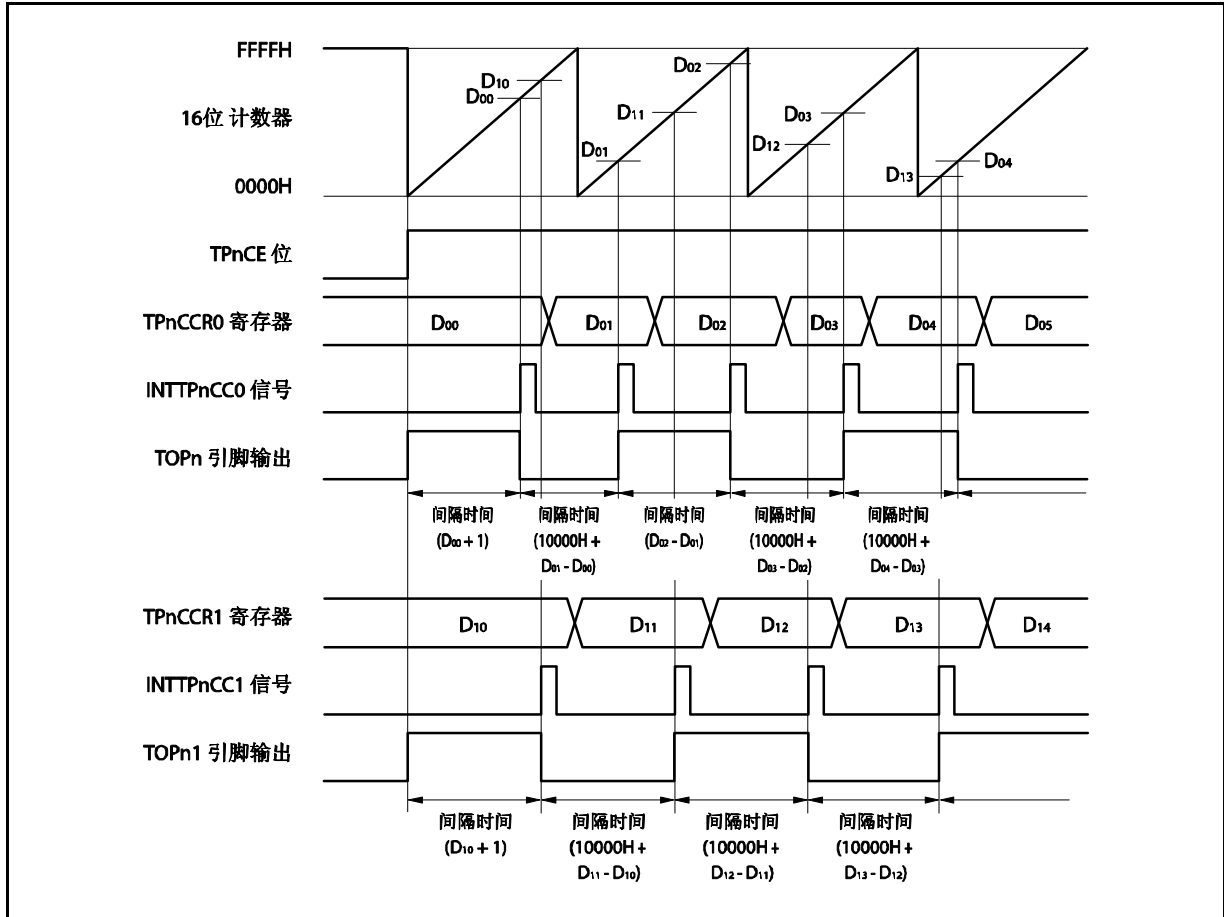
图 6-33. 自由运行定时器模式的软件处理流程（捕捉功能）(2/2)



(2) 自由运行定时器模式的操作时序

(a) 比较寄存器的间隔操作

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



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

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

比较寄存器的最初设置值： $D_m - 1$

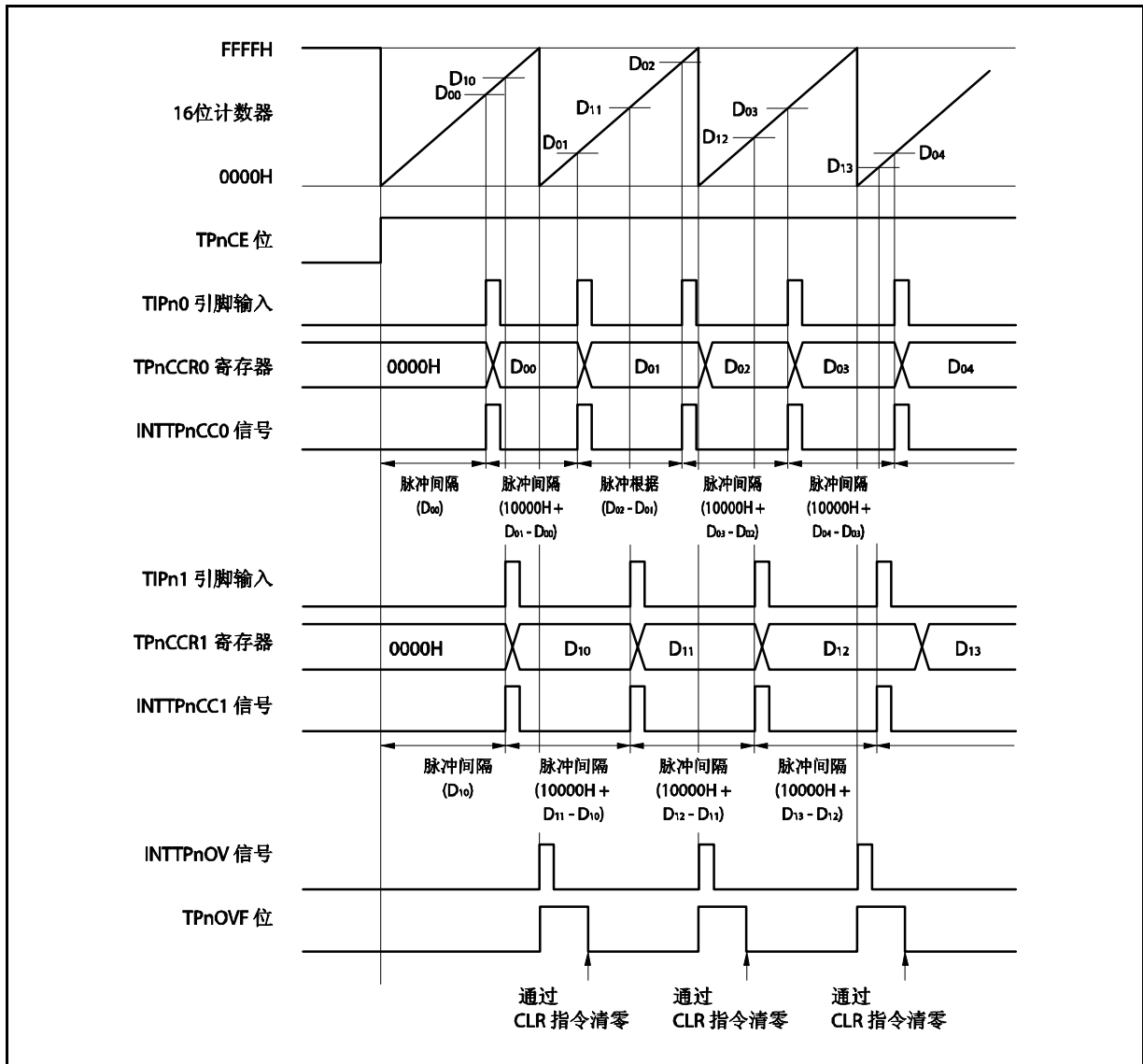
第二次设置比较寄存器以后（含第二次）的设置值：前次设置值 + Dm

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

备注 $n = 0 \sim 3$
 $m = 0, 1$

(b) 使用捕捉寄存器进行脉宽测量

当 TPnCCRm 寄存器被作为捕捉寄存器执行脉冲宽度测量时，这里，需要通过软件处理在每次检测 INTTPnCCm 信号时对捕捉寄存器进行读取并计算间隔时间。

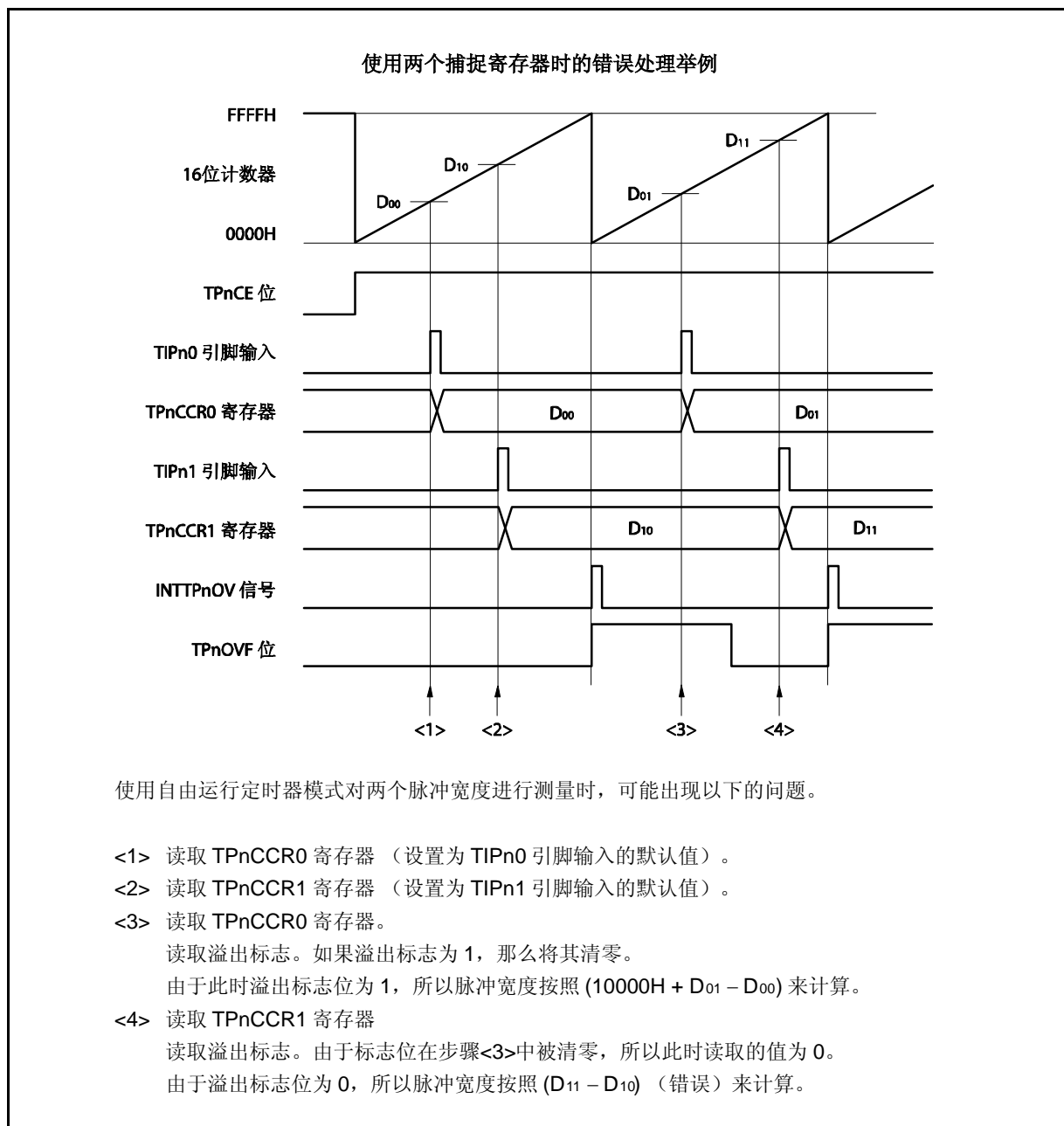


当使用自由运行定时器模式进行脉宽测量操作时，可以在同一通道对两个不同的脉冲宽度进行测量。通过与 INTTPnCCm 信号同步读取 TPnCCRm 寄存器的值并计算该值与上一次读取值的差，就可以求得脉冲的宽度。

备注 $n = 0 \sim 3$
 $m = 0, 1$

(c) 使用两个捕捉寄存器产生溢出时的操作

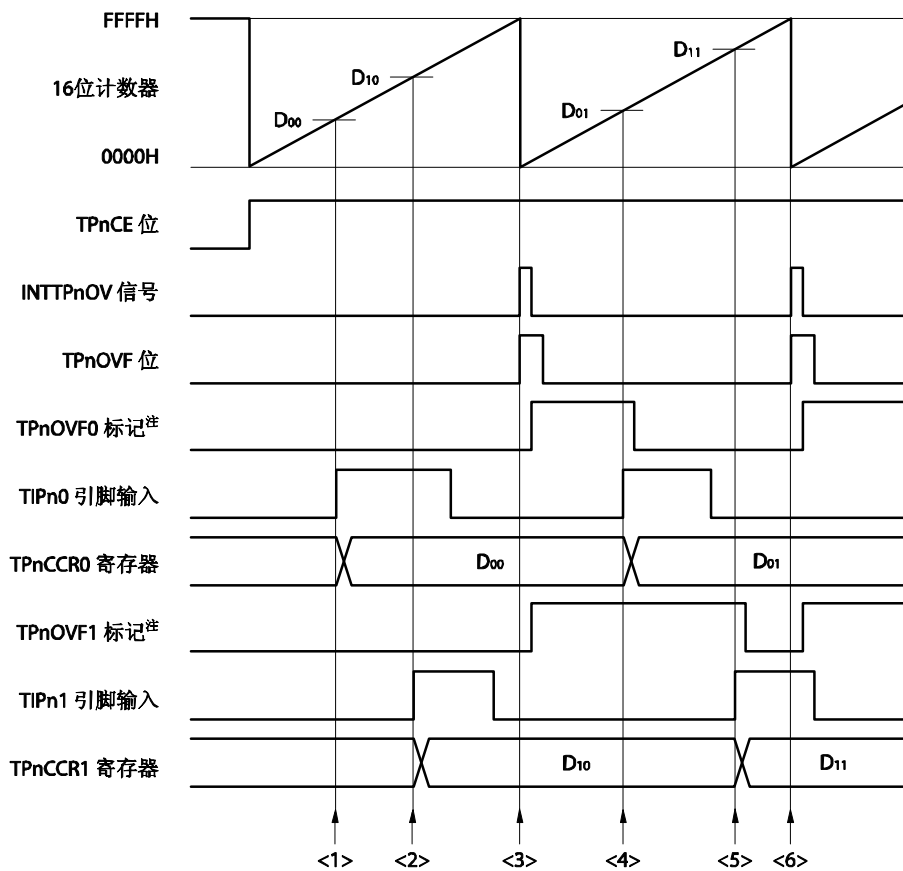
使用两个捕捉寄存器时，请务必慎重处理溢出标志位。首先，给出一个错误处理的例子，如下图所示。



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

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

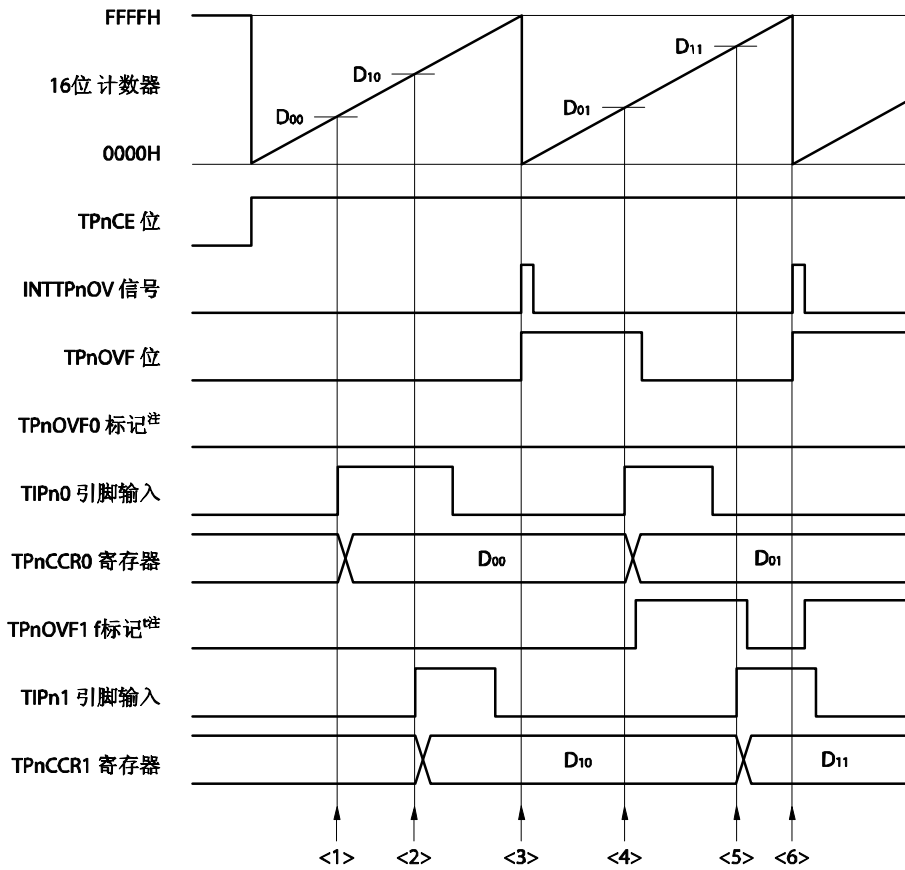
使用两个捕捉寄存器时的处理举例（利用溢出中断）



注 这里的 TPnOVF0 和 TPnOVF1 标志是建立于内部 RAM 上的变量，可由软件任意进行设置。

- <1> 读取 TPnCCR0 寄存器（设置为 TIPn0 引脚输入的默认值）。
- <2> 读取 TPnCCR1 寄存器（设置为 TIPn1 引脚输入的默认值）。
- <3> 产生了溢出。在溢出中断服务程序中将 TPnOVF0 和 TPnOVF1 标志位置 1，同时将硬件溢出标志位清零。
- <4> 读取 TPnCCR0 寄存器。
读取 TPnOVF0 标志位，如果该标志位为 1，那么将其清零。
由于 TPnOVF0 标志位为 1，脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。
- <5> 读取 TPnCCR1 寄存器。
读取 TPnOVF1 标志位，如果该标志位为 1，那么将其清零。（步骤<4>中将 TPnOVF0 位清零但 TPnOVF1 位始终保持为 1。）
由于 TPnOVF1 标志为 1，脉冲宽度按照 $(10000H + D_{11} - D_{10})$ （正确）来计算。
- <6> 同步骤 <3>

使用两个捕捉寄存器时的处理举例（不利用溢出中断）



注 这里的 TPnOV0 和 TPnOV1 标志是建立于内部 RAM 上的变量，可由软件任意进行设置。

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

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

<3> 产生了溢出。但不进行任何软件处理。

<4> 读取 TPnCCR0 寄存器。

读取 TPnOV0 标志。如果 TPnOV0 标志为 1，将该标志清零。

由于 TPnOV0 标志位为 1，脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。

<5> 读取 TPnCCR1 寄存器。

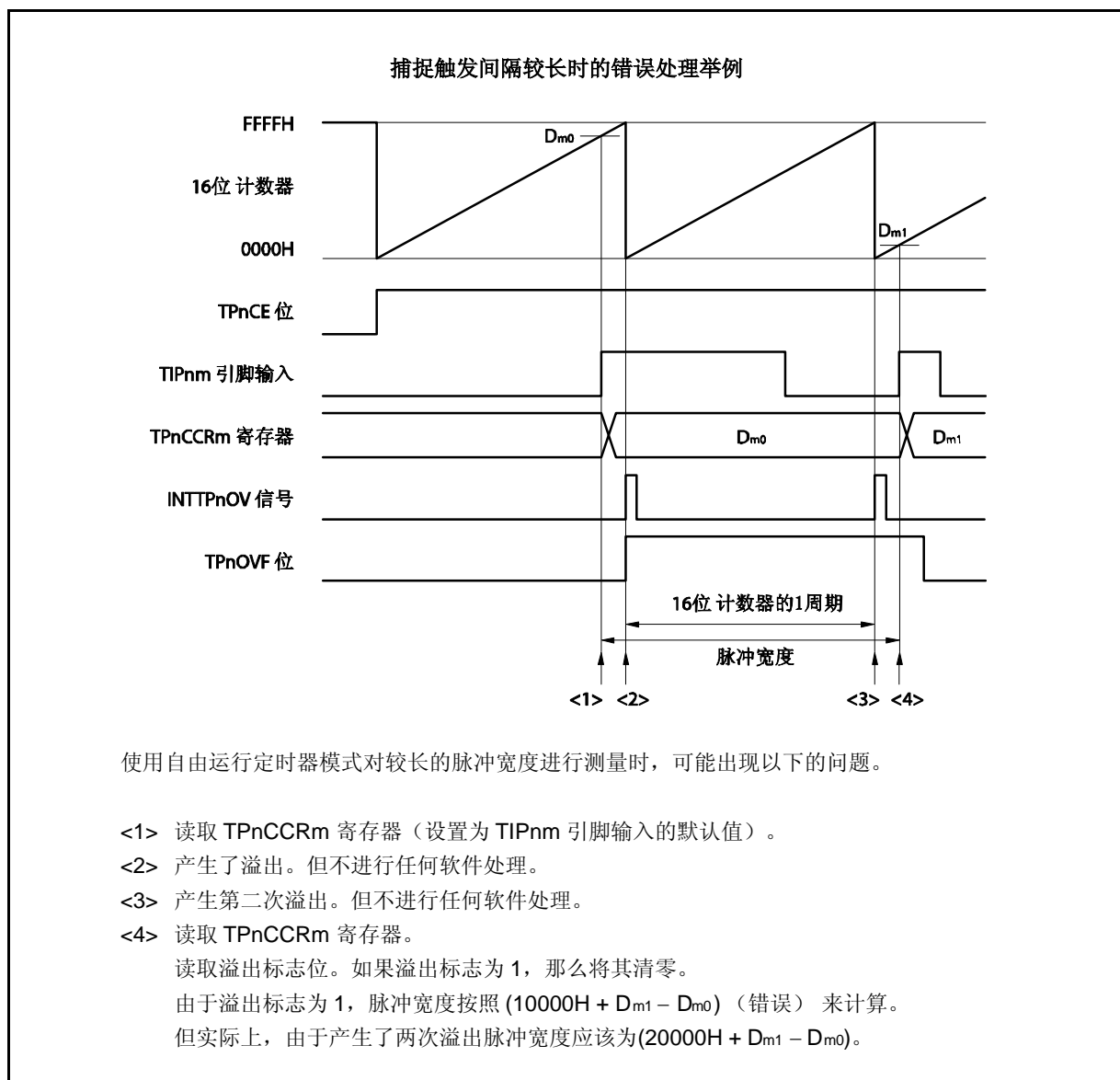
读取 TPnOV1 标志。如果 TPnOV1 标志为 1，那么将其清零。（步骤<4>中将 TPnOV0 位清零但 TPnOV1 位始终保持为 1）。

由于 TPnOV1 标志为 1，脉冲宽度按照 $(10000H + D_{11} - D_{10})$ （正确）来计算。

<6> 同步步骤 <3>

(d) 捕捉触发间隔较长时的溢出处理

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



使用自由运行定时器模式对较长的脉冲宽度进行测量时，可能出现以下的问题。

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

<2> 产生了溢出。但不进行任何软件处理。

<3> 产生第二次溢出。但不进行任何软件处理。

<4> 读取 TPnCCRm 寄存器。

读取溢出标志位。如果溢出标志为 1，那么将其清零。

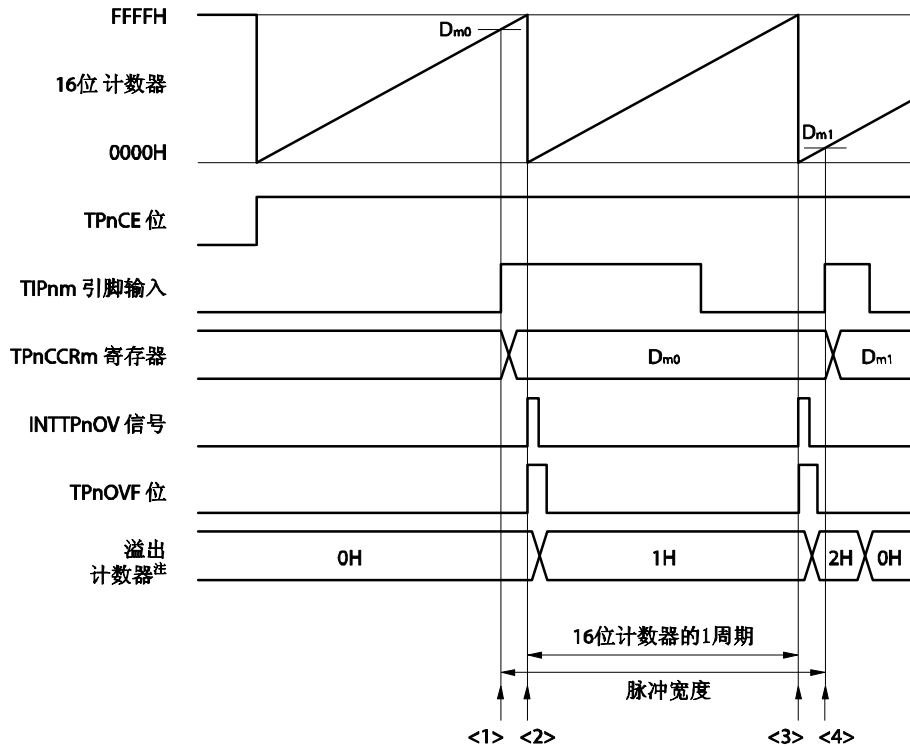
由于溢出标志为 1，脉冲宽度按照 $(10000H + D_{m1} - D_{m0})$ （错误）来计算。

但实际上，由于产生了两次溢出脉冲宽度应该为 $(20000H + D_{m1} - D_{m0})$ 。

如果捕捉触发间隔时间较长以至产生两次或更多次溢出，那么计算结果可能为错误的脉冲宽度。

因此，在捕捉触发间隔时间较长时，请减慢 16 位计数器的计数时钟以延长计数周期或者通过软件的方式避免错误的产生。下面给出一个使用软件方式避免错误的例子。

捕捉触发间隔较长时的处理举例



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

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

<2> 产生了溢出。在溢出中断服务程序中将溢出次数计数器加 1 并将溢出标志清零。

<3> 产生了第二次溢出。在溢出中断服务程序中将溢出次数计数器加 1 并将溢出标志清零。

<4> 读取 TPnCCRm 寄存器。

读取溢出次数计数器的值。

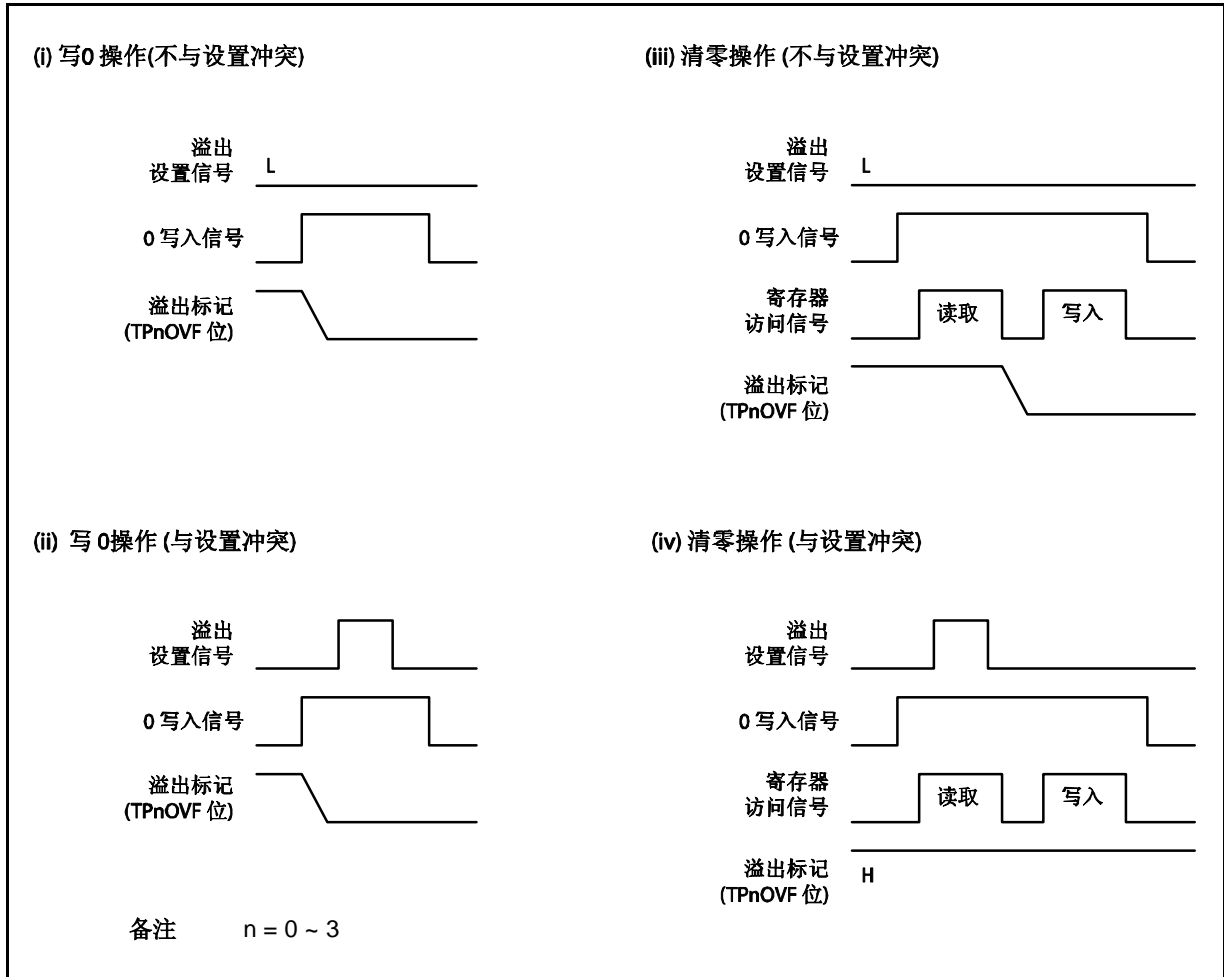
→ 假设此时溢出计数器的值为“N”，则脉冲宽度可按 $(N \times 10000H + D_{m1} - D_{m0})$ 计算求得。

在本例中，由于溢出产生了两次，所以脉冲宽度为 $(20000H + D_{m1} - D_{m0})$ 。

将溢出次数计数器清零(0H)。

(e) 将溢出标志位清零

可以通过使用 CLR 指令清零 TPnOVF 位或向 TPnOPT0 寄存器写入 8 位数据（第 0 位为 0）的方法将溢出标志位清零。为了能够准确检测到溢出的发生，请务必在每次读取到 TPnOVF 位为 1 后，立即使用位操作指令将该位清零。



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

当通过 CLR 指令使溢出标志清为 0 时，如果 CLR 指令的执行与溢出产生的时刻发生冲突，那么即使执行了清零指令，溢出标志保持设置。

6.5.7 脉宽测量模式 (TPnMD2 ~ TPnMD0 位 = 110)

在脉宽测量模式中，TPnCTL0.TPnCE 位被置 1 后，16 位定时器/事件计数器 P 开始计数。每当检测到 TIPnm 引脚输入信号的有效沿时，16 位计数器的计数值就会被保存到 TPnCCRm 寄存器中，同时计数器清零为 0000H。

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

可以选择 TIPn0 或 TIPn1 引脚中的任意一个作为捕捉触发输入引脚。通过 TPnIOC1 寄存器将未使用的引脚设置为“无边沿检测”。

当使用外部时钟来测量脉宽时，只能使用 TIPn1 引脚作为捕捉触发输入引脚，因为外部时钟输入已经固定为 TIPn0 引脚。此时要将 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位清零（捕捉触发输入 (TIPn0 引脚)：无边沿检测）。

图 6-34. 脉宽测量模式的配置图

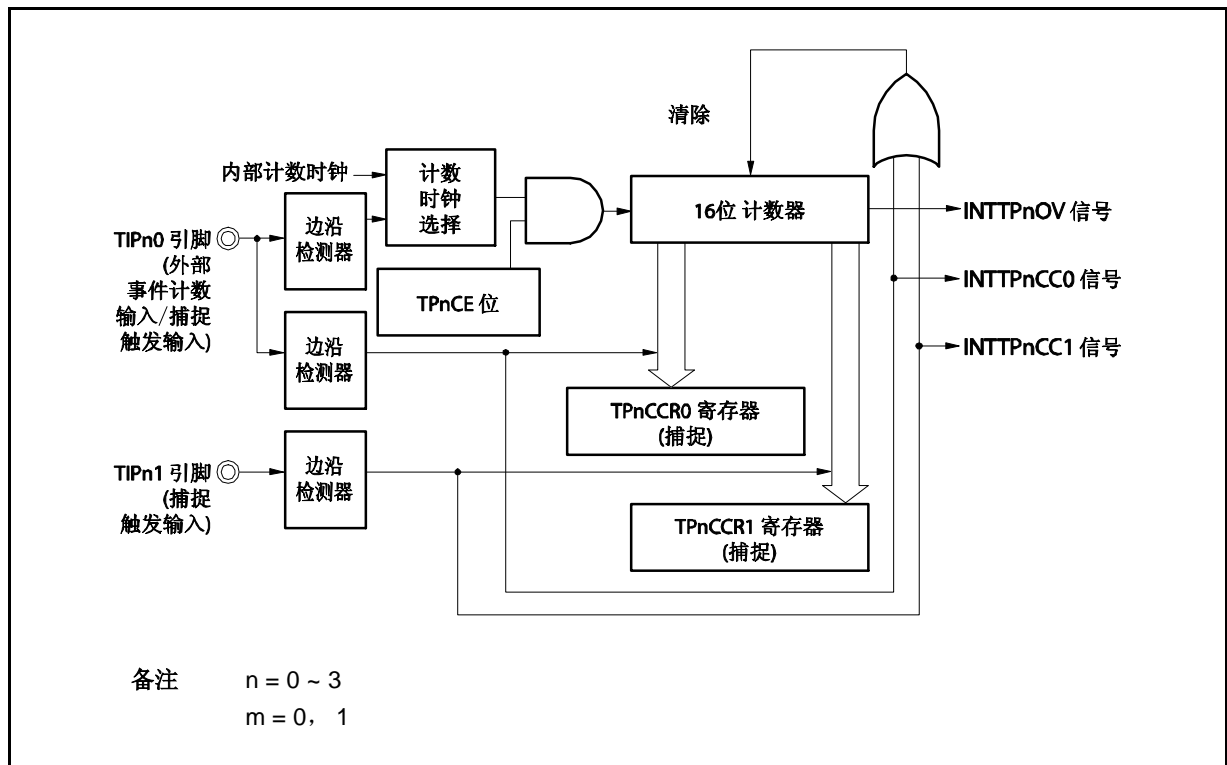
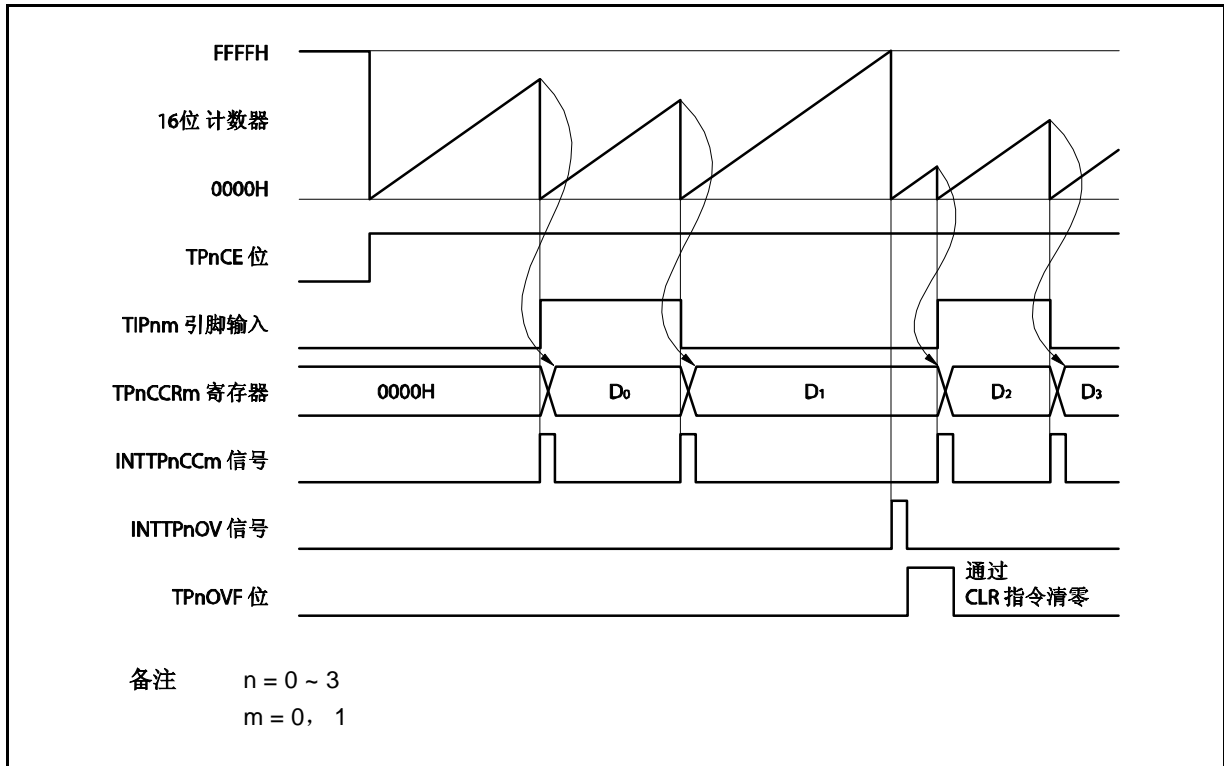


图 6-35. 脉宽测量模式的基本时序



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

脉冲宽度可由下式求得。

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

如果直到 16 位计数器计数到 FFFFH 仍没有检测到 TIPnm 引脚的有效沿，那么将在下一个时钟时产生一个溢出中断请求信号(INTTPnOV)，同时计数器被清零并重新开始计数。这种情况下，溢出标志位 (TPnOPT0.TPnOVF 位) 也将被设置为 1。请通过软件使用 CLR 指令将溢出标志清零。

如果溢出标志被设置为 1，那么脉冲宽度可由下式求得。

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

备注 $n = 0 \sim 3$
 $m = 0, 1$

图 6-36. 脉宽测量模式的寄存器设置 (1/2)

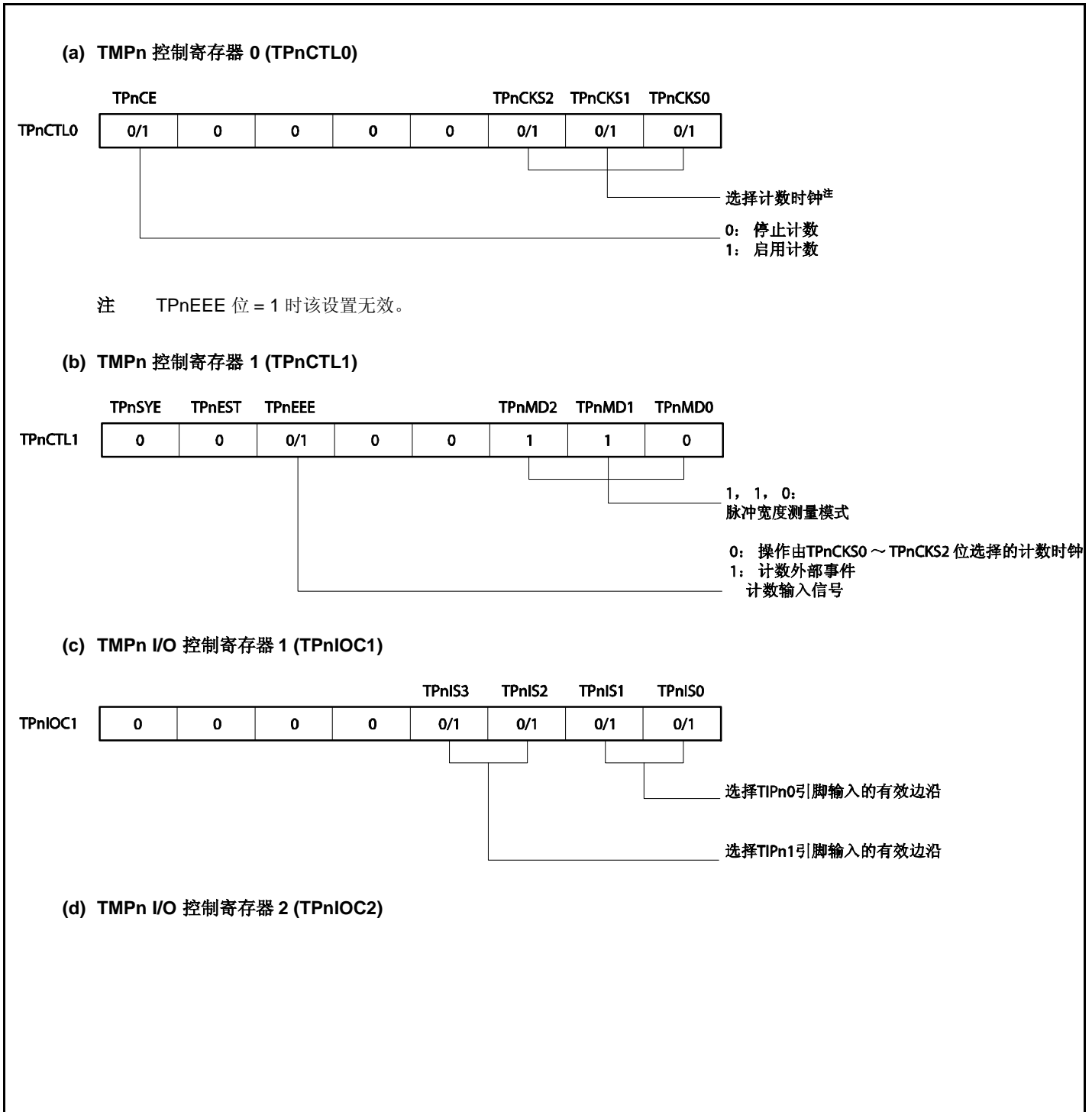


图 6-36. 脉宽测量模式的寄存器设置 (2/2)

(e) TMPn 选项寄存器 0 (TPnOPT0)

(f) TMPn 计数器读取缓冲寄存器 (TPnCNT)

可以通过读取 TPnCNT 寄存器来读取 16 位计数器的计数值。

(g) TMPn 捕捉/比较寄存器 0 和 1 (TPnCCR0 和 TPnCCR1)

当检测到 TIPnm 引脚的有效沿时，16 位计数器的计数值将被保存在这些寄存器中。

备注

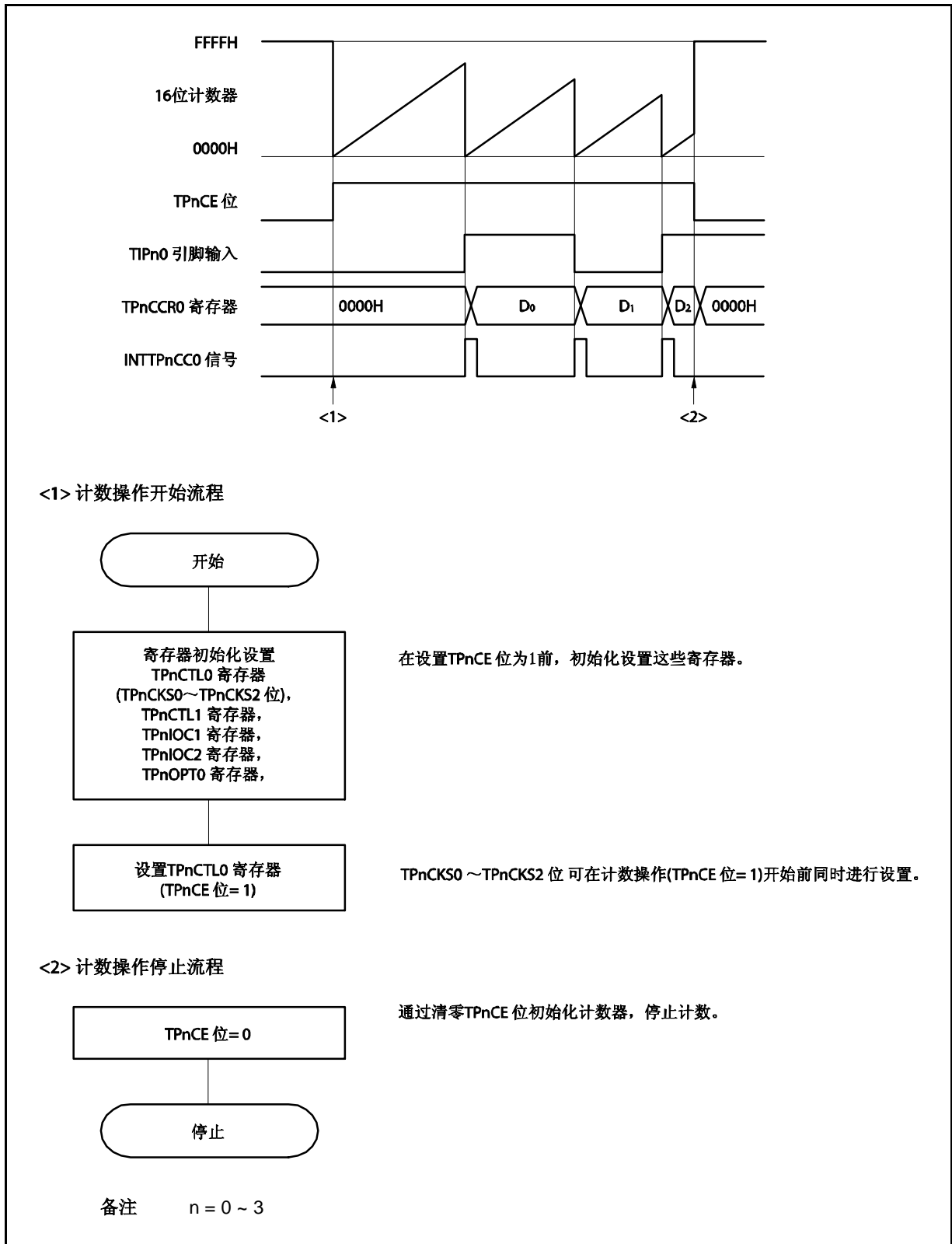
1. 脉宽测量模式下，并未使用 TMPn I/O 控制寄存器 0 (TPnIOC0)。

2. $n = 0 \sim 3$

$m = 0, 1$

(1) 脉宽测量模式的操作流程

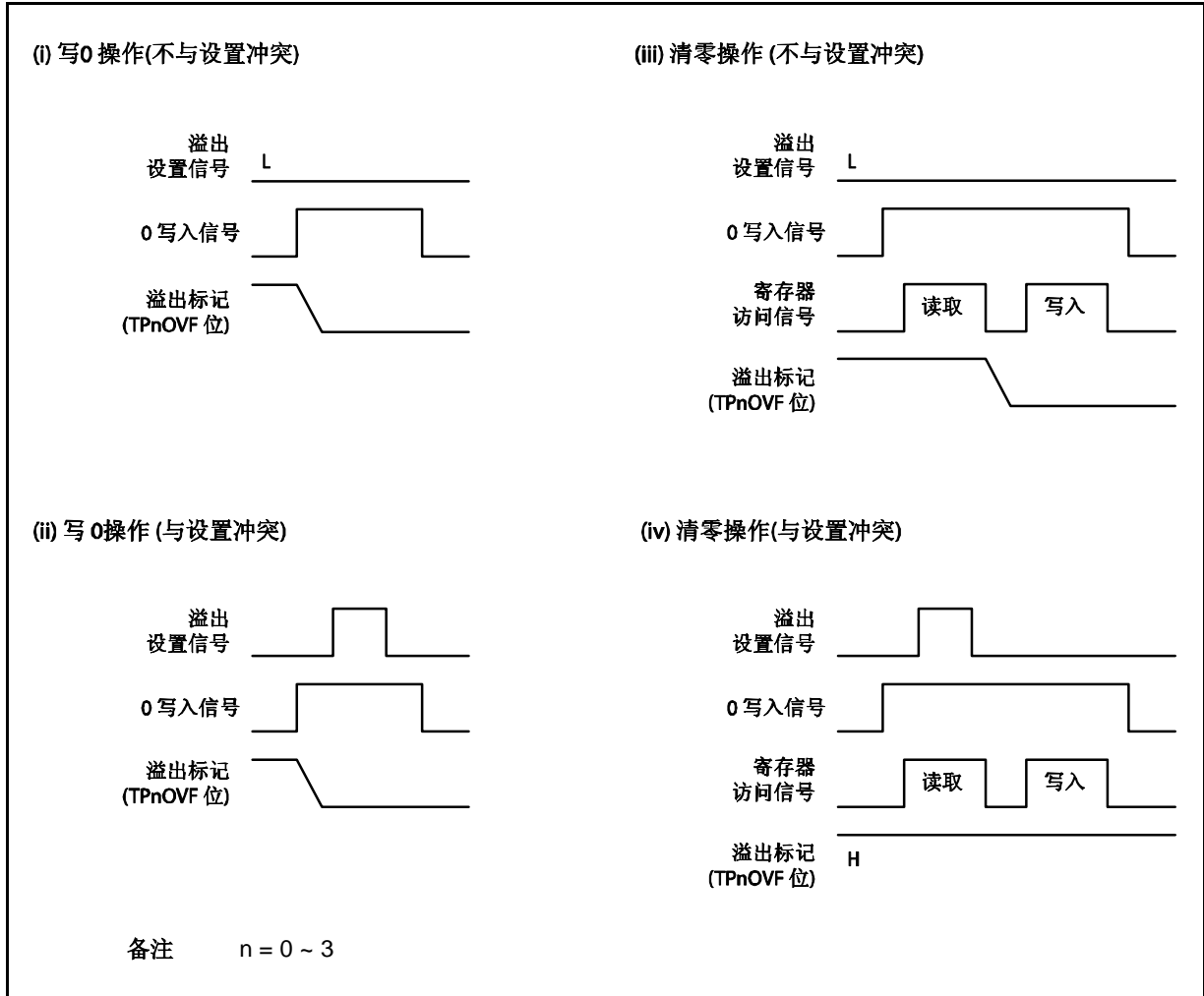
图 6-37. 脉宽测量模式的软件处理流程



(2) 脉宽测量模式的操作时序

(a) 将溢出标志位清零

可以通过使用 CLR 指令清零 TPnOVF 位或向 TPnOPT0 寄存器写入 8 位数据（第 0 位为 0）的方法将溢出标志位清零。为了能够准确检测溢出的发生，请务必在每次读取到 TPnOVF 位为 1 后，立即使用位操作指令将该位清零。



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

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

6.5.8 定时器输出操作

TOPn0 和 TOPn1 引脚的工作模式与输出电平的关系如下表所示。

表 6-4. 各模式下的定时器输出控制

操作模式	TOPn1 引脚	TOPn0 引脚
间隔定时器模式	方波输出	
外部事件计数模式	方波输出	—
外部触发脉冲输出模式	外部触发脉冲输出	方波输出
单脉冲输出模式	单脉冲输出	
PWM 输出模式	PWM 输出	
自由运行定时器模式	方波输出（仅限使用比较功能时）	
脉宽测量模式	—	

备注 n = 0 ~ 3

表 6-5. TOPn0 和 TOPn1 引脚在定时器输出控制位控制下的真值表

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	计数前为低电平，计数过程中为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平，计数过程中为低电平

备注 n = 0 ~ 3
m = 0, 1

6.6 定时器调谐操作功能

定时器 P 及定时器 Q 具有定时器调谐操作功能。

在表 6-6 中所列可被同步操作的定时器。

表 6-6. 定时器的调谐操作模式

主定时器	从定时器	
TMP0	TMP1	-
TMP2	TMP3	TMQ0

- 注意事项**
- 通过 **TPmCTL1.TPmSYE** 和 **TQ0CTL1.TQ0SYE** 位可以启用或禁止调谐操作模式，对于 **TMP2**，**TMP3** 和 **TMQ0** 之一或者两者都可以指定为从定时器。
 - 使用以下程序设置调谐操作模式。
 - <1> 设置从定时器的 **TPmCTL1.TPmSYE** 和 **TQ0CTL1.TQ0SYE** 位以启动调谐操作。
设置从定时器的 **TPmCTL1.TPmMD2** 为 **TPmCTL1.TPmMD0** 以及 **TQ0CTL1.TQ0MD2** 为 **TQ0CTL1.TQ0MD0** 位以进入自由运行模式。
 - <2> 通过使 **TPnCTL1.TPnMD2** 位为 **TPnCTL1.TPnMD0** 位可以设置定时器模式。
此时，不可以设置主定时器的 **TPnCTL1.TPnSYE** 位。
 - <3> 设置主和从定时器的比较寄存器值。
 - <4> 设置从定时器的 **TPmCTL0.TPmCE** 和 **TQ0CTL0.TQ0CE** 以启动对内部运行时钟的操作。
 - <5> 设置主定时器的 **TPnCTL0.TPnCE** 位以启动对内部运行时钟的操作。

备注 m = 1, 3
 n = 0, 2

表 6-7 和 6-8 显示可被用于调谐操作模式的定时器模式 (√: 可设置, ×: 不可设置)。

表 6-7. 可被用于调谐操作模式的定时器模式

主定时器	自由运行模式	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	自由运行模式		PWM 模式		三角波 PWM 模式	
			关闭	开启	关闭	开启	关闭	开启
通道 0	TMP0 (主)	TOP00	PPG		反转		N/A	
		TOP01	PPG		PWM		N/A	
	TMP1 (从)	TOP10	PGP		反转	PWM	N/A	
		TOP11	PPG		PWM		N/A	
通道 1	TMP2 (主)	TOP20	PPG		反转		N/A	
		TOP21	PPG		PWM		N/A	
	TMP3 (从)	TOP30	PPG		反转	PWM	N/A	
		TOP31	PPG		PWM		N/A	
	TMQ0 (从)	TOQ00	PPG		反转	PWM	反转	N/A
		TOQ01 ~ TOQ03	PPG		PWM		三角波 PWM	N/A

备注 从主定时器的比较寄存器到从定时器的比较寄存器的传输数据定时如下。

PPG: CPU 写入时序

反转, PWM, 三角波 PWM: 定时器计数器和比较寄存器与 TOPn0 和 TOQ00 相等的时序 (n = 0 ~ 3)

调谐操作图像 (TMP2, TMP3, TMQ0)

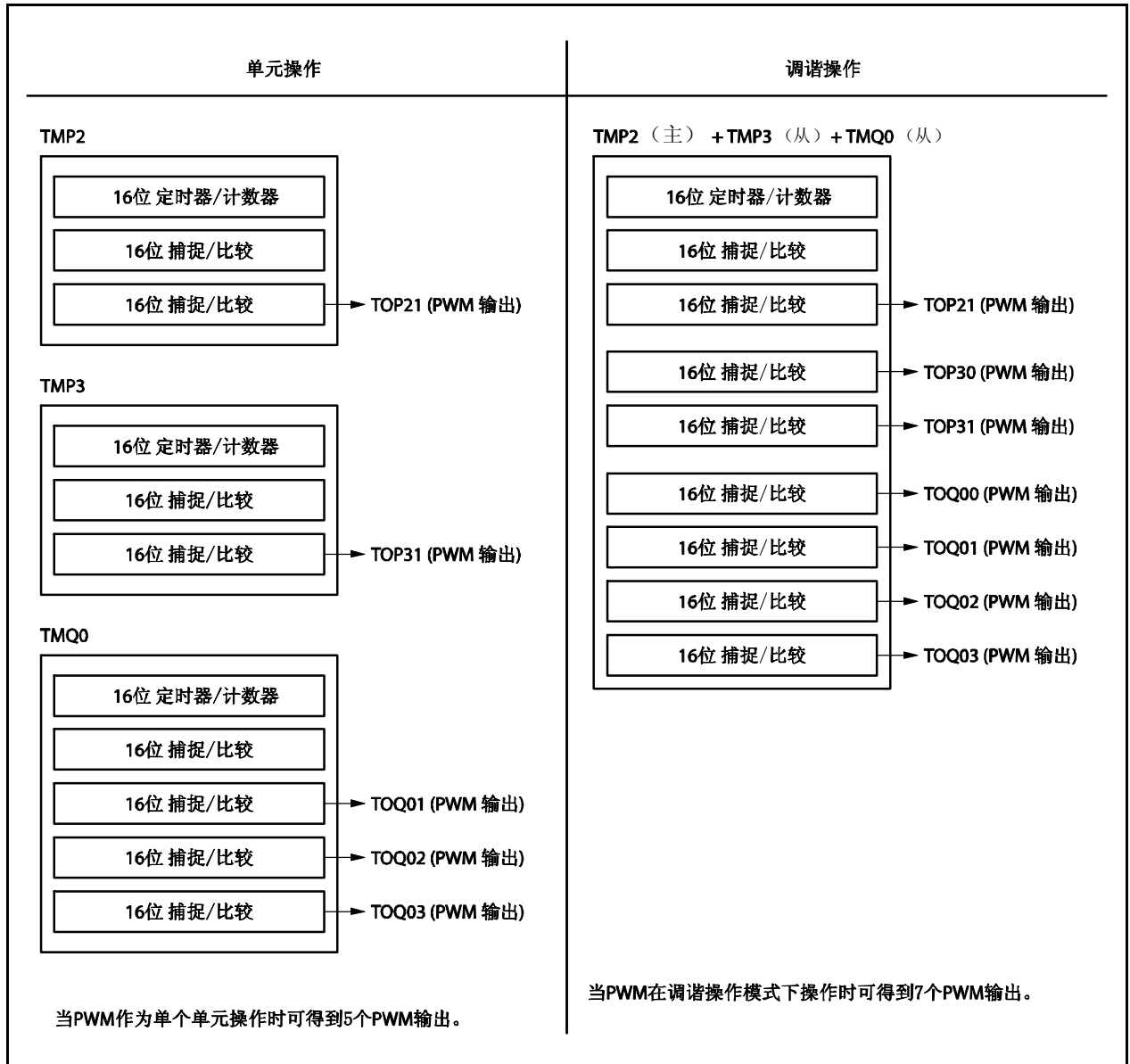
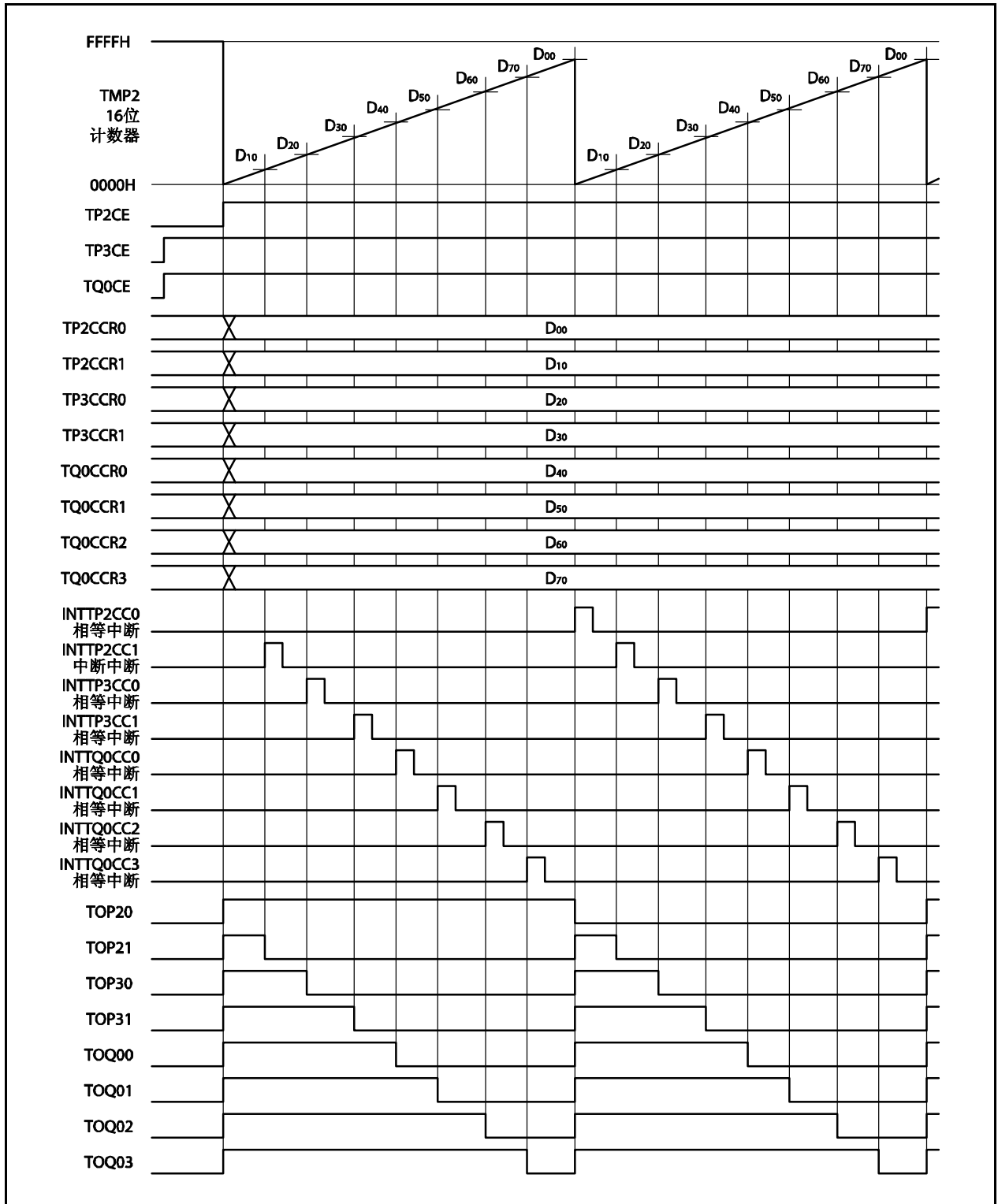


图 6-39. 调谐 PWM 功能的基本操作时序 (TMP2, TMP3, TMQ0)



6.7 选择器功能

在 V850ES/HG2 系列产品中，端口的复用引脚和外围 I/O 器件（TMP，TMM0，或 UARTA）可被用于选择 TMP 的捕捉触发输入。

通过该功能可以实现以下操作。

- TMP1 的 TIP10 和 TIP11 输入信号可在端口/定时器复用引脚（TIP10 和 TIP11 引脚）和 UARTA 接收复用引脚（RXDA0 和 RXDA1）之间进行选择。
→如果选择了 UARTA0 或 UARTA1 的 RXDA0 或 RXDA1 信号作为 TIP10 和 TIP11 输入信号，那么就可以对 UARTA LIN 接收传送波特率误差进行计算了。
- TMP0 的 TIP01 输入信号可在端口/定时器复用功能引脚（TIP01 引脚）信号和 TMM0 的 INTTM0EQ0 信号之间进行选择。

- 注意事项**
1. 当使用选择器功能时，要在连接定时器之前设置 TMP 的捕捉触发输入。
 2. 设置选择器功能之前，要先禁止被连接的周边 I/O 器件（TMP，TMM0，或 UARTA）。

选择器功能的捕捉输入可由以下寄存器指定。

(1) 选择器操作控制寄存器 0 (SELCNT0)

SELCNT0 寄存器是用来选择 TMP0, TMP1 和 TMP 3 的捕捉触发信号的 8 位寄存器。

可以对该寄存器进行 8 位或 1 位的读写操作。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF308H

	7	6	5	4	3	2	1	0
SELCNT0	0	0	ISEL05	ISEL04	ISEL03	ISEL02	0	0

ISEL05	TIP30 输入信号(TMP3)的选择
0	TIP30引脚输入
1	RXDA2 引脚输入

ISEL04	TIP11 输入信号(TMP1)的选择
0	TIP11 引脚输入
1	RXDA1 引脚输入

ISEL03	TIP10 输入信号(TMP1)的选择
0	TIP10引脚输入
1	RXDA0 引脚输入

ISEL02 ^注	TIP01 输入信号(TMP0)的选择
0	TIP01 引脚输入
1	TMM0的INTTMOEQ0 中断

注 在以下条件下使用 INTTMOEQ0 中断信号作为 TIP01 输入信号。

TMM0 操作时钟 ≥ TMP0 操作时钟 × 4

注意事项 1. 若将 ISEL02~ISEL05 位设置为 1, 则请将这些位所对应的引脚模式设置为捕捉输入模式。

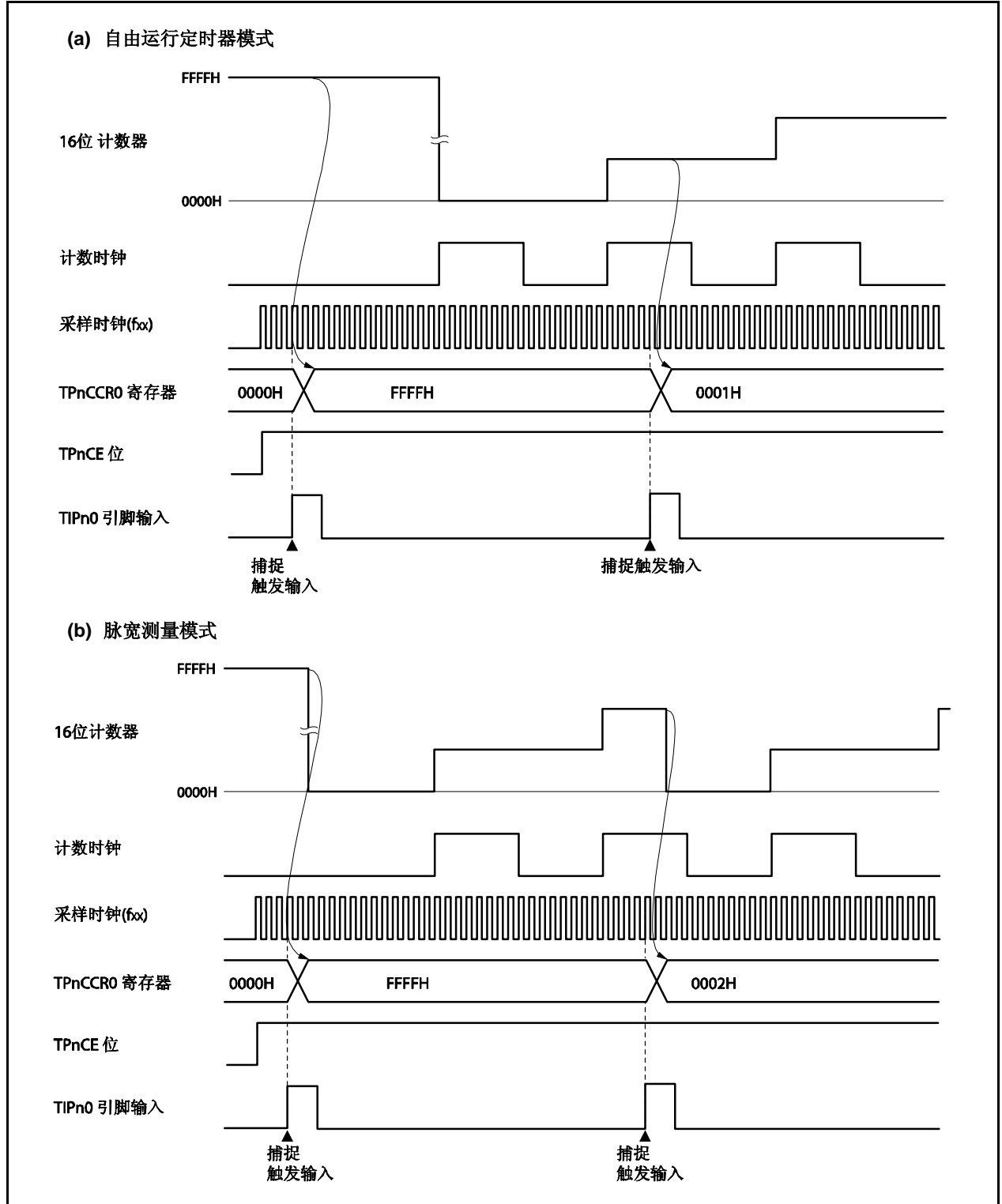
2. 当 TMP0, TM1, TM3, TMM0, 和 UARTA0 ~ UARTA2 停止时, 再设置 ISEL02 ~ ISEL05 位。

<R>

6.8 注意事项

(1) 捕捉操作

若在捕捉操作时使用了一个较慢的时钟信号作为计数时钟，当 TPnCE 位被设置为 1 后，立刻检测到捕捉触发信号时，TPnCCR0 和 TPnCCR1 寄存器中捕捉的计数值将不会是 0000H 而是 FFFFH。



第七章 16 位定时器/事件计数器 Q (TMQ)

定时器 Q (TMQ) 是一个 16 位的定时器/事件计数器。
V850ES/HG2 具有 TMQ0 和 TMQ1。

7.1 概述

以下是 TMQ0 的概况。

- 时钟选择: 8 通道
- 捕捉/触发输入引脚: 4 个
- 外部事件计数输入引脚: 1 个
- 外部触发输入引脚: 1 个
- 定时器/计数器: 1 通道
- 捕捉/比较寄存器: 4 个
- 捕捉/比较相等中断请求信号: 4 个
- 定时器输出引脚: 4 个

备注 n = 0, 1

7.2 功能

TMQn 具有以下功能。

- 间隔定时器
- 外部事件计数器
- 外部触发脉冲输出
- 单脉冲输出
- PWM 输出
- 自由运行定时器
- 脉宽测量
- 三角波 PWM 输出
- 定时器调谐操作功能

备注 n = 0, 1

7.3 配置

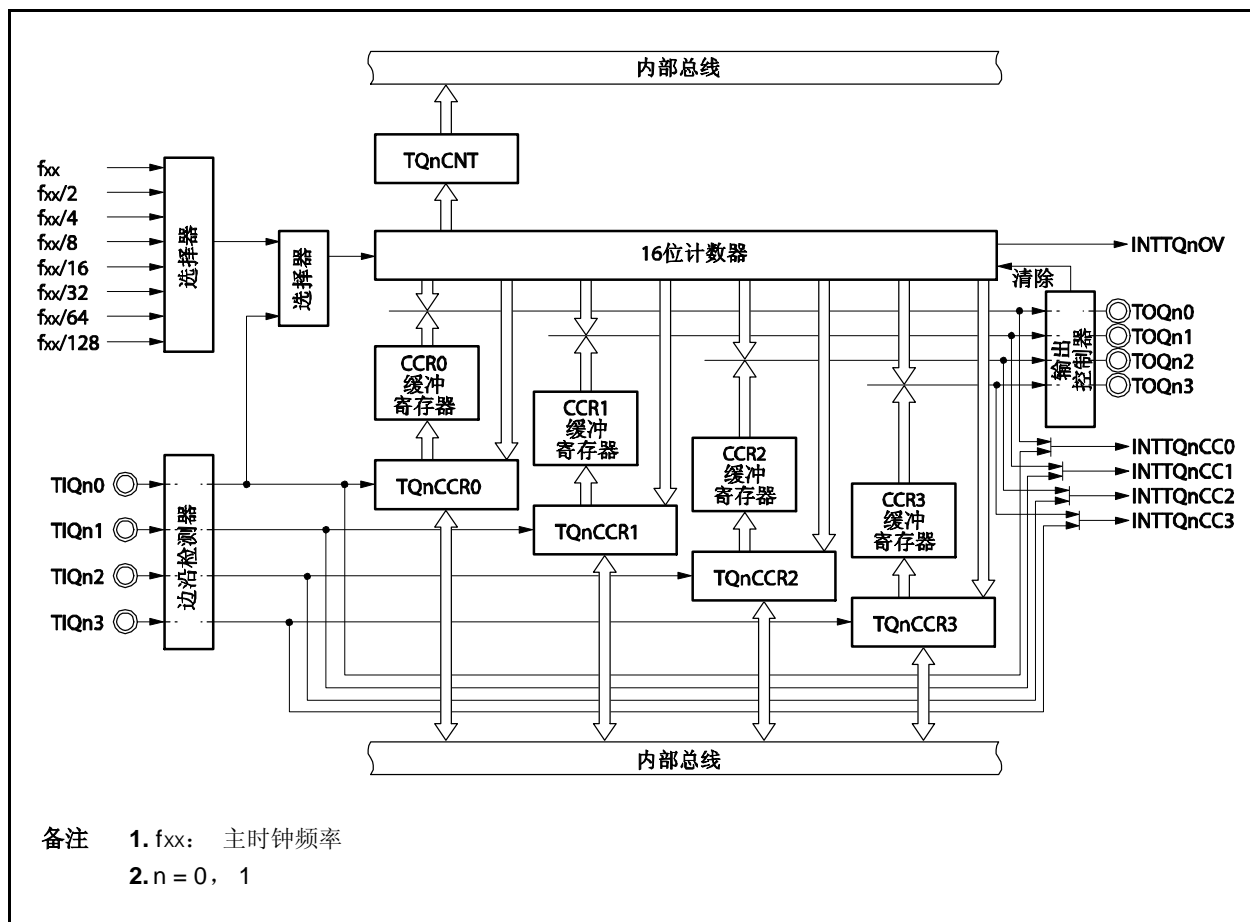
TMQ0 和 TMQ1 包含以下硬件。

表 7-1. TMQ0 和 TMQ1 的配置

器件	配置
定时器寄存器	16 位计数器
寄存器	TMQ0 捕捉/比较寄存器 0 ~ 3 (TQ0CCR0 ~ TQ0CCR3) TMQ0 计数器读取缓冲寄存器 (TQ0CNT) CCR0 ~ CCR3 缓冲寄存器
定时器输入	4 (TIQn0 ^{#1} ~ TIQn3 引脚)
定时器输出	4 (TOQn0 ~ TOQn3 引脚)
控制寄存器 ^{#2}	TMQn 控制寄存器 0, 1 (TQnCTL0, TQnCTL1) TMQn I/O 控制寄存器 0 ~ 2 (TQnIOC0 ~ TQnIOC2) TMQn 选项寄存器 0 (TQnOPT0)

- 注
1. TIQn0 引脚的复用功能包括捕捉触发输入信号，外部事件计数输入信号和外部触发输入信号。
 2. 使用 TIQn0 ~ TIQn3 和 TOQn0 ~ TOQn3 引脚功能时，请参照表 4-15 端口引脚作为复用引脚使用。

图 7-1. TMQ0 和 TMQ1 的框图



(1) 16 位计数器

该 16 位计数器可以对内部时钟或外部事件进行计数。

可以使用 TQnCNT 寄存器读取该计数器的计数值。

当 TQnCTL0.TQnCE 位 = 0 时，16 位计数器的值为 FFFFH。如果此时对 TQnCNT 寄存器进行读取，则读取值将为 0000H。

TQnCE 位复位后的值为 0。因此，16 位计数器也会同时被置为 FFFFH。

(2) CCR0 缓冲寄存器

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

当 TQnCCR0 寄存器被作为比较寄存器使用时，向 TQnCCR0 寄存器写入的数据将被传送到 CCR0 缓冲寄存器。如果 16 位计数器的计数值与 CCR0 缓冲寄存器的值相等，会产生一个比较相等中断请求信号 (INTTQnCC0)。

CCR0 缓冲寄存器不能直接被读写。

复位后 CCR0 缓冲寄存器和 TQnCCR0 寄存器都将被清为 0000H。

(3) CCR1 缓冲寄存器

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

当 TQnCCR1 寄存器被作为比较寄存器使用时，向 TQnCCR1 寄存器写入的数据将被传送到 CCR1 缓冲寄存器。如果 16 位计数器的计数值与 CCR1 缓冲寄存器的值相等，会产生一个比较相等中断请求信号 (INTTQnCC1)。

CCR1 缓冲寄存器不能直接被读写。

复位后 CCR1 缓冲寄存器和 TQnCCR1 寄存器都将被清为 0000H。

(4) CCR2 缓冲寄存器

CCR2 寄存器是一个用于比较 16 位计数器的计数值的 16 位比较寄存器。

当 TQnCCR2 寄存器被作为比较寄存器使用时，向 TQnCCR2 寄存器写入的数据将被传送到 CCR2 缓冲寄存器。如果 16 位计数器的计数值与 CCR2 缓冲寄存器的值相等，会产生一个比较相等中断请求信号 (INTTQnCC2)。

CCR2 缓冲寄存器不能直接被读写。

复位后 CCR2 缓冲寄存器和 TQnCCR2 寄存器都将被清为 0000H。

(5) CCR3 缓冲寄存器

CCR3 寄存器是一个用于比较 16 位计数器的计数值的 16 位比较寄存器。

当 TQnCCR3 寄存器被作为比较寄存器使用时，向 TQnCCR3 寄存器写入的数据将被传送到 CCR3 缓冲寄存器。如果 16 位计数器的计数值与 CCR3 缓冲寄存器的值相等，会产生一个比较相等中断请求信号 (INTTQnCC3)。

CCR3 缓冲寄存器不能直接被读写。

复位后 CCR3 缓冲寄存器和 TQnCCR3 寄存器都将被清为 0000H。

(6) 边沿检测器

该电路用于检测 TIQn0 和 TIQn3 引脚输入的有效边沿。通过 TQnIOC1 和 TQnIOC2 寄存器选择无有效沿，上升沿，下降沿或双沿有效。

(7) 输出控制器

该电路用于控制 TOQn0 ~ TOQn3 引脚的输出。输出控制器是由 TQnIOC0 寄存器控制的。

(8) 选择器

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

7.4 寄存器

用于控制 TMQn 的寄存器如下所述。

- TMQn 控制寄存器 0 (TQnCTL0)
- TMQn 控制寄存器 1 (TQnCTL1)
- TMQn I/O 控制寄存器 0 (TQnIOC0)
- TMQn I/O 控制寄存器 1 (TQnIOC1)
- TMQn I/O 控制寄存器 2 (TQnIOC2)
- TMQn 选项寄存器 0 (TQnOPT0)
- TMQn 捕捉/比较寄存器 0 (TQnCCR0)
- TMQn 捕捉/比较寄存器 1 (TQnCCR1)
- TMQn 捕捉/比较寄存器 2 (TQnCCR2)
- TMQn 捕捉/比较寄存器 3 (TQnCCR3)
- TMQn 计数器读取缓冲寄存器 (TQnCNT)

备注 使用 TIQn0~ TIQn3 和 TOQn0~ TOQn3 引脚功能时，请参照表 4-15 端口引脚作为复用引脚使用。

(1) TMQn 控制寄存器 0 (TQnCTL0)

TQnCTL0 寄存器为用于控制 TMQn 操作的 8 位寄存器。

可以对该寄存器进行 8 位或 1 位的读写操作。

该寄存器复位后的值为 00H。

可使用软件向 TQnCTL0 寄存器写入相同的数值。

复位后: 00H R/W 地址: TQ0CTL0 FFFF540H, TQ1CTL0 FFFF610H

TQnCTL0 (n = 0, 1)	7	6	5	4	3	2	1	0
	TQnCE	0	0	0	0	TQnCKS2	TQnCKS1	TQnCKS0

TQnCE	TMQn 操作控制
0	禁用TMQn 操作 (TMQn 异步 ^注 复位)。
1	启用TMQn 操作。TMQn 操作开始。

TQnCKS2	TQnCKS1	TQnCKS0	内部计数时钟选择
0	0	0	fx
0	0	1	fx/2
0	1	0	fx/4
0	1	1	fx/8
1	0	0	fx/16
1	0	1	fx/32
1	1	0	fx/64
1	1	1	fx/128

注 TQnOPT0.TQnOVF 位, 16 位计数器, 定时器输出 (TOQn0 ~ TOQn3 引脚)

- 注意事项
1. 在 TQnCE 位 = 0 时, 设置 TQnCKS2 ~ TQnCKS0 位。
当将 TQnCE 位的值从 0 改变为 1 时, 可同时对 TQnCKS2 ~ TQnCKS0 位进行设置。
 2. 请务必将第 3 到第 6 位清零。

备注 fx: 主时钟频率

(2) TMQn 控制寄存器 1 (TQnCTL1)

TQnCTL1 寄存器是用于控制 TMQn 操作的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

该寄存器复位后的值为 00H。

(1/2)

复位后: 00H R/W 地址: TQ0CTL1 FFFF541H, TQ1CTL1 FFFF611H															
	7 6 5 4 3 2 1 0														
TQnCTL1 (n = 0, 1)	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 12.5%;">TQnSYE</td> <td style="width: 12.5%;">TQnEST</td> <td style="width: 12.5%;">TQnEEE</td> <td style="width: 12.5%;">0</td> <td style="width: 12.5%;">0</td> <td style="width: 12.5%;">TQnMD2</td> <td style="width: 12.5%;">TQnMD1</td> <td style="width: 12.5%;">TQnMD0</td> </tr> </table>	TQnSYE	TQnEST	TQnEEE	0	0	TQnMD2	TQnMD1	TQnMD0						
TQnSYE	TQnEST	TQnEEE	0	0	TQnMD2	TQnMD1	TQnMD0								
	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">TQnSYE</td> <td style="text-align: center;">调谐操作模式启用控制</td> </tr> <tr> <td style="text-align: center;">0</td> <td>独立操作模式(异步操作模式)</td> </tr> <tr> <td style="text-align: center;">1</td> <td>调谐操作模式(从操作规格) 在该模式下, 定时器P 能够与主定时器同步操作。</td> </tr> <tr> <td></td> <td style="text-align: center;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center;">主定时器</td> <td style="width: 50%; text-align: center;">从定时器</td> </tr> <tr> <td style="text-align: center;">TMP2</td> <td style="text-align: center;">TMP3 TMQ0</td> </tr> </table> </td> </tr> <tr> <td></td> <td>对于调谐操作模式, 参见 7.6 定时器调谐操作功能。</td> </tr> </table>	TQnSYE	调谐操作模式启用控制	0	独立操作模式(异步操作模式)	1	调谐操作模式(从操作规格) 在该模式下, 定时器P 能够与主定时器同步操作。		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center;">主定时器</td> <td style="width: 50%; text-align: center;">从定时器</td> </tr> <tr> <td style="text-align: center;">TMP2</td> <td style="text-align: center;">TMP3 TMQ0</td> </tr> </table>	主定时器	从定时器	TMP2	TMP3 TMQ0		对于调谐操作模式, 参见 7.6 定时器调谐操作功能。
TQnSYE	调谐操作模式启用控制														
0	独立操作模式(异步操作模式)														
1	调谐操作模式(从操作规格) 在该模式下, 定时器P 能够与主定时器同步操作。														
	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center;">主定时器</td> <td style="width: 50%; text-align: center;">从定时器</td> </tr> <tr> <td style="text-align: center;">TMP2</td> <td style="text-align: center;">TMP3 TMQ0</td> </tr> </table>	主定时器	从定时器	TMP2	TMP3 TMQ0										
主定时器	从定时器														
TMP2	TMP3 TMQ0														
	对于调谐操作模式, 参见 7.6 定时器调谐操作功能。														
	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">TQnEST</td> <td style="text-align: center;">软件触发控制</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">-</td> </tr> <tr> <td style="text-align: center;">1</td> <td>发生一有效信号给外部触发输入 · 单脉冲输出模式: 作为触发写入1至 TQnEST 位, 输出一单脉冲。 · 外部触发g脉冲输出模式: 作为触发写入1至 TQnEST 位, 输出一PWM 方波。</td> </tr> </table>	TQnEST	软件触发控制	0	-	1	发生一有效信号给外部触发输入 · 单脉冲输出模式: 作为触发写入1至 TQnEST 位, 输出一单脉冲。 · 外部触发g脉冲输出模式: 作为触发写入1至 TQnEST 位, 输出一PWM 方波。								
TQnEST	软件触发控制														
0	-														
1	发生一有效信号给外部触发输入 · 单脉冲输出模式: 作为触发写入1至 TQnEST 位, 输出一单脉冲。 · 外部触发g脉冲输出模式: 作为触发写入1至 TQnEST 位, 输出一PWM 方波。														

注意事项 1. TQnEST 位仅在外外部触发脉冲输出模式下或单脉冲输出模式下有效。在其它模式下, 对该位的置 1 操作将被忽略。
2. 请务必将第 3 和 4 位清零。

TQnEEE	计数时钟选择
0	禁用外部事件计数输入操作。 (通过 TQnCTL0.TQnCK0~ TQnCK2 位选择计数时钟进行计数操作)
1	启用外部事件计数输入操作。 (在外部事件计数输入信号的有效边沿进行计数。)

TQnEEE 位选择用内部计数时钟进行计数操作 或外部事件计数输入的有效边沿。

TQnMD2	TQnMD1	TQnMD0	定时器模式选择
0	0	0	间隔定时器模式
0	0	1	外部时间计数模式
0	1	0	外部触发脉冲输出模式
0	1	1	单模式输出模式
1	0	0	PWM 输出模式
1	0	1	自由运行定时器模式
1	1	0	脉冲宽度测量模式
1	1	1	三角波PWM 模式

- 注意事项
1. 在外部事件计数模式下，无论 TQnEEE 位如何设置，外部事件计数输入都是被选中的。
 2. 只有在 TQnCTL0.TQnCE 位 = 0 时才可以对 TQnEEE 以及 TQnMD2 ~ TQnMD0 位进行设置（TQnCE 位 = 1 时，可以对这些位写入相同的值）。如果在 TQnCE 位 = 1 时对这些位进行了改写，那么操作结果将不能被保证。若由于误操作引起了改写，则要将 TQnCE 位清零然后再次设置这些寄存器位。

(2/2)

<R>

(3) TMQn I/O 控制寄存器 0 (TQnIOC0)

TQnIOC0 寄存器是用于控制定时器输出的 8 位寄存器 (TOQn0 ~ TOQn3 引脚)。

可对该寄存器进行 8 位或 1 位的读写操作。

该寄存器复位后的值为 00H。

复位后: 00H		R/W	地址: TQ0IOC0 FFFF542H, TQ1IOC0 FFFF612H							
TQnIOC0 (n = 0, 1)			7	6	5	4	3	2	1	0
			TQnOL3	TQnOE3	TQnOL2	TQnOE2	TQnOL1	TQnOE1	TQnOL0	TQnOE0

TQnOLm	TOQnm 引脚输出电平设置 (m = 0 ~ 3) ^注
0	TOQnm 引脚从高电平开始
1	TOQnm 引脚从低电平开始

TQnOEm	TOQnm 引脚输出设置 (m = 0 ~ 3)
0	禁用定时器输出 · 当 TQnOLm 位 = 0: 低电平从 TOQnm 引脚输出 · 当 TQnOLm 位 = 1: 高电平从 TOQnm 引脚输出
1	启用定时器输出 (从 TOQnm 引脚输出一方波)。

注 定时器输出引脚 (TOQnm) 的输出电平由以下 (TOQnm) 位指定。

- 当 TQnOLm 位 = 0
- 当 TQnOLm 位 = 1

16位计数器

TQnCE 位

TOQnm 输出引脚

16位计数器

TQnCE 位

TOQnm 输出引脚

注意事项

1. 只有在 TQnCTL0.TQnCE 位 = 0 时才可以对 TQnOLm 和 TQnOEm 位进行改写 (当 TQnCE 位 = 1 时可以向这些位写入相同的值)。若由于误操作引起了改写, 则要将 TQnCE 位清零然后再次设置这些寄存器位。
2. 当 TQnCE 和 TQnOEm 位为 0 时, 即使对 TQnOLm 位进行了操作, TOQnm 引脚的输出电平也不能确定。

备注 m = 0 ~ 3

(4) TMQn I/O 控制寄存器 1 (TQnIOC1)

TQnIOC1 寄存器是用于控制捕捉触发输入信号 (TIQn0 ~ TIQn3 引脚) 有效沿的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: TQ0IOC1 FFFF543H, TQ1IOC1 FFFF613H

TQnIOC1 (n = 0, 1)	7	6	5	4	3	2	1	0
	TQnIS7	TQnIS6	TQnIS5	TQnIS4	TQnIS3	TQnIS2	TQnIS1	TQnIS0

TQnIS7	TQnIS6	捕捉触发输入信号(TIQn3 引脚) 有效边沿设置
0	0	无边沿检测(捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升和下降沿

TQnIS5	TQnIS4	捕捉触发输入信号(TIQn2 引脚) 有效边沿检测
0	0	无边沿检测(捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升和下降沿

TQnIS3	TQnIS2	捕捉触发输入信号(TIQn1 引脚) 有效边沿设置
0	0	无边沿检测(捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升和下降沿

TQnIS1	TQnIS0	捕捉触发输入信号(TIQn0 引脚) 有效边沿设置
0	0	无边沿检测(捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升和下降沿

- 注意事项**
1. 只有在 TQnCTL0.TQnCE 位 = 0 时才可以对 TQnIS7 ~ TQnIS0 位进行改写 (TQnCE 位 = 1 时只可以向这些位写入相同的值)。若由于误操作引起了改写, 则要将 TQnCE 位清零然后再次设置这些寄存器位。
 2. TQnIS7 ~ TQnIS0 位只有在自由运行定时器模式和脉宽测量模式下才有效。其它模式下, 无法进行捕捉操作。

(5) TMQn I/O 控制寄存器 2 (TQnIOC2)

TQnIOC2 寄存器是用于控制外部事件计数输入信号 (TIQn0 引脚) 有效沿和外部触发输入信号 (TIQn0 引脚) 有效沿的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

该寄存器复位后的值为 00H。

复位后: 00H		R/W	地址: TQ0IOC2 FFFFF544H, TQ1IOC2 FFFFF614H					
TQnIOC2 (n = 0, 1)	7	6	5	4	3	2	1	0
	0	0	0	0	TQnEES1	TQnEES0	TQnETS1	TQnETS0

TQnEES1	TQnEES0	外部事件计数输入信号(TIQn0 引脚) 有效边沿设置
0	0	无边沿检测(外部事件计数无效)
0	1	检测 上升沿
1	0	检测 下降沿
1	1	检测 上升和下降沿

TQnETS1	TQnETS0	外部触发输入信号(TIQn0 引脚) 有效边沿设置
0	0	无边沿检测(外部事件计数无效)
0	1	检测 上升沿
1	0	检测 下降沿
1	1	检测 上升和下降沿

注意事项

1. 在 TQnCTL0.TQnCE 位 = 0 时才可对 TQnEES1, TQnEES0, TQnETS1 和 TQnETS0 位进行改写 (TQnCE 位 = 1 时只能向这些位写入相同的值)。若在 TQnCE 位 = 1 由于误操作引起了改写, 则要将 TQnCE 位清零然后再次设置这些寄存器位。
2. TQnEES1 和 TQnEES0 位只有在 TQnCTL1.TQnEEE 位 = 1 或设置了外部事件计数模式 (TQnCTL1.TQnMD2 ~ TQnCTL1.TQnMD0 位 = 001) 时才有效。
3. TQnETS1 和 TQnETS0 位只有在外部触发脉冲输出模式 (TQnCTL1.TQnMD2 ~ TQnCTL1.TQnMD0 位 = 010) 或单脉冲输出模式 (TQnCTL1.TQnMD2 ~ TQnCTL1.TQnMD0 位 = 011) 下才有效。

(6) TMQn 选项寄存器 0 (TQnOPT0)

TQnOPT0 寄存器是用于设置捕捉/比较操作和检测溢出的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: TQ0OPT0 FFFF545H, TQ1OPT0 FFFF615H

	7	6	5	4	3	2	1	0
TQnOPT0 (n = 0, 1)	TQnCCS3	TQnCCS2	TQnCCS1	TQnCCS0	0	0	0	TQnOVF

TQnCCSm	TQnCCRm 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
The TQnCCSm 位设置仅在自由运行定时器模式下有效。	

TQnOVF	TMQn 溢出检测
设置(1)	发生溢出
复位(0)	TQnOVF 位写入 0 或 TQnCTL0.TQnCE 位 = 0
<ul style="list-style-type: none"> • 当 16 位计数器的计数值在自由运行定时器模式下或脉冲宽度测量模式下从 FFFFH ~ 0000H 溢出时, TQnOVF 位设置为 1。 • 中断请求信号 (INTTQnOV) 在 TQnOVF 位设置为 1 的同时发生。INTTQnOV 信号只能在自由运行定时器模式下或脉冲宽度测量模式下发生。 • TQnOVF 即使当 TQnOVF 位或 TQnOPT0 寄存器在 TQnOVF 位 = 1 时被读取, 也不能清除 TQnOVF 位。 • TQnOVF 位可被读取或写入, 单 TQnOVF 位不能通过软件设置为 1。写 1 操作对 TMQn 操作无影响。 	

- 注意事项**
1. 在 TQnCTL0.TQnCE 位 = 0 时才可对 TQnCCS3 和 TQnCCS0 位进行改写 (TQnCE 位 = 1 时只可以向这些位写入相同的值)。若由于误操作引起了改写, 则要将 TQnCE 位清零然后再次设置这些寄存器位。
 2. 请务必将第 1 到 3 位清零。

备注 m = 0 ~ 3

(7) TMQn 捕捉/比较寄存器 0 (TQnCCR0)

根据选择模式的不同，TQnCCR0 寄存器可用作捕捉寄存器或比较寄存器。

该寄存器只有在自由运行模式下根据 TQnOPT0.TQnCCS0 位设置的不同，可以用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TQnCCR0 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。

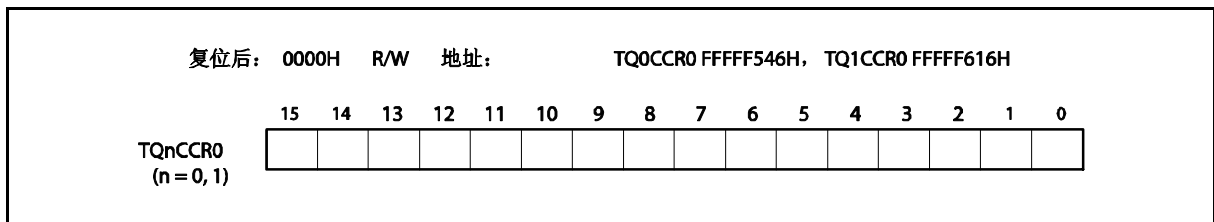
TQnCCR0 寄存器可以在操作过程中进行读写。

可对该寄存器进行 16 位的读写操作。

该寄存器复位后的值为 00H。

注意事项 以下情况下禁止访问 TQnCCR0 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上外围 I/O 寄存器。

- 当 CPU 工作于副时钟 且主时钟振荡停止时
- 当 CPU 工作于内部振荡时钟时



(a) 作为比较寄存器时的功能

TQnCCR0 寄存器可以在 TQnCTL0.TQnCE 位 = 1 时被改写。

向 TQnCCR0 寄存器中设置的数值将被传送到 CCR0 缓冲寄存器中。当 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相等时，会产生一个比较相等中断请求信号(INTTQnCC0)。如果允许了 TOQn0 引脚输出，那么 TOQn0 引脚输出的电平将被反转。

当 TQnCCR0 寄存器在间隔定时器模式，外部事件计数模式，外部触发脉冲输出模式，单脉冲输出模式或三角波 PWM 模式下被用作周期寄存器时，如果 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相等，那么 16 位计数器将被清零(0000H)。

(b) 作为捕捉寄存器时的功能

当 TQnCCR0 寄存器在自由运行定时器模式下被用作捕捉寄存器时，如果检测到了捕捉触发输入引脚（TIQn0 引脚）的有效边沿，那么 16 位计数器的计数值将被保存到 TQnCCR0 寄存器中。在脉宽测量模式下，如果检测到了捕捉触发输入引脚（TIQn0 引脚）的有效边沿，那么 16 位计数器的计数值将被保存到 TQnCCR0 寄存器中并且同时 16 位计数器被清零(0000H)。

即使捕捉操作和读取 TQnCCR0 寄存器操作冲突，仍可正确读出 TQnCCR0 寄存器中的值。

备注 n = 0, 1

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 7-2. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	-
三角波 PWM 模式	比较寄存器	同时写入

(8) TMQn 捕捉/比较寄存器 1 (TQnCCR1)

根据选择模式的不同，TQnCCR1 寄存器可用作捕捉寄存器或比较寄存器。

该寄存器只有在自由运行模式下根据 TQnOPT0.TQnCCS1 位设置的不同，可以用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TQnCCR1 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。

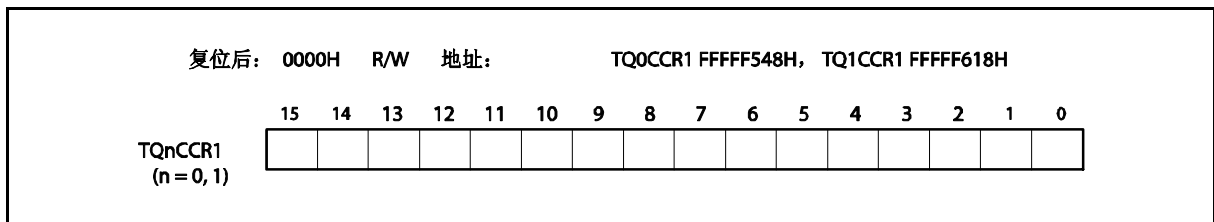
TQnCCR1 寄存器可以在操作过程中进行读写。

可对该寄存器进行 16 位的读写操作。

该寄存器复位后的值为 0000H。

注意事项 以下情况下禁止访问 TQnCCR1 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上外围 I/O 寄存器。

- 当 CPU 工作于副时钟 且主时钟振荡停止时
- 当 CPU 工作于内部振荡时钟时



(a) 作为比较寄存器时的功能

TQnCCR1 寄存器可以在 TQnCTL0.TQnCE 位 = 1 时被改写。

向 TQnCCR1 寄存器中设置的数值将被传送到 CCR1 缓冲寄存器中。当 16 位计数器的值与 CCR1 缓冲寄存器的值相等时，产生比较相等中断请求信号 (INTTQnCC1)。如果此时允许 TOQn1 引脚输出，则 TOQn1 引脚的输出信号的电平将反转。

(b) 作为捕捉寄存器时的功能

在自由运行定时器模式下当 TQnCCR1 寄存器用作捕捉寄存器时，如果检测到捕捉触发输入引脚 (TIQn1 引脚) 的有效沿，那么 16 位计数器的计数值将被保存到 TQnCCR1 寄存器。在脉宽测量模式下，如果检测到捕捉输入引脚 (TIQn1 引脚) 的有效沿，那么 16 位计数器的计数值将被保存到 TQnCCR1 寄存器，同时 16 位计数器清为(0000H)。

即使捕捉操作和读取 TQnCCR1 寄存器的操作产生冲突，仍可正确读出 TQnCCR1 寄存器中的值。

备注 n = 0, 1

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 7-3. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	—
三角波 PWM 模式	比较寄存器	同时写入

(9) TMQn 捕捉/比较寄存器 2 (TQnCCR2)

根据选择模式的不同，TQnCCR2 寄存器可用作捕捉寄存器或比较寄存器。

该寄存器只有在自由运行模式下根据 TQnOPT0.TQnCCS2 位设置的不同，可以用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TQnCCR2 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。

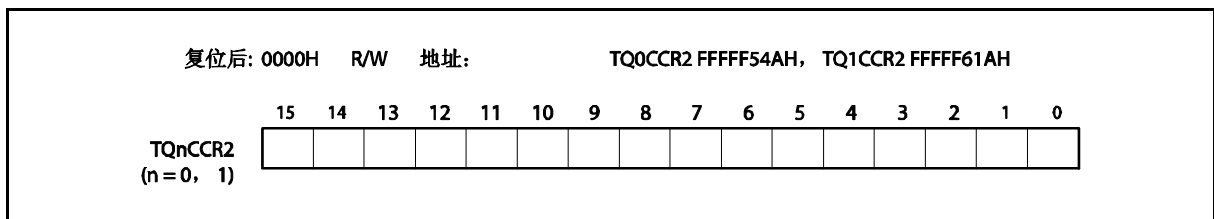
TQnCCR2 寄存器可以在操作过程中进行读写

可对该寄存器进行 16 位的读写操作。

该寄存器复位后的值为 0000H。

注意事项 以下情况下禁止访问 TQnCCR2 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上外围 I/O 寄存器。

- 当 CPU 工作于副时钟 且主时钟振荡停止时
- 当 CPU 工作于内部振荡时钟时



(a) 作为比较寄存器时的功能

TQnCCR2 寄存器可以在 TQnCTL0.TQnCE 位 = 1 时被改写。

向 TQnCCR2 寄存器中设置的数值将被传送到 CCR2 缓冲寄存器中。当 16 位计数器的值与 CCR2 缓冲寄存器的值相等时，产生比较相等中断请求信号 (INTTQnCC2)。如果此时允许 TOQn2 引脚输出，则 TOQn2 引脚的输出信号的电平将反转。

(b) 作为捕捉寄存器时的功能

在自由运行定时器模式下当 TQnCCR2 寄存器用作捕捉寄存器时，如果检测到捕捉触发输入引脚 (TIQn2 引脚) 的有效沿，那么 16 位计数器的计数值将被保存到 TQnCCR2 寄存器。在脉宽测量模式下，如果检测到捕捉输入引脚 (TIQn2 引脚) 的有效沿，那么 16 位计数器的计数值将被保存到 TQnCCR2 寄存器，同时 16 位计数器清为(0000H)。

即使捕捉操作和读取 TQnCCR2 寄存器的操作产生冲突，仍可正确读出 TQ0CCR2 寄存器中的值。

备注 n = 0, 1

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 7-4. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	-
三角波 PWM 模式	比较寄存器	同时写入

(10)TMQn 捕捉/比较寄存器 3 (TQnCCR3)

根据选择模式的不同，TQnCCR3 寄存器可用作捕捉寄存器或比较寄存器。

该寄存器只有在自由运行模式下根据 TQnOPT0.TQnCCS3 位设置的不同，可以用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TQnCCR3 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。

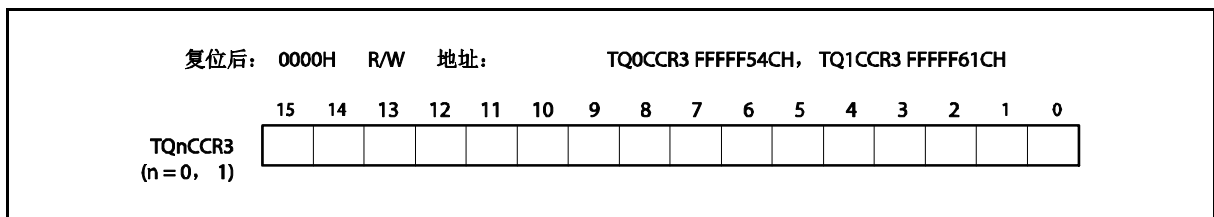
TQnCCR3 寄存器可以在操作过程中进行读写

可对该寄存器进行 16 位的读写操作。

该寄存器复位后的值为 0000H。

★ **注意事项** 以下情况下禁止访问 TQnCCR3 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上外围 I/O 寄存器。

- 当 CPU 工作于副时钟 且主时钟振荡停止时
- 当 CPU 工作于内部振荡时钟时



(a) 作为比较寄存器时的功能

TQnCCR3 寄存器可以在 TQnCTL0.TQnCE 位 = 1 时被改写。

向 TQnCCR3 寄存器中设置的数值将被传送到 CCR3 缓冲寄存器中。当 16 位计数器的值与 CCR3 缓冲寄存器的值相等时，产生比较相等中断请求信号 (INTTQnCC3)。如果此时允许 TOQn3 引脚输出，则 TOQn3 引脚的输出信号的电平将反转。

(b) 作为捕捉寄存器时的功能

在自由运行定时器模式下当 TQnCCR3 寄存器用作捕捉寄存器时，如果检测到捕捉触发输入引脚 (TIQn3 引脚) 的有效沿，那么 16 位计数器的计数值将被保存到 TQnCCR3 寄存器。在脉宽测量模式下，如果检测到捕捉输入引脚 (TIQn3 引脚) 的有效沿，那么 16 位计数器的计数值将被保存到 TQnCCR3 寄存器，同时 16 位计数器清为(0000H)。

即使捕捉操作和读取 TQnCCR3 寄存器的操作产生冲突，仍可正确读出 TQnCCR3 寄存器中的值。

备注 n = 0, 1

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 7-5. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	-
三角波 PWM 模式	比较寄存器	同时写入

(11)TMQn 计数器读取缓冲寄存器 (TQnCNT)

可以通过读取缓冲寄存器 TQn0CNT 来读取 16 位计数器的计数值。

如果该寄存器在 TQnCTL0.TQnCE 位 = 1 时被读取，那么读取的值就是 16 位计数器的计数值。

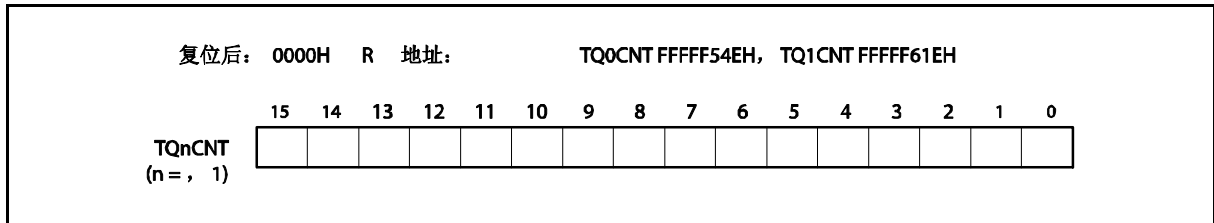
只可以对该寄存器进行 16 位的读取操作。

当 TQnCE 位 = 0 时，TQnCNT 寄存器被清为 0000H。如果此时对 TQnCNT 寄存器进行读取，读取到的值将是 0000H 而不是 16 位计数器的值(FFFFH)。

复位后，TQnCE 位被清零的同时 TQnCNT 寄存器的值也被清零。

注意事项 以下情况下禁止访问 TQnCNT 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上外围 I/O 寄存器。

- 当 CPU 工作于副时钟 且主时钟振荡停止时
- 当 CPU 工作于内部振荡时钟时



(12)TIQnm 引脚噪声消除控制寄存器(QnmNFC)

QnmNFC 寄存器是一个 8 位寄存器，该寄存器用于设置噪声消除的定时器 Q 输入引脚的数字噪声滤波器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: Q00NFC: FFFFFB50H (TIQ00 引脚)
 Q01NFC: FFFFFB54H (TIQ01 引脚)
 Q02NFC: FFFFFB58H (TIQ02 引脚)
 Q03NFC: FFFFFB5CH (TIQ03 引脚)
 Q10NFC: FFFFFB60H (TIQ10 引脚)
 Q11NFC: FFFFFB64H (TIQ11 引脚)
 Q12NFC: FFFFFB68H (TIQ12 引脚)
 Q13NFC: FFFFFB6CH (TIQ13 引脚)

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

(n = 0, 1, m = 0 ~ 3)

NFSTS	通过数字噪声滤波器进行采样次数的设置
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
其它情况			禁止设置

- 注意事项**
1. 请务必将第 3 到 5 位和 7 位清零。
 2. 在设置 QnmNFC 寄存器之前输入到定时器输入引脚(TIQnm)的信号随着信号噪声的消除而输出。
因此，通过使用 QnmNFC 寄存器来设置采样时钟(NFC2 ~ NFC0)和采样次数(NFSTS)，等待初始时间= (采样时钟) × (采样次数)，并启动定时器操作。

备注 可被准确消除的噪声宽度等于 (采样时钟) × (采样次数 - 1)。如果噪声与采样时钟同步，即便是更窄宽度的噪声也可能造成计数错误。

7.5 操作

TMQn 可进行以下操作。

操作	TQnCTL1.TQnEST 位 (软件触发位)	TIQn0 引脚 (外部触发输入)	捕捉/比较寄存器 设置	比较寄存器写入
间隔定时器模式	无效	无效	仅比较	可随时写入
外部事件计数模式 ^{注1}	无效	无效	仅比较	可随时写入
外部触发脉冲输出模式 ^{注2}	有效	有效	仅比较	同时写入
单脉冲输出模式 ^{注2}	有效	有效	仅比较	可随时写入
PWM 输出模式	无效	无效	仅比较	同时写入
自由运行定时器模式	无效	无效	可选	可随时写入
脉宽测量模式 ^{注2}	无效	无效	仅捕捉	不可操作
三角波 PWM 模式	无效	无效	仅比较	同时写入

- 注**
1. 使用外部事件计数模式时，请指定为不检测 TIQn0 引脚捕捉触发输入有效沿（通过将 TQnIOC1.TQnIS1 和 TQnIOC1.TQnIS0 位清为“00”）。
 2. 使用外部触发脉冲输出模式，单脉冲输出模式和脉宽测量模式时，请选择内部时钟作为计数时钟（通过将 TQnCTL1.TQnEEE 位清零）。

备注 n = 0, 1

7.5.1 间隔定时器模式 (TQnMD2 ~ TQnMD0 位 = 000)

在间隔定时器模式下，当 TQnCTL0.TQnCE 位 = 1 时，每隔指定间隔时间都会产生一个中断请求信号 (INTTQnCC0)，同时可以从 TOQn0 引脚输出一个周期为 2 倍间隔时间的方波。

通常情况下，间隔定时器模式中不使用 TQnCCR1 ~ TQnCCR3 寄存器。

图 7-2. 间隔定时器的配置图

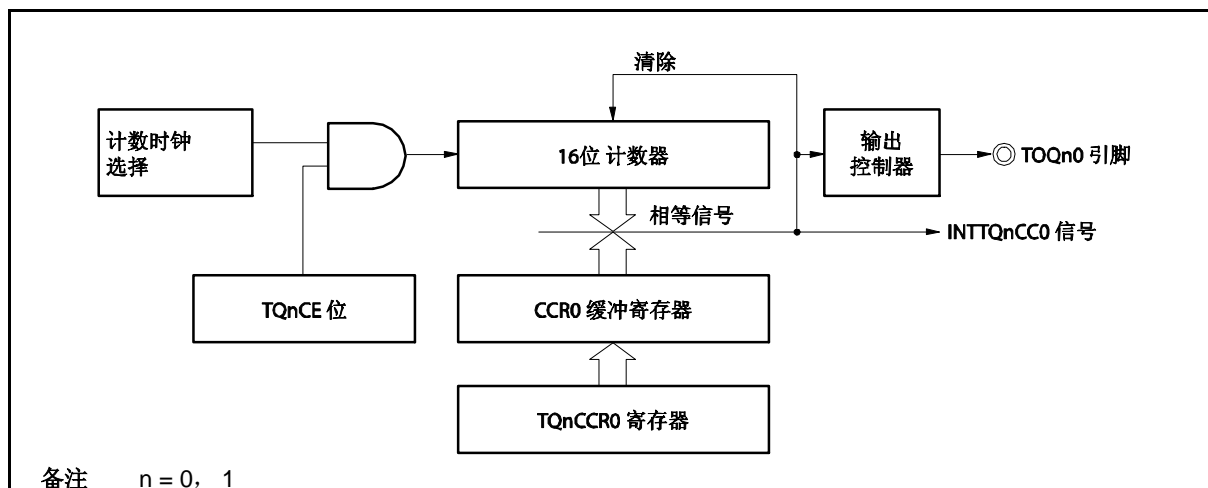
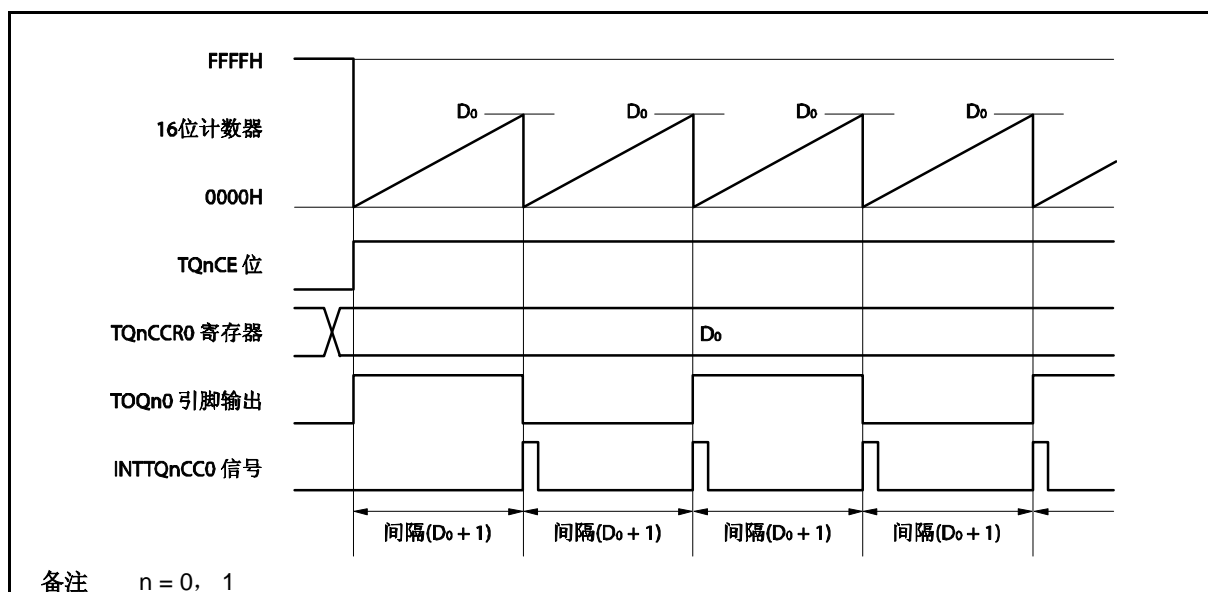


图 7-3. 间隔定时器模式操作的基本时序



16 位计数器将在 TQnCE 位被置 1 时，与计数时钟同步从 FFFFH 清零为 0000H 并开始计数。此时，TOQn0 引脚的输出电平被反转。另外，TQnCCR0 寄存器的设置值也会在此时被传送到 CCR0 缓冲寄存器。

当 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相等时，16 位计数器清零，TOQ00 引脚输出电平反转并且产生一个比较相等中断请求信号 (INTTQnCC0)。

间隔时间可由以下算式求得。

$$\text{间隔时间} = (\text{TPnCCR0 寄存器的设定值} + 1) \times \text{计数时钟周期}$$

图 7-4. 间隔定时器模式操作的寄存器设置(1/2)

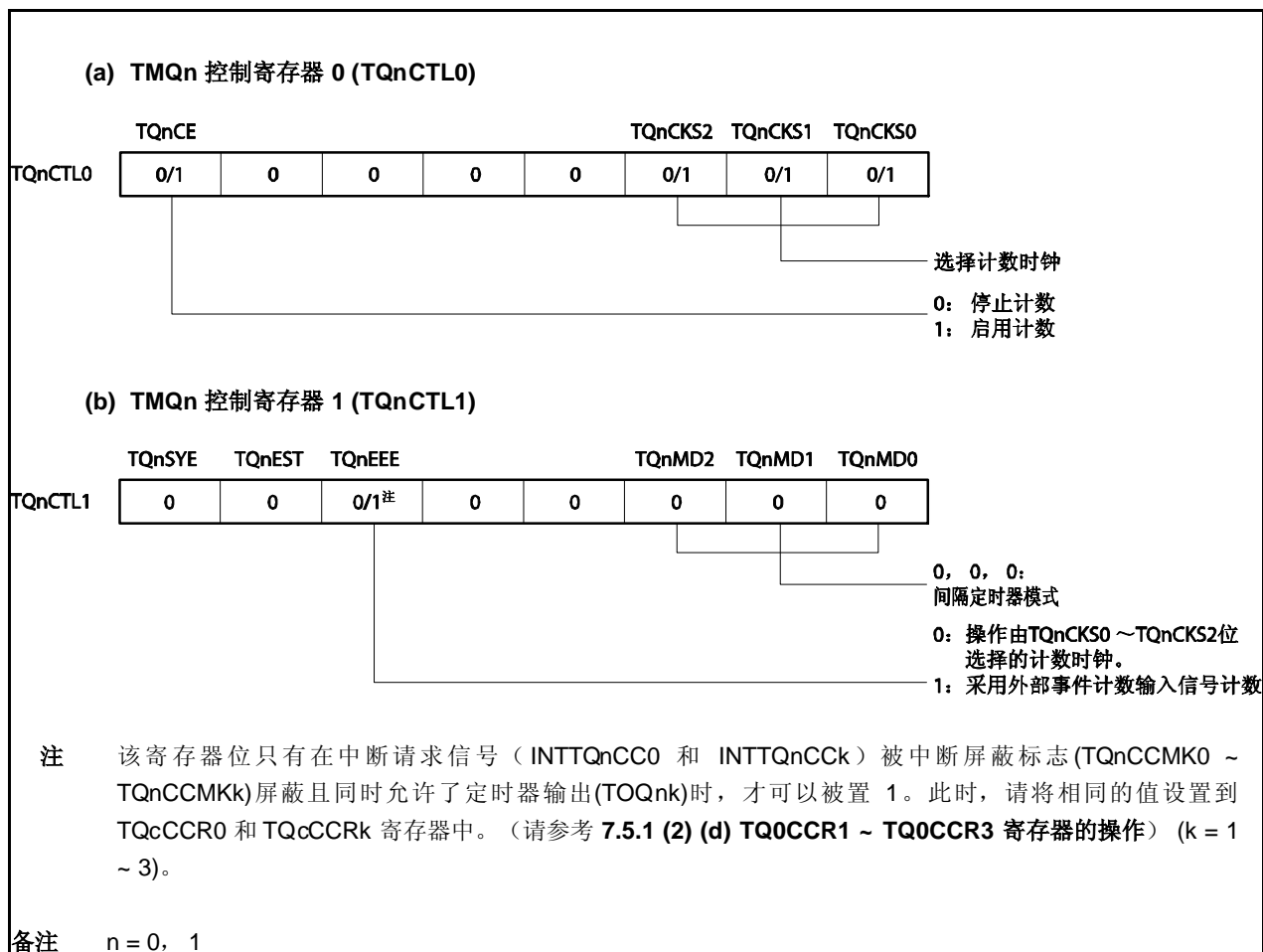
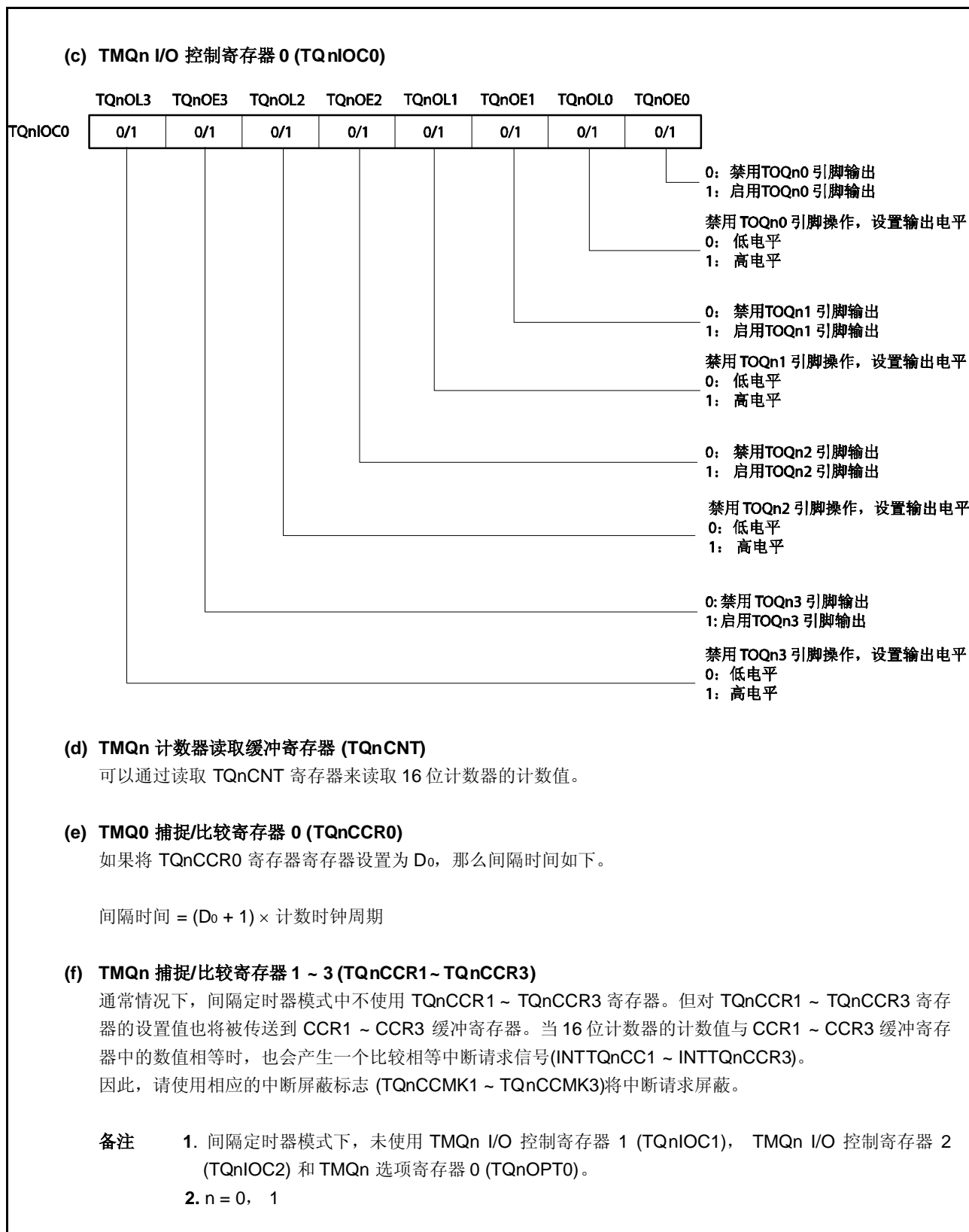
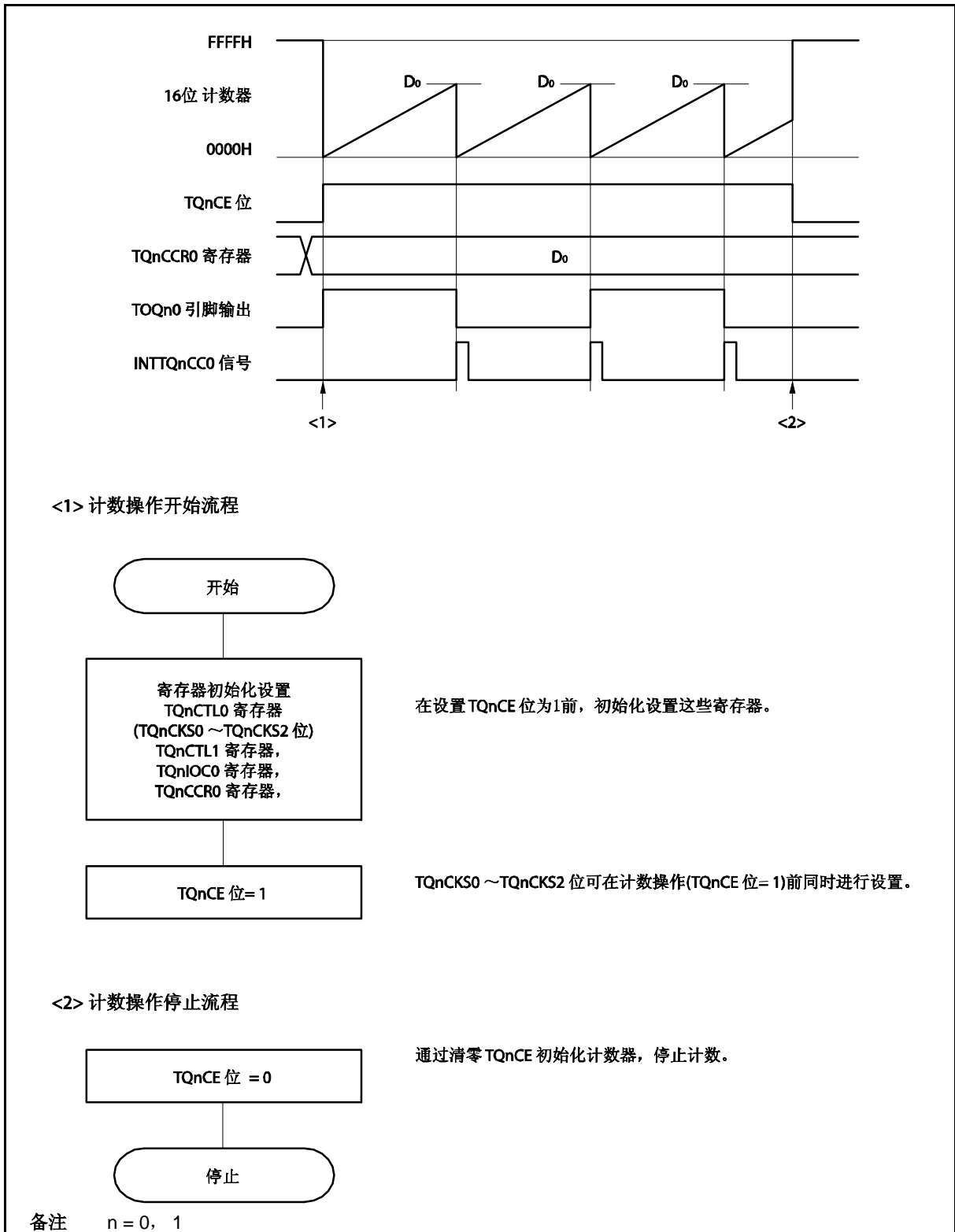


图 7-4. 间隔定时器模式操作的寄存器设置(2/2)



(1) 间隔定时器模式操作流程

图 7-5. 间隔定时器模式下的软件处理流程

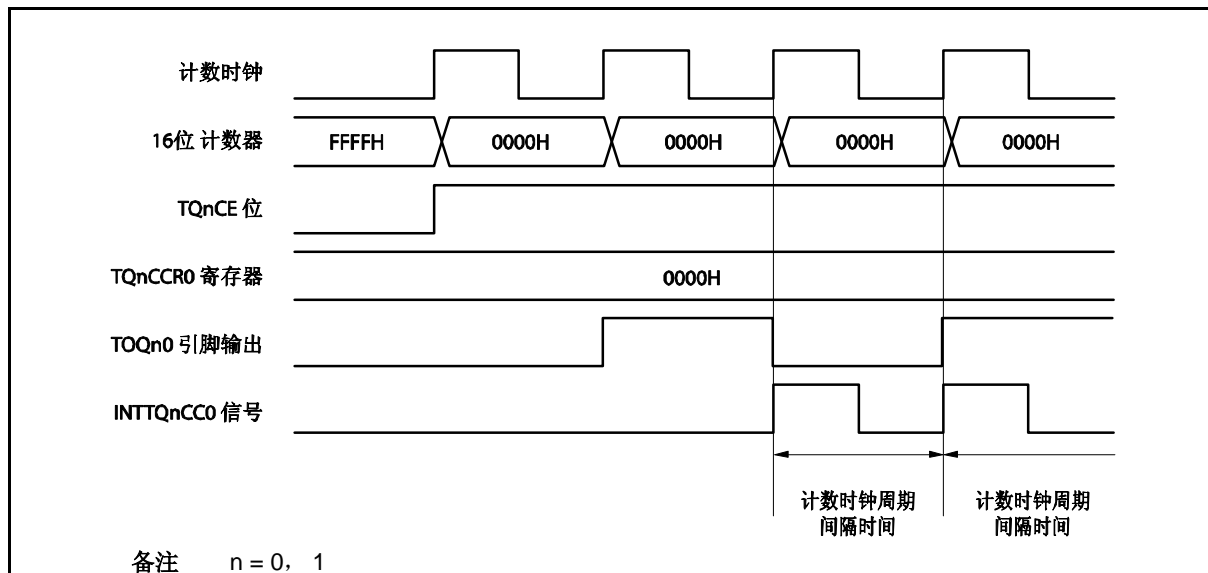


(2) 间隔定时器模式操作时序

(a) TQ0CCR0 寄存器被设置为 0000H 时的操作

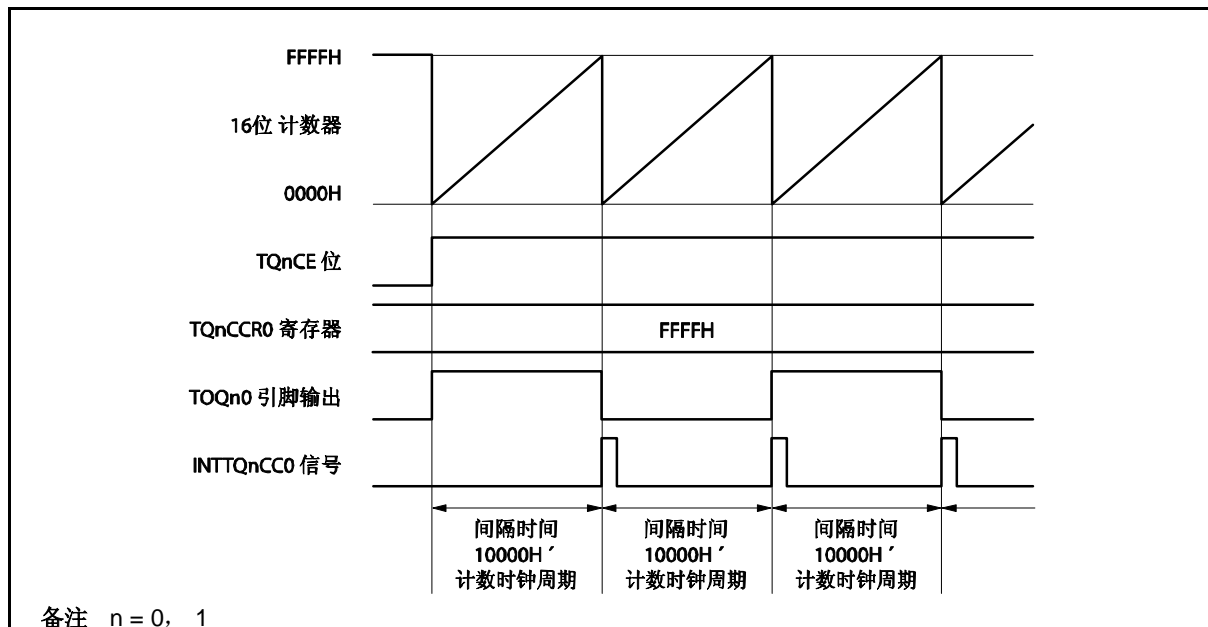
如果 TQ0CCR0 寄存器被设置为 0000H，那么在第二个计数时钟之后（含第二个），都会产生 INTTQ0CC0 信号，并且 TOQ00 引脚的电平都会反转。

16 位计数器的计数值始终保持为 0000H。



(b) TQ0CCR0 寄存器被设置为 FFFFH 时的操作

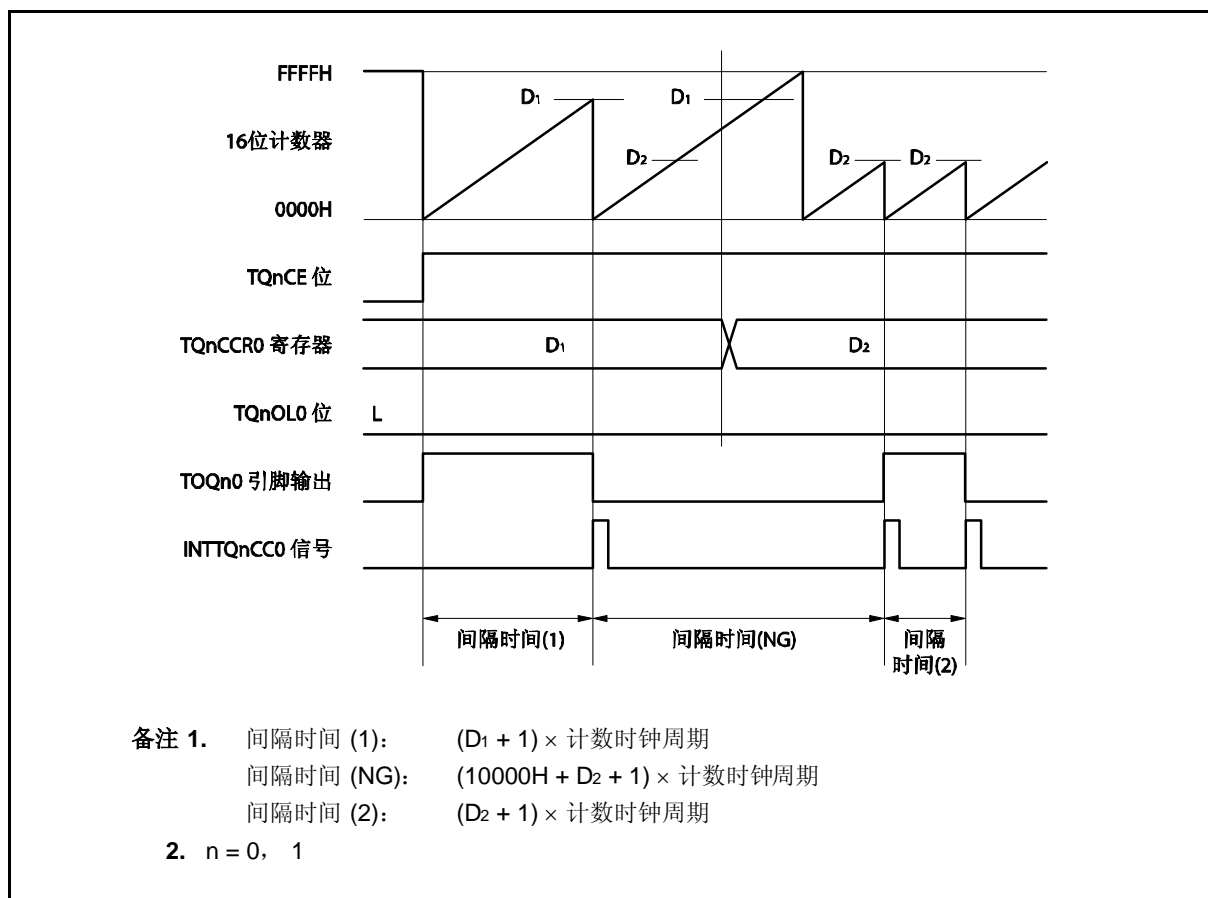
如果 TQ0CCR0 寄存器被设置为 FFFFH，那么 16 位计数器将计数到 FFFFH。并在下一个计数时钟时刻同步被清零。同时产生 INTTQ0CC0 信号，TOQ00 引脚电平反转。此时，不会产生溢出中断请求信号(INTTQ0OV)，溢出标志位 (TQ0OPT0.TQ0OVF 位) 也不会被置 1。



(c) 改写 TQnCCR0 寄存器时需要注意的事项

要将 TQnCCR0 寄存器的值改小，首先停止计数然后再更改设置值。

如果在计数时向 TQnCCR0 寄存器写入更小的值，16 位计数器可能会溢出。



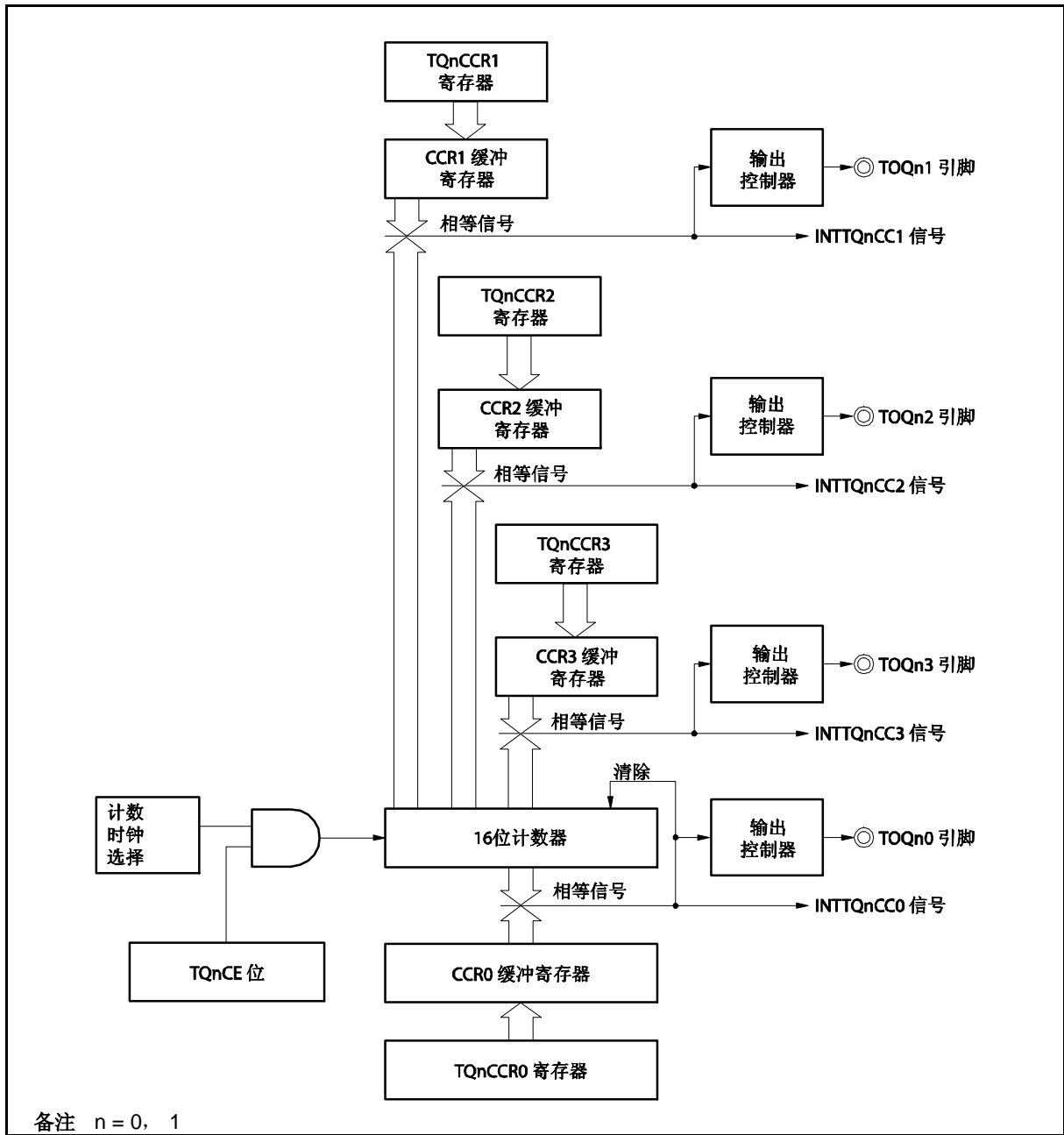
当计数到大于 D_2 小于 D_1 时如果 TQnCCR0 寄存器的值从 D_1 变为 D_2 ，改写 TQnCCR0 寄存器后，此值立即传送到 CCR0 缓冲寄存器。因此，16 位计数器的计数值将与 D_2 进行比较。

因为计数值已经超过了 D_2 ，因此 16 位计数器会继续计数到 FFFFH 并产生溢出，然后从 0000H 开始再次计数。在重新计数过程中，当计数值与 D_2 相等时，产生 INTTQnCC0 信号，TOQn0 引脚的输出信号的电平反转。

因此，可能不会在最初预期的“ $(D_1 + 1) \times$ 计数时钟周期”或“ $(D_2 + 1) \times$ 计数时钟周期”间隔时间产生 INTTQnCC0 信号，而是在“ $(10000H + D_2 + 1) \times$ 计数时钟周期”的间隔时间产生。

(d) 操作 TQnCCR1 ~ TQnCCR3 寄存器

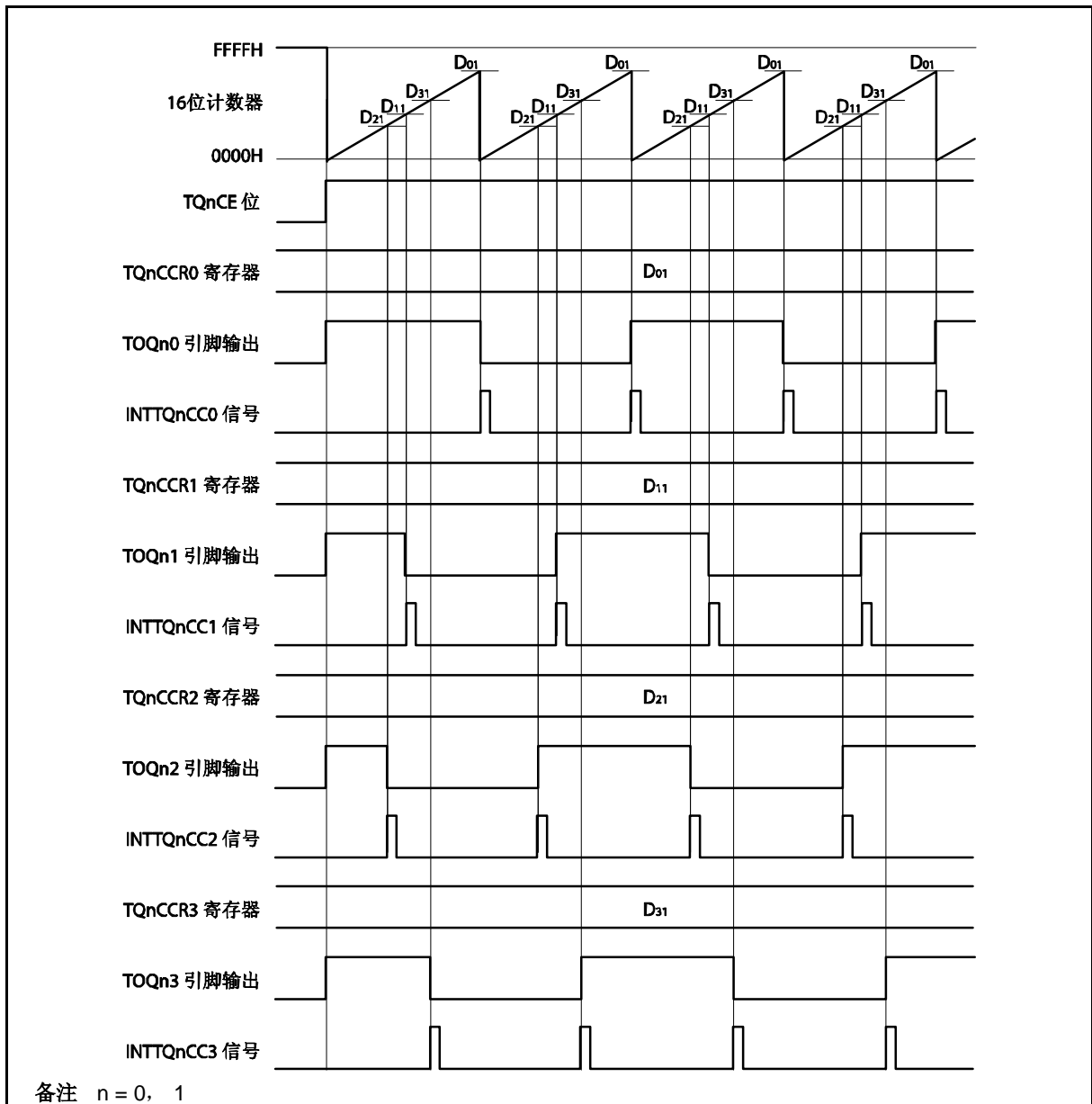
图 7-6. TQnCCR1 ~ TQnCCR3 寄存器的配置



如果 TQnCCRk 寄存器的设置值小于 TQnCCR0 寄存器的设置值，则每个周期产生一次 INTTQnCCk 信号。同时，TOQnk 引脚的输出信号的电平反转。TOQnk 引脚输出的方波周期与 TOQn0 引脚的方波周期相同。

备注 k = 1 ~ 3
n = 0, 1

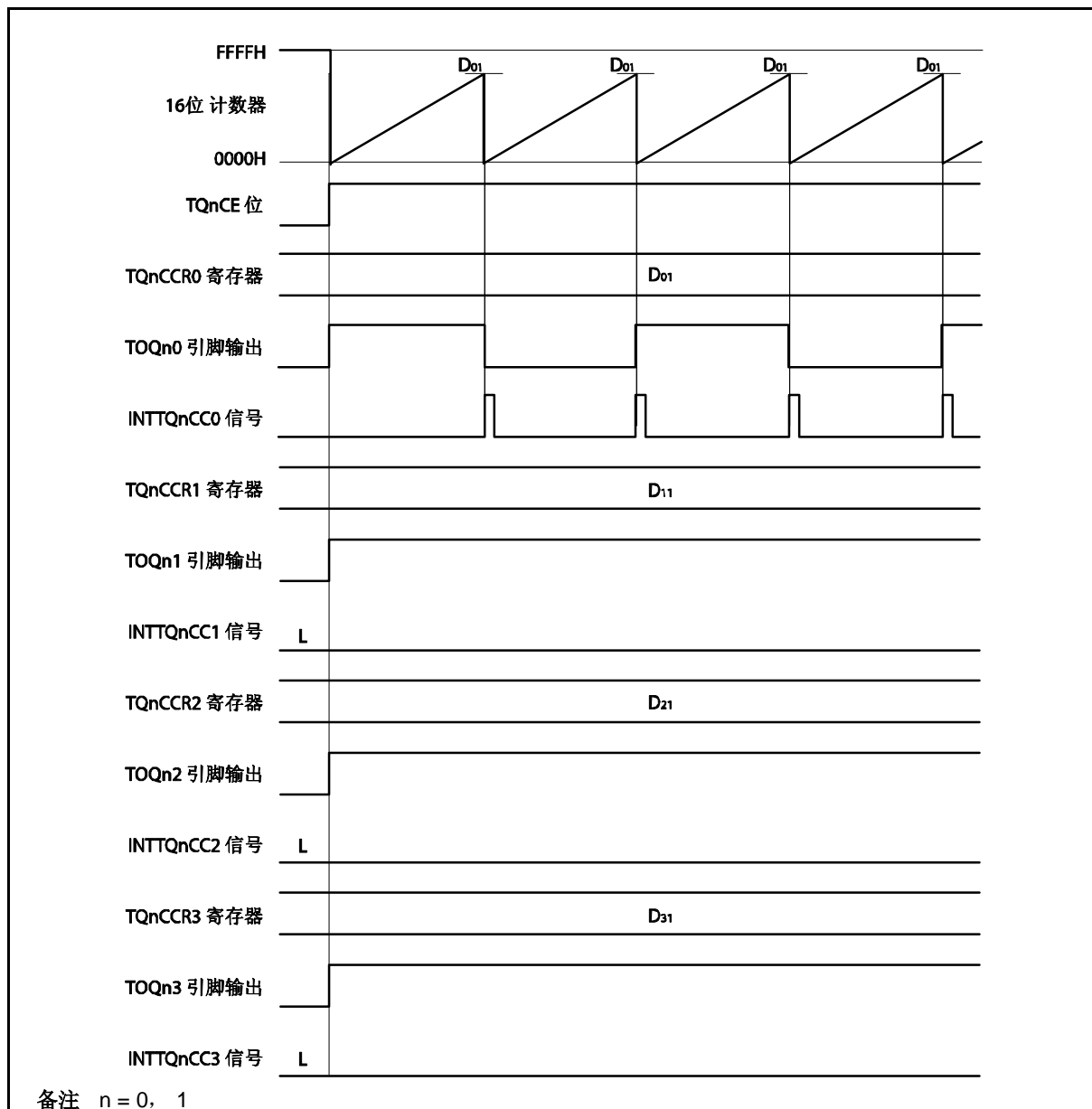
图 7-7. 当 $D_{01} \geq D_{k1}$ 时的时序图



如果 TQnCCRk 寄存器的设置值大于 TQnCCR0 寄存器，16 位计数器的计数值不会与 TQnCCRk 寄存器相等。因此，不产生 INTTQnCCk 信号，TOQnk 引脚的输出也不发生变化。

备注 k = 1 ~ 3
n = 0, 1

图 7-8. 当 $D_{01} < D_{k1}$ 时的时序图



7.5.2 外部事件计数模式 (TQnMD2 ~ TQnMD0 位 = 001)

在外部事件计数模式，只有当 TQnCTL0.TQnCE 位置 1 时外部事件计数输入的有效沿才有效，每次计到指定数量的有效沿后都会产生中断请求信号 (INTTQnCC0)。该模式下不能使用 TOQn0 引脚。

通常，TQnCCR1 ~ TQnCCR3 寄存器不用于外部事件计数模式。

图 7-9. 外部事件计数模式配置

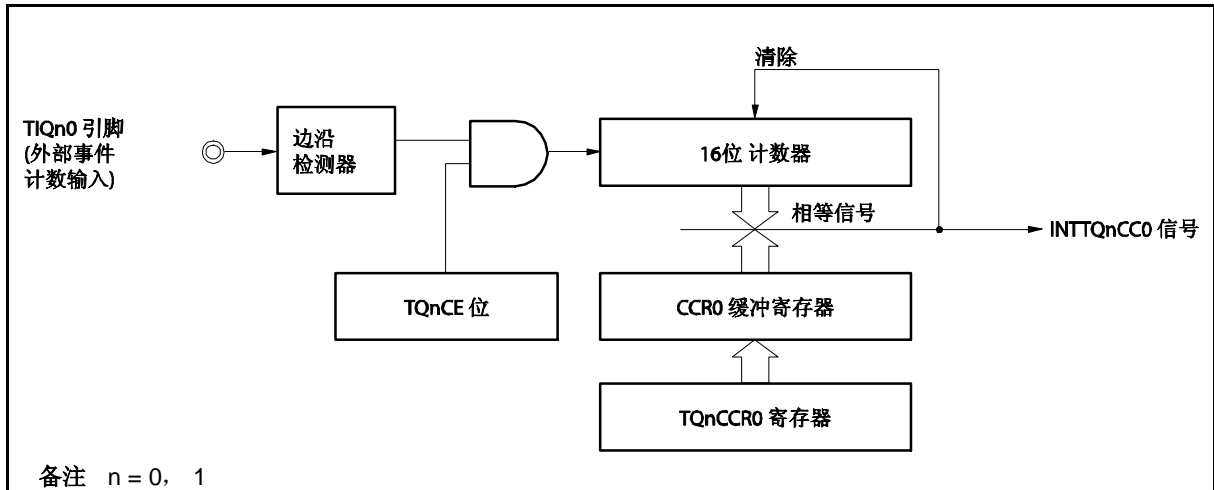
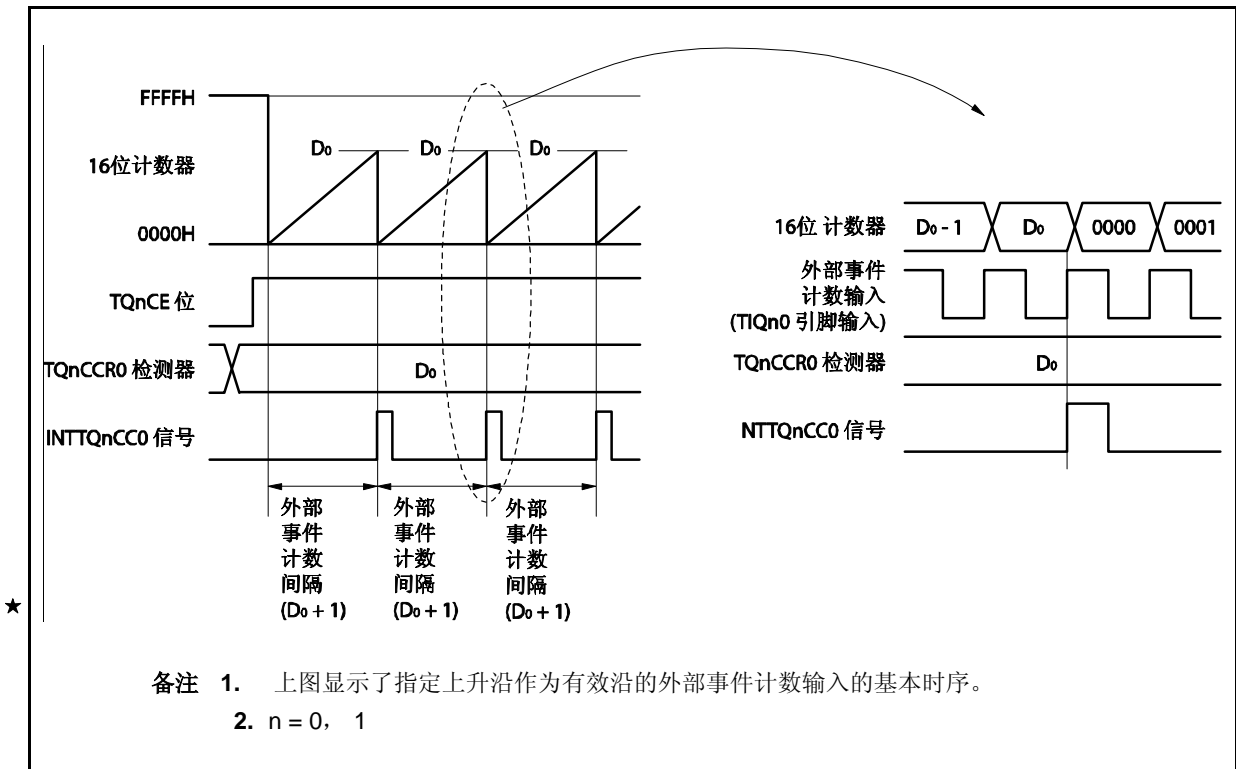


图 7-10. 外部事件计数模式的基本时序



当 TQnCE 位置 1 时，16 位计数器的值从 FFFFH 清除为 0000H。每次外部事件计数输入的有效沿时计数器计数。此外，TQnCCR0 寄存器的设置值发送到 CCR0 缓冲寄存器。

当 16 位计数器的计数值与 CCR0 缓冲寄存器相等时，16 位计数器被清为 0000H，且产生比较相等中断请求信号 (INTTQnCC0)。

每次检测到外部事件计数输入的有效沿时 (TQ0CCR0 寄存器的设置值+ 1)，都会产生 INTTQnCC0 信号。

图 7-11. 外部事件计数模式操作的寄存器设置(1/2)

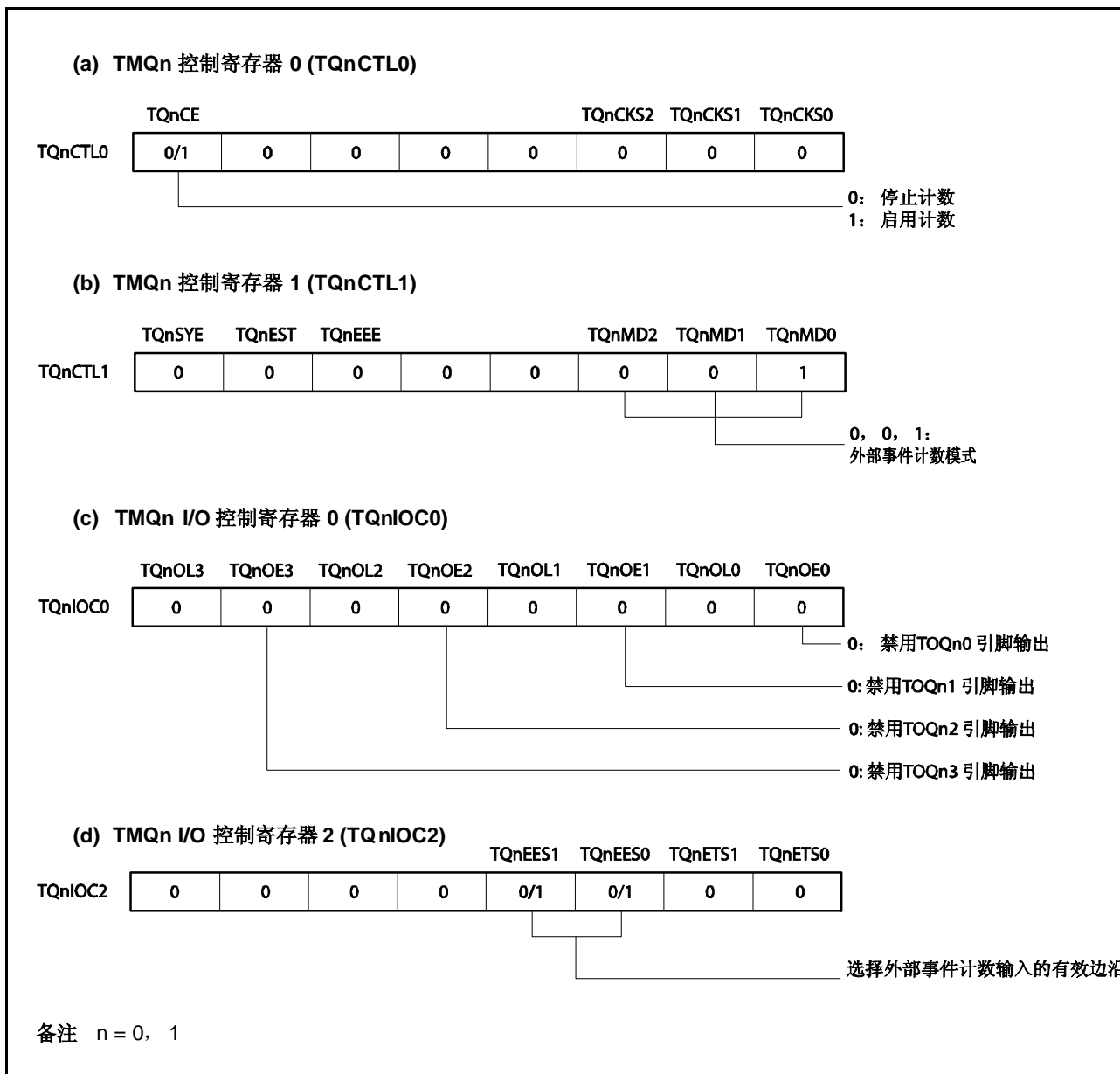


图 7-11. 外部事件计数模式操作的寄存器设置(2/2)

(e) TMQn 计数器读取缓冲寄存器(TQnCNT)

可以通过读取 TQnCNT 寄存器来读取 16 位计数器的计数值。

(f) TMQn 捕捉/比较寄存器 0 (TQnCCR0)

如果向 TQnCCR0 寄存器写入 D_0 ，则计数器被清零，并且将在外部事件计数达到 $(D_0 + 1)$ 时产生比较相等中断请求信号(INTTQnCC0)。

(g) TMQn 捕捉/比较寄存器 1 到 3 (TQnCCR1 ~ TQnCCR3)

通常情况下，外部事件计数模式中不使用 TQnCCR1 ~ TQnCCR3 寄存器。但对 TQnCCR1 ~ TQnCCR3 寄存器设置的数值也将被传送到 CCR1 ~ CCR3 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 ~ CCR3 缓冲寄存器中的数值相等时，也会产生一个比较相等中断请求信号(INTTQ0CC1 ~ INTTQ0CC3)。

因此，请使用相应的中断屏蔽标志(TQnCCMK1 ~ TQnCCMK3) 将中断请求屏蔽。

<R>

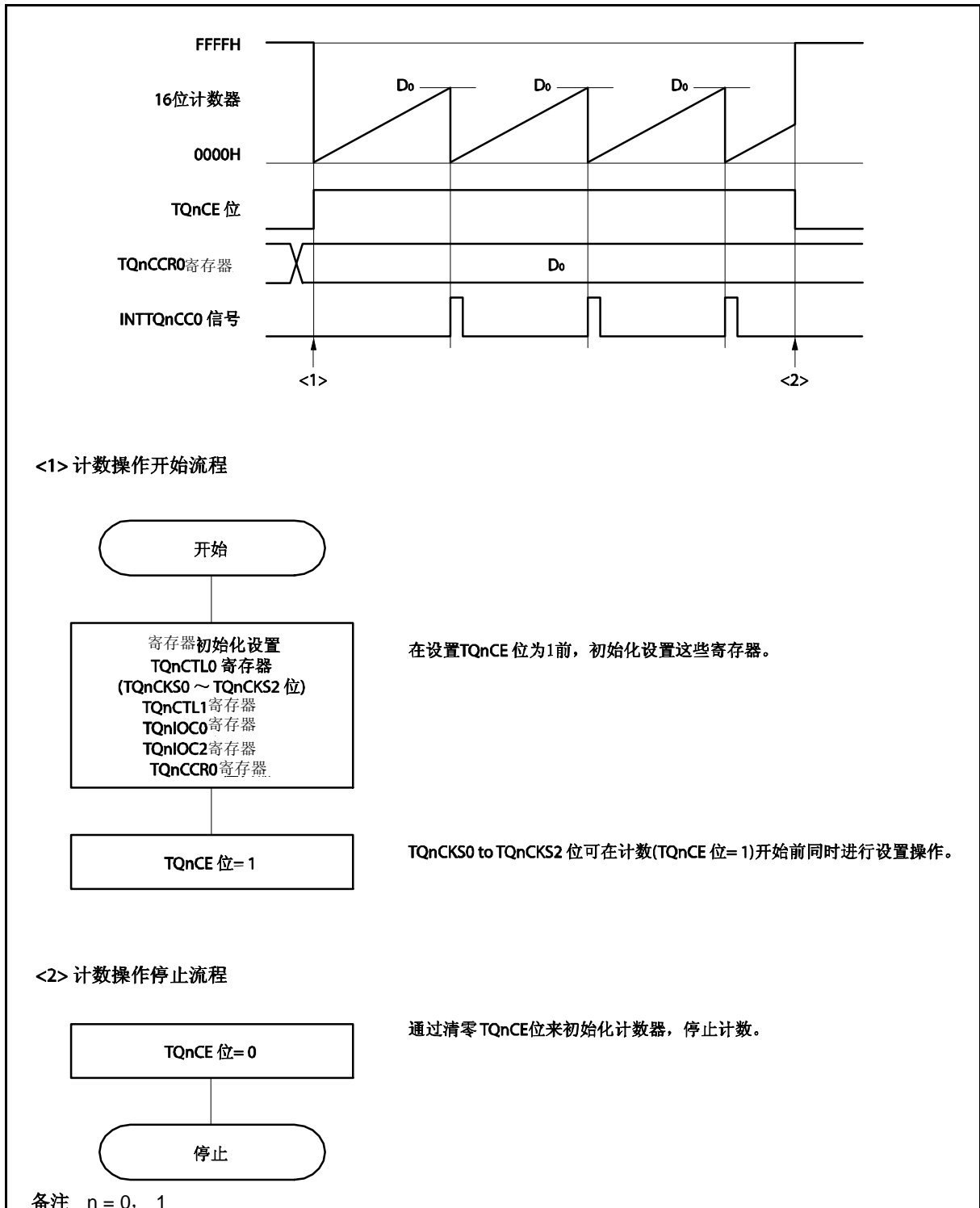
注意事项 当外部时钟用于计数时钟时，外部时钟可以仅从 TIQn0 引脚输入。此时，设置 TQnIOC1.TQnIS1 和 TQnIOC1.TQnIS0 位为 00。（捕捉触发输入（TIQn0 引脚）：无边沿检测）。

备注 1. 外部事件计数模式下，并未使用 TMQn I/O 控制寄存器 1 (TQnIOC1) 以及 TMQn 选项寄存器 0 (TQnOPT0)。

2. $n = 0, 1$

(1) 外部事件计数模式操作流程

图 7-12. 外部事件计数模式的软件处理流程

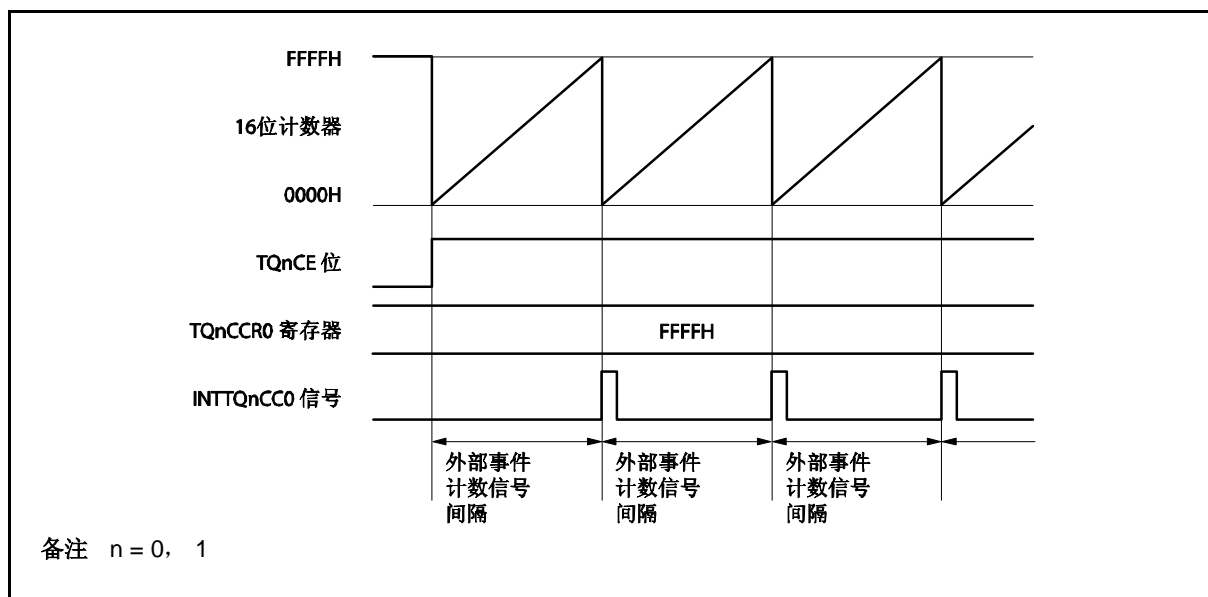


(2) 外部事件计数模式的操作时序

- 注意事项
1. 在外部事件计数模式中，不要将 TQ0CCR0 寄存器设置为 0000H。
 2. 在外部事件计数模式中，禁止使用定时器输出功能。如果使用外部事件计数输入进行定时器输出，那么设置间隔定时器模式，并且通过计数时钟的外部事件计数输入来启用的操作。（TQnCTL1.TQnMD2 ~ TQnCTL1.TQnMD0 位 = 000，TQnCTL1.TQnEEE 位 = 1）。

(a) TQnCCR0 寄存器被设置为 FFFFH 时的操作

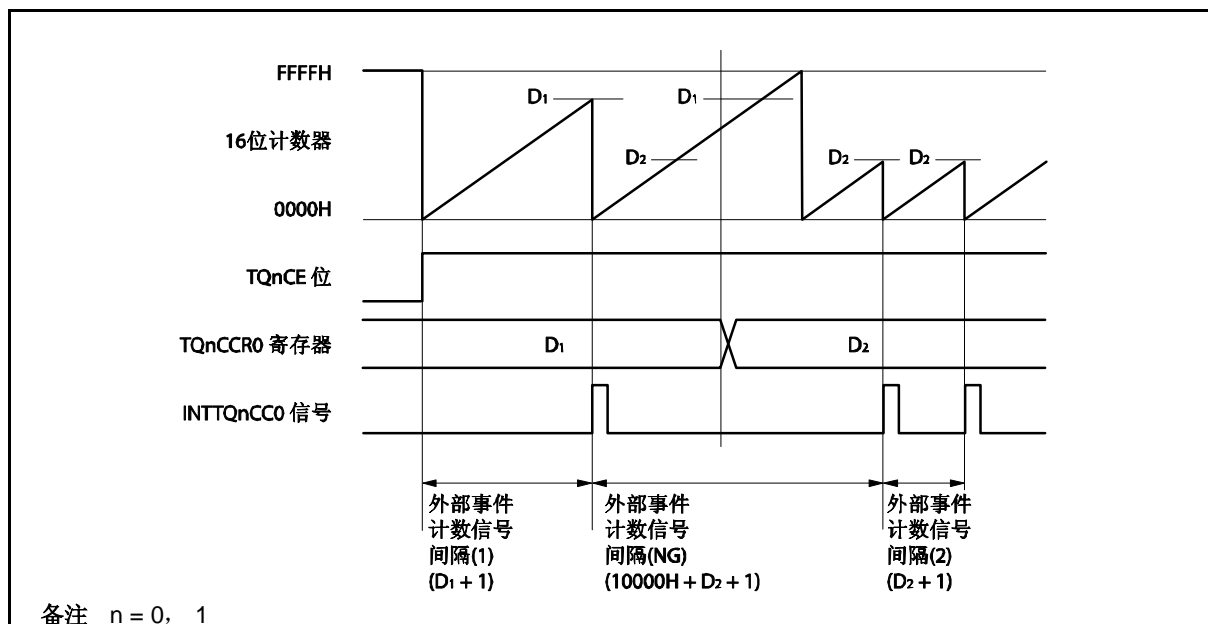
如果 TQnCCR0 寄存器被设置为 FFFFH，那么 16 位计数器将在每次检测到外部事件计数信号的有效沿处计数，直到 FFFFH。并在下一个计数事件时刻被同步清零，同时产生 INTTQnCC0 信号。此时，溢出标志位(TQnOPT0.TQnOVF 位)不会被置 1。



(b) 改写 TQnCCR0 寄存器时需要注意的事项

将 TQnCCR0 寄存器的值改小时，要先停止计数，再对设定值进行改变。

如果在计数过程中将 TQnCCR0 寄存器的值改小，那么 16 位计数器可能产生溢出。



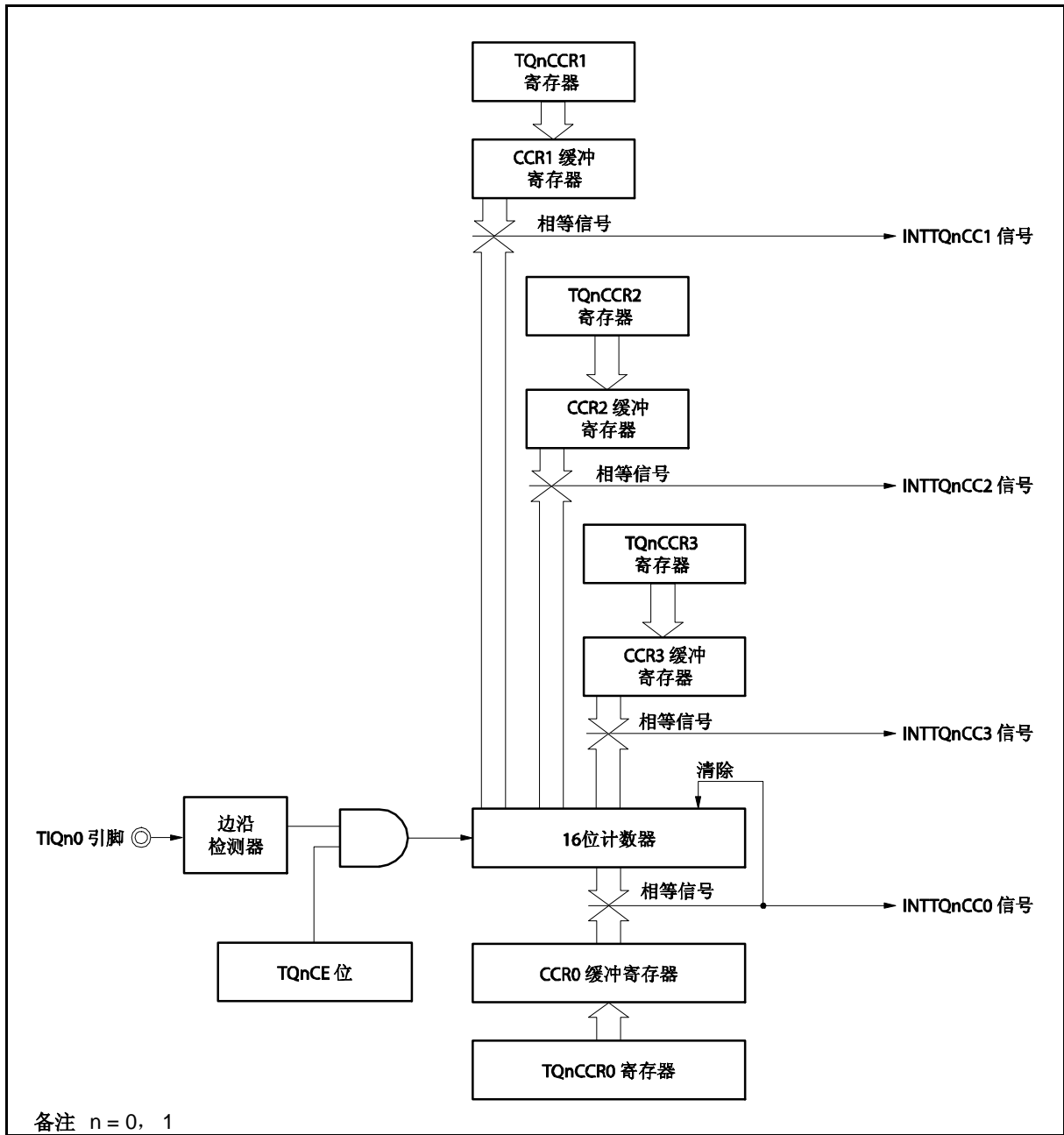
如果当前计数值大于 D_2 但小于 D_1 ，将 TQnCCR0 寄存器的值从 D_1 改变为 D_2 ，新的设定值将在改写后立即从 TQnCCR0 寄存器传送到 CCR0 缓冲寄存器中，那么，16 位计数器的计数值将与 D_2 进行比较。

由于计数值已经超过了 D_2 ，因此 16 位计数器会继续计数到 FFFFH 并产生溢出，之后重新从 0000H 开始计数。在重新计数过程中，计数值与 D_2 相等时，才会产生 INTTQnCC0 信号。

由此可见，INTTQnCC0 信号不会在预期的有效边沿计数次数 “ $(D_1 + 1)$ 次” 或 “ $(D_2 + 1)$ 次” 时产生，而是在一个 “ $(10000H + D_2 + 1)$ 次” 的有效边沿计数次数时产生。

(c) TQ0CCR1 ~ TQ0CCR3 寄存器的操作

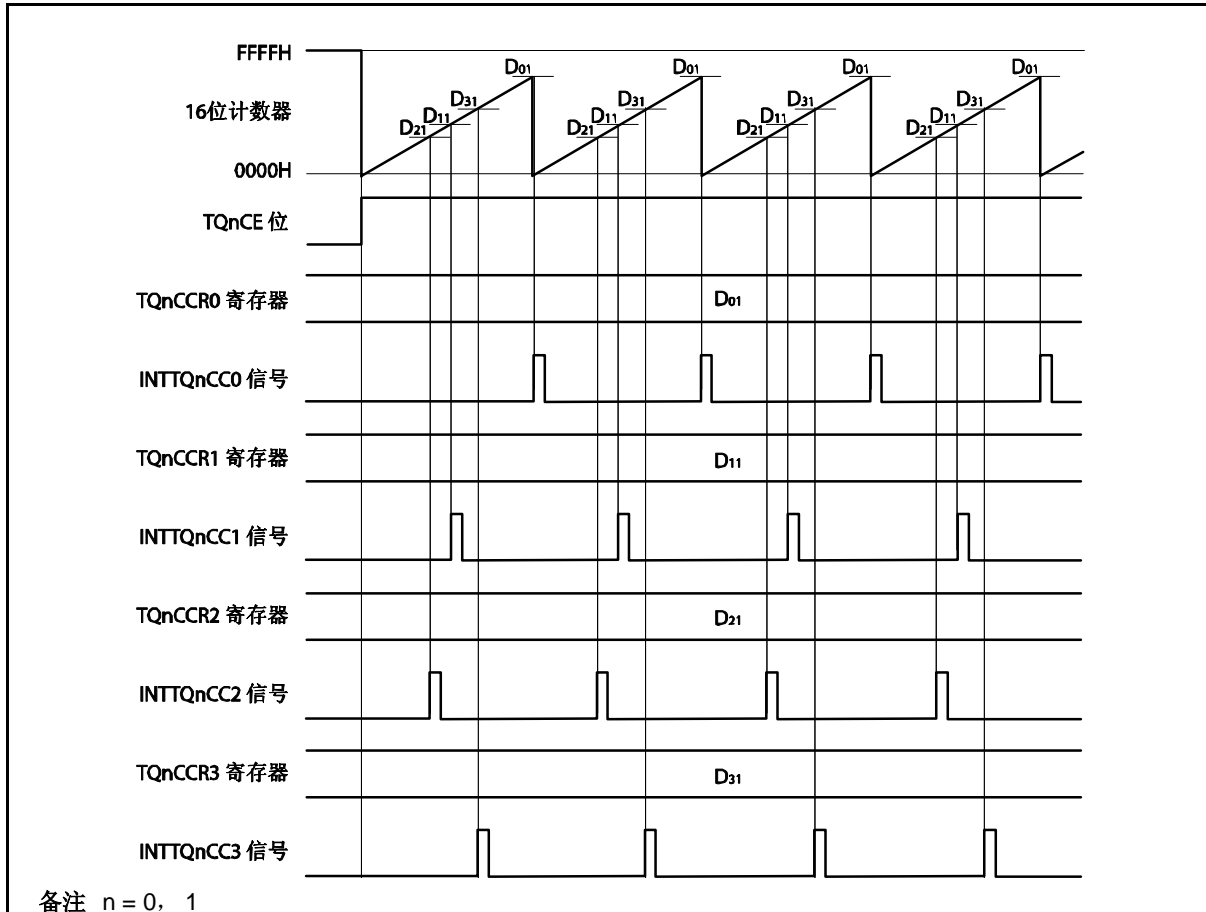
图 7-13. TQnCCR1 ~ TQnCCR3 寄存器的配置



如果 TQnCCRk 寄存器的值小于 TQnCCR0 寄存器的值，那么 INTTQnCCk 信号将每周期产生一次。

备注 k = 1 ~ 3
n = 0, 1

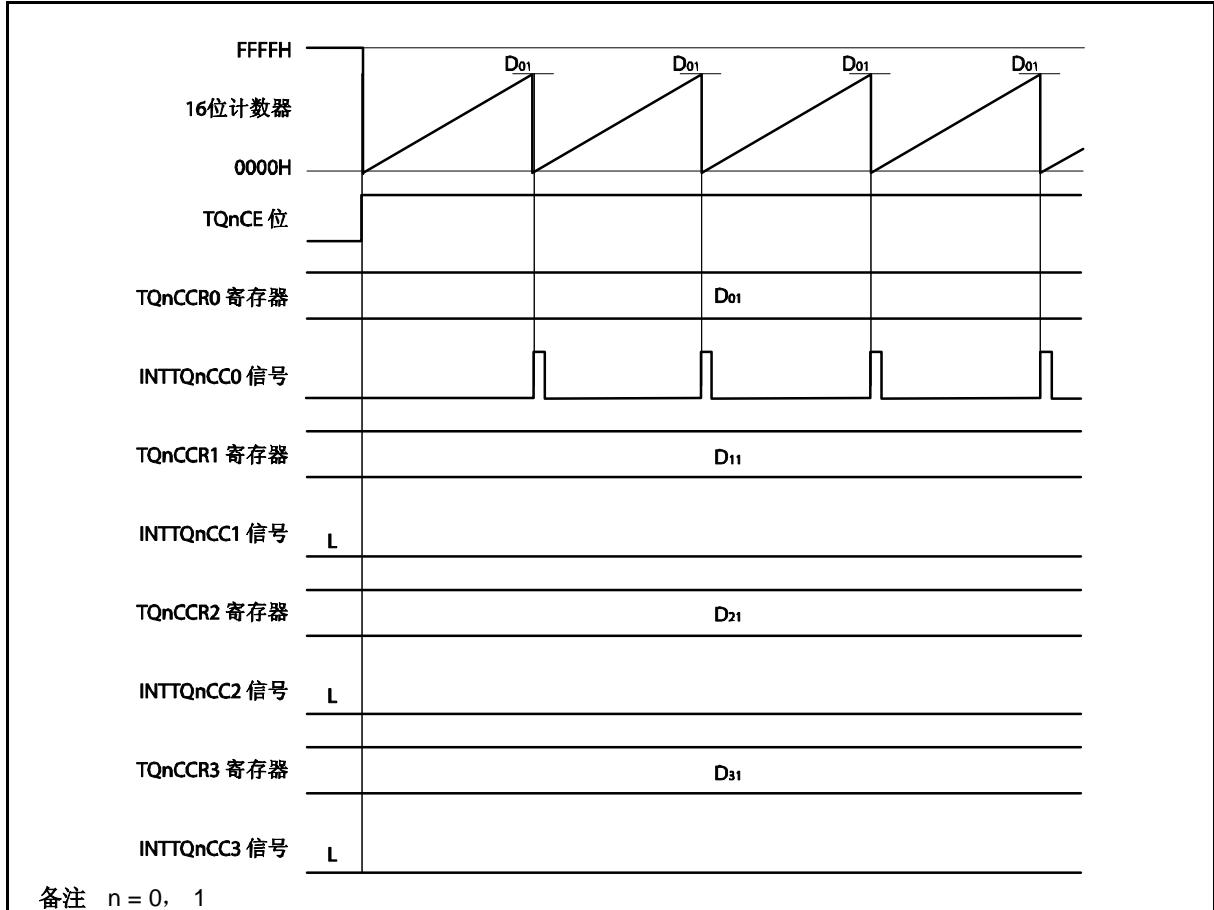
图 7-14. D₀₁ ≥ D_{k1} 时的时序图



如果 TQnCCRk 寄存器的设置值大于 TQnCCR0 寄存器的值，那么 16 位计数器的计数值不会与 TQnCCRk 寄存器中的值相等。因此，不会产生 INTTQnCCk 信号。

备注 k = 1 ~ 3
n = 0, 1

图 7-15. D₀₁ < D_{k1} 时的时序图



7.5.3 外部触发脉冲输出模式 (TQnMD2 ~ TQnMD0 位 = 010)

在外部触发脉冲输出模式中, TQnCTL0.TQnCE 位被置 1 后, 16 位定时器/事件计数器 Q 进入等待触发信号的状态。当检测到一个外部触发输入信号的有效沿时, 16 位定时器/事件计数器 Q 开始计数, 并由 TOQn1 ~ TOQn3 引脚输出 PWM 波形。

也可以使用软件触发代替外部触发来开始脉冲的输出。当使用软件触发时, 还可由 TOQn0 引脚输出一个以 PWM 波形周期的 2 倍为周期的方波。

图 7-16. 外部触发脉冲输出模式的配置

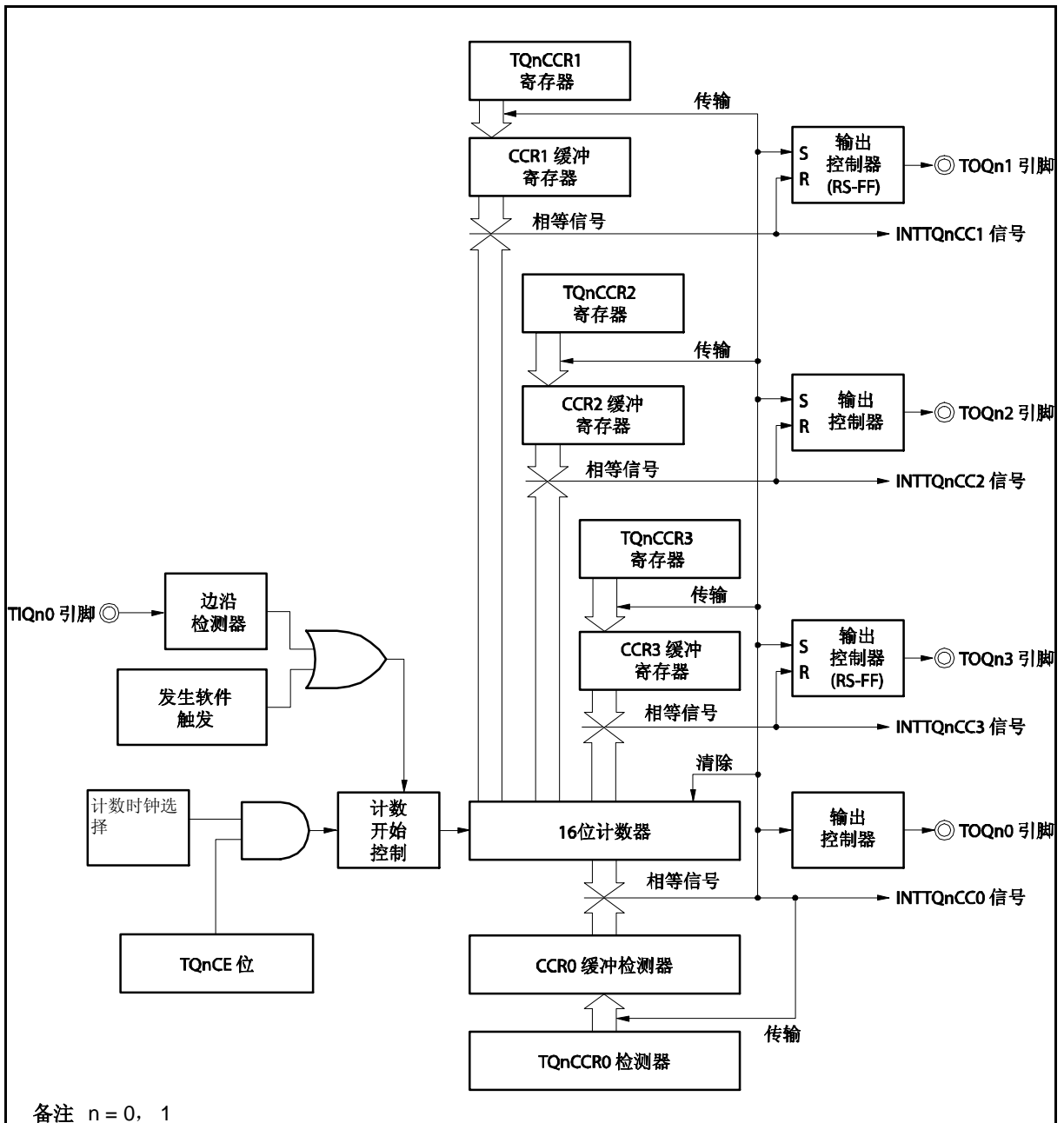
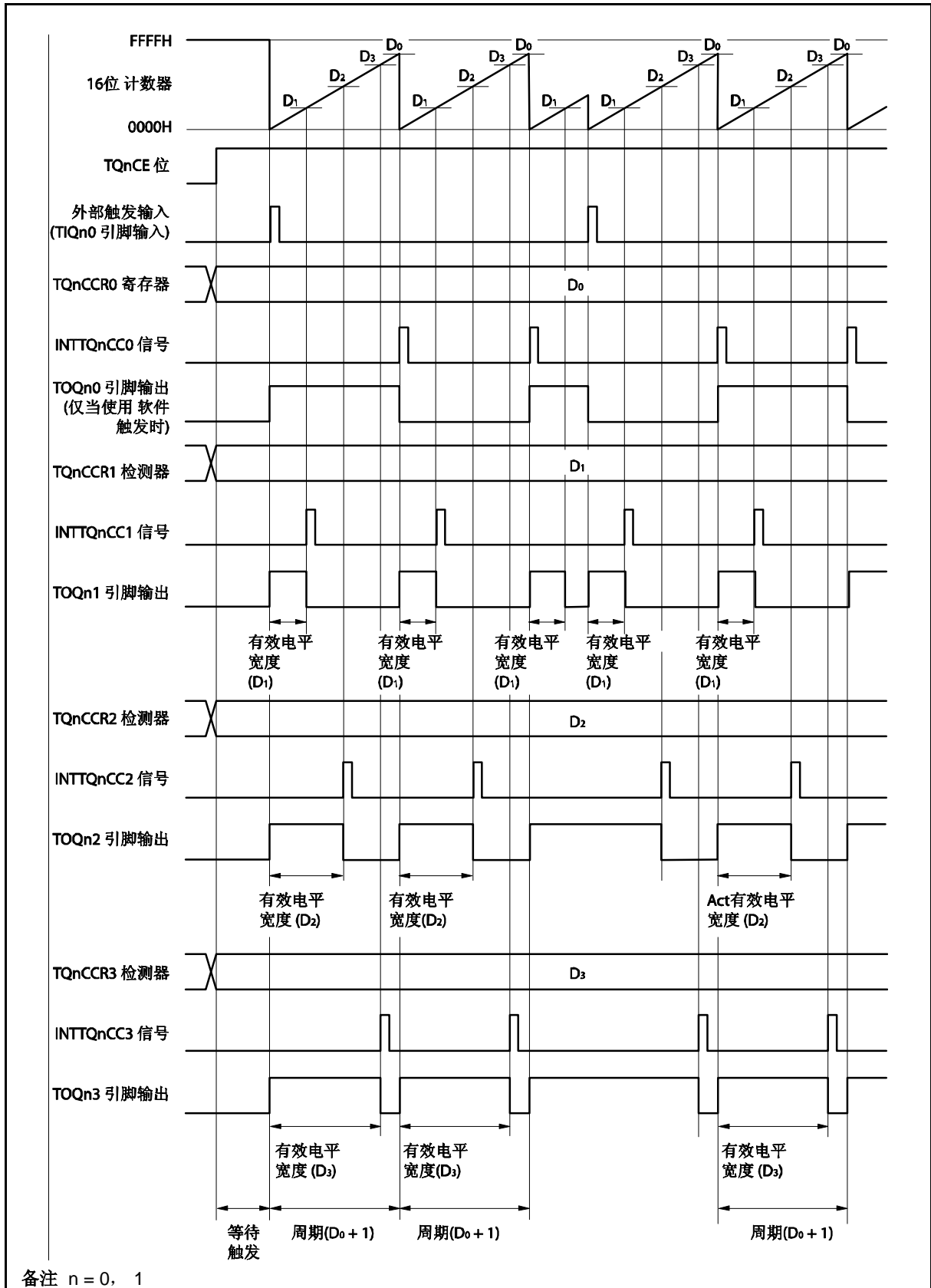


图 7-17. 外部触发脉冲输出模式的基本时序



TQnCE 位被置 1 后，16 位定时器/事件计数器 Q 进入等待触发信号的状态。触发产生后，16 位计数器从 FFFFH 清零为 0000H，同时开始计数并从 TOQnk 引脚输出一个 PWM 波形。如果在计数器计数过程中，再次检测到触发信号，那么计数器将被清为 0000H 并重新开始计数。（TOQn0 引脚的输出电平反转。TOQnk 在触发产生时输出高电平（无论当前状态为高/低）。）

PWM 波形的有效电平宽度，周期以及占空比系数可由以下的算式得出。

$$\begin{aligned} \text{有效电平宽度} &= (\text{TQnCCRk 寄存器的设置值}) \times \text{计数时钟周期} \\ \text{周期} &= (\text{TQnCCR0 寄存器的值} + 1) \times \text{计数时钟周期} \\ \text{占空比系数} &= (\text{TQnCCRk 寄存器的值}) / (\text{TQnCCR0 寄存器的值} + 1) \end{aligned}$$

比较相等中断请求信号 INTTQnCC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器中的值相等后产生。同时 16 位计数器也会被清零。比较相等中断请求信号 INTTQnCCk 会在 16 位计数器的计数值与 CCRk 缓冲寄存器中的值相等时产生。

当 16 位计数器的计数值与 CCRm 缓冲寄存器中的值相等时，TQ0CCRm 寄存器中设置的值会被传送到 CCRm 缓冲寄存器，同时 16 位计数器被清为 0000H。

触发源可以从外部触发输入信号的有效沿和设置软件触发（TQnCTL1.TQnEST 位 = 1）的方式中进行选择。

备注 k = 1 ~ 3
m = 0 ~ 3
n = 0, 1

图 7-18. 外部触发脉冲输出模式的寄存器设置(1/3)

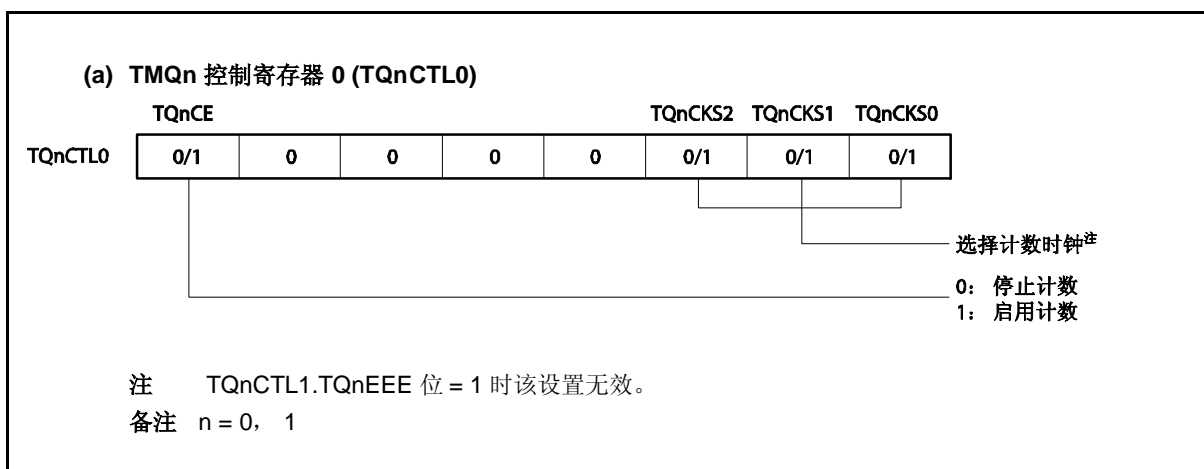


图 7-18. 外部触发脉冲输出模式的寄存器设置(2/3)

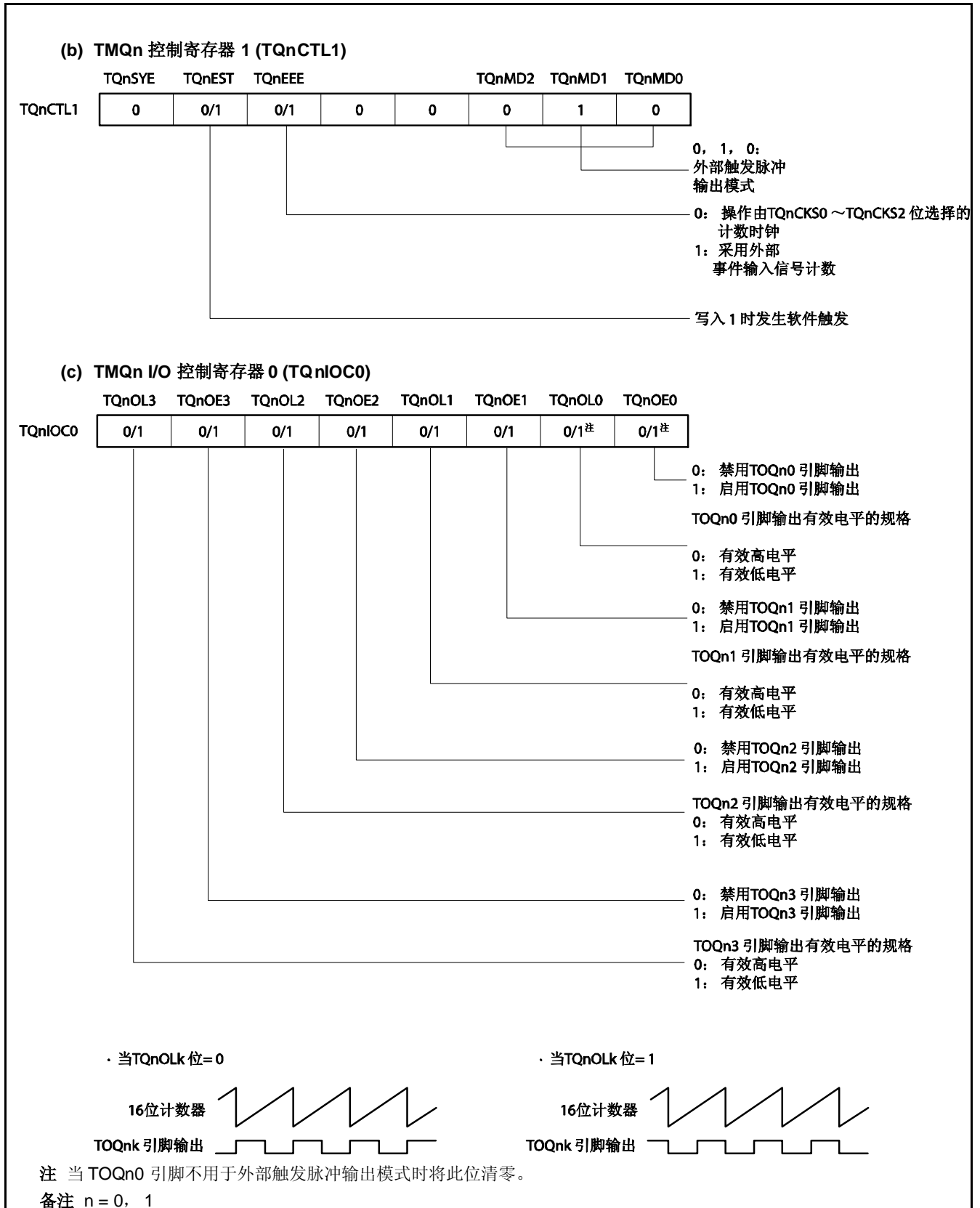
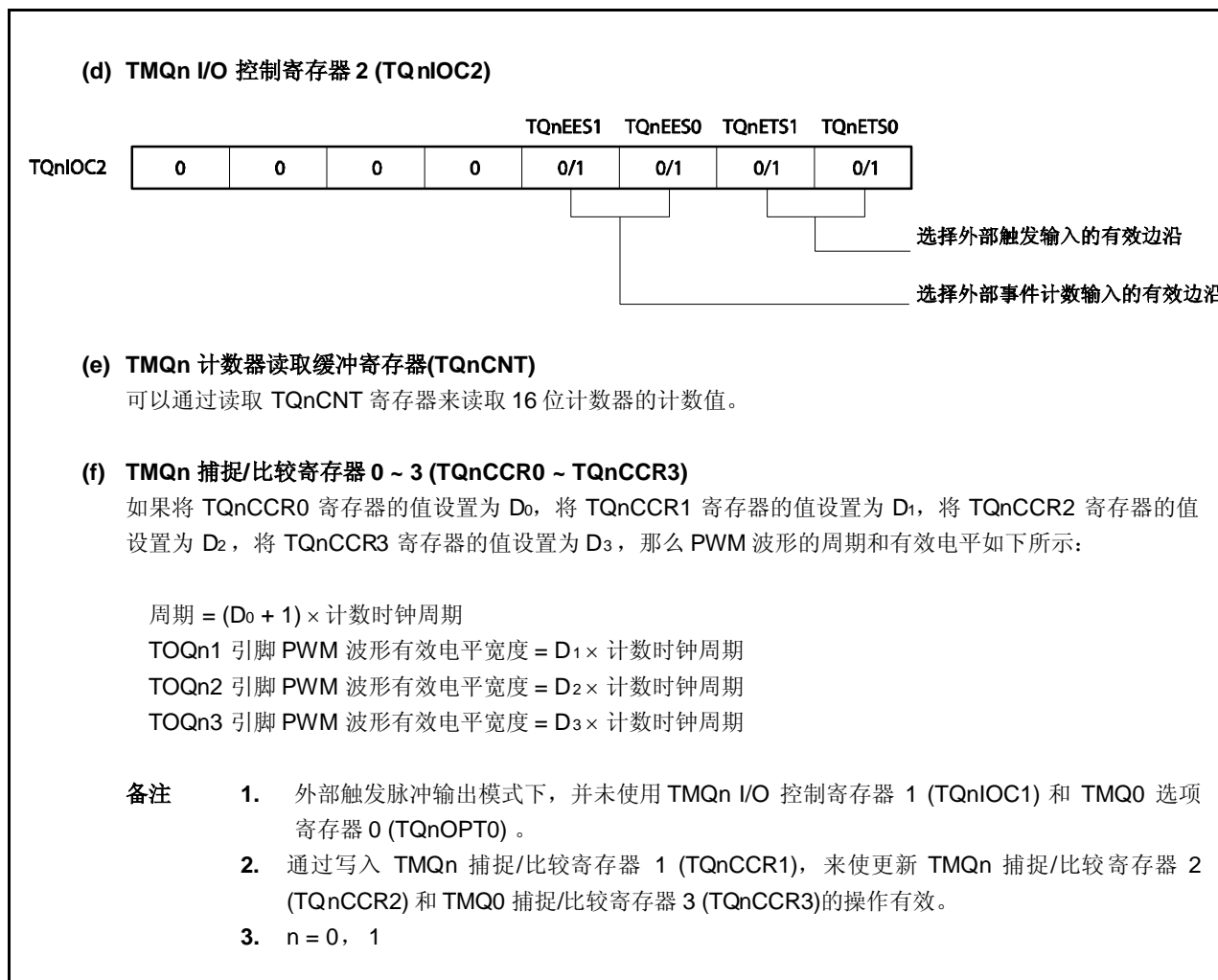


图 7-18. 外部触发脉冲输出模式的寄存器设置(3/3)



(1) 外部触发脉冲输出模式的操作系统

图 7-19. 外部触发脉冲输出模式的软件处理流程(1/2)

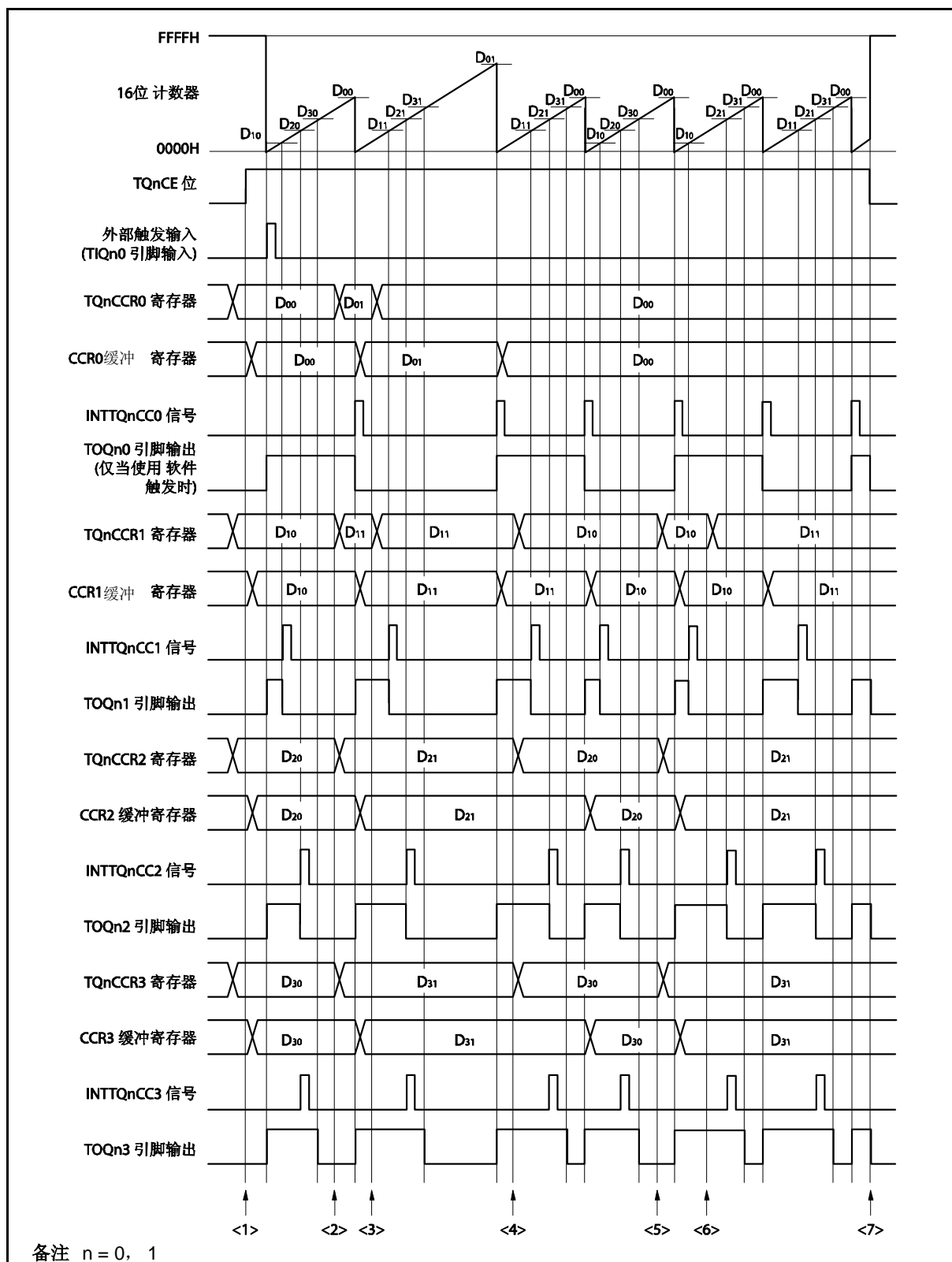
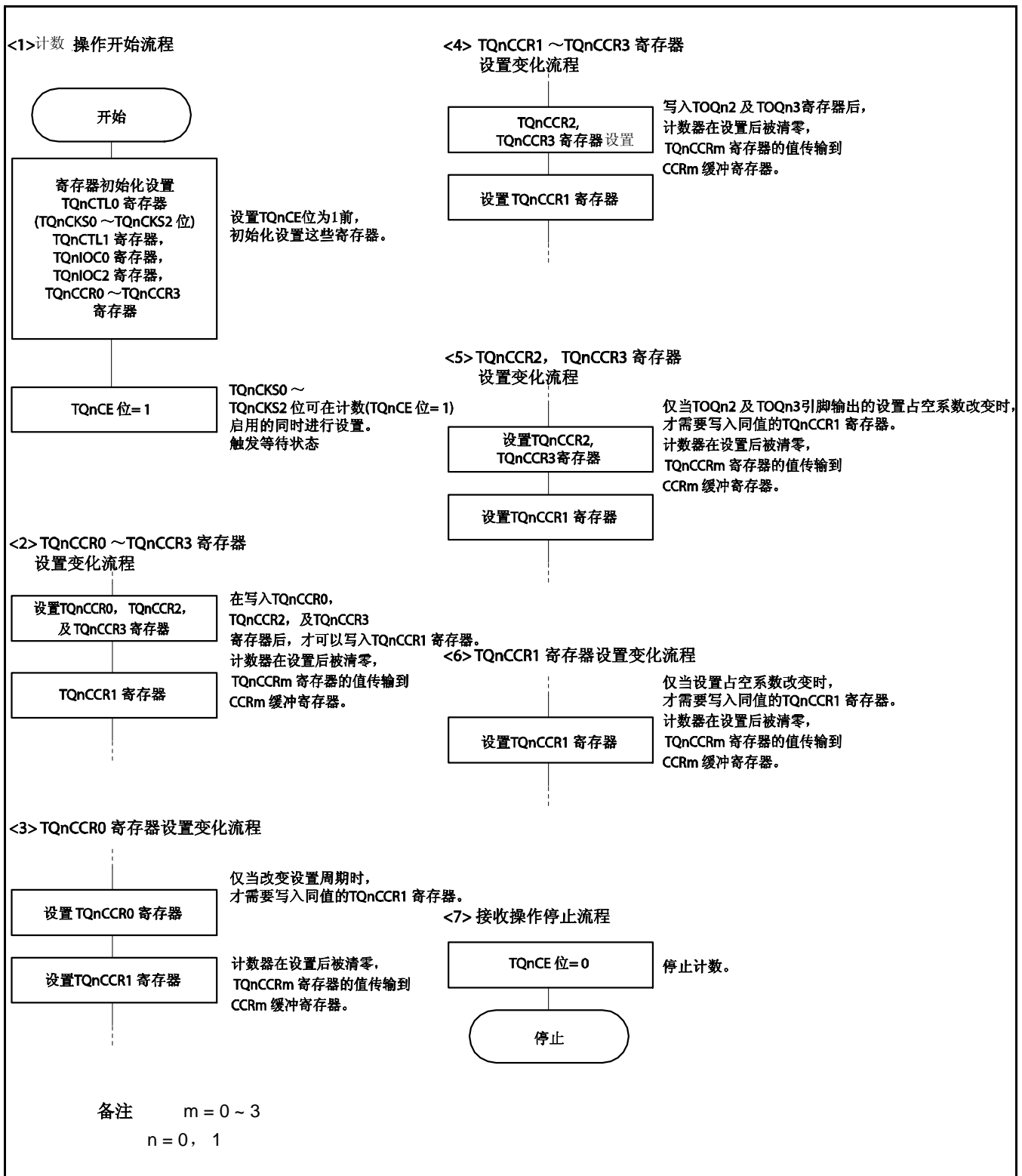


图 7-19. 外部触发脉冲输出模式的软件处理流程(2/2)

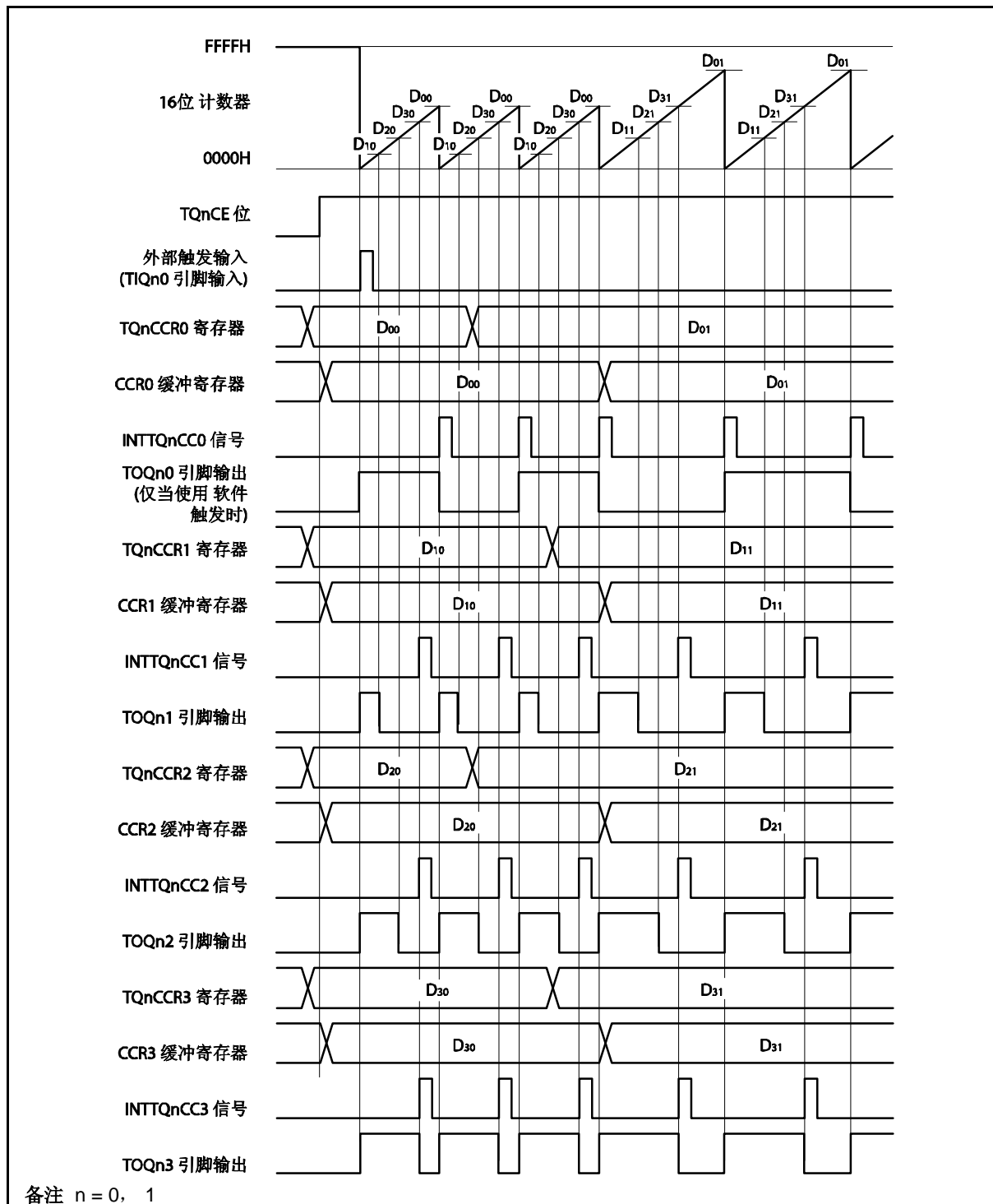


(2) 外部触发脉冲输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

计数器计数过程中，若改变 PWM 波形，则要在最后进行对 TQnCCR1 寄存器的写入操作。

对 TQnCCR1 寄存器进行写入操作之后，若需再次更改 TQnCCRk 寄存器的值，须等待下一个 INTTQnCC0 信号被检测到后，再进行相关操作。



若要将 TQnCCRm 寄存器中设置的数据传送到 CCRm 缓冲寄存器，必须对 TQnCCR1 寄存器进行写入操作。

同时改变 PWM 波形的周期和有效电平时，先将周期值写入 TQnCCR0 寄存器，再将有效电平宽度值写入 TQnCCR2 和 TQnCCR3 寄存器，最后将有效电平写入 TQnCCR1 寄存器。

只改变 PWM 波形的周期时，先将周期值写入 TQnCCR0 寄存器，再向 TQnCCR1 寄存器写入与原来相同的值。

只改变 PWM 波形的有效电平宽度（占空比系数）时，首先需要向 TQnCCR2 和 TQnCCR3 寄存器设置有效电平值，然后再向 TQnCCR1 寄存器设置有效电平。

若只改变 TOQn1 引脚输出的 PWM 波形的有效电平宽度（占空比系数），只需要设置 TQnCCR1 寄存器。

若只改变 TOQn2 和 TOQn3 引脚输出的 PWM 波形的有效电平宽度（占空比系数），首先需要向 TQnCCR2 和 TQnCCR3 寄存器设置有效电平宽度，然后向 TQnCCR1 寄存器写入相同的值。

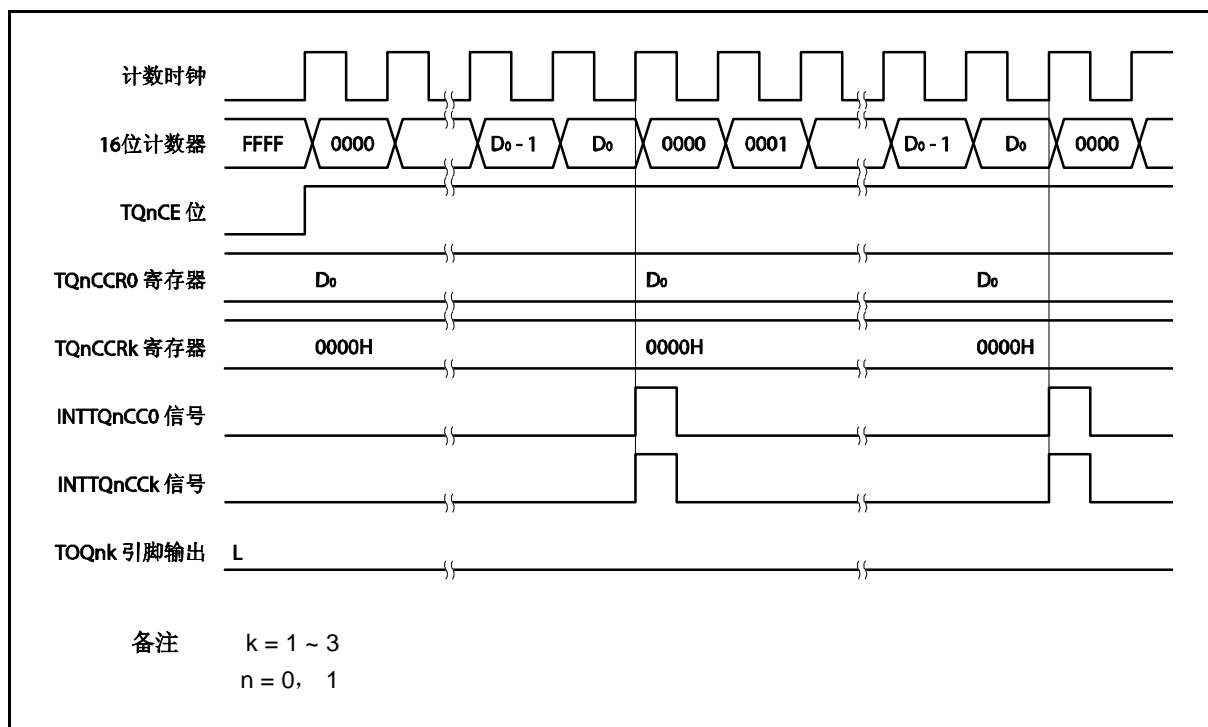
数据被写入 TQnCCR1 寄存器之后，当 16 位计数器被清零时，TQnCCRm 寄存器中的数值才会被传送到 CCRm 缓冲寄存器中。此时 CCRm 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

向 TQnCCR1 寄存器的写入操作完成后，若需再次改写 TQnCCR0 ~ TQnCCR3 寄存器的值，则要在产生 INTTQnCC0 信号之后再进行改写。否则，由于数据从 TQnCCRm 寄存器传送到 CCRm 缓冲寄存器的时序与 TQnCCRm 寄存器的写入时序相冲突，CCRm 缓冲寄存器中所设置的数值可能无法确定。

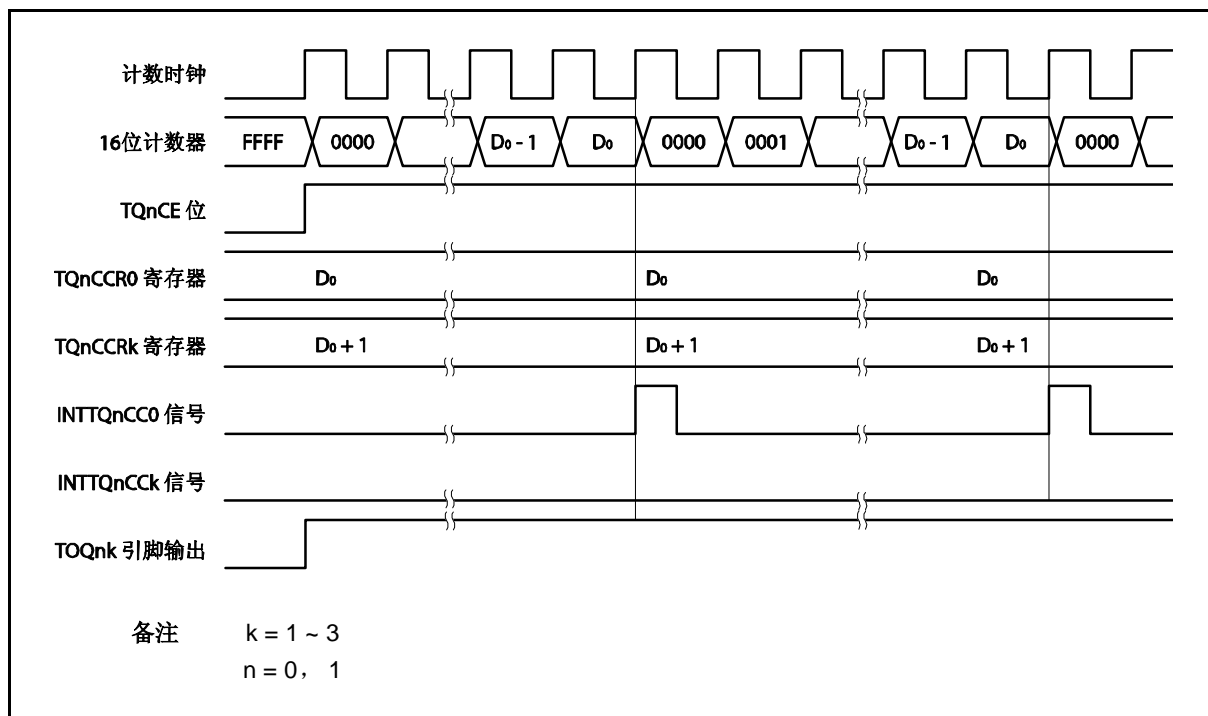
备注 m = 0 ~ 3
 n = 0, 1

(b) PWM 波形的 0%/100% 输出

若要输出 0% 的波形，则需将 TQnCCRk 寄存器设置为 0000H。如果 TQnCCR0 寄存器被设置为 FFFFH，那么 INTTQnCCk 信号将周期性产生。

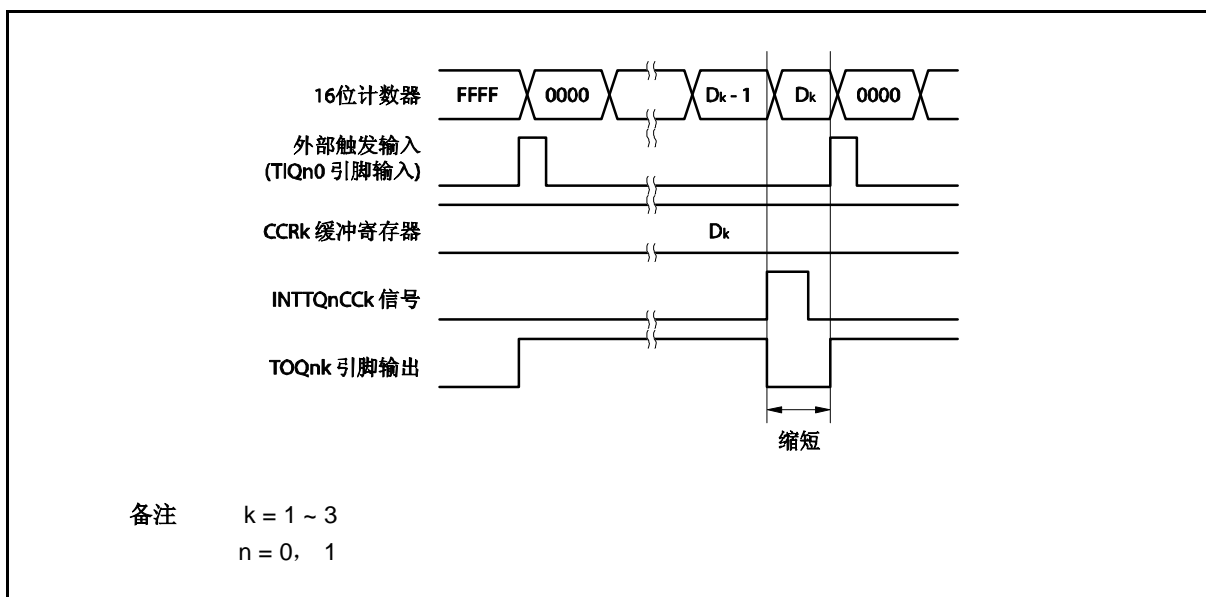


若要输出 100% 波形，则需将 TQnCCRk 寄存器设置为 (TQnCCR0 寄存器值 + 1)。TQnCCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。

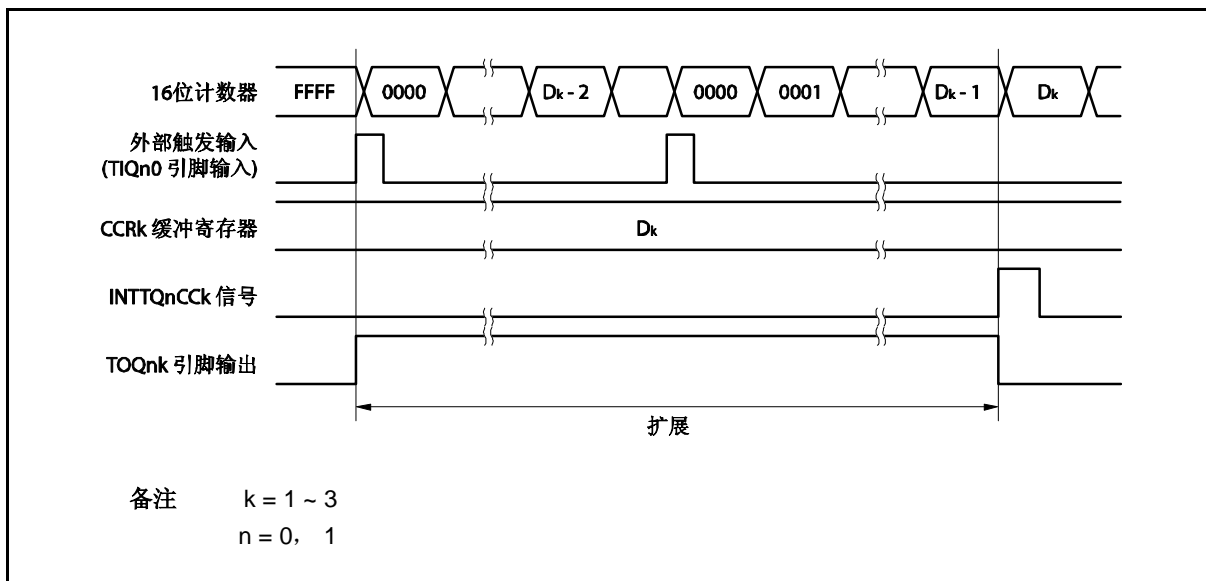


(c) 触发检测和 CCRk 缓冲寄存器相等之间的冲突

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

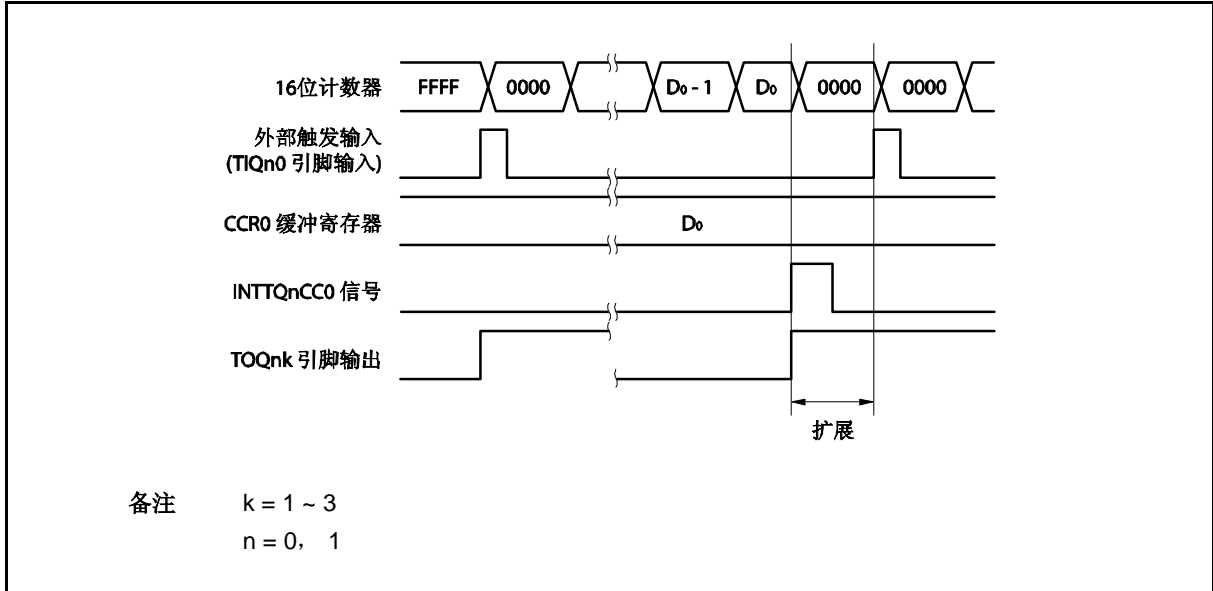


如果在 INTTQnCCk 信号产生之前检测到触发信号，那么 INTTQnCCk 信号将不会产生，此时，16 位计数器被清零并重新开始计数。TOQnk 引脚的输出也会继续保持为有效电平。这种情况下，PWM 波形的有效电平时间就会被延长。

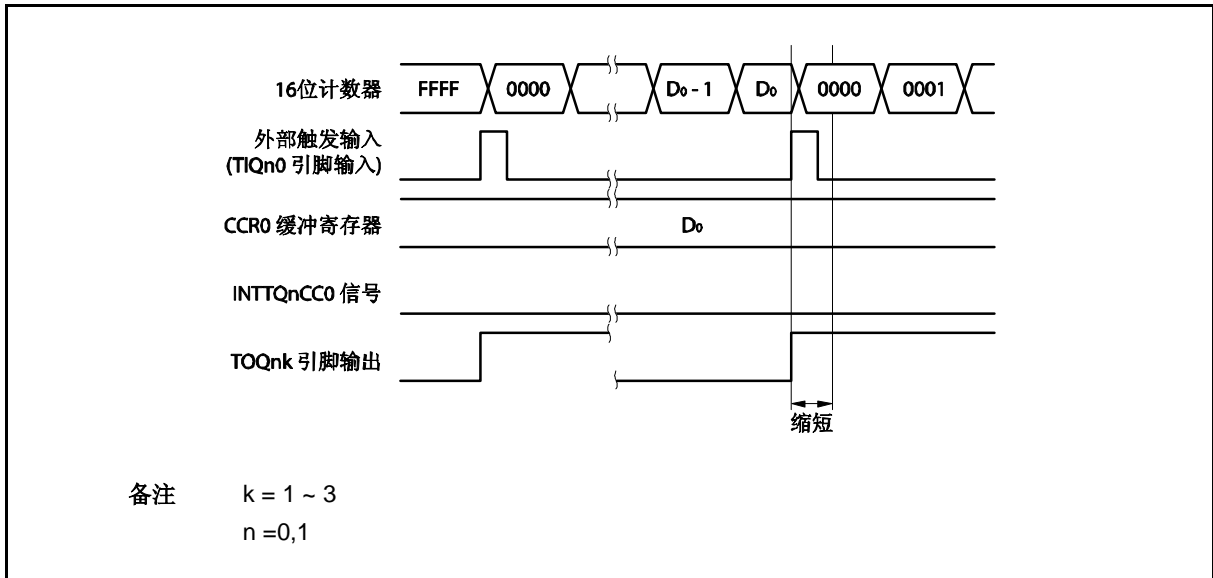


(d) 触发检测和 CCR0 缓冲寄存器相等之间的冲突

如果触发信号紧随 INTTQnCC0 中断信号的产生被检测，那么 16 位计数器将被清零并重新开始计数。因此，TOQnk 引脚输出的有效电平时间就会被延长（从 INTTQnCC0 信号产生到触发信号被检测的时间）。

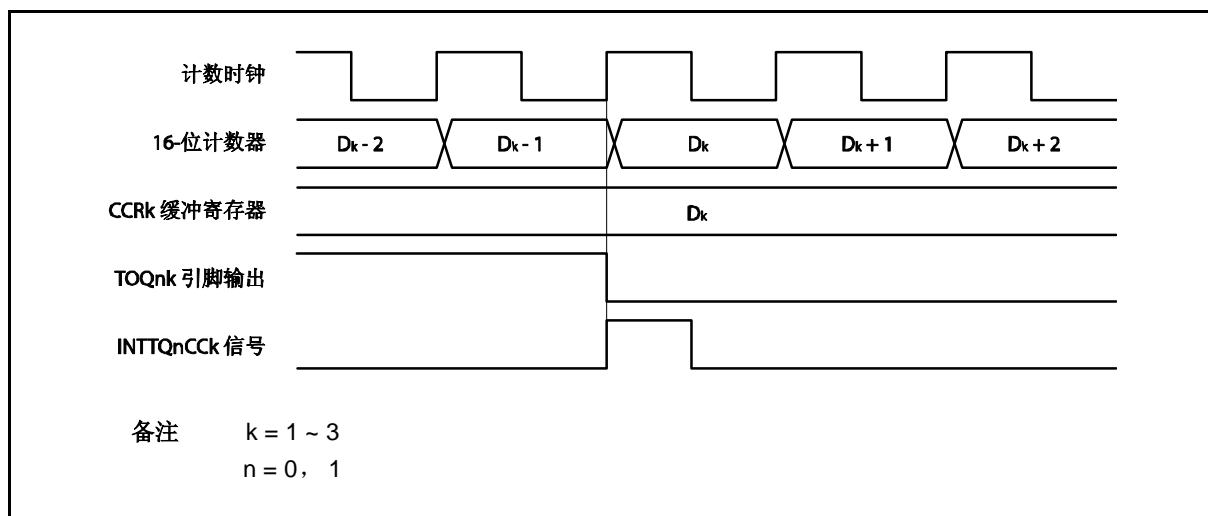


如果在 INTTQnCC0 信号产生之前检测到触发信号，那么 INTTQnCC0 信号将不会产生。16 位计数器将被清零并重新开始计数，TOQnk 引脚输出也会改变为有效电平。这种情况下，PWM 波形的非有效电平时间就会被缩短。



(e) 比较相等中断请求信号的产生时序(INTTQnCCK)

外部触发脉冲输出模式下，INTTQnCCK 信号的产生时序与其它模式下 INTTQnCCK 信号的产生时序有所不同，这里，INTTQnCCK 信号将在 16 位计数器的计数值与 CCRk 缓冲寄存器的值相等时产生。



通常情况下，INTTQnCCK 信号会在 16 位计数器计数值与 CCRk 缓冲寄存器的值相等之后的下一个计数时钟同步产生。

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

7.5.4 单次脉冲输出模式 (TQnMD2 ~ TQnMD0 位 = 011)

在单次脉冲输出模式中，TQnCTL0.TQnCE 位被置 1 后，16 位定时器/事件计数器 Q 进入等待触发信号的状态。当检测到一个外部触发输入信号的有效沿时，16 位定时器/事件计数器 Q 开始计数，并由 TOQn1 ~ TOQn3 引脚输出单脉冲波形。

也可以使用软件触发代替外部触发来开始脉冲的输出。当使用软件触发时，TOQn0 引脚会在 16 位计数器计数时输出有效电平，在计数器停止计数时(等待触发时)输出非有效电平。

图 7-20. 单脉冲输出模式的配置图

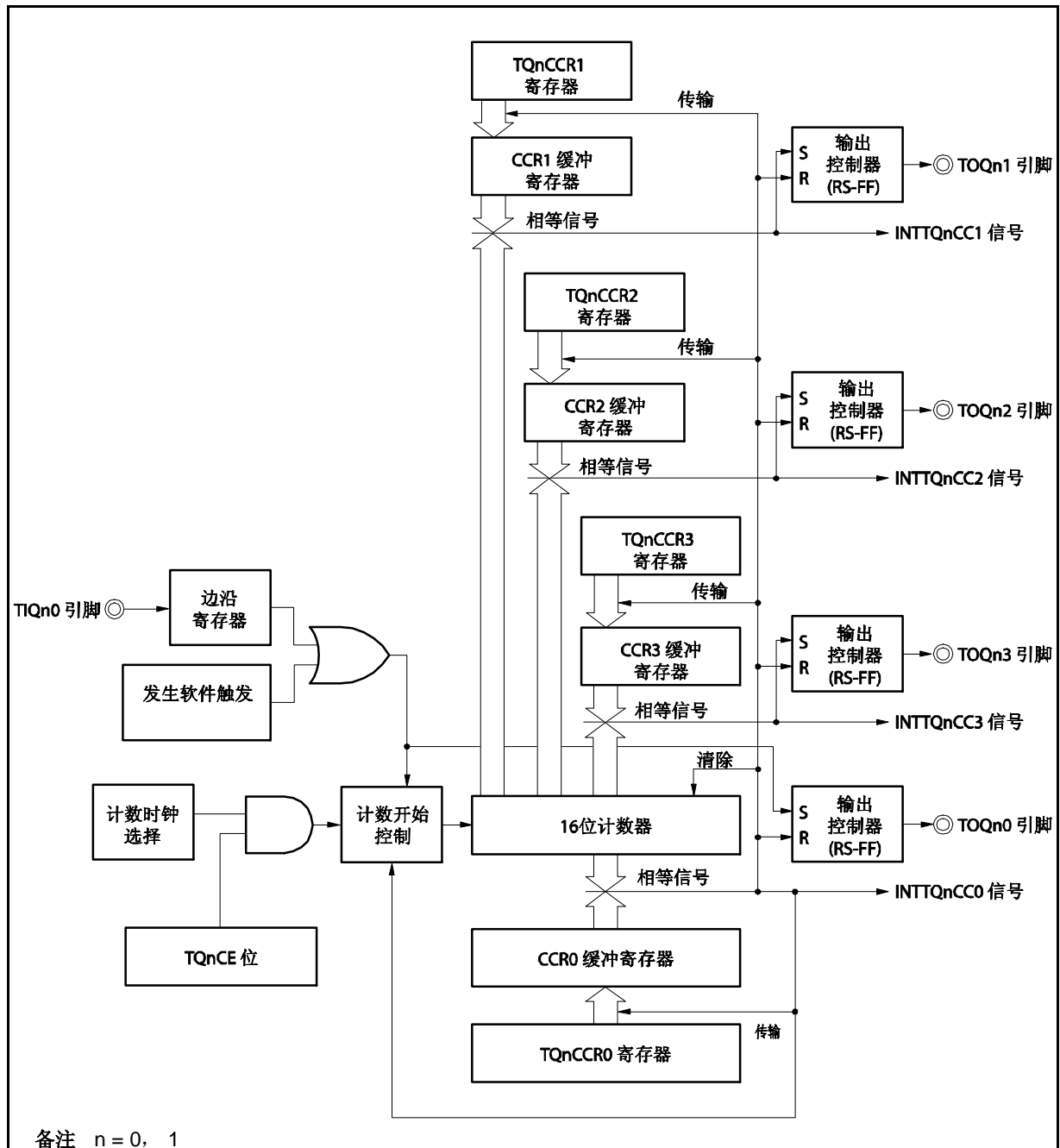
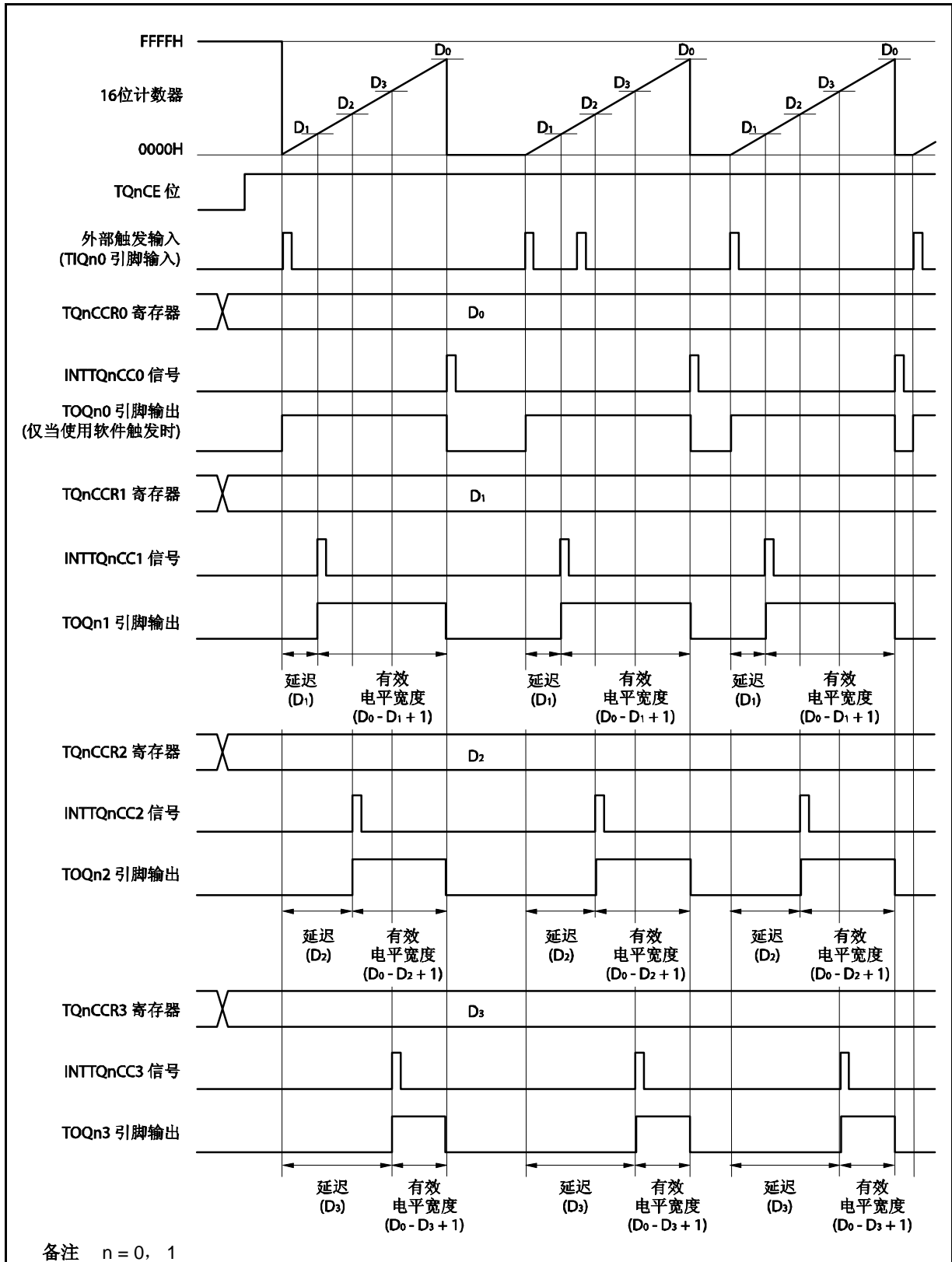


图 7-21. 单脉冲输出模式的基本时序



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

单脉冲波形的输出延迟时间以及有效电平宽度可由以下算式得出。

$$\text{输出延迟时间} = (\text{TQnCCRk 寄存器的设置值}) \times \text{计数时钟周期}$$

$$\text{有效电平宽度} = (\text{TQnCCR0 寄存器的值} - \text{TQnCCRk 寄存器的值} + 1) \times \text{计数时钟周期}$$

比较相等中断请求信号 INTTQnCC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器中的值相等时的下一个计数时钟时产生。比较相等中断请求信号 INTTQnCCk 会在 16 位计数器的计数值与 CCRk 缓冲寄存器中的值相等时产生。

触发源可以从外部触发输入信号的有效沿和设置软件触发(TQnCTL1.TQnEST 位 = 1)置 1 的方式进行选择。

备注 k = 1 ~ 3
n = 0, 1

图 7-22. 单脉冲输出模式的寄存器设置(1/3)

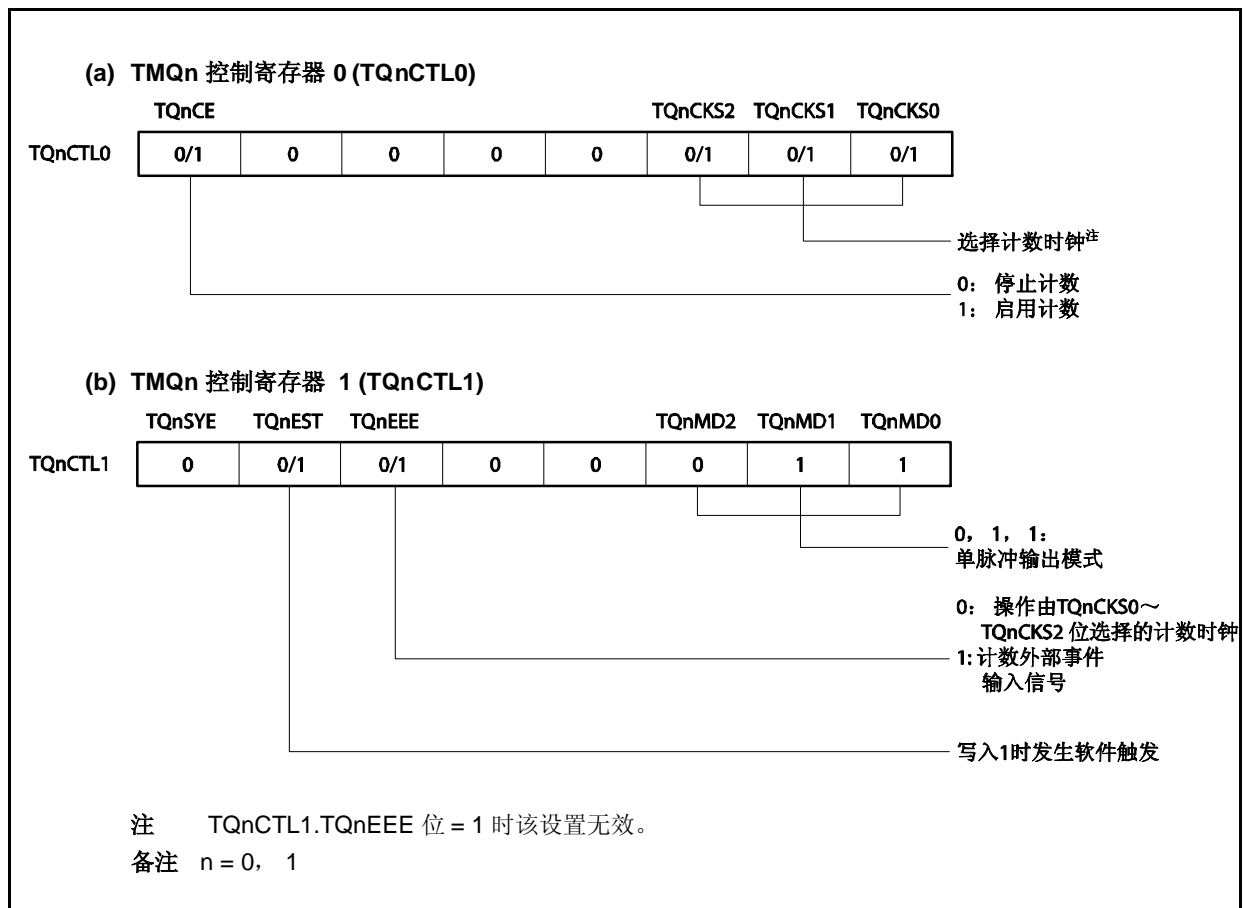


图 7-22. 单脉冲输出模式的寄存器设置(2/3)

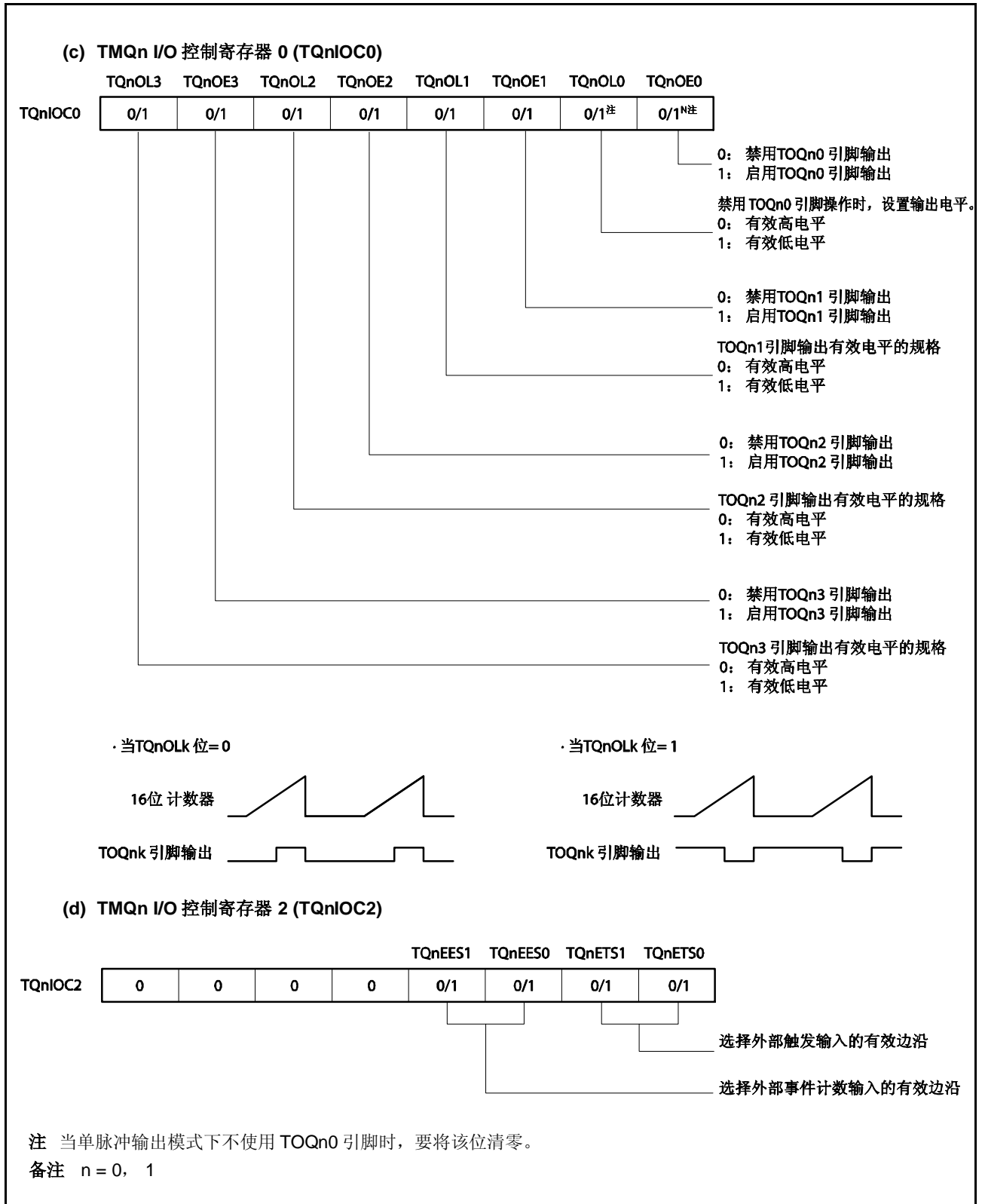


图 7-22. 单脉冲输出模式的寄存器设置(3/3)

(e) TMQn 计数器读取缓冲寄存器(TQnCNT)

可以通过读取 TQnCNT 寄存器来读取 16 位计数器的计数值。

(f) TMQn 捕捉/比较寄存器 0 ~ 3 (TQnCCR0 ~ TQnCCR3)

如果将 TQnCCR0 寄存器的值设置为 D_0 ，将 TQnCCRk 寄存器的值设置为 D_k ，那么单脉冲波形的有效电平宽度和输出延迟时间如下所示。

有效电平宽度 = $(D_0 - D_k + 1) \times$ 计数时钟周期

输出延迟时间 = $(D_k) \times$ 计数时钟周期

注意事项 如果在 TQnCCRk 寄存器中的设置值比 TQnCCR0 寄存器的值大，那么即使在单脉冲输出模式下，单脉冲也不被输出。

备注 1. 单脉冲输出模式下，并未使用 TMQn I/O 控制寄存器 1 (TQnIOC1) 和 TMQn 选项寄存器 0 (TQnOPT0)。

2. $k = 1 \sim 3$

$n = 0, 1$

<R>

(1) 单脉冲输出模式的操作流程

图 7-23. 单脉冲输出模式的软件操作流程(1/2)

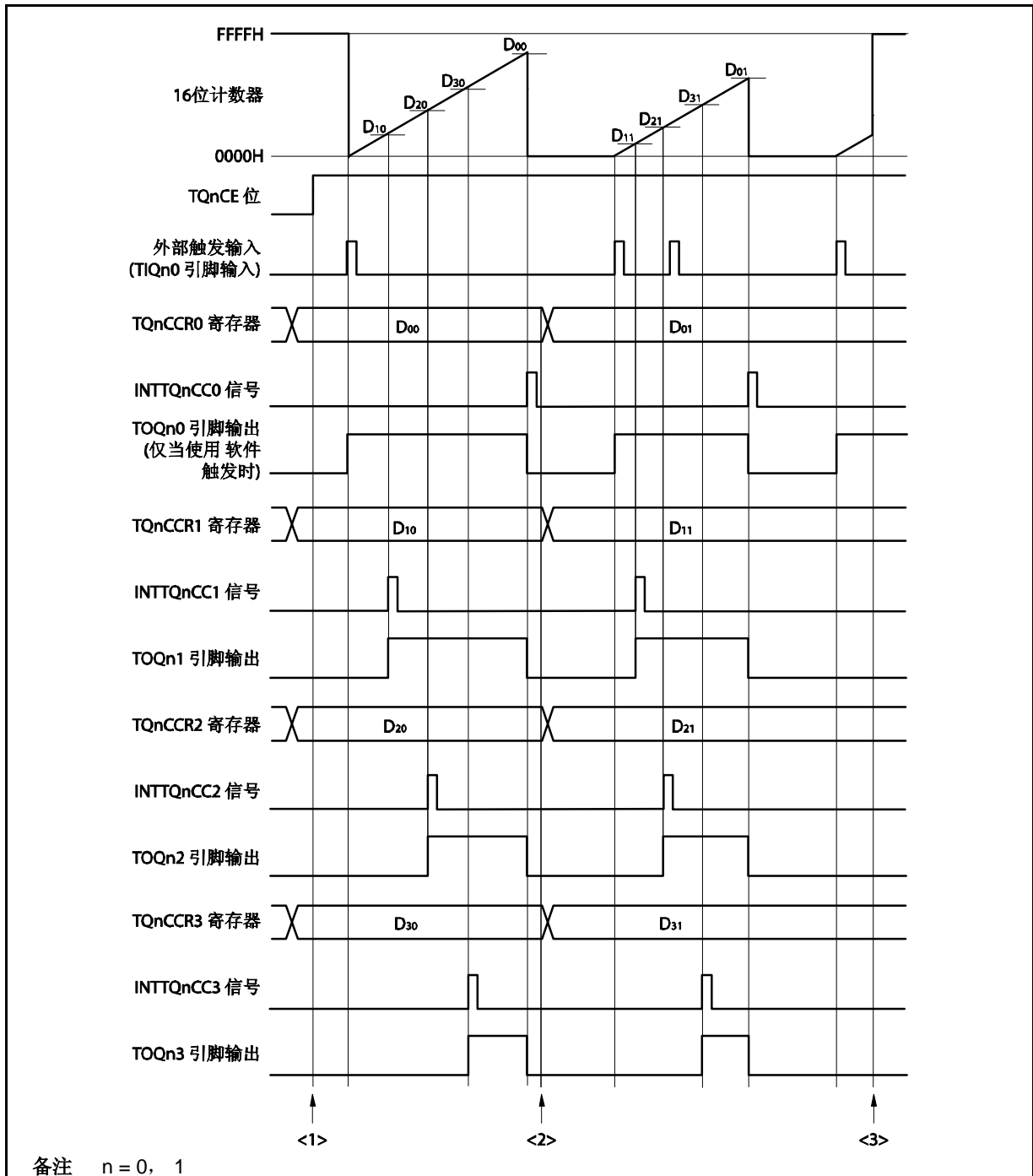
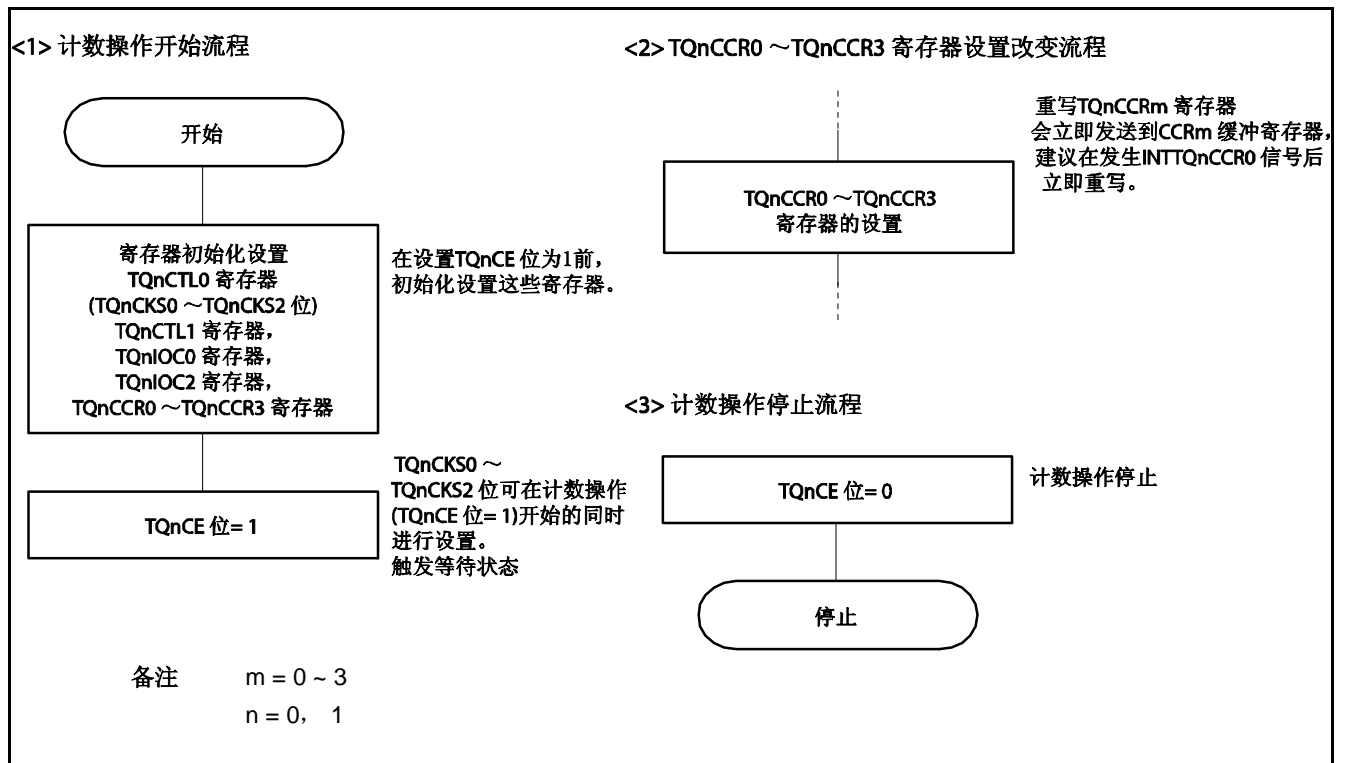


图 7-23. 单脉冲输出模式的软件操作流程(2/2)

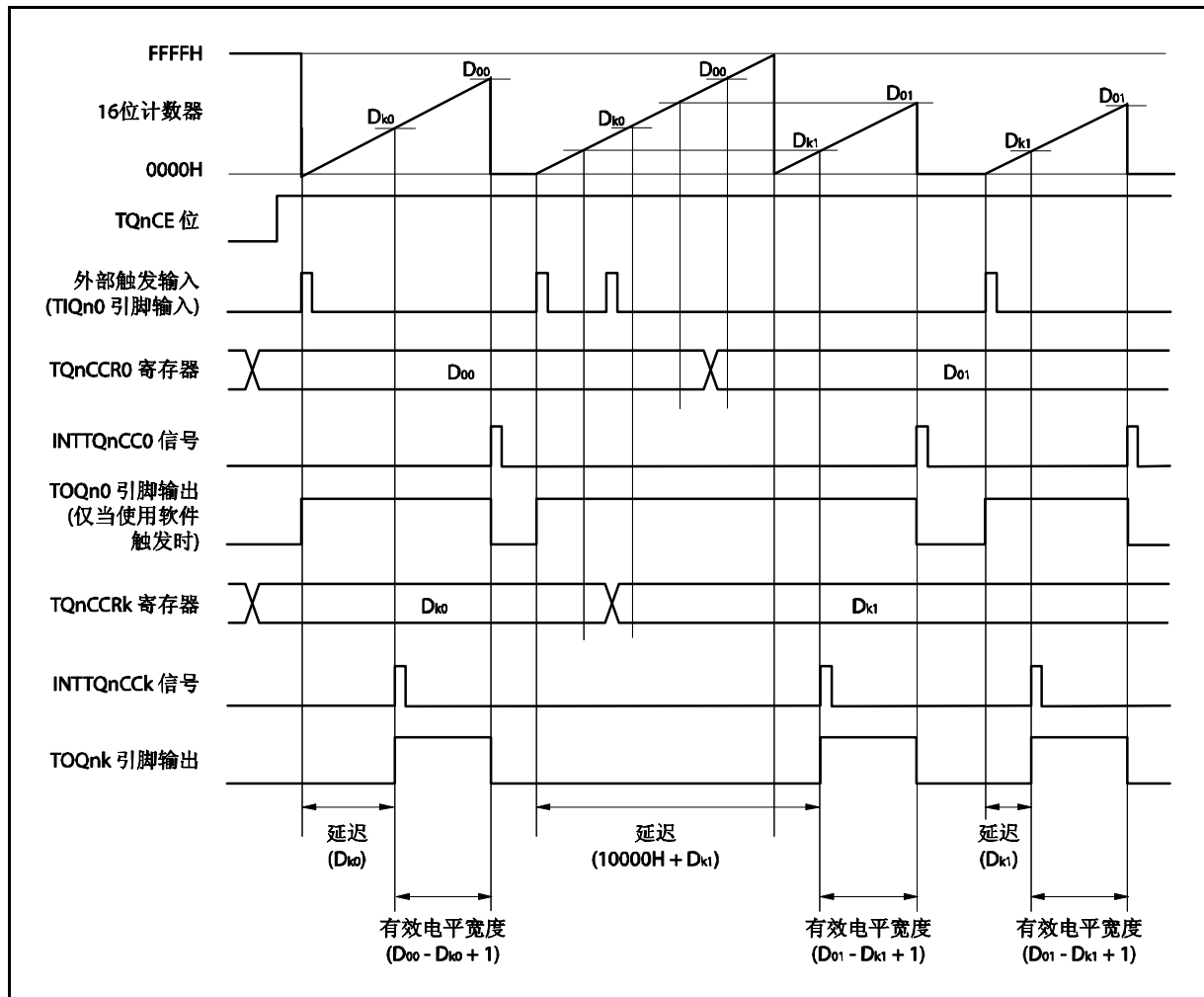


(2) 单脉冲输出模式的操作时序

(a) 改变 TQnCCRm 寄存器时需要注意的事项

将 TQnCCRm 寄存器的值减小时，要先停止计数，再对设定值进行改变。

如果在计数过程中将 TQnCCR0 寄存器的值减小，那么 16 位计数器可能产生溢出。



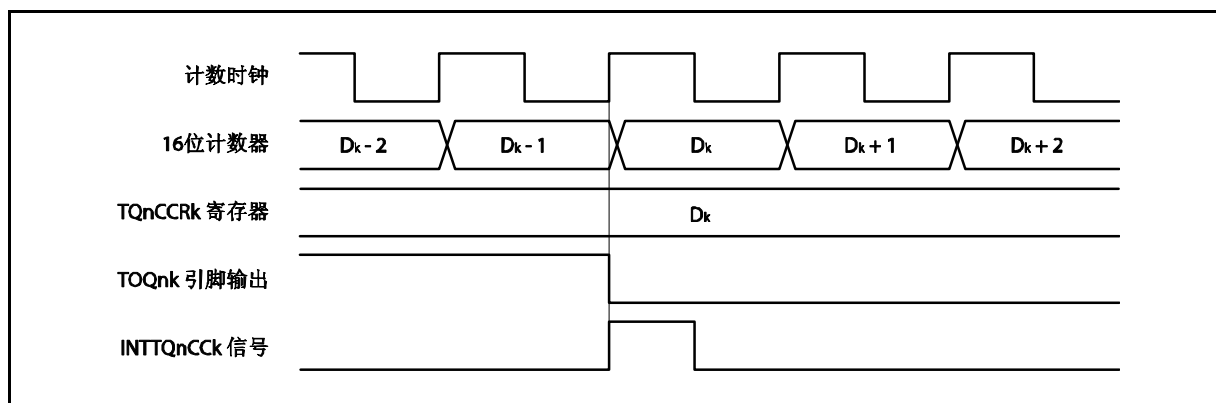
当将 TQnCCR0 寄存器的值从 D_{00} 改写为 D_{01} ，TQnCCRk 寄存器的值从 D_{k0} 改写为 D_{k1} ，这里假设 $D_{00} > D_{01}$ 且 $D_{k0} > D_{k1}$ ，如果 TQnCCRk 寄存器被改写时，16 位计数器的计数值大于 D_{k1} 且小于 D_{k0} ，并且 TQnCCR0 寄存器被改写时，计数器的计数值大于 D_{01} 且小于 D_{00} ，两个设定值都将在改写后立即生效并与计数值进行比较。此时计数器会计数到 FFFFH 然后从 0000H 重新开始计数。当计数值与 D_{k1} 相等时，将产生 INTTQnCCk 信号，同时 TOQnk 引脚电平变为有效电平。当计数值与 D_{01} 相等时，将产生 INTTQnCC0 信号，同时 TOQnk 引脚电平变为非有效电平并且计数器停止计数。

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

备注 k = 1~3
n = 0, 1

(b) 比较相等中断请求信号的产生时序 (INTTQnCCK)

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



通常情况下，INTTQnCCK 信号会在 16 位计数器计数值与 TQnCCRk 寄存器的值相等之后的下一个计数时钟同步产生。

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

备注 $k = 1 \sim 3$
 $n = 0, 1$

7.5.5 PWM 输出模式(TQnMD2~TQnMD0 位 = 100)

PWM 输出模式下, TQnCTL0.TQnCE 位被置 1 后, TOQn1~TOQn3 引脚将输出 PWM 波形。
另外, TOQn0 引脚将输出一个以 PWM 波形周期的 2 倍为周期的脉冲信号。

图 7-24. PWM 输出模式的配置图

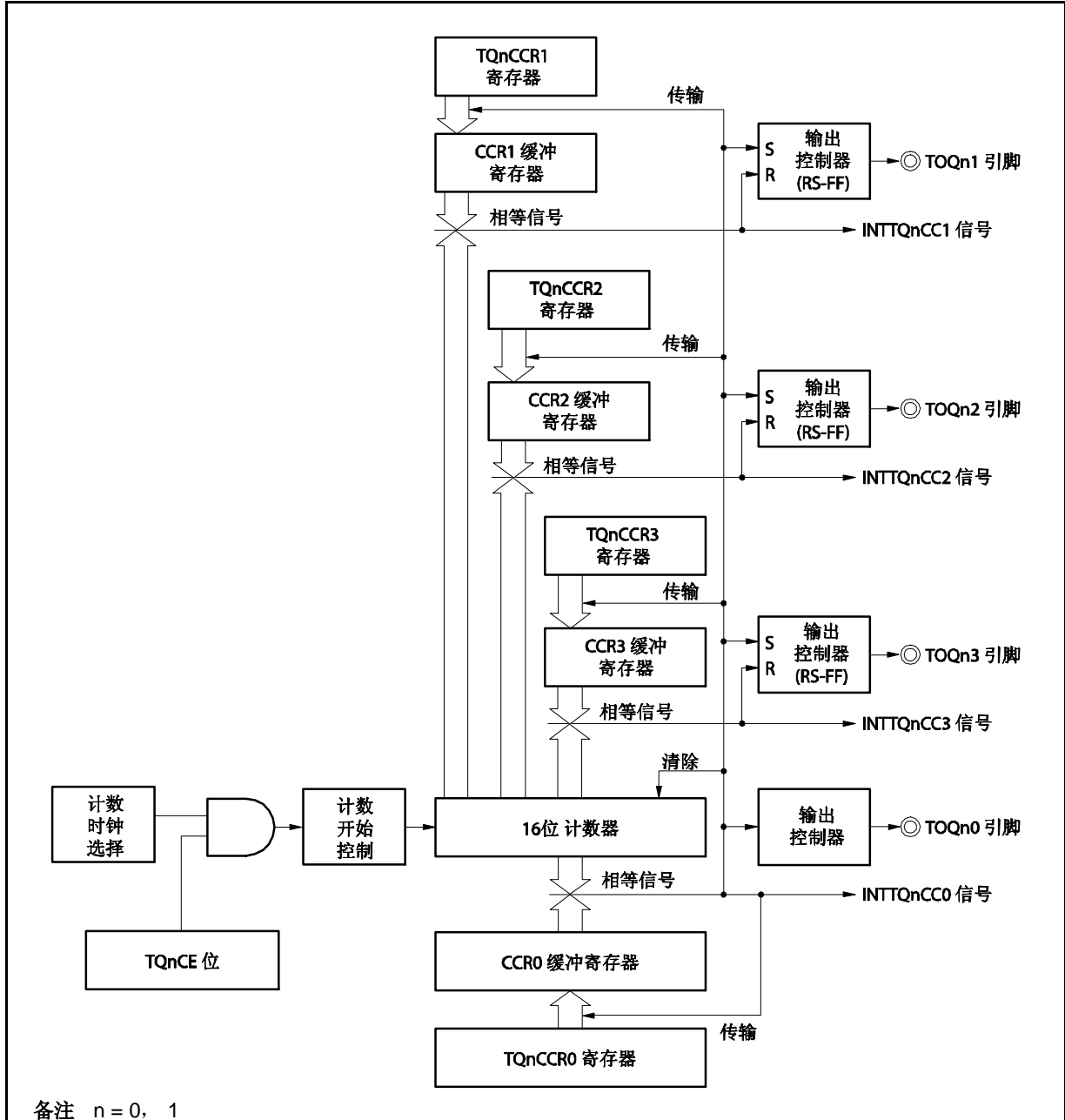
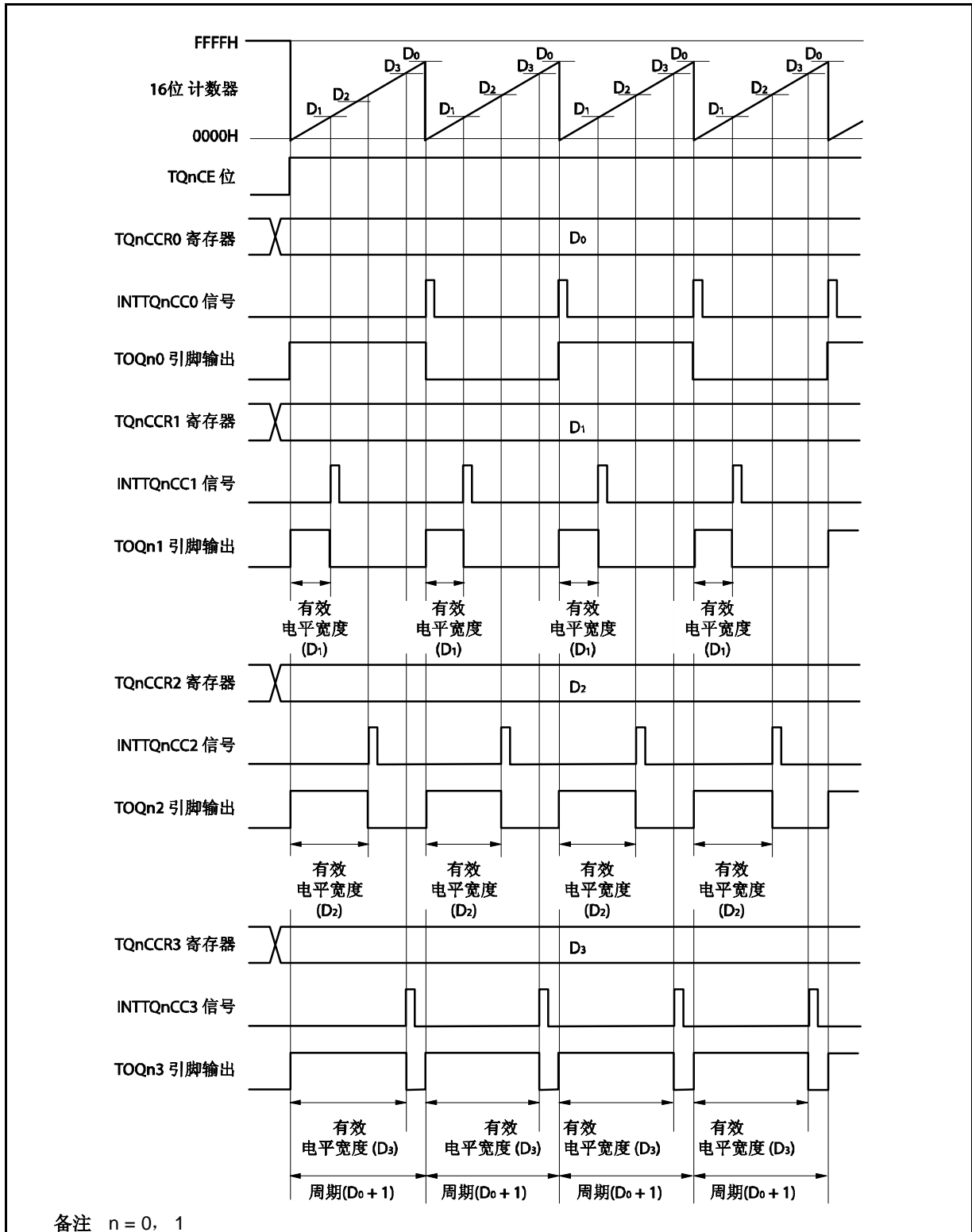


图 7-25. PWM 输出模式的基本时序



TQnCE 位被置 1 后，16 位计数器从 FFFFH 清零为 0000H 并开始计数，同时从 TOQnk 引脚输出 PWM 波形。PWM 波形的有效电平宽度，周期以及占空比系数可由以下算式得出。

$$\begin{aligned} \text{有效电平宽度} &= (\text{TQnCCRk 寄存器的值}) \times \text{计数时钟周期} \\ \text{周期} &= (\text{TQnCCR0 寄存器的值} + 1) \times \text{计数时钟周期} \\ \text{占空比系数} &= (\text{TQnCCRk 寄存器的值}) / (\text{TQnCCR0 寄存器的值} + 1) \end{aligned}$$

可以在计数器计数过程中，通过改写 TQnCCRm 寄存器的值来改变 PWM 波形的参数。改写后的数值将在 16 位计数器的计数值与当前 CCR0 缓冲寄存器的值相等之后生效，同时 16 位计数器也会被清为 0000H。

比较相等中断请求信号 INTTQnCC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器的值相等后的下一个计数时钟时产生，同时 16 位计数器将被清零。比较相等中断请求信号 INTTQnCCk 会在 16 位计数器的计数值与 CCRk 缓冲寄存器的值相等时产生。

备注 k = 1 ~ 3
m = 0 ~ 3
n = 0, 1

图 7-26. PWM 输出模式的寄存器设置(1/3)

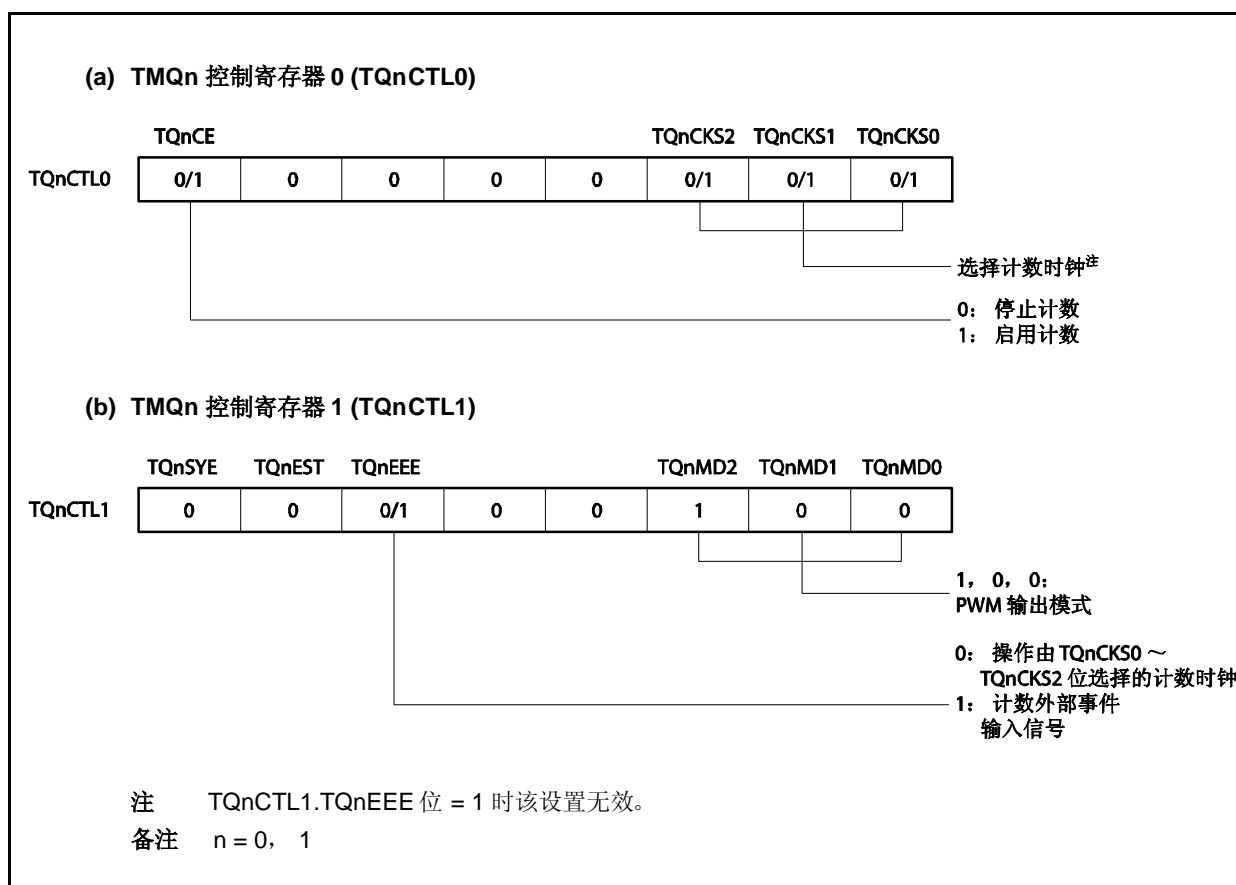


图 7-26. PWM 输出模式的寄存器设置(2/3)

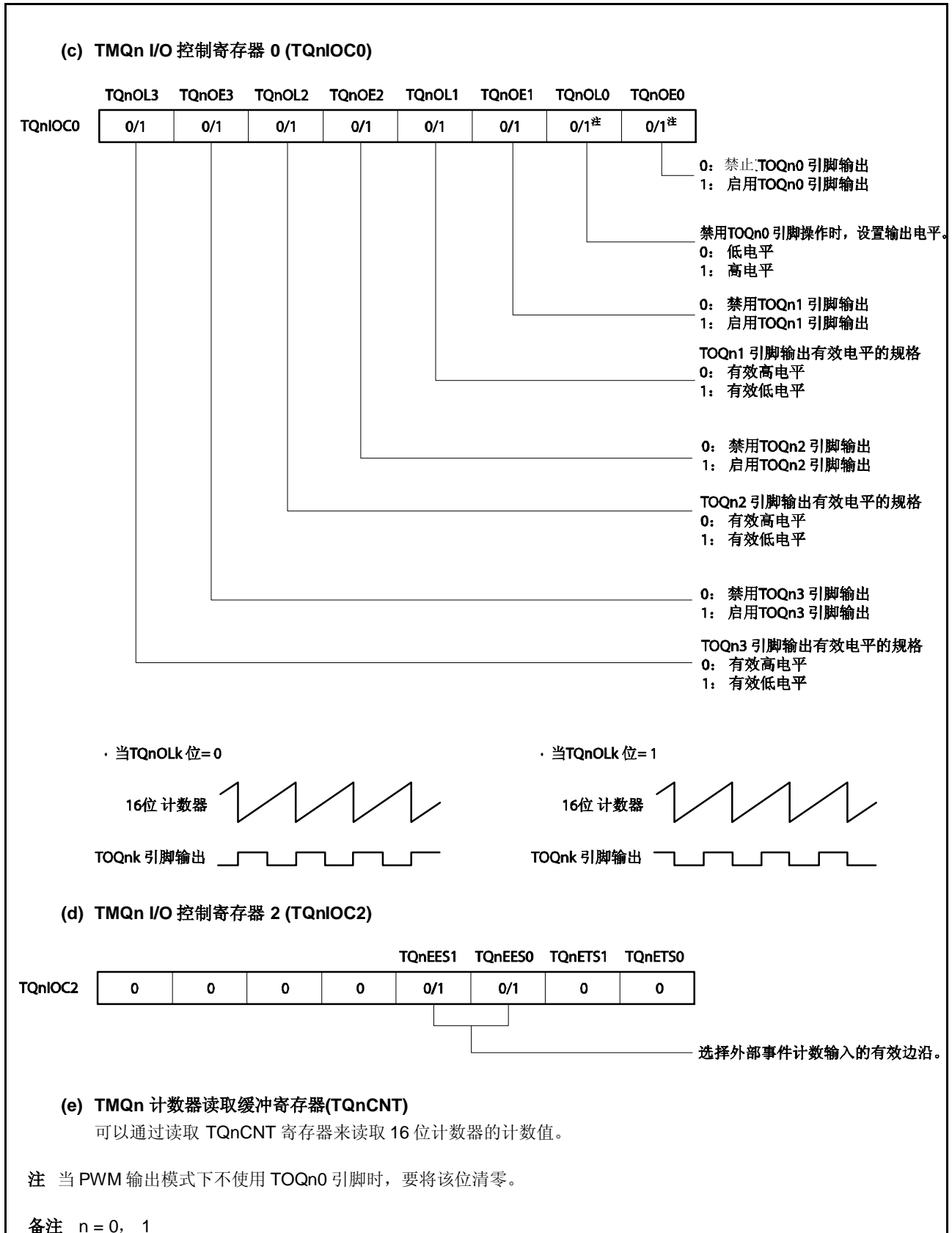


图 7-26. PWM 输出模式的寄存器设置(3/3)

(f) TMQn 捕捉/比较寄存器 0 ~ 3 (TQnCCR0~TQnCCR3)

如果将 TQnCCR0 寄存器的值设置为 D_0 ，将 TQnCCRk 寄存器的值设置为 D_k ，那么 PWM 波形的周期和有效电平如下所示。

周期 = $(D_0 + 1) \times$ 计数时钟周期

有效电平宽度 = $D_k \times$ 计数时钟周期

- 备注**
1. PWM 输出模式下，并未使用 TMQn I/O 控制寄存器 1 (TQnIOC1) 和 TMQn 选项寄存器 0 (TQnOPT0)。
 2. 通过写 TMQn 捕捉/比较寄存器 1 (TQnCCR1) 来使更新 TMQn 捕捉/比较寄存器 2 (TQnCCR2) 和 TMQn 捕捉/比较寄存器 3 (TQnCCR3) 有效。
 3. $n = 0, 1$

(1) PWM 输出模式的操作流程

图 7-27. PWM 输出模式的软件处理流程(1/2)

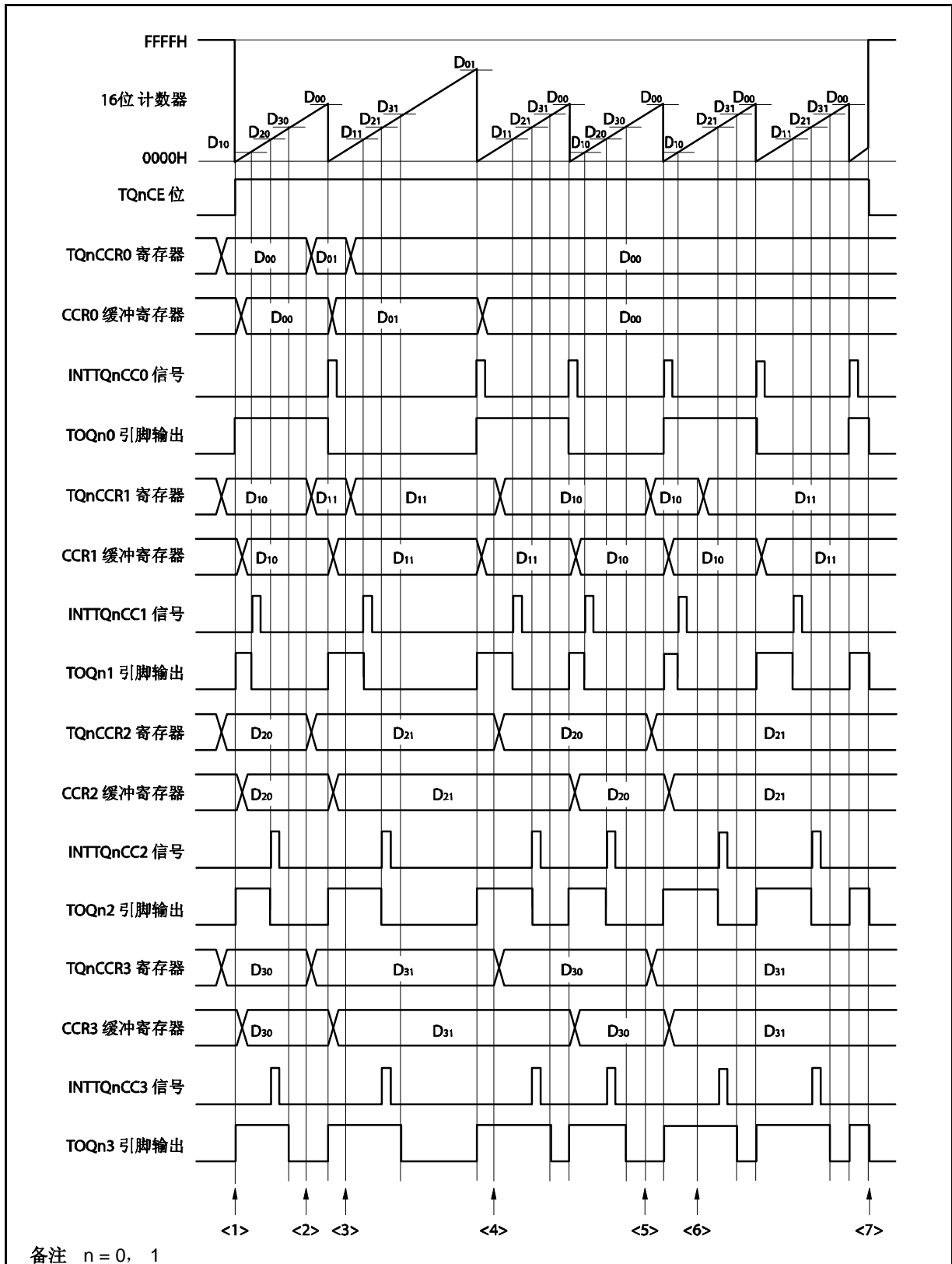
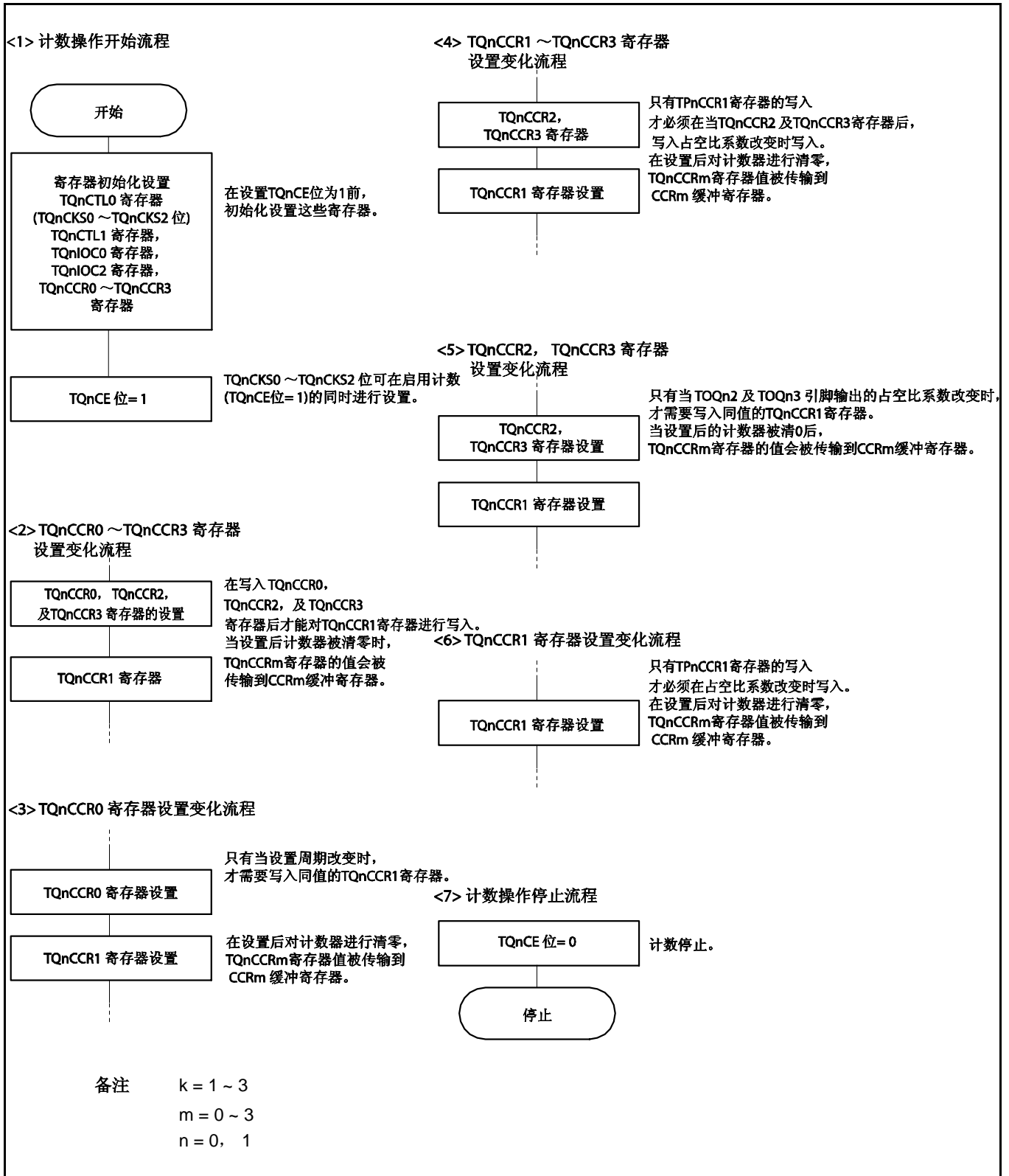


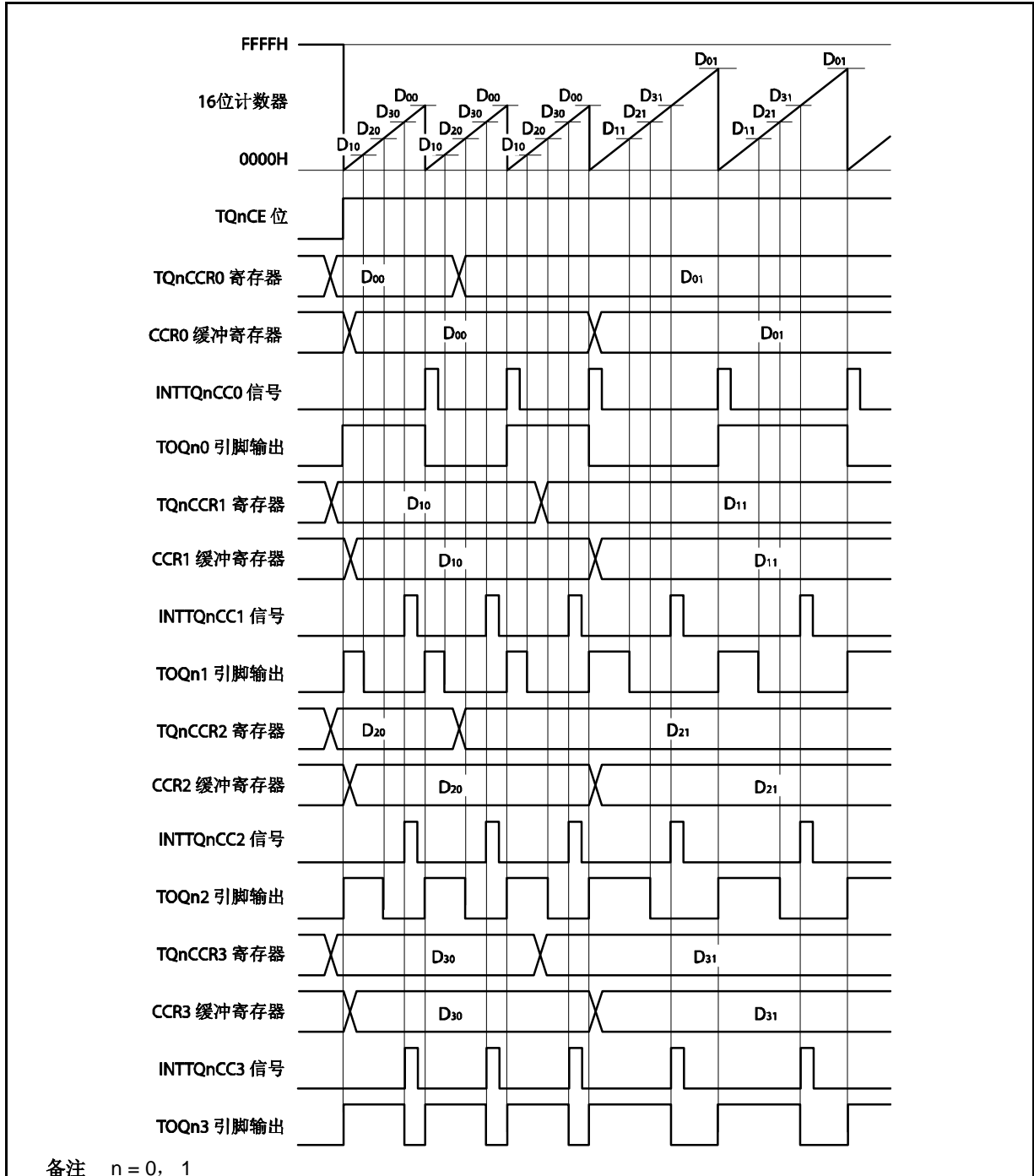
图 7-27. PWM 输出模式的软件处理流程(2/2)



(2) PWM 输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

计数器计数过程中，若改变 PWM 波形，则要在最后进行对 TQnCCR1 寄存器的写入操作。
对 TQnCCR1 寄存器进行写入操作之后，若需再次更改 TQnCCRk 寄存器的值，须等待下一个 INTTQnCC0 信号被检测到后，再进行相关操作。



若要将 TQnCCRm 寄存器中设置的数据传送到 CCRm 缓冲寄存器，必须对 TQnCCR1 寄存器进行写入操作。

同时改变 PWM 波形的周期和有效电平时，先将周期值写入 TQnCCR0 寄存器，然后向 TQnCCR2 和 TQnCCR3 寄存器写入有效电平宽度，最后向 TQnCCR1 寄存器写入有效电平宽度。

只改变 PWM 波形的有效电平宽度（占空比系数）时，首先向 TQnCCR2 和 TQnCCR3 寄存器设置有效电平然后再向 TQnCCR1 寄存器设置有效电平。

若只改变通过 TOQn1 引脚输出的 PWM 波形的有效电平宽度（占空比系数）时，只需要设置 TQnCCR1 寄存器。

若只改变通过 TOQn2 和 TOQn3 引脚输出的 PWM 波形的有效电平宽度（占空比系数）时，先要向 TQnCCR2 和 TQnCCR3 寄存器设置有效电平宽度，然后向 TQnCCR1 寄存器写入相同的值。

数据被写入 TQnCCR1 寄存器之后，16 位计数器被清零时，TQnCCRm 寄存器中的数值才会被同步传送到 CCRm 缓冲寄存器中，此时 CCRm 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

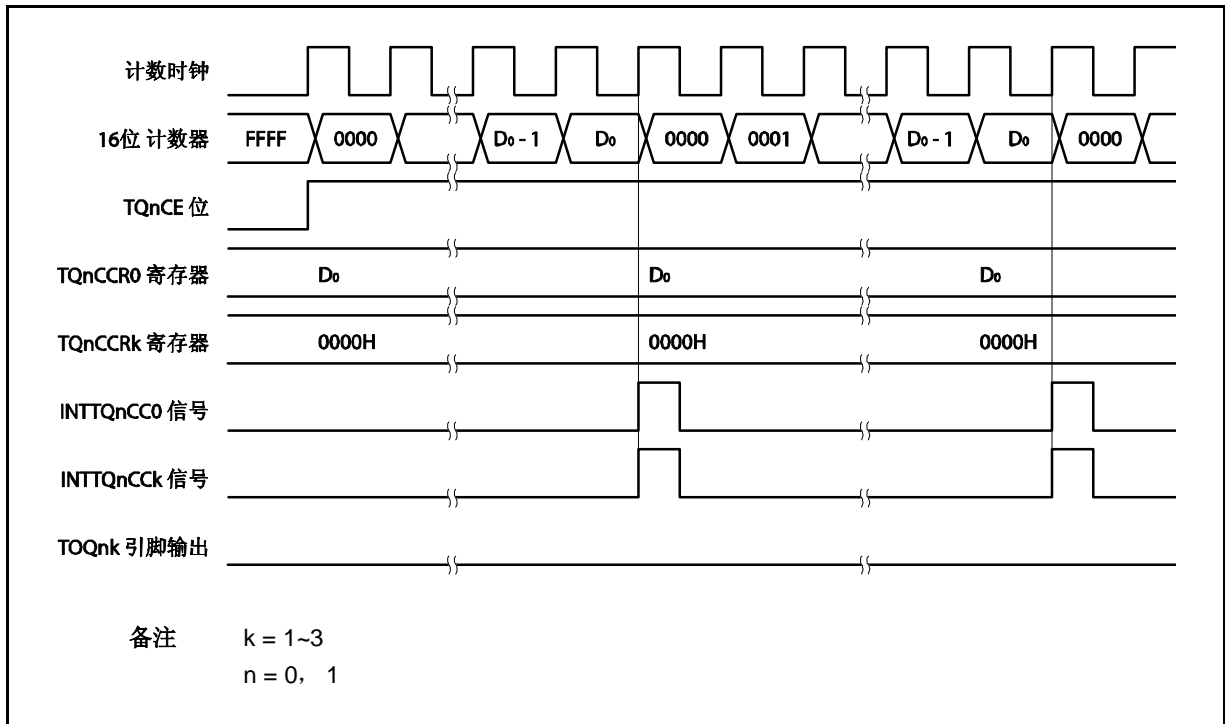
若只改变 PWM 波形的周期，首先要将周期设置于 TQnCCR0 寄存器，然后向 TQnCCR1 寄存器写入相同的值。

若要在写入 TQnCCR1 寄存器后再次写入 TQnCCR0 ~ TQnCCR3 寄存器，需要在产生 INTTQnCC0 信号后再进行此步操作。否则，CCRm 缓冲寄存器的值可能无法确定，因为数据从 TQnCCRm 寄存器传送到 CCRm 缓冲寄存器的时刻与 TQnCCRm 寄存器的写入时刻相冲突。

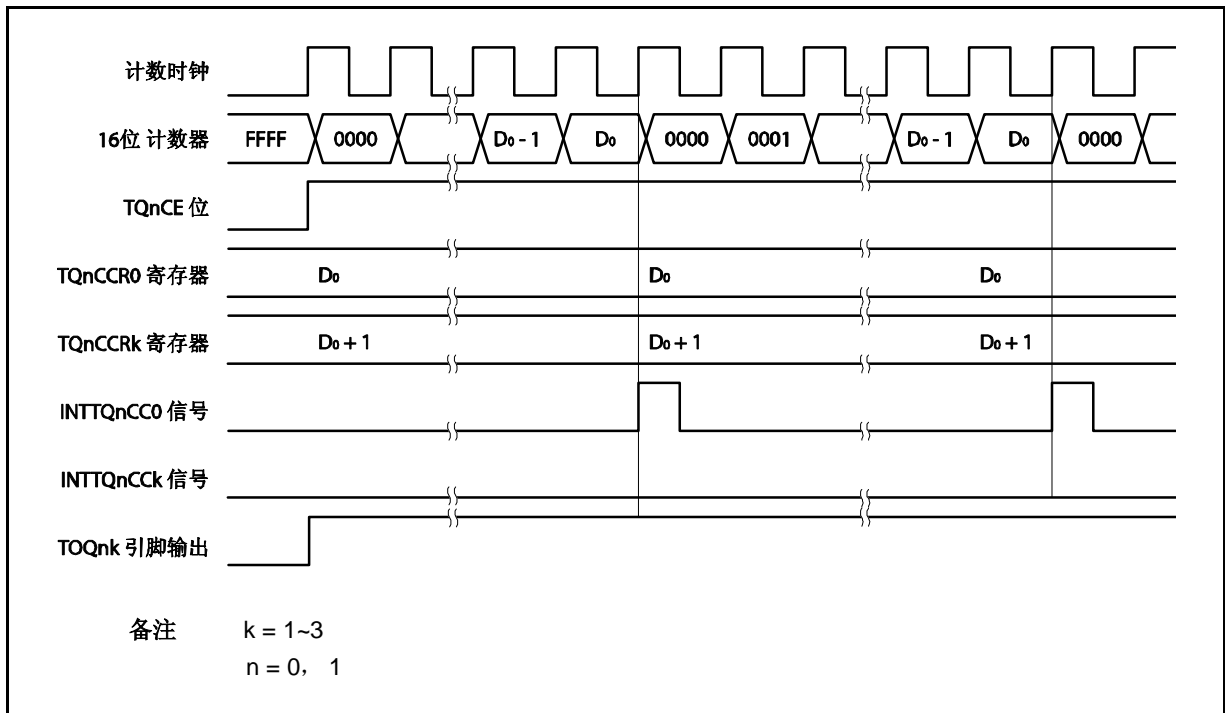
备注 m = 0 ~ 3
 n = 0, 1

(b) PWM 波形的 0%/100% 输出

若要输出 0% 的波形，则需将 TQnCCRk 寄存器设置为 0000H。如果 TQnCCR0 寄存器被设置为 FFFFH，那么 INTTQnCck 信号将周期性产生。

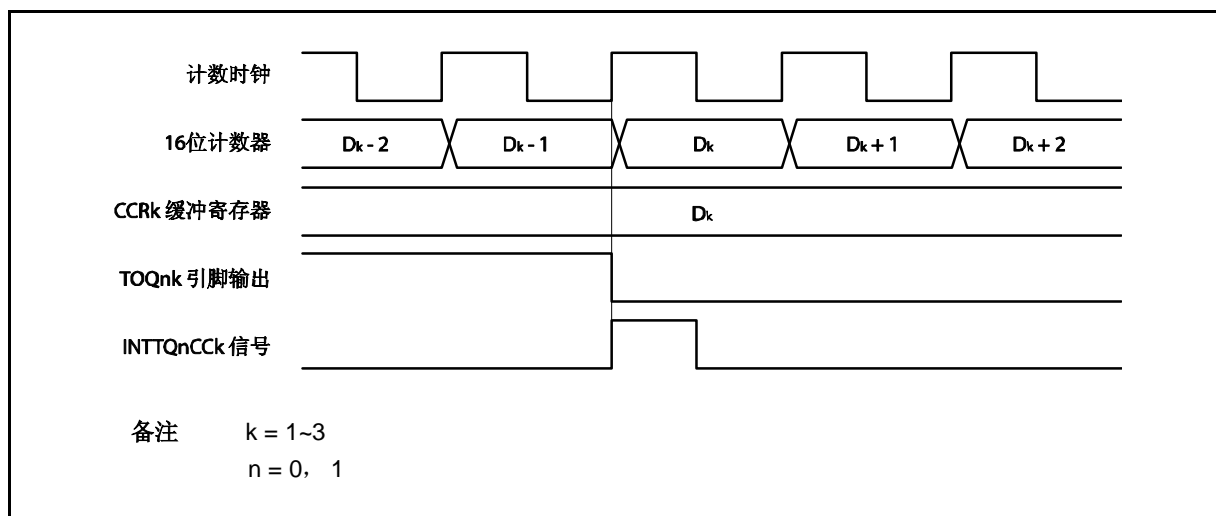


若要输出 100% 波形，则需将 TQnCCRk 寄存器设置为(TQnCCR0 寄存器值 + 1)。TQnCCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。



(c) 比较相等中断请求信号的产生时序(INTTQnCCk)

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



通常情况下，INTTQnCCk 信号会在 16 位计数器计数值与 TQnCCRk 寄存器的值相等之后的下一个计数时钟同步产生。

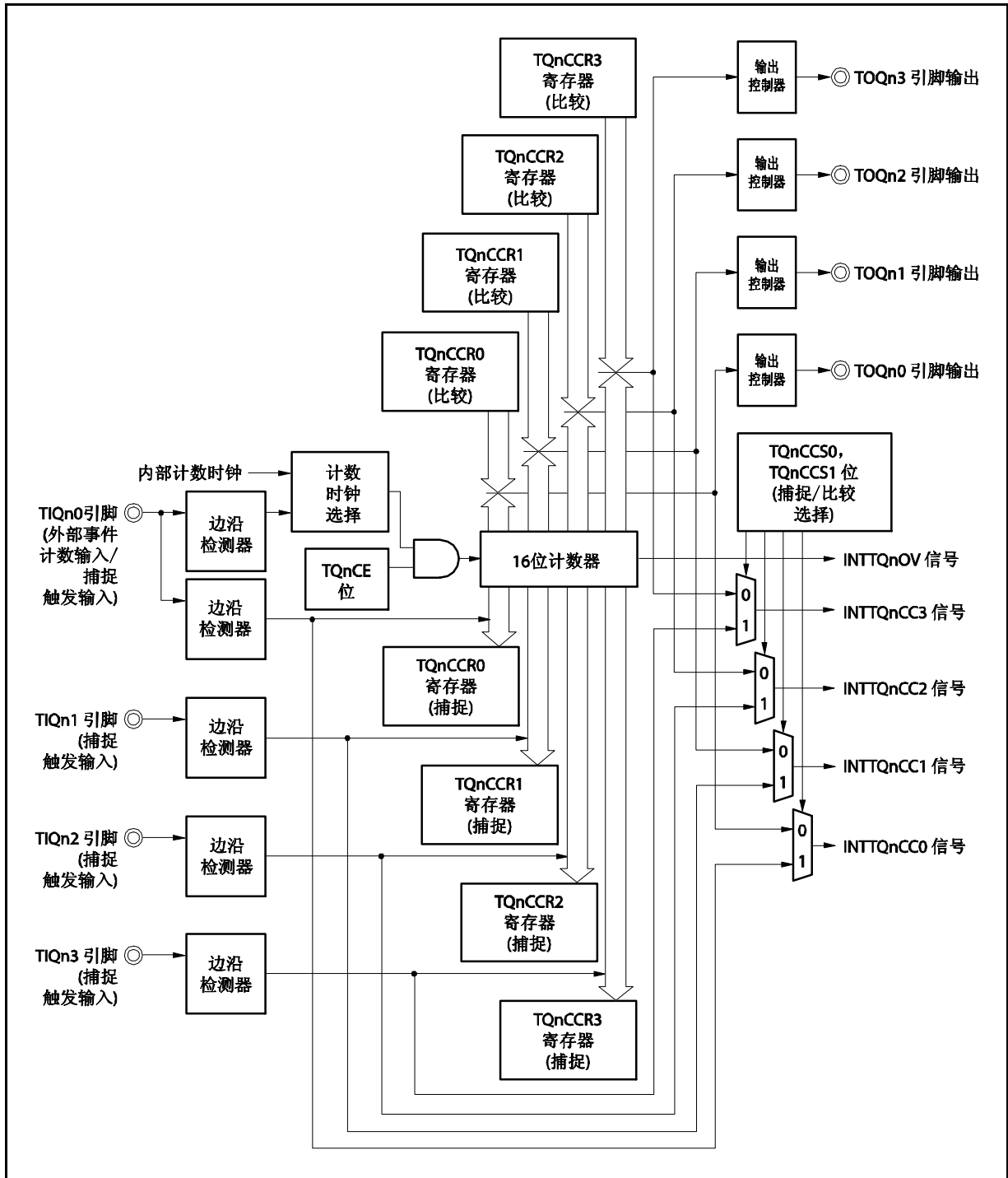
但在 PWM 输出模式下，INTTQnCCk 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOQnk 引脚输出信号变化的时序而改变的。

7.5.6 自由运行定时器模式 (TQnMD2~TQnMD0 位 = 101)

在自由运行定时器模式中, TQnCTL0.TQnCE 位被置 1 后, 16 位定时器/时间计数器 Q 开始计数。此时, 根据 TQnOPT0.TQnCCS0 和 TQnOPT0.TQnCCS1 位的设置 TQnCCRm 寄存器的功能可以在比较寄存器和捕捉寄存器之间进行选择。

备注 m = 0~ 3
n = 0, 1

图 7-28. 自由运行定时器模式的配置图

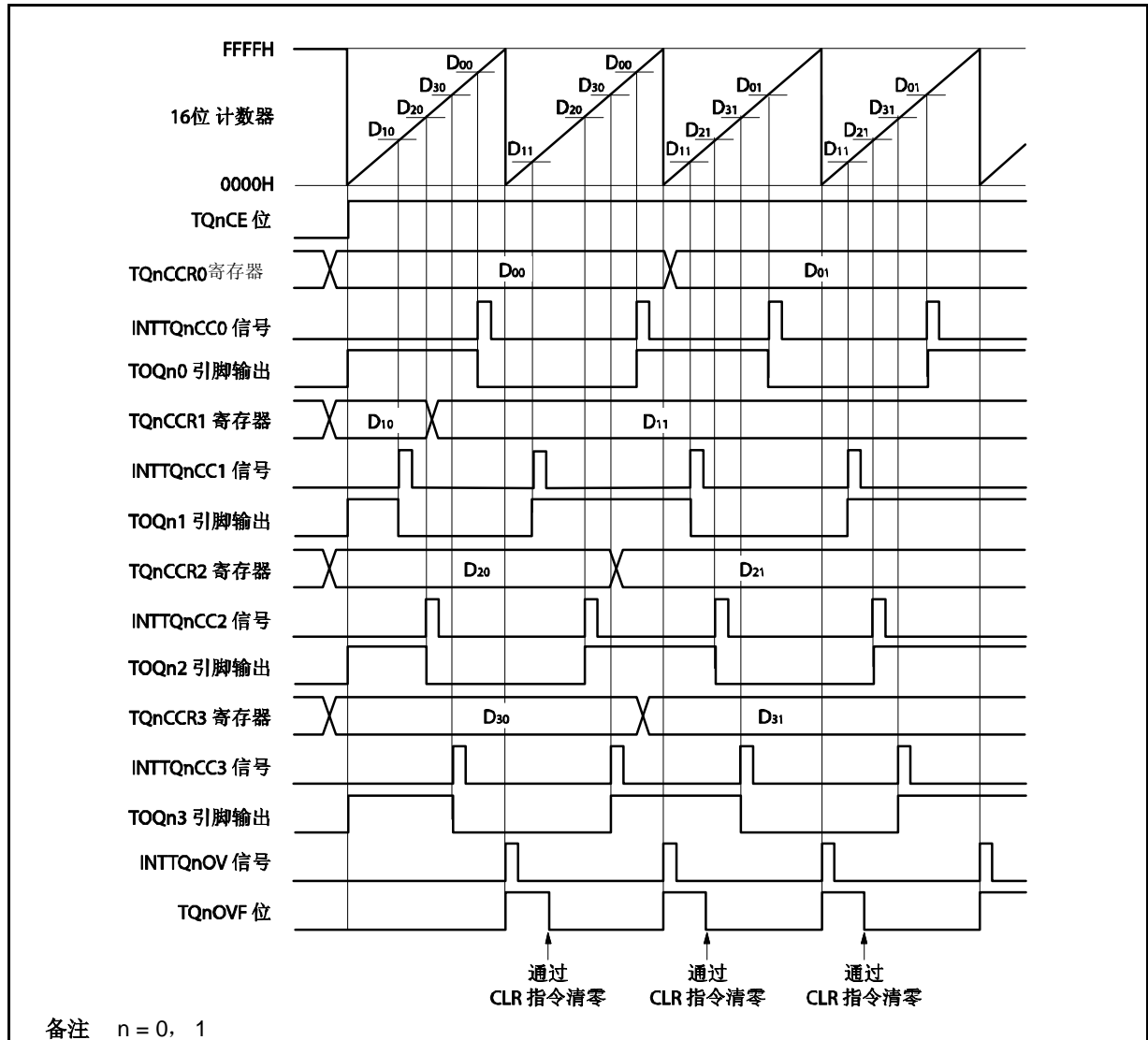


TQnCE 位被置 1 后, 16 位定时/事件计数器 Q 开始计数, 同时 TOQn0~TOQn3 引脚输出信号的电平反转。当 16 位计数器的计数值与 TQnCCRm 寄存器的值相等时将产生一个比较相等中断请求信号(INTTQnCCm), 同时 TOQnm 引脚的输出电平将反转。

此后, 16 位计数器将继续与计数时钟同步计数。当它计数到 FFFFH 时, 将在下一个计数时钟时产生一个溢出中断请求信号(INTTQnOV), 同时计数器被清零并重新开始计数。此时, 溢出标志(TQnOPT0.TQnOVF 位) 也会被置 1。请通过软件使用 CLR 指令将溢出标志位清零。

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

图 7-29. 自由运行定时器模式的基本时序 (比较功能)



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

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

图 7-30. 自由运行定时器模式的基本时序 (捕捉功能)

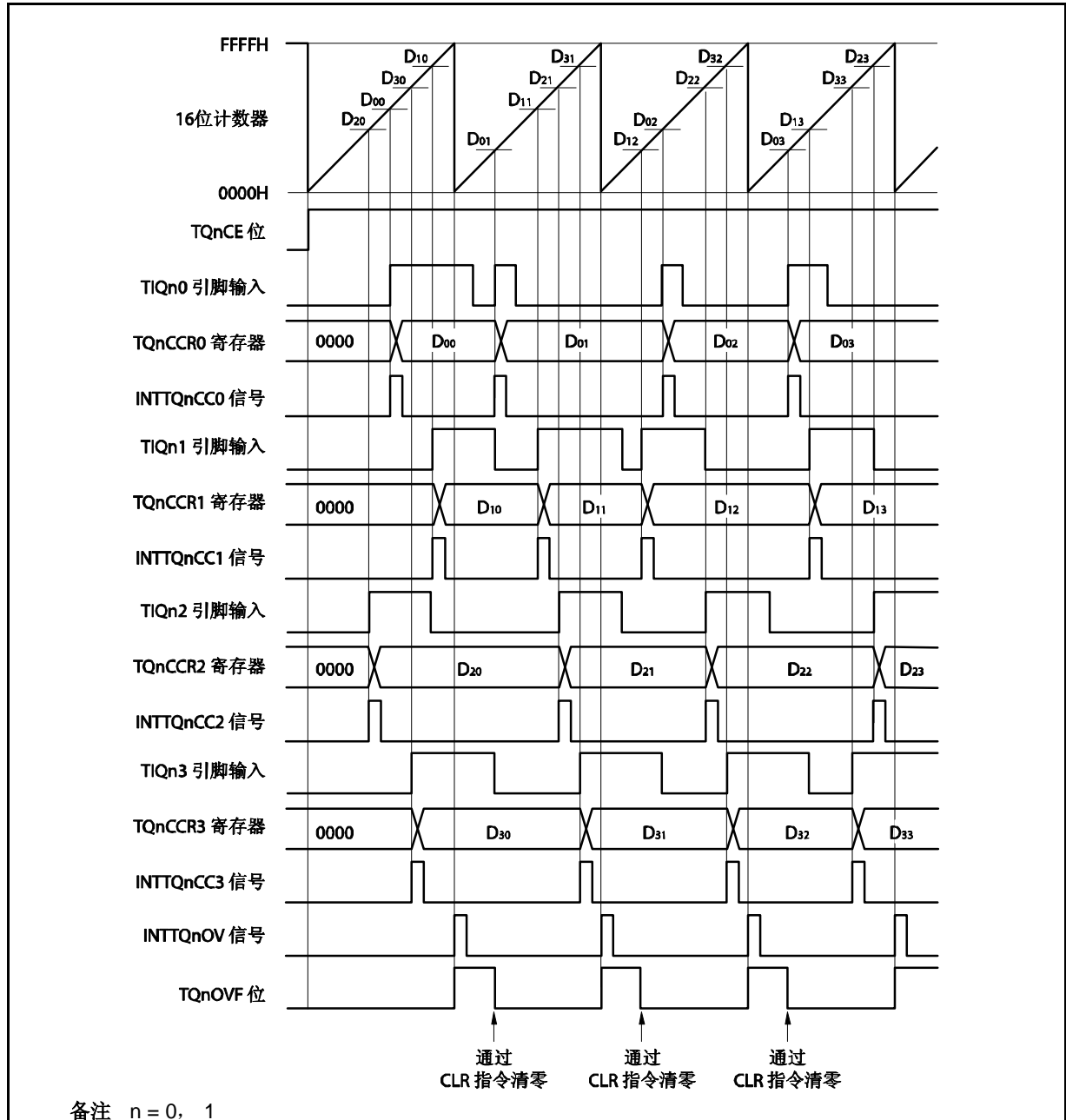


图 7-31. 自由运行定时器模式的寄存器设置(1/3)

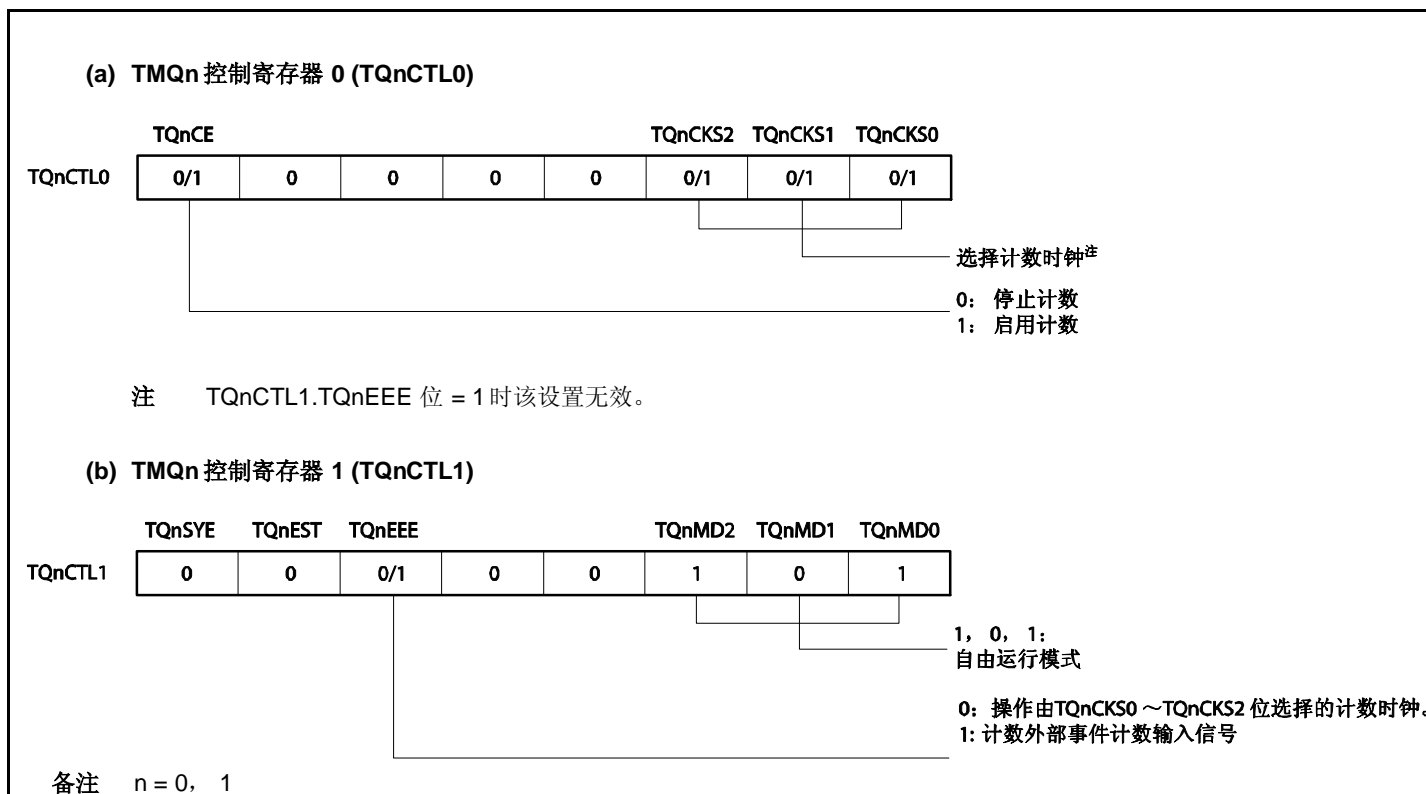


图 7-31. 自由运行定时器模式的寄存器设置 (2/3)

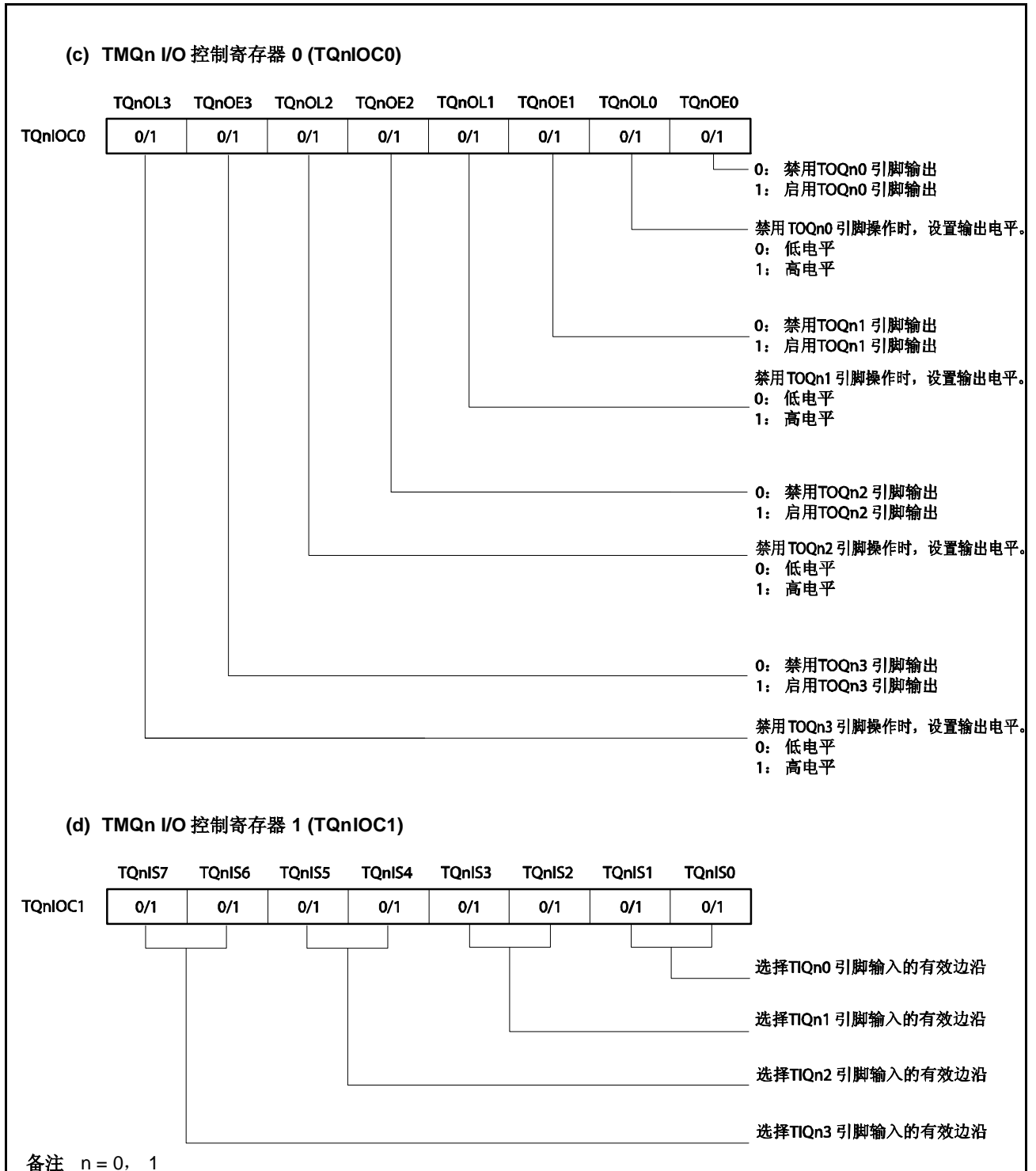
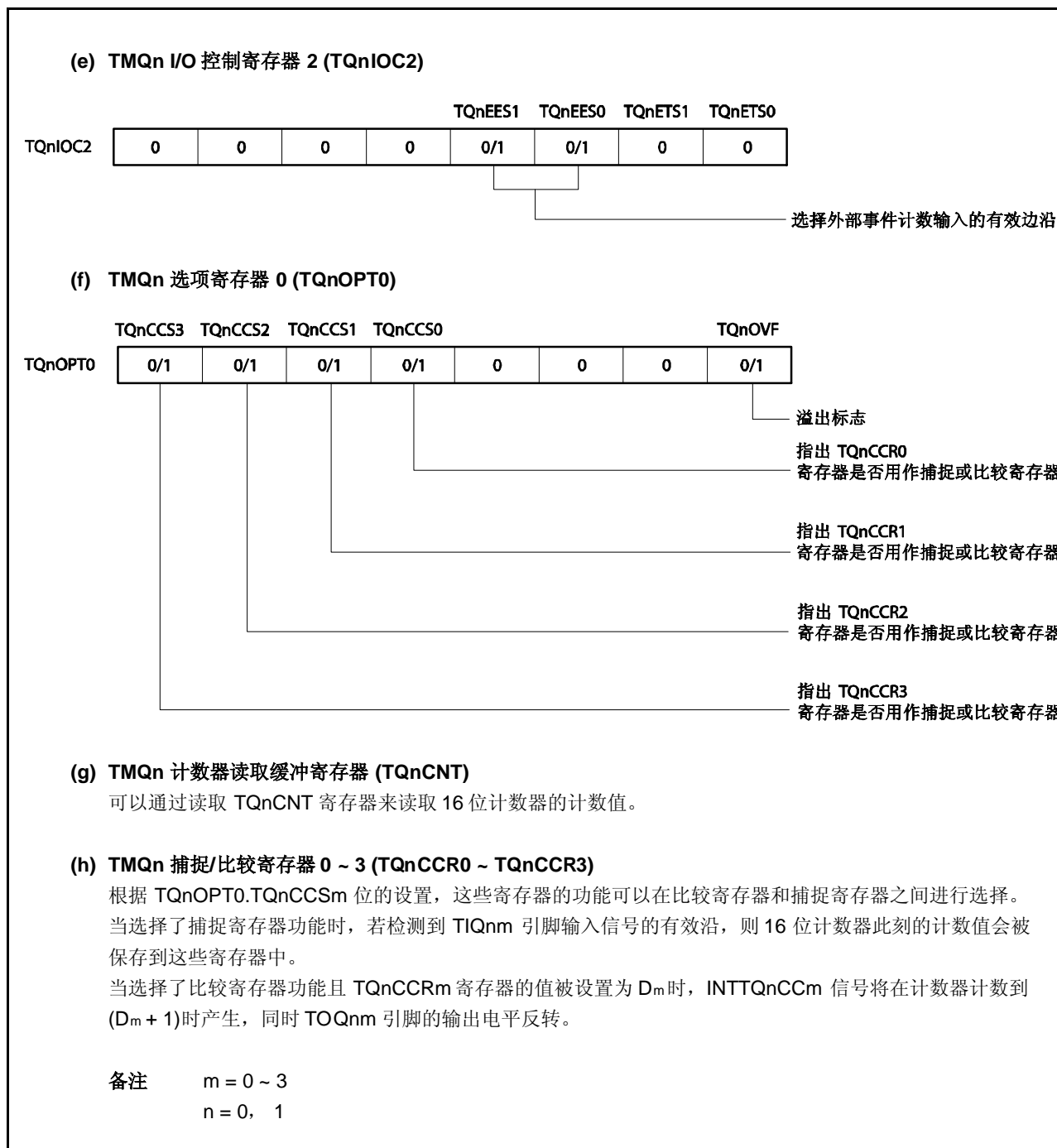


图 8-31. 自由运行定时器模式的寄存器设置 (3/3)



(1) 自由运行定时器模式的操作系统

(a) 当捕捉/比较寄存器用作比较寄存器时

图 7-32. 自由运行定时器模式的软件处理流程(比较功能) (1/2)

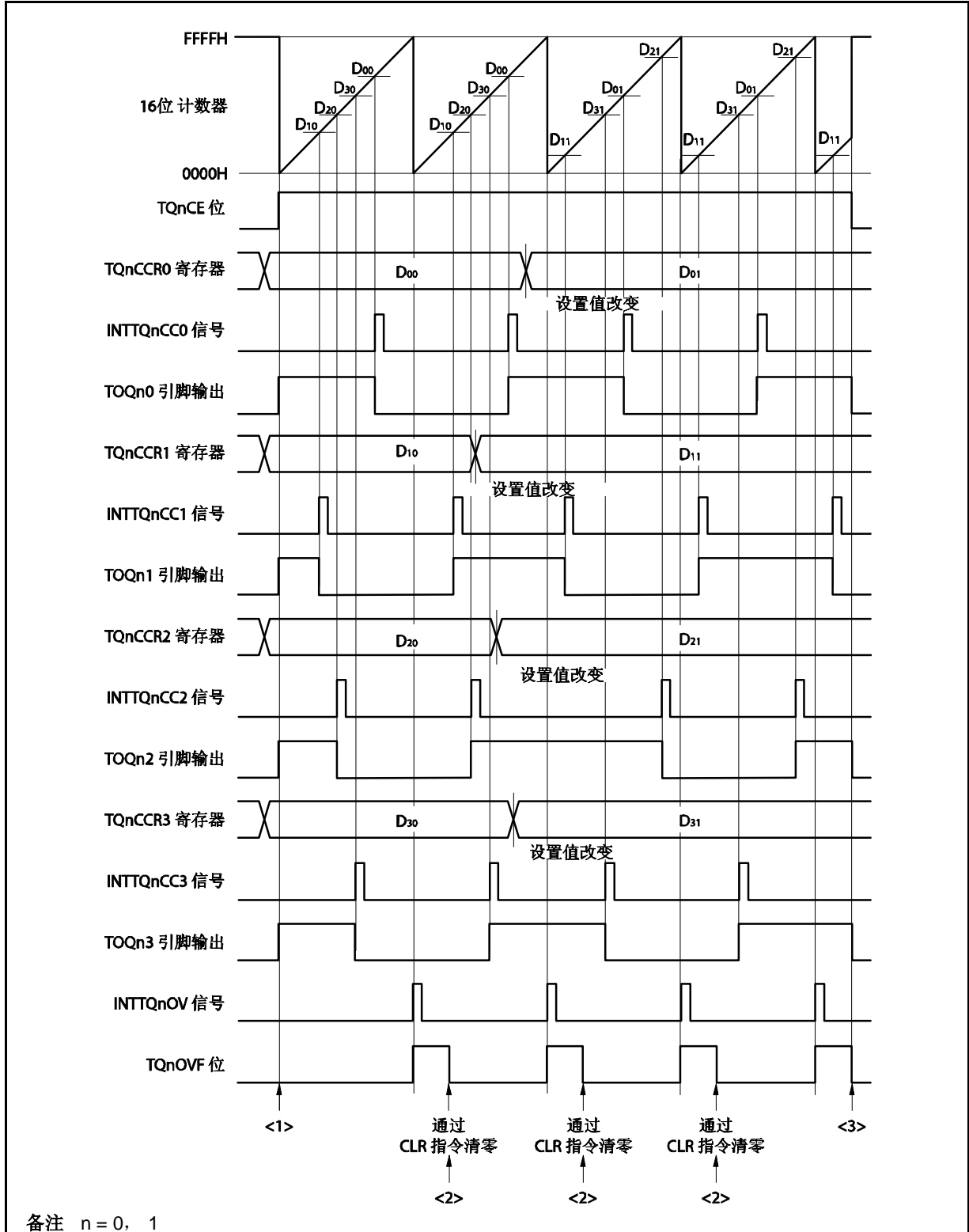
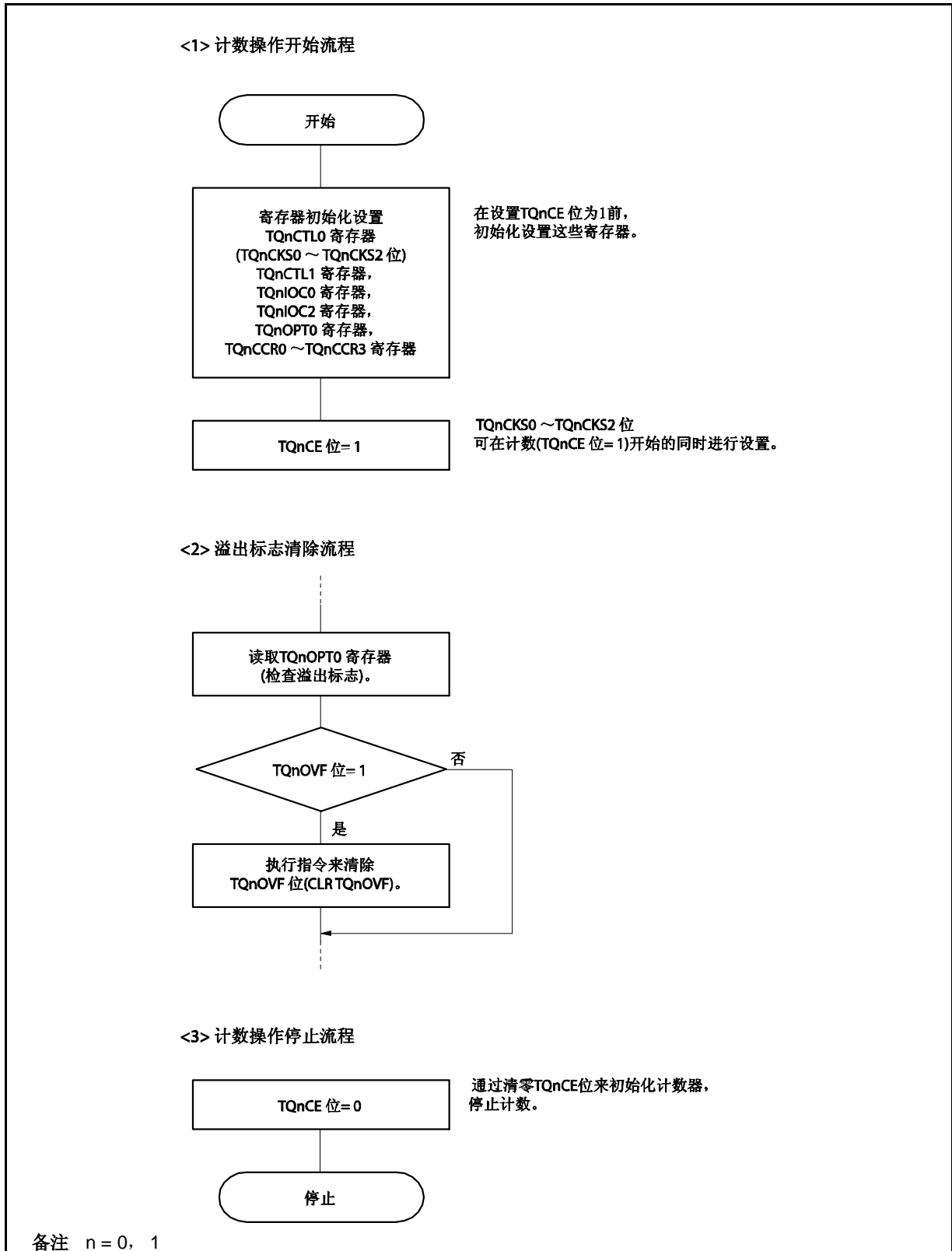


图 7-32. 自由运行定时器模式的软件处理流程(比较功能) (2/2)



(b) 当捕捉/比较寄存器用作捕捉寄存器时

图 7-33. 自由运行定时器模式的软件处理流程（捕捉功能）(1/2)

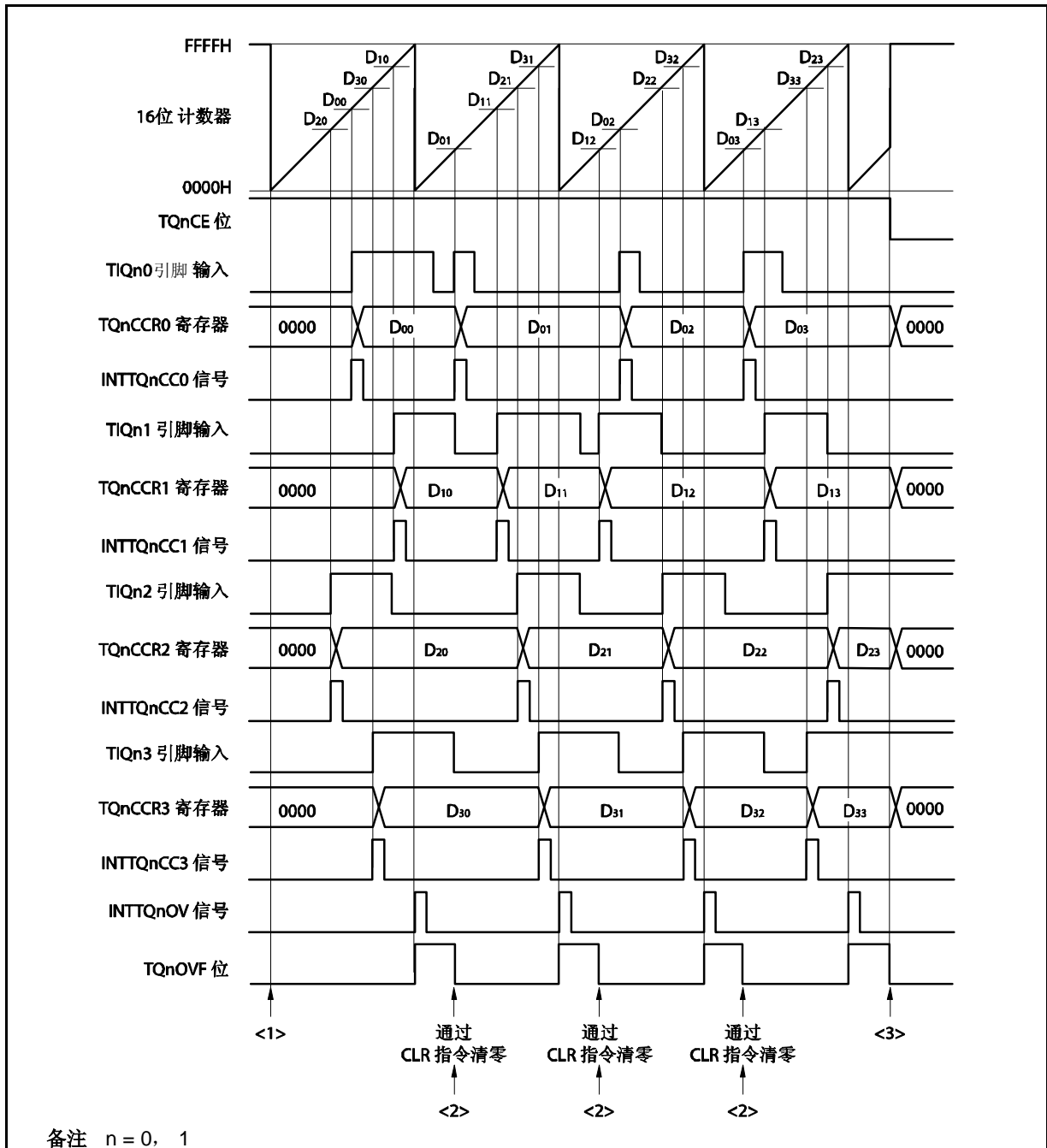
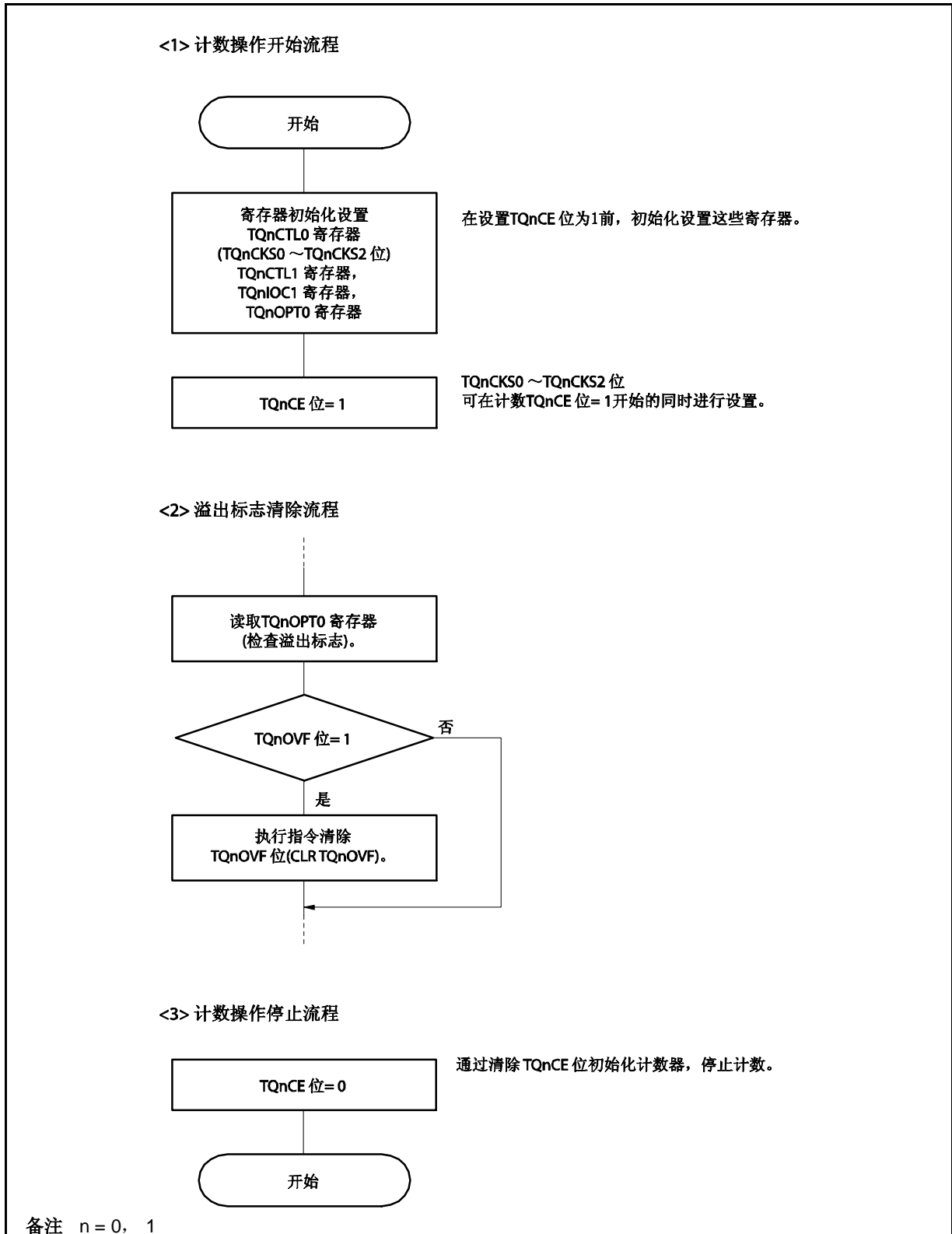


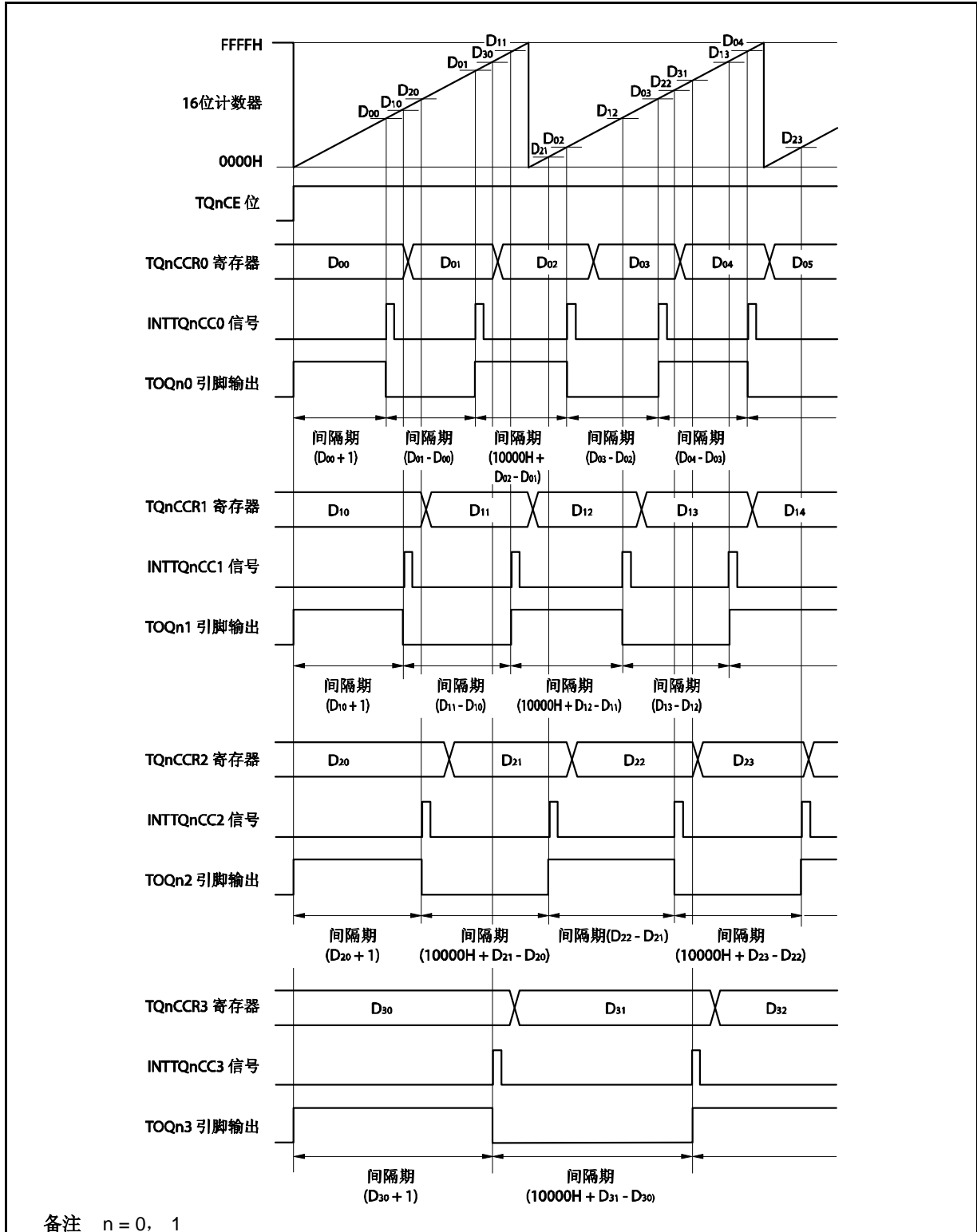
图 7-33. 自由运行定时器模式的软件处理流程（捕捉功能）(2/2)



(2) 自由运行定时器模式的操作时序

(a) 使用比较寄存器功能的时间间隔操作

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



当使用自由运行定时器模式进行间隔时间操作时，可以在同一通道指定四个不同的间隔时间。进行间隔时间操作时，必须在每次检测到 INTTQnCCm 信号后执行的中断服务程序中将相应的值设置到 TQnCCRm 寄存器。

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

比较寄存器的最初设置值： $D_m - 1$

第二次设置比较寄存器以后（含第二次）的设置值：前次设置值 + D_m

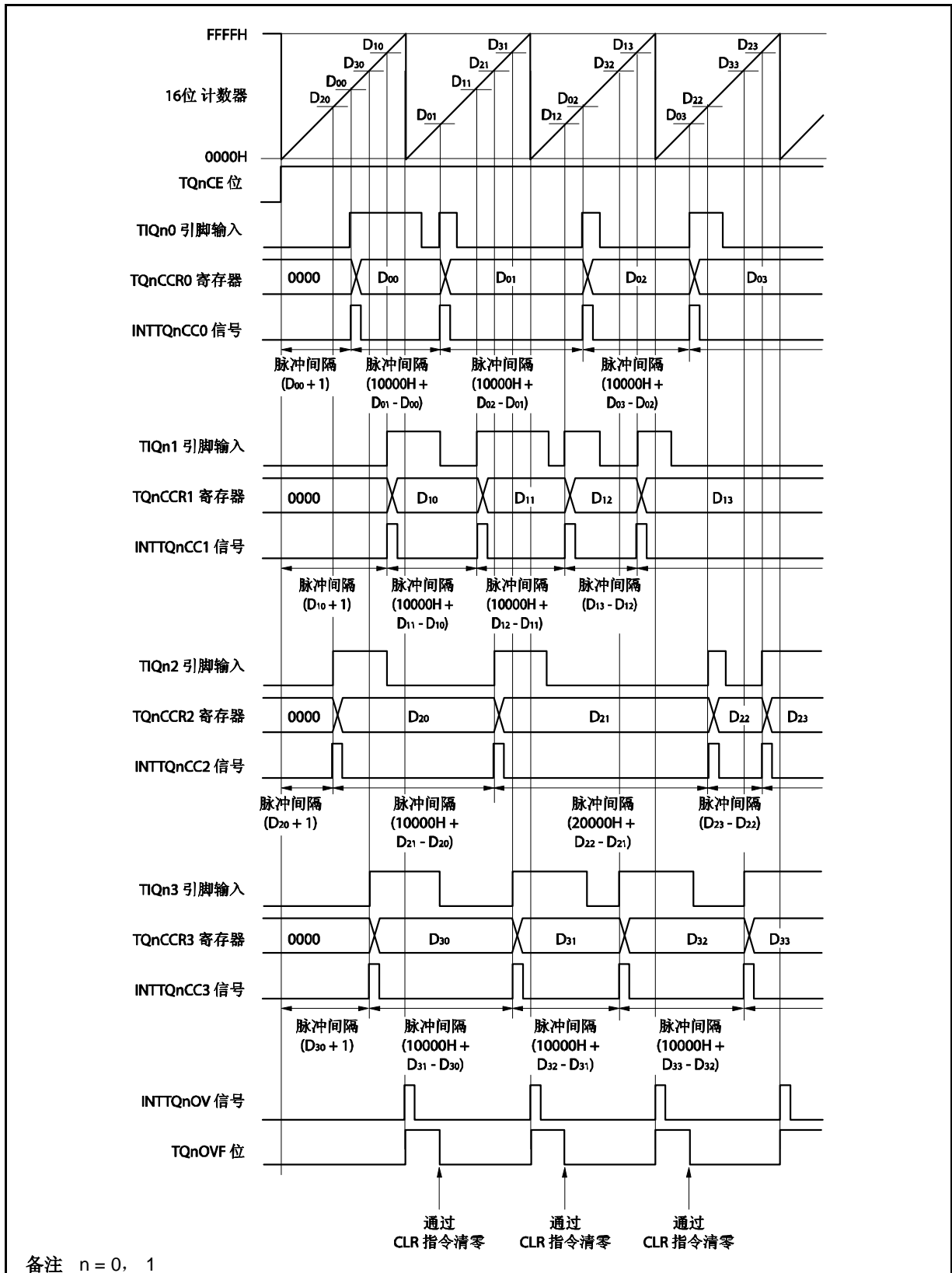
（如果计算结果大于 FFFFH，那么将结果减 10000H 后再设置到寄存器中。）

备注 $m = 0 \sim 3$

$n = 0, 1$

(b) 使用捕捉寄存器进行脉宽测量

当 TQnCCRm 寄存器用作捕捉寄存器进行脉宽测量时，需要通过软件处理在每次检测到中断信号时对捕捉寄存器进行读取并计算间隔时间。

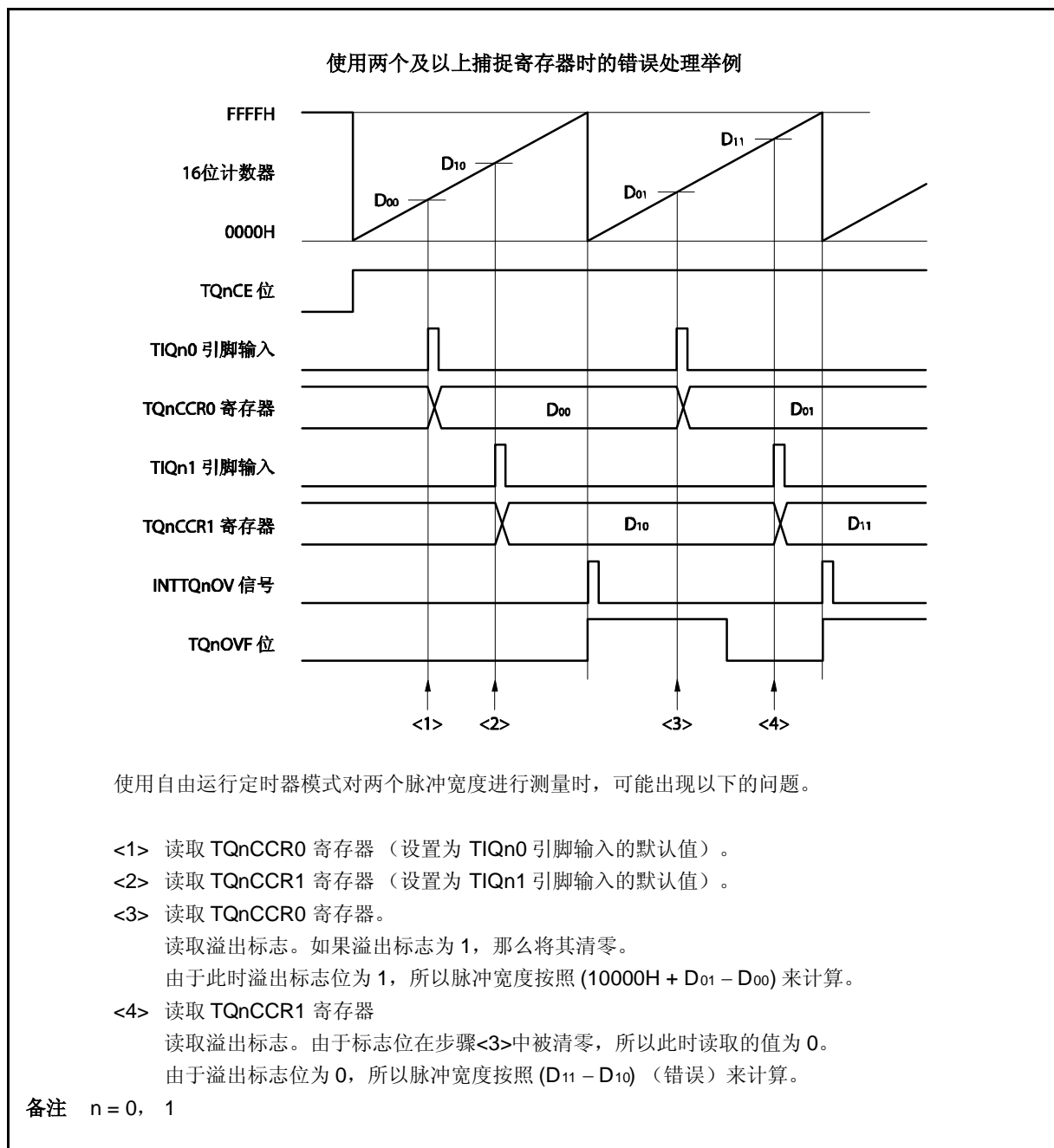


当使用自由运行定时器模式进行脉宽测量操作时，可以在同一通道对四个不同的脉冲宽度进行测量。通过与 INTTQnCCm 信号同步读取 TQnCCRm 寄存器的值并计算该值与上一次读取值的差，就可以求得脉冲的宽度。

备注 m = 0 ~ 3
 n = 0, 1

(c) 使用两个及以上捕捉寄存器产生溢出时的操作

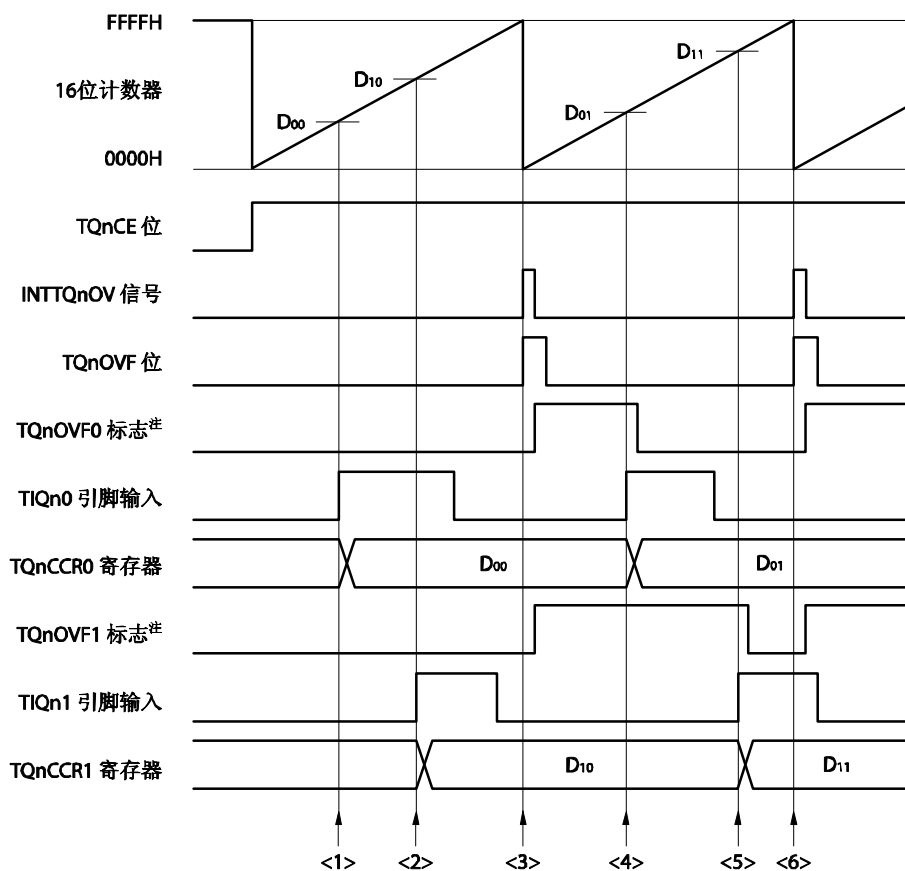
使用两个捕捉寄存器时，请务必慎重处理溢出标志位。首先，给出一个错误处理的例子，如下图所示。



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

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

使用两个捕捉寄存器时的处理举例（利用溢出中断）

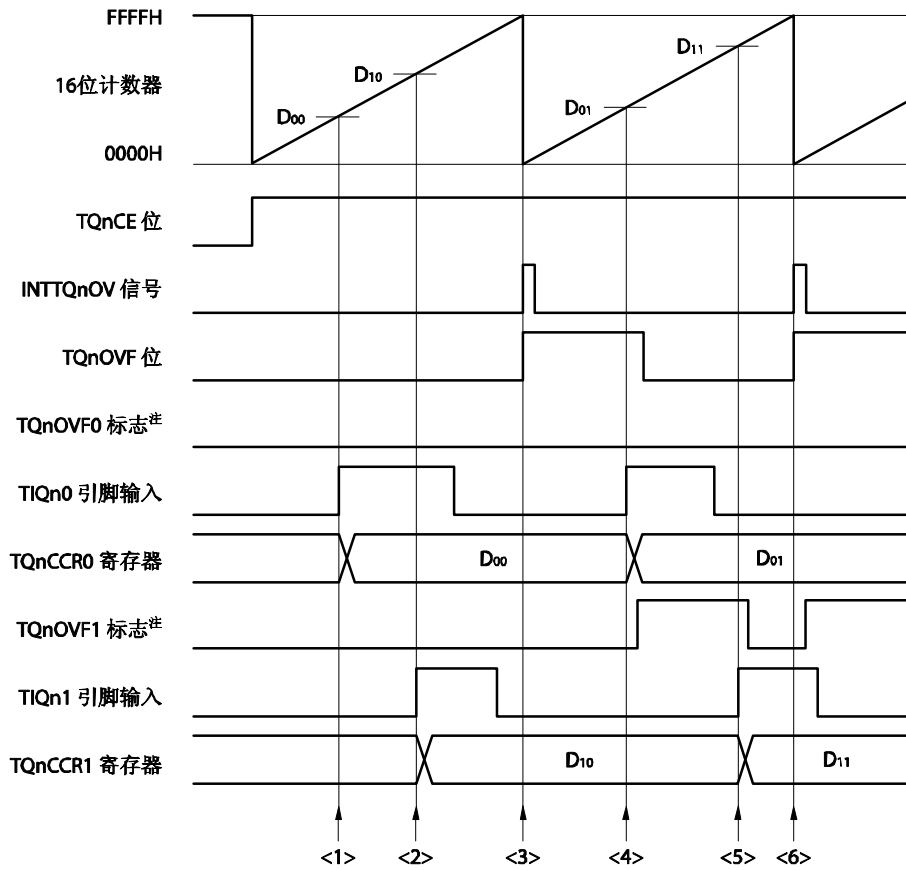


注 TQnOVF0 和 TQnOVF1 标志通过软件设置于内部 RAM。

- <1> 读取 TQnCCR0 寄存器（设置为 TIQn0 引脚输入的默认值）。
- <2> 读取 TQnCCR1 寄存器（设置为 TIQn1 引脚输入的默认值）。
- <3> 产生了溢出。在溢出中断服务程序中将 TQnOVF0 和 TQnOVF1 标志位置 1，同时将硬件溢出标志位清零。
- <4> 读取 TQnCCR0 寄存器。
 读取 TQnOVF0 标志位。如果该标志位为 1，那么将其清零。
 由于 TQnOVF0 标志位为 1，脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。
- <5> 读取 TQnCCR1 寄存器。
 读取 TQnOVF1 标志位，如果该标志位为 1，那么将其清零。（步骤<4>中将 TQnOVF0 位清零但 TQnOVF1 标志始终保持为 1。）
 由于 TQnOVF1 标志为 1，脉冲宽度按照 $(10000H + D_{11} - D_{10})$ （正确）来计算。
- <6> 同步骤 <3>

备注 n = 0, 1

使用两个捕捉寄存器时的处理举例（不利用溢出中断）



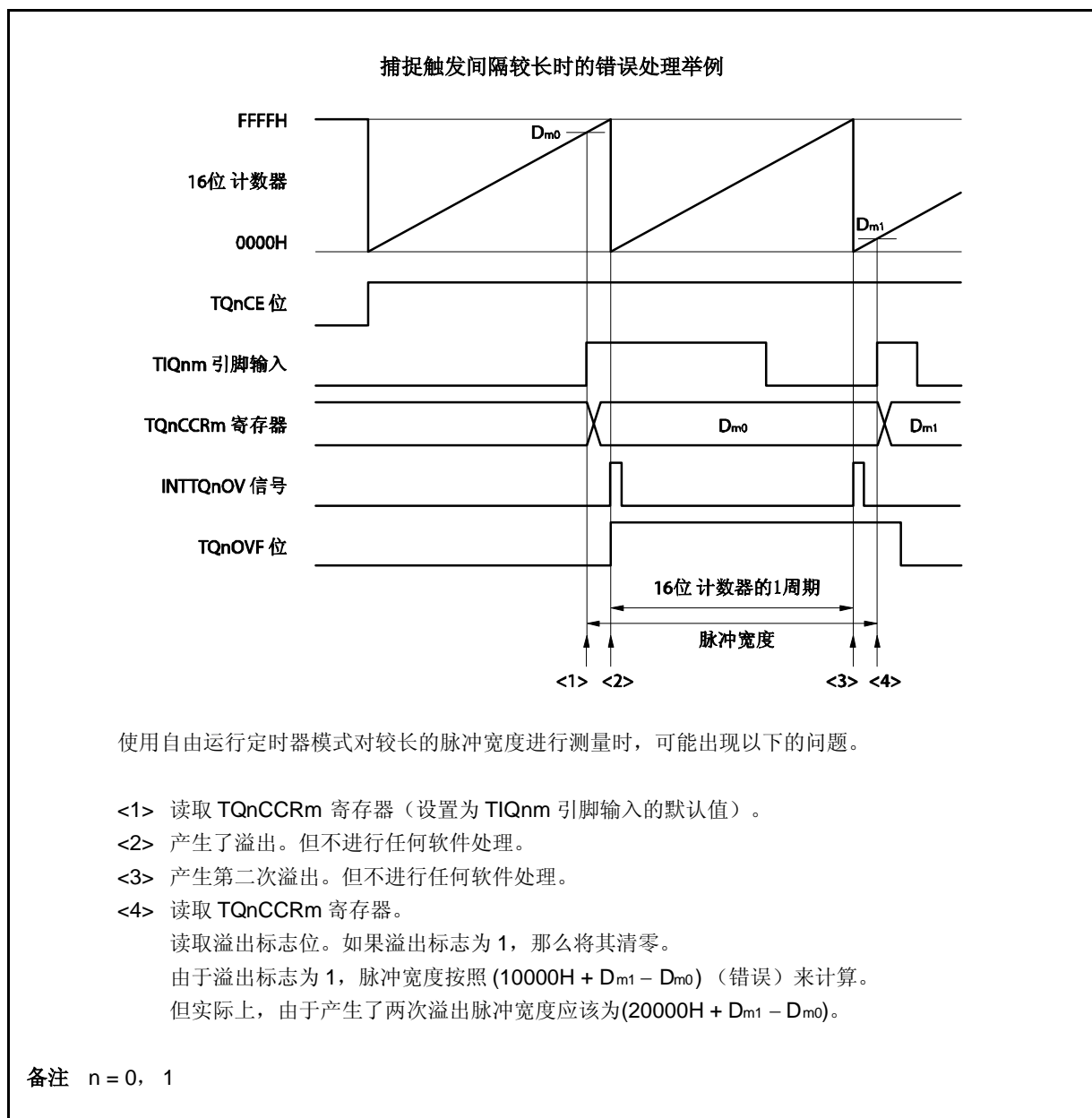
注 TQnOVF0 和 TQnOVF1 标志通过软件设置于内部 RAM。

- <1> 读取 TQnCCR0 寄存器（设置为 TIQn0 引脚输入的默认值）。
- <2> 读取 TQnCCR1 寄存器（设置为 TIQn1 引脚输入的默认值）。
- <3> 产生了溢出。但不进行任何软件处理。
- <4> 读取 TQnCCR0 寄存器。
读取溢出标志。如果溢出标志为 1，那么只把 TQnOVF1 标志设置为 1，同时将溢出标志清零。
由于 TQnOVF0 标志位为 1，脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。
- <5> 读取 TQnCCR1 寄存器。
读取溢出标志。由于溢出标志位在步骤<4>中被清零，所以读取结果为 0。
读取 TQnOVF1 位。如果 TQnOVF1 标志为 1，那么将其清零。
由于 TQnOVF1 标志为 1，脉冲宽度按照 $(10000H + D_{11} - D_{10})$ （正确）来计算。
- <6> 同步步骤 <3>

备注 n = 0, 1

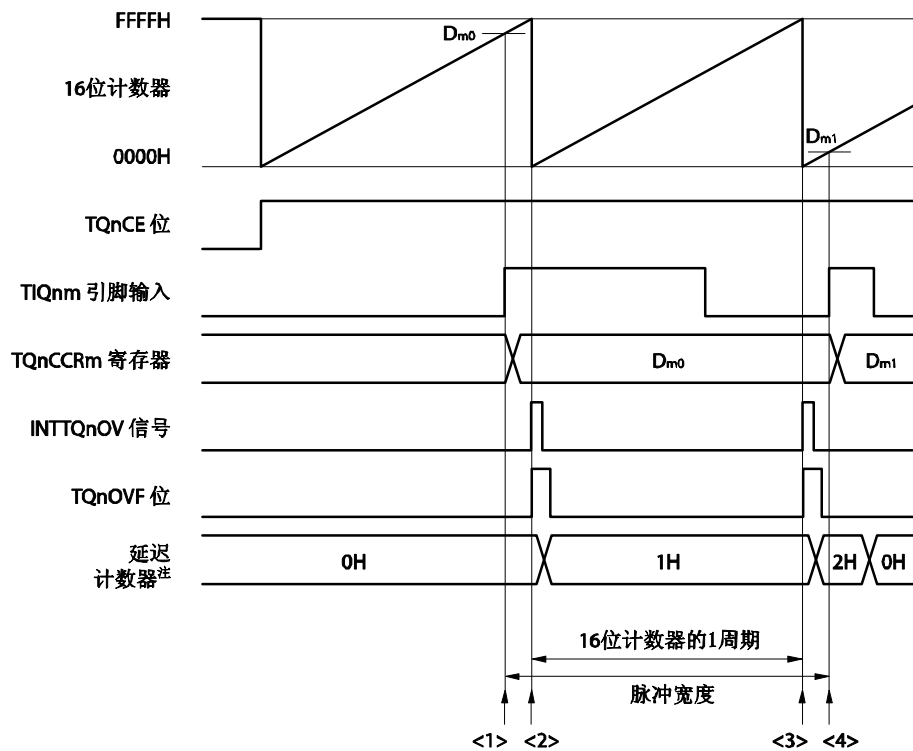
(d) 捕捉触发间隔较长时的溢出处理

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



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

捕捉触发间隔较长时的处理举例



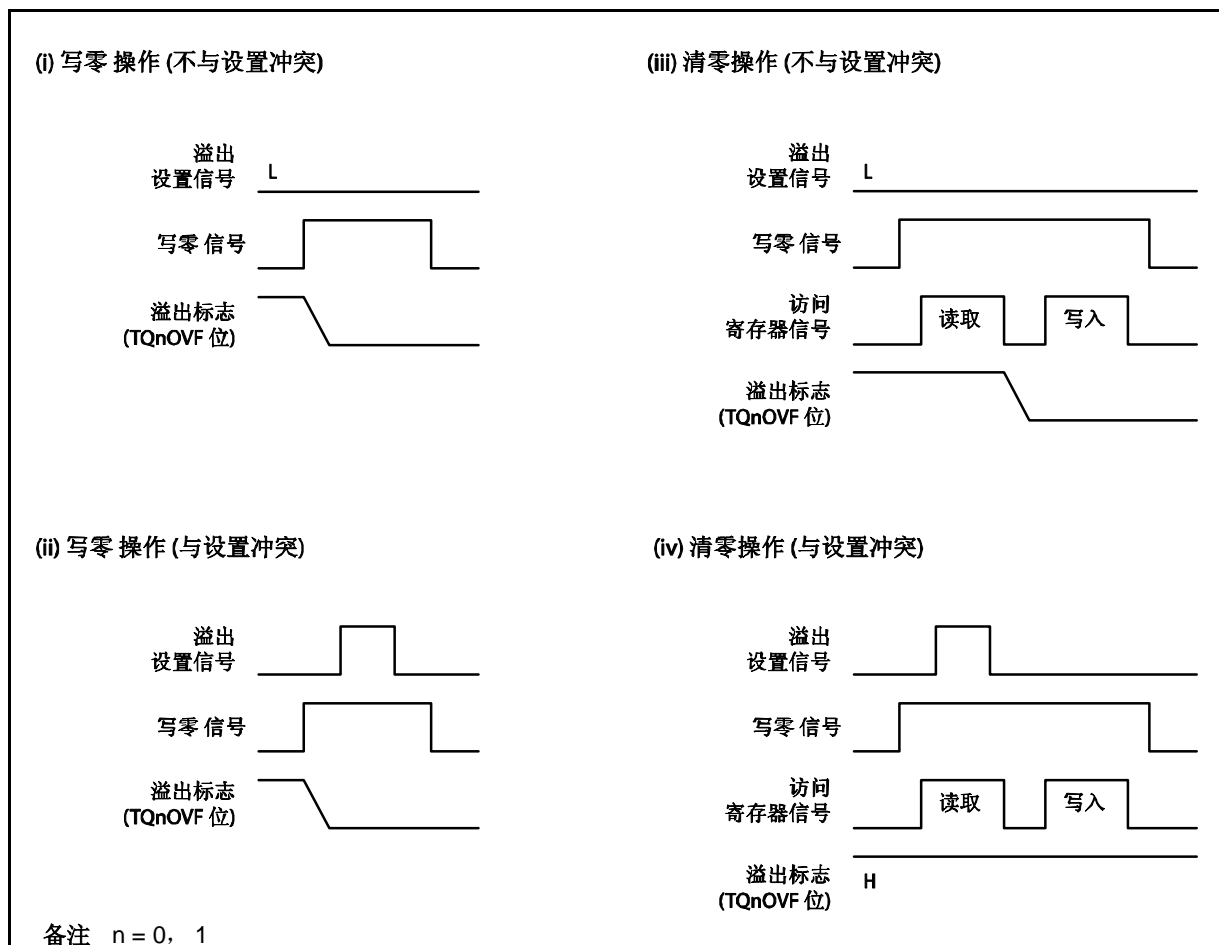
注 这里的溢出次数计数器是建立于内部 RAM 上的变量，可由软件任意进行设置。

- <1> 读取 TQnCCRm 寄存器（设置为 TIQ0m 引脚输入的默认值）。
- <2> 产生了溢出。在溢出中断服务程序中将溢出次数计数器加 1 并将溢出标志清零。
- <3> 产生了第二次溢出。在溢出中断服务程序中将溢出次数计数器加 1 并将溢出标志清零。
- <4> 读取 TQnCCRm 寄存器。
读取溢出次数计数器的值。
→ 假设此时溢出计数器的值为“N”，则脉冲宽度可按 $(N \times 10000H + D_{m1} - D_{m0})$ 计算求得。
在本例中，由于溢出产生了两次，所以脉冲宽度为 $(20000H + D_{m1} - D_{m0})$ 。
将溢出次数计数器清零(0H)。

备注 n = 0, 1

(e) 将溢出标志位清零

可以通过使用 CLR 指令清零 TQnOVF 位或向 TQnOPT0 寄存器写入 8 位数据（第 0 位为 0）的方法将溢出标志位清零。为了能够准确检测到溢出的发生，请务必在每次读取到 TQnOVF 位为 1 后，立即使用位操作指令将该位清零。



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

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

7.5.7 脉宽测量模式 (TQnMD2 ~ TQnMD0 位 = 110)

在脉宽测量模式中, TQnCTL0.TQnCE 位被置 1 后, 16 位计数器 Q 开始计数。每当检测到 TIQnm 引脚输入信号的有效沿时, 16 位计数器的计数值就会被保存到 TQnCCRm 寄存器中, 同时计数器清零为 0000H。

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

可以选择 TIQn0 ~ TIQn3 引脚中的任意一个作为捕捉触发输入引脚。通过 TQnIOC1 寄存器将未使用的引脚设置为“无边沿检测”。

当使用外部时钟来测量脉宽时, 只能使用 TIQ0k 引脚作为捕捉触发输入引脚, 因为外部时钟输入已经固定为 TIQn0 引脚。此时要将 TQnIOC1.TQnIS1 和 TQnIOC1.TQnIS0 位清为 00 (捕捉触发输入 (TIQn0 引脚): 无边沿检测)。

备注 m = 0 ~ 3
n = 0, 1
k = 1~3

图 7-34. 脉宽测量模式的配置图

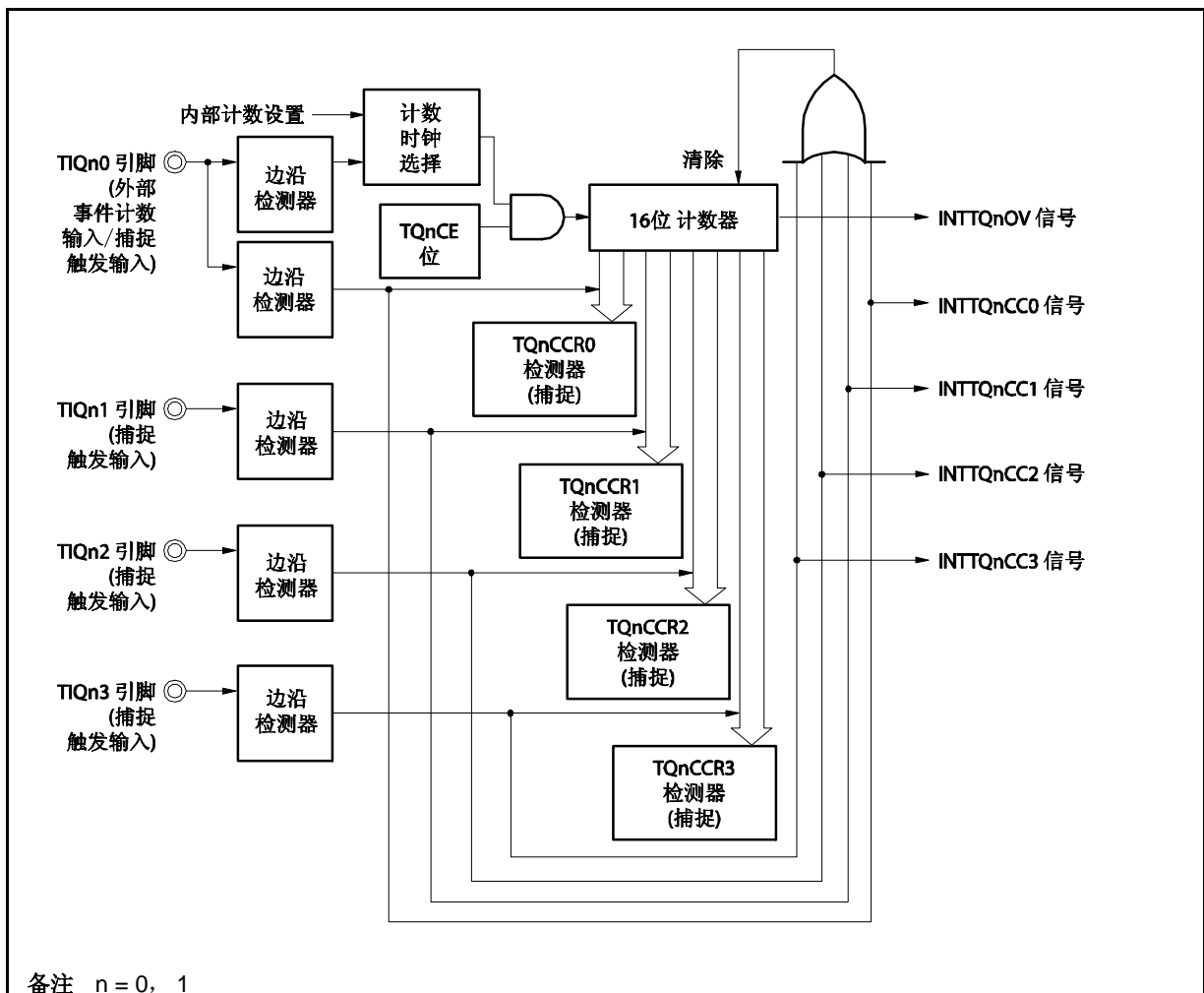
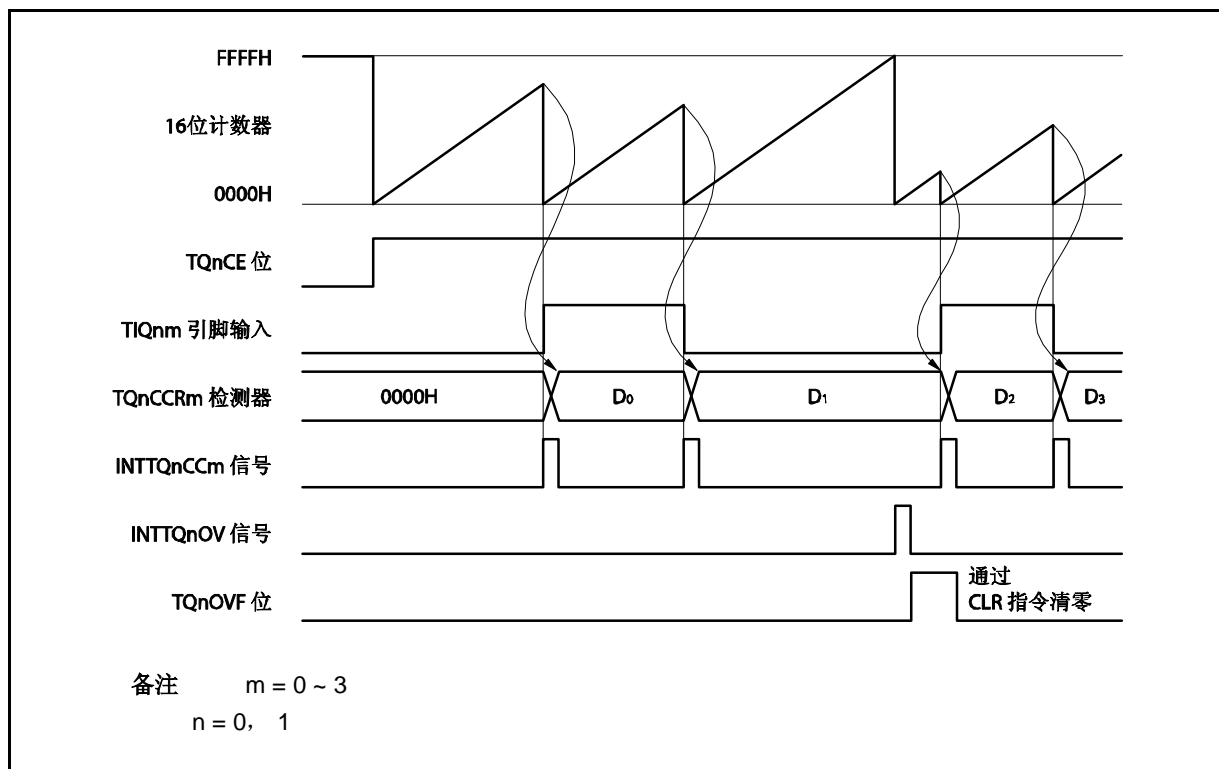


图 7-35. 脉宽测量模式的基本时序



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

脉冲宽度可由下式求得。

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

如果直到 16 位计数器计数到 FFFFH 仍没有检测到 TIQnm 引脚的有效沿，那么将在下一个时钟时产生一个溢出中断请求信号 (INTTQnOV)，同时计数器被清零并重新开始计数。这种情况下，溢出标志位 (TQnOPT0.TQnOVF 位) 也将被设置为 1。请通过软件使用 CLR 指令将溢出标志清零。

如果溢出标志被设置为 1，那么脉冲宽度可由下式求得。

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

备注 $m = 0 \sim 3$
 $n = 0, 1$

图 7-36. 脉宽测量模式的寄存器设置 (1/2)

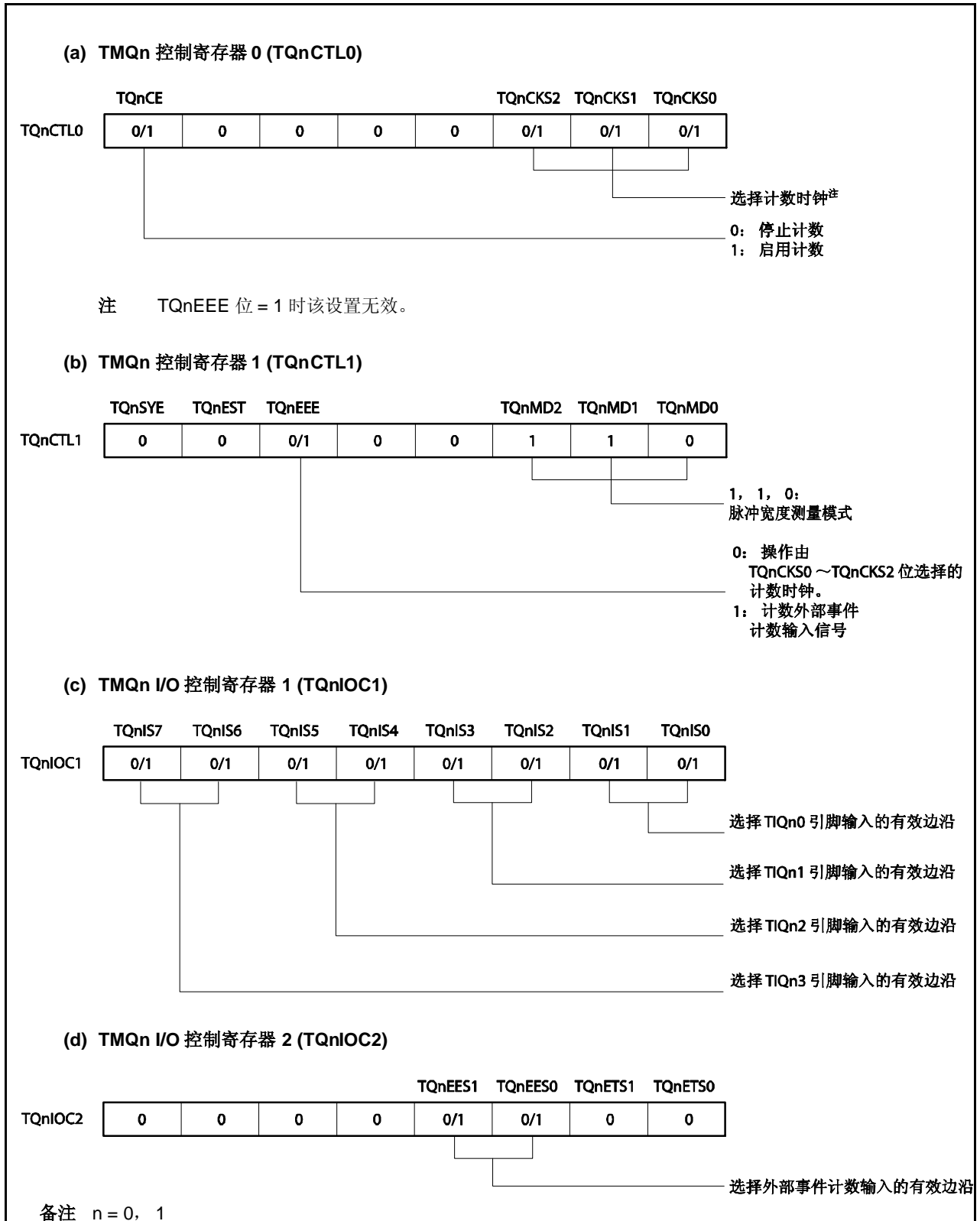
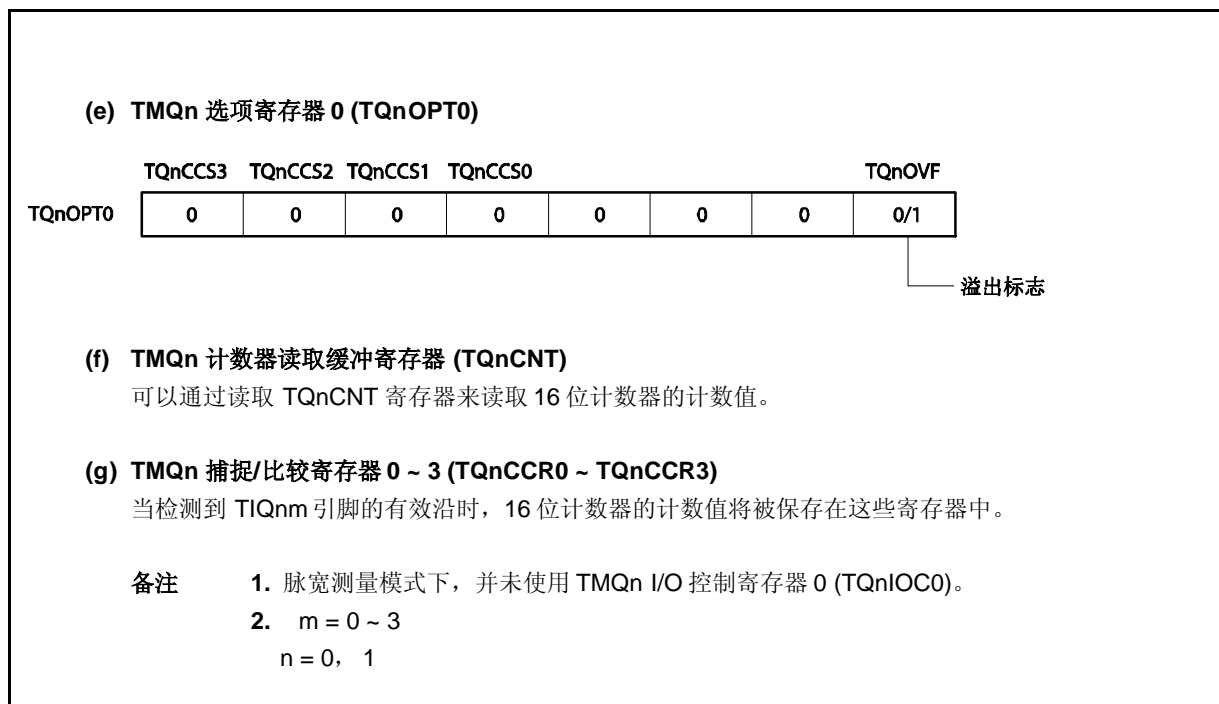
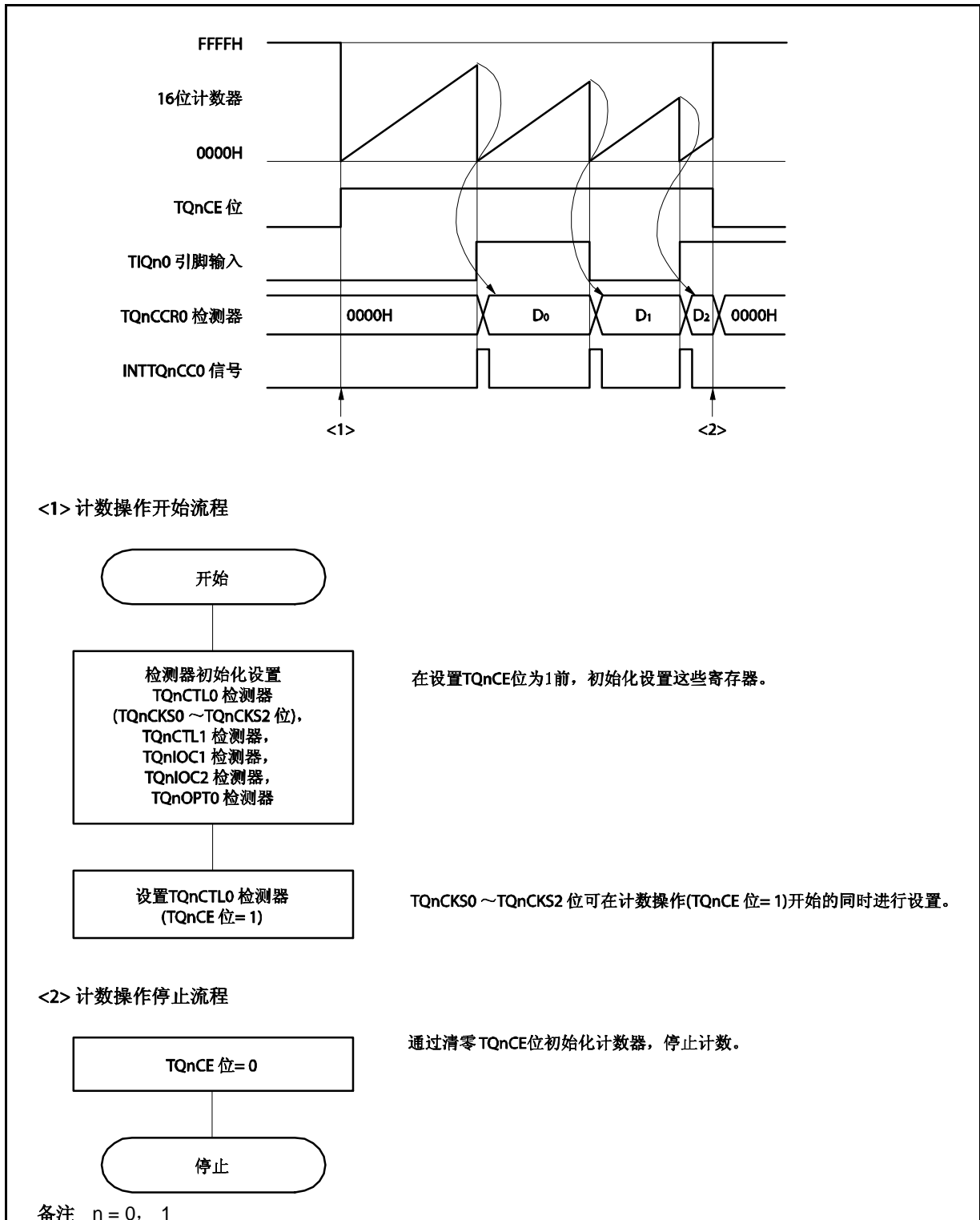


图 7-36. 脉宽测量模式的寄存器设置 (2/2)



(1) 脉宽测量模式的流程图

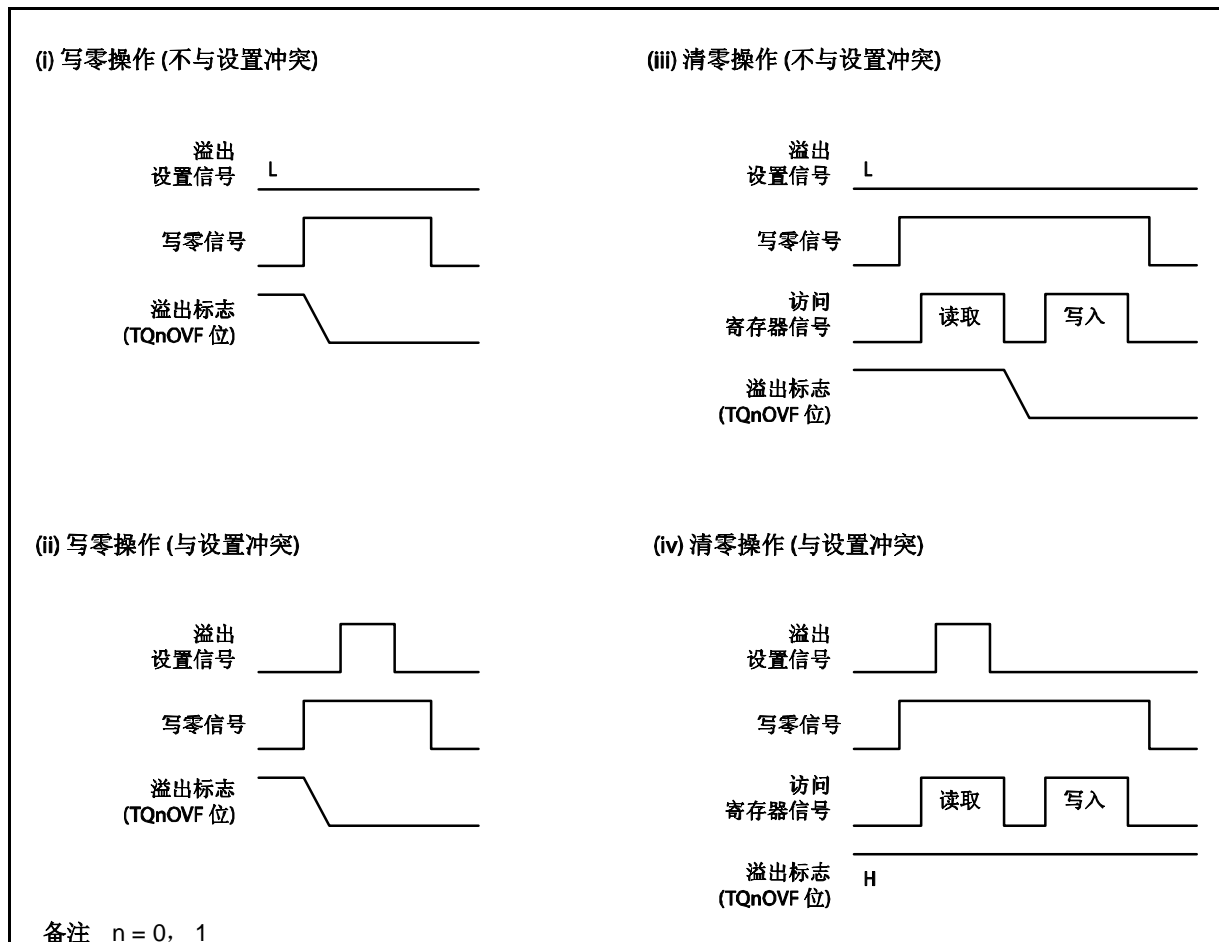
图 7-37. 脉宽测量模式的软件处理流程



(2) 脉宽测量模式的操作时序

(a) 将溢出标志位清零

可以通过使用 CLR 指令清零 TQnOVF 位或向 TQnOPT0 寄存器写入 8 位数据（第 0 位为 0）的方法将溢出标志位清零。为了能够准确检测到溢出的发生，请务必在每次读取到 TQnOVF 位为 1 后，立即使用位操作指令将该位清零。



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

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

7.5.8 三角波 PWM 模式 (TQnMD2 ~ TQnMD0 = 111)

在三角波 PWM 模式下，TMQn 捕捉/比较寄存器 k (TQnCCRk)被用于设置占空比系数，并且 TMQn 捕捉/比较寄存器 0(TQnCCR0)被用于设置周期。

通过使用这四个寄存器和操作定时器，带有不同周期的三角波 PWM 被输出。

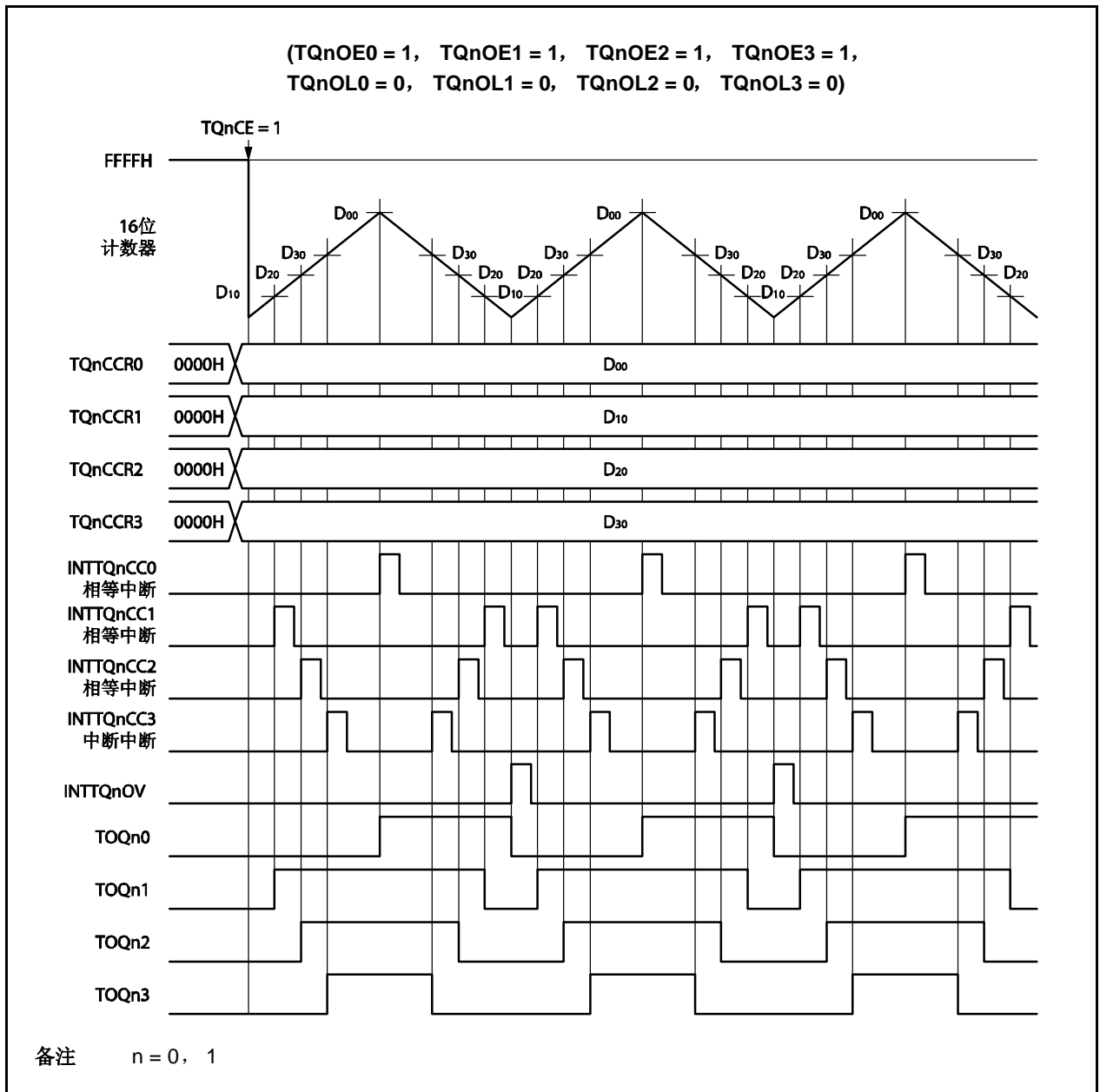
TQnCE = 1 时，TQnCCRm 寄存器的值可以被改写。

如果要停止定时器 Q，则将 TQnCE 清零。PWM 微波从 TOQnk 引脚被输出。当 16 位计数器的值与 TQnCCR0 寄存器的值相等时且当该计数器下溢时，TOQn0 引脚便会产生反转输出。

注意事项 在 PWM 模式中，TQnCCRm 寄存器的捕捉功能未被使用，因为该寄存器可仅用于比较寄存器。

备注 n = 0, 1, m = 0 ~ 3, k = 1 ~ 3

图 7-38. 三角波 PWM 模式中的基本操作时序



7.5.9 定时器输出操作

TOQn0 ~TOQn3 引脚的工作模式与输出电平的关系如下表所示。

表 7-6. 各模式下的定时器输入控制

操作模式	TOQn0 引脚	TOQn1 引脚	TOQn2 引脚	TOQn3 引脚
间隔定时器模式	方波输出			
外部事件计数模式	方波输出	-		
外部触发脉冲输出模式	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
单脉冲输出模式		单脉冲输出	单脉冲输出	单脉冲输出
PWM 输出模式		PWM 输出	PWM 输出	PWM 输出
自由运行定时器模式	方波输出（仅限使用比较功能时）			
脉宽测量模式	-			
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. TOQn0 ~ TOQn3 引脚在定时器输出控制位控制下的真值表

TQnIOC0.TQnOLm 位	TQnIOC0.TQnOEm 位	TQnCTL0.TQnCE 位	TOQnm 引脚的电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前为低电平，计数过程中为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平，计数过程中为低电平

备注 m = 0 ~ 3
n = 0, 1

7.6 定时器调谐操作功能

定时器 P 和定时器 Q 具有调谐操作功能。

在表 7-8 中所列可被同步操作的定时器。

表 7-8. 定时器的调谐操作模式

主定时期	从定时器	
TMP0	TMP1	—
TMP2	TMP3	TMQ0

- 注意事项**
- 通过 **TPmCTL1.TPmSYE** 和 **TQ0CTL1.TQ0SYE** 位可以启用或禁止调谐操作模式，对于 **TMQ2**，**TMQ3** 和 **TMQ0** 之一或者两者都可以指定为从定时器。
 - 使用以下程序设置调谐操作模式。
 - 设置从定时器的 **TPmCTL1.TPmSYE** 和 **TQ0CTL1.TQ0SYE** 位以启动调谐操作。
设置从定时器的 **TPmCTL1.TPmMD2** 为 **TPmCTL1.TPmMD0** 以及 **TQ0CTL1.TQ0MD2** 为 **TQ0CTL1.TQ0MD0** 位以进入自由运行模式。
 - 通过使 **TPnCTL1.TPnMD2** 位为 **TPnCTL1.TPnMD0** 位可以设置定时器模式。
此时，不可以设置主定时器的 **TPnCTL1.TPnSYE** 位。
 - 设置主和从定时器的比较寄存器值。
 - 设置从定时器的 **TPmCTL0.TPmCE** 和 **TQ0CTL0.TQ0CE** 以启动对内部运行时钟的操作。
 - 设置主定时器的 **TPnCTL0.TPnCE** 位以启动对内部运行时钟的操作。

备注 m = 1, 3
 n = 0, 2

表 7-9 和 7-10 显示可被用于调谐操作模式的定时器模式 (√: 可设置, ×: 不可设置)。

表 7-9. 可被用于调谐操作模式的定时器模式

主定时器	自由运行模式	PWM 模式	三角波 PWM 模式
TMP0	√	√	×
TMP2	√	√	×

表 7-10. 定时器输出功能

调谐通道	定时器	引脚	自由运行模式		PWM 模式		三角波 PWM 模式	
			关闭	开启	关闭	开启	关闭	开启
通道 0	TMP0 (主)	TOP00	PPG		反转		N/A	
		TOP01	PPG		PWM		N/A	
	TMP1 (从)	TOP10	PGP		反转	PWM	N/A	
		TOP11	PPG		PWM		N/A	
通道 1	TMP2 (主)	TOP20	PPG		反转		N/A	
		TOP21	PPG		PWM		N/A	
	TMP3 (从)	TOP30	PPG		反转	PWM	N/A	
		TOP31	PPG		PWM		N/A	
	TMQ0 (从)	TOQ00	PPG		反转	PWM	反转	N/A
		TOQ01 ~ TOQ03	PPG		PWM		三角波 PWM	N/A

备注 从主定时器的比较寄存器到从定时器的比较寄存器的传输数据定时如下。

PPG: CPU 写入时序
 反转, PWM, 三角波 PWM: 定时器计数器和比较寄存器与 TOPn0 和 TOQ00 相等的时序 (n = 0, 1)

图 7-39. 调谐操作图像(TMP2, TMP3, TMQ0)

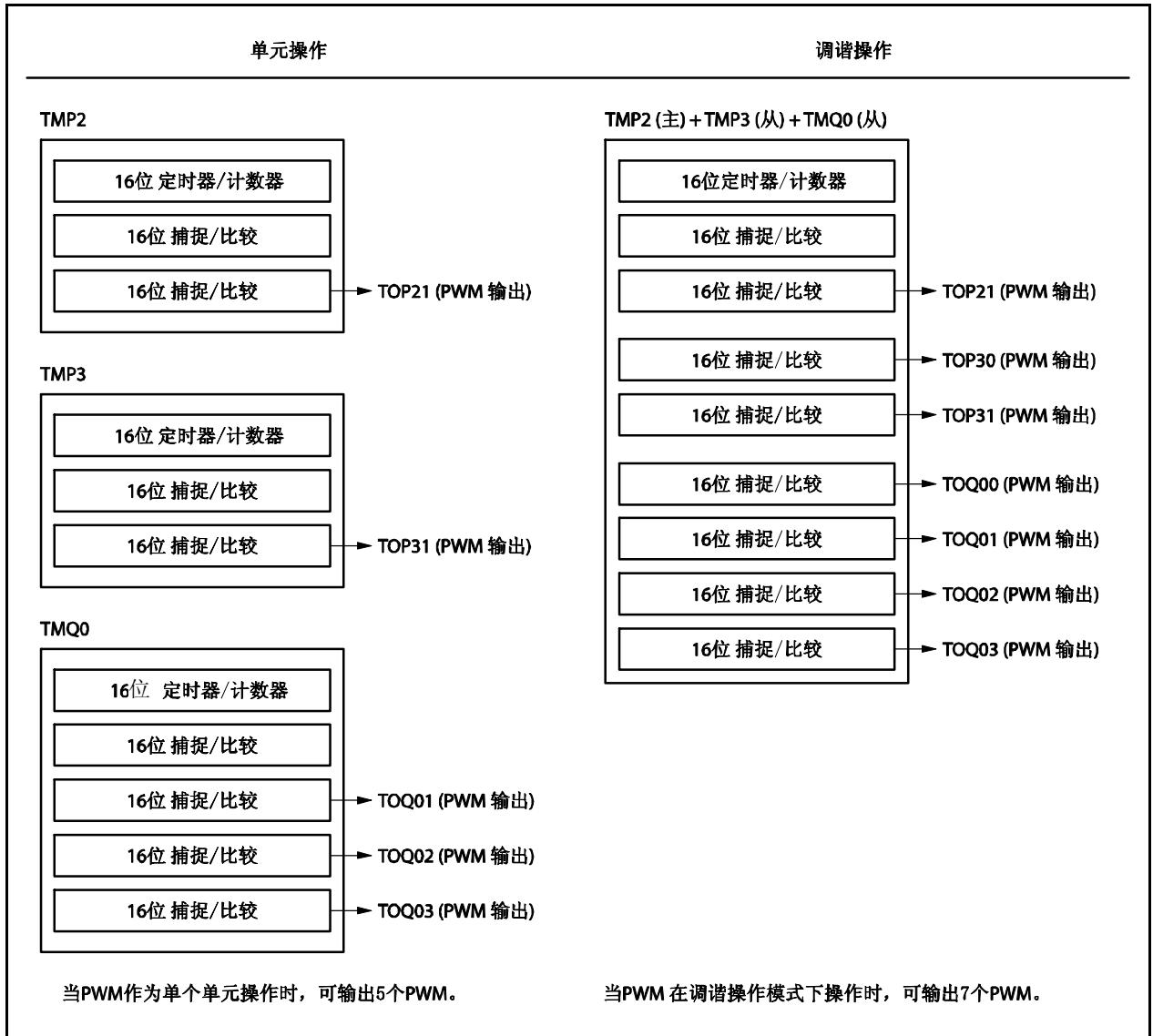
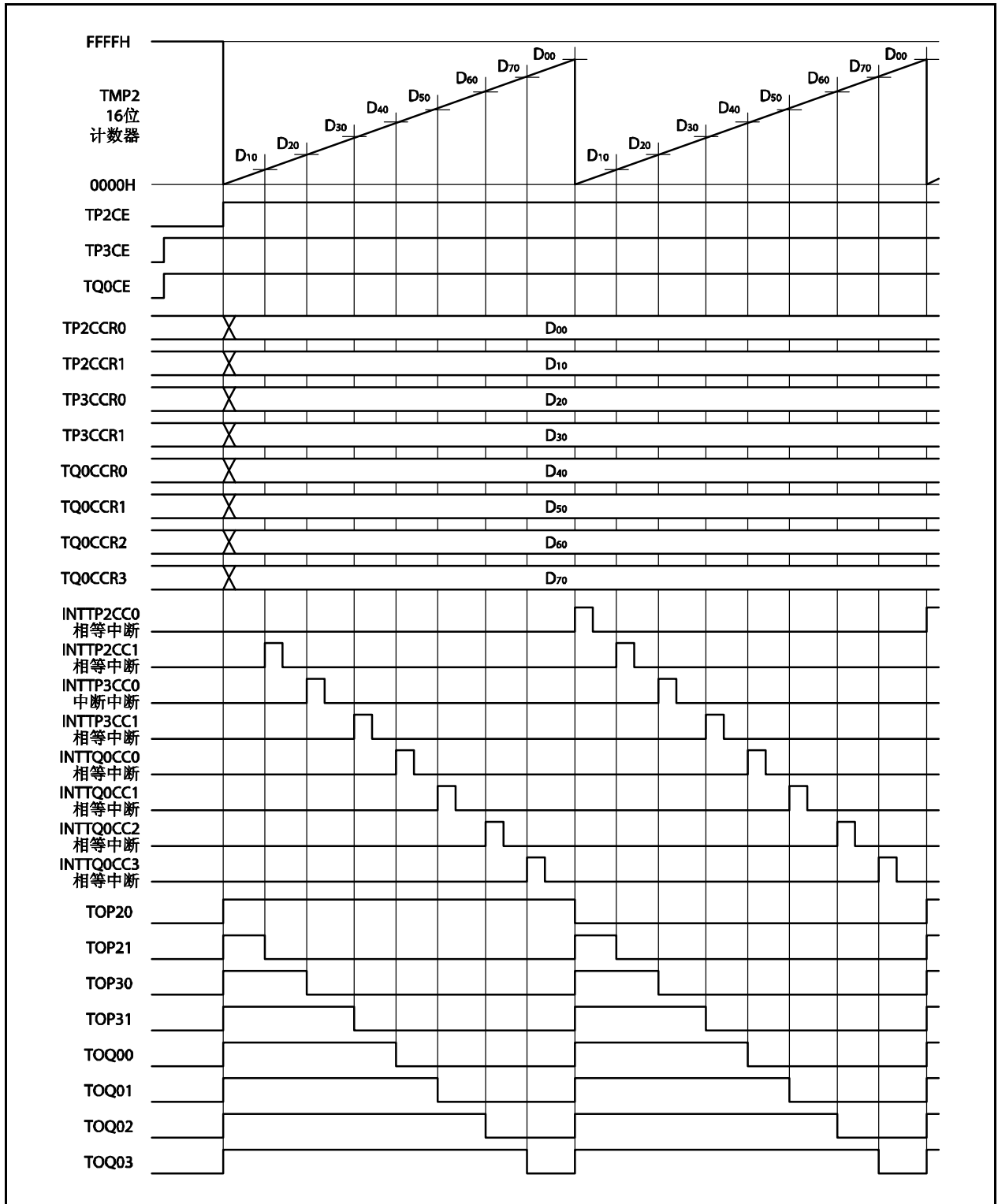


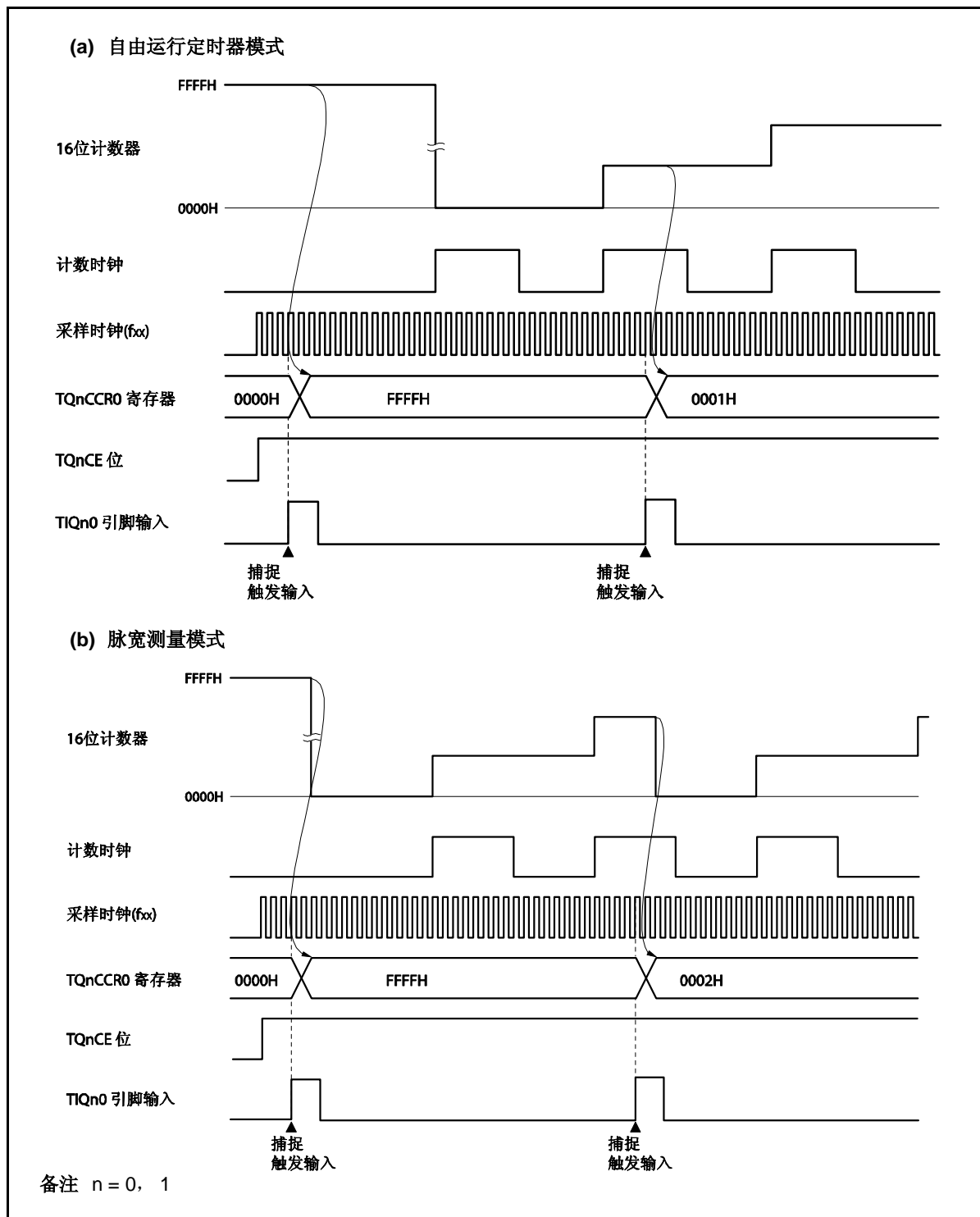
图 7-40. 调谐 PWM 功能的基本操作时序(TMP2, TMP3, TMQ0)



7.7 注意事项

(1) 捕捉操作

若在捕捉操作时使用了一个较慢的时钟信号作为计数时钟，当 TQnCE 位被设置为 1 后，立刻检测到捕捉触发信号时 TQnCCR0, TQnCCR1, TQnCCR2, 和 TQnCCR3 寄存器中捕捉的计数值将不会是 0000H 而是 FFFFH。



第八章 16 位间隔定时器 M (TMM)

8.1 概述

- 间隔功能
- 8 种时钟可选
- 16 位计数器 × 1
(定时器计数操作过程中不可对 16 位计数器进行读取。)
- 比较寄存器 × 1
(定时器计数操作过程中不可向比较寄存器写入数据。)
- 比较相等中断 × 1

定时器 M 仅支持清零&开始模式。不支持自由运行定时器模式。

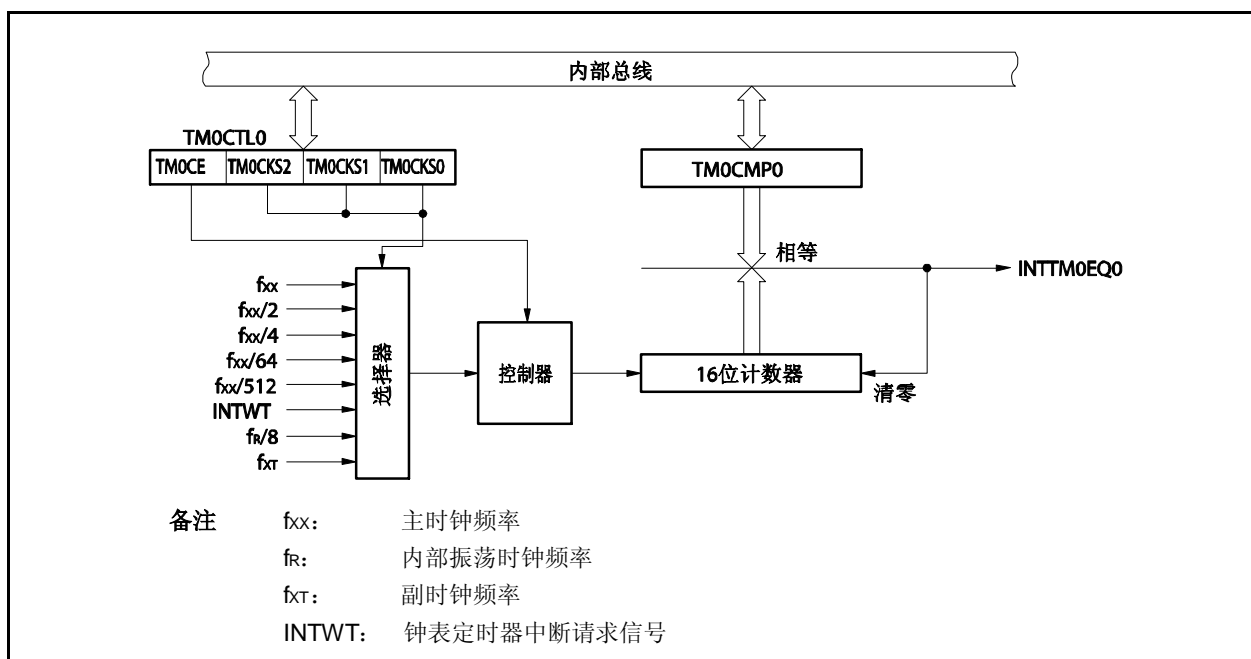
8.2 配置

TMM0 包括以下硬件。

表 8-1. TMM0 的配置

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 (TMOCMP0)
控制寄存器	TMM0 控制寄存器 0 (TMOCTL0)

图 8-1. TMM0 的框图

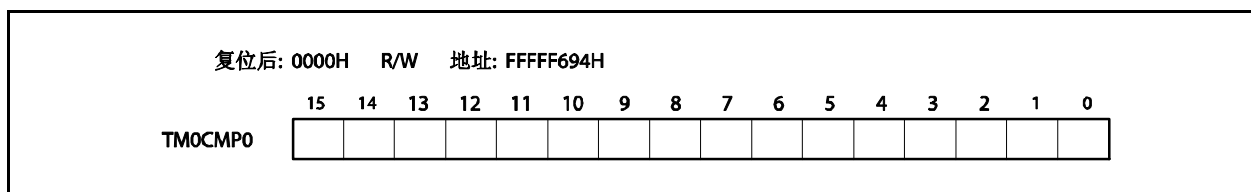


(1) 16 位计数器

该计数器是一个计数内部时钟计数的 16 位计数器。
该 16 位计数器不可以被读写。

(2) TMM0 比较寄存器 0 (TMOCMP0)

TMOCMP0 寄存器是一个 16 位比较寄存器。
该寄存器支持 16 位读写方式。
该寄存器复位后的值为 0000H。
可使用软件向 TMOCMP0 寄存器写入相同的数值。
当 TMOCTL0.TMOCE 位= 1 时，禁止改写 TMOCMP0 寄存器。



8.3 寄存器

(1) TMM0 控制寄存器(TMOCTL0)

TMOCTL0 寄存器是用于控制 TMM0 操作的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

可使用软件向 TMOCTL0 寄存器写入相同的数值。当操作定时器时，除了 TMOCE 位，禁止改写该寄存器。

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	内部时钟操作使能/禁止指定
0	禁止TMM0操作(16位计数器异步复位)。操作时钟停止应用。
1	使能TMM0操作。操作时钟开始应用。开始 TMM0 操作。

对TMM0进行与TMOCE位异步的内部时钟控制和内部电路复位。
当TMOCE位被清零时，TMM0的内部时钟被禁用（固定为低电平），同时16位计数器被异步复位。

TMOCKS2	TMOCKS1	TMOCKS0	计数时钟选择
0	0	0	f_{xx}
0	0	1	$f_{xx}/2$
0	1	0	$f_{xx}/4$
0	1	1	$f_{xx}/64$
1	0	0	$f_{xx}/512$
1	0	1	INTWT
1	1	0	$f_{fr}/8$
1	1	1	f_{xt}

- 注意事项**
1. 当 TMOCE 位 = 0 时，设置 TMOCKS2 ~ TMOCKS0 位。
当 TMOCE 的值从 0 改变位 1 时，不可以同时设置 TMOCKS2 ~ TMOCKS0 位的值。
 2. 请务必将第 3 位到第 6 位清零。

备注

f_{xx} : 主时钟频率
 f_{fr} : 内部振荡时钟频率
 f_{xt} : 副时钟频率

8.4 操作

注意事项 不可以将 **TM0CMP0** 寄存器 设置为 **FFFFH**。

8.4.1 间隔定时器模式

在间隔定时器模式下，如果 **TM0CTL0.TM0CE** 位被设置为 1，中断请求信号(**INTTM0EQ0**)将会按照指定的间隔时间周期性产生。

图 8-2. 间隔定时器的结构图

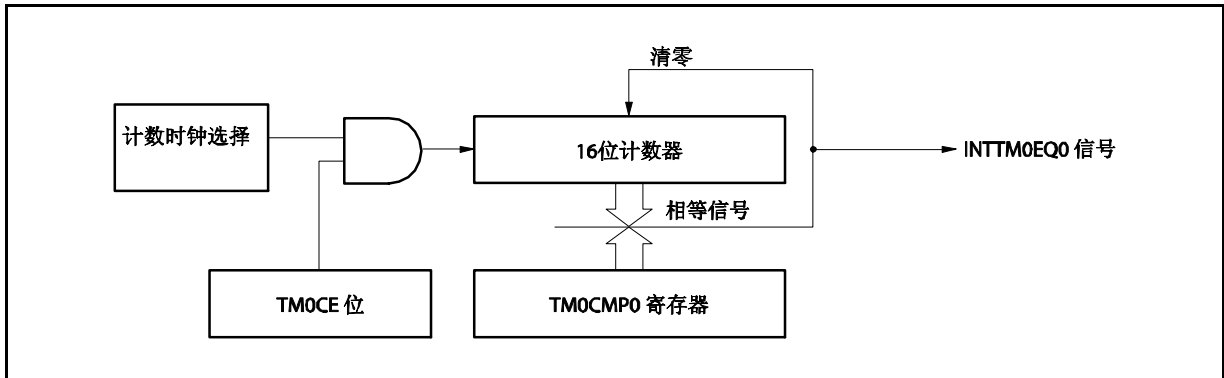
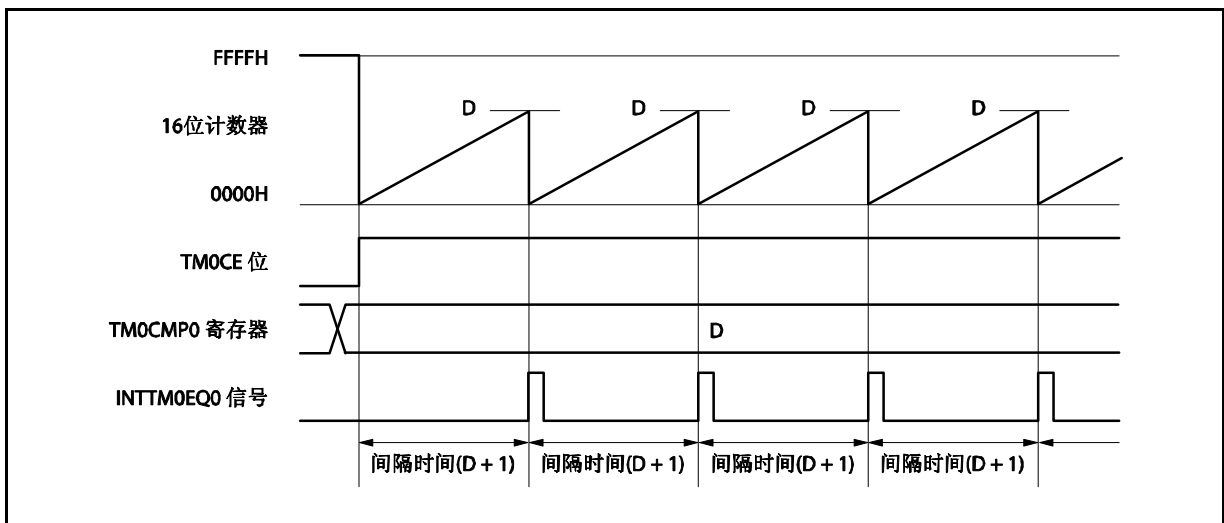


图 8-3. 间隔定时器模式下的基本操作时序



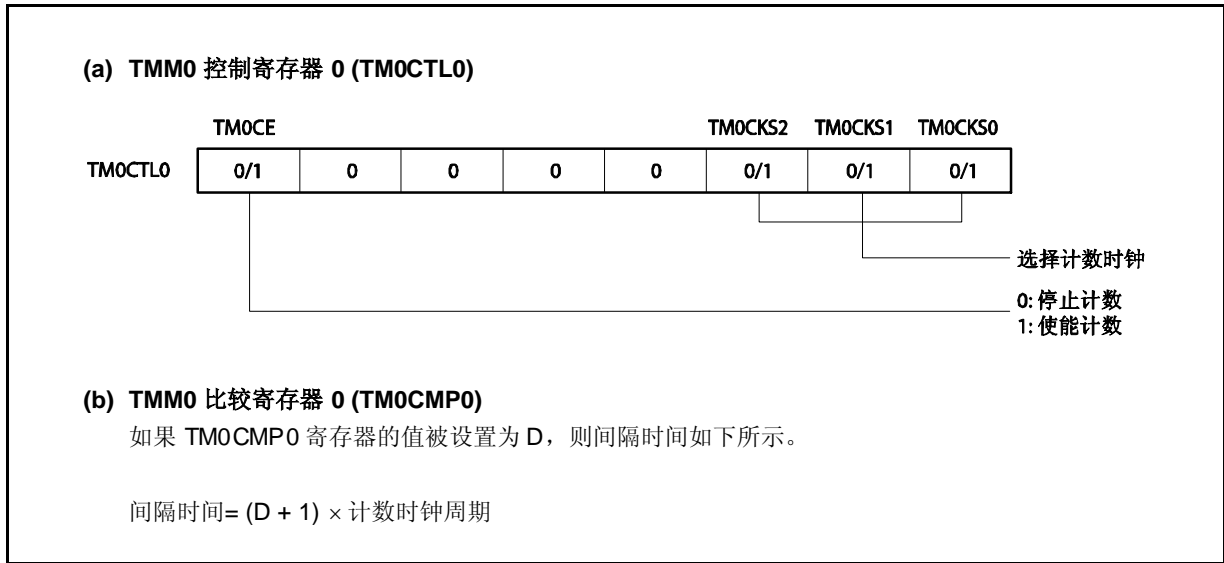
当 **TM0CE** 位被设置为 1 时，16 位计数器会与计数时钟同步将计数值从 **FFFFH** 清为 **0000H**，并且计数器开始计数。

当 16 位计数器的计数值与 **TM0CMP0** 寄存器中的值相等时，16 位计数器被清为 **0000H**，并同时产生一个比较相等中断请求信号(**INTTM0EQ0**)。

间隔时间可由下列算式求得。

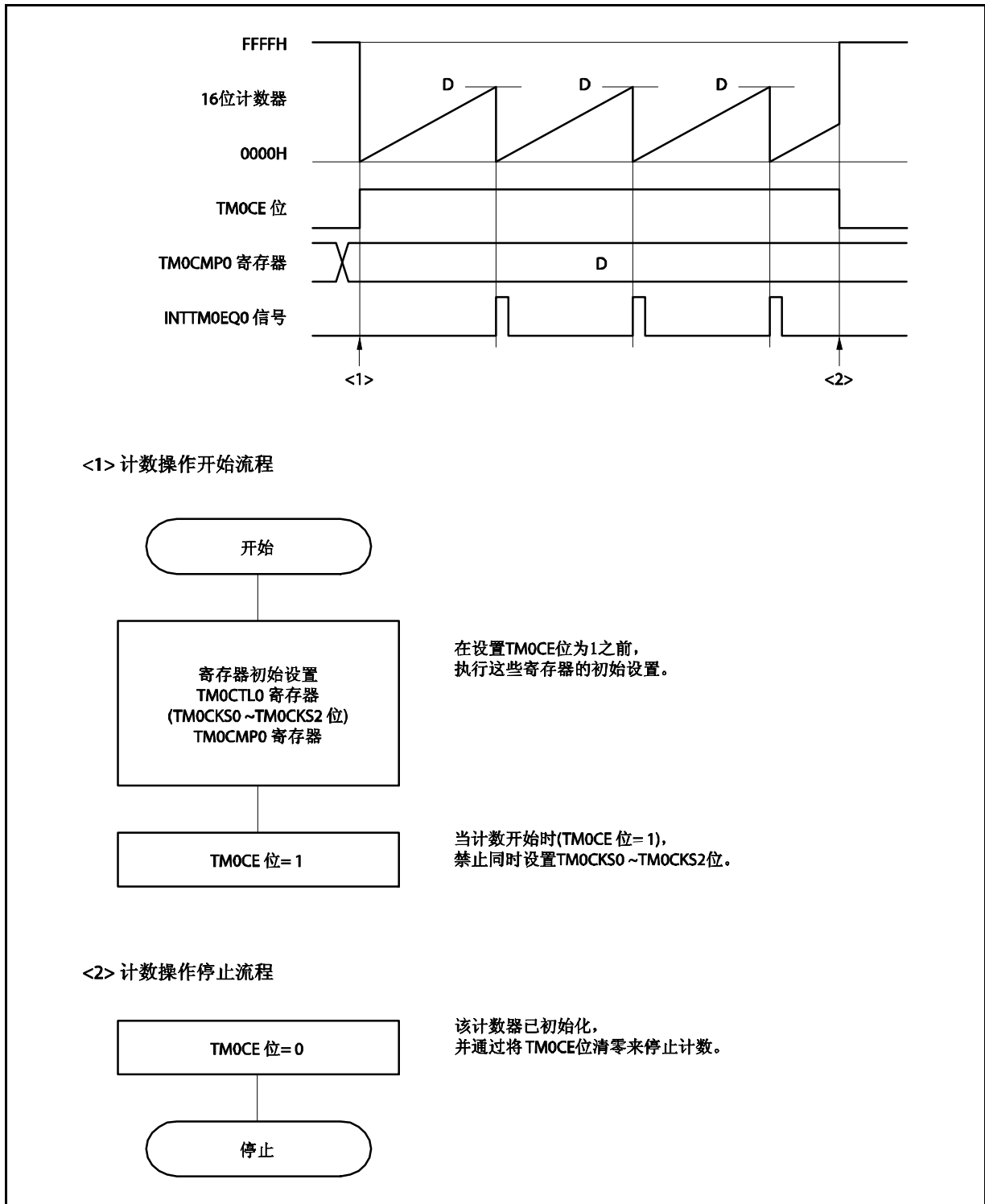
$$\text{间隔时间} = (\text{TM0CMP0 寄存器的设置值} + 1) \times \text{计数时钟周期}$$

图 8-4. 间隔定时器模式操作的寄存器设置



(1) 间隔定时器模式的操作流程

图 8-5. 间隔定时器模式下的软件处理流程

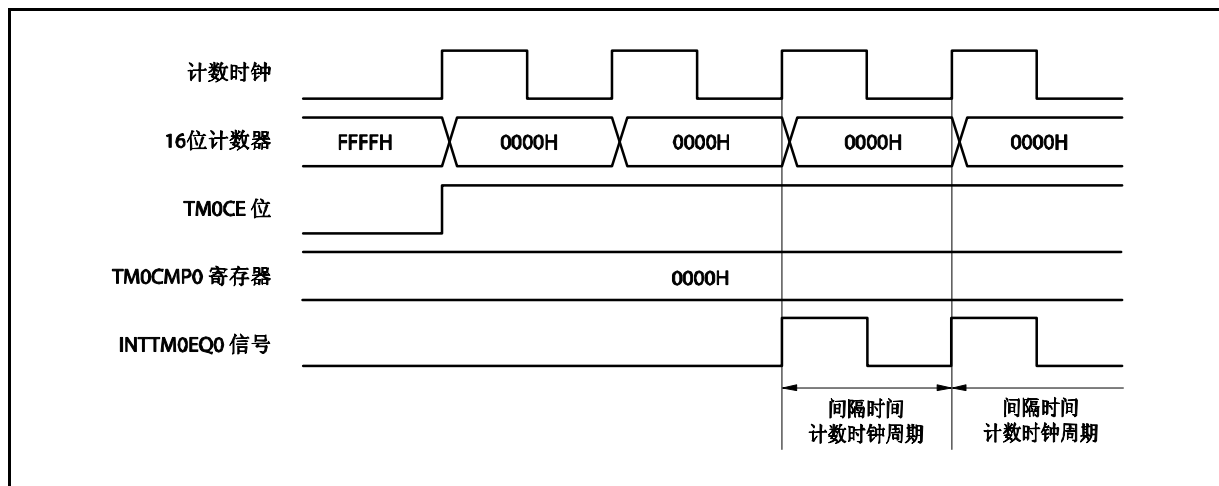


(2) 间隔定时器模式操作时序

注意事项 不可以将 **TM0CMP0** 寄存器设置为 **FFFFH**。

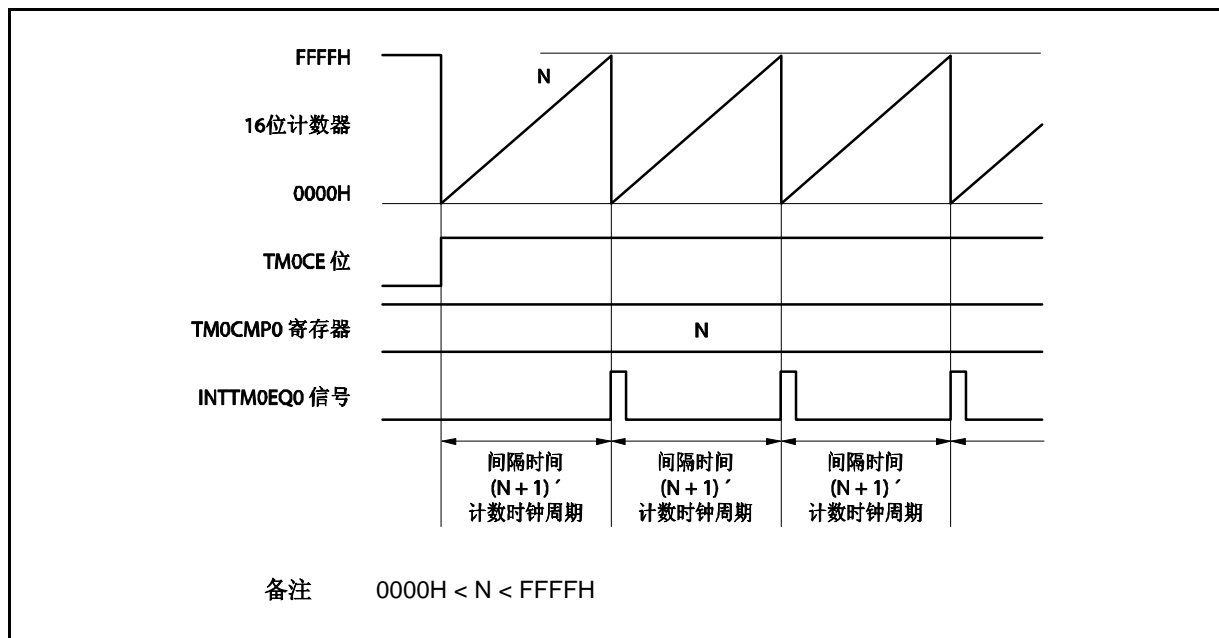
(a) **TM0CMP0** 寄存器被设置为 **0000H** 时的操作

如果 **TM0CMP0** 寄存器被设置为 **0000H**，则每个计数时钟到来时刻都会产生 **INTTM0EQ0** 信号。16 位计数器的值始终保持为 **0000H**。



(b) **TM0CMP0** 寄存器被设置为 **N** 时的操作

如果 **TM0CMP0** 寄存器被设置为 **N**，则 16 位计数器会累加计数到 **N**。然后与下一个计数时钟同步被清为 **0000H**，并产生 **INTTM0EQ0** 信号。



8.4.2 注意事项

- (1) 根据选择的计数时钟的不同，在 `TM0CTL0.TM0CE` 位被设置为 1 后，采用 16 位计数器达到以下时间间隔就开始计数。

所选计数时钟	开始计数前的最大时间
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) TMM0 工作时，禁止改写 `TM0CMP0` 和 `TM0CTL0` 寄存器。

如果在 `TM0CE` 位 = 1 时对这两个寄存器进行改写，那么操作结果将不能被保证。

如果这两个寄存器被错误地改写，请将 `TM0CTL0.TM0CE` 位清零，然后复位这些寄存器。

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。

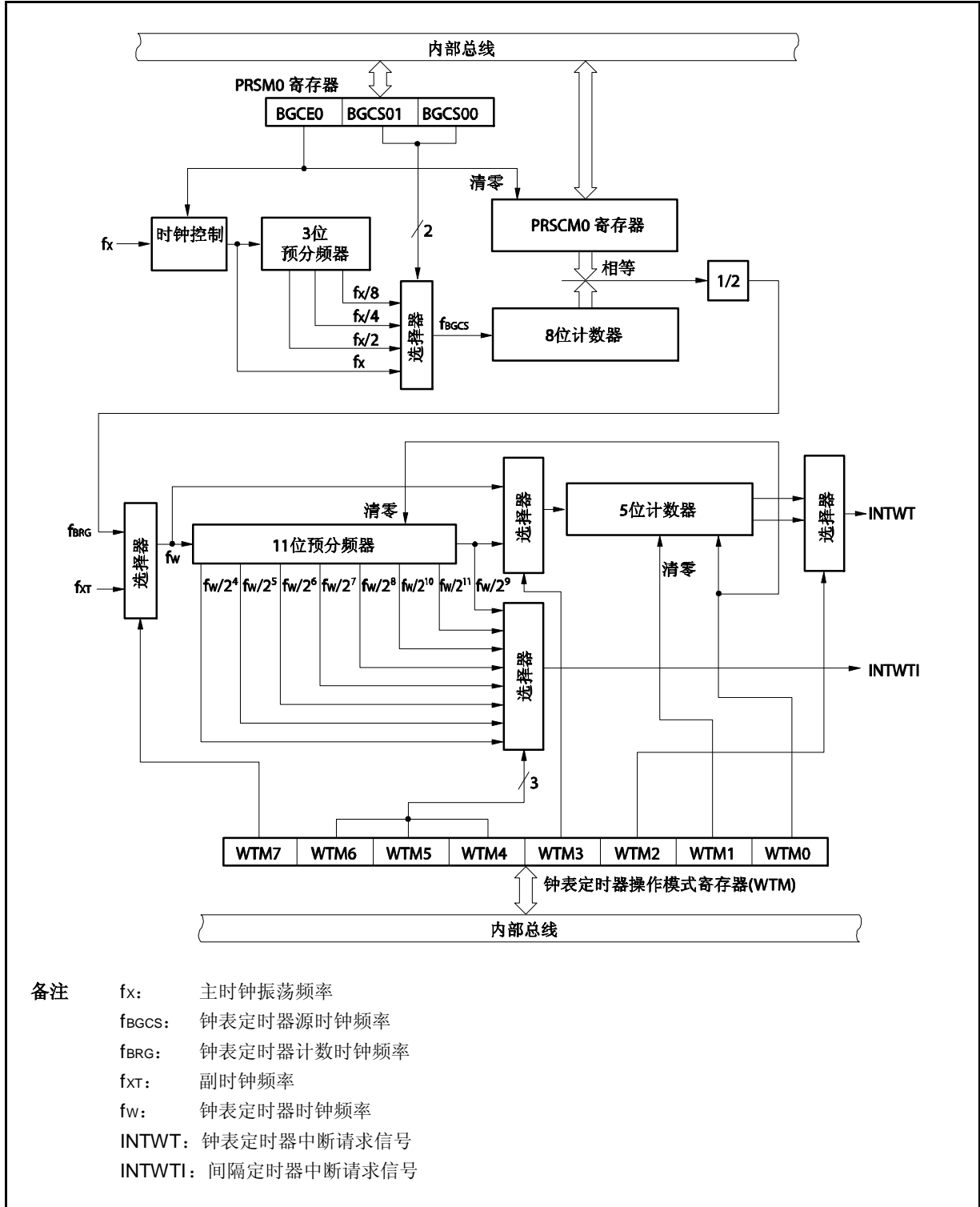
- 钟表定时器： 当使用主时钟或副时钟时，以 0.5 或 0.25 秒的间隔产生中断请求信号(INTWT)。
- 间隔定时器： 以设定的间隔时间产生中断请求信号(INTWTI)。

钟表定时器和间隔定时器可以同时使用。

9.2 配置

钟表定时器的框图如下。

图 9-1. 钟表定时器的框图



- 备注
- f_x : 主时钟振荡频率
 - f_{BGCS} : 钟表定时器源时钟频率
 - f_{BRG} : 钟表定时器计数时钟频率
 - f_{XT} : 副时钟频率
 - f_w : 钟表定时器时钟频率
 - INTWT: 钟表定时器中断请求信号
 - INTWTI: 间隔定时器中断请求信号

(1) 时钟控制

当钟表定时器工作于主时钟时，该模块用于控制操作时钟(f_x)的供给和停止。

(2) 3 位预分频器

该预分频器将 f_x 分频为 $f_x/2$ ， $f_x/4$ ，或 $f_x/8$ 。

(3) 8 位计数器

该 8 位计数器以源时钟(f_{BCKS})为计数时钟进行计数。

(4) 11 位预备分频器

该预分频器将 f_w 分频为 $f_w/2^4 \sim f_w/2^{11}$ 的时钟。

(5) 5 位计数器

该计数器以 f_w 或 $f_w/2^9$ 为计数时钟进行计数，以 $2^4/f_w$ ， $2^5/f_w$ ， $2^{12}/f_w$ ，或 $2^{14}/f_w$ 的间隔产生一个钟表定时器中断请求信号。

(6) 选择器

钟表定时器有以下 5 种选择器。

- 选择 f_x ， $f_x/2$ ， $f_x/4$ ，或 $f_x/8$ 之一为钟表定时器的源时钟作为选择器
- 选择主时钟(f_x)或副时钟(f_{XT})为钟表定时器时钟的选择器
- 选择 f_w 或 $f_w/2^9$ 为 5 位计数器的计数时钟频率的选择器
- 选择 $2^4/f_w$ ， $2^{13}/f_w$ ， $2^5/f_w$ ，或 $2^{14}/f_w$ 为信号 INTWT 信号产生时间间隔的选择器
- 选择 $2^4/f_w \sim 2^{11}/f_w$ 为间隔定时器中断请求信号(INTWTI) 产生时间间隔的选择器

(7) PRSCM 寄存器

该寄存器是用于设置间隔时间的 8 位比较寄存器。

(8) PRSM 寄存器

该寄存器控制提供给钟表定时器的时钟。

(9) WTM 寄存器

该寄存器是 8 位寄存器，用于控制钟表定时器/间隔定时器的操作，并设置中断请求信号产生间隔。

9.3 寄存器

钟表定时器使用下述寄存器。

- 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的产生。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF8B0H								
	7	6	5	4	3	2	1	0
PRSM0	0	0	0	BGCE0	0	0	BGCS01	BGCS00
BGCE0	使能主时钟操作							
0	禁止操作							
1	使能操作							
BGCS01	BGCS00	钟表定时器源时钟(f_{BGCS})的选择						
					5 MHz	4 MHz		
0	0	f_x			200 ns	250 ns		
0	1	$f_x/2$			400 ns	500 ns		
1	0	$f_x/4$			800 ns	1 μ s		
1	1	$f_x/8$			1.6 μ s	2 μ s		

注意事项

1. 钟表定时器操作期间不要改变的 BGCS00 和 BGCS01 位的值。
2. 在设置 BGCE0 位为 1 之前, 设置 PRSM0 寄存器。
3. 根据所使用的主时钟频率设置 PRSM0 和 PRSCM0 寄存器, 以获得 32.768 kHz 的 f_{BRG} 频率。

(2) 预分频器比较寄存器 0 (PRSCM0)

PRSCM0 寄存器是 8 位比较寄存器。

该寄存器支持 8 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF8B1H

	7	6	5	4	3	2	1	0
PRSCM0	PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00

- 注意事项**
1. 在钟表定时器操作期间，不要改写 PRSCM0 寄存器。
 2. 在设置 PRSM0.BGCE0 位为 1 之前，设置 PRSCM0 寄存器。
 3. 根据所使用的主时钟频率设置 PRSM0 和 PRSCM0 寄存器，以获得 32.768 kHz 的 f_{BRG} 频率。

f_{BRG} 的计算如下所示。

$$f_{BRG} = f_{BGCS}/2N$$

- 备注**
- f_{BGCS} : 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 ~ 256
但是，仅当 PRSCM0 寄存器设置为 00H 时，N = 256

(3) 钟表定时器操作模式寄存器 (WTM)

WTM 寄存器用于使能或禁止计数时钟和钟表定时器的操作，设置预分频器的间隔时间，控制 5 位计数器的操作，并且设置钟表标志的确立时间。

在设置 WTM 寄存器之前设置 PRSM0 寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

(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	预分频器间隔时间的选择
0	0	0	0	$2^4/f_w$ (488 ns: $f_w = f_{XT}$)
0	0	0	1	$2^5/f_w$ (977 ns: $f_w = f_{XT}$)
0	0	1	0	$2^6/f_w$ (1.95 ms: $f_w = f_{XT}$)
0	0	1	1	$2^7/f_w$ (3.91 ms: $f_w = f_{XT}$)
0	1	0	0	$2^8/f_w$ (7.81 ms: $f_w = f_{XT}$)
0	1	0	1	$2^9/f_w$ (15.6 ms: $f_w = f_{XT}$)
0	1	1	0	$2^{10}/f_w$ (31.3 ms: $f_w = f_{XT}$)
0	1	1	1	$2^{11}/f_w$ (62.5 ms: $f_w = f_{XT}$)
1	0	0	0	$2^4/f_w$ (488 ns: $f_w = f_{BRG}$)
1	0	0	1	$2^5/f_w$ (977 ns: $f_w = f_{BRG}$)
1	0	1	0	$2^6/f_w$ (1.95 ms: $f_w = f_{BRG}$)
1	0	1	1	$2^7/f_w$ (3.90 ms: $f_w = f_{BRG}$)
1	1	0	0	$2^8/f_w$ (7.81 ms: $f_w = f_{BRG}$)
1	1	0	1	$2^9/f_w$ (15.6 ms: $f_w = f_{BRG}$)
1	1	1	0	$2^{10}/f_w$ (31.2 ms: $f_w = f_{BRG}$)
1	1	1	1	$2^{11}/f_w$ (62.5 ms: $f_w = f_{BRG}$)

WTM7	WTM3	WTM2	钟表标志确立时间的选择
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 μ s: $f_w = f_{XT}$)
0	1	1	$2^4/f_w$ (488 μ 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 μ s: $f_w = f_{BRG}$)
1	1	1	$2^4/f_w$ (488 μ s: $f_w = f_{BRG}$)

WTM1	5位计数器操作的控制
0	在操作停止后清零
1	开始

WTM0	使能钟表定时器操作
0	停止操作(将预分频器和5位计数器清零)
1	使能操作

注意事项 当 WTM0 和 WTM1 位都为 0 时，改写 WTM2 ~ WTM7 位。

- 备注**
1. f_w : 钟表定时器时钟频率
 2. 括号中的值适用于 $f_w = 32.768$ kHz 时的操作
 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 钟表定时器的操作

钟表定时器以固定时间间隔产生中断请求信号(INTWT)。使用副时钟(32.768 kHz)或主时钟以 0.25 或 0.5 秒的时间间隔来操作钟表定时器。

当 WTM.WTM1 和 WTM.WTM0 位被设置为 11 时, 开始计数操作。当 WTM0 位被清零时, 11 位预分频器和 5 位计数器被清零并且计数操作停止。

当定时器被同时作为钟表定时器和间隔定时器使用时, 要通过先清零 WTM1 位再清零 5 位计数器的方法来调整时间。此时, 钟表定时器可能产生一个最大 15.6 ms 的错误, 而间隔定时器并不受影响。

如果主时钟作为钟表定时器的计数时钟, 使用 PRSM0.BGCS01 和 BGCS00 位设置计数时钟, 使用 PRSCM0 寄存器设置 8 位比较值, 以使钟表定时器的计数时钟频率(f_{BRG})为 32.768 kHz。

当 PRSM0.BGCEO 位被设置为 1 时, f_{BRG} 被用于钟表定时器。

可以使用下面的表达式计算 f_{BRG} 。

$$f_{BRG} = f_x / (2^{m+1} \times N)$$

要设置 f_{BRG} 为 32.768 kHz, 可进行下面计算并设置 BGCS01 和 BGCS00 位和 PRSCM0 寄存器。

<1> 设置 $N = f_x / 65, 536$ 。设置 $m = 0$ 。

<2> 当 N 的个位的舍入结果为偶数时, 在进行舍入之前, 请设置 $N = N/2$, $m = m + 1$ 。

<3> 重复步骤 <2> 直到 N 成为奇数或 $m = 3$ 。

<4> 将舍入后 N 值的整数部分设置到 PRSCM0 寄存器, 将 m 设置到 BGCS01 和 BGCS00 位中。

示例: 当 $f_x = 4.00$ MHz 时

<1> $N = 4, 000, 000 / 65, 536 = 61.03\dots$, $m = 0$

<2>, <3> 由于 N (舍入后的整数部分) 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, f_{BRG} 的实际频率如下所示。

$$\begin{aligned} f_{BRG} &= f_x / (2^{m+1} \times N) = 4, 000, 000 / (2 \times 61) \\ &= 32.787 \text{ kHz} \end{aligned}$$

备注 m : 分频值 (BGCS01 和 BGCS00 位的设置值) = 0 ~ 3

N : PRSCM0 寄存器的设置值 = 1 ~ 256

但是, 仅当 PRSCM0 寄存器被设置为 00H 时, $N = 256$

f_x : 主时钟振荡频率

9.4.2 间隔定时器的操作

钟表定时器可以用作间隔定时器，并按照预设的计数值周期性产生中断请求信号(INTWTI)。

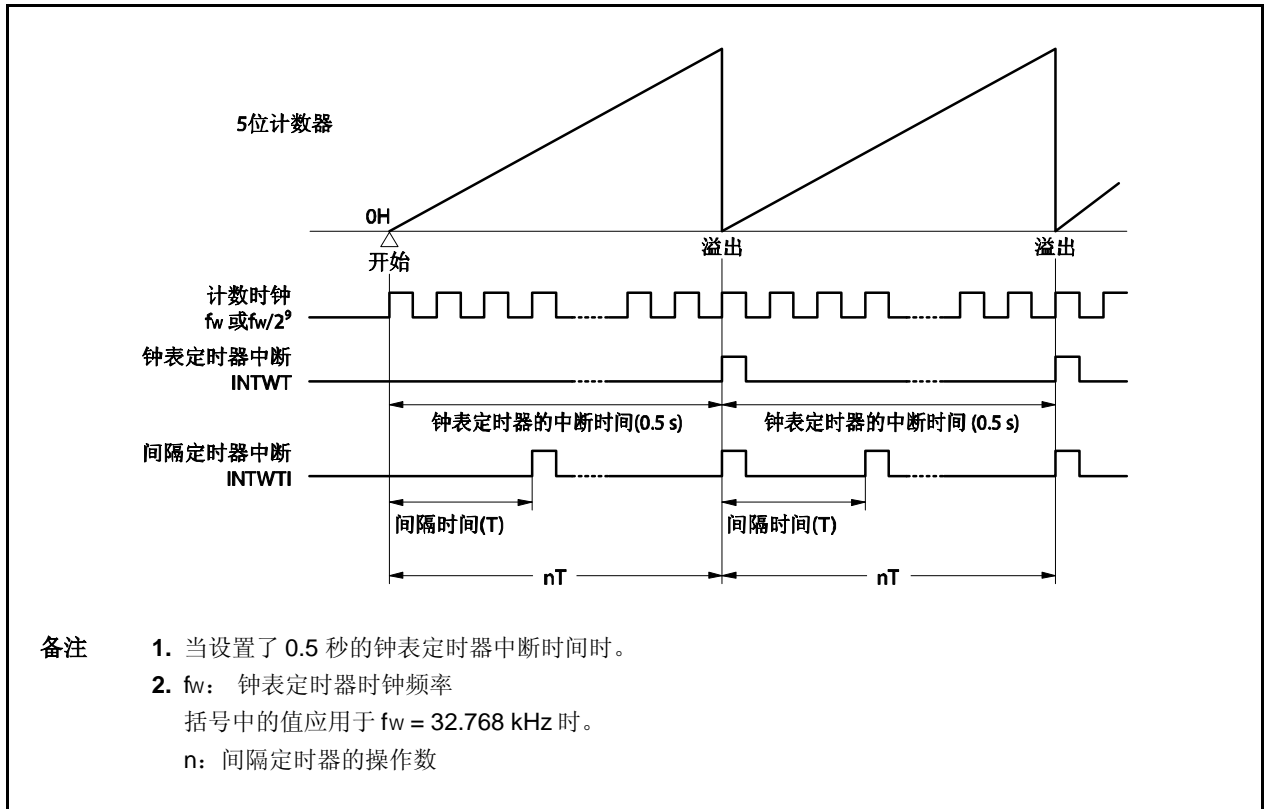
可由 WTM 寄存器的 WTM4 ~ WTM7 位选择间隔时间。

表 9-1. 间隔定时器的间隔时间

WTM7	WTM6	WTM5	WTM4	间隔时间	
0	0	0	0	$2^4 \times 1/fw$	488 μs (工作于 $fw = f_{XT} = 32.768$ kHz)
0	0	0	1	$2^5 \times 1/fw$	977 μs (工作于 $fw = f_{XT} = 32.768$ kHz)
0	0	1	0	$2^6 \times 1/fw$	1.95 ms (工作于 $fw = f_{XT} = 32.768$ kHz)
0	0	1	1	$2^7 \times 1/fw$	3.91 ms (工作于 $fw = f_{XT} = 32.768$ kHz)
0	1	0	0	$2^8 \times 1/fw$	7.81 ms (工作于 $fw = f_{XT} = 32.768$ kHz)
0	1	0	1	$2^9 \times 1/fw$	15.6 ms (工作于 $fw = f_{XT} = 32.768$ kHz)
0	1	1	0	$2^{10} \times 1/fw$	31.3 ms (工作于 $fw = f_{XT} = 32.768$ kHz)
0	1	1	1	$2^{11} \times 1/fw$	62.5 ms (工作于 $fw = f_{XT} = 32.768$ kHz)
1	0	0	0	$2^4 \times 1/fw$	488 μs (工作于 $fw = f_{BRG} = 32.768$ kHz)
1	0	0	1	$2^5 \times 1/fw$	977 μs (工作于 $fw = f_{BRG} = 32.768$ kHz)
1	0	1	0	$2^6 \times 1/fw$	1.95 ms (工作于 $fw = f_{BRG} = 32.768$ kHz)
1	0	1	1	$2^7 \times 1/fw$	3.91 ms (工作于 $fw = f_{BRG} = 32.768$ kHz)
1	1	0	0	$2^8 \times 1/fw$	7.81 ms (工作于 $fw = f_{BRG} = 32.768$ kHz)
1	1	0	1	$2^9 \times 1/fw$	15.6 ms (工作于 $fw = f_{BRG} = 32.768$ kHz)
1	1	1	0	$2^{10} \times 1/fw$	31.3 ms (工作于 $fw = f_{BRG} = 32.768$ kHz)
1	1	1	1	$2^{11} \times 1/fw$	62.5 ms (工作于 $fw = f_{BRG} = 32.768$ kHz)

备注 fw: 钟表定时器时钟频率

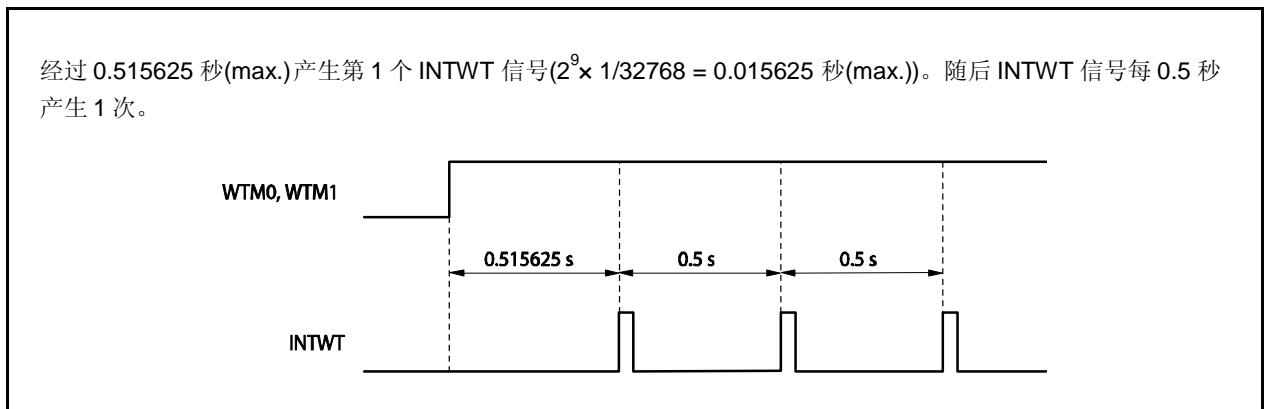
图 9-2. 钟表定时器/间隔定时器的操作时序



9.4.3 注意事项

在允许操作 ($WTM.WTM1$ 和 $WTM.WTM0$ 位 = 1) 后, 第一次钟表定时器中断请求信号 ($INTWT$) 产生前需要一段准备时间。

图 9-3. 钟表定时器产生中断请求信号 ($INTWT$) 的示例
(当中断周期= 0.5 s 时)



第十章 看门狗定时器 2 的功能

10.1 功能

看门狗定时器 2 的功能如下。

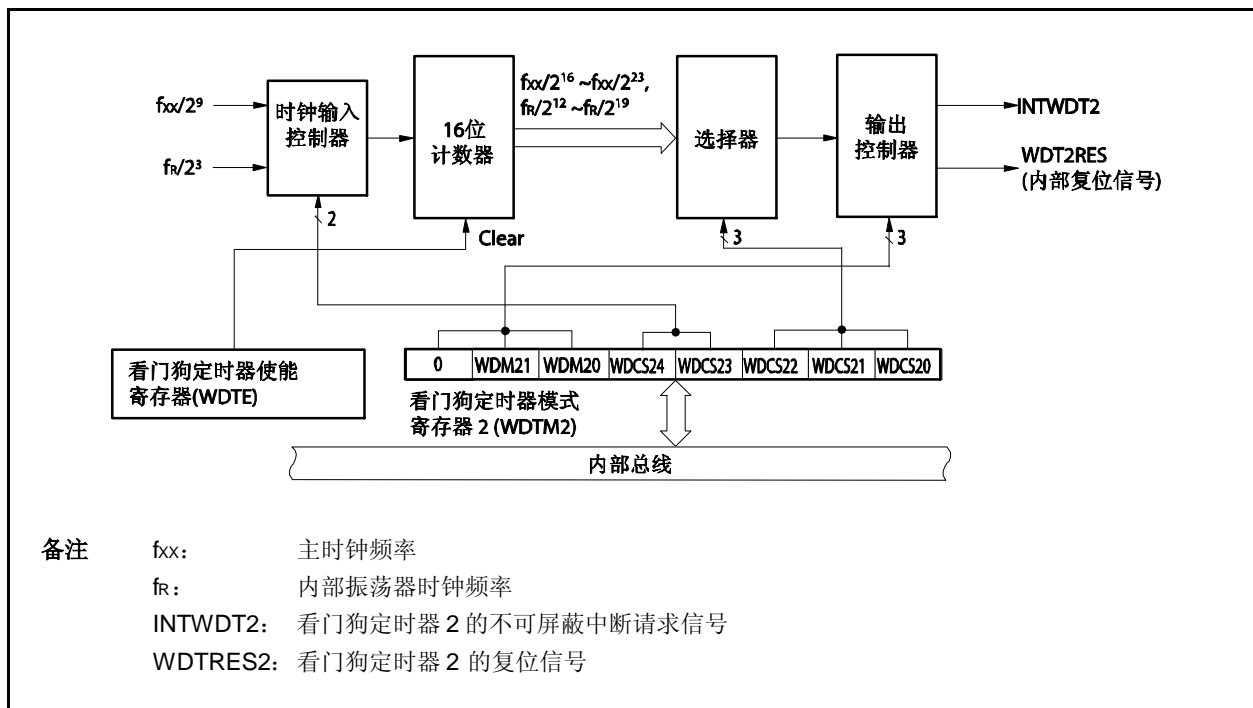
- 看门狗定时器默认开始工作^{注 1}
 - 复位模式：看门狗定时器 2 溢出时的复位操作（产生 WDT2RES 信号）
 - 不可屏蔽中断请求模式：看门狗定时器 2 溢出时的 NMI 操作（产生 INTWDT2 信号）^{注 2}
- 可从主时钟和内部振荡时钟选择作为源时钟输入

- 注**
1. 复位释放后，看门狗定时器 2 在复位模式下自动启动。
当不使用看门狗定时器 2 时，在复位前通过该功能使其停止工作，或清除看门狗定时器 2 并在下一次间隔开始前使其停止工作。
为了确认操作的正确性，即使在不需要改变默认设置（复位模式，间隔时间： $f_R/2^{19}$ ）的情况下，也要对 WDTM2 寄存器执行一次写操作。
 2. 由不可屏蔽中断请求信号(INTWDT2)引发的不可屏蔽中断服务，敬请参阅 **15.2.2 (2) INTWDT2 信号**。

10.2 配置

看门狗定时器 2 的框图如下。

图 10-1. 看门狗定时器 2 的框图



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) 看门狗定时器使能寄存器 (WDTE)

10.3 寄存器

(1) 看门狗定时器模式寄存器 2 (WDTM2)

WDTM2 寄存器设置溢出时间看门狗定时器 2 的操作时钟。

该寄存器支持 8 位读写方式。该寄存器可被多次读取，但它只能在复位释放后被写入一次。

该寄存器复位后的值为 67H。

注意事项 下述状态下，禁止访问 WDTM2 寄存器。关于更多细节，敬请参阅 3.4.8 (2) 访问特定的片上外围 I/O 寄存器。

- 当 CPU 使用副时钟进行操作并且主时钟振荡停止时
- 当 CPU 使用内部振荡时钟进行操作时

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 2 操作模式的选择
0	0	停止操作
0	1	不可屏蔽中断请求模式 (生成 INTWDT2 信号)
1	-	复位模式(生成 WDT2RES 信号)

注 如果通过选项字节功能将 OPB1 位设置为 1（参见第二十四章），则复位模式固定。

注意事项 1. 关于 WDCS20 ~ WDCS24 位的更多细节，敬请参阅表 10-2 看门狗定时器 2 时钟选择。

2. 如果 WDTM2 寄存器在复位后被改写两次，则必定产生溢出信号，计数器复位。

3. 要有意生成一个溢出信号，仅写入 WDTM2 寄存器两次或者写入一个非 ACH 的值到 WDTE 寄存器一次。

然而，当看门狗定时器 2 被设置为停止操作时，即使数据被写入 WDTM2 寄存器仅两次，或者非“ACH”值被写入 WDTE 寄存器仅一次，溢出信号也不会产生。

4. 要停止看门狗定时器 2 的操作，将 RCM.RSTOP 位设置为 1（停止内部振荡）并且将 1FH 写入 WDTM2 寄存器。如果通过选项字节功能将 OPB1 位设置为 1（参见第二十四章），但是，看门狗定时器 2 不能通过复位之外的任何方式停止。

<R>

<R>

表 10-2. 看门狗定时器 2 时钟选择

WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	所选择的时钟	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 _{xx} = 4 MHz		f _{xx} = 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	停止操作			

注意事项 如果通过选项字节功能将 **OPB1** 位设置为 1，则时钟被固定为内部振荡时钟(f_R)（可选择 $2^{12}/f_R \sim 2^{19}/f_R$ ）。关于更多细节，敬请参阅第二十四章 选项字节功能。

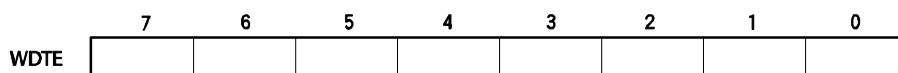
(2) 看门狗定时器使能寄存器 (WDTE)

通过将“ACH”写入 WDTE 寄存器，看门狗定时器 2 的计数器被清零，计数重新开始。

WDTE 寄存器支持 8 位读写方式。

该寄存器复位后的值为 9AH。

复位后: 9AH R/W 地址: FFFF6D1H



- 注意事项**
1. 当非“ACH”值被写入 WDTE 寄存器时，必定输出溢出信号。
 2. 当向 WDTE 寄存器执行 1 位存储器操作指令时，必定输出溢出信号。
 3. 要有意生成一个溢出信号，仅写入 WDTM2 寄存器两次或者写入一个非 ACH 的值到 WDTE 寄存器一次。
然而，当看门狗定时器 2 被设置为停止操作时，即使数据被写入 WDTM2 寄存器仅两次，或者非“ACH”值被写入 WDTE 寄存器仅一次，溢出信号也不会产生。
 4. WDTE 寄存器的读取值为“9AH”（与写入值“ACH”不同）。

<R>

10.4 操作

复位释放后，看门狗定时器 2 在复位模式下自动启动。

采用字节访问复位后，WDTM2 寄存器只可写入一次。要使用看门狗定时器 2，可使用 8 位存储器操作指令将操作模式和间隔定时器写入到 WDTM2 寄存器。之后，看门狗定时器 2 的操作将无法停止。

WDTM2 寄存器的 WDCS24 ~ WDCS20 位被用于选择看门狗定时器 2 循环检测时间间隔。

将 ACH 写入 WDTE 寄存器可清零看门狗定时器 2 的计数器，并重新开始计数操作。在计数操作已经开始后，在循环检测时间间隔内将 ACH 写入 WDTE。

如果在时间间隔记满时没有将 ACH 写入 WDTE 寄存器，就会根据 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 位被设置为 1 时（复位模式），如果在复位或待机释放后振荡稳定期间产生 WDT 溢出，将不产生内部复位且 CPU 时钟将转换成内部振荡时钟。

要不使用看门狗定时器 2，可将 1FH 写入 WDTM2 寄存器。

关于在设置为不可屏蔽中断请求模式时的不可屏蔽中断服务，敬请参阅 15.2.2 (2) INTWDT2 信号。

第十一章 A/D 转换器

11.1 概述

A/D 转换器用于将模拟输入信号转换为数字信号，具有 10 位分辨率，最多可由 16 个通道（ANI0 ~ ANI15）组成。

A/D 转换器具有以下特点。

- 10 位分辨率
- 24 通道
- 逐次逼近法
- 操作电压： $AV_{REF0} = 4.0 \sim 5.5 V$
- 模拟输入电压： $0 V \sim AV_{REF0}$
- 以下功能为所能提供的操作模式。
 - 连续选择模式
 - 连续扫描模式
 - 单脉冲扫描模式
- 以下功能为所能提供的触发模式。
 - 软件触发模式
 - 外部触发模式（外部，1）
 - 定时器触发模式
- 掉电检测功能（转换结果比较功能）

11.2 功能

(1) 10 位分辨率 A/D 转换

从 ANI0 到 ANI15 选择一个模拟通道，A/D 转换操作以 10 位分辨率重复进行。每当 A/D 转换结束，都产生一个中断请求信号(INTAD)。

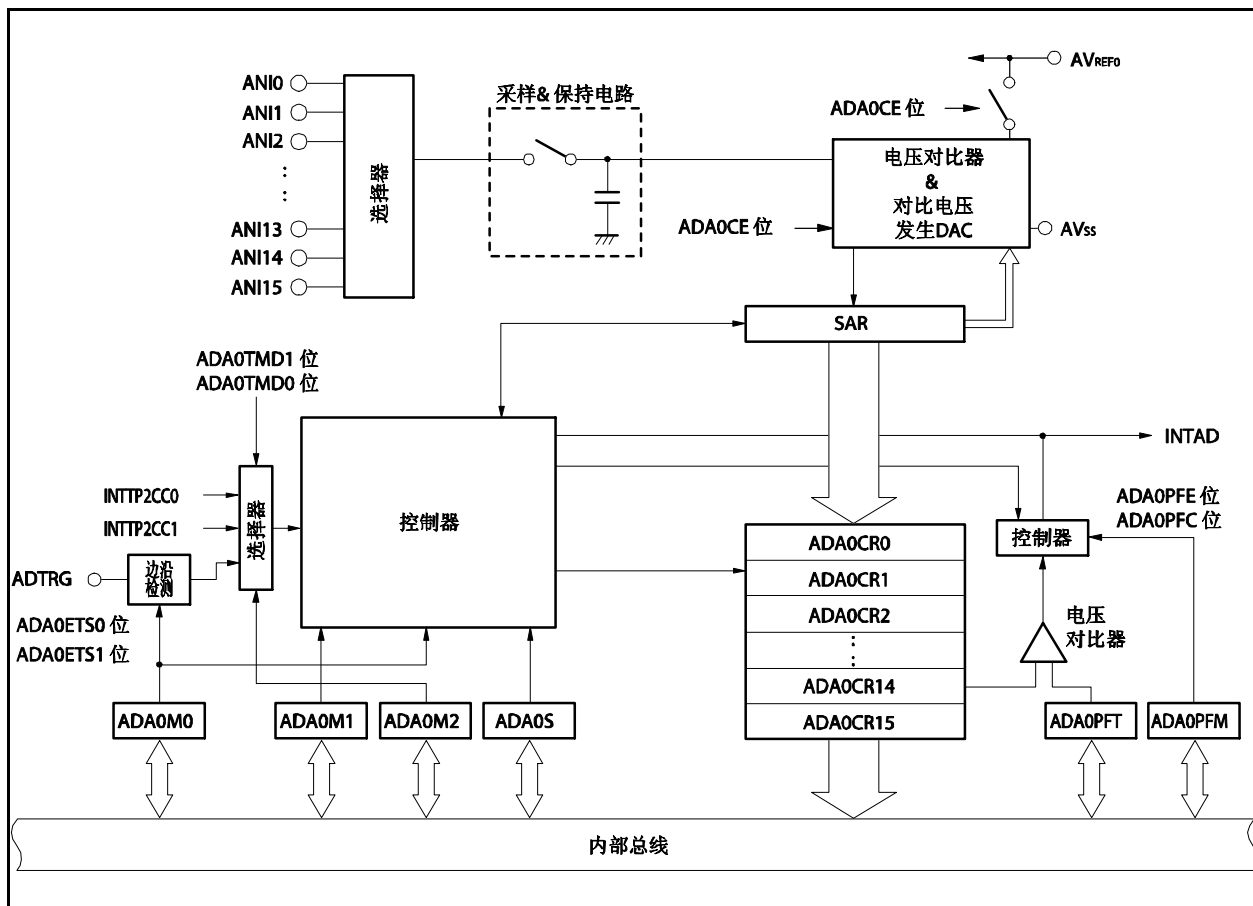
(2) 掉电检测功能

该功能用于检测电池电压的下降。A/D 转换的结果（ADA0CRnH 寄存器的值）与寄存器 ADA0PFT 的值进行比较，只有当指定的比较条件($n = 0 \sim 15$)满足时，才产生中断信号 INTAD。

11.3 配置

如下所示为 A/D 转换器的框图。

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	16 通道 (ANI0 ~ ANI15 引脚)
寄存器	逐次逼近寄存器 (SAR) A/D 转换结果寄存器 0 ~ 15 (ADA0CR0 ~ ADA0CR15) A/D 转换结果寄存器 0H ~ 15H (ADCR0H ~ ADCR15H): 仅可读取高 8 位
控制寄存器	A/D 转换器模式寄存器 0 ~ 2 (ADA0M0 ~ ADA0M2) A/D 转换器通道指定寄存器 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生成 DAC（比较电压）的输出电压进行比较，保持比较结果。比较结果从最高有效位(MSB)开始。

当比较结果被致以最低有效位（LSB）（也就是说，当 A/D 完全转换时），SAR 寄存器的结果被传送到寄存器 ADA0CRn。

备注 $n = 0 \sim 15$

(2) A/D 转换结果寄存器 n (ADA0CRn)，A/D 转换结果寄存器 nH (ADA0CRnH)

ADA0CRn 寄存器是用于存储 A/D 转换结果的 16 位寄存器。ADA0ARn 由 16 个寄存器组成根据输入模拟信号，A/D 转换结果存储于 ADA0CRn 寄存器的高 10 位。（低 6 位一定要为 0。）

(3) A/D 转换模式寄存器 0 (ADA0M0)

该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 转换模式寄存器 1 (ADA0M1)

该寄存器设置被转换的模拟输入信号的转换时间。

(5) A/D 转换模式寄存器 2 (ADA0M2)

该寄存器设置硬件触发模式。

(6) A/D 转换器通道指定寄存器 (ADA0S)

该寄存器设置输入端口，该输入端口输入被转换的模拟电压。

(7) 掉电比较模式寄存器 (ADA0PFM)

该寄存器设置掉电检测模式。

(8) 掉电比较阈值寄存器 (ADA0PFT)

ADA0PFT 寄存器设置一个与 A/D 转换结果寄存器 nH (ADA0CRnH)的值相比较的阈值。ADA0PFT 寄存器中所设置的 8 位数据与 A/D 转换结果寄存器 (ADA0CRnH)的高 8 位相比较。

(9) 控制器

当 A/D 转换完成或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 寄存器的值进行比较，仅当指定的比较条件满足时，才产生中断信号 INTAD。

(10) 采样与保持电路

采样与保持电路采集每个模拟输入信号，这些信号由输入电路选择并将采样数据发送到电压比较器。该电路在 A/D 转换期间保持被采样的模拟输入信号电压。

(11) 电压比较器

电压比较器比较被采样的电压值，并使用比较电压生成 DAC 的电压值来保持。

(12) 比较电压生成 DAC

该比较电压生成 DAC 连接在 AVREF0 和 AVSS 之间，生成一个与模拟输入信号比较的电压。

(13) ANI0 ~ ANI15 引脚

ANI0 ~ ANI15 是 16 个 A/D 转换器通道的模拟输入引脚，用于输入待转换成数字信号的模拟信号。没有被 ADA0S 寄存器选定作为模拟输入引脚可以用于输入端口引脚。

- 注意事项**
1. 确保输入 ANI0 ~ ANI15 的电压不超过额定值。特别是，如果大于或等于 AVREF0 的电压输入某个通道，这个通道的转换值将不确定，其它通道的转换值也将受到影响。
 2. 模拟输入引脚(ANI0 ~ ANI15) 复用用作输入端口引脚(P70 ~ P715)。如果 ANI0 ~ ANI15 中的任何引脚被选去执行 A/D 转换，在转换期间不要执行输入指令到端口 7。否则转换分辨率将被降级。

(14) AVREF0 引脚

该引脚用于输入 A/D 转换器的参考电压。即使没有使用 A/D 转换器，也应保持该引脚电压与 VDD 引脚的一致性。输入到引脚 ANI0 ~ ANI15 的信号基于引脚 AVREF0 和 AVSS 引脚之间的电压被转换成数字信号。

(15) AVSS 引脚

该引脚为 A/D 转换器的地引脚。即使没有使用 A/D 转换器，该引脚的电压也应与 VSS 引脚的电压保持一致。

11.4 寄存器

A/D 转换器由以下寄存器进行控制。

- A/D 转换器模式寄存器 0, 1, 2 (ADA0M0, ADA0M1, ADA0M2)
- A/D 转换器通道指定寄存器 0 (ADA0S)
- 掉电比较模式寄存器 (ADA0PFM)

还使用下面的寄存器。

- 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 转换模式寄存器 0 (ADA0M0)

ADA0M0 寄存器是用于指定操作模式并控制转换操作的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。但是，ADA0EF 位是只读的。

该寄存器复位后的值为 00H。

注意事项 下述情况下禁止访问 ADA0M0 寄存器。关于更多细节，敬请参阅 3.4.8 (2) 访问特定的片上外围 I/O 寄存器。

- 当 CPU 使用副时钟，且主时钟振荡停止时
- 当 CPU 使用内部振荡时钟时

复位后: 00H R/W 地址: FFFF200H

	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	持续选择模式
0	1	持续扫描模式
1	0	禁止设置
1	1	单脉冲扫描模式

ADA0ETS1	ADA0ETS0	外部触发(ADTRG 引脚) 输入有效边沿规格
0	0	无边沿检测
0	1	下降沿检测
1	0	上升沿检测
1	1	上升和下降沿检测

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 转换状态显示
0	A/D 转换停止
1	A/D 转换进行

- 注意事项
1. 第 0 位的写操作将被忽略。
 2. 当 A/D 转换使能(ADA0CE 位= 1)时, 禁止改变 ADA0M1 寄存器的值。
 3. 在 A/D 转换期间(ADA0EF 位= 1), 如果 ADA0M0, ADA0M2, ADA0S, ADA0PFM 和 ADA0PFT 寄存器被写入, 将根据模式进行下述操作。
 - 软件触发模式
A/D 转换停止并从头开始。
 - 硬件触发模式
A/D 转换停止并设置为触发待机状态。
 4. 当不使用 A/D 转换器时, 设置 ADA0CE 位为 0 停止操作以减少功耗。
 5. 在 A/D 转换后输入引脚数据第一次的转换分辨率可能会下降。关于更多细节, 敬请参阅 11.6 (7) AVREF0 引脚。

(2) A/D 转换模式寄存器 1 (ADA0M1)

ADA0M1 寄存器是用于控制转换时间规范的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFFF201H

	7	6	5	4	3	2	1	0
ADA0M1	ADA0HS1	0	0	0	ADA0FR3	ADA0FR2	ADA0FR1	ADA0FR0

注意事项 1. 确保将第 6 ~ 4 位清零。
2. 确保将 ADA0HS1 位设置为“1”。

备注 关于 A/D 转换时间设置示例, 参见表 11-2。

表 11-2. 转换模式下的设置示例

ADA0HS1	ADA0FR3 ~ ADA0FR0				A/D 转换时间	f _{xx} = 20 MHz	f _{xx} = 16 MHz	f _{xx} = 4 MHz	A/D 稳定时间 ^注
	3	2	1	0					
1	0	0	0	0	31/f _{xx}	禁止设置	禁止设置	7.75 μs	16/f _{xx}
	0	0	0	1	62/f _{xx}	3.10 μs	3.88 μs	15.50 μs	31/f _{xx}
	0	0	1	0	93/f _{xx}	4.65 μs	5.81 μs	禁止设置	47/f _{xx}
	0	0	1	1	124/f _{xx}	6.20 μs	7.75 μs	禁止设置	50/f _{xx}
	0	1	0	0	155/f _{xx}	7.75 μs	9.69 μs	禁止设置	50/f _{xx}
	0	1	0	1	186/f _{xx}	9.30 μs	11.63 μs	禁止设置	50/f _{xx}
	0	1	1	0	217/f _{xx}	10.85 μs	13.56 μs	禁止设置	50/f _{xx}
	0	1	1	1	248/f _{xx}	12.40 μs	15.50 μs	禁止设置	50/f _{xx}
	1	0	0	0	279/f _{xx}	13.95 μs	禁止设置	禁止设置	50/f _{xx}
	1	0	0	1	310/f _{xx}	15.50 μs	禁止设置	禁止设置	50/f _{xx}
	1	0	1	0	341/f _{xx}	禁止设置	禁止设置	禁止设置	50/f _{xx}
	1	0	1	1	372/f _{xx}	禁止设置	禁止设置	禁止设置	50/f _{xx}
	1	1	0	0	403/f _{xx}	禁止设置	禁止设置	禁止设置	50/f _{xx}
	1	1	0	1	434/f _{xx}	禁止设置	禁止设置	禁止设置	50/f _{xx}
	1	1	1	0	465/f _{xx}	禁止设置	禁止设置	禁止设置	50/f _{xx}
	1	1	1	1	496/f _{xx}	禁止设置	禁止设置	禁止设置	50/f _{xx}

注 当 ADA0M0 寄存器的 ADA0CE 位从 0 改变为 1 以保证 A/D 转换稳定时间时, 在输入上述其中一个时钟值之后, 第一次 A/D 转换开始。

- <R> **注意事项** 1. 设置 $3.1 \mu\text{s} \leq \text{转换时间} \leq 15.5 \mu\text{s}$.
- <R> 2. 禁止改写 ADA0M0, ADA0M2, ADA0S, ADA0PFM 和 ADA0PFT 寄存器及在稳定时间内的触发输入。

(3) A/D 转换模式寄存器 2 (ADA0M2)

ADA0M2 寄存器指定硬件触发模式。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFFF203H

	7	6	5	4	3	2	1	0
ADA0M2	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 硬件有效边沿)
0	1	定时器触发模式0 (当发生 INTTP2CC0 中断请求)
1	0	定时器触发模式 1 (当发生INTTP2CC1 中断请求)
1	1	禁止设置

注意事项 确保将第 7 ~ 2 清零。

(4) A/D 转换器通道指定寄存器 0 (ADA0S)

ADA0S 寄存器指定输入模拟电压的引脚，该引脚的信号将被转换为数字信号。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: 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 ~ ANI2
0	0	1	1	ANI3	ANI0 ~ ANI3
0	1	0	0	ANI4	ANI0 ~ ANI4
0	1	0	1	ANI5	ANI0 ~ ANI5
0	1	1	0	ANI6	ANI0 ~ ANI6
0	1	1	1	ANI7	ANI0 ~ ANI7
1	0	0	0	ANI8	ANI0 ~ ANI8
1	0	0	1	ANI9	ANI0 ~ ANI9
1	0	1	0	ANI10	ANI0 ~ ANI10
1	0	1	1	ANI11	ANI0 ~ ANI11
1	1	0	0	ANI12	ANI0 ~ ANI12
1	1	0	1	ANI13	ANI0 ~ ANI13
1	1	1	0	ANI14	ANI0 ~ ANI14
1	1	1	1	ANI15	ANI0 ~ ANI15
以上除外				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。

该寄存器支持 16 位或 8 位只读方式。但是，指定 16 位存储器操作指令访问 ADA0CRn 寄存器，8 位存储器操作指令访问 ADA0CRnH 寄存器。10 位转换结果从 ADA0CRn 寄存器的高 10 位读取，低 6 位的读取值则为 0。转换结果的高 8 位从 ADA0CRnH 寄存器读取。

注意事项 在下述状态中禁止访问 ADA0CRn 和 ADA0CRnH 寄存器。关于更多细节，敬请参阅 3.4.8 (2) 访问特定片上外围 I/O 寄存器

- 当 CPU 使用副时钟，且主时钟振荡停止时
- 当 CPU 使用内部振荡时钟时

复位后： 不确定 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,
 ADA0CR12 FFFFF228H, ADA0CR13 FFFFF22AH,
 ADA0CR14 FFFFF22CH, ADA0CR15 FFFFF22EH

	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： 无定义 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,
 ADA0CR12H FFFFF229H, ADA0CR13H FFFFF22BH,
 ADA0CR14H FFFFF22DH, ADA0CR15H FFFFF22FH

	7	6	5	4	3	2	1	0
ADA0CRnH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

备注 n = 0 ~ 15

注意事项 ADA0M0 和 ADA0S 寄存器的写操作可能会引起 ADA0CRn 寄存器的内容不确定。转换后，在写入 ADA0M0 和 ADA0S 寄存器之前读取转换结果。如果不按上述顺序进行，正确的转换结果可能不会被读取。

如下为模拟电压输入到模拟输入引脚(AN10 ~ AN123)和 A/D 转换结果 (ADA0CRn 寄存器) 的关系。

$$SAR = INT \left(\frac{V_{IN}}{AV_{REF0}} \times 1,024 + 0.5 \right)$$

$$ADA0CRn = SAR \times 64$$

或,

$$(SAR - 0.5) \times \frac{AV_{REF0}}{1,024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{REF0}}{1,024}$$

INT(): 函数, 返回()中的整数值

V_{IN}: 模拟输入电压

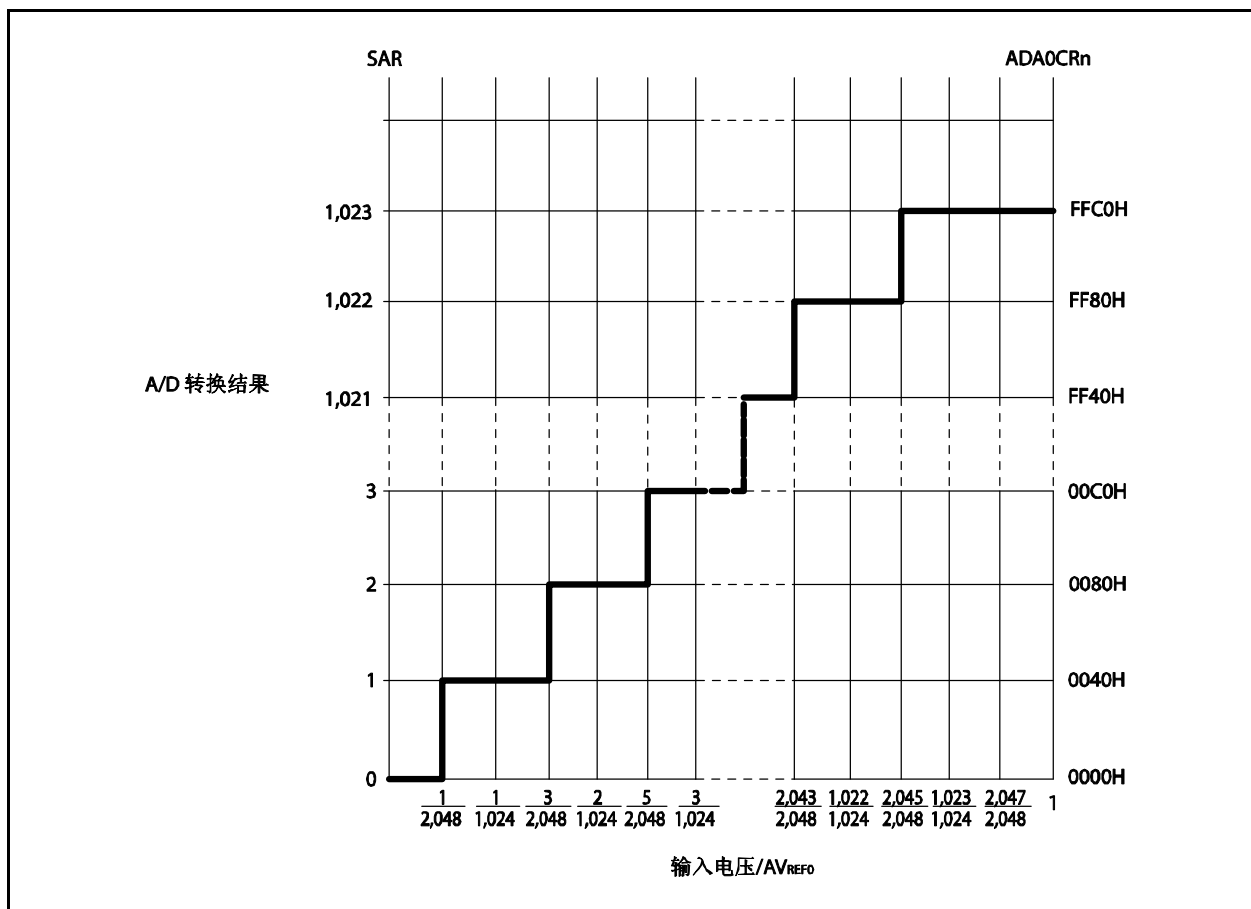
AV_{REF0}: AV_{REF0} 引脚电压

ADA0CR: Value of ADA0CRn 寄存器

注 ADA0CRn 寄存器的低 6 位恒 0。

如下显示了模拟输入电压和 A/D 转换结果之间的关系。

图 11-2. 模拟输入电压 和 A/D 转换结果之间的关系



(6) 掉电比较模式寄存器 (ADA0PFM)

ADA0PFM 寄存器是设置掉电比较模式的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFFF204H								
	7	6	5	4	3	2	1	0
ADA0PFM	ADA0PFE	ADA0PFC	0	0	0	0	0	0

ADA0PFE	掉电对比启用/禁用选择
0	掉电对比禁用
1	掉电对比启用

ADA0PFC	掉电对比模式选择
0	当 ADA0CRnH ³ ADA0PFT 发生一中断请求信号 (INTAD)
1	当 ADA0CRnH < ADA0PFT 发生一中断请求信号 (INTAD)

注意事项

1. 选择模式下，ADA0PFT 寄存器所设置的 8 位数据与 ADA0CRnH 寄存器的值进行比较，ADA0CRnH 寄存器由 ADA0S 寄存器指定。如果结果与由 ADA0PFC 位指定的条件匹配，则转换结果将存储在 ADA0CRn 寄存器中并且产生中断信号 INTAD。如果不匹配，不产生中断信号 INTAD。
2. 扫描模式下，ADA0PFT 寄存器所设置的 8 位数据与 ADA0CR0H 寄存器的内容相比较。如果结果与由 ADA0PFC 位指定的条件匹配，则转换结果将存储在 ADA0CR0 寄存器中并且生成中断信号 INTAD。如果不匹配，不产生中断信号 INTAD。无论比较结果如何，扫描模式继续工作，转换结果存储于 ADA0CRn 寄存器中，直到扫描操作完成。但是，扫描操作完成后不生成中断信号 INTAD。

(7) 掉电比较阈值寄存器 (ADA0PFT)

<R>

当与 A/D 转换结果寄存器 nH (ADA0CRnH) 比较时，ADA0PFT 寄存器设置阈值。

ADA0PFT 寄存器所设置的 8 位数据与 ADA0CRnH 寄存器的值相比较。

ADA0PFT 寄存器支持 8 位或 1 位的读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFFF205H								
	7	6	5	4	3	2	1	0
ADA0PFT								

11.5 操作

11.5.1 基本操作

- <1> 使用 ADA0M0, ADA0M1, ADA0M2 和 ADA0S 寄存器设置操作模式, 触发模式和执行 A/D 转换的转换时间。当 ADA0M0 寄存器的 ADA0CE 位被设置时, 在软件触发模式下转换开始, 在外部或时间触发模式下, A/D 转换器等待触发。
- <2> 当 A/D 转换开始时, 采样与保持电路对被选入模拟输入通道的输入电压进行采样。
- <3> 当采样和保持电路采集输入信号到指定状态, 电路进入保持状态, 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) 的第 9 位, 以设置比较电压生成 DAC 为 $(1/2) AV_{REF0}$ 。
- <5> 比较电压生成 DAC 和模拟输入电压之间的电压差通过电压比较器比较。如果模拟输入电压高于 $(1/2) AV_{REF0}$, SAR 寄存器的最高有效位(MSB)保留设置。如果模拟输入电压低于 $(1/2) AV_{REF0}$, 则 MSB 复位。
- <6> 接写来, SAR 寄存器的第 8 位自动设置, 并且进入下一个比较过程。根据第 9 位的预置值选择比较电压生成 DAC, 具体描述如下所示。
 - 第 9 位 = 1: $(3/4) AV_{REF0}$
 - 第 9 位 = 0: $(1/4) AV_{REF0}$该比较电压与模拟输入电压进行比较, 根据结果, 操作 SAR 的第 8 位, 如下所示。
模拟输入电压 \geq 比较电压: 第 8 位 = 1
模拟输入电压 \leq 比较电压: 第 8 位 = 0
- <7> 按此方式继续进行比较, 直至 SAR 寄存器的第 0 位。
- <8> 全部 10 位比较完成后, 在 SAR 中保留一个有效的数值结果, 然后将结果传送并存储到 ADA0CRn 寄存器。之后产生 A/D 转换结束中断请求信号(INTAD)。

11.5.2 触发模式

通过设置触发模式确定开始转换操作的时序。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，和外部触发模式。ADA0M0.ADA0TMD 位用于设置触发模式。硬件触发模式由 ADA0M2.ADA0TMD1 和 ADA0M2.ADA0TMD0 位设置。

(1) 软件触发模式

当 ADA0M0.ADA0CE 位被设置为 1，由 ADA0S 寄存器指定的模拟输入引脚 (ANI0 ~ ANI15) 的信号被转换。当转换完成，结果存储于 ADA0CRn 寄存器中。同时，产生 A/D 转换结束中断请求信号 (INTAD)。

如果由 ADA0M0.ADA0MD1 和 ADA0M0.ADA0MD0 位指定的操作模式是连续选择/扫描模式，下一次扫描开始，除非第一次转换完成后将 ADA0CE 位清零。执行一次转换，并且当操作模式为单选择/扫描模式时停止。

当转换开始，ADA0M0.ADA0EF 位被设置为 1（表明转换正在进行）。

如果在转换期间 ADA0M0，ADA0M2，ADA0S，ADA0PFM，或 ADA0PFT 寄存器进行写操作，则转换中止并从开始处重新开始。

(2) 外部触发模式

该模式下，当外部触发信号输入到 ADTRG 引脚时，由 ADA0S 寄存器指定的模拟输入引脚 (ANI0 ~ ANI15) 开始转换。通过 ADA0M0.ADA0ETS1 和 ADA0M0.ADA0ETS0 位可以指定被检测的外部触发边沿（如，上升沿，下降沿，或上升沿和下降沿）。当 ADA0CE 位被设置为 1，A/D 转换器等待触发，并且在外部触发被输入后开始转换。

当转换完成时，转换结果存储到 ADA0CRn 寄存器中，无论是否设置 ADA0MD1 和 ADA0MD0 位的连续选择，连续扫描，或单脉冲扫描模式为操作模式。同时，生成中断信号 INTAD，A/D 转换器等待再次触发。

当开始转换时，ADA0EF 位被设置为 1（表明转换正在进行）。然而，当 A/D 转换器等待触发时，ADA0EF 位被清零（表明转换停止）。转换操作期间如果输入有效触发，则转换中止并从开始处重新开始。

转换操作期间，如果 ADA0M0，ADA0M2，ADA0S，ADA0PFM，或 ADA0PFT 寄存器进行写操作，转换不会中止，A/D 转换器等待再次触发。

(3) 定时器触发模式

该模式下，当连接到定时器的捕捉/比较寄存器产生比较相等中断请求信号(INTTP2CC0 或 INTTP2CC1)时，由 ADA0S 寄存器指定的模拟输入引脚(ANI0 ~ ANI15)的信号开始转换。使用 ADA0TMD1 和 ADA0TMD0 位选择 INTTP2CC0 或 INTTP2CC1 信号，在指定比较相等中断请求信号的上升沿，转换开始。当 ADA0CE 位被设置为 1，A/D 转换器等待触发，并在定时器的比较相等中断请求信号输入时开始转换。

当转换完成时，无论是否设置 ADA0MD1 和 ADA0MD0 位的连续选择，连续扫描，或单脉冲扫描模式为操作模式，转换结果存储于 ADA0CRn 寄存器中。同时生成中断信号 INTAD，A/D 转换器等待再次触发。

当转换开始时，ADA0EF 位被设置为 1（表明转换正在进行）。然而，当 A/D 转换器等待触发时，ADA0EF 位被清零（表明转换停止）。转换操作期间，如果输入有效触发，则转换中止并从开始处重新开始。

转换期间，如果 ADA0M0，ADA0M2，ADA0S，ADA0PFM，或 ADA0PFT 寄存器进行写操作，则转换停止且 A/D 转换器等待再次触发。

11.5.3 操作模式

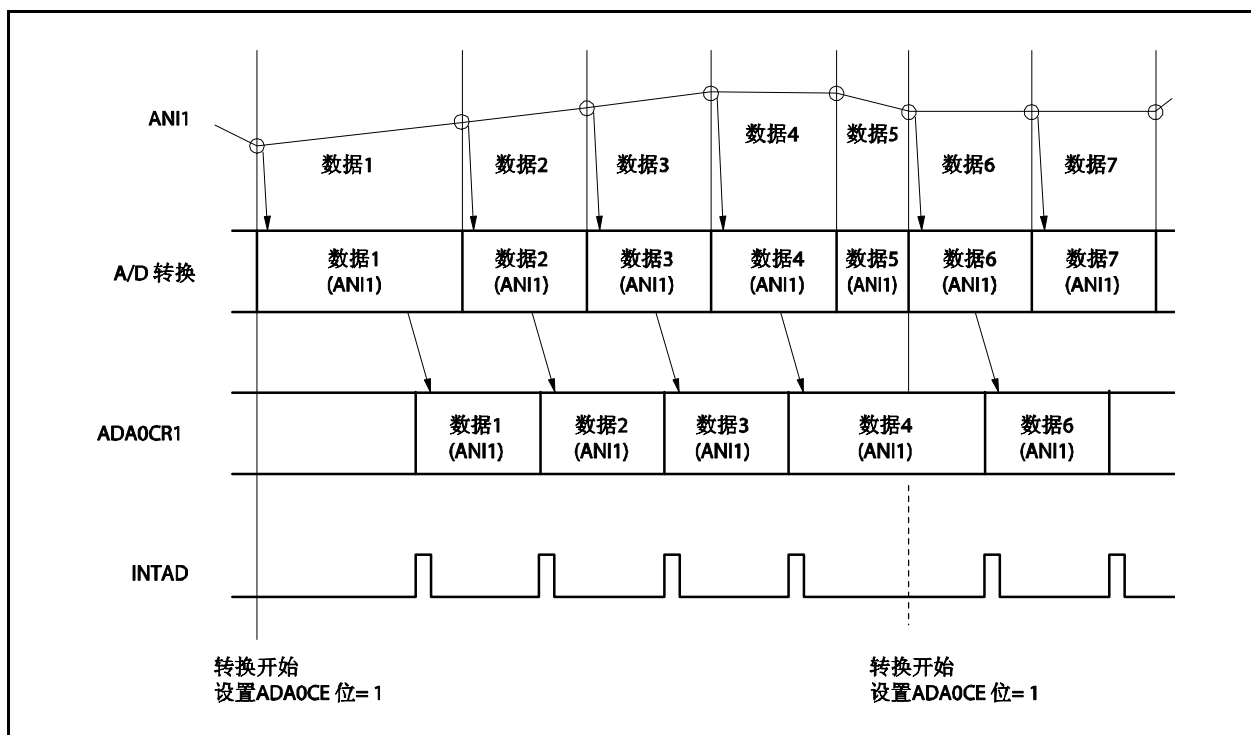
有三种操作模式可用来自设置 ANI0 ~ ANI15 引脚：连续选择模式，连续扫描模式和单脉冲扫描模式。
使用 ADA0M0.ADA0MD1 和 ADA0M0.ADA0MD0 位来选择操作模式。

(1) 连续选择模式

该模式下，由 ADA0S 寄存器选择的某个模拟输入引脚的电压连续转换成数字量。

与模拟输入引脚对应的转换结果存储于 ADA0CRn 寄存器中。该模式下，模拟输入引脚与 ADA0CRn 寄存器一一对应。每次 A/D 转换完成，产生 A/D 转换结束中断请求信号 (INTAD)。转换完成后，除非 ADA0M0.ADA0CE 位被清零 (n = 0 ~ 15)，否则下一次转换开始。

图 11-3. 连续选择模式的操作时序示例 (ADA0S 寄存器 = 01H)

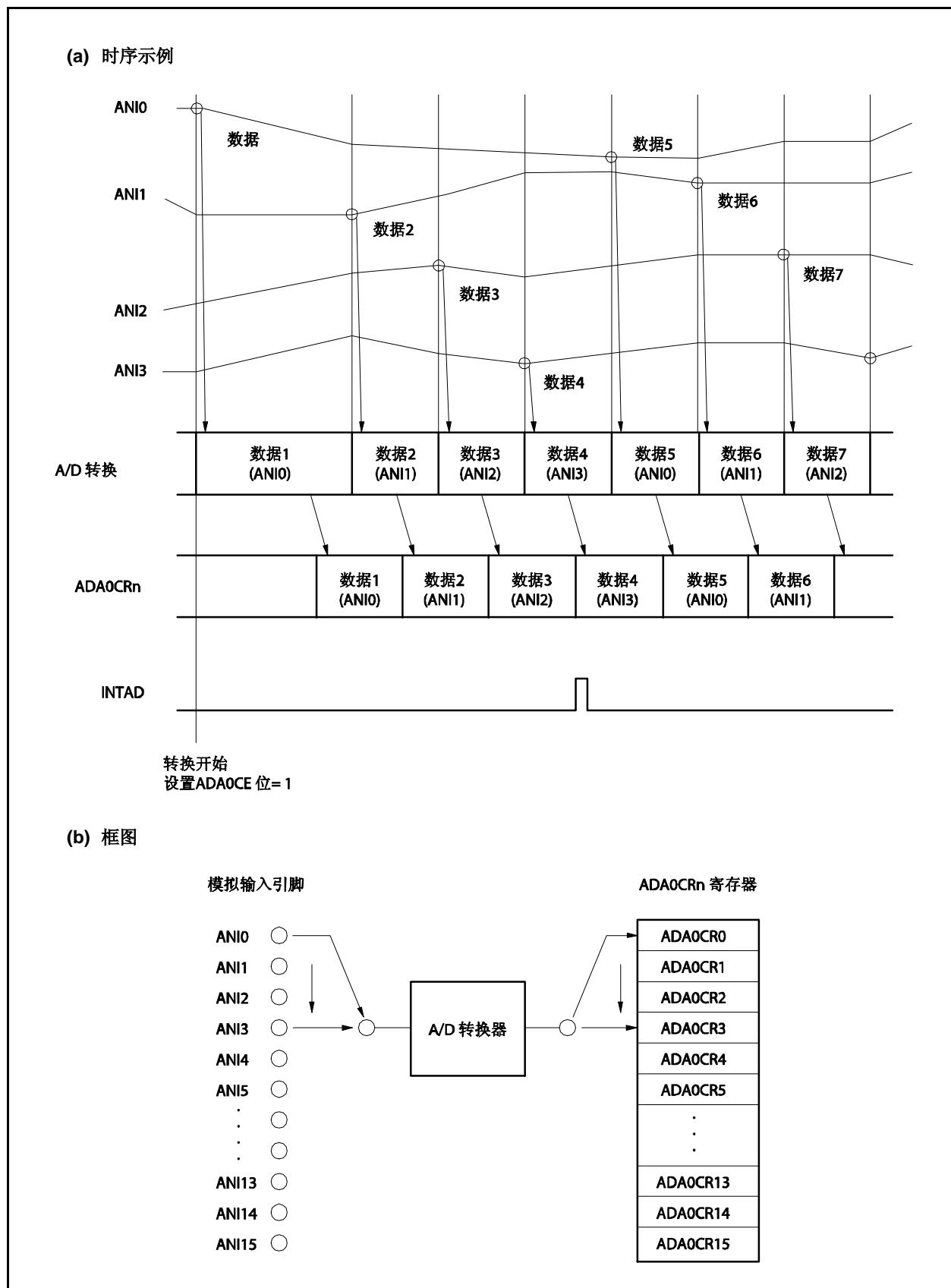


(2) 连续扫描模式

该模式下，模拟输入引脚被依次选择（从 ANI0 引脚到 ADA0S 寄存器所指定的引脚），并将其值转换成数字量。

每次转换的结果存储于与模拟输入引脚对应的 ADA0CRn 寄存器中。当由 ADA0S 寄存器指定的模拟输入引脚转换完成时，产生 INTAD 信号，除非 ADA0CE 位被清零 (n = 0 ~ 15)，否则 A/D 转换从 ANI0 引脚再次开始。

图 11-4. 连续扫描模式的操作时序示例 (ADA0S 寄存器 = 03H)

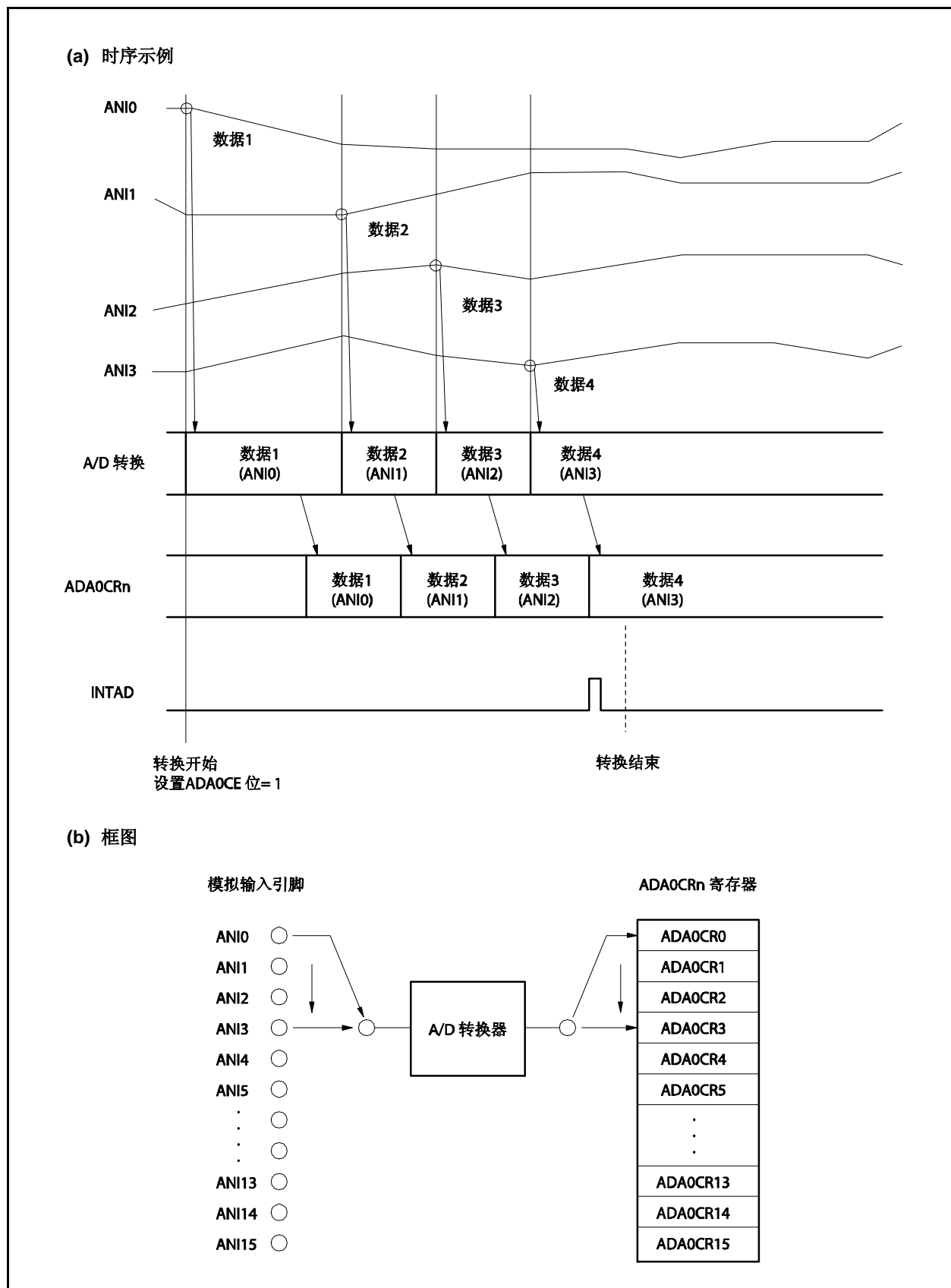


(3) 单脉冲扫描模式

该模式下，模拟输入引脚被依次选择（从 ANI0 引脚到 ADA0S 寄存器所指定的引脚），并将其值转换成数字量。

每次的转换结果存储于与模拟输入引脚对应的 ADA0CRn 寄存器中。当由 ADA0S 寄存器指定的模拟输入引脚转换完成时，产生 INTAD 信号，A/D 转换完成后(n = 0 ~ 15)，转换操作停止。

图 11-5. 单脉冲扫描模式的操作时序示例(ADA0S 寄存器 = 03H)



11.5.4 掉电比较模式

通过 ADA0PFM 和 ADA0PFT 寄存器可以控制 A/D 转换结束中断请求信号 (INTAD)，如下所示。

- 当 ADA0PFM.ADA0PFE 位 = 0，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 且当 ADA0PFM.ADA0PFC 位 = 0，ADA0CRnH 寄存器的值与 ADA0PFT 寄存器的值进行比较。当转换完成时，仅当 $ADA0CRnH \geq ADA0PFT$ 时产生 INTAD 信号。
- 当 ADA0PFE 位 = 1 且当 ADA0PFC 位 = 1，ADA0CRnH 寄存器的值与 ADA0PFT 寄存器的值进行比较。当转换完成时，仅当 $ADA0CRnH < ADA0PFT$ 时产生 INTAD 信号。

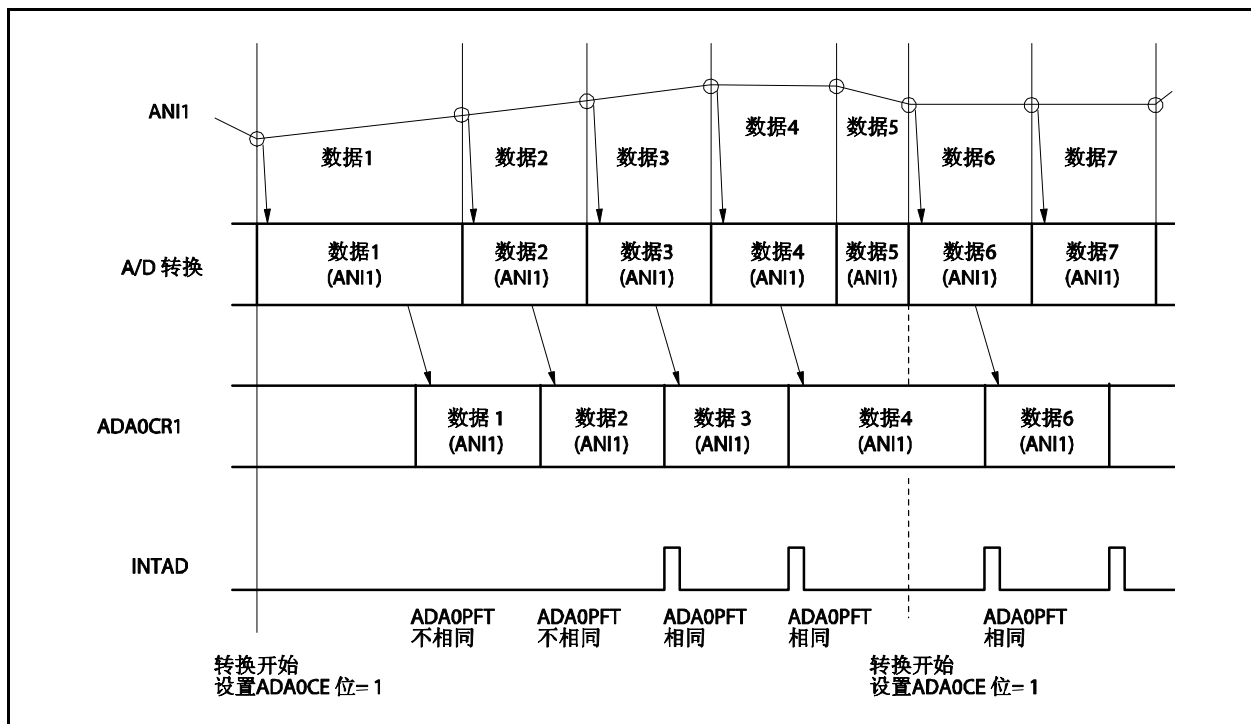
备注 n = 0 ~ 15

掉电比较模式下，有三种操作模式可用来设置 ANI0 ~ ANI15 引脚：连续选择模式，连续扫描模式和单脉冲扫描模式。

(1) 连续选择模式

该模式下，由 ADA0S 寄存器指定的模拟输入引脚的转换电压结果与 ADA0PFT 寄存器的设置值比较。如果掉电比较的结果与 ADA0PFC 位设置的条件匹配，转换结果存储于 ADA0CRn 寄存器，并产生 INTAD 信号。如果不匹配，转换结果存储于 ADA0CRn 寄存器，不产生 INTAD 信号。第一次转换完成后，除非 ADA0M0.ADA0CE 位被清零 (n = 0 ~ 15)，否则下一次转换开始。

图 11-6. 连续选择模式的操作时序示例
(掉电比较模式下: ADA0S 寄存器 = 01H)

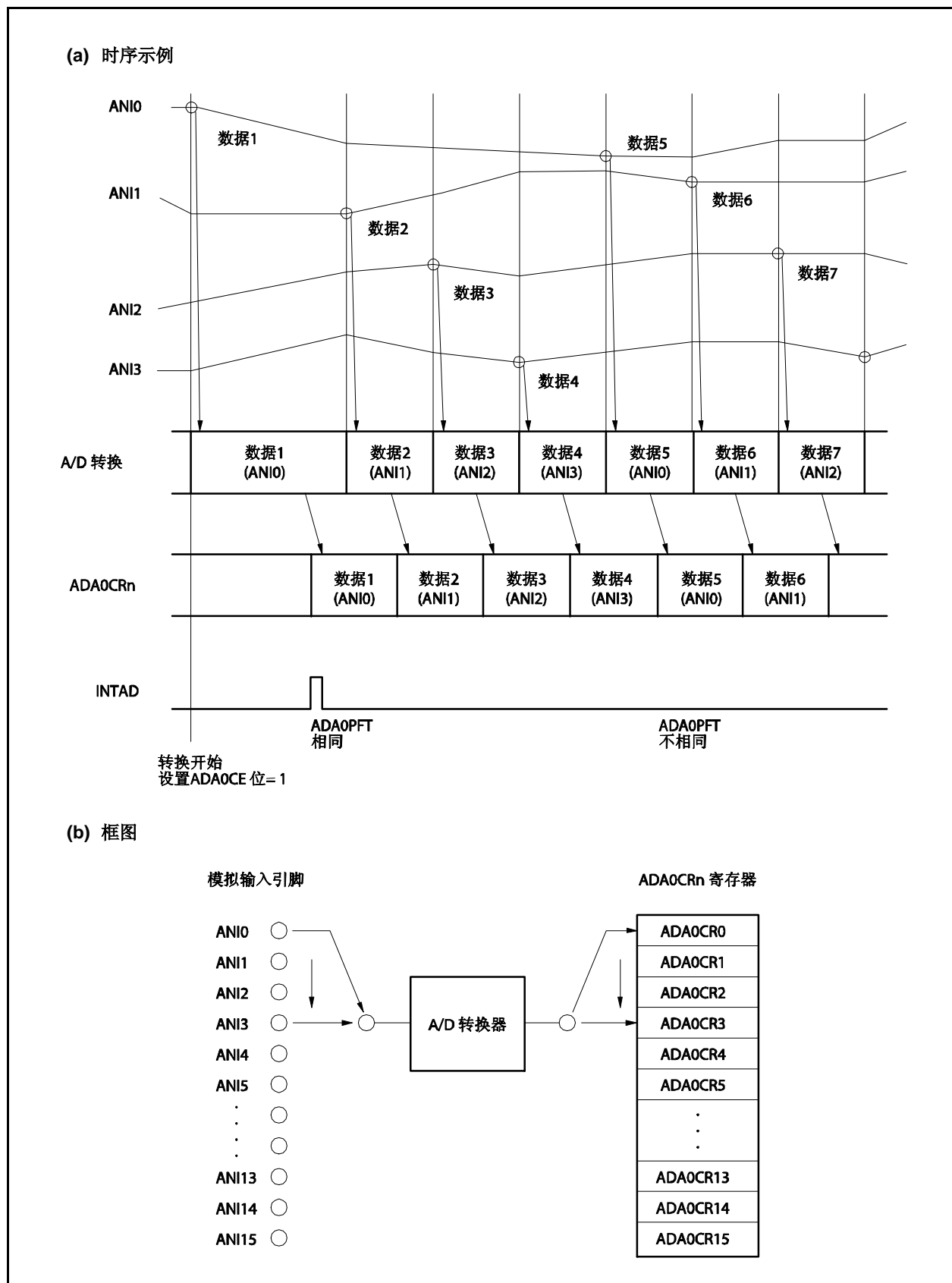


(2) 连续扫描模式

该模式下，模拟输入引脚被依次选择（从 ANI0 引脚到 ADA0S 寄存器指定的引脚），模拟输入引脚的电压转换结果被存储，ADA0CR0H 寄存器 0 通道的设置与 ADA0PFT 寄存器的值比较。如果掉电比较的结果与由 ADA0PFC 位设置的条件匹配，转换结果存储于 ADA0CR0 寄存器中，并产生 INTAD 信号。如果不匹配，转换结果存储于 ADA0CR0 寄存器中，不产生 INTAD 信号。

第一次转换结果存储于 ADA0CR0 寄存器后，模拟输入引脚到 ADA0S 寄存器指定引脚的电压连续转换结果连续被存储。转换完成后，除非 ADA0CE 位被清零，否则下一次转换从 ANI0 引脚再次开始。

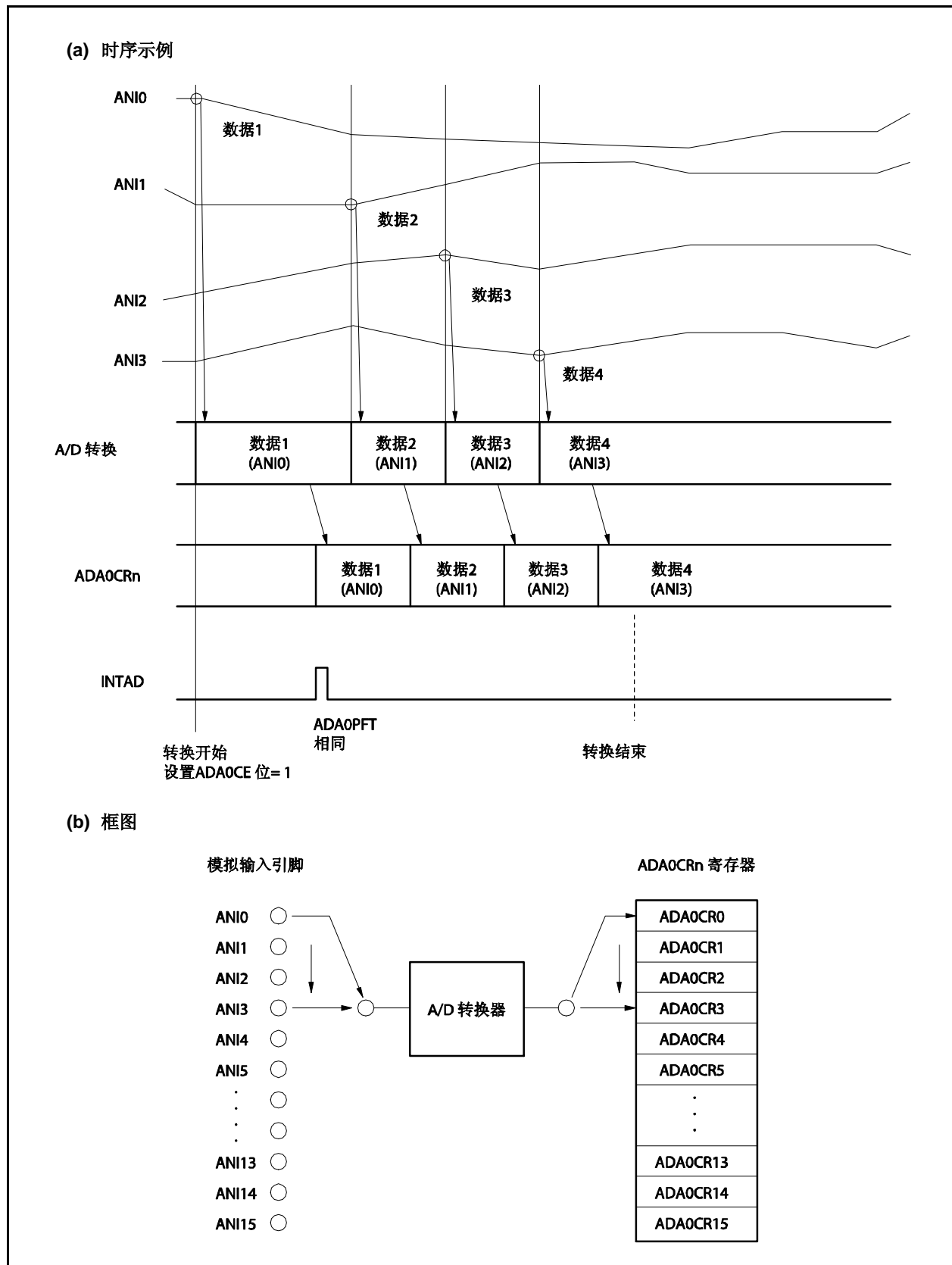
图 11-7. 连续扫描模式的操作时序示例
(掉电比较模式下: ADA0S 寄存器 = 03H)



(3) 单脉冲扫描模式

该模式下，模拟输入引脚被依次选择（从 ANI0 引脚到 ADA0S 寄存器指定的引脚），模拟输入引脚的电压转换结果被存储，ADA0CR0H 寄存器 0 通道的设置值与 ADA0PFT 寄存器的设置值比较。如果掉电比较的结果与由 ADA0PFC 位设置的条件匹配，转换结果存储于 ADA0CR0 寄存器并产生 INTAD 信号。如果不匹配，转换结果存储于 ADA0CR0 寄存器，不产生 INTAD0 信号。第一次转换结果存储于 ADA0CR0 寄存器之后，由 ADA0S 寄存器指定的模拟输入引脚的转换信号结果连续被存储。A/D 转换完成后，转换操作停止。

图 11-8. 单脉冲扫描模式的操作时序示例
(掉电比较模式下: ADA0S 寄存器 = 03H)



11.6 注意事项

(1) 不使用 A/D 转换器

当不使用 A/D 转换器时，通过将 ADA0M0.ADA0CE 清零减少功耗。

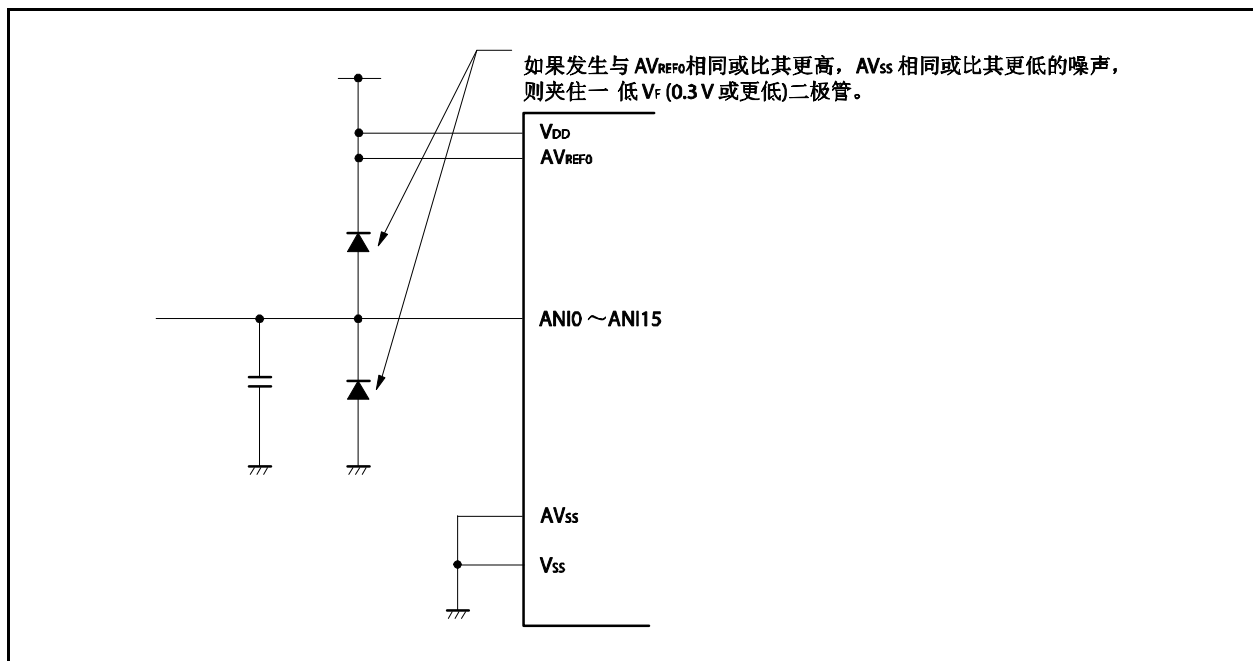
(2) ANI0 ~ ANI15 引脚的输入范围

输入指定范围的电压值到 ANI0 ~ ANI15 引脚。如果大于等于 AV_{REF0} 或小于等于 AV_{SS} （即使在最大绝对值范围内）的电压被输入到任何一个引脚中，则那个通道的转换值不确定，其它通道的转换值也会受到影响。

(3) 抑制噪声的方法

为了确保 10 位分辨率，ANI0 ~ ANI15 引脚必须有效地抑制噪声。噪声影响随着模拟输入源的输出阻抗的增加而提高。为了降低噪声，推荐连接一个外部电容，如图 11-9 所示。

图 11-9. 模拟输入引脚的处理



(4) I/O 复用

模拟输入引脚 (ANI0 ~ ANI15) 可功能复用为端口引脚。当从 ANI0 ~ ANI15 引脚中选择一个执行 A/D 转换，不要在转换期间执行读取输入端口或写入输出端口指令，否则转换分辨率将下降。

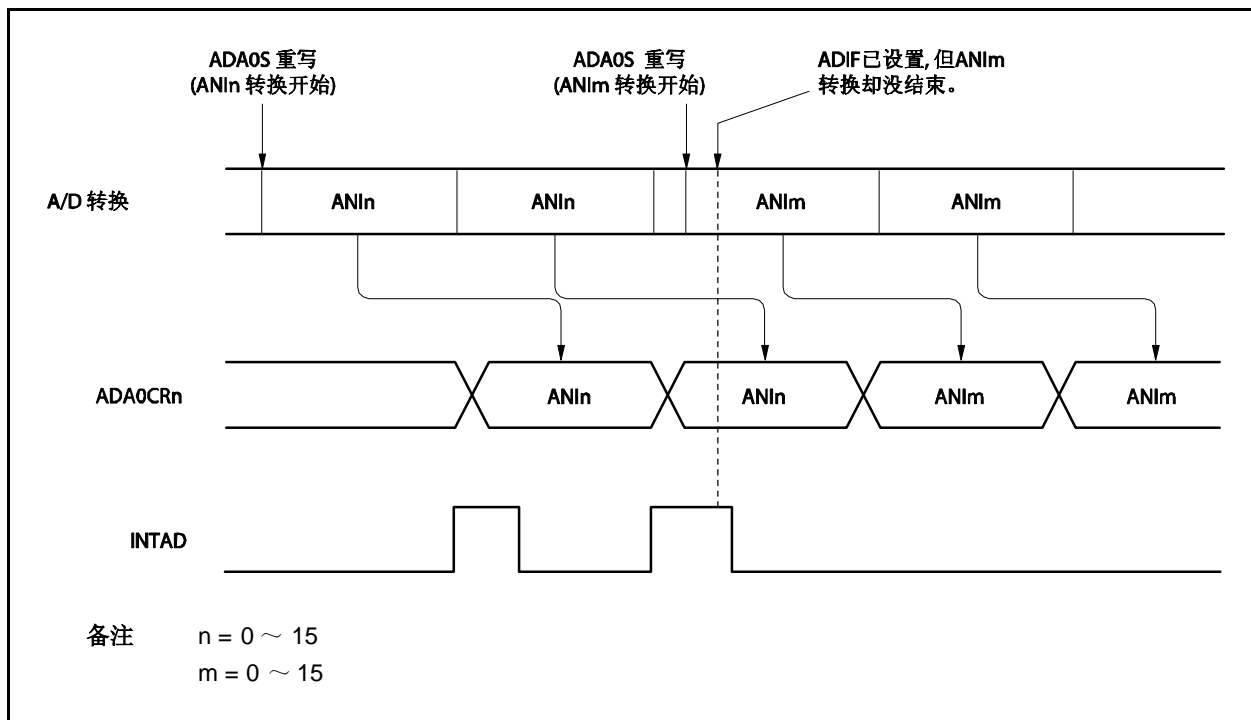
A/D 转换期间将引脚设置为输出端口引脚同样会使转换分辨率下降。其原因是连接端口引脚的外部电路影响电流流向。

如果某引脚的输入信号正在进行 A/D 转换，其临近引脚传输一个数字脉冲，则由于耦合噪声的影响，A/D 转换值可能无法同预期一样。因此，在 A/D 转换过程中，确保不使用临近引脚传输脉冲信号。

(5) 中断请求标志 (ADIF)

即使 ADA0S 寄存器的内容改变，中断请求标志 (ADIF) 也不清零。因此，A/D 转换期间模拟输入引脚改变，上次所选择的模拟输入信号的转换结果可能被存储并且转换结束中断请求标志在 ADA0S 寄存器被改写前立即被设置。如果在 ADA0S 寄存器被改写后立即读取 ADIF 标志，即使新选择的模拟输入引脚的 A/D 转换还未完成，ADIF 标志也可能被设置。当 A/D 转换停止时，在再次转换前将 ADIF 标志清零。

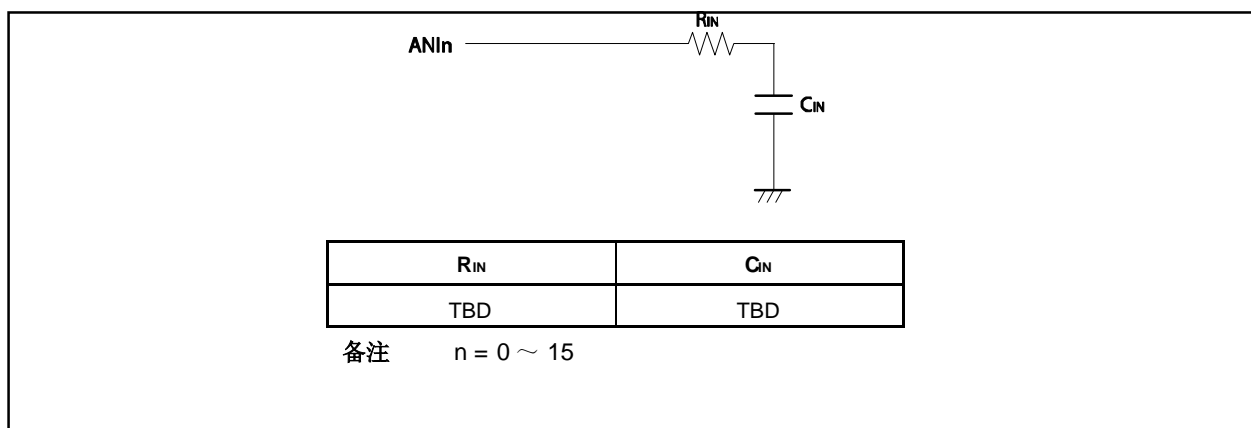
图 11-10. A/D 转换 结束中断请求的产生时序



(6) 内部等效电路

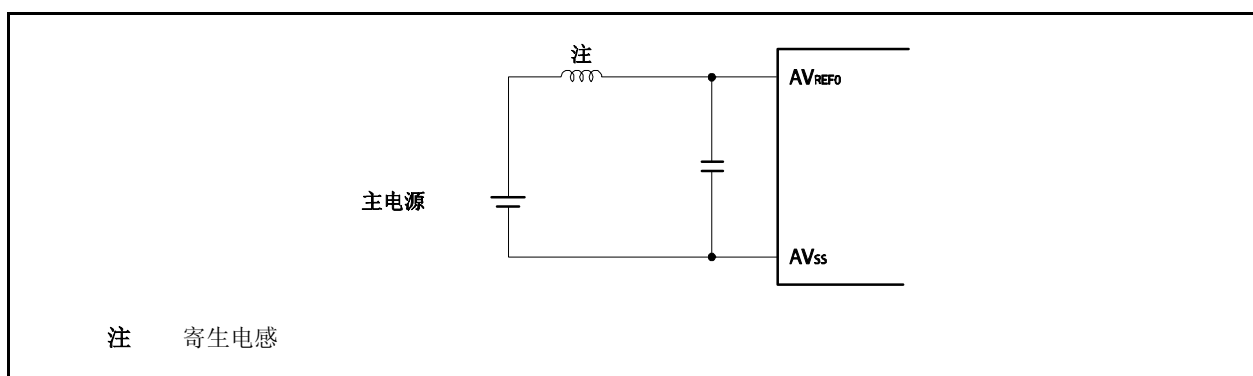
如下显示了模拟输入模块的等效电路。

图 11-11. ANIn 引脚的内部等效电路



(7) AV_{REF0} 引脚

- (a) AV_{REF0} 引脚用于 A/D 转换器的电源引脚，也用于为复用功能端口供电。如图 11-12 所示，当用于备用电源时，确保向 AV_{REF0} 引脚提供与 V_{DD} 相同的电压。
- (b) 如果提供给 AV_{REF0} 引脚的源电源具有高阻抗，或者如果电源低电流负载能力低，参考电压可能被转换中的电流影响（特别是，当转换操作使能位 ADA0CE 刚被设置为 1 之后）。结果，转换精度会下降。如图 11-12 所示，为了避免上述情况，推荐连接一个电容在 AV_{REF0} 和 AV_{SS} 引脚之间以抑制参考电压的波动。
- (c) 如果提供给 AV_{REF0} 引脚的源电源具有高直流阻抗（例如，由于插入一个二极管），当转换使能时的电压可能比转换停止时的电压低，因为 A/D 转换电流引起电压失真。

图 11-12. AV_{REF0} 引脚处理过程的示例

<R>

(8) 读取 ADA0CRn 结果

当 ADA0M0 ~ ADA0M2, ADA0S, ADA0PFM, 或 ADAP0FT 寄存器进行写操作时，ADA0CRn 寄存器的内容可能不确定。转换完成后，并且在写入 ADA0M0 ~ ADA0M2, ADA0S, ADA0PFM 和 ADA0PFT 寄存器之前，读取转换结果。同时，当外部/定时器触发被识别时，ADA0CRn 寄存器的内容可能不确定。转换完成后，并且在外部/定时器触发被识别前，读取转换结果。在时序不同于如上所述时，正确的转换结果可能不会被读取。

(9) A/D 转换结果

如果在模拟输入引脚和参考电压输入引脚中存在噪声，则噪声可能会造成非法的转换结果。为了避免非法转换结果对系统造成的负面影响，需要使用软件处理。下面显示了软件处理的一个例子。

- 取几次 A/D 转换的平均结果并将其作为 A/D 转换的结果。
- 连续进行几次 A/D 转换，去掉可能得到的异常结果。
- 如果一个 A/D 转换的结果被判定造成了系统的故障，在进行故障处理前必须重新检查系统故障。

<R> (10) 待机模式

因为 A/D 转换器在 STOP 模式下停止操作，转换结果无效，所以功耗可以降低。释放 STOP 模式后，操作恢复，但 STOP 模式释放后的 A/D 转换结果无效。当 STOP 模式释放后使用 A/D 转换器，应在设置 STOP 模式前或释放 STOP 模式后，将 ADA0M0.ADA0CE 位清零，然后在释放 STOP 模式后设置 ADA0CE 位为 1。

在 IDLE1, IDLE2, 或副时钟操作模式下，继续操作。因此，为了减小功耗，应将 ADA0M0.ADA0CE 位清零。在 IDLE1 和 IDLE2 模式下，由于模拟输入电压值不能保留，A/D 转换结果在 IDLE1 和 IDLE2 模式释放后无效。在 IDLE1 和 IDLE2 模式被设置前，转换结果有效。

<R> (11) 稳定时间内寄存器和触发器输入的改写

禁止在稳定时间内改写 ADA0M0, ADA0M2, ADA0S, ADA0PFM 及 ADA0PFT 寄存器和触发器输入。

(12) A/D 转换结果的漂移

A/D 转换的结果可能会根据电压的波动而发生变化，或者可能受到噪声影响。为了减小漂移，使用多次测量求平均值的方法。

(13) A/D 转换转换结果滞后性

逐次逼近 A/D 转换器维持内部采样与保持电容的模拟输入电压，同时进行 A/D 转换。A/D 转换完成后模拟输入电压保留在内部采样与保持电容中。结果产生以下现象。

- 当同一个通道用于 A/D 转换，如果电压高于或低于先前 A/D 转换，则出现滞后性，转换结果受到上次转换值影响。因此，即使在相同电压上进行转换也会产生不同结果。
- 当变换模拟输入通道时，可能出现滞后性，转换结果受到上次转换通道影响。这是因为 A/D 转换器用于 A/D 转换。因此，即使在相同电压上进行转换也会产生不同结果。

<R> 因此，为了获得更精确的转换结果，在相同的通道连续执行两次 A/D 转换，并且丢弃第一次的转换结果。

11.7 A/D 转换器特征表的阅读方法

以下介绍 A/D 转换器中的专用术语。

(1) 分辨率

这是可识别的最小的模拟输入电压，即每位数字输出的模拟输入电压的百分比，称为 1 LSB（最低有效位）。对于满度的 1 LSB 的百分比用%FSR（满度范围）表示。%FSR 是以百分比形式表示的一定范围的可变模拟输入电压的比率。与分辨率无关，可如下表示。

$$\begin{aligned} 1\%FSR &= (\text{可变模拟输入电压的最大值} - \text{可变模拟输入电压的最小值}) / 100 \\ &= (AV_{REF0} - 0) / 100 \\ &= AV_{REF0} / 100 \end{aligned}$$

当分辨率为 10 位时 1LSB 表示如下：

$$\begin{aligned} 1 \text{ LSB} &= 1/2^{10} = 1/1,024 \\ &= 0.098\%FSR \end{aligned}$$

精确度与分辨率无关，而由总误差决定。

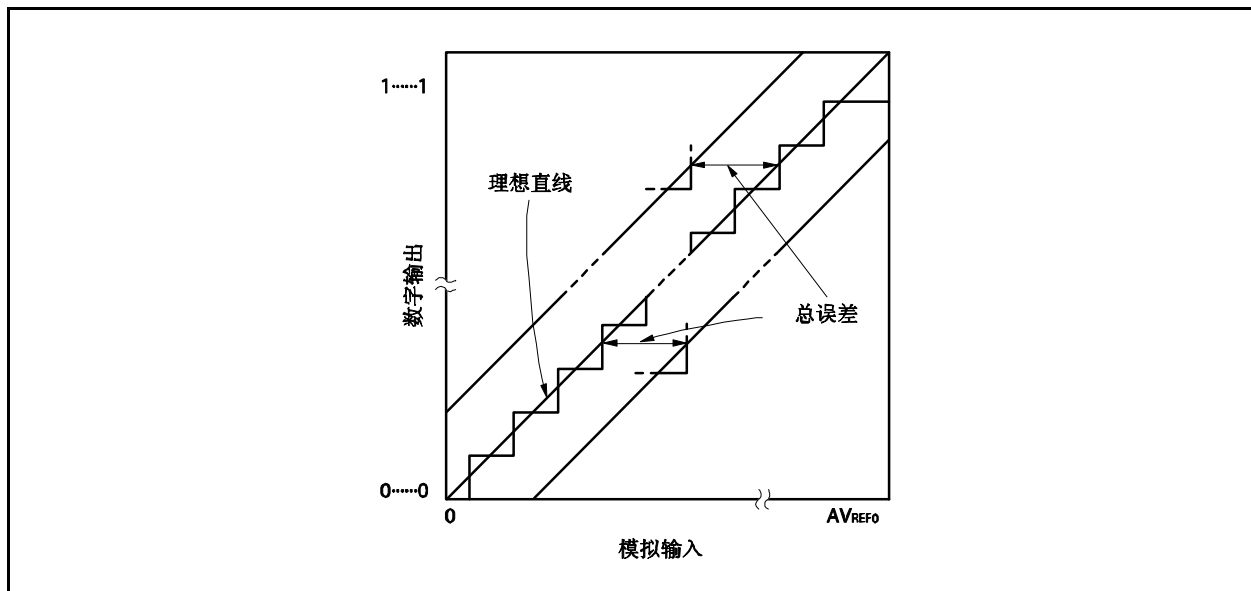
(2) 总误差

总误差是指实际测量值与理论值之间的最大误差。

零度误差、满度误差、积分线性误差和积分非线性误差等组合起来表示总误差。

量化误差不属于特征表中总误差的范围。

图 11-13. 总误差

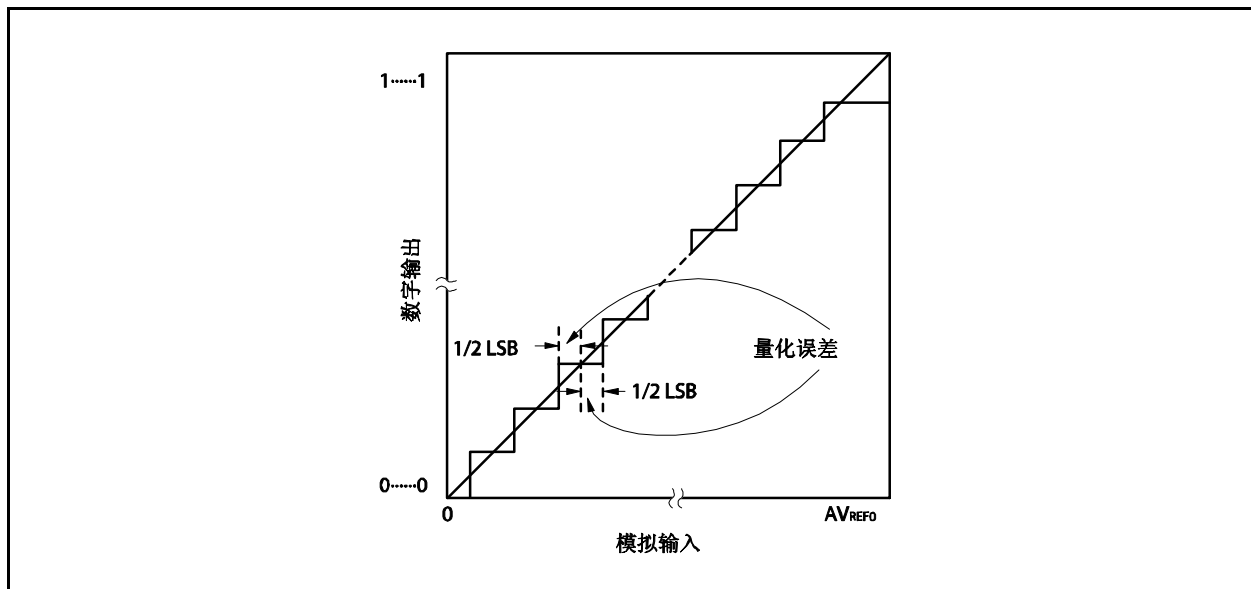


(3) 量化误差

当模拟值转换成数字值时，通常会产生 $\pm 1/2\text{LSB}$ 的误差。在一个 A/D 转换器中，相差 $\pm 1/2\text{LSB}$ 的模拟输入电压被转换成相同的数字代码，因此量化误差不可避免。

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

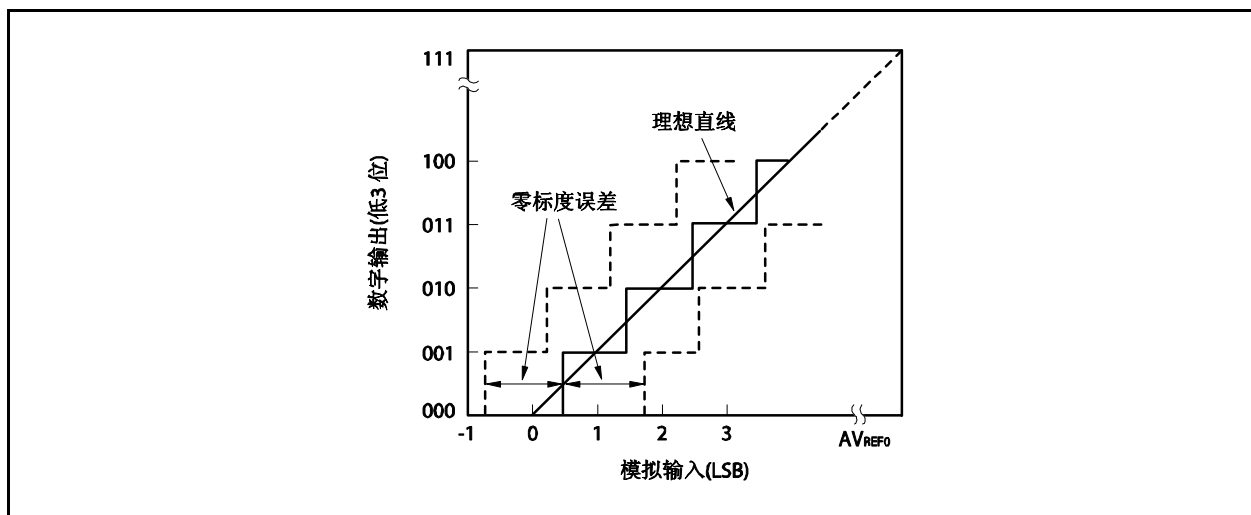
图 11-14. 量化误差



(4) 零度误差

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

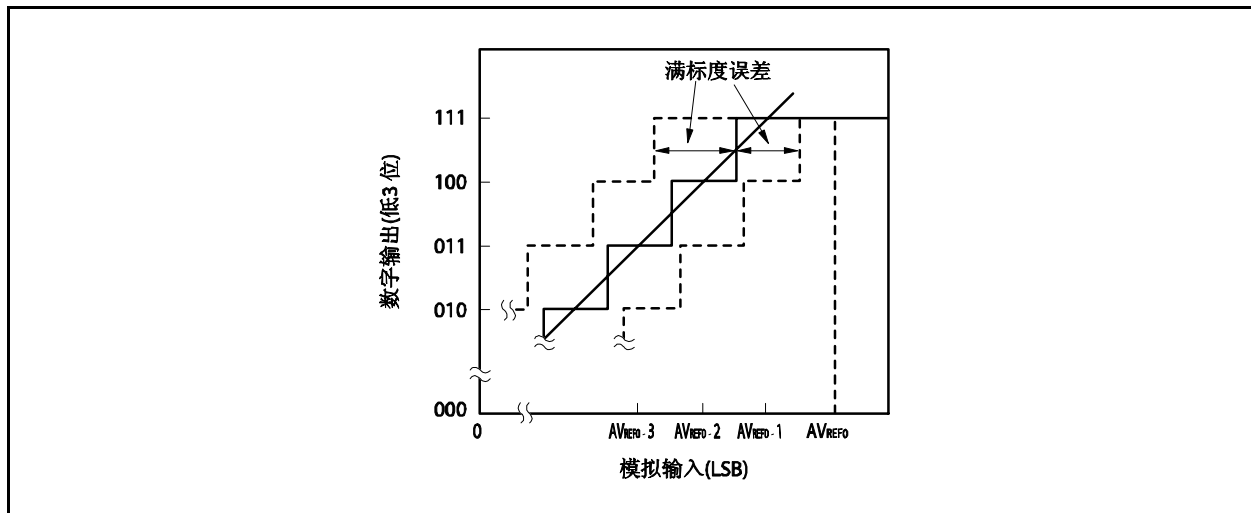
图 11-15. 零度误差



(5) 满度误差

满度误差表示当数字输出范围在 $1 \dots 110 \sim 1 \dots 111$ 之间时模拟输入电压的实际测量值与理论值(满度 - $3/2\text{LSB}$)之间的误差。

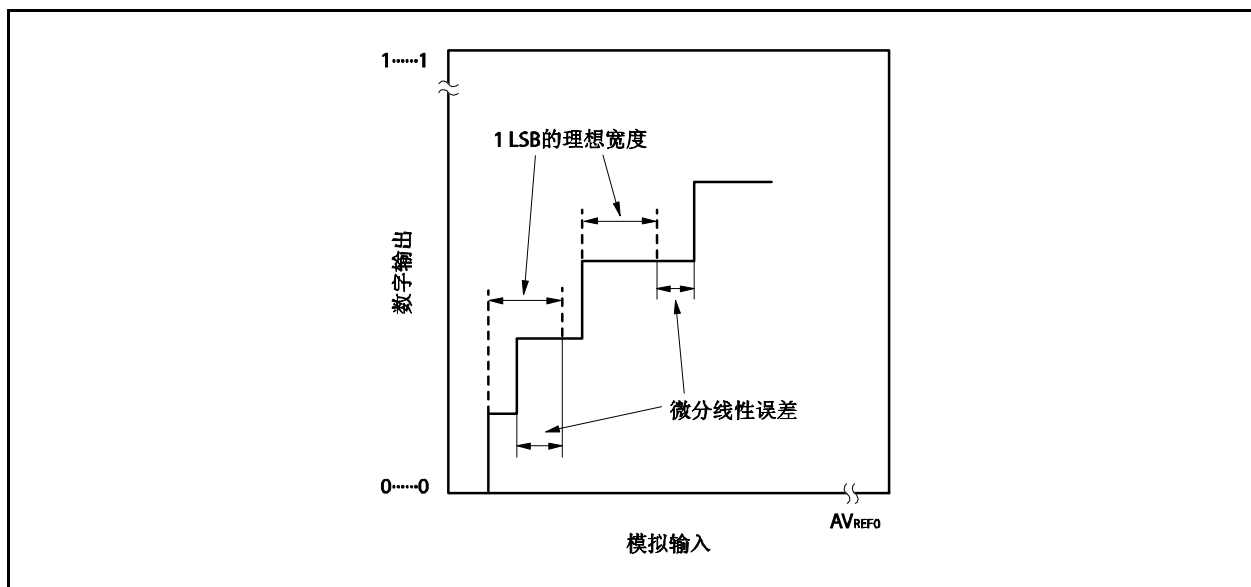
图 11-16. 满度误差



(6) 微分线性误差

理论上，代码输出的宽度为 1LSB 。当输出一个特定码，微分线性误差表示实际测量值与理想值之间的差距。当同一个通道的模拟输入引脚的电压一字节一字节的从 AV_{SS} 到 AV_{REF0} 持续增加，这显示了 A/D 转换的基本特征。当输入电压增加或减小，或两个(包括两个以上)被使用，参见 11.7 (2)总误差。

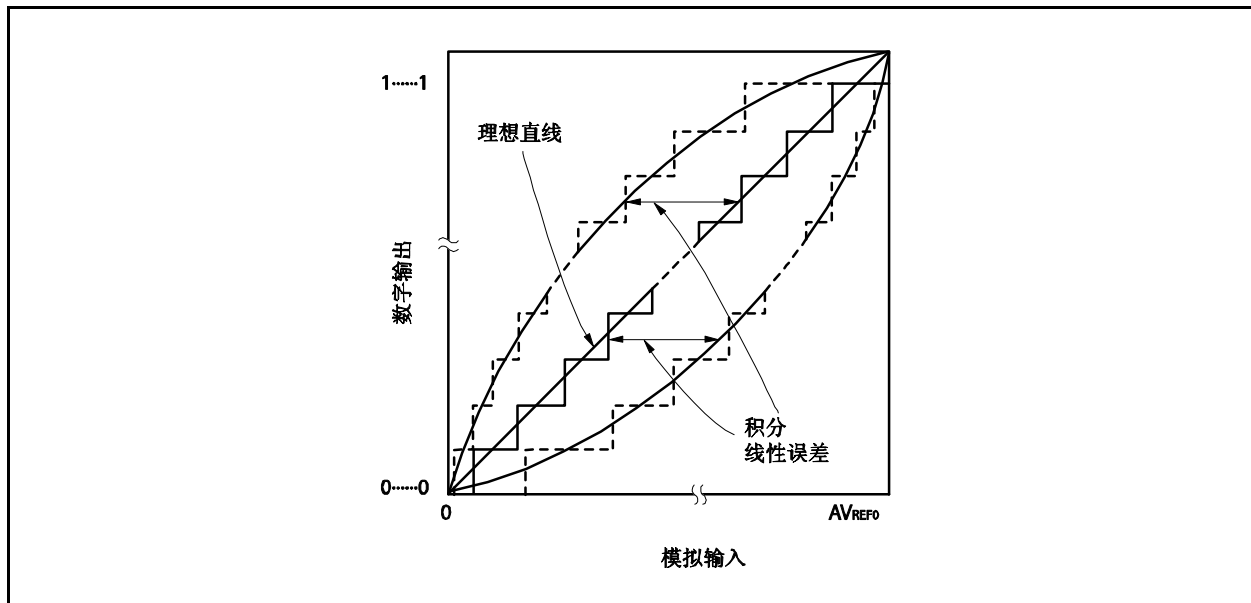
图 11-17. 微分线性误差



(7) 积分线性误差

积分线性误差说明了转换特征偏离理想线性关系的程度。它表示当零度误差和满度误差均为 0 时实际测量值与理想直线之间误差的最大值。

图 11-18. 积分线性误差



(8) 转换时间

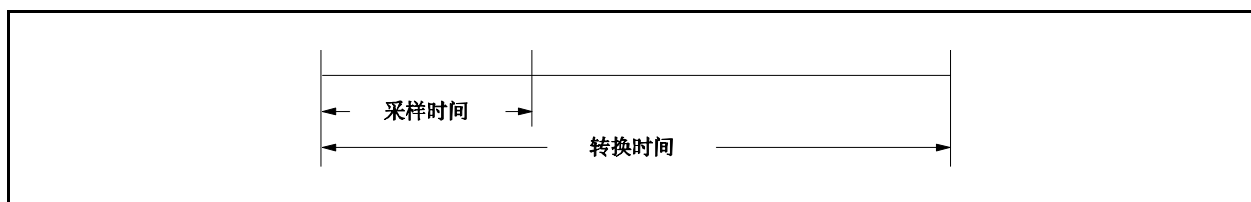
转换时间表示从产生触发到获取数字输出所经历的时间。

采样时间包含在特征表中的转换时间中。

(9) 采样时间

采样时间表示模拟电压的模拟开关被打开到模拟电压被采样与保持电路采样所需的时间。

图 11-19. 采样时间



第十二章 异步串行接口 A (UARTA)

在 V850ES/HG2 包括 3 种异步串行接口 A (UARTA) 的通道。

12.1 特点

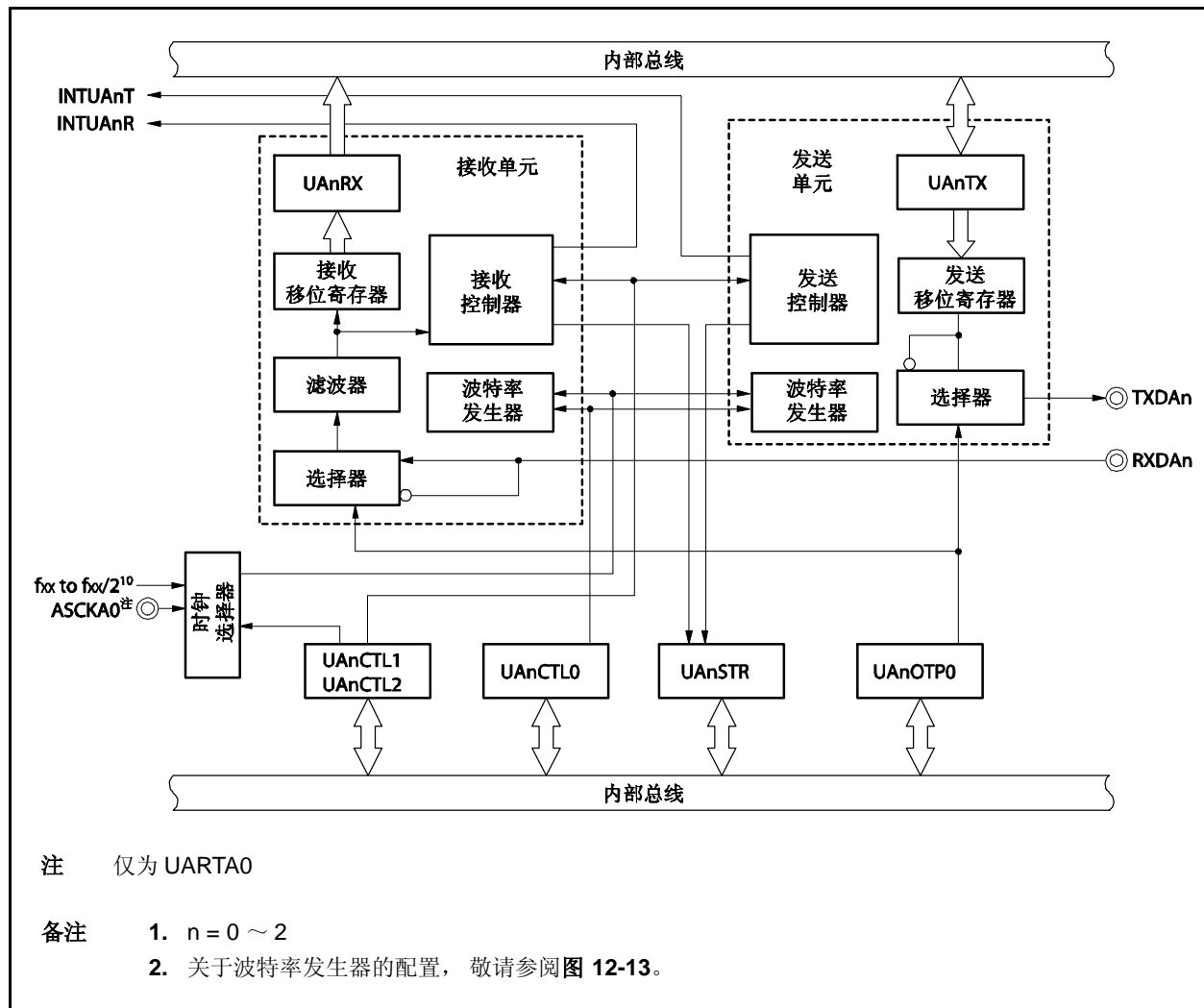
- 发送速度：300 bps ~ 312.5 kbps (使用 20 MHz 的内部系统时钟和专用波特率发生器)
- 全双工通讯：
 - 内部 UARTAn 接收数据寄存器 (UAnRX)
 - 内部 UARTAn 发送数据寄存器 (UAnTX)
- 两引脚配置：
 - TXDAn: 发送数据输出引脚
 - RXDAn: 接收数据输入引脚
- 接收错误输出功能
 - 奇偶校验错误
 - 帧错误
 - 过载错误
- 中断源：2
 - 接收完成中断 (INTUAnR): 在接收使能状态下通过识别三类接收错误生成一个中断。在串行发送完成后从接收移位寄存器发送接收数据到接收数据寄存器时也会生成一个中断。
 - 发送使能中断 (INTUAnT): 此中断在发送使能时能状态下从发送数据寄存器传送数据到发送移位寄存器后发生。
- 字符长度：7, 8 位
- 奇偶校验功能：奇校验、偶校验 0 校验或无校验
- 发送停止位：1, 2 位
- 片上专用波特率发生器
- 可选最高有效位(MSB)优先/最低有效位(LSB)优先发送
- 发送/接收数据逆转输入/输出是可能的
- 以本地互连网络 (LIN) 通讯格式进行的同步中断区 (SBF) 发送/接收是可能的
 - 同步中断区 (SBF) 发送 13~20 位可选
 - 可以为同步中断区 (SBF) 接收识别 11 位或更多位
 - 提供同步中断区 (SBF) 接收标志

备注 n = 0 ~ 2

12.2 配置

如下所示为 UARTAn 的框图。

图 12-1. 异步串行接口 An 的框图



UARTAn 包括以下硬件单元。

表 12-1. UARTAn 的配置

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 发送移位寄存器 UARTAn 发送数据寄存器 (UAnTX)

(1) UARTAn 控制寄存器 0 (UAnCTL0)

UAnCTL0 寄存器是用于指定 UARTAn 操作的 8 位寄存器。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

The UAnCTL1 寄存器是用于选择 UARTAn 输入时钟的 8 位寄存器。

(3) UARTAn 控制寄存器 2 (UAnCTL2)

The UAnCTL2 寄存器是用于控制 UARTAn 波特率的 8 位寄存器。

(4) UARTAn 选项控制寄存器 0 (UAnOPT0)

The UAnOPT0 寄存器是用于控制 UARTAn 串行发送的 8 位寄存器。

(5) UARTAn 状态寄存器 (UAnSTR)

UAnSTRn 该寄存器包含发生接收错误时的错误标志。当发生接收错误时，相应的接收错误标志设置为 1。

(6) UARTAn 接收移位寄存器

该移位寄存器用于将输入到 RXDAn 引脚的串行数据转换为并行数据。当接收到 1 字节的数据并检测到停止位，接收数据被传送到 UAnRX 寄存器。

该寄存器不可以直接操作。

(7) UARTAn 接收数据寄存器 (UAnRX)

UAnRX 寄存器是保持接收数据的 8 位寄存器。当接收到 7 个字符时，0 被存储于最高位（当数据以 LSB 优先模式接收）。

接收使能状态下，接收数据以同步方式在每帧移位过程中，从 UARTAn 接收移位寄存器同步发送到 UAnRX 寄存器。

发送到 UAnRX 寄存器也会引起输出接收完成中断请求信号 (INTUAnR)。

(8) UARTAn 发送移位寄存器

该寄存器是移位寄存器，用于将寄存器 UAnTX 传出的并行数据转换成串行数据。

当 1 字节数据从 UAnTX 传送寄存器，移位寄存器数据从 TXDAn 引脚输出。

该寄存器不可以直接操作。

(9) UARTAn 发送数据寄存器 (UAnTX)

UAnTX 寄存器是一个 8 位的发送数据缓冲器。当发送数据被写入 UAnTX 寄存器时，开始发送。当数据被写入 UAnTX 寄存器（当一帧数据从 UAnTX 寄存器发送到 UARTAn 发送移位寄存器），生成发送使能中断请求信号 (INTUAnT)。

12.3 寄存器

(1) UARTAn 控制寄存器 0 (UAnCTL0)

UAnCTL0 寄存器是用于控制 UARTAn 串行发送操作的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 10H。

(1/2)

复位后: 10H	R/W	地址:	UA0CTL0 FFFFFFFA00H, UA1CTL0 FFFFFFFA10H, UA2CTL0 FFFFFFFA20H					
UAnCTL0 (n = 0 ~ 2)	7	6	5	4	3	2	1	0
	UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL
	UAnPWR	UARTAn 操作控制						
	0	禁用UARTAn 操作(UARTAn 异步复位)						
	1	启用UARTAn 操作						
	UARTAn 操作由 UAnPWR 位控制。通过清零 UAnPWR 位, TXDAn 引脚输出连至高电平 (如果 UAnOPT0.UAnTDL 位 = 1, 则连至低电平)。							
	UAnTXE	启用发送操作						
	0	禁用发送操作						
	1	启用发送操作						
	· 通过设置 UAnPWR 位为 1 然后 设置 UAnTXE 位为 1 来开始发送。 通过清零 UAnTXE 位及 UAnPWR 位来停止发送。 · 通过清零 UAnTXE 位, 等待基址时钟的两周期时间, 然后 设置再次 UAnTXE 位为 1 来初始化发送单元。不然, 初始化操作可能无法执行 (至于基址时钟, 参见 12.6 (1) (a) 基址时钟)。							
	UAnRXE	启用接收操作						
	0	禁用接收操作						
	1	启用接收操作						
	通过设置 UAnPWR 位为 1 然后 设置 UAnTXE 位为 1 来开始接收。 通过清零 UAnTXE 位及 UAnPWR 位来停止接收。 · 通过清零 UAnTXE 位, 等待基址时钟的两周期时间, 然后 设置再次 UAnTXE 位为 1 来初始化接收单元。不然, 初始化操作可能无法执行 (至于基址时钟, 参见 12.6 (1) (a) 基址时钟)。							

<R>

UAnDIR	传输方向选择
0	最高有效位 (MSB) 先传输
1	最低有效位 (LSB) 先传输

• 此寄存器只有在UAnPWR位=0或UAnTXE位= UAnRXE位=0时才能被重新写入。
 • 当以LIN格式进行传输和接收时，将UAnDIR位设为1。

UAnPS1	UAnPS0	在传输时的奇偶校验检查	在接收时的奇偶校验检查
0	0	无奇偶校验输出	无奇偶校验接收
0	1	0校验输出	0校验接收
1	0	奇校验输出	奇校验接收
1	1	偶校验输出	偶校验接收

• 此寄存器只有在UAnPWR位=0或UAnTXE位= UAnRXE位=0时才能被重新写入。
 • 如果在接收时选择“无奇偶校验接收”，则不会进行奇偶校验。
 因此不设置UAnSTR和UAnPE位。
 • 当以LIN格式进行传输和接收时，将UAnPS1和UAnPS0位清为00。

<R>

UAnCL	指定一帧传输/接收数据的数据字符长度
0	7 位
1	8 位

• 此寄存器只有在UAnPWR位=0或UAnTXE位= UAnRXE位=0时才能被重新写入。
 当以LIN格式进行传输和接收时，将UAnCL位设为1。

UAnSL	指定传输数据停止位的长度
0	1 位
1	2 位

此寄存器只有在UAnPWR位=0或UAnTXE位= UAnRXE位=0时才能被重新写入。

备注 关于校验位的更多细节，敬请参阅 12.5.9 校验位的类型和操作。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

关于更多细节，敬请参阅 12.6 (2) UARTAn 控制寄存器 1 (UAnCTL1)。

(3) UARTAn 控制寄存器 2 (UAnCTL2)

关于更多细节，敬请参阅 12.6 (3) UARTAn 控制寄存器 2 (UAnCTL2)。

(4) UARTAn 选项控制寄存器 0 (UAnOPT0)

The UAnOPT0 寄存器是控制 UARTAn 寄存器的串行发送操作的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 14H。

(1/2)

<R>

复位后: 14H		R/W	地址: UA0OPT0 FFFFA03H, UA1OPT0 FFFFA13H, UA2OPT0 FFFFA23H					
UAnOPT0 (n = 0 ~ 2)	7	6	5	4	3	2	1	0
	UAnSRF	UAnSRT	UAnSTT	UAnSLS2	UAnSLS1	UAnSLS0	UAnTDL	UAnRDL
	UAnSRF	SBF 接收标志						
	0	当设置UAnCTL0.UAnPWR位=UAnCTL0.UAnRXE位=0时。还有在SBF接收正常结束时。						
1	SBF接收时。							
•SBF接收在LIN通讯过程中判断。 •当出现SBF接收错误时，UAnSRF保持在1，然后SBF接收开始。 •UAnSRF是一个只读位。								
UAnSRT	SBF接收触发器							
0	-							
1	SBF接收触发器							
•此为LIN通讯过程中的SBF接收触发器位，当读取时，总是读取“0”。对于同步间隔场接收，将UAnSRT设为1以启用SBF接收。 •在设置UAnPWR位=UAnRXE位=1后设置UAnSRT位。								
UAnSTT	同步 SBF 传输触发器							
0	-							
1	同步 SBF 传输触发器							
•此为LIN通讯过程中的同步 SBF 传输触发器位，当读取时，总是读取 0。 •在设置UAnPWR位=UAnTXE位=1后设置UAnSTT位。								
注意事项 在 SBF 接收(UAnSRF 位= 1)期间，不要设置设置 UAnSRT 和 UAnSTT 位（为 1）。								

UAnSLS2	UAnSLS1	UAnSLS0	同步 SBF 传输长度选择
1	0	1	13位 输出(访问值)
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UAnTXE位=0时，可设置该寄存器。

UAnTDL	传输数据电平位
0	传输数据的正常输出
1	传输数据的倒转输出

•TXDAn引脚的输出电平可以通过UAnTDL位倒转。
•当UAnPWR位=0或UAnTXE位=0时可设置此寄存器。

UAnRDL	接收数据电平位
0	接收数据的正常输入
1	接收数据的倒转输入

•TXDAn引脚的输入电平可以通过UAnRDL位倒转。
•当UAnPWR位=0或UAnRXE位=0时可设置该寄存器。

(5) UARTAn 状态寄存器 (UAnSTR)

UAnSTR 寄存器是显示 UARTAn 发送状态和接收错误内容的 8 位寄存器。

该寄存器支持 8 位或 1 位读写方式，但是当 UAnPE，UAnFE 和 UAnOVE 位可同时读取时，UAnTSF 位为只读位。然而，这些位只可通过写入 0 清零；不可通过写入 1 设置（即使写入 1，该值仍然保持）。初始化条件显示如下。

寄存器/位	初始化条件
UAnSTR 寄存器	<ul style="list-style-type: none"> • 复位 • UAnCTL0.UAnPWR = 0
UAnTSF 位	<ul style="list-style-type: none"> • UAnCTL0.UAnTXE = 0
UAnPE, UAnFE, UAnOVE 位	<ul style="list-style-type: none"> • 写入 0 • UAnCTL0.UAnRXE = 0

复位后: 00H R/W 地址: UA0STR FFFFA04H, UA1STR FFFFA14H,
UA2STR FFFFA24H

UAnSTR (n=0~2)	7	6	5	4	3	2	1	0
	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

UAnTSF	传输状态标记
0	<ul style="list-style-type: none"> •当已经设置UAnPWR位= 0或UAnTXE位=0时。 •在传输完成后没有下一个数据从UAnTX寄存器传出时。
1	写入 UAnTX 寄存器
<p>当进行连续传输时，UAnTSF位始终为1。 在初始化传输单元时，在开始进行初始化前检查UAnTSF位是否等于0。 当UAnTSF位=1时进行初始化将无法保证数据的传输。</p>	

UAnPE	奇偶校验错误标记
0	<ul style="list-style-type: none"> •当已经设置UAnPWR位= 0或UAnTXE位=0时。 •当已经写入0时。
1	在接收过程中数据的奇偶性与奇偶校验位不匹配。
<p>•UAnPE位的操作是通过设置UAnCTL0. UAnPS1和UAnCTL0. UAnPS0位控制的。 •UAnPE位可以被读取和写入，但只能通过写入0才能清除而不能通过写入1来设置。 当写入1到UAnPE位时，该值被保留。</p>	

UAnFE	成帧误差标记
0	<ul style="list-style-type: none"> •当已经设置UAnPWR位= 0或UAnRXE位=0时。 •当已经写入0时。
1	在接收过程中没有检测到停止位时。
<p>•只检查接收数据停止位的第一位，而不论UAnCTL0. UAnSL位的值如何。 •UAnFE位可以被读取和写入，但只能通过写入0才能清除而不能通过写入1来设置。 当写入1到UAnFE位时，该值被保留。</p>	

UAnOVE	超限误差标记
0	<ul style="list-style-type: none"> •当已经设置UAnPWR位= 0或UAnRXE位=0时。 •当已经写入0时。
1	当已经设置接收数据到UAnRX寄存器而该接收数据未被读取前 下一个接收数据操作还没有完成。
<p>•当出现超限误差时，数据被丢弃而下一个接收数据不写入接收缓冲器。 •UAnOVE位可以被读取和写入，但只能通过写入0才能清除而不能通过写入1来设置。 当写入1到UAnOVE位时，该值被保留。</p>	

(6) UARTAn 接收数据寄存器 (UAnRX)

UAnRX 寄存器是 8 位缓冲寄存器，该寄存器存储通过接收移位寄存器转换的并行数据。

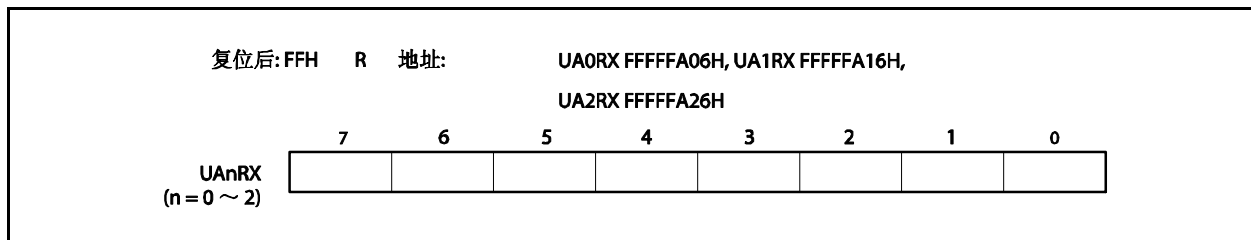
存储在接收移位寄存器中的数据在完成 1 字节数据的接收后发送到 UAnRX 寄存器。

LSB 优先接收期间，当数据长度被指定为 7 位时，接收数据从 UAnRX 寄存器的第 6 位发送到第 0 位，并且 MSB 始终为 0。MSB 优先接收期间，接收数据从 UAnRX 寄存器的第 7 位发送到第 1 位，并且 LSB 始终为 0。

当发生过载错误 (UAnOVE) 时，此时接收数据不能被发送到 UAnRX 寄存器并被丢弃。

该寄存器支持 8 位只读方式。

除了复位输入，UAnRX 寄存器可以通过将 UAnCTL0.UAnPWR 位清零来设置为 FFH。

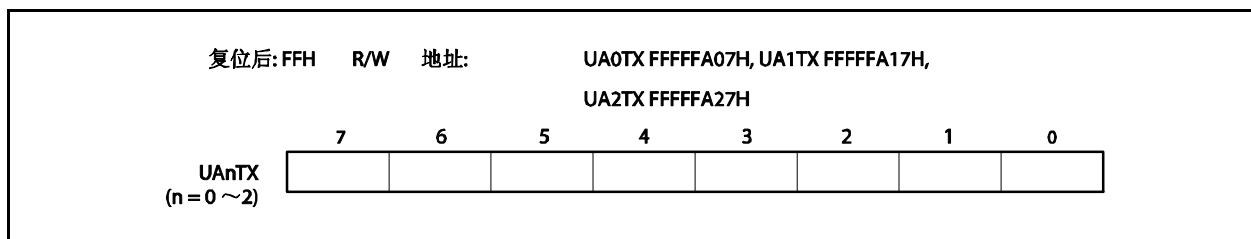


(7) UARTAn 发送数据寄存器 (UAnTX)

UAnTX 寄存器是用于设置发送数据的 8 位寄存器。

该寄存器支持 8 位读写方式。

该寄存器复位后的值为 FFH。



12.4 中断请求信号

UARTAn 生成以下两种中断请求信号。

- 接收完成中断请求信号 (INTUAnR)
- 发送使能中断请求信号 (INTUAnT)

这两种中断请求信号的默认优先级是先接收完成中断请求信号，然后发送使能中断请求信号。

表 12-2. 中断和默认优先级

中断	优先级
接收完成	高
发送使能	低

(1) 接收完成中断请求信号 (INTUAnR)

当数据被移位到接收移位寄存器中时，输出接收完成中断请求信号并在接收使能状态下将该信号发送到 UAnRX 寄存器。

<R> 当发生接收错误时，也可输出接收完成中断请求信号。因此，当接收到接收完成中断请求信号并且该信号被读取时，读取 UAnSTR 寄存器，并检测接收结果不是错误的。

在接收禁止状态下，不生成接收完成中断请求信号。

(2) 发送使能中断请求信号 (INTUAnT)

如果在允许发送时，发送数据从 UAnTX 寄存器发送到 UARTAn 发送移位寄存器，则生成发送使能中断请求信号。

12.5 操作

12.5.1 数据格式

可以进行全双工串行数据接收和发送。

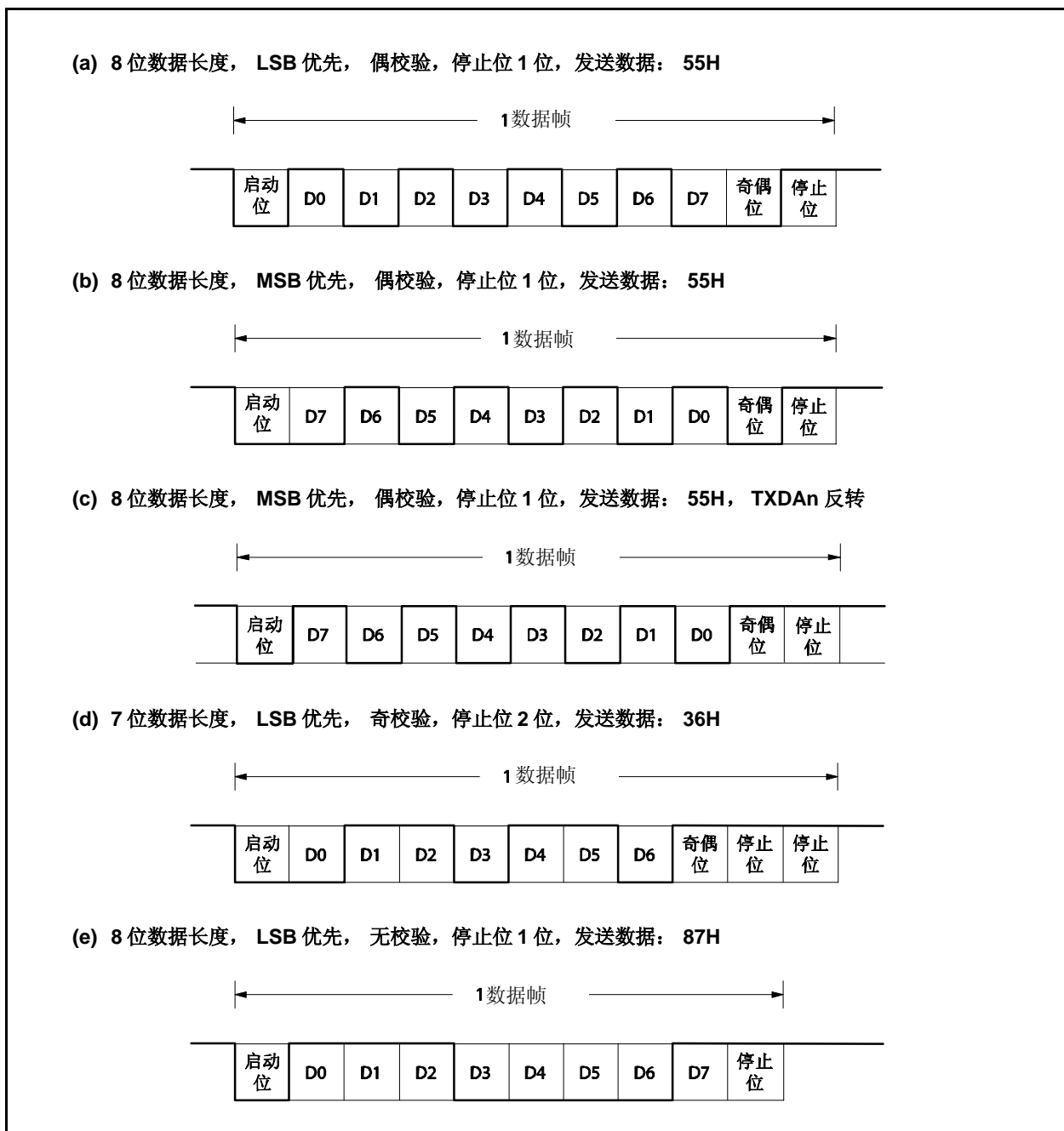
如图 12-2 所示，发送/接收数据包括启动位、字符位、奇偶校验位和停止位的数据帧。

使用 UAnCTL0 寄存器来指定数据帧内的字符位长度，校验位选择，停止位长度和 MSB/LSB 优先发送。

此外，使用 UAnOPT0.UAnTDL 位来控制 TXDAn 位的 UART 输出/反相输出的进行。

- 启动位 1 位
- 字符位 7 位/8 位
- 奇偶校验位 偶校验位/奇校验位/0 校验位/无校验位
- 停止位 1 位/2 位

图 12-2. UARTA 发送/接收数据格式



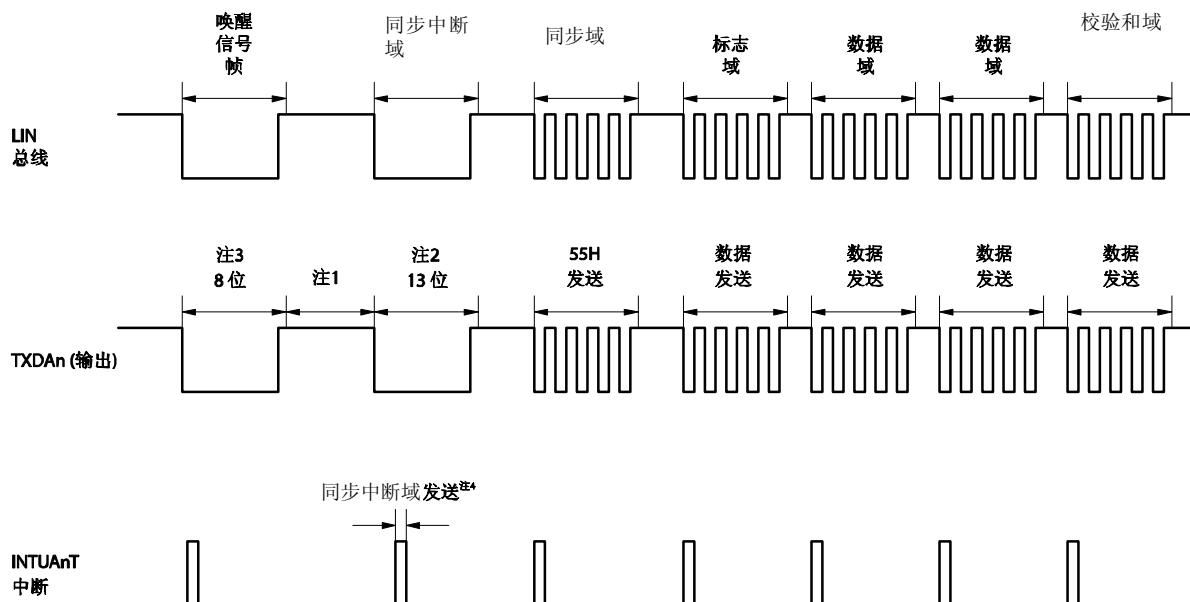
12.5.2 SBF 发送/接收格式

V850ES/HG2 具有 SBF（同步中断区域）发送/接收控制功能，用于使能 LIN 功能。

备注 LIN 表示局部互联网，是一个低速(1 ~ 20 kbps) 串行通信协议，可用于降低网络费用。
 LIN 是一种单主设备通信方式，一个主设备上最多可以连接 15 个从设备。
 LIN 的从设备用于控制开关、制动器和传感器，这些设备通过 LIN 网与 LIN 主设备相连。
 通常，LIN 主设备与一个网络，如 CAN（控制器局域网）相连。
 此外，LIN 总线采用单线方式，通过符合 ISO9141 的收发器与各节点相连。
 在 LIN 协议中，主设备发送带有波特率信息的一帧数据，从设备接收数据并校正波特率错误。因此当从设备端的波特率错误在±15%范围内时，可以进行通信。

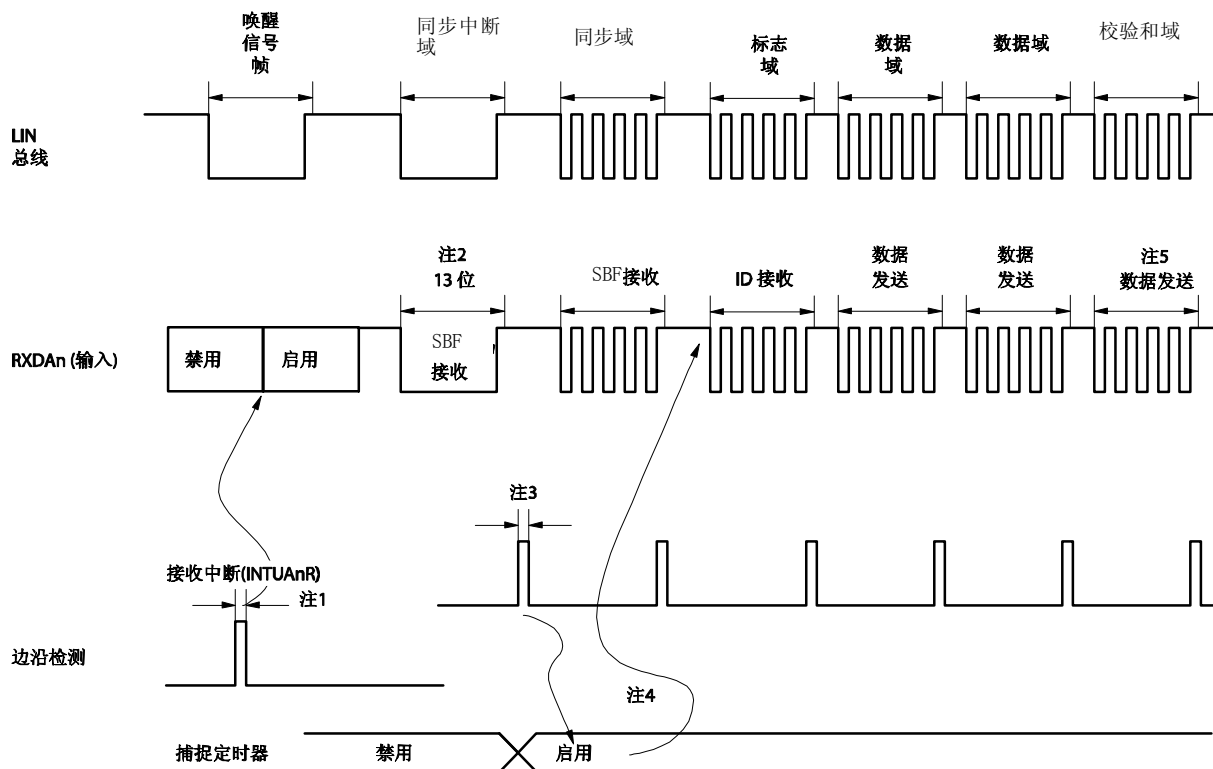
图 12-3 和 12-4 概括了 LIN 的发送和接收操作。

图 12-3. LIN 发送操作略图



- 注**
1. 每个区域的间隔由软件控制。
 2. SBF 输出由硬件控制。UAnOPT0.UAnSBL2 ~ UAnOPT0.UAnSBL0 位设置的输出宽度为位长度。如果要求更合适的宽度调整，可以使用 UAnCTLn.UAnBRS7 ~ UAnCTLn.UAnBRS0 位。
 3. 在 8 位发送模式中，可通过发送 80H 替代唤醒信号帧。
 4. 发送使能中断请求信号 (INTUAnT)在每次发送开始时输出。INTUAnT 信号在每次 SBF 发送开始时也会输出。

图 12-4. LIN 接收操作略图



- 注
1. 在引脚的边沿处检测唤醒信号，使能 UARTAn，设置 SBF 接收模式。
 2. 当检测到停止位时才停止接收操作。在检测到至少 11 位低电平数据的 SBF 时，认为 SBF 已被正确接收，并输出一个中断信号。在检测到少于 11 位低电平数据的 SBF 时，认为出现 SBF 接收错误。此时不输出中断信号，而是恢复 SBF 接收模式。
 3. 如果 SBF 被正确接收，将输出一个中断信号。用这个 SBF 接收完成中断来启动 16 位定时器。此外，禁止检测 UAnSTR.UAnOVE，UAnSTR.UanPE 和 UAnSTR.UAnFE 中的错误，不进行 UART 通信的错误检测，也不执行 UARTAn 接收移位寄存器和 UAnRX 寄存器的数据发送。UARTAn 的接收移位寄存器保持初始值 FFH。
 4. RXDAn 引脚连接到定时器的 TI（捕捉输入），计算发送速率和波特率错误。禁用 UARTA 允许再次被置位时，通过更正波特率错误后，可以获取 UAnCTL2 寄存器的值。
 5. 用软件区分校验和区。在接收到校验和区后通过软件对 UARTAn 进行初始化，并再次通过软件设置 SBF 接收模式。

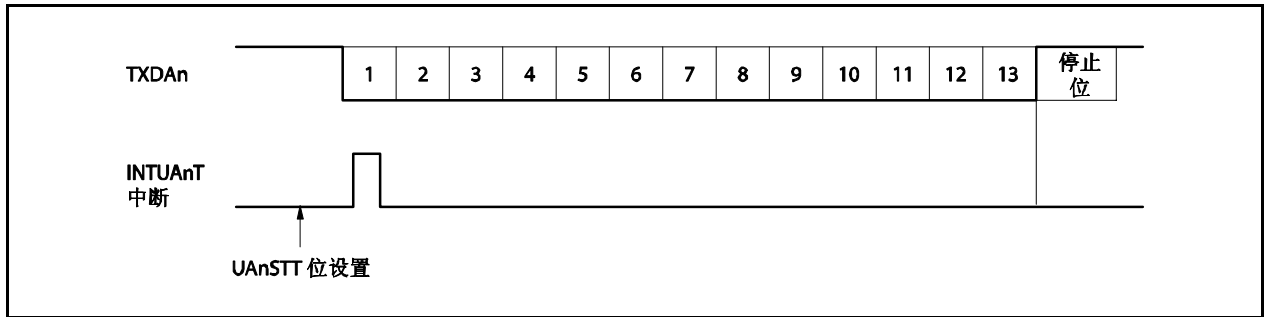
12.5.3 SBF 发送

当 UAnCTL0.UAnPWR 位 = UAnCTL0.UAnTXE 位 = 1，进入发送使能状态，通过设置 SBF 发送触发 (UAnOPT0.UAnSTT 位) (为 1) 后，SBF 发送开始。

之后，输出 UAnOPT0.UAnSLS2 ~ UAnOPT0.UAnSLS0 位来指定第 13~20 位的低电平宽度。SBF 开始发送时，产生发送使能中断请求信号 (INTUAnT)。SBF 发送完之后，UAnSTT 位自动清零。之后，恢复 UART 发送模式。

直到下一个被发送的数据写入 UAnTX 寄存器，或直到设置发送触发 (UAnSTT 位) 时，才停止发送。

图 12-5. SBF 发送



12.5.4 SBF 接收

当设置 UAnCTL0.UAnPWR 位为 1，并设置 UAnCTL0.UAnRXE 位也为 1 时，进入接收使能状态。

由设置 SBF 接收触发 (UAnOPT0.UAnSTR 位) 为 1 时，设置 SBF 接收等待状态。

在 SBF 接收等待状态下，正如 UART 接收等待状态下一样，RXDAn 引脚被监测，并进行启动位检测。

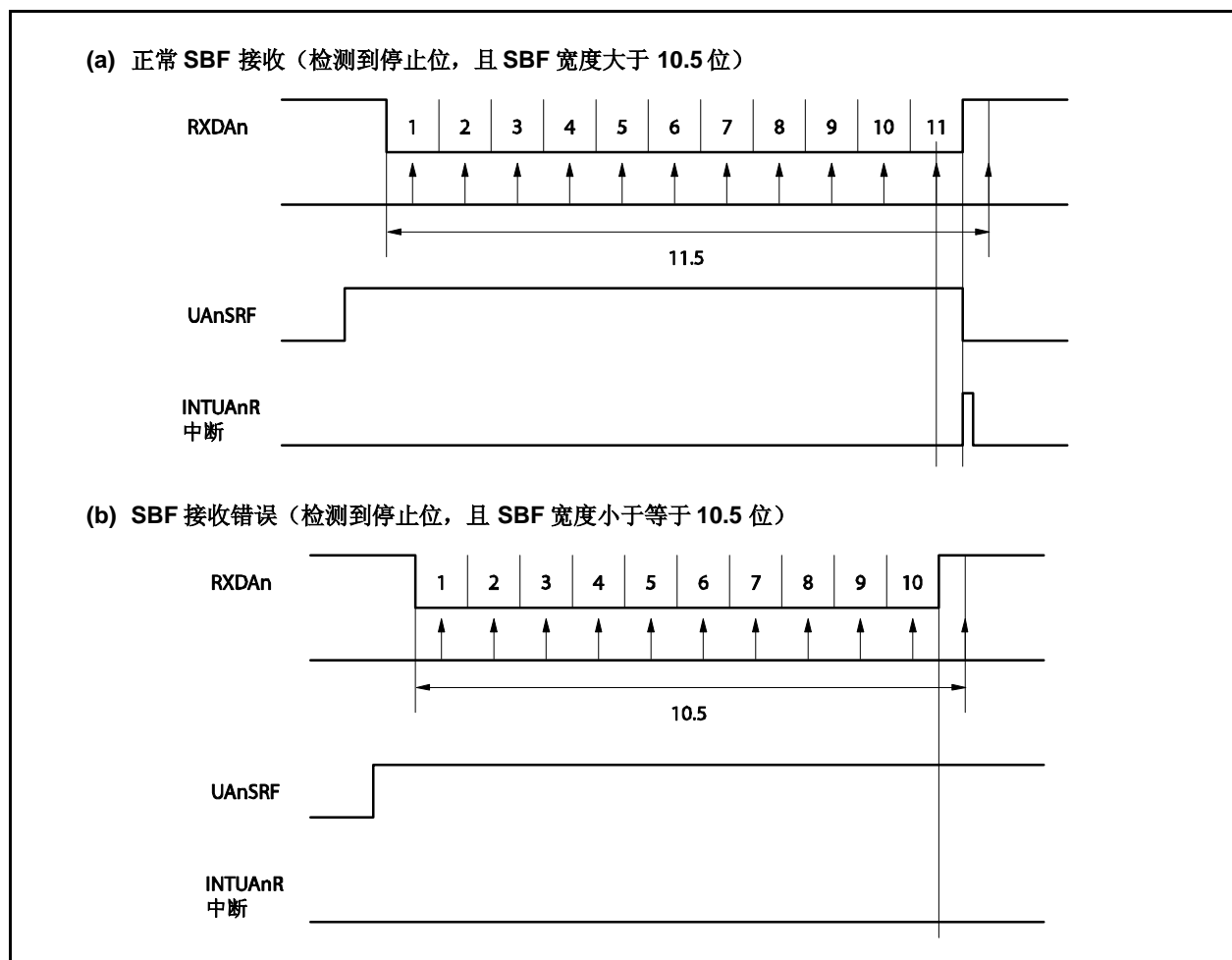
检测到启动位之后，开始接收，内部计数器根据设定的波特率开始计数。

当接收到停止位，如果 SBF 的宽度大于等于 11 位，属正常操作并输出接收完成中断请求信号(INTUAnR)。

UAnOPT0.UAnSRF 位自动清零， SBF 接收结束。禁止检测 UAnSTR.UAnOVE， UAnSTR.UAnPE 和 UAnSTR.UanFE 位中的错误，不进行 UART 通信的错误检测，也不执行 UARTAn 接收移位寄存器和 UAnRX 寄存器的数据传送，并且保持初始值 FFH。如果 SBF 的宽度小于等于 10 位，接收终止，不输出中断，返回 SBF 接收模式。此时，UAnSRF 位不清零。

- <R> 注意事项 1. 在数据接收期间发送 SBF，就会发生帧错误。
- <R> 2. 在 SBF 接收(UAnSRF = 1)期间，不要设置 SBF 接收触发位 (UAnSRT) 和 SBF 发送触发位 (UAnSTT) 为 1。

图 12-6. SBF 接收



12.5.5 UART 发送

通过设置 UAnCTL0.UAnPWR 位为 1，使 TXDAn 引脚输出高电平。

之后，通过设置 UAnCTL0.UAnTXE 位为 1，设置发送使能状态，通过写入发送数据到 UAnTX 寄存器，开始发送。启动位，奇偶校验位和停止位会自动添加。

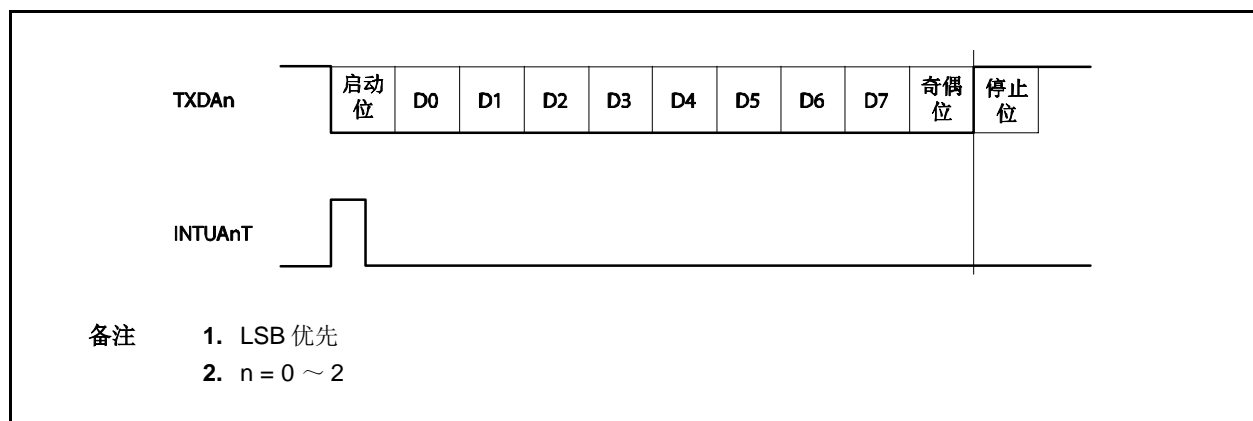
由于 UARTAn 不提供 CTS（发送使能信号）输入引脚，所以使用某个端口检测发送终端的接收状态是否使能。

开始发送时，UAnTX 寄存器中的数据被发送到 UARTAn 发送移位寄存器。

完成 UAnTX 寄存器到 UARTAn 发送移位寄存器的数据发送时，产生发送使能中断请求信号 (INTUAnT)，然后 UARTAn 发送移位寄存器的内容被输出到 TXDAn 引脚。

产生 INTUAnT 信号后，可以写入下一个发送数据到 UAnTX 寄存器。

图 12-7. UART 发送



12.5.6 连续发送

当 UARTAn 发送移位寄存器开始移位操作时，UARTAn 可以写入下一个发送数据到 UAnTX 寄存器。UARTAn 发送移位寄存器的发送时序可以通过发送使能中断请求信号 (INTUAnT)判断。

通过发送过程中向 UAnTX 寄存器写入下一个被发送的数据，判断有效通信率。

连续发送过程中，在产生发送请求中断信号(INTUAnT)之前，将发送数据写入到 UAnTX 寄存器并将其发送到 UARTAn 发送移位寄存器之后，不要将下一个发送数据写入 UAnTX 寄存器。如果在产生发送请求终端信号之前将值写入 UAnTX 寄存器，则之前设定的发送数据被最新的发送数据重写。

注意事项 在执行连续发送过程中发送初始化，确保 UAnSTR.UAnTSF = 0，然后进行初始化。当 UAnTSF = 1 时初始化数据，其发送过程不能保证。

在连续发送的情况下，从停止位到下一个数据的开始位的通信率扩大为普通通信 率加两个运行时钟。

图 12-8. 连续发送操作流程

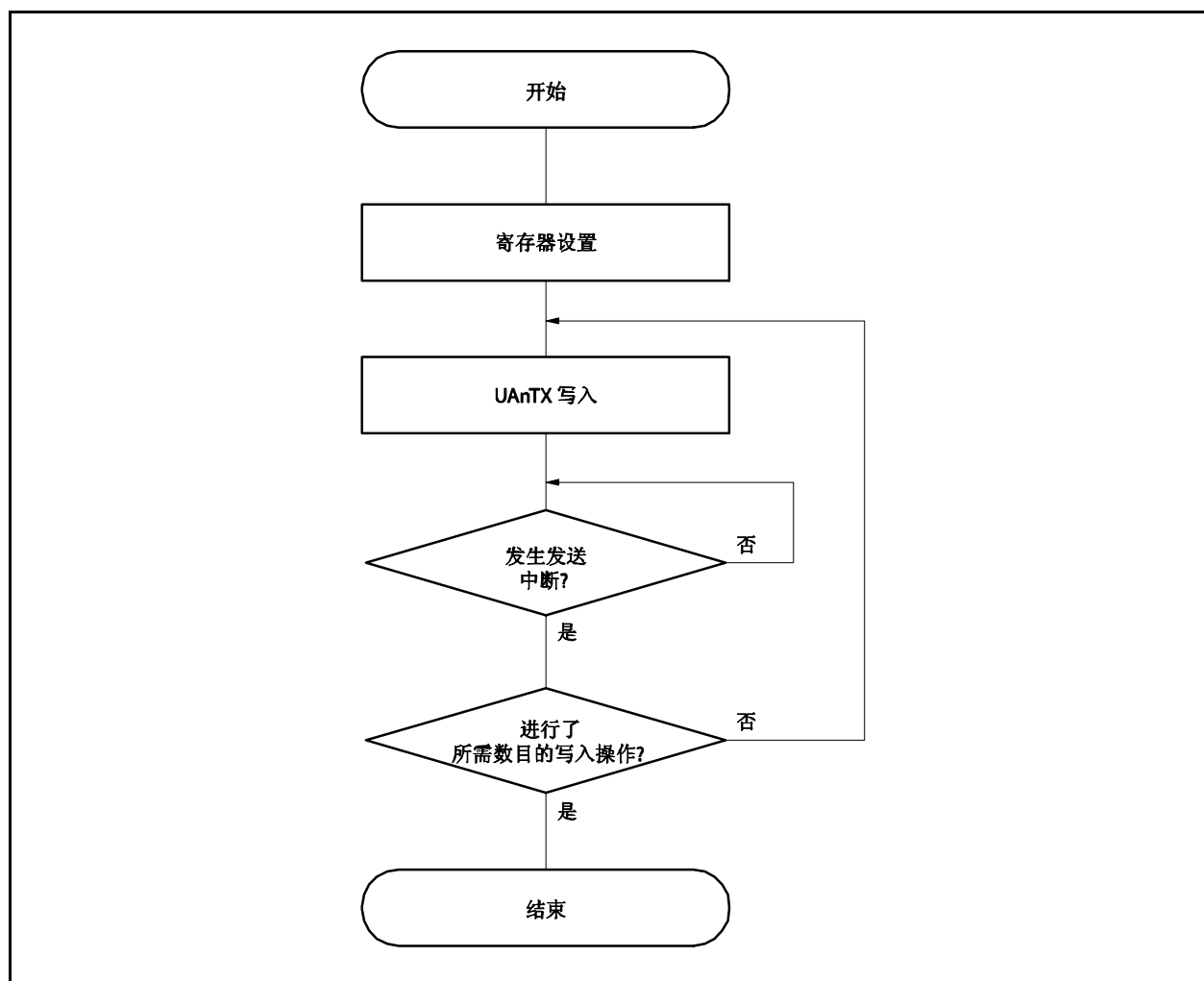
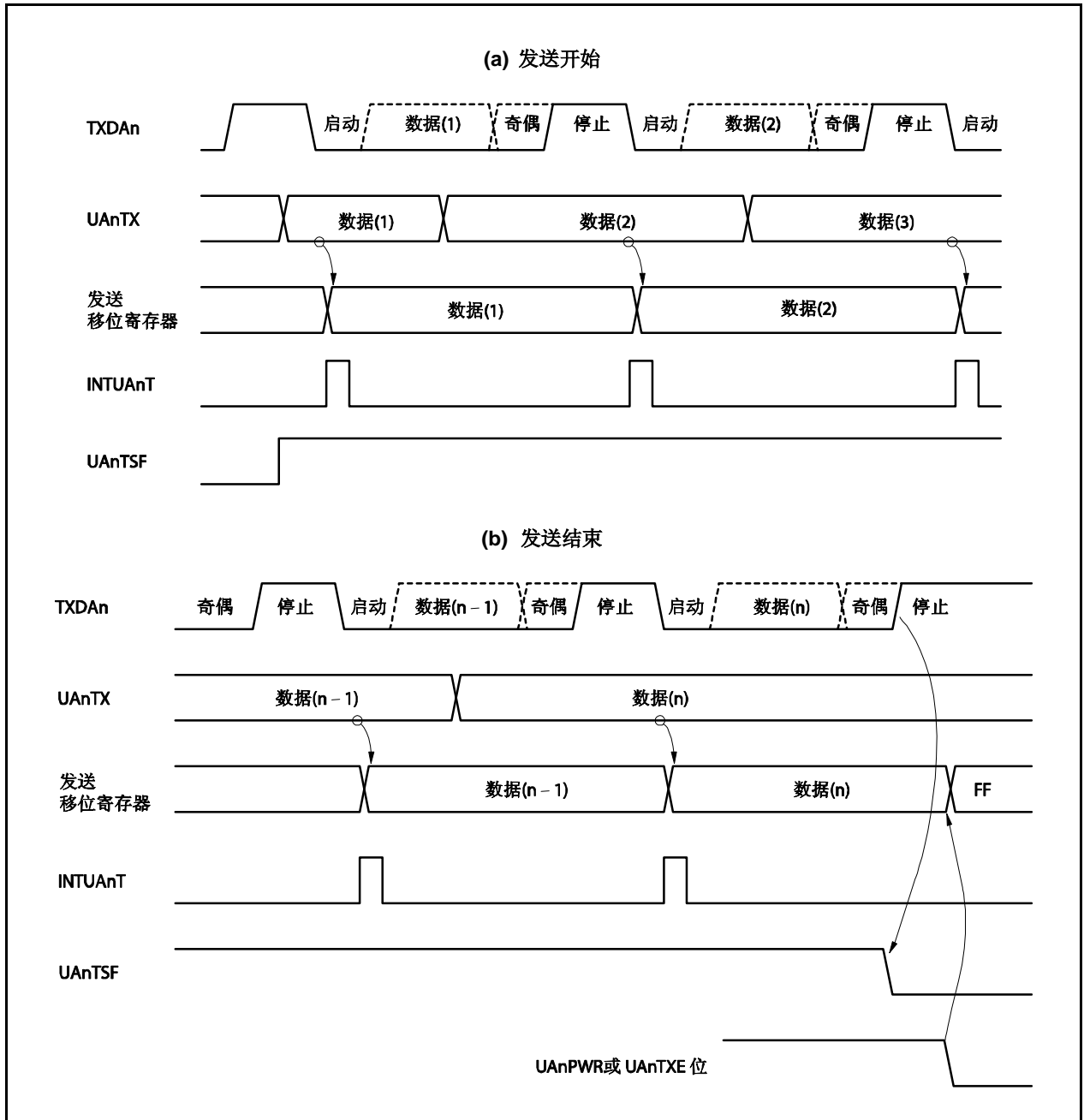


图 12-9. 连续发送操作时序



12.5.7 UART 接收

通过设置 UAnCTL0.UAnPWR 位为 1 并且设置 UAnCTL0.UAnRXE 位也为 1，设置接收等待状态。接收等待状态下，RXDAn 引脚被监测，并且进行启动位检测。

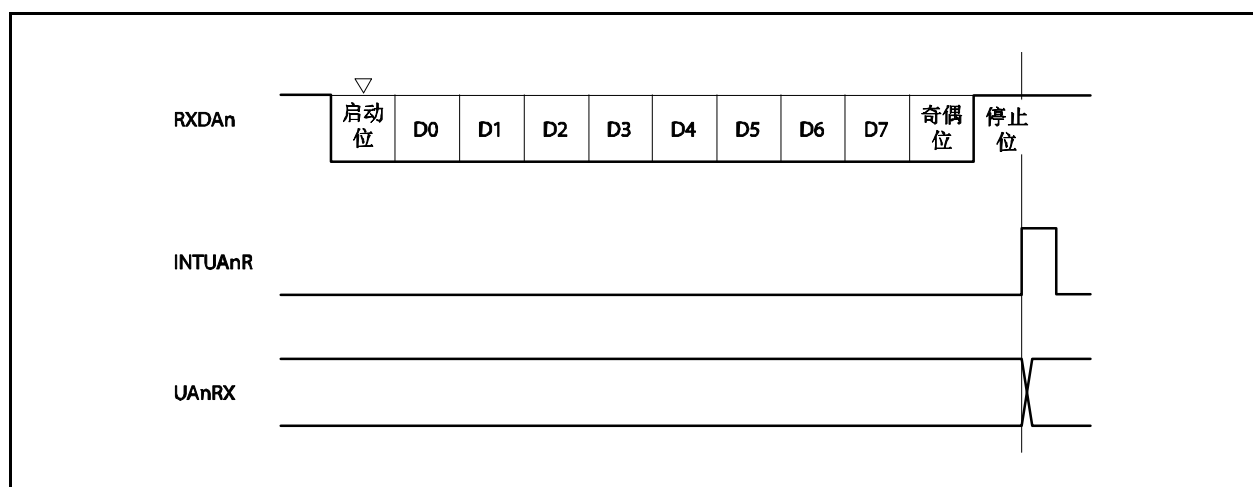
使用两步检测程序进行启动位检测。

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，引脚 RXDAn 为低电平，认为检测到启动位。检测到启动位后，接收操作开始，串行数据根据设定好的波特率保存到 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)，将 UARTAn 接收移位寄存器的数据写入到 UAnRX 寄存器。但是，如果发生过载错误 (UAnSTR.UAnOVE 位)，此时接收数据不被写入到 UAnRX 寄存器中而是被丢弃。

即使在接收过程中产生奇偶校验错误 (UAnSTR.UAnPE 位) 或帧错误 (UAnSTR.UAnFE 位)，直到接收到第一个停止位否则继续接收，并且在接收完成后输出 INTUAnR。

图 12-10. UART 接收



- 注意事项**
1. 即使产生接收错误，也要确保读取 UAnRX 寄存器。如果不读取 UAnRX 寄存器，在接收下一个数据时就会产生过载错误，而且会继续地产生不确定的接收错误。
 2. 接收始终按“停止位的个数 = 1”的情况执行，第 2 个停止位被忽略。
 3. 当接收完成时，在生成接收完成中断请求信号(INTUAnR)后读取 UAnRX 寄存器，并将 UAnPWR 或 UAnRXE 位清零。如果在生成 INTUAnR 信号之前，将 UAnPWR 或 UAnRXE 位清零，则 UAnRX 寄存器的读取值将不能保证。
 4. 如果 UARTAn 接收完成过程中（产生 INTUAnR 信号）UAnPWR 位= 0 或 UAnRXE 位= 0 产生冲突，无论数据是否被存储到 UAnRX 寄存器中都产生 INTUAnR 信号。
为了在不等待 INTUAnR 信号产生的情况下完成接收，在设置中断控制寄存器 (UAnRIC)的中断屏蔽标志(UAnRMK)为 1，且 UAnPWR 位= 0 或 UAnRXE 位= 0 之后，确保将 UAnRIC 寄存器的中断请求标志(UAnRIF)清零。

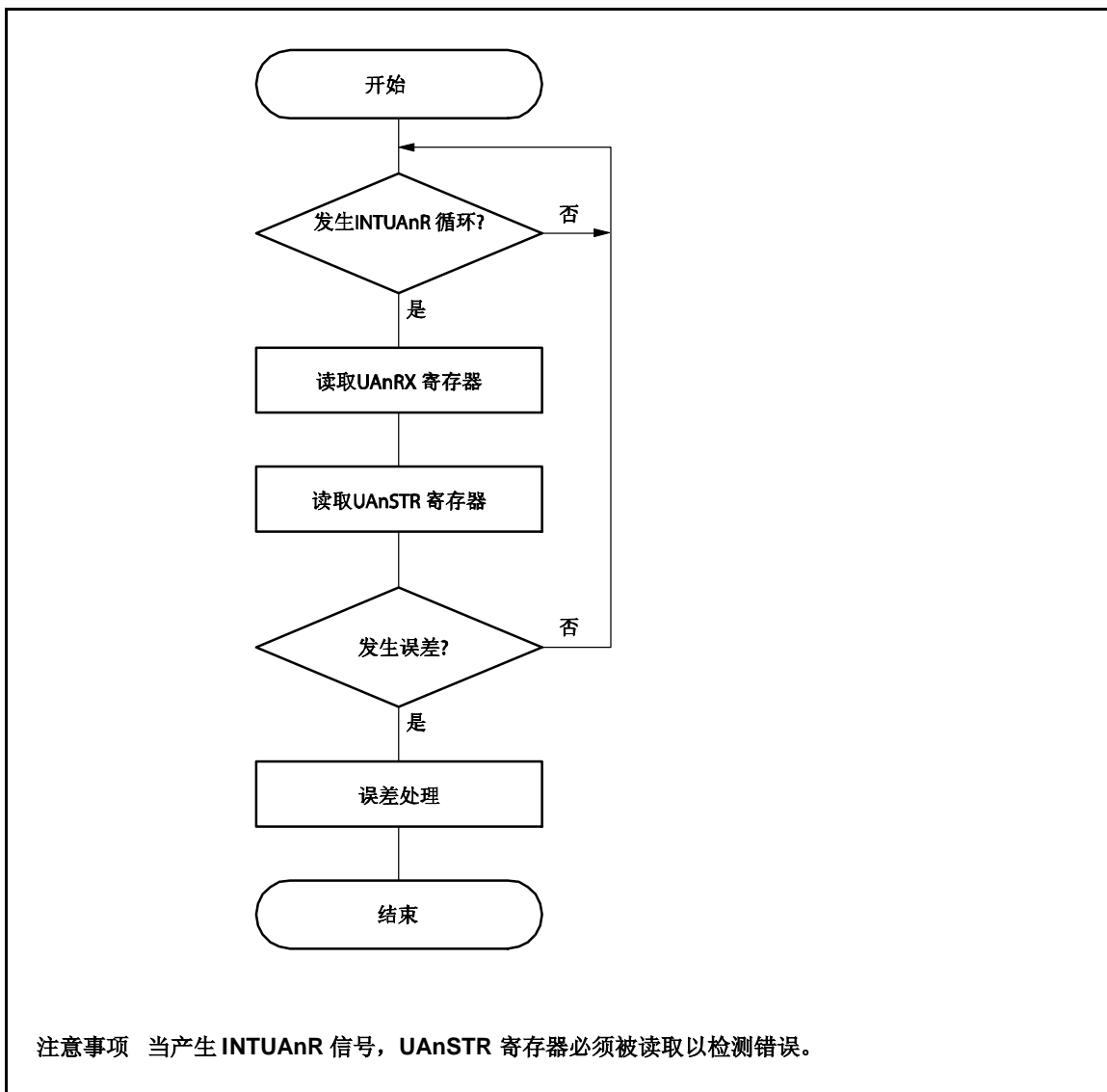
12.5.8 接收错误

接受操作过程中的错误有三种：奇偶校验错误，帧错误和过载错误。在 UAnSTR 寄存器中设置数据接收结果错误，当错误产生时，输出接收完成中断请求信号 (INTUAnR)。

通过读取 UAnSTR 寄存器的内容判断在接收过程中出现何种错误。

在读取接收错误标志后，将该标志清零。

- 接收数据读取流程



- 接收错误原因

错误标志	接收错误	原因
UAnPE	奇偶校验错误	接收的校验位与设置不匹配
UAnFE	帧错误	没检测到停止位
UAnOVE	过载错误	在从接收缓冲器中读取数据前完成了下一个数据的接收

当产生接收错误时，根据错误类型进行如下的相应处理。

- 奇偶校验错误
如果接收过程中由于噪声等原因接收到的错误数据，丢弃该数据并重新发送。
- 帧错误
在发送方和接收方之间可能产生波特率错误，或者启动位可能检测错误。由于帧错误是通信格式的致命错误，检查发送方操作停止，双方重新初始化，然后重新开始通信。
- 过载错误
由于在读取接收到的数据前，下一次接收完成，1 帧的数据将被丢弃。如果需要，重新发送该数据。

注意事项 连续接收过程中产生接收错误中断，在下次接收完成前必须读取 **UAnSTR** 寄存器中的内容，并执行错误处理。

12.5.9 校验的类型和操作

注意事项 当使用 LIN 功能，UAnCTL0.UAnPS1 和 UAnCTL0.UAnPS0 位恒为 00。

奇偶校验位用于检测通信数据中的位错误。通常，发送方和接收方使用同种校验方式。偶校验和奇校验模式下，可以检测到奇数位错误。零校验和无校验模式下，不能检测错误。

(a) 偶校验

(i) 发送

控制发送数据，包括校验位，使得数据中“1”的个数为偶数。校验位的取值如下。

- 如果发送数据有奇数个“1”：1
- 如果发送数据有偶数个“1”：0

(ii) 接收

计算接收数据(包括校验位)中“1”的个数。如果“1”的个数为奇数，则产生校验错误。

(b) 奇校验

(i) 发送

与偶校验相反，控制发送数据，包括校验位，使得数据中“1”的个数为奇数。校验位的取值如下。

- 如果发送数据有奇数个“1”：0
- 如果发送数据有偶数个“1”：1

(ii) 接收

计算接收数据(包括校验位)中“1”的个数。如果“1”的个数为偶数，则输出校验错误。

(c) 0 校验

无论发送数据为何值，发送时校验位清零。

接收数据时不检测校验位。因此无论校验位为何值，都不会产生校验错误。

(d) 无校验

发送数据中没有校验位。

进行接收操作时认为接收数据中没有校验位。因此也不会产生校验错。

12.5.10 接收数据的噪声过滤器

使用由预分频模块输出的基准时钟，该过滤器采样 RXDAn 引脚。

当读取两次相同的采样值时，则匹配检测器的输出会发生变化，并把采样的 RXDAn 信号作为输入数据。因此，不超过 2 个时钟宽度的数据被判断为噪声，不向内部电路发送（参见图 12-12）。敬请参阅 12.6 (1) (a)基本时钟。

此外，如图 12-11 所示的电路结构，接收操作的内部过程与外部信号状态相比延迟三个时钟才执行。

图 12-11. 噪声过滤器电路

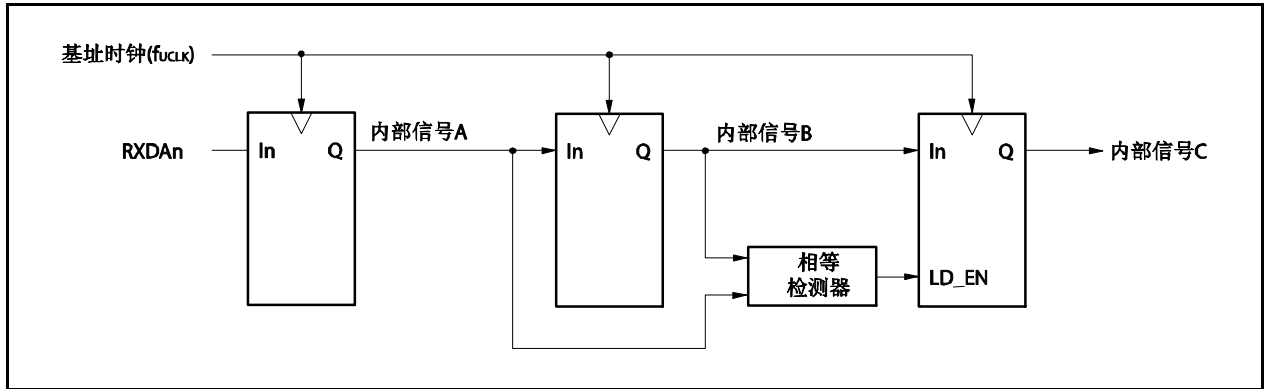
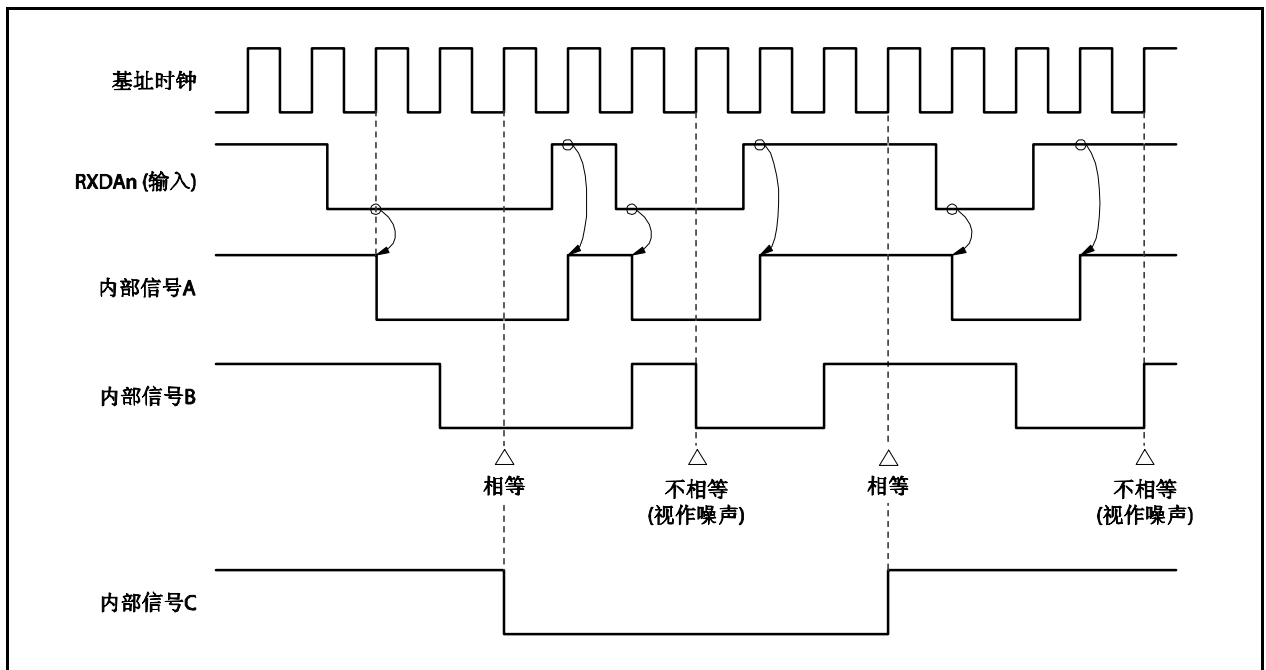


图 12-12. RXDAn 信号判断为噪声的时序



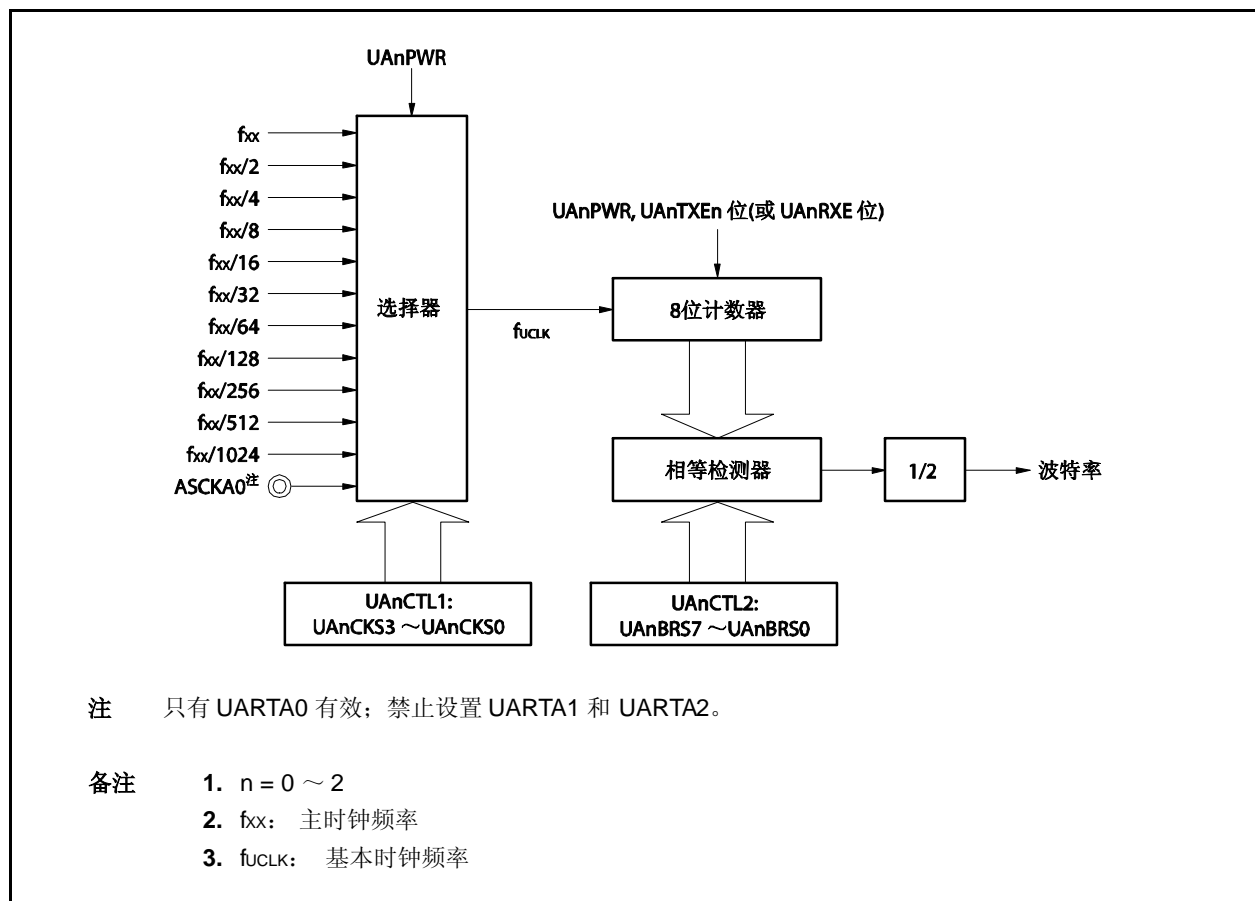
12.6 专用波特率发生器

专用波特率发生器由一个源时钟选择器和一个 8 位可编程计数器组成，用于产生 UARTAn 发送和接收过程中的串行时钟。依据串行时钟，每个通道可选择专用波特率发生器输出。

一个 8 位计数器用于发送，另一个用于接收。

(1) 波特率发生器的配置

图 12-13. 波特率发生器的配置



(a) 基本时钟

当 UAnCTL0.UAnPWR 位为 1 时，由 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 位选择的时钟用作 8 位计数器。该时钟称为基本时钟(fCLK)。

(b) 串行时钟发生器

通过设置 UAnCTL1 寄存器和 UAnCTL2 寄存器(n = 0 ~ 2)，产生串行时钟。

由 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 位选择基本时钟。

通过使用 UAnCTL2.UAnBRS7 ~ UAnCTL2.UAnBRS0 位设置 8 位计数器的分频值。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

UAnCTL1 寄存器是选择 UARTAn 基本时钟的 8 位寄存器。

该寄存器支持 8 位读写方式。

该寄存器复位后的值为 00H。

注意事项 在改写 UAnCTL1 寄存器前，将 UAnCTL0.UAnPWR 位清零。

复位后: 00H R/W 地址: UA0CTL1 FFFFA01H, UA1CTL1 FFFFA11H, UA2CTL1 FFFFA21H

	7	6	5	4	3	2	1	0
UAnCTL1 (n = 0 ~ 2)	0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	基址时钟(fuclk) 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 ^注 (ASCKA0 引脚)
以上除外				禁止设置

注 只有 UARTA0 有效；禁止设置 UARTA1 和 UARTA2。

备注 fxx: 主时钟频率

(3) UARTAn 控制寄存器 2 (UAnCTL2)

UAnCTL2 寄存器是选择 UARTAn 的波特率（串行发送速度）时钟的 8 位寄存器。

该寄存器支持 8 位读写方式。

该寄存器复位后的值为 FFH。

注意事项 在改写 UAnCTL2 寄存器前，将 UAnCTL0.UAnPWR 位清零或将 UAnTXE 和 UAnRXE 位清为 00。

复位后: FFH R/W 地址: UA0CTL2 FFFFFFFA02H, UA1CTL2 FFFFFFFA12H,
UA2CTL2 FFFFFFFA22H

	7	6	5	4	3	2	1	0	
UAnCTL2 (n = 0 ~ 2)	UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0	

UAn BRS7	UAn BRS6	UAn BRS5	UAn BRS4	UAn BRS3	UAn BRS2	UAn BRS1	UAn BRS0	默认 (k)	串行 时钟
0	0	0	0	0	0	-	-	-	禁止 设置
0	0	0	0	0	1	0	0	4	f _{CLK} /4
0	0	0	0	0	1	0	1	5	f _{CLK} /5
0	0	0	0	0	1	1	0	6	f _{CLK} /6
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	f _{CLK} /252
1	1	1	1	1	1	0	1	253	f _{CLK} /253
1	1	1	1	1	1	1	0	254	f _{CLK} /254
1	1	1	1	1	1	1	1	255	f _{CLK} /255

备注 f_{CLK}: 由 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 位选择的时钟频率

(4) 波特率

通过下列公式计算波特率。

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

当使用内部时钟，公式如下（在 UARTA0 工作时使用 ASCKA0 引脚，使用上述公式计算）。

$$\text{波特率} = \frac{f_{\text{XX}}}{2^{m+1} \times k} \text{ [bps]}$$

备注 f_{UCLK} = 由 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 位选择的基本时钟频率
 f_{XX} : 主时钟频率
 m = 由 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 设置的数值($m = 0 \sim 10$)
 k = 由 UAnCTL2.UAnBRS7 ~ UAnCTL2.UAnBRS0 设置的数值($k = 4 \sim 255$)

通过下列公式计算波特率误差。

$$\begin{aligned} \text{误差}(\%) &= \left(\frac{\text{实际波特率 (有误差的波特率)}}{\text{预期波特率 (正确的波特率)}} - 1 \right) \times 100 [\%] \\ &= \left(\frac{f_{\text{UCLK}}}{2 \times k \times \text{预期波特率}} - 1 \right) \times 100 [\%] \end{aligned}$$

当使用内部时钟，公式如下（在 UARTA0 工作时使用 ASCKA0 引脚，使用上述公式计算波特率误差）。

$$\text{误差}(\%) = \left(\frac{f_{\text{XX}}}{2^{m+1} \times k \times \text{目标波特率}} - 1 \right) \times 100 [\%]$$

- 注意事项**
1. 发送期间必须保持波特率误差在接收端允许的误差范围内。
 2. 接收期间，波特率误差必须满足“(5)接收期间允许的波特率范围”中所描述的范围。

为了设置波特率，进行如下计算并设置 UAnCTL1 和 UAnCTL2 寄存器（当使用内部时钟时）。

- <1> 设置 $k = f_{xx}/(2 \times \text{预期波特率})$ 。 设置 $m = 0$ 。
- <2> 设置 $k = k/2$ 和 $m = m + 1$ ，其中 $k \geq 256$ 。
- <3> 重复步骤<2> 直到 $k < 256$ 。
- <4> 上舍入 k 的第一个小数位。
如果通过上舍入 $k = 256$ ，再进行一次<2>（ k 会变成 128）。
- <5> 设置 m 到 UAnCTL1 寄存器并设置 k 到 UAnCTL2 寄存器。

例： 当 $f_{xx} = 20 \text{ MHz}$ 且预期波特率= 153, 600 bps
 <1> $k = 20,000,000/(2 \times 153,600) = 65.10\dots$, $m = 0$
 <2>, <3> $k = 65.10\dots < 256$, $m = 0$
 <4> 设置 UAnCTL2 寄存器的值: $k = 65 = 41\text{H}$, 设置 UAnCTL1 寄存器的值: $m = 0$

$$\begin{aligned} \text{实际波特率} &= 20,000,000/(2 \times 65) \\ &= 153,846 \text{ [bps]} \end{aligned}$$

$$\begin{aligned} \text{波特率误差} &= \{20,000,000/(2 \times 65 \times 153,600) - 1\} \times 100 \\ &= 0.160 \text{ [%]} \end{aligned}$$

波特率设置的典型示例如下所示。

表 12-4. 波特率发生器的设置数据

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

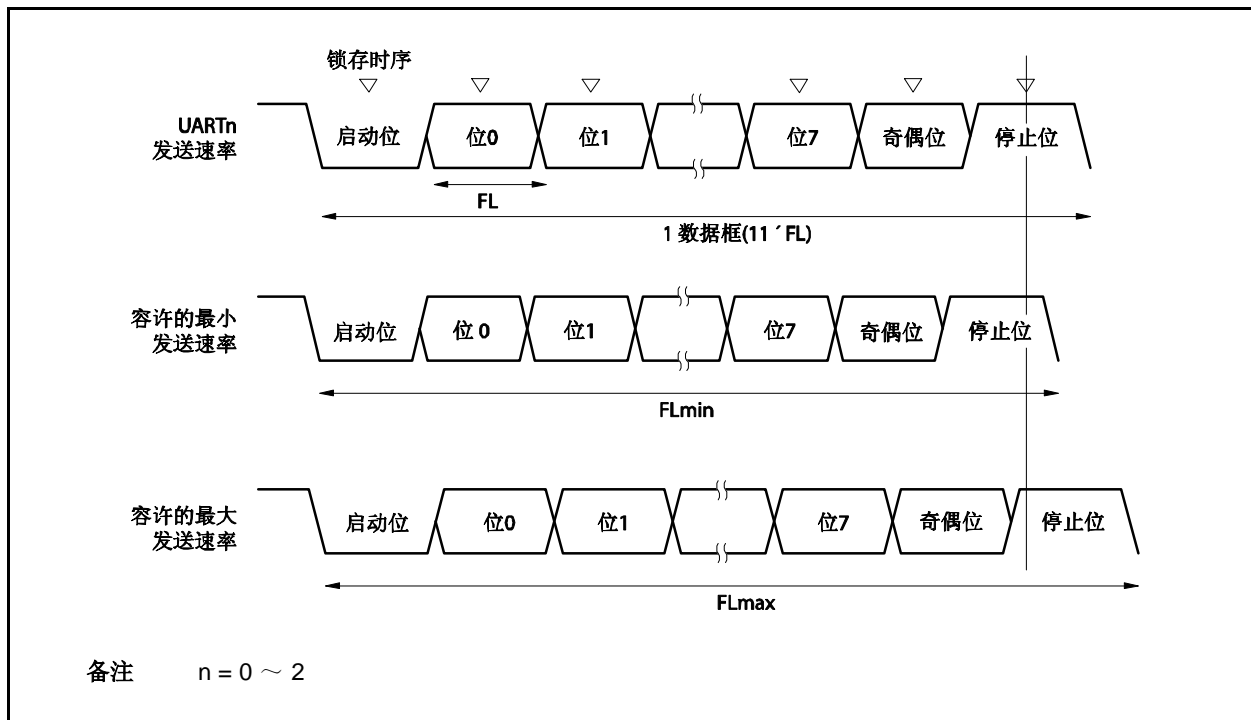
备注 f_{xx} : 主时钟频率
 ERR: 波特率误差 (%)

(5) 接收期间允许的波特率范围

接收期间来自发送端的波特率可允许的误差范围如下所示。

注意事项 必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。

图 12-14. 接收期间允许的波特率范围



如图 12-14 所示，当检测到启动位后，接收数据的锁存时序由 UAnCTL2 寄存器设置的计数器确定。如果数据的最后一位（停止位）满足该锁存时序，则发送数据可被正确接收。假定接收数据为 11 位，各项理论值计算如下。

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

Brate: UARTAn 波特率 (n = 0 ~ 2)

k: UAnCTL2.UAnBRS7 ~ UAnCTL2.UAnBRS0 位的设置值(n = 0 ~ 2)

FL: 1 位数据长度

锁存时序极限: 2 个时钟

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

因此在发送端设定的可接收的最大波特率如下所示。

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

同样，可计算允许的最大数据帧宽度。

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

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

因此在接收端可接收的最小波特率如下所示。

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

可通过上述最小和最大波特率公式计算在 UARTAn 与接收端之间允许的波特率误差，如下所示。

表 12-4. 允许的最大/最小波特率误差

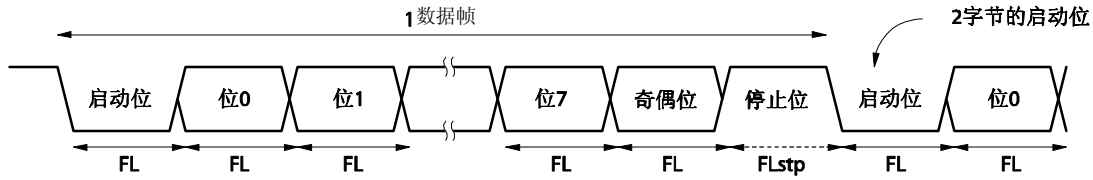
分频比(k)	允许的最大波特率误差	允许的最小波特率误差
4	+2.32%	-2.43%
8	+3.52%	-3.61%
20	+4.26%	-4.30%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.72%

- 备注**
1. 接收准确度取决于每帧的每帧的位数、输入时钟频率和分频比(k)。输入时钟频率和分频比(k)越高，准确度就更高。
 2. k: UAnCTL2.UAnBRS7 ~ UAnCTL2.UAnBRS0 位的设置值(n = 0 ~ 2)

(6) 连续发送期间的波特率

当连续发送数据时，从停止位到下一个启动位的传送速率一般为 2 个基本时钟。不过，由于在检测到启动位时接收端的时序被初始化，因此发送结果不会受到影响。

图 12-15. 连续发送期间的传送速率



设定 1 位数据长度：FL；停止位宽度：FLstp；基本时钟频率：f_{uCLK}，有以下公式成立。

$$FLstp = FL + 2/f_{uCLK}$$

因此，连续发送期间的发送速率如下所示。

$$\text{发送速率} = 11 \times FL + (2/f_{uCLK})$$

12.7 注意事项

- (1) 当 UARTAn 所提供的时钟停止工作（例如，进入 IDLE1，IDLE2，或 STOP 模式），操作停止，每个寄存器保持时钟停止前的值。TXDAn 引脚输出也保持时钟停止前的值。然而，时钟恢复后，操作不被保证。因此，时钟恢复后，通过设置 UAnCTL0.UAnPWR，UAnCTL0.UAnRXEn 和 UAnCTL0.UAnTXEn 为 000，使电路初始化。
- (2) RXDA1 和 KR7 引脚不可以同时使用。要使用 RXDA1 引脚，就不能使用 KR7 引脚。要使用 KR7 引脚，就不能使用 RXDA1 引脚（推荐设置 PFC91 位为 1 且将 PFCE91 位清零）。
- (3) UARTAn 下，不产生由通信错误引起的中断。当使用 DMA 传送发送数据和接收数据时，即使传送期间出现错误（校验，过载，帧错误），也不进行处理。可以在 DMA 传送后读取 UAnSTR 寄存器查看是否出错，或者在通信过程中读取 UAnSTR 寄存器查看是否出错。
- (4) 按照下述次序启动 UARTAn。
 - <1> 设置 UAnCTL0.UAnPWR 位为 1。
 - <2> 设置端口。
 - <3> 设置 UAnCTL0.UAnTXE 位为 1，UAnCTL0.UAnRXE 位为 1。
- (5) 按照下述次序停止 UARTAn。
 - <1> 设置 UAnCTL0.UAnTXE 位为 0，UAnCTL0.UAnRXE 位为 0。
 - <2> 设置端口，设置 UAnCTL0.UAnPWR 位为 0（如果端口设置不改变则不需要修改）。
- (6) 发送模式下(UAnCTL0.UAnPWR 位= 1 和 UAnCTL0.UAnTXE 位= 1)，不要通过软件重写相同值到 UAnTX 寄存器，因为写入该寄存器后发送开始。为了连续发送相同的值，重写相同的值。
- (7) 连续发送模式下，从停止位到下一个启动位的通信速率比正常情况下的 2 个基本时钟要长。然而，接收端通过检测启动位初始化时序，所以结果不受影响。
- (8) 如果在片上调试(OCD)模式下执行中止命令，并且 UART 已接收数据，则产生过载错误。

第十三章 3 线可变长度串行 I/O (CSIB)

V850ES/HG2 具有两个 3 线串行接口 (CSIB) 的通道。

13.1 特点

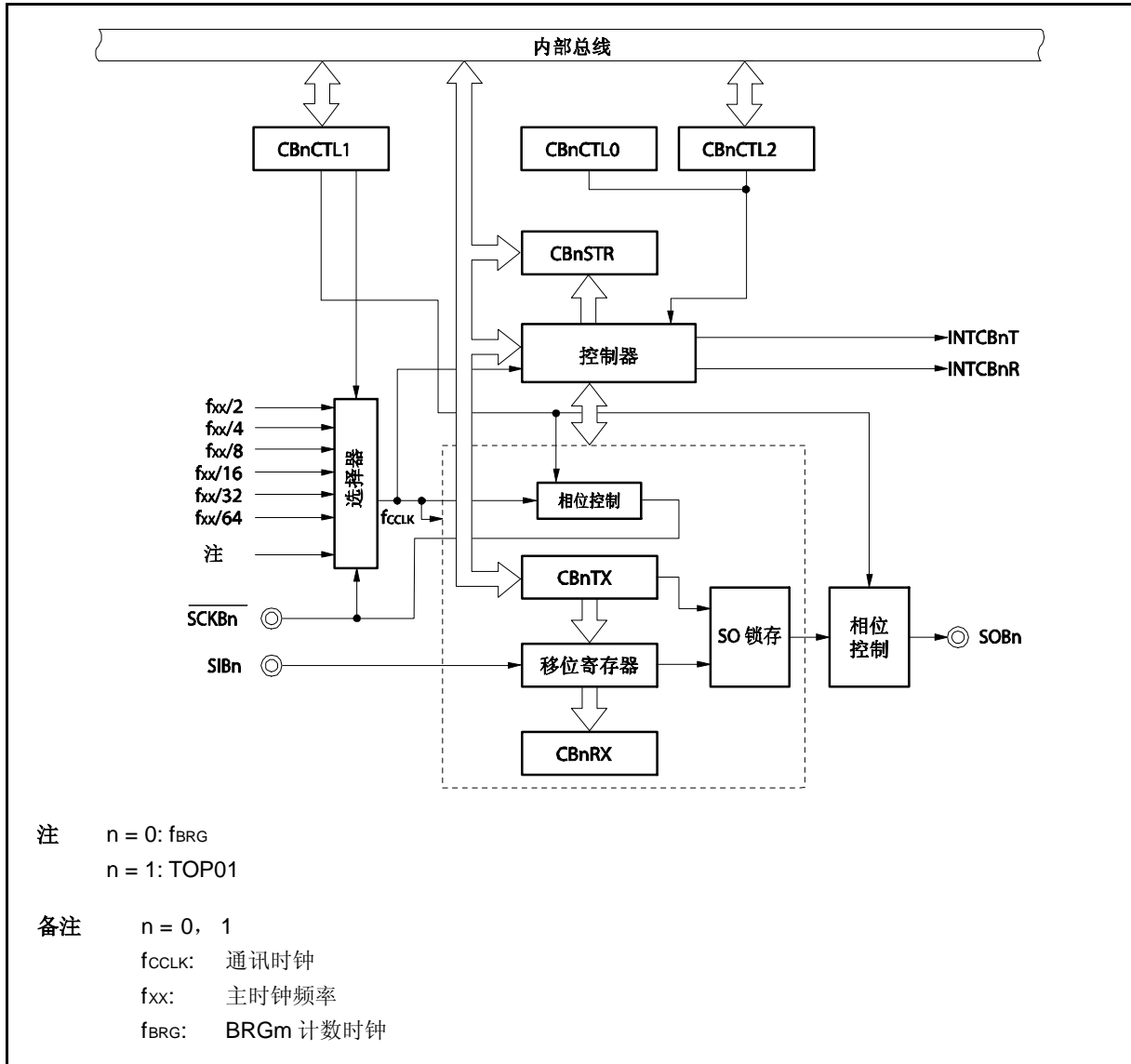
- 传输速率: 最大值 8 Mbps ($f_{xx} = 20 \text{ MHz}$, 使用内部时钟)
 - 可选主模式和从模式
 - 8 位~16 位传输, 3 线串行接口
 - 中断请求信号 (INTCBnT, INTCBnR) $\times 2$
 - 可选串行时钟和数据相位
 - 可由 1 位操作指令在 8 和 16 位中选择传输数据长度
 - 可转换 MSB/LSB 优先传输数据
 - 3 线传输
 - SOBn: 串行数据输出
 - SIBn: 串行数据输入
 - SCKBn: 串行时钟 I/O
- 可指定发送模式, 接收模式且发送/接收模式

备注 n = 0, 1

13.2 配置

CSIBn 的框图如下。

图 13-1. CSIBn 的框图



CSIBn 包括以下硬件。

表 13-1. CSIBn 的配置

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 16 位缓冲寄存器，用于保持接收数据。

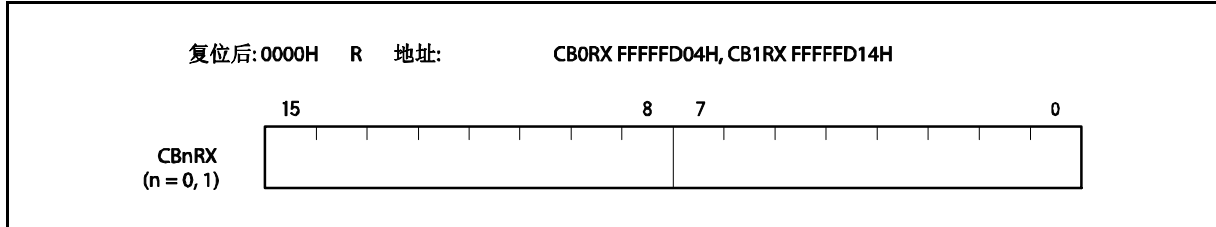
该寄存器支持 16 位只读方式。

在接收使能状态下，通过读取寄存器 CBnRX 开始接收操作。

如果传输数据的长度是 8 位，该寄存器的低 8 位是只读的，由 8 位存储器操作指令设置寄存器 CBnRXL。

该寄存器复位后的值为 0000H。

除了复位输入，通过清除（置 0）寄存器 CBnCTL0 的 CBnPWR 位，寄存器 CBnRX 可被初始化。



(2) CSIB 发送数据寄存器 (CBnTX)

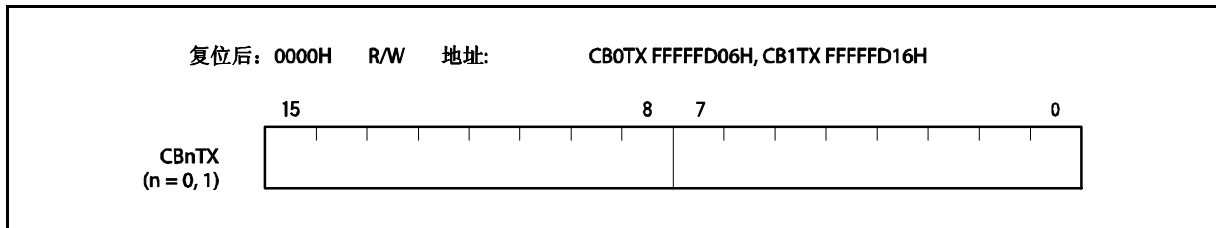
CBnTX 寄存器是 16 位缓冲寄存器，用于写入 CSIBn 发送数据。

该寄存器支持 16 位读写方式。

在发送使能状态下，通过写入数据到 CBnTX 寄存器中开始发送操作。

如果传输数据的长度是 8 位，该寄存器的低 8 位是只读的，由 8 位存储器操作指令设置寄存器 CBnTXL。

该寄存器复位后的值为 0000H。



备注 如下为传输开始的条件。

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

读取 CBnRX 寄存器

13.3 寄存器

下述寄存器用于控制 CSIBn。

- 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 寄存器 CSIBn 控制串行传输操作。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 01H。

(1/3)

复位后: 01H	R/W	地址:	CB0CTL0 FFFFFFFD00H, CB1CTL0 FFFFFFFD10H					
CBnCTL0 (n = 0, 1)	7	6	5	4	3	2	1	0
	CBnPWR	CBnTXE [‡]	CBnRXE [‡]	CBnDIR [‡]	0	0	CBnTMS [‡]	CBnSCE
CBnPWR		CSIBn 操作禁用/启用规格						
0		禁用CSIBn 操作及 复位 CBnSTR 寄存器						
1		启用CSIBn 操作						
		•CBnPWR 位控制 CSIBn 操作及 复位内部电路。						
CBnTXE [‡]		发送操作禁用/启用规格						
0		禁用发送操作						
1		启用发送操作						
		• 当 CBnTXE 位为 0 时, SOBn 输出为低电平。						
CBnRXE [‡]		接收操作禁用/启用规格						
0		禁用接收操作						
1		启用接收操作						
		• 当 CBnRXE 位清零时, 即使 通过传输指定 数据来禁用接收操作, 也无接收完整中断输出, 及接收数据 (CBnRX 寄存器) 没有得到更新。						

注 这些位只能在 CBnPWR 位 = 0 时重写。但在重写这些位时可以设置 CBnPWR 位 = 1。

注意事项 为了强行停止发送/接收, 清除 CBnPWR 位, 而不是 CBnRxE 位或 CBnTXE 位。同时, 时钟输出停止。

CBnDIR ^注	传输方向模式规格(MSB/LSB)
0	最高有效位 (MSB) 先传输
1	最低有效位 (LSB) 先传输

CBnTMS ^注	传输模式规格
0	单一传输模式
1	连续传输模式

[单一传输模式]

发生接收完成中断请求信号 (INTCBnR)。即使启用传输 (CBnTXE位=1)，则不发生传输启用中断请求信号 (INTCBnT)。如果在通讯过程中写入下一个传输数据 (CBnSTR.CBnTSF位=1)，则数据被忽略，下一个数据传输不会开始。同时，如果设置了只接收的通讯 (CBnTXE位=0, CBnRXE位=1)，则即使在通讯过程中 (CBnSTR.CBnTSF位=1) 读取接收数据，下一次通讯仍会开始。

[连续传输模式]

可以通过在通讯过程中 (CBnSTR.CBnTSF位=1) 写入下一个传输数据来启用连续传输模式。在传输启用中断 (INTCBnR) 出现后启用写入下一个传输数据。如果在连续传输模式下设置了只接收的通讯 (CBnTXE位=0, CBnRXE位=1)，则在一次接收完成中断 (INTCBnR) 后开始连续的下一接收而不管CBnR寄存器的读取操作。因此，立即从CBnR寄存器读取接收数据。如果显示读取操作，则出现过限误差 (UAnOVE位=1)。

注 这些位只能在 CBnPWR 位= 0 时重写。但在重写这些位时可以设置 CBnPWR 位= 1。

CBnSCE	开始传输禁用/启用规格
0	通讯开始触发器无效
1	通讯开始触发器有效
<p>•在主模式下 该位启用或禁用通讯开始触发器。 (a) 在单一传输或传输/接收模式下或者连续传输或传输/接收模式下 CBnSCE位的设置对通讯操作没有影响。</p> <p>(b) 在单一接收模式下 在读取最后的接收数据前将CBnSCE位设为0，因为接收是通过读取接收数据开始的， 从而禁用接收启动^{注1}。</p> <p>(c) 在连续接收模式下 在接收最后的接收数据完成的一个通讯时钟前将CBnSCE位设为0， 从而在接收最后的数据后禁用接收启动^{注2}。</p> <p>•在从模式下 该位启用或禁用通讯开始触发器。 将CBnSCE位设为1。</p> <p>[CBnSCE位的使用]</p> <p>•在单一接收模式下</p> <p>(1) 当INTCBnR中断服务完成了最后的数据接收后， 在读取CBnR寄存器前将CBnSCE位设为0。</p> <p>(2) 在确认CBnSTR.CBnTSF位=0后，将CBnRXE位设为0以禁止接收。 要继续接收，将CBnRXE位设为1以通过虚拟读取CBnRX寄存器开始下一次接收</p> <p>•在连续接收模式下</p> <p>(1) 在INTCBnR中断服务接收最后的数据过程中，将CBnSCE位设为0。</p> <p>(2) 读取CBnRX寄存器。</p> <p>(3) 在确认CBnTIR中断后通过读取CBnRX寄存器读取最后的接收数据。</p> <p>(4) 在确认CBnSTR.CBnTSF位=0，将CBnRXE位设为0以禁止接收。 要继续接收，将CBnSCE位设为1等待下一次通过虚拟读取CBnRX寄存器的接收。</p>	

- 注
1. 当 CBnSCE 位为 1 时被读取，下一次传输操作开始。
 2. CBnSCE 位在接收到最后一个数据接收前不能被清零，下一次传输操作自动开始。

注意事项 确保将位 3 和 2 清零。

(2) CSIBn 控制寄存器 1 (CBnCTL1)

CBnCTL1 是 8 位寄存器，用于控制 CSIBn 串行传输操作。
 该寄存器支持 8 位或 1 位读写方式。
 该寄存器复位后的值为 00H。

<R> **注意事项** 只有当 CBnCTL0.CBnPWR = 0 或 CBnCTL0.CBnTXE 和 CBnRXE 位都为 0 时，寄存器 CBnCTL1 可被重写。

复位后: 00H R/W 地址: CB0CTL1 FFFFD01H, CB1CTL1 FFFFD11H

	7	6	5	4	3	2	1	0
CBnCTL1 (n = 0, 1)	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

	CBnCKP	CBnDAP	与 SCKBn 相关的数据传输/接收 时序规格					
通讯类型1	0	0						
通讯类型2	0	1						
通讯类型3	1	0						
通讯类型4	1	1						

CBnCKS2	CBnCKS1	CBnCKS0	通讯时钟(f _{CLK}) ^{注1}		模式
			n = 0	n = 1	
0	0	0	f _{CLK} /2		主模式
0	0	1	f _{CLK} /4		主模式
0	1	0	f _{CLK} /8		主模式
0	1	1	f _{CLK} /16		主模式
1	0	0	f _{CLK} /32		主模式
1	0	1	f _{CLK} /64		主模式
1	1	0	f _{BRG} ^{注2}	TMPO (TOP01)	主模式
1	1	1	外部时钟(SCKBn)		从模式

<R> **注**

1. 设置使通讯时钟 (f_{CLK}) 小于等于 8 MHz。
2. 关于更多细节，敬请参阅 13.7 波特率发生器。

(3) CSIBn 控制寄存器 2 (CBnCTL2)

CBnCTL2 是 8 位寄存器，用于控制 CSIBn 串行传输的字节长度。

该寄存器支持 8 位读写方式。

该寄存器复位后的值为 00H。

注意事项 只有当 **CBnCTL0.CBnPWR = 0** 或者 **CBnTXE** 和 **CBnRXE** 位都为 0 时，寄存器 **CBnCTL2** 可被重写。

复位后: 00H R/W 地址: CB0CTL2 FFFFFFFD02H, CB1CTL2 FFFFFFFD12H

	7	6	5	4	3	2	1	0
CBnCTL2 (n = 0, 1)	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

CBnCL3	CBnCL2	CBnCL1	CBnCL0	串行寄存器位长
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

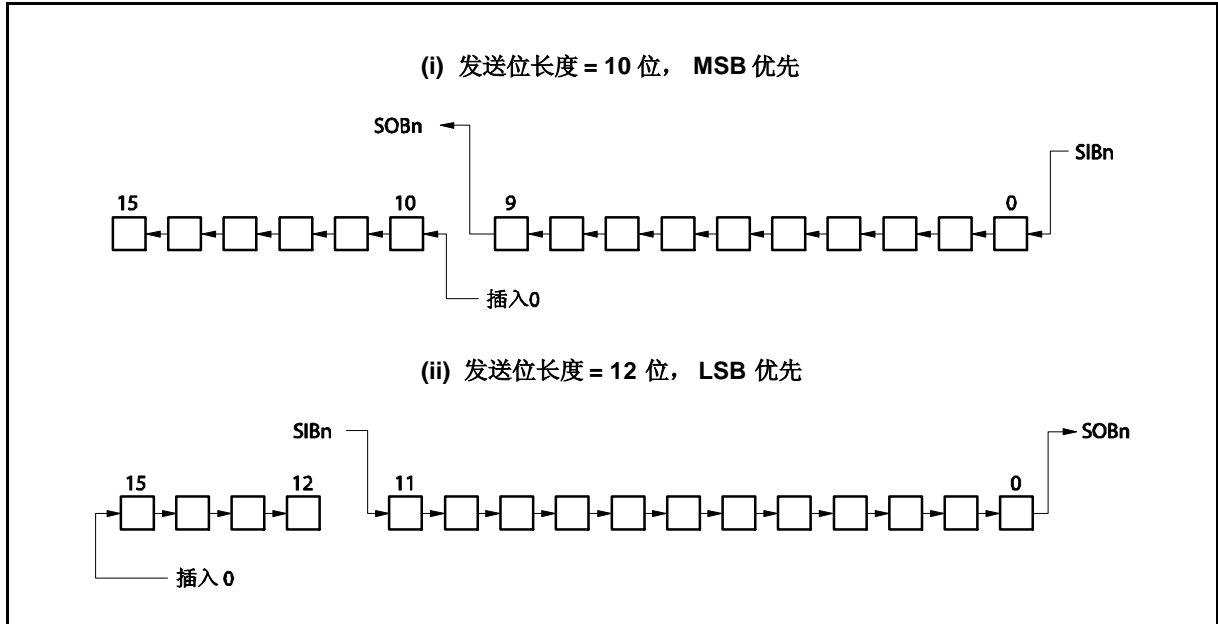
备注

1. 如果发送字节数不是 8 或者 16 位，准备和使用存储于 CBnTX 和 CBnRX 寄存器的数据的最低有效位。
2. × 忽略

(a) 用传输数据长度改变功能

使用 $CBnCTL2.CBnCL3 \sim CBnCTL2.CBnCL0$ 位，以 1 位操作指令设置长度为 8~16 位的 CSIBn 传输数据。

当发送位长度设置为非 16 位，从 LSB 开始将数据放置到寄存器 $CBnTX$ 或 $CBnRX$ ，无论传输起始位是 MSB 或者 LSB。任何数据都可设置到那些不用的高字节位，但是接收的数据的高字节位在串行发送后都变为 0。



(4) CSIBn 状态寄存器 (CBnSTR)

CBnSTR 是 8 位寄存器，用于显示 CSIBn 的状态。

该寄存器支持 8 位或 1 位读写方式，但 CBnTSF 标志是只读的。

该寄存器复位后的值为 00H。

除了复位输入，CBnSTR 寄存器可通过请零 CBnCTL0.CBnPWR 位使其初始化。

复位后: 00H		R/W	地址: CB0STR FFFFFFFD03H, CB1STR FFFFFFFD13H					
CBnSTR (n = 0, 1)	7	6	5	4	3	2	1	0
	CBnTSF	0	0	0	0	0	0	CBnOVE
CBnTSF	通讯状态标记							
0	通讯停止							
1	通讯中							
<ul style="list-style-type: none"> 在发送中，当数据由CBnTX寄存器准备时，设置该寄存器。在接收中，当运行CBnRX寄存器中的伪读取时，设置该寄存器。当发送结束时，该标记在时钟的最后边沿被清至零。 								
CBnOVE	过限误差标记							
0	没有过限							
1	过限							
<ul style="list-style-type: none"> 当下一次接收在没有读取CPU的接收缓冲值的情况下完成，或在接收操作完成后，会发生过限误差。 CBnOVE 标记在该情况下展示的是 过限误差发生状态。 CBnOVE 位在单程传输模式下仍然有效。因此，当只使用发送时，注意以下情形。 <ul style="list-style-type: none"> 勿检查 CBnOVE 标记。 Read this bit 即使不要求读取接收的数据也要读取该位。 CBnOVE 标记通过写0来进行清除。即使写入1也无法设置 CBnOVE 标记。 								

13.4 中断请求信号

CSIBn 可以产生以下两种中断请求信号。

- 接收完成中断请求信号 (INTCBnR)
- 发送使能中断请求信号 (INTCBnT)

在这两种中断请求信号中，接收完成中断请求信号在默认值中具有更高的优先级，而发送使能中断请求信号的优先级则低一些。

表 13-2. 中断和默认优先级

中断	优先级
接收完成	高
发送使能	低

(1) 接收完成中断请求信号 (INTCBnR)

在接收使能时，接收数据被传输到 CBnRX 寄存器，此时将产生接收完成中断请求信号。

在过载错误发生时同样会产生此信号。

当确认接收完成中断请求信号并读取数据后，读 CBnSTR 寄存器检查接收结果是否为错误。

在单向传输模式中，INTCBnR 中断请求信号在发送完成后产生，即使当仅有发送命令被执行。

(2) 发送使能中断请求信号 (INTCBnT)

在连续的发送模式或连续的发送/接收模式下，发送数据从 CBnTX 寄存器中传输。一旦数据写入 CBnTX 寄存器被启用，将产生发送使能中断请求信号。

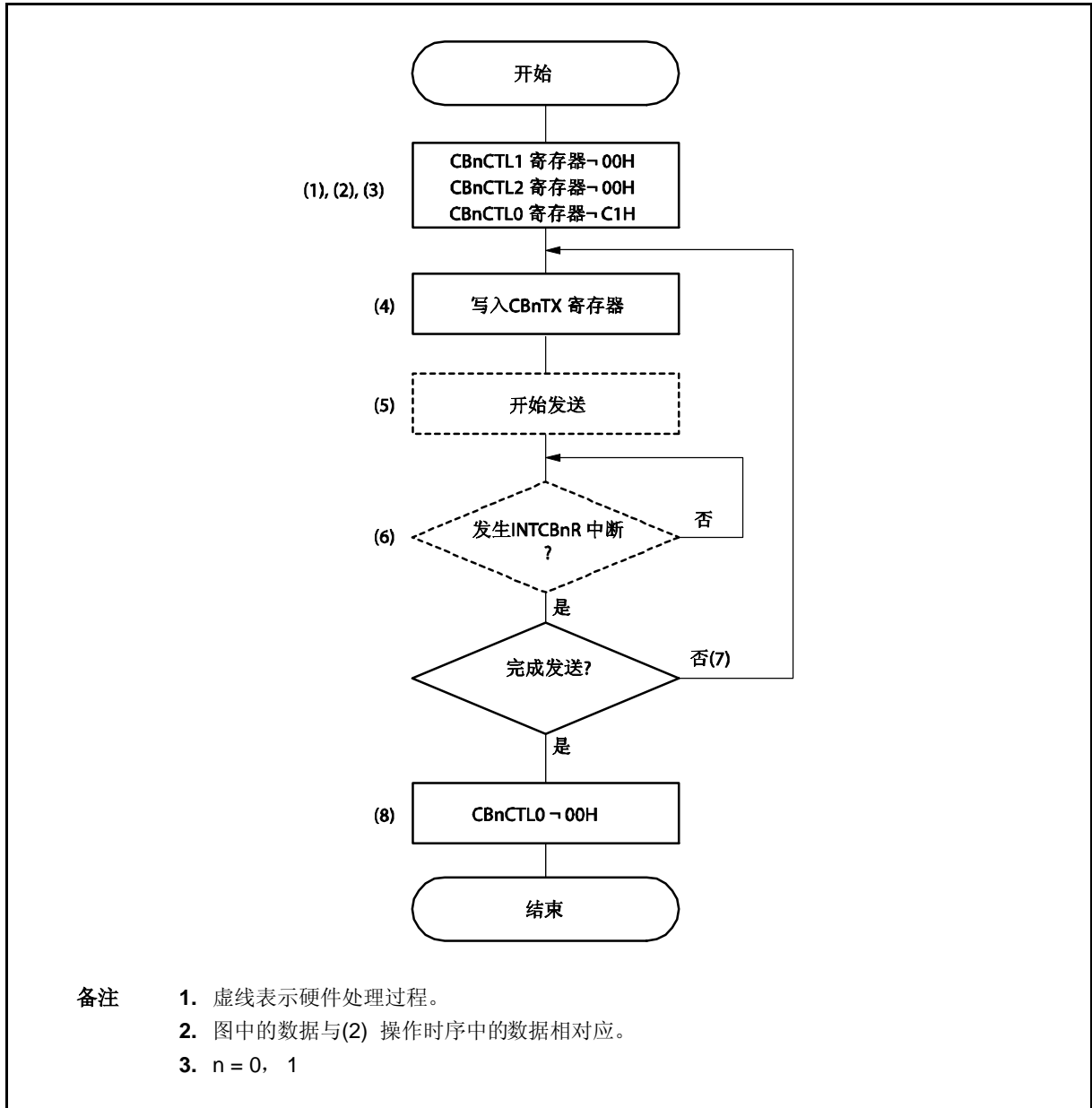
在单向发送模式和单向发送/接收模式下不产生发送使能中断请求信号。

<R> 13.5 操作

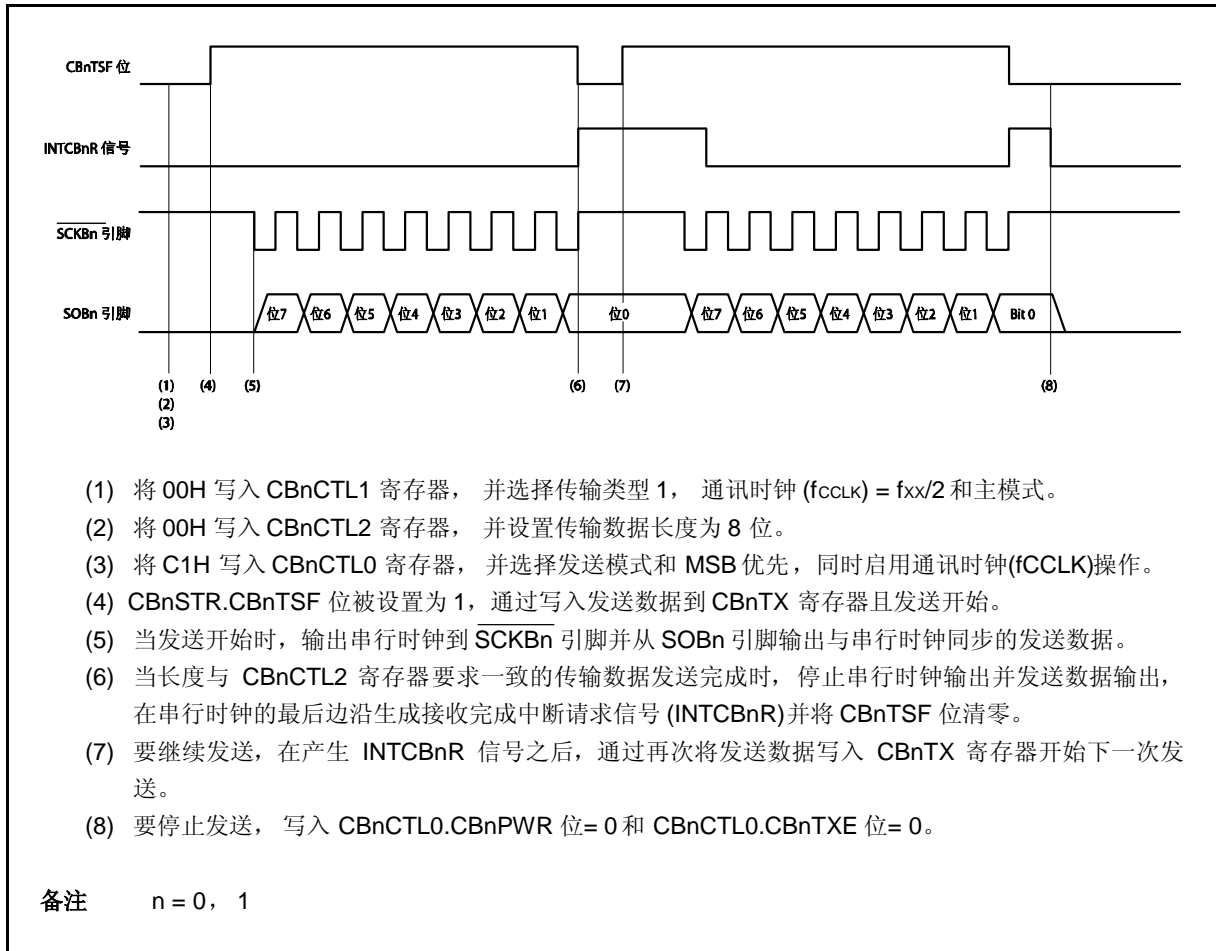
13.5.1 单传输模式（主模式，发送模式）

MSB 优先 (CBnCTL0.CBnDIR 位 = 0)，传输类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通讯时钟 (f_{CCLK}) = f_{xx}/2 (CBnCTL1.CBnCKS2 ~ CBnCTL1.CBnCKS0 位 = 000)，传输数据长度 = 8 位 (CBnCTL2.CBnCL3 ~ CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



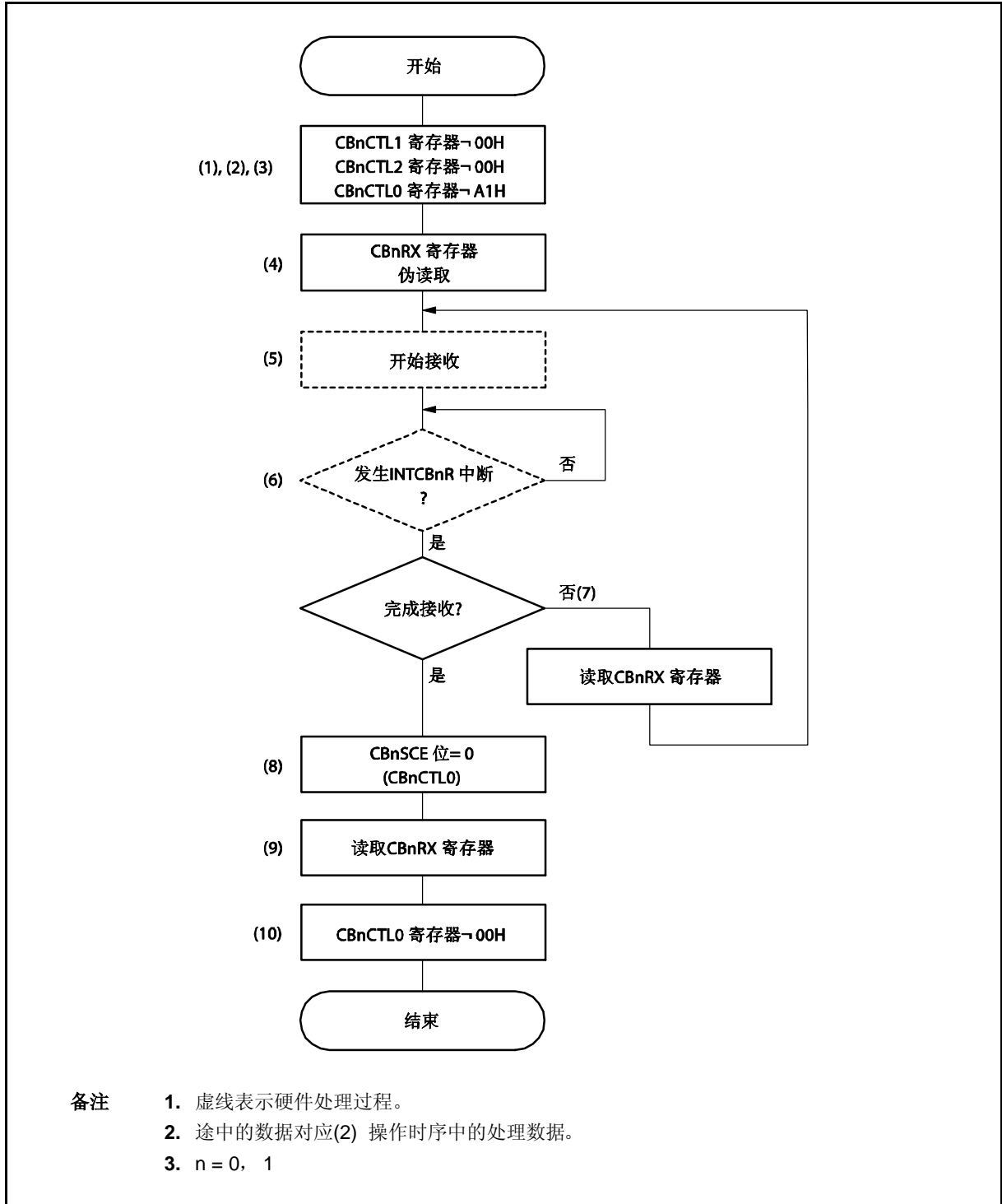
(2) 操作时序



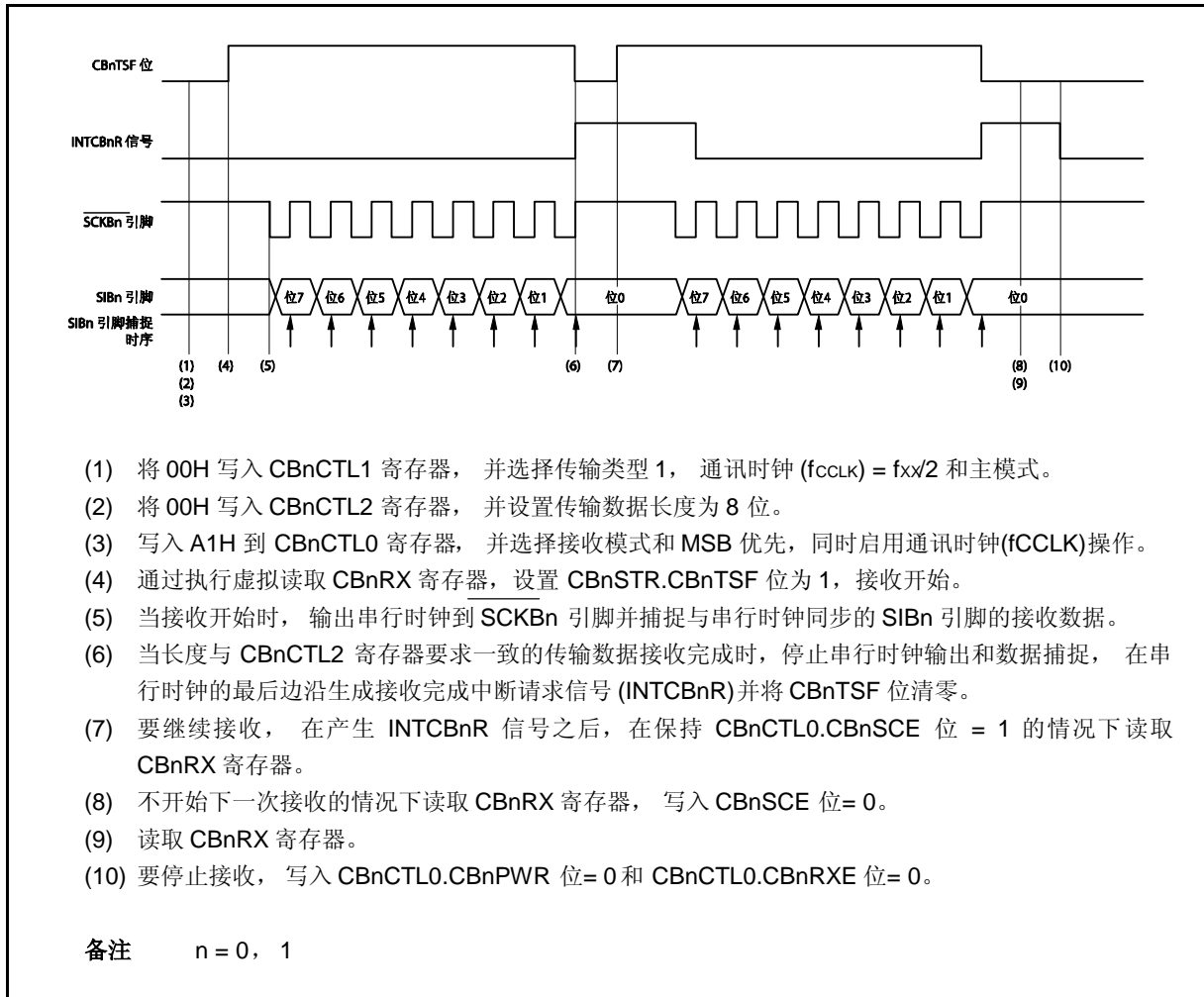
13.5.2 单传输模式（主模式，接收模式）

MSB 优先 (CBnCTL0.CBnDIR 位= 0)，传输类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通讯时钟 (f_{CCLK}) = f_{xx}/2 (CBnCTL1.CBnCKS2 ~ CBnCTL1.CBnCKS0 位 = 000)，传输数据长度 = 8 位 (CBnCTL2.CBnCL3 ~ CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



(2) 操作时序



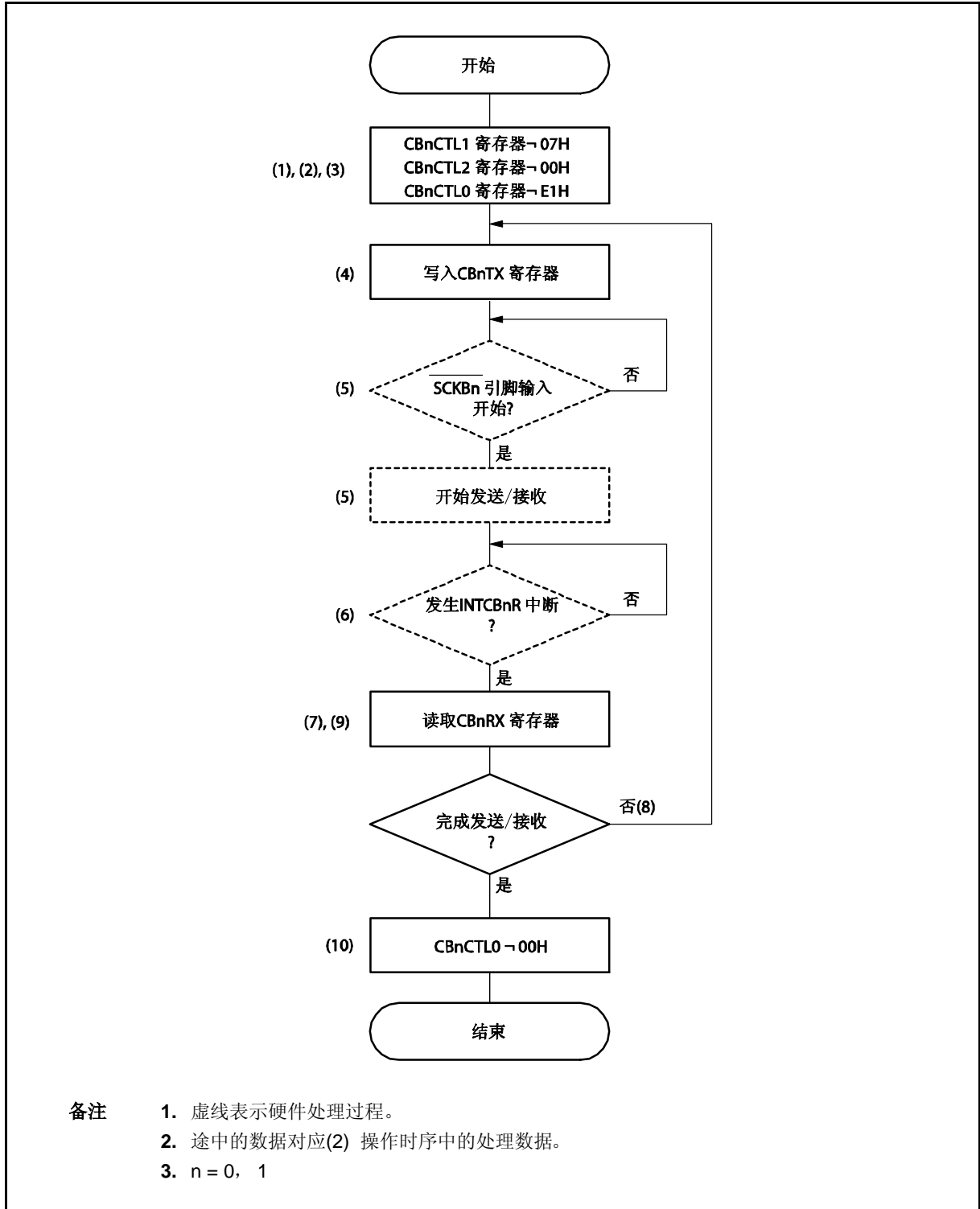
- (1) 将 00H 写入 CBnCTL1 寄存器，并选择传输类型 1，通讯时钟 (f_{CCLK}) = $f_x/2$ 和主模式。
- (2) 将 00H 写入 CBnCTL2 寄存器，并设置传输数据长度为 8 位。
- (3) 写入 A1H 到 CBnCTL0 寄存器，并选择接收模式和 MSB 优先，同时启用通讯时钟(f_{CCLK})操作。
- (4) 通过执行虚拟读取 CBnRX 寄存器，设置 CBnSTR.CBnTSF 位为 1，接收开始。
- (5) 当接收开始时，输出串行时钟到 SCKBn 引脚并捕捉与串行时钟同步的 SIBn 引脚的接收数据。
- (6) 当长度与 CBnCTL2 寄存器要求一致的传输数据接收完成时，停止串行时钟输出和数据捕捉，在串行时钟的最后边沿生成接收完成中断请求信号 (INTCBnR)并将 CBnTSF 位清零。
- (7) 要继续接收，在产生 INTCBnR 信号之后，在保持 CBnCTL0.CBnSCE 位 = 1 的情况下读取 CBnRX 寄存器。
- (8) 不开始下一次接收的情况下读取 CBnRX 寄存器，写入 CBnSCE 位= 0。
- (9) 读取 CBnRX 寄存器。
- (10) 要停止接收，写入 CBnCTL0.CBnPWR 位= 0 和 CBnCTL0.CBnRXE 位= 0。

备注 n = 0, 1

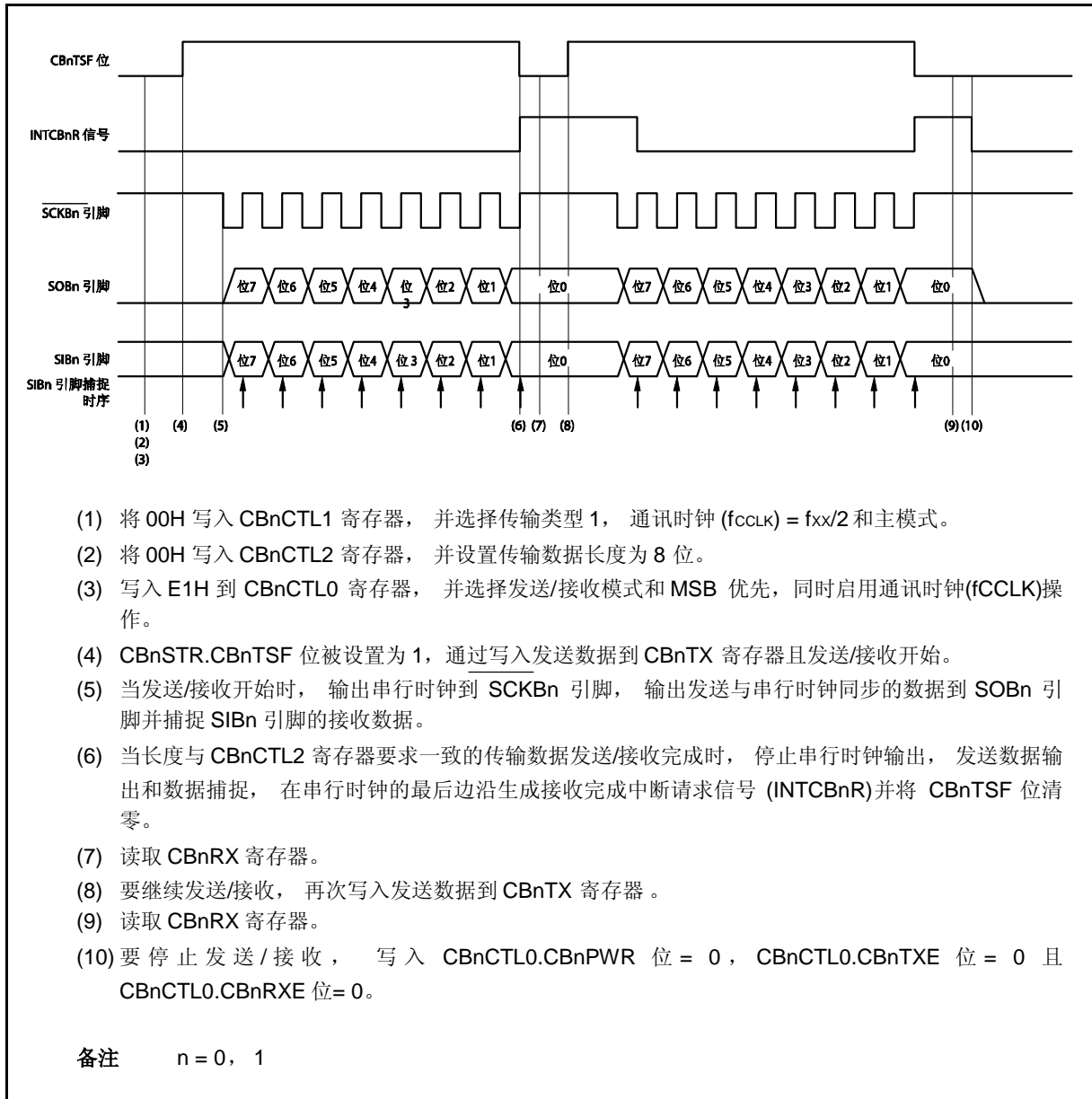
13.5.3 单传输模式（主模式，发送/接收模式）

MSB 优先 (CBnCTL0.CBnDIR 位= 0)，传输类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通讯时钟 (f_{CLK}) = f_{xx}/2 (CBnCTL1.CBnCKS2 ~ CBnCTL1.CBnCKS0 位 = 000)，传输数据长度 = 8 位 (CBnCTL2.CBnCL3 ~ CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



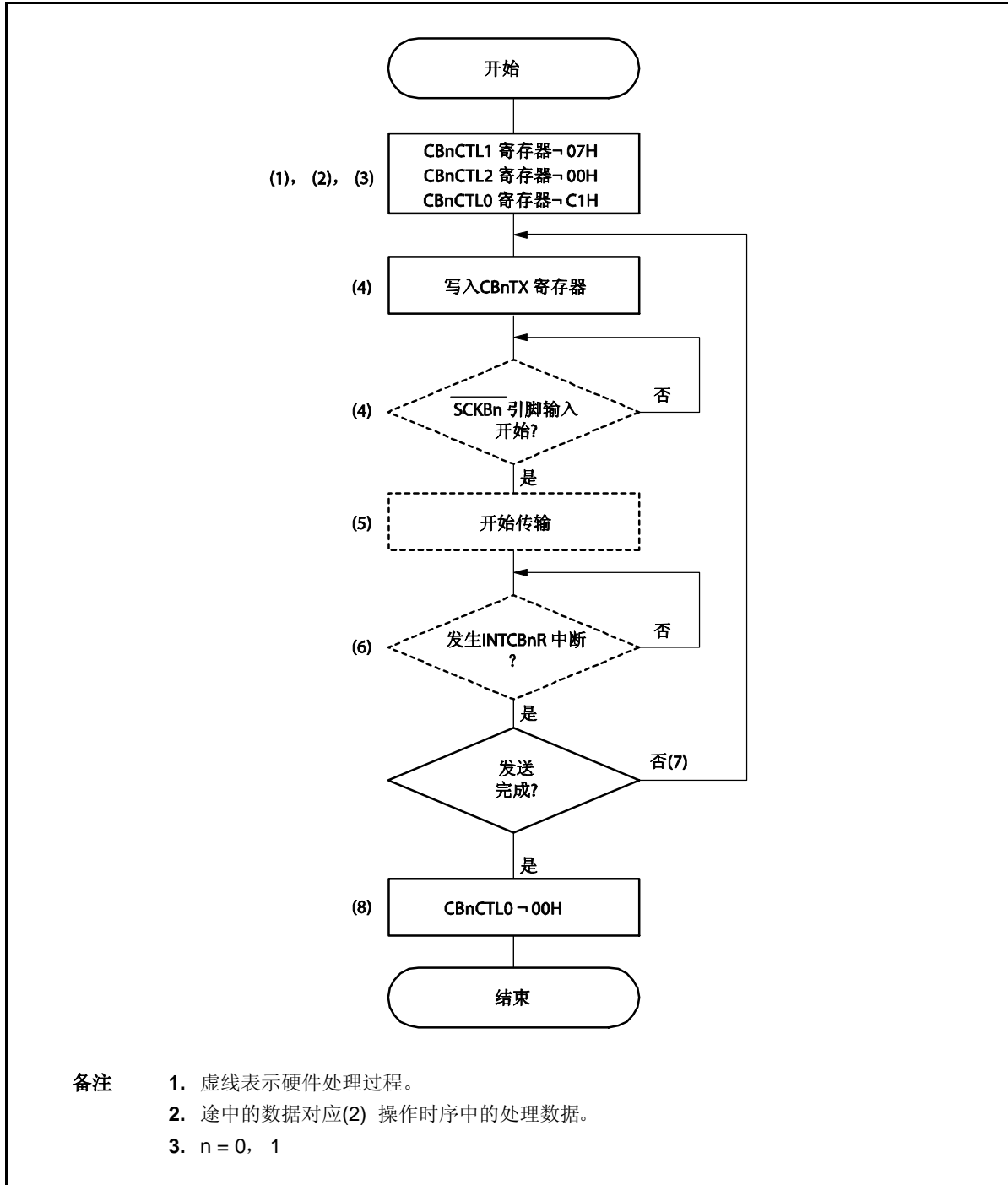
(2) 操作时序



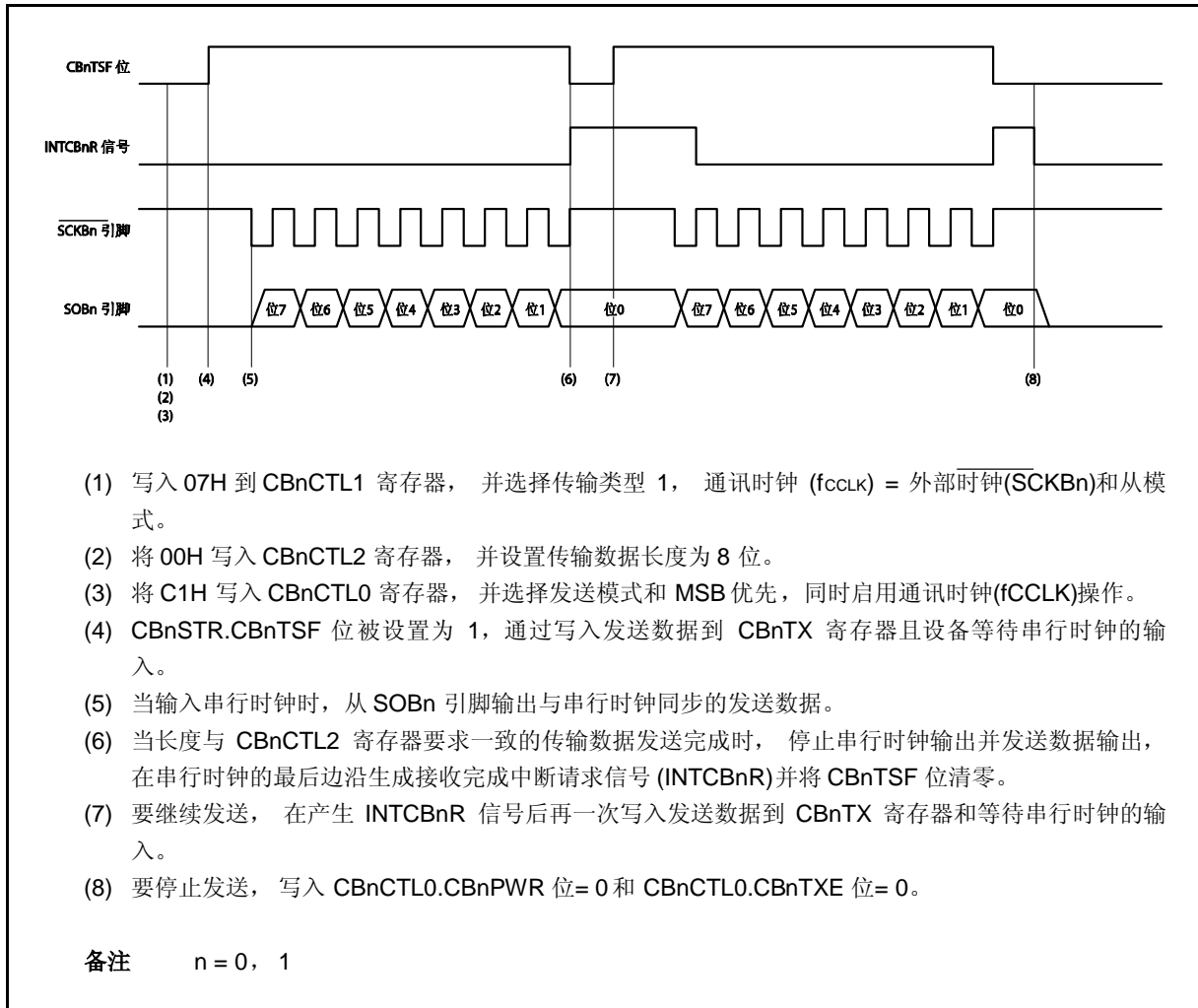
13.5.4 单传输模式（从模式，发送模式）

MSB 优先 (CBnCTL0.CBnDIR 位= 0)，传输类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通讯时钟 (f_{CLK}) = 外部时钟(SCKBn) (CBnCTL1.CBnCKS2 ~ CBnCTL1.CBnCKS0 位 = 111)，传输数据长度 = 8 位 (CBnCTL2.CBnCL3 ~ CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



(2) 操作时序



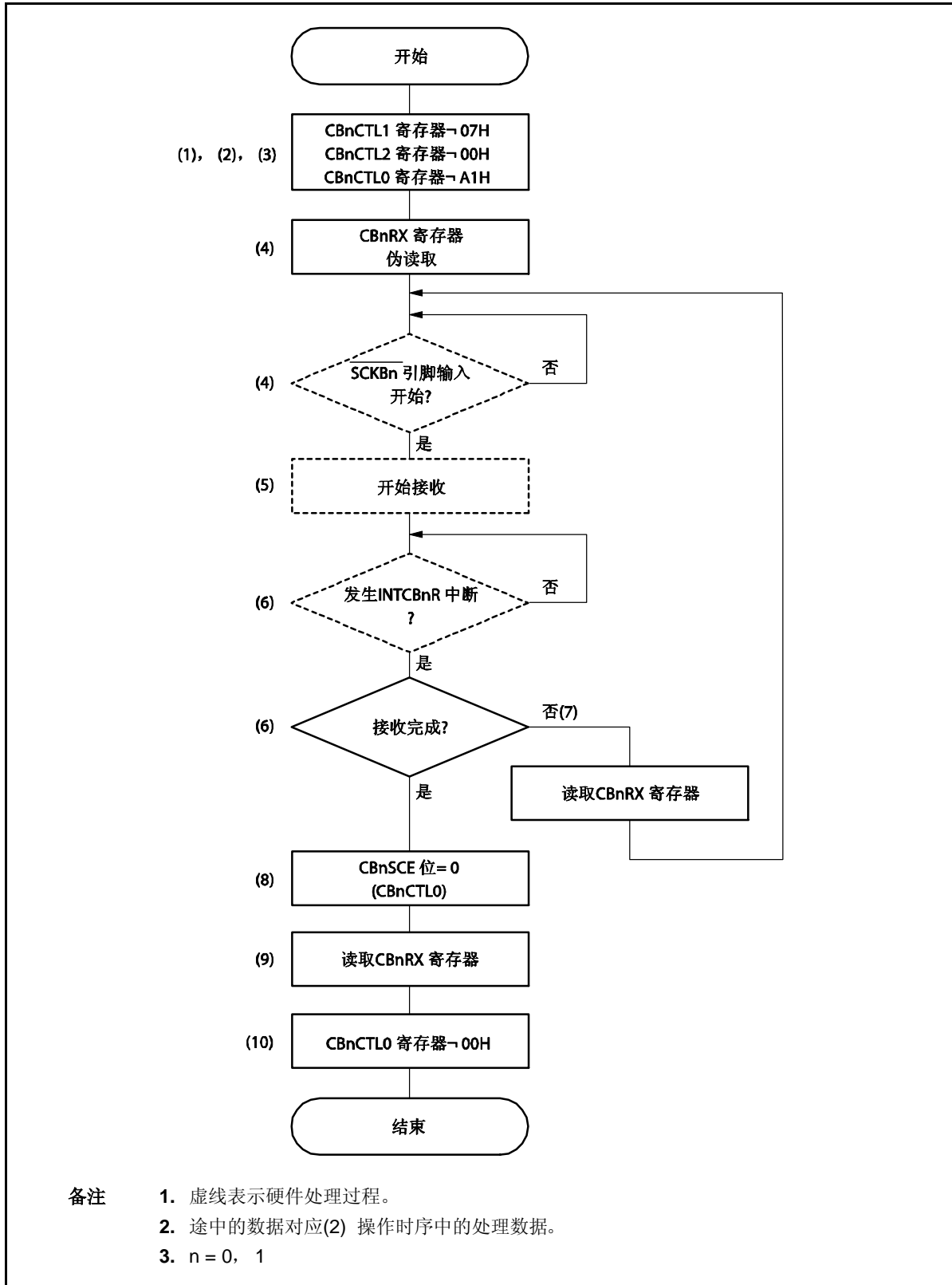
- (1) 写入 07H 到 CbNCTL1 寄存器， 并选择传输类型 1， 通讯时钟 (fcCLK) = 外部时钟(SCKbN)和从模式。
- (2) 将 00H 写入 CbNCTL2 寄存器， 并设置传输数据长度为 8 位。
- (3) 将 C1H 写入 CbNCTL0 寄存器， 并选择发送模式和 MSB 优先， 同时启用通讯时钟(fcCLK)操作。
- (4) CbNSTR.CbNtSF 位被设置为 1， 通过写入发送数据到 CbNTX 寄存器且设备等待串行时钟的输入。
- (5) 当输入串行时钟时， 从 SOBn 引脚输出与串行时钟同步的发送数据。
- (6) 当长度与 CbNCTL2 寄存器要求一致的传输数据发送完成时， 停止串行时钟输出并发送数据输出， 在串行时钟的最后边沿生成接收完成中断请求信号 (INTCbNR)并将 CbNtSF 位清零。
- (7) 要继续发送， 在产生 INTCbNR 信号后再一次写入发送数据到 CbNTX 寄存器和等待串行时钟的输入。
- (8) 要停止发送， 写入 CbNCTL0.CbNPWR 位= 0 和 CbNCTL0.CbNTXE 位= 0。

备注 n = 0, 1

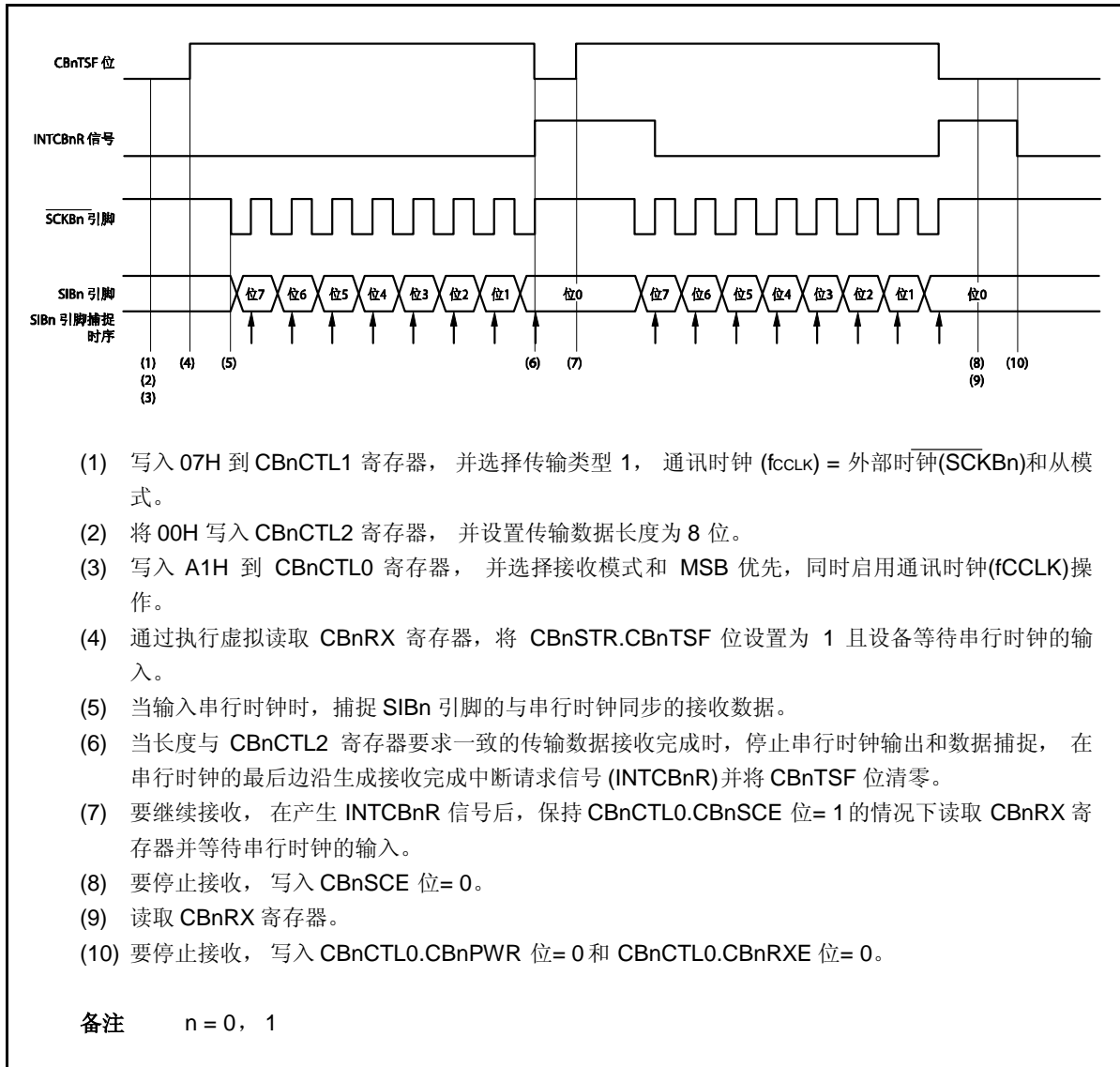
13.5.5 单传输模式（从模式，接收模式）

MSB 优先 (CBnCTL0.CBnDIR 位= 0)，传输类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通讯时钟 (f_{CLK}) = 外部时钟(SCKBn) (CBnCTL1.CBnCKS2 ~ CBnCTL1.CBnCKS0 位 = 111)，传输数据长度 = 8 位 (CBnCTL2.CBnCL3 ~ CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



(2) 操作时序



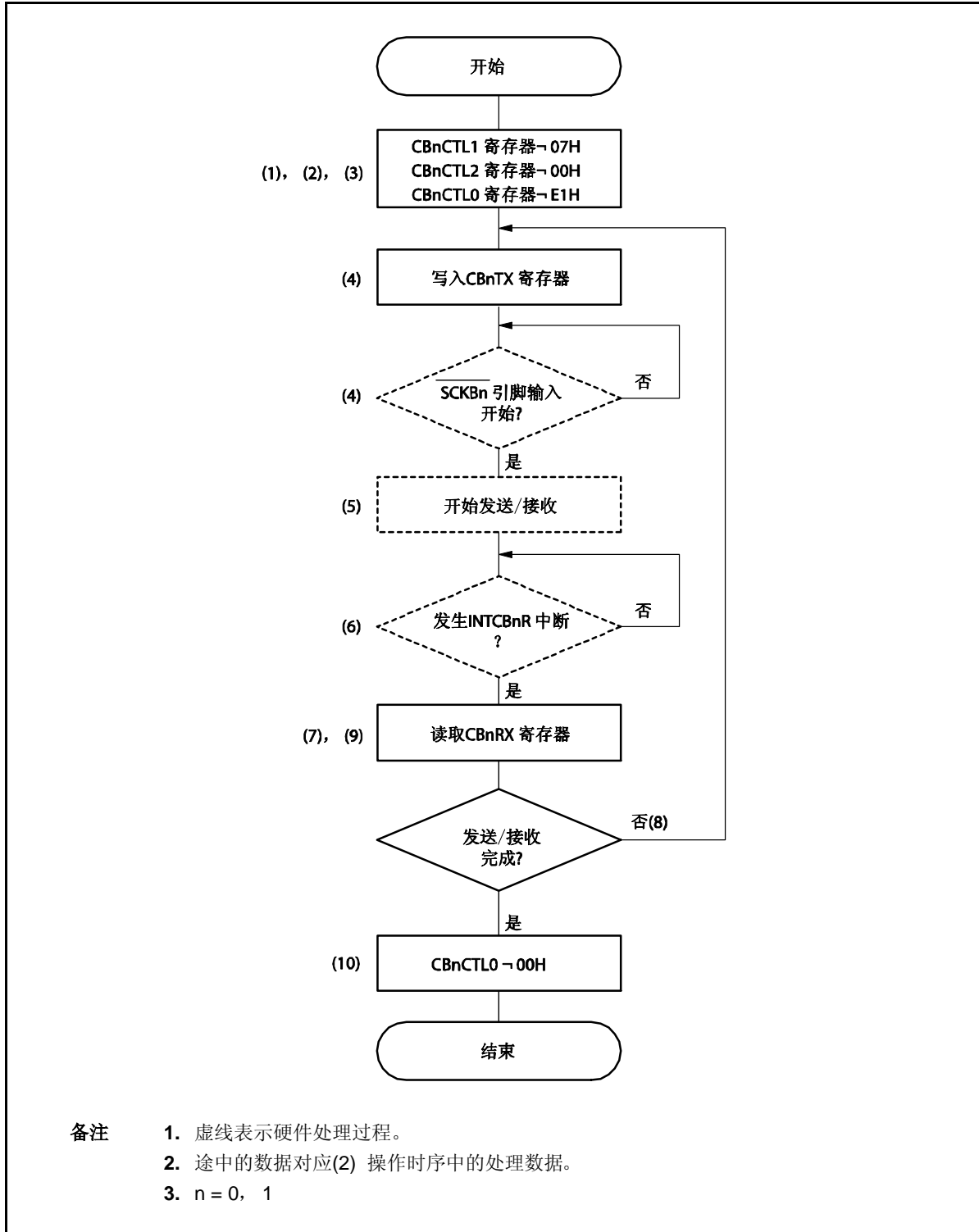
- (1) 写入 07H 到 CBnCTL1 寄存器，并选择传输类型 1，通讯时钟 (f_{CCLK}) = 外部时钟(SCKn)和从模式。
- (2) 将 00H 写入 CBnCTL2 寄存器，并设置传输数据长度为 8 位。
- (3) 写入 A1H 到 CBnCTL0 寄存器，并选择接收模式和 MSB 优先，同时启用通讯时钟(f_{CCLK})操作。
- (4) 通过执行虚拟读取 CBnRX 寄存器，将 CBnSTR.CBnTSP 位设置为 1 且设备等待串行时钟的输入。
- (5) 当输入串行时钟时，捕捉 SIBn 引脚的与串行时钟同步的接收数据。
- (6) 当长度与 CBnCTL2 寄存器要求一致的传输数据接收完成时，停止串行时钟输出和数据捕捉，在串行时钟的最后边沿生成接收完成中断请求信号 (INTCnR)并将 CBnTSP 位清零。
- (7) 要继续接收，在产生 INTCnR 信号后，保持 CBnCTL0.CBnSCE 位= 1 的情况下读取 CBnRX 寄存器并等待串行时钟的输入。
- (8) 要停止接收，写入 CBnSCE 位= 0。
- (9) 读取 CBnRX 寄存器。
- (10) 要停止接收，写入 CBnCTL0.CBnPWR 位= 0 和 CBnCTL0.CBnRXE 位= 0。

备注 n = 0, 1

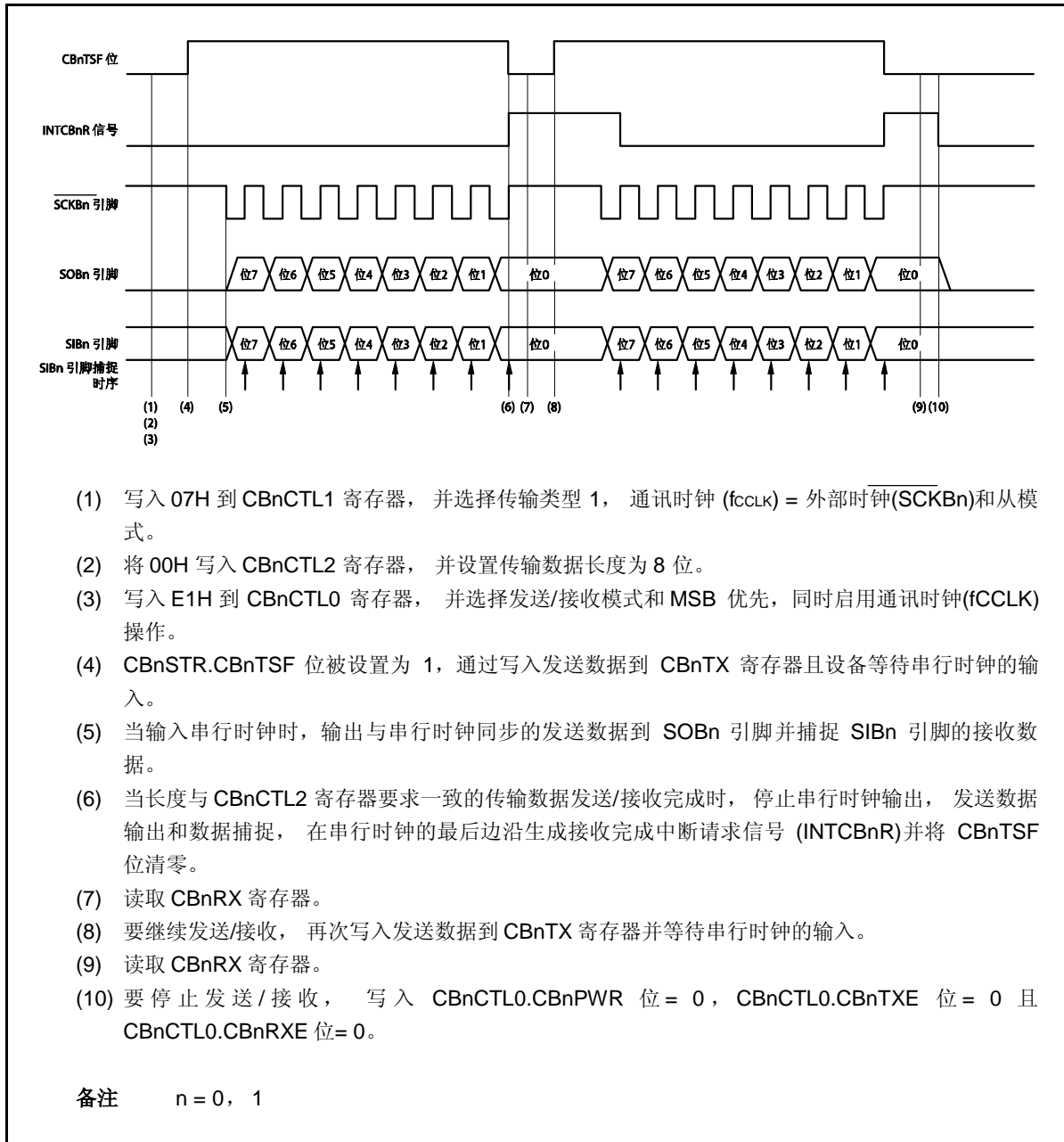
13.5.6 单传输模式（从模式，发送/接收模式）

MSB 优先 (CBnCTL0.CBnDIR 位= 0)，传输类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通讯时钟 (f_{CLK}) = 外部时钟(SCKBn) (CBnCTL1.CBnCKS2 ~ CBnCTL1.CBnCKS0 位 = 111)，传输数据长度 = 8 位 (CBnCTL2.CBnCL3 ~ CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



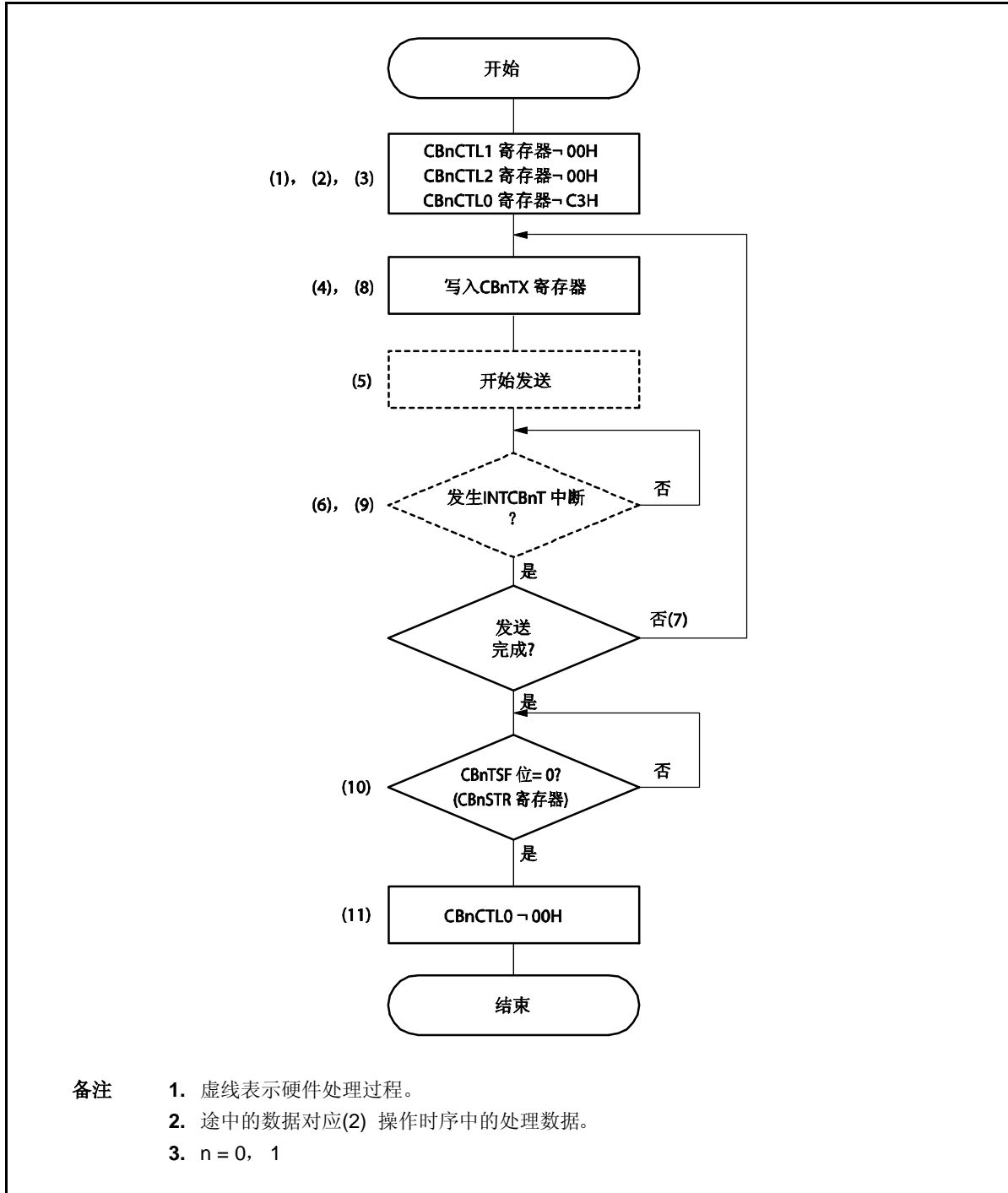
(2) 操作时序



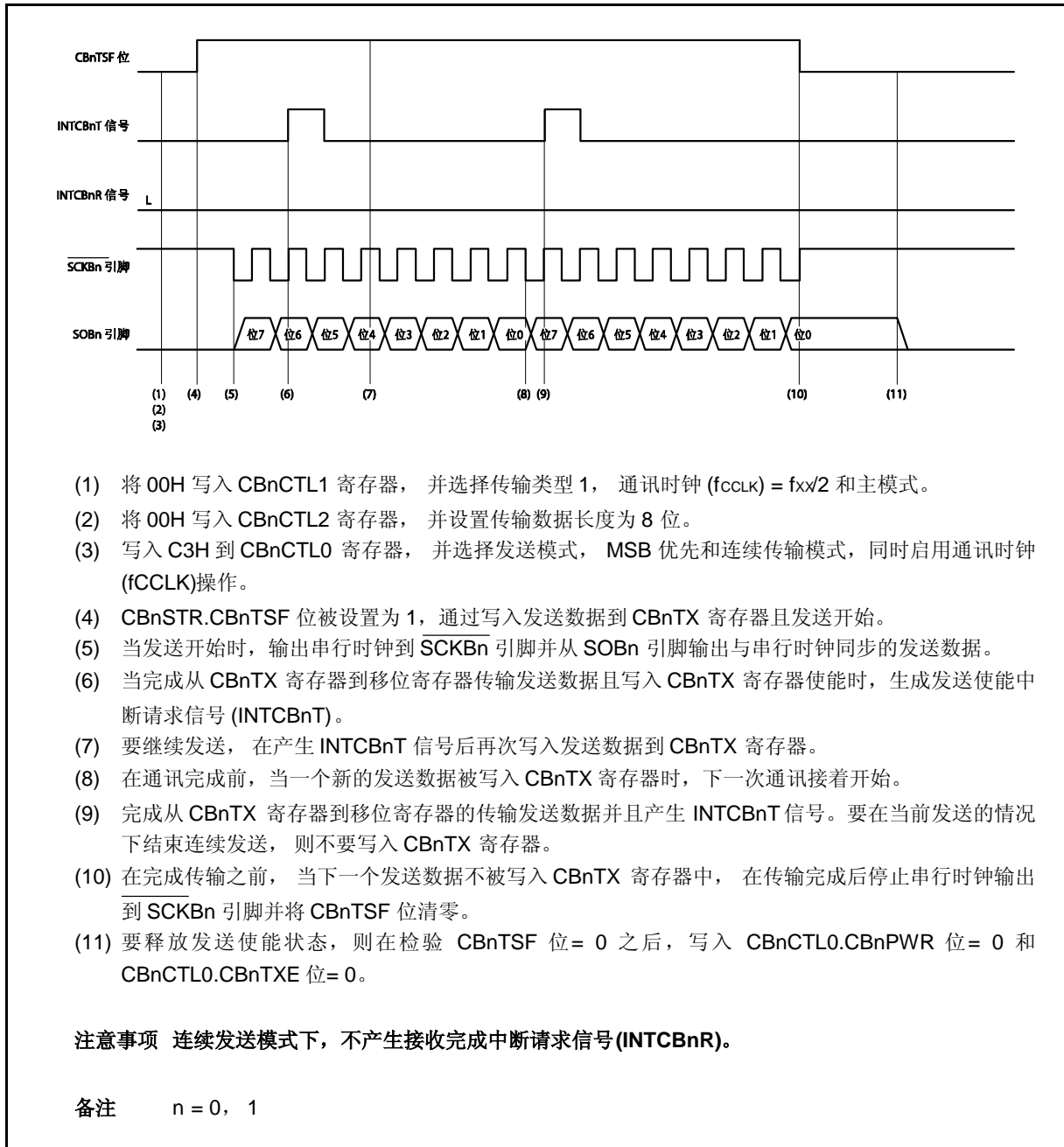
13.5.7 连续传输模式（主模式，发送模式）

MSB 优先 (CBnCTL0.CBnDIR 位= 0)，传输类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通讯时钟 (f_{CLK}) = f_{xx}/2 (CBnCTL1.CBnCKS2 ~ CBnCTL1.CBnCKS0 位 = 000)，传输数据长度 = 8 位 (CBnCTL2.CBnCL3 ~ CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



(2) 操作时序



- (1) 将 00H 写入 CBnCTL1 寄存器，并选择传输类型 1，通讯时钟 ($f_{\text{CCLK}} = f_x/2$) 和主模式。
- (2) 将 00H 写入 CBnCTL2 寄存器，并设置传输数据长度为 8 位。
- (3) 写入 C3H 到 CBnCTL0 寄存器，并选择发送模式，MSB 优先和连续传输模式，同时启用通讯时钟 (f_{CCLK}) 操作。
- (4) CBnSTR.CBnTSF 位被设置为 1，通过写入发送数据到 CBnTX 寄存器且发送开始。
- (5) 当发送开始时，输出串行时钟到 SCKBn 引脚并从 SOBn 引脚输出与串行时钟同步的发送数据。
- (6) 当完成从 CBnTX 寄存器到移位寄存器传输发送数据且写入 CBnTX 寄存器使能时，生成发送使能中断请求信号 (INTCBnT)。
- (7) 要继续发送，在产生 INTCBnT 信号后再次写入发送数据到 CBnTX 寄存器。
- (8) 在通讯完成前，当一个新的发送数据被写入 CBnTX 寄存器时，下一次通讯接着开始。
- (9) 完成从 CBnTX 寄存器到移位寄存器的传输发送数据并且产生 INTCBnT 信号。要在当前发送的情况下结束连续发送，则不要写入 CBnTX 寄存器。
- (10) 在完成传输之前，当下一个发送数据不被写入 CBnTX 寄存器中，在传输完成后停止串行时钟输出到 SCKBn 引脚并将 CBnTSF 位清零。
- (11) 要释放发送使能状态，则在检验 CBnTSF 位 = 0 之后，写入 CBnCTL0.CBnPWR 位 = 0 和 CBnCTL0.CBnTXE 位 = 0。

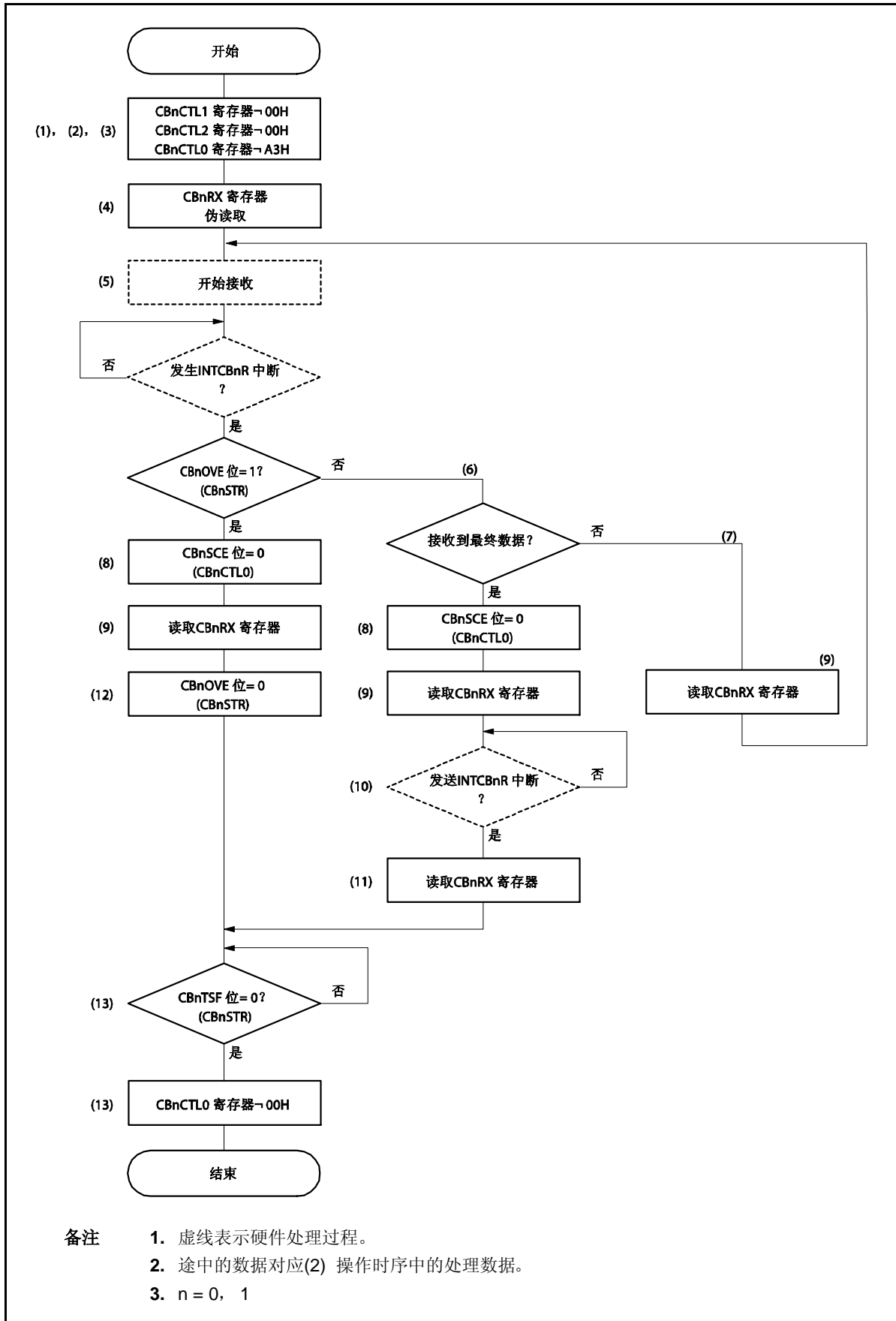
注意事项 连续发送模式下，不产生接收完成中断请求信号 (INTCBnR)。

备注 $n = 0, 1$

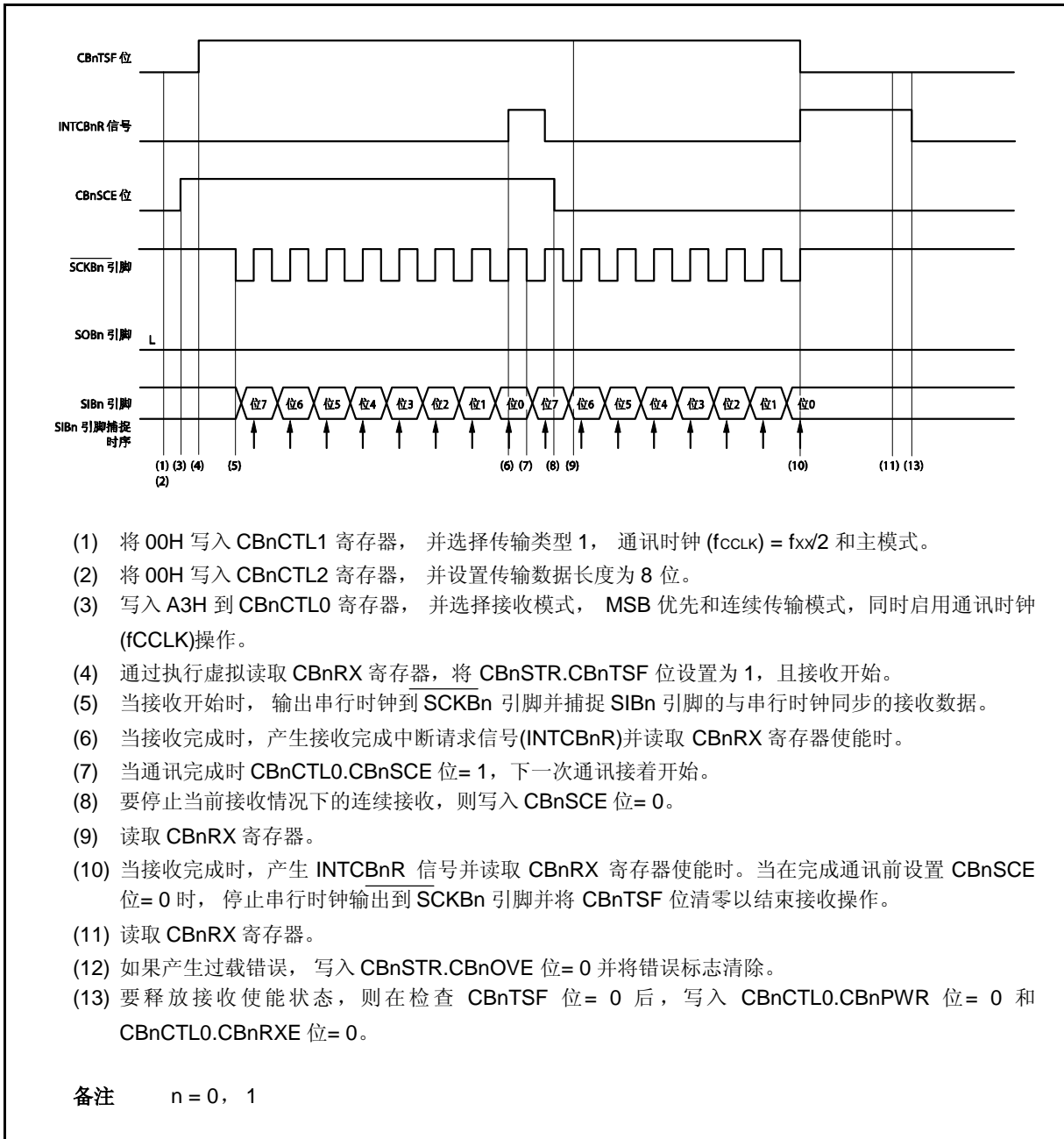
13.5.8 连续传输模式（主模式，接收模式）

MSB 优先 (CBnCTL0.CBnDIR 位= 0)，传输类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通讯时钟 (f_{CLK}) = $f_{\text{xx}}/2$ (CBnCTL1.CBnCKS2 ~ CBnCTL1.CBnCKS0 位 = 000)，传输数据长度 = 8 位 (CBnCTL2.CBnCL3 ~ CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



(2) 操作时序



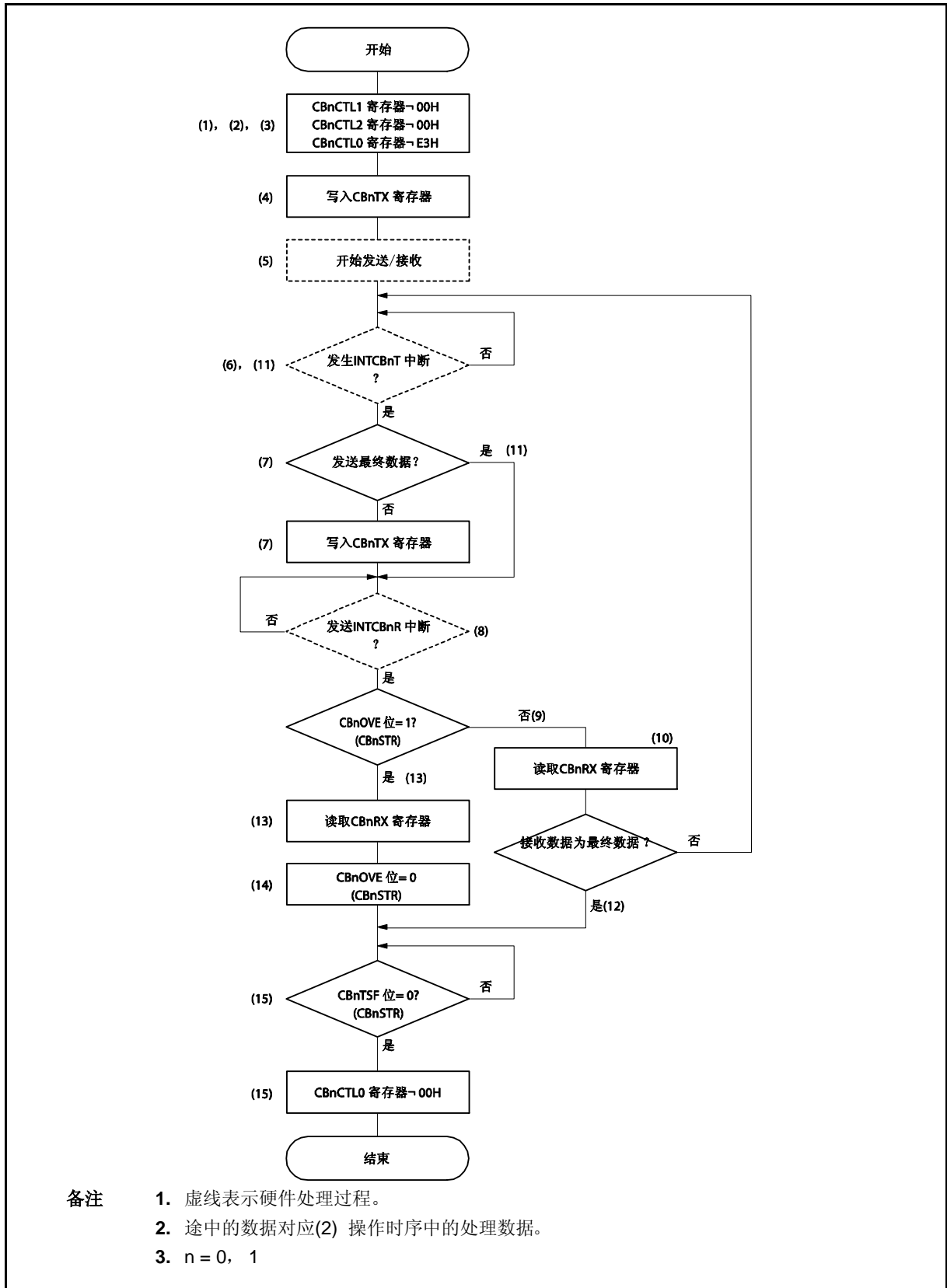
- (1) 将 00H 写入 CBnCTL1 寄存器，并选择传输类型 1，通讯时钟 (fCCLK) = f_x/2 和主模式。
- (2) 将 00H 写入 CBnCTL2 寄存器，并设置传输数据长度为 8 位。
- (3) 写入 A3H 到 CBnCTL0 寄存器，并选择接收模式，MSB 优先和连续传输模式，同时启用通讯时钟 (fCCLK)操作。
- (4) 通过执行虚拟读取 CBnRX 寄存器，将 CBnSTR.CBnTSF 位设置为 1，且接收开始。
- (5) 当接收开始时，输出串行时钟到 SCKBn 引脚并捕捉 SIBn 引脚的与串行时钟同步的接收数据。
- (6) 当接收完成时，产生接收完成中断请求信号(INTCBnR)并读取 CBnRX 寄存器使能时。
- (7) 当通讯完成时 CBnCTL0.CBnSCE 位= 1，下一次通讯接着开始。
- (8) 要停止当前接收情况下的连续接收，则写入 CBnSCE 位= 0。
- (9) 读取 CBnRX 寄存器。
- (10) 当接收完成时，产生 INTCBnR 信号并读取 CBnRX 寄存器使能时。当在完成通讯前设置 CBnSCE 位= 0 时，停止串行时钟输出到 SCKBn 引脚并将 CBnTSF 位清零以结束接收操作。
- (11) 读取 CBnRX 寄存器。
- (12) 如果产生过载错误，写入 CBnSTR.CBnOVE 位= 0 并将错误标志清除。
- (13) 要释放接收使能状态，则在检查 CBnTSF 位= 0 后，写入 CBnCTL0.CBnPWR 位= 0 和 CBnCTL0.CBnRXE 位= 0。

备注 n = 0, 1

13.5.9 连续传输模式（主模式，发送/接收模式）

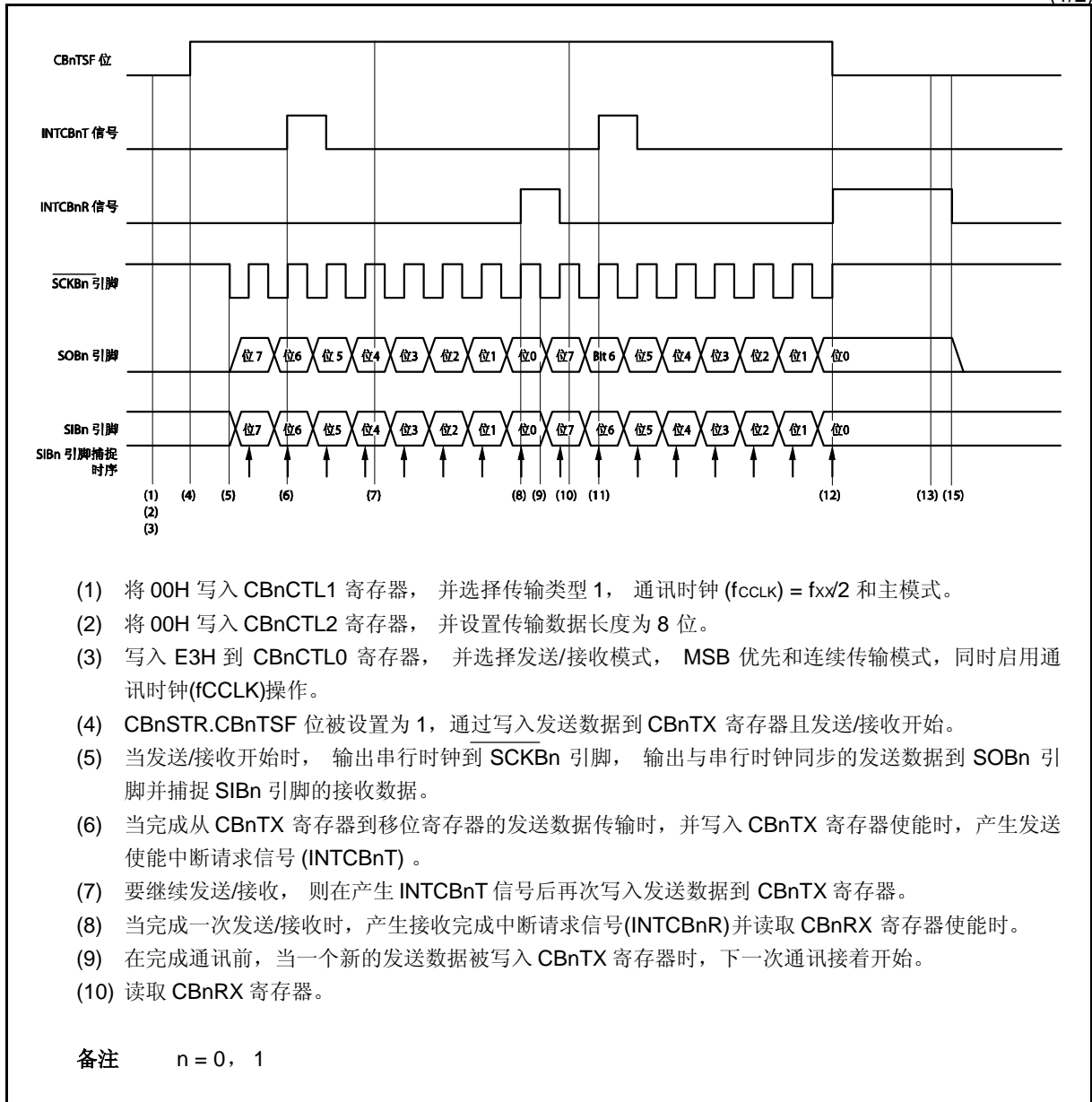
MSB 优先 (CBnCTL0.CBnDIR 位= 0)，传输类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通讯时钟 (f_{CLK}) = $f_{\text{xx}}/2$ (CBnCTL1.CBnCKS2 ~ CBnCTL1.CBnCKS0 位 = 000)，传输数据长度 = 8 位 (CBnCTL2.CBnCL3 ~ CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



(2) 操作时序

(1/2)



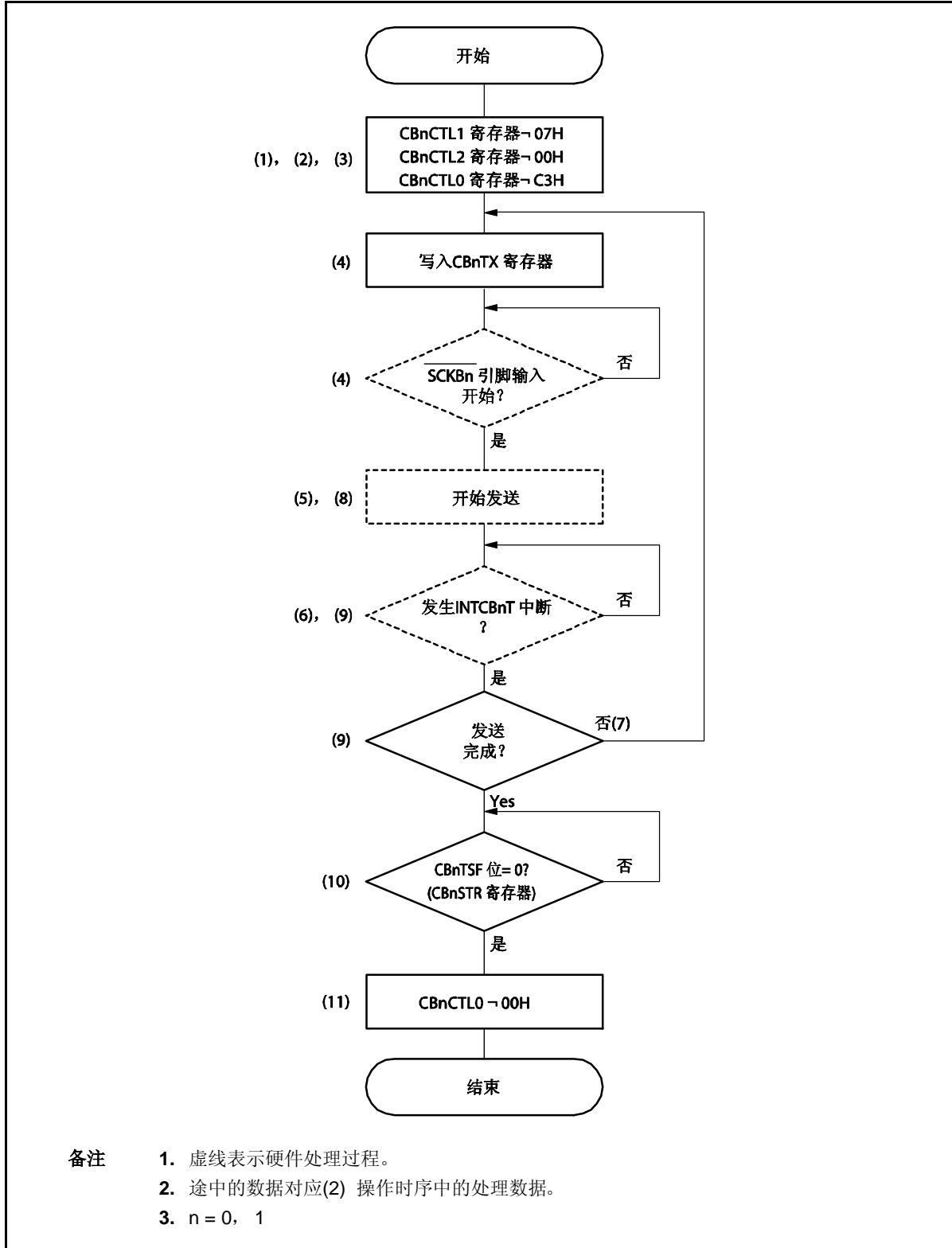
- (11) 完成从 CBnTX 寄存器到移位寄存器的发送数据的传输并生成 INTCBnT 信号。要结束当前发送/接收情况下的连续发送/接收，不要写入 CBnTX 寄存器。
- (12) 在完成传输前，当下一个发送数据不被写入 CBnTX 寄存器时，完成传输后停止串行时钟输出到 SCKBn 引脚并将 CBnTSF 位清零。
- (13) 当产生接收错误中断请求信号(INTCBnR)时，读取 CBnRX 寄存器。
- (14) 如果产生过载错误，写入 CBnSTR.CBnOVE 位= 0 并将错误标志清除。
- (15) 要释放发送/接收使能状态，则检查 CBnTSF 位= 0 之后，写入 CBnCTL0.CBnPWR 位= 0，CBnCTL0.CBnTXE 位= 0 且 CBnCTL0.CBnRXE 位= 0。

备注 n = 0, 1

13.5.10 连续传输模式 (从模式, 发送模式)

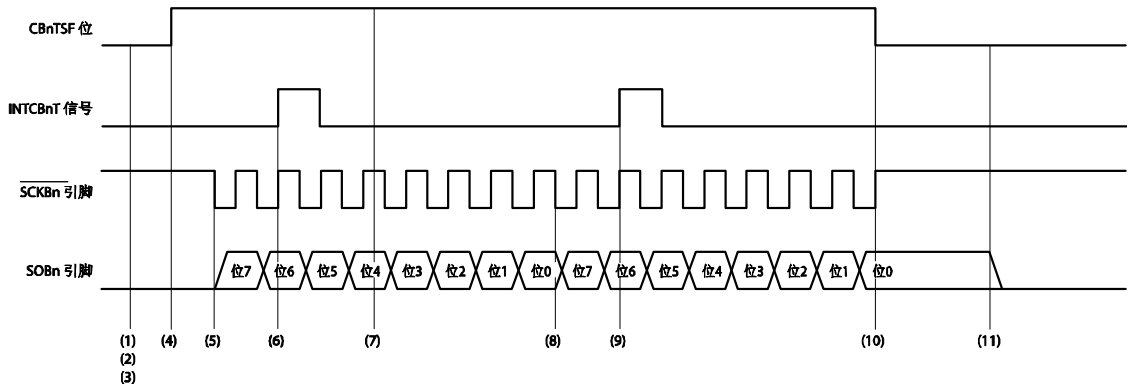
MSB 优先 (CBnCTL0.CBnDIR 位= 0), 传输类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通讯时钟 (f_{CCLK}) = 外部时钟(SCKBn) (CBnCTL1.CBnCKS2 ~ CBnCTL1.CBnCKS0 位 = 111), 传输数据长度 = 8 位 (CBnCTL2.CBnCL3 ~ CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



- 备注
1. 虚线表示硬件处理过程。
 2. 途中的数据对应(2) 操作时序中的处理数据。
 3. n = 0, 1

(2) 操作时序



- (1) 写入 07H 到 CBnCTL1 寄存器，并选择传输类型 1，通讯时钟 (f_{CLK}) = 外部时钟(SCKBn)和从模式。
- (2) 将 00H 写入 CBnCTL2 寄存器，并设置传输数据长度为 8 位。
- (3) 写入 C3H 到 CBnCTL0 寄存器，并选择发送模式，MSB 优先和连续传输模式，同时启用通讯时钟 (f_{CLK})操作。
- (4) CBnSTR.CBnTSF 位被设置为 1，通过写入发送数据到 CBnTX 寄存器且设备等待串行时钟的输入。
- (5) 当输入串行时钟时，从 SOBn 引脚输出与串行时钟同步的发送数据。
- (6) 当完成从 CBnTX 寄存器到移位寄存器的发送数据传输时，并写入 CBnTX 寄存器使能时，产生发送使能中断请求信号 (INTCBnT)。
- (7) 要继续发送，在产生 INTCBnT 信号之后，再次写入发送数据到 CBnTX 寄存器。
- (8) 当输入串行时钟时，在长度与 CBnCTL2 寄存器要求一直的传输数据发送完成之后，开始连续发送。
- (9) 当完成从 CBnTX 寄存器到移位寄存器的发送数据传输时，并写入 CBnTX 寄存器使能时，产生 INTCBnT 信号。要停止当前发送情况下的连续发送，则不要写入 CBnTX 寄存器。
- (10) 当与 CBnCTL2 寄存器一起设置的传输数据长度时钟没有写入到 CBnTX 寄存器，将 CBnTSF 位清零以结束发送。
- (11) 要释放发送使能状态，则在检查 CBnTSF 位 = 0 之后，写入 CBnCTL0.CBnPWR 位 = 0 和 CBnCTL0.CBnTXE 位 = 0。

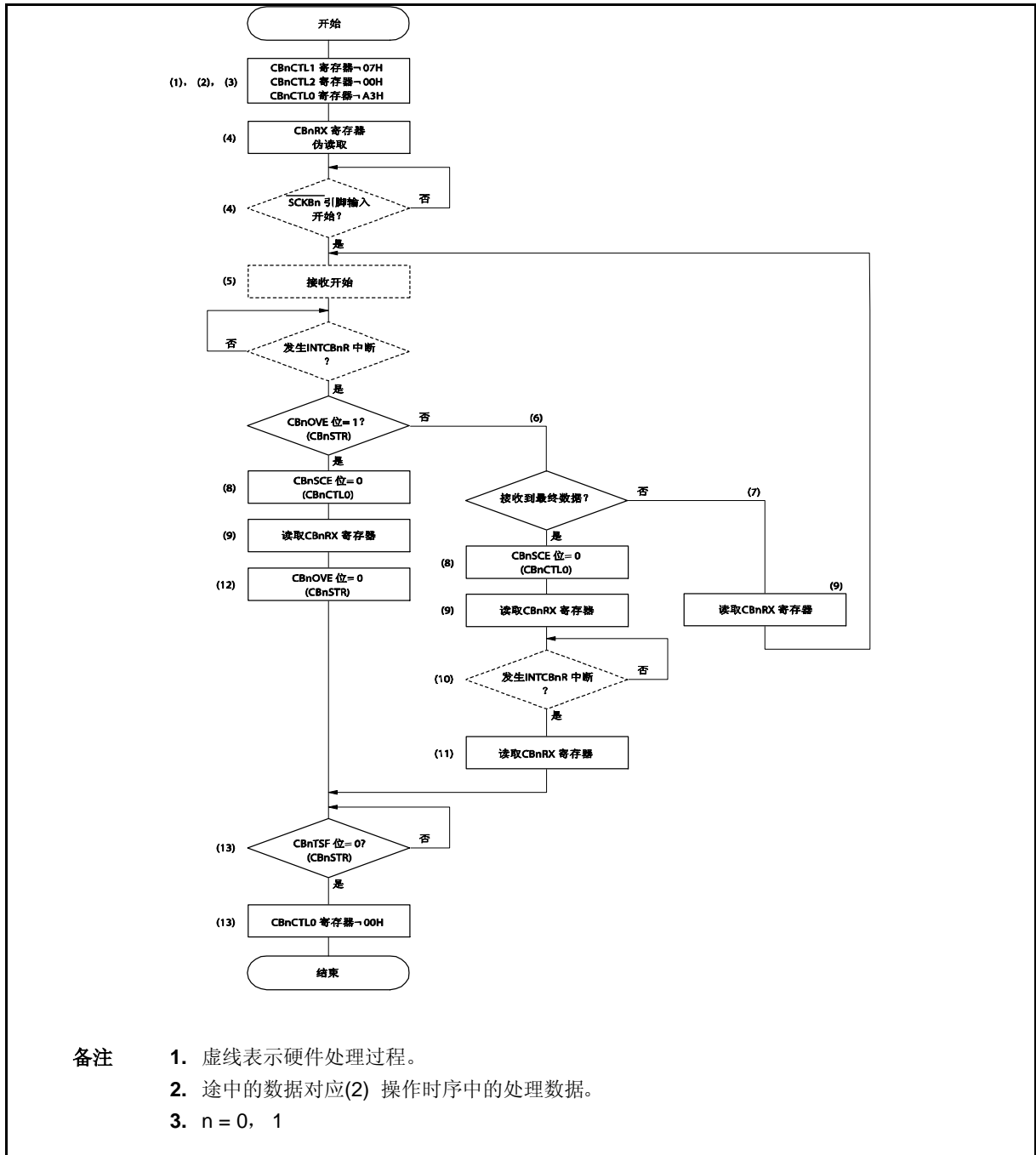
注意事项 连续发送模式下，不产生接收完成中断请求信号 (INTCBnR)。

备注 n = 0, 1

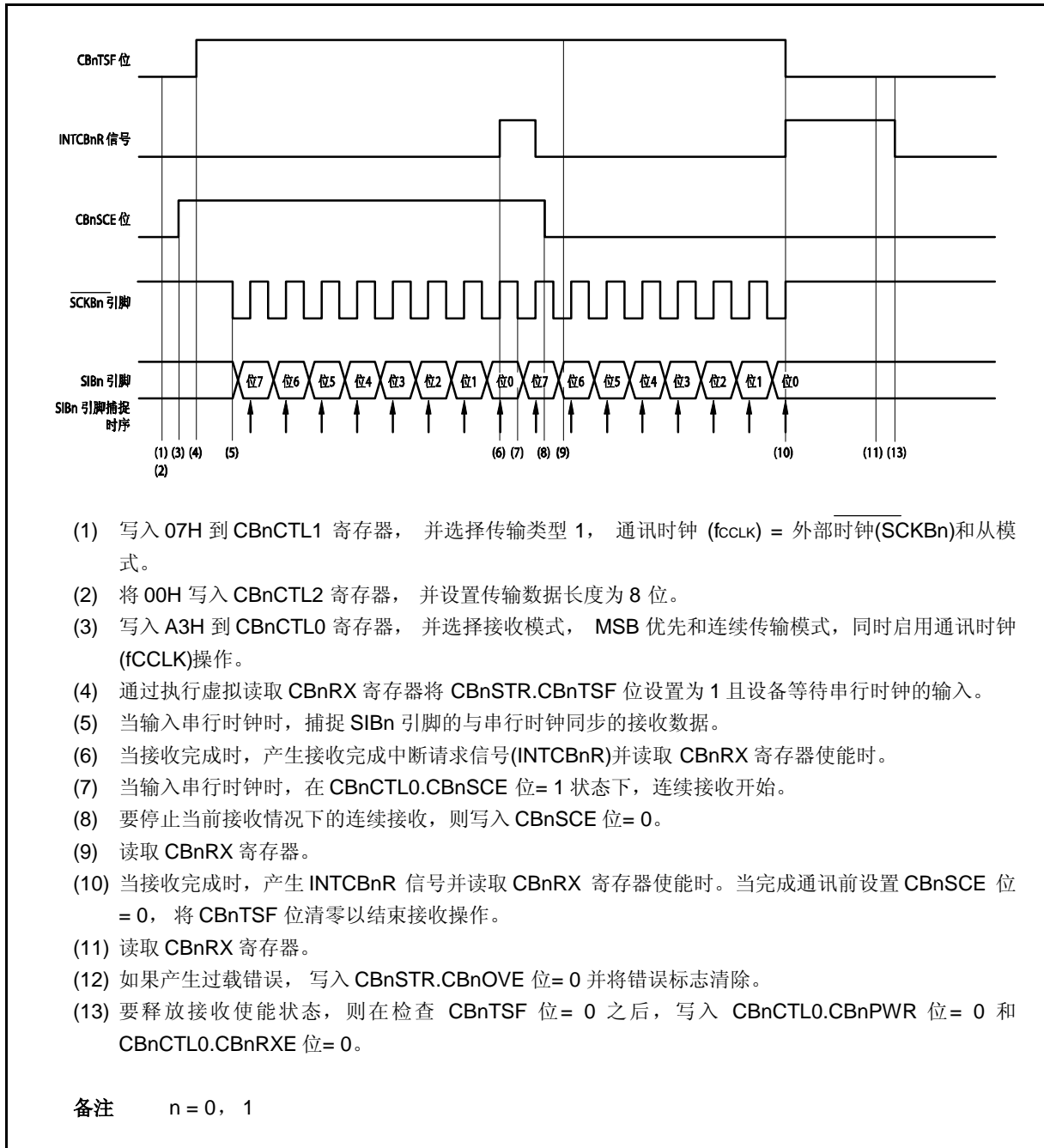
13.5.11 连续传输模式（从模式，接收模式）

MSB 优先 (CBnCTL0.CBnDIR 位= 0)，传输类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通讯时钟 (f_{CLK}) = 外部时钟(SCK_{Bn}) (CBnCTL1.CBnCKS2 ~ CBnCTL1.CBnCKS0 位 = 111)，传输数据长度 = 8 位 (CBnCTL2.CBnCL3 ~ CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



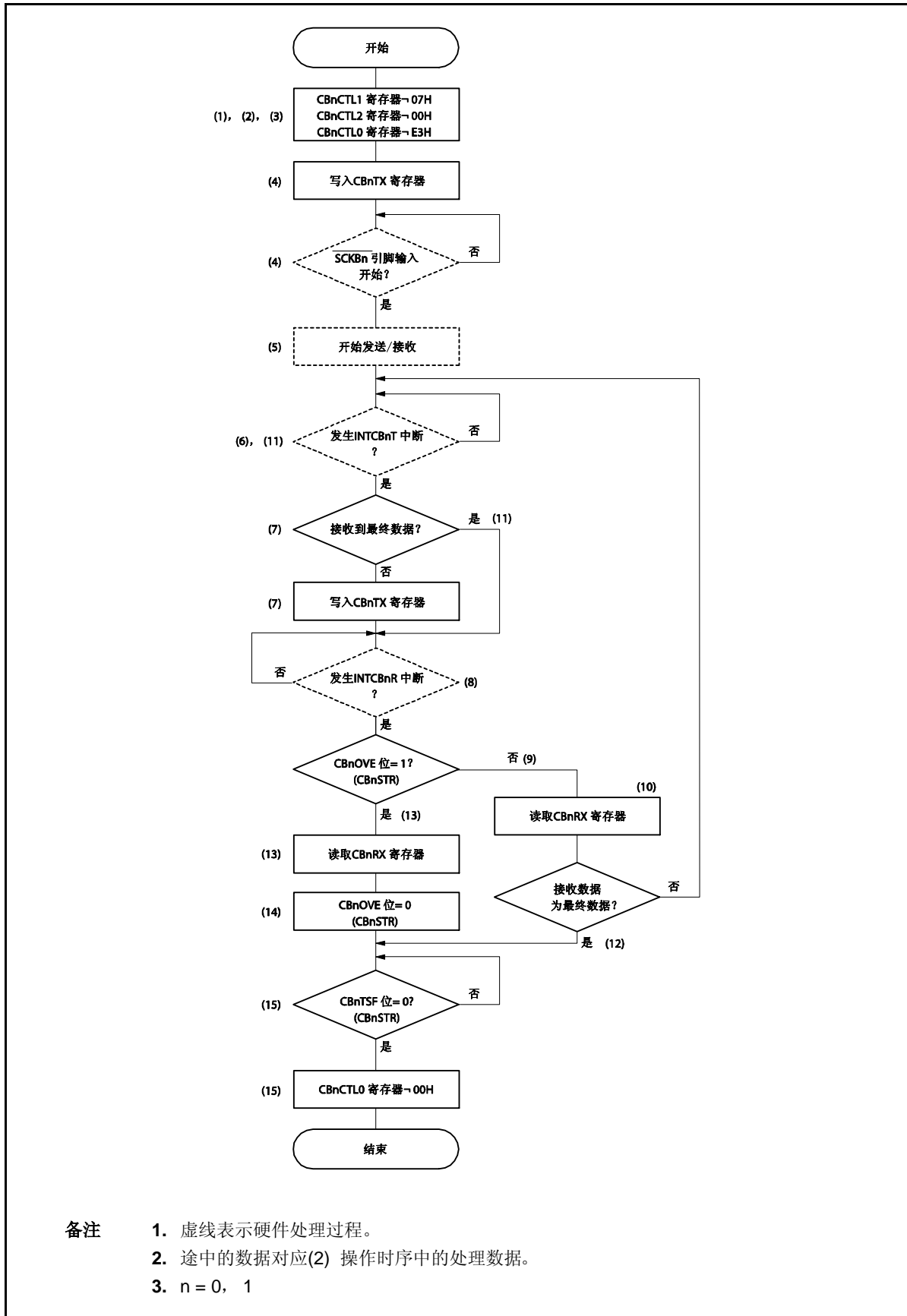
(2) 操作时序



13.5.12 连续传输模式（从模式，发送/接收模式）

MSB 优先 (CBnCTL0.CBnDIR 位= 0)，传输类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通讯时钟 (f_{CLK}) = 外部时钟(SCK_{Bn}) (CBnCTL1.CBnCKS2 ~ CBnCTL1.CBnCKS0 位 = 111)，传输数据长度 = 8 位 (CBnCTL2.CBnCL3 ~ CBnCTL2.CBnCL0 位 = 0000)

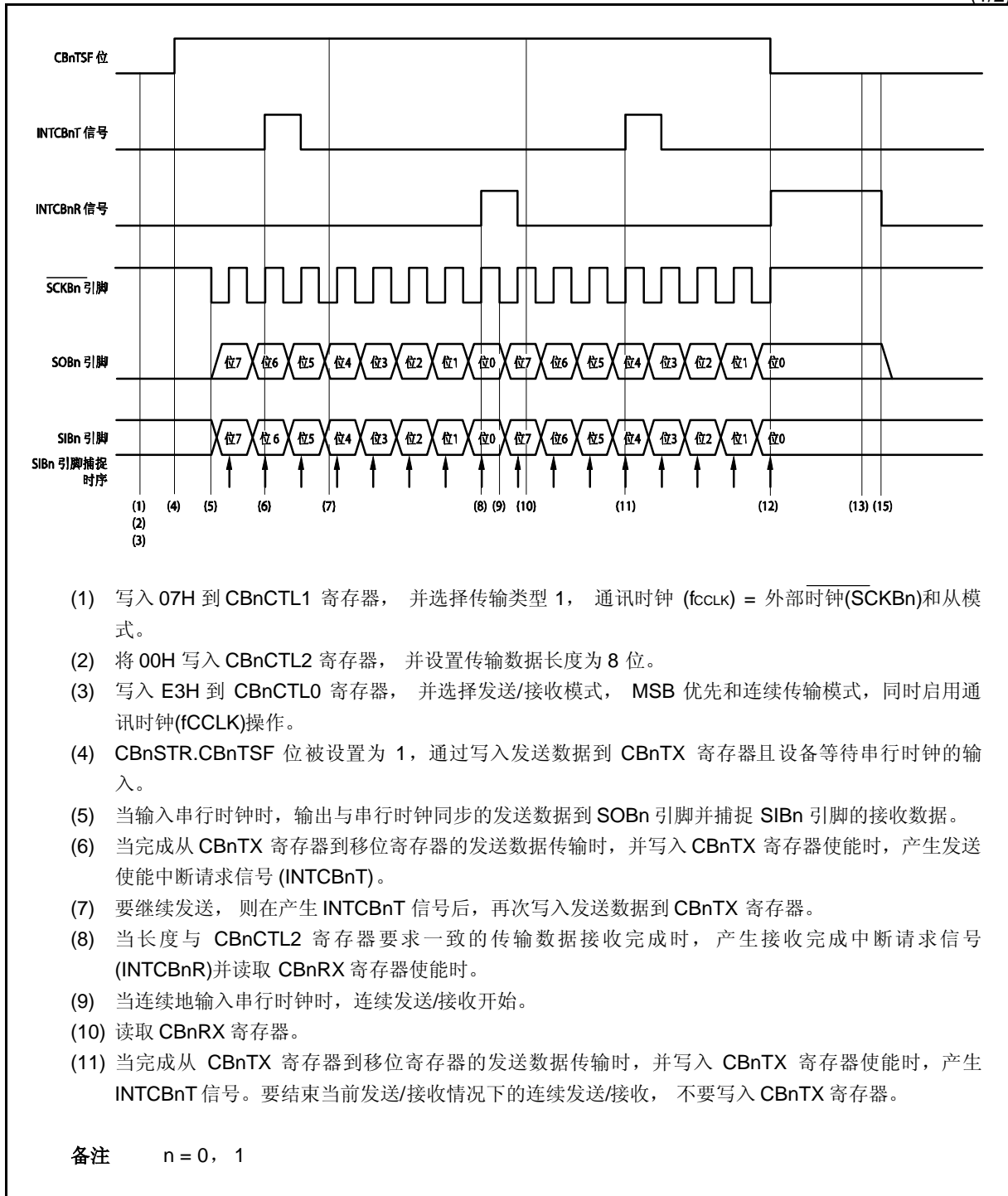
(1) 操作流程



- 备注
1. 虚线表示硬件处理过程。
 2. 途中的数据对应(2) 操作时序中的处理数据。
 3. n = 0, 1

(2) 操作时序

(1/2)



- (1) 写入 07H 到 **CBnCTL1** 寄存器， 并选择传输类型 1， 通讯时钟 (f_{cCLK}) = 外部时钟(**SCKBn**)和从模式。
- (2) 将 00H 写入 **CBnCTL2** 寄存器， 并设置传输数据长度为 8 位。
- (3) 写入 E3H 到 **CBnCTL0** 寄存器， 并选择发送/接收模式， **MSB** 优先和连续传输模式， 同时启用通讯时钟(f_{cCLK})操作。
- (4) **CBnSTR.CBnTSF** 位被设置为 1， 通过写入发送数据到 **CBnTX** 寄存器且设备等待串行时钟的输入。
- (5) 当输入串行时钟时， 输出与串行时钟同步的发送数据到 **SOBn** 引脚并捕捉 **SIBn** 引脚的接收数据。
- (6) 当完成从 **CBnTX** 寄存器到移位寄存器的发送数据传输时， 并写入 **CBnTX** 寄存器使能时， 产生发送使能中断请求信号 (**INTCBnT**)。
- (7) 要继续发送， 则在产生 **INTCBnT** 信号后， 再次写入发送数据到 **CBnTX** 寄存器。
- (8) 当长度与 **CBnCTL2** 寄存器要求一致的传输数据接收完成时， 产生接收完成中断请求信号 (**INTCBnR**)并读取 **CBnRX** 寄存器使能时。
- (9) 当连续地输入串行时钟时， 连续发送/接收开始。
- (10) 读取 **CBnRX** 寄存器。
- (11) 当完成从 **CBnTX** 寄存器到移位寄存器的发送数据传输时， 并写入 **CBnTX** 寄存器使能时， 产生 **INTCBnT** 信号。要结束当前发送/接收情况下的连续发送/接收， 不要写入 **CBnTX** 寄存器。

备注 $n = 0, 1$

- (12) 当与 CBnCTL2 寄存器一起设置的传输数据长度时钟没有写入到 CBnTX 寄存器，产生 INTCBnR 信号。将 CBnTSF 位清零以停止发送/接收。
- (13) 当产生 INTCBnR 信号时，读取 CBnRX 寄存器。
- (14) 如果产生过载错误，写入 CBnSTR.CBnOVE 位= 0 并将错误标志清除。
- (15) 要释放发送/接收使能状态，则在检查 CBnTSF 位= 0 后，写入 CBnCTL0.CBnPWR 位= 0，CBnCTL0.CBnTXE 位= 0 且 CBnCTL0.CBnRXE 位= 0。

备注 n = 0, 1

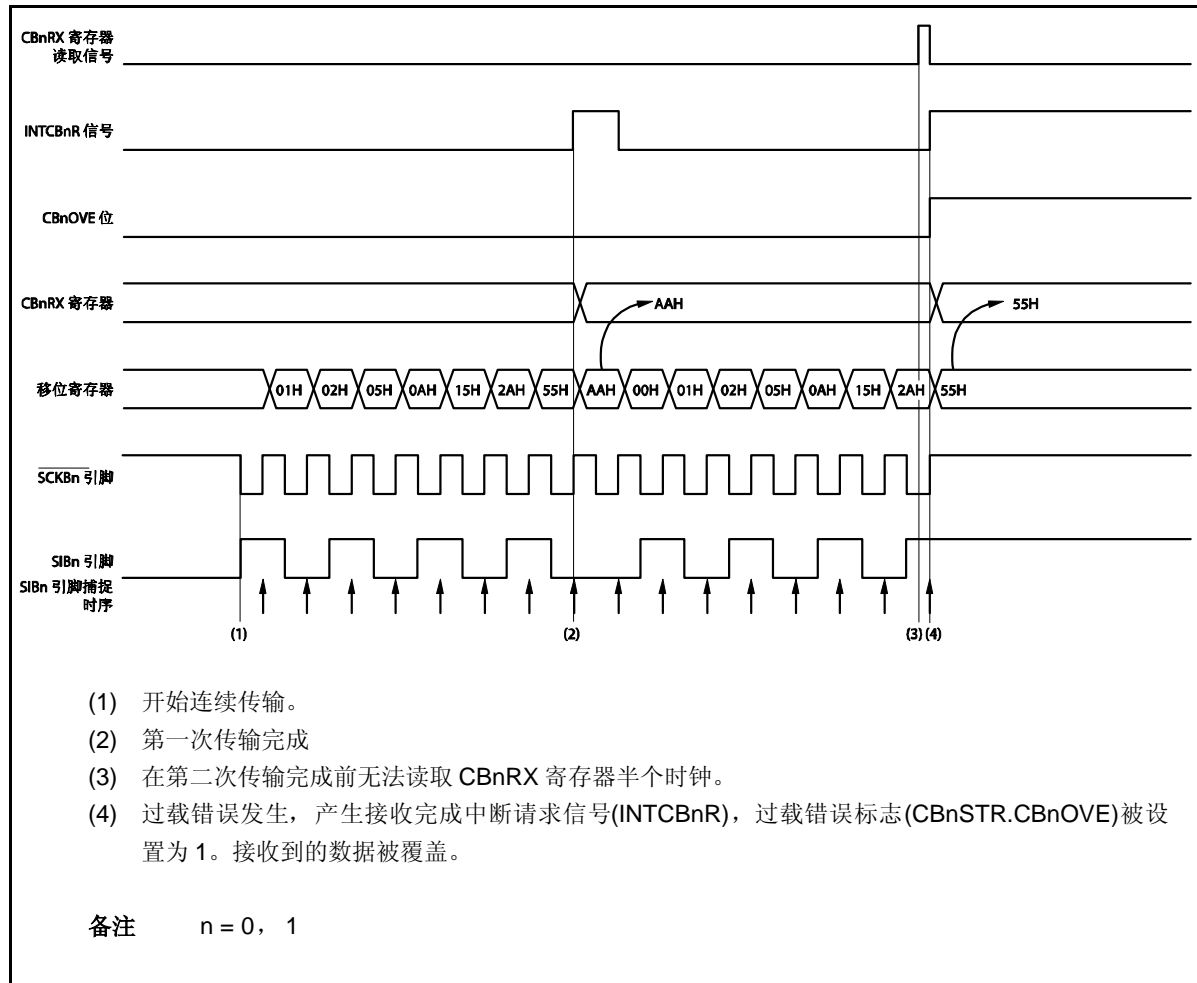
13.5.13 接收错误

在连续传输模式接收启用 (CBnCTL0.CBnRXE 位为 1) 时执行传输, 当下一次接收操作完成在 CBnRX 寄存器读取前与接收完成中断请求信号 (INTCBnR) 产生后之间时, 接收完成中断请求信号 (INTCBnR) 会再次产生, 且过载错误标志 (CBnSTR.CBnOVE) 被设置为 1。

当过载错误发生时, 前一次接收的数据会因 CBnRX 寄存器被更新而丢失。过载错误发生时, 如果下一次接收完成时 CBnRX 寄存器未被读取, 接收完成中断请求信号 (INTCBnR) 会再次产生。

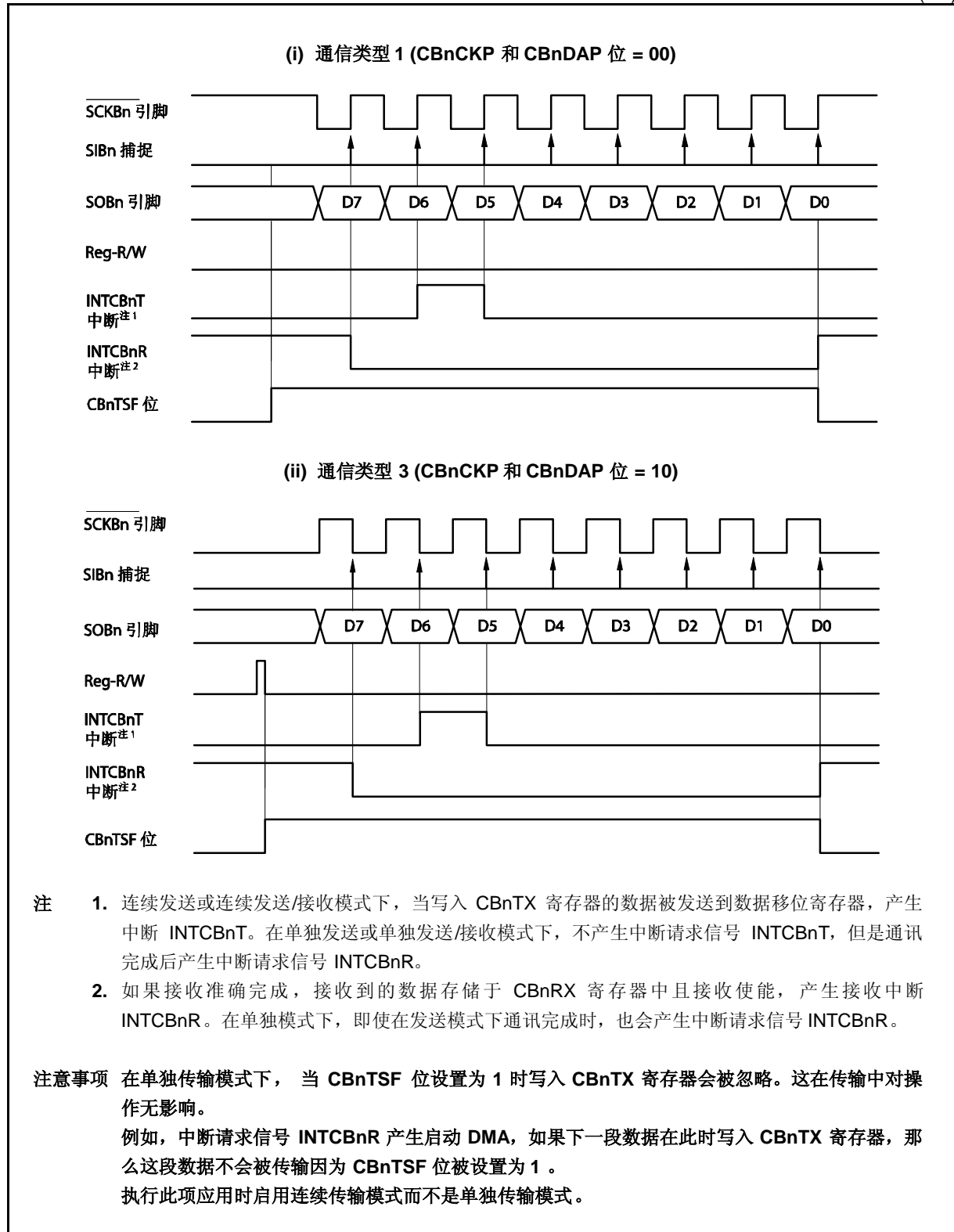
避免发生过载错误, 在从 INTCBnR 信号发生器取样下一次接收数据的是最后位前完整读取 CBnRX 寄存器半个时钟。

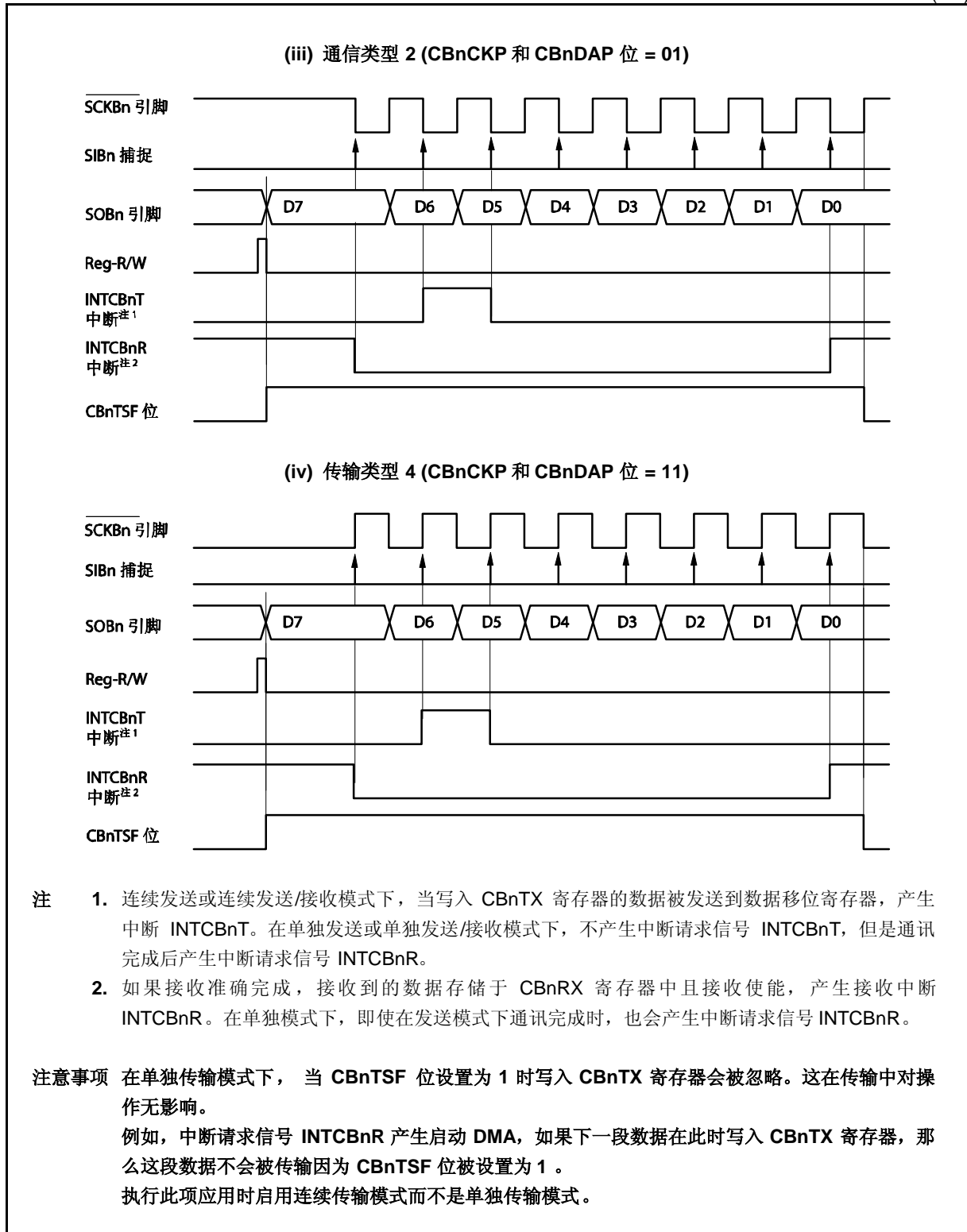
(1) 操作时序



13.5.14 时钟时序

(1/2)





13.6 操作禁用状态输出引脚

(1) $\overline{\text{SCKBn}}$ 引脚

当 CSIBn 的操作禁止(CBnCTL0.CBnPWR = 0)时, 引脚 $\overline{\text{SCKBn}}$ 的输出状态如下。

CBnCKS2	CBnCKS1	CBnCKS0	CBnCKP	$\overline{\text{SCKBn}}$ 引脚输出
1	1	1	x	高阻抗
以上除外			0	固定为高电平
以上除外			1	固定为低电平

- 备注**
1. 如果 CBnCTL1.CBnCKP 和 CBnCKS2 ~ CBnCKS0 被重写, 引脚 $\overline{\text{SCKBn}}$ 的输出电平将改变。
 2. n = 0, 1
 3. x: 忽略

(2) SOBn 引脚

当 CSIBn 的操作禁止(CBnPWR 位 = 0)时, 引脚 SOBn 输出状态如下。

CBnTXE	CBnDAP	CBnDIR	SOBn 引脚输出
0	x	x	固定为低电平
1	0	x	SOBn 锁存值 (低电平)
	1	0	CBnTX 寄存器的值 (MSB)
		1	CBnTX 寄存器的值 (LSB)

- 备注**
1. 当 CBnCTL0.CBnTXE, CBnCTL0.CBnDIR 位和 BnCTL1.CBnDAP 位其中的任何一个被重写时, SOBn 引脚时改变。
 2. n = 0, 1
 3. x: 忽略

13.7 波特率发生器

由波特率发生器（预分频器 3）产生的时钟应用于钟表定时器和 CSIB0。

(1) 预分频模式寄存器 0 (PRSM0)

PRSM0 寄存器为 CSIB 控制波特率信号的发生。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后：00H R/W 地址：FFFF8B0H

	7	6	5	4	3	2	1	0
PRSM0	0	0	0	BGCE0	0	0	BGCS01	BGCS00

BGCE0	波特率输出	
0	禁用	
1	启用	

BGCS01	BGCS00	计数时钟选择 (f _{BGCS})		
			5 MHz	4 MHz
0	0	f _x	200 ns	250 ns
0	1	f _x /2	400 ns	500 ns
1	0	f _x /4	800 ns	1 μ s
1	1	f _x /8	1.6 μ s	2 μ s

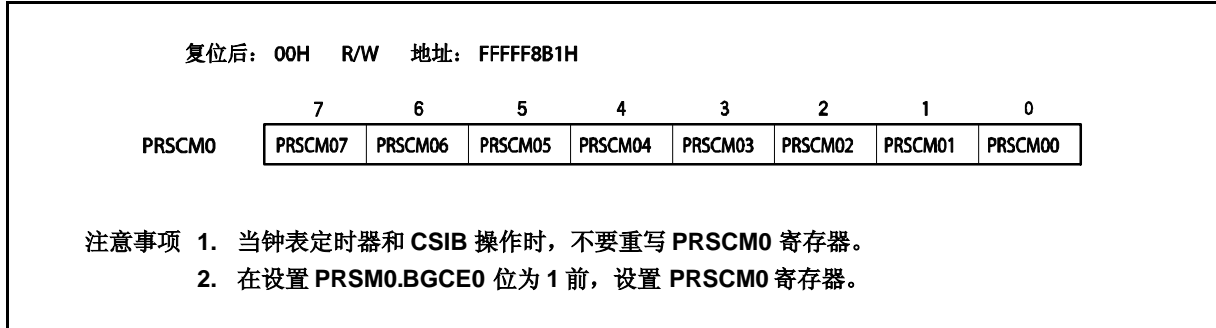
- 注意事项
1. 当钟表定时器和 CSIB0 操作时，不要写入 PRSM0 寄存器。
 2. 在设置 BGCE0 位为 1 前，设置 PRSM0 寄存器。

(2) 预分频器比较寄存器 0 (PRSCM0)

PRSCM0 寄存器是 8 位的比较寄存器。

该寄存器支持 8 位读写方式。

该寄存器复位后的值为 00H。

**13.7.1 波特率发生器**

对主时钟分频产生发送/接收时钟。由主时钟产生波特率通过下列公式计算。

$$f_{BRG} = \frac{f_{XX}}{2^{k+1} \times N}$$

<R> **注意事项** 设置通信时钟(f_{BRG})频率小于等于 8 MHz。

备注

- f_{BRG} : BRG 计数时钟
- f_{XX} : 主时钟振荡频率
- k: PRSM0 寄存器设置值 = 0 ~ 3
- N: PRSCM0 寄存器设置值 = 1 ~ 256

但是, 仅当 PRSCM0 寄存器被设为 00H 时 N = 256。

13.8 注意事项

(1) 当使用 DMA 传送接收和发送的数据，即使串行传送期间出现溢出错误，也不进行处理。可以在 DMA 传送后读取 CBnSTR.CBnOVE 寄存器，查看是否出现溢出错误。

(2) 考虑到寄存器禁止在操作期间(CBnCTL0.CBnPWR 为 1)重写，如果操作期间由于错误操作而重写寄存器，设置 CBnCTL0.CBnPWR 为 0，然后初始化 CSIBn。

禁止在操作期间重写的寄存器如下。

- CBnCTL0 寄存器: CBnTXE, CBnRXE, CBnDIR, CBnTMS 位
- CBnCTL1 寄存器: CBnCKP, CBnDAP, CBnCKS2 ~ CBnCKS0 位
- CBnCTL2 寄存器: CBnCL3 ~ CBnCL0 位

(3) 传输类型 2 和 4 (CBnCTL1.CBnDAP 位= 1)，产生接收完成中断(INTCBnR)后， \overline{SCKBn} 时钟执行到一半时 CBnSTR.CBnTSF 被清零。

在单传输模式下，通讯过程中(CBnTSF = 1)忽略写入的下一个发送数据，下一次传输不开始。同样，在只能接收的通讯中(CBnCTL0.CBnTXE = 0, CBnCTL0.CBnRXE = 1)，如果接收到的数据在传输过程中(CBnTSF = 1)被读取，则下一次传输不开始。

因此，在传输模式 2 或 4(CBnDAP = 1)下的单独发送模式，特别要注意下述问题。

- 要开始下一次发送，确保 CBnTSF = 0，然后向寄存器 CBnTX 写入要发送的数据。
- 只能接收传输模式下(CBnTXE = 0, CBnRXE = 1)，要连续执行下一次接收，确保 CBnTSF = 0，然后读取寄存器 CBnTX。

或者，使用连续发送模式而不用单独发送模式。推荐在使用 DMA 时使用连续发送模式。

备注 n = 0, 1

第十四章 DMA 功能 (DMA 控制器)

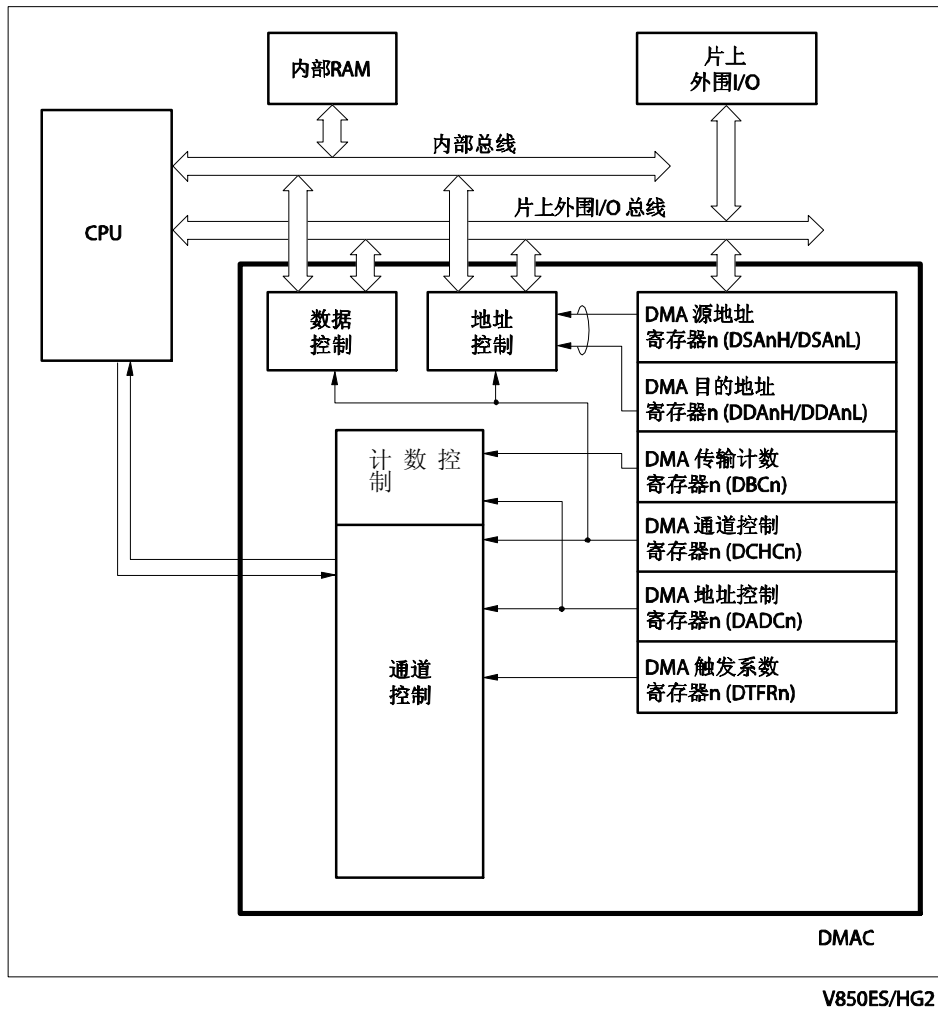
V850ES/HG2 包含了存储器直接访问(DMA) 控制器(DMAC) , 它用来执行和控制 DMA 传送。

DMAC 控制基于 DMA 请求的存储器和端口, 或端口和端口之间的数据传送, 这种请求通常是由片上外围端口(串行接口, 定时器/计数器和 A/D 转换器), 外部输入引脚的中断, 或软件触发(内部 RAM 存储器) 发出的。

14.1 特点

- 4 路独立的 DMA 通道
- 传送单元: 8/16 位
- 最大传送计数值: 65, 536 (2^{16})
- 传送类型: 双周期传送
- 传送模式: 单一的传送模式
- 传送请求
 - 通过片上外围端口(串行接口, 定时器/计数器, A/D 转换器) 中断或是外部输入引脚的中断而产生的请求
 - 通过软件触发产生的请求
- 传送目标
 - 内部 RAM ↔ 外围 I/O
 - 外围 I/O ↔ 外围 I/O

14.2 配置



备注 n = 0 ~ 3

14.3 寄存器

(1) DMA 源地址寄存器 0 ~ 3 (DSA0 ~ DSA3)

DSA0 ~ DSA3 寄存器为 DMA 通道 n (n = 0 ~ 3) 设定 DMA 源地址 (每隔寄存器 26 位)。这些寄存器被分为两个 16 位寄存器, DSA_nH 和 DSA_nL。这些寄存器支持 16 位读写方式。

复位后: 未定R/W 地址: DSA0H FFFF082H, DSA1H FFFF08AH,
 DSA2H FFFF092H, DSA3H FFFF09AH,
 DSA0L FFFF080H, DSA1L FFFF088H,
 DSA2L FFFF090H, DSA3L FFFF098H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSA _n H (n = 0 ~ 3)	IR	0	0	0	0	0	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSA _n L (n = 0 ~ 3)	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0

IR	DMA 传输源规格
0	片上外围I/O
1	内部RAM

SA25 ~ SA16	设置 DMA 传输源地址(A25 ~ A16)(默认值未定)。在DMA 传输中, 保留下一条 DMA 传输源地址。当DMA 传输完成后, 首先保留DMA 地址设置。
-------------	--

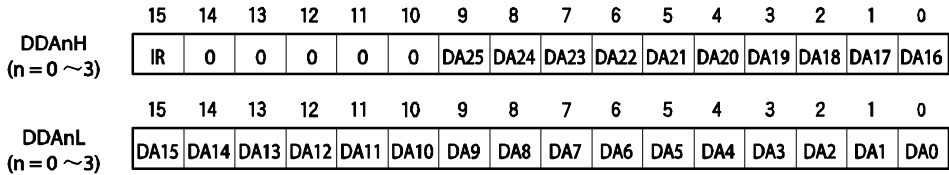
SA15 ~ SA0	设置 DMA 传输源地址(A15 ~ A0)(默认值未定)。在DMA 传输中, 保留下一条 DMA 传输源地址。当DMA 传输完成后, 首先保留DMA 地址设置。
------------	---

- 注意事项
1. 确保将 DSA_nH 寄存器的第 14~10 位清零。
 2. 当 DMA 传送被禁止时 (DCHCn.Enn 位= 0), 在下列时序下设定 DSA_nH 和 DSA_nL 寄存器。
 - 从复位后到开始第一个 DMA 传送之间的时期
 - 从通过 DCHCn.INITn 位将通道初始化后到开始 DMA 传送之间的时期
 - 从完成 DMA 传送 (DCHCn.TCn 位= 1) 到开始下一个 DMA 传送之间的时期
 3. 当 DSA_n 寄存器中的数值被读取, 两个 16 位寄存器 DSA_nH 和 DSA_nL 的值被读取。如果读取操作与数值更新发生冲突, 可能会读取正在被更新的数值 (参见 14.13 注意事项)。
 4. 复位后, 在开始 DMA 传送之前设定 DSA_nH, DSA_nL, DDA_nH, DDA_nL 和 DBCn 寄存器。否则当 DMA 传送开始传送时, 操作无法得到保证。

(2) DMA 目的地址寄存器 0 ~ 3 (DDA0 ~ DDA3)

DDA0 ~ DDA3 寄存器为 DMA 通道 n (n = 0 ~ 3) 设定 DMA 目的地址 (每个寄存器 26 位)。这些寄存器被分为两个 16 位寄存器, DDA_nH 和 DDA_nL。这些寄存器支持 16 位读写方式。

复位后: 未定R/W 地址: DDA0H FFFF086H, DDA1H FFFF08EH,
 DDA2H FFFF096H, DDA3H FFFF09EH,
 DDA0L FFFF084H, DDA1L FFFF08CH,
 DDA2L FFFF094H, DDA3L FFFF09CH



IR	DMA 传输目的的规格
0	片上外围I/O
1	内部RAM

DA25 ~ DA16	设置 DMA 传输目的的地址 (A25 ~ A16) (默认值未定)。在DMA 传输中, 保留下一条 DMA 传输目的地址。在DMA 传输i完成后, 首先保留 DMA 传输源地址设置。
-------------	--

DA15 ~ DA0	设置 DMA 传输目的的地址 (A15 ~ A0) (默认值未定)。在DMA 传输中, 保留下一条 DMA 传输目的地址。在DMA 传输i完成后, 首先保留 DMA 传输源地址设置。
------------	---

- 注意事项**
1. 确保将 DDA_nH 寄存器的第 14~10 位清零。
 2. 当 DMA 传送被禁止时 (DCHC_n.Enn 位= 0), 在下列时序下设定 DDA_nH 和 DDA_nL 寄存器。
 - 从复位后到开始第一个 DMA 传送之间的时期
 - 从通过 DCHC_n.INIT_n 位将通道初始化后到开始 DMA 传送之间的时期
 - 从完成 DMA 传送 (DCHC_n.TC_n 位= 1) 到开始下一个 DMA 传送之间的时期
 3. 当 DDA_n 寄存器中的数值被读取, 两个 16 位寄存器 DDA_nH 和 DDA_nL 的值被读取。如果读取操作与数值更新发生冲突, 可能会读取正在被更新的数值 (参见 14.13 注意事项)。
 4. 复位后, 在开始 DMA 传送之前设定 DSA_nH, DSA_nL, DDA_nH, DDA_nL 和 DBC_n 寄存器。否则当 DMA 传送开始传送时, 操作无法得到保证。

(3) DMA 传送计数器寄存器 0 ~ 3 (DBC0 ~ DBC3)

DBC0 ~ DBC3 寄存器是为 DMA 通道 n (n = 0 ~ 3) 设置字节传送计数器的 16 位寄存器。在 DMA 传送过程中, 这些寄存器保持剩余的发送字节数。

这些寄存器在每次传送之后会自动减 1, 不考虑传送的数据单元(8/16 位), 并且在发生借位是中止发送。

这些寄存器支持 16 位读写方式。

复位后: 未定 R/W 地址: DBC0 FFFF0C0H, DBC1 FFFF0C2H,
DBC2 FFFF0C4H, DBC3 FFFF0C6H

DBCn (n=0~3)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0

BC15 ~ BC0	字节传输计数设置或在DMA传输中保留 字节传输计数
0000H	字节传输计数1 或保留字节传输计数
0001H	字节传输计数2 或保留字节传输计数
:	:
FFFFH	字节传输计数65,536 (2 ¹⁶) 或保留字节传输计数
当DMA 传输完成后, 首先保留传输数据设置的数目。	

注意事项 1. 当 DMA 传送被禁止时 (DHCn.Enn 位= 0), 在下列时序下设定 DBCn 寄存器。

- 从复位后到开始第一个 DMA 传送之间的时期
- 从通过 DHCn.INITn 位将通道初始化后到开始 DMA 传送之间的时期
- 从完成 DMA 传送 (DHCn.TCn 位= 1) 到开始下一个 DMA 传送之间的时期

2. 复位后, 在开始 DMA 传送之前设定 DSA nH, DSA nL, DDA nH, DDA nL 和 DBCn 寄存器。否则当 DMA 传送开始传送时, 操作无法得到保证。

(4) DMA 寻址控制寄存器 0 ~ 3 (DADC0 ~ DADC3)

DADC0 ~ DADC3 寄存器是为 DMA 通道 n (n = 0 ~ 3) 控制 DMA 传送模式的 16 位寄存器。这些寄存器支持 16 位读写方式。这些寄存器复位后的值为 0000H。

复位后: 0000H R/W 地址: DADC0 FFFF0D0H, DADC1 FFFF0D2H,
DADC2 FFFF0D4H, DADC3 FFFF0D6H

DADCn (n = 0 ~ 3)	15	14	13	12	11	10	9	8
	0	DS0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	SAD1	SAD0	DAD1	DAD0	0	0	0	0

DS0	传输数据规格的设置	
0	8 位	
1	16 位	

SAD1	SAD0	传输源地址计数方向的设置
0	0	增加
0	1	减少
1	0	固定值
1	1	禁止设置

DAD1	DAD0	目的地址计数方向的设置
0	0	增加
0	1	减少
1	0	固定值
1	1	禁止设置

- 注意事项
1. 确保将 DADCn 寄存器的第 15, 13 ~ 8 位清零。
 2. 当 DMA 传送被禁止时 (DCHCn.Enn 位= 0), 在下列时序下设定 DADCn 寄存器。
 - 从复位后到开始第一个 DMA 传送之间的时期
 - 传送从通过 DCHCn.INITn 位将通道初始化后到开始 DMA 传送之间的时期
 - 从完成 DMA 传送 (DCHCn.TCn 位= 1) 到开始下一个 DMA 传送之间的时期
 3. DS0 位指定了传送数据的大小, 但不能控制总线宽度。因此如果设定为 8 位数据 (DS0 位=0), 较低位的数据总线并不经常使用。
 4. 如果传送数据设定为 16 位 (DS0 位= 1), 传送不能从奇数地址开始。传送只能从低地址的首位为 0 的地址开始。
 5. 如果由片上外围端口寄存器 (无论作为发送的源还是目的) 执行 DMA 传送, 一定要按照寄存器的大小来设定传送数据的大小。例如, 在一个 8 位寄存器上执行 DMA 传送, 一定要指定 8 位传送。

(5) DMA 通道控制寄存器 0 ~ 3 (DCHC0 ~ DCHC3)

DCHC0 ~ DCHC3 寄存器是为 DMA 通道 n 控制 DMA 传送操作的 8 位寄存器。

这些寄存器支持 8 位或 1 位的读写方式。(但是, 第 7 位是只读位, 第 1 和第 2 位是只写位。如果读取第 1 和第 2 位, 则读取值永远为 0。)

这些寄存器复位后的值为 00H。

复位后: 00H R/W 地址: DCHC0 FFFF0E0H, DCHC1 FFFF0E2H, DCHC2 FFFF0E4H, DCHC3 FFFF0E6H								
	7	6	5	4	3	2	1	0
DCHCn (n = 0 ~ 3)	TCn ^{注1}	0	0	0	0	INITn ^{注2}	STGn ^{注2}	Enn
	TCn ^{注1}	用于指示通过 DMA 通道 n 进行的 DMA 传输是否完成的状态标志						
	0	没有完成 DMA 传输。						
	1	完成 DMA 传输						
	在最终 DMA 的传输时, 它被设置为 1。当它被读取时, 将被清零。							
	INITn ^{注2}	如果 INITn 位设置为 1, 禁用 DMA 传输 (Enn 位 = 0), 可初始化 DMA 传输状态。 在 DMA 传输完成前 (TCn 位设置为 1 前), 当重新设置 DMA 传输状态 (重新设置 DDA nH, DDA nL, DSA nH, DSA nL, DBCn, 及 DADCn 寄存器) 一定要初始化 DMA 通道。 当初始化 DMA 控制器时, 然而, 一定要注意 14.13 注意事项中所描述的步骤。						
	STGn ^{注2}	该位为 DMA 传输的软件启动触发器。 如果在启用 DMA 传输状态 (TCn 位 = 0, Enn 位 = 1) 时该位设置为 1, 将开始 DMA 传输。						
	Enn	对通过 DMA 通道 n 进行的 DMA 传输进行启用或禁用的设置						
	0	禁用 DMA 传输						
	1	启用 DMA 传输						
	当 Enn 位 i 设置为 1 时, 启用 DMA 传输。 当完成 DMA 传输后 (当发生一终端计数), 该位将自动清零。 可用软件清零 Enn 位来中止 DMA 传输。如需启用, 再次设置 Enn 位为 1。 当中止或启用 DMA 传输时, 然而, 一定要注意 14.13 注意事项中所描述的步骤。							

注 1. TCn 位为只读位。
 2. INITn 和 STGn 位为只写位。

注意事项 1. 确保将 DCHCn 寄存器的第 6~3 位清零。
 2. 当 DMA 传送完成时 (中止计数产生), Enn 位清零然后 TCn 位设置为 1。如果在其数值被更新时读取 DCHCn 寄存器, 指示“传送未完成和传送被禁止”(TCn 位 = 0 且 Enn 位 = 0) 的数值可以被读取。

(6) DMA 触发因素寄存器 0 ~ 3 (DTFR0 ~ DTFR3)

DTFR0 ~ DTFR3 寄存器是 8 位寄存器，它通过片上外围 I/O 的中断请求信号控制 DMA 传送的开始触发。由这些寄存器设置的中断请求信号被作为 DMA 开始传送因素对待。这些寄存器支持 8 位读写方式。但是，DFn 位支持 1 位读写方式。这些寄存器复位后的值为 00H。

复位后: 00H R/W 地址: DTFR0 FFFF810H, DTFR1 FFFF812H,
DTFR2 FFFF814H, DTFR3 FFFF816H

	7	6	5	4	3	2	1	0
DTFRn	DFn	0	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0
(n = 0 ~ 3)								
	DFn ^注	DMA 传输请求状态标志						
	0	无DMA 传输请求						
	1	DMA 传输请求						

注 不要通过软件将 DFn 位设置为“1”。如果在 DMA 传送禁止期间，一个指定作为引起 DMA 传送的中断发送，对该位写入 0 将清除一个 DMA 传送请求。

- 注意事项**
- 当 DMA 传送被禁止时 (DCHCn.Enn 位= 0)，在下列时序下设定 IFCn5 ~ IFCn0 位。
 - 从复位后到开始第一个 DMA 传送之间的时期
 - 从通过 DCHCn.INITn 位将通道初始化后到开始 DMA 传送之间的时期
 - 从完成 DMA 传送 (DCHCn.TCn 位= 1) 到开始下一个 DMA 传送之间的时期
 - 在待机模式下产生的中断 (IDEL1, IDLE2, STOP, 或副 IDLE 模式) 不会开始 DMA 传送周期 (也不会将 DFn 位设置为 1)。
 - 如果通过 IFCn5~IFCn0 位选择了 DMA 启动系数，当被选择的片上外围端口的中断产生时，DFn 位被设置为 1，而忽略 DMA 传送是否被允许。在这种状态下如果允许 DMA，那么 DMA 传送立即开始。

备注 关于 IFCn5 ~ IFCn0 位，敬请参阅表 14-1 DMA 启动系数。

<R>

表 14-1. DMA 启动系数 (1/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	中断源
0	0	0	0	0	0	禁止通过中断的 DMA 请求
0	0	0	0	0	1	INTLVI
0	0	0	0	1	0	INTP0
0	0	0	0	1	1	INTP1
0	0	0	1	0	0	INTP2
0	0	0	1	0	1	INTP3
0	0	0	1	1	0	INTP4
0	0	0	1	1	1	INTP5
0	0	1	0	0	0	INTP6
0	0	1	0	0	1	INTP7
0	0	1	0	1	0	INTTQ0OV
0	0	1	0	1	1	INTTQ0CC0
0	0	1	1	0	0	INTTQ0CC1
0	0	1	1	0	1	INTTQ0CC2
0	0	1	1	1	0	INTTQ0CC3
0	0	1	1	1	1	INTTP0OV
0	1	0	0	0	0	INTTP0CC0
0	1	0	0	0	1	INTTP0CC1
0	1	0	0	1	0	INTTP1OV
0	1	0	0	1	1	INTTP1CC0
0	1	0	1	0	0	INTTP1CC1
0	1	0	1	0	1	INTTP2OV
0	1	0	1	1	0	INTTP2CC0
0	1	0	1	1	1	INTTP2CC1
0	1	1	0	0	0	INTTP3OV
0	1	1	0	0	1	INTTP3CC0
0	1	1	0	1	0	INTTP3CC1
0	1	1	0	1	1	INTTM0EQ0
0	1	1	1	0	0	INTCB0R
0	1	1	1	0	1	INTCB0T
0	1	1	1	1	0	INTCB1R
0	1	1	1	1	1	INTCB1T
1	0	0	0	0	0	INTUA0R
1	0	0	0	0	1	INTUA0T
1	0	0	0	1	0	INTUA1R
1	0	0	0	1	1	INTUA1T
1	0	0	1	0	0	INTAD
1	0	1	0	0	1	INTKR
1	0	1	0	1	0	INTTQ1OV

备注 n = 0 ~ 3

表 14-1. DMA 启动系数 (2/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	中断源
1	0	1	0	1	1	INTTQ1CC0
1	0	1	1	0	0	INTTQ1CC1
1	0	1	1	0	1	INTTQ1CC2
1	0	1	1	1	0	INTTQ1CC3
1	0	1	1	1	1	INTUA2R
1	1	0	0	0	0	INTUA2T

备注 n = 0 ~ 3

14.4 传送目标

表 14-2 显示了传送目标之间的关系 (√: 允许传送, ×: 禁止传送)。

表 14-2. 传送目标值之间的关系

		传送目的		
		内部 ROM	片上外围端口	内部 RAM
源	片上外围端口	×	√	√
	内部 RAM	×	√	×
	内部 ROM	×	×	×

注意事项 表 14-2 中标记为“×”的传送目的和源之间执行操作，结果将不会得到保证。

14.5 传送模式

传送模式支持单一传送。

在单一传送模式中，总线在每一次字节/半字传送后被释放。如果存在后续的 DMA 传送请求，传送就会被再次执行。这种操作直到发生了中断计数才会停止。

当 DMAC 释放总线后，如果发出一个具有更高优先级的 DMA 传送请求，该请求始终优先。

如果在传送周期内同一个通道发出一个新的传送请求同时另外一个通道产生一个具有较低优先级的传送请求，则在 CPU 释放总线之后，执行具有较低优先级通道的 DMA 传送（在传送周期中同一通道新的传送请求被忽略）。

14.6 传送类型

传送类型支持双周期传送。

在双周期传送中，数据传送在两个周期内执行，一个读取周期和一个写入周期。

在读取周期中，传送源地址被输出，从源到 DMAC 执行读取操作。在写入周期中，传送的目的地址被输出，从 DMAC 到目的执行写入操作。

时钟的空闲周期永远插入在读取周期和写入周期之间。如果数据总线宽度与用于进行双周期 DMA 传送的传送源和目的不一致，操作将按照如下步骤执行。

<16 位数据传送>

<1> 32 位总线 → 16 位总线的传送

在产生一个写入周期（16 位）之后，紧接着产生一个读取周期（高 16 位处于高阻抗状态）。

<2> 16/32 位总线到 8 位总线的传送

16 位读取周期产生一次，随后两次产生 8 位写入周期。

<3> 8 位总线到 16/32 位总线的传送

8 位读取周期产生两次，随后 16 位写入周期产生一次。

<4> 16 位总线和 32 位总线之间的传送

16 位读取周期产生一次，随后 16 位写入周期产生一次。

执行到一个片上外围端口寄存器（源/目的）的 DMA 传送，一定要指定与寄存器大小相同的传送单元。例如，对一个 8 位寄存器执行 DMA 传送，一定要指定字节（8 位）传送。

备注 每一种目的的总线宽度（传送源/目的）如下所示。

- 片上外围端口： 16 位总线宽度
- 内部 RAM： 32 位总线宽度

14.7 DMA 通道优先级

DMA 通道优先级固定如下。

DMA 通道 0 > DMA 通道 1 > DMA 通道 2 > DMA 通道 3

在每一个传送周期检测优先级。

14.8 DMA 传送的相关时间

响应 DMA 请求所需要的时间和 DMA 传送所需的最小时钟周期如下所示。

单一传送: DMA 响应时间(<1>) + 传送源存储器访问(<2>) + 1^{注1} + 传送目的存储器访问 (<2>)

DMA 周期		最小执行周期
<1> DMA 请求响应时间		4 时钟 (最小) + 噪声消除时间 ^{注2}
<2> 存储器访问	内部 RAM 访问	2 时钟 ^{注3}
	外围端口寄存器访问	3 时钟 + 由 VSWC 寄存器指定的等待周期数 ^{注4}

- 注**
1. 在 DMA 传送的读取周期和写入周期之间始终要插入一个时钟。
 2. 如果指定一个外部中断 (INTPn) 作为触发开始 DMA 传送, 噪音消除时间将增加(n = 0 ~ 10)。
 3. DMA 周期需要两个时钟。
 4. 访问特定的外围端口寄存器需要更多的等待周期 (关于更多细节, 敬请参阅 3.4.8 (2))。

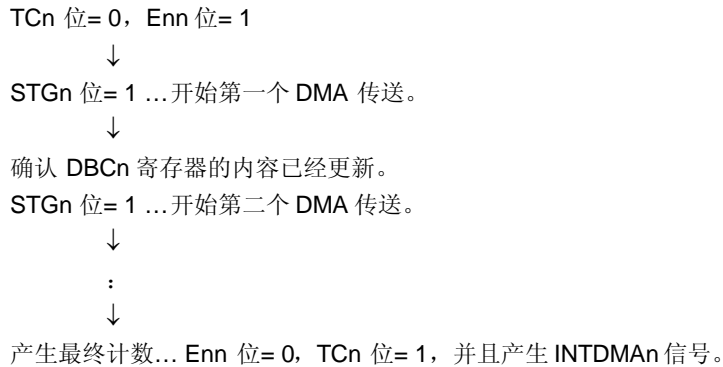
14.9 DMA 传送启动系数

有两种类型的 DMA 传送启动系数，如下所示。

(1) 通过软件发出请求

如果当 $DCHCn.TCn$ 位 = 1 且 Enn 位 = 1 (允许 DMA 传送) 时把 $STGn$ 位设置为 1，则 DMA 传送开始。

为了在此之后立即请求下一个 DMA 传送周期，通过 $DBCn$ 寄存器确认先前的 DMA 传送周期已经完成，然后再次将 $STGn$ 位设置为 1 ($n = 0 \sim 3$)。



(2) 通过片上外围 I/O 产生请求

如果当 $DCHCn.TCn$ 位 = 0 且 Enn 位 = 1 (允许 DMA 传送) 时，由 $DTFRn$ 寄存器设定的片上外围 I/O 产生中断请求，DMA 传送开始。

- 注意事项**
1. 两种启动系数 (软件触发和硬件触发) 不能应用在同一 DMA 通道。如果两种启动系数在一个 DMA 通道中同时产生，则其中仅有一个有效。有效的启动系数不能被识别。
 2. 在先前的 DMA 传送请求产生后或是先前的 DMA 传送周期后产生的新的传送请求将被忽略 (清除)。
 3. 相同的 DMA 通道传送请求间隔的变化取决于在 DMA 传送周期中总线等待的设定，其它通道的开始状态或是外部总线保持请求。特别是注意事项 2 中的描述，在 DMA 传送周期之前或是 DMA 传送周期中同一通道产生的新的发送请求将被忽略。因此，相同 DMA 通道的传送请求间隔必须通过系统充分分离。当应用软件触发时，先前产生的 DMA 传送周期完成可以通过更新 $DBCn$ 寄存器检测。

14.10 DMA 中止因素

如果发生总线保持 DMA 传送将被中止。

如果传送在内部存储器/片上外围端口和内部存储器/片上外围端口之间执行时, 会出现相同的应用。

当清除总线保持后, DMA 传送恢复。

14.11 DMA 传送结束

当 DCHCn.Enm 位清零且 TCn 位设置为 1 后, 如果 DMA 传送已经完成了在 DBCn 寄存器中设定的次数, 将会为中断控制器(INTC)(n = 0 ~ 3)产生一个 DMA 传送结束中断请求信号(INTDMAn)。

V850ES/HG2 不会向外部设备输出最终计数信号。因此, 要通过 DMA 传送结束中断或是 TCn 位检测确认 DMA 传送完成。

14.12 操作时序

图 14-1 ~ 14-4 显示了 DMA 操作时序。

图 14-1. DMA 的优先级 (1)

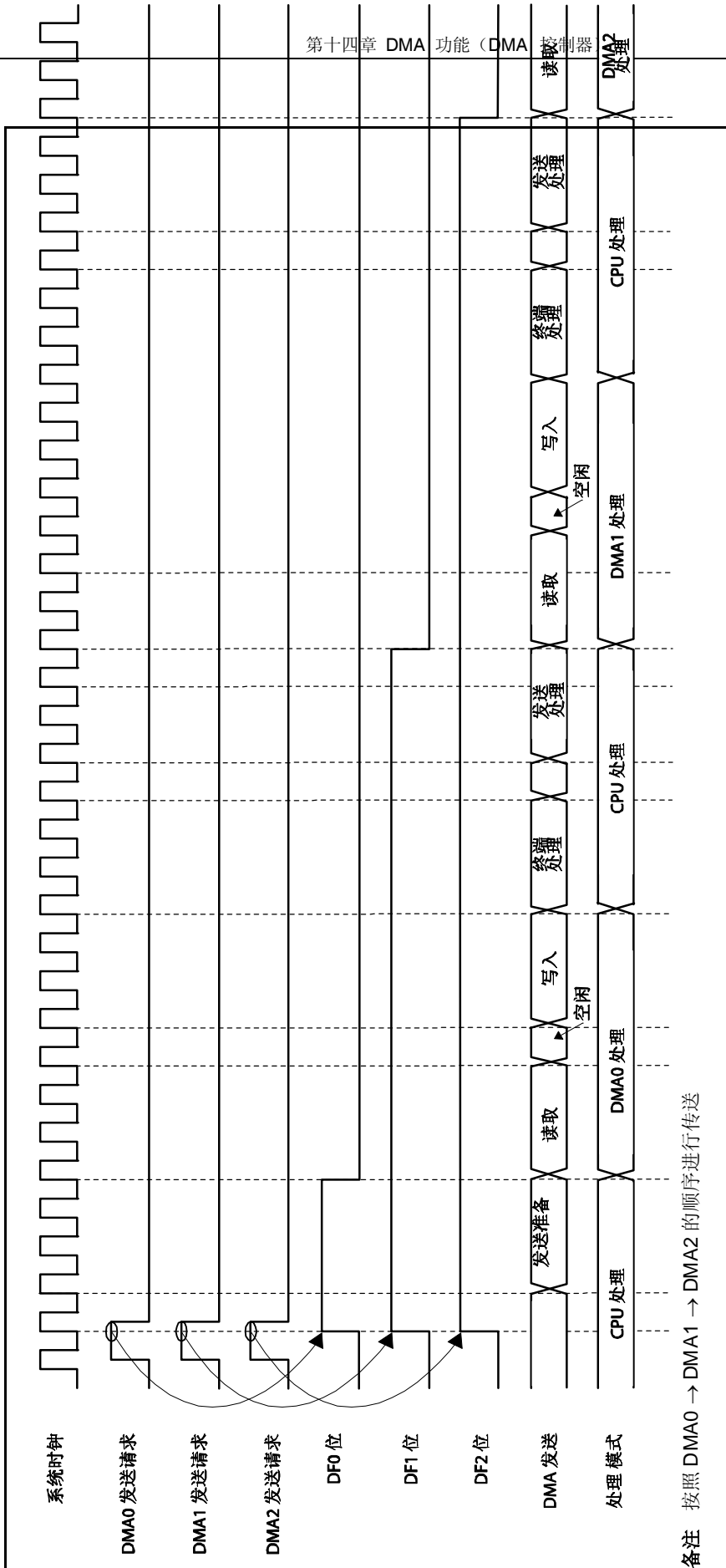


图 14-2. DMA 的优先级 (2)

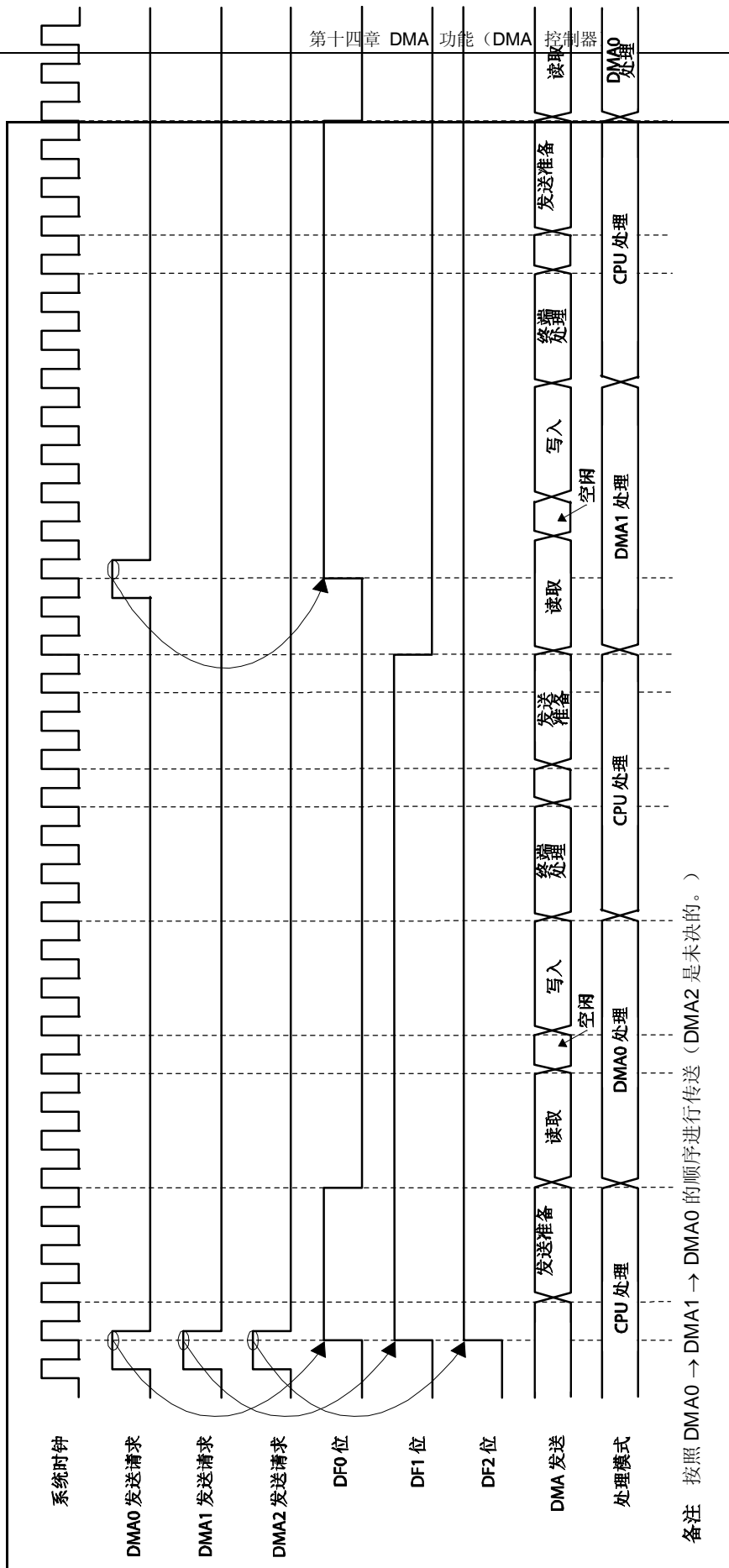


图 14-3. DMA 传送请求被忽略的时期 (1)

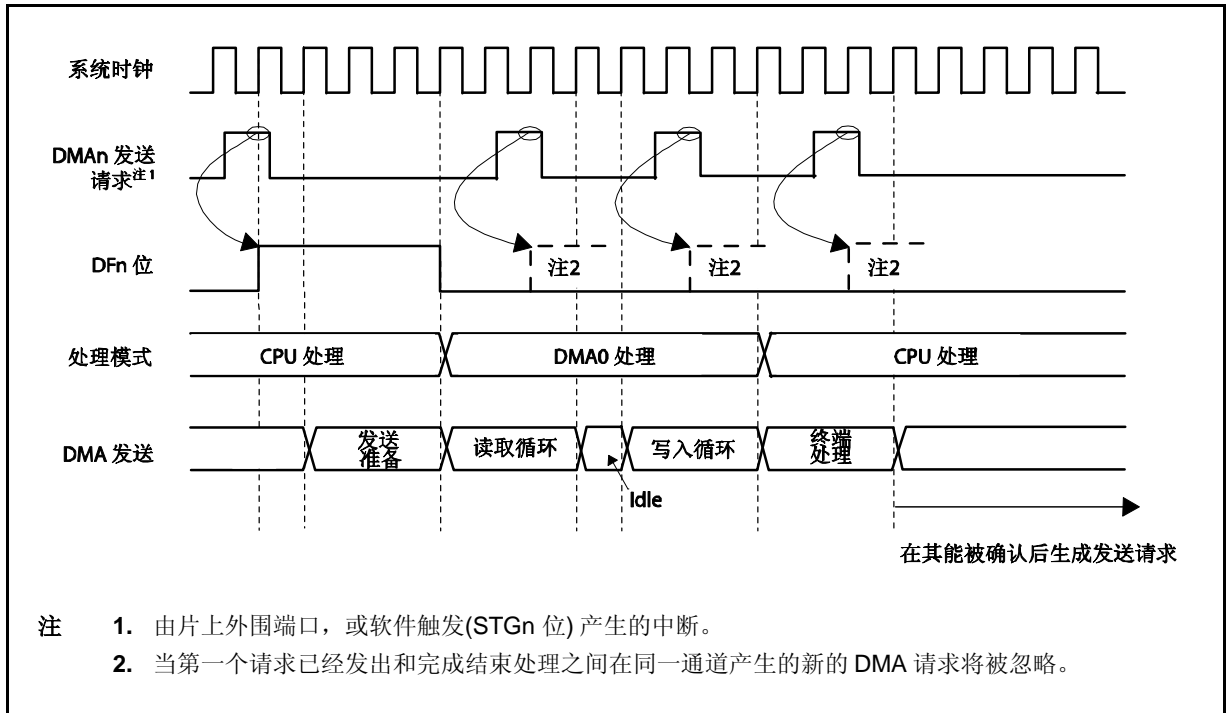
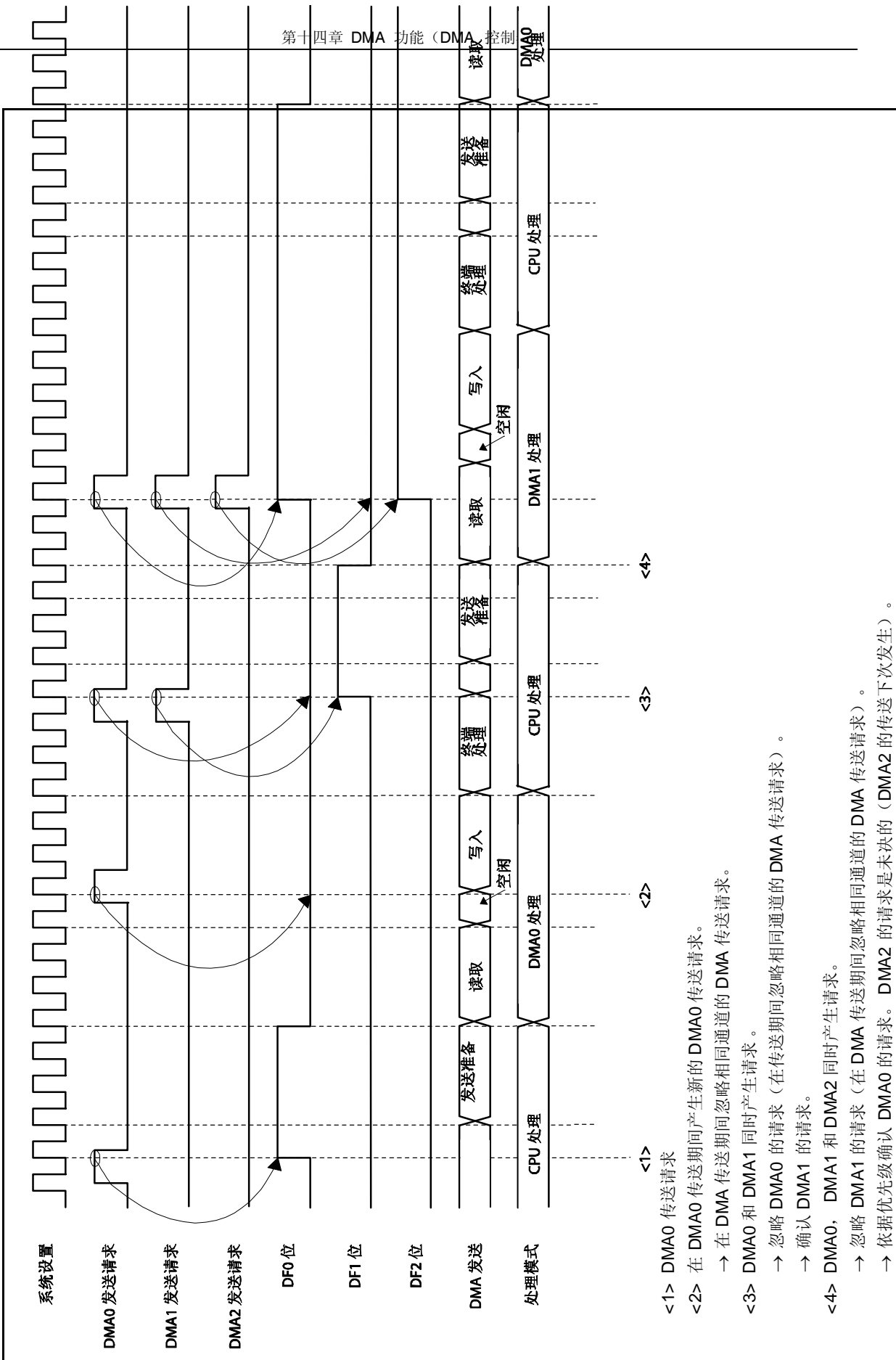


图 14-4. DMA 传送请求被忽略的时期 (2)



14.13 注意事项

(1) VSWC 寄存器的注意事项

当使用 VSWC 寄存器的默认值 (77H)，或者为 VSWC 寄存器设定的值不恰当时，不会执行正确的操作（关于 VSWC 寄存器的更多细节，敬请参阅 3.4.8 (1) (a) 系统等待控制寄存器 (VSWC)）。

(2) 在内部 RAM 中执行 DMA 传送的注意事项

当在内部 RAM 中执行下列指令时，不能在内部 RAM 中（源/目的）执行 DMA 传送，因为 CPU 后来可能不会正确地运行。

- 在内部 RAM 中对于非线性地址的数据访问指令

相反地，当在内部 RAM 中（传送源/目的）执行 DMA 传送时，不要执行上述两种指令。

(3) 读取 DCHCn.TCn 位的注意事项(n = 0 ~ 3)

当对 TCn 位进行读取操作时它将被清零，但是即使是在一个特殊的时序，它也并不是自动被清零。为了正确地将 TCn 位清零，要加上下列处理过程。

(a) 当通过检测 TCn 位的方式等待 DMA 传送完成

确认 TCn 位已经被设置为 1（在 TCn 位 = 1 被读取后），然后另外在读取 TCn 位三次。

(b) 当在中断服务程序种读取 TCn 位时

执行读取 TCn 位 3 次。

(4) DMA 传送初始化过程 (将 DCHCn.INITn 位设置为 1)

当要执行 DMA 传送的通道被初始化时, 即使将 INITn 位设置为 1, 该通道也不会被初始化。为了正确地初始化通道, 需要执行下列两个过程之一。

(a) 临时中止所有通道的 DMA 传送

初始化正在执行 DMA 传送的通道需要按以下步骤<1>~<7> 执行。

注, 但是当执行第<5>步时 TCn 位被清零。确定其它处理程序不要预期 TCn 位为 1。

<1> 禁止中断(DI)。

<2> 读取被强行中止的 DMA 通道之外的其它 DMA 通道的 DCHCn.Enn 位, 然后将这个值传送到通用寄存器。

<3> 将正在被使用的 DMA 通道的 Enn 位 (包括被强行中止的通道) 清零。为了将最后一个 DMA 通道的 Enn 位清零, 需要执行两次清除指令。如果 DMA 传送的目的 (传送源/目的) 是内部 RAM, 需要将该指令执行三次。

例: 如果正在使用通道 0, 1 和 2 (如果传送的目的不是内部 RAM), 按照以下顺序执行指令。

- 将 DCHC0.E00 位清零。
- 将 DCHC1.E11 位清零。
- 将 DCHC2.E22 位清零。
- 再次将 DCHC2.E22 位清零。

<4> 将被强行中止的 DMA 通道的 INITn 位设置为 1。

<5> 读取除被强行中止的通道外的其它通道的 TCn 位。如果在第<2>步中读取的 TCn 位和 Enn 位的值都为 1 (逻辑与的结果为 1), 将保存的 Enn 位清零。

<6> 在第<5>步操作之后, 将 Enn 位的值写入 DCHCn 寄存器。

<7> 允许中断(EI)。

注意事项 一定要执行上述的第<5>步来防止非法的对完成第<2>和<3>步 DMA 传送正常操作的通道的 Enn 位的设置。

(b) 重复执行设定 INITn 位直到传送被正确的强行中止

- <1> 抑制要被强行中止的通道 DMA 请求源发出的请求 (停止片上外围端口的操作)。
- <2> 通过使用 DTFRn.DFn 位来检测要被强行中止的通道 DMA 传送请求未处于保持状态。如果一个 DMA 传送请求处于保持状态, 需要等待直到执行完成该保持请求。
- <3> 如果确定要被强行中止的通道 DMA 请求未处于保持状态, 将 Enn 位清零。
- <4> 再次将要被强行中止的通道 Enn 位清零。
如果传送的目的是要被强行中止的通道 (传送源/目的) 是内部 RAM, 再次执行该操作。
- <5> 拷贝要被强行中止的通道传送的初始值到通用寄存器。
- <6> 将要被强行中止的通道 INITn 位设置为 1。
- <7> 读取要被强行中止的通道 DBCn 寄存器的值, 并与第<5>步中的拷贝值相比较。如果两者不匹配, 重复第<6>步和第<7>步的操作。

- 备注**
- 1. 当在第<7>步中读取了 DBCn 寄存器的值后, 如果强行中止正确的完成, 那么传送的初始值被读取。否则, 传送的剩余值被读取。
 - 2. 注意如果一个除了被强行中止的通道之外的 DMA 通道应用经常的使用 DMA 传送, 那么模式(b)可能会占用很长的时间。

(5) 临时停止 DMA 传送的程序 (将 Enn 位)

停止和恢复 DMA 传送需要按以下步骤执行。

- <1> 抑制 DMA 请求源的传送请求 (停止片上外围端口)。
- <2> 使用 DFn 位来检测 DMA 传送请求未处于保持状态 (检测 DFn 位是否为 0)。
如果保持请求, 等待直到被保持的请求执行完毕。
- <3> 如果确认没有 DMA 传送请求被保持, 将 Enn 位清零 (该操作停止 DMA 传送)。
- <4> 将 Enn 位设置为 1 恢复 DMA 传送。
- <5> 恢复被停止的 DMA 请求源的操作 (开始片上外围端口的操作)。

(6) 存储器边界

如果在 DMA 传送期间传送的源或者目的地址超出了 DMA 目标的区域, 那么操作将不会得到保证 (内部 RAM 或片上外围端口)。

(7) 传送非线性数据

不支持在 16 为总线宽度上传送非线性数据。

如果一个奇数地址被指定为传送的源或者目的, 地址的最少的有意义的位被强制假定为 0。

(8) 对于 CPU 的总线仲裁

由于 DMA 控制器对于总线的控制权优先于 CPU，因此发生在 DMA 传送期间的 CPU 访问被保持直到 DMA 传送周期完成且将总线释放给 CPU。

但是，CPU 可以访问那些未执行 DMA 传送的片上外围端口和内部 RAM。

(9) 在 DMA 操作期间寄存器/位必须进行重新写入

在 DMA 操作未被执行时在下面的时序中设定下列寄存器。

[寄存器]

- DSAnH, DSAnL, DDA nH, DDA nL, DBCn 和 DADCn 寄存器
- DTFRn.IFCn5 ~ DTFRn.IFCn0 位

[时序设定]

- 从复位后到开始第一个 DMA 传送之间的时期
- 从通道初始化到开始 DMA 传送的时间
- 在完成第一个 DMA 传送 (TCn 位 = 1) 到开始下一个 DMA 传送之间的时期

(10) 一定要将下列寄存器的位清零。

- DSAnH 寄存器的第 14~10 位
- DDA nH 寄存器的第 14~10 位
- DADCn 寄存器的第 15, 13~8 和 3~0 位
- DCHCn 寄存器的第 6~3 位

(11) DMA 启动系数

<R>

在同一个启动系数上不要启动 2 个或更多的 DMA 通道。如果使用同一个系数启动两个或更多的通道，可能会启动已经设置过的 DMA 通道，或具有较低优先级的 DMA 通道比具有更高优先级的 DMA 通道先得到确认。操作的正确性将无法得到保证。

(12) 读取 DSAn 和 DDA n 寄存器的值

在 DMA 传送期间，可以在 DSAn 和 DDA n 寄存器中读取更新的中间值 (n = 0 ~ 3)。

例如，如果当 DMA 传送源地址 (DSAn 寄存器) 为 0000FFFFH 且计数方式为递增 (DADCn.SAD1 和 DADCn.SAD0 位 = 00)，此时读取 DSAnH 寄存器和 DSAnL 寄存器，DSAn 寄存器的值的区别如下所示，这将取决于在 DSAnH 寄存器被读取之后 DMA 传送是否被立即执行。

(a) 如果在 DSAn 寄存器被读取时 DMA 传送没有发生

- <1> DSAnH 寄存器的读取值: DSAnH = 0000H
- <2> DSAnL 寄存器的读取值: DSAnL = FFFFH

(b) 如果在 DSAn 寄存器被读取时发生了 DMA 传送

- <1> DSAnH 寄存器的读取值: DSAnH = 0000H
- <2> 出现 DMA 传送
- <3> 增加 DSAn 寄存器: DSAn = 00100000H
- <4> DSAnL 寄存器的读取值: DSAnL = 0000H

第十五章 中断/异常处理功能

V850ES/HG2 为中断服务提供一个专门的中断控制器 (INTC)，它可以处理总共 55 个中断请求。

中断是指不受程序执行影响的事件发生，异常是指依赖于程序执行的事件发生。

V850ES/HG2 可以处理由片上外围硬件和外部事件源引起的中断请求信号。此外，异常处理可以通过 TRAP 指令（软件异常）或者产生一个异常事件（例如：存在非法操作码）（异常陷阱）开始。

15.1 特点

○ 中断

- 非屏蔽中断： 2 个中断源
- 可屏蔽中断： 外部： 11，内部： 42 个中断源
- 8 个可编程中断优先级级别（可屏蔽中断）
- 依据优先级进行复合中断控制
- 针对每个可屏蔽中断请求，可以指定屏蔽选项。
- 噪声消除，边沿检测和外部中断请求信号有效沿规范。

○ 异常

- 软件异常： 32 个异常源
- 异常陷阱： 2 个异常源（非法的操作码异常，调试陷阱）

中断/异常源见表 15-1。

表 15-1. 中断源列表 (1/2)

类型	级别	缺省优先级	名称	触发	产生单元	异常代码	处理地址	恢复的 PC	中断控制寄存器
复位	中断	-	RESET	RESET 引脚输入 通过内部源的复位输入	RESET	0000H	00000000H	无定义	-
非屏蔽	中断	-	NMI	NMI 引脚有效沿输入	引脚	0010H	00000010H	下一个 PC	-
		-	INTWDT2	WDT2 溢出	WDT2	0020H	00000020H	注 1	-
软件异常	异常	-	TRAP0n ^{注 2}	TRAP 指令	-	004nH ^{注 2}	00000040H	下一个 PC	-
		-	TRAP1n ^{注 2}	TRAP 指令	-	005nH ^{注 2}	00000050H	下一个 PC	-
异常陷阱	异常	-	ILGOP/ DBG0	非法操作数/ DBTRAP 指令	-	0060H	00000060H	下一个 PC	-
可屏蔽	中断	0	INTLVI	低电压检测	POCLVI	0080H	00000080H	下一个 PC	LVIIIC
		1	INTP0	外部中断引脚输入边沿检测 (INTP0)	引脚	0090H	00000090H	下一个 PC	PIC0
		2	INTP1	外部中断引脚输入边沿检测 (INTP1)	引脚	00A0H	000000A0H	下一个 PC	PIC1
		3	INTP2	外部中断引脚输入边沿检测 (INTP2)	引脚	00B0H	000000B0H	下一个 PC	PIC2
		4	INTP3	外部中断引脚输入边沿检测 (INTP3)	引脚	00C0H	000000C0H	下一个 PC	PIC3
		5	INTP4	外部中断引脚输入边沿检测 (INTP4)	引脚	00D0H	000000D0H	下一个 PC	PIC4
		6	INTP5	外部中断引脚输入边沿检测 (INTP5)	引脚	00E0H	000000E0H	下一个 PC	PIC5
		7	INTP6	外部中断引脚输入边沿检测 (INTP6)	引脚	00F0H	000000F0H	下一个 PC	PIC6
		8	INTP7	外部中断引脚输入边沿检测 (INTP7)	引脚	0100H	00000100H	下一个 PC	PIC7
		9	INTTQ0OV	TMQ0 溢出	TMQ0	0110H	00000110H	下一个 PC	TQ0OVIC
		10	INTTQ0CC0	TMQ0 捕捉 0/比较 0 匹配	TMQ0	0120H	00000120H	下一个 PC	TQ0CCIC0
		11	INTTQ0CC1	TMQ0 捕捉 1/比较 1 匹配	TMQ0	0130H	00000130H	下一个 PC	TQ0CCIC1
		12	INTTQ0CC2	TMQ0 捕捉 2/比较 2 匹配	TMQ0	0140H	00000140H	下一个 PC	TQ0CCIC2
		13	INTTQ0CC3	TMQ0 捕捉 3/比较 3 匹配	TMQ0	0150H	00000150H	下一个 PC	TQ0CCIC3
		14	INTTP0OV	TMP0 溢出	TMP0	0160H	00000160H	下一个 PC	TP0OVIC
		15	INTTP0CC0	TMP0 捕捉 0/比较 0 匹配	TMP0	0170H	00000170H	下一个 PC	TP0CCIC0
		16	INTTP0CC1	TMP0 捕捉 1/比较 1 匹配	TMP0	0180H	00000180H	下一个 PC	TP0CCIC1
		17	INTTP1OV	TMP1 溢出	TMP1	0190H	00000190H	下一个 PC	TP1OVIC
		18	INTTP1CC0	TMP1 捕捉 0/比较 0 匹配	TMP1	01A0H	000001AH	下一个 PC	TP1CCIC0
		19	INTTP1CC1	TMP1 捕捉 1/比较 1 匹配	TMP1	01B0H	000001B0H	下一个 PC	TP1CCIC1
		20	INTTP2OV	TMP2 溢出	TMP2	01C0H	000001C0H	下一个 PC	TP2OVIC
21	INTTP2CC0	TMP2 捕捉 0/比较 0 匹配	TMP2	01D0H	000001D0H	下一个 PC	TP2CCIC0		

注 1. 在 INTWDT2 例子中的恢复，敬请参阅 15.2.2 (2) INTWDT2 信号。
 2. n = 0H ~ FH

表 15-1. 中断源列表 (2/2)

类型	级别	缺省优先级	名称	触发	产生单元	异常代码	处理地址	恢复的 PC	中断控制寄存器
可屏蔽	中断	22	INTTP2CC1	TMP2 捕捉 1/比较 1 匹配	TMP2	01E0H	000001E0H	下一个 PC	TP2CCIC1
		23	INTTP3OV	TMP3 溢出	TMP3	01F0H	000001F0H	下一个 PC	TP3OVIC
		24	INTTP3CC0	TMP3 捕捉 0/比较 0 匹配	TMP3	0200H	00000200H	下一个 PC	TP3CCIC0
		25	INTTP3CC1	TMP3 捕捉 1/比较 1 匹配	TMP3	0210H	00000210H	下一个 PC	TP3CCIC1
		26	INTTM0EQ0	TMM0 比较匹配	TMM0	0220H	00000220H	下一个 PC	TM0EQIC0
		27	INTCB0R	CSIB0 接收完成	CSIB0	0230H	00000230H	下一个 PC	CB0RIC
		28	INTCB0T	CSIB0 连续发送写入使能	CSIB0	0240H	00000240H	下一个 PC	CB0TIC
		29	INTCB1R	CSIB1 接收完成	CSIB1	0250H	00000250H	下一个 PC	CB1RIC
		30	INTCB1T	CSIB1 连续发送写入使能	CSIB1	0260H	00000260H	下一个 PC	CB1TIC
		31	INTUA0R	UARTA0 接收完成	UARTA0	0270H	00000280H	下一个 PC	UA0RIC
		32	INTUA0T	UARTA0 发送使能	UARTA0	0280H	00000280H	下一个 PC	UA0TIC
		33	INTUA1R	UARTA1 接收完成/UARTA1 接收错误	UARTA1	0290H	00000290H	下一个 PC	UA1RIC
		34	INTUA1T	UARTA1 发送使能	UARTA1	02A0H	000002A0H	下一个 PC	UA1TIC
		35	INTAD	A/D 转换完成	A/D	02BH	000002B0H	下一个 PC	ADIC
		36	INTKR	按键返回中断请求	KR	0300H	00000300H	下一个 PC	KRIC
		37	INTWTI	钟表定时器间隔	WT	0310H	00000310H	下一个 PC	WTIIC
		38	INTWT	钟表定时器参考时间	WT	0320H	00000320H	下一个 PC	WTIC
		39	INTP8	外部中断引脚输入边沿检测 (INTP8)	引脚	0330H	00000330H	下一个 PC	PIC8
		40	INTP9	外部中断引脚输入边沿检测 (INTP9)	引脚	0340H	00000340H	下一个 PC	PIC9
		41	INTP10	外部中断引脚输入边沿检测 (INTP10)	引脚	0350H	00000350H	下一个 PC	PIC10
		42	INTTQ1OV	TMQ1 溢出	TMQ1	0360H	00000360H	下一个 PC	TQ1OVIC
		43	INTTQ1CC0	TMQ1 捕捉 0/比较 0 匹配	TMQ1	0370H	00000370H	下一个 PC	TQ1CCIC0
		44	INTTQ1CC1	TMQ1 捕捉 1/比较 1 匹配	TMQ1	0380H	00000380H	下一个 PC	TQ1CCIC1
		45	INTTQ1CC2	TMQ1 捕捉 2/比较 2 匹配	TMQ1	0390H	00000390H	下一个 PC	TQ1CCIC2
		46	INTTQ1CC3	TMQ1 捕捉 3/比较 3 匹配	TMQ1	03A0H	000003A0H	下一个 PC	TQ1CCIC3
		47	INTUA2R	UARTA2 接收完成/错误	UARTA2	03B0H	000003B0H	下一个 PC	UA2RIC
		48	INTUA2T	UARTA2 发送使能	UARTA2	03C0H	000003C0H	下一个 PC	UA2TIC
		49	INTDMA0	DMA0 发送完成	DMA	0410H	00000410H	下一个 PC	DMAIC0
		50	INTDMA1	DMA1 发送完成	DMA	0420H	00000420H	下一个 PC	DMAIC1
		51	INTDMA2	DMA2 发送完成	DMA	0430H	00000430H	下一个 PC	DMAIC2
		52	INTDMA3	DMA3 发送完成	DMA	0440H	00000440H	下一个 PC	DMAIC3

- 备注**
1. 缺省优先级: 表示当两个或更多的可屏蔽中断请求同时发生时, 它们的优先顺序。最高值为 0。
非屏蔽中断的优先级顺序为 INTWDT2 > NMI。
 - 恢复的 PC: 当中断服务开始时被保存到 EIPC, FEPC, 或是 DBPC 的程序计数器(PC)的值。但是, 当非屏蔽中断或者可屏蔽中断被确认, 且下列指令之一正在被执行时, 恢复的 PC 不会成为下一个 PC (如果在中断执行期间确认中断, 则该操作停止并在中断服务结束后恢复)。
 - 载入指令 (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
 - 除法指令(DIV, DIVH, DIVU, DIVHU)
 - 预备, 处理指令 (仅限中断在堆栈指针更新之前产生的情况)
 - 下一个 PC: 在中断/异常处理之后的开始处理的 PC 的值。
 2. 当一个非法的操作代码异常发生时, 非法指令的执行地址通过保存的 PC 值减去 4 得出。

15.2 非屏蔽中断

非屏蔽中断请求会被无条件的确认，即使是在禁止中断的条件下(DI)。一个非屏蔽中断不会受到优先级的控制，并且相对于其它的中断请求而言具有绝对的优先权。

在本产品中有下列两种非屏蔽中断请求信号。

- NMI 引脚输入 (NMI)
- 非屏蔽中断请求信号可以通过看门狗定时器溢出产生(INTWDT2)

NMI 引脚的有效沿可以选择以下四种类型：“上升沿”，“下降沿”，“上升沿和下降沿”和“无边沿检测”。

T NMI 引脚的功能将通过设置 PMC0.PMC02 位为 1 使能，并且 INTF0.INTF02 位和 INTR0.INTR02 位达到理想值时被激活，并且它将指定引脚的有效沿。

当 WDTM2.WDM21 位和 WDTM2.WDM20 位设为 01 时，由看门狗定时器 2 溢出而产生的非屏蔽中断请求信号 (INTWDT2)生效。

如果两个或是更多的非屏蔽中断请求同时产生，具有较高优先级的中断首先生效，如下所示（具有较低优先级的中断请求信号将被忽略）。

INTWDT2 > NMI

如果在非屏蔽中断正在进行的同时，发出一个新的 NMI 或者 INTWDT2 请求信号，将按照如下方式进行。

(1) 如果在非屏蔽中断正在进行的同时，发出一个新的 NMI 请求信号

新的 NMI 请求信号被保持，忽略 PSW.NP 位的值。而新的 NMI 请求信号会在当前的 NMI 运行完以后被确认（在 RETI 指令执行后）。

(2) 如果在非屏蔽中断正在进行的同时，发出一个 INTWDT2 请求信号

当存在正在进行的 NMI 时，如果 NP 位的值是(1)，则 INTWDT2 请求信号被保持。被保持的 INTWDT2 请求信号会在当前的 NMI 运行完以后被确认（在 RETI 指令执行后）。

当存在正在进行的 NMI 时，如果 NP 位为 (0)，则立即执行新产生的 INTWDT2 请求信号 (NMI 服务停止)。

注意事项 由非屏蔽中断请求信号(INTWDT2)执行的非屏蔽中断服务，参考 15.2.2 (2) INTWDT2 信号。

图 15-1. 非屏蔽中断请求信号确认操作(1/2)

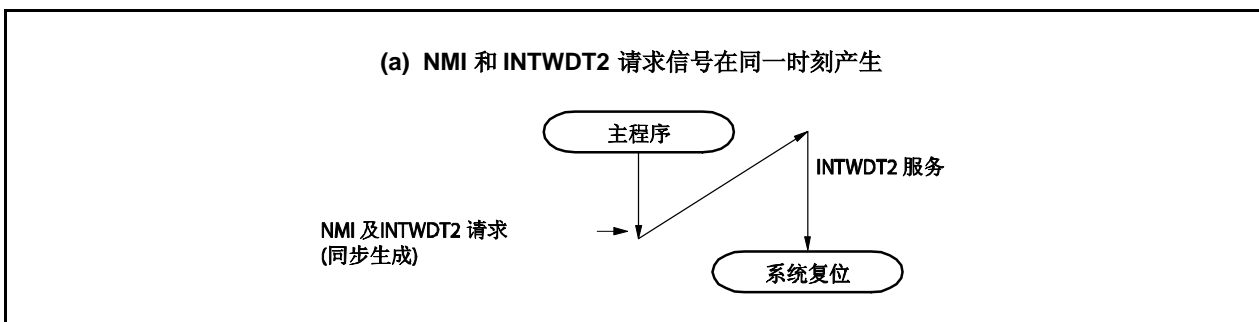
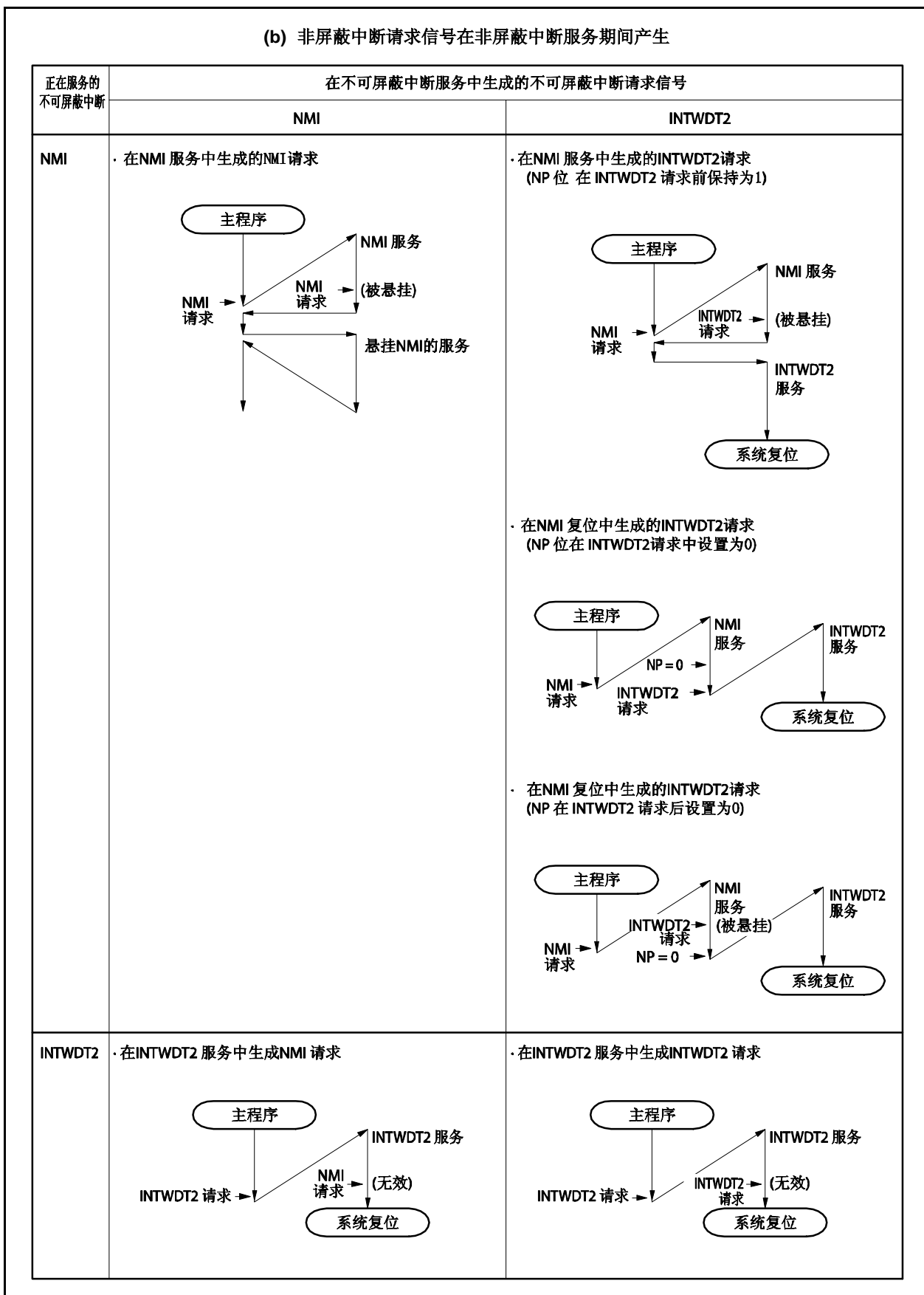


图 15-1. 非屏蔽中断请求信号确认操作(2/2)



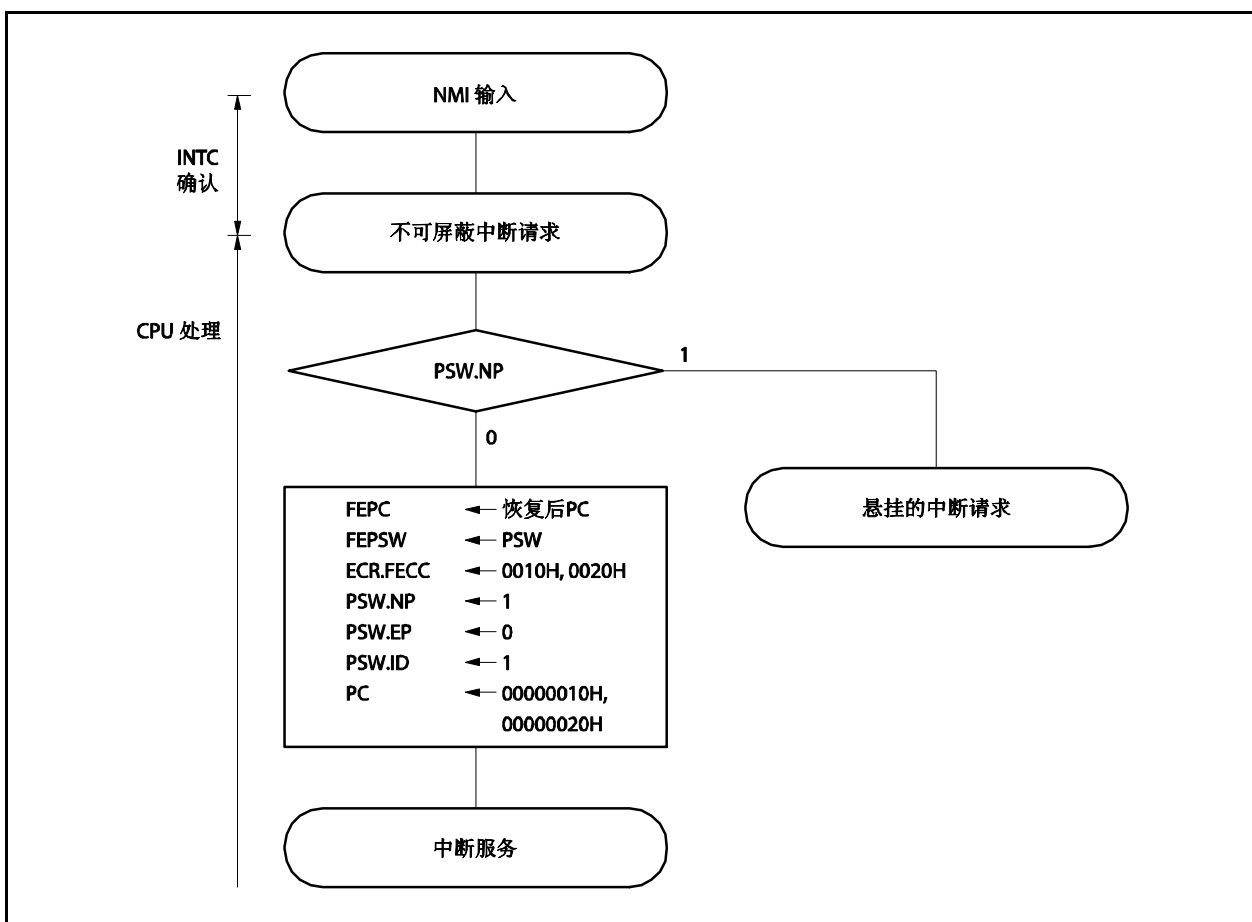
15.2.1 操作

如果产生一个非屏蔽中断请求信号，CPU 执行下列处理，并且将操作权转移给处理例程。

- <1> 将恢复 PC 保存的 FEPC。
- <2> 将当前的 PSW 保存到 FEPSW。
- <3> 向 ECR 的高半字(FECC)写入异常代码 (0010H, 0020H)。
- <4> 将 PSW.NP 位和 PSW.ID 位置 1 并将 PSW.EP 位清零。
- <5> 在 PC 中设置相应的非屏蔽中断的处理地址(00000010H, 00000020H)，然后转移操作权。

非屏蔽中断的服务配置如图 15-2 所示。

图 15-2. 非屏蔽中断的服务配置



15.2.2 恢复

(1) 通过 NMI 引脚输入

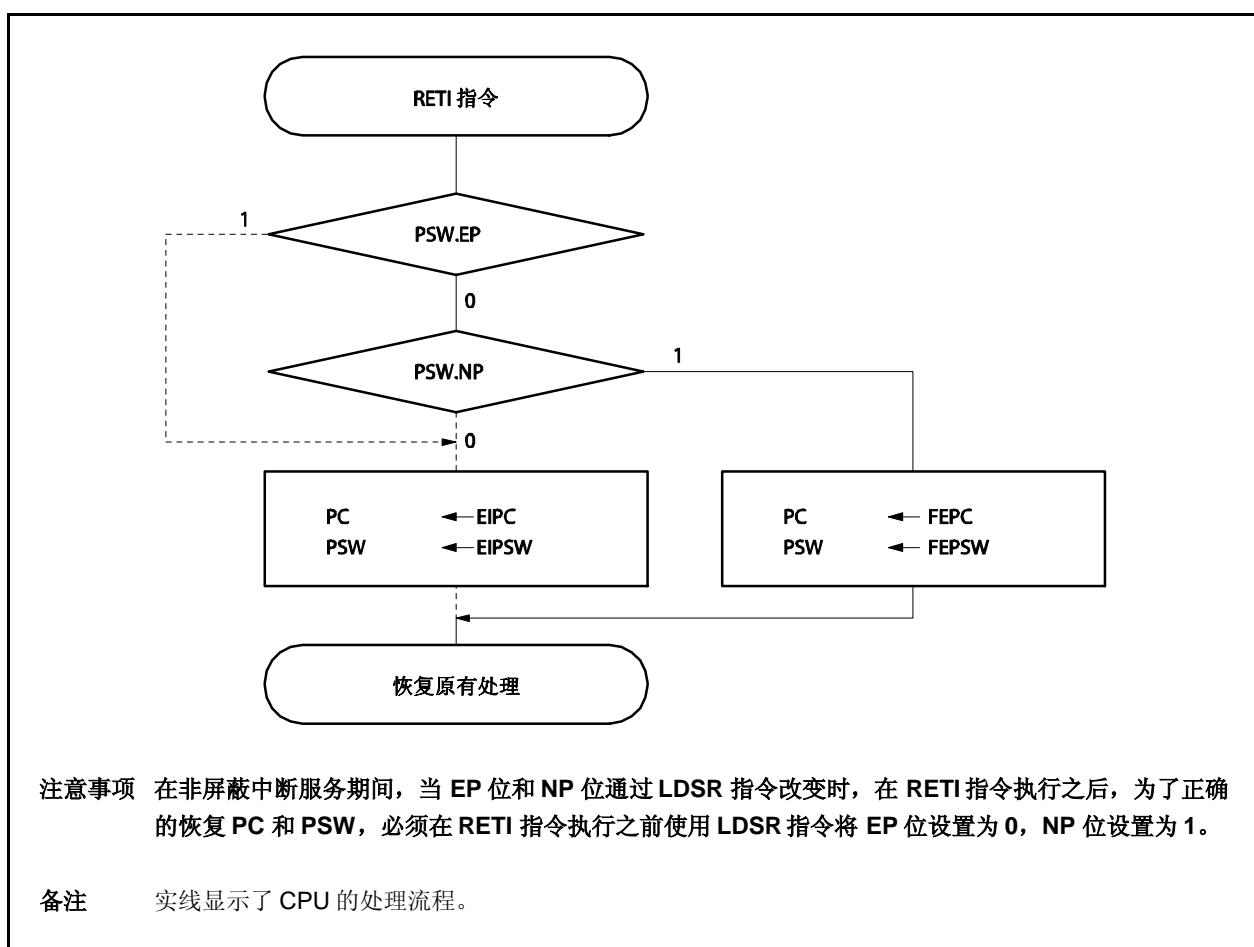
通过 RETI 指令从 NMI 服务中执行恢复。

当执行 RETI 指令时，CPU 执行下列处理，并且将操作权转移给 PC 存储的地址。

- <1> 分别从 FEPC 和 FEPSW 中加载被保存的 PC 和 PSW 的值，因为 PSW.EP 位的值是 0 且 PSW.NP 的值为 1。
- <2> 操作权转移给恢复的 PC 和 PSW 的地址。

图 15-3 示例说明了 RETI 指令是如何处理的。

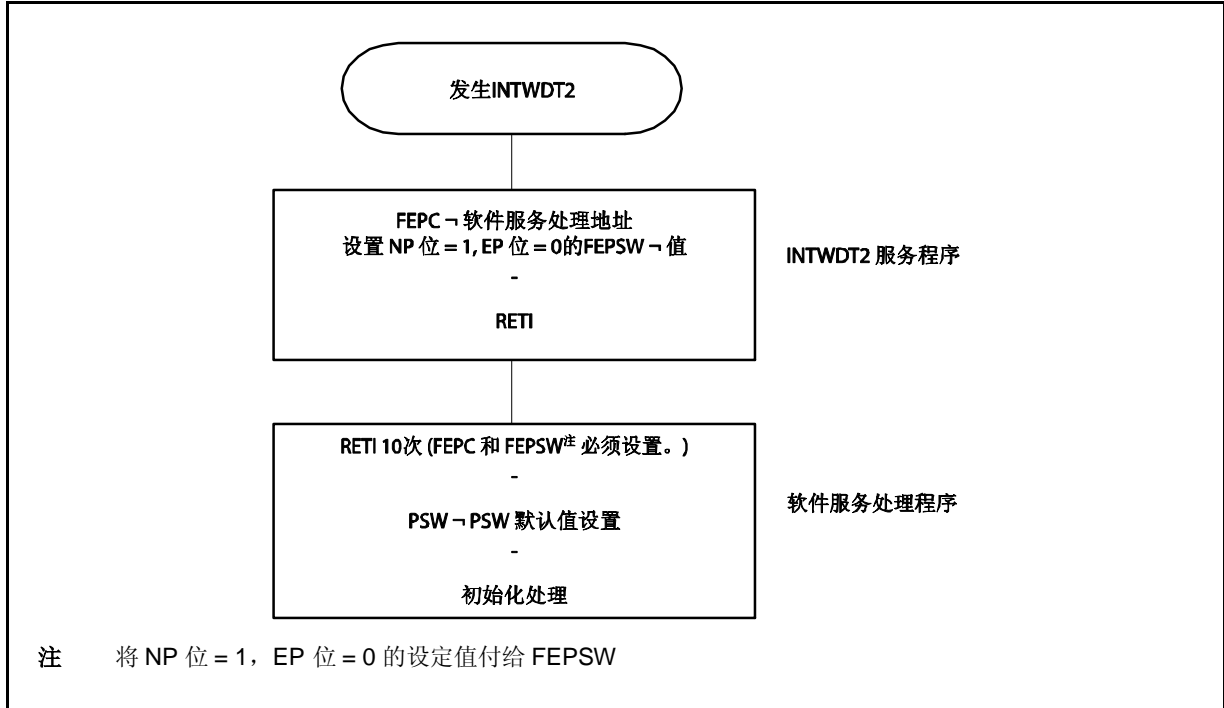
图 15-3. RETI 指令处理过程



(2) 通过 INTWDT2 信号

禁止使用 RETI 指令从非屏蔽中断服务(通过非屏蔽中断请求 INTWDT2 产生)中恢复。执行下列软件复位处理。

图 15-4. 软件复位处理



15.2.3 NP 标志

NP 标志是一个状态标志，它指示了正在执行非屏蔽中断。

当非屏蔽中断请求被确认时会设置该标志，并且还可以保证多重非屏蔽中断的情况出现。

复位后: 00000020H

	31		8	7	6	5	4	3	2	1	0
PSW	0			NP	EP	ID	SAT	CY	OV	S	Z

NP	NMI 中断服务状态
0	无NMI 中断服务
1	当前正在使用NMI 中断服务

15.3 可屏蔽中断

可屏蔽中断请求信号可以通过中断控制寄存器屏蔽。V850ES/HG2 拥有 53 个可屏蔽中断源。

如果有两个或者更多的可屏蔽中断请求信号同时产生，它们将依照缺省优先级的顺序被确认。除了缺省优先级之外，还有八个优先级等级可以使用，这需要使用中断控制寄存器（可变成优先级控制）。

如果一个中断请求信号被确认，将禁止对其它可屏蔽中断请求信号的确认，从而进入了中断禁止状态(DI)。

当在中断服务程序中执行了 EI 指令时，进入中断允许状态(EI)，这将允许具有比正在进行处理的当前中断请求信号更高优先级（通过中断控制寄存器设定）的中断。注意只有优先级更高的中断有这种功能，具有相同优先级的中断不能实现嵌套。

但是为了允许多重中断，在执行 EI 指令之前需要将 EIPC 和 EIPSW 保存到内存或是通用寄存器当中，并且在 RETI 指令恢复 EIPC 和 EIPSW 的原始值之前执行 DI 指令。

15.3.1 操作

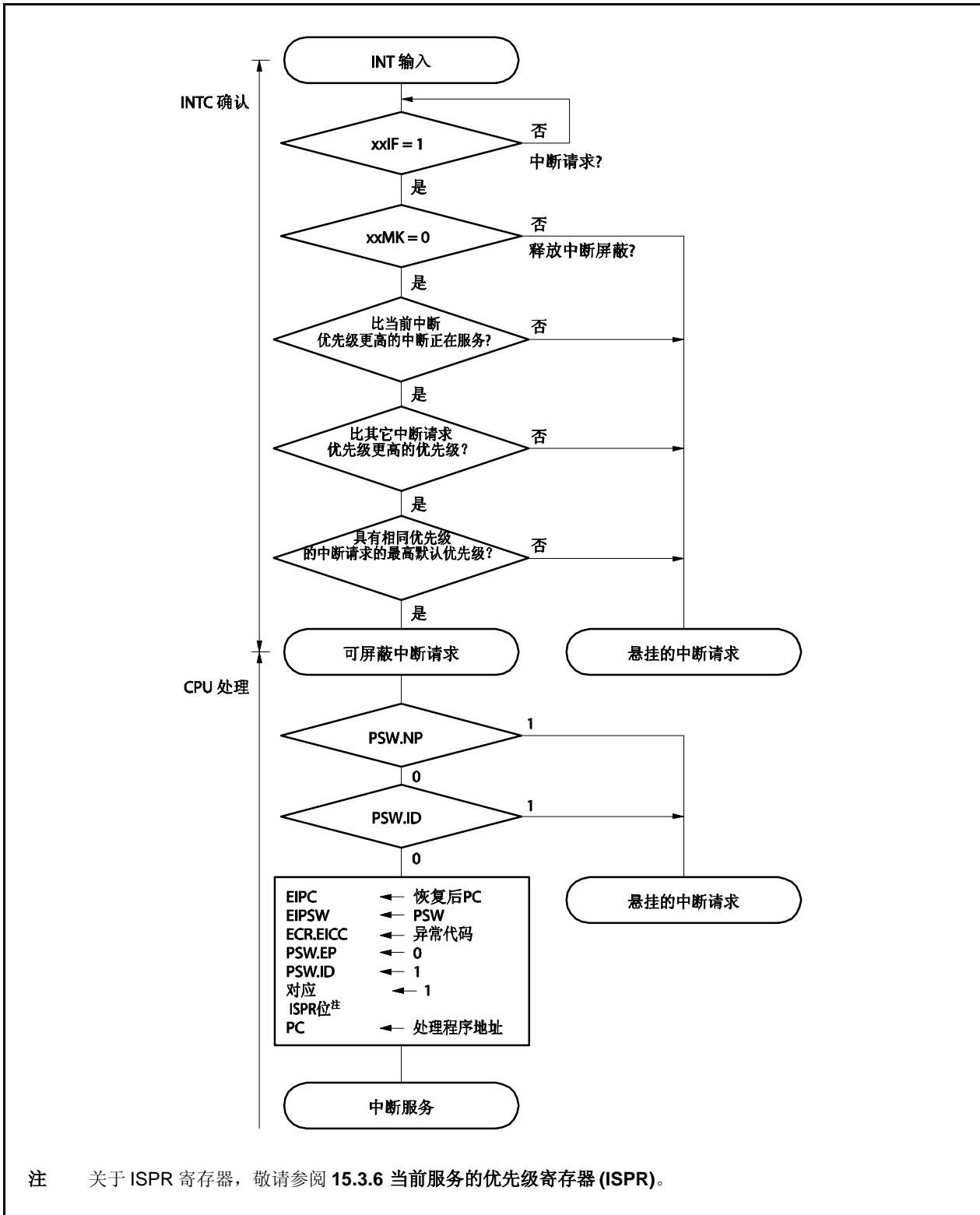
如果产生了可屏蔽中断，CPU 将执行下列处理，并且将操作权转移给处理例程。

- <1> 将恢复的 PC 保存到 EIPC。
- <2> 将当前的 PSW 保存到 EIPSW。
- <3> 向 ECR 的低半字 (EICC)写入一个异常代码。
- <4> 将 PSW.ID 位置 1 且将 PSW.EP 位清零。
- <5> 将相应中断的处理地址设置给 PC，并且转移控制权。

通过 INTC 屏蔽的可屏蔽中断请求信号和在另一个中断正在执行过程中（当 PSW.NP 位=1 或 PSW.ID 位=1）产生的可屏蔽中断请求信号在 INTC 中被保持。在这种情况下，一个新的可屏蔽中断服务开始是与可屏蔽中断请求的优先级设定是一致的。这需要可屏蔽中断未被屏蔽，而且没有通过 RETI 指令或是 LDSR 指令将 NP 位和 ID 位清零。

可屏蔽中断是如何执行的将在下面示例说明。

图 15-5. 可屏蔽中断服务



15.3.2 恢复

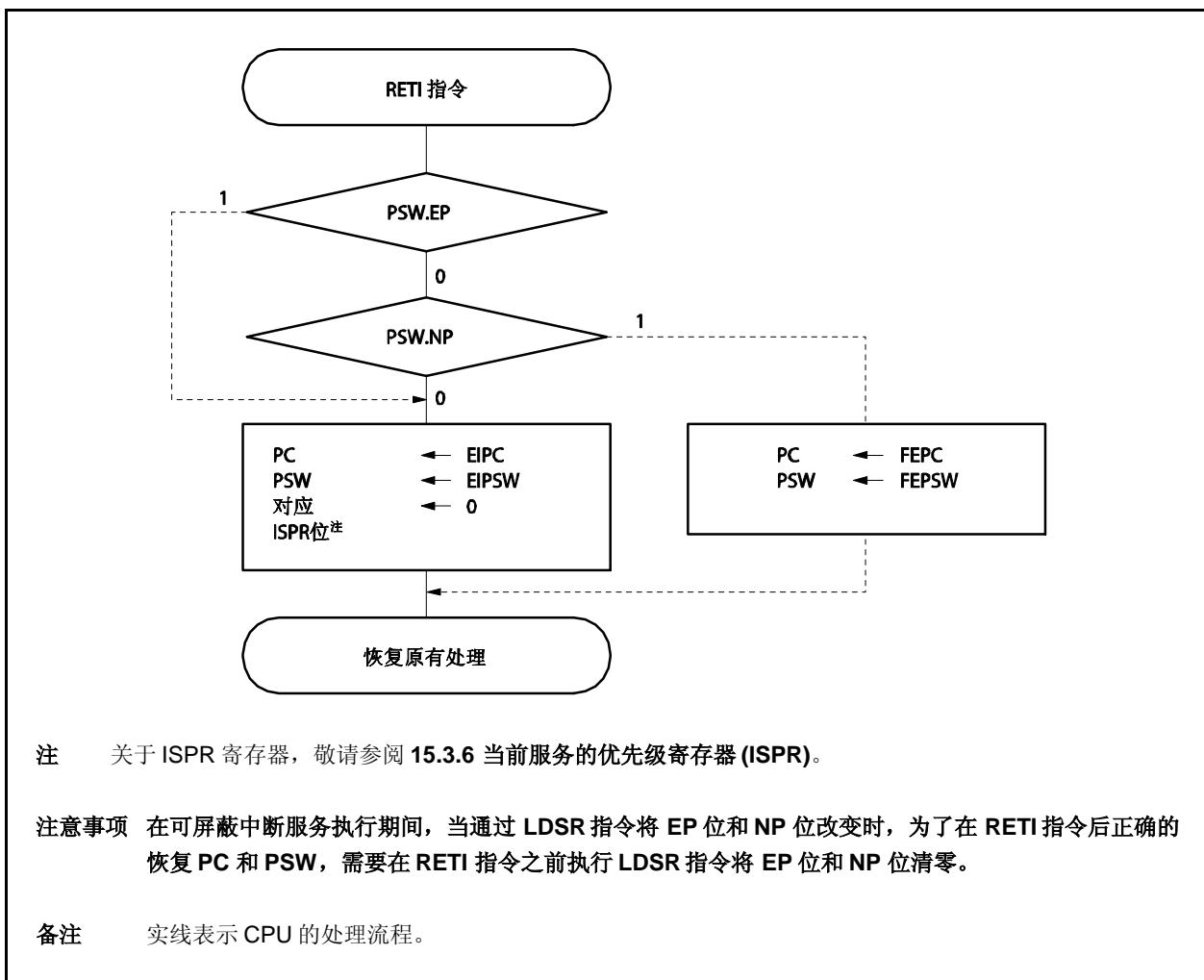
通过执行 RETI 指令能够从可屏蔽中断服务中恢复。

当 RETI 指令执行后，CPU 将执行下列步骤，且将操作权转移给恢复的 PC 地址。

- <1> 从 EIPC 和 EIPSW 中加载恢复的 PC 和 PSW，这是因为 PSW.EP 位为 0 并且 PSW.NP 位为 0。
- <2> 将操作权转移给恢复 PC 和 PSW 的地址。

图 15-6 示例说明 RETI 指令的处理过程。

图 15-6. RETI 指令处理过程



15.3.3 可屏蔽中断的优先级

INTC 执行多重中断服务，这是指当一个中断服务正在进行的时候确认另外一个中断。多重中断可以通过优先级来控制。

一共有两种类型的优先级别控制：基于缺省优先级别的控制和基于可编程优先级别的控制，后者是通过在中断控制(**xxICn**)寄存器的中断优先级指定位(**xxPRn**)的设置来实现的。当两个具有相同优先级的中断同时产生时，中断请求信号将依照预先指定给每一种中断类型的优先级别（缺省优先级别）顺序进行中断服务。关于更多细节，敬请参阅**表 15-1 中断/异常源列表**。可编程的优先级通过设定优先级别指定标志分八个等级对用户化的中断请求信号进行控制。

注意当一个中断请求信号被确认后，**PSW.ID** 标志将自动置 1。因此，当使用复合中断时，需要预先将 **ID** 标志清零来设定中断允许模式（例如在中断服务程序中设置 **EI** 指令）。

备注 **xx**: 表示任意的外围单元的名称（参考**表 15-2 中断控制寄存器(xxICn)**）
 n: 外围单元编号（参考**表 15-2 中断控制寄存器(xxICn)**）。

图 15-7. 处理过程中当在一个中断正在服务时，发生另一个中断请求信号的示例(1/2)

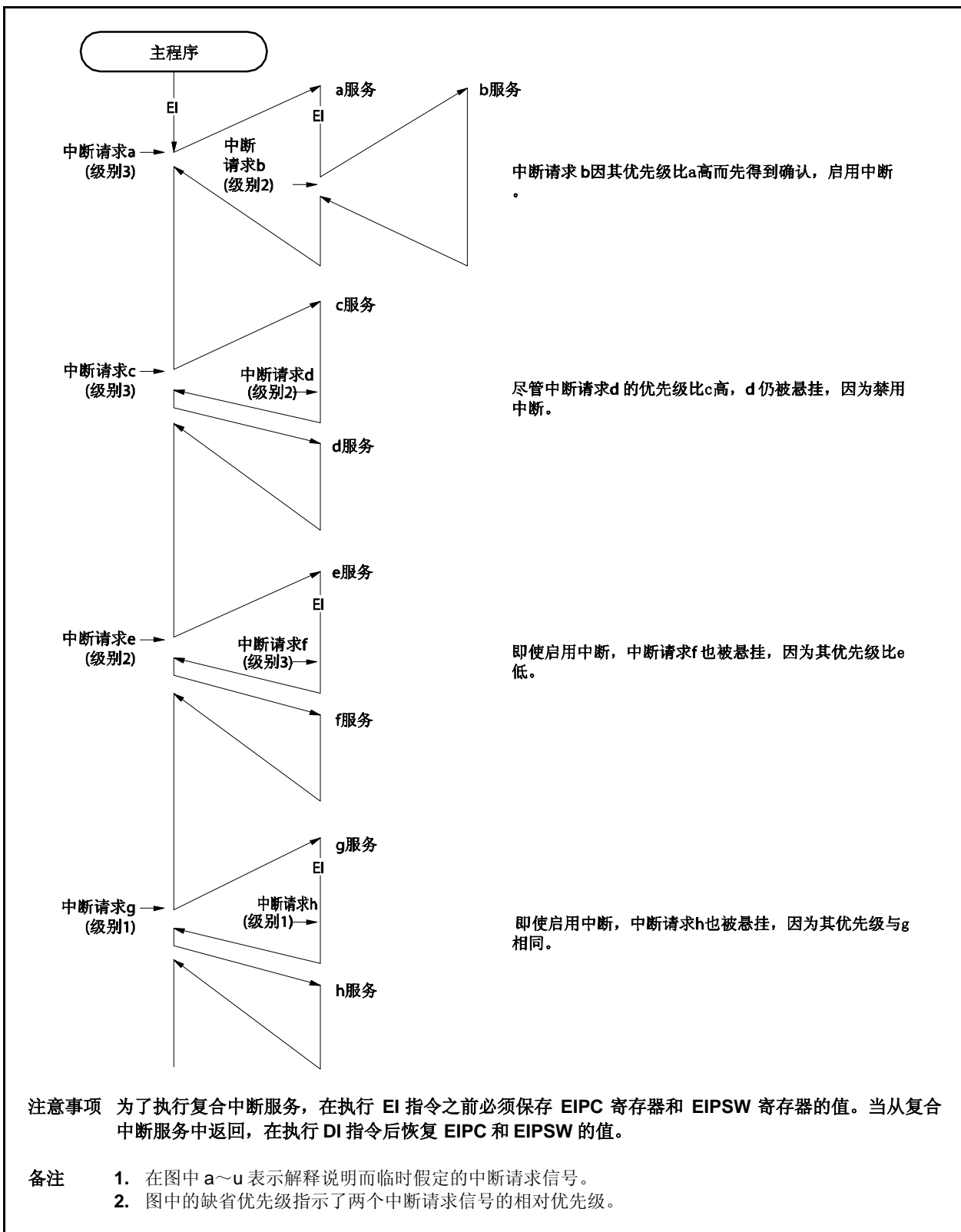


图 15-7. 处理过程中当在一个中断正在服务时，发生另一个中断请求信号的示例(2/2)

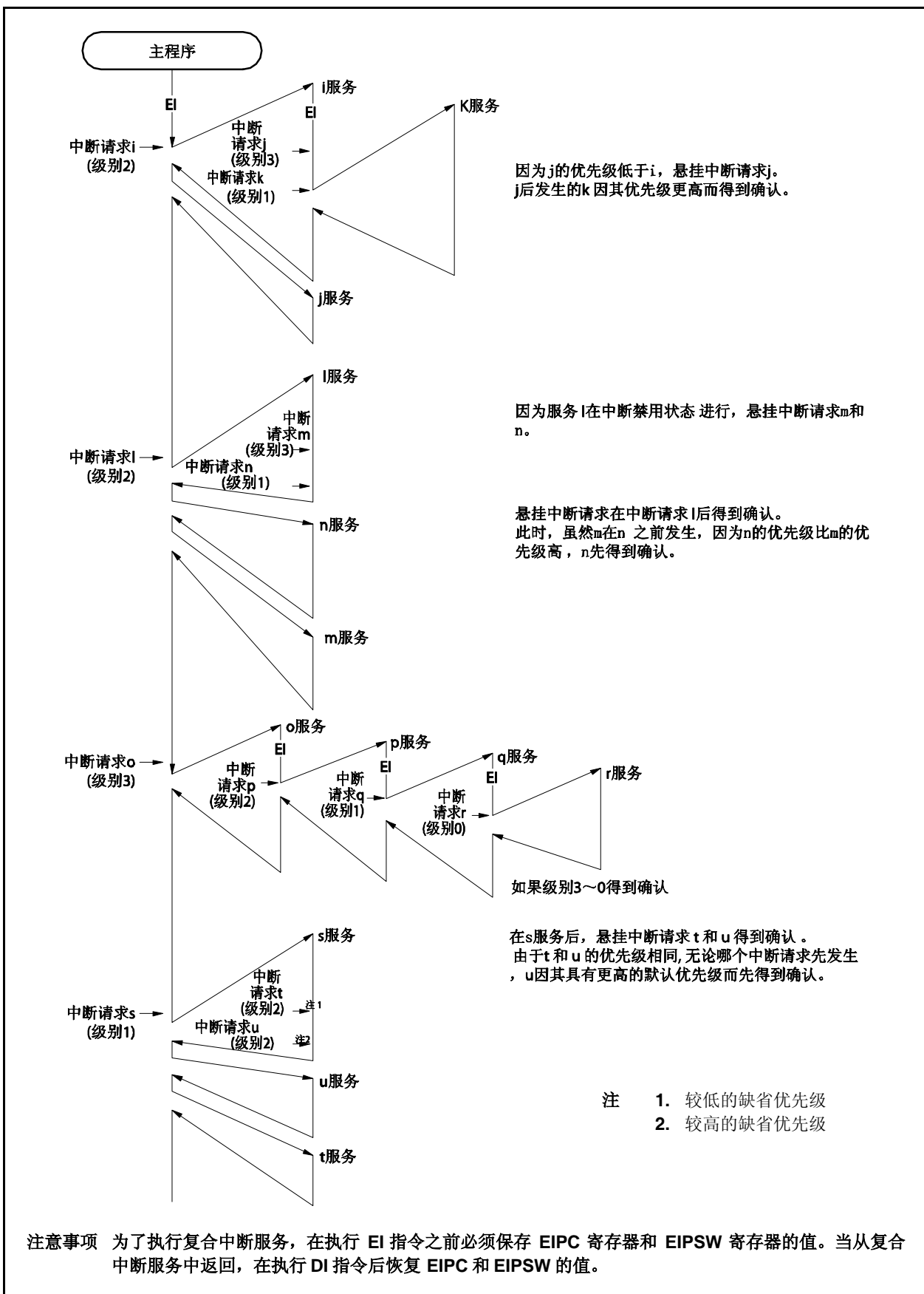
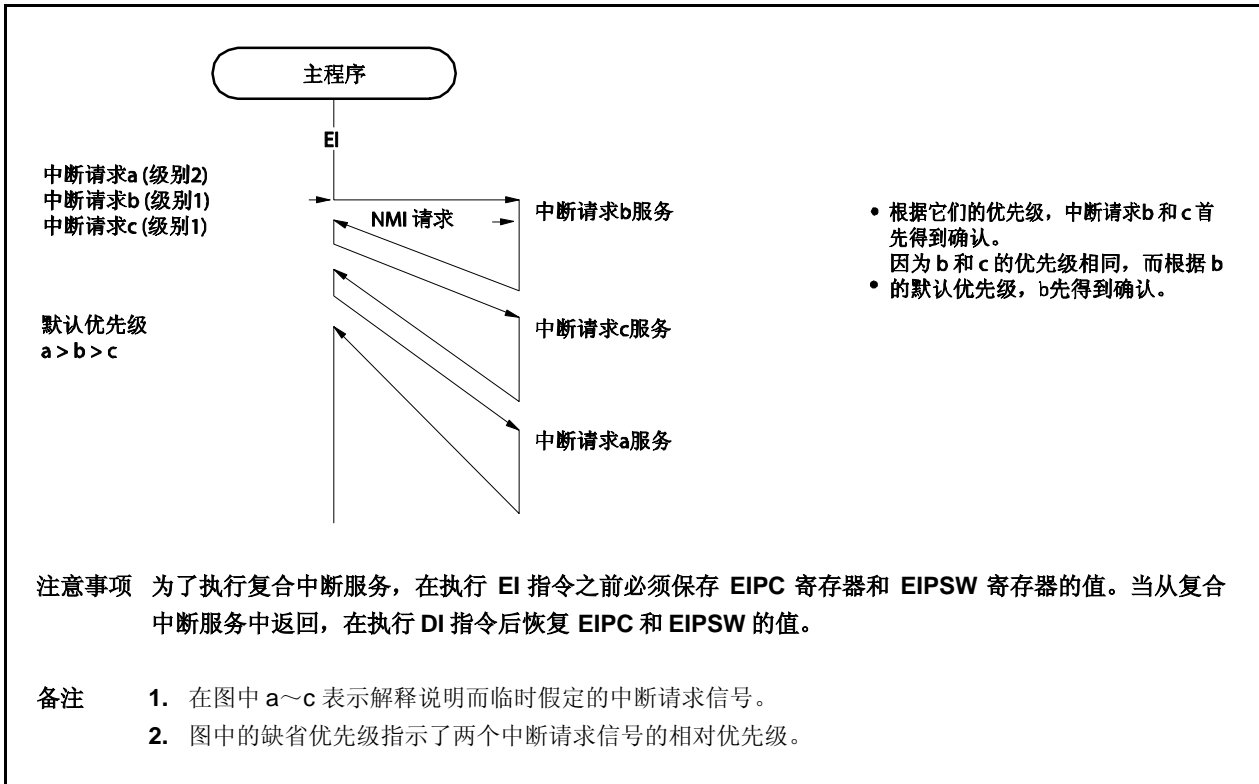


图 15-8. 服务中断请求信号同时发生示例



15.3.4 中断控制寄存器 (xxICn)

xxICn 寄存器被分配给每一个中断请求信号（可屏蔽中断）并且设置每一个可屏蔽中断请求控制条件。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 47H。

注意事项 禁止中断 (DI)或者屏蔽中断来读取 xxICn.xxIFn 位。如果在允许中断或是中断未被屏蔽的时候，读取 xxIFn 位，确认中断且读取该位冲突时数据可能有误。

复位后：47H R/W 地址：FFFF110H ~FFFF188H

	7	6	5	4	3	2	1	0
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	中断请求标志 ^注
0	没有发出中断请求
1	发出中断请求

xxMKn	中断屏蔽标志
0	启用中断服务
1	禁用中断服务(待定)

xxPRn2	xxPRn1	xxPRn0	中断优先级规格无
0	0	0	0 级别(最高)。
0	0	1	1 级别。
0	1	0	2 级别。
0	1	1	3 级别
1	0	0	4 级别。
1	0	1	5 级别。
1	1	0	6 级别。
1	1	1	7 级别(最低)。

注 如果确认中断信号后，xxIFn 标志通过硬件自动复位。

备注 xx: 表示任意的外围单元的名称（参考表 15-2 中断控制寄存器 (xxICn)）
n: 外围单元编号（参考表 15-2 中断控制寄存器 (xxICn)）。

中断控制寄存器的位和地址如下所示。

表 15-2. 中断控制寄存器 (xxICn) (1/2)

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
FFFFF166H	PIC8	PIF8	PMK8	0	0	0	PPR82	PPR81	PPR80
FFFFF168H	PIC9	PIF9	PMK9	0	0	0	PPR92	PPR91	PPR90
FFFFF16AH	PIC10	PIF10	PMK10	0	0	0	PPR102	PPR101	PPR100

表 15-2. 中断控制寄存器 (xxICn) (2/2)

地址	寄存器	位							
		7	6	5	4	3	2	1	0
FFFFFF16CH	TQ1OVIC	TQ1OVIF	TQ1OVMK	0	0	0	TQ1OVPR2	TQ1OVPR1	TQ1OVPR0
FFFFFF16EH	TQ1CCIC0	TQ1CCIF0	TQ1CCMK0	0	0	0	TQ1CCPR2	TQ1CCPR1	TQ1CCPR0
FFFFFF170H	TQ1CCIC1	TQ1CCIF1	TQ1CCMK1	0	0	0	TQ1CCPR12	TQ1CCPR11	TQ1CCPR10
FFFFFF172H	TQ1CCIC2	TQ1CCIF2	TQ1CCMK2	0	0	0	TQ1CCPR22	TQ1CCPR21	TQ1CCPR20
FFFFFF174H	TQ1CCIC3	TQ1CCIF3	TQ1CCMK3	0	0	0	TQ1CCPR32	TQ1CCPR31	TQ1CCPR30
FFFFFF176H	UA2RIC	UA2RIF	UA2RMK	0	0	0	UA2RPR2	UA2RPR1	UA2RPR0
FFFFFF178H	UA2TIC	UA2TIF	UA2TMK	0	0	0	UA2TPR2	UA2TPR1	UA2TPR0
FFFFFF182H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFFF184H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFFF186H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFFF188H	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30

15.3.5 中断屏蔽寄存器 0~3 (IMR0~IMR3)

IMR0~IMR3 寄存器为可屏蔽中断设定中断屏蔽状态。IMR0~IMR3 寄存器的 xxMKn 位相当于 xxICn.xxMKn 位。

IMRm 寄存器支持 16 位读写方式 (m = 0 ~ 3)。

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，这些寄存器支持 8 位或 1 位读写方式(m = 0~3)。

这些寄存器复位后的值为 FFFFH。

注意事项 该设备文件定义了 xxICn.xxMKn 位为保留字。如果有其它的位使用 xxMKn 作为名称进行操作，xxICn 寄存器的内容会被写入，而不是 IMRm 寄存器（结果，IMRm 寄存器的内容也会被覆盖）。

复位后: FFFFH		R/W	地址:		IMR3 FFFF106H, IMR3L FFFF106H, IMR3H FFFF107H			
	15	14	13	12	11	10	9	8
IMR3 (IMR3H ^注)	1	1	1	DMAMK3	DMAMK2	DMAMK1	DMAMK0	1
	7	6	5	4	3	2	1	0
IMR3L	1	1	1	UA2TMK	UA2RMK	TQ1CCMK3	TQ1CCMK2	TQ1CCMK1
复位后: FFFFH		R/W	地址:		IMR2 FFFF104H, IMR2L FFFF104H, IMR2H FFFF105H			
	15	14	13	12	11	10	9	8
IMR2 (IMR2H ^注)	TQ1CCMK0	TQ1OVMK	PMK10	PMK9	PMK8	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 FFFF102H, IMR1L FFFF102H, IMR1H FFFF103H			
	15	14	13	12	11	10	9	8
IMR1 (IMR1H ^注)	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 FFFF100H, IMR0L FFFF100H, IMR0H FFFF101H			
	15	14	13	12	11	10	9	8
IMR0 (IMR0H ^注)	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	启用中断服务							
1	禁用中断服务							

注 为了以 8 位或 1 位为单元读取 IMR0~IMR3 寄存器的位 8~位 15，需要将它们设定为 IMR0H~IMR3H 寄存器的位 0~位 7。

注意事项 设置 IMR3 寄存器的位 15~13 和 8~5 及 IMR2 寄存器的位 7~4 为“1”。如果这些位的设定值改变，操作将不会得到保证。

备注 xx: 表示任意的外围单元的名称（参考表 15-2 中断控制寄存器 (xxICn)）。
n: 外围单元编号（参考表 15-2 中断控制寄存器 (xxICn)）

15.3.6 当前服务的优先级寄存器(ISPR)

ISPR 寄存器当前被确认的可屏蔽中断的优先级别。当一个中断请求被确认时，寄存器中对应该中断请求信号优先级别的位置 1，且在中断服务程序执行期间保持。

当执行 RETI 指令后，对应具有最高优先级的中断请求信号的位通过硬件自动复位为 0。但是，当从非屏蔽中断服务或是异常处理中返回时，该位不会复位为 0。

该寄存器支持 8 位或 1 位只读方式。

该寄存器复位后的值为 00H。

注意事项 如果在中断允许状态(EI)对 ISPR 寄存器进行读取时确认一个中断，在寄存器的位通过确认中断的方式被设置之后，可以读取 ISPR 寄存器的值。为了在确认中断之前正确的读取 ISPR 寄存器的值，需要禁止中断(DI)。

复位后：00H R 地址：FFFF1FAH

	7	6	5	4	3	2	1	0
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	当前确认的中断优先级
0	带有优先级n的中断请求信号没被确认
1	确认带有优先级n的中断请求信号

备注 n = 0 ~ 7 (优先级别)

15.3.7 ID 标志

该标志控制可屏蔽中断的操作状态，并且存储有关允许或是禁止中断请求信号的控制信息。中断禁止标志(ID)位于 PSW 之中。

该标志复位后的值为 00000020H。

复位后：00000020H

	31		8	7	6	5	4	3	2	1	0	
PSW	0				NP	EP	ID	SAT	CY	OV	S	Z

ID	可屏蔽中断服务 ^注 的规格
0	启用可屏蔽中断请求信号确认
1	禁用可屏蔽中断请求信号确认（待定）

注 中断禁止标志(ID)功能
 当执行 DI 指令时将这一位置 1，当执行 EI 指令时将这一位置清零。当参考 PSW 时，也可以通过 RETI 指令或是 LDSR 指令修改它的值。
 非屏蔽中断请求信号和异常的确认为不会考虑这个标志的设定。当一个可屏蔽中断请求信号被确认后，ID 标志位会通过硬件自动置 1。
 在确认被禁止的时期(ID 标志 = 1)产生的中断请求信号会在 xxICn.xxIFn 位置 1，ID 标志清零时被确认。

15.3.8 看门狗定时器模式寄存器 2 (WDTM2)

该寄存器支持 8 位读写方式（关于更多细节，敬请参阅第十章 看门狗定时器 2 的功能）。

该寄存器复位后的值为 67H。

复位后：67H R/W 地址：FFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	0	0	0	0	0

WDM21	WDM20	选择看门狗定时器操作模式
0	0	停止模式
0	1	不可屏蔽中断请求模式
1	1	复位模式(初始值)

15.4 软件异常

当 CPU 执行 TRAP 指令的时候会产生一个软件的异常，且总是会被确认。

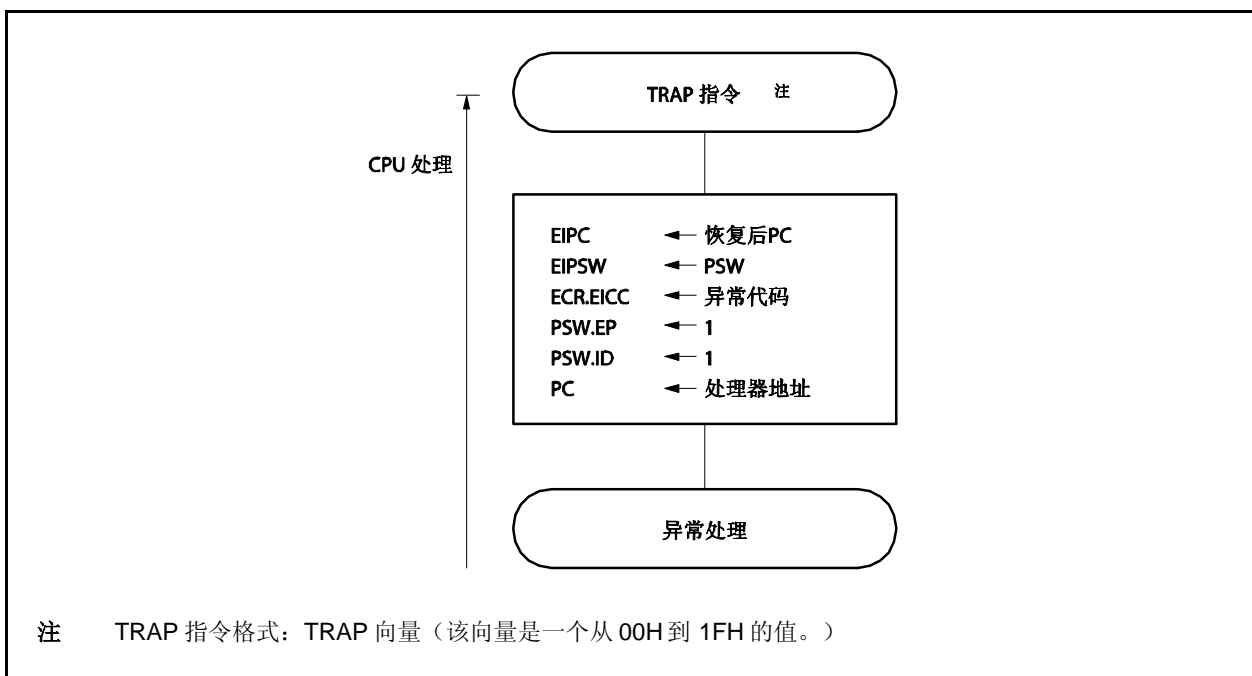
15.4.1 操作

如果一个软件异常产生，CPU 将执行下列处理过程，并且将操作权转移给处理例程。

- <2> 将当前的 PSW 保存在 EIPSW 中。
- <3> 在 ECR（中断源）的低 16 位(EICC)中写入一个异常代码。
- <4> 将 PSW.EP 位和 PSW.ID 位置 1。
- <5> 在 PC 中设置符合软件异常的处理地址(00000040H 或 00000050H)，并且转移控制权。

图 15-9 示例说明软件异常的处理过程。

图 15-9. 软件异常处理过程



处理地址是通过 TRAP 指令的操作数(向量)决定的。如果该向量是 00H~0FH，它将变为 00000040H；而如果向量的值是 10H~1FH，它将变为 00000050H。

15.4.2 恢复

通过执行 RETI 指令可以从软件异常进程中恢复。

通过执行 RETI 指令，CPU 执行下列处理过程，并且将控制权转移到恢复 PC 的地址。

- <1> 由于 PSW.EP 位为 1，从 EIPC 和 EIPSW 中加载恢复的 PC 和 PSW。
- <2> 将控制权转移到恢复 PC 和 PSW 的地址。

图 15-10 示例说明了 RETI 指令的处理过程。

图 15-10. RETI 指令处理过程

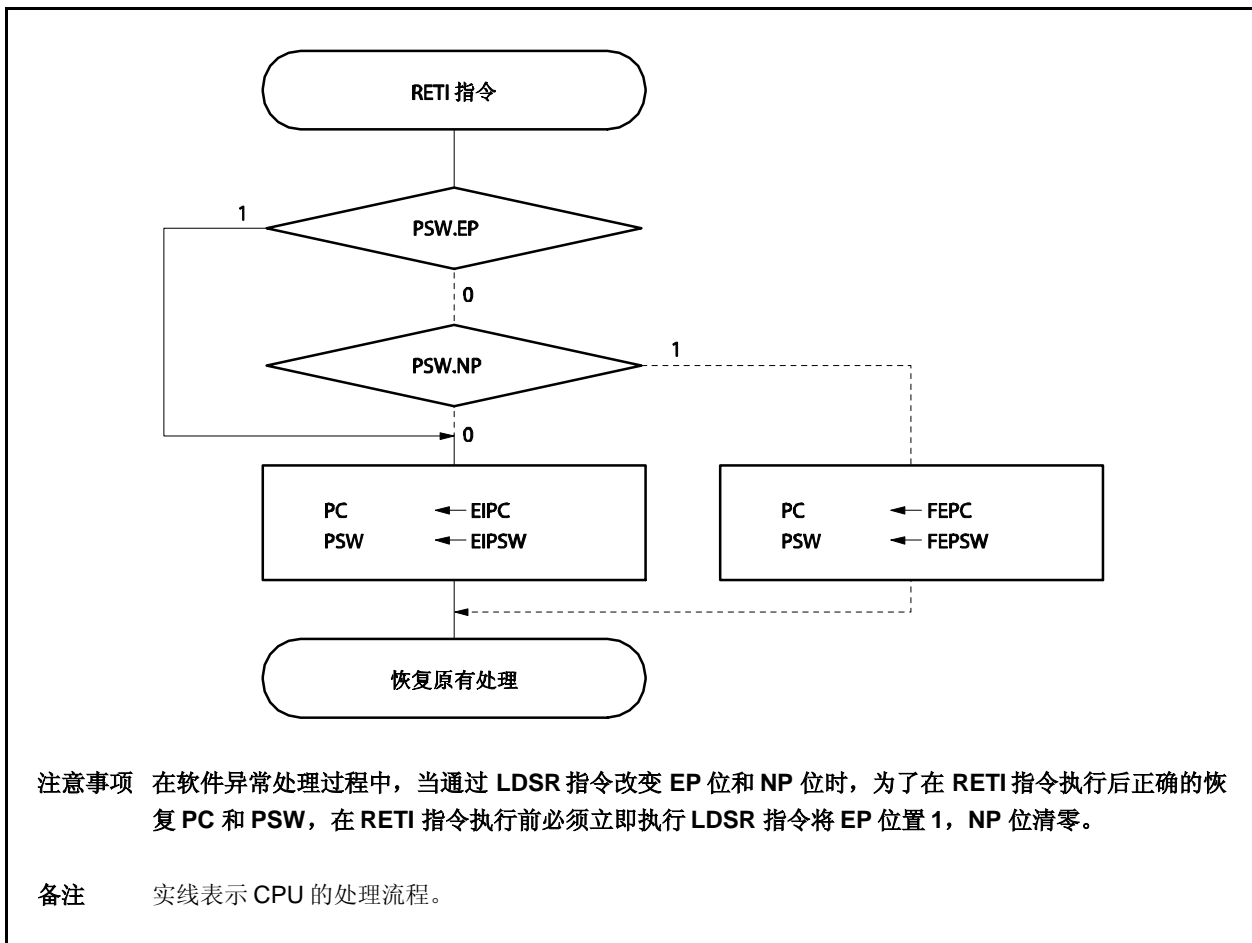
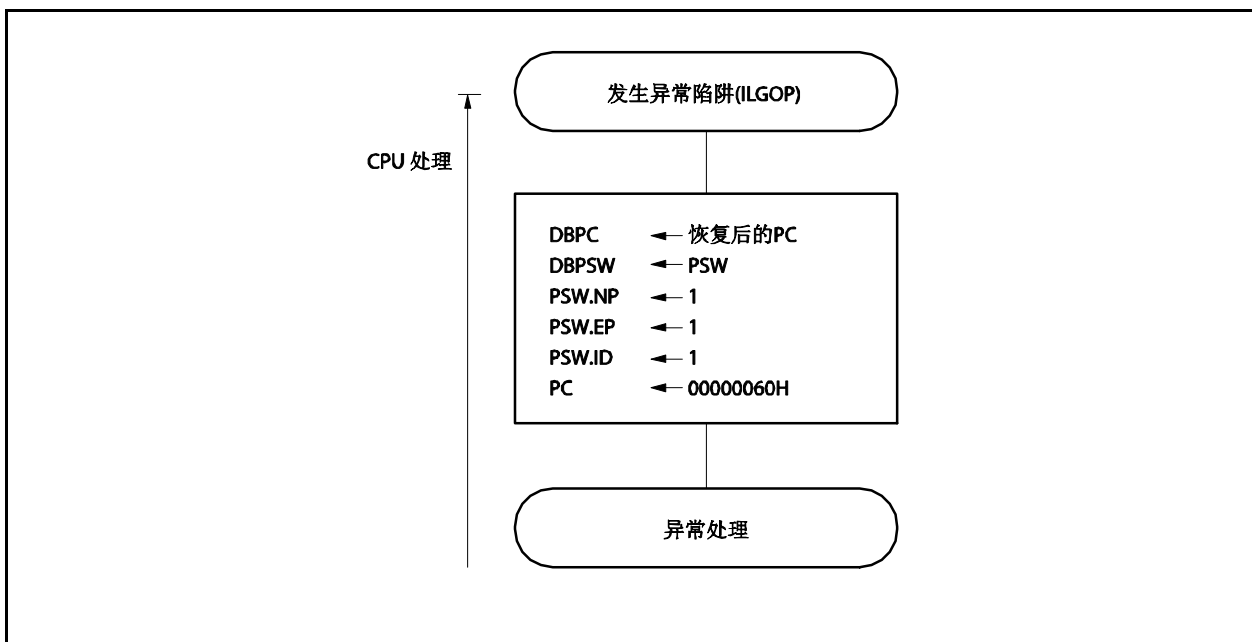


图 15-11. 异常陷阱处理过程



(2) 恢复

通过执行 DBRET 指令实现从异常陷阱中恢复。通过执行 DBRET 指令，CPU 执行下列处理过程并且控制着恢复 PC 的地址。

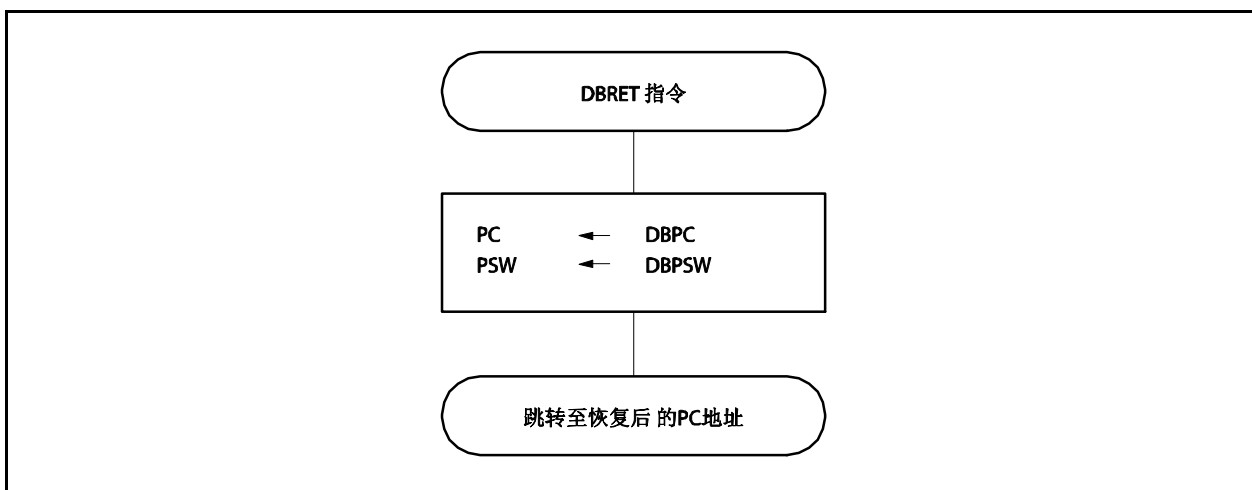
<1> 从 DBPC 和 DBPSW 中加载恢复的 PC 和 PSW。

<2> 向恢复的 PC 和 PSW 代表的地址转移控制权。

注意事项 仅在执行非法操作数和 DBRET 指令的间隔过程中，可以访问 DBPC 和 DBPSW。

图 15-12 示例说明从异常陷阱中恢复的处理过程。

图 15-12. 从异常陷阱中恢复的处理过程



15.5.2 调试陷阱

调试陷阱是一个异常，它在 DBTRAP 指令执行时产生，且总是会被确认。

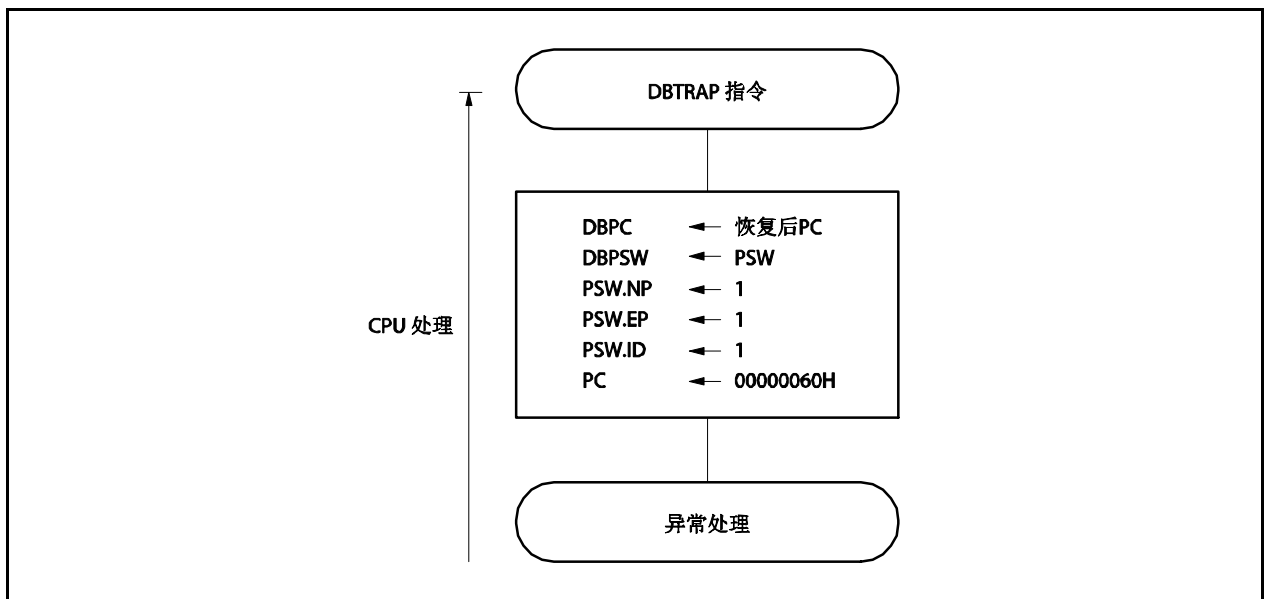
(1) 操作

在出现调试陷阱之后，CPU 将执行如下处理过程

- <1> 将恢复的 PC 保存到 DBPC。
- <2> 将当前的 PSW 保存到 DBPSW。
- <3> 将 PSW.NP 位，PSW.EP 位，和 PSW.ID 位置 1。
- <4> 在 PC 中设置相应调试陷阱的处理地址(00000060H)，并且转移控制权。

图 15-13 显示了调试陷阱的处理格式。

图 15-13. 调试陷阱的处理格式



(2) 恢复

通过执行 DBRET 指令可以从调试陷阱中恢复。

执行 DBRET 指令，CPU 将执行以下步骤，并将控制权转移到恢复 PC 的地址。

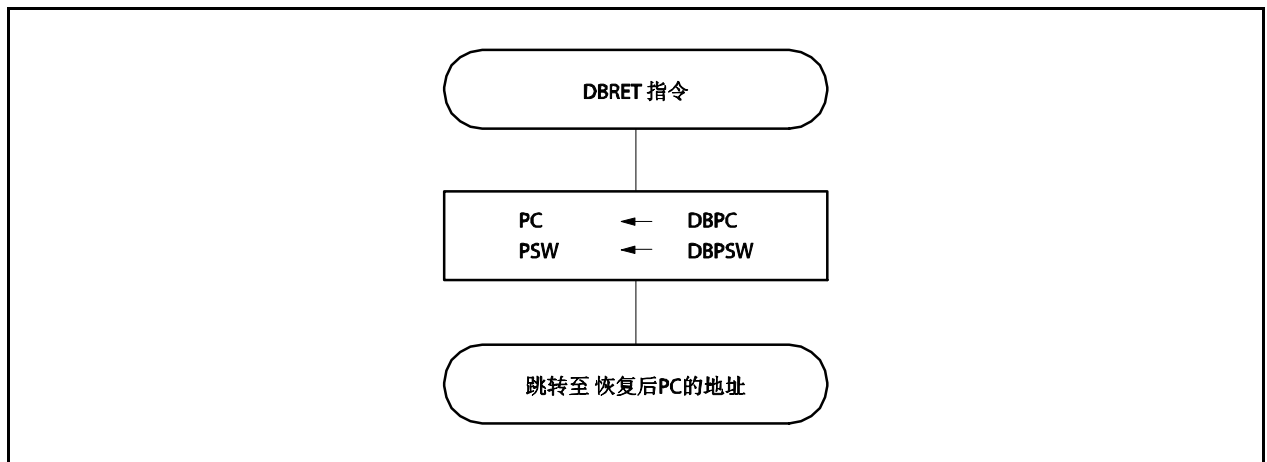
<1> 从 DBPC 和 DBPSW 中加载恢复的 PC 和 PSW。

<2> 向恢复的 PC 和 PSW 代表的地址转移控制权。

注意事项 仅在执行 DBTRAP 指令和 DBRET 指令的间隔过程中，可以访问 DBPC 和 DBPSW。

图 15-14 显示了从调试陷阱中恢复的处理格式。

图 15-14. 从调试陷阱中恢复的处理格式



15.6 外部中断请求输入引脚(NMI 和 INTP0 ~ INTP10)

15.6.1 噪声消除

(1) NMI 引脚的噪声消除

NMI 引脚拥有内部噪声消除电路，它使用模拟延迟。因此，NMI 引脚的输入电平不会作为边沿被检测，除非保持一定的时间或者更长。所以只有在特定的时间才会检测边沿。

NMI 引脚可以用来释放 STOP 模式。在 STOP 模式中，使用系统时钟不会消除噪声，因为内部系统时钟是停止的。

(2) INTP0~INTP7 引脚的噪声消除

INTP0~INTP10 引脚拥有内部噪声消除电路，它使用模拟延迟。因此，这些引脚的输入电平不会作为边沿被检测，除非保持一定的时间或者更长。所以只有在特定的时间才会检测边沿。

15.6.2 边沿检测

每一个 NMI 和 INTP0~INTP7 引脚的有效沿可以在以下四种类型中选择。

- 上升沿
- 下降沿
- 上升沿和下降沿
- 无边沿检测

在复位后，NMI 引脚不会检测边沿。因此，中断请求信号不会被确认，除非使用 INTF0 和 INTR0 寄存器允许有效沿（NMI 引脚功能作为一个端口引脚）。

(1) 外部中断下降，上升沿指定寄存器 0 (INTF0, INTR0)

INTF0 和 INTR0 寄存器是 8 位寄存器，它们通过位 2 指定了 NMI 引脚的上升沿和下降沿的检测，通过位 3~位 6 指定了外部中断引脚(INTP0 ~ INTP3)的上升沿和下降沿的检测。

这些寄存器支持 8 位或 1 位读写方式。

这些寄存器复位后的值为 00H。

注意事项 当引脚的功能从外部中断功能（复用功能）转变到端口功能时，可以进行边沿检测。因此，首先要将 INTF0n 位和 INTR0n 位清除为 00，然后再设定端口模式。

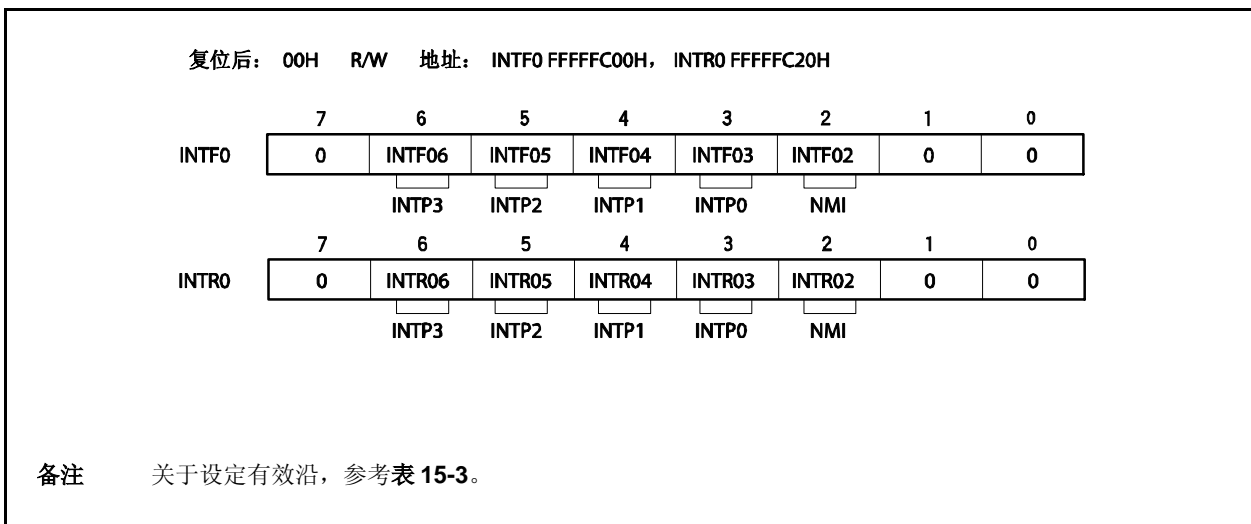


表 15-3. 有效沿指定

INTF0n	INTR0n	有效沿指定(n = 2 ~ 6)
0	0	无检测边沿
0	1	上升沿
1	0	下降沿
1	1	上升沿和下降沿

注意事项 当这些寄存器不被用作 NMI 或 INTP0~INTP3 时，一定要将 INTF0n 和 INTR0n 位清除为 00。

备注 n = 2: 控制 NMI 引脚
n = 3 ~ 6: 控制 INTP0 ~ INTP3 引脚

(2) 外部中断下降，上升沿指定寄存器 1 (INTR1, INTF1)

INTR1, INTF1 寄存器是 8 位寄存器，它指定了中断引脚 INTP9 和 INTP10 的上升沿和下降沿的检测。这些寄存器支持 8 位或 1 位读写方式。
 这些寄存器复位后的值为 00H。

注意事项 当引脚的功能从外部中断功能（复用功能）转变到端口功能时，可以进行边沿检测。因此，首先要将 INTF1n 位和 INTR1n 位清除为 00，然后再设定端口模式。

复位后: 00H R/W 地址: INTR1 FFFFC22H, INTF1 FFFFC02H

	7	6	5	4	3	2	1	0
INTR1	0	0	0	0	0	0	INTR11	INTR10
							INTP10	INTP9

	7	6	5	4	3	2	1	0
INTF1	0	0	0	0	0	0	INTF11	INTF10
							INTP10	INTP9

备注 关于如何设定有效沿，参考表 15-4。

表 15-4. 有效沿设定

INTF1n	INTR1n	有效沿设定(n = 0, 1)
0	0	无检测边沿
0	1	上升沿
1	0	下降沿
1	1	上升沿和下降沿

注意事项 当这些寄存器不被用作 INTP9 和 INTP10 时，一定要将 INTF1n 和 INTR1n 位清除为 00。

备注 n = 0: 控制 INTP9 引脚
 n = 1: 控制 INTP10 引脚

(3) 外部中断下降，上升沿指定寄存器 3 (INTR3, INTF3)

INTR3 和 INTF3 寄存器是 8 位寄存器，它们指定了外部中断引脚 INTP7 和 INTP8 的上升沿和下降沿的检测。这些寄存器支持 16 位读写。

但是，当 INTF3 寄存器用于高 8 位如 INTF3H 寄存器和低 8 位如 INTF3L 寄存器，这些寄存器支持 8 位或 1 位读写方式。

这些寄存器复位后的值为 00H。

注意事项 当引脚的功能从外部中断功能（复用功能）转变到端口功能时，可以进行边沿检测。因此，首先要将 INTF3n 位和 INTR3n 位清除为 00，然后再设定端口模式。

复位后: 0000H	R/W	地址:	INTF3 FFFFFFFC06H, INTF3L FFFFFFFC06H, INTF3H FFFFFFFC07H					
INTF3 (INTF3H ⁹)	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	INTF39	0
							└──┬──┘	
							INTP8	
(INTF3L)	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	INTF31	0
							└──┬──┘	
							INTP7	
复位后: 0000H	R/W	地址:	INTR3 FFFFFFFC26H, INTR3L FFFFFFFC26H, INTR3H FFFFFFFC27H					
INTR3 (INTR3H ⁹)	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	INTR39	0
							└──┬──┘	
							INTP8	
(INTR3L)	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	INTR31	0
							└──┬──┘	
							INTP7	

注意事项 当 INTF3 和 INTR3 寄存器的位 8~15 支持 8 位或 1 位读写方式时，指定它们为 INTF3H 和 INTR3H 寄存器为位 0~7。

备注 关于如何指定有效沿，参考表 15-5。

表 15-5. 有效沿指定

INTF3n	INTR3n	有效沿指定 (n = 1, 9)
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升沿和下降沿

注意事项 当这些寄存器不被用作 INTP7 和 INTP8 引脚时，一定要将 INTF3n 和 INTR3n 位清除为 00。

备注 n = 1: 控制 INTP7 引脚
n = 9: 控制 INTP8 引脚

(4) 外部中断下降，上升沿指定寄存器 9H (INTF9H, INTR9H)

INTF9H 和 INTR9H 寄存器是指定外部中断引脚(INTP4 ~ INTP6)的上升沿和下降沿检测情况的 8 位寄存器。

这些寄存器支持 8 位或 1 位读写方式。

这些寄存器复位后的值为 00H。

注意事项 当引脚的功能从外部中断功能（复用功能）转变到端口功能时，可以进行边沿检测。因此，首先要将 INTF9n 位和 INTR9n 位清零，然后再设定端口模式。

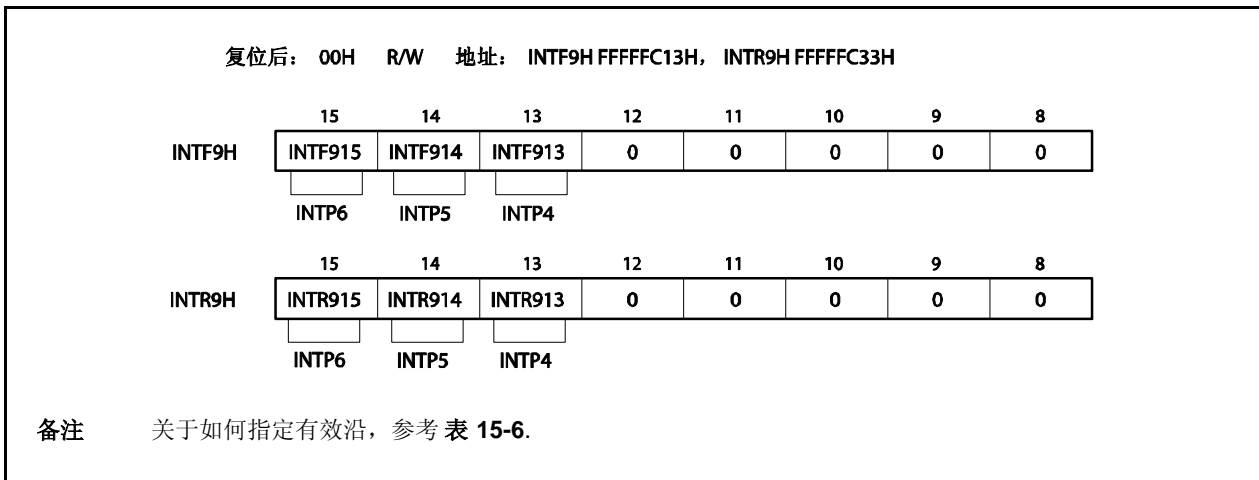


表 15-6. 有效沿指定

INTF9n	INTR9n	有效沿指定 (n = 13 ~ 15)
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升沿和下降沿

注意事项 当这些寄存器不被用作 INTP4~INTP6 时，一定要将 INTF9n 和 INTR9n 位清除为 00。

备注 n = 13 ~ 15: 控制 INTP4 ~ INTP6 引脚

(5) 噪声消除控制寄存器(NFC)

可以为 INTP3 引脚选择数字噪声消除。噪声消除设定通过使用 NFC 寄存器执行。

当选择了数字噪声消除，数字采样的时钟可以从 $f_{xx}/64$ ， $f_{xx}/128$ ， $f_{xx}/256$ ， $f_{xx}/512$ ， $f_{xx}/1,024$ 和 f_{XT} 中选择。采样频率使用 NFC.NFSTS 位。

当数字噪声消除被选中，如果在待机状态下的采样时钟停止了，那么 INTP3 的中断请求信号不能用来释放待机模式。当 f_{XT} 被用作采样时钟，INTP3 中断请求信号用来释放副时钟操作模式 或者是 IDLE1/IDLE2/STOP/副 IDLE 模式。

该寄存器支持 8 位读写方式。

该寄存器复位后的值为 00H。

注意事项 在改变采样时钟之后，需要用采样时钟数 \times NFSTS 设定时间的次数的个数个采样时钟来初始化数字噪声消除器。如果在采样时钟改变后的采样时钟数 \times NFSTS 设定时间的次数个时钟之后 INTP3 输入有效沿，将产生中断请求。因此，当使用中断和 DMA 功能的时候，一定要注意以下几点。

- 当使用中断功能时，在经过采样时钟数 \times NFSTS 设定时间的次数的个数个采样时钟之后，在中断请求标志(PIC3.PIF3 位)被清除后，要允许中断。
- 当使用 DMA 功能时（由 INTP3 启动），在经过采样时钟数 \times NFSTS 设定时间的次数的个数个采样时钟之后允许 DMA。

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	INTP3 引脚噪声清除设置
0	模拟噪声清除
1	数字噪声清除

NFSTS	数字噪声清除采样次数的设置
0	采样次数`3次
1	采样次数`2次

NFC2	NFC1	NFC0	数字采样时钟
0	0	0	fxo/64
0	0	1	fxo/128
0	1	0	fxo/256
0	1	1	fxo/512
1	0	0	fxo/1,024
1	0	1	fxi (副时钟)
以上除外			禁止设置

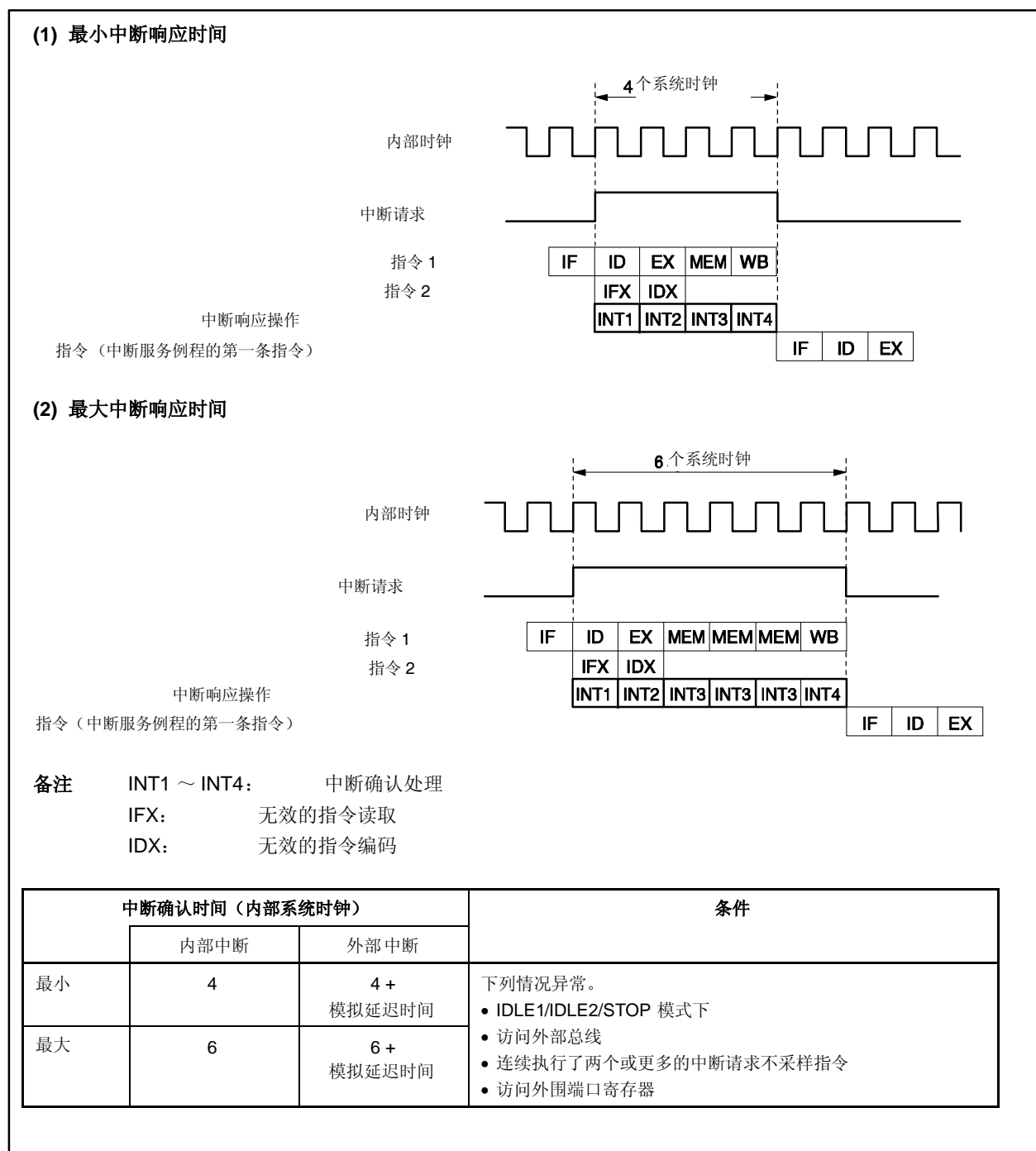
- 备注**
1. 当采样执行 3 次之后, 可靠的噪声消除宽度是 2 个采样时钟。
 2. 在噪声的宽度小于 2 个采样时钟的情况下, 如果有与采样时钟同步的噪声输入将会产生中断请求信号。

15.7 CPU 的中断确认时间

除非出现下列情况，CPU 的中断确认时间最少需要 4 个时钟周期。为了成功的输入中断请求信号，中断输入的间隔最少应该间隔 5 个时钟。

- 在 IDLE1/IDLE2/STOP 模式下
- 当访问外部总线时
- 当成功地执行了中断请求不采样指令（参考 15.8 CPU 不对中断进行确认的时期）
- 当访问中断控制寄存器时

图 15-15. 在中断请求信号确认过程中的传递操作（略图）



15.8 CPU 不对中断进行响应的时期

当执行指令时，中断会通过 CPU 确认。但是，在两个中断请求不采样指令之间（中断被保持）。中断不采样信号指令如下。

- EI 指令
- DI 指令
- PSW 的 LDSR reg2, 0x5 指令
- PRCMD 寄存器的存储指令
- 以下寄存器的存储，SET1, NOT1, 或是 CLR1 指令。
 - 有关中断的寄存器：
 - 中断控制寄存器 (xxICn)，中断屏蔽寄存器 0 ~ 3 (IMR0 ~ IMR3)
 - 节电控制寄存器 (PSC)
 - 片上调试模式寄存器 (OCDM)
 - 外围仿真寄存器 1 (PEMU1):

备注 xx: 表示任意的外围单元的名称（参考表 15-2 中断控制寄存器 (xxICn)）
 n: 外围单元编号（参考表 15-2 中断控制寄存器 (xxICn)）

15.9 注意事项

NMI 引脚和 P02 引脚是功能复用引脚，复位后的功能是一个正常的端口。为了执行 NMI 的功能，在 PMC0 寄存器中确认 NMI 引脚。NMI 引脚的初始设定是不检测边沿。使用 INTF0 寄存器和 INTR0 寄存器设定 NMI 引脚的有效沿。

第十六章 按键中断功能

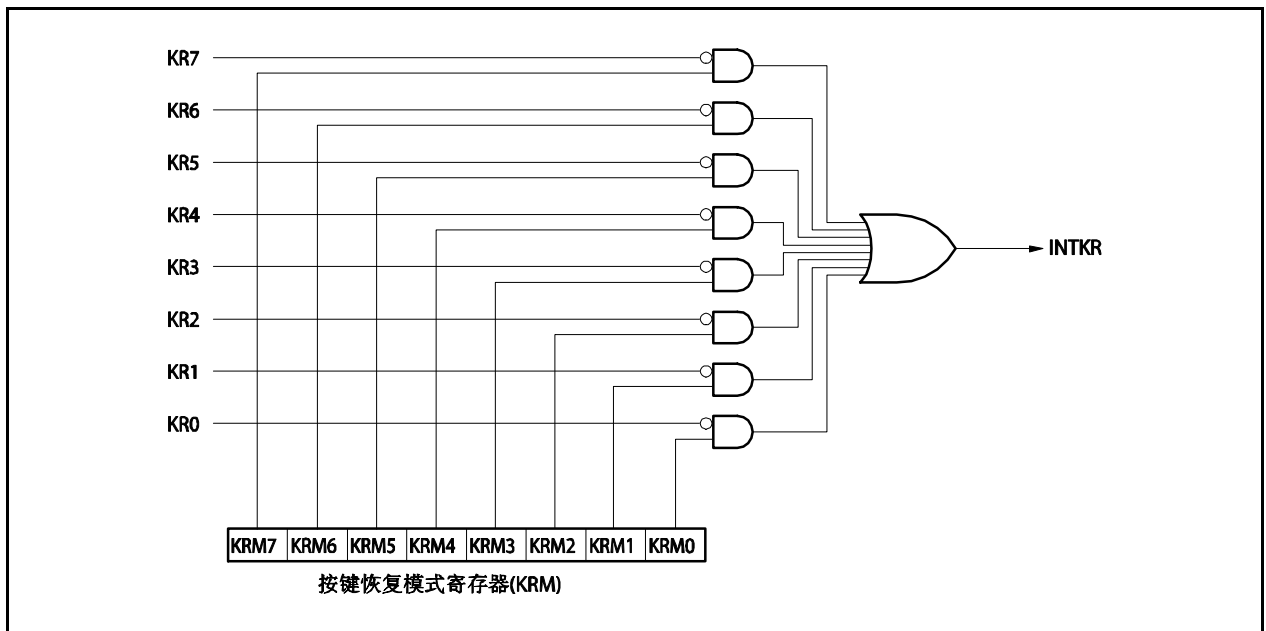
16.1 功能

通过设置 KRM 寄存器，给八个按键输入引脚(KR0 ~ KR7) 输入下降沿，可以产生按键中断请求信号(INTKR)。

表 16-1. 按键返回检测引脚的分配

标志	引脚描述
KRM0	位选控制 KR0 信号
KRM1	位选控制 KR1 信号
KRM2	位选控制 KR2 信号
KRM3	位选控制 KR3 信号
KRM4	位选控制 KR4 信号
KRM5	位选控制 KR5 信号
KRM6	位选控制 KR6 信号
KRM7	位选控制 KR7 信号

图 16-1. 按键返回框图



16.2 寄存器

(1) 按键返回模式寄存器 (KRM)

KRM 寄存器使用 KR0 ~ KR7 信号控制 KRM0 ~ KRM7 位。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF300H								
KRM	7	6	5	4	3	2	1	0
	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0
	KRMn	按键恢复模式控制						
	0	没有检测到按键恢复信号						
	1	检测到按键恢复信号						

注意事项 在 KRM 寄存器被清为 00H 后, 改写 KRM 寄存器。

备注 关于复用功能引脚设置, 敬请参阅表 4-15 使用端口引脚作为复用功能引脚。

16.3 注意事项

- (1) 如果低电平被输入给 KR0 ~ KR7 中的任何一个, 即使其它引脚的下降沿被输入, 也不会产生 INTKR 信号。
- (2) 不能同时使用 RXDA1 和 KR7 引脚。为了使用 RXDA1 引脚, 就不要使用 KR7 引脚。为了使用 KR7 引脚, 就不要使用 RXDA1 引脚 (推荐设置 PFC91 位为 1 并且将 PFCE91 位清零)。
- (3) 如果改变 KRM 寄存器, 则会产生中断请求信号 (INTKR)。为避免这种情况, 在禁止中断 (DI) 或屏蔽中断之后改变 KRM 寄存器, 然后将中断请求标志 (KRIC、KRIF 位) 清零, 再允许中断或清除屏蔽。
- (4) 为了使用按键中断功能, 确信设置端口引脚为按键返回引脚, 然后使用 KRM 寄存器允许操作。为了从按键返回引脚切换到端口引脚, 使用 KRM 寄存器禁止操作, 然后设置为端口引脚。

第十七章 待机功能

17.1 概述

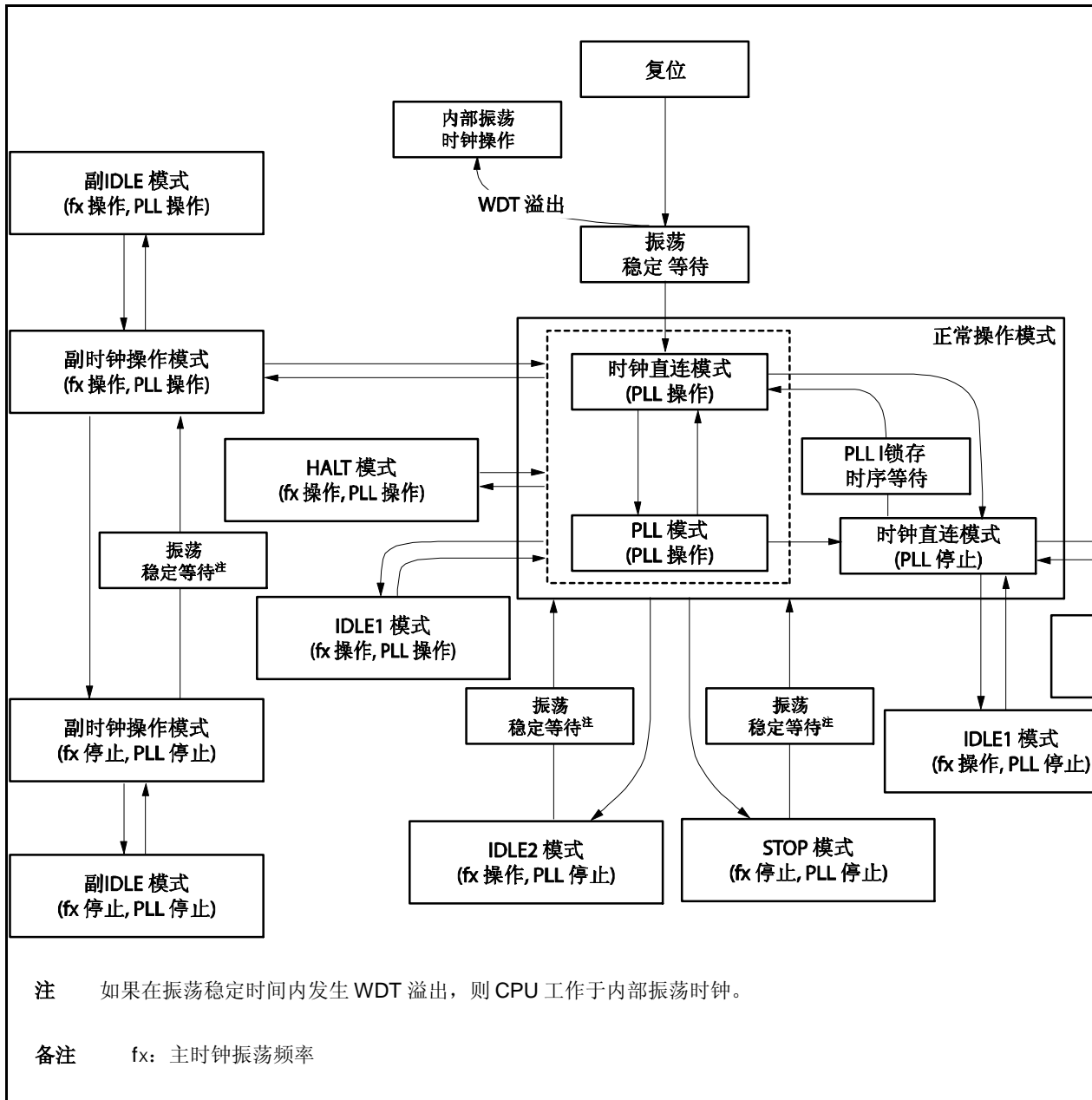
通过结合使用待机模式并选择合适的应用模式，可以有效的降低系统功耗。可用到的待机模式在表 17-1 中列出。

表 17-1. 待机模式

模式	功能概要
HALT 模式	此模式仅停止 CPU 的操作时钟
IDLE1 模式	此模式停止除振荡器之外的全部内部电路，PLL ^注 ，和 flash 存储器的操作
IDLE2 模式	此模式停止除了振荡器之外的所有片内操作
STOP 模式	此模式停止除了副时钟振荡器之外的所有片内操作
副时钟操作模式	此模式副时钟被用作内部系统时钟
Sub-IDLE 模式	此模式停止除了振荡器之外的全部片内操作，工作在副时钟操作模式下

注 PLL 保持预先操作状态。

图 17-1. 状态转换



17.2 寄存器

(1) 功能节控制寄存器(PSC)

PSC 寄存器是控制待机功能的 8 位寄存器。此寄存器的 STP 位被用于指定 STOP 模式。此寄存器是仅能使用特殊顺序组合写入的特殊寄存器（参见 3.4.7 特殊寄存器）。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后：00H R/W 地址：FFFF1FEH

	7	6	5	4	3	2	1	0
PSC	0	NMI1M	NMI0M	INTM	0	0	STP	0

NMI1M	发生INTWDT2 信号时的待机模式释放控制
0	启用 INTWDT2 信号释放待机模式
1	禁用 INTWDT2 信号释放待机模式

NMI0M	通过NMI 引脚输入释放待机模式
0	启用NMI 引脚输入释放待机模式
1	禁用NMI 引脚输入释放待机模式

INTM	通过可屏蔽中断请求信号控制释放待机模式
0	启用可屏蔽中断请求信号释放待机模式
1	禁用可屏蔽中断请求信号释放待机模式

STP	待机模式 ^注 设置
0	正常模式
1	待机模式

注 通过 STP 位设置待机模式：IDLE1、IDLE2、STOP 或副 IDLE 模式。

- 注意事项
1. 在设置 IDLE1，IDLE2，STOP 或副 IDLE 模式之前，先设置 PSMR.PSM1 和 PSMR.PSM0 位，然后设置 STP 位。
 2. 当释放 HALT 模式时对 NMI1M，NMI0M 和 INTM 位进行设置是非法的。
 3. 如果在 NMI1M，NMI0M，或 INTM 位设置为 1 的同时，STP 位也设置为 1，则 NMI1M，NMI0M，或 INTM 位设置无效。当设置 IDLE1/IDLE2/STOP 模式时，如果有一特定未屏蔽中断请求信号，设置对应中断请求信号(NMI1M，NMI0M，或 INTM)的位为 1，然后设置 STP 位为 1。

<R>

(2) 功率节省模式寄存器 (PSMR)

PSMR 寄存器是一个 8 位寄存器，其控制省电模式中的操作状态和时钟操作。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H R/W 地址: FFFF820H

	7	6	5	4	3	2	1	0
PSMR	0	0	0	0	0	0	PSM1	PSM0

PSM1	PSM0	软件待机模式中操作规格
0	0	IDLE1
0	1	STOP 模式
1	0	IDLE2, sub-IDLE 模式
1	1	STOP 模式

- 注意事项**
1. 确定将位 2~位 7 清零。
 2. 仅当 PSC.STP 位为 1 时，PSM0 和 PSM1 位是有效的。

- 备注**
- IDLE1:** 在此模式中，除了振荡器之外的所有操作和一些其它的电路（flash 存储器和 PLL）都被停止。
在 IDLE1 模式被释放后，重新恢复为正常操作模式而无须确保振荡稳定时间，就像 HALT 模式一样。
- IDLE2:** 在此模式中，除了振荡器操作之外的所有操作都被停止。
在 IDLE2 模式被释放后，等待通过 OSTS 寄存器指定的设置时间后恢复为正常操作模式（flash 存储器和 PLL）。
- STOP:** 在此模式中，除了副时钟振荡器操作之外的所有操作都被停止。
在 STOP 模式被释放后，等待通过 OSTS 寄存器指定的振荡稳定时间后恢复为正常操作模式。
- Sub-IDLE:** 在此模式中，除了振荡器操作之外的所有其它操作都被暂停。在通过中断请求信号释放 IDLE 模式之后，在等待 12 个副时钟周期之后重新恢复副时钟操作模式。

(3) 振荡稳定时间选择寄存器(OSTS)

通过 OSTS 寄存器控制 STOP 模式释放后到振荡稳定的等待时间或 IDLE2 模式释放后到内置 flash 存储器稳定的等待时间。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后值为 06H。

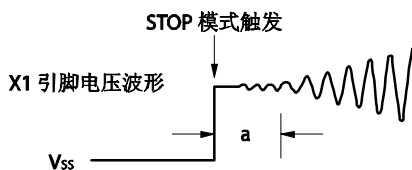
复位后：06H R/W 地址：FFFF6C0H

	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/设置时间 ^a 选择	fx	
				4 MHz	5 MHz
				0	0
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 当 STOP 模式和 IDLE2 模式被释放后分别要求振荡稳定时间和设置时间。

注意事项 1. STOP 模式释放后的等待时间不包括从释放 STOP 模式到时钟振荡启动这段时间（即下图“a”所示的部分），不管通过复位或中断请求信号的产生是否释放了 STOP 模式。



2. 确定将位 3~位 7 清零。

3. 复位释放后的振荡稳定时间是 $2^{16}/f_x$ （因为 OSTS 寄存器的初始值=06H）。

备注 f_x = 主时钟振荡频率

17.3 HALT 模式

17.3.1 设置和操作状态

当在正常操作模式中执行专用指令（HALT）时，HALT 模式被设置。

在 HALT 模式中，时钟振荡器继续操作。仅停止对 CPU 的时钟供应，继续提供给其它内置外围设备功能的时钟供应。

结果，程序停止执行，并且内部 RAM 保持 HALT 模式设置之前的内容。独立于 CPU 指令处理的内置外围设备功能可继续操作。

表 17-3 显示了 HALT 模式中的操作状态。

通过使用 HALT 模式并结合正常操作模式的间歇操作可降低系统的平均电流消耗。

注意事项 1. 在 HALT 指令之后插入五个或更多的 NOP 指令。

2. 如果在执行 HALT 指令时有一个未屏蔽的中断请求被保持等待响应，则此状态被转移到 HALT 模式中，然后通过这个等待响应的中断请求，HALT 模式被立即释放。

17.3.2 释放 HALT 模式

可通过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 ~ INTP7 引脚输入），来自于在 HALT 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，WDT2RES 信号，上电清零电路 (POC)，低电压检测器 (LVI) 或时钟监视器 (CLM) 产生的复位）来释放 HALT 模式。

在 HALT 模式释放后，恢复正常操作模式。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 HALT 模式

通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT1，INTWDT2 信号），未屏蔽的可屏蔽中断请求信号，及复位信号（RESET 引脚输入，WDTRES1，WDTRES2 信号）释放 HALT 模式。在 HALT 模式释放后，恢复正常操作模式。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 HALT 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 HALT 模式，并且响应此中断请求信号。

表 17-2. 通过中断请求信号释放 HALT 模式后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	执行对处理地址的跳转	
可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令	执行下一条指令

(2) 通过复位释放 HALT 模式

执行与正常复位操作相同的操作。

表 17-3. HALT 模式的操作状态

HALT 模式设置		操作状态	
		当不使用副时钟时	使用副时钟时
主时钟振荡器		使能振荡	
副时钟振荡器		-	使能振荡
内部振荡器		使能振荡	
PLL		可操作	
CPU		停止操作	
DMA		可操作	
中断控制器		可操作	
定时器 P (TMP0 ~ TMP3)		可操作	
定时器 Q (TMQ0, TMQ1)		可操作	
定时器 M (TMM0)		当非 f_{XT} 时钟被选择作为计数时钟时可操作	可操作
钟表定时器		当 f_{XT} (BRG 分频) 时钟被选择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 ~ UARTA2	可操作	
A/D 转换器		可操作	
按键中断功能 (KR)		可操作	
端口功能		在 HALT 模式被设置之前保持状态	
内部数据		CPU 寄存器, 状态, 数据和全部其它的内部数据, 例如内部 RAM 的内容, 被保持为 HALT 模式设置之前的状态。	

17.4 IDLE1 模式

17.4.1 设置和操作状态

通过在正常操作模式中对 PSMR.PSM1 和 PSMR.PSM0 位清零，并设置 PSC.STP 位为 1 设置 IDLE1 模式。

在 IDLE1 模式中，时钟振荡器，PLL 和 flash 存储器继续操作但停止给 CPU 和其它内置外围设备功能的时钟供应。

结果，停止程序执行，并且保持在设置 IDLE1 模式之前的内部 RAM 的内容。CPU 和其它内置外围设备功能停止操作。但是，使用副时钟或外部时钟的内置外围设备功能继续操作。

表 17-5 显示了在 IDLE1 模式中的操作状态。

因为 IDLE1 模式停止了内置外围设备功能的操作，所以 IDLE1 模式比 HALT 模式更能降低功耗。因为不停止主时钟振荡器，所以当用与释放 HALT 模式相同的方法释放 IDLE1 模式时，无须等待振荡稳定时间就能恢复到正常操作模式。

- 注意事项 1.** 在执行往 PSC 寄存器中存入数据来设置 IDLE1 模式的指令之后，插入五个或更多的 NOP 指令。
- 2.** 如果设置 IDLE1 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则 IDLE1 模式立即被此等待响应的中断请求释放。

17.4.2 释放 IDLE1 模式

可通过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 ~ INTP10 引脚输入），来自于在 IDLE1 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，WDT2RES 信号，上电清零电路 (POC)，低电压检测器 (LVI) 或时钟监视器 (CLM) 产生的复位）来释放 IDLE1 模式。

在 IDLE1 模式释放后，恢复正常操作模式。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 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释放 IDLE1 模式，而不管中断请求信号的优先级。

如果在某个中断服务程序中设置 IDLE1 模式，则之后产生的中断请求信号按如下所述来执行。

- 注意事项 1.** 通过设置 PSC.NMI1M，PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号是无效的，IDLE1 模式也不会被释放。
- 2.** 如果通过使用 NFC 寄存器选择数字噪声清除，采样时钟选自 fxx/64，fxx/128，fxx/256，fxx/512，和 fxx/1024，IDLE1 模式不能通过 INTP3 引脚中断请求信号释放。关于更多细节，敬请参阅 15.6.2 (5) 噪声清除控制寄存器 (NFC)。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 IDLE1 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 IDLE1 模式，并且响应此中断请求信号。

表 17-4. 通过中断请求信号释放 IDLE1 模式后的操作

释放源	中断使能 (EI) 状态	中断禁用 (DI) 状态
不可屏蔽中断请求信号	执行对处理地址的跳转	
可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令	执行下一条指令

(2) 通过复位释放 IDLE1 模式

执行与正常复位操作相同的操作。

表 17-5. IDLE1 模式中的操作状态

项目	IDLE1 模式	操作状态	
		当不使用副时钟时	当使用副时钟时
主时钟振荡器		使能振荡	
副时钟振荡器		-	使能振荡
内部振荡器		使能振荡	
PLL		可操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作 (但是可操作释放待机模式)	
定时器 P (TMP0 ~ TMP3)		停止操作	
定时器 Q (TMQ0, TMQ1)		停止操作	
定时器 M (TMM0)		当选择 $f_{R/8}$ 作为计数时钟时可操作	当选择 $f_{R/8}$ 或 f_{τ} 作为计数时钟时可操作
钟表定时器		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 \overline{SCKBn} 输入时钟作为计数时钟时可操作 ($n = 0, 1$)	
	UARTA0~UARTA2	停止操作 (但是当选择 ASCKA0 输入时钟是可操作的)	
A/D 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 IDLE1 模式之前的状态	
内部数据		CPU 寄存器, 状态, 数据和全部其它的内部数据, 例如内部 RAM 的内容, 被保持为 IDLE1 模式设置之前的状态。	

注 为实现低功耗, 在设置 IDLE1 模式之前停止 A/D 转换器。

17.5 IDLE2 模式

17.5.1 设置和操作状态

通过在正常操作模式中设置 PSMR.PSM1 和 PSMR.PSM0 位为 10，并设置 PSC.STP 位为 1 设置 IDLE2 模式。

在 IDLE2 模式中，时钟振荡器继续操作但停止给 CPU，PLL，flash 存储器和其它内置外围设备功能的时钟供应。

结果，停止程序执行，并且保持在设置 IDLE2 模式之前的内部 RAM 的内容。CPU，PLL 和其它内置外围设备功能停止操作。但是，使用副时钟或外部时钟的内置外围设备功能继续操作。

表 17-7 显示了在 IDLE2 模式中的操作状态。

因为 IDLE2 模式停止了内置外围设备功能，PLL 和 flash 存储器的操作，所以 IDLE2 模式比 IDLE1 模式更能降低功耗。然而，因为 PLL 和 flash 存储器被停止，所以当释放 IDLE2 模式时，对 PLL 和 flash 存储器要求有一段设置时间。

- 注意事项**
1. 在执行往 PSC 寄存器中存入数据来设置 IDLE2 模式的指令之后，插入五个或更多的 NOP 指令。
 2. 如果设置 IDLE2 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则 IDLE2 模式立即被此等待响应的中断请求释放。

17.5.2 释放 IDLE2 模式

可通过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 ~ INTP10 引脚输入），来自于在 IDLE2 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，WDT2RES 信号，上电 清零电路 (POC)，低电压检测器（LVI）或时钟监视器（CLM）产生的复位）来释放 IDLE2 模式。在设置 IDLE2 模式之前，PLL 返回到操作状态。

在 IDLE2 模式释放后，恢复正常操作模式。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 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释放 IDLE2 模式，而不管中断请求信号的优先级。如果在某个中断服务程序中设置 IDLE2 模式，则之后产生的中断请求信号按如下所述来执行。

- 注意事项**
1. 通过设置 PSC.NMI1M，PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号是无效的，IDLE2 模式也不会被释放。
 2. 如果通过使用 NFC 寄存器选择数字噪声清除，采样时钟选自 fxx/64，fxx/128，fxx/256，fxx/512，和 fxx/1024，IDLE1 模式不能通过 INTP3 引脚中断请求信号释放。关于更多细节，敬请参阅 15.6.2 (5) 噪声清除控制寄存器（NFC）。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 IDLE2 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 IDLE2 模式，并且响应此中断请求信号。

表 17-6. 通过中断请求信号释放 IDLE2 模式后的操作

释放源	中断使能 (EI) 状态	中断禁用 (DI) 状态
不可屏蔽中断请求信号	在保证规定的设置时间之后执行对处理地址的跳转。	
可屏蔽中断请求信号	在保证规定的设置时间之后执行对处理地址的跳转或执行下一条指令。	在保证规定的设置时间之后执行下一条指令。

(2) 通过复位释放 IDLE2 模式

执行与正常复位操作相同的操作。

表 17-7. IDLE2 模式中的操作状态

项目	IDLE2 模式设置	操作状态	
		当不使用副时钟时	当使用副时钟时
主时钟振荡器		使能振荡	
副时钟振荡器		—	使能振荡
内部振荡器		使能振荡	
PLL		停止操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作 (但是可操作释放待机模式)	
定时器 P (TMP0~TMP3)		停止操作	
定时器 Q (TMQ0, TMQ1)		停止操作	
定时器 M (TMM0)		当选择 $f_{R/8}$ 作为计数时钟时可操作	当选择 $f_{R/8}$ 或 f_{τ} 作为计数时钟时可操作
钟表定时器		当选择 f_x (BRG 分频) 作为计数时钟时可选择	可操作
看门狗定时器 2		可操作	
串行接口	CSIB0, CSIB1	当选择 $\overline{\text{SCKBn}}$ 输入时钟作为计数时钟时可操作 ($n = 0, 1$)	
	UARTA0 ~UARTA2	停止操作 (但是当选择 ASCKA0 输入时钟时 UARTA0 是可操作的)	
A/D 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 IDLE2 模式之前的状态	
内部数据		CPU 寄存器, 状态, 数据和全部其它的内部数据, 例如内部 RAM 的内容, 被保持为 IDLE2 模式设置之前的状态。	

注 为实现低功耗, 在设置 IDLE2 模式之前停止 A/D 转换器。

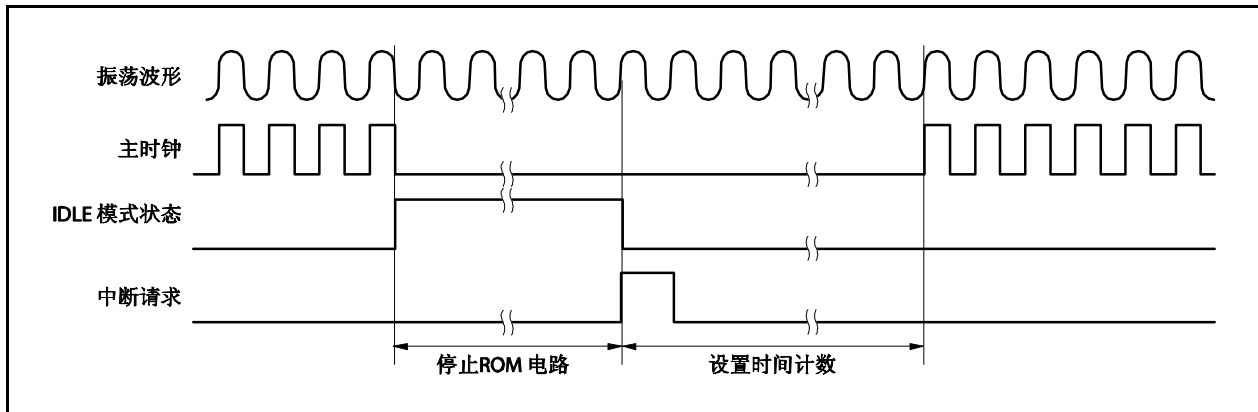
17.5.3 当释放 IDLE2 模式时保证设置时间

在释放 IDLE2 模式之后需保证 ROM (flash 存储器) 的设置时间，这是因为在设置 IDLE2 模式之后，除了主时钟振荡器外，块的操作也被停止了。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE2 模式

通过设置 OSTS 寄存器保证指定的设置时间。

当产生释放源时，根据 OSTS 寄存器的设置，专用的内部定时器开始计数。当它溢出时，正常操作模式被恢复。



(2) 通过复位 ($\overline{\text{RESET}}$ 引脚输入, WDT2RES 产生) 释放

此操作与正常复位操作相同。

振荡稳定时间是 OSTS 寄存器的初始值, $2^{16}/f_x$ 。

17.6 STOP 模式

17.6.1 设置和操作状态

通过在正常操作模式中设置 PSMR.PSM1 和 PSMR.PSM0 位为 01 或 11，并设置 PSC.STP 位为 1 设置 STOP 模式。

在 STOP 模式中，副时钟振荡器继续操作，主时钟振荡器停止。停止给 CPU 和内置外围设备功能的时钟供应。

结果，停止程序执行，并且保持在设置 STOP 模式之前的内部 RAM 的内容。使用副时钟振荡器或外部时钟的内置外围设备功能继续操作。

表 17-9 显示了在 STOP 模式中的操作状态。

因为 STOP 模式停止了主时钟振荡器的操作，所以 STOP 模式比 IDLE2 模式更能降低功耗。如果不使用副时钟振荡器，内部振荡器和外部时钟，则功耗最低仅为漏电流消耗的功耗。

- 注意事项**
1. 在执行往 PSC 寄存器中存入数据来设置 STOP 模式的指令之后，插入五个或更多的 NOP 指令。
 2. 如果设置 STOP 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则 STOP 模式立即被此等待响应的中断请求释放。

17.6.2 释放 STOP 模式

可通过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 ~ INTP10 引脚输入），来自于在 STOP 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（ $\overline{\text{RESET}}$ 引脚输入，WDT2RES 信号，上电清零电路 (POC) 或低电压检测器 (LVI) 产生的复位）来释放 STOP 模式。

在 STOP 模式释放后，在振荡稳定时间被保证之后，恢复正常操作模式。

- 注意事项**
1. 通过设置 PSC.NMI1M，PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号是无效的，STOP 模式也不会被释放。
 2. 如果通过使用 NFC 寄存器选择数字噪声清除，采样时钟选自 fxx/64，fxx/128，fxx/256，fxx/512，和 fxx/1024，IDLE1 模式不能通过 INTP3 引脚中断请求信号释放。关于更多细节，敬请参阅 15.6.2 (5) 噪声清除控制寄存器 (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释放 STOP 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 STOP 模式，而不管中断请求信号的优先级。如果在某个中断服务程序中设置 STOP 模式，则之后产生的中断请求信号按如下所述来执行。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 STOP 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 STOP 模式，并且响应此中断请求信号。

表 17-8. 通过中断请求信号释放 STOP 模式后的操作

释放源	中断使能 (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之后执行对处理地址的跳转。	
可屏蔽中断请求信号	在保证振荡稳定时间之后执行对处理地址的跳转或执行下一条指令。	在保证振荡稳定时间之后执行下一条指令。

(2) 通过复位释放 STOP 模式

执行与正常复位操作相同的操作。

表 17-9. STOP 模式中的操作状态

项目	STOP 模式设置	操作状态	
		当不使用副时钟时	当使用副时钟时
主时钟振荡器		停止振荡	
副时钟振荡器		—	使能操作
内部振荡器		使能操作	
PLL		停止操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是可操作释放待机模式）	
定时器 P (TMP0~ TMP3)		停止操作	
定时器 Q (TMQ0, TMQ1)		停止操作	
定时器 M (TMM0)		当选择 $f_{R/8}$ 作为计数时钟时可操作	当选择 $f_{R/8}$ 或 f_{XT} 作为计数时钟时可操作
钟表定时器		停止操作	当选择 f_{XT} 作为计数时钟时可操作
看门狗定时器 2		当选择 f_{R} 作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 ~ UARTA2	停止操作（但是当选择 ASCKA0 输入时钟时, UARTA0 是可操作的）	
A/D 转换器		停止操作停止操作（转换结果不确定） ^{※1, 2}	
按键中断功能 (KR)		可操作	
端口功能		保持 STOP 模式设置之前的状态。	
内部数据		CPU 寄存器, 状态, 数据和全部其它的内部数据, 例如内部 RAM 的内容, 被保持为 STOP 模式设置之前的状态。	

- 注
1. 当操作 A/D 转换器时, 如果设置 STOP 模式, 则 A/D 转换器自动停止操作, 并且在 STOP 模式释放后再次启动操作。然而, 在这种情况下, 释放 STOP 模式后的 A/D 转换结果是无效的。所有设置 STOP 模式之前的 A/D 转换结果也是无效的。
 2. 即使在操作 A/D 转换器时设置 STOP 模式, 功耗降低也与在设置 STOP 模式之前停止 A/D 转换器时的一样。

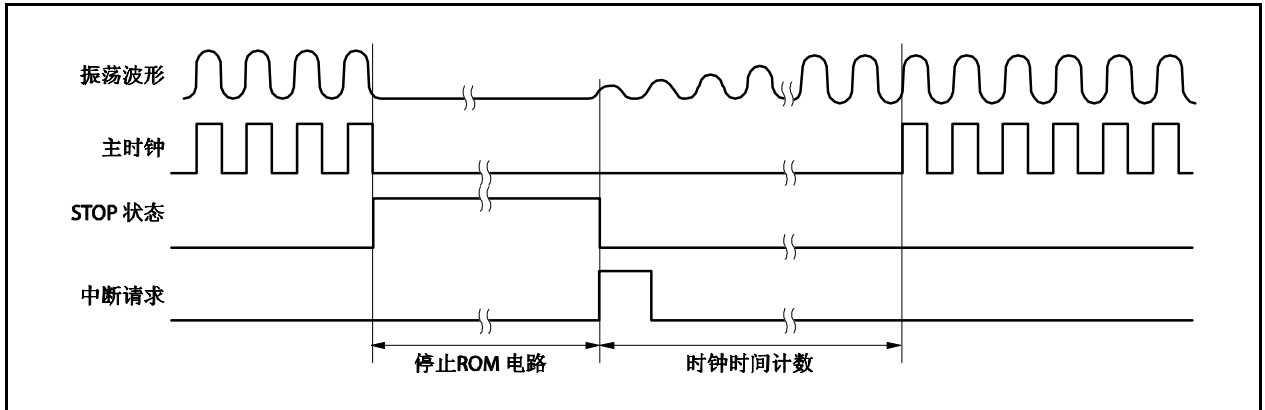
17.6.3 当释放 STOP 模式时保证振荡稳定时间

在释放 STOP 模式后需保证主时钟振荡器的振荡稳定时间，这是因为在设置 STOP 模式之后主时钟振荡器操作停止。

(1) 通过不可屏蔽中断请求信号或未屏蔽的中断请求信号释放 STOP 模式

通过设置 OSTS 寄存器保证振荡稳定时间。

当产生释放源时，根据 OSTS 寄存器的设置，专用内部定时器开始计数。当它溢出时，恢复为正常操作模式。



(2) 通过复位释放

此操作与正常复位操作相同。

振荡稳定时间是 OSTS 寄存器的初始值， $2^{16}/f_x$ 。

17.7 副时钟操作模式

17.7.1 设置和操作状态

通过在正常操作模式中设置 PCC.CK3 位为 1 来设置副时钟操作模式。

当副时钟操作模式被设置时，内部系统时钟从主时钟转换为副时钟。通过使用 PCC.CLS 位检测是否时钟已被切换。

当 PCC.MCK 位被设置为 1 时，主时钟振荡器的操作被停止。这样，系统仅以副时钟来操作。

在副时钟操作模式中，与正常操作模式相比，功耗降到一个较低的水平上，这是因为副时钟被用作内部系统时钟。

除此之外，通过停止主时钟振荡器的操作，功耗能更进一步降低到 STOP 模式的等级。

表 17-10 显示了副时钟操作模式的状态。

- 注意事项**
1. 当操作 CK3 位时，不要改变 PCC.CK2 ~ PCC.CK0 位（推荐使用位操作指令来操作）的设置值。对于 PCC 寄存器的细节描述，参见 5.3 (1) 处理器时钟控制寄存器(PCC)。
 2. 如果以下的条件不满足，改变 CK2 ~ CK0 位的设置以便满足条件并且设置副时钟操作模式。
内部系统时钟(f_{CLK}) > 副时钟 (f_{XT}) $\times 4$

备注 内部系统时钟 (f_{CLK}) : CK2 ~ CK0 位设置的同时，主时钟发生一内部时钟。

17.7.2 释放副时钟操作模式

当 CK3 位被清零时，可以通过复位信号（通过 RESET 引脚输入，WDT2RES 信号，上电清零电路 (POC)，低电压检测器 (LVI) 或时钟监视器 (CLM) 复位) 释放副时钟操作模式。

如果停止主时钟 (MCK 位 = 1)，设置 MCK 位为 0，通过软件保证主时钟的振荡稳定时间，并将 CK3 位清零。

当副时钟操作模式释放后，恢复为正常操作模式。

- 注意事项** 当操作 CK3 位时，不要改变 CK2 ~ CK0 位（推荐使用位操作指令来操作）的设置值。
关于 PCC 寄存器的更多细节，敬请参阅 5.3 (1) 处理器时钟控制寄存器(PCC)。

表 17-10. 副时钟操作模式中的操作状态

副时钟操作模式设置		操作状态	
		当主时钟振荡时	当主时钟停止时
副时钟振荡器		使能操作	
内部振荡器		使能操作	
PLL		可操作	停止操作 ^注
CPU		可操作	
DMA		可操作	
中断控制器		可操作	
定时器 P (TMP0~ TMP3)		可操作	停止操作
定时器 Q (TMQ0, TMQ1)		可操作	停止操作
定时器 M (TMM0)		可操作	当选择 $f_{R/8}$ 或 f_{XT} 作为计数时钟时可操作
钟表 定时器		可操作	当选择 f_{XT} 作为计数时钟时可操作
看门狗定时器 2		可操作	当选择 f_R 作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~ UARTA2	可操作	停止操作 (但是当选择 ASCKA0 输入时钟时 UARTA0 是可操作的)
A/D 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 停止主时钟之前确定停止 PLL (PLLCTL.PLLON 位 = 0)。

注意事项 当 CPU 工作在副时钟并且主时钟振荡停止时，不能访问发生等待的寄存器。如果产生等待，则其只能通过复位来释放 (见 3.4.8(2))。

17.8 Sub-IDLE 模式

17.8.1 设置和操作状态

通过在 sub-IDLE 操作模式中设置 PSMR.PSM1 和 PSMR.PSM0 位为 10，并设置 PSC.STP 位为 1 设置 sub-IDLE 模式。

在此模式中，时钟振荡器继续操作，停止给 CPU，flash 存储器和其它内置外围设备功能的时钟供应

结果，停止程序执行，并且保持设置 sub-IDLE 模式之前的内部 RAM 的内容。停止 CPU 和其它内置外围设备功能。

然而，可使用副时钟或外部时钟操作的内置外围设备功能继续操作。

因为 sub-IDLE 模式停止了 CPU，flash 存储器和其它外围设备功能的操作，所以它比副时钟操作模式更能降低功耗。如果在主时钟停止后设置 sub-IDLE 模式，当前功耗能降低至 STOP 模式的等级。

表 17-12 显示了 sub-IDLE 模式中的操作状态。

- 注意事项**
1. 在执行往 PSC 寄存器中存入数据来设置 sub-IDLE 模式的指令之后，插入五个或更多的 NOP 指令。
 2. 如果设置 sub-IDLE 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则副 IDLE 模式立即被此等待响应的中断请求释放。

17.8.2 释放 sub-IDLE 模式

可通过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 ~ INTP10 引脚输入），来自于在副 IDLE 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，WDT2RES 信号，上电清零电路 (POC)，低电压检测器 (LVI) 或时钟监视器 (CLM) 产生的复位) 来释放副 IDLE 模式。在设置副 IDLE 模式之前，PLL 返回到操作状态。

当通过中断请求信号释放 sub-IDLE 模式时，副时钟操作模式被设置。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 sub-IDLE 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 sub-IDLE 模式，而不管中断请求信号的优先级。

如果在某个中断服务程序中设置 sub-IDLE 模式，则之后产生的中断请求信号按如下所述来执行。

注意事项 1. 通过设置 PSC.NMI1M, PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号是无效的，sub-IDLE 模式也不会被释放。

2. 当释放 sub-IDLE 模式时，从产生释放副 IDLE 模式的中断请求信号开始到副 IDLE 模式被释放之间须等待 12 个副时钟周期（大约 366 μ s）。

3. 如果通过使用 NFC 寄存器选择数字噪声清除，采样时钟选自 fxx/64, fxx/128, fxx/256, fxx/512, 和 fxx/1024, IDLE1 模式不能通过 INTP3 引脚中断请求信号释放。更多细节，敬请参阅 15.6.2 (5) 噪声清除控制寄存器 (NFC)。

(a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 sub-IDLE 模式，但是不响应此中断请求信号。此中断请求信号被保持。

(b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 sub-IDLE 模式，并且响应此中断请求信号。

表 17-11. 通过中断请求信号释放 Sub-IDLE 模式之后的操作

释放源	中断使能 (EI) 状态	中断禁用 (DI) 状态
不可屏蔽中断请求信号	执行对处理地址的跳转。	
可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令。	执行下一条指令。

(2) 通过复位释放 sub-IDLE 模式

执行与正常复位操作相同的操作。

表 17-12. 在 Sub-IDLE 模式中的操作状态

Sub-IDLE 模式设置		操作状态	
		当主时钟振荡时	当主时钟停止时
副时钟振荡器		使能操作	
内部振荡器		使能操作	
PLL		可操作	停止操作 ^{注1}
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是可操作释放待机模式）	
定时器 P (TMP0 ~TMP3)		停止操作	
定时器 Q (TMQ0, TMQ1)		停止操作	
定时器 M (TMM0)		当选择 $f_R/8$ 或 f_{XT} 作为计数时钟时可操作	
钟表定时器		停止操作	选择 f_{XT} 作为计数时钟时可操作
看门狗定时器 2		当选择 f_R 作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 ~UARTA2	停止操作（但是当选择 ASCKA0 输入时钟时 UARTA0 是可操作的）	
A/D 转换器		保持操作（保持转换结果） ^{注2}	
按键中断功能(KR)		可操作	
端口功能		保持设置 sub-IDLE 模式之前的状态	
内部数据		CPU 寄存器, 状态, 数据和全部其它的内部数据, 例如内部 RAM 的内容, 被保持为 sub-IDLE 模式设置之前的状态。	

- 注**
1. 停止主时钟之前确定停止 PLL（PLLCTL.PLLON 位 = 0）。
 2. 为了实现低功耗, 在进入 sub-IDLE 模式之前停止 A/D 转换器。

第十八章 复位功能

18.1 概述

如下复位功能可用。

(1) 四种复位源

- 由 **RESET** 引脚输入的外部复位
- 由看门狗定时器 2 (WDT2) 溢出 (WDT2RES) 引起的复位
- 由于低电压检测器(LV1)比较供电电压和检测电压引起的系统复位
- 由于检测到时钟监控器(CLM)停止振荡引起的系统复位
- 通过上电清零电路来复位系统

当一个复位信号释放后，复位源通过复位源标志寄存器(RESF)来确认。

(2) 紧急操作模式

复位后在主时钟振荡稳定时间内如果 WDT2 溢出，判断为一个不规则的主时钟振荡并且 CPU 在内部振荡时钟下开始操作。

注意事项 当 CPU 在内部振荡时钟下操作，有权使用在等待状态下禁止使用的寄存器。关于等待状态下禁止使用的寄存器，参见 3.4.8 (2) 访问特殊片上外围 I/O 寄存器。

18.2 检测复位源的寄存器

V850ES/HG2 有 4 个复位源。当一个复位释放后，出现的复位源可以通过复位源标志寄存器(RESF)检测。

(1) 复位源标志寄存器 (RESF)

RESF 寄存器是一个特殊的寄存器，只能够通过特殊的顺序来写入 (参见 3.4.7 特殊寄存器)。

RESF 寄存器表明了复位信号产生的源。

该寄存器支持 8 位或 1 位的读写方式。

$\overline{\text{RESET}}$ 引脚输入该寄存器 POC 复位后的值为 00H。如果复位源不是 $\overline{\text{RESET}}$ 引脚信号，缺省值则有所不同。

复位后: 00H^注 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	没有发生
1	发生

CLMRF	来自CLM的复位信号
0	没有发生
1	发生

LVIRF	来自LVI的复位信号
0	没有发生
1	发生

注 由 $\overline{\text{RESET}}$ 引脚引起的复位将把 RESF 寄存器的值清为 00H。由看门狗定时器 2(WDT2)，低电压检测器 (LVI)，或者时钟监视器(CLM)引起的复位，寄存器复位标志(WDT2RF 位，CLMRF 位，和 LVIRF 位)被设置。但是，其他源保留。

注意事项 寄存器的每一位只能被写入“0”。如果写入“0”和设置标志(复位事件)冲突，设置标志优先。

18.3 操作

18.3.1 通过 $\overline{\text{RESET}}$ 引脚的复位操作

当 $\overline{\text{RESET}}$ 引脚输入一个低电平，系统复位，并且所有硬件初始化。

当 $\overline{\text{RESET}}$ 引脚的电平由低变为高，复位状态释放。

表 18-1. $\overline{\text{RESET}}$ 引脚输入后的硬件状态

项目	复位期间	复位后
主时钟振荡器 (fx)	停止振荡	开始振荡
副时钟振荡器 (fxT)	晶体振荡器	振荡继续
	RC 振荡器	停止振荡 开始振荡
内部振荡器	停止振荡	开始振荡
外围时钟 (fx ~ fx/1, 024)	停止操作	安全振荡稳定时间之后开始振荡
内部系统时钟 (fclk), CPU 时钟 (fcPU)	停止操作	安全振荡稳定时间之后开始操作(初始化为 fx/8)
CPU	初始化	安全振荡稳定时间之后开始执行程序
看门狗定时器 2	停止操作(初始化到 0)	开始操作
内部 RAM	如果上电复位或者 CPU 通路和复位输入冲突(数据被破坏)则不确定。 其余值在复位后则立即保留 ^{注1}	
I/O 总线(端口/复用功能引脚)	高阻抗 ^{注2}	
片上外围 I/O 寄存器	初始化为指定状态, OCDM 寄存器被设置为(01H)。	
其他片上外围功能	停止操作	安全振荡稳定时间之后可以开始操作

<R>

- 注 1. V850ES/HG2 的固件使用了的系统复位释放后操作内部 RAM 的部分一部分，支持一个启动交换功能。因此，一些 RAM 区域的部分不能保持上电复位值，更多细节，敬请参阅 18.4 复位释放后的操作。
2. 当电源开启，下列引脚可能临时在复位时输出一个不确定的电平。

- P53/KR3/TIQ00/TOQ00/DDO 引脚

注意事项 $\overline{\text{RESET}}$ 引脚的输入初始化 OCDM 寄存器。因此，注意注意事项，复位释放后在 OCDM.OCDM0 位被清除前如果一个高电平输入到 P05/DRST 引脚，可能进入片上调试模式。更多细节，敬请参阅第四章端口功能。

图 18-1. RESET 引脚输入引起的复位操作时序

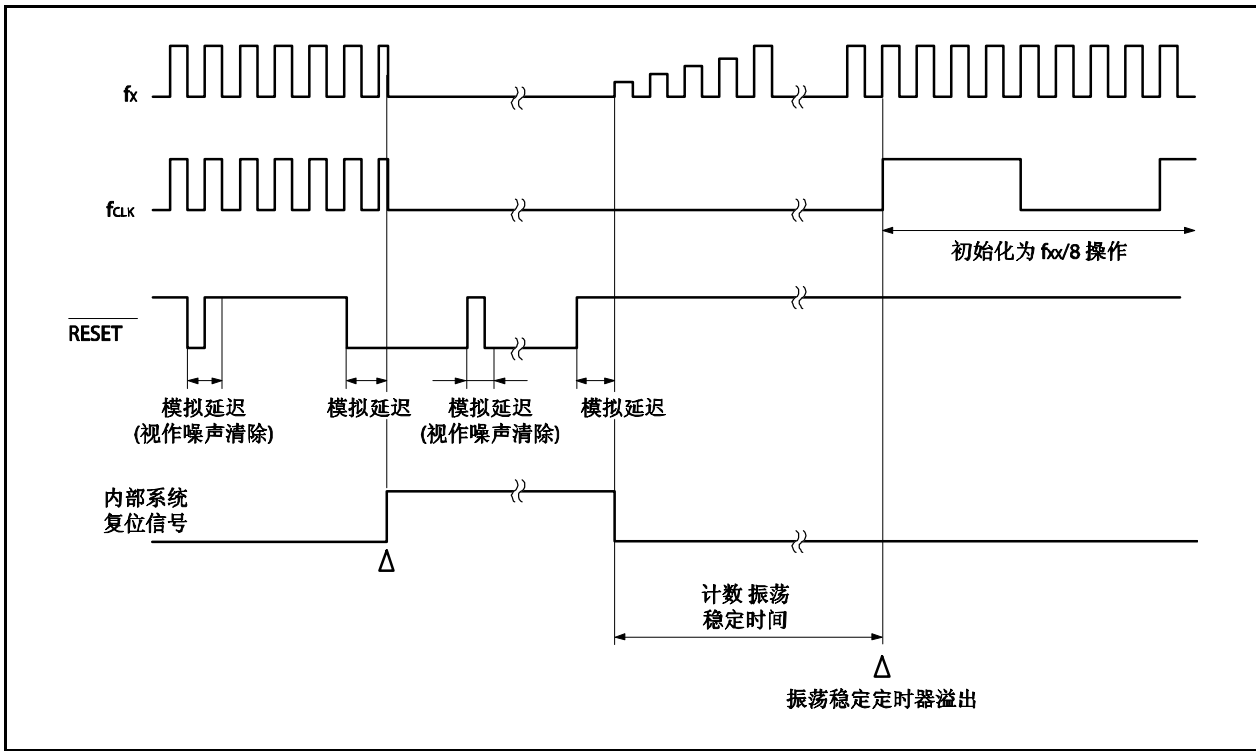
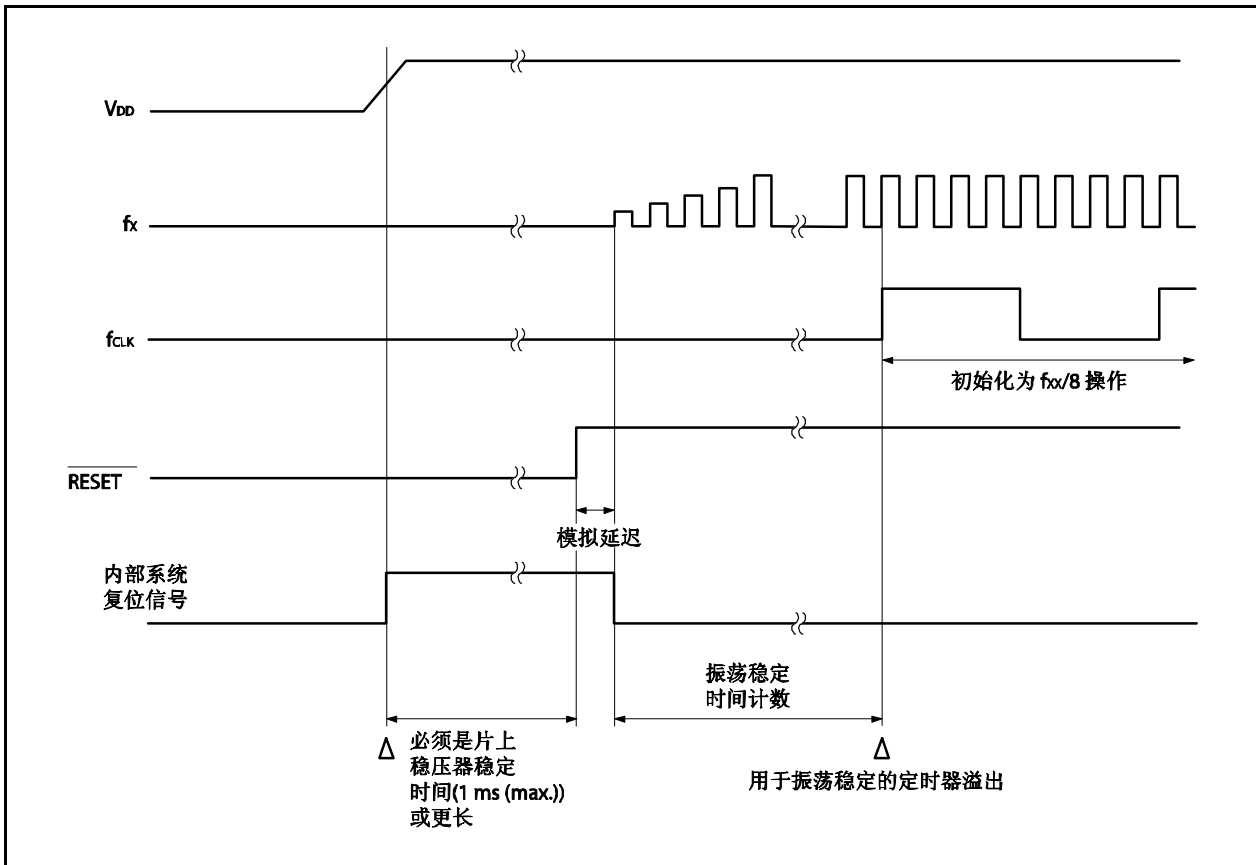


图 18-2. 上电复位操作时序



18.3.2 由看门狗定时器 2 引起的复位操作

当看门狗定时器 2 溢出被设置为复位操作模式，在看门狗定时器 2 溢出上(产生 WDT2RES 信号)，执行系统复位并且硬件初始化到初始化状态。

在看门狗定时器 2 溢出后，进入复位状态并且持续到预定时间(模拟延时)，并且复位状态自动释放。

在复位期间主时钟振荡停止。

表 18-2. 看门狗定时器 2 复位操作时的硬件状态

项目		复位期间	复位后
主时钟振荡器(f_x)		停止振荡	开始振荡
副时钟振荡器 (f_{xt})	晶体振荡器	继续振荡	
	RC 振荡器	停止振荡	开始振荡
内部振荡器		停止振荡	开始振荡
外围时钟 ($f_{xx} \sim f_{xx}/1, 024$)		停止操作	安全振荡稳定时间之后开始操作
内部系统时钟 (f_{xx}), CPU 时钟 (f_{CPU})		停止操作	安全振荡稳定时间之后开始振荡(初始化为 $f_{xx}/8$)
CPU		初始化	安全振荡稳定时间之后开始执行程序
看门狗定时器 2		停止操作(初始化为 0)	开始操作
内部 RAM		如果上电复位或者 CPU 通路和复位输入冲突(数据被破坏)则不确定。 其余值在复位后则立即保留 ^注 。	
I/O 总线 (端口/复用功能引脚)		高阻抗	
片上外围 I/O 寄存器		初始化为指定状态, OCDM 寄存器值保留。	
其他片上外围功能		停止操作	安全振荡稳定时间之后可以开始操作

<R> 注 V850ES/HG2 的固件使用了的系统复位释放后操作内部 RAM 的部分一部分，支持一个启动交换功能。因此，一些 RAM 区域的部分不能保持上电复位值。关于更多细节，敬请参阅 18.4 复位释放后的操作。

18.3.3 上电清零电路的复位操作

当启用上电清零操作时，将对电源电压及检测电压进行比较。如果电源电压低于检测电压（包括使用电源时），复位系统，硬件单元初始化至默认值。

复位状态自检测到电压降低一直持续至电源电压高于检测电压，然后自动清零。在清零复位状态后，擦除主时钟振荡器（OSTS 寄存器：默认值 $2^{16}/f_x$ ）振荡的所需稳定时间。关于更多细节，敬请参阅**第二十章片上清零电路**。

18.3.4 低电压检测器引起的复位操作

当启用 LVI 操作，LVIM.LVIMD 位设置为 1 时，将比较电源电压及检测电压。如果电源电压低于检测电压，复位系统，硬件单元初始化至默认值。

复位状态自检测电压降低一直持续至电源电压高于检测电压，然后自动清零。在清零复位状态后，擦除主时钟振荡器（OSTS 寄存器：默认值 $2^{16}/f_x$ ）振荡的所需稳定时间。然后 CPU 开始执行程序。

关于更多细节，敬请参阅**第二十一章低电压检测器**。

18.3.5 时钟监视器的复位操作

当启用时钟监视操作时，通过使用采样时钟（内部振荡器）监视主时钟。当检测到主时钟停止时，复位系统，硬件单元初始化至默认值。

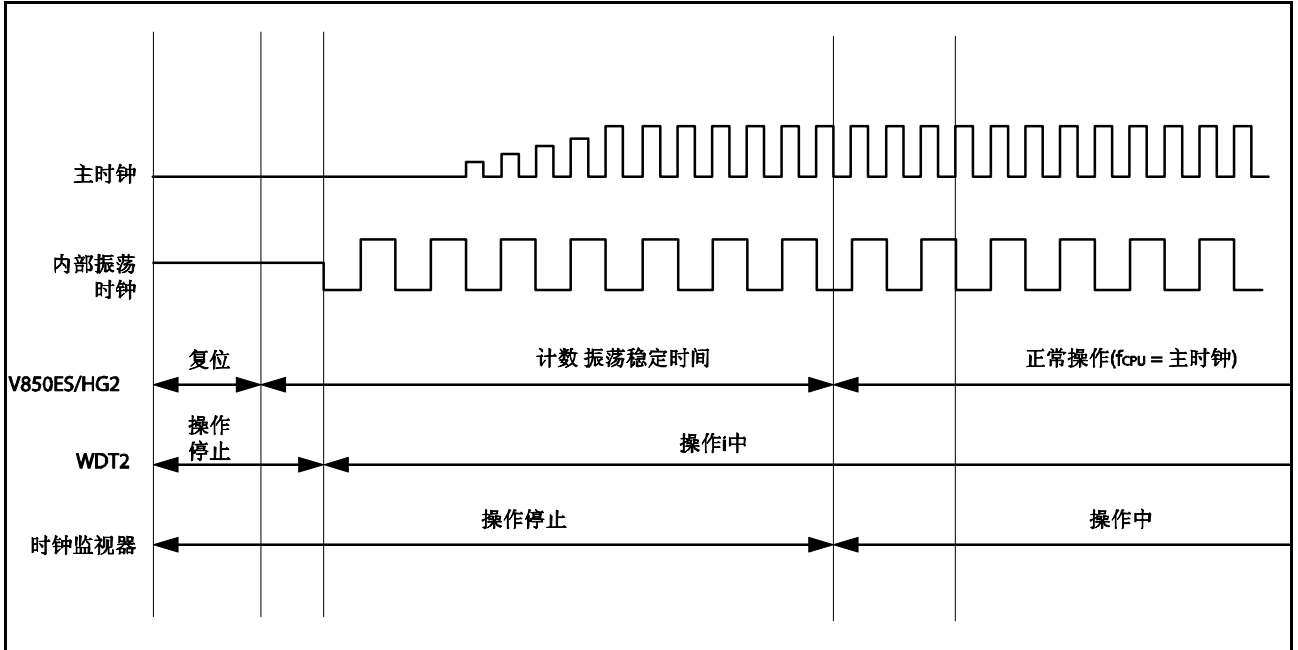
关于更多细节，敬请参阅**第十九章 时钟监视器**。

<R>

18.4 复位释放后的操作

复位释放以后，主时钟开始振荡并且振荡稳定时间(OSTS 寄存器初始值为： $2^{16}/f_x$)可靠，CPU 开始执行程序。在复位释放后 WDT2 立即以内部振荡时钟为源时钟开始操作。

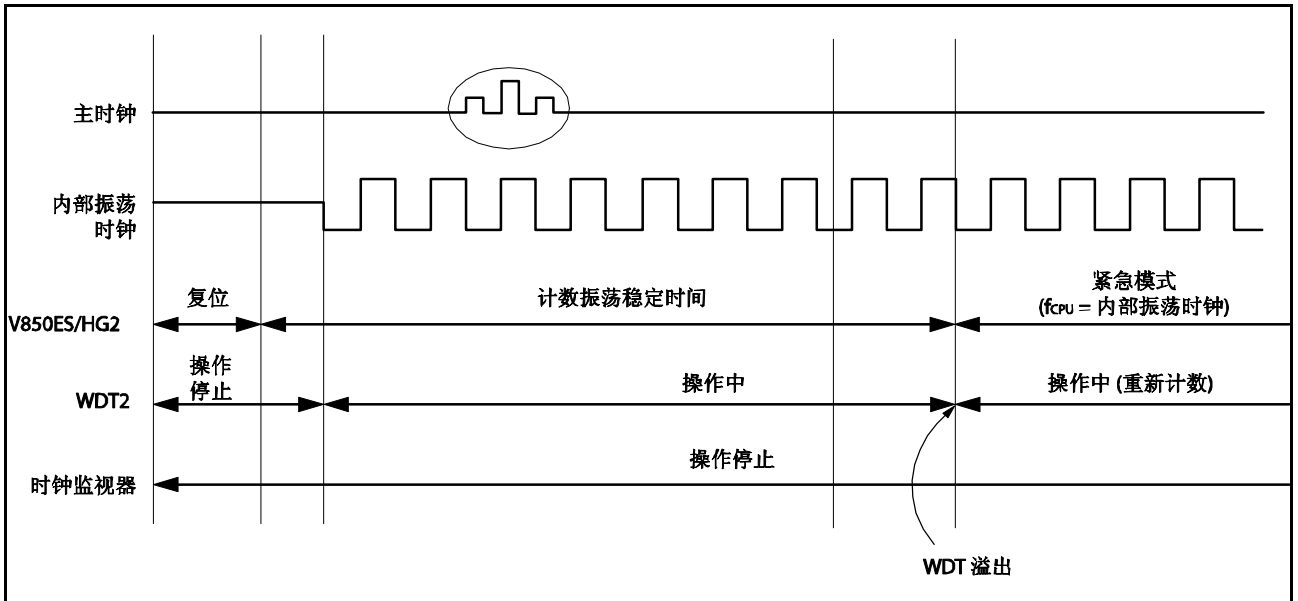
图 18-3. 复位释放后的操作



(1) 紧急操作模式

如果在安全振荡稳定时间之前一个主时钟的异常发生。WDT2 在执行 CPU 程序之前溢出。此时，CPU 使用内部振荡时钟作为源时钟开始执行程序。

图 18-4. 复位释放后的操作

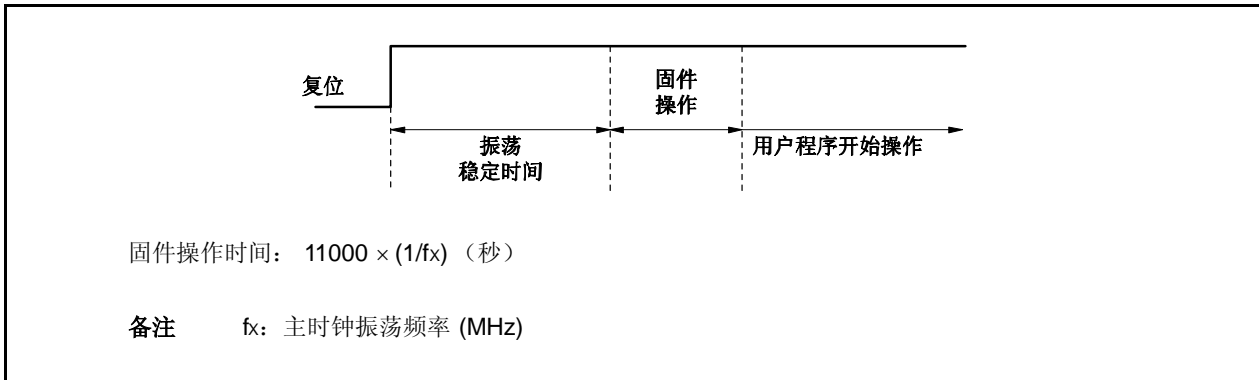


CPU 操作时钟状态可以通过 CPU 操作时钟状态寄存器(CCLS)来检测。

(2) 固件操作

V850ES/HG2 复位释放以后，开始执行用户程序之前片上固件操作支持启动切换功能。

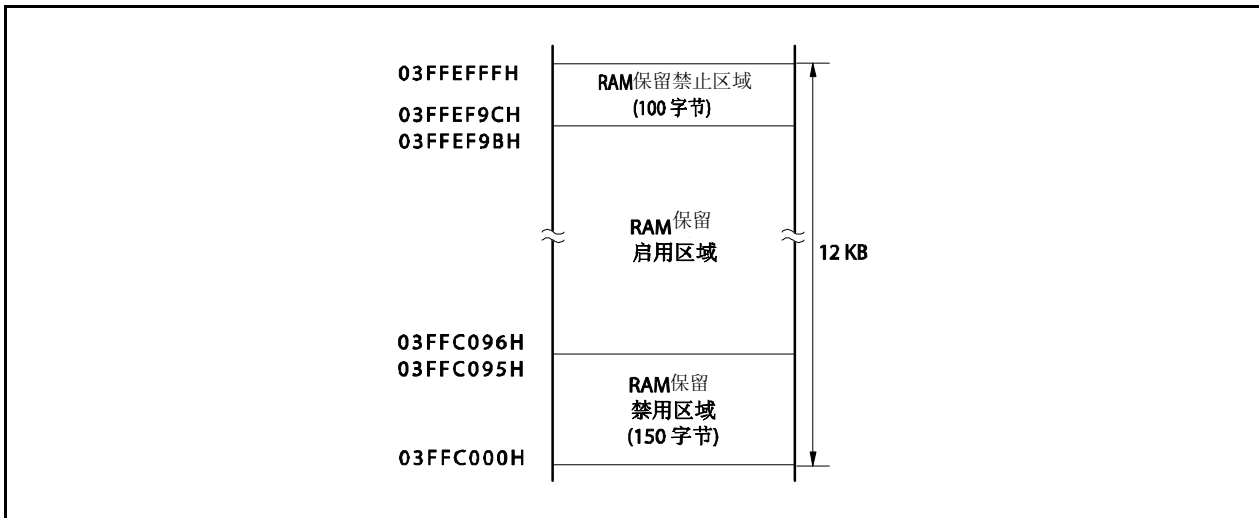
图 18-5. 固件操作



自从固件使用一部分内部 RAM，下列 RAM 区域在复位上电状态连接不连续。

- 12 KB RAM 版本: 03FFC000H~ 03FFC095H, 03FFEF9CH~ 03FFEFFFH

图 18-6. RAM 保留启用区域



第十九章 时钟监视器

19.1 功能

时钟监视器通过使用内部振荡时钟抽取主时钟，并且在主时钟停止振荡的时候产生一个复位信号。

一旦通过操作使能标志允许时钟监视器的操作，除了复位无法通过其余任何方法来清零。

由时钟监视器产生复位时，RESF.CLMRF 位被设置，关于 RESF 寄存器的更多细节，敬请参见 **18.2 检测复位源的寄存器**。

时钟监视器在下列条件下自动停止。

- STOP 模式释放后的振荡稳定时间内
- 当主时钟停止时(从副时钟操作 PCC.MCK 位= 1 开始，到主时钟操作 PCC.CLS 位= 0 结束)
- 当采样时钟(内部振荡时钟)停止时
- 当 CPU 使用内部振荡时钟操作时

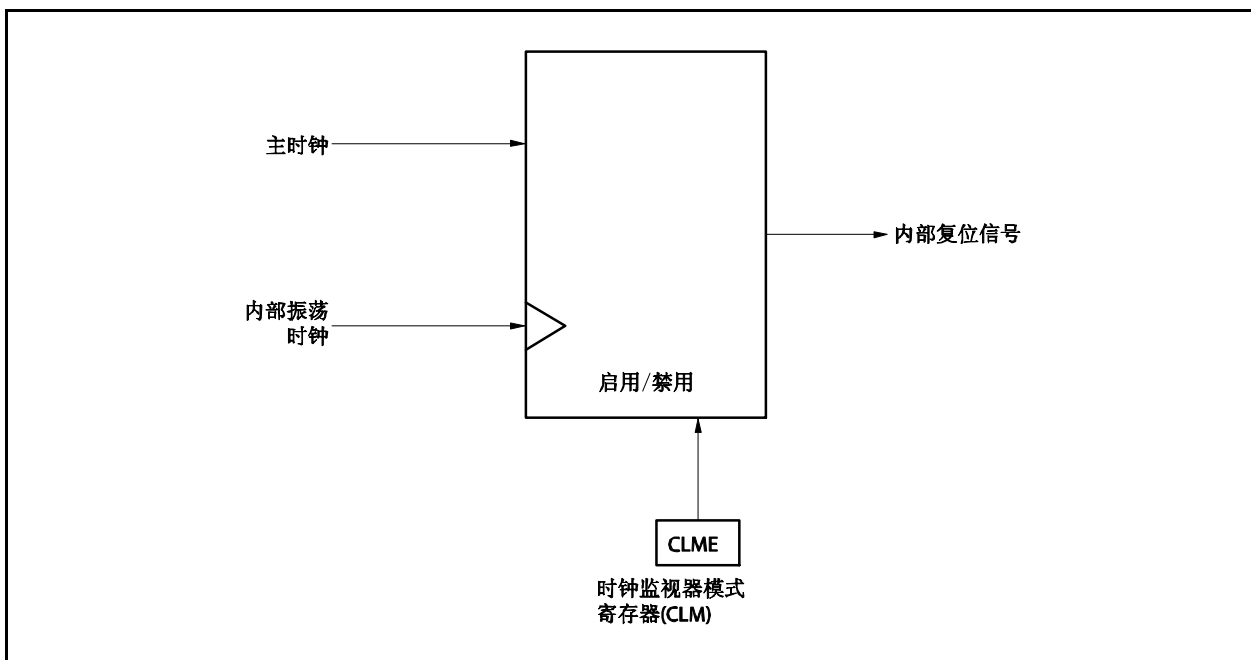
19.2 配置

时钟监视器有下列硬件组成。

表 19-1. 时钟监视器的配置

项目	配置
控制寄存器	时钟监视器模式寄存器 (CLM)

图 19-1. 时钟监视器的框图



19.3 寄存器

时钟监视器通过时钟监视器模式寄存器(CLM)来控制。

(1) 时钟监视器模式寄存器(CLM)

CLM 寄存器是一个特殊寄存器。只能够通过特殊的顺序来写入(参见 3.4.7 特殊寄存器)。

这个寄存器用来设置操作时钟监视器的模式。

该寄存器支持 8 位或 1 位读写方式。

该寄存器复位后的值为 00H。

复位后: 00H	R/W	地址: FFFFF870H						
	7	6	5	4	3	2	1	0
CLM	0	0	0	0	0	0	0	CLME
CLME	时钟监视器操作使能或禁用							
0	禁用时钟监视器操作。							
1	使能时钟监视器操作。							

- 注意事项**
1. 一旦 **CLME** 位被设为 1，除了复位其余任何操作都不能将其清为零。
 2. 由时钟监视器引起的一个复位，**CLME** 位被清为零并且 **RESF.CLMRF** 被设置为 1。

19.4 操作

这个部分解释了时钟监视器的功能。下面是开始和停止条件。

<开始操作>

当 CLM.CLME 位设置为 1 时使能操作。

<停止条件>

- 当 STOP 模式释放后开始计数振荡稳定时间时
- 当主时钟停止(从副时钟操作时 PCC.MCK 位=1 到主时钟操作时 PCC.CLS 位=0 为止)时
- 当采样时钟停止(内部振荡时钟)时
- 当 CPU 使用内部振荡时钟操作时

表 19-2. 时钟监视器的操作状态
(当 CLM.CLME 位 = 1, 在内部振荡时钟操作时)

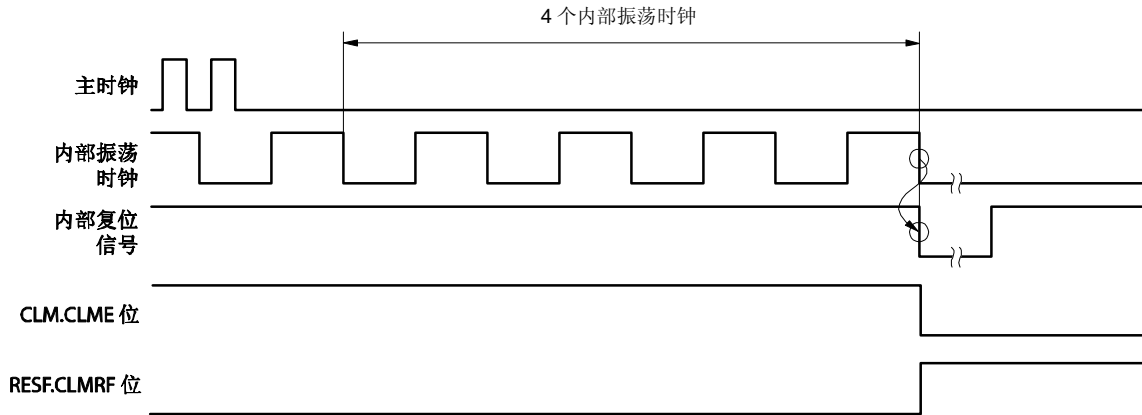
CPU 操作时钟	操作模式	主时钟状态	内部振荡时钟状态	时钟监视器状态
主时钟	HALT 模式	振荡	振荡 ^{注1}	操作 ^{注2}
	IDLE1, IDLE2 模式	振荡	振荡 ^{注1}	操作 ^{注2}
	STOP 模式	停止	振荡 ^{注1}	停止
副时钟(PCC 寄存器的 MCK 位 = 0)	Sub-IDLE 模式	振荡	振荡 ^{注1}	操作 ^{注2}
副时钟 (PCC 寄存器的 MCK 位 = 1)	Sub-IDLE 模式	停止	振荡 ^{注1}	停止
内部振荡时钟	—	停止	振荡 ^{注1}	停止
复位期间	—	停止	停止	停止

- 注**
1. 可通过使用选项字节功能（参见**第二十四章**）来停止内部振荡器，设置 RCM.RSTOP 位为 1。
 2. 当内部振荡器停止操作时，时钟监视器停止。

(1) 当主时钟振荡器停止时(CLME 位 = 1)的操作

当 CLME 位=1 的时候主时钟的振荡停止，一个如图 19-2 的内部复位信号就会产生。

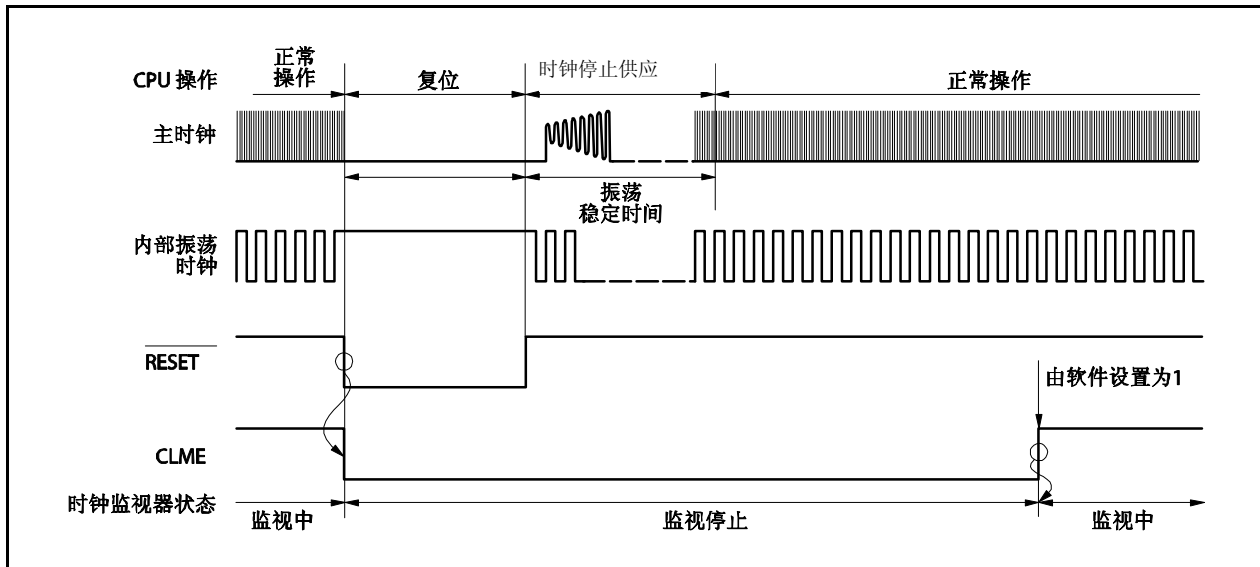
图 19-2. 主时钟振荡停止引起的复位周期



(2) $\overline{\text{RESET}}$ 输入后的时钟监视器状态

RESET 输入清除 CLM.CLME 位为 0 并且停止时钟监视器的操作。在主时钟振荡稳定时间后当通过软件把 CLME 位设为 1，监控操作开始。

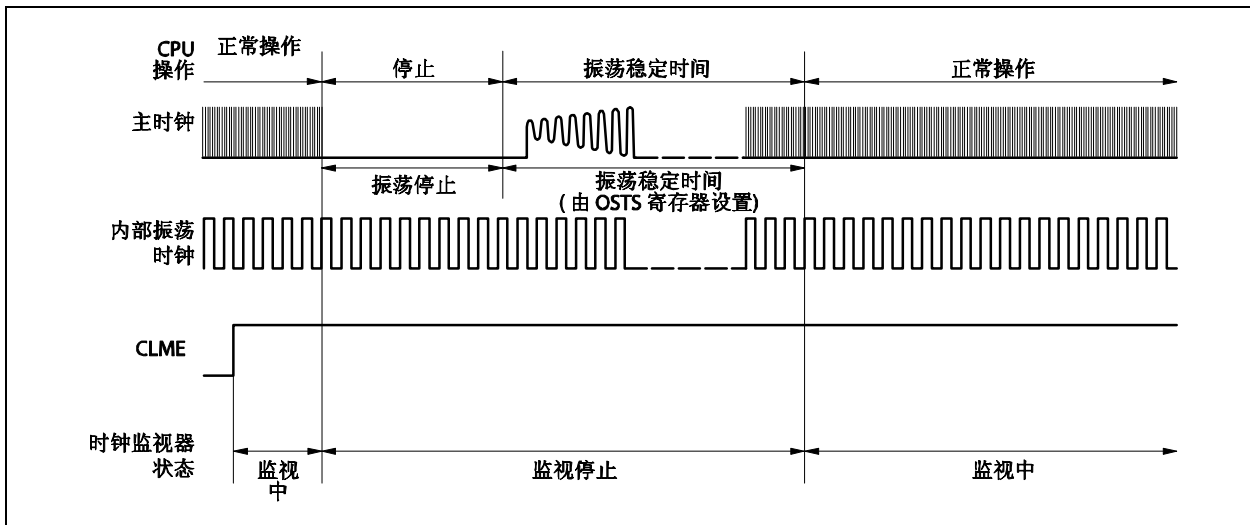
图 19-3. $\overline{\text{RESET}}$ 输入后的时钟监视器状态
RESET 输入并且在主时钟振荡稳定时间后设置 CLM.CLME 位=1)



(3) STOP 模式 或 STOP 模式释放后的操作

当设置 CLM.CLME 位=1 时进入 STOP 模式，在 STOP 模式和振荡稳定时间计数下监控操作停止。振荡稳定时间过后，监视器自动开始操作。

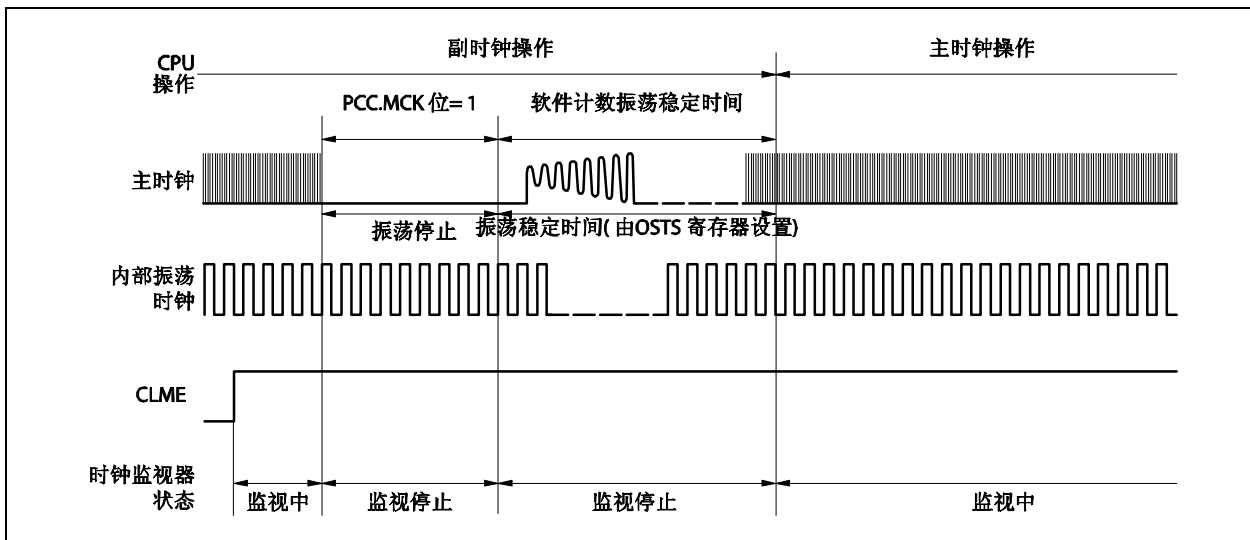
图 19-4. STOP 模式 或 STOP 模式 释放后的操作



(4) 主时钟停止时的操作 (任意时刻)

副时钟操作期间(PCC.CLS 位=1)或者通过设置 PCC.MCK 位为 1 来停止主时钟，到主时钟开始操作之前 (PCC.CLS 位=0)监视器停止操作。直到主时钟开始操作监视器才自动开始操作。

图 19-5. 主时钟停止时的操作 (任意时刻)



(5) 当 CPU 使用内部振荡时钟时的操作(CCLS.CCLS F 位 = 1)

当 CCLS F 位设为 1 时监视器的操作不停止，即时 CLME 位被设为 1。

第二十章 上电清零电路

20.1 功能

上电清零电路（POC）功能显示如下。

- 当使用电源时产生一个复位信号。
- 比较电压(V_{DD})及探测电压(V_{POC0})，当 $V_{DD} < V_{POC0}$ (探测电压(V_{POC0}): $3.7\text{ V} \pm 0.2\text{ V}$)时产生一个复位信号。

备注 1. V850ES/HG2 具有多个内部硬件单元，他们可以产生一个内部复位信号。当系统被看门狗定时器 2 (WDT2RES)，低电压探测器 (LVI)，或时钟监测器 (CLM) 等设备复位时，与复位源相对应的标志将分配到复位源标志寄存器 (RESF)。

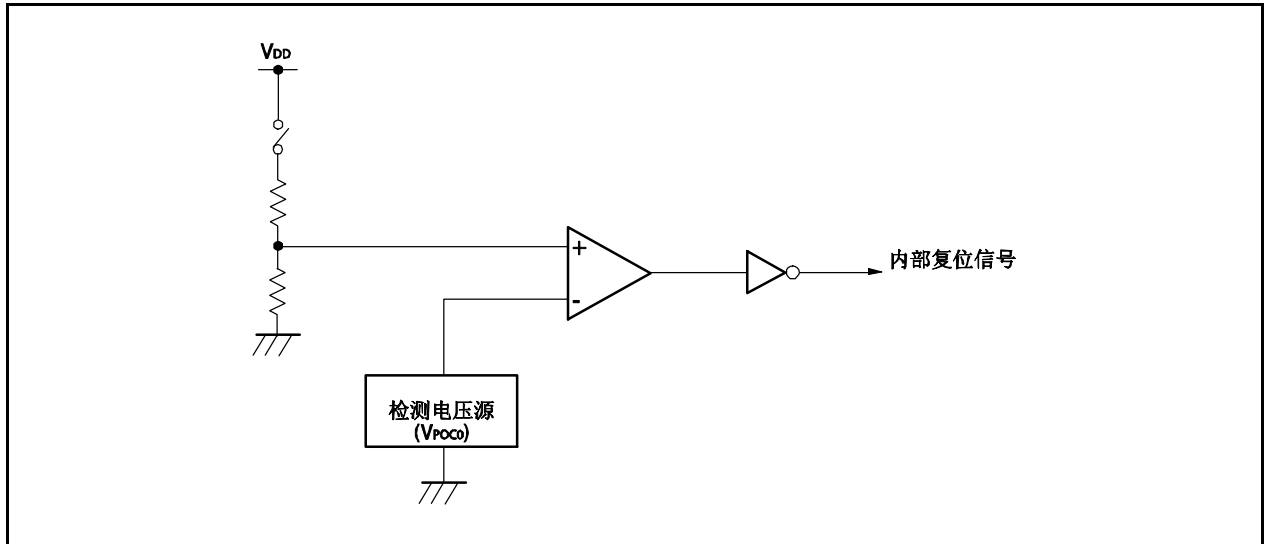
当 WDT2RES, LVI 或时钟监测器产生一个内部复位信号时，RESF 寄存器不会被清除，并且与复位源相对应的标志将设置为 1。关于 RESF 寄存器的更多细节，敬请参阅第十八章 复位功能。

2. 如果外接共鸣器的工作频率为 5MHz 时，使用电源到开始执行程序的时间是“从使用电源到释放复位的时间+16ms”。然而，根据外部原因的不同，时间也会发生变化（如果微控制器电压的状态及谐振器的稳定时间）。

20.2 配置

模块框图显示如下。

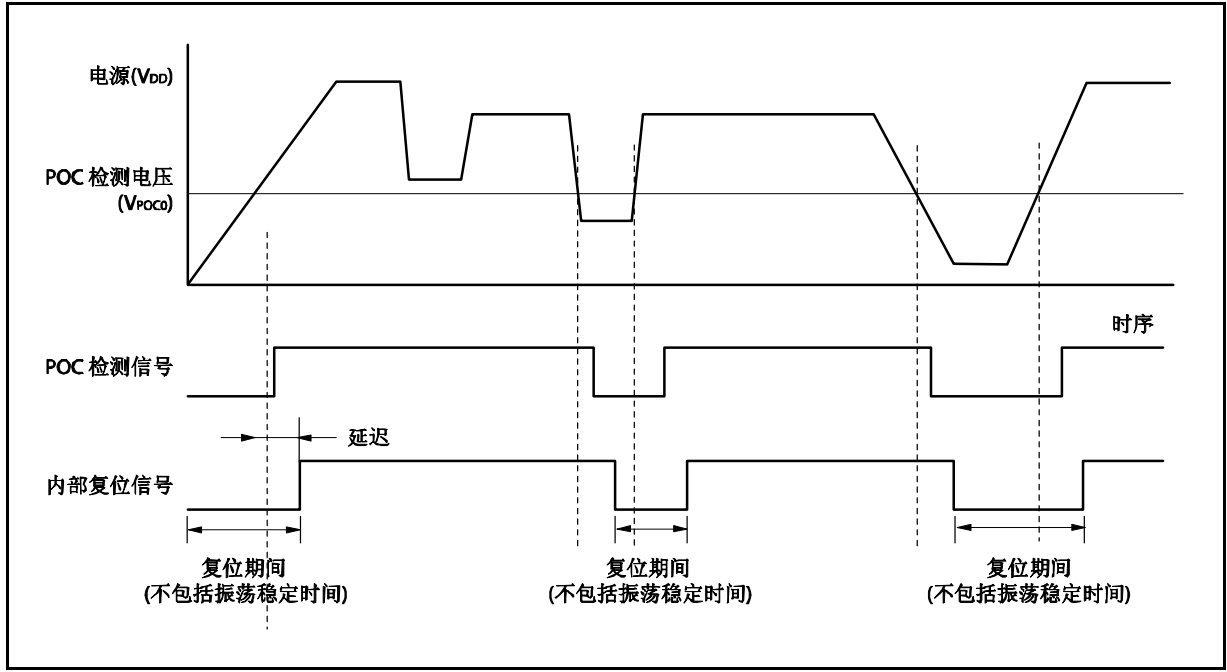
图 20-1. 上电清零电路的模块框图



20.3 运行

当比较电源电压及探测电压时，如果电源电压比探测电压低（包括使用电源时），系统将复位并且每个硬件返回到特定的状态。

图 20-2. 上电清零电路产生的复位信号定时



第二十一章 低电压检测器

21.1 功能

低电压检测器 (LVI) 有以下功能。

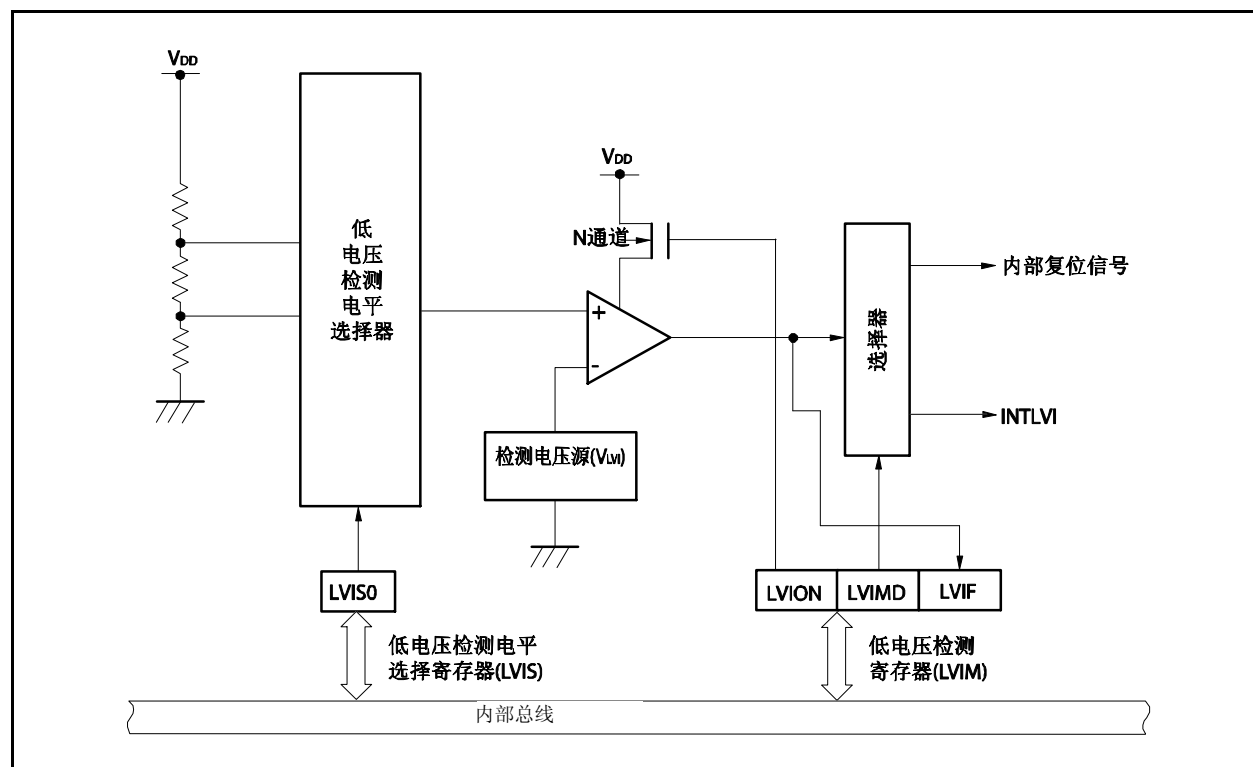
- LVI 电路比较供电电压 (V_{DD}) 和检测电压 (V_{LVI})，当 $V_{DD} < V_{LVI}$ 时产生内部复位信号或内部中断信号。
- 可由软件改变供电电压的检测等级 (2 个等级)。
- 可以选择中断请求信号或内部复位信号。
- 在 STOP 模式下可操作。
- 可以通过软件停止操作。

当低电压检测电路用作产生复位时，如果复位产生，复位控制标志寄存器 (RESF) 的第 0 位 (LVIRF) 被置 1。如欲了解 RESF 的详细信息，敬请参阅第十八章复位功能。

21.2 配置

框图如下所示。

图 21-1. 低电压检测器的框图



21.3 寄存器

(1) 低电压检测寄存器(LVIM)

LVIM 寄存器用来使能或禁用低电压检测，并设置其操作模式。

此寄存器是一个特殊的寄存器。它只有在特殊的顺序组合时才可以写(参见 3.4.7 特殊寄存器)。

该寄存器支持 8 位或 1 位读写方式。然而，LVIF 位只读。

复位后: 00H	R/W	地址: FFFFF890H							
		7	6	5	4	3	2	1	0
LVIM		LVION	0	0	0	0	0	LVIMD	LVIF
	LVION	使能或禁用低电压检测器操作							
	0	禁用操作							
	1	使能操作							
	LVIMD	低电压操作模式的选择							
	0	当供应电压小于检测电压时，发生中断请求信号 INTLVI。							
	1	当供应电压小于检测电压时，发生内部复位信号 LVIRE。							
	LVIF	低电压检测标志							
	0	当供电电压 > 检测电压，或当禁用操作时							
	1	供电电压 < 检测电压							

- 注意事项**
1. 在设置 LVION 位为 1 后，在使用 LVIF 位检测电压前等待 0.2 ms (MAX.)。
 2. LVION 位=1 及 LVIMD 位=0 时，LVIF 标志的值作为输出信号 INTLVI 输出。
 3. 一定要将位 2~6 清零。
 4. 在 LVION 位及 LVIMD 位设置为 1 后，发生了非低电压检测，因此，低电压检测电路无法停止除非接收到复位信号。

<R>

(2) 低电压检测等级选择寄存器(LVIS)

该寄存器用于选择低电压检测等级。

该寄存器支持 8 位读写方式。

复位后: 00H	R/W	地址: FFFFF891H						
	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	0	0	LVIS0
LVIS0	检测等级							
0	4.4 V \pm 0.2 V							
1	4.2 V \pm 0.2 V							

注意事项 1. 在 LVION.LVION 位及 LVIMD.LVIMD 位设置为 1 后, 发生了非低电压检测, 因此, 无法写入该寄存器除非接收到复位请求。

2. 一定要将位 1~7 清零。

(3) 内部 RAM 数据状态寄存器(RAMS)

RAMS 寄存器是一个标志寄存器, 用来指示内部 RAM 是否有效。

RAMS 寄存器是一个特殊寄存器。它只有在特殊的顺序组合时才可以写(参见 3.4.7 特殊寄存器)。

对于 RAMS 寄存器, 参见 21.5RAM 保持电压检测操作。

该寄存器支持 8 位或 1 位读写方式。

注意事项 下面所示的是复位后的具体顺序。

- 检测到比设定电压低的电压
 - 由指令设置
 - 通过看门狗定时器溢出发生复位信号
 - 当访问 RAM 时, 产生复位信号
 - 通过时钟监测器发生复位信号
- 清除条件: 以特定顺序写入零

复位后: 01H	R/W	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							

21.4 操作

依据 LVIM.VIMD 位的设置，产生中断请求信号 (INTLVI) 或者内部复位信号。

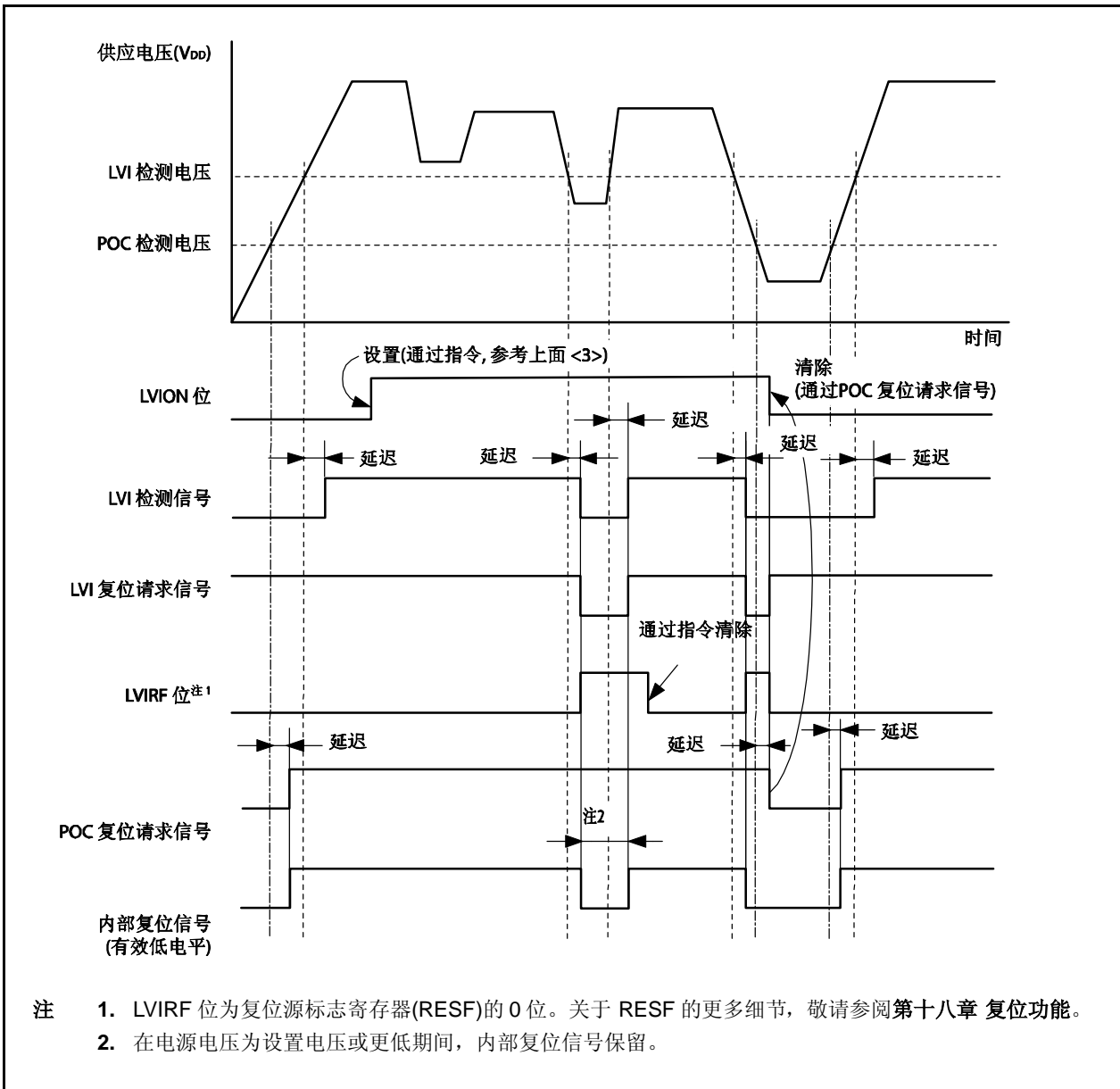
21.4.1 使用内部复位信号

<开始操作>

- <1> 屏蔽 LVI 中断。
- <2> 由 LVIS.LVIS0 位选择检测的电压。
- <3> 设置 LVIM.LVION 位为 1 (使能操作)。
- <4> 由软件插入 0.2 ms (max.) 的等待周期。
- <5> 由 LVIM.LVIF 位，检验是否供电源电压 > 检测电压。
- <6> 设置 LVIM.LVIMD 位为 1 (产生内部复位信号)。

注意事项 如果 LVIMD 位设置为 1，那么 LVIM 和 LVIS 寄存器的内容不能改变，直到 LVI 之外的复位请求产生。

图 21-2. 低电压检测器的操作时序 (LVIMD 位 = 1)



21.4.2 用作中断

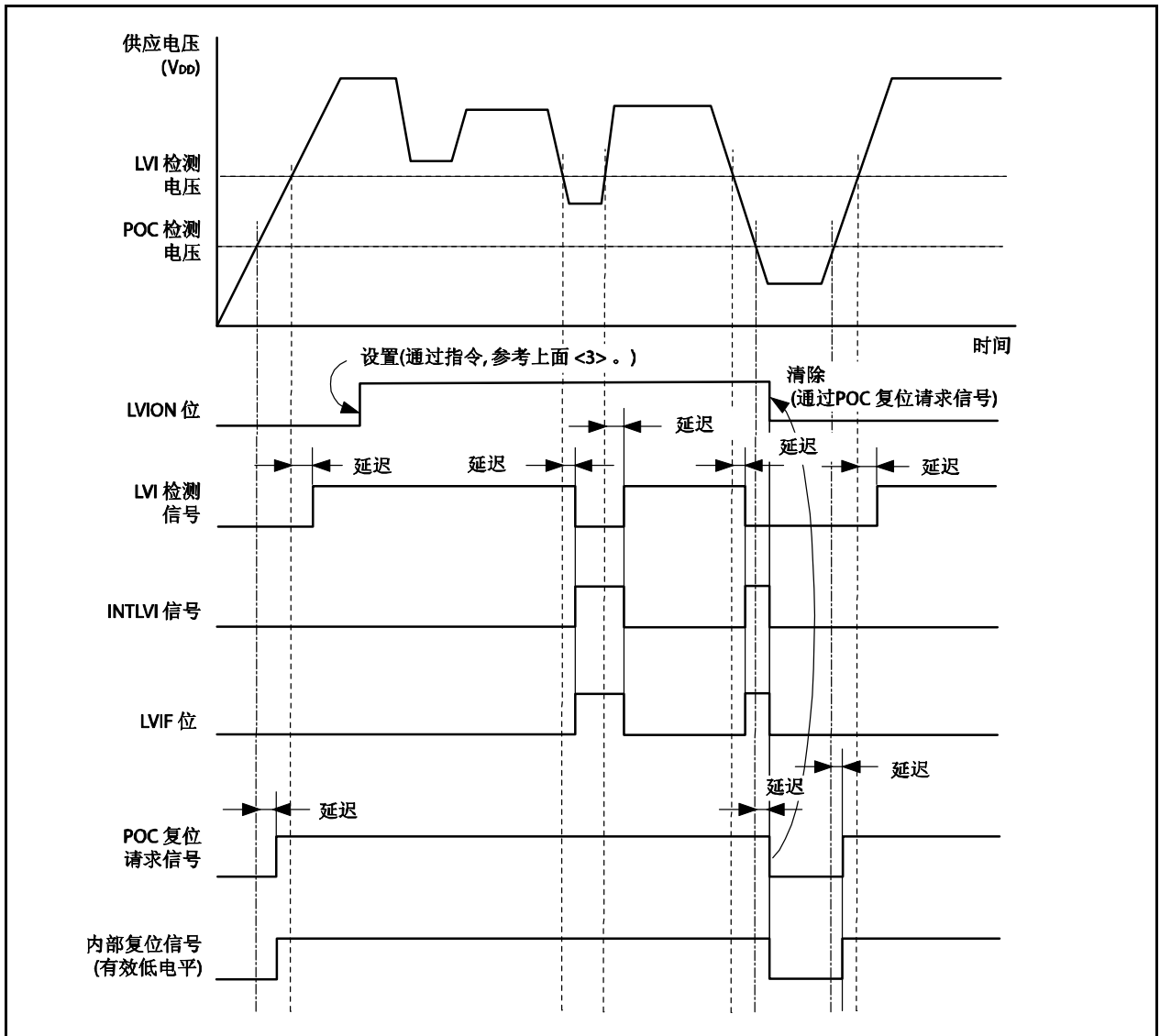
<开始操作>

- <1> 屏蔽 LVI 中断。
- <2> 由 LVIS.LVIS0 位选择检测的电压。
- <3> 设置 LVIM.LVION 位为 1 (使能操作)。
- <4> 由软件插入 0.2 ms (max.) 的等待周期。
- <5> 由 LVIM.LVIF 位, 检验是否供电电压 > 检测电压。
- <6> 清除 LVI 的中断请求标志。
- <7> 打开 LVI 中断。

<停止操作>

将 LVION 位清零。

图 21-3. 低电压检测器的操作时序 (LVIM 位 = 0)

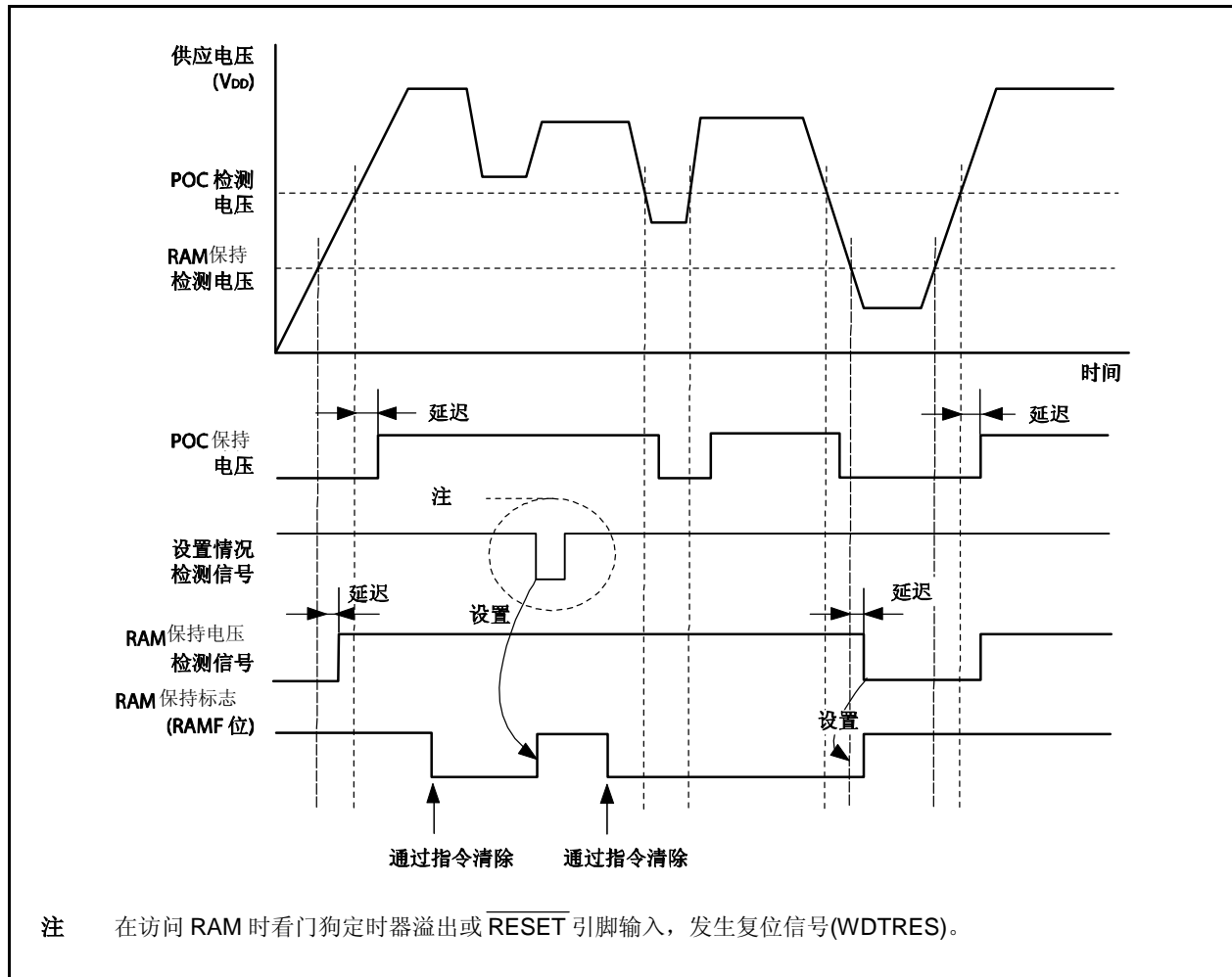


21.5 RAM 保持电压检测操作

比较供电电压和检测电压。当其低于检测电压时(包括上电), RAMS.RAMF 位设置为 1。

当不使用 POC 功能, 而使用 RAM 保留电压监测功能时, 如果检测电压低于操作电压, 一定要输入一外部访问信号。

图 21-4. RAM 保持电压检测器功能的操作时序



21.6 仿真功能

当使用在线仿真器时，RAM 保持标志(RAMS.RAMF 位)的操作可以通过操作调试器上的 PEMU1 寄存器进行伪控制和仿真。

该寄存器只有在仿真模式下有效，正常模式下无效。

(1) 外围仿真寄存器 1 (PEMU1)

复位后: 00H	R/W	地址: FFFFF9FEH							
		7	6	5	4	3	2	1	0
PEMU1		0	0	0	0	0	EVARAMIN	0	0
	EVARAMIN	RAM 保持电压检测信号的假定							
	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 此位不能自动清除。

[用法]

当使用线上仿真器时，RAMF 的伪仿真通过重写调试器上的寄存器实现。

- <1> CPU 停止 (CPU 操作停止。)
- <2> 通过使用寄存器写命令设置 EVARAMIN 位为 1。
通过设置 EVARAMIN 位为 1，RAMF 位硬件上设置为 1。(内部 RAM 数据无效)。
- <3> 再次通过寄存器写命令将 EVARAMIN 位清零。
除非在进行此操作(清 EVARAMIN 位为 0)，否则 RAMF 位不能由 CPU 操作指令清零。
- <4> 运行 CPU 并重新仿真。

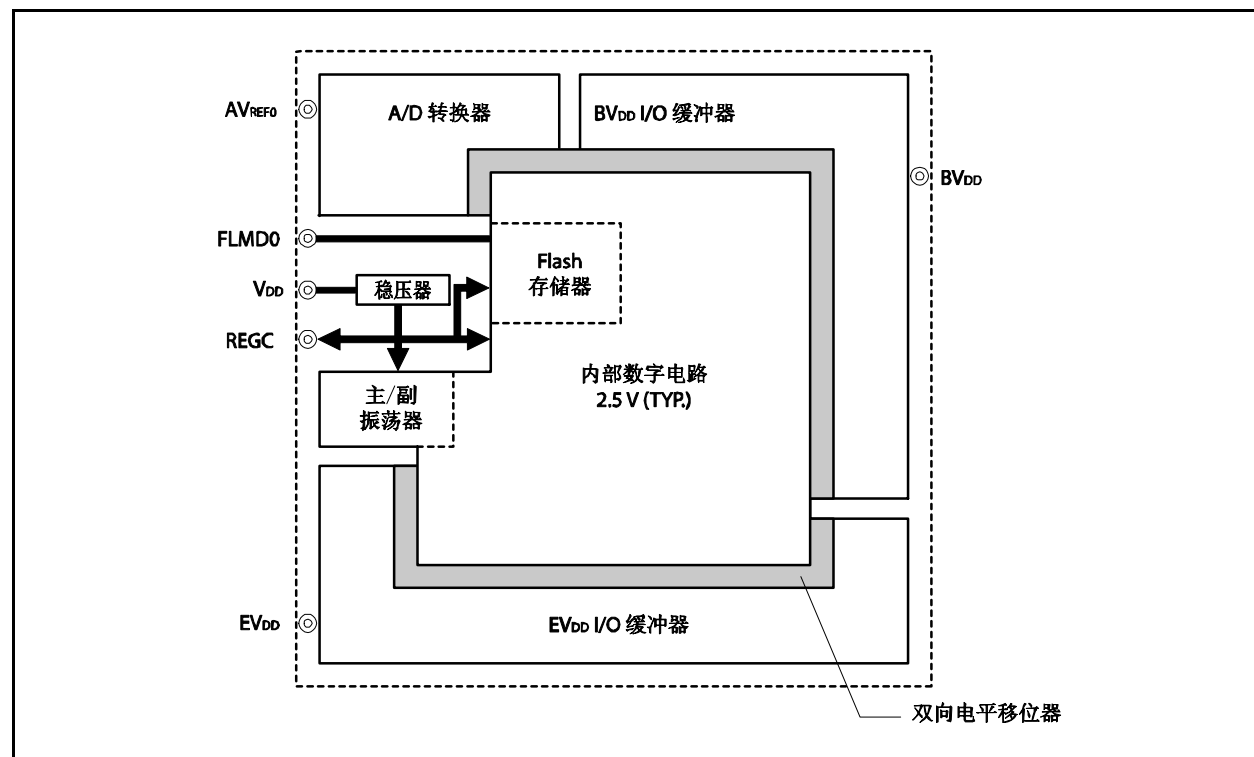
第二十二章 稳压器

22.1 概述

V850ES/HG2 包括一个可以降低能耗和噪声的稳压器。

此稳压器供应一个分阶下降的供电源 V_{DD} 给振荡器模块和内部逻辑电路(除了 A/D 转换器, D/A 转换器和输出缓冲器)。稳压器输出电压设置为 2.5 V (TYP.)。

图 22-1. 稳压器

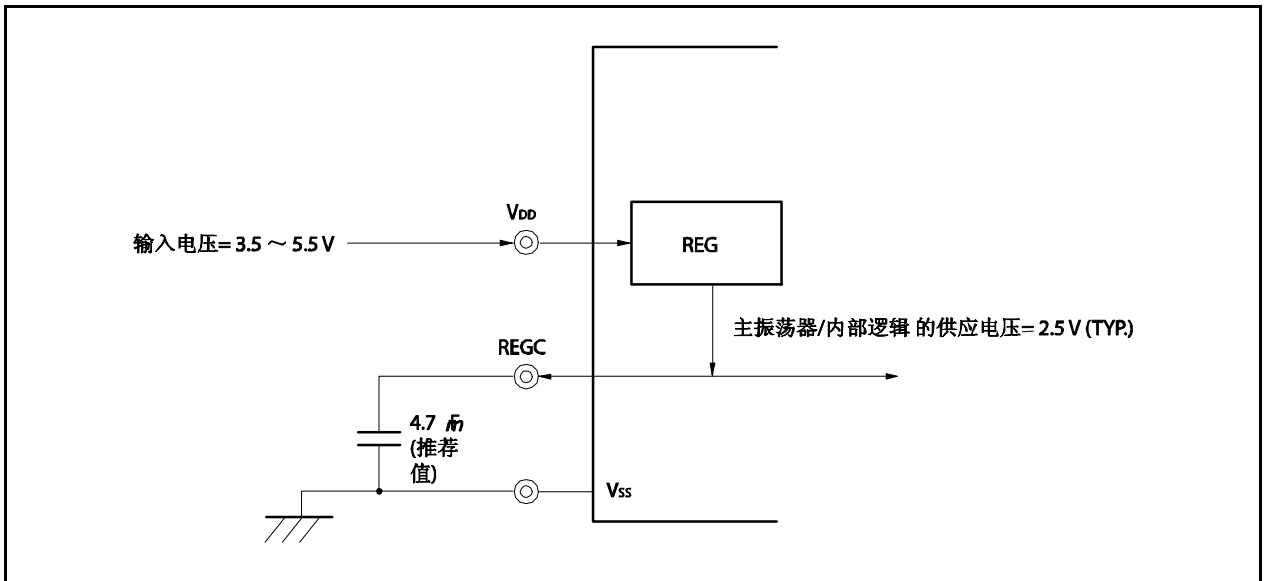


22.2 操作

产品的稳压器总是运行于各种模式下(正常操作模式, HALT 模式, IDLE1 模式, IDLE2 模式, STOP 模式或者复位)。一定要给 REGC 引脚连接一个电容器(4.7 μF (推荐值))来稳定稳压器输出。

稳压器引脚连接方法的方框图如下所示。

图 22-2. REGC 引脚连接



第二十三章 FLASH 存储器

以下可视为使用 flash 存储器版本的开发环境及大规模生产应用。

- 在将 V850ES/HG2 焊接到目标系统改写软件。
- 在开始大批量生产时的数据调整。
- 根据各种型号的小规模生产中的规格细分软件。
- 辅助存货管理。
- 在出厂后升级软件。

23.1 特点

- 4 字节/1 时钟的访问（当取出指令时）
- 容量： 256 KB/128 KB
- 写入电压： 用单独的电源擦除/写入
- 重写方法
 - 通过串行接口与专门的 Flash 存储器编程器通信来进行重写入（片上/片下编程）
 - 通过用户程序重新写入 Flash 存储器（自编程）
- 支持 Flash 存储器写禁止功能（安全功能）
- 使用引导交换功能通过自编程安全地重新写入到整个 Flash 存储器区域
- 可以在自编程过程中确认中断

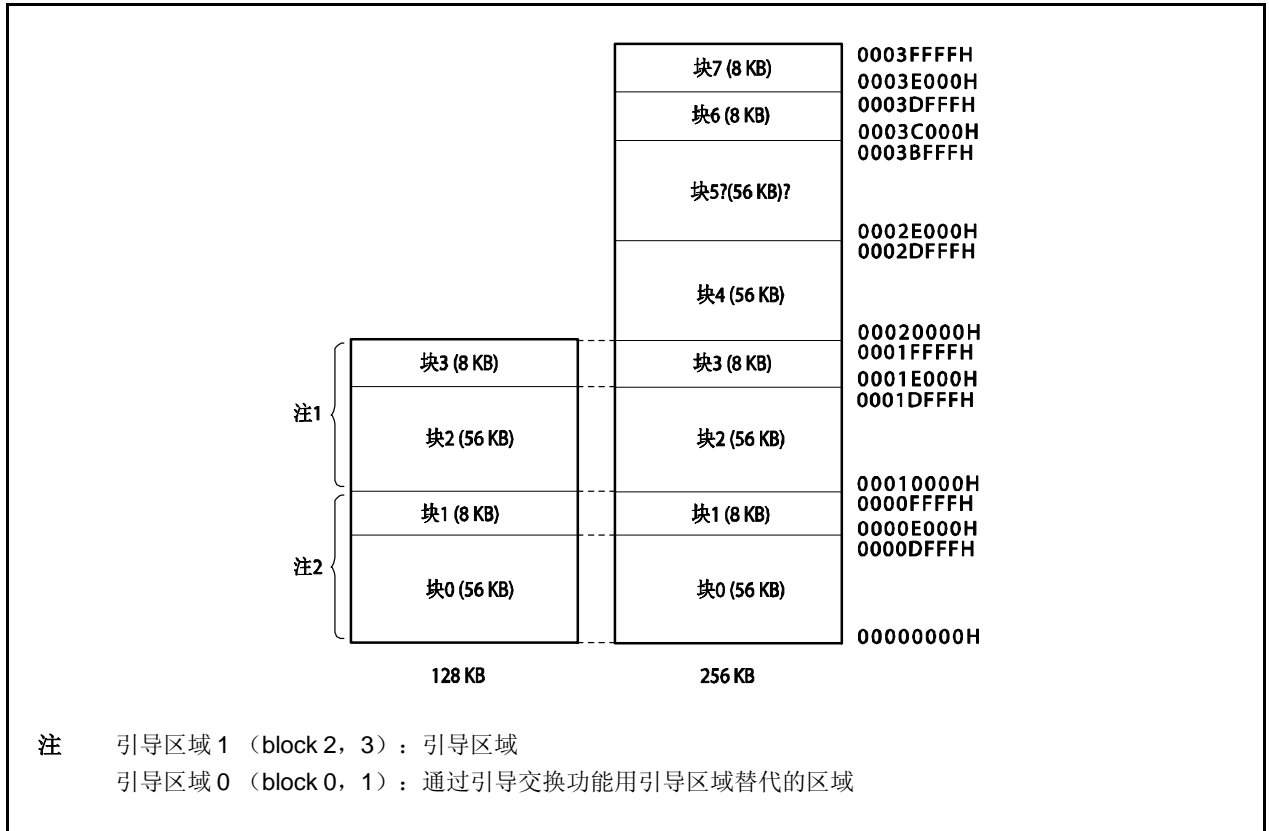
<R>

23.2 存储器配置

256/128 KB 内部 Flash 存储器区被分为 8/4 个 block，可以 block 为单位编程/擦除。也可以一次擦除所有的 block。

当使用引导交换功能时，位于引导区域 0 地址的物理内存（block0，1）被引导区域 1 地址的物理内存（block2，3）替代。关于引导交换功能的更多细节，敬请参阅 23.5 通过自编程重写。

图 23-1. Flash 存储器映射



<R> 23.3 功能概述

不论 V850ES/HG2 是否已经被安装到目标系统（on-board/off-board 编程），可以用专门的 Flash 存储器编程器的重写功能重新写入 V850ES/HG2 的内部 Flash 存储器。

同时还支持禁止重写已经写入到内部 Flash 存储器的用户程序的安全功能，这样未经授权的人就无法改动程序了。

使用用户程序（自编程）进行重写的功能适用于假设程序在目标系统生产/运输后改变的情况。同时还支持重新安全地重写整个 Flash 存储器区域的引导交换功能。此外，在自编程过程中还支持中断服务，因此可以在各种条件下重写 Flash 存储器，如在与外部设备通信过程中。

表 23-1. 重写方法

重写方法	功能概述	操作模式
片上编程	通过使用专门的 Flash 存储器编程器，可以在设备安装到目标系统后重写 Flash 存储器。	Flash 存储器编程模式
片下编程	通过使用专门的 Flash 存储器编程器和专门的程序适配器板（FA 系列），可以在设备安装到目标系统前重写 Flash 存储器。	
自编程	可以执行事先通过片上/片下编程写入到 Flash 存储器的用户程序重写 Flash 存储器。（在自编程的过程中，无法从内部 Flash 存储器区域取出指令或无法访问内部 Flash 存储器的数据。因此，必须提前将重写程序传输到内部 RAM 或外部存储器）。	正常操作方法

备注 FA系列是Naito Densei Machida Mfg. Co., Ltd.的产品。

表 23-2. 基本功能

功能	功能概述	支持 (√: 支持, ×: 不支持)	
		on-board/off-board 编程	自编程
block 擦除	擦除指定内存 block 的内容。	√	√
片擦除	一次擦除整个内存 block 的内容。	√	×
写入	写入到指定的地址并检查写入级是否可靠。	√	√
校验/校验和	将从内存读取的数据与 Flash 存储器编程器传来的数据比较。	√	× (无法通过用户程序读取)
空白检查	检查整个内存的擦除状态。	√	√
安全设置	可以禁止使用 block 擦除命令、片擦除命令、编程命令、读取命令和重写引导区域。	√	× (只有设置从启用改为禁用时才支持)

下表列出了安全功能。在出厂时默认启用 block 擦除命令禁止、片擦除命令禁止和编程命令禁止，可以通过片上/片下编程重写来设置安全。每个安全功能可以同时与其他安全功能组合使用。

表 23-3. 安全设置

功能	功能概述
block 擦除命令禁止	禁止在所有的 block 上执行 block 擦除命令。禁止的设置可以通过执行一个片擦除命令初始化。
片擦除命令禁止	禁止在所有的 block 上执行片擦除命令。 在设置禁止后，由于不能执行片擦除命令，禁止的设置无法初始化。
编辑命令禁止	禁止在所有的 block 上执行程序 and block 擦除命令。禁止的设置可以通过执行一个片擦除命令初始化。
读取命令禁止	禁止在所有的 block 上执行读取命令。禁止的设置可以通过执行一个片擦除命令初始化。
引导区域重写禁止	不支持。

表 23-4. 安全设置

功能	当设置每个安全时的擦除、读、写操作 (√: 可执行, ×: 不可执行, -: 不支持)		关于安全设定的说明	
	片上/片下编程	自编程	on-board/off-board 编程	自编程
block 擦除命令禁止	block 擦除命令: × 片擦除命令: √ 编程命令: √ 读取命令: √	block 擦除 (FlashBlockErase): √ 片擦除: - 写 (FlashWordWrite): √ 读 (FlashWordRead): √	可以用片擦除命令初始化禁止的设置。	仅支持设置从允许到禁止时
片擦除命令禁止	block 擦除命令: × 片擦除命令: × 编程命令: √ [#] 读取命令: √	block 擦除 (FlashBlockErase): √ 片擦除: - 写 (FlashWordWrite): √ 读 (FlashWordRead): √	无法初始化禁止的设置。	
编程命令禁止	block 擦除命令: × 片擦除命令: √ 编程命令: × 读取命令: √	block 擦除 (FlashBlockErase): √ 片擦除: - 写 (FlashWordWrite): √ 读 (FlashWordRead): √	可以用片擦除命令初始化禁止的设置。	
读取命令禁止	block 擦除命令: √ 片擦除命令: √ 编程命令: √ 读取命令: ×	block 擦除 (FlashBlockErase): √ 片擦除: - 写 (FlashWordWrite): √ 读 (FlashWordRead): √	可以用片擦除命令初始化禁止的设置。	

注 在此情况下，由于擦除命令是无效的，无法写入与已经写入到 Flash 存储器的数据不同的数据。

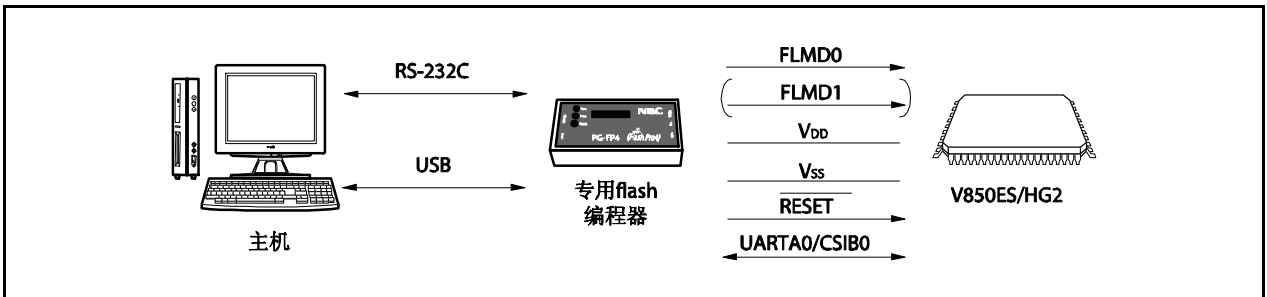
23.4 通过专门的Flash 存储器编程器重写

在安装 V850ES/HG2 到目标系统后用专门的 Flash 存储器编程器重写 Flash 存储器（on-board 编程）。也可以在安装设备到目标系统前（off-board 编程）用专门的编程适配器（FA 系列）重写 Flash 存储器。

23.4.1 编程环境

下面显示的是写入程序到 V850ES/HG2 的 Flash 存储器所要求的环境。

图 23-2. 写入程序到 Flash 存储器所要求的环境



要求用一台主机控制专门的 Flash 存储器编程器。

UARTA0 或 CSIB0 用于专门的 Flash 存储器编程器和执行写入、擦除等操作的 V850ES/HG2 之间的接口。要进行片下写入，要求专门的编程适配器（FA 系列）。

- FA-70F3707GC-8EA-MX（已经配线）
- FA-100GC-8EU-A（没有配线：要求配线）

备注 FA 系列是電盛町田制造有限公司的产品。

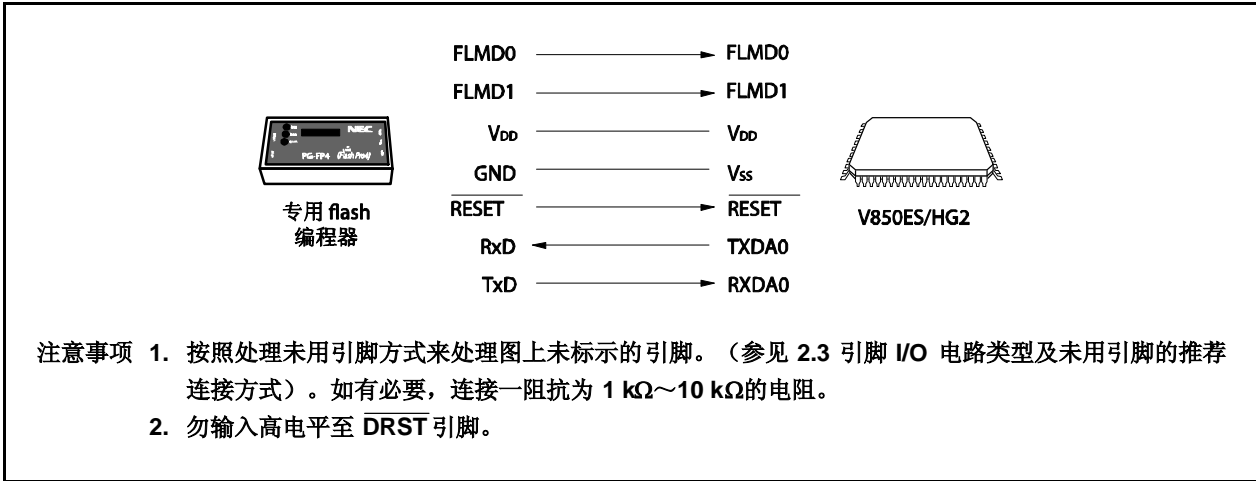
23.4.2 通信模式

专门的 Flash 存储器编程器和 V850ES/HG2 之间的通信是通过 V850ES/HG2 的 UARTA0 或 CSIB0 接口的串行通信进行的。

<R> (1) UARTA0

传输速率： 9, 600, 19, 200, 31, 250, 38, 400, 76, 800, 153, 600 bps
 (不支持 57, 600, 115, 200, 和 128, 000 bps。)

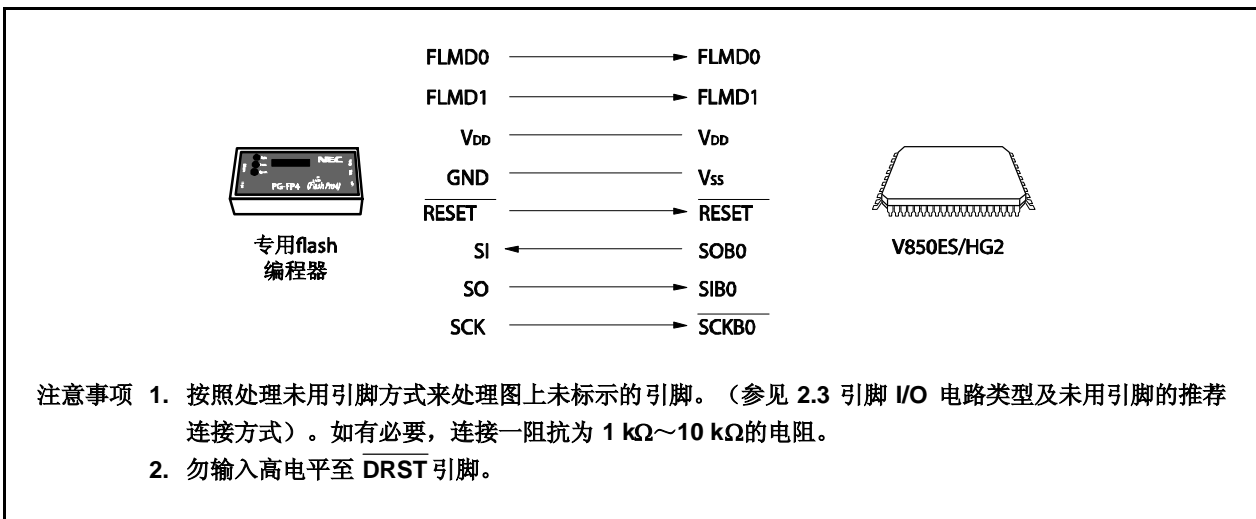
图 23-3. 与专门的 Flash 存储器编程器 (UARTA0) 通信



(2) CSIB0

串行时钟： 2.4 kHz~ 2.5 MHz (最高有效位优先)

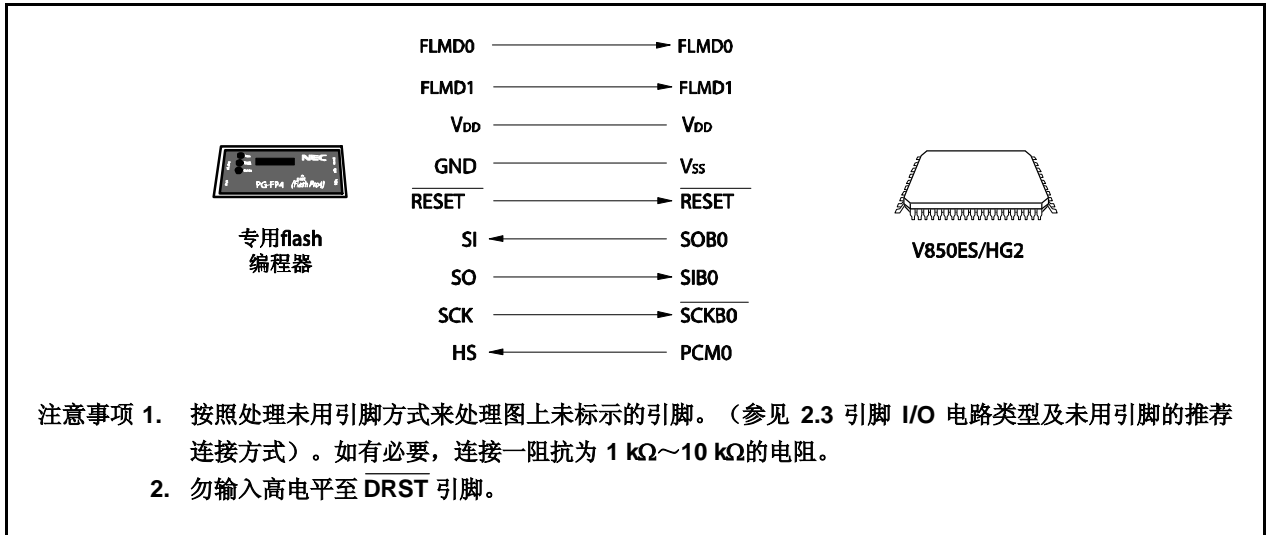
图 23-4. 与专用 Flash 存储器编程器 (CSIB0) 通信



(3) CSIB0 + HS

串行时钟： 2.4 kHz ~2.5 MHz（最高有效位优先）

图 23-5. 与专用 Flash 存储器编程器（CSIB0 + HS）通信



专门的 Flash 存储器编程器输出传输时钟，V850ES/HG2 作为从设备运行。

当 PG-FP4 用作专门的 Flash 存储器编程器时，向 V850ES/HG2 生成以下信号。关于更多细节，敬请参阅 PG-FP4 用户手册（U15260E）。

表 23-5. 专门的 Flash 存储器编程器（PG-FP4）的信号连接

PG-FP4			V850ES/HG2	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	写入启用/禁止	FLMD0	○	○	○
FLMD1	输出	写入启用/禁止	FLMD1	○ ^{注1}	○ ^{注1}	○ ^{注1}
VDD	-	V _{DD} 电压生成/电压监控	V _{DD}	○	○	○
GND	-	接地	V _{SS}	○	○	○
CLK	输出	输出到 V850ES/HG2	X1, X2	× ^{注2}	× ^{注2}	× ^{注2}
RESET	输出	复位信号	RESET	○	○	○
SI/RxD	输入	接收信号	SOB0, TXDA0	○	○	○
SO/TxD	输出	发送信号	SIB0, RXDA0	○	○	○
SCK	输出	传输时钟	SCKB0	×	○	○
HS	输入	CSIB0 + HS 通信的握手信号	PCMO	×	×	○

- 注 1. 连接图 23-6 所示的引脚，或然后通过板上下拉电阻连接至 GND。
 2. 无法通过 flash 编程器的 CLK 引脚提供时钟。在板上创建振荡器来提供时钟。

备注 ○: 必须连接。
 ×: 不需要连接。

表 23-6. 连接 V850ES/HG2 (FA-100GC-8EU-A) 系列的 Flash 写入适配器

Flash 编程器 (PG-FP4) 的连接 引脚			FA 板上的 引脚名称	当使用 CSIB0 + HS 时		当使用 CSIB0 时		当使用 UARTA0 时	
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	接收信号	SI	P41/SOB0	23	P41/SOB0	23	P30/TXDA0	25
SO/TxD	输出	发送信号	SO	P40/SIB0	22	P40/SIB0	22	P31/RXDA0/INTP7	26
SCK	输出	传输时钟	SCK	P42/SCKB0	24	P42/SCKB0	24	不需要	-
CLK	输出	设置时钟到 V850ES/HG2	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	写入电压	FLMD0	FLMD0	8	FLMD0	8	FLMD0	8
FLMD1	输出	写入电压	FLMD1	PDL5/FLMD1	76	PDL5/FLMD1	76	PDL5/FLMD1	76
HS	输入	CSI0 + HS 通 信的握手信号	RESERVE/ HS	PCM0	61	不需要	-	不需要	-
VDD	-	V _{DD} 电压生成/电 压监控	VDD	V _{DD}	9	V _{DD}	9	V _{DD}	9
				BV _{DD}	70	BV _{DD}	70	BV _{DD}	70
				EV _{DD}	5, 34	EV _{DD}	5, 34	EV _{DD}	5, 34
				AV _{REF0}	1	AV _{REF0}	1	AV _{REF0}	1
GND	-	接地	GND	V _{SS}	11	V _{SS}	11	V _{SS}	11
				AV _{SS}	2	AV _{SS}	2	AV _{SS}	2
				BV _{SS}	69	BV _{SS}	69	BV _{SS}	69
				EV _{SS}	33	EV _{SS}	33	EV _{SS}	33

- 注意事项 1. 务必通过一值为 4.7 μ F (推荐值) 的电容器来将 REGC 引脚连接至 GND。
2. 无法通过 flash 编程器的 CLK 引脚提供时钟。在板上创建振荡器来提供时钟。

图 23-6. V850ES/HG2Flash 存储器写入适配器 (FA-100GC-8EU-A) 的接线示例
(CSIB0 + HS 模式) (1/2)

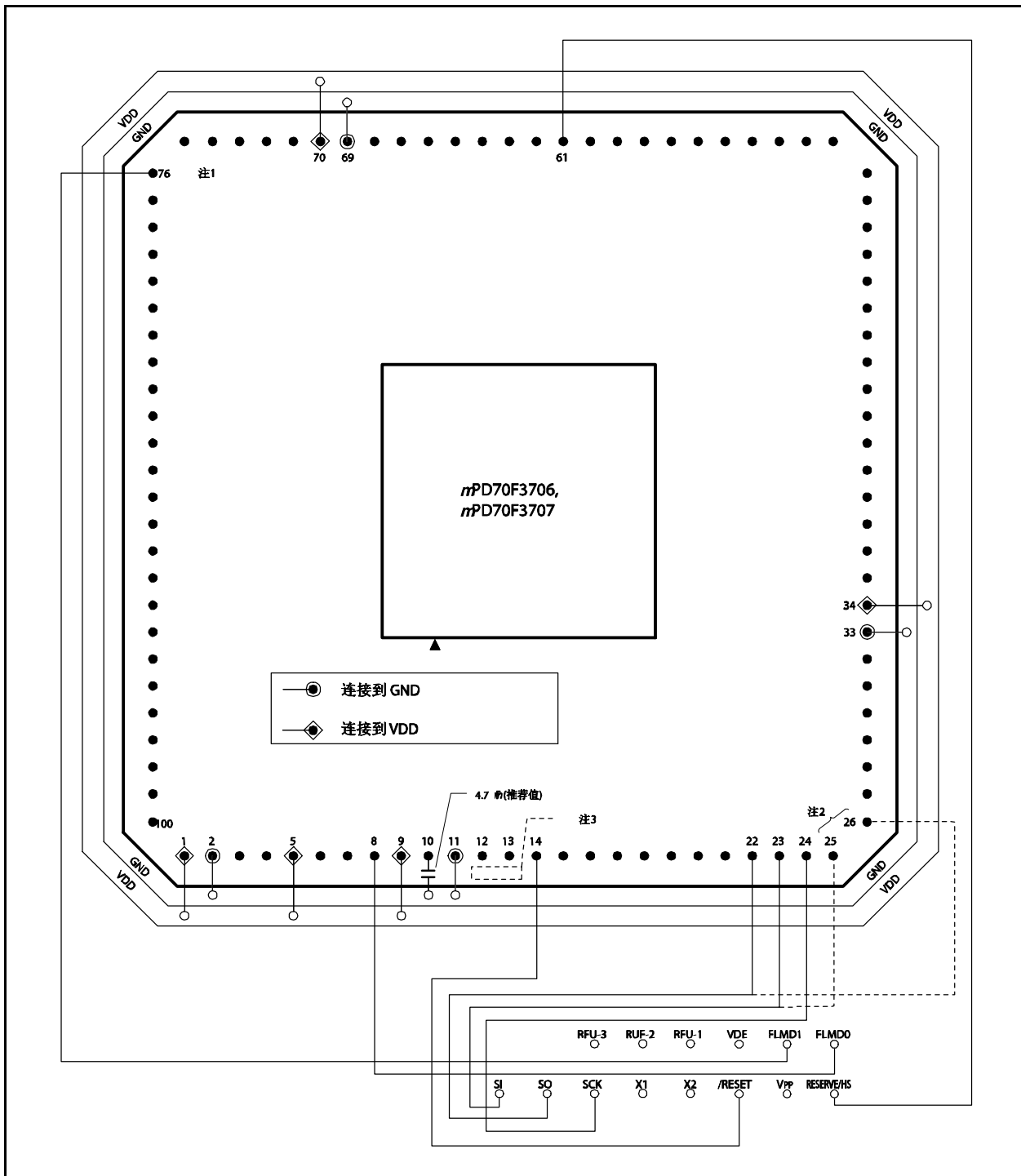
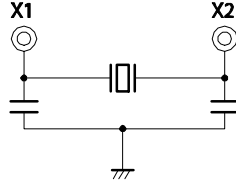


图 23-6. V850ES/HG2 Flash 存储器写入适配器 (FA-100GC-8EU-A) 的接线示例
(CSIB0 + HS 模式) (2/2)

- 注
1. 如图所示连接 FLMD1 引脚，或者通过下拉电阻连接该引脚到板上的 GND。
 2. 当使用 UAR0A0 时，启用引脚。
 3. 通过在 flash 存储器写入适配器（在虚线内）上创建一个振荡器提供一个时钟。以下是振荡器的一个实例。

实例



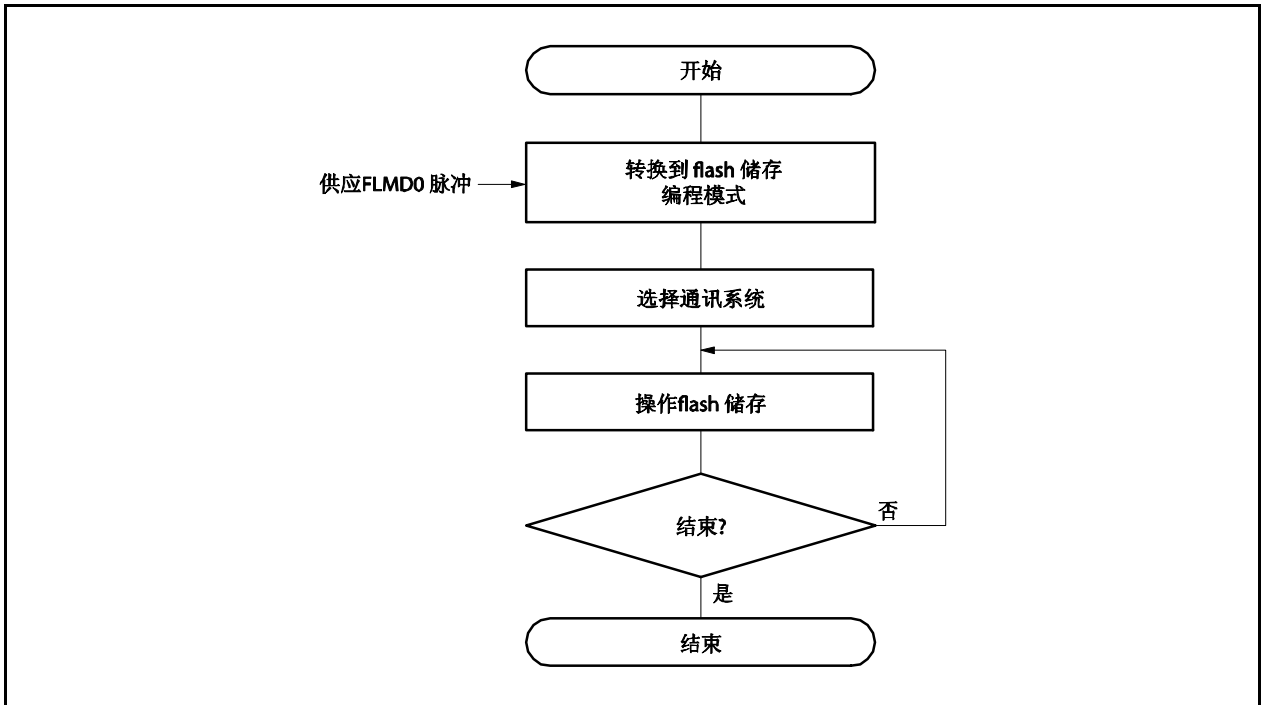
注意事项 勿输入高电平至 $\overline{\text{DRST}}$ 引脚。

- 备注
1. 按照处理未用引脚方式来处理图上未标示的引脚。（参见 2.3 引脚 I/O 电路类型及未用引脚的推荐连接方式）。
 2. 该适配器用于 100 引脚塑料 LQFP 包。

23.4.3 Flash 存储器控制

下图显示了 Flash 存储器控制的步骤。

图 23-7. Flash 存储器控制的步骤

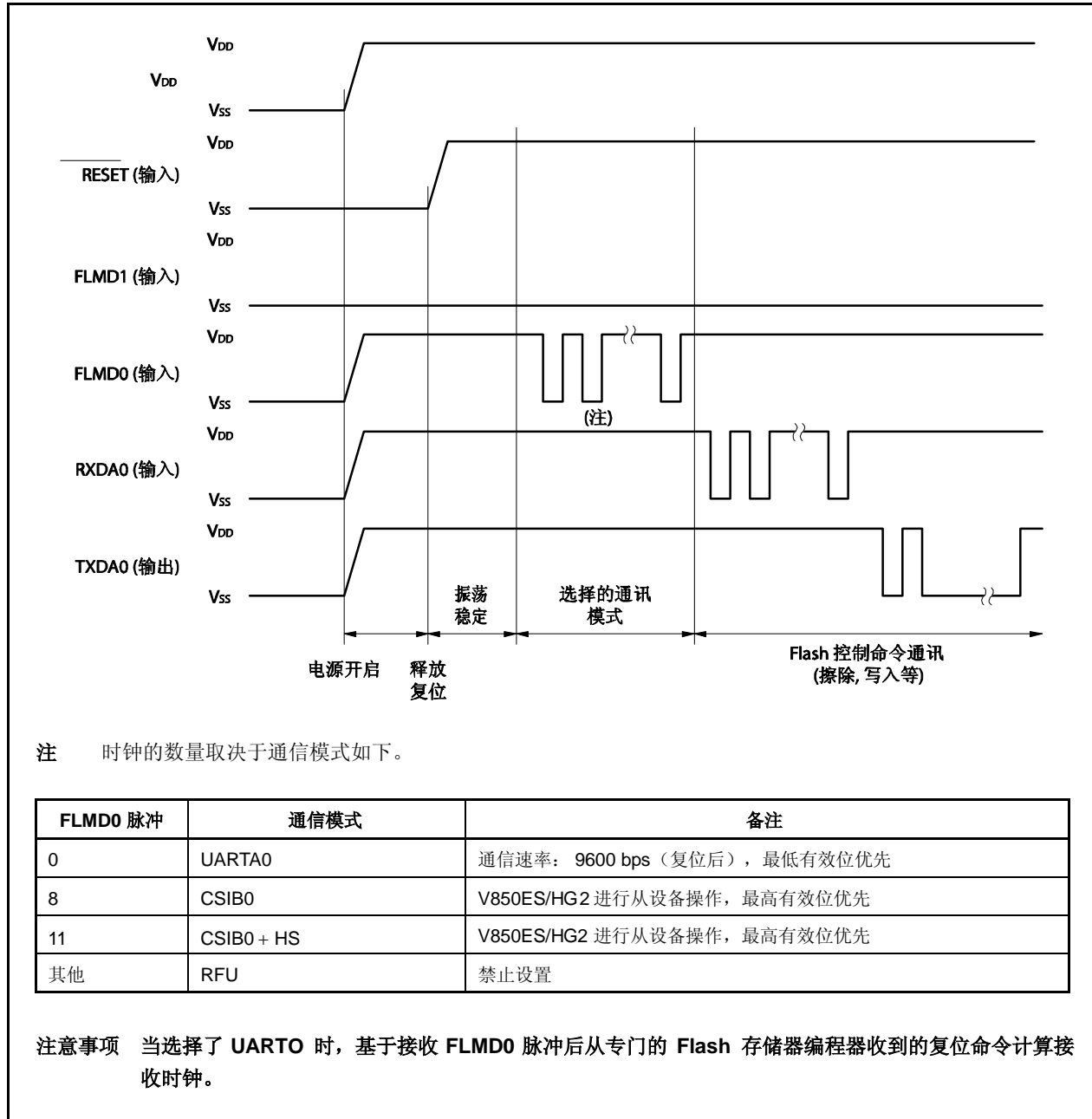


23.4.4 通信模式的选择

在 V850ES/HG2 中，通信模式是通过切换到 Flash 存储器编程模式后输入脉冲（最多可输入 11 个脉冲）到 FLMD0 引脚。FLMD0 脉冲可以由专门的 Flash 存储器编程器生成。

下图显示的是脉冲的次数和通信模式之间的关系。

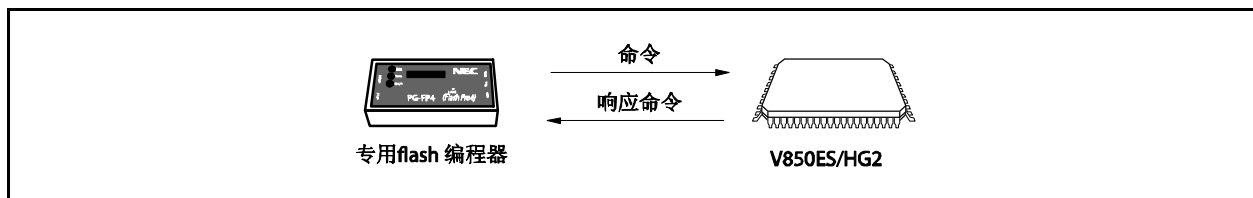
图 23-8. 通信模式的选择



23.4.5 通信命令

V850ES/HG2 通过命令的方式与专门的 Flash 存储器编程器通信。从专门的 Flash 存储器编程器发往 V850ES/HG2 的信号称为“命令”。从 V850ES/HG2 发出到专门的 Flash 存储器编程器的响应信号称为“响应命令”。

图 23-9. 通信命令



下表中显示了 V850ES/HG2 中的 Flash 存储器控制命令。所有的这些命令从专门的 Flash 存储器编程器发出，V850ES/HG2 根据命令进行处理。

表 23-7. Flash 存储器控制命令

分类	命令名称	支持			功能
		CSIB0	CSIB0 + HS	UARTA0	
空白检查	block 空白检查命令	√	√	√	检查指定 block 中的内容是否被正确擦除。
擦除	片擦除命令	√	√	√	擦除整个内存的内容。
	block 擦除命令	√	√	√	擦除指定内存 block 的内容。
写入	编程命令	√	√	√	写入到指定的地址范围，执行检查的内容。
校验	校验命令	√	√	√	比较指定地址范围中的内存内容与从 Flash 存储器编程器中传输来的数据。
	检查和命令	√	√	√	读取指定地址范围中的检查和。
系统设置, 控制	硅签名命令	√	√	√	读取硅签名信息。
	安全设置命令	√	√	√	禁用片擦除命令, block 擦除命令, 编程命令, 及读取命令。

23.4.6 引脚连接

当进行片上写入时，将连接器安装到目标系统以连接到专门的 Flash 存储器编程器。同时，在板上提供从正常模式切换到 Flash 存储器编程模式。

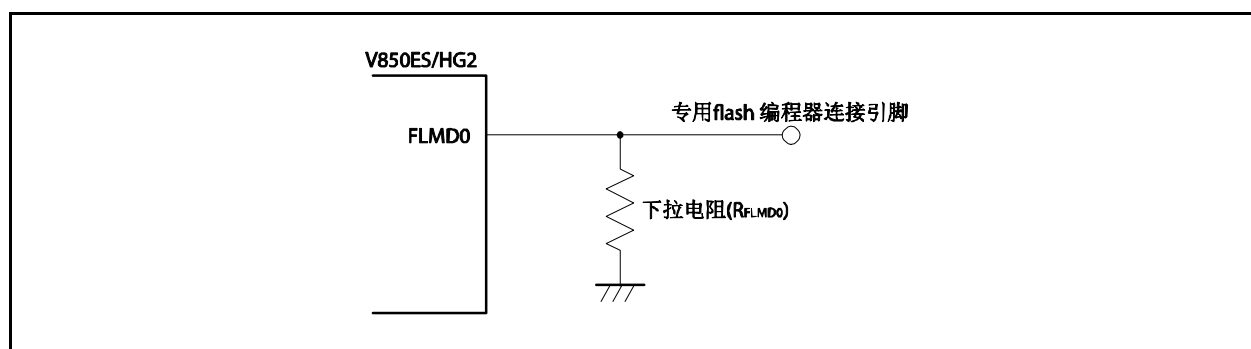
在 Flash 存储器编程模式中，所有的不用于 Flash 存储器编程的引脚变成与复位后相同的状态。因此，当外部设备在复位后没有立即确认状态时应要求引脚处理。

(1) FLMD0 引脚

在正常运行模式下，输入电压 V_{SS} 到 FLMD0 引脚。在 Flash 存储器编程模式下，向 FLMD0 引脚提供一个写入电压 V_{DD} 。

由于 FLMD0 引脚作为自编程模式下的写保护引脚，在写入到 Flash 存储器前必须通过端口控制向 FLMD0 引脚提供电压 V_{DD} 。关于更多细节，敬请参阅 23.5.5 (1) FLMD0 引脚。

图 23-10. FLMD0 引脚连接示例



(2) FLMD1 引脚

当输入到 FLMD0 引脚的电压为 0V 时，FLMD1 引脚不工作。当向 FLMD0 引脚提供 V_{DD} 电压时，进入 Flash 存储器编程模式，因此必须向 FLMD1 引脚输入 0V 电压。以下显示的是 FLMD1 引脚连接的示例。

图 23-11. FLMD1 引脚连接示例

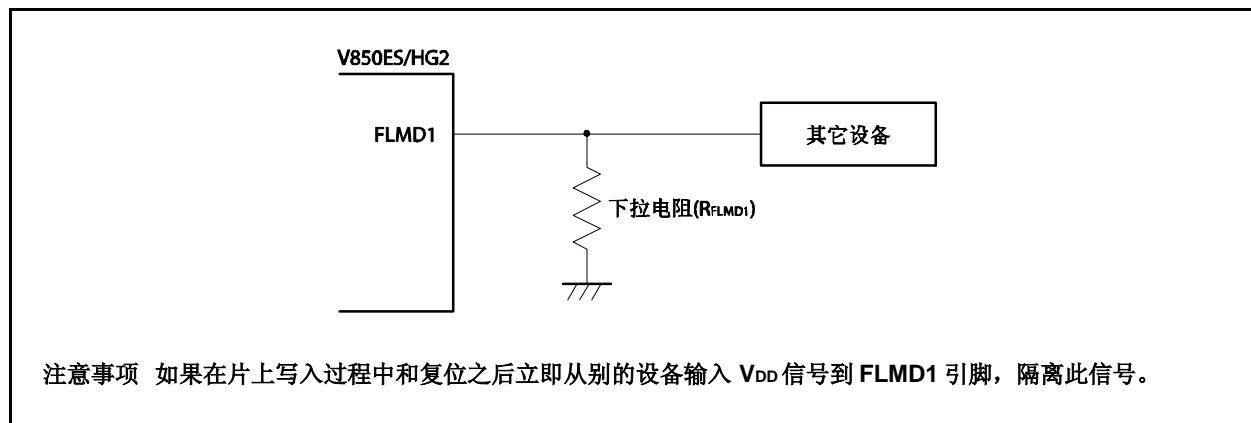


表 23-8. 当释放复位时 FLMD0 和 FLMD1 引脚与操作模式之间的关系

FLMD0	FLMD1	操作模式
0	忽略	正常操作模式
V _{DD}	0	Flash 存储器编程模式
V _{DD}	V _{DD}	禁止设置

(3) 串行接口引脚

下表显示的是每个串行接口使用的引脚。

表 23-9. 串行接口使用的引脚

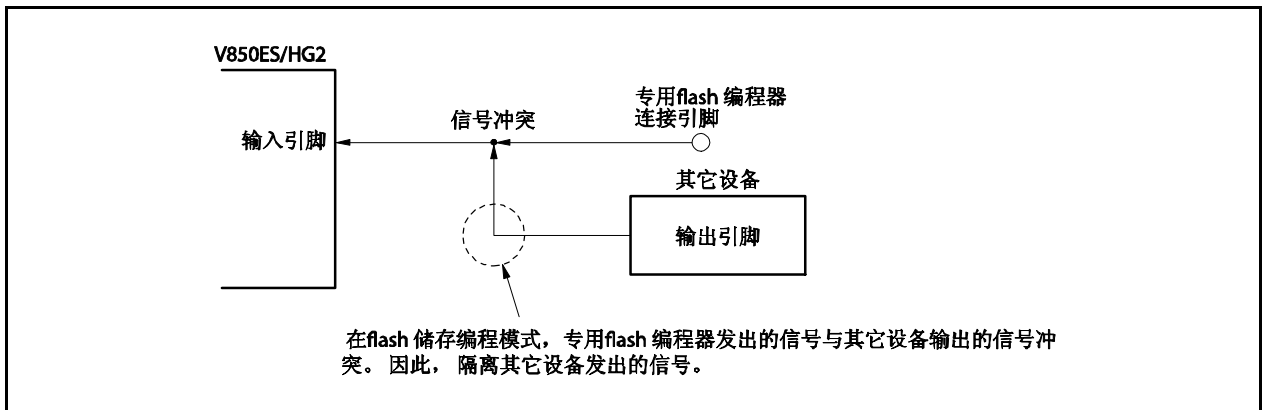
串行接口	使用的引脚
UARTA0	TXDA0, RXDA0
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, PCM0

当连接专门的 Flash 存储器编程器到与另一个板上设备连接的串行接口引脚时，应小心避免信号的冲突和另一个设备的故障。

(a) 信号的冲突

当连接专门的 Flash 存储器编程器（输出）到与另一个板上设备（输出）连接的串行接口引脚（输入）时，会出现信号冲突。要避免信号冲突，应隔离到其他设备的连接或设置其他设备为输出高阻抗状态。

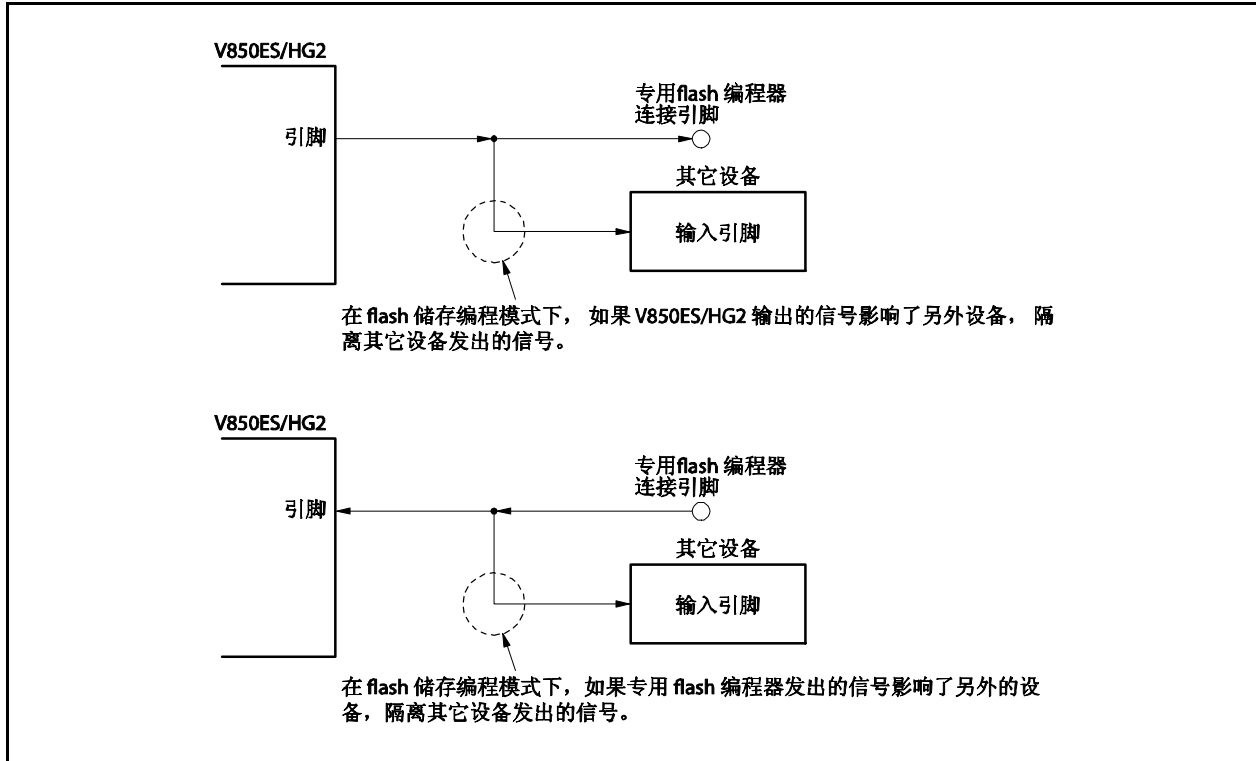
图 23-12. 信号冲突（串行接口输入引脚）



(b) 其他设备故障

当连接专门的 Flash 存储器编程器（输出或输入）到与另一个板上设备（输入）连接的串行接口针脚（输入或输出）时，信号输出到其他设备，造成设备发生故障。要避免这种情况，应隔离到其他设备的连接。

图 23-13. 其他设备的故障

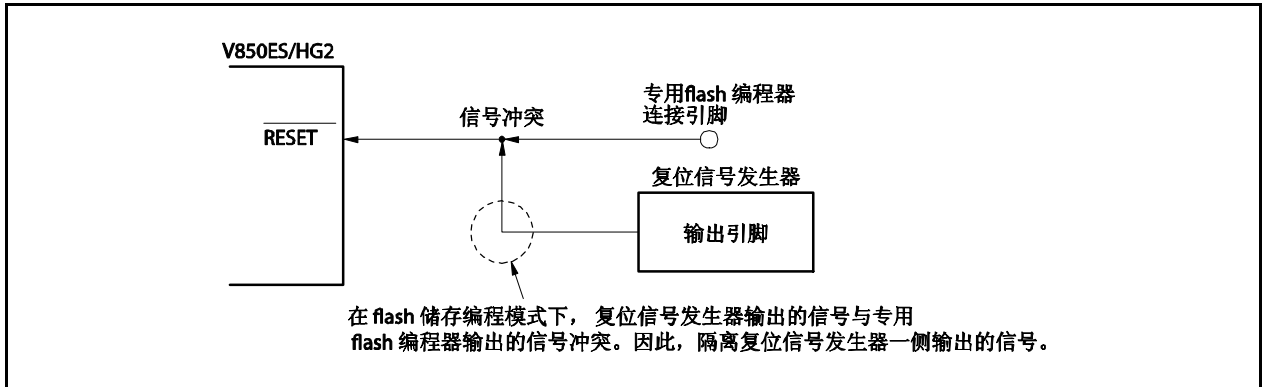


(4) RESET 引脚

当专门的 Flash 存储器编程器的复位信号连接到与板上复位信号生成器的 RESET 引脚时，会发生信号冲突。要避免信号冲突，必须隔离到复位信号生成器的连接。

当在 Flash 存储器编程模式下从用户系统输入复位信号时，将无法正确地进行编程操作。因此，不要输入来自专门的 Flash 存储器编程器的复位信号以外的信号。

图 23-14. 信号的冲突 (RESET 引脚)

**(5) 端口引脚 (包括 NMI)**

当系统切换到 Flash 存储器编程模式时，所有不用于 Flash 存储器编程的引脚与复位后的状态相同。如果与此端口连接的外部设备不识别复位后端口的状态，需要对引脚进行适当的处理，如通过一个电阻连接到 VDD 或者 VSS。

(6) 其他信号引脚

与正常模式下的状态一样的连接 X1, X2, XT1, 和 XT2。

在 flash 存储器编程中，输入低电平至 DRST 引脚或使其保持开放状态。勿输入高电平。

(7) 电源

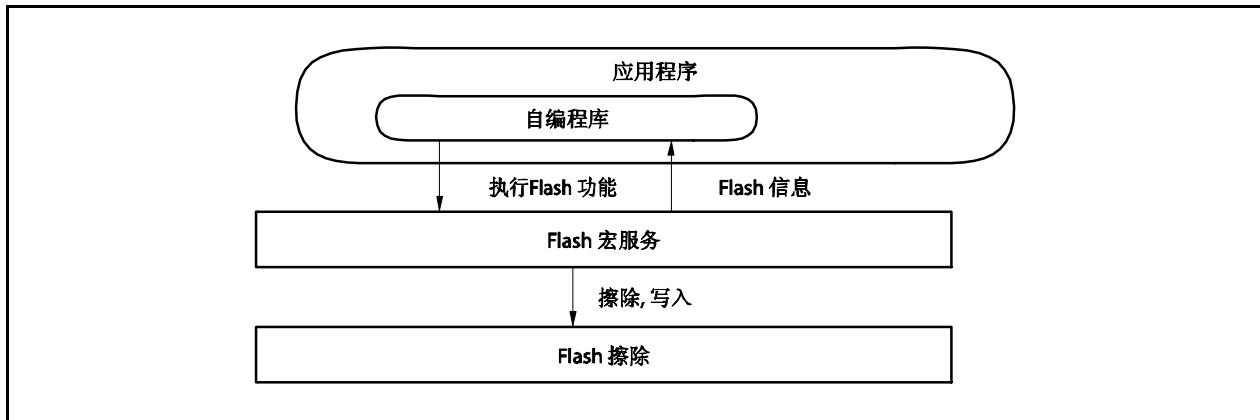
提供与正常操作模式下一样的电源 (VDD, VSS, EVDD, EVSS, BVDD, BVSS, AVREF0, AVSS, REGC)。

23.5 通过自编程重写

23.5.1 概述

V850ES/HG2 支持允许用户程序自己重写内部 Flash 存储器的 Flash 存储器宏服务。通过使用此接口和用于使用用户应用程序重写 Flash 存储器的自编程库，可以通过事先传输到内部 RAM 或外部存储器的用于应用程序重写 Flash 存储器。因此可以升级用户程序和不断重写数据。

图 23-15. 自编程的概念

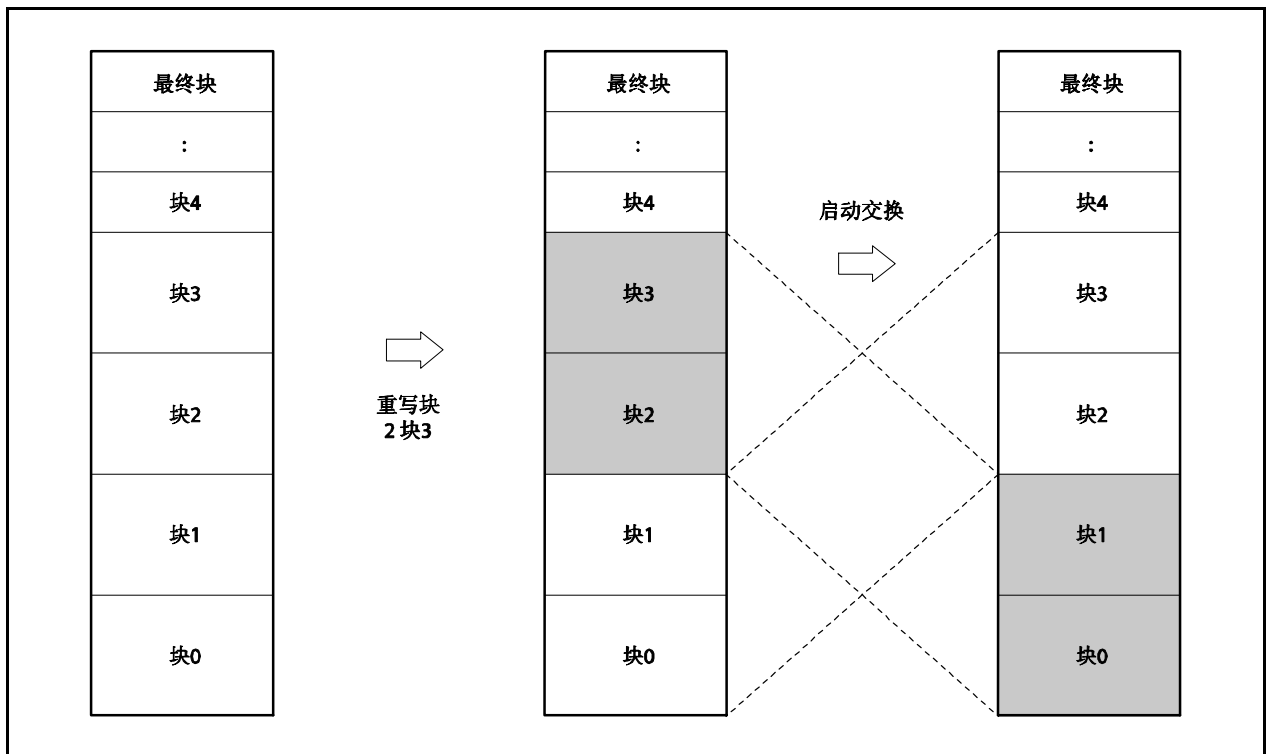


23.5.2 特点

(1) 安全的自编程（引导交换功能）

V850ES/HG2 支持引导交换功能，即将引导区域 block0 及 block1 的物理内存与引导区域 block2 及 block3 的物理内存互换。通过事先写入要被重写到引导区域 block2 及 3 中的启动程序然后交换物理内存，即使在重写过程中出现断电也可以安全地重写整个区域，因为正确的用户程序始终存在于引导区域 block0 及 block1。

图 23-16. 重写整个内存区域（启动交换）



(2) 中断支持

在自编程过程中不能从 Flash 存储器获取指令。按照常规做法，即使出现中断也不能使用写入到 Flash 存储器的用户程序。

因此在 V850ES/HG2 中，如果要在自编程过程中使用中断，必须在内部 RAM 中处理到指定地址*的转移。将转移处理的转移指令分配到内部 RAM 中指定地址**的用户中断服务。

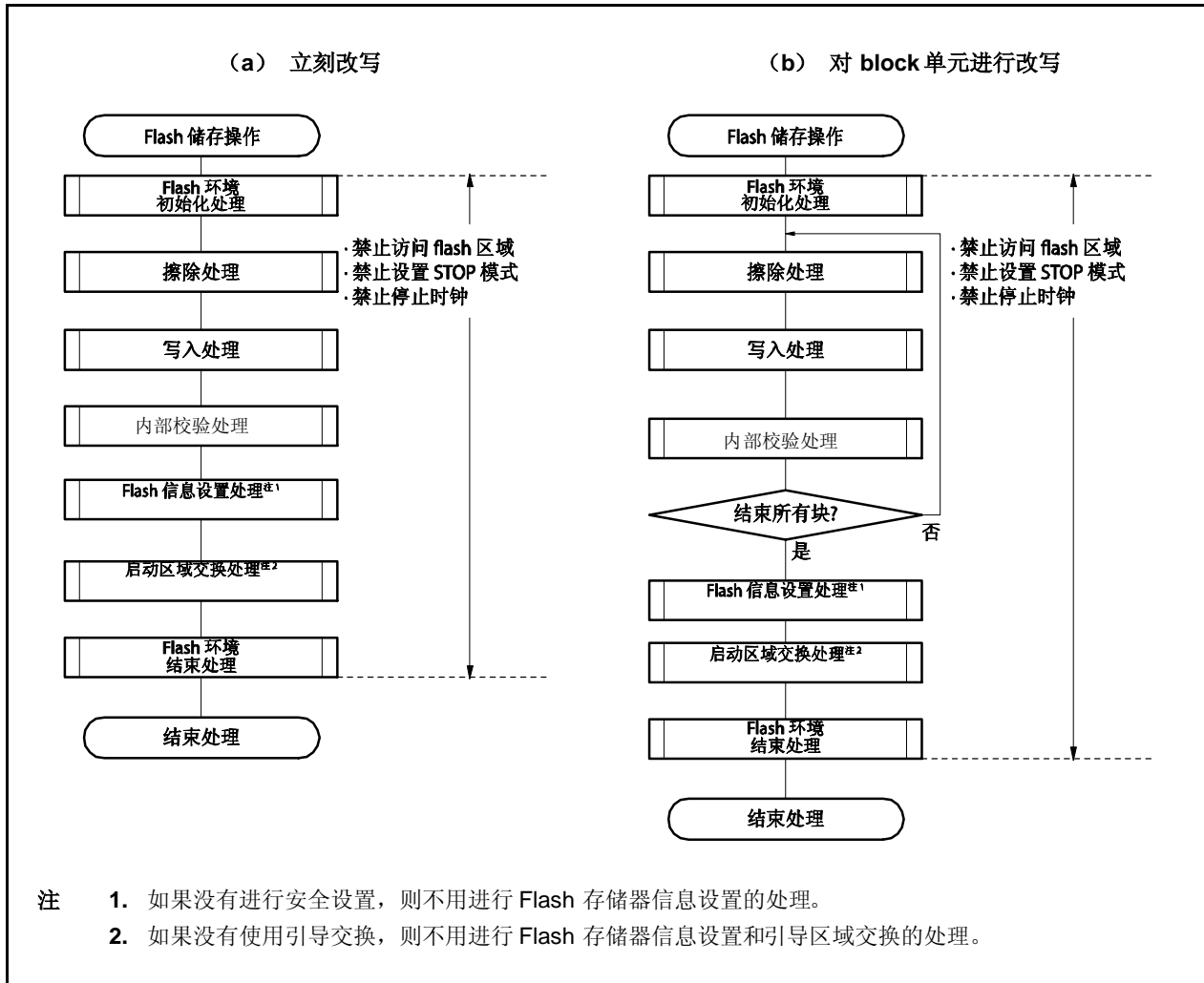
注 NMI 中断: 内部 RAM 的开始地址
 可屏蔽的中断: 内部 RAM 的开始地址+ 4 个地址

23.5.3 标准自编程流程

通过 Flash 存储器自编程重写 Flash 存储器的整个程序如下所示。

<R>

图 23-17. 标准自编程流程



23.5.4 Flash 存储器功能

表 23-10. Flash 存储器功能清单

功能名称	概述	支持
FlashEnv	Flash 存储器控制宏的初始化	√
FlashBlockErase	只擦除指定的一个 block	√
FlashWordWrite	从指定的地址写入	√
FlashBlockVerify	指定一个 block 的内部核查	√
FlashBlockBlankCheck	指定一个 block 的空白检查	√
FlashFLMDCheck	检查 FLMD 引脚	√
FlashStatusCheck	对之前的操作状态进行检查	√
FlashGetInfo	阅读 Flash 存储器信息	√
FlashSetInfo	设置 Flash 存储器信息	√
FlashBootSwap	引导区域交换	√
FlashWordRead	从指定的地址读取数据	√
FlashSetUserHandler	用户中断处理寄存功能	√

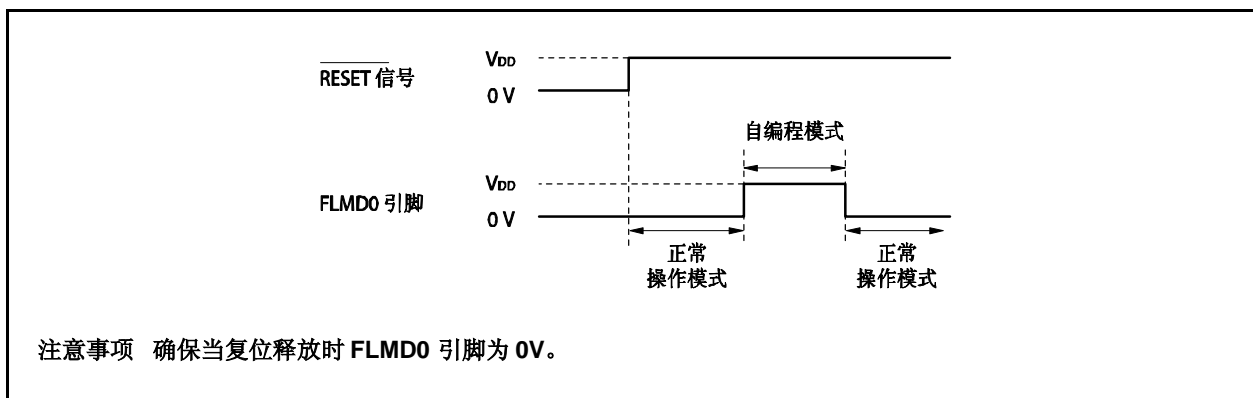
23.5.5 引脚处理

(1) FLMD0 引脚

FLMD0 引脚用于在释放复位时设置操作模式，从而保护 Flash 存储器在自重写过程中不被写入。因此必须在释放复位并执行正常操作时保持供到 FLMD0 引脚的电压为 0V。在重写内存前通过引脚控制进行自编程过程中，必须向 FLMD0 引脚供应 V_{DD} 的电压。

当自编程完成时，FLMD0 引脚的电压必须返回 0V。

图 23-18. 模式改变时序



23.5.6 使用的内部资源

下表列出了用于自编程的内部资源。这些内部资源也可以自由地用于自编程以外的目的。

表 23-11. 使用的内部资源

资源名称	描述
堆栈区 (用户堆栈区 + 300 字节)	用户使用的堆栈区的延长区可被库使用 (可在内部 RAM 及外部 RAM 内使用)。
<R> 库代码 (约 2, 500 字节)	库程序实体 (可以在要操作的 Flash 存储器 block 以外的任何地方使用)。
应用程序	作为用户应用程序执行。 调用 Flash 存储器函数。
可屏蔽的中断	可以在用户应用程序执行状态或自编程状态中使用。要在自编程状态中使用此中断, 由于处理将转移到内部 RAM 开始地址 +4 个地址的地址, 必须事先将转移处理的转移指令分配到内部 RAM 开始地址+4 个地址的用户中断服务。
NMI 中断	可以在用户应用程序执行状态或自编程状态中使用。要在自编程状态中使用此中断, 由于处理将转移到内部 RAM 开始地址, 必须事先将转移处理的转移指令分配到内部 RAM 开始地址的用户中断服务。

第二十四章 选项字节功能

选项字节作为 8 位数据存贮在内部 flash 的地址 000007AH 处（内部 ROM 区域）。

当将程序写入 V850ES/HG2 时，确保在地址 000007AH 处设置与程序中以下选项对应的选项数据作为缺省数据。

在程序执行过程中，此区域的数据不能进行重写。

地址: 000007AH							
7	6	5	4	3	2	1	0
OPB7	OPB6	-	-	-	-	OPB1	OPB0
OPB7	OPB6	副时钟操作模式的设定					
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 模式的设定						
0	可选择操作时钟 (f_c/f_R) 可选择INTWDT2 模式/WDTRES模式						
1	固定为内部振荡器时钟 (f_R) 固定为 WDTRES 模式						
OPB0	允许/禁止停止内部振荡器						
0	允许停止						
1	禁止停止						

使用 CA850 的示例程序显示如下。

[示例程序]

```
#-----  
# 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 确保在此扇区中写入 6 字节，如果少于 6 字节，则在连接器运行时会发生错误。
错误消息: F4112: 非法“OPTION_BYTES”扇区大小。

备注 将地址 007BH ~ 007FH 设置为 0x00。

V850ES/HG2 系列的片上调试功能可通过以下两种方式来实现。

- 使用 DCU（调试控制单元）
片上调试功能可通过 V850ES/HG2 系列的片上 DCU 来实现，将 $\overline{\text{DRST}}$ ，DCK，DMS，DDI，和 DDO 引脚作为调试接口引脚。
- 不使用 DCU
片上调试功能可通过 MINICUBE2 或类似产品来实现，使用用户资源而非 DCU。

下表显示了两个片上调试功能的特性。

表 25-1. 片上调试功能特性

		使用 DCU 调试	不使用 DCU 调试
调试接口引脚		$\overline{\text{DRST}}$ ，DCK，DMS，DDI， DDO	<ul style="list-style-type: none"> • 使用 UARTA0 时 RXD0，TXD0 • 使用 CSIB0，SIB0，SOB0， SCKB0，HS（PCM0）时
保护用户资源		不支持	需要
硬件断点功能		2 点	2 点
软件断点功能	内部 ROM 区域	4 点	4 点
	内部 RAM 区域	2000 点	2000 点
实时 RAM 监视功能 ^{注 1}		可用	可用
动态存储器修改功能（DMM）功能 ^{注 2}		可用	可用
屏蔽功能		$\overline{\text{Reset}}$ ，NMI，INTWDT2	$\overline{\text{RESET}}$ 引脚
ROM 安全功能		10 字节 ID 码加密	10 字节 ID 码加密
使用的硬件		NINICUBE [®] ，等等	NINICUBE2，等等
追踪功能		不支持	不支持
调试中断接口功能（DBINT）		不支持	不支持

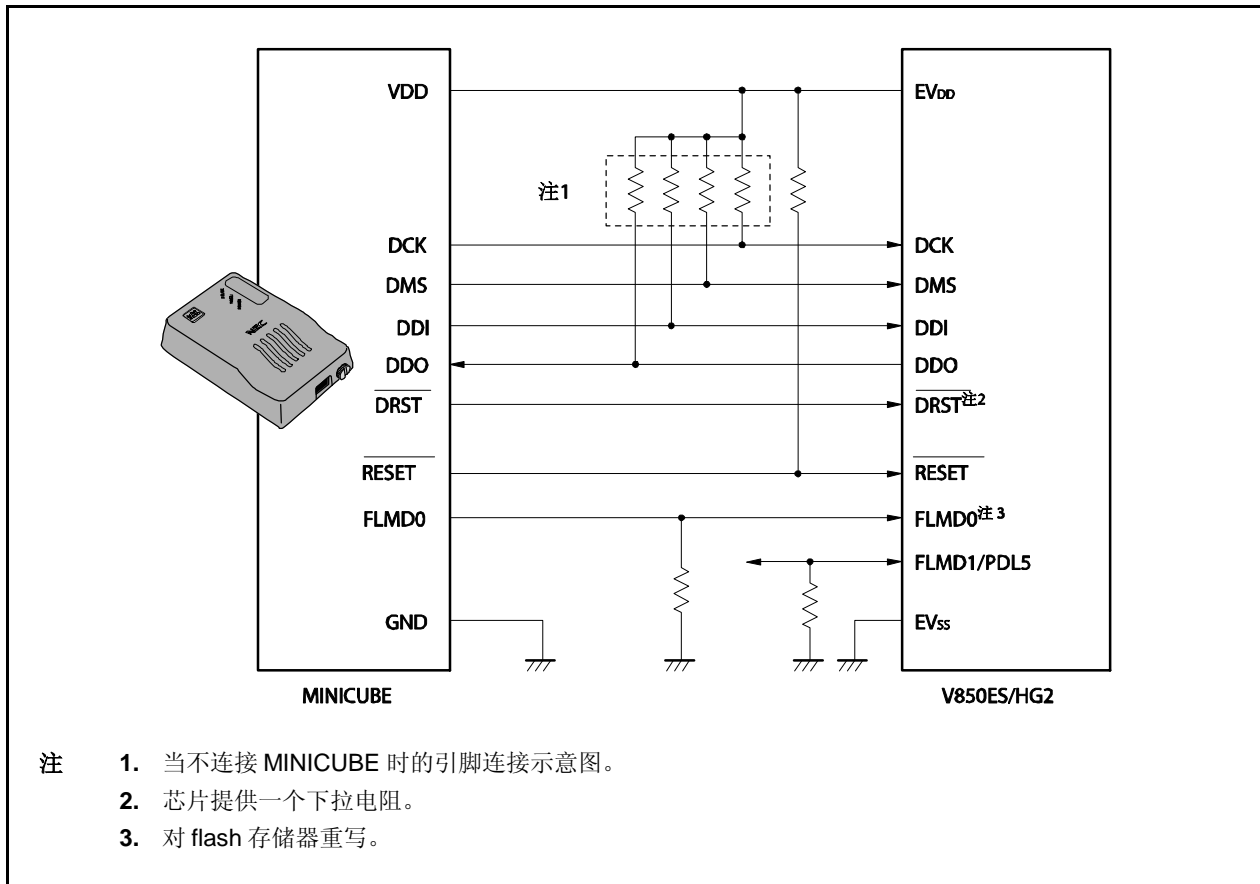
- 注
1. 此功能在程序执行时读出存储器内容。
 2. 此功能在程序执行时改写 RAM 的内容。

25.1 使用 DCU 调试

可通过将调试接口引脚 ($\overline{\text{DRST}}$, DCK, DMS, DDI, 及 DDO) 连接到片上调试仿真器 (MINICUBE) 来对程序进行调试。

25.1.1 电路连接示例

图 25-1. 当调试接口引脚用作通信接口的电路连接示意图



25.1.2 接口信号

接口信号说明如下。

(1) $\overline{\text{DRST}}$

对于片上调试单元这是一个复位输入信号。它是异步初始化调试控制单元的负逻辑信号。

在集成调试器启动后，当 MINICUBE 检测到目标系统的 V_{DD} 后拉高 $\overline{\text{DRST}}$ 信号，并且启动设备的片上调试单元。

当 $\overline{\text{DRST}}$ 信号为高时，CPU 也产生一个复位信号。

当通过启动集成调试器开始调试时，CPU 也产生复位信号。

(2) DCK

这是一个时钟输入信号。它由 MINICUBE 提供一个 20 MHz 的时钟。在片上调试单元，当 DCK 信号上升沿时 DMS 和 DDI 信号被采样，在下降沿时数据 DDO 被输出。

(3) DMS

这是发送模式选择信号。测试单元的发送状态根据 DMS 信号的电平改变。

(4) DDI

这是一个数据输入信号。DCK 上升沿时在片上调试单元采样。

(5) DDO

这是一个数据输出信号。DCK 信号下降沿时从片上调试单元输出。

(6) EV_{DD}

此信号用于检测目标系统的 VDD。如果目标系统的 VDD 没有被检测到，MINICUBE ($\overline{\text{DRST}}$, DCK, DMS, DDI, FLMD0, 和 RESET) 信号输出为高阻抗状态。

(7) FLMD0

flash 自编程功能用于通过集成调试器下载数据到 flash 存储器。在 flash 自编程期间，FLMD0 引脚必须保持为高。另外，将下拉电阻连接到 FLMD0 引脚。

FLMD0 引脚也可以用如下两种方式控制。

<1> 通过 MINICUBE 控制

连接 MINICUBE 的 FLMD0 信号到 FLMD0 引脚。

在正常模式下，MINICUBE 不产生任何驱动（高阻抗）。

中断期间，当集成调试器的下载功能执行时 MINICUBE 将 FLMD0 引脚置高电平。

<2> 通过端口控制

连接设备的任意端口到 FLMD0 引脚。

通过用户程序实现 flash 自编程功能任意端口都可以使用。

在集成调试器的控制台上，执行下载功能前设置端口引脚为高电平，或者执行完下载功能后置端口引脚为低。

更多细节，敬请参阅 ID850QB Ver. 3.10 集成调试器操作用户手册（U17435E）。

(8) $\overline{\text{RESET}}$

这是一个系统复位输入引脚。如果 $\overline{\text{DRST}}$ 引脚由用户程序设置的 OCDM 寄存器的 OCDM0 位的数值而无效，那么不能执行片上调试功能。因此，使用 $\overline{\text{RESET}}$ 引脚，通过 MINICUBE 实现复位，使 $\overline{\text{DRST}}$ 引脚有效（初始化）。

25.1.3 可屏蔽功能

复位，NMI，INTWDT2 信号可被屏蔽。

调试器（ID850QB）的可屏蔽功能以及相应的 V850ES/HG2 功能如下。

表 25-2. 可屏蔽功能

ID850QB 的可屏蔽功能	相应的 V850ES/HG2 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	通过输入 $\overline{\text{RESET}}$ 引脚，低压探测器，时钟监视器或看门狗定时器（WDT2）溢出来生成复位信号。
WAIT	-

25.1.4 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。该寄存器是特殊寄存器，并且只有在对规定的顺序组合下才能写入（参见 3.4.7 特殊寄存器）。

此寄存器也用来规定提供片上调试功能的引脚是作为片上调试引脚还是作为一般的端口/外围功能引脚。也可用于断开 P05/INTP2/ $\overline{\text{DRST}}$ 引脚的内部下拉电阻。

OCDM 寄存器仅当 $\overline{\text{DRST}}$ 引脚输入为低电平时可以写入。

该寄存器支持 8 位或 1 位读写方式。

复位后: 01H^注 R/W 地址: FFFF9FCH

	7	6	5	4	3	2	1	<0>
OCDM	0	0	0	0	0	0	0	OCDM0

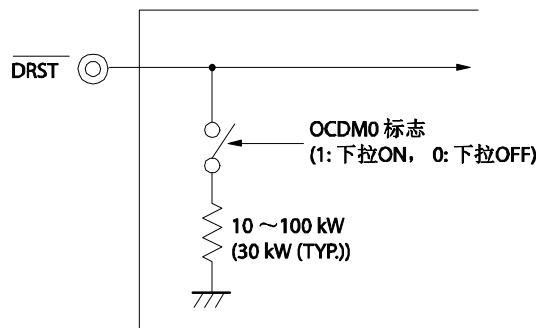
OCDM0	操作模式
0	选择正常操作模式(在该模式下引脚 轮流作为片上调试功能引脚及 端口/外围功能引脚使用)及断开P05/INTP2/DRST 引脚的片上下拉电阻。
1	当DRST 引脚处于低电平: 正常操作模式(在该模式下引脚 轮流作为片上调试功能引脚及 端口/外围功能引脚使用) 当DRST 引脚处于高电平: 片上调试模式(在该模式下引脚 轮流作为片上调试功能引脚及片上调试模式引脚使用)。

注 RESET 输入设置此寄存器为 01H。通过 WDT2RES 信号复位后，时钟监视器 (CLM)，或低电压探测器 (LVI)，OCDM 寄存器的值仍被保留。

注意事项 1. 外部复位后，当 DDI，DDO，DCK，和 DMS 引脚不用于片上调试引脚而作为端口引脚时，可以进行下面的任何一项操作。

- 输入低电压到 P05/INTP2/DRST 引脚。
- 设置 OCDM0 位。在这种情况下，进行如下操作。
 - <1> OCDM0 位清零。
 - <2> P05/INTP2/DRST 引脚恒为低电平直到<1> 完成。

2. DRST 引脚有片上下拉电阻。这个电阻当 OCDM0 标志清零时断开。



25.1.5 操作

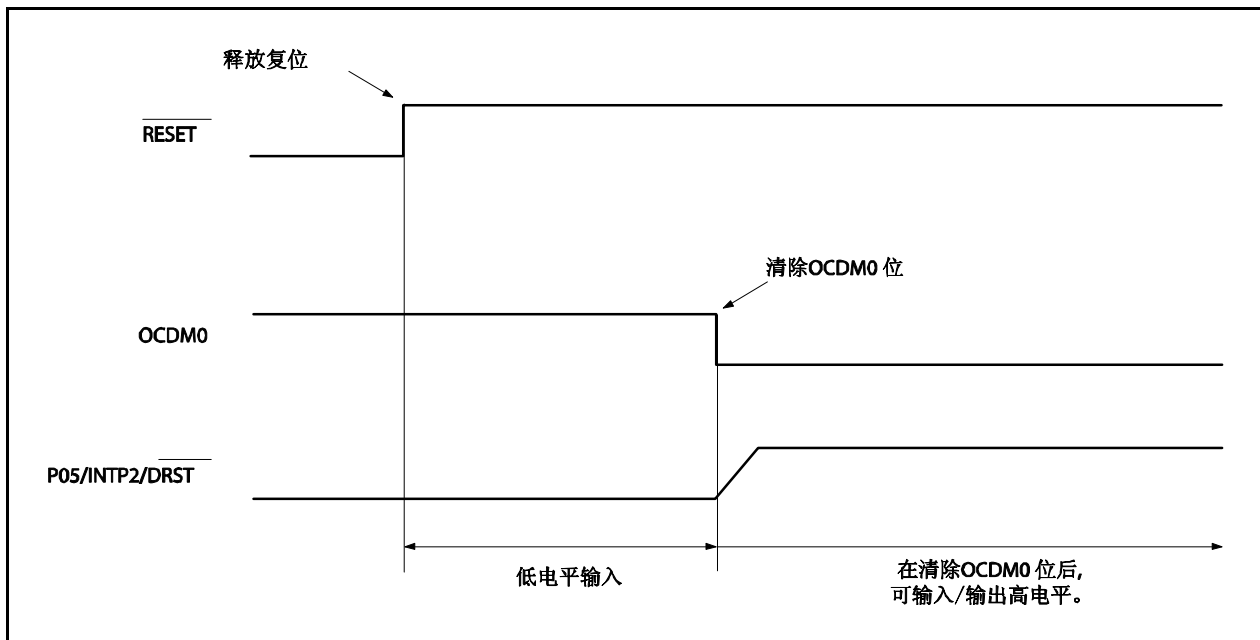
在下表中列出的条件下片上调试功能无效。

当不使用此功能时，保持 $\overline{\text{DRST}}$ 引脚为低直到 OCDM.OCDM0 标志被清零。

OCDM0 标志 $\overline{\text{DRST}}$ 引脚	0	1
L	无效	无效
H	无效	有效

备注 L: 低电平输入
H: 高电平输入

图 25-2. 不使用片上调试功能时的时序



25.1.6 注意事项

- (1) RUN（程序执行）期间，如果输入复位信号（从目标系统或从内部复位源来的复位信号），暂停功能就可能出现故障。
- (2) 即使复位信号由屏蔽功能屏蔽，如果复位信号是从引脚输入的话，I/O 缓冲器（端口引脚）也可能复位。
- (3) 因为内部 flash 存储器内的软件断点设置是通过 ROM 修正功能实现的，它可以通过目标复位或由看门狗定时器 2 产生的内部复位强制为无效。当硬件暂停或者强制暂停发生时，断点再次有效，在此期间不产生软件暂停。
- (4) 暂停期间屏蔽引脚复位屏蔽，CPU 和 外围 I/O 不复位。用户程序执行的时候，flash 存储器一旦由 DMM 重写或者由 RAM 监视功能读取就马上产生引脚复位或内部复位的情况下，CPU 和外围 I/O 可能不正确复位。
- (5) 当下列条件 (a) 和 (b) 满足时，由于暂停等原因仿真器（IECUBE[®]，MINICUBE）操作停止的情况下，看门狗定时器 2 并不停止，并且产生复位或不可屏蔽中断。当发生复位时，调试器挂起。
 - (a) 主时钟或副时钟用作看门狗定时器 2 的源时钟。
 - (b) 内部振荡时钟停止（RCM.RSTOP 位 = 1）。

为避免这种情况，可进行如下两种处理。

- 当使用仿真器时，内部振荡时钟用作源时钟。
 - 当使用仿真器时，内部振荡器不要停止。
- (6) 当下列条件 (a) 和 (b) 满足时，由于暂停等原因仿真器（IECUBE，MINICUBE）操作停止的情况下，TMM 并不停止，即使外围暂停功能设置为“Break”。
 - (a) INTWT，内部振荡时钟（ $f_r/8$ ），或者选择副时钟用作 TMM 源时钟。
 - (b) 主时钟停止。

为避免这种情况，可进行如下两种处理。

- 当使用仿真器时，主时钟（ f_{xx} ， $f_{xx}/2$ ， $f_{xx}/4$ ， $f_{xx}/64$ ， $f_{xx}/512$ ）用作源时钟。
 - 当使用仿真器时，禁止主时钟振荡。
- (7) 在片上调试模式下，DDO 引脚强制为高电平输出。

25.2 不使用 DCU 调试

下面描述如何用 MINICUBE2 实现片上调试功能。UARTA0 引脚（RXDA0 和 TXDA0）或 CSIB0（SIB0，SOB0，SCKB0，和 HS（PMC0））作为调试引脚，不使用 DCU。

25.2.1 电路连接示例

图 25-3. 使用 UARTA0/CSIB0 作为通信接口的电路连接示例

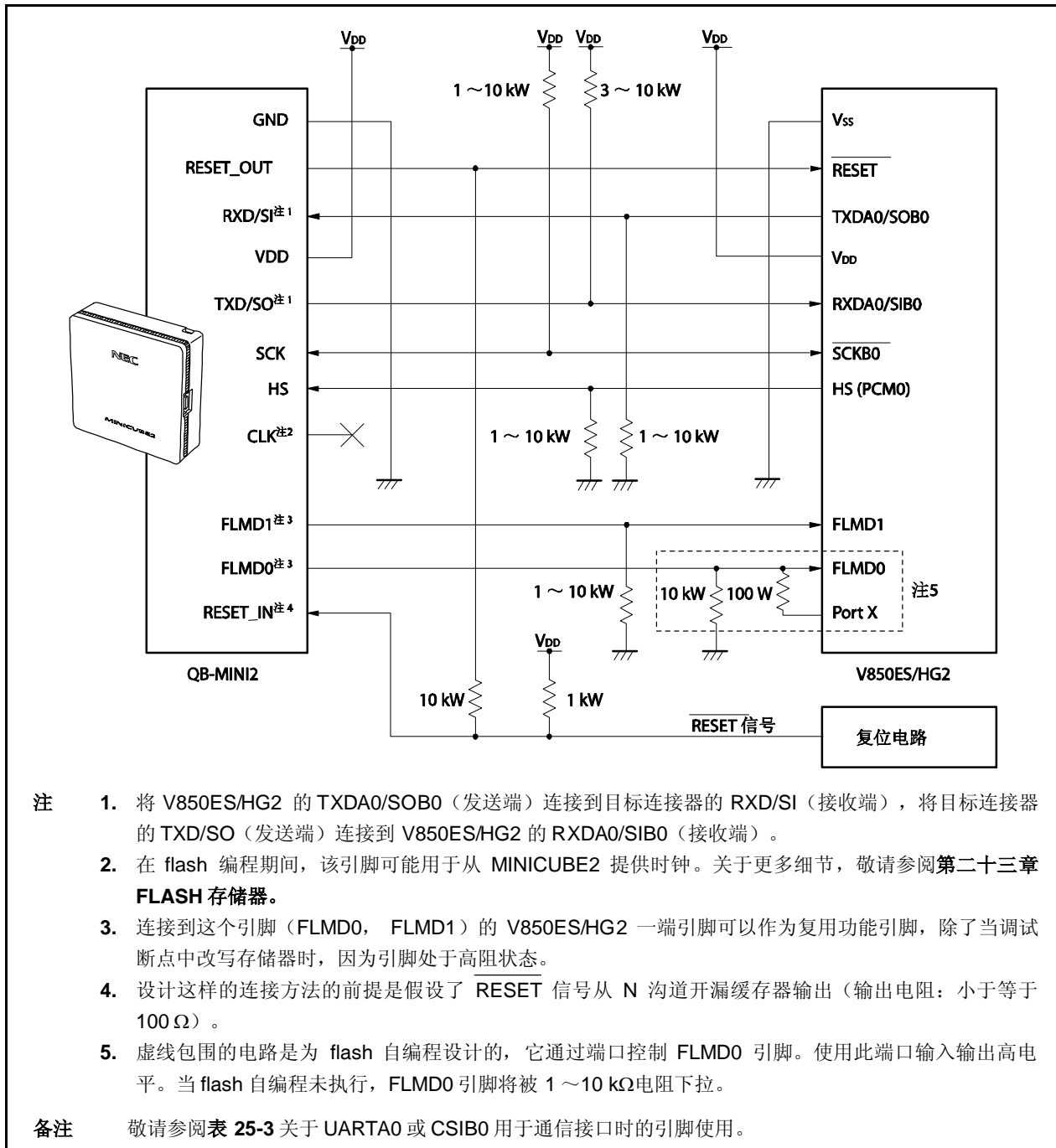


表 25-3. V850ES/HG2 与 MINICUBE2 之间的连线

MINICUBE2 (QB-MINI2) 的引脚配置			通过 CSIB0-HS		通过 UARTA	
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	从 V850ES/HG2 接收命令和数据引脚	P41/SOB0	23	P30/TXD0	25
SO/TxD	输出	向 V850ESHG2 发送命令和数据引脚	P40/SIB0	22	P31/RXD0	26
SCK	输出	3 线串行通信时钟输出引脚	P42/SCKB0	24	不需要	-
CLK ^注	输出	向 V850ES/HG2 时钟输出引脚	不需要 ^注	-	不需要 ^注	-
			不需要 ^注	-	不需要 ^注	-
RESET_OUT	输出	向 V850ES/HG2 复位输出引脚	RESET	14	RESET	14
FLMD0	输出	设置 V850ES/HG2 调试模式或可编程模式的输出引脚	FLMD0	8	FLMD0	8
FLMD1	输出	设置可编程模式的输出引脚	PDL5/FLMD1	76	PDL5/FLMD1	76
HS	输入	CSIB0 + HS 通信的握手信号	PCM0	61	不需要	-
GND	-	地	V _{SS}	11	V _{SS}	11
			AV _{SS}	2	AV _{SS}	2
			BV _{SS}	69	BV _{SS}	69
			EV _{SS}	33	EV _{SS}	33
RESET_IN	输入	目标系统的复位输入引脚				

注 作为 MINICUBE2 FLASH 编程器的时钟输出。关于更多细节，敬请参阅第二十三章 FLASH 存储器。

25.2.2 可屏蔽功能

只有复位信号可被屏蔽。

调试器 (ID850QB) 的可屏蔽功能以及相应的 V850ES/HG2 功能如下。

表 25-4. 可屏蔽功能

ID850QB 的可屏蔽功能	相应的 V850ES/HG2 功能
NMI0	-
NMI1	-
NMI2	-
STOP	-
HOLD	-
RESET	RESET 引脚输入产生的 RESET 信号
WAIT	-

25.2.3 保护用户资源

用户必须做如下准备，以执行 MINICUBE2 和目标设备的通信，并实现每种调试功能。这些项目需要在用户程序中设置或使用编译器选项。

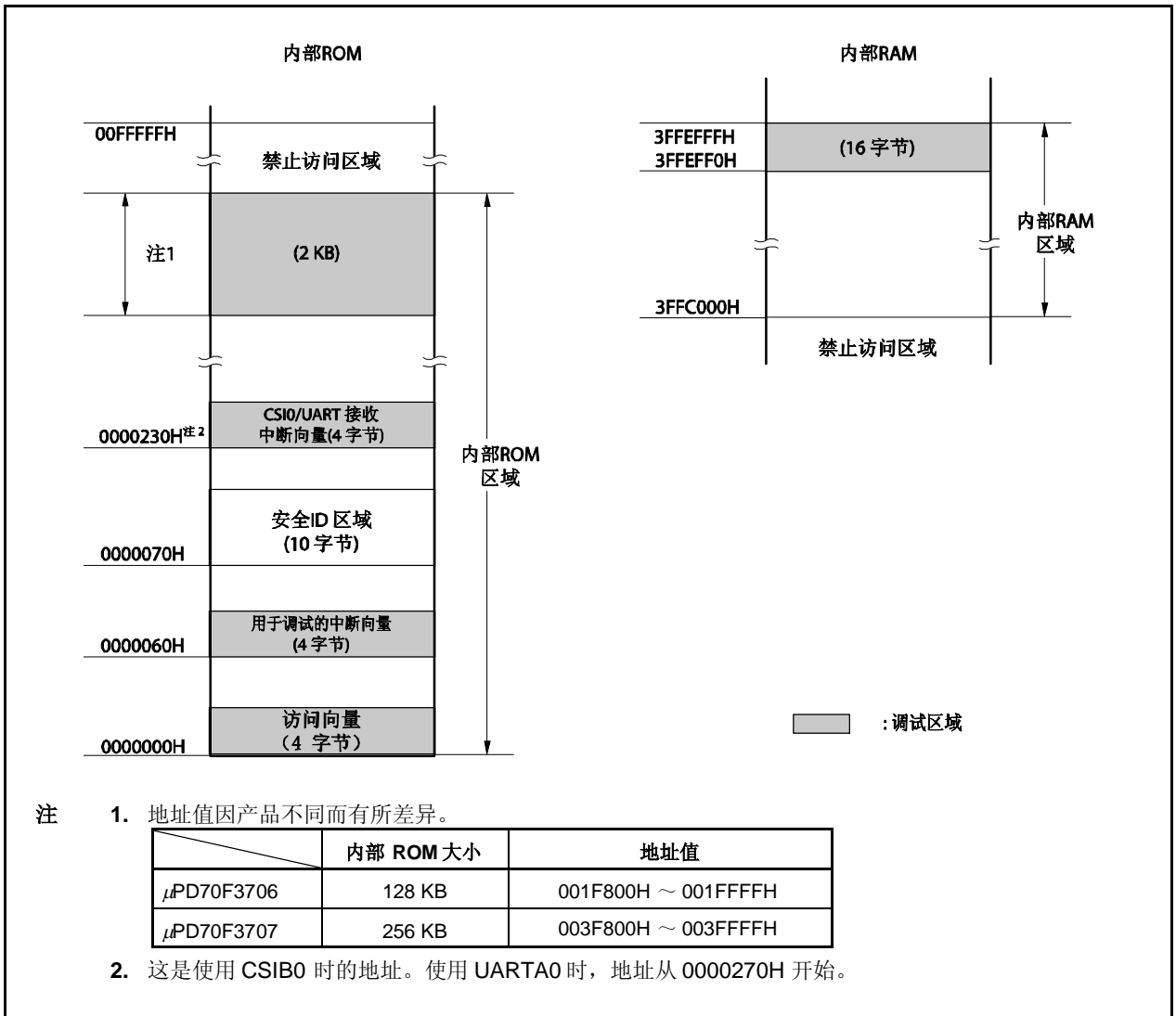
(1) 保护内存空间

图 25-4 阴影部分是放置调试监测程序的保留区域，因此用户程序和数据不能分配到这些空间。这些空间必须受到保护，不能被用户程序使用。

(2) 安全 ID 设置

ID 码必须嵌入 0000070H~0000079H 之间的区域，如图 25-4，以防止内存被没有授权的人读取。关于更多细节，敬请参阅 **25.3ROM 安全功能**。

图 25-4. 调试监测程序分配的内存空间



(3) 复位向量

一个复位向量包含调试监测程序的跳转指令。

[如何保护区域]

无需有意保护此区域。但当下载程序时，调试器将复位向量按如下情况改写。如果改写类型与如下情况不匹配，调试器产生错误（F0C34 当使用 ID850QB 时）。

(a) 从地址 0 开始连续放置 2 条 nop 指令

改写前		改写后
0x0 nop	→	跳转到 0x0 处调试监测程序
0x2 nop		0x4 xxxx
0x4 xxxx		

(b) 从地址 0 连续放置 2 个 0xFFFF（已擦除设备）

改写前		改写后
0x0 0xFFFF	→	跳转到 0x0 处调试监测程序
0x2 0xFFFF		0x4 xxxx
0x4 xxxx		

(c) jr 指令放在地址 0 处（使用 CA850 时）

改写前		改写后
0x0 jr disp22	→	跳转到 0x0 处调试监测程序
		0x4 jr disp22 - 4

(d) mov32 和 jmp 从地址 0 连续放置（使用 IAR 编译器 ICCV850）

改写前		改写后
0x0 mov imm32, reg1	→	跳转到 0x0 处调试监测程序
0x6 jmp [reg1]		0x4 mov imm32, reg1
		0xa jmp [reg1]

(e) 地址 0 处放置调试监测程序的跳转指令

改写前		改写后
跳转到 0x0 处调试监测程序	→	不改变

(4) 调试监测程序区域的保护

图 25-4 阴影部分是分配调试监测程序的区域。监测程序对调试通信接口执行初始化处理，对 CPU 执行断点处理。内部 ROM 区域必须用 0xFF 填充。该区域不得被用户程序重写。

[如何保护区域]

如果用户程序不用此区域则无需保护此区域。

为避免可能在调试启动时发生的问题，建议用编译器预先保护此区域。

下面为保护此区域的示例，应用 NEC 电子编译器 CA850。添加汇编源文件和链接指示码如下所示。

- 汇编源（添加如下代码作为汇编源文件。）

```
-- 确保监视器 ROM 区域有 2 KB 的空间
.区域 "监视器 ROM", const
.空间 0x800, 0xff

--确保中断向量用于调试
.区域 "DBG0"
.空间 4, 0xff

--确保中断向量用于串行通信
--根据使用的串行通信模式改变区域名称
.区域 "INTCB0R"
.空间 4, 0xff

--确保监视器 RAM 区域有 16 字节的空间
.区域 "监视器 RAM", bss
.lcomm monitorramsym, 16, 4 -- 定义标识 monitorramsym
```

- 直连（将下列代码加入直连文件。）

如下所示为内部 ROM 为 256KB（终端地址为 003FFFFH）内部 RAM 为 24KB（终端地址为 3FFEFFFH）。

```
MROMSEG      : !LOAD ?R V0x03f800{
                MonitorROM    = $PROGBITS    ?A MonitorROM;
};

MRAMSEG      : !LOAD ?RW V0x03ffeff0{
                MonitorRAM    = $NOBITS      ?AW MonitorRAM;
};
```

(5) 通信串行接口的保护

UARTA0 或 CSIB0 用于在 MINICUBE2 和目标系统之间通信。与串行接口模式相关的设置由调试监测程序执行，但是如果设置被用户程序改变，将可能发生通信错误。

为防止这样的问题发生，通信串行接口需要在用户程序中被保护。

[如何保护通信串行接口]

- 片上调试模式寄存器 (OCDM)

对于使用 UARTA0 或 CSIB0 的片上调试功能，设置 OCDM 寄存器为正常模式。一定按以下步骤设置。

- 输入低电平至 P05/INTP2/ $\overline{\text{DRST}}$ 引脚。
- 按如下所示设置 OCDM0 位
 - <1> OCDM0 位清零。
 - <2> 将 P05/INTP2/ $\overline{\text{DRST}}$ 引脚输入设置为低电平直至步骤<1>结束。

- 串行接口寄存器

勿对与用户程序中 CSIB0 或 UARTA0 有关的寄存器进行设置。

- 中断屏蔽寄存器

在使用 CSIB0 时，不能屏蔽传输端中断 (INTCB0R)。在使用 UARTA0 时，不能屏蔽接收端中断 (INTUA0R)。

(a) 使用 CSIB0 时

	7	6	5	4	3	2	1	0
CB0RIC	x	0	x	x	x	x	x	x

(b) 使用 UARTA0 时

	7	6	5	4	3	2	1	0
UA0RIC	x	0	x	x	x	x	x	x

备注 x: 忽略

• 使用 UARTA0 时的端口寄存器

当使用 UARTA0 时，端口寄存器通过调试监测程序设置为使得 TXDA0 和 RXDA0 引脚有效。当用户程序调试时，不要改变如下寄存器设置。（同样的值可以多次写入。）

	7	6	5	4	3	2	1	0
PMC3L	x	x	x	x	x	x	1	1

备注 x: 忽略

• 使用 CSIB0 时的端口设置

使用 CSIB0 时，端口寄存器通过调试监测程序设置为使得 SIB0，SOB0， $\overline{\text{SCKB0}}$ ，和 HS（PMC0）引脚有效。调试时不要在用户程序中改变如下寄存器设置。（同样的值可以多次写入。）

(a) SIB0，SOB0，和 $\overline{\text{SCKB0}}$ 设置

	7	6	5	4	3	2	1	0
PMC4	x	x	x	x	x	1	1	1

(b) HS（PMC0 引脚）设置

	7	6	5	4	3	2	1	0
PMCM	x	x	x	x	x	x	x	0

	7	6	5	4	3	2	1	0
PCM	'	'	'	'	'	'	'	注

注 禁止写入该位。
与 HS 引脚对应的端口值根据调试器状态被监测程序改变。要以 8 位单元执行端口寄存器设置，用户程序通常可以使用读-修改-写。如果调试中断在写入前发生，将可能执行一个不可预料的操作。

备注 x: 忽略

25.2.4 注意事项

(1) 用于调试的设备的处理

不要将用于调试的设备安装在大批量生产的产品上，因为调试过程中 flash 存储器会被改写，flash 存储器改写的次数不能保证。此外，不要把调试监测程序嵌入大批量生产的产品中。

(2) 断点不能执行时

如果满足下列条件之一，强制断点不能执行。

- 中断禁止 (DI)
- 发给串行接口并用于 MINICUBE2 和目标设备之前通信的中断被屏蔽
- 禁任由可屏蔽中断释放待机时进入待机模式
- MINICUBE2 与目标设备之间通信模式为 UARTA0 且主时钟被 MINICUBE2 停止

(3) 当伪实时 RAM 监测器 (RRM) 功能和 DMM 功能不工作时

满足如下条件之一时，伪实时 RAM 监测器 RRM 功能和 DMM 功能不工作。

- 中断禁止 (DI)
- 发给串行接口用于 MINICUBE2 和目标设备之间通信的中断被屏蔽
- 由于禁止可屏蔽中断释放待机而进入待机模式
- MINICUBE2 与目标设备之间通信模式为 UARTA0，且主时钟被 MINICUBE2 停止
- MINICUBE2 与目标设备之间通信模式为 UARTA0，且用于通信的时钟与调试器指定的时钟不同

(4) 使能伪 RRM 和 DMM 功能时的待机释放

如果如下条件之一满足，待机模式由伪 RRM 和 DMM 功能释放。

- MINICUBE2 和目标设备之间的通信模式为 CSIB0
- MINICUBE2 和目标设备之间的通信模式为 UARTA0，且主时钟已加入。

(5) 应用 DMM 功能写需要特定顺序的外围 I/O 寄存器

需要特定顺序的外围 I/O 寄存器不能用 DMM 功能写入。

(6) 使调试器启动变慢的设备

当调试器初次启动时，将进行芯片清除，写入用于调试的监视程序，但该操作需 10 多秒钟。

(7) 写入用于调试的监视程序

当 CPU 操作时钟设置与调试器一起改变时，调试器重写监视程序。所需时间与上述 (6) 相同。对于集成调试器 ID850QB，当改变时钟条及配置对话框时该程序仍然适用。

(8) flash 自编程

如果分配了调试监测程序的空间被 flash 自编程改写，调试器不能再正常工作。

25.3 ROM 安全功能

25.3.1 安全 ID

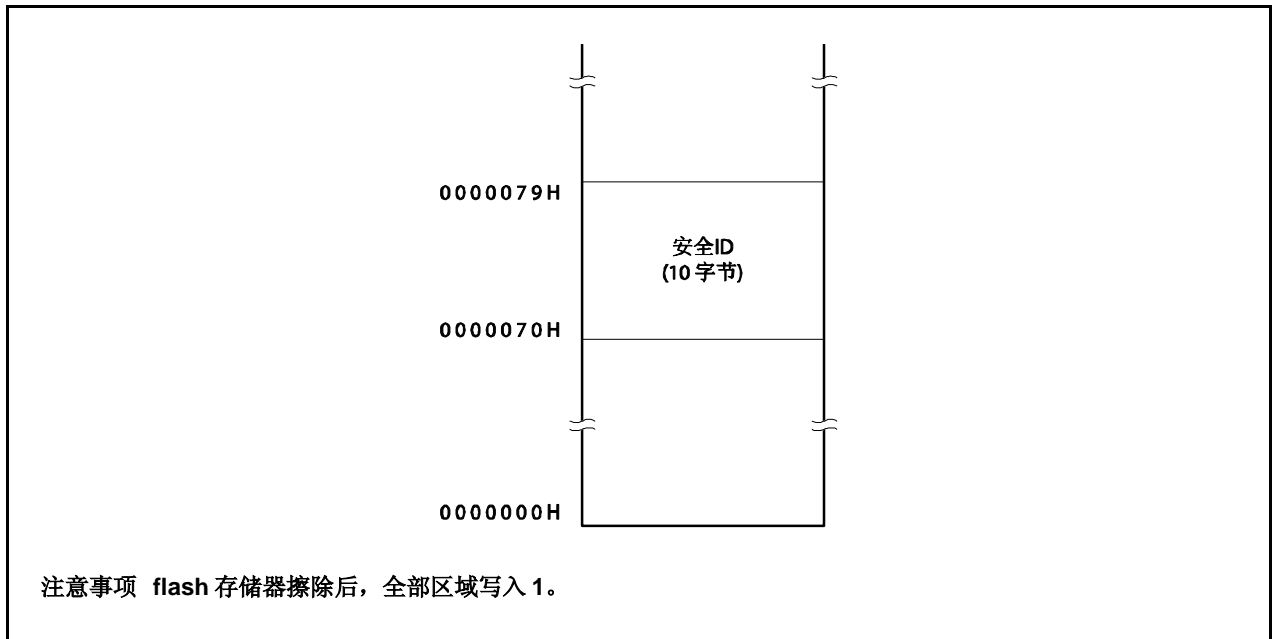
V850ES/HG2 的 flash 存储器版使用 10 个字节的 ID 码来进行加密，以防在片上调试期间 flash 存储器的内容被没有授权的人通过片上调试仿真器读取。

在 10 个字节的片上 flash 从 0000070H~0000079H 的存储区域进行设置，对调试器进行 ID 加密。

如果 IDs 一致，那么就释放安全保密功能，读取和使用片上调试仿真器均可。

- 在 0000070H ~ 0000079H 之间设置 10 个字节的 ID 码。
- 0000079H 的位 7 是片上调试仿真器的使能标志。
(0: 禁止, 1: 使能)
- 当片上调试仿真器启动时，调试器要求输入 ID 码。当输入的 ID 码与 0000070H ~ 0000079H 中设置的 ID 号码一致时，调试器开始工作。
- 如果片上调试仿真器使能标志为 0，即使 ID 号码匹配，也不能进行调试。

图 25-5. 安全 ID 区域



25.3.2 设置

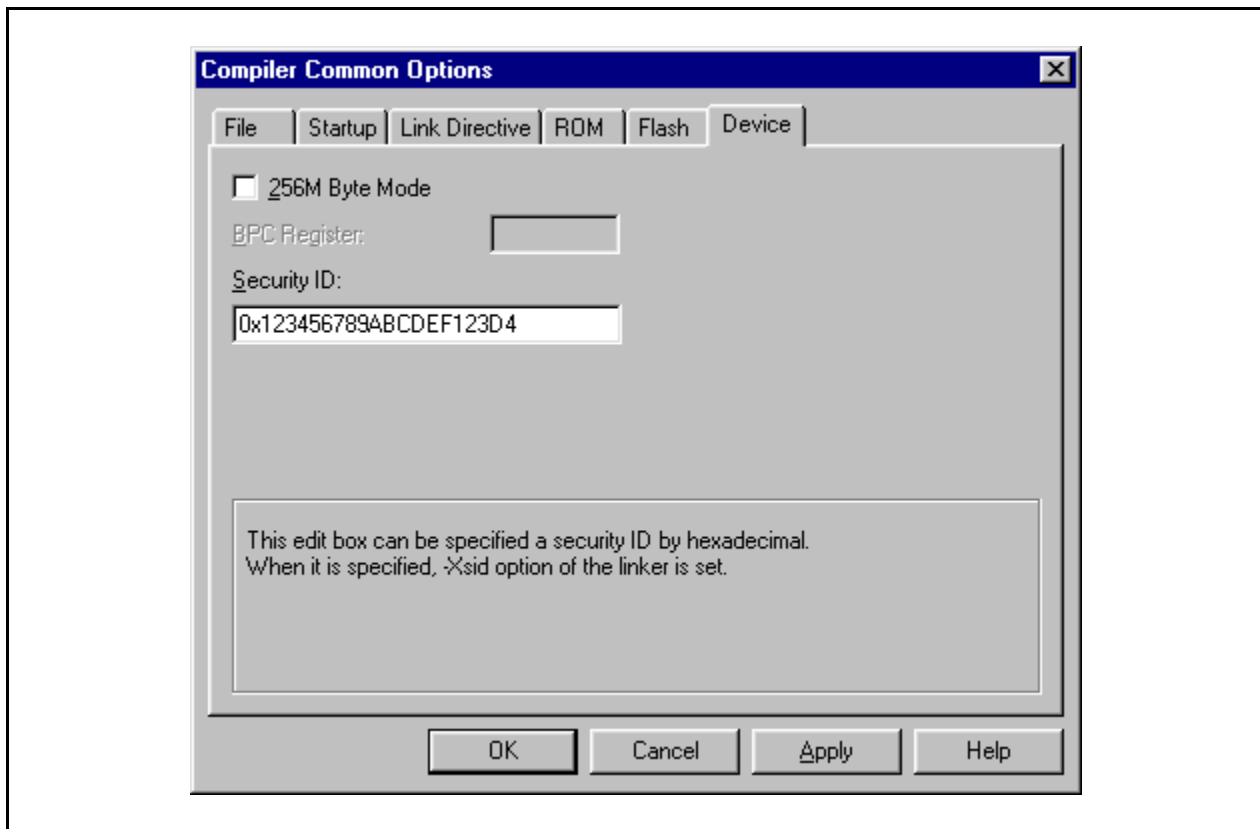
表 25-5 显示如何设置 ID 码。

当 ID 码如表 25-5 所示设置，ID850QB 的配置对话框中的 ID 码输入为“123456789ABCDEF123D4”（不区分 ID 码大小写）。

表 25-5. ID 码

地址	值
0x70	0x12
0x71	0x34
0x72	0x56
0x73	0x78
0x74	0x9A
0x75	0xBC
0x76	0xDE
0x77	0xF1
0x78	0x23
0x79	0xD4

对于支持 CA850 Ver.3.10 或以上版本以及可以应用 PM+ 编译器普通选项设置的安全 ID 的设备文件，可以指定 ID 码。



[程序示例（使用 CA850 Ver. 3.10 或以上版本）]

```
#-----  
# SECURITYID  
#-----  
    .section    "SECURITY_ID"    --中断处理地址 0x70  
    .word      0x78563412        --0-3 字节代码  
    .word      0xF1DEBC9A        --4-7 字节代码  
    .hword     0xD423            --8-9 字节代码
```

备注 将以上程序添加到启动文件中。

第二十六章 电气特性

26.1 最大额定值

最大额定值 (T_A = 25°C) (1/2)

参数	符号	条件	额定值	单位
电源电压	V _{DD}	V _{DD} = EV _{DD} = BV _{DD}	-0.5 ~ +6.5	V
	BV _{DD}	V _{DD} = EV _{DD} = BV _{DD}	-0.5 ~ +6.5	V
	EV _{DD}	V _{DD} = EV _{DD} = BV _{DD}	-0.5 ~ +6.5	V
	AV _{REF0}		-0.5 ~ +6.5	V
	V _{SS}	V _{SS} = EV _{SS} = BV _{SS} = AV _{SS}	-0.5 ~ +0.5	V
	AV _{SS}	V _{SS} = EV _{SS} = BV _{SS} = AV _{SS}	-0.5 ~ +0.5	V
	BV _{SS}	V _{SS} = EV _{SS} = BV _{SS} = AV _{SS}	-0.5 ~ +0.5	V
	EV _{SS}	V _{SS} = EV _{SS} = BV _{SS} = AV _{SS}	-0.5 ~ +0.5	V
输入电压	V _{I1}	P00 ~ P06, P10, P11, P30 ~ P39, P40 ~ P42, P50 ~ P55, P90 ~ P915, RESET, FLMD0	-0.5 ~ EV _{DD} + 0.5 ^注	V
	V _{I2}	PCM0 ~ PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 ~ PDL13	-0.5 ~ BV _{DD} + 0.5 ^注	V
	V _{I3}	X1, X2, XT1, XT2	-0.5 ~ V _{RO} + 0.5	V
模拟输入电压	V _{IAN}	P70 ~ P715	-0.5 ~ AV _{REF0} + 0.5 ^注	V

注 确保不要超过每个电源电压的最大额定值（最大值）。

- 注意事项**
1. 避免 IC 设备输出引脚（或 I/O）及 V_{DD} 或 V_{CC} 与 GND 的直接接触。
 2. 任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。
DC 特性和 AC 特性中指出的额定值和条件是正常工作的质量保证。
 3. 当将外部电路与处于高电阻状态的引脚直接相连时，设计的时序可使外部电路上避免输出冲突。

备注 除非另外说明，复用功能引脚的特性与端口引脚的相同。

最大额定值 (T_A = 25°C) (2/2)

参数	符号	条件	额定值	单位	
输出电流, 低	I _{OL}	P00 ~ P06, P10, P11, P30 ~ P39, P40 ~ P42, P50 ~ P55, P90 ~ P915	每个引脚	4	mA
			所有引脚总和	50	mA
		P70 ~ P715	每个引脚	4	mA
			所有引脚总和	20	mA
		PCM0 ~ PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 ~ PDL13	每个引脚	4	mA
			所有引脚总和	50	mA
输出电流, 高	I _{OH}	P00 ~ P06, P10, P11, P30 ~ P39, P40 ~ P42, P50 ~ P55, P90 ~ P915	每个引脚	-4	mA
			所有引脚总和	-50	mA
		P70 ~ P715	每个引脚	-4	mA
			所有引脚总和	-20	mA
		PCM0 ~ PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 ~ PDL13	每个引脚	-4	mA
			所有引脚总和	-50	mA
工作温度	T _A	正常操作模式	-40 ~ +85	°C	
		Flash 存储器编程模式			
存储温度	T _{stg}		-40 ~ +125	°C	

- 注意事项**
1. 不要直接将 IC 产品的输出 (或 I/O) 引脚连接起来, 或连接到 V_{DD}, V_{CC} 和 GND。
 2. 任何一项参数哪怕是在瞬间超过最大额定值, 都会使产品质量受到影响。也就是说, 最大额定值是产品濒临物理损坏的临界点, 因而, 必须保证产品在不超过最大额定值的条件下使用。
DC 特性和 AC 特性中指出的额定值和条件是正常工作的质量保证。
 3. 当将外部电路与处于高电阻状态的引脚直接相连时, 设计的时序可使外部电路上避免输出冲突。

备注 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

26.2 电容

($T_A = 25^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = BV_{DD} = V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
I/O 电容	C_{IO}	$f_X = 1\text{ MHz}$ 不可测量的引脚返回 0 V。			10	pF

26.3 操作条件

($T_A = -40 \sim +85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
内部系统时钟频率	f_{CLK}	REGC = 4.7 μF , 在主时钟下工作	4		20	MHz
		REGC = 4.7 μF , 在副时钟下工作 (晶体振荡器)	32		35	kHz
		REGC = 4.7 μF , 在副时钟下工作 (RC 振荡器)	12.5 ^注		27.5 ^注	kHz

注 内部系统时钟频率为振荡器频率的一半。

26.4 振荡器特性

26.4.1 主时钟振荡器特性

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

振荡器	电路示例	参数	条件	MIN.	TYP.	MAX.	单位
陶瓷振荡器		振荡频率 (fx) ^{注1}		4		5	MHz
		振荡稳定时间 ^{注2}	复位释放后			$2^{16}/f_x$	s
			STOP 模式释放后	0.5 ^{注3}	注4	ms	
晶体振荡器		振荡频率 (fx) ^{注1}		4		5	MHz
		振荡稳定时间 ^{注2}	复位释放后			$2^{16}/f_x$	s
			STOP 模式释放后	0.5 ^{注3}	注4	ms	
			IDLE2 模式释放后	0.35	注4	ms	

- 注
1. 仅指振荡器特性。
 2. 复位操作或 STOP 模式释放后谐振稳定的时间。
 3. 振荡稳定时间访问内部 flash 存储器。
 4. 数值根据 OSTS 寄存器的设置有所变化。

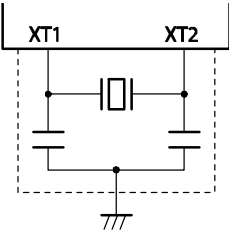
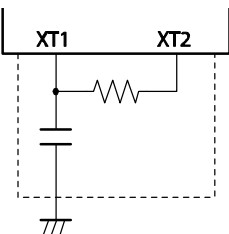
注意事项 1. 当使用主时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。

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

2. 当主时钟停止副时钟工作时，在转换回主时钟前等待直到振荡稳定时间，由程序保证。

26.4.2 副时钟振荡器特性

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5 \text{ V} \sim 5.5 \text{ V}$, $4.0 \text{ V} \leq AV_{REF0} \leq 5.5 \text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 \text{ V}$)

振荡器	电路示例	参数	条件	MIN.	TYP.	MAX.	单位
晶体振荡器		振荡频率 (f_{XT}) ^{注1}		32	32.768	35	kHz
		振荡稳定时间 ^{注2}				10	s
RC 振荡器		振荡频率 ^{注1, 4}	$R = 390 \text{ k}\Omega \pm 5\%$ ^{注3} $C = 47 \text{ pF} \pm 10\%$ ^{注3}	25	40	55	kHz
		振荡稳定时间 ^{注2}				100	μs

- 注
1. 只指示振荡器的特性。关于 CPU 操作时钟，参见 26.8AC 特性。
 2. 所需时间为从 V_{DD} 到达振荡器电压范围（最小值：3.5 V）到振荡器稳定之间的间隔。
 3. 通过尽可能的缩短连接线来防止连接线电容产生不利影响。
 4. RC 振荡器频率为 40 kHz（TYP.）。该时钟内部一分为二。在使用 RC 振荡器的情况下，内部系统时钟频率为振荡器频率的一半。最小值= 12.5 kHz，TYP. = 20 kHz，最大值= 27.5 kHz。

注意事项 1. 当使用副时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。

- 连接线越短越好。
 - 连接线不应与其他信号线交叉。
 - 流经的电流变化较大的信号线不要在振荡器周围布线。
 - 要保持振荡器电容器的接地点电压与 V_{SS} 相同。
 - 不要将电容的地信号接入大电流地。
 - 不要从振荡器获取信号。
2. 为了缩减能耗，副时钟振荡器被设计为低幅电路，并且它比主时钟振荡器更容易受到噪声的影响，此在使用副时钟时更要注意布线方法。

26.4.3 PLL 特性

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
输入频率	f_x		4		5	MHz
输出频率	f_{xx}		16		20	MHz
锁时间	t_{PLL}	V_{DD} 达到 3.5 V 之后 (最小值)			800	μs

26.4.4 内部振荡器特性

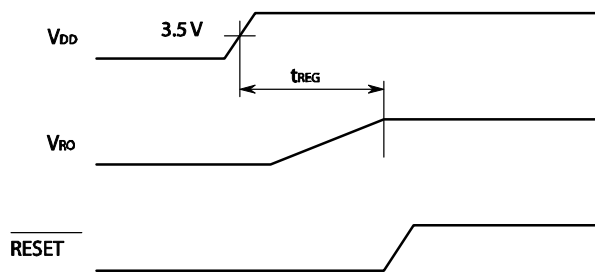
($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
输出频率	f_R		100	200	400	kHz

26.5 电压调节器特性

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
输入频率	V_{DD}		3.5		5.5	V
输出频率	V_{RO}			2.5		V
锁时间	t_{REG}	V_{DD} 达到 3.5 V 之后 (最小值), $C = 4.7\ \mu\text{F} \pm 20\%$ 连接到 REGC 引脚			1	ms



<R>

注意事项 当 $\overline{\text{RESET}} = V_{SS} = 0\text{ V}$ 确保 V_{DD} 电压上升。

26.6 DC 特性

26.6.1 I/O 电平

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)
(1/2)

参数	符号	条件	MIN.	TYP.	MAX.	单位
输入电压, 高	V_{IH1}	P30, P34, P36 ~ P38, P41, P98, P911	$0.7EV_{DD}$		EV_{DD}	V
	V_{IH2}	P00 ~ P06, P10, P11, P31 ~ P33, P35, P39, P40, P42, P50 ~ P55, P90 ~ P97, P99, P910, P912 ~ P915	$0.8EV_{DD}$		EV_{DD}	V
	V_{IH3}	PCM0 ~ PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 ~ PDL13	$0.7BV_{DD}$		BV_{DD}	V
	V_{IH4}	P70 ~ P715	$0.7AV_{REF0}$		AV_{REF0}	V
	V_{IH5}	$\overline{\text{RESET}}$, FLMD0	$0.8EV_{DD}$		EV_{DD}	V
输出电压, 低	V_{IL1}	P30, P34, P36 ~ P38, P41, P98, P911	EV_{SS}		$0.3EV_{DD}$	V
	V_{IL2}	P00 ~ P06, P10, P11, P31 ~ P33, P35, P39, P40, P42, P50 ~ P55, P90 ~ P97, P99, P910, P912 ~ P915	EV_{SS}		$0.2EV_{DD}$	V
	V_{IL3}	PCM0 ~ PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 ~ PDL13	BV_{SS}		$0.3BV_{DD}$	V
	V_{IL4}	P70 ~ P715	AV_{SS}		$0.3AV_{REF0}$	V
	V_{IL5}	$\overline{\text{RESET}}$, FLMD0	EV_{SS}		$0.2EV_{DD}$	V

备注 除非另有说明, 复用功能引脚的特性与其他端口引脚相同。

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)
(2/2)

参数	符号	条件	MIN.	TYP.	MAX.	单位	
输出电压, 高 ^{注1}	V _{OH1}	P00 ~ P06, P10, P11, P30 ~ P39, P40 ~ P42, P50 ~ P55, P90 ~ P915	I _{OH} = -1.0 mA	EV _{DD} - 1.0		EV _{DD}	V
			I _{OH} = -0.1 mA	EV _{DD} - 0.5		EV _{DD}	V
	V _{OH2}	PCM0 ~ PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 ~ PDL13	I _{OH} = -1.0 mA	BV _{DD} - 1.0		BV _{DD}	V
			I _{OH} = -0.1 mA	BV _{DD} - 0.5		BV _{DD}	V
	V _{OH3}	P70 ~ P715	I _{OH} = -1.0 mA	AV _{REF0} - 1.0		AV _{REF0}	V
			I _{OH} = -0.1 mA	AV _{REF0} - 0.5		AV _{REF0}	V
输出电压, 低 ^{注1}	V _{OL1}	P00 ~ P06, P10, P11, P30 ~ P39, P40 ~ P42, P50 ~ P55, P90 ~ P915	I _{OL} = 1.0 mA	0		0.4	V
	V _{OL2}	PCM0 ~ PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 ~ PDL13	I _{OL} = 1.0 mA	0		0.4	V
	V _{OL3}	P70 ~ P715	I _{OL} = 1.0 mA	0		0.4	V
上拉电阻	R ₁	V _I = 0 V	10	30	100	kΩ	
下拉电阻 ^{注2}	R ₂	V _I = V _{DD}	10	30	100	kΩ	

- 注 1. 各电源 (EV_{DD}, BV_{DD}, AV_{REF0}) 的所有 I_{OH}/I_{OL} 相加的最大值为 20 mA/-20 mA。
2. 仅为 DRST 引脚。

备注 除非另行说明, 复用功能引脚与其它端口引脚的特性相同。

26.6.2 引脚漏电流

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
输入漏电流, 高	I _{LIH1}	V _{IN} = V _{DD}	模拟引脚		+0.2	μA
			非模拟引脚		+0.5	
输入漏电流, 低	I _{LIL1}	V _{IN} = 0 V	模拟引脚		-0.2	μA
			非模拟引脚		-0.5	
输出漏电流, 高	I _{LOH1}	V _O = V _{DD}	模拟引脚		+0.2	μA
			非模拟引脚		+0.5	
输出漏电流, 低	I _{LOL1}	V _O = 0 V	模拟引脚		-0.2	μA
			非模拟引脚		-0.5	

注意事项 FLMD0 引脚值如下。

- 输入漏电流, 高: 2 μA (最大。)
- 输入漏电流, 低: -2 μA (最大。)

26.6.3 电源电流

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5 \text{ V} \sim 5.5 \text{ V}$, $4.0 \text{ V} \leq AV_{REF0} \leq 5.5 \text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 \text{ V}$)

参数	符号	条件		MIN.	TYP.	MAX.	单位	
电源电流 ^{注 1}	I _{DD1}	正常工作模式	f _{xx} = 20 MHz (f _x = 5 MHz)	所有外围功能操作		30	45	mA
				停止所有外围功能		23		mA
	I _{DD2}	HALT模式	f _{xx} = 20 MHz (f _x = 5 MHz)	所有外围功能操作		18	28	mA
				停止所有外围功能		11		mA
	I _{DD3}	IDLE1模式	f _{xx} = 5 MHz (f _x = 5 MHz), PLL 关闭			0.6	0.9	mA
	I _{DD4}	IDLE2模式	f _{xx} = 5 MHz (f _x = 5 MHz), PLL 关闭			0.25	0.7	mA
	I _{DD5}	副时钟操作模式 ^{注 2 3}	晶体振荡器 (f _{XT} = 32.768 kHz)			200	400	μA
			RC 振荡器 (f _{XT} = 40 kHz ^{注 4})			200	400	μA
	I _{DD6}	副 IDLE 模式 ^{注 2 3}	晶体振荡器 (f _{XT} = 32.768 kHz)			20	120	μA
			RC 振荡器 (f _{XT} = 40 kHz ^{注 4})			35	140	μA
	I _{DD7}	STOP 模式 ^{注 2 5}	停止 POC, 停止内置振荡器			7	50	μA
			停止 POC 操作, 停止内置振荡器			10	55	μA
			停止 POC, 停止内置振荡器			15	65	μA
停止 POC 操作, 停止内置振荡器				18	70	μA		

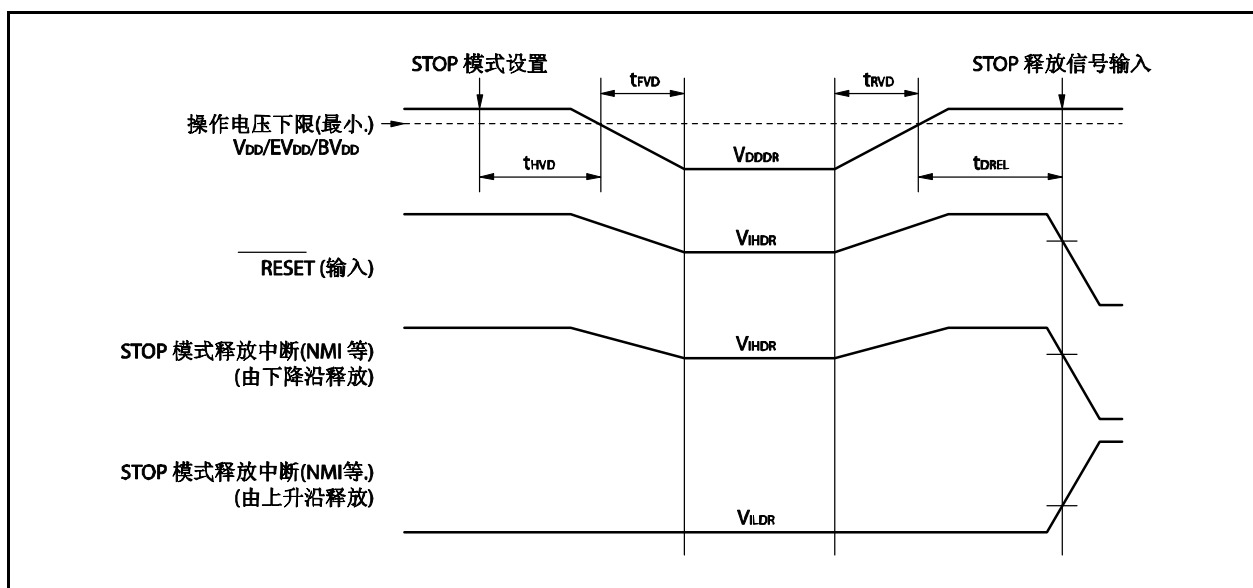
- 注
1. V_{DD}, EV_{DD} 和 BV_{DD} 电流总和 (所有端口关闭) 不包括 AV_{REF0} 电流, 及通过片上上拉/下拉电阻的端口缓冲电流。
 2. 当主时钟振荡停止时。
 3. 操作 POC, 操作内置振荡器。
 4. RC 振荡器频率为 40 kHz (TYP.) 该时钟内部一分为二。
 5. 当副时钟振荡不使用时。

26.7 数据保持特性

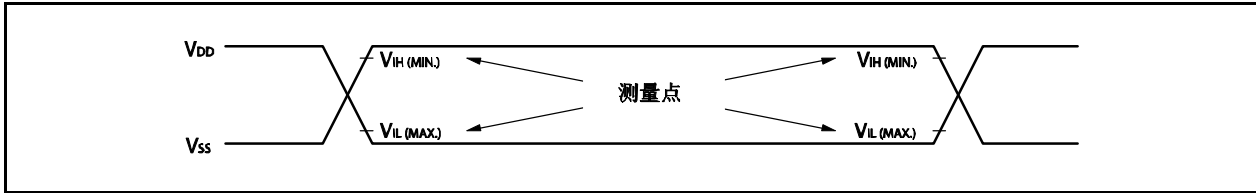
STOP 模式 ($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 1.9\text{ V} \sim 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
数据保持电压	V_{DDDR}	在 STOP 模式下 (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	μA
电源电压上升时间	t_{RVD}		1			μs
电源电压下降时间	t_{FVD}		1			μs
电源电压稳定时间	t_{HVD}	释放 STOP 模式后	0			ms
STOP 释放信号输入时间	t_{DREL}	V_{DD} 达到最小电压 3.5V 后	0			ms
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

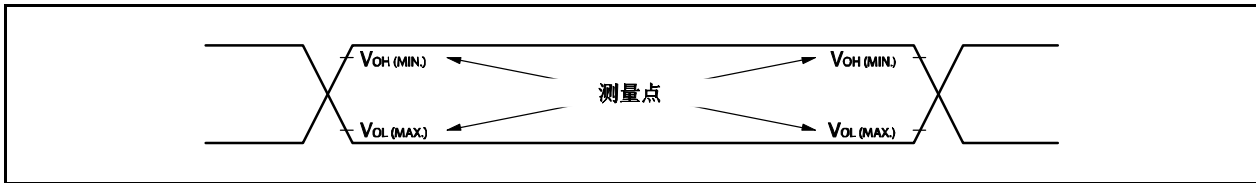
注意事项 转换到 STOP 模式和从 STOP 模式还原必须在额定的工作范围完成。



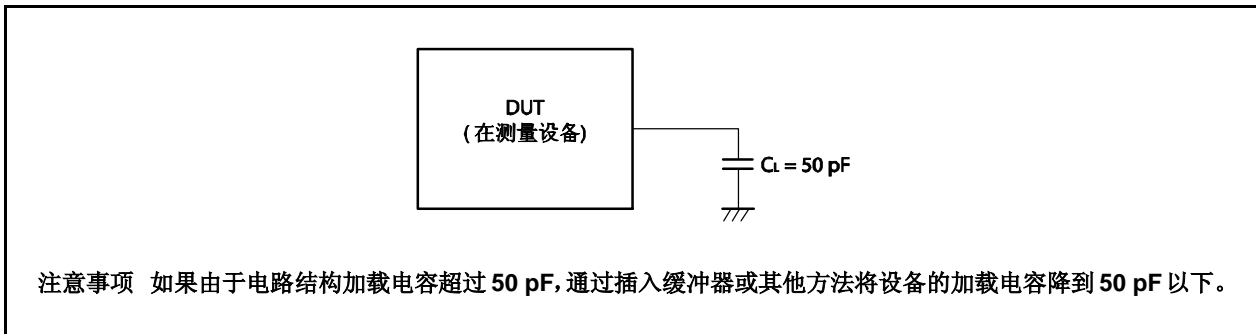
26.8 AC 特性

(1) AC 输入测试点 (V_{DD} , AV_{REF0} , EV_{DD} , BV_{DD})

(2) AC 输出测试点



(3) 加载条件

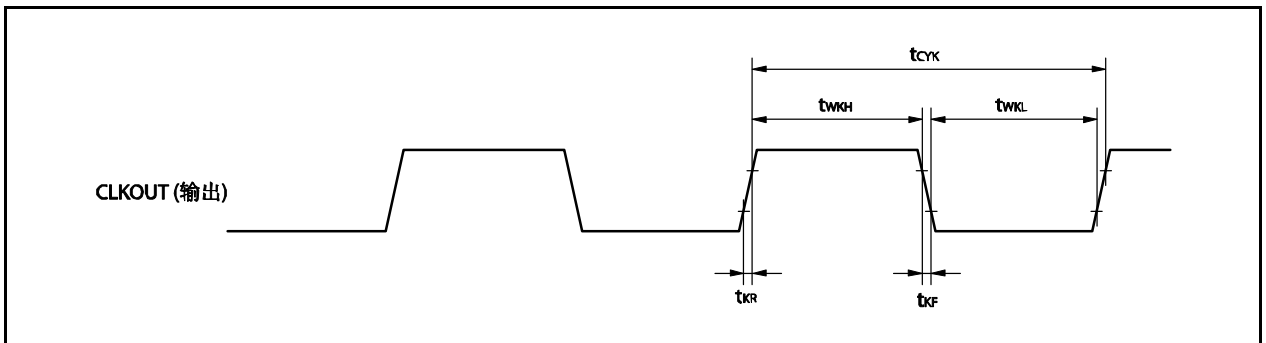


26.8.1 CLKOUT 输出时序

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
输出周期	t_{CYK}		50 ns	80 μs	
高电平宽度	t_{VKH}		$t_{CYK}/2 - 15$		ns
低电平宽度	t_{VKL}		$t_{CYK}/2 - 15$		ns
上升时间	t_{KR}			15	ns
下降时间	t_{KF}			15	ns

时钟时序



26.9 基本操作

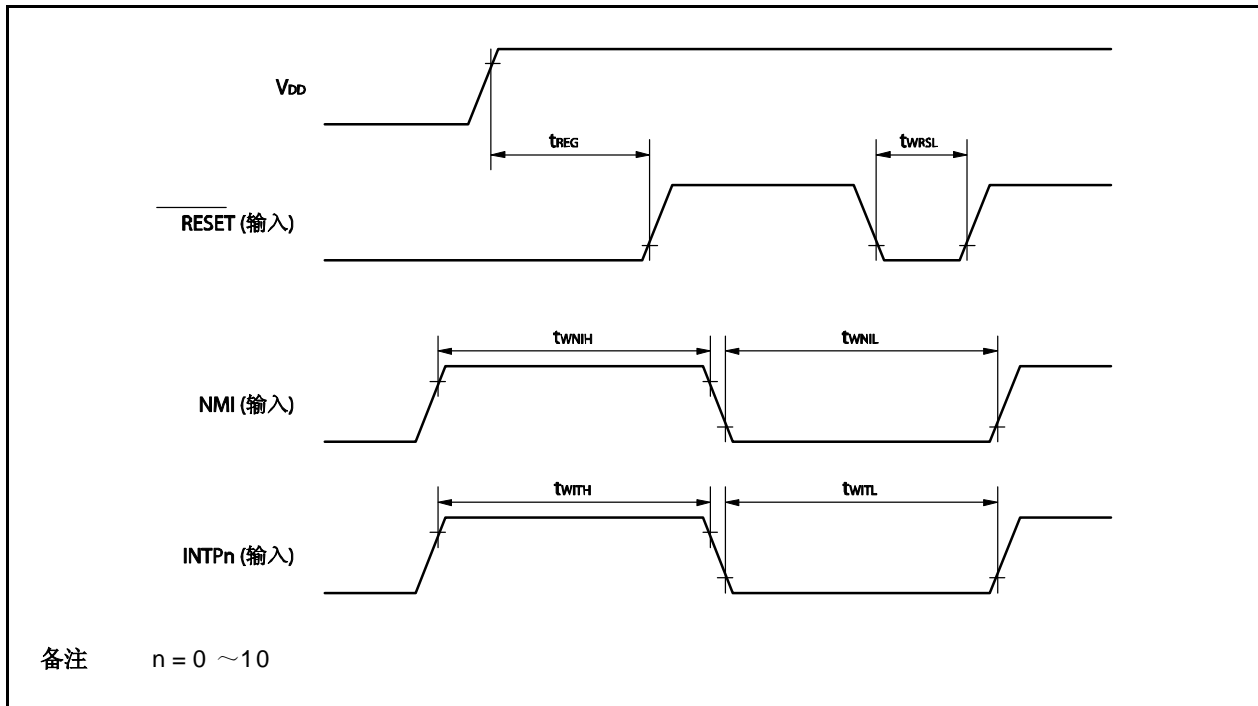
(1) 复位, 中断时序

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
RESET 低电平宽度	t_{WRSL}		500		ns
NMI 高电平宽度	t_{WNIH}	模拟噪声消除	500		ns
NMI 低电平宽度	t_{WNIL}	模拟噪声消除	500		ns
INTPn ^{※1} 高电平宽度	t_{WTH}	模拟噪声消除 ($n = 0 \sim 10$)	500		ns
		选择数字噪声消除时 ($n = 3$)	注 2		ns
INTPn ^{※1} 低电平宽度	t_{WTL}	模拟噪声消除 ($n = 0 \sim 10$)	500		ns
		选择数字噪声消除时 ($n = 3$)	注 2		ns

- 注 1. 在 ADTRG 引脚的情况下, 应用与 INTP0/P03 引脚值相同的数值。在 DRST 引脚情况下, 应用与 NTP2/P05 引脚值相同的数值
2. $2T_{\text{samp}} + 20$ 或 $3T_{\text{samp}} + 20$
 T_{samp} : 用于噪声清除的采样时钟

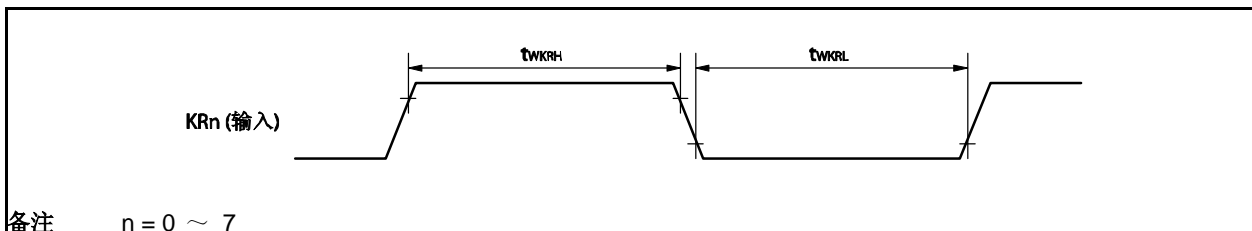
复位/中断



(2) 按键中断时序

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5 \text{ V} \sim 5.5 \text{ V}$, $4.0 \text{ V} \leq AV_{REF0} \leq 5.5 \text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 \text{ V}$, $C_L = 50 \text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
KRn 输入高电平宽度	t_{WKRH}	模拟噪声消除 ($n = 0 \sim 7$)	500		ns
KRn 输入低电平宽度	t_{WKRL}		500		ns

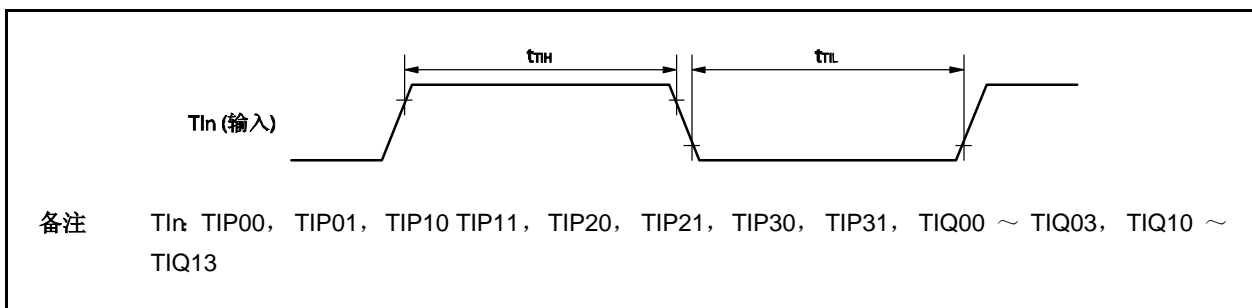


(3) 定时器输入定时

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5 \text{ V} \sim 5.5 \text{ V}$, $4.0 \text{ V} \leq AV_{REF0} \leq 5.5 \text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 \text{ V}$, $C_L = 50 \text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
TIn 高电平宽度	t_{TIH}	TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, TIP31,	注 2		ns
TIn 低电平宽度	t_{TIL}	TIQ00~TIQ03, TIQ10 ~TIQ13	注 2		ns

- 注
- 当输入捕捉信号时，可清除 TIP00, TIP10, TIP20, TIP30, TIQ00, 和 TIQ10 引脚上的噪声。当输入外部触发信号或外部事件计数器时，噪声无法被清除。
 - $2T_{\text{samp}} + 20$ 或 $3T_{\text{samp}} + 20$
 T_{samp} : 用于噪声清除的采样时钟



(4) CSIB 时序

(a) 主设备模式

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
SCKBn 周期时间	t_{CYn}		125		ns
$\overline{\text{SCKBn}}$ 高电平宽度	t_{KHn}		$t_{CYn}/2 - 15$		ns
$\overline{\text{SCKBn}}$ 低电平宽度	t_{LN}		$t_{CYn}/2 - 15$		ns
SIBn 建立时间 (到 $\overline{\text{SCKBn}}\uparrow$)	t_{SIKn}		30		ns
SIBn 保持时间 (自 $\overline{\text{SCKBn}}\uparrow$)	t_{SIn}		25		ns
自 $\overline{\text{SCKBn}}\downarrow$ 到 SOBn 输出的延迟时间	t_{SON}			25	ns

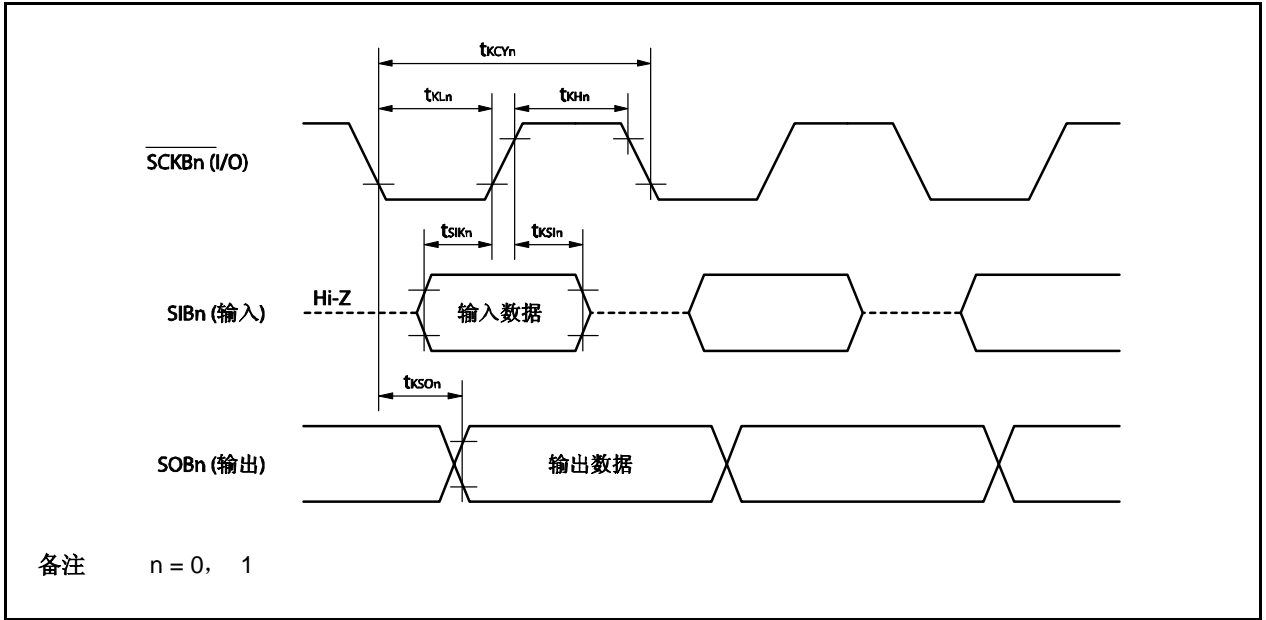
备注 $n = 0, 1$

(b) 从模式

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
$\overline{\text{SCKBn}}$ 循环时间	t_{CYn}		200		ns
$\overline{\text{SCKBn}}$ 高电平宽度	t_{KHn}		90		ns
$\overline{\text{SCKBn}}$ 低电平宽度	t_{LN}		90		ns
SIBn 设定时间 (到 $\overline{\text{SCKBn}}\uparrow$)	t_{SIKn}		50		ns
SIBn 保持时间 (自 $\overline{\text{SCKBn}}\uparrow$)	t_{SIn}		50		ns
自 $\overline{\text{SCKBn}}\downarrow$ 到 SOBn 输出延迟时间	t_{SON}			50	ns

备注 $n = 0, 1$



(5) UARTA 时序

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
通讯速率				312.5	kbps
ASCK0 周期时间				10	MHz

(6) A/D 转换器

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
分辨率					10	位
总误差 ^注		$4.0 \leq AV_{REF0} \leq 5.5\text{ V}$		± 0.15	± 0.3	%FSR
转换时间	t_{CONV}		3.1		16	μs
模拟输入电压	V_{IAN}		AV_{SS}		AV_{REF0}	V
AVREF0 电流	I_{AREF0}	使用 A/D 转换时		5	10	mA
		不用 A/D 转换时		1	10	μA

注 不包括量化误差 ($\pm 0.05\% \text{FSR}$)。指示比值为满值 (%FSR)。

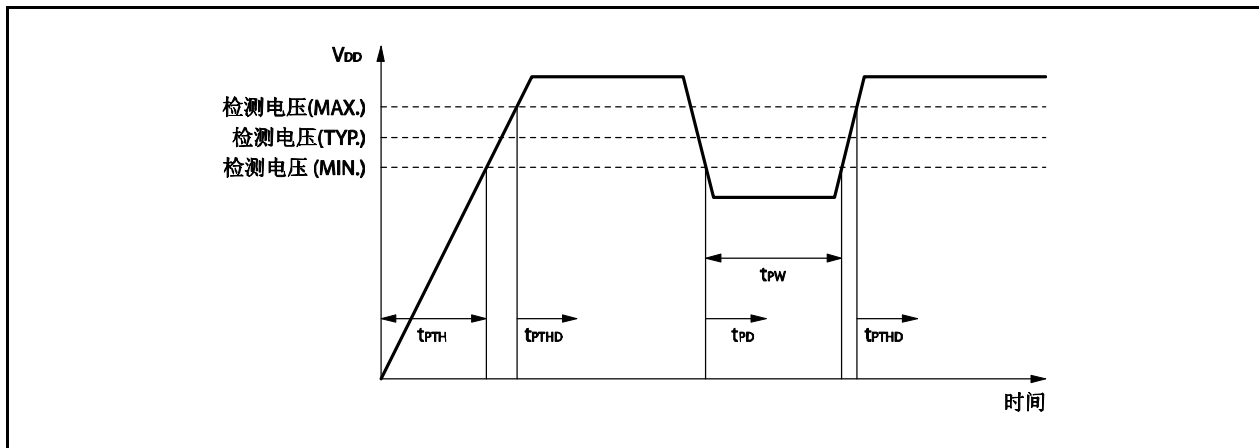
备注 FSR: 满度范围

(7) POC 电路特性

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	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
反应延迟时间 1 ^{#1}	t_{PTHD}	在加电源时 V_{DD} 电压达到 3.9V 后			3.0	ms
反应延迟时间 2 ^{#2}	t_{PD}	在失电时 V_{DD} 电压降到 3.5V 后			1	ms
最小 V_{DD} 宽度	t_{PW}		0.2			ms

- 注
1. 在检测到检测电压后所需的复位释放时间。
 2. 在检测到检测电压后所需的输出复位时间。

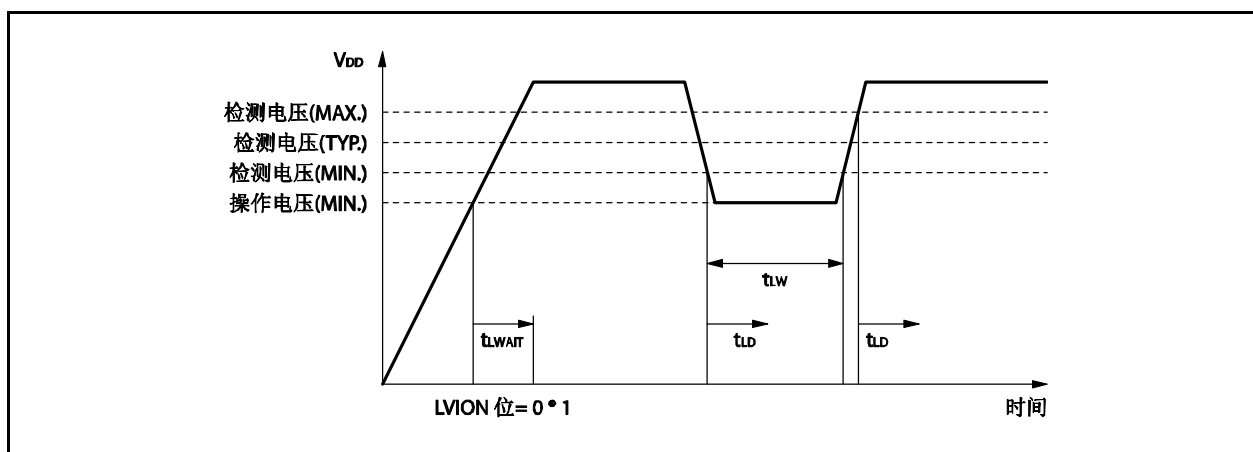


(8) LVI 电路特性

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
检测电压	V_{LVI0}		4.2	4.4	4.6	V
	V_{LVI1}		4.0	4.2	4.4	V
反应时间 ^{注1}	t_{LD}	在 V_{DD} 达到 V_{LVI0}/V_{LVI1} (MAX.) 或降到 V_{LVI0}/V_{LVI1} 之下后 (MIN.)		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 ^{注2}	t_{LWAIT}	在 V_{DD} 达到 3.5 V 或 LVION 位 (LVIM.位 7) 由 0 变为 1 后		0.1	0.2	ms

- 注 1. 在检测到检测电压后, 所需的输出中断/复位时间。
 2. 当使用 POC 功能时, 不需要该稳定时间。

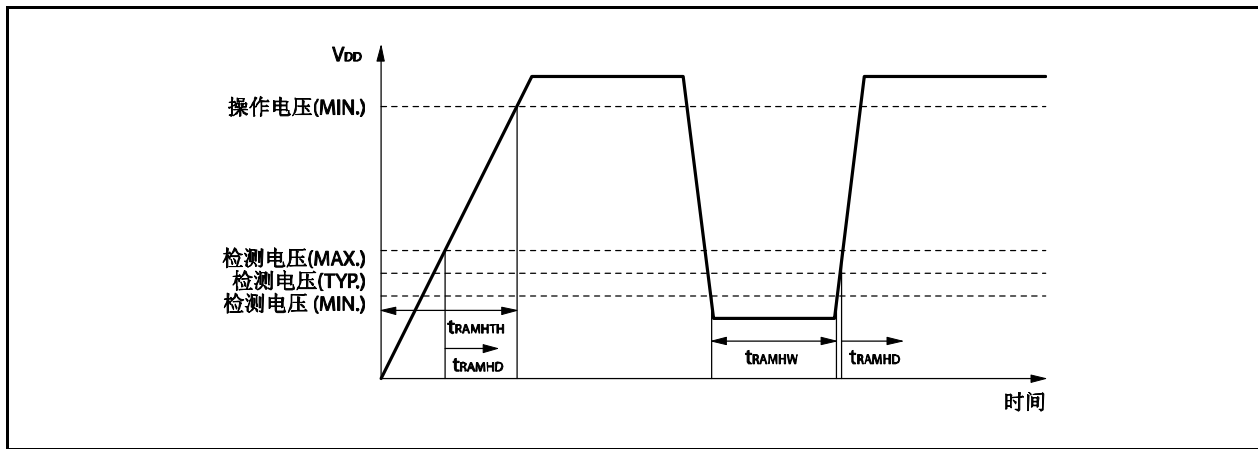


(9) RAM 保持标记特性

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5\text{ V} \sim 5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.最小	TYP.	MAX.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
反应时间 ^注	t_{RAMHD}	在电源电压达到检测电压后 (最大)		0.2	2.0	ms
最小 V_{DD} 宽度	t_{RAMHW}		0.2			ms

注 在检测到检测电压后所需的设置 RAMF 位时间。



26.10 Flash 存储器编程特性

(1) 基本特性

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5 \text{ V} \sim 5.5 \text{ V}$, $4.0 \text{ V} \leq AV_{REF0} \leq 5.5 \text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 \text{ V}$, $C_L = 50 \text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
编程工作频率	f_{CPU}		4		20	MHz
电源电压	V_{DD}		3.5		5.5	V
重写次数	$C_{\text{WRT}}^{\#}$				100	次
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	$t_{\text{WRT}} + t_{\text{ERASE}}$				TBD	s
编程温度	t_{PRG}		-40		+85	$^\circ\text{C}$

注 在出货后第一次对产品进行写操作时，“擦除 → 写”和“只写”作为一次改写。

示例 示例 (P: 写入, E: 擦除)

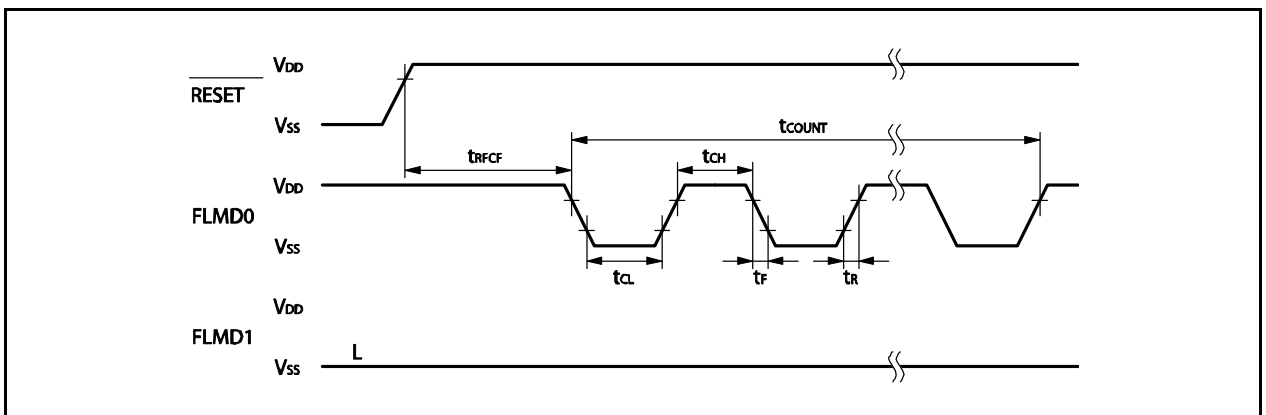
出库产品 → P → E → P → E → P: 3 次改写

出库产品 → E → P → E → P → E → P: 3 次改写

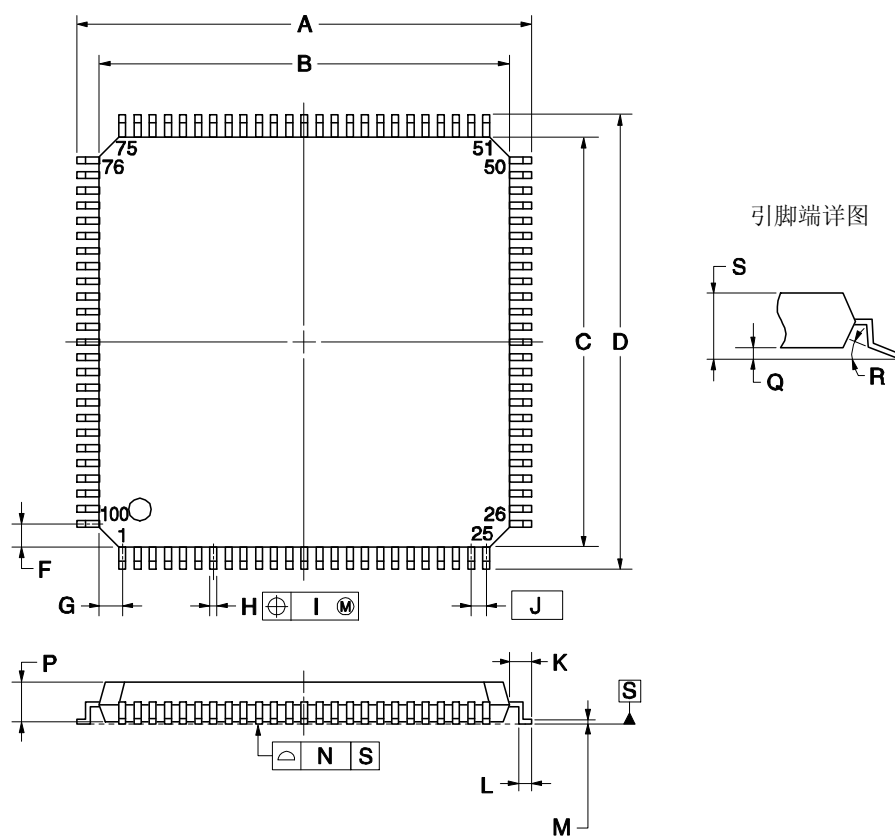
(2) 串行写入操作特性

($T_A = -40 \sim +85^\circ\text{C}$, $V_{DD} = EV_{DD} = BV_{DD} = 3.5 \text{ V} \sim 5.5 \text{ V}$, $4.0 \text{ V} \leq AV_{REF0} \leq 5.5 \text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 \text{ V}$, $C_L = 50 \text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
来自 RESET \uparrow 的 FLMD0 设置时间	t_{RFCF}		70536/ f_x			s
执行计数时间	t_{COUNT}				3	ms
FLMD0 高电平宽度	t_{CH}		10		100	μs
FLMD0 低电平宽度	t_{CL}		10		100	μs
FLMD0 上升时间	t_{R}				50	ns
FLMD0 下降时间	t_{F}				50	ns



100 引脚塑封 LQFP (密脚距) (14X14)



注
该结构在最大使用材料情况下，每条引脚的中心线位于其实际位置的 0.08mm (T.P.) 内

项目	尺寸
A	16.00±0.20
B	14.00±0.20
C	14.00±0.20
D	16.00±0.20
F	1.00
G	1.00
H	0.22 ^{+0.05} _{-0.04}
I	0.08
J	0.50 (T.P.)
K	1.00±0.20
L	0.50±0.20
M	0.17 ^{+0.03} _{-0.07}
N	0.08
P	1.40±0.05
Q	0.10±0.05
R	3° ^{+7°} _{-3°}
S	1.60 MAX.

S100GC-50-8EU, 8EA-2

第二十八章 推荐焊接条件

V850ES/HG2 应在如下推荐条件下焊接和装配。

关于技术信息，敬请参阅如下网站。

半导体设备装配手册(<http://www.necel.com/pkg/en/mount/index.html>)

表 28-1. 贴装焊接条件

μPD70F3706GC-8EA-A: 100 引脚塑料 LQFP (密脚距) (14 × 14)

μPD70F3707GC-8EA-A: 100 引脚塑料 LQFP (密脚距) (14 × 14)

焊接方法	焊接条件	推荐型号
红外线	最高温度: 260°C, 时间: 最多 60 秒 (220°C 或更高), 次数小于等于 3 次, 暴露限制: 7 天 ^注 (之后在 125°C 预烘 20 ~ 72 小时)	IR60-207-3
局部加热	针脚温度: 最大 350°C, 时间: 最长 3 秒。(行针)	-

注 干燥下打开后, 存储在低于 25°C 和 65% RH 下, 或减少存储时间。

注意事项 不要同时使用不同的焊接方式 (除局部加热外)。

- 备注**
1. 零件号以 A 结尾的产品是无铅产品。
 2. 关于软焊方法和状况的更多细节, 请联系 NEC 电子客服代理。

在使用 V850ES/HG2 的系统开发中可应用如下开发工具。

图 A-1 所示为开发工具的组成。

- **支持 PC98-NX 系列**

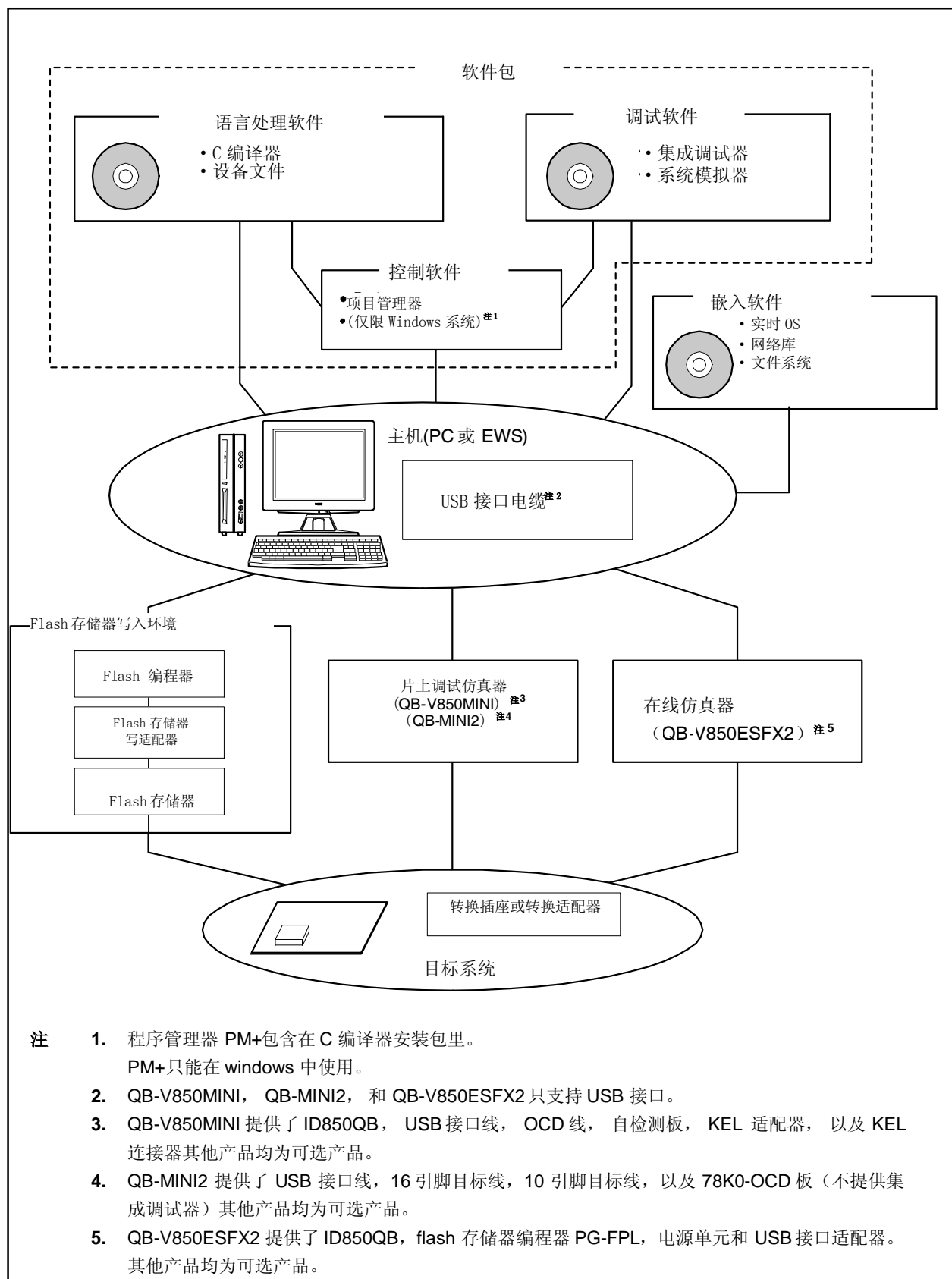
除非特别说明，IBM PC/AT™ 兼容机支持的产品与 PC98-NX 系列计算机是兼容的。当使用 PC98-NX 系列计算机时，参阅 IBM PC/AT 兼容机的使用说明。

- **Windows™**

除非特别说明，“Windows”指以下几种操作系统。

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

图 A-1. 开发工具的组成



A.1 软件包

SP850 V850 微控制器系列软件包	本软件包包含适用于 V850 微控制器的系列开发工具（软件） 产品型号： $\mu S_{xxxx}SP850$
-------------------------	---

备注 产品型号中的 xxxx 随主机和使用的 OS 而变化。

$\mu S_{xxxx}SP850$

xxxx	主机	OS	存储介质
AB17	PC-9800 系列, IBM PC/AT 兼容机	Windows (日文版)	CD-ROM
BB17		Windows (英文版)	

A.2 语言处理软件

CA850 C 编译器安装包	此编译器将 C 语言程序转换为微控制器可执行的目标代码。此编译器由编程管理器 PM+ 启动。 产品型号： $\mu S_{xxxx}CA703000$
DF703712 设备文件	该文件包含设备特有的信息。 该设备文件应结合工具(CA850, V850ES/Hx2 的 SM+ 和 ID850 QB)一起使用。 相应的 OS 和主机随使用工具而变化。

备注 产品型号中的 xxxx 随主机和使用的 OS 而变化。

$\mu S_{xxxx}CA703000$

xxxx	主机	OS	存储介质
AB17	PC-9800 系列	Windows(日文版)	CD-ROM
BB17	IBM PC/AT 兼容机	Windows (英文版)	
3K17	SPARC 站™	SunOS™ (参考 4.1.4), Solaris™ (参考 2.5.1)	

A.3 控制软件

PM+ 编程管理器	这是一个控制软件，可使用户在 Windows 环境下有效地进行软件开发。用户程序开发中进行的任何操作，如启动编辑器、构造程序和启动调试器，都可以由 PM+ 执行。 <注意事项> PM+ 包含在 C 编译器安装包 CA850。 仅在 Windows 下使用。
--------------	--

A.4 调试工具 (硬件)

A.4.1 当使用 IECUBE QB-V850ESFX2 时

当连接 QB-V850ESFX2 到主机 (PC-9800 系列, PC/AT 兼容机) 时的系统配置如下所示。即使没有准备可选产品, 也可进行连接。

图 A-2. 系统配置(使用 QB-V850ESFX2) (1/2)

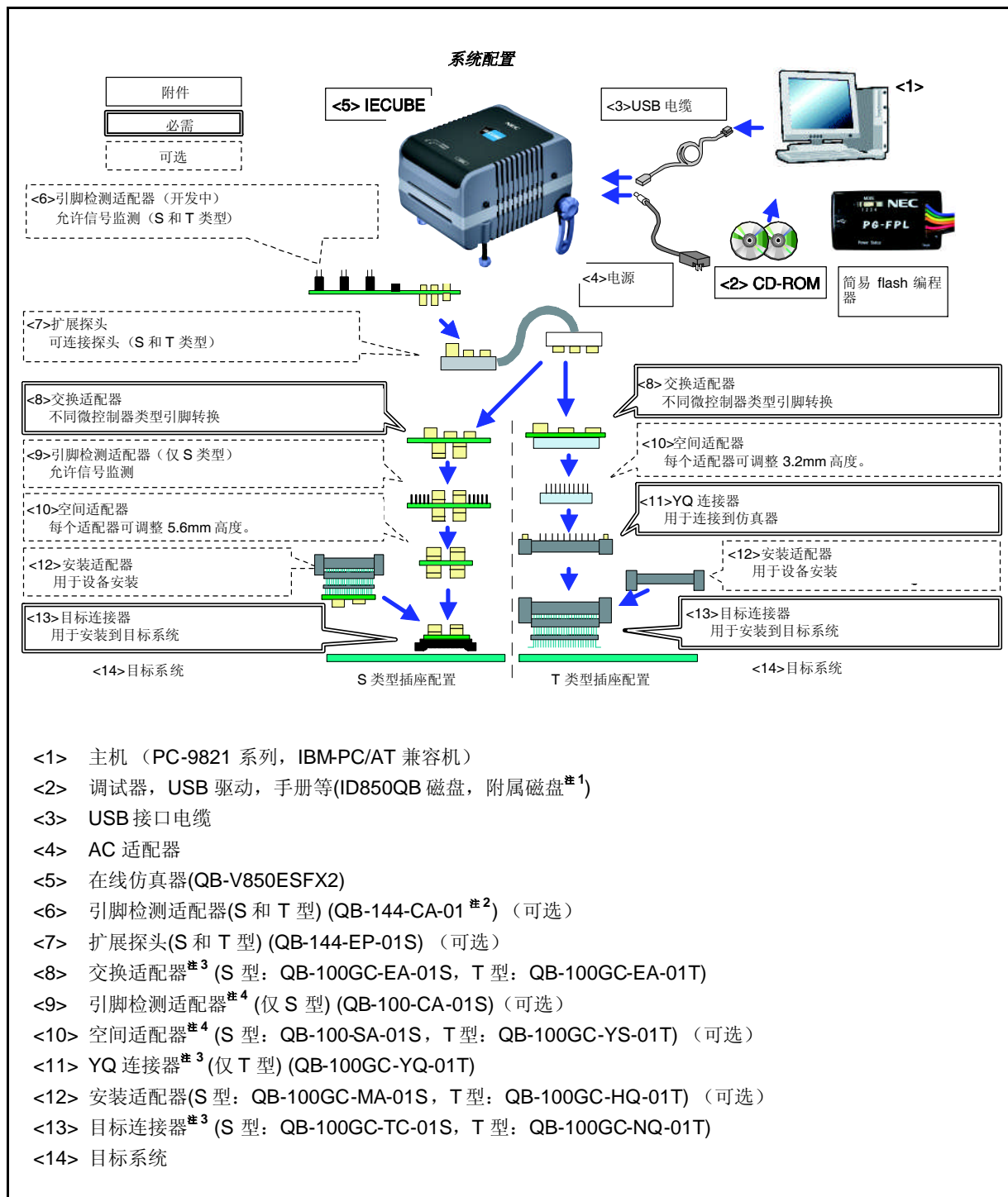


图 A-2. 系统配置(使用 QB-V850ESFX2) (2/2)

注	<ol style="list-style-type: none"> 1. 从 NEC Electronics 网站下载设备文件。 http: //www.necel.com/micro/ods/eng/ 2. 开发中 3. 依照订购的数字提供器件。 <ul style="list-style-type: none"> • 订购 QB-V850ESFX2-ZZZ 时 不提供交换适配器和目标连接器。 • 订购 QB-V850ESFX2-S100GC 时 提供 QB-100GC-EA-01S 和 QB-100GC-TC-01S。 • 订购 QB-V850ESFX2-T100GC 时 提供 QB-100GC-EA-01T, QB-100GC-YQ-01T 和 QB-100GC-NQ-01T。 4. 当同时使用<9>和<10>时, 不必关心<9>和<10>之间的顺序。
---	---

<5> QB-V850ESFX2 ^注 在线仿真器	该在线仿真器用于使用 V850ES/HG2 开发的应用系统的硬件和软件调试。它支持集成调试 ID850 QB。该仿真器应当与电源和仿真头结合使用。使用 USB 接口电缆连接该仿真器和主机。
<3> USB 接口电缆	连接主机与 QB-V850ESFX2 的电缆。
<4> AC 适配器	通过更换 AC 插座可支持 100 到 240V 电压。
<8> QB-100GC-EA-01S QB-100GC-EA-01T 交换适配器	适配器进行引脚转换。
<9> QB-100-CA-01S 检查引脚适配器	使用示波器等用于波形检测的适配器。
<10> QB-100-SA-01S QB-100GC-YS-01T 空间适配器	适配器调整高度。
<11> QB-100GC-YQ-01T YQ 连接器	该连接器连接目标连接器和转换适配器。
<12> QB-100GC-MA-01S QB-100GC-HQ-01T 安装适配器	通过插座安装 V850ES/HG2 的适配器。
<13> QB-100GC-TC-01S QB-100GC-NQ-01T 目标连接器	到目标系统上焊料的接头。

注 QB-V850ESFX2 提供了电源单元, USB 接口电缆和编程器 PG-FPL。它还提供了集成调试器 ID850QB 作为控制软件。

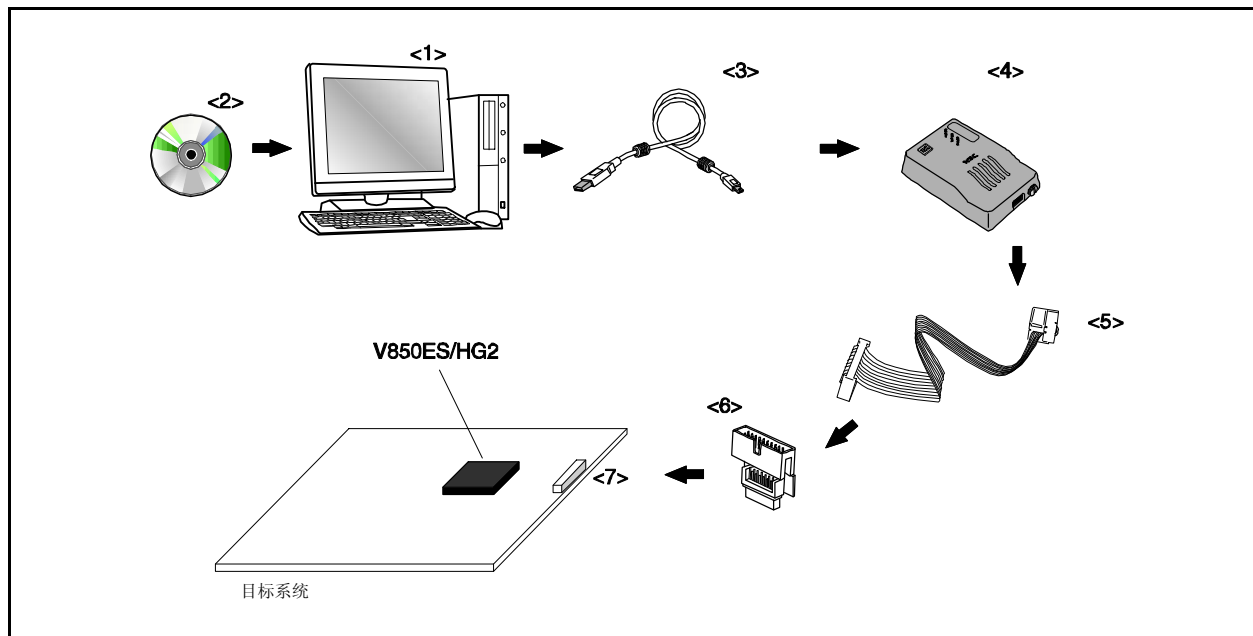
备注 角括号中的数字对应图 A-2 中的数字。

A.4.2 当使用 MINICUBE QB-V850MINI 时

(1) 使用 MINICUBE 的片上调试器

当连接 MINICUBE 到主机（PC-9821 系列，PC/AT 兼容机）时的系统配置如下所示。

图 A-3. 片上调试器系统的配置



<1> 主机	带有 USB 端口的 PC
<2> CD-ROM ^{#1}	CD-ROM 包括集成调试器 ID850QB, N 道线检测器, 设备驱动, 以及文件, 并自带 MINICUBE。
<3> USB 接口电缆	USB 线连接主机及 MINICUBE, USB 线自带 MINICUBE。线长大约为 2m.。
<4> MINICUBE 片上调试模拟器	当使用 V850ES/HG2 开发应用系统时, 该片上调试模拟器用于调试软硬件。其自带集成调试器 ID850QB。
<5> OCD 电缆	用于连接 MINICUBE 和目标系统的电缆 与 MINICUBE 一起提供。电缆的长度大约为 20 厘米。
<6> 连接器转换板 KEL 适配器	该转换板由 MINICUBE 提供。
<7> MINICUBE 连接器 KEL 连接器 ^{#2}	8830E-026-170S (由 MINICUBE 提供) 8830E-026-170L (单独销售)

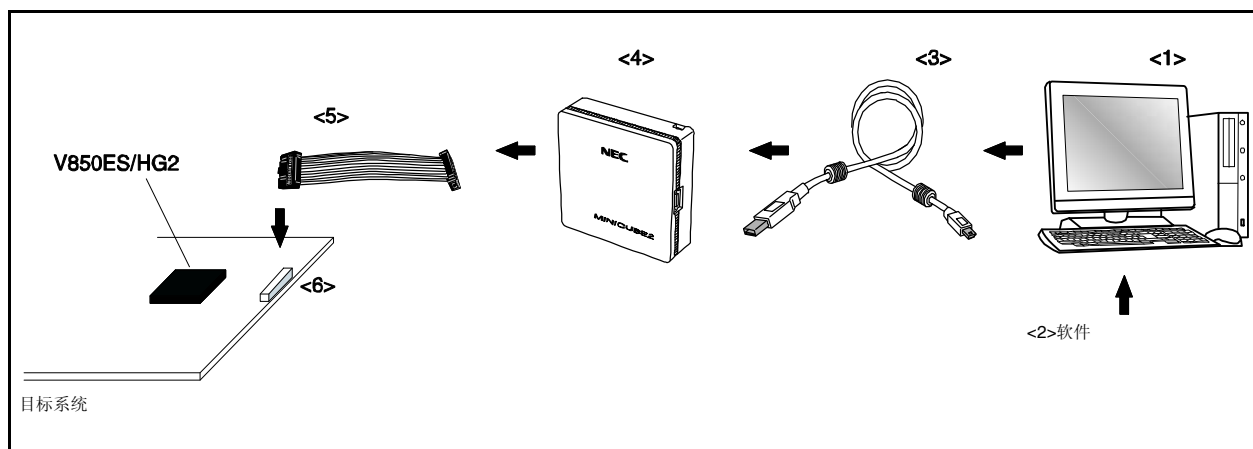
- 注
1. 从 NEC Electronics 网站下载设备文件。
<http://www.necel.com/micro/ods/eng/index.html>
 2. KEL 公司的产品

备注 角括号中的数字对应图 A-3。

A.4.3 当使用 MINICUBE2 QB-MINI2 时

当把 MINICUBE2 和主机（PC-9821 系列，PC/AT 兼容机）连接时，系统配置显示如下。

图 A-4. 片上仿真器系统的系统配置



<1> 主机	使用 USB 端口的 PC 机
<2> 软件	集成调试器 ID850QB, 设备文件, 等等。 从 NEC Electronics 网站下载设备文件。 http://www.necel.com/micro/ods/eng/
<3> USB 接口电缆	使用 USB 电缆连接主机和 MINICUBE2。它提供了 MINICUBE2。电缆长度接近 2m。
<4> MINICUBE2 片上调试仿真器	当使用 V850ES/HG2 开发应用系统时, 该片上调试仿真器提供调试硬件和软件。它支持集成调试器 ID850QB。
<5> 16 引脚目标电缆	使用电缆连接 MINICUBE2 和目标系统。 它提供了 MINICUBE2。电缆长度接近 15 cm。
<6> 目标连接器 (独立销售)	使用截距为 2.54mm 的 16 引脚的多用连接器。

备注 角括号中的数字对应图 A-4 中的数字。

A.5 调试工具（软件）

V850ES/Hx2 的 SM+ [®] 系统模拟器 （开发中）	该系统模拟器支持 V850 系列。V850ES/Hx2 和 SM+ 是基于 Windows 的软件。当模拟目标系统操作时用于在主机上调试 C 源程序或汇编程序。使用 V850ES/Hx2 和 SM+ 允许执行逻辑测试应用和基于硬件开发的独立测试，因此提供了很高的开发效率和软件质量。V850ES/Hx2 和 SM+ 应与设备文件一起使用。 产品型号：μSxxxxSM703712-B
ID850QB 集成调试器	此调试器支持 V850 系列在线仿真器。ID850 和 ID850QB 是基于 Windows 的软件。它拥有改良的 C 语言编译调试功能，并且能够通过结合源程序的集成窗口功能，分解显示，和存储器显示来显示追踪源程序的结果。改功能需要使用器件文件（独立销售）。 产品型号：μSxxxx ID703000-QB (ID850QB)

备注 产品型号中的 xxxx 使用的 OS 而变化。

μSxxxxID703000-QB

xxxx	主机	OS	存储介质
AB17	PC-9800 系列，	Windows(日文版)	CD-ROM
BB17	IBM PC/AT 兼容机	Windows (英文版)	

A.6 嵌入式软件

RX850, RX850 Pro 实时操作系统	RX850 和 RX850 Pro 是依照 μ ITRON 3.0 规范的实时操作系统。 提供一个为了产生多个信息表格的工具（配置器）。 RX850 Pro 比 RX850 具有更多的功能。
	产品型号： μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$ (RX850) μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$ (RX850 Pro)
Applilet [®]	这是一个驱动配置，它可自动为 V850ES/HG2 生成采样程序。
RX-FS850 (文件系统)	这是 FAT 文件系统功能。 此文件系统支持 CD-ROM 文件系统功能。 此文件系统用在实时操作系统 RX850 Pro。

注 关于如何获得 Applilet，请咨询 NEC 电子销售代理。

注意事项 为了购买 RX850 或 RX850 Pro，首先需要填写购买申请表并且签署用户协议。

备注 产品型号中的 xxxx 和 $\Delta\Delta\Delta\Delta$ 随主机和使用的 OS 而变化。

μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$

μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$

$\Delta\Delta\Delta\Delta$	产品概述	用于大规模生产的上限
001	评价目标	不用于批量生产
100K	大规模生产目标	10 万单元
001M		100 万单元
010M		1000 万单元
S01	源程序	量产的目标源程序

xxxx	主机	OS	存储介质
AB17	PC-9800 系列, IBM PC/AT 兼容机	Windows (日文版)	CD-ROM
BB17		Windows (英文版)	
3K17	SPARC 站	Solaris (参考 2.5.1)	

A.7 Flash 存储器写入工具

Flashpro IV (产品型号: PG-FP4) Flash 编程器	Flash 编程器专用于有片上 flash 存储器的微控制器。
QB-MINI2 (MINICUBE2)	带有编程功能的片上调试仿真器。
FA-100GC-8EU-A Flash 存储器写入适配器	Flash 存储器写入适配器用于连接 Flashpro IV, 等。(没有配线)
FA-70F3707GC-8EA-MX Flash 存储器写入适配器	Flash 存储器写入适配器用于连接 Flashpro IV, 等。(已经配线)

备注 FA-100GC-8EU-A 和 FA-3707GC-8EA-MX 是 Naito Densai Machida Mfg. Co., Ltd 的产品。
TEL: +81-42-750-4172

附录 B 寄存器索引

(1/9)

符号	名称	单元	页码
ADA0CR0	A/D 转换结果寄存器 0	ADC	404
ADA0CR0H	A/D 转换结果寄存器 0H	ADC	404
ADA0CR1	A/D 转换结果寄存器 1	ADC	404
ADA0CR10	A/D 转换结果寄存器 10	ADC	404
ADA0CR10H	A/D 转换结果寄存器 10H	ADC	404
ADA0CR11	A/D 转换结果寄存器 11	ADC	404
ADA0CR11H	A/D 转换结果寄存器 11H	ADC	404
ADA0CR12	A/D 转换结果寄存器 12	ADC	404
ADA0CR12H	A/D 转换结果寄存器 12H	ADC	404
ADA0CR13	A/D 转换结果寄存器 13	ADC	404
ADA0CR13H	A/D 转换结果寄存器 13H	ADC	404
ADA0CR14	A/D 转换结果寄存器 14	ADC	404
ADA0CR14H	A/D 转换结果寄存器 14H	ADC	404
ADA0CR15	A/D 转换结果寄存器 15	ADC	404
ADA0CR15H	A/D 转换结果寄存器 15H	ADC	404
ADA0CR1H	A/D 转换结果寄存器 1H	ADC	404
ADA0CR2	A/D 转换结果寄存器 2	ADC	404
ADA0CR2H	A/D 转换结果寄存器 2H	ADC	404
ADA0CR3	A/D 转换结果寄存器 3	ADC	404
ADA0CR3H	A/D 转换结果寄存器 3H	ADC	404
ADA0CR4	A/D 转换结果寄存器 4	ADC	404
ADA0CR4H	A/D 转换结果寄存器 4H	ADC	404
ADA0CR5	A/D 转换结果寄存器 5	ADC	404
ADA0CR5H	A/D 转换结果寄存器 5H	ADC	404
ADA0CR6	A/D 转换结果寄存器 6	ADC	404
ADA0CR6H	A/D 转换结果寄存器 6H	ADC	404
ADA0CR7	A/D 转换结果寄存器 7	ADC	404
ADA0CR7H	A/D 转换结果寄存器 7H	ADC	404
ADA0CR8	A/D 转换结果寄存器 8	ADC	404
ADA0CR8H	A/D 转换结果寄存器 8H	ADC	404
ADA0CR9	A/D 转换结果寄存器 9	ADC	404
ADA0CR9H	A/D 转换结果寄存器 9H	ADC	404
ADA0M0	A/D 转换模式寄存器 0	ADC	399
ADA0M1	A/D 转换模式寄存器 1	ADC	401
ADA0M2	A/D 转换模式寄存器 2	ADC	402
ADA0PFM	掉电比较模式寄存器	ADC	406
ADA0PFT	掉电比较阈值寄存器	ADC	406
ADA0S	A/D 转换通道指定寄存器 0	ADC	403
ADIC	中断控制寄存器	INTC	547
CB0CTL0	CSIB0 控制寄存器 0	CSI	463
CB0CTL1	CSIB0 控制寄存器 1	CSI	466
CB0CTL2	CSIB0 控制寄存器 2	CSI	467

符号	名称	单元	页码
CB0RIC	中断控制寄存器	INTC	547
CB0RX	CSIB0 接收数据寄存器	CSI	462
CB0RXL	CSIB0 接收数据寄存器 L	CSI	462
CB0STR	CSIB0 状态寄存器	CSI	469
CB0TIC	中断控制寄存器	INTC	547
CB0TX	CSIB0 发送数据寄存器	CSI	462
CB0TXL	CSIB0 发送数据寄存器 L	CSI	462
CB1CTL0	CSIB1 控制寄存器 0	CSI	463
CB1CTL1	CSIB1 控制寄存器 1	CSI	466
CB1CTL2	CSIB1 控制寄存器 2	CSI	467
CB1RIC	中断控制寄存器	INTC	547
CB1RX	CSIB1 接收数据寄存器	CSI	462
CB1RXL	CSIB1 接收数据寄存器 L	CSI	462
CB1STR	CSIB1 状态寄存器	CSI	469
CB1TIC	中断控制寄存器	INTC	547
CB1TX	CSIB1 发送数据寄存器	CSI	465
CB1TXL	CSIB1 发送数据寄存器 L	CSI	465
CCLS	CPU 操作时钟状态寄存器	CG	162
CLM	时钟监视模式寄存器	CLM	600
CTBP	CALLT 基础指针	CPU	51
CTPC	CALLT 执行状态保存寄存器	CPU	50
CTPSW	CALLT 执行状态保存寄存器	CPU	50
DADC0	DMA 寻址控制寄存器 0	DMA	513
DADC1	DMA 寻址控制寄存器 1	DMA	513
DADC2	DMA 寻址控制寄存器 2	DMA	513
DADC3	DMA 寻址控制寄存器 3	DMA	513
DBC0	DMA 传送计数寄存器 0	DMA	512
DBC1	DMA 传送计数寄存器 1	DMA	512
DBC2	DMA 传送计数寄存器 2	DMA	512
DBC3	DMA 传送计数寄存器 3	DMA	512
DBPC	异常/调试陷阱状态保存寄存器	CPU	51
DBPSW	异常/调试陷阱状态保存寄存器	CPU	51
DCHC0	DMA 通道控制寄存器 0	DMA	514
DCHC1	DMA 通道控制寄存器 1	DMA	514
DCHC2	DMA 通道控制寄存器 2	DMA	514
DCHC3	DMA 通道控制寄存器 3	DMA	514
DDA0H	DMA 目的地址寄存器 0H	DMA	511
DDA0L	DMA 目的地址寄存器 0L	DMA	511
DDA1H	DMA 目的地址寄存器 1H	DMA	511
DDA1L	DMA 目的地址寄存器 1L	DMA	511
DDA2H	DMA 目的地址寄存器 2H	DMA	511
DDA2L	DMA 目的地址寄存器 2L	DMA	511
DDA3H	DMA 目的地址寄存器 3H	DMA	511
DDA3L	DMA 目的地址寄存器 3L	DMA	511
DMAIC0	中断控制寄存器	INTC	547

符号	名称	单元	页码
DMAIC1	中断控制寄存器	INTC	547
DMAIC2	中断控制寄存器	INTC	547
DMAIC3	中断控制寄存器	INTC	547
DSA0H	DMA 源地址寄存器 0H	DMA	510
DSA0L	DMA 源地址寄存器 0L	DMA	510
DSA1H	DMA 源地址寄存器 1H	DMA	510
DSA1L	DMA 源地址寄存器 1L	DMA	510
DSA2H	DMA 源地址寄存器 2H	DMA	510
DSA2L	DMA 源地址寄存器 2L	DMA	510
DSA3H	DMA 源地址寄存器 3H	DMA	510
DSA3L	DMA 源地址寄存器 3L	DMA	510
DTFR0	DMA 触发因素寄存器 0	DMA	515
DTFR1	DMA 触发因素寄存器 1	DMA	515
DTFR2	DMA 触发因素寄存器 2	DMA	515
DTFR3	DMA 触发因素寄存器 3	DMA	515
ECR	中断源寄存器	CPU	48
EIPC	中断状态保存寄存器	CPU	47
EIPSW	中断状态保存寄存器	CPU	47
FEPC	NMI 状态保存寄存器	CPU	48
FEPSW	NMI 状态保存寄存器	CPU	48
IMR0	中断屏蔽寄存器 0	INTC	547
IMR0H	中断屏蔽寄存器 0H	INTC	547
IMR0L	中断屏蔽寄存器 0L	INTC	547
IMR1	中断屏蔽寄存器 1	INTC	547
IMR1H	中断屏蔽寄存器 1H	INTC	547
IMR1L	中断屏蔽寄存器 1L	INTC	547
IMR2	中断屏蔽寄存器 2	INTC	547
IMR2H	中断屏蔽寄存器 2H	INTC	547
IMR2L	中断屏蔽寄存器 2L	INTC	547
IMR3	中断屏蔽寄存器 3	INTC	547
IMR3H	中断屏蔽寄存器 3H	INTC	547
IMR3L	中断屏蔽寄存器 3L	INTC	547
INTF0	外部中断下降沿指定寄存器 0	INTC	561
INTF1	外部中断下降沿指定寄存器 1	INTC	562
INTF3	外部中断下降沿指定寄存器 3	INTC	563
INTF3H	外部中断下降沿指定寄存器 3H	INTC	563
INTF3L	外部中断下降沿指定寄存器 3L	INTC	563
INTF9H	外部中断下降沿指定寄存器 9H	INTC	564
INTR0	外部中断上升沿指定寄存器 0	INTC	561
INTR1	外部中断上升沿指定寄存器 1	INTC	562
INTR3	外部中断上升沿指定寄存器 3	INTC	563
INTR3H	外部中断上升沿指定寄存器 3H	INTC	563
INTR3L	外部中断上升沿指定寄存器 3L	INTC	563
INTR9H	外部中断上升沿指定寄存器 9H	INTC	564
ISPR	当前服务的优先级寄存器	INTC	551

符号	名称	单元	页码
KRIC	中断控制寄存器	INTC	547
KRM	按键返回模式寄存器	KR	570
LOCKR	锁定寄存器	CG	165
LVIIC	中断控制寄存器	INTC	547
LVIM	低电压探测寄存器	LVI	607
LVIS	低电压探测电平选择寄存器	LVI	608
NFC	噪声消除控制寄存器	INTC	565
OCDM	片上调试模式寄存器	DCU	644
OSTS	振荡稳定时间选择寄存器	WDT	575
P0	端口 0	端口	83
P00NFC	TIP00 引脚噪声消除控制寄存器	定时器	185
P01NFC	TIP01 引脚噪声消除控制寄存器	定时器	185
P1	端口 1	端口	87
P10NFC	TIP10 引脚噪声消除控制寄存器	定时器	185
P11NFC	TIP11 引脚噪声消除控制寄存器	定时器	185
P20NFC	TIP20 引脚噪声消除控制寄存器	定时器	185
P21NFC	TIP21 引脚噪声消除控制寄存器	定时器	185
P3	端口 3	端口	90
P30NFC	TIP30 引脚噪声消除控制寄存器	定时器	185
P31NFC	TIP31 引脚噪声消除控制寄存器	定时器	185
P3H	端口 3H	端口	90
P3L	端口 3L	端口	90
P4	端口 4	端口	96
P5	端口 5	端口	99
P7	端口 7	端口	105
P7H	端口 7H	端口	105
P7L	端口 7L	端口	105
P9	端口 9	端口	107
P9H	端口 9H	端口	107
P9L	端口 9L	端口	107
PC	程序计数器	CPU	45
PCC	处理时钟控制寄存器	CG	158
PCLM	可编程时钟模式寄存器	CG	167
PCM	端口 CM	端口	115
PCS	端口 CS	端口	117
PCT	端口 CT	端口	119
PDL	端口 DL	端口	121
PDLH	端口 DLH	端口	121
PDLL	端口 DLL	端口	121
PEMU1	外围仿真寄存器 1	LVI	613
PFC0	端口功能控制寄存器 0	端口	85
PFC3L	端口功能控制寄存器 3L	端口	92
PFC5	端口功能控制寄存器 5	端口	101
PFC9	端口功能控制寄存器 9	端口	110

符号	名称	单元	页码
PFC9H	端口功能控制寄存器 9H	端口	110
PFC9L	端口功能控制寄存器 9L	端口	110
PFCE3L	端口功能控制扩展寄存器 3L	端口	93
PFCE5	端口功能控制扩展寄存器 5	端口	101
PFCE9	端口功能控制扩展寄存器 9	端口	110
PFCE9H	端口功能控制扩展寄存器 9H	端口	110
PFCE9L	端口功能控制扩展寄存器 9L	端口	110
PIC0	中断控制寄存器	INTC	547
PIC1	中断控制寄存器	INTC	547
PIC10	中断控制寄存器	INTC	547
PIC2	中断控制寄存器	INTC	547
PIC3	中断控制寄存器	INTC	547
PIC4	中断控制寄存器	INTC	547
PIC5	中断控制寄存器	INTC	547
PIC6	中断控制寄存器	INTC	547
PIC7	中断控制寄存器	INTC	547
PIC8	中断控制寄存器	INTC	547
PIC9	中断控制寄存器	INTC	547
PLLCTL	PLL 控制寄存器	CG	164
PLLS	PLL 入锁时间指定寄存器	CG	166
PM0	端口模式寄存器 0	端口	83
PM1	端口模式寄存器 1	端口	87
PM3	端口模式寄存器 3	端口	90
PM3H	端口模式寄存器 3H	端口	90
PM3L	端口模式寄存器 3L	端口	90
PM4	端口模式寄存器 4	端口	96
PM5	端口模式寄存器 5	端口	99
PM7	端口模式寄存器 7	端口	105
PM7H	端口模式寄存器 7H	端口	105
PM7L	端口模式寄存器 7L	端口	105
PM9	端口模式寄存器 9	端口	107
PM9H	端口模式寄存器 9H	端口	107
PM9L	端口模式寄存器 9L	端口	107
PMC0	端口模式控制寄存器 0	端口	84
PMC1	端口模式控制寄存器 1	端口	88
PMC3	端口模式控制寄存器 3	端口	91
PMC3H	端口模式控制寄存器 3H	端口	91
PMC3L	端口模式控制寄存器 3L	端口	91
PMC4	端口模式控制寄存器 4	端口	97
PMC5	端口模式控制寄存器 5	端口	100
PMC9	端口模式控制寄存器 9	端口	108
PMC9H	端口模式控制寄存器 9H	端口	108
PMC9L	端口模式控制寄存器 9L	端口	108
PMCCM	端口模式控制寄存器 CM	端口	115
PMCM	端口模式寄存器 CM	端口	115

符号	名称	单元	页码
PMCS	端口模式寄存器 CS	端口	117
PMCT	端口模式寄存器 CT	端口	119
PMDL	端口模式寄存器 DL	端口	121
PMDLH	端口模式寄存器 DLH	端口	121
PMDLL	端口模式寄存器 DLL	端口	121
PRCMD	命令寄存器	CPU	73
PRSCM0	预分频比较寄存器 0	WT	383, 506
PRSM0	预分频模式寄存器 0	WT	382, 505
PSC	功率节省 控制寄存器	CG	573
PSMR	功率节省 模式寄存器	CG	574
PSW	程序状态字	CPU	49
PU0	上拉电阻选项寄存器 0	端口	85
PU1	上拉电阻选项寄存器 1	端口	88
PU3	上拉电阻选项寄存器 3	端口	94
PU3H	上拉电阻选项寄存器 3H	端口	94
PU3L	上拉电阻选项寄存器 3L	端口	94
PU4	上拉电阻选项寄存器 4	端口	97
PU5	上拉电阻选项寄存器 5	端口	103
PU9	上拉电阻选项寄存器 9	端口	113
PU9H	上拉电阻选项寄存器 9H	端口	113
PU9L	上拉电阻选项寄存器 9L	端口	113
Q00NFC	TIQ00 引脚噪声消除控制寄存器	定时器	285
Q01NFC	TIQ01 引脚噪声消除控制寄存器	定时器	285
Q02NFC	TIQ02 引脚噪声消除控制寄存器	定时器	285
Q03NFC	TIQ03 引脚噪声消除控制寄存器	定时器	285
Q10NFC	TIQ10 引脚噪声消除控制寄存器	定时器	285
Q11NFC	TIQ11 引脚噪声消除控制寄存器	定时器	285
Q12NFC	TIQ12 引脚噪声消除控制寄存器	定时器	285
Q13NFC	TIQ13 引脚噪声消除控制寄存器	定时器	285
r0 to r31	通用寄存器	CPU	45
RAMS	内部 RAM 数据状态寄存器	CG	608
RCM	内部振荡模式寄存器	CG	162
RESF	复位源标志寄存器	CG	592
SELCNT0	选择器操作控制寄存器 0	定时器	262
SYS	系统状态寄存器	CPU	74
TM0CMP0	TMM0 比较寄存器 0	定时器	372
TM0CTL0	TMM0 控制寄存器 0	定时器	373
TM0EQIC0	中断控制寄存器	INTC	547
TP0CCIC0	中断控制寄存器	INTC	547
TP0CCIC1	中断控制寄存器	INTC	547
TP0CCR0	TMP0 比较/比较寄存器 0	定时器	180
TP0CCR1	TMP0 比较/比较寄存器 1	定时器	182
TP0CNT	TMP0 计数器读取缓冲寄存器	定时器	184
TP0CTL0	TMP0 控制寄存器 0	定时器	173
TP0CTL1	TMP0 控制寄存器 1	定时器	174

符号	名称	单元	页码
TP0IOC0	TMP0 I/O 控制寄存器 0	定时器	176
TP0IOC1	TMP0 I/O 控制寄存器 1	定时器	177
TP0IOC2	TMP0 I/O 控制寄存器 2	定时器	178
TP0OPT0	TMP0 选项寄存器 0	定时器	179
TP0OVIC	中断控制寄存器	INTC	547
TP1CCIC0	中断控制寄存器	INTC	547
TP1CCIC1	中断控制寄存器	INTC	547
TP1CCR0	TMP1 捕捉/比较寄存器 0	定时器	180
TP1CCR1	TMP1 捕捉/比较寄存器 1	定时器	182
TP1CNT	TMP1 计数器读取缓冲寄存器	定时器	184
TP1CTL0	TMP1 控制寄存器 0	定时器	173
TP1CTL1	TMP1 控制寄存器 1	定时器	174
TP1IOC0	TMP1 I/O 控制寄存器 0	定时器	176
TP1IOC1	TMP1 I/O 控制寄存器 1	定时器	177
TP1IOC2	TMP1 I/O 控制寄存器 2	定时器	178
TP1OPT0	TMP1 选项寄存器 0	定时器	179
TP1OVIC	中断控制寄存器	INTC	547
TP2CCIC0	中断控制寄存器	INTC	547
TP2CCIC1	中断控制寄存器	INTC	547
TP2CCR0	TMP2 捕捉/比较寄存器 0	定时器	180
TP2CCR1	TMP2 捕捉/比较寄存器 1	定时器	182
TP2CNT	TMP2 计数器读取缓冲寄存器	定时器	184
TP2CTL0	TMP2 控制寄存器 0	定时器	173
TP2CTL1	TMP2 控制寄存器 1	定时器	174
TP2IOC0	TMP2 I/O 控制寄存器 0	定时器	176
TP2IOC1	TMP2 I/O 控制寄存器 1	定时器	177
TP2IOC2	TMP2 I/O 控制寄存器 2	定时器	178
TP2OPT0	TMP2 选项寄存器 0	定时器	179
TP2OVIC	中断控制寄存器	INTC	547
TP3CCIC0	中断控制寄存器	INTC	547
TP3CCIC1	中断控制寄存器	INTC	547
TP3CCR0	TMP3 捕捉/比较寄存器 0	定时器	180
TP3CCR1	TMP3 捕捉/比较寄存器 1	定时器	182
TP3CNT	TMP3 计数器读取缓冲寄存器	定时器	184
TP3CTL0	TMP3 控制寄存器 0	定时器	173
TP3CTL1	TMP3 控制寄存器 1	定时器	174
TP3IOC0	TMP3 I/O 控制寄存器 0	定时器	176
TP3IOC1	TMP3 I/O 控制寄存器 1	定时器	177
TP3IOC2	TMP3 I/O 控制寄存器 2	定时器	178
TP3OPT0	TMP3 选项寄存器 0	定时器	179
TP3OVIC	中断控制寄存器	INTC	547
TQ0CCIC0	中断控制寄存器	INTC	547
TQ0CCIC1	中断控制寄存器	INTC	547
TQ0CCIC2	中断控制寄存器	INTC	547
TQ0CCIC3	中断控制寄存器	INTC	547

符号	名称	单元	页码
TQ0CCR0	TMQ0 捕捉/比较寄存器 0	定时器	276
TQ0CCR1	TMQ0 捕捉/比较寄存器 1	定时器	278
TQ0CCR2	TMQ0 捕捉/比较寄存器 2	定时器	280
TQ0CCR3	TMQ0 捕捉/比较寄存器 3	定时器	282
TQ0CNT	TMQ0 计数器读取缓冲寄存器	定时器	284
TQ0CTL0	TMQ0 控制寄存器 0	定时器	269
TQ0CTL1	TMQ0 控制寄存器 1	定时器	270
TQ0IOC0	TMQ0 I/O 控制寄存器 0	定时器	272
TQ0IOC1	TMQ0 I/O 控制寄存器 1	定时器	273
TQ0IOC2	TMQ0 I/O 控制寄存器 2	定时器	274
TQ0OPT0	TMQ0 选项寄存器 0	定时器	275
TQ0OVIC	中断控制寄存器	INTC	547
TQ1CCIC0	中断控制寄存器	INTC	547
TQ1CCIC1	中断控制寄存器	INTC	547
TQ1CCIC2	中断控制寄存器	INTC	547
TQ1CCIC3	中断控制寄存器	INTC	547
TQ1CCR0	TMQ1 捕捉/比较寄存器 0	定时器	276
TQ1CCR1	TMQ1 捕捉/比较寄存器 1	定时器	278
TQ1CCR2	TMQ1 捕捉/比较寄存器 2	定时器	280
TQ1CCR3	TMQ1 捕捉/比较寄存器 3	定时器	282
TQ1CNT	TMQ1 计数器读取缓冲寄存器	定时器	284
TQ1CTL0	TMQ1 控制寄存器 0	定时器	269
TQ1CTL1	TMQ1 控制寄存器 1	定时器	270
TQ1IOC0	TMQ1 I/O 控制寄存器 0	定时器	272
TQ1IOC1	TMQ1 I/O 控制寄存器 1	定时器	273
TQ1IOC2	TMQ1 I/O 控制寄存器 2	定时器	274
TQ1OPT0	TMQ1 定时器选项寄存器 0	定时器	275
TQ1OVIC	中断控制寄存器	INTC	547
UA0CTL0	UARTA0 控制寄存器 0	UART	430
UA0CTL1	UARTA0 控制寄存器 1	UART	452
UA0CTL2	UARTA0 控制寄存器 2	UART	453
UA0OPT0	UARTA0 选项控制寄存器 0	UART	432
UA0RIC	中断控制寄存器	INTC	547
UA0RX	UARTA0 接收数据寄存器	UART	435
UA0STR	UARTA0 状态寄存器	UART	433
UA0TIC	中断控制寄存器	INTC	547
UA0TX	UARTA0 发送数据寄存器	UART	435
UA1CTL0	UARTA1 控制寄存器 0	UART	430
UA1CTL1	UARTA1 控制寄存器 1	UART	432
UA1CTL2	UARTA1 控制寄存器 2	UART	453
UA1OPT0	UARTA1 选项控制寄存器 0	UART	432
UA1RIC	中断控制寄存器	INTC	547
UA1RX	UARTA1 接收数据寄存器	UART	435
UA1STR	UARTA1 状态寄存器	UART	433
UA1TIC	中断控制寄存器	INTC	547

符号	名称	单元	页码
UA1TX	UARTA1 发送数据寄存器	UART	435
UA2CTL0	UARTA2 控制寄存器 0	UART	430
UA2CTL1	UARTA2 控制寄存器 1	UART	432
UA2CTL2	UARTA2 控制寄存器 2	UART	453
UA2OPT0	UARTA2 选项控制寄存器 0	UART	432
UA2RIC	中断控制寄存器	INTC	547
UA2RX	UARTA2 接收数据寄存器	UART	435
UA2STR	UARTA2 状态寄存器	UART	433
UA2TIC	中断控制寄存器	INTC	547
UA2TX	UARTA2 发送数据寄存器	UART	435
VSWC	系统等待控制寄存器	CPU	75
WDTE	看门狗定时器使能寄存器	WDT	393
WDTM2	看门狗定时器 模式寄存器 2	WDT	391, 552
WTIC	中断控制寄存器	INTC	547
WTIIC	中断控制寄存器	INTC	547
WTM	钟表定时器操作模式寄存器	WT	384

附录 C 指令集列表

C.1 常规指令

(1) 用于描述操作数的寄存器符号

寄存器符号	解释
reg1	通用寄存器：用作源寄存器。
reg2	通用寄存器：主要用作目的寄存器。在有些指令中也用作源寄存器。
reg3	通用寄存器：主要用作存放除法运算结果的余数或乘法运算结果的高 32 位。
bit#3	用于指定位编号的 3 位数据
immX	X 位立即数据
dispX	X 位偏移量数据
regID	系统寄存器编号
vector	用于指定陷阱向量的 5 位数据(00H ~ 1FH)
cccc	表示条件代码的 4 位数据
sp	堆栈指针 (r3)
ep	元素指针 (r30)
listX	X 个寄存器列表

(2) 用于描述操作码的寄存器符号

寄存器符号	解释
R	指定 reg1 或 regID 的 1 位数据码
r	指定 reg2 的 1 位数据码
w	指定 reg3 的 1 位数据码
d	1 位位移数据
l	1 位立即数据 (指示立即数据的高位)
i	1 位立即数据
cccc	表示条件代码的 4 位数据
CCCC	表示 Bcond 指令的条件代码的 4 位数据
bbb	用于指定位编号的 3 位数据
L	用于指定在寄存器列表中的程序寄存器的 1 位数据

(3) 用于描述操作数的寄存器符号

寄存器符号	解释
←	输入到
GR []	通用寄存器
SR []	系统寄存器
zero-extend (n)	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 读数据 b。
store-memory (a, b, c)	写长度为 c 的数据 b 到地址 a。
load-memory-bit (a, b)	读地址 a 的 b 位。
store-memory-bit (a, b, c)	写入 c 到地址 a 的 b 位。
saturated (n)	对 n 执行饱和处理 (n 为二进制补码)。 l 计算结果, 如果, n ≥ 7FFFFFFFH, 则将结果设置为 7FFFFFFFH。 n ≤ 80000000H, 则将结果设置为 80000000H。
result	反映标志位的结果
Byte	字节 (8 位)
Halfword	半个字 (16 位)
Word	字 (32 位)
+	加
-	减
	位连接
×	乘法
÷	除法
%	除法结果余数
AND	逻辑与
OR	逻辑或
XOR	异或
NOT	逻辑非
logically shift left by	逻辑左移
logically shift right by	逻辑右移
arithmetically shift right by	算术右移

(4) 用于执行时钟寄存器符号

寄存器符号	解释
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) 用于描述标志操作的寄存器符号

标识符	解释
(Blank)	不变
0	清零
X	依据结果设置或清零。
R	恢复预先保存的值。

(6) 条件代码

条件代码 (cccc)	条件公式	解释
0 0 0 0	$OV = 1$	溢出
1 0 0 0	$OV = 0$	没有溢出
0 0 0 1	$CY = 1$	进位 低于(小于)
1 0 0 1	$CY = 0$	没有进位 不低于(大于或等于)
0 0 1 0	$Z = 1$	零
1 0 1 0	$Z = 0$	非零
0 0 1 1	$(CY \text{ or } Z) = 1$	不大于(小于或等于)
1 0 1 1	$(CY \text{ or } Z) = 0$	高于(大于)
0 1 0 0	$S = 1$	负
1 1 0 0	$S = 0$	正
0 1 0 1	—	一直(无论什么情况下)
1 1 0 1	$SAT = 1$	饱和
0 1 1 0	$(S \text{ xor } OV) = 1$	小于有符号数
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（按字母顺序）

(1/6)

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
ADD	reg1, reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]	1	1	1	×	×	×	×	
	imm5, reg2	rrrrr010010iiii	GR[reg2]←GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×	
ADDI	imm16, reg1, reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1	×	×	×	×	
AND	reg1, reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]	1	1	1		0	×	×	
ANDI	imm16, reg1, reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)	1	1	1		0	×	×	
Bcond	disp9	dddd1011ddcccc 注 1	如果条件满足 则 PC←PC+sign-extend(disp9)	当条件满足时	2	2	2				
			当条件不满足时	注 2	注 2	注 2					
BSH	reg2, reg3	rrrrr11111100000 wwwww01101000010	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 wwwww01101000000	GR[reg3]←GR[reg2] (7: 0) GR[reg2] (15: 8) GR [reg2] (23: 16) GR[reg2] (31: 24)	1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr, Halfword))	4	4	4					
CLR1	bit#3, disp16[reg1]	10bbb111110RRRRR dddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr, bit#3)) Store-memory-bit(adr, bit#3, 0)	3	3	3					×
	reg2, [reg1]	rrrrr111111RRRRR 000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr, reg2)) Store-memory-bit(adr, reg2, 0)	3	3	3					×
CMOV	cccc, imm5, reg2, reg3	rrrrr111111iiii wwwww011000cccc0	如果条件满足 则 GR[reg3]←sign-extended(imm5) 否则 GR[reg3]←GR[reg2]	1	1	1					
	cccc, reg1, reg2, reg3	rrrrr111111RRRRR wwwww011001cccc0	如果条件满足 则 GR[reg3]←GR[reg1] 否则 GR[reg3]←GR[reg2]	1	1	1					
CMP	reg1, reg2	rrrrr001111RRRRR	result←GR[reg2]-GR[reg1]	1	1	1	×	×	×	×	
	imm5, reg2	rrrrr010011iiii	result←GR[reg2]-sign-extend(imm5)	1	1	1	×	×	×	×	
CTRET		000001111100000 000000101000100	PC←CTPC PSW←CTPSW	3	3	3	R	R	R	R	R
DBRET		000001111100000 000000101000110	PC←DBPC PSW←DBPSW	3	3	3	R	R	R	R	R

助记符	操作数	操作码	操作	执行时钟			标志					
				i	r	l	CY	OV	S	Z	SAT	
DBTRAP		1111100001000000	DBPC←PC+2 (保存的 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	0000011001iiiiL LLLLLLLLLLLL000000	sp←sp+zero-extend(imm5 逻辑左移 2 位 2) GR[reg in list12]←Load-memory(sp, Word) sp←sp+4 逻辑左移 2 位 2	n+1 注 4	n+1 注 4	n+1 注 4						
	imm5, list12, [reg1]	0000011001iiiiL 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 www01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×		
DIVH	reg1, reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1] ^{#6}	35	35	35		×	×	×		
	reg1, reg2, reg3	rrrrr11111RRRRR www01010000000	GR[reg2]←GR[reg2]÷GR[reg1] ^{#6} GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×		
DIVHU	reg1, reg2, reg3	rrrrr11111RRRRR www01010000010	GR[reg2]←GR[reg2]÷GR[reg1] ^{#6} GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×		
DIVU	reg1, reg2, reg3	rrrrr11111RRRRR www01011000010	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 www01101000100	GR[reg3]←GR[reg2](15: 0) GR[reg2] (31: 16)	1	1	1	×	0	×	×		
JARL	disp22, reg2	rrrrr11110dddddd dddddddddddddd0 注 7	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2						
JMP	[reg1]	0000000011RRRRR	PC←GR[reg1]	3	3	3						
JR	disp22	0000011110dddddd dddddddddddddd0 注 7	PC←PC+sign-extend(disp22)	2	2	2						
LD.B	disp16[reg1], reg2	rrrrr11100RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr, Byte))	1	1	注 11						
LD.BU	disp16[reg1], reg2	rrrrr11110bRRRRR ddddddddddddddd1 注 8, 10	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr, Byte))	1	1	注 11						

助记符	操作数	操作码	操作	执行时钟			标志					
				i	r	l	CY	OV	S	Z	SAT	
LD.H	disp16[reg1], reg2	rrrrr111001RRRRR dddddddddddd0 注 8	adr←GR[reg1]+sign-extend(dispatch16) GR[reg2]←sign-extend(Load-memory(adr, Halfword))	1	1	注 11						
LDSR	reg2, regID	rrrrr111111RRRRR 000000000100000 注 12	SR[regID]←GR[reg2]	1	1	1						
			Other than regID = PSW regID = PSW	1	1	1	x	x	x	x	x	
LD.HU	disp16[reg1], reg2	rrrrr111111RRRRR dddddddddddd1 注 8	adr←GR[reg1]+sign-extend(dispatch16) GR[reg2]←zero-extend(Load-memory(adr, Halfword))	1	1	注 11						
LD.W	disp16[reg1], reg2	rrrrr111001RRRRR dddddddddddd1 注 8	adr←GR[reg1]+sign-extend(dispatch16) GR[reg2]←Load-memory(adr, Word)	1	1	注 11						
MOV	reg1, reg2	rrrrr00000RRRRR	GR[reg2]←GR[reg1]	1	1	1						
	imm5, reg2	rrrrr010000iiii	GR[reg2]←sign-extend(imm5)	1	1	1						
	imm32, reg1	00000110001RRRRR iiiiiiiiiiiiiiii llllllllllllllllll	GR[reg1]←imm32	2	2	2						
MOVEA	imm16, reg1, reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1						
MOVHI	imm16, reg1, reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 0 ¹⁶)	1	1	1						
MUL	reg1, reg2, reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] GR[reg2]←GR[reg2]xGR[reg1] 注 14	1	4	5						
	imm9, reg2, reg3	rrrrr111111iiii wwwww01001111100 注 13	GR[reg3] GR[reg2]←GR[reg2]xsign-extend(imm9)	1	4	5						
MULH	reg1, reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2] ^{#6} xGR[reg1] ^{#6}	1	1	2						
	imm5, reg2	rrrrr010111iiii	GR[reg2]←GR[reg2] ^{#6} xsign-extend(imm5)	1	1	2						
MULHI	imm16, reg1, reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] ^{#6} ximm16	1	1	2						
MULU	reg1, reg2, reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] GR[reg2]←GR[reg2]xGR[reg1] 注 14	1	4	5						
	imm9, reg2, reg3	rrrrr111111iiii wwwww0100111110 注 13	GR[reg3] GR[reg2]←GR[reg2]xzero-extend(imm9)	1	4	5						
NOP		0000000000000000	Pass at least one clock cycle doing nothing.	1	1	1						
NOT	reg1, reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])	1	1	1	0	x	x			
NOT1	bit#3, disp16[reg1]	01bbb11110RRRRR dddddddddddd	adr←GR[reg1]+sign-extend(dispatch16) Z flag←Not(Load-memory-bit(adr, bit#3)) Store-memory-bit(adr, bit#3, Z flag)	3 注 3	3 注 3	3 注 3				x		
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr, reg2)) Store-memory-bit(adr, reg2, Z flag)	3 注 3	3 注 3	3 注 3				x		

助记符	操作数	操作码	操作	执行时钟			标志					
				i	r	l	CY	OV	S	Z	SAT	
OR	reg1, reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×		
ORI	imm16, reg1, reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×		
PREPARE	list12, imm5	0000011110iiiiL LLLLLLLLLLLL00001	Store-memory(sp-4, GR[reg in list12], Word) sp←sp-4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 注4	n+1 注4	n+1 注4						
	list12, imm5, sp/imm ^{#15}	0000011110iiiiL LLLLLLLLLLLLff011 imm16/imm32 注 16	Store-memory(sp-4, GR[reg in list12], Word) sp←sp+4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend (imm5) ep←sp/imm	n+2 注4 注17	n+2 注4 注17	n+2 注4 注17						
RETI		000001111100000 000000010100000	如果 PSW.EP=1 则 PC ←EIPC PSW ←EIPSW 否则如果 PSW.NP=1 则 PC ←FEPC PSW ←FEPSW 否则 PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R	R
SAR	reg1, reg2	rrrrr11111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×		
	imm5, reg2	rrrrr010101iiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×		
SASF	cccc, reg2	rrrrr111110cccc 000000100000000	如果条件满足 则 GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H 否则 GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1						
SATADD	reg1, reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×	×
	imm5, reg2	rrrrr010001iiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5))	1	1	1	×	×	×	×	×	×
SATSUB	reg1, reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]-GR[reg1])	1	1	1	×	×	×	×	×	×
SATSUBI	imm16, reg1, reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]-sign-extend(imm16))	1	1	1	×	×	×	×	×	×
SATSUBR	reg1, reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]-GR[reg2])	1	1	1	×	×	×	×	×	×
SETF	cccc, reg2	rrrrr111110cccc 000000000000000	如果条件满足 则 GR[reg2]←00000001H 否则 GR[reg2]←00000000H	1	1	1						

助记符	操作数	操作码	操作	执行时钟			标志					
				i	r	l	CY	OV	S	Z	SAT	
SET1	bit#3, disp16[reg1]	00bbb111110RRRRR dddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr, bit#3)) Store-memory-bit(adr, bit#3, 1)	3 注3	3 注3	3 注3					×	
	reg2, [reg1]	rrrrr11111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr, reg2)) Store-memory-bit(adr, reg2, 1)	3 注3	3 注3	3 注3						×
SHL	reg1, reg2	rrrrr11111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×		
	imm5, reg2	r r r r r 0 1 0 1 1 0 i i i i i	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×		
SHR	reg1, reg2	rrrrr11111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×		
	imm5, reg2	r r r r r 0 1 0 1 0 0 i i i i i	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×		
SLD.B	disp7[ep], reg2	rrrrr0110dddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr, Byte))	1	1	注9						
SLD.BU	disp4[ep], reg2	rrrrr0000110ddd 注 18	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr, Byte))	1	1	注9						
SLD.H	disp8[ep], reg2	rrrrr1000dddddd 注 19	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr, Halfword))	1	1	注9						
SLD.HU	disp5[ep], reg2	rrrrr0000111ddd 注 18, 20	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr, Halfword))	1	1	注9						
SLD.W	disp8[ep], reg2	rrrrr1010dddddd 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]	rrrrr1010dddddd 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 dddddddddddddd 注 8	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr, GR[reg2], Halfword)	1	1	1						
ST.W	reg2, disp16[reg1]	rrrrr111011RRRRR dddddddddddddd 注 8	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr, GR[reg2], Word)	1	1	1						
STSR	regID, reg2	rrrrr11111RRRRR 0000000010000000	GR[reg2]←SR[regID]	1	1	1						

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1, reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]-GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1, reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]-GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	0000000010RRRRR	adr←(PC+2) + (GR [reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr, Halfword))) 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 (保存的 PC) EIPSW ←PSW ECR.EICC ←中断代码 PSW.EP ←-1 PSW.ID ←-1 PC ←0000040H (当中断矢量由 00H 变为 0FH 时) 0000050H (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	×	×	
TST1	bit#3, disp16[reg1]	11bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr, bit#3))	3 注3	3 注3	3 注3					×
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr, reg2))	3 注3	3 注3	3 注3					×
XOR	reg1, reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16, reg1, reg2	rrrrr110101RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7: 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15: 0))	1	1	1					

- 注
1. dddddddd: disp9 的高 8 位。
 2. 如果有重写即刻的 PSW 内容的指令，那么时钟数为 3。
 3. 如果没有等待状态（3 + 读取访问等待状态的数目）。
 4. n 是 12 个负载寄存器的总数。（根据等待状态的数目。同时，如果没有等待状态，n 就是列出的 12 个寄存器的总数。如果 n = 0，当 n = 1 时执行相同的操作。）
 5. RRRRR: 00000 除外
 6. 仅低半字数据有效。
 7. dddddddddddddddddddd: disp22 的高 21 位。
 8. dddddddddddddddd: disp16 的高 15 位。
 9. 根据等待状态的数目（如果没有等待状态为 1）。
 10. b: disp16 的第 0 位。
 11. 根据等待状态的数目（如果没有等待状态为 2）。

- 注** 12. 在该条指令中，为便于记忆，源寄存器简述为 **reg2**，但 **reg1** 域用于操作码。因此，在描述及操作码中的寄存器规格的涵义与其它指令不相同。
- 指定 **rrrrr** = **regID**
指定 **RRRRR** = **reg2**
13. **iiii**: **imm9** 的低 5 位
IIII: **imm9** 的高 4 位
14. 对于通用寄存器 **reg1** 和 **reg3**，不要指定相同的寄存器。
15. **sp/imm**: 由副操作码的位 19 和位 20 指定。
16. **ff** = 00: 加载 **sp** 于 **ep**。
01: 加载符号扩展的 16 位立即数据 (第 47 ~ 32 位) 于 **ep**。
10: 加载 16 位逻辑左移的 16 位立即数据 (第 47 ~ 32 位) 于 **ep**。
11: 加载 32 位立即数(第 63 ~ 32 位) 于 **ep**。
17. 如果 **imm** = **imm32**，**n** + 3 个时钟。
18. **rrrrr**: 00000 除外。
19. **ddddddd**: **disp8** 的高 7 位。
20. **dddd**: **disp5** 的高 4 位。
21. **dddddd**: **disp8** 的高 6 位。

附录 D 注意事项列表

该附录列举了文件中所述的注意事项。

如下表分类（硬件/软件）。

硬件： 微控制器内/外硬件的注意事项

软件： 软件的注意事项，例如寄存器设置或程序

(1/33)

章节	分类	功能	功能细节	注意事项	页码
第一章	硬件	说明	FLMDO	正常模式下，将该引脚与Vss 相连接。	p. 21 <input type="checkbox"/>
			REGC	通过一个4.7 μF（推荐值）的电容器连接REGC引脚到Vss。	p. 21 <input type="checkbox"/>
第二章	软件	引脚功能	NMI	NMI引脚可功能复用为P02引脚。复位后，引脚功能为P02引脚。要使能NMI引脚，设置PMC0.PMC02 位为 1。NMI引脚的初始设置为“无边沿检测”。使用INTF0 和 INTR0 寄存器选择NMI引脚的有效沿。	p. 29 <input type="checkbox"/>
			FLMDO	如果在自编程期间，输入到RESET引脚的噪声超过噪声消除宽度，当一个电容器连接到FLMDO引脚时，根据电容电荷端定时，可能进入flash存储器板级模式。因此，不要将电容器连接到FLMDO引脚。	p. 40 <input type="checkbox"/>
	当开启电源时		注意以下引脚可能暂时输出一个未定义电平，即使在使用电源复位期间。 • P53/KR3/TIQ00/TOQ00/DDO 引脚	p. 42 <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	由于系统寄存器只有一组，如果要实现多重中断服务，就必须用程序对这组寄存器的内容进行保存。	p. 46 <input type="checkbox"/>
			EIPC, FEPC	即使EIPC 或 FEPC，或CTPC的第0位被LDSR指令置为1，在中断服务程序执行后由RETI指令返回到主程序时，第0位仍然被忽略（这是因为PC的第0位固定为0）。将EIPC, FEPC和CTPC设置为偶数值（第0位=0）。	p. 46 <input type="checkbox"/>
			程序空间	由于地址03FFF000H ~ 03FFFFFFH的4 KB范围是片上外围I/O区域，因此指令不能从这个区域取址。因此，要避免执行可能使分支地址计算结果为该范围中值的操作。	p. 54 <input type="checkbox"/>
			片上外围 I/O 区域	当寄存器被一个字操作指令读写时，一个字的空间被分为低16位和高16位，分两次被字节操作先后读写，其中最低2位地址被忽略。	p. 59 <input type="checkbox"/>
				若对能进行字节操作的寄存器进行半字长操作，那么读取该寄存器时，高8位数数据未被指定，将数据写入低8位数据。	p. 59 <input type="checkbox"/>
				未被指定为寄存器的地址为将来扩展保留。当这些地址被访问时，该操作未被指定且操作结果的有效性无法保证。	p. 59 <input type="checkbox"/>
			内部 RAM 区域	如果分支指令在内部RAM区域的上限处发生，那么将不会产生跨越片上外围I/O区域的预取址（无效取址）。	p. 59 <input type="checkbox"/>
向特殊寄存器写入数据	设置IDLE 1模式，IDLE 2模式，STOP 模式，或副STOP 模式（通过将PSC.STP位置1）后，要立即插入5条NOP指令。	p. 72 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第三章	软件	CPU 功能	向特殊寄存器写入数据	当对命令寄存器进行写操作时，系统将不响应中断。存储指令应该是按上述的第<3>和<4>步骤连续执行的。若在步骤<3>和<4>之间执行了其他指令，而这个指令又进行了中断的响应，则将打乱上述的顺序导致设备误操作。	p. 72 <input type="checkbox"/>
				尽管对PRCMD写入的数据为虚数据，但还是请使用与设置特殊寄存器(例子中的<4>)的通用寄存器相同的寄存器来向PRCMD寄存器(例子中的<3>)写入虚数据。这个过程也同样可以用于使用通用寄存器进行寻址的操作中。	p. 72 <input type="checkbox"/>
			SYS 寄存器	如果在对PRCMD寄存器进行写入之后，向SYS寄存器（并非特殊寄存器）的PRERR位写入0，那么PRERR位将被清零（写入命令优先）。	p. 74 <input type="checkbox"/>
				如果在对PRCMD寄存器进行写入之后，又执行了对PRCMD寄存器（并非特殊寄存器）的写入操作，那么PRERR位将被置1。	p. 74 <input type="checkbox"/>
			首次设置的寄存器	在使用V850ES/HG2器件时，请务必首先对以下寄存器进行设置。 <ul style="list-style-type: none"> •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) 	p. 75 <input type="checkbox"/>
			VSWC 寄存器	访问一个片上外围I/O寄存器需要3个时钟周期的时间(不包含等待周期)。V850ES/HG2 需要根据不同的操作频率插入不同的等待周期数。因此，请根据所使用的时钟，将以下对应的数值设置到VSWC寄存器中。	p. 75 <input type="checkbox"/>
		访问特定的片上外围I/O寄存器	以下状态中，禁止访问上述的寄存器。如果这时产生了等待状态，那么只有复位才可退出等待状态。 <ul style="list-style-type: none"> • 当主时钟振荡停止，系统工作于副时钟时 • 当CPU工作于内部振荡器时钟时 	p. 76 <input type="checkbox"/>	
第四章	硬件	端口 功能	端口 功能	尽管1位存储器操作指令操作了1位，该存储器访问8位单元的端口，如果该端口具有输入和输出引脚，则在输入模式下设置的引脚输出锁存器的内容没有定义，甚至引脚不能进行操作。	p. 80 <input type="checkbox"/>
	软件		端口 0	NMI引脚用作P02引脚的复用引脚，在复位后其用作P02引脚。要使能NMI引脚，设置PMC0.PMC02位为1。NMI引脚的初始设置为“无边沿检测”。使用INTF0和INTR0寄存器选择NMI引脚的有效沿。	p. 82 <input type="checkbox"/>
	硬件, 软件			P05引脚的可功能复用为片上调试功能。外部复位后，P05/INTP2/DRST引脚的初始设置为片上调试引脚（DRST）。若使用P05引脚作为端口引脚，而不是片上调试引脚，应采取以下措施。 <1> 将OCDM.OCDM0位（特殊寄存器）清零。 <2> 采取上述措施后，将P05/INTP2/DRST引脚固定到低电平。 当不使用片上调试功能，在采取上述措施之前，输入高电平到DRST引脚会引起故障（CPU死锁）。 在处理P05引脚时要十分小心。 当没有输入高电平到P05/INTP2/DRST引脚时（当该引脚固定为低电平），不需要操作OCDM.OCDM0位。 为下拉电阻（30 kΩ TYP.）连接到P05/INTP2/DRST引脚的缓冲区时，该引脚不必通过外部源固定为低电平。通过将OCDM0位清零，可以断开下拉电阻。	p. 82 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码		
第四章	硬件	端口 功能	端口 0	P00 ~ P06引脚在复用功能的输入模式下具有滞后性，但在端口模式下不具有滞后性。	p. 83	<input type="checkbox"/>	
			PMC0 寄存器	注意事项 无论PMC05位的值如何，当OCDM.OCDM0 位为1时，05/INTP2/DRST 引脚都会变为DRST引脚。	p. 84	<input type="checkbox"/>	
			端口 1	P10 和 P11引脚在复用功能的输入模式下具有滞后性，但在端口模式下不具有滞后性。	p. 86	<input type="checkbox"/>	
			端口 3	P31 ~ P35和 P39引脚在复用功能的输入模式下具有滞后性，但在端口模式下不具有滞后性。	p. 89	<input type="checkbox"/>	
			软件	P3 寄存器	要以8位 或 1位为单位读写P3 寄存器的位 8 ~ 15 ， 需要将他们指定为P3H 寄存器的位 0 ~ 7。	p. 90	<input type="checkbox"/>
				PM3 寄存器	要以8位 或 1位为单位读写PM3寄存器的位 8 ~ 15 ， 需要将他们指定为PM3H寄存器的位 0 ~ 7。	p. 90	<input type="checkbox"/>
				PMC3 寄存器	要以8位 或 1位为单位读写PMC3寄存器的位 8 ~ 15 ， 需要将他们指定为PMC3H寄存器的位 0 ~ 7。	p. 91	<input type="checkbox"/>
					INTP8引脚用作RXDA2引脚的复用引脚，为了作为RXDA2引脚使用，使复用功能INTP8引脚的边沿检测功能无效（通过固定INTF3.INTF39位和INTR3.INTR39位为0）。为了作为INTP8引脚使用，停止UARTA2的接收操作（通过将UA2CTL0.UA2RXE位清零）。	p. 91	<input type="checkbox"/>
			软件	PMC3 寄存器	INTP7引脚用作RXDA0引脚的复用引脚，为了作为RXDA0引脚使用，使复用功能INTP7引脚的边沿检测功能无效（通过固定INTF3.INTF31位和INTR3.INTR31位为0）。为了作为INTP7引脚使用，停止UARTA0的接收操作（通过将UA0CTL0.UA0RXE位清零）。	p. 92	<input type="checkbox"/>
					PU3 寄存器	要以8位 或 1位为单位读写PU3寄存器的位 8 ~ 15 ， 需要将他们指定为PU3H寄存器的位 0 ~ 7。	p. 94
			硬件	端口 4	P40 和 P42引脚在复用功能的输入模式下具有滞后性，但在端口模式下不具有滞后性。	p. 95	<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。采用如下步骤将DDI, DDO, DCK和DMS引脚作为端口引脚，而不是片上调试引脚使用。 <1> 将OCDM寄存器（特殊寄存器）的OCDM0位清零。 <2> 采取上述措施后，将P05/INTP2/DRST引脚固定为低电平。 当不使用片上调试功能，在采取上述措施之前，输入高电平到DRST引脚会引起故障（CPU死锁）。在处理P05引脚时要十分小心。 当没有输入高电平到P05/INTP2/DRST 引脚时（当该引脚固定为低电平），不需要操作OCDM.OCDM0 位。 因为下拉电阻 (30 kΩ TYP.) 连接到P05/INTP2/DRST 引脚的缓冲区时，该引脚不必通过外部源固定为低电平。通过将OCDM0位清零，可以断开下拉电阻。	p. 98	<input type="checkbox"/>
					P50 ~ P55引脚在复用功能的输入模式下具有滞后性，但在端口模式下不具有滞后性。	p. 98	<input type="checkbox"/>
			软件	PMC5 寄存器	当PFC5.PFC5n和PFCE5.PFCE5n位是默认值（0）时，如果由PMC5寄存器指定控制模式，则输出变成无定义。 因此，首先设置PFC5.PFC5n和PFCE5.PFCE5n位，然后将PMC5n位设置为1以设置控制模式。	pp. 100, 101	<input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码
第四章	软件	端口 功能	端口5复用功能的规格	KRn引脚用作TIQ0m引脚的复用引脚，为了作为TIQ0m引脚使用，使复用功能KRn引脚的按键返回检测功能无效（通过将KRM.KRMn位为0）。为了作为KRn引脚使用，使复用功能TIQ0m引脚的边沿检测功能无效(n = 0 ~ 3, m = 0 ~ 3)。	p. 102 <input type="checkbox"/>
			P7H 寄存器, P7L 寄存器	在A/D 转换期间不要读取P7H 和 P7L 寄存器。	p. 105 <input type="checkbox"/>
			PM7H 寄存器, PM7L 寄存器	当使用 P7n 引脚的复用功能时，需要将PM7n 位置1。	p. 105 <input type="checkbox"/>
	硬件	软件	端口 9	P90 ~ P97, P99和 P913 ~ P915引脚在复用功能的输入模式下具有滞后性，但在端口模式下不具有滞后性。	p. 106 <input type="checkbox"/>
			P9 寄存器	要以8位 或 1位为单位读写P9寄存器的位 8 ~ 15，需要将他们指定为P9H寄存器的位 0 ~ 7。	p. 107 <input type="checkbox"/>
			PM9 寄存器	要以8位 或 1位为单位读写PM9寄存器的位 8 ~ 15，需要将他们指定为PM9H寄存器的位 0 ~ 7。	p. 107 <input type="checkbox"/>
			PMC9 寄存器	当PFC9.PFC9n 位和PFCE9.PFCE9n位是默认值（0）时，如果由PMC9寄存器指定控制模式，则输出变成无定义。因此，首先设置PFC9.PFC9n和PFCE9.PFCE9n位为1，然后将PMC9n位设置为1以设置控制模式。	p. 108 <input type="checkbox"/>
				要以8位 或 1位为单位读写PMC9寄存器的位 8 ~ 15，需要将他们指定为PMC9H寄存器的位 0 ~ 7。	p. 108 <input type="checkbox"/>
			PFC9 寄存器	要以8位 或 1位为单位读写PFC9寄存器的位 8 ~ 15，需要将他们指定为PFC9H寄存器的位 0 ~ 7。	p. 110 <input type="checkbox"/>
			PFCE9 寄存器	要以8位 或 1位为单位读写PFCE9寄存器的位 8 ~ 15，需要将他们指定为PFCE9H寄存器的位 0 ~ 7。	p. 110 <input type="checkbox"/>
			P9引脚控制模式的规格	当PFC9.PFC9n 和 PFCE9.PFCE9n位是默认值（0）时，如果由PMC9寄存器指定控制模式，则输出变成无定义。因此，首先设置PFC9.PFC9n 和 PFCE9.PFCE9n位，然后将PMC9n位设置为1以设置控制模式。	p. 111 <input type="checkbox"/>
				KR7和 RXDA1为一组复用功能引脚。当用作RXDA1引脚时，需要禁止KR7引脚的按键返回检测。（将KRM寄存器的KRM7位清零。）同时，当用作KR7引脚时，推荐将PFC91位置1并且将PFCE91位清零。	p. 112 <input type="checkbox"/>
			PU9 寄存器	要以8位 或 1位为单位读写PU9寄存器的位 8 ~ 15，需要将他们指定为PU9H寄存器的位 0 ~ 7。	p. 113 <input type="checkbox"/>
			端口 DL	由于需要在flash编程模式下使用FLMD1引脚，因此不需要操作端口控制寄存器。关于更多细节，敬请参阅 第二十三章 Flash 存储器 。	p. 120 <input type="checkbox"/>
			PDL 寄存器	要以8位 或 1位为单位读写PDL寄存器的位 8 ~ 15，需要将他们指定为PDLH寄存器的位 0 ~ 7。	p. 121 <input type="checkbox"/>
			PMDL 寄存器	要以8位 或 1位为单位读写PMDL寄存器的位 8 ~ 15，需要将他们指定为PMDLH寄存器的位 0 ~ 7。	p. 121 <input type="checkbox"/>

第 四 章	软件	功能	功能细节	注意事项	页码
第 四 章	软件	端口 功能	从端口模式切换到复用功能模式的注意事项	无论是端口模式/复用功能模式，读写Pn寄存器如下。 <ul style="list-style-type: none"> •读Pn寄存器： 读取端口输出锁存器的值（当PMn.PMnm 位= 0），或读取引脚状态（PMn.PMnm 位= 1）。 •写Pn寄存器： 写入端口输出锁存器 	p. 154 <input type="checkbox"/>
			复用功能模式的注意事项（输入）	当 PMCn.PMCnm 位为0时，由于PMCn寄存器设置值以及引脚电平AND输出，因此输入到复用功能模块的信号为低电平。因而，由于端口设置和复用功能操作使能时序，可能会发生意外的操作。因此，要按照以下顺序从端口模式切换到复用功能 <ul style="list-style-type: none"> • 要从端口模式切换到复用功能模式（输入） 使用 PMCn 寄存器将引脚设置为复用功能模式，然后允许复用功能操作。 • 要从复用功能模式（输入）切换到端口模式 停止复用功能操作然后将引脚切换到端口模式。 	p. 154 <input type="checkbox"/>
第 五 章	软件	时钟发生器 功能	时钟发生器电流	当看门狗定时器2在振荡稳定过程中溢出时，内部振荡器时钟将被选中。	p. 156 <input type="checkbox"/>
			PCC 寄存器	在通过CLKOUT向外输出时钟信号时，不要改变CPU时钟的频率（通过CK3~CK0位）。	p. 159 <input type="checkbox"/>
				对CK3位操作时，请使用位操作指令。如果使用8位操作指令，不要改变CK2~CK0位的值。	p. 159 <input type="checkbox"/>
				当停止主时钟工作时，也要同时停止PLL。另外，工作于主时钟频率的片上外围功能也被停止。	p. 160 <input type="checkbox"/>
				如果不满足下面的条件，那么请改变CK2~CK0位的设置以满足该条件，然后再切换到副时钟工作模式。 内部系统时钟 (fCLK) > 副时钟(fXT) × 4	p. 160 <input type="checkbox"/>
				只有在主时钟振荡稳定之后才可以使工作于主时钟频率的片上外围功能开始工作。如果它们在振荡稳定之前就被启动，则可能引起误操作。	p. 161 <input type="checkbox"/>
			RCM 寄存器	通过设置选项字节，使RCM寄存器的设置是有效的。 关于更多细节，敬请参阅第二十四章 选项字节功能。	p. 162 <input type="checkbox"/>
				CPU工作于内部振荡器时钟时(CCLS.CCLS F 位 = 1)，不可以停止内部振荡器的工作。不要将RSTOP位置1。	p. 162 <input type="checkbox"/>
				即使RSTOP位为1，如果CCLS.CCLS F位被置1（振荡稳定时间中产生WDT溢出时），内部振荡器也会开始振荡。此时，RSTOP位仍为1。	p. 162 <input type="checkbox"/>
			CCLS 寄存器	如果复位释放后在振荡稳定时间中产生WDT的溢出，那么CCLS F位将被置1，并且CCLS寄存器的复位值变为01H。	p. 162 <input type="checkbox"/>
			PLLCTL 寄存器	当PLLON位被清零时，SELPLL位也会被自动清零（进入时钟直通模式）。	p. 164 <input type="checkbox"/>
				只有在PLL时钟频率稳定后，SELPLL位才能够被置1。如果时钟频率没有稳定（未入锁），那么无论向该位写入的数据为何值，SELPLL位都会被写入"0"。	p. 164 <input type="checkbox"/>
			LOCKR 寄存器	LOCK寄存器不能实时反映PLL的锁定状态。置位和清零条件如下。	p. 165 <input type="checkbox"/>
			PLLS 寄存器	将入锁时间设置为800 μs 或更长。	p. 166 <input type="checkbox"/>
入锁期间不要改变PLLS寄存器的设置。	p. 166 <input type="checkbox"/>				
PCLM 寄存器	首先设置端口相关的控制寄存器（PM，PMC，PFC和PFCE寄存器等），然后将PCLE位置1。	p. 167 <input type="checkbox"/>			
	仅在PLL工作期间将PCLE位置1。为了停止PLL，可将PCLE位清零。	p. 167 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第六章	软件	16-位定时器/事件计数器P (TMP)	TPnCTL0 寄存器	在 TPnCE 位 = 0 时才可对 TPnCKS2 ~ TPnCKS0 位进行设置。	p. 173 <input type="checkbox"/>
				当将 TPnCE 位的值从 0 改变为 1 时, 可同时对 TPnCKS2 ~ TPnCKS0 位进行设置。 请务必将第 3 到第 6 位清零。	p. 173 <input type="checkbox"/>
			TPnCTL1 寄存器	对 TPnEST 位的设置只有在外部触发脉冲输出模式或单脉冲输出模式才有效。在其它模式下, 对该位的置 1 操作将被忽略。	p. 174 <input type="checkbox"/>
				请务必将第 3, 4 位清零。	p. 174 <input type="checkbox"/>
				在外部事件模式下, 无论 TPnEEE 位如何设置, 外部事件计数输入都是被选中的。	p. 175 <input type="checkbox"/>
				只有在 TPnCTL0.TPnCE 位 = 0 时才可以对 TPnEEE 和 TQnMD2 ~ TPnMD0 位进行设置 (TPnCE 位 = 1 时, 可以对这些位写入相同的值)。如果在 TPnCE 位 = 1 时对这些位进行了改写, 那么操作结果将不能被保证。若由于误操作引起了改写, 则要将 TQnCE 位清零然后再次设置这些寄存器位。	p. 175 <input type="checkbox"/>
			TPnIOC0 寄存器	在 TPnCTL0.TPnCE 位 = 0 时才可对 TPnOL1, TPnOE1, TPnOL0 和 TPnOE0 位进行改写 (TPnCE 位 = 1 时只可以向这些位写入相同的值)。若在 TPnCE 位 = 1 由于误操作引起了改写, 则要将 TPnCE 位清零然后再次设置这些寄存器位。	p. 176 <input type="checkbox"/>
				在 TPnCE 位和 TPnOEm 位为 0 时, 即便对 TPnOLm 位进行了操作, TOPnm 引脚的输出电平也不能确定。	p. 176 <input type="checkbox"/>
			TPnIOC1 寄存器	在 TPnCTL0.TPnCE 位 = 0 时才可对 TPnIS3 ~ TPnIS0 位进行改写 (TPnCE 位 = 1 时只可以向这些位写入相同的值)。若在 TPnCE 位 = 1 由于误操作引起了改写, 则要将 TPnCE 位清零然后再次设置这些寄存器位。	p. 177 <input type="checkbox"/>
				TPnIS3 ~ TPnIS0 位只有在自由运行定时器模式和脉宽测量模式下才有效。其它模式下, 无法进行捕捉操作。	p. 177 <input type="checkbox"/>
			TPnIOC2 寄存器	在 TPnCTL0.TPnCE 位 = 0 时才可对 TPnEES1, TPnEES0, TPnETS1 和 TPnETS0 位进行改写 (TPnCE 位 = 1 时只可以向这些位写入相同的值)。若在 TPnCE 位 = 1 由于误操作引起了改写, 则要将 TPnCE 位清零然后再次设置这些寄存器位。	p. 178 <input type="checkbox"/>
				TPnEES1 和 TPnEES0 位只有在 TPnCTL1.TPnEEE 位 = 1 或设置了外部事件计数模式 (TPnCTL1.TPnMD2 ~ TPnCTL1.TPnMD0 位 = 001) 时才有效。	p. 178 <input type="checkbox"/>
				TPnETS1 和 TPnETS0 位只有在外部触发脉冲输出模式 (TPnCTL1.TPnMD2 ~ TPnCTL1.TPnMD0 位 = 010) 或单脉冲输出模式 (TPnCTL1.TPnMD2 ~ TPnCTL1.TPnMD0 = 011) 下才有效。	p. 178 <input type="checkbox"/>
			TPnOPT0 寄存器	在 TPnCE 位 = 0 时才可对 TPnCCS1 和 TPnCCS0 位进行改写 (TPnCE 位 = 1 时只可以向这些位写入相同的值)。若在 TPnCE 位 = 1 由于误操作引起了改写, 则要将 TPnCE 位清零然后再次设置这些寄存器位。	p. 179 <input type="checkbox"/>
请务必将第 1~3 位, 第 6 和第 7 位清零。	p. 179 <input type="checkbox"/>				
TPnCCR0 寄存器	以下情况下禁止访问 TPnCCR0 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上外围 I/O 寄存器。 • 当 CPU 工作于副时钟且主时钟振荡停止时 • 当 CPU 工作于内部振荡时钟时	p. 180 <input type="checkbox"/>			
TPnCCR1 寄存器	以下情况下禁止访问 TPnCCR1 寄存器。详细内容请参考 3.4.8(2) 访问特定的片上外围 I/O 寄存器。 • 当 CPU 工作于副时钟且主时钟振荡停止时 • 当 CPU 工作于内部振荡时钟时	p. 182 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第六章	软件	16位定时器/事件计数器P (TMP)	TPnCNT 寄存器	以下情况下禁止访问TPnCNT寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上外围I/O寄存器。 • 当CPU工作于副时钟且主时钟振荡停止时 • 当CPU工作于内部振荡时钟时	p. 184 <input type="checkbox"/>
			PnmNFC 寄存器	请务必将第3到第5位以及第7位清零。 在设置PnmNFC寄存器之前输入到定时器输入引脚(TIPnm)的信号随着信号噪声的消除而输出。 因此, 通过使用PnmNFC寄存器来设置采样时钟(NFC2 ~NFC0)和采样次数(NFSTS), 等待初始时间=(采样时钟)×(采样次数), 并且启动定时器操作。	p. 185 <input type="checkbox"/>
			操作	使用外部事件计数模式时, 请关闭对TIPn0引脚捕捉触发输入的有效边沿的检测(通过将TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位清为“00”)。	p. 186 <input type="checkbox"/>
				使用外部触发脉冲输出模式, 单脉冲输出模式和脉宽测量模式时, 请选择内部时钟作为计数时钟(通过将TPnCTL1.TPnEEE 位清零)。	p. 186 <input type="checkbox"/>
			间隔定时器模式 (TPnMD2 ~ TPnMD0位 = 000)	该寄存器位只有在中断请求信号 (INTTPnCC0 和 INTTPnCC1) 被中断屏蔽标志 (TPnCCMK0 和TPnCCMK1) 屏蔽且同时允许了定时器输出(TOPn1)时, 才可以被置1。此时, 请将相同的值设置到TPnCCR0 和 TPnCCR1 寄存器中。(请参考 6.5.1 (2) (d) TPnCCR1 寄存器的操作)。	p. 188 <input type="checkbox"/>
			改写TPnCCR0寄存器时需要注意的事项	将TPnCCR0寄存器的值减小时, 要先停止计数, 再对设定值进行改变。 如果在计数过程中将TPnCCR0寄存器的值减小, 那么16位计数器可能产生溢出。	p. 193 <input type="checkbox"/>
			外部事件计数模式操作的寄存器设置	当外部时钟用于计数时钟时, 外部时钟可以仅从TIPn0引脚输入。此时, 设置TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0位为00。(捕捉触发器输入 (TIPn0 引脚): 无边沿检测)。	p. 199 <input type="checkbox"/>
			外部事件计数模式的操作时序	在外部事件计数模式中, 不要将TPnCCR0 寄存器设置为0000H。	p. 201 <input type="checkbox"/>
				在外部事件计数模式中, 禁止使用定时器输出功能。如果要将外部事件计数输入作为定时器输出使用, 那么请将工作模式设置为间隔定时器模式并将外部事件计数输入设定为计数时钟 (TPnCTL1.TPnMD2 ~ TPnCTL1.TPnMD0 位 = 000, TPnCTL1.TPnEEE 位 = 1)。	p. 201 <input type="checkbox"/>
			改写TPnCCR0寄存器时需要注意的事项	将TPnCCR0寄存器的值减小时, 要先停止计数, 再对设定值进行改变。 如果在计数过程中将TPnCCR0寄存器的值减小, 那么16位计数器可能产生溢出。	p. 202 <input type="checkbox"/>
			TPnIOC0.TPnOE0, TPnOL0 位	外部触发脉冲输出模式下不使用TOPn0引脚时, 要将该位清零。	p. 207 <input type="checkbox"/>
			操作过程中改变脉冲宽度时需要注意的事项	计数器计数过程中, 若改变PWM波形, 则要在最后进行对TPnCCR1寄存器的写入操作。 对TPnCCR1寄存器进行写入操作之后, 若需再次更改TPnCCRm 寄存器的值, 须等待下一个INTTPnCC0信号被检测到后, 再进行相关操作。	p. 211 <input type="checkbox"/>
TPnIOC0.TPnOE0, TPnOL0位	当单脉冲输出模式下不使用TOPn0引脚时, 要将该位清零。	p. 219 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第六章	软件	16-位定时器/事件计数器P (TMP)	单脉冲输出模式的寄存器设置	如果TPnCCR1寄存器中的设置值比TPnCCR0寄存器中的值大，即使在单脉冲输出模式下单脉冲也不被输出。	p. 220 <input type="checkbox"/>
			改写 TPnCCRM 寄存器时需要注意的事项	将TPnCCRM寄存器的值减小时，要先停止计数，再对设定值进行改变。如果在计数过程中将TPnCCRM寄存器的值减小，那么16位计数器可能产生溢出。	p. 222 <input type="checkbox"/>
			TPnIOC0.TPnOE0, TPnOL0 位	当PWM输出模式下不使用TOPn0引脚时，要将该位清零。	p. 226 <input type="checkbox"/>
			定时器调谐操作功能	通过TPmCTL1.TPmSYE 和 TQ0CTL1.TQ0SYE位可以启用或禁止调谐操作模式，对于TMP2, TMP3和TMQ0之一或者两者都可以指定为从定时器。	p. 257 <input type="checkbox"/>
				使用以下程序设置调谐操作模式。 <1> 设置从定时器的 TPmCTL1.TPmSYE 和 TQ0CTL1.TQ0SYE 位以启动调谐操作。 设置从定时器的TPmCTL1.TPmMD2为TPmCTL1.TPmMD0以及TQ0CTL1.TQ0MD2为TQ0CTL1.TQ0MD0位以进入自由运行模式。 <2> 通过使TPnCTL1.TPnMD2位为TPnCTL1.TPnMD0位可以设置定时器模式。此时，不可以设置主定时器的TPnCTL1.TPnSYE位。 <3> 设置主和从定时器的比较寄存器值。 <4> 设置从定时器的TPmCTL0.TPmCE 和 TQ0CTL0.TQ0CE以启动对内部运行时钟的操作。 <5> 设置主定时器的TPnCTL0.TPnCE位以启动对内部运行时钟的操作。	p. 257 <input type="checkbox"/>
			选择器功能	当使用选择器功能时，要在连接定时器之前设置TMP的捕捉触发输入。	p. 261 <input type="checkbox"/>
				设置选择器功能之前，要先禁止被连接的周边I/O器件（TMP, TMM0, 或UARTA）。	p. 261 <input type="checkbox"/>
			SELCNT0 寄存器	在以下条件下使用INTTM0EQ0中断信号作为TIP01输入信号。 TMM0操作时钟 ≥ TMP0操作时钟 × 4	p. 262 <input type="checkbox"/>
				若将ISEL02~ISEL05位设置为1，则请将这些位所对应的引脚模式设置为捕捉输入模式。	p. 262 <input type="checkbox"/>
				当TMP0, TM1, TM3, TMM0, 和 UARTA0 ~ UARTA2 停止时，再设置ISEL02 ~ ISEL05位。	p. 262 <input type="checkbox"/>
			捕捉操作	若在捕捉操作时使用了一个较慢的时钟信号作为计数时钟，当TPnCE位被设置为1后，立刻检测到捕捉触发信号时，TPnCCR0和TPnCCR1寄存器中捕捉的计数值将不会是0000H而是FFFFH。	p. 263 <input type="checkbox"/>
第七章	软件	16位 定时器/事件计数器 Q (TMQ)	TQnCTL0 寄存器	在 TQnCE 位 = 0时，设置TQnCKS2 ~ TQnCKS0 位。当将TQnCE位的值从0改变为1时，可同时对TQnCKS2 ~ TQ0CKS0位进行设置。 请务必将第3到第6位清零。	p. 269 <input type="checkbox"/>
			TQnCTL1 寄存器	TQnEST位仅在外外部触发脉冲输出模式下或单脉冲输出模式下有效。在其它模式下，对该位的置1操作将被忽略。 请务必将第3和4位清零。	p. 270 <input type="checkbox"/>
			在外部事件计数模式下，无论TQnEEE位如何设置，外部事件计数输入都是被选中的。	p. 271 <input type="checkbox"/>	
				p. 270 <input type="checkbox"/>	
				p. 271 <input type="checkbox"/>	

章节	分类	功能	功能细节	注意事项	页码
第七章	软件	16位定时器/事件计数器 Q (TMQ)	TQnCTL1 寄存器	只有在TQnCTL0.TQnCE 位 = 0时才可以对TQnEEE 以及 TQnMD2 ~ TQnMD0位进行设置 (TQnCE 位 = 1时, 可以对这些位写入相同的值)。如果在TQnCE位 = 1时对这些位进行了改写, 那么操作结果将不能被保证。若由于误操作引起了改写, 则要将TQnCE位清零然后再次设置这些寄存器位。	p. 271 <input type="checkbox"/>
			TQnIOC0 寄存器	只有在TQnCTL0.TQnCE 位 = 0时才可以对TQnOLm 和 TQnOEm位进行改写 (当TQnCE 位 = 1时可以向这些位写入相同的值)。若由于误操作引起了改写, 则要将TQnCE位清零然后再次设置这些寄存器位。 当 TQnCE 和 TQnOEm 位为0时, 即使对TQnOLm位进行了操作, TOQnm 引脚的输出电平也不能确定。	p. 272 <input type="checkbox"/> p. 272 <input type="checkbox"/>
			TQnIOC1 寄存器	只有在 TQnCTL0.TQnCE 位 = 0时才可以对TQnIS7 ~ TQnIS0位进行改写 (TQnCE 位 = 1时只能向这些位写入相同的值)。若由于误操作引起了改写, 则要将TQnCE位清零然后再次设置这些寄存器位。 TQnIS7 ~ TQnIS0位只有在自由运行定时器模式和脉宽测量模式下才有效。其它模式下, 无法进行捕捉操作。	p. 273 <input type="checkbox"/> p. 273 <input type="checkbox"/>
			TQnIOC2 寄存器	在TQnCTL0.TQnCE 位 = 0时才可对TQnEES1, TQnEES0, TQnETS1和TQnETS0位进行改写 (TQnCE 位 = 1时只能向这些位写入相同的值)。若在TQnCE位 = 1由于误操作引起了改写, 则要将TQnCE位清零然后再次设置这些寄存器位。 TQnEES1 和 TQnEES0 位只有在 TQnCTL1.TQnEEE 位 = 1 或设置了外部事件计数模式 (TQnCTL1.TQnMD2 ~ TQnCTL1.TQnMD0位 = 001) 时才有效。 TQnETS1 和 TQnETS0 位只有在外部触发脉冲输出模式 (TQnCTL1.TQnMD2 ~ TQnCTL1.TQnMD0 位 = 010) 或单脉冲输出模式 (TQnCTL1.TQnMD2 ~ TQnCTL1.TQnMD0 = 011) 下才有效。	p. 274 <input type="checkbox"/> p. 274 <input type="checkbox"/> p. 274 <input type="checkbox"/>
			TQnOPT0 寄存器	在TQnCTL0.TQnCE 位 = 0时才可对TQnCCS3和TQnCCS0位进行改写 (TQnCE 位 = 1时只能向这些位写入相同的值)。若由于误操作引起了改写, 则要将TQnCE位清零然后再次设置这些寄存器位。 请务必将第1到3位清零。	p. 275 <input type="checkbox"/> p. 275 <input type="checkbox"/>
			TQnCCR0 寄存器	以下情况下禁止访问TQnCCR0寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上外围I/O寄存器。 • 当CPU工作于副时钟 且主时钟振荡停止时 • 当CPU工作于内部振荡时钟时	p. 276 <input type="checkbox"/>
			TQnCCR1 寄存器	以下情况下禁止访问TQnCCR1寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上外围I/O寄存器。 • 当CPU工作于副时钟 且主时钟振荡停止时 • 当CPU工作于内部振荡时钟时	p. 278 <input type="checkbox"/>
			TQnCCR2 寄存器	以下情况下禁止访问TQnCCR2寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上外围I/O寄存器。 • 当CPU工作于副时钟 且主时钟振荡停止时 • 当CPU工作于内部振荡时钟时	p. 280 <input type="checkbox"/>
			TQnCCR3 寄存器	以下情况下禁止访问TQnCCR3寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上外围I/O寄存器。 • 当CPU工作于副时钟 且主时钟振荡停止时 • 当CPU工作于内部振荡时钟时	p. 282 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码
第七章	软件	16位定时器/事件计数器 Q (TMQ)	TQnCNT 寄存器	以下情况下禁止访问TQnCNT寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上外围I/O寄存器。 ● 当CPU工作于副时钟 且主时钟振荡停止时 ● 当CPU工作于内部振荡时钟时	p. 284 <input type="checkbox"/>
			QnmNFC 寄存器	请务必将第 3 到 5 位和 7 位清零。 在设置QnmNFC寄存器之前输入到定时器输入引脚(TIQnm)的信号随着信号噪声的消除而输出。 因此, 通过使用QnmNFC寄存器来设置采样时钟(NFC2 ~ NFC0)和采样次数(NFSTS), 等待初始时间=(采样时钟) × (采样次数), 并启动定时器操作。	p. 285 <input type="checkbox"/> p. 285 <input type="checkbox"/>
			外部事件计数模式	使用外部事件计数模式时, 请指定为不检测TIQn0 引脚捕捉触发输入有效沿(通过将TQnIOC1.TQnIS1 和 TQnIOC1.TQnIS0 位清为“00”)。	p. 286 <input type="checkbox"/>
			外部触发脉冲输出模式, 单脉冲输出模式, 脉宽测量模式	使用外部触发脉冲输出模式, 单脉冲输出模式和脉宽测量模式时, 请选择内部时钟作为计数时钟(通过将TQnCTL1.TQnEEE 位清零)。	p. 286 <input type="checkbox"/>
			TQnCTL1.TQnEEE 位	该寄存器位只有在中断请求信号 (INTTQnCC0 和 INTTQnCCk) 被中断屏蔽标志 (TQnCCMK0 ~ TQnCCMKk)屏蔽且同时允许了定时器输出(TOQnk)时, 才可以被置1。此时, 请将相同的值设置到TQcCCR0 和TQcCCRk 寄存器中。(请参考 7.5.1 (2) (d) TQ0CCR1 ~ TQ0CCR3 寄存器的操作) (k = 1 ~ 3)。	p. 288 <input type="checkbox"/>
			改写 TQnCCR0 寄存器时需要注意的事项	要将TQnCCR0 寄存器的值改小, 首先停止计数然后再更改设置值。 如果在计数时向 TQnCCR0 寄存器写入更小的值, 16位计数器可能会溢出。	pp. 292, 301 <input type="checkbox"/>
			外部事件计数模式操作的寄存器设置	当外部时钟用于计数时钟时, 外部时钟可以仅从TIQn0引脚输入。此时, 设置TQnIOC1.TQnIS1和 TQnIOC1.TQnIS0位为00。(捕捉触发输入 (TIQn0引脚): 无边沿检测)。	p. 298 <input type="checkbox"/>
			外部事件计数模式	在外部事件计数模式中, 不要将 TQ0CCR0寄存器设置为0000H。 在外部事件计数模式中, 禁止使用定时器输出功能。如果使用外部事件计数输入进行定时器输出, 那么设置间隔定时器模式, 并且通过计数时钟的外部事件计数输入来启用的操作。(TQnCTL1.TQnMD2 ~ TQnCTL1.TQnMD0位 = 000, TQnCTL1.TQnEEE 位 = 1)。	p. 300 <input type="checkbox"/> p. 300 <input type="checkbox"/>
			TQnIOC0.TQnOE0, TQnOLO 位	当TOQn0 引脚不用于外部触发脉冲输出模式时将此位清零。	p. 308 <input type="checkbox"/>
			操作过程中改变脉冲宽度时需要注意的事项	计数器计数过程中, 若改变PWM波形, 则要在最后进行对TQnCCR1寄存器的写入操作。 对TQnCCR1寄存器进行写入操作之后, 若需再次更改TQnCCRk 寄存器的值, 须等待下一个INTTQnCC0信号被检测到后, 再进行相关操作。	p. 312 <input type="checkbox"/>
			TQnIOC0.TQnOE0, TQnOLO 位	当单脉冲输出模式下不使用 TOQn0引脚时, 要将该位清零。	p. 321 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码
第七章	软件	16位定时器/事件计数器 Q (TMQ)	单脉冲输出模式的寄存器设置	如果在TQnCCRk寄存器中的设置值比TQnCCR0寄存器的值大，那么即使在单脉冲输出模式下，单脉冲也不被输出。	p. 322 <input type="checkbox"/>
			改变 TQnCCRM 寄存器时需要注意的事项	将 TQnCCRM 寄存器的值减小时，要先停止计数，再对设定值进行改变。如果在计数过程中将 TQnCCR0 寄存器的值减小，那么16位计数器可能产生溢出。	p. 325 <input type="checkbox"/>
			TQnIC0.TQnOE0, TQnOL0 位	当PWM输出模式下不使用TOQn0引脚时，要将该位清零。	p. 330 <input type="checkbox"/>
			三角波PWM模式 (TQnMD2 ~ TQnMD0 = 111)	在PWM模式中，TQnCCRM 寄存器的捕捉功能未被使用，因为该寄存器仅用于比较寄存器。	p. 364 <input type="checkbox"/>
			定时器调谐操作功能	通过TPmCTL1.TPmSYE 和 TQ0CTL1.TQ0SYE位可以启用或禁止调谐操作模式，对于TMQ2, TMQ3和TMQ0之一或者两者都可以指定为从定时器。	p. 366 <input type="checkbox"/>
				使用以下程序设置调谐操作模式。 <1> 设置从定时器的 TPmCTL1.TPmSYE 和 TQ0CTL1.TQ0SYE 位以启动调谐操作。 设置从定时器的TPmCTL1.TPmMD2为TPmCTL1.TPmMD0以及 TQ0CTL1.TQ0MD2为TQ0CTL1.TQ0MD0位以进入自由运行模式。 <2> 通过使TPnCTL1.TPnMD2位为TPnCTL1.TPnMD0位可以设置定时器模式。此时，不可以设置主定时器的TPnCTL1.TPnSYE位。 <3> 设置主和从定时器的比较寄存器值。 <4> 设置从定时器的TPmCTL0.TPmCE 和 TQ0CTL0.TQ0CE以启动对内部运行时钟的操作。 <5> 设置主定时器的TPnCTL0.TPnCE 位以启动对内部运行时钟的操作。	p. 366 <input type="checkbox"/>
			捕捉操作	若在捕捉操作时使用了一个较慢的时钟信号作为计数时钟，当 TQnCE位被设置为1后，立刻检测到捕捉触发信号时TQnCCR0, TQnCCR1, TQnCCR2, 和 TQnCCR3寄存器中捕捉的计数值将不会是0000H而是FFFFH。	p. 370 <input type="checkbox"/>
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 ~ TM0CKS0位。 当TM0CE 的值从0改变位1时，不可以同时设置TM0CKS2 ~ TM0CKS0位的值。 请务必将第3位到第6位清零。	p. 373 <input type="checkbox"/> p. 373 <input type="checkbox"/>
			间隔定时器模式操作时序	不可以将TM0CMP0 寄存器设置为FFFFH。	pp. 374, 377 <input type="checkbox"/>
			计数开始	根据选择的计数时钟的不同，在TM0CTL0.TM0CE位被设置为1后，采用16位计数器达到以下时间间隔就开始计数。	p. 378 <input type="checkbox"/>
			TM0CMP0, TM0CTL0 寄存器s	TMM0工作时，禁止改写 TM0CMP0 和 TM0CTL0 寄存器。 如果在TM0CE 位 = 1时对这两个寄存器进行改写，那么操作结果将不能被保证。如果这两个寄存器被错误地改写，请将TM0CTL0.TM0CE位清零，然后复位这些寄存器。	p. 378 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码	
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不要改变的BGCS00 和 BGCS01位的值。	p. 382 <input type="checkbox"/>	
				在设置BGCE0位为1之前, 设置PRSM0寄存器。	p. 382 <input type="checkbox"/>	
				根据所使用的主时钟频率设置PRSM0 和PRSCM0 寄存器, 以获得32.768 kHz的fBRG 频率。	p. 382 <input type="checkbox"/>	
			PRSCM0 寄存器	在钟表定时器操作期间, 不要改写PRSCM0寄存器。	p. 383 <input type="checkbox"/>	
				在设置 PRSM0.BGCE0 位为1之前, 设置PRSCM0寄存器。	p. 383 <input type="checkbox"/>	
	硬件			根据所使用的主时钟频率设置PRSM0 和PRSCM0寄存器, 以获得32.768 kHz的fBRG 频率。	p. 383 <input type="checkbox"/>	
				WTM 寄存器	当WTM0 和WTM1位都为0时, 改写WTM2 ~ WTM7位。	p. 385 <input type="checkbox"/>
				注意事项	在允许操作 (WTM.WTM1 和WTM.WTM0位 = 1)后, 第一次钟表定时器中断请求信号(INTWT)产生前需要一段准备时间。	p. 388 <input type="checkbox"/>
					经过0.515625 秒(max.)产生第1个INTWT信号($2^9 \times 1/32768 = 0.015625$ 秒(max.)). 随后INTWT信号每0.5 秒产生1次。	p. 388 <input type="checkbox"/>
				第十章	软件	看门狗定时器2的功能
由不可屏蔽中断请求信号(INTWDT2)引发的不可屏蔽中断服务, 敬请参阅 15.2.2 (2) INTWDT2 信号 。	p. 389 <input type="checkbox"/>					
WDTM2 寄存器	下述状态下, 禁止访问WDTM2 寄存器。关于更多细节, 敬请参阅3.4.8 (2) 访问特定的片上外围I/O寄存器。	p. 391 <input type="checkbox"/>				
	<ul style="list-style-type: none"> 当CPU使用副时钟进行操作并且主时钟振荡停止时 当CPU 使用内部振荡时钟进行操作时 					
	如果通过选项字节功能将OPB1位设置为1(参见 第二十四章), 则复位模式固定。	p. 391 <input type="checkbox"/>				
	关于WDCS20 ~ WDCS24 位的更多细节, 敬请参阅表 10-2 看门狗定时器 2 时钟选择。	p. 391 <input type="checkbox"/>				
	如果WDTM2寄存器在复位后被改写两次, 则必定产生溢出信号, 计数器复位。	p. 391 <input type="checkbox"/>				
	要有意生成一个溢出信号, 仅写入WDTM2寄存器两次或者写入一个非ACH的值到WDTE寄存器一次。 然而, 当看门狗定时器 2被设置为停止操作时, 即使数据被写入WDTM2寄存器仅两次, 或者非"ACH"值被写入WDTE寄存器仅一次, 溢出信号也不会产生。	p. 391 <input type="checkbox"/>				
	要停止看门狗定时器 2的操作, 将RCM.RSTOP 位设置为1(停止内部振荡)并且将1FH写入WDTM2寄存器。如果通过选项字节功能将OPB1位设置为1(参见 第二十四章), 但是, 看门狗定时器 2不能通过复位之外的任何方式停止。	p. 391 <input type="checkbox"/>				
	如果通过选项字节功能将OPB1位设置为1, 则时钟被固定为内部振荡时钟(fr)(可选择 $2^{12}/fr \sim 2^{19}/fr$)。关于更多细节, 敬请参阅 第二十四章 选项字节功能 。	p. 392 <input type="checkbox"/>				
WDTE 寄存器	当非"ACH"值被写入WDTE寄存器时, 必定输出溢出信号。	p. 393 <input type="checkbox"/>				

章节	分类	功能	功能细节	注意事项	页码	
第十章	软件	看门狗定时器2的功能	WDTE 寄存器	当向WDTE寄存器执行1位存储器操作指令时, 必定输出溢出信号。	p. 393 <input type="checkbox"/>	
				要有意生成一个溢出信号, 仅写入WDTM2寄存器两次或者写入一个非ACH的值到WDTE寄存器一次。	p. 393 <input type="checkbox"/>	
				然而, 当看门狗定时器2被设置为停止操作时, 即使数据被写入WDTM2寄存器仅两次, 或者非“ACH”值被写入WDTE寄存器仅一次, 溢出信号也不会产生。		
			WDTE寄存器的读取值为“9AH”(与写入值“ACH”不同)。	p. 393 <input type="checkbox"/>		
第十一章	硬件	A/D转换器	ANI0 ~ ANI15 引脚	确保输入ANI0 ~ ANI15的电压不超过额定值。特别是, 如果大于或等于AVREF0的电压输入某个通道, 这个通道的转换值将不确定, 其它通道的转换值也将受到影响。	p. 398 <input type="checkbox"/>	
				模拟输入引脚(ANI0 ~ ANI15)复用作输入端口引脚(P70 ~ P715)。如果ANI0 ~ ANI15中的任何引脚被选去执行A/D转换, 在转换期间不要执行输入指令到端口7。否则转换分辨率将被降级。	p. 398 <input type="checkbox"/>	
	软件			ADA0M0 寄存器	下述情况下禁止访问ADA0M0寄存器。关于更多细节, 敬请参阅3.4.8 (2) 访问特定的片上外围I/O寄存器。	p. 399 <input type="checkbox"/>
					<ul style="list-style-type: none"> 当CPU使用副时钟, 且主时钟振荡停止时 当CPU使用内部振荡时钟时 	
					第0位的写操作将被忽略。	p. 400 <input type="checkbox"/>
					当A/D转换使能(ADA0CE 位= 1)时, 禁止改变ADA0M1寄存器的值。	p. 400 <input type="checkbox"/>
					在A/D转换期间(ADA0EF 位= 1), 如果ADA0M0, ADA0M2, ADA0S, ADA0PFM和 ADA0PFT寄存器被写入, 将根据模式进行下述操作。	p. 400 <input type="checkbox"/>
					<ul style="list-style-type: none"> 软件触发模式 A/D转换停止并从头开始。 硬件触发模式 A/D转换停止并设置为触发待机状态。 	
					当不使用A/D转换器时, 设置ADA0CE 位为0停止操作以减少功耗。	p. 400 <input type="checkbox"/>
					在A/D转换后输入引脚数据第一次的转换分辨率可能会下降。关于更多细节, 敬请参阅11.6 (7) AVREF0 引脚。	p. 400 <input type="checkbox"/>
	ADA0M1 寄存器	确保将第6 ~ 4位清零。	p. 401 <input type="checkbox"/>			
		保持ADA0HS1 位设置为“1”。	p. 401 <input type="checkbox"/>			
在高速转换模式下的转换时间选择	设置 $3.1 \mu\text{s} \leq \text{转换时间} \leq 15.5 \mu\text{s}$ 。	p. 401 <input type="checkbox"/>				
	禁止改写ADA0M0, ADA0M2, ADA0S, ADA0PFM和 ADA0PFT寄存器及在稳定时间内的触发输入。	p. 401 <input type="checkbox"/>				
ADA0M2 寄存器	确保将第7 ~ 2 清零。	p. 402 <input type="checkbox"/>				
ADA0CRn, ADA0CRnH 寄存器	在下述状态中禁止访问ADA0CRn 和ADA0CRnH 寄存器。关于更多细节, 敬请参阅3.4.8 (2) 访问特定片上外围I/O寄存器。	p. 404 <input type="checkbox"/>				
	<ul style="list-style-type: none"> 当CPU使用副时钟, 且主时钟振荡停止时 当CPU使用内部振荡时钟时 					

章节	分类	功能	功能细节	注意事项	页码
第十一章	软件	A/D 转换器	ADA0CRn, ADA0CRnH 寄存器	ADA0M0 和 ADA0S 寄存器的写操作可能会引起 ADA0CRn 寄存器的内容不确定。转换后, 在写入 ADA0M0 和 ADA0S 寄存器之前读取转换结果。如果不按上述顺序进行, 正确的转换结果可能不会被读取。	p. 404 <input type="checkbox"/>
			ADA0PFM 寄存器	选择模式下, ADA0PFT 寄存器所设置的8位数据与 ADA0CRnH 寄存器的值进行比较, ADA0CRnH 寄存器由 ADA0S 寄存器指定。如果结果与由 ADA0PFC 位指定的条件匹配, 则转换结果将存储在 ADA0CRn 寄存器中并且产生中断信号 INTAD。如果不匹配, 不产生中断信号 INTAD。 扫描模式下, ADA0PFT 寄存器所设置的8位数据与 ADA0CR0H 寄存器的内容相比较。如果结果与由 ADA0PFC 位指定的条件匹配, 则转换结果将存储在 ADA0CR0 寄存器中并且生成中断信号 INTAD。如果不匹配, 不产生中断信号 INTAD。无论比较结果如何, 扫描模式继续工作, 转换结果存储于 ADA0CRn 寄存器中, 直到扫描操作完成。但是, 扫描操作完成后不生成中断信号 INTAD。	p. 406 <input type="checkbox"/>
			当不使用 A/D 转换器时	当不使用 A/D 转换器时, 通过将 ADA0M0.ADA0CE 清零减少功耗。	p. 419 <input type="checkbox"/>
			ANI0 ~ ANI15 引脚的输入范围	输入指定范围的电压值到 ANI0 ~ ANI15 引脚。如果大于等于 AVREF0 或小于等于 AVss (即使在最大绝对值范围内) 的电压被输入到任何一个引脚中, 则那个通道的转换值不确定, 其它通道的转换值也会受到影响。	p. 419 <input type="checkbox"/>
			抑制噪声的方法	为了确保 10 位分辨率, ANI0 ~ ANI15 引脚必须有效地抑制噪声。噪声影响随着模拟输入源的输出阻抗的增加而提高。为了降低噪声, 推荐连接一个外部电容, 如图 11-9 所示。	p. 419 <input type="checkbox"/>
			I/O 复用	模拟输入引脚 (ANI0 ~ ANI15) 可功能复用为端口引脚。当从 ANI0 ~ ANI15 引脚中选择一个执行 A/D 转换, 不要在转换期间执行读取输入端口或写入输出端口指令, 否则转换分辨率将下降。 A/D 转换期间将引脚设置为输出端口引脚同样会使转换分辨率下降。其原因是连接端口引脚的外部电路影响电流流向。 如果某引脚的输入信号正在进行 A/D 转换, 其临近引脚传输一个数字脉冲, 则由于耦合噪声的影响, A/D 转换值可能无法同预期一样。因此, 在 A/D 转换过程中, 确保不使用临近引脚传输脉冲信号。	p. 419 <input type="checkbox"/>
			中断请求标志 (ADIF)	即使 ADA0S 寄存器的内容改变, 中断请求标志 (ADIF) 也不清零。因此, A/D 转换期间模拟输入引脚改变, 上次所选择的模拟输入信号的转换结果可能被存储并且转换结束中断请求标志在 ADA0S 寄存器被改写前立即被设置。如果在 ADA0S 寄存器被改写后立即读取 ADIF 标志, 即使新选择的模拟输入引脚的 A/D 转换还未完成, ADIF 标志也可能被设置。当 A/D 转换停止时, 在再次转换前将 ADIF 标志清零。	p. 420 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码	
第十一章	硬件	A/D 转换器	AV _{REF0} 引脚	(a) AV _{REF0} 引脚用于 A/D 转换器的电源引脚，也用于为复用功能端口供电。如图 11-12 所示，当用于备用电源时，确保向 AV _{REF0} 引脚提供与 V _{DD} 相同的电压。 (b) 如果提供给 AV _{REF0} 引脚的源电源具有高阻抗，或者如果电源低电流负载能力低，参考电压可能被转换中的电流影响（特别是，当转换操作使能位 ADA0CE 刚被设置为 1 之后）。结果，转换精度会下降。如图 11-12 所示，为了避免上述情况，推荐连接一个电容在 AV _{REF0} 和 AV _{SS} 引脚之间以抑制参考电压的波动 (c) 如果提供给 AV _{REF0} 引脚的源电源具有高直流阻抗（例如，由于插入一个二极管），当转换使能时的电压可能比转换停止时的电压低，因为 A/D 转换电流引起电压失真。	p. 421	<input type="checkbox"/>
			读 ADA0CRn 寄存器	当 ADA0M0 ~ ADA0M2, ADA0S, ADA0PFM, 或 ADA0PFT 寄存器进行写操作时，ADA0CRn 寄存器的内容可能不确定。转换完成后，并且在写入 ADA0M0 ~ ADA0M2, ADA0S, ADA0PFM 和 ADA0PFT 寄存器之前，读取转换结果。同时，当外部/定时器触发被识别时，ADA0CRn 寄存器的内容可能不确定。转换完成后，并且在外部/定时器触发被识别前，读取转换结果。在时序不同于如上所述时，正确的转换结果可能不会被读取。	p. 421	<input type="checkbox"/>
	软件	A/D 转换结果	如果在模拟输入引脚和参考电压输入引脚中存在噪声，则噪声可能会造成非法的转换结果。为了避免非法转换结果对系统造成的负面影响，需要使用软件处理。下面显示了软件处理的一个例子。 • 取几次 A/D 转换的平均结果并将其作为 A/D 转换的结果。 • 连续进行几次 A/D 转换，去掉可能得到的异常结果。 • 如果一个 A/D 转换的结果被判定造成了系统的故障，在进行故障处理前必须重新检查系统故障。	p. 421	<input type="checkbox"/>	
		待机模式	因为 A/D 转换器在 STOP 模式下停止操作，转换结果无效，所以功耗可以降低。释放 STOP 模式后，操作恢复，但 STOP 模式释放后的 A/D 转换结果无效。当 STOP 模式释放后使用 A/D 转换器，应在设置 STOP 模式前或释放 STOP 模式后，将 ADA0M0.ADA0CE 位清零，然后在释放 STOP 模式后设置 ADA0CE 位为 1。在 IDLE1, IDLE2, 或副时钟操作模式下，继续操作。因此，为了减小功耗，应将 ADA0M0.ADA0CE 位清零。在 IDLE1 和 IDLE2 模式下，由于模拟输入电压值不能保留，A/D 转换结果在 IDLE1 和 IDLE2 模式释放后无效。在 IDLE1 和 IDLE2 模式被设置前，转换结果有效。	p. 422	<input type="checkbox"/>	
		稳定时间内寄存器和触发器输入的改写	禁止在稳定时间内改写 ADA0M0, ADA0M2, ADA0S, ADA0PFM 及 ADA0PFT 寄存器和触发器输入。	p. 422	<input type="checkbox"/>	

章节	分类	功能	功能细节	注意事项	页码
第十一章	软件	A/D 转换器	A/D 转换结果的漂移	A/D 转换的结果可能会根据电压的波动而发生变化，或者可能受到噪声影响。为了减小漂移，使用多次测量求平均值的方法。	p. 422 <input type="checkbox"/>
	硬件		A/D 转换转换结果滞后性	逐次逼近A/D转换器维持内部采样与保持电容的模拟输入电压，同时进行A/D转换。A/D转换完成后模拟输入电压保留在内部采样与保持电容中。结果产生以下现象。 <ul style="list-style-type: none"> 当同一个通道用于A/D转换，如果电压高于或低于先前A/D转换，则出现滞后性，转换结果受到上次转换值影响。因此，即使在相同电压上进行转换也会产生不同结果。 当变换模拟输入通道时，可能出现滞后性，转换结果受到上次转换通道影响。这是因为A/D转换器用于A/D转换。因此，即使在相同电压上进行转换也会产生不同结果。 	p. 422 <input type="checkbox"/>
第十二章	软件	异步串行接口A (UARTA)	UAnOPT0 寄存器	在SBF接收(UAnSRF位=1)期间，不要设置UAnSRT和UAnSTT位(为1)。	p. 432 <input type="checkbox"/>
			SBF接收	在数据接收期间发送SBF，就会发生帧错误。 在SBF接收(UAnSRF=1)期间，不要设置SBF接收触发位(UAnSRT)和SBF发送触发位(UAnSTT)为1。	p. 442 <input type="checkbox"/> p. 442 <input type="checkbox"/>
			连接发送	在执行连续发送过程中发送初始化，确保UAnSTR.UAnTSF=0，然后进行初始化。当UAnTSF=1时初始化数据，其发送过程不能保证。 在连续发送的情况下，从停止位到下一个数据的开始位的传信率扩大为普通传信率加两个运行时钟。	p. 444 <input type="checkbox"/>
			UART接收	即使产生接收错误，也要确保读取UAnRX寄存器。如果不读取UAnRX寄存器，在接收下一个数据时就会产生过载错误，而且会继续地产生不确定的接收错误。	p. 446 <input type="checkbox"/>
				接收始终按“停止位的个数=1”的情况执行，第2个停止位被忽略。	p. 446 <input type="checkbox"/>
				当接收完成时，在生成接收完成中断请求信号(INTUAnR)后读取UAnRX寄存器，并将UAnPWR或UAnRXE位清零。如果在生成INTUAnR信号之前，将UAnPWR或UAnRXE位清零，则UAnRX寄存器的读取值将不能保证。	p. 446 <input type="checkbox"/>
				如果UARTAn接收完成过程中(产生INTUAnR信号)UAnPWR位=0或UAnRXE位=0产生冲突，无论数据是否被存储到UAnRX寄存器中都产生INTUAnR信号。为了在不等待INTUAnR信号产生的情况下完成接收，在设置中断控制寄存器(UAnRIC)的中断屏蔽标志(UAnRMK)为1，且UAnPWR位=0或UAnRXE位=0之后，确保将UAnRIC寄存器的中断请求标志(UAnRIF)清零。	p. 446 <input type="checkbox"/>
			接收错误	当产生INTUAnR信号，UAnSTR寄存器必须被读取以检测错误。	p. 447 <input type="checkbox"/>
				连续接收过程中产生接收错误中断，在下次接收完成前必须读取UAnSTR寄存器中的内容，并执行错误处理。	p. 448 <input type="checkbox"/>
LIN功能	当使用LIN功能，固定UAnCTL0.UAnPS1和UAnCTL0.UAnPS0位为00。	p. 449 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第十二章	软件	异步串行接口 A (UARTA)	UAnCTL1 寄存器	在改写UAnCTL1 寄存器前, 将UAnCTL0.UAnPWR位清零。	p. 452 <input type="checkbox"/>
			UAnCTL2 寄存器	在改写UAnCTL2 寄存器前, 将UAnCTL0.UAnPWR 位清零或将UAnTXE 和 UAnRXE 位清为00。	p. 453 <input type="checkbox"/>
			波特率错误	发送期间必须保持波特率误差在接收端允许的误差范围内。	p. 454 <input type="checkbox"/>
				接收期间, 波特率误差必须满足“(5)接收期间允许的波特率范围”中所描述的范围。	p. 454 <input type="checkbox"/>
			接收期间允许的波特率范围	必须确保接收期间波特率误差在允许的误差范围内, 可用以下公式计算。	p. 456 <input type="checkbox"/>
			当UARTAn所提供的时钟停止时	当UARTAn所提供的时钟停止工作(例如, 进入IDLE1, IDLE2, 或 STOP模式), 操作停止, 每个寄存器保持时钟停止前的值。TXDAn引脚输出也保持时钟停止前的值。然而, 时钟恢复后, 操作不被保证。因此, 时钟恢复后, 通过设置UAnCTL0.UAnPWR, UAnCTL0.UAnRXEn和UAnCTL0.UAnTXEn为 000, 使电路初始化。	p. 459 <input type="checkbox"/>
			RXDA1 引脚和 KR7 引脚	RXDA1 和KR7 引脚不可以同时使用。要使用RXDA1 引脚, 就不能使用KR7 引脚。要使用KR7 引脚, 就不能使用RXDA1 引脚(推荐设置 PFC91 位为 1 且将 PFCE91 位清零)。	p. 459 <input type="checkbox"/>
			当使用DMA 传送发送数据和接收数据时	UARTAn下, 不产生由通信错误引起的中断。当使用DMA 传送发送数据和接收数据时, 即使传送期间出现错误(校验, 过载, 帧错误), 也不进行处理。可以在DMA传送后读取UAnSTR 寄存器查看是否出错, 或者在通信过程中读取UAnSTR 寄存器查看是否出错。	p. 459 <input type="checkbox"/>
			启动UARTAn	按照下述次序启动UARTAn。 <1> 设置 UAnCTL0.UAnPWR 位为 1。 <2> 设置端口。 <3> 设置 UAnCTL0.UAnTXE 位为 1, UAnCTL0.UAnRXE 位为 1。	p. 459 <input type="checkbox"/>
			停止 UARTAn	按照下述次序停止UARTAn。 <1> 设置UAnCTL0.UAnTXE 位为 0, UAnCTL0.UAnRXE 位为 0。 <2> 设置端口, 设置 UAnCTL0.UAnPWR 位为0(如果端口设置不改变则不需要修改)。	p. 459 <input type="checkbox"/>
			发送模式	发送模式下(UAnCTL0.UAnPWR 位= 1和UAnCTL0.UAnTXE 位= 1), 不要通过软件重写相同值到UAnTX寄存器, 因为写入该寄存器后发送开始。为了连续发送相同的值, 重写相同的值。	p. 459 <input type="checkbox"/>
			连续发送模式	连续发送模式下, 从停止位到下一个启动位的通信速率比正常情况下的2个基本时钟要长。然而, 接收端通过检测启动位初始化时序, 所以结果不受影响。	p. 459 <input type="checkbox"/>
片上调试模式	如果在片上调试(OCD)模式下执行中止命令, 并且UART已接收数据, 则产生过载错误。	p. 459 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第十三章	软件	3线可变量度串行I/O (CSIB)	CBnCTL0 寄存器	为了强行停止发送/接收, 清除CBnPWR位, 而不是CBnRxE位或CBnTXE位。同时, 时钟输出停止。	p. 463 <input type="checkbox"/>
				确保将位3 和2 清零。	p. 465 <input type="checkbox"/>
			CBnCTL1 寄存器	只有当CBnCTL0.CBnPWR = 0或CBnCTL0.CBnTXE 和CBnRxE 位都为0时, 寄存器CBnCTL1可被重写。	p. 466 <input type="checkbox"/>
				设置使通讯时钟 (fcCLK) 小于等于8 MHz 。	p. 466 <input type="checkbox"/>
			CBnCTL2 寄存器	只有当CBnCTL0.CBnPWR = 0或者CBnTXE 和 CBnRxE 位都为0时, 寄存器CBnCTL2可被重写。	p. 467 <input type="checkbox"/>
			连续发送模式 (主模式, 发送模式)	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"/>
			连续传输模式 (从模式, 发送模式)	连续发送模式下, 不产生接收完成中断请求信号(INTCBnR)。	p. 493 <input type="checkbox"/>
			时钟时序	在单独传输模式下, 当 CBnTSF 位设置为1时写入CBnTX寄存器会被忽略。这在传输中对操作无影响。 例如, 中断请求信号INTCBnR产生启动DMA, 如果下一段数据在此时写入CBnTX寄存器, 那么这段数据不会被传输因为CBnTSF 位被设置为1。 执行此项应用时启用连续传输模式而不是单独传输模式。	pp. 502, 503 <input type="checkbox"/>
			PRSM0 寄存器	当钟表定时器和CSIB0 操作时, 不要写入PRSM0寄存器。	p. 505 <input type="checkbox"/>
				在设置BGCE0 位为1前, 设置PRSM0寄存器。	p. 505 <input type="checkbox"/>
			PRSCM0 寄存器	当钟表定时器和CSIB操作时, 不要重写PRSCM0寄存器。	p. 506 <input type="checkbox"/>
				在设置PRSM0.BGCE0 位为1前, 设置PRSCM0寄存器。	p. 506 <input type="checkbox"/>
			波特率发生器	设置通讯时钟(fBRG)频率小于等于8 MHz 。	p. 506 <input type="checkbox"/>
当使用DMA传送接收和发送的数据	当使用DMA传送接收和发送的数据, 即使串行传送期间出现溢出错误, 也不进行处理。可以在DMA传送后读取CBnSTR.CBnOVE 寄存器, 查看是否出现溢出错误。	p. 507 <input type="checkbox"/>			
CBnCTL0 寄存器 CBnCTL1 寄存器 CBnCTL2 寄存器	考虑到寄存器禁止在操作期间(CBnCTL0.CBnPWR为1)重写, 如果操作期间由于错误操作而重写寄存器, 设置CBnCTL0.CBnPWR为0, 然后初始化CSIBn。 禁止在操作期间重写的寄存器如下。 • CBnCTL0 寄存器: CBnTXE, CBnRxE, CBnDIR, CBnTMS 位 • CBnCTL1 寄存器: CBnCKP, CBnDAP, CBnCKS2 ~CBnCKS0 位 • CBnCTL2 寄存器: CBnCL3 ~CBnCL0位	p. 507 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第十三章	软件	3线可变长度串行I/O (CSIB)	传输类型 2 和4	<p>传输类型 2 和4 (CBnCTL1.CBnDAP 位= 1), 产生接收完成中断(INTCBnR)后, SCKBn 时钟执行到一半时CBnSTR.CBnTSF被清零。</p> <p>在单传输模式下, 通讯过程中(CBnTSF = 1)忽略写入的下一个发送数据, 下一次传输不开始。同样, 在只能接收的通讯中 (CBnCTL0.CBnTXE = 0, CBnCTL0.CBnRXE = 1), 如果接收到的数据在传输过程中(CBnTSF = 1)被读取, 则下一次传输不开始。因此, 在传输模式 2或4(CBnDAP = 1)下的单独发送模式, 特别要注意下述问题。</p> <ul style="list-style-type: none"> 要开始下一次发送, 确保CBnTSF = 0, 然后向寄存器CBnTX写入要发送的数据。 只能接收传输模式下(CBnTXE = 0, CBnRXE = 1), 要连续执行下一次接收, 确保CBnTSF = 0, 然后读取寄存器CBnTX。 <p>或者, 使用连续发送模式而不用单独发送模式。推荐在使用DMA时使用连续发送模式。</p>	p. 507 <input type="checkbox"/>
第十四章	软件	DMA 功能 (DMA 控制器)	DSA0 ~ DSA3 寄存器	<p>确保将DSAnH 寄存器的第14~10位清零。</p>	p. 510 <input type="checkbox"/>
				<p>当DMA 传送被禁止时 (DCHCn.Enn 位= 0), 在下列时序下设定DSAnH 和DSAnL 寄存器。</p> <ul style="list-style-type: none"> 从复位后到开始第一个DMA 传送之间的时期 从通过DCHCn.INITn 位将通道初始化后到开始DMA 传送之间的时期 从完成DMA 传送 (DCHCn.TCn 位= 1) 到开始下一个DMA 传送之间的时期 	p. 510 <input type="checkbox"/>
				<p>当DSAn 寄存器中的数值被读取, 两个16位寄存器DSAnH 和DSAnL的值被读取。如果读取操作与数值更新发生冲突, 可能会读取正在被更新的数值 (参见14.13 注意事项)。</p>	p. 510 <input type="checkbox"/>
				<p>复位后, 在开始DMA 传送之前设定DSAnH, DSAnL, DDAAnH, DDAAnL和DBCn 寄存器。否则当DMA 传送开始传送时, 操作无法得到保证。</p>	p. 510 <input type="checkbox"/>
			DDA0 ~ DDA3 寄存器	<p>确保将DDAnH 寄存器的第14~10位清零。</p>	p. 511 <input type="checkbox"/>
				<p>当DMA 传送被禁止时 (DCHCn.Enn 位= 0), 在下列时序下设定DDAnH 和DDAnL 寄存器。</p> <ul style="list-style-type: none"> 从复位后到开始第一个DMA 传送之间的时期 从通过DCHCn.INITn 位将通道初始化后到开始DMA 传送之间的时期 从完成DMA 传送 (DCHCn.TCn 位= 1) 到开始下一个DMA 传送之间的时期 	p. 511 <input type="checkbox"/>
				<p>当DDAn 寄存器中的数值被读取, 两个16位寄存器DDAnH 和DDAnL的值被读取。如果读取操作与数值更新发生冲突, 可能会读取正在被更新的数值 (参见14.13 注意事项)。</p>	p. 511 <input type="checkbox"/>
DBC0 ~ DBC3 寄存器	<p>复位后, 在开始DMA 传送之前设定DSAnH, DSAnL, DDAAnH, DDAAnL和DBCn 寄存器。否则当DMA 传送开始传送时, 操作无法得到保证。</p>	p. 511 <input type="checkbox"/>			
	<p>当DMA 传送被禁止时 (DCHCn.Enn 位= 0), 在下列时序下设定DBCn寄存器。</p> <ul style="list-style-type: none"> 从复位后到开始第一个DMA 传送之间的时期 从通过DCHCn.INITn 位将通道初始化后到开始DMA 传送之间的时期 从完成DMA 传送 (DCHCn.TCn 位= 1) 到开始下一个DMA 传送之间的时期 	p. 512 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第十四章	软件	DMA 功能 (DMA 控制器)	DBC0 ~ DBC3 寄存器	复位后, 在开始DMA 传送之前设定DSAnH, DSAnL, DDAnH, DDAnL和DBCn 寄存器。否则当DMA 传送开始传送时, 操作无法得到保证。	p. 512 <input type="checkbox"/>
			DADC0~ DADC3 寄存器	确保将DADCn寄存器的第15, 13 ~ 8位清零。	p. 513 <input type="checkbox"/>
				当DMA 传送被禁止时 (DCHCn.Enn位= 0), 在下列时序下设定DADCn寄存器。 <ul style="list-style-type: none"> 从复位后到开始第一个DMA 传送之间的时期 传送从通过DCHCn.INITn位将通道初始化后到开始DMA 传送之间的时期 从完成DMA 传送 (DCHCn.TCn位= 1) 到开始下一个DMA 传送之间的时期 	p. 513 <input type="checkbox"/>
				DS0位指定了传送数据的大小, 但不能控制总线宽度。因此如果设定为8位数据 (DS0位=0), 较低位的数据总线并不经常使用。	p. 513 <input type="checkbox"/>
				如果传送数据设定为16位(DS0位= 1), 传送不能从奇数地址开始。传送只能从低地址的首位为0的地址开始。	p. 513 <input type="checkbox"/>
				如果由片上外围端口寄存器 (无论作为发送的源还是目的) 执行DMA传送, 一定要按照寄存器的大小来设定传送数据的大小。例如, 在一个8位寄存器上执行DMA 传送, 一定要指定8位传送。	p. 513 <input type="checkbox"/>
			DCHC0~ DCHC3 寄存器	TCn位为只读位。	p. 514 <input type="checkbox"/>
				INITn 和STGn位为只写位。	p. 514 <input type="checkbox"/>
				确保将DCHCn寄存器的第6~3位清零。	p. 514 <input type="checkbox"/>
			DTFR0 ~ DTFR3 寄存器	当DMA传送完成时 (中止计数产生), Enn位清零然后TCn位设置为1。如果在其数值被更新时读取DCHCn 寄存器, 指示"传送未完成和传送被禁止"(TCn位=0且 Enn位=0)的数值可以被读取。	p. 514 <input type="checkbox"/>
				要通过软件将DFn位设置为 "1"。如果在DMA传送禁止期间, 一个指定作为引起DMA传送的中断发送, 对该位写入0将清除一个DMA传送请求。	p. 515 <input type="checkbox"/>
				当DMA 传送被禁止时 (DCHCn.Enn位= 0), 在下列时序下设定IFCn5 ~ IFCn0 位。 <ul style="list-style-type: none"> 从复位后到开始第一个DMA 传送之间的时期 从通过DCHCn.INITn位将通道初始化后到开始DMA 传送之间的时期 从完成DMA 传送 (DCHCn.TCn位= 1) 到开始下一个DMA 传送之间的时期 	p. 515 <input type="checkbox"/>
				在待机模式下产生的中断 (IDEL1, IDLE2, STOP, 或副IDLE模式) 不会开始DMA 传送周期 (也不会将DFn位设置为1)。	p. 515 <input type="checkbox"/>
				如果通过IFCn5~IFCn0位选择了DMA启动系数, 当被选择的片上外围端口的中断产生时, DFn位被设置为1, 而忽略DMA传送是否被允许。在这种状态下如果允许DMA, 那么DMA传送立即开始。	p. 515 <input type="checkbox"/>
传送目标之间的关系	表14-2中标记为"x"的传送目的和源之间执行操作, 结果将不会得到保证。	p. 518 <input type="checkbox"/>			
通过片上外围 I/O产生请求	两种启动系数 (软件触发和硬件触发) 不能应用在同一个DMA通道。如果两种启动系数在一个DMA通道中同时产生, 则其中仅有一个有效。有效的启动系数不能被识别。	p. 521 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第十四章	软件	DMA 功能 (DMA 控制器)	通过片上外围 I/O 产生请求	在先前的 DMA 传送请求产生后或是先前的 DMA 传送周期后产生的新的传送请求将被忽略（清除）。	p. 521 <input type="checkbox"/>
				相同的 DMA 通道传送请求间隔的变化取决于在 DMA 传送周期中总线等待的设定，其它通道的开始状态或是外部总线保持请求。特别是注意事项 2 中的描述，在 DMA 传送周期之前或是 DMA 传送周期中同一通道产生的新的发送请求将被忽略。因此，相同 DMA 通道的传送请求间隔必须通过系统充分分离。当应用软件触发时，先前产生的 DMA 传送周期完成可以通过更新 DBCn 寄存器检测。	p. 521 <input type="checkbox"/>
			VSWC 寄存器的注意事项	当使用 DMAC 时，根据操作频率一定要给 VSWC 寄存器设置一个合适值。当使用 VSWC 寄存器的默认值 (77H)，或者为 VSWC 寄存器设定的值不恰当时，不会执行正确的操作（关于 VSWC 寄存器的更多细节，敬请参阅 3.4.8 (1) (a) 系统等待控制寄存器 (VSWC)）。	p. 527 <input type="checkbox"/>
			在内部 RAM 中执行 DMA 传送的注意事项	当在内部 RAM 中执行下列指令时，不能在内部 RAM 中（源/目的）执行 DMA 传送，因为 CPU 后来可能不会正确地运行。 • 在内部 RAM 中对于非线性地址的数据访问指令 相反地，当在内部 RAM 中（传送源/目的）执行 DMA 传送时，不要执行上述两种指令。	p. 527 <input type="checkbox"/>
			读取 DCHCn.TCn 位的注意事项	当对 TCn 位进行读取操作时它将被清零，但是即使是在一个特殊的时序，它也并不是自动被清零。为了正确地将 TCn 位清零，要加上下列处理过程。 (a) 当通过检测 TCn 位的方式等待 DMA 传送完成 确认 TCn 位已经被设置为 1（在 TCn 位 = 1 被读取后），然后另外在读取 TCn 位三次。 (b) 当在中断服务程序种读取 TCn 位时 执行读取 TCn 位 3 次。	p. 527 <input type="checkbox"/>
			DMA 传送初始化过程（将 DCHCn.INITn 位设置为 1）	当要执行 DMA 传送的通道被初始化时，即使将 INITn 位设置为 1，该通道也不会被初始化。为了正确地初始化通道，需要执行下列两个过程之一。 (a) 临时中止所有通道的 DMA 传送 初始化正在执行 DMA 传送的通道需要按以下步骤 <1>~<7> 执行。 注，但是当执行第 <5> 步时 TCn 位被清零。确定其它处理程序不要预期 TCn 位为 1。 <1> 禁止中断 (DI)。 <2> 读取被强行中止的 DMA 通道之外的其它 DMA 通道的 DCHCn.Enn 位，然后将这个值传送到通用寄存器。 <3> 将正在被使用的 DMA 通道的 Enn 位（包括被强行中止的通道）清零。为了将最后一个 DMA 通道的 Enn 位清零，需要执行两次清除指令。如果 DMA 传送的目的（传送源/目的）是内部 RAM，需要将该指令执行三次。 例： 如果正在使用通道 0, 1 和 2（如果传送的目的不是内部 RAM），按照以下顺序执行指令。 • 将 DCHC0.E00 位清零。 • 将 DCHC1.E11 位清零。 • 将 DCHC2.E22 位清零。 • 再次将 DCHC2.E22 位清零。 <4> 将被强行中止的 DMA 通道的 INITn 位设置为 1。 <5> 读取除被强行中止的通道外的其它通道的 TCn 位。如果在第 <2> 步中读取的 TCn 位和 Enn 位的值都为 1（逻辑与的结果为 1），将保存的 Enn 位清零。 <6> 在第 <5> 步操作之后，将 Enn 位的值写入 DCHCn 寄存器。 <7> 允许中断 (EI)。	p. 528 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码
第十四章	软件	DMA 功能 (DMA 控制器)	DMA 传送初始化过程 (将 DCHCn.INITn 位设置为1)	一定要执行上述的第<5>步来防止非法的对完成第<2>和<3>步DMA传送正常操作的通道的Enn位的设置。	p. 528 <input type="checkbox"/>
				(b) 重复执行设定INITn 位直到传送被正确的强行中止 <1> 抑制要被强行中止的通道的DMA请求源发出的请求 (停止片上外围端口的操作)。 <2> 通过使用DTFRn.DFn位来检测要被强行中止的通道的DMA传送请求未处于保持状态。如果一个DMA传送请求处于保持状态, 需要等待直到执行完成该保持请求。 <3> 如果确定要被强行中止的通道的DMA请求未处于保持状态, 将Enn位清零。 <4> 再次将要被强行中止的通道的Enn位清零。 如果传送的目的是要被强行中止的通道 (传送源/目的) 是内部RAM, 再次执行该操作。 <5> 拷贝要被强行中止的通道的传送的初始值到通用寄存器。 <6> 将要被强行中止的通道的INITn位设置为1。 <7> 读取要被强行中止的通道的DBCn寄存器的值, 并与第<5>步中的拷贝值相比较。如果两者不匹配, 重复第<6>步和第<7>步的操作。	p. 529 <input type="checkbox"/>
			临时停止DMA传送的程序 (清除 Enn 位)	停止和恢复DMA传送需要按以下步骤执行。 <1> 抑制DMA请求源的传送请求 (停止片上外围端口)。 <2> 使用DFn位来检测DMA传送请求未处于保持状态 (检测 DFn位是否为0)。如果保持请求, 等待直到被保持的请求执行完毕。 <3> 如果确认没有DMA传送请求被保持, 将Enn位清零 (该操作停止DMA传送)。 <4> 将Enn 位设置为1恢复DMA 传送。 <5> 恢复被停止的DMA请求源的操作 (开始片上外围端口的操作)。	p. 529 <input type="checkbox"/>
			存储器边界	如果在DMA传送期间传送的源或者目的地址超出了DMA目标的区域, 那么操作将不会得到保证 (内部RAM 或片上外围端口)。	p. 529 <input type="checkbox"/>
			传送非线性数据	不支持在16为总线宽度上传送非线性数据。 如果一个奇数地址被指定为传送的源或者目的, 地址的最少的有意义的位被强制假定为0。	p. 529 <input type="checkbox"/>
		对于CPU的总线仲裁	由于DMA控制器对于总线的控制权优先于CPU, 因此发生在DMA传送期间的CPU访问被保持直到DMA传送周期完成且将总线释放给CPU。 但是, CPU可以访问那些未执行DMA传送的片上外围端口和内部RAM。	p. 530 <input type="checkbox"/>	

章节	分类	功能	功能细节	注意事项	页码
第十四章	软件	DMA 功能 (DMA 控制器)	在DMA操作期间寄存器/位必须进行重新写入	在 DMA 操作未被执行时在下面的时序中设定下列寄存器。 [寄存器s] • DSA _n H, DSA _n L, DDA _n H, DDA _n L, DBC _n , 和 DADC _n 寄存器 • DTFR _n .IFC _n 5 ~ DTFR _n .IFC _n 0 位 [时序设定] • 从复位后到开始第一个DMA传送之间的时期 • 从通道初始化到开始DMA传送的时间 • 在完成第一个DMA传送 (TC _n 位 = 1) 到开始下一个DMA传送之间的时期	p. 530 <input type="checkbox"/>
			DSA _n H 寄存器 DDA _n H 寄存器 DADC _n 寄存器 DCHC _n 寄存器	一定要将下列寄存器的位清零。 • DSA _n H 寄存器的第14~10位 • DDA _n H 寄存器的第14~10位 • DADC _n 寄存器的第15, 13~8和13~0位 • DCHC _n 寄存器的第6~3位	p. 530 <input type="checkbox"/>
			DMA 启动系数	在同一个启动系数上不要启动2个或更多的DMA通道。如果使用同一个系数启动两个或更多的通道, 可能会启动已经设置过的DMA通道, 或具有较低优先级的DMA通道比具有更高优先级的DMA通道先得到确认。操作的正确性将无法得到保证。	p. 530 <input type="checkbox"/>
			读取DSA _n 和 DDA _n 寄存器的值	在 DMA 传送期间, 可以在 DSA _n 和 DDA _n 寄存器中读取更新的中间值(n = 0 ~ 3)。例如, 如果当 DMA 传送源地址(DSA _n 寄存器)为 0000FFFFH 且计数方式为递增(DADC _n .SAD1 和 DADC _n .SAD0 位= 00), 此时读取 DSA _n H 寄存器和 DSA _n L 寄存器, DSA _n 寄存器的值的区别如下所示, 这将取决于在 DSA _n H 寄存器被读取之后 DMA 传送是否被立即执行。 (a) 如果在DSA _n 寄存器被读取时DMA传送没有发生 <1> DSA _n H 寄存器的读取值: DSA _n H = 0000H <2> DSA _n L 寄存器的读取值: DSA _n L = FFFFH (b) 如果在DSA _n 寄存器被读取时发生了DMA传送 <1> DSA _n H 寄存器的读取值: DSA _n H = 0000H <2> 出现DMA传送 <3> 增加DSA _n 寄存器: DSA _n = 00100000H <4> DSA _n L 寄存器的读取值: DSA _n L = 0000H	p. 530 <input type="checkbox"/>
第十五章	软件	中断/异常处理功能	非屏蔽中断	由非屏蔽中断请求信号(INTWDT2)执行的非屏蔽中断服务, 参考 15.2.2 (2) INTWDT2 信号。	p. 535 <input type="checkbox"/>
				在非屏蔽中断服务期间, 当EP位和NP位通过LDSR指令改变时, 在RETI指令执行之后, 为了正确的恢复PC和PSW, 必须在RETI指令执行之前使用LDSR指令将EP位设置为0, NP位设置为1。	p. 538 <input type="checkbox"/>
			可屏蔽中断	在可屏蔽中断服务执行期间, 当通过LDSR指令将EP位和NP位改变时, 为了在RETI指令后正确的恢复PC和PSW, 需要在RETI指令之前执行LDSR指令将EP位和NP位清零。	p. 542 <input type="checkbox"/>
			多次中断	为了执行复合中断服务, 在执行EI指令之前必须保存EIPC寄存器和EIPSW寄存器的值。当从复合中断服务中返回, 在执行DI指令后恢复EIPC和EIPSW的值。	pp. 544 ~ 546 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码
第十五章	软件	中断/异常处理功能	中断控制 寄存器	禁止中断 (DI) 或者屏蔽中断来读取xxICn.xxIFn位。如果在允许中断或是中断未被屏蔽的时候, 读取xxIFn位, 确认中断且读取该位冲突时数据可能有误。	p. 547 <input type="checkbox"/>
				如果确认中断信号后, xxIFn标志通过硬件自动复位。	p. 547 <input type="checkbox"/>
			IMR0 ~ IMR3 寄存器	该设备文件定义了xxICn.xxMKn位为保留字。如果有其它的位使用xxMKn作为名称进行操作, xxICn寄存器的内容会被写入, 而不是IMRm寄存器 (结果, IMRm寄存器的内容也会被覆盖)。	p. 549 <input type="checkbox"/>
				为了以8位或1位为单元读取IMR0~IMR3寄存器的位8~位15, 需要将它们设定为IMR0H~IMR3H寄存器的位0~位7。	p. 550 <input type="checkbox"/>
				设置IMR3寄存器的位15~13和8~5及IMR2寄存器的位7~4为“1”。如果这些位的设定值改变, 操作将不会得到保证。	p. 550 <input type="checkbox"/>
			ISPR 寄存器	如果在中断允许状态(EI)对ISPR寄存器进行读取时确认一个中断, 在寄存器的位通过确认中断的方式被设置之后, 可以读取 ISPR寄存器的值。为了在确认中断之前正确的读取ISPR寄存器的值, 需要禁止中断(DI)。	p. 551 <input type="checkbox"/>
			软件异常处理的恢复	在软件异常处理过程中, 当通过LDSR指令改变EP位和NP位时, 为了在RETI指令执行后正确的恢复PC和PSW, 在RETI指令执行前必须立即执行LDSR指令将EP位置1, NP位清零。	p. 554 <input type="checkbox"/>
			非法操作代码的定义	由于将来有可能分配该指令到一个非法的操作代码, 推荐不要使用它。	p. 556 <input type="checkbox"/>
			非法操作代码定义的恢复	仅在执行非法操作数和DBRET指令的间隔过程中, 可以访问DBPC 和 DBPSW。	p. 557 <input type="checkbox"/>
			调试陷阱的恢复	仅在执行DBTRAP指令和DBRET指令的间隔过程中, 可以访问DBPC 和 DBPSW。	p. 559 <input type="checkbox"/>
			INTF0, INTR0 寄存器	当引脚的功能从外部中断功能(复用功能)转变到端口功能时, 可以进行边沿检测。因此, 首先要将INTF0n位和INTR0n位清除为00, 然后再设定端口模式。	p. 561 <input type="checkbox"/>
				当这些寄存器不被用作NMI或INTP0~INTP3时, 一定要将INTF0n和INTR0n位清除为00。	p. 561 <input type="checkbox"/>
			INTF1, INTR1 寄存器	当引脚的功能从外部中断功能(复用功能)转变到端口功能时, 可以进行边沿检测。因此, 首先要将INTF1n位和INTR1n位清除为00, 然后再设定端口模式。	p. 562 <input type="checkbox"/>
				当这些寄存器不被用作INTP9 和 INTP10时, 一定要将INTF1n 和 INTR1n位清除为00。	p. 562 <input type="checkbox"/>
			INTF3, INTR3 寄存器	当引脚的功能从外部中断功能(复用功能)转变到端口功能时, 可以进行边沿检测。因此, 首先要将INTF3n位和INTR3n位清除为00, 然后再设定端口模式。	p. 563 <input type="checkbox"/>
当INTF3 和 INTR3寄存器的位8~15支持8位或1位读写方式时, 指定它们为INTF3H 和 INTR3H寄存器为位0~7。	p. 563 <input type="checkbox"/>				
当这些寄存器不被用作INTP7和INTP8引脚时, 一定要将INTF3n 和 INTR3n位清除为00。	p. 563 <input type="checkbox"/>				

章节	分类	功能	功能细节	注意事项	页码
第十五章	软件	中断/异常处理功能	INTF9H, INTR9H 寄存器	当引脚的功能从外部中断功能（复用功能）转变到端口功能时，可以进行边沿检测。因此，首先要将INTF9n位和INTR9n位清零，然后再设定端口模式。	p. 564 <input type="checkbox"/>
				当这些寄存器不被用作INTP4~INTP6时，一定要将INTF9n 和 INTR9n位清除为00。	p. 564 <input type="checkbox"/>
			NFC 寄存器	在改变采样时钟之后，需要用采样时钟数xNFSTS设定时间的次数的个数个采样时钟来初始化数字噪声消除器。如果在采样时钟改变后的采样时钟数xNFSTS设定时间的次数个时钟之后INTP3输入有效沿，将产生中断请求。因此，当使用中断和DMA功能的时候，一定要注意以下几点。 <ul style="list-style-type: none"> 当使用中断功能时，在经过采样时钟数xNFSTS设定时间的次数的个数个采样时钟之后，在中断请求标志(PIC3.PIF3位)被清除后，要允许中断。 当使用DMA功能时（由INTP3启动），在经过采样时钟数xNFSTS设定时间的次数的个数个采样时钟之后允许DMA。 	p. 565 <input type="checkbox"/>
			NMI 引脚	NMI引脚和P02引脚是功能复用引脚，复位后的功能是一个正常的端口。为了执行NMI的功能，在PMCO寄存器中确认NMI引脚。NMI引脚的初始设定是不检测边沿。使用INTF0寄存器和INTR0寄存器设定NMI引脚的有效沿。	p. 568 <input type="checkbox"/>
第十六章	软件	按键中断功能	KRM 寄存器	在KRM 寄存器被清为00H后，改写KRM 寄存器。	p. 570 <input type="checkbox"/>
				如果改变KRM寄存器，则会产生中断请求信号（INTKR）。为避免这种情况，在禁止中断（DI）或屏蔽中断之后改变KRM寄存器，然后将中断请求标志（KRIC、KRIF位）清零，再允许中断或清除屏蔽。	p. 570 <input type="checkbox"/>
			KR0~ KR7 引脚	如果低电平被输入给KR0 ~ KR7中的任何一个，即使其它引脚的下降沿被输入，也不会产生INTKR信号。	p. 570 <input type="checkbox"/>
			RXDA1 引脚 KR7 引脚	不能同时使用RXDA1和KR7引脚。为了使用RXDA1引脚，就不要使用KR7引脚。为了使用KR7引脚，就不要使用RXDA1引脚（推荐设置PFCE91位为1并且将PFCE91位清零）。	p. 570 <input type="checkbox"/>
		使用按键中断功能	如果改变KRM寄存器，则会产生中断请求信号（INTKR）。为避免这种情况，在禁止中断（DI）或屏蔽中断之后改变KRM寄存器，然后将中断请求标志（KRIC、KRIF位）清零，再允许中断或清除屏蔽。	p. 570 <input type="checkbox"/>	
第十七章	软件	待机功能	PSC 寄存器	在设置IDLE1, IDLE2, STOP或副IDLE模式之前，先设置PSMR.PSM1和PSMR.PSM0位，然后设置STP位。	p. 573 <input type="checkbox"/>
				当释放HALT模式时对NMI1M, NMI0M和INTM位进行设置是非法的。	p. 573 <input type="checkbox"/>
				如果在NMI1M, NMI0M, 或 INTM位设置为1的同时，STP位也设置为1，则NMI1M, NMI0M, 或 INTM位设置无效。当设置IDLE1/IDLE2/STOP 模式时，如果有一特定未屏蔽中断请求信号，设置对应中断请求信号(NMI1M, NMI0M, 或 INTM)的位为1，然后设置STP位为1。	p. 573 <input type="checkbox"/>
			PSMR 寄存器	确定将位2~位7清零。	p. 574 <input type="checkbox"/>
				仅当PSC.STP 位为1时，PSM0和PSM1位是有效的。	p. 574 <input type="checkbox"/>
OSTS 寄存器	STOP模式释放后的等待时间不包括从释放STOP模式到时钟振荡启动这段时间（即下图“a”所示的部分），不管通过复位或中断请求信号的产生是否释放了STOP模式。	p. 575 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第十七章	软件	待机功能	OSTS 寄存器	确定将位3~位7清零。	p. 575 <input type="checkbox"/>
				复位释放后的振荡稳定时间是 $2^{16}/f_x$ （因为OSTS寄存器的初始值= 06H）。	p. 575 <input type="checkbox"/>
			HALT 模式	在HALT指令之后插入五个或更多的NOP指令。	p. 576 <input type="checkbox"/>
				如果在执行HALT指令时有一个未被屏蔽的中断请求被保持等待响应，则此状态被转移到HALT模式中，然后通过这个等待响应的中断请求，HALT模式被立即释放。	p. 576 <input type="checkbox"/>
			IDLE1 模式	在执行往PSC寄存器中存入数据来设置IDLE1模式的指令之后，插入五个或更多的NOP指令。	p. 578 <input type="checkbox"/>
				如果设置IDLE1模式时有一个未被屏蔽的中断请求信号被保持等待响应，则IDLE1模式立即被此等待响应的中断请求释放。	p. 578 <input type="checkbox"/>
			释放IDLE1 模式	通过设置PSC.NMI1M，PSC.NMI0M和PSC.INTM位为1来禁止中断请求信号，这样中断请求信号是无效的，IDLE1模式也不会被释放。	p. 578 <input type="checkbox"/>
				如果通过使用NFC寄存器选择数字噪声清除，采样时钟选自fxx/64，fxx/128，fxx/256，fxx/512，和fxx/1024，IDLE1模式不能通过INTP3引脚中断请求信号释放。关于更多细节，敬请参阅15.6.2（5）噪声清除控制寄存器（NFC）。	p. 578 <input type="checkbox"/>
			IDLE2 模式	在执行往PSC寄存器中存入数据来设置IDLE2模式的指令之后，插入五个或更多的NOP指令。	p. 580 <input type="checkbox"/>
				如果设置IDLE2模式时有一个未被屏蔽的中断请求信号被保持等待响应，则IDLE2模式立即被此等待响应的中断请求释放。	p. 580 <input type="checkbox"/>
			释放IDLE2 模式	通过设置PSC.NMI1M，PSC.NMI0M和PSC.INTM位为1来禁止中断请求信号，这样中断请求信号是无效的，IDLE2模式也不会被释放。	p. 580 <input type="checkbox"/>
				如果通过使用NFC寄存器选择数字噪声清除，采样时钟选自fxx/64，fxx/128，fxx/256，fxx/512，和fxx/1024，IDLE1模式不能通过INTP3引脚中断请求信号释放。关于更多细节，敬请参阅15.6.2（5）噪声清除控制寄存器（NFC）。	p. 580 <input type="checkbox"/>
			STOP 模式	在执行往PSC寄存器中存入数据来设置STOP模式的指令之后，插入五个或更多的NOP指令。	p. 583 <input type="checkbox"/>
				如果设置STOP模式时有一个未被屏蔽的中断请求信号被保持等待响应，则STOP模式立即被此等待响应的中断请求释放。	p. 583 <input type="checkbox"/>
			释放STOP 模式	通过设置PSC.NMI1M，PSC.NMI0M和PSC.INTM位为1来禁止中断请求信号，这样中断请求信号是无效的，STOP模式也不会被释放。	p. 583 <input type="checkbox"/>
				如果通过使用NFC寄存器选择数字噪声清除，采样时钟选自fxx/64，fxx/128，fxx/256，fxx/512，和fxx/1024，IDLE1模式不能通过INTP3引脚中断请求信号释放。关于更多细节，敬请参阅15.6.2（5）噪声清除控制寄存器（NFC）。	p. 583 <input type="checkbox"/>
STOP模式中的操作状态	当操作A/D转换器时，如果设置STOP模式，则A/D转换器自动停止操作，并且在STOP模式释放后再次启动操作。然而，在这种情况下，释放STOP模式后的A/D转换结果是无效的。所有设置STOP模式之前的A/D转换结果也是无效的。	p. 584 <input type="checkbox"/>			
	即使在操作A/D转换器时设置STOP模式，功耗降低也与在设置STOP模式之前停止A/D转换器时的一样。	p. 584 <input type="checkbox"/>			

章节	分类	功能	功能细节	注意事项	页码
第十七章	软件	待机功能	副时钟操作模式	当操作CK3位时, 不要改变CK2 ~ CK0位 (推荐使用位操作指令来操作) 的设置值。关于PCC寄存器的更多细节, 敬请参阅5.3 (1) 处理器时钟控制寄存器(PCC)。	p. 586 <input type="checkbox"/>
				如果以下的条件不满足, 改变CK2 ~ CK0位的设置以便满足条件并且设置副时钟操作模式。 内部系统时钟(fCLK) > 副时钟 (fXT) ×4	p. 586 <input type="checkbox"/>
			释放副时钟操作模式	当操作CK3位时, 不要改变CK2 ~ CK0位 (推荐使用位操作指令来操作) 的设置值。 关于PCC寄存器的更多细节, 敬请参阅5.3 (1) 处理器时钟控制寄存器(PCC)。	p. 586 <input type="checkbox"/>
			释放副时钟操作模式	停止主时钟之前确定停止PLL (PLLCTL.PLLON位 = 0)。	p. 587 <input type="checkbox"/>
				当CPU工作在副时钟并且主时钟振荡停止时, 不能访问发生等待的寄存器。如果产生等待, 则其只能通过复位来释放 (见3.4.8(2))。	p. 587 <input type="checkbox"/>
			Sub-IDLE 模式	在执行往PSC寄存器中存入数据来设置sub-IDLE模式的指令之后, 插入五个或更多的NOP指令。	p. 588 <input type="checkbox"/>
				如果设置sub-IDLE模式时有一个未被屏蔽的中断请求信号被保持等待响应, 则副IDLE模式立即被此等待响应的中断请求释放。	p. 588 <input type="checkbox"/>
			释放sub-IDLE模式	通过设置PSC.NMI1M, PSC.NMI0M和PSC.INTM位为1来禁止中断请求信号, 这样中断请求信号是无效的, sub-IDLE模式也不会被释放。	p. 589 <input type="checkbox"/>
				当释放sub-IDLE模式时, 从产生释放副IDLE模式的中断请求信号开始到副IDLE模式被释放之间须等待12个副时钟周期 (大约366 μs)。	p. 589 <input type="checkbox"/>
				如果通过使用NFC寄存器选择数字噪声清除, 采样时钟选自fxx/64, fxx/128, fxx/256, fxx/512, 和 fxx/1024, IDLE1 模式不能通过INTP3引脚中断请求信号释放。更多细节, 敬请参阅15.6.2 (5) 噪声清除控制寄存器 (NFC)。	p. 589 <input type="checkbox"/>
sub-IDLE 模式中的操作状态	停止主时钟之前确定停止PLL (PLLCTL.PLLON位 = 0)。	p. 590 <input type="checkbox"/>			
	为了实现低功耗, 在进入sub-IDLE模式之前停止A/D转换器。	p. 590 <input type="checkbox"/>			
第十八章	软件	复位功能	紧急事件操作模式	当CPU在内部振荡时钟下操作, 有权使用在等待状态下禁止使用的寄存器。关于等待状态下禁止使用的寄存器, 参见3.4.8 (2) 访问特殊片上外围I/O寄存器。	p. 591 <input type="checkbox"/>
		复位功能	RESF 寄存器	寄存器的每一位只能被写入"0"。如果写入"0"和设置标志(复位事件)冲突, 设置标志优先。	p. 592 <input type="checkbox"/>
	硬件	内部 RAM 复位后的状态	V850ES/HG2的固件使用了的系统复位释放后操作内部RAM的部分一部分, 支持一个启动交换功能。因此, 一些RAM区域的部分不能保持上电复位值, 更多细节, 敬请参阅18.4 复位释放后的操作。	pp. 593, 595 <input type="checkbox"/>	
		RESET引脚输入的硬件状态	当电源开启, 下列引脚可能临时在复位时输出一个不确定的电平。 • P53/KR3/TIQ00/TOQ00/DDO 引脚	p. 593 <input type="checkbox"/>	
			RESET引脚的输入初始化OCDM寄存器。因此, 注意注意事项, 复位释放后在OCDM.OCDM0位被清除前如果一个高电平输入到P05/DRST 引脚, 可能进入片上调试模式。更多细节, 敬请参阅第四章端口功能。	p. 593 <input type="checkbox"/>	

章节	分类	功能	功能细节	注意事项	页码
第十九章	软件	时钟监视器	CLM 寄存器	一旦CLME位被设为1，除了复位其余任何操作都不能将其清为零。	p. 600 <input type="checkbox"/>
				由时钟监视器引起的一个复位，CLME位被清为零并且RESF.CLMRF被设置为1。	p. 600 <input type="checkbox"/>
			内部振荡器	可通过使用选项字节功能（参见第二十四章）来停止内部振荡器，设置RCM.RSTOP位为1。	p. 601 <input type="checkbox"/>
				当内部振荡器停止操作时，时钟监视器停止。	p. 601 <input type="checkbox"/>
第二十一章	软件	低电压检测器 (LVI)	LVIM 寄存器	在设置LVION位为1后，在使用LVIF位检测电压前等待0.2 ms (MAX.)。	p. 607 <input type="checkbox"/>
				LVION位=1及LVIMD位=0时，LVIF标志的值作为输出信号INTLVI输出。	p. 607 <input type="checkbox"/>
				一定要将位2~6清零。	p. 607 <input type="checkbox"/>
				在LVION位及LVIMD位设置为1后，发生了非低电压检测，因此，低电压检测电路无法停止除非接收到复位信号。	p. 607 <input type="checkbox"/>
			LVIS 寄存器	在LVION.LVION位及LVIMD.LVIMD位设置为1后，发生了非低电压检测，因此，无法写入该寄存器除非接收到复位请求。	p. 608 <input type="checkbox"/>
				一定要将位1~7清零。	p. 608 <input type="checkbox"/>
			RAMS 寄存器	下面所示的是复位后的具体顺序。	p. 608 <input type="checkbox"/>
				<ul style="list-style-type: none"> 设置条件 <ul style="list-style-type: none"> 检测到比设定电压低的电压 由指令设置 通过看门狗定时器溢出发生复位信号。 当访问RAM时，产生复位信号 通过时钟监视器发生复位信号。 清除条件: <ul style="list-style-type: none"> 以特定顺序写入零 	
		使用内部复位信号	如果LVIMD位设置为1，那么LVIM和LVIS寄存器的内容不能改变，直到LVI之外的复位请求产生。	p. 609 <input type="checkbox"/>	
		PEMU1 寄存器	EVARAMIN不能自动清除。	p. 613 <input type="checkbox"/>	
第二十三章	硬件	Flash 存储器	通讯模式	按照处理未用引脚方式来处理图上未标示的引脚。（参见2.3引脚I/O电路类型及未用引脚的推荐连接方式）。如有必要，连接一阻抗为1 kΩ~10 kΩ的电阻。	pp. 622, 623 <input type="checkbox"/>
				勿输入高电平至DRST引脚。	pp. 622, 623 <input type="checkbox"/>
			PG-FP4	连接图23-6所示的引脚，或然后通过板上下拉电阻连接至GND。	p. 623 <input type="checkbox"/>
				无法通过flash编程器的CLK引脚提供时钟。在板上创建振荡器来提供时钟。	p. 623 <input type="checkbox"/>
			FA-100GC-8EU-A	务必通过一值为4.7 μF（推荐值）的电容器来将REGC引脚连接至GND。	p. 624 <input type="checkbox"/>
				无法通过flash编程器的CLK引脚提供时钟。在板上创建振荡器来提供时钟。	p. 624 <input type="checkbox"/>
				如图所示连接FLMD1引脚，或者通过下拉电阻连接该引脚到板上的GND。	p. 626 <input type="checkbox"/>
				通过在flash存储器写入适配器（在虚线内）上创建一个振荡器提供一个时钟。以下是振荡器的一个实例。	p. 626 <input type="checkbox"/>
		勿输入高电平至DRST引脚。	p. 626 <input type="checkbox"/>		

章节	分类	功能	功能细节	注意事项	页码	
第二十三章	硬件	Flash 存储器	通讯模式的选择	当选择了UARTA0时，基于接收FLMD0脉冲后从专门的Flash 存储器编程器收到的复位命令计算接收时钟。	p. 628 <input type="checkbox"/>	
			FLMD1 引脚	如果在片上写入过程中和复位之后立即从别的设备输入V _{DD} 信号到FLMD1引脚，隔离此信号。	p. 630 <input type="checkbox"/>	
			FLMD0 引脚	确保当复位释放时FLMD0引脚为0V。	p. 637 <input type="checkbox"/>	
第二十四章	硬件	选项字节功能	示例程序CA850	确保在此扇区中写入6字节，如果少于6字节，则在连接器运行时会发生错误。 错误消息： F4112： 非法“OPTION_BYTES”扇区大小。	p. 640 <input type="checkbox"/>	
第二十五章	硬件, 软件	片上调试功能	OCDM 寄存器	外部复位后，当DDI， DDO， DCK， 和DMS 引脚不用于片上调试引脚而作为端口引脚时，可以进行下面的任何一项操作。 • 输入低电压到P05/INTP2/DRST 引脚。 • 设置ODCM0 位。在这种情况下，进行如下操作。 <1> OCDM0 位清零。 <2> P05/INTP2/DRST引脚恒为低电平直到<1> 完成。 DRST引脚有片上下拉电阻。这个电阻当OCDM0 标志清零时断开。	p. 645 <input type="checkbox"/>	
			注意事项 (DUC)	RUN（程序执行）期间，如果输入复位信号（从目标系统或从内部复位源来的复位信号），暂停功能就可能出现故障。 即使复位信号由屏蔽功能屏蔽，如果复位信号是从引脚输入的话，I/O 缓冲器（端口引脚）也可能复位。 因为内部 flash 存储器内的软件断点设置是通过ROM 修正功能实现的，它可以通过目标复位或由看门狗定时器2产生的内部复位强制为无效。当硬件暂停或者强制暂停发生时，断点再次有效，在此期间不产生软件暂停。 暂停期间屏蔽引脚复位屏蔽，CPU 和外围 I/O 不复位。用户程序执行的时候，flash 存储器一旦由DMM重写或者由RAM监视功能读取就马上产生引脚复位或内部复位的情况下，CPU和外围 I/O 可能不正确复位。 当下列条件（a）和（b）满足时，由于暂停等原因仿真器（IECUBE [®] ，MINICUBE）操作停止的情况下，看门狗定时器2并不停止，并且产生复位或不可屏蔽中断。当发生复位时，调试器挂起。 (a) 主时钟或副时钟用作看门狗定时器2的源时钟。 (b) 内部振荡时钟停止（RCM.RSTOP 位 = 1）。 为避免这种情况，可进行如下两种处理。 • 当使用仿真器时，内部振荡时钟用作源时钟。 • 当使用仿真器时，内部振荡器不要停止。	p. 647 <input type="checkbox"/>	
	软件				即使复位信号由屏蔽功能屏蔽，如果复位信号是从引脚输入的话，I/O 缓冲器（端口引脚）也可能复位。	p. 647 <input type="checkbox"/>
					因为内部 flash 存储器内的软件断点设置是通过ROM 修正功能实现的，它可以通过目标复位或由看门狗定时器2产生的内部复位强制为无效。当硬件暂停或者强制暂停发生时，断点再次有效，在此期间不产生软件暂停。	p. 647 <input type="checkbox"/>
					暂停期间屏蔽引脚复位屏蔽，CPU 和外围 I/O 不复位。用户程序执行的时候，flash 存储器一旦由DMM重写或者由RAM监视功能读取就马上产生引脚复位或内部复位的情况下，CPU和外围 I/O 可能不正确复位。	p. 647 <input type="checkbox"/>
				当下列条件（a）和（b）满足时，由于暂停等原因仿真器（IECUBE [®] ，MINICUBE）操作停止的情况下，看门狗定时器2并不停止，并且产生复位或不可屏蔽中断。当发生复位时，调试器挂起。 (a) 主时钟或副时钟用作看门狗定时器2的源时钟。 (b) 内部振荡时钟停止（RCM.RSTOP 位 = 1）。 为避免这种情况，可进行如下两种处理。 • 当使用仿真器时，内部振荡时钟用作源时钟。 • 当使用仿真器时，内部振荡器不要停止。	p. 647 <input type="checkbox"/>	
				当下列条件（a）和（b）满足时，由于暂停等原因仿真器（IECUBE [®] ，MINICUBE）操作停止的情况下，TMM 并不停止，即使外围暂停功能设置为“Break”。 (a) INTWT，内部振荡时钟（f _R /8），或者选择副时钟用作 TMM 源时钟。 (b) 主时钟停止。 为避免这种情况，可进行如下两种处理。 • 使用仿真器时，主时钟（f _{xx} ，f _{xx} /2，f _{xx} /4，f _{xx} /64，f _{xx} /512）用作源时钟。 • 在片上调试模式下，DDO 引脚强制为高电平输出。	p. 647 <input type="checkbox"/>	

章节	分类	功能	功能细节	注意事项	页码
第二十五章	硬件	片上调试功能	注意事项 (DUC)	在片上调试模式下，DDO 引脚强制为高电平输出。	p. 647 <input type="checkbox"/>
			注意事项 (非 DUC)	不要将用于调试的设备安装在大批量生产的产品上，因为调试过程中flash存储器会被改写，flash存储器改写的次数不能保证。此外，不要把调试监测程序嵌入大批量生产的产品中。	p. 656 <input type="checkbox"/>
	软件			如果满足下列条件之一，强制断点不能执行。 <ul style="list-style-type: none"> 中断禁止 (DI) 发给串行接口并用于MINICUBE2和目标设备之前通信的中断被屏蔽 禁用由可屏蔽中断释放待机时进入待机模式 MINICUBE2与目标设备之间通信模式为UARTA0且主时钟被MINICUBE2停止 	p. 656 <input type="checkbox"/>
				满足如下条件之一时，伪实时RAM监测器RRM功能和DMM功能不工作。 <ul style="list-style-type: none"> 中断禁止 (DI) 发给串行接口用于MINICUBE2和目标设备之间通信的中断被屏蔽 由于禁止可屏蔽中断释放待机而进入待机模式 MINICUBE2与目标设备之间通信模式为UARTA0，且主时钟被MINICUBE2停止 MINICUBE2与目标设备之间通信模式为UARTA0，且用于通信的时钟与调试器指定的时钟不同。 	p. 656 <input type="checkbox"/>
				如果如下条件之一满足，待机模式由伪RRM和DMM功能释放。 <ul style="list-style-type: none"> MINICUBE2和目标设备之间的通信模式为CSIB0 MINICUBE2和目标设备之间的通信模式为UARTA0，且主时钟已加入。 	p. 656 <input type="checkbox"/>
				需要特定顺序的外围I/O寄存器不能用DMM功能写入。	p. 656 <input type="checkbox"/>
				当调试器初次启动时，将进行芯片清除，写入用于调试的监视程序，但该操作大约需10多秒钟。	p. 656 <input type="checkbox"/>
				当CPU操作时钟设置与调试器一起改变时，调试器重写监视程序。所需时间与上述(6)相同。对于集成调试器ID850QB，当改变时钟条及配置对话框时该程序仍然适用。	p. 656 <input type="checkbox"/>
				如果分配了调试监测程序的空间被flash自编程改写，调试器不能再正常工作。	p. 656 <input type="checkbox"/>
				安全ID	flash存储器擦除后，全部区域写入1。
第二十六章	硬件	电气特性	最大额定值	确保不要超过每个电源电压的最大额定值（最大值）。	p. 660 <input type="checkbox"/>
				避免IC设备输出引脚（或I/O）及V _{DD} 或V _{CC} 与GND的直接接触。	pp. 660, 661 <input type="checkbox"/>
				任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。 DC特性和AC特性中指出的额定值和条件是正常工作的质量保证。	pp. 660, 661 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码
第二十六章	硬件	电气特性	最大额定值	当将外部电路与处于高电阻状态的引脚直接相连时，设计的时序可使外部电路上避免输出冲突。	pp. 660, 661 <input type="checkbox"/>
			主时钟振荡器特性	当使用主时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。 <ul style="list-style-type: none"> • 连接线越短越好。 • 连接线不应与其他信号线交叉。 • 流经的电流变化较大的信号线不要在振荡器周围布线。 • 要保持振荡器电容器的接地点电压与V_{SS}相同。 • 不要将电容的地信号接入大电流地。 • 不要从振荡器获取信号。 	p. 663 <input type="checkbox"/>
	软件			当主时钟停止副时钟工作时，在转换回主时钟前等待直到振荡稳定时间，由程序保证。	p. 663 <input type="checkbox"/>
	硬件		副时钟振荡器特性	当使用副时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。 <ul style="list-style-type: none"> • 连接线越短越好。 • 连接线不应与其他信号线交叉。 • 流经的电流变化较大的信号线不要在振荡器周围布线。 • 要保持振荡器电容器的接地点电压与V_{SS}相同。 • 不要将电容的地信号接入大电流地。 • 不要从振荡器获取信号。 	p. 664 <input type="checkbox"/>
				为了缩减能耗，副时钟振荡器被设计为低幅电路，并且它比主时钟振荡器更容易受到噪声的影响，此在使用副时钟时更要注意布线方法。	p. 664 <input type="checkbox"/>
			电压调节器特性	当RESET = V _{SS} = 0 V确保V _{DD} 电压上升。	p. 665 <input type="checkbox"/>
			引脚漏电流	T FLMD0引脚值如下。 <ul style="list-style-type: none"> • 输入漏电流，高：2 μA（最大。） • 输入漏电流，低：-2 μA（最大。） 	p. 667 <input type="checkbox"/>
			数据保持特性	转换到STOP模式和从STOP模式还原必须在额定的工作范围完成。	p. 669 <input type="checkbox"/>
			AC特性	如果由于电路结构加载电容超过50 pF，通过插入缓冲器或其他方法将设备的加载电容降到50 pF以下。	p. 670 <input type="checkbox"/>
	软件		编程器特性	在出货后第一次对产品进行写操作时，“擦除 → 写”和“只写”作为一次改写。 示例（P：写入，E：擦除） 出库产品 → P → E → P → E → P：3 改写 出库产品 → E → P → E → P → E → P：3 改写	p. 679 <input type="checkbox"/>

章节	分类	功能	功能细节	注意事项	页码
第二十八章	硬件	推荐焊接条件	贴装焊接条件	不要同时使用不同的焊接方式（除局部加热外）。	p. 681 <input type="checkbox"/>
附录 A	软件	开发工具	RX850, RX850 Pro	为了购买RX850 或 RX850 Pro，首先需要填写购买申请表并且签署用户协议。	p. 690 <input type="checkbox"/>
附录 C	软件	指令集列表	设置指令	对于通用寄存器reg1 和 reg3，不要指定相同的寄存器。	p. 710 <input type="checkbox"/>

E.1 在此版本中的主要修订

(1/2)

页数	描述
71 页	增加 3.4.7 特殊寄存器的描述
154 页	增加 4.5.1 (b)复用功能模式的注意事项 (输入)
156 页	修改图 5-1 时钟发生器
157 页	修改 5.2 (8) 预分频器 4 的描述
163 页	修改表 5-1 各时钟的操作状态
167 页	修改 5.5.2 (4) 可编程时钟模式寄存器 (PCLM)
176 页	修改 6.4 (3) TMPn I/O 控制寄存器 0 (TPnIOC0)
199 页	增加图 6-11 外部事件计数模式操作的寄存器设置中的注意事项
220 页	增加图 6-22 单脉冲输出模式的寄存器设置中的注意事项
262 页	增加 6.7 (1) 选择器操作控制寄存器 0 (SELCNT0)中的注
272 页	增加 7.4 (3) TMQn I/O 控制寄存器 0 (TQnIOC0)中的注
298 页	增加图 7-11 外部事件计数模式操作的寄存器设置中的注意事项
322 页	增加图 7-22 单脉冲输出模式的寄存器设置中的注意事项
391 页	修改 10.3 (1) 看门狗定时器模式寄存器 2 (WDTM2)中的注意事项 3, 4
393 页	修改 10.3 (2) 看门狗定时器使能寄存器 (WDTE)中的注意事项 3
401 页	增加表 11-2 转换模式下的设置示例 中的注意事项
406 页	修改 11.4 (7) 掉电比较阈值寄存器 (ADA0PFT)中的描述
421 页	修改 11.6 (8)读取 ADA0CRn 结果中的描述
422 页	增加 11.6 (10) 待机模式
422 页	增加 11.6 (11)稳定时间内寄存器和触发器输入的改写
422 页	修改 11.6 (13) A/D 转换转换结果滞后性中的描述
431 页	增加 12.3 (1) UARTAn 控制寄存器 0 (UAnCTL0)中的描述
432 页	增加 12.3 (4) UARTAn 选项控制寄存器 0 (UAnOPT0) 中的描述
436 页	增加 12.4 (1) 接收完成中断请求信号 (INTUAnR) 中的描述
442 页	增加 12.5.4 SBF 接收中的注意事项
463 页	修改 13.3 (1) CSIBn 控制寄存器 0 (CBnCTL0)中的注意事项
466 页	修改 注意事项, 增加 13.3 (2) CSIBn 控制寄存器 1 (CBnCTL1)中的注 1
471 页~ 503 页	修改 13.5 操作
506 页	增加 13.7.1 波特率发生器的注意事项
515 页	修改 14.3 (6) DMA 触发因素寄存器 0 ~ 3 (DTFR0 ~ DTFR3)中的注
530 页	增加 14.13 (11) DMA 启动系数中的描述
565 页	修改 15.6.2 (5) 噪声消除控制寄存器(NFC)中的描述
573 页	增加 17.2 (1) 功能节控制寄存器(PSC)中的注意事项 3
593 页	修改表 18-1 RESET 引脚输入后的硬件状态中注 1
595 页	修改表 18-2 看门狗定时器 2 复位操作时的硬件状态中的注
597 页	增加 18.4 复位释放后的操作
607 页	增加 21.3 (1) 低电压检测寄存器(LVIM)中的注意事项 4

页数	描述
610 页	修改图 21-2 低电压检测器的操作时序 (LVIMD 位= 1)
617 页	修改 23.2 内存配置
618 页	修改 23.3 功能概述
622 页	修改 23.4.2 (1) UARTA0 中传输速率
629 页	修改表 23-7 Flash 存储器控制命令
636 页	修改图 23-17 标准自编程流程
638 页	修改表 23-11 使用的内部资源
640 页	增加第二十四章中 选项字节功能中的描述
641 页~ 659 页	修改第二十五章 片上调试功能
665 页	增加 26.5 电压调节器特性的注意事项
681 页	增加第二十八章 推荐焊接条件
682 页	增加附录 A 开发工具
711 页	增加附录 D 注意事项列表
744 页	增加附录 E 修订履历

详细信息请联系:

(中国区)

网址:

<http://www.cn.necel.com/>

<http://www.necel.com/>

[北京]

日电电子(中国)有限公司
中国北京市海淀区知春路 27 号
量子芯座 7, 8, 9, 15 层
电话: (+86) 10-8235-1155
传真: (+86) 10-8235-7679

[深圳]

日电电子(中国)有限公司深圳分公司
深圳市福田区益田路卓越时代广场大厦 39
楼 3901, 3902, 3909 室
电话: (+86) 755-8282-9800
传真: (+86) 755-8282-9899

[上海]

日电电子(中国)有限公司上海分公司
中国上海市浦东新区银城中路 200 号
中银大厦 2409-2412 和 2509-2510 室
电话: (+86) 21-5888-5400
传真: (+86) 21-5888-5230

[香港]

香港日电电子有限公司
香港九龙旺角太子道西 193 号新世纪广场
第 2 座 16 楼 1601-1613 室
电话: (+852) 2886-9318
传真: (+852) 2886-9022
2886-9044

上海恩益禧电子国际贸易有限公司
中国上海市浦东新区银城中路 200 号
中银大厦 2511-2512 室
电话: (+86) 21-5888-5400
传真: (+86) 21-5888-5230