

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.



用户手册初稿

V850ES/JG3-U, V850ES/JH3-U

32 位单片微控制器

硬件

V850ES/JG3-U

μ PD70F3763

μ PD70F3764

V850ES/JH3-U

μ PD70F3768

μ PD70F3769

文档编号: U19287CA1V0UD00 (第 1 版)

发行日期: 2009 年 3 月 N

© NEC Electronics Corporation 2009

日本印刷

[备忘录]

CMOS 设备注意事项

① 输入引脚处的电压波形

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

② 未使用的输入引脚的处理

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

③ ESD防护措施

如果MOS设备周围有强电场,将会击穿氧化栅极,从而影响设备的运行。因此必须采取措施,尽可能防止静电产生。一旦有静电,必须立即释放。对于环境必须有适当的控制。如果空气干燥,应当使用增湿器。建议避免使用容易产生静电的绝缘体。半导体设备的存放和运输必须使用抗静电容器、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。

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

EEPROM 是 **NEC 电子**的注册商标。

MINICUBE 是一个 **NEC Electronics Corporation** 在日本和德国的注册商标或在美国的标志。

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

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

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

PC/AT 是 **International Business Machines Corporation** 的商标。

SPARCstation是**SPARC International, Inc.**的商标。

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

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

ITRON 是 **Industrial TRON** 的缩写。

- 本档所登载的内容有效期截止至 2009 年 3 月，信息先于产品的生产周期发布。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。否则因本档所登载内容引发的错误，本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的，对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或其他人所有的专利、版权以及其它知识产权作出任何明示或默示的许可及授权。
- 本文件中的电路、软件以及相关信息仅用以说明半导体产品的运作和应用实例。用户如在设备设计中应用本文件中的电路、软件以及相关信息，应自行负责。对于用户或其他人因使用了上述电路、软件以及相关信息而引起的任何损失，本公司概不负责。
- 虽然本公司致力于提高半导体产品的质量及可靠性，但用户应同意并知晓，我们仍然无法完全消除出现产品缺陷的可能。为了最大限度地减少因本公司半导体产品故障而引起的对人身、财产造成损害（包括死亡）的危险，用户务必在其设计中采用必要的安全措施，如冗余度、防火和防故障等安全设计。
- 本公司产品质量分为：

“标准等级”、“专业等级”以及“特殊等级”三种质量等级。

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

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

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

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

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

（注）

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

M5 02.11-1

前言

读者	本手册旨在使用户了解 V850ES/JG3-U 和 V850ES/JH3-U 的功能及使用 V850ES/JG3-U 和 V850ES/JH3-U 设计应用系统。
目的	本手册旨在使用户了解 V850ES/JG3-U 和 V850ES/JH3-U 硬件功能，下面显示了手册详细组织
组织	手册的产品分为两部分: 硬件 (本手册)和结构化 (V850ES 结构化用户手册)。

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 (目标值)

结构化

- 数据类型:
- 寄存器设置
- 指令格式和指令设置
- 中断和运算
- 流水线操作

如何阅读本手册	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电子工程方面的基础知识。
---------	--

要了解 V850ES/JG3-U 和 V850ES/JH3-U 的全部功能
→ 请按 **目录** 顺序阅读本手册

要查找详细的寄存器名称
→ 请参阅 **附录 C 寄存器索引**

寄存器格式
→ 尖括号(<>)中的二进制位名称在设备文件中被定义为保留字。

要了解指令功能的详细内容
→ **请参考 V850ES 结构化用户手册**

要了解 V850ES/JG3-U 和 V850ES/JH3-U 的电气特性
→ 请参考 **第 33 章 电气特性 (目标值)**

本手册中“xxx 寄存器”的“yyy 位”描述为“xxx.yyy 位”。 注意当如果在程序中写成“xxx.yyy”，那么汇编器/编译器将无法正确识别它。

规则

数据规则:

有效低电平表示法:

存储空间地址:

注:

注意事项:

备注:

数值的表示:

前缀表示 2 的乘幂

(地址空间, 存储器容量) capacity):

数据的高位部分在左边, 低位部分在右边

$\overline{\text{xxx}}$ (在引脚和信号名称上加划一条线)

高位地址在上部, 低位地址在下方

文中用**注**标注的相关术语的脚注

需要特别关注的信息

补充信息

二进制 ... xxxx 或 xxxxB

十进制 ... xxxx

十六进制 ... xxxxH

K (K): $2^{10} = 1,024$

M (兆): $2^{20} = 1,024^2$

G (G): $2^{30} = 1,024^3$

相关文档

本手册中提到的相关文档可能包括有初稿版本。但是，初稿版本没有特别注明。

V850ES/JG3-U和V850ES/JH3-U相关文档

文档名称	文档编号
V850ES 结构化用户手册	U15943E
V850ES/JG3-U, V850ES/JH3-U 硬件用户手册	本手册

开发工具相关文档

文档名称	文档编号	
QB-V850ESJX3H [®] 片上仿真器	准备中	
QB-V850MINI 片上调试仿真器	U17638E	
QB-MINI2 具有编程功能的片上调试仿真器	U18371E	
RA78K0R Ver. 3.20 C 编译器	操作篇	U18512E
	C 语言篇	U18513E
	汇编语言篇	U18514E
	链接跳转	U18515E
PM+ Ver. 6.30 项目管理器	U18416E	
ID850QB Ver. 3.40 综合调试器	操作篇	U18604E
SM850 Ver. 2.50 系统仿真器	操作篇	U16218E
SM850 Ver. 2.00 或更高版本系统仿真器	外部用户打开接口规格	U14873E
SM+ 系统仿真器	操作篇	U18601E
	用户开放接口	U18212E
RX850 Ver. 3.20 实时操作系统	基础篇	U13430E
	安装篇	U17419E
	技术篇	U13431E
	任务调试器	U17420E
RX850 Pro Ver. 3.21 实时操作系统	基础篇	U18165E
	内部结构	U18164E
	任务调试器	U17422E
AZ850 Ver. 3.30 系统执行分析器	U17423E	
PG-FP4 存储器编程器	U15260E	
PG-FP5 存储器编程器	U18865E	

注： 开发中

目录

第一章 导言	21
1.1 概述	21
1.2 特性	24
1.3 应用领域	26
1.4 订购信息	26
1.5 引脚配置（顶视图）	27
1.6 功能框图结构	30
1.6.1 内部框图	30
1.6.2 内部单元	32
第二章 引脚功能	35
2.1 引脚功能列表	35
2.2 引脚状态	49
2.3 引脚 I/O 电路类型、I/O 缓冲器电源、及未使用引脚的连接	50
2.4 注意事项	55
第三章 CPU 功能	56
3.1 特性	56
3.2 CPU 寄存器组	57
3.2.1 程序寄存器组	58
3.2.2 系统寄存器组	59
3.3 操作模式	65
3.3.1 指定操作模式	65
3.4 地址空间	66
3.4.1 CPU 地址空间	66
3.4.2 CPU 地址空间的绕回	67
3.4.3 存储器映射	68
3.4.4 存储区域	70
3.4.5 地址空间的使用建议	74
3.4.6 外设 I/O 寄存器	77
3.4.7 专用寄存器	91
3.4.8 注意事项	95
第四章 端口功能	99
4.1 特性	99
4.2 端口基本配置	99
4.3 端口配置	101
4.3.1 端口 0	107
4.3.2 端口 1	112
4.3.3 端口 2（仅限 V850ES/JH3-U）	113
4.3.4 端口 3	117
4.3.5 端口 4	122
4.3.6 端口 5	125
4.3.7 端口 6	132

4.3.8	端口 7	136
4.3.9	端口 9	138
4.3.10	端口 CM	148
4.3.11	端口 CS (仅限 V850ES/JH3-U)	151
4.3.12	端口 CT	153
4.3.13	端口 DH (仅限 V850ES/JH3-U)	156
4.3.14	端口 DL	158
4.4	使用复用功能时的端口寄存器设置	160
4.5	注意事项	171
4.5.1	设置端口引脚的注意事项	171
4.5.2	端口 n 寄存器 (Pn) 位操作指令的注意事项	174
4.5.3	片上调试引脚的注意事项 (仅限 V850ES/JG3-U)	175
4.5.4	P56/INTP05/DRST 引脚的注意事项	175
4.5.5	电源开启时 P10, P11 和 P53 引脚的注意事项	175
4.5.6	迟滞特性	175
第五章	总线控制功能	176
5.1	特性	176
5.2	总线控制引脚	177
5.2.1	访问片内 ROM 存储器、片内 RAM 存储器或片上外设输入/输出端口时的引脚状态	178
5.2.2	各种工作模式下的引脚状态	178
5.3	存储模块功能	179
5.4	总线访问	180
5.4.1	总线访问时钟数目	180
5.4.2	总线宽度设置功能	180
5.4.3	根据不同总线宽度进行访问	181
5.5	等待功能	188
5.5.1	可编程等待功能	188
5.5.2	外部等待功能	189
5.5.3	可编程等待和外部等待之间的关系	190
5.5.4	可编程地址等待功能	191
5.6	空闲状态插入功能	192
5.7	总线保持功能 (仅限 V850ES/JH3-U)	193
5.7.1	功能概述	193
5.7.2	总线保持步骤	194
5.7.3	节电模式时的操作	194
5.8	总线优先权	195
5.9	总线时序	196
第六章	时钟发生功能	199
6.1	概述	199
6.2	配置	200
6.3	寄存器	202
6.4	操作	207
6.4.1	各时钟的操作	207
6.4.2	时钟输出功能	207
6.5	PLL 功能	208
6.5.1	概述	208

6.5.2	寄存器	208
6.5.3	用法	211
第七章	16 位定时器/事件计数器 AA (TAA)	212
7.1	概述	212
7.2	功能	212
7.3	配置	213
7.3.1	引脚配置	215
7.4	寄存器	216
7.5	操作	233
7.5.1	间隔定时器模式 (TAA _n MD2 至 TAA _n MD0 位=000)	239
7.5.2	外部事件计数模式 (TAA _n MD2 至 TAA _n MD0 位=001)	249
7.5.3	外部触发脉冲输出模式 (TAA _n MD2 至 TAA _n MD0 位= 010)	257
7.5.4	单次触发脉冲输出模式 (TAA _n MD2 至 TAA _n MD0 位=011)	269
7.5.5	PWM 输出模式 (TAA _n MD2 至 TAA _n MD0 位=100)	276
7.5.6	自由运行定时器模式 (TAA _n MD2 至 TAA _n MD0 位=101)	285
7.5.7	脉冲宽度测量模式 (TAA _n MD2 至 TAA _n MD0 位=110)	302
7.5.8	定时器输出操作	308
7.6	定时器调谐操作功能	309
7.6.1	自由运行定时器模式 (在定时器调谐操作期间)	311
7.6.2	PWM 输出模式 (定时器调谐操作期间)	318
7.7	同时启动功能	320
7.7.1	PWM 输出模式 (同时启动操作)	321
7.8	级联	323
7.9	选择器功能	328
7.10	注意事项	329
第八章	16 位定时器/事件计数器 AB (TAB)	330
8.1	概述	330
8.2	功能	330
8.3	配置	331
8.4	寄存器	334
8.5	操作	351
8.5.1	间隔定时器模式 (TAB _n MD2 至 TAB _n MD0 位=000)	352
8.5.2	外部事件计数模式 (TAB _n MD2 至 TAB _n MD0 位=001)	361
8.5.3	外部触发脉冲输出模式 (TAB _n MD2 至 TAB _n MD0 位=010)	370
8.5.4	单次触发脉冲输出模式 (TAB _n MD2 至 TAB _n MD0 位 = 011)	383
8.5.5	PWM 输出模式 (TAB _n MD2 至 TAB _n MD0 位=100)	392
8.5.6	自由运行定时器模式 (TAB _n MD2 至 TAB _n MD0 位=101)	403
8.5.7	脉冲宽度测量模式 (TAB _n MD2 至 TAB _n MD0 位=110)	423
8.5.8	三角波 PWM 模式 (TAB _n MD2 至 TAB _n MD0 位 = 111)	429
8.5.9	定时器输出操作	431
8.6	定时器调谐操作功能/同时启动功能	432
8.7	注意事项	433
第九章	16 位定时器/事件计数器 T (TMT)	434
9.1	概述	434
9.2	功能	434

9.3	配置	435
9.3.1	引脚配置	438
9.4	寄存器	439
9.5	定时器输出操作	459
9.6	操作	460
9.6.1	间隔定时器模式 (TT0MD3 至 TT0MD0 位=0000)	469
9.6.2	外部事件计数模式 (TT0MD3 至 TT0MD0 位=0001)	479
9.6.3	外部触发脉冲输出模式 (TT0MD3 至 TT0MD0 位=0010)	490
9.6.4	单次触发脉冲输出模式 (TT0MD3 至 TT0MD0 位 = 0011)	503
9.6.5	PWM 输出模式 (TT0MD3 至 TT0MD0 位=0100)	510
9.6.6	自由运行定时器模式 (TT0MD3 至 TT0MD0 位=0101)	519
9.6.7	脉冲宽度测量模式 (TT0MD3 至 TT0MD0 位=0110)	537
9.6.8	三角波 PWM 输出模式 (TT0MD3 至 TT0MD0 位 = 0111)	543
9.6.9	编码器计数功能	545
9.6.10	编码器比较模式 (TT0MD3 至 TT0MD0 位 = 1000)	561
第十章	16 位间隔定时器 M (TMM)	569
10.1	概述	569
10.2	配置	570
10.3	寄存器	572
10.4	操作	574
10.4.1	间隔计时器模式	574
10.4.2	注意事项	578
第十一章	电机控制功能	579
11.1	功能概述	579
11.2	配置	580
11.3	控制寄存器	584
11.4	操作	594
11.4.1	系统概述	594
11.4.2	死区时间控制 (产生负相位波形信号)	599
11.4.3	中断分选功能	606
11.4.4	重写带有传送功能的寄存器的操作	613
11.4.5	用于 A/D 转换启动触发信号输出的 TAA4 调谐操作	631
11.4.6	A/D 转换启动触发输出功能	634
第 12 章	实时计数器	639
12.1	实时计数器的功能	639
12.2	实时计数器的配置	640
12.2.1	引脚配置	642
12.2.2	中断功能	642
12.3	控制实时计数器的寄存器	643
12.4	操作	658
12.4.1	初始化设置	658
12.4.2	在时钟工作期间重写各个计数器	659
12.4.3	在时钟工作期间读取各个计数器	660
12.4.4	在时钟工作期间改变 INTRTC0 中断设置	661
12.4.5	在时钟工作期间改变 INTRTC1 中断设置	662

12.4.6	INTRTC2 中断设置初始化	663
12.4.7	在时钟工作期间改变 INTRTC2 中断设置.....	664
12.4.8	实时计数器初始化.....	665
12.4.9	实时计数器的钟表误差修正示例	666
第十三章	看门狗定时器 2 的功能	670
13.1	功能.....	670
13.2	配置.....	671
13.3	寄存器	672
13.4	操作.....	674
第十四章	实时输出功能(RTO).....	675
14.1	功能.....	675
14.2	配置.....	676
14.3	寄存器	678
14.4	操作.....	680
14.5	用法.....	681
14.6	注意事项.....	681
第十五章	A/D 转换器.....	682
15.1	概述.....	682
15.2	功能.....	682
15.3	配置.....	683
15.4	寄存器	686
15.5	操作.....	697
15.5.1	基本操作.....	697
15.5.2	转换操作时序.....	698
15.5.3	触发模式.....	699
15.5.4	操作模式.....	701
15.5.5	掉电比较模式.....	705
15.6	注意事项.....	710
15.7	A/D 转换器特征表的阅读方法.....	714
第十六章	D/A 转换器.....	718
16.1	功能.....	718
16.2	配置.....	718
16.3	寄存器	719
16.4	操作.....	721
16.4.1	正常模式下的操作.....	721
16.4.2	实时输出模式下的操作	721
16.4.3	注意事项.....	722
第十七章	异步串行接口 C (UARTC)	723
17.1	特性.....	723
17.2	配置.....	724
17.3	UARTC 和其它串行接口的模式转换.....	726
17.3.1	CSIF4 和 UARTC0 的模式转换	726

17.3.2	UARTC1 和 I ² C02 的模式转换	727
17.3.3	UARTC2 和 CSIF3 的模式转换	728
17.3.4	UARTC3 和 I ² C00 的模式转换	729
17.3.5	UARTC4、CSIF0 和 I ² C01 的模式转换	730
17.4	寄存器	731
17.5	中断请求信号	741
17.6	操作	742
17.6.1	数据格式	742
17.6.2	SBF 发送/接收格式	744
17.6.3	SBF 发送	746
17.6.4	SBF 接收	747
17.6.5	UART 发送	748
17.6.6	连续发送过程	749
17.6.7	UART 接收	751
17.6.8	接收错误	752
17.6.9	奇偶类型和操作	754
17.6.10	接收数据噪声过滤器	755
17.7	专用波特率发生器	756
17.8	注意事项	764
第十八章	3-线可变长度串行 I/O (CSIF)	765
18.1	CSIF 和其它串行接口的模式转换	765
18.1.1	CSIF4 和 UARTC0 的模式转换	765
18.1.2	CSIF0、UARTC4 和 I ² C01 的模式转换	766
18.1.3	CSIF3 和 UARTC2 的模式转换	767
18.2	特性	768
18.3	配置	769
18.4	寄存器	772
18.5	中断请求信号	779
18.6	操作	780
18.6.1	单次传输模式 (主模式, 发送模式)	780
18.6.2	单次传输模式 (主模式, 接收模式)	782
18.6.3	单次传输模式 (主模式, 发送/接收模式)	784
18.6.4	单次传输模式 (从模式, 发送模式)	786
18.6.5	单次传输模式 (从模式, 接收模式)	788
18.6.6	单次传输模式 (从模式, 发送/接收模式)	790
18.6.7	连续传输模式 (主模式, 发送模式)	792
18.6.8	连续传输模式 (主模式, 接收模式)	794
18.6.9	连续传输模式 (主模式, 发送/接收模式)	797
18.6.10	连续传输模式 (从模式, 发送模式)	801
18.6.11	连续传输模式 (从模式, 接收模式)	803
18.6.12	连续传输模式 (从模式, 发送/接收模式)	806
18.6.13	接收错误	810
18.6.14	时钟时序	811
18.7	输出引脚	813
18.8	波特率发生器	814
18.8.1	波特率的产生	815
18.9	注意事项	816

第十九章 I²C 总线	817
19.1 I²C 总线和其它串行接口的模式转换	817
19.1.1 UARTC3 和 I ² C00 模式转换	817
19.1.2 UARTC4、CSIF0 和 I ² C01 模式转换	818
19.1.3 UARTC1 和 I ² C02 模式转换	819
19.2 特性	820
19.3 配置	821
19.4 寄存器	825
19.5 I²C 总线模式功能	840
19.5.1 引脚配置	840
19.6 I²C 总线的定义和控制方法	841
19.6.1 开始条件	841
19.6.2 地址	842
19.6.3 传输方向指示	843
19.6.4 ACK	844
19.6.5 停止条件	845
19.6.6 等待状态	846
19.6.7 等待状态取消方法	848
19.7 I²C 中断请求信号 (INTIICn)	849
19.7.1 主设备操作	849
19.7.2 从设备操作 (当接收从设备地址数据 (地址匹配) 时)	852
19.7.3 从设备操作 (当接收扩展码时)	856
19.7.4 不通信时的操作	860
19.7.5 仲裁失败操作 (仲裁失败后作为从设备操作)	860
19.7.6 发生仲裁失败时的操作 (在仲裁失败后不进行通信)	862
19.8 中断请求信号 (INTIICn) 的产生时序和等待控制	869
19.9 地址匹配检测方法	871
19.10 错误检测	871
19.11 扩展代码	871
19.12 仲裁	872
19.13 唤醒功能	873
19.14 通信保留	874
19.14.1 使能通信保留功能 (IICFn.IICRSVn 位 = 0) 时	874
19.14.2 禁止通信保留功能时 (IICFn.IICRSVn 位 = 1)	878
19.15 注意事项	879
19.16 通信操作	880
19.16.1 主设备工作于单主系统	881
19.16.2 主设备工作于多主系统	882
19.16.3 从设备操作	885
19.17 数据通信的时序	888
第二十章 USB 功能控制器 USBF	895
20.1 综述	895
20.2 配置	896
20.2.1 框图	896
20.2.2 USB 存储器映射	897
20.3 外部电路配置	898
20.3.1 概述	898

20.3.2	连接配置	899
20.4	注意事项	901
20.5	请求	902
20.5.1	自动请求	902
20.5.2	其它请求	909
20.6	寄存器配置	910
20.6.1	USB 控制寄存器	910
20.6.2	USB 功能控制器寄存器列表	912
20.6.3	EPC 控制寄存器	928
20.6.4	数据保持寄存器	980
20.6.5	EPC 请求数据寄存器	1003
20.6.6	桥寄存器	1018
20.6.7	DMA 寄存器	1022
20.6.8	批量-in 寄存器	1026
20.6.9	批量-out 寄存器	1027
20.6.10	外设控制寄存器	1028
20.7	STALL 握手或无握手	1032
20.8	特定状态下寄存器的值	1033
20.9	FW 处理	1035
20.9.1	初始化处理	1037
20.9.2	中断服务	1040
20.9.3	USB 主处理	1041
20.9.4	挂起/恢复处理	1067
20.9.5	电源应用后的处理	1070
20.9.6	在 DMA 模式下为批量传输 (OUT) 接收数据	1073
20.9.7	在 DMA 模式下为批量传输 (IN) 发送数据	1077
第二十一章	USB 主机控制器 (USBH)	1082
21.1	概述	1082
21.2	配置	1083
21.2.1	框图	1083
21.2.2	USB 主机控制器存储器映射	1084
21.2.3	数据访问的注意事项	1085
21.3	外部电路配置	1086
21.3.1	概述	1086
21.3.2	连接配置	1087
21.3.3	USB 供电	1088
21.4	注意事项	1089
21.5	控制寄存器	1090
21.5.1	USB 控制寄存器	1090
21.6	PCI 主机桥	1091
21.6.1	PCI 主机桥	1091
21.6.2	PCI 主机桥控制器	1092
21.7	OHCI 主机控制器	1103
21.7.1	OHCI 主机控制器功能	1103
21.7.2	OHCI 主机配置寄存器	1104
21.7.3	OHCI 操作寄存器	1117
21.7.4	来自 USB 主机控制器的中断	1145

第二十二章 DMA 功能 (DMA 控制器)	1147
22.1 特性	1147
22.2 配置	1148
22.3 寄存器	1149
22.4 传输对象	1158
22.5 传输模式	1158
22.6 传输类型	1159
22.7 DMA 通道优先级	1160
22.8 与 DMA 传输有关的时间	1160
22.9 DMA 传输启动因素	1161
22.10 DMA 中止因素	1162
22.11 DMA 传输结束	1162
22.12 操作时序	1162
22.13 注意事项	1167
第二十三章 中断/异常处理功能	1172
23.1 特性	1172
23.2 不可屏蔽中断	1183
23.2.1 操作	1185
23.2.2 恢复	1186
23.2.3 NP 标志	1187
23.3 可屏蔽中断	1188
23.3.1 操作	1188
23.3.2 恢复	1190
23.3.3 可屏蔽中断的优先级	1191
23.3.4 中断控制寄存器 (xxlCn)	1195
23.3.5 中断屏蔽寄存器 0 至 5 (IMR0 至 IMR5)	1199
23.3.6 正在服务中优先级寄存器 (ISPR)	1201
23.3.7 ID 标志	1202
23.3.8 看门狗定时器模式寄存器 2 (WDTM2)	1202
23.4 软件异常	1203
23.4.1 操作	1203
23.4.2 恢复	1204
23.4.3 EP 标志	1205
23.5 异常陷阱	1206
23.5.1 非法操作码	1206
23.5.2 调试陷阱	1208
23.6 外部中断请求输入引脚 (NMI 和 INTP00 至 INTP18)	1210
23.6.1 噪声消除	1210
23.6.2 边沿检测	1210
23.7 CPU 中断响应时间	1218
23.8 中断不被 CPU 响应期间	1219
23.9 注意事项	1219
第二十四章 按键中断功能	1220
24.1 功能	1220
24.2 寄存器	1221
24.3 注意事项	1221

第二十五章 待机功能	1222
25.1 概述	1222
25.2 寄存器	1224
25.3 HALT 模式	1227
25.3.1 设置和操作状态	1227
25.3.2 解除 HALT 模式	1227
25.4 IDLE1 模式	1229
25.4.1 设置和操作状态	1229
25.4.2 解除 IDLE1 模式	1230
25.5 IDLE2 模式	1232
25.5.1 设置和操作状态	1232
25.5.2 解除 IDLE2 模式	1233
25.5.3 解除 IDLE2 模式时确保建立时间	1235
25.6 STOP 模式	1236
25.6.1 设置和操作状态	1236
25.6.2 解除 STOP 模式.....	1236
25.6.3 解除 STOP 模式时保障振荡稳定时间	1239
25.7 副时钟操作模式	1240
25.7.1 设置和操作状态	1240
25.7.2 解除副时钟操作模式	1240
25.8 sub-IDLE 模式	1242
25.8.1 设置和操作状态	1242
25.8.2 解除 sub-IDLE 模式.....	1242
第二十六章 复位功能	1244
26.1 概述	1244
26.2 检查复位源的寄存器	1245
26.3 操作	1246
26.3.1 通过 RESET 引脚的复位操作	1246
26.3.2 通过看门狗定时器 2 的复位操作	1248
26.3.3 通过低电压检测器的复位操作	1250
26.3.4 复位解除后的操作.....	1251
26.3.5 复位功能操作流程.....	1252
第二十七章 时钟监控器	1253
27.1 功能	1253
27.2 配置	1253
27.3 寄存器	1254
27.4 操作	1255
第二十八章 低电压检测器 (LVI)	1258
28.1 功能	1258
28.2 配置	1258
28.3 寄存器	1259
28.4 操作	1261
28.4.1 使用内部复位信号.....	1261
28.4.2 用作中断	1262

28.5	RAM 保持电压检测操作	1263
第二十九章	CRC 功能	1264
29.1	功能	1264
29.2	配置	1264
29.3	寄存器	1265
29.4	操作	1266
29.5	使用方法	1267
第三十章	稳压器	1269
30.1	概述	1269
30.2	操作	1270
第三十一章	FLASH 存储器	1271
31.1	特性	1271
31.2	存储器配置	1272
31.3	功能概述	1273
31.4	通过 flash 存储器编程器进行重写	1276
31.4.1	编程环境.....	1276
31.4.2	通信模式.....	1277
31.4.3	Flash 存储器控制.....	1288
31.4.4	通信模式的选择.....	1289
31.4.5	通信命令.....	1290
31.4.6	引脚连接.....	1291
31.5	通过自编程进行重写	1295
31.5.1	概述.....	1295
31.5.2	特性.....	1296
31.5.3	标准自编程流程.....	1297
31.5.4	Flash 功能.....	1298
31.5.5	引脚处理.....	1298
31.5.6	用到的内部资源.....	1299
第三十二章	片上调试功能	1300
32.1	用 DCU 调试	1301
32.1.1	连接电路示例.....	1301
32.1.2	接口信号.....	1301
32.1.3	可屏蔽功能.....	1303
32.1.4	寄存器.....	1303
32.1.5	操作.....	1305
32.1.6	注意事项.....	1305
32.2	不使用 DCU 调试	1306
32.2.1	电路连接示例.....	1306
32.2.2	可屏蔽功能.....	1309
32.2.3	用户资源的保留.....	1310
32.2.4	注意事项.....	1317
32.3	ROM 安全功能	1318
32.3.1	安全 ID.....	1318

32.3.2 设置	1319
第三十三章 电气规范	1321
33.1 最大额定值	1321
33.2 电容	1323
33.3 工作条件	1323
33.4 振荡器特性	1324
33.4.1 主时钟振荡器特性	1324
33.4.2 副时钟振荡器特性	1325
33.4.3 PLL 特性	1326
33.4.4 内部振荡器特性	1326
33.5 DC 特性	1327
33.5.1 输入/输出电平	1327
33.5.2 供电电流	1329
33.6 数据保持特性	1330
33.7 AC 特性	1331
33.7.1 CLKOUT 输出时序	1332
33.7.2 总线时序	1333
33.8 基本操作	1340
33.9 Flash 存储器编程特性	1352
第三十四章 封装图	1354
附录 A 开发工具	1356
A.1 软件包	1358
A.2 语言处理软件	1358
A.3 控制软件	1358
A.4 调试工具(硬件)	1359
A.4.1 使用 IECUBE QB-V850ESJX3H ^注	1359
A.4.2 使用 MINICUBE QB-V850MINI	1362
A.4.3 使用 MINICUBE2 QB-MINI2	1363
A.5 调试工具(软件)	1364
A.6 嵌入式软件	1365
A.7 Flash 存储器烧写工具	1365
附录 B V850ES/Jx3-U 和 V850ES/Jx3 的主要区别	1366
附录 C 寄存器索引	1367
附录 D 指令集列表	1381
D.1 规则	1381
D.2 指令集 (按字母先后次序)	1384

第一章 引言

V850ES/JG3-U 和 V850ES/JH3-U 是 NEC 电子 V850 单片微控制器低功耗系列的产品，专门针对实时控制应用而设计。

1.1 概述

V850ES/JG3-U 和 V850ES/JH3-U 是 32 位单片微控制器，它包括 V850ES CPU 内核以及相关外设功能，例如 ROM/RAM、定时器/计数器、串行接口、A/D 转换器、D/A 转换器、DMA 控制器、USB 功能控制器以及 USB 主机控制器。

除了高速实时响应特性和单时钟基本指令之外，V850ES/JG3-U 和 V850ES/JH3-U 还支持硬件乘法器实现的乘法指令、饱和运算指令、位操作指令。

表 1-1 列出了 V850ES/JG3-U 的产品，表 1-2 列出了 V850ES/JH3-U 的产品。

表 1-1. V850ES/JG3-U 产品列表

类属名		V850ES/JG3-U	
部件编号		μ PD70F3763	μ PD70F3764
内部存储器	Flash 存储器	384KB	512 KB
	RAM	48 KB	56 KB
存储空间	逻辑空间	64 MB	
	外部存储区	64KB	
外部总线接口		地址数据总线: 16 复用总线	
通用寄存器		32 位 \times 32 个寄存器	
时钟	主时钟	PLL 模式下: $f_x = 3$ 至 6 MHz, $f_{xx} = 24$ 至 48 MHz (8 倍频), 在时钟直通模式下: $f_x = 3$ 至 6 MHz (内部: $f_{xx} = 3$ 至 6 MHz)	
	副时钟	$f_{XT} = 32.768$ kHz	
	内部振荡器	$f_R = 220$ kHz (典型值)	
	最短指令执行时间	20.8 ns (主时钟 (f_{xx}) = 48 MHz)	
I/O 端口 (5V 耐压)		I/O: 75 (22)	
定时器	16 位 TAA	6 通道 (包括一个仅可用作间隔功能的通道)	
	16 位 TAB	2 通道	
	16 位 TMM	4 通道	
	16 位 TMT	1 通道	
	电机控制	1 通道 (与 TAA 和 TAB 联合的功能; 包括高阻输出控制功能)	
	钟表定时器	1 通道 (RTC)	
	WDT	1 通道	
实时输出端口。		6 位 \times 1 通道	
10 位 A/D 转换器		12 通道	
8 位 D/A 转换器		2 通道	
串行接口	CSIF/UARTC	2 通道	2 通道
	CSIF/UARTC/I ² C	1 通道	1 通道
	CSIF	2 通道	2 通道
	UARTC/I ² C	2 通道	2 通道
	USB 主机	1 通道	1 通道
	USB 功能	1 通道	1 通道
DMA 控制器		4 通道 (传输目标: 片上外设 I/O, 内部 RAM, 外部存储器)	
中断源	外部 ^{注 2,3}	15 (15)	15 (15)
	内部	72	72
省电功能		HALT/IDLE1/IDLE2/STOP/副时钟/sub-IDLE 模式	
复位源		RESET 引脚输入, 看门狗定时器 2 (WDT2), 时钟监视器 (CLM), 低压检测器 (LVI)	
片上调试		支持 MINICUBE [®] 、MINICUBE2	
工作电源电压		2.85 至 3.6 V	
运行环境温度		-40 至 +85°C	
封装		100 针 LQFP (密间距) (14 \times 14 mm)	

- 注
1. 包括 8 KB 仅限数据的 RAM 区域。
 2. 括号中的数字表示能够解除 STOP 模式的外部中断数量。
 3. 包括 NMI。

表 1-2. V850ES/JH3-U 产品列表

类属名		V850ES/JH3-U	
部件编号		μ PD70F3768	μ PD70F3769
内部存储器	Flash 存储器	384KB	512 KB
	RAM	48 KB	56 KB
存储空间	逻辑空间	64 MB	
	外部存储区	13 MB	
外部总线接口		地址总线: 24 地址数据总线: 16 分离总线 / 复用总线	
通用寄存器		32 位 \times 32 个寄存器	
时钟	主时钟	PLL 模式下: $f_x = 3$ 至 6 MHz, $f_{xx} = 24$ 至 48 MHz (8 倍频), 在时钟直通模式下: $f_x = 3$ 至 6 MHz (内部: $f_{xx} = 3$ 至 6 MHz)	
	副时钟	$f_{XT} = 32.768$ kHz	
	内部振荡器	$f_R = 220$ kHz (典型值)	
	最短指令执行时间	20.8 ns (主时钟 (f_{xx}) = 48 MHz)	
I/O 端口 (5V 耐压)		I/O: 96 (27)	
定时器	16 位 TAA	6 通道 (包括一个仅可用作间隔功能的通道)	
	16 位 TAB	2 通道	
	16 位 TMM	4 通道	
	16 位 TMT	1 通道	
	电机控制	1 通道 (与 TAA 和 TAB 联合的功能; 包括高阻输出控制功能)	
	钟表定时器	1 通道 (RTC)	
	WDT	1 通道	
实时输出端口。		6 位 \times 1 通道	
10 位 A/D 转换器		12 通道	
8 位 D/A 转换器		2 通道	
串行接口	CSIF/UARTC	2 通道	2 通道
	CSIF/UARTC/I ² C	1 通道	1 通道
	CSIF	2 通道	2 通道
	UARTC/I ² C	2 通道	2 通道
	USB 主机	1 通道	1 通道
	USB 功能	1 通道	1 通道
DMA 控制器		4 通道 (传输目标: 片上外设 I/O, 内部 RAM, 外部存储器)	
中断源	外部 ^{2,3}	20 (20)	
	内部	72	72
省电功能		HALT/IDLE1/IDLE2/STOP/副时钟/sub-IDLE 模式	
复位源		RESET 引脚输入, 看门狗定时器 2 (WDT2), 时钟监视器 (CLM), 低压检测器 (LVI)	
片上调试		支持 MINICUBE [®] 、MINICUBE2	
工作电源电压		2.85 至 3.6 V	
运行环境温度		-40 至 +85°C	
封装		128 针 LQFP (密间距) (14 \times 20 mm)	

- 注
1. 包括 8 KB 仅限数据的 RAM 区域。
 2. 括号中的数字表示能够解除 STOP 模式的外部中断数量。
 3. 包括 NMI。

1.2 特性

- 最短指令执行周期: 20.8ns (主时钟 (f_{xx}) = 48 MHz: V_{DD}= 2.85 至 3.6 V)
30.5μs (副时钟 (f_{xT}) = 32.768 kHz)
- 通用寄存器: 32 位 × 32 个寄存器
- CPU 特性: 带符号乘法 (16 × 16 → 32): 1 或 2 个时钟。
带符号乘法 (32 × 32 → 64): 1 至 5 个时钟。
溢出处理 (包括上、下溢出检测功能)。
32 位移位指令: 1 个时钟
位操作指令。
长/短两种格式的数据装载/存储指令。
- 存储空间: 64 MB 线性地址空间 (用于程序和数据)
外部扩展: 最多 16 MB (包括用作内部 ROM/RAM 的 1 MB 空间)
 - 内部存储器: RAM: 48/56 KB (参见表 1-1 和表 1-2)
Flash 存储器: 384/512 KB (参见表 1-1 和表 1-2)
 - 外部总线接口: 可选择分离总线/复用总线输出
(V850ES/JG3-U 中只有复用总线)
8 位/16 位数据总线功能
等待功能
 - 可编程等待功能。
 - 外部等待功能
 空闲状态功能
总线保持功能
- 中断和异常处理:

		内部			外部		
		不可屏蔽	可屏蔽	总共	不可屏蔽	可屏蔽	总共
V850ES/JG3-U	μPD70F3763	1	71	72	1	14	15
	μPD70F3764	1	71	72	1	14	15
V850ES/JH3-U	μPD70F3768	1	71	72	1	19	20
	μPD70F3769	1	71	72	1	19	20

软件异常: 32 个中断源

异常陷阱: 2 个中断源

- I/O 线: I/O 端口: 75 (V850ES/JG3-U)
96 (V850ES/JH3-U)
- 定时器功能: 16 位间隔定时器 M (TMM): 4 通道
16 位定时器/事件计数器 AA (TAA): 6 通道
16 位定时器/事件计数器 AB (TAB): 2 通道
16 位定时器/事件计数器 T (TMT): 1 通道
电机控制功能 (使用的定时器: TAB1, TAA4)
16 位精度的 6 相带死区功能的 PWM 功能
高阻抗输出控制功能
通过定时器调谐操作功能产生 A/D 触发
任意周期设置功能
任意死区时间设置功能
实时计数器 (RTC): 1 通道
看门狗定时器: 1 通道

- 实时输出端口: 6 位×1 通道
- 串行接口: 异步串行接口 C (UARTC)
3 线长度可变串行接口 F (CSIF)
I²C 总线接口 (I²C)
USB 主机接口
USB 功能接口
 - UARTC/ CSIF 2 通道
 - UARTC/ CSIF/ I2C 1 通道
 - UARTC/I2C 2 通道
 - CSIF 2 通道
 - USB 主机 1 通道
 - USB 功能 1 通道
- A/D 转换器: 10 位分辨率: 12 通道
- D/A 转换器: 8 位分辨率: 2 通道
- DMA 控制器: 4 通道
- DCU (调试控制单元): JTAG 接口
- 时钟发生器: 在主时钟或副时钟操作期间
有 7 个级别 CPU 时钟可选 (fxx, fxx/2, fxx/4, fxx/8, fxx/16, fxx/32, fxt)
时钟直通模式/PLL 模式可选
- 内部振荡时钟: 220 kHz (典型值)
- 省电功能: HALT/IDLE1/IDLE2/STOP/副时钟/sub-IDLE 模式
- 封装: 100 针塑封 LQFP (密间距) (14 × 14) (V850ES/JG3-U)
128 针塑封 LQFP (密间距) (14 × 20) (V850ES/JH3-U)

1.3 应用领域

需要 USB 接口的设备，比如家庭音响系统，打印机和扫描仪。

1.4 订购信息

- V850ES/JG3-U

部件编号	封装	内部 Flash 存储器
μ PD70F3763GC-UEU-AX	100 针塑封 LQFP (密间距) (14 × 14)	384 KB
μ PD70F3764GC-UEU-AX	100 针塑封 LQFP (密间距) (14 × 14)	512 KB

- V850ES/JH3-U

部件编号	封装	内部 Flash 存储器
μ PD70F3768GF-GAT-AX	128 针塑封 LQFP (密间距) (14 × 20)	384 KB
μ PD70F3769GF-GAT-AX	128 针塑封 LQFP (密间距) (14 × 20)	512 KB

备注 V850ES/JG3-U 和 V850ES/JH3-U 微控制器是无铅产品。

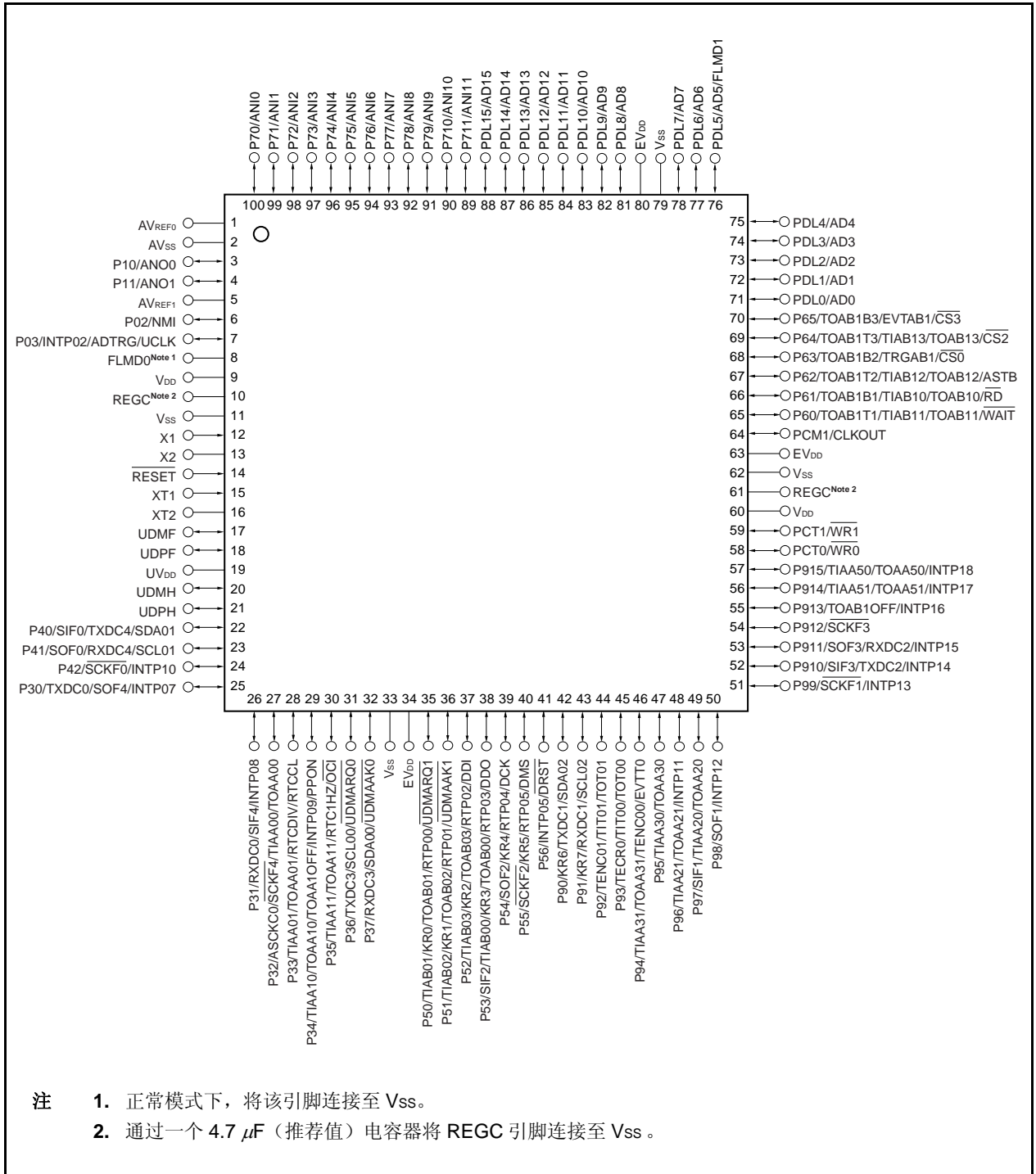
1.5 引脚配置 (顶视图)

• V850ES/JG3-U

100 针塑封 LQFP (密间距) (14 × 14)

μPD70F3763GC-UEU-AX

μPD70F3764GC-UEU-AX



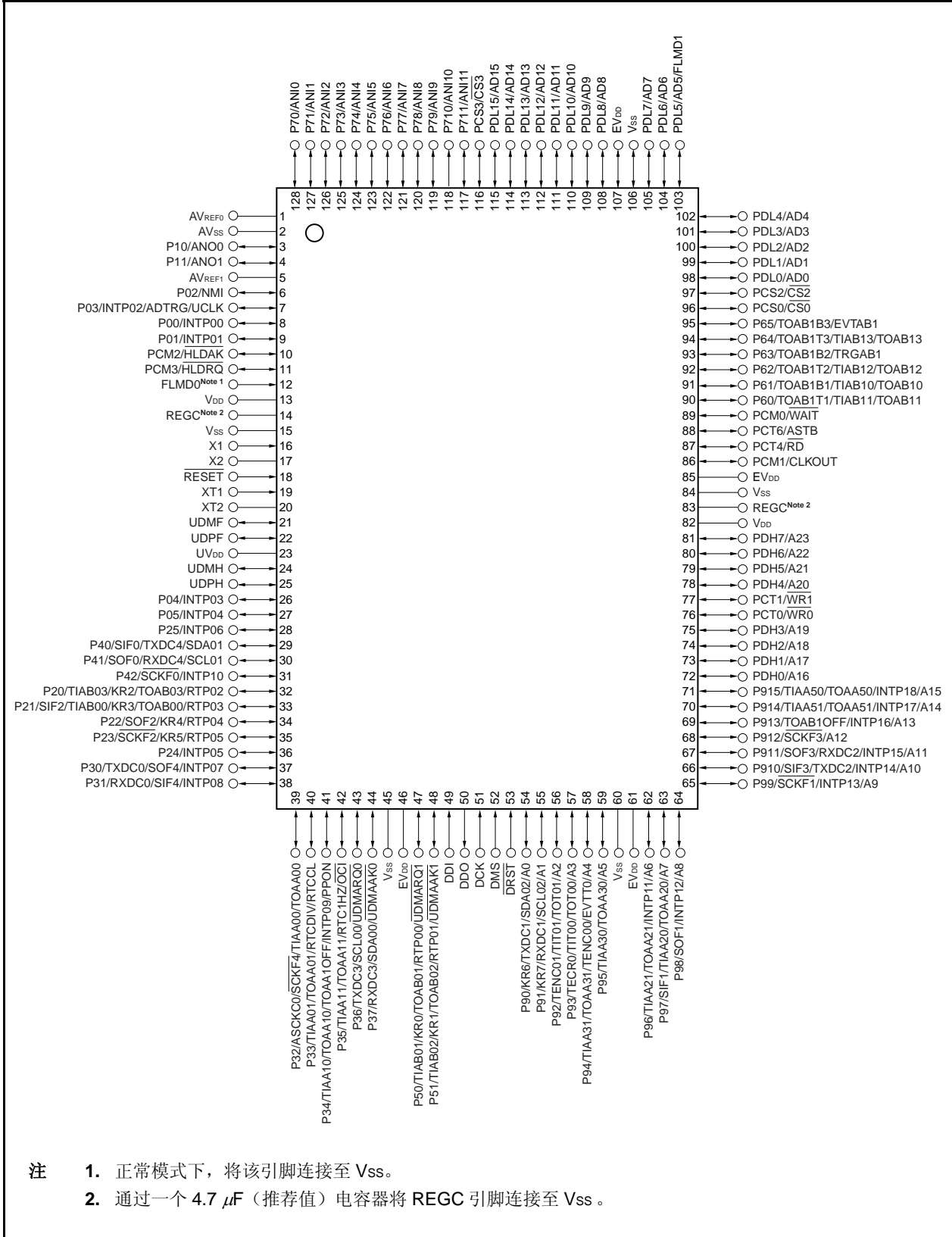
- 注
1. 正常模式下, 将该引脚连接至 Vss。
 2. 通过一个 4.7 μF (推荐值) 电容器将 REGC 引脚连接至 Vss。

• V850ES/JH3-U

128 针塑封 LQFP (密间距) (14 × 20)

μPD70F3768GF-GAT-AX

μPD70F3769GF-GAT-AX



- 注
1. 正常模式下, 将该引脚连接至 Vss。
 2. 通过一个 4.7 μF (推荐值) 电容器将 REGC 引脚连接至 Vss。

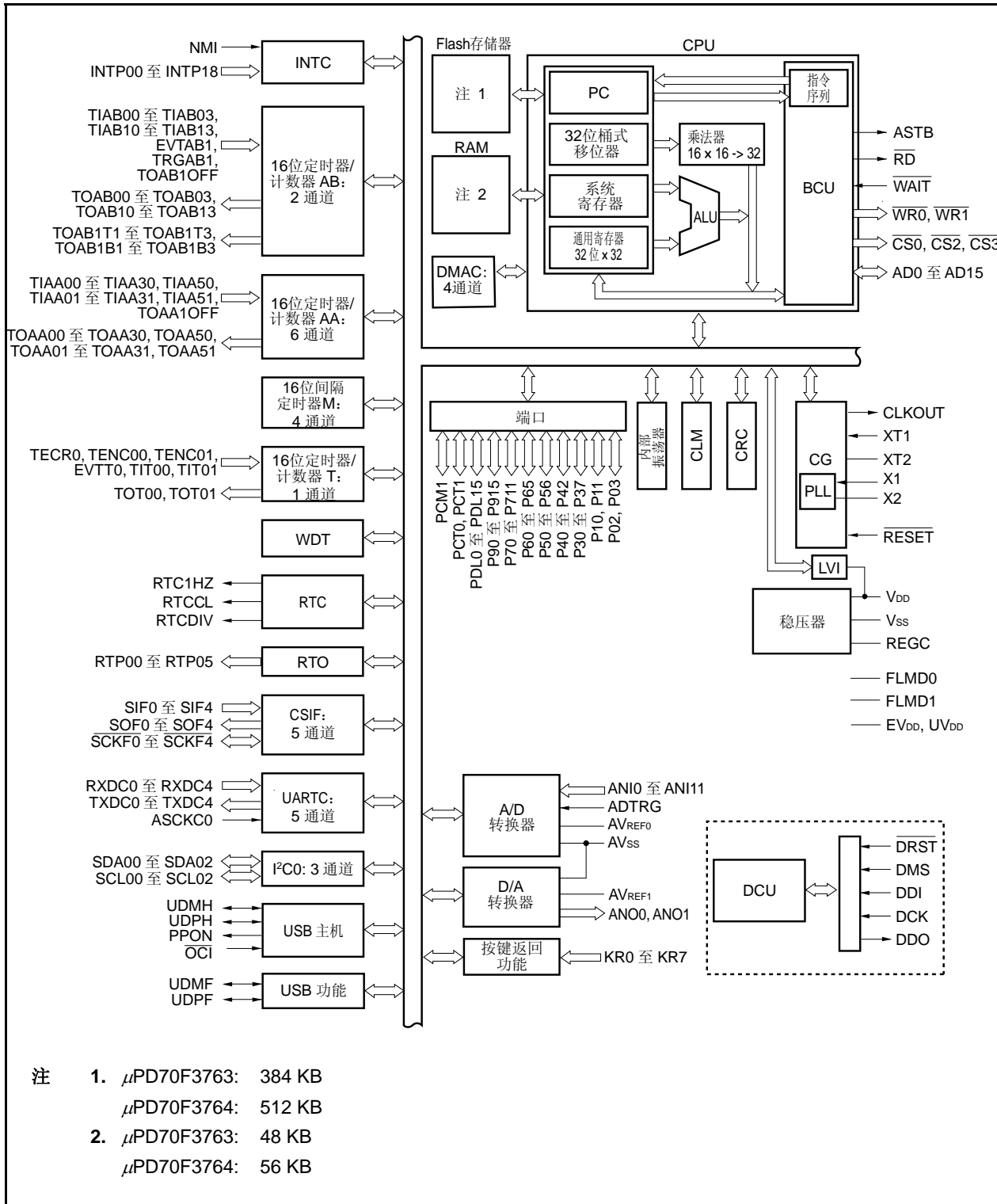
引脚名称

A0 至 A23:	地址总线	RTP00 至 RTP05:	实时输出端口
AD0 至 AD15:	地址/数据总线	RXDC0 至 RXDC4:	接收数据
ADTRG:	A/D 触发信号输入	$\overline{\text{SCKF0}}$ 至 $\overline{\text{SCKF4}}$:	串行时钟
ANI0 至 ANI11:	模拟输入	$\overline{\text{SCL00}}$ 至 $\overline{\text{SCL02}}$:	串行时钟
ANO0, ANO1:	模拟输出	SDA00 至 SDA02:	串行数据
ASCKC0:	异步串行时钟	SIF0 至 SIB4:	串行输入
ASTB:	地址选通	SOF0 至 SOF4:	串行输出
AVREF0, AVREF1:	模拟参考电压	TECR0:	定时器编码器清除输入
AVSS:	模拟 V _{SS}	TENC00, TENC01:	定时器编码器输入
CLKOUT:	时钟输出	TIAA00, TIAA01, TIAA10, TIAA11, TIAA20, TIAA21, TIAA30, TIAA31, TIAA50, TIAA51, TIAB00 至 TIAB03, TIAB10, TIAB13, TIT00, TIT01:	定时器输入
$\overline{\text{CS0}}$, $\overline{\text{CS2}}$, $\overline{\text{CS3}}$:	片选	TOAA00, TOAA01, TOAA10, TOAA11, TOAA20, TOAA21, TOAA30, TOAA31, TOAA50 至 TOAA51, TOAB00 至 TOAB03, TOAB10 至 TOAB13, TOAB1B1 至 TOAB1B3, TOAB1T1 至 TOAB1T3, TOT00, TOT01:	定时器输出
DCK:	调试时钟	TOAA10FF, TOAB10FF:	定时器输出关闭
DDI:	调试数据输入	TRGAB1:	定时器触发输入
DDO:	调试数据输出	TXDC0 至 TXDC4:	发送数据
DDMS:	调试方式选择	UCLK:	USB 时钟
$\overline{\text{DRST}}$:	调试复位	UDMAAK0, $\overline{\text{UDMAAK1}}$: $\overline{\text{UDMARQ0}}$, $\overline{\text{UDMARQ1}}$:	对外部 USB 的 DMA 应答 对外部 USB 的 DMA 请求
EVDD:	外部引脚的电源	UDMF:	USB 数据 I/O (-) 功能
EVTTO, EVTAB1:	外部事件计数输入	UDMH:	USB 数据 I/O (-) 主机
FLMD0, FLMD1:	Flash 编程模式	UDPF:	USB 数据 I/O (+) 功能
$\overline{\text{HLDK}}$:	保持应答	UDPH:	USB 数据 I/O (+) 主机
$\overline{\text{HLDRQ}}$:	保持请求	UVDD:	USB 的电源
INTP00 至 INTP18:	外部中断输入	V _{DD} :	电源
KR0 至 KR7:	按键返回	V _{SS} :	地
NMI:	不可屏蔽中断请求	$\overline{\text{WAIT}}$:	等待
$\overline{\text{OCI}}$:	过流检测输入	$\overline{\text{WR0}}$:	低字节写选通
P00 至 P05:	端口 0	$\overline{\text{WR1}}$:	高字节写选通
P10, P11:	端口 1	X1, X2:	主时钟晶振
P20 至 P25:	端口 2	XT1, XT2:	副时钟晶振
P30 至 P37:	端口 3		
P40 至 P42:	端口 4		
P50 至 P56:	端口 5		
P60 至 P65:	端口 6		
P70 至 P711:	端口 7		
P90 至 P915:	端口 9		
PCM0 至 PCM3:	端口 CM		
PCS0, PCS2, PCS3:	端口 CS		
PCT0, PCT1, PCT4, PCT6:	端口 CT		
PDH0 至 PDH7:	端口 DH		
PDL0 至 PDL15:	端口 DL		
PPON:	USB 的电源供给输出		
$\overline{\text{RD}}$:	读选通		
REGC:	稳压器控制		
$\overline{\text{RESET}}$:	系统复位		
RTC1HZ, RTCCL, RTCDIV:	实时计数器时钟输出		

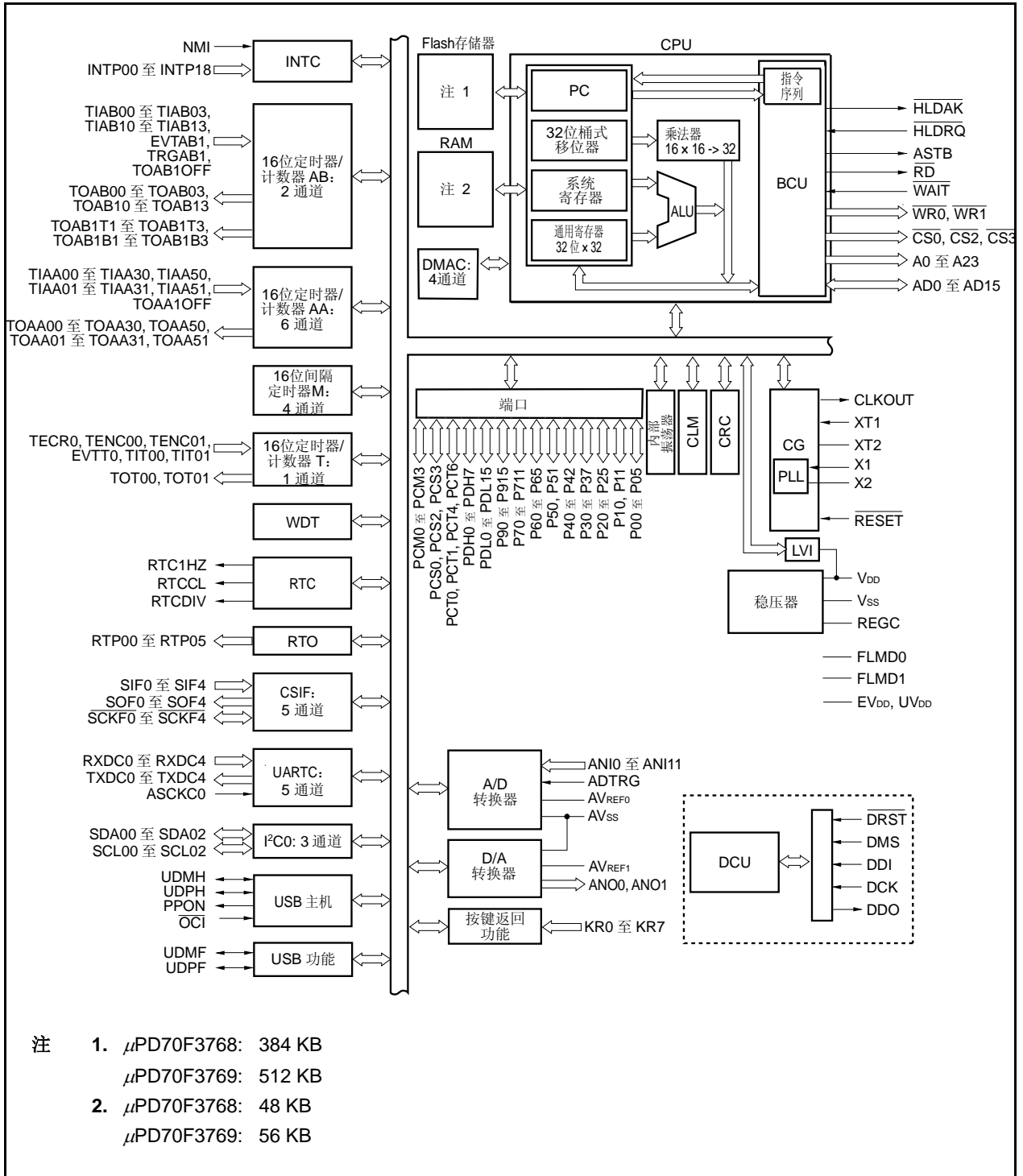
1.6 功能框图结构

1.6.1 内部框图

• V850ES/JG3-U



• V850ES/JH3-U



1.6.2 内部单元

(1) CPU

CPU 使用五级流水线控制，单时钟周期内可以完成地址运算、算术逻辑运算、数据传输以及几乎所有指令的处理。

其它片内固有硬件，诸如乘法器（16 位×16 位→32 位）和桶式移位器（32 位），加速处理复杂操作。

(2) 总线控制单元（BCU）

总线控制单元根据从 CPU 获取的物理地址，启动所需的外部总线周期。当从外部存储空间取到指令且 CPU 不发送总线周期开始请求时，BCU 将产生预取地址和预取指令代码。预取指令代码存储在指令队列中。

(3) Flash 存储器（ROM）

这是一个 512/384 KB flash 存储器，其映射地址为 0000000H 至 007FFFFH /0000000H 至 005FFFFH。

取指令期间，CPU 可在一个时钟周期内访问 Flash 存储器（ROM）。

(4) RAM

这是一个 48/40 KB RAM，其映射地址为 3FF3000H 至 3FF5000H/3FF5000H 至 3FF5000H。CPU 可在一个时钟周期内完成 RAM 数据存取访问。还包括地址 00280000H 至 00281FFFH 上的 8KB 仅限数据使用的 RAM。

(5) 中断控制器（INTC）

该控制器用来处理片上外设和外部硬件产生的硬件中断请求（NMI、INTP00 至 INTP18）。可以为这些中断请求指定 8 个级别的中断优先级，同时也能执行多重服务控制。

(6) 时钟发生器（CG）

提供主时钟振荡器和副时钟振荡器，并且产生相应的主时钟振荡频率（fx）和副时钟振荡频率（fx_T）。有两种模式：在时钟直通模式中，fx 直接用作主时钟频率（fxx）。在 PLL 模式下，使用 fx 的 8 倍频。

CPU 工作频率（f_{cpu}）可以有以下选择：fxx，fxx/2，fxx/4，fxx/8，fxx/16，fxx/32，和 fx_T。

(7) 内部振荡器

芯片提供内部振荡器。振荡频率为 220 kHz（典型值）。内部振荡器为看门狗定时器 2 和定时器 M 提供时钟。

(8) 定时器/计数器

芯片提供了六通道的 16 位定时器/事件计数器 AA（TAA），二通道的 16 位定时器/事件计数器 AB（TAB），一通道的 16 位定时器/事件计数器 T（TMT）和四通道的 16 位间隔定时器 M（TMM）。电机控制功能可以通过 TAB1 和 TAA4 的联合使用来实现。

(9) 实时计数器 (用作钟表)

实时计数器基于副时钟 (32.768 kHz) 或主时钟计数来计算参考时间周期 (一秒钟)。同时也可以基于主时钟用作间隔定时器。年、月、日、星期、小时、分、秒都提供有专用的硬件计数器, 可以计数达 99 年。

(10) 看门狗定时器 2

片内集成的看门狗定时器用于检测无意中形成的程序死循环、系统异常等。

内部振荡时钟、主时钟、或副时钟都可以被选为源时钟。

溢出发生后, 看门狗定时器 2 产生一个不可屏蔽中断请求信号 (INTWDT2) 或系统复位信号 (WDT2RES)。

(11) 串行接口

V850ES/JG3-U 和 V850ES/JH3-U 包含三种串行接口 (异步串行接口 C (UARTC)、3 线长度可变串行接口 F (CSIF)、和 I²C 总线接口 (I²C)), 一个 USB 功能控制器 (USBF), 以及一个 USB 主机控制器 (USBH)。

UARTC 时, 数据经由 TXDC0 引脚至 TXDC2 引脚和 RXDC0 引脚至 RXDC2 引脚进行传输

CSIF 时, 数据经由 SOF0 引脚至 SOF4 引脚、SIF0 引脚至 SIF4 引脚、和 SCKF0 引脚至 SCKF4 引脚进行传输。

I²C 时, 数据经由 SDA00 引脚至 SDA02 引脚和 SCL00 引脚至 SCL02 引脚进行传输。

USBF 通过 UDMF 和 UDPF 引脚传输数据。

USBH 通过 UDMH 和 UDPH 引脚传输数据。

(12) A/D 转换器

10 位 A/D 转换器包含 12 个模拟输入引脚。使用逐次逼近的方式进行转换。

(13) D/A 转换器

片内集成了一个 2 通道的 8 位 D/A 转换器, 采用 R-2R 阶梯变换方式。

(14) DMA 控制器

芯片提供了一个 4 通道的 DMA 转换器。根据片上外设 I/O 设备发出的中断请求, 该控制器在内部 RAM 和片上外设 I/O 设备, 以及外部存储器之间进行数据传输。

(15) 按键中断功能

有下降沿输入到按键输入引脚 (8 个通道), 将产生按键中断请求信号 (INTKR)。

(16) 实时输出功能

定时器比较寄存器匹配信号产生时, 实时输出功能将预置的 6 位数据传送至输出锁存器。

(17) CRC 功能

芯片提供 CRC 操作电路, 该电路根据 8 位数据的设置产生 16 位 CRC (循环冗余检测) 码。

(18) DCU (调试控制单元)

提供使用 JTAG (Joint Test Action Group) 通讯规范的片内调试功能。通过控制引脚输入电平和 OCDM 寄存器来进行正常端口功能和片内调试功能之间的切换。

(19) 端口

可使用以下通用寄存器端口功能和控制引脚功能：

• V850ES/JG3-U

端口	I/O	复用功能
P0	2 位 I/O	NMI、外部中断、A/D 转换器触发、串行接口
P1	2 位 I/O	D/A 转换器模拟输出
P3	10 位 I/O	外部中断、实时计数器、串行接口、定时器 I/O
P4	3 位 I/O	串行接口、外部中断
P5	7 位 I/O	定时器 I/O、串行接口、实时输出、按键中断输入、调试 I/O
P6	6 位 I/O	外部控制信号、定时器 I/O、外部控制信号
P7	12 位 I/O	A/D 转换器模拟输入
P9	16 位 I/O	串行接口、按键中断输入、定时器 I/O、外部中断
PCM	1 位 I/O	外部控制信号
PCT	2 位 I/O	外部控制信号
PDL	16 位 I/O	外部地址/数据总线

• V850ES/JH3-U

端口	I/O	复用功能
P0	6 位 I/O	NMI、外部中断、A/D 转换器触发、串行接口
P1	2 位 I/O	D/A 转换器模拟输出
P2	6 位 I/O	定时器 I/O、实时输出、按键中断输入、串行接口
P3	10 位 I/O	外部中断、实时计数器、串行接口、定时器 I/O
P4	3 位 I/O	串行接口、外部中断
P5	2 位 I/O	定时器 I/O、实时输出、按键中断输入
P6	6 位 I/O	外部控制信号、定时器 I/O
P7	12 位 I/O	A/D 转换器模拟输入
P9	16 位 I/O	外部地址总线、串行接口、按键中断输入、定时器 I/O、外部中断
PCM	4 位 I/O	外部控制信号
PCS	3 位 I/O	外部控制信号
PCT	4 位 I/O	外部控制信号
PDH	8 位 I/O	外部地址总线
PDL	16 位 I/O	外部地址/数据总线

第二章 引脚功能

2.1 引脚功能列表

下表描述了 V850ES/JG3-U 和 V850ES/JH3-U 单片机中引脚的名称和功能。

有四种类型的引脚 I/O 缓冲器电源：AVREF0、AVREF1、EVDD 和 UVDD。这些电源和引脚的关系如下：

表 2-1. 引脚 I/O 缓冲器电源

电源	对应引脚	
	V850ES/JG3-U	V850ES/JH3-U
AVREF0	端口 7	端口 7
AVREF1	端口 1	端口 1
EVDD	RESET, 端口 0, 3 至 6, 9, CM, CT, DL	RESET, 端口 0, 2 至 6, 9, CM, CS, CT, DH, DL
UVDD	UDPF, UDMF, UDPH, UDMH	UDPF, UDMF, UDPH, UDMH

(1) 端口引脚

(1/4)

引脚名称	I/O	功能	复用功能	引脚编号	
				JG3-U	JH3-U
P00	I/O	端口 0 6 位 I/O 端口 (V850ES/JH3-U) 2 位 I/O 端口 (V850ES/JG3-U) 可按位指定输入/输出。	INTP00	-	8
P01			INTP01	-	9
P02			NMI	6	6
P03			INTP02/ADTRG/UCLK	7	7
P04			INTP03	-	26
P05			INTP04	-	27
P10	I/O	端口 1 2 位 I/O 端口 可按位指定输入/输出。	ANO0	3	3
P11			ANO1	4	4
P20	I/O	端口 2 6 位 I/O 端口 可按位指定输入/输出。	TIAB03/KR2/TOAB03/RTP02	-	32
P21			SIF2/TIAB00/KR3/TOAB00/RTP03	-	33
P22			SOF2/KR4/RTP04	-	34
P23			$\overline{\text{SCKF2}}/\text{KR5/RTP05}$	-	35
P24			INTP05	-	36
P25			INTP06	-	28
P30	I/O	端口 3 8 位 I/O 端口 可按位指定输入/输出。	TXDC0/SOF4/INTP07	25	37
P31			RXDC0/SIF4/INTP08	26	38
P32			ASCKC0/ $\overline{\text{SCKF4}}$ /TIAA00/TOAA00	27	39
P33			TIAA01/TOAA01/RTCDIV/RTCCL	28	40
P34			TIAA10/TOAA10/TOAA1OFF/INTP09 /PPON	29	41
P35			TIAA11/TOAA11/RTC1HZ/ $\overline{\text{OCI}}$	30	42
P36			TXDC3/SCL00/ $\overline{\text{UDMARQ0}}$	31	43
P37			RXDC3/SDA00/ $\overline{\text{UDMAAK0}}$	32	44
P40	I/O	端口 4 3 位 I/O 端口 可按位指定输入/输出。	SIF0/TXDC4/SDA01	22	29
P41			SOF0/RXDC4/SCL01	23	30
P42			$\overline{\text{SCKF0}}$ /INTP10	24	31
P50	I/O	端口 5 2 位 I/O 端口 (V850ES/JH3-U) 7 位 I/O 端口 (V850ES/JG3-U) 可按位指定输入/输出。	TIAB01/KR0/TOAB01/RTP00 / $\overline{\text{UDMARQ1}}$	35	47
P51			TIAB02/KR1/TOAB02/RTP01 / $\overline{\text{UDMAAK1}}$	36	48
P52			TIAB03/KR2/TOAB03/RTP02/DDI	37	-
P53			SIF2/TIAB00/KR3/TOAB00 /RTP03/DDO	38	-
P54			SOF2/KR4/RTP04/DCK	39	-
P55			$\overline{\text{SCKF2}}/\text{KR5/RTP05/DMS}$	40	-
P56			INTP05/DRST	41	-

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

(2/4)

引脚名称	I/O	功能	复用功能	引脚编号	
				JG3-U	JH3-U
P60	I/O	端口 6 6 位 I/O 端口 可按位指定输入/输出。	TOAB1T1/TOAB11/TIAB11/ $\overline{\text{WAIT}}$	65	-
			TOAB1T1/TOAB11/TIAB11	-	90
P61			TOAB1B1/TIAB10/TOAB10/ $\overline{\text{RD}}$	66	-
			TOAB1B1/TIAB10/TOAB10	-	91
P62			TOAB1T2/TOAB12/TIAB12/ASTB	67	-
			TOAB1T2/TOAB12/TIAB12	-	92
P63			TOAB1B2/TRGAB1/ $\overline{\text{CS0}}$	68	-
	TOAB1B2/TRGAB1	-	93		
P64	TOAB1T3/TOAB13/TIAB13/ $\overline{\text{CS2}}$	69	-		
	TOAB1T3/TOAB13/TIAB13	-	94		
P65	TOAB1B3/EVTAB1/ $\overline{\text{CS3}}$	70	-		
	TOAB1B3/EVTAB1	-	95		
P70	I/O	端口 7 12 位 I/O 端口 可按位指定输入/输出。	ANI0	100	128
P71			ANI1	99	127
P72			ANI2	98	126
P73			ANI3	97	125
P74			ANI4	96	124
P75			ANI5	95	123
P76			ANI6	94	122
P77			ANI7	93	121
P78			ANI8	92	120
P79			ANI9	91	119
P710			ANI10	90	118
P711			ANI11	89	117
P90	I/O	端口 9 16 位 I/O 端口 可按位指定输入/输出。	KR6/TXDC1/SDA02	42	-
			KR6/TXDC1/SDA02/A0	-	54
P91			KR7/RXDC1/SCL02	43	-
			KR7/RXDC1/SCL02/A1	-	55
P92			TENC01/TIT01/TOT01	44	-
			TENC01/TIT01/TOT01/A2	-	56
P93			TECR0/TIT00/TOT00	45	-
			TECR0/TIT00/TOT00/A3	-	57
P94			TIAA31/TOAA31/TENC00/EVTT0	46	-
			TIAA31/TOAA31/TENC00/EVTT0/A4	-	58
P95			TIAA30/TOAA30	47	-
			TIAA30/TOAA30/A5	-	59
P96			TIAA21/TOAA21/INTP11	48	-
			TIAA21/TOAA21/INTP11/A6	-	62
P97			SIF1/TIAA20/TOAA20	49	-
	SIF1/TIAA20/TOAA20/A7	-	63		

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

引脚名称	I/O	功能	复用功能	引脚编号	
				JG3-U	JH3-U
P98	I/O	端口 9 16 位 I/O 端口 可按位指定输入/输出。	SOF1/INTP12	50	–
			SOF1/INTP12/A8	–	64
P99			SCKF1/INTP13	51	–
			SCKF1/INTP13/A9	–	65
P910			SIF3/TXDC2/INTP14	52	–
			SIF3/TXDC2/INTP14/A10	–	66
P911			SOF3/RXDC2/INTP15	53	–
			SOF3/RXDC2/INTP15/A11	–	67
P912			SCKF3	54	–
			SCKF3/A12	–	68
P913			TOAB1OFF/INTP16	55	–
			TOAB1OFF/INTP16/A13	–	69
P914			TIAA51/TOAA51/INTP17	56	–
			TIAA51/TOAA51/INTP17/A14	–	70
P915			TIAA50/TOAA50/INTP18	57	–
			TIAA50/TOAA50/INTP18/A15	–	71
PCM0	I/O	端口 CM 4 位 I/O 端口 (V850ES/JH3-U) 1 位 I/O 端口 (V850ES/JG3-U) 可按位指定输入/输出。	WAIT	–	89
PCM1			CLKOUT	64	86
PCM2			HLDK	–	10
PCM3			HLDRQ	–	11
PCS0	I/O	端口 CS 3 位 I/O 端口 可按位指定输入/输出。	CS0	–	96
PCS2			CS2	–	97
PCS3			CS3	–	116
PCT0	I/O	端口 CT 4 位 I/O 端口 (V850ES/JH3-U) 2 位 I/O 端口 (V850ES/JG3-U) 可按位指定输入/输出。	WR0	58	76
PCT1			WR1	59	77
PCT4			RD	–	87
PCT6			ASTB	–	88
PDH0	I/O	端口 DH 8 位 I/O 端口 可按位指定输入/输出。	A16	–	72
PDH1			A17	–	73
PDH2			A18	–	74
PDH3			A19	–	75
PDH4			A20	–	78
PDH5			A21	–	79
PDH6			A22	–	80
PDH7			A23	–	81

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

(4/4)

引脚名称	I/O	功能	复用功能	引脚编号	
				JG3-U	JH3-U
PDL0	I/O	端口 DL 16 位 I/O 端口 可按位指定输入/输出。	AD0	71	98
PDL1			AD1	72	99
PDL2			AD2	73	100
PDL3			AD3	74	101
PDL4			AD4	75	102
PDL5			AD5/FLMD1	76	103
PDL6			AD6	77	104
PDL7			AD7	78	105
PDL8			AD8	81	108
PDL9			AD9	82	109
PDL10			AD10	83	110
PDL11			AD11	84	111
PDL12			AD12	85	112
PDL13			AD13	86	113
PDL14			AD14	87	114
PDL15			AD15	88	115

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

(2) 非端口引脚

(1/9)

引脚名称	I/O	功能	复用功能	引脚编号	
				JG3-U	JH3-U
A0	输出	外部存储器地址总线 (当使用分离总线时)	P90/KR6/TXDC1/SDA02	-	54
A1			P91/KR7/RXDC1/SCL02	-	55
A2			P92/TENC01/TIT01/TOT01	-	56
A3			P93/TECR00/TIT00/TOT00	-	57
A4			P94/TIAA31/TOAA31/TENC0/EVTT0	-	58
A5			P95/TIAA30/TOAA30	-	59
A6			P96/TIAA21/TOAA21/INTP11	-	62
A7			P97/SIF1/TIAA20/TOAA20	-	63
A8			P98/SOF1/INTP12	-	64
A9			P99/ $\overline{\text{SCKF1}}$ /INTP13	-	65
A10			P910/SIF3/TXDC2/INTP14	-	66
A11			P911/SOF3/RXDC2/INTP15	-	67
A12			P912/ $\overline{\text{SCKF3}}$	-	68
A13			P913/TOAB1OFF/INTP16	-	69
A14			P914/TIAA51/TOAA51/INTP17	-	70
A15			P915/TIAA50/TOAA50/INTP18	-	71
A16			PDH0	-	72
A17			PDH1	-	73
A18			PDH2	-	74
A19			PDH3	-	75
A20			PDH4	-	78
A21			PDH5	-	79
A22			PDH6	-	80
A23	PDH7	-	81		
AD0	I/O	外部存储器的地址/ 数据总线	PDL0	71	98
AD1			PDL1	72	99
AD2			PDL2	73	100
AD3			PDL3	74	101
AD4			PDL4	75	102
AD5			PDL5/FLMD1	76	103
AD6			PDL6	77	104
AD7			PDL7	78	105
AD8			PDL8	81	108
AD9			PDL9	82	109
AD10			PDL10	83	110
AD11			PDL11	84	111
AD12			PDL12	85	112
AD13			PDL13	86	113
AD14			PDL14	87	114
AD15	PDL15	88	115		
ADTRG	输入	A/D 转换器外部触发输入	P03/INTP02/UCLK	7	7

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

(2/9)

引脚名称	I/O	功能	复用功能	引脚编号	
				JG3-U	JH3-U
ANI0	输入	A/D 转换器模拟电压输入	P70	100	128
ANI1			P71	99	127
ANI2			P72	98	126
ANI3			P73	97	125
ANI4			P74	96	124
ANI5			P75	95	123
ANI6			P76	94	122
ANI7			P77	93	121
ANI8			P78	92	120
ANI9			P79	91	119
ANI10			P710	90	118
ANI11			P711	89	117
ANO0	输出	D/A 转换器模拟电压输出	P10	3	3
ANO1			P11	4	4
ASCKC0	输入	UARTC0 波特率时钟输入	P32/SCKF4/TIAA00/TOAA00	27	39
ASTB	输出	外部存储器的地址选通信号	P62/TOAB1T2/TIAB12/TOAB12 PCT6	67 -	- 88
AV _{REF0}	-	A/D 转换器的参考电压， 用于端口 7 的正极电源供应	-	1	1
AV _{REF1}	-	用于 D/A 转换器的参考电压， 用于端口 1 的正极电源供应	-	5	5
AV _{SS}	-	用于 A/D 转换器和 D/A 转换器的地电位	-	2	2
CLKOUT	输出	内部系统时钟输出	PCM1	64	86
CS ₀	输出	片选输出	P63/TOAB1B2/TRGAB1 PCS0	68 -	- 96
CS ₂	输出	片选输出	P64/TOAB1T3/TIAB13/TOAB13 PCS2	69 -	- 97
CS ₃	输出	片选输出	P65/TOAB1B3/EVTAB1 PCS3	70 -	- 116
DCK	输入	片上调试时钟输入	P54/SOF2/KR4/RTP04 -	39 -	- 51
DDI	输入	片上调试数据输入	P52/TIAB03/KR2/TOAB03/RTP02 -	37 -	- 49
DDO	输出	片上调试数据输出 在片上调试模式下，强行设置为高电平输出	P53/SIF2/TIAB00/KR3/TOAB00/RTP03 -	38 -	- 50
DMS	输入	片上调试模式选择信号输入	P55/SCKF2/KR5/RTP05 -	40 -	- 52
DRST	输入	片上调试复位信号输入	P56/INTP05 -	41 -	- 53

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

(3/9)

引脚名称	I/O	功能	复用功能	引脚编号	
				JG3-U	JH3-U
EV _{DD}	-	用于外部设备的正极电源供应（与 V _{DD} 电位相同）	-	34, 63, 80	46, 61, 85, 107
EVT _{T0}	输入	TMT ₀ 的外部事件计数输入	P94/TIAA31/TOAA31/TENC00 P94/TIAA31/TOAA31/TENC00/A4	46 -	- 58
EVTAB ₁	输入	TAB ₁ 的外部事件计数输入	P65/TOAB1B3/ $\overline{\text{CS3}}$ P65/TOAB1B3	70 -	- 95
FLMD ₀	输入	Flash 存储器编程模式设置引脚	-	8	12
FLMD ₁	输入		PDL5/AD5	76	103
$\overline{\text{HLD}}\text{AK}$	输出	总线保持应答输出	PCM2	-	10
$\overline{\text{HLDR}}\text{Q}$	输入	总线保持请求输入	PCM3	-	11
INTP ₀₀	输入	外部中断请求输入 (可屏蔽, 模拟噪声消除) 为 INTP ₀₂ 引脚选择模拟噪声消除或数字噪声消除。	P00	-	8
INTP ₀₁			P01	-	9
INTP ₀₂			P03/ADTRG/UCLK	7	7
INTP ₀₃			P04	-	26
INTP ₀₄			P05	-	27
INTP ₀₅			P56/ $\overline{\text{DRST}}$ P24	41 -	- 36
INTP ₀₆			P25	-	28
INTP ₀₇			P30/TXDC0/SOF4	25	37
INTP ₀₈			P31/RXDC0/SIF4	26	38
INTP ₀₉			P34/TIAA10/TOAA10/TOAA1OFF/PPON	29	41
INTP ₁₀			P42/ $\overline{\text{SCKF0}}$	24	31
INTP ₁₁			P96/TIAA21/TOAA21/A6 P96/TIAA21/TOAA21	- 48	62 -
INTP ₁₂			P98/SOF1 P98/SOF1/A8	50 -	- 64
INTP ₁₃			P99/ $\overline{\text{SCKF1}}$ P99/SCKF1/A9	51 -	- 65
INTP ₁₄			P910/SIF3/TXDC2 P910/SIF3/TXDC2/A10	52 -	- 66
INTP ₁₅			P911/SOF3/RXDC2 P911/SOF3/RXDC2/A11	53 -	- 67
INTP ₁₆			P913/TOAB1OFF P913/TOAB1OFF/A13	55 -	- 69
INTP ₁₇			P914/TIAA51/TOAA51 P914/TIAA51/TOAA51/A14	56 -	- 70
INTP ₁₈			P915/TIAA50/TOAA50 P915/TIAA50/TOAA50/A15	57 -	- 71

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

(4/9)

引脚名称	I/O	功能	复用功能	引脚编号	
				JG3-U	JH3-U
KR0	输入	按键中断请求（片上模拟噪声消除器）	P50/TIAB01/TOAB01/RTP00 /UDMARQ1	35	47
KR1			P51/TIAB02/TOAB02/RTP01 /UDMARQ1	36	48
KR2			P52/TIAB03/TOAB03/RTP02/DDI	37	-
			P20/TIAB03/TOAB03/RTP02	-	32
KR3			P53/SIF2/TIAB00/TOAB00/RTP03/DDO	38	-
			P21/SIF2/TIAB00/TOAB00/RTP03	-	33
KR4			P54/SOF2/RTP04/DCK	39	-
			P22/SOF2/RTP04	-	34
KR5			P55/SCKF2/RTP05/DMS	40	-
	P23/SCKF2/RTP05	-	35		
KR6	P90/TXDC1/SDA02	42	-		
	P90/TXDC1/SDA02/A0	-	54		
KR7	P91/RXDC1/SCL02	43	-		
	P91/RXDC1/SCL02/A1	-	55		
NMI	输入	外部中断输入 （不可屏蔽，模拟噪声消除）	P02	6	6
\overline{OCI}	输入	过流检测输入	P35/TIAA11/TOAA11/RTC1HZ	30	42
PPON	输出	USB 电源输出	P34/TIAA10/TOAA10/TOAA10OFF/INTP09	29	41
\overline{RD}	输出	外部存储器“读”选通信号输出	P61/TOAB1B1/TIAB10/TOAB10	66	-
			PCT4	-	87
REGC	-	稳压器输出稳定电容的连接（4.7 μ F：推荐值）	-	10, 61	14, 83
\overline{RESET}	输入	系统复位输入	-	14	18
RTC1HZ	输出	实时计数器修正时钟(1 Hz) 输出	P35/TIAA11/TOAA11/ \overline{OCI}	30	42
RTCCL	输出	实时计数器时钟(32 kHz 主振荡器) 输出	P33/TIAA01/TOAA01/RTCDIV	28	40
RTCDIV	输出	实时计数器时钟(32 kHz 分频) 输出	P33/TIAA01/TOAA01/RTCCL	28	40
RTP00	输出	实时输出端口 N 沟道漏极开路输出可选	P50/TIAB01/KR0/TOAB01/UDMARQ1	35	47
RTP01			P51/TIAB02/KR1/TOAB02/UDMAAK1	36	48
RTP02			P52/TIAB03/KR2/TOAB03/DDI	37	-
			P20/TIAB03/KR2/TOAB03	-	32
RTP03			P53/SIF2/TIAB00/KR3/TOAB00/DDO	38	-
			P21/SIF2/TIAB00/KR3/TOAB00	-	33
RTP04			P54/SOF2/KR4/DCK	39	-
			P22/SOF2/KR4	-	34
RTP05	P55/SCKF2/KR5/DMS	40	-		
	P23/SCKF2/KR5	-	35		

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

引脚名称	I/O	功能	复用功能	引脚编号	
				JG3-U	JH3-U
RXDC0	输入	串行接收数据输入 (UARTC0 至 UARTC4)	P31/SIF4/INTP08	26	38
RXDC1			P91/KR7/SCL02	43	—
			P91/KR7/SCL02/A1	—	55
RXDC2			P911/SOF3/INTP15	53	—
			P911/SOF3/INTP15/A11	—	67
RXDC3			P37/SDA00/UDMAAK0	32	44
RXDC4		P41/SOF0/SCL01	23	30	
SCKF0	I/O	串行时钟 I/O (CSIF0 至 CSIF4) N 沟道漏极开路输出可选	P42/INTP10	24	31
SCKF1			P99/INTP13	51	—
			P99/INTP13/A9	—	65
SCKF2			P55/KR5/RTP05/DMS	40	—
			P23/KR5/RTP05	—	35
SCKF3			P912	54	—
	P912/A12	—	68		
SCKF4		P32/ASCKC0/TIAA00/TOAA00	27	39	
SCL00	I/O	串行时钟 I/O (I ² C00 至 I ² C02) N 沟道漏极开路输出可选	P36/TXDC3/UDMARQ0	31	43
SCL01			P41/SOF0/RXDC4	23	30
SCL02			P91/KR7/RXDC1	43	—
		P91/KR7/RXDC1/A1	—	55	
SDA00	I/O	串行发送/接收数据 I/O (I ² C00 至 I ² C02) N 沟道漏极开路输出可选	P37/RXDC3/UDMAAK0	32	44
SDA01			P40/SIF0/TXDC4	22	29
SDA02			P90/KR6/TXDC1	42	—
		P90/KR6/TXDC1/A0	—	54	
SIF0	输入	串行接收数据输入 (CSIF0 至 CSIF4)	P40/TXDC4/SDA01	22	29
SIF1			P97/TIAA20/TOAA20	49	—
			P97/TIAA20/TOAA20/A7	—	63
SIF2			P53/TIAB00/KR3/TOAB00/RTP03/DDO	38	—
			P21/TIAB00/KR3/TOAB00/RTP03	—	33
SIF3			P910/TXDC2/INTP14	52	—
	P910/TXDC2/INTP14/A10	—	66		
SIF4		P31/RXDC0/INTP08	26	38	
SOF0	输出	串行发送数据输出 (CSIF0 至 CSIF4) N 沟道漏极开路输出可选	P41/RXDC4/SCL01	23	30
SOF1			P98/INTP12	50	—
			P98/INTP12/A8	—	64
SOF2			P54/KR4/RTP04/DCK	39	—
			P22/KR4/RTP04	—	34
SOF3			P911/RXDC2/INTP15	53	—
	P911/RXDC2/INTP15/A11	—	67		
SOF4		P30/TXDC0/INTP07	25	37	

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

引脚名称	I/O	功能	复用功能	引脚编号	
				JG3-U	JH3-U
TECR0	输入	TMT0 编码器清除输入	P93/TIT00/TOT00	45	–
			P93/TIT00/TOT00/A3	–	57
TENC00		TMT0 编码器输入	P94/TIAA31/TOAA31/EVTT0	46	–
	P94/TIAA31/TOAA31/EVTT0/A4		–	58	
TENC01		P92/TIT01/TOT01	44	–	
		P92/TIT01/TOT01/A2	–	56	
TIAA00	输入	外部事件计数输入/捕获触发输入/外部触发输入 (TAA0)	P32/ASCKC0/SCKF4/TOAA00	27	39
TIAA01		捕获触发输入 (TAA0)	P33/TOAA01/RTCDIV/RTCCCL	28	40
TIAA10		外部事件计数输入/捕获触发输入/外部触发输入 (TAA1)	P34/TOAA10/TOAA10FF/INTP09/PPON	29	41
TIAA11		捕获触发输入 (TAA1)	P35/TOAA11/RTC1HZ/OCI	30	42
TIAA20		外部事件计数输入/捕获触发输入/外部触发输入 (TAA2)	P97/SIF1/TOAA20	49	–
			P97/SIF1/TOAA20/A7	–	63
TIAA21		捕获触发输入 (TAA2)	P96/TOAA21/INTP11	48	–
			P96/TOAA21/INTP11/A6	–	62
TIAA30		外部事件计数输入/捕获触发输入/外部触发输入 (TAA3)	P95/TOAA30	47	–
			P95/TOAA30/A5	–	59
TIAA31		捕获触发输入 (TAA3)	P94/TOAA31/TENC00/EVTT0	46	–
			P94/TOAA31/TENC00/EVTT0/A4	–	58
TIAA50		外部事件计数输入/捕获触发输入/外部触发输入 (TAA5)	P915/TOAA50/INTP18	57	–
		P915/TOAA50/INTP18/A15	–	71	
TIAA51	捕获触发输入 (TAA5)	P914/TOAA51/INTP17	56	–	
		P914/TOAA51/INTP17/A14	–	70	
TIAB00	输入	外部事件计数输入/捕获触发输入/外部触发输入 (TAB0)	P53/SIF2/KR3/TOAB00/RTP03/DDO	38	–
			P21/SIF2/KR3/TOAB00/RTP03	–	33
TIAB01		捕获触发输入 (TAB0)	P50/KR0/TOAB01/RTP00/UDMARQ1	35	47
TIAB02			P51/KR1/TOAB02/RTP01/UDMAAK1	36	48
TIAB03		P52/KR2/TOAB03/RTP02/DDI	37	–	
		P20/KR2/TOAB03/RTP02	–	32	
TIAB10	输入	捕获触发输入 (TAB1)	P61/TOAB1B1/TOAB10/RD	66	–
			P61/TOAB1B1/TOAB10	–	91
TIAB11			P60/TOAB1T1/TOAB11/WAIT	65	–
			P60/TOAB1T1/TOAB11	–	90
TIAB12			P62/TOAB1T2/TOAB12/ASTB	67	–
			P62/TOAB1T2/TOAB12	–	92
TIAB13			P64/TOAB1T3/TOAB13/CS2	69	–
			P64/TOAB1T3/TOAB13	–	94

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

引脚名称	I/O	功能	复用功能	引脚编号		
				JG3-U	JH3-U	
TIT00	输入	TMT0 外部触发输入/捕获触发输入	P93/TECR0/TOT00	45	–	
			P93/TECR0/TOT00/A3	–	57	
TIT01	输入	TMT0 捕获触发输入	P92/TENC01/TOT01	44	–	
			P92/TENC01/TOT01/A2	–	56	
TOAA00	输出	定时器输出 (TAA0) N 沟道漏极开路输出可选	P32/ASCKC0/SCKF4/TIAA00	27	39	
TOAA01			P33/TIAA01/RTCDIV/RTCCL	28	40	
TOAA10			定时器输出 (TAA1) N 沟道漏极开路输出可选	P34/TIAA10/TOAA1OFF/INTP09/PPON	29	41
TOAA11				P35/TIAA11/RTC1HZ/OCI	30	42
TOAA1OFF	输入	TAA1 高阻抗输出控制信号输入	P34/TIAA10/TOAA10/INTP09/PPON	29	41	
TOAA20	输出	定时器输出 (TAA2) N 沟道漏极开路输出可选	P97/SIF1/TIAA20	49	–	
			P97/SIF1/TIAA20/A7	–	63	
TOAA21			P96/TIAA21/INTP11	48	–	
			P96/TIAA21/INTP11/A6	–	62	
TOAA30		定时器输出 (TAA3) N 沟道漏极开路输出可选	P95/TIAA30	47	–	
			P95/TIAA30/A5	–	59	
TOAA31			P94/TIAA31/TENC00/EVTT0	46	–	
			P94/TIAA31/TENC00/EVTT0/A4	–	58	
TOAA50	输出	定时器输出 (TAA5) N 沟道漏极开路输出可选	P915/TIAA50/INTP18	57	–	
			P915/TIAA50/INTP18/A15	–	71	
TOAA51			P914/TIAA51/INTP17	56	–	
			P914/TIAA51/INTP17/A14	–	70	
TOAB00	输出	定时器输出 (TAB0) N 沟道漏极开路输出可选	P53/SIF2/TIAB00/KR3/RTP03/DDO	38	–	
			P21/SIF2/TIAB00/KR3/RTP03	–	33	
TOAB01			P50/TIAB01/KR0/RTP00/UDMARQ1	35	47	
TOAB02			P51/TIAB02/KR1/RTP01/UDMAAK1	36	48	
TOAB03			P52/TIAB03/KR2/RTP02/DDI	37	–	
			P20/TIAB03/KR2/RTP02	–	32	
TOAB1OFF	输入	TAB1 高阻抗输出控制信号输入	P913/INTP16	55	–	
			P913/INTP16/A13	–	69	
TOAB10	输出	定时器输出 (TAB1)	P61/TOAB1B1/TIAB10/RD	66	–	
			P61/TOAB1B1/TIAB10	–	91	
TOAB11			P60/TOAB1T1/TIAB11/WAIT	65	–	
			P60/TOAB1T1/TIAB11	–	90	
TOAB12			P62/TOAB1T2/TIAB12/ASTB	67	–	
			P62/TOAB1T2/TIAB12	–	92	
TOAB13			P64/TOAB1T3/TIAB13/CS2	69	–	
			P64/TOAB1T3/TIAB13	–	94	

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

引脚名称	I/O	功能	复用功能	引脚编号	
				JG3-U	JH3-U
TOAB1B1	输出	用于 TAB1 的 6 相 PWM 下臂的脉冲信号输出	P61/TIAB10/TOAB10/RD	66	-
			P61/TIAB10/TOAB10	-	91
TOAB1B2			P63/TRGAB1/CS0	68	-
			P63/TRGAB1	-	93
TOAB1B3			P65/EVTAB1/CS3	70	-
			P65/EVTAB1	-	95
TOAB1T1	输出	用于 TAB1 的 6 相 PWM 上臂的脉冲信号输出	P60/TOAB11/TIAB11/WAIT	65	-
			P60/TIAB11/TOAB11	-	90
TOAB1T2			P62/TIAB12/TOAB12/ASTB	67	-
			P62/TIAB12/TOAB12	-	92
TOAB1T3			P64/TOAB13/TIAB13/CS2	69	-
			P64/TIAB13/TOAB13	-	94
TOT00	输出	TMT0 定时器 输出	P93/TECR0/TIT00	45	-
			P93/TECR0/TIT00/A3	-	57
TOT01			P92/TENC01/TIT01	44	-
			P92/TENC01/TIT01/A2	-	56
TRGAB1	输入	TAB1 的外部触发输入	P63/TOAB1B2/CS0	68	-
			P63/TOAB1B2	-	93
TXDC0	输出	串行发送数据输出 (UARTC0 至 UARTC4) N 沟道漏极开路输出可选	P30/SOF4/INTP07	25	37
TXDC1			P90/KR6/SDA02	42	-
			P90/KR6/SDA02/A0	-	54
TXDC2			P910/SIF3/INTP14	52	-
			P910/SIF3/INTP14/A10	-	66
TXDC3			P36/SCL00/UDMARQ0	31	43
TXDC4	P40/SIF0/SDA01	22	29		
UCLK	输入	USB 时钟信号输入	P03/INTP02/ADTRG	7	7
UDMAAK0	输出	USB 的 DMA 应答	P37/RXDC3/SDA00	32	44
UDMAAK1		USB 的 DMA 应答	P51/TIAB02/KR1/TOAB02/RTP01	36	48
UDMARQ0	输入	USB 的 DMA 请求	P36/TXDC3/SCL00	31	43
UDMARQ1		USB 的 DMA 请求	P50/TIAB01/KR0/TOAB01/RTP00	35	47
UDMF	I/O	USB 功能数据 I/O (-)	-	17	21
UDMH		USB 主机数据 I/O (-)	-	20	24
UDPF		USB 功能数据 I/O (+)	-	18	22
UDPH		USB 主机数据 I/O (+)	-	21	25
UVDD	-	3.3 V USB 电源正极	-	19	23
VDD	-	内部单元电源正极	-	9, 60	13, 82
VSS	-	内部单元地电平	-	11, 33 62, 79	15, 45 60, 84 106

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

(9/9)

引脚名称	I/O	功能	复用功能	引脚编号	
				JG3-U	JH3-U
$\overline{\text{WAIT}}$	输入	外部等待输入	P60/TOAB1T1/TIAB11/TOAB11	65	–
			PCM0	–	89
$\overline{\text{WR0}}$	输出	外部存储器的写选通（低 8 位）	PCT0	58	76
$\overline{\text{WR1}}$		外部存储器的写选通（高 8 位）	PCT1	59	77
X1	输入	主时钟连接的振荡器	–	12	16
X2	–		–	13	17
XT1	输入	副时钟连接的振荡器	–	15	19
XT2	–		–	16	20

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

2.2 引脚状态

各种模式下的引脚工作状态描述如下：

表 2-2. 各种模式下的引脚工作状态

引脚名称	接通电源时 ^{注1}	复位期间 (接通电源除外)	HALT 模式 ^{注2}	IDLE1, IDLE2, Sub-IDLE 模式 ^{注2}	STOP 模式 ^{注2}	空闲状态 ^{注3}	总线保持				
$\overline{\text{DRST}}$	下拉	下拉 ^{注4}	保持	保持	保持	保持	保持				
P10/ANO0, P11/ANO1	不确定	高阻	保持	保持	高阻	保持	保持				
AD0 to AD15	高阻 ^{注5}	高阻 ^{注5}	注 s 6, 7	高阻	高阻	Held	高阻				
A0 to A15			不确定 ^{注6,8}								
A16 to A21			不确定 ^{注6}								
$\overline{\text{WAIT}}$			-								
CLKOUT			运行					L	L	运行	运行
$\overline{\text{WR0}}, \overline{\text{WR1}}$			H ^{注6}					H	H	H	高阻
$\overline{\text{RD}}$											
ASTB											
$\overline{\text{HLDAK}}$			运行 ^{注6}								L
$\overline{\text{HLDRQ}}$										-	-
Other port pins	高阻	高阻	保持	保持	保持	保持	保持				

- 注
1. 接通电源时，供电电压达到工作电压范围（下限）之后，持续等待 1 ms。
 2. 复用功能工作时的操作。
 3. 在分离总线模式下，在 T2 状态后，插入该引脚在空闲状态时的状态。在复用总线模式下，在 T3 状态后，插入该引脚在空闲状态时的状态。
 4. 在外部复位期间下拉。在看门狗定时器或时钟监视器等引发的内部复位期间，该引脚的状态根据 OCDM.OCDM0 位的设置有所不同。
 5. 总线控制引脚作为端口引脚，因此将其初始化为输入模式（端口模式）。
 6. DMA 操作期间，即使在 HALT 模式下仍然工作。
 7. 在分离总线模式下：高阻（Hi-Z）
在复用总线模式下：不确定
 8. 在分离总线模式下

备注

高阻：高阻抗
保持：保持上一个外部总线周期的状态。
L：低电平输出
H：高电平输出
-：没有采样的输入（不响应）

2.3 引脚I/O电路类型、I/O缓冲器电源、及未使用引脚的连接

表 2-3. 引脚 I/O 电路类型、I/O 缓冲器电源、及未使用引脚的连接(1/4)

引脚名称	复用功能	I/O 电路类型	推荐连接方式	JG3-U	JH3-U
P00	INTP00	10-D	输入： 通过电阻单独连接至 EV _{DD} 或 V _{SS} 输出： 保持开路	-	√
P01	INTP01			-	√
P02	NMI			√	√
P03	INTP02/ADTRG/UCLK			√	√
P04	INTP03			-	√
P05	INTP04			-	√
P10	ANO0	12-D	输入： 通过电阻单独连接至 AV _{REF1} 或 AV _{SS} 输出： 保持开路	√	√
P11	ANO1			√	√
P20	TIAB03/KR2/TOAB03/RTP02	10-D	输入： 通过电阻单独连接至 EV _{DD} 或 V _{SS} 输出： 保持开路	-	√
P21	SIF2/TIAB00/KR3/TOAB00/RTP03			-	√
P22	SOF2/KR4/RTP04			-	√
P23	SCKF2/KR5/RTP05			-	√
P24	INTP05			-	√
P25	INTP06			-	√
P30	TXDC0/SOF4/INTP07	10-D	输入： 通过电阻单独连接至 EV _{DD} 或 V _{SS} 输出： 保持开路	√	√
P31	RXDC0/SIF4/INTP08			√	√
P32	ASCKC0/SCKF4/TIAA00/TOAA00			√	√
P33	TIAA01/TOAA01/RTCDIV/RTCCCL			√	√
P34	TIAA10/TOAA10/TOAA1OFF/INTP09/PPON			√	√
P35	TIAA11/TOAA11/RTC1HZ/OCI			√	√
P36	TXDC3/SCL00/UDMARQ0			√	√
P37	RXDC3/SDA00/UDMAAK0			√	√
P40	SIF0/TXDC4/SDA01	10-D	输入： 通过电阻单独连接至 EV _{DD} 或 V _{SS} 输出： 保持开路	√	√
P41	SOF0/RXDC4/SCL01			√	√
P42	SCKF0/INTP10			√	√
P50	TIAB01/KR0/TOAB01/RTP00/UDMARQ1	10-D	输入： 通过电阻单独连接至 EV _{DD} 或 V _{SS} 输出： 保持开路	√	-
P51	TIAB02/KR1/TOAB02/RTP01/UDMAAK1			√	-
P52	TIAB03/KR2/TOAB03/RTP02/DDI			√	-
P53	SIF2/TIAB00/KR3/TOAB00/RTP03/DDO			√	-
P54	SOF2/KR4/RTP04/DCK			√	-
P55	SCKF2/KR5/RTP05/DMS			√	-
P56	INTP05/DRST	10-N	输入： 通过电阻单独连接至 V _{SS} 禁止固定为 V _{DD} 电平 输出： 保持开路 在 RESET 引脚复位后，内部下拉	√	-

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

表 2-3. 引脚 I/O 电路类型、I/O 缓冲器电源、及未使用引脚的连接(2/4)

引脚名称	复用功能	I/O 电路类型	推荐连接方式	JG3-U	JH3-U		
P60	TOAB1T1/TIAB11/TOAB11/ $\overline{\text{WAIT}}$	10-D	输入：通过电阻单独连接至 EV_{DD} 或 V_{SS} 输出：保持开路	√	–		
	TOAB1T1/TIAB11/TOAB11			–	√		
P61	TOAB1B1/TIAB10/TOAB10/ $\overline{\text{RD}}$			√	–		
	TOAB1B1/TIAB10/TOAB10			–	√		
P62	TOAB1T2/TIAB12/TOAB12/ $\overline{\text{ASTB}}$			√	–		
	TOAB1T2/TIAB12/TOAB12			–	√		
P63	TOAB1B2/TRGAB1/ $\overline{\text{CS0}}$			√	–		
	TOAB1B2/TRGAB1			–	√		
P64	TOAB1T3/TIAB13/TOAB13/ $\overline{\text{CS2}}$			√	–		
	TOAB1T3/TIAB13/TOAB13			–	√		
P65	TOAB1B3/EVTAB1/ $\overline{\text{CS3}}$			√	–		
	TOAB1B3/EVTAB1			–	√		
P70 至 P711	ANI0 至 ANI11			11-G	输入：通过电阻单独连接至 AV_{RE0} 或 AV_{SS} 输出：保持开路	√	√
P90	KR6/TXDC1/SDA02			10-D	输入：通过电阻单独连接至 EV_{DD} 或 V_{SS} 输出：保持开路	√	–
	KR6/TXDC1/SDA02/A0					–	√
P91	KR7/RXDC1/SCL02					√	–
	KR7/RXDC1/SCL02/A1	–	√				
P92	TENC01/TIT01/TOT01	√	–				
	TENC01/TIT01/TOT01/A2	–	√				
P93	TECR0/TIT00/TOT00	√	–				
	TECR0/TIT00/TOT00/A3	–	√				
P94	TIAA31/TOAA31/TENC00/EVTT0	√	–				
	TIAA31/TOAA31/TENC00/EVTT0/A4	–	√				
P95	TIAA30/TOAA30	√	–				
	TIAA30/TOAA30/A5	–	√				
P96	TIAA21/TOAA21/INTP11	√	–				
	TIAA21/TOAA21/INTP11/A6	–	√				
P97	SIF1/TIAA20/TOAA20	√	–				
	SIF1/TIAA20/TOAA20/A7	–	√				
P98	SOF1/INTP12	√	–				
	SOF1/INTP12/A8	–	√				
P99	$\overline{\text{SCKF1}}$ /INTP13	√	–				
	$\overline{\text{SCKF1}}$ /INTP13/A9	–	√				
P910	SIF3/TXDC2/INTP14	√	–				
	SIF3/TXDC2/INTP14/A10	–	√				
P911	SOF3/RXDC2/INTP15	√	–				
	SOF3/RXDC2/INTP15/A11	–	√				
P912	$\overline{\text{SCKF3}}$	√	–				
	$\overline{\text{SCKF3}}$ /A12	–	√				

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

表 2-3. 引脚 I/O 电路类型、I/O 缓冲器电源、及未使用引脚的连接(3/4)

引脚名称	复用功能	I/O 电路类型	推荐连接方式	JG3-U	JH3-U
P913	TOAB1OFF/INTP16	10-D	输入： 通过电阻单独连接至 EV _{DD} 或 V _{SS} 输出： 保持开路	√	-
	TOAB1OFF/INTP16/A13			-	√
P914	TIAA51/TOAA51/INTP17			√	-
	TIAA51/TOAA51/INTP17/A14			-	√
P915	TIAA50/TOAA50/INTP18			√	-
	TIAA50/TOAA50/INTP18/A15			-	√
PCM0	WAIT	5	输入： 通过电阻单独连接至 EV _{DD} 或 V _{SS} 输出： 保持开路	-	√
PCM1	CLKOUT			√	√
PCM2	HLD \overline{AK}			-	√
PCM3	HLDRQ			-	√
PCS0	$\overline{CS0}$	5	输入： 通过电阻单独连接至 EV _{DD} 或 V _{SS} 输出： 保持开路	-	√
PCS2	$\overline{CS2}$			-	√
PCS3	$\overline{CS3}$			-	√
PCT0	$\overline{WR0}$	5	输入： 通过电阻单独连接至 EV _{DD} 或 V _{SS} 输出： 保持开路	√	√
PCT1	$\overline{WR1}$			√	√
PCT4	\overline{RD}			-	√
PCT6	ASTB			-	√
PDH0 至 PDH7	A16 至 A23	5	输入： 通过电阻单独连接至 EV _{DD} 或 V _{SS} 输出： 保持开路	-	√
PDL0 至 PDL4	AD0 至 AD4	5	输入： 通过电阻单独连接至 EV _{DD} 或 V _{SS} 输出： 保持开路	√	√
PDL5	AD5/FLMD1			√	√
PDL6 至 PDL15	AD6 至 AD15			√	√
AV _{REF0}	-	-	直接连接至 V _{DD} 且始终供电	√	√
AV _{REF1}	-	-		√	√
AV _{SS}	-	-	直接连接至 V _{SS} 且始终供电	√	√
DCK	-	-	连接至 V _{SS}	-	√
DDI	-	-	连接至 V _{SS}	-	√
DDO	-	-	保持开路	-	√
DMS	-	-	连接至 V _{SS}	-	√
DRST	-	-	连接至 V _{SS}	-	√
EV _{DD}	-	-	直接连接至 V _{DD} 且始终供电	√	√
FLMD0	-	-	在 flash 模式之外的其它模式下，直接连接至 V _{SS}	√	√
REGC	-	-	连接至稳压器输出稳压电容	√	√
RESET	-	2	-	√	√
UDMF	-	-	保持开路	√	√
UDMH	-	-	保持开路	√	√
UDPF	-	-	保持开路	√	√
UDPH	-	-	保持开路	√	√

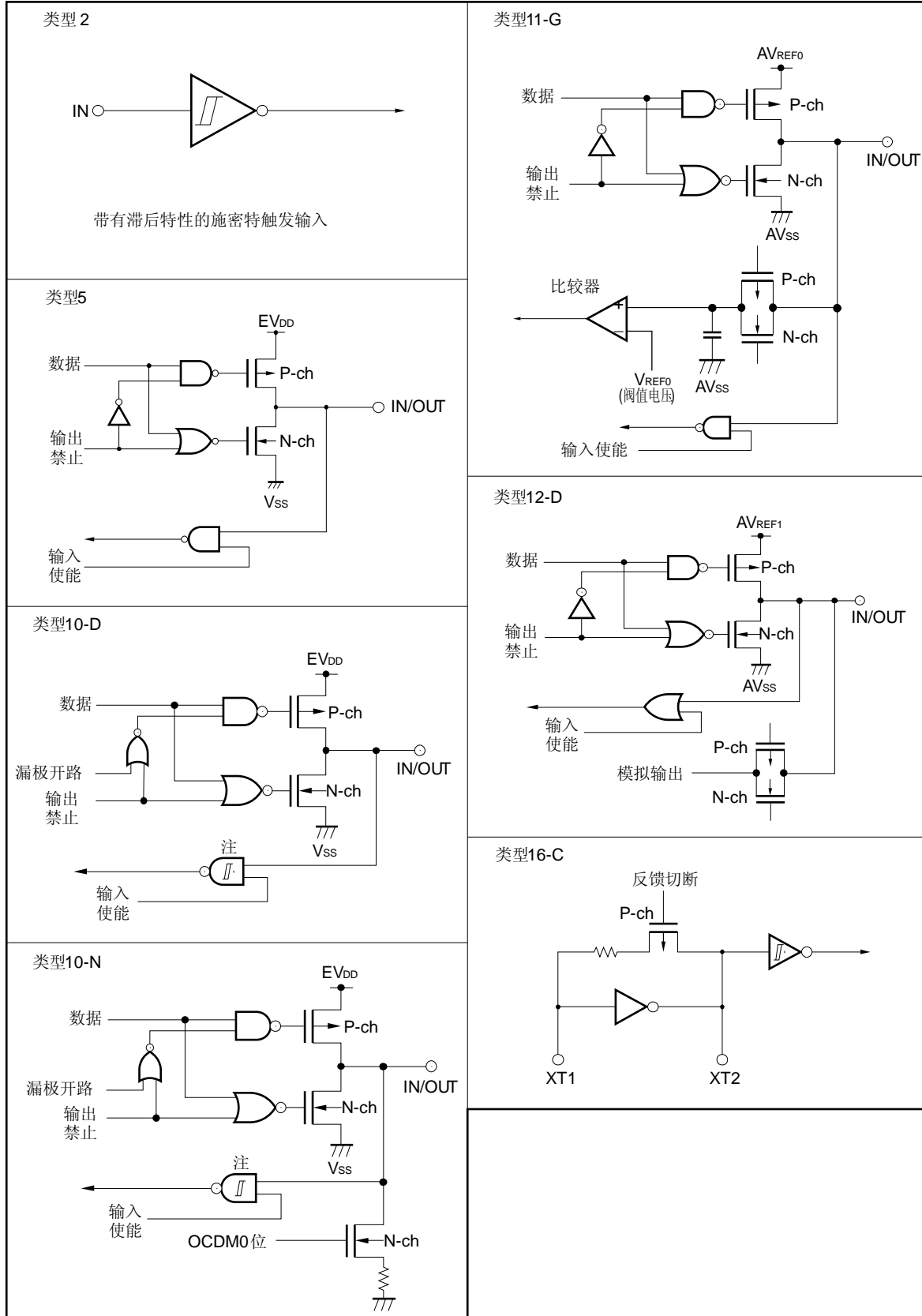
备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

表 2-3. 引脚 I/O 电路类型、I/O 缓冲器电源、及未使用引脚的连接(4/4)

引脚名称	复用功能	I/O 电路类型	推荐连接方式	JG3-U	JH3-U
U _{VDD}	-	-	直接连接至 V _{DD} 且始终供电	√	√
V _{DD}	-	-	-	√	√
V _{SS}	-	-	-	√	√
X1	-	-	-	√	√
X2	-	-	-	√	√
XT1	-	16-C	通过电阻连接至 V _{SS}	√	√
XT2	-	16-C	保持开路	√	√

备注 JG3-U: V850ES/JG3-U, JH3-U: V850ES/JH3-U

图 2-1 引脚 I/O 电路



注 端口模式下，没有迟滞特性。

2.4 注意事项

电源开启时，即便在复位期间，以下引脚也可能会暂时输出一个不确定电平。

- P10/ANO0 引脚
- P11/ANO1 引脚
- DDO 引脚（仅限 V850ES/JH3-U）
- P53/SIF2/TIAB00/KR3/TOAB00/RTP03/DDO 引脚（仅限 V850ES/JG3-U）

第三章 CPU功能

V850ES/JG3-U 和 V850ES/JH3-U 单片机的 CPU 是基于 RISC 架构的，且使用 5 级流水线，几乎所有指令的执行都可在一个时钟周期内完成。

3.1 特性

- 最短指令执行周期：
20.8 ns（运行于 48MHz 的主时钟（f_{XX}）：V_{DD}= 2.85 至 3.6 V）
30.5 μs（运行于 32.768 kHz 的副时钟（f_{XT}））
- 存储空间：
程序（物理地址）空间：64MB 线性
数据（逻辑地址）空间：4GB 线性
- 通用寄存器：32 位×32 个寄存器
- 内部 32 位架构
- 5 级流水线控制
- 乘法/除法指令
- 饱和运算指令
- 32 位移位指令：1 个时钟
- 长/短两种格式的数据装载/存贮指令
- 4 种类型位操作指令：
 - SET1
 - CLR1
 - NOT1
 - TST1

3.2 CPU寄存器组

V850ES/JG3-U 和 V850ES/JH3-U 的寄存器可分为两种类型：通用程序寄存器和专用系统寄存器。所有寄存器都是 32 位宽度。

详细信息，请参见 **V850ES 结构用户手册**。

(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	(汇编程序保留寄存器)	EIPSW	(中断状态保存寄存器)
r2			
r3	(堆栈指针(SP))	FEPC	(NMI 状态保存寄存器)
r4	(全局指针(GP))	FEPSW	(NMI状态保存寄存器)
r5	(文本指针(TP))		
r6		ECR	(中断源寄存器)
r7			
r8		PSW	(程序状态字)
r9			
r10		CTPC	(CALLT 执行状态保存寄存器)
r11		CTPSW	(CALLT 执行状态保存寄存器)
r12			
r13			
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	(连接指针(LP))		
31	0		
PC	(程序计数器)		

3.2.1 程序寄存器组

程序寄存器组包括若干通用寄存器和一个程序计数器。

(1) 通用寄存器 (r0 至 r31)

共有 32 个通用寄存器，r0 至 r31。这些寄存器中的任何一个都可用于存储数据变量或地址变量。

然而，执行某些指令会隐式使用 r0 和 r30 寄存器，当使用这些寄存器时，一定要加以注意。寄存器 r0 的内容总是保持为 0，用于使用数据 0 的操作或 0 偏移量的寻址。当 SLD 和 SST 指令作为基址指针访问存储器时，使用 r30 寄存器。汇编编译器和 C 编译器也隐式使用 r1、r3 至 r5、和 r31。使用这些寄存器之前，必须保存寄存器中的内容以防止丢失，并且在使用这些寄存器之后，将保存的内容恢复到寄存器中。r2 有时会被实时操作系统使用。如果实时操作系统没有使用 r2，那么它可以将用作变量寄存器。

表 3-1. 程序寄存器

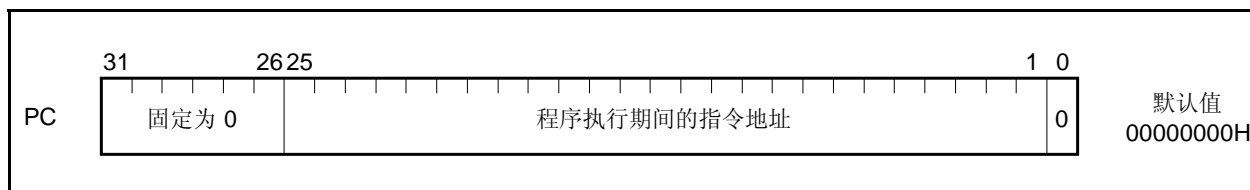
名称	用途	操作
r0	零寄存器	始终保持为零。
r1	汇编程序保留寄存器	用作生成 32 位立即数的工作寄存器
r2	用于地址变量/数据的变量寄存器（如果实时操作系统没有使用寄存器 r2）	
r3	堆栈指针	用于在函数调用时生成堆栈帧
r4	全局指针	用于访问数据区中的全局变量
r5	文本指针	用作指示文本区域起始地址的寄存器（该区域为程序代码存放区域）
r6 至 r29	用于地址/数据变量的寄存器	
r30	元指针	用作访问内存的基址指针
r31	链接指针	编译器调用函数时使用
PC	程序计数器	在程序执行期间，保存指令地址

备注 有关汇编器和 C 编译器中使用 r1、r3 至 r5、和 r31 的更多细节，请参见 CA850 (C 编译程序包) 汇编语言用户手册。

(2) 程序计数器 (PC)

程序计数器用来保存程序执行期间的指令地址。该寄存器的低 32 位有效。其中第 31 位至第 26 位固定为 0，即使第 25 位向第 26 位有进位，也将忽略不计。

第 0 位固定为 0，这意味着不能执行跳转到奇地址处的操作。



3.2.2 系统寄存器组

系统寄存器控制 CPU 的状态，并保存中断信息。

通过使用系统寄存器载入/存储指令（LDSR，STSR 指令）和下表所列的系统寄存器编号，来实现对系统寄存器的读取或写入操作。

表 3-2. 系统寄存器号

系统寄存器编号	系统寄存器名称	操作数说明	
		LDSR Instruction	STSR Instruction
0	中断状态保存寄存器 (EIPC) ^{注1}	√	√
1	中断状态保存寄存器 (EIPSW) ^{注1}	√	√
2	非屏蔽中断状态保存寄存器 (FEPC) ^{注1}	√	√
3	非屏蔽中断状态保存寄存器 (FEPSW) ^{注1}	√	√
4	中断源寄存器 (ECR)	×	√
5	程序状态字 (PSW)	√	√
6 至 15	将来功能扩展预留寄存器 (如果对这些寄存器进行访问，不能保证操作的有效性)	×	×
16	CALLT 指令执行状态保存寄存器 (CTPC)	√	√
17	CALLT 指令执行状态保存寄存器 (CTPSW)	√	√
18	异常/调试陷阱状态保存寄存器 (DBPC)	√ ^{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 指令基址指针 (CTBP)	√	√
21 至 31	将来功能扩展预留寄存器 (如果对这些寄存器进行访问，不能保证操作的有效性)	×	×

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

注意事项 即使 EIPC 或 FEPC、或 CTPC 的第 0 位由 LDSR 指令设置为 1，中断服务之后，由 RETI 指令返回主程序的时候会忽略第 0 位（因为 PC 的第 0 位固定为 0）。设置偶数值至 EIPC、FEPC 和 CTPC（第 0 位 = 0）。

备注 √: 可以访问
×: 禁止访问

(2) NMI 状态保存寄存器 (FEPC 和 FEPSW)

FEPC 和 FEPSW 用于保存产生不可屏蔽中断 (NMI) 时的状态。

如果有 NMI 产生, 那么程序计数器 (PC) 中的内容就保存到 FEPC 寄存器中, 程序状态字 (PSW) 中的内容保存到 FEPSW 寄存器中。

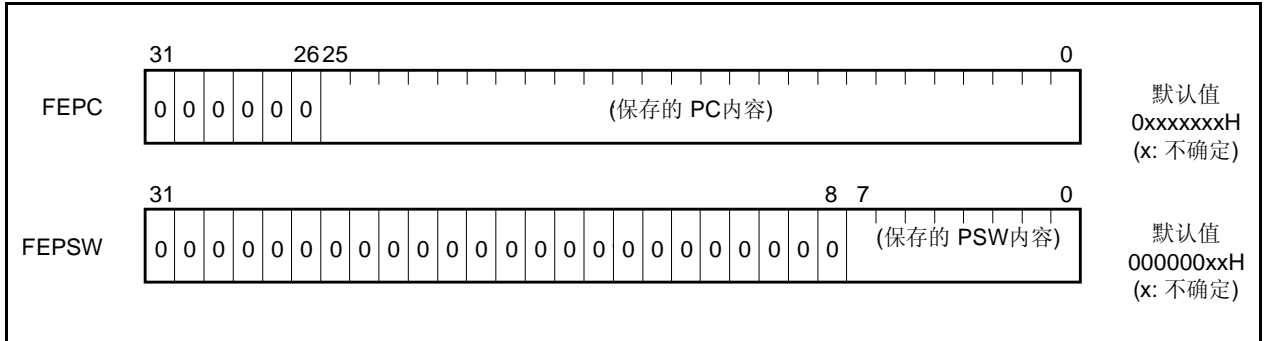
当有 NMI 产生时, 就将当前执行指令的下一条指令地址保存到寄存器 FEPC 中, 某些指令除外。

将 PSW 当前内容保存到 FEPSW 寄存器中。

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

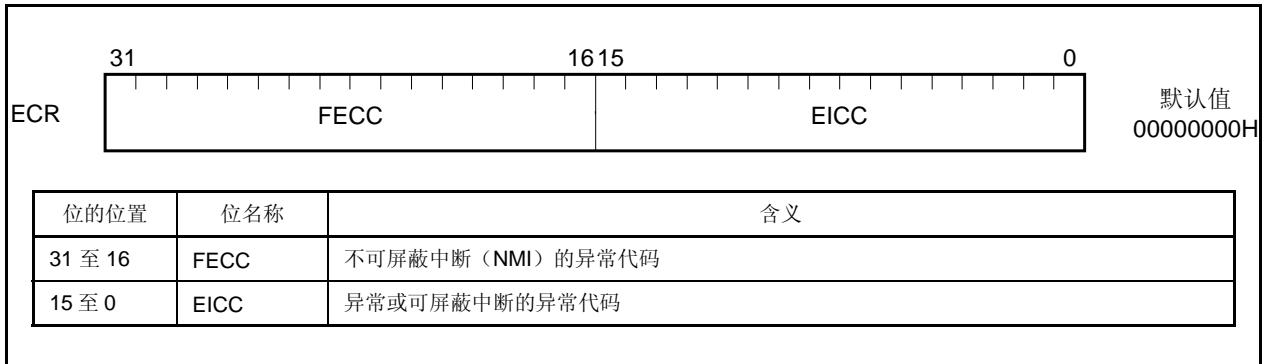
FEPC 的第 31 位至第 26 位以及 FEPSW 的第 31 位至第 8 位作为保留位用于将来功能扩展 (这些位始终固定为 0)。

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



(3) 中断源寄存器 (ECR)

如果有中断或异常产生, 那么, 异常或中断源信息就保存在该中断源寄存器 (ECR) 中。该寄存器保存各中断源的异常代码。由于该寄存器是只读寄存器, 所以, 不能通过 LDSR 指令将数据写入该寄存器。



(4) 程序状态字 (PSW)

程序状态字 (PSW) 是一个标志位的集合, 指示程序状态 (指令执行结果) 和 CPU 状态。

如果使用 LDSR 更改了该寄存器某位的内容, 那么, 在 LDSR 指令完成后, 新内容就会立即生效。然而, 如果将 ID 标志设为 1, 在 LDSR 指令执行期间不会响应中断请求。

该寄存器的第 31 位至第 8 位作为保留位用于将来功能扩展 (这些位固定为 0)。

(1/2)



位的位置	标志名称	含义
31 至 8	RFU	保留区域。固定为“0”。
7	NP	表示正在进行不可屏蔽中断 (NMI) 服务。当响应 NMI 中断请求时, 该标志位置 1, 同时禁止多重中断。 0: NMI 服务没有进行。 1: NMI 服务正在进行。
6	EP	表示正在进行异常处理。异常产生后该标志位置为 1。即使设置该位, 也可以响应中断请求。 0: 没有处理异常。 1: 正在处理异常。
5	ID	标识是否允许响应可屏蔽中断请求。 0: 允许中断 1: 禁止中断
4	SAT ^注	表示执行饱和运算的结果溢出且饱和。由于该位为一累积标志位, 所以, 当饱和运算指令结果饱和时, 该位置 1, 即使是下一个运算结果没有饱和, 该位也不会清 0。使用 LDSR 指令将该位清零。执行算术运算指令时, 该位既不置 1, 也不清 0。 0: 没有饱和 1: 饱和
3	CY	表示运算结果是否有进位或借位产生。 0: 无进位或借位产生 1: 产生进位或借位
2	OV ^注	表示在运算过程中有无溢出。 0: 没有发生溢出。 1: 发生溢出
1	S ^注	标示运算结果是否为负数 0: 运算结果为正数或零。 1: 结果为负数。
0	Z	显示运算结果是否为零。 0: 结果不为 0。 1: 结果为 0。

备注 请阅读下页的“注”。

注 已经执行饱和处理的运算结果由 OV 标志和 S 标志的内容来决定。在执行饱和运算时，只有当 OV 标志位置 1 时，SAT 标志位才置为 1。

运算结果的状态	标志位状态			饱和处理的操作结果
	SAT	OV	S	
超过正值上界	1	1	0	7FFFFFFFH
越过负值下界	1	1	1	80000000H
正值（未超上界）	运算前保持该值	0	0	操作结果本身
负值（未越下界）			1	

(5) CALLT 指令执行状态保存寄存器 (CTPC 和 CTPSW)

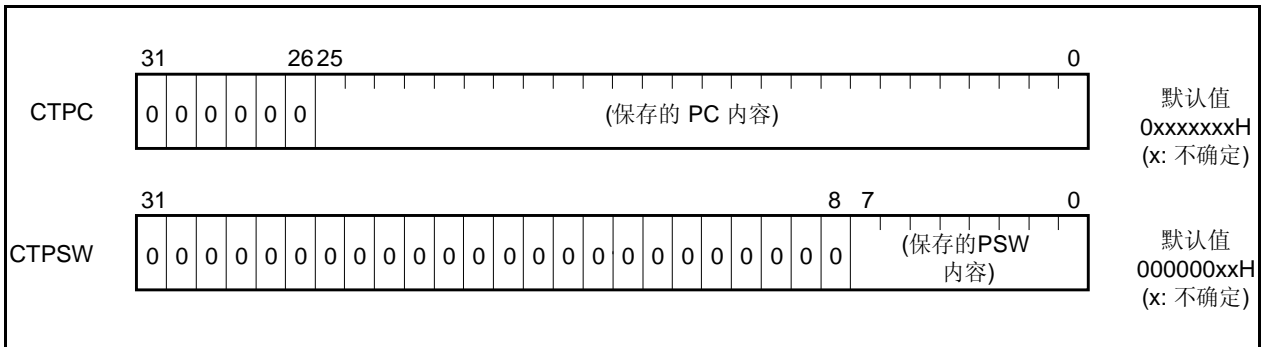
CTPC 和 CTPSW 是 CALLT 指令的执行状态保存寄存器。

当执行 CALLT 指令时，程序计数器 (PC) 的内容就保存到 CTPC 寄存器中，程序状态字 (PSW) 的内容保存到 CTPSW 寄存器中。

寄存器 CTPC 中保存的内容是 CALLT 指令之后下一条指令的地址。

将 PSW 当前内容保存到 CTPSW 寄存器中。

CTPC 的第 31 位至第 26 位以及 CTPSW 的第 31 位至第 8 位作为保留位，用于将来功能扩展（这些位固定为 0）。



(6) 异常/调试陷阱状态保存寄存器 (DBPC 和 DBPSW)

DBPC 和 DBPSW 是异常/调试陷阱状态寄存器。

如果产生异常陷阱或调试陷阱，那么程序计数器 (PC) 的内容就保存到 DBPC 寄存器中，程序状态字 (PSW) 的内容保存到 DBPSW 寄存器中。

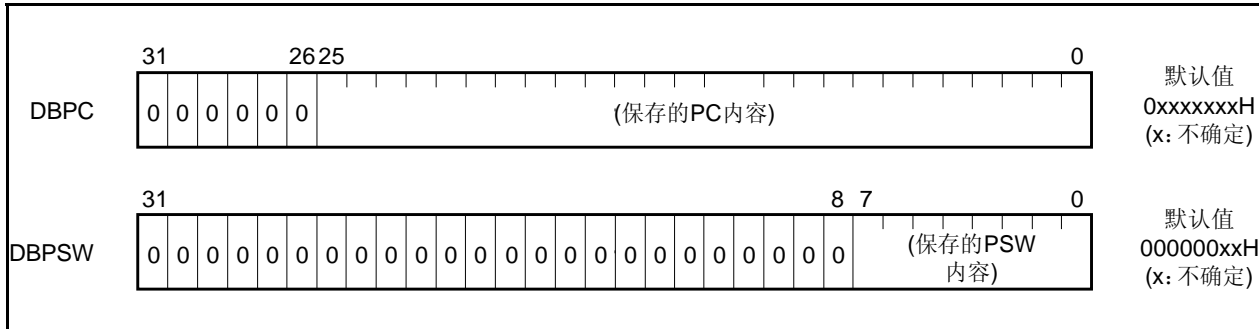
在异常陷阱或者调试陷阱产生时，保存到 DBPC 寄存器中的内容是当前执行指令的下一条指令的地址。

将 PSW 的当前内容保存到 DBPSW 寄存器中。

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

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

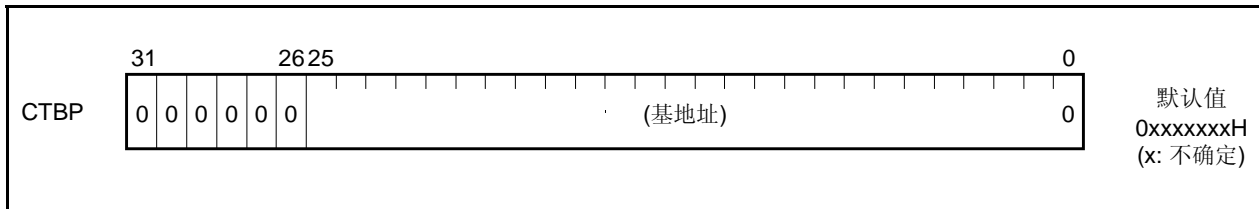
通过执行 DBRET 指令，将寄存器 DBPC 的值恢复到 PC 中，并将寄存器 DBPSW 的值恢复到 PSW 中。



(7) CALLT 指令基址指针 (CTBP)

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

该寄存器的第 31 位至第 26 位作为保留位，用于将来功能扩展 (这些位固定为 0)。



3.3 操作模式

V850ES/JG3-U 和 V850ES/JH3-U 具有以下几种操作模式。

(1) 正常工作模式

该模式下，系统从复位状态中恢复后，与总线接口相关的各个引脚均设置为端口模式。程序跳转到内部 ROM 中的复位入口地址处，然后开始执行指令。

(2) Flash 存储器编程模式

该模式下，使用 flash 编程器可以对内部 flash 进行编程。

(3) 片上调试模式

V850ES/JG3-U 和 V850ES/JH3-U 提供使用 JTAG（Joint Test Action Group）通讯规范的片上调试功能。详情参见第 32 章 片上调试功能。

3.3.1 指定操作模式

利用 FLMD0 引脚和 FLMD1 引脚来指定操作模式。

在正常操作模式下，复位解除后，确保 FLMD0 引脚输入低电平。

连接 flash 编程器后，在 flash 存储器编程模式下，flash 编程器会有高电平输入 FLMD0 引脚。然而，在自编程模式下，必须由外部电路输入高电平。

复位解除后的操作		复位后的操作模式
FLMD0	FLMD1	
L	×	正常工作模式
H	L	Flash 存储器编程模式
H	H	设置禁止

备注

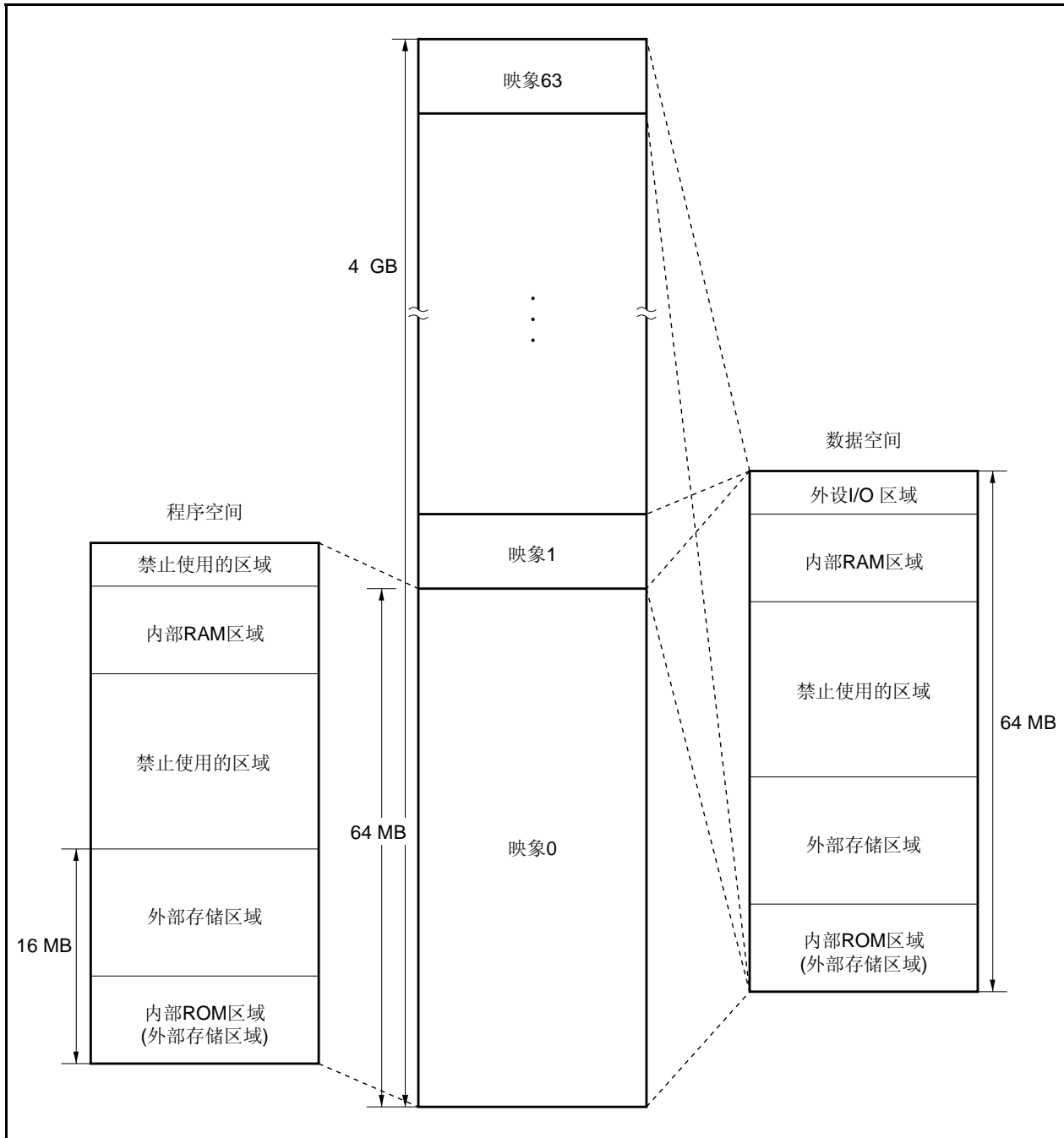
- L: 低电平输入
- H: 高电平输入
- ×: 无需理会

3.4 地址空间

3.4.1 CPU地址空间

指令寻址可达 64MB，支持线性地址空间（程序空间），包括多达 16 MB 的外部储存区域和内部 ROM 区域，以及内部 RAM 区域。操作数寻址（数据访问）最大支持 4 GB 的线性地址空间（数据空间）。但是，4 GB 的地址空间可以看作是 64 个 64 MB 的物理地址空间映像。这意味着不管第 31 位至第 26 位的取值如何，都是对相同的 64 MB 物理地址空间进行访问。

图 3-1. 地址空间映射



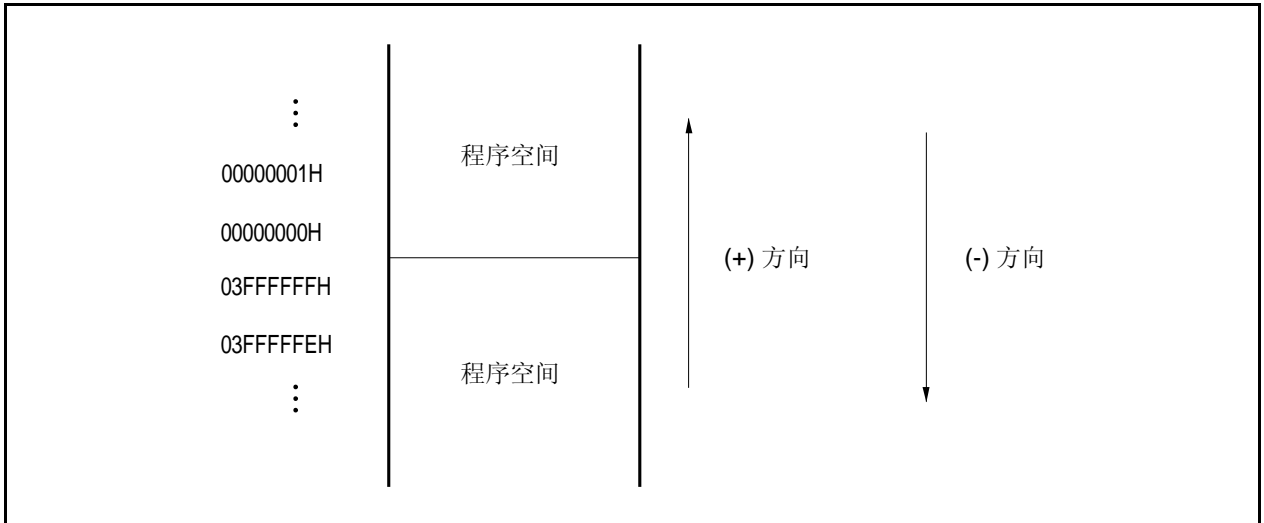
3.4.2 CPU 地址空间的绕回

(1) 程序空间

对于 32 位的 PC（程序计数器），其高 6 位固定为 0，且只有低 26 位是有效的。在计算跳转地址时，高 6 位会忽略第 25 位至第 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 是相邻的连续地址。像这种程序空间的最高地址和最低地址为相邻的连续地址的情况，称为地址的绕回。

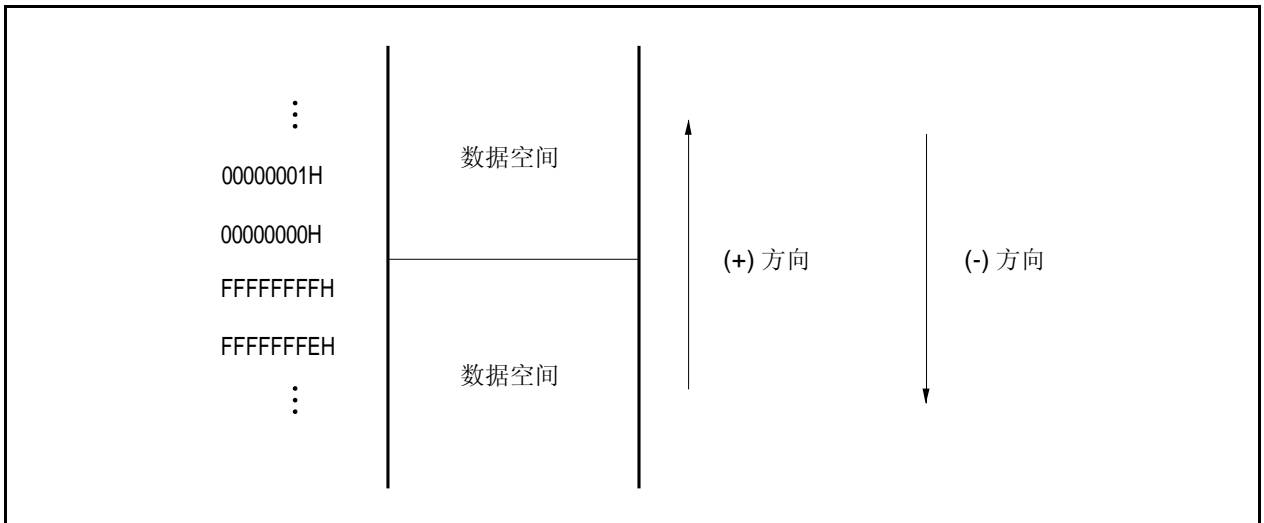
注意事项 由于地址 03FFF000H 至 03FFFFFFH 的 4 KB 区域为片上外设 I/O 区域，所以不能从该地址区域取指令。因此，如果跳转地址的计算结果会影响该区域，则不要执行此操作。



(2) 数据空间

操作数地址计算的结果超出 32 位时，该结果地址将被忽略。

因此，数据空间的上限地址 FFFFFFFFH 和下限地址 00000000H 是相邻的连续地址，在这些地址的边界会发生绕回。



3.4.3 存储器映射

在 V850ES/JG3-U 和 V850ES/JH3-U 内保留了下列区域。

图 3-2. 数据存储空间映射图（物理地址）

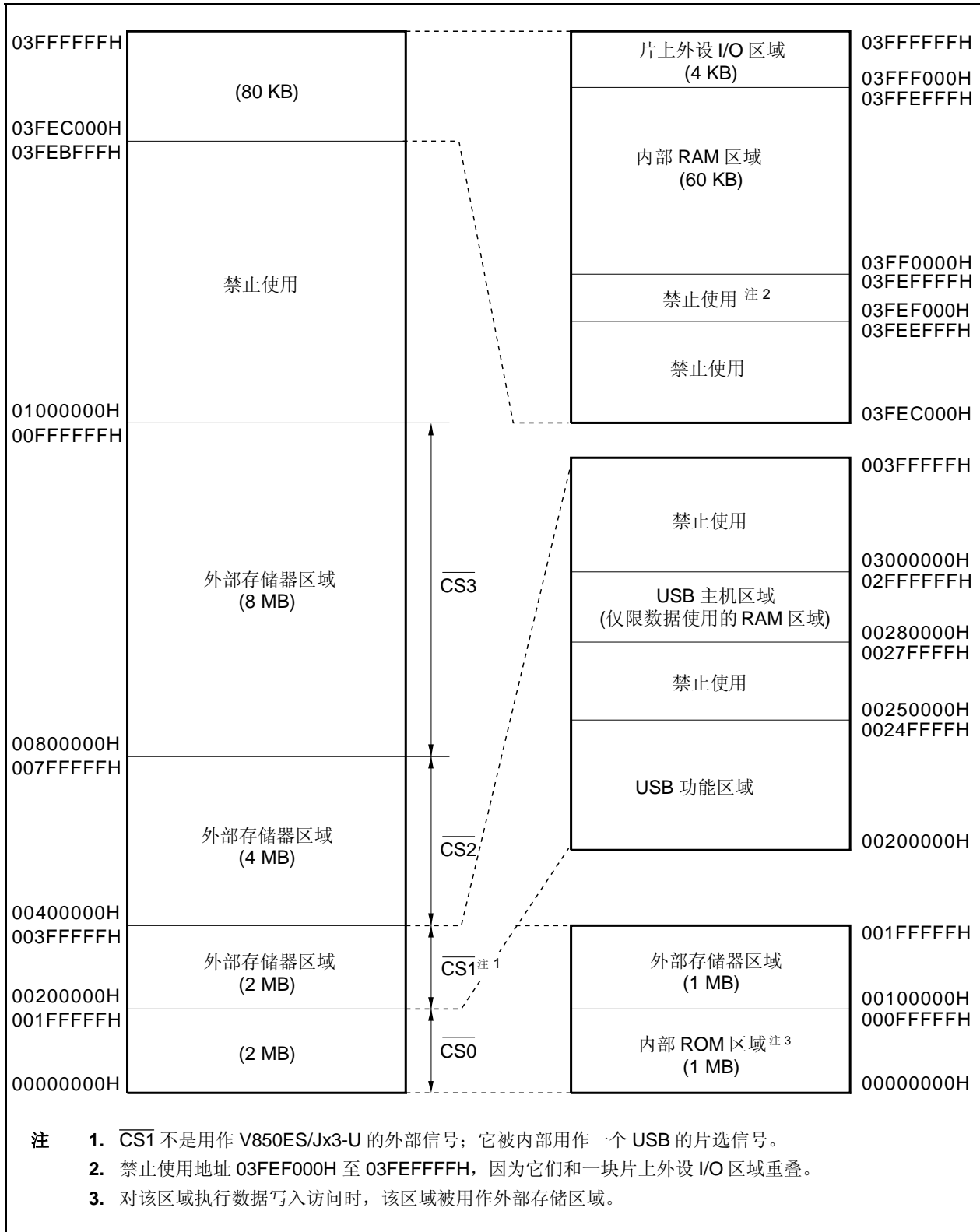
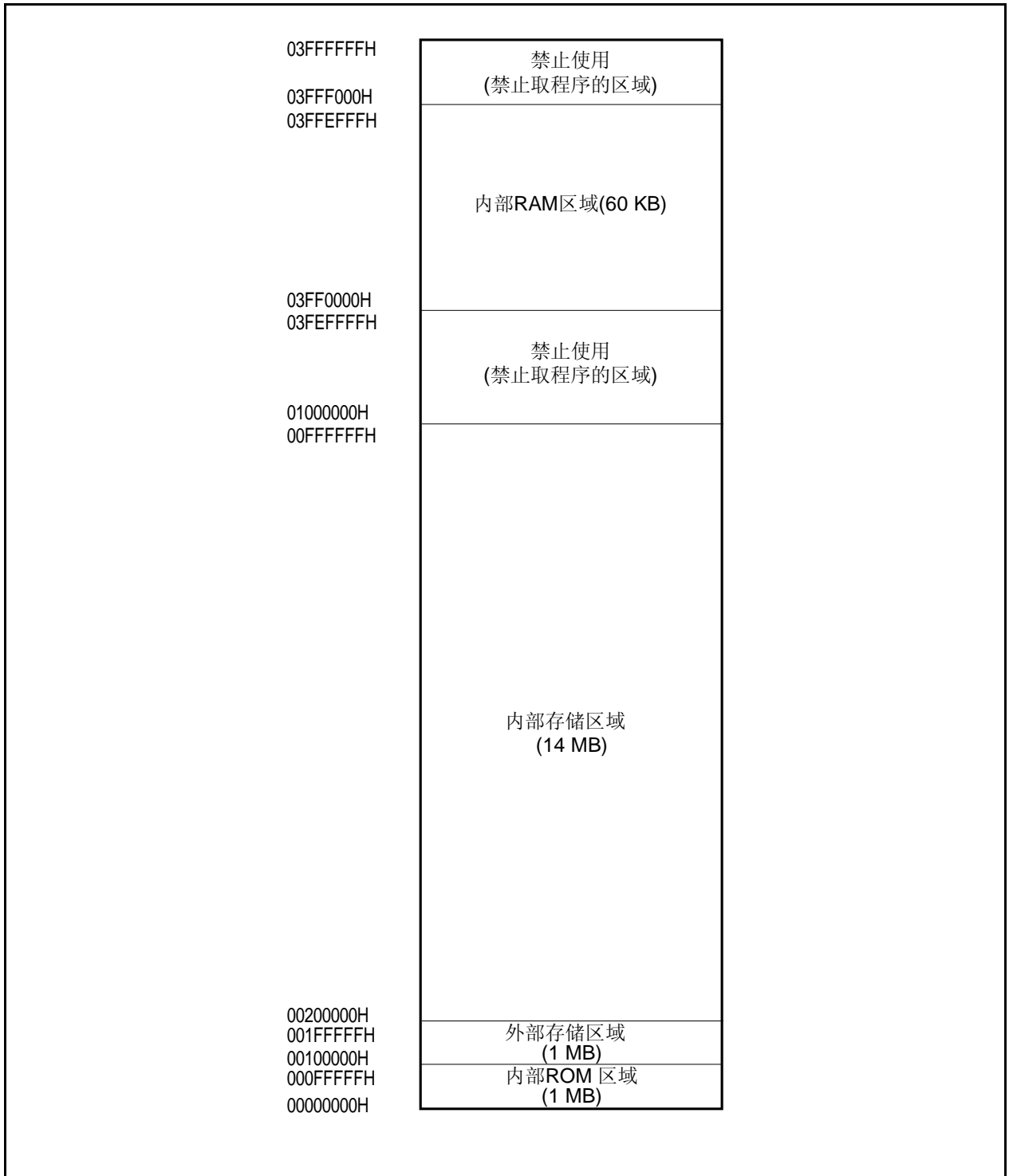


图 3-3. 程序存储器映射图



3.4.4 存储区域

(1) 内部 ROM 区域

保留 1 MB 的区域作为内部 ROM 区域。

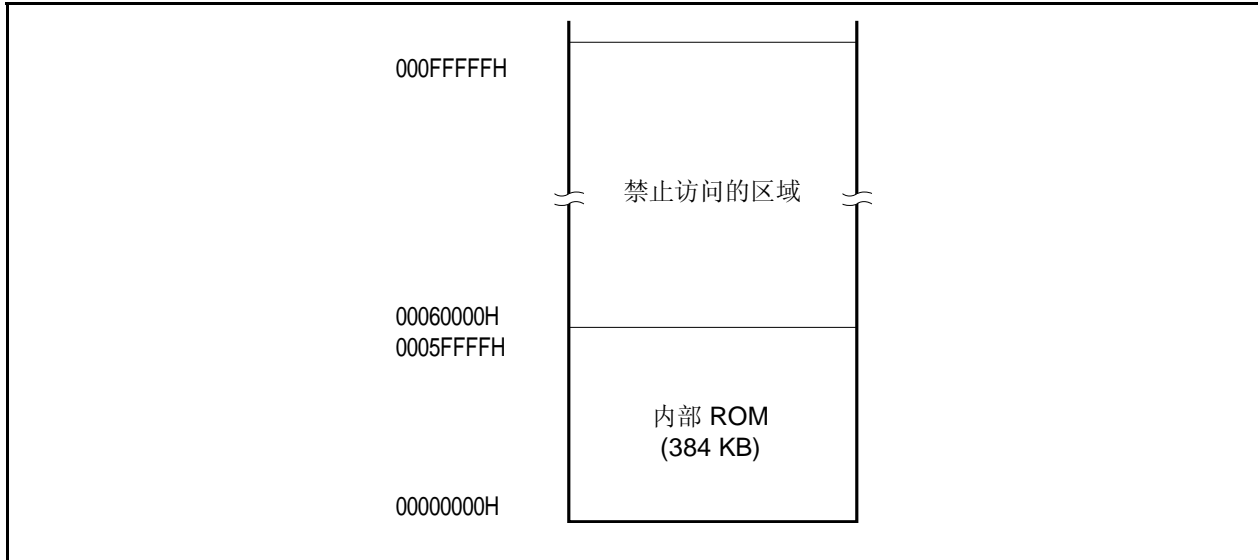
(a) 内部 ROM (384KB)

在下列产品中，384 KB 被分配在地址 00000000H 至 0005FFFFH。

禁止访问地址 00060000H 至 000FFFFFFH。

- μ PD70F3763, 70F3768

图 3-4. 内部 ROM 区域 (384 KB)



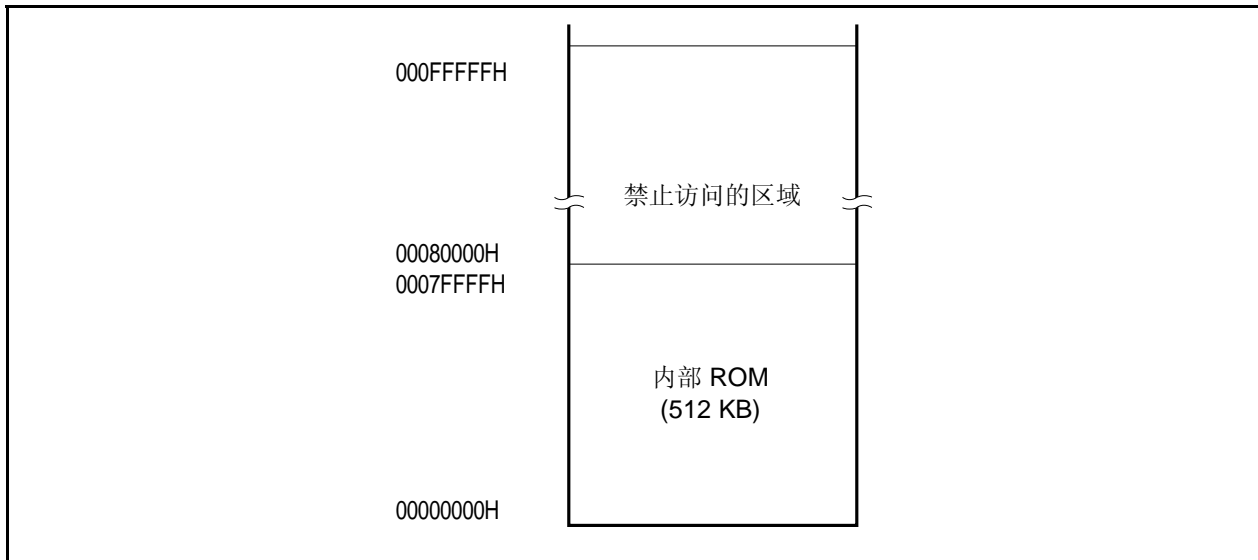
(b) 内部 ROM (512 KB)

在下列产品中，512 KB 分配在地址 00000000H 至 0007FFFFH。

禁止访问地址 00080000H 至 000FFFFFFH。

- μ PD70F3764, 70F3769

图 3-5. 内部 ROM 区域 (512 KB)



(2) 内部 RAM 区域

有 60 KB 的区域可以保留作为内部 RAM 区域。

V850ES/JG3-U 和 V850ES/JH3-U 在内部 RAM 之外，还包括一个 8KB 仅限数据使用的 RAM。如果不使用 USB 主机或 USB 功能，这个仅限数据使用的 RAM 可以被用作内部 RAM。

V850ES/JG3-U 和 V850ES/JH3-U 的 RAM 容量如下所示。

表 3-3. RAM 区域

类属名	产品名称	内部RAM	仅限数据使用的RAM	总RAM
V850ES/JG3-U	μPD70F3763	40 KB	8 KB	48 KB
	μPD70F3764	48 KB		56 KB
V850ES/JH3-U	μPD70F3768	40 KB		48 KB
	μPD70F3769	48 KB		56KB

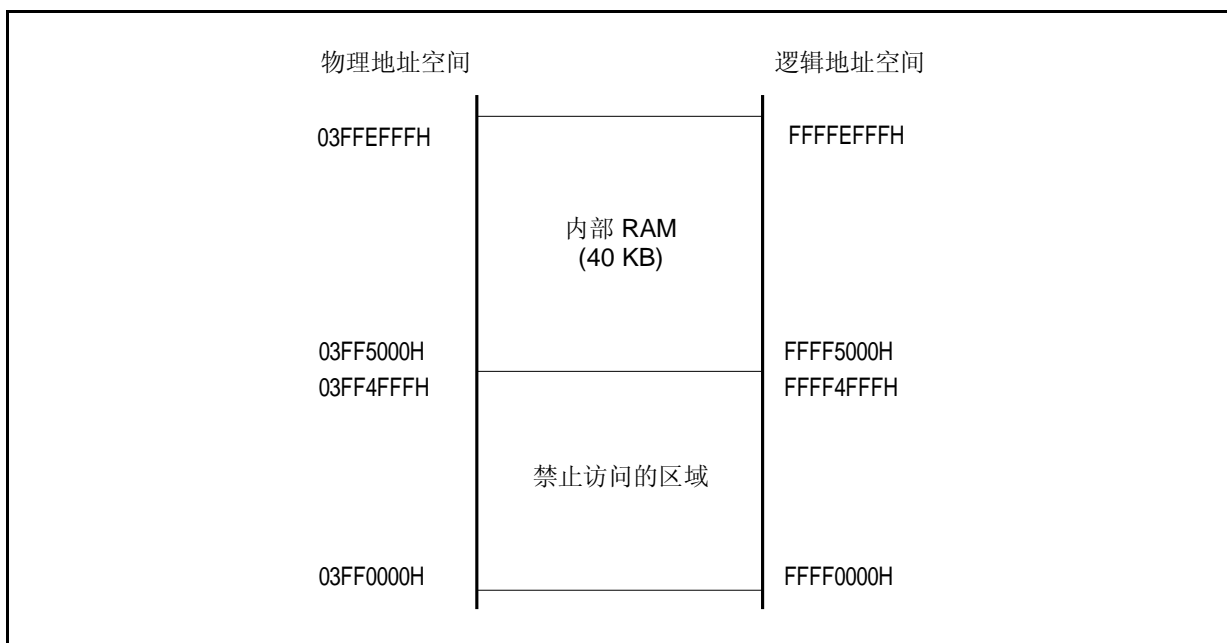
(a) 内部 RAM (40 KB)

在下列产品中，40 KB 分配在地址 03FF5000H 至 03FF4FFFH。

禁止访问地址 03FF0000H 至 03FF4FFFH。

- μPD70F3763, 70F3768

图 3-6. 内部 RAM 区域 (40 KB)

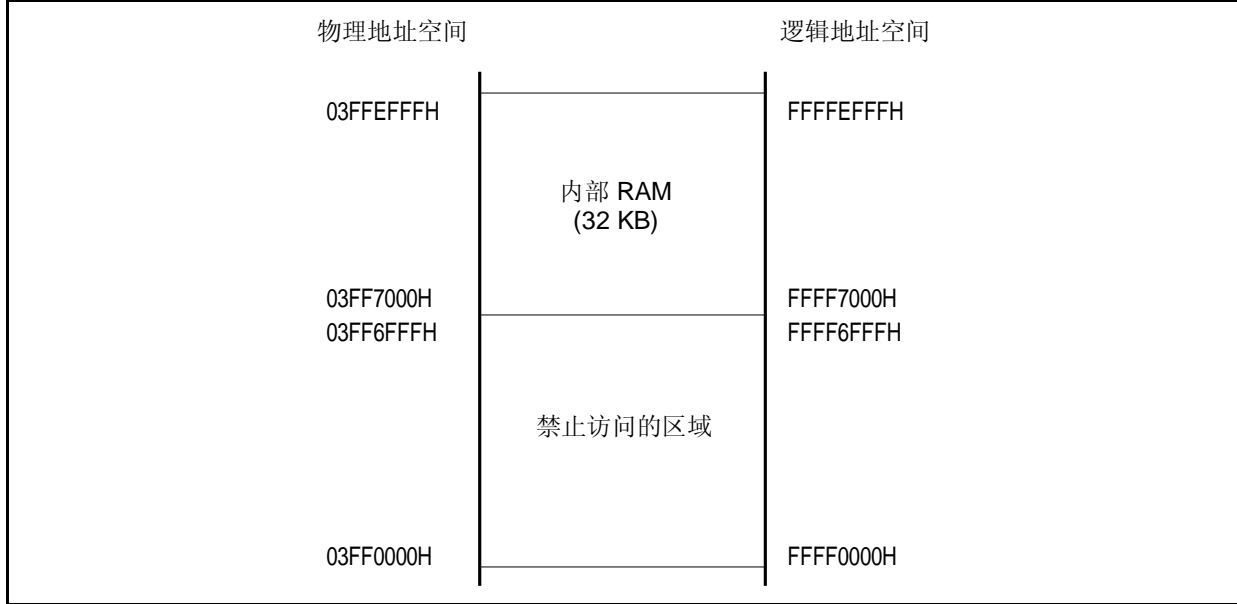


(b) 内部 RAM (48 KB)

在下列产品中，48 KB 分配在地址 03FF3000H 至 03FFFEFFFH。
禁止访问地址 03FF0000H 至 03FF2FFFFH。

- μ PD70F3764, 70F3769

图 3-7. 内部 RAM 区域 (48 KB)

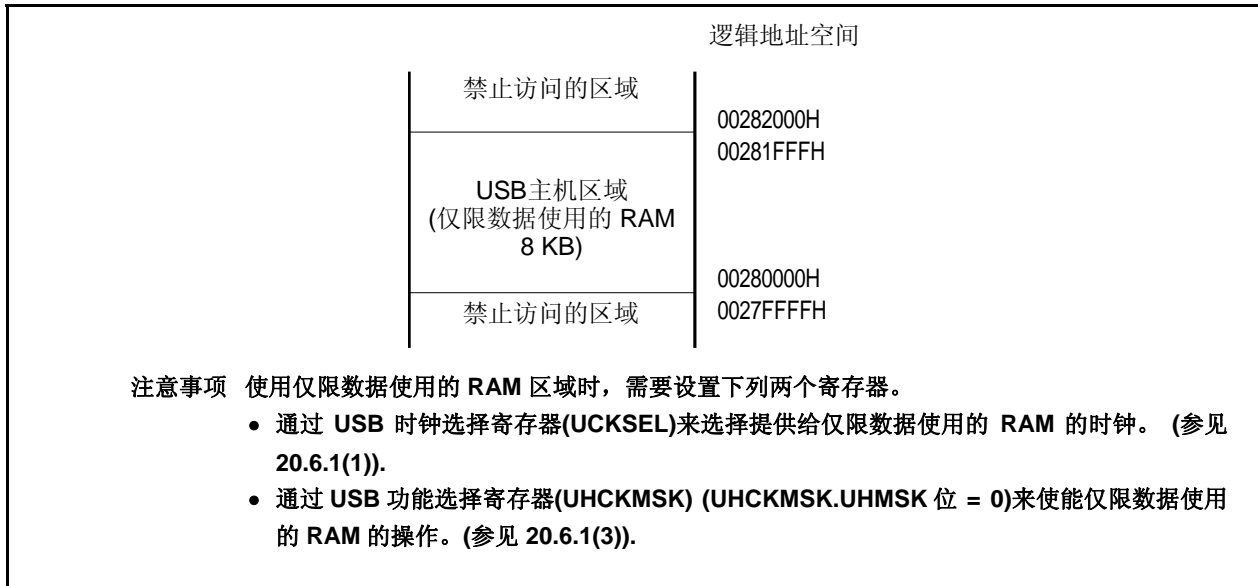


(c) 仅限数据使用的 RAM (8 KB)

在 V850ES/JG3-U 和 V850ES/JH3-U 中，8 KB 仅限数据使用的 RAM 被分配在地址 00280000H 至 00281FFFFH。

当使用 USB 主机控制器时，仅限数据使用的 RAM 被用作 USB 主机区域。如果不使用 USB 主机控制器，它可以用作仅限数据使用的 RAM。

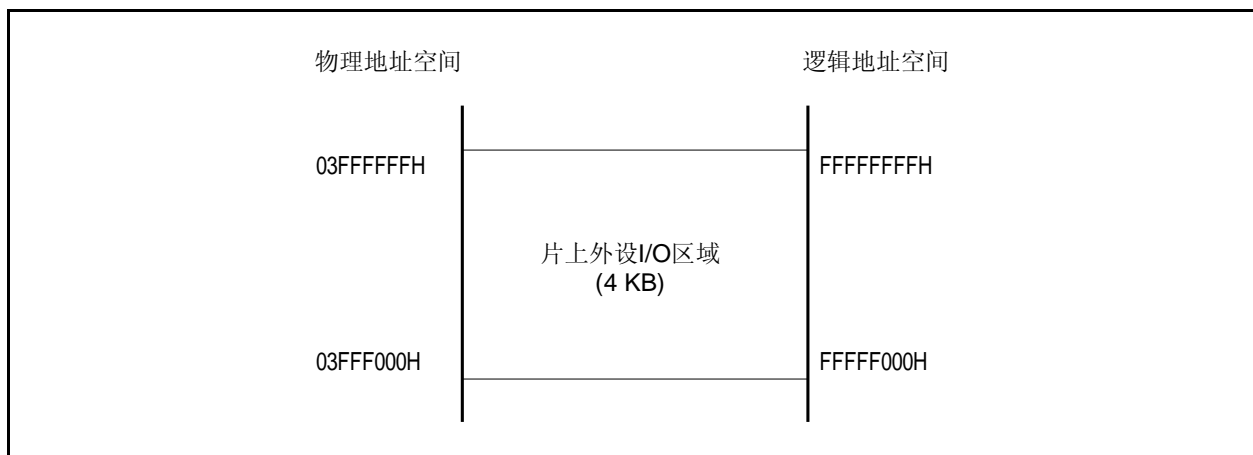
图 3-8. 仅限数据使用的 RAM 区域 (8 KB)



(3) 片上外设 I/O 区

保留地址 03FFFF00H 至 03FFFFFFH 的 4 KB 存储区域作为片上外设 I/O 区域。

图 3-9. 片上外设 I/O 区域



某些外设 I/O 寄存器映射到片内 I/O 区域，这些寄存器可以设定操作模式并监控片上外设 I/O 的状态。不能从该区域中取程序。

- 注意事项**
1. 对寄存器进行字访问，会按照先低位后高位的顺序，对该字区域执行两次半字访问，其中地址的低 2 位被忽略。
 2. 支持字节访问的寄存器，如果以半字为单位进行访问，当访问操作是读取操作时，那么该寄存器的高 8 位将不确定，数据被写入低 8 位。
 3. 没有定义为寄存器的地址区域被保留为将来扩展。如果对这些地址进行访问，其操作不确定而且也不保证操作的安全性。
 4. 内部 ROM/RAM 区域和片上外围 I/O 区域，都被分配在连续地址。
使用指针或类似的方式，递增或递减地址来访问内部 ROM/RAM 区域时，一定要注意不要越过内部 ROM/RAM 区域而错误访问到片上外围 I/O 区域。

(4) 外部存储区

13 MB (00100000H 至 001FFFFFFH, 00400000H 至 00FFFFFFH) 被分配作为外部存储区域。详情参见第五章 总线控制功能。

3.4.5 地址空间的使用建议

V850ES/JG3-U 和 V850ES/JH3-U 架构要求访问数据空间的操作数时，必须有一个寄存器用作指针以产生地址。该指针±32 KB 范围的地址可以通过操作数指令直接存取。由于能够用作指针的通用寄存器数量是有限的，当改变指针值计算地址时，为了不影响系统的性能，应该尽可能多的指定通用寄存器来保存变量，这样也可以减少程序所占的空间。

(1) 程序空间

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

要使用内部 RAM 作为程序空间，请访问下列地址。

注意事项 如果跳转指令在内部 RAM 区间的地址上限处发生，那么将不会产生跨越片上外围 I/O 区域中的预取址操作（无效取址）。

RAM 容量	访问地址
48 KB	03FF3000H 至 03FFEFFFH
40 KB	03FF5000H 至 03FFEFFFH

(2) 数据空间

对于 V850ES/JG3-U 和 V850ES/JH3-U 来说，4GB CPU 地址空间内可看作是 64 个 64MB 的地址空间。因此，26 位地址的最低有效位（第 25 位）作为符号扩展到 32 位，并作为地址分配。

(a) 地址绕回应用示例

如果 LD/ST disp16[R]指令使用 R = r0（零寄存器），对于从 00000000H 开始的±32KB 地址范围空间，就可以通过 disp16 的带符号扩展进行寻址。通过一个指针就可以对所有资源，包括内部硬件，进行寻址。零寄存器（r0）是由硬件将其值固定为 0 的，实际上不需要专门用作指针的寄存器。

示例：μPD70F3769

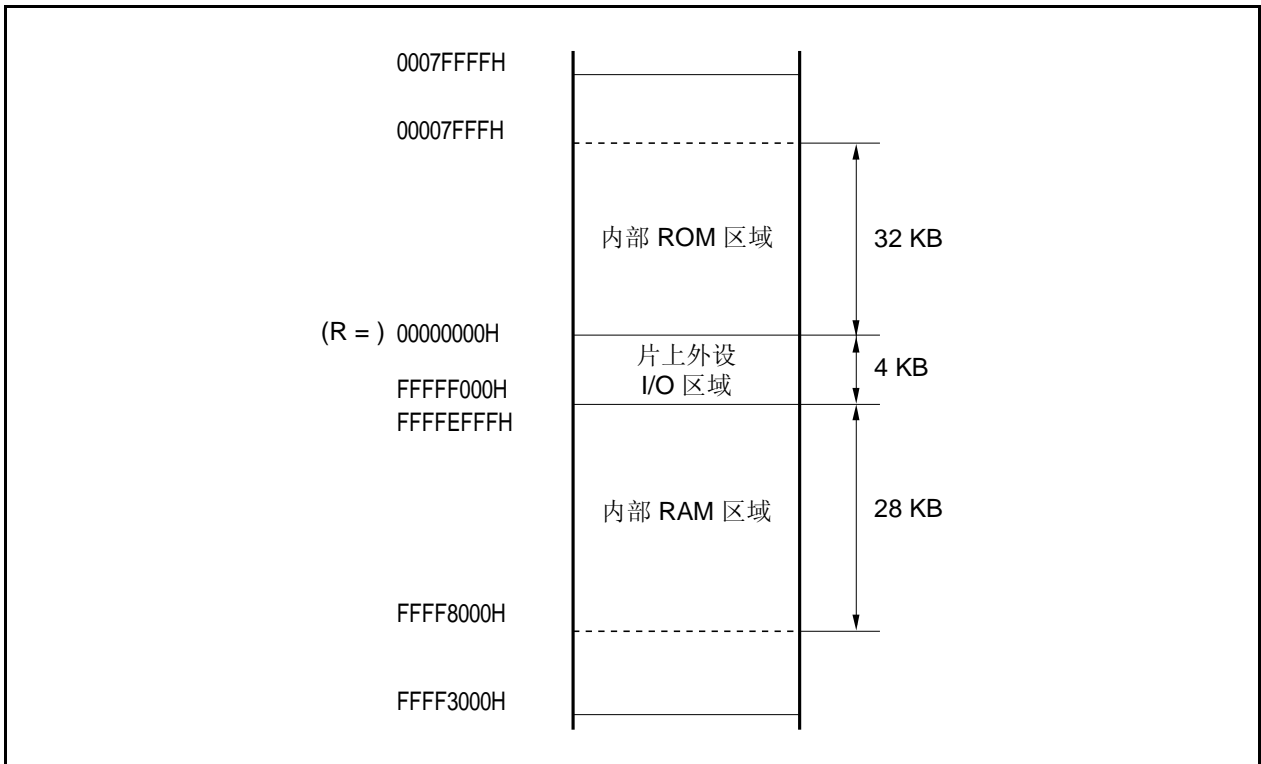
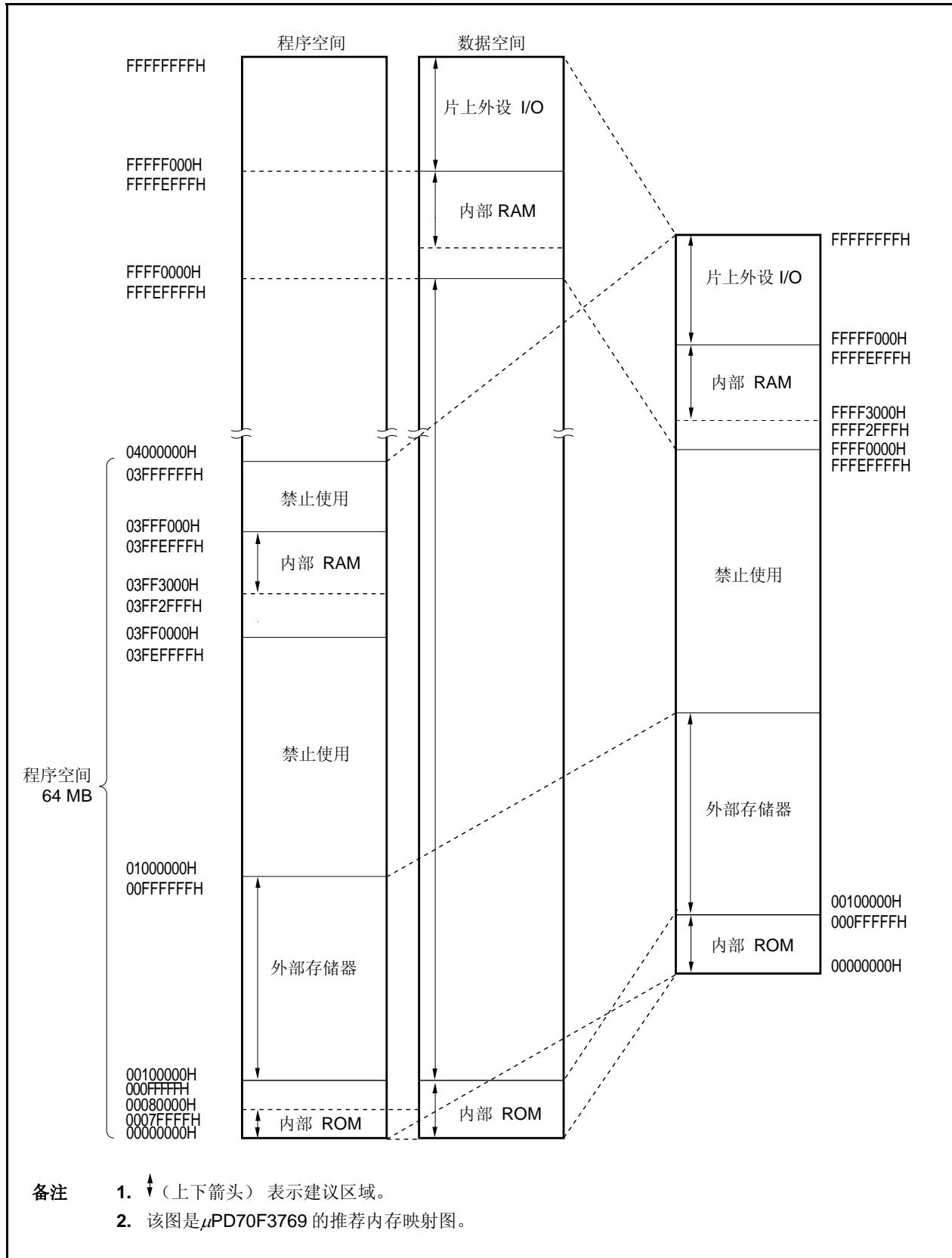


图 3-10. 建议的存储器映射图



3.4.6 外设I/O 寄存器

(1/14)

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF004H	端口 DL 寄存器	PDL	R/W			√	0000H ^{注1}
FFFFF004H	端口 DL 寄存器 L	PDLL		√	√		00H ^{注1}
FFFFF005H	端口 DL 寄存器 H	PDLH		√	√		00H ^{注1}
FFFFF006H	端口 DH 寄存器 ^{注2}	PDH ^{注2}		√	√		00H ^{注1}
FFFFF008H	端口 CS 寄存器 ^{注2}	PCS ^{注2}		√	√		00H ^{注1}
FFFFF00AH	端口 CT 寄存器	PCT		√	√		00H ^{注1}
FFFFF00CH	端口 CM 寄存器	PCM		√	√		00H ^{注1}
FFFFF024H	端口 DL 模式寄存器	PMDL				√	FFFFH
FFFFF024H	端口 DL 模式寄存器 L	PMDLL		√	√		FFH
FFFFF025H	端口 DL 模式寄存器 H	PMDLH		√	√		FFH
FFFFF026H	端口 DH 模式寄存器 ^{注2}	PMDH ^{注2}		√	√		FFH
FFFFF028H	端口 CS 模式寄存器 ^{注2}	PMCS ^{注2}		√	√		FFH
FFFFF02AH	端口 CT 模式寄存器	PMCT		√	√		FFH
FFFFF02CH	端口 CM 模式寄存器	PMCM		√	√		FFH
FFFFF044H	端口 DL 模式控制寄存器	PMCDL				√	0000H
FFFFF044H	端口 DL 模式控制寄存器 L	PMCDLL		√	√		00H
FFFFF045H	端口 DL 模式控制寄存器 H	PMCDLH		√	√		00H
FFFFF046H	端口 DH 模式控制寄存器 ^{注2}	PMCDH ^{注2}		√	√		00H
FFFFF048H	端口 CS 模式控制寄存器 ^{注2}	PMCCS ^{注2}		√	√		00H
FFFFF04AH	端口 CT 模式控制寄存器	PMCCT		√	√		00H
FFFFF04CH	端口 CM 模式控制寄存器	PMCCM		√	√		00H
FFFFF066H	总线宽度配置寄存器	BSC				√	5555H
FFFFF06EH	系统等待控制寄存器	VSWC			√		77H
FFFFF080H	DMA 源地址寄存器 0L	DSA0L				√	不确定
FFFFF082H	DMA 源地址寄存器 0H	DSA0H				√	不确定
FFFFF084H	DMA 目的地址寄存器 0L	DDA0L				√	不确定
FFFFF086H	DMA 目的地址寄存器 0H	DDA0H				√	不确定
FFFFF088H	DMA 源地址寄存器 1L	DSA1L				√	不确定
FFFFF08AH	DMA 源地址寄存器 1H	DSA1H			√	不确定	
FFFFF08CH	DMA 目的地址寄存器 1L	DDA1L			√	不确定	
FFFFF08EH	DMA 目的地址寄存器 1H	DDA1H			√	不确定	
FFFFF090H	DMA 源地址寄存器 2L	DSA2L			√	不确定	
FFFFF092H	DMA 源地址寄存器 2H	DSA2H			√	不确定	
FFFFF094H	DMA 目的地址寄存器 2L	DDA2L			√	不确定	
FFFFF096H	DMA 目的地址寄存器 2H	DDA2H			√	不确定	
FFFFF098H	DMA 源地址寄存器 3L	DSA3L			√	不确定	
FFFFF09AH	DMA 源地址寄存器 3H	DSA3H			√	不确定	
FFFFF09CH	DMA 目的地址寄存器 3L	DDA3L			√	不确定	
FFFFF09EH	DMA 目的地址寄存器 3H	DDA3H			√	不确定	

注 1. 输出锁存是 00H 或 0000H。当这些寄存器在输入模式下，读取的是引脚状态。

2. 仅限 V850ES/JH3-U

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF0C0H	DMA 传输计数寄存器 0	DBC0	R/W			√	不确定
FFFFF0C2H	DMA 传输计数寄存器 1	DBC1				√	不确定
FFFFF0C4H	DMA 传输计数寄存器 2	DBC2				√	不确定
FFFFF0C6H	DMA 传输计数寄存器 3	DBC3				√	不确定
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
FFFFF100H	中断屏蔽寄存器 0	IMR0				√	FFFFH
FFFFF100H	中断屏蔽寄存器 0L	IMR0L		√	√		FFH
FFFFF101H	中断屏蔽寄存器 0H	IMR0H		√	√		FFH
FFFFF102H	中断屏蔽寄存器 1	IMR1				√	FFFFH
FFFFF102H	中断屏蔽寄存器 1L	IMR1L	√	√		FFH	
FFFFF103H	中断屏蔽寄存器 1H	IMR1H	√	√		FFH	
FFFFF104H	中断屏蔽寄存器 2	IMR2			√	FFFFH	
FFFFF104H	中断屏蔽寄存器 2L	IMR2L	√	√		FFH	
FFFFF105H	中断屏蔽寄存器 2H	IMR2H	√	√		FFH	
FFFFF106H	中断屏蔽寄存器 3	IMR3			√	FFFFH	
FFFFF106H	中断屏蔽寄存器 3L	IMR3L	√	√		FFH	
FFFFF107H	中断屏蔽寄存器 3H	IMR3H	√	√		FFH	
FFFFF108H	中断屏蔽寄存器 4	IMR4			√	FFFFH	
FFFFF108H	中断屏蔽寄存器 4L	IMR4L	√	√		FFH	
FFFFF109H	中断屏蔽寄存器 4H	IMR4H	√	√		FFH	
FFFFF10AH	中断屏蔽寄存器 5	IMR5			√	FFFFH	
FFFFF10AH	中断屏蔽寄存器 5L	IMR5L	√	√		FFH	
FFFFF10BH	中断屏蔽寄存器 5H	IMR5H	√	√		FFH	
FFFFF110H	中断控制寄存器	LVIIC	√	√		47H	
FFFFF112H	中断控制寄存器	PIC00	√	√		47H	
FFFFF114H	中断控制寄存器	PIC01	√	√		47H	
FFFFF116H	中断控制寄存器	PIC02	√	√		47H	
FFFFF118H	中断控制寄存器	PIC03 ^注	√	√		47H	
FFFFF11AH	中断控制寄存器	PIC04 ^注	√	√		47H	
FFFFF11CH	中断控制寄存器	PIC05	√	√		47H	
FFFFF11EH	中断控制寄存器	PIC06	√	√		47H	
FFFFF120H	中断控制寄存器	PIC07	√	√		47H	
FFFFF122H	中断控制寄存器	PIC08	√	√		47H	

注 仅限 V850ES/JH3-U

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF124H	中断控制寄存器	PIC09	R/W	√	√		47H
FFFFF126H	中断控制寄存器	PIC10		√	√		47H
FFFFF128H	中断控制寄存器	PIC11		√	√		47H
FFFFF12AH	中断控制寄存器	PIC12		√	√		47H
FFFFF12CH	中断控制寄存器	PIC13		√	√		47H
FFFFF12EH	中断控制寄存器	PIC14		√	√		47H
FFFFF130H	中断控制寄存器	PIC15		√	√		47H
FFFFF132H	中断控制寄存器	PIC16		√	√		47H
FFFFF134H	中断控制寄存器	PIC17		√	√		47H
FFFFF136H	中断控制寄存器	PIC18		√	√		47H
FFFFF138H	中断控制寄存器	TAB0OVIC		√	√		47H
FFFFF13AH	中断控制寄存器	TAB0CCIC0		√	√		47H
FFFFF13CH	中断控制寄存器	TAB0CCIC1		√	√		47H
FFFFF13EH	中断控制寄存器	TAB0CCIC2		√	√		47H
FFFFF140H	中断控制寄存器	TAB0CCIC3		√	√		47H
FFFFF142H	中断控制寄存器	TAB1OVIC		√	√		47H
FFFFF144H	中断控制寄存器	TAB1CCIC0		√	√		47H
FFFFF146H	中断控制寄存器	TAB1CCIC1		√	√		47H
FFFFF148H	中断控制寄存器	TAB1CCIC2		√	√		47H
FFFFF14AH	中断控制寄存器	TAB1CCIC3		√	√		47H
FFFFF14CH	中断控制寄存器	TT0OVIC		√	√		47H
FFFFF14EH	中断控制寄存器	TT0CCIC0		√	√		47H
FFFFF150H	中断控制寄存器	TT0CCIC1		√	√		47H
FFFFF152H	中断控制寄存器	TT0IECIC		√	√		47H
FFFFF154H	中断控制寄存器	TAA0OVIC		√	√		47H
FFFFF156H	中断控制寄存器	TAA0CCIC0		√	√		47H
FFFFF158H	中断控制寄存器	TAA0CCIC1		√	√		47H
FFFFF15AH	中断控制寄存器	TAA1OVIC		√	√		47H
FFFFF15CH	中断控制寄存器	TAA1CCIC0		√	√		47H
FFFFF15EH	中断控制寄存器	TAA1CCIC1		√	√		47H
FFFFF160H	中断控制寄存器	TAA2OVIC		√	√		47H
FFFFF162H	中断控制寄存器	TAA2CCIC0		√	√		47H
FFFFF164H	中断控制寄存器	TAA2CCIC1		√	√		47H
FFFFF166H	中断控制寄存器	TAA3OVIC		√	√		47H
FFFFF168H	中断控制寄存器	TAA3CCIC0		√	√		47H
FFFFF16AH	中断控制寄存器	TAA3CCIC1		√	√		47H
FFFFF16CH	中断控制寄存器	TAA4OVIC		√	√		47H
FFFFF16EH	中断控制寄存器	TAA4CCIC0		√	√		47H
FFFFF170H	中断控制寄存器	TAA4CCIC1		√	√		47H
FFFFF172H	中断控制寄存器	TAA5OVIC		√	√		47H
FFFFF174H	中断控制寄存器	TAA5CCIC0	√	√		47H	

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF176H	中断控制寄存器	TAA5CCIC1	R/W	√	√		47H
FFFFF178H	中断控制寄存器	TM0EQIC0		√	√		47H
FFFFF17AH	中断控制寄存器	TM1EQIC0		√	√		47H
FFFFF17CH	中断控制寄存器	TM2EQIC0		√	√		47H
FFFFF17EH	中断控制寄存器	TM3EQIC0		√	√		47H
FFFFF180H	中断控制寄存器	CF0RIC/IICIC1		√	√		47H
FFFFF182H	中断控制寄存器	CF0TIC		√	√		47H
FFFFF184H	中断控制寄存器	CF1RIC		√	√		47H
FFFFF186H	中断控制寄存器	CF1TIC		√	√		47H
FFFFF188H	中断控制寄存器	CF2RIC		√	√		47H
FFFFF18AH	中断控制寄存器	CF2TIC		√	√		47H
FFFFF18CH	中断控制寄存器	CF3RIC		√	√		47H
FFFFF18EH	中断控制寄存器	CF3TIC		√	√		47H
FFFFF190H	中断控制寄存器	CF4RIC		√	√		47H
FFFFF192H	中断控制寄存器	CF4TIC		√	√		47H
FFFFF194H	中断控制寄存器	UC0RIC		√	√		47H
FFFFF196H	中断控制寄存器	UC0TIC		√	√		47H
FFFFF198H	中断控制寄存器	UC1RIC/IICIC2		√	√		47H
FFFFF19AH	中断控制寄存器	UC1TIC		√	√		47H
FFFFF19CH	中断控制寄存器	UC2RIC		√	√		47H
FFFFF19EH	中断控制寄存器	UC2TIC		√	√		47H
FFFFF1A0H	中断控制寄存器	UC3RIC/IICIC0		√	√		47H
FFFFF1A2H	中断控制寄存器	UC3TIC		√	√		47H
FFFFF1A4H	中断控制寄存器	UC4RIC		√	√		47H
FFFFF1A6H	中断控制寄存器	UC4TIC		√	√		47H
FFFFF1A8H	中断控制寄存器	ADIC		√	√		47H
FFFFF1AAH	中断控制寄存器	DMAIC0		√	√		47H
FFFFF1ACH	中断控制寄存器	DMAIC1		√	√		47H
FFFFF1AEH	中断控制寄存器	DMAIC2		√	√		47H
FFFFF1B0H	中断控制寄存器	DMAIC3		√	√		47H
FFFFF1B2H	中断控制寄存器	KRIC	√	√		47H	
FFFFF1B4H	中断控制寄存器	RTC0IC	√	√		47H	
FFFFF1B6H	中断控制寄存器	RTC1IC	√	√		47H	
FFFFF1B8H	中断控制寄存器	RTC2IC	√	√		47H	
FFFFF1C2H	中断控制寄存器	UHIC0	√	√		47H	
FFFFF1C4H	中断控制寄存器	UHIC1	√	√		47H	
FFFFF1C6H	中断控制寄存器	UHIC2	√	√		47H	
FFFFF1C8H	中断控制寄存器	UFIC0	√	√		47H	
FFFFF1CAH	中断控制寄存器	UFIC1	√	√		47H	
FFFFF1FAH	正在服务的优先级寄存器	ISPR	R	√	√	00H	
FFFFF1FCH	命令寄存器	PRCMD	W		√	不确定	
FFFFF1FEH	省电控制寄存器	PSC	R/W	√	√	00H	

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF200H	A/D 转换器 模式寄存器 0	ADA0M0	R/W	√	√		00H
FFFFF201H	A/D 转换器 模式寄存器 1	ADA0M1		√	√		00H
FFFFF202H	A/D 转换器 通道指定寄存器	ADA0S		√	√		00H
FFFFF203H	A/D 转换器 模式寄存器 2	ADA0M2		√	√		00H
FFFFF204H	掉电比较模式寄存器	ADA0PFM		√	√		00H
FFFFF205H	掉电比较阈值寄存器	ADA0PFT		√	√		00H
FFFFF210H	A/D 转换结果 寄存器 0	ADA0CR0	R			√	不确定
FFFFF211H	A/D 转换结果 寄存器 0H	ADA0CR0H			√		不确定
FFFFF212H	A/D 转换结果 寄存器 1	ADA0CR1				√	不确定
FFFFF213H	A/D 转换结果 寄存器 1H	ADA0CR1H			√		不确定
FFFFF214H	A/D 转换结果 寄存器 2	ADA0CR2				√	不确定
FFFFF215H	A/D 转换结果 寄存器 2H	ADA0CR2H			√		不确定
FFFFF216H	A/D 转换结果 寄存器 3	ADA0CR3				√	不确定
FFFFF217H	A/D 转换结果 寄存器 3H	ADA0CR3H			√		不确定
FFFFF218H	A/D 转换结果 寄存器 4	ADA0CR4				√	不确定
FFFFF219H	A/D 转换结果 寄存器 4H	ADA0CR4H			√		不确定
FFFFF21AH	A/D 转换结果 寄存器 5	ADA0CR5				√	不确定
FFFFF21BH	A/D 转换结果 寄存器 5H	ADA0CR5H			√		不确定
FFFFF21CH	A/D 转换结果 寄存器 6	ADA0CR6				√	不确定
FFFFF21DH	A/D 转换结果 寄存器 6H	ADA0CR6H			√		不确定
FFFFF21EH	A/D 转换结果 寄存器 7	ADA0CR7				√	不确定
FFFFF21FH	A/D 转换结果 寄存器 7H	ADA0CR7H			√		不确定
FFFFF220H	A/D 转换结果 寄存器 8	ADA0CR8				√	不确定
FFFFF221H	A/D 转换结果 寄存器 8H	ADA0CR8H			√		不确定
FFFFF222H	A/D 转换结果 寄存器 9	ADA0CR9				√	不确定
FFFFF223H	A/D 转换结果 寄存器 9H	ADA0CR9H			√		不确定
FFFFF224H	A/D 转换结果 寄存器 10	ADA0CR10			√	不确定	
FFFFF225H	A/D 转换结果 寄存器 10H	ADA0CR10H		√		不确定	
FFFFF226H	A/D 转换结果 寄存器 11	ADA0CR11			√	不确定	
FFFFF227H	A/D 转换结果 寄存器 11H	ADA0CR11H		√		不确定	
FFFFF280H	D/A 转换值设置寄存器 0	DA0CS0	R/W		√		00H
FFFFF281H	D/A 转换值设置寄存器 1	DA0CS1			√		00H
FFFFF282H	D/A 转换器模式寄存器	DA0M		√	√		00H
FFFFF300H	按键返回模式寄存器	KRM		√	√		00H
FFFFF308H	选择器操作控制寄存器 0	SELCNT0		√	√		00H
FFFFF310H	CRC 输入寄存器	CRCIN			√		00H
FFFFF312H	CRC 数据寄存器	CRCD				√	0000H
FFFFF320H	预分频模式寄存器 1	PRSM1		√	√		00H
FFFFF321H	预分频比较寄存器 1	PRSCM1			√		00H
FFFFF324H	预分频 模式寄存器 2	PRSM2		√	√		00H
FFFFF325H	预分频比较寄存器 2	PRSCM2			√		00H
FFFFF328H	预分频 模式寄存器 3	PRSM3		√	√		00H
FFFFF329H	预分频比较寄存器 3	PRSCM3			√		00H

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF340H	IIC 分频时钟选择寄存器 0	OCKS0	R/W		√		00H
FFFFF344H	IIC 分频时钟选择寄存器 1	OCKS1			√		00H
FFFFF400H	端口 0 寄存器	P0		√	√		00H ^{注1}
FFFFF402H	端口 1 寄存器	P1		√	√		00H ^{注1}
FFFFF404H	端口 2 寄存器 ^{注2}	P2 ^{注2}		√	√		00H ^{注1}
FFFFF406H	端口 3 寄存器	P3		√	√		00H ^{注1}
FFFFF408H	端口 4 寄存器	P4		√	√		00H ^{注1}
FFFFF40AH	端口 5 寄存器	P5		√	√		00H ^{注1}
FFFFF40CH	端口 6 寄存器	P6		√	√		00H ^{注1}
FFFFF40EH	端口 7 寄存器 L	P7L		√	√		00H ^{注1}
FFFFF40FH	端口 7 寄存器 H	P7H		√	√		00H ^{注1}
FFFFF412H	端口 9 寄存器	P9				√	0000H ^{注1}
FFFFF412H	端口 9 寄存器 L	P9L		√	√		00H ^{注1}
FFFFF413H	端口 9 寄存器 H	P9H		√	√		00H ^{注1}
FFFFF420H	端口 0 模式 寄存器	PM0		√	√		FFH
FFFFF422H	端口 1 模式 寄存器	PM1		√	√		FFH
FFFFF424H	端口 2 模式 寄存器 ^{注2}	PM2 ^{注2}		√	√		FFH
FFFFF426H	端口 3 模式 寄存器	PM3		√	√		FFH
FFFFF428H	端口 4 模式 寄存器	PM4		√	√		FFH
FFFFF42AH	端口 5 模式 寄存器	PM5		√	√		FFH
FFFFF42CH	端口 6 模式 寄存器	PM6		√	√		FFH
FFFFF42EH	端口 7 模式 寄存器 L	PM7L		√	√		FFH
FFFFF42FH	端口 7 模式 寄存器 H	PM7H		√	√		FFH
FFFFF432H	端口 9 模式 寄存器	PM9				√	FFFFH
FFFFF432H	端口 9 模式 寄存器 L	PM9L		√	√		FFH
FFFFF433H	端口 9 模式 寄存器 H	PM9H		√	√		FFH
FFFFF440H	端口 0 模式 控制 寄存器	PMC0		√	√		00H
FFFFF444H	端口 2 模式 控制 寄存器 ^{注2}	PMC2 ^{注2}		√	√		00H
FFFFF446H	端口 3 模式 控制 寄存器	PMC3		√	√		00H
FFFFF448H	端口 4 模式 控制 寄存器	PMC4		√	√		00H
FFFFF44AH	端口 5 模式 控制 寄存器	PMC5	√	√		00H	
FFFFF44CH	端口 6 模式 控制 寄存器	PMC6	√	√		00H	
FFFFF452H	端口 9 模式 控制 寄存器	PMC9			√	0000H	
FFFFF452H	端口 9 模式 控制 寄存器 L	PMC9L	√	√		00H	
FFFFF453H	端口 9 模式 控制 寄存器 H	PMC9H	√	√		00H	
FFFFF460H	端口 0 功能 控制 寄存器	PFC0	√	√		00H	
FFFFF464H	端口 2 功能 控制 寄存器 ^{注2}	PFC2 ^{注2}	√	√		00H	
FFFFF466H	端口 3 功能 控制 寄存器	PFC3	√	√		00H	
FFFFF468H	端口 4 功能 控制 寄存器	PFC4	√	√		00H	

注 1. 输出锁存是 00H 或 0000H。当这些寄存器在输入模式下，读取的是引脚状态。

2. 仅限 V850ES/JH3-U

地址	功能寄存器名称	符号	R/W	操作位数			默认值	
				1	8	16		
FFFFF46AH	端口 5 功能控制寄存器	PFC5	R/W	√	√		00H	
FFFFF46CH	端口 6 功能控制寄存器	PFC6		√	√		00H	
FFFFF472H	端口 9 功能控制寄存器	PFC9				√	0000H	
FFFFF472H	端口 9 功能控制寄存器 L	PFC9L		√	√		00H	
FFFFF473H	端口 9 功能控制寄存器 H	PFC9H		√	√		00H	
FFFFF484H	数据等待控制寄存器 0	DWC0				√	7777H	
FFFFF488H	地址等待控制寄存器	AWC				√	FFFFH	
FFFFF48AH	总线周期控制寄存器	BCC				√	AAAAH	
FFFFF540H	TAB0 控制寄存器 0	TAB0CTL0		√	√		00H	
FFFFF541H	TAB0 控制寄存器 1	TAB0CTL1		√	√		00H	
FFFFF542H	TAB0 I/O 控制寄存器 0	TAB0IOC0		√	√		00H	
FFFFF543H	TAB0 I/O 控制寄存器 1	TAB0IOC1		√	√		00H	
FFFFF544H	TAB0 I/O 控制寄存器 2	TAB0IOC2		√	√		00H	
FFFFF545H	TAB0 选项寄存器 0	TAB0OPT0		√	√		00H	
FFFFF546H	TAB0 捕获/比较寄存器 0	TAB0CCR0			√	0000H		
FFFFF548H	TAB0 捕获/比较寄存器 1	TAB0CCR1			√	0000H		
FFFFF54AH	TAB0 捕获/比较寄存器 2	TAB0CCR2			√	0000H		
FFFFF54CH	TAB0 捕获/比较寄存器 3	TAB0CCR3			√	0000H		
FFFFF54EH	TAB0 计数器读取缓冲寄存器	TAB0CNT	R		√	0000H		
FFFFF550H	TAB0 I/O 控制寄存器 4	TAB0IOC4	R/W	√	√		00H	
FFFFF560H	TAB1 控制寄存器 0	TAB1CTL0		√	√		00H	
FFFFF561H	TAB1 控制寄存器 1	TAB1CTL1		√	√		00H	
FFFFF562H	TAB1 I/O 控制寄存器 0	TAB1IOC0		√	√		00H	
FFFFF563H	TAB1 I/O 控制寄存器 1	TAB1IOC1		√	√		00H	
FFFFF564H	TAB1 I/O 控制寄存器 2	TAB1IOC2		√	√		00H	
FFFFF565H	TAB1 选项寄存器 0	TAB1OPT0		√	√		00H	
FFFFF566H	TAB1 捕获/比较寄存器 0	TAB1CCR0				√	0000H	
FFFFF568H	TAB1 捕获/比较寄存器 1	TAB1CCR1				√	0000H	
FFFFF56AH	TAB1 捕获/比较寄存器 2	TAB1CCR2				√	0000H	
FFFFF56CH	TAB1 捕获/比较寄存器 3	TAB1CCR3				√	0000H	
FFFFF56EH	TAB1 计数器读取缓冲寄存器	TAB1CNT		R		√	0000H	
FFFFF570H	TAB1 I/O 控制寄存器 4	TAB1IOC4		R/W	√	√		00H
FFFFF580H	TAB1 选项寄存器 1	TAB1OPT1			√	√		00H
FFFFF581H	TAB1 选项寄存器 2	TAB1OPT2	√		√		00H	
FFFFF582H	TAB1 I/O 控制寄存器 3	TAB1IOC3	√		√		A8H	
FFFFF584H	TAB1 死区时间比较寄存器 1	TAB1DTC				√	0000H	
FFFFF590H	高阻抗输出控制寄存器 0	HZACTL0	√		√		00H	
FFFFF591H	高阻抗输出控制寄存器 1	HZACTL1	√		√		00H	
FFFFF600H	TMT0 控制寄存器 0	TT0CTL0	√		√		00H	
FFFFF601H	TMT0 控制寄存器 1	TT0CTL1	√		√		00H	
FFFFF602H	TMT0 控制寄存器 2	TT0CTL2	√		√		00H	
FFFFF603H	TMT0 I/O 控制寄存器 0	TT0IOC0	√		√		00H	
FFFFF604H	TMT0 I/O 控制寄存器 1	TT0IOC1	√		√		00H	

地址	功能寄存器名称	符号	R/W	操作位数			默认值	
				1	8	16		
FFFFF605H	TMT0 I/O 控制寄存器 2	TT0IOC2	R/W	√	√		00H	
FFFFF606H	TMT0 I/O 控制寄存器 3	TT0IOC3		√	√		00H	
FFFFF607H	TMT0 选项寄存器 0	TT0OPT0		√	√		00H	
FFFFF608H	TMT0 选项寄存器 1	TT0OPT1		√	√		00H	
FFFFF609H	TMT0 选项寄存器 2	TT0OPT2		√	√		00H	
FFFFF60AH	TMT0 捕获/比较寄存器 0	TT0CCR0				√	0000H	
FFFFF60CH	TMT0 捕获/比较寄存器 1	TT0CCR1				√	0000H	
FFFFF60EH	TMT0 计数器读取缓冲寄存器	TT0CNT	R			√	0000H	
FFFFF610H	TMT0 计数器写入寄存器	TT0TCW	R/W			√	0000H	
FFFFF630H	TAA0 控制寄存器 0	TAA0CTL0		√	√		00H	
FFFFF631H	TAA0 控制寄存器 1	TAA0CTL1		√	√		00H	
FFFFF632H	TAA0 I/O 控制寄存器 0	TAA0IOC0		√	√		00H	
FFFFF633H	TAA0 I/O 控制寄存器 1	TAA0IOC1		√	√		00H	
FFFFF634H	TAA0 I/O 控制寄存器 2	TAA0IOC2		√	√		00H	
FFFFF635H	TAA0 选项寄存器 0	TAA0OPT0		√	√		00H	
FFFFF636H	TAA0 捕获/比较寄存器 0	TAA0CCR0				√	0000H	
FFFFF638H	TAA0 捕获/比较寄存器 1	TAA0CCR1				√	0000H	
FFFFF63AH	TAA0 计数器读取缓冲寄存器	TAA0CNT		R			√	0000H
FFFFF63CH	TAA0 I/O 控制寄存器 4	TAA0IOC4		√	√		00H	
FFFFF63DH	TAA0 选项寄存器 1	TAA0OPT1		√	√		00H	
FFFFF640H	TAA1 控制寄存器 0	TAA1CTL0		√	√		00H	
FFFFF641H	TAA1 控制寄存器 1	TAA1CTL1		√	√		00H	
FFFFF642H	TAA1 I/O 控制寄存器 0	TAA1IOC0		√	√		00H	
FFFFF643H	TAA1 I/O 控制寄存器 1	TAA1IOC1		√	√		00H	
FFFFF644H	TAA1 I/O 控制寄存器 2	TAA1IOC2		√	√		00H	
FFFFF645H	TAA1 选项寄存器 0	TAA1OPT0		√	√		00H	
FFFFF646H	TAA1 捕获/比较寄存器 0	TAA1CCR0				√	0000H	
FFFFF648H	TAA1 捕获/比较寄存器 1	TAA1CCR1			√	0000H		
FFFFF64AH	TAA1 计数器读取缓冲寄存器	TAA1CNT	R			√	0000H	
FFFFF64CH	TAA1 I/O 控制寄存器 4	TAA1IOC4	√	√		00H		
FFFFF650H	TAA2 控制寄存器 0	TAA2CTL0	√	√		00H		
FFFFF651H	TAA2 控制寄存器 1	TAA2CTL1	√	√		00H		
FFFFF652H	TAA2 I/O 控制寄存器 0	TAA2IOC0	√	√		00H		
FFFFF653H	TAA2 I/O 控制寄存器 1	TAA2IOC1	√	√		00H		
FFFFF654H	TAA2 I/O 控制寄存器 2	TAA2IOC2	√	√		00H		
FFFFF655H	TAA2 选项寄存器 0	TAA2OPT0	√	√		00H		
FFFFF656H	TAA2 捕获/比较寄存器 0	TAA2CCR0			√	0000H		
FFFFF658H	TAA2 捕获/比较寄存器 1	TAA2CCR1			√	0000H		
FFFFF65AH	TAA2 计数器读取缓冲寄存器	TAA2CNT	R			√	0000H	
FFFFF65CH	TAA2 I/O 控制寄存器 4	TAA2IOC4	√	√		00H		
FFFFF65DH	TAA2 选项寄存器 1	TAA2OPT1	√	√		00H		
FFFFF660H	TAA3 控制寄存器 0	TAA3CTL0	√	√		00H		
FFFFF661H	TAA3 控制寄存器 1	TAA3CTL1	√	√		00H		

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFF662H	TAA3 I/O 控制 寄存器 0	TAA3IOC0	R/W	√	√		00H
FFFF663H	TAA3 I/O 控制 寄存器 1	TAA3IOC1		√	√		00H
FFFF664H	TAA3 I/O 控制 寄存器 2	TAA3IOC2		√	√		00H
FFFF665H	TAA3 选项寄存器 0	TAA3OPT0		√	√		00H
FFFF666H	TAA3 捕获/比较寄存器 0	TAA3CCR0				√	0000H
FFFF668H	TAA3 捕获/比较寄存器 1	TAA3CCR1				√	0000H
FFFF66AH	TAA3 计数器读取缓冲寄存器	TAA3CNT	R			√	0000H
FFFF66CH	TAA3 I/O 控制 寄存器 4	TAA3IOC4	R/W	√	√		00H
FFFF670H	TAA4 控制 寄存器 0	TAA4CTL0		√	√		00H
FFFF671H	TAA4 控制 寄存器 1	TAA4CTL1		√	√		00H
FFFF676H	TAA4 捕获/比较 寄存器 0	TAA4CCR0				√	0000H
FFFF678H	TAA4 捕获/比较 寄存器 1	TAA4CCR1				√	0000H
FFFF67AH	TAA4 计数器读取缓冲寄存器	TAA4CNT	R			√	0000H
FFFF680H	TAA5 控制 寄存器 0	TAA5CTL0	R/W	√	√		00H
FFFF681H	TAA5 控制 寄存器 1	TAA5CTL1		√	√		00H
FFFF682H	TAA5 I/O 控制 寄存器 0	TAA5IOC0		√	√		00H
FFFF683H	TAA5 I/O 控制 寄存器 1	TAA5IOC1		√	√		00H
FFFF684H	TAA5 I/O 控制 寄存器 2	TAA5IOC2		√	√		00H
FFFF685H	TAA5 选项 寄存器 0	TAA5OPT0		√	√		00H
FFFF686H	TAA5 捕获/比较寄存器 0	TAA5CCR0				√	0000H
FFFF688H	TAA5 捕获/比较寄存器 1	TAA5CCR1				√	0000H
FFFF68AH	TAA5 计数器读取缓冲寄存器	TAA5CNT	R			√	0000H
FFFF68CH	TAA5 I/O 控制寄存器 4	TAA5IOC4	R/W	√	√		00H
FFFF6C0H	振荡稳定时间选择寄存器	OSTS			√		06H
FFFF6C1H	PLL 入锁时间指定寄存器	PLLS			√		03H
FFFF6D0H	看门狗定时器模式寄存器 2	WDTM2			√		67H
FFFF6D1H	看门狗定时器使能寄存器	WDTE			√		9AH
FFFF6E0H	实时输出缓冲寄存器 0L	RTBL0		√	√		00H
FFFF6E2H	实时输出缓冲寄存器 0H	RTBH0		√	√		00H
FFFF6E4H	实时输出端口模式寄存器 0	RTPM0		√	√		00H
FFFF6E5H	实时输出端口控制寄存器 0	RTPC0		√	√		00H
FFFF700H	端口 0 功能控制扩展寄存器	PFCE0		√	√		00H
FFFF704H	端口 2 功能控制扩展寄存器 [※]	PFCE2 [※]		√	√		00H
FFFF706H	端口 3 功能控制扩展寄存器	PFCE3		√	√		00H
FFFF708H	端口 4 功能控制扩展寄存器	PFCE4		√	√		00H
FFFF70AH	端口 5 功能控制扩展寄存器	PFCE5		√	√		00H
FFFF70CH	端口 6 功能控制扩展寄存器	PFCE6		√	√		00H
FFFF712H	端口 9 功能控制扩展寄存器	PFCE9				√	0000H
FFFF712H	端口 9 功能控制扩展寄存器 L	PFCE9L		√	√		00H
FFFF713H	端口 9 功能控制扩展寄存器 H	PFCE9H		√	√		00H
FFFF724H	TAA 噪声消除控制寄存器	TANFC		√	√		00H
FFFF726H	TMT 噪声消除控制寄存器	TTNFC		√	√		00H

注 仅限 V850ES/JH3-U

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF728H	噪声消除控制寄存器	INTNFC	R/W	√	√		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 触发因素寄存器 0	DTFR0		√	√		00H
FFFFF812H	DMA 触发因素寄存器 1	DTFR1		√	√		00H
FFFFF814H	DMA 触发因素寄存器 2	DTFR2		√	√		00H
FFFFF816H	DMA 触发因素寄存器 3	DTFR3		√	√		00H
FFFFF820H	省电模式寄存器	PSMR		√	√		00H
FFFFF822H	时钟控制寄存器	CKC		√	√		0AH
FFFFF824H	锁定寄存器	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
FFFFF870H	时钟监视器模式寄存器	CLM	R/W	√	√		00H
FFFFF888H	复位源标志寄存器	RESF		√	√		00H
FFFFF890H	低电压检测寄存器	LVIM		√	√		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 ^注	OCDM ^注		√	√		01H
FFFFFA00H	UARTC0 控制寄存器 0	UC0CTL0		√	√		10H
FFFFFA01H	UARTC0 控制寄存器 1	UC0CTL1			√		00H
FFFFFA02H	UARTC0 控制寄存器 2	UC0CTL2			√		FFH
FFFFFA03H	UARTC0 选项控制寄存器 0	UC0OPT0	√	√		14H	
FFFFFA04H	UARTC0 状态寄存器	UC0STR	√	√		00H	
FFFFFA06H	UARTC0 接收数据寄存器	UC0RX	R			√	01FFH
FFFFFA06H	UARTC0 接收数据寄存器 L	UC0RXL			√		FFH
FFFFFA08H	UARTC0 发送数据寄存器	UC0TX	R/W			√	01FFH
FFFFFA08H	UARTC0 发送数据寄存器 L	UC0TXL			√		FFH
FFFFFA0AH	UARTC0 选项控制寄存器 1	UC0OPT1	√	√		00H	
FFFFFA10H	UARTC1 控制寄存器 0	UC1CTL0	√	√		10H	
FFFFFA11H	UARTC1 控制寄存器 1	UC1CTL1		√		00H	
FFFFFA12H	UARTC1 控制寄存器 2	UC1CTL2		√		FFH	
FFFFFA13H	UARTC1 选项控制寄存器 0	UC1OPT0	√	√		14H	
FFFFFA14H	UARTC1 状态寄存器	UC1STR	√	√		00H	
FFFFFA16H	UARTC1 接收数据寄存器	UC1RX	R			√	01FFH
FFFFFA16H	UARTC1 接收数据寄存器 L	UC1RXL			√		FFH
FFFFFA18H	UARTC1 发送数据寄存器	UC1TX	R/W			√	01FFH
FFFFFA18H	UARTC1 发送数据寄存器 L	UC1TXL			√		FFH
FFFFFA1AH	UARTC1 选项控制寄存器 1	UC1OPT1	√	√		00H	
FFFFFA20H	UARTC2 控制寄存器 0	UC2CTL0	√	√		10H	
FFFFFA21H	UARTC2 控制寄存器 1	UC2CTL1		√		00H	
FFFFFA22H	UARTC2 控制寄存器 2	UC2CTL2		√		FFH	

注 仅限 V850ES/JG3-U

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFFA23H	UARTC2 选项控制寄存器 0	UC2OPT0	R/W	√	√		14H
FFFFFA24H	UARTC2 状态寄存器	UC2STR		√	√		00H
FFFFFA26H	UARTC2 接收数据寄存器	UC2RX	R			√	01FFH
FFFFFA26H	UARTC2 接收数据寄存器 L	UC2RXL			√		FFH
FFFFFA28H	UARTC2 发送数据寄存器	UC2TX	R/W			√	01FFH
FFFFFA28H	UARTC2 发送数据寄存器 L	UC2TXL			√		FFH
FFFFFA2AH	UARTC2 选项控制寄存器 1	UC2OPT1		√	√		00H
FFFFFA30H	UARTC3 控制寄存器 0	UC3CTL0		√	√		10H
FFFFFA31H	UARTC3 控制寄存器 1	UC3CTL1			√		00H
FFFFFA32H	UARTC3 控制寄存器 2	UC3CTL2			√		FFH
FFFFFA33H	UARTC3 选项控制寄存器 0	UC3OPT0		√	√		14H
FFFFFA34H	UARTC3 状态寄存器	UC3STR		√	√		00H
FFFFFA36H	UARTC3 接收数据寄存器	UC3RX	R			√	01FFH
FFFFFA36H	UARTC3 接收数据寄存器 L	UC3RXL			√		FFH
FFFFFA38H	UARTC3 发送数据寄存器	UC3TX	R/W			√	01FFH
FFFFFA38H	UARTC3 发送数据寄存器 L	UC3TXL			√		FFH
FFFFFA3AH	UARTC3 选项控制寄存器 1	UC3OPT1		√	√		00H
FFFFFA40H	UARTC4 控制寄存器 0	UC4CTL0		√	√		10H
FFFFFA41H	UARTC4 控制寄存器 1	UC4CTL1			√		00H
FFFFFA42H	UARTC4 控制寄存器 2	UC4CTL2			√		FFH
FFFFFA43H	UARTC4 选项控制寄存器 0	UC4OPT0		√	√		14H
FFFFFA44H	UARTC4 状态寄存器	UC4STR		√	√		00H
FFFFFA46H	UARTC4 接收数据寄存器	UC4RX	R			√	01FFH
FFFFFA46H	UARTC4 接收数据寄存器 L	UC4RXL			√		FFH
FFFFFA48H	UARTC4 发送数据寄存器	UC4TX	R/W			√	01FFH
FFFFFA48H	UARTC4 发送数据寄存器 L	UC4TXL			√		FFH
FFFFFA4AH	UARTC4 选项控制寄存器 1	UC4OPT1		√	√		00H
FFFFFA80H	TMM0 控制寄存器 0	TM0CTL0		√	√		00H
FFFFFA84H	TMM0 比较寄存器 0	TM0CMP0				√	0000H
FFFFFA90H	TMM1 控制寄存器 0	TM1CTL0		√	√		00H
FFFFFA94H	TMM1 比较寄存器 0	TM1CMP0				√	0000H
FFFFFAA0H	TMM2 控制寄存器 0	TM2CTL0		√	√		00H
FFFFFAA4H	TMM2 比较寄存器 0	TM2CMP0				√	0000H
FFFFFAB0H	TMM3 控制寄存器 0	TM3CTL0		√	√		00H
FFFFFAB4H	TMM3 比较寄存器 0	TM3CMP0				√	0000H
FFFFFAD0H	副计数寄存器	RC1SUBC	R			√	0000H
FFFFFAD2H	秒计数寄存器	RC1SEC	R/W		√		00H
FFFFFAD3H	分钟计数寄存器	RC1MIN			√		00H
FFFFFAD4H	小时计数寄存器	RC1HOUR			√		12H
FFFFFAD5H	星期计数寄存器	RC1WEEK			√		00H
FFFFFAD6H	日计数寄存器	RC1DAY			√		01H
FFFFFAD7H	月计数寄存器	RC1MONTH			√		01H
FFFFFAD8H	年计数寄存器	RC1YEAR			√		00H

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFAD9H	时间误差修正寄存器	RC1SUBU	R/W	√	√		00H
FFFFADAH	报警分钟设置寄存器	RC1ALM			√		00H
FFFFADBH	报警小时设置寄存器	RC1ALH			√		12H
FFFFADCH	报警星期设置寄存器	RC1ALW		√	√		00H
FFFFADDH	RTC 控制寄存器 0	RC1CC0		√	√		00H
FFFFADEH	RTC 控制寄存器 1	RC1CC1		√	√		00H
FFFFADFH	RTC 控制寄存器 2	RC1CC2		√	√		00H
FFFFAE0H	RTC 控制寄存器 3	RC1CC3		√	√		00H
FFFFC00H	外部中断下降沿指定寄存器 0	INTF0		√	√		00H
FFFFC04H	外部中断下降沿指定寄存器 2 ^{※1}	INTF2 ^{※1}		√	√		00H
FFFFC06H	外部中断下降沿指定寄存器 3	INTF3		√	√		00H
FFFFC08H	外部中断下降沿指定寄存器 4	INTF4		√	√		00H
FFFFC0AH	外部中断下降沿指定寄存器 5 ^{※2}	INTF5 ^{※2}		√	√		00H
FFFFC12H	外部中断下降沿指定寄存器 9	INTF9				√	0000H
FFFFC12H	外部中断下降沿指定寄存器 9H	INTF9H	√	√		00H	
FFFFC13H	外部中断下降沿指定寄存器 9L	INTF9L	√	√		00H	
FFFFC20H	外部中断上升沿指定寄存器 0	INTR0	√	√		00H	
FFFFC24H	外部中断上升沿指定寄存器 2 ^{※1}	INTR2 ^{※1}	√	√		00H	
FFFFC26H	外部中断上升沿指定寄存器 3	INTR3	√	√		00H	
FFFFC28H	外部中断上升沿指定寄存器 4	INTR4	√	√		00H	
FFFFC2AH	外部中断上升沿指定寄存器 5 ^{※2}	INTR5 ^{※2}	√	√		00H	
FFFFC32H	外部中断上升沿指定寄存器 9	INTR9			√	0000H	
FFFFC32H	外部中断上升沿指定寄存器 9H	INTR9H	√	√		00H	
FFFFC33H	外部中断上升沿指定寄存器 9L	INTR9L	√	√		00H	
FFFFC60H	端口 0 功能寄存器	PF0	√	√		00H	
FFFFC64H	端口 2 功能寄存器 ^{※1}	PF2 ^{※1}	√	√		00H	
FFFFC66H	端口 3 功能寄存器	PF3	√	√		00H	
FFFFC68H	端口 4 功能寄存器	PF4	√	√		00H	
FFFFC6AH	端口 5 功能寄存器	PF5	√	√		00H	
FFFFC72H	端口 9 功能寄存器	PF9			√	0000H	
FFFFC72H	端口 9 功能寄存器 L	PF9L	√	√		00H	
FFFFD00H	CSIF0 控制寄存器 0	CF0CTL0	√	√		01H	
FFFFD01H	CSIF0 控制寄存器 1	CF0CTL1	√	√		00H	
FFFFD02H	CSIF0 控制寄存器 2	CF0CTL2		√		00H	
FFFFD03H	CSIF0 状态寄存器	CF0STR	√	√		00H	
FFFFD04H	CSIF0 接收数据寄存器	CF0RX	R		√	0000H	
FFFFD04H	CSIF0 接收数据寄存器 L	CF0RXL			√	00H	
FFFFD06H	CSIF0 发送数据寄存器	CF0TX	R/W		√	0000H	
FFFFD06H	CSIF0 发送数据寄存器 L	CF0TXL			√	00H	
FFFFD10H	CSIF1 控制寄存器 0	CF1CTL0	√	√		01H	
FFFFD11H	CSIF1 控制寄存器 1	CF1CTL1	√	√		00H	
FFFFD12H	CSIF1 控制寄存器 2	CF1CTL2		√		00H	

注 1. 仅限 V850ES/JH3-U

2. 仅限 V850ES/JG3-U

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFD13H	CSIF1 状态 寄存器	CF1STR	R/W	√	√		00H
FFFFD14H	CSIF1 接收数据寄存器	CF1RX	R			√	0000H
FFFFD14H	CSIF1 接收数据寄存器 L	CF1RXL			√		00H
FFFFD16H	CSIF1 发送数据寄存器	CF1TX	R/W			√	0000H
FFFFD16H	CSIF1 发送数据寄存器 L	CF1TXL			√		00H
FFFFD20H	CSIF2 控制 寄存器 0	CF2CTL0		√	√		01H
FFFFD21H	CSIF2 控制 寄存器 1	CF2CTL1		√	√		00H
FFFFD22H	CSIF2 控制 寄存器 2	CF2CTL2			√		00H
FFFFD23H	CSIF2 状态 寄存器	CF2STR		√	√		00H
FFFFD24H	CSIF2 接收数据寄存器	CF2RX	R			√	0000H
FFFFD24H	CSIF2 接收数据寄存器 L	CF2RXL			√		00H
FFFFD26H	CSIF2 发送数据寄存器	CF2TX	R/W			√	0000H
FFFFD26H	CSIF2 发送数据寄存器 L	CF2TXL			√		00H
FFFFD30H	CSIF3 控制 寄存器 0	CF3CTL0		√	√		01H
FFFFD31H	CSIF3 控制 寄存器 1	CF3CTL1		√	√		00H
FFFFD32H	CSIF3 控制 寄存器 2	CF3CTL2			√		00H
FFFFD33H	CSIF3 状态 寄存器	CF3STR		√	√		00H
FFFFD34H	CSIF3 接收数据寄存器	CF3RX	R			√	0000H
FFFFD34H	CSIF3 接收数据寄存器 L	CF3RXL			√		00H
FFFFD36H	CSIF3 发送数据寄存器	CF3TX	R/W			√	0000H
FFFFD36H	CSIF3 发送数据寄存器 L	CF3TXL			√		00H
FFFFD40H	CSIF4 控制 寄存器 0	CF4CTL0		√	√		01H
FFFFD41H	CSIF4 控制 寄存器 1	CF4CTL1		√	√		00H
FFFFD42H	CSIF4 控制 寄存器 2	CF4CTL2			√		00H
FFFFD43H	CSIF4 状态 寄存器	CF4STR		√	√		00H
FFFFD44H	CSIF4 接收数据寄存器	CF4RX	R			√	0000H
FFFFD44H	CSIF4 接收数据寄存器 L	CF4RXL			√		00H
FFFFD46H	CSIF4 发送数据寄存器	CF4TX	R/W			√	0000H
FFFFD46H	CSIF4 发送数据寄存器 L	CF4TXL			√		00H
FFFFD80H	IIC 移位 寄存器 0	IIC0			√		00H
FFFFD82H	IIC 控制 寄存器 0	IICC0		√	√		00H
FFFFD83H	从机 地址 寄存器 0	SVA0			√		00H
FFFFD84H	IIC 时钟选择寄存器 0	IICCL0		√	√		00H
FFFFD85H	IIC 功能扩展寄存器 0	IICX0		√	√		00H
FFFFD86H	IIC 状态寄存器 0	IICS0	R	√	√		00H
FFFFD8AH	IIC 标志 寄存器 0	IICF0	R/W	√	√		00H
FFFFD90H	IIC 移位寄存器 1	IIC1				√	
FFFFD92H	IIC 控制寄存器 1	IICC1		√	√		00H
FFFFD93H	从机 地址寄存器 1	SVA1			√		00H
FFFFD94H	IIC 时钟选择寄存器 1	IICCL1		√	√		00H
FFFFD95H	IIC 功能扩展寄存器 1	IICX1		√	√		00H
FFFFD96H	IIC 状态寄存器 1	IICS1	R	√	√		00H
FFFFD9AH	IIC 标志寄存器 1	IICF1	R/W	√	√		00H

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFFFDA0H	IIC 移位寄存器 2	IIC2	R/W		√		00H
FFFFFFDA2H	IIC 控制寄存器 2	IICC2		√	√		00H
FFFFFFDA3H	从机地址寄存器 2	SVA2			√		00H
FFFFFFDA4H	IIC 时钟选择寄存器 2	IICCL2		√	√		00H
FFFFFFDA5H	IIC 功能扩展寄存器 2	IICX2		√	√		00H
FFFFFFDA6H	IIC 状态寄存器 2	IICS2	R	√	√		00H
FFFFFFDAAH	IIC 标志寄存器 2	IICF2	R/W	√	√		00H
FFFFFF40H	USB 时钟选择寄存器	UCKSEL		√	√		00H
FFFFFF41H	USB 功能控制寄存器	UFCKMSK		√	√		03H
FFFFFF42H	USB 功能选择寄存器	UHCKMSK		√	√		03H
FFFFFF60H	外部 DMA 请求使能寄存器	EXDRQEN			√		00H

3.4.7 专用寄存器

专用寄存器是受到保护的寄存器，可以防止程序循环引起的数据非法写入。V850ES/JG3-U 和 V850ES/JH3-U 具有以下八个专用寄存器。

- 节电控制寄存器 (PSC)
- 时钟控制寄存器 (CKC)
- 处理器时钟控制寄存器 (PCC)
- 时钟监控模式寄存器 (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) (仅限 V850ES/JG3-U)

此外，还提供 PRCDM 寄存器来防止对专用寄存器的非法写操作，这样应用系统就不会因为程序循环而意外停止工作。对专用寄存器的写操作需要特定流程实现，非法保存操作将记录在 SYS 寄存器中。

(1) 设置专用寄存器数据

按照以下流程设置专用寄存器数据。

- <1> 禁止 DMA 操作
- <2> 需要写入专用寄存器的数据放入通用寄存器中。
- <3> 将步骤<2>准备好的数据写入 PRCMD 寄存器中。
- <4> 将设置数据写入专用寄存器中（使用如下指令完成）。
 - 存储指令（ST/SST 指令）
 - 位操作指令（SET1/CLR1/NOT1 指令）。
- （<5> 至 <9> 插入 NOP 指令（5 条））^註
- <10> 如果需要，则使能 DMA 操作。

[实例] 使用 PSC 寄存器（设置待机模式）

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

读取专用寄存器时没有特定流程。

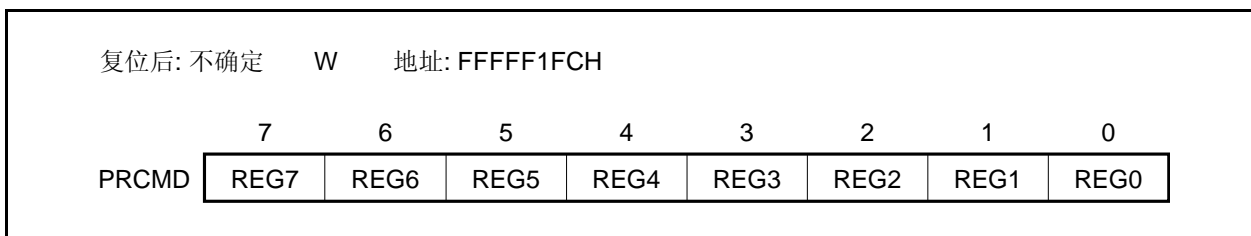
注 设置 IDLE1 模式、IDLE2 模式或 STOP 模式（通过将 PSC.STP 位置 1）后，要立即插入 5 条或更多的 NOP 指令。

- 注意事项**
1. 执行存储指令来对命令寄存器进行写操作时，不响应中断。原因是假设上述步骤<3>和<4>是通过连续的存储指令执行。若在步骤<3>和<4>之间插入了其他指令，且该指令又引起了中断响应，则将打乱上述的顺序，导致错误操作。
 2. 虽然写入 PRCMD 寄存器的是伪数据，仍然使用在设置专用寄存器（例中的步骤<4>）时所用的通用寄存器将数据写入至 PRCMD 寄存器（例中的步骤<3>）。当通用寄存器用于寻址时，同样使用这种用法。

(2) 命令寄存器 (PRCMD)

PRCMD 是 8 位寄存器，用于保护那些会严重影响应用系统的寄存器不被写入，从而使系统不会因为程序中止而意外停止工作。预先准备的数据写入 PRCMD 寄存器后，对专用寄存器的写操作第一步才是有效的。这样，专用寄存器的值仅能按照特定的流程进行写入，这样就可以防止对专用寄存器的非法写操作。

PRCMD 寄存器仅能按字节进行写入操作（若对其进行读取，则读到的数据不确定）。



(3) 系统状态寄存器 (SYS)

该寄存器包含了整个系统的操作状态的状态标志。

该寄存器可进行字节读写或者按位读写。

系统复位后，该寄存器的值被置为 00H。

复位后: 00H	R/W	地址: FFFFFFF802H						
	7	6	5	4	3	2	1	<0>
SYS	0	0	0	0	0	0	0	PRERR
	PRERR	检测保护错误						
	0	没有发生保护错误						
	1	发生保护错误						

PRERR 标志的操作是按下列条件进行的。

(a) 设置条件 (PRERR 标志=1)

- (i) 当数据写入专用寄存器时，却没有任何内容写入 PRCMD 寄存器，（没有执行 **3.4.7 (1) 设置数据至专用寄存器**的步骤 <3>，就执行步骤 <4>）
- (ii) 对 PRCMD 寄存器写入数据后，数据写入了片上周边 I/O 寄存器而没有写入专用寄存器（如果 **3.4.7 (1) 设置数据至专用寄存器**中的步骤<4>不是对专用寄存器的设置）

备注 即使在写入 PRCMD 寄存器与写入专用寄存器的操作之间，读取片上外围 I/O 寄存器（位操作指令除外），也不会设置 PRERR 标志，且设置数据可以写入专用寄存器。

(b) 清零条件 (PRERR 标志=0)

- (i) 当将 0 写入 PRERR 标志位时。
- (ii) 系统被复位时。

- 注意事项**
1. 如果在对 PRCMD 寄存器进行写访问之后，立即将 0 写入 SYS 寄存器（并非专用寄存器）的 PRERR 位，那么 PRERR 位将清为零（写访问优先）。
 2. 如果在对 PRCMD 寄存器进行写访问之后，立即将数据写入 PRCMD 寄存器（该寄存器并不是专用寄存器），那么 PRERR 位被设为 1。

3.4.8 注意事项

(1) 需要先行设置的寄存器

当使用 V850ES/JG3-U 和 V850ES/JH3-U 单片机时，必需确保首先对下列寄存器进行设置。

- 系统等待控制寄存器 (VSWC)
- 片上调试模式寄存器 (OCDM) (仅限 V850ES/JG3-U)
- 看门狗定时器模式寄存器 2 (WDTM2)

设置 VSWC、OCDM 和 WDTM2 寄存器后，再按照具体需求对其它寄存器进行设置。

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

(a) 系统等待控制寄存器 (VSWC)

VSWC 寄存器控制访问片上外围 I/O 寄存器时的总线等待时间。

访问片上外围 I/O 寄存器（没有等待周期）需要三个时钟。根据工作频率，V850ES/JG3-U 和 V850ES/JH3-U 需要不同的等待周期。按照下表设置不同工作频率下的 VSWC 寄存器值。

VSWC 寄存器按字节进行读取或写入。

系统复位后，该寄存器被置为 77H。

复位后: 77H R/W 地址: FFFFF06EH								
	7	6	5	4	3	2	1	0
VSWC								
工作频率 (f _{CPU})	VSWC 的设置值	等待周期						
f _{CPU} < 16.6 MHz	00H	0 (无等待)						
16.6 MHz ≤ f _{CPU} ≤ 25 MHz	01H	1						
25 MHz ≤ f _{CPU} ≤ 33.3 MHz	11H	2						
33.3 MHz ≤ f _{CPU} ≤ 48 MHz	12H	3						

(b) 片上调试模式寄存器 (OCDM)

详情参见第 32 章 片上调试功能。

(c) 看门狗定时器模式寄存器 2 (WDTM2)

WDTM2 寄存器用以设置看门狗定时器 2 的溢出时间和工作时钟。

复位解除后，在复位模式下看门狗定时器 2 会自动启动。通过写入 WDTM2 寄存器来激活该操作。

详情参见第 13 章 看门狗定时器 2 的功能。

(2) 访问指定的片上外设 I/O 寄存器

本产品有两种类型的内部系统总线。

一种是 CPU 总线，另外一种是用来与低速外设硬件接口的外设总线。

CPU 总线时钟与外设总线时钟是异步的。因此，当访问 CPU 与访问外设硬件发生冲突的时候，就可能会传输无法预料的非法数据。所以，如果访问外设硬件时有可能发生这种冲突，就需要改变访问 CPU 的时钟周期数量，以便正确传输数据。这样，CPU 就不会起动下一条指令的处理，而是进入等待状态。当这种等待状态发生时，执行一条指令所需的时钟数量就会增加下表列出的等待时钟长度。

当执行实时处理任务时，需要特别注意这种情况。

当访问指定的片上外设 I/O 寄存器时，除了由 VSWC 寄存器设置的等待状态之外，可能还需要更多的等待。

访问条件以及如何计算需要插入的等待状态数量（CPU 时钟数）如下表所示：

外设功能	寄存器名称	访问类型	k
16 位定时器/事件计数器 AA (TAA) (n = 0 至 5, m = 0 至 3, 5)	TAA nCNT	读取	1 或 2
	TAA nCCR0, TAA nCCR1	写入	• 第一次访问: 无等待 • 连续写入: 0 至 3
		读取	1 或 2
	TAA mIOC4	写入	• 第一次访问: 无等待 • 连续写入: 0 至 3
		读取	1 或 2
16 位定时器/事件计数器 AB (TAB) (n = 0, 1)	TAB nCNT	读取	1 或 2
	TAB nCCR0 至 TAB nCCR3	写入	• 第一次访问: 无等待 • 连续写入: 0 至 3
		读取	1 或 2
	TAB nIOC4	写入	• 第一次访问: 无等待 • 连续写入: 0 至 3
		读取	1 或 2
电机控制	TAB0OPT1	写入	• 第一次访问: 无等待 • 连续写入: 0 至 3
	TAB0DTC	写入	• 第一次访问: 无等待 • 连续写入: 0 至 3
TMT	TT0CNT	读取	1 或 2
	TT0TCR0, TT0TCR1	写入	• 第一次访问: 无等待 • 连续写入: 0 至 3
		读取	1 或 2
看门狗定时器 2 (WDT2)	WDTM2	写入 (当 WDT2 工作时)	3
实时输出功能 (RTO)	RTBL0, RTBH0	写入 (RTPC0.RTPOE0 位 = 0)	1
A/D 转换器	ADA0M0	读取	1 或 2
	ADA0CR0 至 ADA0CR11	读取	1 或 2
	ADA0CR0H 至 ADA0CR11H	读取	1 或 2
I ² C00 至 I ² C02	IICS0 至 IICS2	读取	1
CRC	CRCD	写入	1

访问时必需的时钟数目 = $3 + i + j + (2 + j) \times k$

注意事项 下列状态下禁止访问上述寄存器。如果产生等待周期，只能通过复位清除。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
- 当 CPU 运行于内部振荡时钟时。

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

(3) sld 指令和中断请求发生冲突的限制

(a) 描述

如果在指令<1>执行完毕之前，指令<2>的译码操作和某个中断请求产生冲突，那么，指令<1>的执行结果就可能无法储存到寄存器中。指令<2>的位置紧邻在 sld 指令之前，且位于指令<1>之后。

指令 <1>

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

指令<2>

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

<举例>

<i> ld.w [r11], r10 如果在 ld 指令<i>执行完毕之前，mov 指令<ii>的译码操作和某个中断请求产生冲突，那么，指令<i>的执行结果就可能无法储存到寄存器中。mov 指令<ii>紧邻在 sld 指令<iii>之前。

 ·

 ·

<ii> mov r10, r28

<iii> sld.w 0x28, r10

(b) 对策

- <1> 当使用编译器（CA850）时
请使用 CA850 Ver. 2.61 或更高版本，因为它将自动禁止相应指令序列的产生。
- <2> 当使用汇编器时
在指令<ii>之后就立即执行 sld 指令，可以采用以下任一种方法来避免上述操作。

- 紧贴在 sld 指令之前插入 nop 指令。
- 在紧邻 sld 指令之前的指令<ii>中，不要使用同一个寄存器作为 sld 指令目标寄存器。

第四章 端口功能

4.1 特性

- I/O 端口: 84
 - V850ES/JG3-U: 75
 - 5 V 耐压/N 沟道漏极开路输出可选: 22
 - V850ES/JH3-U: 96
 - 5 V 耐压/N 沟道漏极开路输出可选: 27
- 可按位指定输入/输出

4.2 端口基本配置

V850ES/JG3-U 总共包括 75 个输入/输出端口, 由端口 0, 1, 3 至 7, 9, CM, CT 以及 DL 组成。

V850ES/JH3-U 总共包括 96 个输入/输出端口, 由端口 0 至 7, 9, CM, CS, CT, DH 以及 DL 组成。

端口配置如下所示:

表 4-1. 引脚的 I/O 缓冲器电源 (V850ES/JG3-U)

电源	对应引脚
AV _{REF0}	端口 7
AV _{REF1}	端口 1
EV _{DD}	$\overline{\text{RESET}}$, 端口 0, 3 至 6, 9, CM, CT, DL

表 4-2. 引脚的 I/O 缓冲器电源 (V850ES/JH3-U)

电源	对应引脚
AV _{REF0}	端口 7
AV _{REF1}	端口 1
EV _{DD}	$\overline{\text{RESET}}$, 端口 0, 2 至 6, 9, CM, CS, CT, DH, DL

图 4-1. 端口配置图 (V850ES/JG3-U)

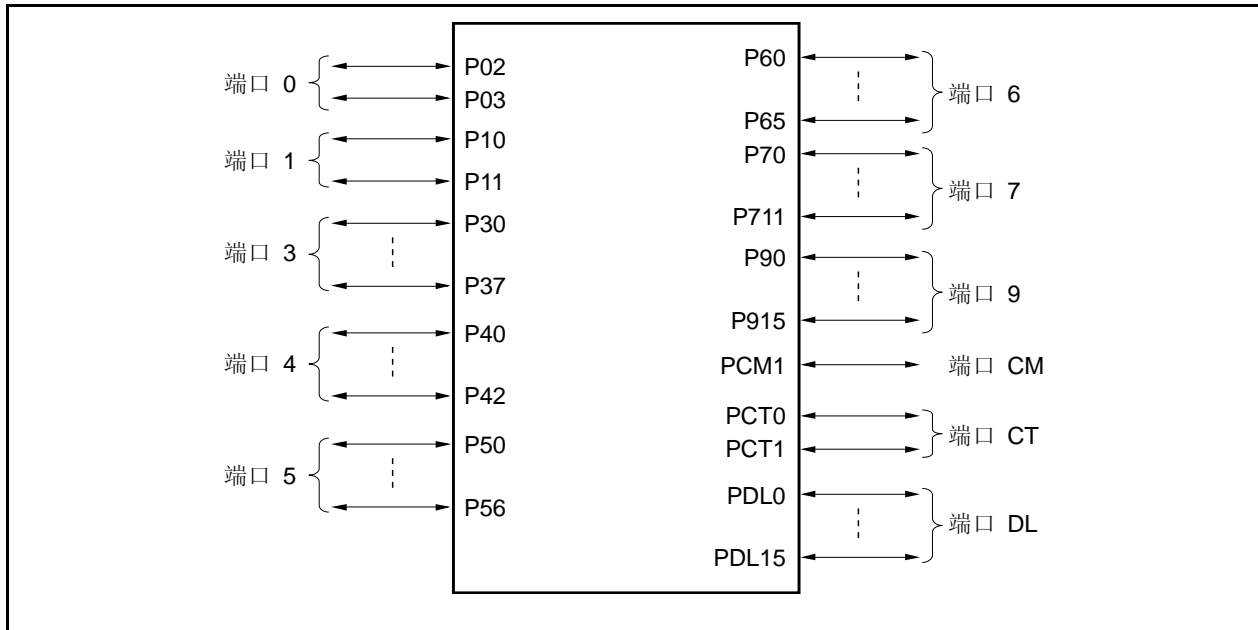
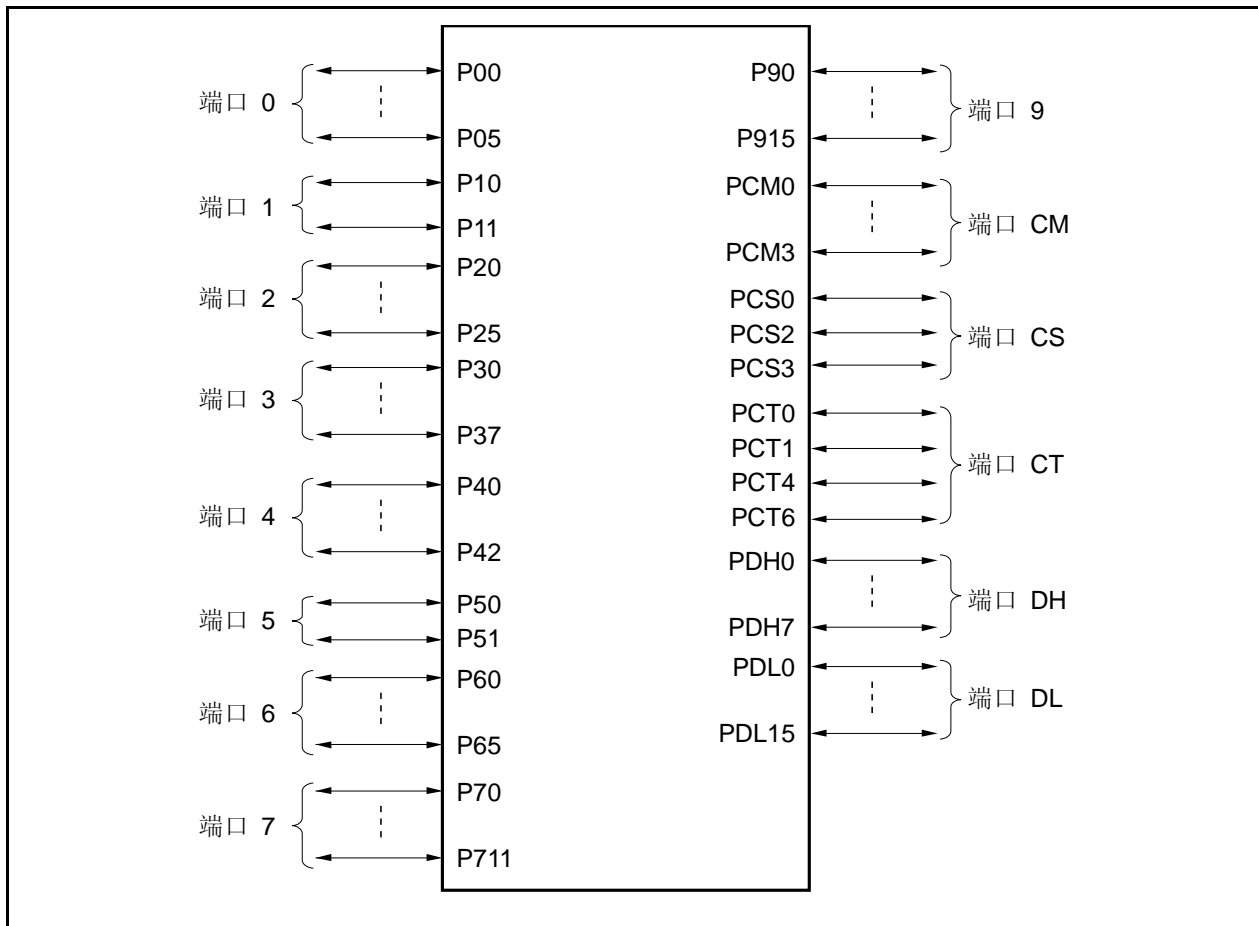


图 4-2. 端口配置图 (V850ES/JH3-U)



4.3 端口配置

表 4-3. 端口配置 (V850ES/JG3-U)

项目	配置
控制寄存器	端口 n 模式寄存器 (PMn: n = 0, 1, 3 至 7, 9, CM, CT, DL) 端口 n 模式控制寄存器 (PMcn: n = 0, 3 至 5, 9, CM, CT, DL) 端口 n 功能控制寄存器 (PFCn: n = 0, 3 至 6, 9) 端口 n 功能控制扩展寄存器 (PFCEn: n = 4 至 6, 9) 端口 n 功能寄存器 (PFn: n = 0, 3 至 5, 9)
端口	I/O: 75

表 4-4. 端口配置 (V850ES/JH3-U)

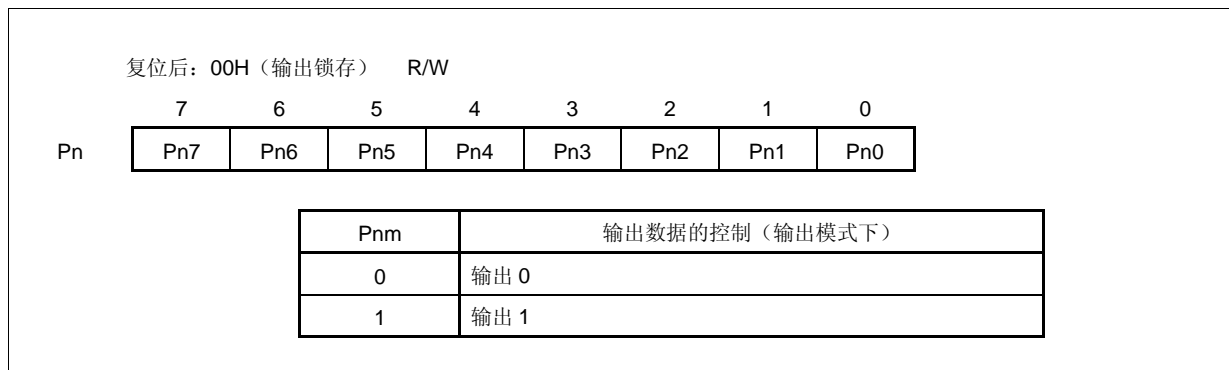
项目	配置
控制寄存器	端口 n 模式寄存器 (PMn: n = 0 至 7, 9, CM, CS, CT, DH, DL) 端口 n 模式控制寄存器 (PMcn: n = 0, 2 至 6, 9, CM, CS, CT, DH, DL) 端口 n 功能控制寄存器 (PFCn: n = 0, 2 至 6, 9) 端口 n 功能控制扩展寄存器 (PFCEn: n = 4 至 6, 9) 端口 n 功能寄存器 (PFn: n = 0, 2 至 5, 9)
端口	I/O: 96

(1) 端口 n 寄存器 (Pn)

通过写入或读取 Pn 寄存器，从外部器件输入数据，或输出数据到外部器件。

Pn 寄存器由一个保存输出数据的端口锁存器和一个读取引脚状态的电路组成。

Pn 寄存器的每一位对应端口的一个引脚，而且可以按位读出或写入。



无论 PMCn 寄存器如何设置，对 Pn 寄存器的数据写入或读取如下所示：

表 4-5. Pn 寄存器的写入/读出

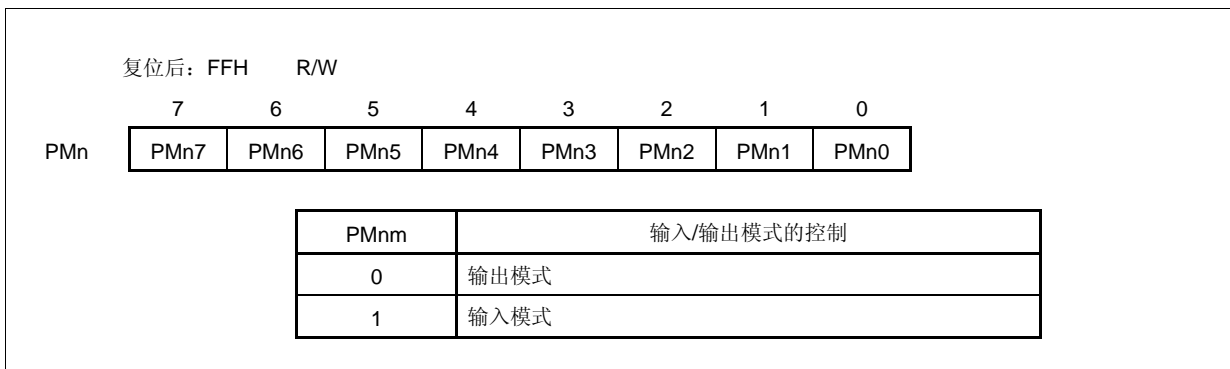
寄存器 PMn 的设置	写入 Pn 寄存器	读取 Pn 寄存器
输出模式 (PMnm 位 = 0)	数据写入输出锁存器 ^注 。 端口模式下 (PMc _n = 0)，输出锁存器的内容从相应引脚输出。	读取输出锁存器的值。
输入模式 (PMnm = 1)	数据写入输出锁存器。 引脚状态不受影响 ^注 。	读取引脚状态。

注 写入到输出锁存器的值会一直保留，直到有新的数值写入输出锁存器。

(2) 端口 n 模式寄存器 (PMn)

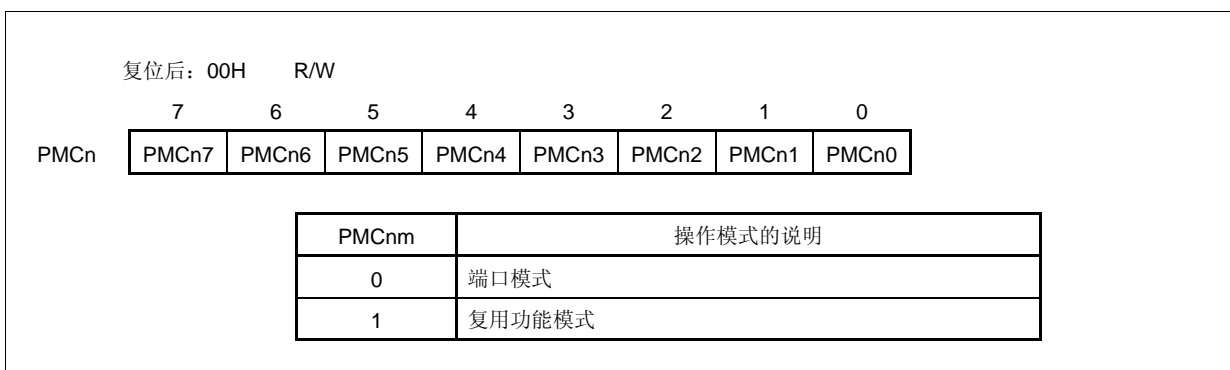
PMn 寄存器用于指定对应端口引脚的输入或输出模式。

该寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定为输入或输出模式。

**(3) 端口 n 模式控制寄存器 (PMcn)**

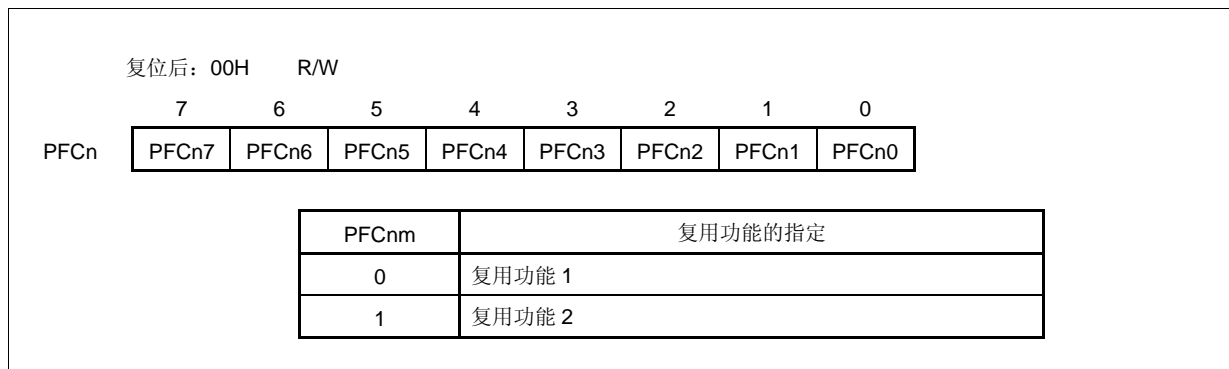
PMcn 寄存器指定端口模式或复用功能模式。

该寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定端口的工作模式。

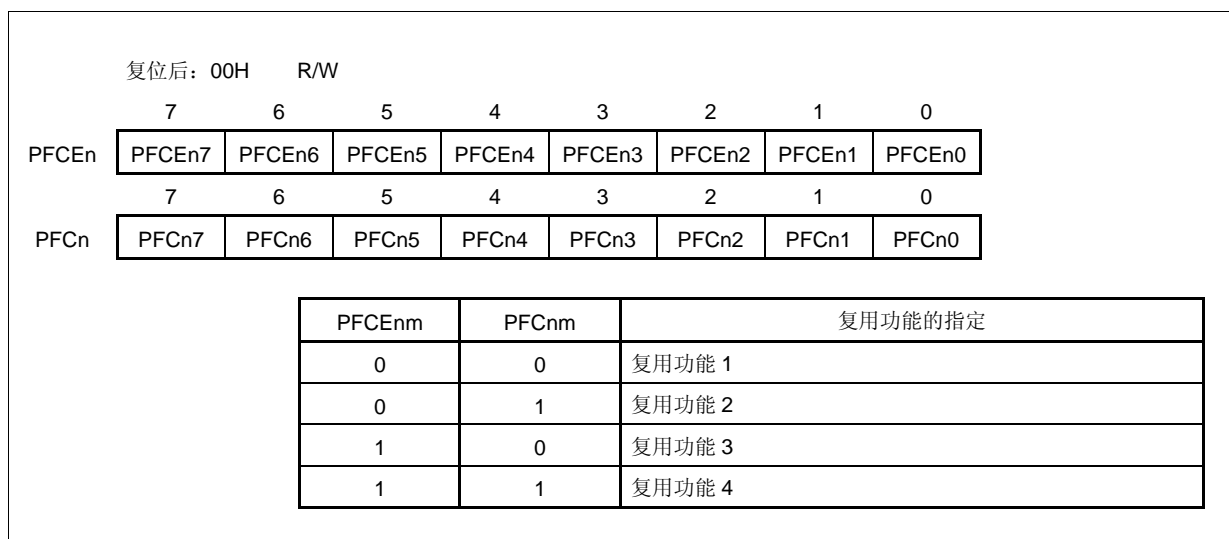


(4) 端口 n 功能控制寄存器 (PFCn)

如果引脚有两个复用功能，PFCn 寄存器用来指定需要使用的引脚复用功能。
该寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定端口引脚的复用功能。

**(5) 端口 n 功能控制扩展寄存器 (PFCEn)**

如果引脚有三个或更多的复用功能，PFCEn 寄存器用来指定需要使用的引脚复用功能。
该寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定端口引脚的复用功能。



(6) 端口 n 功能寄存器 (PFn)

PFn 寄存器指定端口是正常输出还是 N 沟道漏极开路输出。

该寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定端口引脚的输出模式。

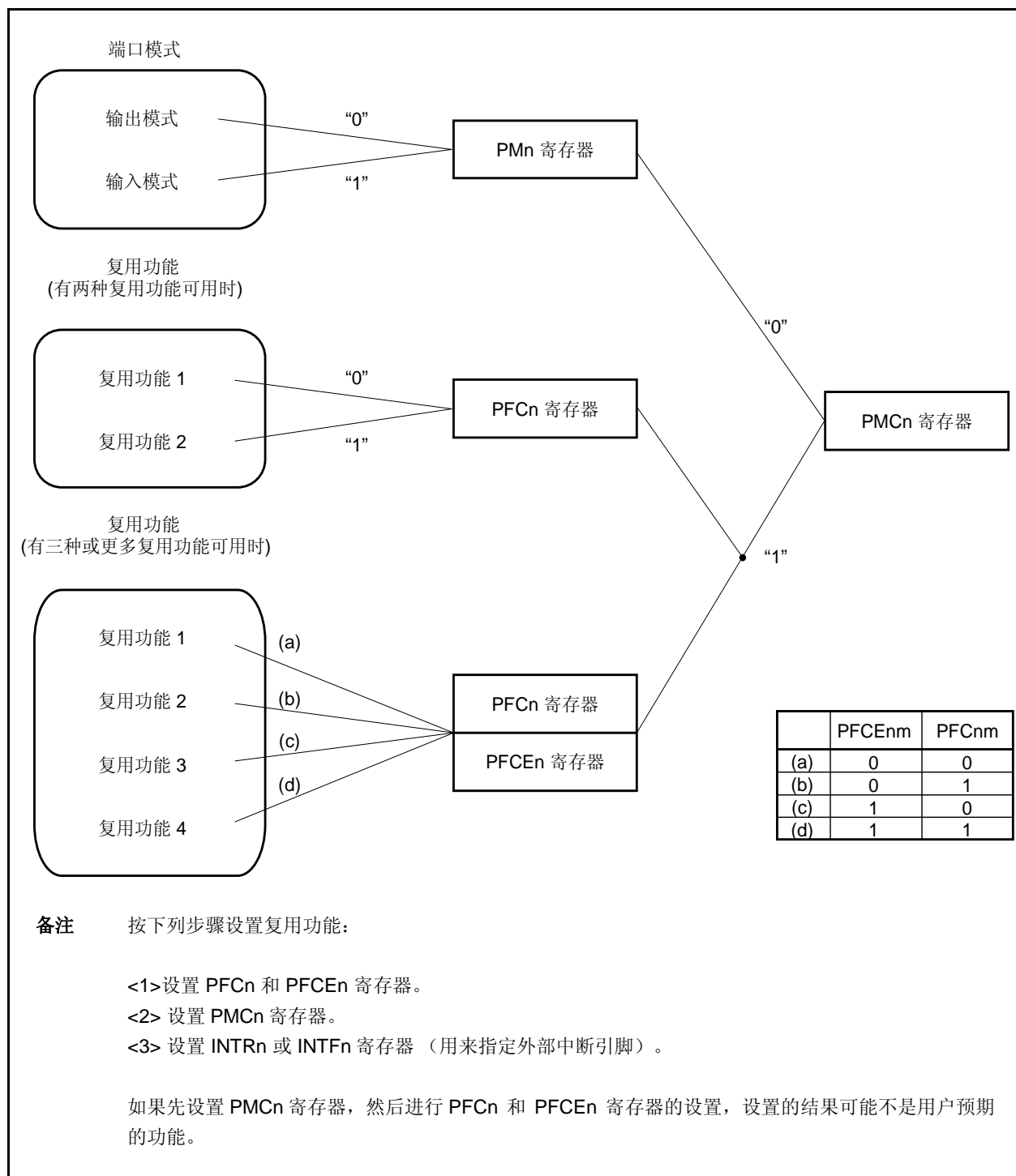
复位后: 00H		R/W						
	7	6	5	4	3	2	1	0
PFn	PFn7	PFn6	PFn5	PFn4	PFn3	PFn2	PFn1	PFn0
	PFnm [#]		正常输出 / N沟道漏极开路输出的控制					
	0		正常输出 (CMOS 输出)					
	1		N 沟道漏极开路输出					

注 在端口模式下 (PM_{Cnm} 位=0)，只有 PM_n 寄存器的 PM_{nm} 位为 0 时 (指定为输出模式时)，PFn 寄存器的 PFn_m 位才有效。当 PM_{nm} 位为 1 (指定为输入模式时)，PFn 寄存器设定值无效。

(7) 端口设置

端口设置如下所示：

图 4-3. 各个寄存器的设置和引脚功能



4.3.1 端口 0

端口 0 有 2 位(V850ES/JG3-U) /6 位(V850ES/JH3-U)输入输出端口，可以按位控制输入输出设置。

端口 0 包括以下复用功能引脚：

表 4-6. 端口 0 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
P00	-	8	INTP00	输入	可选作 N 沟道漏极开路输出
P01	-	9	INTP01	输入	
P02	6	6	NMI	输入	
P03	7	7	INTP02/ADTRG/UCLK	输入	
P04	-	26	INTP03	输入	
P05	-	27	INTP04	输入	

注意事项 在复用功能引脚的输入模式下，P02 至 P05 具有迟滞特性，但端口模式没有迟滞特性。

(1) 端口 0 寄存器 (P0)

(a) V850ES/JG3-U

复位后: 00H (输出锁存) R/W 地址: FFFFF400H

	7	6	5	4	3	2	1	0
P0	0	0	0	0	P03	P02	0	0

P0n	输出数据控制 (输出模式下) (n = 2, 3)
0	输出 0
1	输出 1

(b) V850ES/JH3-U

复位后: 00H (输出锁存) R/W 地址: FFFFF400H

	7	6	5	4	3	2	1	0
P0	0	0	P05	P04	P03	P02	P01	P00

P0n	输出数据控制 (输出模式下) (n = 0 至 5)
0	输出 0
1	输出 1

(2) 端口 0 模式寄存器 (PM0)

(a) V850ES/JG3-U

复位后: FFH R/W 地址: FFFFF420H

	7	6	5	4	3	2	1	0
PM0	1	1	1	1	PM03	PM02	1	1

PM0n	输入/输出模式控制 (n = 2, 3)
0	输出模式
1	输入模式

(b) V850ES/JH3-U

复位后: FFH R/W 地址: FFFFF420H

	7	6	5	4	3	2	1	0
PM0	1	1	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	输入/输出模式控制 (n = 0至5)
0	输出模式
1	输入模式

(3) 端口 0 模式控制寄存器 (PMC0)

(1/2)

(a) V850ES/JG3-U

复位后: 00H R/W 地址: FFFFF440H

	7	6	5	4	3	2	1	0
PMC0	0	0	0	0	PMC03	PMC02	0	0

PMC03	P03引脚操作模式的指定
0	I/O 端口
1	INTP02 输入/ADTRG 输入/UCLK 输入

PMC02	P02 引脚操作模式的指定
0	I/O 端口
1	NMI 输入

(b) V850ES/JH3-U

复位后: 00H R/W 地址: FFFFF440H

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

PMC05	P05引脚操作模式的指定	
0	I/O 端口	
1	INTP04 输入	
PMC04	P04 引脚操作模式的指定	
0	I/O 端口	
1	INTP03 输入	
PMC03	P03 引脚操作模式的指定	
0	I/O 端口	
1	INTP02 输入/ADTRG 输入/UCLK 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	INTP01 输入	
PMC00	P00 引脚操作模式的指定	
0	I/O 端口	
1	INTP00 输入	

(4) 端口 0 功能控制寄存器 (PFC0)

复位后: 00H R/W 地址: FFFFF460H

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

备注 复用功能说明的详情, 参见 **4.3.1 (6) 端口 0 复用功能说明**。

(5) 端口 0 功能控制扩展寄存器 (PFCE0)

复位后: 00H	R/W	地址: FFFFF700H								
			7	6	5	4	3	2	1	0
PFCE0			0	0	0	0	PFCE03	0	0	0
备注	复用功能说明的详情, 参见 4.3.1 (6) 端口 0 复用功能说明。									

(6) 端口 0 复用功能指定

PFCE03	PFC03	P03 引脚复用功能的指定
0	0	INTP02 输入
0	1	ADTRG 输入
1	0	UCLK 输入
1	1	禁止设置

(7) 端口 0 功能寄存器 (PF0)

(a) V850ES/JG3-U										
复位后: 00H	R/W	地址: FFFFC60H								
			7	6	5	4	3	2	1	0
PF0			0	0	0	0	PF03	PF02	0	0
PF0n	正常输出 / N沟道漏极开路输出的控制 (n = 2, 3)									
0	正常输出									
1	N 沟道漏极开路输出									
(b) V850ES/JH3-U										
复位后: 00H	R/W	地址: FFFFC60H								
			7	6	5	4	3	2	1	0
PF0			0	0	PF05	PF04	PF03	PF02	PF01	PF00
PF0n	正常输出 / N沟道漏极开路输出的控制 (n = 0至5)									
0	正常输出									
1	N 沟道漏极开路输出									

4.3.2 端口 1

端口 1 有 2 位输入输出端口，可以按位控制输入输出设置。

端口 1 包括以下复用功能引脚：

表 4-7. 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	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
P10	3	3	AN00	输出	-
P11	4	4	AN01	输出	

注意事项 电源打开后的复位期间，P10 和 P11 引脚可能会暂时输出不确定电平。

(1) 端口 1 寄存器 (P1)

复位后：00H (输出锁存) R/W 地址：FFFF402H

	7	6	5	4	3	2	1	0
P1	0	0	0	0	0	0	P11	P10

P1n	输出数据控制 (输出模式下) (n = 0, 1)
0	输出 0
1	输出 1

注意事项 D/A 转换期间，不要读取或写入 P1 寄存器 (参见 16.4.3 注意事项)。

(2) 端口 1 模式寄存器 (PM1)

复位后：FFH R/W 地址：FFFF422H

	7	6	5	4	3	2	1	0
PM1	1	1	1	1	1	1	PM11	PM10

PM1n	输入/输出模式控制 (n = 0, 1)
0	输出模式
1	输入模式

- 注意事项**
1. P1n 用作复用功能 (ANOn 引脚输出) 时，将 PM1n 位设置为 1。
 2. 当使用 PM10 和 PM11 引脚之一作为 I/O 端口，而另一个用作 D/A 输出引脚。在 D/A 输出期间，如果端口 I/O 电平不改变，则在应用系统中可以这样使用。

4.3.3 端口 2（仅限V850ES/JH3-U）

端口 2 有 6 位输入输出端口，可以按位控制输入输出设置。

端口 2 包括以下复用功能引脚：

表 4-8. 端口 2 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
P20	-	32	TIAB03/KR2/TOAB03/RTP02	输入/输出	可选作 N 沟道漏极开路输出
P21	-	33	SIF2/KR3/TIAB00/TOAB00 /RTP03	输入/输出	
P22	-	34	SOF2/KR4/RTP04	输入/输出	
P23	-	35	SCKF2/KR5/RTP05	输入/输出	
P24	-	36	INTP05	输入	
P25	-	28	INTP06	输入	

注意事项 在复用功能引脚的输入模式下，P20 至 P25 具有迟滞特性，但端口模式没有迟滞特性。

(1) 端口 2 寄存器 (P2)

复位后: 00H (输出锁存) R/W 地址: FFFFF404H														
	7	6	5	4	3	2	1	0						
P2	0	0	P25	P24	P23	P22	P21	P20						
	<table border="1"> <thead> <tr> <th>P2n</th> <th>输出数据控制 (输出模式下) (n = 0至5)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>输出 0</td> </tr> <tr> <td>1</td> <td>输出 1</td> </tr> </tbody> </table>								P2n	输出数据控制 (输出模式下) (n = 0至5)	0	输出 0	1	输出 1
P2n	输出数据控制 (输出模式下) (n = 0至5)													
0	输出 0													
1	输出 1													

(2) 端口 2 模式寄存器 (PM2)

复位后: FFH R/W 地址: FFFFF424H														
	7	6	5	4	3	2	1	0						
PM2	1	1	PM25	PM24	PM23	PM22	PM21	PM20						
	<table border="1"> <thead> <tr> <th>PM2n</th> <th>输入/输出模式控制 (n = 0至5)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>输出模式</td> </tr> <tr> <td>1</td> <td>输入模式</td> </tr> </tbody> </table>								PM2n	输入/输出模式控制 (n = 0至5)	0	输出模式	1	输入模式
PM2n	输入/输出模式控制 (n = 0至5)													
0	输出模式													
1	输入模式													

(3) 端口 2 模式控制寄存器 (PMC2)

复位后: 00H R/W 地址: FFFF444H

	7	6	5	4	3	2	1	0
PMC2	0	0	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20

PMC25	P25引脚操作模式的指定	
0	I/O 端口	
1	INTP06 输入	
PMC24	P24 引脚操作模式的指定	
0	I/O 端口	
1	INTP05 输入	
PMC23	P23 引脚操作模式的指定	
0	I/O 端口	
1	SCKF2 输入/输出/ KR5 输入/RTP05 输出	
PMC22	P22 引脚操作模式的指定	
0	I/O 端口	
1	SOF2 输出/KR4 输入/RTP04 输出	
PMC21	P21 引脚操作模式的指定	
0	I/O 端口	
1	SIF2 输出/KR3 输入/TIAB00 输入/TOAB00 输出/RTP03 输出	
PMC20	P20 引脚操作模式的指定	
0	I/O 端口	
1	TIAB03 输入/KR2 输入/TOAB03 输出/RTP02 输出	

(4) 端口 2 功能控制寄存器 (PFC2)

复位后: 00H R/W 地址: FFFF464H

	7	6	5	4	3	2	1	0
PFC2	0	0	0	0	PFC23	PFC22	PFC21	PFC20

备注 复用功能说明的详情, 参见 4.3.3 (6) 端口 2 复用功能说明。

(5) 端口 2 功能控制扩展寄存器 (PFCE2)

复位后: 00H	R/W	地址: FFFFF704H								
			7	6	5	4	3	2	1	0
PFCE2			0	0	0	0	PFCE23	PFCE22	PFCE21	PFCE20
备注	复用功能说明的详情, 参见 4.3.3 (6) 端口 2 复用功能说明。									

(6) 端口 2 复用功能指定

PFCE23	PFC23	P23 引脚复用功能的指定
0	0	SCKF2 I/O
0	1	KR5 输入
1	0	RTP05 输出
1	1	禁止设置

PFCE22	PFC22	P22 引脚复用功能的指定
0	0	SOF2 输出
0	1	KR4 输入
1	0	RTP04 输出
1	1	禁止设置

PFCE21	PFC21	P21 引脚复用功能的指定
0	0	SIF2 输入
0	1	KR3 输入/TIAB00 输入 ^注
1	0	TOAB00 输出
1	1	RTP03 输出

注 KR3 和 TIAB00 是复用功能。当引脚用作 TIAB00 时, 禁止复用功能 KR3 的引脚按键返回检测 (清除 KRM.KRM3 位为 0)。同样, 当引脚用作 KR3 时, 禁止复用功能 TIAB00 的引脚边沿检测 (TAB0IOC1.TAB0TIG0 和 TAB0TIG1 位 = 00B, TAB0IOC2 寄存器 = 00H)。

PFCE20	PFC20	P20 引脚复用功能的指定
0	0	TIAB03 输入
0	1	KR2 输入
1	0	TOAB03 输出
1	1	RTP02 输出

(7) 端口 2 功能寄存器 (PF2)

复位后: 00H R/W 地址: FFFFC64H

	7	6	5	4	3	2	1	0
PF2	0	PF26	PF25	PF24	PF23	PF22	PF21	PF20

PF2n	正常输出 / N沟道漏极开路输出的控制 (n = 0至6)
0	正常输出
1	N 沟道漏极开路输出

4.3.4 端口 3

端口 3 有 8 位输入输出端口，可以按位控制输入输出设置。

端口 3 包括以下复用功能引脚：

表 4-9. 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	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
P30	25	37	TXDC0/SOF4/INTP07	输入/输出	可选作 N 沟道漏极开路输出
P31	26	38	RXDC0/SIF4/INTP08	输入	
P32	27	39	ASCKC0/SCKF4/TIAA00/TOAA00	输入/输出	
P33	28	40	TIAA01/TOAA01/RTCDIV/RTCCL	输入/输出	
P34	29	41	TIAA10/TOAA10/TOAA1OFF/INTP09 /PPON	输入/输出	
P35	30	42	TIAA11/TOAA11/RTC1HZ/OCI	输入/输出	
P36	31	43	TXDC3/SCL00/UDMARQ0	输入/输出	
P37	32	44	RXDC3/SDA00/UDMAAK0	输入/输出	

注意事项 在复用功能引脚的输入模式下，P30 至 P37 具有迟滞特性，但端口模式没有迟滞特性。

(1) 端口 3 寄存器 (P3)

复位后: 00H (输出锁存) R/W 地址: FFFFF406H								
	7	6	5	4	3	2	1	0
P3	P37	P36	P35	P34	P33	P32	P31	P30
P3n	输出数据控制 (输出模式下) (n = 0至7)							
0	输出 0							
1	输出 1							

(2) 端口 3 模式寄存器 (PM3)

复位后: FFH R/W 地址: FFFFF426H								
	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30
PM3n	输入/输出模式控制 (n = 0至7)							
0	输出模式							
1	输入模式							

(3) 端口 3 模式控制寄存器 (PMC3)

复位后: 00H R/W 地址: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

PMC37	P37引脚操作模式的指定
0	I/O 端口
1	RXDC3 输入/SDA00 I/O/UDMAAK0 输出
PMC36	P36引脚操作模式的指定
0	I/O 端口
1	TXDC3 输出/SCL00 I/O/UDMARQ0 输入
PMC35	P35引脚操作模式的指定
0	I/O 端口
1	TIAA11 输入/TOAA11 输出/RTC1HZ 输出/OCI 输入
PMC34	P34 引脚操作模式的指定
0	I/O 端口
1	TIAA10 输入/TOAA10 输出/TOAA1OFF 输入/INTP09 输入/PPON 输出
PMC33	P33 引脚操作模式的指定
0	I/O 端口
1	TIAA01 输入/TOAA01 输出/RTCDIV 输出/RTCCL 输出
PMC32	P32 引脚操作模式的指定
0	I/O 端口
1	ASCKA0 输入/SCKF4 I/O/TIAA00 输入/TOAA00 输出
PMC31	P31 引脚操作模式的指定
0	I/O 端口
1	RXDC0 输入/SIF4 输入/INTP08 输入
PMC30	P30 引脚操作模式的指定
0	I/O 端口
1	TXDC0 输出/SOF4 输出/INTP07 输入

(4) 端口 3 功能控制寄存器 (PFC3)

复位后: 00H	R/W	地址: FFFFF466H								
			7	6	5	4	3	2	1	0
PFC3			PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30
备注	复用功能说明的详情, 参见 4.3.4 (6) 端口 3 复用功能说明。									

(5) 端口 3 功能控制扩展寄存器 L (PFCE3L)

复位后: 00H	R/W	地址: FFFFF706H								
			7	6	5	4	3	2	1	0
PFCE3			PFCE37	PFCE36	PFCE35	PFCE34	PFCE33	PFCE32	PFCE31	PFCE30
备注	复用功能说明的详情, 参见 4.3.4 (6) 端口 3 复用功能说明。									

(6) 端口 3 复用功能规范

PFCE37	PFC37	P37 引脚复用功能的指定
0	0	RXDC3 输入
0	1	SDA00 I/O
1	0	禁止设置
1	1	UDMAK0 输出

PFCE36	PFC36	P36 引脚复用功能的指定
0	0	TXDC3 输出
0	1	SCL00 I/O
1	0	禁止设置
1	1	UDMARQ0 输入

PFCE35	PFC35	P35 引脚复用功能的指定
0	0	TIAA11 输入
0	1	TOAA11 输出
1	0	RTC1HZ 输出
1	1	\overline{OCI} 输入

PFCE34	PFC34	P34 引脚复用功能的指定
0	0	TIAA10 输入
0	1	TOAA10 输出
1	0	TOAA1OFF 输入/INTP09 输入 [※]
1	1	PPON 输出

注 TOAA1OFF 和 INTP09 是复用功能。当引脚用作 TOAA1OFF 时，禁止复用功能 INTP09 的引脚边沿检测。同样，当引脚用作 INTP09 时，停止高阻抗输出控制器。

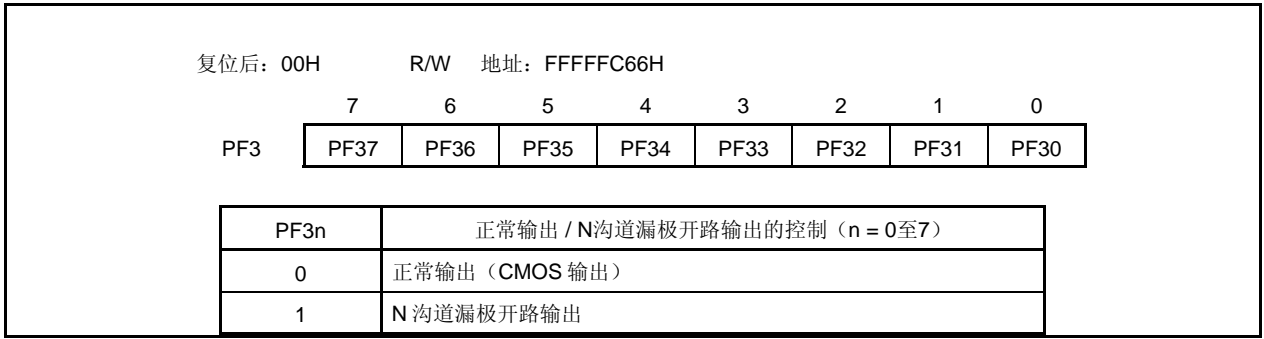
PFCE33	PFC33	P33 引脚复用功能的指定
0	0	TIAA01 输入
0	1	TOAA01 输出
1	0	RTCDIV 输出
1	1	RTCCL 输出

PFCE32	PFC32	P32 引脚复用功能的指定
0	0	ASCKC0 输入
0	1	SCKF4 I/O
1	0	TIAA00 输入
1	1	TOAA00 输出

PFCE31	PFC31	P31 引脚复用功能的指定
0	0	RXDC0 输入
0	1	SIF4 输入
1	0	INTP08 输入
1	1	禁止设置

PFCE30	PFC30	P30 引脚复用功能的指定
0	0	TXDC0 输出
0	1	SOF4 输出
1	0	INTP07 输入
1	1	禁止设置

(7) 端口 3 功能寄存器 (PF3)



4.3.5 端口 4

端口 4 有 3 位输入输出端口，可以按位控制输入输出设置。

端口 4 包括以下复用功能引脚：

表 4-10. 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	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
P40	22	29	SIF0/TXDC4/SDA01	输入/输出	可选作 N 沟道漏极开路输出
P41	23	30	SOF0/RXDC4/SCL01	输入/输出	
P42	24	31	SCKF0/INTP10	输入/输出	

注意事项 在复用功能引脚的输入模式下，P40 至 P42 具有迟滞特性，但端口模式下没有迟滞特性。

(1) 端口 4 寄存器 (P4)

复位后：00H（输出锁存） R/W 地址：FFFF408H

	7	6	5	4	3	2	1	0
P4	0	0	0	0	0	P42	P41	P40

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

(2) 端口 4 模式寄存器 (PM4)

复位后：FFH R/W 地址：FFFF428H

	7	6	5	4	3	2	1	0
PM4	1	1	1	1	1	PM42	PM41	PM40

PM4n	输入/输出模式控制（n = 0至2）
0	输出模式
1	输入模式

(3) 端口 4 模式控制寄存器 (PMC4)

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	SCKF0 I/O/INTP10 输入									
PMC41	P41引脚操作模式的指定									
0	I/O 端口									
1	SOF0 输出/RXDC4 输入/SCL01 I/O									
PMC40	P40引脚操作模式的指定									
0	I/O 端口									
1	SIF0 输入/TXDC4 输出/SDA01 I/O									

(4) 端口 4 功能控制寄存器 (PFC4)

复位后: 00H	R/W	地址: FFFFF468H								
			7	6	5	4	3	2	1	0
PFC4			0	0	0	0	0	PFC42	PFC41	PFC40
备注	复用功能说明的详情, 参见 4.3.5 (6) 端口 4 复用功能说明。									

(5) 端口 4 功能控制扩展寄存器 (PFCE4)

复位后: 00H	R/W	地址: FFFFF708H								
			7	6	5	4	3	2	1	0
PFCE4			0	0	0	0	0	0	PFCE41	PFCE40
备注	复用功能说明的详情, 参见 4.3.5 (6) 端口 4 复用功能说明。									

(6) 端口 4 复用功能指定

PFC42	P42引脚复用功能的指定
0	SCKF0 I/O
1	INTP10 输入

PFCE41	PFC41	P41 引脚复用功能的指定
0	0	SOF0 输出
0	1	RXDC4 输入
1	0	SCL01 I/O
1	1	禁止设置

PFCE40	PFC40	P40 引脚复用功能的指定
0	0	SIF0 输入
0	1	TXDC4 输出
1	0	SDA01 I/O
1	1	禁止设置

(7) 端口 4 功能寄存器 (PF4)

复位后: 00H R/W 地址: FFFFC68H

	7	6	5	4	3	2	1	0
PF4	0	0	0	0	0	PF42	PF41	PF40

PF4n	正常输出 / N沟道漏极开路输出的控制 (n = 0至2)
0	正常输出 (CMOS 输出)
1	N 沟道漏极开路输出

4.3.6 端口 5

端口 0 有 6 位(V850ES/JG3-U) /2 位(V850ES/JH3-U)输入输出端口，可以按位控制输入输出设置。

端口 5 包括以下复用功能引脚：

表 4-11. 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	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
P50	35	47	TIAB01/KR0/TOAB01/RTP00 /UDMARQ1	输入/输出	可选作 N 沟道漏极开路输出
P51	36	48	TIAB02/KR1/TOAB02/RTP01 /UDMAAK1	输入/输出	
P52	37	–	TIAB03/KR2/TOAB13/RTP02 /DDI [‡]	输入/输出	
P53	38	–	SIF2/TIAB00/KR3/TOAB10 /RTP03/DDO [‡]	输入/输出	
P54	39	–	SOF2/KR4/RTP04/DCK [‡]	输入/输出	
P55	40	–	SCKF2/KR5/RTP05/DMS [‡]	输入/输出	
P56	41	–	INTP05/DRST [‡]	输入	

注 DDI, DDO, DCK 和 DMS 和 DRST 引脚用于片上调试。

如果不使用片上调试功能，在 RESET 引脚的复位信号释放后，当 OCDM.OCDM0 位为 0 时，将 P05/INTP2/DRST 引脚固定为低电平。

详情参见 4.5.3 片上调试引脚的注意事项。

- 注意事项
1. 电源打开后的复位期间，P53 引脚可能暂时输出不确定电平。
 2. 在复用功能引脚输入模式下，P50 至 P56 具有迟滞特性，但在端口模式下没有迟滞特性。

(1) 端口 5 寄存器 (P5)

(a) V850ES/JG3-U

复位后: 00H (输出锁存) R/W 地址: FFFFF40AH

	7	6	5	4	3	2	1	0
P5	0	P56	P55	P54	P53	P52	P51	P50

P5n	输出数据控制 (输出模式下) (n = 0至6)
0	输出 0
1	输出 1

(b) V850ES/JH3-U

复位后: 00H (输出锁存) R/W 地址: FFFFF40AH

	7	6	5	4	3	2	1	0
P5	0	0	0	0	0	0	P51	P50

P5n	输出数据控制 (输出模式下) (n = 0, 1)
0	输出 0
1	输出 1

(2) 端口 5 模式寄存器 (PM5)

(a) V850ES/JG3-U

复位后: FFH R/W 地址: FFFFF42AH

	7	6	5	4	3	2	1	0
PM5	1	PM56	PM55	PM54	PM53	PM52	PM51	PM50

PM5n	输入/输出模式控制 (n = 0至6)
0	输出模式
1	输入模式

(b) V850ES/JH3-U

复位后: FFH R/W 地址: FFFFF42AH

	7	6	5	4	3	2	1	0
PM5	1	1	1	1	1	1	PM51	PM50

PM5n	输入/输出模式控制 (n = 0, 1)
0	输出模式
1	输入模式

(3) 端口 5 模式控制寄存器 (PMC5)

(a) V850ES/JG3-U

复位后: 00H R/W 地址: FFFFF44AH

	7	6	5	4	3	2	1	0
PMC5	0	PMC56	PMC55	PMC54	PMC53	PMC52	PMC51	PMC50

PMC56	P56引脚操作模式的指定
0	I/O 端口
1	INTP05 输入

PMC55	P55引脚操作模式的指定
0	I/O 端口
1	SCKF2 I/O/KR5 输入/RTP05 输出

PMC54	P54 引脚操作模式的指定
0	I/O 端口
1	SOF2 输出/KR4 输入/RTP04 输出

PMC53	P53 引脚操作模式的指定
0	I/O 端口
1	SIF2 输入/KR3 输入/TIAB00 输入/TOAB00 输出/RTP03 输出

PMC52	P52 引脚操作模式的指定
0	I/O 端口
1	TIAB03 输入/KR2 输入/TOAB03 输出/RTP02 输出

PMC51	P51 引脚操作模式的指定
0	I/O 端口
1	TIAB02 输入/KR1 输入/TOAB02 输出/RTP01 输出/UDMAK1 输出

PMC50	P50 引脚操作模式的指定
0	I/O 端口
1	TIAB01 输入/KR0 输入/TOAB01 输出/RTP00 输出/UDMARQ1 输入

(b) V850ES/JH3-U

复位后: 00H R/W 地址: FFFFF44AH

	7	6	5	4	3	2	1	0
PMC5	0	0	0	0	0	0	PMC51	PMC50

PMC51	P51引脚操作模式的指定
0	I/O 端口
1	TIAB02 输入/KR1 输入/TOAB02 输出/RTP01 输出/UDMAK1 输出

PMC50	P50 引脚操作模式的指定
0	I/O 端口
1	TIAB01 输入/KR0 输入/TOAB01 输出/RTP00 输出/UDMARQ1 输入

(4) 端口 5 功能控制寄存器 (PFC5)

(a) V850ES/JG3-U

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

(b) V850ES/JH3-U

复位后: 00H R/W 地址: FFFFF46A H

	7	6	5	4	3	2	1	0
PFC5	0	0	0	0	0	0	PFC51	PFC50

备注 复用功能说明的详情, 参见 4.3.6 (6) 端口 5 复用功能说明。

(5) 端口 5 功能控制扩展寄存器 (PFCE5)

(a) V850ES/JG3-U

复位后: 00H R/W 地址: FFFFF70AH

	7	6	5	4	3	2	1	0
PFCE5	0	0	PFCE55	PFCE54	PFCE53	PFCE52	PFCE51	PFCE50

(b) V850ES/JH3-U

复位后: 00H R/W 地址: FFFFF70A H

	7	6	5	4	3	2	1	0
PFCE5	0	0	0	0	0	0	PFCE51	PFCE50

备注 复用功能说明的详情, 参见 4.3.6 (6) 端口 5 复用功能说明。

(6) 端口 5 复用功能指定

PFCE55 ^{※1}	PFC55 ^{※1}	P55 引脚复用功能的指定 ^{※1}
0	0	SCKF2 I/O
0	1	KR5 输入
1	0	RTP05 输出
1	1	禁止设置

PFCE54 ^{※1}	PFC54 ^{※1}	P54 引脚复用功能的指定 ^{※1}
0	0	SOF2 输出
0	1	KR4 输入
1	0	RTP04 输出
1	1	禁止设置

PFCE53 ^{注1}	PFC53 ^{注1}	P53引脚复用功能的指定 ^{注1}
0	0	SIF2 输入
0	1	TIAB00 输入/KR3 ^{注2} 输入
1	0	TOAB00 输出
1	1	RTP03 输出

PFCE52 ^{注1}	PFC52 ^{注1}	P52 引脚复用功能的指定 ^{注1}
0	0	TIAB03 输入/KR2 ^{注2} 输入
0	1	TOAB03 输出
1	0	RTP02 输出
1	1	禁止设置

PFCE51	PFC51	P51 引脚复用功能的指定
0	0	TIAB02 输入/KR1 ^{注2} 输入
0	1	TOAB02 输出
1	0	RTP01 输出
1	1	$\overline{\text{UDMAAK1}}$ 输出

PFCE50	PFC50	P50 引脚复用功能的指定
0	0	TIAB01 输入/KR0 ^{注2} 输入
0	1	TOAB01 输出
1	0	RTP00 输出
1	1	$\overline{\text{UDMARQ1}}$ 输入

注 1. 仅限 V850ES/JG3-U

2. KRn 和 TIAB0m 是复用功能。当引脚用作 TIAB0m 时，禁止复用功能 KRn 的引脚按键返回检测（清除 KRM.KRMn 位为 0）。同样，当引脚用作 KRn 时，禁止复用功能 TIAB0m 的引脚边沿检测（n = 0 至 3, m = 0 至 3）。

引脚名称	用作 TIAB0m 引脚	用作 KRn 引脚
KR0/TIAB01	KRM.KRM0 位 = 0	TAB0IOC1.TAB0TIG2, TAB0TIG3 位 = 0
KR1/TIAB02	KRM.KRM1 位 = 0	TAB0IOC1.TAB0TIG4, TAB0TIG5 位 = 0
KR2/TIAB03	KRM.KRM2 位 = 0	TAB0IOC1.TAB0TIG6, TAB0TIG7 位 = 0
KR3/TIAB00	KRM.KRM3 位 = 0	TAB0IOC1.TAB0TIG0, TAB0TIG1 位 = 0 TAB0IOC2.TAB0EES0, TAB0EES1 位 = 0 TAB0IOC2.TAB0ETS0, TAB0ETS1 位 = 0

(7) 端口 5 功能寄存器 (PF5)

(a) V850ES/JG3-U

复位后: 00H R/W 地址: FFFF6A0H

	7	6	5	4	3	2	1	0
PF5	0	PF56	PF55	PF54	PF53	PF52	PF51	PF50

PF5n	正常输出 / N沟道漏极开路输出的控制 (n = 0至6)
0	正常输出 (CMOS 输出)
1	N 沟道漏极开路输出

(b) V850ES/JH3-U

复位后: 00H R/W 地址: FFFF6A0H

	7	6	5	4	3	2	1	0
PF5	0	0	0	0	0	0	PF51	PF50

PF5n	正常输出 / N沟道漏极开路输出的控制 (n = 0, 1)
0	正常输出 (CMOS 输出)
1	N 沟道漏极开路输出

4.3.7 端口 6

端口 6 有 6 位输入输出端口，可以按位控制输入输出设置。

端口 6 包括以下复用功能引脚：

表 4-12. 端口 6 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
P60	65	90	TOAB1T1/TOAB11/TIAB11 /WAIT [‡]	输入/输出	-
P61	66	91	TOAB1B1/TOAB10/TIAB10 /RD [‡]	输入/输出	
P62	67	92	TOAB1T2/TOAB12/TIAB12 /ASTB [‡]	输入/输出	
P63	68	93	TOAB1B2/TRGAB1/ $\overline{\text{CS0}}$ [‡]	输入/输出	
P64	69	94	TOAB1T3/TOAB13/TIAB13 / $\overline{\text{CS2}}$ [‡]	输入/输出	
P65	70	95	TOAB1B3/EVTAB1/ $\overline{\text{CS3}}$ [‡]	输入/输出	

注 仅限 V850ES/JG3-U

注意事项 在复用功能引脚的输入模式下，P60 至 P65 具有迟滞特性，但端口模式下没有迟滞特性。

(1) 端口 6 寄存器 (P6)

复位后：00H（输出锁存） R/W 地址：FFFF40CH								
	7	6	5	4	3	2	1	0
P6	0	0	P65	P64	P63	P62	P61	P60
	输出数据控制（输出模式下）（n = 0至5）							
	P6n							
	0	输出 0						
	1	输出 1						

(2) 端口 6 模式寄存器 (PM6)

复位后: FFH R/W 地址: FFFFF42CH

	7	6	5	4	3	2	1	0
PM6	1	1	PM65	PM64	PM63	PM62	PM61	PM60

PM6n	输入/输出模式控制 (n = 0至5)
0	输出模式
1	输入模式

(3) 端口 6 模式控制寄存器 (PMC6)

复位后: 00H R/W 地址: FFFFF44CH

	7	6	5	4	3	2	1	0
PMC6	0	0	PMC65	PMC64	PMC63	PMC62	PMC61	PMC60

PMC65	P65引脚操作模式的指定
0	I/O 端口
1	TOAB1B3 输出/EVTAB1 输入/ $\overline{\text{CS3}}$ 输出 ^注

PMC64	P64引脚操作模式的指定
0	I/O 端口
1	TOAB1T3 输出/TOAB13 输出/TIAB13 输入/ $\overline{\text{CS2}}$ 输出 ^注

PMC63	P63引脚操作模式的指定
0	I/O 端口
1	TOAB1B2 输出/TRGAB1 输入/ $\overline{\text{CS0}}$ 输出 ^注

PMC62	P62引脚操作模式的指定
0	I/O 端口
1	TOAB1T2 输出/TOAB12 输出/TIAB12 输出/ASTB 输出 ^注

PMC61	P61引脚操作模式的指定
0	I/O 端口
1	TOAB1B1 输出/TIAB10 输入/TOAB10 输出/ $\overline{\text{RD}}$ 输出 ^注

PMC60	P60引脚操作模式的指定
0	I/O 端口
1	TOAB1T1 输出/TOAB11 输出/TIAB11 输入/ $\overline{\text{WAIT}}$ 输出 ^注

注 仅限 V850ES/JG3-U

(4) 端口 6 功能控制寄存器 (PFC6)

复位后: 00H R/W 地址: FFFFF468H

	7	6	5	4	3	2	1	0
PFC6	0	0	PFC65	PFC64	PFC63	PFC62	PFC61	PFC60

备注 复用功能说明的详情, 参见 4.3.7 (6) 端口 6 复用功能说明。

(5) 端口 6 功能控制扩展寄存器 (PFCE6)

(a) V850ES/JG3-U

复位后: 00H R/W 地址: FFFFF70CH

	7	6	5	4	3	2	1	0
PFCE6	0	0	PFCE65	PFCE64	PFCE63	PFCE62	PFCE61	PFCE60

(b) V850ES/JH3-U

复位后: 00H R/W 地址: FFFFF70CH

	7	6	5	4	3	2	1	0
PFCE6	0	0	0	0	0	0	PFCE61	0

备注 复用功能说明的详情, 参见 4.3.7 (6) 端口 6 复用功能说明。

(6) 端口 6 复用功能说明

PFCE65 ^注	PFC65	P65 引脚复用功能的指定
0	0	TOAB1B3 输出
0	1	EVTAB1 输入
1	0	$\overline{CS3}$ 输出 ^注
1	1	禁止设置 ^注

PFCE64 ^注	PFC64	P64 引脚复用功能的指定
0	0	TOAB1T3 输出/TOAB13 输出
0	1	TIAB13 输入
1	0	$\overline{CS2}$ 输出 ^注
1	1	禁止设置 ^注

PFCE63 ^注	PFC63	P63 引脚复用功能的指定
0	0	TOAB1B2 输出
0	1	TRGAB1 输入
1	0	CS0 输出 ^注
1	1	禁止设置 ^注

PFCE62 ^注	PFC62	P62 引脚复用功能的指定
0	0	TOAB1T2 输出/TOAB12 输出
0	1	TIAB12 输入
1	0	ASTB 输出 ^注
1	1	禁止设置 ^注

PFCE61	PFC61	P61 引脚复用功能的指定
0	0	TOAB1B1 输出
0	1	TIAB10 输入
1	0	TOAB10 输出
1	1	R \bar{D} 输出 (V850ES/JG3-U) 禁止设置 (V850ES/JH3-U)

PFCE60 ^注	PFC60	P60 引脚复用功能的指定
0	0	TOAB1T1 输出/TOAB11 输出
0	1	TIAB11 输入
1	0	WAIT 输出 ^注
1	1	禁止设置 ^注

注 仅限 V850ES/JG3-U

4.3.8 端口 7

端口 7 有 12 位输入输出端口，可以按位控制输入输出设置。

端口 7 包括以下复用功能引脚：

表 4-13.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	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
P70	100	128	ANI0	输入	-
P71	99	127	ANI1	输入	
P72	98	126	ANI2	输入	
P73	97	125	ANI3	输入	
P74	96	124	ANI4	输入	
P77	95	123	ANI5	输入	
P76	94	122	ANI6	输入	
P77	93	121	ANI7	输入	
P78	92	120	ANI8	输入	
P79	91	119	ANI9	输入	
P710	90	118	ANI10	输入	
P711	89	117	ANI11	输入	

(1) 端口 7 寄存器 H，端口 7 寄存器 L (P7H, P7L)

复位后：00H（输出锁存） R/W 地址：P7L FFFFF40EH, P7H FFFFF40FH

	7	6	5	4	3	2	1	0
P7H	0	0	0	0	P711	P710	P79	P78
	7	6	5	4	3	2	1	0
P7L	P77	P76	P75	P74	P73	P72	P71	P70

P7n	输出数据控制（输出模式下）（n = 0至11）
0	输出 0
1	输出 1

注意事项 A/D 转换期间，不要读取或写入 P7H 和 P7L 寄存器（参见 13.6（4） 复用 I/O）。

备注 这些寄存器不可以作为 P7 寄存器按 16 位宽度访问。可以作为 P7H 和 P7L 寄存器按照字节或按位进行读取或写入。

(2) 端口 7 模式寄存器 H, 端口 7 模式寄存器 L (PM7H, PM7L)

复位后: FFH R/W 地址: PM7L FFFFF42EH, PM7H FFFFF42FH

	7	6	5	4	3	2	1	0
PM7H	1	1	1	1	PM711	PM710	PM79	PM78
	7	6	5	4	3	2	1	0
PM7L	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	输入/输出模式控制 (n = 0至11)
0	输出模式
1	输入模式

注意事项 P7n 引脚用作复用功能 (ANIn 引脚) 时, 将 PM7n 位设置为 1。

备注 这些寄存器不可以作为 PM7 寄存器按 16 位宽度访问。可以作为 PM7H 和 PM7L 寄存器按照字节或按位进行读取或写入。

4.3.9 端口 9

端口 9 有 16 位输入输出端口，可以按位控制输入输出设置。

端口 9 包括以下复用功能引脚：

表 4-14. 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	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
P90	42	54	KR6/TXDC1/SDA02/A0 ^注	输入/输出	可选作 N 沟道漏极开路输出
P91	43	55	KR7/RXDC1/SCL02/A1 ^注	输入/输出	
P92	44	56	TENC01/TIT01/TOT01/A2 ^注	输入/输出	-
P93	45	57	TECR0/TIT00/TOT00/A3 ^注	输入/输出	
P94	46	58	TIAA31/TOAA31/TENC00 /EVTT0/A4 ^注	输入/输出	
P95	47	59	TIAA30/TOAA30/A5 ^注	输入/输出	
P96	48	62	TIAA21/TOAA21/INTP11/A6 ^注	输入/输出	
P97	49	63	SIF1/TIAA20/TOAA20/A7 ^注	输入/输出	
P98	50	64	SOF1/INTP12/A8 ^注	输入/输出	
P99	51	65	$\overline{\text{SCKF1}}$ /INTP13/A9 ^注	输入/输出	
P910	52	66	SIF3/TXDC2/INTP14/A10 ^注	输入/输出	
P911	53	67	SOF3/RXDC2/INTP15/A11 ^注	输入/输出	
P912	54	68	$\overline{\text{SCKF3}}$ /A12 ^注	输入/输出	
P913	55	69	TOAB1OFF/INTP16/A13 ^注	输入/输出	
P914	56	70	TIAA51/TOAA51/INTP17/A14 ^注	输入/输出	
P915	57	71	TIAA50/TOAA50/INTP18/A15 ^注	输入/输出	

注 仅限 V850ES/JH3-U

注意事项 在复用功能引脚的输入模式下，P90 至 P915 具有迟滞特性，但在端口模式下没有迟滞特性。

(1) 端口 9 寄存器 (P9)

复位后: 0000H (输出锁存) R/W 地址: P9 FFFFF412H,
P9L FFFFF412H, P9H 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	输出数据控制 (输出模式下) (n = 0至15)
0	输出 0
1	输出 1

- 备注**
1. P9 寄存器可以按 16 位宽度进行读写。
但是当寄存器 P9 的高 8 位和低 8 位分别用作 P9H 寄存器和 P9L 寄存器时, P9 可以按字节或按位读出或写入。
 2. 若要按字节或按位读取或写入 P9 的第 8 位至第 15 位, 将这些位作为 P9H 寄存器的第 0 位至第 7 位。

(2) 端口 9 模式寄存器 (PM9)

复位后: FFFFH R/W 地址: PM9 FFFFF432H,
PM9L FFFFF432H, PM9H 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	输入/输出模式控制 (n = 0至15)
0	输出模式
1	输入模式

- 备注**
1. PM9 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PM9 的高 8 位和低 8 位分别用作 PM9H 寄存器和 PM9L 寄存器时, PM9 可以按字节和按位读出或写入。
 2. 若要按字节或按位读取或写入 PM9 的第 8 位至第 15 位, 将这些位作为 PM9H 寄存器的第 0 位至第 7 位。

(3) 端口 9 模式控制寄存器 (PMC9)

(1/2)

复位后: 0000H R/W 地址: PMC9 FFFF452H,
 PMC9L FFFF452H, PMC9H FFFF453H

	15	14	13	12	11	10	9	8
PMC9(PMC9H)	PMC915	PMC914	PMC913	PMC912	PMC911	PMC910	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	TIAA50 输入/TOAA50 输出/INTP18 输入/A15 输出 ^注
PMC914	P914引脚操作模式的指定
0	I/O 端口
1	TIAA51 输入/TOAA51 输出/INTP17 输入/A14 输出 ^注
PMC913	P913引脚操作模式的指定
0	I/O 端口
1	TOAB1OFF 输入/INTP16 输入/A13 输出 ^注
PMC912	P912 引脚操作模式的指定
0	I/O 端口
1	SCKF3 I/O/A12 输出 ^注
PMC911	P911 引脚操作模式的指定
0	I/O 端口
1	SOF3 输出/RXDC2 输入/INTP15 输入/A11 输出 ^注
PMC910	P910 引脚操作模式的指定
0	I/O 端口
1	SIF3 输入/TXDC2 输出/INTP14 输入/A10 输出 ^注
PMC99	P99 引脚操作模式的指定
0	I/O 端口
1	SCKF1 I/O/INTP13 输入/A9 输出 ^注
PMC98	P98 引脚操作模式的指定
0	I/O 端口
1	SOF1 输出/INTP12 输入/A8 输出 ^注

注 仅限 V850ES/JG3-U

备注

1. PMC9 寄存器可以按 16 位宽度进行读写。
 但是当寄存器 PMC9 的高 8 位和低 8 位分别用作 PMC9H 寄存器和 PMC9L 寄存器时, PMC9 可以按字节或按位读出或写入。
2. 若要按字节或按位读取或写入 PMC9 的第 8 位至第 15 位, 将这些位作为 PMC9H 寄存器的第 0 位至第 7 位。

PMC97	P97引脚操作模式的指定
0	I/O 端口
1	SIF1 输入/TIAA20 输入/TOAA20 输出/A7 输出 ^注
PMC96	P96引脚操作模式的指定
0	I/O 端口
1	TIAA21 输入/TOAA21 输出/INTP11 输入/A6 输出 ^注
PMC95	P95引脚操作模式的指定
0	I/O 端口
1	TIAA30 输入/TOAA30 输出/A5 输出 ^注
PMC94	P94 引脚操作模式的指定
0	I/O 端口
1	TIAA31 输入/TOAA31 输出/TENC00 输入/EVTT0 输入/A4 输出 ^注
PMC93	P93 引脚操作模式的指定
0	I/O 端口
1	TECR0 输入/TIT00 输入/TOT00 输出/A3 输出 ^注
PMC92	P92 引脚操作模式的指定
0	I/O 端口
1	TENC01 输入/TIT01 输入/TOT01 输出/A2 输出 ^注
PMC91	P91 引脚操作模式的指定
0	I/O 端口
1	KR7 输入/RXDC1 输入/SCL02 I/O/A1 输出 ^注
PMC90	P90 引脚操作模式的指定
0	I/O 端口
1	KR6 输入/TXDC1 输出/SDA02 I/O/A0 输出 ^注

注 仅限 V850ES/JG3-U

注意事项 当 A0 至 A15 引脚作为 P90 至 P915 引脚的复用功能使用时，请确保一次将 PMC9 寄存器的所有 16 位设置为 FFFFH。

(4) 端口 9 功能控制寄存器 (PFC9)

注意事项 当执行分离地址总线输出(A0 至 A15)时, 在将 PFC9 和 PFCE9 寄存器清除为 0000H 之后, 一次将 PMC9 寄存器的所有 16 位设置为 FFFFH。(仅限 V850ES/JH3-U)

(a) V850ES/JG3-U

复位后: 0000H R/W 地址: PFC9 FFFFF472H,
PFC9L FFFFF472H, PFC9H FFFFF473H

	15	14	13	12	11	10	9	8
PFC9(PFC9H)	PFC915	PFC914	0	0	PFC911	PFC910	PFC99	PFC98
	7	6	5	4	3	2	1	0
PFC9L	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

(b) V850ES/JH3-U

复位后: 0000H R/W 地址: PFC9 FFFFF472H,
PFC9L FFFFF472H, PFC9H FFFFF473H

	15	14	13	12	11	10	9	8
PFC9(PFC9H)	PFC915	PFC914	PFC913	PFC912	PFC911	PFC910	PFC99	PFC98
	7	6	5	4	3	2	1	0
PFC9L	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

- 备注**
1. 复用功能说明的详情, 参见 4.3.9 (6) 端口 9 复用功能说明。
 2. PFC9 寄存器可以按 16 位宽度进行读取或写入。
但是当寄存器 PFC9 的高 8 位和低 8 位分别用作 PFC9H 寄存器和 PFC9L 寄存器时, PFC9 可以按字节和按位读出或写入。
 3. 若要按字节或按位读取或写入 PFC9 的第 8 位至第 15 位, 将这些位作为 PFC9H 寄存器的第 0 位至第 7 位。

(5) 端口 9 功能控制扩展寄存器 (PFCE9)

注意事项 当执行分离地址总线输出 (A0 至 A15) 时, 在 PFCE9 和 PFCE9 清除为 0000H 之后, 一次将 PMC9 寄存器的所有 16 位设置为 FFFFH。

(a) V850ES/JG3-U

复位后: 0000H R/W 地址: PFCE9 FFFFF712H,
PFCE9L FFFFF712H, PFCE9H FFFFF713H

	15	14	13	12	11	10	9	8
PFCE9(PFCE9H)	PFCE915	PFCE914	0	0	PFCE911	PFCE910	0	0
	7	6	5	4	3	2	1	0
PFCE9L	PFCE97	PFCE96	0	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

(b) V850ES/JH3-U

复位后: 0000H R/W 地址: PFCE9 FFFFF712H,
PFCE9L FFFFF712H, PFCE9H FFFFF713H

	15	14	13	12	11	10	9	8
PFCE9(PFCE9H)	PFCE915	PFCE914	0	0	PFCE911	PFCE910	PFCE99	PFCE98
	7	6	5	4	3	2	1	0
PFCE9L	PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

备注

1. 复用功能说明的详情, 参见 4.3.9 (6) 端口 9 复用功能说明。
2. PFCE9 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PFCE9 的高 8 位和低 8 位分别用作 PFCE9H 寄存器和 PFCE9L 寄存器时, PFCE9 可以按字节和按位读出或写入。
3. 若要按字节或按位读取或写入 PFCE9 的第 8 位至第 15 位, 将这些位作为 PFCE9H 寄存器的第 0 位至第 7 位。

(6) 端口 9 复用功能规范

PFCE915	PFC915	P915 引脚复用功能的指定
0	0	TIAA50 输入
0	1	TOAA50 输出
1	0	INTP18 输入
1	1	禁止设置 (V850ES/JG3-U) A15 输出 (V850ES/JH3-U)

PFCE914	PFC914	P914 引脚复用功能的指定
0	0	TIAA51 输入
0	1	TOAA51 输出
1	0	INTP17 输入
1	1	禁止设置 (V850ES/JG3-U) A14 输出 (V850ES/JH3-U)

PFC913 [※]	P913 引脚复用功能的指定
0	TOAB1OFF 输入/INTP16 输入
1	A13 输出 [※]

PFC912 [※]	P912 引脚复用功能的指定
0	$\overline{\text{SCKF3}}$ I/O
1	A12 输出 [※]

PFCE911	PFC911	P911 引脚复用功能的指定
0	0	SOF3 输出
0	1	RXDC2 输入
1	0	INTP15 输入
1	1	禁止设置 (V850ES/JG3-U) A11 输出 (V850ES/JH3-U)

PFCE910	PFC910	P910 引脚复用功能的指定
0	0	SIF3 输入
0	1	TXDC2 输出
1	0	INTP14 输入
1	1	禁止设置 (V850ES/JG3-U) A10 输出 (V850ES/JH3-U)

PFCE99 [※]	PFC99	P99 引脚复用功能的指定
0	0	$\overline{\text{SCKF1}}$ I/O
0	1	INTP14 输入
1	0	A9 输出 [※]
1	1	禁止设置 [※]

PFCE98 ^注	PFC98	P98 引脚复用功能的指定
0	0	SOF1 输出
0	1	INTP12 输入
1	0	A8 输出 ^注
1	1	禁止设置 ^注

PFCE97	PFC97	P97 引脚复用功能的指定
0	0	SIF1 输入
0	1	TIAA20 输入
1	0	TOAA20 输出
1	1	禁止设置 (V850ES/JG3-U) A7 输出 (V850ES/JH3-U)

PFCE96	PFC96	P96 引脚复用功能的指定
0	0	TIAA21 输入
0	1	TOAA21 输出
1	0	INTP11 输入
1	1	禁止设置(V850ES/JG3-U) A6 输出 (V850ES/JH3-U)

PFCE95 ^注	PFC95	P95 引脚复用功能的指定
0	0	TIAA30 输入
0	1	TOAA30 输出
1	0	A5 输出 ^注
1	1	禁止设置 ^注

PFCE94	PFC94	P94 引脚复用功能的指定
0	0	TIAA31 输入
0	1	TOAA31 输出
1	0	TENC00 输入/EVTT0 输入
1	1	禁止设置 (V850ES/JG3-U) A4 输出 (V850ES/JH3-U)

PFCE93	PFC93	P93 引脚复用功能的指定
0	0	TECR0 输入
0	1	TIT00 输入
1	0	TOT00 输出
1	1	禁止设置 (V850ES/JG3-U) A3 输出 (V850ES/JH3-U)

注 仅限 V850ES/JH3-U

PFCE92	PFC92	P92 引脚复用功能的指定
0	0	TENC01 输入
0	1	TIT01 输入
1	0	TOT01 输出
1	1	禁止设置 (V850ES/JG3-U) A2 输出 (V850ES/JH3-U)

PFCE91	PFC91	P91 引脚复用功能的指定
0	0	KR7 输入
0	1	RXDC1 输入
1	0	SCL02 I/O
1	1	禁止设置 (V850ES/JG3-U) A1 输出 (V850ES/JH3-U)

PFCE90	PFC90	P90 引脚复用功能的指定
0	0	KR6 输入
0	1	TXDC1 输出
1	0	SDA02 I/O
1	1	禁止设置 (V850ES/JG3-U) A0 输出 (V850ES/JH3-U)

(7) 端口 9 功能寄存器 (PF9)

复位后: 0000H R/W 地址: PF9 FFFFC72H,
PF9L FFFF C72H

	15	14	13	12	11	10	9	8
PF9	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
(PF9L)	0	0	0	0	0	0	P91	P90

PF9n	正常输出 / N沟道漏极开路输出的控制 (n = 0,1)
0	正常输出 (CMOS 输出)
1	N 沟道漏极开路输出

注意事项 若输出引脚 P90 和 P91 被上拉至 EV_{DD} 或更高电压时，确保将对应的 PF9n 位设置为 1。

备注 PF9 寄存器可以按 16 位宽度进行读取或写入。
但是当寄存器 PF9 的低 8 位被用作 PF9L 寄存器时，可以按字节和按位读出或写入。

4.3.10 端口 CM

端口 CM 有 1 位(V850ES/JG3-U) /4 位(V850ES/JH3-U)输入输出端口，可以按位控制输入输出设置。

端口 CM 包括以下复用功能引脚：

表 4-15. 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	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
PCM0	-	89	WAIT	输入	-
PCM1	64	86	CLKOUT	输出	
PCM2	-	10	HLD $\overline{\text{AK}}$	输出	
PCM3	-	11	HLD $\overline{\text{RQ}}$	输入	

(1) 端口 CM 寄存器 (PCM)

(a) V850ES/JG3-U

复位后：00H（输出锁存） R/W 地址：FFFFF00CH

	7	6	5	4	3	2	1	0
PCM	0	0	0	0	0	0	PCM1	0

PCM1	输出数据控制（输出模式下）
0	输出 0
1	输出 1

(b) V850ES/JH3-U

复位后：00H（输出锁存） R/W 地址：FFFFF00CH

	7	6	5	4	3	2	1	0
PCM	0	0	0	0	PCM3	PCM2	PCM1	PCM0

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

(2) 端口 CM 模式寄存器 (PMCM)

(a) V850ES/JG3-U

复位后: FFH R/W 地址: FFFFF02CH

	7	6	5	4	3	2	1	0
PMCM	1	1	1	1	1	1	PMCM 1	1

PMCMn	输入/输出模式控制
0	输出模式
1	输入模式

(b) V850ES/JH3-U

复位后: FFH R/W 地址: FFFFF02CH

	7	6	5	4	3	2	1	0
PMCM	1	1	1	1	PMCM3	PMCM2	PMCM 1	PMCM 0

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

(3) 端口 CM 模式控制寄存器 (PMCCM)

(a) V850ES/JG3-U

复位后: 00H R/W 地址: FFFF04CH

	7	6	5	4	3	2	1	0
PMCCM	0	0	0	0	0	0	PMCCM1	0

PMCCM1	PCM1 引脚操作模式的指定
0	I/O 端口
1	CLKOUT 输出

(b)V850ES/JH3-U

复位后: 00H R/W 地址: FFFF04CH

	7	6	5	4	3	2	1	0
PMCCM	0	0	0	0	PMCCM3	PMCCM2	PMCCM1	PMCCM0

PMCCM 3	PCM3 引脚操作模式的指定
0	I/O 端口
1	$\overline{\text{HLDRQ}}$ 输入

PMCCM 2	PCM2 引脚操作模式的指定
0	I/O 端口
1	$\overline{\text{HLDAK}}$ 输出

PMCCM 1	PCM1 引脚操作模式的指定
0	I/O 端口
1	CLKOUT 输出

PMCCM 0	PCM0 引脚操作模式的指定
0	I/O 端口
1	$\overline{\text{WAIT}}$ 输入

4.3.11 端口CS（仅限V850ES/JH3-U）

端口 CS 有 3 位输入输出端口，可以按位控制输入输出设置。

端口 CS 包括以下复用功能引脚：

表 4-16. 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	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
PCS0	-	96	$\overline{\text{CS0}}$	输出	-
PCS2	-	97	$\overline{\text{CS2}}$	输出	
PCS3	-	116	$\overline{\text{CS3}}$	输出	

(1) 端口 CS 寄存器 (PCS)

复位后: 00H (输出锁存)		R/W	地址: FFFFF008H					
	7	6	5	4	3	2	1	0
PCS	0	0	0	0	PCS3	PCS2	0	PCS0
PCS _n	输出数据控制 (输出模式下) (n = 0, 2, 3)							
0	输出 0							
1	输出 1							

(2) 端口 CS 模式寄存器 (PMCS)

复位后: FFH		R/W	地址: FFFFF028H					
	7	6	5	4	3	2	1	0
PMCS	1	1	1	1	PMCS3	PMCS2	1	PMCS0
PMCS _n	输入/输出模式控制 (n = 0, 2, 3)							
0	输出模式							
1	输入模式							

(3) 端口 CS 模式控制寄存器 (PMCCS)

复位后: 00H R/W 地址: FFFF048H

	7	6	5	4	3	2	1	0
PMCCS	0	0	0	0	PMCCS3	PMCCS2	0	PMCCS0

PMCCS3	PCS3 引脚操作模式的指定
0	I/O 端口
1	$\overline{\text{CS3}}$ 输出

PMCCS2	PCS2 引脚操作模式的指定
0	I/O 端口
1	$\overline{\text{CS2}}$ 输出

PMCCS1	PCS1 引脚操作模式的指定
0	I/O 端口
1	$\overline{\text{CS0}}$ 输出

4.3.12 端口CT

端口 CT 有 2 位(V850ES/JG3-U) /4 位(V850ES/JH3-U)输入输出端口，可以按位控制输入输出设置。

端口 CT 包括以下复用功能引脚：

表 4-17. 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	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
PCT0	58	76	$\overline{WR0}$	输出	-
PCT1	59	77	$\overline{WR1}$	输出	
PCT4	-	87	\overline{RD}	输出	
PCT6	-	88	ASTB	输出	

(1) 端口 CT 寄存器 (PCT)

(a) V850ES/JG3-U

复位后: 00H (输出锁存) R/W 地址: FFFFF00AH

	7	6	5	4	3	2	1	0
PCT	0	0	0	0	0	0	PCT1	PCT0

PCTn	输出数据控制 (输出模式下) (n = 0, 1)
0	输出 0
1	输出 1

(b) V850ES/JH3-U

复位后: 00H (输出锁存) R/W 地址: FFFFF00AH

	7	6	5	4	3	2	1	0
PCT	0	PCT6	0	4	0	0	PCT1	PCT0

PCTn	输出数据控制 (输出模式下) (n = 0, 1, 4, 6)
0	输出 0
1	输出 1

(2) 端口 CT 模式寄存器 (PMCT)

(a) V850ES/JG3-U

复位后: FFH R/W 地址: FFFFF02AH

	7	6	5	4	3	2	1	0
PMCT	1	1	1	1	1	1	PMCT1	PMCT0

PMCTn	输入/输出模式控制 (n = 0, 1)
0	输出模式
1	输入模式

(b) V850ES/JH3-U

复位后: FFH R/W 地址: FFFFF02CAH

	7	6	5	4	3	2	1	0
PMCT	1	PMCT6	1	PMCT4	1	1	PMCT1	PMCT0

PMCMn	输入/输出模式控制 (n = 0, 1, 4, 6)
0	输出模式
1	输入模式

(3) 端口 CT 模式控制寄存器 (PMCCT)

(a) V850ES/JG3-U

复位后: 00H R/W 地址: FFFF04AH

	7	6	5	4	3	2	1	0
PMCCT	0	0	0	0	0	0	PMCCT1	PMCCT0

PMCCT1	PCT1 引脚操作模式的指定
0	I/O 端口
1	$\overline{WR1}$ 输出

PMCCT0	PCT0 引脚操作模式的指定
0	I/O 端口
1	$\overline{WR0}$ 输出

(b) V850ES/JH3-U

复位后: 00H R/W 地址: FFFF04AH

	7	6	5	4	3	2	1	0
PMCCT	0	PMCCT5	0	PMCCT4	0	0	PMCCT1	PMCCT0

PMCCT6	PCT6 引脚操作模式的指定
0	I/O 端口
1	ASTB 输出

PMCCT4	PCT4 引脚操作模式的指定
0	I/O 端口
1	RD 输出

PMCCT1	PCT1 引脚操作模式的指定
0	I/O 端口
1	$\overline{WR1}$ 输出

PMCCT0	PCT0 引脚操作模式的指定
0	I/O 端口
1	$\overline{WR0}$ 输出

4.3.13 端口 DH (仅限V850ES/JH3-U)

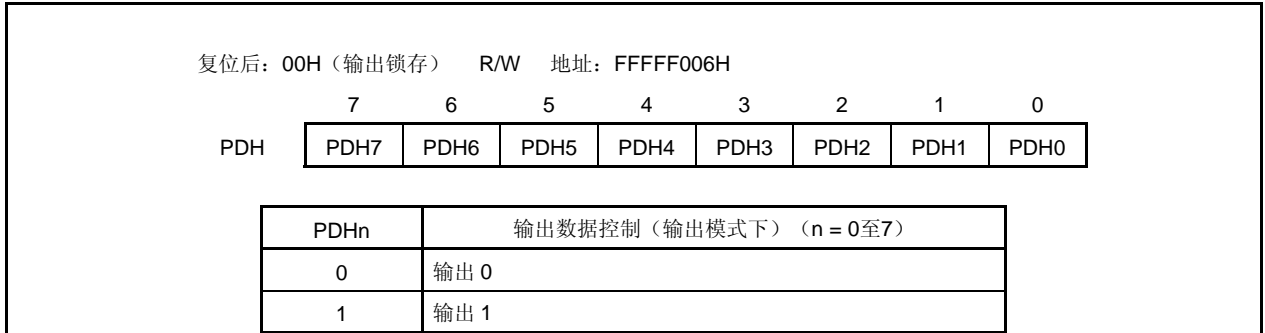
端口 DH 有 8 位输入输出端口，可以按位控制输入输出设置。

端口 DH 包括以下复用功能引脚：

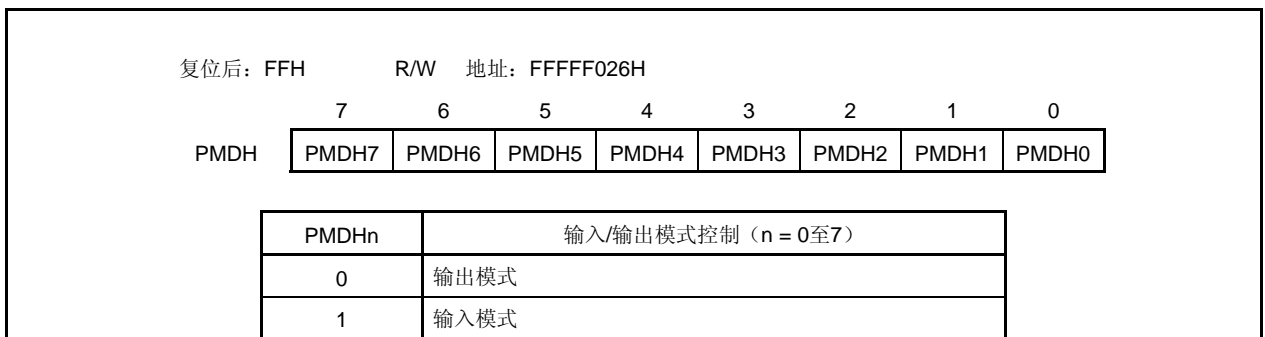
表 4-18. 端口 DH 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
PDH0	-	72	A16	输出	-
PDH1	-	73	A17	输出	
PDH2	-	74	A18	输出	
PDH3	-	75	A19	输出	
PDH4	-	78	A20	输出	
PDH5	-	79	A21	输出	
PDH6	-	80	A22	输出	
PDH7	-	81	A23	输出	

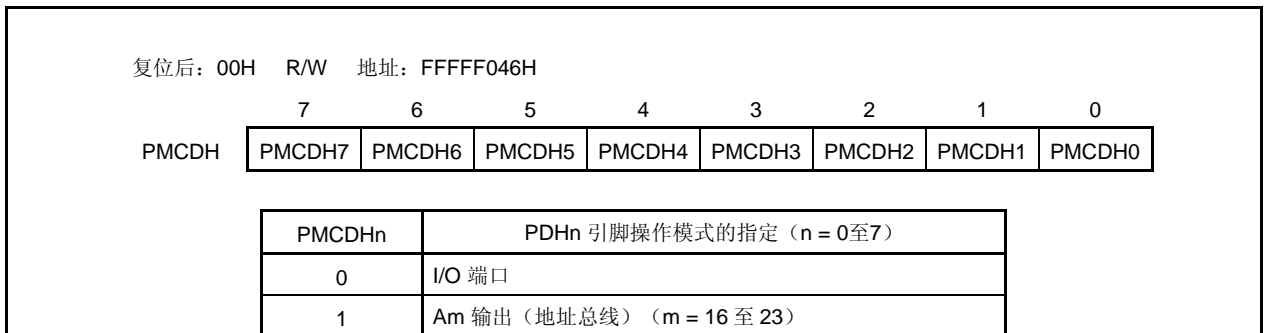
(1) 端口 DH 寄存器 (PDH)



(2) 端口 DH 模式寄存器 (PMDH)



(3) 端口 DH 模式控制寄存器 (PMCDH)



4.3.14 端口 DL

端口 DL 有 16 位输入输出端口，可以按位控制输入输出设置。

端口 DL 包括以下复用功能引脚：

表 4-19. 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	输入/输出	备注
	V850ES/ JG3-U	V850ES/ JH3-U			
PDL0	71	98	AD0	输入/输出	-
PDL1	72	99	AD1	输入/输出	
PDL2	73	100	AD2	输入/输出	
PDL3	74	101	AD3	输入/输出	
PDL4	75	102	AD4	输入/输出	
PDL5	78	103	AD5/FLMD1 ^注	输入/输出	
PDL6	79	104	AD6	输入/输出	
PDL7	80	105	AD7	输入/输出	
PDL8	81	108	AD8	输入/输出	
PDL9	82	109	AD9	输入/输出	
PDL10	83	110	AD10	输入/输出	
PDL11	84	111	AD11	输入/输出	
PDL12	85	112	AD12	输入/输出	
PDL13	86	113	AD13	输入/输出	
PDL14	87	114	AD14	输入/输出	
PDL15	88	115	AD15	输入/输出	

注 该引脚设置为 flash 存储器编程模式后，不需要通过端口控制寄存器进行操作。详情参见第 31 章 FLASH 存储器。

(1) 端口 DL 寄存器 (PDL)

复位后: 0000H (输出锁存) R/W 地址: PDL FFFFF004H,
PDLH FFFFF004H, PDLH FFFFF005H

	15	14	13	12	11	10	9	8
PDL(PDLH)	PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8
	7	6	5	4	3	2	1	0
(PDLL)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

PDLn	输出数据控制 (输出模式下) (n = 0至15)
0	输出 0
1	输出 1

- 备注**
1. PDL 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PDL 的高 8 位和低 8 位分别用作 PDLH 寄存器和 PDLL 寄存器时，PDL 可以按字节或按位读出或写入。
 2. 若要按字节或按位读取或写入 PDL 的第 8 位至第 15 位，将这些位作为 PDLH 寄存器的第 0 位至第 7 位。

(2) 端口 DL 模式寄存器 (PMDL)

复位后: FFFFH R/W 地址: PMDL FFFFF024H,
PMDLL FFFFF024H, PMDLH FFFFF025H

	15	14	13	12	11	10	9	8
PMDL(PMDLH)	PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8
	7	6	5	4	3	2	1	0
(PMDLL)	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0

PMDLn	输入/输出模式控制 (n = 0至15)
0	输出模式
1	输入模式

- 备注**
1. PMDL 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PMDL 的高 8 位和低 8 位分别用作 PMDLH 寄存器和 PMDLL 寄存器时，PMDL 可以按字节或按位读出或写入。
 2. 若要按字节或按位读取或写入 PMDL 的第 8 位至第 15 位，将这些位作为 PMDLH 寄存器的第 0 位至第 7 位。

(3) 端口 DL 模式控制寄存器 (PMCDL)

复位后: 0000H R/W 地址: PMCDL FFFF044H,
PMCDLL FFFF044H, PMCDLH FFFF045H

	15	14	13	12	11	10	9	8
PMCDL(PMCDLH)	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8
	7	6	5	4	3	2	1	0
PMCDLL	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0

PMCDLn	PDLn 引脚操作模式的指定 (n = 0至15)
0	I/O 端口
1	ADn 输入/输出 (地址/数据总线输入/输出)

备注

1. PMCDL 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PMCDL 的高 8 位和低 8 位分别用作 PMCDLH 寄存器和 PMCDLL 寄存器时, PMCDL 可以按字节或按位读出或写入。
2. 若要按字节或按位读取或写入 PMCDL 的第 8 位至第 15 位, 将这些位作为 PMCDLH 寄存器的第 0 位至第 7 位。

4.4 使用复用功能时的端口寄存器设置

表 4-20 显示各个端口用作复用功能时的端口寄存器设置。当一个端口引脚用作复用功能引脚时, 请参考各引脚的详细描述。

表 4-20. 端口引脚用作复用功能引脚(1/10)

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P00 ^{#1}	INTP00	输入	P00 = 无需设置	PM00 = 无需设置	PMC00 = 1	-	-	
P01 ^{#1}	INTP01	输入	P01 = 无需设置	PM01 = 无需设置	PMC01 = 1	-	-	
P02	NMI	输入	P02 = 无需设置	PM02 = 无需设置	PMC02 = 1	-	-	
P03	INTP02	输入	P03 = 无需设置	PM03 = 无需设置	PMC03 = 1	PFCE03 = 0	PFC03 = 0	
	ADTRG	输入	P03 = 无需设置	PM03 = 无需设置	PMC03 = 1	PFCE03 = 0	PFC03 = 1	
	UCLK	输入	P03 = 无需设置	PM03 = 无需设置	PMC03 = 1	PFCE03 = 1	PFC03 = 0	
P04 ^{#1}	INTP03 ^{#1}	输入	P04 = 无需设置	PM04 = 无需设置	PMC04 = 1	-	-	
P05 ^{#1}	INTP04 ^{#1}	输入	P05 = 无需设置	PM05 = 无需设置	PMC05 = 1	-	-	
P10	ANO0	输出	P10 = 无需设置	PM10 = 1	-	-	-	
P11	ANO1	输出	P11 = 无需设置	PM11 = 1	-	-	-	
P20 ^{#1}	TIAB03	输入	P20 = 无需设置	PM20 = 无需设置	PMC20 = 1	PFCE20 = 0	PFC20 = 0	
	KR2	输入	P20 = 无需设置	PM20 = 无需设置	PMC20 = 1	PFCE20 = 0	PFC20 = 1	
	TOAB03	输出	P20 = 无需设置	PM20 = 无需设置	PMC20 = 1	PFCE20 = 1	PFC20 = 0	
	RTP02	输出	P20 = 无需设置	PM20 = 无需设置	PMC20 = 1	PFCE20 = 1	PFC20 = 1	
P21 ^{#1}	SIF2	输入	P21 = 无需设置	PM21 = 无需设置	PMC21 = 1	PFCE21 = 0	PFC21 = 0	
	KR3 ^{#2}	输入	P21 = 无需设置	PM21 = 无需设置	PMC21 = 1	PFCE21 = 0	PFC21 = 1	
	TIAB00 ^{#2}	输入	P21 = 无需设置	PM21 = 无需设置	PMC21 = 1	PFCE21 = 0	PFC21 = 1	
	TOAB00	输出	P21 = 无需设置	PM21 = 无需设置	PMC21 = 1	PFCE21 = 1	PFC21 = 0	
	RTP03	输出	P21 = 无需设置	PM21 = 无需设置	PMC21 = 1	PFCE21 = 1	PFC21 = 1	

- 注
1. 仅限 V850ES/JH3-U
 2. KR3 和 TIAB00 是复用功能。当引脚用作 TIAB00 时，不要使用复用功能 KR3。同样，当引脚用作 KR3 时，不要使用复用功能 TIAB00。

注意事项 电源打开后的复位期间，P10 和 P11 引脚可能会暂时输出不确定电平。

表 4-20. 端口引脚用作复用功能引脚 (2/10)

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P22 ^注	SOF2	输出	P22 = 无需设置	PM22 = 无需设置	PMC22 = 1	PFCE22 = 0	PFC22 = 0	
	KR4	输入	P22 = 无需设置	PM22 = 无需设置	PMC22 = 1	PFCE22 = 0	PFC22 = 1	
	RTP04	输出	P22 = 无需设置	PM22 = 无需设置	PMC22 = 1	PFCE22 = 1	PFC22 = 0	
P23 ^注	$\overline{\text{SCKF2}}$	I/O	P23 = 无需设置	PM23 = 无需设置	PMC23 = 1	PFCE23 = 0	PFC23 = 0	
	KR5	输入	P23 = 无需设置	PM23 = 无需设置	PMC23 = 1	PFCE23 = 0	PFC23 = 1	
	RTP05	输出	P23 = 无需设置	PM23 = 无需设置	PMC23 = 1	PFCE23 = 1	PFC23 = 0	
P24 ^注	INTP05	输入	P24 = 无需设置	PM24 = 无需设置	PMC24 = 1	-	-	
P25 ^注	INTP06	输入	P25 = 无需设置	PM25 = 无需设置	PMC25 = 1	-	-	
P30	TXDC0	输出	P30 = 无需设置	PM30 = 无需设置	PMC30 = 1	PFCE30 = 0	PFC30 = 0	
	SOF4	输出	P30 = 无需设置	PM30 = 无需设置	PMC30 = 1	PFCE30 = 0	PFC30 = 1	
	INTP07	输入	P30 = 无需设置	PM30 = 无需设置	PMC30 = 1	PFCE30 = 1	PFC30 = 0	
P31	RXDC0	输入	P31 = 无需设置	PM31 = 无需设置	PMC31 = 1	PFCE31 = 0	PFC31 = 0	
	SIF4	输入	P31 = 无需设置	PM31 = 无需设置	PMC31 = 1	PFCE31 = 0	PFC31 = 1	
	INTP08	输入	P31 = 无需设置	PM31 = 无需设置	PMC31 = 1	PFCE31 = 1	PFC31 = 0	
P32	ASCKC0	输入	P32 = 无需设置	PM32 = 无需设置	PMC32 = 1	PFCE32 = 0	PFC32 = 0	
	$\overline{\text{SCKF4}}$	I/O	P32 = 无需设置	PM32 = 无需设置	PMC32 = 1	PFCE32 = 0	PFC32 = 1	
	TIAA00	输入	P32 = 无需设置	PM32 = 无需设置	PMC32 = 1	PFCE32 = 1	PFC32 = 0	
	TOAA00	输出	P32 = 无需设置	PM32 = 无需设置	PMC32 = 1	PFCE32 = 1	PFC32 = 1	
P33	TIAA01	输入	P33 = 无需设置	PM33 = 无需设置	PMC33 = 1	PFCE33 = 0	PFC33 = 0	
	TOAA01	输出	P33 = 无需设置	PM33 = 无需设置	PMC33 = 1	PFCE33 = 0	PFC33 = 1	
	RTCDIV	输出	P33 = 无需设置	PM33 = 无需设置	PMC33 = 1	PFCE33 = 1	PFC33 = 0	
	RTCCL	输出	P33 = 无需设置	PM33 = 无需设置	PMC33 = 1	PFCE33 = 1	PFC33 = 1	

注 仅限 V850ES/JH3-U

表 4-20. 端口引脚用作复用功能引脚 (3/10)

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P34	TIAA10	输入	P34 = 无需设置	PM34 = 无需设置	PMC34 = 1	PFCE34 = 0	PFC34 = 0	
	TOAA10	输出	P34 = 无需设置	PM34 = 无需设置	PMC34 = 1	PFCE34 = 0	PFC34 = 1	
	TOAA10FF [※]	输入	P34 = 无需设置	PM34 = 无需设置	PMC34 = 1	PFCE34 = 1	PFC34 = 0	
	INTP09 [※]	输入	P34 = 无需设置	PM34 = 无需设置	PMC34 = 1	PFCE34 = 1	PFC34 = 0	
	PPON	输出	P34 = 无需设置	PM34 = 无需设置	PMC34 = 1	PFCE34 = 1	PFC34 = 1	
P35	TIAA11	输入	P35 = 无需设置	PM35 = 无需设置	PMC35 = 1	PFCE35 = 0	PFC35 = 0	
	TOAA11	输出	P35 = 无需设置	PM35 = 无需设置	PMC35 = 1	PFCE35 = 0	PFC35 = 1	
	RTC1HZ	输出	P35 = 无需设置	PM35 = 无需设置	PMC35 = 1	PFCE35 = 1	PFC35 = 0	
	OCI	输入	P35 = 无需设置	PM35 = 无需设置	PMC35 = 1	PFCE35 = 1	PFC35 = 1	
P36	TXDC3	输出	P36 = 无需设置	PM36 = 无需设置	PMC36 = 1	PFCE36 = 0	PFC36 = 0	
	SCL00	I/O	P36 = 无需设置	PM36 = 无需设置	PMC36 = 1	PFCE36 = 0	PFC36 = 1	PF36 (PF3) = 1
	UDMARQ0	输入	P36 = 无需设置	PM36 = 无需设置	PMC36 = 1	PFCE36 = 1	PFC36 = 1	
P37	RXDC3	输入	P37 = 无需设置	PM37 = 无需设置	PMC37 = 1	PFCE37 = 0	PFC37 = 0	
	SDA00	I/O	P37 = 无需设置	PM37 = 无需设置	PMC37 = 1	PFCE37 = 0	PFC37 = 1	PF37 (PF3) = 1
	UDMAAK0	输出	P37 = 无需设置	PM37 = 无需设置	PMC37 = 1	PFCE37 = 1	PFC37 = 1	
P40	SIF0	输入	P40 = 无需设置	PM40 = 无需设置	PMC40 = 1	PFCE40 = 0	PFC40 = 0	
	TXDC4	输出	P40 = 无需设置	PM40 = 无需设置	PMC40 = 1	PFCE40 = 0	PFC40 = 1	
	SDA01	I/O	P40 = 无需设置	PM40 = 无需设置	PMC40 = 1	PFCE40 = 1	PFC40 = 0	PF40 (PF4) = 1
P41	SOF0	输出	P41 = 无需设置	PM41 = 无需设置	PMC41 = 1	PFCE41 = 0	PFC41 = 0	
	RXDC4	输入	P41 = 无需设置	PM41 = 无需设置	PMC41 = 1	PFCE41 = 0	PFC41 = 1	
	SCL01	I/O	P41 = 无需设置	PM41 = 无需设置	PMC41 = 1	PFCE41 = 1	PFC41 = 0	PF41 (PF4) = 1
P42	SCKF0	I/O	P42 = 无需设置	PM42 = 无需设置	PMC42 = 1	-	PFC42 = 0	
	INTP10	输入	P42 = 无需设置	PM42 = 无需设置	PMC42 = 1	-	PFC42 = 1	

注 TOAA10FF 和 INTP09 是复用功能引脚。当引脚用作 TOAA10FF 时，禁止复用功能 INTP09 的引脚边沿检测(设置 INTF3.INTF34 = 0, INTR3.INTR34 = 0)。同样，当引脚用作 INTP09 时，停止高阻抗输出控制器。

表 4-20. 端口引脚用作复用功能引脚 (4/10)

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P50	TIAB01	输入	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 0	PFC50 = 0	KRM0 (KRM) = 0
	KR0	输入	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 0	PFC50 = 0	TAB0TIG2, TAB0TIG3 (TAB0IOC1) = 0
	TOAB01	输出	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 0	PFC50 = 1	
	RTP00	输出	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 1	PFC50 = 0	
	UDMARQ1	输入	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 1	PFC50 = 1	
P51	TIAB02	输入	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 0	PFC51 = 0	KRM1 (KRM) = 0
	KR1	输入	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 0	PFC51 = 0	TAB0TIG4, TAB0TIG5 (TAB0IOC1) = 0
	TOAB02	输出	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 0	PFC51 = 1	
	RTP01	输出	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 1	PFC51 = 0	
	UDMAAK1	输出	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 1	PFC51 = 1	
P52 ^注	TIAB03	输入	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 0	PFC52 = 0	KRM2 (KRM) = 0
	KR2	输入	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 0	PFC52 = 0	TAB0TIG6, TAB0TIG7 (TAB0IOC1) = 0
	TOAB03	输出	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 0	PFC52 = 1	
	RTP02	输出	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 1	PFC52 = 0	
P53 ^注	SIF2	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 0	PFC53 = 0	
	TIAB00	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 0	PFC53 = 1	KRM3 (KRM) = 0
	KR3	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 0	PFC53 = 1	TAB0TIG0, TAB0TIG1 (TAB0IOC1) = 0, TAB0EES0, TAB0EES1 (TAB0IOC2) = 0, TAB0ETS0, TAB0ETS1 (TAB0IOC2) = 0
	TOAB00	输出	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 1	PFC53 = 0	
	RTP03	输出	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 1	PFC53 = 1	
P54 ^注	SOF2	输出	P54 = 无需设置	PM54 = 无需设置	PMC54 = 1	PFCE54 = 0	PFC54 = 0	
	KR4	输入	P54 = 无需设置	PM54 = 无需设置	PMC54 = 1	PFCE54 = 0	PFC54 = 1	
	RTP04	输出	P54 = 无需设置	PM54 = 无需设置	PMC54 = 1	PFCE54 = 1	PFC54 = 0	

注 仅限 V850ES/JG3-U

表 4-20. 端口引脚用作复用功能引脚 (5/10)

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P55 [※]	$\overline{\text{SCKF2}}$	I/O	P55 = 无需设置	PM55 = 无需设置	PMC55 = 1	PFCE55 = 0	PFC55 = 0	
	KR5	输入	P55 = 无需设置	PM55 = 无需设置	PMC55 = 1	PFCE55 = 0	PFC55 = 1	
	RTP05	输出	P55 = 无需设置	PM55 = 无需设置	PMC55 = 1	PFCE55 = 1	PFC55 = 0	
P60	TOAB1T1	输出	P60 = 无需设置	PM60 = 无需设置	PMC60 = 1	PFCE60 = 0 [※]	PFC60 = 0	
	TOAB11	输出	P60 = 无需设置	PM60 = 无需设置	PMC60 = 1	PFCE60 = 0 [※]	PFC60 = 0	
	TIAB11	输入	P60 = 无需设置	PM60 = 无需设置	PMC60 = 1	PFCE60 = 0 [※]	PFC60 = 1	
	$\overline{\text{WAIT}}^{\text{※}}$	输出	P60 = 无需设置	PM60 = 无需设置	PMC60 = 1	PFCE60 = 1 [※]	PFC60 = 0	
P61	TOAB1B1	输出	P61 = 无需设置	PM61 = 无需设置	PMC61 = 1	PFCE61 = 0	PFC61 = 0	
	TIAB10	输入	P61 = 无需设置	PM61 = 无需设置	PMC61 = 1	PFCE61 = 0	PFC61 = 1	
	TOAB10	输出	P61 = 无需设置	PM61 = 无需设置	PMC61 = 1	PFCE61 = 1	PFC61 = 0	
	$\overline{\text{RD}}^{\text{※}}$	输出	P61 = 无需设置	PM61 = 无需设置	PMC61 = 1	PFCE61 = 1	PFC61 = 1	
P62	TOAB1T2	输出	P62 = 无需设置	PM62 = 无需设置	PMC62 = 1	PFCE62 = 0 [※]	PFC62 = 0	
	TOAB12	输出	P62 = 无需设置	PM62 = 无需设置	PMC62 = 1	PFCE62 = 0 [※]	PFC62 = 0	
	TIAB12	输入	P62 = 无需设置	PM62 = 无需设置	PMC62 = 1	PFCE62 = 0 [※]	PFC62 = 1	
	ASTB [※]	输出	P62 = 无需设置	PM62 = 无需设置	PMC62 = 1	PFCE62 = 1 [※]	PFC62 = 0	
P63	TOAB1B2	输出	P63 = 无需设置	PM63 = 无需设置	PMC63 = 1	PFCE63 = 0 [※]	PFC63 = 0	
	TRGAB1	输入	P63 = 无需设置	PM63 = 无需设置	PMC63 = 1	PFCE63 = 0 [※]	PFC63 = 1	
	$\overline{\text{CS0}}^{\text{※}}$	输出	P63 = 无需设置	PM63 = 无需设置	PMC63 = 1	PFCE63 = 1 [※]	PFC63 = 0	
P64	TOAB1T3	输出	P64 = 无需设置	PM64 = 无需设置	PMC64 = 1	PFCE64 = 0 [※]	PFC64 = 0	
	TOAB13	输出	P64 = 无需设置	PM64 = 无需设置	PMC64 = 1	PFCE64 = 0 [※]	PFC64 = 0	
	TIAB13	输入	P64 = 无需设置	PM64 = 无需设置	PMC64 = 1	PFCE64 = 0 [※]	PFC64 = 1	
	$\overline{\text{CS2}}^{\text{※}}$	输出	P64 = 无需设置	PM64 = 无需设置	PMC64 = 1	PFCE64 = 1 [※]	PFC64 = 0	
P65	TOAB1B3	输出	P65 = 无需设置	PM65 = 无需设置	PMC65 = 1	PFCE65 = 0 [※]	PFC63 = 0	
	EVTAB1	输入	P65 = 无需设置	PM65 = 无需设置	PMC65 = 1	PFCE65 = 0 [※]	PFC65 = 1	
	$\overline{\text{CS3}}^{\text{※}}$	输出	P65 = 无需设置	PM65 = 无需设置	PMC65 = 1	PFCE65 = 1 [※]	PFC65 = 0	

注 仅限 V850ES/JG3-U

表 4-20. 端口引脚用作复用功能引脚 (6/10)

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P70	ANI0	输入	P70 = 无需设置	PM70 = 1	-	-	-	
P71	ANI1	输入	P71 = 无需设置	PM71 = 1	-	-	-	
P72	ANI2	输入	P72 = 无需设置	PM72 = 1	-	-	-	
P73	ANI3	输入	P73 = 无需设置	PM73 = 1	-	-	-	
P74	ANI4	输入	P74 = 无需设置	PM74 = 1	-	-	-	
P75	ANI5	输入	P75 = 无需设置	PM75 = 1	-	-	-	
P76	ANI6	输入	P76 = 无需设置	PM76 = 1	-	-	-	
P77	ANI7	输入	P77 = 无需设置	PM77 = 1	-	-	-	
P78	ANI8	输入	P78 = 无需设置	PM78 = 1	-	-	-	
P79	ANI9	输入	P79 = 无需设置	PM79 = 1	-	-	-	
P710	ANI10	输入	P710 = 无需设置	PM710 = 1	-	-	-	
P711	ANI11	输入	P711 = 无需设置	PM711 = 1	-	-	-	
P90	KR6	输入	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 0	PFC90 = 0	
	TXDC1	输出	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 0	PFC90 = 1	
	SDA02	I/O	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 1	PFC90 = 0	PF90 (PF9) = 1
	A0 ^{#1}	输出	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 1	PFC90 = 1	注 2
P91	KR7	输入	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 0	PFC91 = 0	
	RXDC1	输入	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 0	PFC91 = 1	
	SCL02	I/O	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 1	PFC91 = 0	PF91 (PF9) = 1
	A1 ^{#1}	输出	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 1	PFC91 = 1	注 2
P92	TENC01	输入	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 0	PFC92 = 0	
	TIT01	输入	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 0	PFC92 = 1	
	TOT01	输出	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 1	PFC92 = 0	
	A2 ^{#1}	输出	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 1	PFC92 = 1	注 2

- 注
1. 仅限 V850ES/JH3-U
 2. 当用作 A0 至 A15 引脚，请确保一次将 PMC9 寄存器的所有 16 位设置为 FFFFH。

表 4-20. 端口引脚用作复用功能引脚 (7/10)

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P93	TECR0	输入	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 0	PFC93 = 0	
	TIT00	输入	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 0	PFC93 = 1	
	TOT00	输出	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 1	PFC93 = 0	
	A3 ^{#1}	输出	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 1	PFC93 = 0	注 2
P94	TIAA31	输入	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 0	PFC94 = 0	
	TOAA31	输出	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 0	PFC94 = 1	
	TENC00	输入	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 1	PFC94 = 0	
	EVTT0	输入	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 1	PFC94 = 0	
	A4 ^{#1}	输出	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 1	PFC94 = 1	注 2
P95	TIAA30	输入	P95 = 无需设置	PM95 = 无需设置	PMC95 = 1	PFCE95 = 0	PFC95 = 0	
	TOAA30	输出	P95 = 无需设置	PM95 = 无需设置	PMC95 = 1	PFCE95 = 0	PFC95 = 1	
	A5 ^{#1}	输出	P95 = 无需设置	PM95 = 无需设置	PMC95 = 1	PFCE95 = 1	PFC95 = 0	注 2
P96	TIAA21	输入	P96 = 无需设置	PM96 = 无需设置	PMC96 = 1	PFCE96 = 0	PFC96 = 0	
	TOAA21	输出	P96 = 无需设置	PM96 = 无需设置	PMC96 = 1	PFCE96 = 1	PFC96 = 0	
	INTP11	输入	P96 = 无需设置	PM96 = 无需设置	PMC96 = 1	PFCE96 = 1	PFC96 = 0	
	A6 ^{#1}	输出	P96 = 无需设置	PM96 = 无需设置	PMC96 = 1	PFCE96 = 1	PFC96 = 0	注 2
P97	SIF1	输入	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 0	PFC97 = 0	
	TIAA20	输入	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 0	PFC97 = 1	
	TOAA20	输出	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 1	PFC97 = 0	
	A7 ^{#1}	输出	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 1	PFC97 = 1	注 2
P98	SOF1	输出	P98 = 无需设置	PM98 = 无需设置	PMC98 = 1	PFCE98 = 0	PFC98 = 0	
	INTP12	输入	P98 = 无需设置	PM98 = 无需设置	PMC98 = 1	PFCE98 = 0	PFC98 = 1	
	A8 ^{#1}	输出	P98 = 无需设置	PM98 = 无需设置	PMC98 = 1	PFCE98 = 1	PFC98 = 0	注 2

- 注
1. 仅限 V850ES/JH3-U
 2. 当用作 A0 至 A15 引脚, 请确保一次将 PMC9 寄存器的所有 16 位设置为 FFFFH。

表 4-20. 端口引脚用作复用功能引脚 (8/10)

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P99	SCKF1	I/O	P99 = 无需设置	PM99 = 无需设置	PMC99 = 1	PFCE99 = 0	PFC99 = 0	
	INTP14	输入	P99 = 无需设置	PM99 = 无需设置	PMC99 = 1	PFCE99 = 0	PFC99 = 1	
	A9 ^{#1}	输出	P99 = 无需设置	PM99 = 无需设置	PMC99 = 1	PFCE99 = 1	PFC99 = 0	注 2
P910	SIF3	输入	P910 = 无需设置	PM910 = 无需设置	PMC910 = 1	PFCE910 = 0	PFC910 = 0	
	TXDC2	输出	P910 = 无需设置	PM910 = 无需设置	PMC910 = 1	PFCE910 = 0	PFC910 = 1	
	INTP14	输入	P910 = 无需设置	PM910 = 无需设置	PMC910 = 1	PFCE910 = 1	PFC910 = 0	
	A10 ^{#1}	输出	P910 = 无需设置	PM910 = 无需设置	PMC910 = 1	PFCE910 = 1	PFC910 = 1	注 2
P911	SOF3	输出	P911 = 无需设置	PM911 = 无需设置	PMC911 = 1	PFCE911 = 0	PFC911 = 0	
	RXDC2	输入	P911 = 无需设置	PM911 = 无需设置	PMC911 = 1	PFCE911 = 0	PFC911 = 1	
	INTP15	输入	P911 = 无需设置	PM911 = 无需设置	PMC911 = 1	PFCE911 = 1	PFC911 = 0	
	A11 ^{#1}	输出	P911 = 无需设置	PM911 = 无需设置	PMC911 = 1	PFCE911 = 1	PFC911 = 1	注 2
P912	SCKF3	I/O	P912 = 无需设置	PM912 = 无需设置	PMC912 = 1	-	PFC912 = 0	
	A12 ^{#1}	输出	P912 = 无需设置	PM912 = 无需设置	PMC912 = 1	-	PFC912 = 1	注 2
P913	TOAB1OFF	输入	P913 = 无需设置	PM913 = 无需设置	PMC913 = 1	-	PFC913 = 0	
	INTP16	输入	P913 = 无需设置	PM913 = 无需设置	PMC913 = 1	-	PFC913 = 0	
	A13 ^{#1}	输出	P913 = 无需设置	PM913 = 无需设置	PMC913 = 1	-	PFC913 = 1	注 2
P914	TIAA51	输入	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 0	PFC914 = 0	
	TOAA51	输出	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 0	PFC914 = 1	
	INTP17	输入	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 1	PFC914 = 0	
	A14 ^{#1}	输出	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 1	PFC914 = 1	注 2
P915	TIAA50	输入	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 0	PFC915 = 0	
	TOP50	输出	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 0	PFC915 = 1	
	INTP18	输入	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 1	PFC915 = 0	
	A15 ^{#1}	输出	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 1	PFC915 = 1	注 2

- 注
1. 仅限 V850ES/JH3-U
 2. 当用作 A0 至 A15 引脚，请确保一次将 PMC9 寄存器的所有 16 位设置为 FFFFH。

表 4-20. 端口引脚用作复用功能引脚 (9/10)

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
PCM0	$\overline{\text{WAIT}}^{\#}$	输入	PCM0 = 无需设置	PCM0 = 无需设置	PMCCM0 = 1	-	-	
PCM1	CLKOUT	输出	PCM1 = 无需设置	PCM1 = 无需设置	PMCCM1 = 1	-	-	
PCM2	$\overline{\text{HLDAK}}^{\#}$	输出	PCM2 = 无需设置	PCM2 = 无需设置	PMCCM2 = 1	-	-	
PCM3	$\overline{\text{HLDRQ}}^{\#}$	输入	PCM3 = 无需设置	PCM3 = 无需设置	PMCCM3 = 1	-	-	
PCS0	$\overline{\text{CS0}}^{\#}$	输出	PCS0 = 无需设置	PMCS0 = 无需设置	PMCCS0 = 1	-	-	
PCS2	$\overline{\text{CS2}}^{\#}$	输出	PCS2 = 无需设置	PMCS2 = 无需设置	PMCCS2 = 1	-	-	
PCS3	$\overline{\text{CS3}}^{\#}$	输出	PCS3 = 无需设置	PMCS3 = 无需设置	PMCCS3 = 1	-	-	
PCT0	$\overline{\text{WR0}}$	输出	PCT0 = 无需设置	PMCT0 = 无需设置	PMCCCT0 = 1	-	-	
PCT1	$\overline{\text{WR1}}$	输出	PCT1 = 无需设置	PMCT1 = 无需设置	PMCCCT1 = 1	-	-	
PCT4	$\overline{\text{RD}}^{\#}$	输出	PCT4 = 无需设置	PMCT4 = 无需设置	PMCCCT4 = 1	-	-	
PCT6	$\overline{\text{ASTB}}^{\#}$	输出	PCT6 = 无需设置	PMCT6 = 无需设置	PMCCCT6 = 1	-	-	
PDH0	A16	输出	PDH0 = 无需设置	PMDH0 = 无需设置	PMCDH0 = 1	-	-	
PDH1	A17	输出	PDH1 = 无需设置	PMDH1 = 无需设置	PMCDH1 = 1	-	-	
PDH2	A18	输出	PDH2 = 无需设置	PMDH2 = 无需设置	PMCDH2 = 1	-	-	
PDH3	A19	输出	PDH3 = 无需设置	PMDH3 = 无需设置	PMCDH3 = 1	-	-	
PDH4	A20	输出	PDH4 = 无需设置	PMDH4 = 无需设置	PMCDH4 = 1	-	-	
PDH5	A21	输出	PDH5 = 无需设置	PMDH5 = 无需设置	PMCDH5 = 1	-	-	
PDL0	AD0	I/O	PDL0 = 无需设置	PMDL0 = 无需设置	PMCDL0 = 1	-	-	
PDL1	AD1	I/O	PDL1 = 无需设置	PMDL1 = 无需设置	PMCDL1 = 1	-	-	
PDL2	AD2	I/O	PDL2 = 无需设置	PMDL2 = 无需设置	PMCDL2 = 1	-	-	
PDL3	AD3	I/O	PDL3 = 无需设置	PMDL3 = 无需设置	PMCDL3 = 1	-	-	
PDL4	AD4	I/O	PDL4 = 无需设置	PMDL4 = 无需设置	PMCDL4 = 1	-	-	

注 仅限 V850ES/JH3-U

表 4-20. 端口引脚用作复用功能引脚 (10/10)

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
PDL5	AD5	I/O	PDL5 = 无需设置	PMDL5 = 无需设置	PMCDL5 = 1	-	-	
	FLMD1 ^注	输入	PDL5 = 无需设置	PMDL5 = 无需设置	PMCDL5 = 无需设置	-	-	
PDL6	AD6	I/O	PDL6 = 无需设置	PMDL6 = 无需设置	PMCDL6 = 1	-	-	
PDL7	AD7	I/O	PDL7 = 无需设置	PMDL7 = 无需设置	PMCDL7 = 1	-	-	
PDL8	AD8	I/O	PDL8 = 无需设置	PMDL8 = 无需设置	PMCDL8 = 1	-	-	
PDL9	AD9	I/O	PDL9 = 无需设置	PMDL9 = 无需设置	PMCDL9 = 1	-	-	
PDL10	AD10	I/O	PDL10 = 无需设置	PMDL10 = 无需设置	PMCDL10 = 1	-	-	
PDL11	AD11	I/O	PDL11 = 无需设置	PMDL11 = 无需设置	PMCDL11 = 1	-	-	
PDL12	AD12	I/O	PDL12 = 无需设置	PMDL12 = 无需设置	PMCDL12 = 1	-	-	
PDL13	AD13	I/O	PDL13 = 无需设置	PMDL13 = 无需设置	PMCDL13 = 1	-	-	
PDL14	AD14	I/O	PDL14 = 无需设置	PMDL14 = 无需设置	PMCDL14 = 1	-	-	
PDL15	AD15	I/O	PDL15 = 无需设置	PMDL15 = 无需设置	PMCDL15 = 1	-	-	

注 该引脚设置为 flash 存储器编程模式后，不需要通过端口控制寄存器进行操作。详情参见第 31 章 FLASH 存储器。

4.5 注意事项

4.5.1 设置端口引脚的注意事项

(1) V850ES/JG3-U 和 V850ES/JH3-U 中，通用端口功能和一些外围功能 I/O 引脚共用引脚。通过设置 PMCn 寄存器，可以在通用端口（端口模式）与外围功能 I/O 引脚（复用功能模式）之间相互转换。有关该寄存器的设置步骤，注意以下事项：

(a) 从端口模式转换至复用功能模式的注意事项

从端口模式至复用功能模式的转换操作按以下顺序进行：

- | | |
|--|------------|
| <1> 设置 PFn 寄存器 ^{#1} ： | N 沟道漏极开路设置 |
| <2> 设置 PFCn 和 PFCEn 寄存器： | 复用功能选择 |
| <3> 设置 PMCn 寄存器的相应位为 1： | 转换至复用功能模式 |
| <4> 设置 INTRn 和 INTFn 寄存器 ^{#2} ： | 外部中断设置 |

如果先设置 PMCn 寄存器，需要特别注意的是，此时可能会发生不可预料的操作。或者按照 PFn，PFCn，和 PFCEn 寄存器的设置使得脚状态发生变化时，也有可能发生误操作。

下面会有一个具体举例。

- 注
1. 只有 N 沟道漏极开路输出
 2. 只有选择外部中断功能时

注意事项 不考虑是端口模式还是复用功能模式，Pn 寄存器都可以按照以下方法进行读取和写入。

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

[实例] SCL01 引脚设置举例。

SCL01 引脚复用作 P41/SOF0 引脚。通过 PMC4，PFC4 和 PF4 寄存器选择有效引脚功能。

PMC41 位	PFC41 位	PF41 位	有效引脚功能
0	无需理会	1	P41（输出端口模式，N 沟道漏极开路输出）
1	0	1	SOF0 输出（N 沟道漏极开路输出）
	1	1	SCL01 输入/输出（N 沟道漏极开路输出）

从 P41 引脚转换至 SCL01 引脚时，如下的设置顺序有可能发生误操作：

设置顺序	设置内容	引脚状态	引脚电平
<1>	初始值 (PMC41 位 = 0, PFC41 位 = 0, PF41 位 = 0)	端口模式 (输入)	高阻
<2>	PMC41 位 ← 1	SOF0 输出	低电平 (根据 CSIF0 的设置为高电平)
<3>	PFC41 位 ← 1	SCL01 I/O	高电平 (CMOS 输出)
<4>	PF41 位 ← 1	SCL01 I/O	高阻 (N 沟道漏极开路输出)

在步骤<2>中，因为复用功能 SOF0 的信号是输出至该引脚，所以 I²C 通信可能会受影响。在步骤<2>或<3>的 CMOS 输出期间，有可能产生不必要的电流。

(b) 复用功能模式 (输入) 的注意事项

当 PMCn 寄存器的设定值和该引脚的电平进行与 (AND) 输出，使得 PMCn.PMCnm 位为 0 时，输入到复用功能模块的信号为低电平。因此，根据端口设置和复用功能的操作使能时序，可能会发生不可预料的操作。所以从端口模式转换至复用功能模式的操作按照以下顺序进行。

- 从端口模式转换至复用功能模式 (输入)
通过 PMCn 寄存器将引脚设置为复用功能模式，然后使能复用功能操作。
- 从复用功能模式转换至端口模式 (输入)
停止复用功能操作，然后将引脚转换至端口模式。

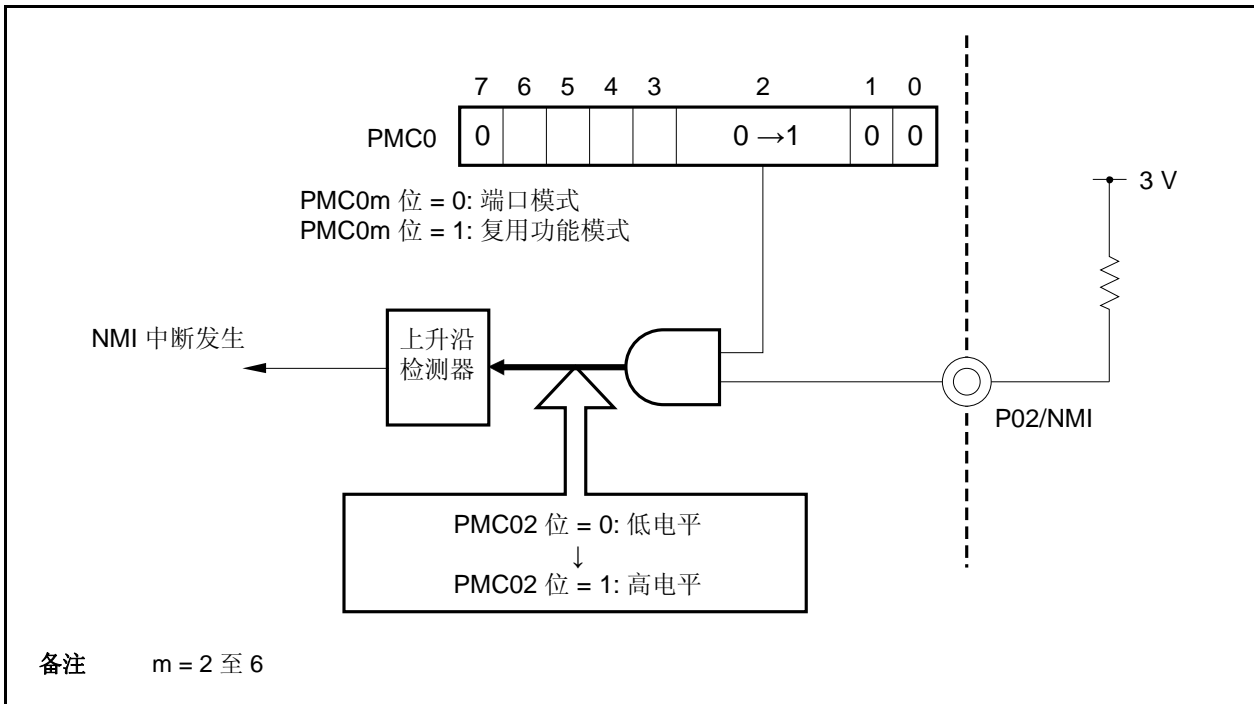
下面的例 1 和例 2 是具体举例。

[例 1] 从通用端口 (P02) 转换至外部中断引脚 (NMI)

P02/NMI 引脚上拉如图 4-4，且上升沿指定为 NMI 引脚边沿检测设置时，在 P02 引脚转换至 NMI 引脚 (PMC02 位 = 0 → 1) 期间，即使高电平连续输入至 NMI 引脚，也会被检测认为是上升沿，就像低电平跳变到高电平那样，产生 NMI 中断。

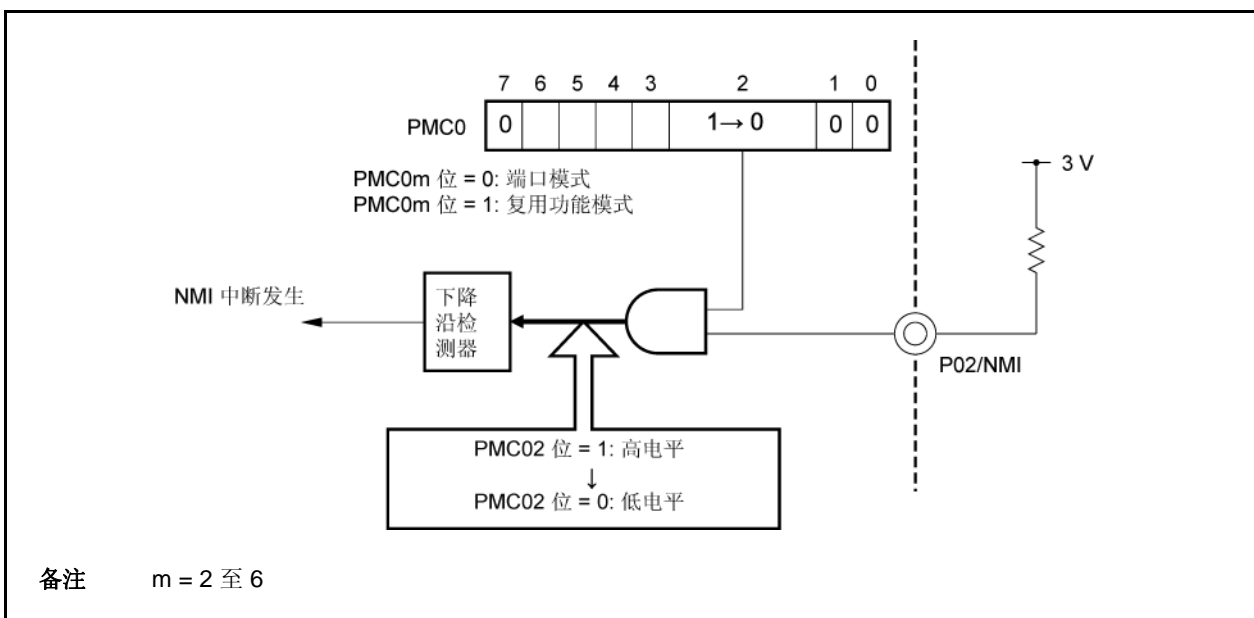
为避免该情况，从 P02 引脚转换至 NMI 功能完成后，再设置 NMI 引脚的有效检测沿。

图 4-4. 从 P02 至 NMI 转换举例（不正确）



[例 2] 从外部引脚（NMI）转换至通用端口（P02）
 P02/NMI 引脚上拉如图 4-5，且上升沿被指定为 NMI 引脚边沿检测设置时，在 NMI 引脚转换至 P02 引脚（PMC02 位 = 1 → 0）期间，即使高电平连续输入至 NMI 引脚，也会被检测认为是下降沿，就像高电平跳变到低电平那样，产生 NMI 中断。
 为避免此情况，在转换至 P02 引脚完成之前，设置 NMI 引脚边沿为“无边沿检测”。

图 4-5. 从 NMI 至 P02 转换举例（不正确）



(2) 端口模式下，PFn.PFn_m 位仅在输出模式（PMn.PMn_m 位 = 0）下有效。输入模式（PMn_m 位 = 1）下，缓冲器中不能反映 PFn_m 位的值。

4.5.2 端口n寄存器 (Pn) 位操作指令的注意事项

在同时具有输入和输出功能的端口执行位操作指令时，除了正常写入目标位外，还可能将该值写入其它输入端口位的输出锁存器。

所以，当一个端口由输入模式转变为输出模式时，建议重写输出锁存器。

<举例> 当 P90 为一个输出端口，同时，P91 到 P97 作为输入端口（所有引脚状态为高电平），端口锁存器的值为 00H。如果 P90 引脚的输出由于位操作指令的执行由低电平变为高电平，则端口锁存器的值为 FFH。

解释说明：向/从一个端口的 Pn 寄存器写入/读取的目标是不同的，PMnm 位为 1 的写入目标是输出锁存器，读取目标是引脚状态

V850ES/JG3-U 和 V850ES/JH3-U 单片机按以下顺序执行位处理指令：

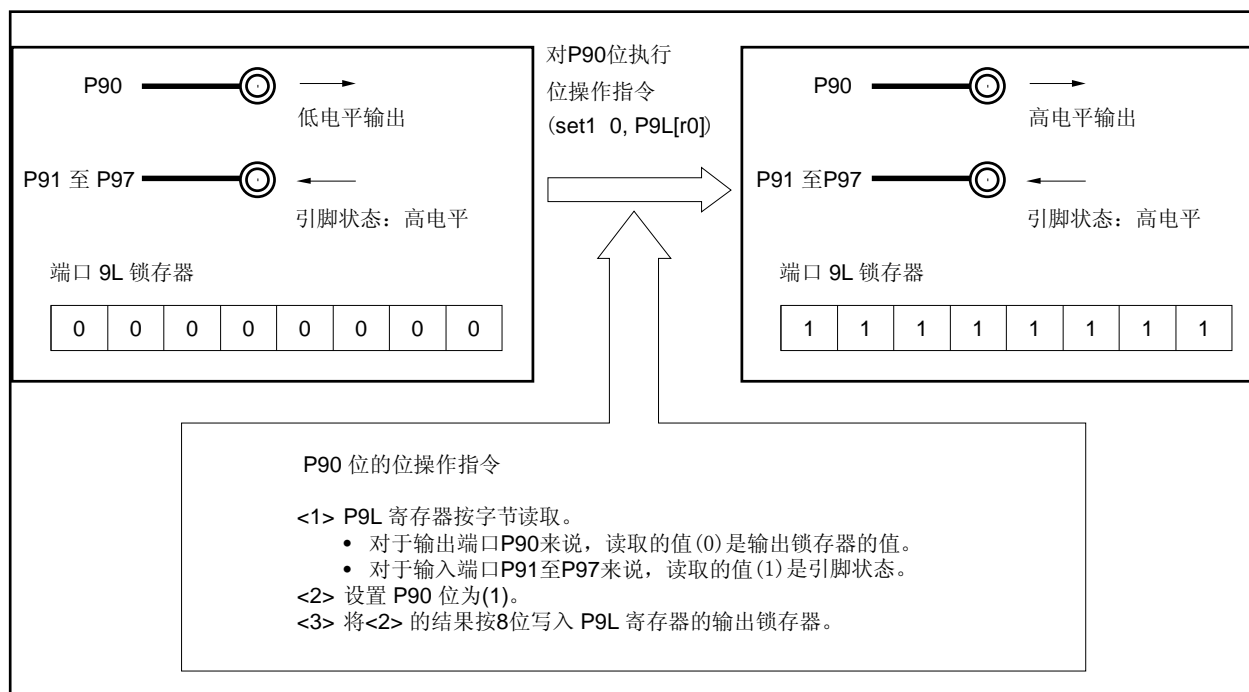
- <1> 寄存器 Pn 按 8 位读取。
- <2> 操作目标位。
- <3> 按 8 位写入寄存器 Pn。

在步骤<1>中，作为输出端口的 P90 引脚的输出锁存器 (O) 的值被读取时，也读到了作为输入端口的 P91 至 P97 引脚状态。如果此时 P91 至 P97 引脚的状态为高电平，则读取的值为 FEH。

在步骤 <2>时，位操作过程使该值变为 FFH。

在步骤 <3>时，位操作将 FFH 写入输出锁存器。

图 4-6. 位处理指令 (P90 引脚)



4.5.3 片上调试引脚的注意事项（仅限V850ES/JG3-U）

$\overline{\text{DRST}}$, DCK, DMS, DDI 和 DDO 引脚是片上调试引脚。

由 $\overline{\text{RESET}}$ 引脚引发的复位之后, P56/INTP05/ $\overline{\text{DRST}}$ 引脚被初始化作为片上调试引脚 ($\overline{\text{DRST}}$)。如果此时有高电平输入 $\overline{\text{DRST}}$ 引脚, 则进入片上调试功能, 就可以使用 DCK, DMS, DDI 和 DDO 引脚。

如果不使用片上调试功能, 必须进行以下操作。

- 将 OCDM 寄存器（特殊寄存器）的 OCDM0 位清 0。

此时, 从 $\overline{\text{RESET}}$ 引脚的复位解除直到上述操作完成, 应使 P56/INTP05/ $\overline{\text{DRST}}$ 引脚固定为低电平。

如果上述动作之前有高电平输入 $\overline{\text{DRST}}$ 引脚, 将会引起一个错误操作（CPU 死锁）。操作 P56 引脚时必须格外小心。

注意事项 其它复位情况, 比如 WDT2RES 信号, 时钟监视器（CLM）, 或低压检测器（LVI）引发复位之后, P56/INTP05/ $\overline{\text{DRST}}$ 引脚并不会初始化作为片上调试功能引脚 ($\overline{\text{DRST}}$)。OCDM 寄存器保持当前值。

4.5.4 P56/INTP05/ $\overline{\text{DRST}}$ 引脚的注意事项

P56/INTP05/ $\overline{\text{DRST}}$ 引脚具有内部下拉电阻（典型值 30 k Ω ）。在 $\overline{\text{RESET}}$ 引脚引发复位之后, 下拉电阻被连接。当 OCDM0 位清除为 0 时, 下拉电阻被断开。

4.5.5 电源开启时P10, P11 和 P53 引脚的注意事项

电源开启后的复位期间, 下列引脚可能暂时输出不确定电平。

- P10/ANO0 引脚
- P11/ANO1 引脚
- P53/SIF2/TIAB00/KR3/TOAB00/RTP03/DDO 引脚（仅限 V850ES/JG3-U）

4.5.6 迟滞特性

在端口模式下, 以下端口引脚没有迟滞特性。

P00 至 P05
 P20 至 P25
 P30 至 P37
 P40 至 P42
 P50 至 P56
 P60 至 P65
 P90 至 P915

第五章 总线控制功能

V850ES/JG3-U 和 V850ES/JH3-U 提供外部总线接口，通过该接口可以连接外部存储器设备，诸如 ROM、RAM 和 I/O 设备等。

5.1 特性

- 可以选择最小 3 个总线周期的复用总线输出，或分离总线输出
(仅限 V850ES/JH3-U; V850ES/JG3-U 仅支持复用总线。)
- 8 位/16 位数据总线可选
- 等待功能：
 - 最高可达 7 个周期的可编程等待功能
 - 使用 $\overline{\text{WAIT}}$ 引脚的外部等待功能
- 空闲状态功能
- 总线保持功能

5.2 总线控制引脚

用于连接外部设备的引脚列举如下表：

表 5-1. V850ES/JH3-U 总线控制引脚（复用总线）

总线控制引脚	复用功能引脚	输入/输出	功能
AD0 至 AD15	PDL0 至 PDL15	I/O	地址/数据总线
A16 至 A23	PDH0 至 PDH5	输出	地址总线
$\overline{\text{WAIT}}$	PCM0	输入	外部等待控制
$\overline{\text{CLKOUT}}$	PCM1	输出	内部系统时钟
$\overline{\text{WR0}}, \overline{\text{WR1}}$	PCT0, PCT1	输出	写选通信号
$\overline{\text{RD}}$	PCT4	输出	读选通信号
ASTB	PCT6	输出	地址选通信号
$\overline{\text{HLDRQ}}$	PCM3	输入	总线保持控制
$\overline{\text{HLDK}}$	PCM2	输出	
$\overline{\text{CS0}}, \overline{\text{CS2}}, \overline{\text{CS3}}$	PCS0, PCS2, PCS3	输出	片选

表 5-2. V850ES/JH3-U 总线控制引脚（分离总线）

总线控制引脚	复用功能引脚	输入/输出	功能
AD0 至 AD15	PDL0 至 PDL15	I/O	数据总线
A0 至 A15	P90 至 P915	输出	地址总线
A16 至 A21	PDH0 至 PDH5	输出	地址总线
$\overline{\text{WAIT}}$	PCM0	输入	外部等待控制
$\overline{\text{CLKOUT}}$	PCM1	输出	内部系统时钟
$\overline{\text{WR0}}, \overline{\text{WR1}}$	PCT0, PCT1	输出	写选通信号
$\overline{\text{RD}}$	PCT4	输出	读选通信号
$\overline{\text{HLDRQ}}$	PCM3	输入	总线保持控制
$\overline{\text{HLDK}}$	PCM2	输出	
$\overline{\text{CS0}}, \overline{\text{CS2}}, \overline{\text{CS3}}$	PCS0, PCS2, PCS3	输出	片选

表 5-3. V850ES/JG3-U 总线控制引脚（复用总线）

总线控制引脚	复用功能引脚	输入/输出	功能
AD0 至 AD15	PDL0 至 PDL15	I/O	地址/数据总线
$\overline{\text{WAIT}}$	PCM0	输入	外部等待控制
$\overline{\text{CLKOUT}}$	PCM1	输出	内部系统时钟
$\overline{\text{WR0}}, \overline{\text{WR1}}$	PCT0, PCT1	输出	写选通信号
$\overline{\text{RD}}$	P61	输出	读选通信号
ASTB	P62	输出	地址选通信号
$\overline{\text{CS0}}, \overline{\text{CS2}}, \overline{\text{CS3}}$	P63, P64, P65	输出	片选

5.2.1 访问片内ROM存储器、片内RAM存储器或片上外设输入/输出端口时的引脚状态

当访问片内 ROM 存储器、片内 RAM 存储器或片上外设输入/输出端口时，引脚状态如下：

表 5-4. 当访问片内 ROM 存储器、片内 RAM 存储器或片上外设输入/输出端口时的引脚状态

总线控制引脚	分离总线模式		复用总线模式	
	内部 ROM/RAM	外设 I/O	内部 ROM/RAM	外设 I/O
地址/数据总线(AD15 至 AD0)	不确定	不确定	不确定	不确定
地址总线(A23 至 A16)	不确定	不确定（访问期间地址输出）	不确定	不确定（访问期间地址输出）
地址总线(A15 至 A0)	不确定	不确定（访问期间地址输出）	不确定	不确定（访问期间地址输出）
控制信号	无效	无效	无效	无效

注意事项 当对内部 ROM 区域进行写入时，地址总线、数据总线和控制总线的信号都激活，和访问外部存储器区域的激活方式相同。

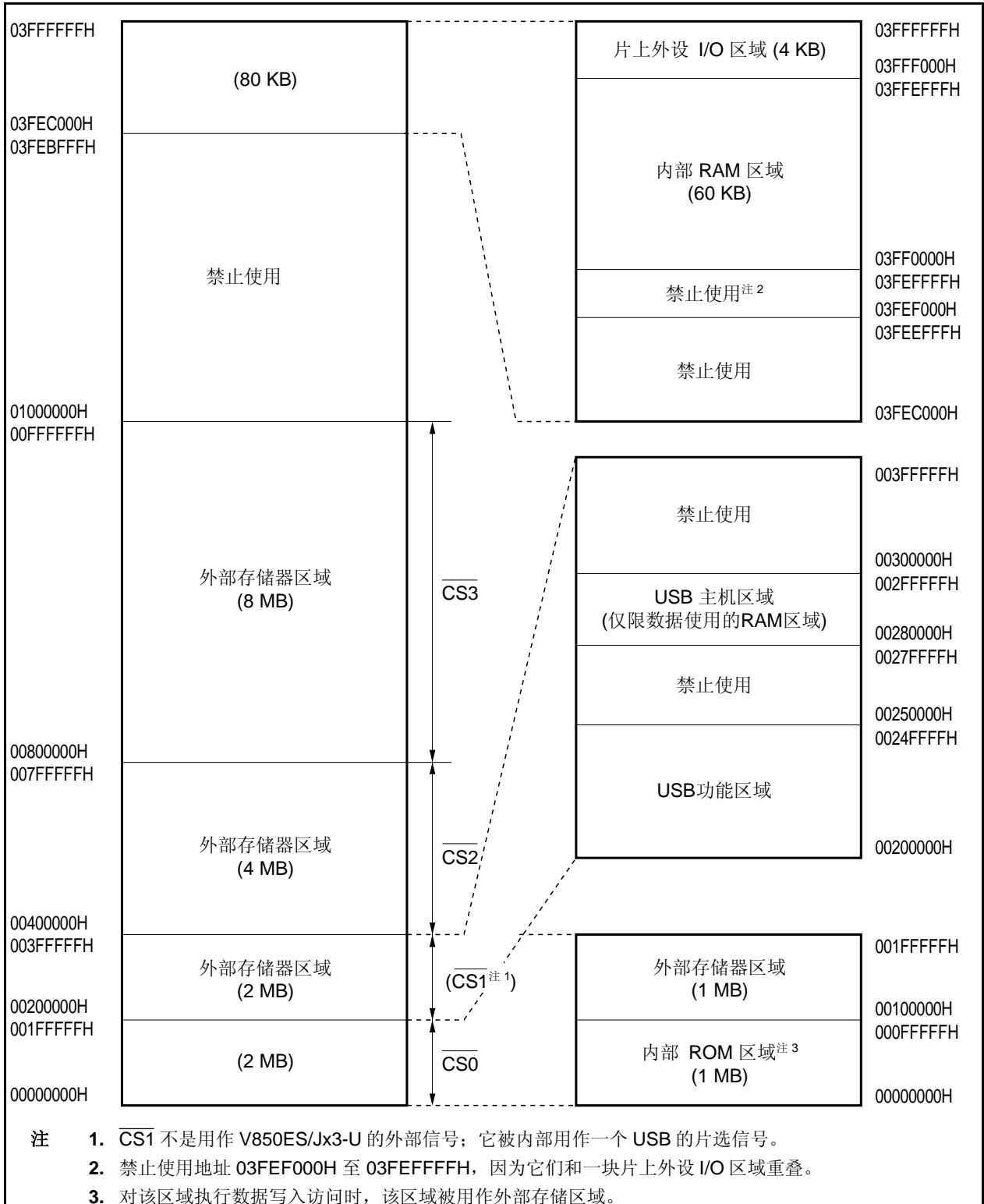
5.2.2 各种工作模式下的引脚状态

关于 V850ES/JG3-U 和 V850ES/JH3-U 单片机在各种工作模式下的引脚状态，参见 2.2 引脚状态。

5.3 存储模块功能

16 MB 外部存储空间从存储器空间的低地址开始被划分成 2 MB、4 MB 和 8 MB 的存储器块。对这些块中每一个块的可编程等待功能和总线周期操作模式都可以进行独立控制，该控制在一个块（One-Block）单元中完成。

图 5-1. 数据存储映射：物理地址



5.4.1 总线访问时钟数目

访问各个存储资源所需的基本时钟数目如下表：

总线周期类型 \ 区域 (总线宽度)	内部 ROM (32 位)	内部 RAM (32 位)	外部存储器 (16 位)	
			复用	分离 ^{注1}
取指周期 (常规访问)	1	1 ^{注2}	3 + n	
取指周期 (分支跳转)	2	2 ^{注1}	3 + n	
操作数访问	5	1	3 + n	

- 注
1. 仅限 V850ES/JH3-U。
 2. 如果与数据访问发生冲突则加 1。

备注 单位：时钟数/访问

5.4.2 总线宽度设置功能

由存储模块 \overline{CSn} 选择的每一个外部存储区都可以通过 BSC 寄存器设置。然而，总线宽度只可以设置为 8 位和 16 位。

V850ES/JG3-U 和 V850ES/JH3-U 单片机的外部存储区由存储模块 $\overline{CS0}$ ， $\overline{CS2}$ 和 $\overline{CS3}$ 选择。

(1) 总线宽度设置寄存器 (BSC)

BSC 寄存器可以按 16 位宽度进行读写。
系统复位后，该寄存器的值为 5555H。

注意事项 复位完成后对 BSC 寄存器执行写操作，之后不要再改变该设定值。并且，在 BSC 寄存器初始设置完成之前，不要访问外部存储器。

复位后: 5555H R/W 地址: FFFFF066H

	15	14	13	12	11	10	9	8
BSC	0	1	0	1	0	1	0	1
	7	6	5	4	3	2	1	0
	0	BS30	0	BS20	0	1	0	BS00
		$\overline{CS3}$		$\overline{CS2}$				$\overline{CS0}$

BSn0	存储模块 CSn 空间的数据总线宽度 (n = 0, 2, 3)
0	8 位
1	16 位

注意事项 请确保将第 14、12、10、8 位和第 2 位设置为“1”，并将第 15、13、11、9、7、5、3 位和第 1 位清除为“0”。

5.4.3 根据不同总线宽度进行访问

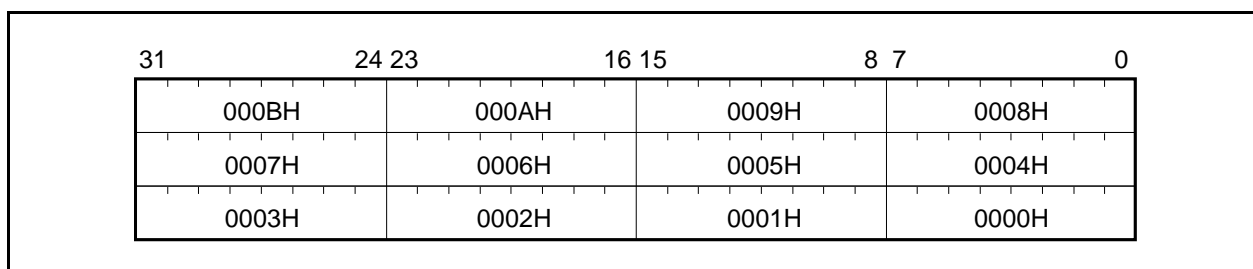
V850ES/JG3-U 和 V850ES/JH3-U 单片机按照 8 位、16 位或 32 位总线宽度访问片上外设输入/输出端口和外部存储器。总线宽度说明如下：

- 片上外设输入/输出端口的总线宽度固定为 16 位。
- 外部存储器的总线宽度可选择 8 位或 16 位（通过 BSC 寄存器选择）。

以上各种访问时的操作描述如下。所有数据访问都从低位开始。

V850ES/JG3-U 和 V850ES/JH3-U 单片机仅支持“小端”数据格式。

图 5-2. 字中的“小端”地址



(1) 数据空间

V850ES/JG3-U 和 V850ES/JH3-U 单片机具有地址不对齐功能。

基于该功能，数据可以放在任何地址，无论该数据是何种格式（字或半字）。然而，如果字数据或半字数据没有在边界对齐，那么，一个总线访问周期至少产生两次，导致总线效率降低。

(a) 访问半字长度数据（16 位）

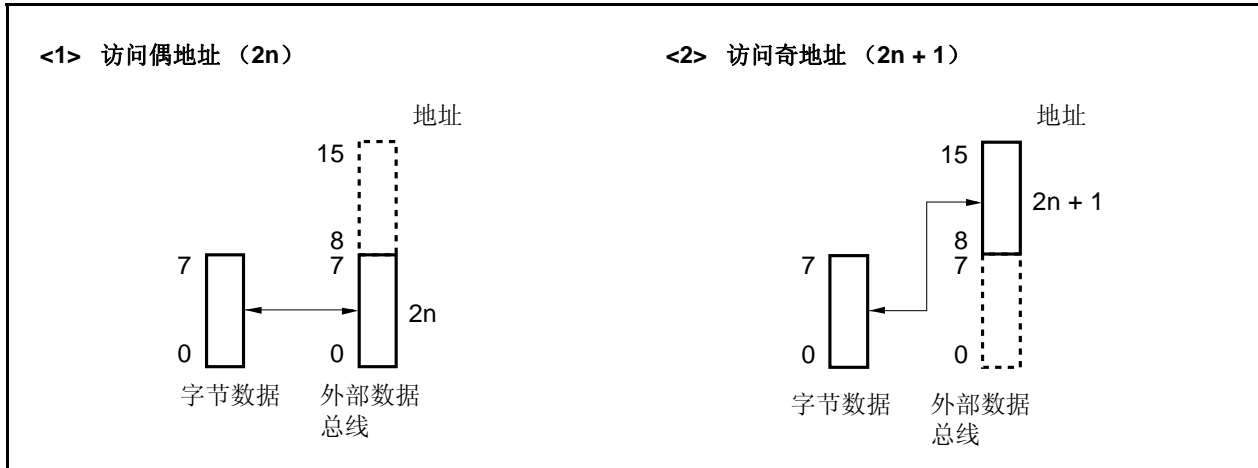
如果地址的最低有效位为 1，一个字节长度的总线访问周期需要产生两次。

(b) 访问字长度数据（32 位）

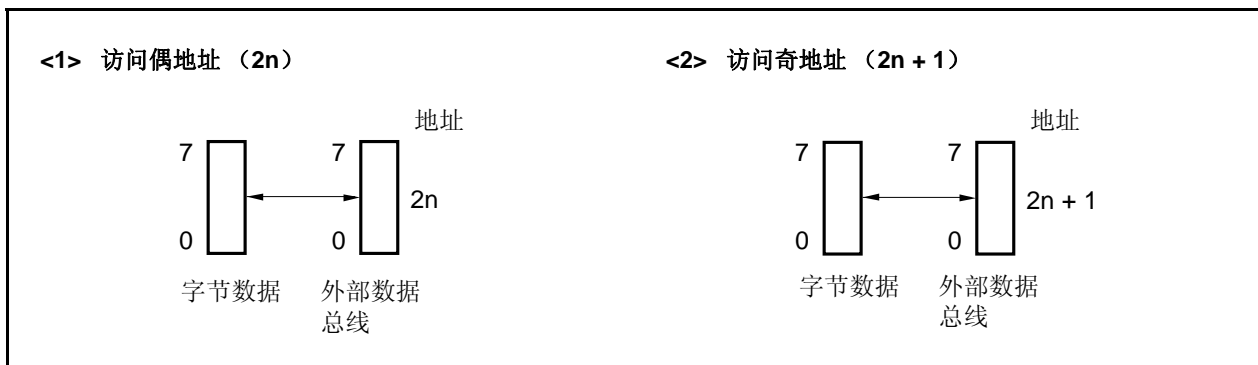
- (i) 如果地址的最低有效位为 1，所需总线访问周期次序：一个字节访问周期、一个半字访问周期、一个字节访问周期。
- (ii) 如果地址的最低两个有效位为 10，需要产生两次半字长度的总线访问周期。

(2) 字节访问 (8 位)

(a) 16 位数据总线宽度

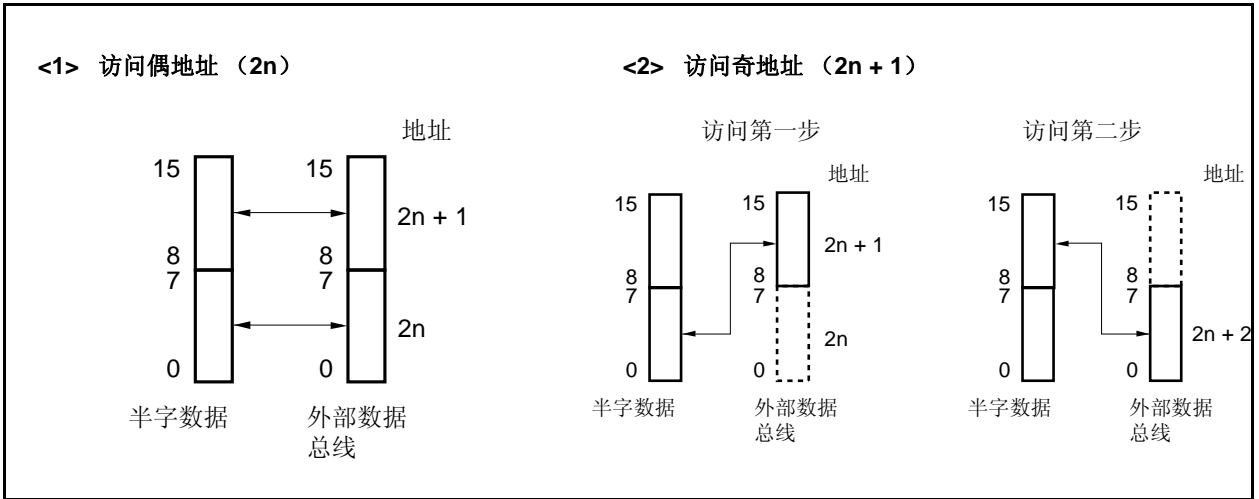


(b) 8 位数据总线宽度

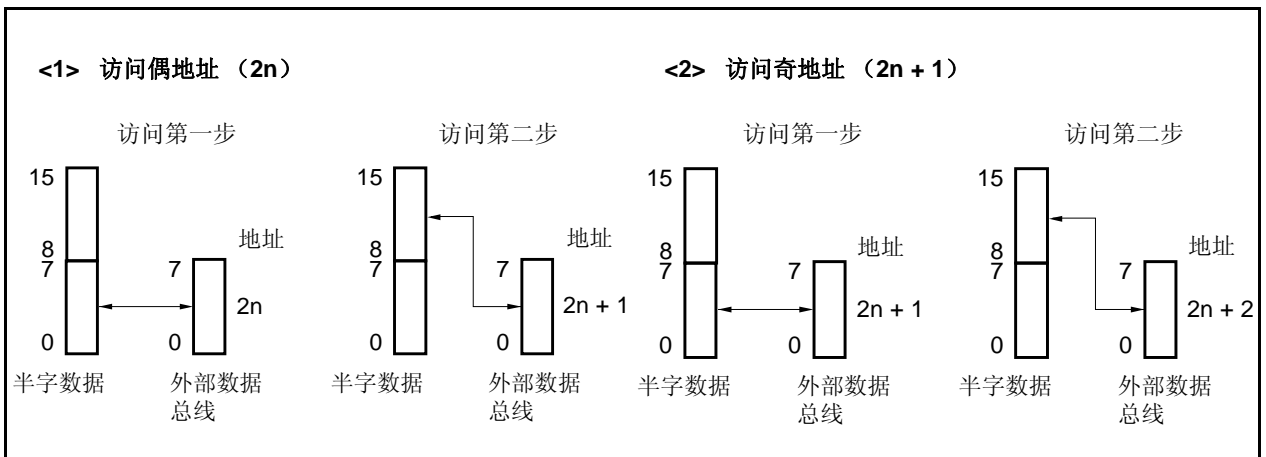


(3) 半字访问 (16 位)

(a) 16 位数据总线宽度



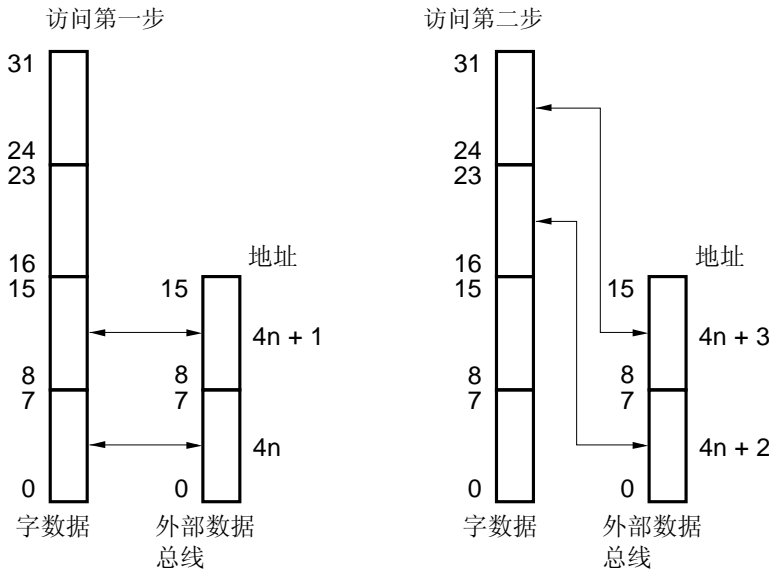
(b) 8 位数据总线宽度



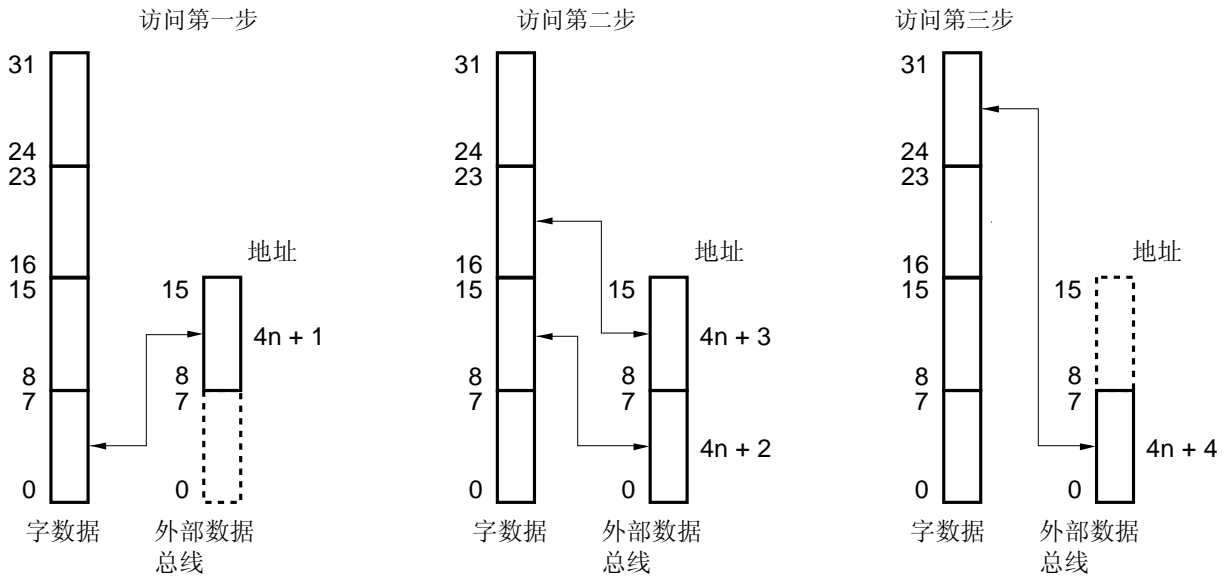
(4) 字访问 (32 位)

(a) 16 位数据总线宽度 (1/2)

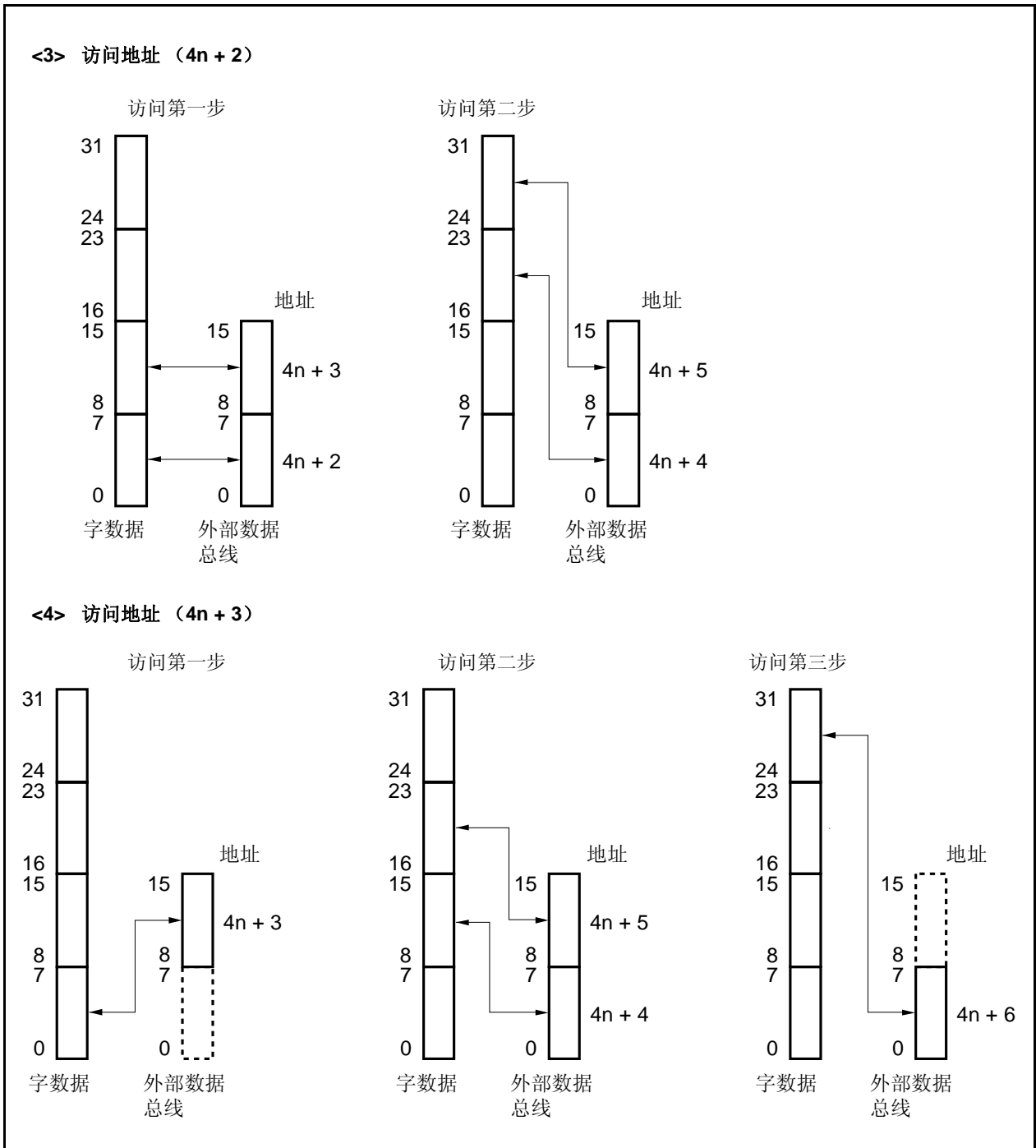
<1> 访问地址 (4n)



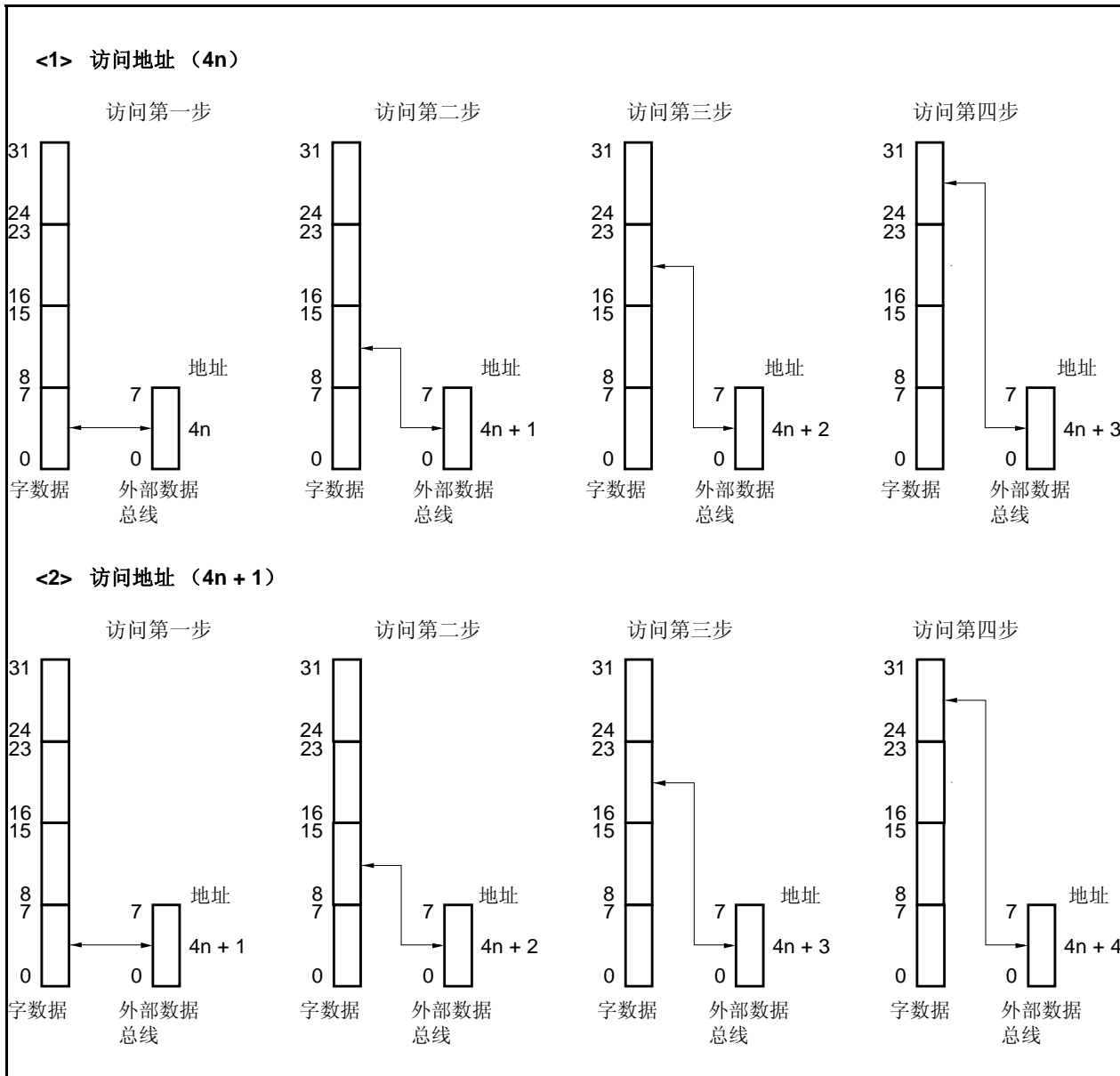
<2> 访问地址 (4n + 1)



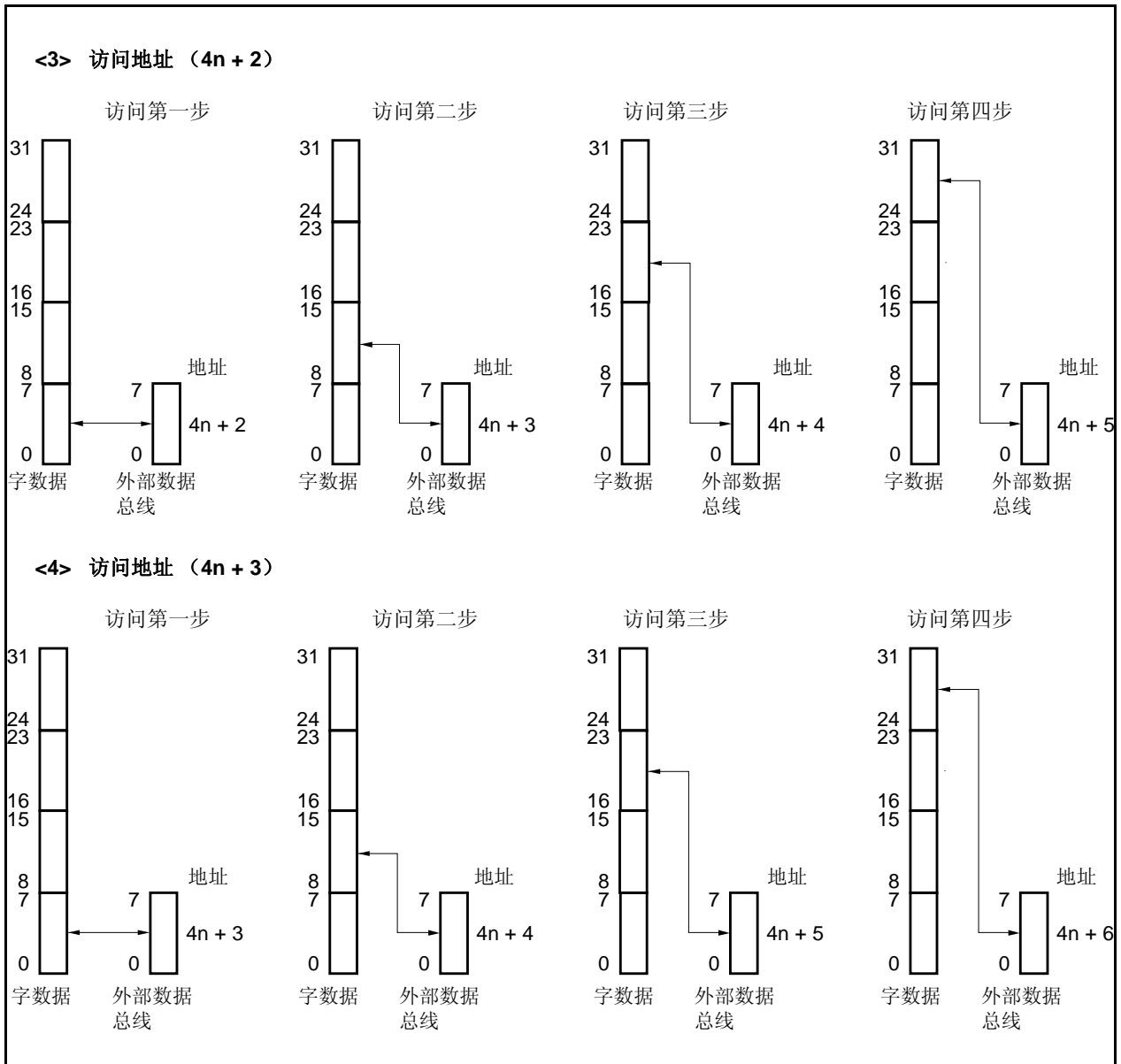
(a) 16 位数据总线宽度 (2/2)



(b) 8 位数据总线宽度 (1/2)



(b) 8 位数据总线宽度 (2/2)



5.5 等待功能

5.5.1 可编程等待功能

(1) 数据等待控制寄存器 0 (DWC0)

为了和低速存储器或者和 I/O 端口实现连接，访问每个 CS 空间时，都可以在其总线周期内插入多达 7 个数据等待状态。

等待状态的数量可以使用 DWC0 寄存器编程。系统复位后，访问所有存储块的总线周期都立即插入 7 个数据等待状态。

DWC0 寄存器可以按 16 位宽度进行读写。

系统复位后，DWC0 寄存器被置为 7777H。

- 注意事项**
1. 对内部 ROM 和内部 RAM 的访问不需要可编程等待，总是可以无等待访问。片上外设输入/输出端口也不需要可编程等待，而是根据各自外设输入/输出端口的控制功能执行等待。
 2. 系统复位完成后写入 DWC0 寄存器，此后不要再改变 DWC0 寄存器的值。而且，DWC0 寄存器设置完成之前，不要访问外部存储空间。

复位后: 7777H R/W 地址: FFFFF484H

	15	14	13	12	11	10	9	8
DWC0	0	DW32	DW31	DW30	0	DW22	DW21	DW20
		CS3				CS2		
	7	6	5	4	3	2	1	0
	0	1	1	1	0	DW02	DW01	DW00
						CS0		

DWn2	DWn1	DWn0	在CSn空间插入的等待状态数量 (n = 0, 2, 3)
0	0	0	无
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

注意事项 请确保将第 6 位至第 4 位设置位“1”，并确保将第 15，11，7 位和第 3 位设置为“0”。

5.5.2 外部等待功能

为了和外部低速存储器、输入/输出端口或异步设备同步，可以利用外部等待引脚 ($\overline{\text{WAIT}}$) 在总线周期中插入任意数目的等待状态。

当 P60^{#1} 或 PCM0^{#2} 引脚设置为复用功能时，使能外部等待功能。

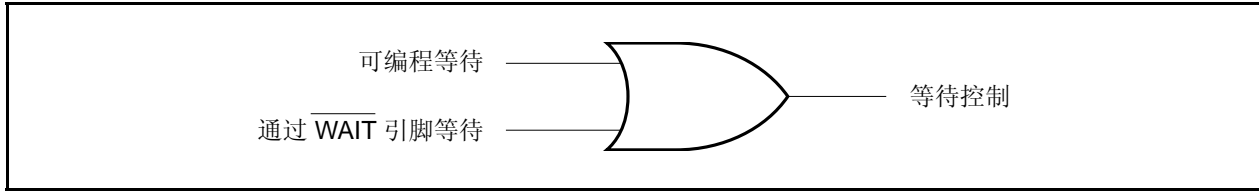
对内部 ROM、内部 RAM 的任何区域以及片上外设输入/输出端口的访问，均不需要外部等待功能，这一点和可编程等待功能一样。

$\overline{\text{WAIT}}$ 信号可以异步输入至 CLKOUT，在复用总线模式下，总线周期的 T2 和 TW 状态的时钟下降沿对该信号采样。在分离总线模式下，在紧邻总线周期的 T2 和 TW 状态之后的时钟上升沿对该信号采样。如果不能满足采样时序的建立/保持时间，在下一状态中插入一个等待状态，或者根本不插入。

- 注
1. V850ES/JG3-U
 2. V850ES/JH3-U

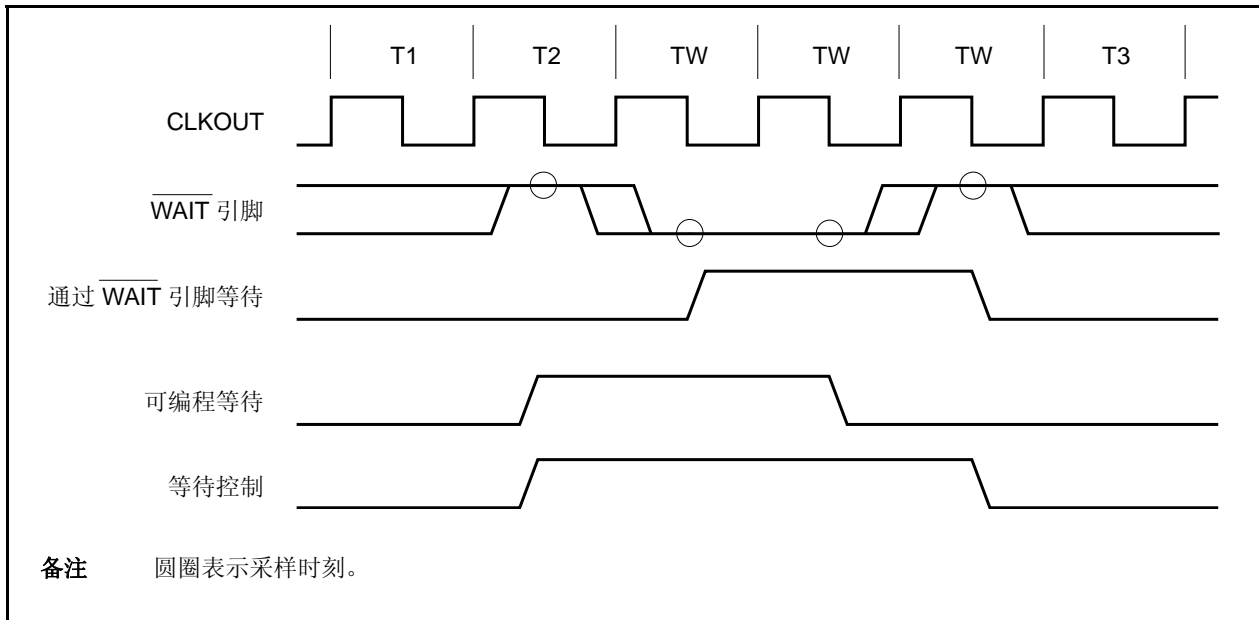
5.5.3 可编程等待和外部等待之间的关系

对可编程等待设定的等待周期和 $\overline{\text{WAIT}}$ 引脚控制的等待周期执行或(OR)操作，其结果是插入的等待周期数目。



例如，假设可编程等待时序和 $\overline{\text{WAIT}}$ 引脚等待时序如下所示，则插入到总线访问周期数目为 3 个等待状态。

图 5-3. 插入等待举例



5.5.4 可编程地址等待功能

通过 AWC 寄存器，可以设置插入到各个总线周期中的地址建立等待或地址保持等待。对每个片选区域 ($\overline{CS0}$, $\overline{CS2}$, $\overline{CS3}$) 都需要设置地址等待插入。

当在总线周期中插入一个地址建立等待时，表现为 T1 状态的高电平时钟被延长了时钟周期。当插入一个地址保持等待时，表现为 T1 状态的低电平时钟被延长了时钟周期。

(1) 地址等待控制寄存器 (AWC)

AWC 寄存器可以按 16 位宽度进行读写。

系统复位后，AWC 寄存器被设置为 FFFFH。

- 注意事项**
1. 访问内部 ROM 区、内部 RAM 区和片上外设输入/输出区时，不需要插入地址建立等待周期和地址保持等待周期。
 2. 系统复位完成后写入 AWC 寄存器，此后不要再改变该设定值。而且，AWC 寄存器初始设置完成之前，不要访问外部存储空间。

复位后: FFFFH		R/W	地址: FFFFF488H					
	15	14	13	12	11	10	9	8
AWC	1	1	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
	AHW3 ASW3		AHW2 ASW2		1	1	AHW0 ASW0	
	$\overline{CS3}$		$\overline{CS2}$		$\overline{CS0}$			

AHWn	地址保持等待的插入规范 (n = 0, 2, 3)	
0	不插入	
1	插入	

ASWn	地址建立等待的插入规范 (n = 0, 2, 3)	
0	不插入	
1	插入	

注意事项 请确保将第 15 至 8 位，第 3 位和第 2 位设置为“1”。

5.6 空闲状态插入功能

为了方便和低速存储器的连接，对片选信号选择的各个空间执行访问时，在总线周期的 T3 状态后插入一个空闲状态（TI）。在读取访问时插入一个空闲状态，可以保证存储器数据输出的浮动延迟时间（写访问期间，不能插入空闲状态）。

通过 BCC 寄存器，可以设置是否插入空闲状态。

系统复位之后，对所有区域的访问时序都会立刻插入一个空闲状态。

(1) 总线周期控制寄存器（BCC）

BCC 寄存器可以按 16 位宽度进行读写。

系统复位后，该寄存器被置为 AAAAH。

- 注意事项**
1. 内部 ROM、内部 RAM、片上外设 I/O 端口等区域的访问不支持空闲插入。
 2. 系统复位完成后写入 BCC 寄存器，此后不要再改变 BCC 寄存器的值。而且，BCC 寄存器设置完成之前，不要访问外部存储空间。

复位后: AAAAH		R/W	地址: FFFFF48AH					
	15	14	13	12	11	10	9	8
BCC	1	0	1	0	1	0	1	0
	7	6	5	4	3	2	1	0
	BC31	0	BC21	0	1	0	BC01	0
	<input type="checkbox"/>		<input type="checkbox"/>				<input type="checkbox"/>	
	CS3		CS2				CS0	
BCn1	空闲状态的插入规范 (n = 0, 2, 3)							
0	不插入							
1	插入							

注意事项 请确保将第 15、13、11、9 位和第 3 位设置为“1”，并将第 14、12、10、8、6、4、2 位和第 0 位清除为“0”。

5.7 总线保持功能（仅限V850ES/JH3-U）

5.7.1 功能概述

如果 PCM2 和 PCM3 引脚被设为复用功能，则 $\overline{\text{HLDRQ}}$ 和 $\overline{\text{HLDK}}$ 功能有效。

当 $\overline{\text{HLDRQ}}$ 引脚有效时（低电平），表示有其它总线主控器已经申请总线控制权，这时，外部地址/数据总线进入高阻状态并被释放（总线保持状态）。如果总线控制权的请求被清除，则 $\overline{\text{HLDRQ}}$ 引脚无效（高电平），引脚驱动工作又一次开始。

在总线保持期间，执行内部 ROM 和内部 RAM 中的程序，直到访问片上外设输入/输出寄存器或访问外部存储器为止。

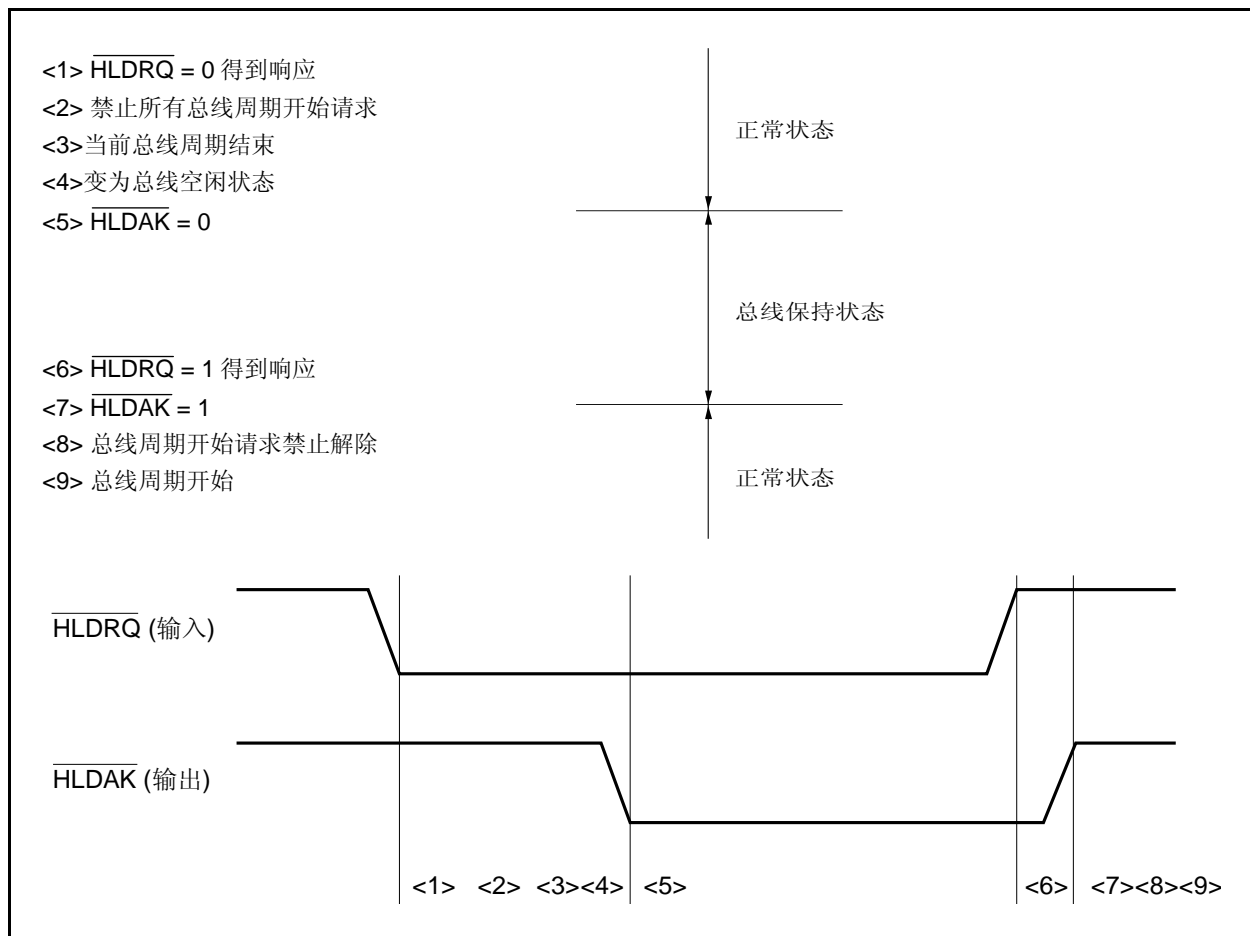
总线保持状态通过 $\overline{\text{HLDK}}$ 引脚的有效（低电平）来表示。总线保持功能允许多处理器系统的配置，其中存在两个或更多个总线主控器。

请注意，在由总线宽度功能或位操作指令发起的多重访问循环中，总线保持请求将得不到响应。

状态	数据总线宽度	访问类型	总线保持请求无响应的时间
CPU 总线锁定	16 位	字访问偶地址	第一步和第二步访问之间
		字访问奇地址	第一步和第二步访问之间 第二步和第三步访问之间
		半字访问奇地址	第一步和第二步访问之间
	8 位	字访问	第一步和第二步访问之间 第二步和第三步访问之间 第三步和第四步访问之间
		半字访问	第一步和第二步访问之间
		位操作指令的读-修改-写访问	-

5.7.2 总线保持步骤

总线保持状态转变步骤如下所示：



5.7.3 节电模式时的操作

由于在 STOP、IDLE1 和 IDLE2 模式下内部系统时钟停止工作，所以即使 $\overline{\text{HLDARQ}}$ 引脚有效，也不会进入总线保持状态。

HALT 模式下， $\overline{\text{HLDARQ}}$ 引脚和 $\overline{\text{HLDARQ}}$ 引脚都变为低电平（即有效状态），会进入总线保持状态。当 $\overline{\text{HLDARQ}}$ 引脚随后变为无效状态后， $\overline{\text{HLDARQ}}$ 引脚也变为无效状态，则退出总线保持状态。

5.8 总线优先权

总线保持、DMA 传输、运算对象数据访问、取指令（分支跳转）、取指令（连续）等操作在外部总线周期中执行。总线保持具有最高的优先级，接着是 DMA 传输、运算对象数据访问、取指令（分支跳转）、取指令（顺序取值）。在读-修改-写访问中，读访问和写访问之间可以插入一个取指令操作。

如果一条指令执行需要进行两次或两次以上的数据访问时，由于总线长度的限制，在数据访问期间不插入取指令操作和总线保持状态。

表 5-5. 总线优先级

优先级	外部总线周期	总线主控器
高  低	总线保持	外部设备
	DMA 传输	DMAC
	操作数访问	CPU
	取指周期（分支跳转）	CPU
	取指周期（顺序取指）	CPU

5.9 总线时序

图 5-4. 复用/分离总线读时序 (总线宽度: 16 位, 16 位存取访问)

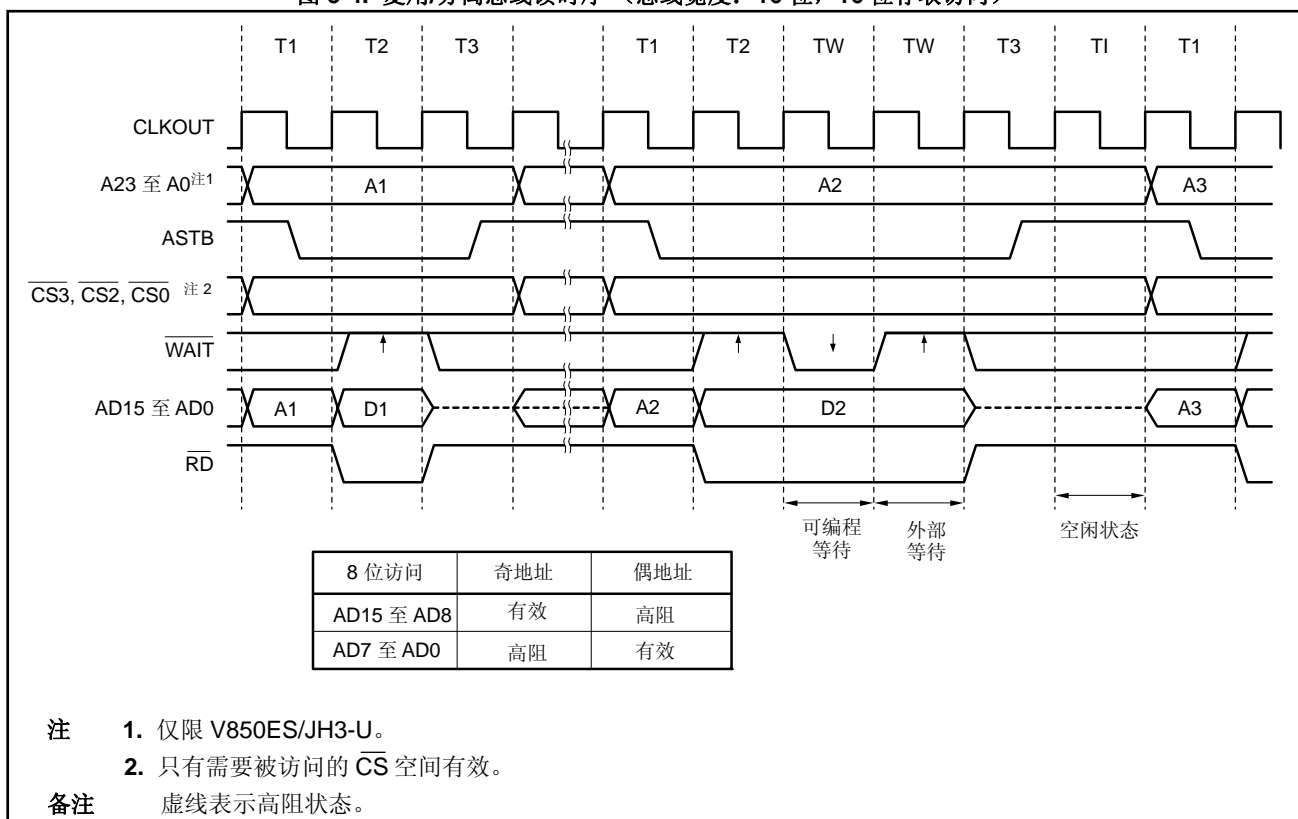


图 5-5. 复用/分离总线读时序 (总线宽度: 8 位)

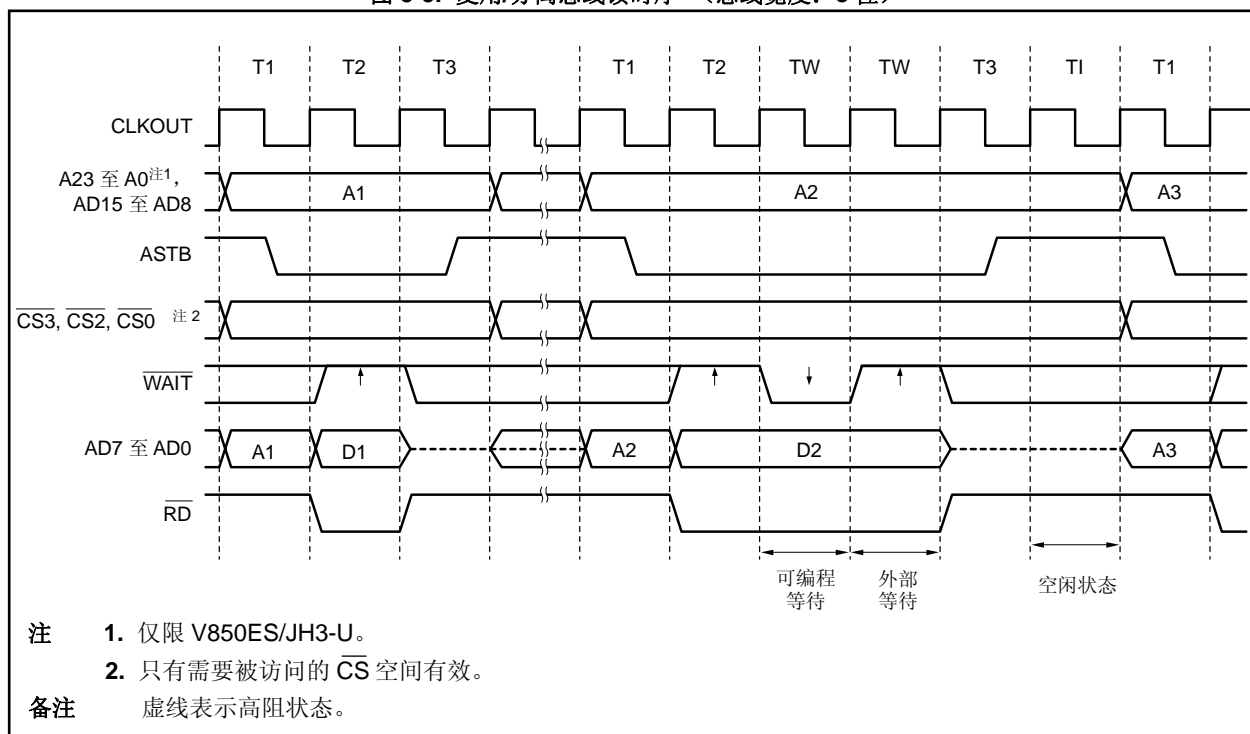


图 5-6. 复用/分离总线写入时序（总线宽度：16 位，16 位存取访问）

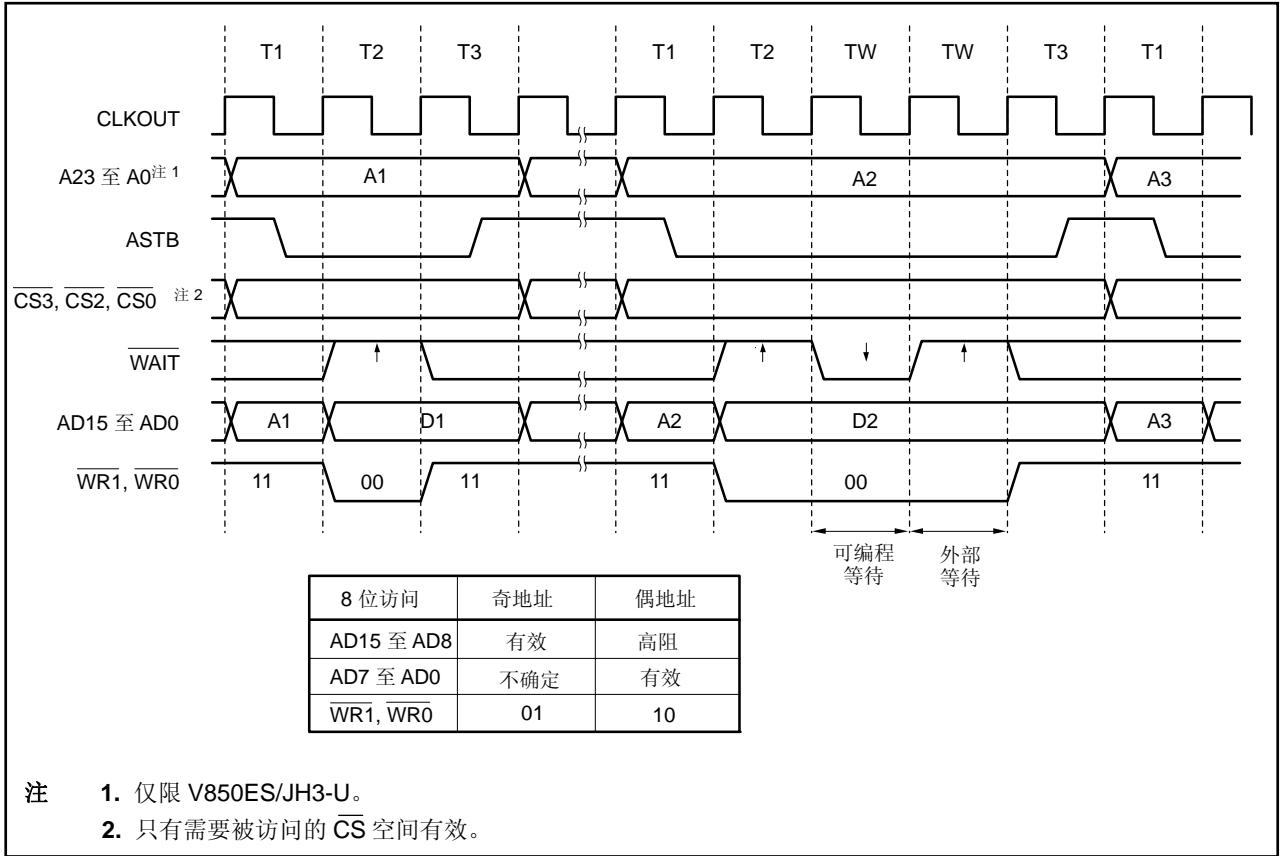


图 5-7. 复用/分离总线写入时序（总线宽度：8 位）

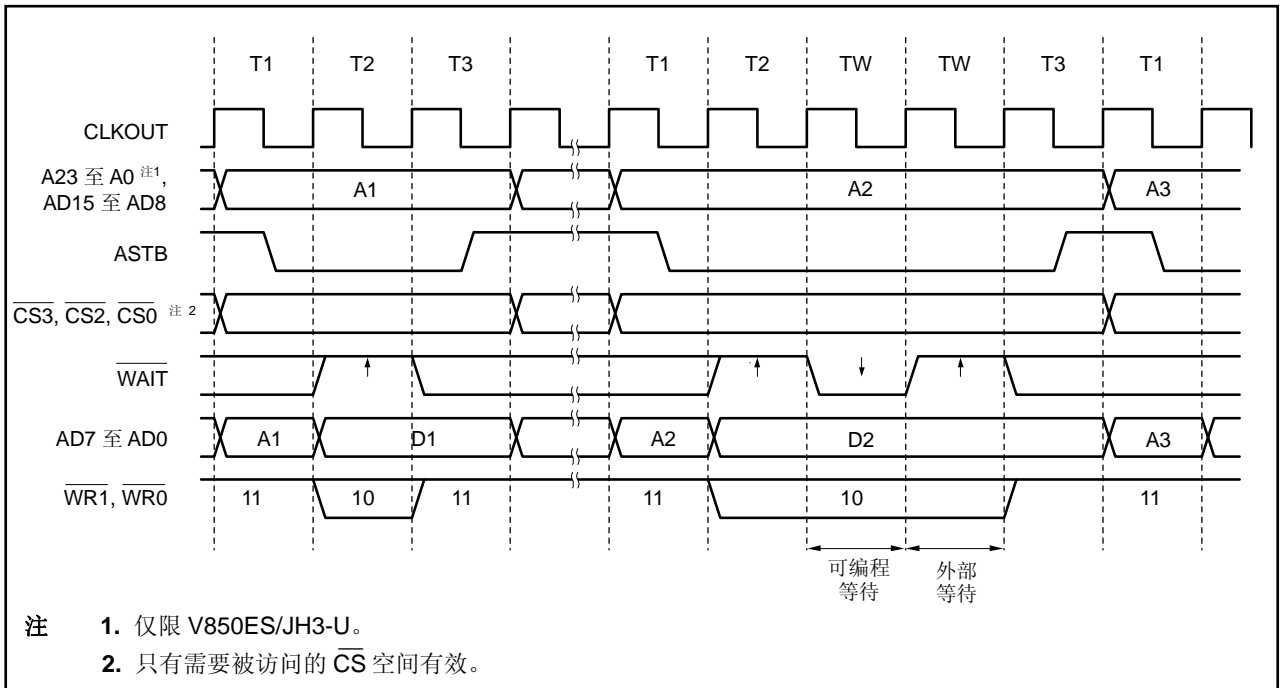
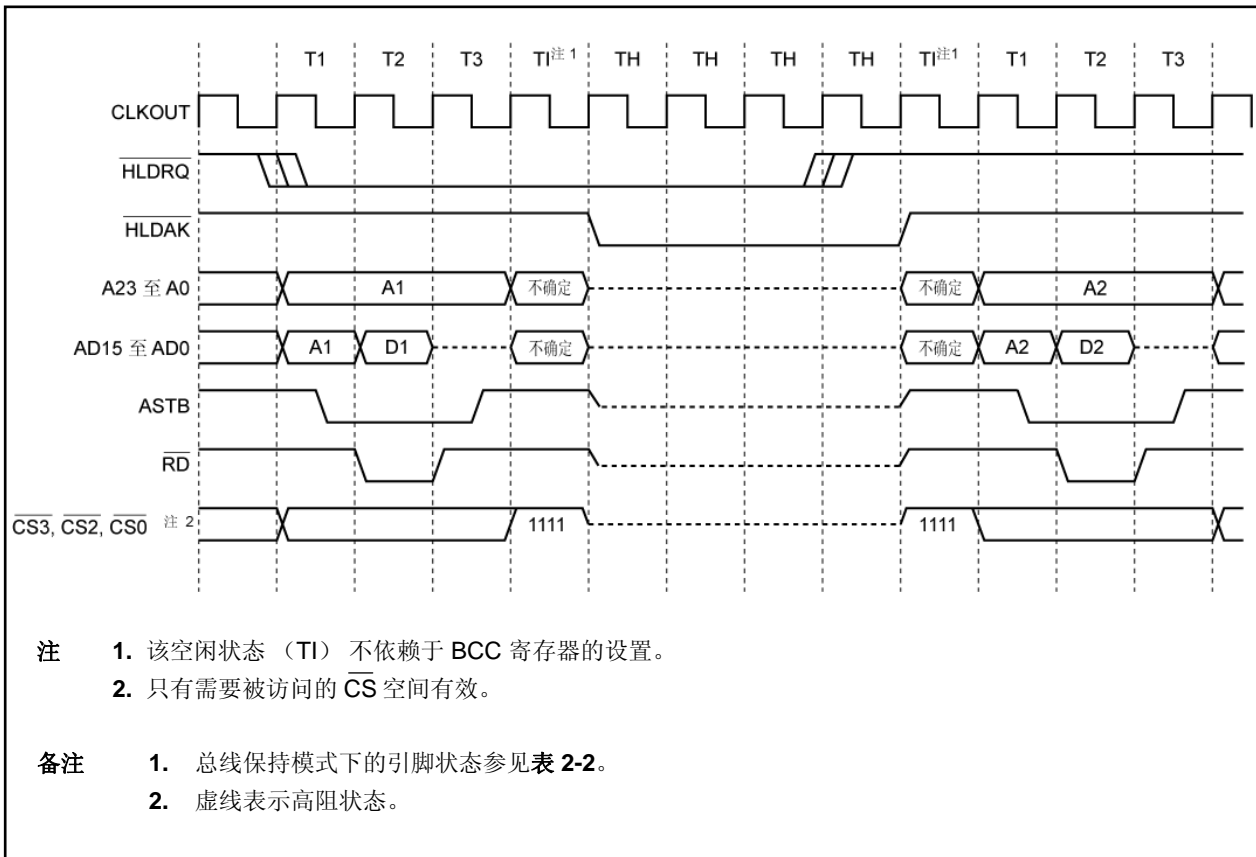


图 5-8. 复用/分离总线保持时序（总线宽度：16 位，16 位存取访问）（仅限 V850ES/JH3-U）



6.1 概述

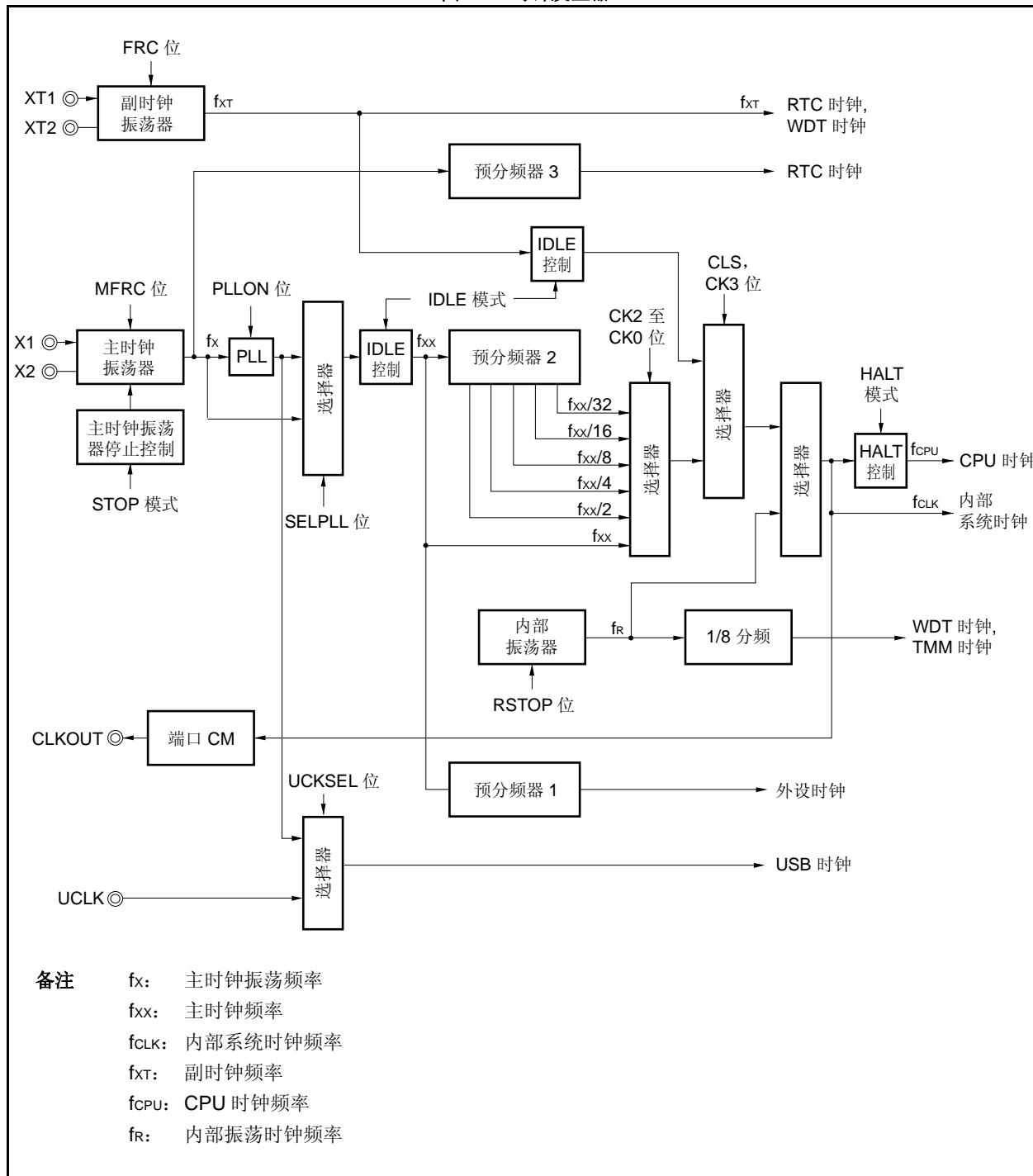
具备以下时钟发生功能：

- 主时钟振荡器
 - 时钟直通模式
 $f_x = 3.0$ 至 6.0 MHz ($f_{xx} = 3.0$ 至 6.0 MHz)
 - PLL 模式
 $f_x = 3.0$ 至 6.0 MHz ($\times 8$: $f_{xx} = 24$ 至 48 MHz)
- 副时钟振荡器
 - $f_{XT} = 32.768$ kHz
- PLL (锁相环) 倍频 ($\times 8$) 功能
 - 时钟直通模式/PLL 模式可选
- 内部振荡器
 - $f_R = 220$ kHz (典型值)
- 内部系统时钟的产生
 - 7 个级别 (f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/8$, $f_{xx}/16$, $f_{xx}/32$, f_{XT})
- 外设时钟产生功能
- 时钟输出功能

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

6.2 配置

图 6-1. 时钟发生器



(1) 主时钟振荡器

主时钟振荡器振荡产生以下频率 (f_x)。

- 时钟直通模式
 $f_x = 3.0$ 至 6.0 MHz
- PLL 模式
 $f_x = 3.0$ 至 6.0 MHz ($\times 8$)

(2) 副时钟振荡器

副时钟振荡器振荡于 32.768 kHz (f_{XT}) 频率。

(3) 主时钟振荡器停止控制

该电路产生一个控制信号，可以使主时钟振荡器停止工作。

系统处于 STOP 模式时，或者当 PCC.MCK 位 = 1 时（只有在 PCC.CLS 位 = 1 时有效），主时钟振荡器停止工作。

(4) 内部振荡器

以 220 kHz（典型值）频率振荡。

(5) 预分频器 1

该预分频器产生 (f_{xx} 至 $f_{xx} / 1,024$) 频率的时钟，用来供给以下片上外设工作：TAA, TAB, TMM, TMT, CSIF, UARTC, I²C, ADC, DAC, 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 产生的时钟 (f_{xx} 至 $f_{xx} / 32$) 提供给一个选择器，产生 CPU 时钟 (f_{CPU}) 和内部系统时钟 (f_{CLK})。

f_{CLK} 时钟提供给 INTC、ROM 和 RAM 模块，还可以从 CLKOUT 引脚输出。

(7) 预分频器 3

该电路通过对主时钟振荡器产生的时钟 (f_x) 进行分频，得到特定频率 (32.768 kHz) 的时钟，并将该时钟供给实时计数器模块。

(8) PLL

本电路对主时钟振荡器产生的时钟 (f_x) 进行 8 倍频。

有两种操作模式：时钟直通模式， f_x 直接输出；锁相环模式，倍频时钟输出。可以通过 PLLCTL.SELPLL 位设置不同的操作模式。

6.3 寄存器

(1) 处理器时钟控制寄存器 (PCC)

处理器时钟控制寄存器 PCC 是一个特殊寄存器，只有通过特定的流程才能将数据写入该寄存器（见 3.4.7 特殊寄存器）。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被置为 03H。

复位后: 03H R/W 地址: FFFFF828H

	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	未使用							
MCK	主时钟振荡器控制							
0	允许振荡							
1	停止振荡							
<ul style="list-style-type: none"> 当主时钟作为CPU时钟工作时，即使MCK位设置为(1)，主时钟的操作也不会停止。CPU时钟源切换为副时钟后，主时钟停止操作。 将 MCK 位从0变为1之前，停止所有使用主时钟操作的片上外设功能。 主时钟停止，器件工作于副时钟时，清除 MCK位为(0)并且在CPU时钟转换为主时钟之前，或操作片上外设功能之前，通过软件确保振荡稳定时间。 								
MFRC	主时钟片上反馈电阻的使用							
0	使用 (使用陶瓷/晶体谐振器时)							
1	不使用 (使用外部时钟时)							
CLS ^注	CPU 时钟的状态 (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	x	禁止设置				
1	x	x	x	f _{XT}				

注 CLS 为只读位。

注意事项 1. 当 CLKOUT 正在输出时，不要改变 CPU 时钟（使用 CK3 至 CK0 位）。

2. 使用一个位处理指令来操作 CK3 位。当使用 8 位处理指令时，不要改变 CK2 至 CK0 位的设定值。

备注 x: 无需理会

(a) 设置主时钟运行→副时钟运行的举例：

- <1> CK3 位 ← 1: 建议使用一个位处理指令。不要改变 CK2 至 CK0 位。
- <2> 副时钟操作: 读取 CLS 位, 检查副时钟是否开始工作。在 CK3 位设置完成后需要以下时间, 副时钟才能工作:
最大值: $1/f_{XT}$ (1/副时钟频率)。
- <3> MCK 位 ← 1: 只有在主时钟停止时, 才可以设定 MCK 位为 1。

- 注意事项**
1. 当停止主时钟时, 停止锁相环。同样使用主时钟的片上外设也停止工作。
 2. 如果以下条件不能满足, 需要调整 CK2 至 CK0 位的值来满足条件, 然后切换到副时钟工作模式。

$$\text{内部系统时钟 (fCLK)} > \text{副时钟 (f}_{XT}\text{: 32.768 kHz)} \times 4$$

备注 内部系统时钟 (fCLK): 通过设置 CK2 至 CK0 位, 由主时钟 (f_{xx}) 产生时钟。

[说明示例]

```

_DMA_DISABLE:
  clr 0, DCHCn[r0]      -- DMA 操作禁止。 n = 0 至 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, 直到副时钟开始工作。
  bz   _CHECK_CLS
<3> _STOP_MAIN_CLOCK :
  st.b r0, PRCMD[r0]
  set1 6, PCC[r0]      -- MCK 位 ← 1, 主时钟停止。
_DMA_ENABLE:
  set1 0, DCHCn[r0]    -- DMA 操作使能。 n = 0 至 3

```

备注 以上描述仅为举例。注意<2>部分, 在一个闭合循环中读取 CLS 位。

(b) 设置副时钟工作→主时钟工作的举例：

- <1> MCK 位 ← 0: 主时钟振荡器开始工作。
- <2> 在程序中插入等待指令，保证主时钟振荡器稳定时间。
- <3> CK3 位 ← 0: 建议使用一个位处理指令。不要改变 CK2 至 CK0 位。
- <4> 主时钟工作: 从 CK3 位设置完成后，直到主时钟开始工作，需要以下时间：
 最大值：1/f_{XT}（1/副时钟频率）。
 所以，CK3 位清为 0 后，应紧接着插入一条 NOP 指令，或通过读取 CLS 位来确定主时钟是否开始工作。

注意事项 只有在主时钟稳定振荡后，才允许那些使用主时钟驱动的片上外设功能运行。如果这些操作在振荡稳定时间完成之前就已经使能，则可能发生误操作。

[说明示例]

```

_DMA_DISABLE:
    clr1      0, DCHCn[r0]          -- DMA 操作禁止。 n = 0 至 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]            -- 主时钟开始振荡。
<2> movea   0x55, r0, r11         -- 等待振荡稳定时间。
    _WAIT_OST :
    nop
    nop
    nop
    addi    -1, r11, r11
    cmp     r0, r11
    bne     _WAIT_OST
<3> st.b    r0, PRCMD[r0]
    clr1    3, PCC[r0]             -- CK3 ← 0
<4> _CHECK_CLS :
    tst1    4, PCC[r0]             --等待，直到主时钟开始工作。
    bnz     _CHECK_CLS
_DMA_ENABLE:
    setl    0, DCHCn[r0]          -- DMA 操作使能。 n = 0 至 3

```

备注 以上描述仅为举例。注意<4>部分，在一个闭合循环中读取 CLS 位。

(2) 内部振荡模式寄存器 (RCM)

RCM 寄存器是 8 位寄存器，用于设置内部振荡器的工作模式。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H R/W 地址: FFFFF80CH

	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. 内部振荡时钟提供给 CPU 作为工作时钟源时 (CCLS.CCLS F 位 = 1)，内部振荡器不能被停止。不要将 RSTOP 位设置为 1。
 2. 如果 CCLS.CCLS F 位设置为 1 (振荡稳定期间发生看门狗定时器溢出)，即使 RSTOP 位设置为 1 时，内部振荡器会继续振荡。同时，RSTOP 位保持为 1。

(3) CPU 工作时钟状态寄存器 (CCLS)

CCLS 寄存器表明 CPU 工作时钟的状态。

本寄存器为只读寄存器，可以按字节读取或按位读取。

系统复位后，该寄存器被设为 00H。

复位后: 00H^注 R 地址: FFFFF82EH

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

CCLS F	CPU 工作时钟状态
0	工作于主时钟 (f _x) 或副时钟 (f _{xT})。
1	工作于内部振荡时钟 (f _R)。

注 复位释放后，如果在振荡稳定期间发生看门狗定时器溢出，CCLS F 设置为 1，且复位值为 01H。

6.4 操作

6.4.1 各时钟的操作

下表显示各个时钟的操作状态：

表 6-1. 各时钟的操作状态

寄存器设置和操作状态	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	振荡稳定定时器计数期间	HALT 模式	IDLE1, IDLE2 模式	STOP 模式	副时钟模式	Sub-IDLE 模式	副时钟模式	Sub-IDLE 模式
目标时钟									
主时钟振荡器 (fx)	×	○	○	○	×	○	○	×	×
副时钟振荡器 (fxT)	○	○	○	○	○	○	○	○	○
CPU 时钟 (fcPU)	×	×	×	×	×	○	×	○	×
内部系统时钟 (fCLK)	×	×	○	×	×	○	×	○	×
主时钟 (PLL 模式, fxx)	×	○ ^注	○	×	×	○	○	×	×
外设时钟 (fxx 至 fxx/1,024)	×	×	○	×	×	○	×	×	×
钟表定时器时钟 (主时钟源)	×	○	○	○	×	○	○	×	×
钟表定时器时钟 (副时钟源)	○	○	○	○	○	○	○	○	○
WDT2 时钟 (内部振荡)	×	○	○	○	○	○	○	○	○
WDT2 时钟 (主时钟源)	×	×	○	×	×	○	×	×	×
WDT2 时钟 (副时钟源)	○	○	○	○	○	○	○	○	○

注 入锁时间

备注 ○： 可运行

×： 停止

6.4.2 时钟输出功能

时钟输出功能用于从 CLKOUT 引脚输出内部系统时钟 (fCLK)。

通过 PCC.CK3 至 PCC.CK0 位选择系统内部时钟 (fCLK)。

CLKOUT 引脚功能可以通过端口 CM 控制寄存器选择，复用为 PCM1 引脚或时钟输出引脚。

CLKOUT 引脚的状态与表 6-1 中的内部系统时钟相同，在可操作状态下，此引脚可以输出时钟；在停止状态下，它输出低电平。但是，复位后 CLKOUT 引脚默认为端口模式 (PCM1 引脚：输入模式)，直到该引脚被设置为输出模式。因此，该引脚为高阻状态。

6.5 PLL功能

6.5.1 概述

V850ES/JG3-U 和 V850ES/JH3-U 单片机中，CPU 和片上外设功能的工作时钟可以选择 PLL 功能输出的 8 倍振荡频率，或选择时钟直通模式。

当使用锁相环功能时($\times 8$): 输入时钟 = 3.0 至 6.0 MHz (输出: 24 至 48 MHz)
 时钟直通模式: 输入时钟 = 3.0 至 6.0 MHz (输出: 3.0 至 6.0 MHz)

6.5.2 寄存器

(1) 锁相环控制寄存器 (PLLCTL)

PLLCTL 寄存器是 8 位寄存器，用于控制 PLL 功能。

寄存器可进行字节读写或按位读写。

系统复位后，本寄存器被设置为 01H。

复位后: 01H R/W 地址: FFFFF82CH

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

PLLON	PLL 工作停止寄存器
0	PLL 停止
1	PLL 工作 (PLL 开始工作后，需要一段入锁时间使频率稳定。)

SELPLL	CPU 工作时钟选择寄存器
0	时钟直通模式
1	PLL 模式

- 注意事项**
1. 当 PLLON 位被清 0 时，SELPLL 位自动清除为 0 (时钟直通模式)。
 2. 只有在 PLL 时钟频率稳定后，SELPLL 位才能够被设置为 1。如果不是这样 (未锁定)，向其写入数据都会将“0”写入 SELPLL 位。

(2) 时钟控制寄存器 (CKC)

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

CKC 寄存器控制 PLL 模式下的内部系统时钟。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 0AH。

复位后：0AH R/W 地址：FFFFFF822H

	7	6	5	4	3	2	1	0
CKC	0	0	0	0	1	0	1	CKDIV0

CKDIV0	PLL模式下的内部系统时钟(f_{xx})
0	禁止设置
1	$f_{xx} = 8 \times f_x$ ($f_x = 3.0$ 至 6.0 MHz)

- 注意事项**
1. 请确保设置 CKC 寄存器为 0BH。如果设置的值不是 0BH，或者保留初始值而没有将其设置为 0BH，则禁止使能 PLL 操作（PLLCTL.SELPLL = 1）。
 2. 请确保将第 3 位和第 1 位设置为“1”，并清除第 7 位至第 4 位和第 2 位为“0”。

备注 CPU 时钟和外设时钟都由 CKC 寄存器分频，但只有 CPU 时钟由 PCC 寄存器分频。

(3) 锁定寄存器 (LOCKR)

上电后或 STOP 模式刚被释放后，在给定频率上发生锁相，需要等待的稳定时间就是入锁时间（频率稳定时间）。直到稳定之前的状态称为入锁状态，稳定后的状态称为锁定状态。

LOCKR 寄存器包括 LOCK 位，该位用于反映 PLL 频率稳定状态。

本寄存器为只读寄存器，按字节读取或按位读取。

系统复位后，该寄存器被设为 00H。

复位后: 00H R 地址: FFFFF824H

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

LOCK	PLL 锁存状态检查
0	锁定状态
1	未锁定状态

注意事项 实时情况下 LOCK 寄存器不能反映 PLL 的锁存状态。设置/清除条件如下：

[设置条件]

- 在系统复位时[#]
- IDLE2 或 STOP 模式下
- 在将 PLL 设置为停止（将 PLLCTL.PLLON 位清除为 0）时
- 在停止主时钟并使用副时钟提供给 CPU 工作（将 PCC.CK3 位设置为 1 且将 PCC.MCK 为设置为 1）时

注 复位将该寄存器设置为 01H，复位解除且振荡稳定时间之后清为 00H。

[清除条件]

- 在复位解除后振荡稳定时间（OSTS 寄存器缺省时间（参见 25.2 (3) 振荡稳定时间选择寄存器（OSTS）））溢出时
- 在 STOP 模式释放后振荡稳定定时器（时间由 OSTS 寄存器设置）溢出时，该 STOP 模式是在 PLL 操作状态下设置的
- 在 PLL 锁存时间定时器（时间由 PLLS 寄存器设置）溢出时，当 PLLCTL.PLLON 位从 0 变到 1 后
- 在 IDLE2 模式释放时插入的建立时间（时间由 OSTS 寄存器设置）完成后，该 IDLE2 模式是在 PLL 操作状态下设置的

(4) PLL 锁定时间规范寄存器 (PLLS)

PLLS 寄存器是 8 位寄存器，用于选择 PLLCTL.PLLON 位由 0 变到 1 时的 PLL 入锁时间。

本寄存器可以按字节读取或写入。

系统复位后，该寄存器被置为 03H。

复位后: 03H		R/W	地址: FFFFF6C1H					
	7	6	5	4	3	2	1	0
PLLS	0	0	0	0	0	0	PLLS1	PLLS0

PLLS1	PLLS0	PLL 锁存时间的选择
0	0	$2^{10}/f_x$
0	1	$2^{11}f_x$
1	0	$2^{12}/f_x$
1	1	$2^{13}/f_x$ (缺省值)

注意事项

1. 如此设置，入锁时间为 $800\mu s$ 或更久。
2. 入锁期间不要改变 PLLS 寄存器的设置。

6.5.3 用法**(1) 当使用锁相环时**

- 复位信号释放后，锁相环开始工作（PLLCTL.PLLON 位 = 1），但是由于缺省模式为时钟直通模式（PLLCTL.SELPLL 位 = 0），请选择锁相环模式（SELPLL 位 = 1）。
- 若要使能锁相环工作，首先将 PLLON 位设为 1，然后在 LOCKR.LOCK 位 = 0 之后，设置 SELPLL 位为 1。若要停止锁相环工作，首先设置为时钟直通模式（SELPLL 位 = 0），等待 8 个时钟周期或更长时间后，再停止锁相环工作（PLLON 位 = 0）。
- 无论如何设置，在转变为 IDLE2 或 STOP 模式期间，以及从 IDLE2 或 STOP 模式恢复至转变前的状态期间，PLL 都会停止工作。恢复操作需要的时间如下：

(a) 从时钟直通模式转变至 IDLE2 或 STOP 模式

- STOP 模式：设置 OSTS 寄存器，使得振荡稳定时间为 1 ms（最小值）或更长。
- IDLE2 模式：设置 OSTS 寄存器，使得建立时间为 $350\mu s$ （最小值）或更长。

(b) 在 PLL 模式中转变至 IDLE 2 或 STOP 模式

- STOP 模式：设置 OSTS 寄存器，使得振荡稳定时间为 1 ms（最小值）或更长。
- IDLE2 模式：设置 OSTS 寄存器，使得建立时间为 $800\mu s$ （最小值）或更长。

转变至 IDLE1 模式时，PLL 不会自动停止。如果需要，可以停止 PLL 工作。

(2) 当未使用锁相环时

- 复位信号释放后，选择时钟直通模式（SELPLL 位 = 0），但是锁相环处于工作状态（PLLON 位 = 1），因此必须将其停止（PLLON 位 = 0）。

第七章 16 位定时器/事件计数器AA (TAA)

定时器 AA (TAA) 是 16 位定时器/事件计数器。

V850ES/JG3-U 和 V850ES/JH3-U 单片机有六个定时器/事件计数器通道: TAA0 至 TAA5。

7.1 概述

TAA_n 概况如下:

- 时钟选择: 8 路
- 捕获/触发输入引脚: 2
- 外部事件计数输入引脚^注: 1
- 外部触发输入引脚^注: 1
- 定时器/计数器: 1
- 捕获/比较寄存器: 2
(通过 TAA0 和 TAA1, TAA2 和 TAA3 的级联, 可以使用 32 位捕获定时器功能)
- 捕获/比较匹配中断请求信号: 2
- 定时器输出引脚: 2

注 外部事件计数输入引脚和外部触发输入引脚都可以被复用作为捕获/触发输入引脚(TIAAm0)

备注 n = 0 至 5, m = 0 至 3, 5

7.2 功能

TAA_n 具有如下功能:

- 间隔定时器
- 外部事件计数器
- 外部触发脉冲输出
- 单次触发脉冲输出
- PWM 输出
- 自由运行定时器
- 脉冲宽度测量
- 定时器调谐功能
- 同时启动功能

7.3 配置

TAA_n 包括下列硬件：

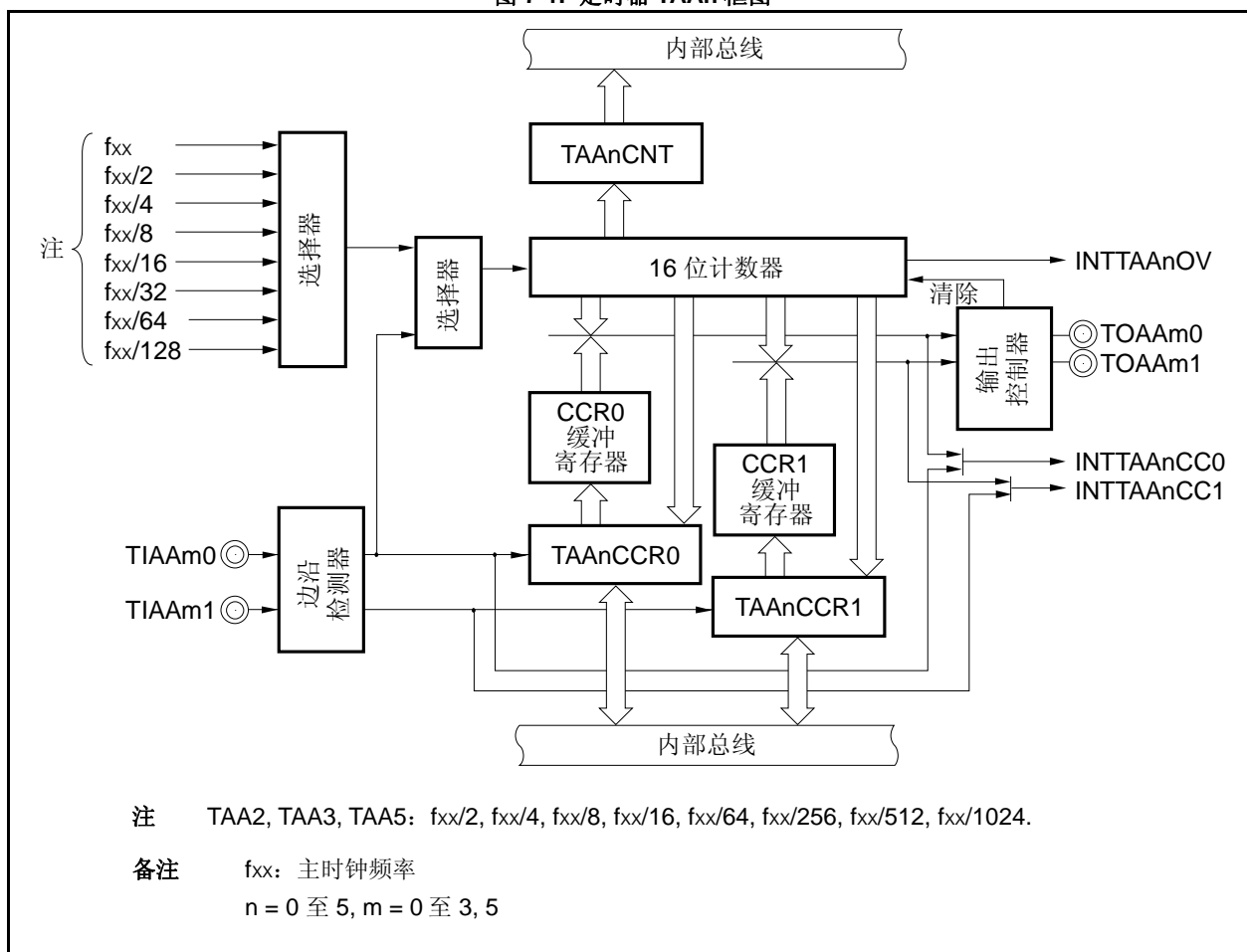
表 7-1. 定时器 TAA_n 的配置

项目	配置
寄存器	16 位计数器 TAA _n 捕获/比较寄存器 0, 1 (TAA _n CCR0, TAA _n CCR1) TAA _n 计数器读取缓冲寄存器 (TAA _n CNT) CCR0, CCR1 缓冲寄存器 TAA _n 控制寄存器 0, 1 (TAA _n CTL0, TAA _n CTL1) TAA _m 输入/输出控制寄存器 0 至 2, 4 (TAA _m IOC0 至 TAA _m IOC2, TAA _m IOC4) TAA _m 选项寄存器 0, 1 (TAA _m OPT0, TAA _m OPT1) TAA 噪声消除控制寄存器 (TANFC)
定时器输入 ^{注 1}	2 (TIAA _m 0 ^{注 2} , TIAA _m 1 引脚)
定时器输出 ^{注 1}	2 (TOAA _m 0, TOAA _m 1 引脚)

- 注 1. 当使用 TIAA_m0, TIAA_m1, TOAA_m0 和 TOAA_m1 引脚功能时, 参见表 4-20 端口引脚用作复用功能引脚。
2. TIAA_m0 引脚功能可以复用为捕获触发、外部事件计数以及外部触发的信号输入引脚。

备注 n = 0 至 5, m = 0 至 3, 5

图 7-1. 定时器 TAA_n 框图



(1) 16 位计数器

此 16 位计数器可以计数内部时钟或外部事件。

该计数器的计数值可以通过读取 TAAncNT 寄存器得到。

当 TAAncTL0.TAAncCE 位= 0 时，16 位计数器的值为 FFFFH。如果此时读取 TAAncNT 寄存器，则读回的值为 0000H。

系统复位将 TAAncCE 位清 0，因此，16 位计数器被设置为 FFFFH。

(2) CCR0 缓冲寄存器

CCR0 缓冲寄存器是一个 16 位比较寄存器，用于比较 16 位计数器的计数值。

当 TAAncCCR0 寄存器用作比较寄存器时，写入 TAAncCCR0 寄存器的值会传送到 CCR0 缓冲寄存器。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值相匹配时，就会产生一个比较匹配中断请求信号 (INTTAAncCC0)。

CCR0 缓冲寄存器不能直接读写。

TAAncCCR0 缓冲寄存器在复位后被清除为 0000H，因此，CCR0 寄存器被清除为 0000H。

(3) CCR1 缓冲寄存器

这是一个 16 位比较寄存器，用于比较 16 位计数器的计数值。

当 TAAncCCR1 寄存器用作比较寄存器时，写入 TAAncCCR1 寄存器的值会传送到 CCR1 缓冲寄存器。当 16 位计数器的计数值和 CCR1 缓冲寄存器的值匹配时，也会产生一个比较匹配中断请求信号 (INTTAAncCC1)。

CCR1 缓冲寄存器不能直接读写。

TAAncCCR1 缓冲寄存器在复位后被清除为 0000H，CCR1 寄存器被清除为 0000H。

(4) 边沿检测器

该电路检测输入到 TIAAm0 和 TIAAm1 引脚信号的有效边沿。通过设置 TAAmIOC1 和 TAAmIOC2 寄存器，可以选择无边沿、上升沿、下降沿、上升/下降双边沿作为有效沿。

(5) 输出控制器

该电路控制 TOAAm0 和 TOAAm1 引脚的输出。TOAAm0 和 TOAAm1 引脚的输出由 TAAmIOC0 寄存器控制。

(6) 选择器

此选择器选择 16 位计数器的计数时钟。可选择 8 种内部时钟或一个外部事件作为计数时钟。

7.3.1 引脚配置

TAA_n 配置的定时器输入和输出和下列端口共享。当使用各个引脚时必须设置端口功能 (参见表 4-20 端口引脚用作复用功能引脚)。

表 7-2 引脚配置

通道	端口	定时器 AA 输入	定时器 AA 输出	其它复用功能
TAA0	P32	TIAA00 ^{注1}	TOAA00	ASCK0/SCKF4
	P33	TIAA01	TOAA01	RTCDIV/RTCCL
TAA1	P34	TIAA10 ^{注1}	TOAA10	TOAA1OFF/INTP09/PPON
	P35	TIAA11	TOAA11	RTC1HZ/OCI
TAA2	P97	TIAA20 ^{注1}	TOAA20	SIF1/A7 ^{注2}
	P96	TIAA21	TOAA21	INTP11/A6 ^{注2}
TAA3	P95	TIAA30 ^{注1}	TOAA30	A5 ^{注2}
	P94	TIAA31	TOAA31	TENC00/EVTT0/A4 ^{注2}
TAA4	-	-	-	-
	-	-	-	-
TAA5	P915	TIAA50 ^{注1}	TOAA50	INTP18/A15 ^{注2}
	P914	TIAA51	TOAA51	INTP17/A14 ^{注2}

注 1. TIAAm0 引脚功能可以复用为捕获触发、外部事件计数以及外部触发的信号输入引脚。

2. 仅限 V850ES/JH3-U

备注 TAA4 没有定时器输入也没有定时器输出。于是 TAA4 自身只可以用作间隔定时器。但是，可以通过 TAA4 和 TAB1 的联合使用实现 6 相 PWM 输出功能。

7.4 寄存器

以下寄存器控制 TAA_n:

- TAA_n 控制寄存器 0 (TAA_nCTL0)
- TAA_n 控制寄存器 1 (TAA_nCTL1)
- TAA_n 输入/输出控制寄存器 0 (TAA_mIOC0)
- TAA_n 输入/输出控制寄存器 1 (TAA_mIOC1)
- TAA_n 输入/输出控制寄存器 2 (TAA_mIOC2)
- TAA_n 输入/输出控制寄存器 4 (TAA_mIOC4)
- TAA_n 选项寄存器 0 (TAA_mOPT0)
- TAA_n 选项寄存器 1 (TAA_mOPT1)
- TAA_n 捕获/比较寄存器 0 (TAA_nCCR0)
- TAA_n 捕获/比较寄存器 1 (TAA_nCCR1)
- TAA_n 计数器读取缓冲寄存器 (TAA_nCNT)
- TAA_n 噪声消除控制寄存器 (TANFC)

- 备注**
1. 当使用 TIAA_m0, TIAA_m1, TOAA_m0 和 TOAA_m1 引脚的功能时, 参见表 4-20 端口引脚用作复用功能。
 2. $n = 0$ 至 5, $m = 0$ 至 3, 5

(2) TAA_n 控制寄存器 1 (TAA_nCTL1)

TAA_nCTL1 寄存器是 8 位寄存器，用于控制定时器 TAA_n 的操作。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

(1/2)

复位后: 00H R/W 地址: TAA0CTL1 FFFFF631H, TAA1CTL1 FFFFF641H,
TAA2CTL1 FFFFF651H, TAA3CTL1 FFFFF661H,
TAA4CTL1 FFFFF671H, TAA5CTL1 FFFFF681H

	7	6	5	4	3	2	1	0
TAA0CTL1	TAA0SYE	TAA0EST	TAA0EEE	TAA0SYM	0	TAA0MD2	TAA0MD1	TAA0MD0
TAA1CTL1	0	TAA1EST	TAA1EEE	0	0	TAA1MD2	TAA1MD1	TAA1MD0
TAA2CTL1	TAA2SYE	TAA2EST	TAA2EEE	TAA2SYM	0	TAA2MD2	TAA2MD1	TAA2MD0
TAA3CTL1	0	TAA3EST	TAA3EEE	0	0	TAA3MD2	TAA3MD1	TAA3MD0
TAA4CTL1	TAA4SYE	0	0	TAA4SYM	0	TAA4MD2	TAA4MD1	TAA4MD0
TAA5CTL1	TAA5SYE	TAA5EST	TAA5EEE	TAA5SYM	0	TAA5MD2	TAA5MD1	TAA5MD0

TAA _m SYE	TAA _m SYM	调谐操作模式使能控制 (m = 0, 2, 4, 5)
0	0	独立操作模式 (异步操作模式)
0	1	禁止设置
1	0	调谐操作功能 (从操作的说明)
1	1	同时启动功能 (从定时器的说明)

这些位只能对从定时器设置 (禁止对主定时器设置这些位)。
主定时器和从定时器的关系如下。

主定时器	从定时器
TAA1	TAA0
TAA3	TAA2
TAB0	TAA5
TAB1	TAA4

关于调谐操作功能，参见 7.6 定时器调谐操作功能。
关于同时启动功能，参见 7.7 同时启动功能。

TAA _n EST	软件触发控制 (n = 0 至 3, 5)
0	-
1	为外部触发输入产生一个有效信号 • 在单次触发脉冲输出模式下： 向TAA _n EST位写入1作为触发信号，输出一个单次触发脉冲。 • 在外部触发脉冲输出模式下： 向TAA _n EST位写入1作为触发信号，输出一个PWM波形。

- 注意事项**
1. 只有在外触发脉冲输出模式或单次触发脉冲输出模式下，TAA_nEST 位有效。在其它模式下，向该位写入 1 的操作都被忽略。
 2. 请确保将 TAA_nCTL1 寄存器中各个指定为 0 的通道清为“0”。

TAAmEEE	计数时钟选择
0	禁止外部事件计数输入的操作 (执行对计数时钟的计数, 该时钟由 TAAmCTL0.TAAmCK0 至 TAAmCK2位选择)
1	使能外部事件计数输入的操作 (对外部事件计数输入信号的每个有效边沿执行计数)

通过TAAmEEE位选择使用内部计数时钟计数,
还是使用外部事件计数输入的有效边沿来计数。

TAAmMD2	TAAmMD1	TAAmMD0	定时器模式选择
0	0	0	间隔定时器模式
0	0	1	外部事件计数模式
0	1	0	外部触发脉冲输出模式
0	1	1	单次触发脉冲输出模式
1	0	0	PWM 输出模式
1	0	1	自由运行定时器模式
1	1	0	脉宽测量模式
1	1	1	禁止设置

- 注意事项**
1. 不论 **TAAmEEE** 位的值如何, 在外部事件计数模式下, 都会选择外部事件输入。
 2. 当 **TAAmCTL0.TAAmCE = 0** 时, 设置 **TAAmEEE** 和 **TAAmMD2** 至 **TAAmMD0** 位 (**TAAmCE** 位=1 时, 可以写入相同的值)。在 **TAAmCE** 位= 1 时, 重写操作是无法保证的, 若重写发生错误, 应把 **TAAmCE** 位清为 0, 再重新对这些位进行设定(m = 0 至 3, 5)。

(3) TAA_n 输入/输出控制寄存器 0 (TAA_nIOC0)

TAA_nIOC0 寄存器是 8 位寄存器，用于控制定时器的输出 (TOAA_n0, TOAA_n1 引脚)。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: TAA0IOC0 FFFFF632H, TAA1IOC0 FFFFF642H,
TAA2IOC0 FFFFF652H, TAA3IOC0 FFFFF662H,
TAA5IOC0 FFFFF682H

	7	6	5	4	3	2	1	0
TAA _n IOC0 (n = 0 至 3, 5)	0	0	0	0	TAA _n OL1	TAA _n OE1	TAA _n OL0	TAA _n OE0

TAA _n OL1	TOAA _n 1 引脚输出电平设置 ^注
0	TOAA _n 1 引脚输出从高电平开始
1	TOAA _n 1 引脚输出从低电平开始

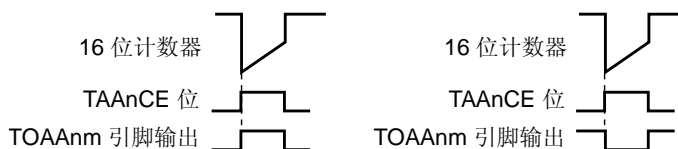
TAA _n OE1	TOAA _n 1 引脚输出设置
0	定时器输出禁止 • 当 TAA _n OL1 位 = 0: 从 TOAA _n 1 引脚输出低电平 • 当 TAA _n OL1 位 = 1: 从 TOAA _n 1 引脚输出高电平
1	定时器输出使能 (从 TOAA _n 1 引脚输出一个方波)

TAA _n OL0	TOAA _n 0 引脚输出电平设置 ^注
0	TOAA _n 0 引脚输出从高电平开始
1	TOAA _n 0 引脚输出从低电平开始

TAA _n OE0	TOAA _n 0 引脚输出设置
0	定时器输出禁止 • 当 TAA _n OL0 位 = 0: 从 TOAA _n 0 引脚输出低电平 • 当 TAA _n OL0 位 = 1: 从 TOAA _n 0 引脚输出高电平
1	定时器输出使能 (从 TOAA _n 0 引脚输出一个方波)

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

- 当 TAA_nOL_m 位 = 0
- 当 TAA_nOL_m 位 = 1



- 注意事项**
1. 当 TAA_nCTL0.TAA_nCE 位 = 0 时，重新写入 TAA_nOL1, TAA_nOE1, TAA_nOL0 和 TAA_nOE0 位 (当 TAA_nCE 位 = 1 时，可以写入相同的值)。如果重新写入不能正确执行，将 TAA_nCE 位清为 0，然后重新设定该位。
 2. 即使在 TAA_nCE 和 TAA_nOEm 位均为 0 时对 TAA_nOL_m 位进行操作，TOAA_nm 引脚的输出电平也会改变。

备注 m = 0, 1

(4) TAA_n 输入/输出控制寄存器 1 (TAA_nIOC1)

TAA_nIOC1 寄存器是 8 位寄存器，用于控制捕获触发输入信号 (TIAAn0, TIAAn1 引脚) 的有效边沿。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: TAA0IOC1 FFFFF633H, TAA1IOC1 FFFFF643H,
TAA2IOC1 FFFFF653H, TAA3IOC1 FFFFF663H,
TAA5IOC1 FFFFF683H

	7	6	5	4	3	2	1	0
TAA _n IOC1	0	0	0	0	TAA _n IS3	TAA _n IS2	TAA _n IS1	TAA _n IS0

(n = 0 至 3, 5)

TAA _n IS3	TAA _n IS2	捕获触发输入信号 (TIAAn1 引脚) 有效边沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双沿

TAA _n IS1	TAA _n IS0	捕获触发输入信号 (TIAAn0 引脚) 有效边沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双沿

- 注意事项**
1. 当 TAA_nCTL0.TAA_nCE 位=0 时，重新写入 TAA_nIS3 至 TAA_nIS0 位 (当 TAA_nCE 位=1 时，可写入相同值)。如果重写发生错误，将 TAA_nCE 位清为 0，然后重新进行位的设定。
 2. 只有在自由运行定时器模式和脉冲宽度测量模式下，TAA_nIS3 至 TAA_nIS0 位才有效。在所有其它模式下，无法进行捕获操作。

(5) TAA_n 输入/输出控制寄存器 2 (TAA_nIOC2)

TAA_nIOC2 寄存器是 8 位寄存器，用于控制外部事件计数器输入信号 (TIAAn0 引脚) 和外部触发输入信号 (TIAAn0 引脚) 的有效边沿。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: TAA0IOC2 FFFFF634H, TAA1IOC2 FFFFF644H,
 TAA2IOC2 FFFFF654H, TAA3IOC2 FFFFF664H.
 TAA5IOC2 FFFFF684H

	7	6	5	4	3	2	1	0
TAA _n IOC2 (n = 0 至 3, 5)	0	0	0	0	TAA _n EES1	TAA _n EES0	TAA _n ETS1	TAA _n ETS0

TAA _n EES1	TAA _n EES0	外部事件计数输入信号 (TIAAn0 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	检测双沿

TAA _n ETS1	TAA _n ETS0	外部事件计数输入信号 (TIAAn0 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	检测双沿

- 注意事项**
1. 当 TAA_nCTL0.TAA_nCE 位=0 时，重新写入 TAA_nEES1，TAA_nEES0，TAA_nETS1 和 TAA_nETS0 位 (当 TAA_nCE 位=1 时，可以写入相同的值)。如果执行重写发生错误，将 TAA_nCE 位清为 0，然后重新进行位设定。
 2. 只有当 TAA_nCTL1.TAA_nEEE 位= 1 或设置为外部事件计数模式 (TAA_nCTL1.TAA_nMD2 至 TAA_nCTL1.TAA_nMD0 位= 001) 时，TAA_nEES1 和 TAA_nEES0 位有效。
 3. 只有在外围触发脉冲输出模式 (TAA_nCTL1.TAA_nMD2 至 TAA_nCTL1.TAA_nMD0 位=010) 或在单次触发脉冲输出模式 (TAA_nCTL1.TAA_nMD2 至 TAA_nCTL1.TAA_nMD0 位=011) 时，TAA_nETS1 和 TAA_nETS0 位有效。

(6) TAA_n 输入/输出控制寄存器 4 (TAA_nIOC4)

TAA_nIOC4 寄存器是 8 位寄存器，用于定时器的输出。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。停止定时器操作(TAA_nCTL0.TAA_nCE = 0)，该寄存器不会被复位。

注意事项 1. 在下列状态时，禁止访问 TAA_nIOC4 寄存器。详情参见 3.4.8(2) 访问专用片上外设 I/O 寄存器。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
- 当 CPU 运行于内部振荡时钟时。

2. 只有在间隔定时器模式和自由运行定时器模式下，可以设置 TAA_nIOC4 寄存器。在其它模式下，请确保设置 TAA_nIOC4 寄存器为 00H (模式设置的详情参见 7.4 (2) TAA_n 控制寄存器 1 (TAA_nCTL1))。如果 TAA_nCCR0 和 TAA_nCCR1 寄存器被设置为捕获功能，即使在自由运行模式下，TAA_nIOC4 寄存器的设置无效。

复位后: 00H	R/W	地址:	TAA0IOC4 FFFFFFF63CH, TAA1IOC4 FFFFFFF64CH, TAA2IOC4 FFFFFFF65CH, TAA3IOC4 FFFFFFF66CH, TAA5IOC4 FFFFFFF68CH					
TAA _n IOC4	7	6	5	4	3	2	1	0
(n = 0 至 3, 5)	0	0	0	0	TAA _n OS1	TAA _n OR1	TAA _n OS0	TAA _n OR0
		TAA _n OS1	TAA _n OR1	TIAAn1 引脚的切换控制				
		0	0	无请求。正常切换操作。				
		0	1	重置请求 在16 位计数器的值和TAA _n CCR1寄存器的值再次匹配之前，固定为无效电平。				
		1	0	设置请求 在16 位计数器的值和TAA _n CCR1寄存器的值再次匹配之前，固定为有效电平。				
		1	1	保持请求 保持当前输出电平				
		TAA _n OS0	TAA _n OR0	TIAAn0 引脚的切换控制				
		0	0	无请求。正常切换操作。				
		0	1	重置请求 在16 位计数器的值和TAA _n CCR0寄存器的值再次匹配之前，固定为无效电平。				
		1	0	设置请求 在16 位计数器的值和TAA _n CCR0寄存器的值再次匹配之前，固定为有效电平。				
		1	1	保持请求 保持当前输出电平				

(8) TAA_n 选项寄存器 1 (TAA_nOPT1)

TAA_nOPT1 寄存器是 8 位寄存器，用于控制级联实现的 32 位捕获功能。

在定时器操作期间(TAA_nCTL0.TAA_nCE = 1)，禁止重写该寄存器。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H	R/W	地址:	TAA0OPT1 FFFFF63DH, TAA2OPT1 FFFFF65DH
----------	-----	-----	--

	7	6	5	4	3	2	1	0
TAA _n OPT1	TAA _n CSE	0	0	0	0	0	0	0

(n = 0, 2)

TAA _n CSE	级联控制
0	单独操作或作为级联功能的低位操作
1	作为级联功能的高位操作

注意事项

1. 级联和定时器调谐操作不能一起使用。级联操作时，请确保 TAA_nCTL1.TAA_nSYE 被清除为 0。
2. 级联操作时，设置自由运行定时器模式，且使用 TAA_nCCR0 和 TAA_nCCR1 寄存器作为捕获寄存器。
级联的详情参见 7.8 级联

(9) TAA_n 捕获/比较寄存器 0 (TAA_nCCR0)

根据不同工作模式，TAA_nCCR0 寄存器可以用作捕获寄存器或比较寄存器。

在自由计时器模式下，根据 TAA_nOPT0.TAA_nCCS0 位的设定，TAA_nCCR0 寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TAA_nCCR0 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以被用作比较寄存器。

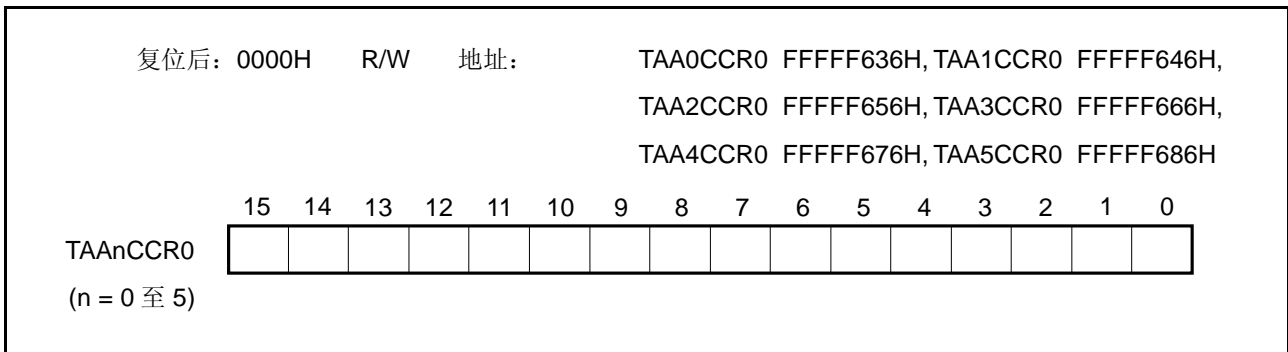
TAA_nCCR0 寄存器可以在操作中进行读取和写入。

此寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

注意事项 在以下状态禁止访问 TAA_nCCR0 寄存器。详情参见 3.4.8 (2) 访问片上外设输入/输出专用寄存器。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
- 当 CPU 运行于内部振荡时钟时。



(a) 作为比较寄存器时的功能

即便在 TAA_nCTL0.TAA_nCE 位=1 时, TAA_nCCR0 寄存器仍可以重新写入。

TAA_nCCR0 寄存器的设定值传送给 CCR0 缓冲寄存器。当 16 位计数器的值与 CCR0 缓冲寄存器的值相匹配时, 就会生成一个比较匹配中断请求信号 (INTTAA_nCC0)。如果此时 TOAA_n0 引脚输出允许, 则 TOAA_n0 引脚的输出会反相。

在间隔定时器模式、外部事件计数器模式、外部触发脉冲输出模式、单次触发脉冲输出模式或 PWM 输出模式时, 当 TAA_nCCR0 寄存器作为周期寄存器时, 如果 16 位计数器的计数值与 CCR0 缓冲寄存器数值匹配, 则计数器清除为 (0000H)。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下, 当 TAA_nCCR0 寄存器用作捕获寄存器时, 若检测到捕获触发输入引脚 (TIAA_n0 引脚) 的有效边沿, 则 16 位计数器的当前计数值存储于 TAA_nCCR0 寄存器中。在脉宽测量模式下, 若检测到捕获触发输入引脚 (TIAA_n0 引脚) 的有效边沿, 16 位计数器的当前计数值就储存在 TAA_nCCR0 寄存器中, 且 16 位计数器清零 (0000H)。

即使捕获操作和读取 TAA_nCCR0 寄存器的操作发生冲突, 还是可以正确读取 TAA_nCCR0 寄存器的值。

下表显示了捕获/比较寄存器在各工作模式下的功能, 以及如何向比较寄存器写入数据。

表 7-3. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

操作模式	捕获/比较寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	任何时候均可写入
外部事件计数器	比较寄存器	任何时候均可写入
外部触发脉冲输出	比较寄存器	批量写入
单次触发脉冲输出	比较寄存器	任何时候均可写入
PWM (脉宽调制信号) 输出	比较寄存器	批量写入
自由运行定时器	捕获/比较寄存器	任何时候均可写入
脉冲宽度测量	捕获寄存器	-

(10) TAA_n 捕获/比较寄存器 1 (TAA_nCCR1)

根据不同工作模式，TAA_nCCR1 寄存器可以用作捕获寄存器或比较寄存器。

在自由计时器模式下，根据 TAA_nOPT0.TAA_nCCS1 位的设定，TAA_nCCR1 寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TAA_nCCR1 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以被用作比较寄存器。

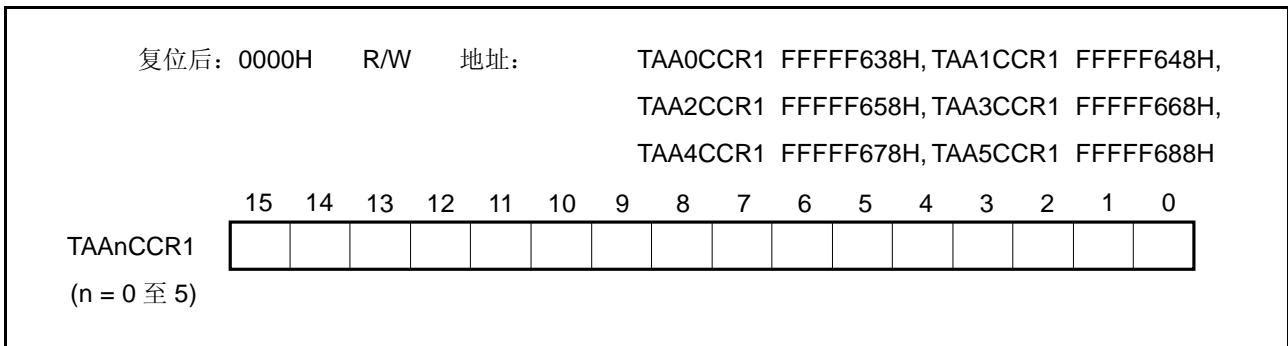
TAA_nCCR1 寄存器可以在操作中进行读取和写入。

该寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

注意事项 在以下状态时禁止访问 TAA_nCCR1 寄存器。详情参见 3.4.8 (2) 访问片上外设输入/输出专用寄存器。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
- 当 CPU 运行于内部振荡时钟时。



(a) 用作比较寄存器时的功能

即使 TAA nCTL0.TAA nCE 位=1 时, TAA nCCR1 寄存器仍可以重新写入。

TAA nCCR1 寄存器的设定值传送给 CCR1 缓冲寄存器。当 16 位计数器的值与 CCR1 缓冲寄存器的值相匹配时, 就会生成一个比较匹配中断请求信号 (INTTAA nCC1)。如果此时 TOAA n1 引脚输出允许, 则 TOAA n1 引脚的输出会反相。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下, 当 TAA nCCR1 寄存器用作捕获寄存器时, 若检测到捕获触发输入引脚 (TIAA n1 引脚) 的有效边沿, 则 16 位计数器的当前计数值存储于 TAA nCCR1 寄存器中。在脉宽测量模式下, 若检测到捕获触发输入引脚 (TIAA n1 引脚) 的有效边沿, 16 位计数器的当前计数值就储存在 TAA nCCR1 寄存器中, 且 16 位计数器清零 (0000H)。

即使捕获操作和读取 TAA nCCR1 寄存器的操作发生冲突, 还是可以正确读取 TAA nCCR1 寄存器的值。

下表显示了捕获/比较寄存器在各工作模式下的功能, 以及如何向比较寄存器写入数据。

表 7-4. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

操作模式	捕获/比较寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	任何时候均可写入
外部事件计数器	比较寄存器	任何时候均可写入
外部触发脉冲输出	比较寄存器	批量写入
单次触发脉冲输出	比较寄存器	任何时候均可写入
PWM (脉宽调制信号) 输出	比较寄存器	批量写入
自由运行定时器	捕获/比较寄存器	任何时候均可写入
脉冲宽度测量	捕获寄存器	-

(11) TAA_n 计数器读取缓冲寄存器 (TAA_nCNT)

TAA_nCNT 寄存器是读取缓冲寄存器，可以读取 16 位计数器的计数值。

如果在 TAA_nCTL0.TAA_nCE 位=1 时读取该寄存器，可以读出 16 位定时器的计数值。

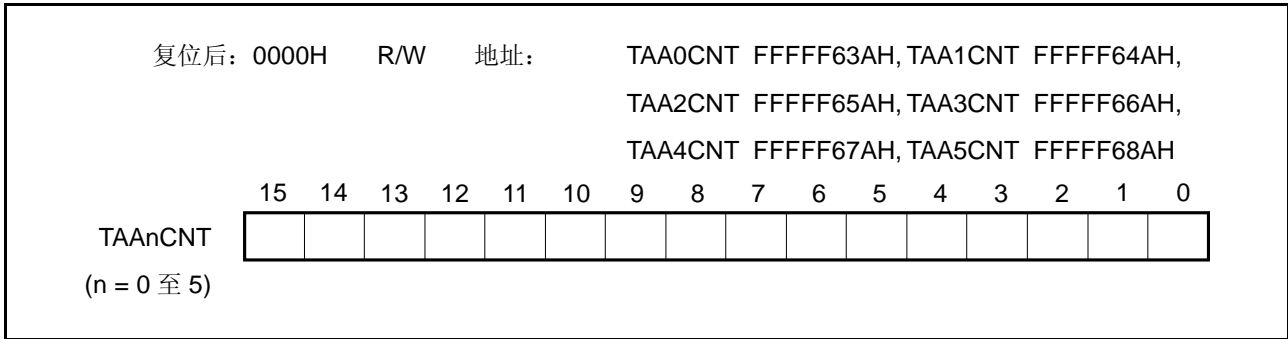
该寄存器为只读寄存器，按 16 位宽度读取。

当 TAA_nCE 位=0 时，TAA_nCNT 寄存器的值被清为 0000H。如果此时读取 TAA_nCNT 寄存器，则不会读到 16 位计数器的值 (FFFFH)，读到的值为 0000H。

在复位后 TAA_nCE 位的值被清为 0000H，因此 TAA_nCNT 寄存器被清为 0000H。

注意事项 在以下状态时，禁止访问 TAA_nCNT 寄存器。详情参见 3.4.8 (2) 访问片上外设输入/输出专用寄存器。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
- 当 CPU 运行于内部振荡时钟时。



(12) 噪声消除控制寄存器 (TANFC)

可以为 TIAAn0 和 TIAAn1 引脚选择数字噪声消除。噪声消除的设置由 TANFC 寄存器选择。

当选择选择数字噪声消除时，数字采样的采样时钟可以选 fxx 和 fxx/4。采样执行 3 次。

该寄存器可以按字节进行读取或写入。

复位输入将该寄存器清为 00H。

注意事项 在采样时钟被更改后，需要 3 个采样时钟的时间来初始化数字噪声消除器。如果在采样时钟被更改后，在 3 个采样时钟的时间完成前，有 TIAAn0 和 TIAAn1 的有效边沿输入，于是，可能会产生一个中断请求信号。因此，当使用 TAA 的外部触发功能、外部事件功能和捕获触发功能时，在经过 3 个采样时钟的时间后，使能 TAA 的操作。

备注 n = 0 至 3, 5

复位后: 00H R/W 地址: FFFFF724H

	7	6	5	4	3	2	1	0
TANFC	TANFEN	0	0	0	0	0	0	TANFC0

TANFEN	数字噪声消除的设置
0	不执行数字噪声消除
1	执行数字噪声消除

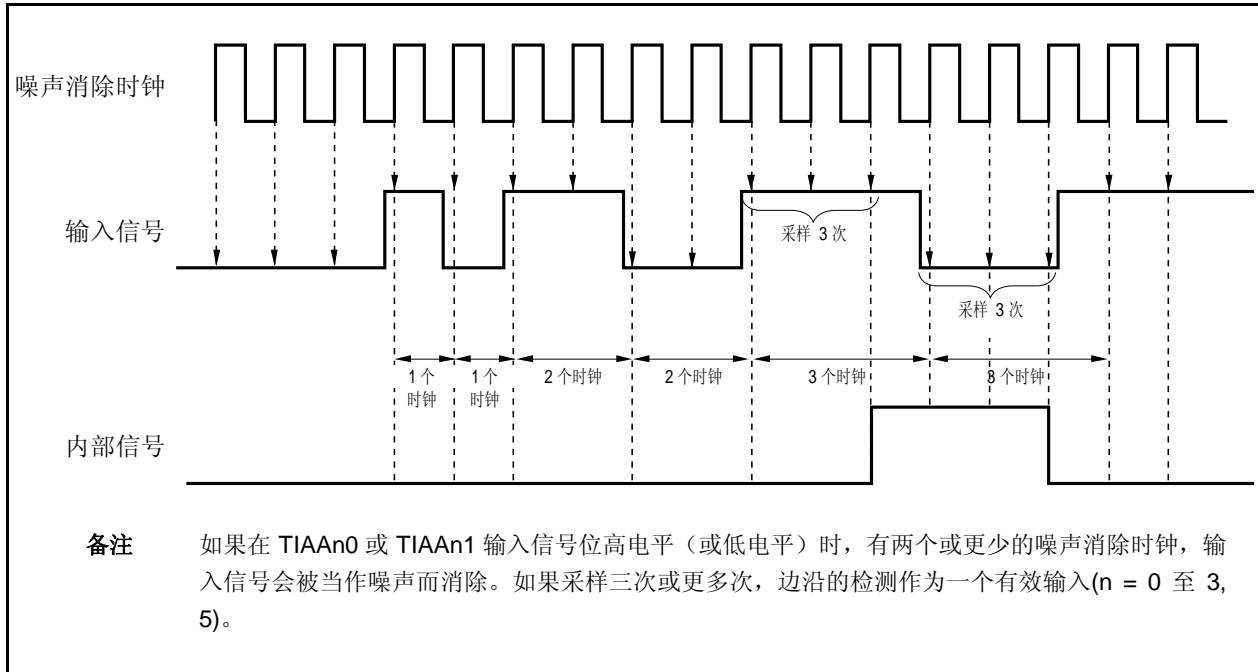
TANFC0	数字采样时钟
0	fxx
1	fxx/4

备注

1. 由于采样执行 3 次，可以可靠消除的噪声宽度是 2 个采样时钟周期。
2. 当噪声宽度小于 2 个采样时钟周期的情况时，如果噪声和采样时钟同步输入，则产生一个中断请求。

图 7-2 所示的是一个通过定时器 AA 输入引脚数字滤波器执行噪声消除的时序示例。

图 7-2. 数字噪声消除时序的示例



7.5 操作

定时器 TAA_n 可以执行如下操作。

操作	TAA _n CTL1.TAA _n EST 位 (软件触发位)	TIAAn0 引脚 (外部触发输入)	捕获/比较寄存器 设置	比较寄存器写入
间隔计时器模式	无效	无效	仅作比较寄存器	任何时候均可写入
外部事件计数器模式 ^{注 1}	无效	无效	仅作比较寄存器	任何时候均可写入
外部触发脉冲输出模式 ^{注 2}	有效	有效	仅作比较寄存器	批量写入
单次触发脉冲输出模式 ^{注 2}	有效	有效	仅作比较寄存器	任何时候均可写入
PWM 输出模式	无效	无效	仅作比较寄存器	批量写入
自由定时器模式	无效	无效	可以切换	任何时候均可写入
脉冲宽度测量模式 ^{注 2}	无效	无效	仅作捕获寄存器	无效

- 注**
1. 在外部事件计数器模式下，应该指定对 TIAAn0 引脚捕获触发输入的有效边沿不作检测（将 TAA_nIOC1.TAA_nIS1 和 TAA_nIOC1.TAA_nIS0 位清为“00”）。
 2. 在外部触发脉冲输出模式、单次触发脉冲输出模式和脉冲宽度测量模式下，选择内部时钟作为计数时钟。（将 TAA_nCTL1.TAA_nEEE 位清为 0）。

备注 n = 0 至 3, 5

(1) 随时写入和批量写入

在 TAA_n 定时器操作期间(TAA_nCTL0.TAA_nCE 位 = 1)，可以写入 TAA_nCCR0 和 TAA_nCCR1 寄存器，但是 CCR0 和 CCR1 缓冲寄存器的写入方法（随时写入，批量写入）取决于不同的模式。

(a) 随时写入

该模式下，在定时器操作期间的随时，数据可以从 TAA_nCCR0 和 TAA_nCCR1 寄存器传送到 CCR0 和 CCR1 缓冲寄存器。

图 7-3. 随时写入的基本操作流程图示例（TAA0 的间隔定时器模式）

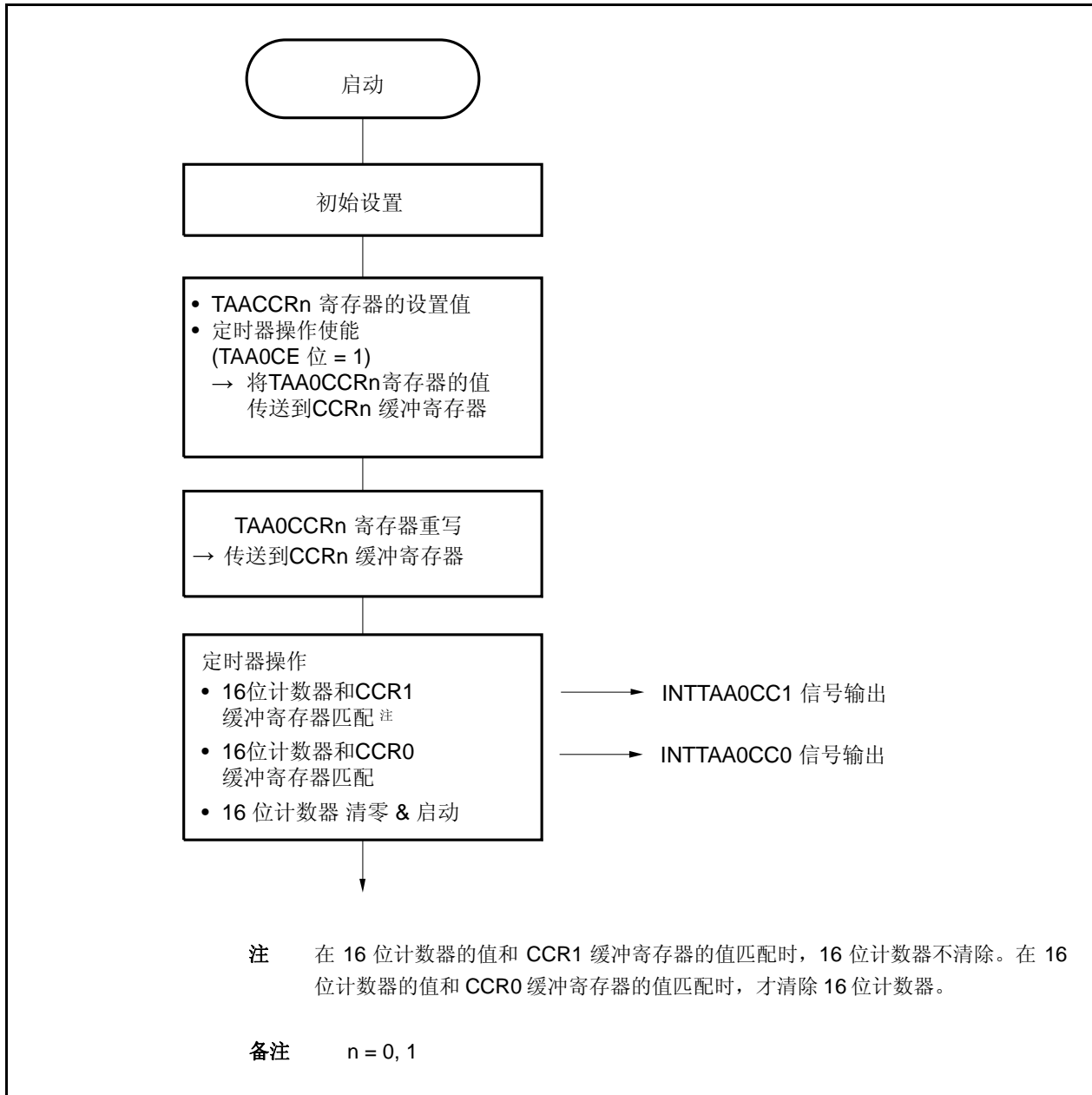
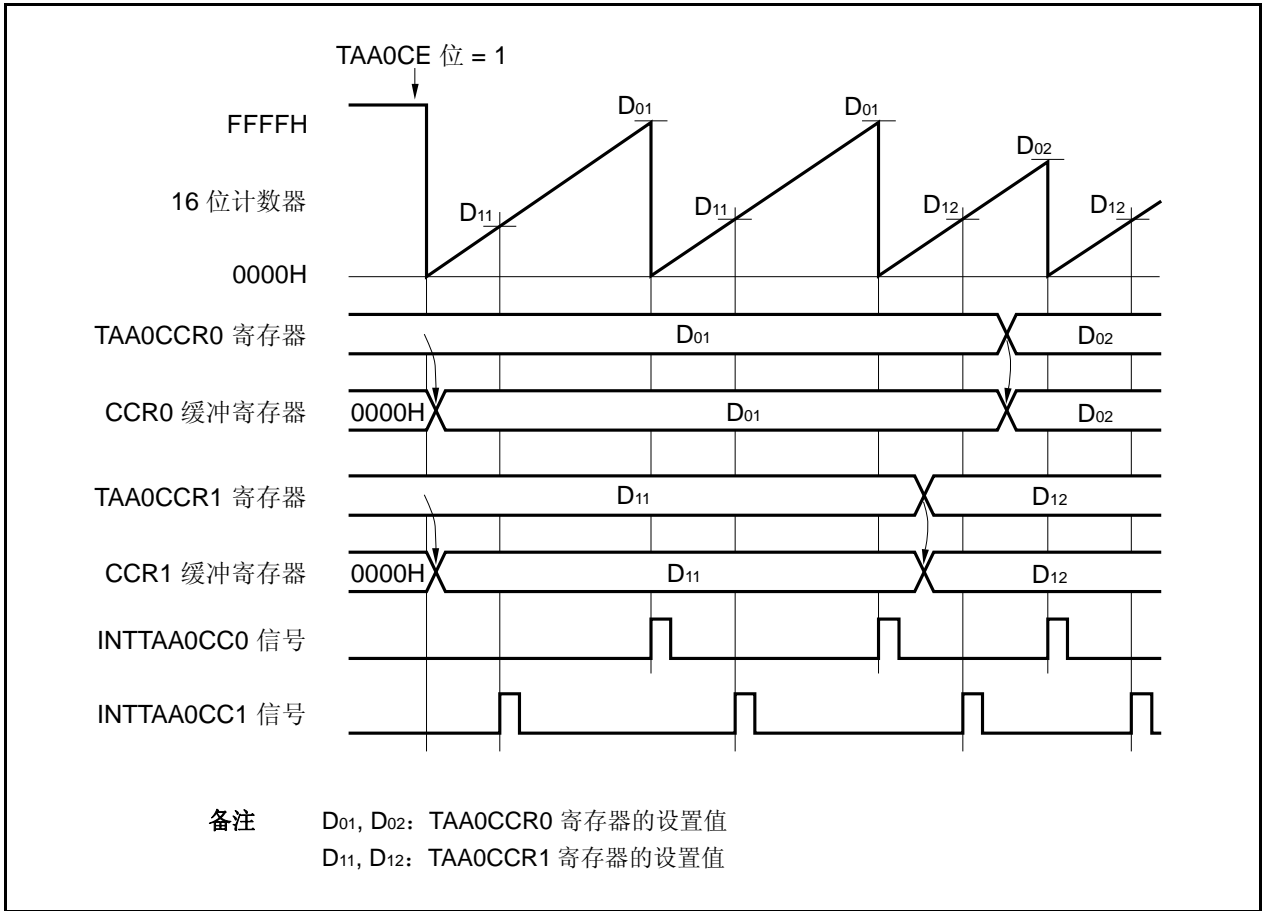


图 7-4. 随时写入的时序示例 (TAA0 的间隔定时器模式)



(b) 批量写入

该模式下，在定时器操作期间，数据一次性从 TAA_nCCR0 和 TAA_nCCR1 寄存器传送到 CCR0 和 CCR1 缓冲寄存器。在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配时，传送该数据。写入 TAA_nCCR1 寄存器就可以使能传送。是否写入 TAA_nCCR1 寄存器可以控制使能或禁止下一次传送时序。

为了让重写 TAA_nCCR0 和 TAA_nCCR1 寄存器的设置值成为 16 位计数器比较值（换句话说，为了该值被传送到 CCR0 和 CCR1 缓冲寄存器），必需重写 TAA_nCCR0 寄存器，然后在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配之前写入 TAA_nCCR1 寄存器。因此，在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配时，TAA_nCCR0 和 TAA_nCCR1 寄存器的值被传送到 CCR0 和 CCR1 缓冲寄存器。所以，即使只想重写 TAA_nCCR0 寄存器的值，也需要将相同的值（和 TAA_nCCR1 寄存器的当前值相同）写入 TAA_nCCR1 寄存器。

图 7-5. 批量写入的基本操作流程图示例 (TAA0 的 PWM 输出模式)

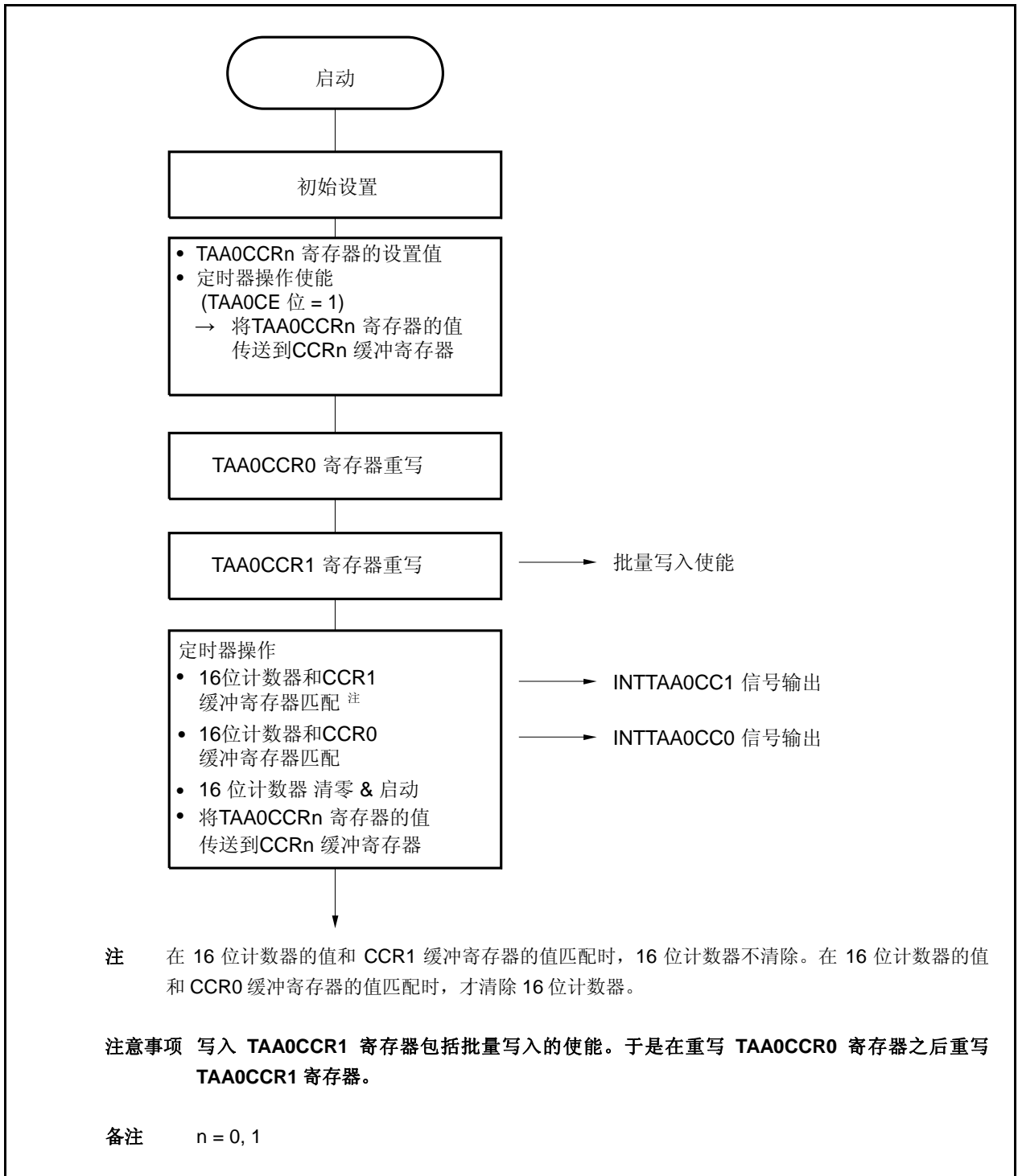
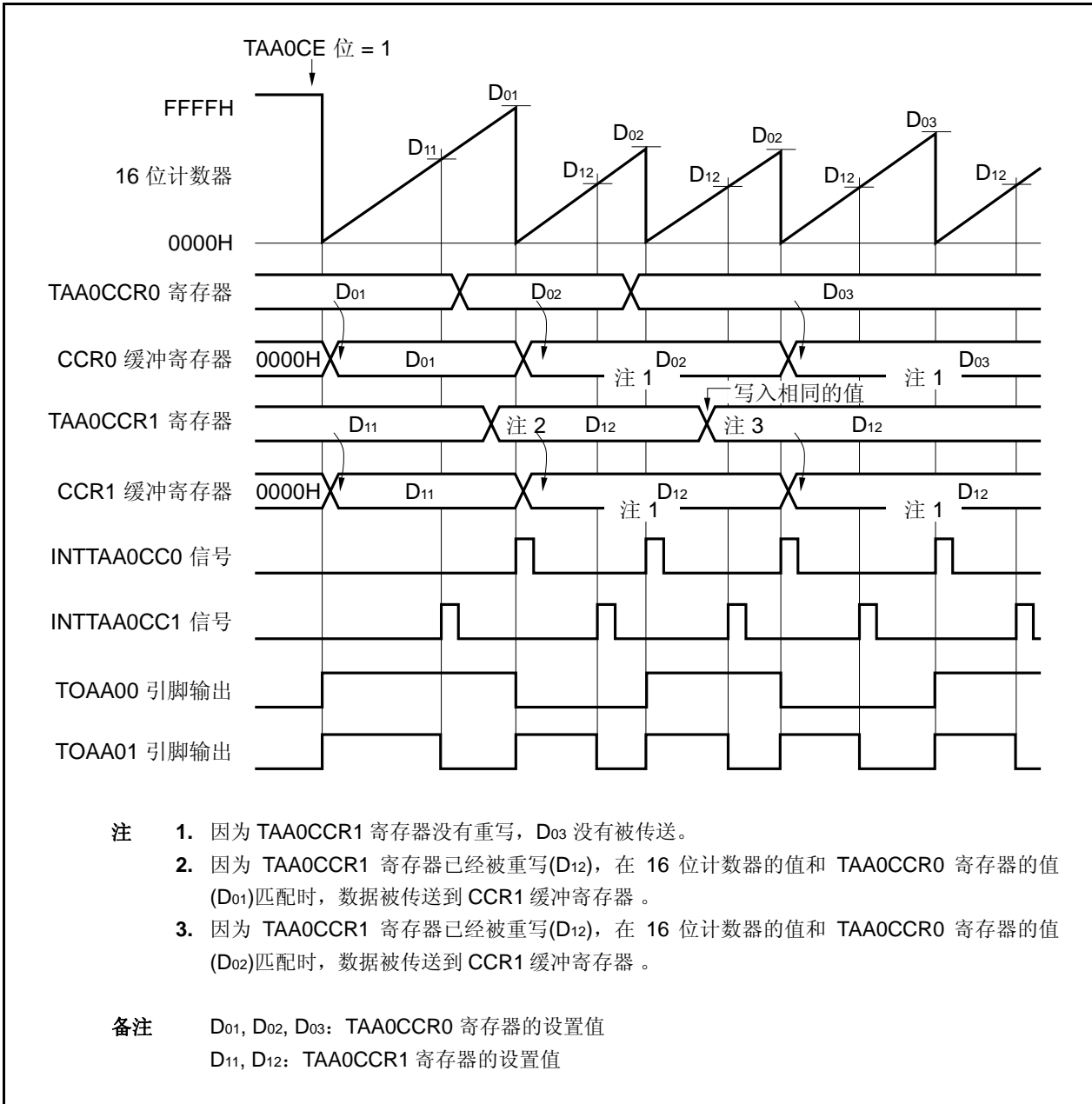


图 7-6. 批量写入的时序示例 (TAA0 的间隔定时器模式)



7.5.1 间隔定时器模式 (TAAmMD2 至 TAAmMD0 位=000)

在间隔定时器模式下, 如果 TAAAnCTL0.TAAAnCE 位被置为 1, 则会以指定时间间隔生成中断请求信号 (INTTAAAnCC0), 从 TOAAAn0 引脚输出方波, 指定的时间间隔等于方波的半周期。

通常在间隔定时器模式下不使用 TAAAnCCR1 寄存器。

图 7-7. 间隔定时器的配置

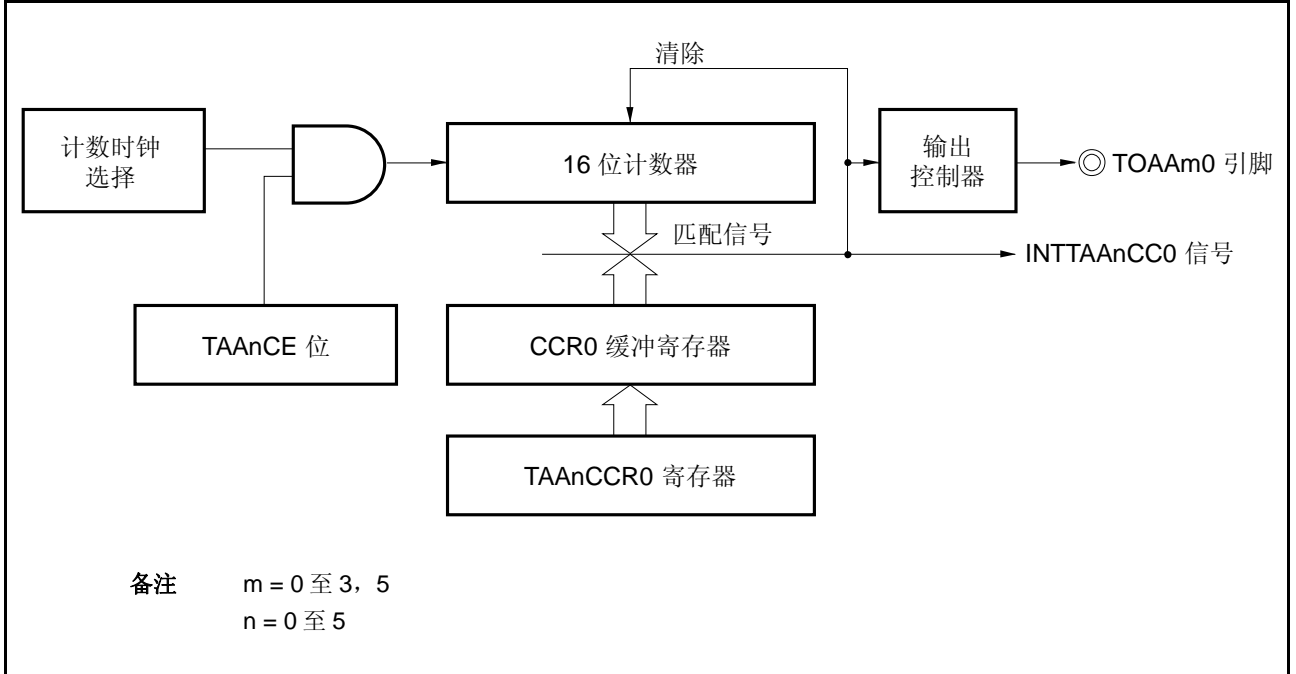
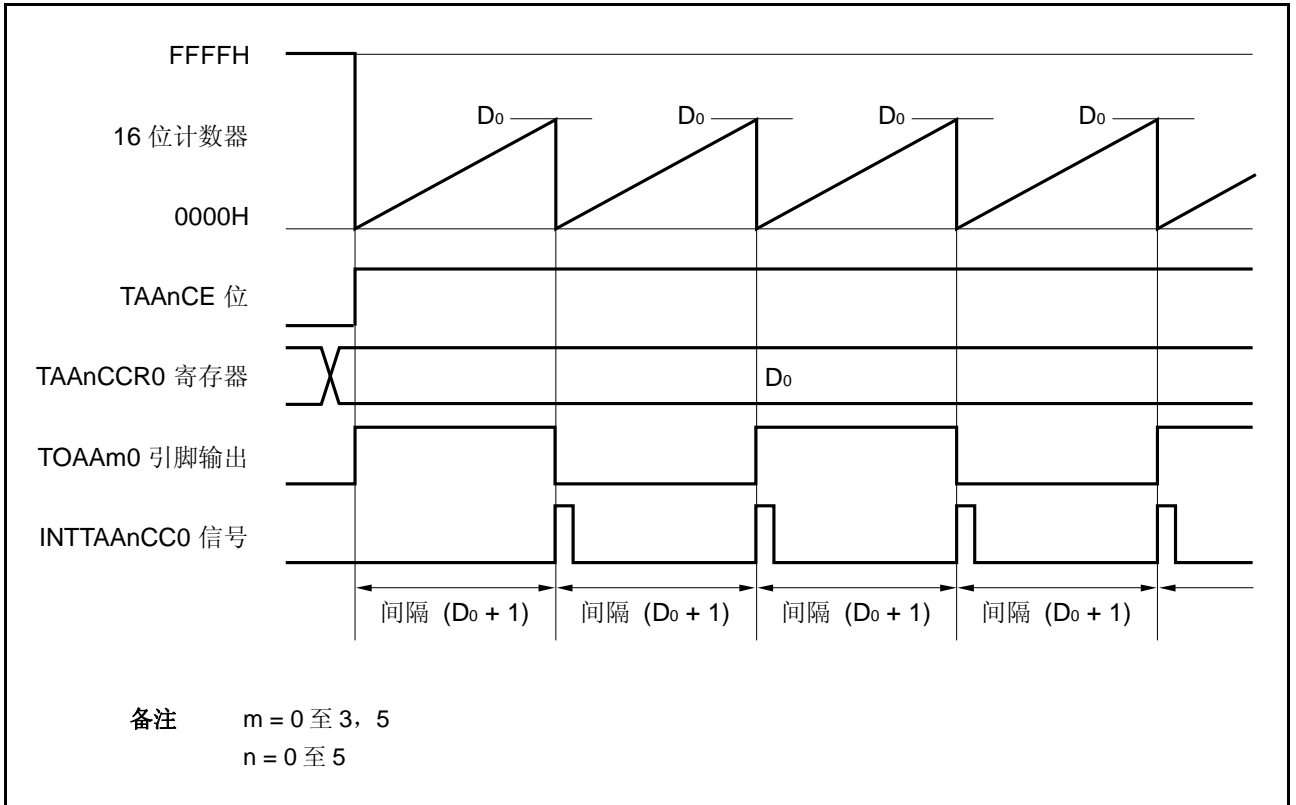


图 7-8. 间隔定时器模式操作的基本时序



当 TAA_nCE 位被置为 1 时，16 位计数器的值由计数时钟同步从 FFFFH 清除为 0000H，计数器开始计数。此时，TOAA_n0 引脚的输出反相。此外，TAA_nCCR0 寄存器的设定值传送给 CCR0 缓冲寄存器。

当 16 位计数器的计数值和 CCR0 缓冲寄存器的值相匹配时，则 16 位计数器清为 0000H，TOAA_n0 引脚输出反相，并生成一个比较匹配中断请求信号 (INTTAA_nCC0)。

时间间隔可用下述表达式计算。

$$\text{间隔时间} = (\text{TAA}_{n}\text{CCR0 寄存器的设定值} + 1) \times \text{计数时钟周期}$$

备注 m = 0 至 3, 5
n = 0 至 5

图 7-9. 间隔定时器模式操作时的寄存器设置 (1/2)

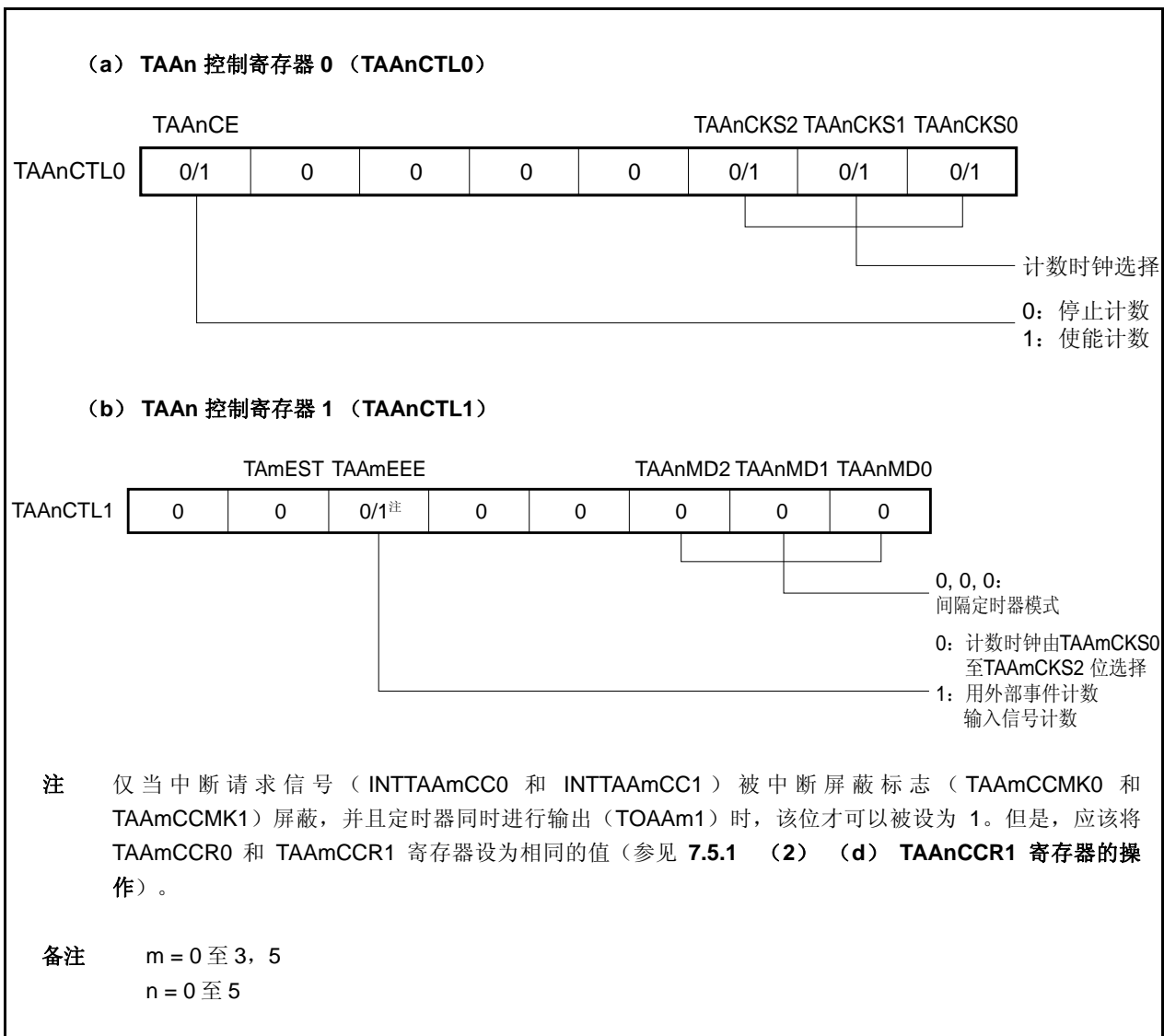
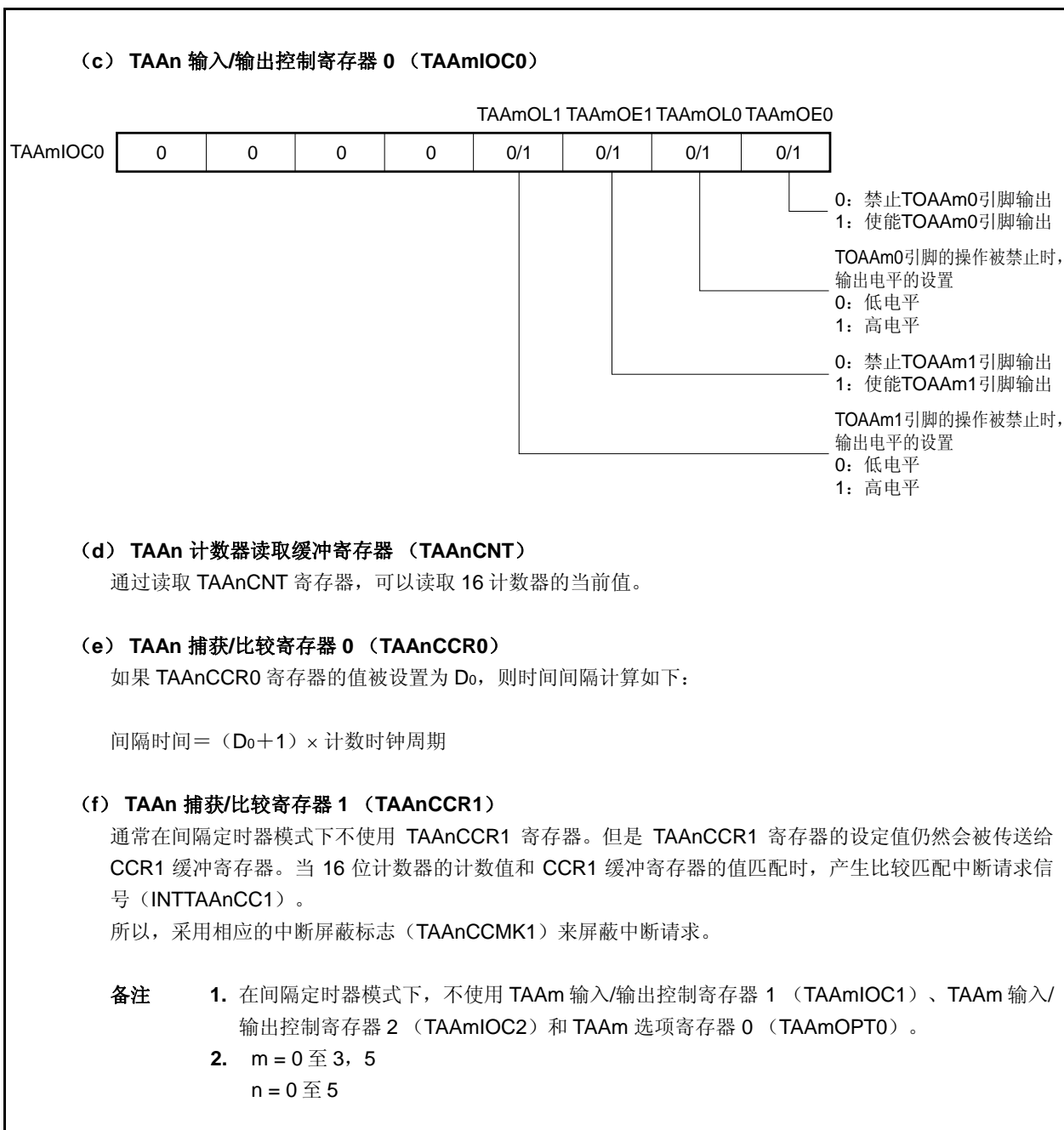
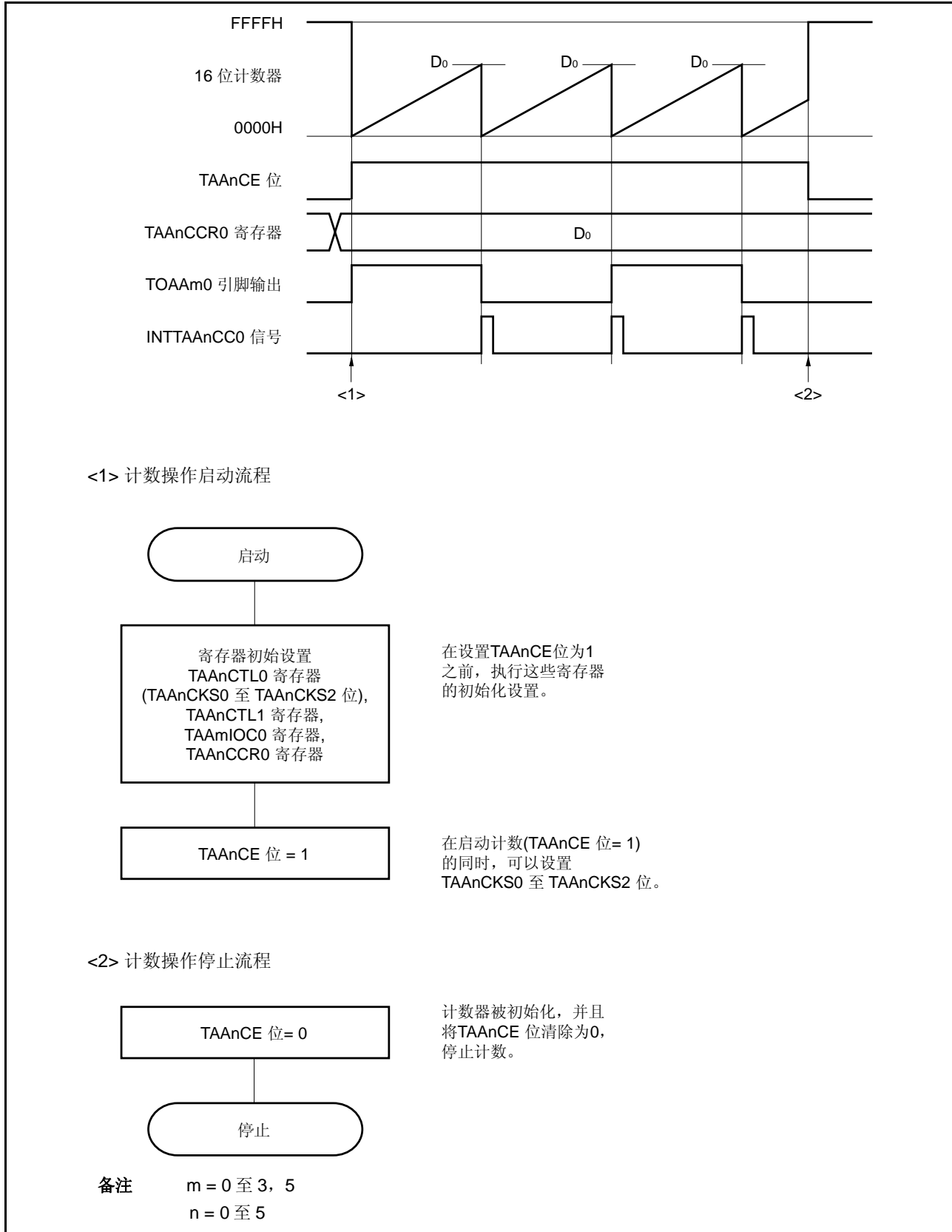


图 7-9. 间隔定时器模式操作时的寄存器设置 (2/2)



(1) 间隔定时器模式操作流程

图 7-10. 间隔定时器模式下的软件处理流程

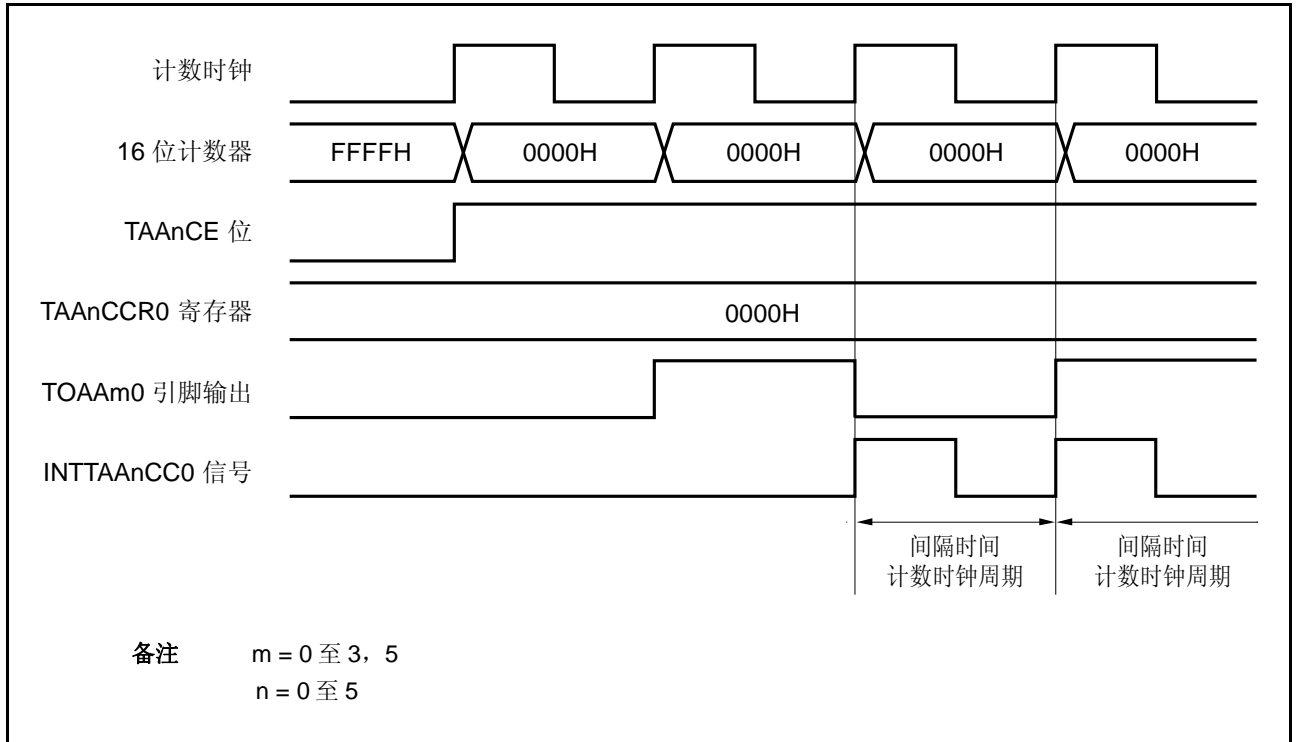


(2) 间隔定时器模式操作时序

(a) TAA_nCCR0 寄存器清为 0000H 时的操作

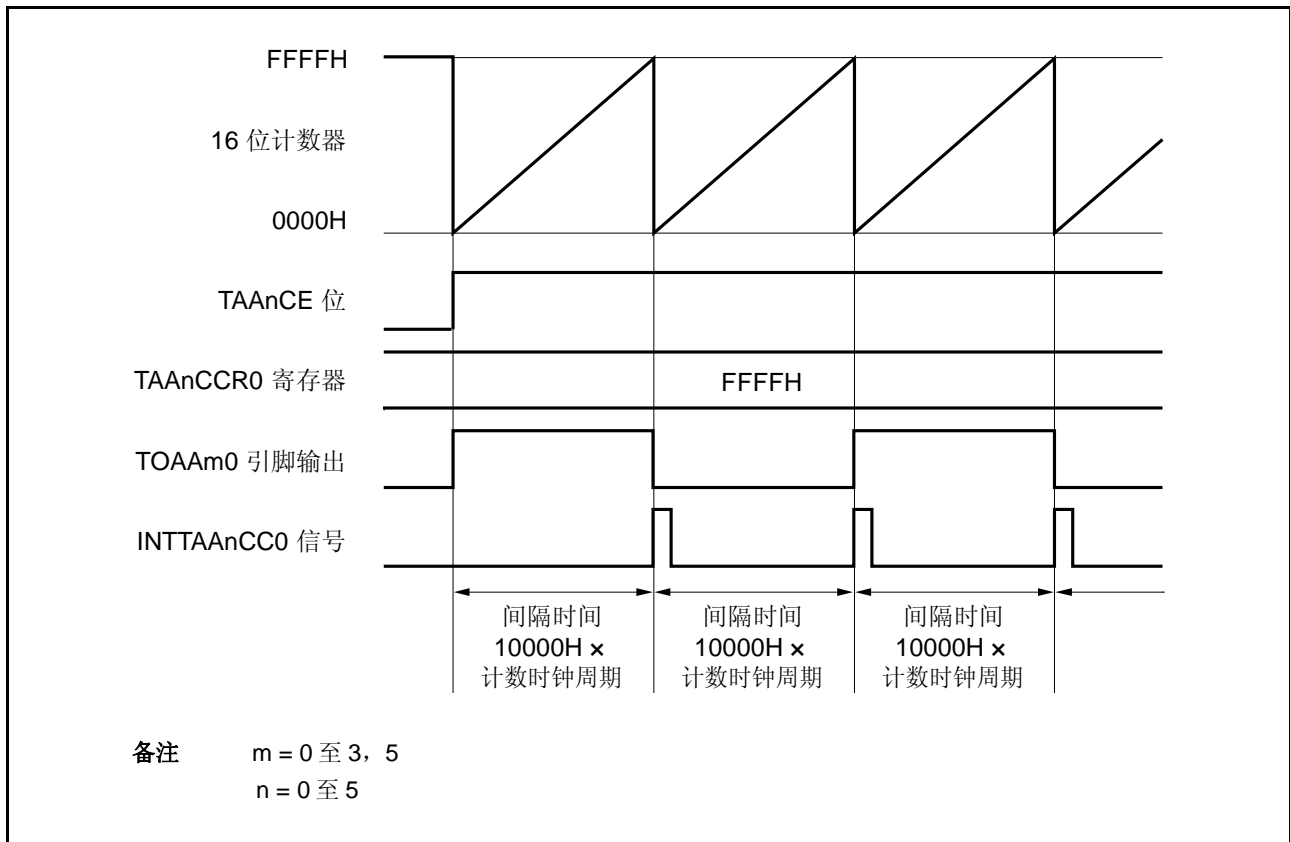
如果 TAA_nCCR0 寄存器被清为 0000H，第二个时钟及其后的每个时钟计数时都会产生 INTTAA_nCC0 信号，而且，TOAA_n0 引脚的输出反相。

16 位计数器的值总是 0000H。



(b) TAA_nCCR0 寄存器被置为 FFFFH 时的操作

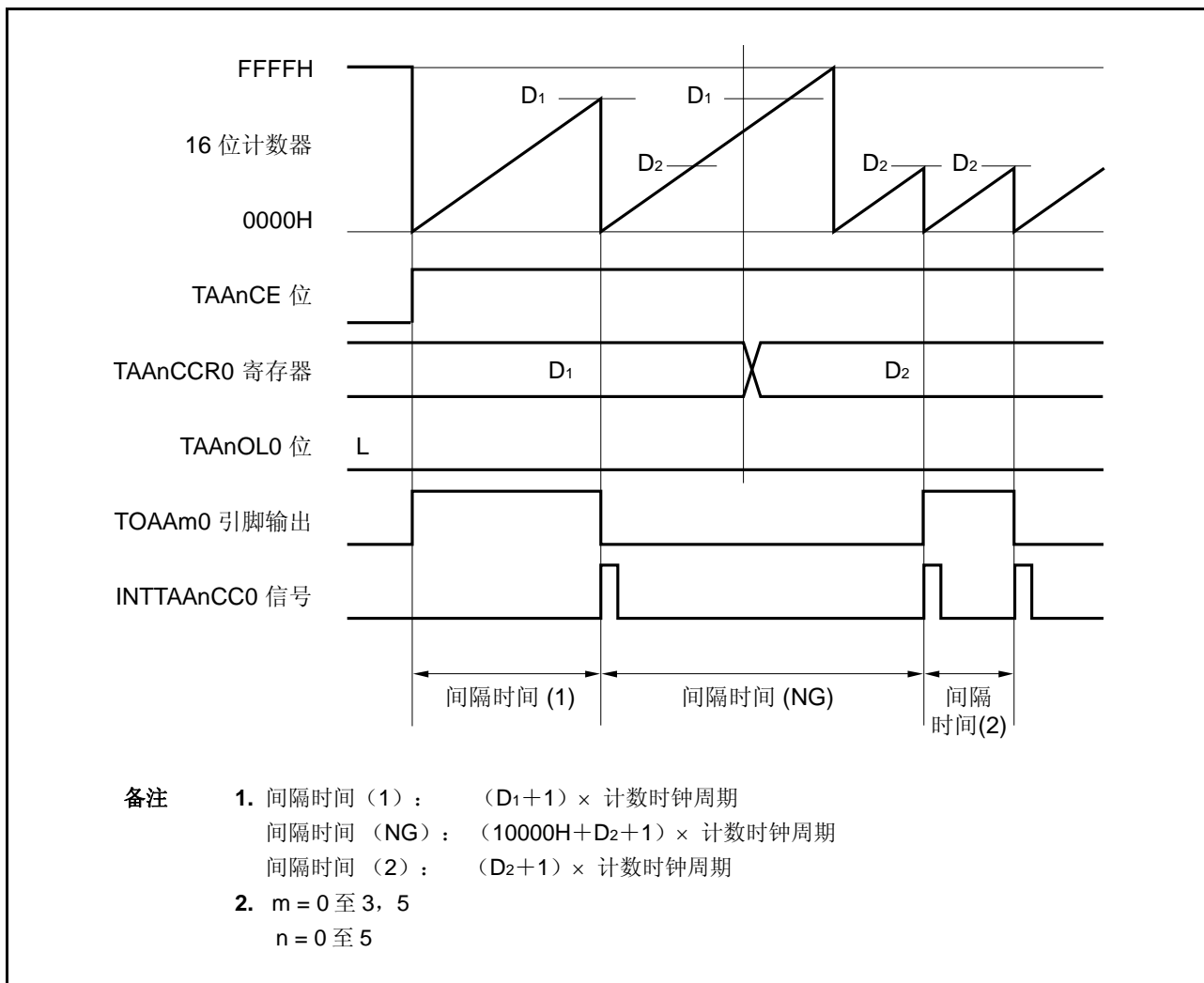
如果 TAA_nCCR0 寄存器被置为 FFFFH，那么 16 位计数器计数到 FFFFH。该计数器在下一个计数时刻被同步清零。产生 INTTAA_nCC0 信号，并且 TOAA_n0 引脚输出反相。此时，不产生溢出中断请求信号 (INTTAA_nOV)，而且，溢出标志 (TAA_mOPT0.TAA_mOVF 位) 也不会置 1。



(c) 重新写入 TAA_nCCR0 寄存器的注意事项

要将 TAA_nCCR0 寄存器的值改为比当前值较小的值，应停止计数，然后改变其设定值。

如果在计数器工作期间，将 TAA_nCCR0 寄存器的值重写为比当前值较小的值，16 位计数器可能会溢出。



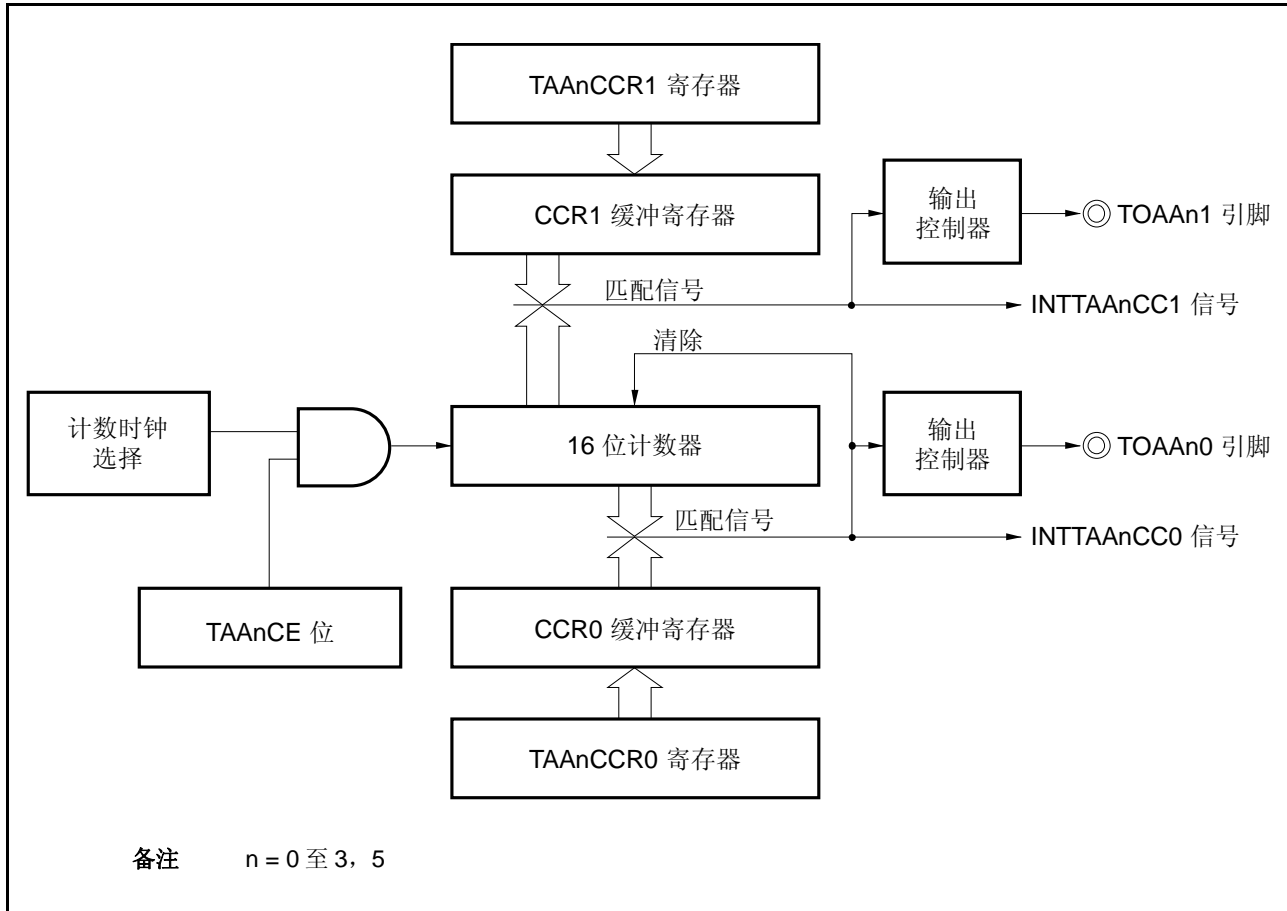
当 16 位计数器计数值大于 D_2 而小于 D_1 时，如果将 TAA_nCCR0 寄存器的值从 D_1 变为 D_2 ，在重写 TAA_nCCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。结果，用来比较的 16 位计数器的值是 D_2 。

但是，由于计数值已经超过了 D_2 ，所以 16 位计数器就会向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值和 D_2 相符时才产生 INTTAA_nCC0 信号，且 TOAA_n0 引脚的输出反相。

因此，可能不会在原先预计的间隔时间“ $(D_1 + 1) \times$ 计数时钟周期”或“ $(D_2 + 1) \times$ 计数时钟周期”时产生 INTTAA_nCC0 信号，而会以“ $(10000H + D_2 + 1) \times$ 计数时钟周期”的间隔产生中断请求信号。

(d) TAA_nCCR1 寄存器的操作

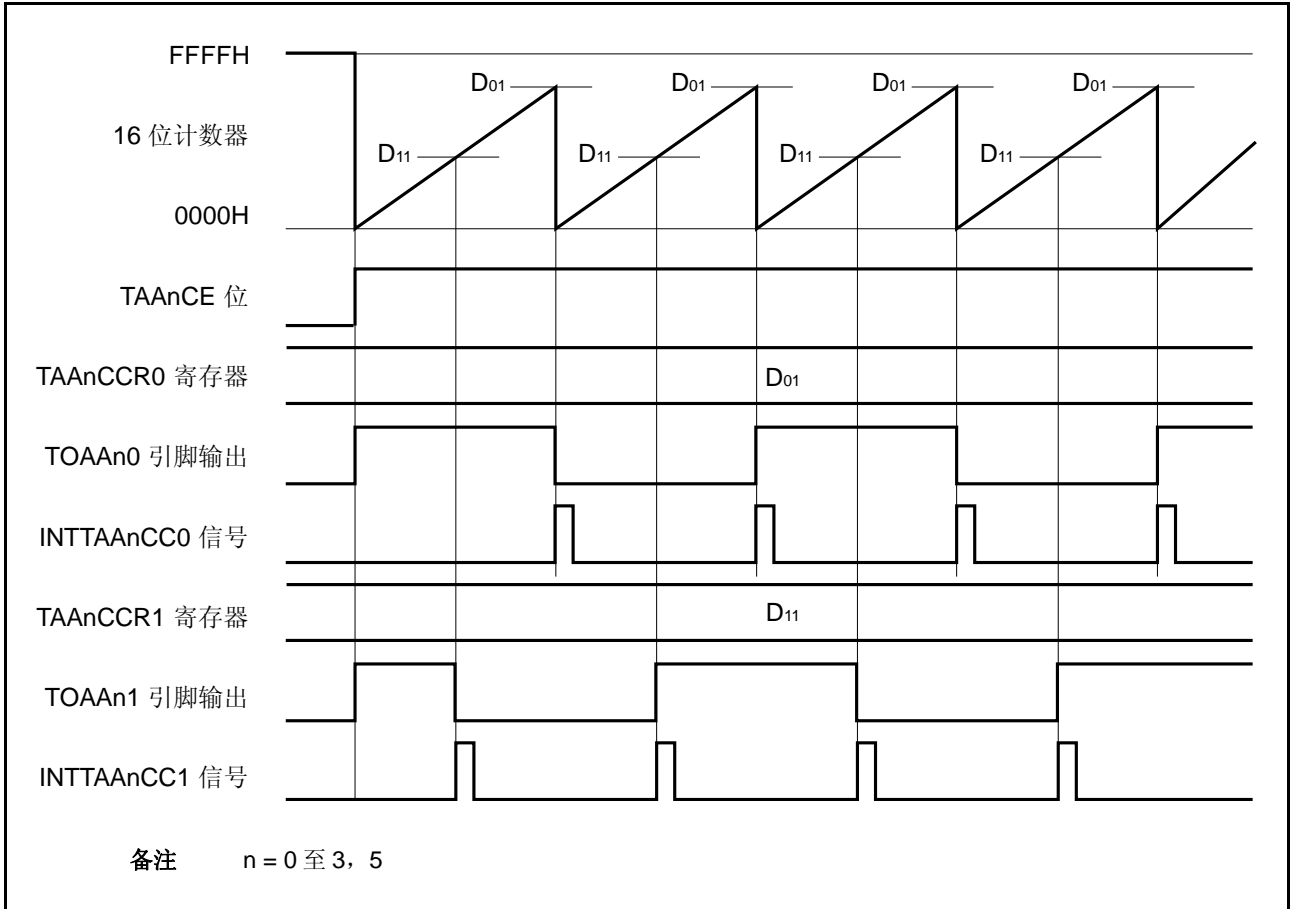
图 7-11. TAA_nCCR1 寄存器的配置



如果 TAA_nCCR1 寄存器的设定值小于 TAA_nCCR0 寄存器的设定值，则每个计数周期产生一次 INTTAA_nCC1 中断请求信号。同时，TOAA_n1 引脚的输出反相。

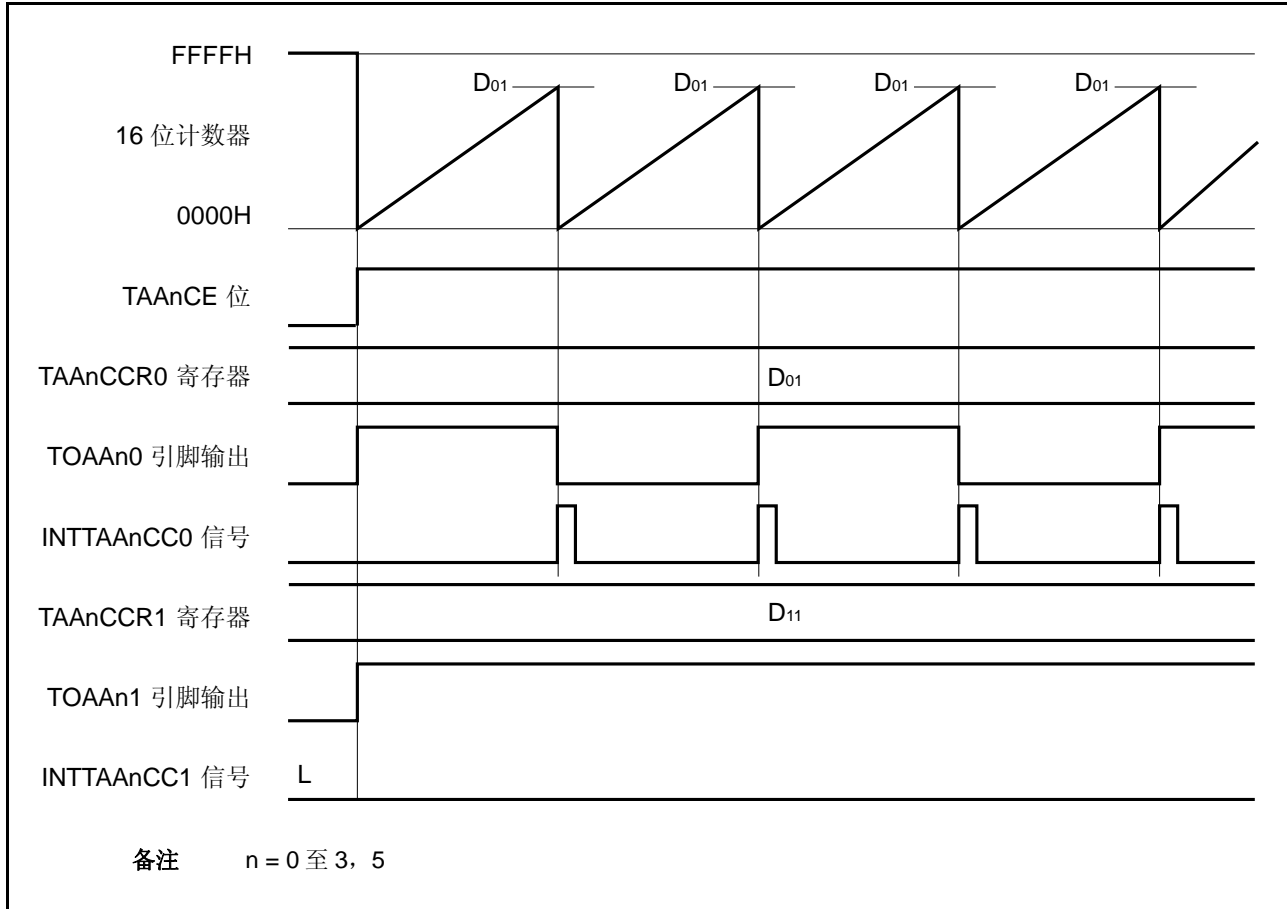
TOAA_n1 引脚输出一个方波，其周期与 TOAA_n0 引脚的输出周期相同。

图 7-12. 当 $D_{01} \geq D_{11}$ 时的时序图



如果 TAAAnCCR1 寄存器的设定值大于 TAAAnCCR0 寄存器的设定值，16 位计数器的计数值不会和 TAAAnCCR1 寄存器的值发生匹配。因此也就不会产生 INTTAAAnCC1 信号，TOAAAn1 引脚的输出也不会改变。

图 7-13. 当 $D_{01} < D_{11}$ 时的时序图



7.5.2 外部事件计数模式 (TAA_nMD2 至 TAA_nMD0 位=001)

在外部事件计数器模式下，当 TAA_nCTL0.TAA_nCE 位被置为 1 时，对外部事件计数输入信号的有效边沿进行计数，每次有效边沿的计数数量达到指定数目时，产生一个中断请求信号 (INTTAA_nCC0)。不能使用 TOAA_n0 引脚。

通常在外部事件计数器模式下，不使用 TAA_nCCR1 寄存器。

图 7-14. 外部事件计数模式的配置

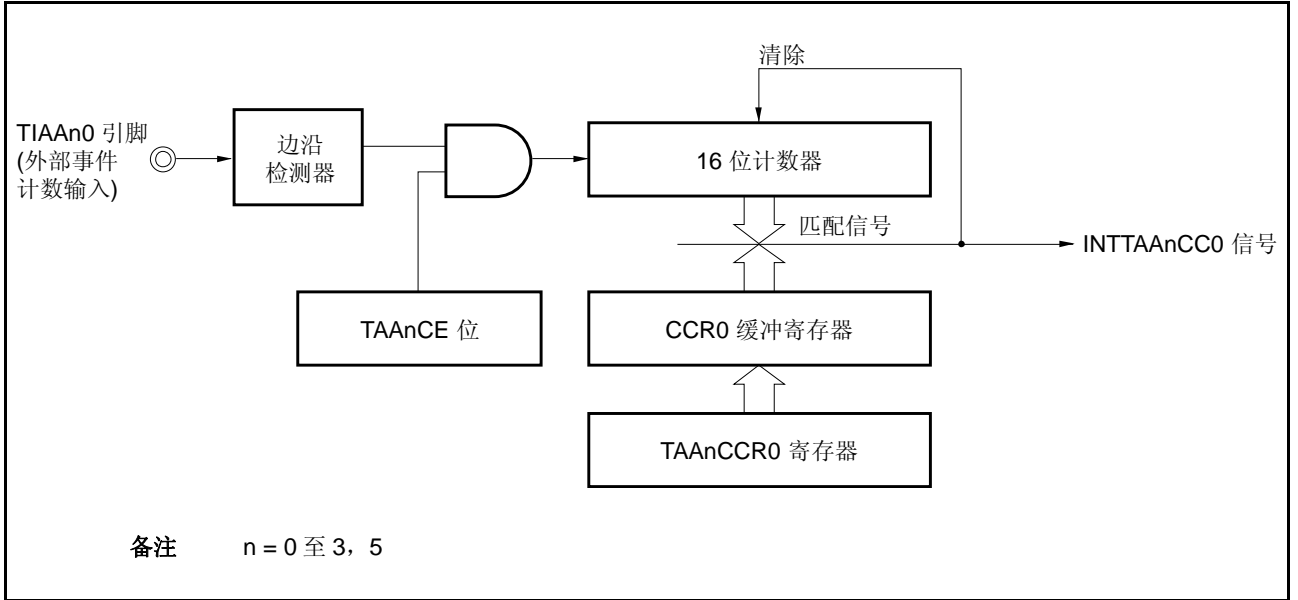
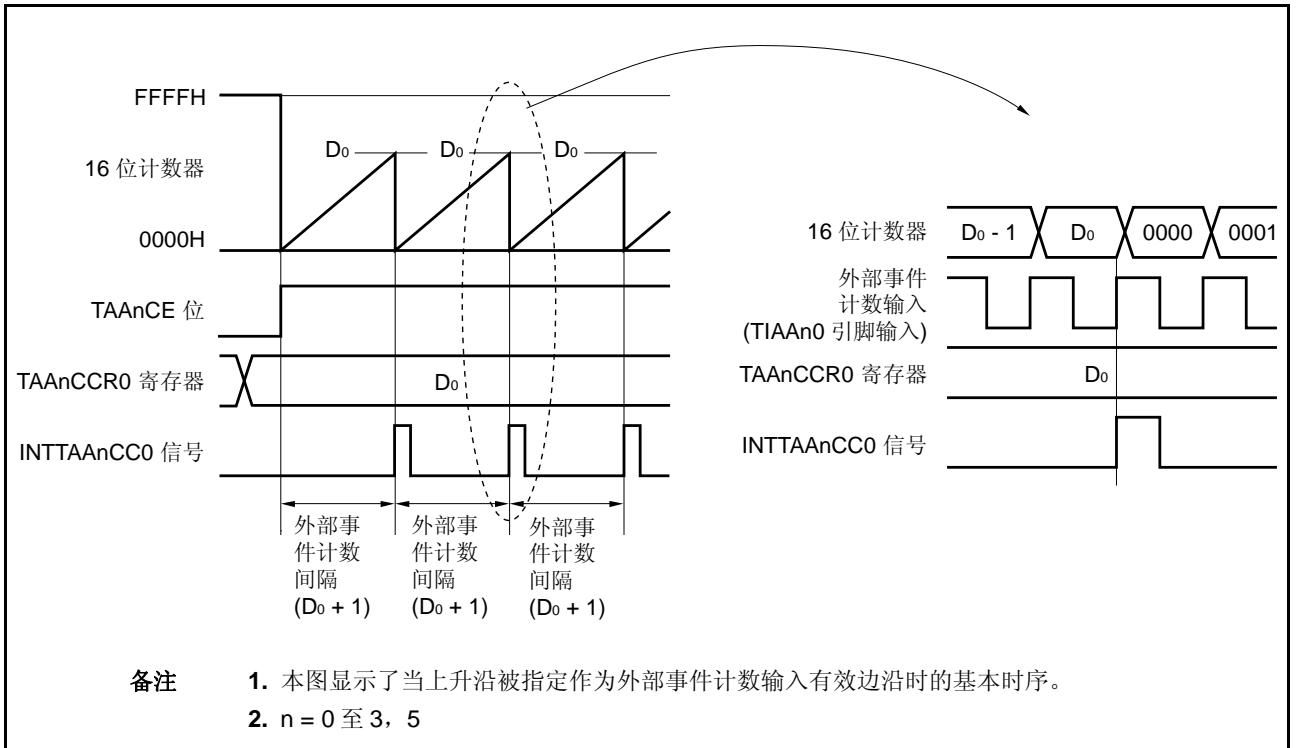


图 7-15. 外部事件计数模式的基本时序



当 TAA_nCE 位被置为 1 时，16 位计数器的值从 FFFFH 被清为 0000H。计数器对检测到的每一次外部事件计数输入的有效边沿进行计数。此外，TAA_nCCR0 寄存器的设定值被传送给 CCR0 缓冲寄存器。

当 16 位计数器的计数值和 CCR0 缓冲寄存器的数值相匹配时，16 位计数器清除为 0000H，同时生成一个比较匹配中断请求信号 (INTTAA_nCC0)。

每当外部事件计数输入的有效边沿数量达到 (TAA_nCCR0 寄存器设定值+1) 时，就会生成 INTTAA_nCC0 (比较匹配中断请求) 信号。

图 7-16. 外部事件计数模式下的寄存器操作设置 (1/2)

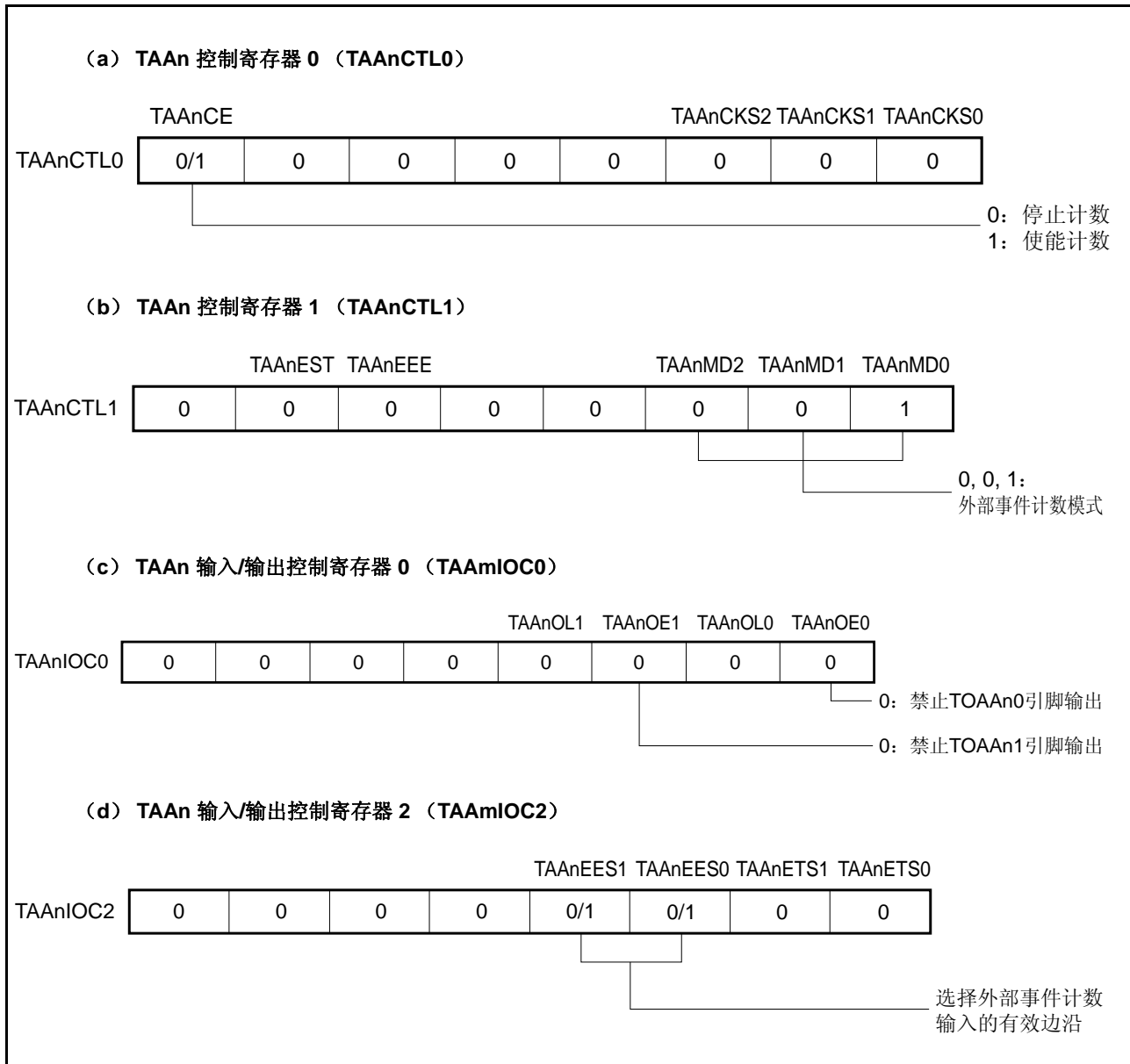


图 7-11. 外部事件计数模式下的寄存器操作设置 (2/2)

(e) TAA_n 计数器读取缓冲寄存器 (TAA_nCNT)

可以通过读取 TAA_nCNT 寄存器，读取 16 位计数器的计数值。

(f) TAA_n 捕获/比较寄存器 0 (TAA_nCCR0)

如果 TAA_nCCR0 寄存器的值被设定为 D_0 ，当外部事件计数值达到 (D_0+1) 时，计数器被清 0，同时产生一个比较匹配中断请求信号 (INTTAA_nCC0)。

(g) TAA_n 捕获/比较寄存器 1 (TAA_nCCR1)

通常在外部事件计数器模式中不使用 TAA_nCCR1 寄存器。但是，TAA_nCCR1 寄存器的设定值仍然被传送给 CCR1 缓冲寄存器。当 16 位计数器的计数值和 CCR1 缓冲寄存器的值匹配时，会产生一个比较匹配中断请求信号 (INTTAA_nCC1)。

所以，需要使用中断屏蔽标志 (TAA_nCCMK1) 来屏蔽该中断请求信号。

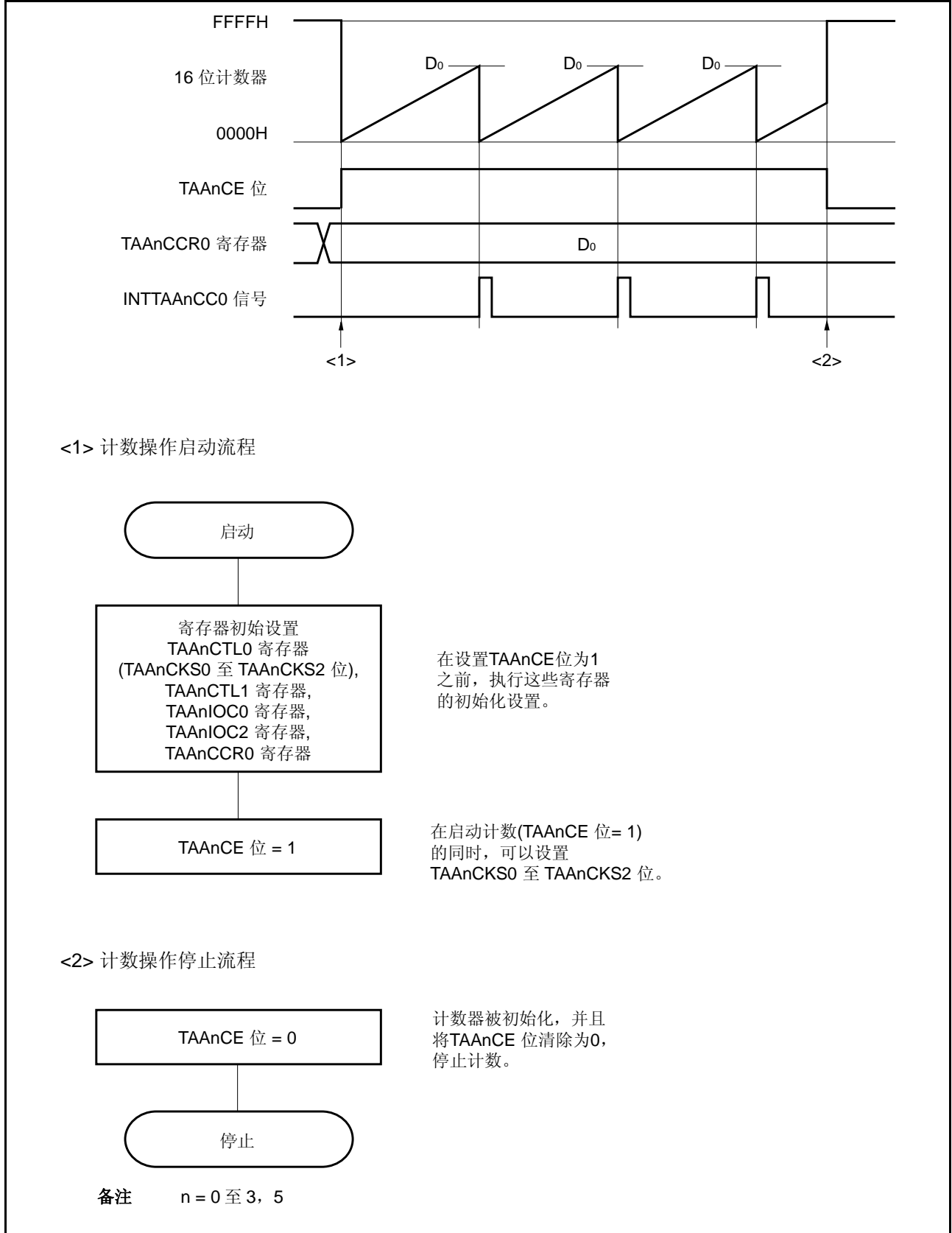
注意事项 当外部时钟用作计数时钟时，外部时钟只能从 TIAA_n0 引脚输入。此时，将 TAA_nIOC1.TAA_nIS1 和 TAA_nIOC1.TAA_nIS0 位清为 00 (捕获触发输入 (TIAA_n0 引脚)：无边沿检测)。

备注

1. 在外部事件计数模式下，不使用 TAA_n 输入/输出控制寄存器 1 (TAA_nIOC1) 和 TAA_n 选项寄存器 0 (TAA_nOPT0)。
2. $n = 0$ 至 3, 5

(1) 外部事件计数模式的操作系统

图 7-17. 外部事件计数模式下的软件处理流程



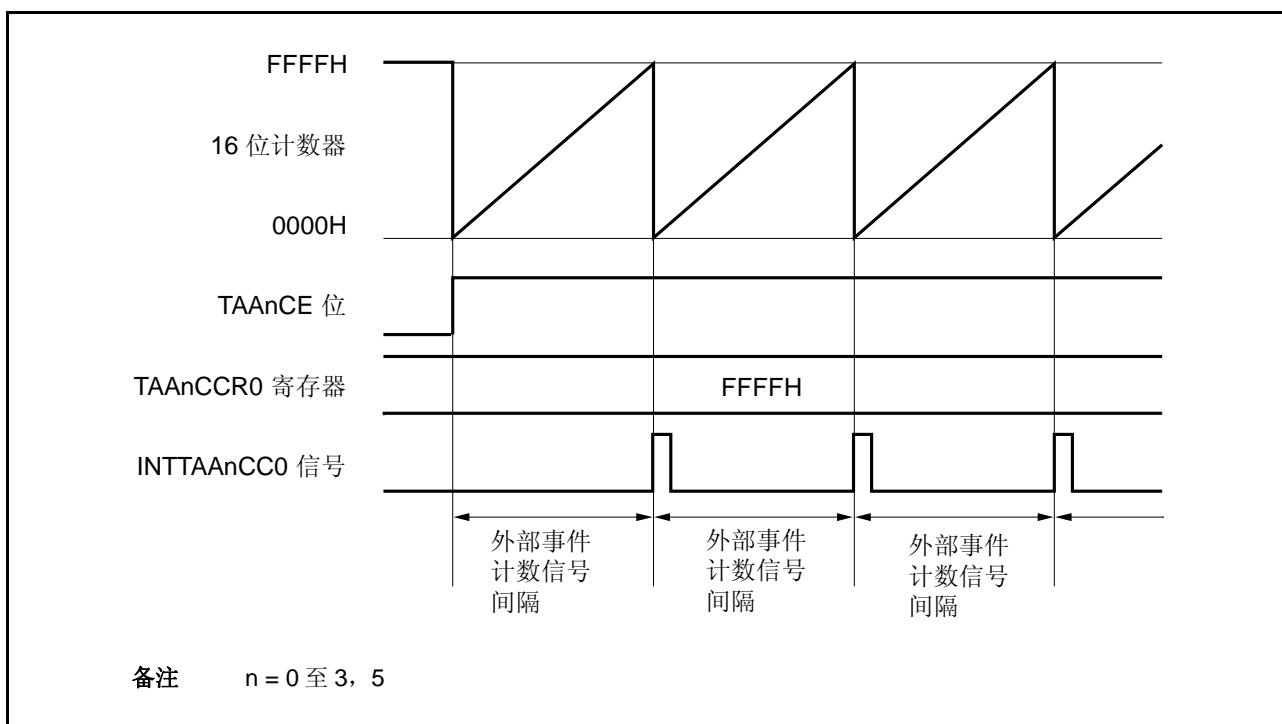
(2) 外部事件计数模式的操作时序

注意事项 1. 外部事件计数模式下，不要将 TAAAnCCR0 寄存器设置为 0000H。

2. 外部事件计数模式下，禁止使用定时器输出。如果使用外部事件计数输入来进行定时器输出，应设置为间隔定时器模式，并且选择外部事件计数输入允许的操作作为计数时钟 (TAAAnCTL1.TAAAnMD2 至 TAAAnCTL1.TAAAnMD0 位= 000, TAAAnCTL1.TAAAnEEE 位= 1)。

(a) 如果 TAAAnCCR0 寄存器被设置为 FFFFH 时的操作

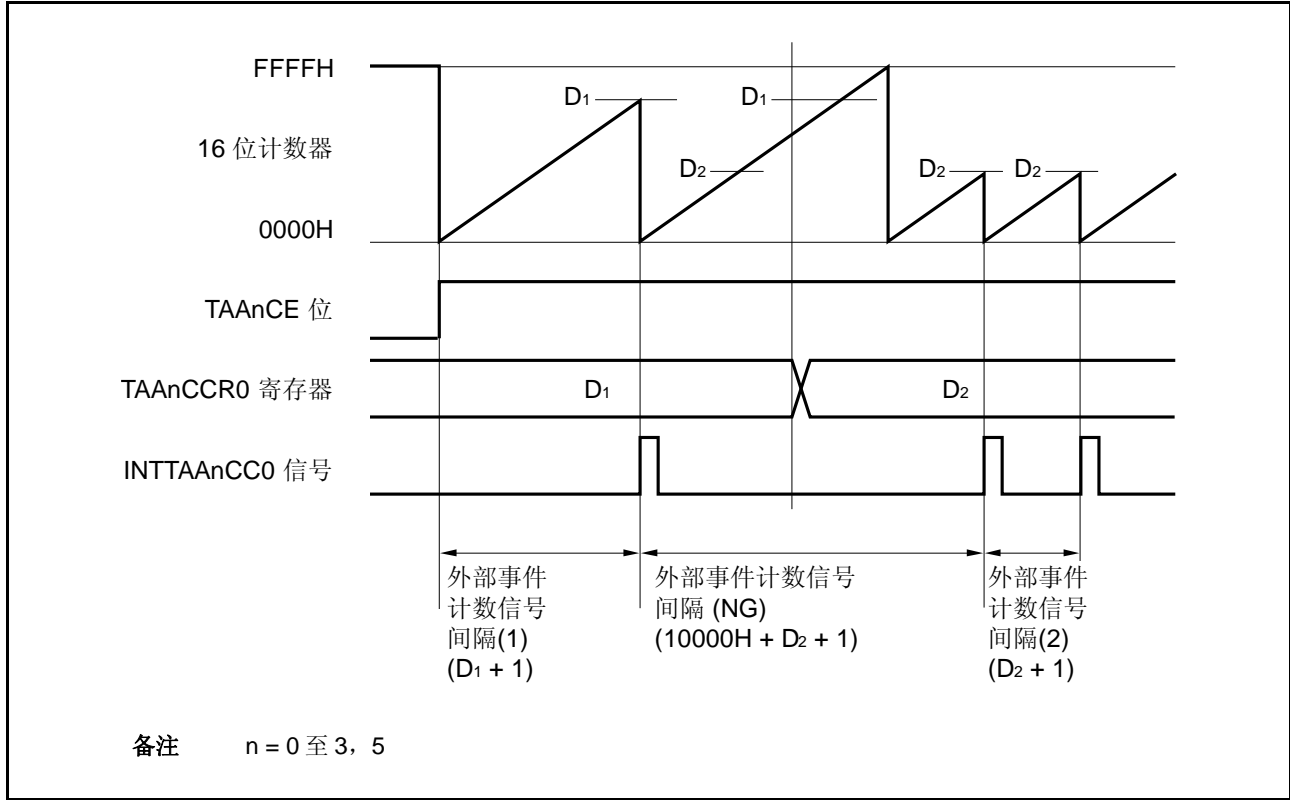
如果 TAAAnCCR0 寄存器被设置为 FFFFH，16 位计数器每次检测到的外部事件输入信号有效边沿时，都计数到 FFFFH。当下一个计数到来时，16 位计数器被同步清为 0000H，并且产生 INTTAAAnCC0 信号。此时，不设置 TAAAnOPT0.TAAAnOVF 位。



(b) 重写 TAA_nCCR0 寄存器的注意事项

要将 TAA_nCCR0 寄存器的值改为比当前值较小的值，应停止计数，然后改变其设定值。

如果在计数器工作期间，将 TAA_nCCR0 寄存器的值重写为比当前值较小的值，16 位计数器可能会溢出。



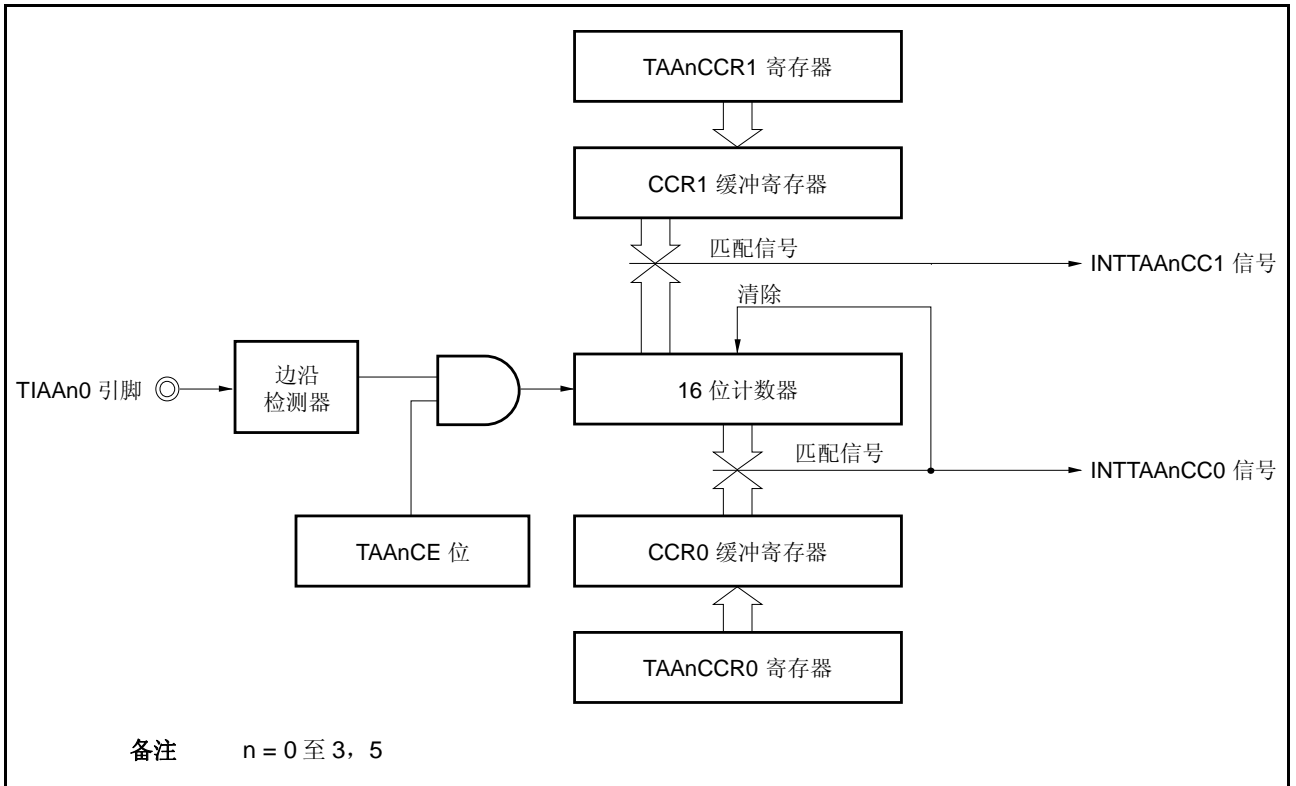
当 16 位计数器计数值大于 D₂ 而小于 D₁ 时，如果将 TAA_nCCR0 寄存器的值从 D₁ 变为 D₂，则在重写 TAA_nCCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。因而，与 16 位计数器相比较的值会是 D₂。

但是，由于计数值已经超过了 D₂，所以 16 位计数器就会向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值与 D₂ 匹配时，会生成 INTTAA_nCC0 信号。

因此，可能不会在原先预计的“(D₁+1)次”或“(D₂+1)次”有效边沿生成 INTTAA_nCC0 中断请求信号，而可能在“(10000H+D₂+1)次”有效边沿生成中断请求信号。

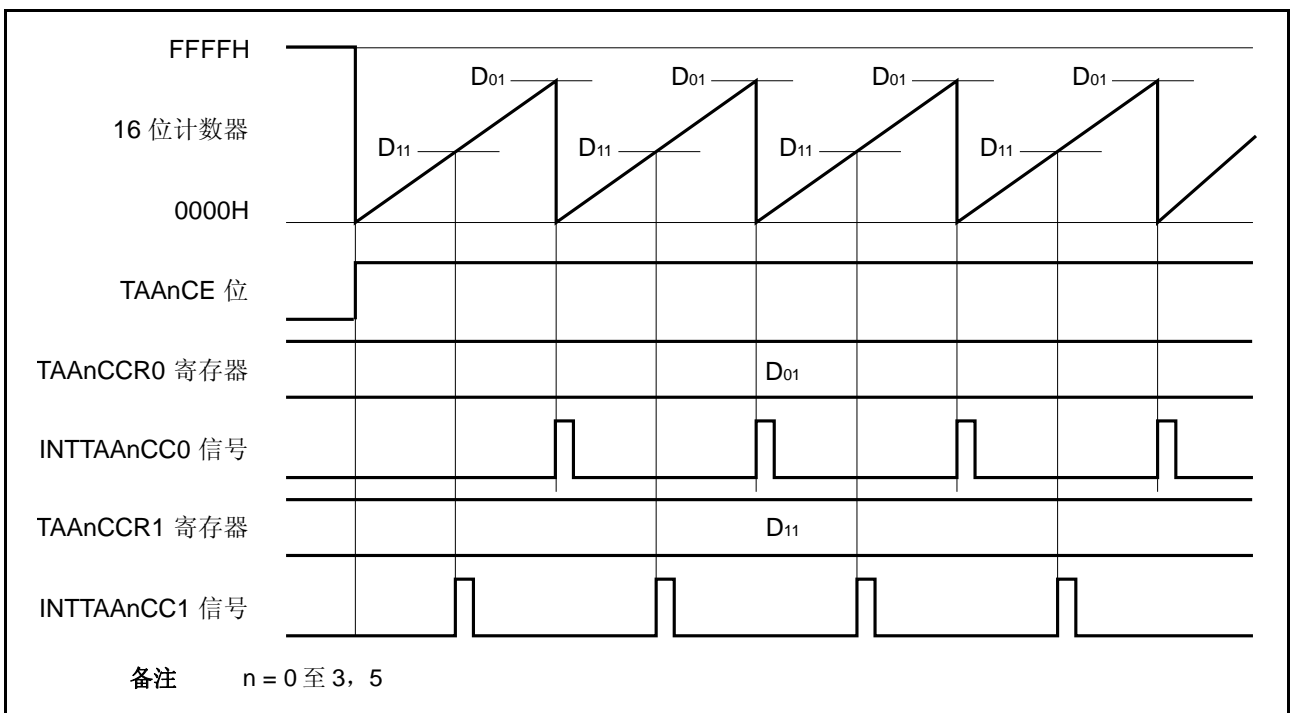
(c) TAA_nCCR1 寄存器的操作

图 7-18. TAA_nCCR1 寄存器的配置



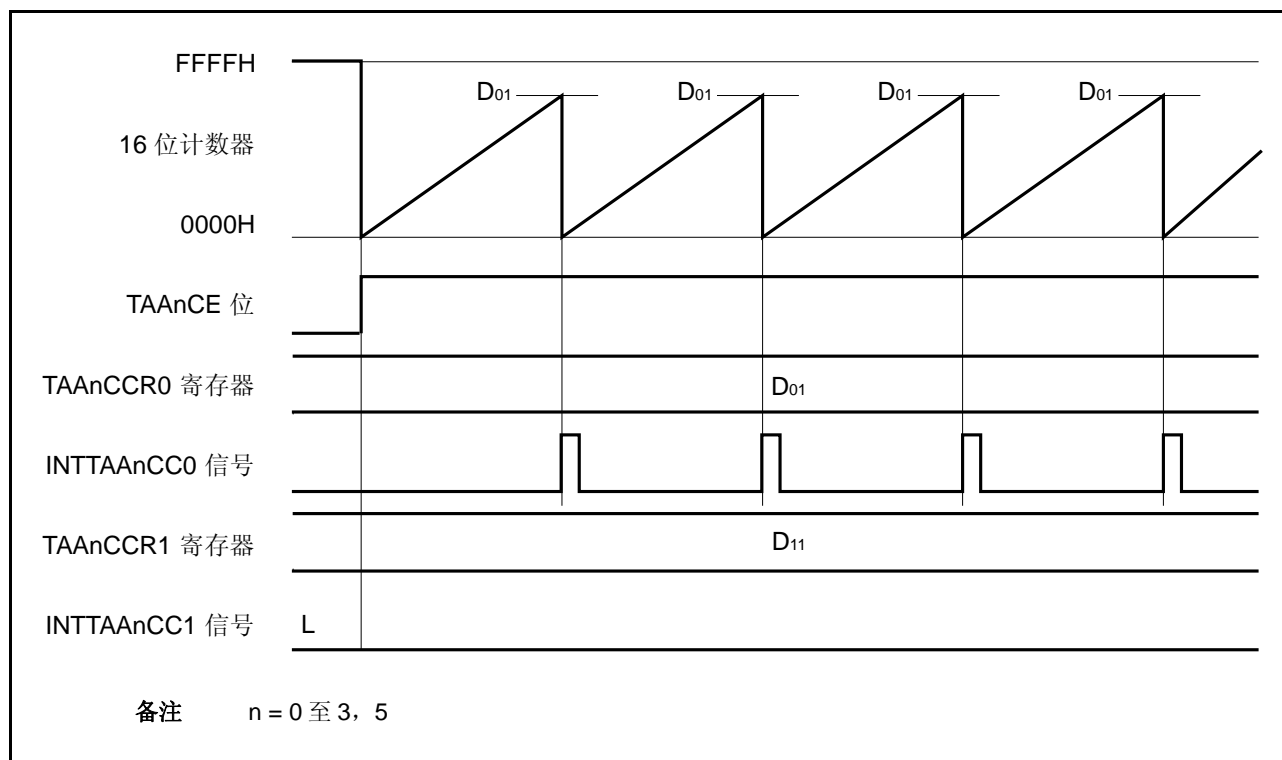
如果 TAA_nCCR1 寄存器的设定值小于 TAA_nCCR0 寄存器的设定值，则每个计数周期都产生一次 INTTAA_nCC1 中断请求信号。

图 7-19. 当 D₀₁ ≥ D₁₁ 时的时序图



如果 TAAAnCCR1 寄存器的设定值大于 TAAAnCCR0 寄存器的设定值，16 位计数器的计数值不会和 TAAAnCCR1 寄存器的值发生匹配。因此也就不会产生 INTTAAAnCC1 中断请求信号。

图 7-20. 当 $D_{01} < D_{11}$ 时的时序图



7.5.3 外部触发脉冲输出模式 (TAA_nMD2 至 TAA_nMD0 位= 010)

在外部触发脉冲输出模式下，当 TAA_nCTL0.TAA_nCE 位被置为 1 时，16 位定时器/事件计数器 AA 等待触发信号，当检测到外部触发输入信号的有效边沿后，16 位定时器/事件计数器 AA 开始计数，并且从 TOA_n1 引脚输出一个 PWM 波形。

不使用外部触发，而是使用软件触发，同样可以得到输出脉冲。当使用软件触发时，TOA_n0 引脚输出一个方波，其半周期等于输出 PWM 的一个周期。

图 7-21. 外部触发脉冲输出模式的配置

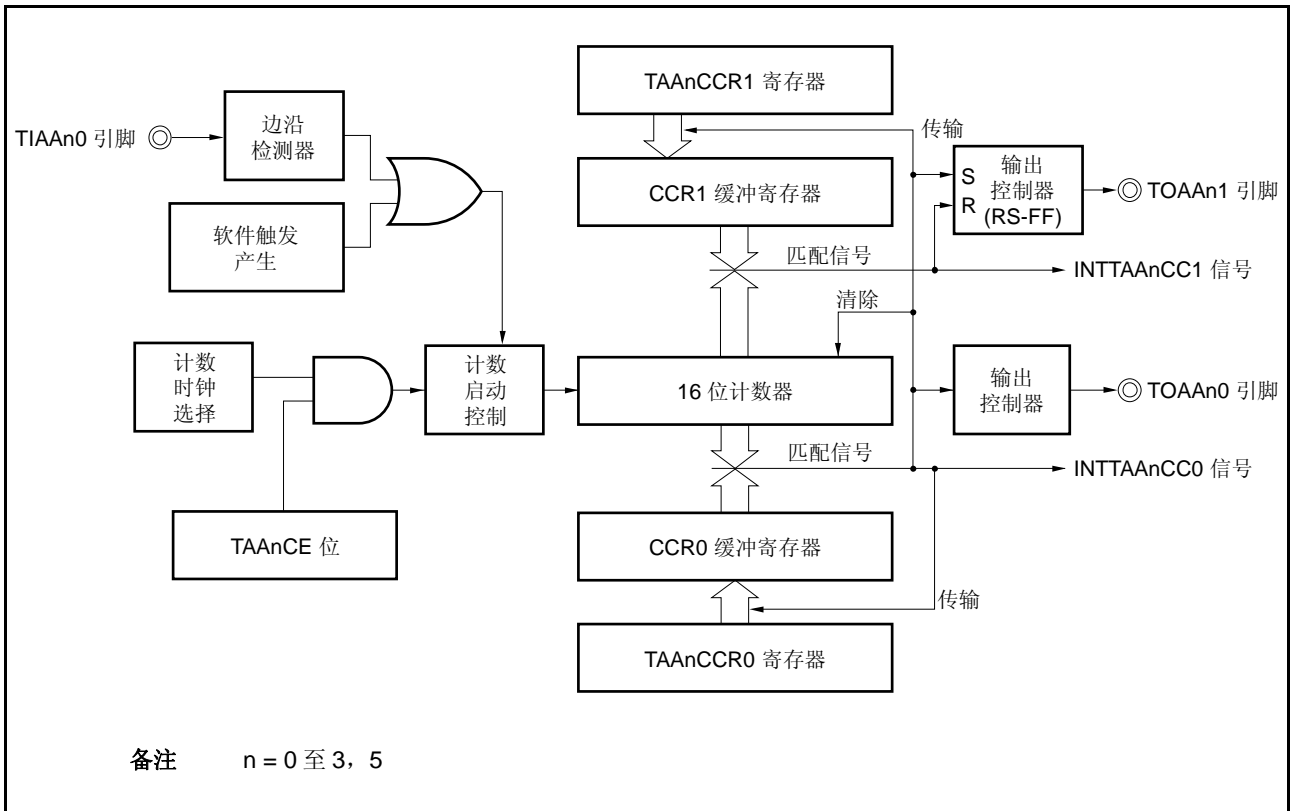
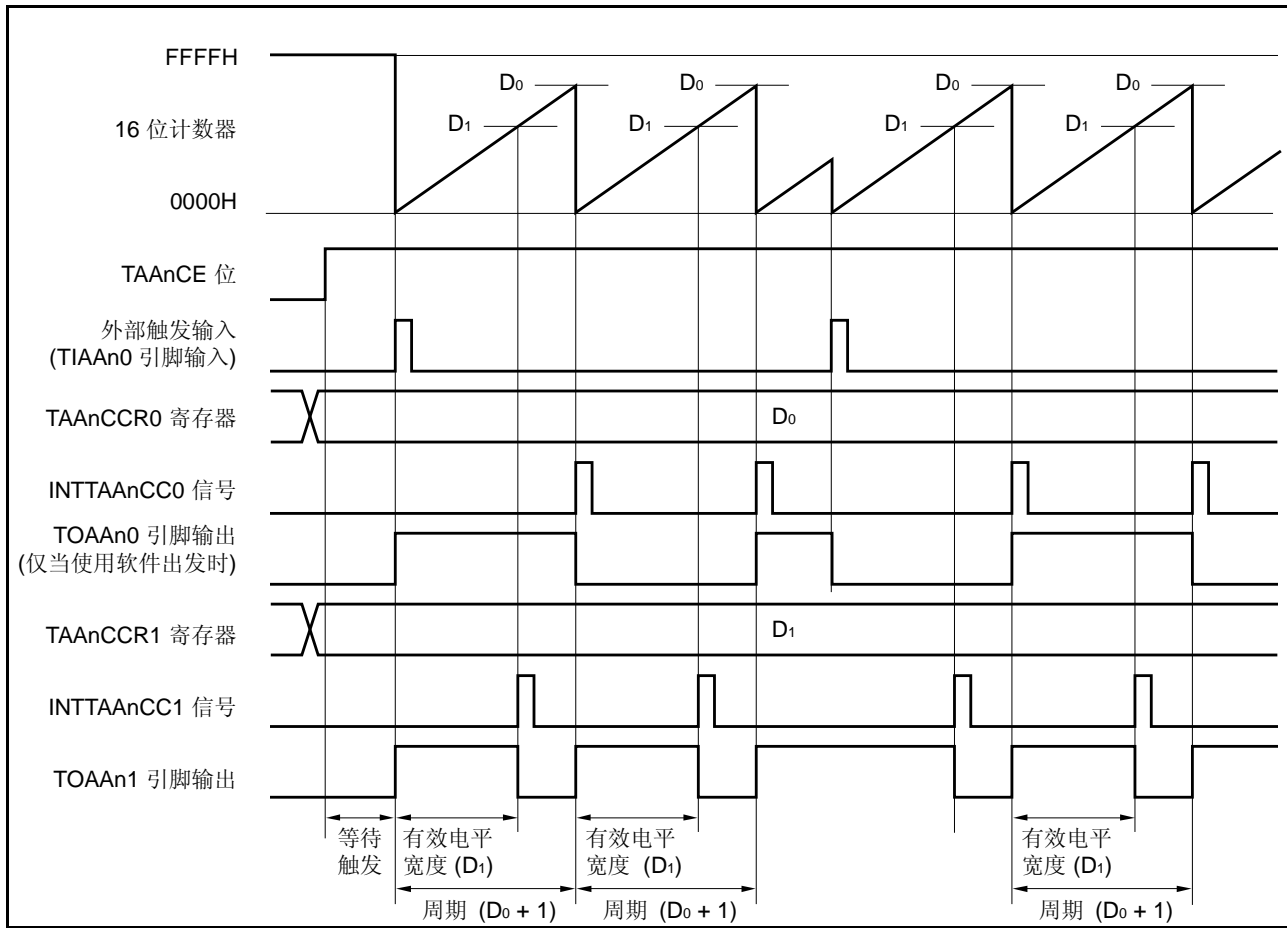


图 7-22. 外部触发脉冲输出模式的基本时序



当 TAAAnCE 位设置为 1 时，16 位定时器/事件计数器 AA 等待触发。当触发产生时，16 位计数器从 FFFFH 清除为 0000H，同时开始计数，并从 TOAAAn1 引脚输出一个 PWM 波形。如果在计数运行中再次产生触发，则计数器清为 0000H 并重新开始计数。（TOAAAn0 引脚输出反相。在出现触发时，不管当前状态如何（高/低），TOAAAn1 引脚都会输出一个高电平。

PWM 波形的有效电平宽度、周期、占空比的计算如下。

$$\begin{aligned} \text{有效电平宽度} &= (\text{TAAAnCCR1 寄存器的设置值}) \times \text{计数时钟周期} \\ \text{周期} &= (\text{TAAAnCCR0 寄存器的设置值} + 1) \times \text{计数时钟周期} \\ \text{占空比系数} &= (\text{TAAAnCCR1 寄存器的设置值}) / (\text{TAAAnCCR0 寄存器的设置值} + 1) \end{aligned}$$

当 16 位计数器在其计数值与 CCR0 缓冲寄存器的值匹配之后，进行下一次计数时就会同步产生比较匹配中断请求信号 INTTAAAnCC0，且 16 位计数器同时被清除为 0000H。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTAAAnCC1。

当 16 位计数器的计数值与 CCRm 缓冲寄存器的值匹配，而且 16 位计数器清除为 0000H 时，TAAAnCCRm 寄存器的设定值就被传送到 CCRm 缓冲寄存器。

外部触发输入信号的有效边沿，或将软件触发（TAAAnCTL1.TAAAnEST 位）设为 1，都可以用作触发信号。

备注 n = 0 至 3, 5, m = 0, 1

图 7-23. 外部触发脉冲输出模式的寄存器设置 (1/2)

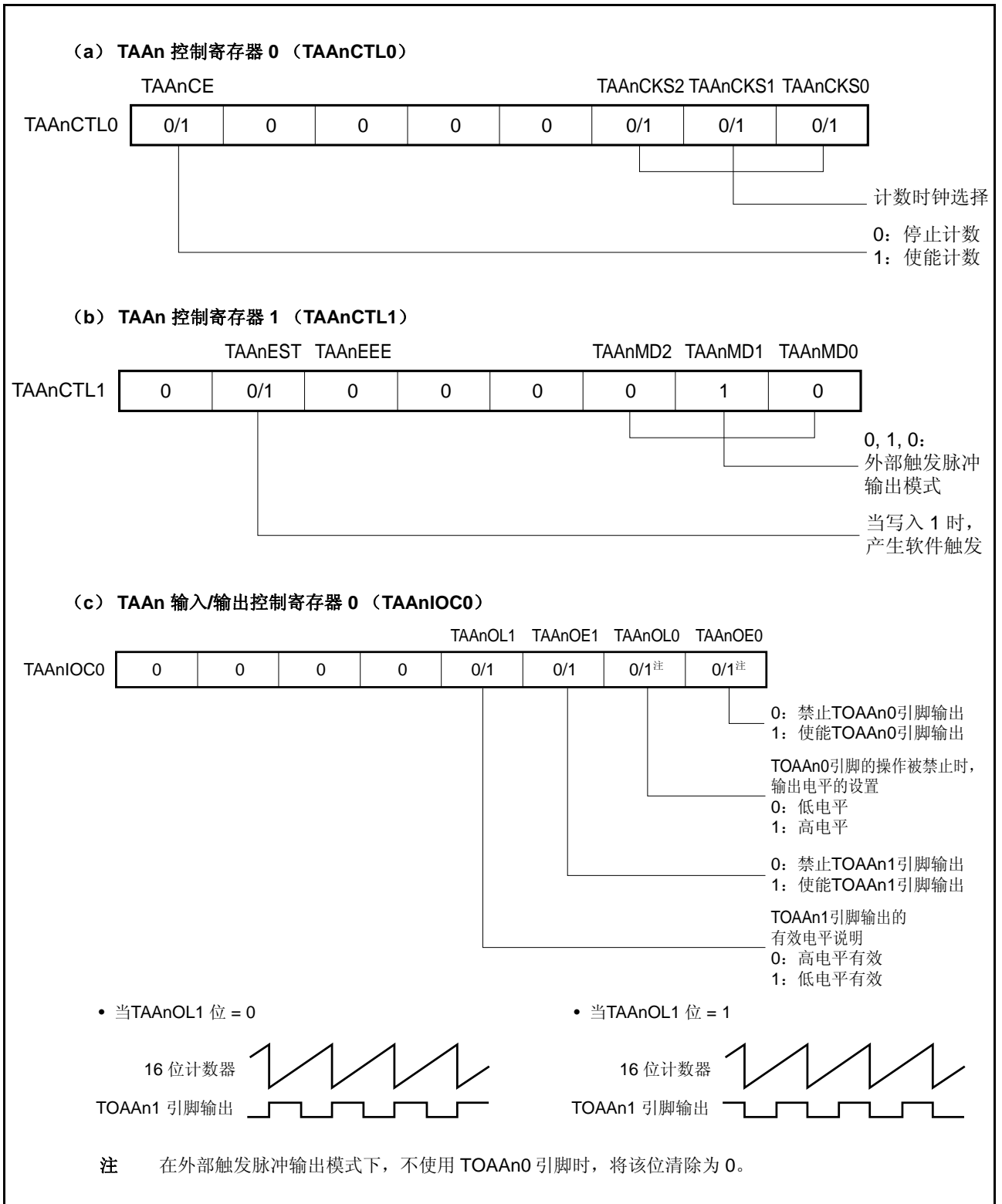


图 7-23. 外部触发脉冲输出模式的寄存器设置 (2/2)



(1) 外部触发脉冲输出模式的操作系统

图 7-24. 外部触发脉冲输出模式的软件处理流程 (1/2)

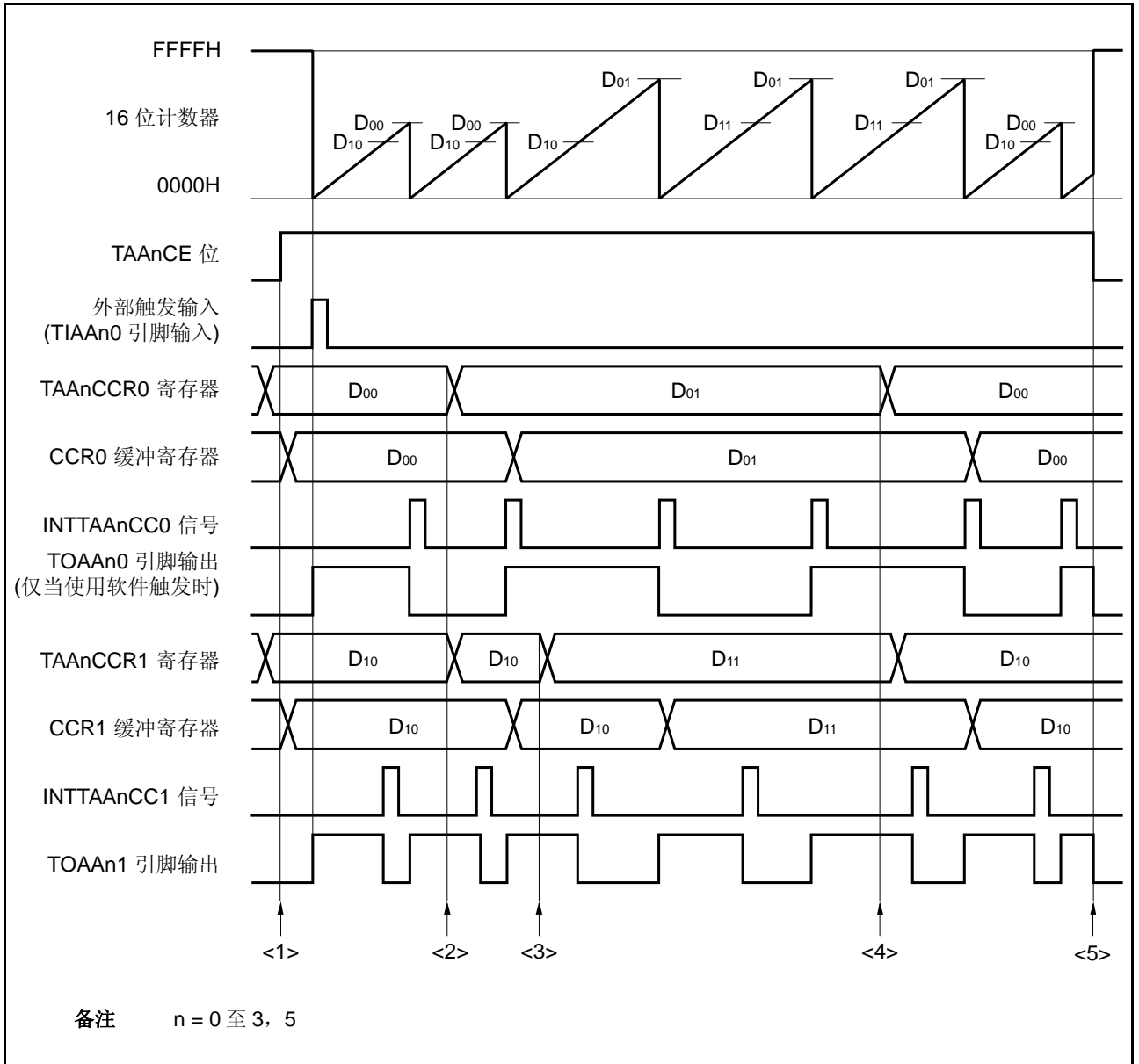
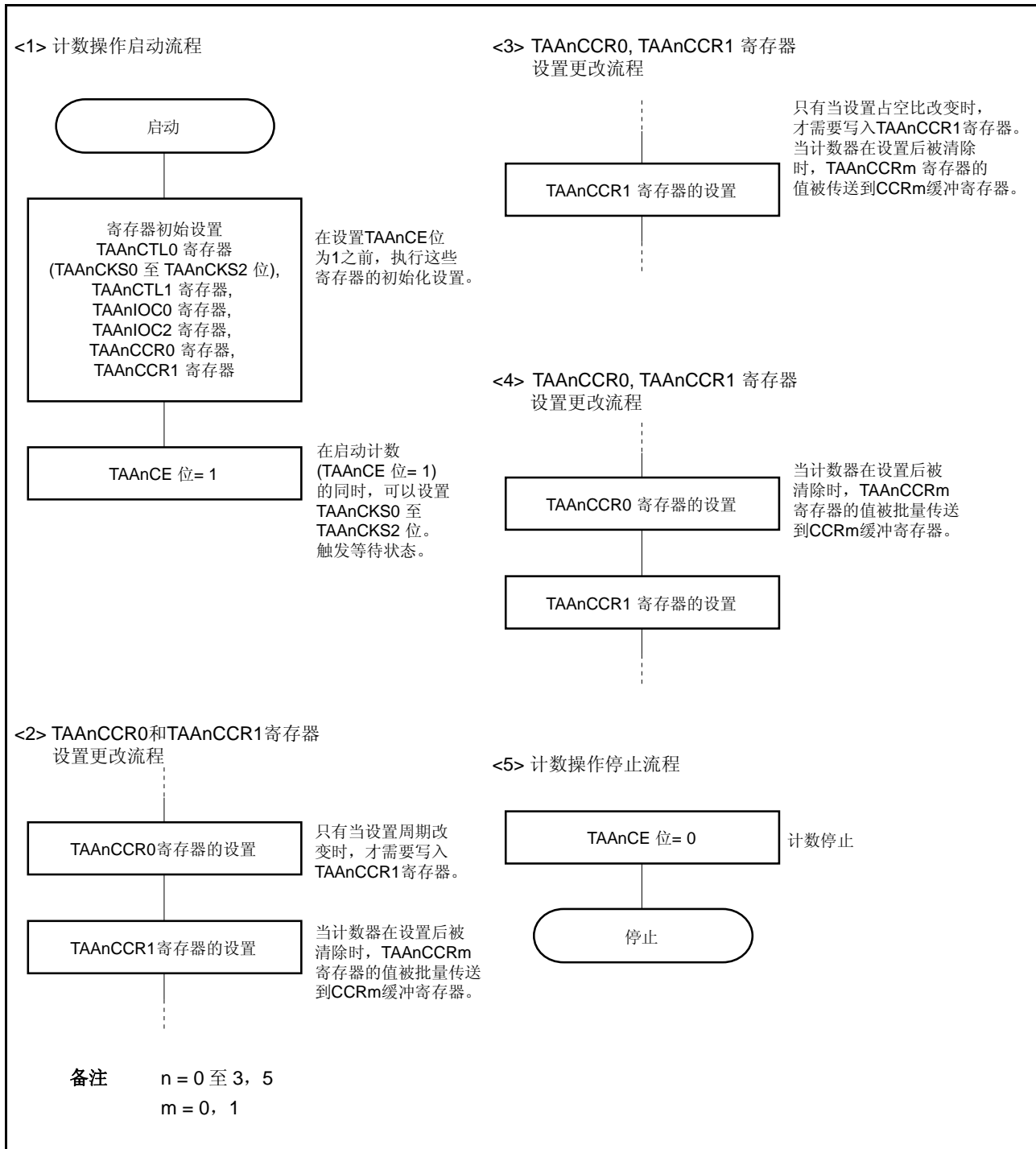


图 7-24. 外部触发脉冲输出模式的软件处理流程 (2/2)

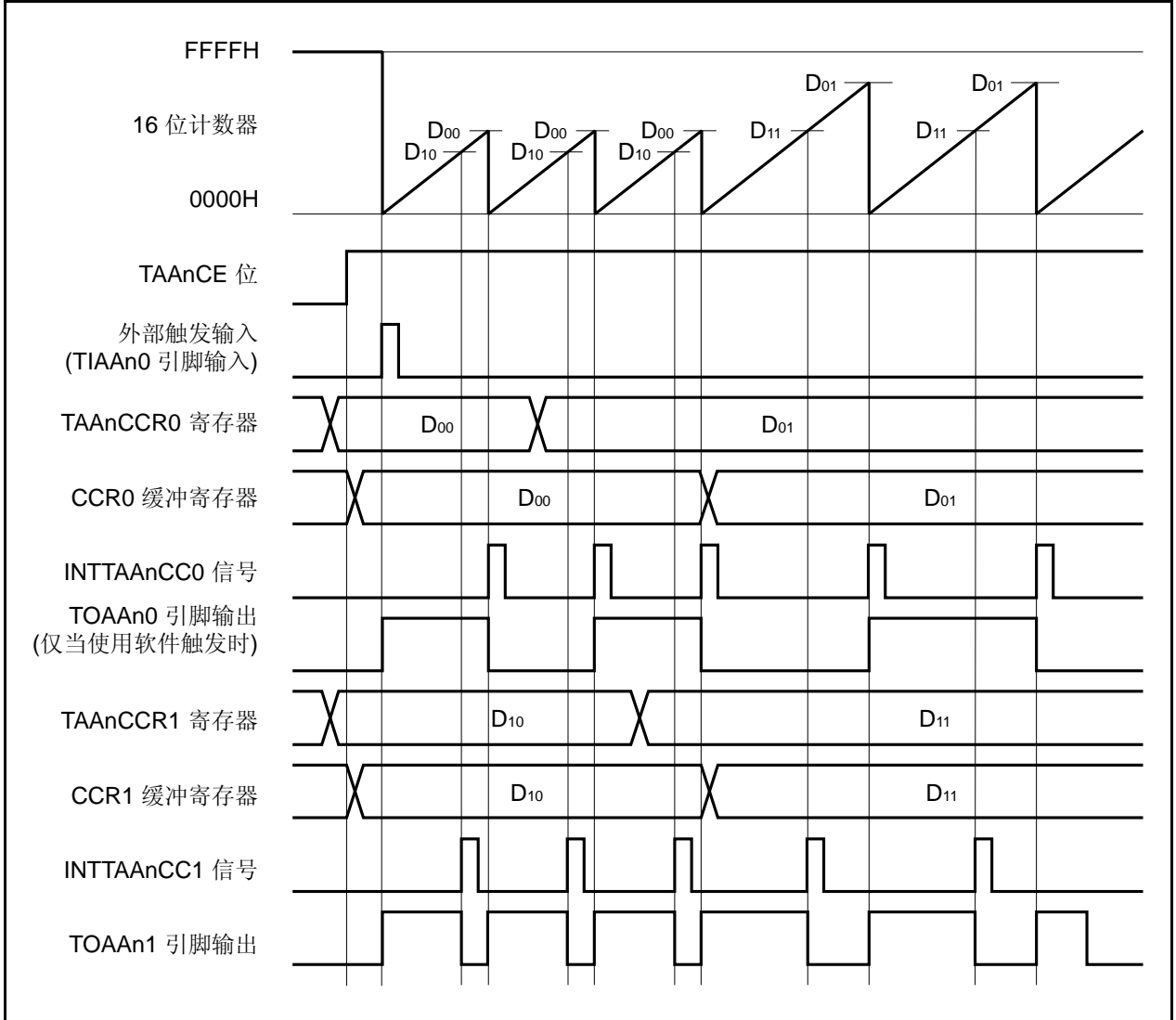


(2) 外部触发脉冲输出模式的操作时序

(a) 操作中改变脉冲宽度的注意事项

计数器计数过程中，若要改变 PWM 波形，应重新写入 TAA_nCCR1 寄存器。

当检测到 INTTAA_nCC0 信号以后，在 TAA_nCCR1 寄存器写入操作之后重写 TAA_nCCR_m 寄存器。



为了将数据从 TAA_nCCR_m 寄存器传送到 CCR_m 缓冲寄存器，必须写入 TAA_nCCR1 寄存器。

如果 PWM 波形的周期和有效电平宽度都要改变，应先将周期设定到 TAA_nCCR0 寄存器，然后将有效电平宽度设定到 TAA_nCCR1 寄存器。

如果只改变 PWM 波形的周期，应先把周期设定给 TAA_nCCR0 寄存器，然后将相同值（原值）写入 TAA_nCCR1 寄存器。

只改变 PWM 波形的有效电平宽度（占空比）时，仅需要设置 TAA_nCCR1 寄存器。

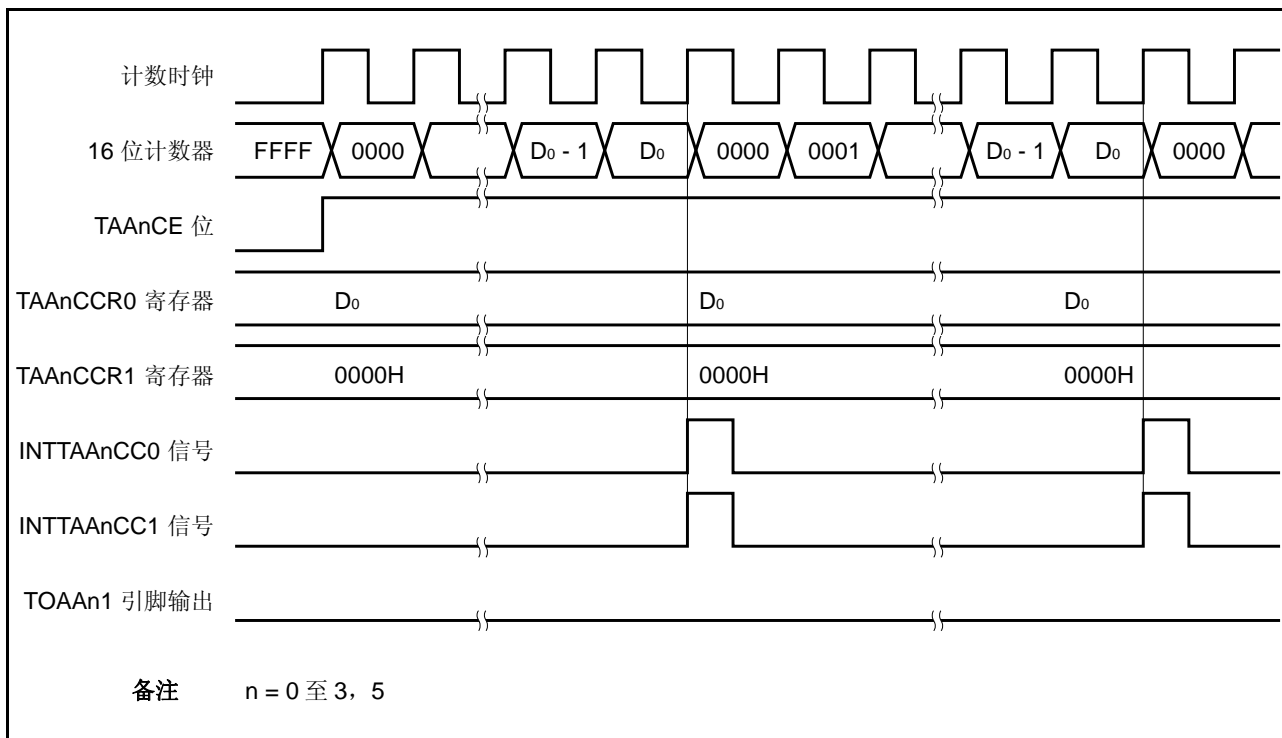
当数据写入 TAA_nCCR1 寄存器之后，在对 16 位计数器清零的同时，写入到 TAA_nCCR_m 寄存器的值被传送到 CCR_m 缓冲寄存器，并作为与 16 位计数器进行比较的值。

在写入 TAA_nCCR1 寄存器一次之后，要再次写 TAA_nCCR0 或 TAA_nCCR1 寄存器，应该在 INTTAA_nCC0 信号产生之后进行。否则，CCR_m 缓冲寄存器的值可能不确定，因为把数据从 TAA_nCCR_m 寄存器传送到 CCR_m 缓冲寄存器的时序会和写入 TAA_nCCR_m 寄存器的时序发生冲突。

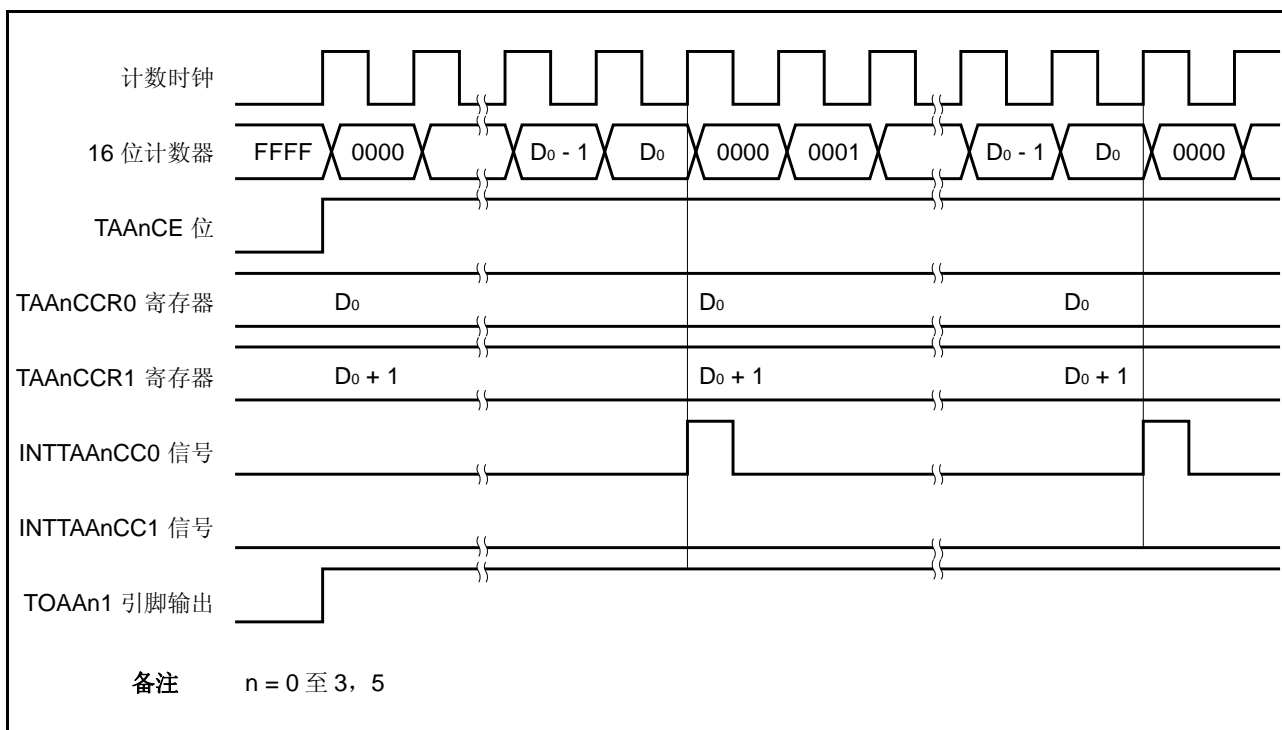
备注 n = 0 至 3, 5
 m = 0, 1

(b) PWM 波形的 0%和 100% 输出

为了输出 0%的波形，需要将 TAAAnCCR1 寄存器设置为 0000H。如果 TAAAnCCR0 寄存器的设定值为 FFFFH，则会在每个周期产生 INTTAAAnCC1 中断请求信号。

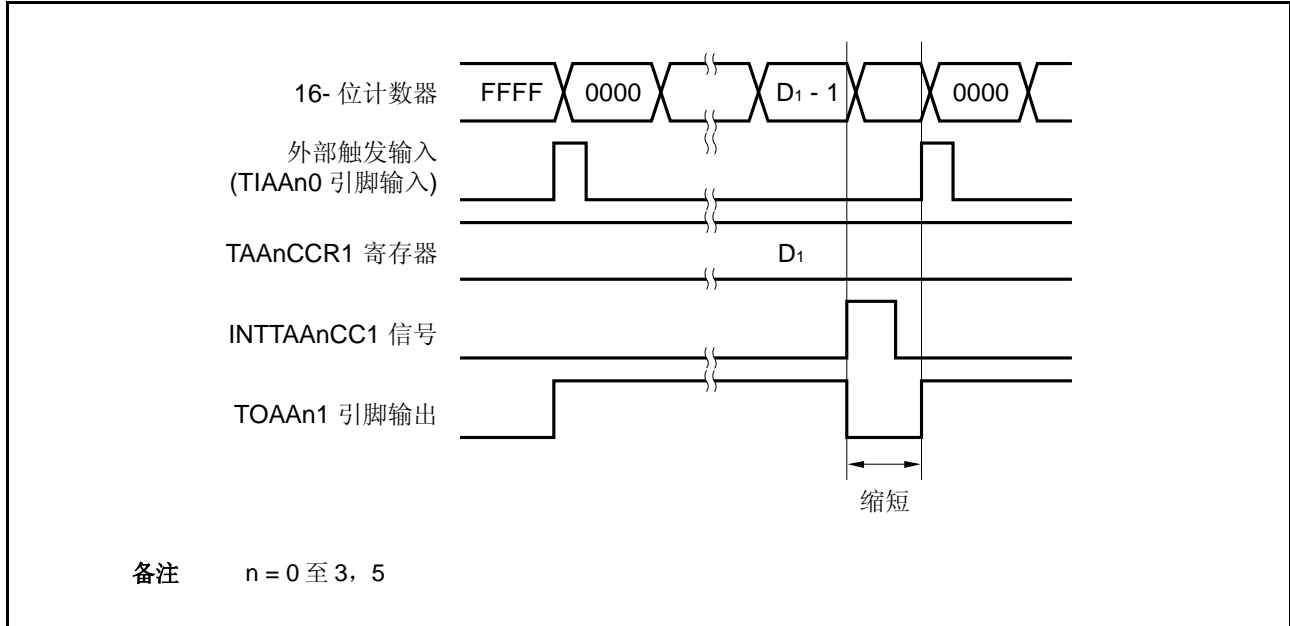


为了输出 100%的波形，需要将 TAAAnCCR1 寄存器的值设为：(TAAAnCCR0 寄存器的设定值+1)。如果 TAAAnCCR0 寄存器的设定值为 FFFFH，则不会产生 100%波形输出。

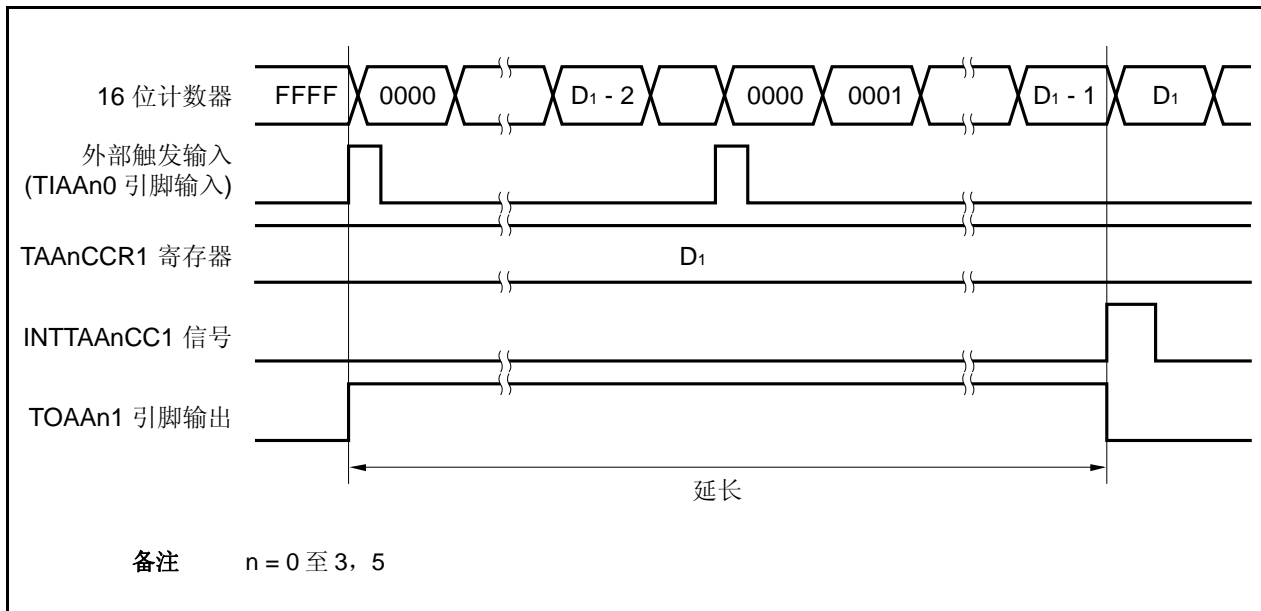


(c) 触发检测和 TAAAnCCR1 寄存器匹配之间的冲突

如果在 INTTAAAnCC1 信号产生后，立刻检测到触发信号，则 16 位计数器同时被清除为 0000H，TOAAAn1 引脚的输出信号生效，计数器继续计数。结果，PWM 波形的无效时段被缩短。

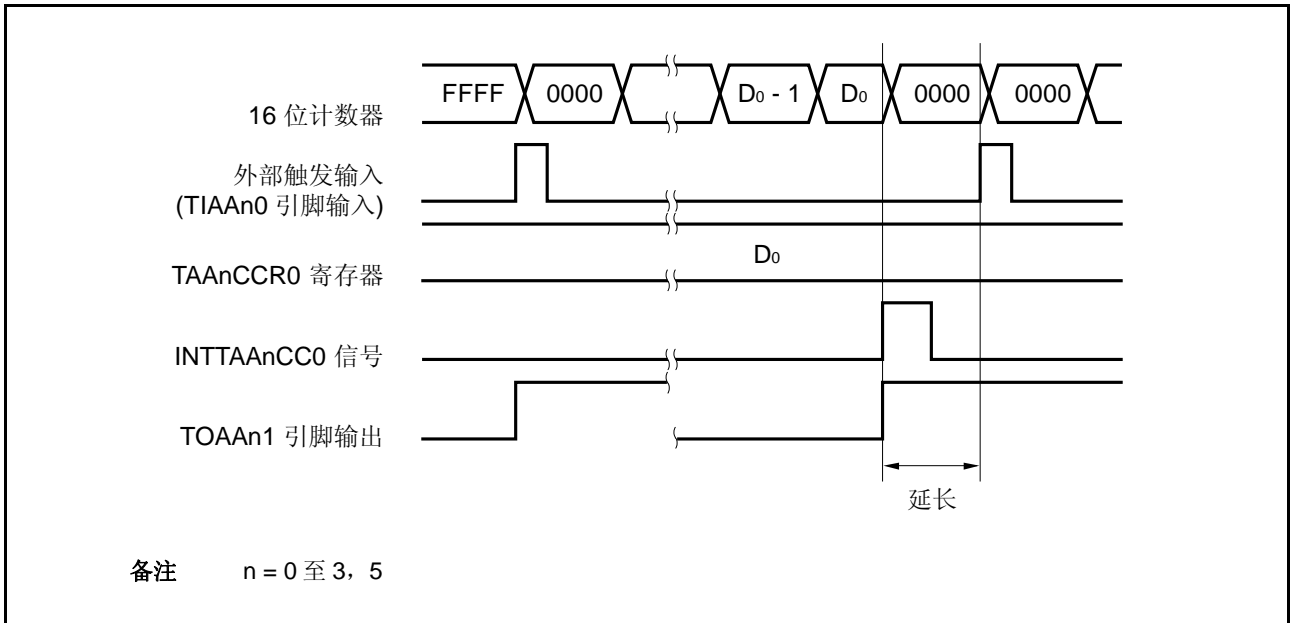


如果紧邻在 INTTAAAnCC1 信号产生之前检测到触发信号，则不会产生 INTTAAAnCC1 信号，而 16 位计数器清除为 0000H，计数继续进行。TOAAAn1 引脚的输出信号仍然保持有效。结果，PWM 波形的有效时段被延长。

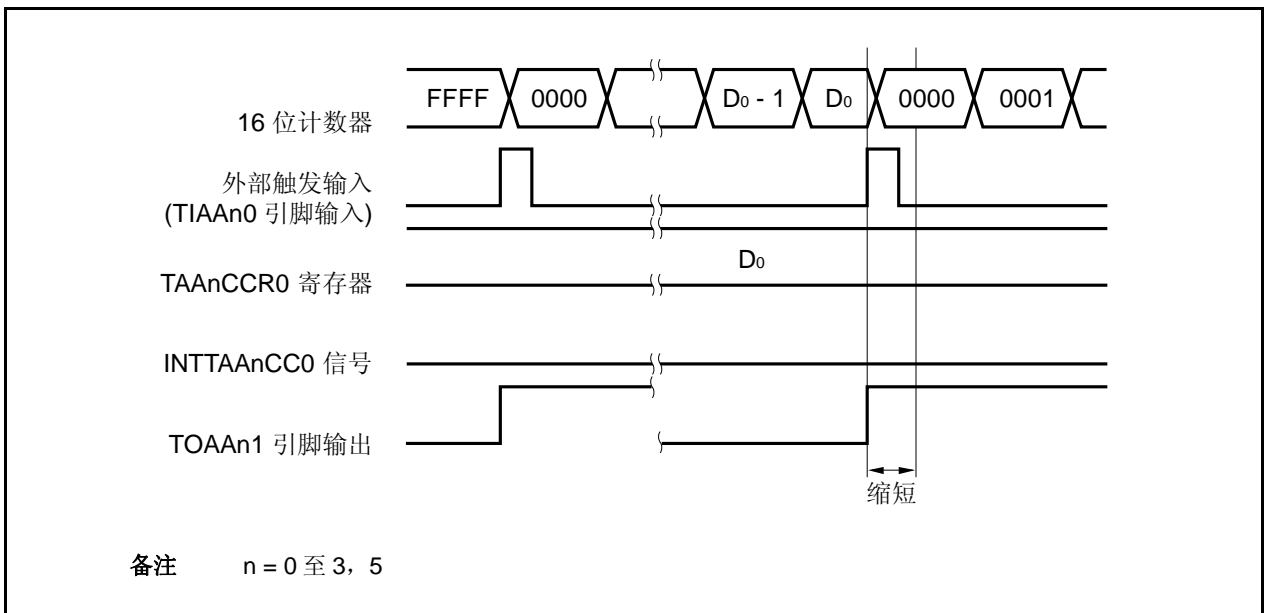


(d) 触发检测和 TAAAnCCR0 寄存器匹配之间的冲突

如果在 INTTAAAnCC0 信号产生后，立刻检测到触发信号，则 16 位计数器再次被清除为 0000H，并继续计数。因此，TOAAAn1 引脚的有效时段被延长，延长长度为 INTTAAAnCC0 信号产生时刻至检测到触发信号时刻的时间间隔。

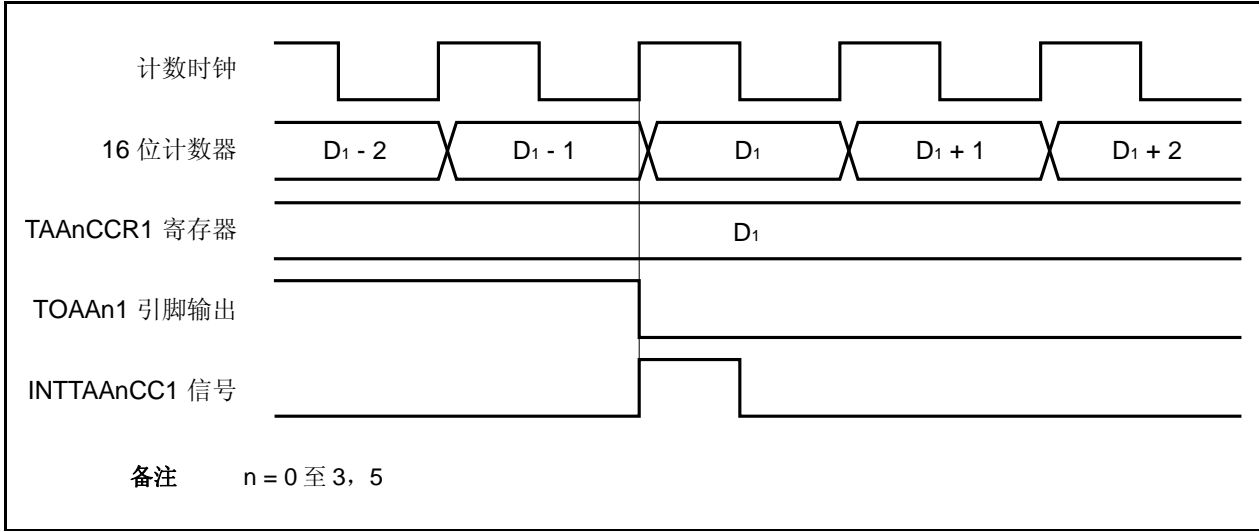


如果紧邻在 INTTAAAnCC0 信号产生之前检测到触发信号，则不会产生 INTTAAAnCC0 信号。16 位计数器清除为 0000H，TOAAAn1 引脚输出有效，计数器继续计数。结果，PWM 波形的无效时段被缩短。



(e) 比较匹配中断请求信号 (INTTAA_nCC1) 的发生时序

在外部触发脉冲输出模式下，INTTAA_nCC1 信号的发生时序与其它情况下 INTTAA_nCC1 信号的发生时序不同；当 16 位计数器的计数值与 TAA_nCCR1 寄存器的值匹配时，产生 INTTAA_nCC1 中断请求信号。



通常在 16 位计数器的计数值与 TAA_nCCR1 寄存器的值匹配之后，在下次计数时同步发生 INTTAA_nCC1 信号。

但是，在外部触发脉冲输出模式下，INTTAA_nCC1 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOA_n1 引脚输出信号变化的时序。

7.5.4 单次触发脉冲输出模式 (TAA_nMD2 至 TAA_nMD0 位=011)

在单次触发脉冲输出模式下, 当 TAA_nCTL0.TAA_nCE 位被置为 1 时, 16 位定时器/事件计数器 AA 等待触发。在检测到外部触发输入的有效边沿时, 16 位定时器/事件计数器 AA 开始计数, 同时从 TOAAn1 引脚输出一个单次触发脉冲。

除外部触发外, 也可以产生软件触发信号来输出该脉冲。在使用软件触发时, TOAAn0 引脚在 16 位计数器计数时输出有效电平, 在计数器停止 (等待触发) 时输出无效电平。

图 7-25. 单次触发脉冲输出模式的配置

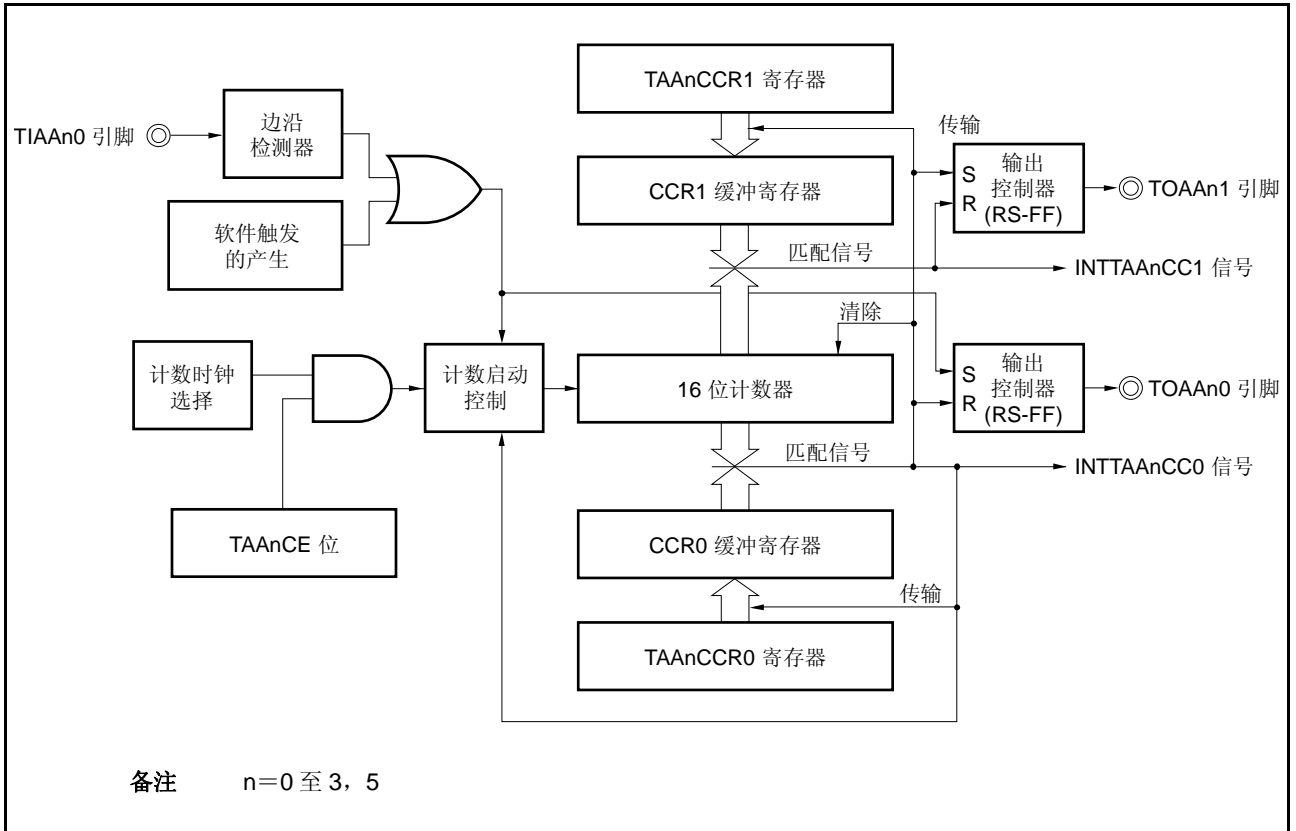
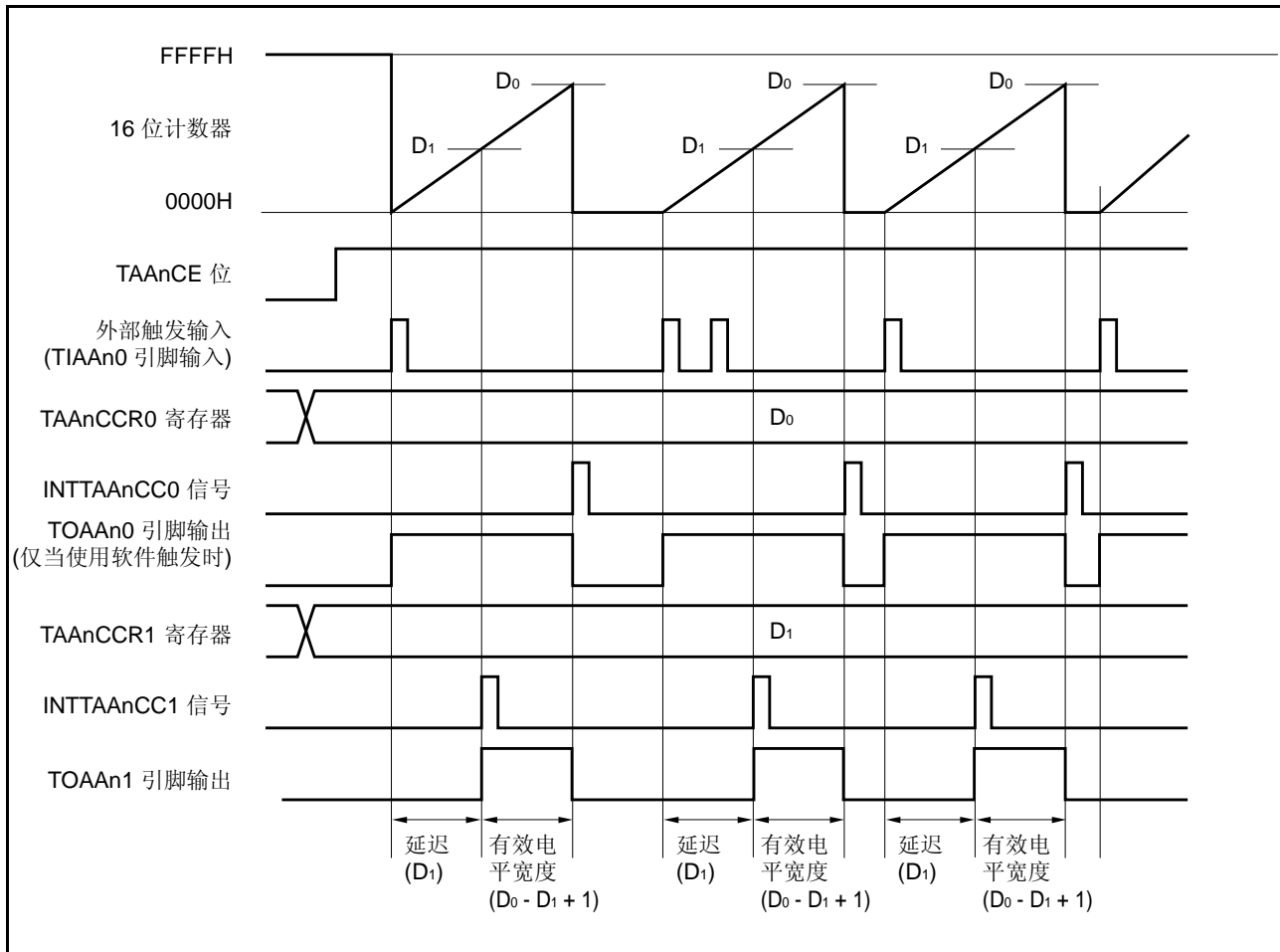


图 7-26. 单次触发脉冲输出模式的基本时序



当 TAAAnCE 位置为 1 时，16 位定时器/事件计数器 AA 等待触发。在触发信号产生时，16 位计数器从 FFFFH 清除为 0000H，开始进行计数，并从 TOAAAn1 引脚输出一个单次触发脉冲。在输出单次触发脉冲后，16 位计数器被设置为 FFFFH，计数停止，并重新等待触发。如果在单次触发脉冲输出期间再次产生触发信号，则该触发信号被忽略。

单次触发脉冲的输出延迟时间和有效电平宽度可计算如下：

$$\text{输出延迟时间} = (\text{TAAAnCCR1 寄存器的设置值}) \times \text{计数时钟周期}$$

$$\text{有效电平宽度} = (\text{TAAAnCCR0 寄存器的设置值} - \text{TAAAnCCR1 寄存器的设置值} + 1) \times \text{计数时钟周期}$$

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配之后，产生比较匹配中断请求信号 INTTAAAnCC0。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTAAAnCC1。

外部触发输入信号的有效边沿或将软件触发 (TAAAnCTL1.TAAAnEST 位) 信号设置为 1，都可被用作触发信号。

备注 n=0 至 3, 5

图 7-27. 单次触发脉冲输出模式时的寄存器设置 (1/2)

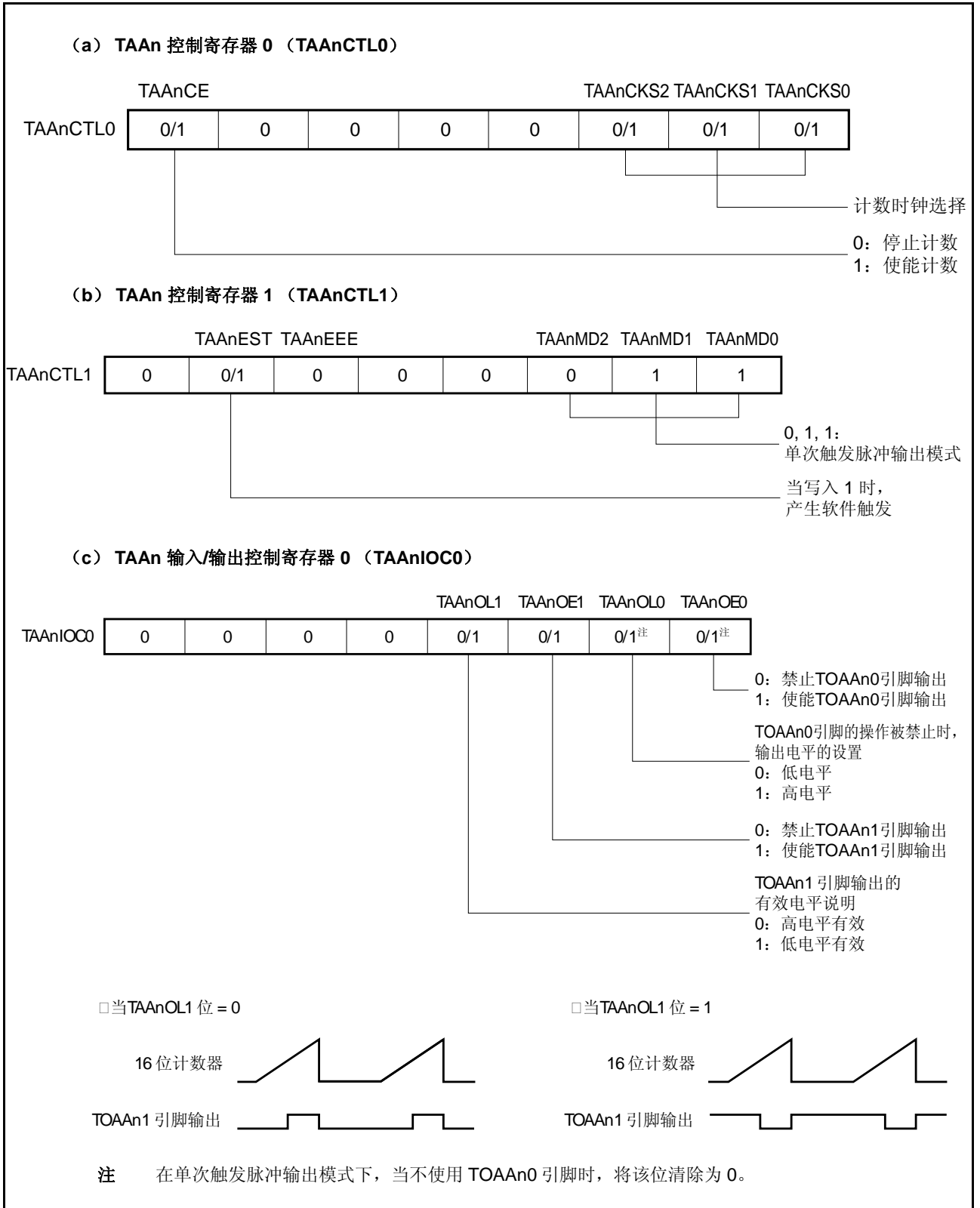
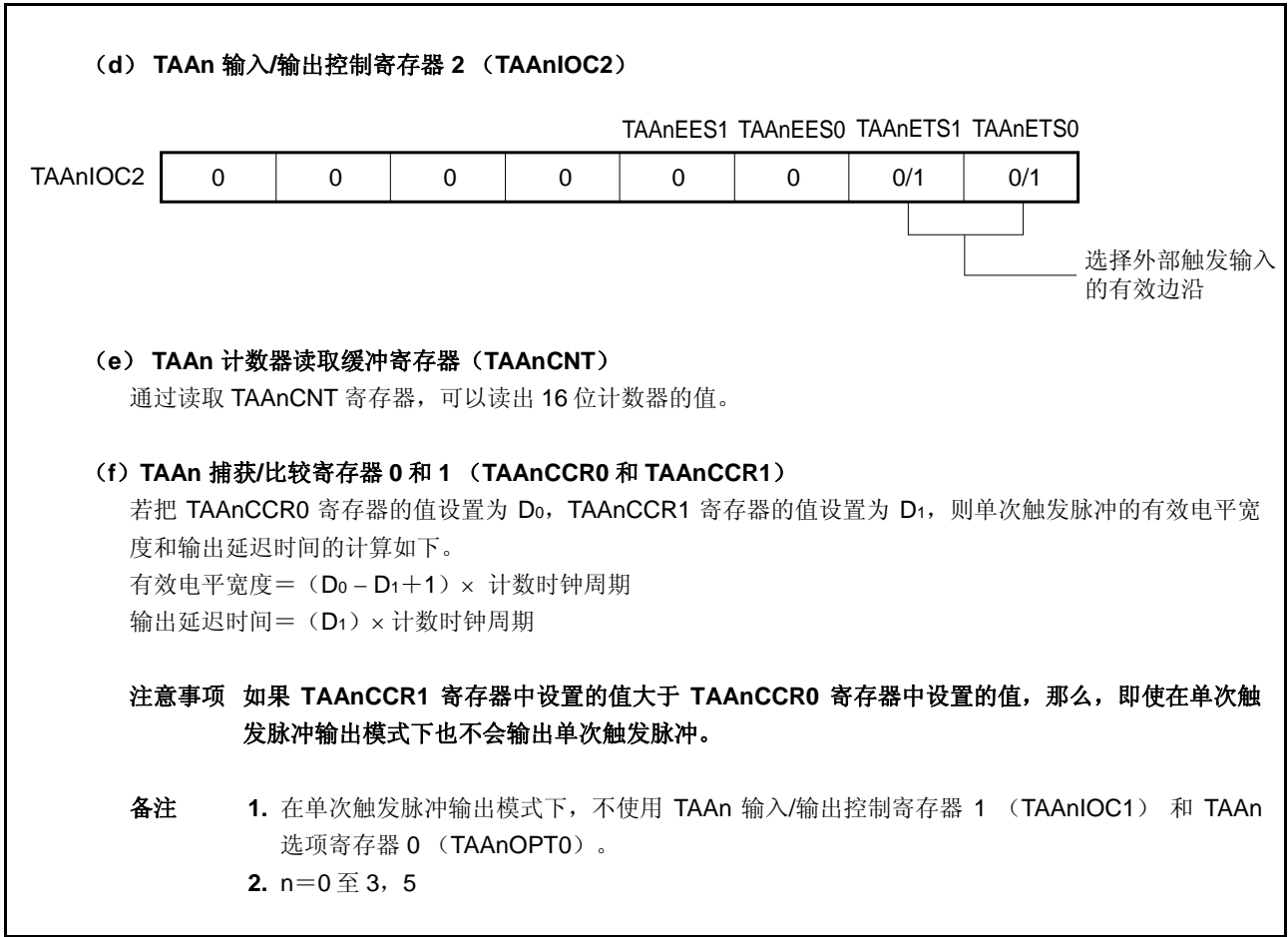
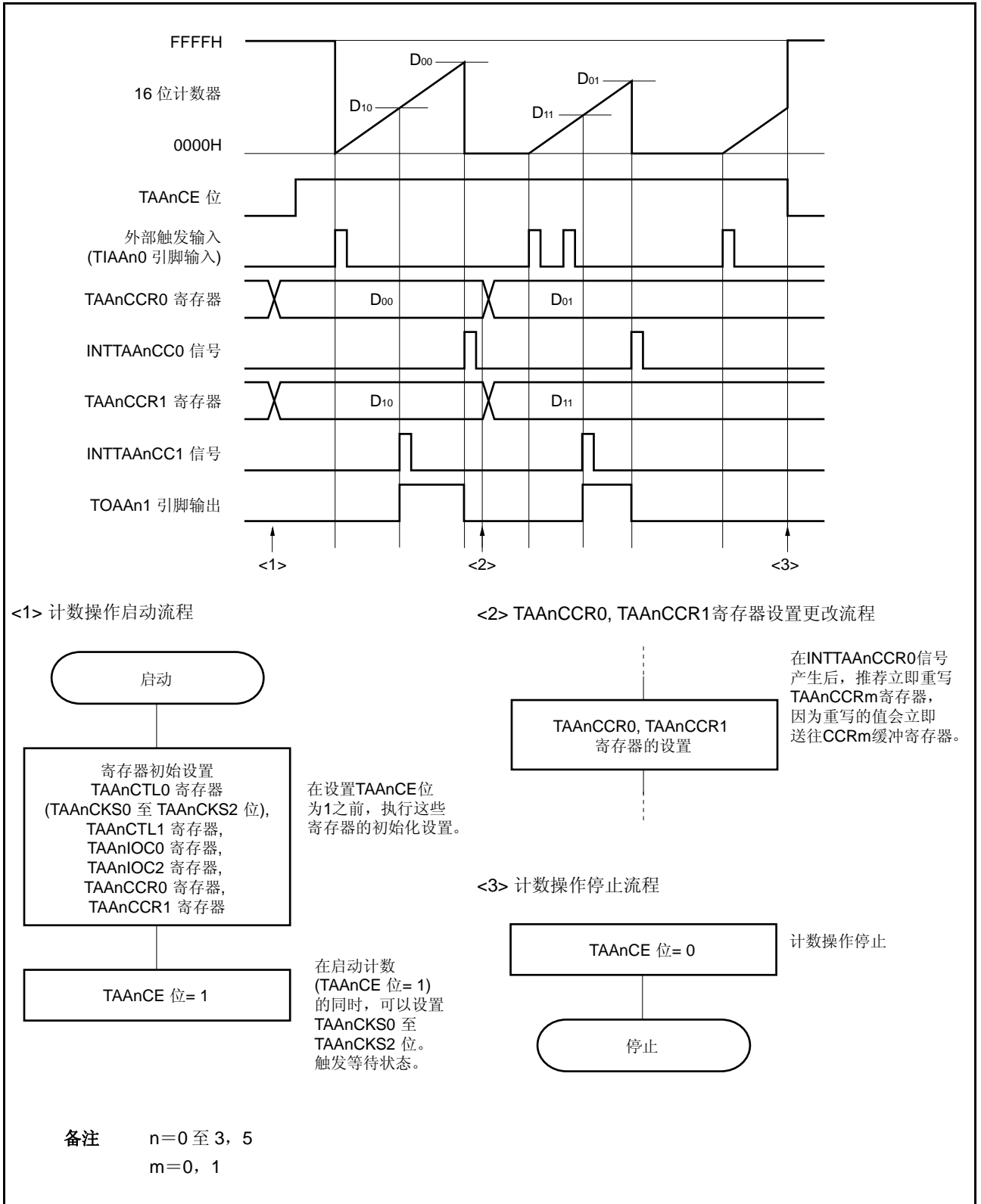


图 7-27. 单次触发脉冲输出模式时的寄存器设置 (2/2)



(1) 单次触发脉冲输出模式下的操作流程

图 7-28. 单次触发脉冲输出模式的软件处理流程

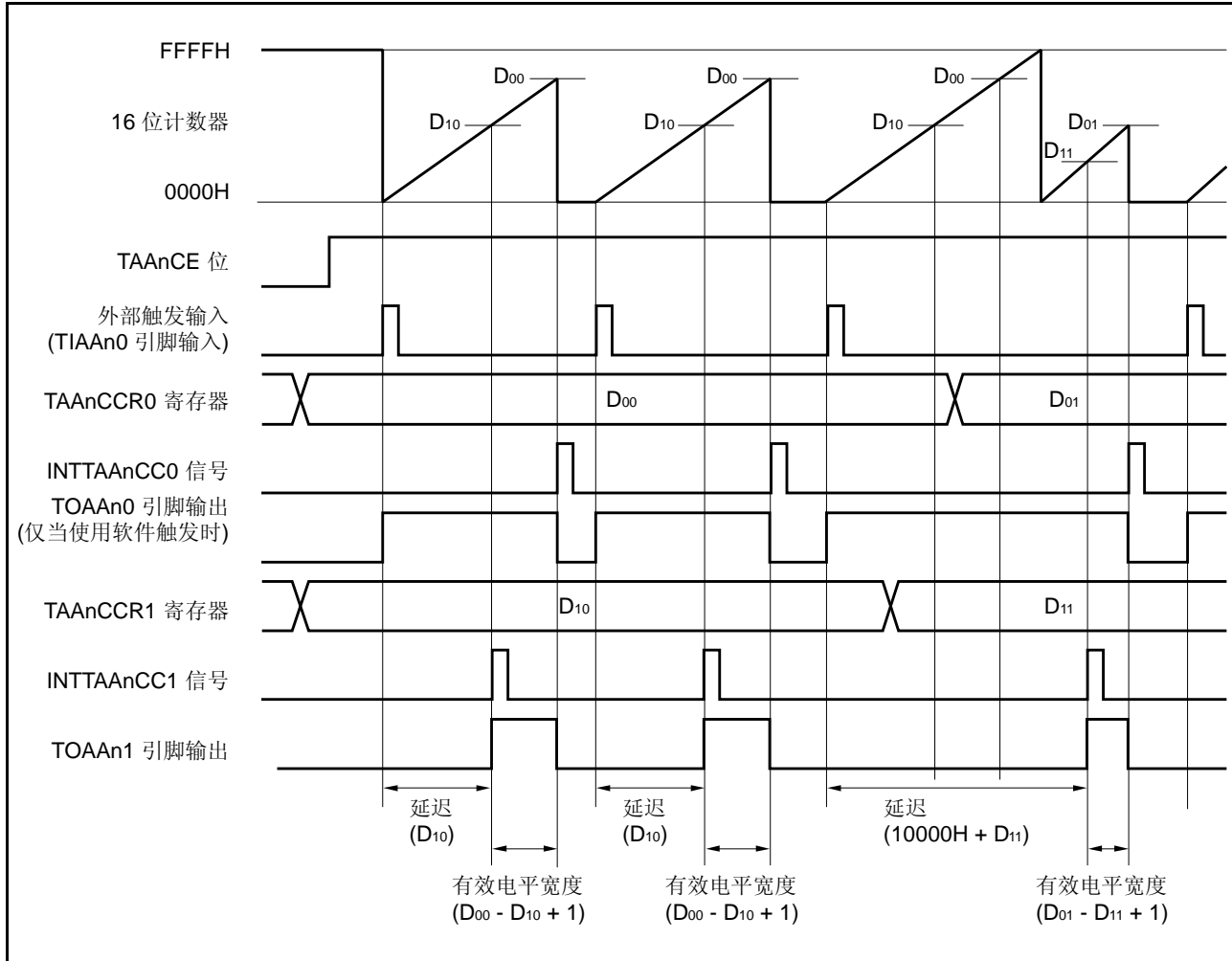


(2) 单次触发脉冲输出模式的操作时序

(a) 重写 TAAAnCCRm 寄存器的注意事项

要将 TAAAnCCRm 寄存器的值改为比当前值较小的值，应停止计数，然后改变其设定值。

如果在计数工作期间，将 TAAAnCCRm 寄存器的值重写为比当前值较小的值，16 位计数器可能会溢出。



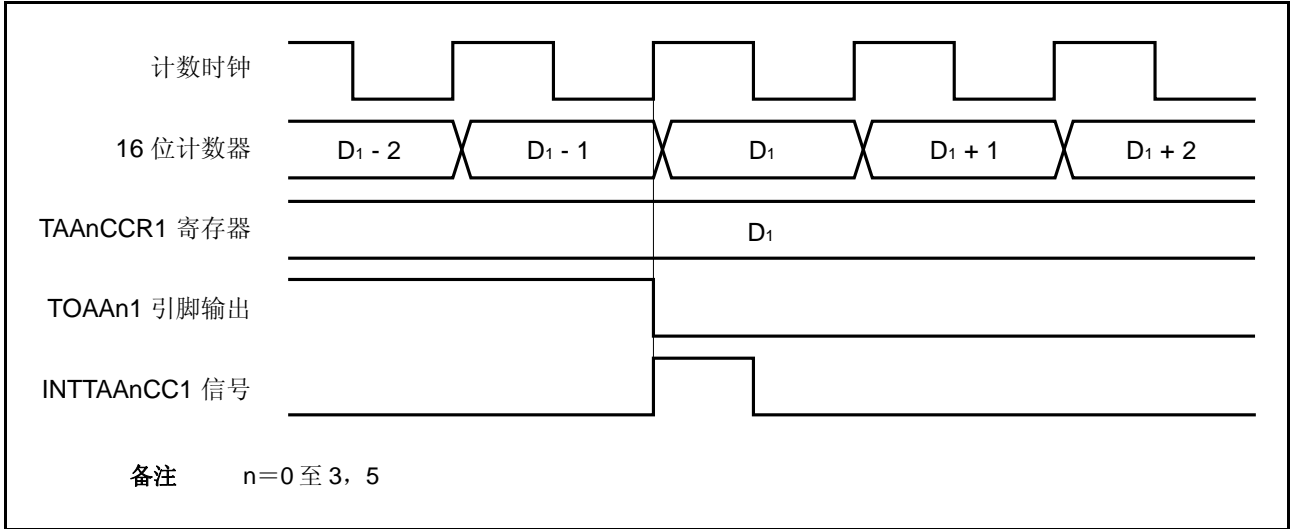
假设 $D_{00} > D_{01}$ 而且 $D_{10} > D_{11}$ ，TAAAnCCR0 寄存器由 D_{00} 重写为 D_{01} ，TAAAnCCR1 寄存器由 D_{10} 重写为 D_{11} ，如果在 16 位计数器的计数值大于 D_{11} 且小于 D_{10} 时重写 TAAAnCCR1 寄存器，并且在计数值大于 D_{01} 且小于 D_{00} 时重写 TAAAnCCR0 寄存器，则一旦重写寄存器，就会在各设置值上反映出来，并立即与计数值进行比较。计数器向上计数到 FFFFH 后，从 0000H 开始重新向上计数。当计数值与 D_{11} 匹配时，计数器会产生 INTTAAAnCC1 信号并使 TOAAAn1 引脚有效。当计数值与 D_{01} 匹配时，计数器产生 INTTAAAnCC0 信号，使 TOAAAn1 引脚失效，并停止计数。

因此，计数器输出脉冲的延迟时段或有效时段可能会与原先预计的单次触发脉冲有所不同。

备注 n=0 至 3, 5
m=0, 1

(b) 比较匹配中断请求信号 (INTTAAAnCC1) 的发生时序

在单次触发脉冲输出模式下，比较匹配中断请求信号 (INTTAAAnCC1) 的发生时序和其它情况下的 INTTAAAnCC1 信号发生时序不同，当 16 位计数器的计数值与 TAAAnCCR1 寄存器的值匹配时，产生 INTTAAAnCC1 中断请求信号。



通常在 16 位计数器的当前计数值与 TAAAnCCR1 寄存器的值匹配之后，在下一次计数时同步发生 INTTAAAnCC1 信号。

但是，在单次触发脉冲输出模式下，INTTAAAnCC1 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOAAn1 引脚输出信号变化的时序。

7.5.5 PWM输出模式 (TAA_nMD2 至TAA_nMD0 位=100)

在 PWM 输出模式下，当 TAA_nCTL0.TAA_nCE 位被置为 1 时，PWM 波形从 TOAA_n1 引脚输出。
而且，由 TOAA_n0 引脚输出的脉冲，其半周期等于 PWM 的全周期。

图 7-29. PWM 输出模式的配置

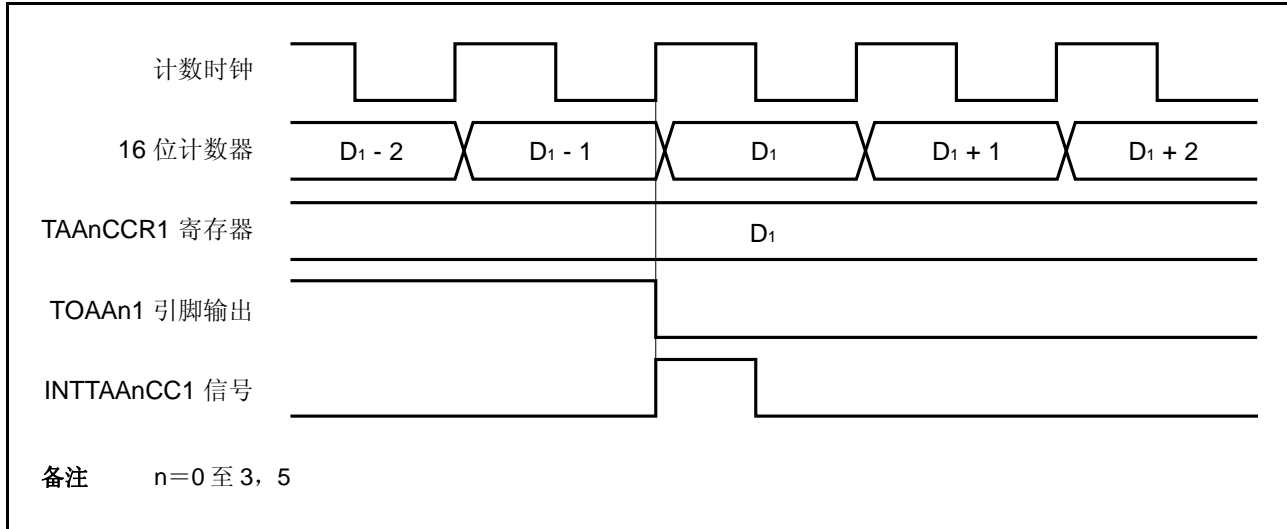
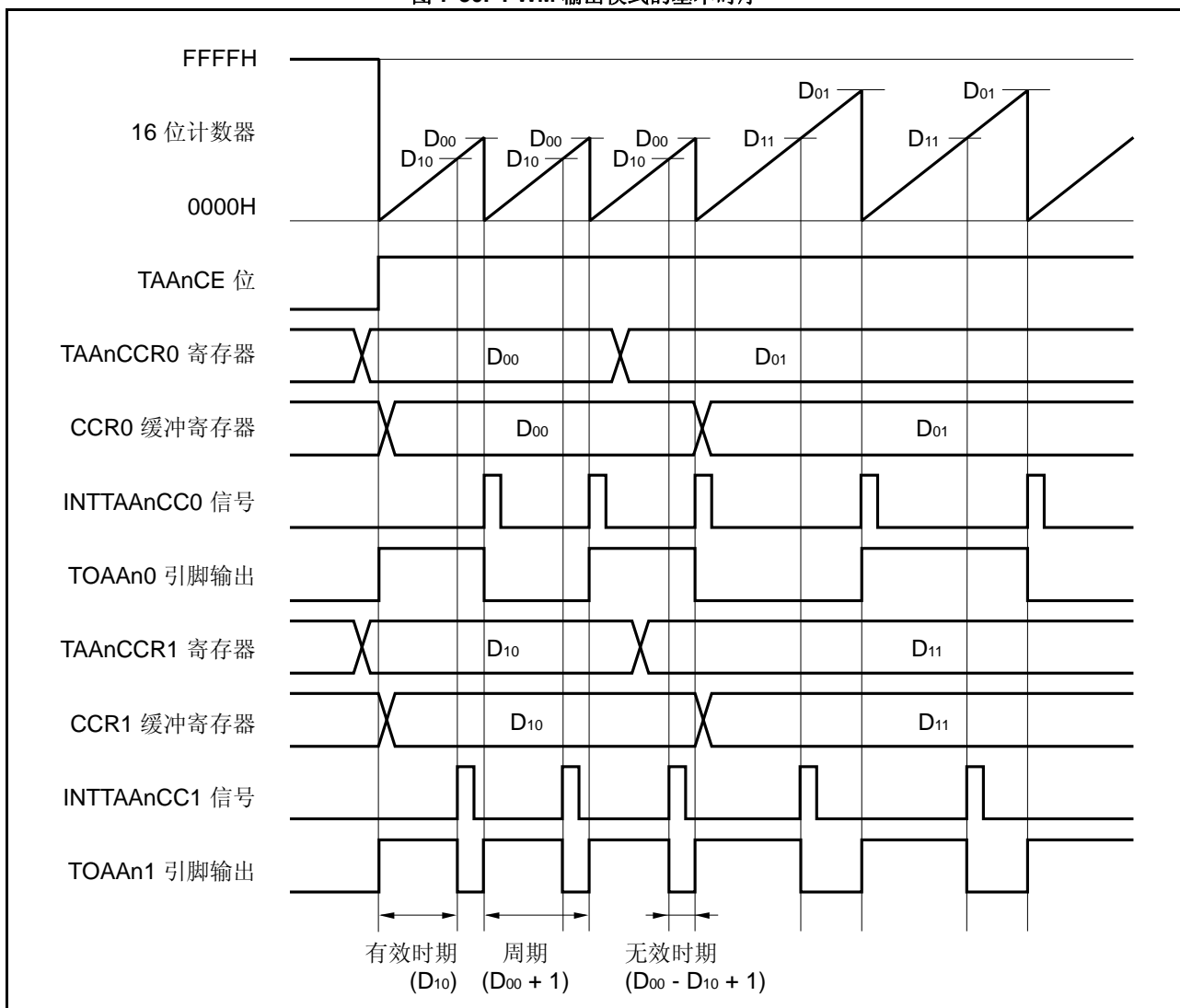


图 7-30. PWM 输出模式的基本时序



当 TAAAnCE 位置为 1 时，16 位计数器从 FFFFH 清除为 0000H，计数开始，同时从 TOAAAn1 引脚输出 PWM 波形。PWM 波形的有效电平宽度、周期、占空比系数的计算如下。

有效电平宽度 = (TAAAnCCR1 寄存器的设置值) × 计数时钟周期

周期 = (TAAAnCCR0 寄存器的设置值 + 1) × 计数时钟周期

占空比系数 = (TAAAnCCR1 寄存器的设置值) / (TAAAnCCR0 寄存器的设置值 + 1)

在计数操作时，可通过重写 TAAAnCCRm 寄存器来改变 PWM 波形。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配时，16 位计数器清除为 0000H 时，新写入的值会发生作用。

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配之后，进行下一次计数时，会同步产生比较匹配中断请求信号 INTTAAAnCC0，且 16 位计数器清除为 0000H。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTAAAnCC1。

当 16 位计数器的计数值与 CCRm 缓冲寄存器的值匹配，而且 16 位计数器清除为 0000H 时，TAAAnCCRm 寄存器的设定值就被传送到 CCRm 缓冲寄存器。

备注 n=0 至 3, 5, m=0, 1

图 7-31. PWM 输出模式时的寄存器设置 (1/2)

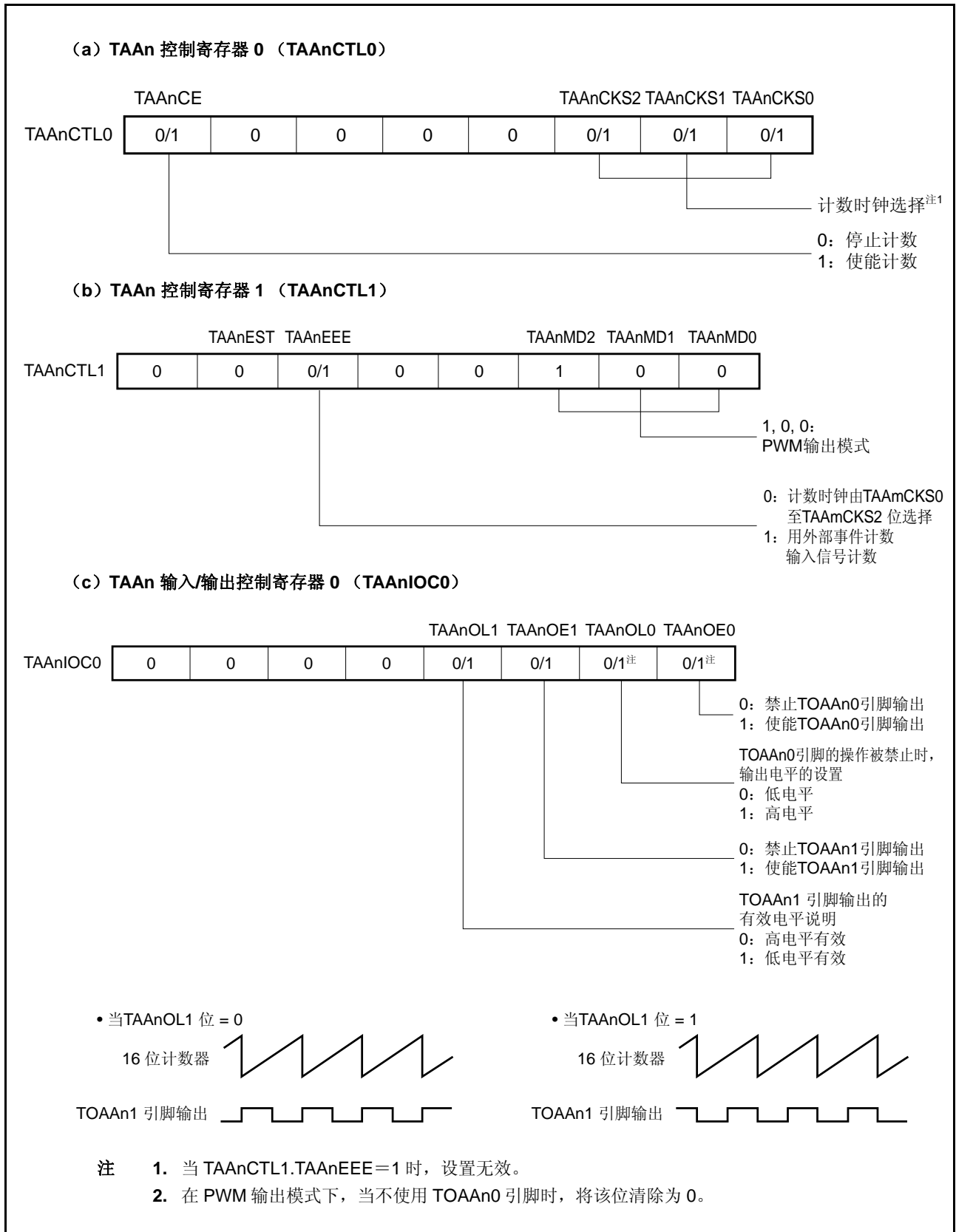
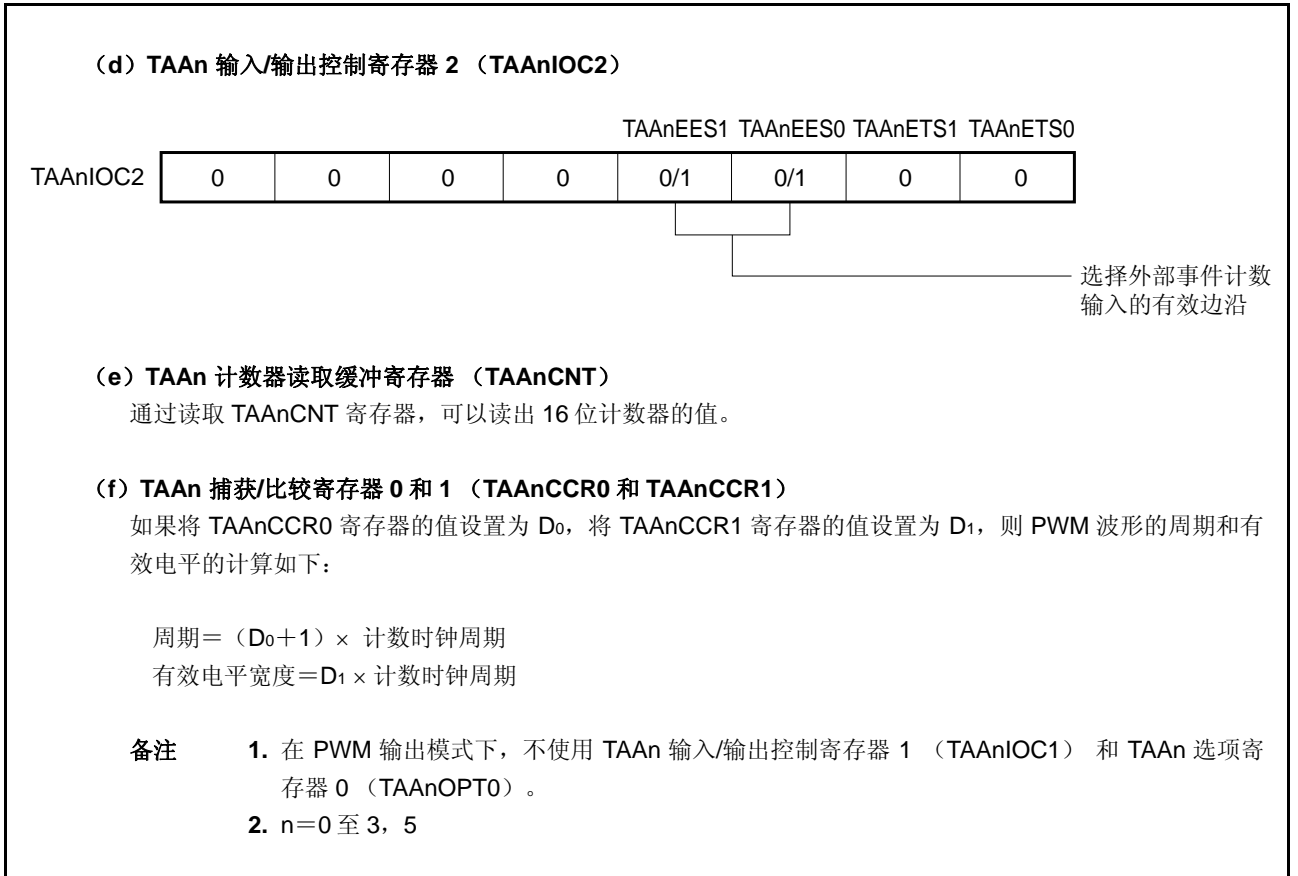


图 7-31. PWM 输出模式时的寄存器设置 (2/2)



(1) PWM 输出模式下的操作流程

图 7-32. PWM 输出模式的软件处理流程 (1/2)

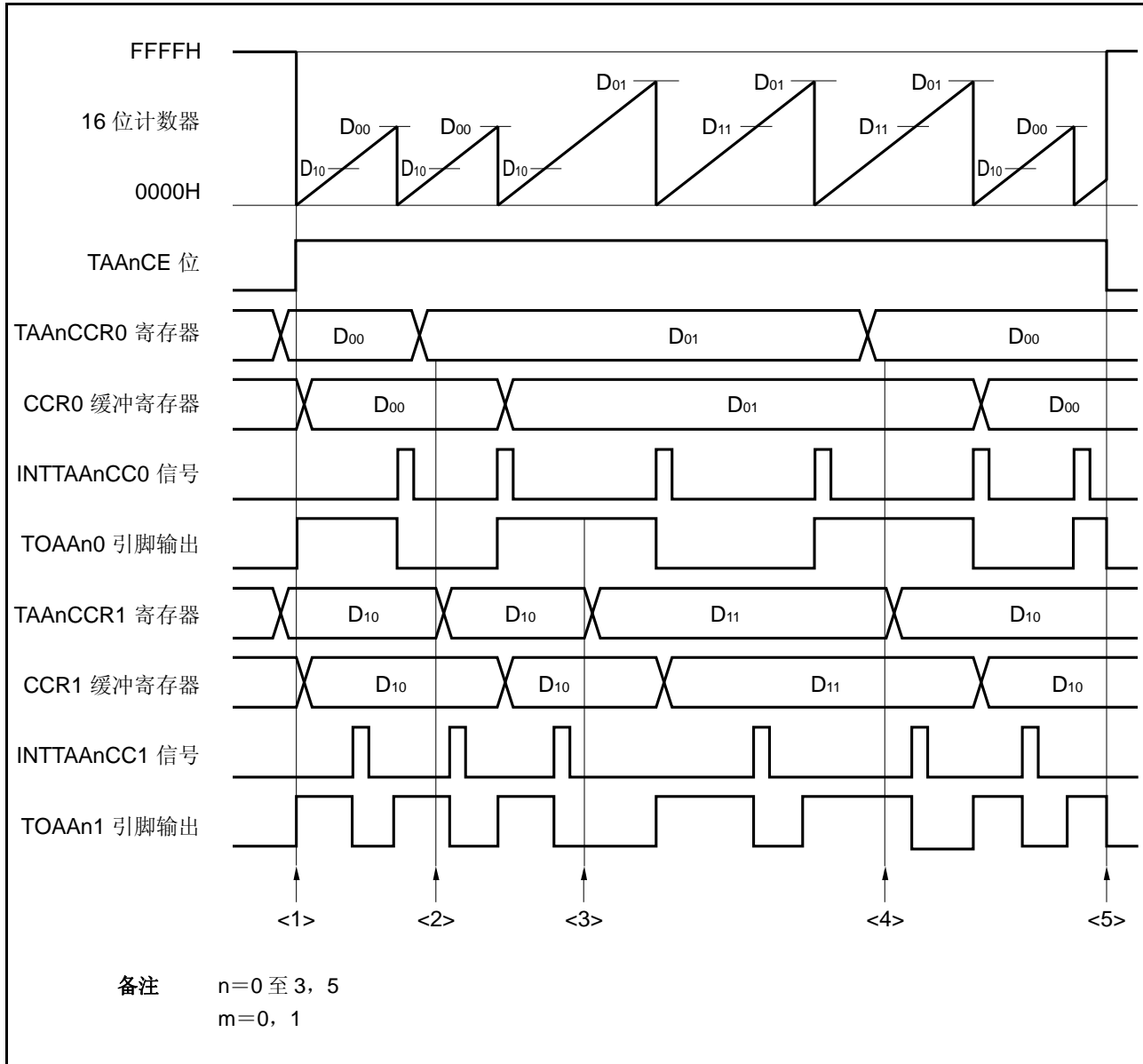
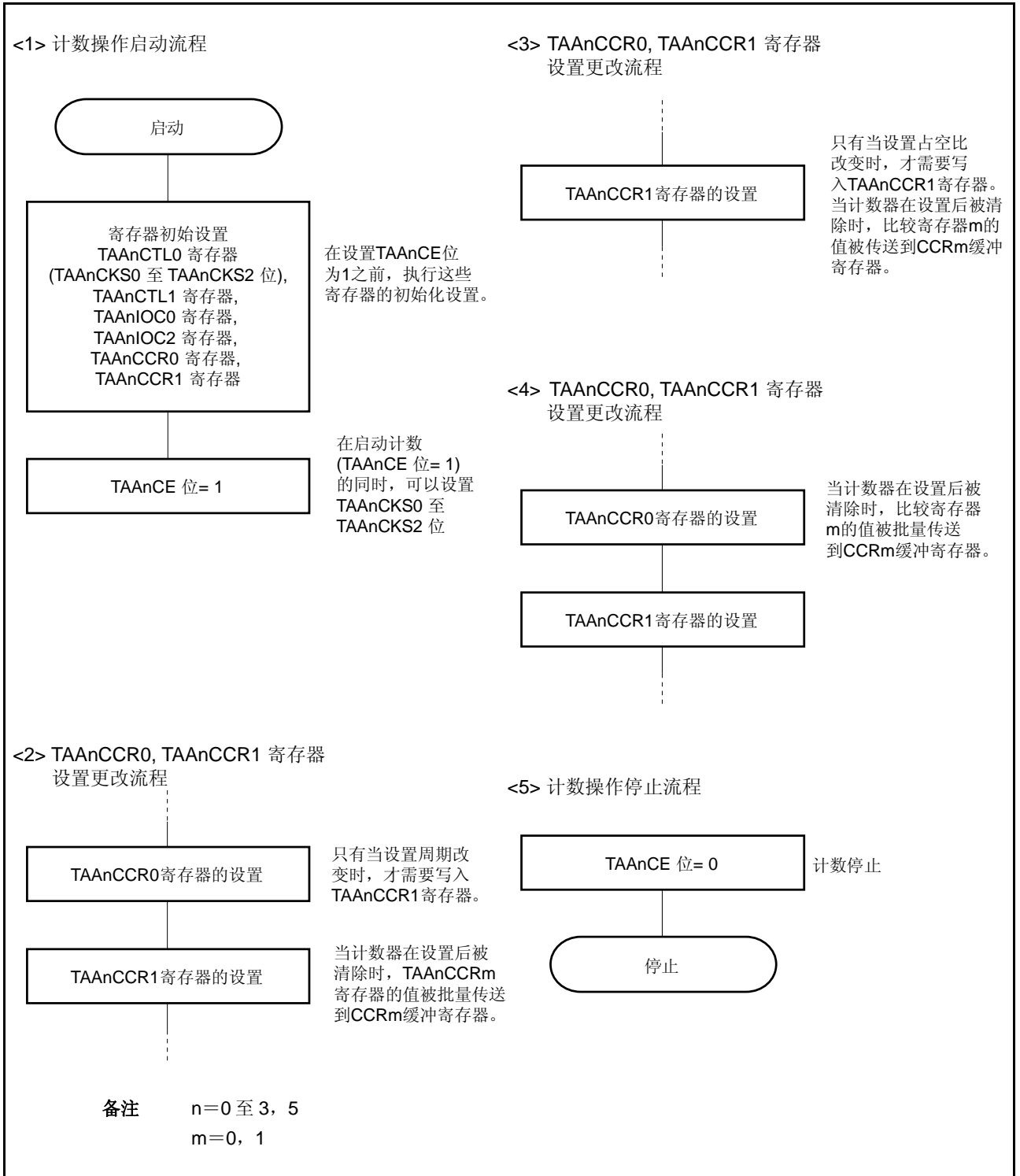


图 7-32. PWM 输出模式时的软件处理流程 (2/2)

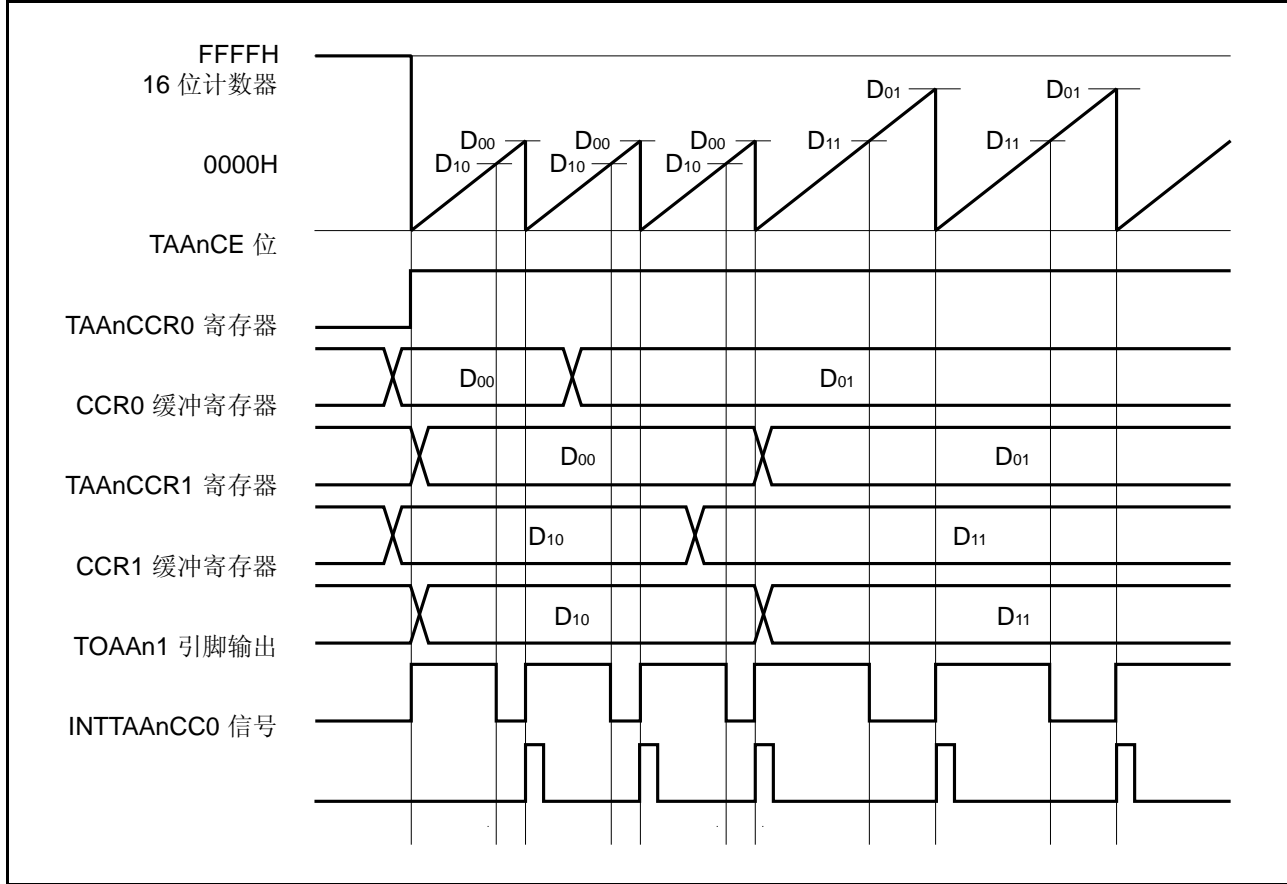


(2) PWM 输出模式的操作时序

(a) 操作过程中改变脉冲宽度

计数器操作过程中，若要改变 PWM 波形，应重新写入 TAAAnCCR1 寄存器。

当检测到 INTTAAAnCC1 信号以后，在写入 TAAAnCCR1 寄存器之后重写 TAAAnCCRm 寄存器。



要把数据从 TAAAnCCRm 寄存器传送到 CCRm 缓冲寄存器，必须对 TAAAnCCR1 寄存器进行写入操作。

此时如果 PWM 波形的周期和有效电平都要改变，应先将周期设定到 TAAAnCCR0 寄存器，然后再将有效电平设定到 TAAAnCCR1 寄存器。

如果只改变 PWM 波形的周期，应首先把周期设定给 TAAAnCCR0 寄存器，然后将相同值（原值）写入 TAAAnCCR1 寄存器。

如果只改变 PWM 波形的有效电平宽度（占空比），仅需要设置 TAAAnCCR1 寄存器。

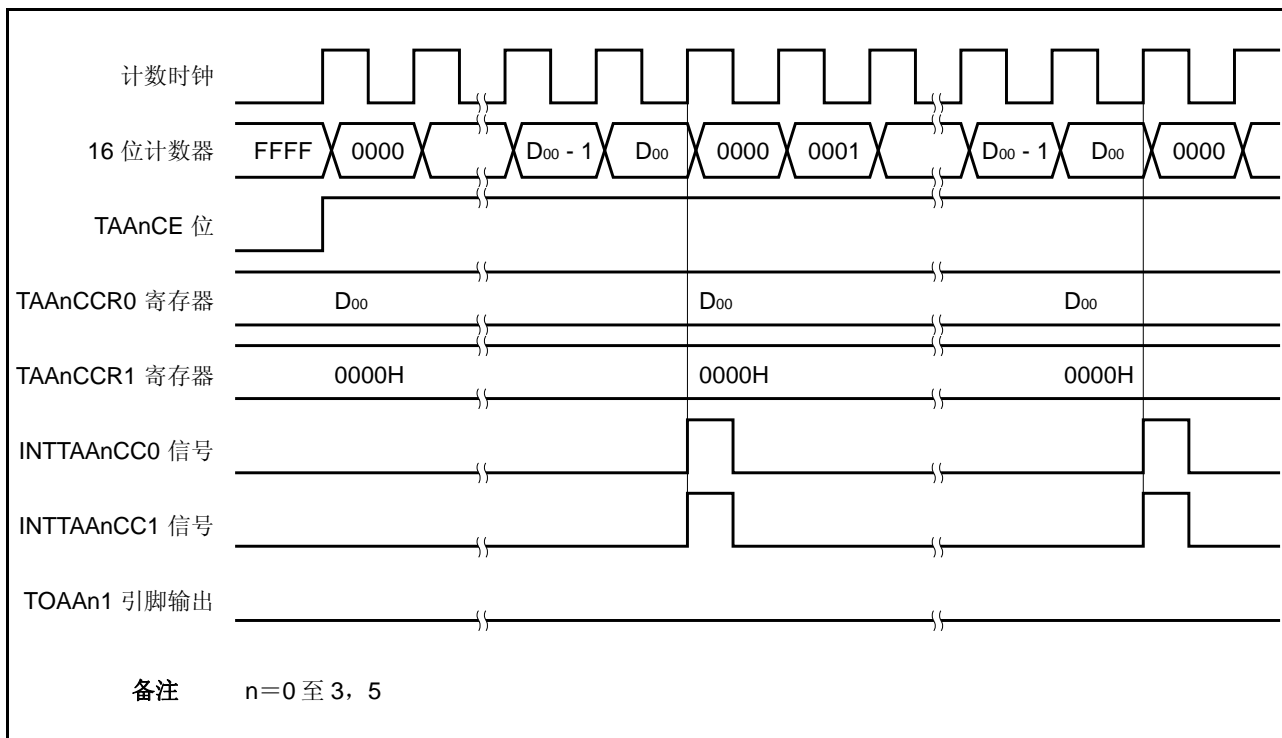
当数据写入 TAAAnCCR1 寄存器之后，在对 16 位计数器清零的同时，写入 TAAAnCCRm 寄存器的值被传送到 CCRm 缓冲寄存器，并作为与 16 位计数器进行比较的值。

在写入 TAAAnCCR1 寄存器一次之后，要再次写入 TAAAnCCR0 或 TAAAnCCR1 寄存器，应该在 INTTAAAnCC0 信号产生之后进行。否则，CCRm 缓冲寄存器的值可能不确定，因为把数据从 TAAAnCCRm 寄存器传送到 CCRm 缓冲寄存器的时序会和写入 TAAAnCCRm 寄存器时序发生冲突。

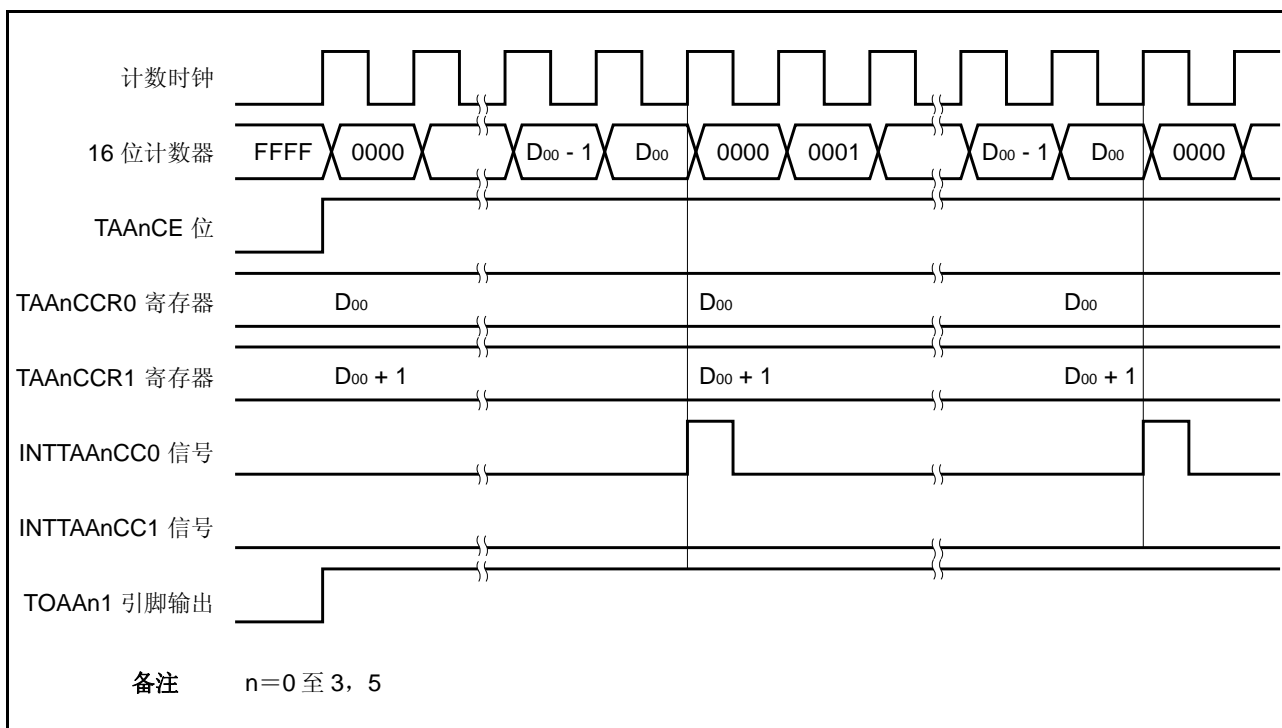
备注 n=0 至 3, 5, m=0, 1

(b) PWM 波形的 0%和 100% 输出

为了输出 0%的波形，需要将 TAAAnCCR1 寄存器设置为 0000H。如果 TAAAnCCR0 寄存器的设定值为 FFFFH，则会在每个周期产生 INTTAAAnCC1 中断请求信号。

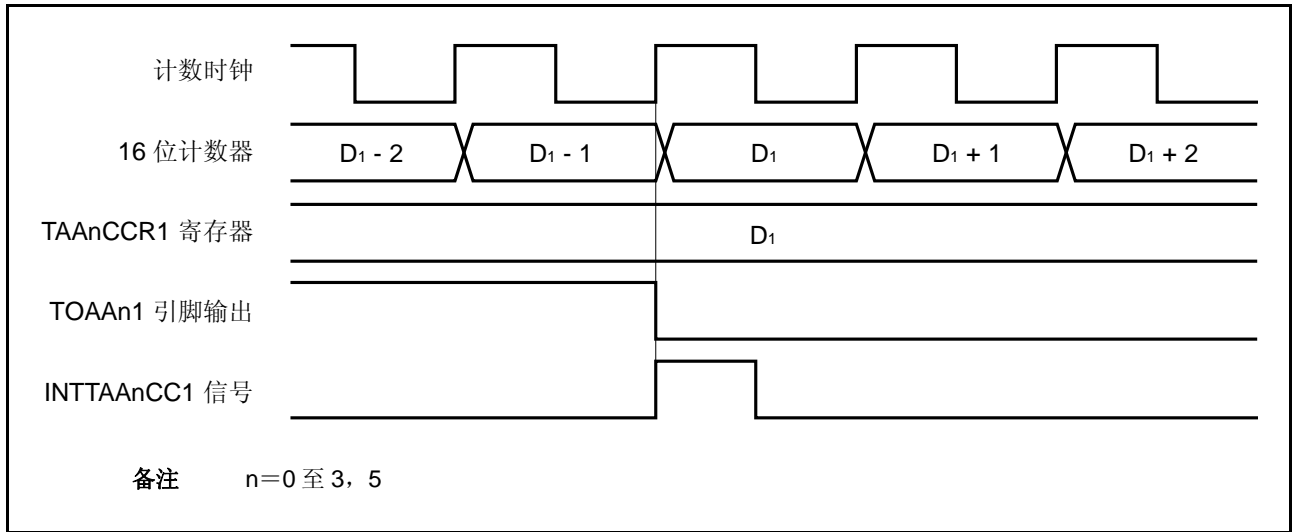


为了输出 100%的波形，需要将 TAAAnCCR1 寄存器的值设为： $(\text{TAAAnCCR0 寄存器的设定值} + 1)$ 。如果 TAAAnCCR0 寄存器的设定值为 FFFFH，则不会产生 100%波形输出。



(c) 比较匹配中断请求信号 (INTTAAAnCC1) 的发生时序

在 PWM 输出模式下，比较匹配中断请求信号 (INTTAAAnCC1) 的发生时序与其它情况下 INTTAAAnCC1 信号的发生时序有所不同；当 16 位计数器的计数值与 TAAAnCCR1 寄存器的值匹配时，产生 INTTAAAnCC1 信号。



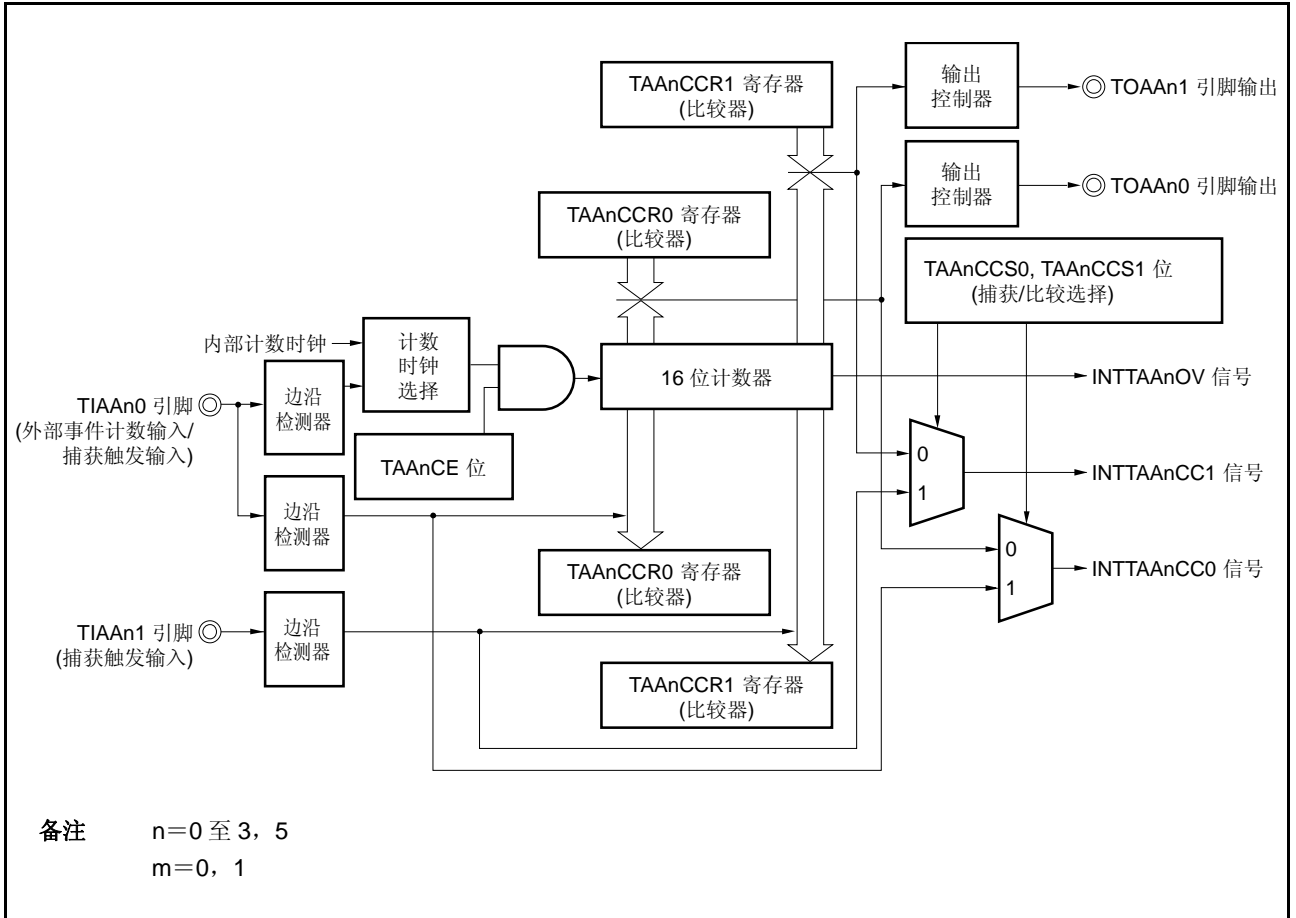
通常在 16 位计数器的计数值与 TAAAnCCR1 寄存器的值匹配之后，在下次计数时同步发生 INTTAAAnCC1 信号。

但是，在 PWM 输出模式下，INTTAAAnCC1 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOAAAn1 引脚输出信号变化的时序。

7.5.6 自由运行定时器模式 (TAA_nMD2 至 TAA_nMD0 位 = 101)

在自由运行定时器模式下，当 TAA_nCTL0.TAA_nCE 位置为 1 时，16 位定时器/事件计数器 AA 开始计数。此时，根据 TAA_nOPT0.TAA_nCCS0 和 TAA_nOPT0.TAA_nCCS1 位的设置，TAA_nCCR_m 寄存器可以用作比较寄存器或捕获寄存器。

图 7-33. 自由运行定时器模式的配置

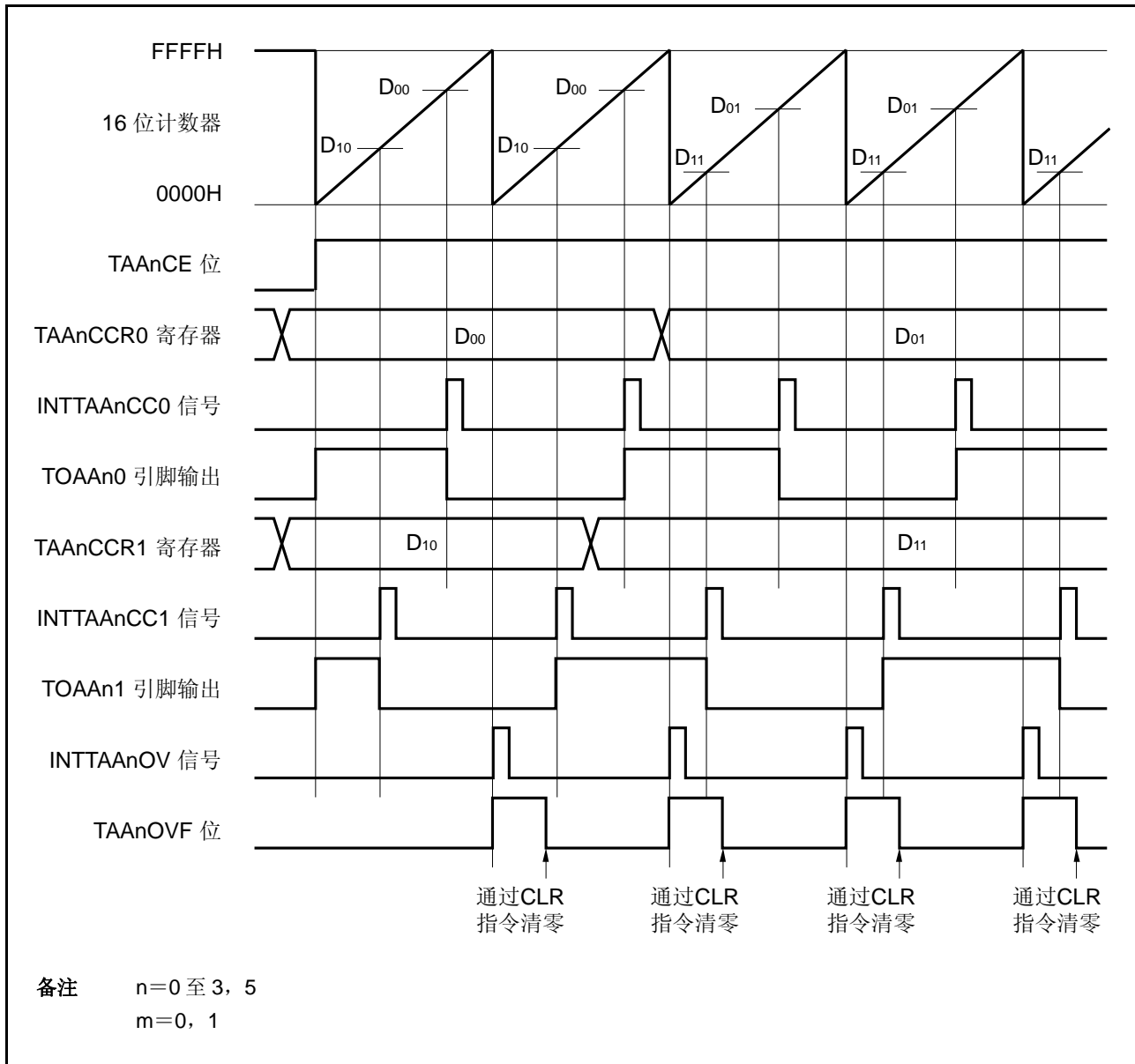


当 TAA_nCE 位被置为 1 时，16 位定时器/事件计数器 AA 开始计数，TOAAn0 和 TOAAn1 引脚的输出信号反相。此后当 16 位计数器的计数值与 TAA_nCCR_m 寄存器的设定值匹配时，产生一个比较匹配中断请求信号 (INTTAA_nCC_m)，而且 TOAAn_m 引脚的输出信号反相。

16 位计数器继续对计数时钟同步计数。当它计数到 FFFFH 时，会在下一个计数时钟时产生一个溢出中断请求信号 (INTTAA_nOV)，同时计数器清除为 0000H，然后继续进行计数。此时，溢出标志 (TAA_nOPT0.TAA_nOVF 位) 被置为 1，通过软件运行 CLR 指令可以把溢出标志位清 0。

当计数器工作时，可以重写 TAA_nCCR_m 寄存器。如果进行重写，则新写入的值会立刻反映出来，并与计数值进行比较。

图 7-34. 自由运行定时器模式下的基本时序 (比较功能)



当 TAA_nCE 位被置为 1 时，16 位计数器开始计数。当检测到 TIAAn_m 引脚输入信号的有效边沿时，16 位计数器的计数值存储在 TAA_nCCR_m 寄存器中，并产生一个捕获中断请求信号 (INTTAA_nCC_m)。

16 位计数器继续对计数时钟同步计数。当它计数到 FFFFH 时，会在下一个计数时钟时产生一个溢出中断请求信号 (INTTAA_nOV)，同时计数器清除为 0000H，然后继续进行计数。此时，溢出标志 (TAA_nOPT0.TAA_nOVF 位) 被置为 1，通过软件运行 CLR 指令可以把溢出标志位清 0。

图 7-35. 自由运行定时器模式下的基本时序 (捕获功能)

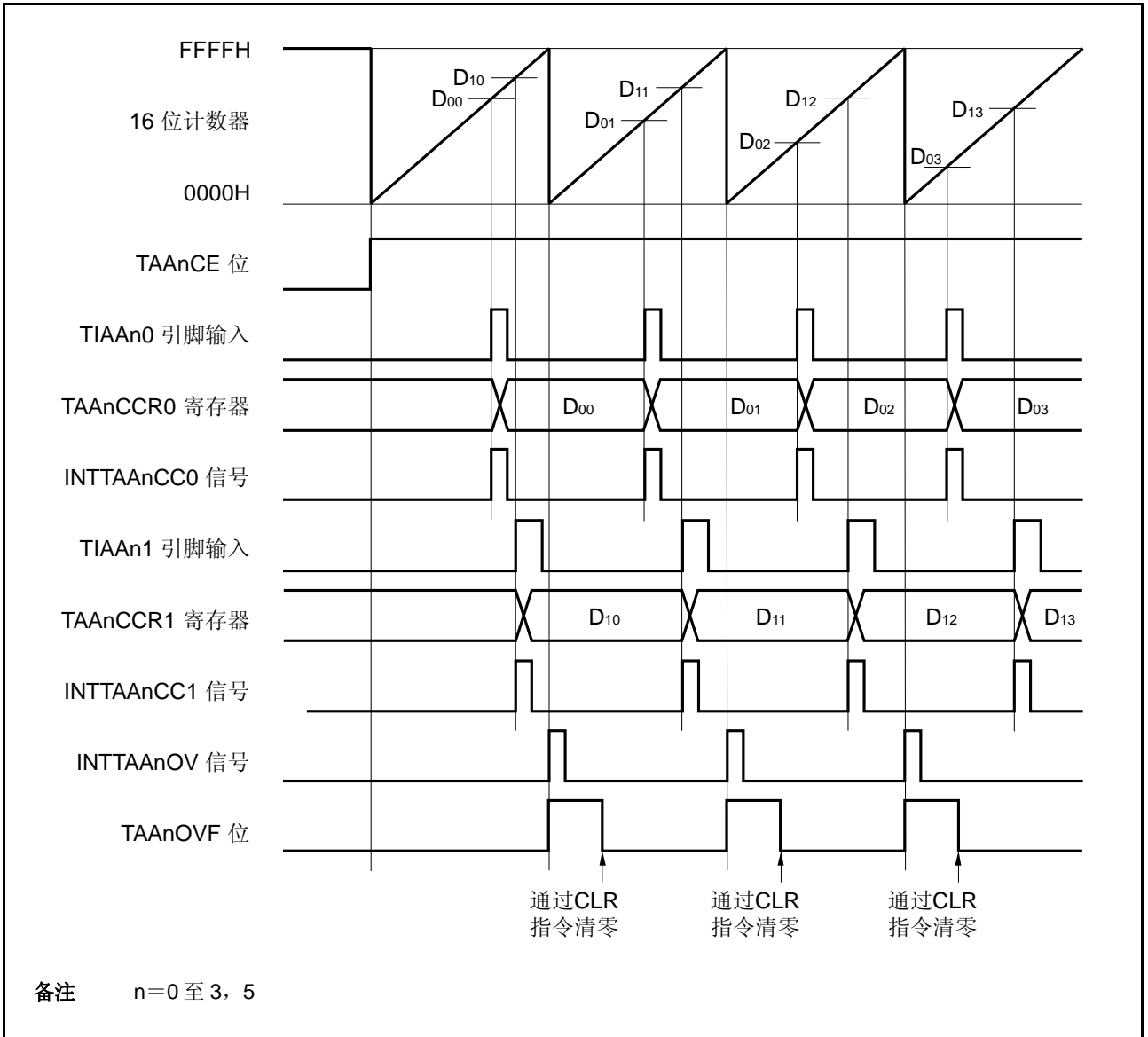


图 7-36. 自由运行定时器模式下的寄存器设置 (1/2)

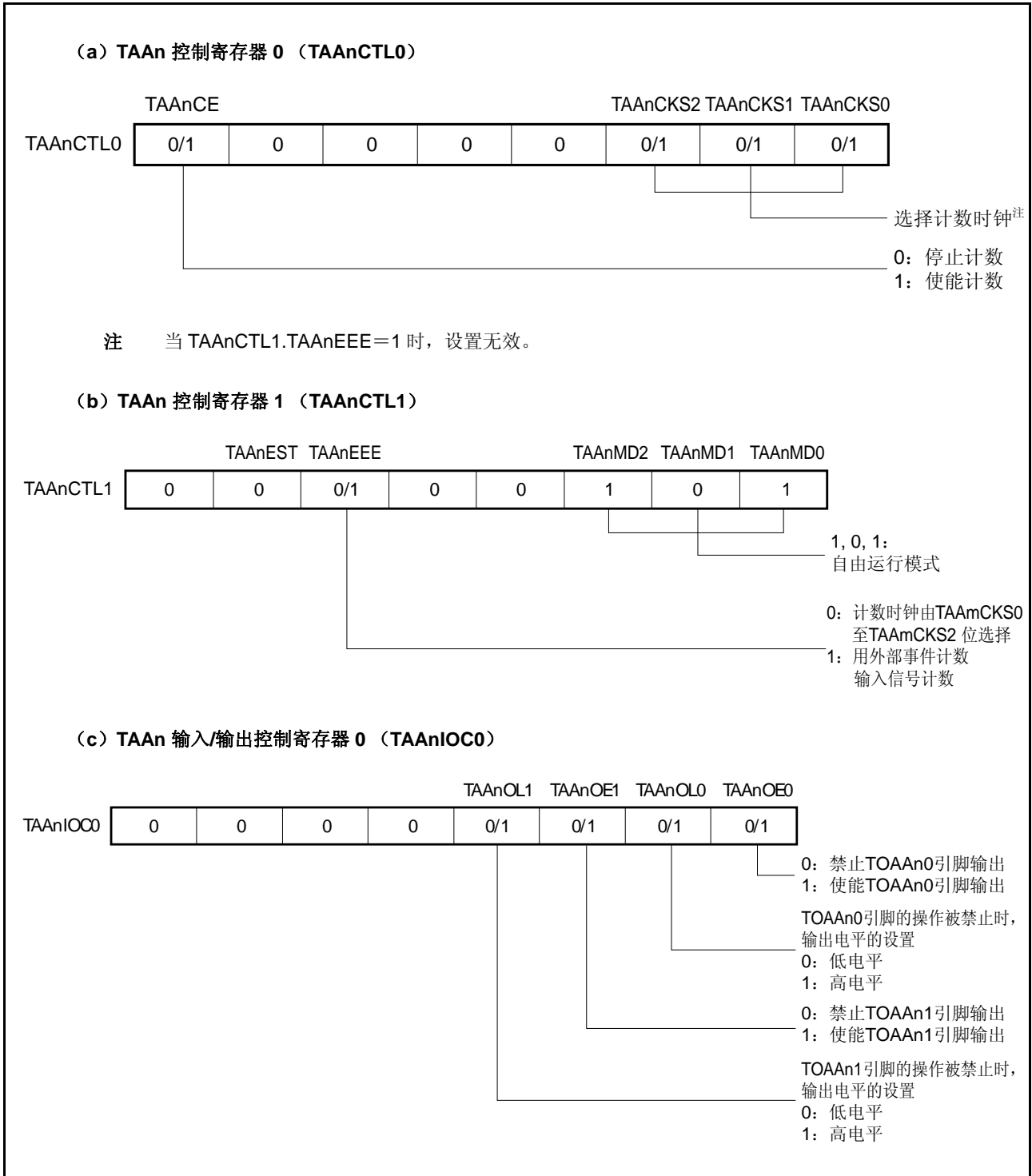
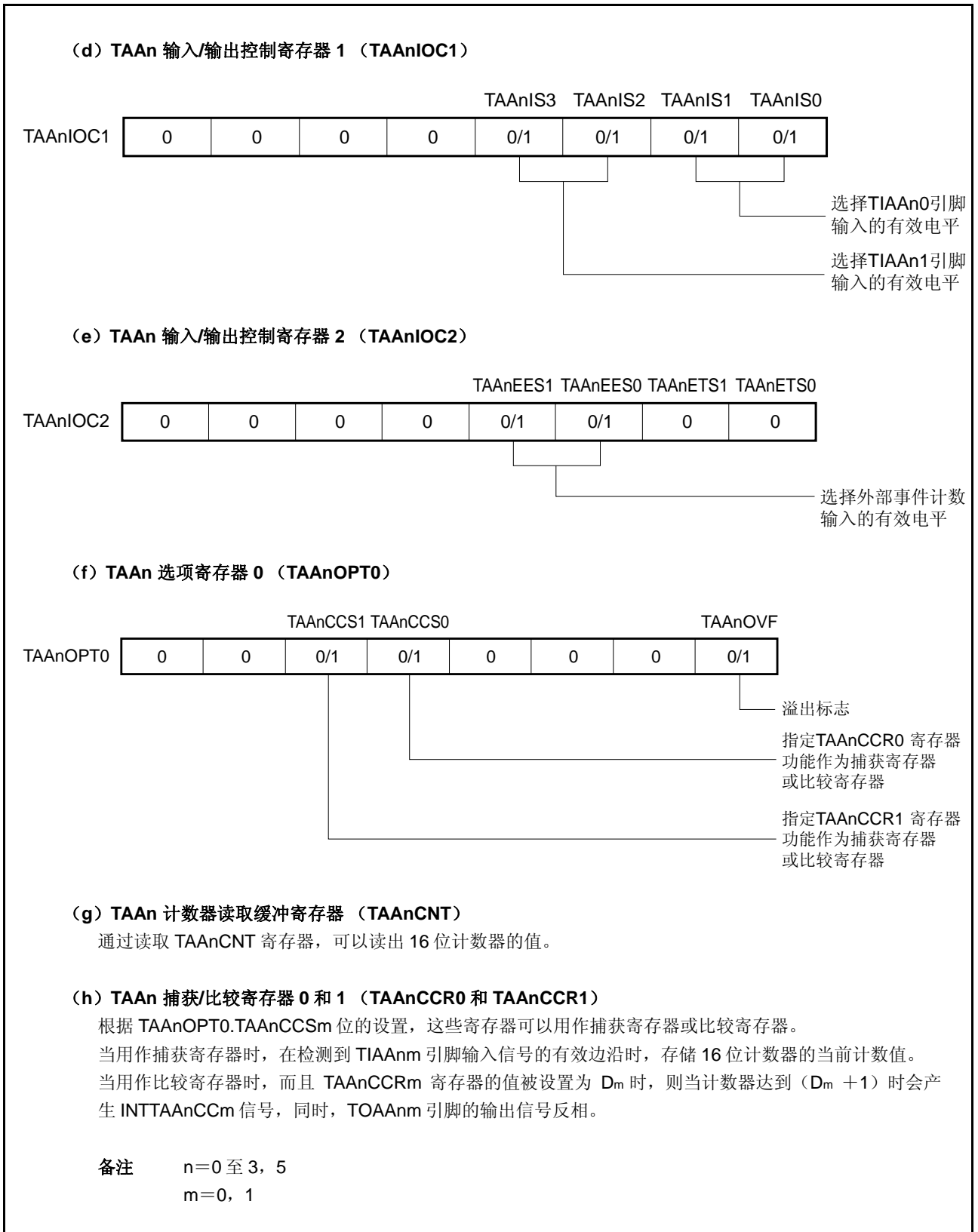


图 7-36. 自由运行定时器模式下的寄存器设置 (2/2)



(1) 自由运行定时器模式的操作流程

(a) 当捕获/比较寄存器用作比较寄存器时

图 7-37. 自由运行定时器模式的软件处理流程（比较功能）（1/2）

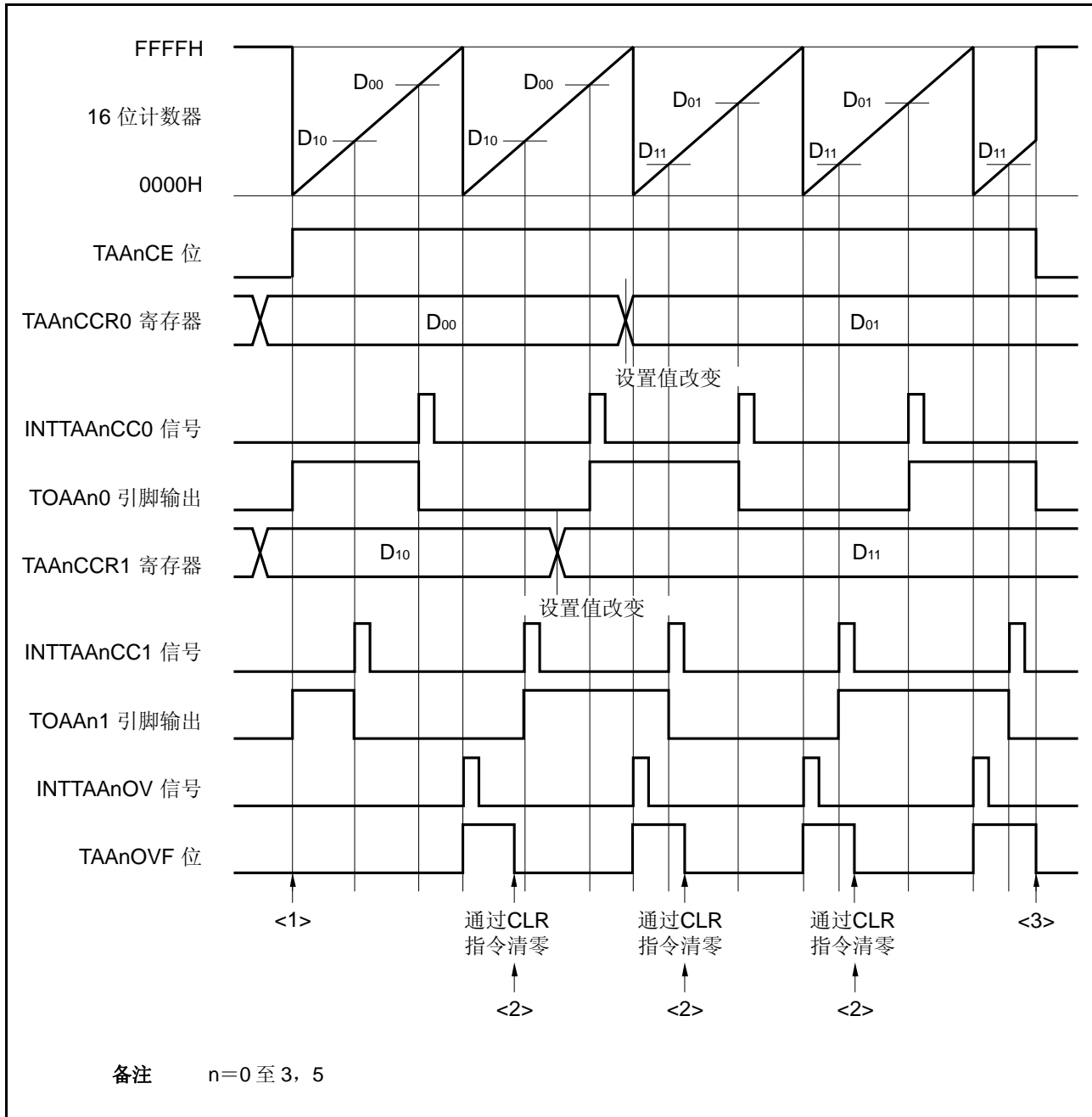
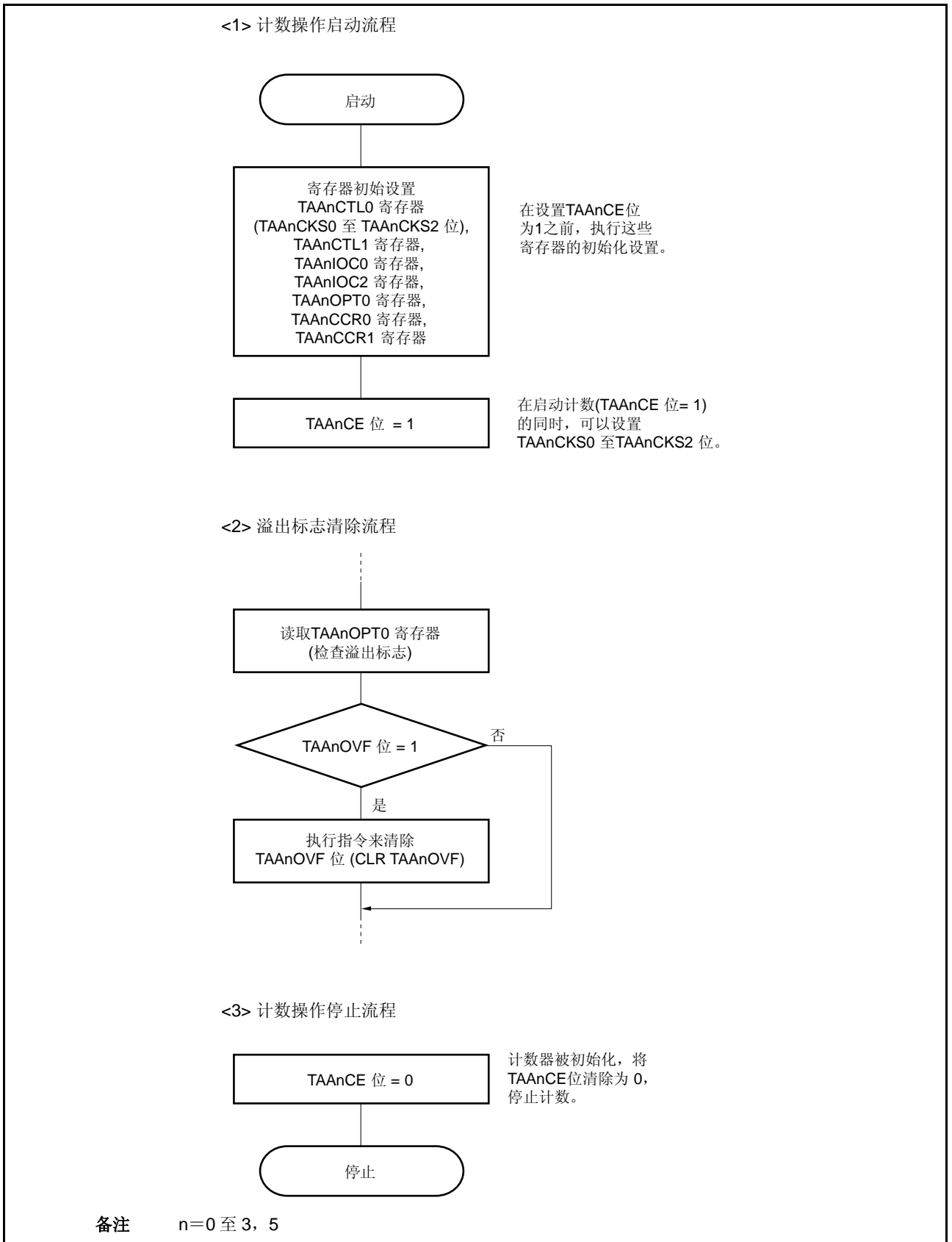


图 7-37. 自由运行定时器模式的软件处理流程（比较功能）（2/2）



(b) 当捕获/比较寄存器用作捕获寄存器时

图 7-38. 自由运行定时器模式的软件处理流程 (捕获功能) (1/2)

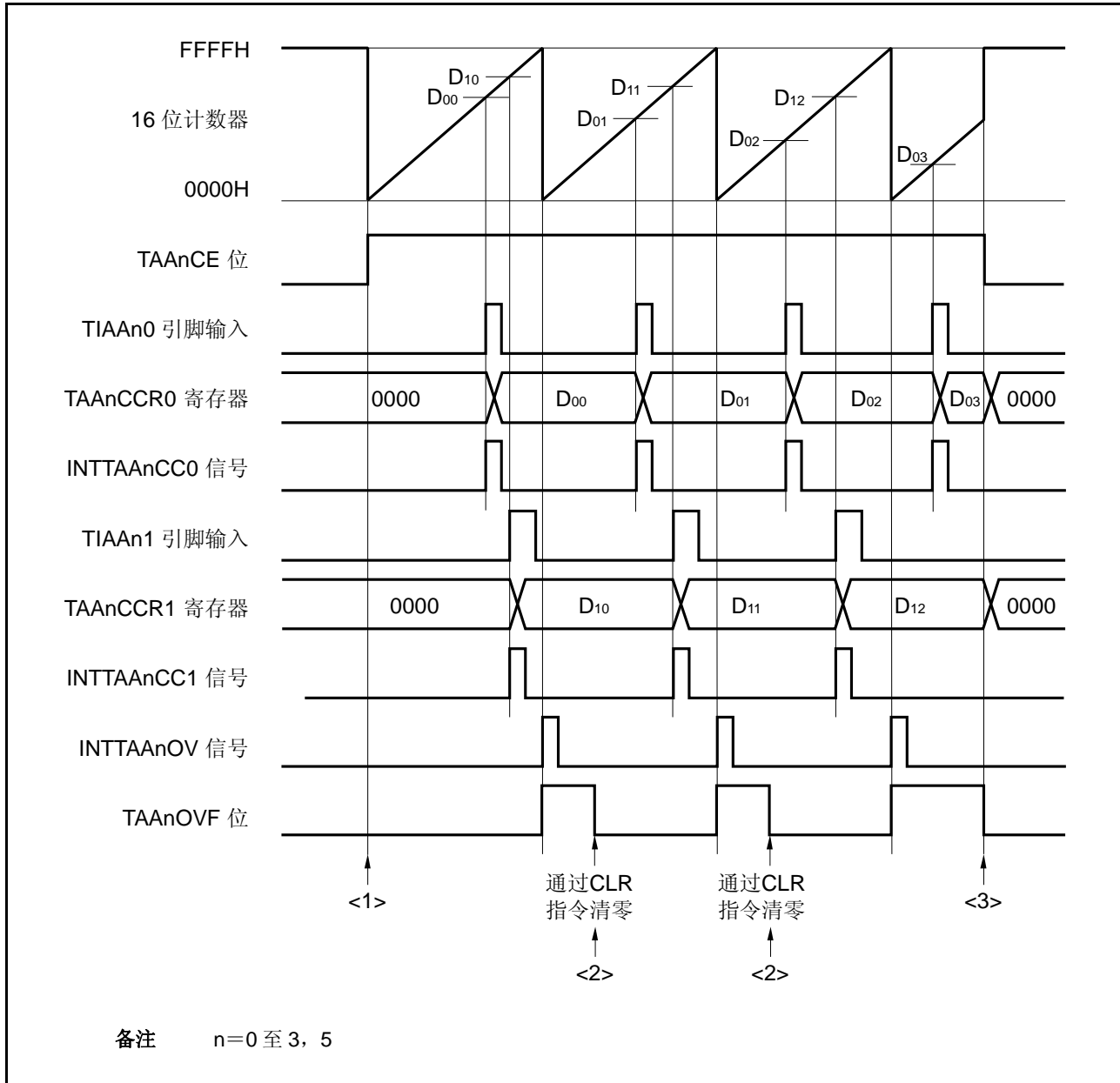
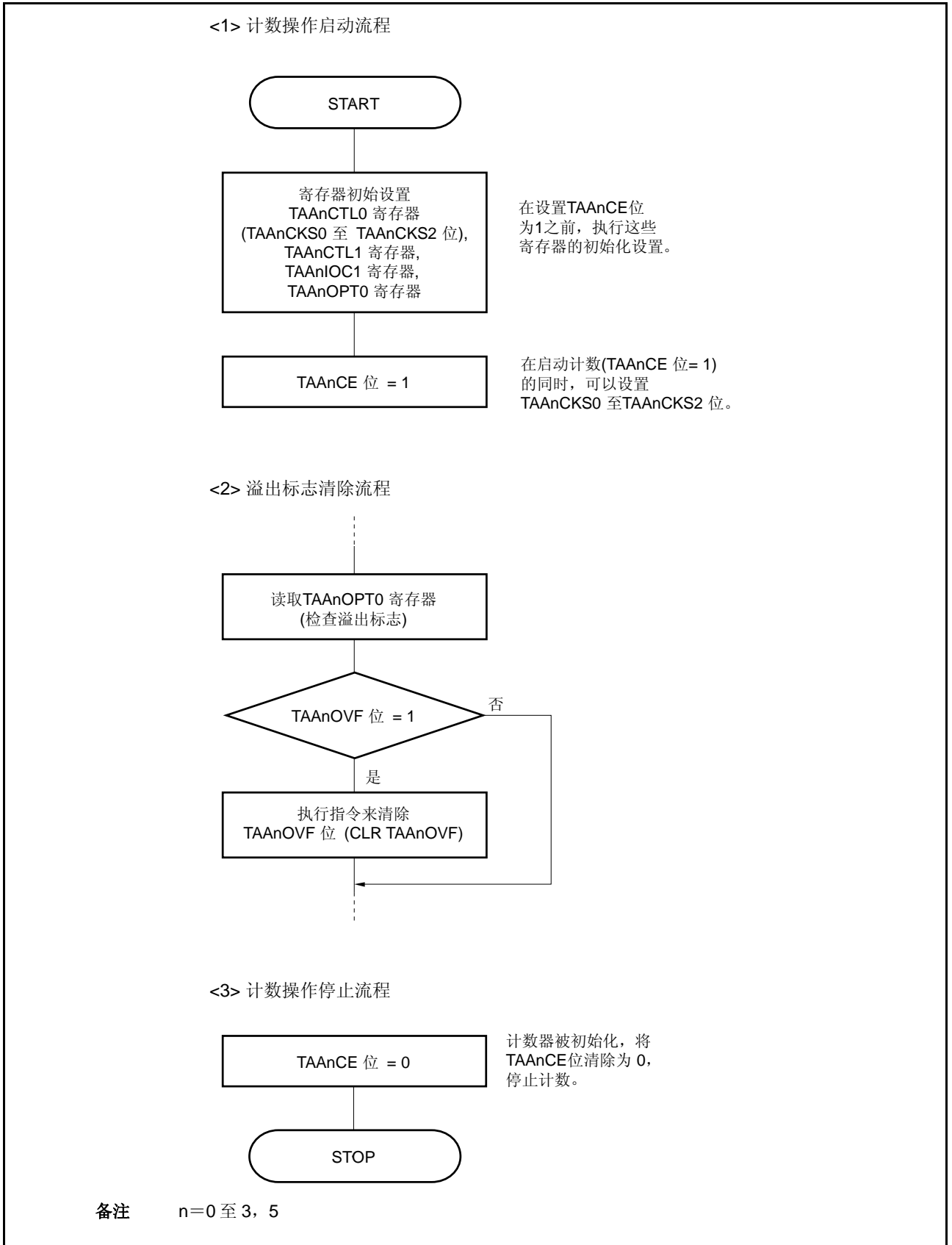


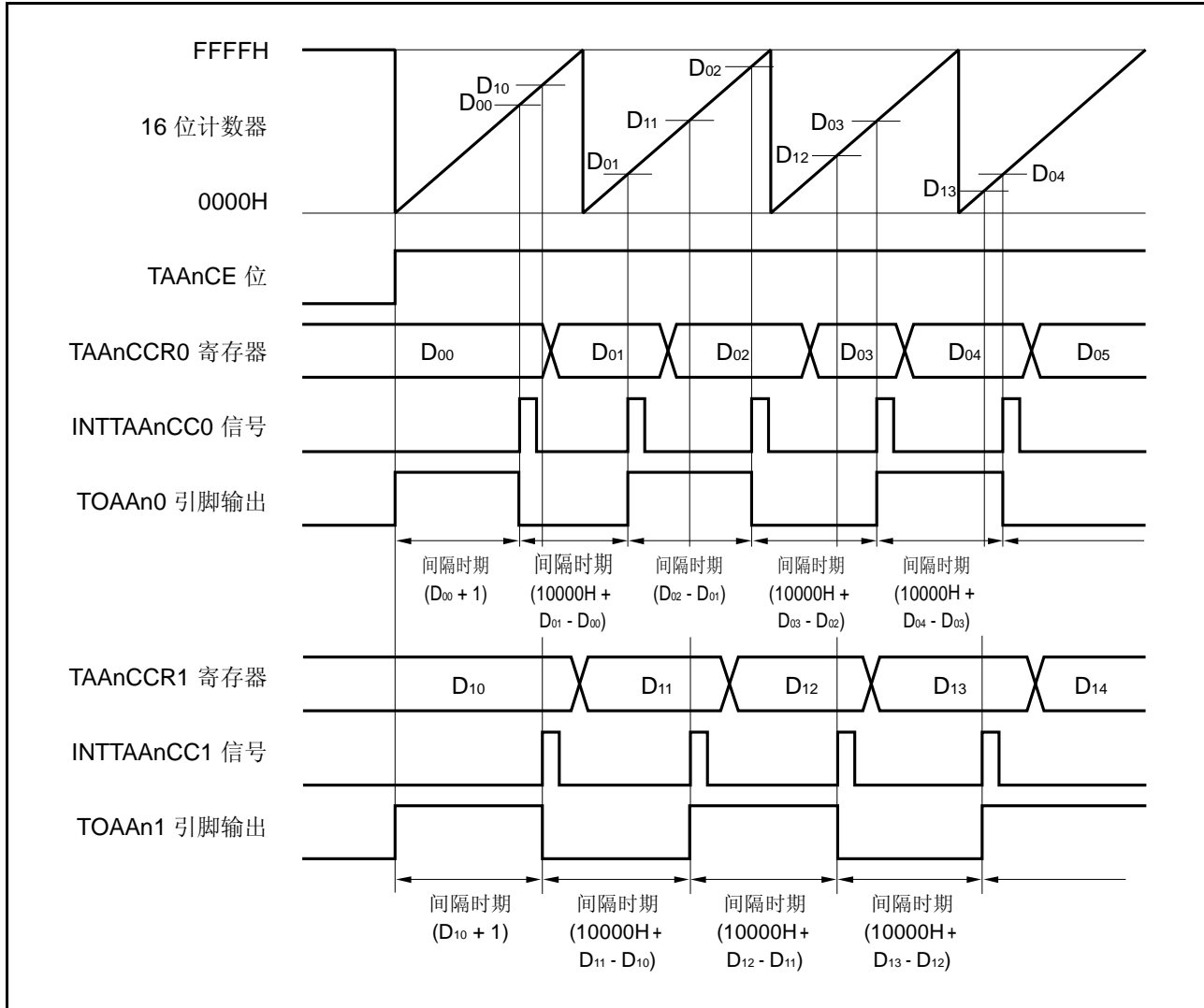
图 7-38. 自由运行定时器模式的软件处理流程 (捕获功能) (2/2)



(2) 自由运行定时器模式的操作时序

(a) TAA_nCCR_m 寄存器用作比较寄存器的间隔定时操作

当 16 位定时器/事件计数器 AA 用作间隔定时器, TAA_nCCR_m 寄存器用作比较寄存器时, 在每次检测到 INTTAA_nCC_m 信号后, 都需要通过软件设置一个比较值以产生下一个中断请求信号。



在自由运行定时器模式下进行时间间隔操作时, 一个通道可以设置两个时间间隔。

要进行时间间隔操作, 当检测到 INTTAA_nCC_m 信号后, 必须在中断服务程序中重置相应的 TAA_nCCR_m 寄存器的值。

重置 TAA_nCCR_m 寄存器的设置值可由下式计算, 其中“D_m”是间隔时间:

比较寄存器默认值: $D_m - 1$

比较寄存器的第二次及其以后的设置值: 上次设置值 + D_m

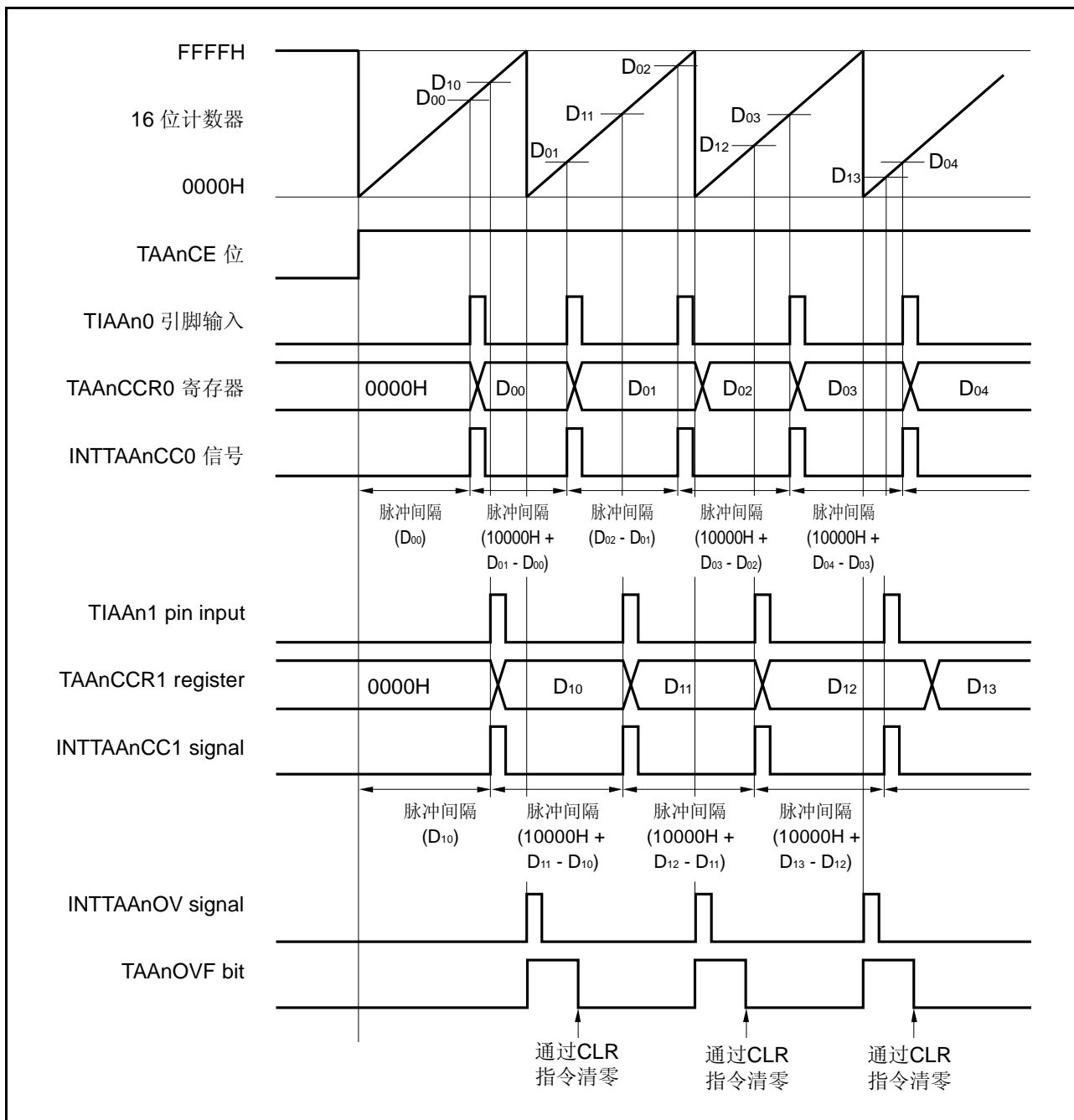
(若计算结果大于 FFFFH, 则从结果中减去 10000H, 并将此值设置给寄存器。)

备注 n=0 至 3, 5

m=0, 1

(b) TAAAnCCRm 寄存器用作捕获寄存器进行脉冲宽度测量

在用 TAAAnCCRm 寄存器作为捕获寄存器进行脉宽测量时，每次检测到 INTTAAAnCCm 信号时，需要通过软件读取捕获寄存器并计算间隔时间。

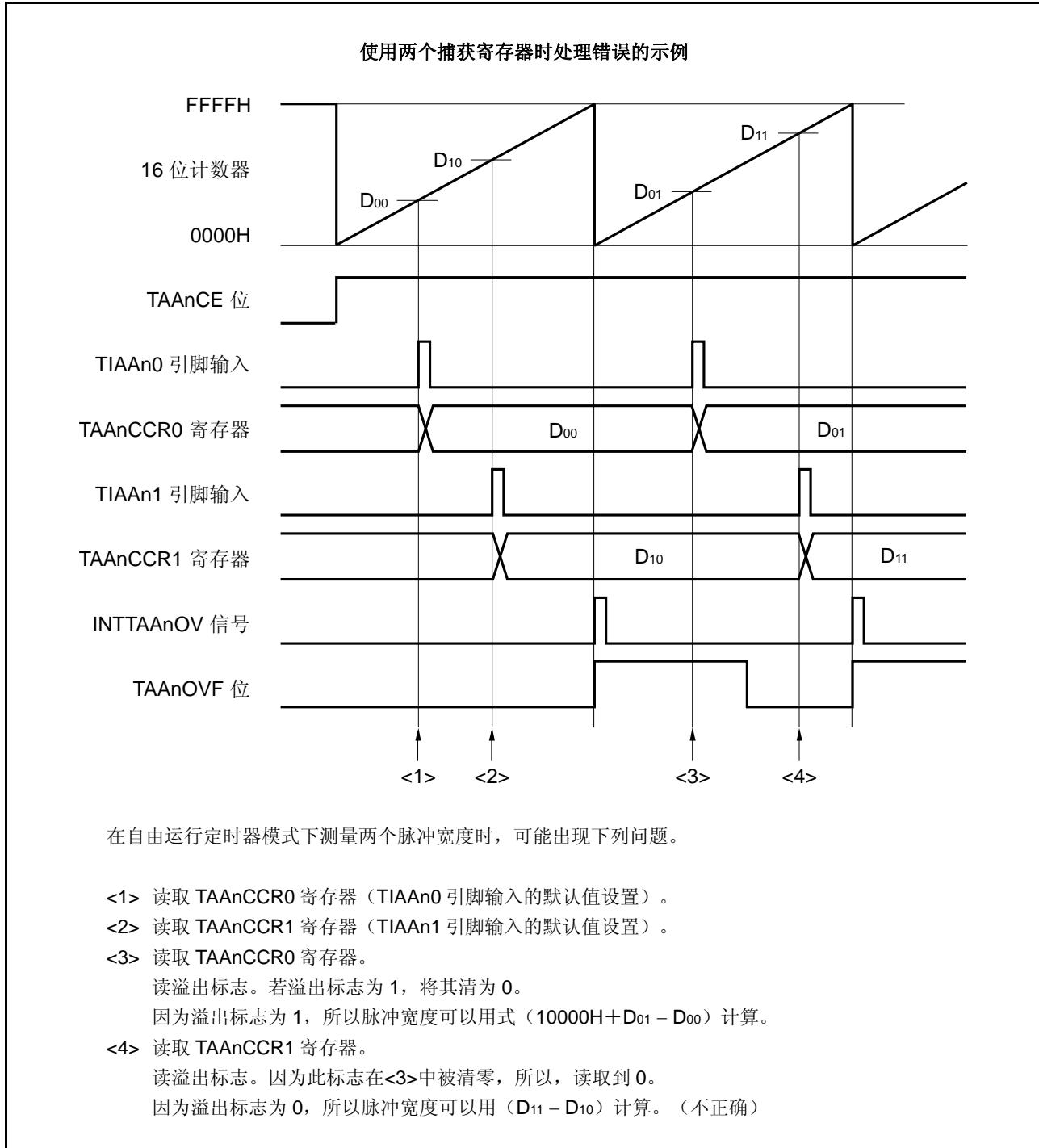


在自由运行定时器模式下执行脉宽测量时，可以用一个通道测量两个脉冲宽度。要测量脉冲宽度，可以与 INTTAAAnCCm 信号同步读取 TAAAnCCRm 寄存器的值，并计算读取值与先前读取值之间的差，得到脉冲宽度。

备注 n=0 至 3, 5
m=0, 1

(c) 当使用两个捕获寄存器时对溢出的处理

当使用两个捕获寄存器时必须格外小心。首先是一个不正确处理的示例，如下所示。

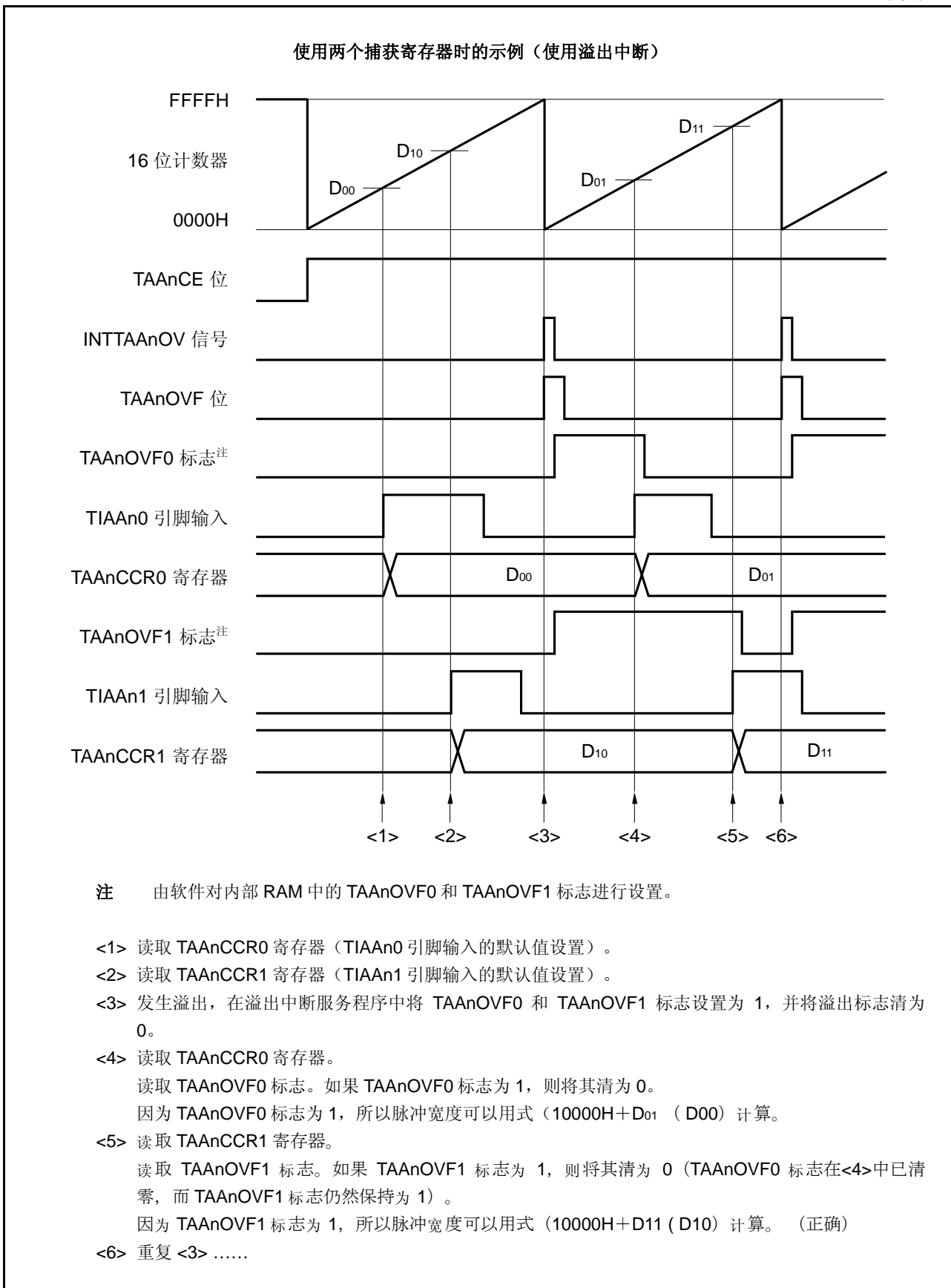


在自由运行定时器模式下测量两个脉冲宽度时，可能出现下列问题。

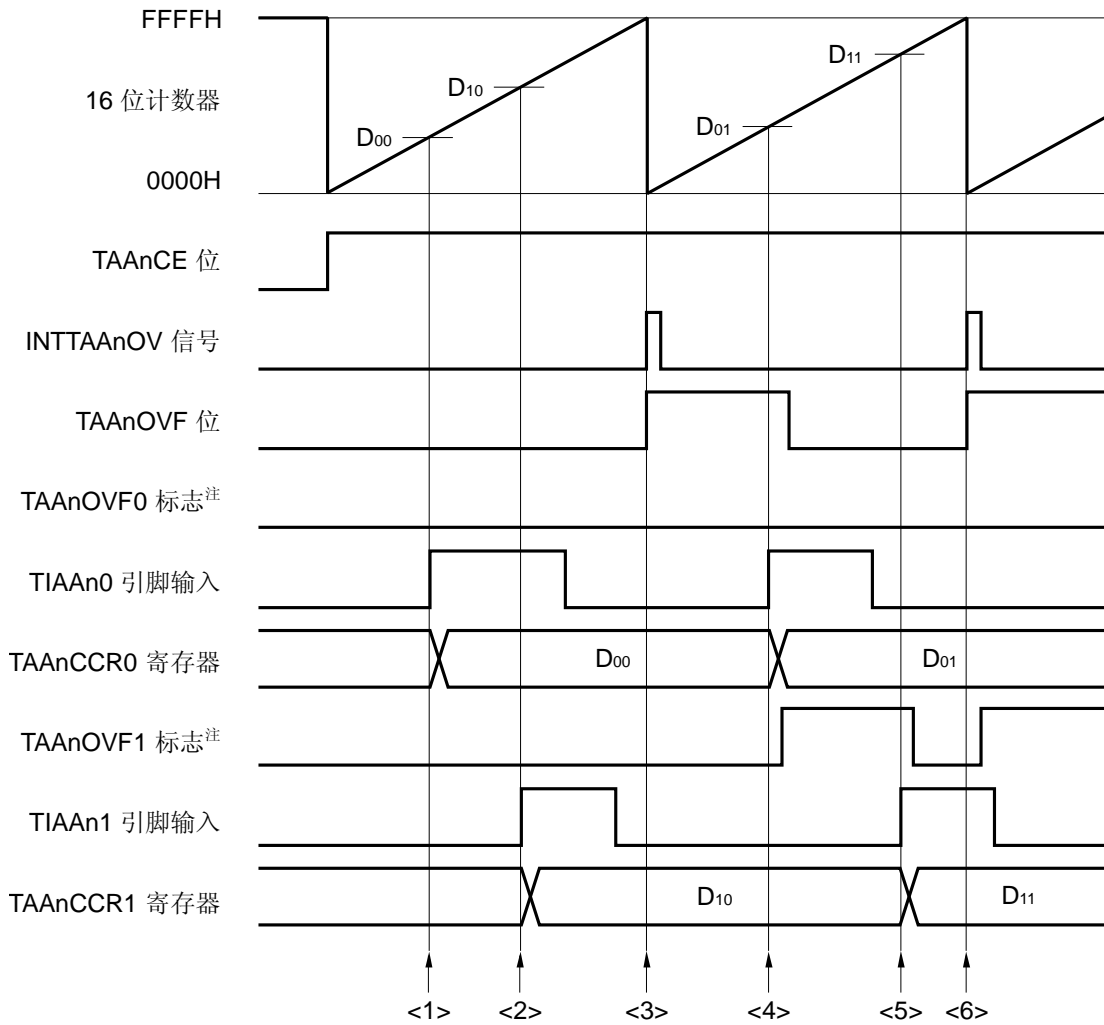
- <1> 读取 TAAAnCCR0 寄存器（TIAAn0 引脚输入的默认值设置）。
- <2> 读取 TAAAnCCR1 寄存器（TIAAn1 引脚输入的默认值设置）。
- <3> 读取 TAAAnCCR0 寄存器。
读溢出标志。若溢出标志为 1，将其清为 0。
因为溢出标志为 1，所以脉冲宽度可以用式 $(10000H + D_{01} - D_{00})$ 计算。
- <4> 读取 TAAAnCCR1 寄存器。
读溢出标志。因为此标志在<3>中被清零，所以，读取到 0。
因为溢出标志为 0，所以脉冲宽度可以用 $(D_{11} - D_{10})$ 计算。（不正确）

在使用两个捕获寄存器时，如果溢出标志由一个捕获寄存器清为 0，则另一个捕获寄存器可能得不到正确的脉冲宽度。

在使用两个捕获寄存器时，使用软件解决。关于如何使用软件的示例如下。



使用两个捕获寄存器时的示例（不使用溢出中断）



注 由软件对内部 RAM 中的 TAAAnOVF0 和 TAAAnOVF1 标志进行设置。

<1> 读取 TAAAnCCR0 寄存器（TIAAn0 引脚输入的默认值设置）。

<2> 读取 TAAAnCCR1 寄存器（TIAAn1 引脚输入的默认值设置）。

<3> 发生溢出，软件不作任何处理。

<4> 读取 TAAAnCCR0 寄存器。

读溢出标志。若溢出标志为 1，则只把 TAAAnOVF1 标志置为 1，并把溢出标志清为 0。

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

<5> 读取 TAAAnCCR1 寄存器。

读溢出标志。因为溢出标志在<4>中被清零，所以读取到 0。

读取 TAAAnOVF1 标志。如果 TAAAnOVF1 标志为 1，则将其清为 0。

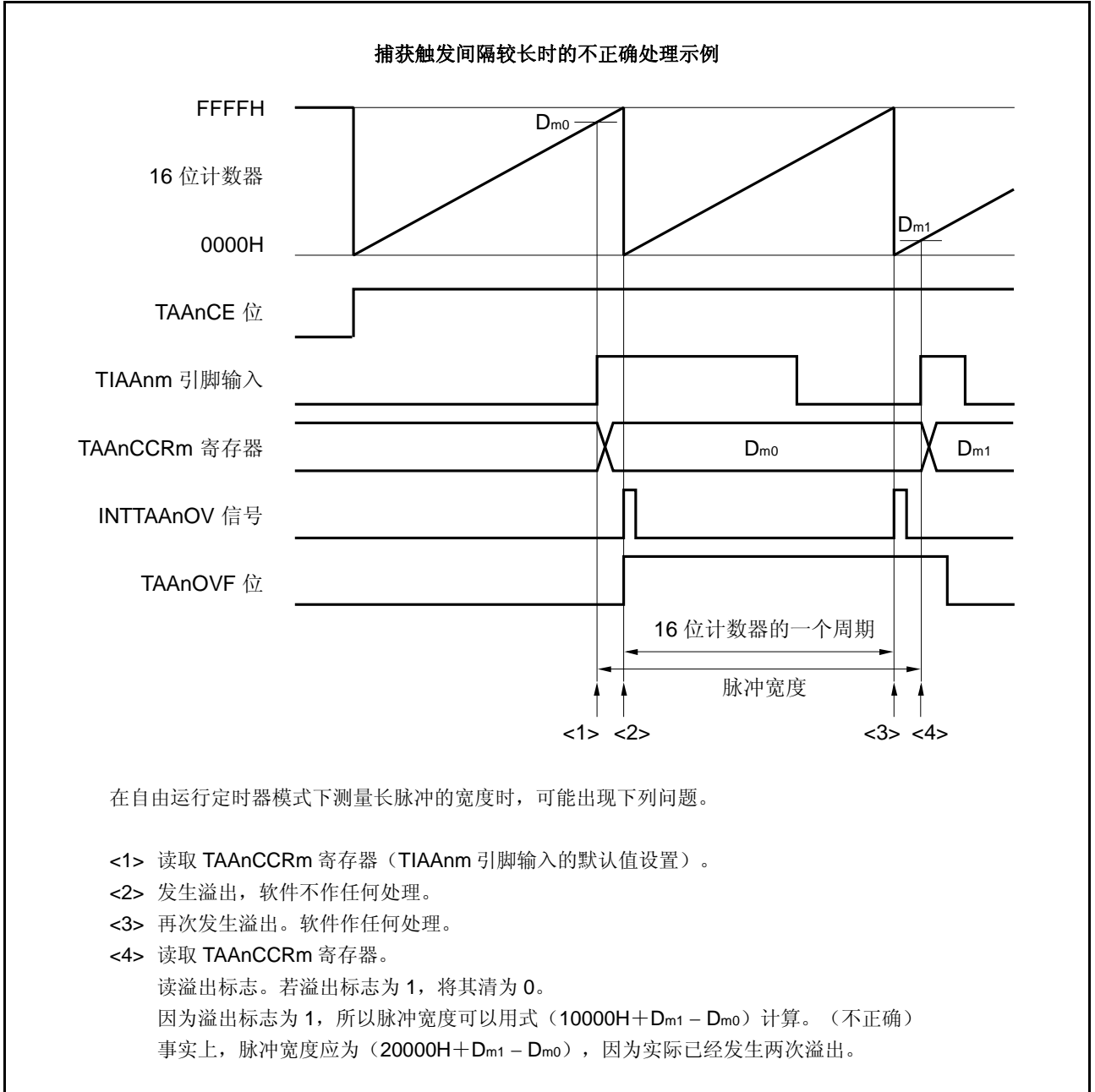
因为 TAAAnOVF1 标志为 1，所以脉冲宽度可以用式 $(10000H + D_{11} - D_{10})$ 计算。（正确）

<6> 重复 <3>

备注 n=0 至 3, 5

(d) 当捕获触发间隔较长时的溢出处理

如果脉冲宽度大于 16 位计数器的一个计数循环周期，则必须加以注意，因为从第一个捕获触发信号到下一个触发信号之间可能发生多次溢出。首先是一个不正确处理的示例，如下所示。

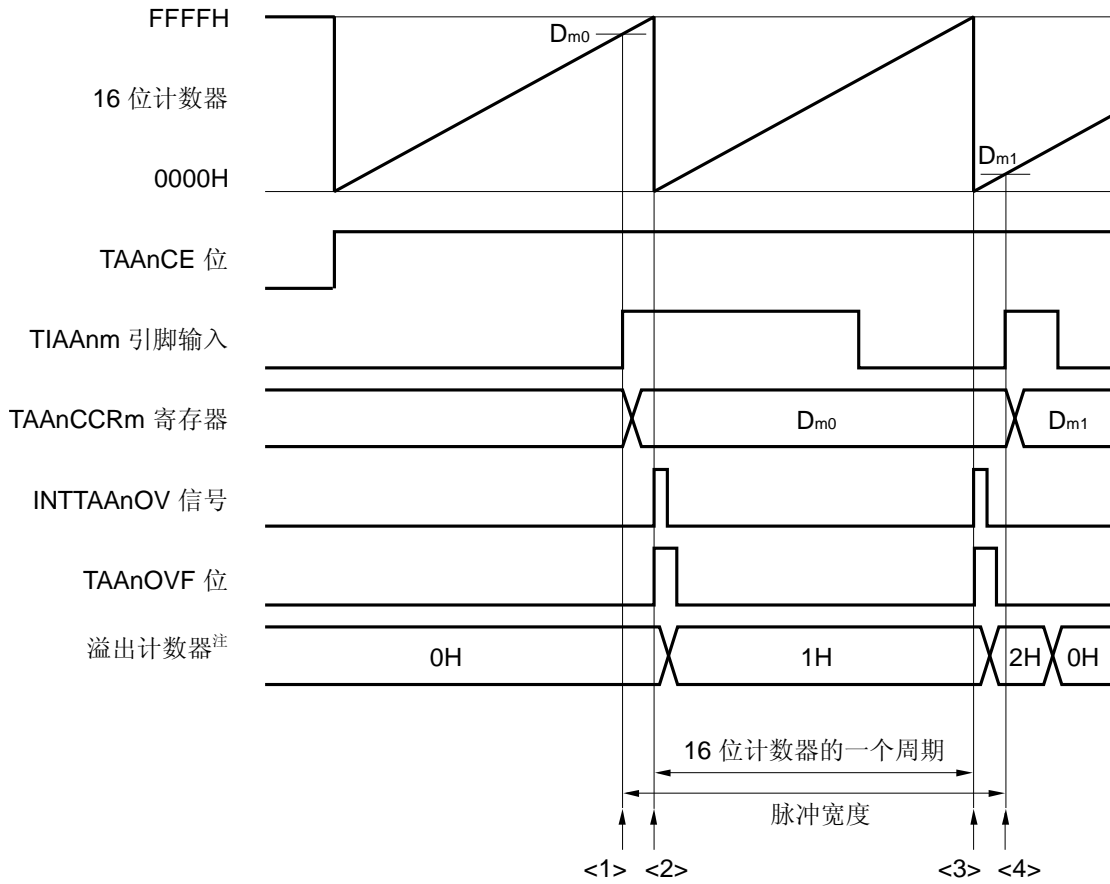


如果在捕获触发间隔期间，发生两次或更多次的溢出，则无法获得正确的脉冲宽度。

如果捕获触发间隔时间较长时，应该减慢计数时钟以延长 16 位计数器的循环周期，或者使用软件来解决。关于如何使用软件，示例如下。

备注 n=0 至 3, 5
m=0, 1

捕获触发间隔较长时的示例



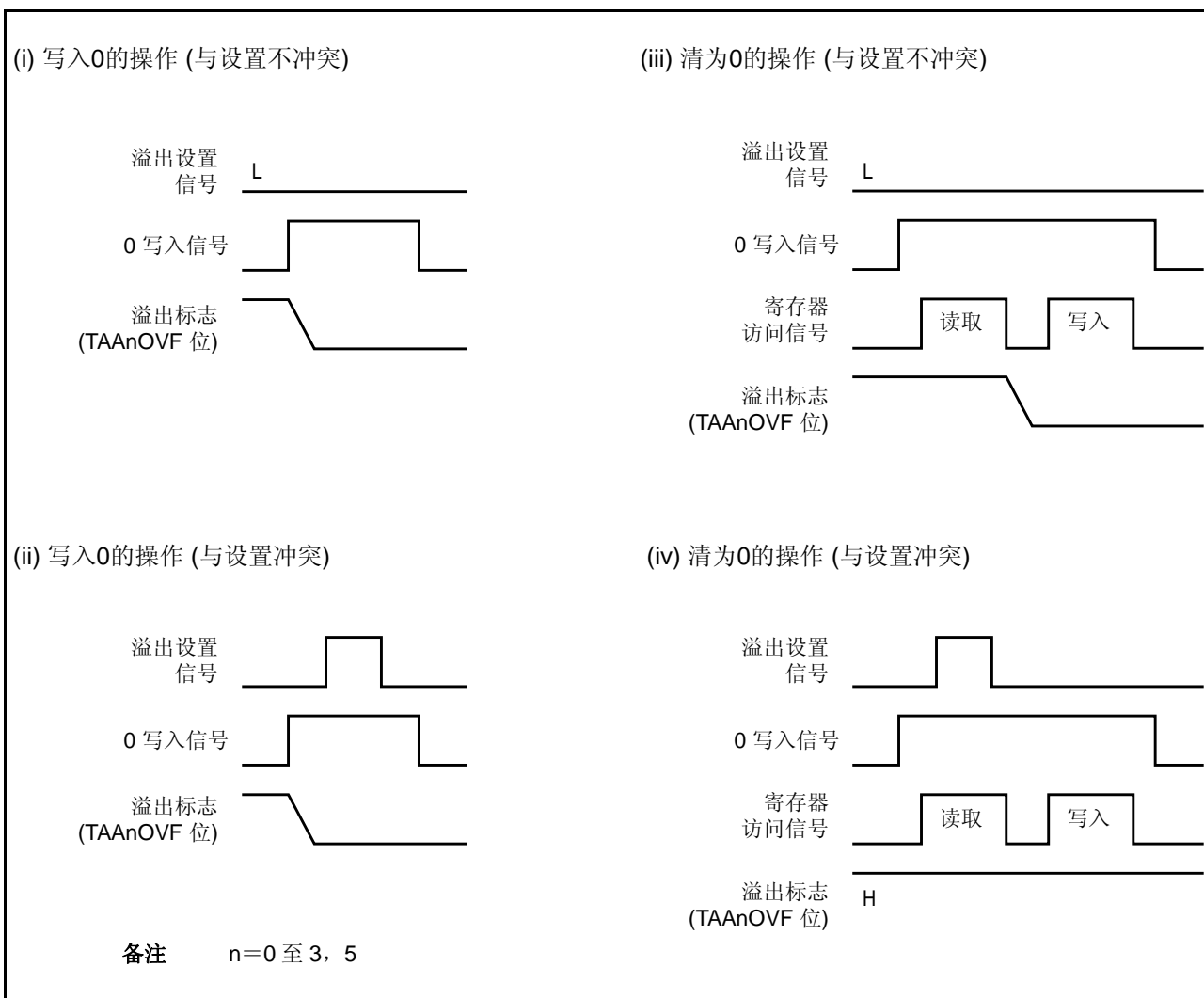
注 内部 RAM 中的溢出计数器由软件任意设置。

- <1> 读取 TAAAnCCRm 寄存器 (TIAAnm 引脚输入的默认值设置)。
- <2> 发生溢出, 在溢出中断服务程序中增加溢出计数器的值, 并将溢出标志清为 0。
- <3> 再次发生溢出。在溢出中断服务程序中增加 (+1) 溢出计数器的值, 并把溢出标志清为 0。
- <4> 读取 TAAAnCCRm 寄存器。
读取溢出计数器。
→ 当溢出计数器为“N”时, 脉冲宽度可用式 $(N \times 10000H + D_{m1} - D_{m0})$ 计算。
在本例中, 脉冲宽度为 $(20000H + D_{m1} - D_{m0})$, 因为溢出实际发生了两次。
溢出计数器清零 (0H)。

备注 n=0 至 3, 5
m=0, 1

(e) 清除溢出标志

通过 CLR 指令将 TAA_nOVF 位清为 0，以及将 8 位数据（第 0 位是 0）写入 TAA_nOPT0 寄存器，可以将溢出标志清 0。要准确检测溢出，当 TAA_nOVF 位为 1 时读取 TAA_nOVF 位，然后用位操作指令将溢出标志清零。



要将溢出标志清为 0，应读取溢出标志，检查该标志如果已经被设置为 1，用 CLR 指令把它清零。若在未检查溢出标志是否为 1 就把 0 写入溢出标志中，则溢出的设置信息可能因为写入 0 而被清除（上图中的 (ii)）。因此，即使实际发生了溢出，软件也可能会判断为无溢出发生。

如果用 CLR 指令将溢出标志清为 0 时，CLR 指令的执行与当前发生的溢出相冲突，则在清零指令执行后溢出标志仍然会保持置位（1）状态。

7.5.7 脉冲宽度测量模式 (TAA_nMD2 至TAA_nMD0 位=110)

在脉冲宽度测量模式下，当 TAA_nCTL0.TAA_nCE 位被置为 1 时，16 位计时器/事件计数器 AA 开始计数。每次检测到 TIAA_nm 引脚输入信号的有效边沿时，16 位计数器的计数值就存储在 TAA_nCCR_m 寄存器中，且 16 位计数器清除为 0000H。

在出现捕获中断请求信号 (INTTAA_nCC_m) 之后，可以通过读取 TAA_nCCR_m 寄存器来测量有效边沿的间隔。

选择 TIAA_n0 或 TIAA_n1 引脚作为捕获触发信号输入引脚。使用 TAA_nIOC1 寄存器将未使用的引脚指定为“无边沿检测”。

当一个外部时钟用作计数时钟，测试 TIAA_n1 引脚的脉冲宽度，因为外部时钟固定连接到 TIAA_n0 引脚。此时，清除 TAA_nIOC1.TAA_nIS1 和 TAA_nIOC1.TAA_nIS0 位为 00 (捕获触发输入 (TIAA_n0 引脚)：无边沿检测)。

图 7-39. 脉冲宽度测量模式的配置

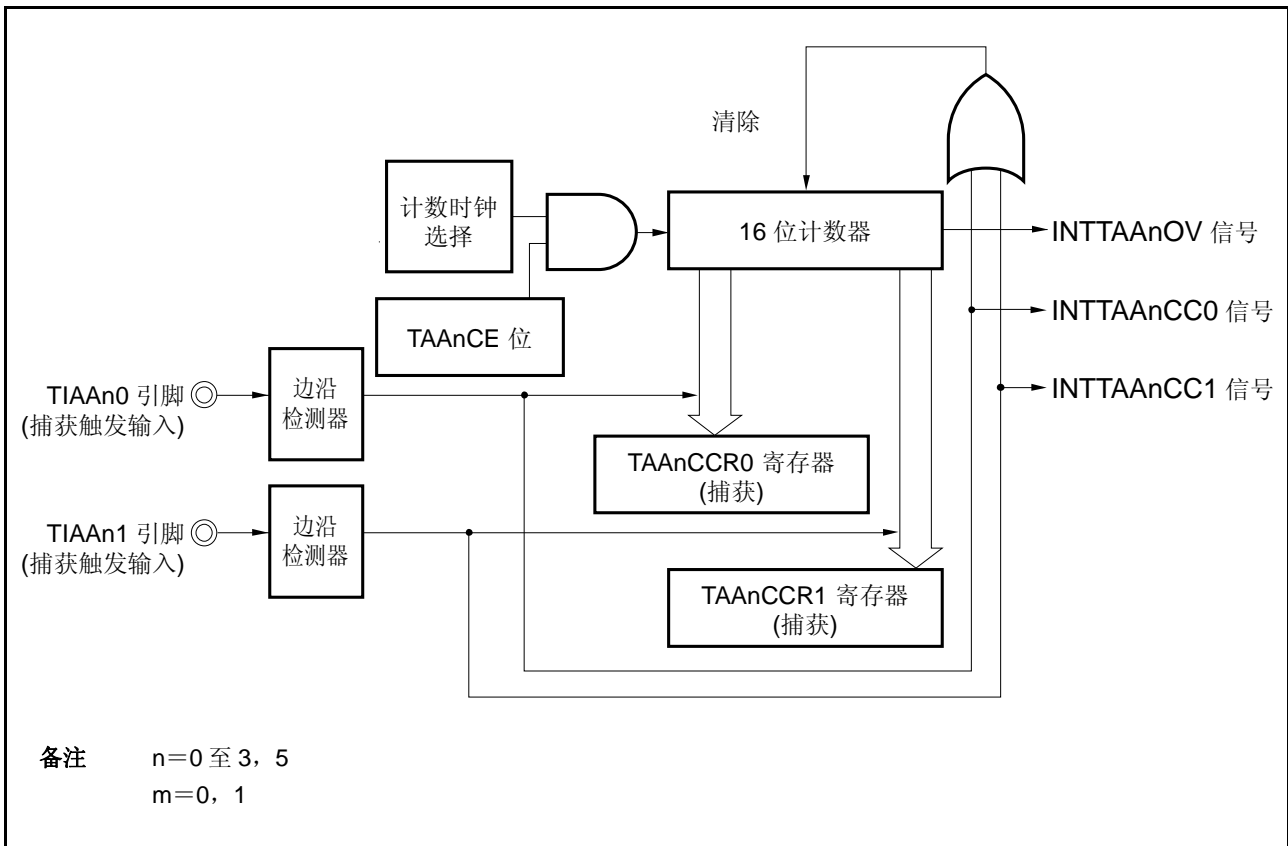
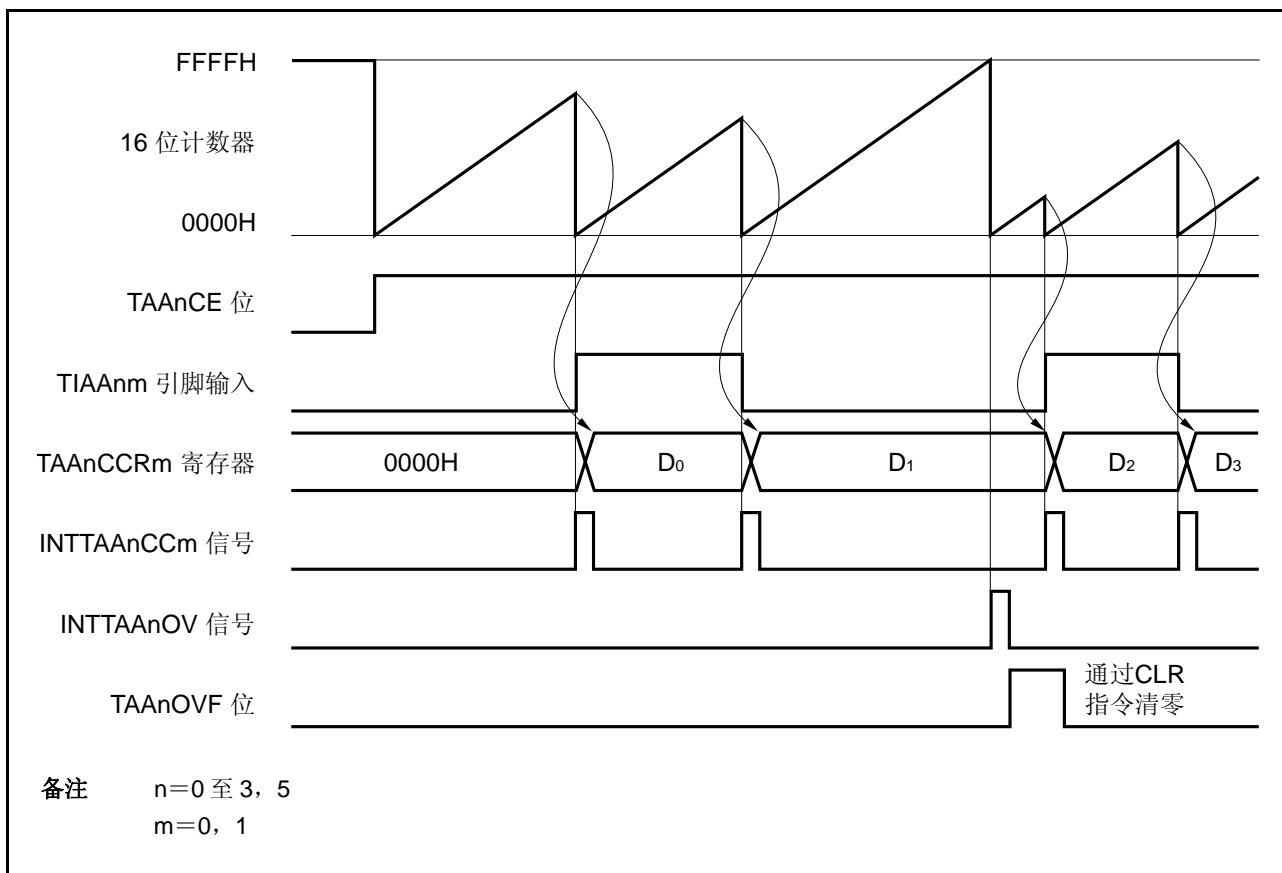


图 7-40. 脉冲宽度测量模式的基本时序



当 TAAAnCE 位被置为 1 时，16 位计数器开始计数。当检测到 TIAAnm 引脚输入信号的有效边沿时，16 位计数器的计数值被存储到 TAAAnCCRm 寄存器中，16 位计数器清除为 0000H，并产生一个捕获中断请求信号 (INTTAAAnCCm)。

脉冲宽度的计算如下。

$$\text{脉冲宽度} = \text{捕获值} \times (\text{计数时钟周期})$$

当 16 位计数器计数到 FFFFH 时，即使没有有效边沿输入 TIAAnm 引脚，在下一个计数时钟时，仍然会产生溢出中断请求信号 (INTTAAAnOV)，且计数器清除为 0000H，然后计数器继续计数。此时，溢出标志 (TAAAnOPT0.TAAAnOVF 位) 也被置为 1。使用软件执行 CLR 指令将溢出标志清为 0。

若溢出标志被设置为 1，则脉冲宽度的计算如下。

$$\text{脉冲宽度} = (10000H (\text{TAAAnOVF 位置位 (1) 的次数} + \text{捕获值}) \times (\text{计数时钟周期}))$$

备注 n=0 至 3, 5
m=0, 1

图 7-41. 脉冲宽度测量模式时的寄存器设置 (1/2)

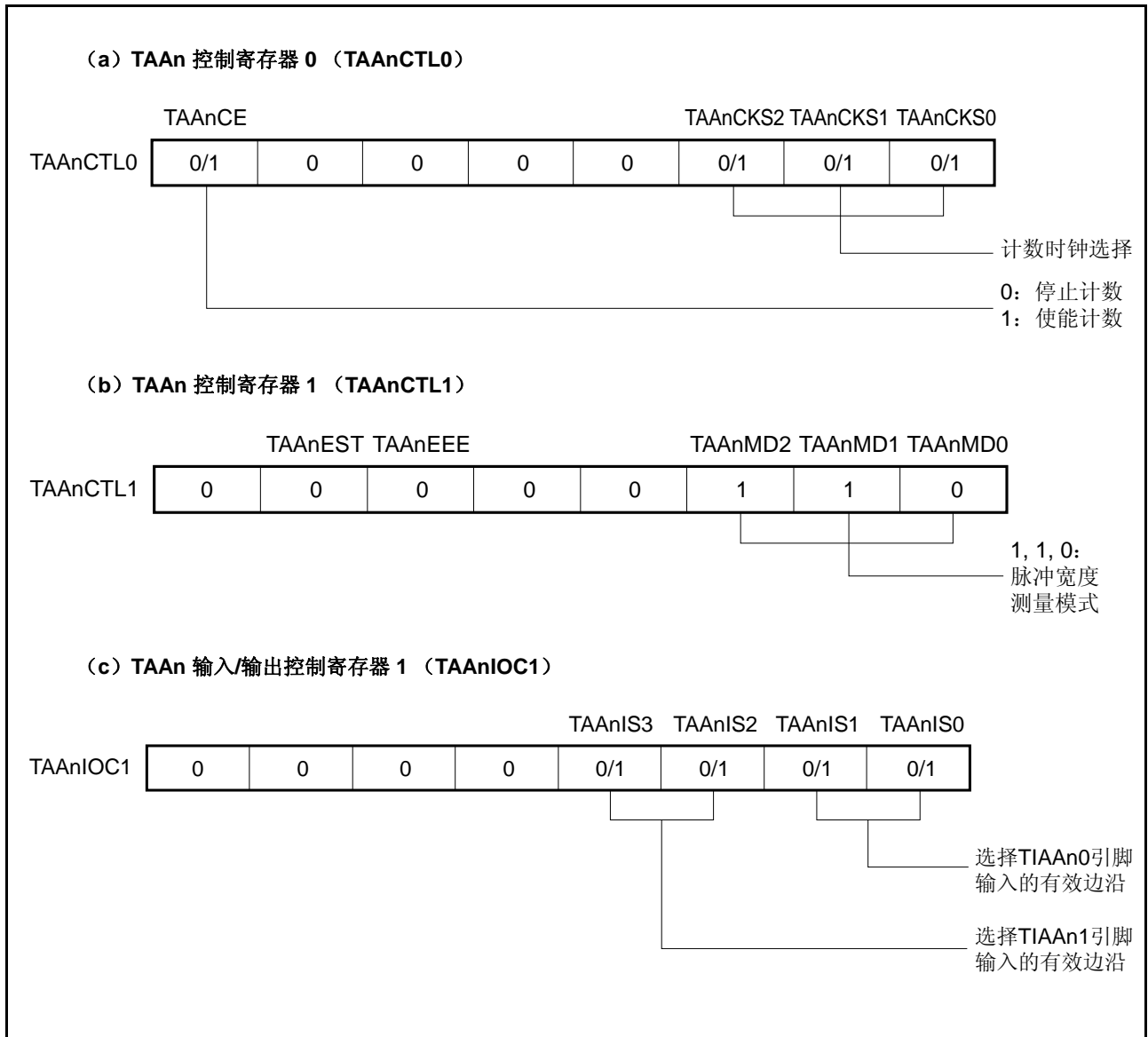
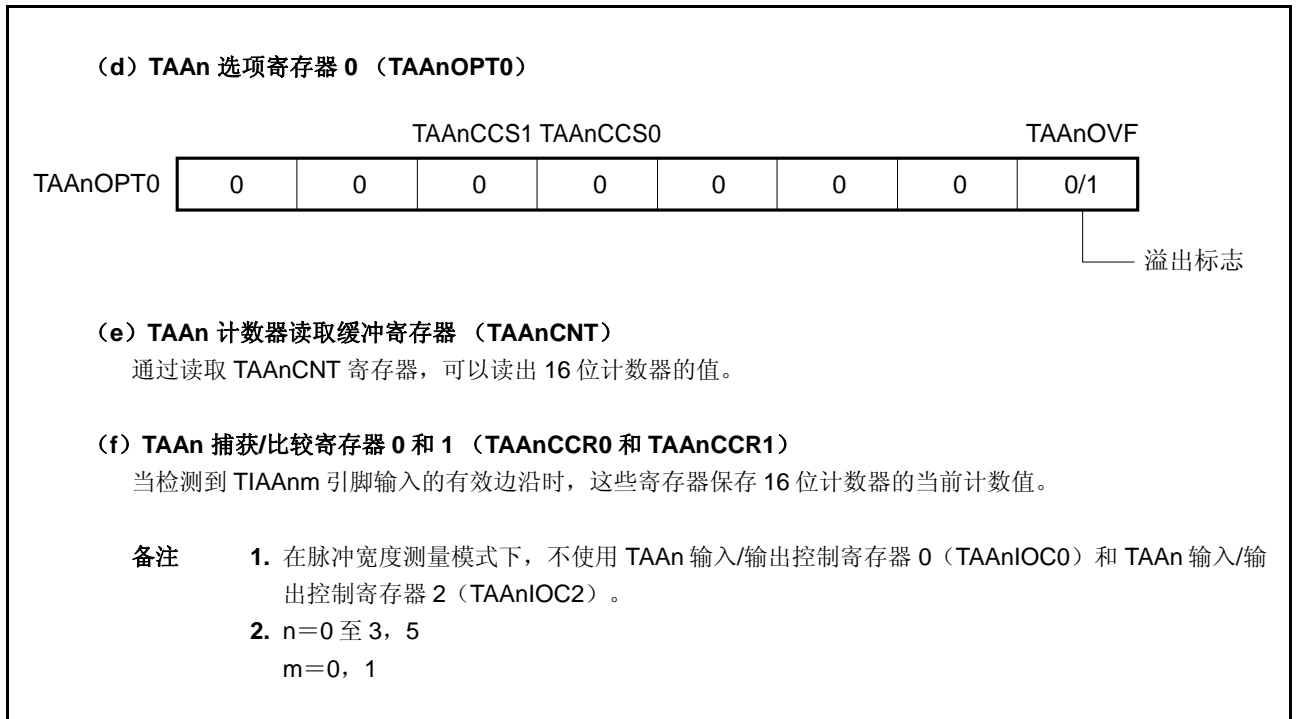
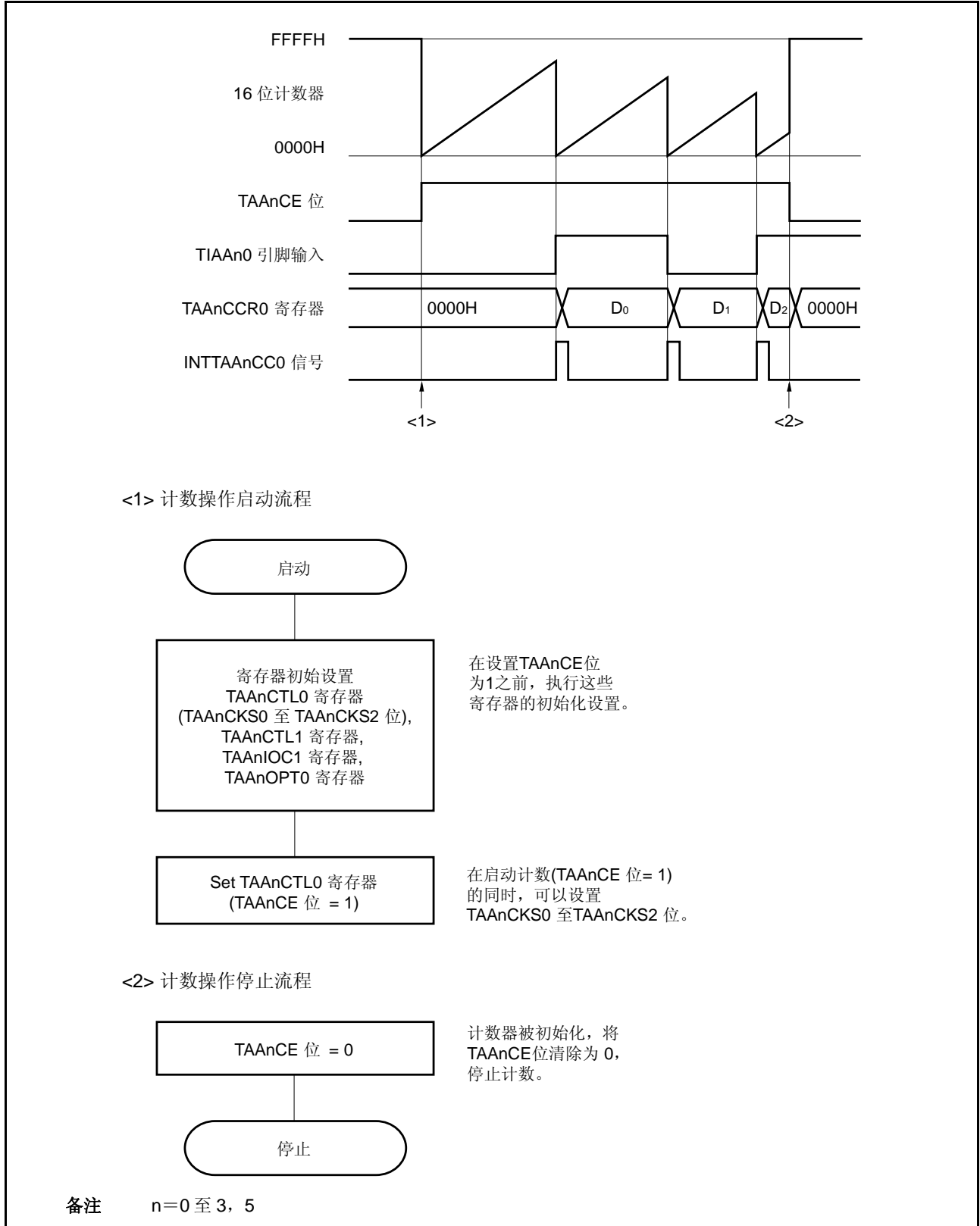


图 7-41. 脉冲宽度测量模式时的寄存器设置 (2/2)



(1) 脉冲宽度测量模式的操作系统

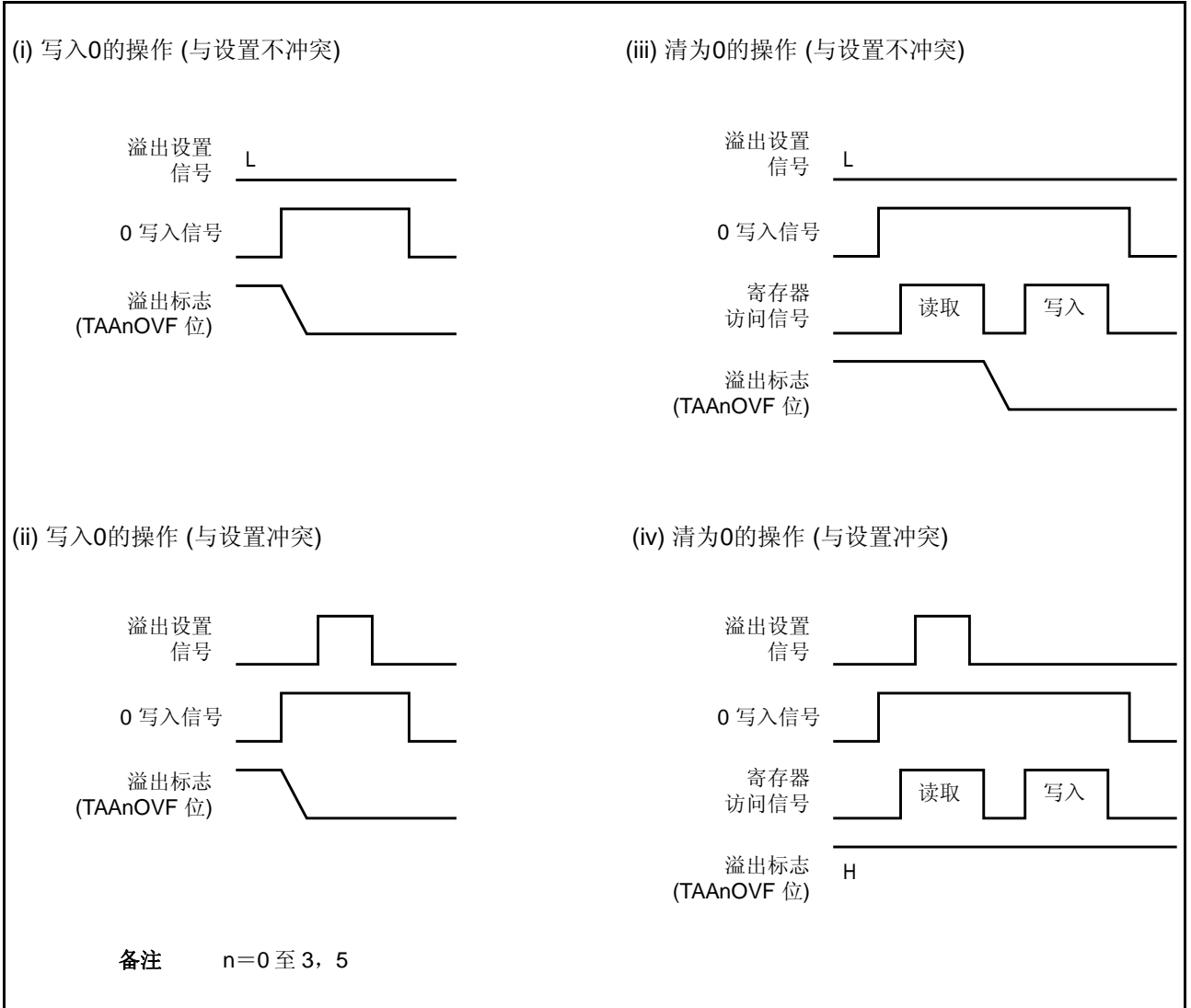
图 7-42. 脉冲宽度测量模式时的软件处理流程



(2) 脉冲宽度测量模式的操作时序

(a) 清除溢出标志

通过 CLR 指令将 TAA_nOVF 位清为 0，以及将 8 位数据（第 0 位是 0）写入 TAA_nOPT0 寄存器，可以将溢出标志清 0。要准确检测溢出，当 TAA_nOVF 位为 1 时读取 TAA_nOVF 位，然后用位操作指令将溢出标志清零。



要将溢出标志清为 0，应读取溢出标志，检查该标志如果已经被设置为 1，用 CLR 指令把它清零。若在未检查溢出标志是否为 1 就把 0 写入溢出标志中，则溢出的设置信息可能因为写入 0 而被清除（上图中的（ii））。因此，即使实际发生了溢出，软件也可能会判断为无溢出发生。

如果用 CLR 指令将溢出标志清为 0 时，CLR 指令的执行与当前发生的溢出相冲突，则在清零指令执行后溢出标志仍然会保持置位（1）状态。

7.5.8 定时器输出操作

下表显示了 TOAAn0 和 TOAAn1 引脚的操作和输出电平。

表 7-5. 各种模式下的定时器输出控制

操作模式	TOAAn1 引脚	TOAAn0 引脚
间隔计时器模式	方波输出	
外部事件计数模式	方波输出	-
外部触发脉冲输出模式	外部触发脉冲输出	方波输出
单次触发脉冲输出模式	单次触发脉冲输出	
PWM 输出模式	PWM (脉宽调制信号) 输出	
自由定时器模式	方波输出 (只有使用比较功能时可用)	
脉冲宽度测量模式	-	

备注 n=0 至 3, 5

表 7-6. 定时器输出控制位对 TOAAn0 和 TOAAn1 引脚影响的真值表

TAAAnIOC0.TAAAnOLm 位	TAAAnIOC0.TAAAnOEm 位	TAAAnCTL0.TAAAnCE 位	TOAAnm 引脚电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前为低电平, 开始计数后为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平, 开始计数后为低电平

备注 n=0 至 3, 5
m=0, 1

7.6 定时器调谐操作功能

定时器 AA 和定时器 AB 具有调谐操作功能。

定时器调谐操作功能用来调谐 V850ES/JG3-U 和 V850ES/JH3-U 的内部定时器，于是从定时器的捕获或比较寄存器数量（从定时器的定时器输出数量和比较匹配中断数量）可以加给主定时器。可以调谐的定时器在表 7-7 中列举如下。

表 7-7. 定时器的调谐操作模式

主定时器	从定时器
TAA1	TAA0
TAA3	TAA2
TAB0	TAA5

调谐操作功能有下列两种模式。

- PWM 输出模式
- 自由运行定时器模式

图 7-43 展示了 PWM 输出模式下的单独操作和 TAA0（作为主定时器）与 TAA1（作为从定时器）调谐操作的示例。

图 7-43. 使用 TAA0 与 TAA1 进行单独操作和调谐操作的区别

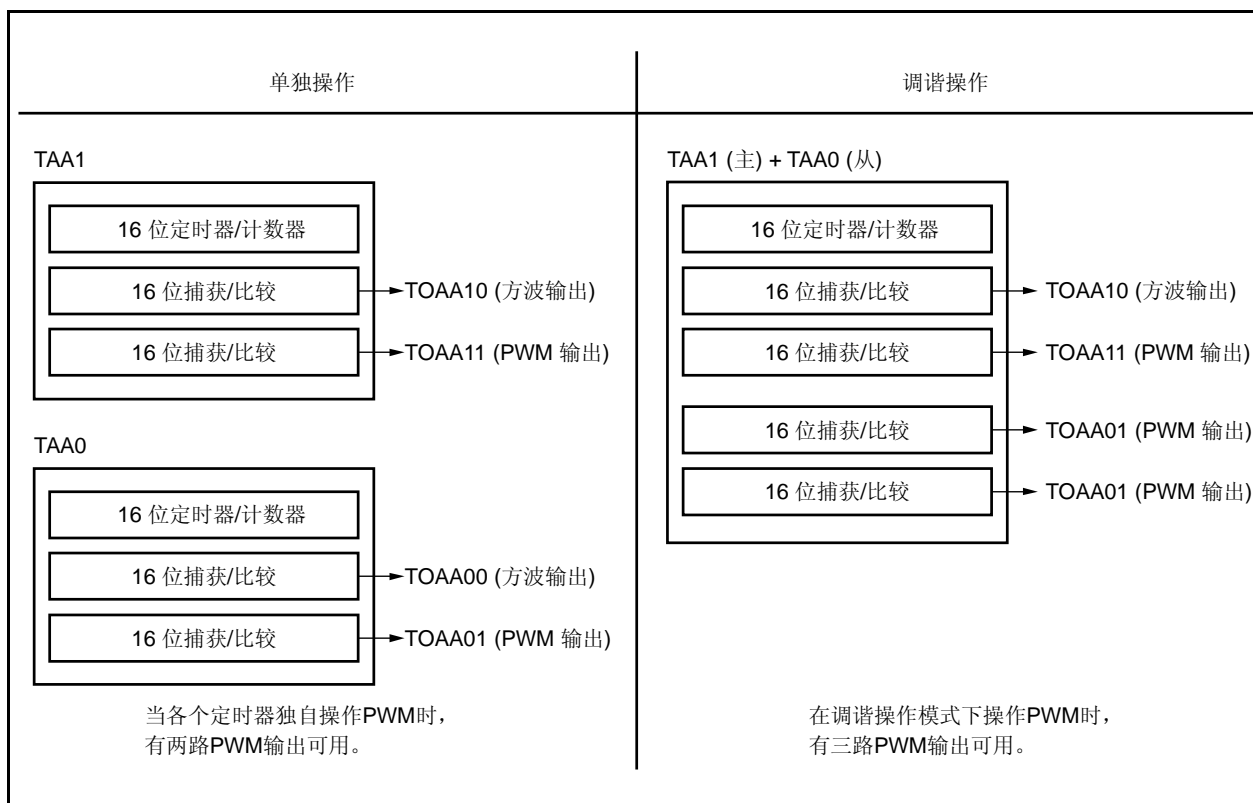


表 7-8 列出在调谐模式下可以使用的定时器模式，表 7-9 展示了单独操作和调谐操作的定时器输出功能的区别。
(√: 可以设置, ×: 不可以设置)

表 7-8. 调谐模式下可用的定时器模式

主定时器	从定时器	自由运行定时器模式	PWM 模式
TAA1	TAA0	√	√
TAA3	TAA2	√	√
TAB0	TAA5	√	√

表 7-9. 定时器输出功能

调谐通道	定时器	引脚	自由运行定时器模式		PWM 模式	
			单独操作	调谐操作	单独操作	调谐操作
Ch0	TAA1 (主)	TOAA10	PPG	←	切换	←
		TOAA11	PPG	←	PWM	←
	TAA0 (从)	TOAA00	PGP	←	切换	PWM
		TOAA01	PPG	←	PWM	←
Ch1	TAA3 (主)	TOAA30	PPG	←	切换	←
		TOAA31	PPG	←	PWM	←
	TAA2 (从)	TOAA20	PPG	←	切换	PWM
		TOAA21	PPG	←	PWM	←
Ch2	TAB0 (主)	TOAB00	PPG	←	切换	←
		TOAB01 至 TOAB03	PPG	←	PWM	←
	TAA5 (从)	TOAA50	PPG	←	切换	PWM
		TOAA51	PPG	←	PWM	←

备注 由主定时器的比较寄存器发送数据到从定时器的比较寄存器的时序如下。

- PPG: CPU 写入时序
- 切换, PWM, 三角波 PWM: 定时计数器和比较寄存器匹配 TOAA_n0 和 TOAB_m0 的时序

7.6.1 自由运行定时器模式（在定时器调谐操作期间）

该章节解释了定时器调谐操作的自由运行定时器模式。关于定时器调谐操作的组合，参见表 7-7。在本章节中，给出一个使用 TAA1 和 TAA0 进行定时器调谐操作的示例。

(i) 选择捕获/比较寄存器

当 TAA1 和 TAA0 相互连接，用作定时器调谐操作的自由运行定时器模式，TAA1 的两个捕获/比较寄存器和 TAA0 的两个捕获/比较寄存器可以组合使用。

如何组合捕获和比较寄存器没有限制，可以通过主或从定时器的 TAA_nCCSn 位来选择。选择比较寄存器时，在操作期间可以重写比较寄存器的设置值，重写方法是随时写入 (n = 0, 1)。

(ii) 溢出

如果计数器溢出，产生一个主定时器的溢出中断 (INTTAA1OV)，并且溢出标志 (TAA1OVF) 被置为“1”。

从定时器的溢出中断 (INTTAA0OV) 和溢出标志 (TAA0OVF) 不工作，并且总是低电平。

(1) 自由运行定时器模式的设置 (比较功能)

[初始设置]

主定时器: TAA1CTL0.TAA1CE = 0 (禁止操作)

从定时器: TAA0CTL0.TAA0CE = 0 (禁止操作)

[主定时器 (TAA1) 的初始设置]

- TAA1CTL1.TAA1MD2 至 TAA1CTL1.TAA1MD0 = 101 (自由运行定时器模式的设置)
- TAA1OPT0.TAA1CCS1 和 TAA1OPT0.TAA1CCS0 = 00 (捕获/比较的选择位设置为“比较”)
- TAA1CTL1.TAA1CKS2 至 TAA1CTL1.TAA1CKS0 (计数时钟的设置 (任意))
- TAA1CCR1 和 TAA1CCR0 寄存器被设置

[从定时器 (TAA0) 的初始设置]

- TAA0CTL1.TAA0SYE = 1 (定时器调谐操作的设置)
- TAA0CTL1.TAA0MD2 至 TAA0CTL1.TAA0MD0 = 101 (自由运行定时器模式的设置)
- TAA0OPT0.TAA0CCS1 和 TAA0OPT0.TAA0CCS0 = 00 (捕获/比较的选择位设置为“比较”)
- TAA0CCR0 和 TAA0CCR1 寄存器被设置

备注 主定时器和从定时器的初始设置可以按照任意顺序执行。

[启动计数]

<1> 设置主定时器的 TAA1CTL0.TAA1CE 为 1。

<2> 启动计数。

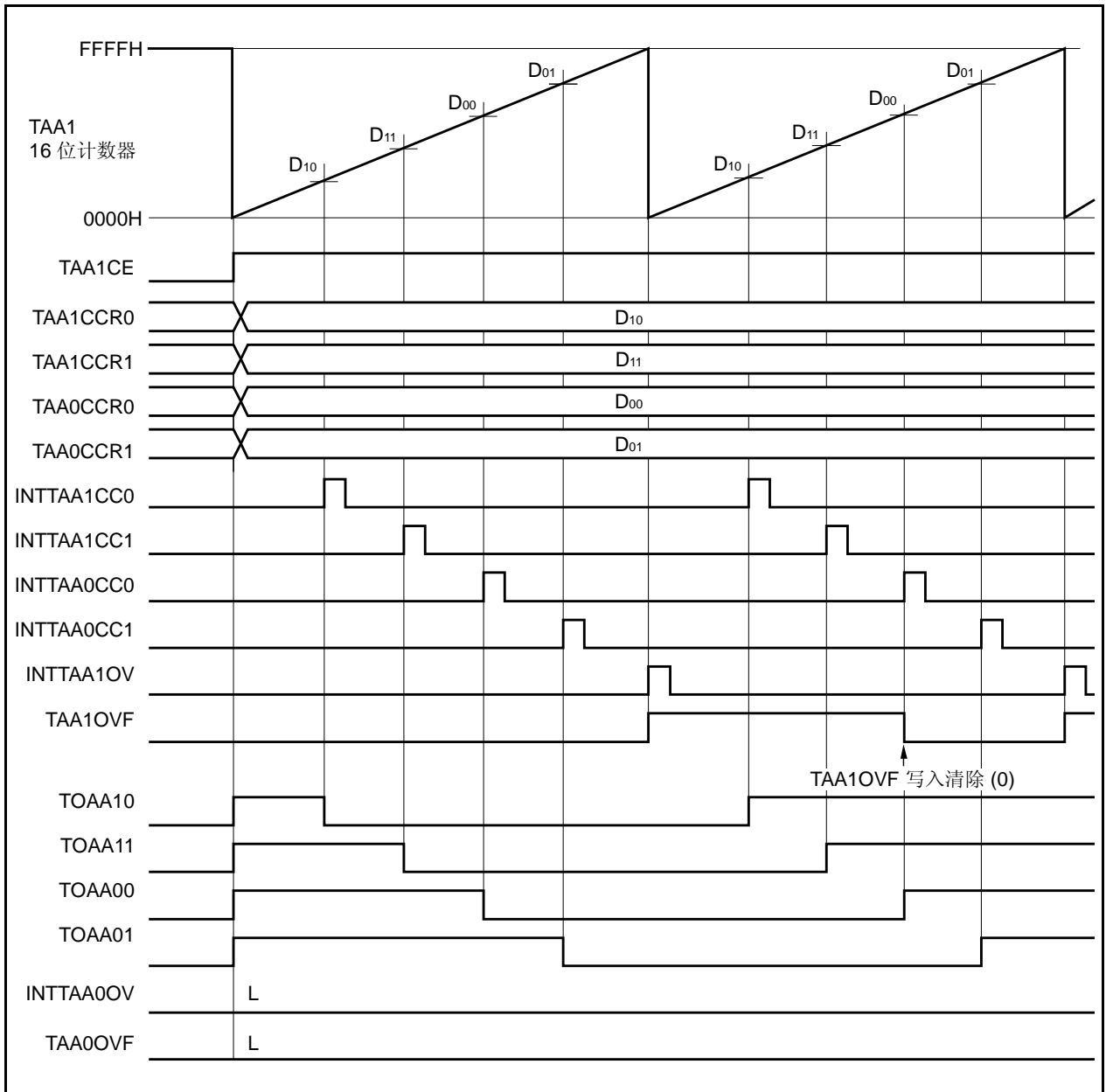
<3> 在操作期间更改寄存器的设置

- 可以重写比较寄存器 (随时写入)

[结束条件]

- 设置主定时器的 TAA1CTL0.TAA1CE 为 0。

图 7-44. 自由运行模式的时序示例 (比较功能)



(2) 自由运行定时器模式的设置 (捕获功能)

[初始设置]

主定时器: TAA1CTL0.TAA1CE = 0 (禁止操作)

从定时器: TAA0CTL0.TAA0CE = 0 (禁止操作)

[主定时器 (TAA1) 的初始设置]

- TAA1CTL1.TAA1MD2 至 TAA1CTL1.TAA1MD0 = 101 (自由运行定时器模式的设置)
- TAA1OPT0.TAA1CCS1 和 TAA1OPT0.TAA1CCS0 = 11 (捕获/比较的选择位设置为“捕获”)
- TAA1CTL1.TAA1CKS2 至 TAA1CTL1.TAA1CKS0 (计数时钟的设置 (任意))
- TAA1IOC1.TAA1IS3 至 TAA1IOC1.TAA1IS0 (捕获触发有效边沿的指定)

[从定时器 (TAA0) 的初始设置]

- TAA0CTL1.TAA0SYE = 1 (定时器调谐操作的设置)
- TAA0CTL1.TAA0MD2 至 TAA0CTL1.TAA0MD0 = 101 (自由运行定时器模式的设置)
- TAA0OPT0.TAA0CCS1 和 TAA0OPT0.TAA0CCS0 = 11 (捕获/比较的选择位设置为“捕获”)
- TAA0IOC1.TAA0IS3 至 TAA0IOC1.TAA0IS0 (捕获触发有效边沿的指定)

备注 主定时器和从定时器的初始设置可以按照任意顺序执行。

[启动计数]

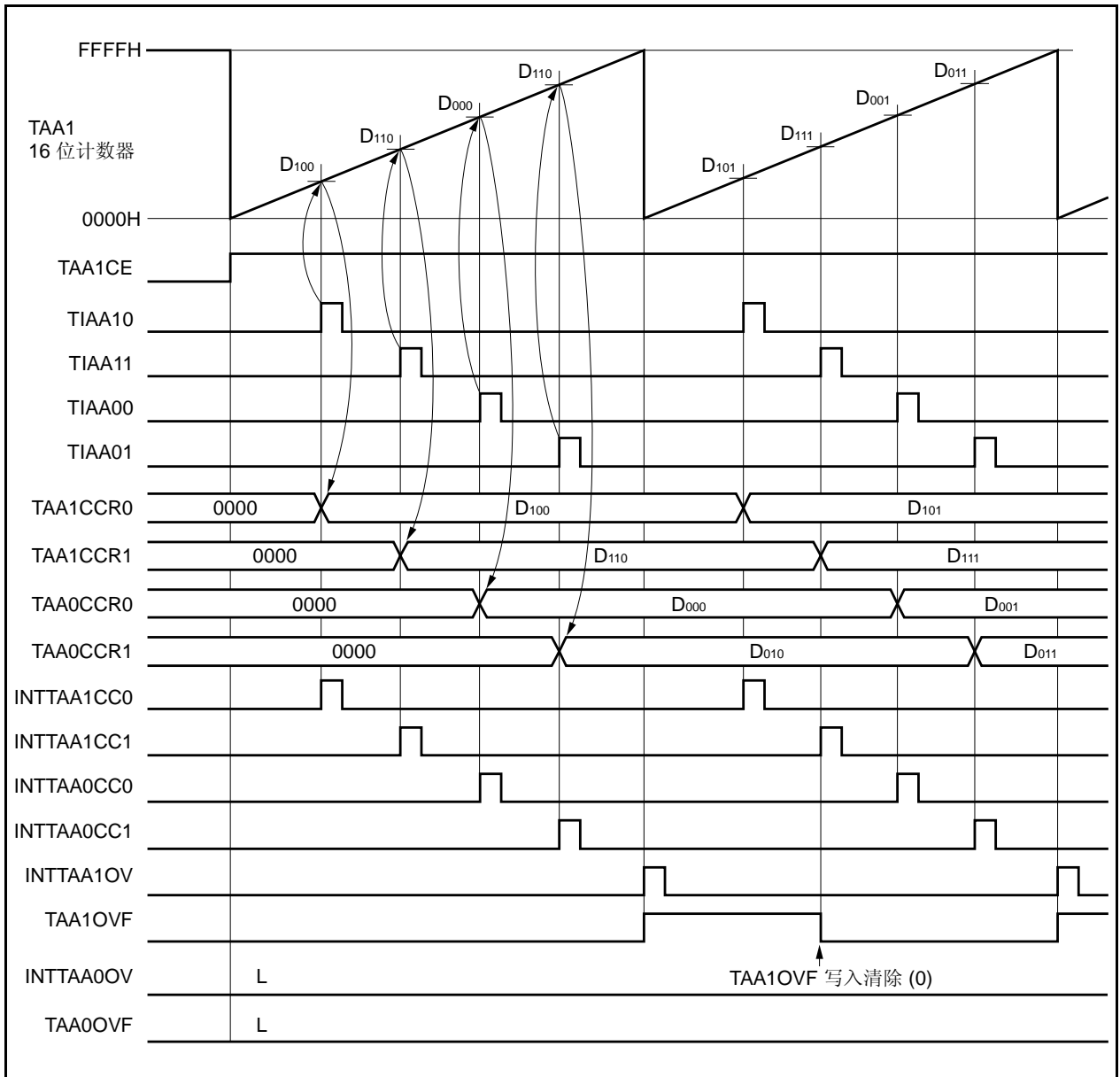
<1> 设置主定时器的 TAA1CTL0.TAA1CE 为 1。

<2> 启动计数

[结束条件]

- 设置主定时器的 TAA1CTL0.TAA1CE 为 0。

图 7-45. 自由运行模式的时序示例 (捕获功能)



(3) 自由运行定时器模式的设置 (捕获/比较一起使用)

使用 TAA0 作为捕获寄存器且 TAA1 作为比较寄存器的示例如下。

[初始设置]

主定时器: TAA1CTL0.TAA1CE = 0 (禁止操作)

从定时器: TAA0CTL0.TAA0CE = 0 (禁止操作)

[主定时器 (TAA1) 的初始设置]

- TAA1CTL1.TAA1MD2 至 TAA1CTL1.TAA1MD0 = 101 (自由运行定时器模式的设置)
- TAA1OPT0.TAA1CCS1 和 TAA1OPT0.TAA1CCS0 = 11 (捕获/比较的选择位设置为“捕获”)
- TAA1CTL1.TAA1CKS2 至 TAA1CTL1.TAA1CKS0 (计数时钟的设置 (任意))
- TAA1.TAA0IS3 至 TAA1.TAA1IS0 (捕获触发有效边沿的指定)

[从定时器 (TAA0) 的初始设置]

- TAA0CTL1.TAA0SYE = 1 (定时器调谐操作的设置)
- TAA0CTL1.TAA0MD2 至 TAA0CTL1.TAA0MD0 = 101 (自由运行定时器模式的设置)
- TAA0OPT0.TAA0CCS1 和 TAA0OPT0.TAA0CCS0 = 00 (捕获/比较的选择位设置为“比较”)
- TAA0CCR0 和 TAA0CCR1 寄存器被设置。

备注 主定时器和从定时器的初始设置可以按照任意顺序执行。

[启动计数]

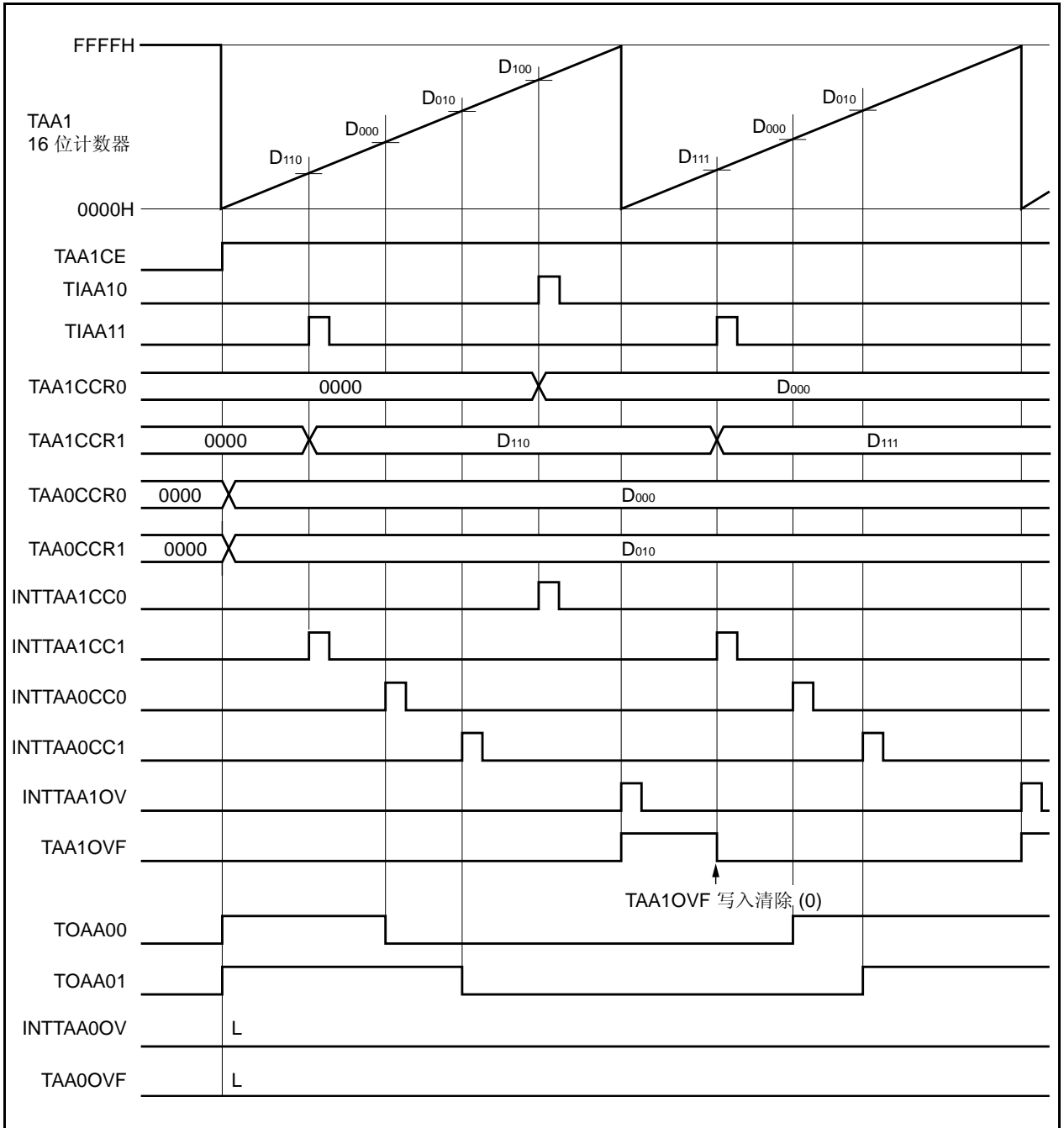
<1> 设置主定时器的 TAA1CTL0.TAA1CE 为 1。

<2> 启动计数

[结束条件]

- 设置主定时器的 TAA1CTL0.TAA1CE 为 0。

图 7-46. 自由运行模式的时序示例 (捕获/比较一起使用)



7.6.2 PWM输出模式（定时器调谐操作期间）

本章节解释了定时器调谐操作的 PWM 输出模式。关于定时器调谐操作的组合，参见表 7-7。在本章节中，给出一个使用 TAB0 和 TAA5 进行定时器调谐操作的示例。

主定时器（TAB0）的 TAB0CCR0 寄存器被用作比较寄存器来设置周期，主定时器（TAB0）的 TAB0CCR1, TAB0CCR2 和 TAB0CCR3 寄存器以及从定时器（TAA5）的 TAA5CCR0 和 TAA5CCR1 寄存器被用作比较寄存器来设置占空比。

在操作期间，可以重写这些比较寄存器，重写方式是批量写入。

当主定时器（TAB0）的 TAB0CCR1 寄存器被重写时，批量写入使能，并且在产生中断时，主和从定时器的所有比较寄存器被重写，或者相同的值被写入。主定时器（TAB0）的 TAB0CCR0 寄存器的值和定时计数器的值匹配时，产生该中断。

（1）PWM 输出模式的设置

[初始设置]

主定时器：TAB0CTL0.TAB0CE = 0（禁止操作）

从定时器：TAA5CTL0.TAA5CE = 0（禁止操作）

[主定时器（TAB0）的初始设置]

- TAB0CTL1.TAB0MD2 至 TAB0CTL1.TAB0MD0 = 100（PWM 输出模式的设置）
- TAB0OPT0.TAB0CCS3 至 TAB0OPT0.TAB0CCS0 = 0000（捕获/比较的选择位设置为“比较”）
- TAB0CCR0, TAB0CCR1, TAB0CCR2 和 TAB0CCR3 寄存器被设置。

[从定时器（TAA5）的初始设置]

- TAA5CTL1.TAA5SYE = 1（定时器调谐操作的设置）
- TAA5CTL1.TAA5MD2 至 TAA5CTL1.TAA5MD0 = 101（自由运行定时器模式的设置）
- TAA5OPT0.TAA5CCS1 和 TAA5OPT0.TAA5CCS0 = 00（捕获/比较的选择位设置为“比较”）
- TAA5CCR0 和 TAA5CCR1 寄存器被设置。

备注 主定时器和从定时器的初始设置可以按照任意顺序执行。

[启动计数]

<1> 设置主定时器的 TAB0CTL0.TAB0CE 为 1。

<2> 启动计数

<3> 在操作期间更改寄存器的设置

- 可以重写比较寄存器（批量重写）

[结束条件]

- 设置主定时器的 TAB0CTL0.TAB0CE 为 0。

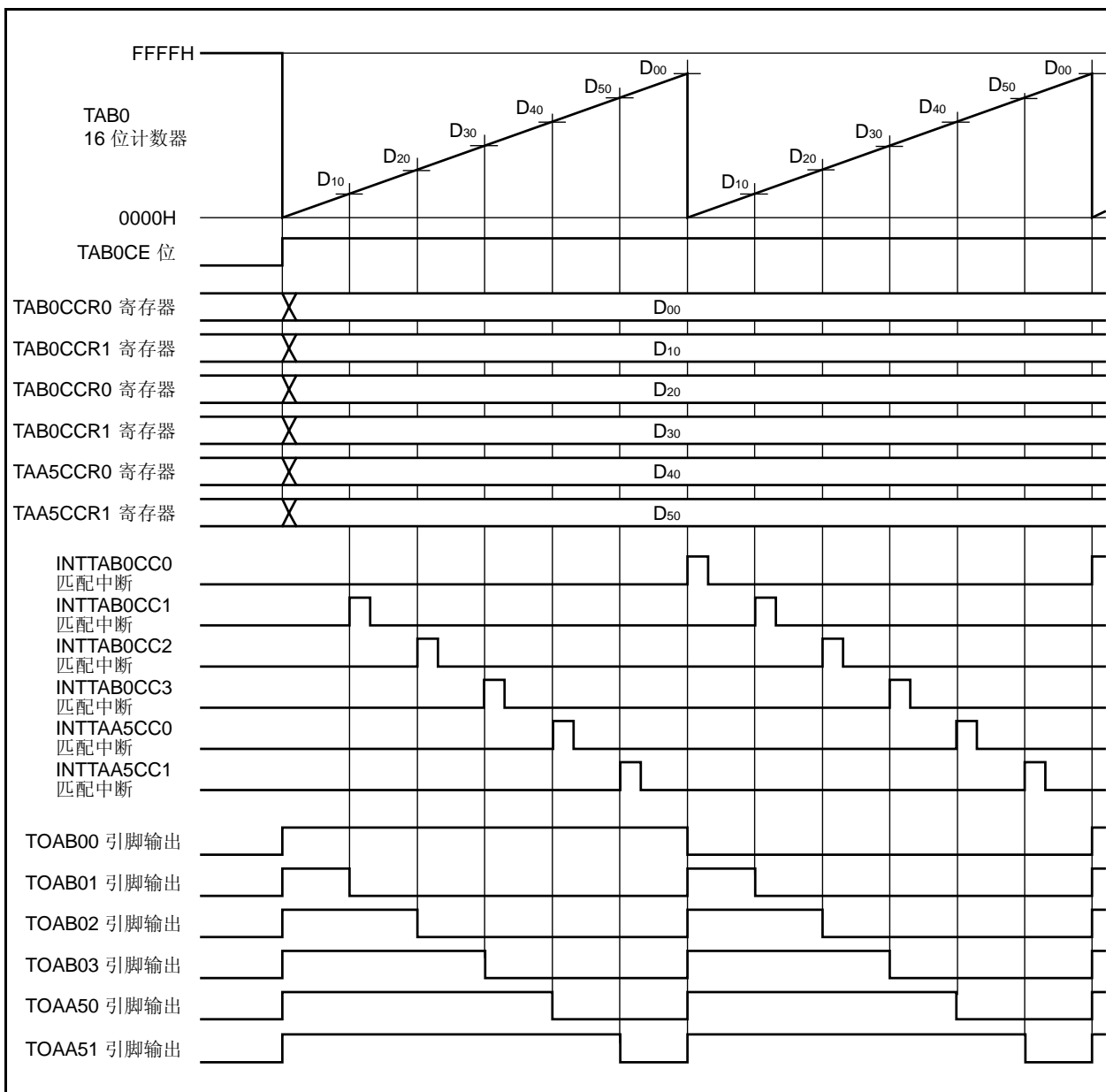
[批量写入]

在 PWM 输出模式下，通过写入主定时器 (TAB0) 的 TAB0CCR1 寄存器，可以使能下次批量写入。在所有必须要重写的比较寄存器已经被重写之后，主定时器 (TAB0) 的 TAB0CCR1 寄存器必须进行重写。

当定时计数器的值和周期比较寄存器 (TAB0CCR0) 的值匹配时，执行批量写入。

如果没有重写主定时器 (TAB0) 的 TAB0CCR1 寄存器，即使另一个比较寄存器被重写，批量写入也不会使能。结果，即使当定时计数器的值和周期比较寄存器 (TAB0CCR0) 的值匹配时，比较寄存器的值也不会被重写。

图 7-47. 调谐 PWM 功能的时序示例 (TAB0, TAA5)



7.7 同时启动功能

定时器 AA 和定时器 AB 具有定时器调谐操作功能。

通过使用同时启动功能，可以执行主定时器和从定时器的同步操作，操作启动时序和向上计数时序都可以同步。

只有在 PWM 输出模式下可以使用同时启动功能。

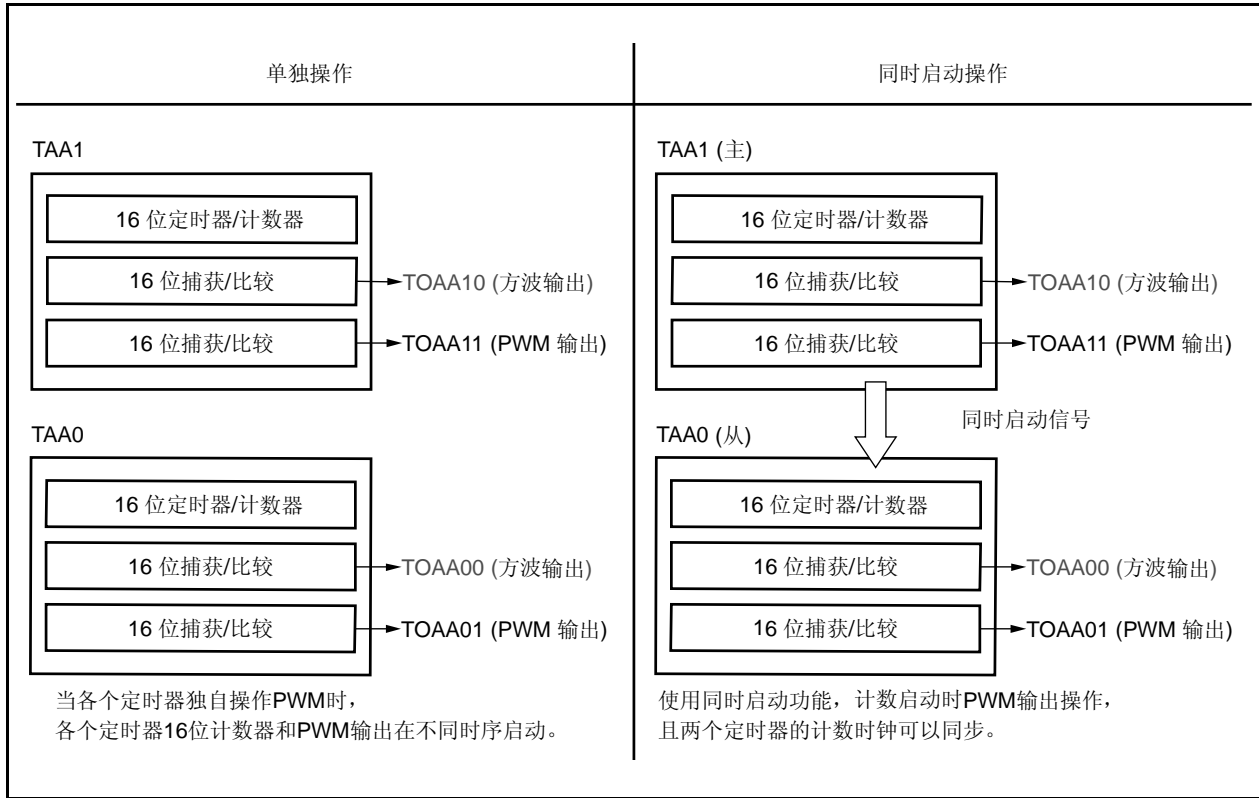
可以使用同时启动功能的定时器组合在表 7-10 中列举。

表 7-10. 定时器同时启动功能

主定时器	从定时器
TAA1	TAA0
TAA3	TAA2
TAB0	TAA5

图 7-48 展示了 PWM 输出模式下的单独操作和 TAA0（作为主定时器）与 TAA1（作为从定时器）同时启动操作的示例。

图 7-48. 使用 TAA0 与 TAA1 进行单独操作和同时启动操作的区别



7.7.1 PWM输出模式（同时启动操作）

本章节解释了同时启动功能的操作。其中 TAA1 作为主定时器且 TAA0 作为从定时器。

当 TAA1CTL0.TAA0CE 位被置 1 时，主定时器（TAA1）和从定时器（TAA0）同时开始操作。从定时器运行于主定时器（TAA1）提供的计数时钟。但是，在从定时器开始操作后，即使在主定时器（TAA1）16 位计数器的值和 TAA1CCR0 寄存器匹配，主定时器（TAA1）的 16 位计数器被清除为 0000H，从定时器（TAA0）的 16 位计数器不被清除，因为各个定时器单独操作。

同理可知，如果主定时器（TAA1）比较寄存器的值被批量写入方式重写，从定时器的比较寄存器不受影响。

[初始设置]

主定时器：TAA1CTL0.TAA1CE = 0（禁止操作）

从定时器：TAA0CTL0.TAA0CE = 0（禁止操作）

[主定时器（TAA1）的初始设置]

- TAA1CTL1.TAA1MD2 至 TAA1CTL1.TAA1MD0 = 100（PWM 输出模式的设置）
- TAA1CTL1.TAA1CKS2 至 TAA1CTL1.TAA1CKS0（计数时钟的设置（任意））
- TAA1CCR1, TAA1CCR0（捕获触发有效边沿的指定）
- TAA1IOC0（捕获触发有效边沿的指定）

[从定时器（TAA0）的初始设置]

- TAA0CTL1.TAA0SYE = 1, TAA0SYM = 1（同时启动操作）
- TAA0CTL1.TAA0MD2 至 TAA0CTL1.TAA0MD0 = 100（PWM 输出模式的设置）
- TAA0CCR0, TAA1CCR1（捕获触发有效边沿的指定）
- TAA0IOC0（捕获触发有效边沿的指定）

备注 主定时器和从定时器的初始设置可以按照任意顺序执行。

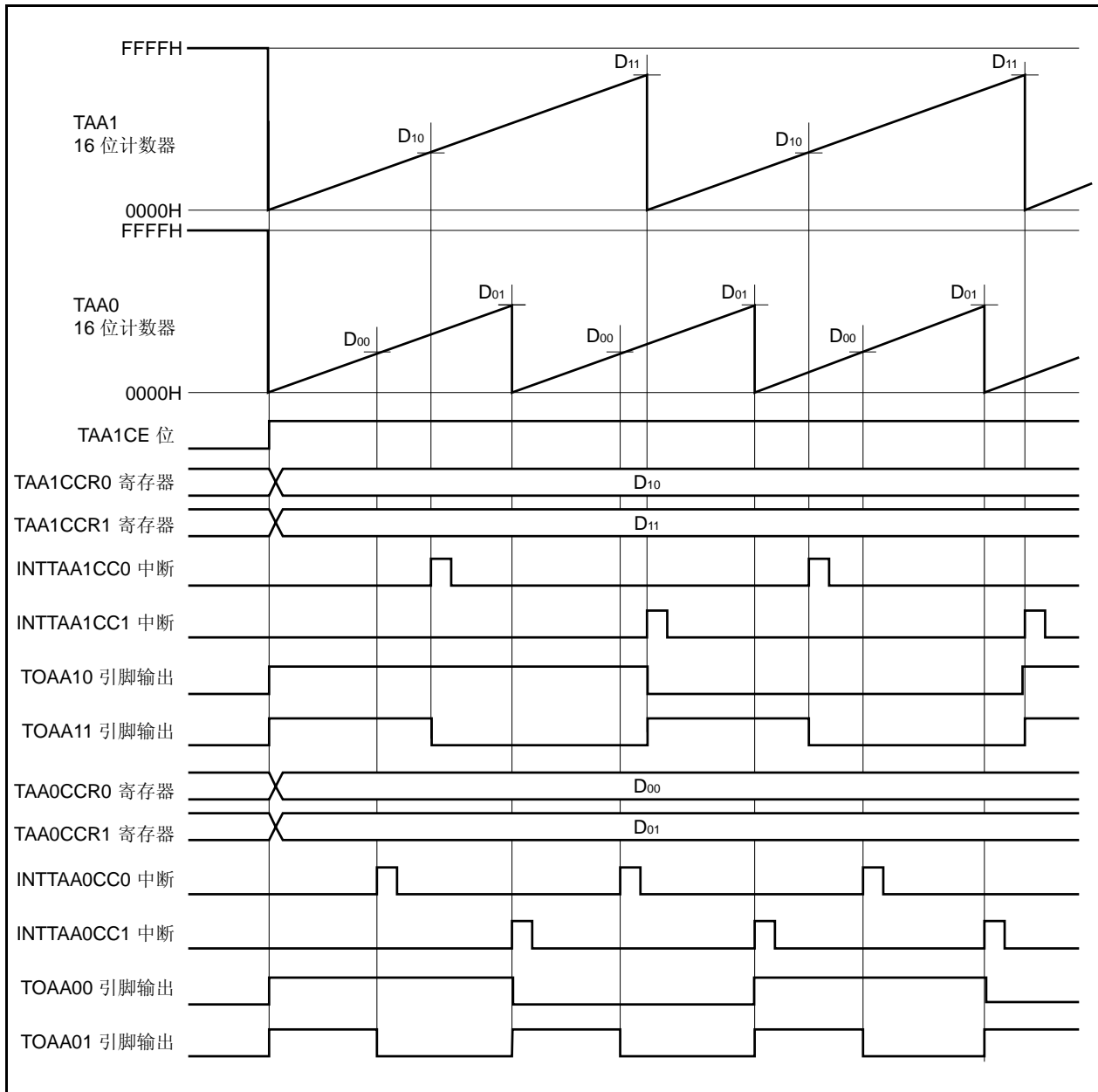
[启动计数]

- <1> 设置主定时器的 TAA1CTL0.TAA1CE 为 1。
- <2> 启动计数
- <3> 在操作期间更改寄存器的设置
 - 可以重写比较寄存器（随时写入）

[结束条件]

- 设置主定时器的 TAA1CTL0.TAA0CE 为 0。

图 7-49. 同时启动功能的时序示例 (TAA1: 主, TAA0: 从)



7.8 级联

该章节解释了在级联中连接 TAA 的两个通道组成 32 位捕获定时器的操作。

对于级联，必须设置为自由运行定时器模式，并且所有的捕获/比较寄存器必须被设置为捕获寄存器 (TAA0CCSn = 1)。

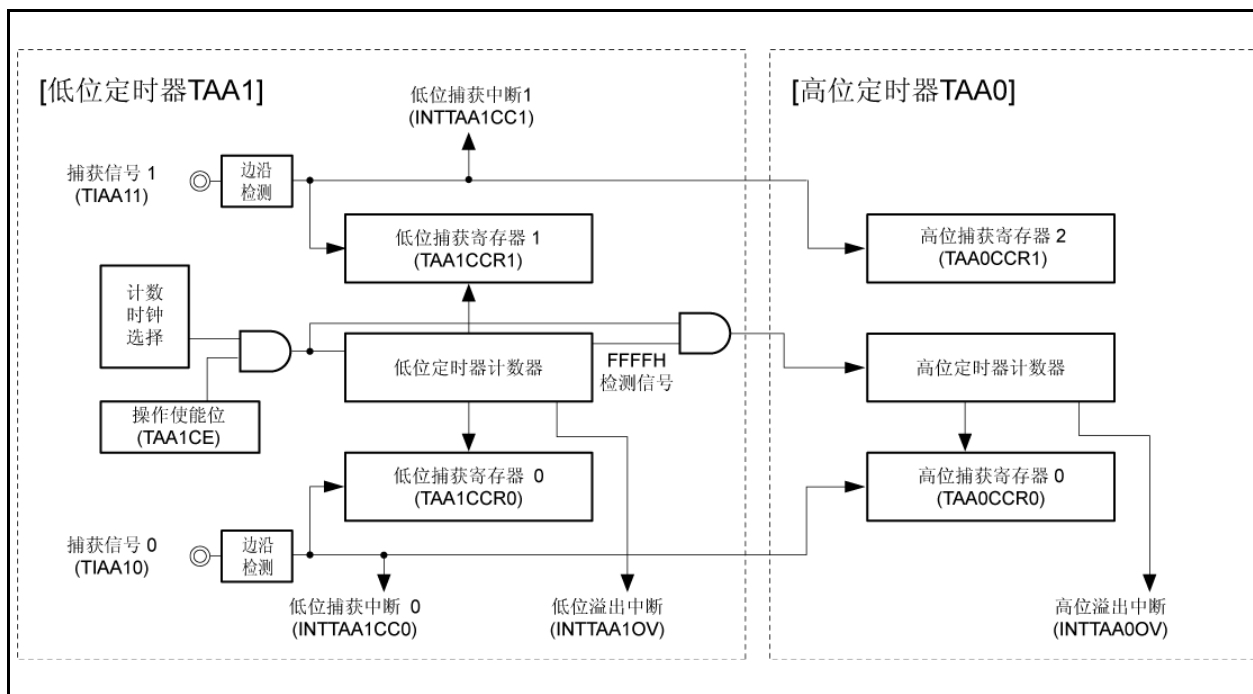
在级联中可以连接的 TAA 通道组合在下表中列出。

表 7-11. TAA 的级联

低位 定时器 (主定时器)	高位定时器 (从定时器)
TAA1	TAA0
TAA3	TAA2

在下面的示例中，TAA1 被用作低位定时器 (主定时器) 且 TAA0 被用作高位定时器 (从定时器)，通过级联将他们当作 32 位捕获定时器来使用。

图 7-50. 级联示例



当 TAA1 和 TAA0 在级联中连接使用，各个引脚和信号展示如下。

表 7-12. 级联中的状态

名称	高位/低位	功能	操作
TIAA10 引脚输入	低位	捕获输入 0	当检测到该输出的有效边沿时，低位定时器计数器的值被保存在 TAA1CCR0 寄存器中，且高位定时器计数器的值被保存在 TAAOCCR0 寄存器中。
TIAA11 引脚输入	低位	捕获输入 1	当检测到该输出的有效边沿时，低位定时器计数器的值被保存在 TAA1CCR1 寄存器中，且高位定时器计数器的值被保存在 TAAOCCR1 寄存器中。
INTTAA1CCR0 中断信号	低位	捕获中断 0	当检测到 TIAA10 引脚的有效边沿时，产生该中断。
INTTAA1CCR1 中断信号	低位	捕获中断 1	当检测到 TIAA11 引脚的有效边沿时，产生该中断。
INTTAA1OV 中断信号	低位	溢出中断	当检测到一个低位定时器计数器的溢出时，产生该中断。
TIAA00 引脚输入	高位	捕获输入 0	不操作
TIAA01 引脚输入	高位	捕获输入 1	不操作
INTTAA0CCR0 中断信号	高位	捕获中断 0	不操作
INTTAA0CCR1 中断信号	高位	捕获中断 1	不操作
INTTAA0OV 中断信号	高位	溢出中断	当检测到一个高位定时器计数器的溢出时，产生该中断。

图 7-51. TAA1 和 TAA0 级联的操作流程 (1/2)

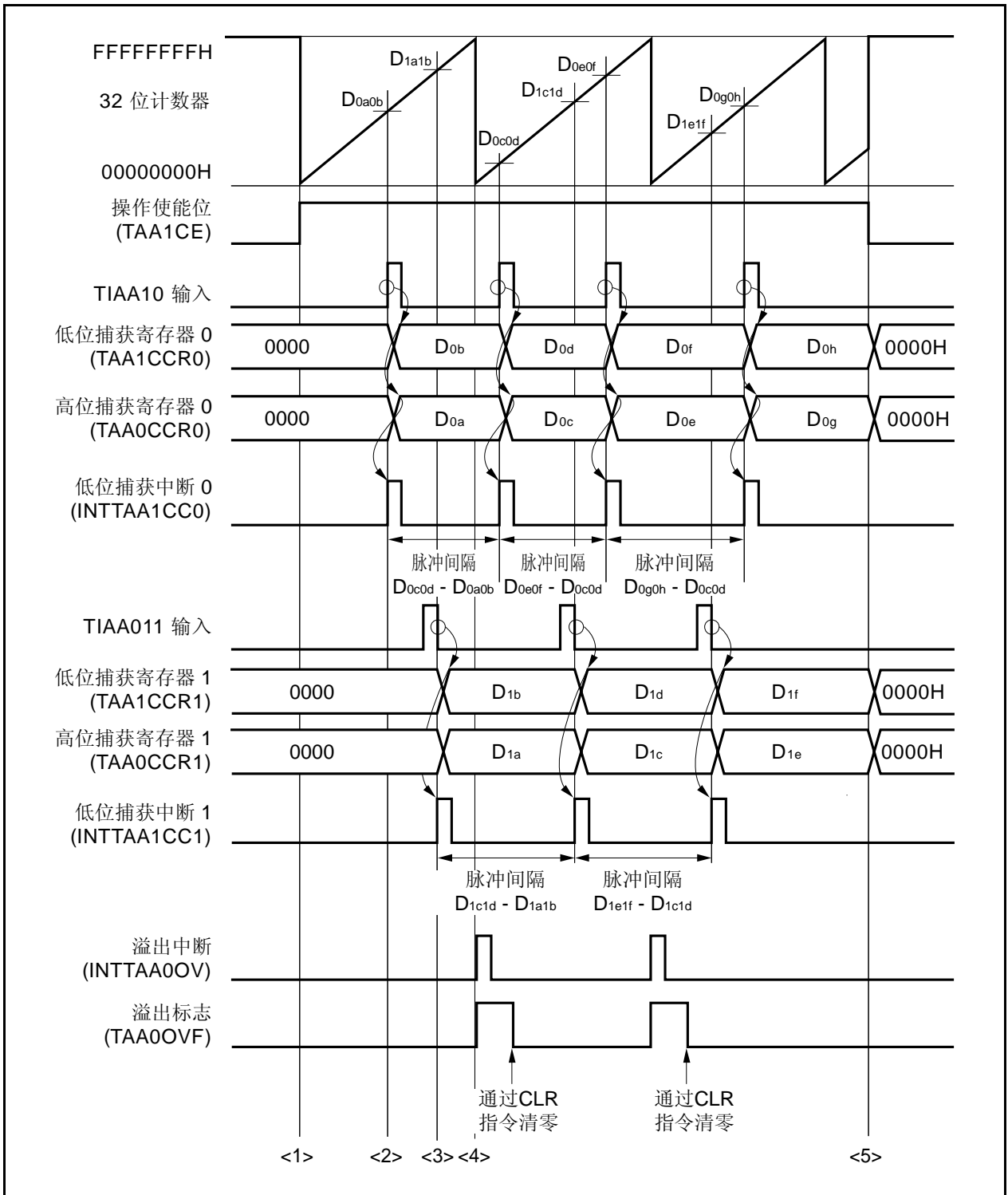


图 7-51. TAA1 和 TAA0 级联的操作流程 (2/2)

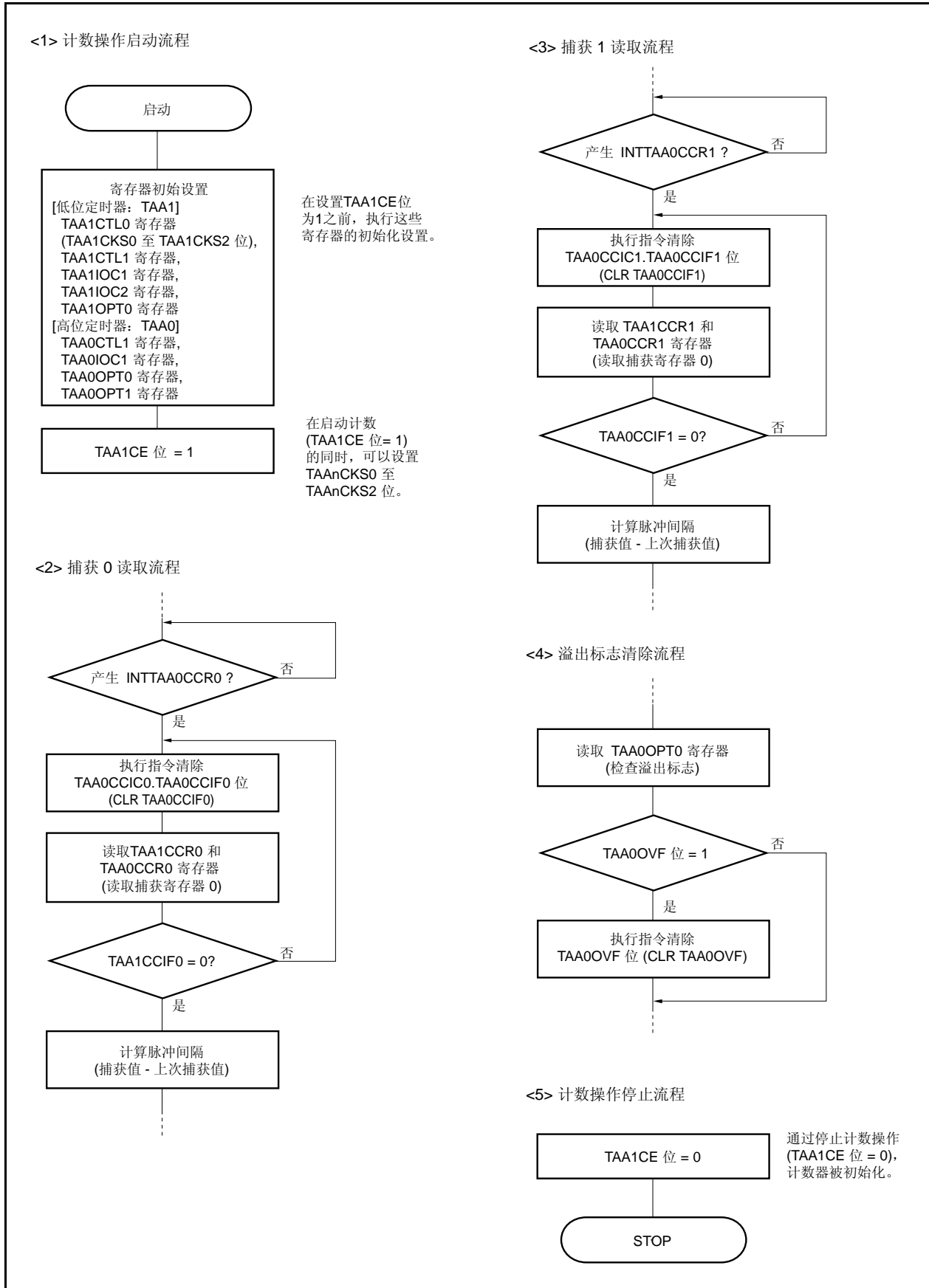
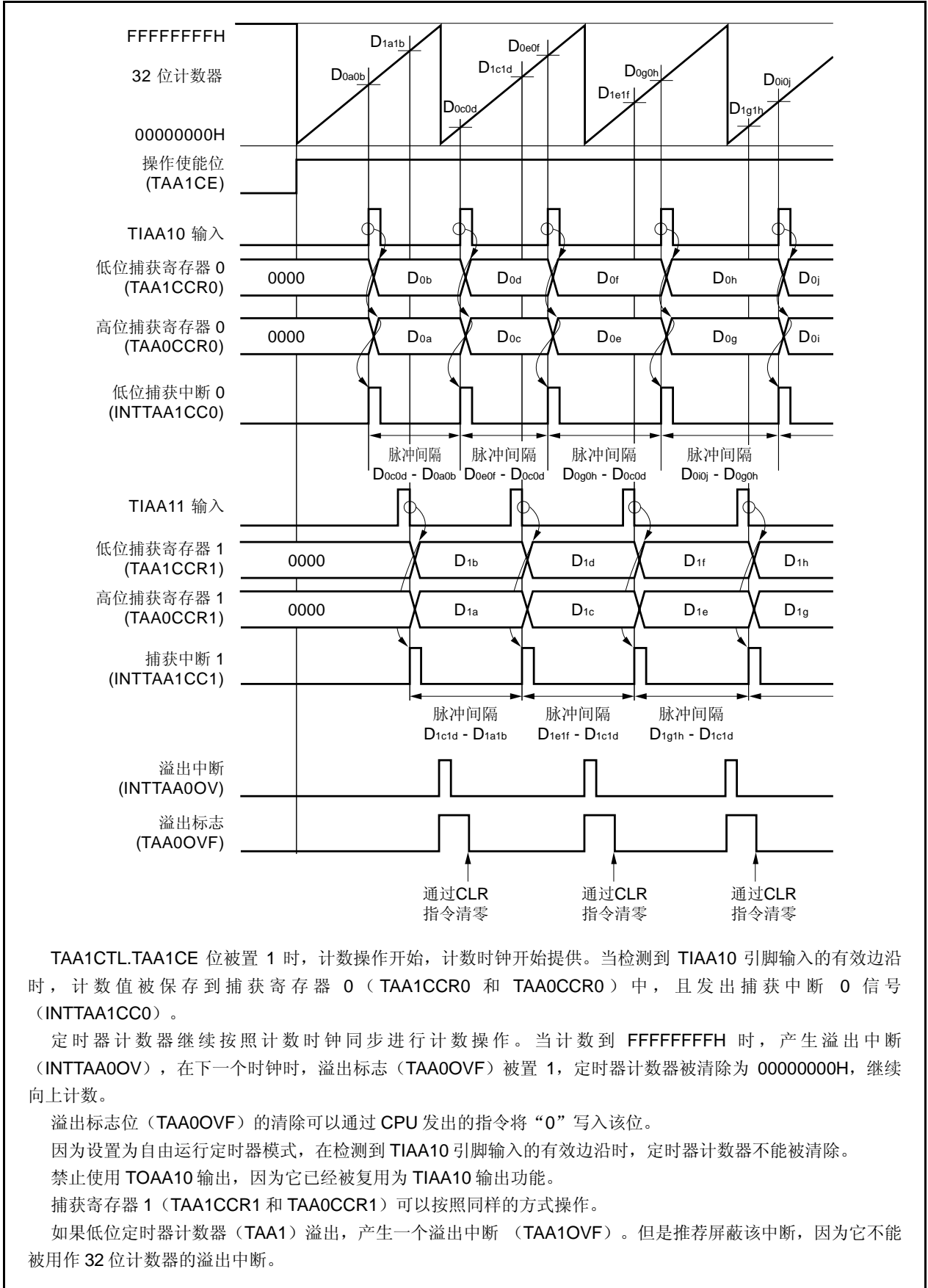


图 7-52. 当 TAA1 和 TAA0 级联时的基本时序示例



TAA1CTL.TAA1CE 位被置 1 时，计数操作开始，计数时钟开始提供。当检测到 TIAA10 引脚输入的有效边沿时，计数值被保存到捕获寄存器 0 (TAA1CCR0 和 TAA0CCR0) 中，且发出捕获中断 0 信号 (INTTAA1CC0)。

定时器计数器继续按照计数时钟同步进行计数操作。当计数到 FFFFFFFH 时，产生溢出中断 (INTTAA0OV)，在下一个时钟时，溢出标志 (TAA0OVF) 被置 1，定时器计数器被清除为 0000000H，继续向上计数。

溢出标志位 (TAA0OVF) 的清除可以通过 CPU 发出的指令将“0”写入该位。

因为设置为自由运行定时器模式，在检测到 TIAA10 引脚输入的有效边沿时，定时器计数器不能被清除。

禁止使用 TOAA10 输出，因为它已经被复用为 TIAA10 输出功能。

捕获寄存器 1 (TAA1CCR1 和 TAA0CCR1) 可以按照同样的方式操作。

如果低位定时器计数器 (TAA1) 溢出，产生一个溢出中断 (TAA1OVF)。但是推荐屏蔽该中断，因为它不能被用作 32 位计数器的溢出中断。

7.9 选择器功能

在 V850ES/JG3-U 和 V850ES/JH3-U 单片机中，端口或外设 I/O (TAA1, UARTC0 或 UARTC1) 的复用功能引脚可以被选为 TAA1 的捕获触发输入。

当使用 RXDCn 时，通过选择器功能选择 UARTCn 引脚输入的信号，可以计算 UARTCn 的 LIN 接收传输率和波特率误差。(n = 0, 1)

(1) 选择器操作控制寄存器 0 (SELCNT0)

SELCNT0 寄存器是 8 位寄存器，用于选择定时器 TAA1 的捕获触发。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF308H

	7	6	5	4	3	2	1	0
SELCNT0	0	0	0	ISEL4	ISEL3	0	0	0

ISEL4	TIAA11 捕获触发输入信号的选择
0	TIAA11 (复用功能为P35) 引脚
1	RXDC1 (复用功能为P91) 引脚

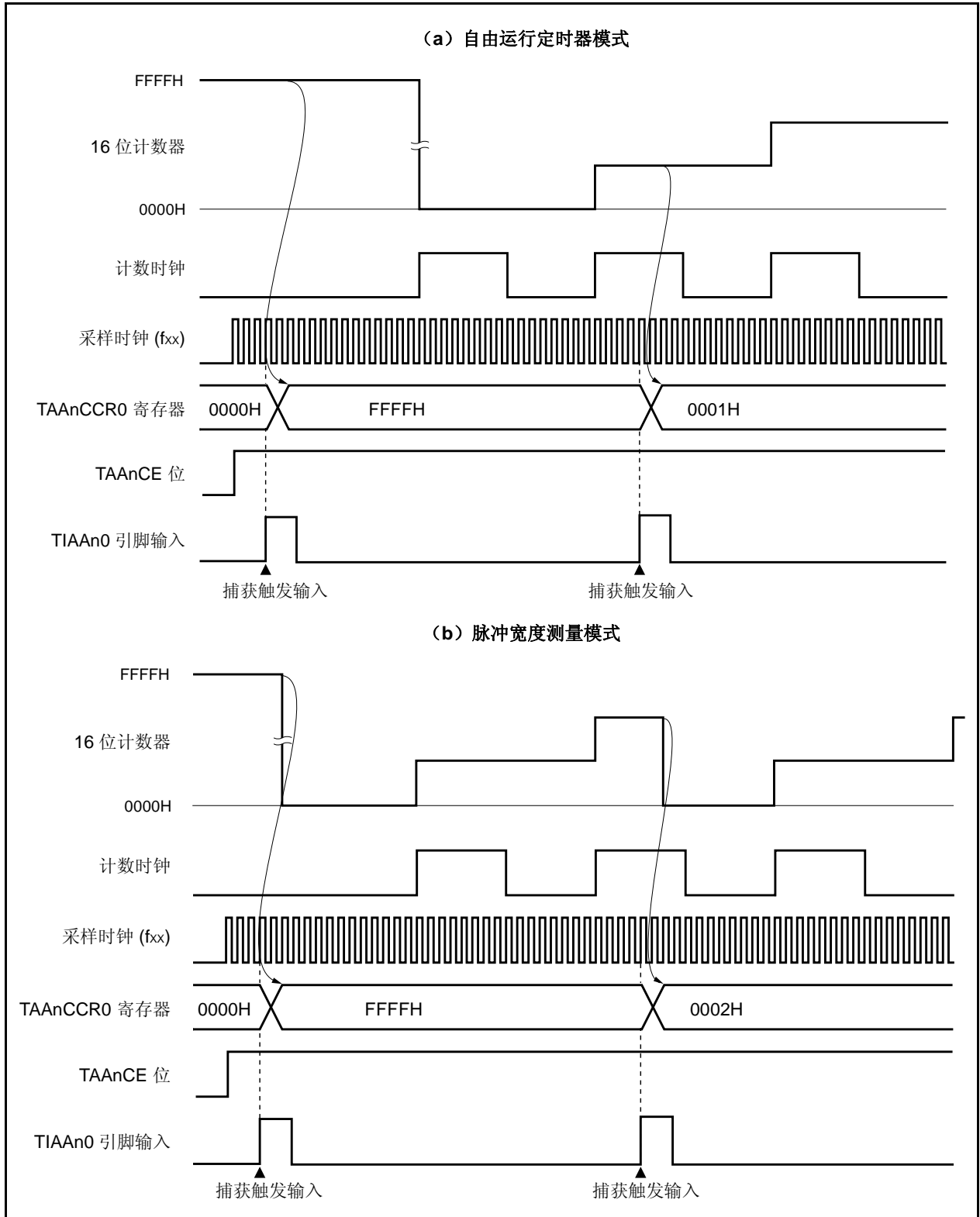
ISEL3	TIAA10 捕获触发输入信号的选择
0	TIAA10 (复用功能为P34) 引脚
1	RXDC0 (复用功能为P31) 引脚

- 注意事项**
1. 要设置 ISEL3 和 ISEL4 位为 1，设置相应功能引脚为捕获输入模式。
 2. 当 TAA1, UARTC0 或 UARTC1 的操作停止时，设置 ISEL3 和 ISEL4 位。
 3. 请确保将第 7 位至第 5 位，以及第 2 位至第 0 位清为“0”。

7.10 注意事项

(1) 捕获操作

当使用捕获操作，并且选择低速时钟作为计数时钟时，如果在 TAAAnCE 位置 1 之后立即输入捕获触发，那么捕获到 TAAAnCCR0 和 TAAAnCCR1 寄存器中的可能是 FFFFH 而不是 0000H。



第八章 16 位定时器/事件计数器AB (TAB)

定时器 AB (TAB) 是 16 位定时器/事件计数器。

V850ES/JG3-U 和 V850ES/JH3-U 单片机内置 TAB0 和 TAB1。

8.1 概述

定时器 TABn 的概况如下所示：

- 时钟选择: 8 路
- 捕获/触发输入引脚: 4
- 外部事件计数输入引脚: 1
- 外部触发输入引脚: 1
- 定时器/计数器: 1
- 捕获/比较寄存器: 4
- 捕获/比较匹配中断请求信号: 4
- 定时器输出引脚: 4

备注 n = 0, 1

8.2 功能

定时器 TABn 具有下列功能：

- 间隔定时器
- 外部事件计数器
- 外部触发脉冲输出
- 单次触发脉冲输出
- PWM 输出
- 自由运行定时器
- 脉冲宽度测量
- 三角波 PWM 输出
- 定时器调谐操作功能
- 同时启动功能

备注 n = 0, 1

8.3 配置

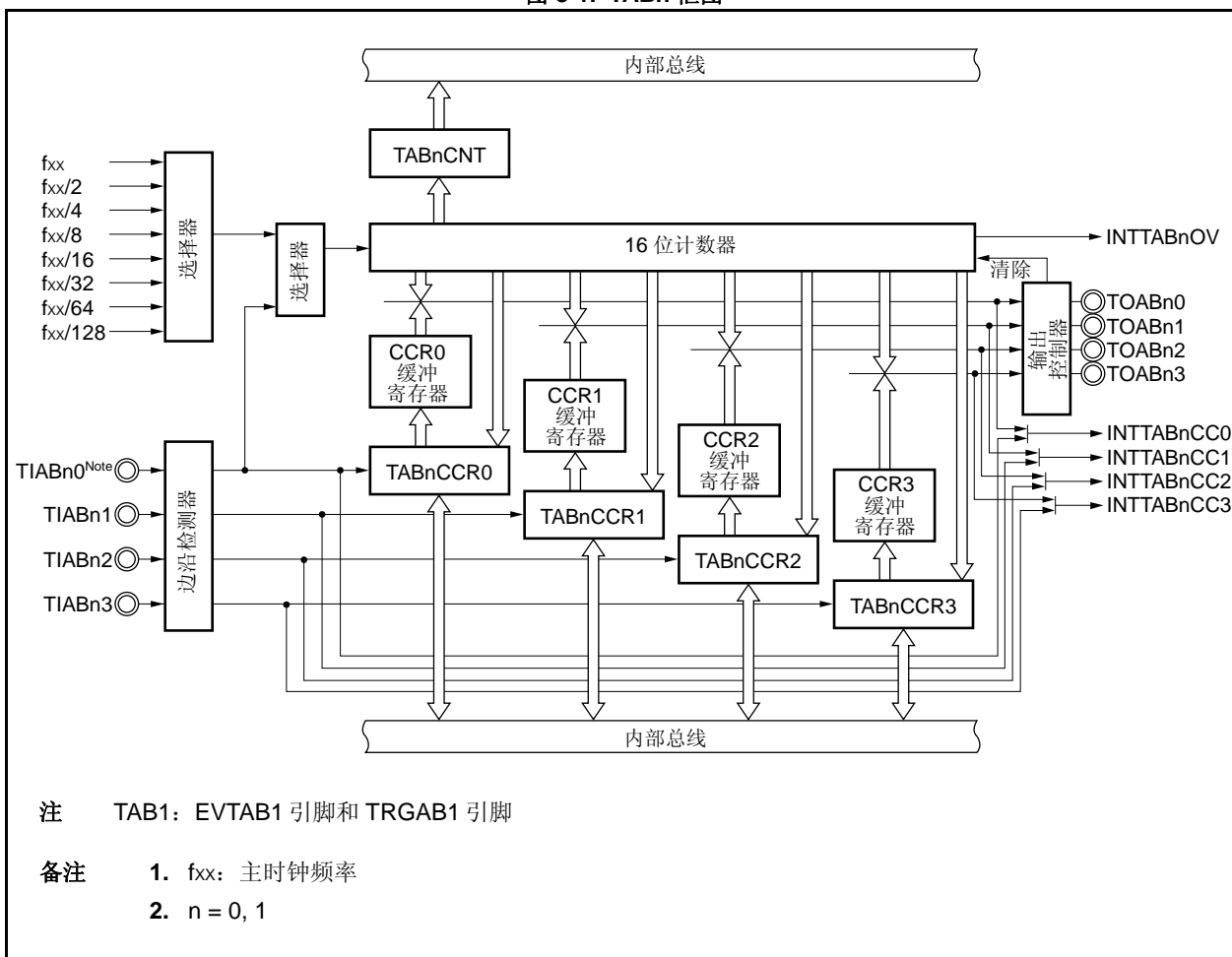
TABn 包括下列硬件：

表 8-1. TABn 的配置

项目	配置
寄存器	16 位计数器 TABn 捕获/比较寄存器 0 至 3 (TABnCCR0 至 TABnCCR3) TABn 计数器读取缓冲寄存器 (TABnCNT) CCR0 至 CCR3 缓冲寄存器 TABn 控制寄存器 0, 1 (TABnCTL0, TABnCTL1) TABn 输入/输出控制寄存器 0 至 2, 4 (TABnIOC0 至 TABnIOC2, TABnIOC4) TABn 选项寄存器 0 (TABnOPT0)
定时器输入 ^{注 2}	4 (TIABn0 ^{注 1} 至 TIABn3 引脚)
定时器输出 ^{注 2}	4 (TOABn0 至 TOABn3 引脚)

- 注 1. 当使用 TIABn0 至 TIABn3 和 TOABn0 至 TOABn3 引脚功能时，参见表 4-20 端口引脚用作复用功能时的设置。
2. TIAB00 引脚功能可以复用为捕获触发、外部事件计数以及外部触发的信号输入引脚。

图 8-1. TABn 框图



(1) 16 位计数器

此 16 位计数器可以计数内部时钟或外部事件。

该计数器的计数值可以通过读取 TABnCNT 寄存器得到。

当 TABnCTL0.TABnCE 位 = 0 时, 16 位计数器的值为 FFFFH。如果此时读取 TABnCNT 寄存器, 则读回的值为 0000H。

系统复位将 TABnCE 位清 0, 因此, 16 位计数器被设置为 FFFFH。

(2) CCR0 缓冲寄存器

CCR0 缓冲寄存器是一个 16 位比较寄存器, 用于比较 16 位计数器的计数值。

当 TABnCCR0 寄存器用作比较寄存器时, 写入 TABnCCR0 寄存器的值会传送到 CCR0 缓冲寄存器。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值相匹配时, 就会产生一个比较匹配中断请求信号 (INTTABnCC0)。

CCR0 缓冲寄存器不能直接读写。

CCR0 缓冲寄存器在复位后清零为 0000H, TABnCCR0 寄存器清零为 0000H。

(3) CCR1 缓冲寄存器

CCR1 缓冲寄存器是一个 16 位比较寄存器, 用于比较 16 位计数器的计数值。

当 TABnCCR1 寄存器用作比较寄存器时, 写入 TABnCCR1 寄存器的值会传送到 CCR1 缓冲寄存器。当 16 位计数器的计数值和 CCR1 缓冲寄存器的值匹配时, 就会产生一个比较匹配中断请求信号 (INTTABnCC1)。

CCR1 缓冲寄存器不能直接读写。

CCR1 缓冲寄存器在复位后清零为 0000H, TABnCCR1 寄存器清零为 0000H。

(4) CCR2 缓冲寄存器

CCR2 缓冲寄存器是一个 16 位比较寄存器, 用于比较 16 位计数器的计数值。

当 TABnCCR2 寄存器用作比较寄存器时, 写入 TABnCCR2 寄存器的值会传送到 CCR2 缓冲寄存器。当 16 位计数器的计数值和 CCR2 缓冲寄存器的值匹配时, 就会产生一个比较匹配中断请求信号 (INTTABnCC2)。

CCR2 缓冲寄存器不能直接读写。

CCR2 缓冲寄存器在复位后清零为 0000H, TABnCCR2 寄存器清零为 0000H。

(5) CCR3 缓冲寄存器

CCR3 缓冲寄存器是一个 16 位比较寄存器, 用于比较 16 位计数器的计数值。

当 TABnCCR3 寄存器用作比较寄存器时, 写入 TABnCCR3 寄存器的值会传送到 CCR3 缓冲寄存器。当 16 位计数器的计数值与 CCR3 缓冲寄存器的值相匹配时, 就会产生一个比较匹配中断请求信号 (INTTABnCC3)。

CCR3 缓冲寄存器不能直接读写。

CCR3 缓冲寄存器在复位后清零为 0000H, TABnCCR3 寄存器清零为 0000H。

(6) 边沿检测器

该电路检测输入到 TIABn0 至 TIABn3 引脚信号的有效边沿。通过设置 TABnIOC1 和 TABnIOC2 寄存器, 可以选择各种有效沿: 无边沿、上升沿、下降沿、上升/下降双边沿作为有效沿。

(7) 输出控制器

该电路控制 TOABn0 至 TOABn3 引脚的输出。输出控制器由 TABnIOC0 寄存器控制。

(8) 选择器

此选择器选择 16 位计数器的计数时钟。可选择 8 种内部时钟或一个外部事件作为计数时钟。

8.4 寄存器

以下寄存器控制 TABn :

- TABn 控制寄存器 0 (TABnCTL0)
- TABn 控制寄存器 1 (TABnCTL1)
- TABn 输入/输出控制寄存器 0 (TABnIOC0)
- TABn 输入/输出控制寄存器 1 (TABnIOC1)
- TABn 输入/输出控制寄存器 2 (TABnIOC2)
- TABn 输入/输出控制寄存器 4 (TABnIOC4)
- TABn 选项寄存器 0 (TABnOPT0)
- TABn 捕获/比较寄存器 0 (TABnCCR0)
- TABn 捕获/比较寄存器 1 (TABnCCR1)
- TABn 捕获/比较寄存器 2 (TABnCCR2)
- TABn 捕获/比较寄存器 3 (TABnCCR3)
- TABn 计数器读取缓冲寄存器 (TABnCNT)

备注 1. 当使用 TIABn0 至 TIABn3 和 TOABn0 至 TOABn3 引脚的功能时, 参见表 4-20 端口引脚用作复用功能时的设置。
2. n = 0, 1

(1) TABn 控制寄存器 0 (TABnCTL0)

TABnCTL0 寄存器是 8 位寄存器，用于控制定时器 TABn 的操作。

该寄存器可以按字节读写或按位读写。

复位输入将该寄存器设为 00H。

随时可以用软件将相同的值写入 TABnCTL0 寄存器。

复位后: 00H R/W 地址: TAB0CTL0 FFFFF540H, TAB1CTL0 FFFFF560H

	7	6	5	4	3	2	1	0
TABnCTL0 (n = 0, 1)	TABnCE	0	0	0	0	TABnCKS2	TABnCKS1	TABnCKS0

TABnCE	TABn 操作控制
0	TABn操作禁止 (TABn 异步复位 ^注)
1	TABn操作使能, TABn操作开始。

TABnCKS2	TABnCKS1	TABnCKS0	内部计数时钟选择
0	0	0	f _{xx}
0	0	1	f _{xx} /2
0	1	0	f _{xx} /4
0	1	1	f _{xx} /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

注 TABnOPT0.TABnOVF 位, 16 位计数器、定时器输出 (TOABn0 至 TOABn3 引脚)。

- 注意事项
1. 当 TABnCE 位=0 时, 设置 TABnCKS2 至 TABnCKS0 位为“1”。
当 TABnCE 位的值从 0 变为 1 时, 可以同时设定 TABnCKS2 至 TABnCKS0 位。
 2. 请确保将第 3 位至第 6 位清为“0”。

备注 f_{xx}: 主时钟频率

(2) TABn 控制寄存器 1 (TABnCTL1)

TABnCTL1 寄存器是 8 位寄存器，用于控制定时器 TABn 的操作。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后：00H R/W 地址：TAB0CTL1 FFFFF541H, TAB1CTL1 FFFFF561H

	7	6	5	4	3	2	1	0
TABnCTL1 (n = 0, 1)	0	TABnEST	TABnEEE	0	0	TABnMD2	TABnMD1	TABnMD0

TABnEST	软件触发控制
0	-
1	为外部触发输入产生一个有效信号 <ul style="list-style-type: none"> 在单次触发脉冲输出模式下： <ul style="list-style-type: none"> 向TABnEST位写入1作为触发信号，输出一个单次触发脉冲。 在外部触发脉冲输出模式下： <ul style="list-style-type: none"> 向TABnEST位写入1作为触发信号，输出一个PWM波形。

TABnEEE	计数时钟选择
0	禁止外部事件计数输入的操作 (执行对计数时钟的计数，该时钟由 TABnCTL0.TABnCK0 至 TABnCK2位选择)
1	使能外部事件计数输入的操作 (对外部事件计数输入信号的有效边沿执行计数)

通过TABnEEE位选择使用内部计数时钟计数，还是使用外部事件计数输入的有效边沿来计数。

TABnMD2	TABnMD1	TABnMD0	定时器模式选择
0	0	0	间隔定时器模式
0	0	1	外部事件计数模式
0	1	0	外部触发脉冲输出模式
0	1	1	单次触发脉冲输出模式
1	0	0	PWM 输出模式
1	0	1	自由运行定时器模式
1	1	0	脉宽测量模式
1	1	1	三角波PWM 模式

- 注意事项**
1. 只有在外触发脉冲输出模式或单次触发脉冲输出模式下，TABnEST 位有效。在其它模式下，向该位写入 1 的操作都被忽略。
 2. 请确保将第 3 位，第 4 位和第 7 位清除为“0”。
 3. 不论 TABnEEE 位的取值如何，在外部事件计数模式下，都会选择外部事件输入。
 4. TABnCTL0.TABnCE 位 = 0 时，设置 TABnEEE 和 TABnMD2 至 TABnMD0 位 (TABnCE 位 = 1 时，可以写入相同的值)。在 TABnCE 位 = 1 时，重写操作是无法保证的，若重写发生错误，应把 TABnCE 位清为 0，再重新对这些位进行设定。

(3) TABn 输入/输出控制寄存器 0 (TABnIOC0)

TABnIOC0 寄存器是 8 位寄存器，用于控制定时器的输出 (TOABn0 至 TOABn3 引脚)。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: TAB0IOC0 FFFFF542H, TAB1IOC0 FFFFF562H

	7	6	5	4	3	2	1	0
TABnIOC0 (n = 0, 1)	TABnOL3	TABnOE3	TABnOL2	TABnOE2	TABnOL1	TABnOE1	TABnOL0	TABnOE0

TABnOLm	TOABnm 引脚输出电平设置 (m = 0 至 3) ^注
0	TOABnm 引脚输出从高电平开始
1	TOABnm 引脚输出从低电平开始

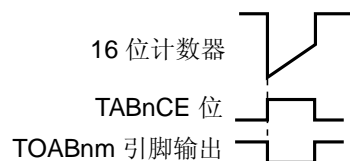
TABnOEm	TOABnm 引脚输出设置 (m = 0 to 3)
0	定时器输出禁止 • 当TABnOLm位 = 0: 从TOABnm引脚输出低电平 • 当TABnOLm位 = 1: 从TOABnm引脚输出高电平
1	定时器输出使能 (从TOABnm引脚输出一个方波)

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

• 当TABnOLm 位= 0



• 当TABnOLm 位= 1



注意事项 1. 当 TABnCTL0.TABnCE 位 = 0 时，重新写入 TABnOLm 和 TABnOEm 位 (当 TABnCE 位 = 1 时，可以写入相同值)。如果重新写入不能正确执行，将 TABnCE 位清为 0，然后重新设定该位。

2. 即使在 TABnCE 和 TABnOEm 位均为 0 时对 TABnOLm 位进行操作，TOABnm 引脚的输出电平也会改变。

备注 m=0 至 3

(4) TABn 输入/输出控制寄存器 1 (TABnIOC1)

TABnIOC1 寄存器是 8 位寄存器，用于控制捕获触发输入信号 (TIABn0 至 TIABn3 引脚) 的有效边沿。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: TAB0IOC1 FFFFF543H, TAB1IOC1 FFFFF563H

	7	6	5	4	3	2	1	0
TABnIOC1 (n = 0, 1)	TABnIS7	TABnIS6	TABnIS5	TABnIS4	TABnIS3	TABnIS2	TABnIS1	TABnIS0

TABnIS7	TABnIS6	捕获触发输入信号 (TIABn3 引脚) 有效边沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双沿

TABnIS5	TABnIS4	捕获触发输入信号 (TIABn2 引脚) 有效边沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双沿

TABnIS3	TABnIS2	捕获触发输入信号 (TIABn1 引脚) 有效边沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双沿

TABnIS1	TABnIS0	捕获触发输入信号 (TIABn0 引脚) 有效边沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双沿

- 注意事项**
1. 当 **TABnCTL0.TABnCE** 位 = 0 时，重写 **TABnIS7** 至 **TABnIS0** 位 (当 **TABnCE** 位=1 时，可以写入相同值)。如果重写不能正确执行，将 **TABnCE** 位清为 0，然后重新进行位的设定。
 2. 只有在自由运行定时器模式和脉冲宽度测量模式下，**TABnIS7** 至 **TABnIS0** 位才有效。在所有其它模式下，无法进行捕获操作。

(5) TABn 输入/输出控制寄存器 2 (TABnIOC2)

TABnIOC2 寄存器是 8 位寄存器，用于控制外部事件计数器输入信号 (TIAB00/EVTAB1 引脚) 和外部触发输入信号 (TIAB00/TRGAB1) 的有效边沿。

该寄存器可按字节字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: TAB0IOC2 FFFFF544H, TAB1IOC2 FFFFF564H

	7	6	5	4	3	2	1	0
TABnIOC2 (n = 0, 1)	0	0	0	0	TABnEES1	TABnEES0	TABnETS1	TABnETS0

TABnEES1	TABnEES0	外部事件计数输入信号(TIAB00/EVTAB1 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	检测双沿

TABnETS1	TABnETS0	外部触发输入信号 (TIAB00/TRGAB1 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	检测双沿

- 注意事项**
1. 当 TABnCTL0.TABnCE 位 = 0 时，重新写入 TABnEES1, TABnEES0, TABnETS1 和 TABnETS0 位 (当 TABnCE 位 = 1, 可以写入相同的值)。如果重新写入不能正确执行，将 TABnCE 位清为 0，然后重新进行位的设定。
 2. 只有当 TABnCTL1.TABnEEE 位 = 1 或设置为外部事件计数模式 (TABnCTL1.TABnMD2 至 TABnCTL1.TABnMD0 = 001) 时，TABnEES1 和 TABnEES0 位有效。
 3. 只有在外外部触发脉冲输出模式 (TABnCTL1.TABnMD2 至 TABnCTL1.TABnMD0 位 = 010) 或在单次触发脉冲输出模式 (TABnCTL1.TABnMD2 至 TABnCTL1.TABnMD0 位 = 011) 时，TABnETS1 和 TABnETS0 位有效。

(6) TABn 输入/输出控制寄存器 4 (TABnIOC4)

TABnIOC4 寄存器是 8 位寄存器，用于控制定时器输出。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。定时器停止操作(TABnCTL0.TABnCE = 0)，该寄存器并不会被重置。

注意事项 1. 在下列状态时禁止访问 TABnIOC4 寄存器。详情参见 3.4.8 (2) 访问专用片上外设 I/O 寄存器。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
- 当 CPU 运行于内部振荡时钟工作时。

2. 只有在间隔定时器模式和自由运行定时器模式下，可以设置 TABnIOC4 寄存器。请确保在其它所有模式下设置 TABnIOC4 寄存器为 00H (模式设置的详情参见 8.4 (2) TABn 控制寄存器 1 (TABnCTL1))。即使在自由运行定时器模式下，如果 TABnCCR0 至 TABnCCR3 寄存器被设置为捕获功能，TABnIOC4 寄存器的设置变为无效。

复位后: 00H R/W 地址: TAB0IOC4 FFFFF550H, TAB1IOC4 FFFFF570H

	7	6	5	4	3	2	1	0
TABnIOC4 (n = 0, 1)	TABnOS3	TABnOR3	TABnOS2	TABnOR2	TABnOS1	TABnOR1	TABnOS0	TABnOR0
	TABnOSm	TABnORm	TIABnm 引脚的切换控制 (m = 0 至 3)					
	0	0	无请求。正常切换操作。					
	0	1	重置请求 在 16 位计数器的值和 TABnCCRm 寄存器的值再次匹配之前，固定为无效电平。					
	1	0	设置请求 在 16 位计数器的值和 TABnCCRm 寄存器的值再次匹配之前，固定为有效电平。					
	1	1	保持请求 保持当前输出电平					

(7) TABn 选项寄存器 0 (TABnOPT0)

TABnOPT0 寄存器为 8 位寄存器，用来设置捕获/比较操作和检测计数器溢出。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: TAB0OPT0 FFFFF545H, TAB1OPT0 FFFFF565H

	7	6	5	4	3	2	1	0
TABnOPT0 (n = 0, 1)	TABnCCS3	TABnCCS2	TABnCCS1	TABnCCS0	0	TAB1CMS ^注	TABnCUF	TABnOVF

TABnCCSm	TABnCCRm 寄存器捕获/比较选择
0	选择比较寄存器
1	选择捕获寄存器
只有在自由运行定时器模式下，TABnCCSm位的设置有效。	

TABnOVF	TABn 溢出检测标志
置位 (1)	发生溢出
复位 (0)	TABnOVF 位写入 0，或TABnCTL0.TABnCE 位 = 0
<ul style="list-style-type: none"> 在自由运行定时器模式或脉宽测量模式下，当16 位计数器的计数值从FFFFH 溢出到 0000H时，TABnOVF位被置为 1。 TABnOVF 位被置为 1 的同时，产生一个中断请求信号 (INTTABnOV)。在自由运行定时器模式和脉宽测量模式之外的模式下，不产生INTTABnOV 信号。 当TABnOVF位 = 1时，即使TABnOVF位或TABnOPT0 寄存器被读取，TABnOVF位也不会被清除。 TABnOVF位可读可写，但是不能通过软件将TABnOVF位位置1。在TABn工作期间写入1 没有影响。 	

注 TAB1CMS 位被用于电机控制功能。详情参见第 11 章 电机控制功能。

- 注意事项
1. 当 TABnCTL0.TABnCE 位= 0 时，重新写入 TABnCCS3 至 TABnCCS0 位（当 TABnCE 位= 1 时，可以写入相同的值）。如果重新写入不能正确执行，将 TABnCE 位清为 0，然后重新进行位的设定。
 2. 请确保将第 3 位清为“0”。不使用电机控制功能时，请确保将第 2 位清除为“0”。

备注 m=0 至 3

(8) TABn 捕获/比较寄存器 0 (TABnCCR0)

根据不同工作模式，TABnCCR0 寄存器可以用作捕获寄存器或比较寄存器。

在自由计时器模式下，根据 TABnOPT0.TABnCCS0 位的设定，此寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TABnCCR0 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

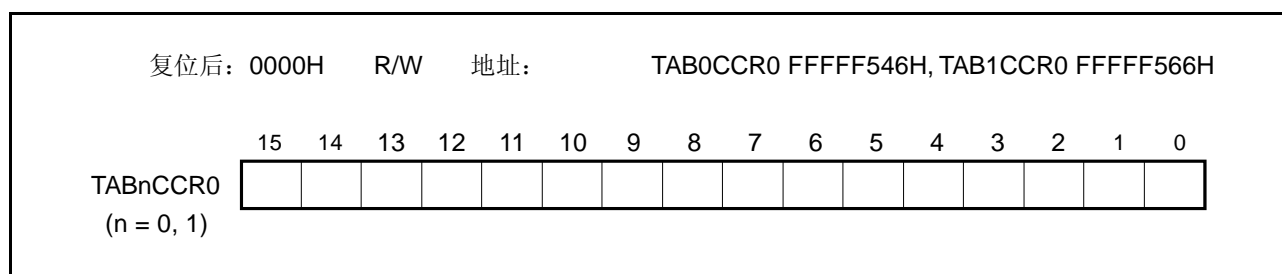
TABnCCR0 寄存器可以在运行中进行读取和写入。

此寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

注意事项 在以下状态禁止访问 TABnCCR0 寄存器。详情参见 3.4.8 (2) 访问片上外设输入/输出专用寄存器。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
- 当 CPU 运行于内部振荡时钟工作时。



(a) 用作比较寄存器时的功能

即便在 TABnCTL0.TABnCE 位=1 时, TABnCCR0 寄存器仍可以重新写入。

TABnCCR0 寄存器的设定值传送给 CCR0 缓冲寄存器。当 16 位计数器的值与 CCR0 缓冲寄存器的值相匹配时, 就会生成一个比较匹配中断请求信号 (INTTABnCC0)。如果此时 TOABn0 引脚输出允许, 则 TOABn0 引脚的输出会反相。

在间隔定时器模式、外部事件计数器模式、外部触发脉冲输出模式、单次触发脉冲输出模式、PWM 输出模式或三角波 PWM 模式时, 当 TABnCCR0 寄存器作为周期寄存器时, 如果 16 位计数器的计数值与 CCR0 缓冲寄存器数值匹配, 则计数器清零为 (0000H)。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下, 当 TABnCCR0 寄存器用作捕获寄存器时, 若检测到捕获触发输入引脚 (TIABn0 引脚) 的有效边沿, 则 16 位计数器的当前计数值存储于 TABnCCR0 寄存器中。在脉宽测量模式下, 若检测到捕获触发输入引脚 (TIABn0 引脚) 的有效边沿, 16 位计数器的当前计数值就储存在 TABnCCR0 寄存器中, 且 16 位计数器清零 (0000H)。

即使捕获操作和读取 TABnCCR0 寄存器的操作发生冲突, 还是可以正确读取 TABnCCR0 寄存器的值。

备注 n = 0, 1

下表显示了捕获/比较寄存器在各工作模式下的功能, 以及如何向比较寄存器写入数据。

表 8-2. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

操作模式	捕获/比较寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	随时写入
外部事件计数器	比较寄存器	随时写入
外部触发脉冲输出	比较寄存器	批量写入
单次触发脉冲输出	比较寄存器	随时写入
PWM (脉宽调制信号) 输出	比较寄存器	批量写入
自由运行定时器	捕获/比较寄存器	随时写入
脉冲宽度测量	捕获寄存器	-
三角波 PWM 模式	比较寄存器	批量写入

(9) TABn 捕获/比较寄存器 1 (TABnCCR1)

根据不同工作模式，TABnCCR1 寄存器可以用作捕获寄存器或比较寄存器。

在自由计时器模式下，根据 TABnOPT0.TABnCCS1 位的设定，此寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TABnCCR1 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

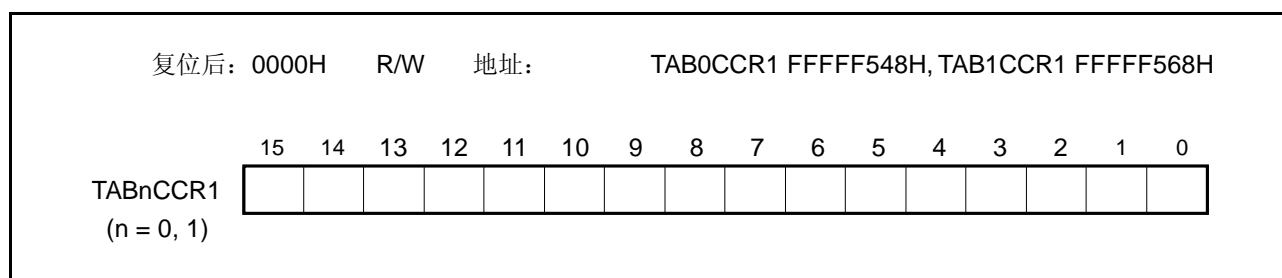
TABnCCR1 寄存器可以在运行中进行读取和写入。

此寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

注意事项 在以下状态禁止访问 TABnCCR1 寄存器。详情参见 3.4.8 (2) 访问片上外设输入/输出专用寄存器。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
- 当 CPU 运行于内部振荡时钟工作时。



(a) 用作比较寄存器时的功能

即便在 TABnCTL0.TABnCE 位= 1 时, TABnCCR1 寄存器仍可以重新写入。

TABnCCR1 寄存器的设定值传送给 CCR1 缓冲寄存器。当 16 位计数器的值与 CCR1 缓冲寄存器的值相匹配时, 就会生成一个比较匹配中断请求信号 (INTTABnCC1)。如果此时 TOABn1 引脚输出允许, 则 TOABn1 引脚的输出会反相。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下, 当 TABnCCR1 寄存器用作捕获寄存器时, 若检测到捕获触发输入引脚 (TIABn1 引脚) 的有效边沿, 则 16 位计数器的当前计数值存储于 TABnCCR1 寄存器中。在脉宽测量模式下, 若检测到捕获触发输入引脚 (TIABn1 引脚) 的有效边沿, 16 位计数器的当前计数值就储存在 TABnCCR1 寄存器中, 且 16 位计数器清零 (0000H)。

即使捕获操作和读取 TABnCCR1 寄存器的操作发生冲突, 还是可以正确读取 TABnCCR1 寄存器的值。

备注 n = 0, 1

下表显示了捕获/比较寄存器在各工作模式下的功能, 以及如何向比较寄存器写入数据。

表 8-3. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

操作模式	捕获/比较寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	随时写入
外部事件计数器	比较寄存器	随时写入
外部触发脉冲输出	比较寄存器	批量写入
单次触发脉冲输出	比较寄存器	随时写入
PWM (脉宽调制信号) 输出	比较寄存器	批量写入
自由运行定时器	捕获/比较寄存器	随时写入
脉冲宽度测量	捕获寄存器	-
三角波 PWM 模式	比较寄存器	批量写入

(10) TABn 捕获/比较寄存器 2 (TABnCCR2)

根据不同工作模式，TABnCCR2 寄存器可以用作捕获寄存器或比较寄存器。

在自由计时器模式下，根据 TABnOPT0.TABnCCS2 位的设定，此寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TABnCCR2 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

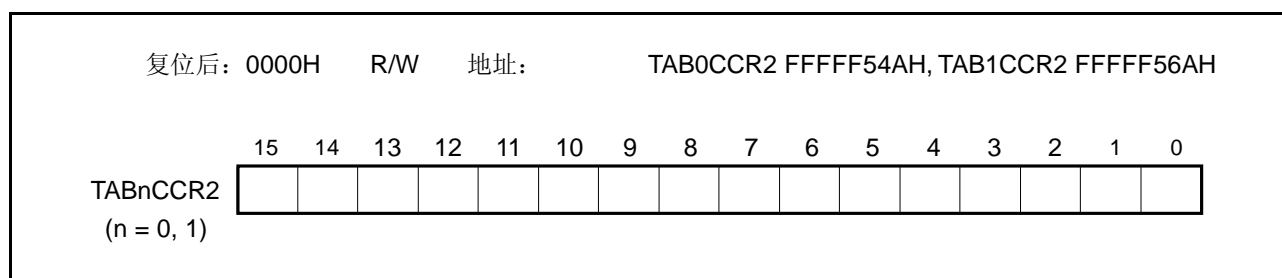
TABnCCR2 寄存器可以在运行中进行读取和写入。

此寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

注意事项 在以下状态禁止访问 TABnCCR2 寄存器。详情参见 3.4.8 (2) 访问片上外设输入/输出专用寄存器。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
- 当 CPU 运行于内部振荡时钟工作时。



(a) 用作比较寄存器时的功能

即便在 TABnCTL0.TABnCE 位= 1 时, TABnCCR2 寄存器仍可以重新写入。

TABnCCR2 寄存器的设定值传送给 CCR2 缓冲寄存器。当 16 位计数器的计数值和 CCR2 缓冲寄存器的值匹配时, 也会产生一个比较匹配中断请求信号 (INTTABnCC2)。如果此时 TOABn2 引脚输出允许, 则 TOABn2 引脚的输出会反相。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下, 当 TABnCCR2 寄存器用作捕获寄存器时, 若检测到捕获触发输入引脚 (TIABn2 引脚) 的有效边沿, 则 16 位计数器的当前计数值存储于 TABnCCR2 寄存器中。在脉宽测量模式下, 若检测到捕获触发输入引脚 (TIABn2 引脚) 的有效边沿, 16 位计数器的当前计数值就储存在 TABnCCR2 寄存器中, 且 16 位计数器清零 (0000H)。

即使捕获操作和读取 TABnCCR2 寄存器的操作发生冲突, 还是可以正确读取 TABnCCR2 寄存器的值。

备注 n = 0, 1

下表显示了捕获/比较寄存器在各工作模式下的功能, 以及如何向比较寄存器写入数据。

表 8-4. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

操作模式	捕获/比较寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	随时写入
外部事件计数器	比较寄存器	随时写入
外部触发脉冲输出	比较寄存器	批量写入
单次触发脉冲输出	比较寄存器	随时写入
PWM (脉宽调制信号) 输出	比较寄存器	批量写入
自由运行定时器	捕获/比较寄存器	随时写入
脉冲宽度测量	捕获寄存器	-
三角波 PWM 模式	比较寄存器	批量写入

(11) TABn 捕获/比较寄存器 3 (TABnCCR3)

根据不同工作模式，TABnCCR3 寄存器可以用作捕获寄存器或比较寄存器。

在自由计时器模式下，根据 TABnOPT0.TABnCCS3 位的设定，此寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TABnCCR3 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

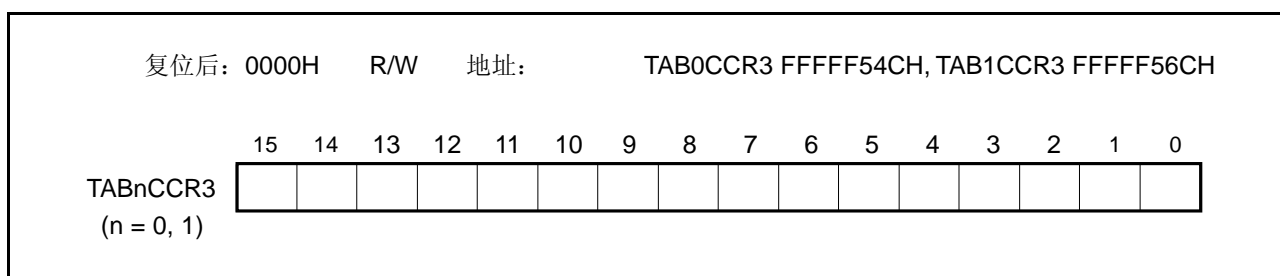
TABnCCR3 寄存器可以在运行中进行读取和写入。

此寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

注意事项 在以下状态禁止访问 TABnCCR3 寄存器。详情参见 3.4.8 (2) 访问片上外设输入/输出专用寄存器。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
- 当 CPU 运行于内部振荡时钟工作时。



(a) 用作比较寄存器时的功能

即便在 TABnCTL0.TABnCE 位= 1 时, TABnCCR3 寄存器仍可以重新写入。

TABnCCR3 寄存器的设定值传送给 CCR3 缓冲寄存器。当 16 位计数器的计数值与 CCR3 缓冲寄存器的值相匹配时, 就会生成一个比较匹配中断请求信号 (INTTABnCC3)。如果此时 TOABn3 引脚输出允许, 则 TOABn3 引脚的输出会反相。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下, 当 TABnCCR3 寄存器用作捕获寄存器时, 若检测到捕获触发输入引脚 (TIABn3 引脚) 的有效边沿, 则 16 位计数器的当前计数值存储于 TABnCCR3 寄存器中。在脉宽测量模式下, 若检测到捕获触发输入引脚 (TIABn3 引脚) 的有效边沿, 16 位计数器的当前计数值就储存在 TABnCCR3 寄存器中, 且 16 位计数器清零 (0000H)。

即使捕获操作和读取 TABnCCR3 寄存器的操作发生冲突, 还是可以正确读取 TABnCCR3 寄存器的值。

备注 n = 0, 1

下表显示了捕获/比较寄存器在各工作模式下的功能, 以及如何向比较寄存器写入数据。

表 8-5. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

操作模式	捕获/比较寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	随时写入
外部事件计数器	比较寄存器	随时写入
外部触发脉冲输出	比较寄存器	批量写入
单次触发脉冲输出	比较寄存器	随时写入
PWM (脉宽调制信号) 输出	比较寄存器	批量写入
自由运行定时器	捕获/比较寄存器	随时写入
脉冲宽度测量	捕获寄存器	-
三角波 PWM 模式	比较寄存器	批量写入

(12) TABn 计数器读取缓冲寄存器 (TABnCNT)

TABnCNT 寄存器是读出缓冲寄存器，可以读取 16 位计数器的计数值。

如果在 TABnCTL0.TABnCE 位=1 时读取该寄存器，可以读出 16 位定时器的计数值。

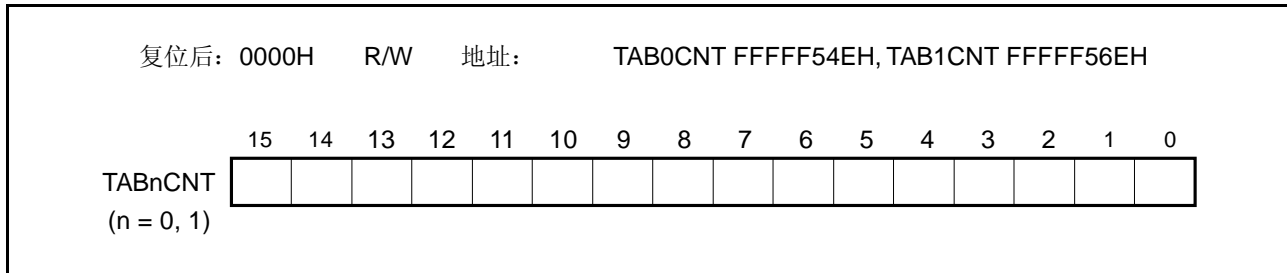
该寄存器为只读寄存器，按 16 位宽度读取。

当 TABnCE 位=0 时，TABnCNT 寄存器的值被清为 0000H。如果此时读取 TABnCNT 寄存器，则不会读出 16 位计数器的值 (FFFFH)，读到的值为 0000H。

在复位后 TABnCNT 寄存器的值被清为 0000H，因为 TPnCE 位被清为 0。

注意事项 在以下状态禁止访问 TABnCNT 寄存器。详情参见 3.4.8 (2) 访问片上外设输入/输出专用寄存器。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
- 当 CPU 运行于内部振荡时钟工作时。



8.5 操作

定时器 TABn 可以执行如下操作。

操作	TABnCTL1.TABnEST 位 (软件触发位)	TIABn0 引脚 (外部触发输入)	捕获/比较寄存器设置	比较寄存器写入
间隔计时器模式	无效	无效	仅作比较寄存器	随时写入
外部事件计数器模式 ^{注 1}	无效	无效	仅作比较寄存器	随时写入
外部触发脉冲输出模式 ^{注 2}	有效	有效	仅作比较寄存器	批量写入
单次触发脉冲输出模式 ^{注 2}	有效	有效	仅作比较寄存器	随时写入
PWM 输出模式	无效	无效	仅作比较寄存器	批量写入
自由定时器模式	无效	无效	可以切换	随时写入
脉冲宽度测量模式 ^{注 2}	无效	无效	仅作捕获寄存器	无效
三角波 PWM 模式	无效	无效	仅作比较寄存器	批量写入

- 注**
1. 在用作外部事件计数器模式下，应该指定对 TIABn0 引脚捕获触发输入的有效边沿不作检测（将 TABnIOC1.TABnIS1 和 TABnIOC1.TABnIS0 位清为“00”）。
 2. 在外部触发脉冲输出模式、单次触发脉冲输出模式和脉冲宽度测量模式下，选择内部时钟作为计数时钟。（将 TABnCTL1.TABnEEE 位清为 0）。

备注 n = 0, 1

8.5.1 间隔定时器模式 (TABnMD2 至TABnMD0 位=000)

在间隔定时器模式下，如果 TABnCTL0.TABnCE 位被置为 1，则会以指定时间间隔生成中断请求信号 (INTTABnCC0)，从 TOABn0 引脚输出方波，指定的时间间隔等于方波的半周期。

通常在间隔定时器模式中不使用 TABnCCR1 至 TABnCCR3 寄存器。

图 8-2. 间隔定时器的配置

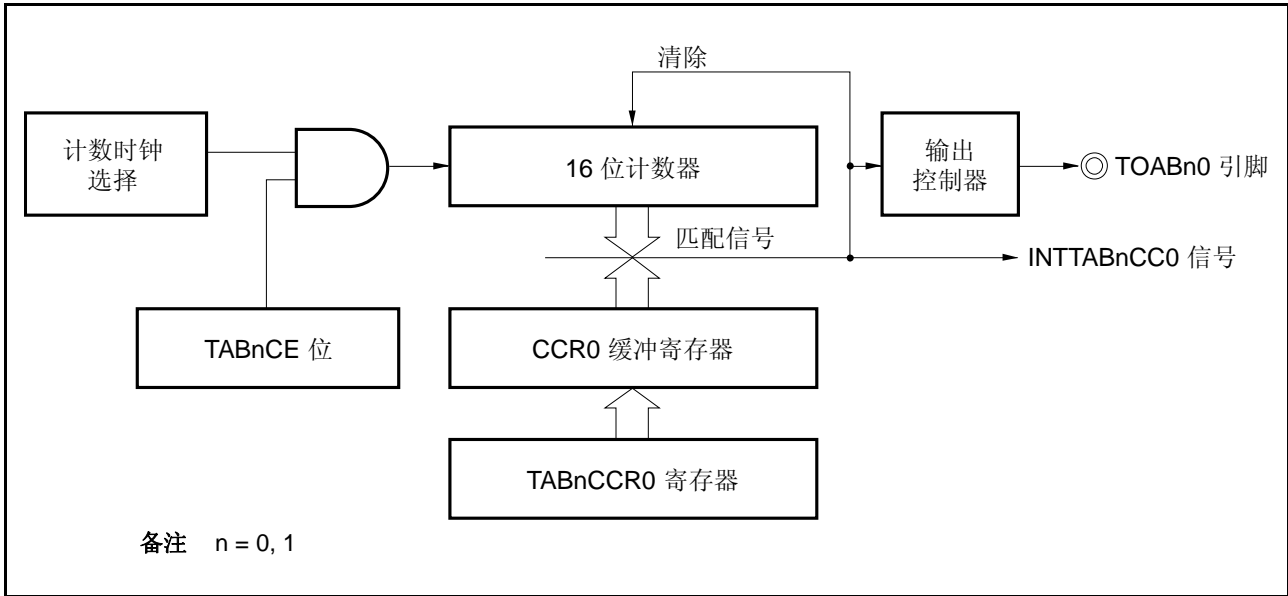
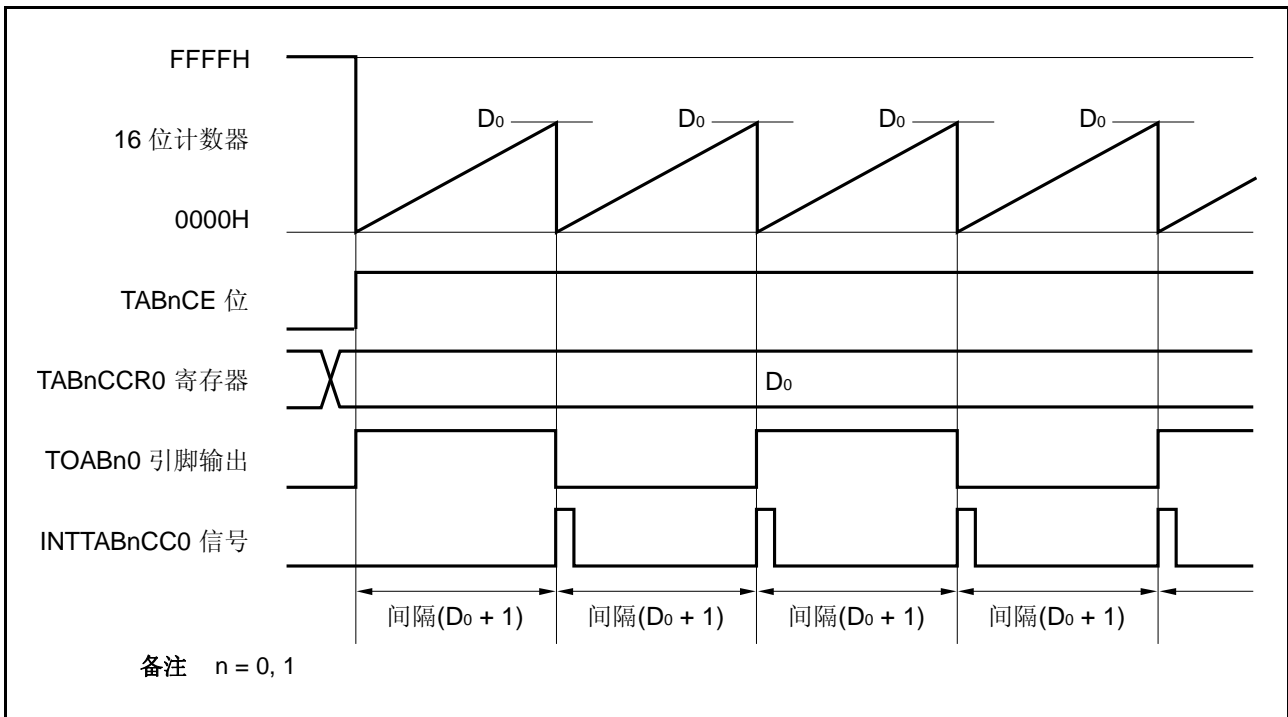


图 8-3. 间隔定时器模式的基本时序



当 TABnCE 位被置为 1 时，16 位计数器的值由计数时钟同步从 FFFFH 清零为 0000H，计数器开始计数。此时，TOABn0 引脚的输出反相。此外，TABnCCR0 寄存器的设定值传送给 CCR0 缓冲寄存器。

当 16 位计数器的计数值和 CCR0 缓冲寄存器的值相匹配时，则 16 位计数器清为 0000H，TOABn0 引脚输出反相，并生成一个比较匹配中断请求信号 (INTTABnCC0)。

时间间隔可用下述表达式计算。

$$\text{时间间隔} = (\text{TABnCCR0 寄存器的设定值} + 1) \times \text{计数时钟周期}$$

图 8-4. 间隔定时器模式操作时的寄存器设置 (1/2)

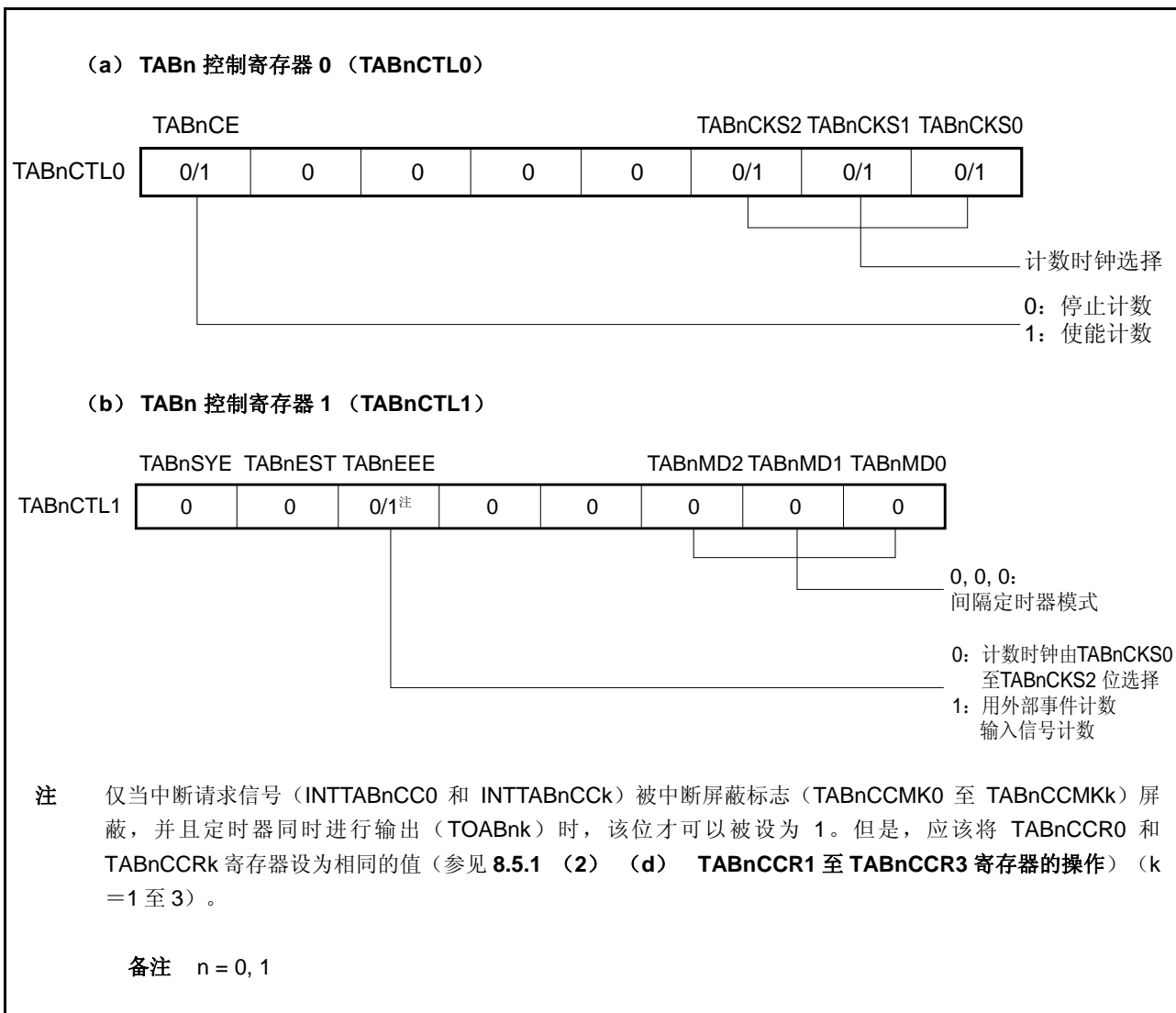
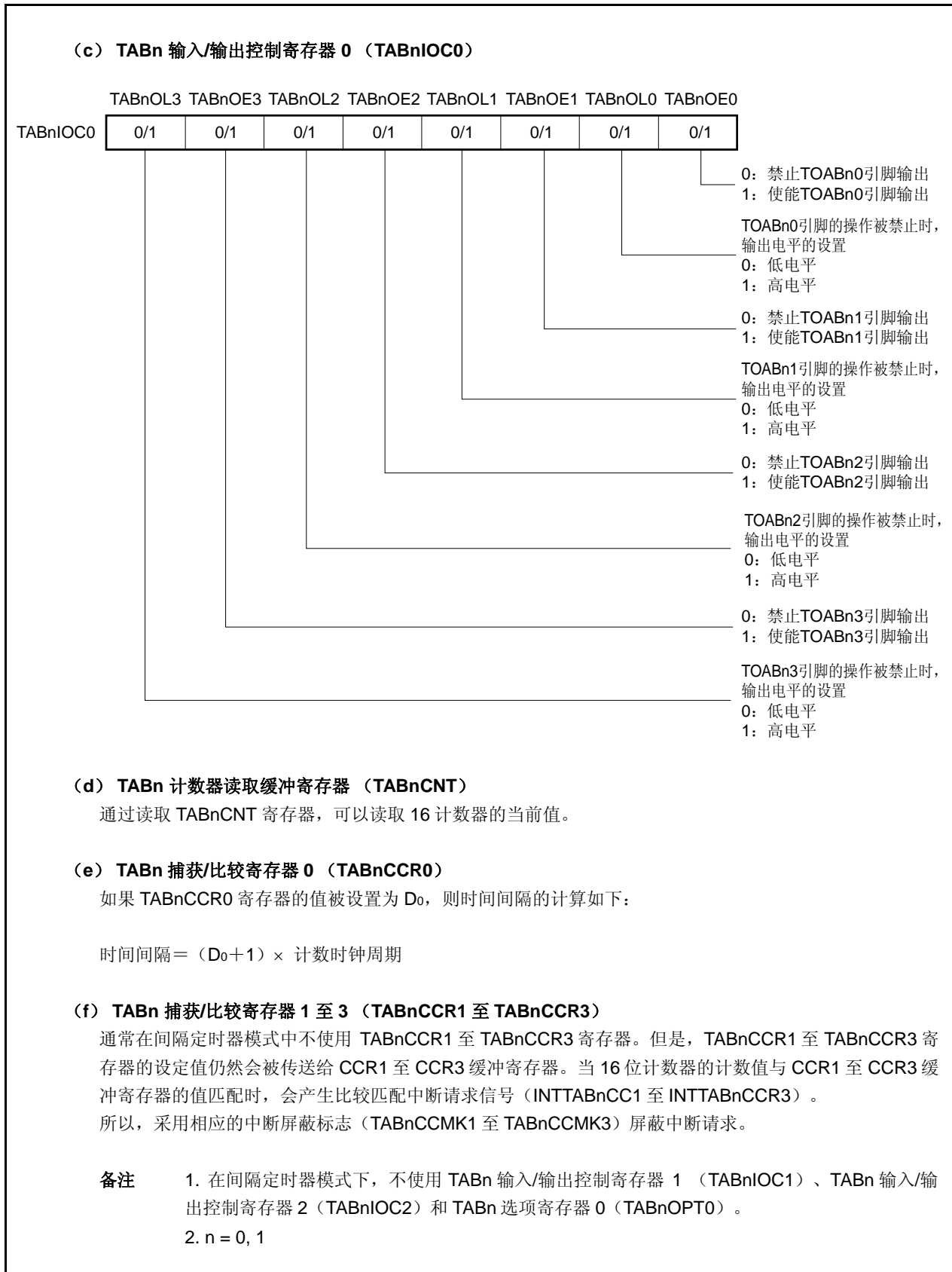
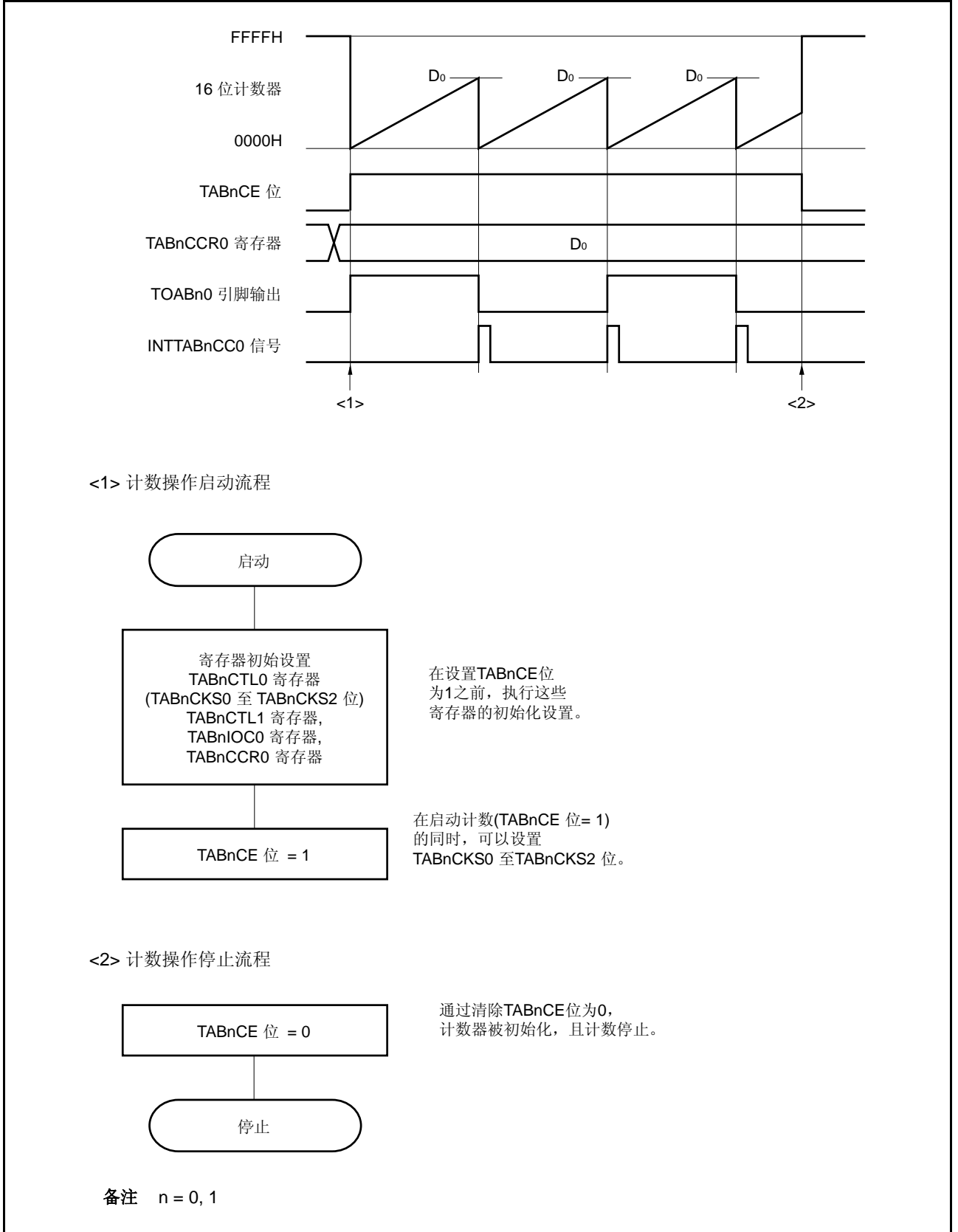


图 8-4. 间隔定时器模式操作时的寄存器设置 (2/2)



(1) 间隔定时器模式操作流程

图 8-5. 间隔定时器模式下的软件处理流程

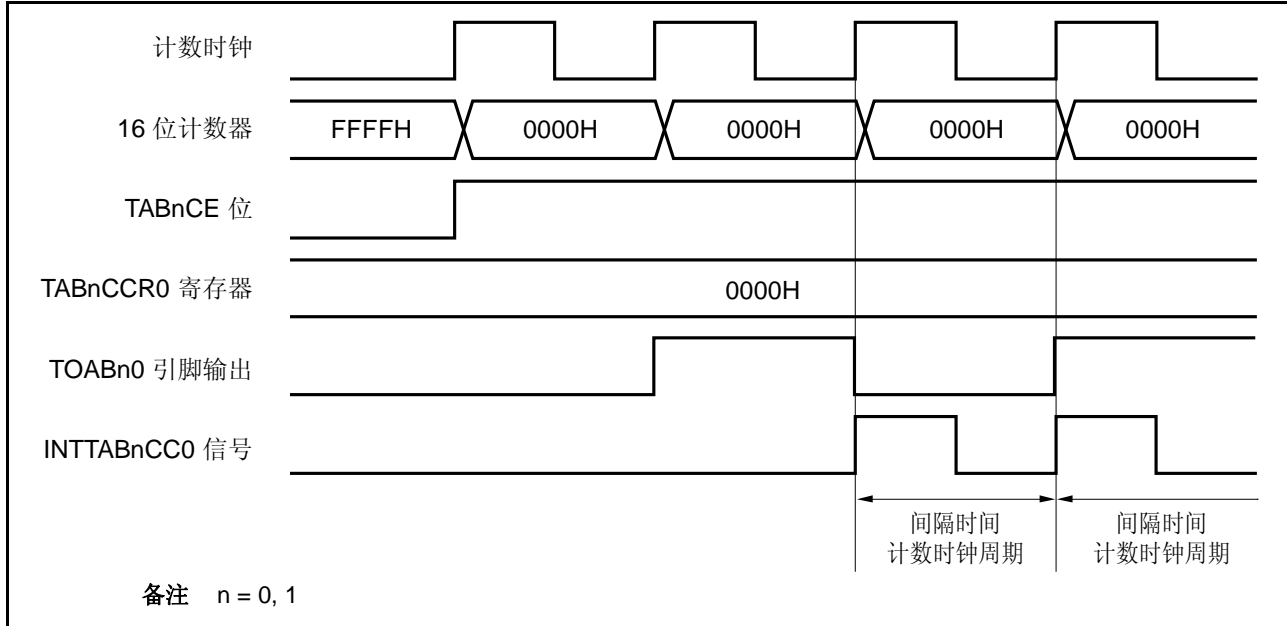


(2) 间隔定时器模式操作时序

(a) TABnCCR0 寄存器清为 0000H 时的操作

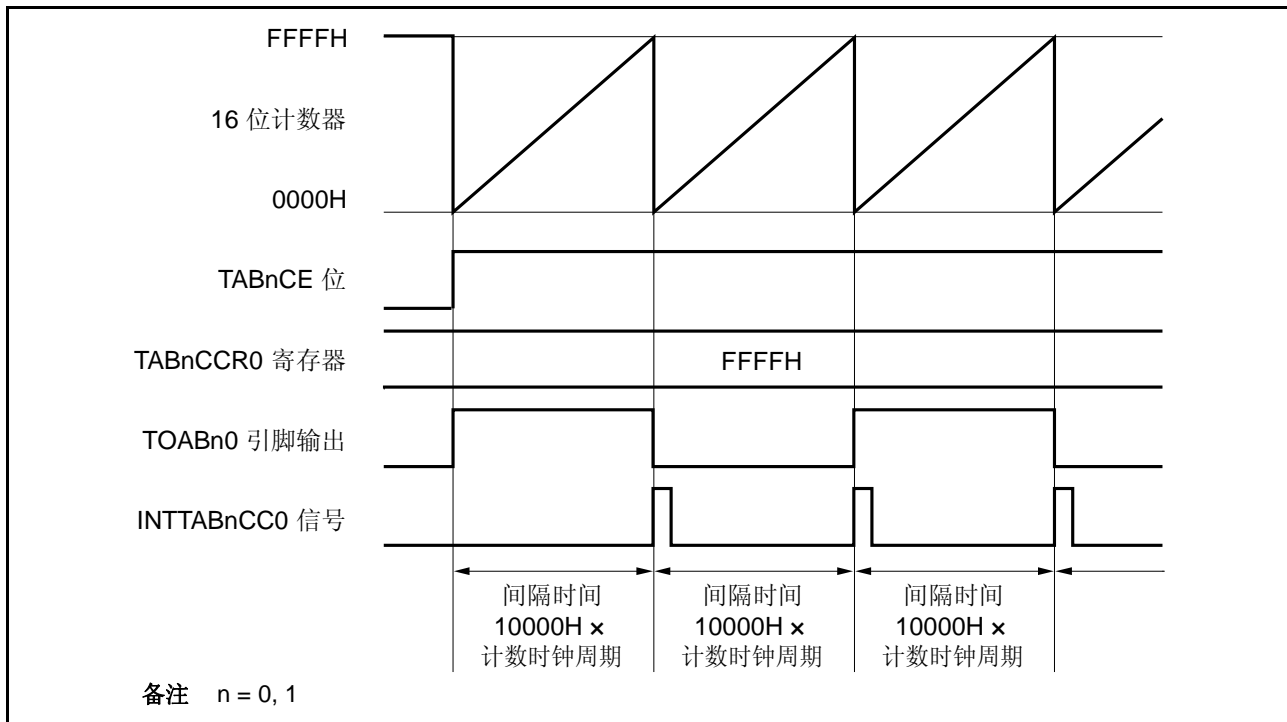
如果 TABnCCR0 寄存器被清为 0000H，第二个时钟和之后的每个时钟计数时都会产生 INTTABnCC0 信号，而且，TOABn0 引脚的输出反相。

16 位计数器的值总是 0000H。



(b) TABnCCR0 寄存器被设置为 FFFFH 时的操作

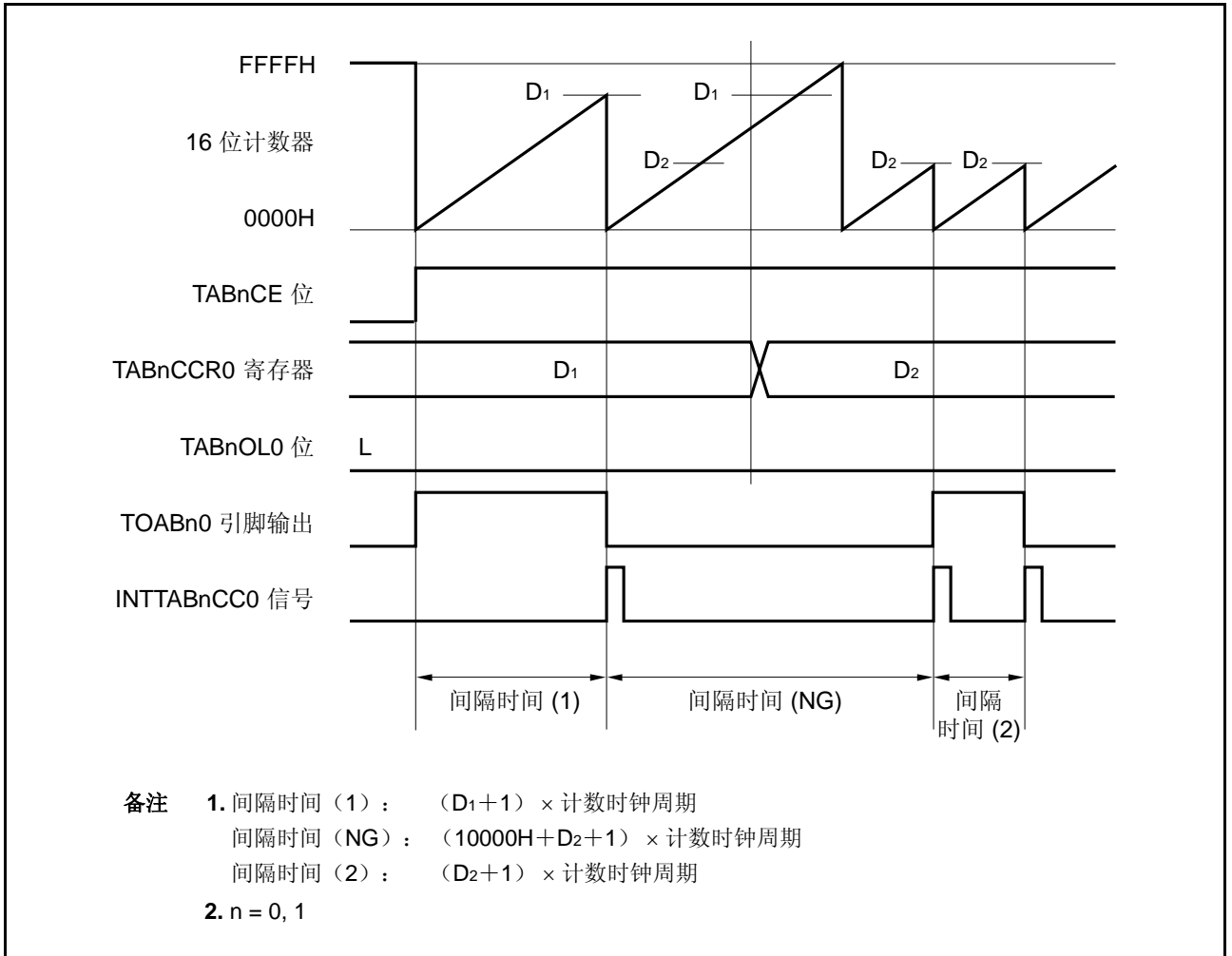
如果 TABnCCR0 寄存器被置为 FFFFH，那么 16 位计数器可以计数到 FFFFH。该计数器在下一个计数时刻被同步清零。产生 INTTABnCC0 信号，并且 TOABn0 引脚输出反相。此时，不产生溢出中断请求信号 (INTTABnOV)，而且，溢出标志 (TABnOPT0.TABnOVF 位) 也不会置 1。



(c) 重新写入 TABnCCR0 寄存器的注意事项

要将 TABnCCR0 寄存器的值变为比当前值小的值，应停止计数，然后改变其设定值。

如果在计数器工作期间，将寄存器 TABnCCR0 的值重写为比当前值小的值，16 位计数器可能会溢出。



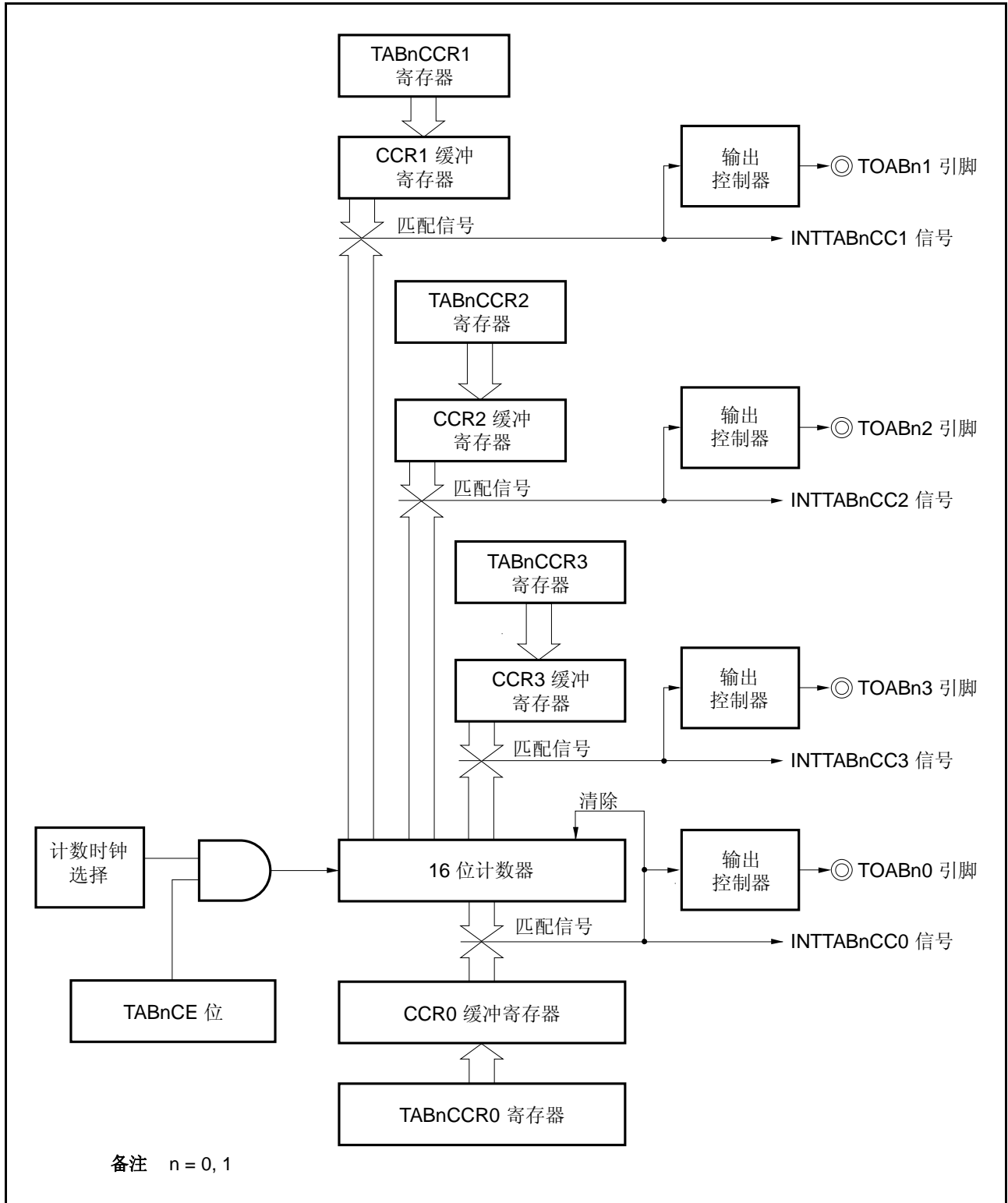
当 16 位计数器计数值大于 D_2 而小于 D_1 时，如果将 TABnCCR0 寄存器的值从 D_1 变为 D_2 ，则在重写 TABnCCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。结果，用来比较的 16 位计数器的值是 D_2 。

但是，由于计数值已经超过了 D_2 ，所以 16 位计数器就会一直向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值和 D_2 相符时才产生 INTTABnCC0 信号，且 TOABn0 引脚的输出反相。

因此，可能不会在原先预计的间隔时间“ $(D_1 + 1) \times$ 计数时钟周期”或“ $(D_2 + 1) \times$ 计数时钟周期”时产生 INTTABnCC0 信号，而会以“ $(10000H + D_2 + 1) \times$ 计数时钟周期”的间隔产生中断请求信号。

(d) TABnCCR1 至 TABnCCR3 寄存器的操作

图 8-6. TABnCCR1 至 TABnCCR3 寄存器的配置

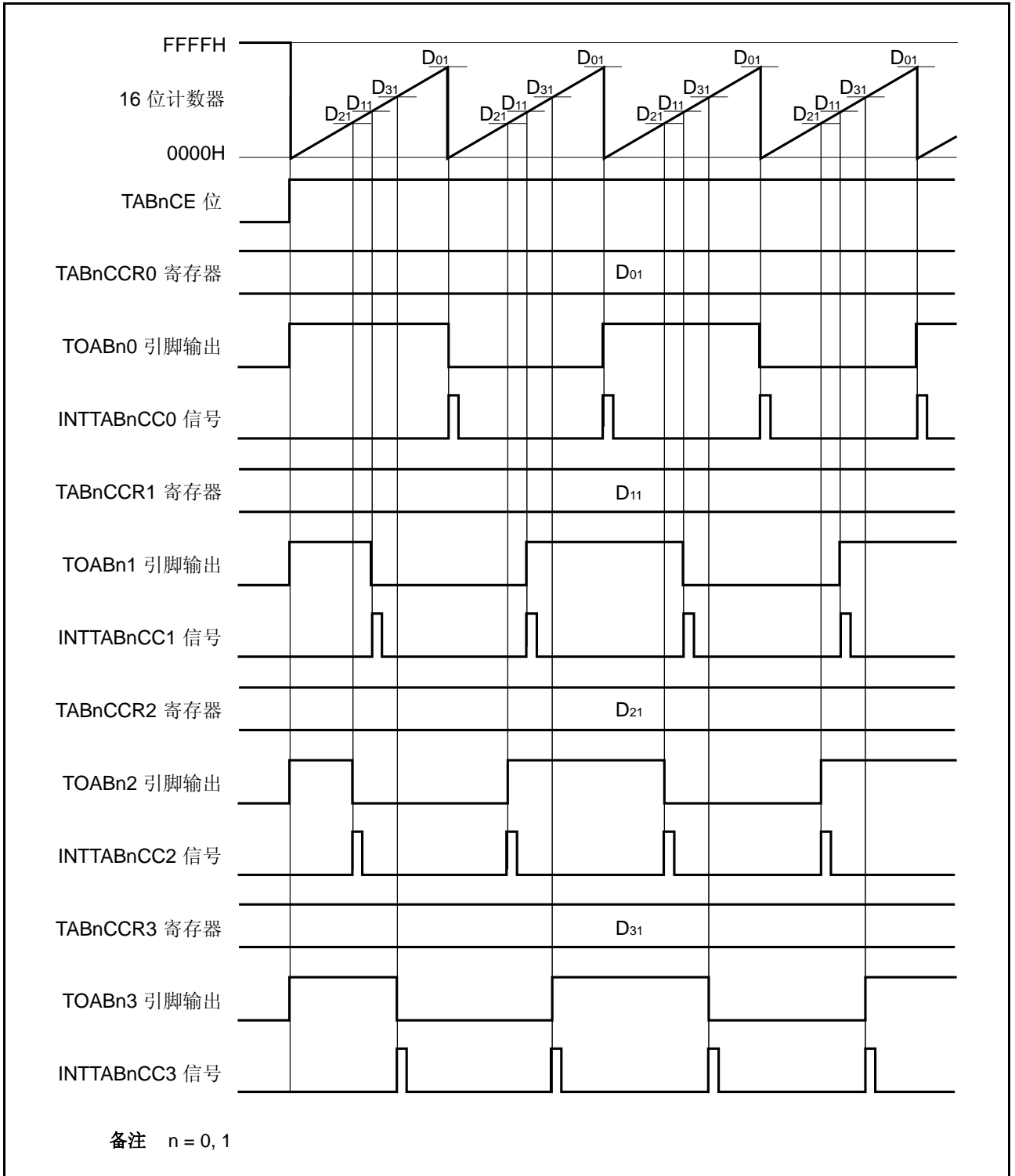


如果 $TABnCCRk$ 寄存器的设定值小于 $TABnCCR0$ 寄存器的设定值，则每个计数周期产生一次 $INTTABnCCk$ 中断请求信号。同时， $TOABnk$ 引脚的输出反相。

$TOABnk$ 引脚输出一个方波，其周期与 $TOABn0$ 引脚的输出周期相同。

备注 $k=1$ 至 3
 $n=0, 1$

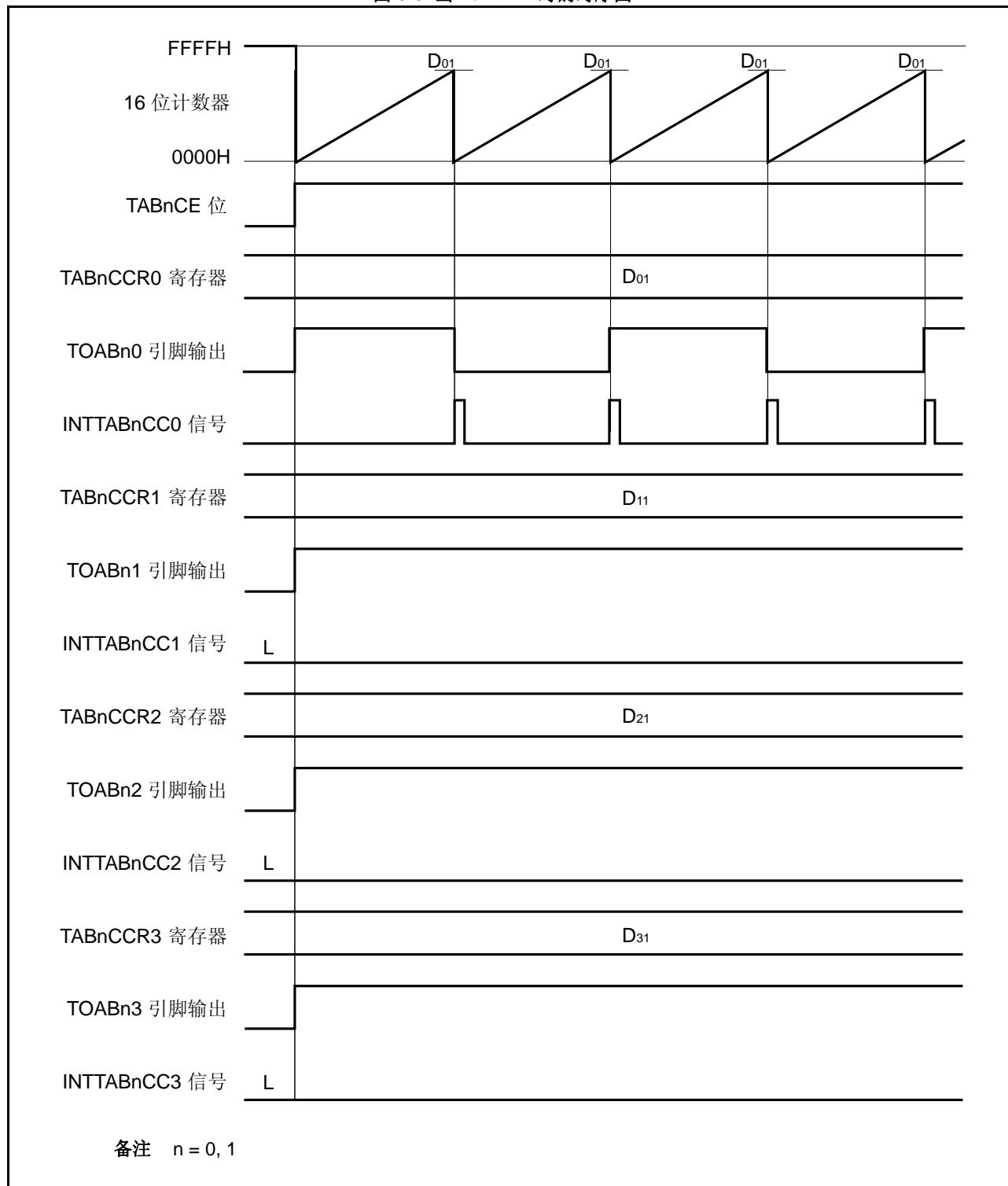
图 8-7. 当 $D_{01} \geq D_{k1}$ 时的时序图



如果 $TABnCCRk$ 寄存器的设定值大于 $TABnCCR0$ 寄存器的设定值，则 16 位计数器的计数值不会和 $TABnCCRk$ 寄存器的值发生匹配。因此也就不会产生 $INTTABnCCk$ 信号， $TOABnk$ 引脚的输出也不会改变。

备注 k=1 至 3

图 8-8. 当 $D_{01} < D_{k1}$ 时的时序图



备注 n = 0, 1

8.5.2 外部事件计数模式 (TABnMD2 至 TABnMD0 位=001)

在外部事件计数器模式下，当 TABnCTL0.TABnCE 位被置为 1 时，对外部事件计数输入信号的有效边沿进行计数，每次有效边沿的计数数量达到指定数目时，产生一个中断请求信号 (INTTABnCC0)。不能使用 TOABn0 引脚。

通常在外部事件计数模式下，不使用 TABnCCR1 至 TABnCCR3 寄存器。

图 8-9. 外部事件计数模式的配置

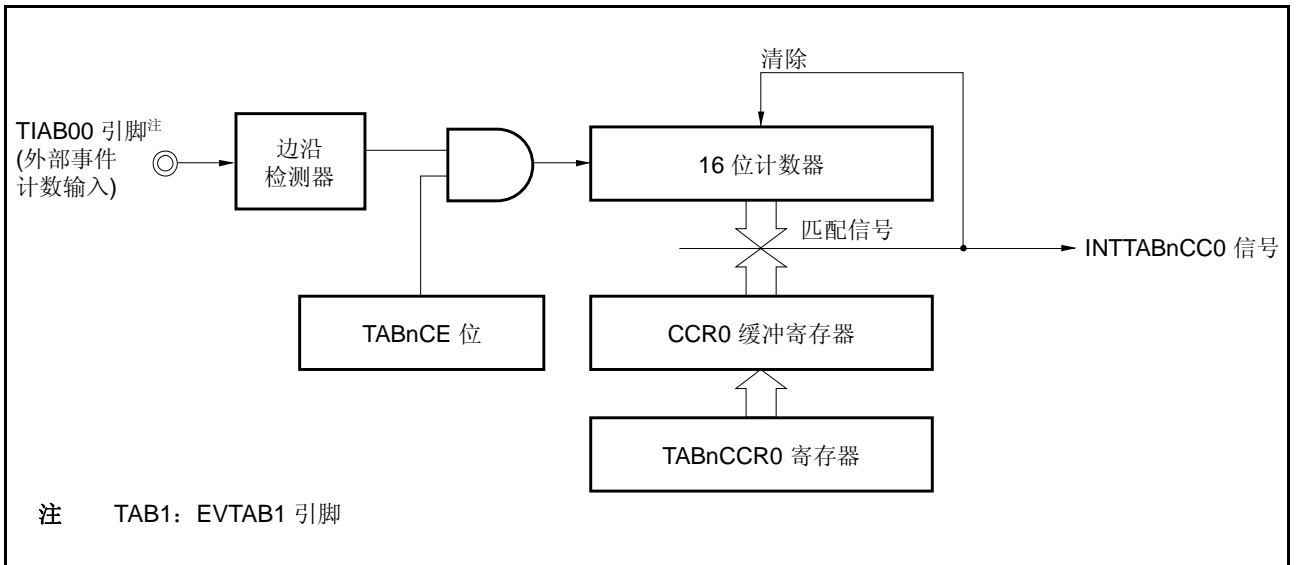
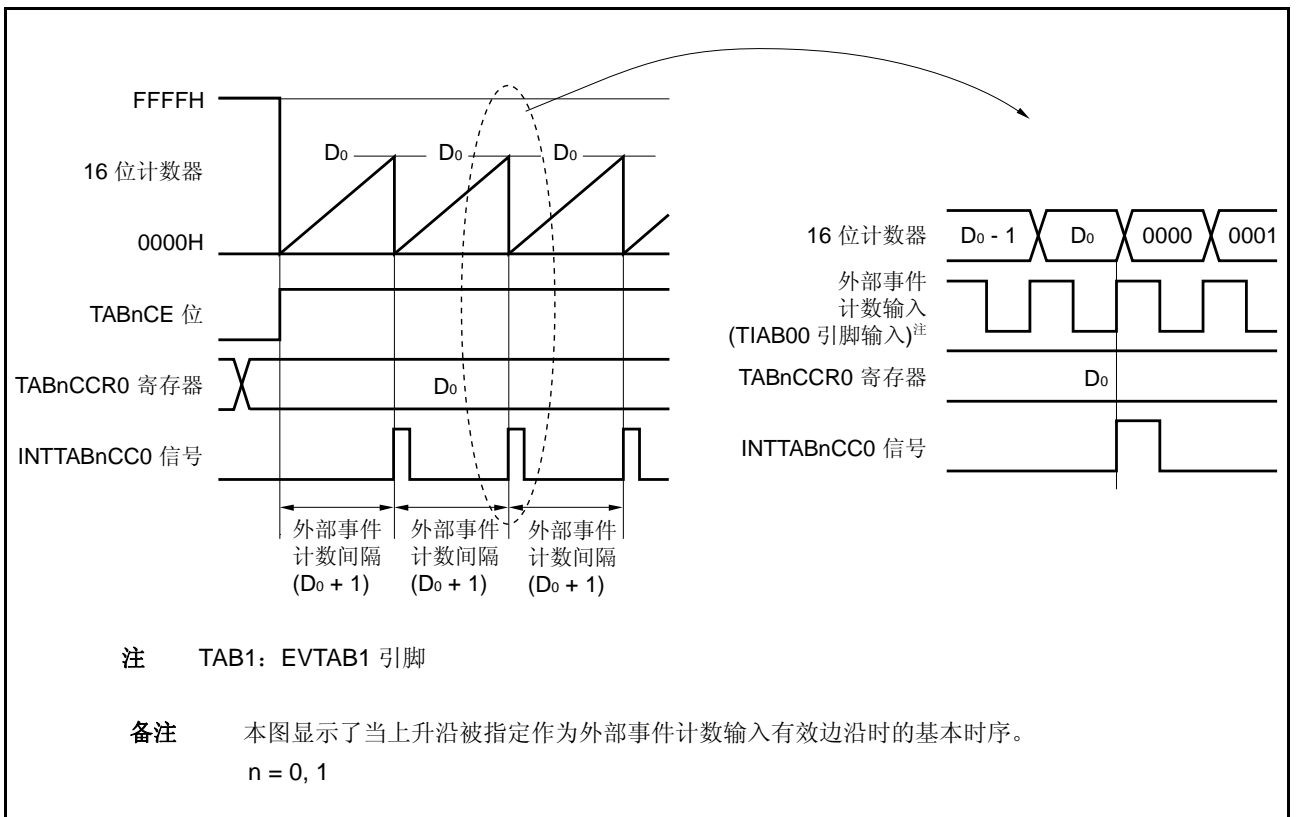


图 8-10. 外部事件计数模式的基本时序



当 TABnCE 位被置为 1 时，16 位计数器的值从 FFFFH 被清为 0000H。计数器对检测到的每一次外部事件计数输入的有效边沿进行计数。此外，TABnCCR0 寄存器的设定值被传送给 CCR0 缓冲寄存器。

当 16 位计数器的计数值和 CCR0 缓冲寄存器的数值相匹配时，16 位计数器清零为 0000H，同时生成一个比较匹配中断请求信号 (INTTABnCC0)。

每当外部事件计数输入的有效边沿数量检测达到 (TABnCCR0 寄存器设定值+1) 时，就会生成 INTTABnCC0 (比较匹配中断请求) 信号。

图 8-11. 外部事件计数模式下的寄存器操作设置 (1/2)

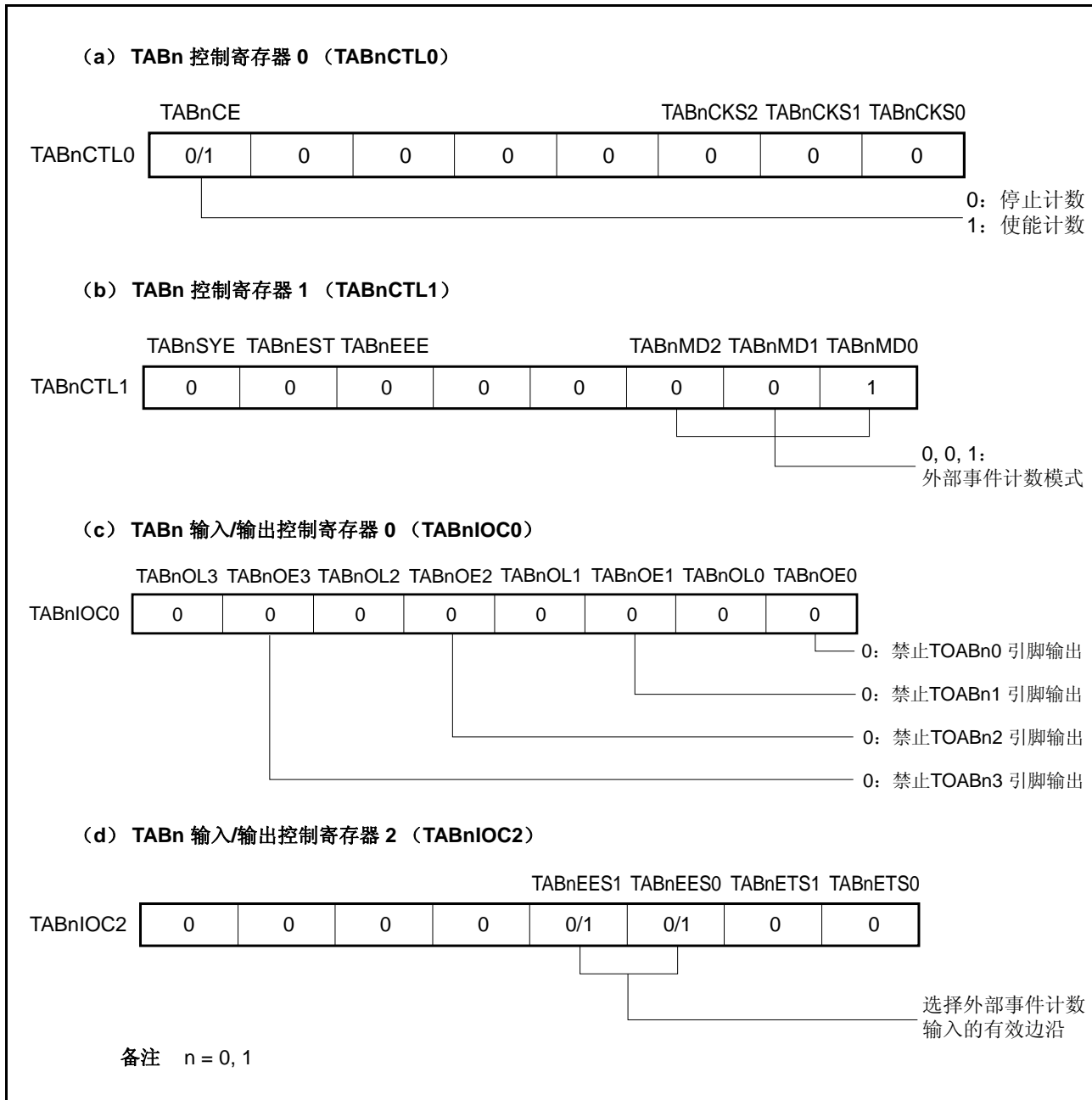


图 8-11. 外部事件计数模式下的寄存器操作设置 (2/2)

(e) TABn 计数器读取缓冲寄存器 (TABnCNT)

可以通过读取 TABnCNT 寄存器，读取 16 位计数器的计数值。

(f) TABn 捕获/比较寄存器 0 (TABnCCR0)

如果 TABnCCR0 寄存器的值被设定为 D_0 ，当外部事件计数值达到 (D_0+1) 时，计数器被清 0，同时产生一个比较匹配中断请求信号 (INTTABnCC0)。

(g) TABn 捕获/比较寄存器 1 至 3 (TABnCCR1 至 TABnCCR3)

通常在外部事件计数模式中不使用 TABnCCR1 至 TABnCCR3 寄存器。但是，TABnCCR1 至 TABnCCR3 寄存器的设定值仍然被传送给 CCR1 至 CCR3 缓冲寄存器。当 16 位计数器的计数值和 CCR1 至 CCR3 缓冲寄存器的值匹配时，会产生一个比较匹配中断请求信号 (INTTABnCC1 至 INTTABnCC3)。

所以，采用中断屏蔽标志 (TABnCCMK1 至 TABnCCMK3) 来屏蔽中断信号。

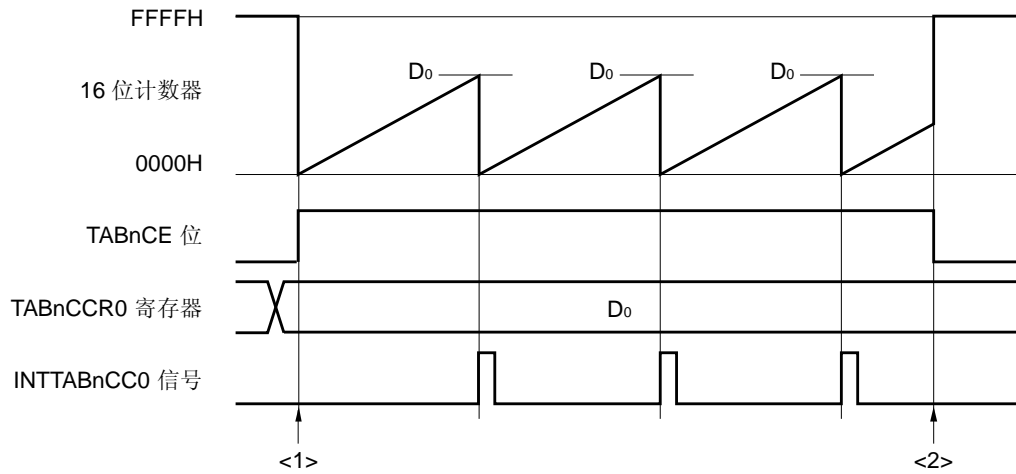
注意事项 当外部时钟用作计数时钟时，外部时钟只能从 TIABn0 引脚输入。此时，将 TABnIOC1.TABnIS1 和 TABnIOC1.TABnIS0 位清为 00 (捕获触发输入 (TIABn0 引脚)：无边沿检测)。

备注 1. 在外部事件计数模式下，不使用 TABn 输入/输出控制寄存器 1 (TABnIOC1) 和 TABn 选项寄存器 0 (TABnOPT0)。

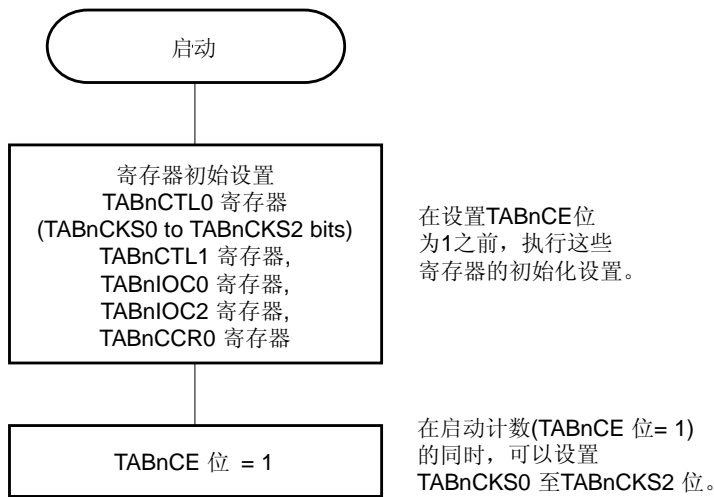
2. $n = 0, 1$

(1) 外部事件计数模式的操作流程

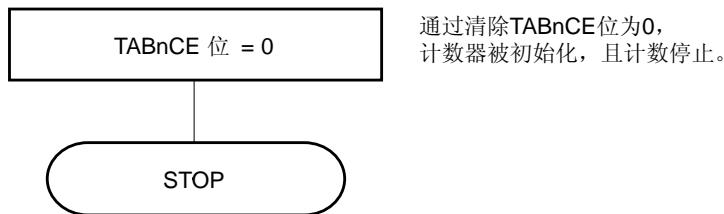
图 8-12. 外部事件计数模式下的软件处理流程



<1> 计数操作启动流程



<2> 计数操作停止流程



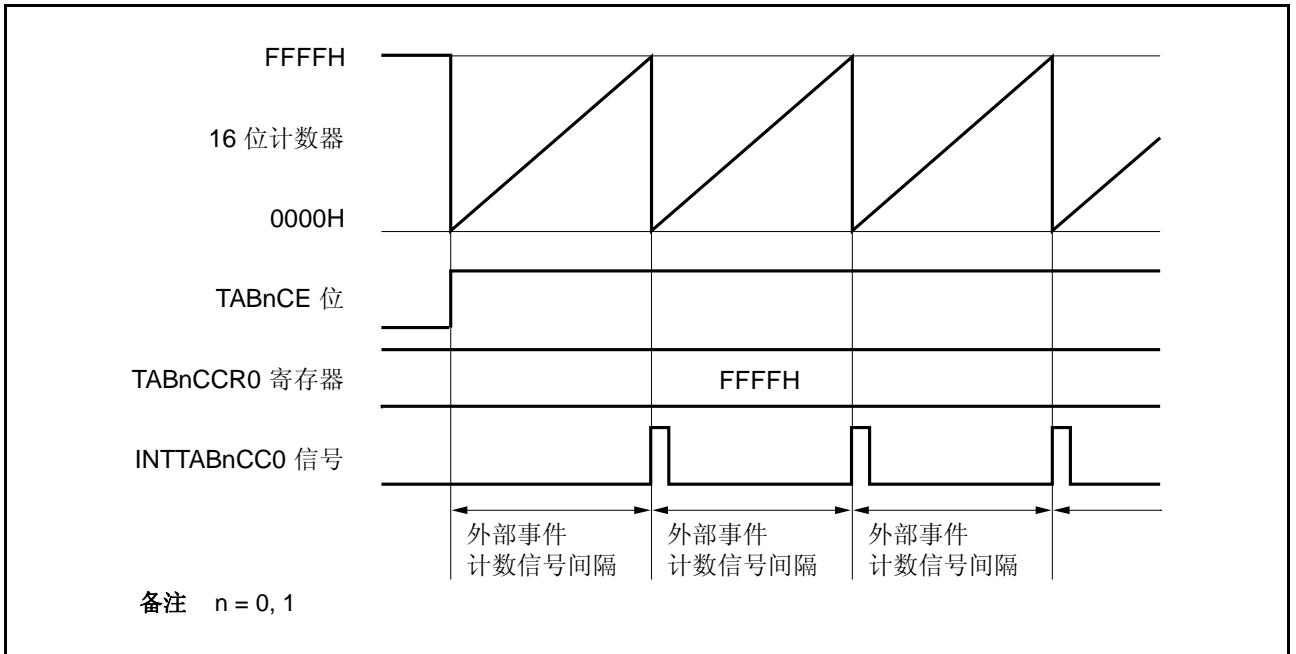
备注 n = 0, 1

(2) 外部事件计数模式的操作时序

- 注意事项
1. 外部事件计数模式下，不要将 TABnCCR0 寄存器设置为 0000H。
 2. 外部事件计数模式下，禁止使用定时器输出。如果使用外部事件计数输入来进行定时器输出，应设置为间隔定时器模式，并且选择外部事件计数输入允许的操作作为计数时钟 (TABnCTL1.TABnMD2 至 TABnCTL1.TABnMD0 位= 000, TABnCTL1.TABnEEE 位= 1)。

(a) 如果 TABnCCR0 寄存器被设置为 FFFFH 时的操作

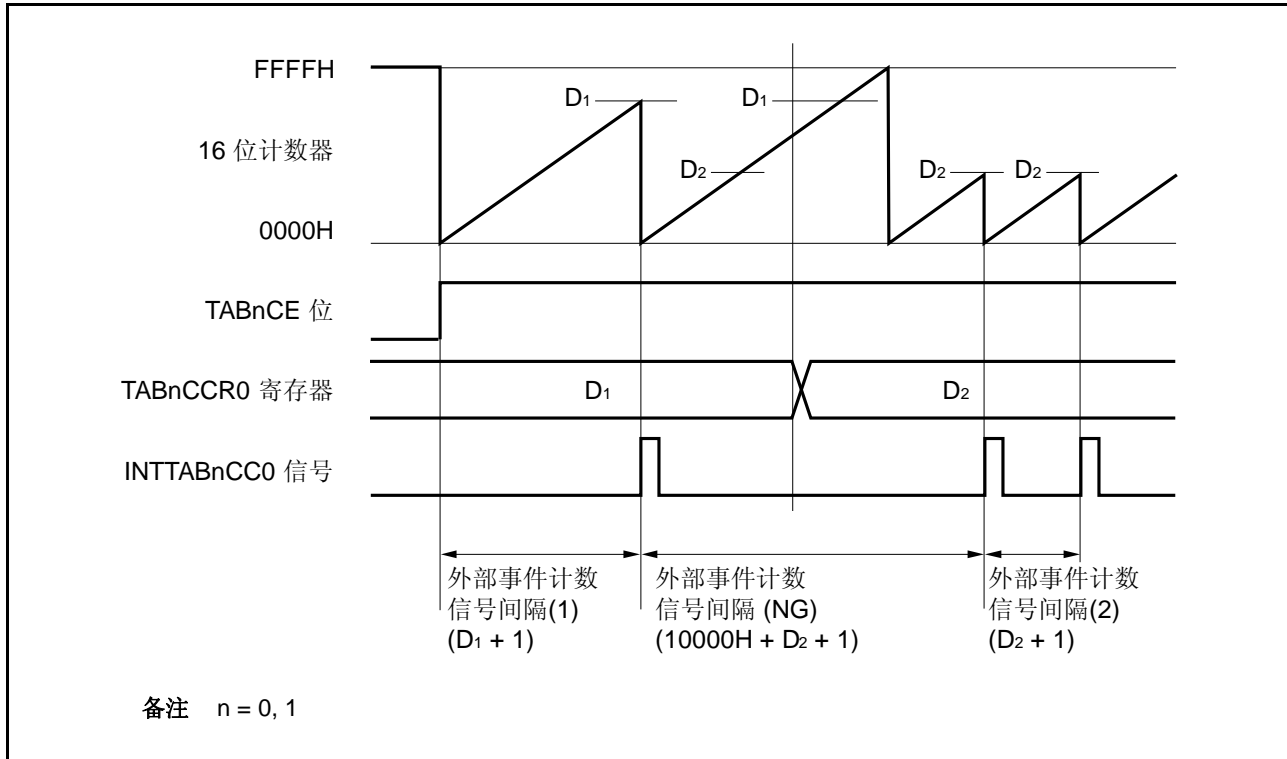
如果 TABnCCR0 寄存器被设置为 FFFFH，16 位计数器每次检测到的外部事件输入信号有效边沿时，都计数到 FFFFH。当下一个计数到来时，16 位计数器被同步清为 0000H，并且产生 INTTABnCC0 信号。此时，不设置 TABnOPT0.TABnOVF 位。



(b) 重写 TABnCCR0 寄存器的注意事项

要将 TABnCCR0 寄存器的值改为比当前值较小的值，应停止计数，然后改变其设定值。

如果在计数器进行计数期间，对寄存器 TABnCCR0 的值进行重写，则 16 位计数器可能会溢出。



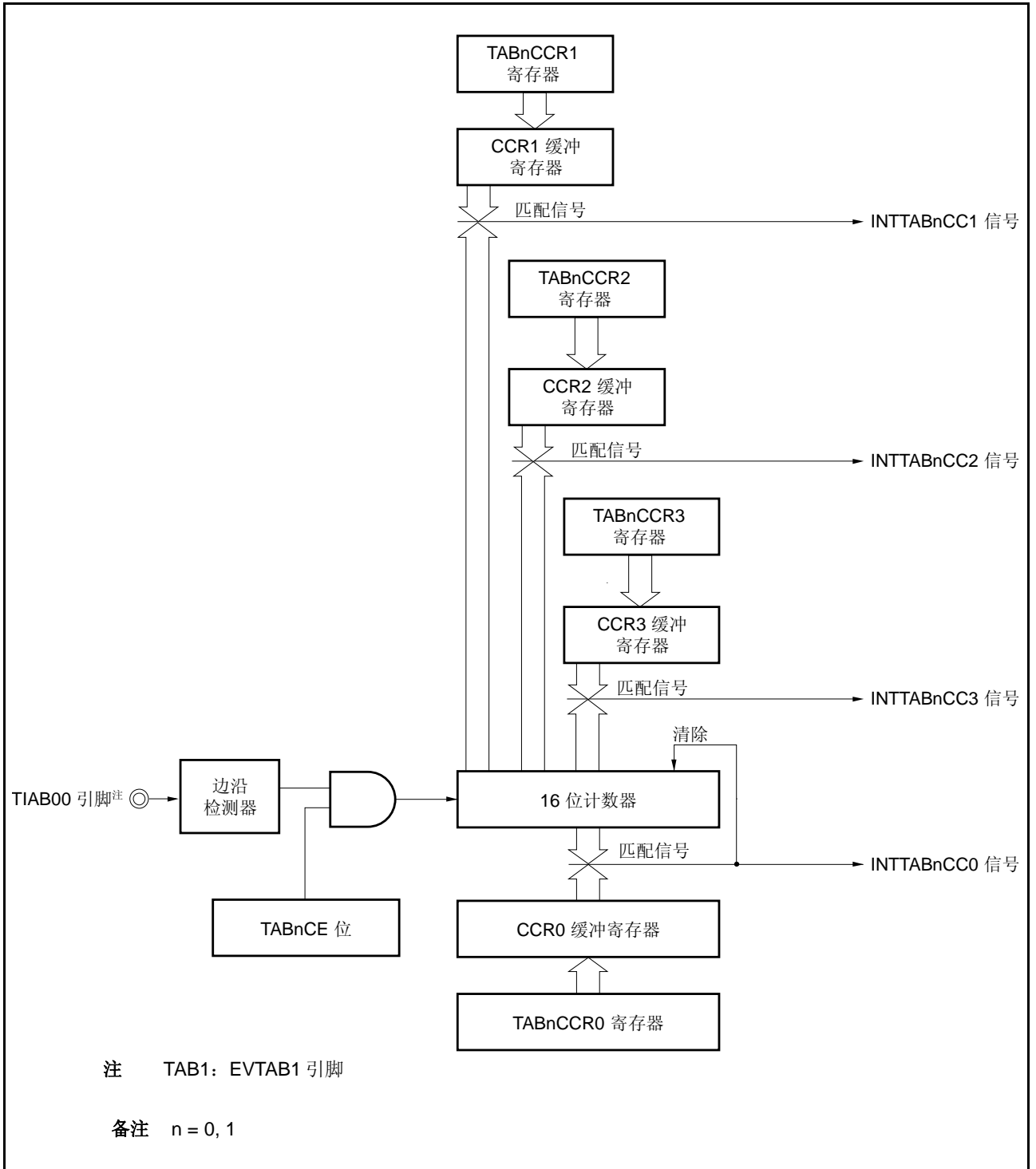
当 16 位计数器计数值大于 D₂ 而小于 D₁ 时，如果将 TABnCCR0 寄存器的值从 D₁ 变为 D₂，则在重写 TABnCCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。因而，与 16 位计数器相比较的值会是 D₂。

但是，由于计数值已经超过了 D₂，所以 16 位计数器就会向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值与 D₂ 匹配时，会生成 INTTABnCC0 信号。

因此，可能不会在原先预计的“(D₁ + 1) 次”或“(D₂ + 1) 次”有效边沿生成 INTTABnCC0 中断请求信号，而可能在“(10000H + D₂ + 1) 次”有效边沿生成中断请求信号。

(c) TABnCCR1 至 TABnCCR3 寄存器的操作

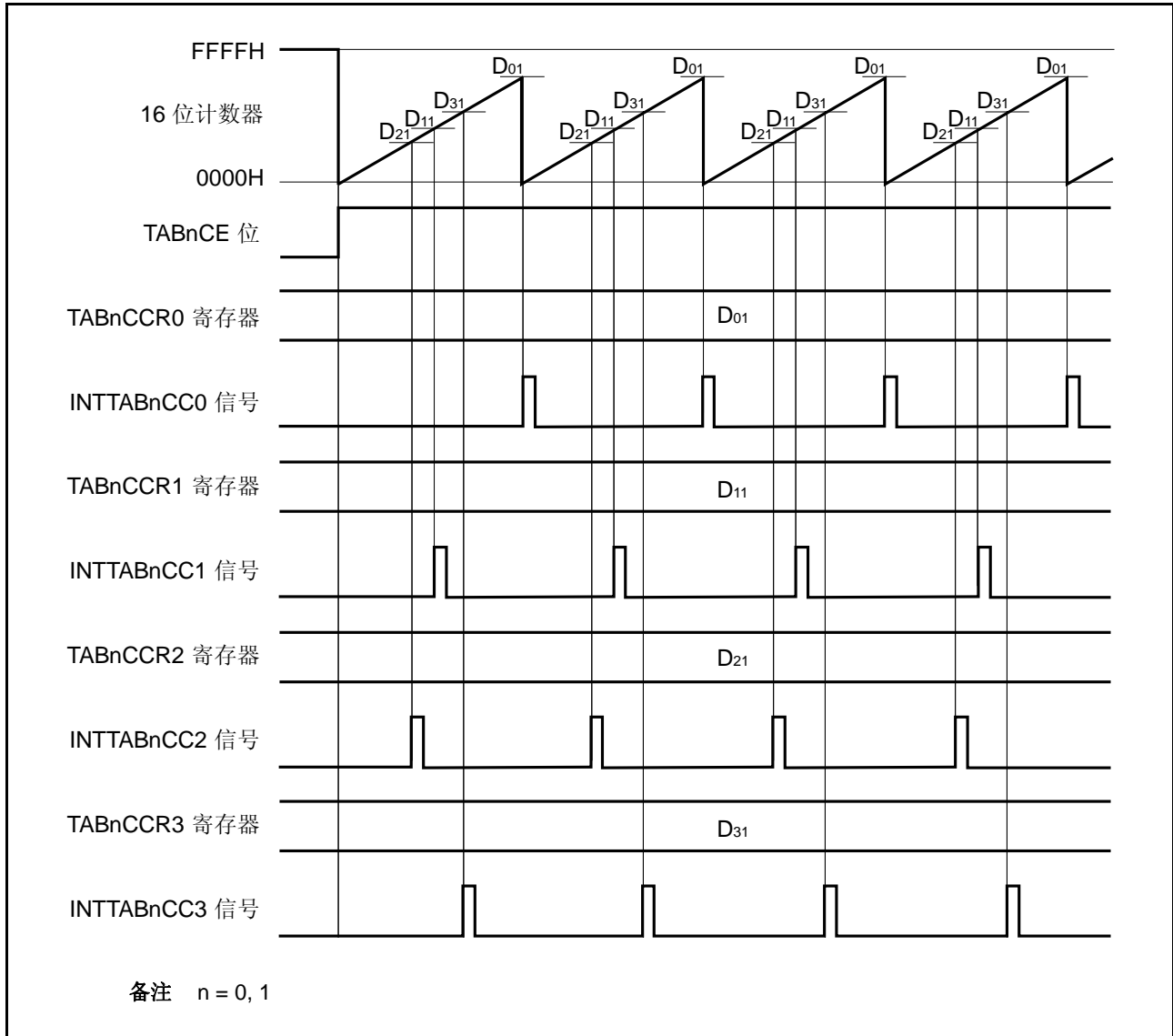
图 8-13. TABnCCR1 至 TABnCCR3 寄存器的配置



如果 $TABnCCRk$ 寄存器的设定值小于 $TABnCCR0$ 寄存器的设定值，则每个计数周期都产生一次 $INTTABnCCk$ 中断请求信号。

备注 $k=1$ 至 3
 $n=0, 1$

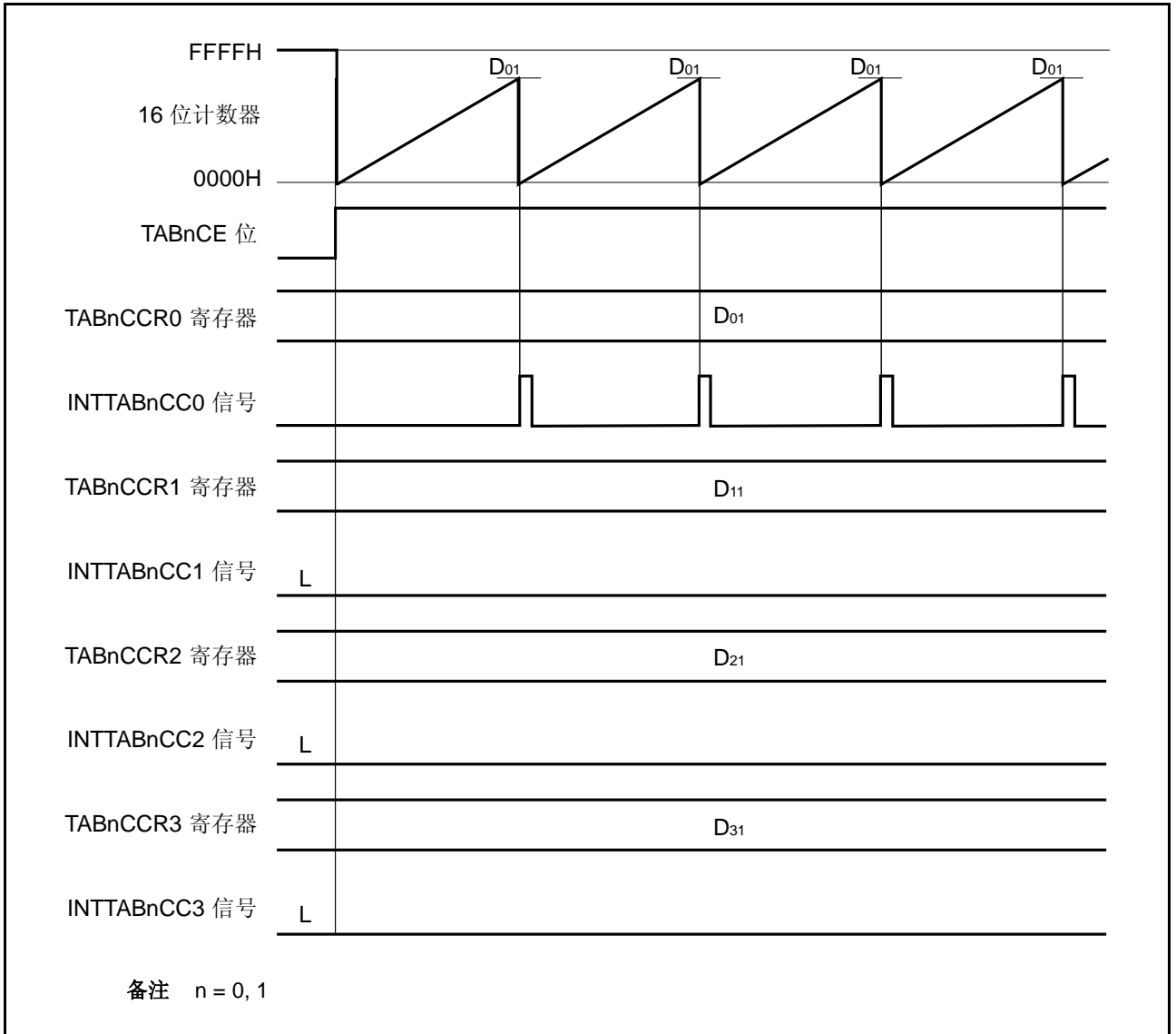
图 8-14. 当 $D_{01} \geq D_{k1}$ 时的时序图



如果 TABnCCRk 寄存器的设定值大于 TABnCCR0 寄存器的设定值，16 位计数器的计数值不会和 TABnCCRk 寄存器的值发生匹配。因此也就不会产生 INTTABnCCk 中断请求信号。

备注 k=1 至 3
n = 0, 1

图 8-15. 当 $D_{01} < D_{k1}$ 时的时序图



8.5.3 外部触发脉冲输出模式 (TABnMD2 至 TABnMD0 位=010)

在外部触发脉冲输出模式下，当 TABnCTL0.TABnCE 位被置为 1 时，TABn 等待触发信号，当检测到外部触发输入信号的有效边沿后，TABn 开始计数，并且从 TOABn1 至 TOABn3 引脚输出一个 PWM 波形。

不使用外部触发，而是使用软件触发，同样可以得到输出脉冲。当使用软件触发时，TOABn0 引脚输出一个方波，其半周期等于输出 PWM 的一个周期。

图 8-16. 外部触发脉冲输出模式的配置

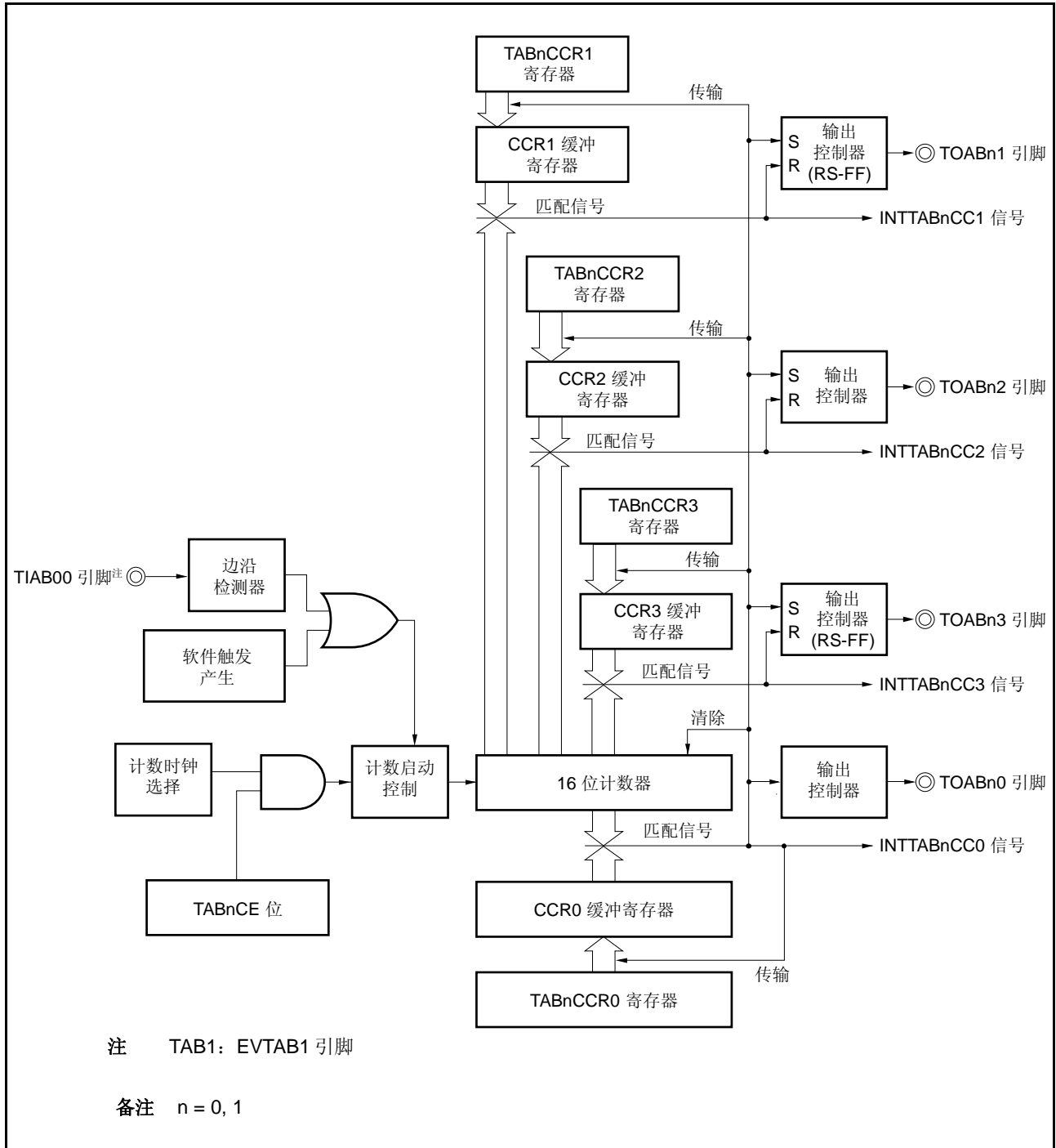
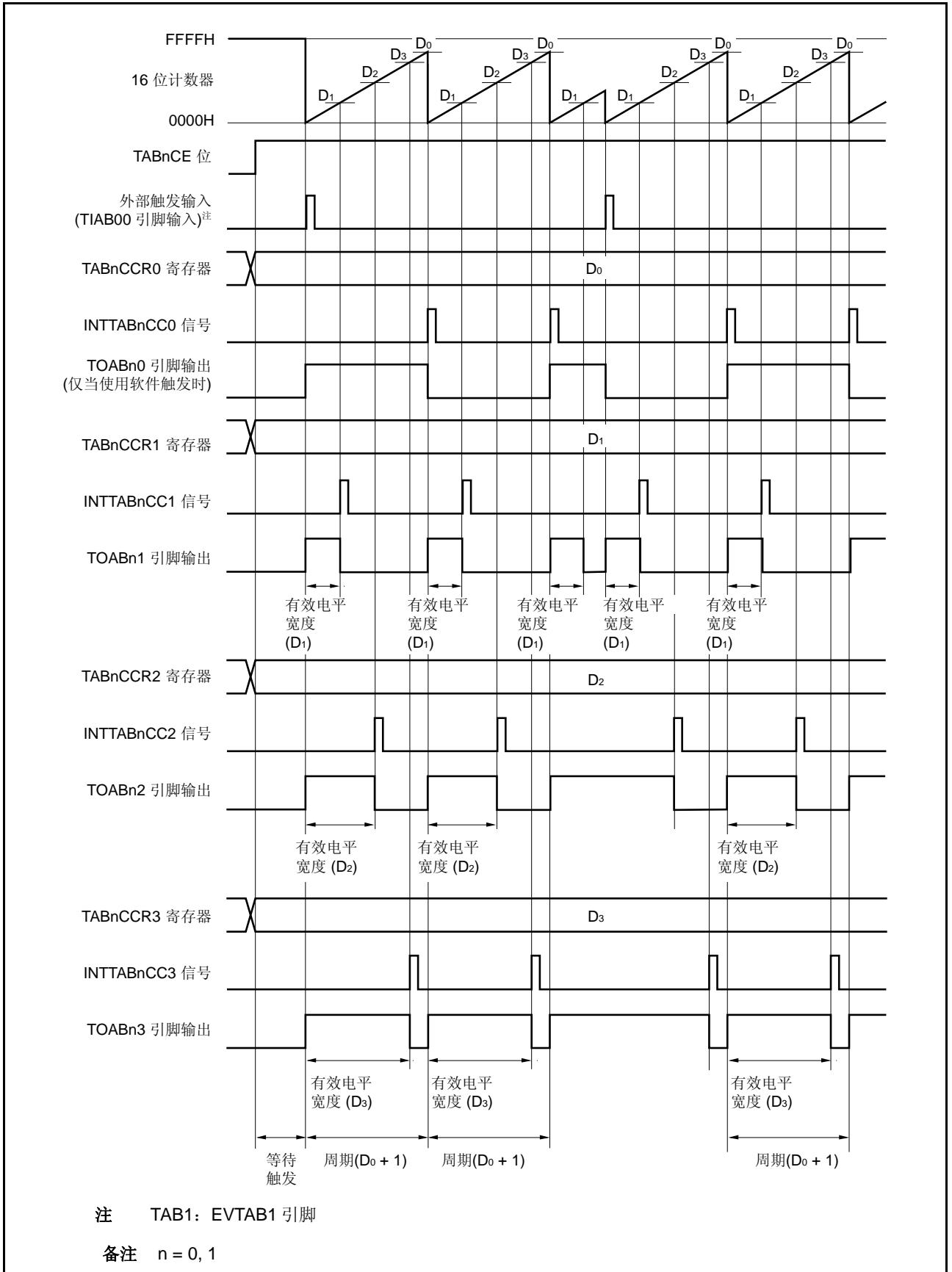


图 8-17. 外部触发脉冲输出模式的基本时序



当 TABnCE 位设置为 1 时, TABn 等待触发。当触发生产生时, 16 位计数器从 FFFFH 清零为 0000H, 同时开始计数, 并从 TOABnk 引脚输出一个 PWM 波形。如果在计数运行中再次产生触发, 则计数器清为 0000H 并重新开始计数。(TOABn0 引脚输出反相。在产生触发时, 不管当前状态如何(高/低), TOABnk 引脚都会输出一个高电平。

PWM 波形的有效电平宽度、周期、占空比系数的计算如下。

有效电平宽度 = (TABnCCRk 寄存器的设置值) × 计数时钟周期

周期 = (TABnCCR0 寄存器的设置值 + 1) × 计数时钟周期

占空比系数 = (TABnCCRk 寄存器的设置值) / (TABnCCR0 寄存器的设置值 + 1)

当 16 位计数器在其计数值与 CCR0 缓冲寄存器的值匹配之后进行下一次计数时就会同步产生比较匹配(中断)请求信号 INTTABnCC0, 且 16 位计数器清零为 0000H。当 16 位计数器的计数值与 CCRk 缓冲寄存器的值匹配时, 会生成比较匹配中断请求信号 INTTABnCCk。

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配, 而且 16 位计数器清零为 0000H 时, TABnCCRm 寄存器的设置值就被传送到 CCRm 缓冲寄存器。

外部触发输入信号的有效边沿, 或将软件触发 (TABnCTL1.TABnEST 位) 设为 1, 都可以用作触发信号。

备注 k=1 至 3
m=0 至 3
n=0, 1

图 8-18. 外部触发脉冲输出模式的寄存器设置 (1/3)

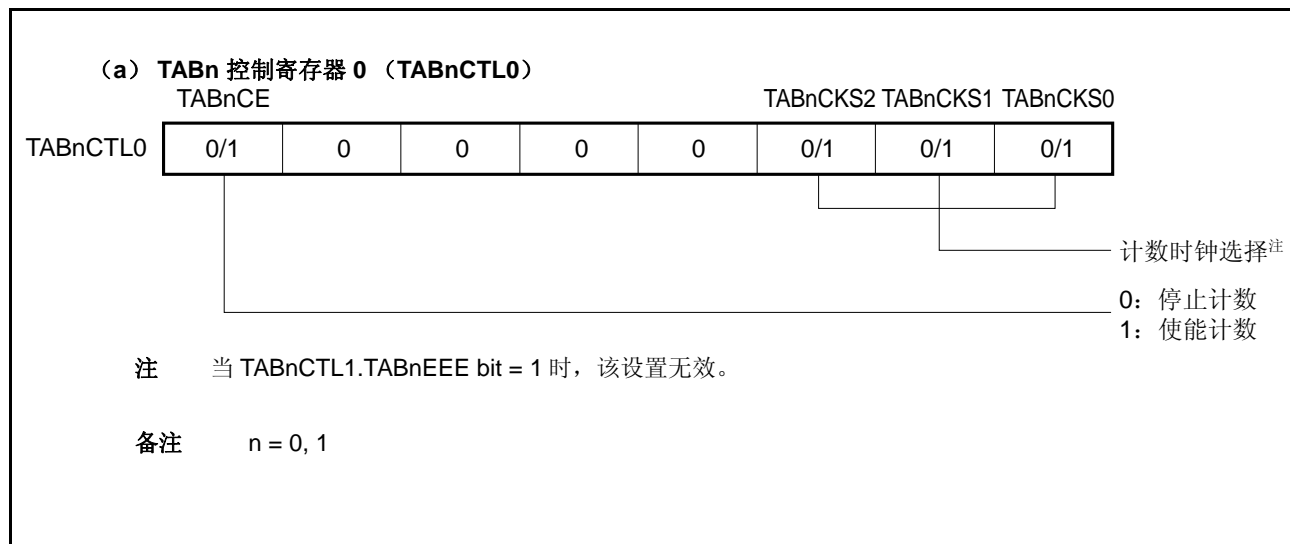


图 8-18. 外部触发脉冲输出模式的寄存器设置 (2/3)

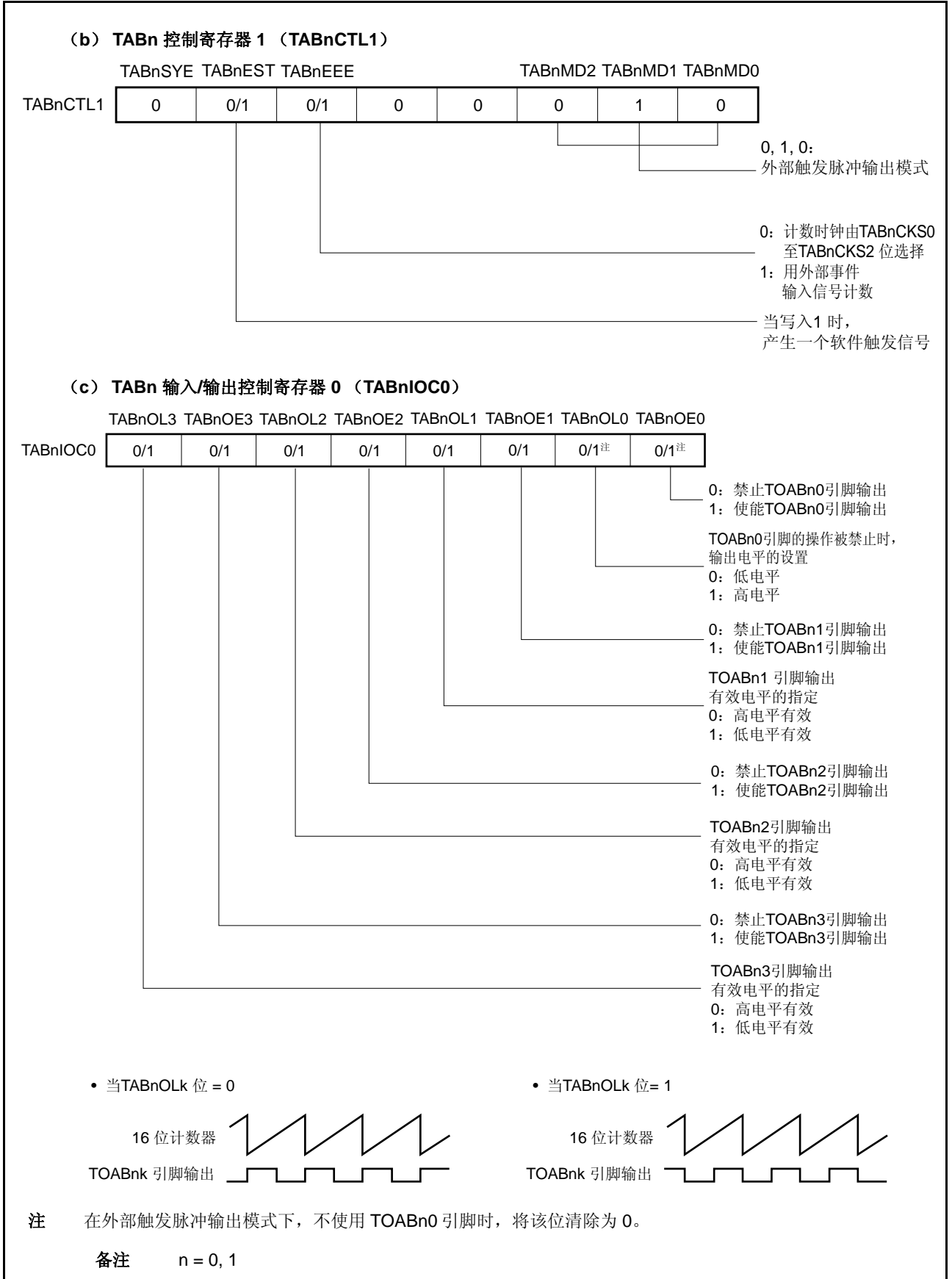
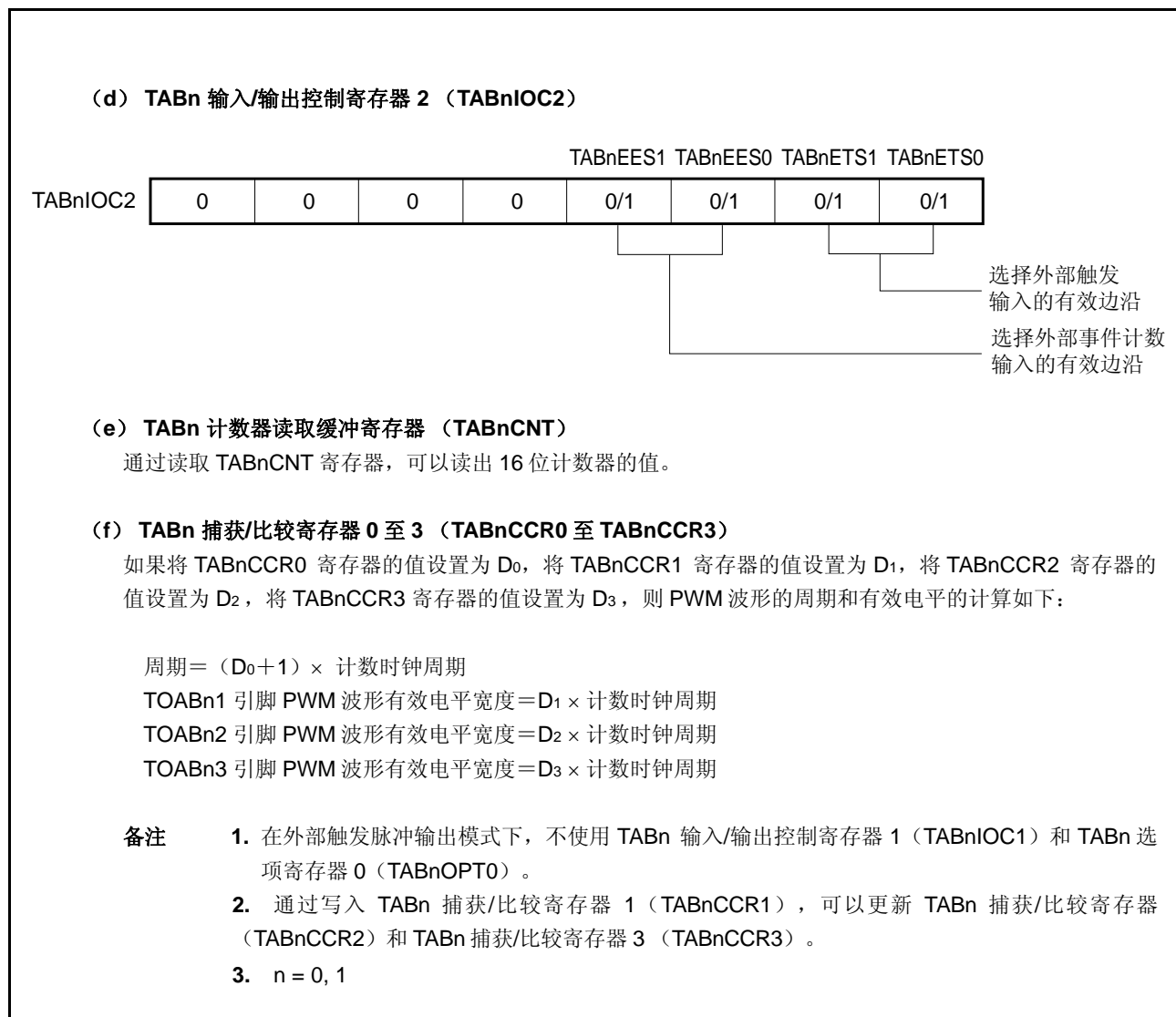


图 8-18. 外部触发脉冲输出模式的寄存器设置 (3/3)



(1) 外部触发脉冲输出模式的软件操作流程

图 8-19. 外部触发脉冲输出模式的软件处理流程 (1/2)

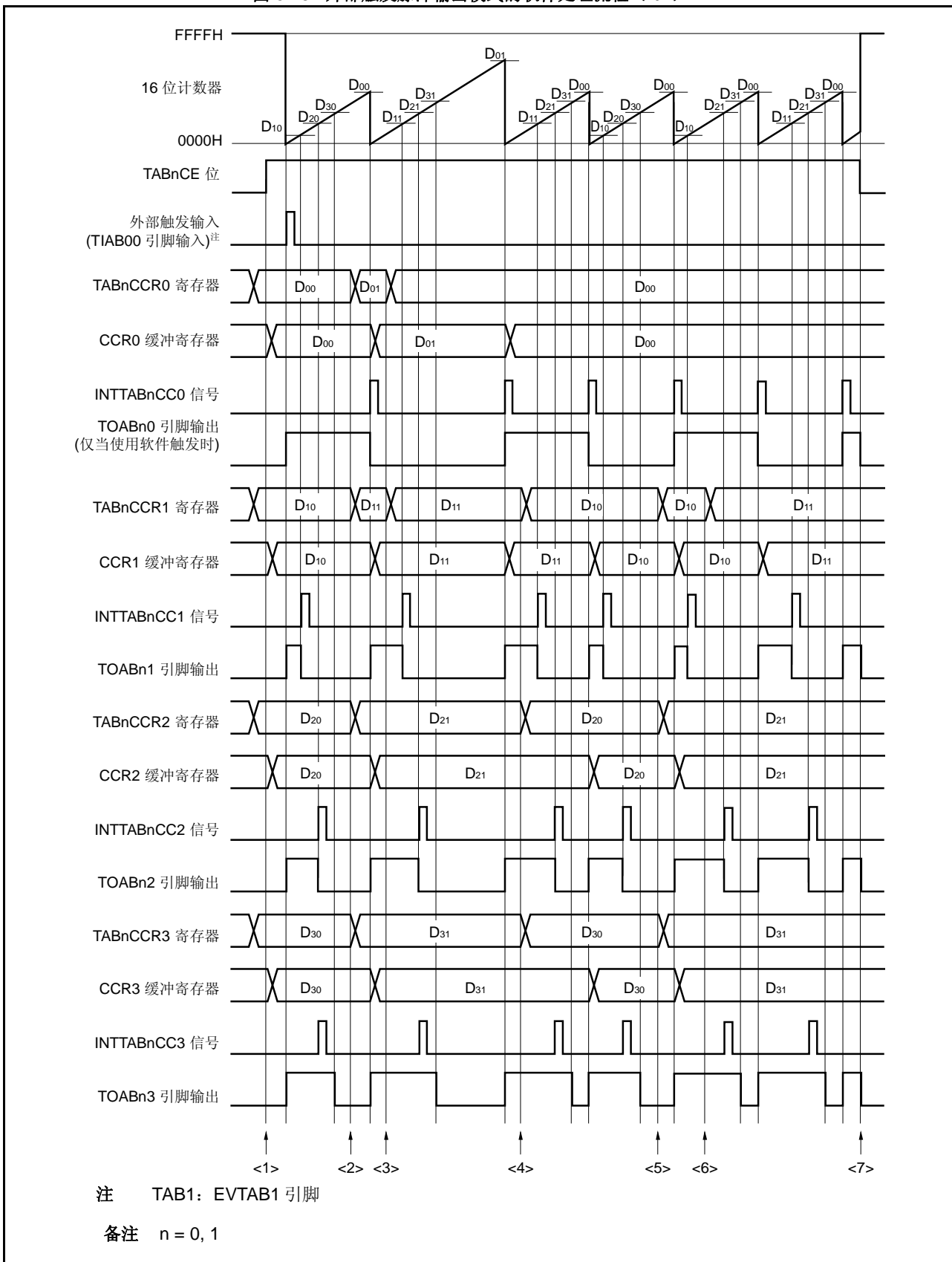
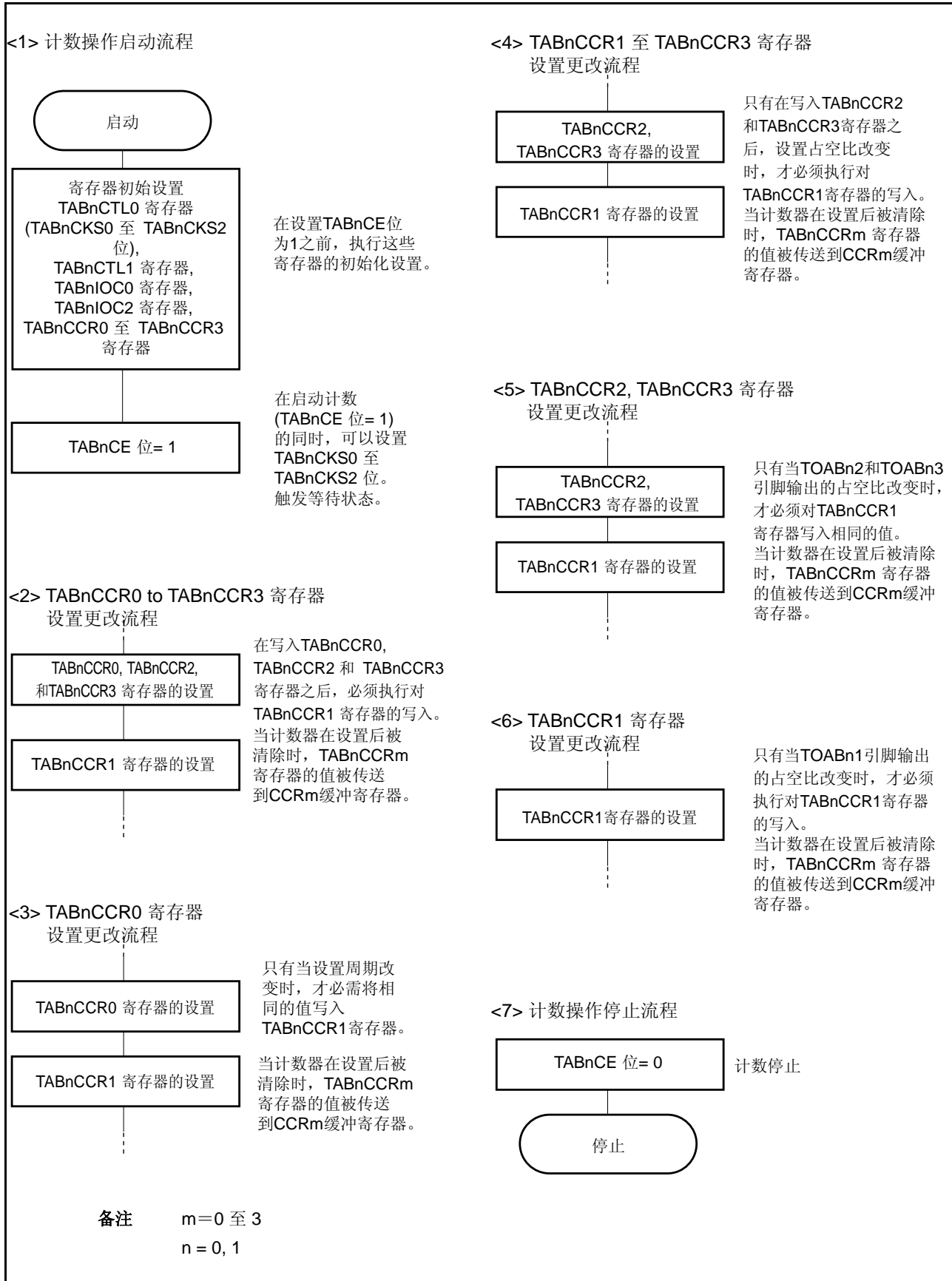


图 8-19. 外部触发脉冲输出模式的软件处理流程 (2/2)

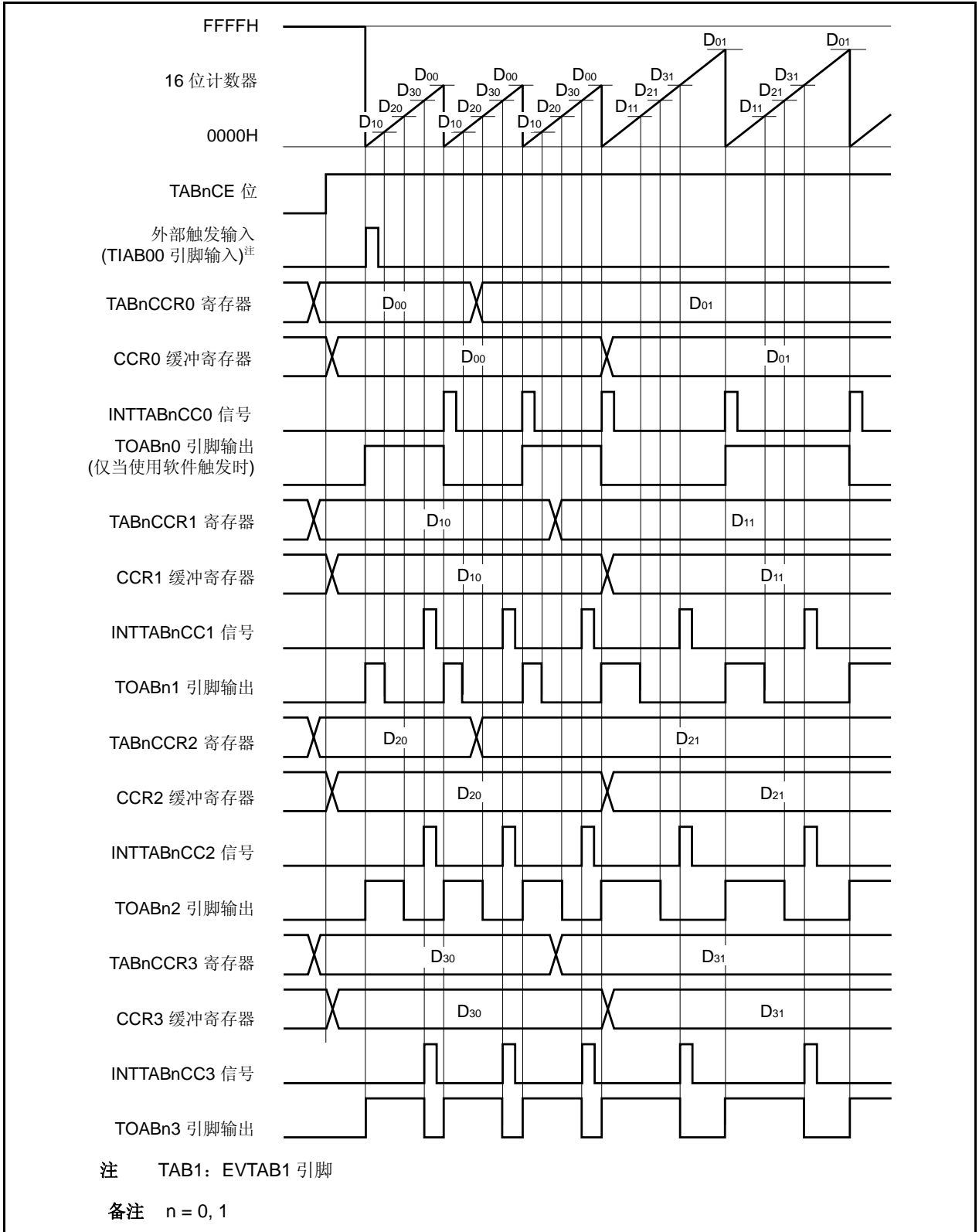


(2) 外部触发脉冲输出模式的操作时序

(a) 有关操作中改变脉冲宽度的注意事项

计数器计数过程中，若要改变 PWM 波形，应重新写入 TABnCCR1 寄存器。

当检测到 INTTABnCC0 信号以后，在 TABnCCR1 寄存器写入操作之后重写 TABnCCRk 寄存器。



为了将数据从 TABnCCRm 寄存器传送到 CCRm 缓冲寄存器，必须写入 TABnCCR1 寄存器。

如果 PWM 波形的周期和有效电平宽度都要改变，先将周期设定到 TABnCCR0 寄存器，将有效电平宽度设定到 TABnCCR2 和 TABnCCR3 寄存器，然后将有效电平设定到 TABnCCR1 寄存器。

如果只改变 PWM 波形的周期，应先把周期设置到 TABnCCR0 寄存器，然后再将相同的值写入 TABnCCR1 寄存器。

如果只改变 PWM 波形的有效电平宽度（占空比系数），需要先向 TABnCCR2 和 TABnCCR3 寄存器设置有效电平，然后再向 TABnCCR1 寄存器设置有效电平。

如果只改变 TOABn1 引脚输出的 PWM 波形的有效电平宽度（占空比系数），仅需要设置 TABnCCR1 寄存器。

如果只改变 TOABn2 和 TOABn3 引脚输出的 PWM 波形的有效电平宽度（占空比系数），要先向 TABnCCR2 和 TABnCCR3 寄存器设置有效电平宽度，然后再将相同的值写入 TABnCCR1 寄存器。

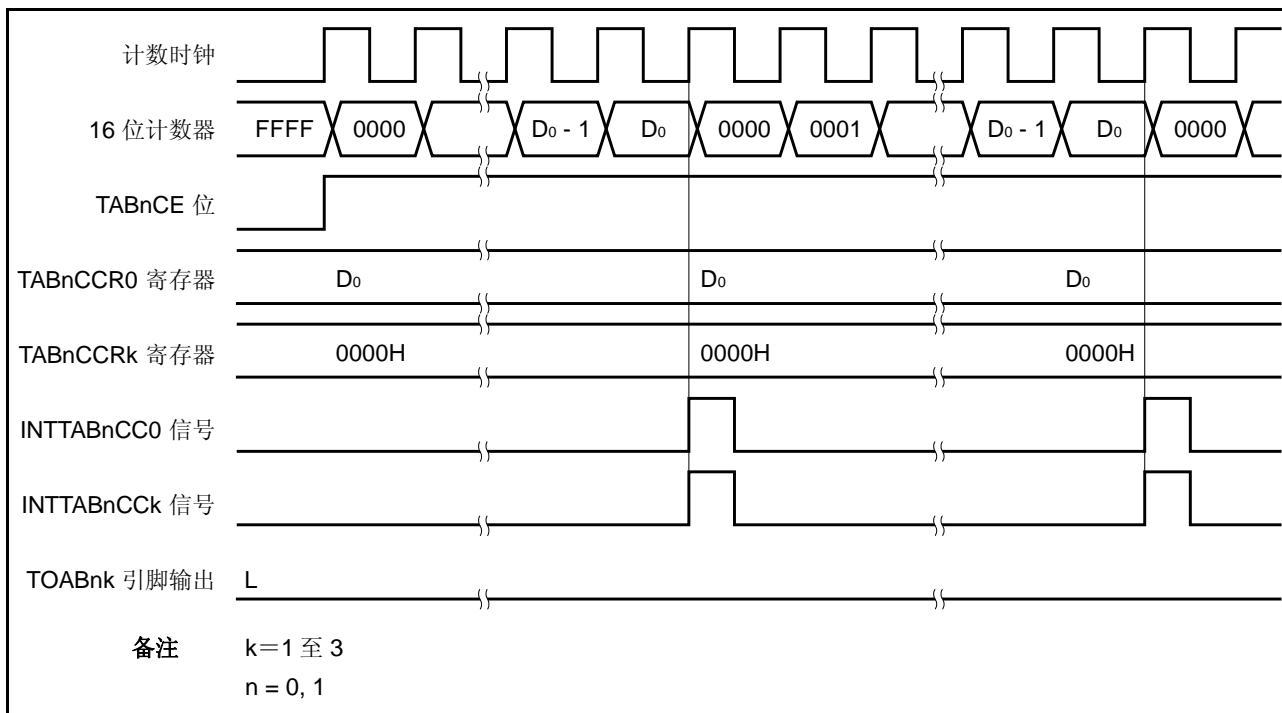
在数据写入 TABnCCR1 寄存器后，在对 16 位计数器清零的同时，写入到 TABnCCRm 寄存器的值会被传送到 CCRm 缓冲寄存器，并作为与 16 位计数器进行比较的值。

在写入 TABnCCR1 寄存器一次之后，要再次写入 TABnCCR0 至 TABnCCR3 寄存器，应该在 INTTABnCC0 信号产生之后进行。否则，CCRm 缓冲寄存器的值可能不确定，因为把数据从 TABnCCRm 寄存器传送到 CCRm 缓冲寄存器的时序会和写入 TABnCCRm 寄存器的时序发生冲突。

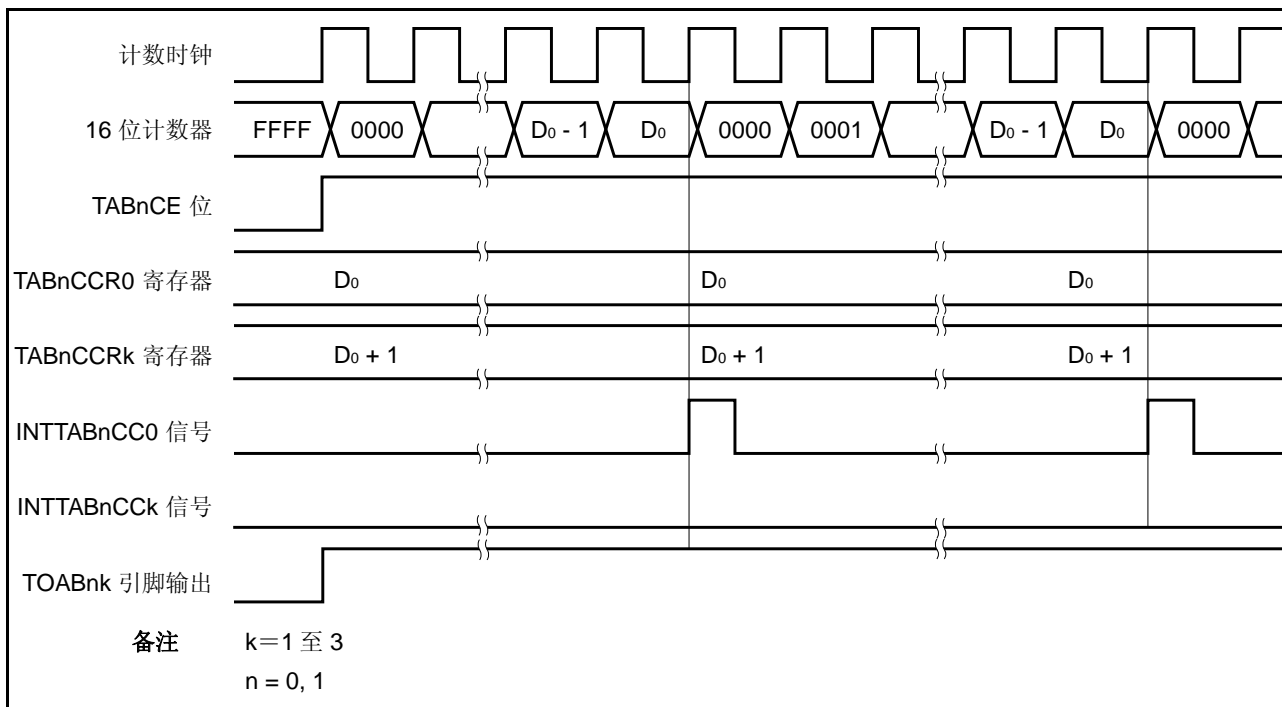
备注 m=0 至 3
 n = 0, 1

(b) PWM 波形的 0%和 100% 输出

为了输出 0%的波形，需要将 $TABnCCRk$ 寄存器设置为 0000H。如果 $TABnCCR0$ 寄存器的设定值为 FFFFH，则会在每个周期产生 $INTTABnCCk$ 中断请求信号。

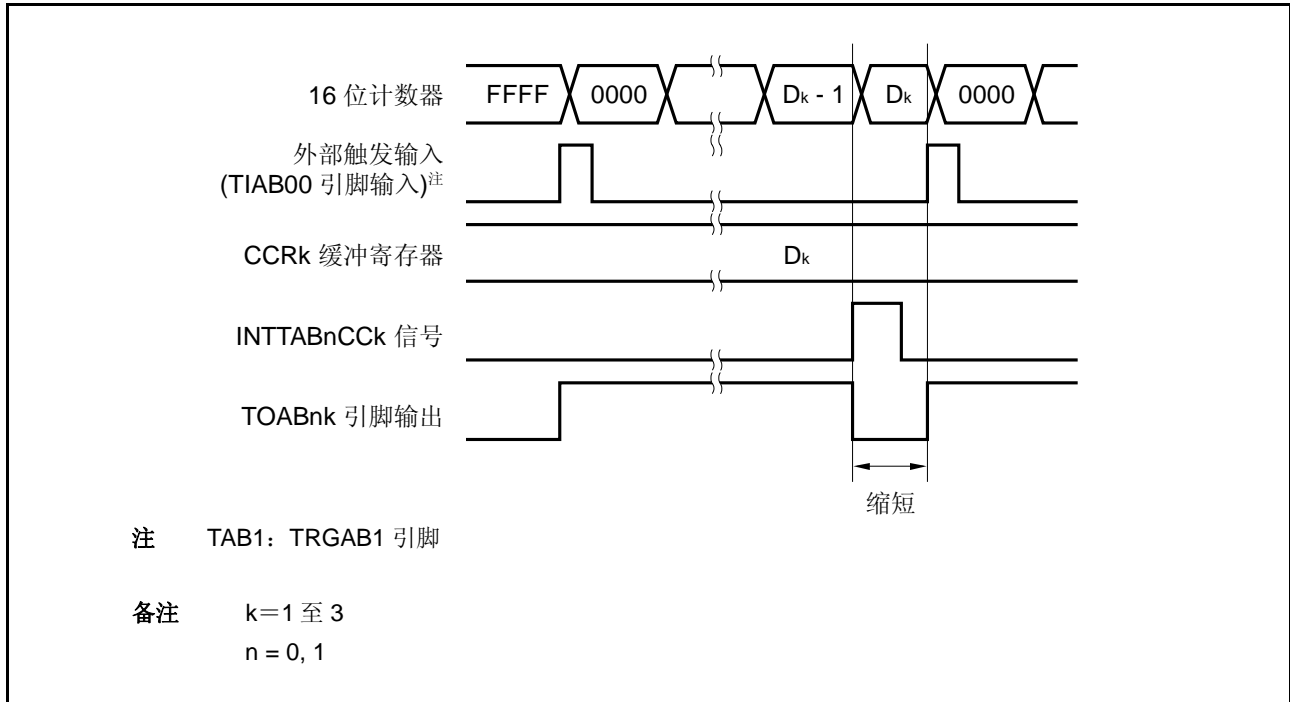


为了输出 100%的波形，需要将 $TABnCCRk$ 寄存器的值设为： $(TABnCCR0$ 寄存器的设定值+1)。如果 $TABnCCR0$ 寄存器的设定值为 FFFFH，则不会产生 100%波形输出。

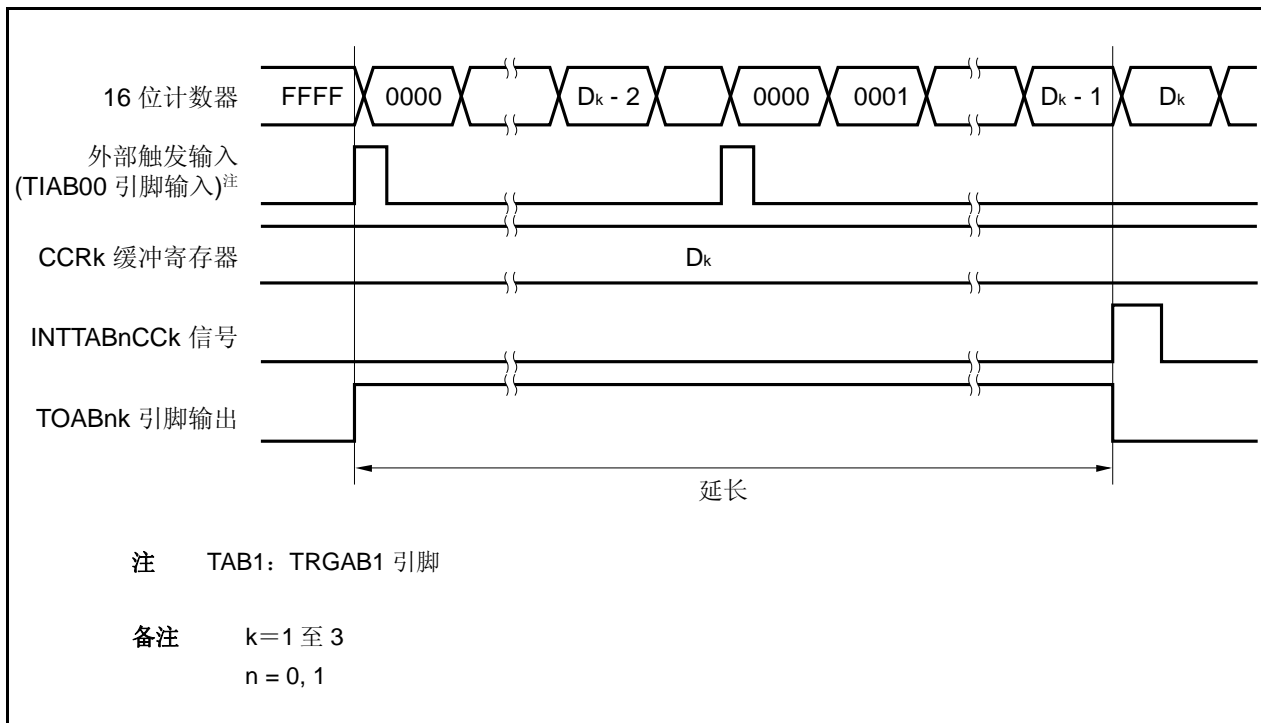


(c) 触发检测和 CCRk 缓冲寄存器匹配之间的冲突

如果在 INTTABnCCk 信号产生后，立刻检测到触发信号，则 16 位计数器立刻清零为 0000H，TOABnk 引脚的输出信号生效，计数器继续计数。结果，PWM 波形的无效时段被缩短。

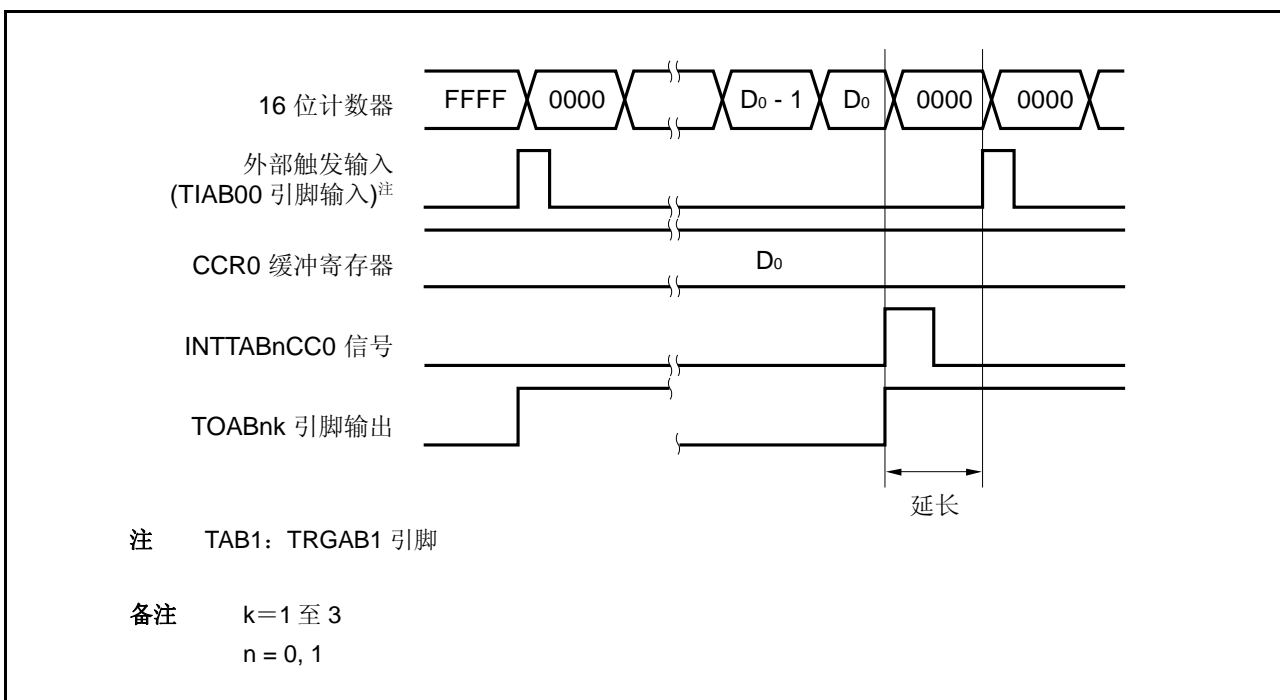


如果紧邻在 INTTABnCCk 信号产生之前检测到触发信号，则不会产生 INTTABnCCk 信号，而 16 位计数器清零为 0000H，计数继续进行。TOABnk 引脚的输出信号仍然保持有效。结果，PWM 波形的有效时段被延长。

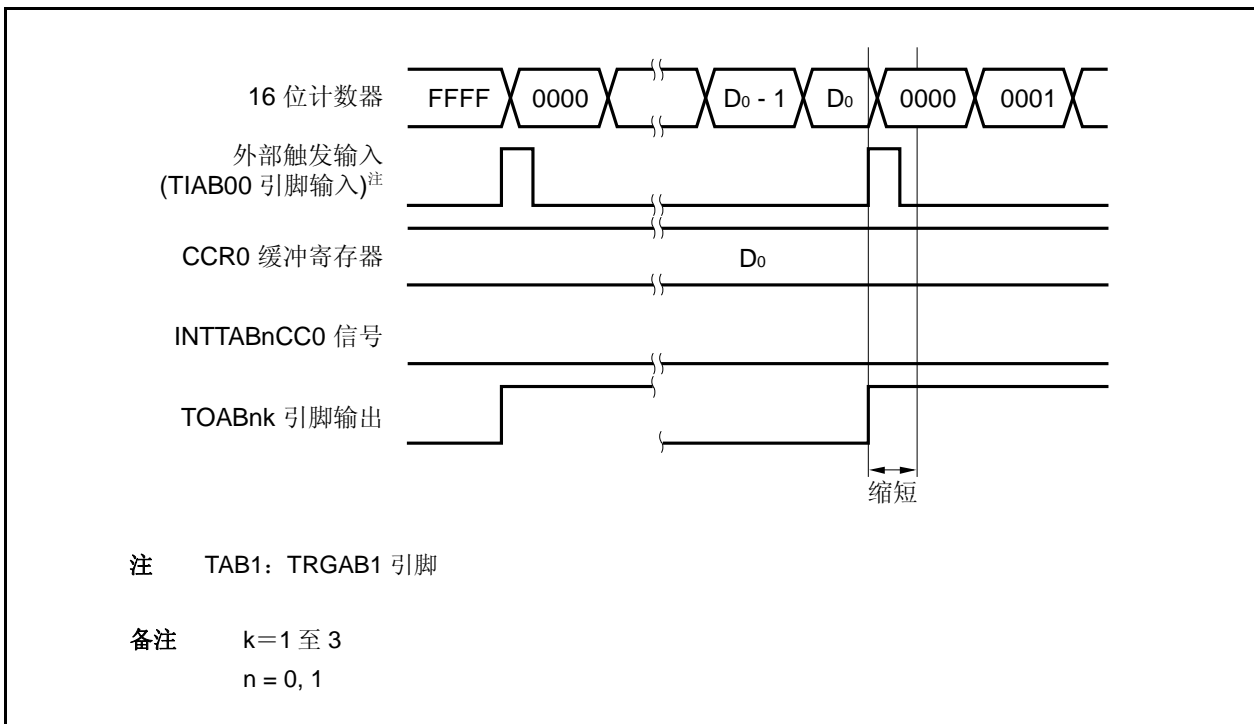


(d) 触发检测和 CCR0 缓冲寄存器匹配之间的冲突

如果在 INTTABnCC0 信号产生后，立刻检测到触发信号，则 16 位计数器立刻清零为 0000H，并继续计数。因此，TOABnk 引脚的有效时段被延长，延长长度为 INTTABnCC0 信号产生时刻至检测到触发信号时刻的时间间隔。

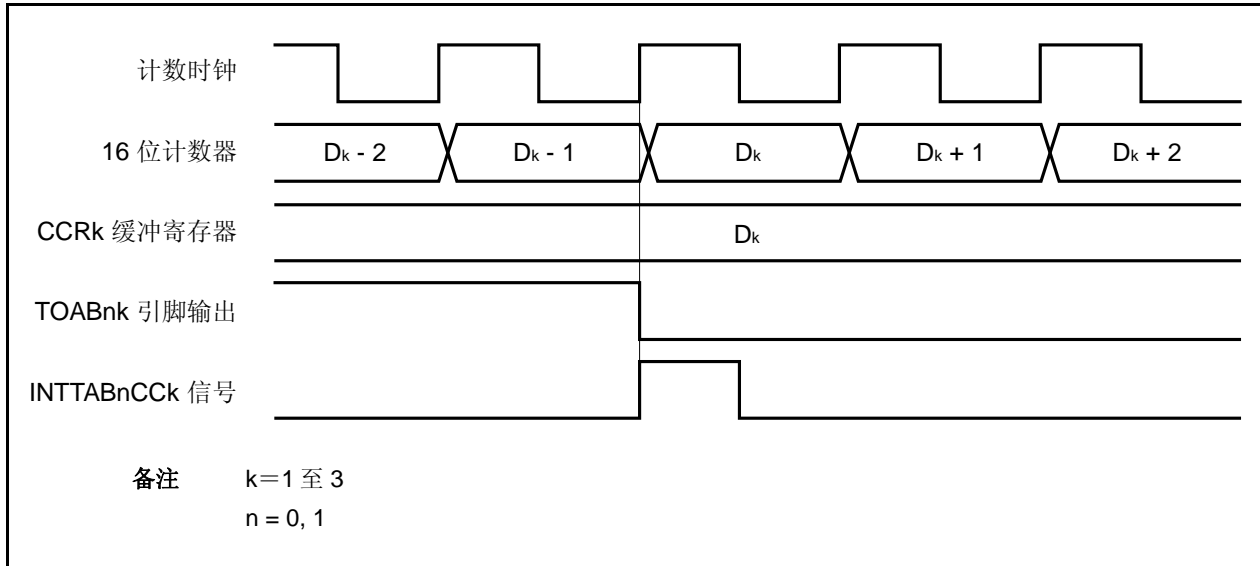


如果紧邻在 INTTABnCC0 信号产生之前检测到触发信号，则不会产生 INTTABnCC0 信号。16 位计数器清零为 0000H，TOABnk 引脚输出有效，计数器继续计数。结果，PWM 波形的无效时段被缩短。



(e) 比较匹配中断请求信号 (INTTABnCCK) 的发生时序

在外部触发脉冲输出模式下，INTTABnCCK 信号的发生时序与其它情况下 INTTABnCCK 信号的发生时序不同；当 16 位计数器的计数值与 CCRk 缓冲寄存器的值匹配时，产生 INTTABnCCK 中断请求信号。



通常在 16 位计数器的当前计数值与 CCRk 缓冲寄存器的值匹配之后，在下一次计数时同步发生 INTTABnCCK 信号。

但是，在外部触发脉冲输出模式下，INTTABnCCK 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOABnk 引脚输出信号变化的时序。

8.5.4 单次触发脉冲输出模式 (TABnMD2 至 TABnMD0 位 = 011)

在单次触发脉冲输出模式下, 当 TABnCTL0.TABnCE 位被置为 1 时, TABn 等待触发。在检测到外部触发输入的有效边沿时, TABn 开始计数, 同时从 TOABn1 至 TOABn3 引脚输出一个单次触发脉冲。

除外部触发外, 也可以生成软件触发信号来输出该脉冲。在使用软件触发时, TOABn0 引脚在 16 位计数器计数时输出有效电平, 在计数器停止 (等待触发) 时输出无效电平。

图 8-20. 单次触发脉冲输出模式的配置

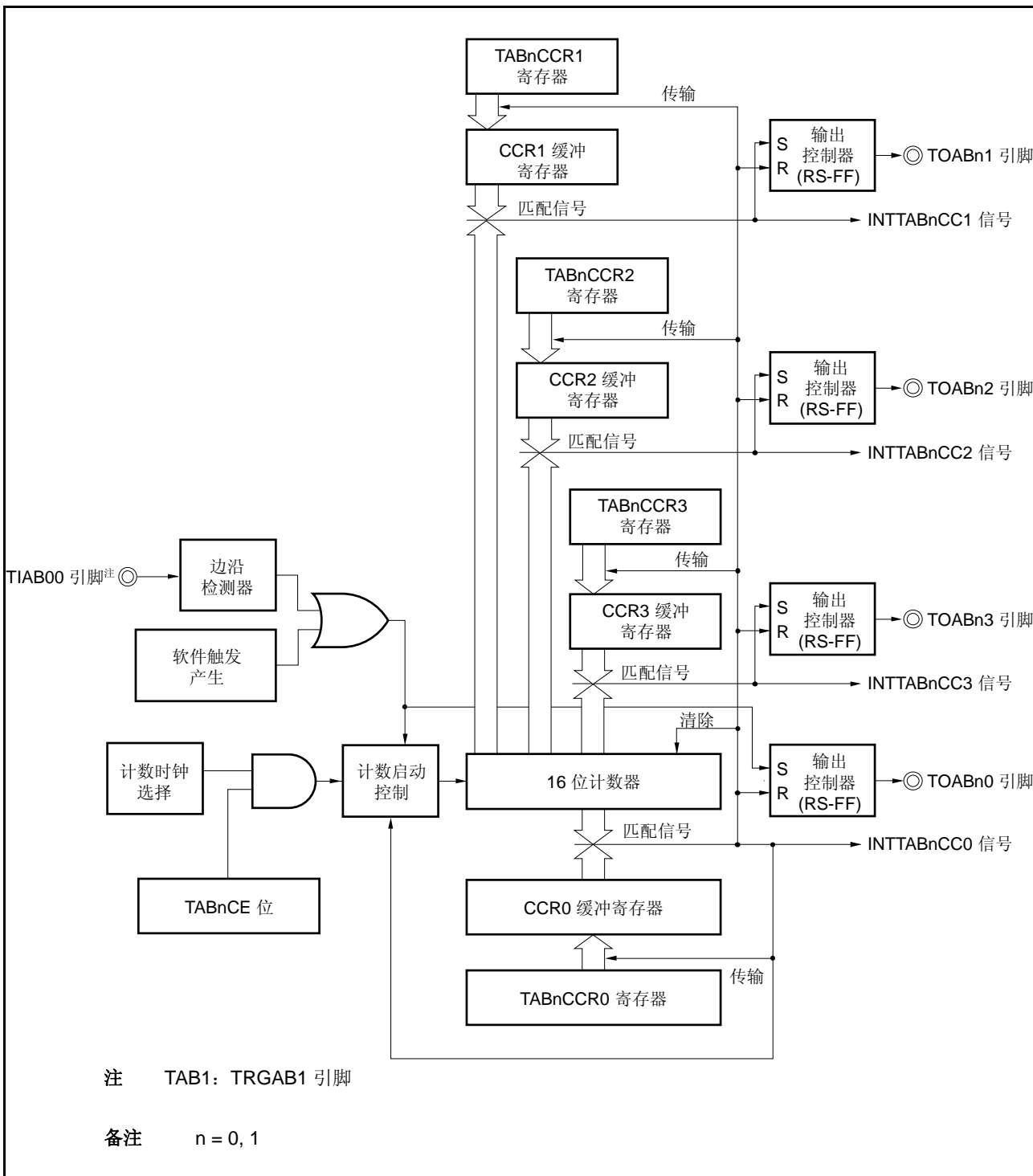
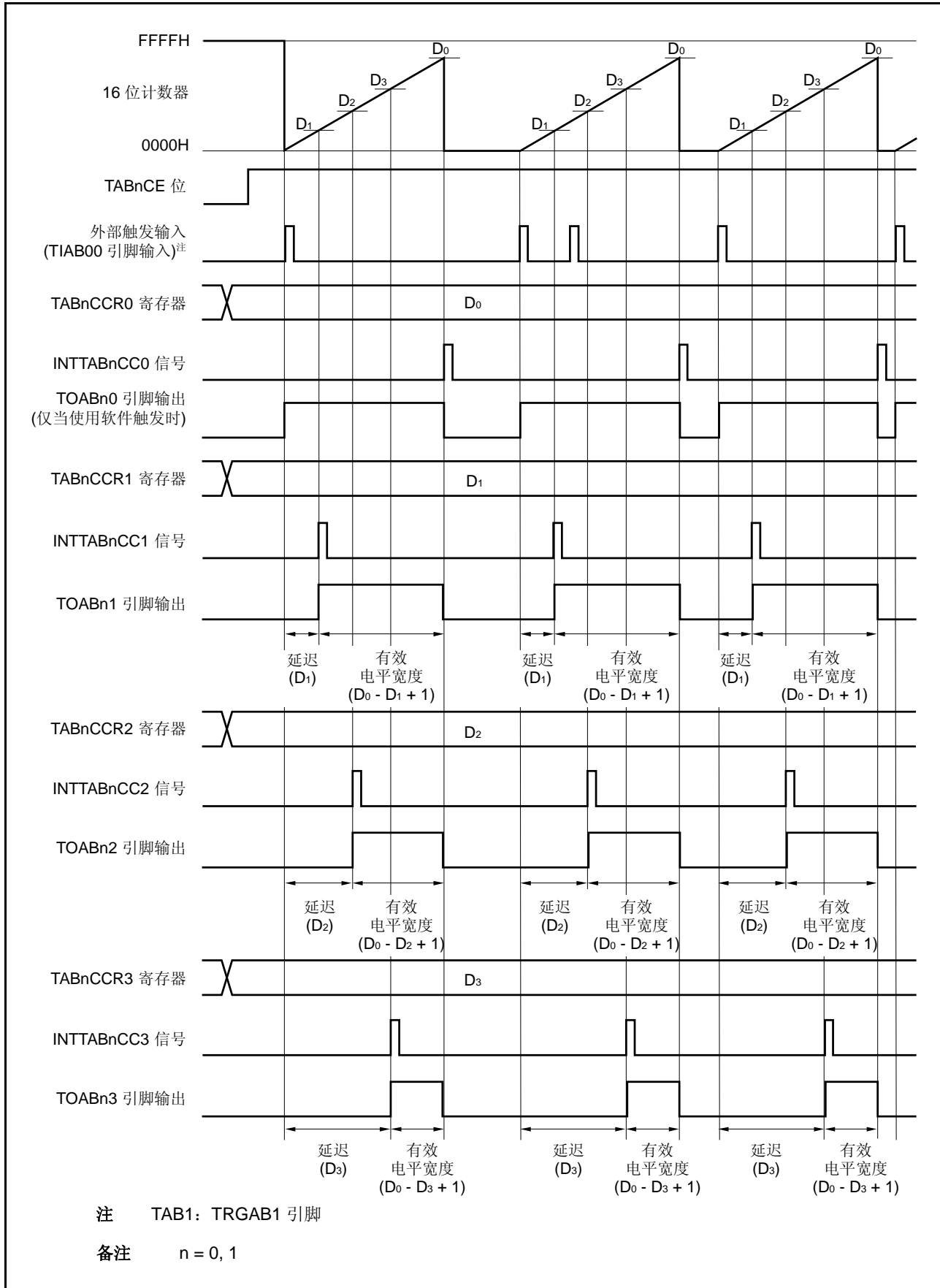


图 8-21. 单次触发脉冲输出模式的基本时序



当 TABnCE 位置为 1 时, TABn 等待触发。在触发信号发生时, 16 位计数器从 FFFFH 清零为 0000H, 开始进行计数, 并从 TOABnk 引脚输出一个单次触发脉冲。在输出单次触发脉冲后, 16 位计数器被设置为 FFFFH, 计数停止, 并重新等待触发。如果在单次触发脉冲输出期间再次产生触发信号, 则该触发信号被忽略。

单次触发脉冲的输出延迟时间和有效电平宽度可计算如下:

$$\text{输出延迟时间} = (\text{TABnCCRk 寄存器的设定值}) \times \text{计数时钟周期}$$

$$\text{有效电平宽度} = (\text{TABnCCR0 寄存器的设定值} - \text{TABnCCRk 寄存器的设定值} + 1) \times \text{计数时钟周期}$$

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配之后, 产生比较匹配中断请求信号 INTTABnCC0。当 16 位计数器的计数值与 CCRk 缓冲寄存器的值匹配时, 会生成比较匹配中断请求信号 INTTABnCCK。

外部触发输入信号的有效边沿或将软件触发 (TABnCTL1.TABnEST 位) 信号设置为 1, 都可被用作触发信号。

备注 k=1 至 3
n = 0, 1

图 8-22. 单次触发脉冲输出模式时的寄存器设置 (1/3)

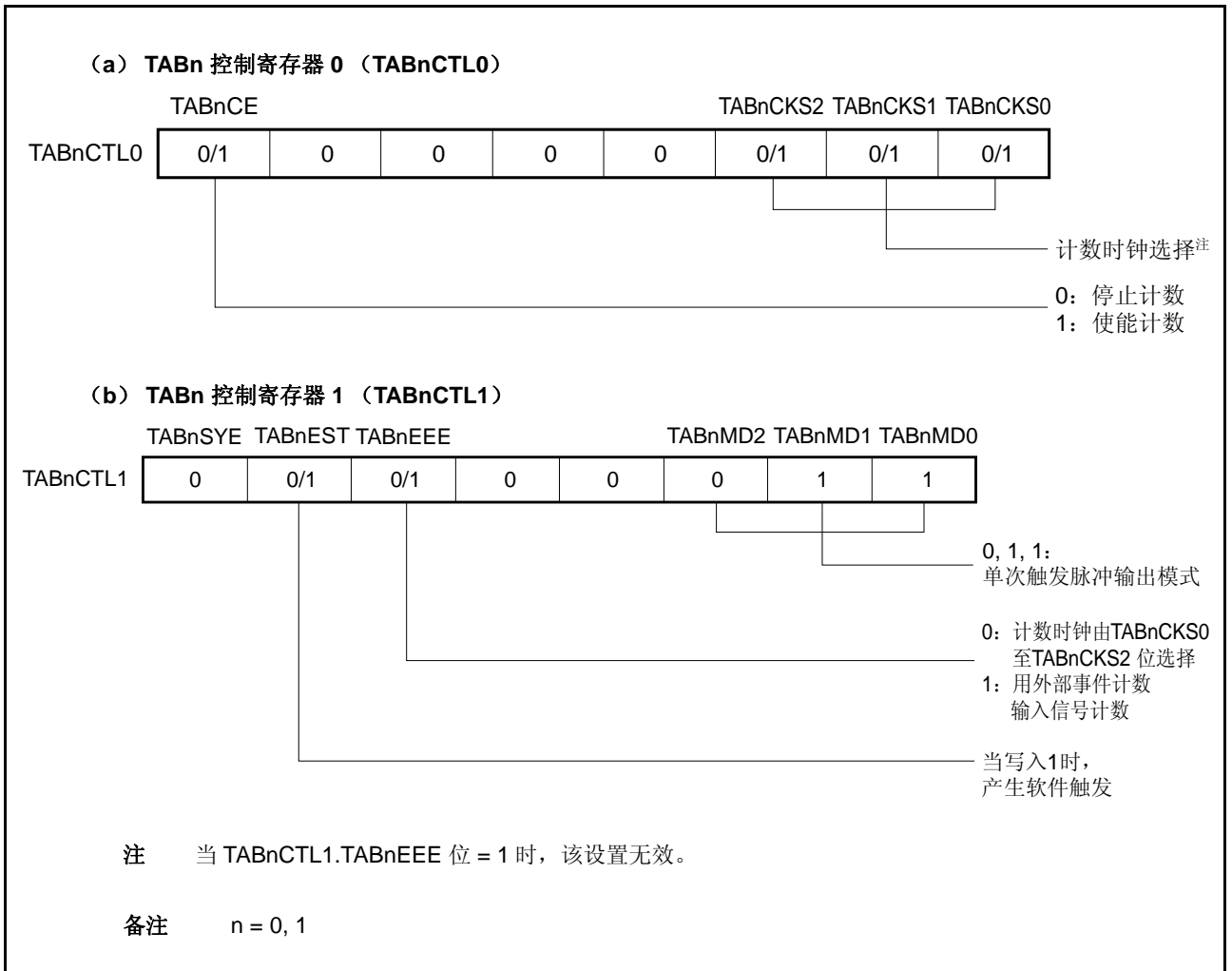


图 8-22. 单次触发脉冲输出模式时的寄存器设置 (2/3)

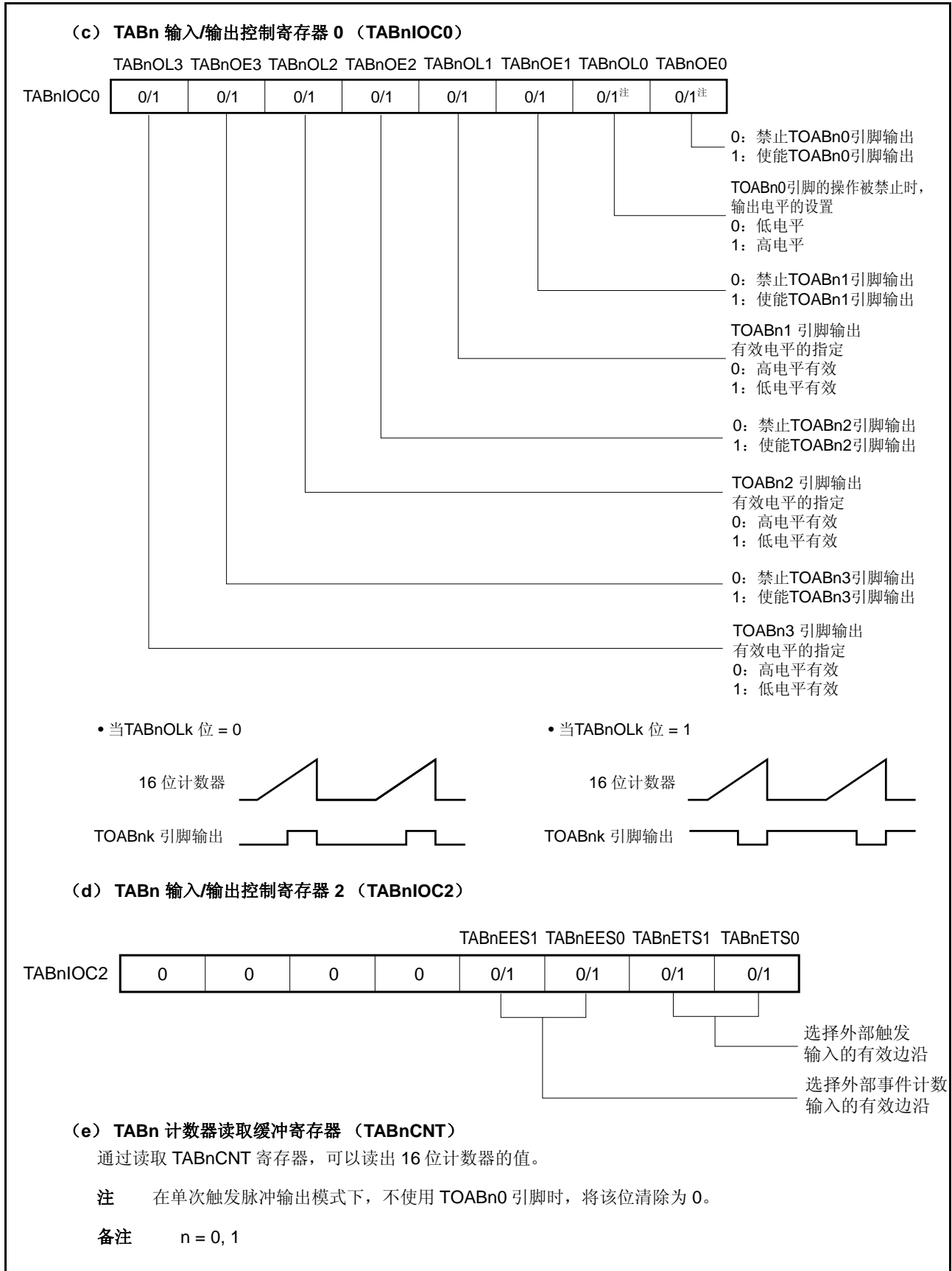


图 8-22. 单次触发脉冲输出模式时的寄存器设置 (3/3)

(f) TABn 捕获/比较寄存器 0 至 3 (TABnCCR0 至 TABnCCR3)

若把 TABnCCR0 寄存器的值设置为 D_0 , TABnCCRk 寄存器的值设置为 D_k , 则单次触发脉冲的有效电平宽度和输出延迟时间的计算如下。

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 TABnCCRk 寄存器中设置的值大于 TABnCCR0 寄存器中设置的值, 那么, 即使在单次触发脉冲输出模式下也不会输出单次触发脉冲。

备注

1. 在单次触发脉冲输出模式下, 不使用 TABn 输入/输出控制寄存器 1 (TABnIOC1) 和 TABn 选项寄存器 0 (TABnOPT0)。
2. $k=1$ 至 3
3. $n=0, 1$

(1) 单次触发脉冲输出模式下的操作流程

图 8-23. 单次触发脉冲输出模式的软件处理流程 (1/2)

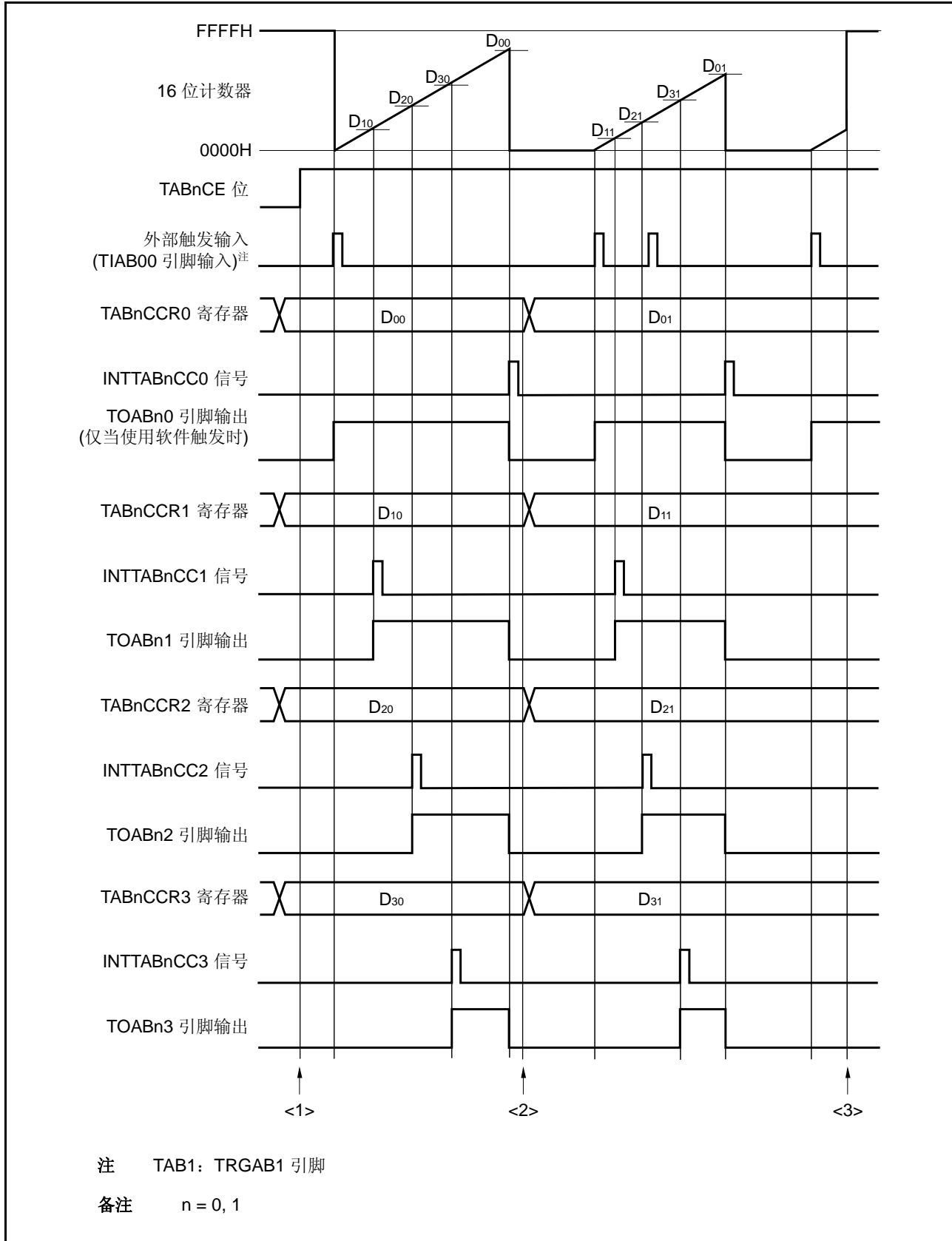
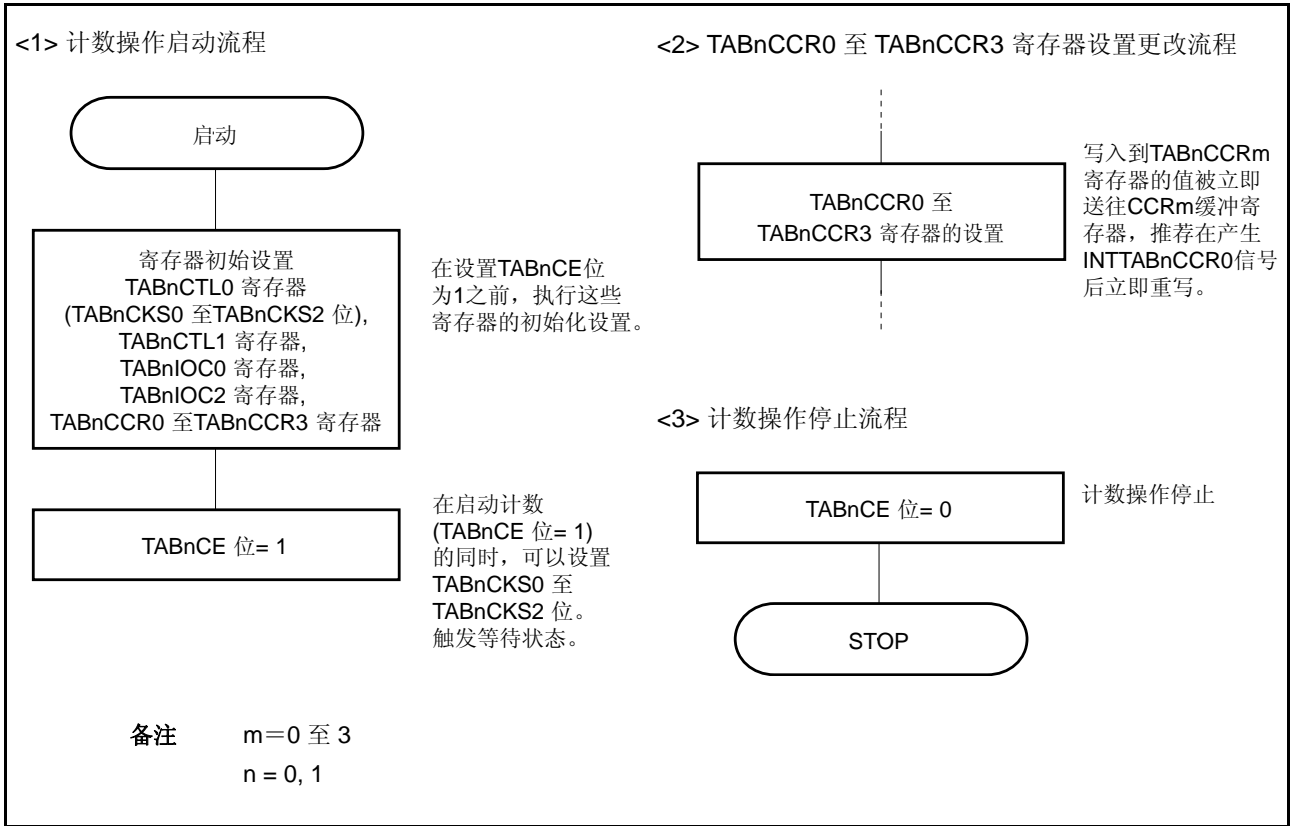


图 8-23. 单次触发脉冲输出模式的软件处理流程 (2/2)

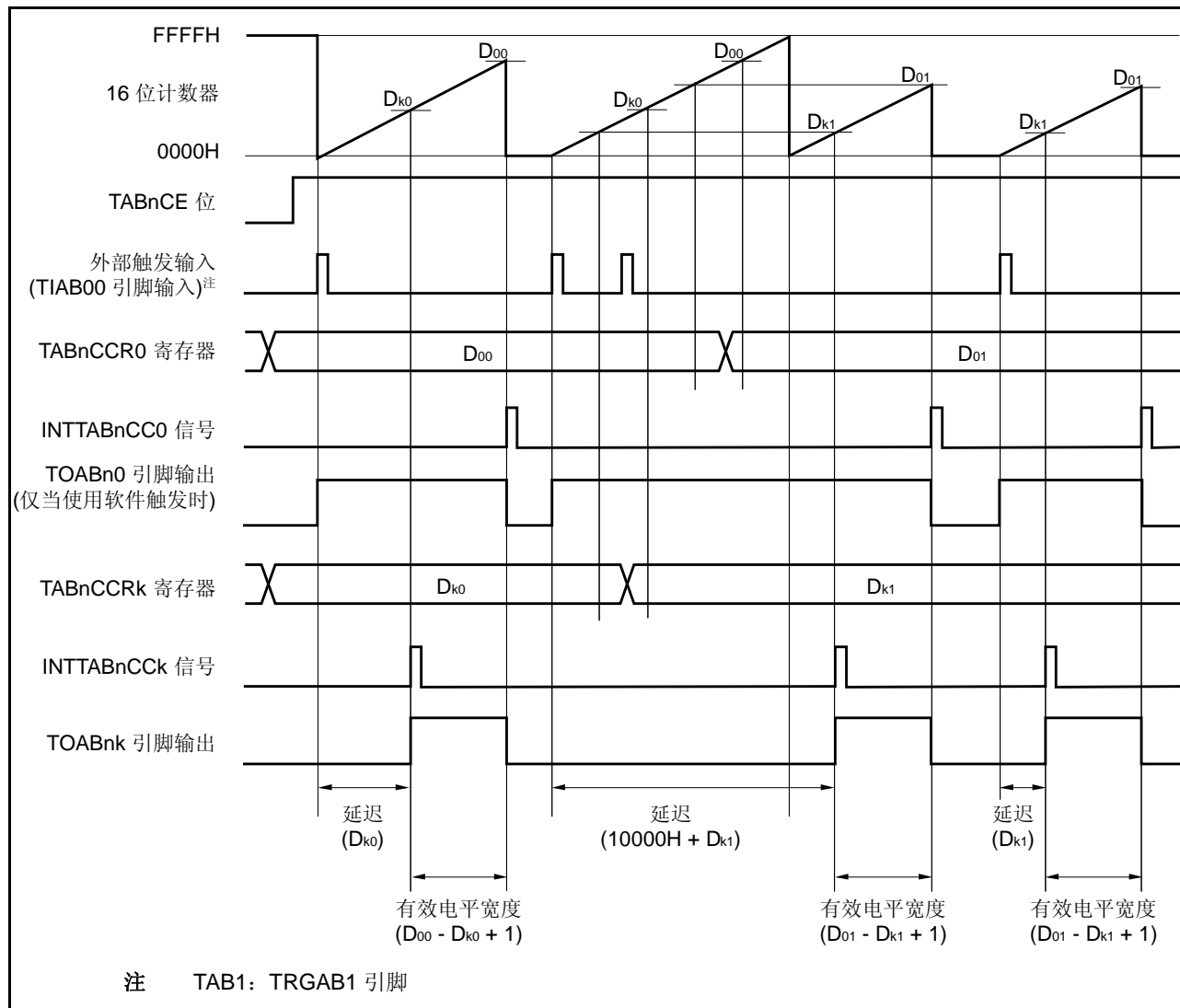


(2) 单次触发脉冲输出模式时的操作时序

(a) 重写 TABnCCRm 寄存器的注意事项

要将 TABnCCRm 寄存器的值改为比当前值较小的值，应停止计数，然后改变其设定值。

如果在计数器工作期间，将 TABnCCR0 寄存器的值重写为比当前值较小的值，则 16 位计数器可能会溢出。



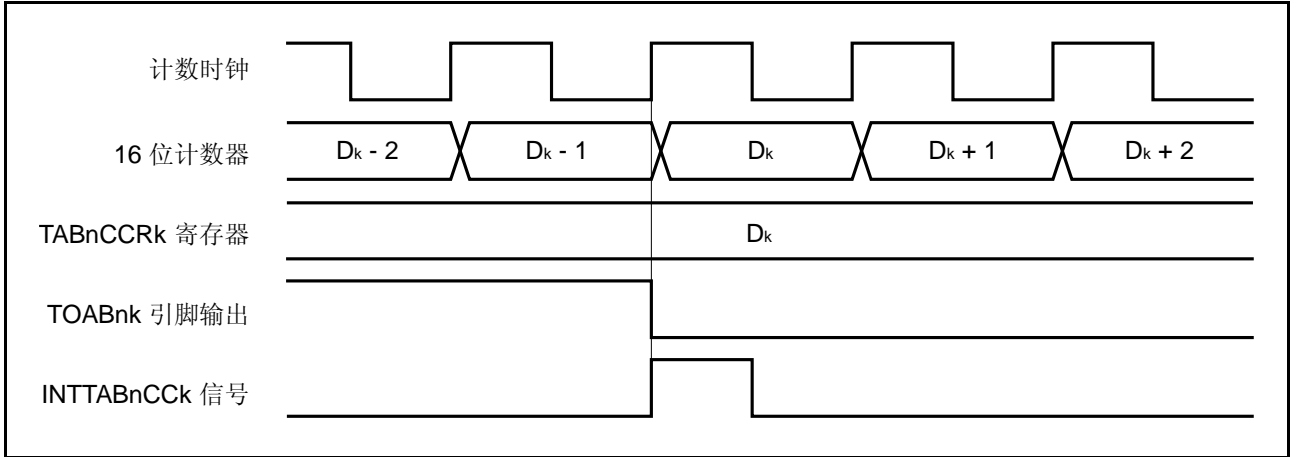
假设 $D_{00} > D_{01}$ 而且 $D_{k0} > D_{k1}$ ，当 TABnCCR0 寄存器由 D_{00} 重写为 D_{01} ，TABnCCRk 寄存器由 D_{k0} 重写为 D_{k1} 时，如果在 16 位计数器的计数值大于 D_{k1} 且小于 D_{k0} 时重写 TABnCCRk 寄存器，并且在计数值大于 D_{01} 且小于 D_{00} 时重写 TABnCCR0 寄存器，则一旦重写寄存器，就会在各设置值上反映出来，并立即与计数值进行比较。计数器向上计数到 FFFFH 后，从 0000H 开始重新向上计数。当计数值与 D_{k1} 匹配时，计数器会产生 INTTABnCCK 信号并使 TOABnk 引脚有效。当计数值与 D_{01} 匹配时，计数器产生 INTTABnCC0 信号，使 TOABnk 引脚失效，并停止计数。

因此，计数器输出脉冲的延迟时段或有效时段可能会与原先预计的单次触发脉冲有所不同。

备注 k=1 至 3
n=0, 1

(b) 比较匹配中断请求信号 (INTTABnCCk) 的发生时序

单次触发脉冲输出模式下，比较匹配中断请求信号 (INTTABnCCk) 的发生时序和其它情况下的 INTTABnCCk 信号发生时序不同，当 16 位计数器的计数值与 TABnCCRk 寄存器的值匹配时，产生 INTTABnCCk 中断请求信号。



通常在 16 位计数器的当前计数值与 TABnCCRk 寄存器的值匹配之后，在下次计数时同步发生 INTTABnCCk 信号。

但是，在单次触发脉冲输出模式下，INTTP0CC1 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOABnk 引脚输出信号变化的时序。

备注 $k=1$ 至 3
 $n=0, 1$

8.5.5 PWM输出模式 (TABnMD2 至TABnMD0 位=100)

在 PWM 输出模式下, 当 TABnCTL0.TABnCE 位被置为 1 时, PWM 波形从 TOABn1 至 TOABn3 引脚输出。而且, 由 TOABn0 引脚输出的脉冲, 其半周期等于 PWM 的全周期。

图 8-24. PWM 输出模式的配置

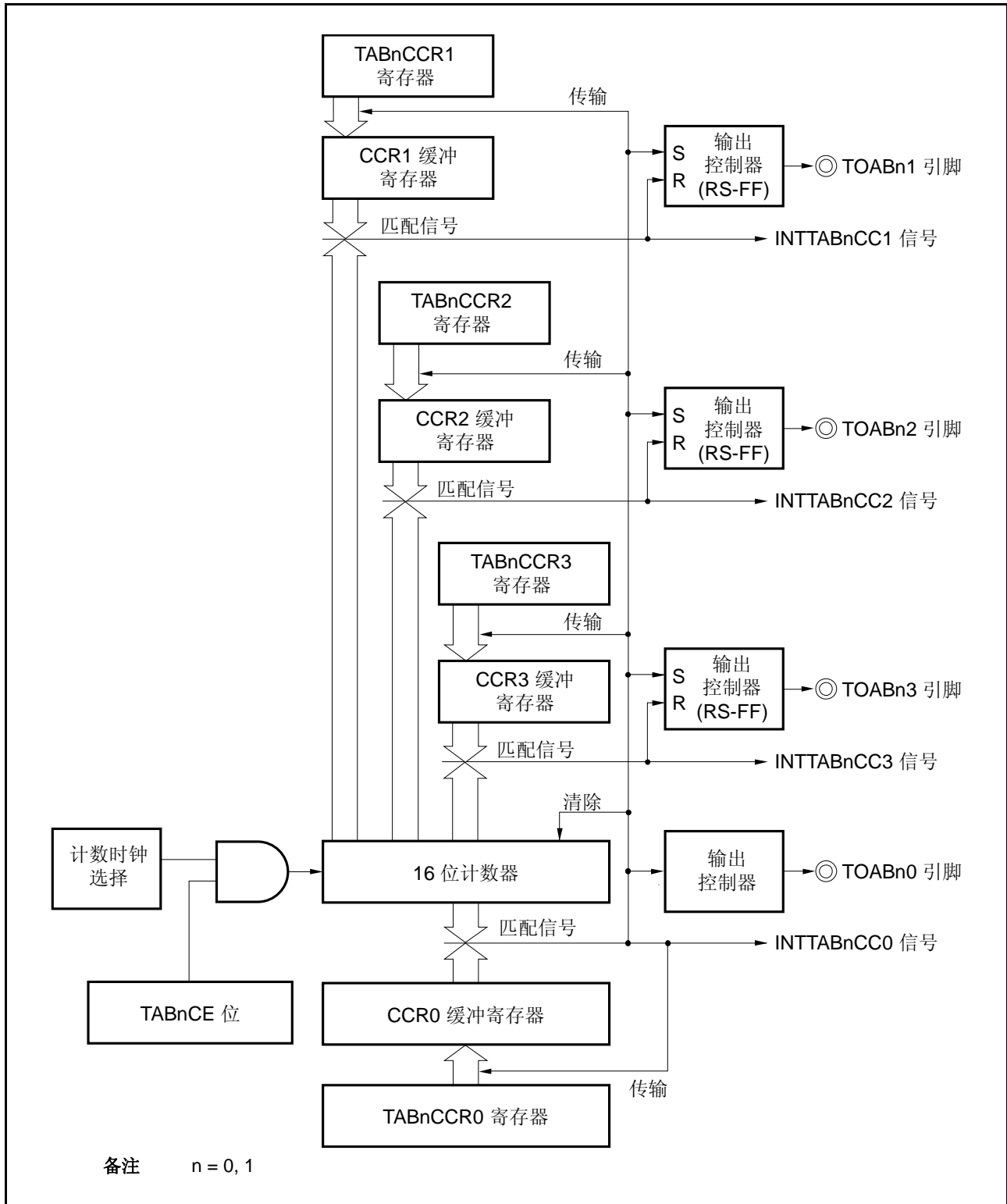
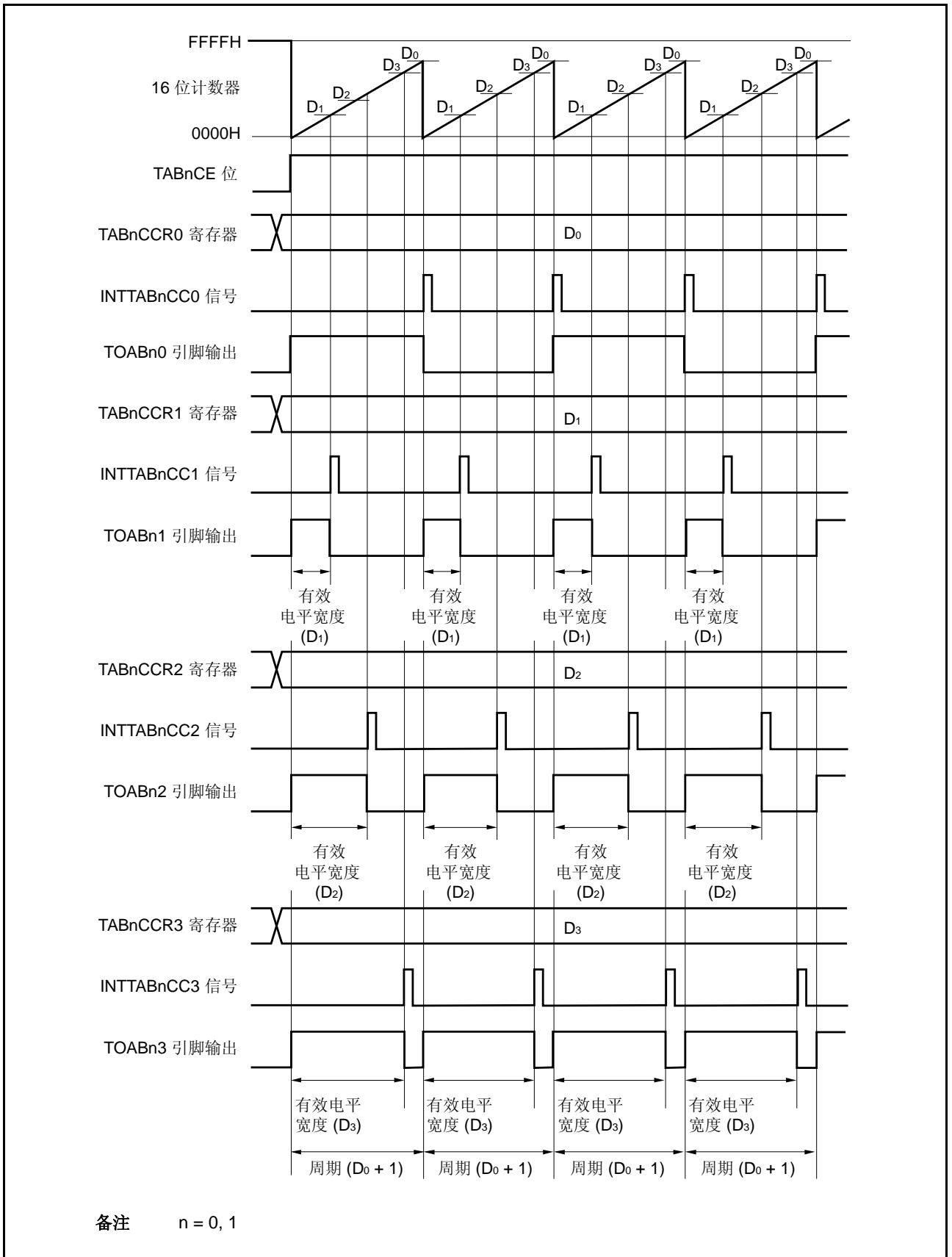


图 8-25. PWM 输出模式的基本时序



当 TABnCE 位置为 1 时，16 位计数器从 FFFFH 清零为 0000H，计数开始，同时从 TOABnk 引脚输出 PWM 波形。PWM 波形的有效电平宽度、周期、占空比系数的计算如下。

有效电平宽度 = (TABnCCRk 寄存器的设定值) × 计数时钟周期

周期 = (TABnCCR0 寄存器的设定值 + 1) × 计数时钟周期

占空比系数 = (TABnCCRk 寄存器的设定值) / (TABnCCR0 寄存器的设定值 + 1)

在计数操作时，可通过重写 TABnCCRm 寄存器来改变 PWM 波形。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配时，16 位计数器清零为 0000H 时，新写入的值会发生作用。

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配之后，进行下一次计数时，会同步产生比较匹配中断请求信号 INTTABnCC0，且 16 位计数器清零为 0000H。当 16 位计数器的计数值与 CCRk 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTABnCk。

备注 k=1 至 3
m=0 至 3
n=0, 1

图 8-26. PWM 输出模式时的寄存器设置 (1/3)

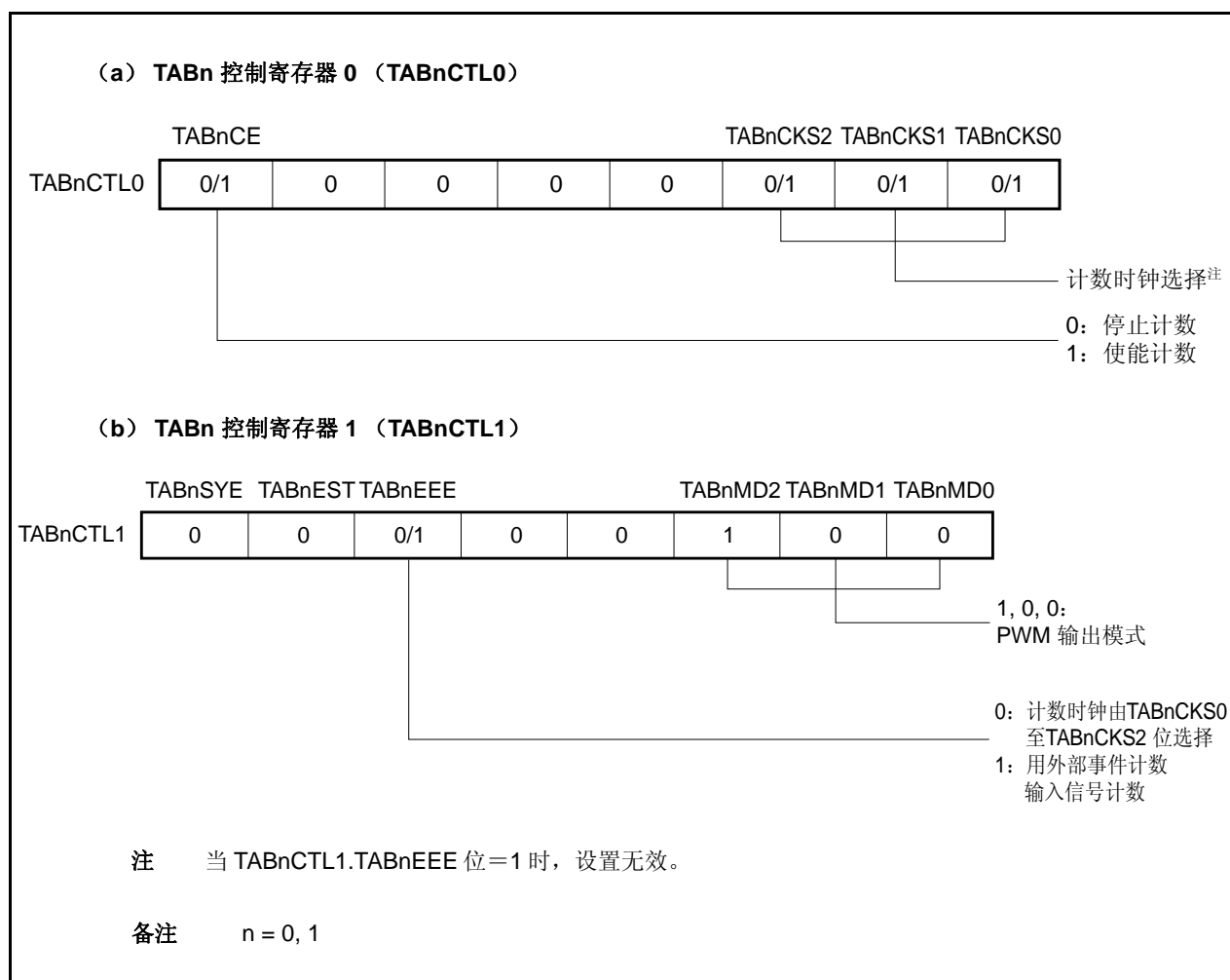


图 8-26. PWM 输出模式时的寄存器设置 (2/3)

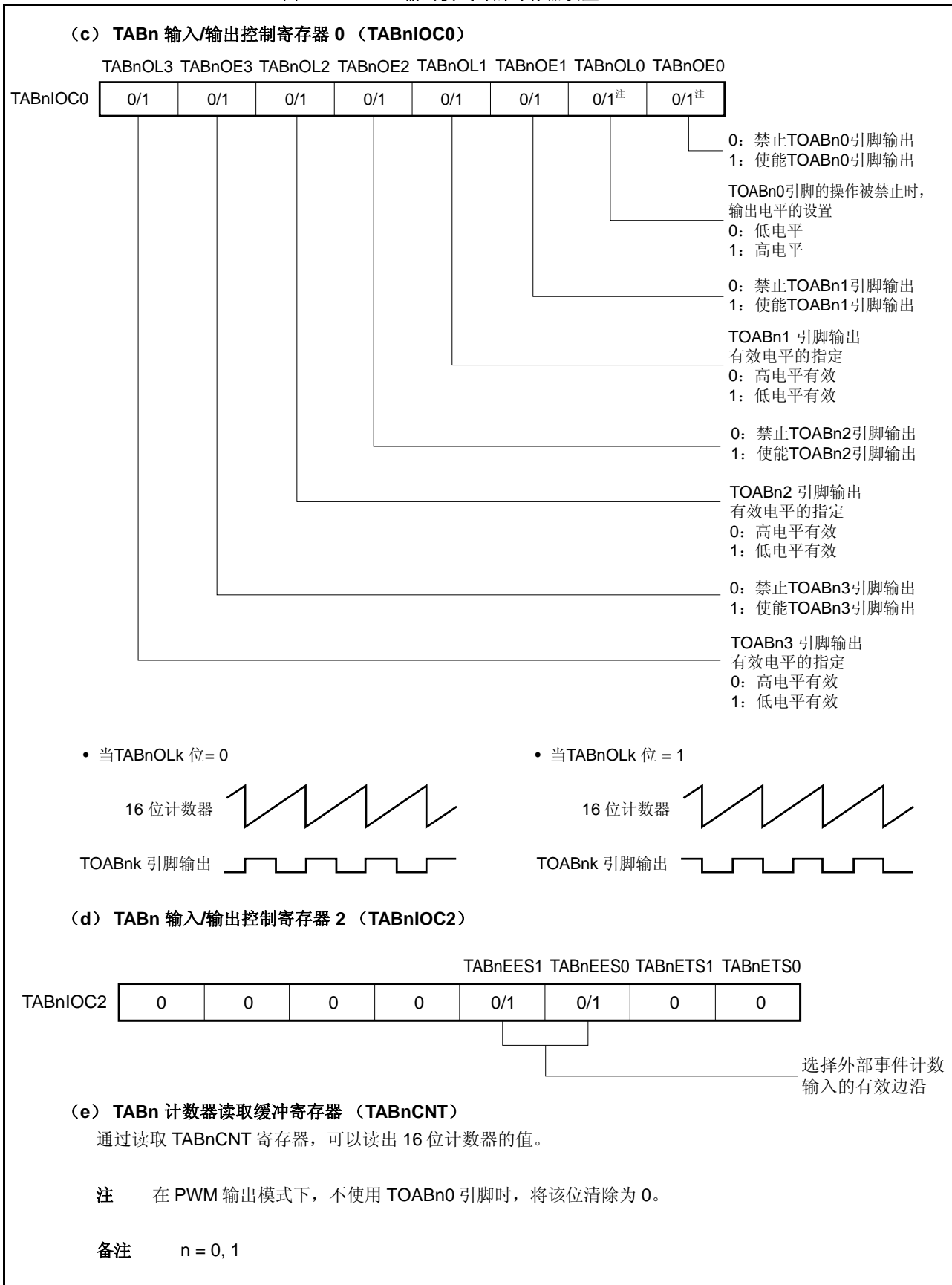


图 8-26. PWM 输出模式时的寄存器设置 (3/3)

(f) TABn 捕获/比较寄存器 0 至 3 (TABnCCR0 至 TABnCCR3)

如果将 TABnCCR0 寄存器的值设置为 D_0 ，将 TABnCCRk 寄存器的值设置为 D_k ，则 PWM 波形的周期和有效电平的计算如下：

$$\text{周期} = (D_0 + 1) \times \text{计数时钟周期}$$

$$\text{有效电平宽度} = D_k \times \text{计数时钟周期}$$

- 备注**
1. 在 PWM 输出模式下，不使用 TABn 输入/输出控制寄存器 1 (TABnIOC1) 和 TABn 选项寄存器 0 (TABnOPT0)。
 2. 通过写入 TABn 捕获/比较寄存器 1 (TABnCCR1)，可以更新 TABn 捕获/比较寄存器 (TABnCCR2) 和 TABn 捕获/比较寄存器 3 (TABnCCR3)。
 3. $n = 0, 1$

(1) PWM 输出模式下的操作流程

图 8-27. PWM 输出模式的软件处理流程 (1/2)

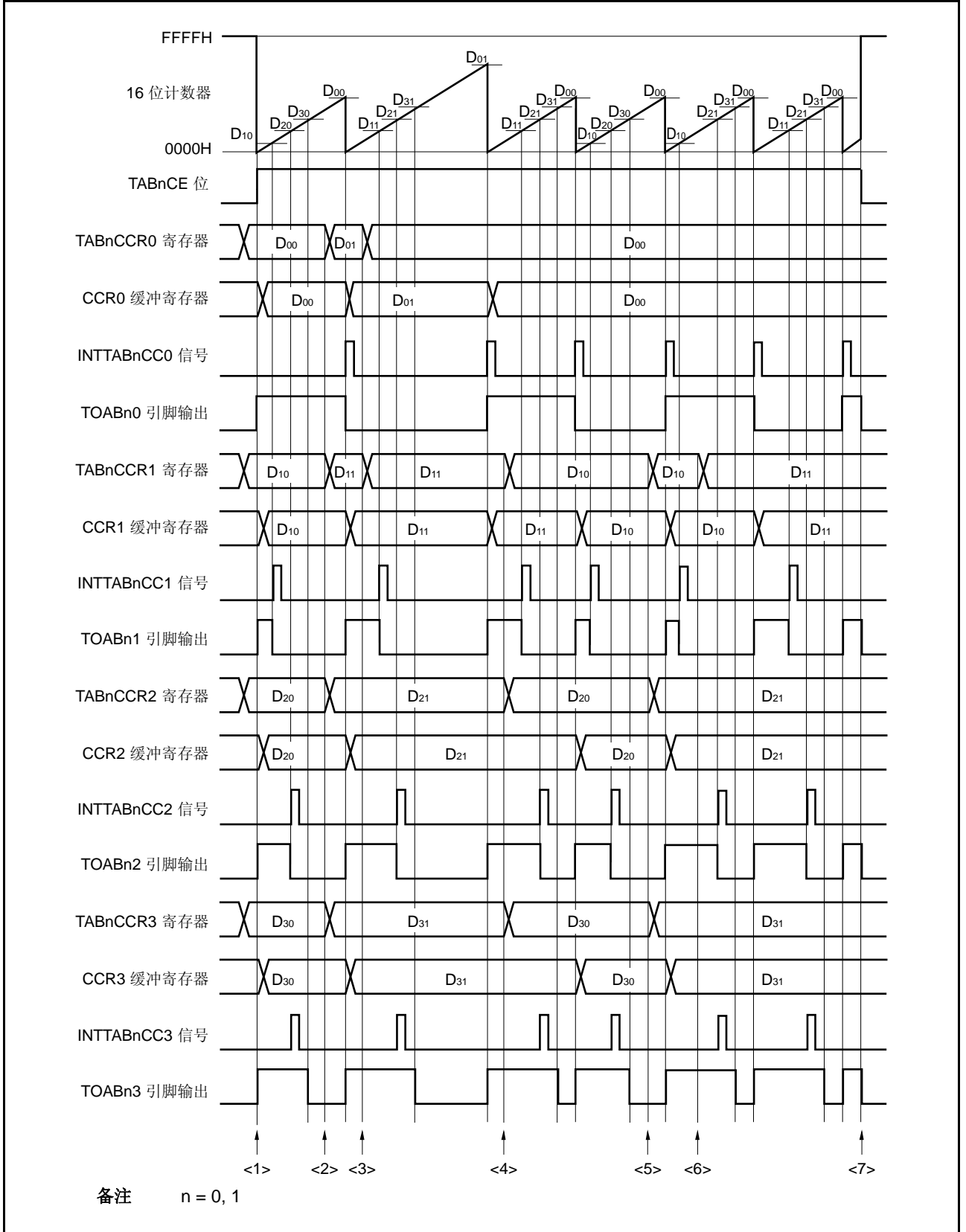
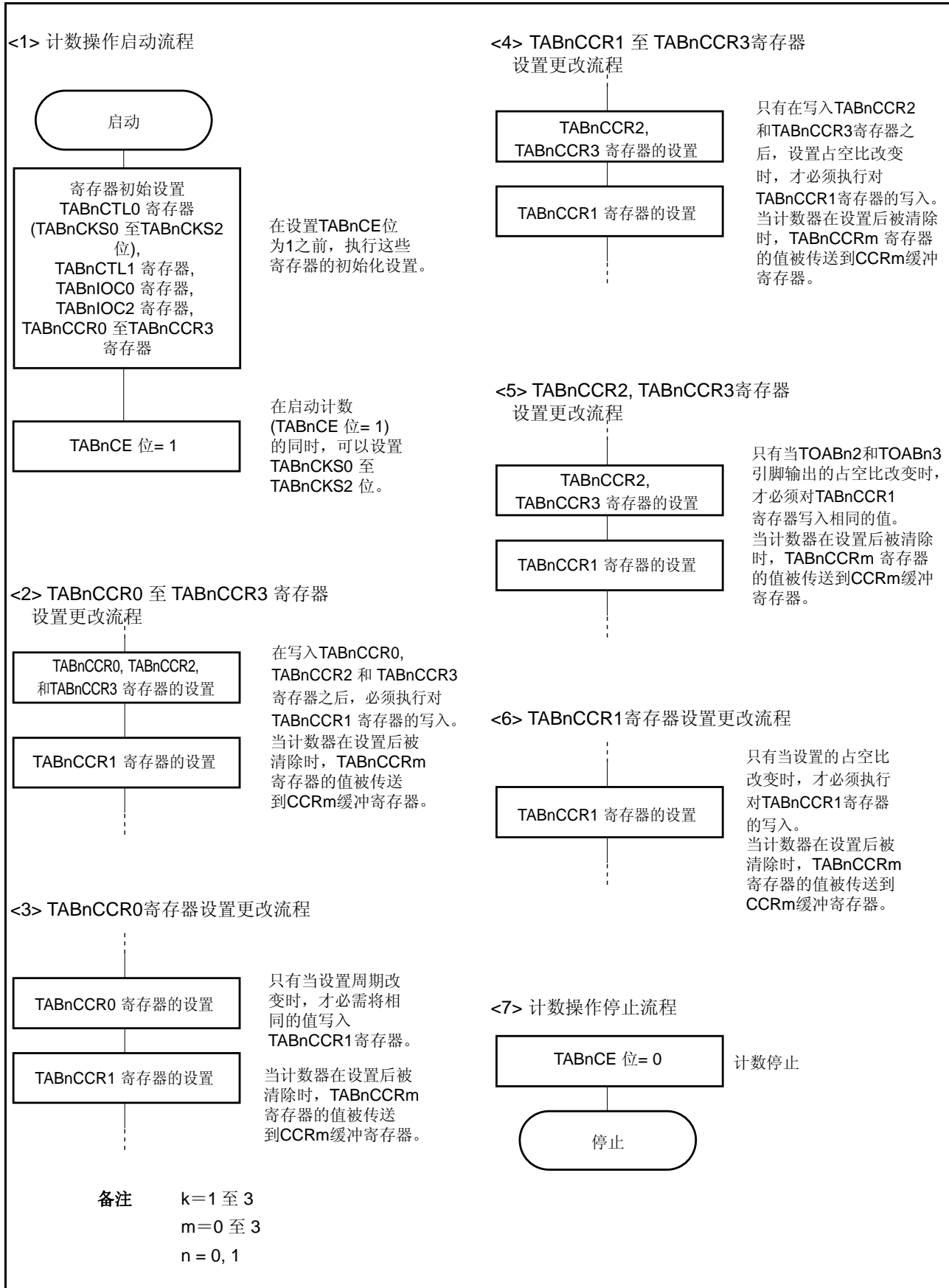


图 8-27. PWM 输出模式时的软件处理流程 (2/2)

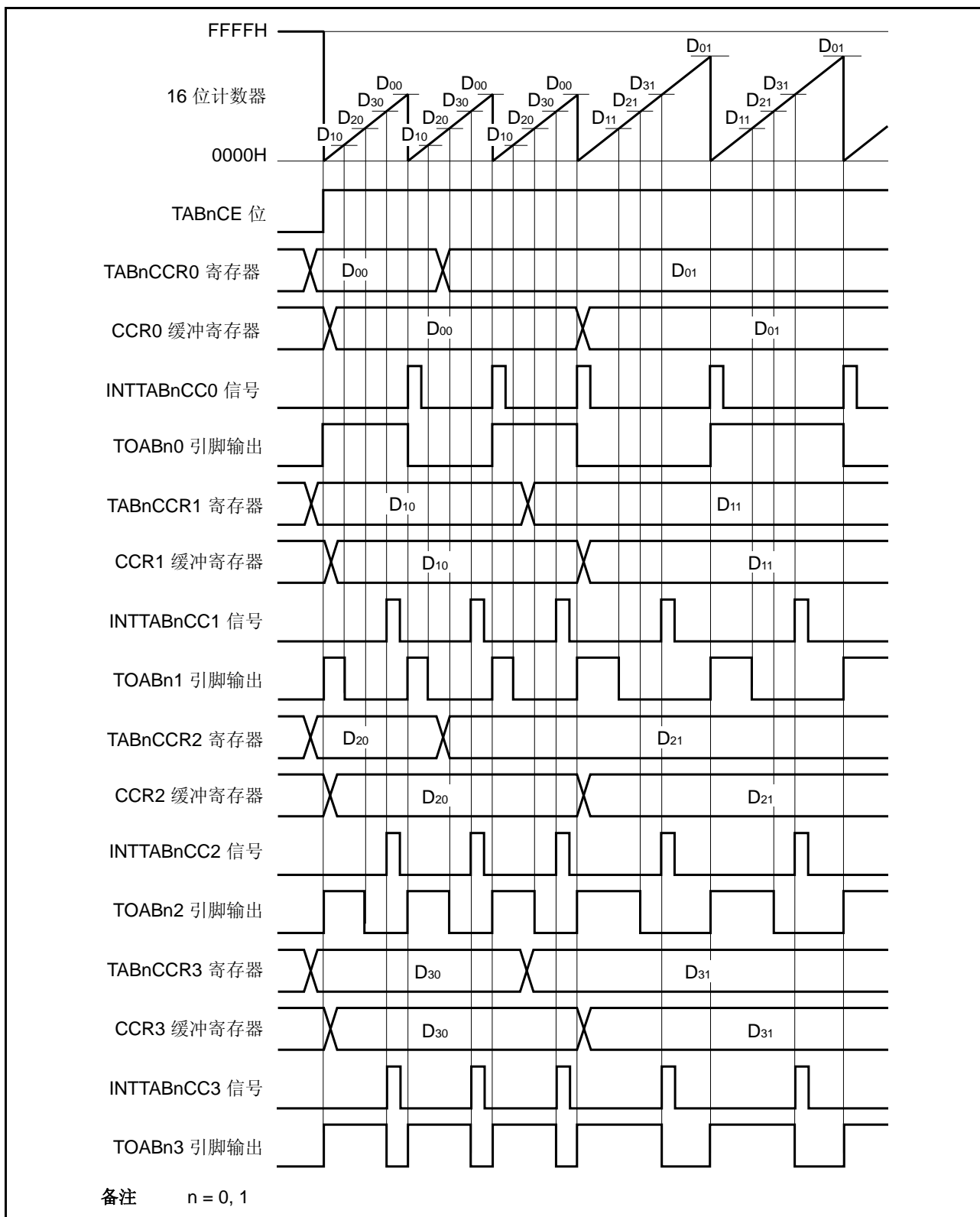


(2) PWM 输出模式的操作时序

(a) 操作过程中改变脉冲宽度

计数器操作过程中，若要改变 PWM 波形，应重新写入 TABnCCR1 寄存器。

当检测到 INTTABnCC1 信号以后，在写入 TABnCCR1 寄存器之后重写 TABnCCRk 寄存器。



为了将数据从 TABnCCRm 寄存器传送到 CCRm 缓冲寄存器，必须对 TABnCCR1 寄存器进行写入操作。此时如果 PWM 波形的周期和有效电平都要改变，应先将周期设定到 TABnCCR0 寄存器，再将有效电平宽度设定到 TABnCCR2 和 TABnCCR3 寄存器，然后将有效电平设定到 TABnCCR1 寄存器。

如果只改变 PWM 波形的周期，应先把周期设置到 TABnCCR0 寄存器，然后将相同的值写入 TABnCCR1 寄存器。

如果只改变 PWM 波形的有效电平宽度（占空比系数），需要先向 TABnCCR2 和 TABnCCR3 寄存器设置有效电平，然后再向 TABnCCR1 寄存器设置有效电平。

如果只改变 TOABn1 引脚输出的 PWM 波形的有效电平宽度（占空比系数），仅需要设置 TABnCCR1 寄存器。

如果只改变 TOABn2 和 TOABn3 引脚输出的 PWM 波形的有效电平宽度（占空比系数），应先向 TABnCCR2 和 TABnCCR3 寄存器设置有效电平宽度，然后将相同的值写入 TABnCCR1 寄存器。

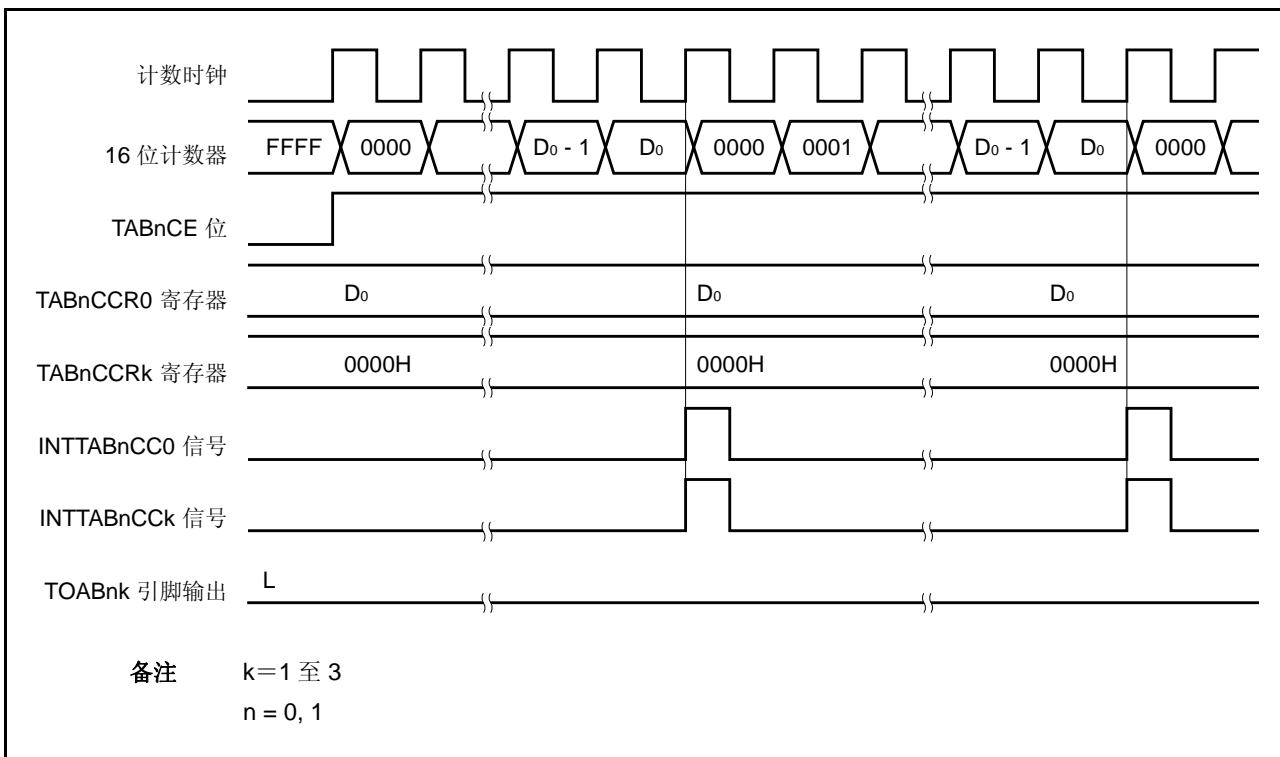
当数据写入 TABnCCR1 寄存器之后，在对 16 位计数器清零的同时，写入到 TABnCCRm 寄存器的值会被传送到 CCRm 缓冲寄存器，并作为与 16 位计数器进行比较的值。

在写入 TABnCCR1 寄存器一次之后，要再次写入 TABnCCR0 至 TABnCCR3 寄存器，应该在 INTTABnCC0 信号产生之后进行。否则，CCRM 缓冲寄存器的值可能不确定，因为把数据从 TABnCCRm 寄存器传送到 CCRm 缓冲寄存器的时序会和写入 TABnCCRm 寄存器的时序发生冲突。

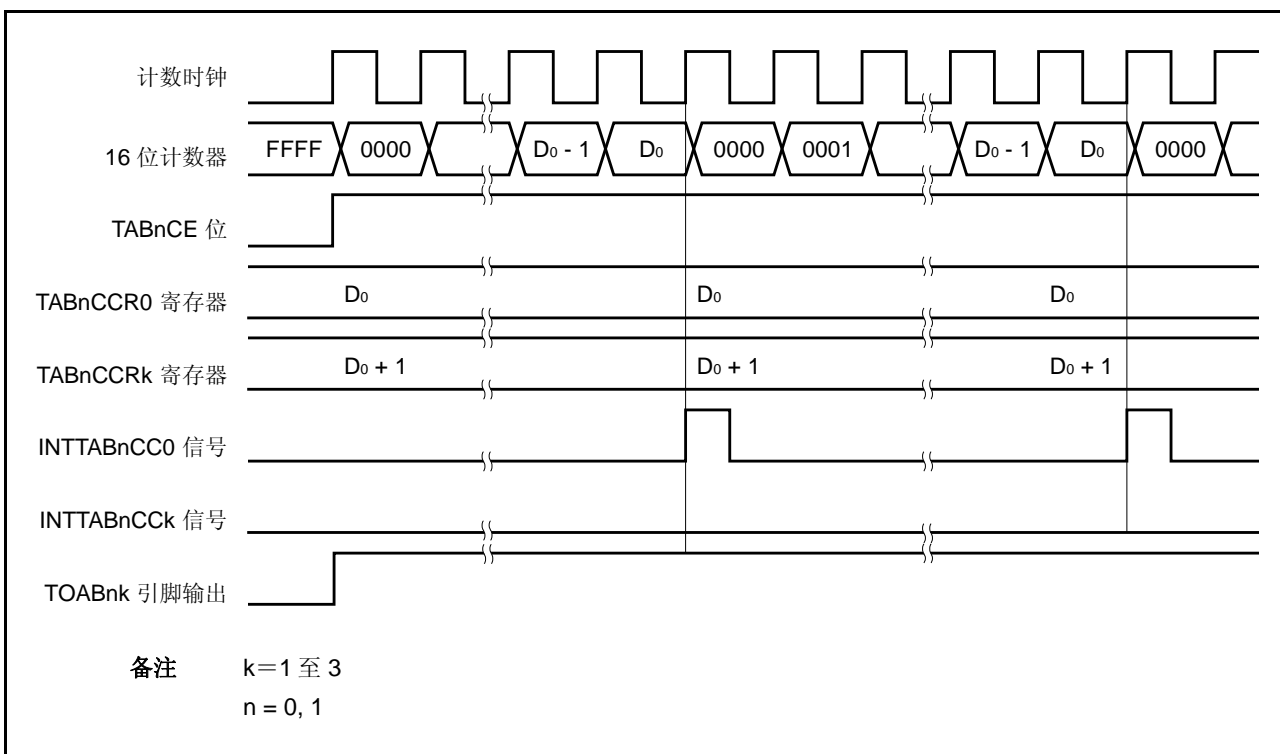
备注 m=0 至 3
 n = 0, 1

(b) PWM 波形的 0%和 100% 输出

为了输出 0%的波形，需要将 TABnCCRk 寄存器设置为 0000H。如果 TABnCCR0 寄存器的设定值为 FFFFH，则会在每个周期产生 INTTABnCCk 中断请求信号。

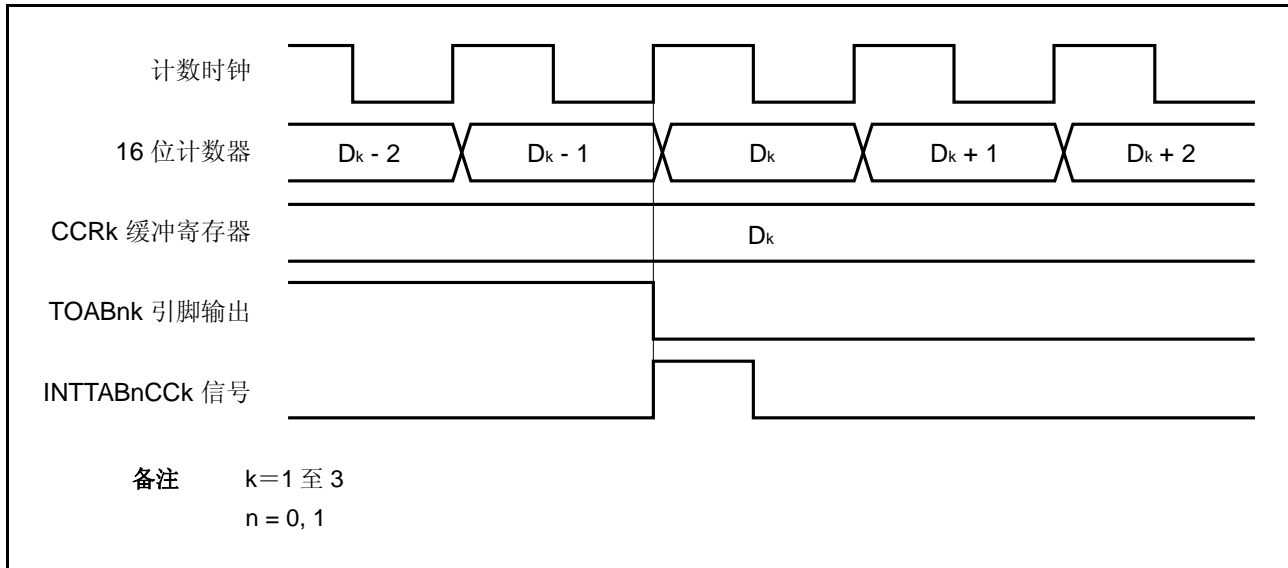


为了输出 100%的波形，需要将 TABnCCRk 寄存器的值设为：（TABnCCR0 寄存器的设定值 +1）。如果 TABnCCR0 寄存器的设定值为 FFFFH，则不会产生 100%波形输出。



(c) 比较匹配中断请求信号 (INTTABnCCk) 的发生时序

在 PWM 输出模式下，比较匹配中断请求信号 (INTTABnCCk) 的发生时序与其它情况下 INTTABnCCk 信号的发生时序有所不同；当 16 位计数器的计数值与 TABnCCRk 寄存器的值匹配时，产生 INTTABnCCk 信号。



通常在 16 位计数器的计数值与 TABnCCRk 寄存器的值匹配之后，在下一次计数时同步发生 INTTABnCCk 信号。

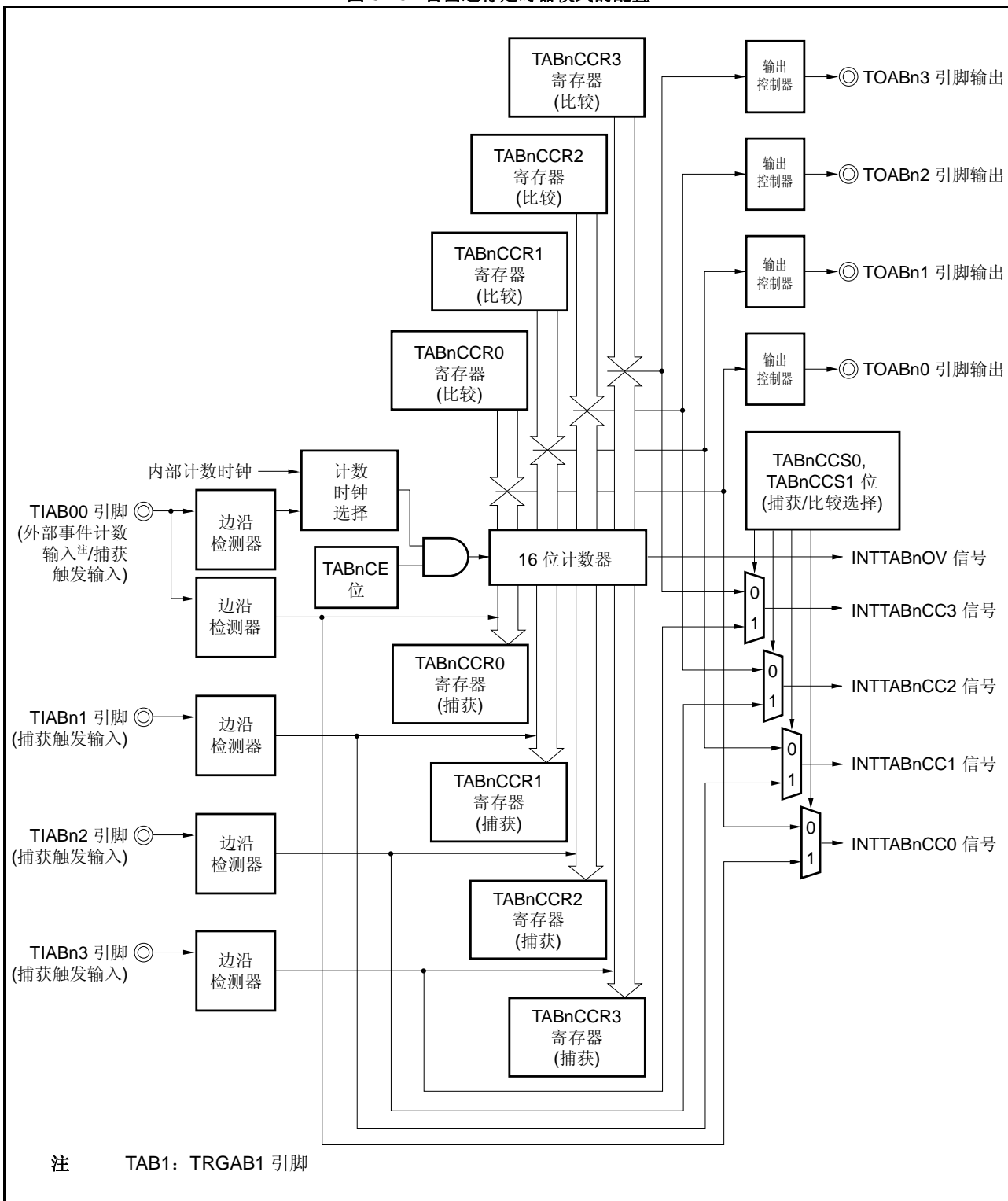
但是，在 PWM 输出模式下，INTTP0CC1 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOABnk 引脚输出信号变化的时序。

8.5.6 自由运行定时器模式 (TABnMD2 至 TABnMD0 位=101)

在自由运行定时器模式下，当 TABnCTL0.TABnCE 位置为 1 时，TABn 开始计数。此时，根据 TABnOPT0.TABnCCS0 和 TABnOPT0.TABnCCS1 位的设置，TABnCCRm 寄存器可以用作比较寄存器或捕获寄存器。

备注 m=0 至 3
n = 0, 1

图 8-28. 自由运行定时器模式的配置



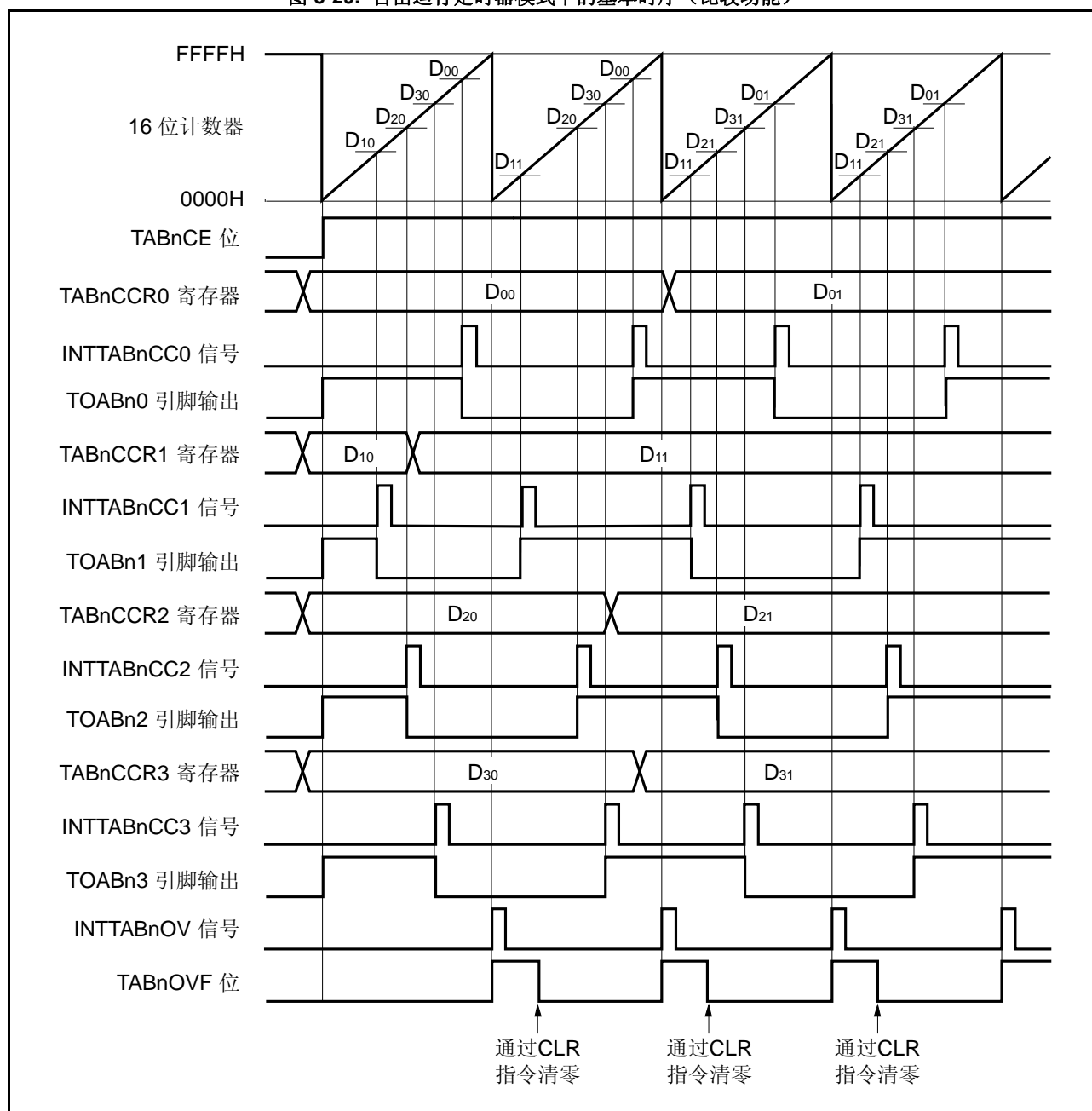
当 TABnCE 位被置为 1 时, TABn 开始计数, TOABn0 至 TOABn3 引脚的输出信号反相。此后当 16 位计数器的计数值与 TABnCCRm 寄存器的设定值匹配时, 产生一个比较匹配中断请求信号 (INTTABnCCm), 而且 TOABnm 引脚的输出信号反相。

16 位计数器继续对计数时钟同步计数。当它计数到 FFFFH 时, 会在下一个计数时钟时产生一个溢出中断请求信号 (INTTABnOV), 同时计数器清零为 0000H, 然后继续进行计数。此时, 溢出标志 (TABnOPT0.TABnOVF 位) 被置为 1, 通过软件运行 CLR 指令可以把溢出标志位清 0。

当计数器工作时, 可以重写 TABnCCRm 寄存器。如果进行重写, 则新写入的值会立刻反映出来, 并与计数值进行比较。

备注 m=0 至 3
n=0, 1

图 8-29. 自由运行定时器模式下的基本时序 (比较功能)



当 TABnCE 位被置为 1 时，16 位计数器开始计数。当检测到 TIABnm 引脚输入信号的有效边沿时，16 位计数器的计数值存储在 TABnCCRm 寄存器中，并产生一个捕获中断请求信号 (INTTABnCCm)。

16 位计数器继续对计数时钟同步计数。当它计数到 FFFFH 时，会在下一个计数时钟时产生一个溢出中断请求信号 (INTTABnOV)，同时计数器清零为 0000H，然后继续进行计数。此时，溢出标志 (TABnOVF 位) 被置为 1，通过软件运行 CLR 指令可以把溢出标志位清 0。

备注 m=0 至 3
n = 0, 1

图 8-30. 自由运行定时器模式下的基本时序 (捕获功能)

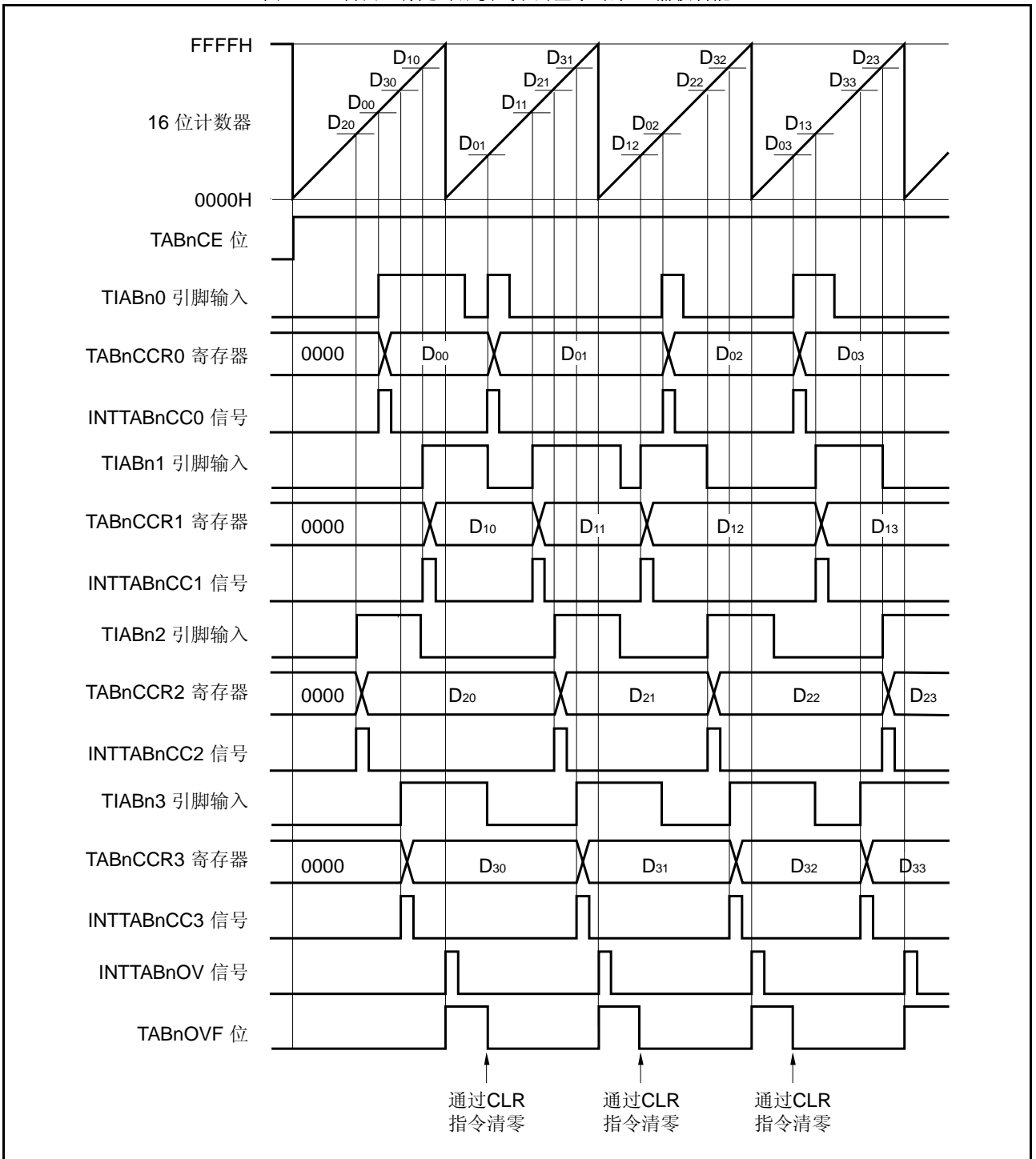


图 8-31. 自由运行定时器模式下的寄存器设置 (1/3)

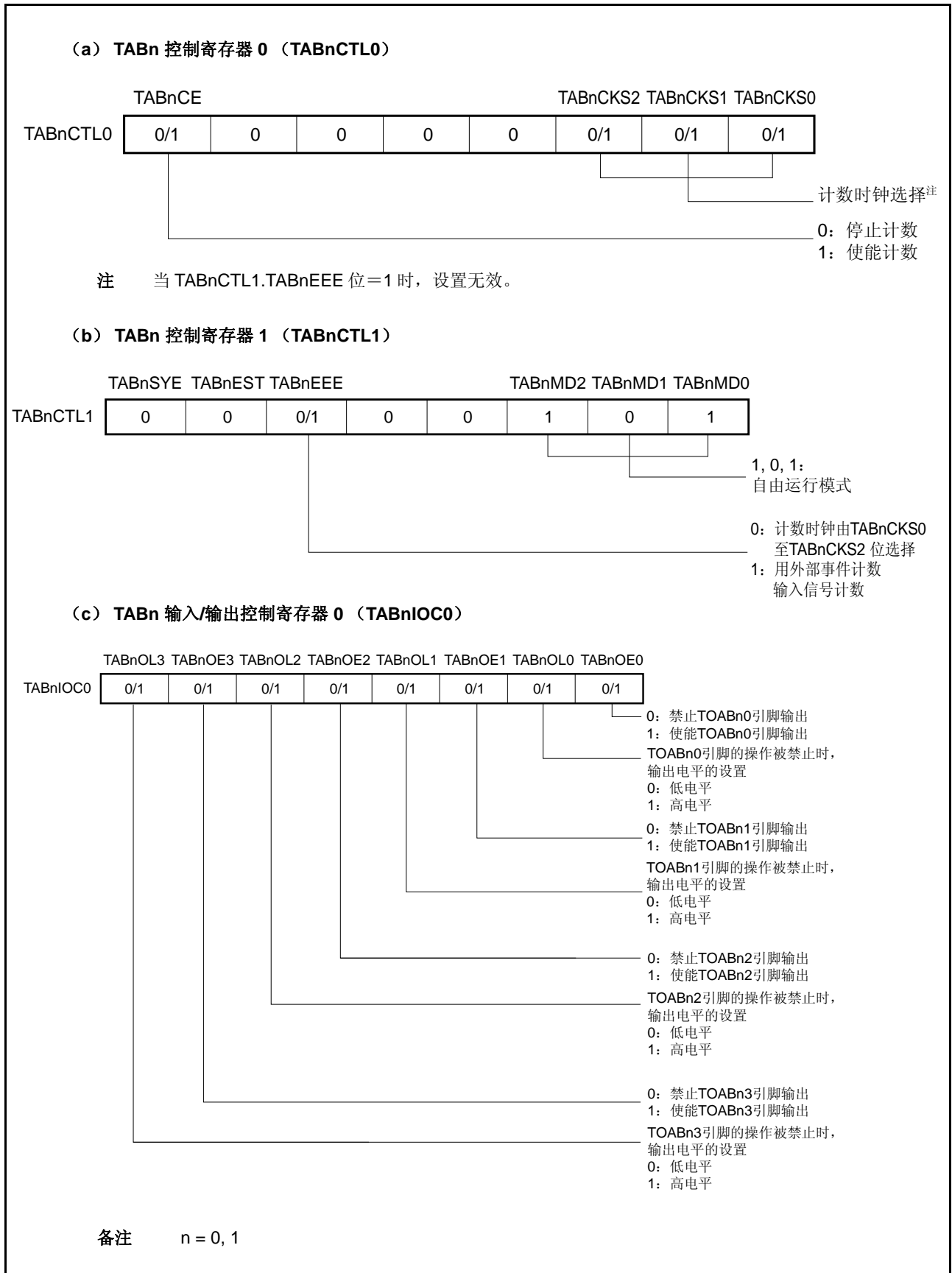


图 8-31. 自由运行定时器模式下的寄存器设置 (2/3)

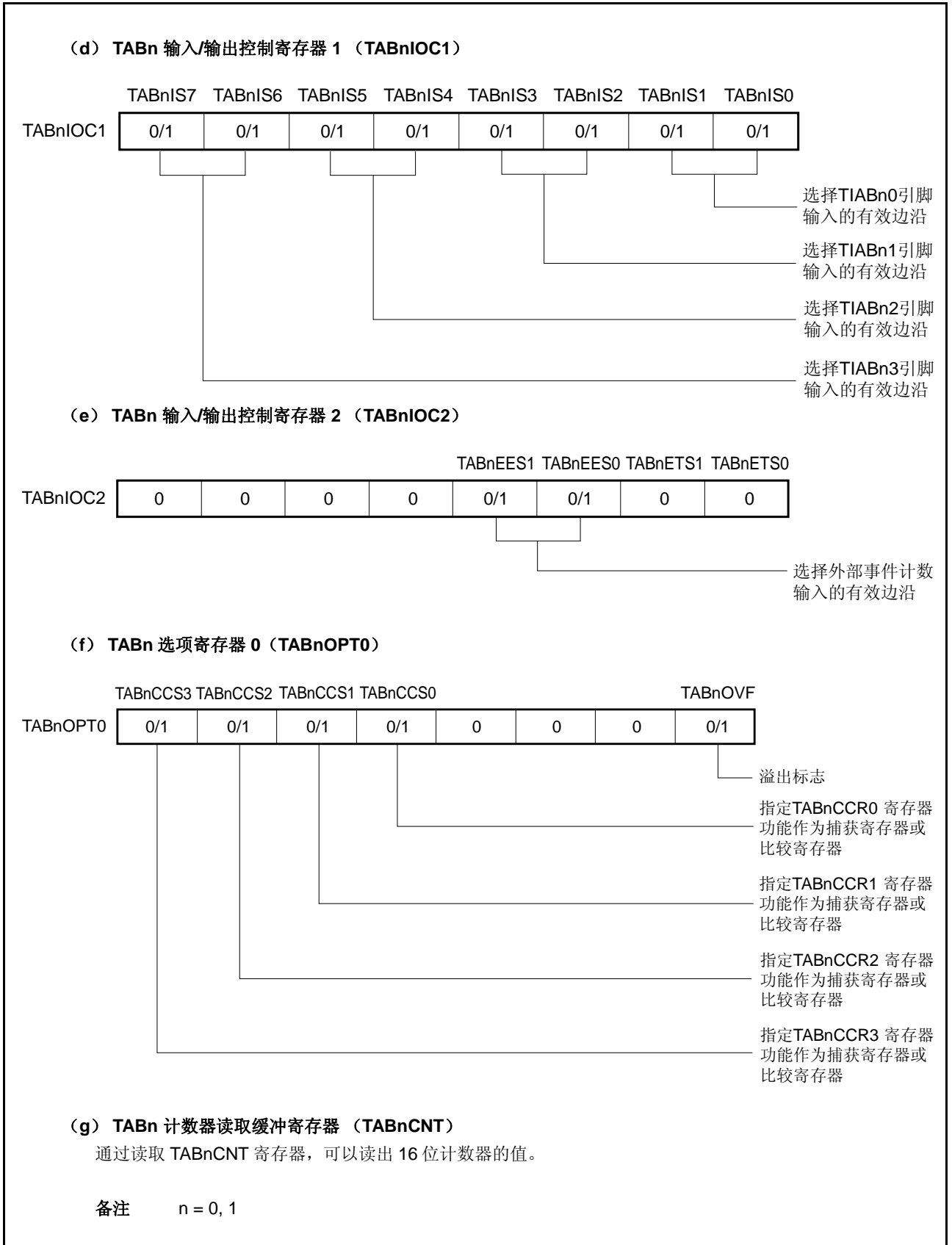


图 8-31. 自由运行定时器模式下的寄存器设置 (3/3)

(h) TABn 捕获/比较寄存器 0 至 3 (TABnCCR0 至 TABnCCR3)

根据 TABnOPT0.TABnCCSm 位的设置，这些寄存器可以用作捕获寄存器或比较寄存器。

当用作捕获寄存器时，在检测到 TIABnm 引脚输入信号的有效边沿时，存储 16 位计数器的当前计数值。

当用作比较寄存器时，而且 TABnCCRm 寄存器的值被设置为 D_m 时，则当计数器达到 $(D_m + 1)$ 时会产生 INTTABnCCm 中断信号，同时，TOABnm 引脚的输出信号反相。

备注 m=0 至 3
 n = 0, 1

(1) 自由运行定时器模式时的操作流程

(a) 当捕获/比较寄存器用作比较寄存器时

图 8-32. 自由运行定时器模式的软件处理流程（比较功能）（1/2）

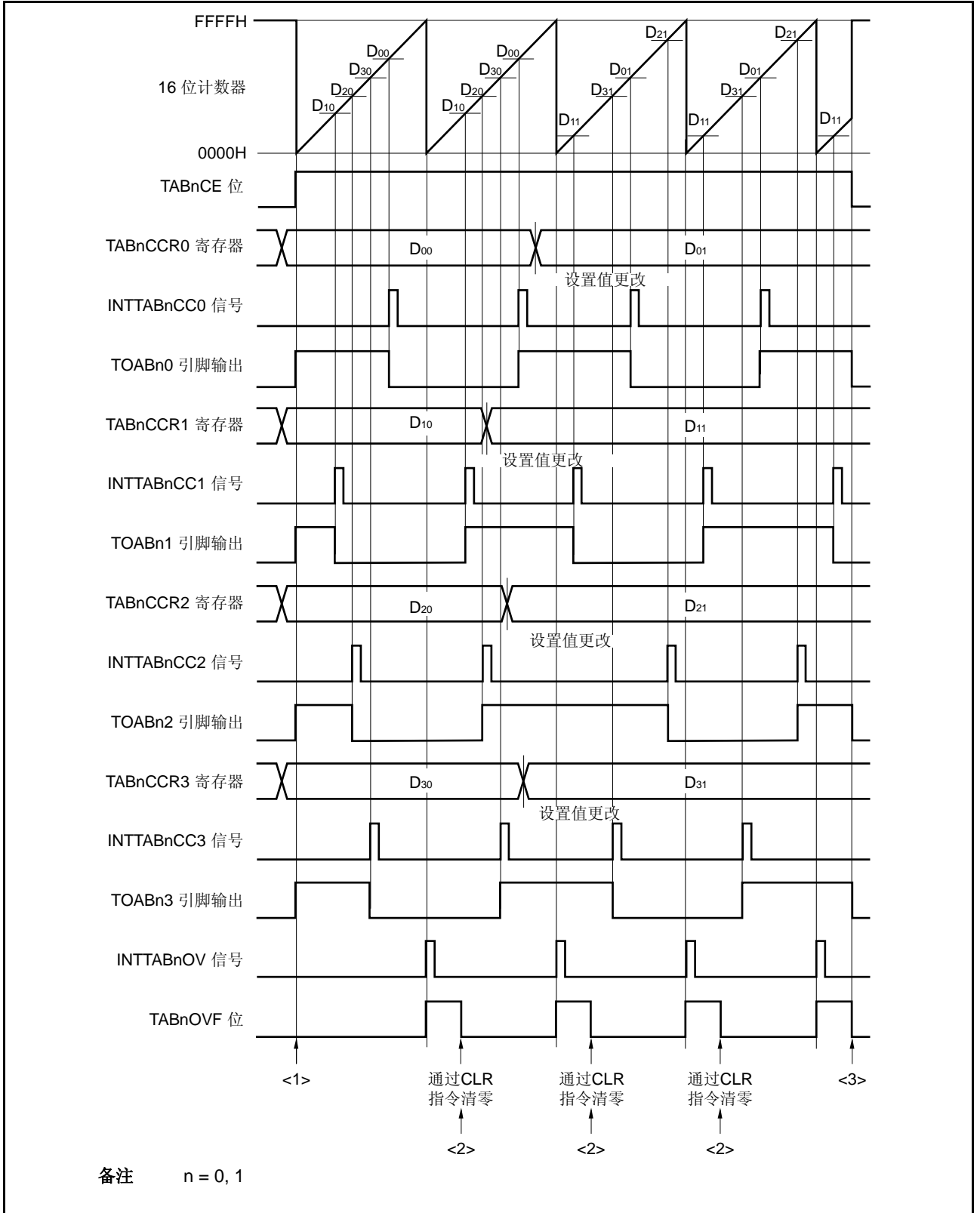
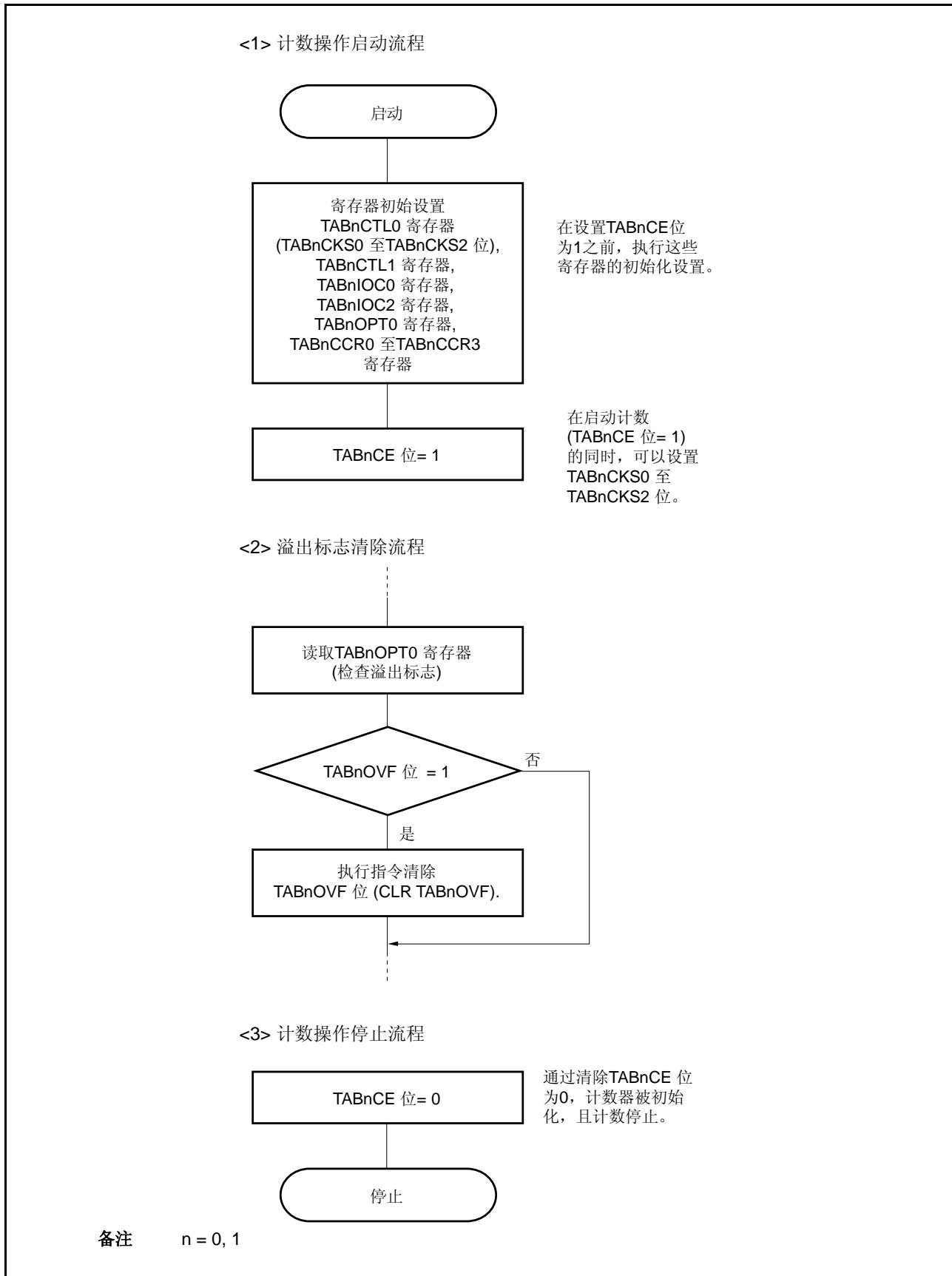


图 8-32. 自由运行定时器模式的软件处理流程 (比较功能) (2/2)



(b) 当捕获/比较寄存器用作捕获寄存器时

图 8-33. 自由运行定时器模式的软件处理流程 (捕获功能) (1/2)

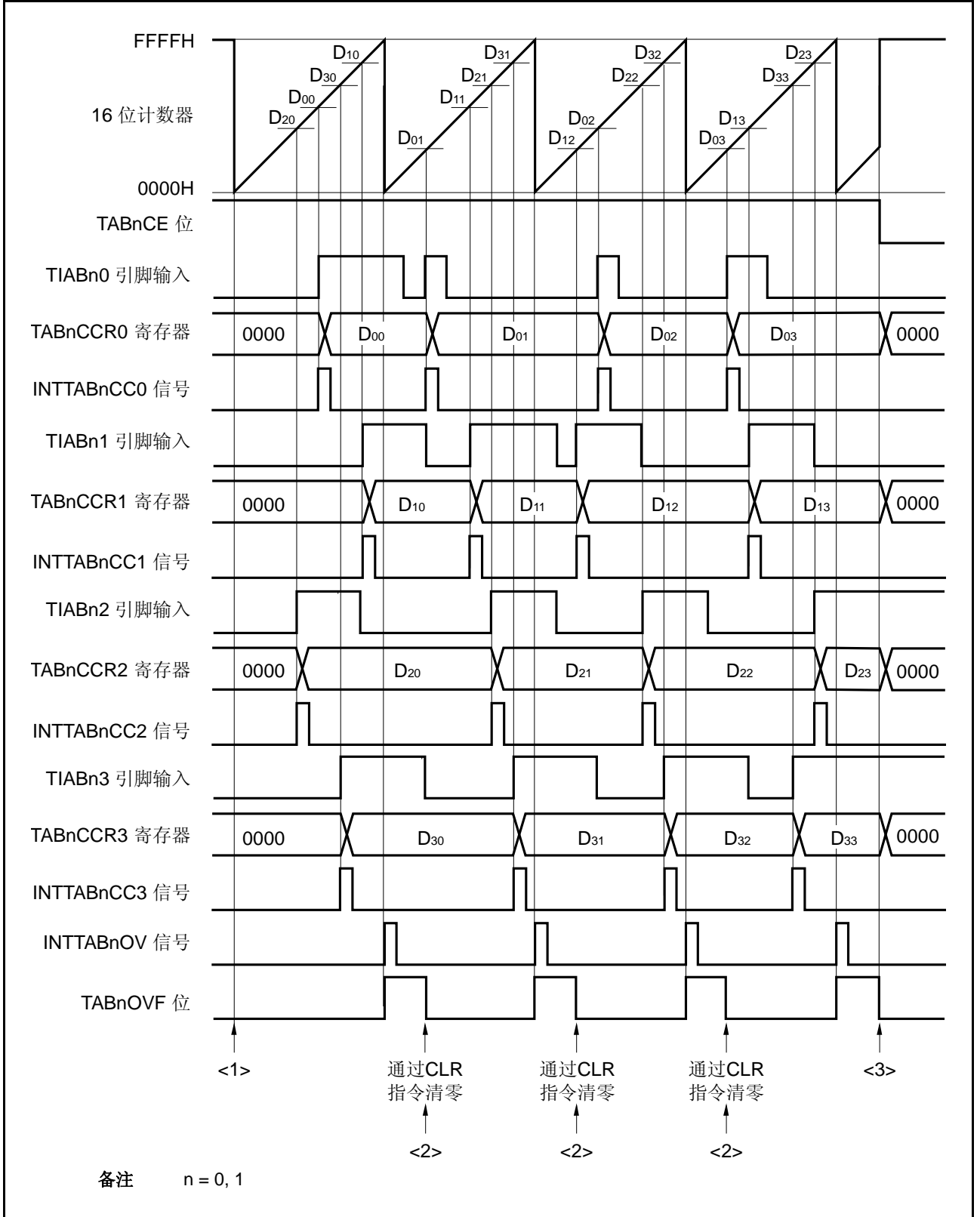
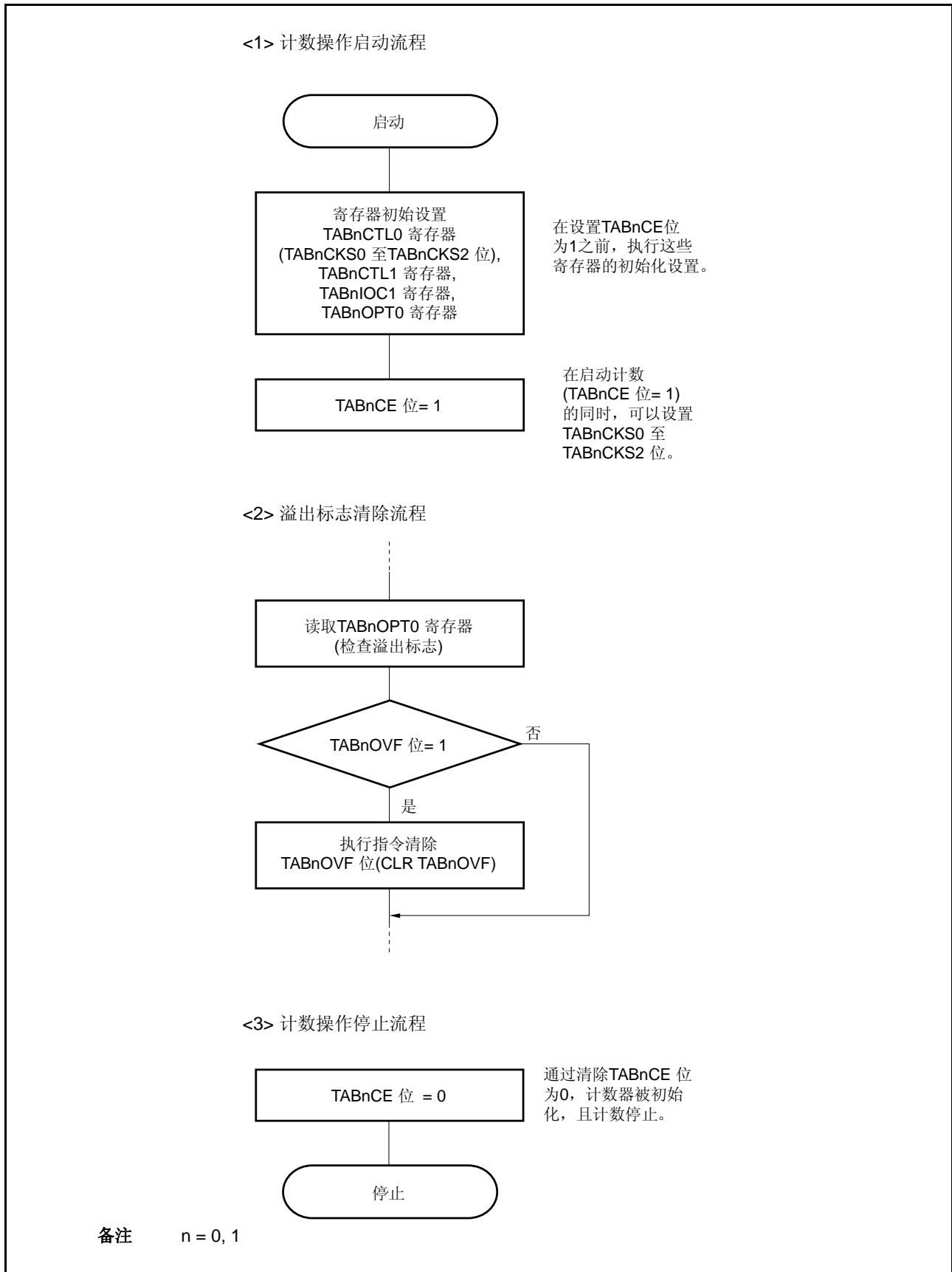


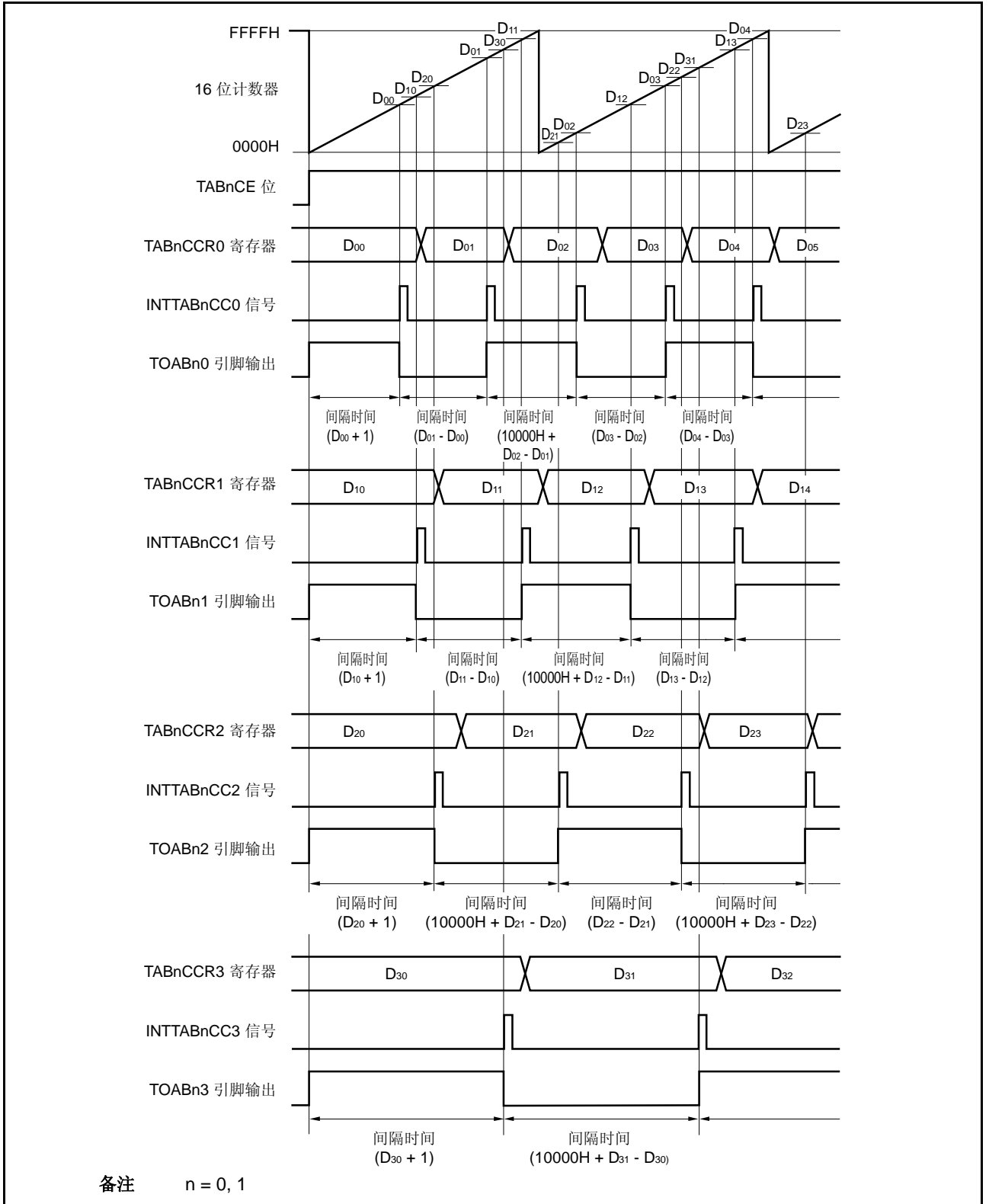
图 8-33. 自由运行定时器模式时的软件处理流程 (捕获功能) (2/2)



(2) 自由运行定时器模式时的操作时序

(a) 比较寄存器的间隔定时操作

当 TABn 用作间隔定时器，TABnCCRm 寄存器用作比较寄存器时，在每次检测到 INTTABnCCm 信号后，都需要通过软件设置一个比较值以产生下一个中断请求信号。



在自由运行定时器模式下进行时间间隔操作时，一个通道可以设置四个时间间隔。

要进行时间间隔操作，当检测到 $INTTABnCCm$ 信号后，必须在中断服务程序中重置相应的 $TABnCCRm$ 寄存器的值。

重置 $TABnCCRm$ 寄存器的设置值可由下式计算，其中“ D_m ”是间隔时间：

比较寄存器默认值： $D_m - 1$

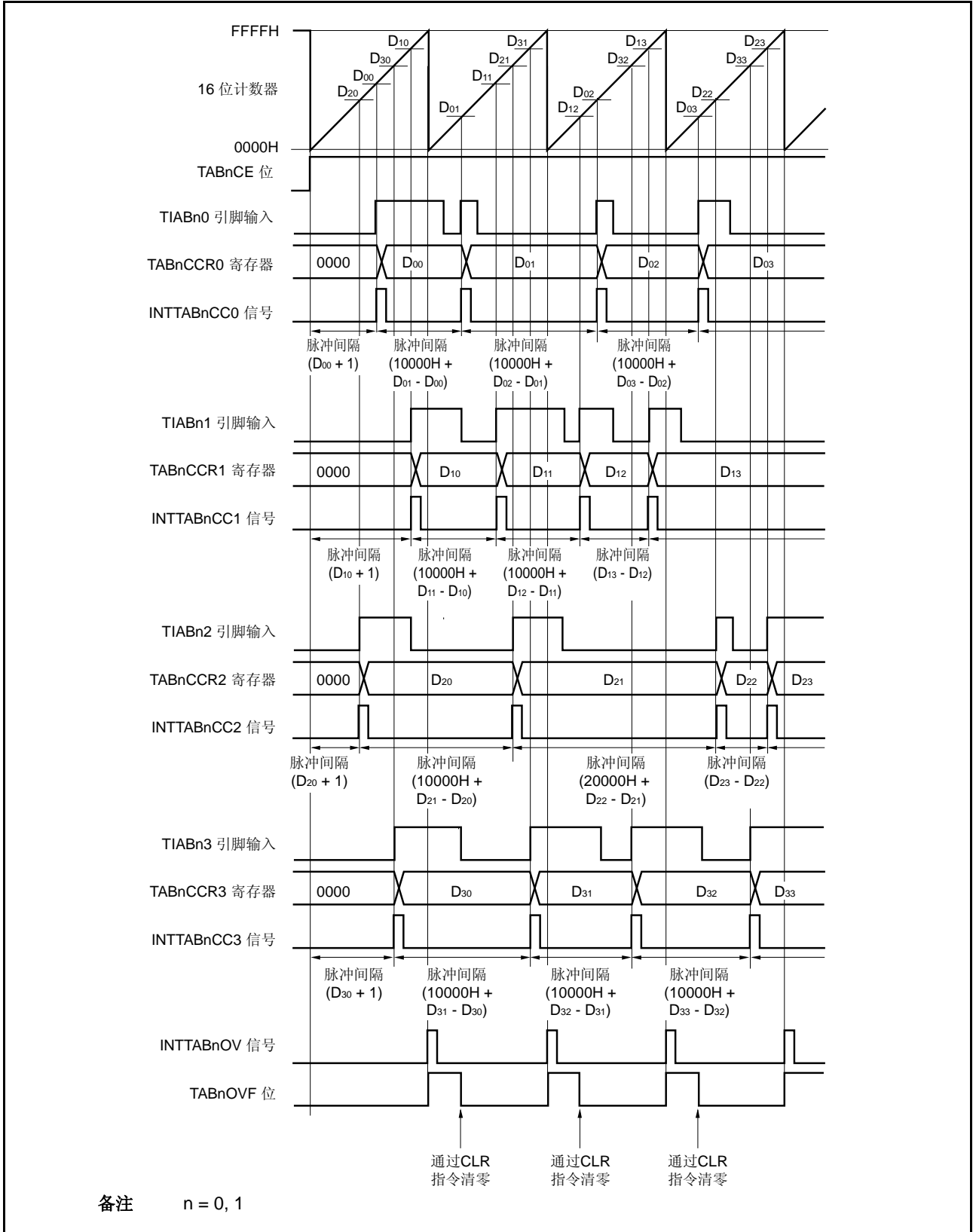
比较寄存器的第二次及其以后的设置值：上次设置值 + D_m

（若计算结果大于 $FFFFH$ ，则从结果中减去 $10000H$ ，并将此值设置给寄存器。）

备注 $m=0$ 至 3
 $n = 0, 1$

(b) 使用捕获寄存器进行脉冲宽度测量

在用 TABnCCRm 寄存器作为捕获寄存器进行脉宽测量时，每次检测到 INTTABnCCm 信号时，都需要通过软件读取捕获寄存器并计算间隔时间。

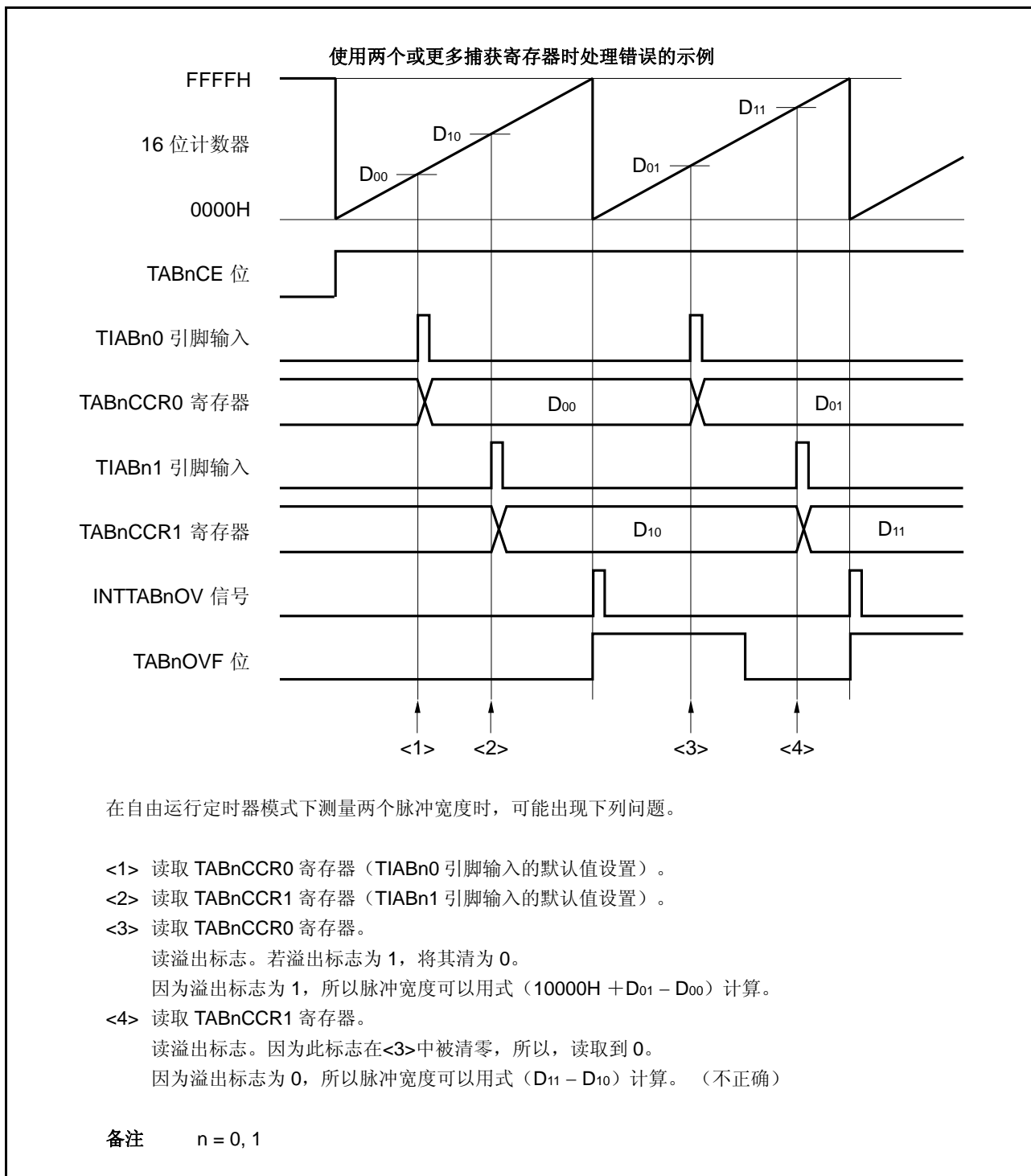


在自由运行定时器模式下执行脉宽测量时，可以用一个通道测量四个脉冲宽度。
要测量脉冲宽度，可以与 INTTABnCCm 信号同步读取 TABnCCRm 寄存器的值，并计算读取值与先前读取值之间的差，得到脉冲宽度。

备注 m=0 至 3
 n = 0, 1

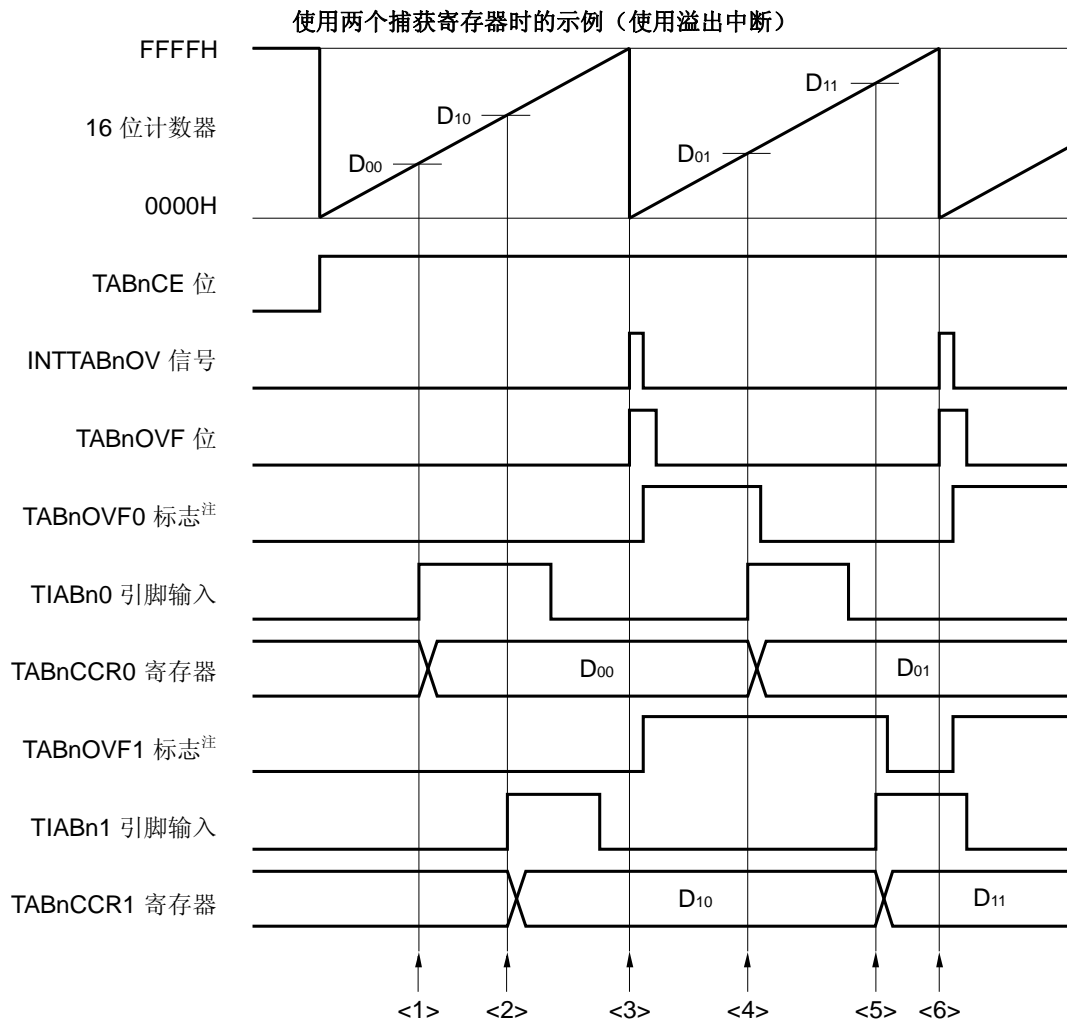
(c) 当使用两个或更多捕获寄存器时对溢出的处理

当使用两个捕获寄存器时必须格外小心。首先是一个不正确处理的示例，如下所示。



在使用两个或更多个捕获寄存器时，如果溢出标志由一个捕获寄存器清为 0，则另一个捕获寄存器可能得不到正确的脉冲宽度。

在使用两个或更多个捕获寄存器时，使用软件解决。关于如何使用软件的示例如下。



注 由软件对内部 RAM 中的 TABnOVF0 和 TABnOVF1 标志进行设置。

<1> 读取 TABnCCR0 寄存器（TIABn0 引脚输入的默认值设置）。

<2> 读取 TABnCCR1 寄存器（TIABn1 引脚输入的默认值设置）。

<3> 发生溢出，在溢出中断服务程序中将 TABnOVF0 和 TABnOVF1 标志设置为 1，并将溢出标志清为 0。

<4> 读取 TABnCCR0 寄存器。

读取 TABnOVF0 标志。如果 TABnOVF0 标志为 1，则将其清为 0。

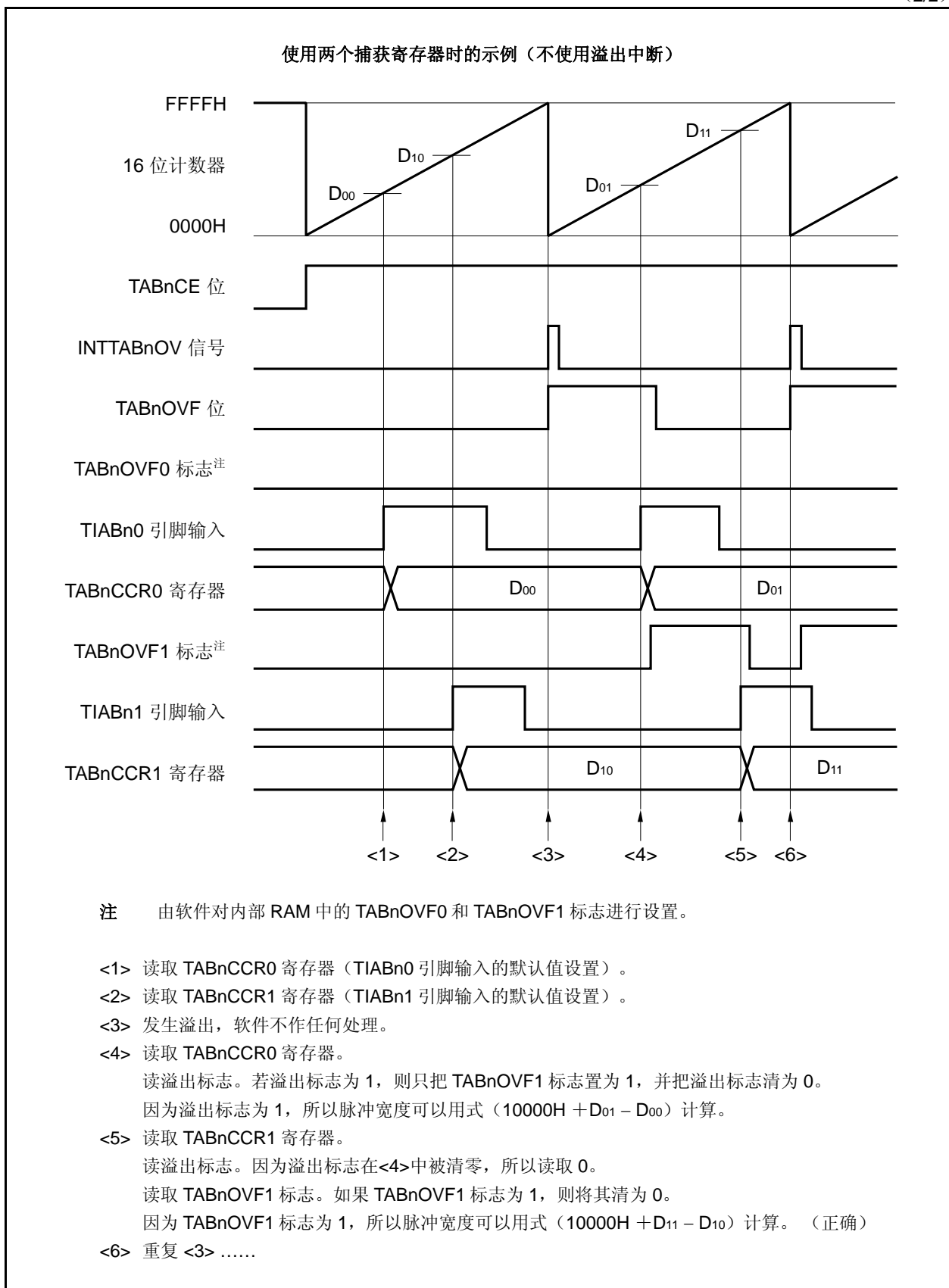
因为 TABnOVF0 标志为 1，所以脉冲宽度可以用式 $(10000H + D_{01} - D_{00})$ 计算。

<5> 读取 TABnCCR1 寄存器。

读取 TABnOVF1 标志。如果 TABnOVF1 标志为 1，则将其清为 0（TABnOVF0 标志在<4>中已清零，而 TABnOVF1 标志仍然保持为 1）。

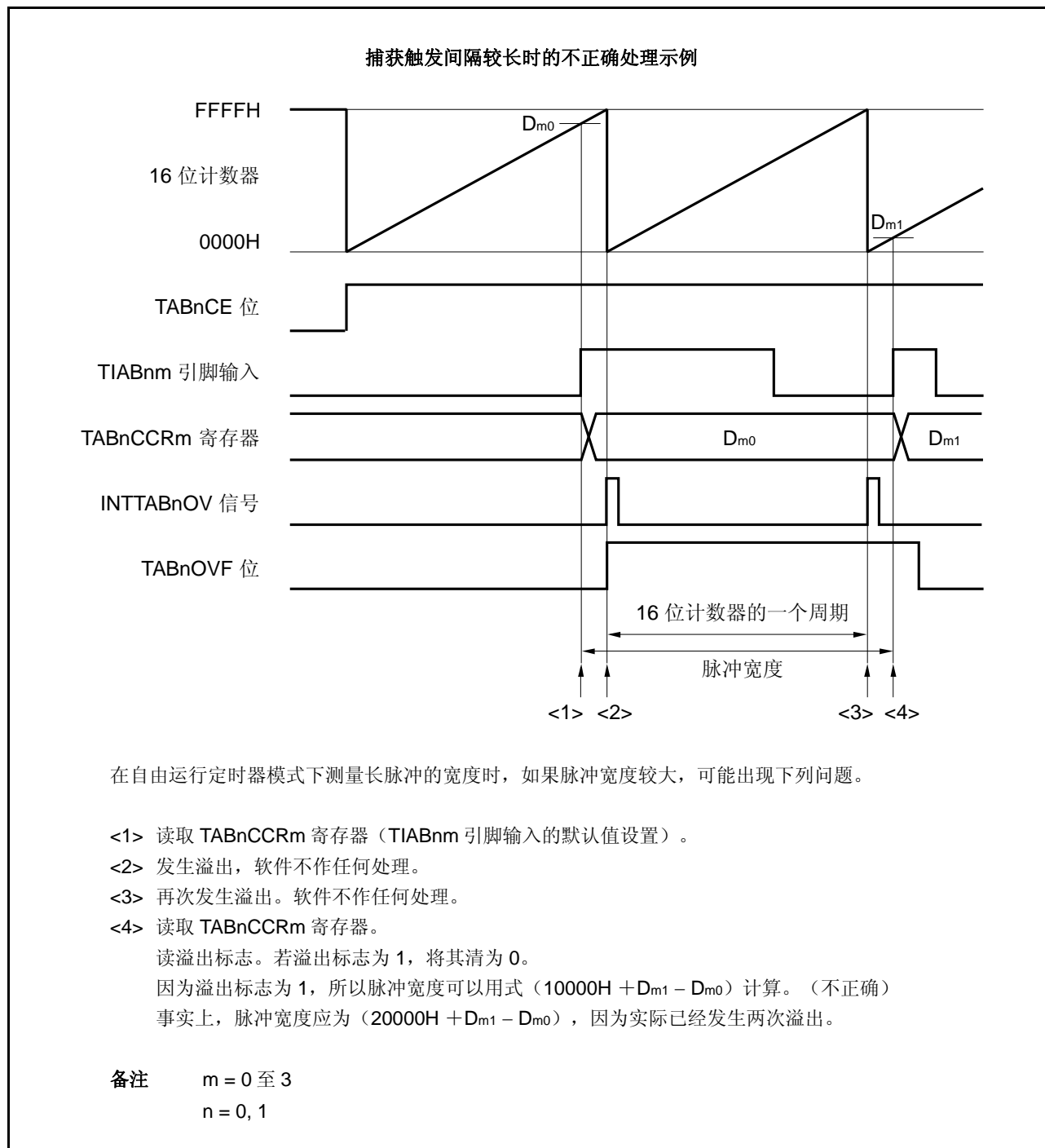
因为 TABnOVF1 标志为 1，所以脉冲宽度可以用式 $(10000H + D_{11} - D_{10})$ 计算。（正确）

<6> 重复 <3> ……



(d) 当捕获触发间隔较长时的溢出处理

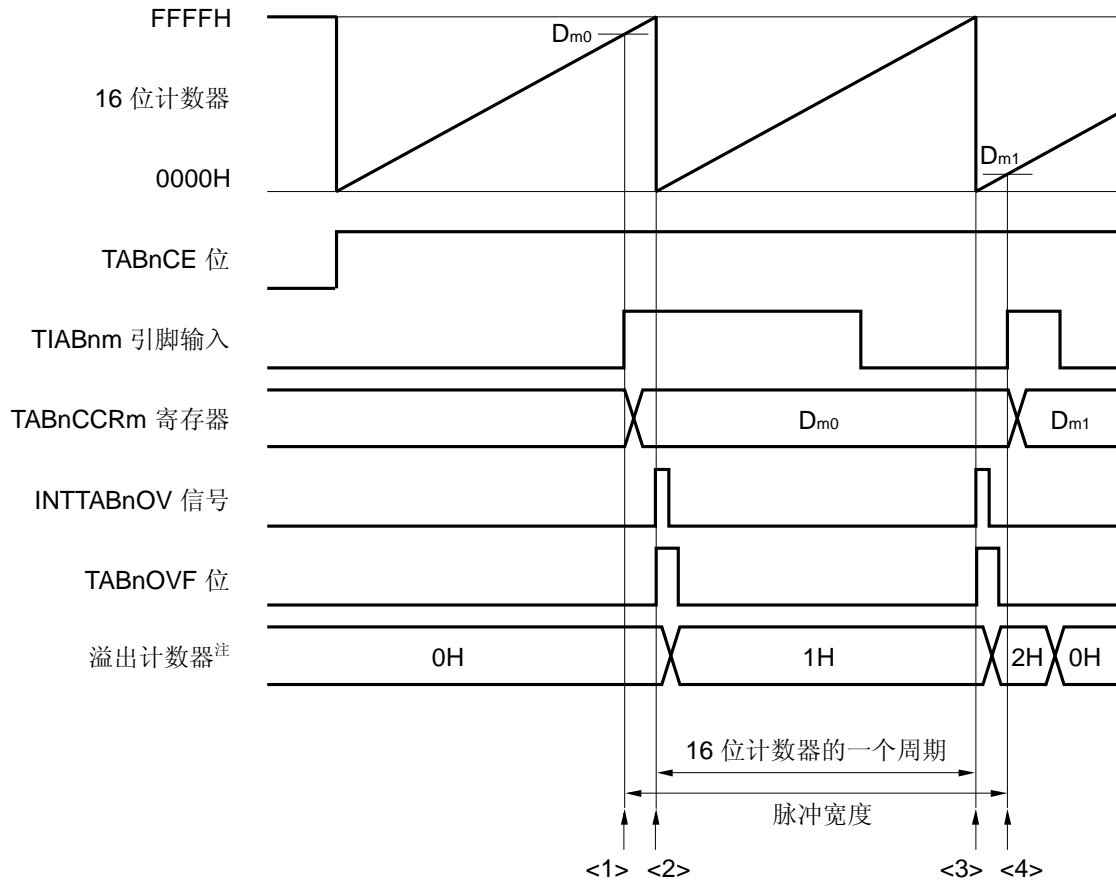
如果脉冲宽度大于 16 位计数器的一个计数循环周期，则必须加以注意，因为从第一个捕获触发信号到下一个触发信号之间可能发生多次溢出。首先是一个不正确处理的示例，如下所示。



如果在捕获触发间隔期间，发生两次或更多次的溢出，则无法获得正确的脉冲宽度。

如果捕获触发间隔时间较长时，应该减慢计数时钟以延长 16 位计数器的循环周期，或者使用软件来解决。关于如何使用软件，示例如下。

捕获触发间隔较长时的示例



注 内部 RAM 中的溢出计数器由软件任意设置。

<1> 读取 TABnCCRm 寄存器 (TIABnm 引脚输入的默认值设置)。

<2> 发生溢出, 在溢出中断服务程序中增加溢出计数器的值, 并将溢出标志清为 0。

<3> 再次发生溢出。在溢出中断服务程序中增加 (+1) 溢出计数器的值, 并把溢出标志清为 0。

<4> 读取 TABnCCRm 寄存器。

读取溢出计数器。

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

在本例中, 脉冲宽度为 $(20000H + D_{m1} - D_{m0})$, 因为溢出实际发生了两次。

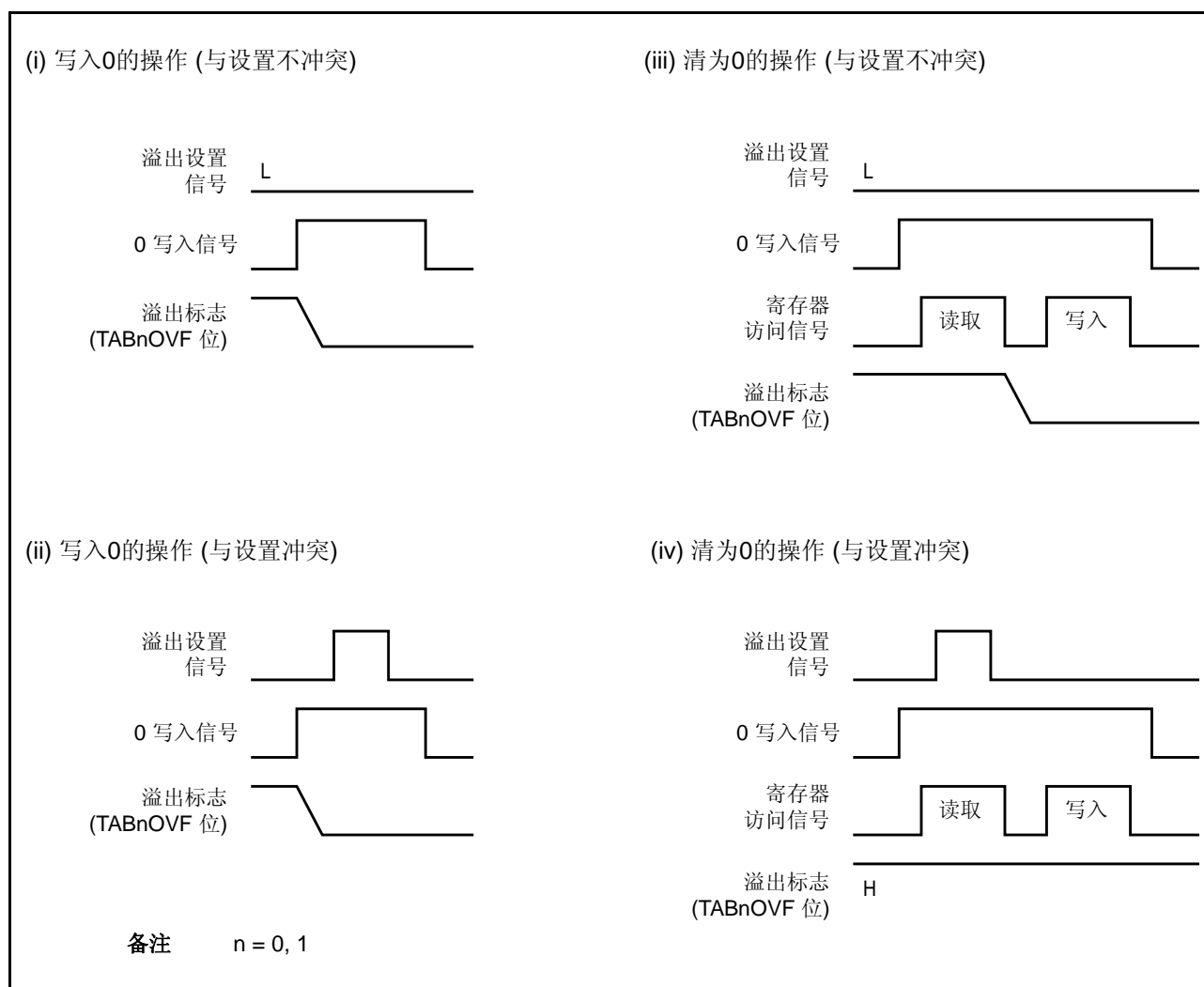
溢出计数器清零 (0H)。

备注 m = 0 至 3

n = 0, 1

(e) 清除溢出标志

通过 CLR 指令将 TABnOVF 位清为 0 以及将 8 位数据 (第 0 位是 0) 写入 TABnOPT0 寄存器, 可以将溢出标志清 0。要准确检测溢出, 当 TABnOVF 位为 1 时读取 TABnOVF 位, 然后用位操作指令将溢出标志清零。



要将溢出标志清为 0, 应读取溢出标志, 检查该标志如果已经被设置为 1, 并用 CLR 指令把它清零。若在未检查溢出标志是否为 1 就把 0 写入溢出标志中, 则溢出的设置信息可能因为写入 0 而被清除 (上图中的 (ii))。因此, 即使实际发生了溢出, 软件也可能会判断为无溢出发生。

如果用 CLR 指令将溢出标志清为 0 时, CLR 指令的执行与当前发生的溢出相冲突, 则在清零指令执行后溢出标志仍然会保持置位状态。

8.5.7 脉冲宽度测量模式 (TABnMD2 至 TABnMD0 位=110)

在脉冲宽度测量模式下，当 TABnCTL0.TABnCE 位被置为 1 时，TABn 开始计数。每次检测到 TIABnm 引脚输入信号的有效边沿时，16 位计数器的计数值就存储在 TABnCCRm 寄存器中，且 16 位计数器清零为 0000H。

在出现捕获中断请求信号 (INTTABnCCm) 之后，可以通过读取 TABnCCRm 寄存器来测量有效边沿的间隔。

选择 TIABn0 至 TIABn3 引脚的其中之一作为捕获触发信号输入引脚。使用 TABnIOC1 寄存器将未使用的引脚指定为“无边沿检测”。

在用外部时钟作为计数时钟时，测量 TIABnk 引脚输入的脉冲宽度，因为外部时钟固定接入 TIABn0 引脚。此时，把 TABnIOC1.TABnIS1 和 TABnIOC1.TABnIS0 位清零为 00 (捕获触发输入 (TIABn0 引脚)：无边沿检测)。

对于 TAB1，外部时钟从 ECTAB1 引脚输入，使用 TIAB10 至 TIAB13 引脚可以测量脉冲宽度。

备注 m=0 至 3

n = 0, 1

k=1 至 3

图 8-34. 脉冲宽度测量模式的配置

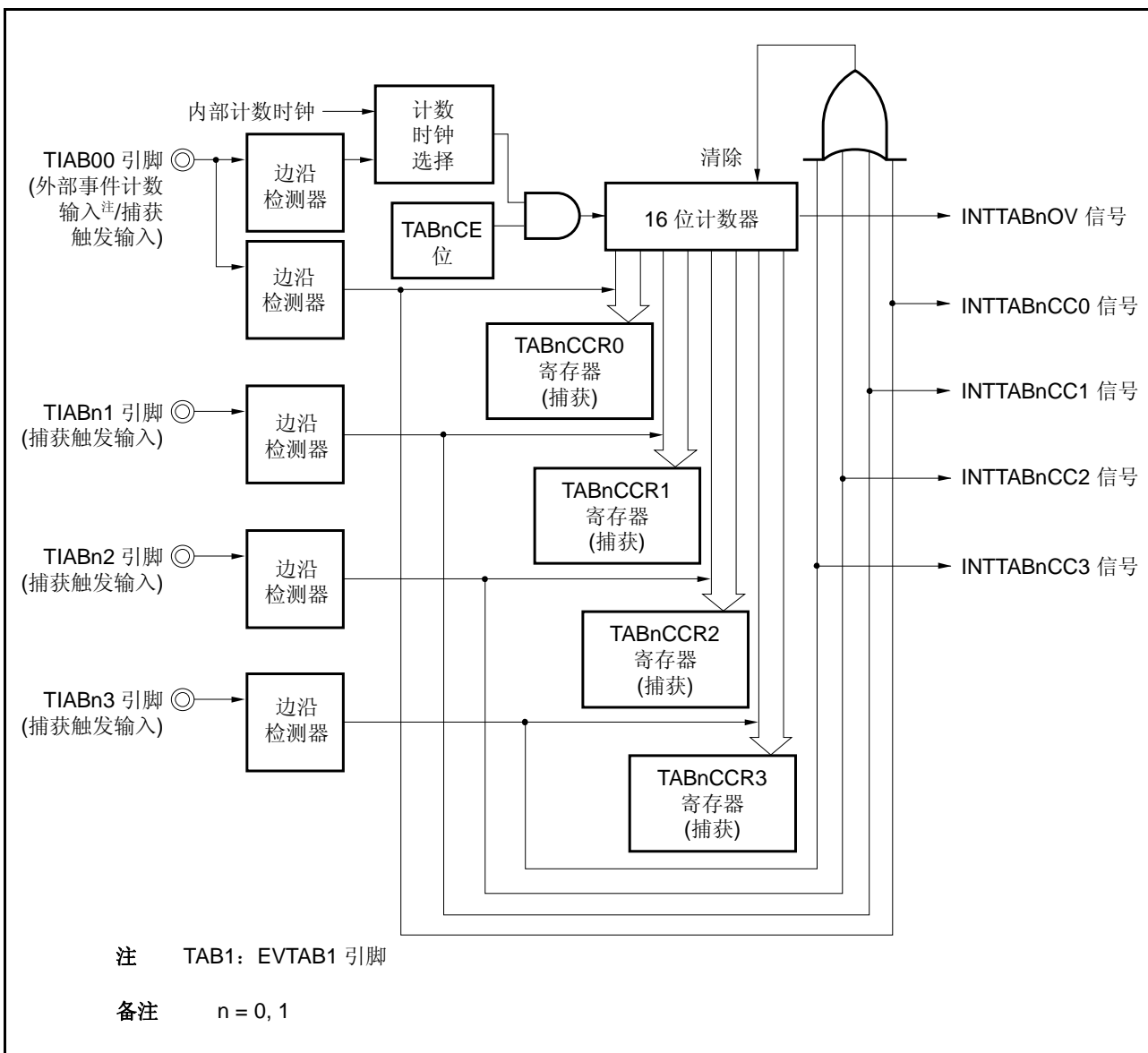
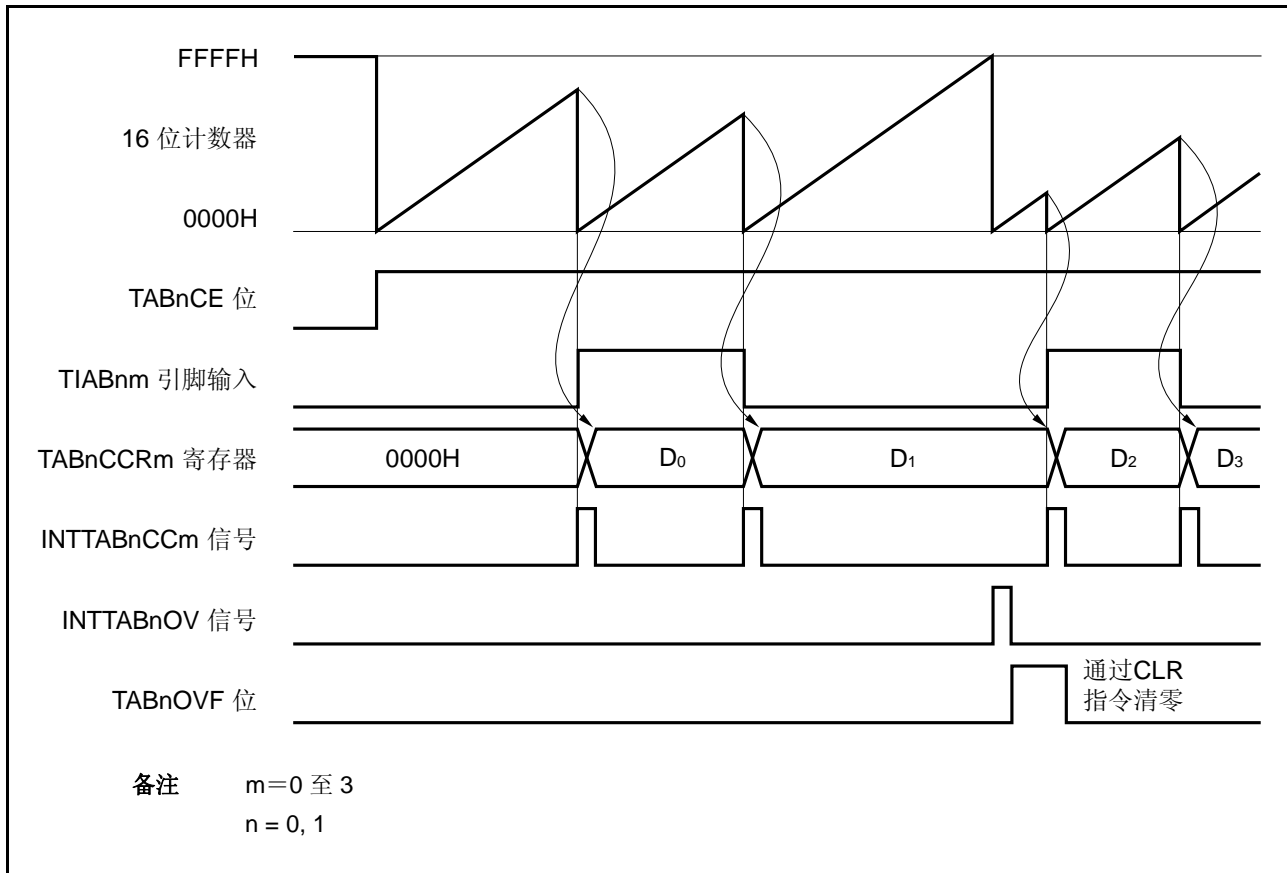


图 8-35. 脉冲宽度测量模式的基本时序



当 $TABnCE$ 位被置为 1 时，16 位计数器开始计数。当检测到 $TIABnm$ 引脚输入信号的有效边沿时，16 位计数器的计数值被存储到 $TABnCCRm$ 寄存器中，16 位计数器清零为 $0000H$ ，并产生一个捕获中断请求信号 ($INTTABnCCm$)。

脉冲宽度的计算如下。

$$\text{脉冲宽度} = \text{捕获值} \times \text{计数时钟周期}$$

当 16 位计数器计数到 $FFFFH$ 时，即使没有有效边沿输入 $TIABnm$ 引脚，在下一个计数时钟时，仍然会产生溢出中断请求信号 ($INTTABnOV$)，且计数器清零为 $0000H$ ，然后计数器继续计数。此时，溢出标志 ($TABnOPT0.TABnOVF$ 位) 也被置为 1。使用软件执行 CLR 指令将溢出标志清为 0。

若溢出标志被设置为 1，则脉冲宽度的计算如下。

$$\text{脉冲宽度} = (10000H \times TABnOVF \text{ 位置位 (1) 的次数} + \text{捕获值}) \times \text{计数时钟周期}$$

备注 $m=0$ 至 3
 $n=0, 1$

图 8-36. 脉冲宽度测量模式时的寄存器设置 (1/2)

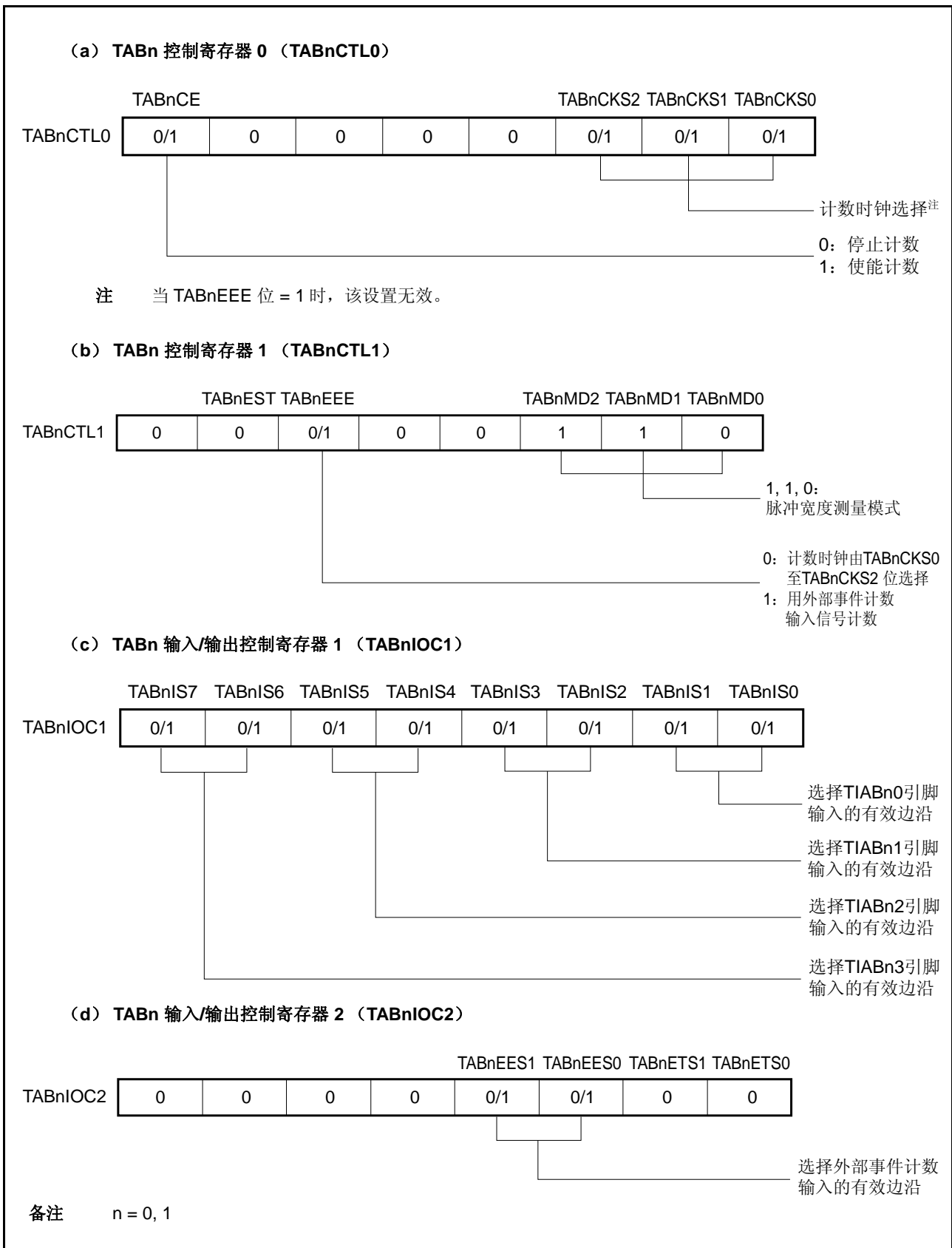
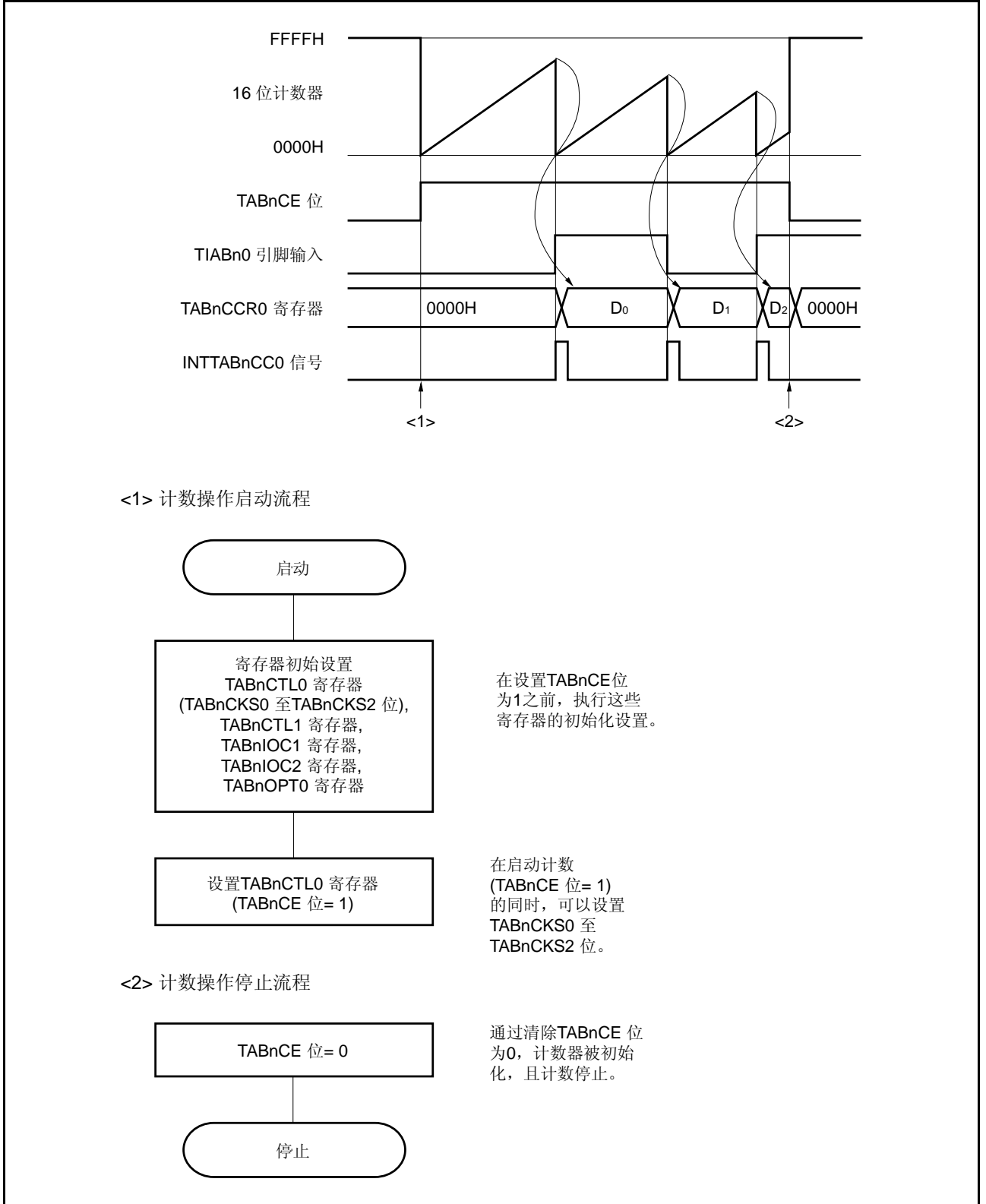


图 8-36. 脉冲宽度测量模式时的寄存器设置 (2/2)



(1) 脉冲宽度测量模式时的操作流程

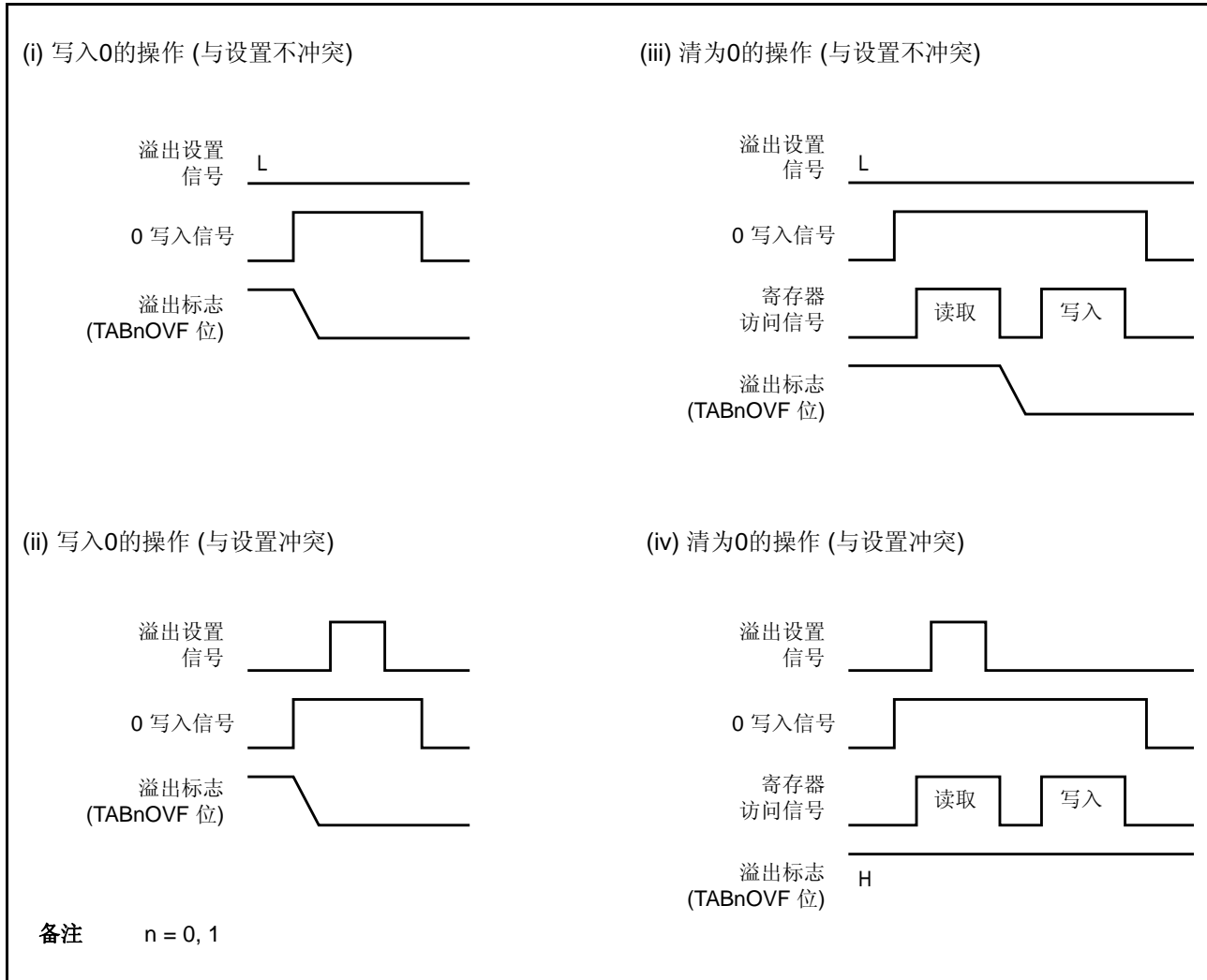
图 8-37. 脉冲宽度测量模式的软件处理流程



(2) 脉冲宽度测量模式时的操作时序

(a) 清除溢出标志

通过 CLR 指令将 TABnOVF 位清为 0，以及将 8 位数据（第 0 位是 0）写入 TABnOPT0 寄存器，可以将溢出标志清 0。要准确检测溢出，当 TABnOVF 位为 1 时读取 TABnOVF 位，然后用位操作指令将溢出标志清零。



要将溢出标志清为 0，应读取溢出标志，检查该标志如果已经被设置为 1，用 CLR 指令把它清零。若在未检查溢出标志是否为 1 就把 0 写入溢出标志中，则溢出的设置信息可能因为写入 0 而被清除（上图中的 (ii)）。因此，即使实际发生了溢出，软件也可能会判断为无溢出发生。

如果用 CLR 指令将溢出标志清为 0 时，CLR 指令的执行与当前发生的溢出相冲突，则在清零指令执行后溢出标志仍然会保持置位状态。

8.5.8 三角波PWM模式 (TABnMD2 至 TABnMD0 位 = 111)

在三角波 PWM 模式下, TABn 捕获/比较寄存器 k (TABnCCRk) 用来设置占空比, TABn 捕获/比较寄存器 0 (TABnCCR0) 用来设置周期。

通过使用这四个寄存器和时钟的操作, 可以输出周期可变的三角波 PWM。

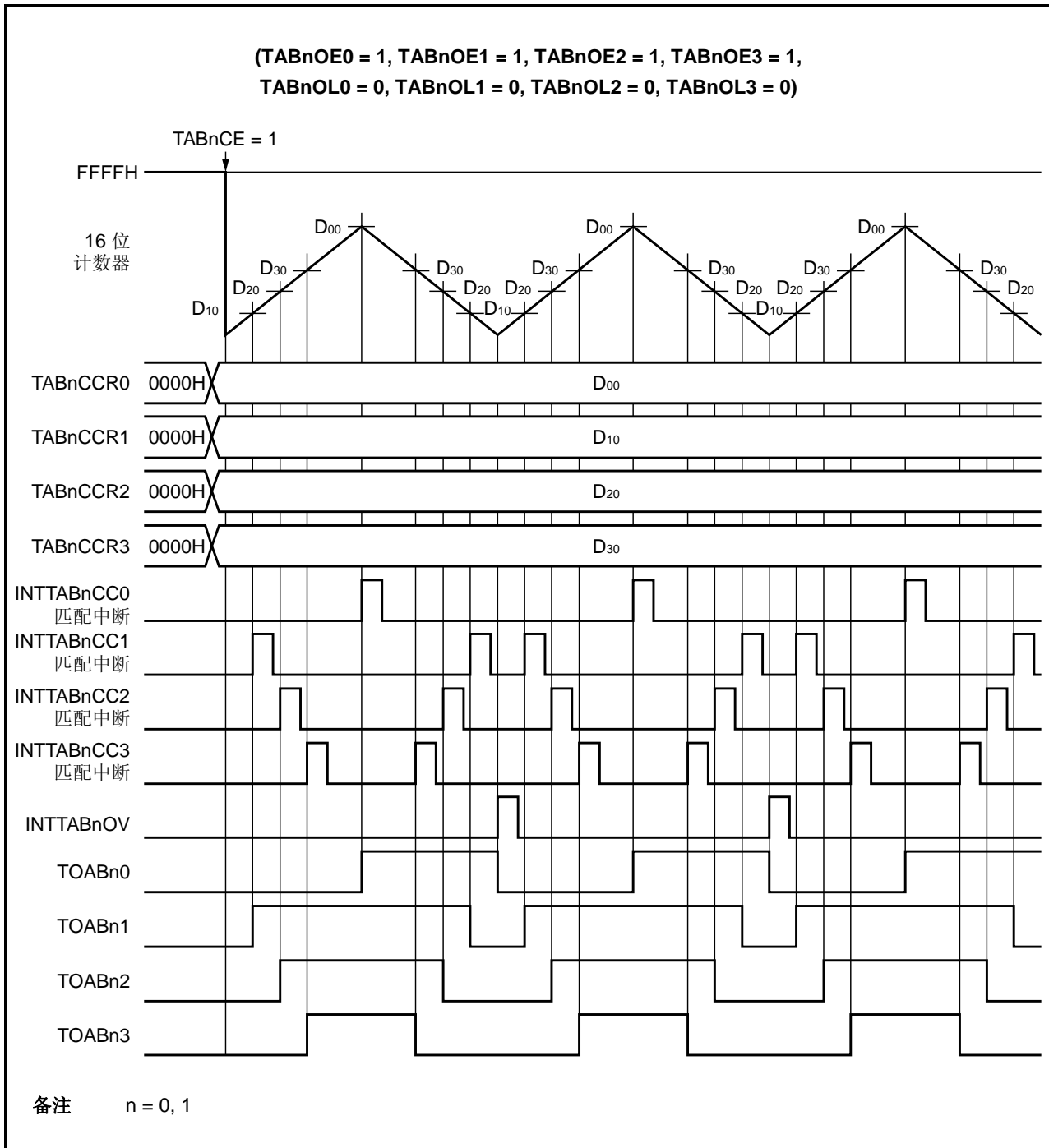
当 TABnCE = 1 时, 可以重写 TABnCCRm 寄存器的值。

要停止定时器 AB, 清除 TABnCE 为 0。PWM 波形从 TOABnk 引脚输出。当 16 位计数器的值和 TABnCCR0 寄存器的值匹配时, 或当计数器下溢时, TOABn0 引脚产生一个切换输出。

注意事项 在 PWM 模式下, TABnCCRm 寄存器的捕获功能不能使用, 因为该寄存器只能用作比较寄存器。

备注 n = 0, 1, m = 0 至 3, k = 1 至 3

图 8-38. 三角波 PWM 模式的基本操作时序



8.5.9 定时器输出操作

下表显示了 TOABn0 至 TOABn3 引脚的操作和输出电平。

表 8-6. 各种模式下的定时器输出控制

操作模式	TOABn0 引脚	TOABn1 引脚	TOABn2 引脚	TOABn3 引脚
间隔计时器模式	方波输出			
外部事件计数模式	方波输出	-		
外部触发脉冲输出模式	方波输出	外部触发脉冲输出	外部触发脉冲输出	外部触发脉冲输出
单次触发脉冲输出模式		单次触发脉冲输出	单次触发脉冲输出	单次触发脉冲输出
PWM 输出模式		PWM (脉宽调制信号) 输出	PWM (脉宽调制信号) 输出	PWM (脉宽调制信号) 输出
自由定时器模式	方波输出 (只有使用比较功能时可用)			
脉冲宽度测量模式	-			
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 8-7. 定时器输出控制位对 TOABn0 至 TOABn3 引脚影响的真值表

TABnIOC0.TABnOLm 位	TABnIOC0.TABnOEm 位	TABnCTL0.TABnCE 位	TOABnm 引脚电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前为低电平, 开始计数后为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平, 开始计数后为低电平

备注 m=0 至 3
n=0, 1

8.6 定时器调谐操作功能/同时启动功能

定时器 AA 和定时器 AB 具有调谐操作功能/同时启动功能。
可以同步的定时器在表 8-8 中列举如下。

表 8-8. 定时器调谐操作模式

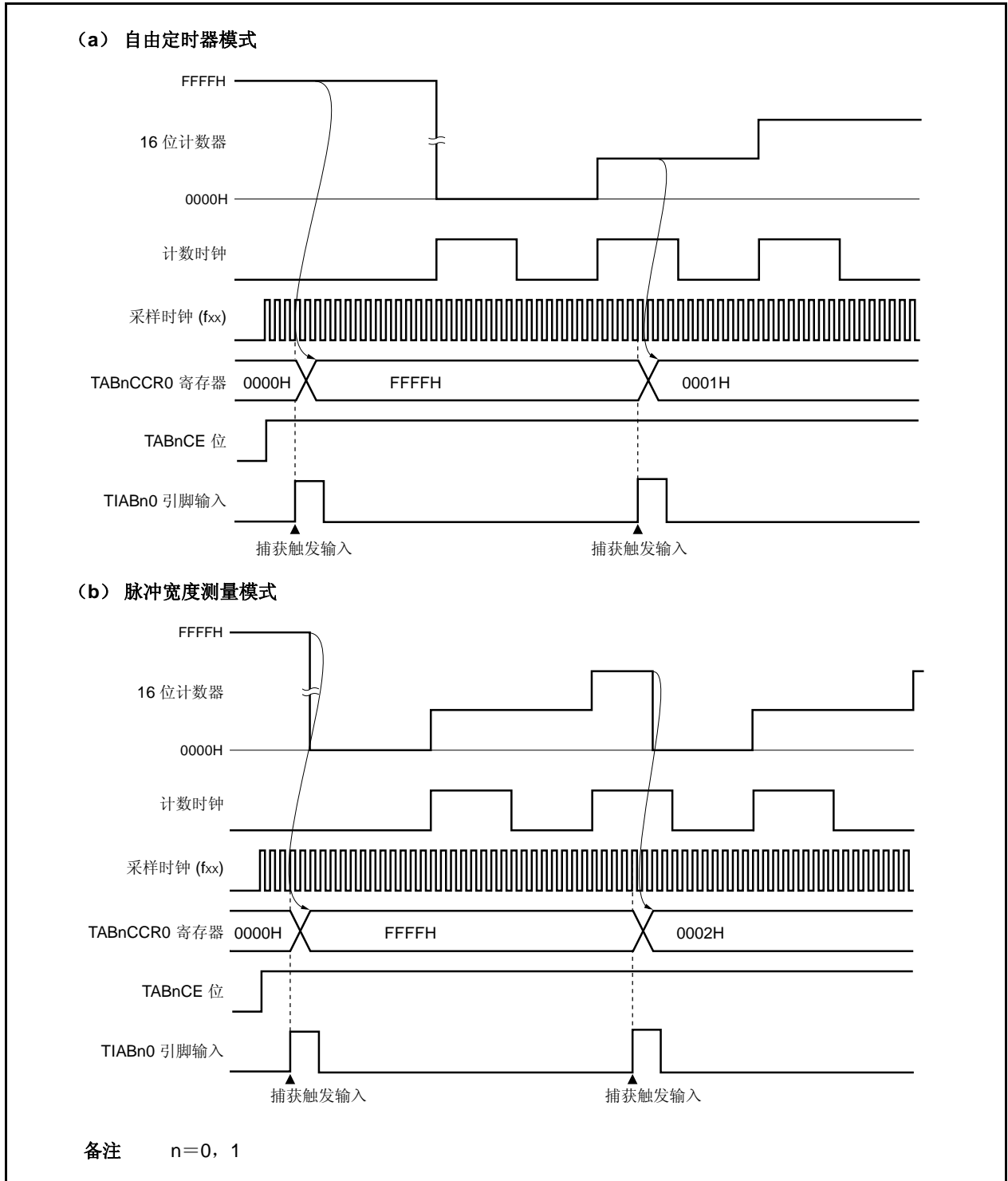
主定时器	从定时器
TAA1	TAA0
TAA3	TAA2
TAB0	TAA5

定时器调谐操作功能的详情参见 7.6 定时器调谐操作功能，同时启动功能的详情参见 7.7 同时启动功能。

8.7 注意事项

(1) 捕获操作

当使用捕获操作，并且选择低速时钟作为计数时钟时，如果在 TABnCE 位置 1 之后立即输入捕获触发，那么捕获到 TABnCCR0, TABnCCR1, TABnCCR2 和 TABnCCR3 寄存器中的可能是 FFFFH 而不是 0000H。



第九章 16 位定时器/事件计数器T (TMT)

定时器 T (TMT) 是 16 位定时器/事件计数器。

在定时器 AA (TMT) 基础上增加了编码器计数功能和其它功能。但是, 当 TMT 运行于内部定时器模式时, 它不具有通过外部事件计数输入进行操作的功能。

V850ES/JG3-U 和 V850ES/JH3-U 单片机内置一路 TMT 通道。

9.1 概述

定时器 TMT0 的概况如下所示:

• 时钟选择:	8 种类型
• 捕获/触发输入引脚(TIT00, TIT01):	2
• 外部事件计数输入引脚(EVTT0):	1
• 编码器输入引脚(TENC00, TENC01):	2
• 编码器清除输入引脚(TECR0):	1
• 外部触发输入引脚 ^注 :	1
• 定时器/计数器:	1
• 捕获/比较寄存器:	2
• 捕获/比较匹配中断请求信号:	2
• 定时器输出引脚:	2

注 外部触发输入引脚和外部事件计数输入引脚(EVTT0)和一个编码器输入引脚(TENC00)共用。

9.2 功能

定时器 TMT0 具有下列功能:

- 间隔定时器
- 外部事件计数器
- 外部触发脉冲输出
- 单次触发脉冲输出
- PWM 输出
- 自由运行定时器
- 脉冲宽度测量
- 三角波 PWM 输出
- 编码器计数

9.3 配置

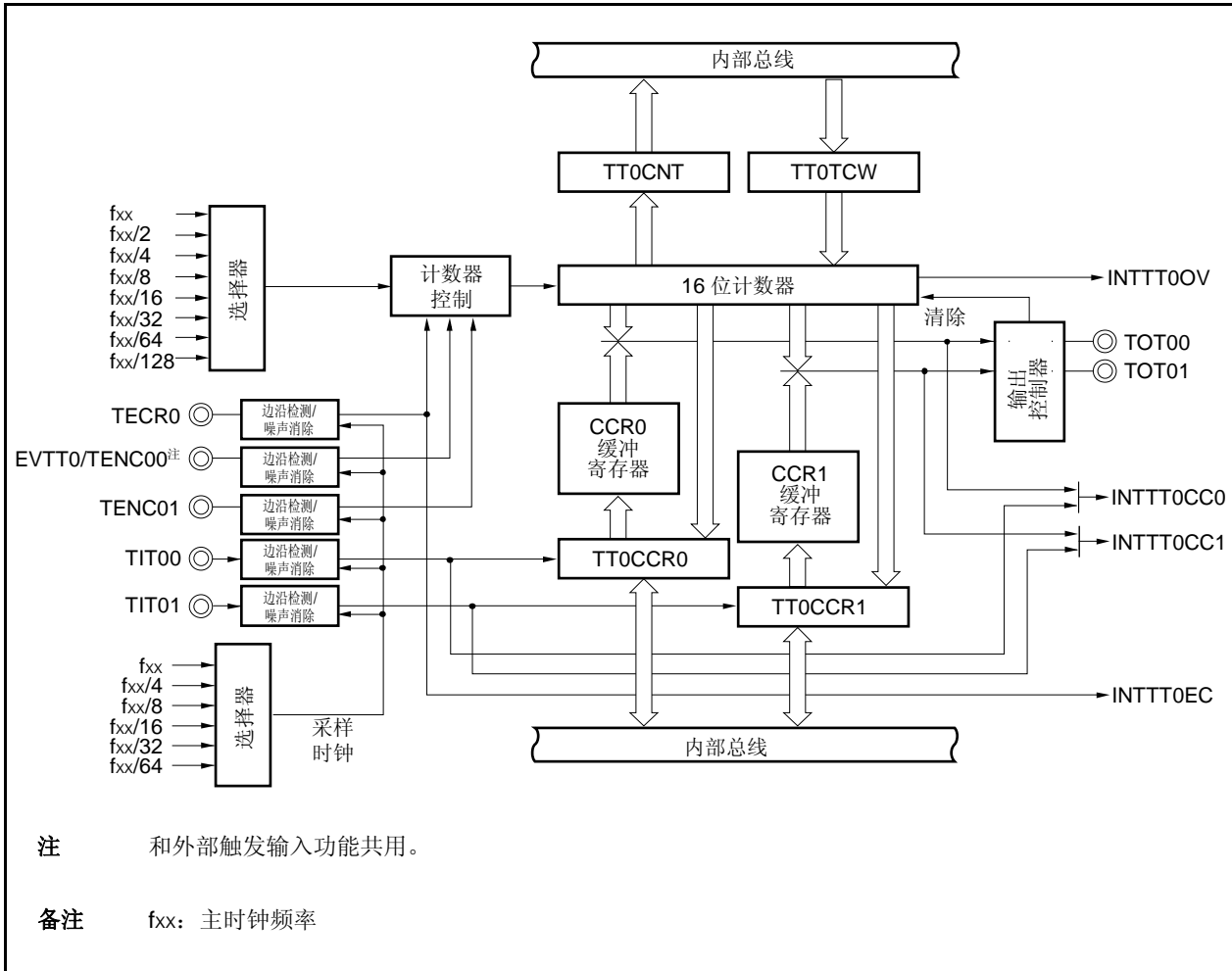
TMT0 包括下列硬件：

表 9-1. TMT0 的配置

项目	配置
寄存器	16 位计数器 TMT0 捕获/比较寄存器 0, 1 (TT0CCR0, TT0CCR1) TMT0 计数器读取缓冲寄存器 (TT0CNT) TMT0 计数写入寄存器 (TT0TCW) CCR0, CCR1 缓冲寄存器 TMT0 控制寄存器 0, 1 (TT0CTL0, TT0CTL1) TMT0 控制寄存器 2 (TT0CTL2) TMT0 输入/输出 控制寄存器 0 至 3 (TT0IOC0 至 TT0IOC3) TMT0 选项寄存器 0 (TT0OPT0) TMT0 选项寄存器 1 (TT0OPT1) TMT 噪声消除控制寄存器 (TTNFC)
定时器输入	<ul style="list-style-type: none"> • TIT00, TIT01 (捕获触发输入引脚) • EVTT0/TENC00 (外部事件输入/编码器 0 输入引脚)^注 • TENC01 (编码器 1 输入引脚) • TENC00 (编码器清除输入引脚)
定时器输出	TOT00, TOT01

注 和外部触发输入功能共用。

图 9-1. TMT0 框图



(1) 16 位计数器

此 16 位计数器可以计数内部时钟或外部事件。

该计数器的计数值可以通过读取 TT0CNT 寄存器得到。

当 TT0CTL0.TT0CE 位= 0 时，16 位计数器的值为 FFFFH。如果此时读取 TT0CNT 寄存器，则读回的值为 0000H。

系统复位将 TT0CE 位清 0。

(2) CCR0 缓冲寄存器

CCR0 缓冲寄存器是一个 16 位比较寄存器，用于比较 16 位计数器的计数值。

当 TT0CCR0 寄存器用作比较寄存器时，写入 TT0CCR0 寄存器的值会传送到 CCR0 缓冲寄存器。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值相匹配时，就会产生一个比较匹配中断请求信号 (INTTT0CC0)。

CCR0 缓冲寄存器不能直接读写。

CCR0 缓冲寄存器在复位后清零为 0000H，TT0CCR0 寄存器清零为 0000H。

(3) CCR1 缓冲寄存器

CCR1 缓冲寄存器是一个 16 位比较寄存器，用于比较 16 位计数器的计数值。

当 TT0CCR1 寄存器用作比较寄存器时，写入 TT0CCR1 寄存器的值会传送到 CCR1 缓冲寄存器。当 16 位计数器的计数值和 CCR1 缓冲寄存器的值匹配时，就会产生一个比较匹配中断请求信号 (INTTT0CC1)。

CCR1 缓冲寄存器不能直接读写。

CCR1 缓冲寄存器在复位后清零为 0000H，TT0CCR1 寄存器清零为 0000H。

(4) 边沿检测器

该电路检测输入到 TIT00, TIT01, EVTT0/TENC00, TENC01 和 TECR0 引脚信号的有效边沿。通过设置 TT0IOC1, TT0IOC2 和 TT0IOC3 寄存器，可以选择各种有效沿：无边沿、上升沿、下降沿、上升/下降双边沿作为有效沿。

(5) 输出控制器

该电路控制 TOT00 和 TOT01 引脚的输出。输出控制器由 TT0IOC0 寄存器控制。

(6) 选择器

此选择器选择 16 位计数器的计数时钟。可选择 8 种内部时钟或一个外部事件作为计数时钟。

(7) 计数器控制

通过 TT0CTL1 寄存器选择的定时器模式来控制计数操作。

9.3.1 引脚配置

TMT0 配置的定时器输入和输出与下列端口共用。当使用各个引脚时，必须设置端口功能 (参见表 4-20 端口引脚用作复用功能引脚)。

表 9-2. 引脚配置

端口	定时器输入引脚		定时器输出	其它复用功能
P92	TIT01 (捕获触发输入 1)	TENC01 (编码器输入)	TOT01	A2 ^{注1}
P93	TIT00 (捕获触发输入 0)	TECR0 (编码器清除输入)	TOT00	A3 ^{注1}
P94	EVTT0/TENC00 ^{注2}			

- 注
1. 仅限 V850ES/JH3-U
 2. 通过端口功能无法控制的状态下，外部事件计数输入(EVTT0)，编码器输入 (TENC00)，和外部触发输入共用。要使用各个功能，在设置他们对应的端口之后，通过 TT0IOC2 和 TT0IOC3 寄存器来设置。

9.4 寄存器

(1) TMT0 控制寄存器 0 (TT0CTL0)

TT0CTL0 寄存器是 8 位寄存器，用于控制定时器 TMT0 的操作。

该寄存器可以按字节读写或按位读写。

复位输入将该寄存器设为 00H。

随时可以用软件将相同的值写入 TT0CTL0 寄存器。

复位后: 00H R/W 地址: FFFFF600H

	7	6	5	4	3	2	1	0
TT0CTL0	TT0CE	0	0	0	0	TT0CKS2	TT0CKS1	TT0CKS0

TT0CE	TMT0 操作控制
0	TMT0 操作禁止 (TMT0 异步复位 ^注)
1	TMT0 操作使能, TMT0 操作开始。

TT0CKS2	TT0CKS1	TT0CKS0	内部计数时钟选择
0	0	0	f _{xx}
0	0	1	f _{xx} /2
0	1	0	f _{xx} /4
0	1	1	f _{xx} /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

注 TT0OPT0.TT0OVF 位和 16 位计数器被同时重置。而且，定时器输出 (TOT00 和 TOT01) 和 16 位计数器同时被重置。

- 注意事项
1. 当 TT0CE 位=0 时，设置 TT0CKS2 至 TT0CKS0 位为“1”。
当 TT0CE 位的值从 0 变为 1 时，可以同时设定 TT0CKS2 至 TT0CKS0 位。
 2. 请确保将第 3 位至第 6 位清为“0”。

备注 f_{xx}: 外设时钟

(2) TMT0 控制寄存器 1 (TT0CTL1)

TT0CTL1 寄存器是 8 位寄存器，用于控制定时器 TMT0 的操作。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

(1/2)

复位后：00H R/W 地址： FFFFF601H

	7	6	5	4	3	2	1	0
TT0CTL1	0	TT0EST	TT0EEE	0	TT0MD3	TT0MD2	TT0MD1	TT0MD0

TT0EST	软件触发控制
0	-
1	为外部触发输入产生一个有效信号 <ul style="list-style-type: none"> 在单次触发脉冲输出模式下： 向 TT0EST 位写入 1 作为触发信号，输出一个单次触发脉冲。 在外部触发脉冲输出模式下： 向 TT0EST 位写入 1 作为触发信号，输出一个 PWM 波形。
TT0EST 位的读取值总是为 0。	

TT0EEE	计数时钟选择
0	禁止外部事件计数输入的操作 (EVTTO 引脚) (执行对计数时钟的计数，该时钟由 TT0CTL0.TT0CK0 至 TT0CK2 位选择)
1	使能外部事件计数输入的操作 (EVTTO 引脚) (对外部事件计数输入信号 (EVTTO 引脚) 的有效边沿执行计数)
通过 TT0EEE 位选择使用内部计数时钟计数， 还是使用外部事件计数输入的有效边沿来计数。	

TT0MD3	TT0MD2	TT0MD1	TT0MD0	定时器模式选择
0	0	0	0	间隔定时器模式
0	0	0	1	外部事件计数模式
0	0	1	0	外部触发脉冲输出模式
0	0	1	1	单次触发脉冲输出模式
0	1	0	0	PWM 输出模式
0	1	0	1	自由运行定时器模式
0	1	1	0	脉宽测量模式
0	1	1	1	三角波 PWM 输出模式
1	0	0	0	编码器比较模式
其它				禁止设置

- 注意事项
1. 只有在外触发脉冲输出模式或单次触发脉冲输出模式下，TT0EST 位有效。在其它模式下，向该位写入 1 的操作都被忽略。
 2. 只有在间隔定时器模式、外部触发脉冲输出模式、单次触发脉冲输出模式、PWM 输出模式、自由运行定时器模式、脉宽测量模式或三角波 PWM 输出模式下，TT0EEE 位才有效。在其它模式下，向该位写入 1 的操作都被忽略。
 3. 在外部事件计数模式或编码器比较模式下，无论 TT0EEE 位的值如何，总是选择外部事件计数输入 (EVTTO) 或编码器输入 (TENC00, TENC01)。
 4. TT0CTL0.TT0CE 位= 0 时，设置 TT0EEE 和 TT0MD3 至 TT0MD0 位 (TT0CE 位= 1 时，可以写入相同的值)。在 TT0CE 位= 1 时，重写操作是无法保证的，若重写发生错误，应把 TT0CE 位清为 0，再重新对这些位进行设定。
 5. 请确保将第 4 位和第 7 位清为“0”。

(3) TMT0 控制寄存器 2 (TT0CTL2)

TT0CTL2 寄存器是 8 位寄存器，用于控制编码器计数功能的操作。

只有在编码器比较模式下，TT0CTL2 寄存器有效。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

注意事项 TT0CTL2 寄存器各个位的详情参见 9.6.9 (5) TT0CTL2 寄存器的控制位。

(1/2)

复位后: 00H		R/W	地址: FFFF602H					
	7	6	5	4	3	2	1	0
TT0CTL2	TT0ECC	0	0	TT0LDE	TT0ECM1	TT0ECM0	TT0UDS1	TT0UDS0
TT0ECC	编码器计数器控制							
0	正常操作							
1	当TT0CTL0.TT0CE 位 = 0时，保持16位计数器的计数值。							
TT0LDE	16位计数器的传输设置							
0	在下溢时，禁止将TT0CCR0的设置值传输给16位计数器。							
1	在下溢时，允许将TT0CCR0的设置值传输给16位计数器。							
TT0ECM1	编码器清除操作1的控制							
0	当计数值和CCR1寄存器的值匹配时， 16位计数器不被清除为0000H。							
1	在计数值和CCR1寄存器的值匹配之后， 如果计数是向下计数，则16位计数器被清除为0000H。							
TT0ECM0	编码器清除操作0的控制							
0	当计数值和CCR0寄存器的值匹配时， 16位计数器不被清除为0000H。							
1	在计数值和CCR0寄存器的值匹配之后， 如果计数是向上计数，则16位计数器被清除为0000H。							

TT0UDS1	TT0UDS0	上/下计数选择
0	0	检测到TENC00 输入的有效边沿时 当TENC01 = 高电平，向下计数。 当TENC01 = 低电平，向上计数。
0	1	检测到TENC00 输入的有效边沿时，向上计数。 检测到TENC01 输入的有效边沿时，向下计数。
1	0	检测到TENC00 输入的上升沿时，向下计数。 检测到TENC00 输入的下降沿时，向上计数。 但是，只有当TENC01 = 低电平时，才执行计数操作。
1	1	TENC00 和TENC01的上升沿和下降沿都检测。 计数操作通过边沿检测和电平检测的组合来自动识别。

- 注意事项**
1. 只有在编码器比较模式下，TT0ECC 位有效。在其它模式下，向该位写入 1 的操作都被忽略。
如果在 TT0ECC 位=1 时，TT0CTL0.TT0CE 位被清除为 0，则定时器/计数器和捕获寄存器 (TT0CCR0 和 TT0CCR1)的值，以及 TT0OPT1, TT0EUF, TT0EOF 和 TT0ESF 标志被保留。
如果在 TT0ECC 位=1 时，TT0CE 位从 0 置为 1，TT0TCW 寄存器的值不会被传输到 16 位计数器。
 2. 只有当 TT0ECM1 和 TT0ECM0 位= 00, 01 时，TT0LDE 位有效。当 TT0ECM1 和 TT0ECM0 位= 10, 11 时，向该位写入“1”的操作都被忽略。
 3. 由 TT0IOC3.TT0EIS1 和 TT0IOC3.TT0EIS0 位指定的 TENC00 和 TENC01 输入边沿检测无效，且当 TT0UDS1 和 TT0UDS0 位= 10, 11 时，固定为双沿检测。
 4. TT0CTL0.TT0CE 位 = 0 时，设置 TT0LDE, TT0ECM1, TT0ECM0, TT0UDS1, 和 TT0UDS0 位 (TT0CE 位= 1 时，可以向这些位写入相同的值)。如果在 TT0CE 位 = 1 时更改这些位的值，操作无法保证。若因错误被更改，应把 TT0CE 位清为 0，再重新对这些位进行设定。
 5. 请确保将第 5 位和第 6 位清为“0”。

(4) TMT0 输入/输出控制寄存器 0 (TT0IOC0)

TT0IOC0 寄存器是 8 位寄存器，用于控制定时器的输出 (TOT00, TOT01 引脚)。

该寄存器可按字节读写或按位读写。

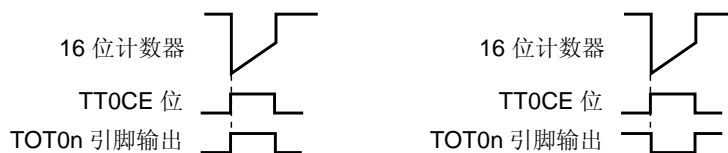
复位输入将该寄存器设为 00H。

复位后: 00H		R/W	地址: FFFFF603H					
	7	6	5	4	3	<2>	1	<0>
TT0IOC0	0	0	0	0	TT0OL1	TT0OE1	TT0OL0	TT0OE0
TT0OL1	TOT01 引脚输出电平设置 ^注							
0	TOT01 引脚输出从高电平开始							
1	TOT01 引脚输出从低电平开始							
TT0OE1	TOT01 引脚输出设置							
0	定时器输出禁止 • 当 TT0OL1 位 = 0 时, 从 TOT01 引脚输出低电平。 • 当 TT0OL1 位 = 1 时, 从 TOT01 引脚输出高电平。							
1	定时器输出使能 (从 TOT01 引脚输出一个脉冲)							
TT0OL0	TOT00 引脚输出电平设置 ^注							
0	TOT00 引脚输出从高电平开始							
1	TOT00 引脚输出从低电平开始							
TT0OE0	TOT00 引脚输出设置							
0	定时器输出禁止 • 当 TT0OL0 位 = 0 时, 从 TOT00 引脚输出低电平。 • 当 TT0OL0 位 = 1 时, 从 TOT00 引脚输出高电平。							
1	定时器输出使能 (从 TOT00 引脚输出一个脉冲)							

注 由 TT0OLn 位指定的定时器输出引脚 (TOT00 和 TOT01) 的输出电平如下所示。(n = 0, 1)

• 当 TT0OLn 位 = 0

• 当 TT0OLn 位 = 1



- 注意事项**
1. 当 TOT00 和 TOT01 输出被设置为端口模式, 如果 TT0IOC0 寄存器的设置更改, 引脚的输出改变。设置该端口为输入模式且置为高阻状态, 引脚状态不会有任何改变。
 2. 当 TT0CTL0.TT0CE 位=0 时, 重新写入 TT0OL1, TT0OE1, TT0OL0 和 TT0OE0 位 (当 TT0CE 位=1 时, 可以写入相同值)。如果重新写入不能正确执行, 将 TT0CE 位清为 0, 然后重新设定该位。
 3. 即使在 TT0CE, TT0OE0 和 TT0OE1 位均为 0 时对 TT0OL0 或 TT0OL1 位进行操作, TOT00 和 TOT01 引脚的输出电平也会改变。

(5) TMT0 输入/输出控制寄存器 1 (TT0IOC1)

TT0IOC1 寄存器是 8 位寄存器，用于控制捕获触发输入信号 (TIT00, TIT01 引脚) 的有效边沿。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF604H

	7	6	5	4	3	2	1	0
TT0IOC1	0	0	0	0	TT0IS3	TT0IS2	TT0IS1	TT0IS0

TT0IS3	TT0IS2	捕获触发输入信号 (TIT01 引脚) 有效边沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双沿

TT0IS1	TT0IS0	捕获触发输入信号 (TIT00 引脚) 有效边沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双沿

- 注意事项**
1. 当 TT0CTL0.TT0CE 位 = 0 时，重新写入 TT0IS3 至 TT0IS0 位（当 TT0CE 位 = 1 时，可以写入相同值）。如果重新写入不能正确执行，将 TT0CE 位清为 0，然后重新进行位的设定。
 2. 只有在自由运行定时器模式(仅当 TT0OPT0.TT0CCS1 位= 1)和脉冲宽度测量模式下，TT0IS3 和 TT0IS2 位才有效。在所有其它模式下，无法进行捕获操作。
 3. 只有在自由运行定时器模式(仅当 TT0OPT0.TT0CCS0 位= 1)和脉冲宽度测量模式下，TT0IS1 和 TT0IS0 位才有效。在所有其它模式下，无法进行捕获操作。

(6) TMT0 输入/输出控制寄存器 2 (TT0IOC2)

TT0IOC2 寄存器是 8 位寄存器，用于控制外部事件计数器输入信号（EVTTO 引脚）和外部触发输入信号（EVTTO 引脚）的有效边沿。

该寄存器可按字节字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后：00H R/W 地址：FFFFF605H

	7	6	5	4	3	2	1	0
TT0IOC2	0	0	0	0	TT0EES1	TT0EES0	TT0ETS1	TT0ETS0

TT0EES1	TT0EES0	外部事件计数输入信号(EVTTO 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	检测双沿

TT0ETS1	TT0ETS0	外部触发输入信号 (EVTTO 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	检测双沿

- 注意事项**
1. 当 TT0CTL0.TT0CE 位 = 0 时，重新写入 TT0EES1, TT0EES0, TT0ETS1 和 TT0ETS0 位（当 TT0CE 位 = 1，可以写入相同的值）。如果重新写入不能正确执行，将 TT0CE 位清为 0，然后重新进行位的设定。
 2. 只有当 TT0CTL1.TT0EEE 位 = 1 或设置为外部事件计数模式（TT0CTL1.TT0MD3 至 TT0CTL1.TT0MD0=0001）时，TT0EES1 和 TT0EES0 位有效。
 3. 只有在外部触发脉冲输出模式或在单次触发脉冲输出模式下，TT0ETS1 和 TT0ETS0 位有效。

(7) TMT0 输入/输出控制寄存器 3 (TT0IOC3)

TT0IOC3 寄存器是 8 位寄存器，用于控制编码器清除功能的操作。

只有在编码器比较模式下，TT0IOC3 寄存器有效。

该寄存器可按字节字节读写或按位读写。

复位输入将该寄存器设为 00H。

(1/2)

复位后: 00H		R/W	地址: FFFFF606H					
	7	6	5	4	3	2	1	0
TT0IOC3	TT0SCE	TT0ZCL	TT0BCL	TT0ACL	TT0ECS1	TT0ECS0	TT0EIS1	TT0EIS0
TT0SCE	编码器清除选择							
0	检测到编码器清除信号的边沿时，清除 16 位计数器。(TECR0 引脚)							
1	检测到 TENC00, TENC01 和 TECR0 引脚的清除电平条件时，清除 16 位计数器。							
<ul style="list-style-type: none"> 当 TT0SCE 位 = 0 时，检测到 TT0ECS1 和 TT0ECS0 位指定的 TECR0 引脚有效边沿，则清除 16 位计数器为 0000H。 当 TT0SCE 位 = 1 时，TT0ZCL, TT0BCL 和 TT0ACL 位的清除电平条件和 TECR0, TENC01 和 TENC00 引脚的输入电平匹配，则清除 16 位计数器为 0000H。 当 TT0SCE 位 = 1 时，TT0ZCL, TT0BCL 和 TT0ACL 位的设置有效，且 TT0ECS1 和 TT0ECS0 位的设置无效。不产生编码器清除中断请求信号 (INTTT0EC)。 当 TT0SCE 位 = 0 时，TT0ZCL, TT0BCL 和 TT0ACL 位的设置无效，且 TT0ECS1 和 TT0ECS0 位的设置有效。当检测到 TT0ECS1 和 TT0ECS0 位指定的有效信边沿时，产生 INTTT0EC 信号。 当 TT0SCE 位 = 1 时，请确保设置 TT0CTL2.TT0UDS1 和 TT0CTL2.TT0UDS0 位为 10 或 11。如果 TT0UDS1 和 TT0UDS0 位 = 00 或 01 且 TT0SCE 位 = 1，则操作无法保证。 								
TT0ZCL	编码器清除信号的清除电平选择 (TECR0 引脚)							
0	TECR0 引脚低电平时清除。							
1	TECR0 引脚高电平时清除。							
仅当 TT0SCE 位 = 1 时，TT0ZCL 位的设置有效。								
TT0BCL	编码器输入信号的清除电平选择 (TENC01 引脚)							
0	TENC01 引脚低电平时清除。							
1	TENC01 引脚高电平时清除。							
仅当 TT0SCE 位 = 1 时，TT0BCL 位的设置有效。								
TT0ACL	编码器输入信号的清除电平选择 (TENC00 引脚)							
0	TENC00 引脚低电平时清除。							
1	TENC00 引脚高电平时清除。							
仅当 TT0SCE 位 = 1 时，TT0ACL 位的设置有效。								

TT0ECS1	TT0ECS0	编码器清除信号 (TECR0 引脚)的有效边沿设置
0	0	无边沿检测 (清除编码器无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双沿

TT0EIS1	TT0EIS0	编码器输入信号 (TENC00, TENC01 引脚) 的有效边沿设置
0	0	无边沿检测 (编码器输入无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双沿

- 注意事项**
1. 当 **TT0CTL0.TT0CE** 位 = 0 时, 重新写入 **TT0SCE**, **TT0ZCL**, **TT0BCL**, **TT0ACL**, **TT0ECS1**, **TT0ECS0**, **TT0EIS1** 和 **TT0EIS0** 位 (当 **TT0CE** 位 = 1, 可以写入相同的值)。如果重新写入不能正确执行, 将 **TT0CE** 位清为 0, 然后重新进行位的设定。
 2. 只有当 **TT0SCE** 位 = 0 且设置为编码器比较模式时, **TT0ECS1** 和 **TT0ECS0** 位有效。
 3. 只有当 **TT0CTL2.TT0UDS1** 和 **TT0CTL2.TT0UDS0** 位 = 00 或 01 时, **TT0EIS1** 和 **TT0EIS0** 位有效。

(8) TMT0 选项寄存器 0 (TT0OPT0)

TT0OPT0 寄存器为 8 位寄存器，用来设置捕获/比较操作和检测计数器的溢出。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF607H

	7	6	5	4	3	2	1	<0>
TT0OPT0	0	0	TT0CCS1	TT0CCS0	0	0	0	TT0OVF

TT0CCS1	TT0CCR1 寄存器捕获/比较选择
0	选择比较寄存器
1	选择捕获寄存器 (通过 TT0CTL0.TT0CE 位 = 0 来清除)
只有在自由运行定时器模式下，TT0CCS1 位的设置有效。	

TT0CCS0	TT0CCR0 寄存器捕获/比较选择
0	选择比较寄存器
1	选择捕获寄存器 (通过 TT0CTL0.TT0CE 位 = 0 来清除)
只有在自由运行定时器模式下，TT0CCS0 位的设置有效。	

TT0OVF	TMT0 溢出检测标志
置位 (1)	发生溢出
复位 (0)	TT0OVF 位写入 0，或 TT0CTL0.TT0CE 位 = 0
<ul style="list-style-type: none"> 在自由运行定时器模式或脉宽测量模式下，当 16 位计数器的计数值从 FFFFH 溢出到 0000H 时，TT0OVF 位被置为 1。 当 TT0OVF 位被置为 1 时，产生一个溢出中断请求信号 (INTTT0OV)。在自由运行定时器模式和脉宽测量模式之外的模式下，不产生 INTTT0OV 信号。 当 TT0OVF 位 = 1 时，即使 TT0OVF 位或 TT0OPT0 寄存器被读取，TT0OVF 位也不会被清除。 在产生 INTTT0OV 信号之后，将 TT0OVF 位清除为 0 前，请确认 (通过读取) TT0OVF 位被置为 1。 TT0OVF 位可读可写，但是不能通过软件将 TT0OVF 位位置 1。在 TMT0 工作期间写入 1 没有影响。 	

注意事项 1. 当 TT0CTL0.TT0CE 位 = 0 时，重新写入 TT0CCS1 和 TT0CCS0 位 (当 TT0CE 位 = 1 时，可以写入相同的值)。如果重新写入不能正确执行，将 TT0CE 位清为 0，然后重新进行位的设定。

2. 请确保将第 1 位至第 3 位，第 6 位和第 7 位清为“0”。

(9) TMT0 选项寄存器 1 (TT0OPT1)

TT0OPT1 寄存器为 8 位寄存器，用来检测溢出，下溢，编码器计数功能的向上/向下计数操作。

只有在编码器比较模式下，TT0OPT1 寄存器有效。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

即使当 TT0CTL0.TT0CE 位 = 1 时，可以重写该寄存器。

(1/2)

复位后: 00H R/W 地址: FFFFF608H

	7	6	5	4	3	<2>	<1>	<0>
TT0OPT1	0	0	0	0	0	TT0EUF	TT0EOF	TT0ESF

TT0EUF	TMT0 下溢检测标志
置位 (1)	发生下溢
复位 (0)	写入 TT0EUF 位来清除，或当 TT0CTL0.TT0CE 位 = 0 时清除
<ul style="list-style-type: none"> 在编码器比较模式下，当 16 位计数器从 0000H 下溢到 FFFFH 时，TT0EUF 位被置为 1。 当 TT0CTL2.TT0LDE 位 = 1，16 位计数器的值从 0000H 更改为 TT0CCR0 寄存器的设置值时，TT0EUF 位被置为 1。 当 TT0EUF 位被置为 1 的同时，产生一个溢出中断请求信号 (INTTTOV0)。 当 TT0EUF 位 = 1 时，即使 TT0EUF 位或 TT0OPT1 寄存器被读取，TT0EUF 位也不会被清除。 当 TT0CTL2.TT0ECC 位 = 1 时，即使 TT0CTL0.TT0CE 位被清除为 0，TT0EUF 位的状态被保留。 在产生 INTTTOV0 信号之后，将 TT0EUF 位清除为 0 前，请确认 (通过读取) TT0EUF 位被置为 1。 TT0EUF 位可读可写，但是不能通过软件将该位置 1。在 TMT0 工作期间设置该位没有影响。 	

TT0EOF	TMT0 编码器功能的溢出检测标志
置位 (1)	发生溢出
重置 (0)	向 TT0EOF 位写入 0 来清除，或当 TT0CTL0.TT0CE 位 = 0 时清除。
<ul style="list-style-type: none"> 在编码器比较模式下，当 16 位计数器从 FFFFH 溢出为 0000H 时，TT0EOF 位被置为 1。 当 TT0EOF 位被置为 1 的同时，产生一个溢出中断请求信号 (INTTTOV0)，同时 TT0OPT0.TT0OVF 位不被置 1。 当 TT0EOF 位 = 1 时，即使 TT0EOF 位或 TT0OPT1 寄存器被读取，TT0EOF 位也不会被清除。 当 TT0CTL2.TT0ECC 位 = 1 时，即使 TT0CTL0.TT0CE 位被清除为 0，TT0EOF 位的状态被保留。 在产生 INTTTOV0 信号之后，将 TT0EOF 位清除为 0 前，请确认 (通过读取) TT0EOF 位被置为 1。 TT0EOF 位可读可写，但是不能通过软件将该位置 1。在 TMT0 工作期间对该位写入 1 没有影响。 	

TT0ESF	TMT0 向上/向下操作状态检测标志
0	TMT0 向上计数
1	TMT0 向下计数
<ul style="list-style-type: none"> 当 TT0CTL2.TT0ECC 位 = 0 时，如果 TT0CTL0.TT0CE 位 = 0，则该位被清除为 0。 当 TT0ECC 位 = 1 时，即使 TT0CE 位 = 0，TT0ESF 位的状态也会被保留。 	

注意事项 请确保将第 3 位至第 7 位清为“0”。

(10) TMT0 捕获/比较寄存器 0 (TT0CCR0)

TT0CCR0 寄存器是 16 位寄存器，根据不同模式，TT0CCR0 寄存器可以被用作捕获寄存器或比较寄存器。

只有在自由计时器模式下，根据 TT0OPT0.TT0CCS0 位的设定，此寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TT0CCR0 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

TT0CCR0 寄存器可以在运行中进行读取和写入。

此寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

复位后: 00H R/W 地址: FFFFF60AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0CCR0																

(a) 用作比较寄存器时的功能

即便在 TT0CTL0.TT0CE 位=1 时，TT0CCR0 寄存器仍可以重新写入。

TT0CCR0 寄存器的设定值传送给 CCR0 缓冲寄存器。当 16 位计数器的值与 CCR0 缓冲寄存器的值相匹配时，就会生成一个比较匹配中断请求信号 (INTTT0CC0)。如果此时 TOT00 引脚输出允许，则 TOT00 引脚的输出会反相。

当 TT0CCR0 寄存器在间隔定时器模式下作为周期寄存器时，或在外部事件计数器模式、外部触发脉冲输出模式、单次触发脉冲输出模式、PWM 输出模式三角波 PWM 输出模式或编码器比较模式下，当 TT0CCR0 寄存器作为周期寄存器时，如果 16 位计数器的计数值与 CCR0 缓冲寄存器数值匹配，则计数器清零为 (0000H)。

设置 TT0CTL0.TT0CE 位为 0，比较寄存器不会被清除。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下 (当 TT0CCR0 寄存器用作捕获寄存器时)，若检测到捕获触发输入引脚 (TIT00 引脚) 的有效边沿，则 16 位计数器的当前计数值存储于 TT0CCR0 寄存器中。在脉宽测量模式下，若检测到捕获触发输入引脚 (TIT00 引脚) 的有效边沿，16 位计数器的当前计数值就储存在 TT0CCR0 寄存器中，且 16 位计数器清零 (0000H)。

即使捕获操作和读取 TT0CCR0 寄存器的操作发生冲突，还是可以正确读取 TT0CCR0 寄存器的值。

设置 TT0CTL0.TT0CE 位为 0，捕获寄存器被清除。

下表显示了捕获/比较寄存器在各工作模式下的功能，以及如何向比较寄存器写入数据。

表 9-3. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

操作模式	TT0CCR0 寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	随时写入
外部事件计数器	比较寄存器	随时写入
外部触发脉冲输出	比较寄存器	批量写入 ^注
单次触发脉冲输出	比较寄存器	随时写入
PWM (脉宽调制信号) 输出	比较寄存器	批量写入 ^注
自由运行定时器	捕获/比较寄存器	随时写入
脉冲宽度测量	捕获寄存器	-
三角波 PWM 输出	比较寄存器	批量写入 ^注
编码器比较	比较寄存器	随时写入

注 写入 TT0CCR1 寄存器作为触发。

备注 任意时刻写入和批量写入，参见 9.6 (2) 任意时刻写入和批量写入。

(11) TMT0 捕获/比较寄存器 1 (TT0CCR1)

TT0CCR0 寄存器是 16 位寄存器，根据不同模式，TT0CCR1 寄存器可以用作捕获寄存器或比较寄存器。

在自由计时器模式下，根据 TT0OPT0.TT0CCS1 位的设定，此寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TT0CCR1 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

TT0CCR1 寄存器可以在运行中进行读取和写入。

此寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

复位后：00H R/W 地址：FFFFFF60CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0CCR1																

(a) 用作比较寄存器时的功能

即便在 TT0CTL0.TT0CE 位= 1 时，TT0CCR1 寄存器仍可以重新写入。

TT0CCR1 寄存器的设定值传送给 CCR1 缓冲寄存器。当 16 位计数器的值与 CCR1 缓冲寄存器的值相匹配时，就会生成一个比较匹配中断请求信号（INTTT0CC1）。如果此时 TOT01 引脚输出允许，则 TOT01 引脚的输出会反相。

设置 TT0CTL0.TT0CE 位为 0，比较寄存器不会被清除。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下（当 TT0CCR1 寄存器用作捕获寄存器时），若检测到捕获触发输入引脚（TIT01 引脚）的有效边沿，则 16 位计数器的当前计数值存储于 TT0CCR1 寄存器中。在脉宽测量模式下，若检测到捕获触发输入引脚（TIT01 引脚）的有效边沿，16 位计数器的当前计数值就储存在 TT0CCR1 寄存器中，且 16 位计数器清零（0000H）。

即使捕获操作和读取 TT0CCR1 寄存器的操作发生冲突，还是可以正确读取 TT0CCR1 寄存器的值。

设置 TT0CTL0.TT0CE 位为 0，捕获寄存器被清除。

下表显示了捕获/比较寄存器在各工作模式下的功能，以及如何向比较寄存器写入数据。

表 9-4. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

操作模式	TT0CCR1 寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	随时写入
外部事件计数器	比较寄存器	随时写入
外部触发脉冲输出	比较寄存器	批量写入
单次触发脉冲输出	比较寄存器	随时写入
PWM（脉宽调制信号）输出	比较寄存器	批量写入
自由运行定时器	捕获/比较寄存器	随时写入
脉冲宽度测量	捕获寄存器	—
三角波 PWM 输出	比较寄存器	批量写入 ^注
编码器比较	比较寄存器	随时写入

注 写入 TT0CCR1 寄存器作为触发。

备注 任意时刻写入和批量写入，参见 9.6 (2) 任意时刻写入和批量写入。

(12) TMT0 计数器写入寄存器 (TT0TCW)

TT0TCW 寄存器被用来设置 16 位计数器的初始值。

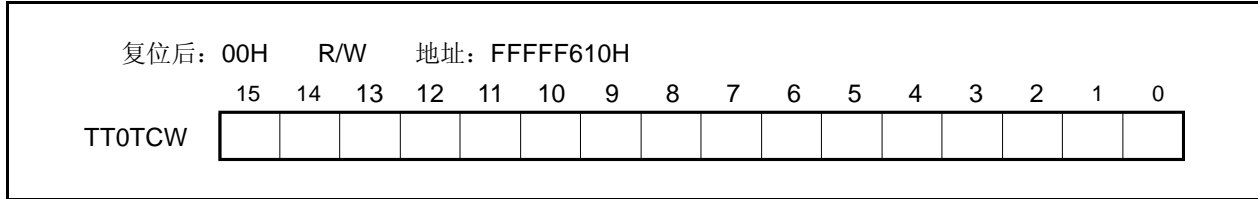
在编码器比较模式下，TT0TCW 寄存器有效。

此寄存器可以按 16 位宽度进行读取或写入。

当 TT0CTL0.TT0CE 位 = 0 时，可以重写 TT0TCW 寄存器。

当 TT0CE 位被置位 (1) 时，TT0TCW 寄存器的值被传送到 16 位计数器。

复位输入将该寄存器清为 0000H。

**(13) TMT0 计数器读取缓冲寄存器 (TT0CNT)**

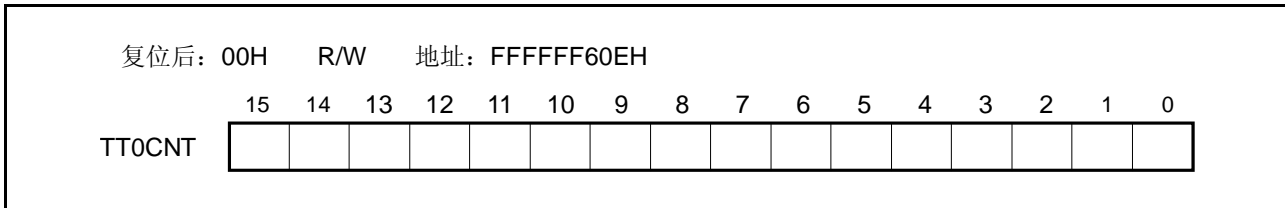
TT0CNT 寄存器是读出缓冲寄存器，可以读取 16 位计数器的计数值。

如果在 TT0CTL0.TT0CE 位=1 时读取该寄存器，可以读出 16 位定时器的计数值。

该寄存器为只读寄存器，按 16 位宽度读取。

当 TT0CTL2.TT0ECC 和 TT0CE 位=0 时，TT0CNT 寄存器的值被清为 0000H。如果此时读取 TT0CNT 寄存器，则不会读出 16 位计数器的值 (FFFFH)，读到的值为 0000H。当 TT0ECC 位 = 1 且 TT0CE 位 = 0 时，TT0CNT 寄存器不是被置为 0000H，而是读出上一次的值。

在复位后 TT0ECC 和 TT0CE 位被置为 0，TT0CNT 寄存器的值被清为 0000H。



(14) 噪声消除控制寄存器 (TTNFC)

可以为 TIT00, TIT01, TENC01, TECR0 和 EVTT00 引脚选择数字噪声消除。噪声消除的设置由 TTNFC 寄存器选择。

当选择选择数字噪声消除时, 数字采样的采样时钟可以选 f_{xx} , $f_{xx}/4$, $f_{xx}/8$, $f_{xx}/16$, $f_{xx}/32$ 和 $f_{xx}/64$ 。采样执行 3 次。

该寄存器可以按字节进行读取或写入。

复位输入将该寄存器清为 00H。

注意事项 在采样时钟被更改后, 需要 3 个采样时钟的时间来初始化数字噪声消除器。如果在采样时钟被更改后, 在 3 个采样时钟的时间完成前, 有 TIT00, TIT01, TENC01, TECR0 和 EVTT 的有效边沿输入, 于是, 可能会产生一个中断请求信号。因此, 当使用 TMT 的外部触发功能、外部事件功能、捕获触发功能和编码器功能时, 在经过 3 个采样时钟的时间后, 使能 TMT 的操作。

复位后: 00H R/W 地址: FFFFFFF26H

	7	6	5	4	3	2	1	0
TTNFC	TTNFEN	0	0	0	0	TTNFC2	TTNFC1	TTNFC0

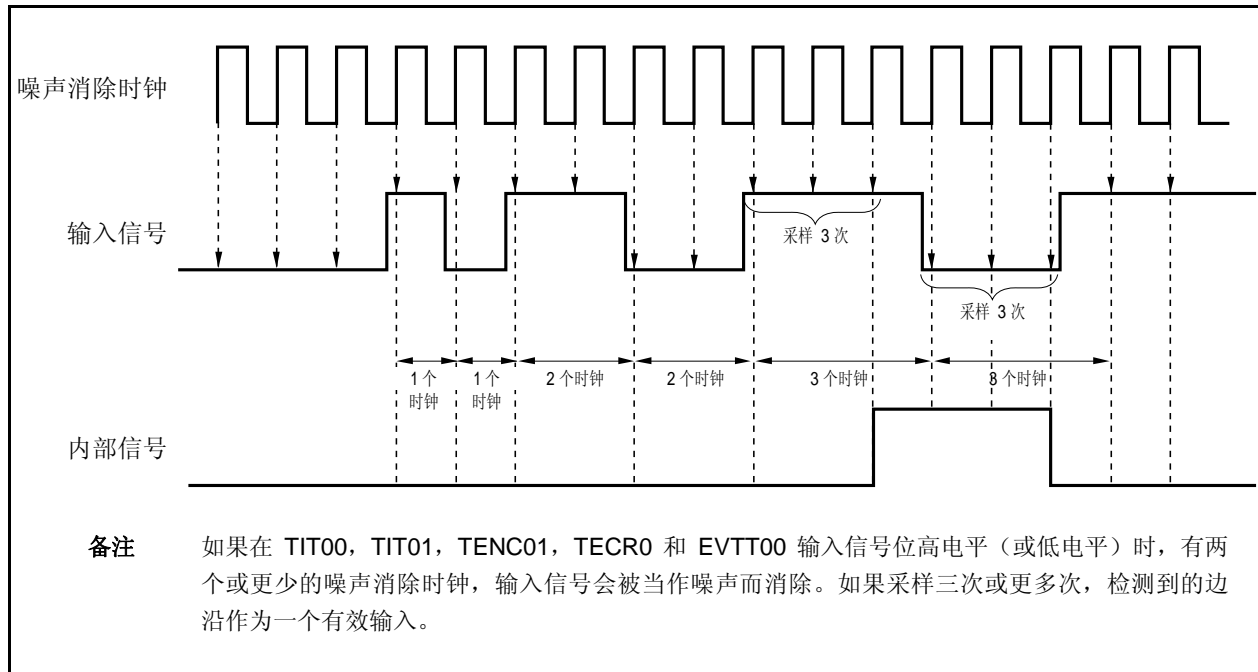
TTNFEN	数字噪声消除的设置
0	不执行数字噪声消除
1	执行数字噪声消除

TTNFC2	TTNFC1	TTNFC0	数字采样时钟
0	0	0	f_{xx}
0	0	1	$f_{xx}/4$
0	1	0	$f_{xx}/8$
0	1	1	$f_{xx}/16$
1	0	0	$f_{xx}/32$
1	0	1	$f_{xx}/64$
其它			禁止设置

- 备注**
1. 由于采样执行 3 次, 可以可靠消除的噪声宽度是 2 个采样时钟周期。
 2. 当噪声宽度小于 2 个采样时钟周期的情况时, 如果噪声和采样时钟同步输入, 则产生一个中断请求。

图 9-2 所示的是一个通过定时器 T 输入引脚数字滤波器执行噪声消除的时序示例。

图 9-2. 数字噪声消除时序的示例



9.5 定时器输出操作

下表列出了 TOT00 和 TOT01 引脚的操作和输出电平。

表 9-5. 各个模式下的定时器输出控制

操作模式	TOT01 引脚	TOT00 引脚
间隔定时器模式	方波输出	
外部事件计数器模式	无	
外部触发脉冲输出模式	外部触发脉冲输出	方波输出
单次触发脉冲输出模式	单次触发脉冲输出	
PWM (脉宽调制信号) 输出模式	PWM 输出	
自由运行定时器模式	方波输出(仅当使用比较功能时)	
脉冲宽度测量模式	无	
三角波 PWM 输出模式	三角波 PWM 输出	
编码器比较模式	无	

表 9-6. 定时器输出控制位控制下的 TOT00 和 TOT01 引脚的真值表

TT0IOC0.TT0OLn 位	TT0IOC0.TT0OEn 位	TT0CTL0.TT0CE 位	TOT0n 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，计数启动后为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平，计数启动后为低电平

备注 n = 0, 1

9.6 操作

每个通道的 TMT0 可以完成的功能不同，各个通道的功能列举如下。

表 9-7. 各个模式下的 TMT0 说明

操作	TT0CTL1.TT0EST 位 (软件触发位)	EVTTO 引脚 (外部触发输入)	捕获/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	仅作比较寄存器	随时写入
外部事件计数器模式	无效	无效	仅作比较寄存器	随时写入
外部触发脉冲输出模式	有效	有效	仅作比较寄存器	批量写入
单次触发脉冲输出模式	有效	有效	仅作比较寄存器	随时写入
PWM (脉宽调制信号) 输出模式	无效	无效	仅作比较寄存器	批量写入
自由运行定时器模式	无效	无效	可以切换	随时写入
脉冲宽度测量模式	无效	无效	仅作捕获寄存器	不适用
三角波 PWM 输出模式	无效	无效	仅作比较寄存器	批量写入
编码器比较模式	无效	无效	仅作比较寄存器	随时写入

(1) 基本计数器操作

本章节解释了 16 位计数器的基本操作。详情参见各种模式下的操作描述。

(a) 计数启动操作**• 编码器比较模式**

计数操作通过 TENC00 和 TENC01 来控制。

将 TT0TCW 寄存器的设置值传送给 16 位计数器并且计数操作启动时(当 TT0CTL2.TT0ECC 位 = 0, 在 TT0CTL0.TT0CE 位从 0 变为 1 的时刻, TT0TCW 寄存器设置值被传送到 16 位计数器), 执行 16 位计数器初始设置。

• 三角波 PWM 模式

16 位计数器从初始值 FFFFH 开始计数。

从 FFFFH 向上计数, 0000H, 0001H, 0002H, 0003H 以此类推。

在向上操作之后, 在 16 位计数器的值和 CCR0 缓冲寄存器匹配的时刻, 计数器向下计数。

• 上述之外的模式

16 位计数器从初始值 FFFFH 开始计数。

从 FFFFH 向上计数, 0000H, 0001H, 0002H, 0003H 以此类推。

(b) 清除操作

当它的值和比较寄存器匹配时, 当它的值被捕获时, 当检测到编码器清除信号的边沿时, 当检测到 TENC00, TENC01 和 TECR0 引脚的清除电平条件时, 16 位计数器被清除为 0000H。在计数器被启动后或当计数器溢出不是清除操作时, 立即发生从 FFFFH 到 0000H 的计数操作。因此不产生 INTTT0CC0 和 INTTT0CC1 中断信号。

(c) 溢出操作

在自由运行模式、脉宽测量模式和编码器比较模式下，当 16 位计数器从 FFFFH 向上计数到 0000H 时，发生溢出。如果在自由运行模式、脉宽测量模式下发生计数器溢出，TT0OPT0.TT0OVF 位被置为 1，且产生一个中断请求信号(INTTT0OV)。

如果在编码器比较模式下发生计数器溢出，TT0OPT1.TT0EOF 位被置为 1，且产生一个中断请求信号(INTTT0OV)。

注意在下列条件时不产生 INTTT0OV 信号。

- 在计数操作刚刚启动后
- 如果计数器的值和比较值 FFFFH 匹配并被清除时
- 在脉宽测量模式下，当捕获到 FFFFH 并清除为 0000H 时

注意事项 在溢出中断请求信号 (INTTT0OV) 产生之后，请确保检查溢出标志(TT0OVF, TT0EOF bits) 被置为 1。

(d) 计数值保持操作

在编码器比较模式下，由 TT0CTL2.TT0ECC 位来控制 16 位计数器的值的保存。当 TT0ECC 位 = 0 且 TT0CTL0.TT0CE 位 = 0 时，16 位计数器的值被重置为 FFFFH。当 TT0CE 位下次被设置为 1 时，TT0TCW 寄存器的设置值被传送到 16 位计数器，并且执行计数操作。

如果 TT0ECC 位 = 1 且 TT0CE 位 = 0，16 位计数器的值被保存。当 TT0CE 位下次被设置为 1 时，计数器从保存值恢复计数操作。

(e) 在计数操作期间读取计数器的操作

在计数操作期间，可以通过 TT0CNT 寄存器来读取 TMT0 的 16 位计数器的值。

当 TT0CTL0.TT0CE 位 = 1，可以通过 TT0CNT 寄存器来读取 TMT0 的 16 位计数器的值。但是，如果在 TT0CTL2.TT0ECC 位 = 0 且 TT0CE 位 = 0 时读取 TT0CNT 寄存器，它的值是 0000H。当 TT0ECC 位 = 1 且 TT0CE 位 = 0 时读取 TT0CNT 寄存器，读到的值是保存值。

(f) 下溢操作

在编码器比较模式下，当 16 位计数器从 0000H 改变为 FFFFH 时，发生一个 16 位计数器的下溢。当下溢发生时，TT0OPT1.TT0EUF 位被置为 1，且产生一个中断请求信号 (INTTT0OV)。

(g) 中断操作

TMT0 产生下列四种类型的中断请求信号

- INTTT0CC0 中断: 该信号可以作为 CCR0 缓冲寄存器的匹配中断请求信号, 也可以作为 TT0CCR0 寄存器的捕获中断请求信号。
- INTTT0CC1 中断: 该信号可以作为 CCR1 缓冲寄存器的匹配中断请求信号, 也可以作为 TT0CCR1 寄存器的捕获中断请求信号。
- INTTT0OV 中断: 该信号作为溢出中断请求信号。
- INTTT0EC 中断: 该信号作为检测到编码器清除输入(TECR0 引脚)有效边沿的中断请求信号。

(2) 任意时刻写入和批量写入

在定时器操作期间(TT0CTL0.TT0CE 位 = 1), 可以重写 TMT0 的 TT0CCR0 和 TT0CCR1 寄存器, 但是 CCR0 和 CCR1 缓冲寄存器的写入方法(任意时刻写入, 批量写入)取决于不同的模式。

(a) 任意时刻写入

该模式下, 在定时器操作期间的任意时刻, 数据可以从 TT0CCR0 和 TT0CCR1 寄存器传送到 CCR0 和 CCR1 缓冲寄存器。

图 9-3. 任意时刻写入的基本操作流程图示例

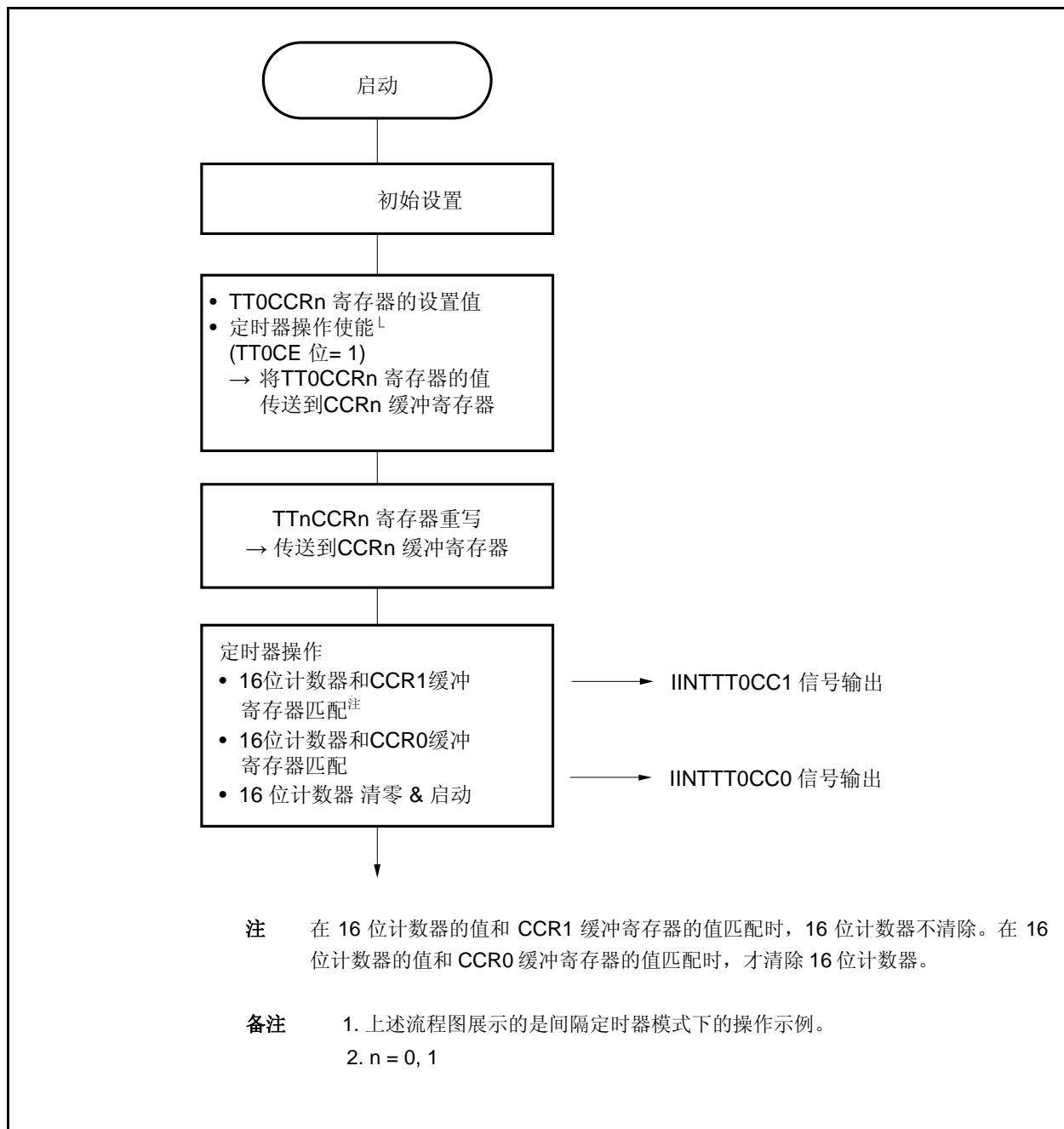
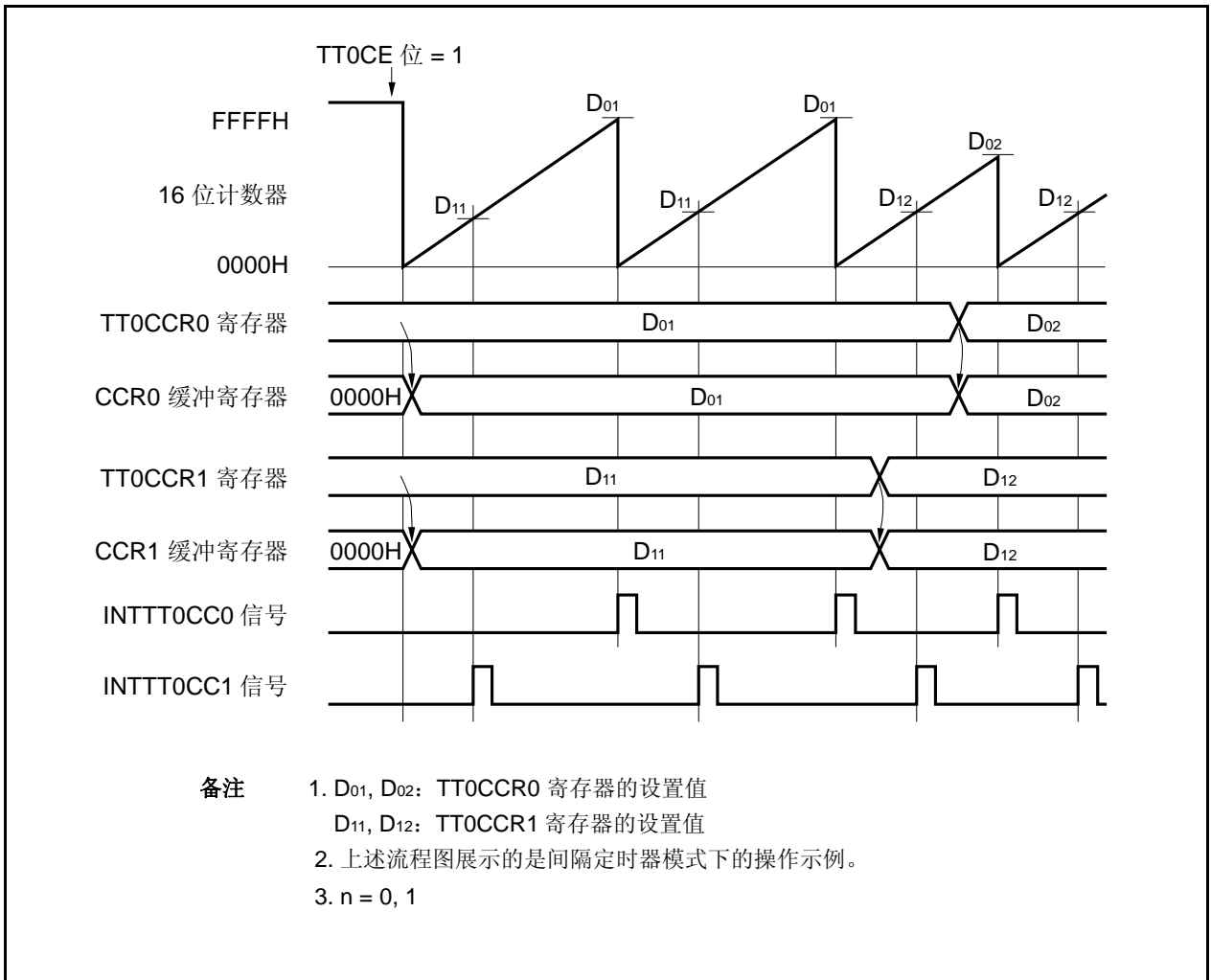


图 9-4. 任意时刻写入的时序示例



(b) 批量写入

该模式下，在定时器操作期间，数据一次性从 TT0CCR0 和 TT0CCR1 寄存器传送到 CCR0 和 CCR1 缓冲寄存器。在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配时，传送该数据。写入 TT0CCR1 寄存器就可以使能传送。是否写入 TT0CCR1 寄存器可以控制使能或禁止下一次传送时序。

为了让重写 TT0CCR0 和 TT0CCR1 寄存器时的设置值成为 16 位计数器比较值（换句话说，为了该值被传送到 CCR0 和 CCR1 缓冲寄存器），必需重写 TT0CCR0 寄存器，然后在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配之前写入 TT0CCR1 寄存器。因此，在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配时，TT0CCR0 和 TT0CCR1 寄存器的值被传送到 CCR0 和 CCR1 缓冲寄存器。所以，即使只想重写 TT0CCR0 寄存器的值，也需要将相同的值（和 TT0CCR1 寄存器的当前值相同）写入 TT0CCR1 寄存器。

图 9-5. 批量写入的基本操作流程图示例

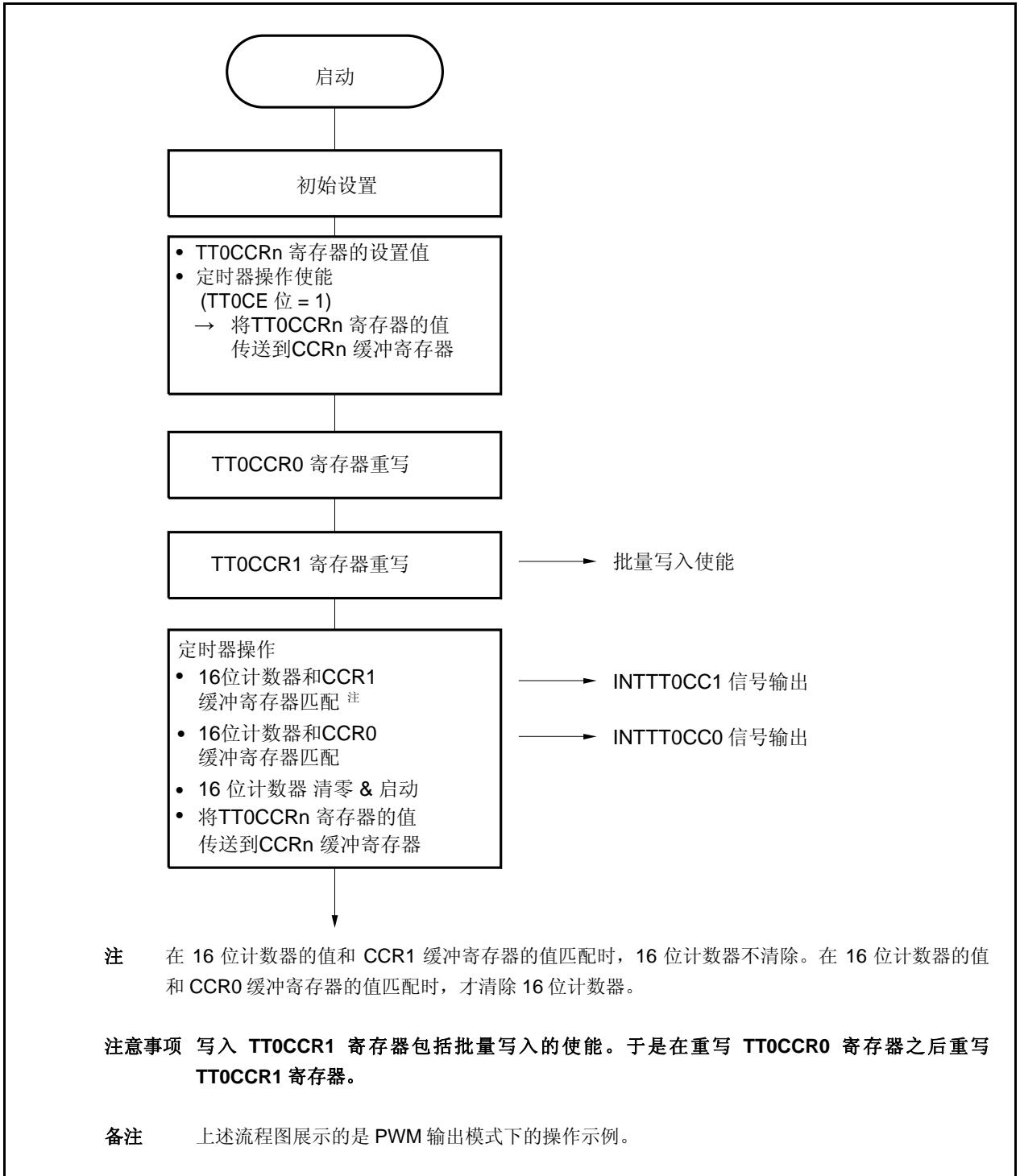
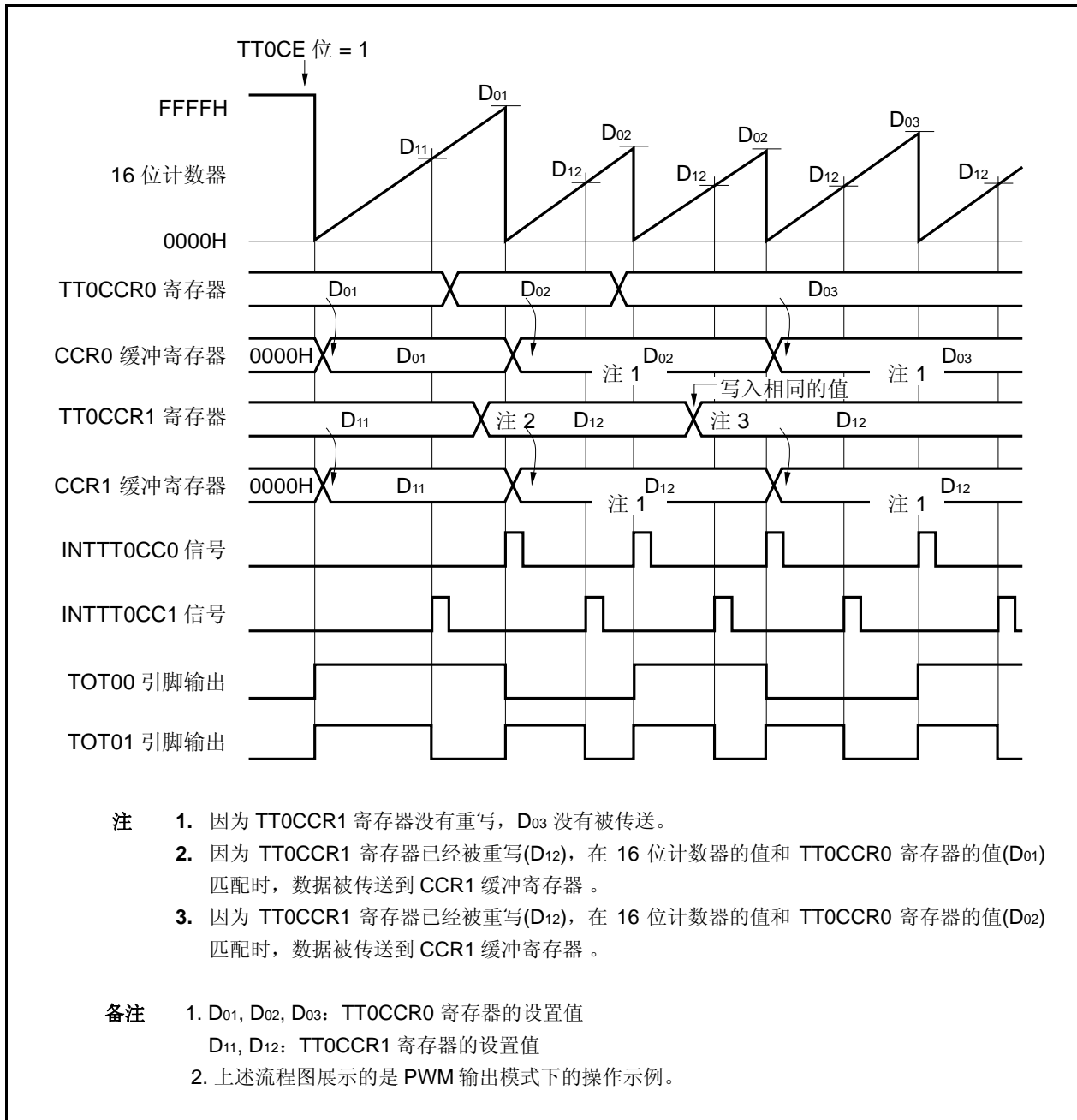


图 9-6. 批量写入的时序示例 (TT0 的间隔定时器模式)



9.6.1 间隔定时器模式 (TT0MD3 至 TT0MD0 位=0000)

在间隔定时器模式下，如果 TT0CTL0.TT0CE 位被置为 1，则会以 TT0CCR0 寄存器指定的时间间隔生成中断请求信号 (INTTT0CC0)，从 TOT00 引脚输出占空比为 50% 的方波，指定的时间间隔等于方波的半周期。

在间隔定时器模式下，不使用 TT0CCR1 寄存器。但是 TT0CCR1 寄存器的设置值被传送到 CCR1 缓冲寄存器，当 16 位计数器的计数值和 CCR1 缓冲寄存器匹配时，产生一个比较匹配中断请求信号 (INTTT0CC1)。并且，从 TOT01 引脚输出占空比 50% 的方波，在产生 INTTT0CC1 信号时电平反转。

即使在定时器操作期间，也可以重写 TT0CCR0 和 TT0CCR1 寄存器的值。

图 9-7. 间隔定时器的配置

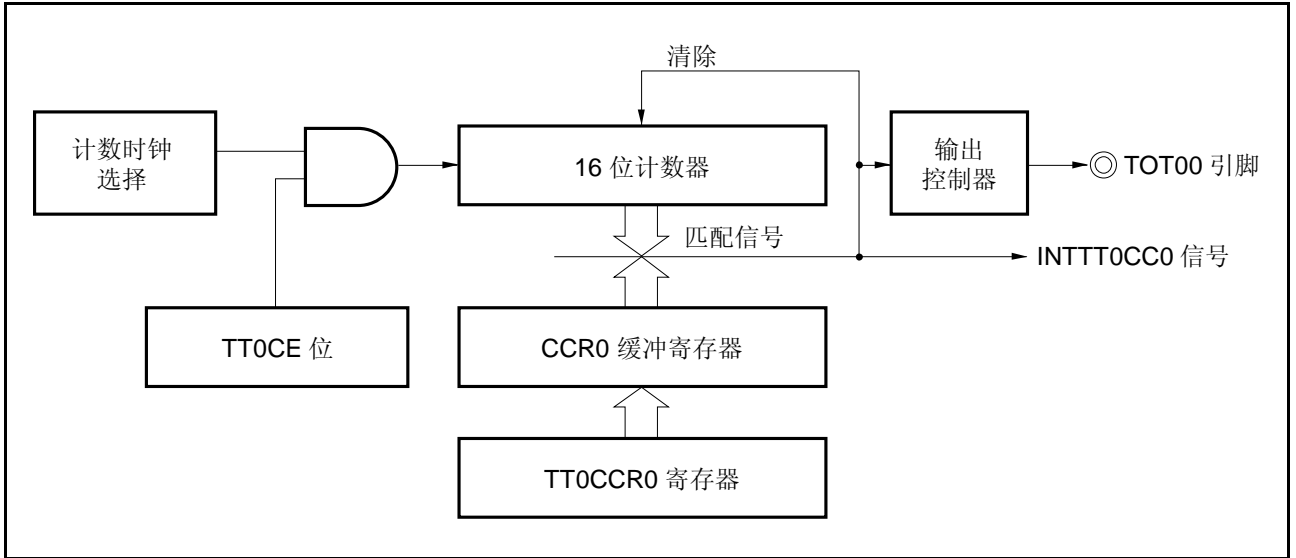
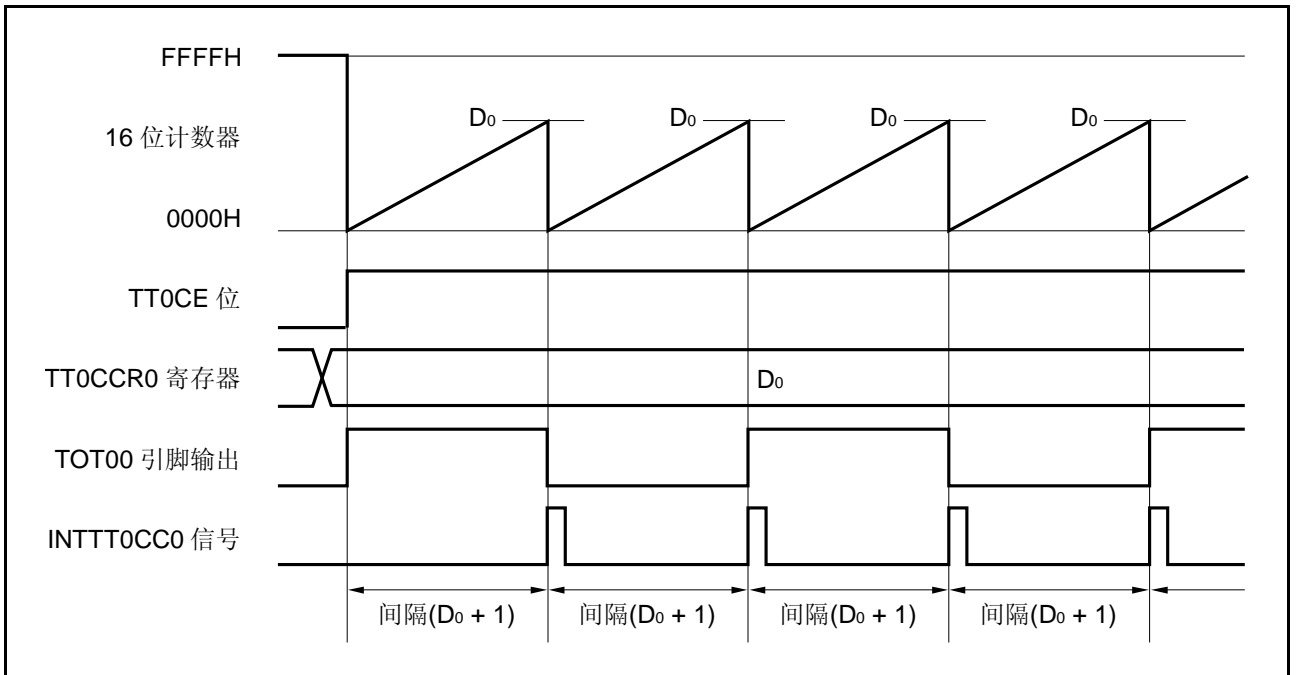


图 9-8. 间隔定时器模式的基本时序



当 TT0CE 位被置为 1 时，16 位计数器的值与计数时钟同步，从 FFFFH 清零为 0000H，计数器开始计数。此时，TOT00 引脚的输出反相。此外，TT0CCR0 寄存器的设定值传送给 CCR0 缓冲寄存器。

当 16 位计数器的计数值和 CCR0 缓冲寄存器的值相匹配时，则 16 位计数器被清为 0000H，TOT00 引脚输出反相，并产生一个比较匹配中断请求信号（INTTT0CC0）。

时间间隔可用下述表达式计算。

$$\text{时间间隔} = (\text{TT0CCR0 寄存器的设定值} + 1) \times \text{计数时钟周期}$$

图 9-9. 间隔定时器模式操作时的寄存器设置 (1/2)

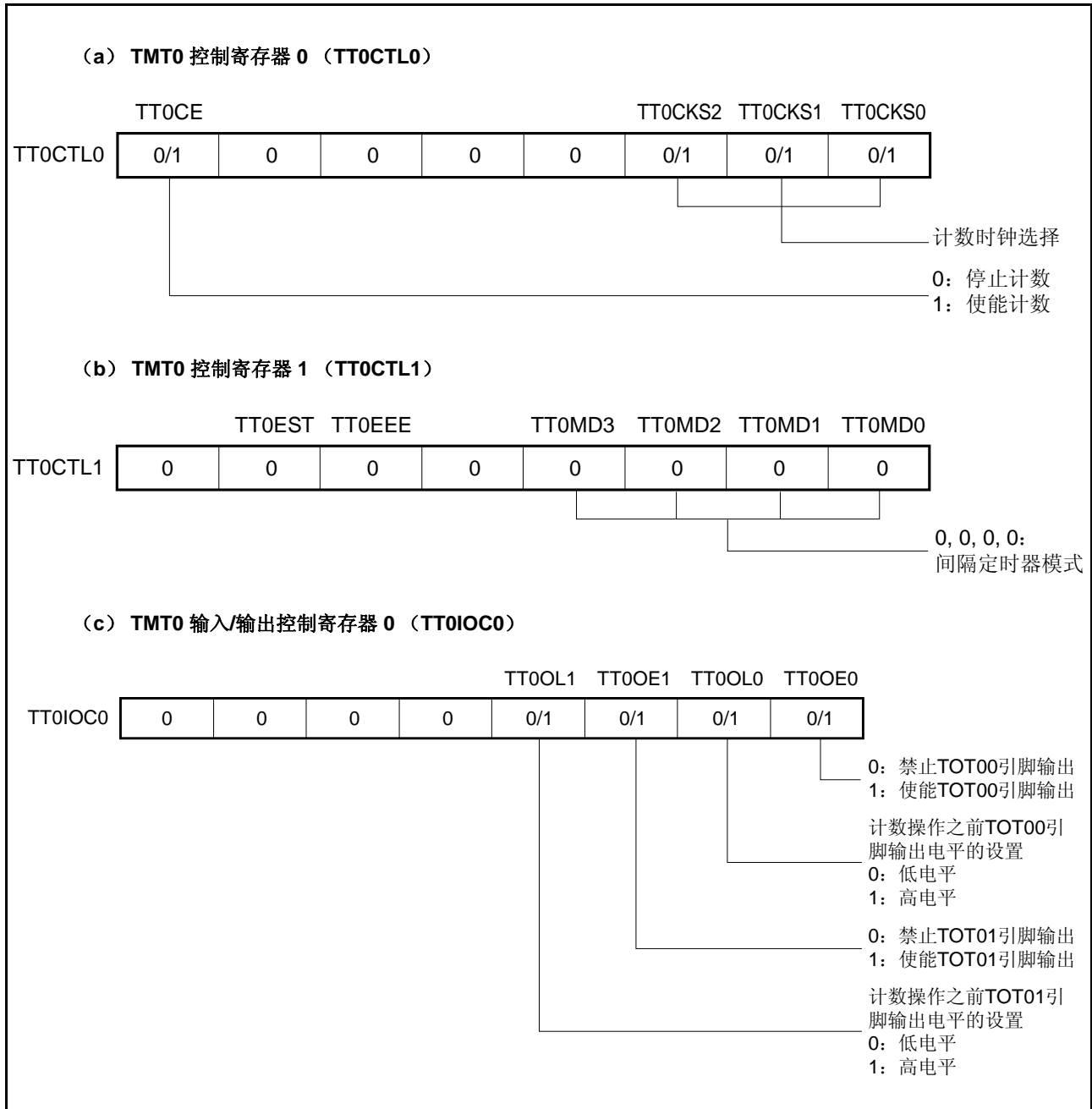


图 9-4. 间隔定时器模式操作时的寄存器设置 (2/2)

(d) TMT0 计数器读取缓冲寄存器 (TT0CNT)

通过读取 TT0CNT 寄存器，可以读取 16 计数器的当前值。

(e) TMT0 捕获/比较寄存器 0 (TT0CCR0)

如果 TT0CCR0 寄存器的值被设置为 D_0 ，则时间间隔的计算如下：

$$\text{时间间隔} = (D_0 + 1) \times \text{计数时钟周期}$$

(f) TMT0 捕获/比较寄存器 1 至 3 (TT0CCR1)

在间隔定时器模式下，不使用 TT0CCR1 寄存器。但是，TT0CCR1 寄存器的设定值仍然会被传送给 CCR1 缓冲寄存器。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会产生比较匹配中断请求信号 (INTTT0CC1)。

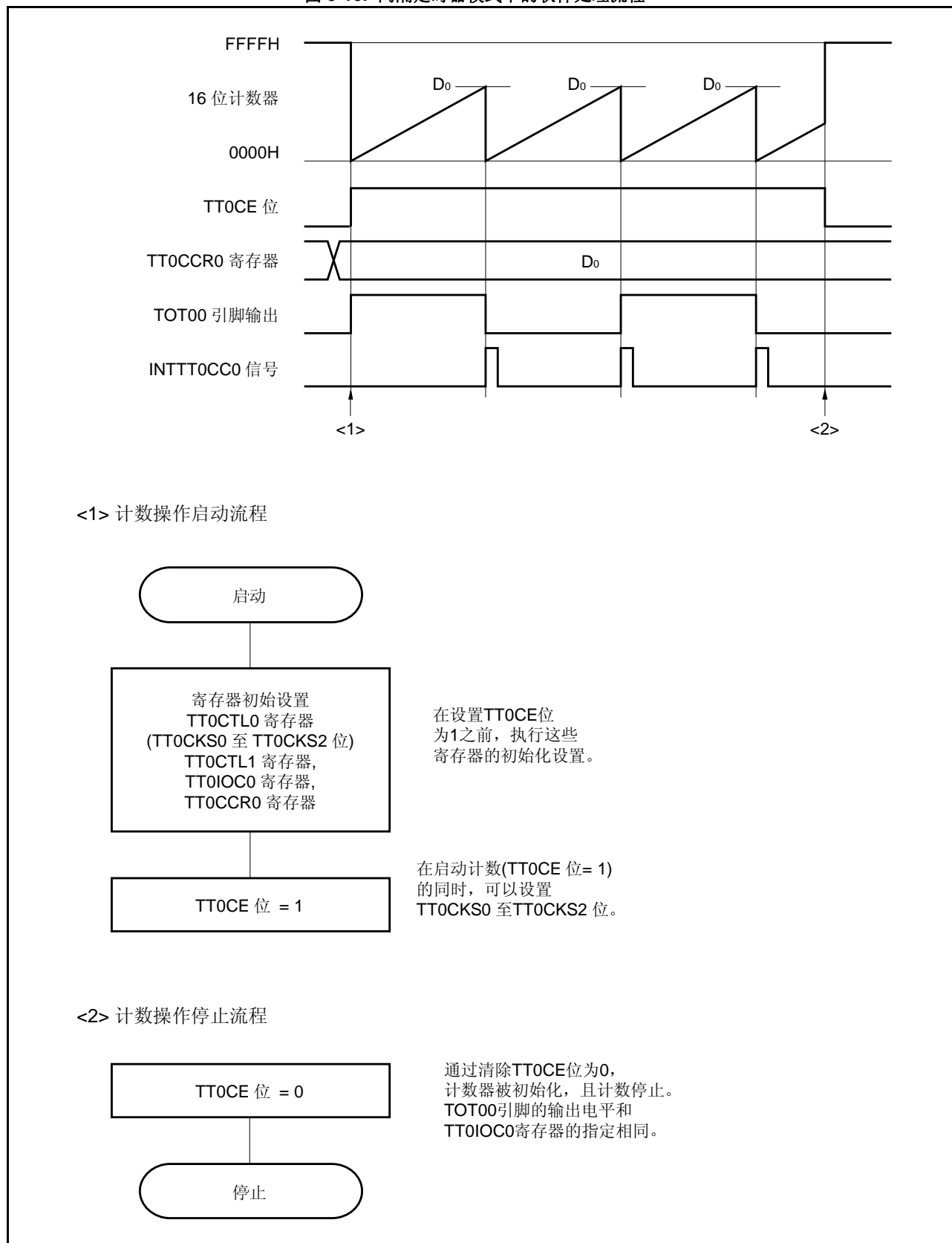
将该寄存器的值设置为和 TT0CCR0 寄存器的设置值相等，可以从 TOT01 引脚输出一个占空比 50% 的方波。

不使用 TT0CCR1 寄存器时，推荐将其值设置为 FFFFH。并且通过相应的中断屏蔽标志 (TT0CCIC1.TT0CCMK1) 屏蔽该寄存器。

备注 在间隔定时器模式下，不使用 TMT0 控制寄存器 2 (TT0CTL2)、TMT0 输入/输出控制寄存器 1 (TT0IOC1)、TMT0 输入/输出控制寄存器 2 (TT0IOC2)、TMT0 选项寄存器 0 (TT0OPT0)、TMT0 选项寄存器 1 (TT0OPT1) 和 TMT0 计数写入寄存器 (TT0TCW)。

(1) 间隔定时器模式操作流程

图 9-10. 间隔定时器模式下的软件处理流程

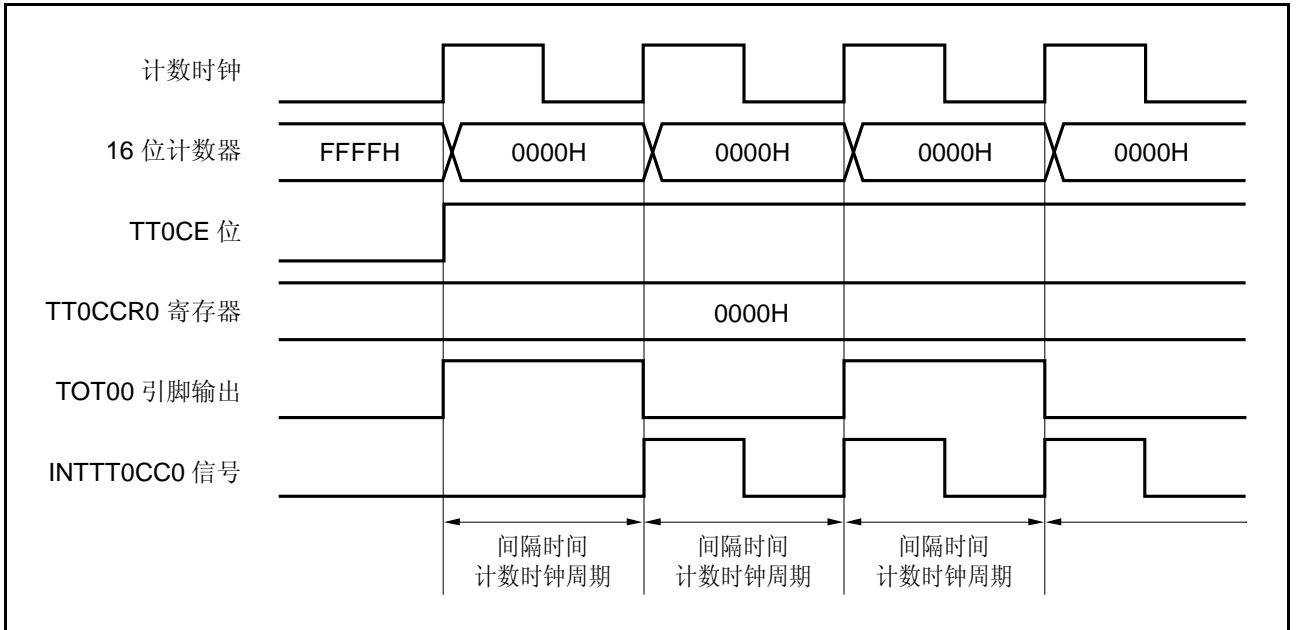


(2) 间隔定时器模式操作时序

(a) TT0CCR0 寄存器清为 0000H 时的操作

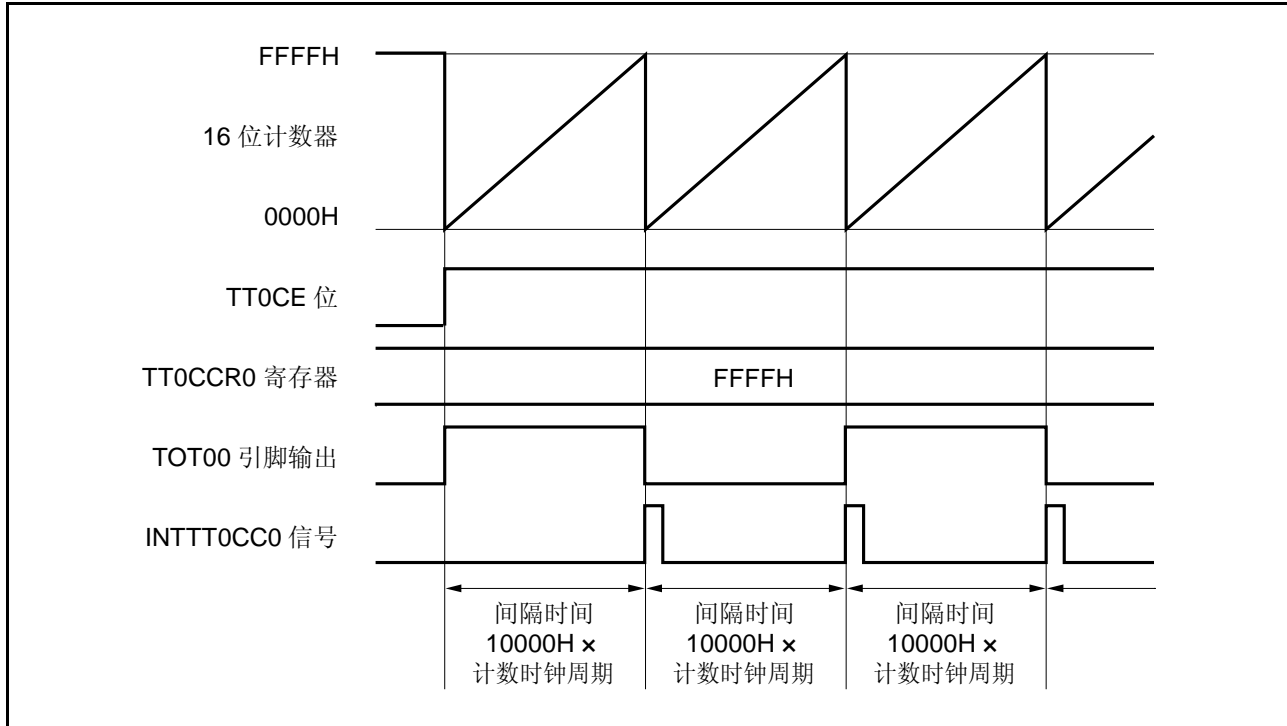
如果 TT0CCR0 寄存器被清为 0000H，在每个计数时钟都会产生 INTTT0CC0 信号，而且，TOT00 引脚的输出反相。

16 位计数器的值总是 0000H。



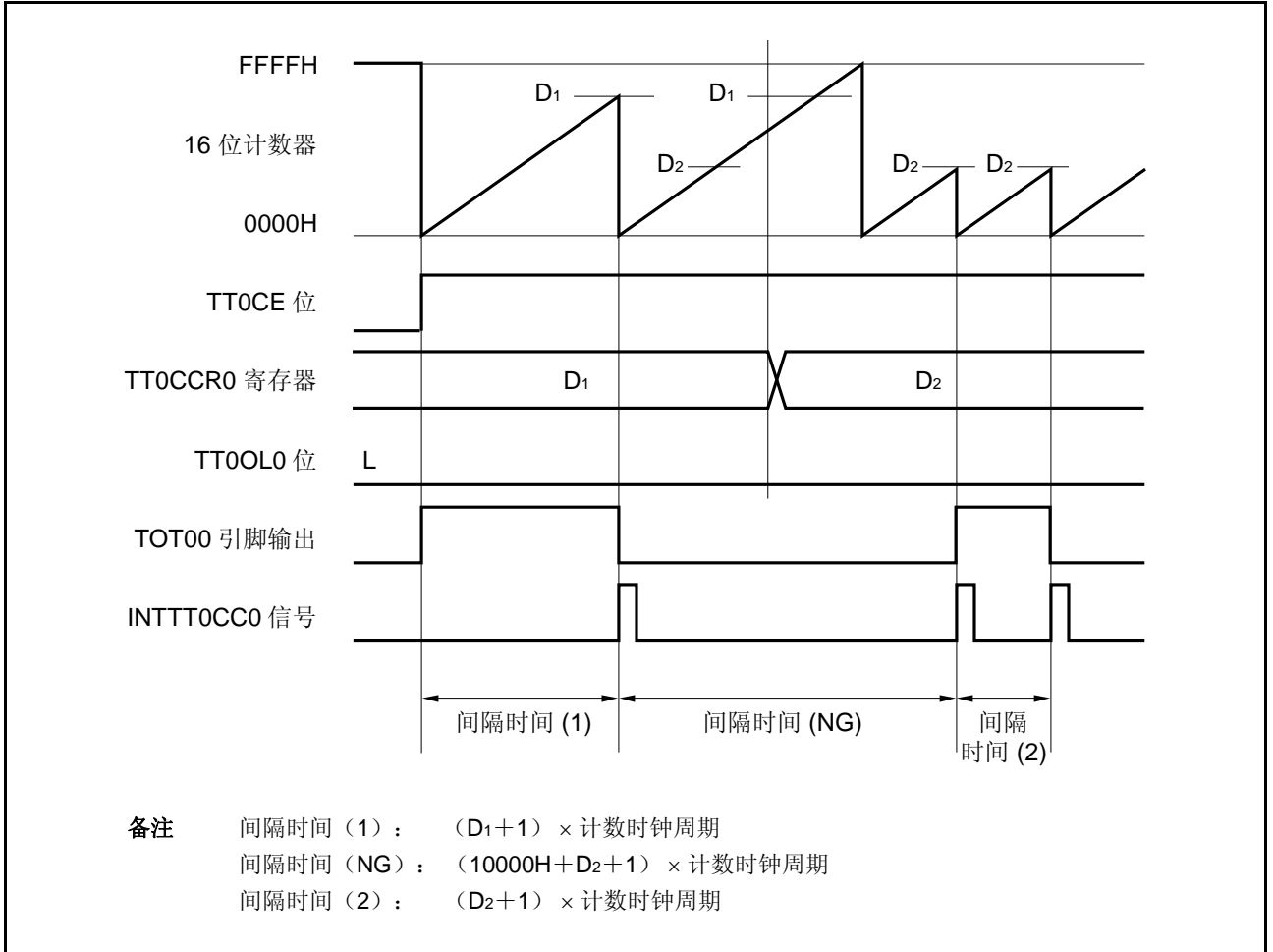
(b) TT0CCR0 寄存器被设置为 FFFFH 时的操作

如果 TT0CCR0 寄存器被置为 FFFFH，那么 16 位计数器可以计数到 FFFFH。该计数器在下一个计数时刻被同步清零。产生 INTTT0CC0 信号，并且 TOT00 引脚输出反相。此时，不产生溢出中断请求信号 (INTTT0OV)，而且，溢出标志 (TT0OPT0.TT0OVF 位) 也不会置 1。



(c) 重新写入 TT0CCR0 寄存器的注意事项

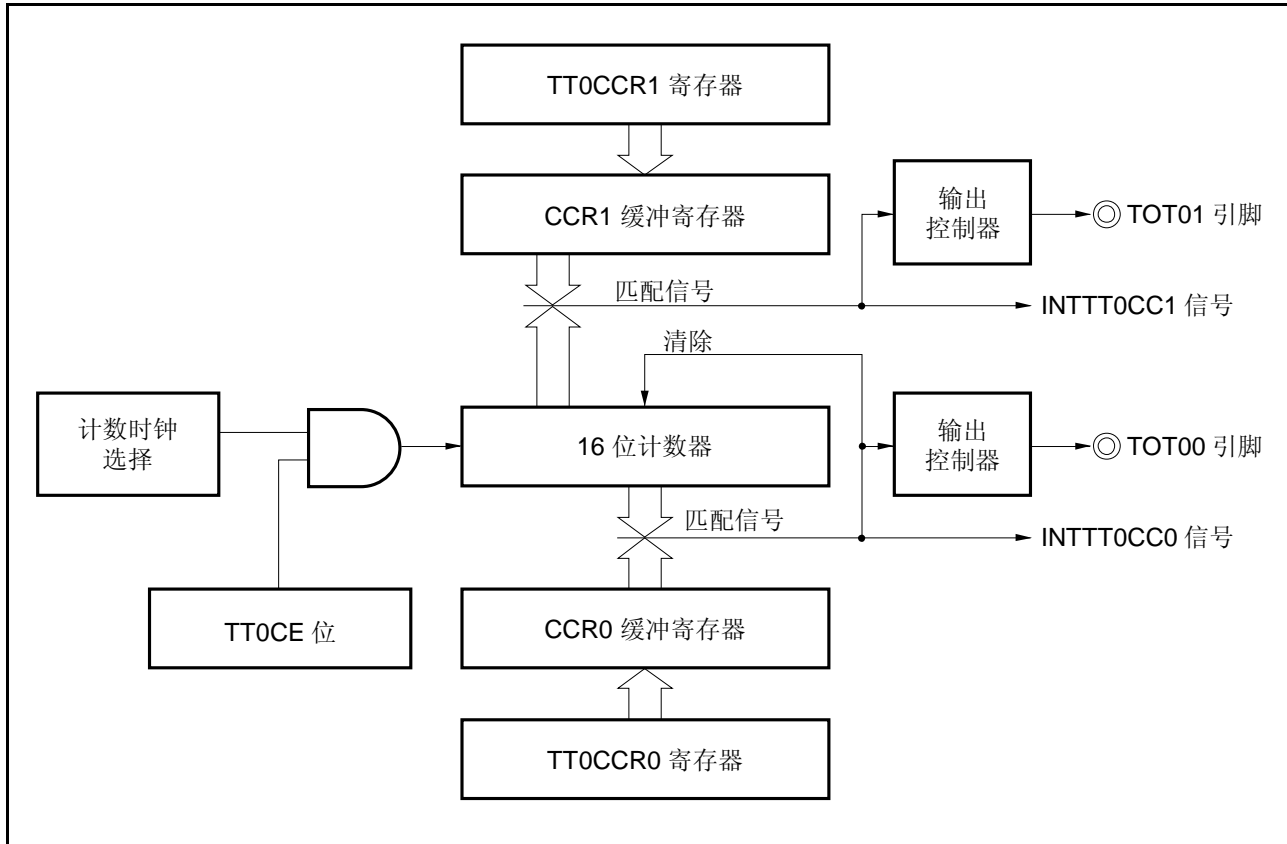
如果在计数器工作期间，将寄存器 TT0CCR0 的值重写为比当前值小的值，16 位计数器可能会溢出。当可能发生溢出时，应停止计数，然后改变其设定值。



当 16 位计数器计数值大于 D_2 而小于 D_1 时，如果将 TT0CCR0 寄存器的值从 D_1 变为 D_2 ，则在重写 TT0CCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。结果，用来比较的 16 位计数器的值是 D_2 。但是，由于计数值已经超过了 D_2 ，所以 16 位计数器就会一直向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值和 D_2 相符时才产生 INTTT0CC0 信号，且 TOT00 引脚的输出反相。因此，可能不会在原先预计的间隔时间“ $(D_1 + 1) \times$ 计数时钟周期”或“ $(D_2 + 1) \times$ 计数时钟周期”时产生 INTTT0CC0 信号，而会以“ $(10000H + D_2 + 1) \times$ 计数时钟周期”的间隔产生中断请求信号。

(d) TT0CCR1 寄存器的操作

图 9-11. TT0CCR1 寄存器的配置



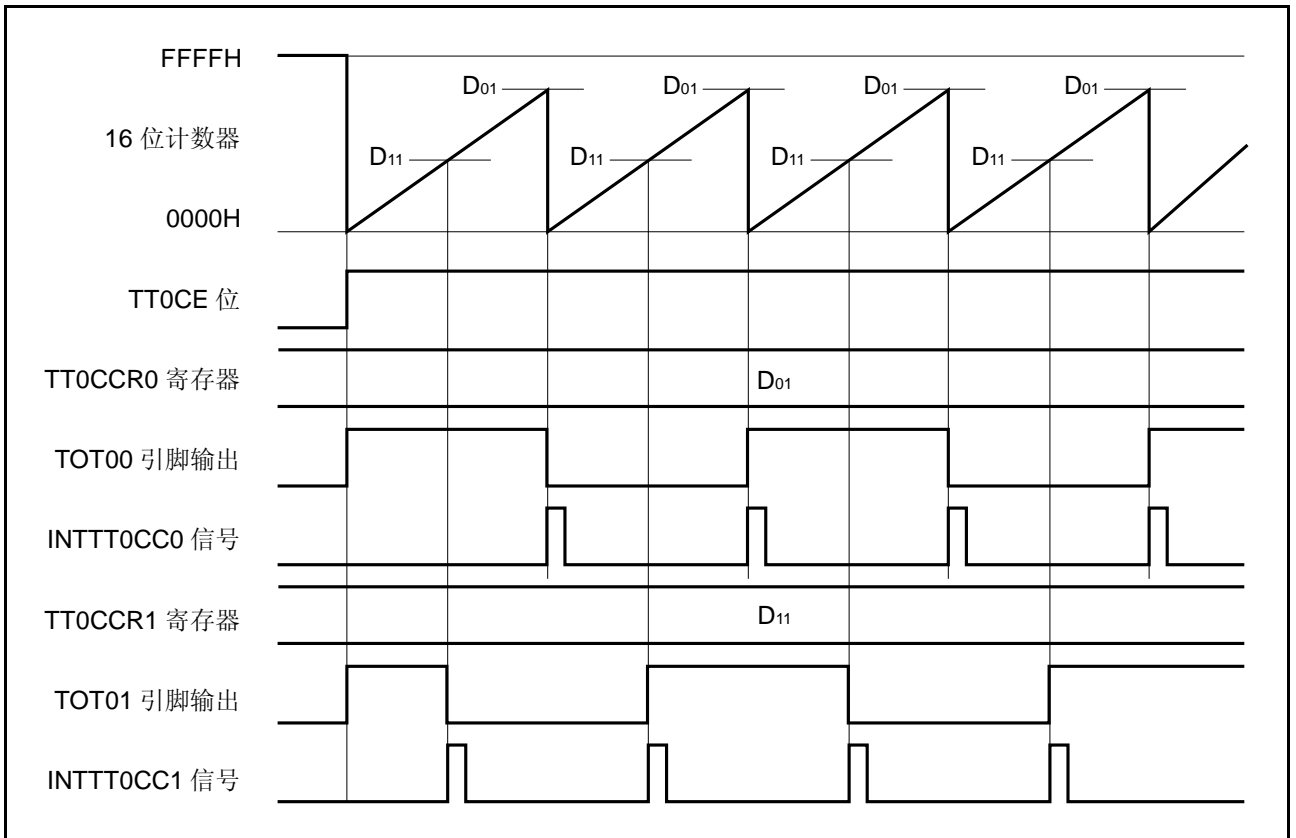
当 TT0CCR1 的值和 TT0CCR0 的设置值相同时，INTTT0CC0 信号与 INTTT0CC1 信号同时产生，且 TOT01 引脚输出反转。换句话说，TOT01 引脚会输出一个占空比 50% 的方波。

下面展示了当 TT0CCR1 寄存器的设置值和 TT0CCR0 寄存器的值不等的操作。

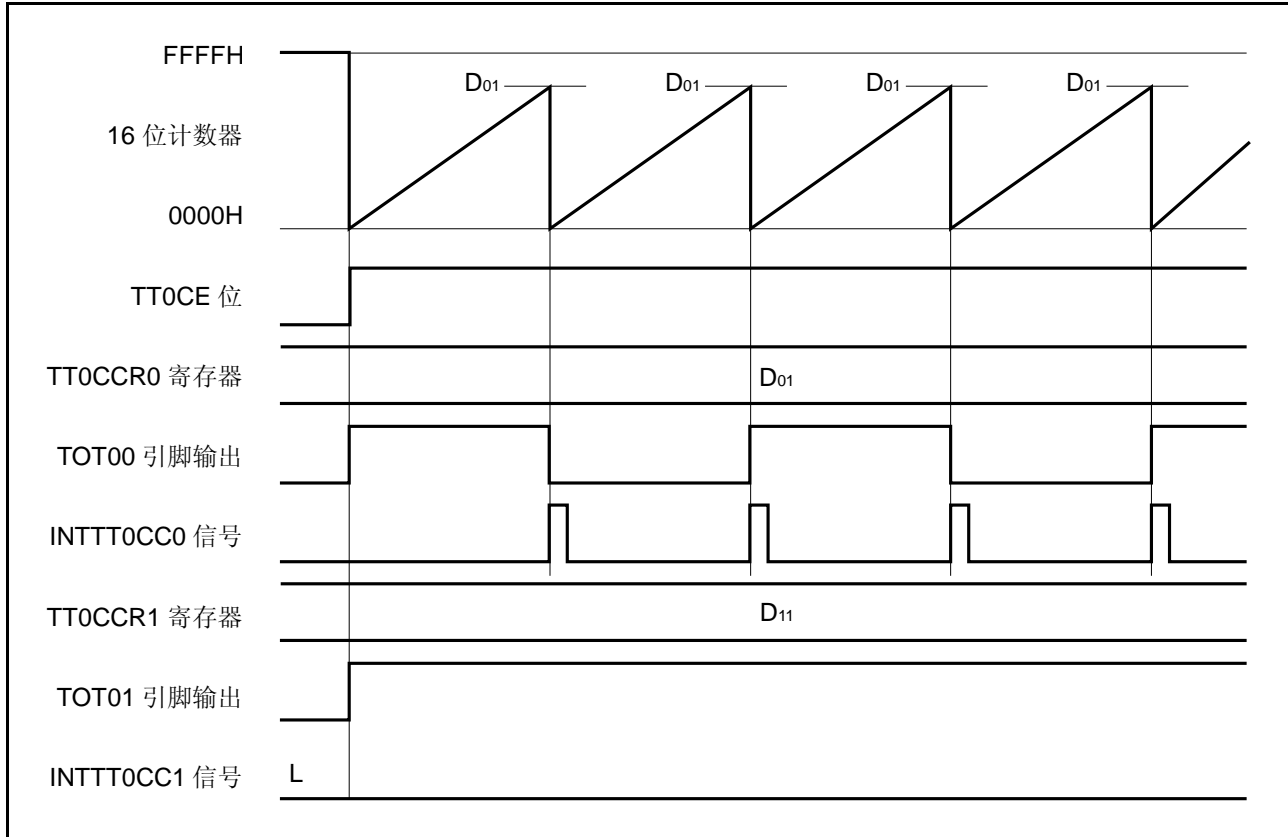
如果 TT0CCR1 寄存器的设定值小于 TT0CCR0 寄存器的设定值，则每个计数周期产生一次 INTTT0CC1 中断请求信号。同时，TOT01 引脚的输出反相。

在输出一个窄宽度脉冲后，TOT01 引脚输出一个占空比 50% 的方波。

图 9-12. 当 $D_{01} \geq D_{11}$ 时的时序图



如果 TT0CCR1 寄存器的设定值大于 TT0CCR0 寄存器的设定值，则 16 位计数器的计数值不会和 TT0CCR1 寄存器的值发生匹配。因此也就不会产生 INTTT0CC1 信号，TOT01 引脚的输出也不会改变。不使用 TT0CCR1 寄存器时，推荐将其值设置为 FFFFH。

图 9-13. 当 $D_{01} < D_{11}$ 时的时序图

9.6.2 外部事件计数模式 (TT0MD3 至 TT0MD0 位=0001)

在外部事件计数器模式下，当 TT0CTL0.TT0CE 位被置为 1 时，对外部事件计数输入(EVTT0)的有效边沿进行计数，每次有效边沿的计数数量达到 TT0CCR0 寄存器指定的数目时，产生一个中断请求信号 (INTTT0CC0)。不能使用 TOT00 和 TOT01 引脚。

在外部事件计数模式下，不使用 TT0CCR1 寄存器。

图 9-14. 外部事件计数模式的配置

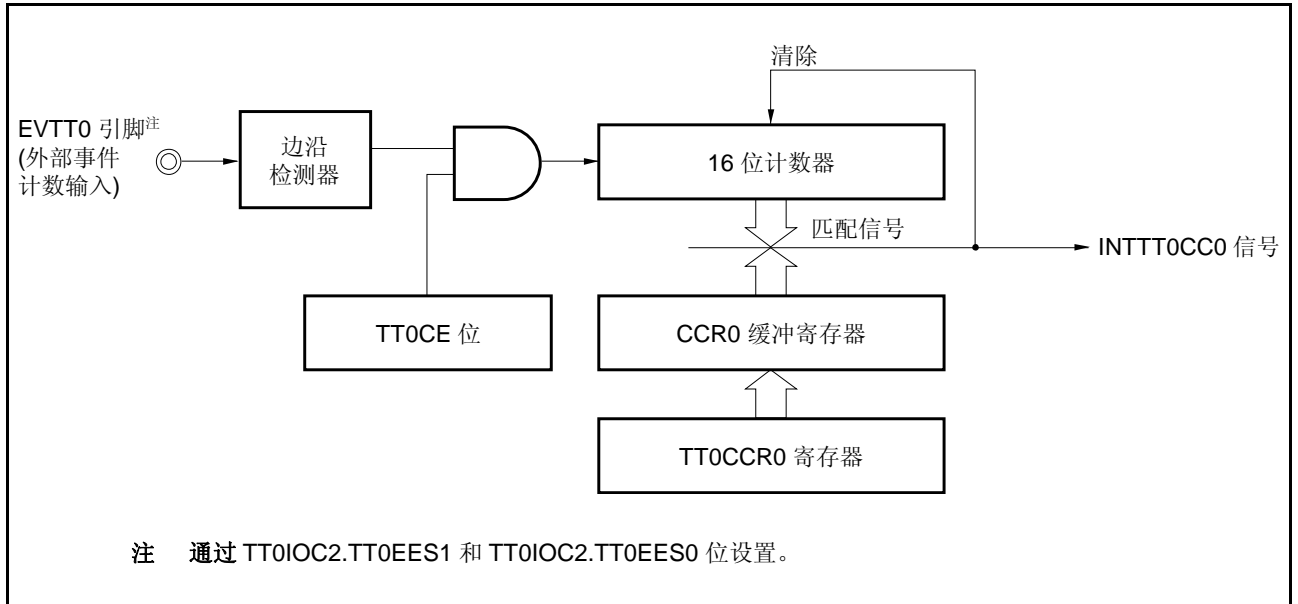
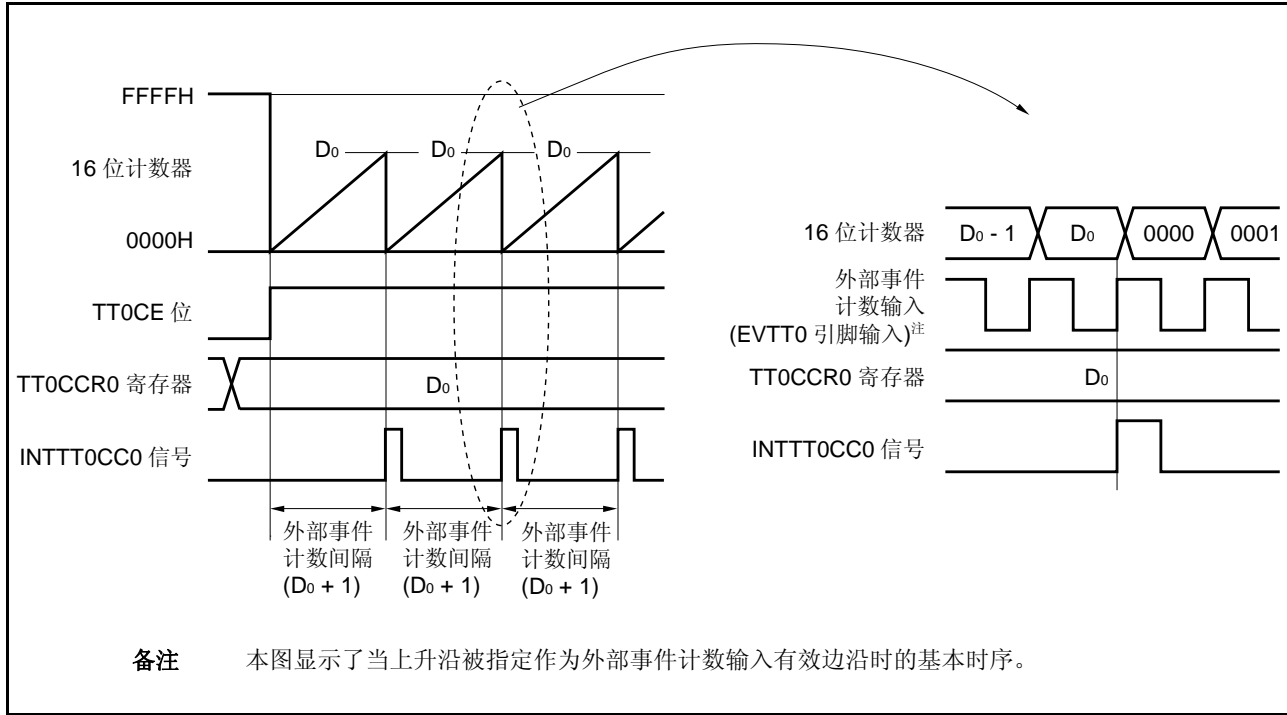


图 9-15. 外部事件计数模式的基本时序



当 TT0CE 位被置为 1 时，16 位计数器的值从 FFFFH 被清为 0000H。计数器对检测到的每一次外部事件计数输入的有效边沿进行计数。此外，TT0CCR0 寄存器的设定值被传送给 CCR0 缓冲寄存器。

当 16 位计数器的计数值和 CCR0 缓冲寄存器的数值相匹配时，16 位计数器清零为 0000H，同时生成一个比较匹配中断请求信号 (INTTT0CC0)。

每当外部事件计数输入的有效边沿数量检测达到 (TT0CCR0 寄存器的设定值+1) 时，就会生成 INTTT0CC0 (比较匹配中断请求) 信号。

图 9-16. 外部事件计数模式下的寄存器操作设置 (1/2)

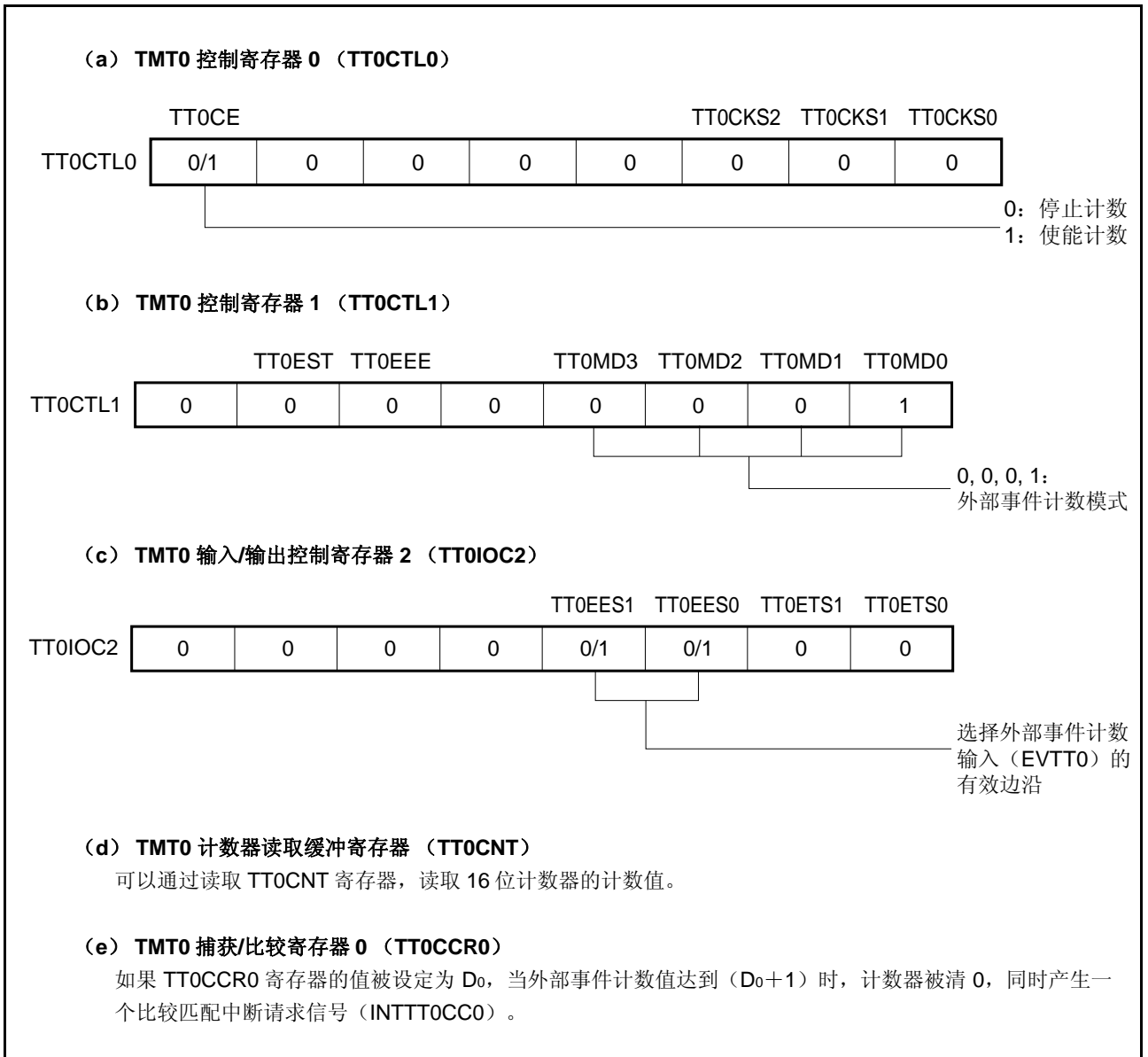


图 9-11. 外部事件计数模式下的寄存器操作设置 (2/2)

(f) TMT0 捕获/比较寄存器 1 (TT0CCR1)

在外部事件计数模式下，不使用 TT0CCR1 寄存器。但是，TT0CCR1 寄存器的设定值仍然被传送给 CCR1 缓冲寄存器。当 16 位计数器的计数值和 CCR1 缓冲寄存器的值匹配时，会产生一个比较匹配中断请求信号 (INTTT0CC1)。

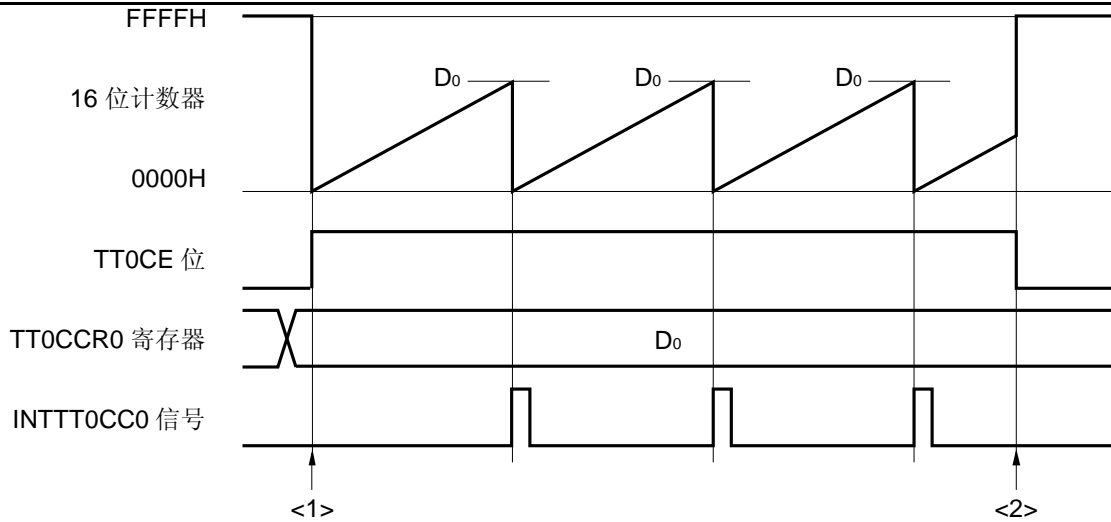
不使用 TT0CCR1 寄存器时，推荐将其值设置为 FFFFH。并且通过中断屏蔽标志 (TT0CCIC1.TT0CCMK1) 来屏蔽该寄存器。

备注 在外部事件计数模式下，不使用 TMT0 控制寄存器 2 (TT0CTL2)、TMT0 输入/输出控制寄存器 0 (TT0IOC0)、TMT0 输入/输出控制寄存器 1 (TT0IOC1)、TMT0 输入/输出控制寄存器 3 (TT0IOC3)、TMT0 选项寄存器 0 (TT0OPT0)、TMT0 选项寄存器 1 (TT0OPT1) 和 TMT0 计数写入寄存器 (TT0TCW)。

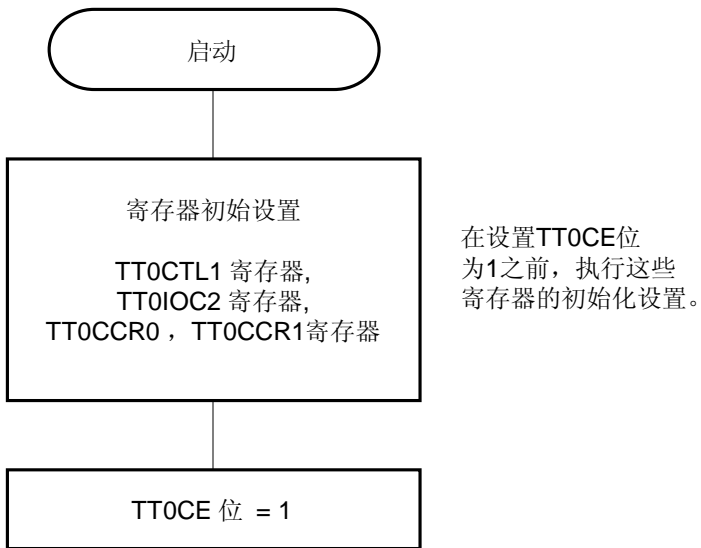
(1) 外部事件计数模式的操作流程

图 9-17. 外部事件计数模式下的软件处理流程

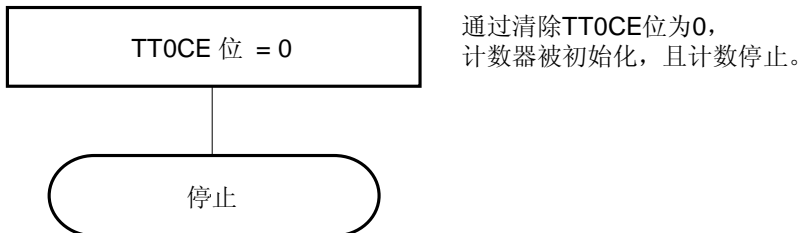




<1> 计数操作启动流程



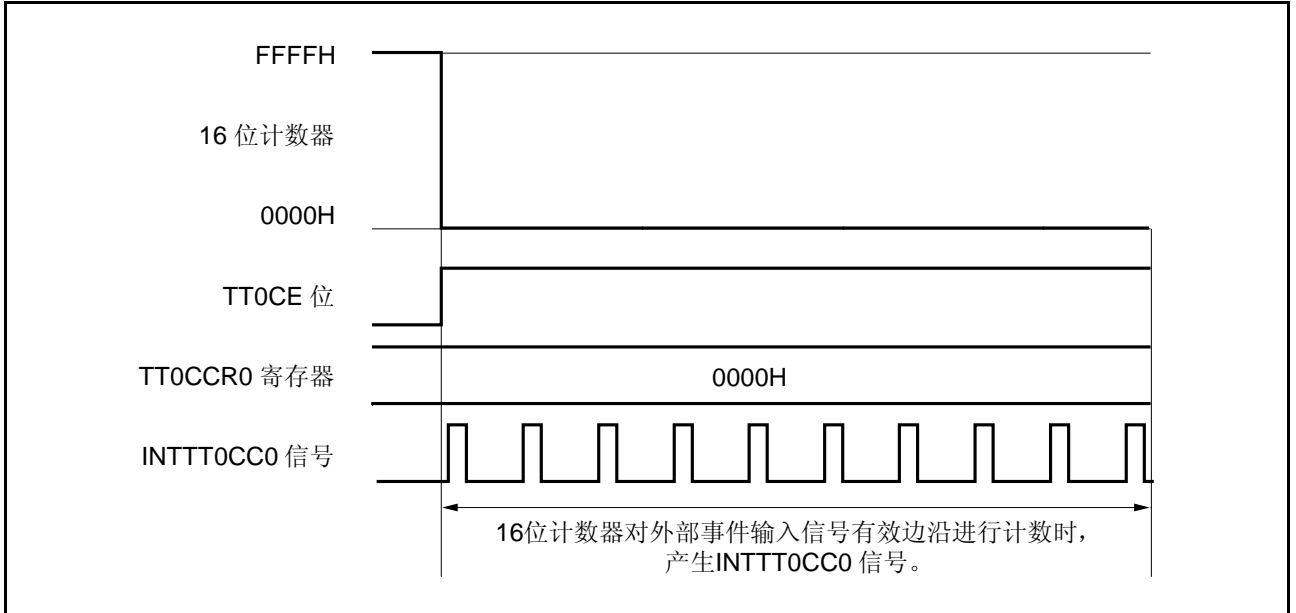
<2> 计数操作停止流程



(2) 外部事件计数模式的操作时序

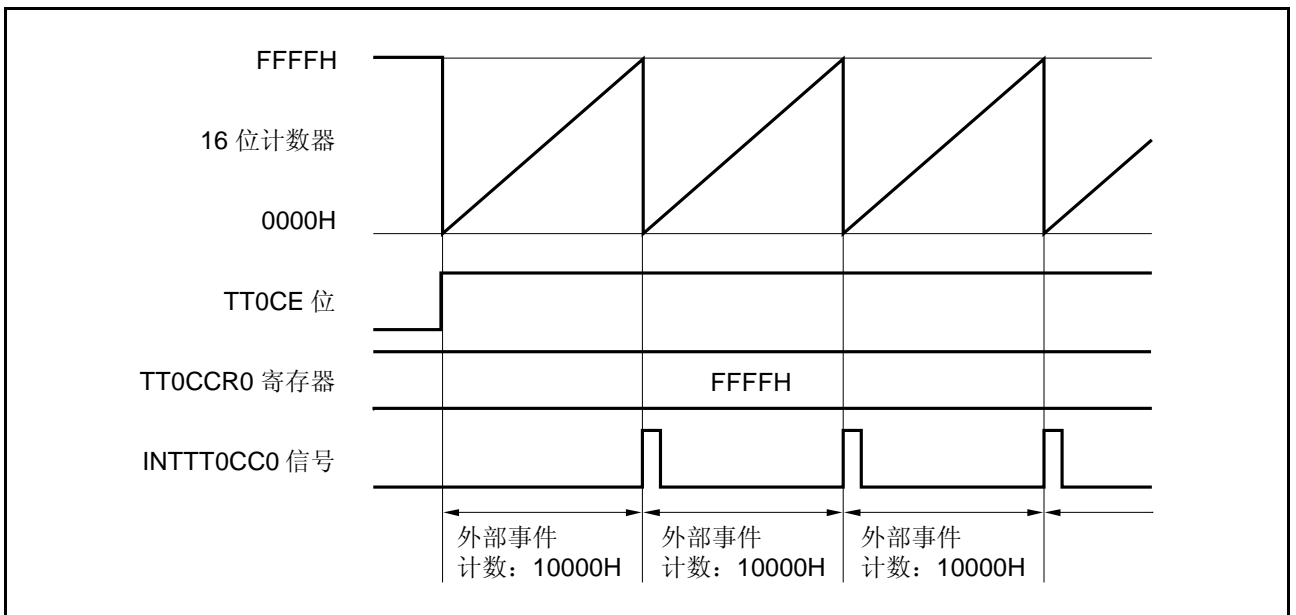
(a) 如果 TT0CCR0 寄存器被设置为 0000H 时的操作

如果 TT0CCR0 寄存器被设置为 0000H，16 位计数器每次检测到的外部事件输入信号有效边沿时，或者它的值和 CCR0 缓冲寄存器的值匹配时，都会被清除为 0000H，并且产生 INTTT0CC0 信号。16 位计数器的值总是 0000H。



(b) 如果 TT0CCR0 寄存器被设置为 FFFFH 时的操作

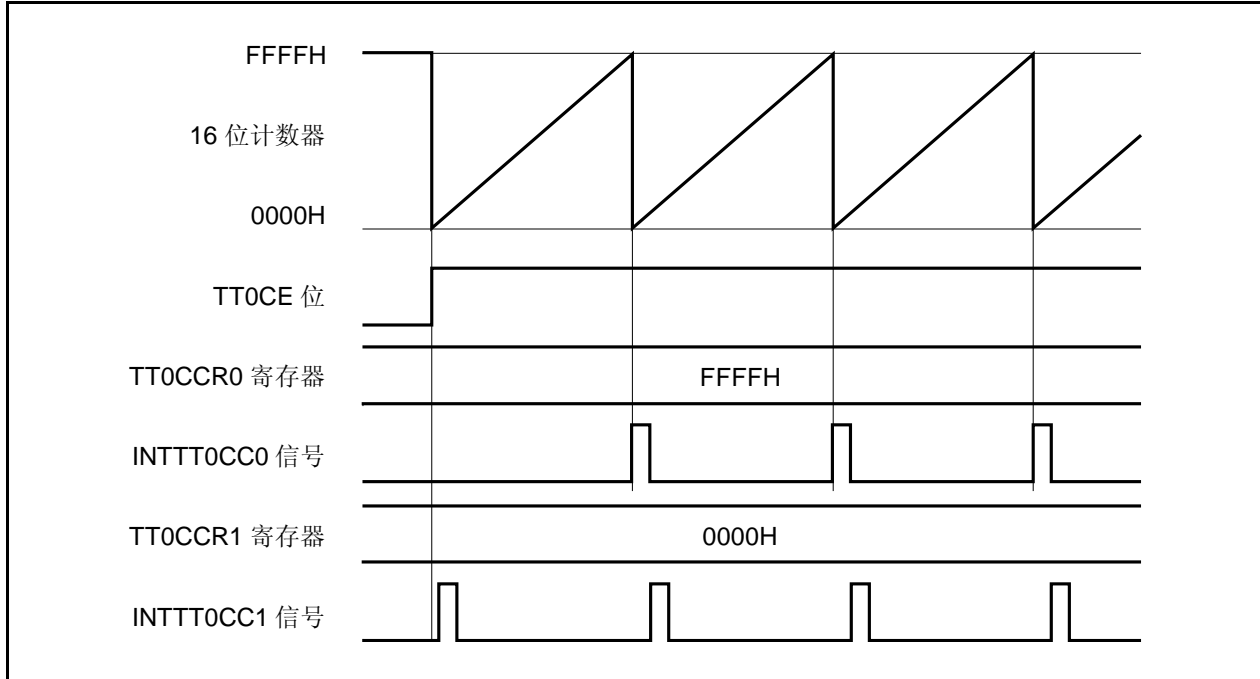
如果 TT0CCR0 寄存器被设置为 FFFFH，16 位计数器每次检测到的外部事件输入信号有效边沿时，都计数到 FFFFH。当下一个计数到来时，16 位计数器被同步清为 0000H，并且产生 INTTT0CC0 信号。此时，不设置 TT0OPT0.TT0OVF 位。



(c) TT0CCR0 寄存器被置为 FFFFH 且 TT0CCR1 寄存器被置为 0000H 的操作

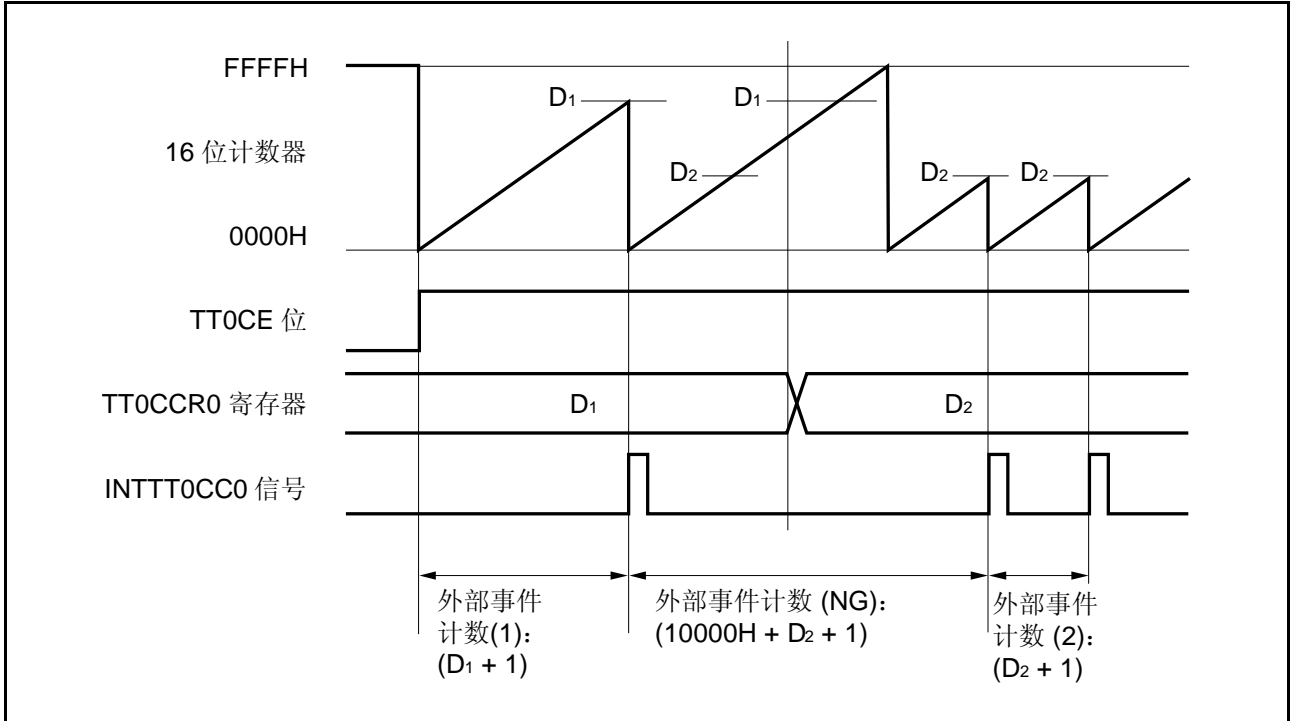
如果 TT0CCR0 寄存器被设置为 FFFFH，16 位计数器每次检测到的外部事件输入信号有效边沿时，都计数到 FFFFH。当下一个计数到来时，16 位计数器被同步清为 0000H，并且产生 INTTT0CC0 信号。此时，不设置 TT0OPT0.TT0OVF 位。

如果 TT0CCR1 寄存器被设置为 0000H，当 16 位计数器被清除为 0000H 时，产生 INTTT0CC1 信号。16 位计数器的值总是 0000H。



(d) 重写 TT0CCR0 寄存器的注意事项

如果在计数器进行计数期间，对寄存器 TT0CCR0 的值进行重写，则 16 位计数器可能会溢出。可能发生溢出时，应停止计数，然后改变其设定值。



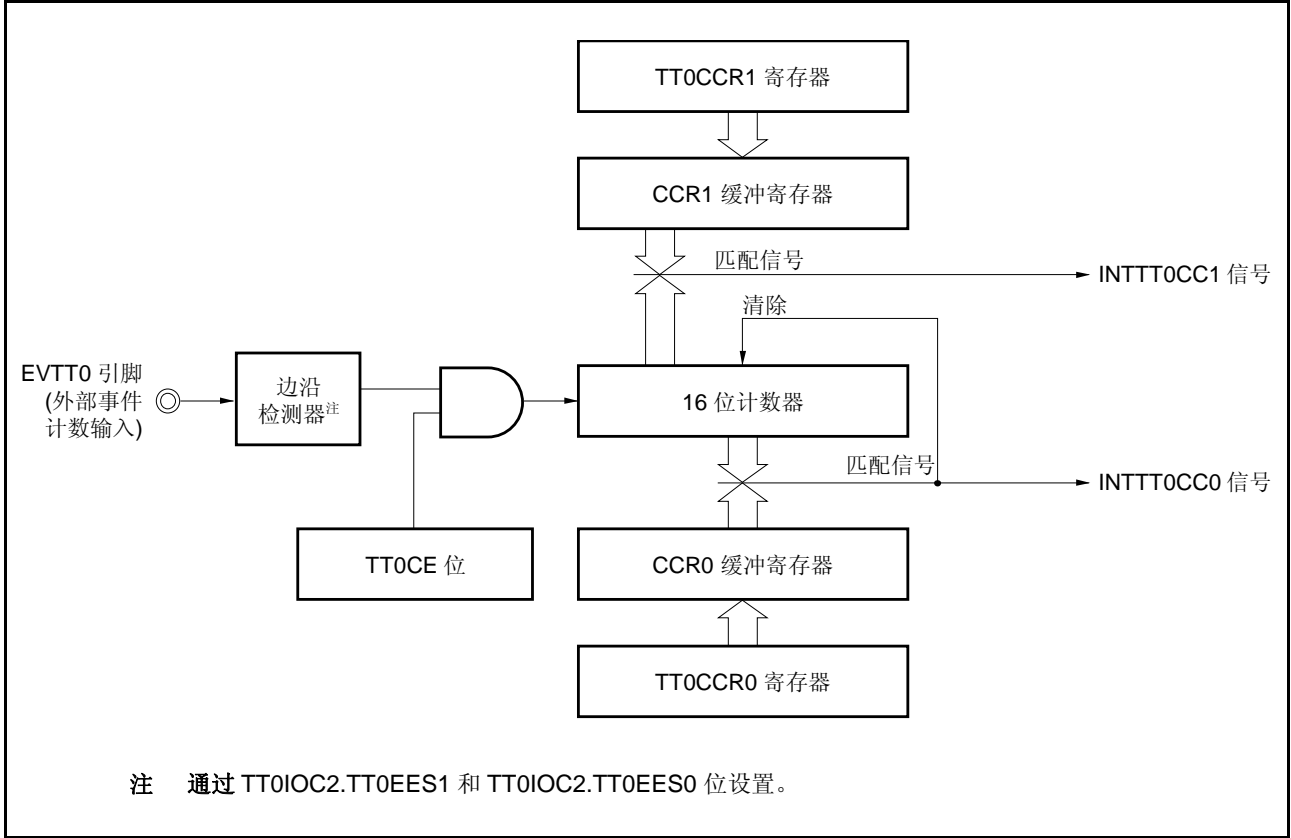
当 16 位计数器计数值大于 D_2 而小于 D_1 时，如果将 TT0CCR0 寄存器的值从 D_1 变为 D_2 ，则在重写 TT0CCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。因而，与 16 位计数器相比较的值会是 D_2 。

但是，由于计数值已经超过了 D_2 ，所以 16 位计数器就会向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值与 D_2 匹配时，会生成 INTTT0CC0 信号。

因此，可能不会在原先预计的“ $(D_1 + 1)$ 次”或“ $(D_2 + 1)$ 次”有效边沿生成 INTTT0CC0 中断请求信号，而可能在“ $(10000H + D_2 + 1)$ 次”有效边沿生成中断请求信号。

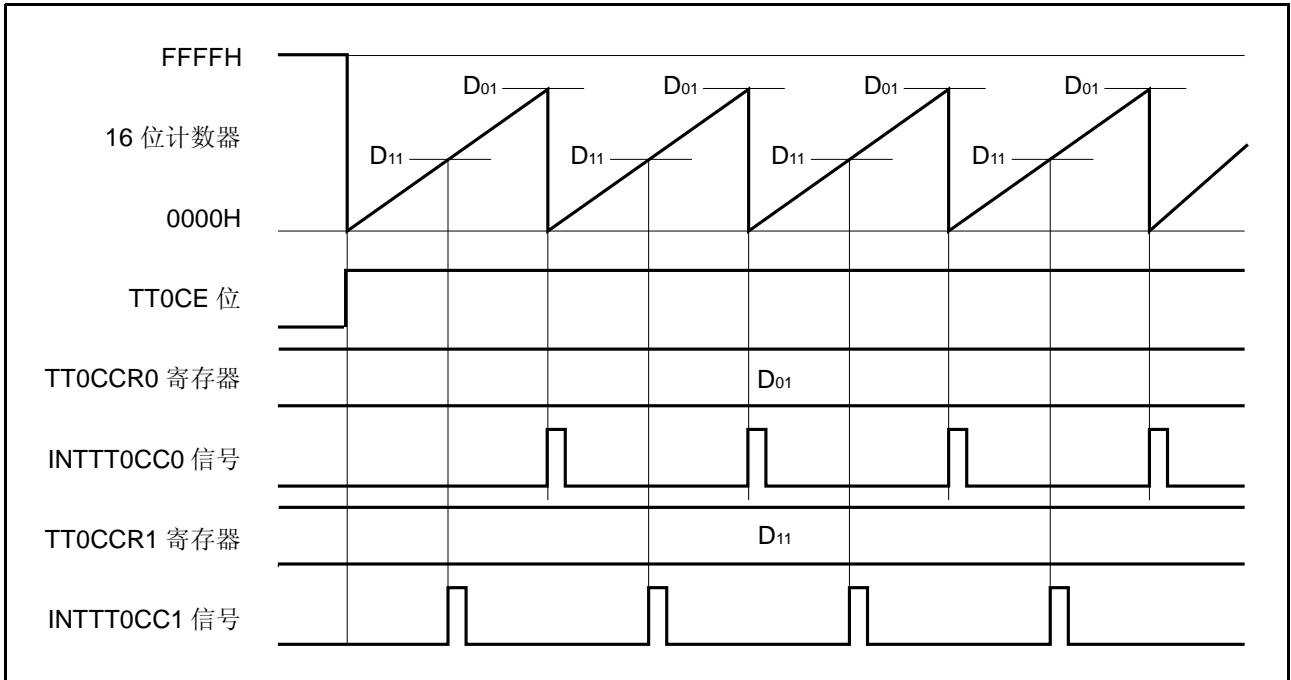
(e) TT0CCR1 寄存器的操作

图 9-18. TT0CCR1 寄存器的配置



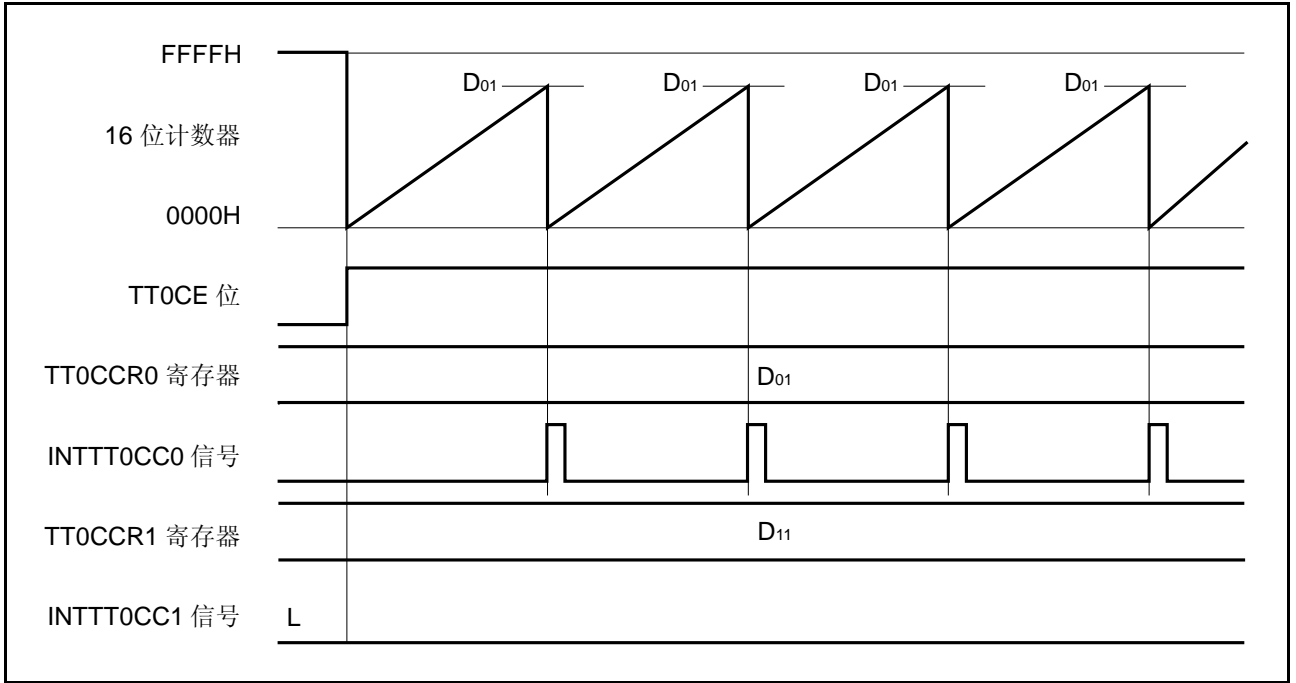
如果 TT0CCR1 寄存器的设定值小于 TT0CCR0 寄存器的设定值，则每个计数周期都产生一次 INTTT0CC1 中断请求信号。

图 9-19. 当 $D_{01} \geq D_{11}$ 时的时序图



如果 TT0CCR1 寄存器的设定值大于 TT0CCR0 寄存器的设定值，16 位计数器的计数值不会和 TT0CCR1 寄存器的值发生匹配。因此也就不会产生 INTTT0CC1 中断请求信号。
不使用 TT0CCR1 寄存器时，推荐将其值设置为 FFFFH。

图 9-20. 当 $D_{01} < D_{11}$ 时的时序图



9.6.3 外部触发脉冲输出模式 (TT0MD3 至 TT0MD0 位=0010)

在外部触发脉冲输出模式下，当 TT0CTL0.TT0CE 位被置为 1 时，16 位定时器/事件计数器 T 等待触发信号，当检测到外部触发输入(EVTT0)的有效边沿后，16 位定时器/事件计数器 T 开始计数，并且从 TOT01 引脚输出一个 PWM 波形。

不使用外部触发，而是使用软件触发，同样可以得到输出脉冲。当使用软件触发时，TOT00 引脚输出一个占空比 50%的方波，其半周期等于 TT0CCR0 寄存器的设置值+ 1。

图 9-21. 外部触发脉冲输出模式的配置

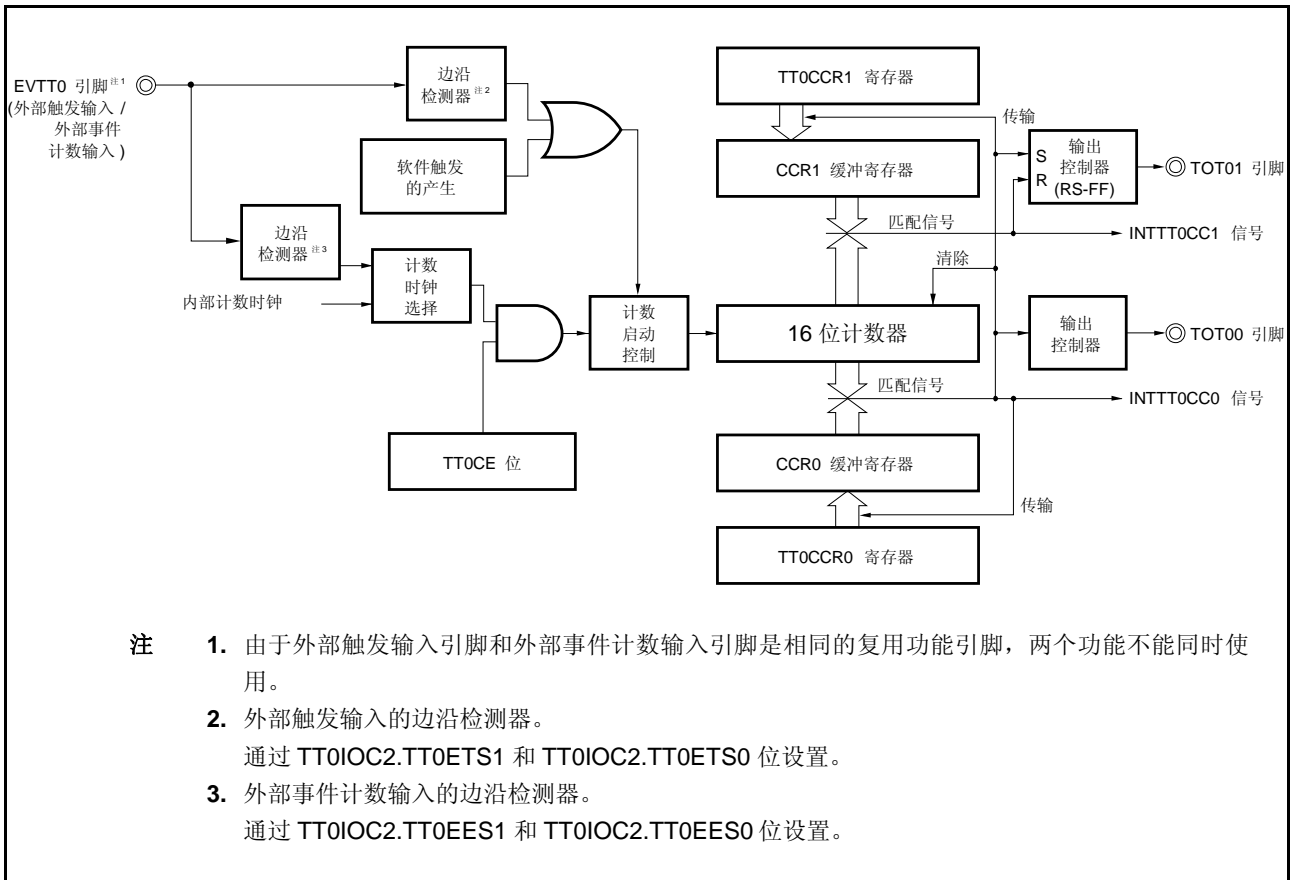
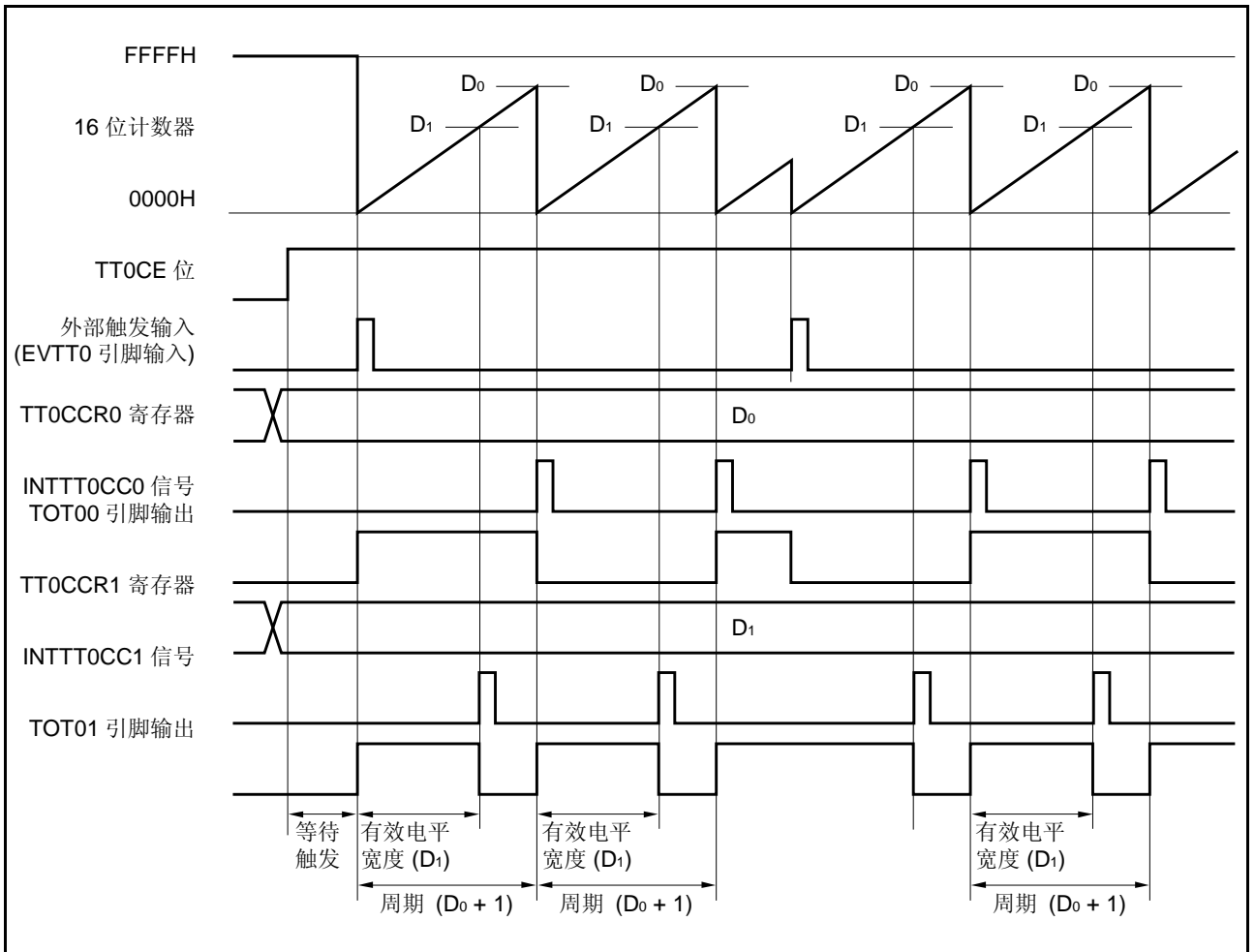


图 9-22. 外部触发脉冲输出模式的基本时序



当 TT0CE 位设置为 1 时，16 位定时器/事件计数器 T 等待触发。当触发产生时，16 位计数器从 FFFFH 清零为 0000H，同时开始计数，并从 TOT01 引脚输出一个 PWM 波形。如果在计数运行中再次产生触发，则计数器清为 0000H 并重新开始计数。（TOT00 引脚输出反相。在产生触发时，不管当前状态如何（高/低），TOT01 引脚都会输出一个高电平。

PWM 波形的有效电平宽度、周期、占空比系数的计算如下。

有效电平宽度 = (TT0CCR1 寄存器的设置值) × 计数时钟周期

周期 = (TT0CCR0 寄存器的设置值 + 1) × 计数时钟周期

占空比系数 = (TT0CCR1 寄存器的设置值) / (TT0CCR0 寄存器的设置值 + 1)

当 16 位计数器在其计数值与 CCR0 缓冲寄存器的值匹配之后进行下一次计数时就会同步产生比较匹配（中断）请求信号 INTTT0CC0，且 16 位计数器清零为 0000H。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTT0CC1。

当 16 位计数器的计数值与 CCRn 缓冲寄存器的值匹配，而且 16 位计数器清零为 0000H 时，TT0CCRn 寄存器的设置值就被传送到 CCRn 缓冲寄存器。

外部触发输入 (EVTT0) 的有效边沿，或将软件触发 (TT0CTL1.TT0EST 位) 设为 1，都可以用作触发信号 (n = 0, 1)。

图 9-23. 外部触发脉冲输出模式的寄存器设置 (1/2)

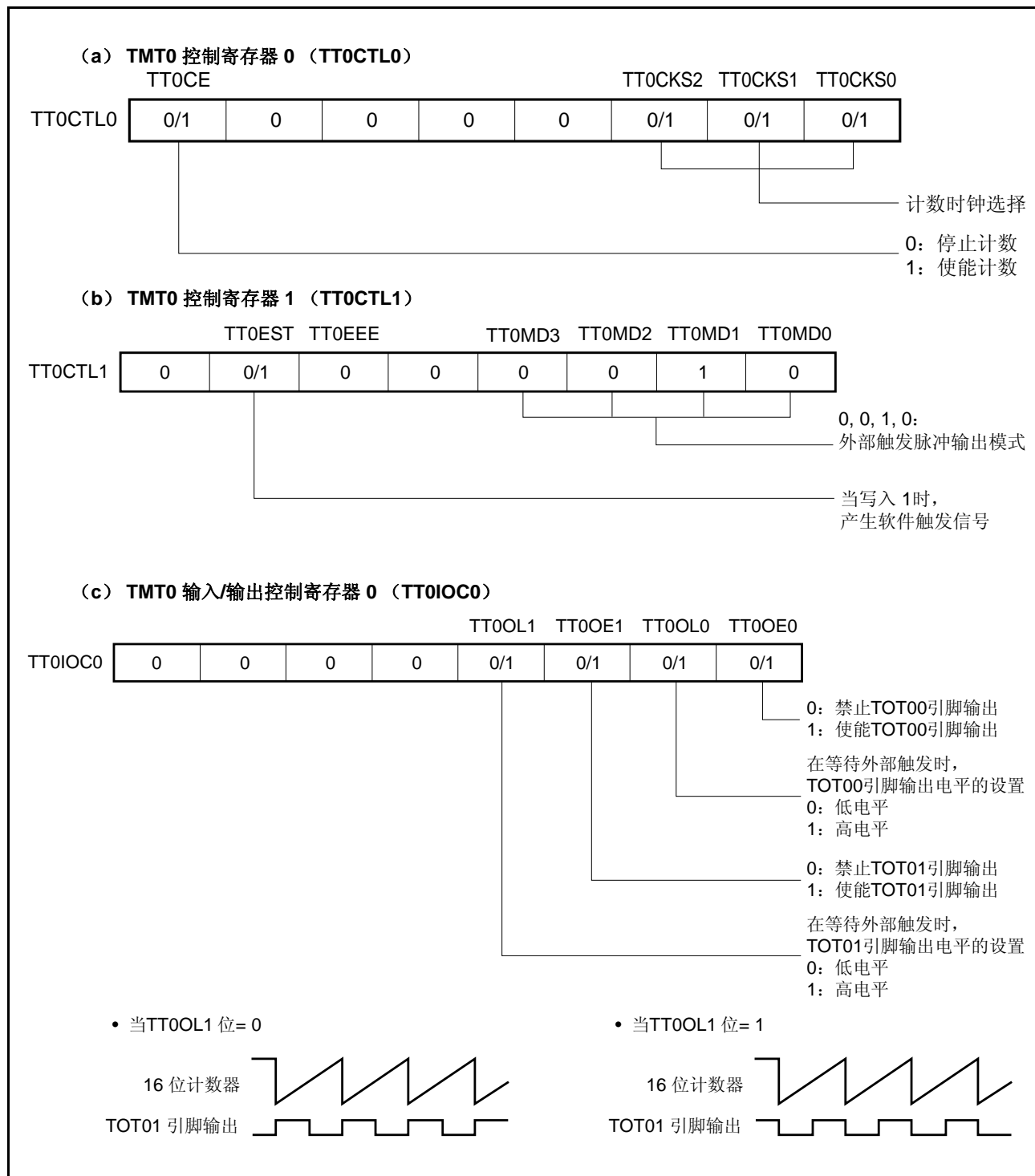
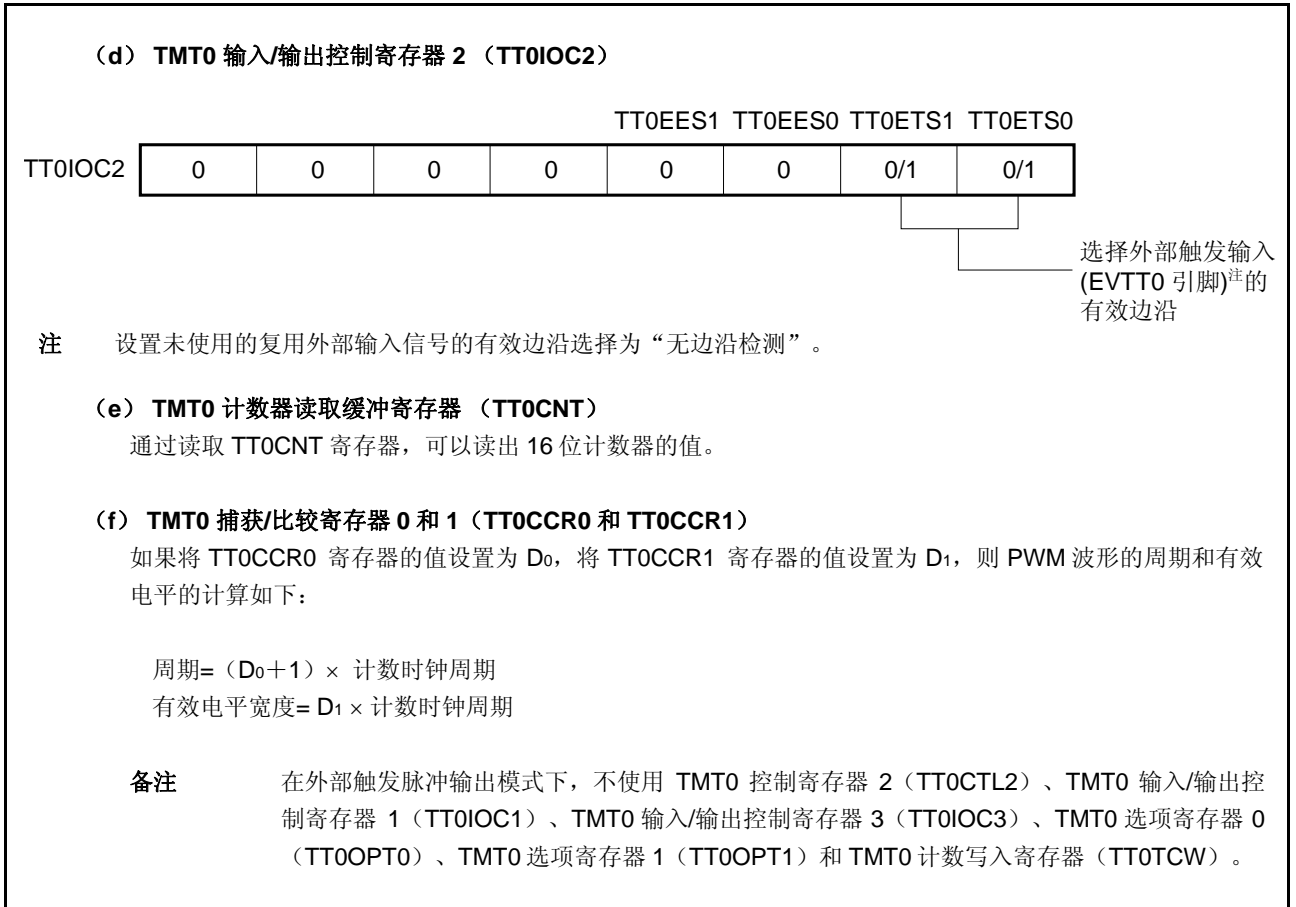


图 9-23. 外部触发脉冲输出模式的寄存器设置 (2/2)



(1) 外部触发脉冲输出模式的操作流程

图 9-24. 外部触发脉冲输出模式的软件处理流程 (1/2)

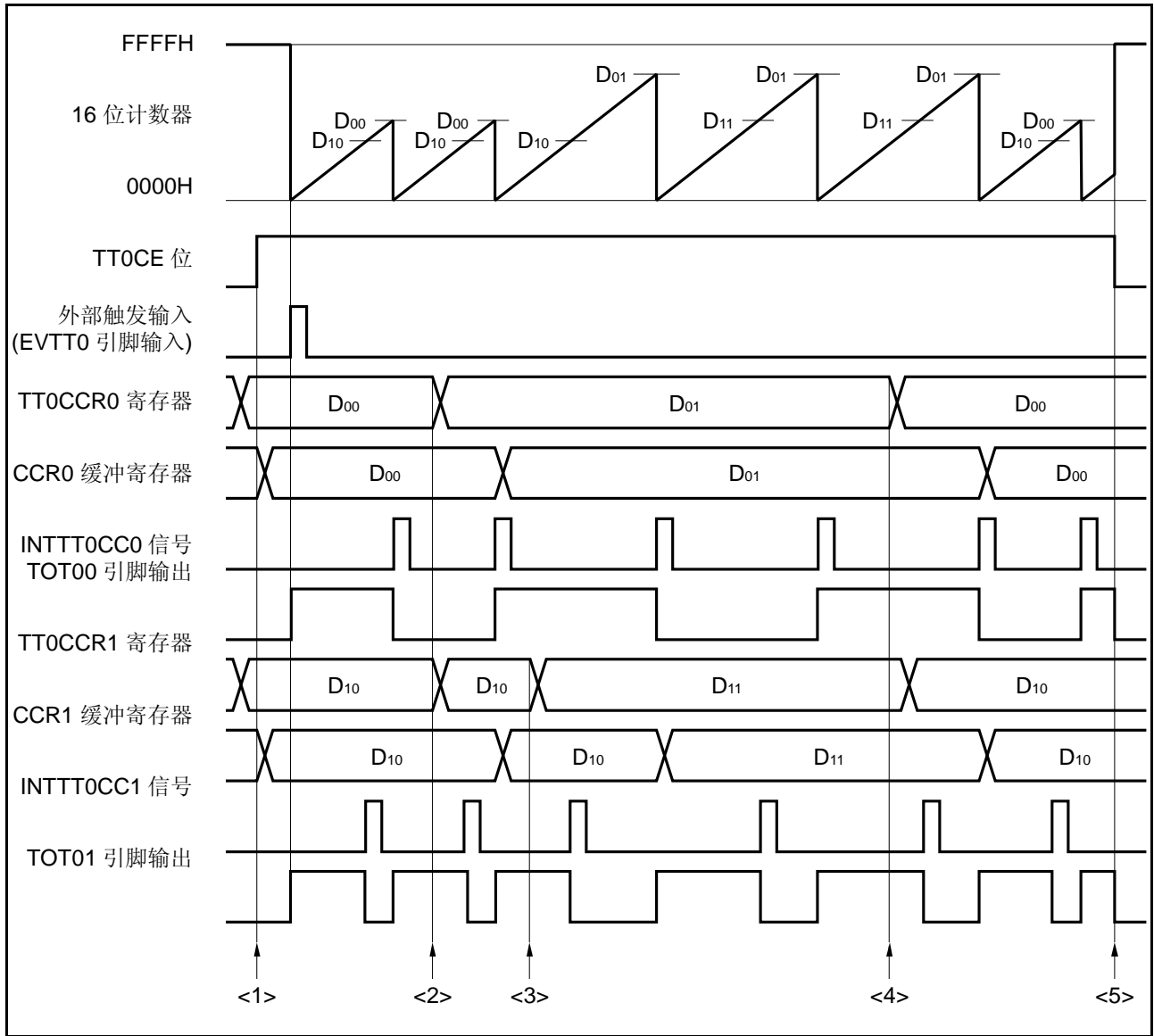
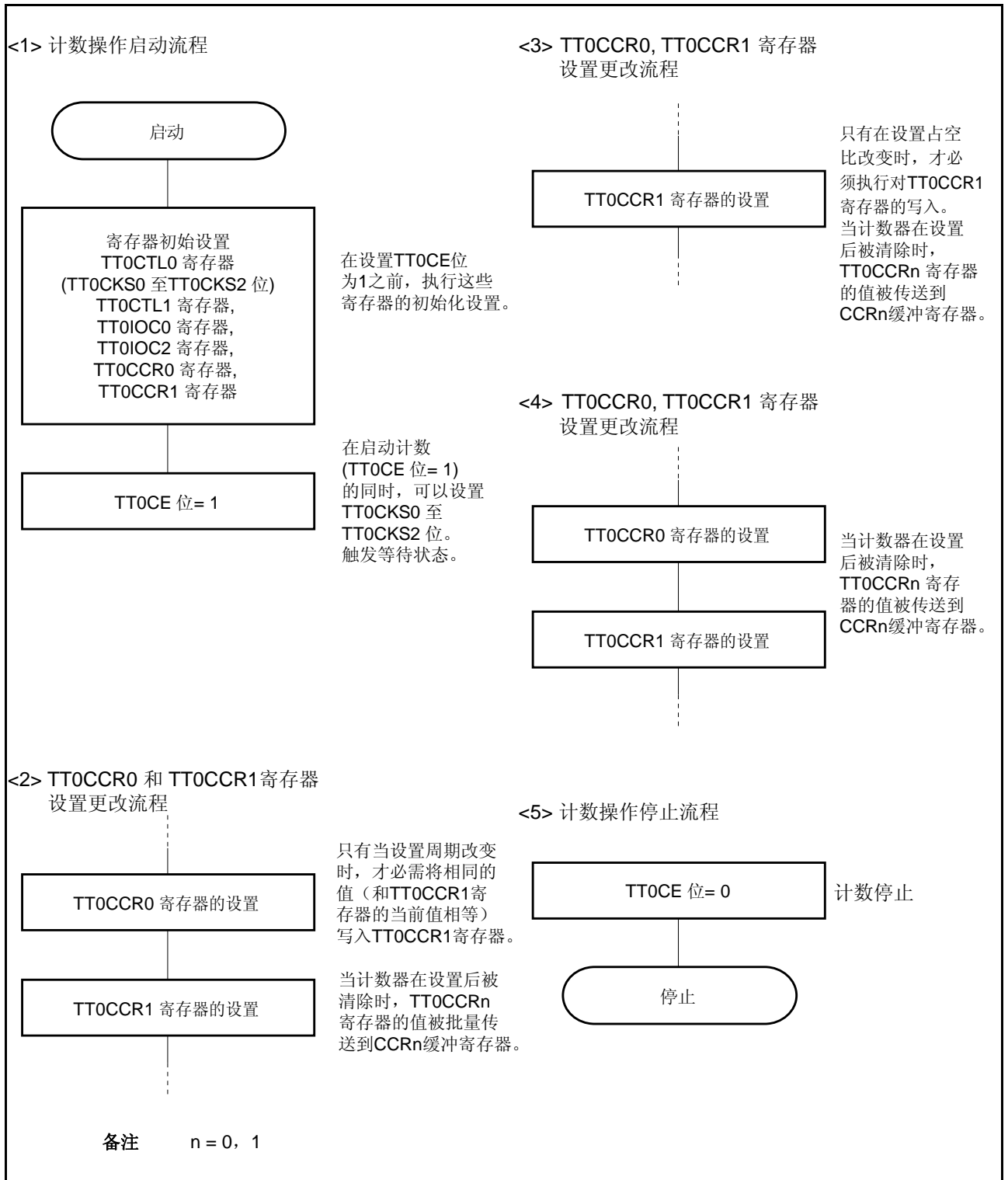


图 9-24. 外部触发脉冲输出模式的软件处理流程 (2/2)

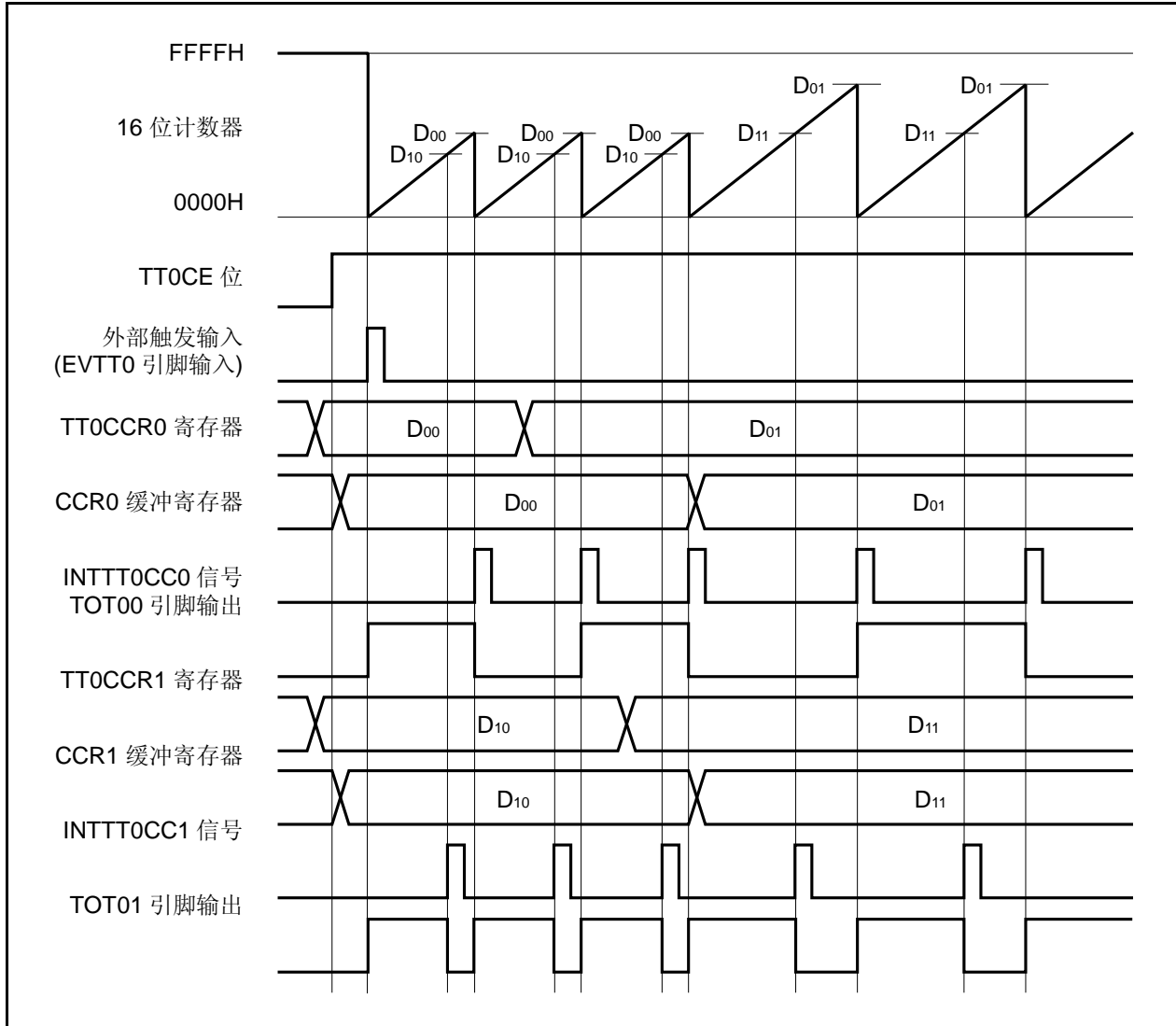


(2) 外部触发脉冲输出模式的操作时序

(a) 有关操作中改变脉冲宽度的注意事项

计数器计数过程中，若要改变 PWM 波形，应重新写入 TT0CCR1 寄存器。

当检测到 INTTT0CC0 信号以后，在 TT0CCR1 寄存器写入操作之后重写 TT0CCRn 寄存器。



为了将数据从 TT0CCRn 寄存器传送到 CCRn 缓冲寄存器，必须写入 TT0CCR1 寄存器。

如果 PWM 波形的周期和有效电平宽度都要改变，先将周期设定到 TT0CCR0 寄存器，将有效电平宽度设定到 TT0CCR1 寄存器。

如果只改变 PWM 波形的周期，应先把周期设置到 TT0CCR0 寄存器，然后再将相同的值（和 TT0CCR1 寄存器的当前值相等）写入 TT0CCR1 寄存器。

如果只改变 PWM 波形的有效电平宽度（占空比系数），仅需设置 TT0CCR1 寄存器。

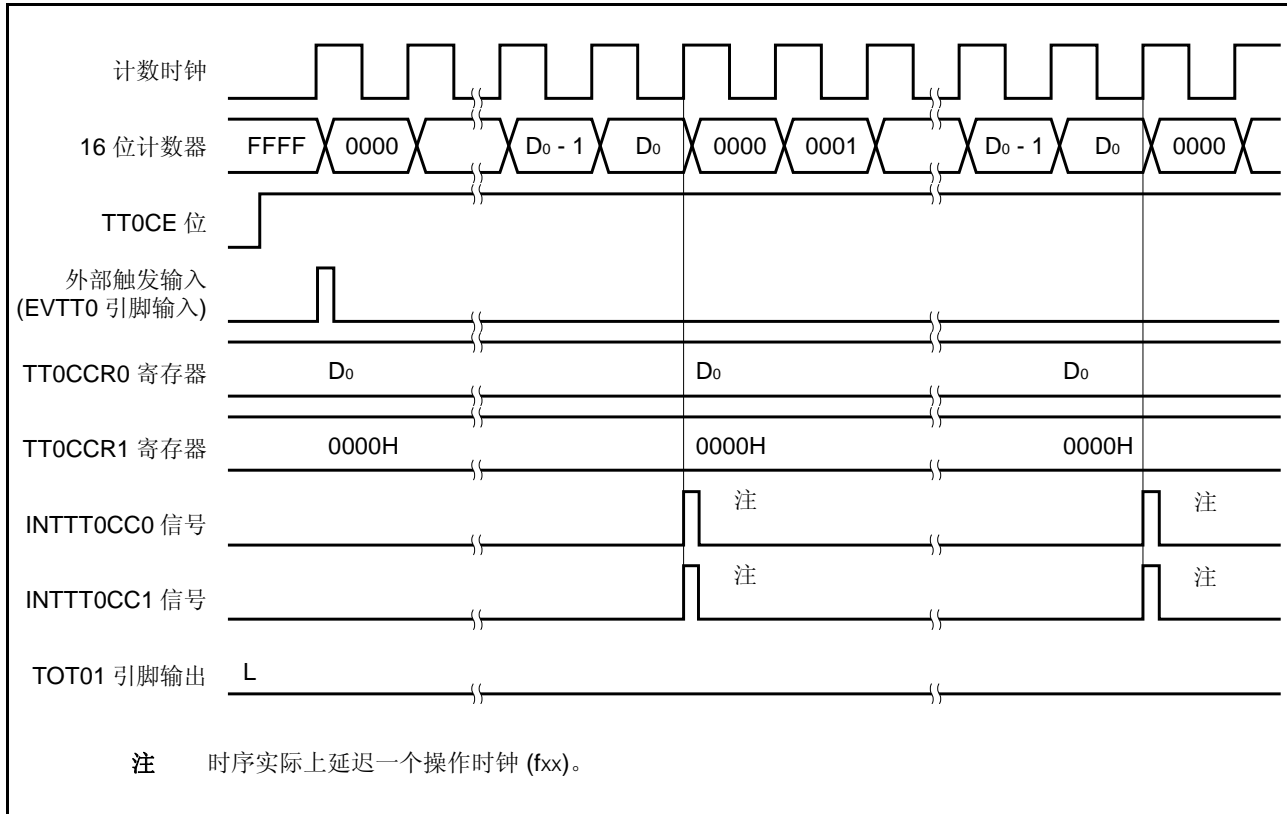
在数据写入 TT0CCR1 寄存器后，在对 16 位计数器清零的同时，写入到 TT0CCRn 寄存器的值会被传送到 CCRn 缓冲寄存器，并作为与 16 位计数器进行比较的值。

在写入 TT0CCR1 寄存器一次之后，要再次写入 TT0CCR0 或 TT0CCR1 寄存器，应该在 INTTT0CC0 信号产生之后进行。否则，CCRn 缓冲寄存器的值可能不确定，因为把数据从 TT0CCRn 寄存器传送到 CCRn 缓冲寄存器的时序会和写入 TT0CCRn 寄存器的时序发生冲突。

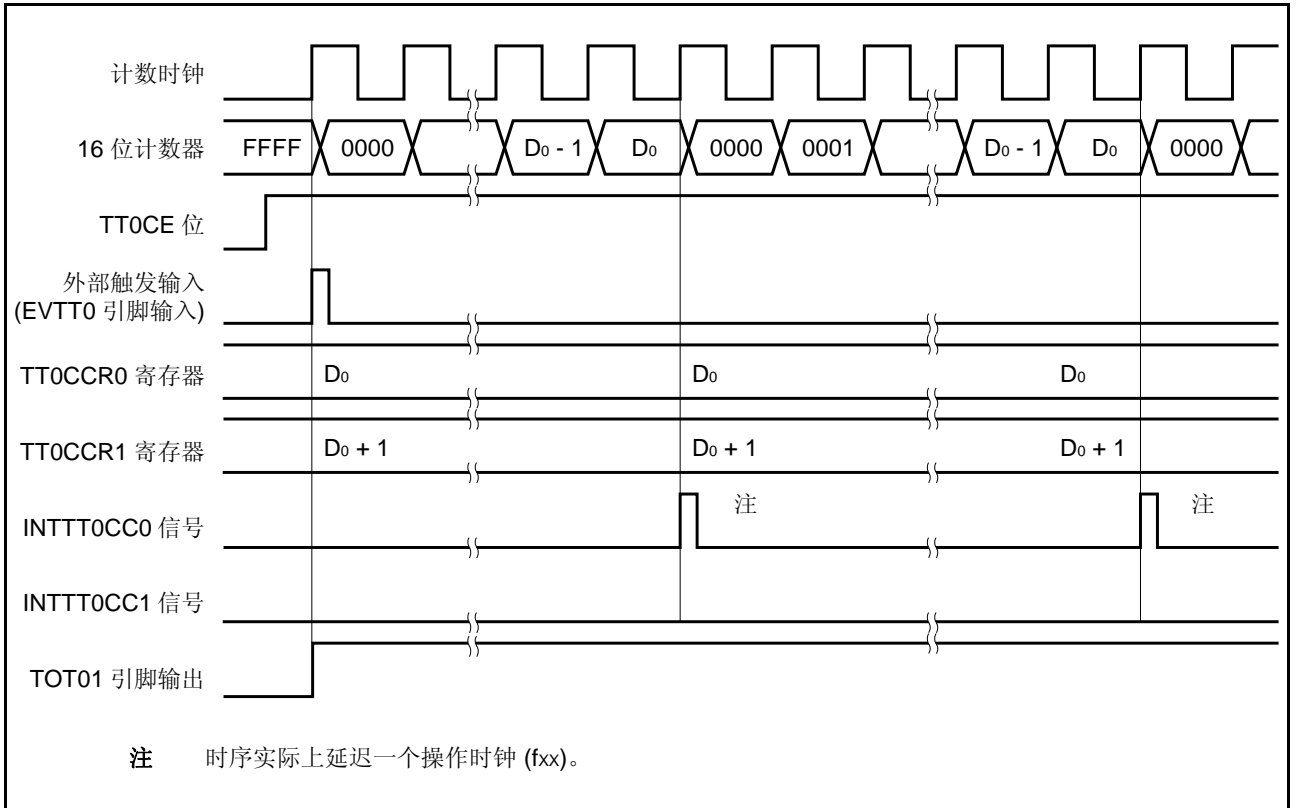
备注 $n = 0, 1$

(b) PWM 波形的 0% 和 100% 输出

为了输出 0% 的波形，需要将 TT0CCR1 寄存器设置为 0000H。16 位计数器的计数值和 CCR0 缓冲寄存器匹配之后，16 位计数器被清除为 0000H，且产生 INTTT0CC0 和 INTTT0CC1 中断请求信号。

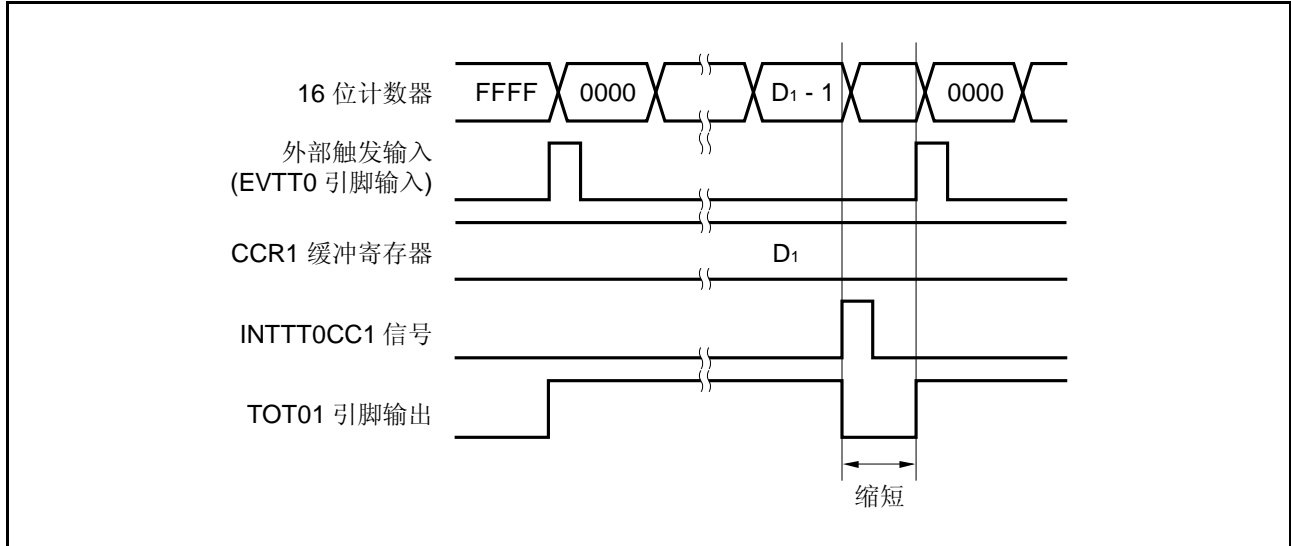


为了输出 100%的波形，需要将 TT0CCR1 寄存器的值设为： $(TT0CCR0 \text{ 寄存器的设定值} + 1)$ 。如果 TT0CCR0 寄存器的设定值为 FFFFH，则不会产生 100%波形输出。

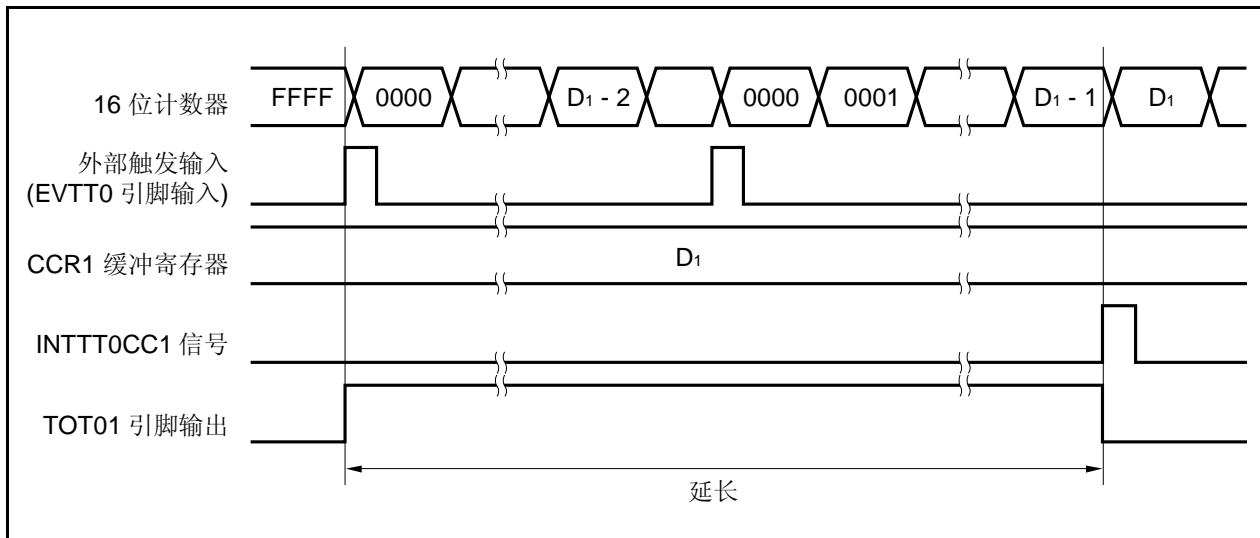


(c) 触发检测和 CCR1 缓冲寄存器匹配之间的冲突

如果在 INTTT0CC1 信号产生后，立刻检测到触发信号，则 16 位计数器立刻清零为 0000H，TOT01 引脚的输出信号生效，计数器继续计数。结果，PWM 波形的无效时段被缩短。

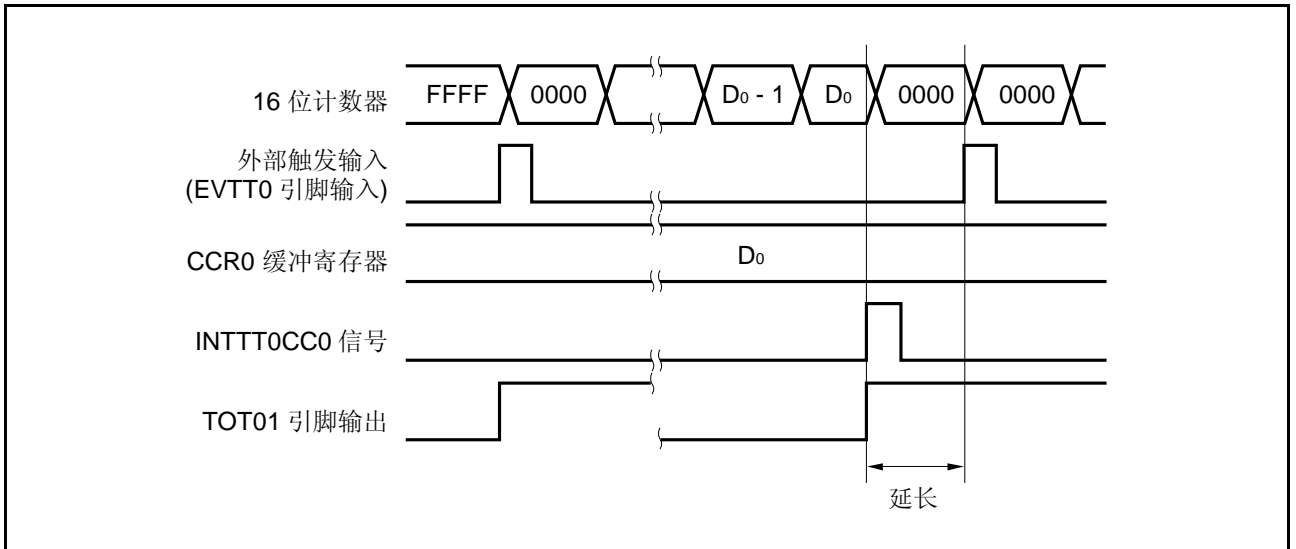


如果紧邻在 INTTT0CC1 信号产生之前检测到触发信号，则不会产生 INTTT0CC1 信号，而 16 位计数器清零为 0000H，计数继续进行。TOT01 引脚的输出信号仍然保持有效。结果，PWM 波形的有效时段被延长。

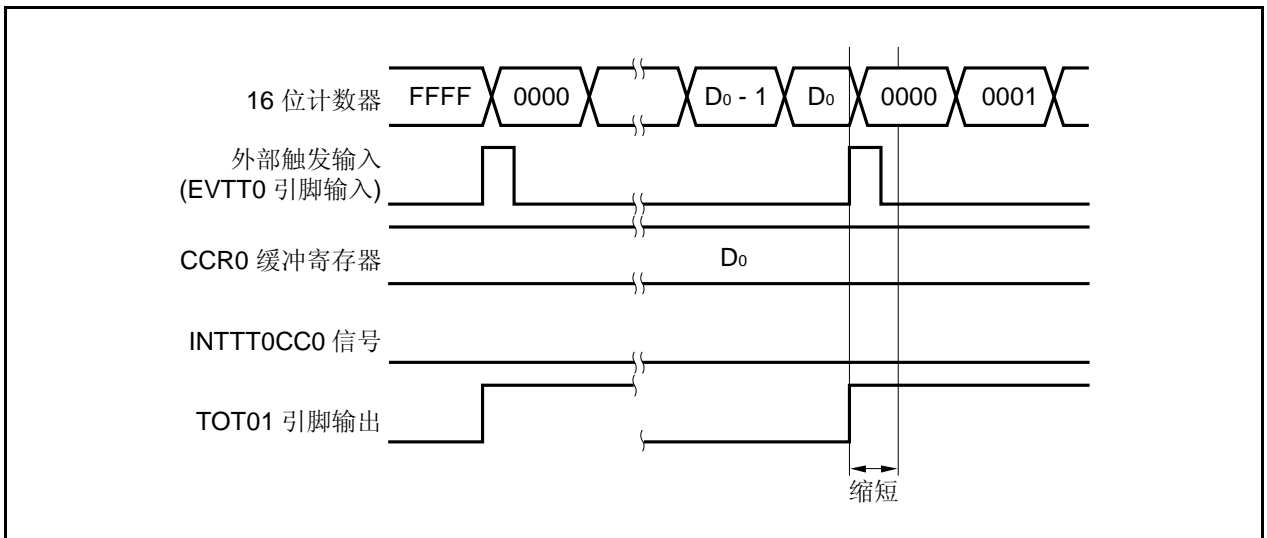


(d) 触发检测和 CCR0 缓冲寄存器匹配之间的冲突

如果在 INTTT0CC0 信号产生后，立刻检测到触发信号，则 16 位计数器立刻清零为 0000H，并继续计数。因此，TOT01 引脚的有效时段被延长，延长长度为 INTTT0CC0 信号产生时刻至检测到触发信号时刻的时间间隔。

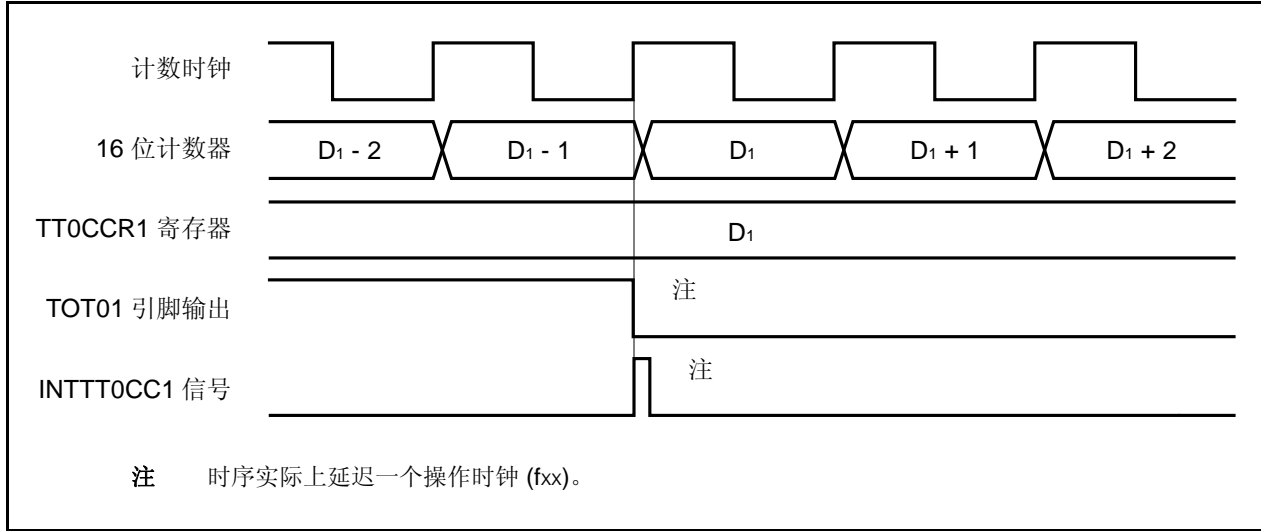


如果紧邻在 INTTT0CC0 信号产生之前检测到触发信号，则不会产生 INTTT0CC0 信号。16 位计数器清零为 0000H，TOT01 引脚输出有效，计数器继续计数。结果，PWM 波形的无效时段被缩短。



(e) 比较匹配中断请求信号 (INTTT0CC1) 的发生时序

在外部触发脉冲输出模式下, INTTT0CC1 信号的发生时序与其它情况下 INTTT0CC1 信号的发生时序不同; 当 16 位计数器的计数值与 TT0CCR1 缓冲寄存器的值匹配时, 产生 INTTT0CC1 中断请求信号。



通常在 16 位计数器的当前计数值与 TT0CCR1 缓冲寄存器的值匹配之后, 在下次计数时同步发生 INTTT0CC1 信号。

但是, 在外部触发脉冲输出模式下, INTTT0CC1 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOT01 引脚输出信号变化的时序。

9.6.4 单次触发脉冲输出模式 (TT0MD3 至 TT0MD0 位 = 0011)

在单次触发脉冲输出模式下，当 TT0CTL0.TT0CE 位被置为 1 时，16 位定时器/事件计数器 T 等待触发。在检测到外部触发输入 (EVTTO) 的有效边沿时，16 位定时器/事件计数器 T 开始计数，同时从 TOT01 引脚输出一个单次触发脉冲。

除外部触发输入 (EVTTO) 外，也可以通过软件触发信号来输出该脉冲。在使用软件触发时，TOT00 引脚在 16 位计数器计数时输出有效电平，在计数器停止 (等待触发) 时输出无效电平。

图 9-25. 单次触发脉冲输出模式的配置

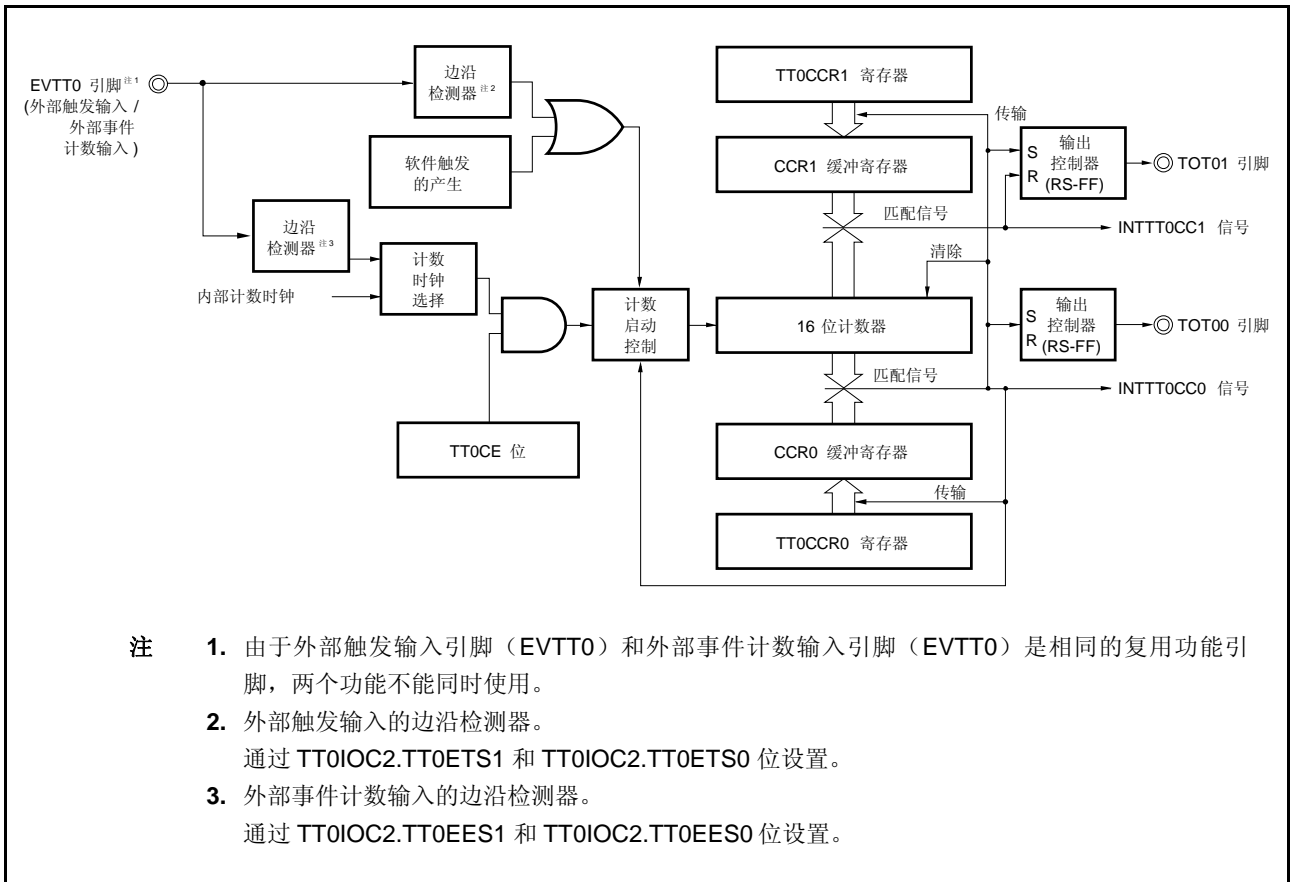
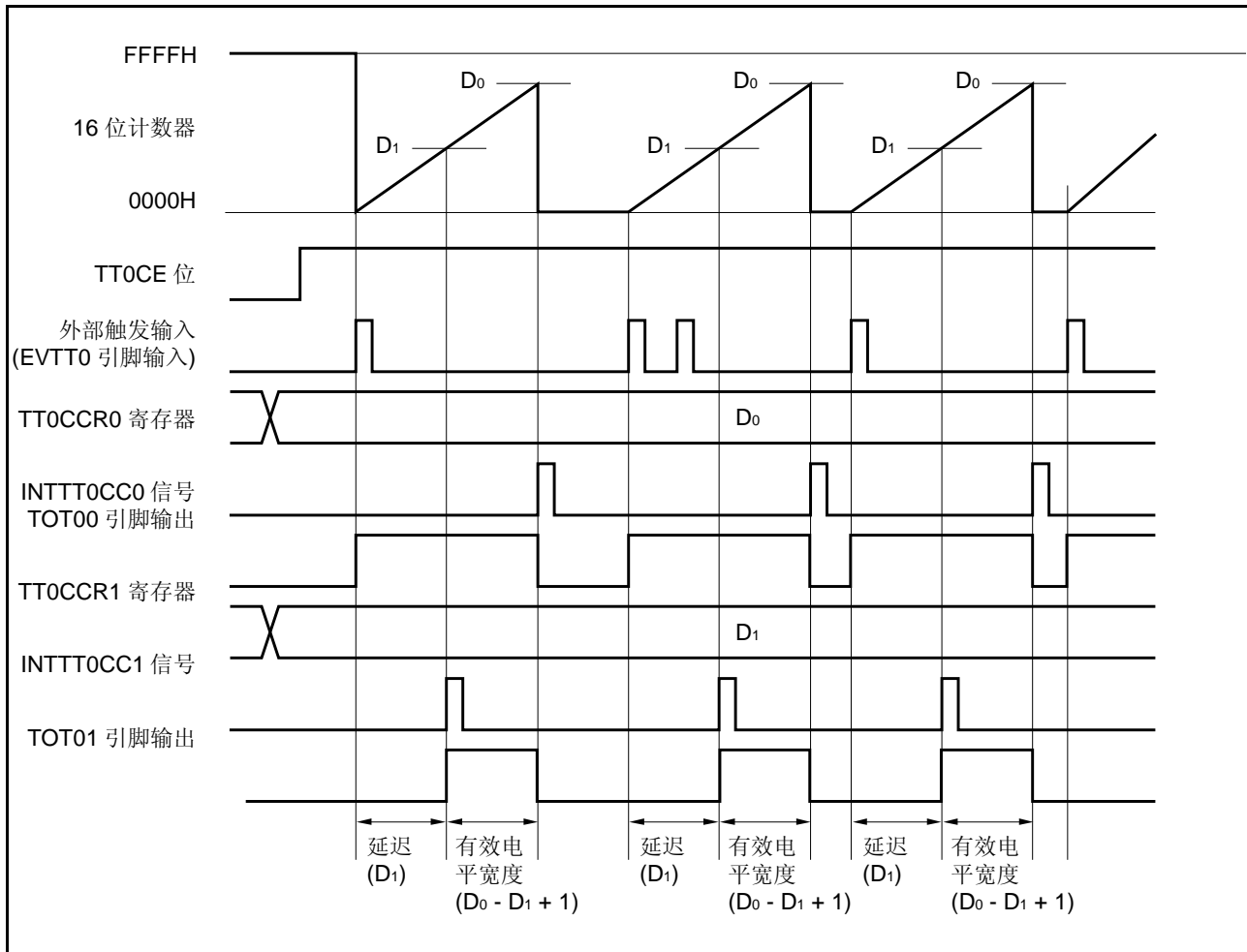


图 9-26. 单次触发脉冲输出模式的基本时序



当 TT0CE 位置为 1 时，16 位定时器/事件计数器 T 等待触发。在触发信号发生时，16 位计数器从 FFFFH 清零为 0000H，开始进行计数，并从 TOT01 引脚输出一个单次触发脉冲。在输出单次触发脉冲后，16 位计数器被设置为 FFFFH，计数停止，并重新等待触发。当触发信号再次产生，16 位计数器从 0000H 开始计数。如果在单次触发脉冲输出期间再次产生触发信号，则该触发信号被忽略。

单次触发脉冲的输出延迟时间和有效电平宽度可计算如下：

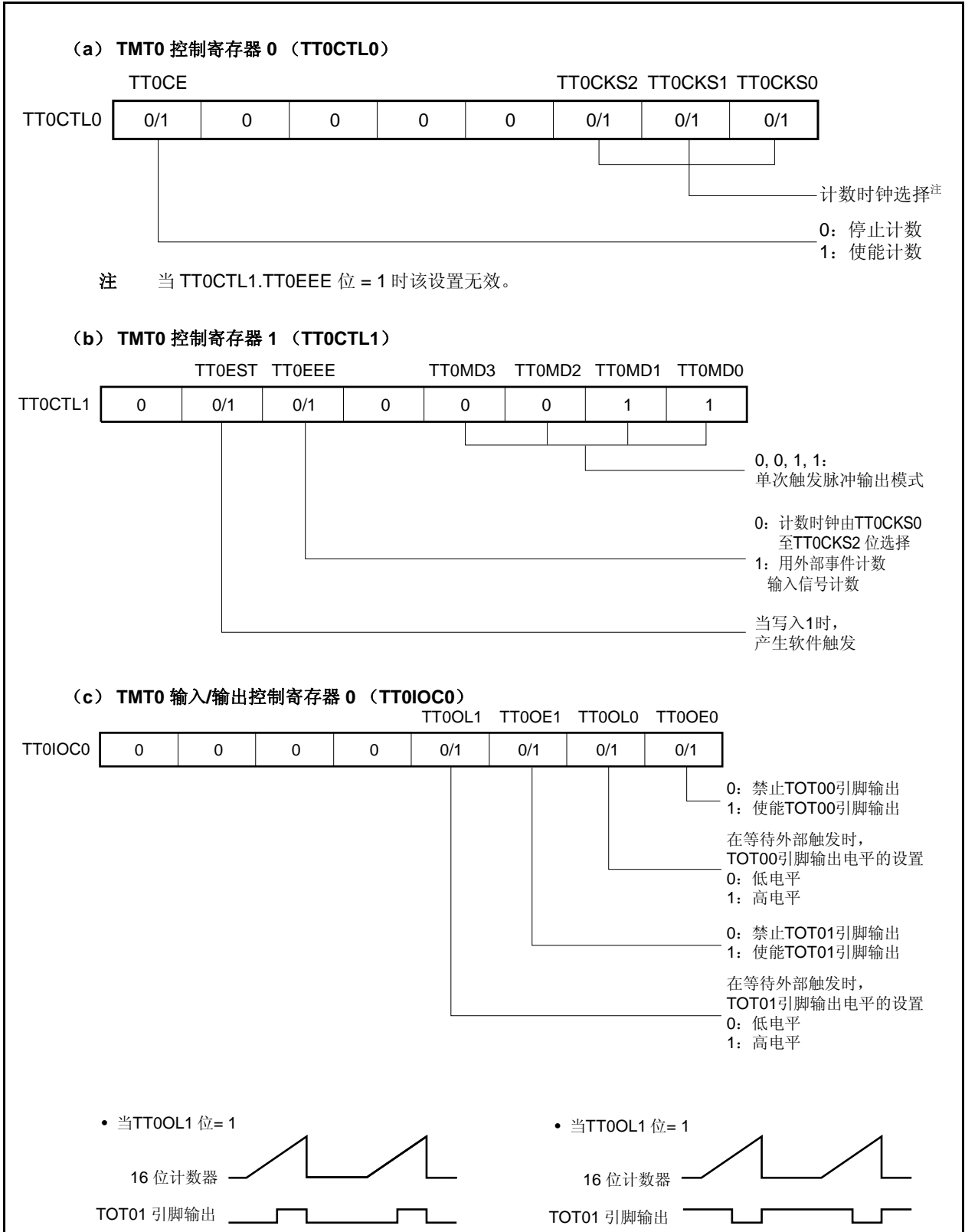
输出延迟时间 = (TT0CCR1 寄存器的设定值) × 计数时钟周期

有效电平宽度 = (TT0CCR0 寄存器的设定值 - TT0CCR1 寄存器的设定值 + 1) × 计数时钟周期

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配之后，产生比较匹配中断请求信号 INTTT0CC0。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTT0CC1。

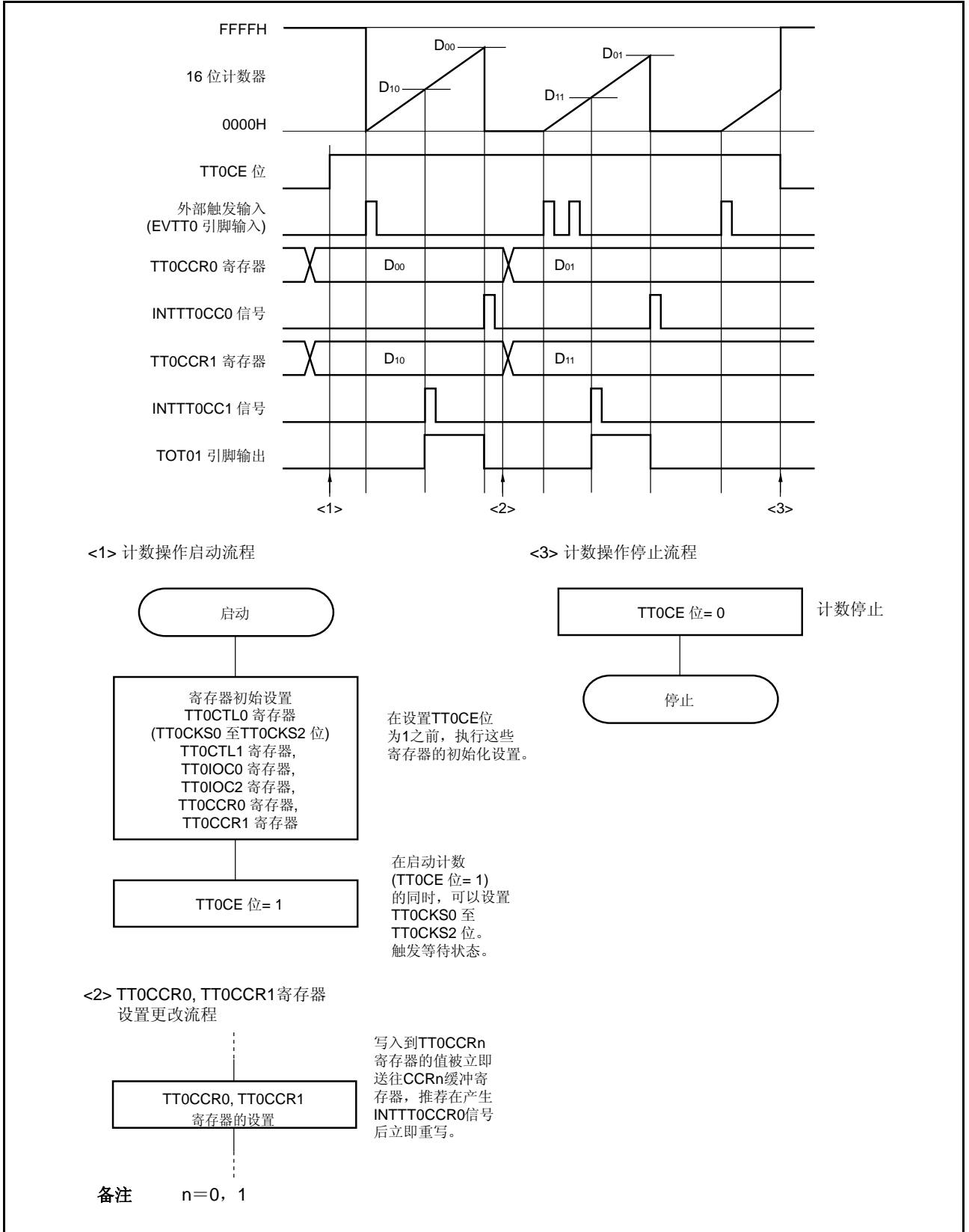
外部触发输入 (EVTTO 引脚) 的有效边沿或将软件触发 (TT0CTL1.TT0EST 位) 信号设置为 1，都可被用作触发信号。

图 9-27. 单次触发脉冲输出模式时的寄存器设置 (1/2)



(1) 单次触发脉冲输出模式下的操作流程

图 9-28. 单次触发脉冲输出模式的软件处理流程

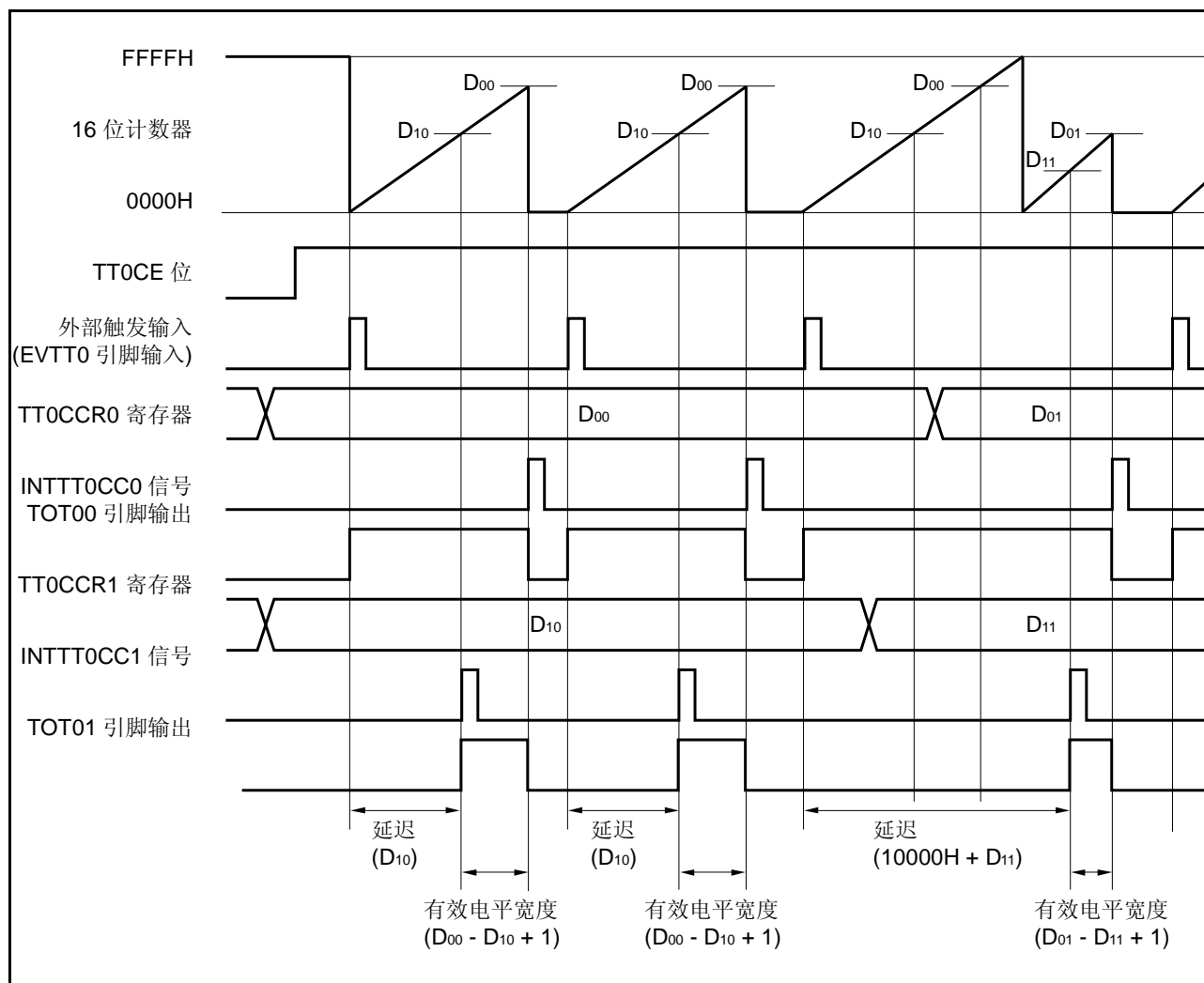


(2) 单次触发脉冲输出模式时的操作时序

(a) 重写 TT0CCRn 寄存器的注意事项

如果在计数器工作期间，将 TT0CCR0 寄存器的值重写为比当前值较小的值，则 16 位计数器可能会溢出。如果可能发生溢出，应停止计数，然后改变其设定值。

备注 n=0, 1

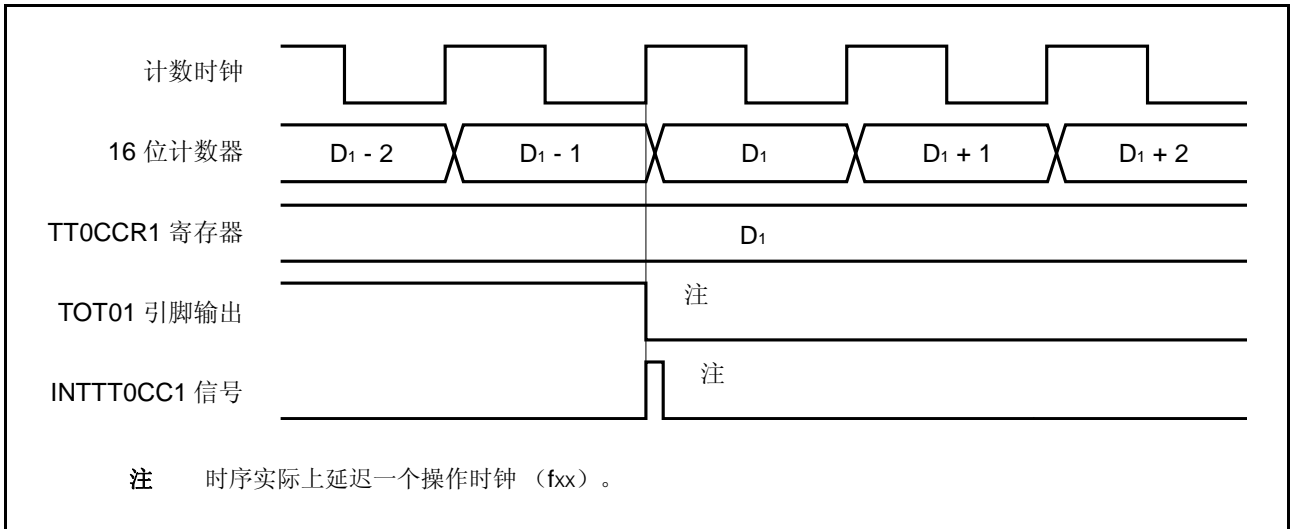


假设 $D_{00} > D_{01}$ 且 $D_{10} > D_{11}$ ，当 TT0CCR0 寄存器由 D_{00} 重写为 D_{01} ，TT0CCR1 寄存器由 D_{10} 重写为 D_{11} 时，如果在 16 位计数器的计数值大于 D_{11} 且小于 D_{10} 时重写 TT0CCR1 寄存器，并且在计数值大于 D_{01} 且小于 D_{00} 时重写 TT0CCR0 寄存器，则一旦重写寄存器，就会在各设置值上反映出来，并立即与计数值进行比较。计数器向上计数到 FFFFH 后，从 0000H 开始重新向上计数。当计数值与 D_{11} 匹配时，计数器会产生 INTTT0CC1 信号并使 TOT01 引脚有效。当计数值与 D_{01} 匹配时，计数器产生 INTTT0CC0 信号，使 TOT01 引脚失效，并停止计数。

因此，计数器输出脉冲的延迟时段或有效时段可能会与原先预计的单次触发脉冲有所不同。

(b) 比较匹配中断请求信号 (INTTT0CC1) 的发生时序

单次触发脉冲输出模式下，比较匹配中断请求信号 (INTTT0CC1) 的发生时序和其它情况下的 INTTT0CC1 信号发生时序不同，当 16 位计数器的计数值与 TT0CCR1 寄存器的值匹配时，产生 INTTT0CC1 中断请求信号。



通常在 16 位计数器的当前计数值与 TT0CCR1 寄存器的值匹配之后，在下次计数时同步发生 INTTT0CC1 信号。

但是，在单次触发脉冲输出模式下，INTTP0CC1 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOT01 引脚输出信号变化的时序。

9.6.5 PWM输出模式 (TT0MD3 至 TT0MD0 位=0100)

在 PWM 输出模式下, 当 TT0CTL0.TT0CE 位被置为 1 时, PWM 波形从 TOT01 引脚输出。
而且, 由 TOT00 引脚输出占空比 50% 的方波, 其半周期等于 TT0CCR0 寄存器的设置值 + 1。

图 9-29. PWM 输出模式的配置

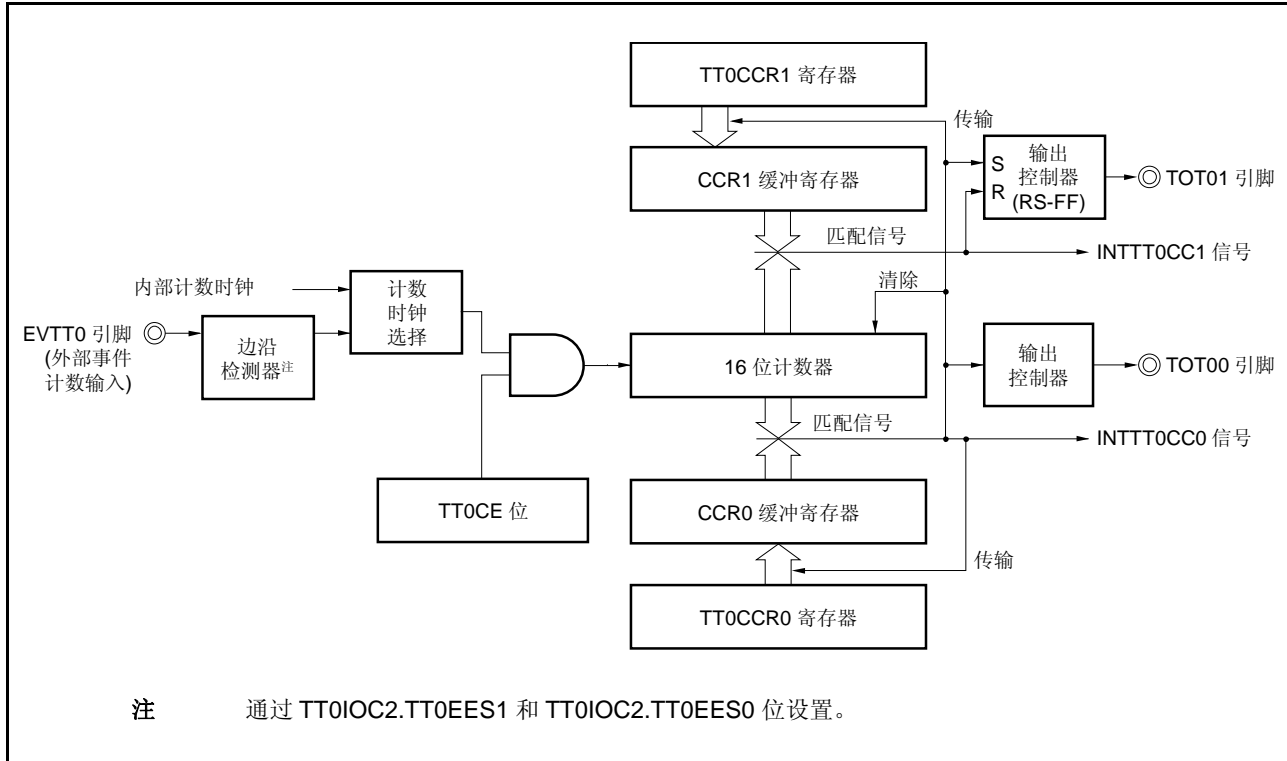
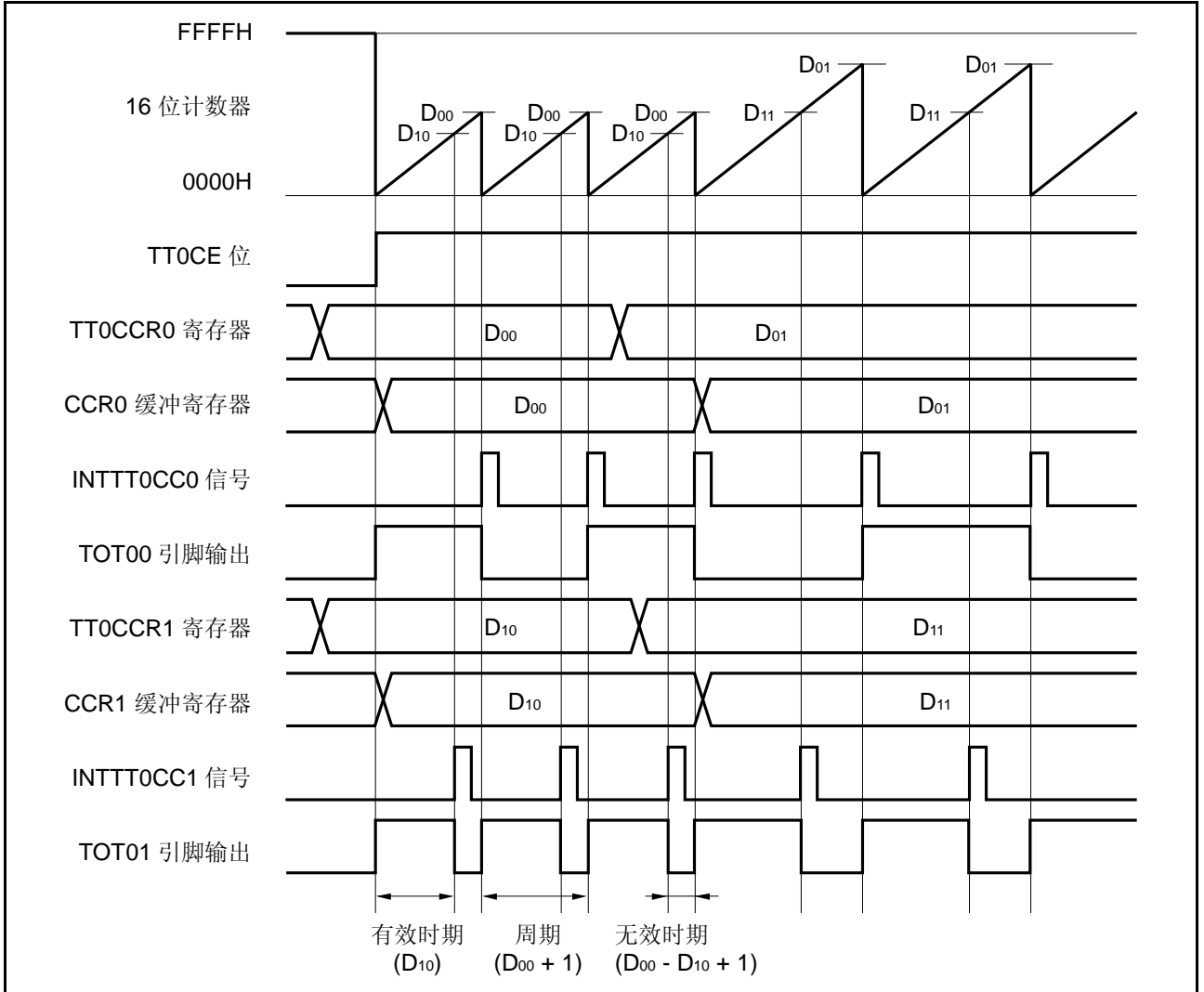


图 9-30. PWM 输出模式的基本时序



当 TT0CE 位置为 1 时，16 位计数器从 FFFFH 清零为 0000H，计数开始，同时从 TOT01 引脚输出 PWM 波形。PWM 波形的有效电平宽度、周期、占空比系数的计算如下。

有效电平宽度 = (TT0CCR1 寄存器的设定值) × 计数时钟周期

周期 = (TT0CCR0 寄存器的设定值 + 1) × 计数时钟周期

占空比系数 = (TT0CCR1 寄存器的设定值) / (TT0CCR0 寄存器的设定值 + 1)

在计数操作时，可通过重写 TT0CCRn 寄存器来改变 PWM 波形。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配时，16 位计数器清零为 0000H 时，新写入的值会发生作用。

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配之后，进行下一次计数时，会同步产生比较匹配中断请求信号 INTTT0CC0，且 16 位计数器清零为 0000H。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTT0CC1。

当 16 位计数器的计数值与 CCRn 缓冲寄存器的值匹配时，TT0CCRn 寄存器的设置值被传送到 CCRn 缓冲寄存器，且 16 位计数器被清除为 0000H。

备注 n=0, 1

图 9-31. PWM 输出模式时的寄存器设置 (1/2)

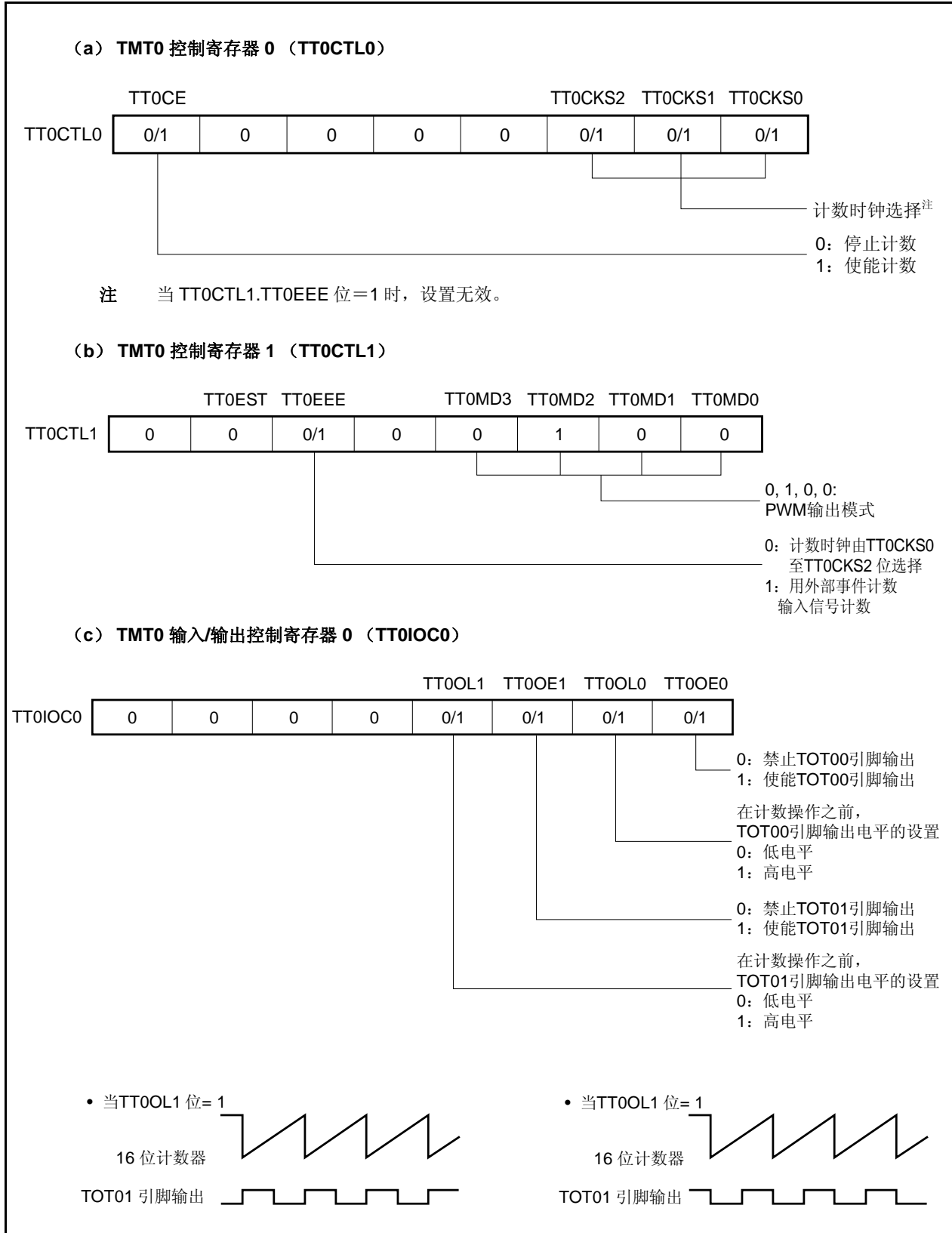


图 9-31. PWM 输出模式时的寄存器设置 (2/2)



(1) PWM 输出模式下的操作流程

图 9-32. PWM 输出模式的软件处理流程 (1/2)

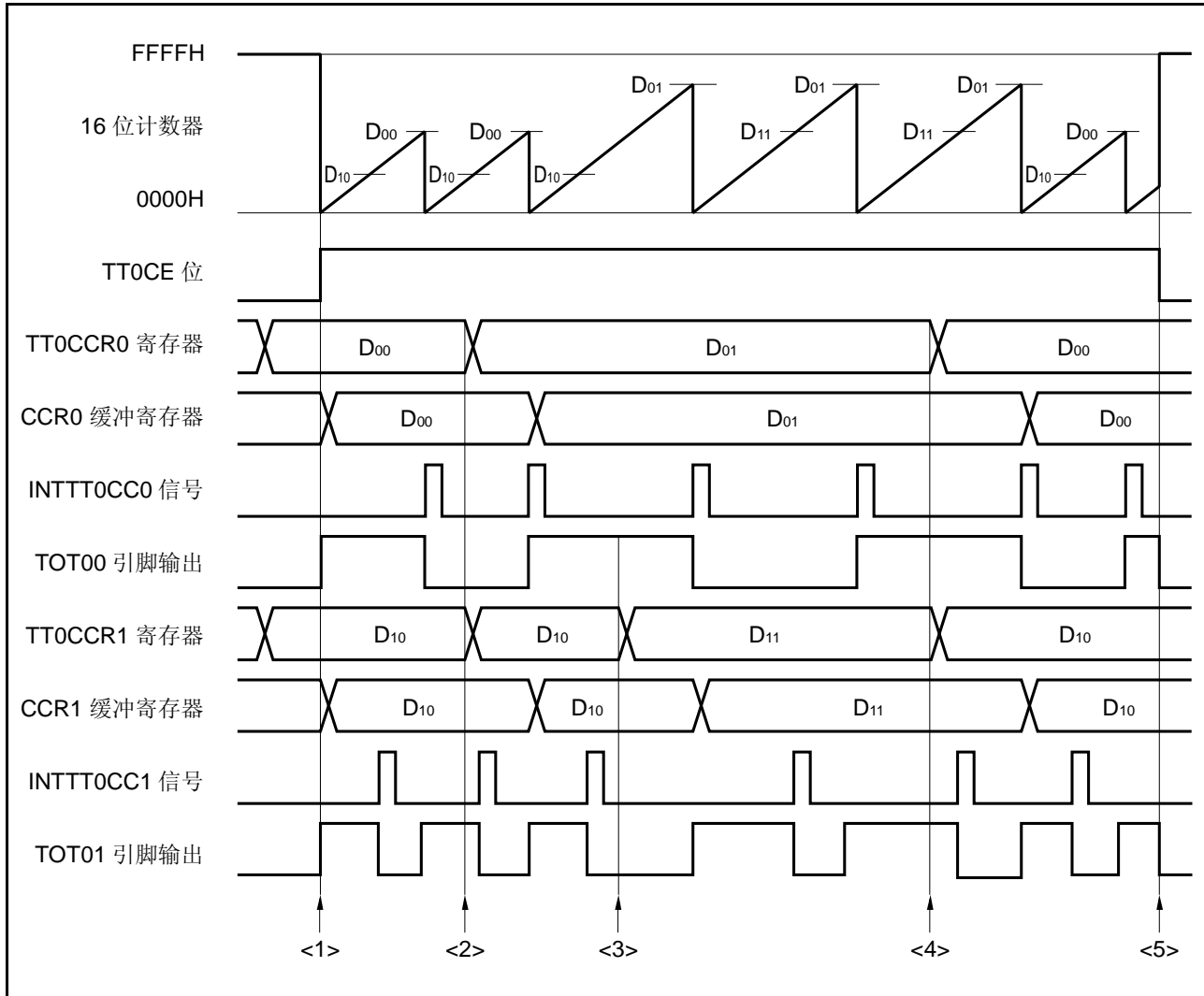
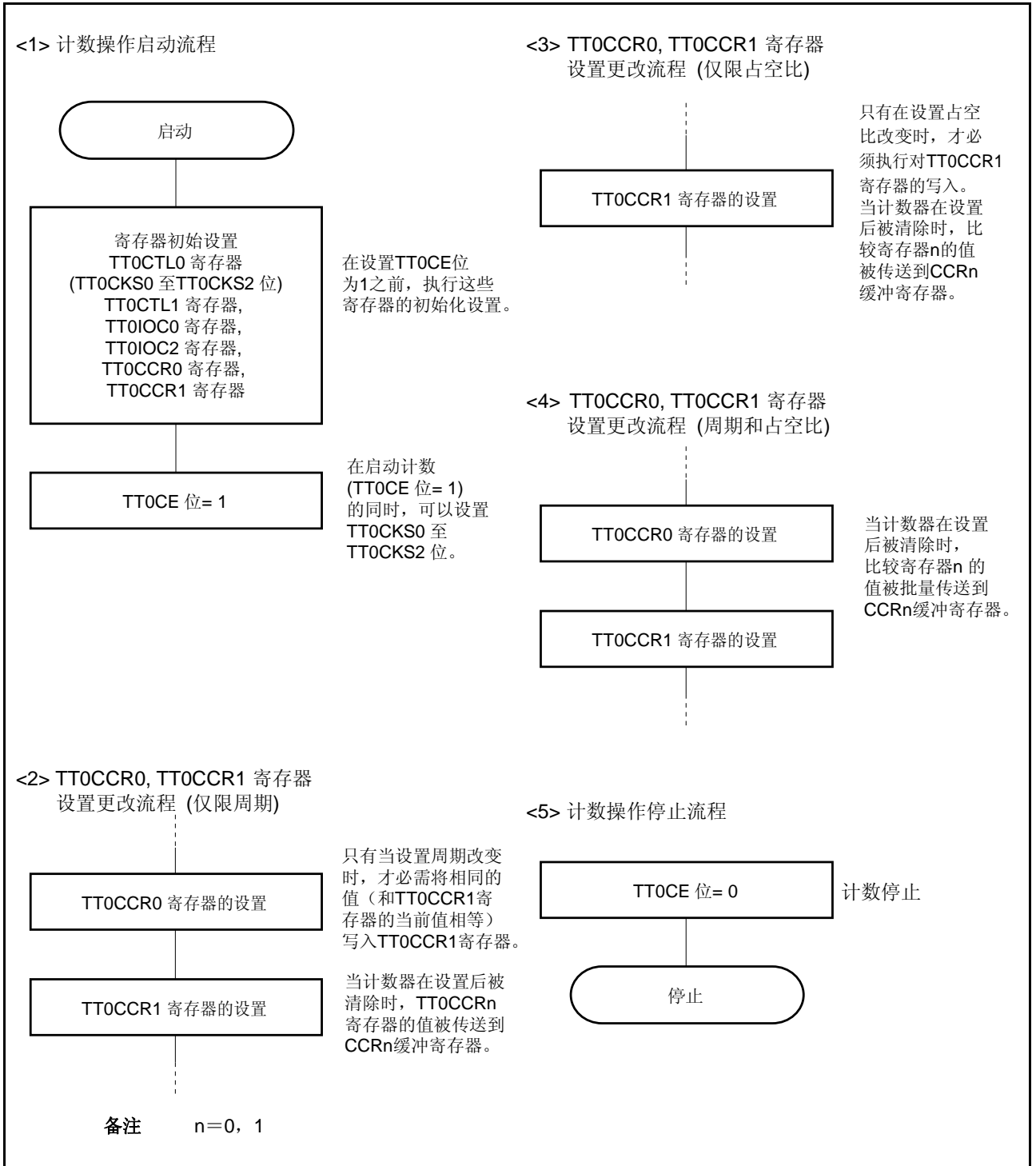


图 9-32. PWM 输出模式时的软件处理流程 (2/2)

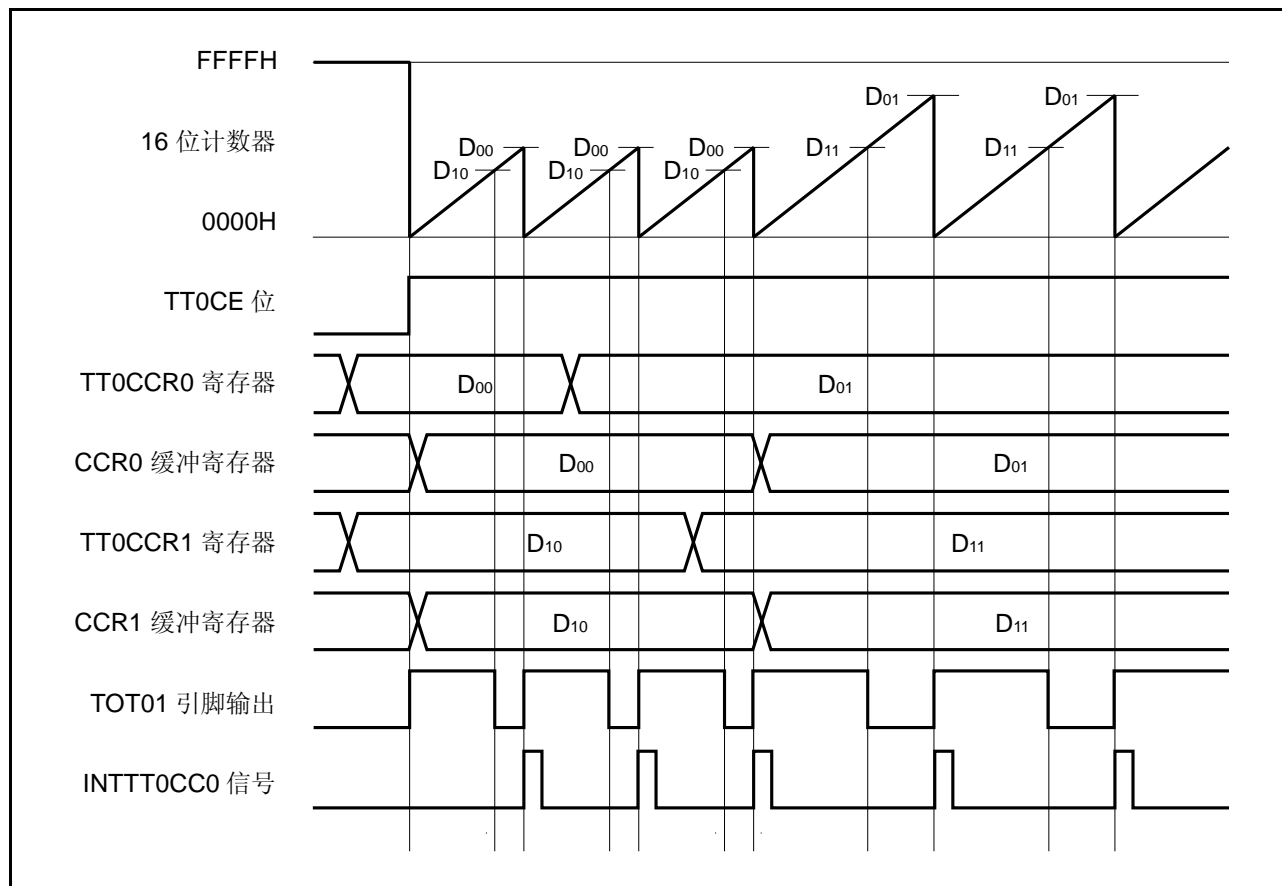


(2) PWM 输出模式的操作时序

(a) 操作过程中改变脉冲宽度

计数器操作过程中，若要改变 PWM 波形，应重新写入 TT0CCR1 寄存器。

当检测到 INTTT0CC1 信号以后，在写入 TT0CCR1 寄存器之后重写 TT0CCR1 寄存器。



为了将数据从 TT0CCRn 寄存器传送到 CCRn 缓冲寄存器，必须对 TT0CCR1 寄存器进行写入操作。

此时如果 PWM 波形的周期和有效电平都要改变，应先将周期设定到 TT0CCR0 寄存器，再将有效电平宽度设定到 TT0CCR1 寄存器。

如果只改变 PWM 波形的周期，应先将周期设置到 TT0CCR0 寄存器，然后将相同的值（和 TT0CCR1 寄存器当前值相等）写入 TT0CCR1 寄存器。

如果只改变 PWM 波形的有效电平宽度（占空比系数），仅需要设置 TT0CCR1 寄存器。

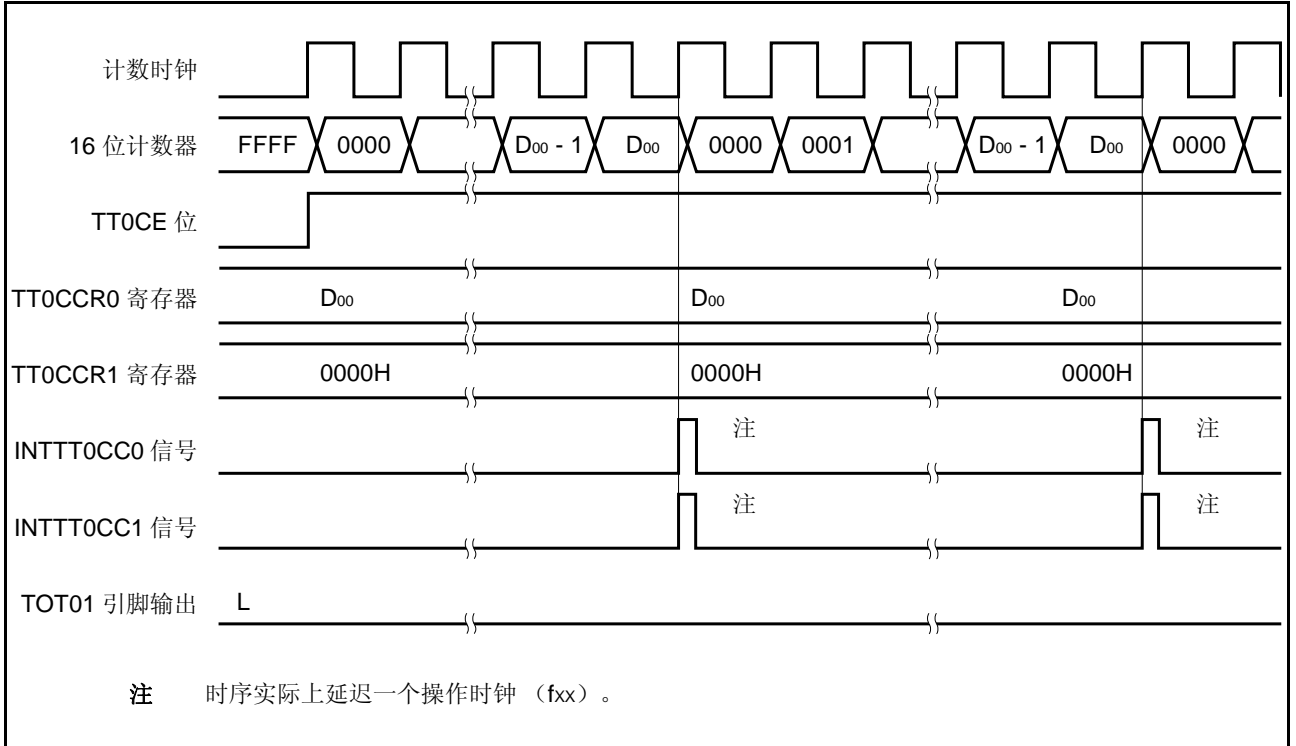
当数据写入 TT0CCR1 寄存器之后，在对 16 位计数器清零的同时，写入到 TT0CCRn 寄存器的值会被传送到 CCRn 缓冲寄存器，并作为与 16 位计数器进行比较的值。

在写入 TT0CCR1 寄存器一次之后，要再次写入 TT0CCR0 或 TT0CCR1 寄存器，应该在 INTTT0CC0 信号产生之后进行。否则，CCRn 缓冲寄存器的值可能不确定，因为把数据从 TT0CCRn 寄存器传送到 CCRn 缓冲寄存器的时序会和写入 TT0CCRn 寄存器的时序发生冲突。

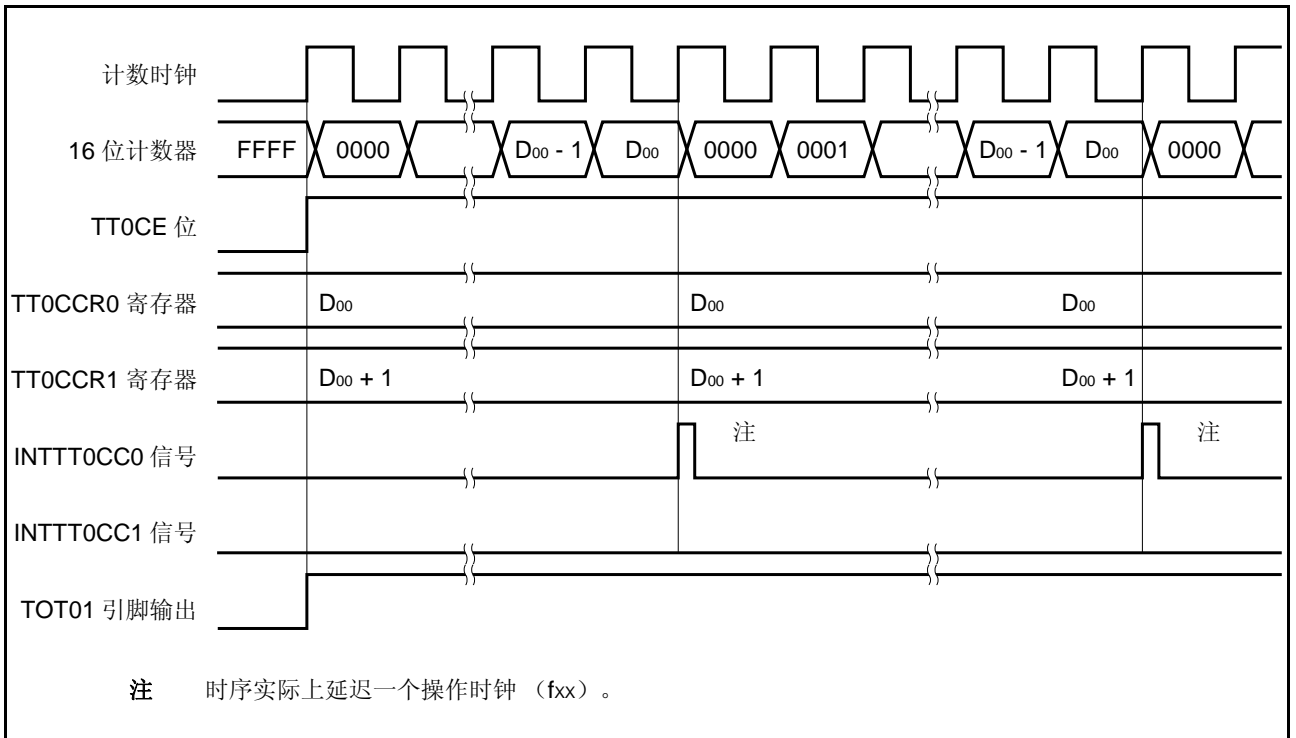
备注 n=0, 1

(b) PWM 波形的 0%和 100% 输出

为了输出 0%的波形，需要将 TT0CCR1 寄存器设置为 0000H。16 位计数器的计数值和 CCR0 缓冲寄存器匹配之后，16 位计数器被清除为 0000H，且产生 INTTT0CC0 和 INTTT0CC1 中断请求信号。

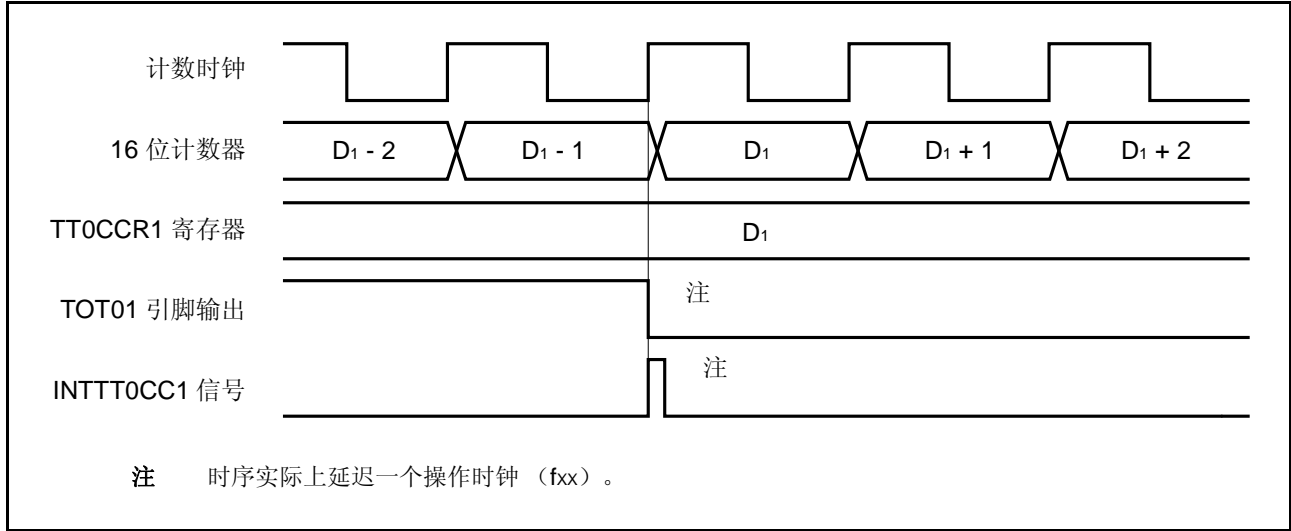


为了输出 100%的波形，需要将 TT0CCR1 寄存器的值设为：(TT0CCR0 寄存器的设定值 +1)。如果 TT0CCR0 寄存器的设定值为 FFFFH，则不会产生 100%波形输出。



(c) 比较匹配中断请求信号 (INTTT0CC1) 的发生时序

在 PWM 输出模式下，比较匹配中断请求信号 (INTTT0CC1) 的发生时序与其它情况下 INTTT0CC1 信号的发生时序有所不同；当 16 位计数器的计数值与 TT0CCR1 寄存器的值匹配时，产生 INTTT0CC1 信号。



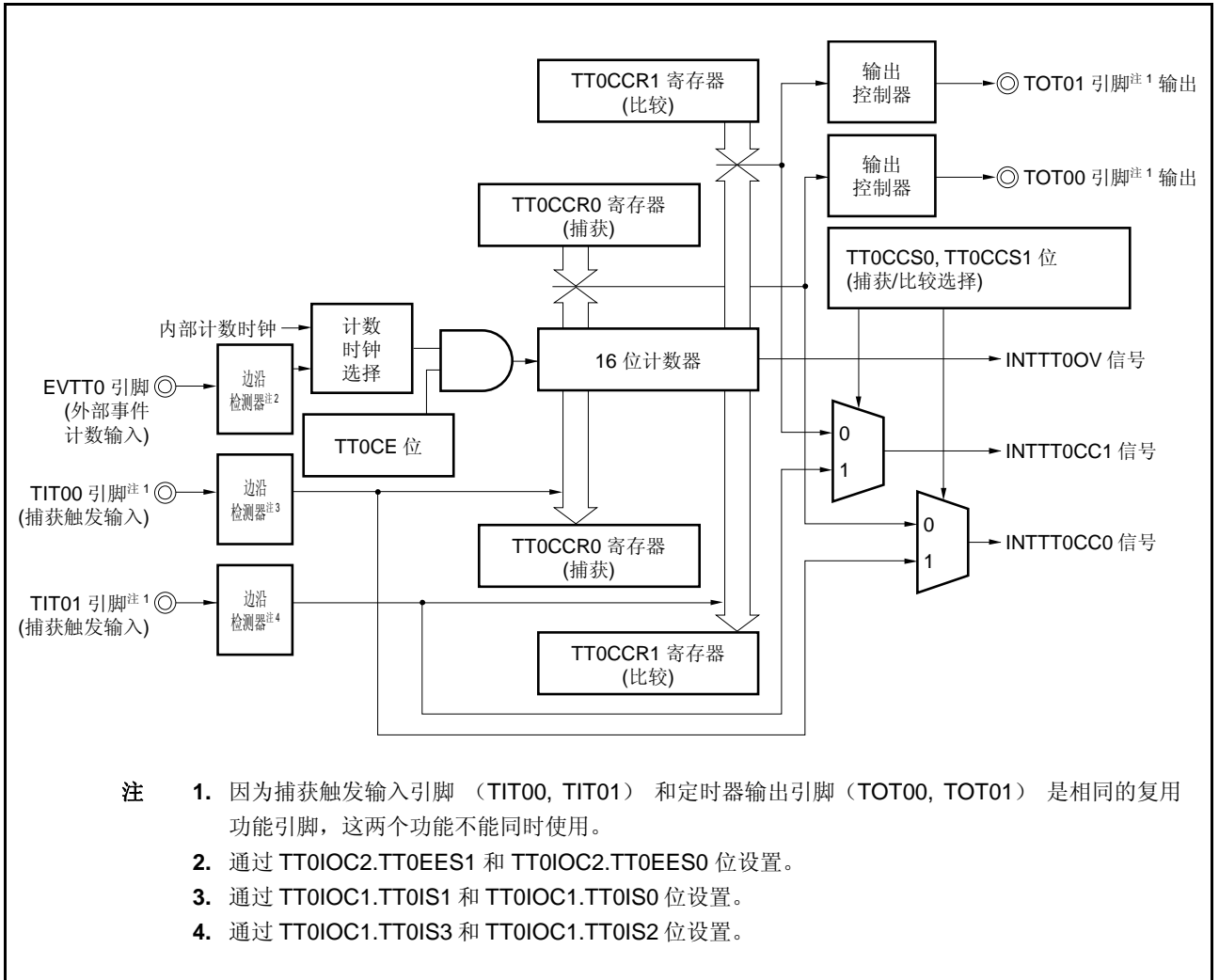
通常在 16 位计数器的计数值与 TT0CCR1 寄存器的值匹配之后，在下次计数时同步发生 INTTT0CC1 信号。

但是，在 PWM 输出模式下，INTTT0CC1 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOT01 引脚输出信号变化的时序。

9.6.6 自由运行定时器模式 (TT0MD3 至 TT0MD0 位=0101)

在自由运行定时器模式下，当 TT0CTL0.TT0CE 位置为 1 时，16 位定时器/事件计数器 T 开始计数。此时，根据 TT0OPT0.TT0CCS0 和 TT0OPT0.TT0CCS1 位的设置，TT0CCR0 和 TT0CCR1 寄存器可以用作比较寄存器或捕获寄存器。

图 9-33. 自由运行定时器模式的配置



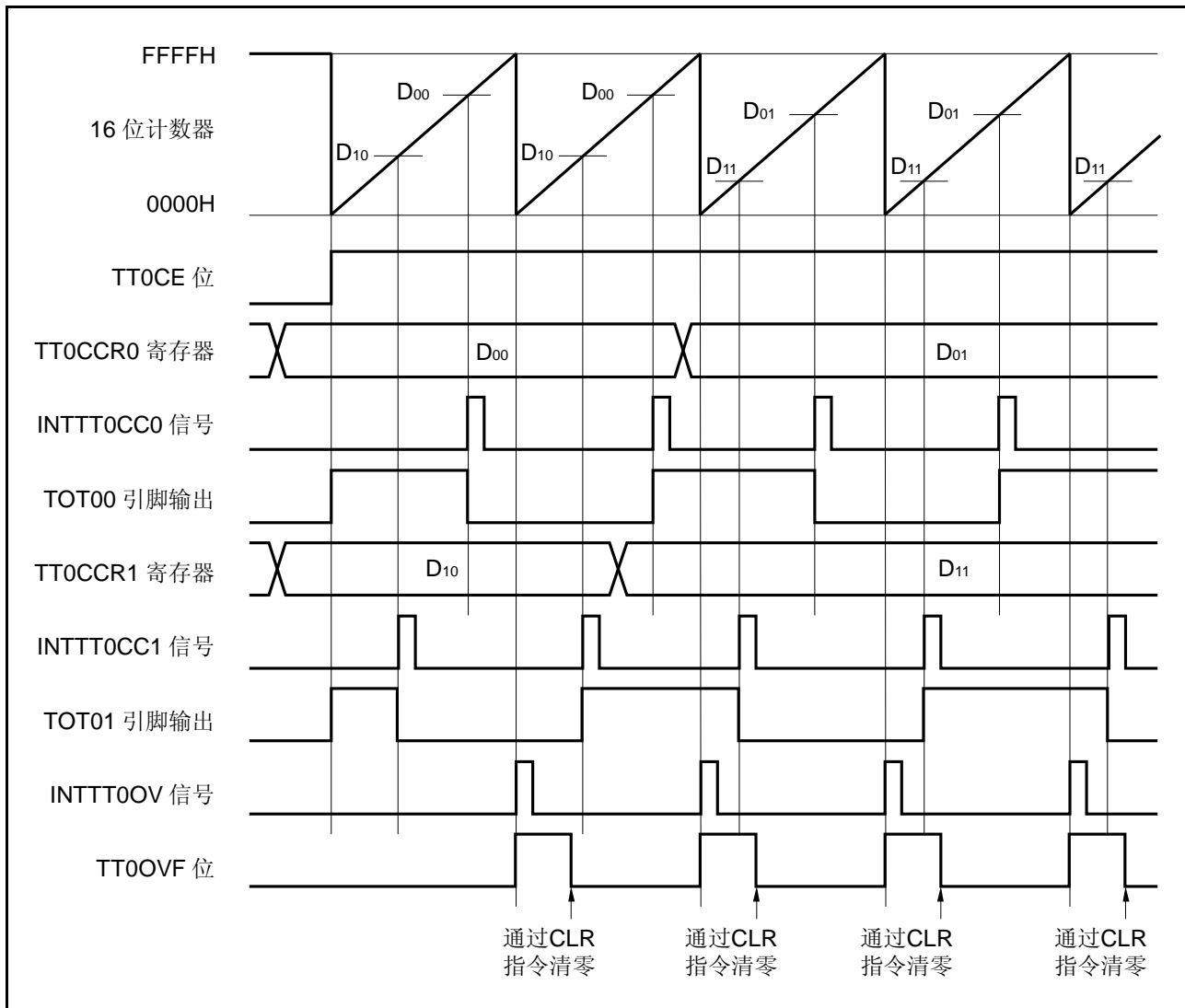
• 比较操作

当 TT0CE 位被置为 1 时，16 位定时器/事件计数器 T 开始计数，TOT0n 引脚的输出信号反相。此后当 16 位计数器的计数值与 TT0CCRn 寄存器的设定值匹配时，产生一个比较匹配中断请求信号 (INTTT0CCn)，而且 TOT0n 引脚的输出信号反相。

16 位计数器继续对计数时钟同步计数。当它计数到 FFFFH 时，会在下一个计数时钟时产生一个溢出中断请求信号 (INTTT0OV)，同时计数器清零为 0000H，然后继续进行计数。此时，溢出标志 (TT0OPT0.TT0OVF 位) 被置为 1，通过软件确认溢出标志为 1 然后运行 CLR 指令清除为 0。

当计数器工作时，可以重写 TT0CCRn 寄存器。如果进行重写，则新写入的值会在“任意时刻写入”时反映出来，并与计数值进行比较。

图 9-34. 自由运行定时器模式下的基本时序 (比较功能)



- 捕获操作

当 TT0CE 位被置为 1 时，16 位计数器开始计数。当检测到 TIT0n 引脚输入信号的有效边沿时，16 位计数器的计数值存储在 TT0CCRn 寄存器中，并产生一个捕获中断请求信号 (INTTT0CCn)。

16 位计数器继续对计数时钟同步计数。当它计数到 FFFFH 时，会在下一个计数时钟时产生一个溢出中断请求信号 (INTTT0OV)，同时计数器清零为 0000H，然后继续进行计数。此时，溢出标志 (TT0OVF 位) 被置为 1，通过软件确认溢出标志为 1 然后运行 CLR 指令溢出标志位清 0。

图 9-35. 自由运行定时器模式下的基本时序 (捕获功能)

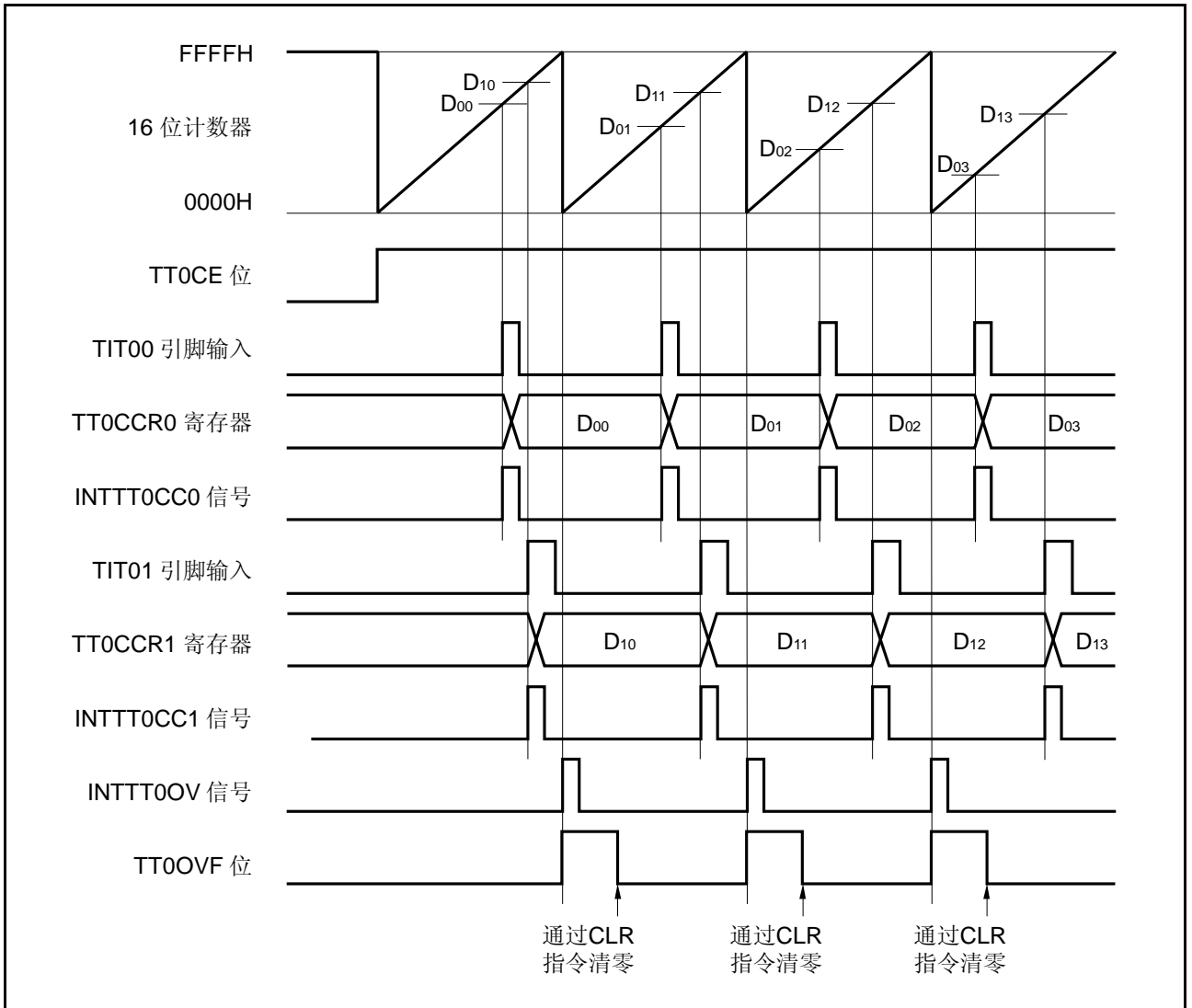


图 9-36. 自由运行定时器模式下的寄存器设置 (1/3)

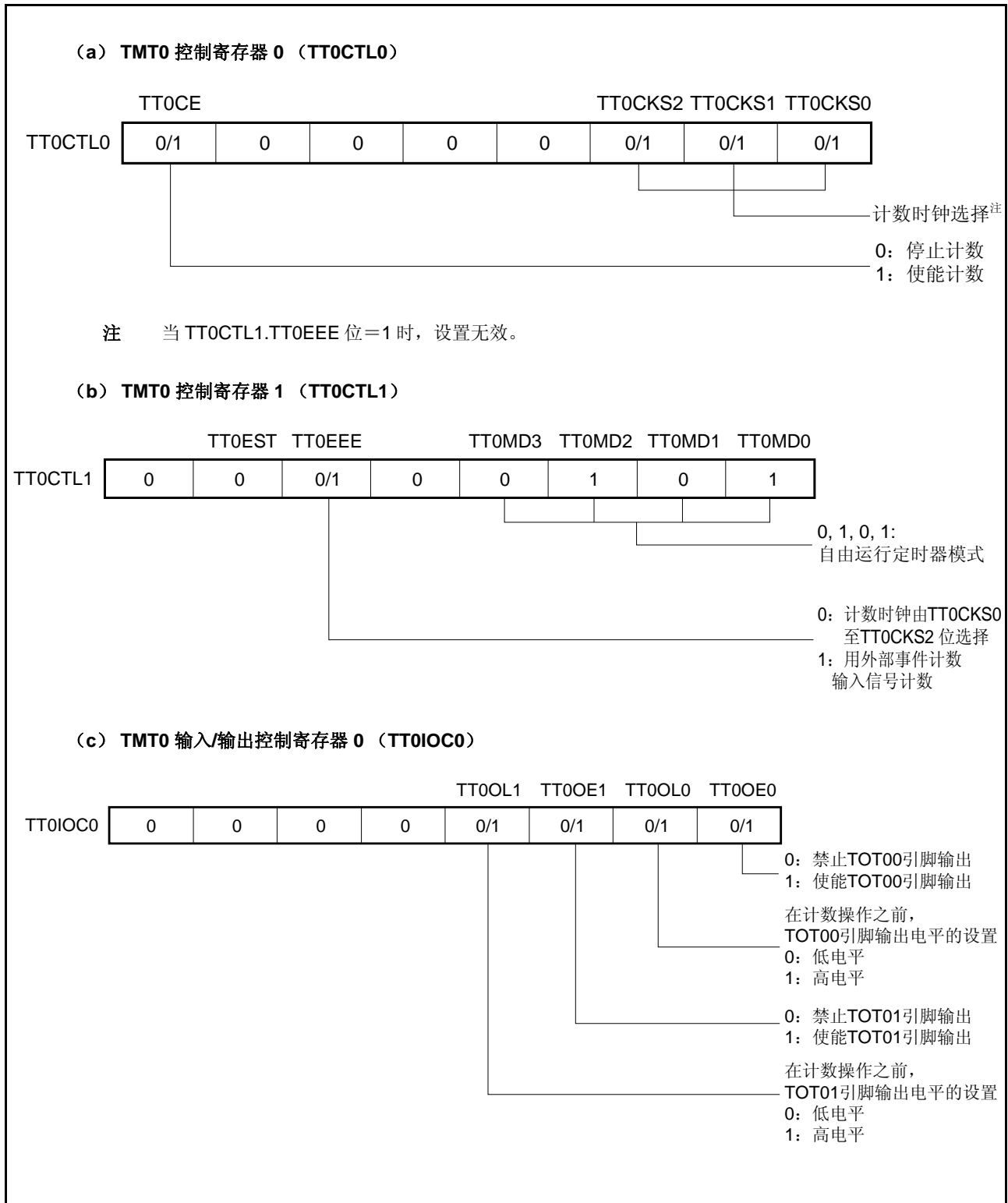


图 9-36. 自由运行定时器模式下的寄存器设置 (2/3)

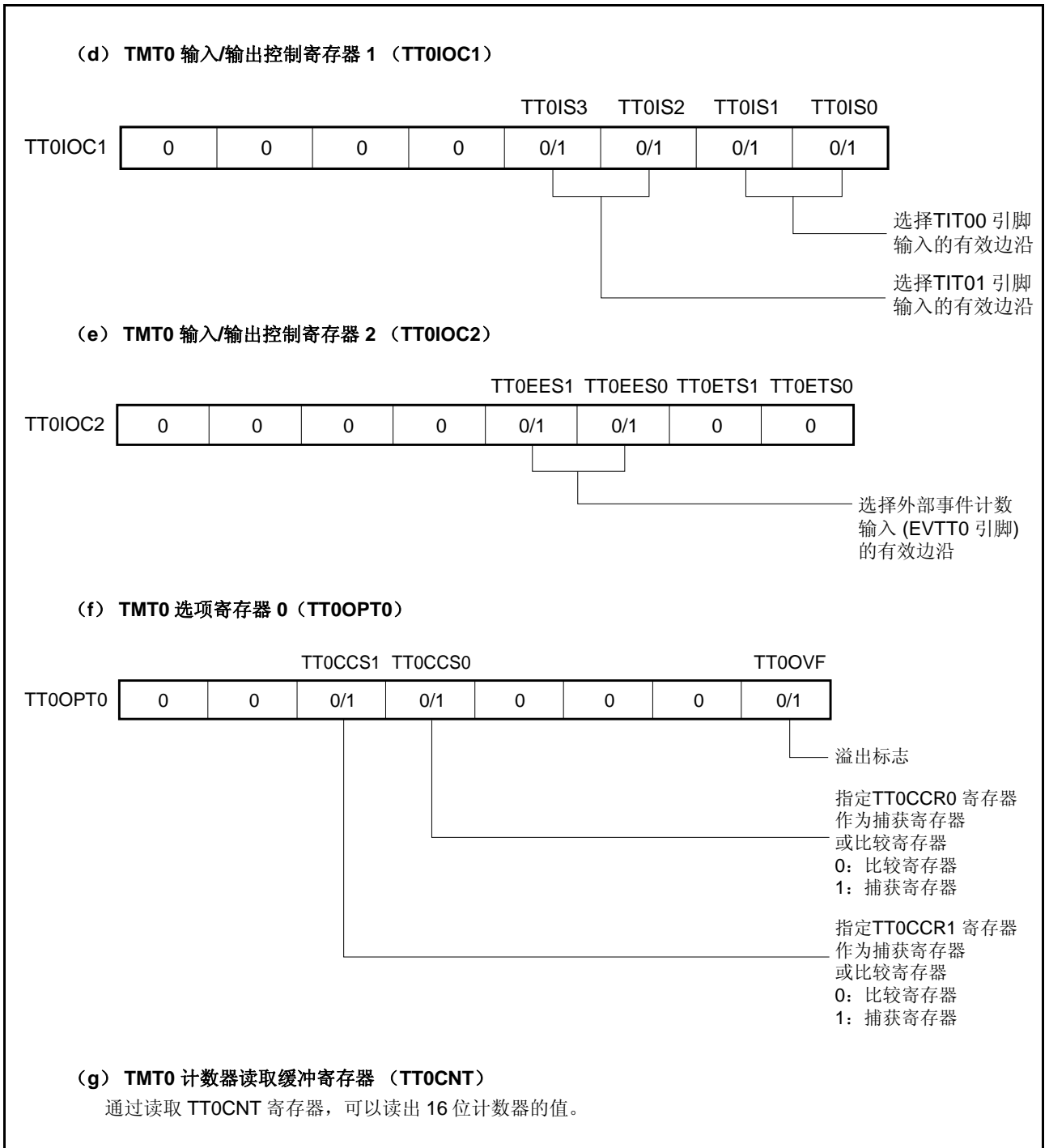


图 9-36. 自由运行定时器模式下的寄存器设置 (3/3)

(h) TMT0 捕获/比较寄存器 0 和 1 (TT0CCR0 和 TT0CCR1)

根据 TT0OPT0.TT0CCSn 位的设置，这些寄存器可以用作捕获寄存器或比较寄存器。

当用作捕获寄存器时，在检测到 TIT0n 引脚输入信号的有效边沿时，存储 16 位计数器的当前计数值。

当用作比较寄存器时，而且 TT0CCRn 寄存器的值被设置为 D_a 时，则当计数器达到 (D_a+1) 时会产生 INTTT0CCn 中断信号，同时，TOT00 和 TOT01 引脚的输出信号反相。

备注 $n=0, 1$

(1) 自由运行定时器模式时的操作流程

(a) 当捕获/比较寄存器用作比较寄存器时

图 9-37. 自由运行定时器模式的软件处理流程（比较功能）（1/2）

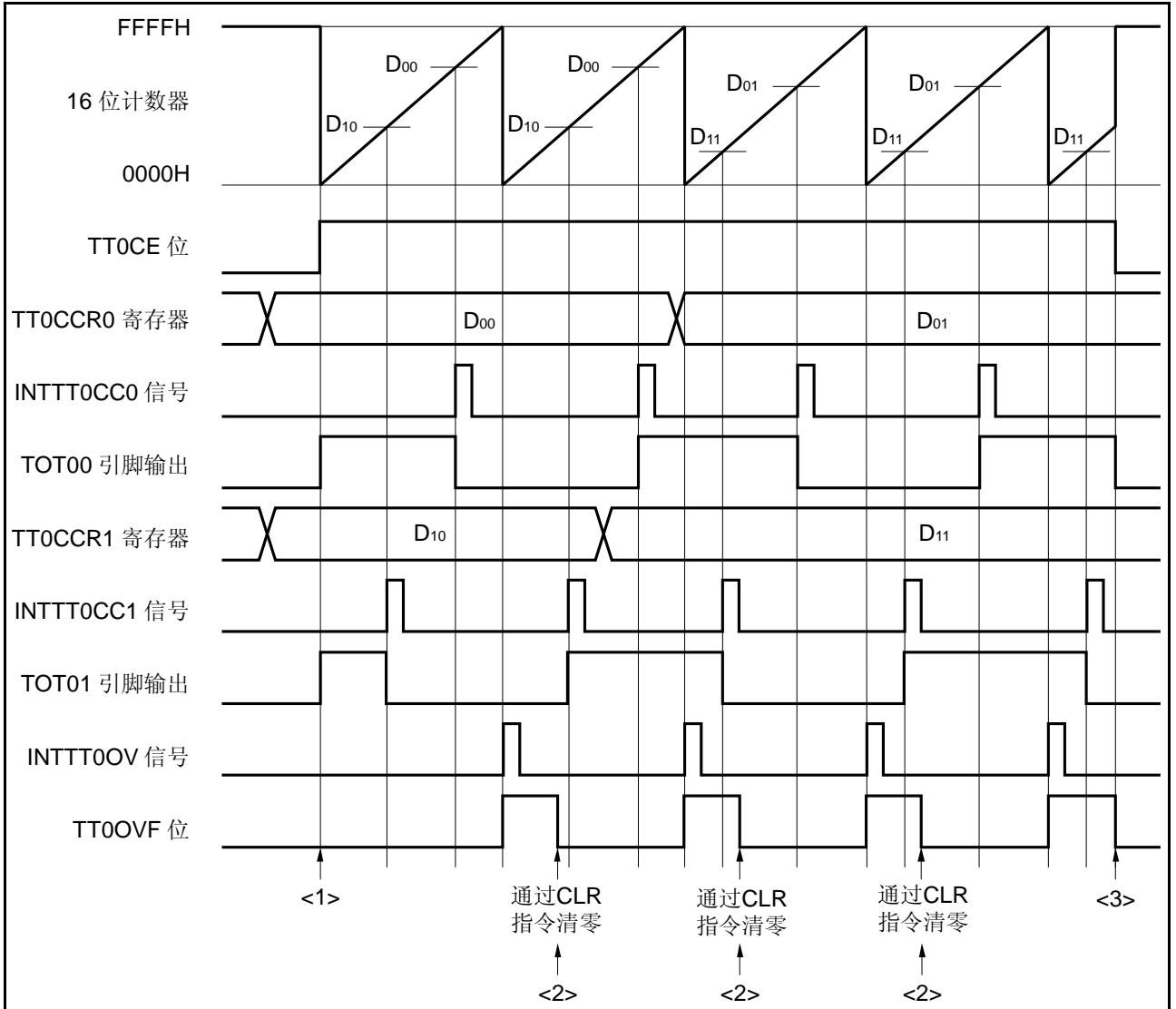
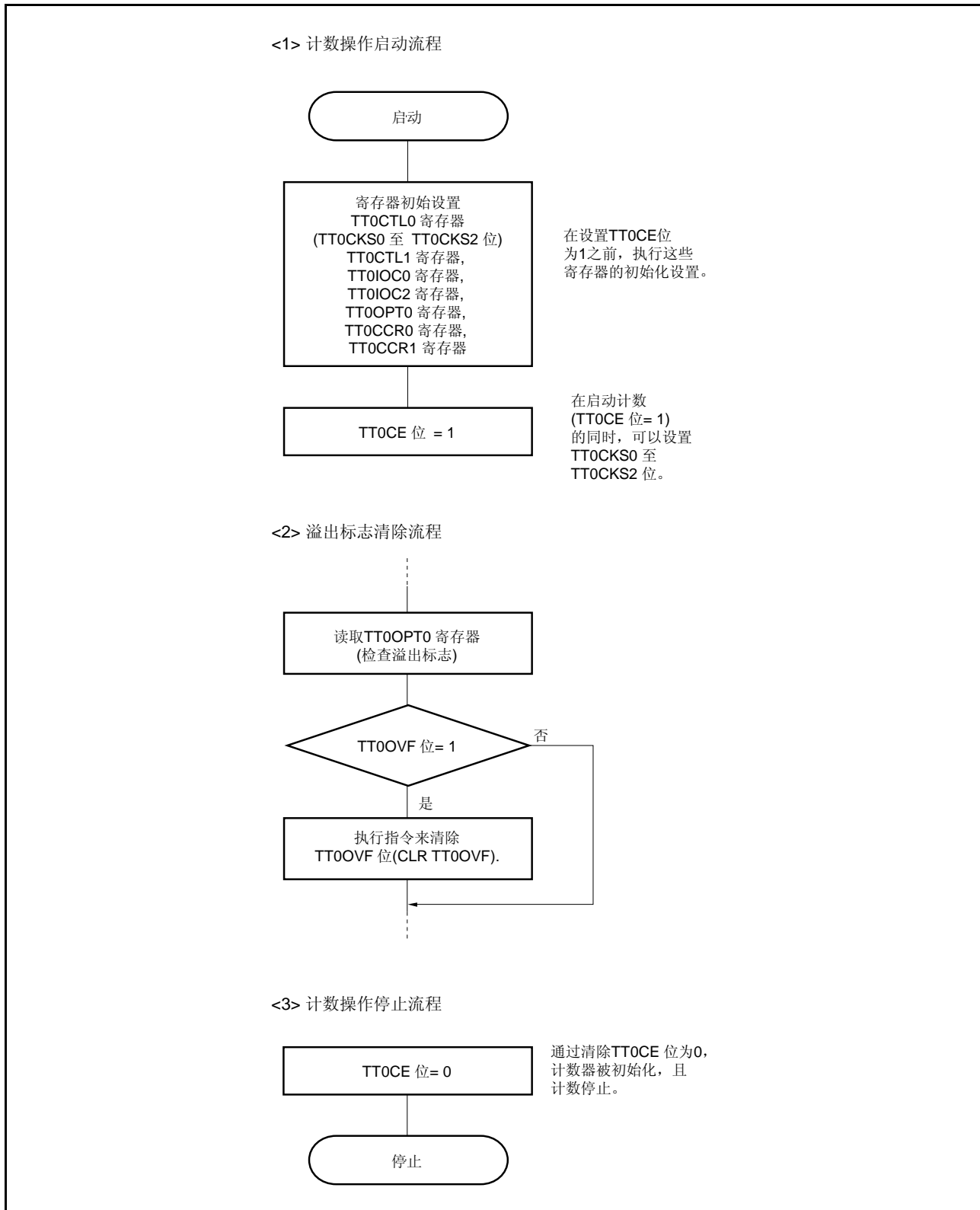


图 9-37. 自由运行定时器模式的软件处理流程（比较功能）（2/2）



(b) 当捕获/比较寄存器用作捕获寄存器时

图 9-38. 自由运行定时器模式的软件处理流程（捕获功能）（1/2）

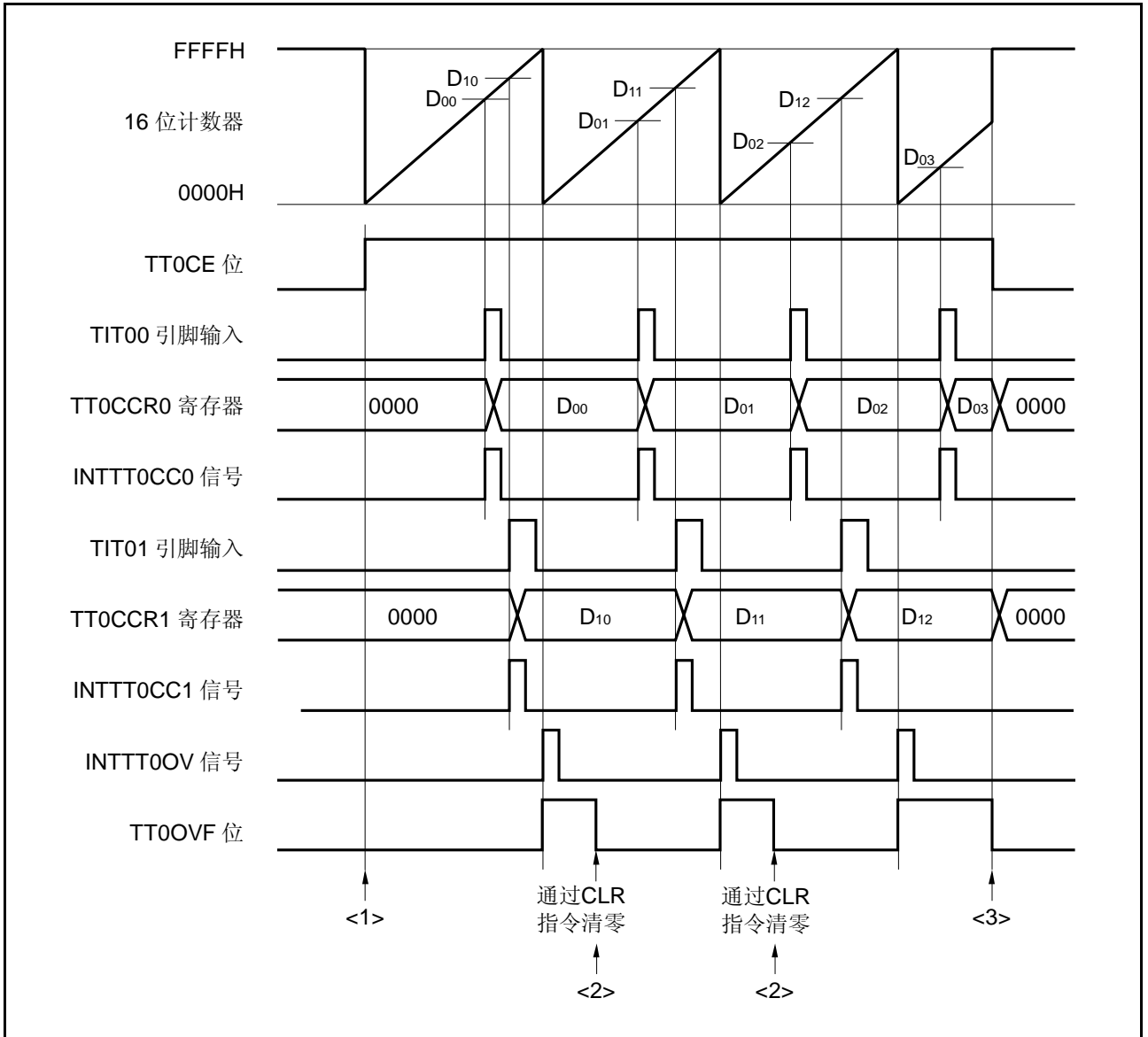
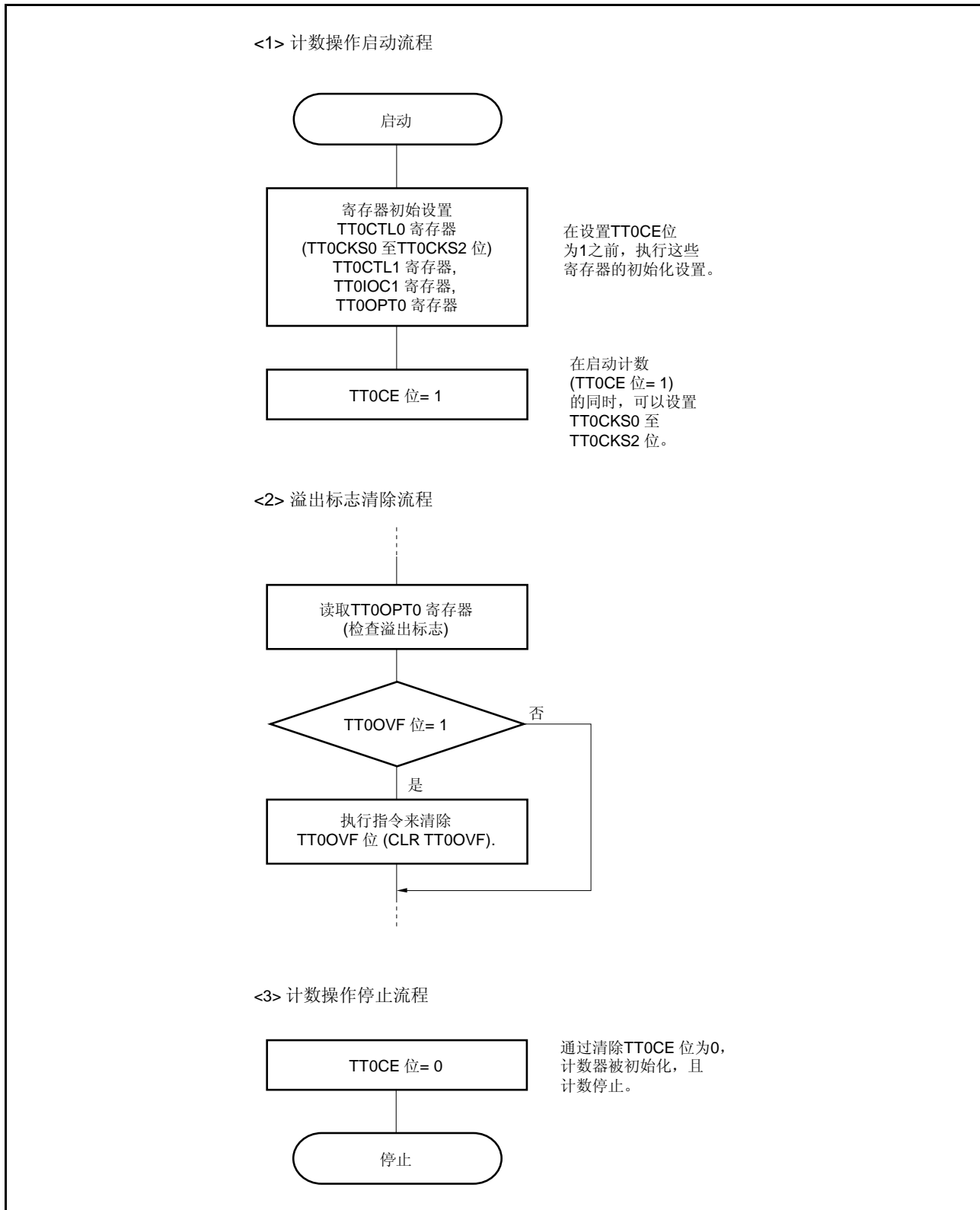


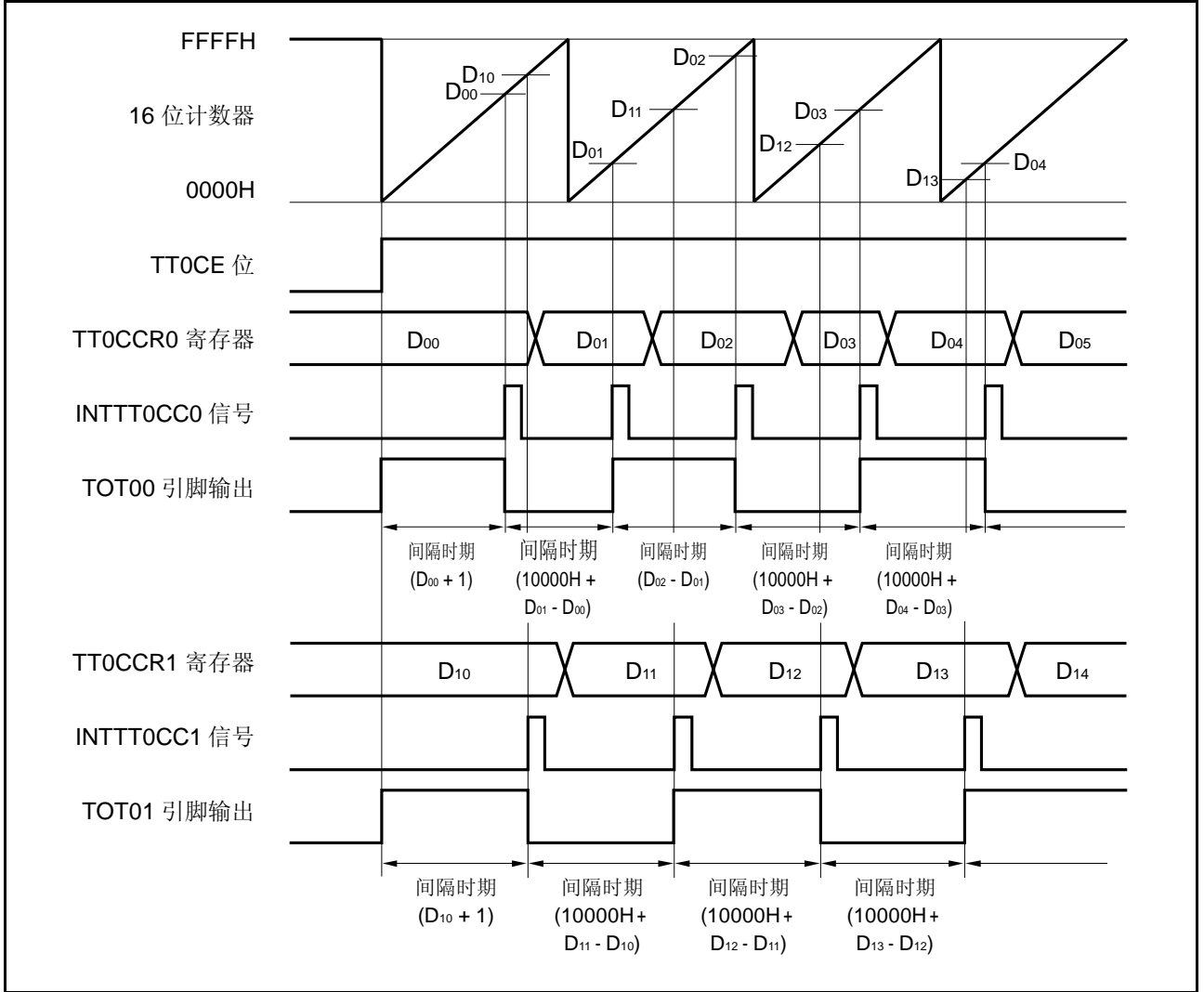
图 9-38. 自由运行定时器模式时的软件处理流程（捕获功能）（2/2）



(2) 自由运行定时器模式时的操作时序

(a) 比较寄存器的间隔定时操作

当 16 位定时器/事件计数器 T 用作间隔定时器，TT0CCRn 寄存器用作比较寄存器时，在每次检测到 INTTT0CCn 信号后，都需要通过软件设置一个比较值以产生下一个中断请求信号。



在自由运行定时器模式下进行时间间隔操作时，一个通道可以设置两个时间间隔。

要进行时间间隔操作，当检测到 INTTT0CCn 信号后，必须在中断服务程序中重置相应的 TT0CCRn 寄存器的值。

重置 TT0CCRn 寄存器的设置值可由下式计算，其中“ D_a ”是间隔时间：

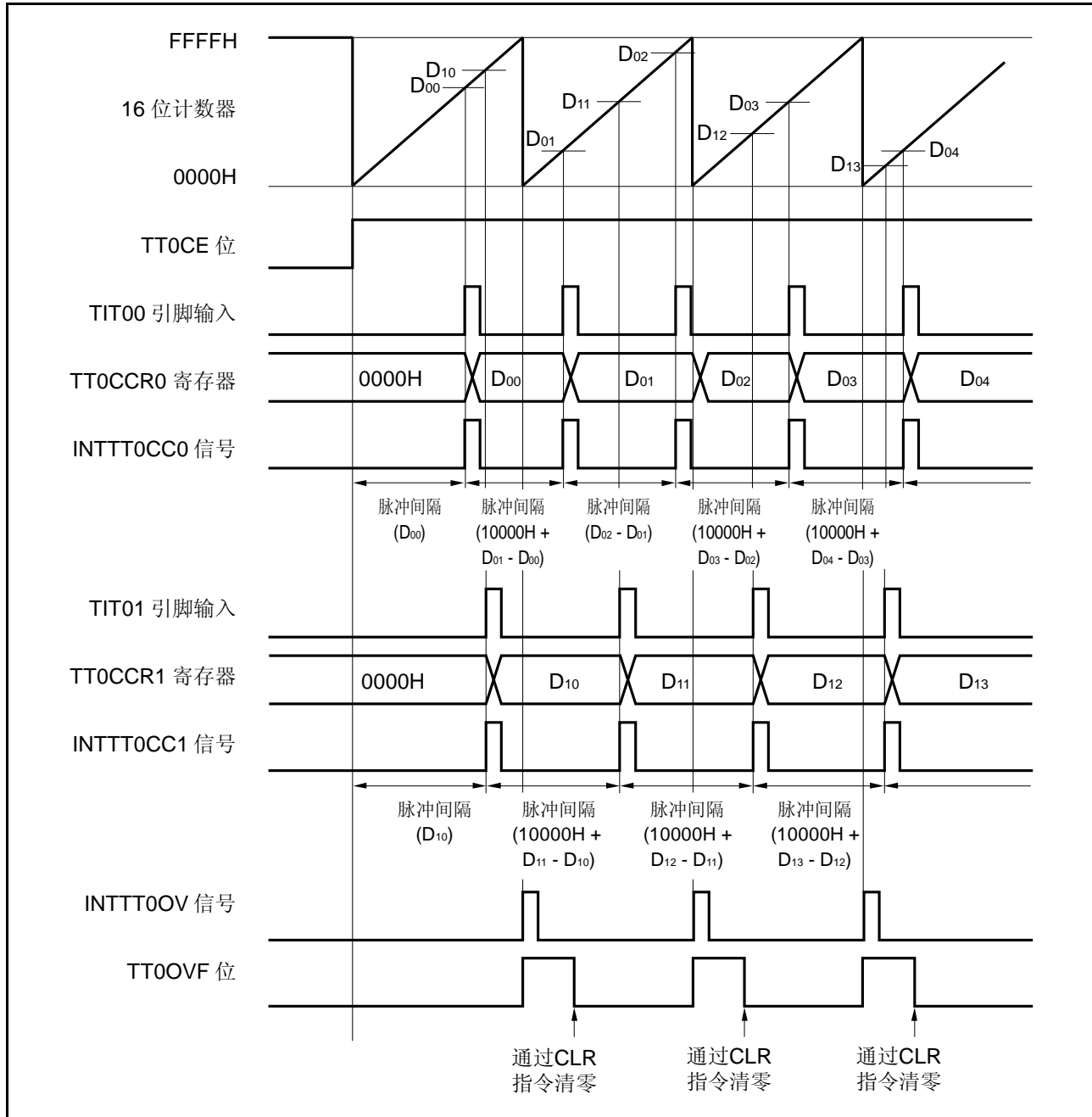
比较寄存器默认值： $D_a - 1$

比较寄存器的第二次及其以后的设置值：上次设置值 + D_a

(若计算结果大于 FFFFH，则从结果中减去 10000H，并将此值设置给寄存器。)

(b) 使用捕获寄存器进行脉冲宽度测量

在用 TT0CCRn 寄存器作为捕获寄存器进行脉宽测量时，每次检测到 INTTT0CCn 信号时，都需要通过软件读取捕获寄存器并计算间隔时间。

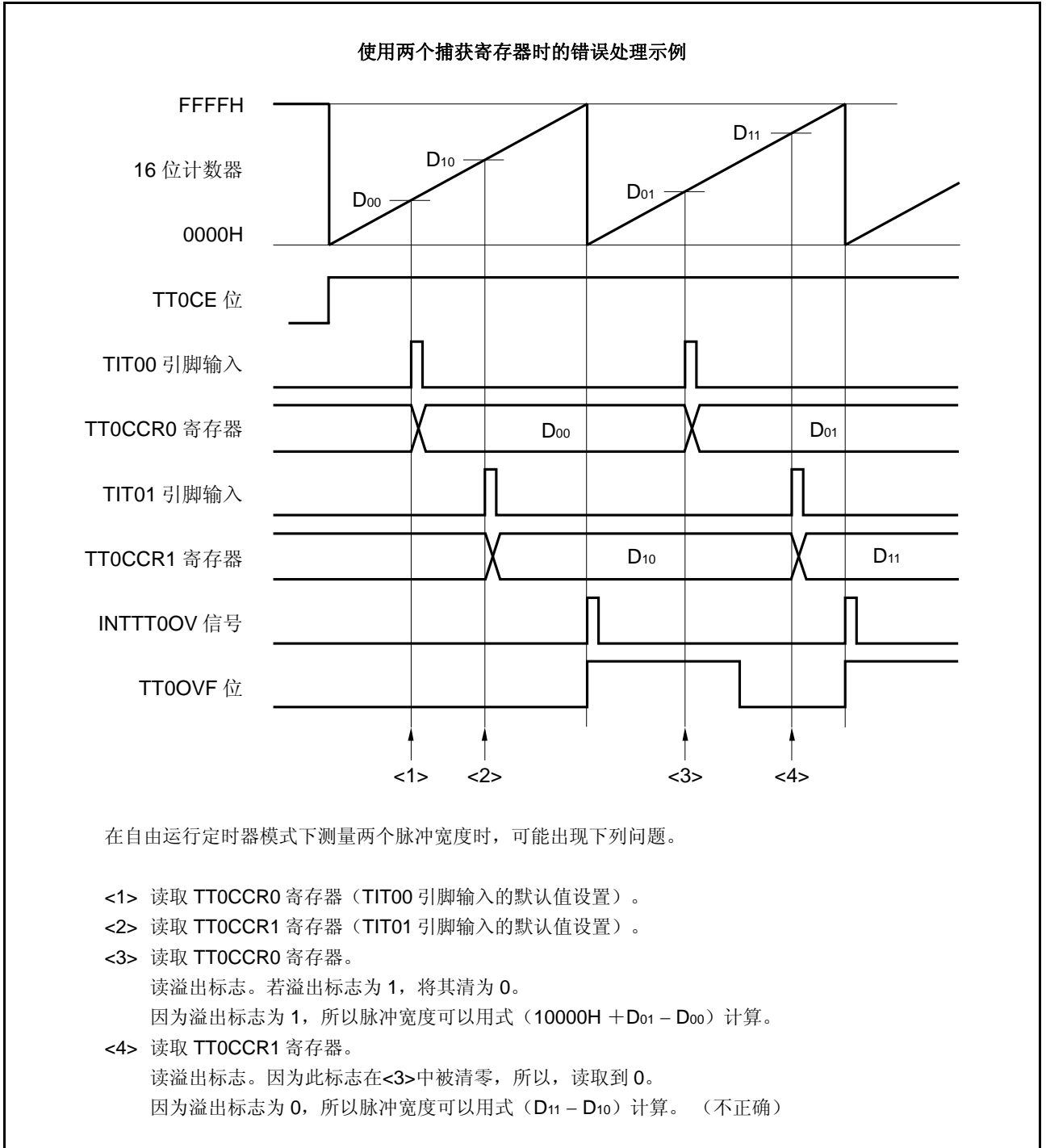


在自由运行定时器模式下执行脉宽测量时，可以用一个通道测量两个脉冲宽度。

要测量脉冲宽度，可以与 INTTT0CCn 信号同步读取 TT0CCRn 寄存器的值，并计算读取值与先前读取值之间的差，得到脉冲宽度。

(c) 当使用两个捕获寄存器时对溢出的处理

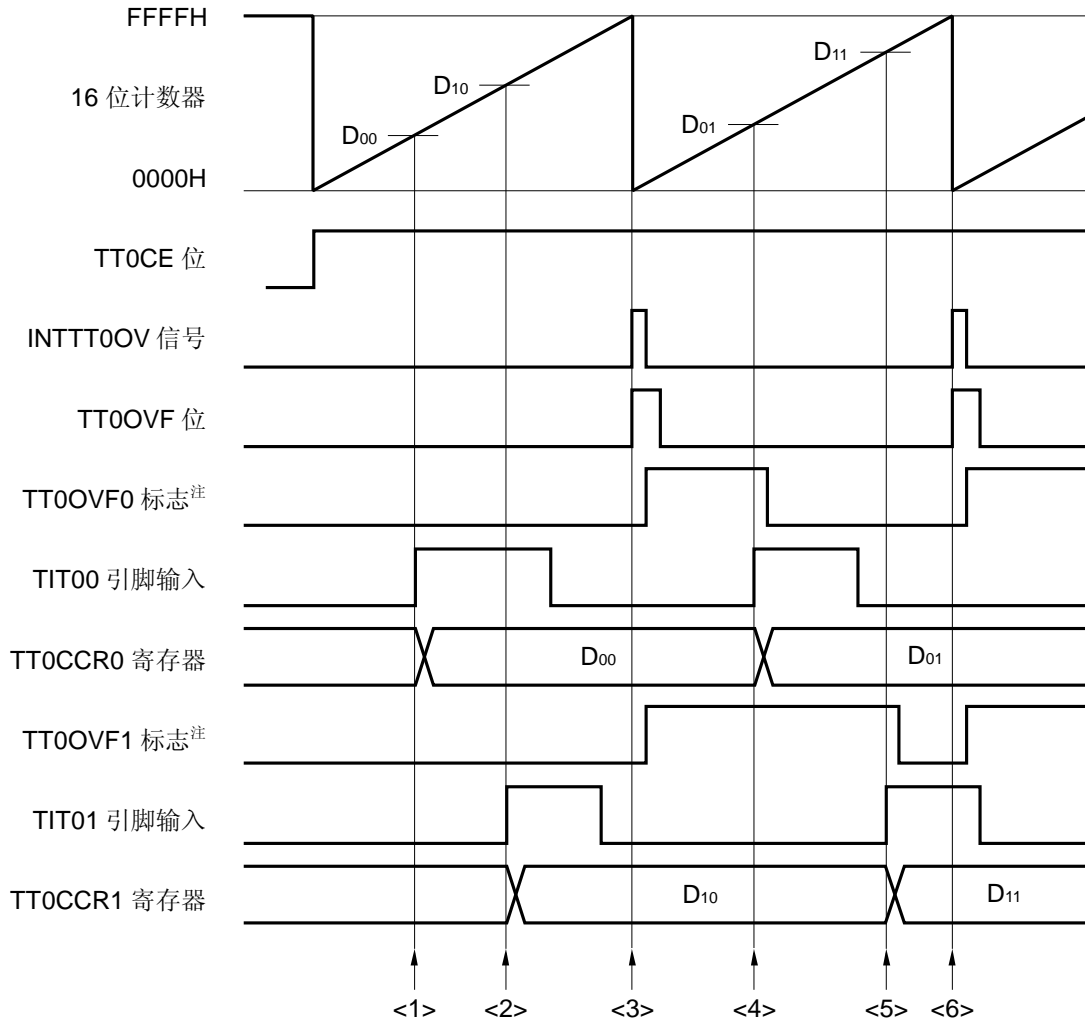
当使用两个捕获寄存器时必须格外小心。首先是一个不正确处理的示例，如下所示。



在使用两个捕获寄存器时，如果溢出标志由一个捕获寄存器清为 0，则另一个捕获寄存器可能得不到正确的脉冲宽度。

在使用两个捕获寄存器时，使用软件解决。关于如何使用软件的示例如下。

使用两个捕获寄存器时的示例（使用溢出中断）



注 由软件对内部 RAM 中的 TT0OVF0 和 TT0OVF1 标志进行设置。

<1> 读取 TT0CCR0 寄存器（TIT00 引脚输入的默认值设置）。

<2> 读取 TT0CCR1 寄存器（TIT01 引脚输入的默认值设置）。

<3> 发生溢出，在溢出中断服务程序中将 TT0OVF0 和 TT0OVF1 标志设置为 1，并将溢出标志清为 0。

<4> 读取 TT0CCR0 寄存器。

读取 TT0OVF0 标志。如果 TT0OVF0 标志为 1，则将其清为 0。

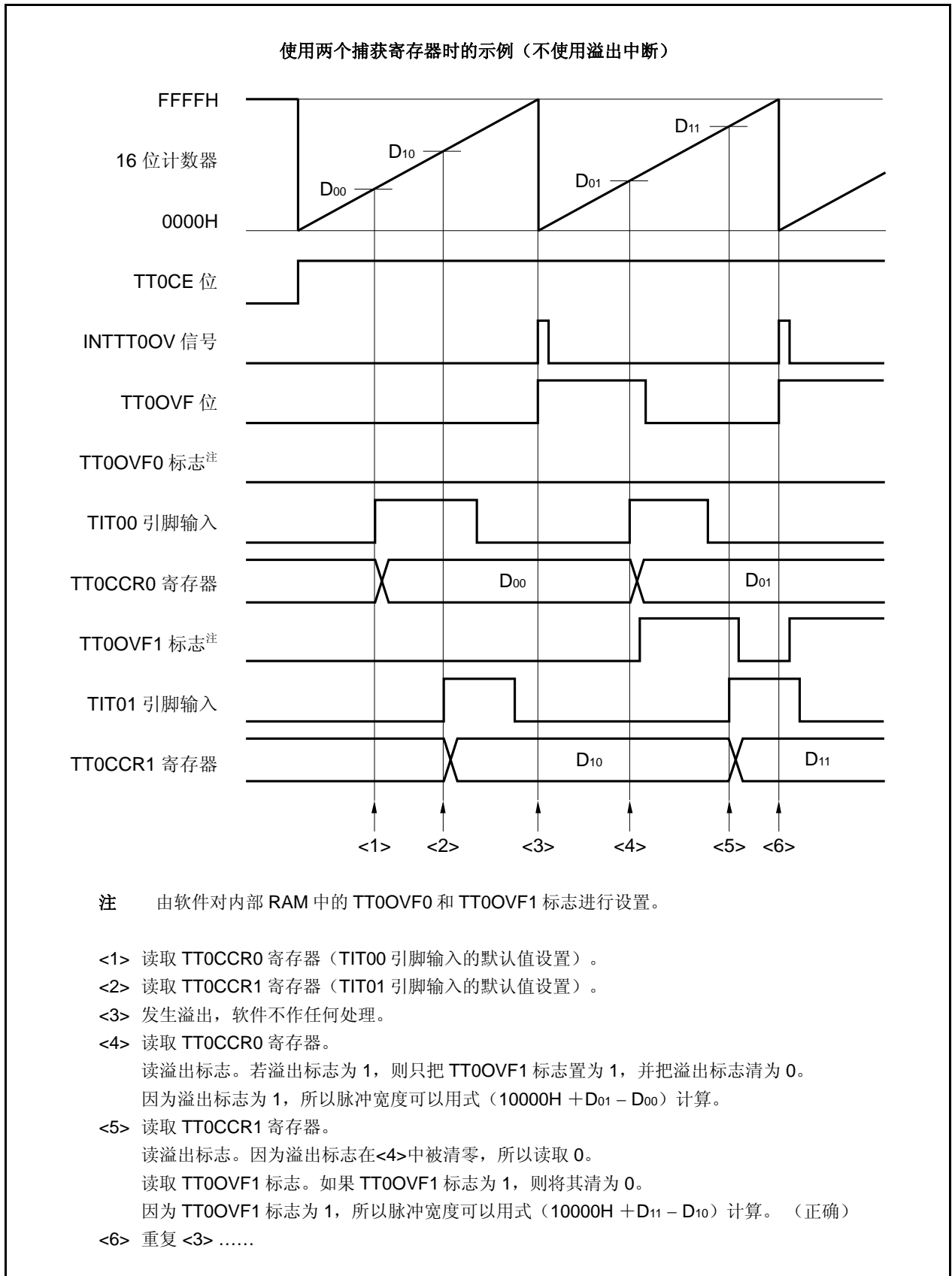
因为 TT0OVF0 标志为 1，所以脉冲宽度可以用式 $(10000H + D_{01} - D_{00})$ 计算。

<5> 读取 TT0CCR1 寄存器。

读取 TT0OVF1 标志。如果 TT0OVF1 标志为 1，则将其清为 0（TT0OVF0 标志在<4>中已清零，而 TT0OVF1 标志仍然保持为 1）。

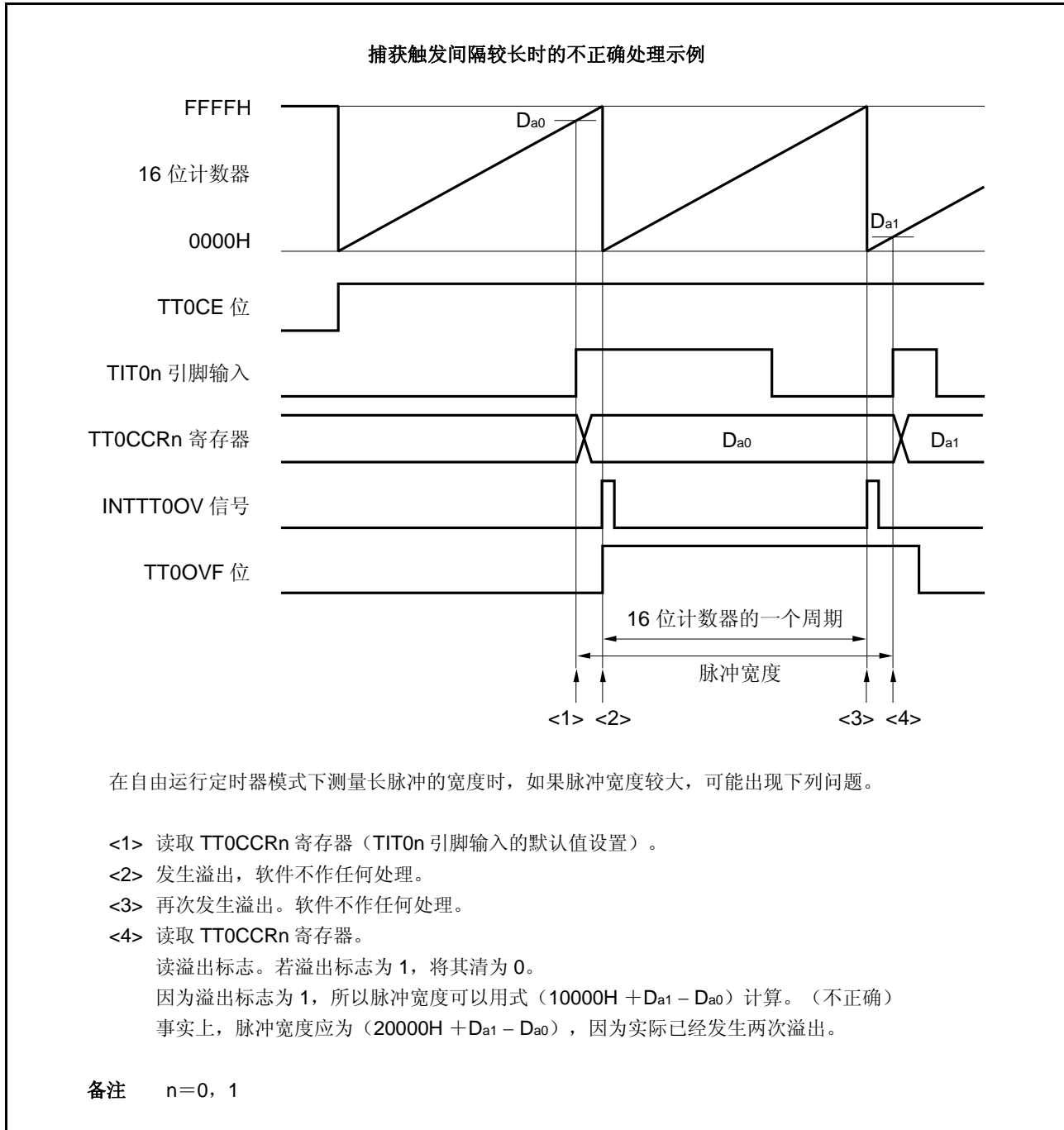
因为 TT0OVF1 标志为 1，所以脉冲宽度可以用式 $(10000H + D_{11} - D_{10})$ 计算。（正确）

<6> 重复 <3>



(d) 当捕获触发间隔较长时的溢出处理

如果脉冲宽度大于 16 位计数器的一个计数循环周期，则必须加以注意，因为从第一个捕获触发信号到下一个触发信号之间可能发生多次溢出。首先是一个不正确处理的示例，如下所示。

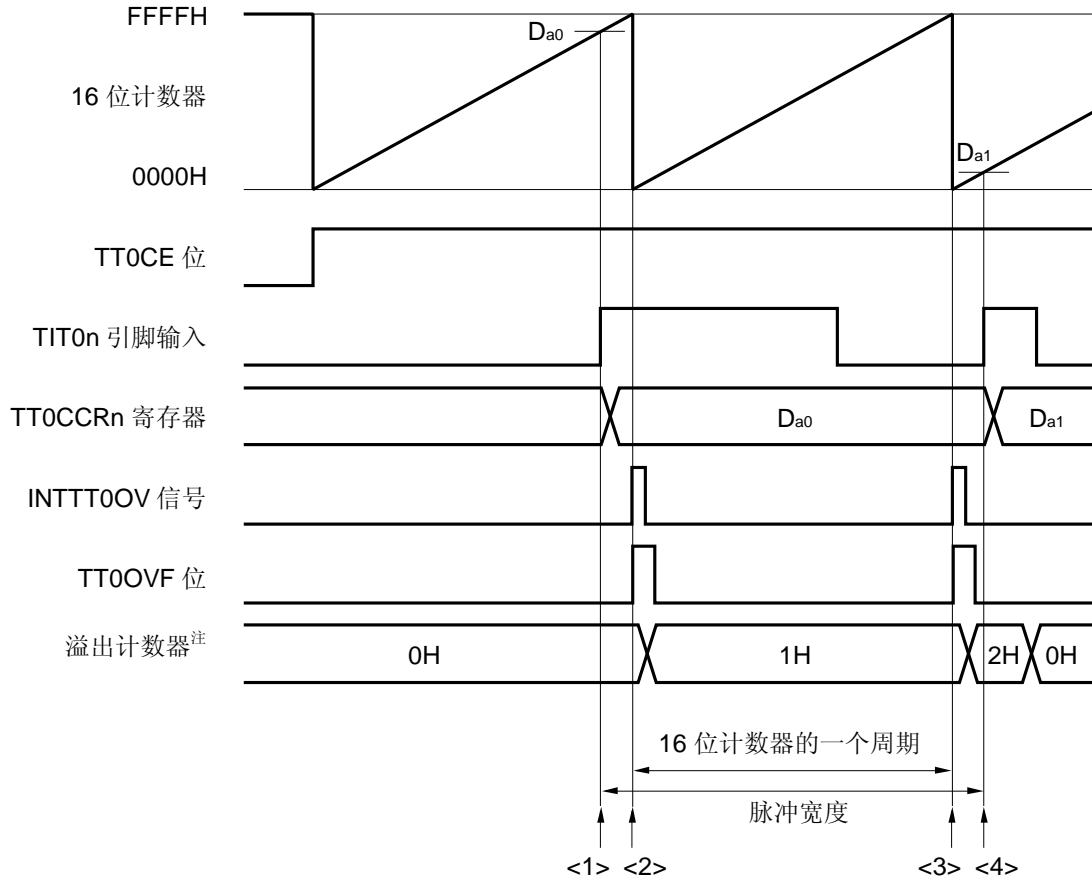


如果在捕获触发间隔期间，发生两次或更多次的溢出，则无法获得正确的脉冲宽度。

如果捕获触发间隔时间较长时，应该减慢计数时钟以延长 16 位计数器的循环周期，或者使用软件来解决。

关于如何使用软件，示例如下。

捕获触发间隔较长时的示例



注 内部 RAM 中的溢出计数器由软件任意设置。

<1> 读取 TT0CCRn 寄存器 (TIT0n 引脚输入的默认值设置)。

<2> 发生溢出, 在溢出中断服务程序中增加溢出计数器的值, 并将溢出标志清为 0。

<3> 再次发生溢出。在溢出中断服务程序中增加溢出计数器的值, 并把溢出标志清为 0。

<4> 读取 TT0CCRn 寄存器。

读取溢出计数器。

→ 当溢出计数器为“N”时, 脉冲宽度可用式 $(N \times 10000H + D_{a1} - D_{a0})$ 计算。

在本例中, 脉冲宽度为 $(20000H + D_{a1} - D_{a0})$, 因为溢出实际发生了两次。

溢出计数器清零 (0H)。

备注 n=0, 1

(e) 清除溢出标志

在溢出标志为 1 时读取 TT0OVF 位之后，通过 CLR 指令将 TT0OVF 位清为 0，且在溢出标志为 1 时读取 TT0OVF 位之后，将 8 位数据（第 0 位是 0）写入 TT0OPT0 寄存器。

9.6.7 脉冲宽度测量模式 (TT0MD3 至 TT0MD0 位=0110)

在脉冲宽度测量模式下，当 TT0CTL0.TT0CE 位被置为 1 时，16 位计时器/事件计数器 T 开始计数。每次检测到 TIT0n 引脚输入信号的有效边沿时，16 位计数器的计数值就保存到 TT0CCRn 寄存器中，且 16 位计数器清零为 0000H。

在发生捕获中断请求信号 (INTTT0CCn) 之后，可以通过读取 TT0CCRn 寄存器来测量有效边沿的间隔。

如图 9-39 所示，选择 TIT00 或 TIT01 引脚作为捕获触发信号输入引脚。使用 TT0IOC1 寄存器将未使用的引脚指定为“无边沿检测”。

图 9-39. 脉冲宽度测量模式的配置

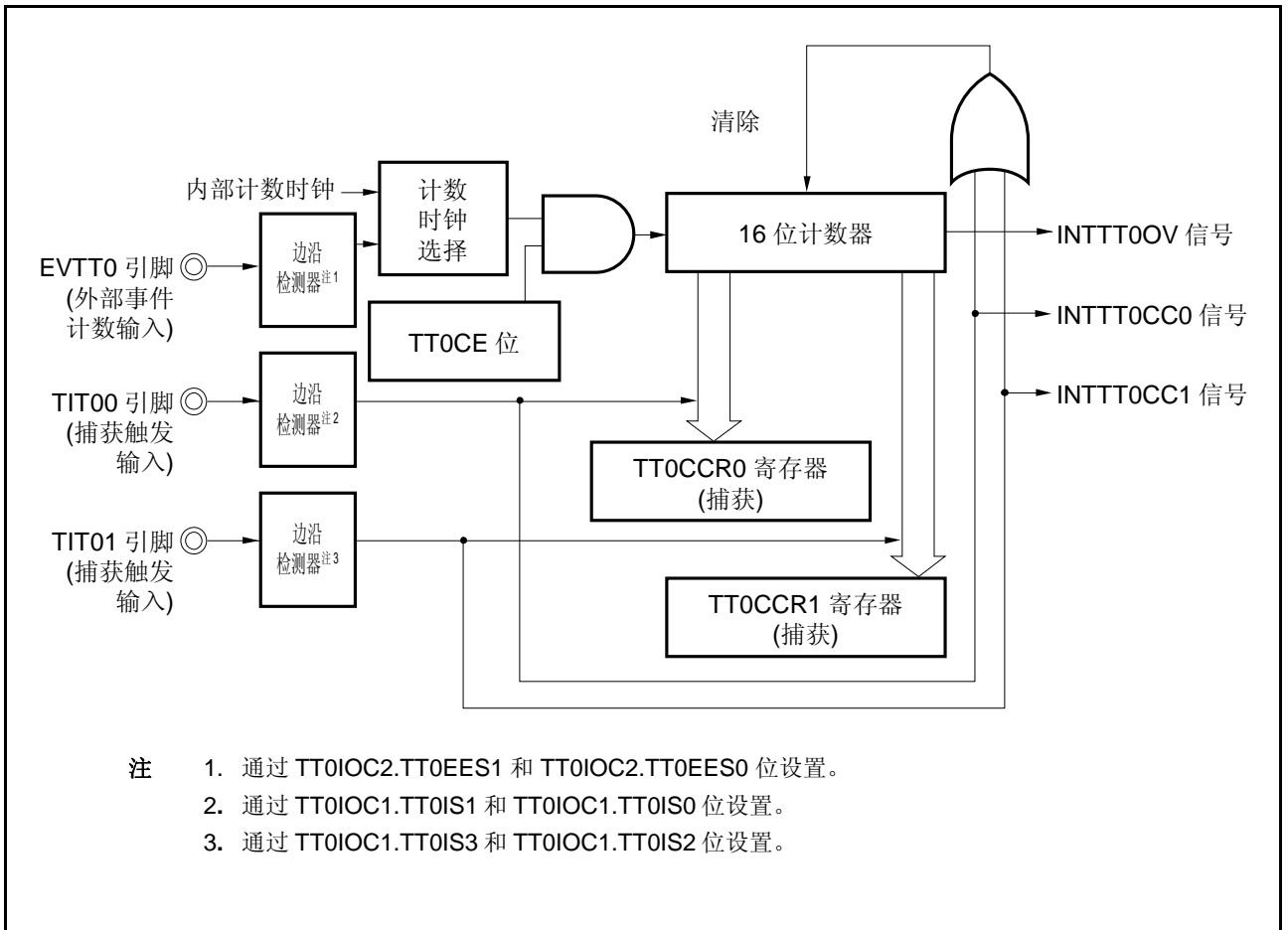
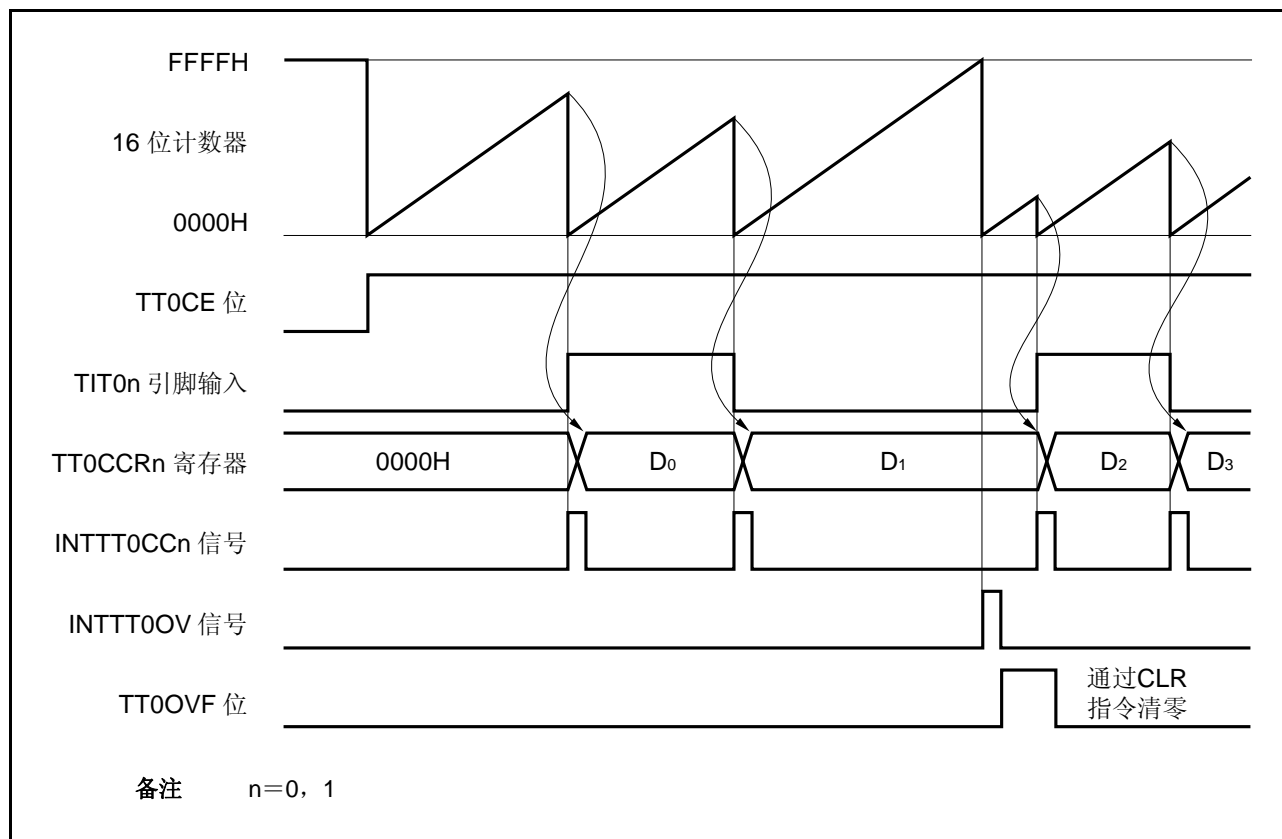


图 9-40. 脉冲宽度测量模式的基本时序



当 TT0CE 位被置为 1 时，16 位计数器开始计数。当检测到 TIT0n 引脚输入信号的有效边沿时，16 位计数器的计数值被存储到 TT0CCRn 寄存器中，16 位计数器清零为 0000H，并产生一个捕获中断请求信号 (INTTT0CCn)。

脉冲宽度的计算如下。

脉冲宽度 = 捕获值 (计数时钟周期)

当 16 位计数器计数到 FFFFH 时，即使没有有效边沿输入 TIT0n 引脚，在下一个计数时钟时，仍然会产生溢出中断请求信号 (INTTT0OV)，且计数器清零为 0000H，然后计数器继续计数。此时，溢出标志 (TT0OPT0.TT0OVF 位) 也被置为 1。使用软件执行 CLR 指令将溢出标志清为 0。

若溢出标志被设置为 1，则脉冲宽度的计算如下。

脉冲宽度 = (10000H × TT0OVF 位置位 (1) 的次数 + 捕获值) × 计数时钟周期

备注 n=0, 1

图 9-41. 脉冲宽度测量模式时的寄存器设置 (1/2)

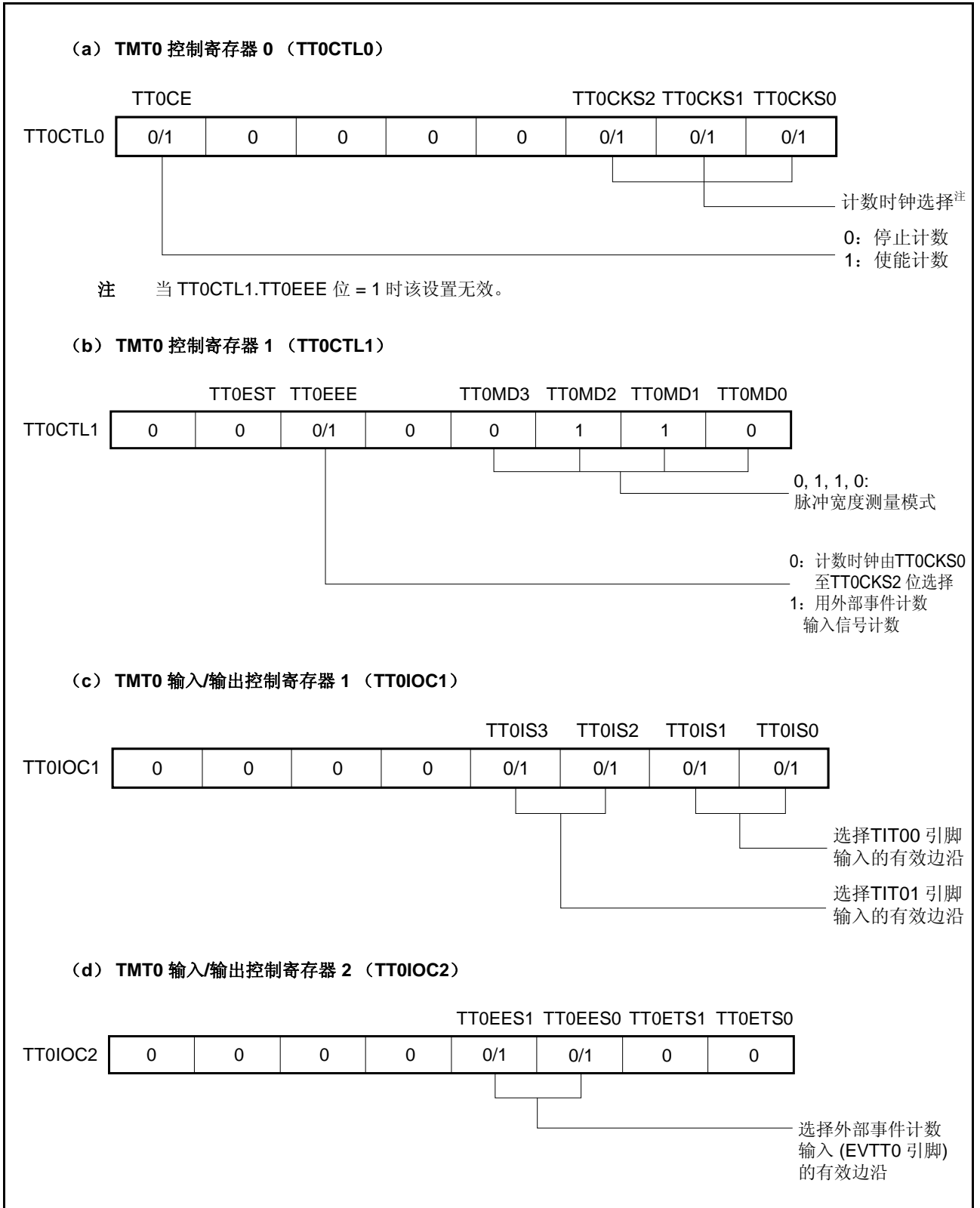
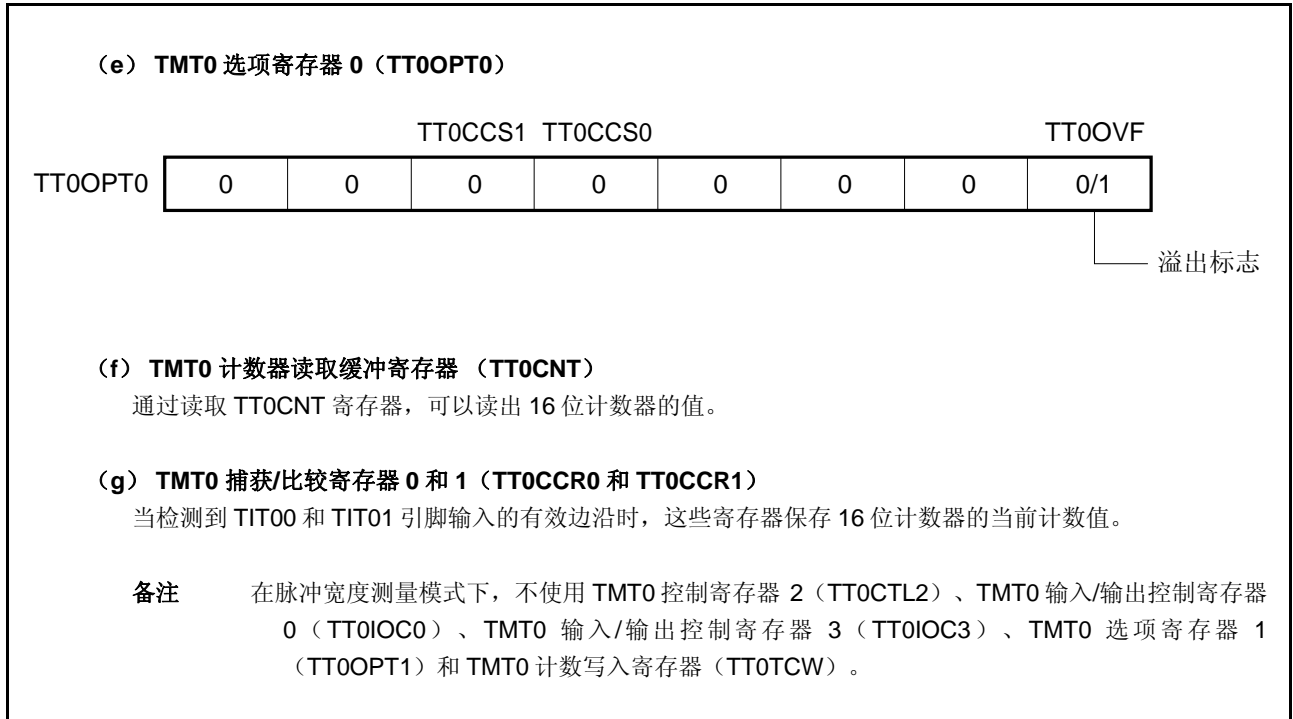
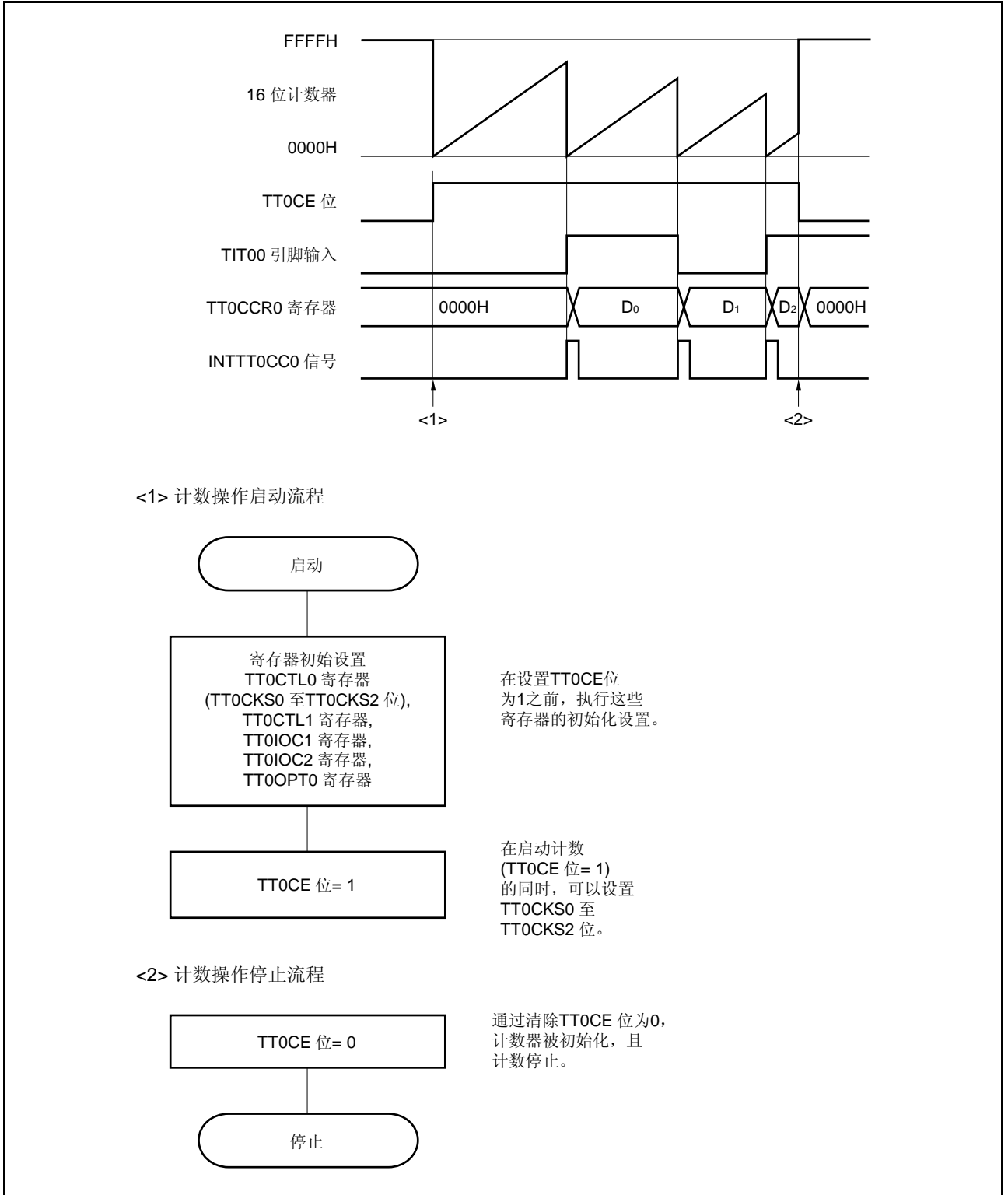


图 9-41. 脉冲宽度测量模式时的寄存器设置 (2/2)



(1) 脉冲宽度测量模式时的操作流程

图 9-42. 脉冲宽度测量模式的软件处理流程



(2) 脉冲宽度测量模式时的操作时序

(a) 清除溢出标志

在溢出标志为 1 时读取 TT0OVF 位之后，通过 CLR 指令将 TT0OVF 位清为 0，且在溢出标志为 1 时读取 TT0OVF 位之后，将 8 位数据（第 0 位是 0）写入 TT0OPT0 寄存器。

9.6.8 三角波 PWM 输出模式 (TT0MD3 至 TT0MD0 位 = 0111)

在三角波 PWM 输出模式下, 当 TT0CTL0.TT0CE 位被置为 1 时, 从 TOT01 引脚输出一个三角波 PWM 波形。

当 16 位计数器的计数值和 CCR0 缓冲寄存器的值匹配时, PWM 波形反相, 当 16 位计数器被置为 0000H 时, 该 PWM 波形从 TOT00 引脚输出。

图 9-43. 三角波 PWM 输出模式的配置

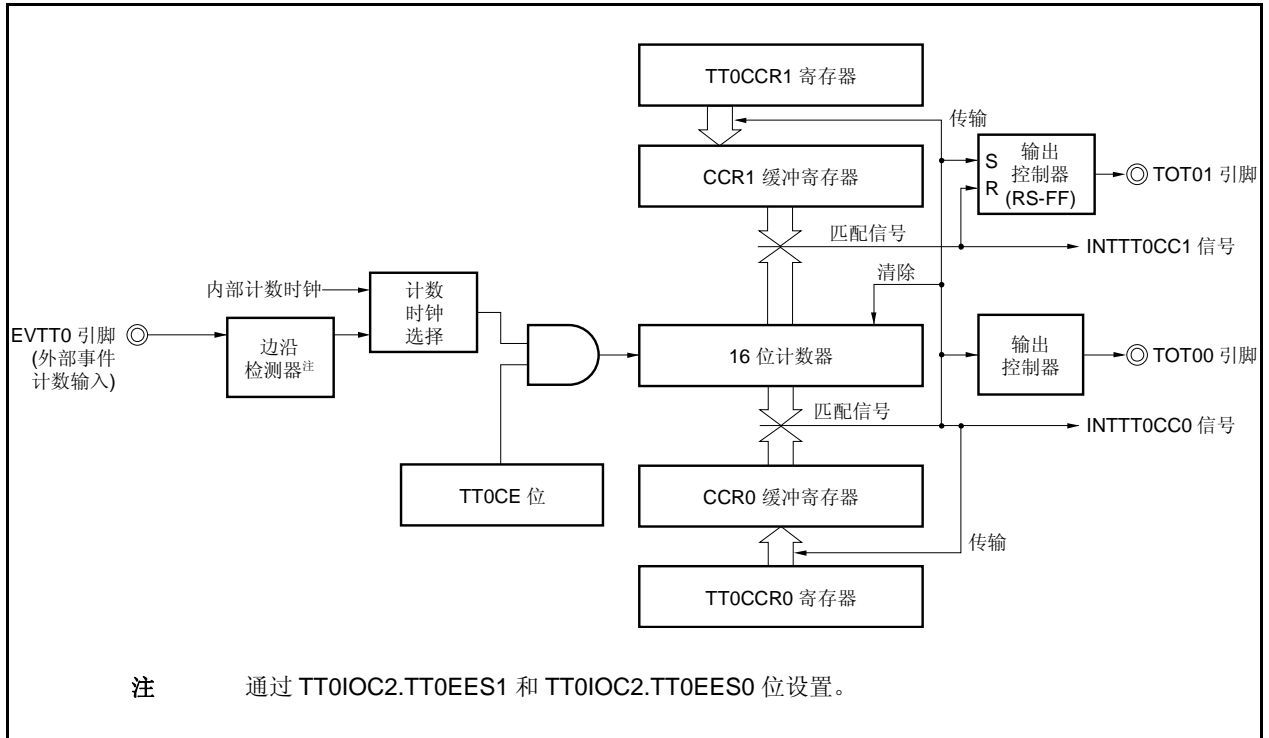
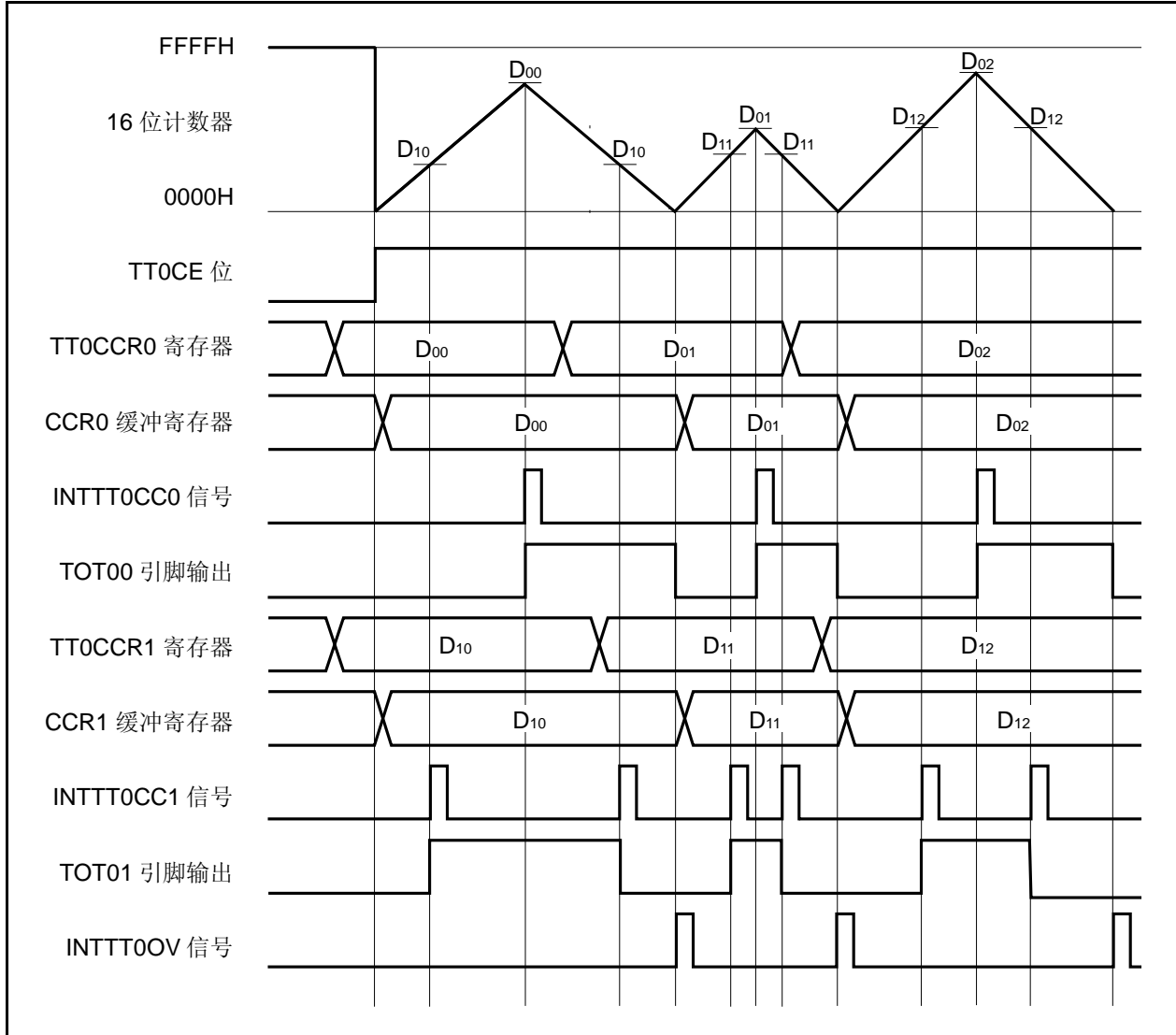


图 9-44. 三角波 PWM 输出模式的基本时序



当 TT0CE 位被置为 1 时，16 位计数器被从 FFFFH 清除为 0000H，且启动计数。三角 PWM 波形从 TOT01 引脚输出。

在三角波 PWM 输出模式下，计数器向上或向下计数。当向下计数时 16 位计数器达到 0000H，产生一个溢出中断请求信号（INTTT0OV）。此时 TT0OPT0.TT0OVF 位不被置为 1。当计数器向上计数时，如果 16 位计数器的计数值和 CCR0 缓冲寄存器的值匹配，产生一个比较匹配中断请求信号（INTTT0CC0）。

当 16 位计数器的值和 CCR0 缓冲寄存器的值匹配时，计数方向从向上改为向下计数，且当计数器被清除为 0000H 时，计数方向从向下改为向上计数。

在操作期间，通过重写 TT0CCRn 寄存器，可以改变 PWM 波形。要在操作期间改变 PWM 波形，最后写入 TT0CCR1 寄存器。

三角 PWM 波形的周期由 TT0CCR0 寄存器设置，占空比由 TT0CCR1 寄存器设置。TT0CCR0 寄存器的设置值范围是“0 ≤ TT0CCR0 ≤ FFFFH”。向下计数时，当 16 位计数器达到 0000H，重写值被反映出来。

即使只需要改变 PWM 波形的周期，应先将周期设置到 TT0CCR0 寄存器，然后将相同的值（和 TT0CCR1 寄存器当前值相等）写入 TT0CCR1 寄存器。

要将数据从 TT0CCRn 寄存器传送到 CCRn 缓冲寄存器，必须将数据写入 TT0CCR1 寄存器（n = 0, 1）。

9.6.9 编码器计数功能

编码器计数功能包括编码器比较模式（参见 9.6.10 编码器比较模式（TT0MD3 至 TT0MD0 位 = 1000））。

模式	TT0CCR0 寄存器	TT0CCR1 寄存器
编码器比较模式	仅作参考器	仅作参考器

(1) 向上/向下计数控制

通过 16 位计数器向上或向下计数，由输入编码器信号的相（TENC00 和 TENC01）以及 TT0CTL2.TT0UDS1 和 TT0CTL2.TT0UDS0 位的设置来控制。

使用编码器计数功能时，不能使用内部计数时钟和外部事件计数输入(EVTT0)。设置 TT0CTL0.TT0CKS2 至 TT0CTL0.TT0CKS0 位为 000 且 TT0CTL1.TT0EEE 位为 0。

(2) 16 位计数器的初始值设置

当 TT0CTL2.TT0ECC 位 = 0 时，在计数器启动操作（TT0CTL0.TT0CE 位 = 0 → 1）之后，设置在 TT0TCW 寄存器的初始计数值立即被传送到 16 位计数器。在检测到编码器输入信号（TENC00 或 TENC01）的有效边沿后，计数器启动操作。

(3) 基本操作

当 16 位计数器的计数值和 CCRn 缓冲寄存器的值匹配时，TT0CCRn 寄存器产生一个比较匹配中断请求信号（INTTT0CCn）。

(4) 清除操作

在编码器比较模式下满足下列条件时，16 位计数器被清除。

- 当 16 位计数器的值和比较寄存器（TT0CTL2.TT0ECM1 和 TT0CTL2.TT0ECM0 位被置位）的值匹配时。
- 当检测到编码器清除输入信号(TECR0)的边沿时。(当 TT0IOC3.TT0SCE 位 = 0 时，TT0ECS1 和 TT0ECS0 位被置位)
- 当检测到 TENC00, TENC01 和 TECR0 引脚清除电平条件时。(当 TT0SCE 位 = 1 时，TT0ZCL, TT0BCL, 和 TT0ACL 位被置位)

备注 n = 0, 1

(5) TT0CTL2 寄存器的控制位

在编码器比较模式下，TT0CTL2 寄存器的设置如下。

表 9-8. TT0CTL2 寄存器的设置

模式	TT0UDS1, TT0UDS0 位 (<1>)	TT0ECM1 位 (<2>)	TT0ECM0 位 (<2>)	TT0LDE 位 (<3>)	计数器清除 (目标比较寄存器)	传送到计数器
编码器比较模式	可以设置为 00, 01, 10, 或 11.	0	0	0	-	-
				1		可能
		1	0	0	TT0CCR0	-
				1		可能 ^注
		1	0	无效	TT0CCR1	-
				1		无效

注 计数器可以操作的范围是从 0000H 到 TT0CCR0 寄存器的设置值。

(a) 各个位的概述

- <1> TT0UDS1 和 TT0UDS0 位通过编码器输入引脚 (TENC00 或 TENC01) 的输入的相来识别 16 位计数器计数方向 (向上或向下)。
- <2> 当 16 位计数器的计数值和 CCR0 或 CCR1 缓冲寄存器的值匹配时, TT0ECM1 和 TT0ECM0 位控制 16 位计数器的清除。
- <3> 当计数器下溢时, TT0LDE 位控制将 TT0CCR0 寄存器的设置值传送给 16 位计数器的功能。只有当 TT0ECM1 和 TT0ECM0 位分别为 00 和 01 时, TT0LDE 位有效。当这些位被置为其他值, 该位无效。

(b) 各个位的详细解释

<1> TT0UDS1 和 TT0UDS0 位：向上/向下计数选择

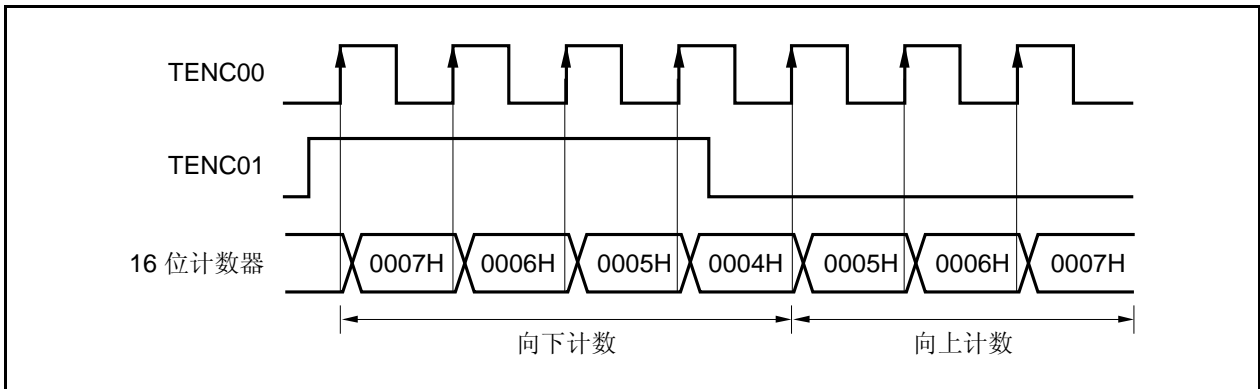
16 位计数器是向上计数或向下计数通过 TENC00 或 TENC01 引脚输入的相来识别，取决于 TT0UDS1 和 TT0UDS0 位的设置。只有在编码器比较模式下，这些位有效。

- 当 TT0UDS1 和 TT0UDS0 位 = 00

TENC00 引脚	TENC01 引脚	计数操作
上升沿	高电平	向下计数
下降沿		
双沿		
上升沿	低电平	向上计数
下降沿		
双沿		

备注 TENC00 引脚的检测边沿由 TT0IOC3.TT0EIS1 和 TT0EIS0 位指定。

图 9-45. 操作示例（当 TENC00 引脚的有效边沿被指定为上升沿且 TENC01 引脚的有效边沿被指定为无边沿检测）

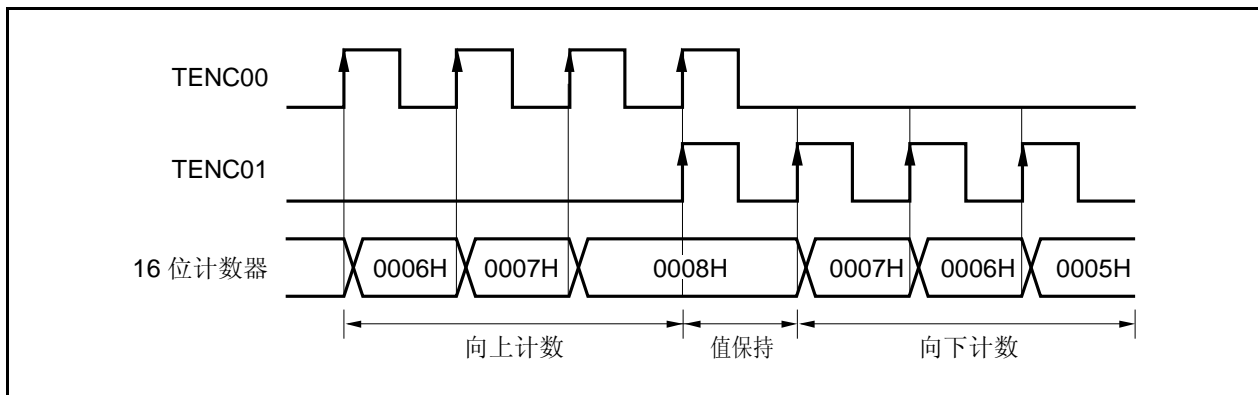


- 当 TT0UDS1 和 TT0UDS0 位 = 01

TENC00 引脚	TENC01 引脚	计数操作
低电平	上升沿	向下计数
	下降沿	
	双沿	
高电平	上升沿	
	下降沿	
	双沿	
上升沿	高电平	向上计数
下降沿		
双沿		
上升沿	低电平	
下降沿		
双沿		
同时输入到 TENC00 和 TENC01 引脚		计数器不执行计数操作，而是保持之前的瞬时值

备注 TENC00 和 TENC01 引脚的边沿检测由 TT0IOC3.TT0EIS1 和 TT0IOC3.TT0EIS0 位指定。

图 9-46. 操作示例（当 TENC00 和 TENC01 引脚的有效边沿被指定为上升沿时）



- 当 TT0UDS1 和 TT0UDS0 位 = 10

TENC00 引脚	TENC01 引脚	计数操作
低电平	下降沿	计数器不执行计数操作，而是保持之前的瞬时值
上升沿	低电平	向下计数
高电平	上升沿	计数器不执行计数操作，而是保持之前的瞬时值
下降沿	高电平	
上升沿	高电平	
高电平	下降沿	向上计数
下降沿	低电平	向上计数
低电平	上升沿	计数器不执行计数操作，而是保持之前的瞬时值
上升沿	上升沿	
下降沿	上升沿	
上升沿	下降沿	向下计数
下降沿	下降沿	向上计数

注意事项 TENC00 和 TENC01 引脚有效边沿的指定无效。

图 9-47. 操作示例 (当 TENC00 和 TENC01 引脚的有效边沿不相同时的计数操作)

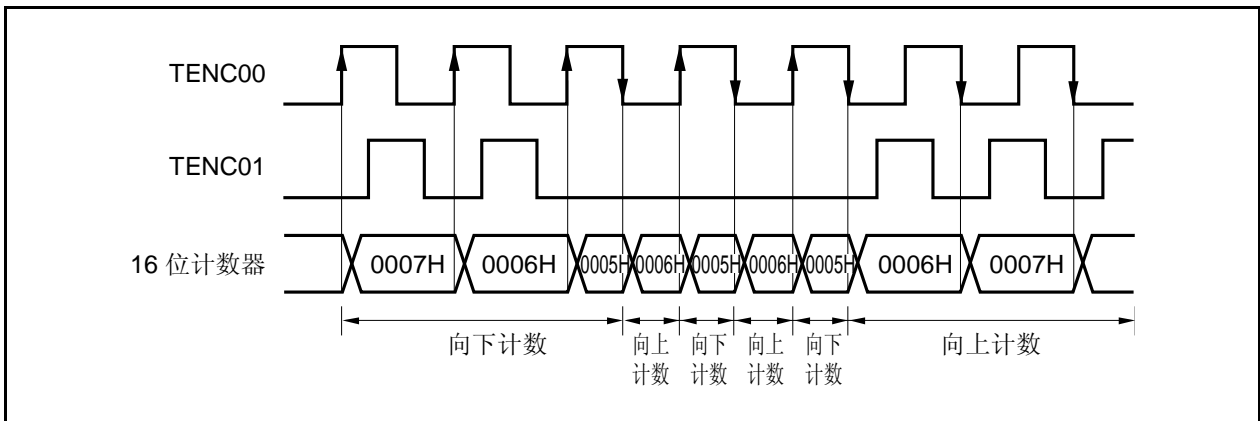
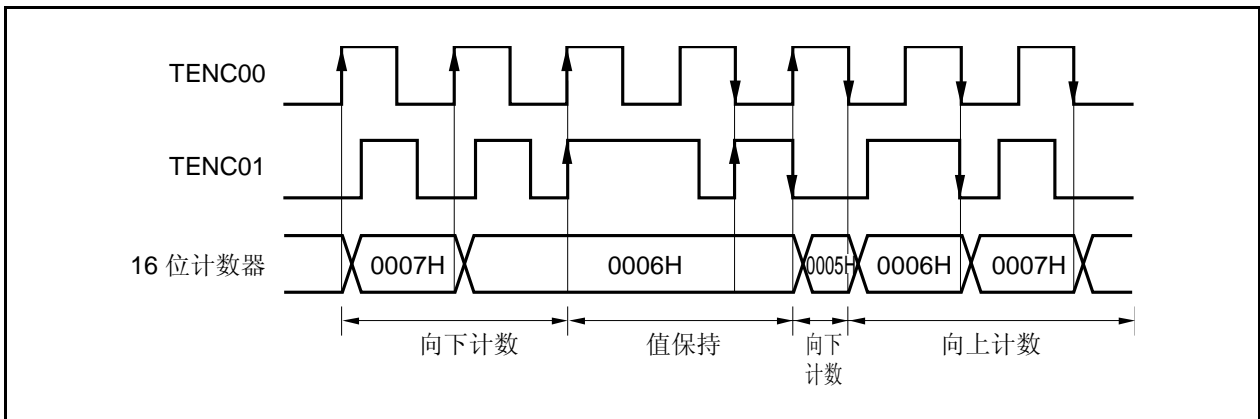


图 9-48. 操作示例 (当 TENC00 和 TENC01 引脚的有效边沿相同时的计数操作)



- 当 TT0UDS1 和 TT0UDS0 位 = 11

TENC00 引脚	TENC01 引脚	计数操作
低电平	下降沿	向下计数
上升沿	低电平	
高电平	上升沿	
下降沿	高电平	
上升沿	高电平	向上计数
高电平	下降沿	
下降沿	低电平	
低电平	上升沿	
同时输入到 TENC00 和 TENC01 引脚		计数器不执行计数操作，而是保持之前的瞬时值

注意事项 TENC00 和 TENC01 引脚有效边沿的指定无效。

图 9-49. 操作示例 (当 TENC00 和 TENC01 引脚的有效边沿不相同时的计数操作)

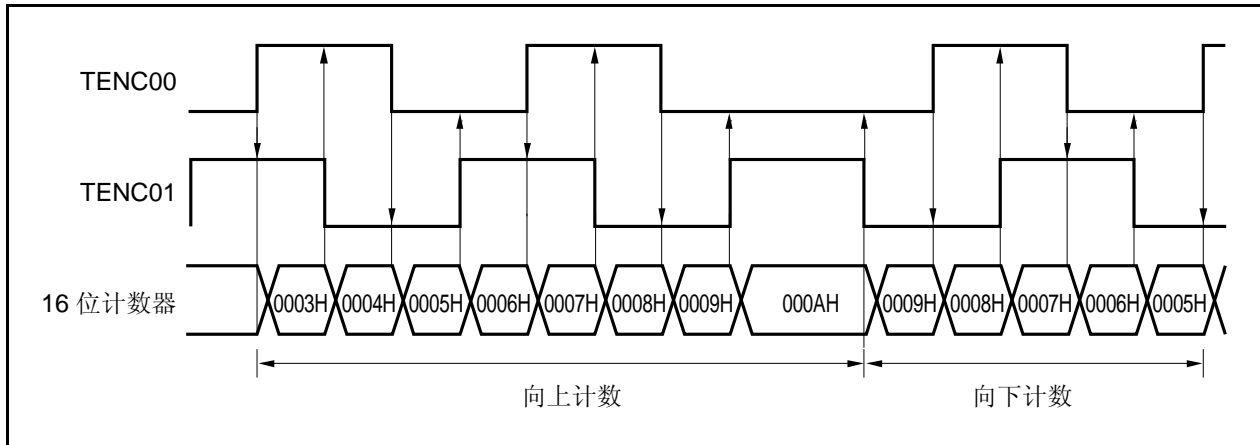
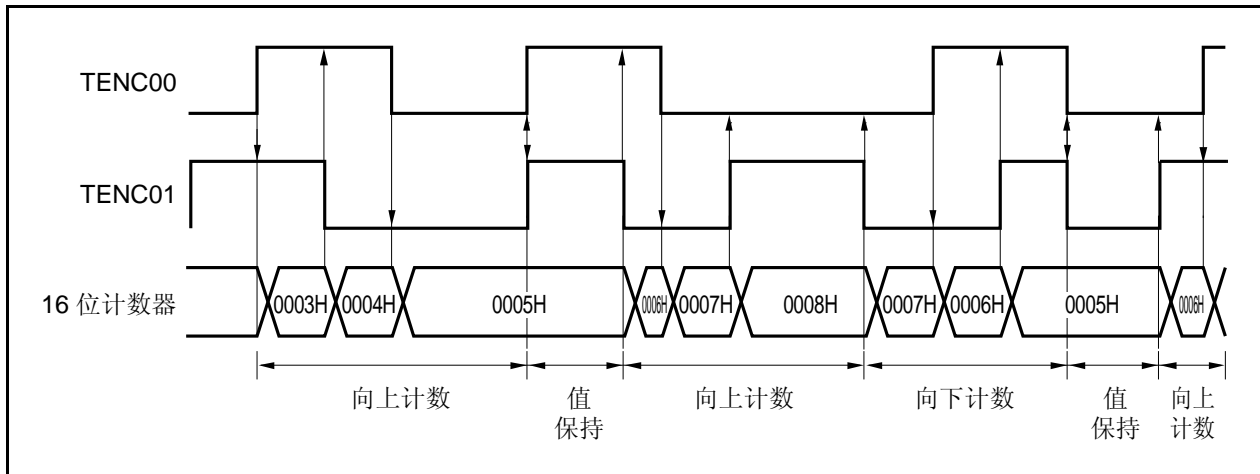


图 9-50. 操作示例 (当 TENC00 和 TENC01 引脚的有效边沿相同时的计数操作)



<2> TT0ECM1 和 TT0ECM0 位: 比较寄存器匹配时的定时器 /计数器清除功能

当计数器的计数值和 CCRn 缓冲寄存器的值匹配时, 16 位计数器根据 TT0ECM1 和 TT0ECM0 位的设置值执行计数操作。

- 当 TT0ECM1 和 TT0ECM0 位 = 00
当 16-位计数器的计数值和 CCRn 缓冲寄存器的值匹配时, 不清除 16 位计数器。
- 当 TT0ECM1 和 TT0ECM0 位 = 01
当 16-位计数器的计数值和 CCR0 缓冲寄存器的值匹配时, 16 位计数器在下列条件下执行一次计数操作。

下一次计数操作	描述
向上计数	16 位计数器被清除为 0000H
向下计数	16 位计数器的计数值向下计数

- 当 TT0ECM1 和 TT0ECM0 位 = 10
当 16-位计数器的计数值和 CCR1 缓冲寄存器的值匹配时, 16 位计数器在下列条件下执行一次计数操作。

下一次计数操作	描述
向上计数	16 位计数器的计数值向上计数
向下计数	16 位计数器被清除为 0000H

- 当 TT0ECM1 和 TT0ECM0 位 = 11
当 16-位计数器的计数值和 CCR0 缓冲寄存器的值匹配时, 16 位计数器在下列条件下执行一次计数操作。

下一次计数操作	描述
向上计数	16 位计数器被清除为 0000H
向下计数	16 位计数器的计数值向下计数

当 16-位计数器的计数值和 CCR1 缓冲寄存器的值匹配时, 16 位计数器在下列条件下执行一次计数操作。

Next 计数操作	描述
向上计数	16 位计数器的计数值向上计数
向下计数	16 位计数器被清除为 0000H

<3> TT0LDE 位: 当计数器下溢时, 将 TT0CCR0 寄存器的设置值传送到 16 位计数器。
 当 TT0LDE 位 = 1 时, 发生计数器下溢, TT0CCR0 寄存器的设置值可以被传送到 16 位计数器。
 只有在编码器比较模式下, TT0LDE 位才有效。

- 计数操作的范围是从 0000H 到 TT0CCR0 寄存器的设置值
 如果当 TT0LDE 位 = 1 且 TT0ECM1 和 TT0ECM0 位 = 01 时, 16 位计数器执行了一次计数操作, 或者当 TT0ECM0 位 = 1 时, 计数器的计数值和 CCR0 缓冲寄存器的值匹配, 如果下一次计数操作是向上计数, 则 16 位计数器被清除为 0000H。
 如果当 TT0LDE 位 = 1 时, 16 位计数器下溢, TT0CCR0 寄存器的设置值被传送到计数器。
 因此, 计数器可以操作的范围是从 0000H 到 TT0CCR0 寄存器的设置值, 其中计数值的上限的值是 TT0CCR0 寄存器的设置值, 下限的值是 0000H。

图 9-51. 操作示例 (计数操作的范围是从 0000H 到 TT0CCR0 寄存器的设置值)

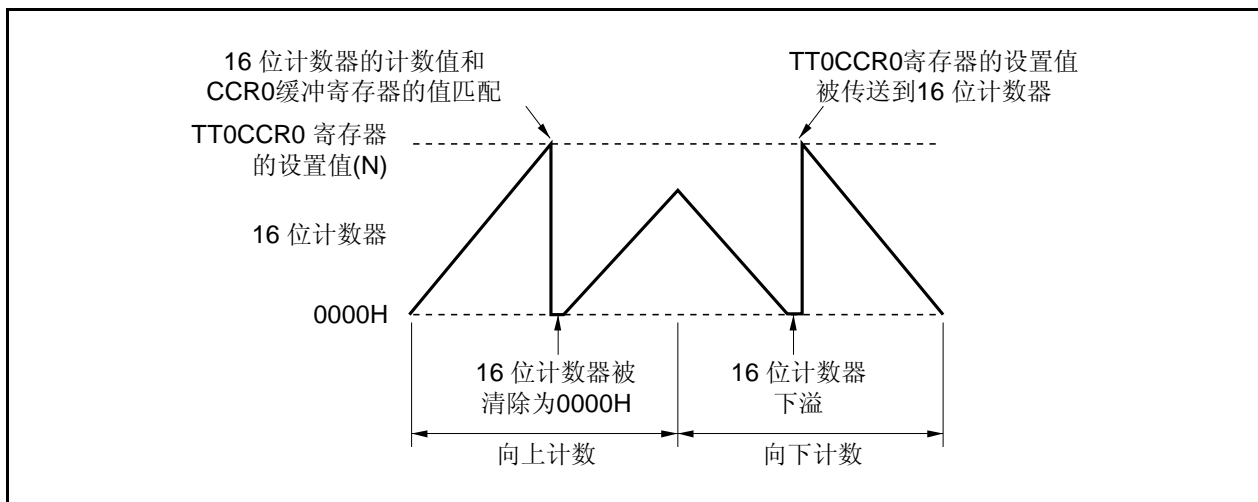
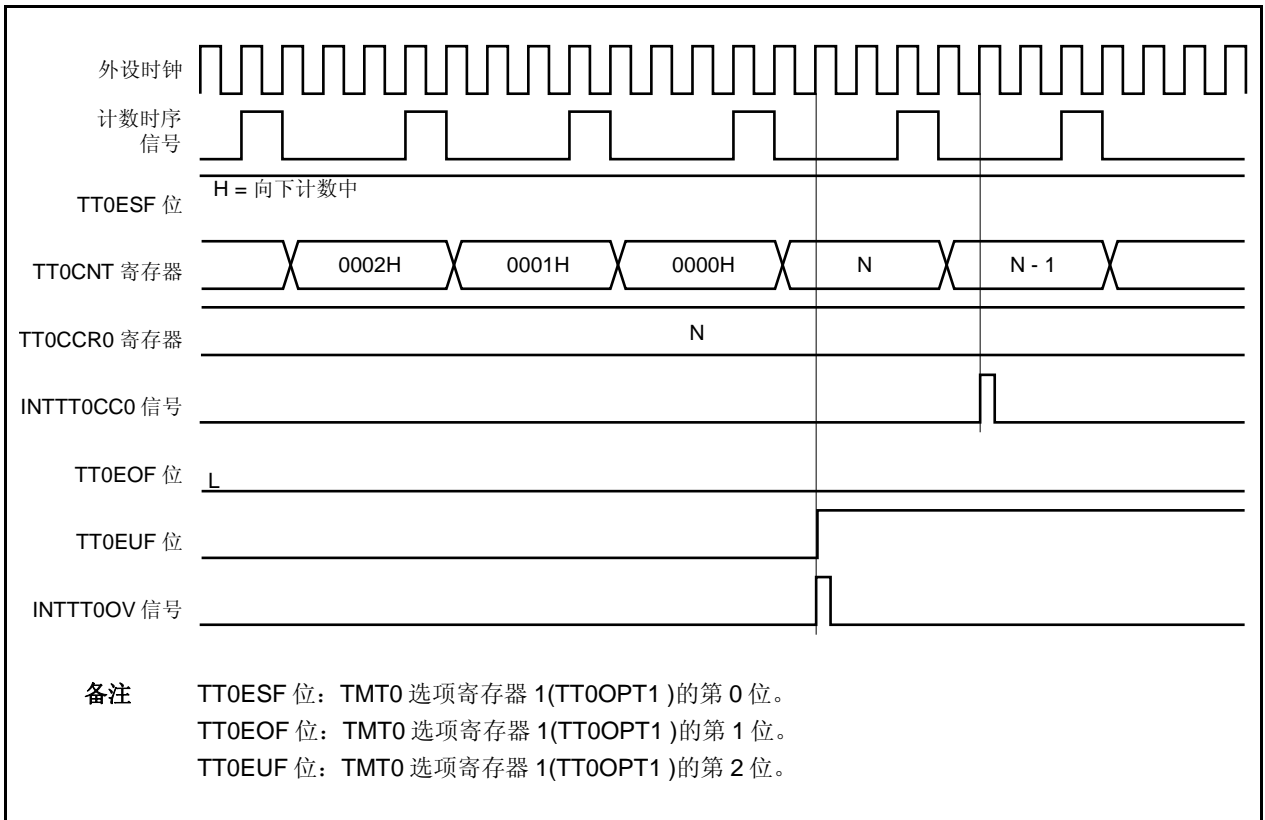


图 9-52. 操作时序 (计数操作的范围是从 0000H 到 TT0CCR0 寄存器的设置值)



(6) 通过编码器清除信号 (TECR0 引脚) 将计数器清除为 0000H 的功能

通过 TECR0 引脚的输入信号，有两种方法可以将 16 位计数器清除为 0000H，清除方法由 TT0IOC3.TT0SCE 位来选择。根据 TT0SCE 位的设置，该位也可以控制 TT0IOC3.TT0ZCL, TT0IOC3.TT0BCL, TT0IOC3.TT0ACL, TT0IOC3.TT0ESC1 和 TT0IOC3.TT0ECS0 位。

只有在编码器比较模式下，才可以通过下面描述的方法来清除计数器。

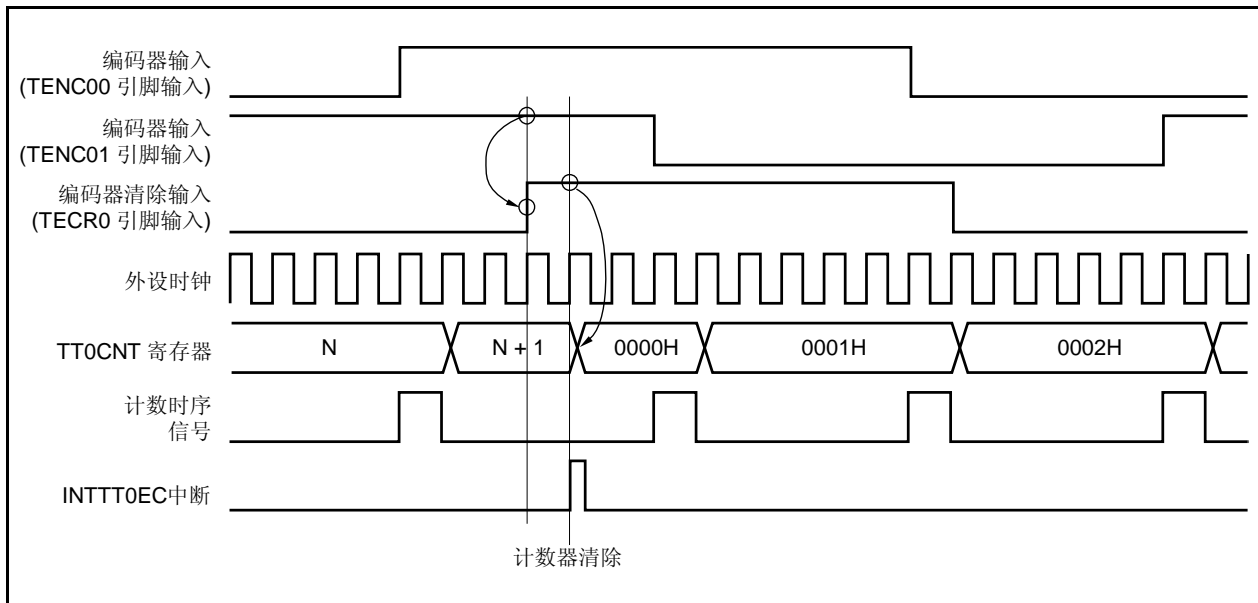
表 9-9. TT0SCE 位和 TT0ZCL, TT0BCL, TT0ACL, TT0ECS1 和 TT0ECS0 位之间的关系

清除方法	TT0SCE 位	TT0ZCL 位	TT0BCL 位	TT0ACL 位	TT0ECS1, TT0ECS0 位
<1>	0	无效	无效	无效	有效
<2>	1	有效	有效	有效	无效

(a) 清除方法 <1>: 通过检测编码器清除信号 (TECR0 引脚) 的边沿 (TT0SCE 位 = 0)

当 TT0SCE 位 = 0 时，如果检测到由 TT0ECS1 和 TT0ECS0 位指定的 TECR0 引脚有效边沿，16 位计数器被外部时钟同步清除为 0000H。此时，产生编码器清除中断请求信号 (INTTT0EC)。当 TT0SCE 位 = 0 时，TT0ZCL, TT0BCL 和 TT0ACL 位的设置无效。

图 9-53. 操作示例 (当 TT0SCE 位 = 0, TT0ECS1 和 TT0ECS0 位 = 01 且 TT0UDS1 和 TT0UDS0 位 = 11)



(b) 清除方法 <2>: 通过检测 TENC00, TENC01 和 TECR0 引脚的清除电平条件 (TT0SCE 位 = 1)

当 TT0SCE 位 = 1 时, 如果检测到由 TT0ZCL, TT0BCL 和 TT0ACL 位指定的 TECR0, TENC00 或 TENC01 引脚的清除电平条件, 16 位计数器被清除为 0000H。此时不产生编码器清除中断请求信号 (INTTT0EC)。当 TT0SCE 位 = 1 时, TT0ECS1 和 TT0ECS0 位的设置无效。

表 9-10. 当 TT0SCE 位 = 1 时的 16 位计数器清除条件

清除电平条件设置			编码器引脚的输入电平		
TT0ZCL 位	TT0BCL 位	TT0ACL 位	TECR0 引脚	TENC01 引脚	TENC00 引脚
0	0	0	L	L	L
0	0	1	L	L	H
0	1	0	L	H	L
0	1	1	L	H	H
1	0	0	H	L	L
1	0	1	H	L	H
1	1	0	H	H	L
1	1	1	H	H	H

注意事项 当 TT0ZCL, TT0BCL 和 TT0ACL 位的清除电平条件和 TECR0, TENC01 或 TENC00 引脚的输入电平匹配时, 16 位计数器被清除为 0000H。

图 9-54. 操作示例 (当 TT0SCE 位 = 1, TT0ZCL 位 = 1, TT0BCL 位 = 0, TT0ACL 位 = 1, TT0UDS1 和 TT0UDS0 位 = 11, TECR0 = 高电平, TENC01 = 低电平且 TENC00 = 高电平) (1/3)

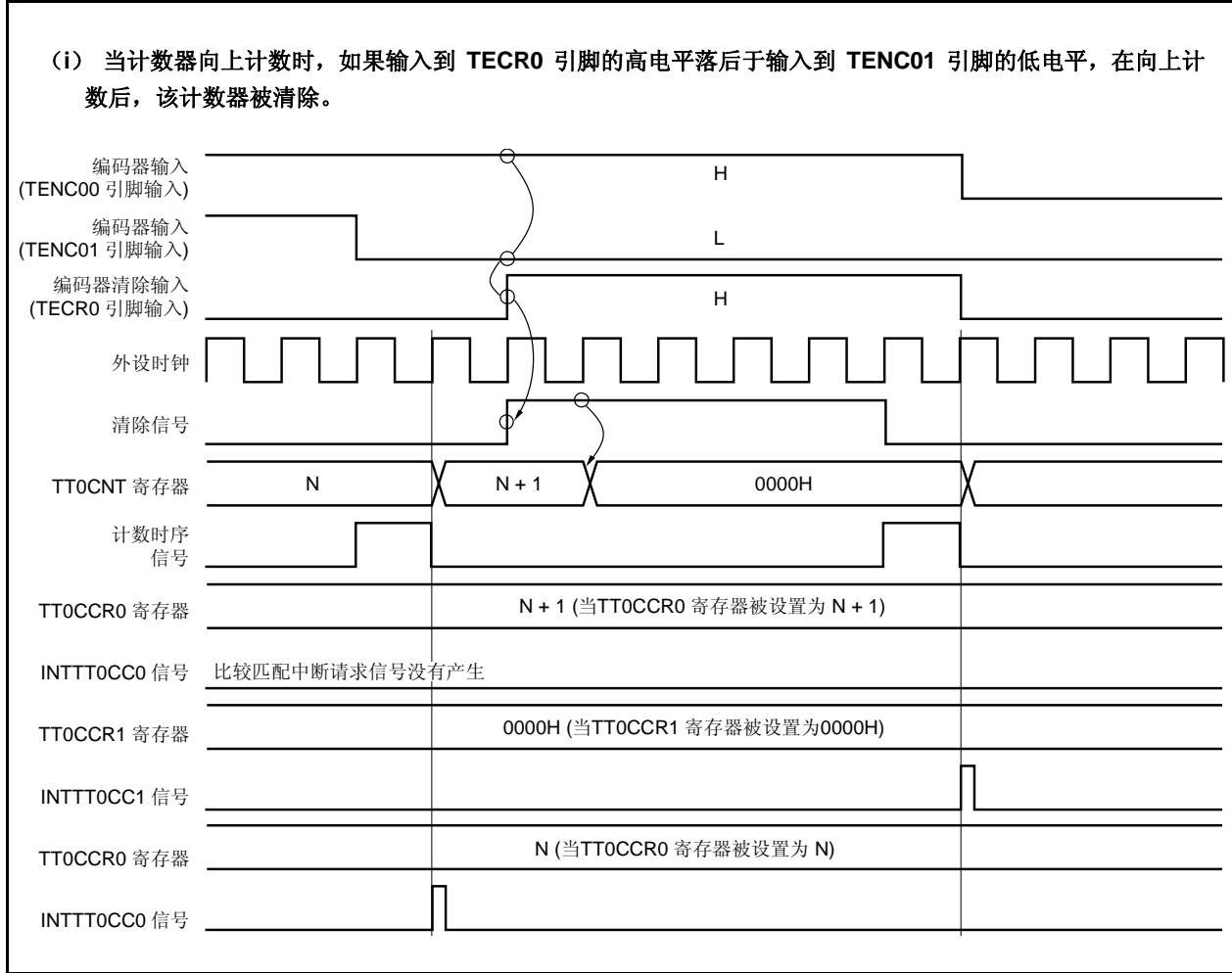


图 9-54. 操作示例 (当 TT0SCE 位 = 1, TT0ZCL 位 = 1, TT0BCL 位 = 0, TT0ACL 位 = 1, TT0UDS1 和 TT0UDS0 位 = 11, TECR0 = 高电平, TENC01 = 低电平且 TENC00 = 高电平) (2/3)

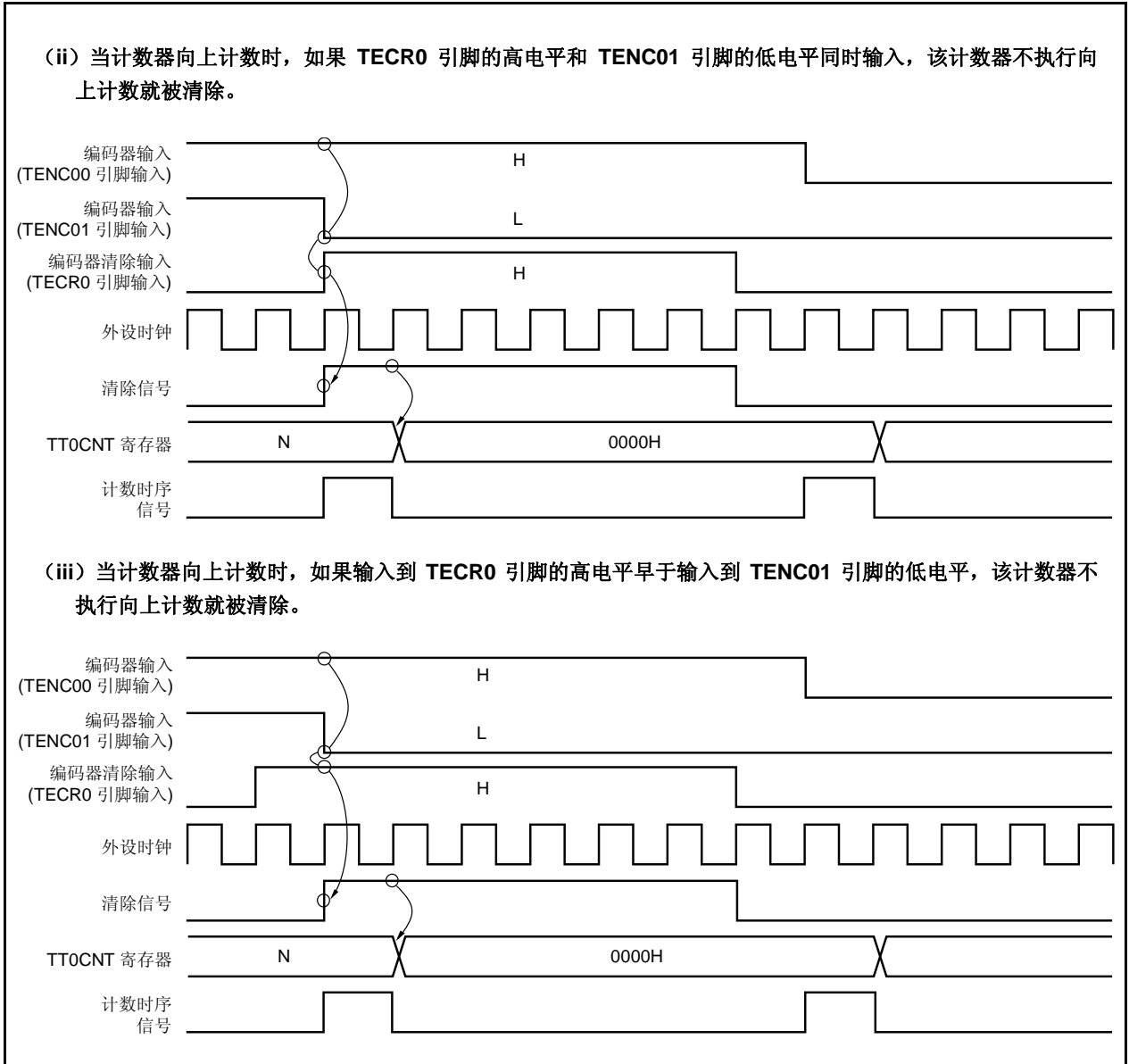
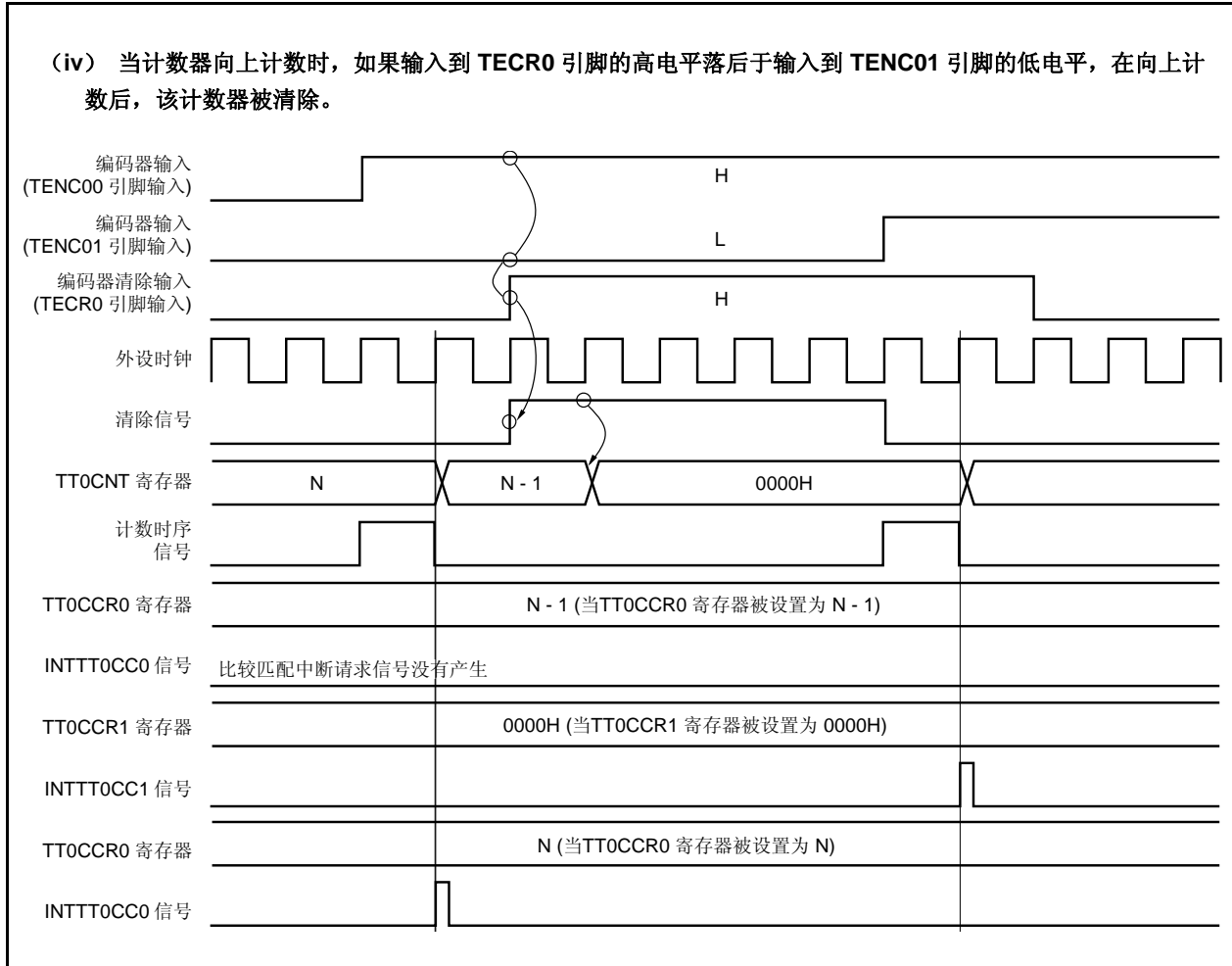


图 9-54. 操作示例 (当 TT0SCE 位 = 1, TT0ZCL 位 = 1, TT0BCL 位 = 0, TT0ACL 位 = 1, TT0UDS1 和 TT0UDS0 位 = 11, TECR0 = 高电平, TENC01 = 低电平且 TENC00 = 高电平) (3/3)

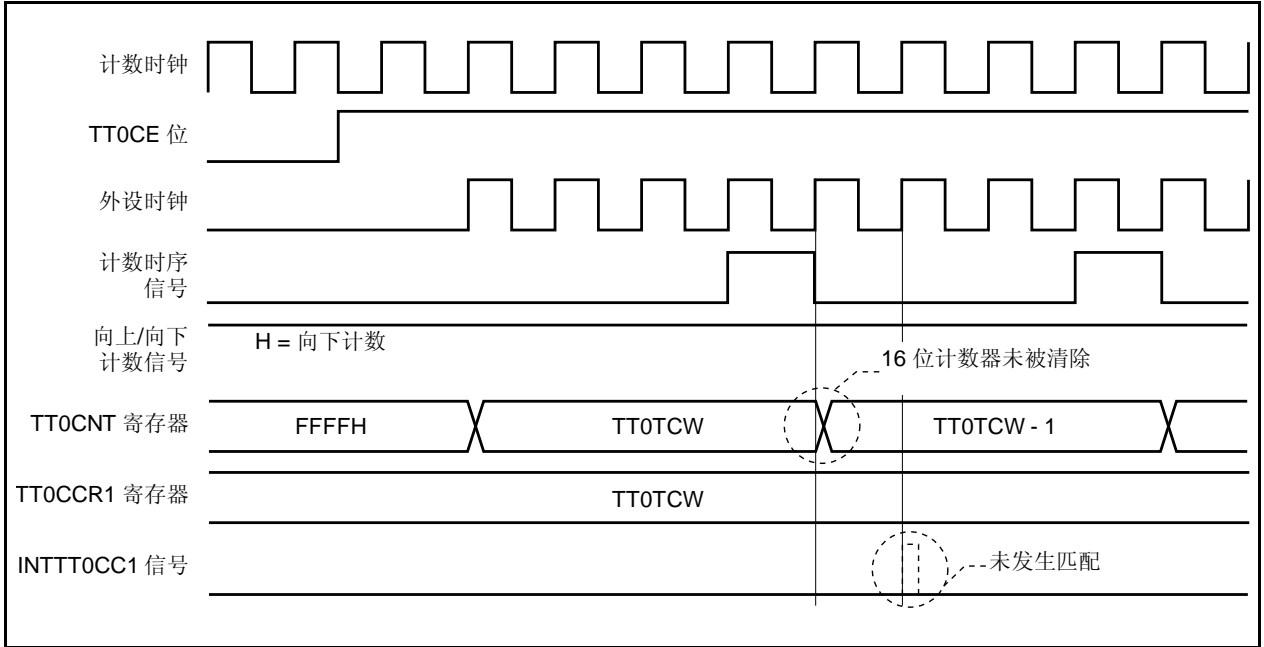


如果计数器按照这种方法清除, 即使输入到 TECR0 引脚的信号比较晚, 也不会发生误计数, 因为设置了 TECR0, TENC01 和 TENC00 引脚的清除电平条件, 当检测到清除电平条件时, 16 位计数器被清除为 0000H。

(7) 使用编码器计数功能的注意事项

(a) 如果在操作启动后，没有立即产生比较匹配中断

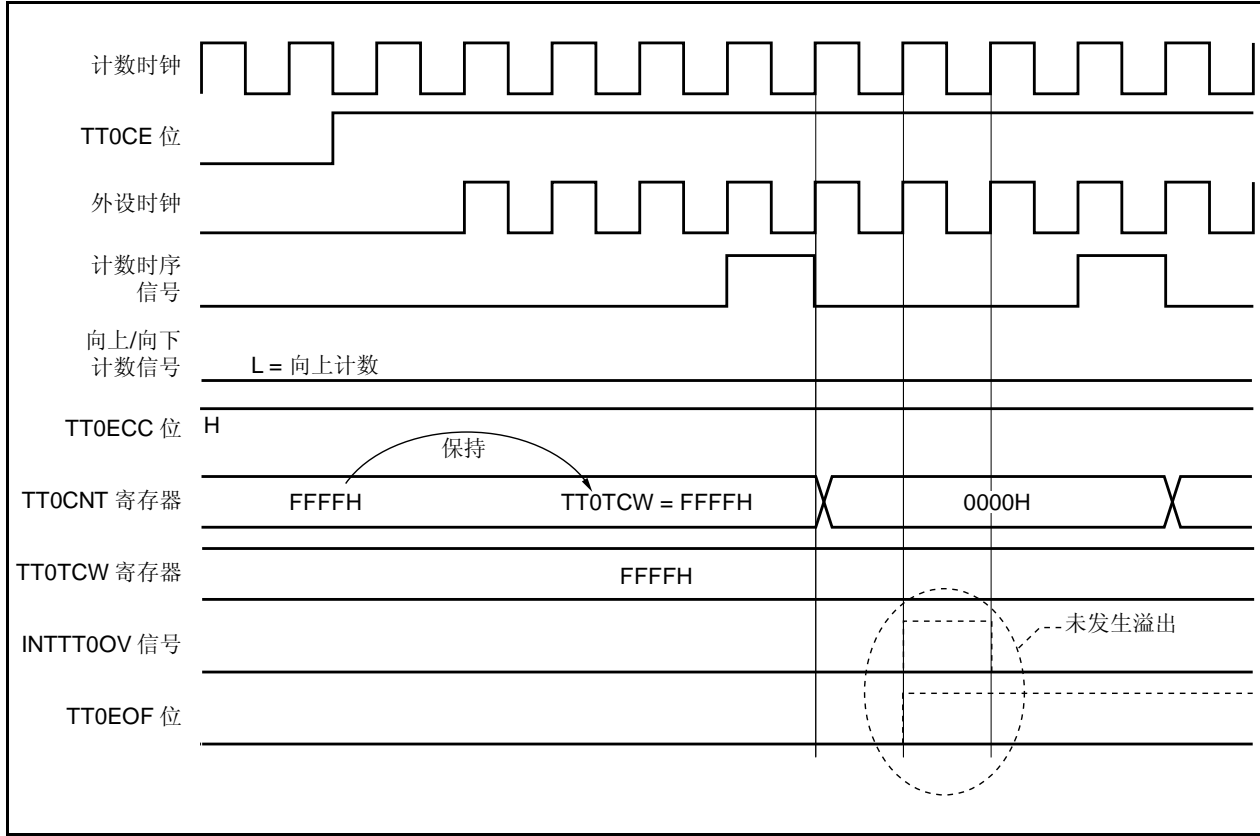
当 $TT0CTL2.TT0ECC$ 位 = 0 时，如果 $TT0TCW$ 寄存器的值被设置到 $TT0CCR0$ 或 $TT0CCR1$ 寄存器，并启动计数器操作，如果在操作启动后，立即发生 16 位计数器的计数值 ($TT0TCW$) 和 $CCRn$ 缓冲寄存器的值匹配，匹配被屏蔽且不产生比较匹配中断请求信号 ($INTTT0CCn$) ($n = 0, 1$)。而且，设置 $TT0CTL2.TT0ECM1$ 和 $TT0CTL2.TT0ECM0$ 位并不会清除 16 位计数器为 0000H。



(b) 如果在操作启动后，没有立即发生溢出

如果在 TT0CTL2.TT0ECC 位 = 1 时恢复计数操作，如果它保持的计数值是 FFFFH 且下一个计数操作是向上计数，16 位计数器不溢出。

在计数器启动操作之后，从一个计数值(TT0TCW 寄存器的值 = FFFFH)开始向上计数，计数器从 FFFFH 溢出到 0000H。但是，溢出的检测被屏蔽，溢出标志(TT0EOF)不被置位，不产生溢出中断请求信号(INTTT0OV)。



9.6.10 编码器比较模式 (TT0MD3 至 TT0MD0 位 = 1000)

在编码器比较模式下，将 TT0CCR0 和 TT0CCR1 寄存器都用作比较寄存器，且输入引脚用作编码器计数功能 (TENC00, TENC01 和 TECR0)，可以控制编码器。

在该模式下，可以通过三种方法将 16 位计数器清除为 0000H：当计数器的计数值和 CCRn 缓冲寄存器的值匹配时（产生比较匹配中断请求信号 (INTTT0CCn)），当检测到编码器清除输入 (TECR0 引脚) 的边沿时，当检测到 TENC00, TENC01 和 TECR0 引脚的清除电平条件时。

当 16 位计数器下溢时，TT0CCR0 寄存器的设置值被传送到计数器。

(1) 编码器比较模式操作流程

图 9-55. 编码器比较模式操作流程

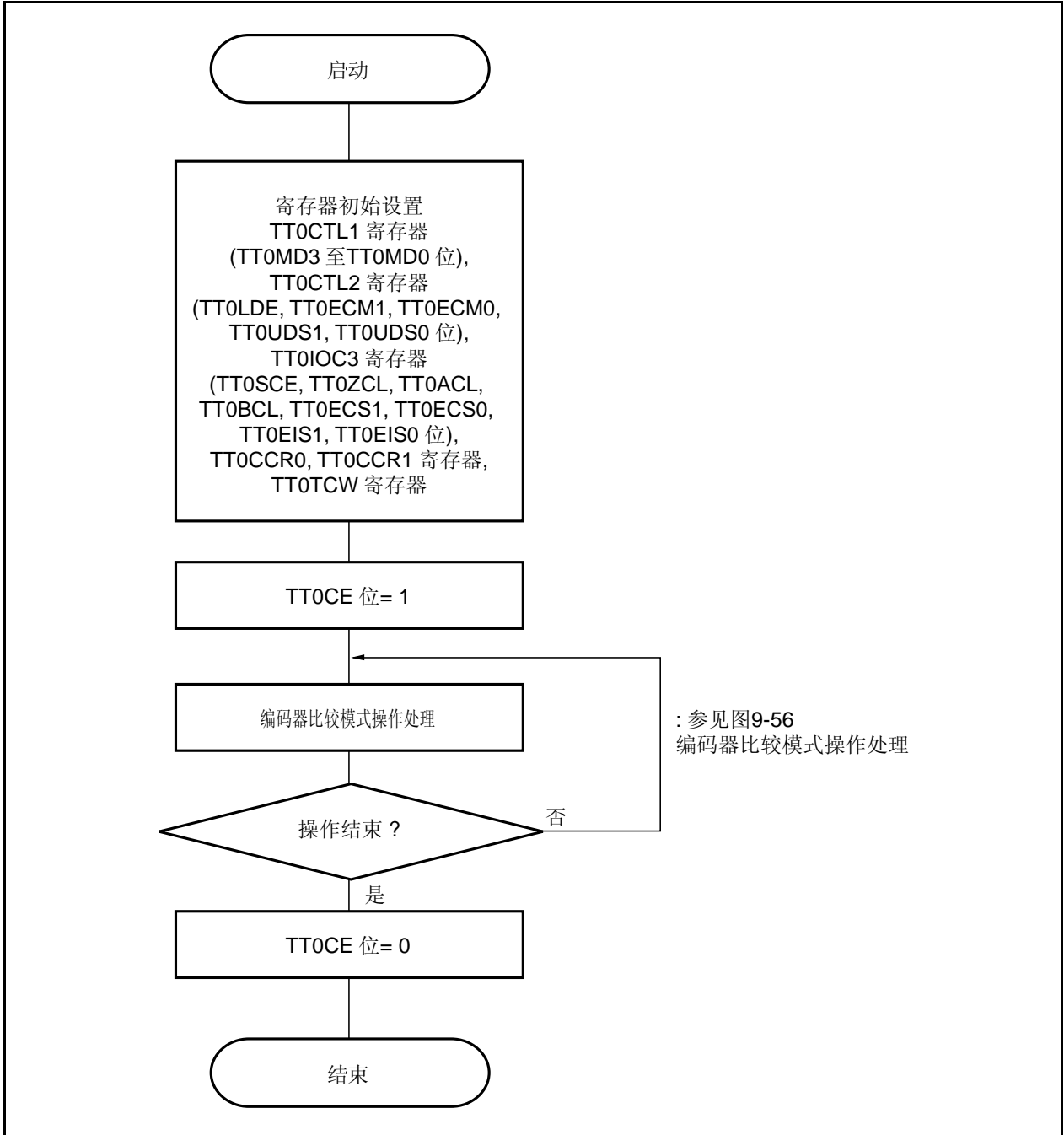
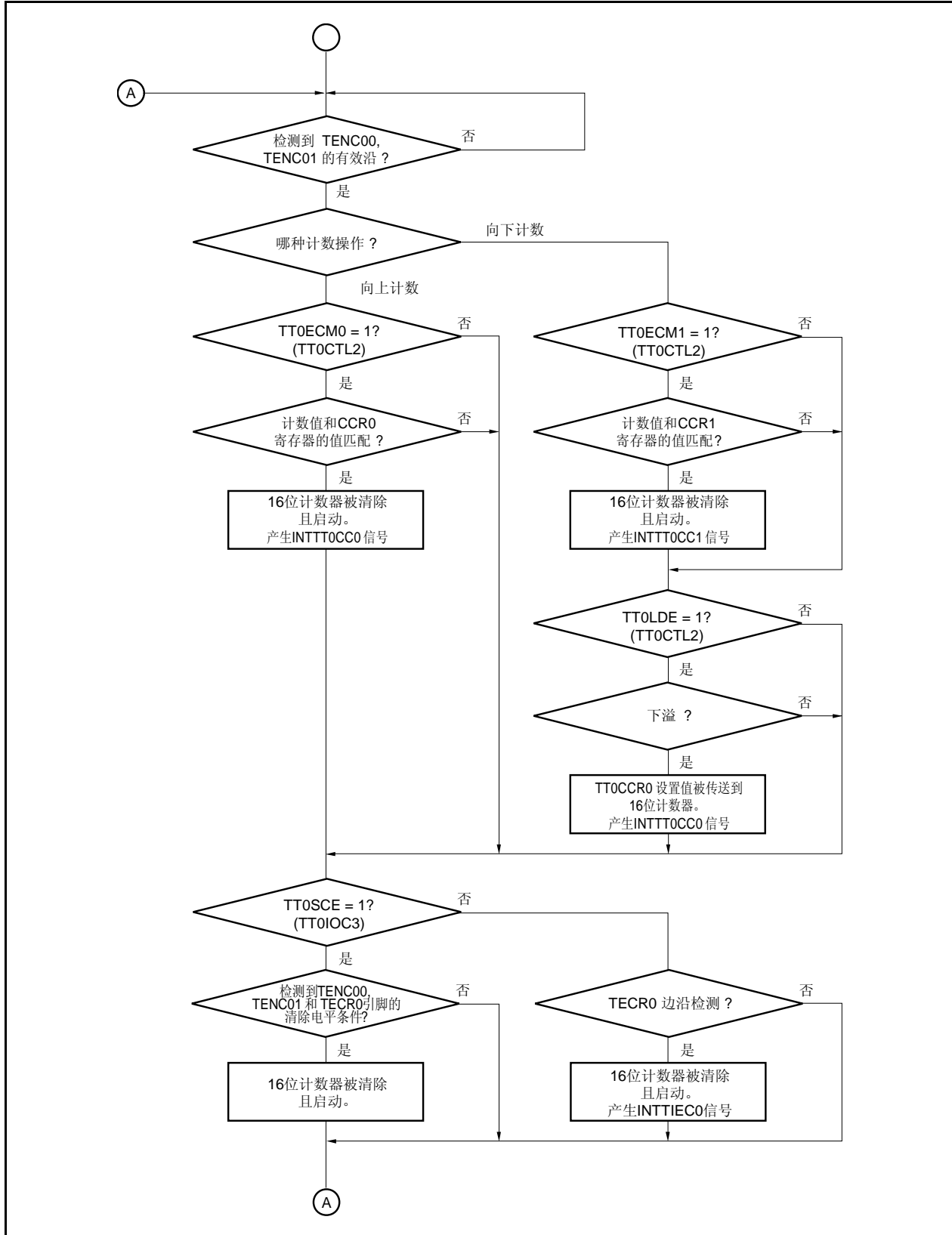


图 9-56. 编码器比较模式操作处理

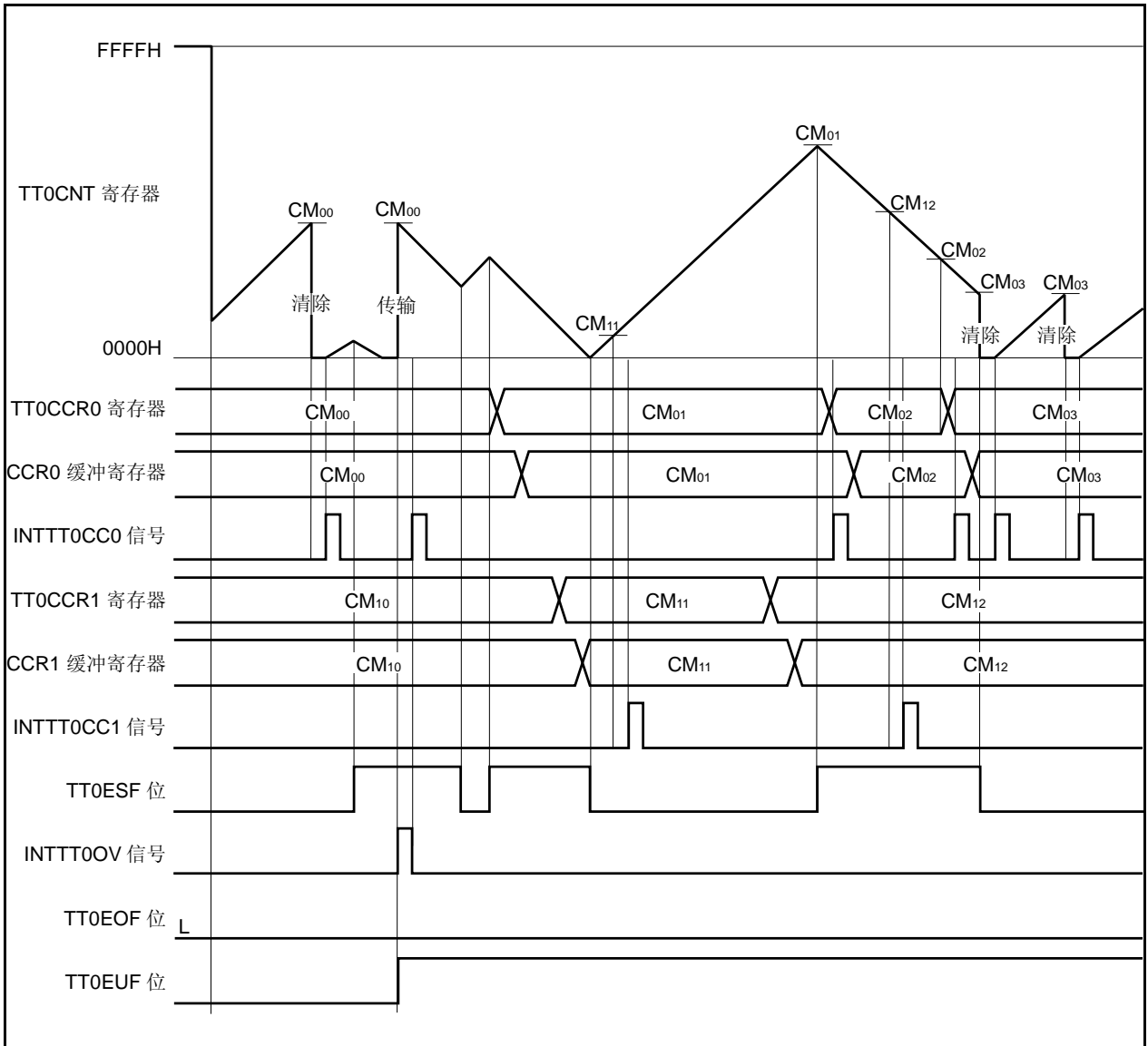


(2) 编码器比较模式操作时序

(a) 基本时序 1

[寄存器设置条件]

- TT0CTL2.TT0ECM1 和 TT0CTL2.TT0ECM0 位 = 01
当 16 位计数器的计数值和 CCR0 缓冲寄存器的值匹配时，16 位计数器被清除为 0000H。
- TT0CTL2.TT0LDE 位 = 1
当 16 位计数器下溢时，TT0CCR0 寄存器的设置值被传送到计数器。
- TT0IOC3.TT0SCE 位 = 0 且 TT0IOC3.TT0ECS1 和 TT0IOC3.TT0ECS0 位 = 00
当检测到编码器清除输入信号 (TECR0 引脚) 的边沿时，清除 16 位计数器的说明。(被指定为无边沿)



当 16 位计数器启动操作 (TT0CE 位 = 0 → 1) 时, TT0TCW 寄存器的设置值被传送到计数器, 且 16 位计数器启动操作。

当计数器的计数值和 CCR0 缓冲寄存器的值匹配时, 产生比较匹配中断请求信号 (INTTT0CC0)。因为 TT0ECM0 位 = 1, 如果下一个计数操作是向上计数, 16 位计数器被清除为 0000H。

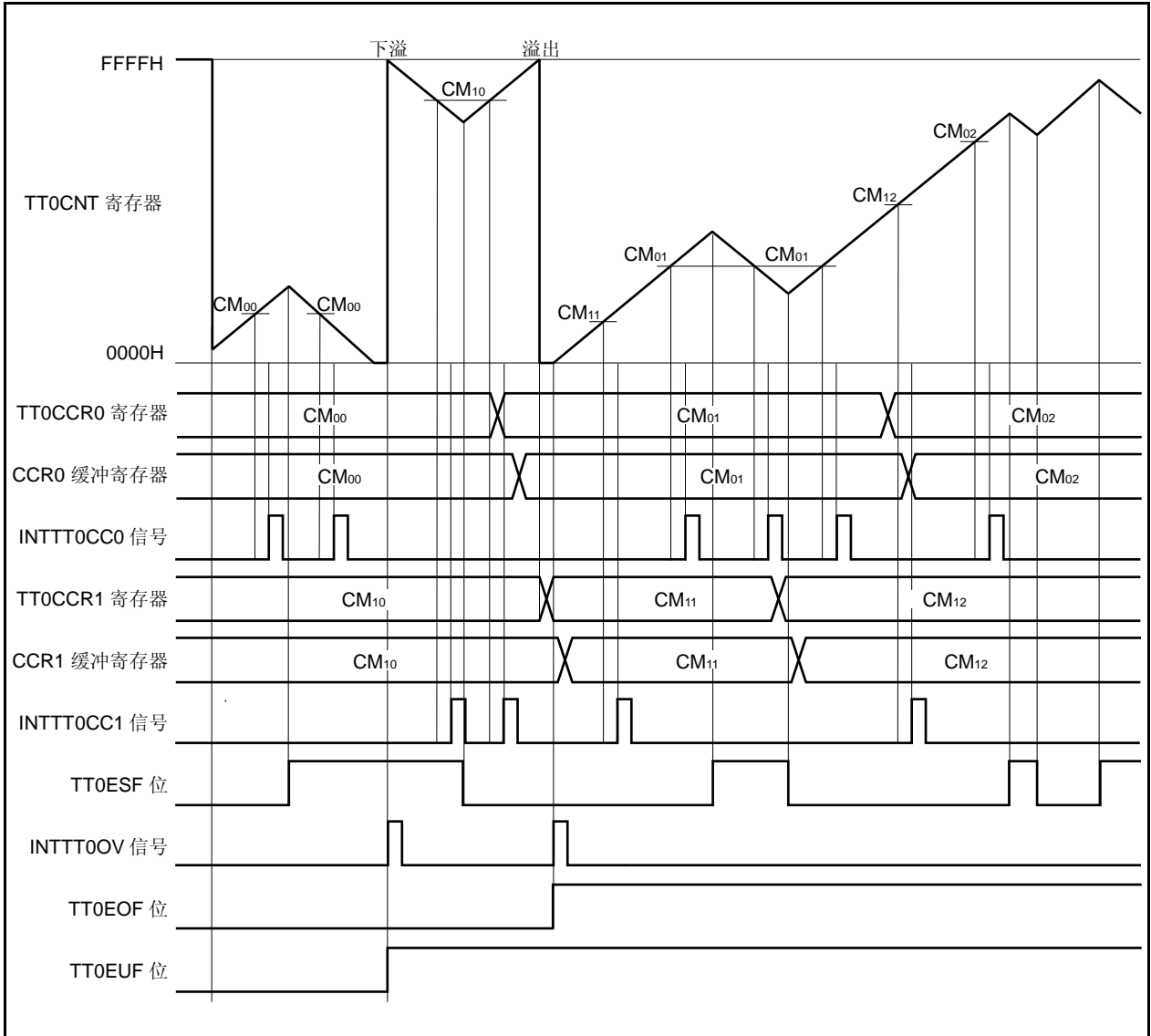
当计数器的计数值和 CCR1 缓冲寄存器的值匹配时, 产生比较匹配中断请求信号 (INTTT0CC1)。因为 TT0ECM1 位 = 0, 当 16 位计数器的值和 CCR1 缓冲寄存器的值匹配时, 16 位计数器被清除为 0000H。

当 TT0LDE 位 = 1 且 TT0ECM0 位 = 1 时, 计数器的操作范围是从 0000H 到 TT0CCR0 寄存器的设置值。

(b) 基本时序 2

[寄存器设置条件]

- TT0CTL2.TT0ECM1 和 TT0CTL2.TT0ECM0 位 = 00
即使 16 位计数器的计数值和 CCRn 缓冲寄存器的值匹配时，16 位计数器也不会被清除。(n = 0, 1)
- TT0CTL2.TT0LDE 位 = 0
在 16 位计数器下溢后，TT0CCR0 寄存器的设置值不被传送到计数器。
- TT0IOC3.TT0SCE 位 = 0, and TT0IOC3.TT0ECS1 and TT0IOC3.TT0ECS0 位 = 00
当检测到编码器清除输入信号 (TECR0 引脚) 的边沿时，清除 16 位计数器的说明。(被指定为无边沿)



当 16 位计数器启动操作 (TT0CE 位 = 0 → 1) 时, TT0TCW 寄存器的设置值被传送到计数器, 且 16 位计数器启动操作。

当计数器的计数值和 CCR0 缓冲寄存器的值匹配时, 产生比较匹配中断请求信号 (INTTT0CC0)。

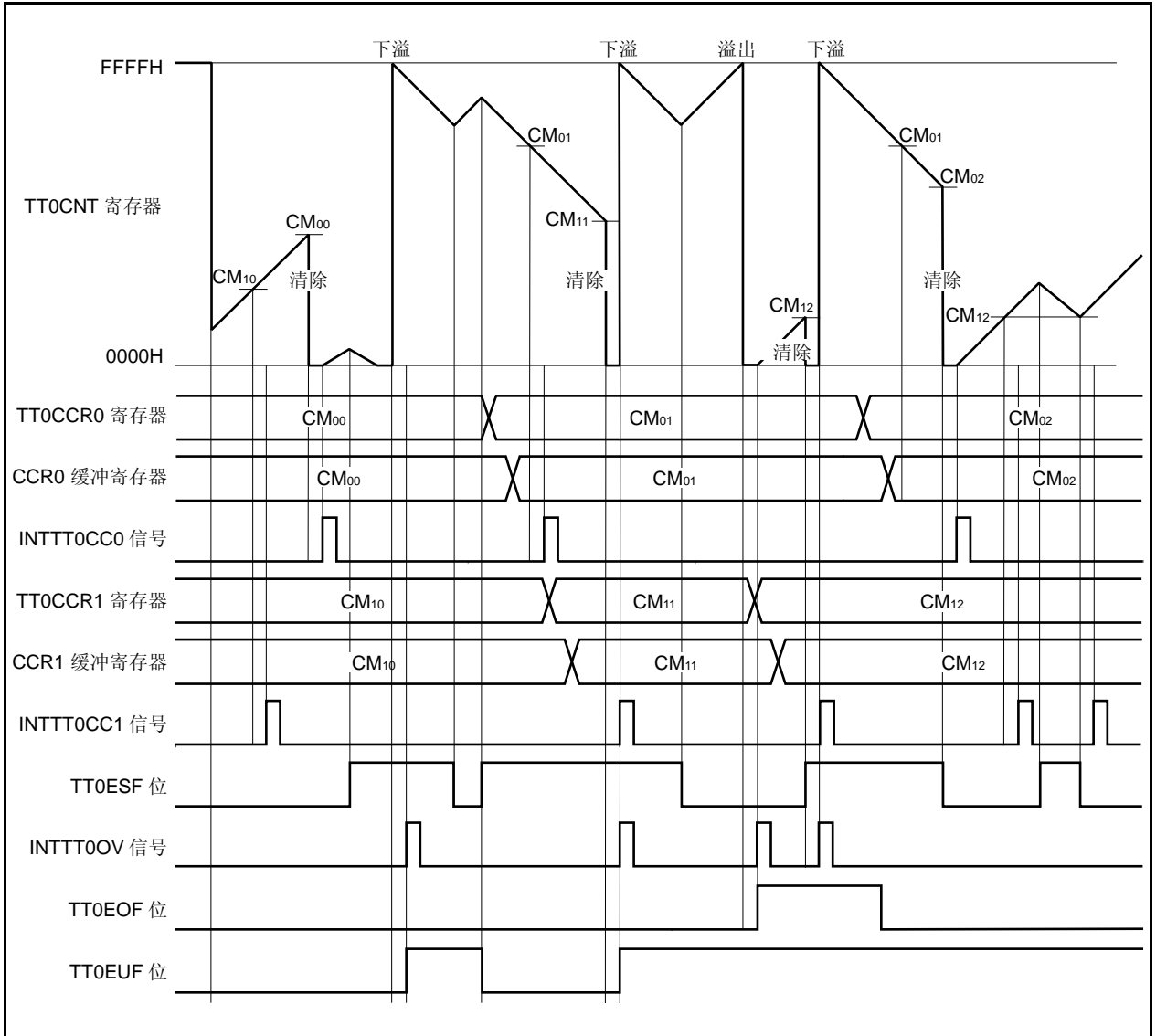
当计数器的计数值和 CCR1 缓冲寄存器的值匹配时, 产生比较匹配中断请求信号 (INTTT0CC1)。

即使 16 位计数器的计数值和 CCRn 缓冲寄存器的值匹配时, 16 位计数器也不会被清除为 0000H, 因为 TT0ECM1 和 TT0ECM0 位 = 00。 (n = 0, 1)

(c) 基本时序 3

[寄存器设置条件]

- TT0CTL2.TT0ECM1 和 TT0CTL2.TT0ECM0 位 = 11
当 16 位计数器的计数值和 CCR0 缓冲寄存器的值匹配时，16 位计数器被清除为 0000H。
当 16 位计数器的计数值和 CCR1 缓冲寄存器的值匹配时，16 位计数器被清除为 0000H。
- TT0CTL2.TT0LDE 位的设置无效。
- TT0IOC3.TT0SCE 位 = 0，且 TT0IOC3.TT0ECS1 和 TT0IOC3.TT0ECS0 位 = 00
当检测到编码器清除输入信号 (TECR0 引脚) 的边沿时，清除 16 位计数器的说明。(被指定为无边沿)



当 16 位计数器启动操作 (TT0CE 位 = 0 → 1) 时, TT0TCW 寄存器的设置值被传送到计数器, 且 16 位计数器启动操作。

当计数器的计数值和 CCR0 缓冲寄存器的值匹配时, 产生比较匹配中断请求信号(INTTT0CC0)。此时, 如果一个计数操作是向上计数, 则 16 位计数器被清除为 0000H。

当计数器的计数值和 CCR1 缓冲寄存器的值匹配时, 产生比较匹配中断请求信号(INTTT0CC1)。此时, 如果一个计数操作是向下计数, 则 16 位计数器被清除为 0000H。

第十章 16 位间隔定时器M (TMM)

V850ES/JG3-U 和 V850ES/JH3-U 单片机内置四路 TMM 通道 (TMMn)。

10.1 概述

定时器 TMMn 具有如下特征：

- 间隔功能
- 8 路时钟可选
- 16 位计数器 × 1
(定时器计数操作过程中, 不能读取 16 位计数器。)
- 比较寄存器 × 1
(定时器计数操作过程中, 不能写入比较寄存器。)
- 比较匹配中断 × 1

TMMn 只支持清零并启动模式。不支持自由运行定时器模式。

备注 n = 0 至 3

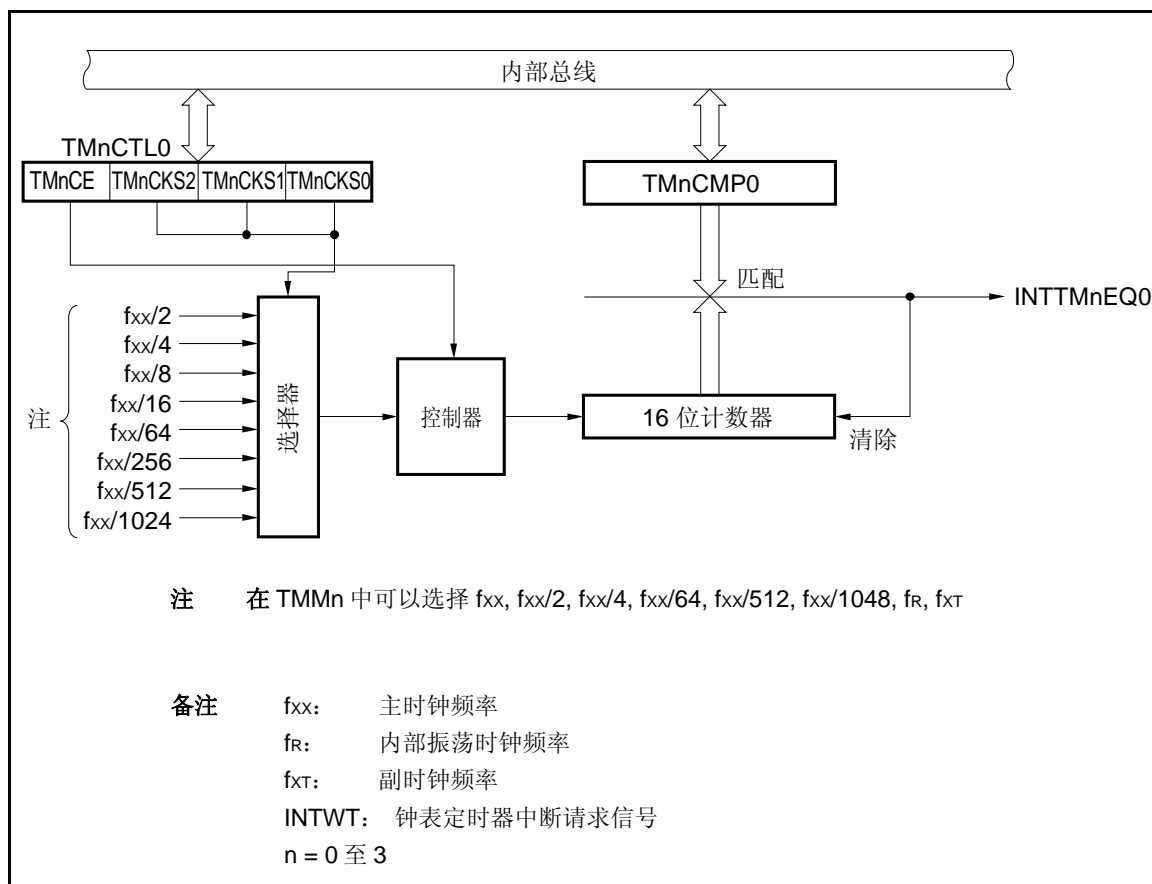
10.2 配置

TMMn 包括下列硬件：

表 10-1. TMMn 的配置

项目	配置
定时器寄存器	16 位计数器
寄存器	TMMn 比较寄存器 0 (TMnCMP0)
控制寄存器	TMMn 控制寄存器 0 (TMnCTL0)

图 10-1. TMMn 框图



(1) 16 位计数器

该计数器是 16 位计数器，对内部时钟进行计数。

该计数器不可以被读写。

(2) TMMn 比较寄存器 0 (TMnCMP0)

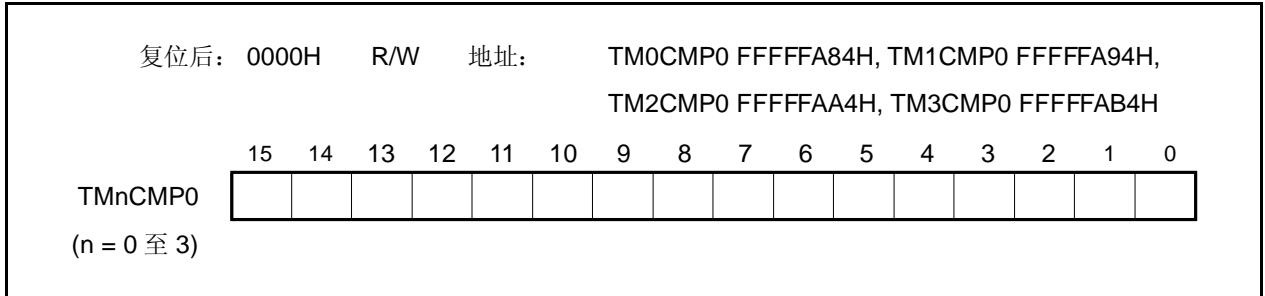
TMnCMP0 寄存器是一个 16 位的比较寄存器。

该寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

随时可以用软件将相同的值写入 TMnCMP0 寄存器。

当 TMnCTL0.TMnCE 位 = 1 时，禁止重写 TMnCMP0 寄存器。



10.3 寄存器

(1) TMMn 控制寄存器 (TMnCTL0)

TMnCTL0 寄存器是 8 位寄存器，用于控制定时器 TMMn 的操作。

该寄存器可以按字节读写或按位读写。

复位输入将该寄存器设为 00H。

随时可以用软件将相同的值写入 TMnCTL0 寄存器。

备注 n = 0 至 3

复位后: 00H R/W 地址: TMOCTL0 FFFFA80H, TM1CTL0 FFFFA90H,
TM2CTL0 FFFFAA0H, TM3CTL0 FFFFB0H

	<7>	6	5	4	3	2	1	0
TMnCTL0 (n = 0 至 3)	TMnCE	0	0	0	0	TMnCKs2	TMnCKs1	TMnCKs0

TMnCE	内部时钟操作使能/禁止 说明
0	TMMn 操作禁止 (16 位计数器异步复位) 停止操作时钟的供给。
1	TMMn 操作使能。开始供给操作时钟。 TMMn 操作启动。

使用TMnCE位对TMMn的内部时钟控制和内部电路重置是异步执行的。
当TMnCE位被清除为0时，TMMn 的内部时钟被禁止 (固定为低电平)，且16位计数器异步复位。

(m = 0)

TMmCKs2	TMmCKs1	TMmCKs0		计数时钟选择		
				fxx = 48 MHz	fxx = 32 MHz	fxx = 24 MHz
0	0	0	fxx	20.8 ns	31.3 ns	41.7 ns
0	0	1	fxx/2	41.7 ns	62.5 ns	83.3 ns
0	1	0	fxx/4	83.3 ns	125 ns	167 ns
0	1	1	fxx/64	1.33 ms	2.00 ms	2.67 ms
1	0	0	fxx/512	10.7 ms	16.0 ms	21.3 ms
1	0	1	fxx/1024	21.3 ms	32.0 ms	42.7 ms
1	1	0	fR/8	36.4 ms	36.4 ms	36.4 ms
1	1	1	fXT	30.5 ms	30.5 ms	30.5 ms

(m = 1 至 3)

TMmCKs2	TMmCKs1	TMmCKs0		计数时钟选择		
				fxx = 48 MHz	fxx = 32 MHz	fxx = 24 MHz
0	0	0	fxx/2	41.7 ns	62.5 ns	83.3 ns
0	0	1	fxx/4	83.3 ns	125 ns	167 ns
0	1	0	fxx/8	167 ns	250 ns	333 ns
0	1	1	fxx/16	333 ns	500 ns	667 ns
1	0	0	fxx/64	1.33 ms	2.00 ms	2.67 ms
1	0	1	fxx/256	5.33 ms	8.00 ms	10.7 ms
1	1	0	fxx/512	10.7 ms	16.0 ms	21.3 ms
1	1	1	fxx/1024	21.3 ms	32.0 ms	42.7 ms

- 注意事项**
1. 当 TMnCE 位=0 时，对 TMnCKs2 至 TMnCKs0 置位。
将 TMnCE 的值从 0 改变为 1 时，不可以同时设置 TMnCKs2 到 TMnCKs0 位的值。
 2. 请确保将第 3 位至第 6 位清为“0”。

备注

fxx: 主时钟频率
fR: 内部振荡时钟频率
fXT: 副时钟频率

10.4 操作

注意事项 请勿将 TMnCMP0 寄存器设置为 FFFFH。

10.4.1 间隔计时器模式

在间隔定时器模式中，如果 TMnCTL0.TMnCE 位被置为 1，则会以指定时间间隔生成中断请求信号 (INTTMnEQ0)。

图 10-2. 间隔定时器的配置

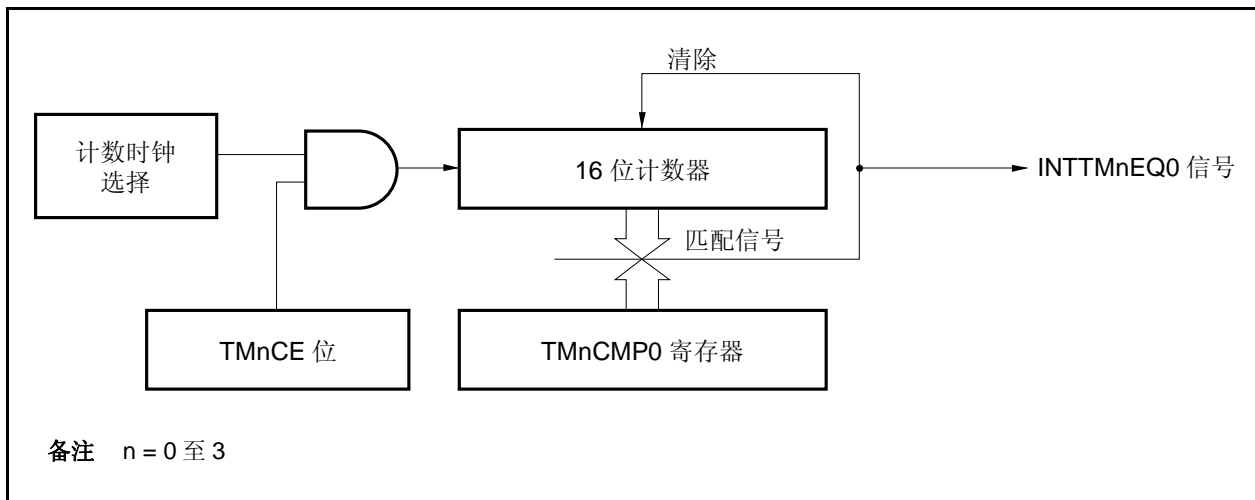
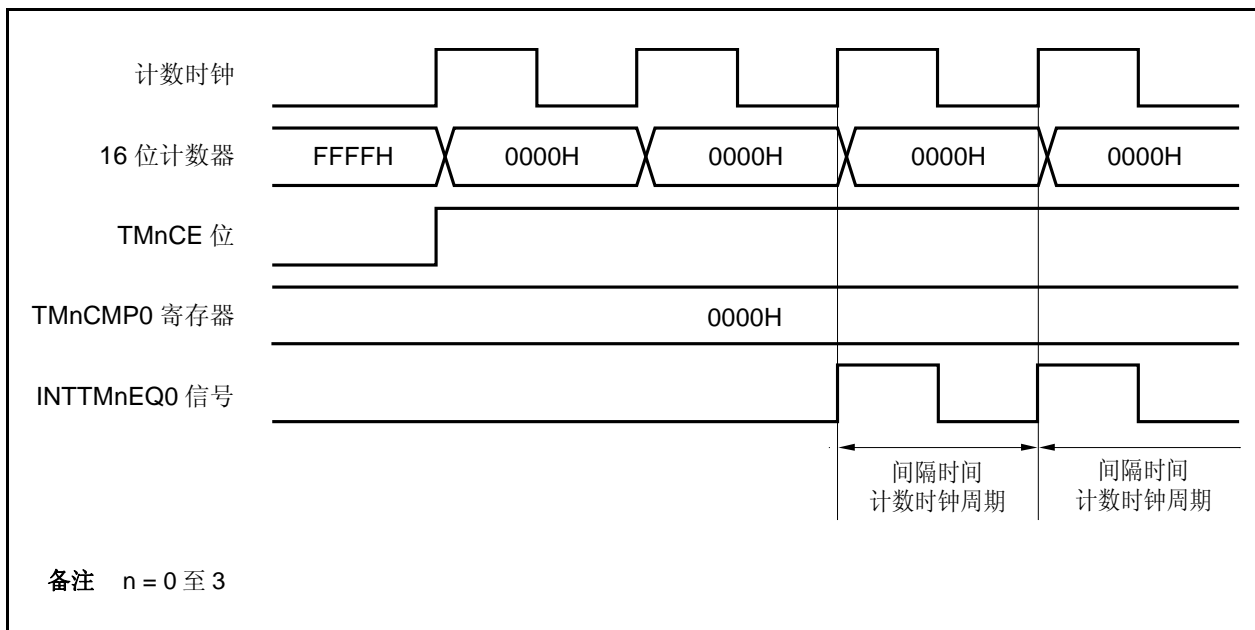


图 10-3. 间隔定时器工作的基本时序



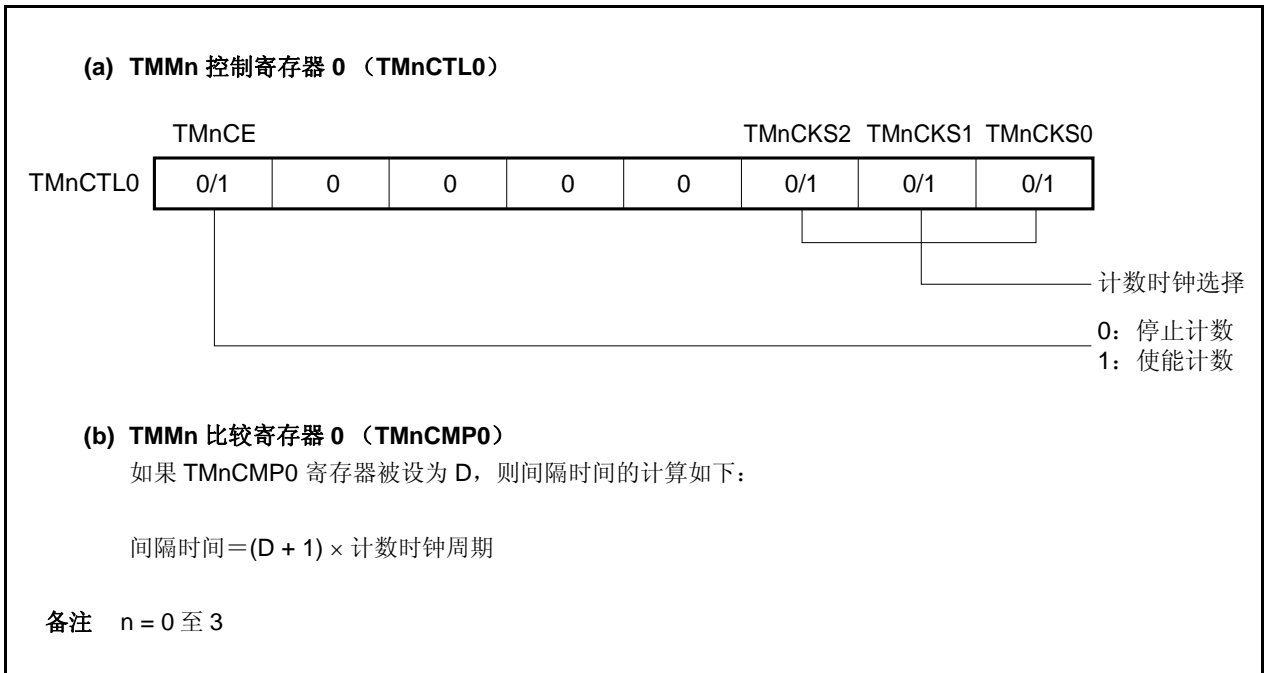
当 TMnCE 位被置为 1 时，16 位计数器的值由计数时钟同步从 FFFFH 清零为 0000H，计数器开始计数。

当 16 位计数器的计数值和 TMnCMP0 寄存器的数值相匹配时，16 位计数器清零为 0000H，同时生成一个比较匹配中断请求信号(INTTMnEQ0)。

时间间隔可用下述表达式计算。

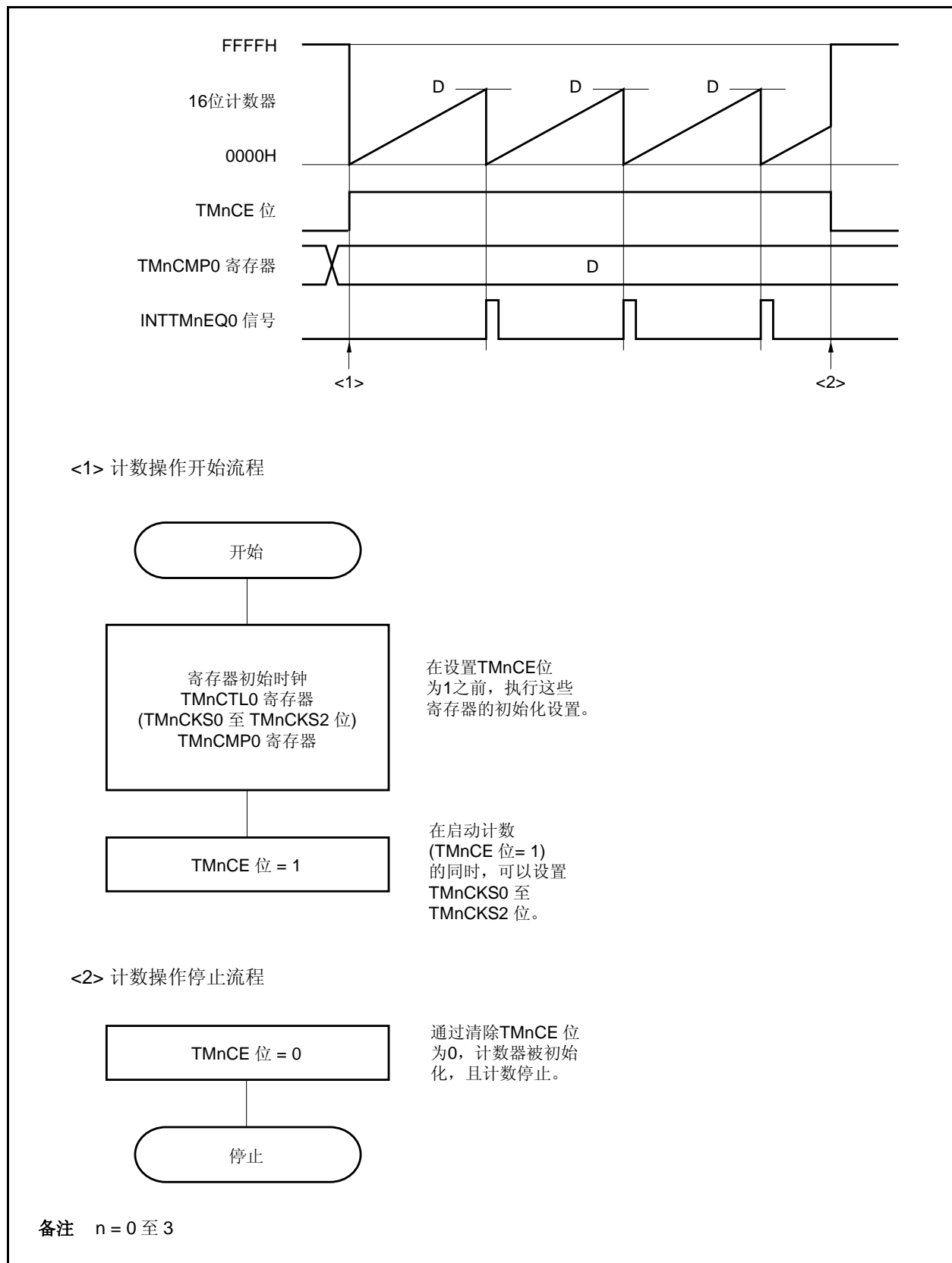
$$\text{时间间隔} = (\text{TMnCMP0 寄存器的设定值} + 1) \times \text{计数时钟周期}$$

图 10-4. 间隔定时器模式操作时的寄存器设置



(1) 间隔定时器模式操作流程

图 10-5. 间隔定时器模式下的软件处理流程

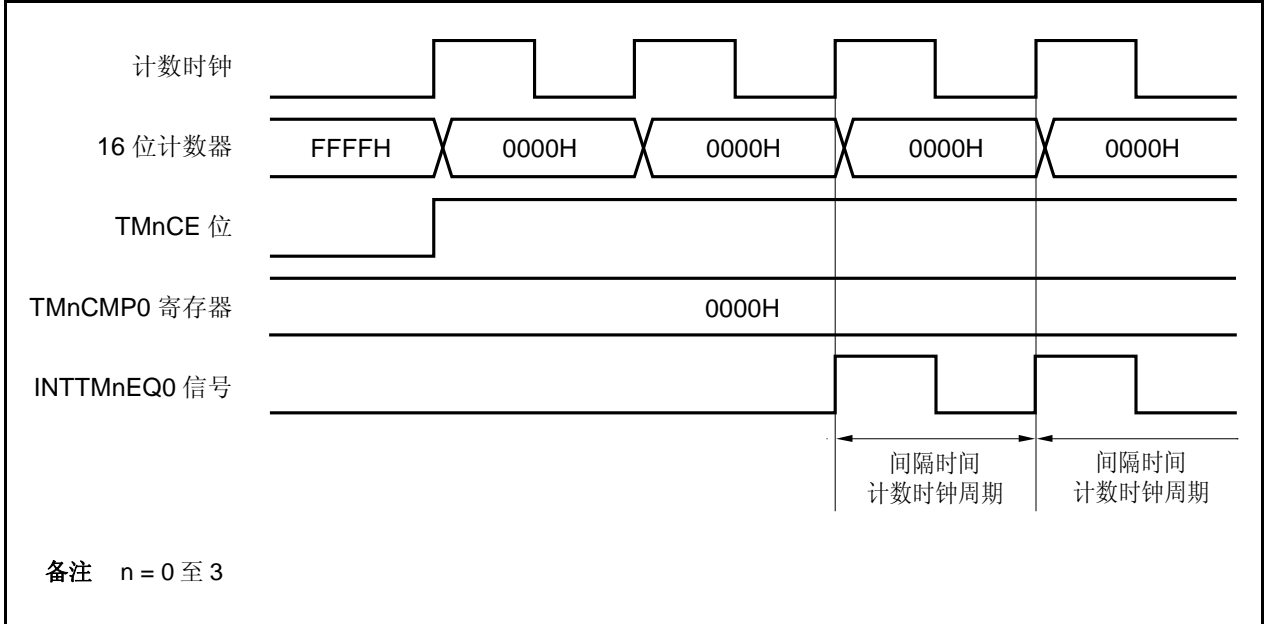


(2) 间隔定时器模式操作时序

注意事项 请勿将 TMnCMP0 寄存器设置为 FFFFH。

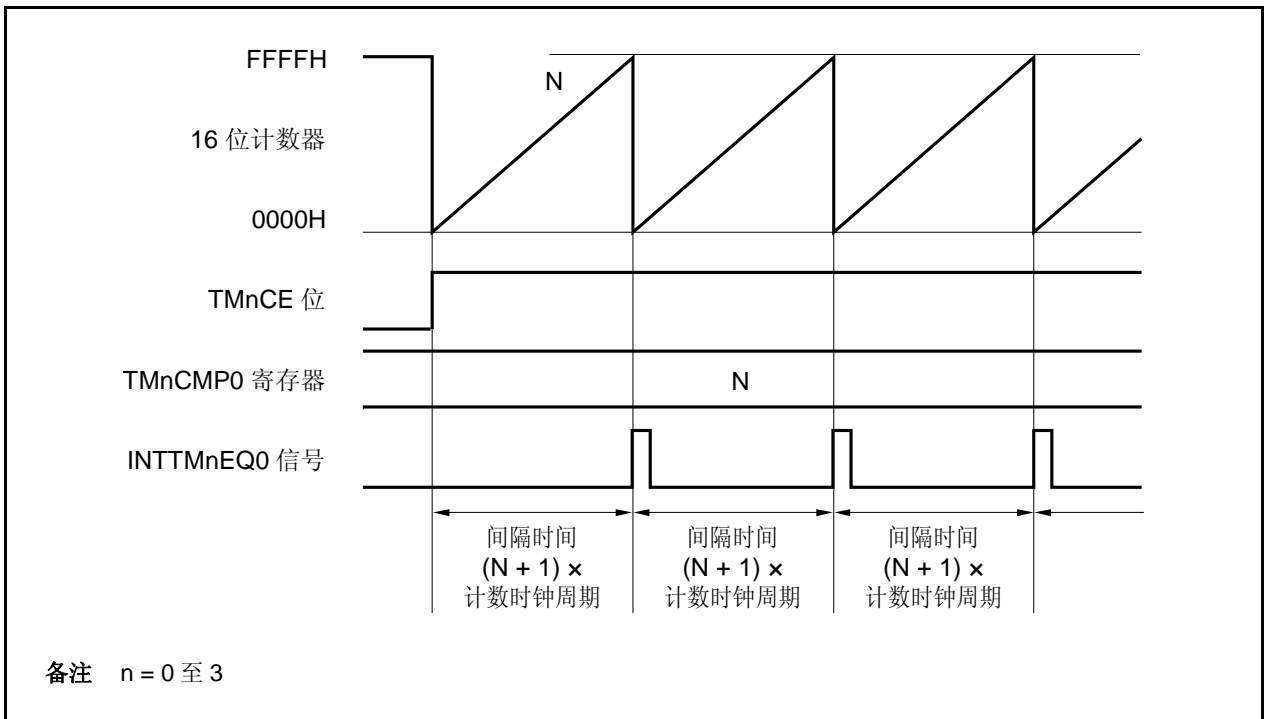
(a) TMnCMP0 寄存器设置为 0000H 时的操作

如果 TMnCMP0 寄存器被清为 0000H，那么每个计数时钟都会产生 INTTMnEQ0 信号。
16 位计数器的值总是 0000H。



(b) TMnCMP0 寄存器被设置为 N 时的操作

如果 TMnCMP0 寄存器设置为 N，则 16 位计数器最大计数至 N。计数器在下一个计数时刻被同步清为 0000H，并产生 INTTMnEQ0 信号。



10.4.2 注意事项

(1) 根据选定的计数时钟，在 $TMnCTL0.TMnCE$ 位置为 1 后，需要 16 位计数器计数以下时间才能启动计数：

(n = 0)

选择的计数时钟	计数启动前的最大时间
f_{xx}	$2/f_{xx}$
$f_{xx}/2$	$3/f_{xx}$
$f_{xx}/4$	$6/f_{xx}$
$f_{xx}/64$	$128/f_{xx}$
$f_{xx}/512$	$1024/f_{xx}$
$f_{xx}/1024$	$2048/f_{xx}$
$f_R/8$	$16/f_R$
f_{XT}	$2/f_{XT}$

(n = 1 至 3)

选择的计数时钟	计数启动前的最大时间
$f_{xx}/2$	$4/f_{xx}$
$f_{xx}/4$	$6/f_{xx}$
$f_{xx}/8$	$12/f_{xx}$
$f_{xx}/16$	$32/f_{xx}$
$f_{xx}/64$	$128/f_{xx}$
$f_{xx}/256$	$512/f_{xx}$
$f_{xx}/512$	$1024/f_{xx}$
$f_{xx}/1024$	$2048/f_{xx}$

(2) TMMn 工作时，禁止重写 $TMnCMP0$ 和 $TMnCTL0$ 寄存器的值。

如果在 $TMnCE$ 位 = 1 时对这两个寄存器进行重写，那么操作结果将无法保证。

如果这些寄存器被错误地改写，先将 $TMnCTL0.TMnCE$ 位清除为 0，然后重置这些寄存器。

备注 n = 0 至 3

11.1 功能概述

定时器 AB1 (TAB1) 和 TMQ0 选项 (TMQOP0) 可以用作电机控制中的变频功能。它可以使用定时器 AA4 (TAA4) 来执行调谐功能, 并且当 TAB1 的值与 TAA4 的值匹配时, 可以启动 A/D 转换器的 A/D 转换。以下操作可以用于实现电机控制功能。

- 16 位精度的 6 相 PWM 输出功能
- 定时器调谐操作功能 (使用 TAA4 时可调)
- 周期设置功能 (在波峰或波谷中断期间可以改变周期)
- 比较寄存器重写: 随时重写, 批量重写, 或间隙重写 (TAB1 操作时可以选择)
- 中断及传送分选功能
- 死区时间设置功能
- A/D 转换器的 A/D 触发时序功能
- 0%输出和 100%输出
- 通过波峰中断和波谷中断可以选择 0%输出和 100%输出
- 强制输出停止功能
 - 当检测到外部引脚输入 (TOAB1OFF, TOAA1OFF) 的有效边沿时
 - 通过 A/D 转换器模块的比较器功能检测到过电压时
 - 通过时钟监视功能检测到主时钟振荡停止时

11.2 配置

电机控制功能包含以下硬件。

项目	配置
定时器寄存器	死区时间计数器
比较寄存器	TAB1 死区时间比较寄存器 (TAB1DTC 寄存器)
控制寄存器	TAB1 选项寄存器 1 (TAB1OPT1) TAB1 选项寄存器 2 (TAB1OPT2) TAB1 输入/输出控制寄存器 3 (TAB1IOC3) 高阻抗输出控制寄存器 0 (HZA0CTL0) 高阻抗输出控制寄存器 1 (HZA0CTL1)

- 使用 TAB1 (TOAB11, TOAB12, TOAB13) 的输出可以产生带死区时间的 6 相 PWM 输出。
- 6 相 PWM 输出的输出电平可以单独设置。
- 16 位定时器 / 计数器 TAB1 向上 / 向下计数产生三角波。当定时器 / 计数器下溢且发生周期匹配时, 会产生一个中断。但是, 中断产生次数被限制为最多 31 次。
- TAA4 可以和 TAB1 同时执行计数 (定时器调谐操作功能)。TAA4 可以设置为三种方式, 它可以产生 A/D 触发源 (TABTADT0) 以及两种类型的中断: TAB1 下溢中断 (INTTAB1OV) 以及周期匹配中断 (INTTAB1CC0)。

图 11-1. 电机控制的框图

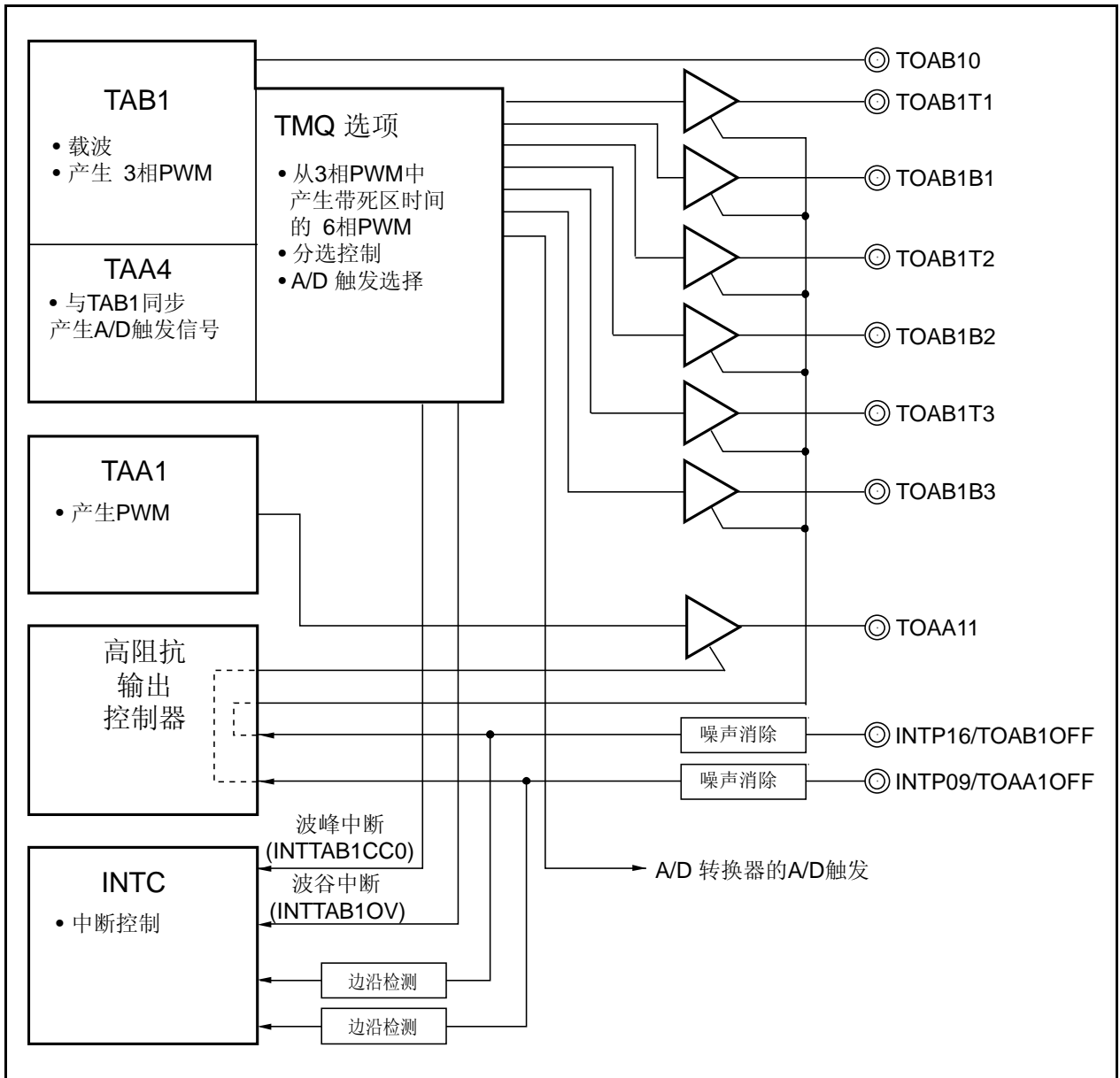
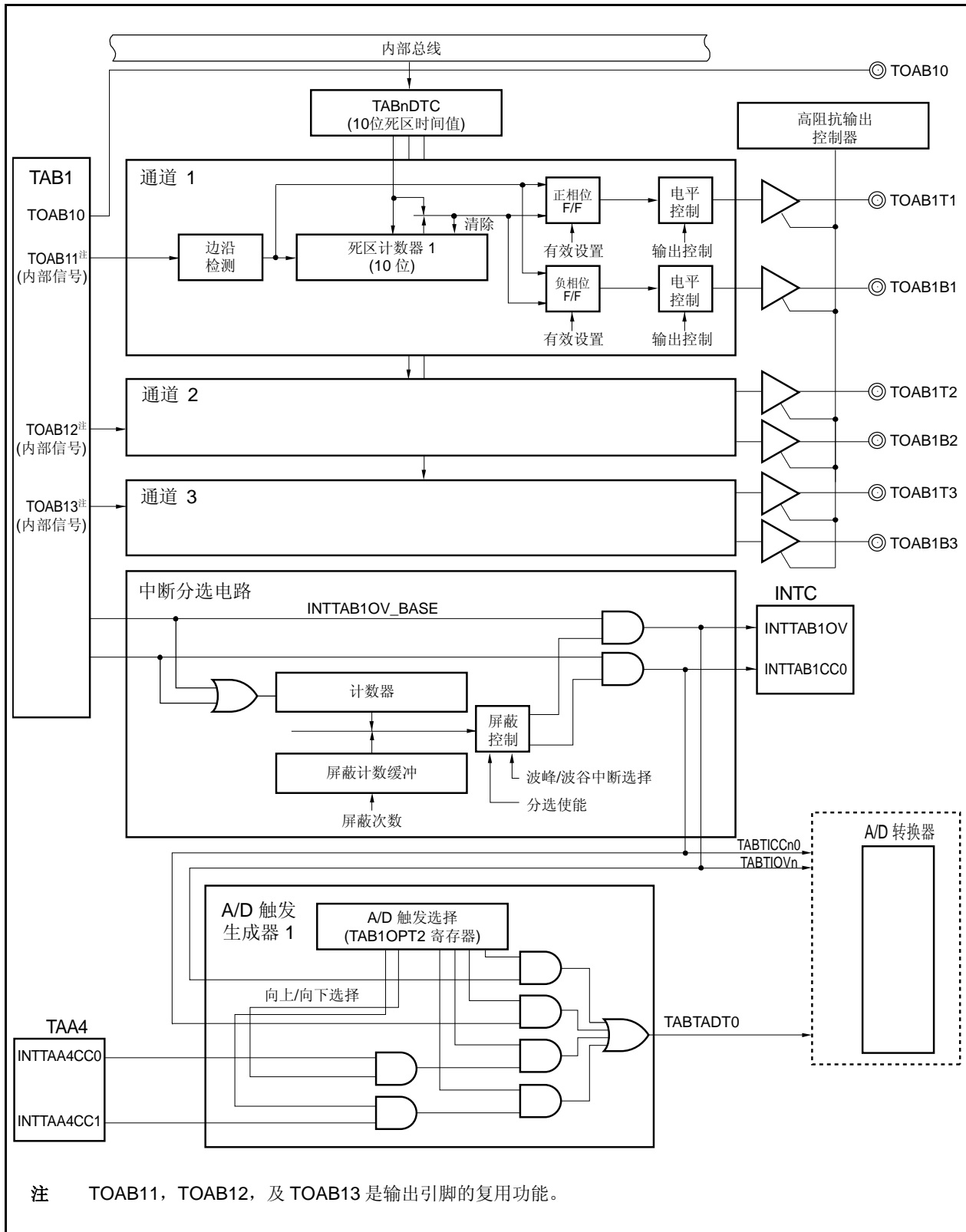


图 11-2. TMQ1 选项



(1) TAB1 死区时间比较寄存器 (TAB1DTC)

TAB1DTC 寄存器是 10 位比较寄存器，用于指定死区时间值。

当 TAB1CTL0.TAB1CE 位 = 1 时，禁止重写该寄存器。

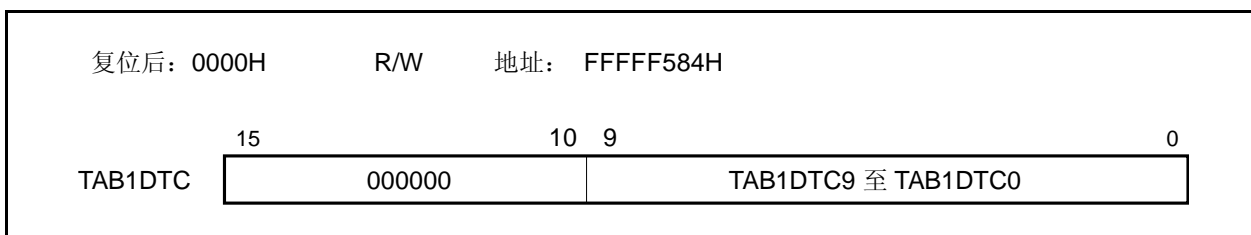
该寄存器可以按 16 位宽度进行读取或写入。

系统复位后，该寄存器被置为 0000H。

注意事项 当产生死区时间时期，将 TAB1DTC 寄存器设置为 1 或更大的值。

注意，当操作停止时 (TAB1CTL0.TAB1CE 位 = 0)，不产生死区时间，所以 TOAB1T1 至 TOAB1T3 及 TOAB1B1 至 TOAB1B3 引脚的输出电平处于默认状态。因此，为了保护系统，需要采取一些措施，比如在停止操作前，将 TOAB1T1 至 TOAB1T3 及 TOAB1B1 至 TOAB1B3 引脚设置为高阻抗状态，或者在切换到端口模式前，设置这些引脚的输出电平。

当不需要死区时间时，将 TAB1DTC 寄存器清除为 0。



(2) 死区时间计数器 1 至 3

死区时间计数器是 10 位计数器，用于计数死区时间。

在 TAB1 的 TOAB1m 输出信号的上升沿或下降沿，这些定时器被清零或向上计数，当它们的计数值与 TAB1DTC 寄存器的值匹配时，它们被清除或者停止计数。这些计数器的计数时钟相同，都由 TAB1 的 TAB1CTL0.TAB1CKS2 至 TAB1CTL0.TAB1CKS0 位设置。

- 备注**
1. 当 TAB1OPT2.TAB1DTM 位 = 1 时，操作不同。详情参见 11.4.2 (4) 自动死区时间宽度缩窄功能 (TAB1OPT2.TAB1DTM 位 = 1)。
 2. m = 1 至 3

11.3 控制寄存器

(1) TAB1 选项寄存器 1 (TAB1OPT1)

TAB1OPT1 寄存器是 8 位寄存器，用于控制由定时器 Q 选项功能所产生的中断请求信号。

当 TAB1CTL0.TAB1CE 位为 1 时，可以重写该寄存器。

根据 TAB1OPT0.TAB1CMS 位的设置，可以选择两种重写模式（批量写入模式和随时写入模式）。

该寄存器可以按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后：00H R/W 地址： FFFFF580H

	<7>	<6>	5	4	3	2	1	0
TAB1OPT1	TAB1ICE	TAB1IOE	0	TAB1ID4	TAB1ID3	TAB1ID2	TAB1ID1	TAB1ID0

TAB1ICE	波峰中断 (INTTAB1CC0 信号) 使能
0	不使用 INTTAB1CC0 信号 (不将其用作中断分选的计数信号)
1	使用 INTTAB1CC0 信号 (将其用作中断分选的计数信号)

TAB1IOE	波谷中断 (INTTAB1OV 信号) 使能
0	不使用 INTTAB1OV 信号(不将其用作中断分选的计数信号)
1	使用 INTTAB1OV 信号(将其用作中断分选的计数信号)

TAB1ID4	TAB1ID3	TAB1ID2	TAB1ID1	TAB1ID0	中断次数
0	0	0	0	0	不分选 (输出所有中断)
0	0	0	0	1	1 次屏蔽 (每2次中断输出一次)
0	0	0	1	0	2 次屏蔽 (每3次中断输出一次)
0	0	0	1	1	3 次屏蔽 (每4次中断输出一次)
:	:	:	:	:	:
1	1	1	0	0	28 次屏蔽(每29次中断输出一次)
1	1	1	0	1	29 次屏蔽(每30次中断输出一次)
1	1	1	1	0	30 次屏蔽(每31次中断输出一次)
1	1	1	1	1	31 次屏蔽(每32次中断输出一次)

(2) TAB1 选项寄存器 2 (TAB1OPT2)

TAB1OPT2 寄存器是 8 位寄存器，用于控制定时器 Q 选项功能。

当 TAB1CTL0.TAB1CE 位为 1 时，可以重写该寄存器。但是，当 TAB1CE 位为 1 时，禁止重写 TAB1DTM 位。可以重写相同的值。

该寄存器可以按字节读写或按位读写。

复位输入将该寄存器设为 00H。

(1/2)

复位后: 00H	R/W	地址: FFFFF581H																
TAB1OPT2	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;"><7></td> <td style="text-align: center;"><6></td> <td style="text-align: center;"><5></td> <td style="text-align: center;"><4></td> <td style="text-align: center;"><3></td> <td style="text-align: center;"><2></td> <td style="text-align: center;"><1></td> <td style="text-align: center;"><0></td> </tr> <tr> <td style="text-align: center;">TAB1RDE</td> <td style="text-align: center;">TAB1DTM</td> <td style="text-align: center;">TAB1ATM3</td> <td style="text-align: center;">TAB1ATM2</td> <td style="text-align: center;">TAB1AT3</td> <td style="text-align: center;">TAB1AT2</td> <td style="text-align: center;">TAB1AT1</td> <td style="text-align: center;">TAB1AT0</td> </tr> </table>	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	TAB1RDE	TAB1DTM	TAB1ATM3	TAB1ATM2	TAB1AT3	TAB1AT2	TAB1AT1	TAB1AT0	
<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>											
TAB1RDE	TAB1DTM	TAB1ATM3	TAB1ATM2	TAB1AT3	TAB1AT2	TAB1AT1	TAB1AT0											
TAB1RDE	传输分选 使能																	
0	不分选传输（每次波峰和波谷时都产生传输时序）																	
1	分选传输的间隔和 TAB1OPT1 寄存器设置的中断分选相同。																	
TAB1DTM	死区时间计数器操作模式选择 (m = 1 至 3)																	
0	如果TAB1的TOAB1m输出是窄间隔（TOAB1m输出宽度<死区时间宽度），死区时间计数器在正常向上计数，则死区时间计数器清零并再次向上计数。																	
1	如果TAB1的TOAB1m输出是窄间隔（TOAB1m输出宽度<死区时间宽度），死区时间计数器在正常向上计数，则死区时间计数器向下计数，且死区控制宽度自动变窄。																	
在定时器运行期间禁止重写TAB1DTM位。如果误写，将TAB1CE位清零来停止定时器，重新设置TAB1DTM位。																		

注意事项 1. 当使用中断分选时（TAB1OPT1.TAB1ID4 至 TAB1OPT1.TAB1ID0 设置为 00000 以外的值），请确保将 TAB1RDE 位设置为 1。
这意味着中断的产生和转移发生在同一时刻。不能单独设置中断和转移。如果中断和转移是分别设置的（TAB1RDE 位 = 0），则转移无法正常执行。

2. 当产生死区时间时，将 TAB1DTC 设置为 1 或更大的值。
注意，当操作停止时（TAB1CTL0.TAB1CE 位 = 0），不产生死区时间，所以 TOAB1T1 至 TOAB1T3 及 TOAB1B1 至 TOAB1B3 引脚的输出电平处于默认状态。因此，为了保护系统，采取一些措施，比如在停止操作前将 TOAB1T1 至 TOAB1T3 及 TOAB1B1 至 TOAB1B3 引脚设置为高阻抗状态，或者在切换到端口模式之前，设置这些引脚的输出电平。
当不需要死区时间时，将 TAB1DTC 寄存器设置为 0。

TAB1ATM3	TAB1ATM3 模式选择
0	当死区时间计数器向上计数时，将INTTAA4CC1中断输出作为A/D触发信号（TABTADT0）。
1	当死区时间计数器向上计数时，将INTTAA4CC1中断输出作为A/D触发信号（TABTADT0）。

TAB1ATM2	TAB1ATM2 模式选择
0	当死区时间计数器向上计数时，将INTTAA4CC0中断输出作为A/D触发信号（TABTADT0）。
1	当死区时间计数器向上计数时，将INTTAA4CC0中断输出作为A/D触发信号（TABTADT0）。

TAB1AT3 ^注	A/D 触发输出控制 3
0	禁止将INTTAA4CC1中断输出为A/D触发信号（TABTADT0）
1	使能将INTTAA4CC1中断输出为A/D触发信号（TABTADT0）

TAB1AT2 ^注	A/D 触发输出控制 2
0	禁止将INTTAA4CC0中断输出为A/D触发信号（TABTADT0）
1	使能将INTTAA4CC0中断输出为A/D触发信号（TABTADT0）

TAB1AT1 ^注	A/D 触发输出控制 1
0	禁止将 INTTAB1CC0中断输出为A/D触发信号（TABTADT0）
1	使能将 INTTAB1CC0中断输出为A/D触发信号（TABTADT0）

TAB1AT0 ^注	A/D 触发输出控制 0
0	禁止将 INTTAB1OV 中断（波谷中断）输出为A/D触发信号（TABTADT0）
1	使能将INTTAB1OV 中断（波谷中断）输出为A/D触发信号（TABTADT0）

注 关于 TAB1AT3 至 TAB1AT0 位的设置，参见第 15 章 A/D 转换器。

(3) TAB1 输入/输出控制寄存器 3 (TAB1IOC3)

TAB1IOC3 寄存器是 8 位寄存器，用于控制定时器 Q 选项功能输出。

若要从 TOAB1Tm 引脚输出，应先将 TAB1IOC0.TAB1OEm 位设置为 1，然后对 TAB1IOC3 寄存器进行设置。

只有在 TAB1CTL0.TAB1CE 位为 0 时，才可以重写 TAB1IOC3 寄存器。

TAB1CTL0.TAB1CE 位为 1 时，禁止对 TAB1IOC3 寄存器中的各个位进行重写；但是，当 TAB1CTL0.TAB1CE 位为 1 时，TAB1IOC3 寄存器的每一位都可以写入相同的值。

该寄存器可以按字节读写或按位读写。

复位输入将该寄存器设为 A8H。

注意事项 如果定时器用于 6 相 PWM 输出模式以外的模式，那么将 TAB1IOC3 寄存器设置为复位值 (A8H)。

备注 1. 通过 TAB1IOC0 寄存器设置 TOAB1Tm 引脚的输出电平。

2. m = 1 至 3

复位后: A8H	R/W	地址: FFFFF582H						
TAB1IOC3	<7>	<6>	<5>	<4>	<3>	<2>	1	0
	TAB1OLB3	TAB1OEB3	TAB1OLB2	TAB1OEB2	TAB1OLB1	TAB1OEB1	0	0
	TAB1OLBm	TOAB1Bm 引脚输出电平的设置 (m = 1 至 3)						
	0	禁止 TOAB1Bm 引脚输出电平的反转						
	1	使能 TOAB1Bm 引脚输出电平的反转						
	TAB1OEBm	TOAB1Bm 引脚输出 (m = 1 至 3)						
	0	禁止 TOAB1Bm 引脚输出 • 当 TAB1OLBm 位 = 0，从 TOAB1Bm 引脚输出低电平 • 当 TAB1OLBm 位 = 1，从 TOAB1Bm 引脚输出高电平						
	1	使能 TOAB1Bm 引脚输出						

(a) 从 TOAB1Tm 和 TOAB1Bm 引脚输出

通过 TAB1IOC0.TAB1OLm 位和 TAB1IOC0.TAB1OEm 位，控制 TOAB1Tm 引脚输出。通过 TAB1IOC3.TAB1OLBm 位和 TAB1IOC3.TAB1OEBm 来控制 TOAB1Bm 引脚输出。

以下显示了 6 相 PWM 输出模式中各种设置的定时器输出。

图 11-3. TOAB1Tm 和 TOAB1Bm 引脚的输出控制（不带死区时间）

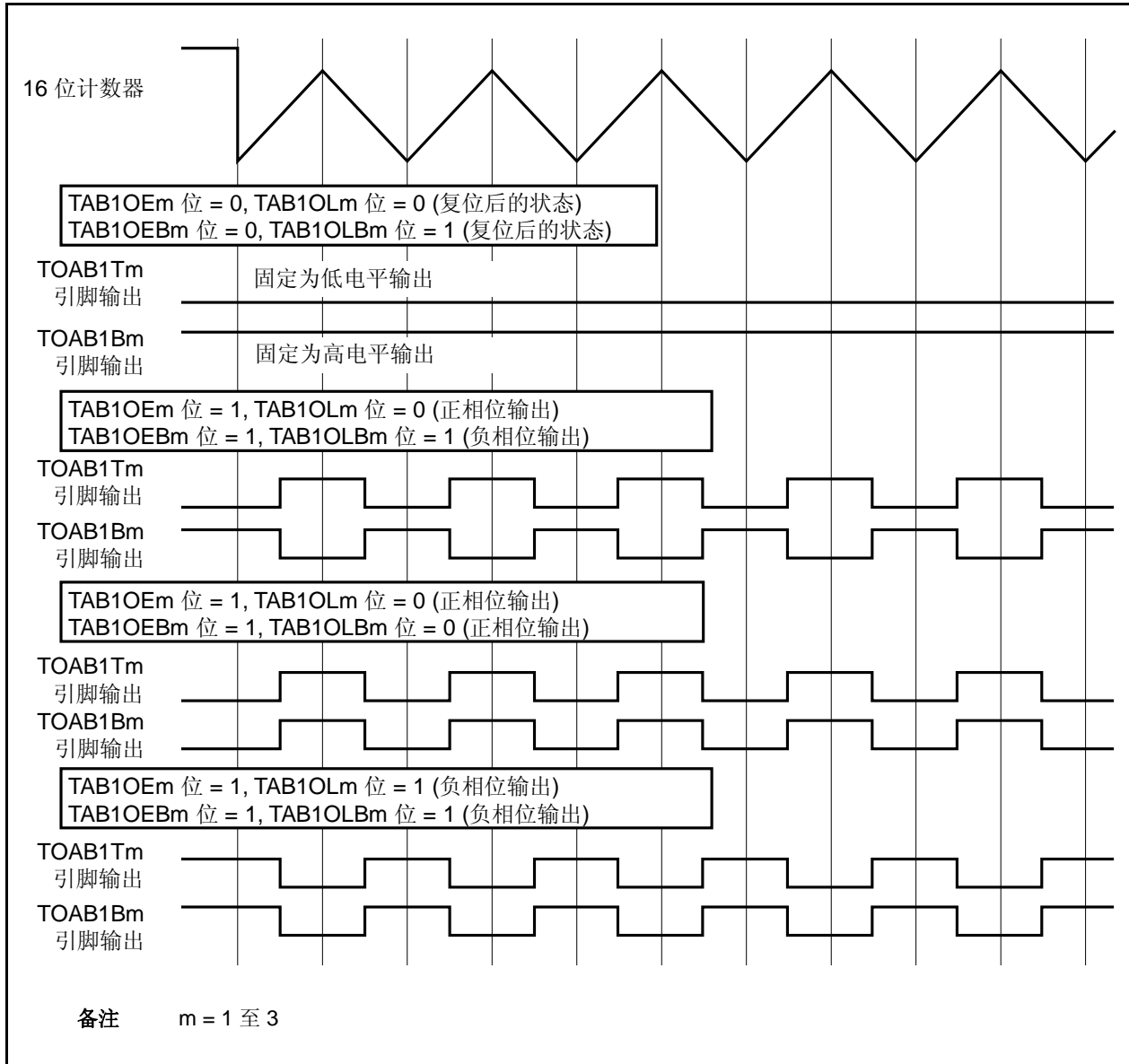


表 11-1. TOAB1Tm 引脚输出

TAB1OLm 位	TAB1OEm 位	TAB1CE 位	TOAB1Tm 引脚输出
0	0	x	低电平输出
	1	0	低电平输出
		1	TOAB1Tm 正相输出
1	0	x	高电平输出
	1	0	高电平输出
		1	TOAB1Tm 负相输出

备注 m = 1 到 3

表 11-2. TOAB1Bm 引脚输出

TAB1OLBm 位	TAB1OEBm 位	TAB1CE 位	TOAB1Bm 引脚输出
0	0	x	低电平输出
	1	0	低电平输出
		1	TOAB1Bm 正相输出
1	0	x	高电平输出
	1	0	高电平输出
		1	TOAB1Bm 负相输出

备注 m = 1 到 3

(6) 高阻抗输出控制寄存器 0, 1 (HZA0CTL0, HZA0CTL1)

HZA0CTL0 和 HZA0CTL1 寄存器是 8 位寄存器，用于控制输出缓冲的高阻抗状态。

这些寄存器可以按字节读写或按位读写。但是，HZA0DCFn 位为只读位，不能对其写入。

不能进行 16 位访问。

复位输入将这些寄存器置为 00H。

总是可以通过软件向 HZA0CTLn 寄存器写入相同的数值。

以下显示了检测因素和控制寄存器之间的关系。

需要进行高阻抗控制的引脚	高阻抗控制因素	控制寄存器
	外部引脚	
当 TOAB1T1 至 TOAB1T3 输出时 当 TOAB1B1 至 TOAB1B3 输出时	TOAB1OFF/INTP16	HZA0CTL0
当 TOAA11 输出时	TOAA1OFF/INTP09	HZA0CTL1

注意事项 只有当目标端口被指定为上表中所显示的目标引脚功能时，才可以执行高阻抗控制。

备注 n = 0, 1

复位后: 00H R/W 地址: HZA0CTL0 FFFFF590H, HZA0CTL1 FFFFF591H

	<7>	<6>	5	4	<3>	<2>	1	<0>
HZA0CTLn (n = 0, 1)	HZA0DCEn	HZA0DCMn	HZA0DCNn	HZA0DCPn	HZA0DCTn	HZA0DCCn	0	HZA0DCFn

HZA0DCEn	高阻抗输出控制
0	禁止高阻抗输出控制操作。引脚用作输出引脚。
1	使能高阻抗输出控制操作。

HZA0DCMn	通过HZA0DCCn位清除高阻抗状态的条件
0	不论外部引脚输入如何，HZA0DCCn位的设置均有效。
1	当外部引脚输入保持的电平被检测为异常（有效电平）时，HZA0DCCn 位的设置无效。
当HZA0DCEn 位 = 0时，重写HZA0DCMn位。	

HZA0DCNn	HZA0DCPn	外部引脚输入沿说明
0	0	无有效边沿（禁止通过外部引脚输入设置HZA0DCFn位）
0	1	外部引脚的上升沿有效（通过上升沿输入检测到异常）
1	0	外部引脚的下降沿有效（通过下降沿输入检测到异常）
1	1	禁止设置

- 当HZA0DCEn 位 = 0时，重写HZA0DCNn 和 HZA0DCPn 位。
- 关于INTP09和INTP16引脚中断的有效边沿说明，参见23.6.2 (3) 外部中断下降沿、上升沿说明寄存器3（INTR3，INTF3）及(6) 外部中断下降沿、上升沿说明寄存器9H（INTR9、INTF9）。
- 从TOAB1OFF和TOAA1OFF引脚开始指定外部引脚的边沿。然后，为TOAB1OFF和TOAA1OFF引脚以外的其它引脚分配边沿。否则，为TOAB1OFF和TOAA1OFF引脚指定边沿时可能会检测到不确定边沿。
- 在使能操作（设置HZA0DCEn位为1）之后，有效边沿输入时执行高阻抗输出控制。当操作使能后，如果该外部引脚处于有效电平，则不执行高阻抗输出控制。

HZA0DCTn	高阻抗输出触发位
0	无操作
1	通过软件将引脚设置为高阻抗状态，且HZA0DCFn位被置为1。

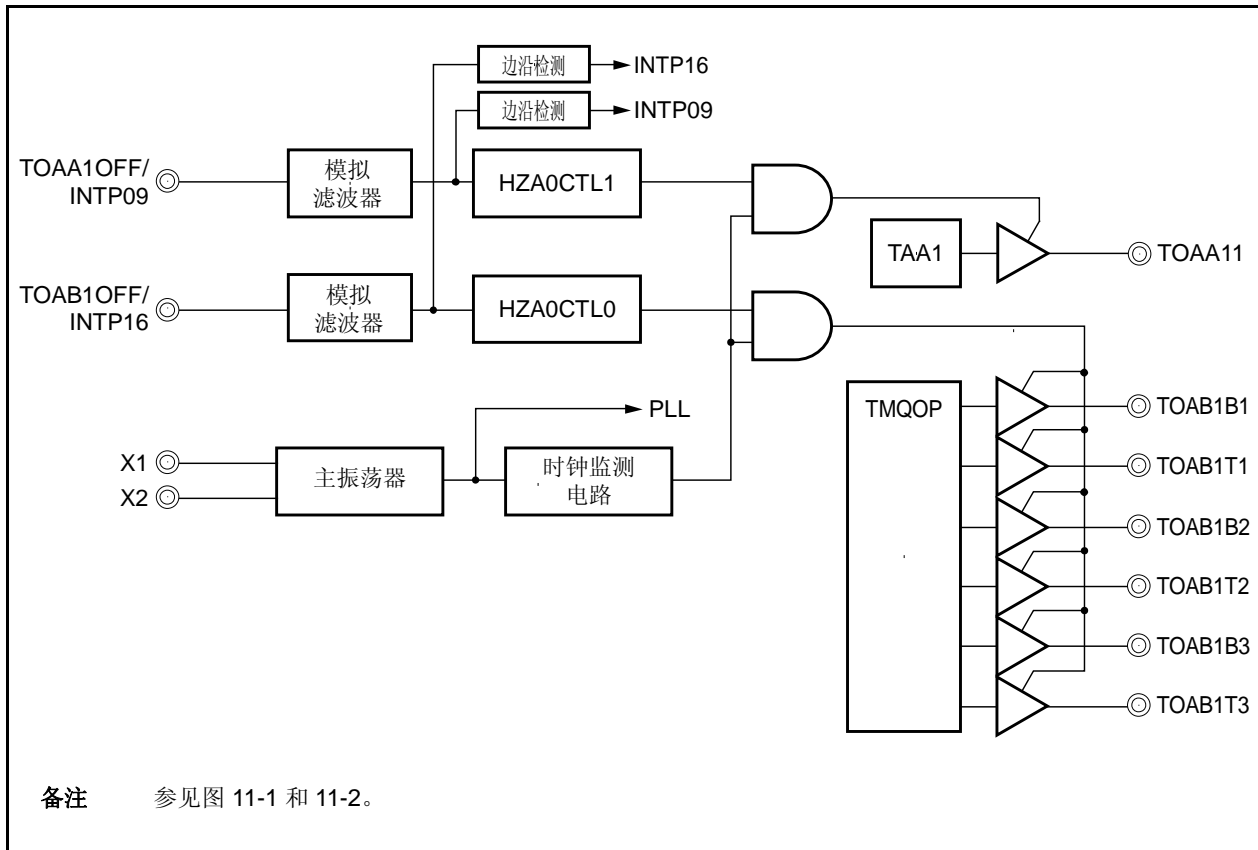
- 当表示异常的边沿从外部引脚输入（根据HZA0DCNn 和 HZA0DCPn 位的设置来检测）时，即使该位被置1，HZA0DCTn 位也无效。
- 当读取HZA0DCTn位时，该位总是0，因为它是软件触发位。
- 当 HZA0DCEn 位 = 0时，即使该位被置1，HZA0DCTn 位也无效。
- 禁止同时将HZA0DCTn 和 HZA0DCCn 位置 1。

HZA0DCCn	高阻抗输出控制清除位
0	无操作
1	已经进入高阻抗状态的引脚可以通过软件设置为使能输出，且HZA0DCFn 位被清除为 0。

- 不论外部引脚的状态如何，当HZA0DCM位=0时，引脚总是可用作输出引脚。
- 当表示异常的边沿从外部引脚输入（根据HZA0DCNn 和 HZA0DCPn 位的设置来检测）时，当 HZA0DCM 位 = 1时，即使该位被置1，HZA0DCCn 位也无效。
- 读取HZA0DCCn位时，它总是0。
- 当HZA0DCEn 位 = 1时，即使该位被置1，HZA0DCCn 位也无效。
- 禁止同时将HZA0DCTn 和 HZA0DCCn 位置 1。

HZA0DCFn	高阻抗输出状态标志
重置 (0)	表示引脚输出已经使能 <ul style="list-style-type: none"> • 当HZA0DCEn位 = 0时，该位被清0。 • 当HZA0DCEn位 = 1时，该位被清0。
置位 (1)	表示引脚进入高阻抗状态 <ul style="list-style-type: none"> • 当HZA0DCTn位 = 1时，该位被置1。 • 当表示异常的边沿从外部引脚输入（根据HZA0DCNn 和 HZA0DCPn 位的设置来检测）时，该位被置1。

图 11-4. 高阻抗输出控制器配置



(a) 设置步骤**(i) 高阻抗控制操作的设置**

<1> 设置 HZA0DCMn, HZA0DCNn, 以及 HZA0DCPn 位。

<2> 将 HZA0DCEn 位设置为 1 (使能高阻抗控制)。

(ii) 在使能高阻抗控制操作后更改设置

<1> 将 HZA0DCEn 位清除为 0 (停止高阻抗控制操作)。

<2> 更改 HZA0DCMn, HZA0DCNn, 及 HZA0DCPn 位的设置。

<3> 将 HZA0DCEn 位设置为 1 (再次使能高阻抗控制操作)。

(iii) 引脚处于高阻抗状态时恢复输出

如果 HZA0DCMn 位为 1, 则在检测到外部引脚的有效边沿后, 将 HZA0DCCn 位设置为 1 以清除高阻抗状态。但是, 除非在外部引脚的输入电平无效时设置该位, 否则将不能清除高阻抗状态。

<1> 将 HZA0DCCn 位设置为 1 (命令信号清除高阻抗状态)。

<2> 读取 HZA0DCFn 位并检测标志状态。

<3> 如果 HZA0DCFn 位为 1, 则返回<1>。必须检查外部引脚的输入电平。

如果 HZA0DCFn 位为 0, 则引脚可以用作输出引脚。

(iv) 通过软件使引脚进入高阻抗状态

当外部引脚的输入电平无效时, 必须通过软件将 HZA0DCTn 位设置为 1, 使引脚进入高阻抗状态。示例过程如下, 其中的设置并不取决于 HZA0DCMn 位的设置。

<1> 将 HZA0DCTn 位设置为 1 (高阻抗输出命令)。

<2> 读取 HZA0DCFn 位, 检测标志状态。

<3> 如果 HZA0DCFn 位为 0, 则返回<1>。必须检查外部引脚的输入电平。

如果 HZA0DCFn 位为 1, 则引脚处于高阻抗状态。

但是, 如果使用外部引脚时 HZA0DCPn 位和 HZA0DCNn 未被清除为 0, 那么在 HZA0DCTn 位被置为 1 时, 引脚将进入高阻抗状态。

备注 n = 0, 1

11.4 操作

11.4.1 系统概述

(1) 6相 PWM 输出概述

通过定时器 AB1 (TAB1) 和 TMQ 选项 (TMQOPA)，可以使用 6 相 PWM 输出模式生成 6 相 PWM 输出波形。

通过将 TAB1 的 TAB1CTL1.TAB1MD2 至 TAB1CTL1.TAB1MD0 位设为“111”，可以使能 6 相 PWM 输出模式。

TAB1 的一个 16 位计数器和四个 16 位比较寄存器用于生成基本的 3 相波形。

比较寄存器的功能如下。

TAA4 可以通过和 TAB1 执行调谐操作，产生用于 A/D 转换器的转换触发源。

比较寄存器	功能	可设置范围
TAB1CCR0 寄存器	设置周期	$0002H \leq m \leq FFFE H$
TAB1CCR1 寄存器	指定 U 相的输出宽度	$0000H \leq i \leq m + 1$
TAB1CCR2 寄存器	指定 V 相的输出宽度	$0000H \leq j \leq m + 1$
TAB1CCR3 寄存器	指定 W 相的输出宽度	$0000H \leq k \leq m + 1$

备注

- m = TAB1CCR0 寄存器的设置值
- i = TAB1CCR1 寄存器的设置值
- j = TAB1CCR2 寄存器的设置值
- k = TAB1CCR3 寄存器的设置值

死区时间间隔从基本 3 相波形中生成，该波形通过使用三个 10 位死区时间计数器和一个比较寄存器来生成，其中比较寄存器用于创建与基本 3 相波形相位相反的波形。然后生成 6 相 PWM 输出波形 (U, \bar{U} , V, \bar{V} , W, 以及 \bar{W})。

用于生成基本 3 相波形的 16 位计数器向上计数或向下计数。操作开始后，该计数器向上计数。当它的计数值与 TAB1CCR0 寄存器中设置的周期匹配时，计数器开始向下计数。当计数值等于 0001H 时，计数器将再次开始向上计数。这意味着载波周期等于两倍的“TAB1CCR0 寄存器设置值+ 1”的值。

生成死区时间间隔的 10 位死区时间计数器 1 至 3 向上计数。因此，TAB1 死区时间比较寄存器 (TAB1DTC) 中的设置值被用作死区时间值。由于使用了三个计数器，因此可以在 U, V, W 三相独立产生死区时间。但是，由于只有一个指定死区时间值 (TAB1DTC) 的寄存器，因此在三个相中使用相同的死区时间值。

图 11-5. 6 相 PWM 输出模式概况

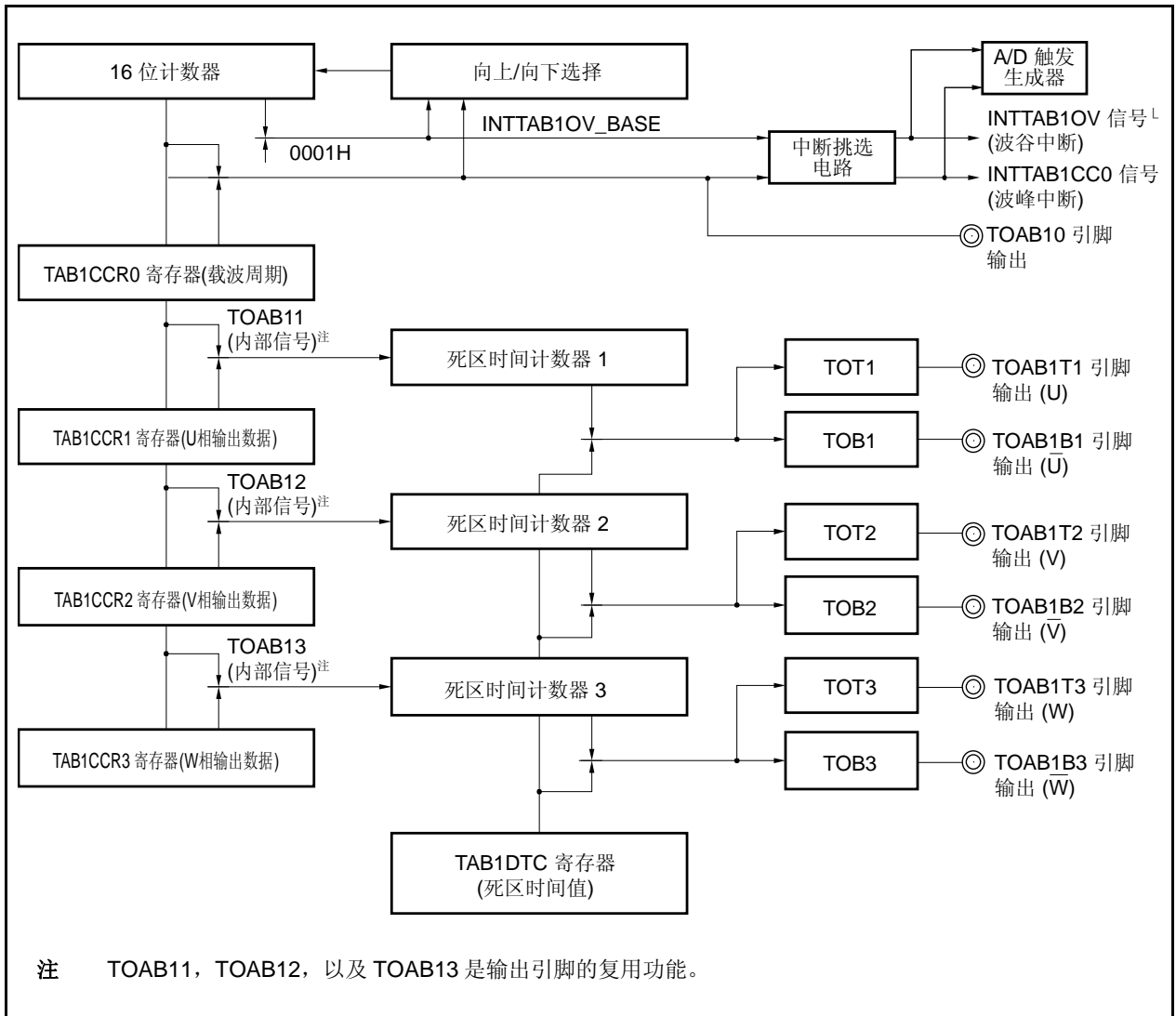
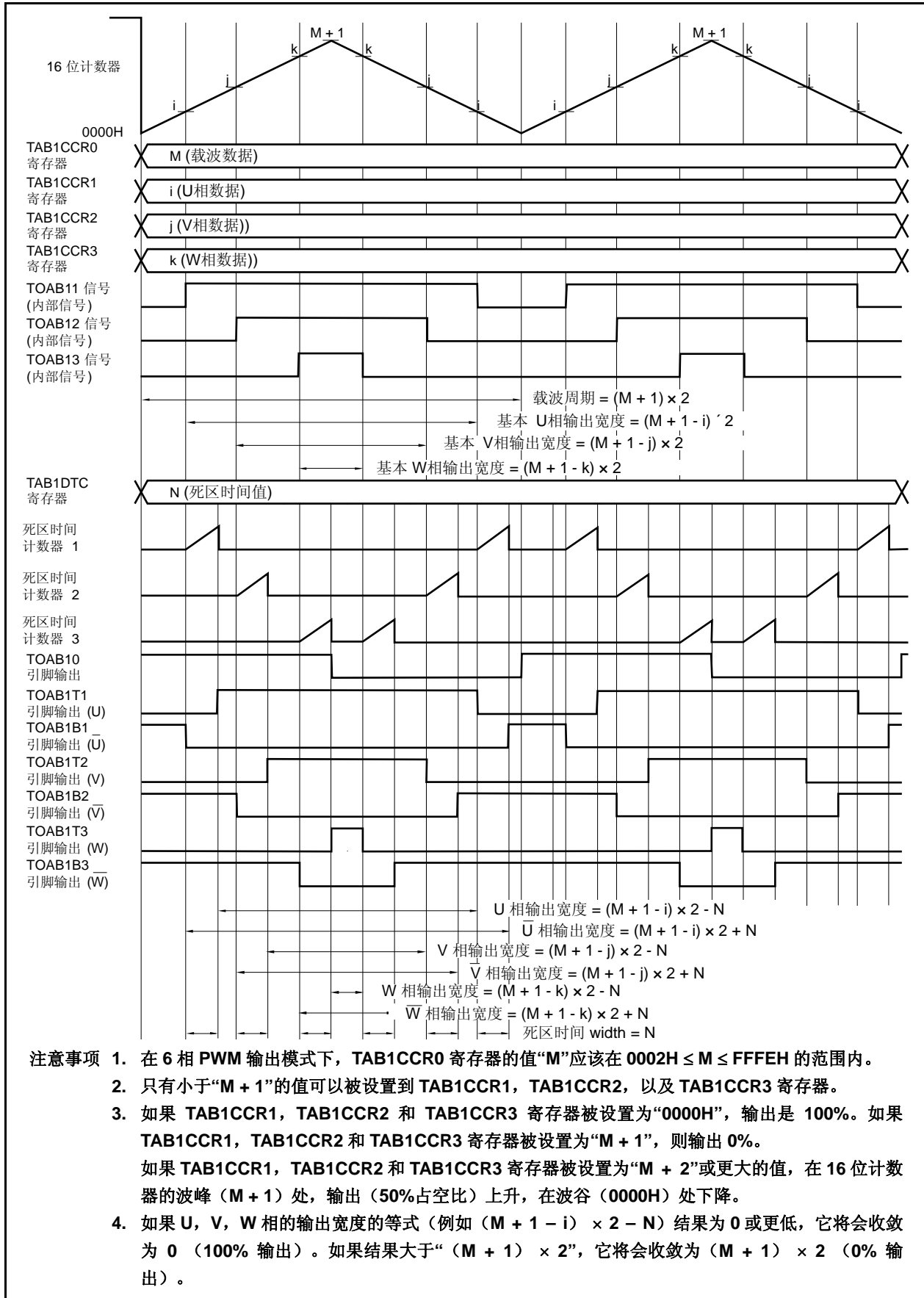


图 11-6. 6 相 PWM 输出模式时序图



(2) 中断请求

可用的中断请求类型有两种：INTTAB1CC0（波峰中断）信号和 INTTAB1OV（波谷中断）信号。

INTTAB1CC0 和 INTTAB1OV 可以使用 TAB1OPT1 寄存器来分选。

关于分选中断的详情参见 **11.4.3 中断分选功能**。

- INTTAB1CC0（波峰中断）信号： 该中断信号表示向上计数的 16 位计数器的值与 TAB1CCR0 寄存器的值匹配
- INTTAB1OV（波谷中断）信号： 该中断信号表示向下计数的 16 位计数器的值等于 0001H

(3) 定时器操作期间重写寄存器

以下寄存器都有缓冲寄存器，可以用随时重写模式，批量重写模式或间隙批量重写模式进行重写。

相关单元	寄存器
定时器 AA1	TAA1 捕捉 / 比较寄存器 0 (TAA1CCR0) TAA1 捕捉 / 比较寄存器 1 (TAA1CCR1)
定时器 AB1	TAB1 捕捉 / 比较寄存器 0 (TAB1CCR0) TAB1 捕捉 / 比较寄存器 1 (TAB1CCR1) TAB1 捕捉 / 比较寄存器 2 (TAB1CCR2) TAB1 捕捉 / 比较寄存器 3 (TAB1CCR3)
定时器 Q1 选项	TAB1 选项寄存器 1 (TAB1OPT1)

关于比较寄存器传送功能的详情参见 **11.4.4 使用传送功能重写寄存器的操作**

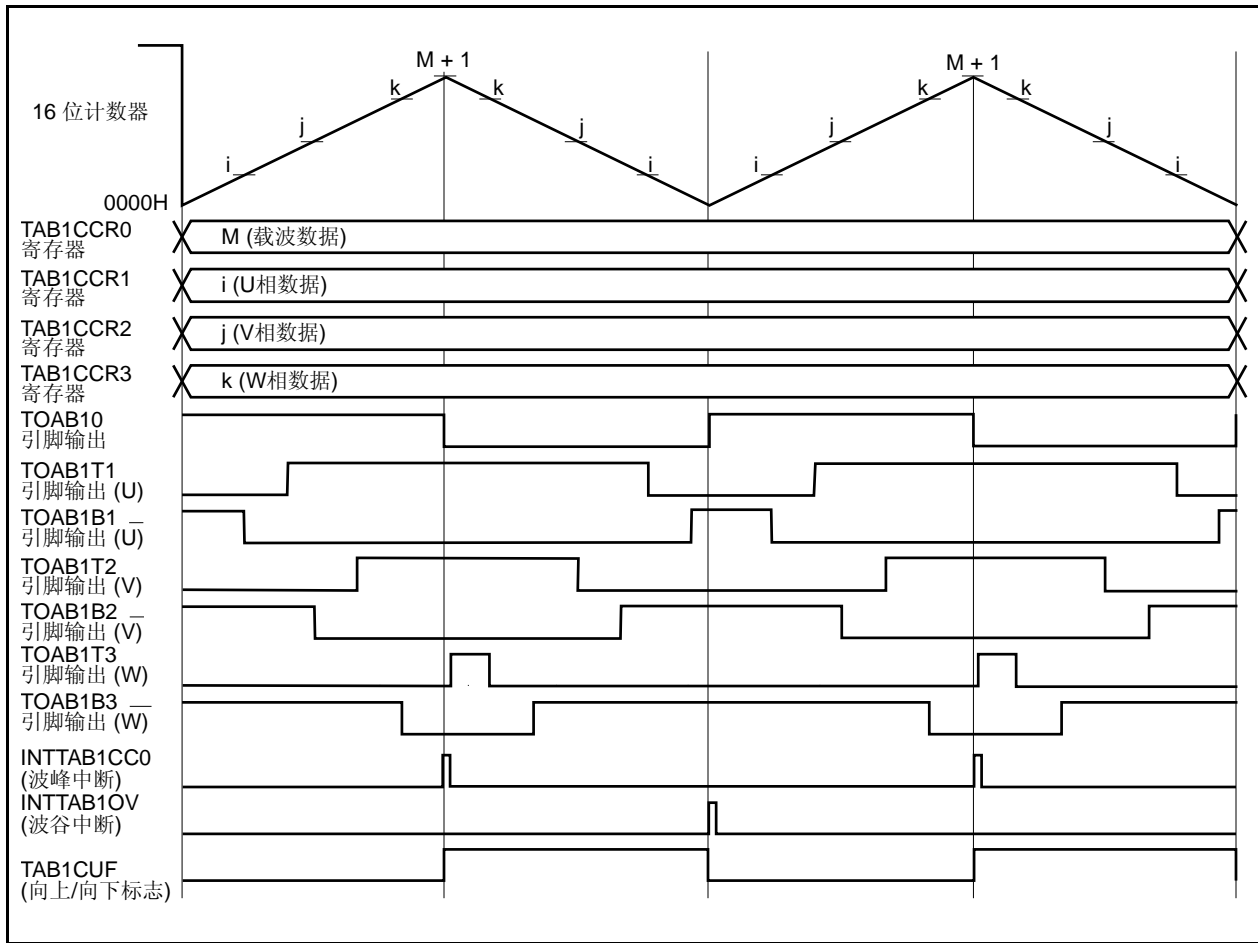
(4) 16 位计数器的向上计数 / 向下计数操作

通过使用 TAB1 选项寄存器 0 (TAB1OPT0) 的 TAB1CUF 位可以检查 16 位计数器的操作状态。

TAB1CUF 位的状态	16 位计数器的状态	16 位计数器值的范围
TAB1CUF 位 = 0	向上计数	0000H – m
TAB1CUF 位 = 1	向下计数	(m + 1) – 0001H

备注 m = TAB1CCR0 寄存器的设置值

图 11-7. 中断和向上 / 向下标志



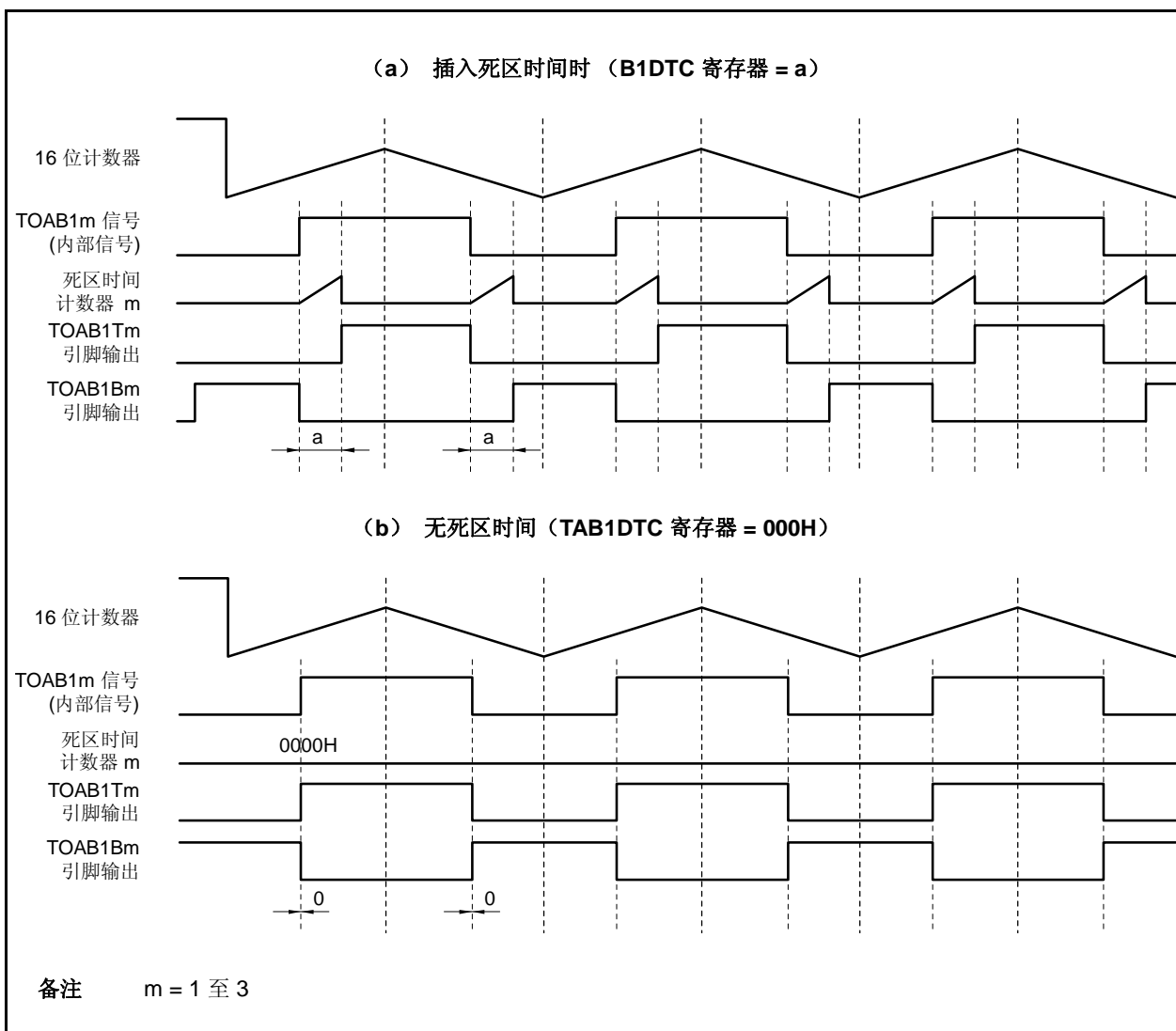
11.4.2 死区时间控制（产生负相位波形信号）

(1) 死区时间控制原理

在 6 相 PWM 输出模式中，比较寄存器 1 至 3（TAB1CCR1， TAB1CCR2 和 TAB1CCR3）用于设置占空比，而比较寄存器 0（TAB1CCR0）用于设置周期。通过设置这四个寄存器并启动 TAB1 操作，可以生成三种占空比可变的 PWM 输出波形（基本 3 相波形）。这三种 PWM 输出波形被输入到定时器 Q 选项单元（TMQOP）中，生成带有死区时间的反转信号以组成三组（六个）PWM 波形。

TMQOP 单元由三个 10 位计数器（死区时间计数器 1 至 3）组成，这三个计数器与 TAB1 的计数时钟同步，通过 TAB1 死区时间比较寄存器（TAB1DTC）指定死区时间。如果 TAB1DTC 寄存器的值被设置为“a”，那么死区时间的值为“a”，此间隔“a”将会出现在正相位波形和负相位波形之间。

图 11-8. 带有死区时间的 PWM 输出波形（1）



(2) 0%/100%的 PWM 输出

V850ES/JG3-U 和 V850ES/JH3-U 可以实现 0%和 100%的 PWM 波形输出。

从 TOAB1Tm 引脚输出的连续低电平作为 0%波形输出。从 TOAB1Tm 引脚中输出的连续高电平作为 100%波形输出。

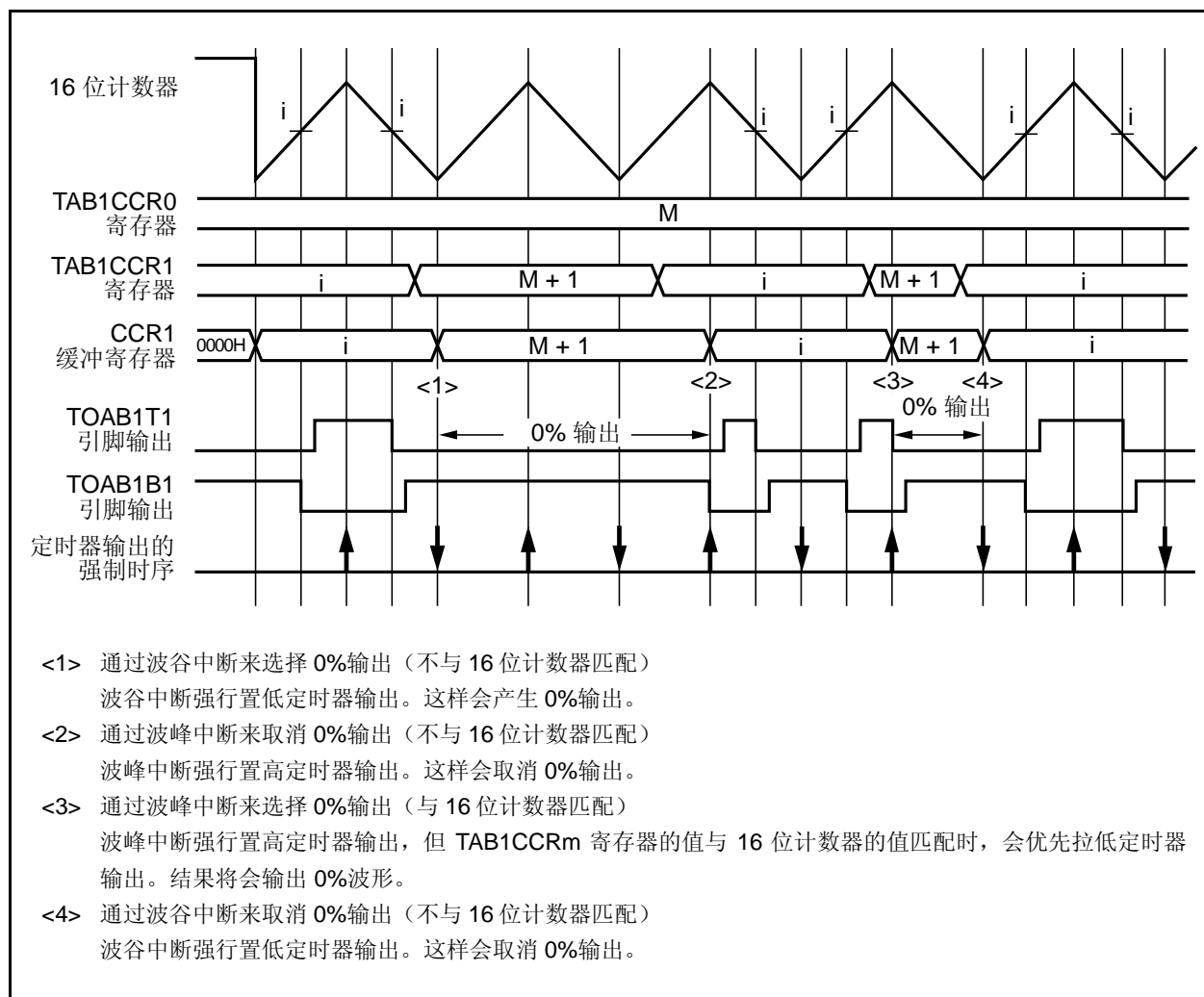
当 TAB1CCR0 寄存器 = M 时，将 TAB1CCRm 寄存器设为“M + 1”，可以输出 0%波形。

将 TAB1CCRm 寄存器设为“0000H”，可以输出 100%波形。

定时器进行操作时，可以对 TAB1CCRm 寄存器进行重写，可以在波峰中断（INTTAB1CC0）和波谷中断（INTTAB1OV）时选择 0%波形输出或 100%波形输出。

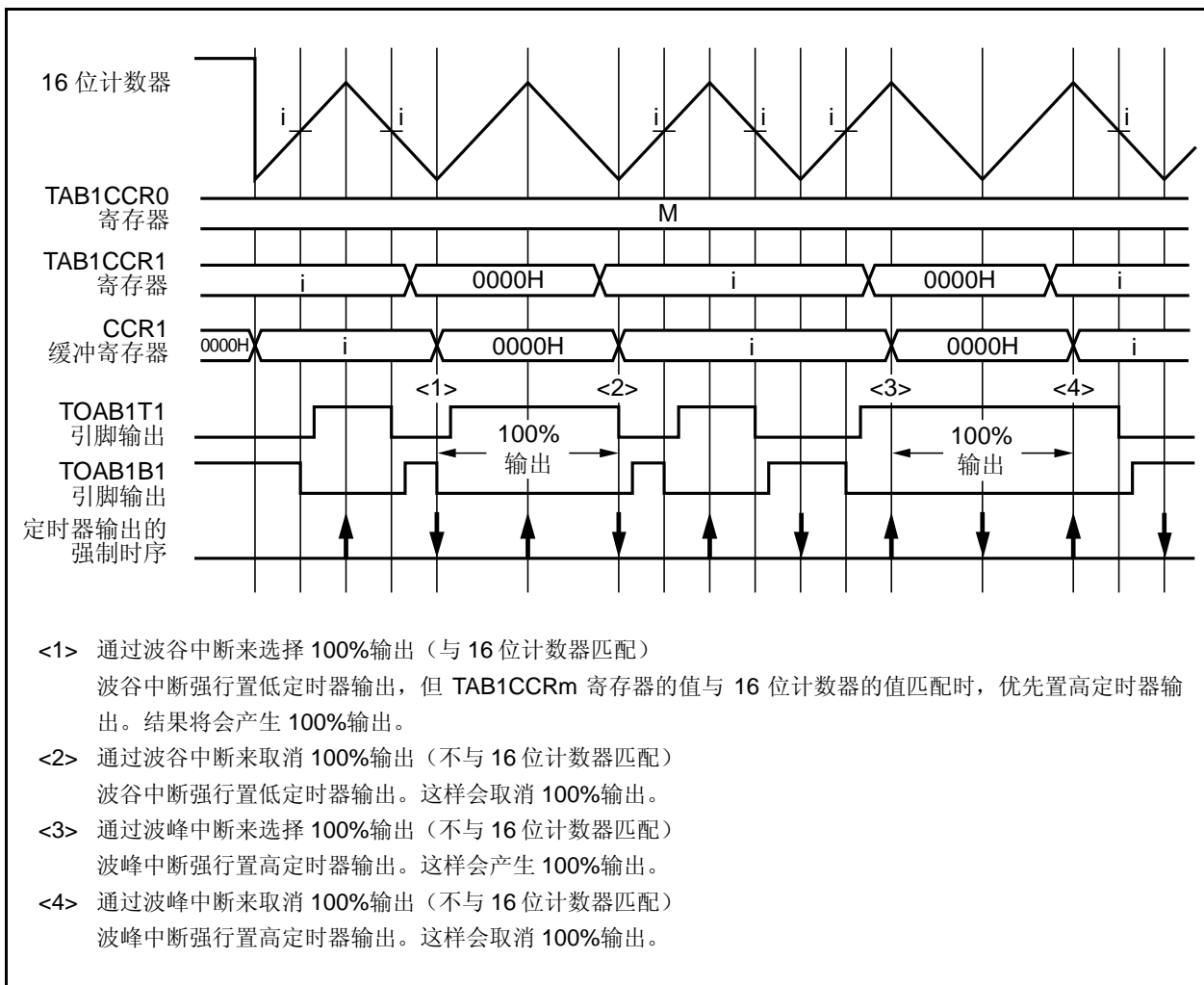
备注 m = 1 至 3

图 11-9. 0% PWM 输出波形（带死区时间）



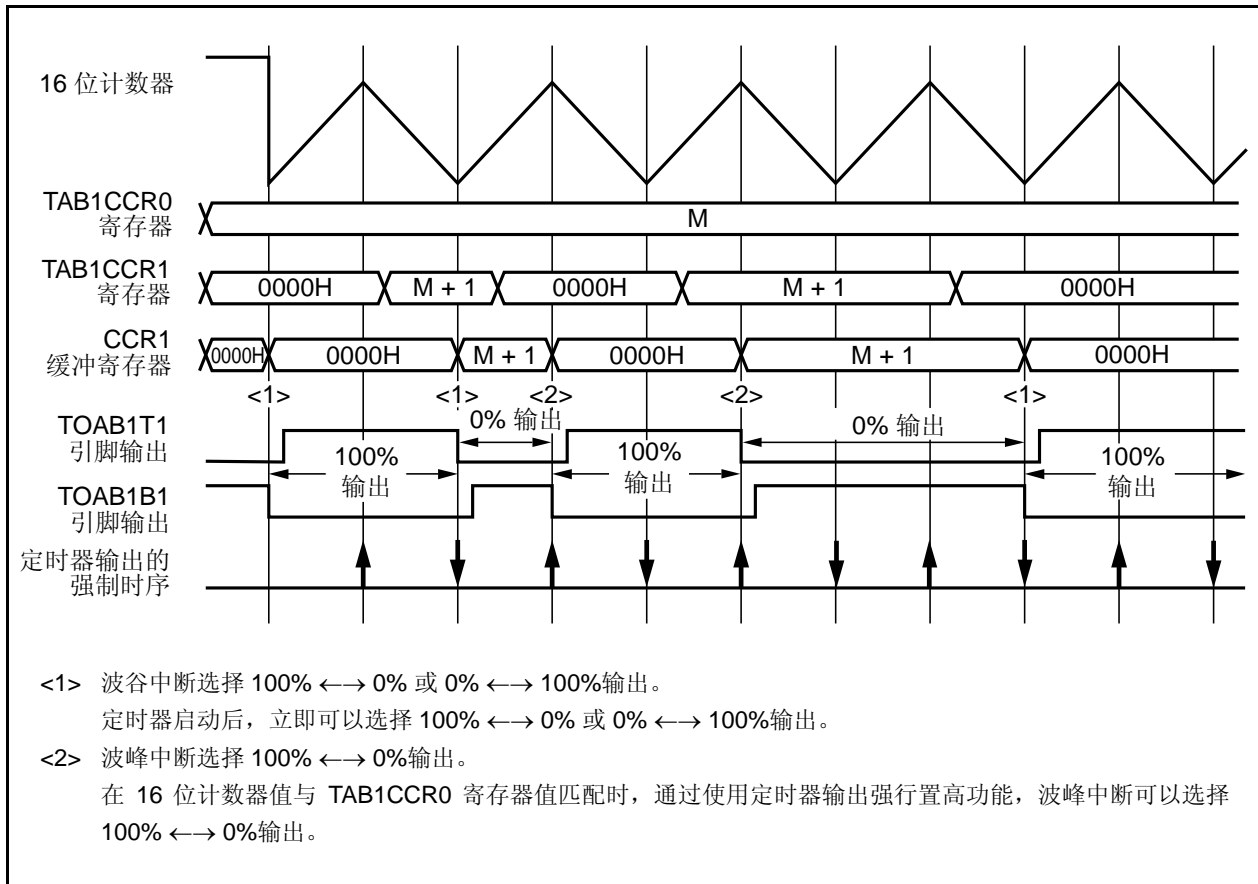
备注 1. ↑表示强行置高，而↓表示强行置低。
2. m = 1 至 3

图 11-10. 100% PWM 输出波形（带死区时间）



备注 1. ↑表示强行置高，而↓表示强行置低。
2. m = 1 至 3

图 11-11. 从 0% 到 100%以及从 100% 到 0%的 PWM 输出波形（带死区时间）

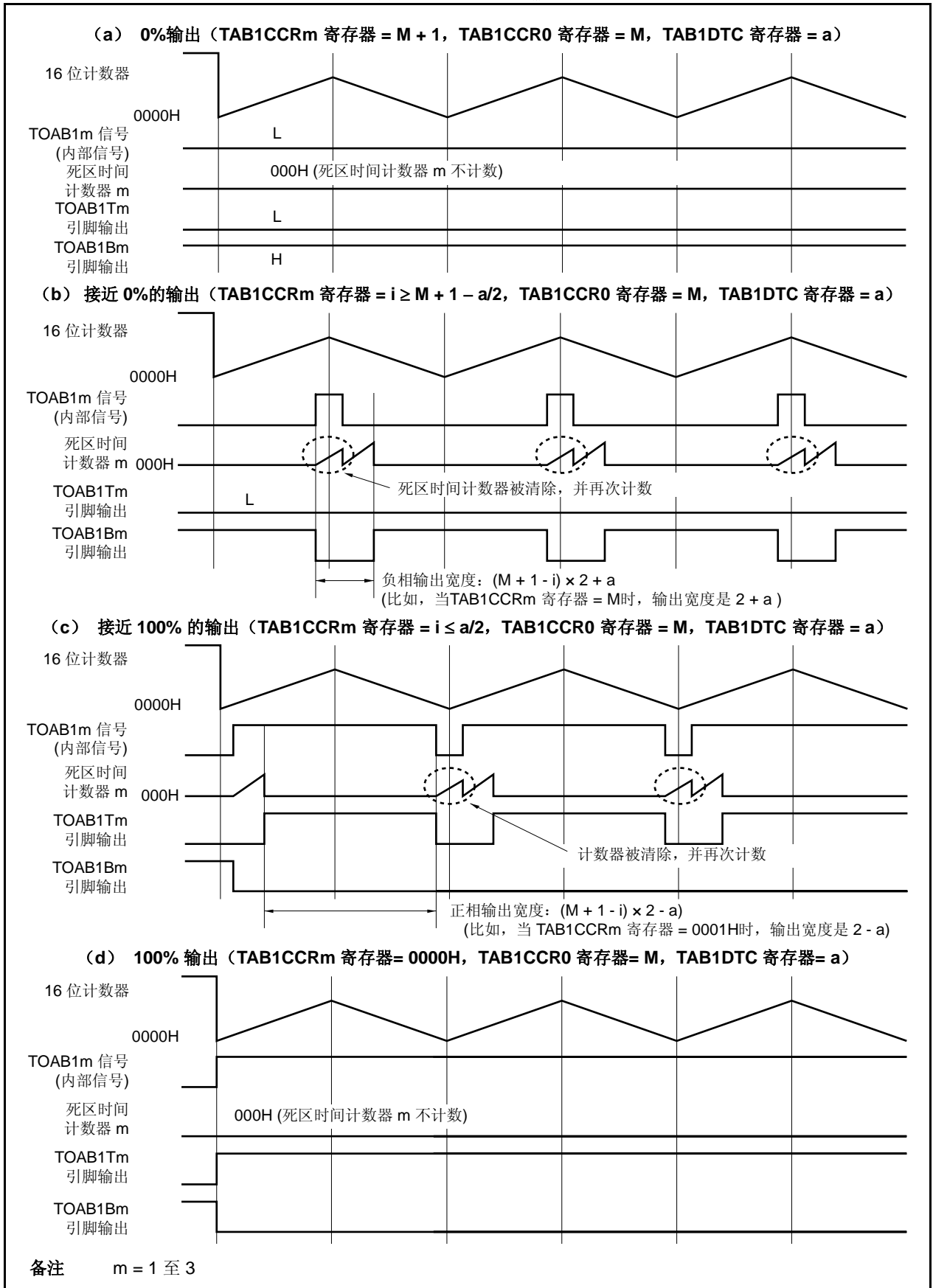


(3) 接近 0% 和 100%输出的输出波形

当死区时间计数时，16 位计数器的值与比较寄存器的值匹配产生中断，那么死区时间计数器将会被清除，并再次开始计数操作。

以下显示了接近 0% 和 100%输出的死区时间控制的输出波形。

图 11-12. 带死区时间的 PWM 输出波形 (2)



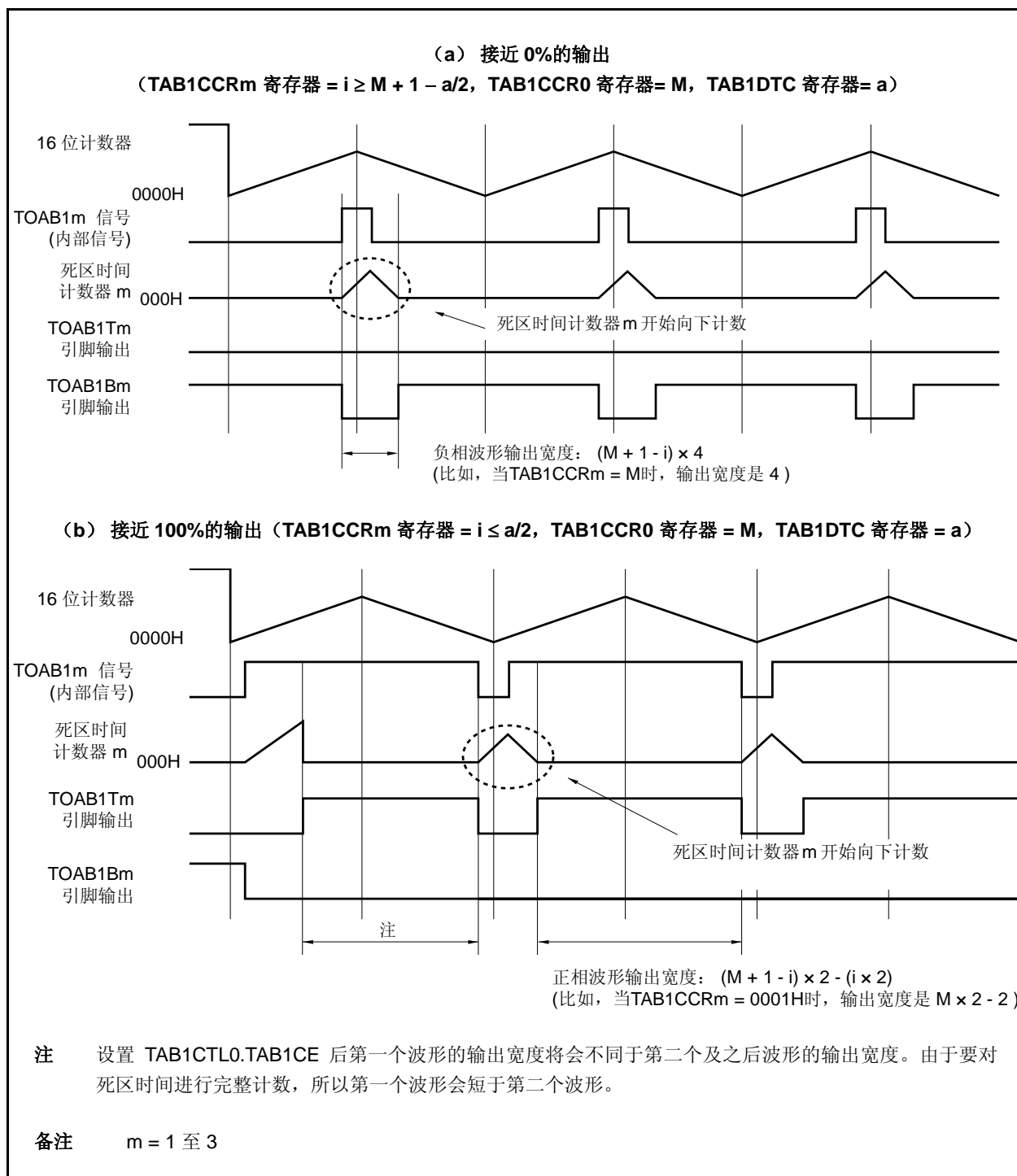
(4) 自动缩小死区时间宽度功能 (TAB1OPT2.TAB1DTM 位 = 1)

将 TAB1OPT2.TAB1DTM 位设置为 1, 可以在接近 0%输出或 100%输出时自动缩小死区时间宽度。

将 TAB1DTM 位设置为 1, 死区时间计数器不会被清除, 但如果在死区时间计数期间改变定时器 AB 的 TOAB1m (内部信号) 输出, 则将会开始向下计数。

以下时序图展示了当 TAB1DTM 位设置为 1 时死区计数器的操作。

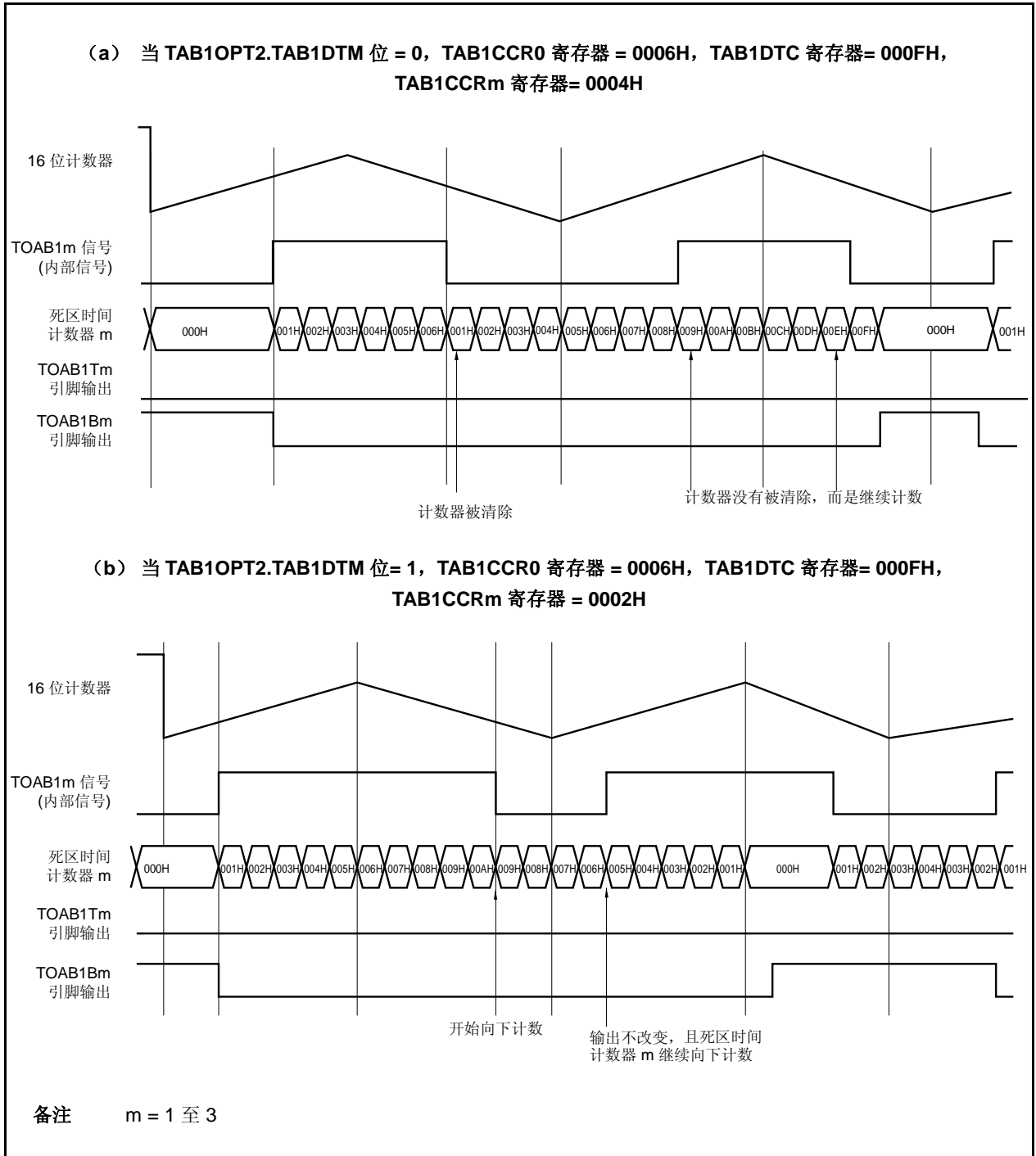
图 11-13. 死区时间计数器 m 的操作 (1)



(5) 设置错误时的死区时间控制

通常情况下，只有在死区时间计数期间接近 0% 和 100% 的输出时，TAB1 的 TOAB1m（内部信号）输出变化一次。本节介绍 TAB1CCR0 寄存器（载波周期）和 TAB1DTC 寄存器（死区时间值）设置错误的示例。如果这些寄存器设置错误，那么在死区时间计数期间 TAB1 的 TOAB1m（内部信号）输出的变化两次或三次。下面的流程图显示了在这种情况下 6 相 PWM 输出波形。

图 11-14. 死区时间计数器 m 的操作 (2)



11.4.3 中断分选功能

- 可以分选的中断有 INTTAB1CC0（波峰中断）及 INTTAB1OV（波谷中断）。
- TAB1OPT1.TAB1ICE 位用于使能 INTTAB1CC0 中断的输出以及可以分选的中断次数。
- TAB1OPT1.TAB1IOE 位用于使能 INTTAB1OV 中断的输出以及可以分选的中断次数。
- TAB1OPT1.TAB1ID4 至 TAB1OPT1.TAB1ID0 位用于指定计数次数，对指定中断进行分选。在指定的计数次数内，中断被屏蔽，在下次时序生成中断。
- TAB1OPT2.TAB1RDE 位用于指定是否对传送进行分选。

如果指定对传送进行分选，那么分选后，传送将会与中断输出同时执行。如果指定不对传送进行分选，那么在已经写入 TAB1CCR1 寄存器之后的传送时序进行传送。

- TAB1OPT0.TAB1CMS 位用于指定带有传送功能的寄存器是执行批量重写还是随时重写。
当 TAB1CMS 位为 0 时，这些寄存器的值会在传送时被同步更新。当 TAB1CMS 位为 1 时，则会新值写入这些寄存器之后立即更新寄存器的值。

从 TAB1CCR_m 寄存器至 CCR_m 缓冲寄存器的传送操作与中断分选时序同步进行。

注意事项 1. 在批量重写模式（传送模式）下使用中断分选功能时，将会按间隙批量重写模式（传送分选模式）来执行操作。

2. 分选后，在该时序产生中断。

备注 m = 1 至 3

(1) 中断分选操作

图 11-15. TAB1OPT1.TAB1ICE 位 = 1, TAB1OPT1.TAB1IOE 位 = 1, TAB1OPT2.TAB1RDE 位 = 1 (波峰/波谷中
断输出) 时的中断分选操作

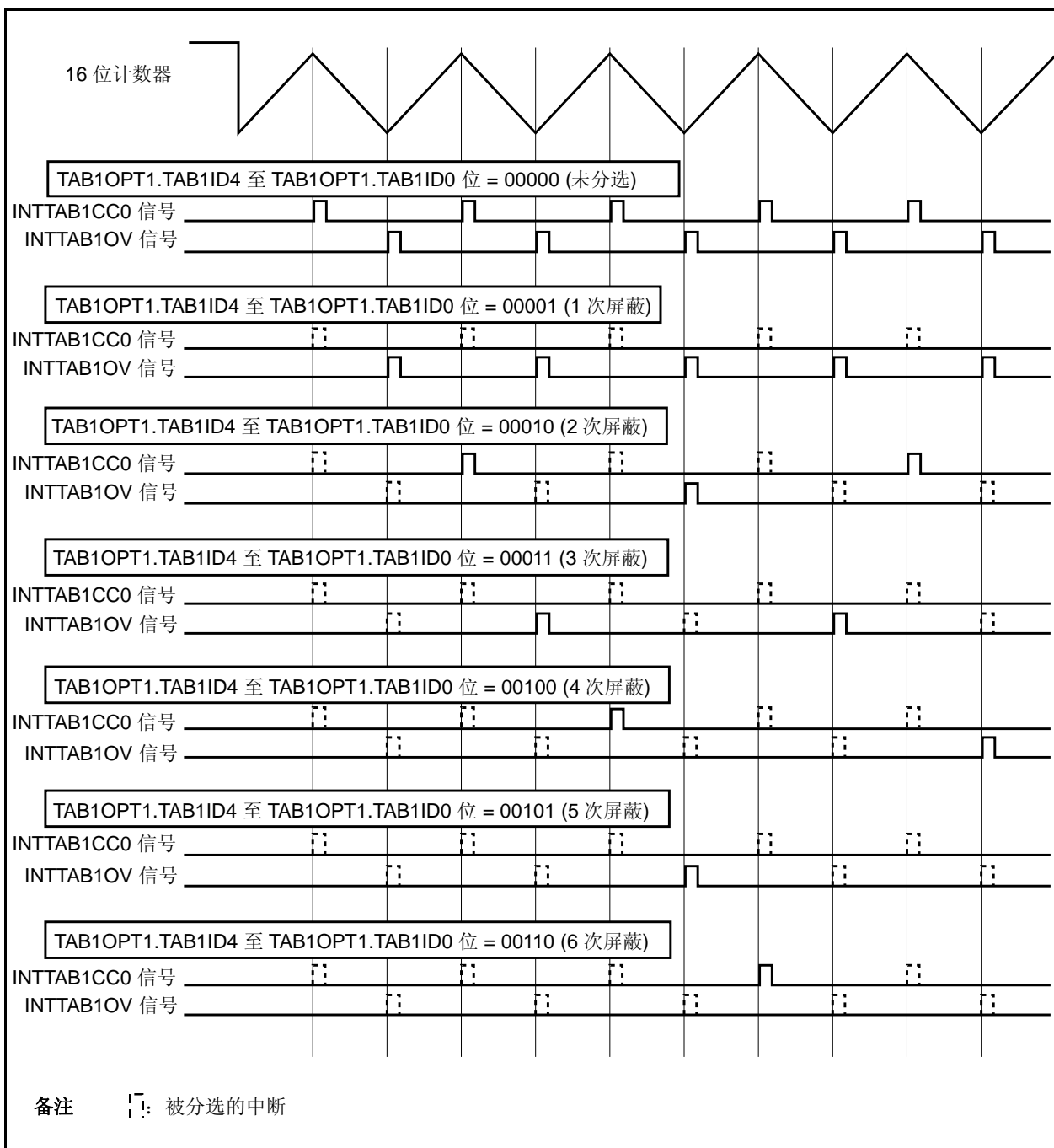


图 11-16. TAB1OPT1.TAB1ICE 位 = 1, TAB1OPT1.TAB1IOE 位 = 0, TAB1OPT2.TAB1RDE 位 = 1 (波峰中断输出) 时的中断分选操作

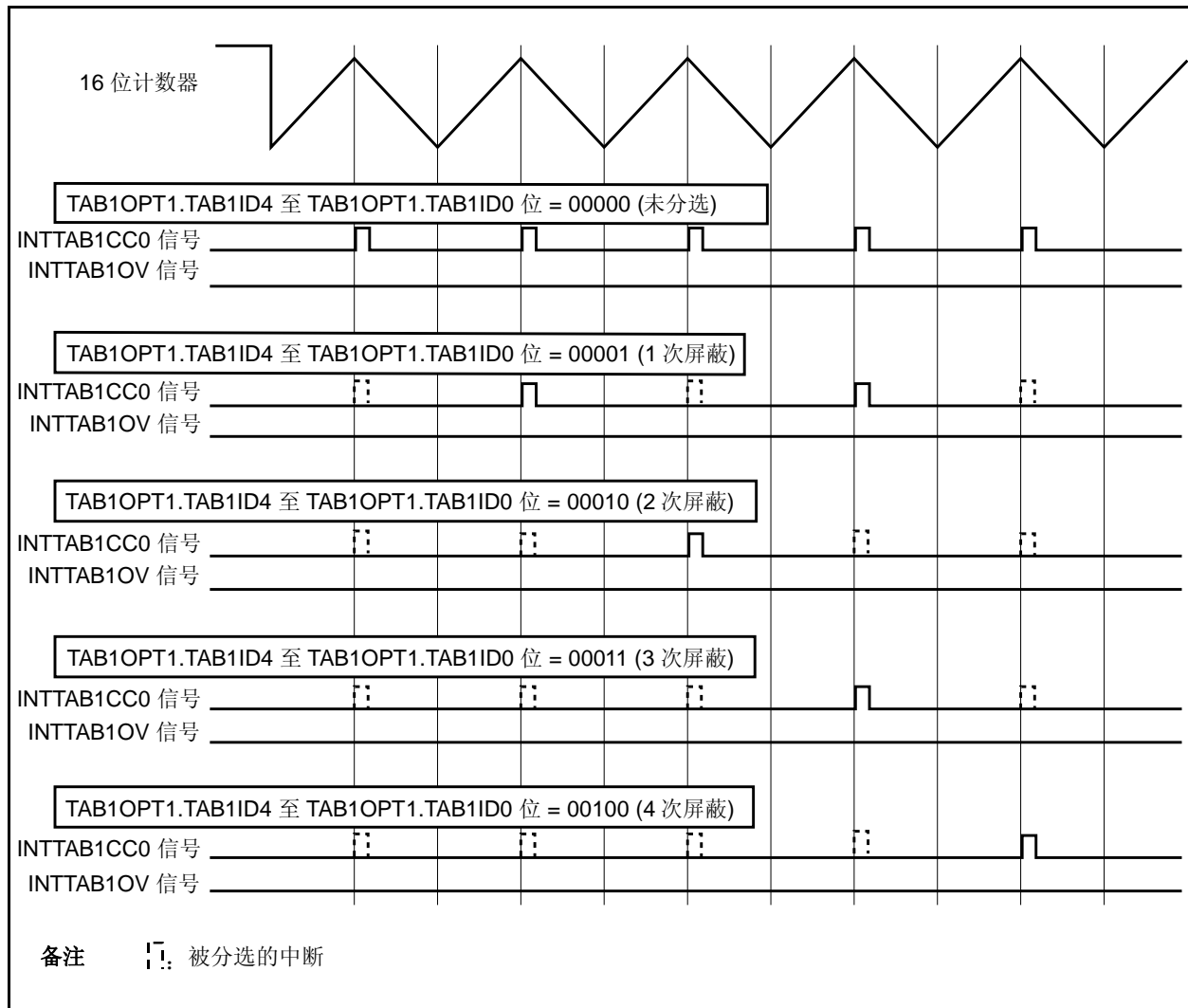
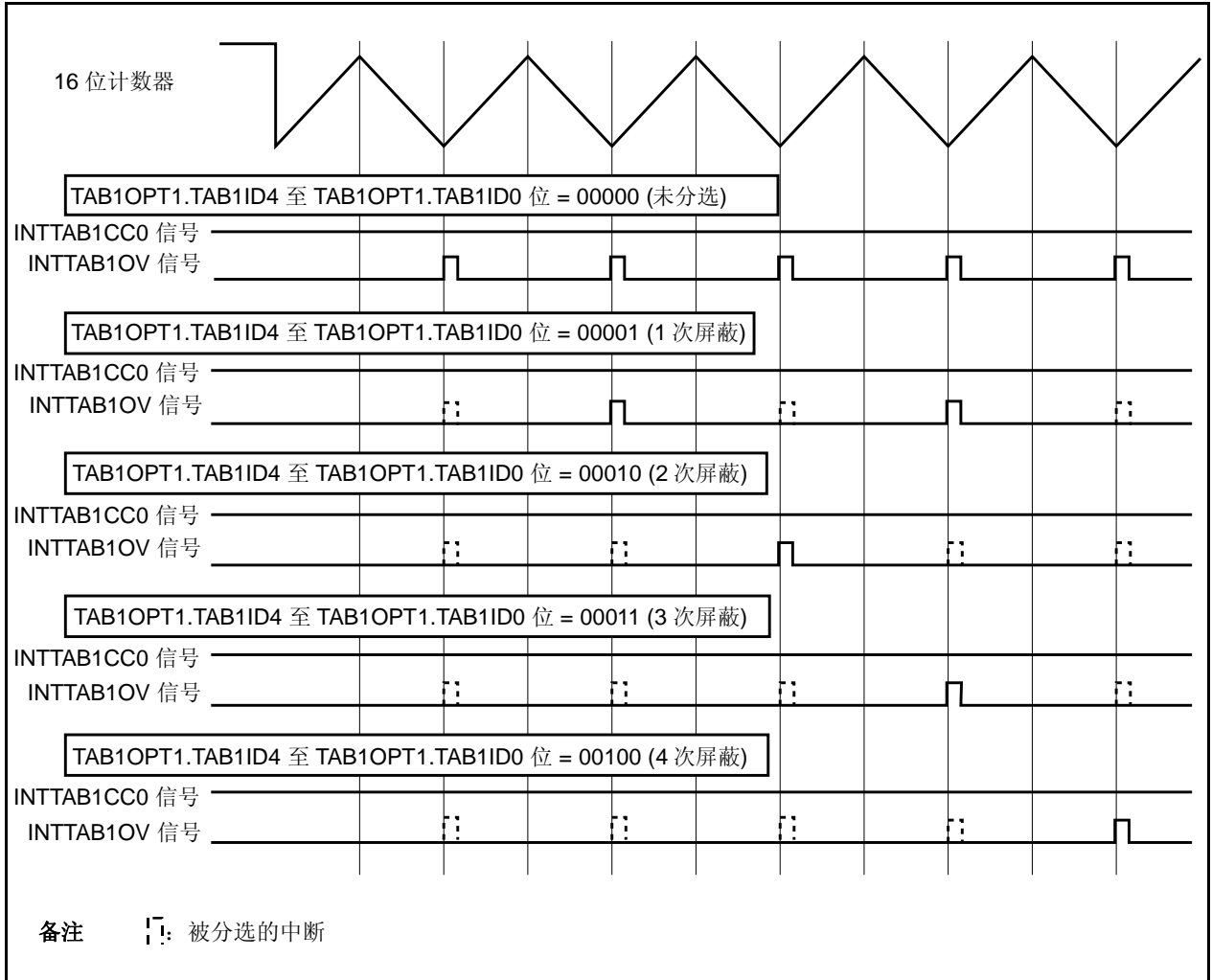


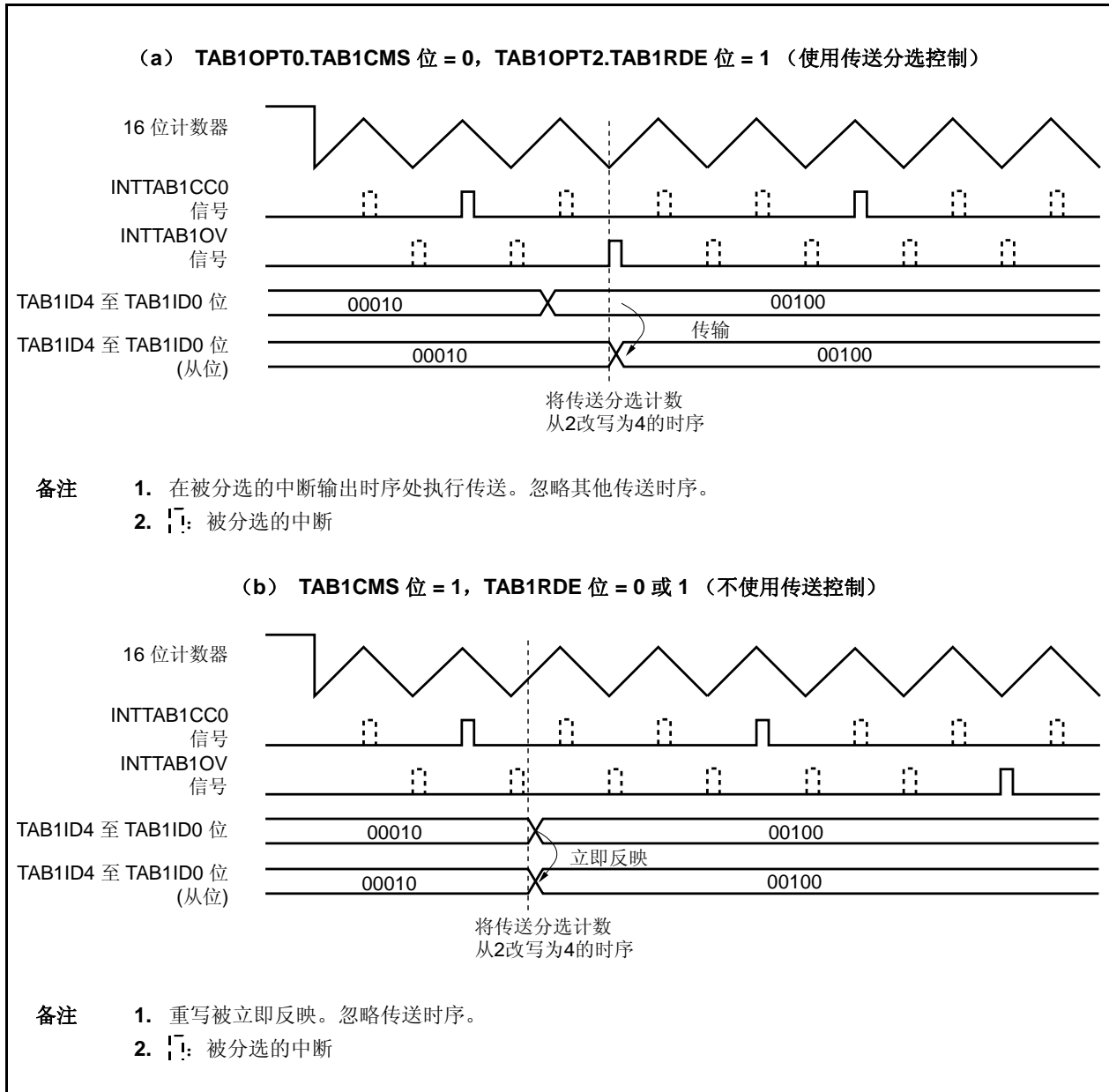
图 11-17. TAB1OPT1.TAB1ICE 位 = 0, TAB1OPT1.TAB1IOE 位 = 1, TAB1OPT2.TAB1RDE 位 = 1 (波谷中断输出) 时的中断分选操作



(2) 交替输出波峰中断 (INTTAB1CC0) 和波谷中断 (INTTAB1OV)

要交替输出波峰中断和波谷中断, 将 TAB1OPT1.TAB1ICE 和 TAB1OPT1.TAB1IOE 位都设置为 1。

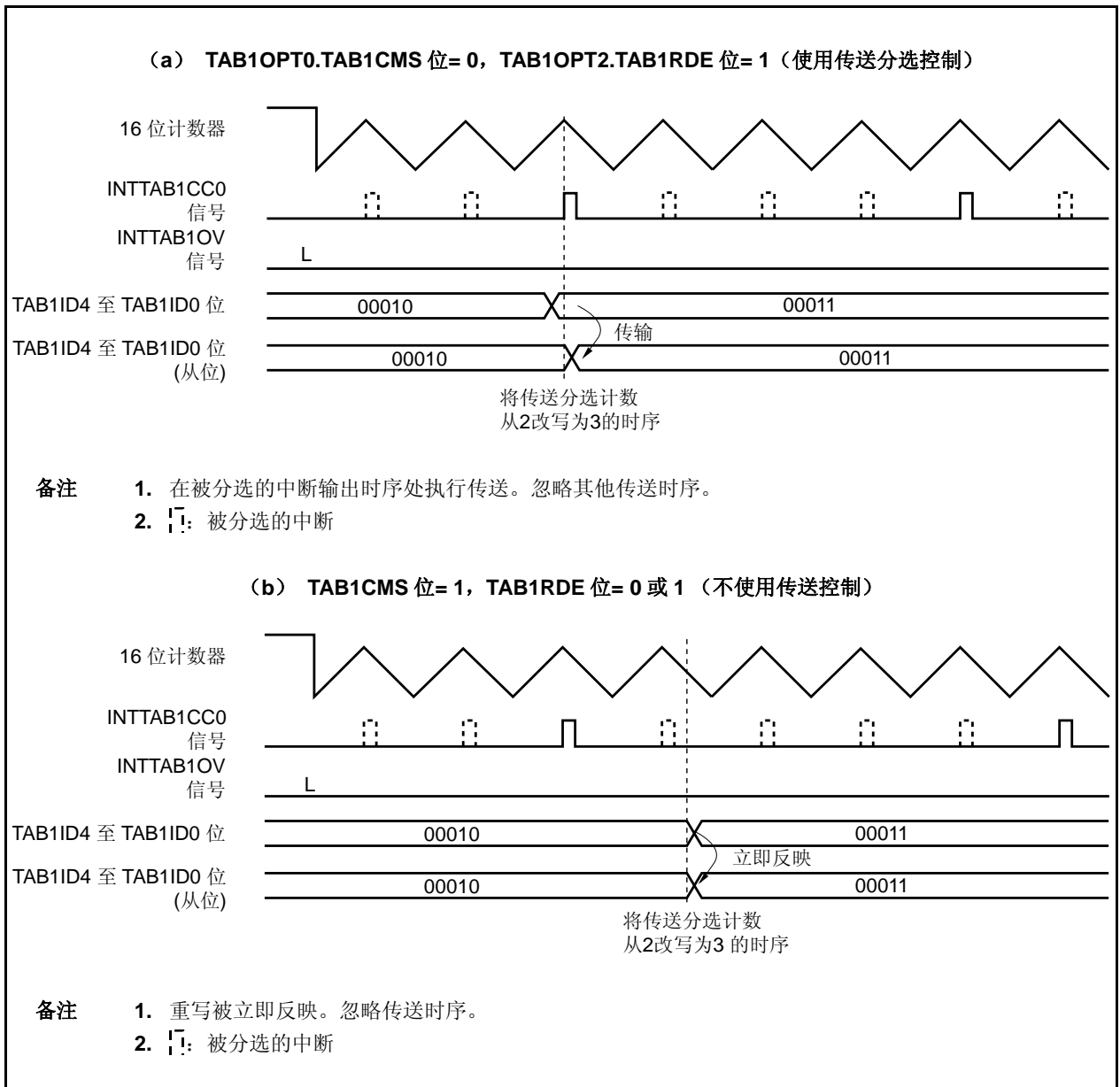
图 11-18. 波峰/波谷中断输出



(3) 只输出波峰中断 (INTTAB1CC0)

设置 TAB1OPT1.TAB1ICE 位为 1, 且将 TAB1OPT1.TAB1IOE 位清 0。

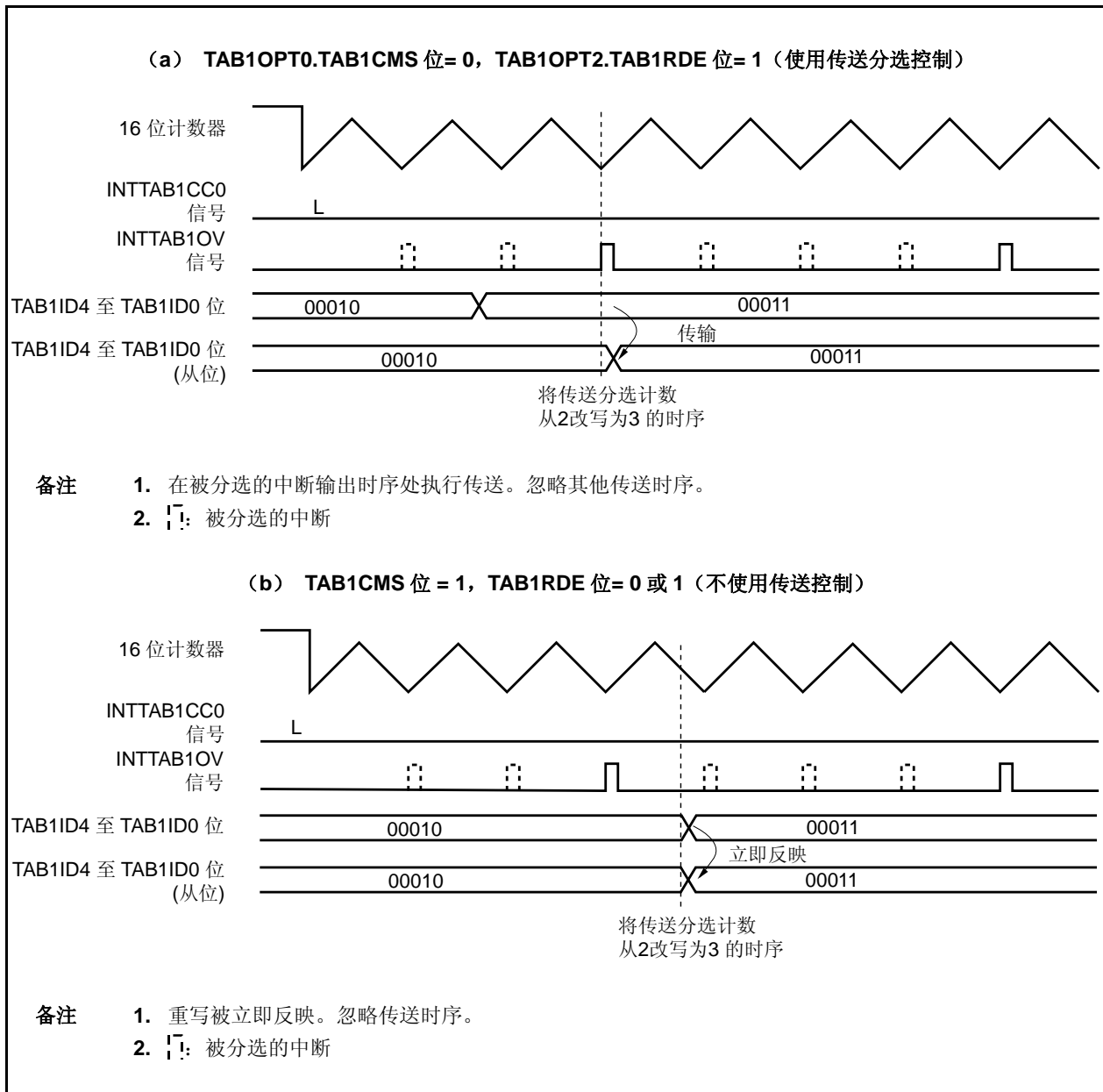
图 11-19. 波峰中断输出



(4) 只输出波谷中断 (INTTAB1OV)

将 TAB1OPT1.TAB1ICE 位清 0 且将 TAB1IOE 位置 1。

图 11-20. 波谷中断输出



11.4.4 重写带有传送功能的寄存器的操作

以下七个寄存器提供有传送功能，且用于控制电机。每个寄存器都有一个缓冲寄存器。

- TAB1CCR0: 指定 16 位计数器 (TAB) 周期的寄存器
- TAB1CCR1: 指定 TOAB1T1 (U) 和 TOAB1B1 (\bar{U}) 占空比的寄存器
- TAB1CCR2: 指定 TOAB1T2 (V) 和 TOAB1B2 (\bar{V}) 占空比的寄存器
- TAB1CCR3: 指定 TOAB1T3 (W) 和 TOAB1B3 (\bar{W}) 占空比的寄存器
- TAB1OPT1: 指定中断分选的寄存器
- TAA4CCR0: 指定 A/D 转换启动触发生成时序 (调谐操作的 TAA4) 的寄存器
- TAA4CCR1: 指定 A/D 转换启动触发生成时序 (调谐操作的 TAA4) 的寄存器

为带有传送功能的寄存器提供以下三种重写模式。

- 随时重写模式

将 TAB1OPT0.TAB1CMS 位置 1 即可设置该模式。TAB1OPT2.TAB1RDE 位的设置被忽略。

在该模式下，每个比较寄存器被独立更新，而比较寄存器的值则会在写入新的值后立即更新。

- 批量重写模式 (传送模式)

将 TAB1OPT0.TAB1CMS 位清 0，将 TAB1OPT1.TAB1ID4 至 TAB1OPT1.TAB1ID0 位置为 00000，并且将 TAB1OPT2.TAB1RDE 位清 0，可以设置该模式。

向 TAB1CCR1 寄存器写入数据时，这七个寄存器中的所有数据会在下一个传送时序立即传送到缓冲寄存器。除非 TAB1CCR1 寄存器被重写，否则即使重写了其余六个寄存器，也不会执行传送操作。

不考虑中断，传送时序是每个波峰 (16 位计数器值与 TAB1CCR0 寄存器值匹配) 和波谷 (16 位计数器值等于 0001H) 的时序。

- 间隙批量重写模式 (传送分选模式)

将 TAB1OPT0.TAB1CMS 位清 0 及 TAB1OPT2.TAB1RDE 位置 1，可以设置该模式。

向 TAB1CCR1 寄存器写入数据时，这七个寄存器中的所有数据会在下一个传送时序立即传送到缓冲寄存器。除非 TAB1CCR1 寄存器被重写，否则即使重写了其余六个寄存器，也不会执行传送操作。

如果通过 TAB1OPT1 寄存器指定了中断分选，和分选中断一样，传送时序也被分选，并且这七个寄存器中的所有数据会在波峰中断 (16 位计数器值与 TAB1CCR0 寄存器值匹配) 或波谷中断 (16 位计数器值与 0001H 匹配) 时被立即传送到缓冲寄存器。

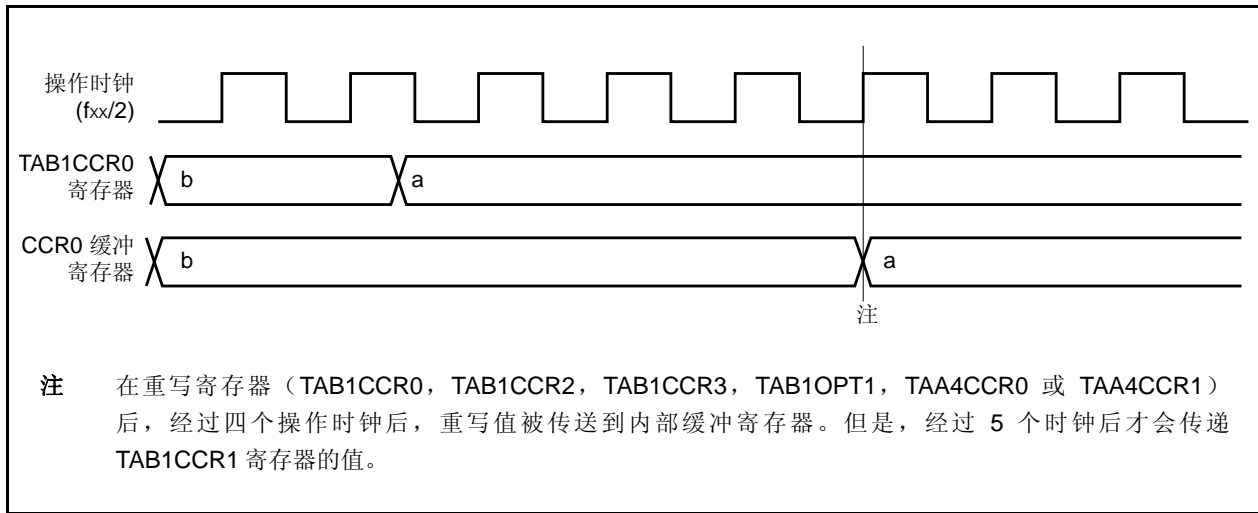
关于中断分选功能的详情参见 11.4.3 中断分选功能。

(1) 随时重写模式

将 TAB1OPT0.TAB1CMS 位置 1，可以设置该模式。TAB1OPT2.TAB1RDE 位的设置被忽略。

在该模式下，写入带有传送功能的各个寄存器中的值被立即传送到内部缓冲寄存器中，并与计数器的值进行比较。如果在该模式下 16 位计数器的计数值与 TAB1CCRm 寄存器的值匹配之后，重写带有传送功能的寄存器，因为发生第一次匹配后的下一次匹配被忽略，因此重写值不会被反映。如果在向上计数过程中重写寄存器，那么在计数器开始向下计数后，新的寄存器值生效。

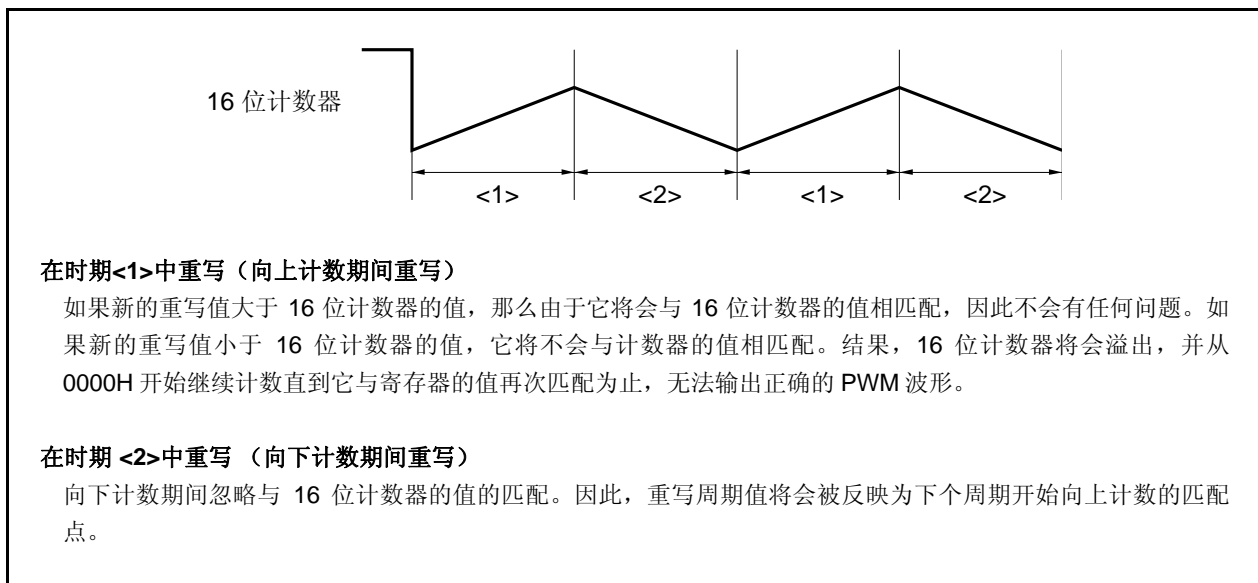
图 11-21. 反映重写值的时序



(a) 重写 TAB1CCR0 寄存器

即使在随时重写模式下重写 TAB1CCR0 寄存器，在某些情况下，也可能无法反映新值。

图 11-22. 重写 TAB1CCR0 寄存器的示例



(b) 重写 TAB1CCRm 寄存器

图 11-24 显示了 16 位计数器的值与 TAB1CCRm 寄存器的值匹配之前的重写时序（图 11-23 中的<a>），图 11-25 显示了 16 位计数器的值与 TAB1CCRm 寄存器的值匹配之后的重写时序（图 11-23 中的）。

图 11-23. 16 位计数器与 TAB1CCRm 寄存器的基本操作

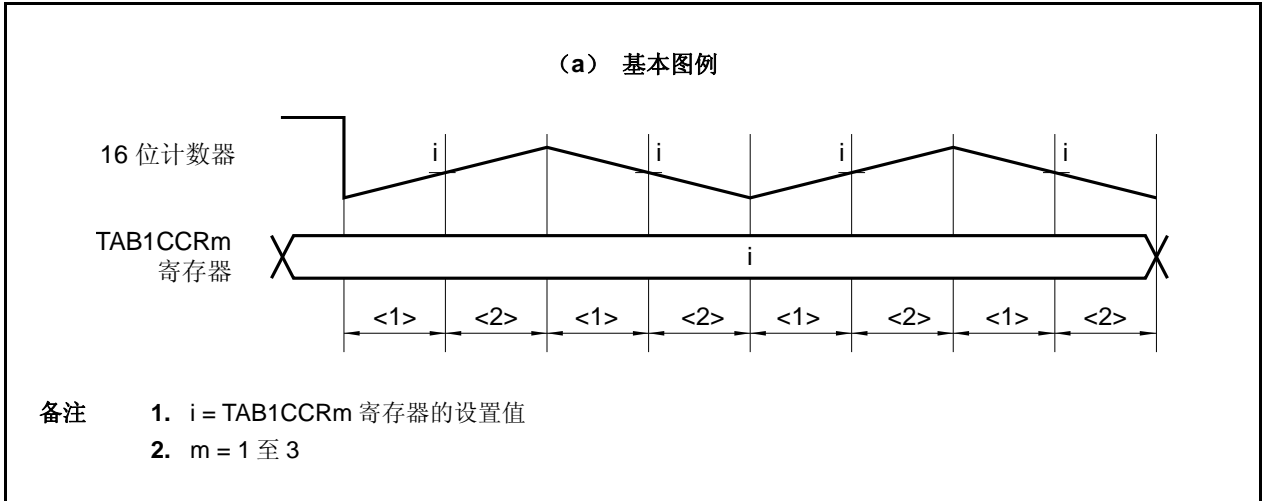
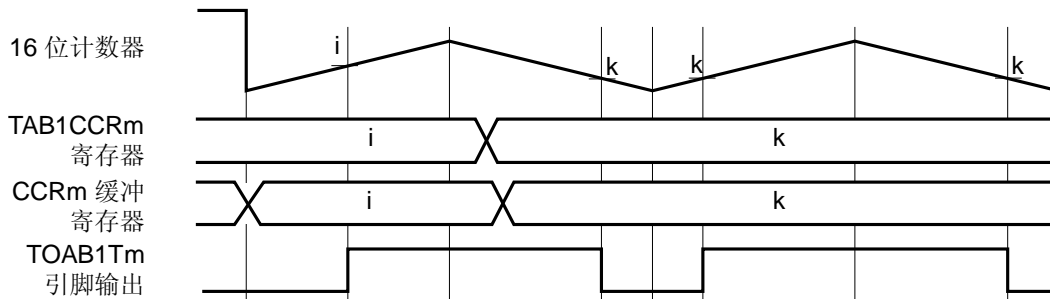


图 11-24. 重写 TAB1CCR1 至 TAB1CCR3 寄存器的示例（匹配发生之前的重写）

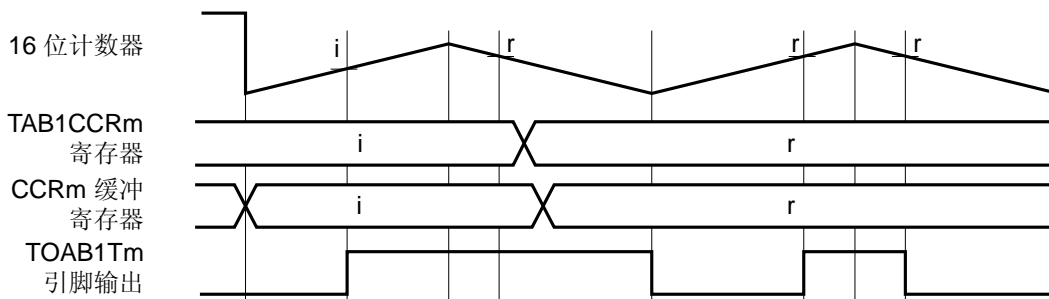
(a)

如果在 TAB1CCRm 寄存器的值与 16 位计数器的值匹配之前重写该寄存器，那么在寄存器被重写后，该寄存器的值将会与 16 位计数器的值相匹配。因此，新的寄存器值将会立即被反映。



(b)

如果写入 TAB1CCRm 寄存器的值小于 16 位计数器的值（如果计数器向下计数，则寄存器值大于 16 位计数器值时），那么由于寄存器的值与计数器的值不发生匹配，输出波形如下。



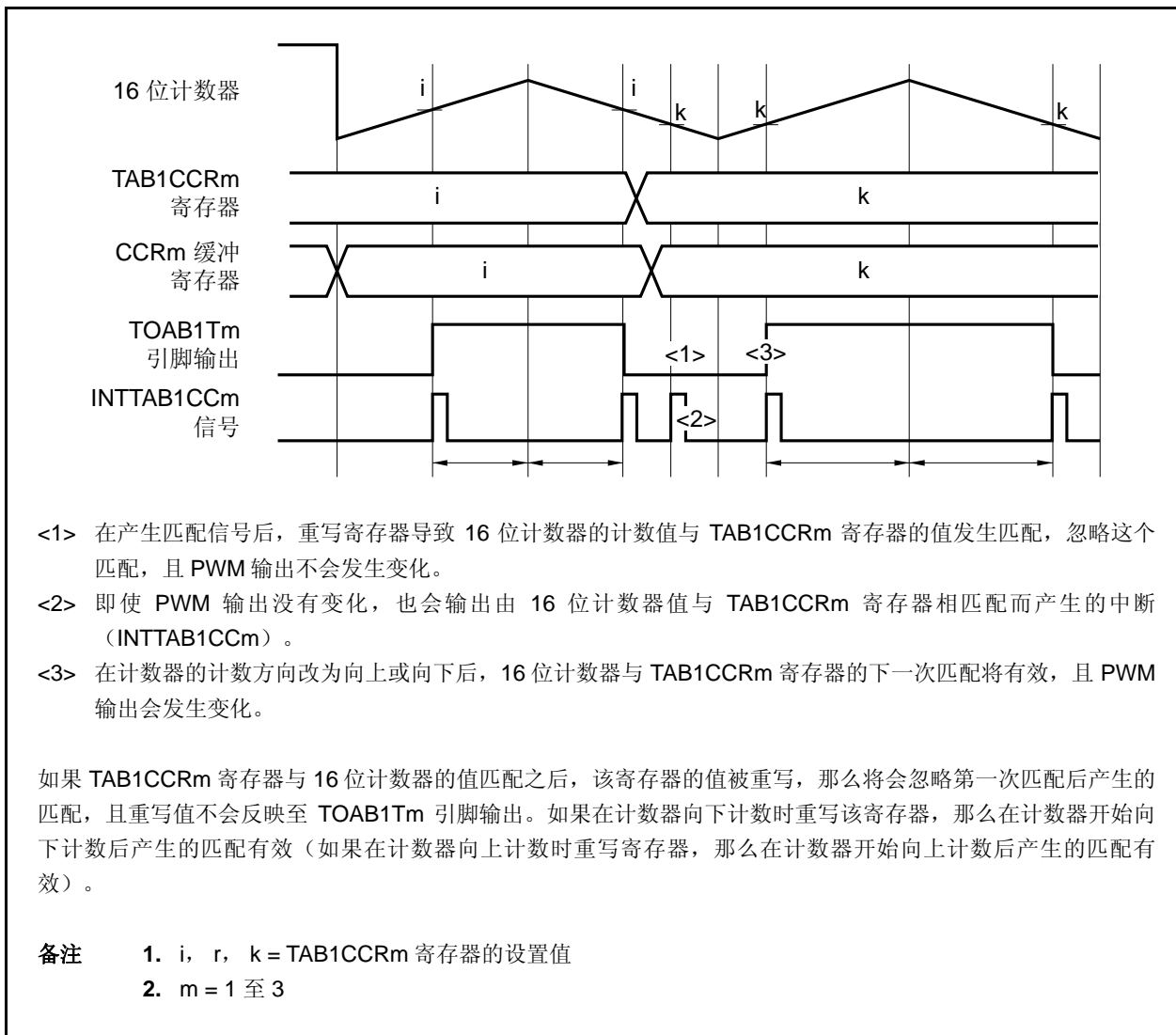
如果寄存器的值与计数器的值不发生匹配，那么 TOAB1Tm 引脚输出不会发生变化。即使 16 位计数器的值与 TAB1CCRm 寄存器的值不发生匹配，TOAB1Tm 引脚输出也会在发生波峰中断时变为高电平，在发生波谷中断时变为低电平。

该功能可以提供 0% 输出和 100% 输出。

详情参见 11.4.2 (2) 0%/100% 的 PWM 输出。

- 备注
1. i, r, k = TAB1CCRm 寄存器的设置值
 2. m = 1 至 3

图 11-25. 重写 TAB1CCR1 至 TAB1CCR3 寄存器的示例（匹配发生之后的重写）



(c) 重写 TAB1OPT1 寄存器

写入 TAB1OPT1 寄存器时，中断分选计数器被清除。中断分选计数器被清除后，已经检测到的中断发生次数被丢弃。结果，中断产生间隔会暂时延长。

为了避免该操作，应在间隙批量重写模式（传送分选模式）下重写 TAB1OPT1 寄存器。

关于重写 TAB1OPT1 寄存器的详情参见 11.4.3 中断分选功能。

(2) 批量重写模式（传送模式）

将 TAB1OPT0.TAB1CMS 位清 0，将 TAB1OPT1.TAB1ID4 至 TAB1OPT1.TAB1ID0 位设为 00000，且将 TAB1OPT2.TAB1RDE 位清 0，可以设置该模式。

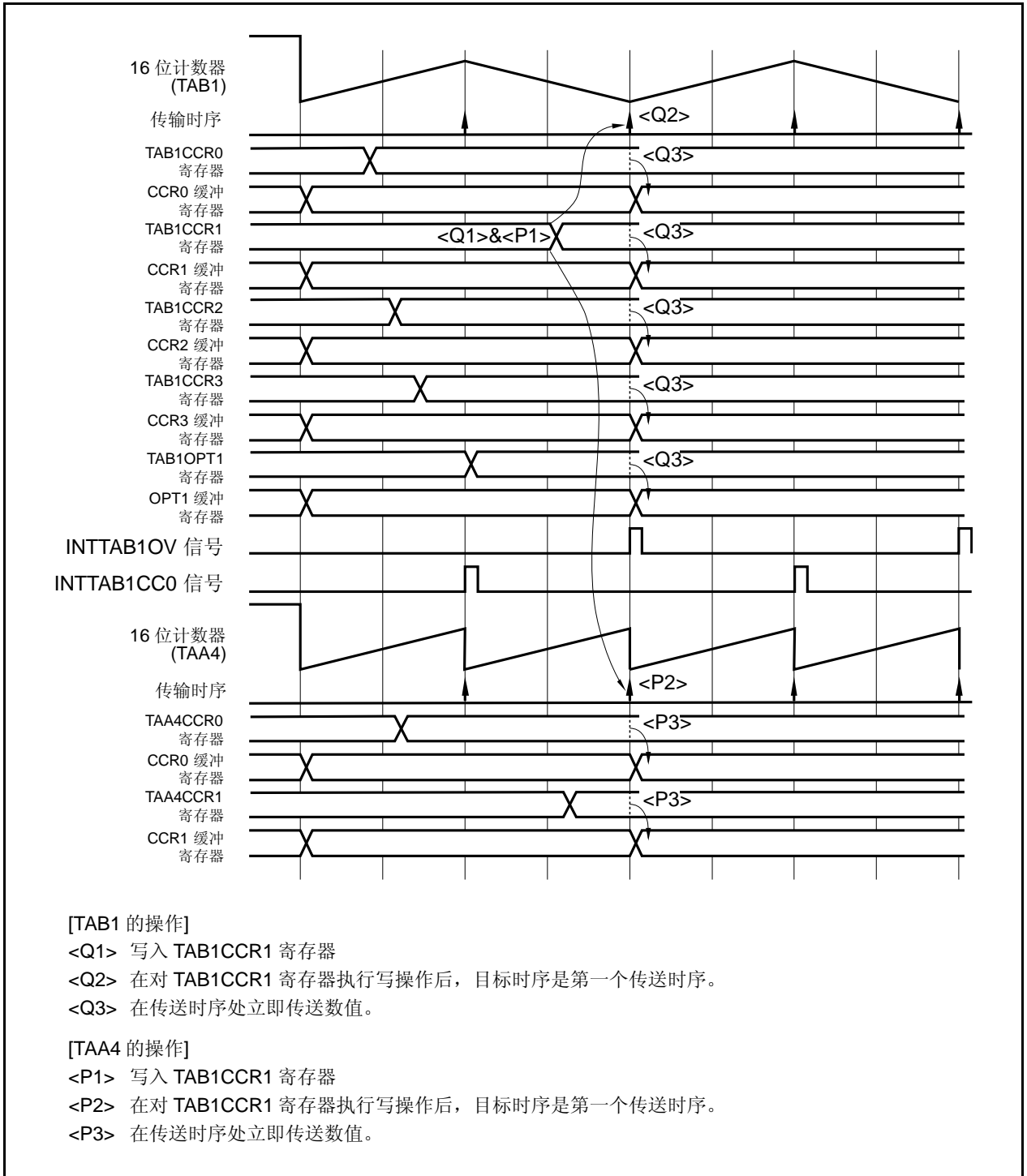
在该模式下，写入带有传送功能的各个寄存器中的值被立即传送到内部缓冲寄存器，并与计数器的值进行比较。

(a) 重写步骤

如果向 TAB1CCR1 寄存器写入数据，那么在下一个传送时序，设置在 TAB1CCR0 至 TAB1CCR3，TAB1OPT1，TAA4CCR0 和 TAA4CCR1 寄存器中的值全部被立即传送到内部缓冲寄存器中。因此，应该最后写入 TAB1CCR1 寄存器。产生传送时序之前，在写入 TAB1CCR1 寄存器之后禁止对其他寄存器进行写操作（直到波峰（16 位计数器值与 TAB1CCR0 寄存器值间的匹配）或波谷（16 位计数器值等于 0001H））。操作步骤如下。

- <1> 对 TAB1CCR0，TAB1CCR2，TAB1CCR3，TAB1OPT1，TAA4CCR0 和 TAA4CCR1 寄存器进行重写。
对于无需重写的寄存器，不要重写。
- <2> 重写 TAB1CCR1 寄存器。
即使无需重写 TAB1CCR1 寄存器，也应该将相同的值重写到该寄存器。
- <3> 保持下次重写挂起直至生成传送时序。
在产生 INTTAB1OV 或 INTTAB1CC0 中断后，对寄存器再次进行重写。
- <4> 返回步骤 <1>。

图 11-26. 批量写入模式中的基本操作



(b) 重写 TAB1CCR0 寄存器

在批量重写模式下对 TAB1CCR0 寄存器进行重写时，根据传送是发生在波峰（16 位计数器值与 TAB1CCR0 寄存器值间的匹配）发生还是在波谷（16 位计数器值等于 0001H），输出波形会有所不同。通常情况下，推荐在 16 位计数器进行向下计数时重写 TAB1CCR0 寄存器，并在波峰时序的传送时序处传送寄存器的值。

图 11-28 展示了在 16 位计数器向上计数时（图 11-27 中的时期<1>）重写 TAB1CCR0 寄存器的示例。图 11-29 显示了在 16 位计数器向下计数时（图 11-27 中的时期 <2>）重写 TAB1CCR0 寄存器的示例。

图 11-27. 16 位计数器的基本操作

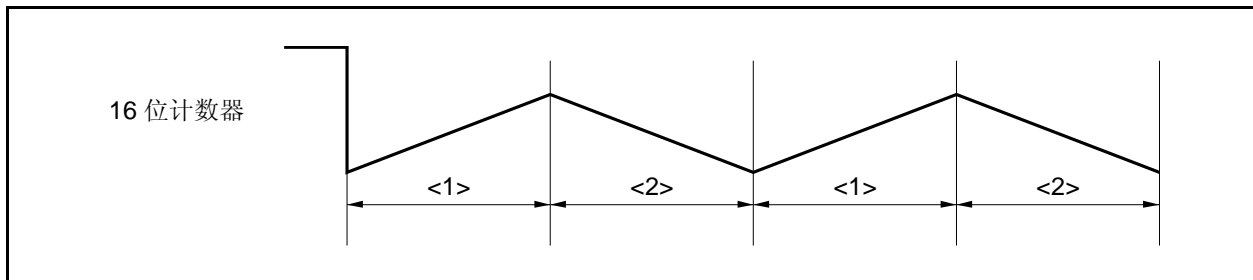


图 11-28 中的传送时序发生在波峰时序的位置。当 16 位计数器向下计数时，周期会发生变化并输出不对称的三角波。由于改变了周期，所以要重写占空比（电压数据值）。

图 11-28. 重写 TAB1CCR0 寄存器示例（向上计数期间）

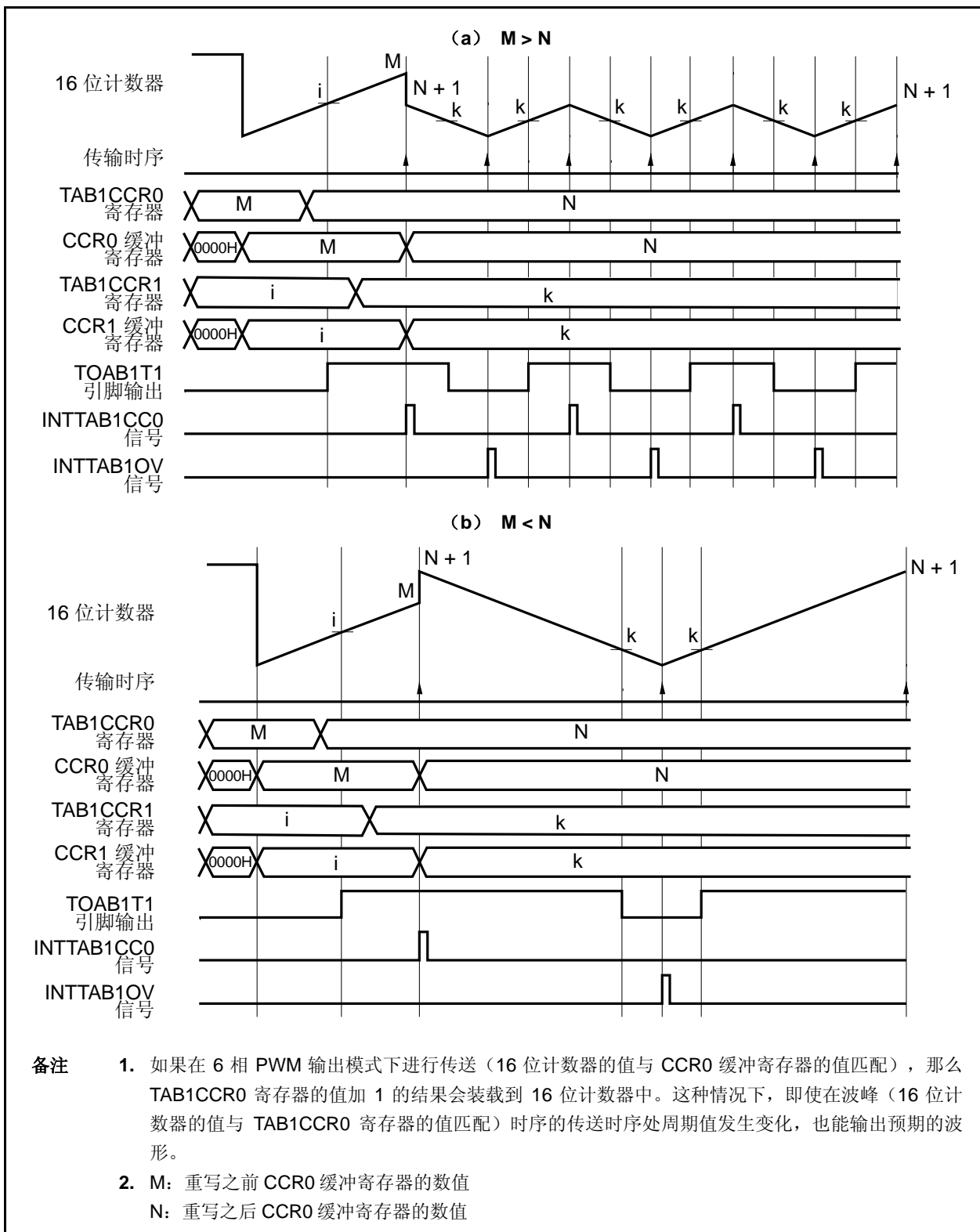
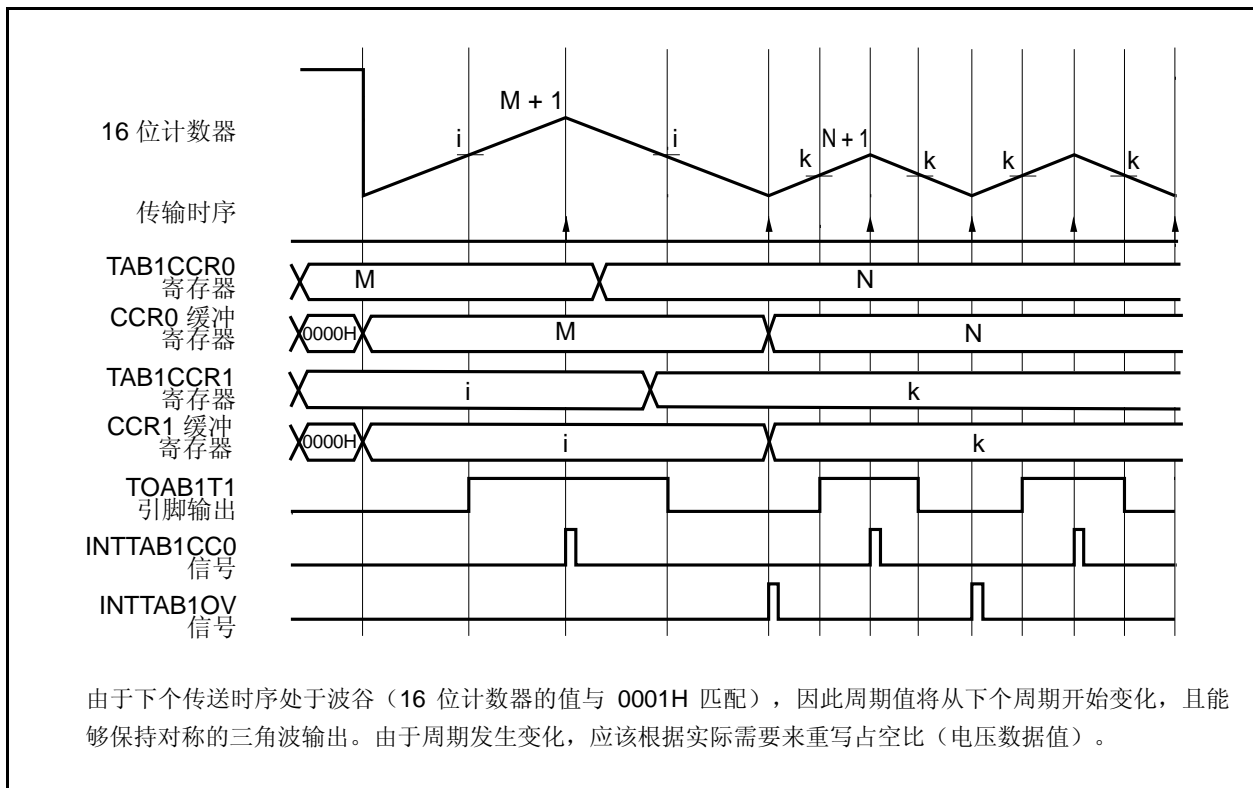
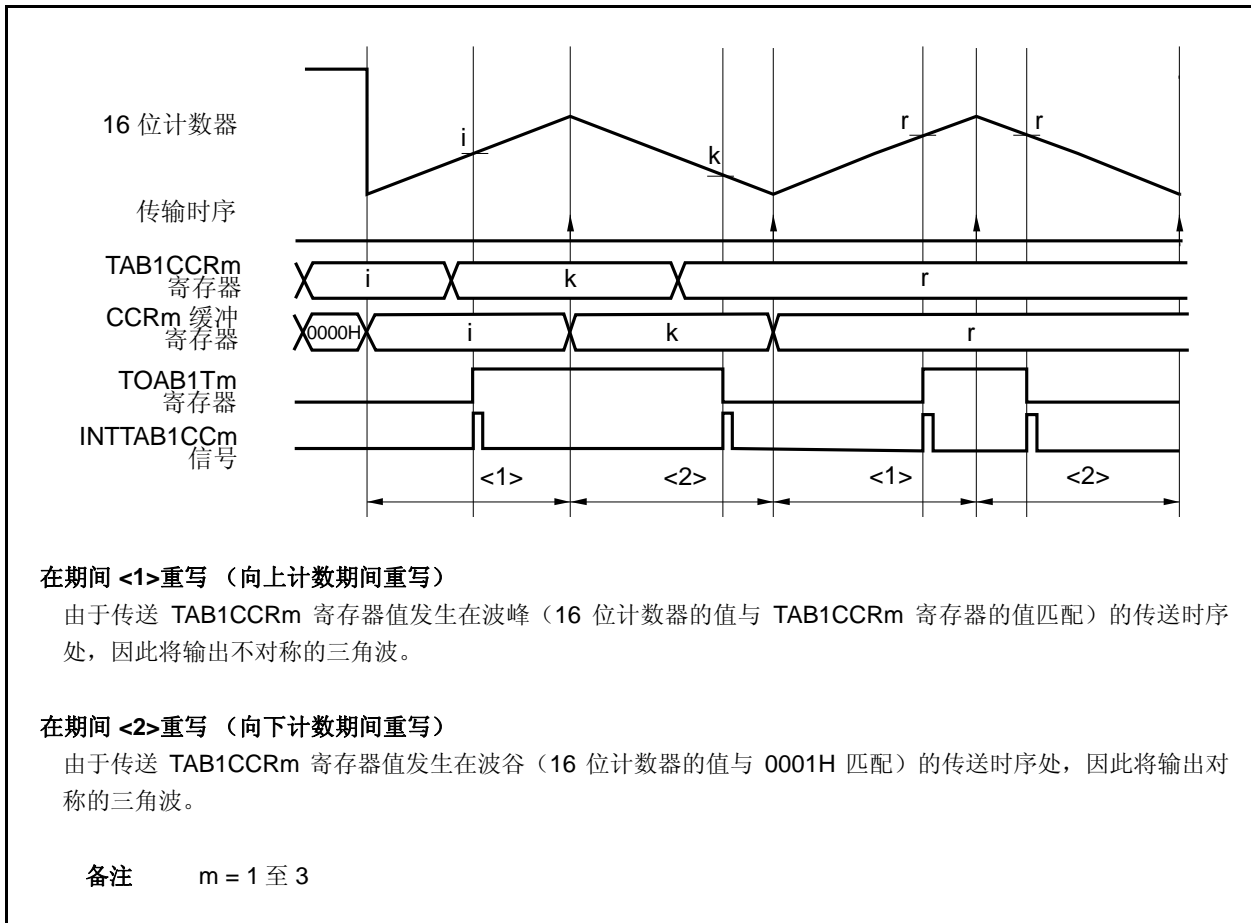


图 11-29. 重写 TAB1CCR0 寄存器的示例（向下计数期间）



(c) 重写 TAB1CCRm 寄存器

图 11-30. 重写 TAB1CCRm 寄存器的示例



(d) 传送 TAB1OPT1 寄存器的值

不要将 TAB1OPT1.TAB1ID4 至 TAB1OPT1.TAB1ID0 位设置为 00000 以外的值。当使用中断分选功能时，在间隙批量重写模式（传送分选模式）中重写 TAB1OPT1 寄存器。
关于重写 TAB1OPT1 寄存器的详情参见 11.4.3 中断分选功能。

(3) 间隙批量重写模式（传送分选模式）

将 TAB1OPT0.TAB1CMS 位清 0，将 TAB1OPT2.TAB1RDE 位设置为 1，可以设置该模式。

在该模式中下，在分选的传送时序后，写入各个比较寄存器的值都会被立即传送到内部缓冲寄存器，并与计数器值进行比较。传送时序是通过中断分选产生中断（INTTAB1CC0，INTTAB1OV）的时序。

关于中断分选功能的详情参见 11.4.3 中断分选功能。

(a) 重写步骤

如果向 TAB1CCR1 寄存器写入数据，那么在下一个传送时序处，设置在 TAB1CCR0 至 TAB1CCR3，TAB1OPT1，TAA4CCR0 及 TAA4CCR1 寄存器中的数值全部被立即传送到内部缓冲寄存器中。因此，TAB1CCR1 寄存器应该最后写入。传送时序生成之前，在写入 TAB1CCR1 寄存器后禁止对其他寄存器进行写操作（直到产生 INTTAB1OV 或 INTTAB1CC0 中断）。操作步骤如下。

<1> 重写 TAB1CCR0，TAB1CCR2，TAB1CCR3，TAB1OPT1，TAA4CCR0 及 TAA4CCR1 寄存器。

对于无需重写的寄存器，不要重写。

<2> 重写 TAB1CCR1 寄存器。

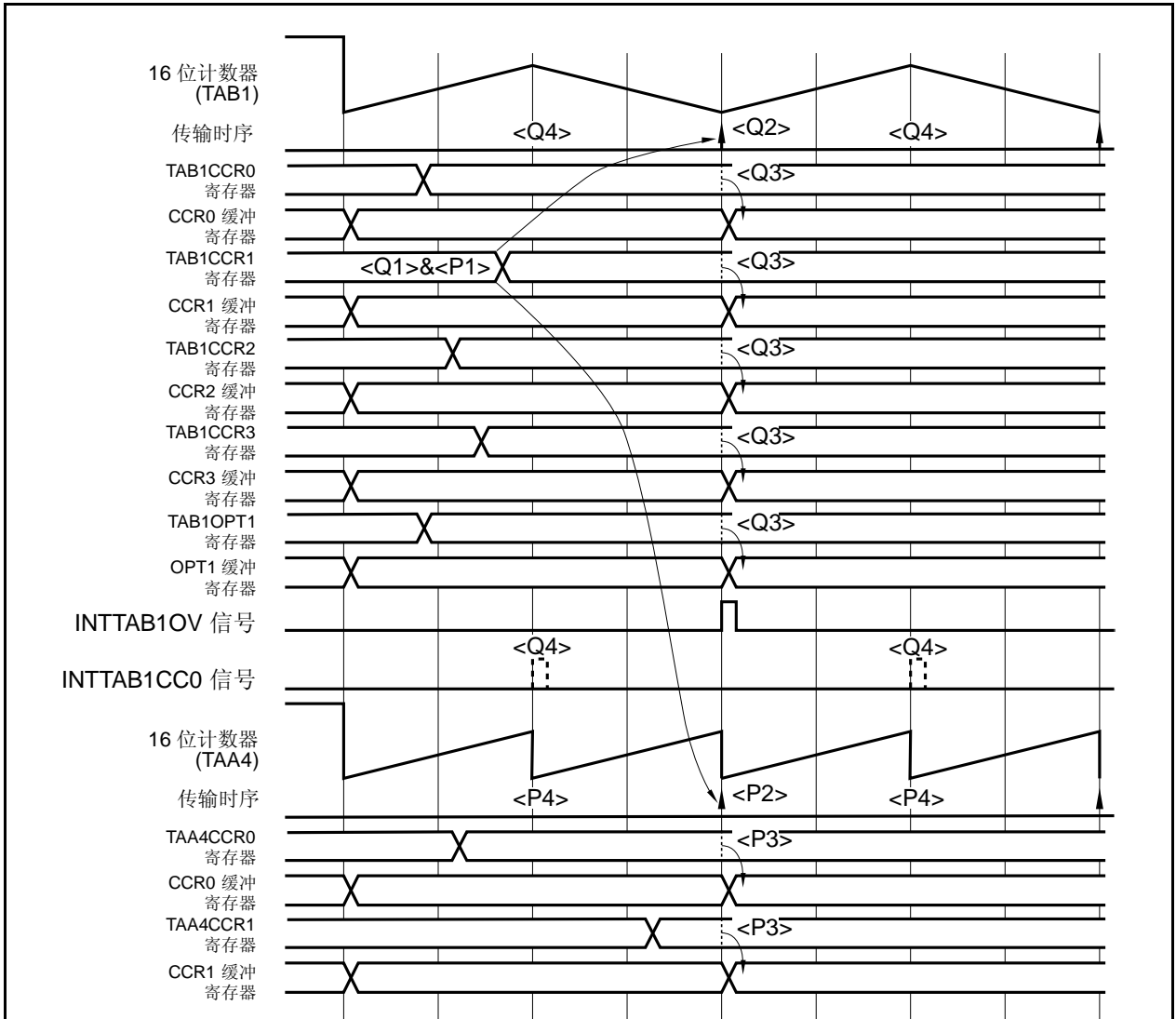
即使无需重写 TAB1CCR1 寄存器，也应该将相同的值重写到该寄存器中。

<3> 保持下次重写为未决状态直至生成传送时序。

产生 INTTAB1OV 或 INTTAB1CC0 中断之后，对寄存器再次进行重写。

<4> 返回步骤<1>。

图 11-31. 间隙批量重写模式的基本操作



[TAB1 操作]

- <Q1> 写入 TAB1CCR1 寄存器。
- <Q2> 在重写 TAB1CCR1 寄存器后产生的传送时序处重写该寄存器。
- <Q3> 在传送时序处这些寄存器全部被立即传送。
- <Q4> 分选中断，则传送时序也被分选。

[TAA4 操作]

- <P1> 写入 TAB1CCR1 寄存器。
- <P2> 在重写 TAB1CCR1 寄存器后产生的传送时序处重写寄存器。
- <P3> 在传送时序处这些寄存器全部被立即传送。
- <P4> 分选中断，则传送时序也被分选。

备注 这是 TAB1OPT1.TAB1ICE 位 = 1，TAB1OPT1.TAB1IOE 位 = 1 且 TAB1OPT1.TAB1ID4 至 TAB1OPT1.TAB1ID0 位 = 00001 时的操作示例。

(b) 重写 TAB1CCR0 寄存器

在间隙批量重写模式中重写 TAB1CCR0 寄存器时，根据中断分选设置指定的波峰或波谷中断，输出波形会有所不同。下图说明了分选中断时输出波形的变化。

图 11-32. 重写 TAB1CCR0 寄存器（设为波峰中断时）

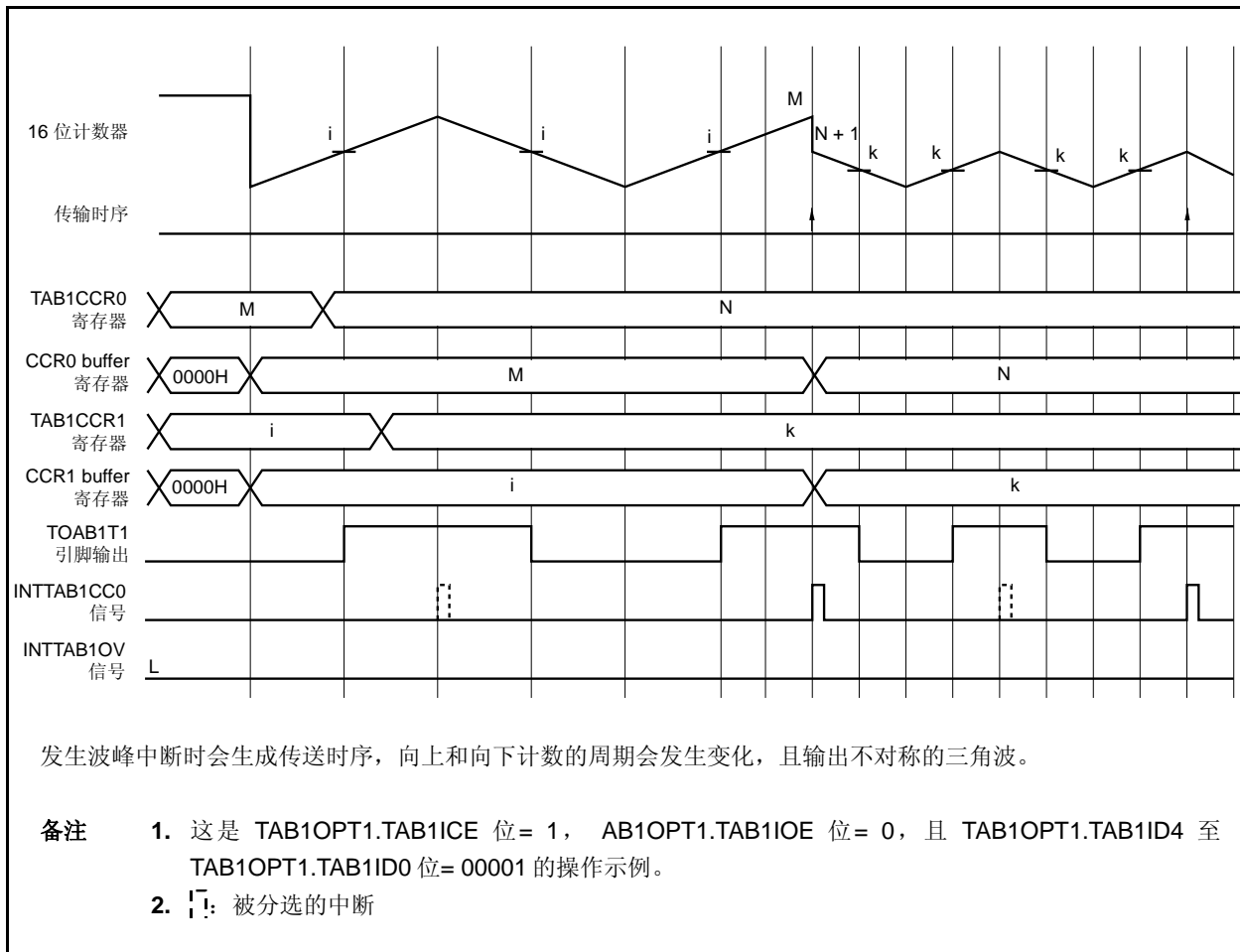
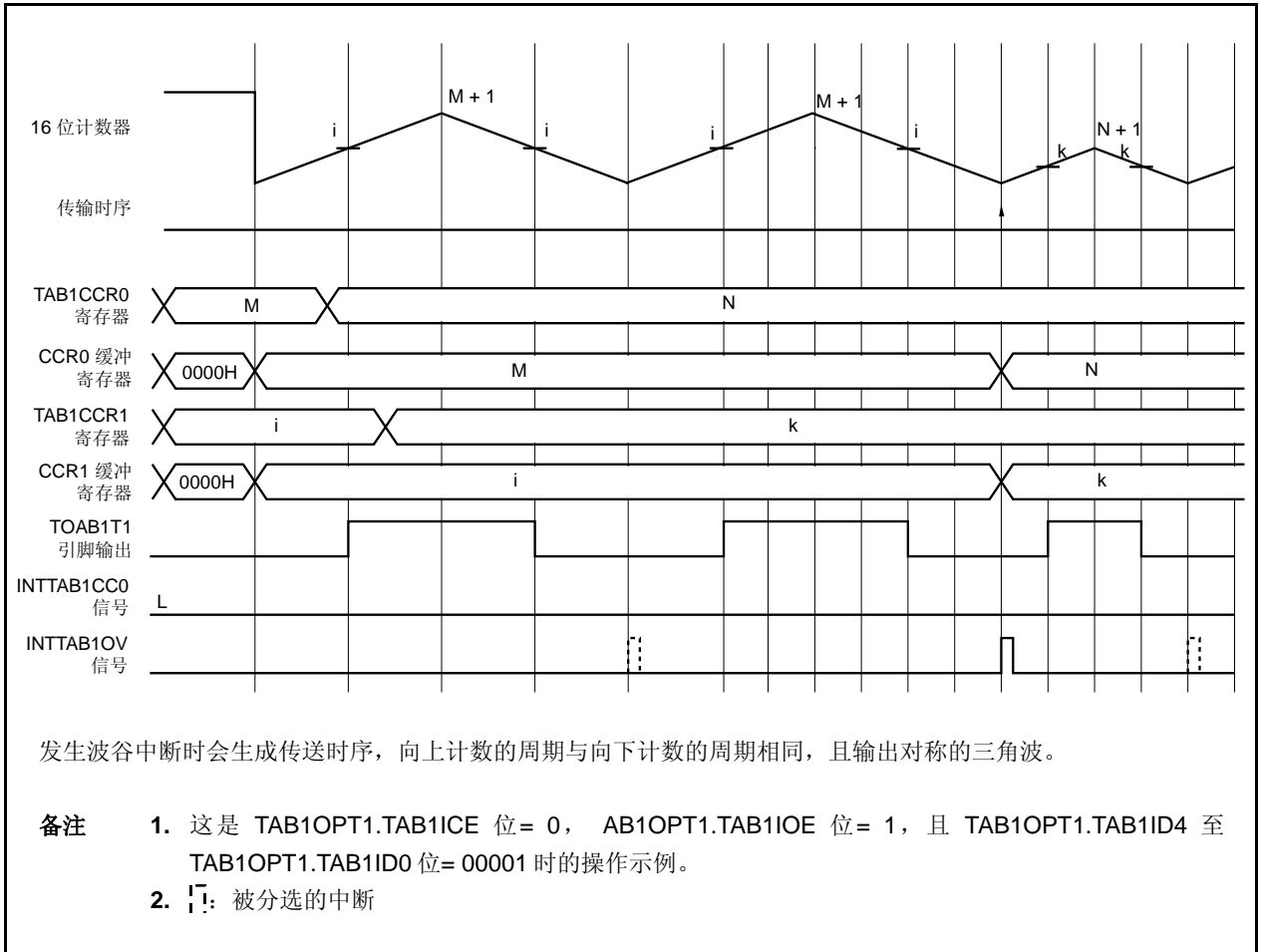


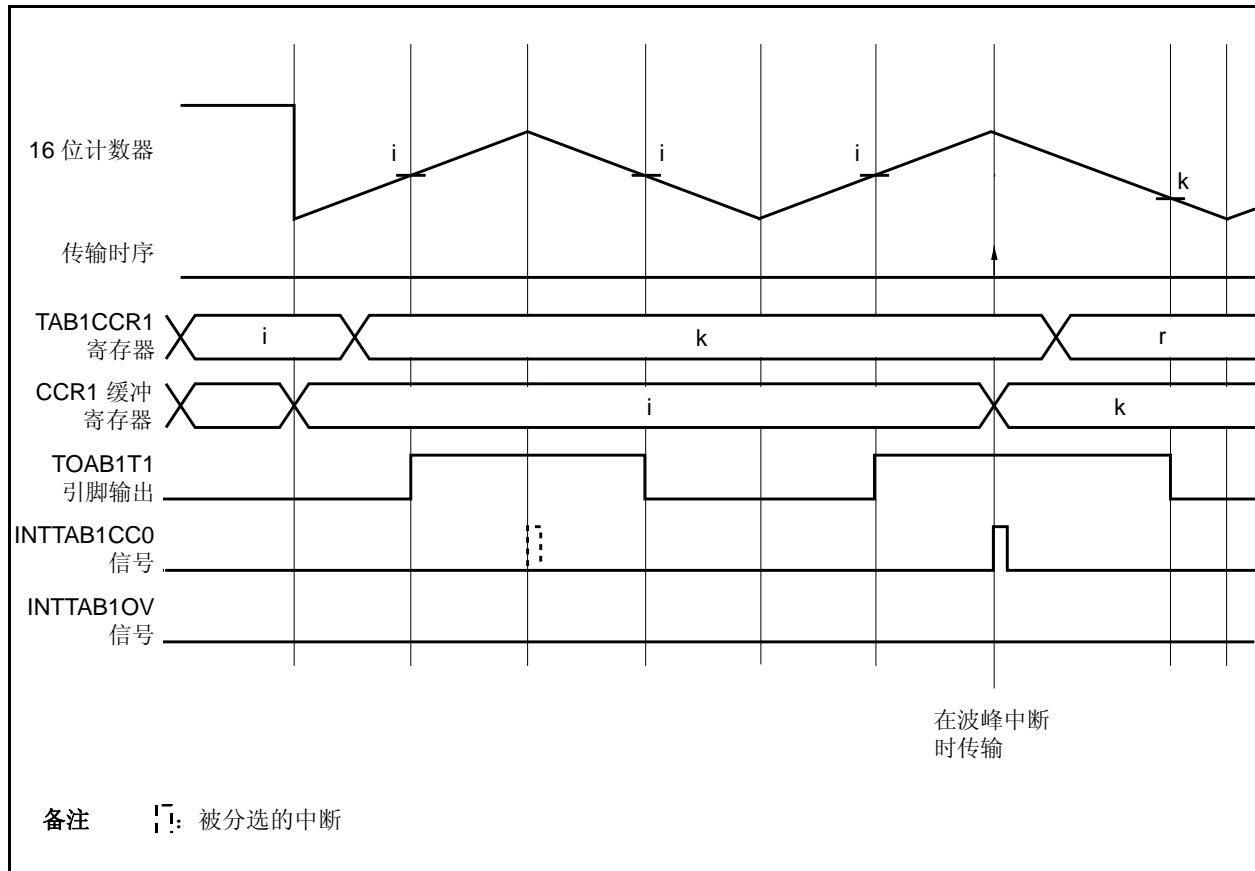
图 11-33. 重写 TAB1CCR0 寄存器（设为波谷中断时）



(c) 重写 TAB1CCR1 至 TAB1CCR3 寄存器

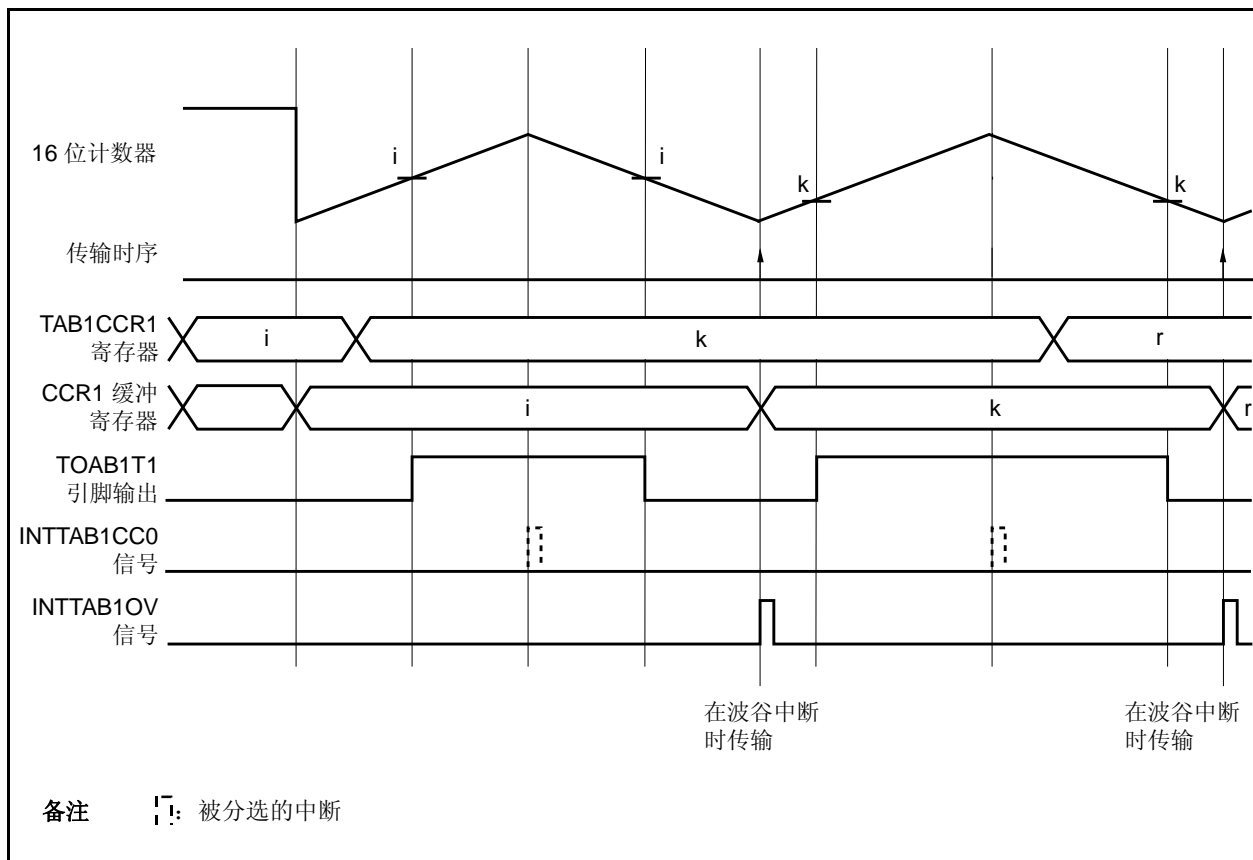
- 设为波峰中断时，在波峰传送
 由于在波峰中断的传送时序处传送这些寄存器，因此会输出不对称的三角波。

图 11-34. 重写 TAB1CCR1 寄存器 (TAB1OPT1.TAB1ICE 位= 1, TAB1OPT1.TAB1IOE 位= 0, TAB1OPT1.TAB1ID4 至 TAB1OPT1.TAB1ID0 位= 00001)



- 设为波谷中断时，在波谷传送
 由于在波谷中断的传送时序处传送这些寄存器，因此会输出对称的三角波。

图 11-35. 重写 TAB1CCR1 寄存器 (TAB1OPT1.TAB1ICE 位 = 1, TAB1OPT1.TAB1IOE 位 = 1, TAB1OPT1.TAB1ID4 至 TAB1OPT1.TAB1ID0 位 = 00001)



(d) 重写 TAB1OPT1 寄存器

当中断分选计数器的值与 16 位计数器的值匹配时，会传送新的中断分选值，因此下次及后续的中断会在设定的间隔时间发生。

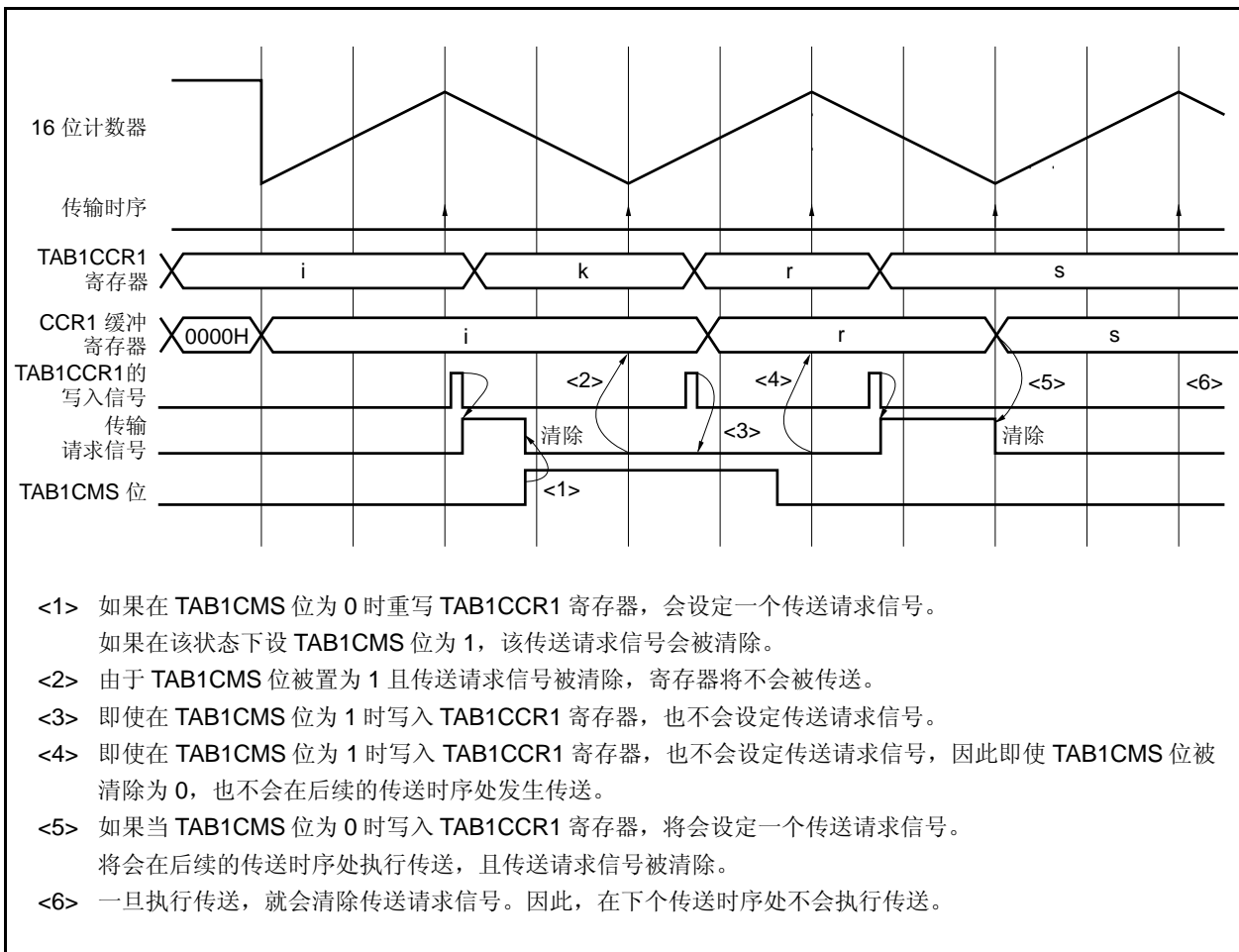
关于重写 TAB1OPT1 寄存器的详情参见 11.4.3 中断分选功能。

(4) 重写 TAB1OPT0.TAB1CMS 位

TAB1CMS 位可以选择随时重写模式和批量重写模式。该位可以在定时器操作过程中 (TAB1CTL0.TAB1CE 位 = 1 时) 进行重写。但是, 图 11-36 中说明的操作和注意事项是必需的。

如果在 TAB1CMS 位被清除为 0 时, 写入 TAB1CCR1 寄存器, 那么将会设定一个传送请求信号 (内部信号)。当设定了传送请求信号时, 那么将会在下个传送时序处传送该寄存器, 且清除传送请求信号。当 TAB1CMS 位被设置为 1 时, 该传送请求信号被清除。

图 11-36. 重写 TAB1CMS 位



- <1> 如果在 TAB1CMS 位为 0 时重写 TAB1CCR1 寄存器, 会设定一个传送请求信号。
如果在该状态下设 TAB1CMS 位为 1, 该传送请求信号会被清除。
- <2> 由于 TAB1CMS 位被置为 1 且传送请求信号被清除, 寄存器将不会被传送。
- <3> 即使在 TAB1CMS 位为 1 时写入 TAB1CCR1 寄存器, 也不会设定传送请求信号。
- <4> 即使在 TAB1CMS 位为 1 时写入 TAB1CCR1 寄存器, 也不会设定传送请求信号, 因此即使 TAB1CMS 位被清除为 0, 也不会在此后的传送时序处发生传送。
- <5> 如果当 TAB1CMS 位为 0 时写入 TAB1CCR1 寄存器, 将会设定一个传送请求信号。
将会在此后的传送时序处执行传送, 且传送请求信号被清除。
- <6> 一旦执行传送, 就会清除传送请求信号。因此, 在下个传送时序处不会执行传送。

11.4.5 用于A/D转换启动触发信号输出的TAA4调谐操作

本节介绍在6相PWM输出模式下TAA4和TAB1的调谐功能。

在6相PWM输出模式下，TAB1为主，TAA4为从，共同执行调谐操作。可以通过TAA4的INTTAA4CC0和INTTAA4CC1信号及TAB1的INTTAB1OV和INTTAB1CC0信号将A/D转换器的转换启动触发信号设置为A/D转换启动触发源。

(1) 调谐操作启动过程

应该按照以下步骤来设置TAA4和TAB1寄存器，用来执行调谐操作。

(a) TAA4寄存器的设置（停止TAB1和TAA4的操作（清除TAB1CTL0.TAB1CE位及TAA4CTL0.TAA4CE位为0））。

- 设置TAA4CTL1寄存器为85H（设置调谐操作从模式以及自由运行定时器模式）。
- 清除TAA4OPT0寄存器为00H（选择比较寄存器）。
- 设置一个合适的值到TAA4CCR0和TAA4CCR1寄存器（为启动操作比较而设定的默认值）。

(b) TAB1寄存器的设置

- 设置TAB1CTL1寄存器为07H（主模式以及6相PWM输出模式）。
- 设置一个合适的值到TAB1IOC0寄存器（设置TOAB1T1至TOAB1T3的输出模式）。
但是，将TAB1OL0位清0并将TAB1OE0位置1（使能正相输出）。除非已经完成该设置，否则不会产生波峰中断（INTTAB1CC0）及波谷中断（INTTAB1OV）。因而，不能正确生成A/D转换器的转换启动触发信号。
- 设置TAB1IOC1和TAB1IOC2寄存器为00H（不使用TAB1的TIAB10至TIAB13、EVTB1及TRGB1引脚）。
- 将TAB1OPT0寄存器清除为00H（选择比较寄存器）。
- 设置一个合适的值到TAB1CCR0至TAB1CCR3寄存器（为启动操作比较而设定的默认值）。
- 设置TAB1CTL0寄存器为0xH（将TAB1CE位清除为0且设置TAB1的操作时钟）。
- 通过TAB1CTL0寄存器设置的TAB1操作时钟也会提供给TAA4，并按相同时序执行计数操作。由TAA4CTL0寄存器设置的TAA4操作时钟被忽略。

(c) TMQOP（TMQ选项）寄存器的设置

- 设置一个合适的值到TAB1OPT1及TAB1OPT2寄存器。
- 设置一个合适的值到TAB1IOC3寄存器（设置TOAB1B1至TOAB1B3为输出模式）。
- 设置一个合适的值到TAB1DTC寄存器（为启动操作比较而设定的默认值）。

(d) 复用功能的设置

- 使用端口控制模式来设置端口的复用功能。

(e) 设置 TAA4CE 位为 1 且立即设置 TAB1CE 位为 1，以启动 6 相 PWM 输出操作

操作期间禁止重写 TAB1CTL0, TAB1CTL1, TAB1IOC1, TAB1IOC2, TAA4CTL0 及 TAA4CTL1 寄存器。如果在操作期间重写了这些寄存器中的任意一个, 无法保证操作, 也无法保证 PWM 输出波形。然后, 允许对 TAB1CTL0.TAB1CE 位的重写, 将其清零。直到 TAA4CTL0.TAA4CE 位被置 1 且随后 TAB1CE 位被置 1, 禁止操作 (读取/写入) TAB1、TAA4 及 TMQ 选项的其它寄存器。

(2) 调谐操作的清除过程

要清除调谐操作并退出 6 相 PWM 输出模式, 应按照以下步骤来设置 TAA4 和 TAB1 寄存器。

- <1> 将 TAB1CTL0.TAB1CE 位清 0, 且停止定时器操作。
- <2> 将 TAA4CTL0.TAA4CE 位清 0, 来分离 TAA4。
- <3> 使用 TAB1IOC0 寄存器停止定时器输出。
- <4> 将 TAA4CTL1.TAA4SYE 位清 0, 清除调谐操作。

注意事项 直到 TAA4CTL0.TAA4CE 位被置 1 且随后 TAB1CE 位被置 1, 禁止操作 (读取/写入) TAB1、TAA4 及 TMQ 选项的其它寄存器。

(3) 没有调谐 TAA4 时

当不需要将 TAA4 的匹配中断信号作为启动 A/D 转换器的转换触发源时, TAA4 可以用作一个不需要调谐的独立的定时器。在这种情况下, TAA4 的匹配中断信号不能在 6 相 PWM 输出模式中被用作启动 A/D 转换的触发源。因此, 应将 TAB1OPT2.TAB1AT0 至 TAB1OPT2.TAB1AT3 位固定为 0。

其他控制位的使用和 TAA4 用于调谐时的方式相同。

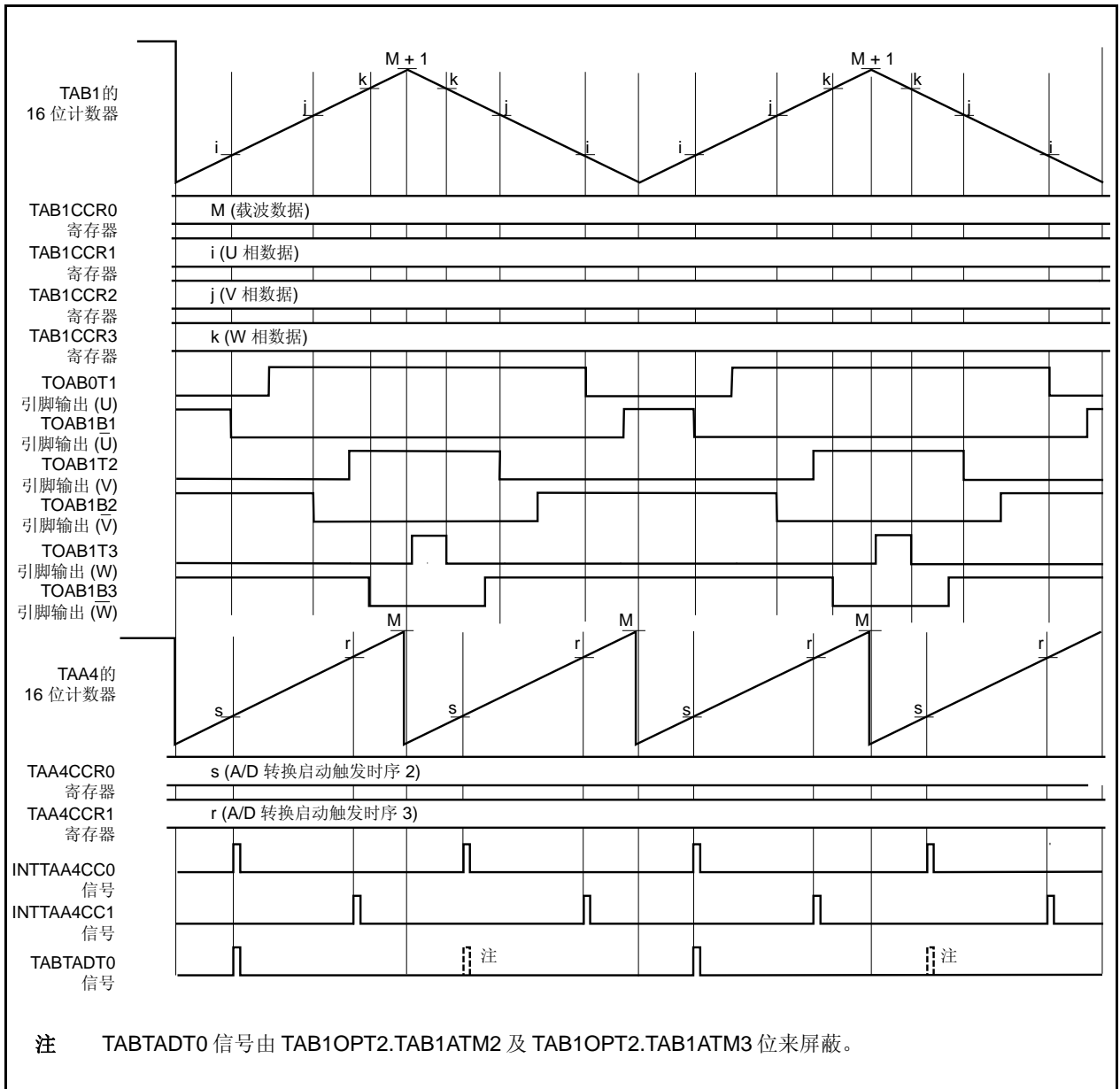
如果 TAA4TAA4 没有用于调谐, 那么 TAA4 的比较寄存器 (TAA4CCR0 及 TAA4CCR1) 将不会受到 TAB1OPT0.TAB1CMS 和 TAB1OPT2.TAB1RDE 位的影响。关于 TAA4 没有用于调谐时的初始化步骤, 请参见 11.4.5 (1) 调谐操作启动过程一节的步骤 (b) 到 (e)。不需要步骤 (a), 因为它是将 TAA4 设置为调谐操作的步骤。

(4) 调谐操作期间 TAA4 的基本操作

TAA4 的 16 位计数器仅能向上计数。16 位计数器通过 TAB1CCR0 寄存器的设定周期值来清除, 并从 0000H 开始再次计数。当 TAB1 的 16 位计数器向上计数时, 它的计数值与该计数器的值相同。但是, 当 TAB1 的 16 位计数器向下计数时, 两个计数值不相同。

- 当 TAB1 向上计数时 (相同的值)
 - TAB1 的 16 位计数器: 0000H → M (向上计数)
 - TAA4 的 16 位计数器: 0000H → M (向上计数)
- 当 TAB1 向下计数时 (不相同的值)
 - TAB1 的 16 位计数器: M + 1 → 0001H (向下计数)
 - TAA4 的 16 位计数器: 0000H → M (向上计数)

图 11-37. 调谐操作期间的 TAA4



11.4.6 A/D转换启动触发输出功能

V850ES/JG3-U 和 V850ES/JH3-U 具有选择四个触发源（INTTAB1OV，INTTAB1CC0，INTTAA4CC0，INTTAA4CC1），可以生成 A/D 转换启动触发信号（TABTADT0）。

通过 TAB1OPT2.TAB1AT0 至 TAB1OPT2.TAB1AT3 位来指定触发源。

- TAB1AT0 位 = 1:
当产生 INTTAB1OV（计数器下溢）时，生成 A/D 转换启动触发信号。
- TAB1AT1 位 = 1:
当产生 INTTAB1CC0（周期匹配）时，生成 A/D 转换启动触发信号。
- TAB1AT2 位 = 1:
当产生 INTTAA4CC0（调谐操作期间 TAA4 的 TAA4CCR0 寄存器匹配）时，生成 A/D 转换启动触发信号。
- TAB1AT3 位 = 1:
当产生 INTTAA4CC1（调谐操作期间 TAA4 的 TAA4CCR1 寄存器匹配）时，生成 A/D 转换启动触发信号。

由 TAB1AT0 至 TAB1AT3 位来选择 A/D 转换启动触发信号，经过或操作后输出。因此，可以同时指定两个或更多个触发源。

由 TAB1AT0 和 TAB1AT1 位选择的 INTTAB1OV 和 INTTAB1CC0 信号是已分选中断信号。

因此，在中断被分选后输出这些信号，除非使能中断输出（通过 TAB1OPT1.TAB1ICE 及 TAB1OPT1.TAB1IOE 位），不会输出 A/D 转换启动触发信号。

如果 TAB1AT2 和 TAB1AT3 位的设置是根据 16 位计数器向上计数 / 向下计数的状态，TAA4 的触发源（INTTAA4CC0 和 INTTAA4CC1）具有屏蔽 A/D 转换启动触发信号的功能。

- TAB1ATM2 位：对应 TAB1AT2 位，且控制 TAA4 的 INTTAA4CC0（匹配中断信号）。
 - TAB1ATM2 位 = 0：当 16 位计数器向上计数（TAB1OPT0.TAB1CUF 位 = 0）时，输出 A/D 转换启动触发信号，当 16 位计数器向下计数（TAB1OPT0.TAB1CUF 位 = 1）时，不输出 A/D 转换启动触发信号。
 - TAB1ATM2 位 = 1：当 16 位计数器向上计数（TAB1OPT0.TAB1CUF 位 = 1）时，输出 A/D 转换启动触发信号，当 16 位计数器向下计数（TAB1OPT0.TAB1CUF 位 = 0）时，不输出 A/D 转换启动触发信号。
- TAB1ATM3 位：对应 TAB1AT3 位，且控制 TAA4 的 INTTAA4CC1（匹配中断信号）。
 - TAB1ATM3 位 = 0：当 16 位计数器向上计数（TAB1OPT0.TAB1CUF 位 = 0）时，输出 A/D 转换启动触发信号，当 16 位计数器向下计数（TAB1OPT0.TAB1CUF 位 = 1）时，不输出 A/D 转换启动触发信号。
 - TAB1ATM3 位 = 1：当 16 位计数器向上计数（TAB1OPT0.TAB1CUF 位 = 1）时，输出 A/D 转换启动触发信号，当 16 位计数器向下计数（TAB1OPT0.TAB1CUF 位 = 0）时，不输出 A/D 转换启动触发信号。

在定时器操作时，可以重写 TAB1ATM3，TAB1ATM2 及 TAB1AT3 至 TAB1AT0 位。如果在定时器操作时重写设置 A/D 转换启动触发信号的位，那么新的设置会立即反映在 A/D 转换启动触发信号的输出状态上。这些控制位不具有传送功能，并且只能在随时重写模式下使用。

- 注意事项
1. 只有在 TAA4 被用作调谐操作中的 TAB1 的从定时器时，由 TAB1AT2 及 TAB1AT3 位设置的 A/D 转换启动触发信号才能输出。如果 TAB1 和 TAA4 没有执行调谐操作，或者没有使用 6 相 PWM 输出模式，那么输出无法保证。
 2. TAB1 信号输出用于在内部识别 16 位计数器是向上计数还是向下计数。因此，通过将 TAB1IOC0.TAB1OL0 位清零且设置 TAB1IOC0.TAB1OE0 位为 1，可以使能 TOAB10 引脚输出。

图 11-38. A/D 转换启动触发 (TABTADT0) 信号输出示例
 (TAB1OPT1.TAB1ICE 位 = 1, TAB1OPT1.TAB1IOE 位 = 1,
 TAB1OPT1.TAB1ID4 至 TAB1OPT1.TAB1ID0 位= 00000: 没有中断分选)

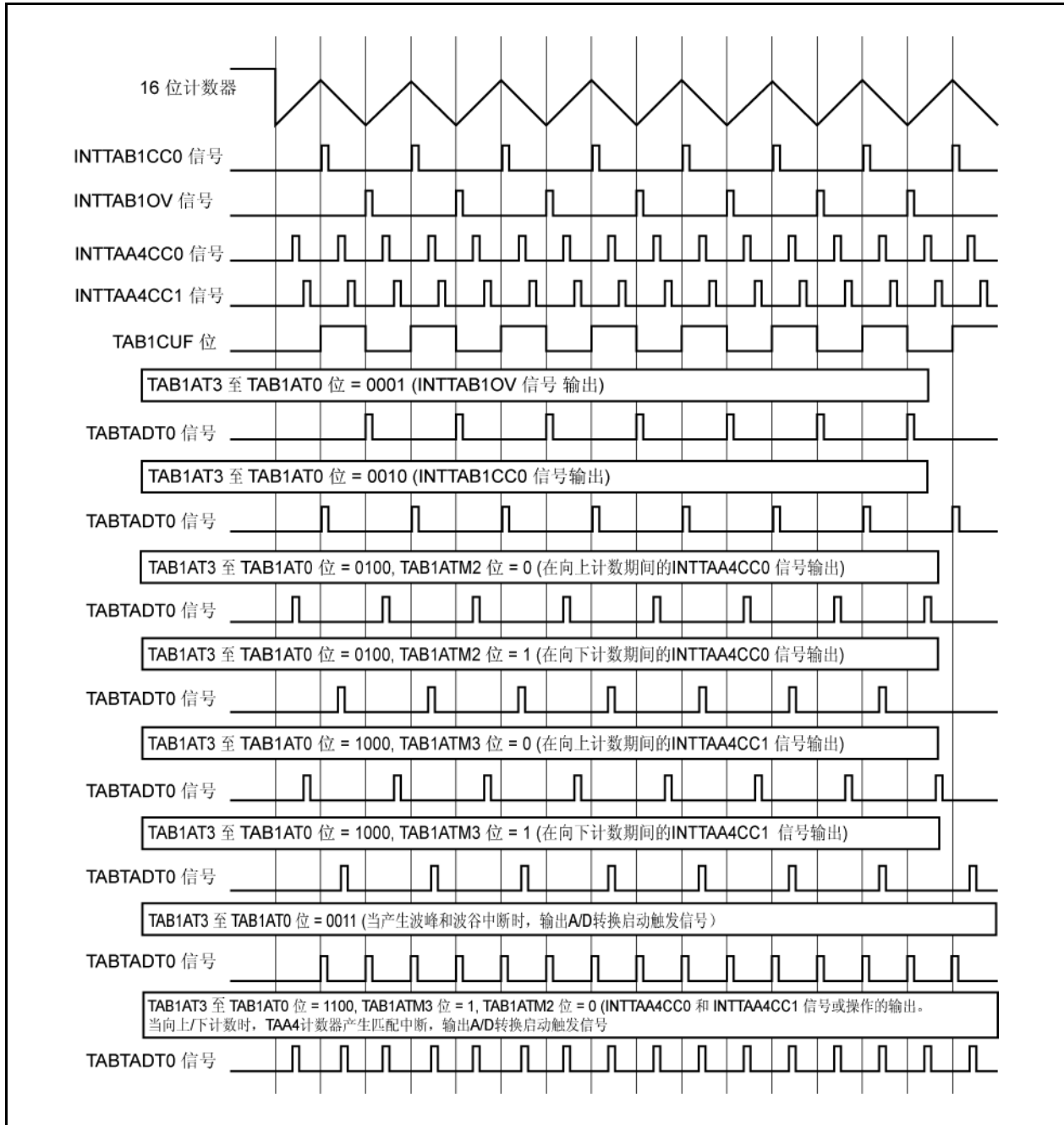


图 11-39. A/D 转换启动触发 (TABTADT0) 信号输出示例
 (TAB1OPT1.TAB1ICE 位 = 0, TAB1OPT1.TAB1IOE 位 = 1,
 TAB1OPT1.TAB1ID4 至 TAB1OPT1.TAB1ID0 位 = 00010: 带有中断分选) (1)

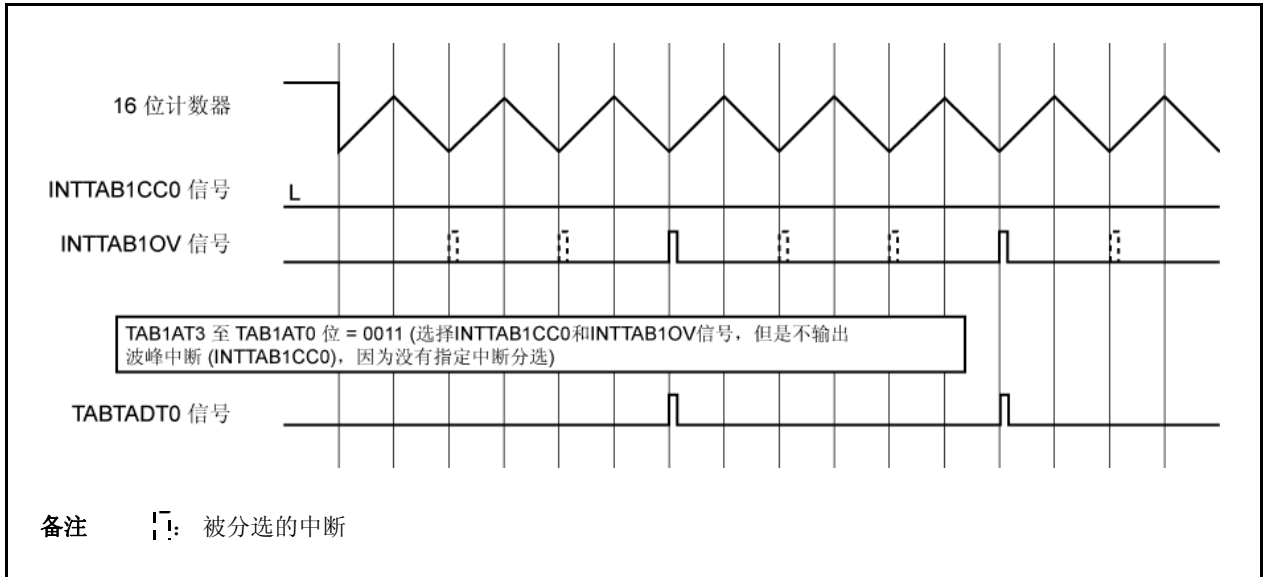
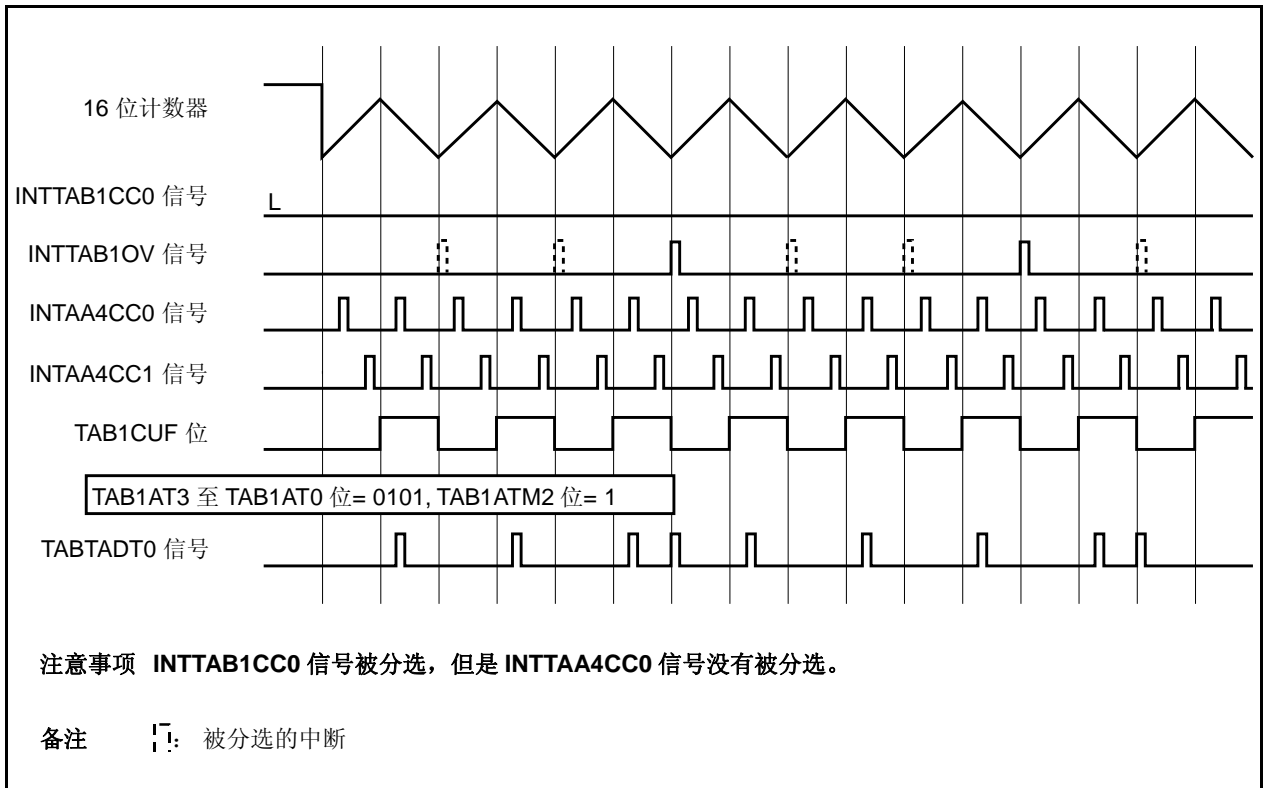


图 11-40. A/D 转换启动触发 (TABTADT0) 信号输出示例
 (TAB1OPT1.TAB1ICE 位 = 0, TAB1OPT1.TAB1IOE 位 = 1,
 TAB1OPT1.TAB1ID4 至 TAB1OPT1.TAB1ID0 位 = 00010: 带有中断分选) (2)



(1) 边界条件下的操作 (16 位计数器与 INTTAA4CC0 信号匹配时的操作)

表 11-3. 当 TAB1CCR0 寄存器 = M, TAB1AT2 位 = 1, TAB1ATM2 位 = 0 时的操作
(选择向上计数期间)

TAA4CCR0 寄存器的值	TAB1 的 16 位计数器的值	TAA4 的 16 位计数器的值	TAB1 的 16 位计数器的状态	由 INTTAA4CC0 信号输出的 TABTADT0 信号
0000H	0000H	0000H	-	输出
0000H	M + 1	0000H	-	不输出
0001H	0001H	0001H	向上计数	输出
0001H	M	0001H	向下计数	不输出
M	M	M	向上计数	输出
M	0001H	M	向下计数	不输出

表 11-4. 当 TAB1CCR0 寄存器 = M, TAB1AT2 位 = 1, TAB1ATM2 位 = 1 的操作
(选择向下计数期间)

TAA4CCR0 寄存器的值	TAB1 的 16 位计数器的值	TAA4 的 16 位计数器的值	TAB1 的 16 位计数器的状态	由 INTTAA4CC0 信号输出的 TABTADT0 信号
0000H	0000H	0000H	-	不输出
0000H	M + 1	0000H	-	输出
0001H	0001H	0001H	向上计数	不输出
0001H	M	0001H	向下计数	输出
M	M	M	向上计数	不输出
M	0001H	M	向下计数	输出

注意事项 当 TAB1CCR0 寄存器 = M 时, TAA4CCRm 寄存器允许“0”至“M”的数值设置。禁止设置“M + 1”或更大的值。

如果设置的值等于或大于“M + 1”, TAA4 的 16 位计数器通过“M”来清除。所以将不会输出 TABTADT0 信号。

备注 m = 0, 1

第十二章 实时计数器

12.1 实时计数器的功能

实时计数器（RTC）有如下特点。

- 使用内置的年、月、星期、日、小时、分钟和秒计数器，可以计数达 99 年
- 使用 BCD 码^注显示年、月、星期、日、小时、分钟和秒计数器
- 闹钟中断功能
- 固定周期中断功能（周期：1 个月至 0.5 秒）
- 间隔中断功能（周期：1.95 ms 至 125 ms）
- 1Hz 引脚输出功能
- 32.768kHz 引脚输出功能
- 512HZ 或 16.384kHz 引脚输出功能
- 钟表误差修正功能
- 可以选择运行于副时钟或主时钟

注 1. BCD 码（二进制编码的十进制）用 4 位二进制格式来表示十进制数字的每一位数字。
2. 使用实时定时器的专用波特率发生器将主时钟分频为 32.768kHz。

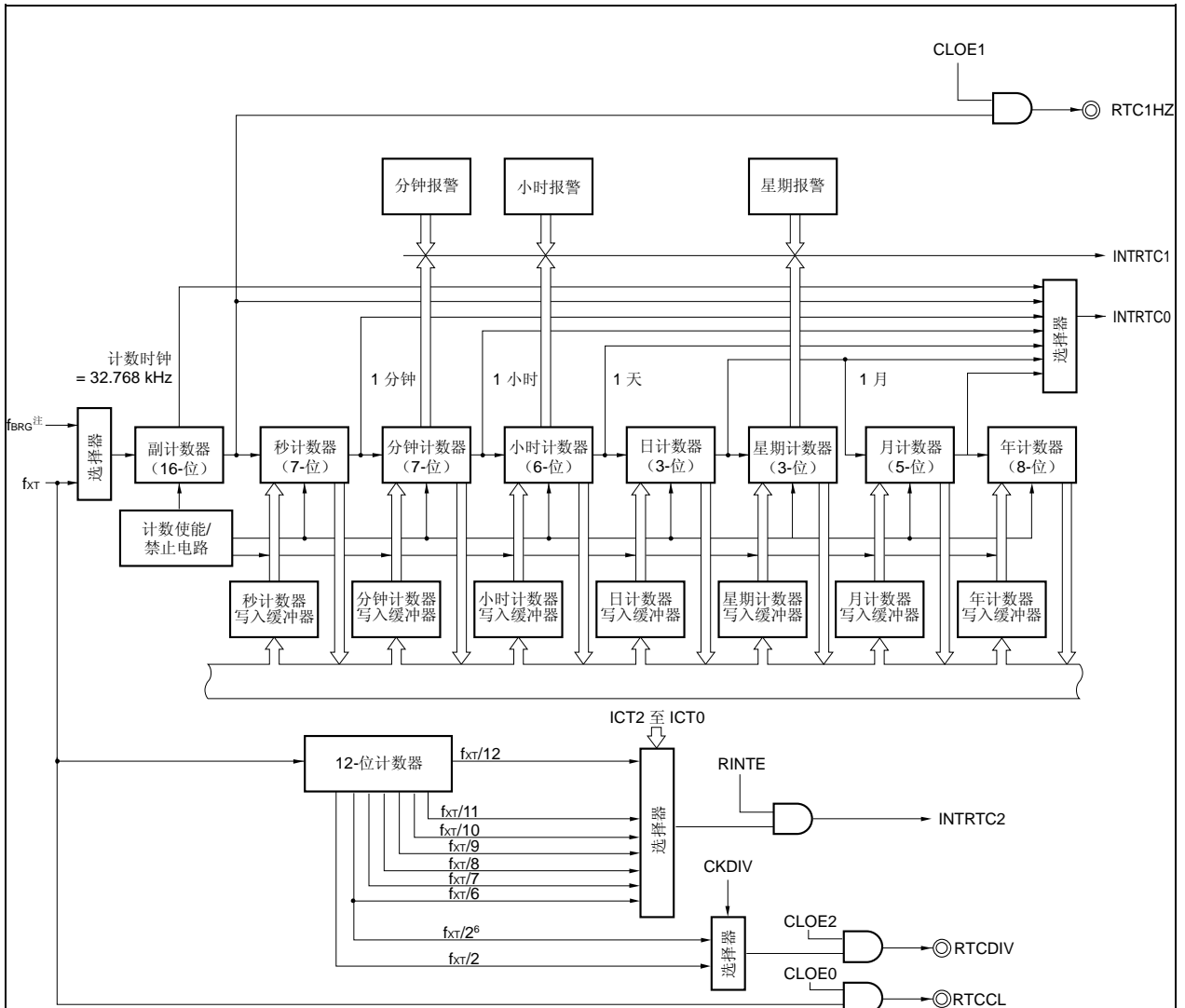
12.2 实时计数器的配置

实时计数器包括下列硬件。

表 12-1. 实时计数器的配置

项目	配置
控制寄存器	实时计数器控制寄存器 0 (RC1CC0)
	实时计数器控制寄存器 1 (RC1CC1)
	实时计数器控制寄存器 2 (RC1CC2)
	实时计数器控制寄存器 3 (RC1CC3)
	副计数寄存器 (RC1SUBC)
	秒计数寄存器 (RC1SEC)
	分钟计数寄存器 (RC1MIN)
	小时计数寄存器 (RC1HOUR)
	日计数寄存器 (RC1DAY)
	星期计数寄存器 (RC1WEEK)
	月计数寄存器 (RC1MONTH)
	年计数寄存器 (RC1YEAR)
	钟表误差修正寄存器 (RC1SUBU)
	分钟闹钟寄存器(RC1ALM)
	小时闹钟寄存器(RC1ALH)
	星期闹钟寄存器(RC1ALW)
	预分频模式寄存器 0(PRSM0)
	预分频比较寄存器 0(PRSCM0)

图 12-1. 实时计数器的框图



注 fBRG的详情参见12.3 (17) 预分频模式寄存器0 (PRSM0) 和 12.3 (18) 预分频比较寄存器0 (PRSCM0)。

- 备注
- fBRG: 实时计数器计数时钟频率
 - fXT: 副时钟频率
 - INTRTC0: 实时计数器固定周期信号
 - INTRTC1: 实时计数器闹钟匹配信号
 - INTRTC2: 实时计数器间隔信号

12.2.1 引脚配置

实时计数器中包括的 RTC 输出可以复用，如表 12-2 所示。当使用每个引脚时必须设置为端口功能（参见表 4-20 端口引脚用作复用功能引脚）。

表 12-2. 引脚配置

引脚编号		端口	RTC输出	其他复用功能
V850ES/JG3-U	V850ES/JH3-U			
30	42	P35	RTC1HZ	TIAA11/TOAA11/ \overline{OCI}
28	40	P33	RTCDIV	TIAA01/TOAA01/RTCCL
28	40	P33	RTCCL	TIAA01/TOAA01/RTCDIV

12.2.2 中断功能

RTC 包括下列三种类型的中断信号。

(1) INTRTC0

每 0.5 秒，每秒，每分钟，每小时，每天或者每月产生一个固定周期中断信号。

(2) INTRTC1

闹钟中断信号

(3) INTRTC2

产生周期为 $f_{XT}/6$, $f_{XT}/7$, $f_{XT}/8$, $f_{XT}/9$, $f_{XT}/10$, $f_{XT}/11$, 或 $f_{XT}/12$ 的间隔中断信号。

12.3 控制实时计数器的寄存器

以下 18 个寄存器用于控制实时计数器。

(1) 实时计数器控制寄存器 0 (RC1CC0)

RC1CC0 寄存器是 8 位寄存器，用于选择实时计数器的输入时钟。

该寄存器可以按字节或按位进行读取或写入。

系统复位后，该寄存器被置为 00H。

复位后: 00H R/W 地址: FFFFFADDH									
符号	7	6	5	4	3	2	1	0	
RC1CC0	RC1PWR	RC1CKS	0	0	0	0	0	0	
	RC1PWR	实时计数器工作控制							
	0	停止实时计数器操作。							
	1	使能实时计数器操作。							
	RC1CKS	工作时钟选择							
	0	选择 f _{XT} 作为工作时钟							
	1	选择 f _{BRG} 作为工作时钟							
<p>注意事项</p> <ol style="list-style-type: none"> 1. 当实时计数器工作时需要停止其操作(RC1PWR = 1 → 0)，请遵循12.4.8 实时计数器初始化的描述。 2. 只有实时计数器停止(RC1PWR 位 = 0)时，才可以写入RC1CKS位。并且，禁止在RC1PWR 位从0被设置为1的同事重写RC1CKS位。 									

(2) 实时计数器控制寄存器 0 (RC1CC1)

RC1CC1 寄存器是 8 位寄存器，用于启动或停止实时计数器操作、控制 RTCCL 和 RTC1HZ 引脚，设置 12 小时或 24 小时系统，并设置固定周期中断功能。

该寄存器可以按字节或按位进行读取或写入。

系统复位后，该寄存器被置为 00H。

复位后: 00H R/W 地址: FFFFFADEH

符号	7	6	5	4	3	2	1	0
RC1CC1	RTCE	0	CLOE1	CLOE0	AMPM	CT2	CT1	CT0
RTCE		控制每个计数器的工作						
0		停止计数器操作。						
1		使能计数器操作。						
CLOE1		RTC1HZ引脚输出控制						
0		禁止 RTC1HZ 引脚输出 (1Hz)						
1		使能 RTC1HZ 引脚输出 (1Hz)						
CLOE0		RTCCL引脚输出控制						
0		禁止 RTCCL 引脚输出 (32.768kHz)						
1		使能 RTCCL 引脚输出 (32.768kHz)						
AMPM		选择12小时/24小时系统						
0		12 小时系统(a.m. 和 p.m. 显示)						
1		24 小时系统						
CT2	CT1	CT0	选择固定周期中断(INTRTC0)					
0	0	0	不使用固定周期中断					
0	0	1	每 0.5 秒一次(与秒累加同步)					
0	1	0	每 1 秒一次(与秒累加同时)					
0	1	1	每 1 分钟一次(每分钟的 00 秒)					
1	0	0	每 1 小时一次 (每小时的 00 分钟 00 秒)					
1	0	1	每 1 日一次 (每日的 00 小时 00 分钟 00 秒)					
1	1	×	每 1 月一次 (每月的某天 a.m 00 小时 00 分 00 秒)					

- 注意事项**
- 当RTCE 位置1时, 禁止向RTCE位写入0。按照12.4.8 实时计数器初始化的描述, 通过清除RC1PWR 来对RTCE位置0。
 - 当CLOE1位的设置改变时, RTC1HZ输出的操作如下:
 - 从0改变为1时: 在两个时钟内, RTC1HZ输出一个1Hz脉冲。
 - 从1改变为0时: 在两个时钟内, RTC1HZ输出停止(固定为低电平)。
 - 关于设置或改变AMPM位, 参见12.4.1初始化设置和12.4.2在时钟工作期间重写每个计数器。并且, 当AMPM位被重写时, RC1HOUR寄存器也被重置。
 - 关于在实时计数器工作(RC1PWR 位 = 1)时重写CT2至CT0位, 参见12.4.4 在时钟工作期间改变INTRTC0中断设置。

(3) 实时计数器控制寄存器 2 (RC1CC2)

RC1CC2 寄存器是 8 位寄存器，用于控制闹钟中断功能和计数器等待时间。

该寄存器可以按字节或按位进行读取或写入。

系统复位后，该寄存器被置为 00H。

复位后: 00H R/W 地址: FFFFFADFH							
符号	7	6	5	4	3	2	1 0
RC1CC2	WALE	0	0	0	0	0	RWST RWAIT
WALE	闹钟中断 (INTRTC1) 操作控制						
0	在闹钟匹配时不产生中断						
1	在闹钟匹配时产生中断						
RWST	实时计数器等待状态						
0	计数器操作中						
1	从秒到年计数器向上计数停止 (可以读或写计数器的值)						
该状态标志显示 RWAIT 位的设置是否有效。 读或写计数器值之前，请确认 RWST 位为 1。							
RWAIT	实时计数器等待控制						
0	设置计数器操作						
1	停止从秒到年计数器的计数 (可以读或写计数器的值)						
该位控制计数器的操作。 当读或写计数器值时，确保该位写入“1”。 如果在 RWAIT = 1 时 RSUBC 寄存器溢出，溢出信息被保留在内部，并且在 RWAIT 位被写入 0 之后的两个时钟内，RC1SEC 寄存器向上计数。 但是，如果 RWAIT 位为 1 时重写秒计数器的值，保留的溢出信息被丢弃。							
<p>注意事项</p> <ol style="list-style-type: none"> 关于在实时计数器工作(RC1PWR 位 = 1)时重写WALE位，参见12.4.5 在时钟工作期间改变INTRTC1中断设置。 当读取或写入每个计数器值时，确认RWST位被置1。 当每个计数器被写入时，RWST位不变为0，即使RWAIT位已经被置0。当每个计数器写入完成后，该位变为0。 							

(4) 实时计数器控制寄存器 2 (RC1CC3)

RC1CC3 寄存器是 8 位寄存器，用于控制间隔中断功能和 RTCDIV 引脚。

该寄存器可以按字节或按位进行读取或写入。

系统复位后，该寄存器被置为 00H。

复位后: 00H R/W 地址: FFFFFAE0H

符号	7	6	5	4	3	2	1	0
RC1CC3	RINTE	CLOE2	CKDIV	0	0	ICT2	ICT1	ICT0

RINTE	间隔中断 (INTRTC2) 控制
0	不产生间隔中断
1	产生间隔中断

CLOE2	RTCDIV引脚输出控制
0	禁止 RTCDIV 引脚输出
1	使能 RTCDIV 引脚输出

CKDIV	RTCCL引脚输出控制
0	RTCCL 引脚输出 512HZ (1.95 ms)
1	RTCCL 引脚输出 16.384kHz (0.061 ms)

ICT2	ICT1	ICT0	选择间隔中断(INTRTC2)
0	0	0	$2^9/f_{XT}$ (1.953125 ms)
0	0	1	$2^7/f_{XT}$ (3.90625 ms)
0	1	0	$2^8/f_{XT}$ (7.8125 ms)
0	1	1	$2^9/f_{XT}$ (15.625 ms)
1	0	0	$2^{10}/f_{XT}$ (31.25 ms)
1	0	1	$2^{11}/f_{XT}$ (62.5 ms)
1	1	×	$2^{12}/f_{XT}$ (125 ms)

注意事项 1. 关于在实时计数器工作 (RC1PWR位= 1) 时重写RINTE位, 参见12.4.7 在时钟工作期间改变 INTRTC2中断设置。

2. 当CLOE2位的设置改变时, RTCDIV输出的操作如下:

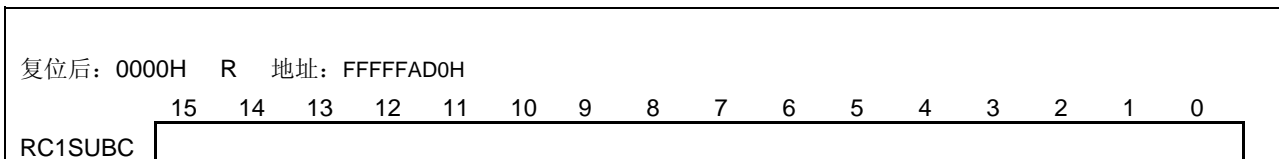
- 从0改变为1时: 在两个时钟内, 输出CKDIV位设置的脉冲。
- 从1改变为0时: 在两个时钟内, RTCDIV输出停止 (固定为低电平)。

3. 关于在实时计数器工作 (RC1PWR位= 1) 时重写ICT2至ICT0位, 参见12.4.7 在时钟工作期间改变 INTRTC2中断设置。

(5) 副计数寄存器 (RSUBC)

RSUB 寄存器是 16 位寄存器，用于计数实时计数器的 1 秒参考时间。它的值是 0000H 至 7FFFH，用 32.768kHz 的时钟来计数 1 秒钟。
 该寄存器是只读的，按照 16 位宽度。
 系统复位后，该寄存器被置为 0000H。

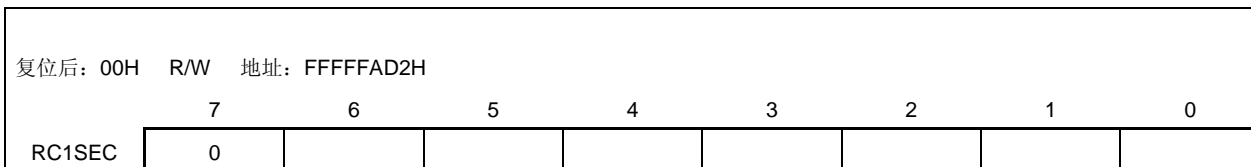
- 注意事项**
1. 当使用 RC1SUBU 寄存器修正，值可能变为 8000H 或更大。
 2. 写入秒计数寄存器，该寄存器被清零。
 3. 如果在操作期间读取该寄存器，不能保证读出值的有效性，因为读到的是变化值。



(6) 秒计数寄存器 (RC1SEC)

RC1SEC 寄存器是 8 位寄存器，它的值为 0 至 59（十进制），表示秒的当前计数值。
 当副计数器溢出时，它向上计数。
 当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（32.768 kHz）后被写入计数器。以 BCD 码格式设置 00 至 59 的十进制数到该寄存器。如果设置值超出范围，寄存器值在 1 个周期后返回正常值。
 该寄存器可以按字节进行读取或写入。
 系统复位后，该寄存器被置为 00H。

备注 当读取或写入 RC1SEC 寄存器时，参见 12.4.1 初始化设置 12.4.2 在时钟工作期间重写每个计数器和 12.4.3 在时钟工作期间读取各个计数器。



(7) 分钟计数寄存器 (RC1MIN)

RC1MIN 寄存器是 8 位寄存器，它的值为 0 至 59（十进制），表示分钟的当前计数值。

当秒计数器溢出时，它向上计数。

当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（32.768 kHz）后被写入计数器。以 BCD 码格式设置 00 至 59 的十进制数到该寄存器。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被置为 00H。

注意事项 禁止对 RC1MIN 寄存器设置 00 至 59 之外的值。

备注 当读取或写入 RC1MIN 寄存器时，参见 12.4.1 初始化设置 12.4.2 在时钟工作期间重写每个计数器和 12.4.3 在时钟工作期间读取各个计数器。

复位后: 00H R/W 地址: FFFFFAD3H								
	7	6	5	4	3	2	1	0
RC1MIN	0							

(8) 小时计数寄存器 (RC1HOUR)

RC1HOUR 寄存器是 8 位寄存器，它的值为 0 至 23 或 1 至 12（十进制），表示小时的当前计数值。

当分钟计数器溢出时，它向上计数。

当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（32.768 kHz）后被写入计数器。以 BCD 码格式设置 00 至 59 的十进制数到该寄存器。

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。以 BCD 码设置 0 至 23，01 至 12 或 21 至 32 到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被置为 12H。

如果复位后 AMPM 位被置为 1，该寄存器的值清为 00H。

- 注意事项**
1. 当 AMPM = 0 (如果选择 12 小时系统)时，RC1HOUR 寄存器的第 5 位表示 a.m. (0) 或 p.m. (1)。
 2. 禁止对 RC1HOUR 寄存器设置 01 至 12，21 至 32 (AMPM bit= 0)或 00 至 32(AMPM bit = 1) 之外的值。

备注 当读取或写入 RC1HOUR 寄存器时，参见 12.4.1 初始化设置 12.4.2 在时钟工作期间重写每个计数器和 12.4.3 在时钟工作期间读取各个计数器。

复位后: 12H R/W 地址: FFFFFAD4H								
	7	6	5	4	3	2	1	0
RC1HOUR	0	0						

表 12-3 展示了 AMPM 位设置值，RC1HOUR 寄存器值和时间之间的关系。

表 12-3. 时间数字显示

12小时显示(AMPM 位 = 0)		24小时显示(AMPM 位 = 1)	
时间	RC1HOUR 寄存器值	时间	RC1HOUR 寄存器值
0:00 a.m.	12 H	0:00	00H
1:00 a.m.	01 H	1:00	01 H
2:00 a.m.	02 H	2:00	02 H
3:00 a.m.	03 H	3:00	03 H
4:00 a.m.	04 H	4:00	04 H
5:00 a.m.	05 H	5:00	05 H
6:00 a.m.	06 H	6:00	06 H
7:00 a.m.	07 H	7:00	07 H
8:00 a.m.	08 H	8:00	08 H
9:00 a.m.	09 H	9:00	09 H
10:00 a.m.	10 H	10:00	10 H
11:00 a.m.	11 H	11:00	11 H
0:00 p.m.	32 H	12:00	12 H
1:00 p.m.	21 H	13:00	13 H
2:00 p.m.	22 H	14:00	14 H
3:00 p.m.	23 H	15:00	15 H
4:00 p.m.	24 H	16:00	16 H
5:00 p.m.	25 H	17:00	17 H
6:00 p.m.	26 H	18:00	18 H
7:00 p.m.	27 H	19:00	19 H
8:00 p.m.	28 H	20:00	20 H
9:00 p.m.	29 H	21:00	21 H
10:00 p.m.	30 H	22:00	22 H
11:00 p.m.	31 H	23:00	23 H

如果 AMPM 位为 0，RC1HOUR 寄存器值的显示是 12 小时格式，当 AMPM 位为 1 时，显示是 24 小时格式。在 12 小时格式显示中，a.m.或 p.m. 通过 RCHOUR 的第 5 位来表示：0 表示中午之前(a.m.)，1 表示中午或下午(p.m.)。

(9) 日计数寄存器 (RC1DAY)

RC1DAY 寄存器是 8 位寄存器，它的值为 1 至 31（十进制）表示天的当前计数值。

当小时计数器溢出时，它向上计数。

计数器计数如下。

- 01 至 31 (一月、三月、五月、七月、八月、十月、十二月)
- 01 至 30 (四月、六月、九月、十一月)
- 01 至 29 (闰年的二月)
- 01 至 28 (正常年的二月)

当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（32.768 kHz）后被写入计数器。以 BCD 码格式设置 01 至 31 的十进制数到该寄存器。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被置为 01H。

注意事项 禁止对 RC1DAY 寄存器设置 01 至 31 之外的值。设置值不在上述计数范围内，比如“二月 30”也同样被禁止。

备注 当读取或写入 RC1DAY 寄存器时，参见 12.4.1 初始化设置 12.4.2 在时钟工作期间重写每个计数器和 12.4.3 在时钟工作期间读取各个计数器。

复位后: 01H R/W 地址: FFFFAD6H								
	7	6	5	4	3	2	1	0
RC1DAY	0	0						

(10) 星期计数寄存器 (RC1WEEK)

RC1WEEK 寄存器是 8 位寄存器，它的值为 0 至 6（十进制）表示星期的当前计数值。

当日计数器溢出时，它向上计数。

当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（32.768 kHz）后被写入计数器。以 BCD 码格式设置 00 至 06 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被置为 00H。

复位后: 00H R/W 地址: FFFFFAD5H								
	7	6	5	4	3	2	1	0
RC1WEEK	0	0	0	0	0			

- 注意事项**
1. 禁止对 RC1WEEK 寄存器设置 00 至 06 之外的值。
 2. 月计数寄存器和天计数寄存器的对应值不会自动存储到星期寄存器。
在复位解除后，请确保设置如下。

星期几	WEEK
星期日	00H
星期一	01H
星期二	02H
星期三	03H
星期四	04H
星期五	05H
星期六	06H

备注 当读取或写入 RC1WEEK 寄存器时，参见 12.4.1 初始化设置 12.4.2 在时钟工作期间重写每个计数器和 12.4.3 在时钟工作期间读取各个计数器。

(11) 月计数寄存器 (RC1MONTH)

RC1MONTH 寄存器是 8 位寄存器，它的值为 1 至 12（十进制）表示月的当前计数值。

当日计数器溢出时，它向上计数。

当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（32.768 kHz）后被写入计数器。以 BCD 码格式设置 01 至 12 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被置为 01H。

注意事项 禁止对 RC1MONTH 寄存器设置 01 至 12 之外的值。

备注 当读取或写入 RC1MONTH 寄存器时，参见 12.4.1 初始化设置 12.4.2 在时钟工作期间重写每个计数器和 12.4.3 在时钟工作期间读取各个计数器。

复位后: 01H R/W 地址: FFFFFAD7H								
	7	6	5	4	3	2	1	0
RC1MONTH	0	0	0					

(12) 年计数寄存器 (RC1YEAR)

RC1YEAR 寄存器是 8 位寄存器，它的值为 0 至 99（十进制）表示年的当前计数值。

当月计数器溢出时，它向上计数。

00、04、08...92 和 96 表示闰年。

当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（32.768 kHz）后被写入计数器。以 BCD 码格式设置 00 至 99 的十进制数到该寄存器。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被置为 00H。

注意事项 禁止对 RC1YEAR 寄存器设置 00 至 99 之外的值。

备注 当读取或写入 RC1YEAR 寄存器时，参见 12.4.1 初始化设置 12.4.2 在时钟工作期间重写每个计数器和 12.4.3 在时钟工作期间读取各个计数器。

复位后: 00H R/W 地址: FFFFFAD8H								
	7	6	5	4	3	2	1	0
RC1YEAR								

(13) 钟表误差修正寄存器 (RC1SUBCUD)

RC1SUBCUD 寄存器用于高精度修正钟表，当钟表快或慢时，通过改变从副计数器(RSUBC)到秒计数器的溢出值（参考值：7FFFH），来修正钟表。

该寄存器可以按字节或按位进行读取或写入。

系统复位后，该寄存器被置为 00H。

- 备注**
1. 只有当实时计数器被置为初始值时，可以重写 RC1SUBU 寄存器。请认真参阅 12.4.1 初始化设置。
 2. 钟表误差修正的详情参见 12.4.9 实时计数器的钟表误差修正示例。

复位后: 00H R/W 地址: FFFFFAD9H								
	7	6	5	4	3	2	1	0
RC1SUBU	DEV	F6	F5	F4	F3	F2	F1	F0
	DEV	钟表误差修正时序的设置						
	0	当 RC1SEC (秒计数器) 为 00、20 或 40 秒时，修正钟表误差						
	1	当 RC1SEC (秒计数器) 为 00 秒时，修正钟表误差						
	F6	钟表误差修正值的设置						
	0	增加 RC1SUBC 计数值，增加的值由 F5 至 F0 位设置（正向修正） 计算增加值的表达式： $\{(F5、F4、F3、F2、F1、F0 \text{ 位的设定值}) - 1\} \times 2$						
	1	减少 RC1SUBC 计数值，减少的值由 F5 至 F0 位设置（负向修正） 计算减少值的表达式： $\{(/F5、/F4、/F3、/F2、/F1、/F0) + 1\} \times 2 \text{ 减小}$						
当 F6 至 F0 位的值是{1/0、0、0、0、0、0、1/0}时，不执行钟表误差修正。								

(14) 分钟闹钟寄存器(RC1ALM)

RC1ALM 寄存器用于设置闹钟分钟。
 该寄存器可以按字节进行读取或写入。
 系统复位后，该寄存器被置为 00H。

注意事项 以 BCD 码格式设置 00 至 59 的十进制数到该寄存器。如果设置的值超出范围，闹钟将不作检测。

复位后: 00H R/W 地址: FFFFFADAH								
	7	6	5	4	3	2	1	0
RC1ALM	0							

(15) 小时闹钟寄存器(RC1ALH)

RC1ALH 寄存器用于设置闹钟小时。
 该寄存器可以按字节进行读取或写入。
 系统复位后，该寄存器被置为 12H。

- 注意事项**
1. 以 BCD 码设置 00 至 23, 01 至 12 或 21 至 32 到该寄存器。如果设置的值超出范围，闹钟将不检测。
 2. 如果选择了 AMPM 位=0 (12 小时格式)，RC1ALH 的第 5 位表示 a.m.(0)或 p.m.(1)。

复位后: 12H R/W 地址: FFFFFADBH								
	7	6	5	4	3	2	1	0
RC1ALH	0	0						

(16) 星期闹钟寄存器(RC1ALW)

RC1ALW 寄存器用于设置闹钟星期。
该寄存器可以按字节进行读取或写入。
系统复位后，该寄存器被置为 00H。

注意事项 关于在实时计数器工作(RC1PWR 位 = 1)时重写 RC1ALW 位，参见 12.4.5 在时钟工作期间改变 INTRTC1 中断设置。

复位后: 00H		R/W	地址: FFFFFADCH							
			7	6	5	4	3	2	1	0
RC1ALW	0	RC1ALW6	RC1ALW5	RC1ALW4	RC1ALW3	RC1ALW2	RC1ALW1	RC1ALW0		
	RC1ALWn	闹钟中断星期位 (n=0至6)								
	0	如果 RC1WEEK = nH, 不产生闹钟中断								
	1	如果 RC1WEEK = nH, 产生闹钟中断。 (通过 RC1ALM 和 RC1ALH 寄存器设置时间)								

(a) 闹钟中断设置示例 (RC1ALM, RC1ALH 和 RC1ALW 设置示例)

表 12-4 和 12-5 给出了设置示例，如果设置星期天 RC1WEEK = 00，设置星期一 RC1WEEK = 01，设置星期天二 RC1WEEK = 02, …，设置星期六 RC1WEEK = 06。

表 12-4. AMPM = 0 的闹钟中断设置示例 (RC1HOUR 寄存器 12 小时显示)

闹钟设置时间	寄存器	RC1ALW	RC1ALH	RC1ALM
星期天, 7:00 a.m.		01H	07H	00H
星期天/星期一, 00:15 p.m.		03H	32H	15H
星期一/星期二/星期五, 5:30 p.m.		26H	25H	30H
每天, 10:45 p.m.		7FH	30H	45H

表 12-5. AMPM = 1 的闹钟中断设置示例 (RC1HOUR 寄存器 24 小时显示)

闹钟设置时间	寄存器	RC1ALW	RC1ALH	RC1ALM
星期天, 7:00		01H	07H	00H
星期天/星期一, 12:15		03H	12H	15H
星期一/星期二/星期五, 17:30		26H	17H	30H
每天, 22:45		7FH	22H	45H

(17) 预分频模式寄存器 0 (PRSM0)

PRSM0 寄存器用于控制实时计数器计数时钟的产生(f_{BRG})。

该寄存器可以按字节或按位进行读取或写入。

系统复位后，该寄存器被置为 00H。

复位后: 00H R/W 地址: FFFF8B0H

符号	7	6	5	4	3	2	1	0
PRSM0	0	0	0	BGCE0	0	0	BGCS01	BGCS00

BGCE0	主时钟操作使能
0	禁止
1	使能

BGCS01	BGCS00	实时计数器时钟源(f_{BGCs})选择		
			5MHz	4MHz
0	0	f_x	200ns	250ns
0	1	$f_x/2$	400ns	500ns
1	0	$f_x/4$	800ns	1 us
1	1	$f_x/8$	1.6 us	2 us

- 注意事项**
1. 在实时计数器工作期间，不要改变BGCS00和BGCS01位的值。
 2. 置BGCE0位为1之前，设置PRSM0寄存器。
 3. 根据获取32.768 kHz频率 f_{BRG} 的主时钟来设置PRSM0和PRSCM0寄存器。

(18) 预分频比较寄存器 0 (PRSCM0)

PRSCM0 寄存器是 8 位比较寄存器。
 该寄存器可以按字节进行读取或写入。
 系统复位后，该寄存器被置为 00H。

复位后: 00H R/W 地址: FFFFF8B1H

	7	6	5	4	3	2	1	0
PRSCM0	PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00

注意事项

1. 在实时计数器工作期间，不要重写 PRSCM0 寄存器。
2. 置 BGCE0 位为 1 之前，设置 PRSCM0 寄存器。
3. 根据获取 32.768 kHz 频率 fBRG 的主时钟来设置 PRSM0 和 PRSCM0 寄存器。

fBRG 的计算如下。

$$f_{BRG} = f_{BGCS}/2N$$

备注

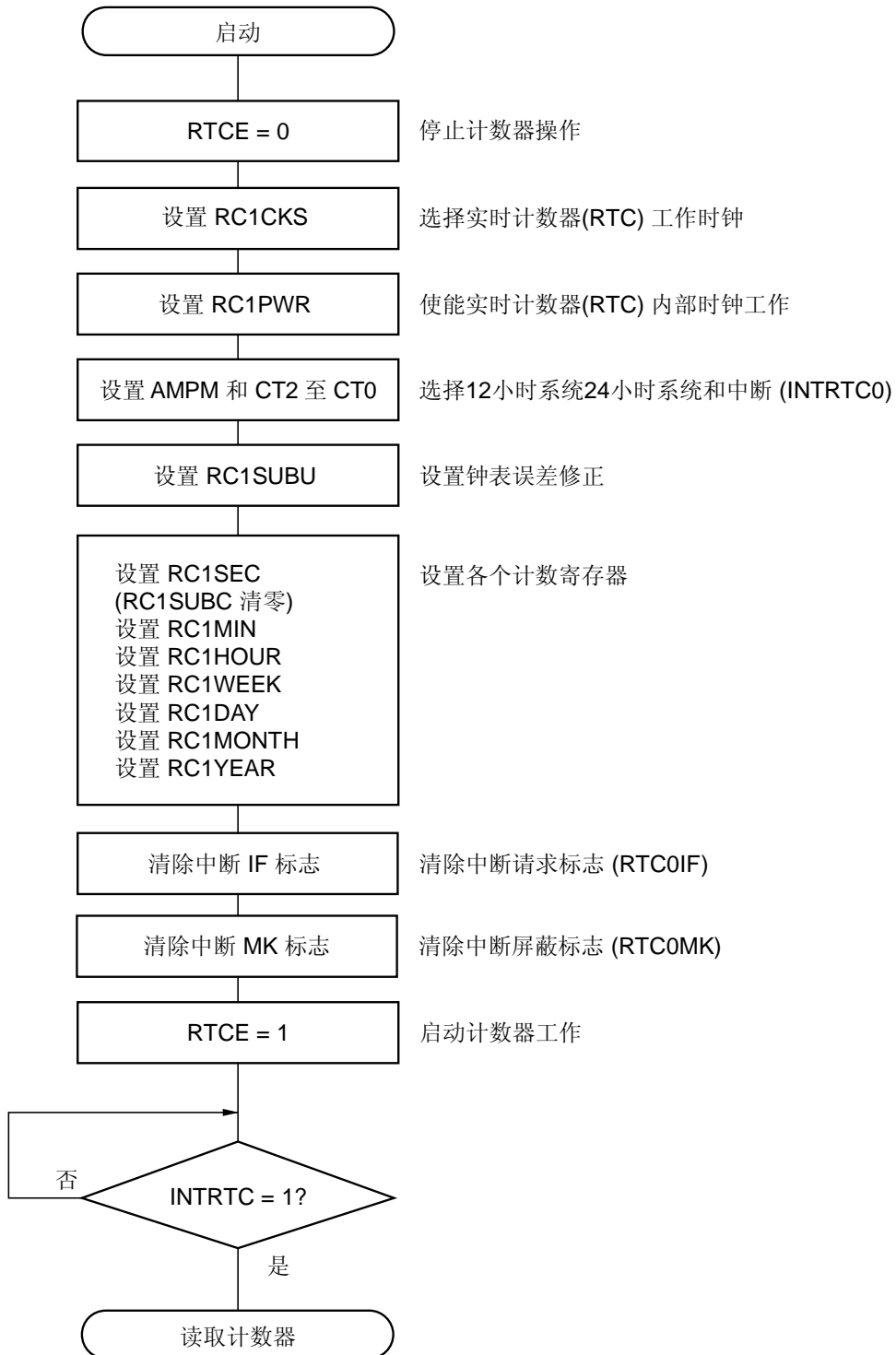
fBGCS: PRSM0 寄存器设置的钟表定时器时钟源
 N: PRSCM0 寄存器的设定值 = 1 至 256
 但是，当 PRSCM0 寄存器被设置为 00H 时，N = 256。

12.4 操作

12.4.1 初始化设置

操作钟表功能并执行固定周期中断操作，所需的初始化。

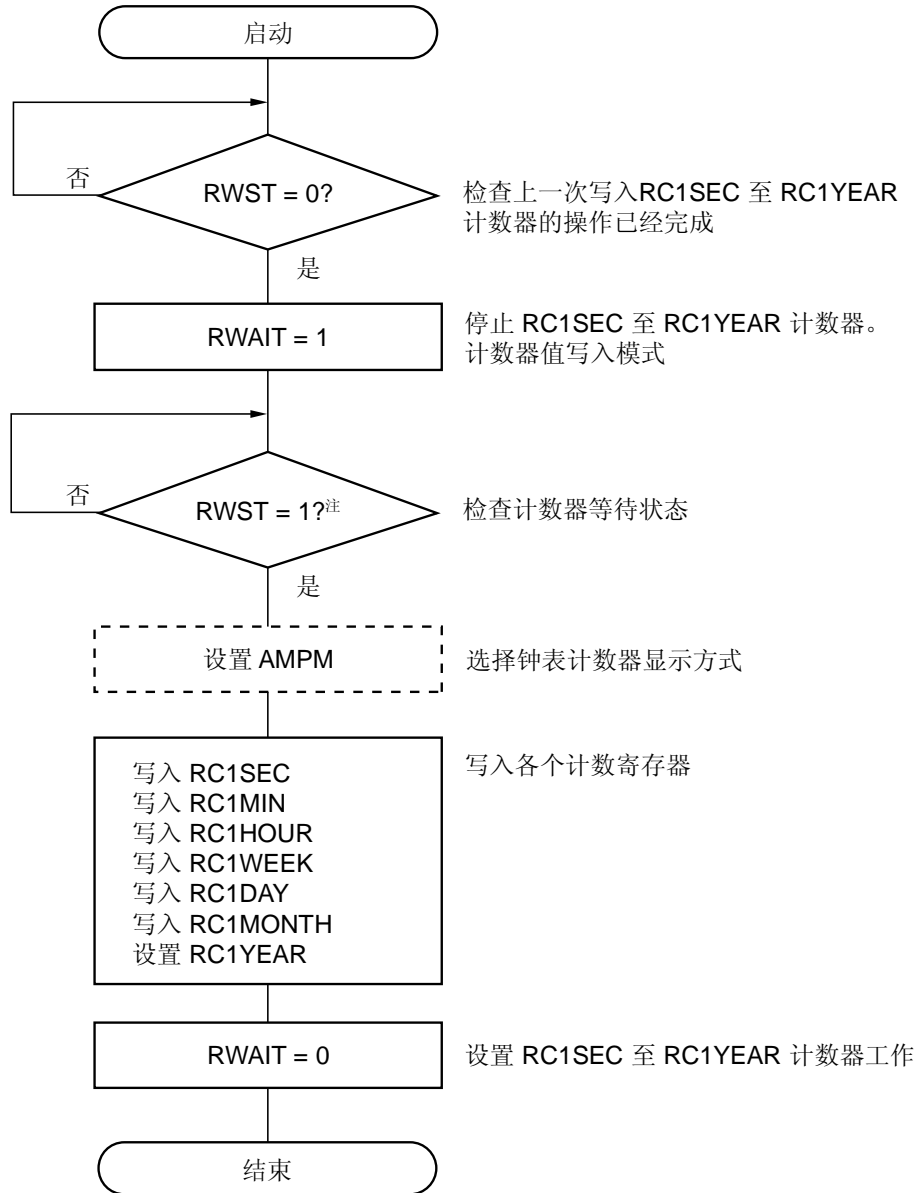
图 12-2. 初始化设置过程



12.4.2 在时钟工作期间重写各个计数器

在时钟工作期间(RC1PWR = 1)重写各个计数器(RC1SEC, RC1MIN, RC1HOUR, RC1WEEK, RC1DAY, RC1MONTH, RC1YEAR)时, 设置如下。

图 12-3. 在时钟工作期间重写各个计数器



注 设置 STOP 模式之前, 请确保设置 RWST = 0。

注意事项 在 1 秒内完成置 RWAIT 为 1 到清 RWAIT 为 0 的系列操作。

如果设置 RWAIT = 1, RC1SEC 至 RC1YEAR 的操作停止。如果在 RWAIT = 1 时, 发生一次从 RC1SUBC 的进位, 可以内部保留一个进位。但是有两个或多个进位发生, 进位的次数无法保留。

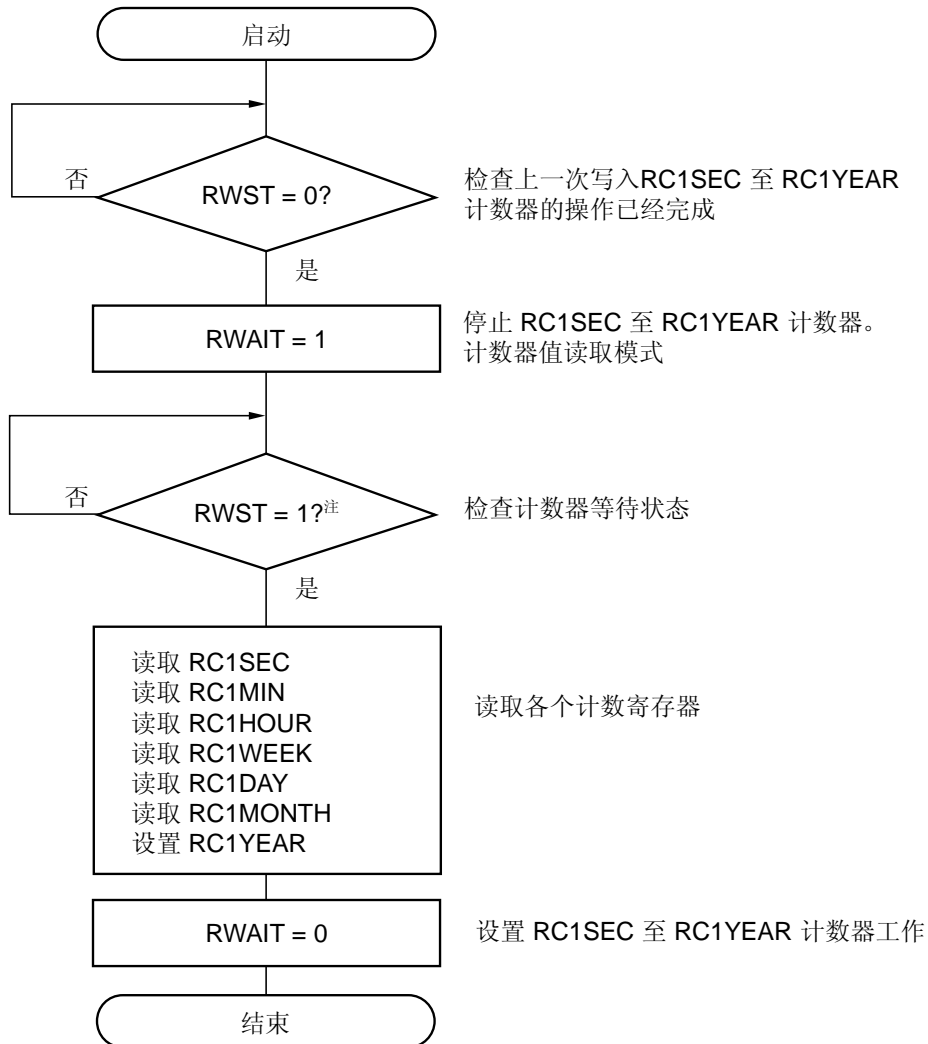
备注 RC1SEC, RC1MIN, RC1HOUR, RC1WEEK, RC1DAY, RC1MONTH, 和 RC1YEAR 可能以任何顺序重写。

不用设置所有寄存器, 仅部分寄存器可被写入。

12.4.3 在时钟工作期间读取各个计数器

在时钟工作期间(RC1PWR = 1) 读取各个计数器(RC1SEC, RC1MIN, RC1HOUR, RC1WEEK, RC1DAY, RC1MONTH, RC1YEAR)时, 设置如下。

图 12-4. 在时钟工作期间读取各个计数器



注 设置 STOP 模式之前, 请确保设置 RWST = 0。

注意事项 在 1 秒内完成置 RWAIT 为 1 到清 RWAIT 为 0 的系列操作。

如果设置 RWAIT = 1, RC1SEC 至 RC1YEAR 的操作停止。如果在 RWAIT = 1 时, 发生一次从 RC1SUBC 的进位, 可以内部保留一个进位。但是有两个或多个进位发生, 进位的次数无法保留。

备注 RC1SEC, RC1MIN, RC1HOUR, RC1WEEK, RC1DAY, RC1MONTH, 和 RC1YEAR 可能以任何顺序重写。

不用设置所有寄存器, 仅部分寄存器可被读取。

12.4.4 在时钟工作期间改变INTRTC0 中断设置

如果在时钟工作期间(RC1PWR = 1)，改变INTRTC0中断(固定周期中断)信号的设置，在输出的INTRTC0中断波形中可能会包括干扰和意外信号。为了屏蔽干扰，在时钟工作期间(RC1PWR = 1)要改变INTRTC0中断信号的设置，设置如下。

图 12-5. 在时钟工作期间改变 INTRTC0 中断设置

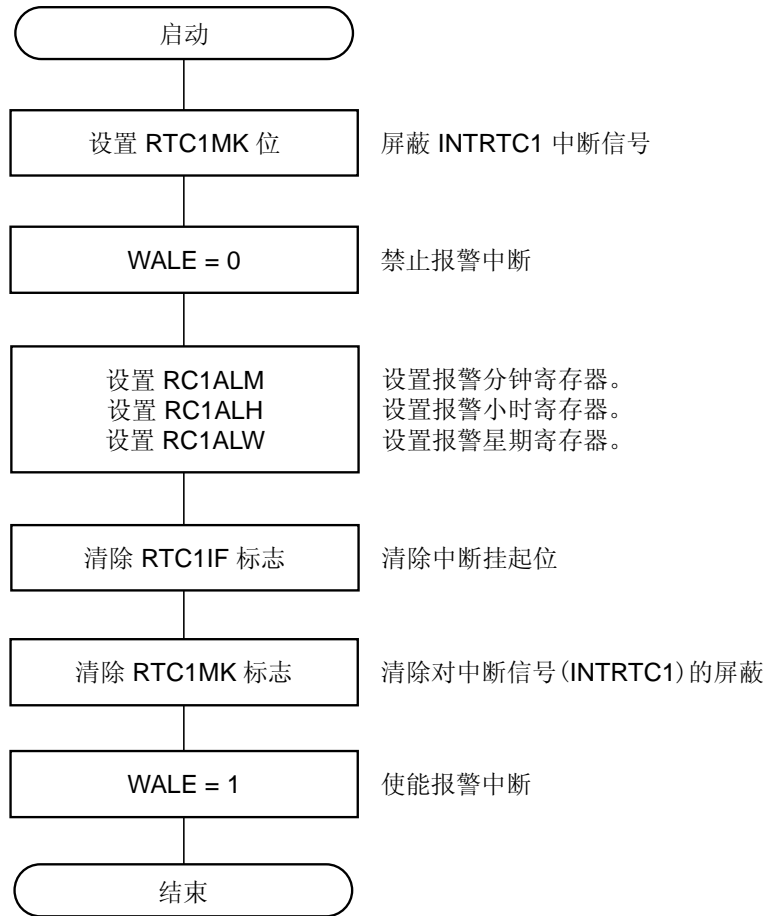


备注 RTC0IF 和 RTC0MK 位的详情参见 23.3.4 中断控制寄存器 (xxICn)。

12.4.5 在时钟工作期间改变INTRTC1 中断设置

如果在时钟工作期间($RC1PWR = 1$)，改变INTRTC1中断(闹钟中断)信号的设置，在输出的INTRTC1中断波形中可能会包括干扰和意外信号。为了屏蔽干扰，在时钟工作期间($RC1PWR = 1$)要改变INTRTC1中断信号的设置，设置如下。

图 12-6. 在时钟工作期间改变 INTRTC1 中断设置

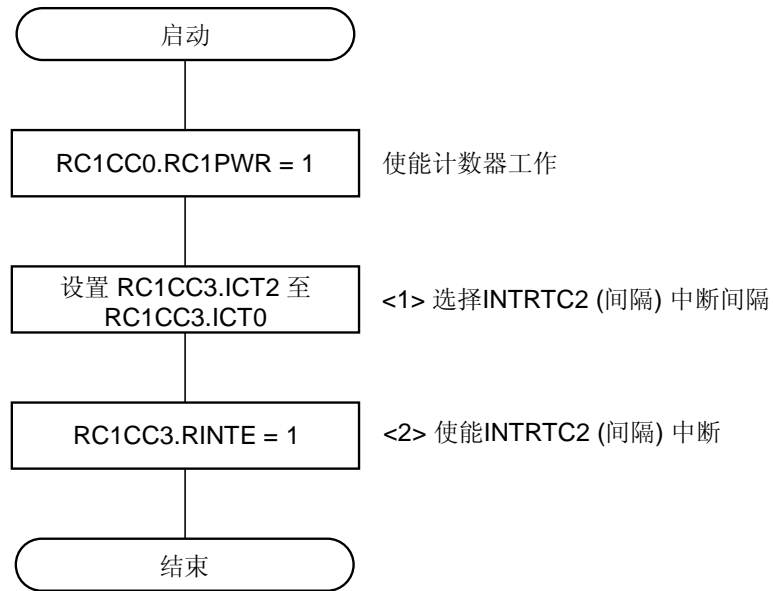


备注 RTC1IF 和 RTC1MK 位的详情参见 23.3.4 中断控制寄存器 (xxICn)。

12.4.6 INTRTC2 中断设置初始化

对 INTRTC2 中断(间隔中断)的设置如下。

图 12-7. INTRTC2 中断设置



注意事项 同时设置<1> 和<2>, 或者先设置<1>。如果先设置<2>, 可能会有意外波形中断发生。

12.4.7 在时钟工作期间改变INTRTC2 中断设置

如果在实时计数器的时钟工作期间($RC1PWR = 1$)，改变INTRTC2中断 (间隔中断)信号的设置，在输出的INTRTC2 中断波形中可能会包括干扰和意外信号。为了屏蔽干扰，在时钟工作期间($RC1PWR = 1$) 要改变INTRTC2中断信号的设置，设置如下。

图 12-8. 在时钟工作期间改变 INTRTC2 中断设置

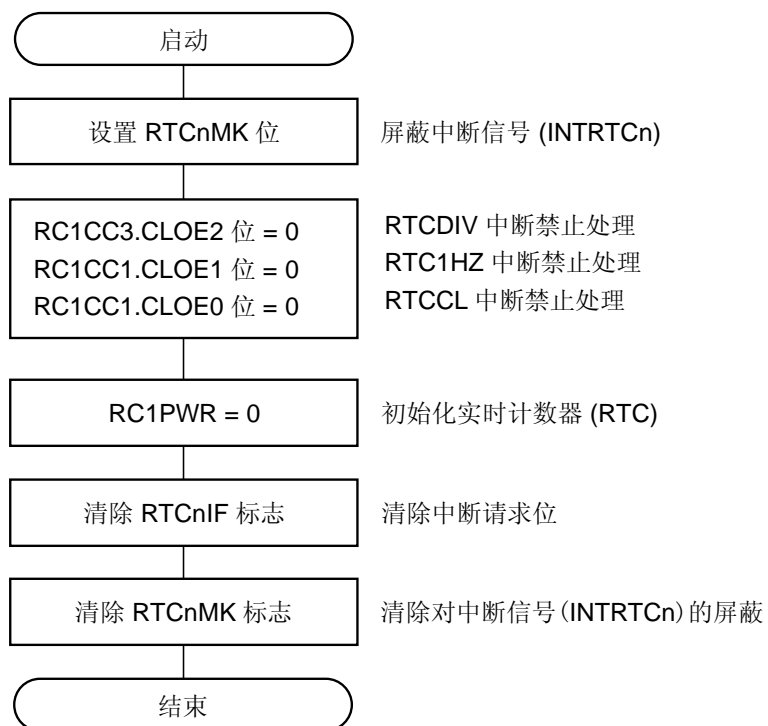


备注 RTC2IF 和 RTC2MK 位的详情参见 23.3.4 中断控制寄存器 (xxICn)。

12.4.8 实时计数器初始化

实时计数器初始化过程如下。

图 12-9. 实时计数器初始化



- 备注
1. RTCnIF 和 RTCnMK 位的详情参见 23.3.4 中断控制寄存器 (xxICn)。
 2. n = 0 至 2

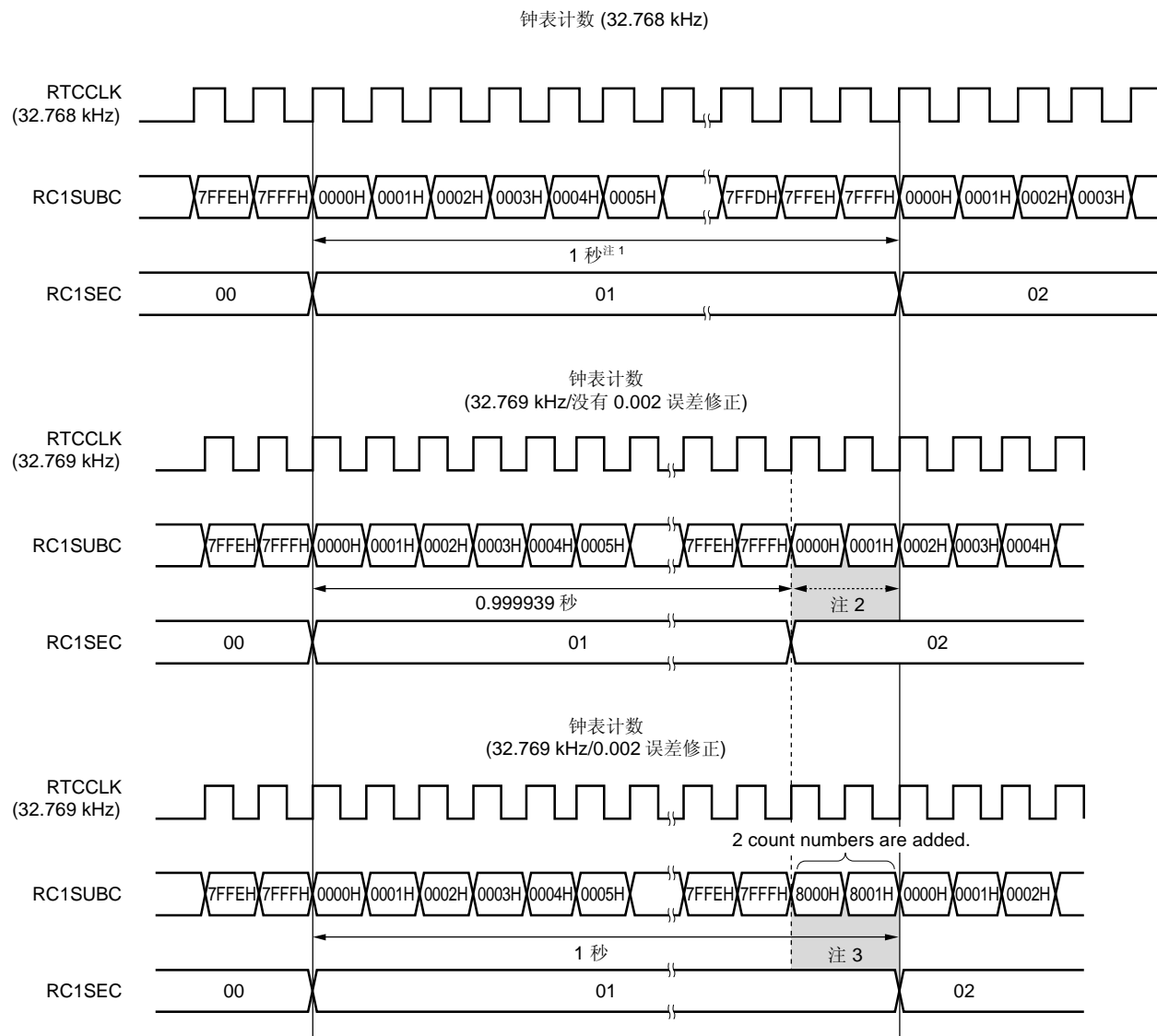
12.4.9 实时计数器的钟表误差修正示例

钟表误差修正功能用于修正连接到 V850ES/Jx3H 微控制器的谐振器的振荡频率偏差。

这里的偏差是指稳定状态偏差，是设计谐振器时频率上的固有偏差。

下面的时序图给出了示例，如果连接到单片机的谐振器设计期望值是 32.768 kHz 但实际为 32.769 kHz，输入时钟会发生错误，并且通过 RC1SUBC 和 RC1SEC 计数操作来修正误差如下所示。

图 12-10. 钟表误差修正示例



- 注
1. 通过对 32.768 kHz 时钟计数 32,768 (0000H 至 7FFFH)次，计数到 1 秒。
 2. 对 32.770 kHz 时钟计数 32,768 (0000H 至 7FFFH)次，得到 $32,768/32,770 = 0.999939$ 秒。每个月产生的误差是 $(32,768/32,770 - 1) \times 60$ (秒) $\times 60$ (分) $\times 24$ (小时) $\times 30$ (天) = -158.2 秒。
 3. 对 32.770 kHz 时钟计数 32,770 (0000H 至 8001H)次，可以误差的计数到 1 秒。

如图 12-10 所示，如果谐振器是正向误差比 32.768 kHz 快，通过增加 RC1SUBC 计数值，钟表可以精确计数。同理，如果谐振器是负向误差比 32.768 kHz 慢，通过减少 RC1SUBC 计数值，钟表也可以精确计数。

RC1SUBC 修正值由 RC1SUBU.F6 至 RC1SUBU.F0 位决定。

F6 位用于决定是增加还是减少 RC1SUBC，F5 至 F0 位决定 RC1SUBC 的值。

(1) 增加 RC1SUBC 计数值

设置 F6 位为 0，RC1SUBC 计数值增加，增加的数值由 F5 至 F0 位设置。

计算增加数值的表达式： $(F5 \text{ 至 } F0 \text{ 位的值} - 1) \times 2$

[增加 RC1SUBC 计数值举例：F6 位 = 0]

如果 F5 至 F0 位设置为 15H (010101B)

$(15H - 1) \times 2 = 40$ (RC1SUBC 计数值增加 40)

RC1SUBC 计数值 = $32,768 + 40 = 32,808$

(2) 减少 RC1SUBC 计数值

设置 F6 位为 1，RC1SUBC 计数值减少，减少的数值由 F5 至 F0 位的设定值取反得到。

计算减少数值的表达式： $(F5 \text{ 至 } F0 \text{ 位设定值取反} + 1) \times 2$

[减少 RC1SUBC 计数值举例：F6 位 = 1]

如果 F5 至 F0 位设置为 15H (010101B)

对 15H (010101B)取反 = 2AH (101010B)

$(2AH + 1) \times 2 = 86$ (RC1SUBC 计数值减少 86)

RC1SUBC 计数值 = $32,768 - 86 = 32,682$

(3) DEV 位

DEV 位决定 F6 至 F0 位的设置何时使能。

F6 至 F0 位的设置值在下一个时序体现，但并不是每次都影响 RC1SUBC 计数值。

表 12-6. DVE 位设置

DEV 位的设置	设定值反映到RC1SUBC的时序
0	当RC1SEC 是00, 20, 或 40秒时
1	当RC1SEC 是 00秒时

[F6 至 F0 位的设置值是 0010101B 时的举例]

- 如果 DEV 位是 0
 - 在 00, 20, 或 40 秒时, RC1SUBC 计数值是 32,808。
 - 其它时间是 32,768。
- 如果 DEV 位是 1
 - 在 00, 20, 或 40 秒时, RC1SUBC 计数值是 32,808。
 - 其它时间是 32,768。

如上所述，为了匹配 RC1SUBC 计数值和谐振器的偏差程序，每 20 秒或 60 秒 RC1SUBC 计数值被修正，而不是每秒都进行修正。

谐振器频率可以被实际修正的范围如下。

- 如果 DEV 位是 0: 32.76180000 kHz 至 32.77420000 kHz
- 如果 DEV 位是 1: 32.76593333 kHz 至 32.77006667 kHz

DEV 位为 0 时可以修正的频率范围是 DEV 位为 1 时的三倍。

但是，DEV 位为 1 时设定频率的准确度是 DEV 位为 0 时的三倍。

表 12-7 和 12-8 展示了 DEV 和 F6 至 F0 位的设定值，以及可以修正的对应频率。

表 12-7. DEV 位为 0 时可以修正的频率范围

F6	F5 至 F0	RC1SUBC修正值	连接的时钟频率 (包括稳定状态偏差)
0	000000	无修正	-
0	000001	无修正	-
0	000010	每20秒RC1SUBC计数值增加2	32.76810000 kHz
0	000011	每20秒RC1SUBC计数值增加4	32.76820000 kHz
0	000100	每20秒RC1SUBC计数值增加6	32.76830000 kHz
⋮			
0	111011	每20秒RC1SUBC计数值增加120	32.77400000 kHz
0	111110	每20秒RC1SUBC计数值增加122	32.77410000 kHz
0	111111	每20秒RC1SUBC计数值增加124	32.77420000 kHz (上限)
1	000000	无修正	-
1	000001	无修正	-
1	000010	每20秒RC1SUBC计数值减少124	32.76180000 kHz (下限)
1	000011	每20秒RC1SUBC计数值减少122	32.76190000 kHz
1	000100	每20秒RC1SUBC计数值减少120	32.76200000 kHz
⋮			
1	11011	每20秒RC1SUBC计数值减少6	32.76770000 kHz
1	11110	每20秒RC1SUBC计数值减少4	32.76780000 kHz
1	11111	每20秒RC1SUBC计数值减少2	32.76790000 kHz

表 12-7. DEV 位为 1 时可以修正的频率范围

F6	F5 至 F0	RC1SUBC修正值	连接的时钟频率 (包括稳定状态偏差)
0	000000	无修正	-
0	000001	无修正	-
0	000010	每60秒RC1SUBC计数值增加2	32.76803333 kHz
0	000011	每60秒RC1SUBC计数值增加4	32.76806667 kHz
0	000100	每60秒RC1SUBC计数值增加6	32.76810000 kHz
⋮			
0	111011	每60秒RC1SUBC计数值增加120	32.77000000 kHz
0	111110	每60秒RC1SUBC计数值增加122	32.77003333 kHz
0	111111	每60秒RC1SUBC计数值增加124	32.77006667 kHz (上限)
1	000000	无修正	-
1	000001	无修正	-
1	000010	每60秒RC1SUBC计数值减少124	32.76593333 kHz (下限)
1	000011	每60秒RC1SUBC计数值减少122	32.76596667 kHz
1	000100	每60秒RC1SUBC计数值减少120	32.76600000 kHz
⋮			
1	11011	每60秒RC1SUBC计数值减少6	32.76790000 kHz
1	11110	每60秒RC1SUBC计数值减少4	32.76793333 kHz
1	11111	每60秒RC1SUBC计数值减少2	32.76796667 kHz

第十三章 看门狗定时器 2 的功能

13.1 功能

看门狗定时器 2 具有以下功能：

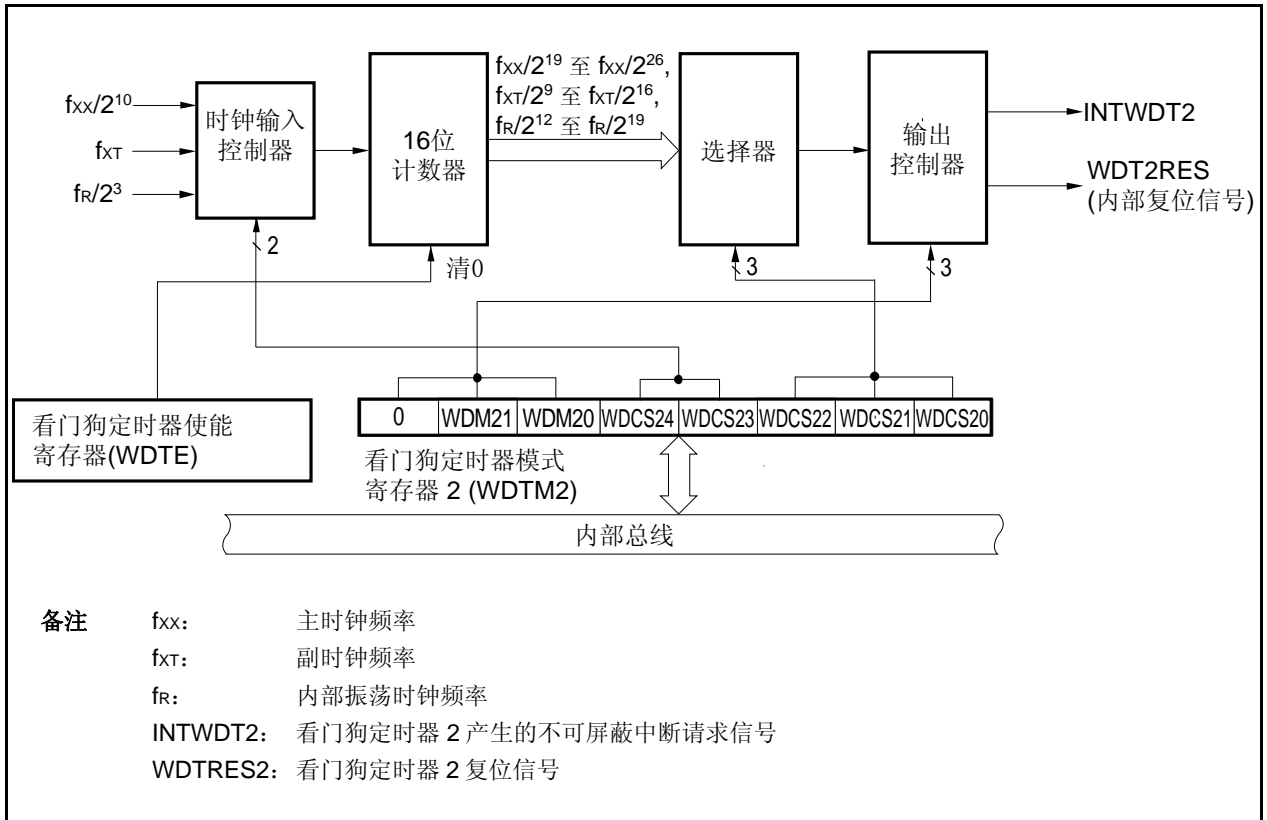
- 默认启动的看门狗定时器^{注1}
 - 复位模式：看门狗定时器 2 溢出后执行复位操作（产生 WDT2RES 信号）
 - 不可屏蔽中断请求模式：看门狗定时器 2 溢出后执行 NMI 操作（产生 INTWDT2 信号）^{注2}
- 时钟源可以选择主时钟、内部振荡时钟或副时钟

- 注**
1. 在复位模式下，复位释放后看门狗定时器 2 就会自动启动。
当不使用看门狗定时器 2 时，或者在系统复位执行前使用该功能将其停止，或者在下一个间隔时间内对看门狗定时器 2 执行一次清除然后再停止其工作。
即使不需要改变默认设置（复位模式，间隔时间： $f_R/2^{19}$ ），为了确保操作的正确性，也要对 WDTM2 寄存器执行一次写入操作。
 2. 关于不可屏蔽中断请求信号（INTWDT2）的不可屏蔽中断服务程序，参见 23.2.2（2）自 INTWDT2 的信号。

13.2 配置

看门狗定时器 2 的功能框图如下所示：

图 13-1. 看门狗定时器 2 的功能框图



看门狗定时器 2 包括下列硬件：

表 13-1. 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)

13.3 寄存器

(1) 看门狗定时器模式寄存器 2 (WDTM2)

WDTM2 寄存器用于设置看门狗定时器 2 的溢出时间和工作时钟。

该寄存器可按字节读取或写入。该寄存器读取次数不限，但是仅能在复位解除后写入一次。

复位输入将该寄存器设置为 67H。

注意事项 下列状态下禁止访问 WDTM2 寄存器。详细说明，参见 3.4.8 (2) 访问片上外设 I/O 专用寄存器。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
- 当 CPU 运行于内部振荡时钟工作时。

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	停止工作
0	1	不可屏蔽中断请求模式 ^L (产生INTWDT2 信号)
1	-	复位模式 (产生WDT2RES 信号)

- 注意事项**
1. 关于 WDCS20 至 WDCS24 位的细节，参见表 13-2 看门狗定时器 2 的时钟选择。
 2. 虽然停止内部振荡器的运行就可以停止看门狗定时器 2 的操作，还是要对 WDTM2 寄存器清零，以确保停止该定时器（为了避免由于误写入操作而选择主时钟或副时钟）。
 3. 复位后，如果对 WDTM2 寄存器执行了两次写入操作，那么将强制产生溢出信号且计数器复位。
 4. 为了有意产生溢出信号，仅需向 WDTM2 寄存器写入两次数据，或者向 WDTM2 寄存器写入一次“ACH”之外的数值。
但是，当看门狗定时器 2 被设置为停止运行时，无论向 WDTM2 寄存器写入两次数据，还是向 WDTM2 寄存器写入一次“ACH”之外的数值，都不会产生溢出信号。
 5. 为了停止看门狗定时器 2，将 RCM.RSTOP 位置为 1（停止内部振荡器），并且将 00H 写入 WDTM2 寄存器。如果 RCM.RSTOP 位不能置为 1，就将 WDCS23 位置为 1（ $2^1/f_{xx}$ 被选择为时钟，且该时钟可以在 IDLE1、IDLW2、sub-IDLE 和副时钟操作模式下被停止）。

表 13-2. 看门狗定时器 2 时钟选择

WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	选择时钟	100 kHz (MIN.)	220 kHz (TYP.)	400 kHz (MAX.)
0	0	0	0	0	$2^{12}/f_R$	41.0 ms	18.6 ms	10.2 ms
0	0	0	0	1	$2^{13}/f_R$	81.9 ms	37.2 ms	20.5 ms
0	0	0	1	0	$2^{14}/f_R$	163.8 ms	74.5 ms	41.0 ms
0	0	0	1	1	$2^{15}/f_R$	327.7 ms	148.9 ms	81.9 ms
0	0	1	0	0	$2^{16}/f_R$	655.4 ms	297.9 ms	163.8 ms
0	0	1	0	1	$2^{17}/f_R$	1,310.7 ms	595.8 ms	327.7 ms
0	0	1	1	0	$2^{18}/f_R$	2,621.4 ms	1,191.6 ms	655.4 ms
0	0	1	1	1	$2^{19}/f_R$ (默认值)	5,242.9 ms	2,383.1 ms	1,310.7 ms
						$f_{XX} = 24 \text{ MHz}$	$f_{XX} = 32 \text{ MHz}$	$f_{XX} = 48 \text{ MHz}$
0	1	0	0	0	$2^{19}/f_{XX}$	21.8 ms	16.4 ms	10.9 ms
0	1	0	0	1	$2^{20}/f_{XX}$	43.7 ms	32.8 ms	21.8 ms
0	1	0	1	0	$2^{21}/f_{XX}$	87.4 ms	65.5 ms	43.7 ms
0	1	0	1	1	$2^{22}/f_{XX}$	174.8 ms	131.1 ms	87.4 ms
0	1	1	0	0	$2^{23}/f_{XX}$	349.5 ms	262.1 ms	174.8 ms
0	1	1	0	1	$2^{24}/f_{XX}$	699.1 ms	524.3 ms	349.5 ms
0	1	1	1	0	$2^{25}/f_{XX}$	1398.1 ms	1048.6 ms	699.1 ms
0	1	1	1	1	$2^{26}/f_{XX}$	2796.2 ms	2097.2 ms	1398.1 ms
						$f_{XT} = 32.768 \text{ kHz}$		
1	×	0	0	0	$2^9/f_{XT}$	15.625 ms		
1	×	0	0	1	$2^{10}/f_{XT}$	31.25 ms		
1	×	0	1	0	$2^{11}/f_{XT}$	62.5 ms		
1	×	0	1	1	$2^{12}/f_{XT}$	125 ms		
1	×	1	0	0	$2^{13}/f_{XT}$	250 ms		
1	×	1	0	1	$2^{14}/f_{XT}$	500 ms		
1	×	1	1	0	$2^{15}/f_{XT}$	1,000 ms		
1	×	1	1	1	$2^{16}/f_{XT}$	2,000 ms		

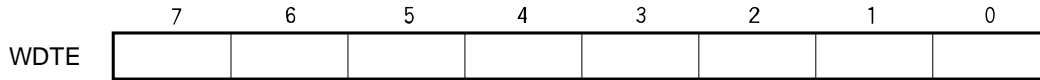
(2) 看门狗定时器使能寄存器 (WDTE)

将“ACH”写入 WDTE 寄存器，可以清除看门狗定时器 2 的计数值，并且重新开始计数。

WDTE 寄存器可以按字节读取或写入。

复位输入将该寄存器设为 9AH。

复位后: 9AH R/W 地址: FFFFF6D1H



- 注意事项**
1. 当“ACH”之外的数值被写入 WDTE 寄存器时，强制输出一个溢出信号。
 2. 对 WDTE 寄存器执行 1 位存储处理指令，强制输出一个溢出信号。
 3. 为了有意产生溢出信号，向 WDTE 寄存器写入一次“ACH”之外的数据，或者向 WDTM2 寄存器写入两次数据即可。
但是，当看门狗定时器 2 被设置为停止运行时，无论向 WDTM2 寄存器写入两次数据，还是向 WDTE 寄存器写入一次“ACH”之外的数值，都不会产生溢出信号。
 4. 寄存器的读出值总是“9AH”（与写入值“ACH”不同）。

13.4 操作

在复位模式下，当复位解除后，看门狗定时器 2 就会自动启动。

复位后，仅能使用字节访问对 WDTM2 寄存器写入一次。要使用看门狗定时器 2，应使用字节存储处理指令将工作模式以及间隔时间写入 WDTM2 寄存器。之后，将无法停止看门狗定时器的操作。

WDTM2.WDCS24 位至 WDTM2.WDCS20 位用来选择看门狗定时器 2 循环检测间隔时间。

向 WDTE 寄存器写入 ACH 可以清除看门狗定时器 2 的计数值，并且再次开始计数。计数操作开始后，在循环检测时间间隔内将 ACH 写入 WDTE。

如果直到时间间隔时间期满，都没有向 WDTE 寄存器写入 ACH，那么根据 WDTM2.WDM21 和 WDTM2.WDM20 位的设定值，将产生一个复位信号（WDT2RES）或不可屏蔽中断请求信号（INTWDT2）。

当 WDTM2.WDM21 位置 1（复位模式）时，如果在复位或待机状态解除后的振荡稳定期间发生 WDT 溢出，那么将不发生内部复位，CPU 时钟将切换到内部振荡时钟。

若不使用看门狗定时器 2，则向 WDTM2 寄存器写入 00H。

关于设置为不可屏蔽中断请求模式时的不可屏蔽中断服务，参见 23.2.2 (2) 自 INTWDT2 的信号。

第十四章 实时输出功能(RTO)

14.1 功能

实时输出功能将预先设置的数据传送至 RTBL0 和 RTBH0 寄存器中，然后在产生定时器中断的同时，通过硬件输出锁存将数据传送到外部设备中。数据传输到外部设备需要使用的引脚构成一个端口，被称为实时输出功能(RTO)端口。

因为通过 RTO 可以输出无抖动信号，所以适用于控制步进电机。

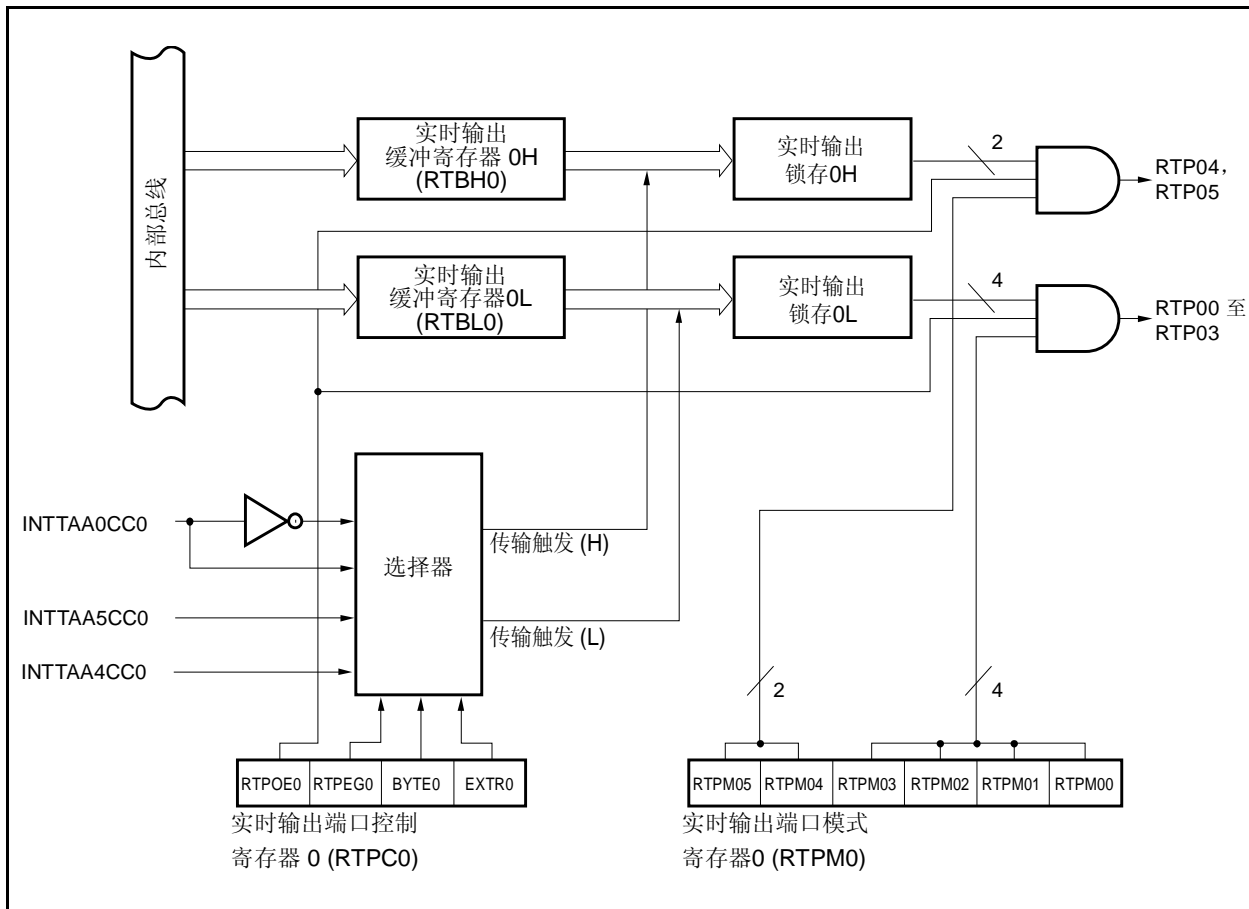
在 V850ES/JG3-U 和 V850ES/JH3-U 单片机中提供一个 6 位的实时输出端口通道。

实时输出端口可以设置为端口模式或按位设置为实时输出端口模式。

14.2 配置

RTO 的功能框图如下所示：

图 14-1. RTO 的功能框图



RTO 包括下列硬件：

表 14-1. RTO 的配置

项目	配置
寄存器	实时输出缓冲寄存器 0L, 0H (RTBL0, RTBH0)
控制寄存器	实时输出端口模式寄存器 0 (RTPM0) 实时输出端口控制寄存器 0 (RTPC0)

(1) 实时输出缓冲寄存器 0L, 0H (RTBL0, RTBH0)

RTBL0 和 RTBH0 寄存器是 4 位寄存器，用于保存预先设置的输出数据。

这些寄存器每一个都被映射到外设 I/O 寄存器区域中的独立地址。

这些寄存器可以按字节或按位进行读写。

复位输入会将这些寄存器设为 00H。

如果指定 4 位×1 通道 或 2 位×1 通道的操作模式(RTPC0.BYTE0 位= 0)，数据可以被分别设置到 RTBL0 和 RTBH0 寄存器。通过指定这些寄存器各自的地址，可以立即读出寄存器的数据。

如果指定 6 位×1 通道的操作模式(BYTE0 位= 1)，向其中任一个寄存器写入数据，都可以把 8 位数据设置到 RTBL0 和 RTBH0 寄存器。此外，通过指定这些寄存器各自的地址，可以立即读出寄存器的数据。

RTBL0 和 RTBH0 寄存器的操作处理过程如表 14-2 所示：

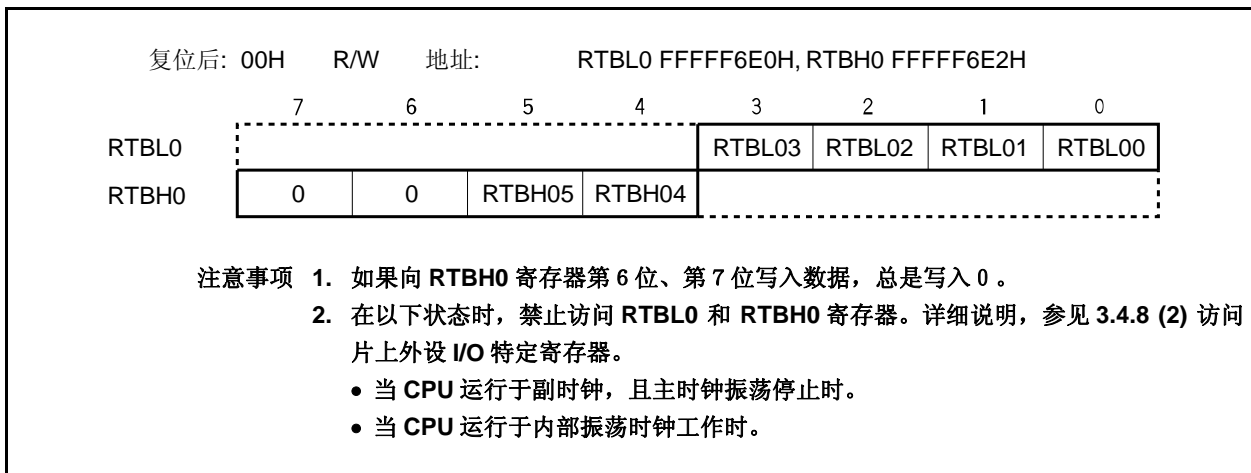


表 14-2. 使用 RTBL0 和 RTBH0 寄存器的处理操作

工作模式	处理的寄存器	读取		写 ^注	
		高 4 位	低 4 位	高 4 位	低 4 位
4 位×1 通道, 2 位×1 通道	RTBL0	RTBH0	RTBL0	无效	RTBL0
	RTBH0	RTBH0	RTBL0	RTBH0	无效
6 位×1 通道	RTBL0	RTBH0	RTBL0	RTBH0	RTBL0
	RTBH0	RTBH0	RTBL0	RTBH0	RTBL0

注 设置为实时输出端口后，在实时输出触发信号产生时，把输出数据设置到 RTBL0 和 RTBH0 寄存器中。

14.3 寄存器

RTO 由以下两个寄存器控制:

- 实时输出端口模式寄存器 0(RTPM0)
- 实时输出端口控制寄存器 0(RTPC0)

(1) 实时输出端口模式寄存器 0(RTPM0)

RTPM0 寄存器用于按位选择实时输出端口模式或端口模式。

该寄存器可以进行字节读写或按位读写。

系统复位后, 该寄存器被设为 00H。

复位后: 00H R/W 地址: FFFFF6E4H

	7	6	5	4	3	2	1	0
RTPM0	0	0	RTPM05	RTPM04	RTPM03	RTPM02	RTPM01	RTPM00

RTPM0m	实时输出端口的控制 (m = 0 至 5)
0	禁止实时输出
1	允许实时输出

- 注意事项**
1. 通过使能实时输出操作 (RTPC0.RTPOE0 位 = 1), RTP00 至 RTP05 引脚中被设置为实时输出的位进行实时输出, 而设置为端口模式的位输出 0。
 2. 如果禁止实时输出操作(RTPOE0 位= 0), 不管 RTPM0 寄存器如何设置, 实时输出引脚(RTP00 至 RTP05) 都输出 0。
 3. 为了将该寄存器用作实时输出引脚(RTP00 至 RTP05), 需用通过 PMC 和 PFC 寄存器将这些引脚设置为实时输出端口引脚。

(2) 实时输出端口控制寄存器 0(RTPC0)

RTPC0 寄存器用于设置操作模式和实时输出端口的输出触发信号。

操作模式和实时输出端口输出触发信号的关系如表 14-3 所示。

该寄存器可以进行字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H R/W 地址: FFFFF6E5H

	<7>	6	5	4	3	2	1	0
RTPC0	RTPOE0	RTPEG0	BYTE0	EXTR0	0	0	0	0

RTPOE0	实时输出操作的控制
0	禁止操作 ^{注 1}
1	允许操作

RTPEG0	INTTP0CC0 信号的有效沿
0	下降沿 ^{注 2}
1	上升沿

BYTE0	实时输出通道配置的说明
0	4位 X 2通道, 2位 X 2通道
1	6位 X 2通道

注 1. 禁止实时输出操作 (RTPOE0 位 = 0) 时, 实时输出信号 (RTP00 至 RTP05) 引脚都输出“0”。

 2. INTTAA0CC0 信号持续输出一个时钟长度, 该时钟是由 TAA0 选择的计数时钟。

注意事项 只有当 RTPOE0 位 = 0 时, 才能设置 RTPEG0、BYTE0 和 EXTR0 位。

表 14-3. 实时输出端口的操作模式和输出触发

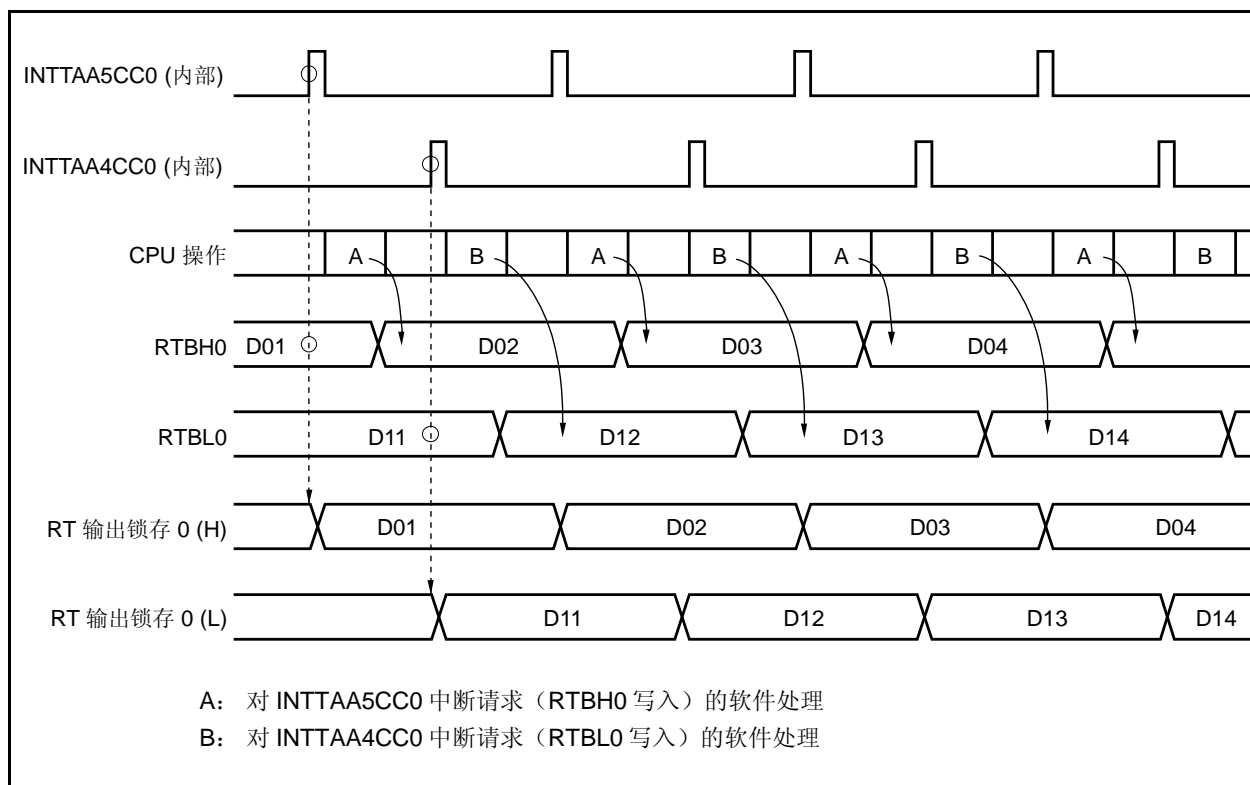
BYTE0	EXTR0	工作模式	RTBH0 (RTP04, RTP05)	RTBL0 (RTP00 至 RTP03)
0	0	4 位×1 通道,	INTTAA5CC0	INTTAA4CC0
	1	2 位×1 通道	INTTAA4CC0	INTTAA0CC0
1	0	6 位×1 通道	INTTAA4CC0	
	1		INTTAA0CC0	

14.4 操作

通过设置 RTPC0.RTPOE0 位为 1，使能实时输出操作，在选定的传送触发信号产生(由 RTPC0.EXTR0 和 RTPC0.BYTE0 位设置)时，RTBH0 和 RTBL0 寄存器的数据被传送到实时输出锁存器中。在已传送的数据中，只有在 RTPM0 寄存器中被设置为允许实时输出的位，其数据才会从 RTP00 至 RTP05 位输出。RTPM0 寄存器中设置为禁止实时输出的位输出 0。

通过将 RTPOE0 位清为 0，禁止实时输出操作，则不管 RTPM0 寄存器如何设置，RTP00 至 RTP05 信号都输出 0。

图 14-2. RTO0 的操作时序示例 (当 EXTR0= 0，BYTE0 位= 0)



备注 关于待机期间的操作，参见第二十五章 待机功能。

14.5 用法

- (1) 禁止实时输出
将 RTPC0.RTPOE0 位清为 0。
- (2) 执行下列初始化操作。
 - 设置端口 2 或端口 5 的复用功能引脚
将 PFC2.PFC2m 和 PFCE2.PFCE2m 位设置为 RTO 引脚之后，将 PMC2.PMC2m 位置为 1 (m = 0 至 3)。
将 PFC5.PFC5m 和 PFCE5.PFCE5m 位设置为 RTO 引脚之后，将 PMC5.PMC5m 位置为 1 (m = 0 至 5)。
 - 按位指定实时输出端口模式或端口模式。
设置 RTPM0 寄存器。
 - 通道配置：选择触发信号和有效沿。
设置 RTPC0.EXTR0、RTPC0.BYTE0 和 RTPC0.RTPEG0 位。
 - 设置 RTBH0 和 RTBL0 寄存器^{注1}的初始值。
- (3) 实时输出使能
设置 RTPOE0 位 = 1。
- (4) 在选择的传送触发信号产生^{注2}时，把下一个输出值设置到 RTBH0 和 RTBL0 寄存器。
- (5) 通过所选触发信号的中断服务程序，按顺序把下一个实时输出值设置到 RTBH0 和 RTBL0 寄存器。

- 注
1. 当 RTPOE0 位 = 0 时，如果写入 RTBH0 和 RTBL0 寄存器，则该值被传送到实时输出锁存 OH 和 OL。
 2. 当 RTPOE0 位 = 1 时，即使写入 RTBH0 和 RTBL0 寄存器，则数据不被传送到实时输出锁存 OH 和 OL。

14.6 注意事项

- (1) 通过软件避免以下的冲突：
 - 实时输出禁止/使能切换(RTPOE0 位)和选择的实时输出输出触发信号之间的冲突。
 - 在实时输出状态下对 RTBH0 和 RTBL0 寄存器的写入操作和选择的实时输出触发信号之间的冲突。
- (2) 在执行初始化之前，禁止实时输出(RTPOE0 位 = 0)。
- (3) 一旦实时输出被禁止 (RTPOE0 位 = 0)，确保重新使能实时输出 (RTPOE0 位 = 0 → 1) 之前，对 RTBH0 和 RTBL0 寄存器初始化。

第十五章 A/D转换器

15.1 概述

A/D 转换器可以将模拟输入信号转换为数字值，具有 10 位分辨率，具有 12 路模拟输入信号通道（ANI0 至 ANI11）。

A/D 转换器具有下列特征：

- 10 位分辨率。
- 12 通道
- 逐次逼近法
- 工作电压： $AV_{REF0} = 3.0$ 至 3.6 V 。
- 模拟输入电压： 0 V 至 AV_{REF0}
- 操作模式具有下列功能：
 - 连续选择模式
 - 连续扫描模式
 - 单次选择模式
 - 单次扫描模式
- 触发模式具有下列功能：
 - 软件触发模式
 - 外部触发模式（外部，1）
 - 定时器触发模式
- 掉电监控功能（转换结果比较功能）

15.2 功能

（1）10 位分辨率的 A/D 转换

从 ANI0 至 ANI11 通道中选择模拟输入通道，且 A/D 转换器以 10 位分辨率进行反复操作。每次 A/D 转换完成后，都会产生一个中断请求信号（INTAD）。

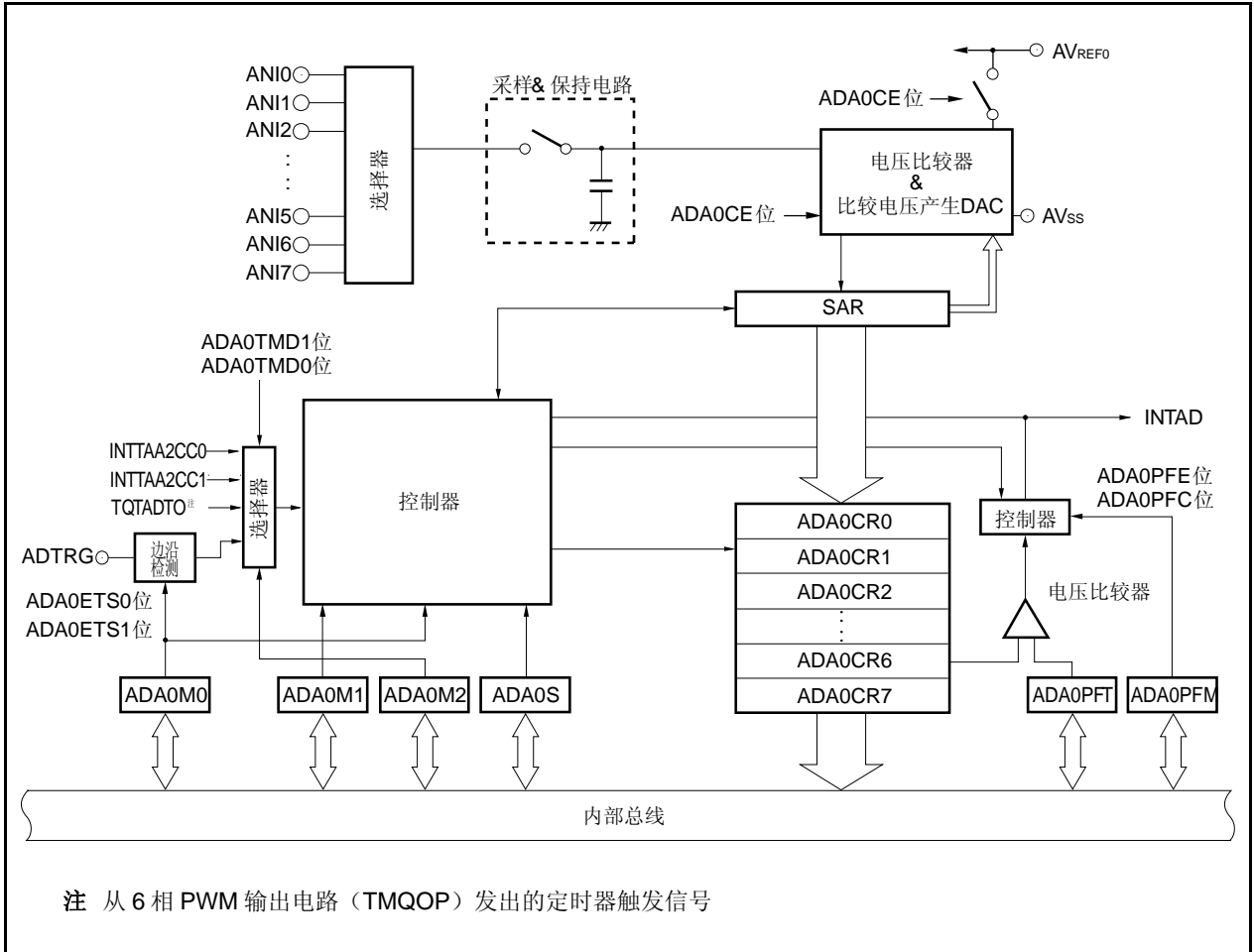
（2）掉电检测功能

此功能用来检测电池电压的下降。将 A/D 转换结果（ADA0CRnH 寄存器的值）和 ADA0PFT 寄存器值进行比较，仅当满足指定的比较条件（ $n = 0$ 至 11 ）时，才会产生 INTAD 信号。

15.3 配置

A/D 转换器的框图如下所示：

图 15-1. A/D 转换器框图



A/D 转换器包括下列硬件：

表 15-1. A/D 转换器的配置

项目	配置
模拟输入	12 个通道 (ANI0 引脚至 ANI11 引脚)
寄存器	逐次逼近寄存器 (SAR) A/D 转换结果寄存器 0 至 11 (ADA0CR0 至 ADA0CR11) A/D 转换结果寄存器 0H 至 11H (ADCR0H 至 ADCR11H)：仅能读取高 8 位
控制寄存器	A/D 转换器模式寄存器 0 至 2 (ADA0M0 至 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) 渐次逼近寄存器 (SAR)

SAR 寄存器将模拟输入信号的电压值与比较电压产生 DAC (比较电压) 的输出值进行比较, 并且从最高有效位 (MSB) 开始保存比较结果。

当比较结果已经保存至最低有效位 (LSB) (也就是 A/D 转换完成) 时, SAR 寄存器的内容将被传送到 ADA0CRn 寄存器。

备注 n = 0 至 11

(2) A/D 转换结果寄存器 n (ADA0CRn), A/D 转换结果寄存器 nH (ADA0CRnH)

ADA0CRn 寄存器是一个存储 A/D 转换结果的 16 位寄存器。ADA0ARn 由 12 个寄存器组成, 并且 A/D 转换结果存储在对应于模拟输入的 AD0CRn 寄存器高 10 位。(低 6 位固定为 0。)

(3) A/D 转换器模式寄存器 0 (ADA0M0)

该寄存器用于指定操作模式并且控制 A/D 转换器的转换操作。

(4) A/D 转换器模式寄存器 1 (ADA0M1)

该寄存器用于设置待转换的模拟输入信号的转换时间。

(5) A/D 转换器模式寄存器 2 (ADA0M2)

该寄存器用于设置硬件触发模式。

(6) A/D 转换器通道指定寄存器 (ADA0S)

该寄存器用于设置输入端口, 该端口用于输入待转换的模拟电压。

(7) 掉电比较模式寄存器 (ADA0PFM)

该寄存器用于设置掉电监控模式。

(8) 掉电比较阈值寄存器 (ADA0PFT)

ADA0PFT 寄存器用于设置阈值, 和 A/D 转换结果寄存器 nH (ADA0CRnH) 的值进行比较。ADA0PFT 寄存器的 8 位数据与 A/D 转换结果寄存器 (ADA0CRnH) 的高 8 位进行比较。

(9) 控制器

当 A/D 转换完成时或使用掉电检测功能时, A/D 控制器将转换结果 (ADA0CRnH 寄存器的值) 与 ADA0PFT 寄存器的值进行比较, 当满足指定的比较条件时, 产生 INTAD 信号。

(10) 采样&保持电路

采样保持电路对输入电路选择的各个模拟输入信号进行采集, 并将采样数据发送到电压比较器。在 A/D 转换期间, 该电路保持采样得到的模拟输入信号电压。

(11) 电压比较器

电压比较器将采样保持电路得到的电压值与比较电压产生 DAC 的输出电压进行比较。

(12) 比较电压产生 DAC

比较电压产生 DAC 连接在 AV_{REF0} 和 AV_{SS} 之间，并产生用于和模拟输入信号进行比较的电压。

(13) ANI0 至 ANI11 引脚

这些引脚是 12 路 A/D 转换通道的模拟输入引脚，用于输入将要转换为数字信号的模拟信号。除了 ADA0S 寄存器选定的模拟输入引脚外，其他引脚都可以用作输入端口引脚。

注意事项 确保输入 ANI0 引脚至 ANI11 引脚的电压没有超过额定值。如果输入至某通道的电压值为 AV_{REF0} 电压或者更高，那么此通道的转换值不确定，并且其它通道的转换值也会受到影响。

(14) AV_{REF0} 引脚

该引脚用于输入 A/D 转换器的参考电压。即使不使用 A/D 转换器，也要让该引脚的电压与 V_{DD} 引脚的电压相同。根据施加在 AV_{REF0} 引脚和 AV_{SS} 引脚之间的电压，将输入到 ANI0 至 ANI11 引脚的信号转化为数字信号。

(15) AV_{SS} 引脚

该引脚为 A/D 转换器的地电平引脚。即使不使用 A/D 转换器，也要保证该引脚的电压与 V_{SS} 引脚的电压相同。

15.4 寄存器

A/D 转换器由下列寄存器控制：

- A/D 转换器模式寄存器 0、1、2 (ADA0M0、ADA0M1、ADA0M2)
- A/D 转换器通道指定寄存器 0 (ADA0S)
- 掉电比较模式寄存器 (ADA0PFM)

以下寄存器也可以使用：

- A/D 转换结果寄存器 n (ADA0CRn)
- A/D 转换结果寄存器 nH (ADA0CRnH)
- 掉电比较阈值寄存器 (ADA0PFT)

(1) A/D 转换器模式寄存器 0 (ADA0M0)

ADA0M0 寄存器是 8 位寄存器，用于指定操作模式并控制转换操作。
该寄存器可进行字节读写或按位读写。但是，ADA0EF 0 是只读位。
系统复位后，该寄存器被设为 00H。

(1/2)

复位后: 00H R/W 地址: FFFFF200H								
	<7>	6	5	4	3	2	1	<0>
ADA0M0	ADA0CE	0	ADA0MD1	ADA0MD0	ADA0ETS1	ADA0ETS0	ADA0TMD	ADA0EF
ADA0CE		A/D 转换控制						
0		停止 A/D 转换						
1		使能 A/D 转换						
ADA0MD1	ADA0MD0	A/D 转换器操作模式说明						
0	0	连续选择模式						
0	1	连续扫描模式						
1	0	单次选择模式						
1	1	单次扫描模式						

ADA0ETS1	ADA0ETS0	外部触发(ADTRG 引脚) 输入有效边沿的说明
0	0	无边沿检测
0	1	下降沿检测
1	0	上升沿检测
1	1	双边沿检测

ADA0TMD	触发模式说明
0	软件触发模式
1	外部触发模式/定时器触发模式

ADA0EF	A/D 转换器状态说明
0	A/D 转换停止
1	A/D 转换进行中

- 注意事项**
- 在下列状态下禁止访问 ADA0M0 寄存器。详情参见 3.4.8 (2) 访问专用片上外设 I/O 寄存器。
 - 当 CPU 运行于副时钟，且主时钟振荡停止时。
 - 当 CPU 运行于内部振荡时钟工作时。
 - 对第 0 位的写入操作将被忽略。
 - A/D 转换使能 (ADA0CE 位 = 1) 时，禁止改变 ADA0M1.ADA0FR2 位至 ADA0M1.ADA0FR0 位。
 - 下列模式中，当 A/D 转换停止 (ADA0CE 位 = 0) 时，向 ADA0M0、ADA0M2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器写入数据，然后再使能 A/D 转换器的操作 (ADA0CE 位 = 1)
 - 正常转换模式下
 - 高速转换模式下的单次选择模式/单次扫描模式
 A/D 转换期间 (ADA0EF 位 = 1)，如果在其它模式下写入 ADA0M0、ADA0M2、ADA0S、ADA0PFM 和 ADA0PFT 寄存器，那么根据模式将会执行如下操作。
 - 软件触发模式下：
 - A/D 转换停止，并再次从头开始执行。
 - 硬件触发模式下：
 - A/D 转换停止，且设置为触发等待状态。
 - 为了选择外部触发模式/定时器触发模式 (ADA0TMD 位 = 1)，设置为高速转换模式 (ADA0M1.ADA0HS1 位 = 1)。A/D 转换操作使能 (ADA0CE 位 = 1) 后，在随后插入的振荡稳定时间期间，不要输入触发信号。
 - 当不使用 A/D 转换器时，将 ADA0CE 位设置为 0，停止转换操作，降低功耗。

(2) A/D 转换器模式寄存器 1 (ADA0M1)

ADA0M1 是 8 位寄存器，用于指定转换时间。

该寄存器可以按字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H R/W 地址: FFFFF201H

	7	6	5	4	3	2	1	0
ADA0M1	ADA0HS1	0	0	0	0	ADA0FR2	ADA0FR1	ADA0FR0

ADA0HS1	正常转换模式/高速模式说明(A/D转换时间)
0	正常转换模式
1	高速转换模式

- 注意事项**
1. A/D 转换使能 (ADA0M0.ADA0CE 位= 1) 时，禁止改变 ADA0M1 寄存器。
 2. 为了选择外部触发模式/定时器触发模式 (ADA0M0.ADA0TMD 位= 1)，设置高速转换模式 (ADA0HS1 位= 1)。A/D 转换操作使能 (ADA0CE 位= 1) 后，在随后插入的振荡稳定时间期间，不要输入触发信号。
 3. 请确保将第 6 位至第 3 位清为“0”。

备注 关于设置 A/D 转换时间的举例，参见表 15-2 和表 15-3。

表 15-2. 正常转换模式下转换时间的选择 (ADA0HS1 位= 0)

ADA0FR3 至 ADA0FR0 位	A/D 转换时间			
	稳定时间 + 转换时间 + 等待时间	48 MHz	32 MHz	24 MHz
0000	$26/f_{xx} + 52/f_{xx} + 54/f_{xx}$	2.75 μs	4.13 μs	5.50 μs
0001	$52/f_{xx} + 104/f_{xx} + 106/f_{xx}$	5.46 μs	8.19 μs	禁止设置
0010	$78/f_{xx} + 156/f_{xx} + 158/f_{xx}$	8.17 μs	禁止设置	禁止设置
0011	$100/f_{xx} + 208/f_{xx} + 210/f_{xx}$	禁止设置	禁止设置	禁止设置
0100	$100/f_{xx} + 260/f_{xx} + 262/f_{xx}$	禁止设置	禁止设置	禁止设置
0101	$100/f_{xx} + 312/f_{xx} + 314/f_{xx}$	禁止设置	禁止设置	禁止设置
0110	$100/f_{xx} + 364/f_{xx} + 366/f_{xx}$	禁止设置	禁止设置	禁止设置
0111	$100/f_{xx} + 416/f_{xx} + 418/f_{xx}$	禁止设置	禁止设置	禁止设置
1000	$100/f_{xx} + 468/f_{xx} + 470/f_{xx}$	禁止设置	禁止设置	禁止设置
1001	$100/f_{xx} + 520/f_{xx} + 522/f_{xx}$	禁止设置	禁止设置	禁止设置
1010	$100/f_{xx} + 572/f_{xx} + 574/f_{xx}$	禁止设置	禁止设置	禁止设置
1011	$100/f_{xx} + 624/f_{xx} + 626/f_{xx}$	禁止设置	禁止设置	禁止设置
1100	$100/f_{xx} + 676/f_{xx} + 678/f_{xx}$	禁止设置	禁止设置	禁止设置
1101	$100/f_{xx} + 728/f_{xx} + 730/f_{xx}$	禁止设置	禁止设置	禁止设置
1110	$100/f_{xx} + 780/f_{xx} + 782/f_{xx}$	禁止设置	禁止设置	禁止设置
1111	$100/f_{xx} + 832/f_{xx} + 834/f_{xx}$	禁止设置	禁止设置	禁止设置
其它	禁止设置			

备注	稳定时间:	A/D 转换建立时间 (1 μs 或更长)
	转换时间:	实际的 A/D 转换时间 (2.17 至 9.75 μs)
	等待时间:	下一次转换之前, 插入的等待时间
	f _{xx} :	主时钟频率

在正常转换模式下, 自 ADA0M0.ADA0CE 位置为 1, 稳定时间过后, 开始转换, 并且 A/D 转换仅在转换时间 (2.17 至 9.75 μs) 执行。转换结束后操作停止, 等待时间过后, 产生 A/D 转换结束中断请求信号 (INTAD)。

因为转换操作在等待时间内停止, 所以降低了工作电流。

- 注意事项**
1. 设置 $2.17 \mu\text{s} \leq A \text{ 转换时间} \leq 9.75 \mu\text{s}$ 。
 2. A/D 转换期间, 如果有数据写入 ADA0M0、ADA0M2、ADA0S、ADA0PFM 以及 ADA0PFT 寄存器或者有触发信号输入, 则重新执行转换。但是, 如果稳定时间结束时序与写入这些寄存器的操作发生冲突, 或者稳定时间结束时序与触发信号输入发生冲突, 那么将会重新插入 64 个时钟的稳定时间。
如果在插入的稳定时间结束时刻再次发生冲突, 则仍然重新插入稳定时间。因此, 不要将触发输入间隔和控制寄存器写入间隔设置为 64 个时钟或更少。

表 15-3. 高速转换模式下转换时间的选择 (ADA0HS1 位= 1)

ADA0FR3 至 ADA0FR0 位	A/D 转换时间			
	转换时间 (+ 等待时间)	48 MHz	32 MHz	24 MHz
0000	$52/f_{xx} (+26/f_{xx})$	禁止设置	禁止设置	2.17 μs
0001	$104/f_{xx} (+52/f_{xx})$	2.17 μs	3.25 μs	4.33 μs
0010	$156/f_{xx} (+78/f_{xx})$	3.25 μs	4.88 μs	6.50 μs
0011	$208/f_{xx} (+100/f_{xx})$	4.33 μs	6.50 μs	8.67 μs
0100	$260/f_{xx} (+100/f_{xx})$	5.42 μs	8.13 μs	禁止设置
0101	$312/f_{xx} (+100/f_{xx})$	6.50 μs	9.75 μs	禁止设置
0110	$364/f_{xx} (+100/f_{xx})$	7.58 μs	禁止设置	禁止设置
0111	$416/f_{xx} (+100/f_{xx})$	8.67 μs	禁止设置	禁止设置
1000	$468/f_{xx} (+100/f_{xx})$	9.75 μs	禁止设置	禁止设置
1001	$520/f_{xx} (+100/f_{xx})$	禁止设置	禁止设置	禁止设置
1010	$572/f_{xx} (+100/f_{xx})$	禁止设置	禁止设置	禁止设置
1011	$624/f_{xx} (+100/f_{xx})$	禁止设置	禁止设置	禁止设置
1100	$676/f_{xx} (+100/f_{xx})$	禁止设置	禁止设置	禁止设置
1101	$728/f_{xx} (+100/f_{xx})$	禁止设置	禁止设置	禁止设置
1110	$780/f_{xx} (+100/f_{xx})$	禁止设置	禁止设置	禁止设置
1111	$832/f_{xx} (+100/f_{xx})$	禁止设置	禁止设置	禁止设置
其它	禁止设置			

备注 稳定时间: A/D 转换建立时间 (1 μs 或更长)
 转换时间: 实际的 A/D 转换时间 (2.17 至 9.75 μs)
 f_{xx}: 主时钟频率

在高速转换模式下, 自 ADA0M0.ADA0CE 位置为 1, 稳定时间过后, 开始转换, 并且 A/D 转换仅在转换时间 (2.17 至 9.75 μs) 执行。转换结束后立即产生 A/D 转换结束中断请求信号 (INTAD)。

在连续转换模式下, 仅在第一次转换前插入稳定时间, 而在第二次转换之后 (A/D 转换器仍然运行) 不会插入。

- 注意事项 1.** 设置为 $2.17 \mu\text{s} \leq \text{转换时间} \leq 9.75 \mu\text{s}$
2. 在高速转换模式下, 振荡稳定期间, 触发信号输入和对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 和 ADA0PFT 寄存器的重写都被禁止。

(3) A/D 转换器模式寄存器 2 (ADA0M2)

ADA0M2 寄存器用于指定硬件触发模式。

该寄存器可以按字节读写或按位读写。

系统复位后，该寄存器被设为 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	硬件触发模式说明
0	0	外部触发模式 (检测到ADTRG引脚有效边沿时)
0	1	定时器触发模式 0 (产生INTTAA2CC0中断请求时)
1	0	定时器触发模式 1 (产生INTTAA2CC中断请求时)
1	1	定时器触发模式 2 (TQTADT0 信号)

- 注意事项**
- 下列模式时，当 A/D 转换停止 (ADA0M0.ADA0CE 位 = 0) 时，向 ADA0M2 寄存器写入数据，然后使能 A/D 转换器的操作 (ADA0CE 位 = 1)
 - 正常转换模式
 - 高速转换模式下的单次选择模式/单次扫描模式
 - 请确保将第 7 位至第 2 位清为“0”。

(4) 模拟输入通道指定寄存器 0 (ADA0S)

ADA0S 寄存器用于指定将模拟电压转换为数字信号的输入引脚。

该寄存器可以按字节或按位进行读写。

系统复位后，该寄存器被设为 00H。

复位后：00H R/W 地址：FFFFFF202H

	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 至 ANI2
0	0	1	1	ANI3	ANI0 至 ANI3
0	1	0	0	ANI4	ANI0 至 ANI4
0	1	0	1	ANI5	ANI0 至 ANI5
0	1	1	0	ANI6	ANI0 至 ANI6
0	1	1	1	ANI7	ANI0 至 ANI7
1	0	0	0	ANI8	ANI0 至 ANI8
1	0	0	1	ANI9	ANI0 至 ANI9
1	0	1	0	ANI10	ANI0 至 ANI10
1	0	1	1	ANI11	ANI0 至 ANI11
1	1	0	0	禁止设置	禁止设置
1	1	0	1	禁止设置	禁止设置
1	1	1	0	禁止设置	禁止设置
1	1	1	1	禁止设置	禁止设置

注意事项 1. 下列模式中，当 A/D 转换停止 (ADA0M0.ADA0CE 位 = 0) 时，向 ADA0S 寄存器写入数据，然后使能 A/D 转换器的操作 (ADA0CE 位 = 1)。

- 正常转换模式
- 高速转换模式下的单次选择模式/单次扫描模式

2. 请确保将第 7 位至第 4 位清为“0”。

(5) A/D 转换结果寄存器 n、nH (ADA0CRn, ADA0CRnH)

ADA0CRn 寄存器和 ADA0CRnH 寄存器用于存储 A/D 转换结果。

这些寄存器是以 8 位或 16 位宽度访问的只读寄存器。但是，ADA0CRn 寄存器通过 16 位访问，指定 ADA0CRnH 寄存器为 8 位访问。高 10 位读取 10 位转换结果读入 ADA0CRn 寄存器，同时低 6 位的读取值为 0。高 8 位转换结果读入 ADA0CRnH 寄存器。

注意事项 下列状态下，禁止访问 ADA0CRn 寄存器和 ADA0CRnH 寄存器。

详细说明，参见 3.4.8 (2) 访问片上外设 I/O 专用寄存器。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
- 当 CPU 运行于内部振荡时钟工作时。

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																																
ADA0CRn (n = 0 至 11)		<table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 0 5px;">15</td><td style="padding: 0 5px;">14</td><td style="padding: 0 5px;">13</td><td style="padding: 0 5px;">12</td><td style="padding: 0 5px;">11</td><td style="padding: 0 5px;">10</td><td style="padding: 0 5px;">9</td><td style="padding: 0 5px;">8</td><td style="padding: 0 5px;">7</td><td style="padding: 0 5px;">6</td><td style="padding: 0 5px;">5</td><td style="padding: 0 5px;">4</td><td style="padding: 0 5px;">3</td><td style="padding: 0 5px;">2</td><td style="padding: 0 5px;">1</td><td style="padding: 0 5px;">0</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">AD9</td><td style="border: 1px solid black; padding: 2px;">AD8</td><td style="border: 1px solid black; padding: 2px;">AD7</td><td style="border: 1px solid black; padding: 2px;">AD6</td><td style="border: 1px solid black; padding: 2px;">AD5</td><td style="border: 1px solid black; padding: 2px;">AD4</td><td style="border: 1px solid black; padding: 2px;">AD3</td><td style="border: 1px solid black; padding: 2px;">AD2</td><td style="border: 1px solid black; padding: 2px;">AD1</td><td style="border: 1px solid black; padding: 2px;">AD0</td><td style="border: 1px solid black; padding: 2px;">0</td><td style="border: 1px solid black; padding: 2px;">0</td><td style="border: 1px solid black; padding: 2px;">0</td><td style="border: 1px solid black; padding: 2px;">0</td><td style="border: 1px solid black; padding: 2px;">0</td><td style="border: 1px solid black; padding: 2px;">0</td> </tr> </table>	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																			
AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0																			
ADA0CRnH (n = 0 至 11)		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																																
		<table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 0 5px;">7</td><td style="padding: 0 5px;">6</td><td style="padding: 0 5px;">5</td><td style="padding: 0 5px;">4</td><td style="padding: 0 5px;">3</td><td style="padding: 0 5px;">2</td><td style="padding: 0 5px;">1</td><td style="padding: 0 5px;">0</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">AD9</td><td style="border: 1px solid black; padding: 2px;">AD8</td><td style="border: 1px solid black; padding: 2px;">AD7</td><td style="border: 1px solid black; padding: 2px;">AD6</td><td style="border: 1px solid black; padding: 2px;">AD5</td><td style="border: 1px solid black; padding: 2px;">AD4</td><td style="border: 1px solid black; padding: 2px;">AD3</td><td style="border: 1px solid black; padding: 2px;">AD2</td> </tr> </table>	7	6	5	4	3	2	1	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2																
7	6	5	4	3	2	1	0																											
AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2																											

注意事项 对 ADA0M0 寄存器和 ADA0S 寄存器的写入操作可能会导致 ADA0CRn 寄存器的内容不确定。转换完成后，在写入 ADA0M0 寄存器和 ADA0S 寄存器之前读取转换结果。如果没有在上述时序处执行读取，则读取结果可能会有误。

输入到模拟输入引脚（ANI0 至 ANI11）的模拟电压与转换结果（ADA0CRn 寄存器）之间的关系如下：

$$SAR = INT \left(\frac{V_{IN}}{AV_{REF0}} \times 1,024 + 0.5 \right)$$

$$ADA0CR_n^* = SAR \times 64$$

或，

$$(SAR - 0.5) \times \frac{AV_{REF0}}{1,024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{REF0}}{1,024}$$

INT () : 返回括号内数值整数部分的函数

V_{IN}: 模拟输入电压

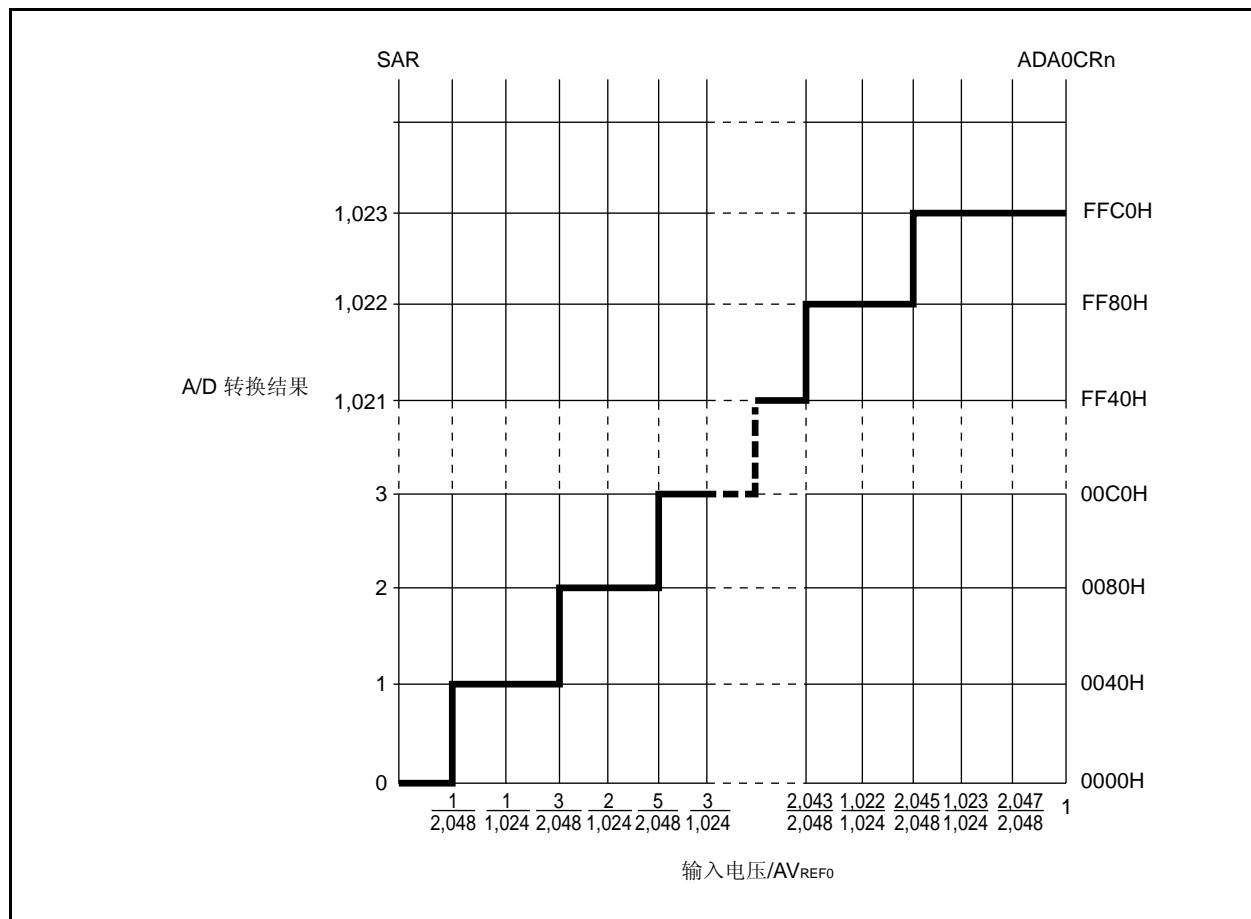
AV_{REF0}: AV_{REF0} 引脚电压

ADA0CR: ADA0CRn 寄存器的值

注 ADA0CRn 寄存器的低 6 位固定为 0。

下图显示了模拟输入电压和 A/D 转换结果之间关系。

图 15-2. 模拟输入电压和 A/D 转换结果的关系



(6) 掉电比较模式寄存器 (ADA0PFM)

ADA0PFM 寄存器是 8 位寄存器，用于设置掉电比较模式。

该寄存器可按字节或按位进行读写。

系统复位后，该寄存器被设为 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	掉电比较使能/禁止的选择
0	掉电比较禁止
1	掉电比较使能

ADA0PFC	掉电比较模式的选择
0	当 $ADA0CRnH \geq ADA0PFT$ 时，产生中断请求信号 (INTAD)
1	当 $ADA0CRnH < ADA0PFT$ 时，产生中断请求信号 (INTAD)

- 注意事项**
1. 在选择模式下，将设置到 ADA0PFT 寄存器的 8 位数据与 ADA0S 寄存器指定的 ADA0CRnH 寄存器的值进行比较。如果该结果符合 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CRn 寄存器，并且产生 INTAD 信号。但是，如果不符合条件，将不产生中断信号。
 2. 在扫描模式下，将设置到 ADA0PFT 寄存器的 8 位数据与 ADA0CR0H 寄存器的内容进行比较。如果该结果符合 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CR0 寄存器中，并且产生 INTAD 信号。但是，如果不符合条件，将不产生 INTAD 信号。无论比较结果如何，扫描操作完成之前，都会继续扫描，并且将转换结果存储在 ADA0CRn 寄存器中。但是，扫描操作完成后不产生 INTAD 信号。
 3. 下列模式时，当 A/D 转换停止 (ADA0M0.ADA0CE 位 = 0) 时，向 ADA0PFM 寄存器写入数据，然后使能 A/D 转换操作 (ADA0CE 位 = 1)。
 - 正常转换模式
 - 高速转换模式下的单次选择模式/单次扫描模式

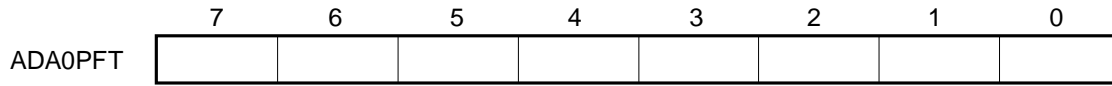
(7) 掉电比较阈值寄存器 (ADA0PFT)

ADA0PFT 寄存器用于设置掉电比较模式下的比较值。

该寄存器可以按字节或按位进行读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H R/W 地址: FFFFF205H



注意事项 下列模式时，当 A/D 转换停止 (ADA0M0.ADA0CE 位 = 0) 时，向 ADA0PFT 寄存器写入数据，然后使能 A/D 转换操作 (ADA0CE 位 = 1)。

- 正常转换模式
- 高速转换模式下的单次选择模式/单次扫描模式

15.5 操作

15.5.1 基本操作

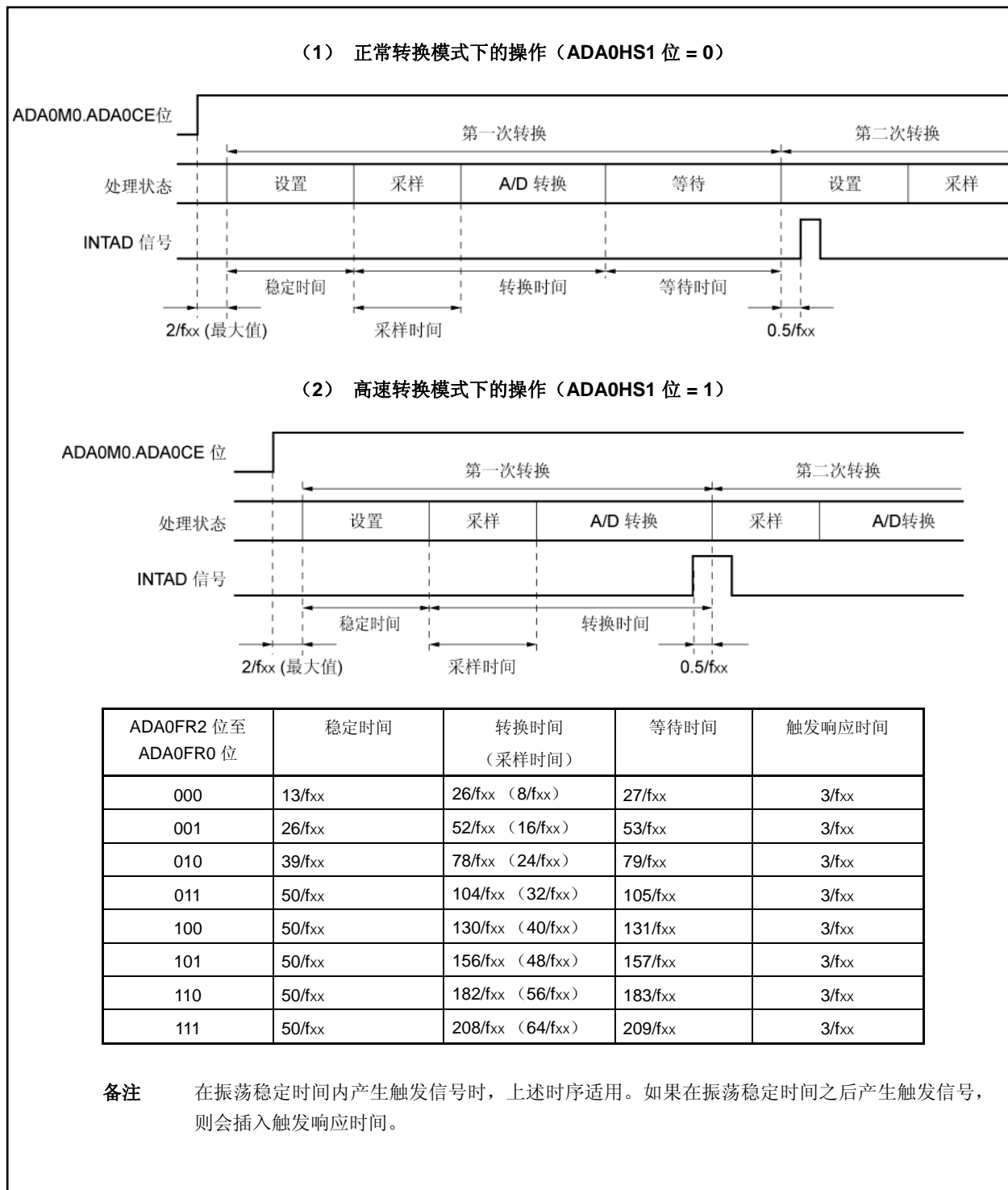
- <1> 使用 ADA0M0、ADA0M1、ADA0M2 和 ADA0S 寄存器来设置执行 A/D 转换的操作模式、触发模式以及转换时间。当置位 ADA0M0 寄存器的 ADA0CE 位时，在软件触发模式下转换开始，在外部或定时器触发模式下 A/D 转换器等待触发信号。
- <2> A/D 转换开始后，采样&保持电路对所选模拟输入通道的输入电压进行采样。
- <3> 采样&保持电路在指定时间内对输入通道进行采样，然后采样&保持电路进入保持状态，并且保持采样得到的输入模拟电压，直到 A/D 转换完成。
- <4> 对逐次逼近寄存器（SAR）的第 9 位置位。设置比较电压产生 DAC 为 $(1/2) AV_{REF0}$ 。
- <5> 电压比较器对比较电压产生 DAC 的电压和模拟输入电压之间的电压差进行比较。如果模拟输入电压高于 $(1/2) AV_{REF0}$ ，SAR 寄存器的 MSB 保持置位。如果模拟输入电压低于 $(1/2) AV_{REF0}$ ，MSB 位清零。
- <6> 接着，SAR 的第 8 位被自动置位，并启动下一次比较。根据先前所设定的位 9 的值，比较电压产生 DAC 的选择如下：
 - 第 9 位 = 1： $(3/4) AV_{REF0}$
 - 第 9 位 = 0： $(1/4) AV_{REF0}$
 该电压和模拟输入电压比较，并且根据比较结果，对第 8 位的操作如下：
 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> 此项比较一直持续，直到 SAR 寄存器的第 0 位。
- <8> 10 位比较完成时，有效数字结果存储在寄 SAR 寄存器，然后再传送并存储在 ADA0CRn 寄存器。之后，产生 A/D 转换结束中断请求信号（INTAD）。
- <9> 单次选择模式下，转换停止^注。单次扫描模式下，扫描完成之后，立刻停止转换^注。连续选择模式下，重复步骤<2>至步骤<8>直到 ADA0M0.ADA0CE 位清为 0。连续扫描模式下，每个通道都重复步骤<2>至步骤<8>。

注 在外部触发模式时、定时器触发模式 0 时或定时器触发模式 1 时，进入触发等待状态。

备注 触发等待状态指稳定时间过后的状态。

15.5.2 转换操作时序

图 15-3. 转换操作时序（连续转换）



15.5.3 触发模式

通过设置触发模式指定转换操作的启动时序。触发模式包括软件触发模式和硬件触发模式。硬件触发模式包含定时器触发模式 0 和 1，以及外部触发模式。ADA0M0.ADA0TMD 位用于设置触发模式。通过 ADA0M2.ADA0TMD1 位和 ADA0M2.ADA0TMD0 位设置硬件触发模式。

(1) 软件触发模式

当 ADA0M0.ADA0CE 位置 1 时，对 ADA0S 寄存器指定的模拟输入引脚（ANI0 引脚至 ANI11 引脚）的信号进行转换。当转换完成时，转换结果存储在 ADA0CRn 寄存器。同时，产生 A/D 转换结束中断信号（INTAD）。

如果通过 ADA0M0.ADA0MD1 位和 ADA0M0.ADA0MD0 位指定操作模式为连续选择/扫描模式，那么第一次转换完成后，将重复下一次转换，直到 ADA0CE 位被清 0。如果操作模式是单次选择/扫描模式，那么转换执行一次后结束。

转换开始时，ADA0M0.ADA0EF 位设为 1（表示转换正在进行）。

如果在转换期间对 ADA0M0、ADA、ADA0S、ADA0PFM 或 ADA0PFT 寄存器执行写入操作，那么此次转换中止，并再次从头开始转换。但是，在正常转换模式和高速转换模式下的单次选择模式/单次扫描模式中，禁止写入这些寄存器。

(2) 外部触发模式

该模式下，当外部触发信号输入（至 ADTRG 引脚）时，对 ADA0S 寄存器指定的模拟输入引脚（ANI0 至 ANI11）的信号进行转换。通过 ADA0M0.ADA0ETS1 位和 ADA0M0.ADA0ETS0 位来指定外部触发器的哪个边沿（也就是上升沿、下降沿或双边沿）被检测。ADA0CE 位设为 1 时，A/D 转换器等待触发，且当外部触发信号输入后，开始转换。

无论通过 ADA0MD1 位和 ADA0MD0 位设置的操作模式是连续选择、连续扫描、单次选择还是单次扫描模式，转换完成后，转换结果都存储在 ADA0CRn 寄存器中。同时，产生 INTAD 信号，并且 A/D 转换器等待下一次触发信号。

转换开始时，ADA0EF 位设为 1（表示转换正在进行）。然而，在 A/D 转换器等待触发期间，ADA0EF 位清为 0。如果在转换操作期间输入有效触发信号，转换操作中止，并再次从头开始。

如果在转换操作期间对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器执行写入操作，那么此次转换停止，并且 A/D 转换器等待再次触发。但是，在单次选择模式/单次扫描模式下，禁止写入这些寄存器。

注意事项 为了选择外部触发模式，需要设置为高速转换模式。A/D 转换操作使能（ADA0M0.ADA0CE 位= 1）后，在随后插入的振荡稳定时间期间，不要输入触发信号。

备注 触发等待状态指稳定时间过后的状态。

(3) 定时器触发模式

该模式下，通过该定时器的捕获/比较寄存器（连接至定时器）的比较匹配中断请求信号（INTTAA2CC0 或 INTTAA2CC1），开始对 ADA0S 寄存器指定的模拟输入引脚（ANI0 至 ANI11）的信号进行转换。通过 ADA0TMD1 位和 ADA0TMD0 位选择 INTTAA2CC0 信号或 INTTAA2CC1 信号，并且在指定的比较匹配中断请求信号的上升沿处开始转换。当 ADA0CE 位设为 1 时，A/D 转换器等待触发，当定时器的比较匹配中断请求信号输入后，开始转换。

转换完成时，无论通过 ADA0MD1 位和 ADA0MD0 位设置的操作模式是连续选择、连续扫描、单次选择还是单次扫描模式，转换结果都存储在 ADA0CRn 寄存器中。同时，产生 INTAD 信号，并且 A/D 转换器等待下一次触发。

转换开始时，ADA0EF 位设为 1（表示转换正在进行）。但是，在 A/D 转换器等待触发期间，ADA0EF 位清为 0。如果在转换操作期间输入有效触发，转换操作中止，并再次从头开始。

如果在转换操作期间对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器执行写入操作，那么此次转换停止，并且 A/D 转换器等待再次触发。但是，在单次选择模式/单次扫描模式下，禁止写入这些寄存器。

注意事项 为了选择定时器触发模式，需要设置为高速转换模式。A/D 转换操作使能（ADA0M0.ADA0CE 位=1）后，在随后插入的振荡稳定时间期间，不要输入触发信号。

备注 触发等待状态指稳定时间过后的状态。

15.5.4 操作模式

有四种操作模式可供选择：连续选择模式，连续扫描模式，单次选择模式以及单次扫描模式。在这四种操作模式下，需设置 ANI0 引脚至 ANI11 引脚。

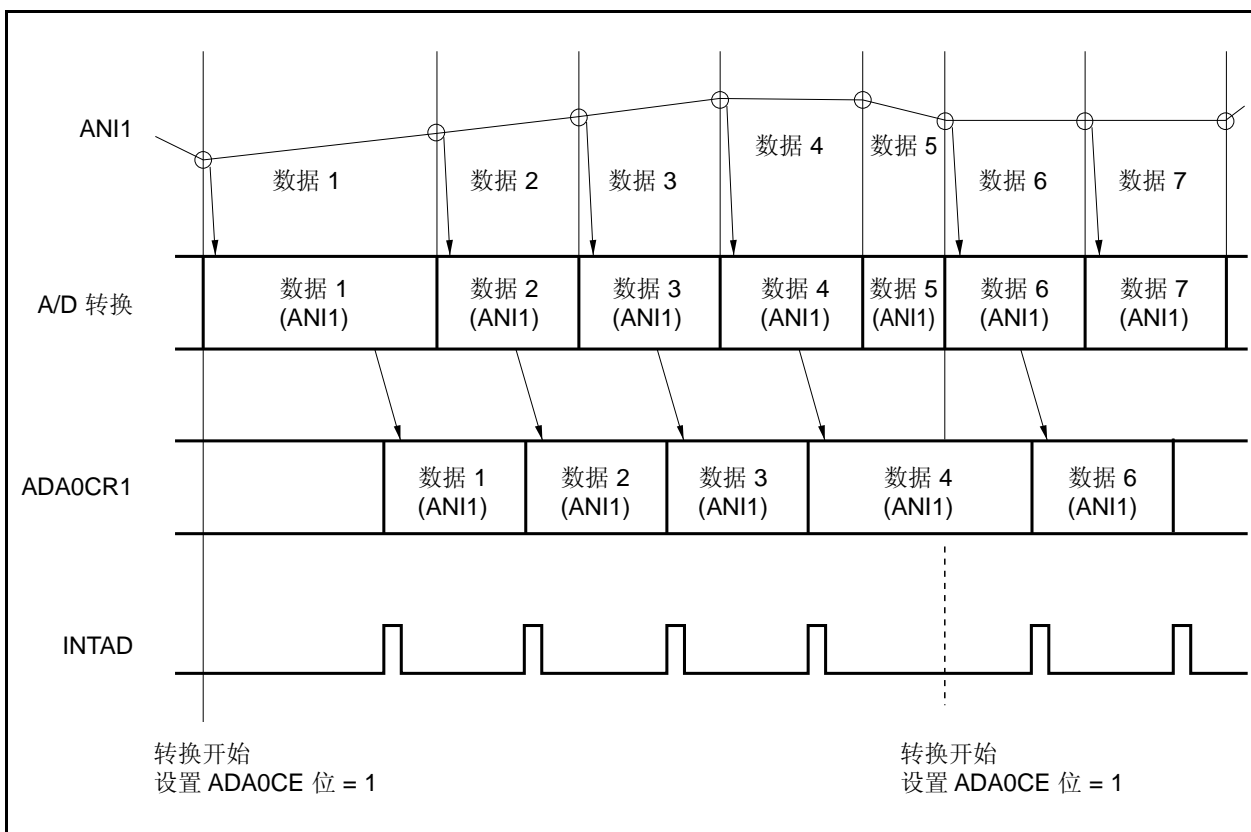
通过 ADA0M0.ADA0MD1 位和 ADA0M0.ADA0MD0 位来选择操作模式。

(1) 连续选择模式

该模式下，ADA0S 寄存器选择一个模拟输入引脚的电压被连续转换为数字值。

转换结果保存在该模拟输入引脚相对应的 ADA0CRn 寄存器中。该模式下，按照一一对应的方式，一个模拟输入引脚对应一个 ADA0CRn 寄存器。每次 A/D 转换完成后，都会产生 A/D 转换结束中断请求信号 (INTAD)。转换完成后，则开始下一次转换，除非 ADA0M0.ADA0CE 位被清为 0 (n = 0 至 11)。

图 15-4. 连续选择模式操作的时序举例 (ADA0S 寄存器= 01H)

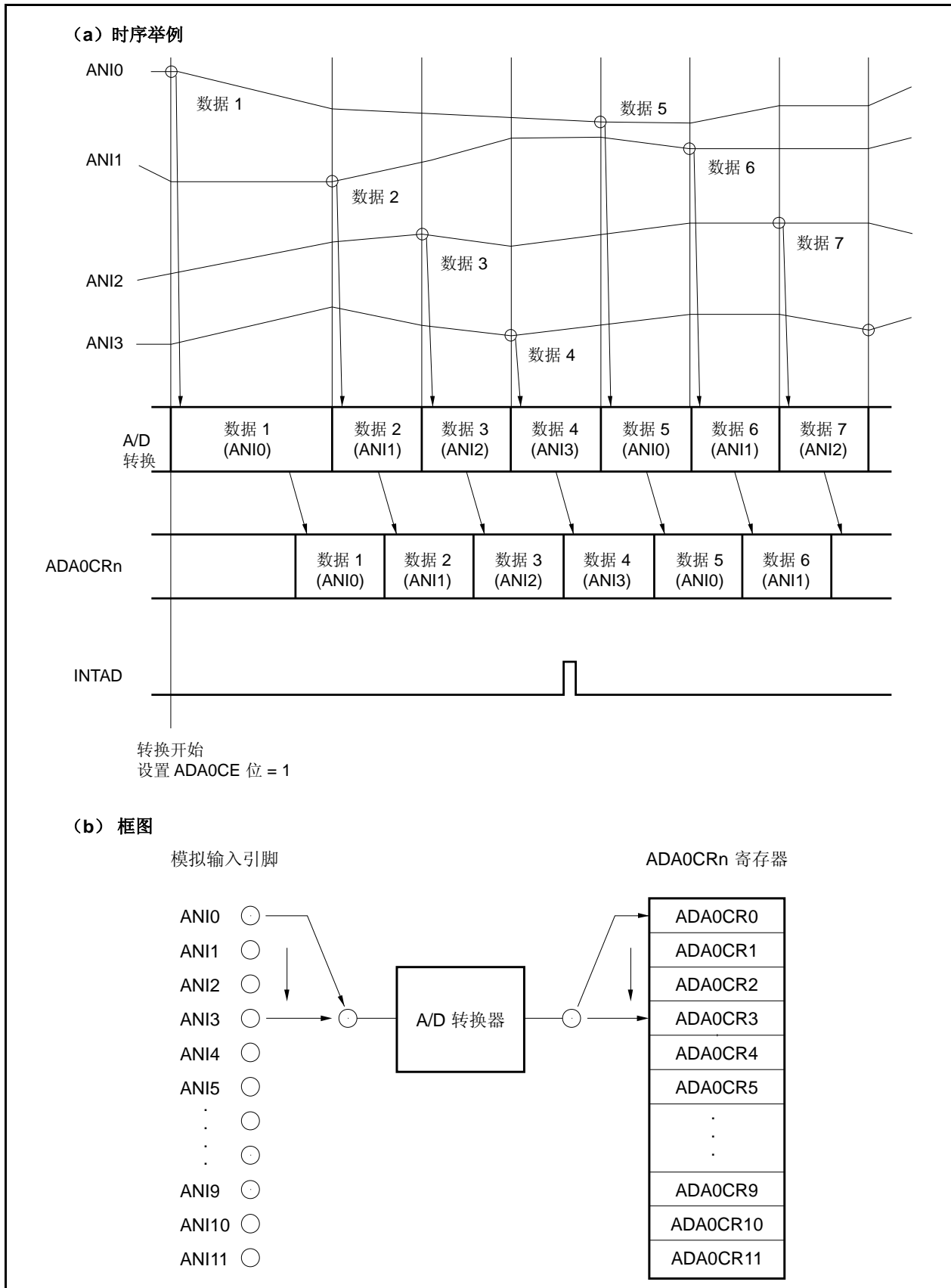


(2) 连续扫描模式

该模式下，模拟输入引脚是顺序选择的，从 ANI0 引脚至 ADA0S 寄存器指定的引脚，并且将它们的输入值连续转换为数字值。

各个通道的转换结果保存在与各个模拟输入引脚相对应的 ADA0CRn 寄存器中。当寄存器指定的模拟输入引脚转换完成时，产生 INTAD 信号，并且又从 ANI0 引脚开始 A/D 扫描转换，除非 ADA0CE 位被清为 0 (n = 0 至 11)。

图 15-5. 连续扫描模式操作的时序举例 (ADA0S 寄存器= 03H)

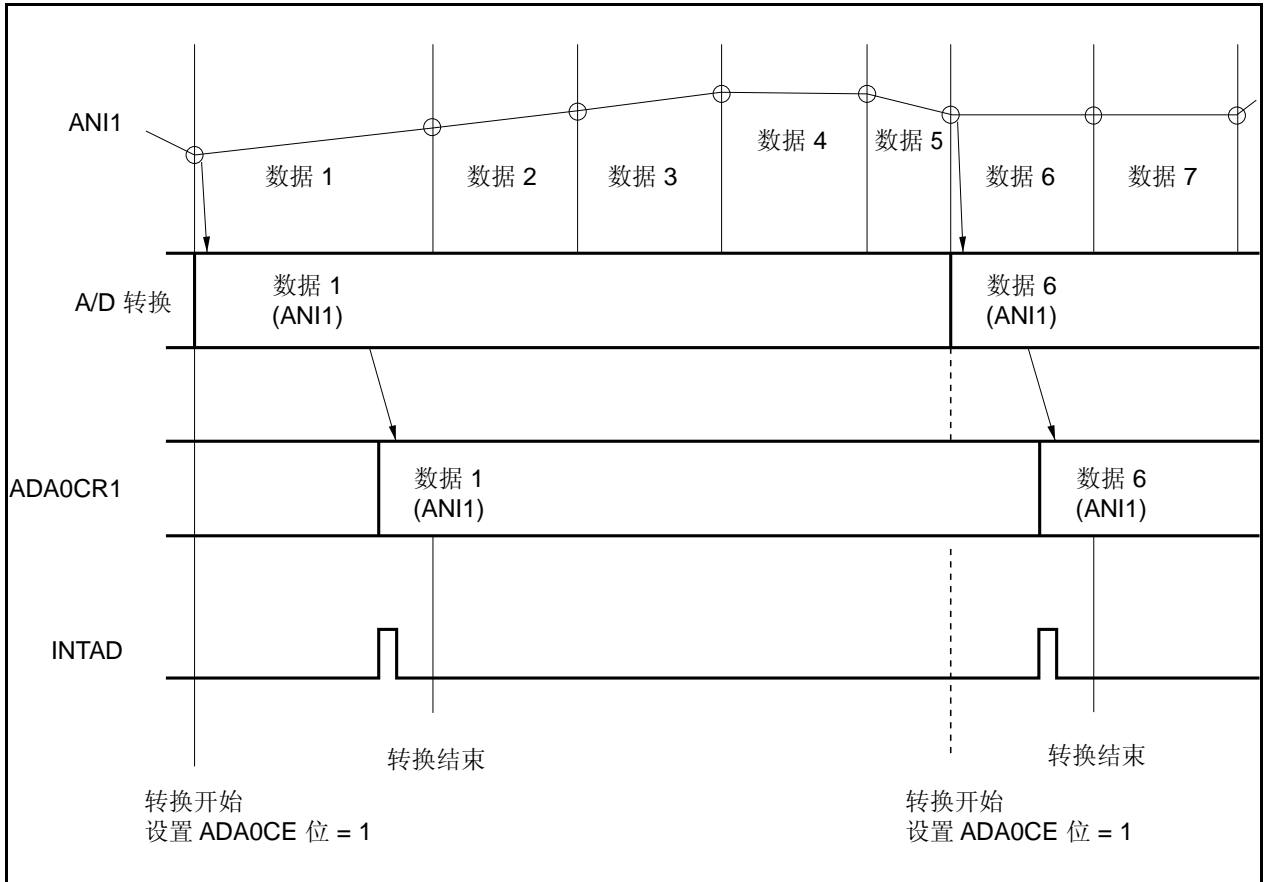


(3) 单次选择模式

该模式下，对 ADA0S 寄存器指定的一路模拟输入引脚电压仅进行一次转换为数字值的操作。

转换结果保存在该模拟输入引脚相对应的 ADA0CRn 寄存器中。该模式下，模拟输入引脚与 ADA0CRn 寄存器一一对应。A/D 转换完成一次后，产生 INTAD 信号。上述操作完成后，A/D 转换操作停止 (n = 0 至 11)。

图 15-6. 单次选择模式操作的时序举例 (ADA0S 寄存器= 01H)

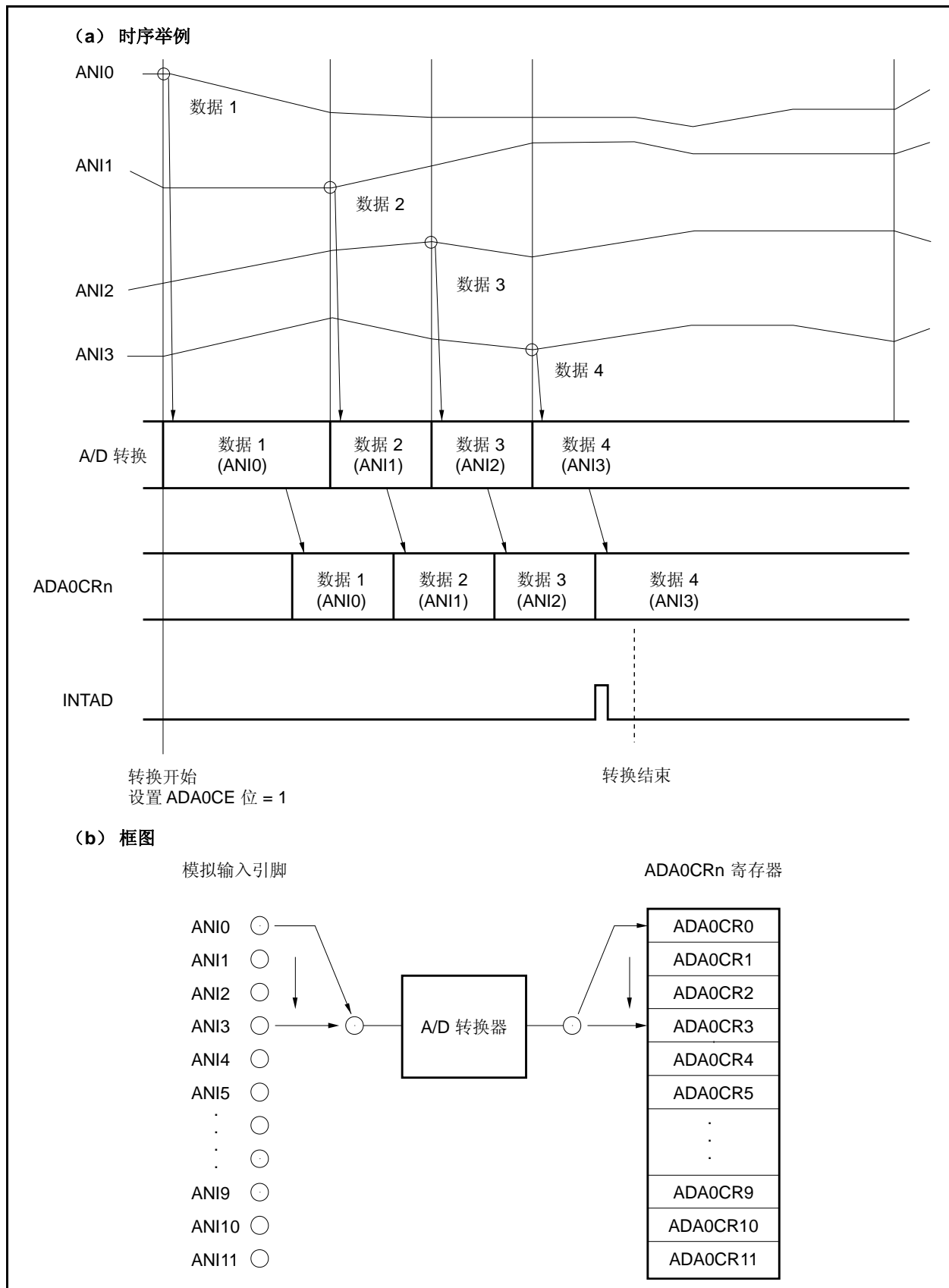


(4) 单次扫描模式

该模式下，模拟输入引脚是顺序选择的，从 ANI0 引脚至 ADA0S 寄存器指定的引脚，并且将它们的输入值转换为数字值。

各个转换结果保存在各个模拟输入引脚相对应的 ADA0CRn 寄存器中。当 ADA0S 寄存器指定的模拟输入引脚转换完成时，产生 INTAD 信号。上述操作完成后，A/D 转换停止 (n=0 到 11)。

图 15-7. 单次扫描模式操作的时序举例 (ADA0S 寄存器= 03H)



15.5.5 掉电比较模式

A/D 转换结束中断请求信号 (INTAD) 可以通过 ADA0PFM 寄存器和 ADA0PFT 寄存器控制。

- 当 ADA0PFM.ADA0PFE 位= 0 时，每次转换完成 (正常使用 A/D 转换器) 都产生 INTAD 信号。
- 当 ADA0PFE 位= 1 且 ADA0PFM.ADA0PFC 位= 0 时，ADA0CRnH 寄存器的值与转换完成时 ADA0PFT 寄存器的值进行比较，并且仅当 $ADA0CRnH \geq ADA0PFT$ 时，产生 INTAD 信号。
- 当 ADA0PFE 位= 1 且 ADA0PFC 位= 1 时，ADA0CRnH 寄存器的值与转换完成时 ADA0PFT 寄存器的值进行比较，并且仅当 $ADA0CRnH < ADA0PFT$ 时，产生 INTAD 信号。

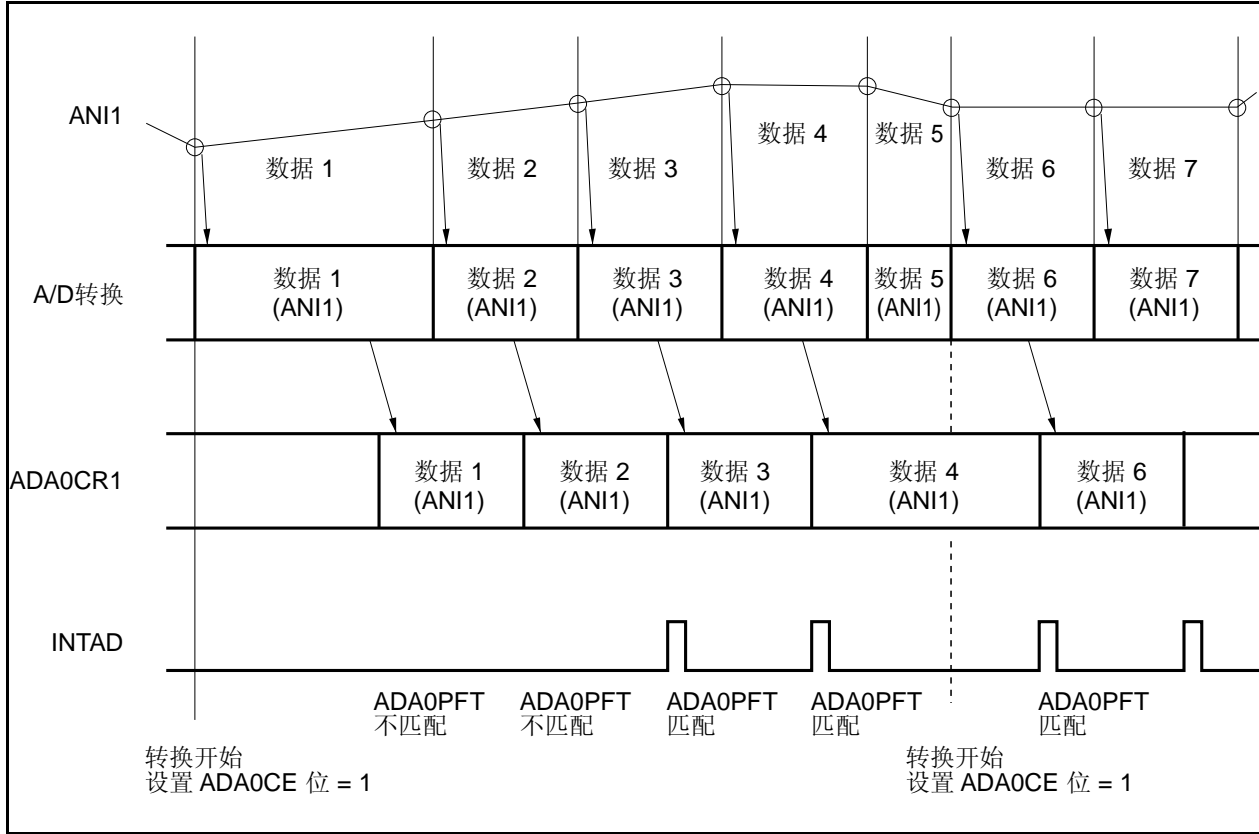
备注 n = 0 至 11

掉电比较模式中，有四种模式可供选择：连续选择模式，连续扫描模式，单次选择模式以及单次扫描模式。在这四种操作模式中，需设置 ANI0 引脚至 ANI11 引脚。

(1) 连续选择模式

该模式下，ADA0S 寄存器指定的模拟输入引脚的电压转换结果与 ADA0PFT 寄存器的设置值相比较。如果掉电比较的结果符合 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CRn 寄存器，并产生 INTAD 信号。如果不符合条件，转换结果仍存储在 ADA0CRn 寄存器中，但不产生 INTAD 信号。第一次转换完成后，则开始下一次转换，除非 ADA0M0.ADA0CE 位被清为 0 (n = 0 至 11)。

图 15-8. 连续选择模式操作的时序举例（掉电比较模式时：ADA0S 寄存器 = 01H）

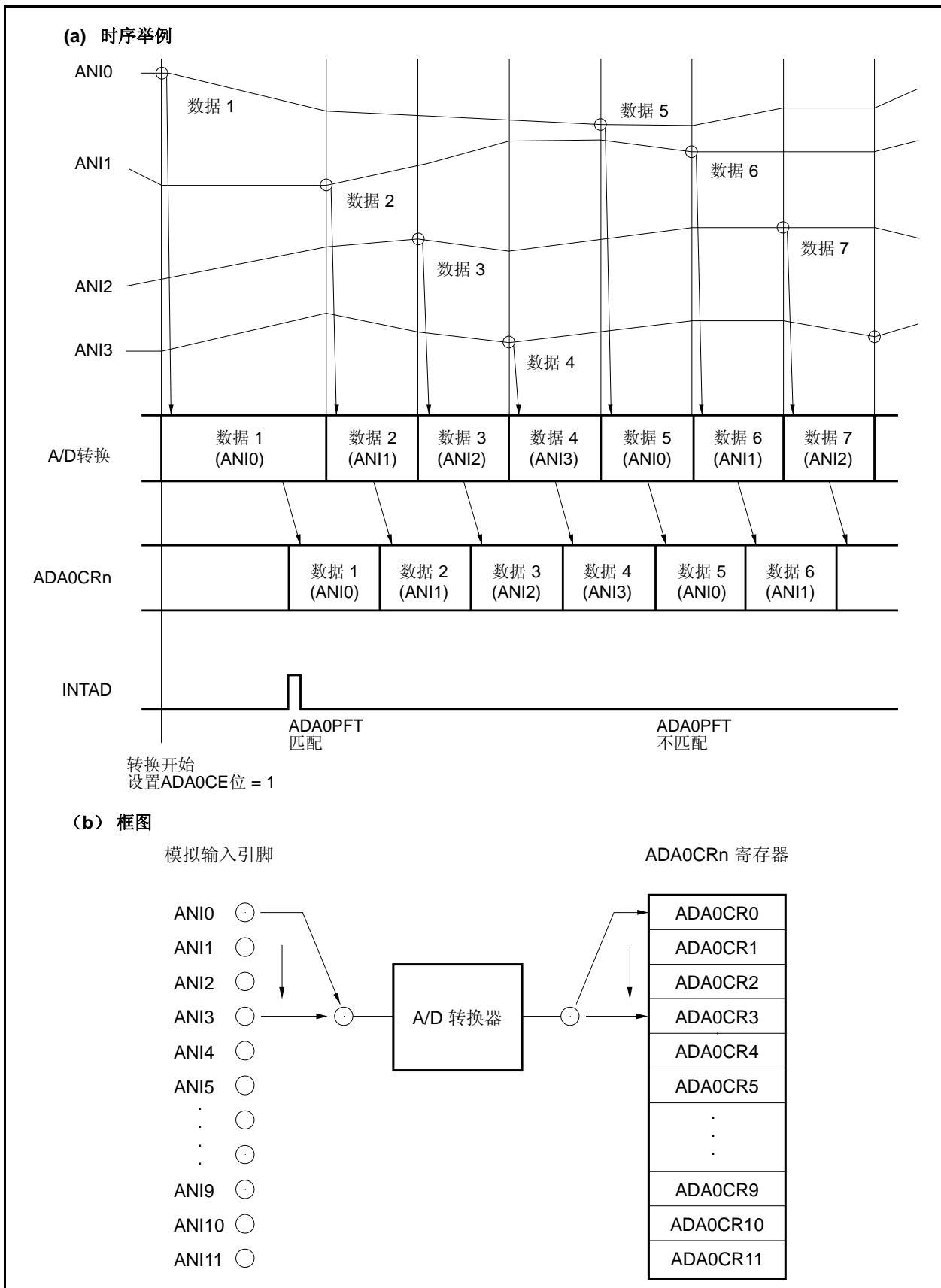


(2) 连续扫描模式

该模式下，模拟输入引脚是顺序选择的，从 ANI0 引脚至 ADA0S 寄存器指定的引脚，对它们的模拟输入值进行 A/D 转换，并且将 ADA0CR0H 寄存器的设置值与 ADA0PFT 寄存器的值作比较。如果掉电比较的结果符合 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CR0 寄存器，并产生 INTAD 信号。如果不符合条件，转换结果仍存储在 ADA0CR0 寄存器中，但不产生 INTAD 信号。

第一次转换结果存储在 ADA0CR0 寄存器之后，顺序对模拟输入引脚上的电压进行转换并存储，至 ADA0S 寄存器指定的模拟输入引脚。转换完成后，又从 ANI0 引脚启动新一轮转换，直到 ADA0CE 位被清为 0。

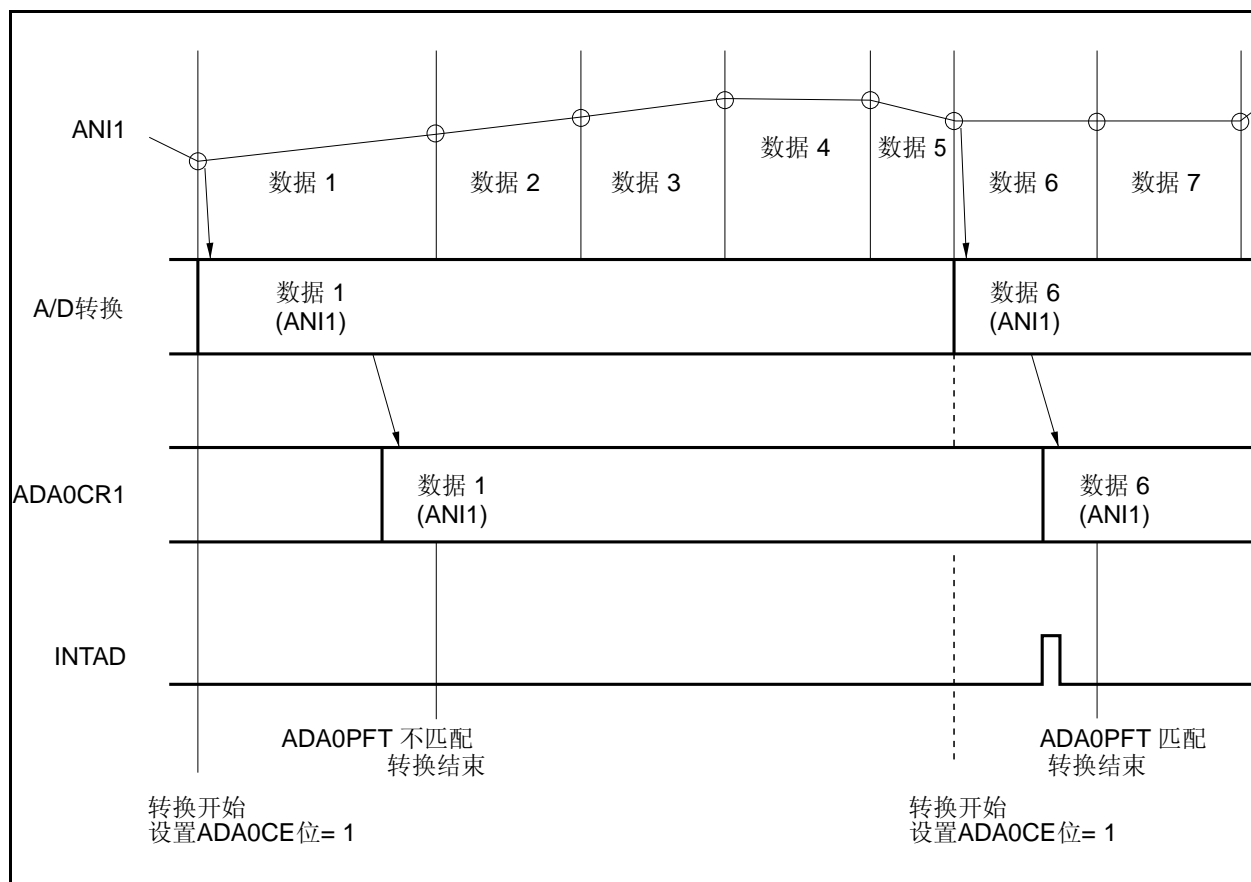
图 15-9. 连续扫描模式操作的时序举例（掉电比较模式时：ADA0S 寄存器 = 03H）



(3) 单次选择模式

该模式下，ADA0S 寄存器指定的模拟输入引脚的电压转换结果与 ADA0PFT 寄存器的设置值相比较。如果掉电比较的结果符合由 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CRn 寄存器，并产生 INTAD 信号。如果不符合条件，转换结果仍存储在 ADA0CRn 寄存器中，但不产生 INTAD 信号。上述操作完成后，停止转换。

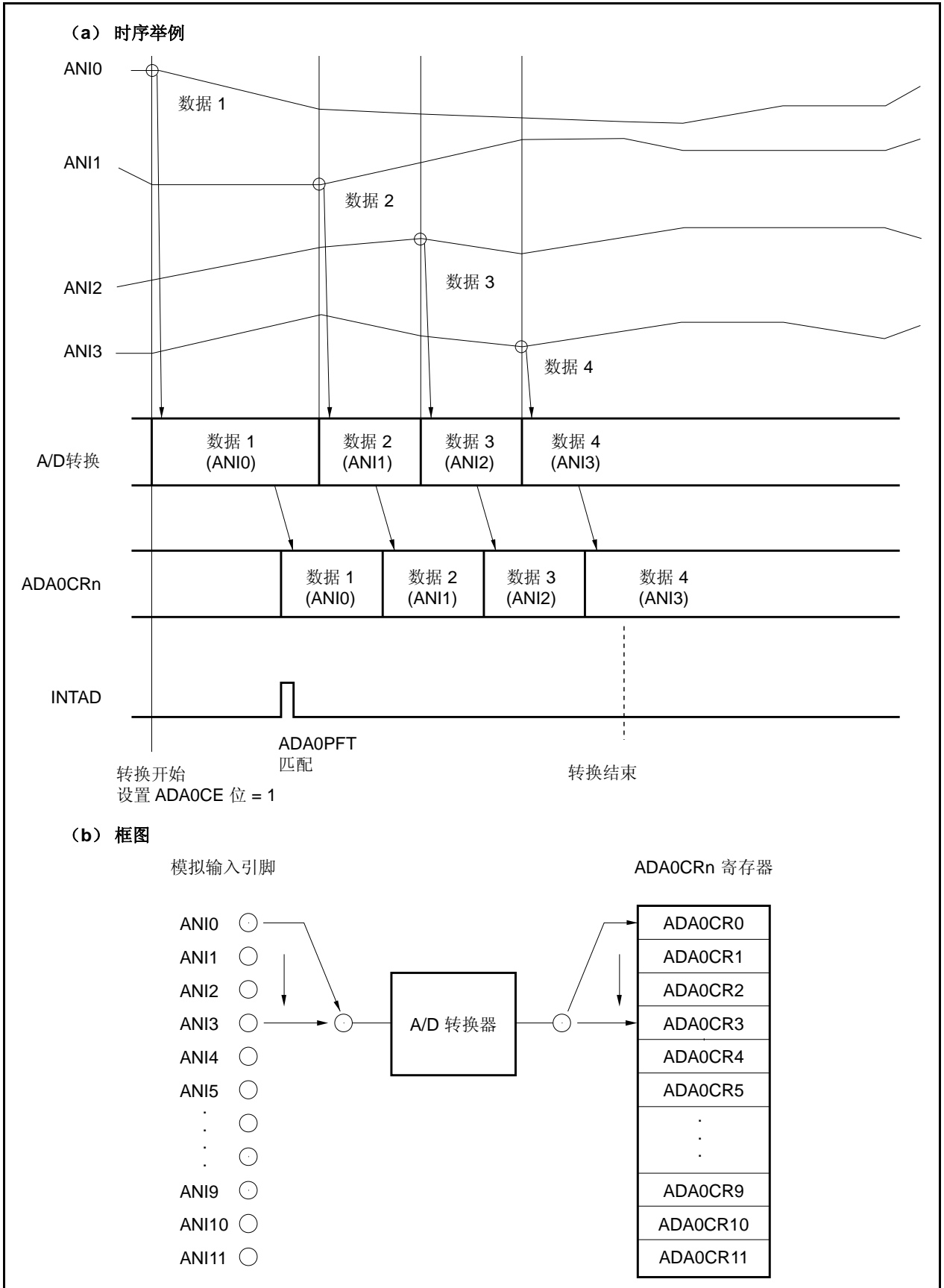
图 15-10. 单次选择模式操作的时序举例（掉电比较模式时：ADA0S 寄存器 = 01H）



(4) 单次扫描模式

该模式下，模拟输入引脚是顺序选择的，从 ANI0 引脚至 ADA0S 寄存器指定的引脚，对它们的模拟输入值进行 A/D 转换，存储电压转换结果，并且将通道 0 的 ADA0CR0H 寄存器设置值与 ADA0PFT 寄存器的设置值作比较。如果掉电比较的结果符合由 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CR0 寄存器，并产生 INTAD 信号。如果不符合条件，转换结果仍存储在 ADA0CR0 寄存器中，但不产生 INTAD0 信号。第一次转换结果存储在 ADA0CR0 寄存器后，依次存储 ADA0S 寄存器指定的模拟输入引脚的转换信号结果。上述操作完成后，停止转换。

图 15-11. 单次扫描模式操作的时序举例（掉电比较模式时：ADA0S 寄存器 = 03H）



15.6 注意事项

(1) 不使用 A/D 转换器时

不使用 A/D 转换器时，将 ADA0M0.ADA0CE 位清 0，可以降低功耗。

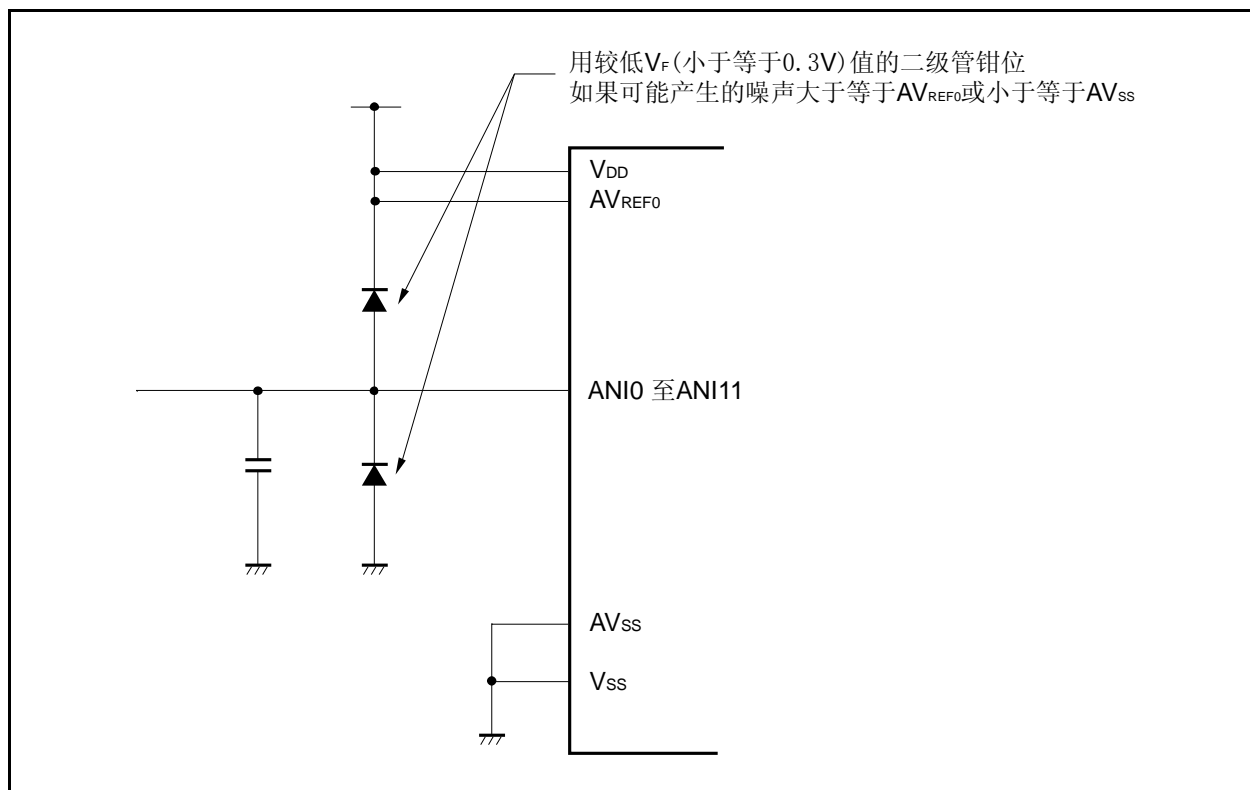
(2) ANI0 至 ANI11 引脚输入范围

向 ANI0 引脚至 ANI11 引脚输入的电压应该在指定范围内。如果大于等于 AV_{REF0} 或小于等于 AV_{SS} （即使在最大额定值范围内）的电压输入到这些引脚中的任意一个引脚，则此通道的转换值将不确定，并且其它通道的转换值也会受到影响。

(3) 应对噪声的措施

为了保证 10 位分辨率，必须有效防止噪声对 ANI0 引脚至 ANI11 引脚的干扰。噪声的影响随着模拟输入源输出阻抗的增大而加强。为了减小噪声，建议连接一个外部电容器，方式如图 15-12 所示。

图 15-12. 模拟输入引脚的处理



(4) 复用 I/O

模拟输入引脚（ANI0 至 ANI11）可以复用为端口引脚。当选择 ANI0 引脚至 ANI11 引脚的其中一个来执行 A/D 转换时，在转换期间不要执行读取输入端口或写入输出端口的指令，因为这样可能会降低转换的分辨率。

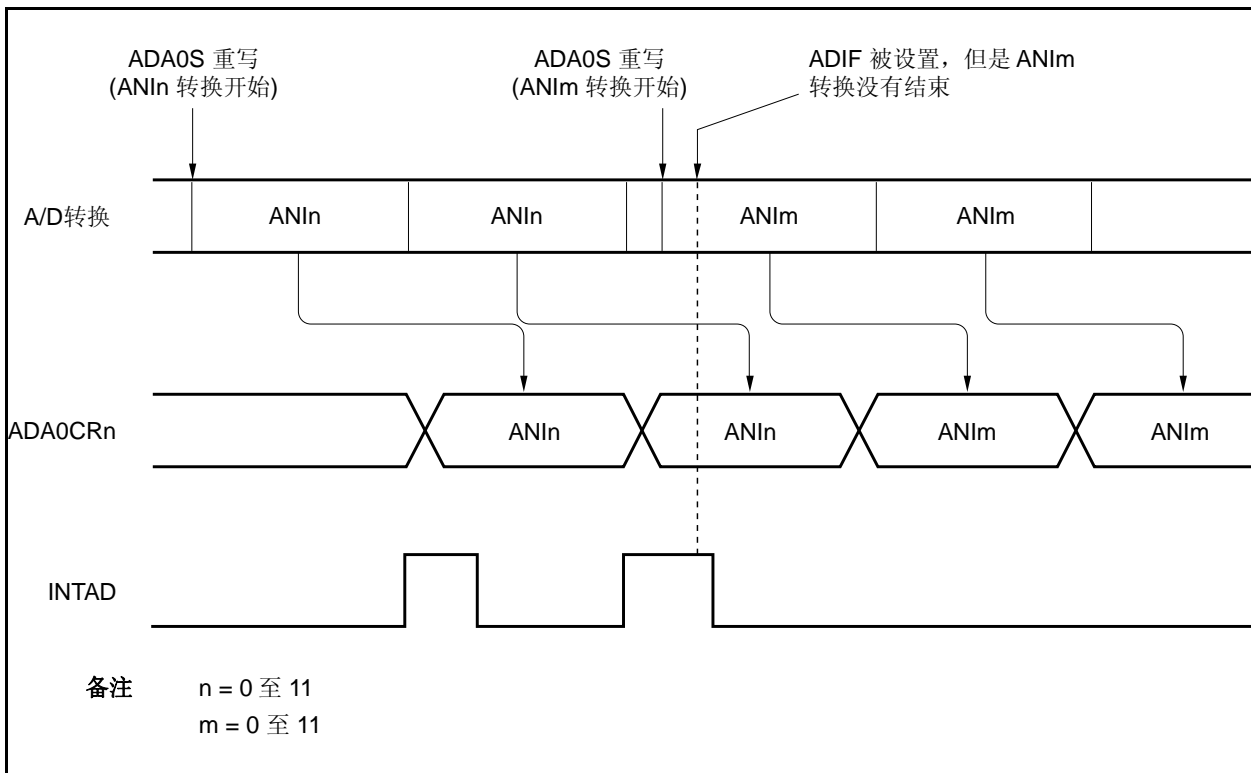
在 A/D 转换期间端口的某些引脚被设置作为输出，由于连接至端口引脚的外部电路干扰使得输出电流波动，输入引脚的转换分辨率也会下降。

如果有数字脉冲施加在与正被转换的输入信号引脚相邻的引脚上，由于耦合噪声的影响，A/D 转换值可能会无法预料。因此，不要在进行 A/D 转换引脚的邻近引脚上施加脉冲。

(5) 中断请求标志 (ADIF)

即使 ADA0S 寄存器的内容改变, 也不会清除中断请求标志 (ADIF)。因此, A/D 转换期间如果改变模拟输入通道, 那么先前所选模拟输入引脚信号的转换结果可能被存储, 且有可能在紧邻 ADA0S 寄存器重写之前, 转换结束中断请求标志被置位。如果写入 ADA0S 寄存器之后立即读取 ADIF 标志, 那么即使新选择的模拟输入引脚的 A/D 转换还没有完成, 也会置位 ADIF 标志。A/D 转换停止时, 重新开始转换之前应该清除 ADIF 标志。

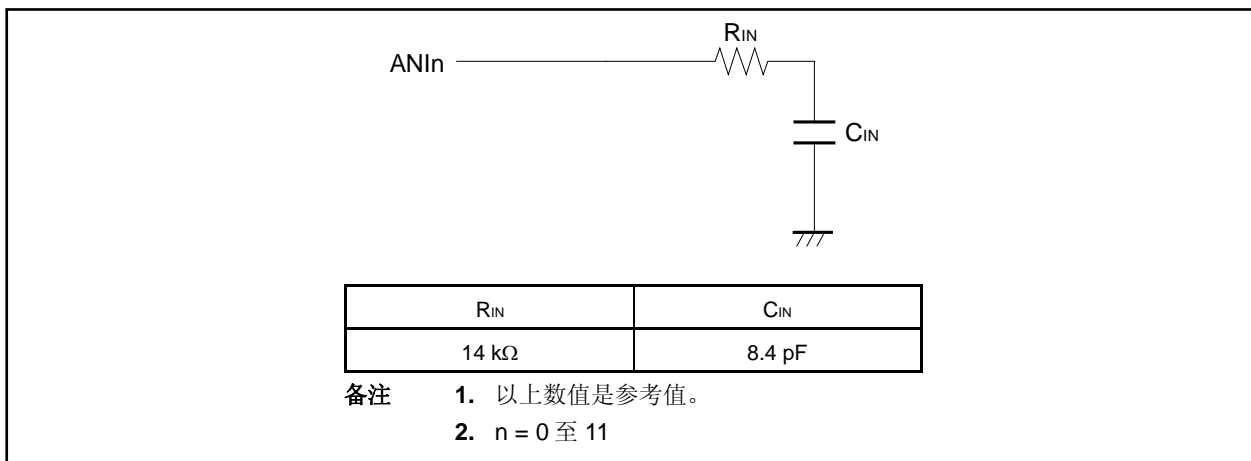
图 15-13. A/D 转换结束中断请求的发生时序



(6) 内部等效电路

以下显示了模拟输入模块的等效电路。

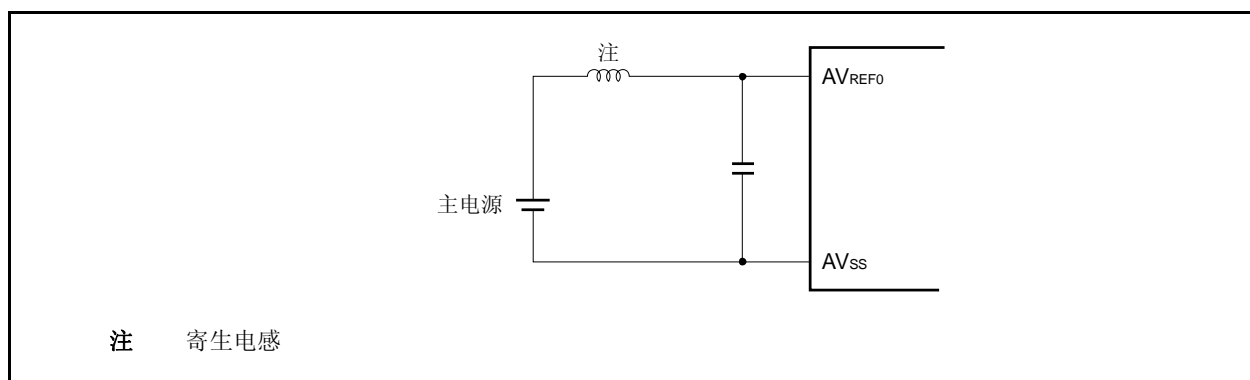
图 15-14. ANIn 引脚的内部等效电路



(7) AVREF0 引脚

- (a) AVREF0 引脚用作 A/D 转换器的电源，也为复用功能端口提供电源。在使用备用电源的应用系统中，确保提供至 AVREF0 引脚的电压与 VDD 相同，如图 15-15 所示。
- (b) AVREF0 引脚用作 A/D 转换器的参考电压引脚。如果 AVREF0 引脚的输入电源具有高阻抗或者电流提供能力较弱，那么转换期间由于电流波动，参考电压可能会波动（尤其紧接在转换操作使能位 ADA0CE 被置为 1 之后）。结果，转换精度可能下降。为了避免这种情况，建议在 AVREF0 引脚和 AVSS 引脚之间连接一个电容器来抑制参考电压的波动，如图 15-15 所示。
- (c) 如果连接至 AVREF0 引脚的电源有具有高直流电阻（例如，加入二极管），由于 A/D 转换电流引起电压下降，所以开始转换时的电压可能低于转换停止时的电压。

图 15-15. AVREF0 引脚处理实例



(8) 读取 ADA0CRn 寄存器

当写入 ADA0M0 至 ADAM2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器时，ADA0CRn 寄存器的内容可能不确定。在转换完成之后且写入 ADA0M0 至 ADAM2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器之前，读取转换结果。当响应外部/定时器触发时，ADA0CRn 寄存器的内容也可能不确定。在转换完成之后且响应下一次外部/定时器触发之前，读取转换结果。不同于上述时序时，可能读取不到正确的转换结果。

(9) 待机模式

在 STOP 模式下，由于 A/D 转换器停止工作，转换结果无效，因此能够降低功耗。STOP 模式解除后，操作重新开始，但是 STOP 模式解除后的 A/D 转换结果无效。STOP 模式解除后，需要使用 A/D 转换器时，在设置 STOP 模式之前或解除 STOP 模式之后，将 ADA0M0.ADA0CE 位清 0。然后在解除 STOP 模式之后，将 ADA0CE 位设为 1。

在 IDLE1、IDLE2 或副时钟操作模式下，操作继续。因此，为了降低功耗，将 ADA0M0.ADA0CE 位清 0。在 IDLE1 和 IDLE2 模式下，由于不能够保持模拟输入引脚的电压值，所以 IDLE1 和 IDLE2 模式解除后，A/D 转换结果无效。设置进入 IDLE1 和 IDLE2 模式之前的转换结果是有效的。

(10) 高速转换模式下

在高速转换模式下，振荡稳定期间，禁止触发信号的输入，也禁止对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 和 ADA0PFT 寄存器执行重写操作。

(11) A/D 转换时间

A/D 转换时间为稳定时间，转换时间，等待时间和触发响应时间的总和（关于这些时间的细节，参见表 15-2 正常转换模式（ADA0HS1 位= 0）下转换时间的选择和表 15-3 高速转换模式（ADA0HS1 位 = 1）下转换时间的选择。

在正常转换模式下，A/D 转换期间，如果对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 以及 ADA0PFT 寄存器执行重写操作或者输入触发信号，则重新执行转换。但是，如果稳定时间结束的时序与写入这些寄存器的时序发生冲突，或者稳定时间结束的时序与触发信号输入的时序发生冲突，那么将会插入 64 个时钟的稳定时间。

如果在插入的稳定时间结束时再次发生时序冲突，则仍然插入稳定时间。因此，不要将触发信号输入间隔和控制寄存器写入间隔设置为 64 个时钟或更少。

(12) A/D 转换结果的变化

因为供应电压的波动或噪声的影响，A/D 转换的结果可能会有变化。为了减小变化，在程序中采用抑制措施，例如取 A/D 转换结果平均值。

(13) A/D 转换结果迟滞特性

逐次比较类型 A/D 转换器在内部采样&保持电容器上保持输入电压，然后进行 A/D 转换。完成后，模拟输入电压仍然保持在内部采样保持电容器上。因此，可能发生以下现象：

- 当相同通道被用于 A/D 转换时，如果当前电压比之前的电压高或低，在转换结果受先前值影响的地方，可能出现滞后特性。因此，即使对相同的电压进行转换，结果也可能不同。
- 当切换模拟输入通道时，在转换结果受先前通道值影响的地方，可能出现滞后特性。这是因为 A/D 转换功能只有一个 A/D 转换器。因此，即使对相同的电压进行转换，结果也可能不同。

15.7 A/D转换器特征表的阅读方法

本部分描述了 A/D 转换器的相关术语。

(1) 分辨率

分辨率为能够识别的最低模拟输入电压，也就是说，每位输出数字所占模拟输入电压的百分比，叫作 1 LSB（最低有效位）。1 LSB 对满量程的比率被表示为 %FSR（满标度量程）。%FSR 是百分比表示的可转换模拟输入电压比率，不管分辨率如何，总是可以用下式表示：

$$\begin{aligned} 1\%FSR & \text{（可转换的模拟输入电压的最大值 - 可转换的模拟输入电压的最小值）} / 100 \\ & = (AV_{REF0} - 0) / 100 \\ & = AV_{REF0} / 100 \end{aligned}$$

当分辨率为 10 位时，1 LSB 表示如下：

$$\begin{aligned} 1 \text{ LSB} & = 1/2^{10} = 1/1,024 \\ & = 0.098\%FSR \end{aligned}$$

精确度是由总体误差决定，与分辨率无关。

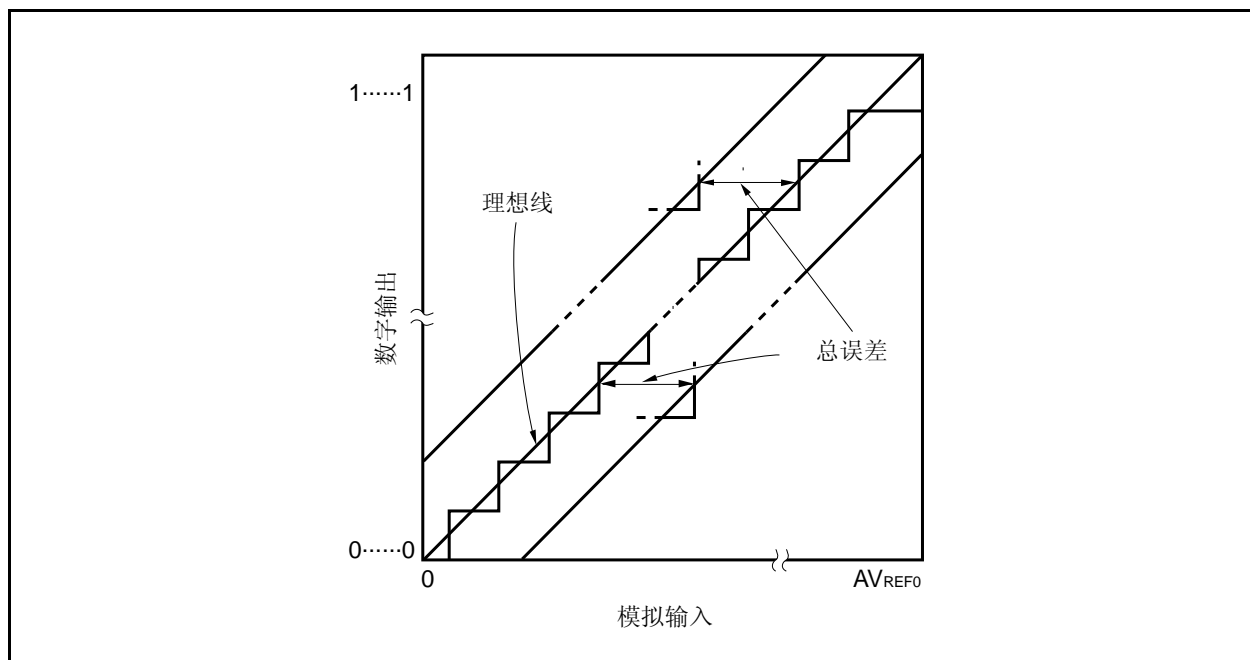
(2) 总体误差

总体误差是实际测量值和理论值之间的最大误差值。

零标度误差、满量程误差、线性误差，这些误差的综合构成总体误差。

在特性表中的总体误差里没有包含量化误差。

图 15-16. 总体误差

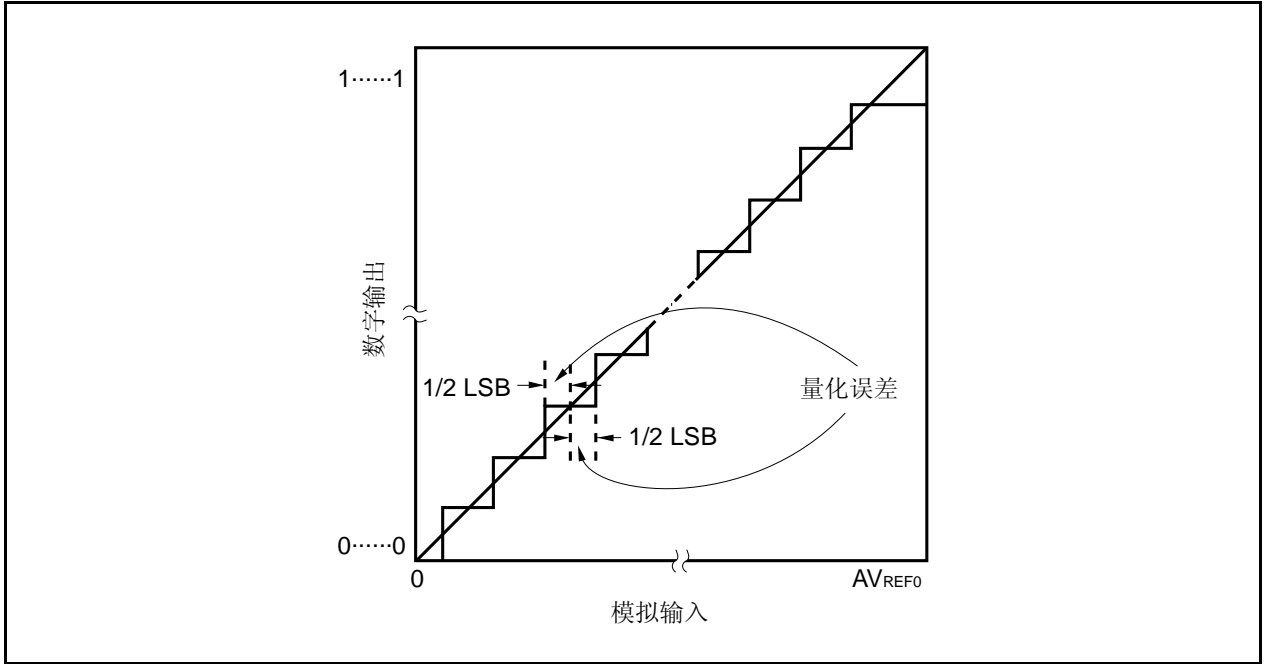


(3) 量化误差

当模拟值转换为数字值时，会不可避免的产生 $\pm 1/2$ LSB 的误差。因为 A/D 转换器将 $\pm 1/2$ LSB 范围内的模拟输入电压转换成相同的数字值，所以量化误差不能避免。

量化误差不包括在参数表中的总体误差、零标度误差、满量程误差、积分线性误差或微分线性误差之中。

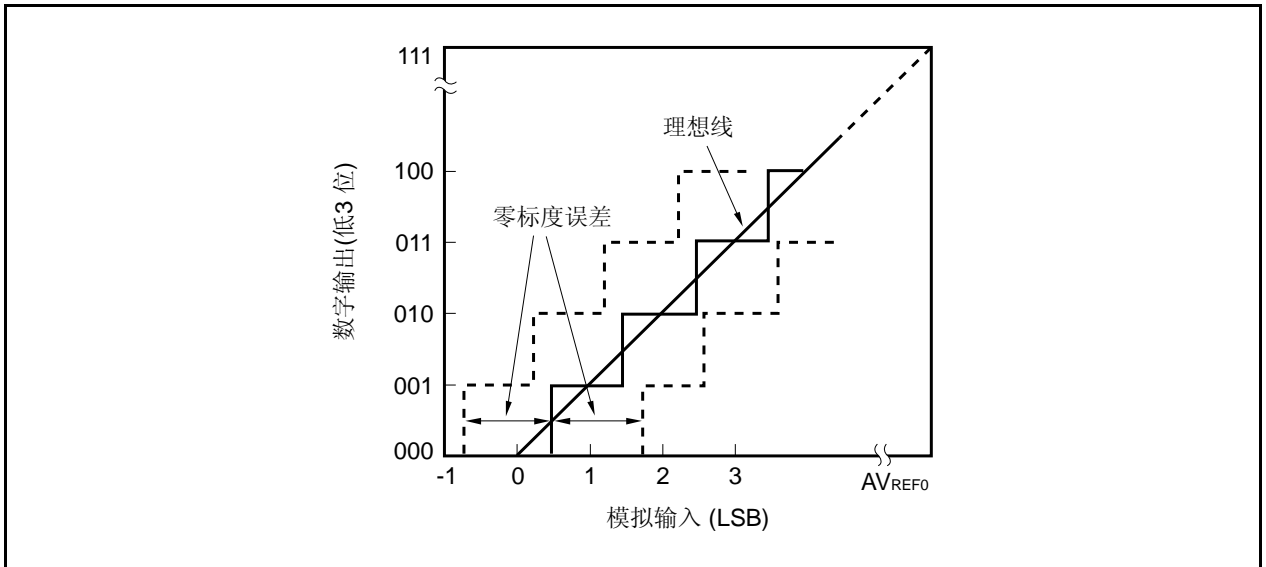
图 15-17. 量化误差



(4) 零标度误差

当数字输出从 0.....000 改变到 0.....001 ($1/2$ LSB) 时，零标度误差就是实际测量的模拟输入电压和理论值之间的差别。

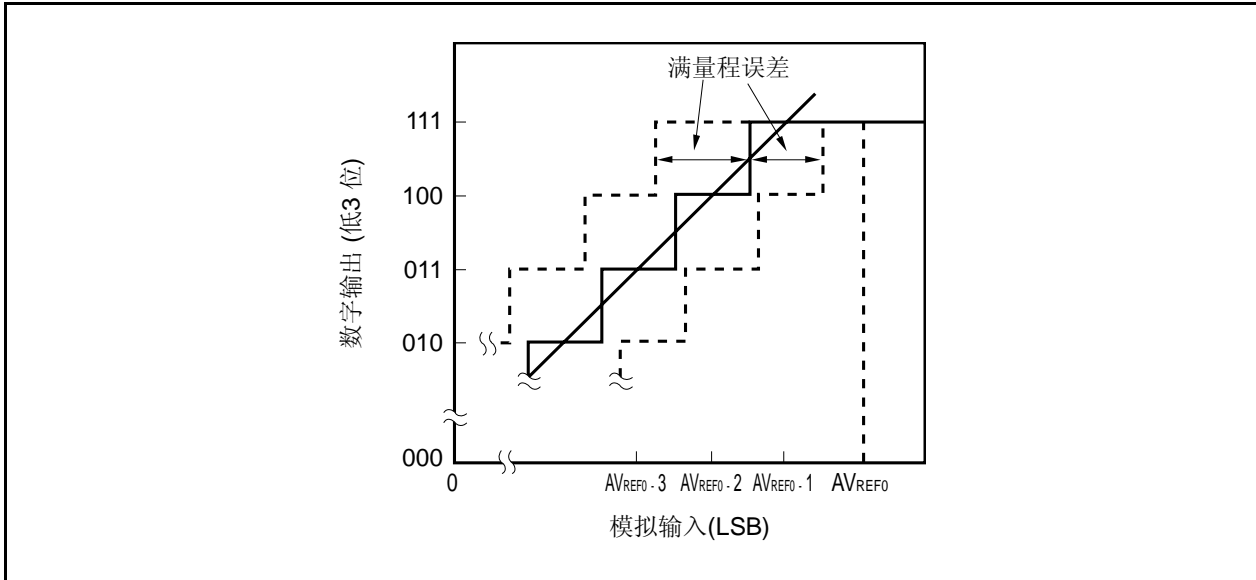
图 15-18. 零标称误差



(5) 满量程误差

当数字输出从 1...110 改变到 1...111 (满量程 - 3/2 LSB) 时, 满量程误差就是实际测量的模拟输入电压与理论值之间的差别。

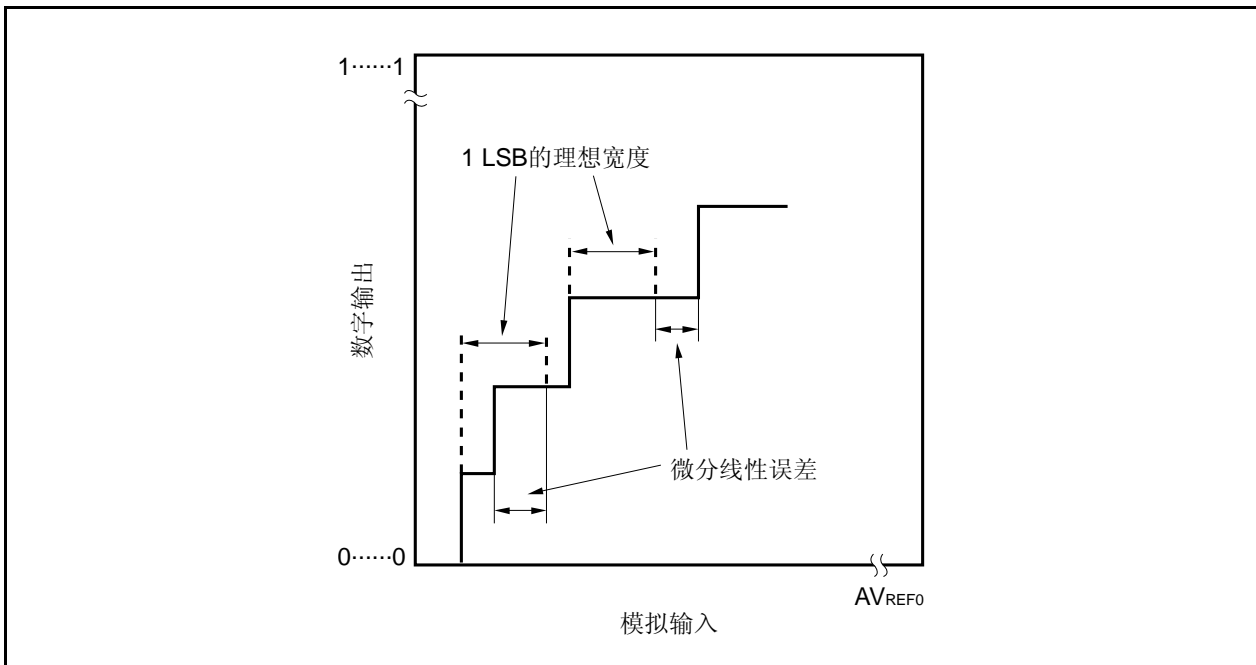
图 15-19. 满量程误差



(6) 微分线性误差

理想状态下, 输出指定编码的宽度是 1 LSB。输出指定编码时, 该误差显示了实际测量值和其理论值之间的差别。输入到相同通道的模拟输入引脚的电压从 AV_{SS} 逐位增加到 AV_{REF0} , 这代表了 A/D 转换的基本特性。当输入电压增加或减小时, 或使用两个或更多的通道时, 参见 15.7 (2) 总体误差。

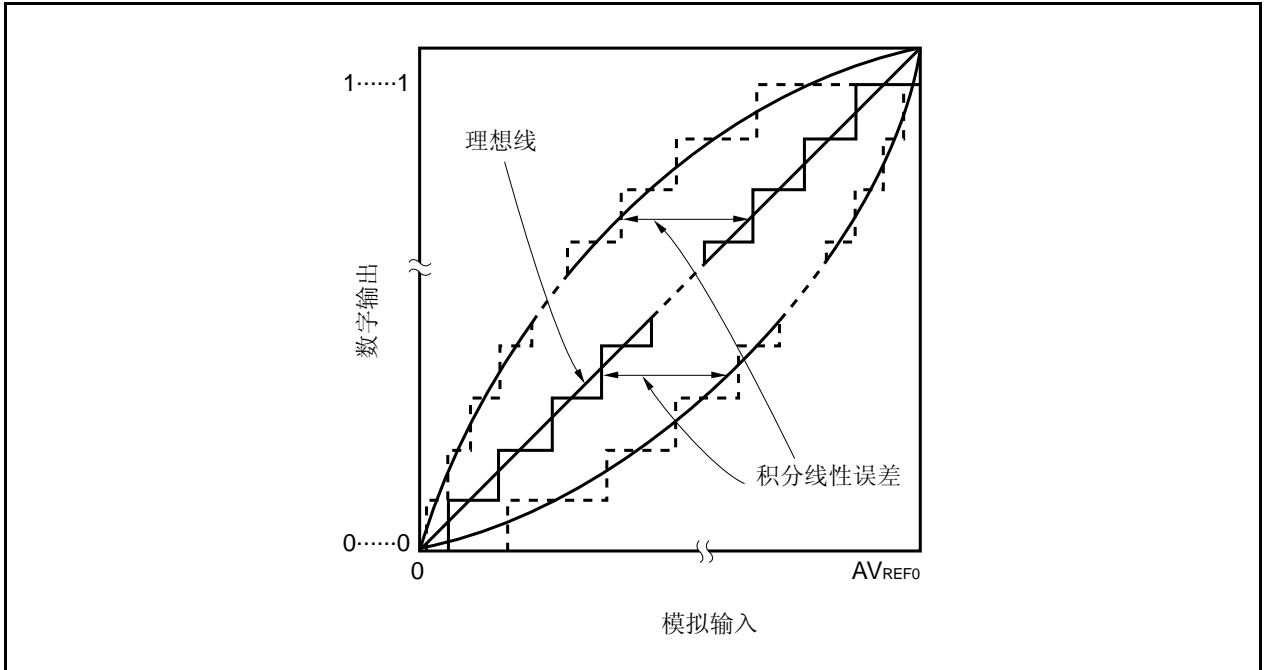
图 15-20. 微分线性误差



(7) 积分线性误差

积分线性误差显示转换特性偏离理想线性关系的程度。它表示当零标度误差和满量程误差均为 0 时，实际测量值和其理论值之间的最大误差。

图 15-21. 积分线性误差

**(8) 转换时间**

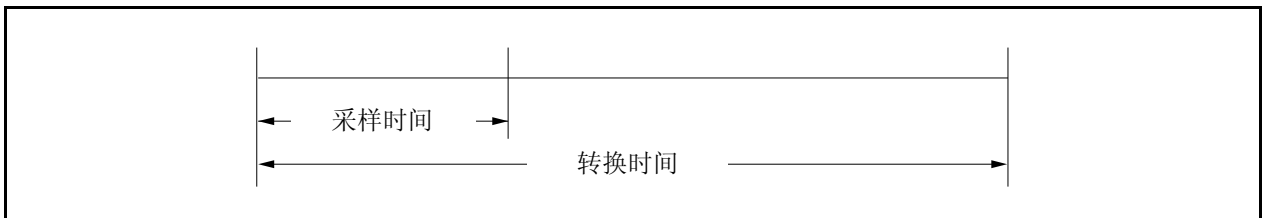
转换时间为触发信号产生后获得数字输出所需的时间。

特性表中，转换时间包含采样时间。

(9) 采样时间

采样时间，即模拟开关开启，将模拟电压载入采样&保持电路的时间。

图 15-22. 采样时间



第十六章 D/A 转换器

16.1 功能

D/A 转换器具有下列功能：

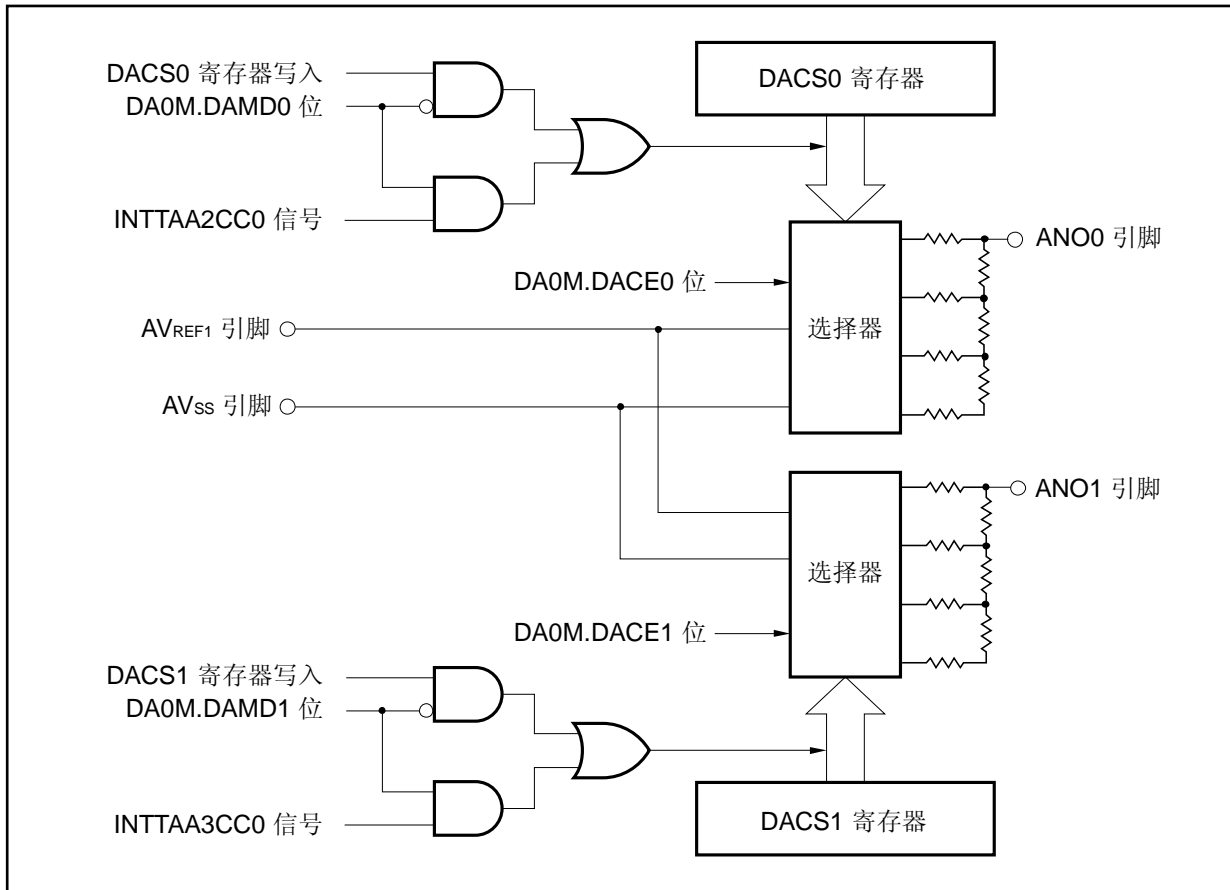
- 8 位精度 × 2 通道 (DA0CS0, DA0CS1)
- R-2R 阶梯变换方法
- 稳定时间：最大 3 μs (AV_{REF1} 为 3.0 至 3.6 V 且外部阻抗 20 pF)
- 模拟输出电压： $AV_{REF1} \times m/256$ ($m = 0$ 至 255; DA0CSn 寄存器的设定值)
- 操作模式：正常模式，实时输出模式

备注 $n = 0, 1$

16.2 配置

D/A 转换器配置如下所示：

图 16-1. D/A 转换器功能框图



注意事项 1. DAC0 和 DAC1 共用 AV_{REF1} 引脚。

2. DAC0 和 DAC1 共用 AV_{SS} 引脚。D/A 转换器和 A/D 转换器共用 AV_{SS} 引脚。

D/A 转换器包括下列硬件：

表 16-1. D/A 转换器的配置

项目	配置
控制寄存器	D/A 转换器模式寄存器 (DA0M) D/A 转换值设置寄存器 0, 1 (DA0CS0, DA0CS1)

16.3 寄存器

控制 D/A 转换器的寄存器如下所示：

- D/A 转换器模式寄存器 (DA0M)
- D/A 转换值设置寄存器 0, 1 (DA0CS0, DA0CS1)

(1) D/A 转换器模式寄存器 (DA0M)

DA0M 寄存器用于控制 D/A 转换器的操作。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

复位后：00H		R/W	地址：FFFFF282H					
	7	6	<5>	<4>	3	2	1	0
DA0M	0	0	DA0CE1	DA0CE0	0	0	DA0MD1	DA0MD0
DA0CEn	D/A 转换器操作使能/禁止的控制 (n = 0, 1)							
0	禁止操作							
1	使能操作							
DA0MDn	D/A 转换器操作模式的选择 (n = 0, 1)							
0	正常模式							
1	实时输出模式 ^注							

注 实时输出模式 (DA0MD0 位 = 1) 的输出触发信号如下：

- 当 n = 0 时：INTTAA2CC0 信号 (参见第七章 16 位定时器/事件计数器 AA (TAA))。
- 当 n = 1 时：INTTAA3CC0 信号 (参见第七章 16 位定时器/事件计数器 AA (TAA))。

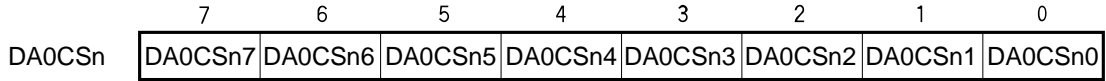
(2) D/A 转换值设置寄存器 0, 1 (DA0CS0, DA0CS1)

DA0CS0 和 DA0CS1 寄存器用于设置输出到 ANO0 和 ANO1 引脚的模拟电压。

这些寄存器可以进行字节读取或写入。

系统复位后, 这些寄存器被设为 00H。

After 复位后: 00H R/W 地址: DA0CS0 FFFF280H, DA0CS1 FFFF281H



注意事项 在实时输出模式 (DA0M.DA0MDn 位 = 1) 下, 在 INTTAA2CC0/INTTAA3CC0 信号产生以前, 设置 DA0CSn 寄存器。当 INTTAA2CC0/INTTAA3CC0 信号产生时, D/A 转换器开始工作。

备注 n = 0, 1

16.4 操作

16.4.1 正常模式下的操作

将写入 DA0CSn 寄存器的操作作为触发信号，执行 D/A 转换操作。

设置方法如下：

- <1> 清除 DA0M.DA0MDn 位为 0（正常模式）。
- <2> 将要输出到 ANOn 引脚的模拟电压值写入 DA0CSn 寄存器。
以上步骤<1>和步骤<2>为初始化设置。
- <3> 设置 DA0M.DA0CEn 位为 1（D/A 转换使能）。
执行该设置时，D/A 转换启动。
- <4> 写入 DA0CSn 寄存器，以进行后续的 D/A 转换。
前一次的 D/A 转换结果会一直保持，直到执行下一次 D/A 转换。

- 备注**
- 1. 关于复用功能引脚的设置，参见表 4-20 当端口引脚用作复用功能时。
 - 2. n = 0, 1

16.4.2 实时输出模式下的操作

使用 TAA2 和 TAA3 的中断请求信号（INTTAA2CC0 和 INTTAA3CC0）作为触发信号，进行 D/A 转换。

设置方法描述如下：

- <1> 设置 DA0M.DA0MDn 位为 1（实时输出模式）。
- <2> 将要输出到 ANOn 引脚的模拟电压值写入 DA0CSn 寄存器。
- <3> 设置 DA0M.DA0CEn 位为 1（D/A 转换使能）。
以上步骤<1>至步骤<3>为初始化设置。
- <4> 操作 TAA2 和 TAA3。
- <5> 当产生 INTTAA2CC0 和 INTTAA3CC0 信号时，D/A 转换器开始工作。
- <6> 此后，在每次产生 INTTAA2CC0 和 INTTAA3CC0 信号时，设置到 DA0CSn 寄存器的值会被输出。

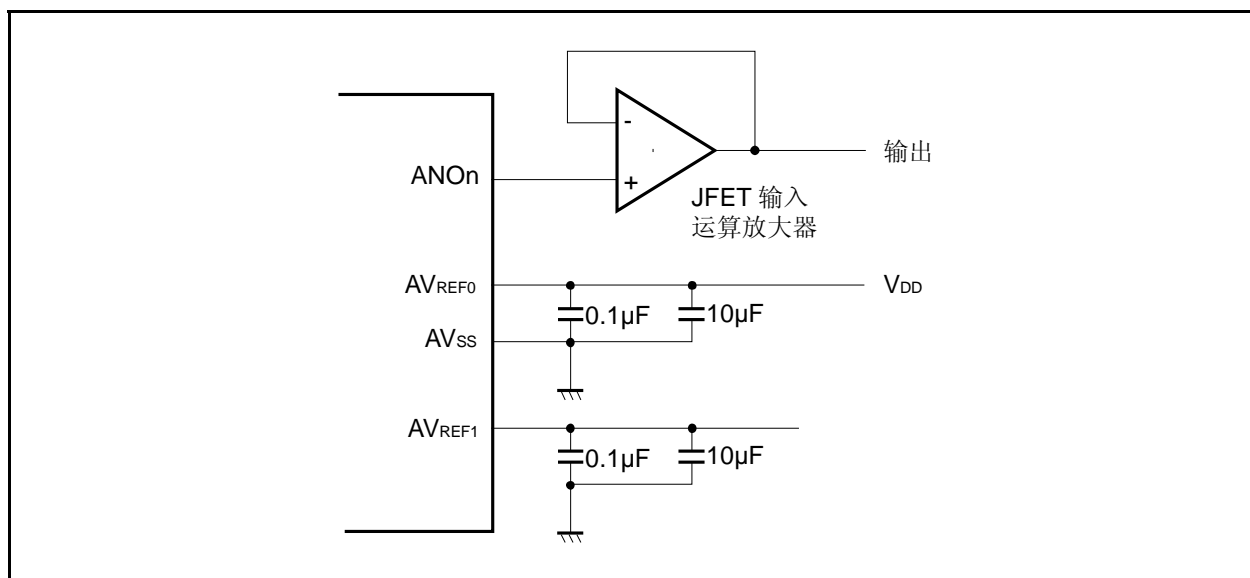
- 备注**
- 1. 步骤<5>之前，ANO0 和 ANO1 引脚的输出值不确定。
 - 2. 关于 ANO0 和 ANO1 引脚在 HALT, IDLE1, IDLE2 和 STOP 模式下的输出值，参见第二十一章 待机功能。
 - 3. 关于复用功能引脚的设置，参见表 4-20 当端口引脚用作复用功能时。

16.4.3 注意事项

使用 V850ES/JG3-U 和 V850ES/JH3-U 的 D/A 转换器时，请遵循以下注意事项：

- (1) 实时输出模式下，触发信号发生时，不要改变 DA0CSn 寄存器的设定值。
- (2) 改变操作模式前，确保将 DA0M.DA0CEn 位清为 0。
- (3) 当使用 P10/AN00 和 P11/AN01 引脚之一作为 I/O 端口，而另一个用作 D/A 输出引脚。在 D/A 输出期间，如果端口 I/O 电平不改变，则在应用系统中可以这样使用。
- (4) 确保 $AV_{REF0} = V_{DD} = AV_{REF1} = 3.0$ 至 3.6 V。如果超出该范围，则操作无法保证。
- (5) 给 AV_{REF1} 上电的时序和给 AV_{REF1} 供电的时序相同。
- (6) 因为 D/A 转换器的输出阻抗高，所以 ANOn 引脚 ($n = 0, 1$) 没有电流输出。如果连接 $2\text{ M}\Omega$ 或更小的电阻，需在电阻和 ANOn 引脚之间插入一个 JFET(结型场效应晶体管)输入运算放大器。

图 16-2. 外部引脚连接举例



- (7) 因为在 STOP 模式下 D/A 转换器停止操作，ANO0 和 ANO1 引脚进入高阻抗状态，因而可以降低功耗。但是，在 IDLE1, IDLE2 或副时钟工作模式下，操作继续。为了降低功耗，将 DA0M.DA0CEn 位清为 0。

第十七章 异步串行接口C (UARTC)

V850ES/JG3-U 和 V850ES/JH3-U 具有 5 路通道 UARTC。

17.1 特性

- 传输速率：300 bps 至 1.25 Mbps（使用 24 MHz 的内部系统时钟和专用波特率发生器）
- 全双工通信：
 - 内部 UARTCn 接收数据寄存器（UCnRX）
 - 内部 UARTCn 发送数据寄存器（UCnTX）
- 2 引脚配置：
 - TXDCn：发送数据输出引脚
 - RXDCn：接收数据输入引脚
- 接收错误检测功能
 - 奇偶错误
 - 帧错误
 - 溢出错误
- 中断源：2 种类型
 - 接收完成中断（INTUCnR）：

在接收使能状态下，串行传输完成后，当接收数据从接收移位寄存器传送到接收数据寄存器时，产生该中断。
 - 发送使能中断（INTUCnT）：

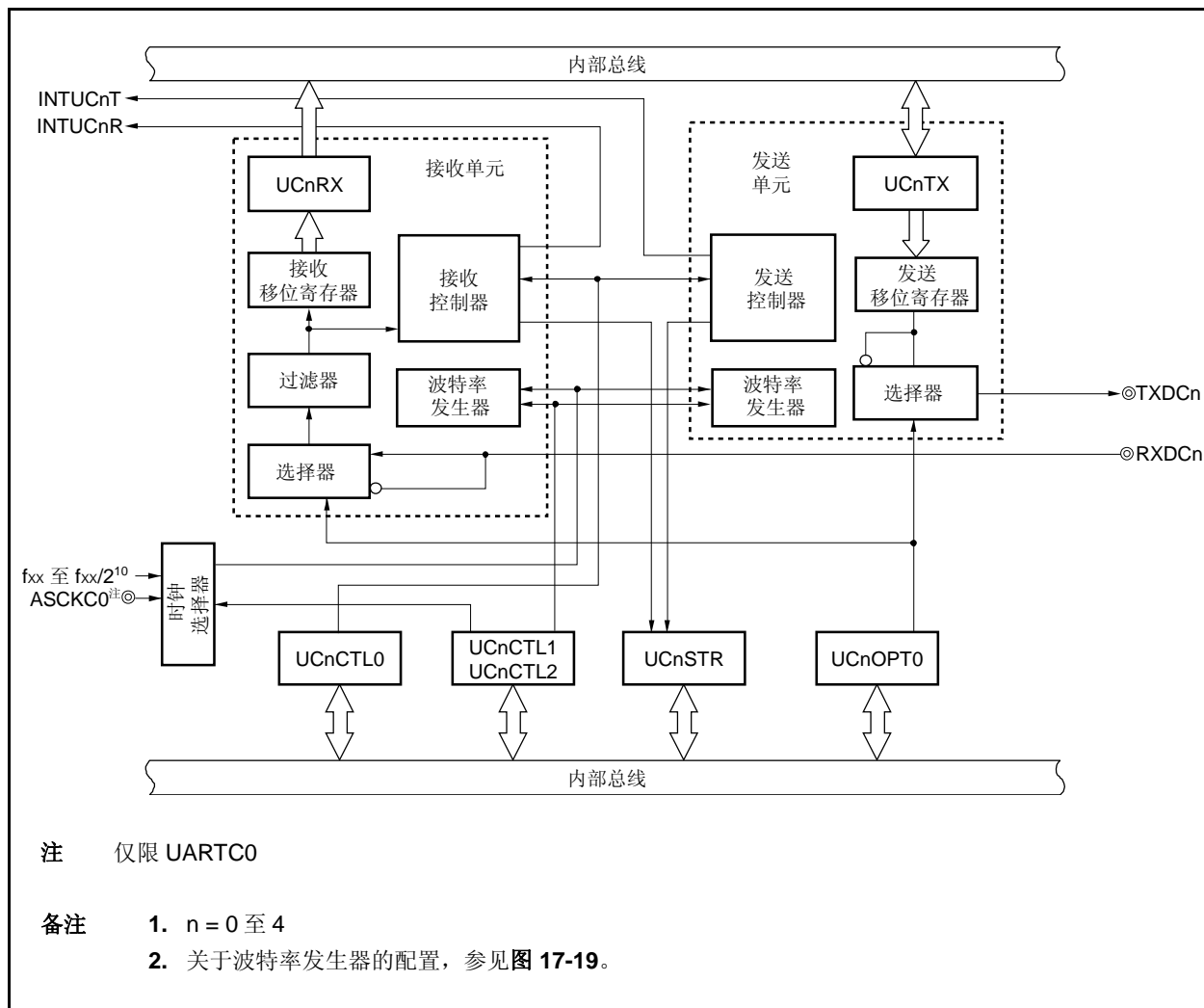
在发送使能状态下，当数据从发送数据寄存器传送到发送移位寄存器时，产生该中断。
- 字符长度：7 至 9 位
- 奇偶校验功能：奇校验、偶校验、0 校验、无校验
- 发送停止位：1 或 2 位
- 片上专用波特率发生器
- 可选择 MSB/LSB 先行传输
- 发送/接收数据可以使输入/输出反相
- 在 LIN（局域网）通信格式下 SBF（同步间隔场）发送
 - SBF 发送时，可选 13 至 20 位
 - SBF 接收时，可以识别 11 位或更多位
 - 提供 SBF 接收标志

备注 n = 0 至 4

17.2 配置

UARTCn 的功能框图如下所示：

图 17-1. 异步串行接口 Cn 的功能框图



UARTCn 包括下列硬件：

表 17-1. UARTCn 的配置

项目	配置
寄存器	UARTCn 控制寄存器 0 (UCnCTL0) UARTCn 控制寄存器 1 (UCnCTL1) UARTCn 控制寄存器 2 (UCnCTL2) UARTCn 选项控制寄存器 0 (UCnOPT0) UARTCn 状态寄存器 (UCnSTR) UARTCn 接收移位寄存器 UARTCn 接收数据寄存器 (UCnRX) UARTCn 发送移位寄存器 UARTCn 发送数据寄存器 (UCnTX)

(1) UARTCn 控制寄存器 0 (UCnCTL0)

UCnCTL0 寄存器是 8 位寄存器，用于指定 UARTCn 的操作。

(2) UARTCn 控制寄存器 1 (UCnCTL1)

UCnCTL1 寄存器是 8 位寄存器，用于选择 UARTCn 的输入时钟。

(3) UARTCn 控制寄存器 2 (UCnCTL2)

UCnCTL2 寄存器是 8 位寄存器，用于控制 UARTCn 的波特率。

(4) UARTCn 选项控制寄存器 0 (UCnOPT0)

UCnOPT0 寄存器是 8 位寄存器，用于控制 UARTCn 的串行传输。

(5) UARTCn 选项控制寄存器 1 (UCnOPT1)

UCnOPT1 寄存器是 8 位寄存器，用于控制 UARTCn 串行传输的 9 位长度。

(6) UARTCn 状态寄存器 (UCnSTR)

UCnSTRn 寄存器由指示接收错误的标志位组成。当发生接收错误时，相应接收错误标志被置位（1）。

(7) UARTCn 接收移位寄存器

本寄存器是移位寄存器，用于将输入至 RXDCn 引脚的串行数据转换为并行数据。接收到 1 字节数据并检测到停止位时，就将接收数据传输至 UCnRX 寄存器。

不能直接操作该寄存器。

(8) UARTCn 接收数据寄存器 (UCnRX)

UCnRX 是 8 位寄存器，用于保存接收数据。当接收到 7 个字符时，最高位清 0 (当数据以 LSB 先行方式接收)。

在接收使能状态下，当 1 帧移入操作完毕时，接收数据同步从 UARTCn 接收移位寄存器传送到 UCnRX 寄存器。

传送到 UCnRX 寄存器也会引发接收完成中断请求信号 (INTUCnR)。

(9) UARTCn 发送移位寄存器

该移位寄存器用于将来自 UCnTX 寄存器的并行数据转换为串行数据。

当一个字节的数据从 UCnTX 寄存器传送过来时，移位寄存器的数据从 TXDCn 引脚输出。

不能直接操作该寄存器。

(10) UARTCn 发送数据寄存器 (UCnTX)

UCnTX 寄存器是 8 位发送数据缓存器。当发送的数据被写入 UCnTX 寄存器后，开始发送。当数据能够写入 UCnTX 寄存器时 (当一帧数据从 UCnTX 寄存器传送到 UARTCn 发送移位寄存器时)，产生发送使能中断请求信号 (INTUCnT)。

17.3 UARTC和其它串行接口的模式转换

17.3.1 CSIF4 和 UARTC0 的模式转换

在 V850ES/JG3-U 和 V850ES/JH3-U 中, CSIF4 和 UARTC0 是相同引脚的复用功能, 因而不能同时使用。使用之前, 用 PMC3 和 PFC3 寄存器预先设置 UARTC0。

注意事项 如果在发送或接收期间切换这些功能, 则 **CSIF4** 和 **UARTC0** 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 17-2. CSIF4 和 UARTC0 模式转换设置

复位后: 00H R/W 地址: FFFFF446H								
PMC3	7	6	5	4	3	2	1	0
	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30
复位后: 00H R/W 地址: FFFFF466H								
PFC3	7	6	5	4	3	2	1	0
	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30
复位后: 00H R/W 地址: FFFFF706H								
PFCE3	7	6	5	4	3	2	1	0
	PFCE37	PFCE36	PFCE35	PFCE34	PFCE33	PFCE32	PFCE31	PFCE30
PMC32	PFCE32	PFC32	操作模式					
0	x	x	Port I/O 模式					
1	0	0	ASCKC0 (UARTC0)					
1	0	1	SCKF4 (CSIF4)					
PMC31	PFCE31	PFC31	操作模式					
0	x	x	Port I/O 模式					
1	0	0	RXDC0 (UARTC)					
1	0	1	SIF4 (CSIF4)					
PMC30	PFCE30	PFC30	操作模式					
0	x	x	Port I/O 模式					
1	0	0	TXDC0 (UARTC)					
1	0	1	SOF4 (CSIF4)					
备注 x = 无需理会								

17.3.2 UARTC1 和 I²C02 的模式转换

在 V850ES/JG3-U 和 V850ES/JH3-U 中, UARTC1 和 I²C02 是相同引脚的复用功能, 因而不能同时使用。使用之前, 用 PMC9, PFC9 和 PFCE9 寄存器预先设置 UARTC1。

注意事项 如果在发送或接收期间切换这些功能, 则 UARTC1 和 I²C02 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 17-3. UARTC1 和 I²C02 模式转换设置

复位后: 0000H R/W 地址: FFFFF452H, FFFFF453H								
PMC9	15	14	13	12	11	10	9	8
	PMC915	PMC914	PMC913	PMC912	PMC911	PMC910	PMC99	PMC98
	7	6	5	4	3	2	1	0
	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90
复位后: 0000H R/W 地址: FFFFF472H, FFFFF473H								
PFC9	15	14	13	12	11	10	9	8
	PFC915	PFC914	PFC913	PFC912	PFC911	PFC910	PFC99	PFC98
	7	6	5	4	3	2	1	0
	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90
复位后: 0000H R/W 地址: FFFFF712H, FFFFF713H								
PFCE9	15	14	13	12	11	10	9	8
	PFCE915	PFCE914	0	0	PFCE911	PFCE910	PFCE99	PFCE98
	7	6	5	4	3	2	1	0
	PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90
PMC91	PFCE91	PFC91	操作模式					
0	×	×	Port I/O 模式					
1	0	1	TXDC1 (UARTC1)					
1	1	0	SDA02 (I ² C02)					
PMC90	PFCE90	PFC90	操作模式					
0	×	×	Port I/O 模式					
1	0	1	RXDC1 (UARTC1)					
1	1	0	SCL02 (I ² C02)					
备注 × = 无须理会								

17.3.3 UARTC2 和 CSIF3 的模式转换

在 V850ES/JG3-U 和 V850ES/JH3-U 中, UARTC2 和 CSIF3 是相同引脚的复用功能, 因而不能同时使用。使用之前, 用 PMC9, PFC9 和 PMCE9 寄存器预先设置 UARTC1。

注意事项 如果在发送或接收期间切换这些功能, 则 UARTC2 和 CSIF3 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 17-4. UARTC2 和 CSIF3 模式转换设置

复位后: 0000H R/W 地址: FFFFF452H, FFFFF453H								
	15	14	13	12	11	10	9	8
PMC9	PMC915	PMC914	PMC913	PMC912	PMC911	PMC910	PMC99	PMC98
	7	6	5	4	3	2	1	0
	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90
复位后: 0000H R/W 地址: FFFFF472H, FFFFF473H								
	15	14	13	12	11	10	9	8
PFC9	PFC915	PFC914	PFC913	PFC912	PFC911	PFC910	PFC99	PFC98
	7	6	5	4	3	2	1	0
	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90
复位后: 0000H R/W 地址: FFFFF712H, FFFFF713H								
	15	14	13	12	11	10	9	8
PFCE9	PFCE915	PFCE914	0	0	PFCE911	PFCE910	PFCE99	PFCE98
	7	6	5	4	3	2	1	0
	PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90
	PMC91	PFCE91	PFC91	操作模式				
	0	x	x	Port I/O 模式				
	1	0	1	TXDC1 (UARTC1)				
	1	1	0	SDA02 (I ² C02)				
	PMC90	PFCE90	PFC90	操作模式				
	0	x	x	Port I/O 模式				
	1	0	1	RXDC1 (UARTC1)				
	1	1	0	SCL02 (I ² C02)				
备注 x = 无须理会								

17.3.4 UARTC3 和 I²C00 的模式转换

在 V850ES/JG3-U 和 V850ES/JH3-U 中, UARTC3 和 I²C00 是相同引脚的复用功能, 因而不能同时使用。使用之前, 用 PMC3, PFC3 和 PFCE3 寄存器预先设置 UARTC3。

注意事项 如果在发送或接收期间切换这些功能, 则 UARTC3 和 I²C00 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 17-5. UARTC3 和 I²C00 模式转换设置

复位后: 00H R/W 地址: FFFFF446H								
	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30
复位后: 00H R/W 地址: FFFFF466H								
	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30
复位后: 00H R/W 地址: FFFFF706H								
	7	6	5	4	3	2	1	0
PFCE3	PFCE37	PFCE36	PFCE35	PFCE34	PFCE33	PFCE32	PFCE31	PFCE30
PMC37	PFCE37	PFC37	操作模式					
0	×	×	Port I/O 模式					
1	0	0	RXDC3 (UARTC3)					
1	0	1	SDA00 (I ² C00)					
1	1	0	CRXD0 (CAN0) ^注					
PMC36	PFCE36	PFC36	操作模式					
0	×	×	Port I/O 模式					
1	0	0	TXDC3 (UARTC3)					
1	0	1	SCL00 (I ² C00)					
1	1	0	CTXD0 (CAN0) ^注					
备注 × = 无须理会								

17.3.5 UARTC4、CSIF0 和 I²C01 的模式转换

在 V850ES/JG3-U 和 V850ES/JH3-U 中，UARTC4、CSIF0 和 I²C01 是相同引脚的复用功能，因而不能同时使用。使用之前，用 PMC4、PFC4 和 PMCE4 寄存器预先设置 UARTC4。

注意事项 如果在发送或接收期间切换这些功能，则 UARTC4、CSIF0 和 I²C01 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 17-6. UARTC4、CSIF0 和 I²C01 模式转换设置

复位后: 00H R/W 地址: FFFFF448H

7	6	5	4	3	2	1	0
0	0	0	0	0	PMC42	PMC41	PMC40

复位后: 00H R/W 地址: FFFFF468H

7	6	5	4	3	2	1	0
0	0	0	0	0	PFC42	PFC41	PFC40

复位后: 00H R/W 地址: FFFFF708H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	PFCE41	PFCE40

PMC41	PFCE41	PFC41	操作模式
0	×	×	Port I/O 模式
1	0	0	SOF0 (CSIF0)
1	0	1	RXDC4 (UARTC4)
1	1	0	SCL01 (I ² C01)

PMC40	PFCE40	PFC40	操作模式
0	×	×	Port I/O 模式
1	0	0	SIF0 (CSIF0)
1	0	1	TXDC4 (UARTC4)
1	1	0	SDA01 (I ² C01)

备注 × = 无须理会

17.4 寄存器

(1) UARTCn 控制寄存器 0 (UCnCTL0)

UCnCTL0 寄存器是 8 位寄存器，用于控制 UARTCn 的串行传输操作。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 10H。

(1/2)

复位后: 10H R/W 地址: UC0CTL0 FFFFA00H, UC1CTL0 FFFFA10H,
UC2CTL0 FFFFA20H, UC3CTL0 FFFFA30H,
UC4CTL0 FFFFA40H

	<7>	<6>	<5>	<4>	3	2	1	0
UCnCTL0 (n = 0 至 4)	UCnPWR	UCnTXE	UCnRXE	UCnDIR	UCnPS1	UCnPS0	UCnCL	UCnSL

UCnPWR	UARTCn 操作 控制
0	禁止 UARTCn 操作 (UARTCn 异步复位)
1	允许 UARTCn 操作
UARTCn 操作由UCnPWR 位控制。通过清除UCnPWR位为0将 TXDCn 引脚输出固定为高电平 (如果UCnOPT0.UCnTDL 位 = 1, 则固定为低电平)。	

UCnTXE	发送操作使能
0	禁止发送操作
1	使能发送操作
<ul style="list-style-type: none"> 若要开始发送，设置 UCnPWR 位为 1 然后设置 UCnTXE 位为 1。 若要停止发送，清除 UCnTXE 位为 0 然后清除 UCnPWR 位为 0。 若要初始化发送单元，清除 UCnTXE 位为 0，等待两个周期的基础时钟，然后再次设置 UCnTXE 位为 1。否则无法执行初始化 (关于基础时钟，参见 17.7 (1) (a) 基础时钟)。 	

UCnRXE	接收操作使能
0	禁止接收操作
1	使能接收操作
<ul style="list-style-type: none"> 若要开始接收，设置 UCnPWR 位为 1 然后设置 UCnRXE 位为 1。 若要停止接收，清除 UCnRXE 位为 0 然后清除 UCnPWR 位为 0。 若要初始化接收单元，清除 UCnRXE 位为 0，等待两个周期的基础时钟，然后再次设置 UCnRXE 位为 1。否则无法执行初始化 (关于基础时钟，参见 17.7 (1) (a) 基础时钟)。 	

UCnDIR	传输方向选择
0	MSB先行
1	LSB先行

- 只有 UCnPWR 位 = 0 或 UCnTXE 位 = UCnRXE 位 = 0 时，才可以重写该寄存器。
- 在 LIN 格式下进行发送和接收时，设置 UCnDIR 位为 1。

UCnPS1	UCnPS0	发送期间奇偶校验的选择	接收期间奇偶校验的选择
0	0	无奇偶校验输出	无奇偶校验接收
0	1	0 奇偶校验输出	0 奇偶校验接收
1	0	奇校验输出	奇校验检测
1	1	偶校验输出	偶校验检测

- 只有 UCnPWR 位 = 0 或 UCnTXE 位 = UCnRXE 位 = 0 时，才可以重写该寄存器。
- 如果接收期间选择“0 奇偶校验接收”，则不进行奇偶校验检查。因此不设置 UCnSTR.UCnPE 位。
- 在 LIN 格式下进行发送和接收时，清除 UCnPS1 和 UCnPS0 位为 00。

UCnCL	1 帧的发送/接收数据字符长度的说明
0	7 位
1	8 位

- 只有 UCnPWR 位 = 0 或 UCnTXE 位 = UCnRXE 位 = 0 时，才可以重写该寄存器。
- 在 LIN 格式下进行发送和接收时，设置 UCnCL 位为 1。

UCnSL	发送数据的停止位长度的说明
0	1 位
1	2 位

只有 UCnPWR 位 = 0 或 UCnTXE 位 = UCnRXE 位 = 0 时，才可以重写该寄存器。

备注 关于奇偶校验的详情，参见 17.6.9 奇偶类型和操作。

(2) UARTCn 控制寄存器 1 (UCnCTL1)

更多详情，参见 17.7 (2) UARTCn 控制寄存器 1 (UCnCTL1)。

(3) UARTCn 控制寄存器 2 (UCnCTL2)

更多详情，参见 17.7 (3) UARTCn 控制寄存器 2 (UCnCTL2)。

(4) UARTCn 选项控制寄存器 0 (UCnOPT0)

UCnOPT0 寄存器是 8 位寄存器，用于控制 UARTCn 寄存器的串行传输操作。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 14H。

(1/2)

复位后: 14H R/W 地址: UC0OPT0 FFFFA03H, UC1OPT0 FFFFA13H,
UC2OPT0 FFFFA23H, UC3OPT0 FFFFA33H,
UC4OPT0 FFFFA43H

	<7>	6	5	4	3	2	1	0
UCnOPT0	UCnSRF	UCnSRT	UCnSTT	UCnSLS2	UCnSLS1	UCnSLS0	UCnTDL	UCnRDL

(n = 0 至 4)

UCnSRF	SBF接收标志
0	UCnCTL0.UCnPWR 位 = UCnCTL0.UCnRXE 位 = 0 时。 或在SBF 接收正常结束时。
1	SBF 接收中

- LIN 通信期间，对 SBF (同步间隔场) 的接收进行判断。
- 发生SBF接收错误时，UCnSRF 位保持为1，然后SBF接收重新开始。
- UCnSRF 位是只读位。

UCnSRT	SBF 接收 触发
0	-
1	SBF 接收触发

- LIN 通信期间该位是SBF接收触发位，读出值总为“0”。
- 为进行SBF 接收, 需设置 UCnSRT 位 (为 1) 以使能SBF接收。
- 设置 UCnPWR 位 = UCnRXE 位 = 1后设置UCnSRT 位。

UCnSTT	SBF 发送 触发
0	-
1	SBF 发送触发

- LIN 通信期间该位是SBF发送触发位，读出值总为“0”。
- 置 UCnPWR 位 = UCnTXE 位 = 1后设置UCnSTT 位。

注意事项 SBF 接收 (UCnSRF 位 = 1) 期间，不要设置 UCnSRT 和 UCnSTT 位 (1)。

UCnSLS2	UCnSLS1	UCnSLS0	SBF 发送长度选择
1	0	1	13-位 输出(复位默认值)
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

UCnPWR 位 = 0 或 UCnTXE 位 = 0时，可设置该寄存器。

UCnTDL	发送数据的电平
0	传输数据正常输出
1	传输数据反相输出

- TXDCn 引脚的输出电平可使用UCnTDL 位进行反相。
- UCnPWR 位 = 0 或UCnTXE 位 = 0时，可设置该寄存器。

UCnRDL	接收数据电平
0	传输数据正常输入
1	传输数据反相输入

- RXDCn 引脚的输入电平可使用UCnRDL 位进行反相。
- UCnPWR 位 = 0 或 UCnRXE 位 = 0时，可设置该寄存器。

(5) UARTCn 选项控制寄存器 1 (UCnOPT1)

UCnOPT1 寄存器是 8 位寄存器，用于控制 UARTCn 寄存器的串行传输操作。
该寄存器可进行字节读写或按位读写。
系统复位后，该寄存器被设为 00H。

注意事项 当 UARTC 的操作被禁止时，设置 UCnEBE 位(UCnCTL0.UCnPWR = 0)。

复位后：00H R/W 地址： UC0OPT1 FFFFA0AH, UC1OPT1 FFFFA1AH,
UC2OPT1 FFFFA2AH, UC3OPT1 FFFFA3AH,
UC4OPT1 FFFFA4AH

	7	6	5	4	3	2	1	0	
UCnOPT1 (n = 0 至 4)	0	0	0	0	0	0	0	0	UCnEBE

UCnEBE	扩展位使能/禁止
0	禁止操作扩展位。按照UCnCTL0.UCnCL位设置的数据长度执行发送/接收。
1	使能操作扩展位。按照9 位字符长度执行发送/接收。

- 当设置 UCnEBE 位为1，且按照9 位长度发送时，请确保设置如下。如果未执行该设置，则 UCnEBE 位的设置无效。
- UCnCTL0.UCnPS1, UCnPS0 = 00 (无校验)
- CnCTL0.UCnCL = 1 (8 位字符长度)
- 如果按照LIN通信格式进行发送或接收，设置 UCnEBE 为 0。

下表展示了寄存器设置值和数据格式之间的关系。

表 17-2. 寄存器设置和数据格式之间的关系

寄存器设置					数据格式				
UCnCTL0				UCnOPT1	D0 至 D6	D7	D8	D9	D10
UCnCL	UCnPS1	UCnPS0	UCnSL	UCnEBE					
0	0	0	0	0	Data	Stop	-	-	-
0	除 00 之外				Data	Parity	Stop	-	-
1	0	0			Data	Data	Stop	-	-
1	除 00 之外				Data	Data	Parity	Stop	-
0	0	0	1	0	Data	Stop	Stop	-	-
0	除 00 之外				Data	Parity	Stop	Stop	-
1	0	0			Data	Data	Stop	Stop	-
1	除 00 之外				Data	Data	Parity	Stop	Stop
0	0	0	0	1	Data	Stop	-	-	-
0	除 00 之外				Data	Parity	Stop	-	-
1	0	0			Data	Data	Data	Stop	-
1	除 00 之外				Data	Data	Parity	Stop	-
0	0	0	1	1	Data	Stop	Stop	-	-
0	除 00 之外				Data	Parity	Stop	Stop	-
1	0	0			Data	Data	Data	Stop	Stop
1	除 00 之外				Data	Data	Parity	Stop	Stop

备注 Data: 数据位
 Stop: 停止位
 Parity: 奇偶校验位

(6) UARTCn 状态寄存器 (UCnSTR)

UCnSTR 寄存器是 8 位寄存器，显示 UARTCn 的传输状态和接收错误的内容。

该寄存器可进行字节读写或按位读写，但是，UCnTSF 位为只读位，而 UCnPE、UCnFE 和 UCnOVE 位既可读也可写。但是，这些位只能写入 0 来清除，而不能通过写 1 来置位(即便写入 1，其值仍保持不变)。

初始化条件如下：

寄存器/位	初始条件
UCnSTR 寄存器	<ul style="list-style-type: none"> • 清零 • UCnCTL0.UCnPWR = 0
UCnTSF 位	<ul style="list-style-type: none"> • UCnCTL0.UCnTXE = 0
UCnPE 位、UCnFE 位、UCnOVE 位	<ul style="list-style-type: none"> • 写入 0 • UCnCTL0.UCnRXE = 0

复位后: 00H R/W 地址: UC0STR FFFFFFFA04H, UC1STR FFFFFFFA14H,
UC2STR FFFFFFFA24H, UC3STR FFFFFFFA34H,
UC4STR FFFFFFFA44H

	<7>	6	5	4	3	<2>	<1>	<0>
UCnSTR (n = 0 至 4)	UCnTSF	0	0	0	0	UCnPE	UCnFE	UCnOVE

UCnTSF	传输状态标志
0	<ul style="list-style-type: none"> 已经设置UCnPWR位 = 0 或 UCnRXE = 0时。 传输完成之后, UCnTX寄存器没有发来下一个数据。
1	写入 UCnTX 寄存器
<p>进行连续发送时, UCnTSF位总是为1。 初始化发送单元时, 在进行初始化前确保UCnTSF位= 0。 如果在UCnTSF位= 1时执行初始化, 则发送数据无法保证。</p>	

UCnPE	奇偶校验 错误标志
0	<ul style="list-style-type: none"> 已经设置UCnPWR位 = 0 或 UCnRXE = 0。 已经写入 0 时。
1	接收期间数据的奇偶位与奇偶校验位不匹配时。
<ul style="list-style-type: none"> 通过设置UCnCTL0.UCnPS1 和 UCnCTL0.UCnPS0 位来控制UCnPE位的操作。 UCnPE位可读可写, 但是只能写入0进行清除, 而不能写入1进行置位, 如果向该位写入1, 其值保持不变。 	

UCnFE	帧错误标志
0	<ul style="list-style-type: none"> 已经设置UCnPWR位 = 0 或 UCnRXE = 0。 已经写入 0 时。
1	接收期间没有检测到停止位。
<ul style="list-style-type: none"> 只检测接收数据停止位的第一位, 而无需考虑UCnCTL0.UCnSL位的值。 UCnFE位可读可写, 但是只能写入0进行清除, 而不能写入1进行置位, 如果向该位写入1, 其值保持不变。 	

UCnOVE	溢出错误标志
0	<ul style="list-style-type: none"> 已经设置UCnPWR位=0 或 UCnRXE=0。 已经写入 0 时。
1	接收数据已经设置到 UCnRX 寄存器 并且在接收数据被读取之前 下一个接收操作已经完成。
<ul style="list-style-type: none"> 发生溢出错误时, 数据被丢弃, 下一个接收数据不会被写入接收缓冲器。 UCnOVE位可读可写, 但是只能写入0进行清除, 而不能写入1进行置位, 如果向该位写入1, 其值保持不变。 	

(7) UARTCn 接收数据寄存器 L (UCnRXL)和 UARTCn 接收数据寄存器(UCnRX)

UCnRXL 和 UCnRX 寄存器是 8 位或 9 位缓冲寄存器，用于存储由接收移位寄存器转换而来的并行数据。

1 个字节的接收完成时，将存储于接收移位寄存器的数据传送到 UCnRXL 和 UCnRX 寄存器。

LSB 接收期间，数据长度指定为 7 位时，接收数据传输至 UCnRXL 寄存器的第 6 位至第 0 位，MSB 位总为 0。

MSB 接收期间，将接收数据传输到 UCnRXL 的第 7 位至第 1 位，且 LSB 位总为 0。

发生溢出 (UCnOVE) 错误时，此时的接收数据不会被传送到 UCnRXL 和 UCnRX 寄存器，而是被丢弃。

根据字符长度，访问单元和复位值有所不同。

- 字符长度 7/8 位 (UCnOPT1.UCnEBE = 0)

该寄存器只读，按照 8 位宽度。

复位或清除 UCnCTL0.UCnPWR 位 = 0，该寄存器被置为 FFH。

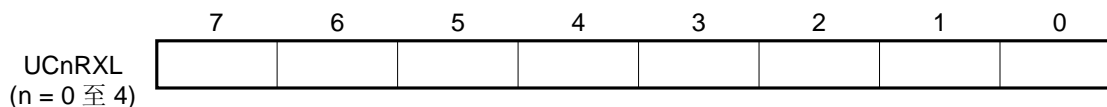
- 字符长度 9 位 (UCnOPT1.UCnEBE = 0)

该寄存器只读，按照 16 位宽度。

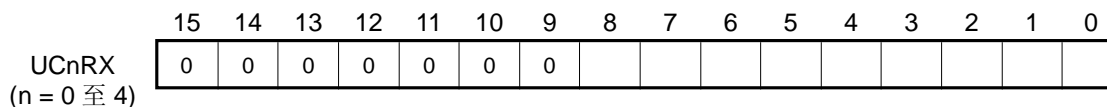
复位或清除 UCnCTL0.UCnPWR 位 = 0，该寄存器被置为 01FFH。

(a) 字符长度 7/8 位(UCnOPT1.UCnEBE = 0)

复位后: FFH R 地址: UC0RXL FFFFFFFA06H, UC1RXL FFFFFFFA16H,
UC2RXL FFFFFFFA26H, UC3RXL FFFFFFFA36H,
UC4RXL FFFFFFFA46H

**(b) 字符长度 9 位 (UCnOPT1.UCnEBE = 1)**

复位后: 01FFH R 地址: UC0RX FFFFFFFA06H, UC1RX FFFFFFFA16H,
UC2RX FFFFFFFA26H, UC3RX FFFFFFFA36H,
UC4RX FFFFFFFA46H



(8) UARTCn 发送数据寄存器 L (UCnTXL)和 UARTCn 发送数据寄存器 (UCnTX)

UCnTXL 和 UCnTX 寄存器是 8 位或 9 位寄存器，用于设置发送数据。

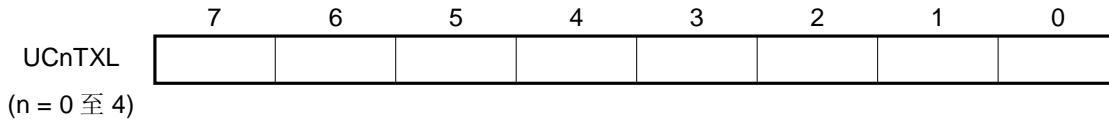
LSB 发送期间，数据长度指定为 7 位时，发送数据传输至 UCnTX 寄存器的第 6 位至第 0 位。MSB 发送期间，将发送数据传输到 UCnTX 的第 7 位至第 1 位。

发生溢出 (UCnOVE) 错误时，此时的接收数据不会被传送到 UCnRXL 和 UCnRX 寄存器，而是被丢弃。根据字符长度，访问单元和复位值有所不同。

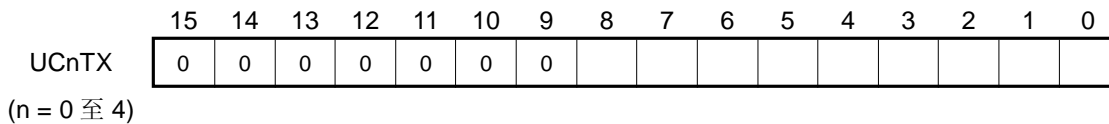
- 字符长度 7/8 位 (UCnOPT1.UCnEBE = 0)
 - 该寄存器按照 8 位宽度进行读取和写入。
 - 复位后该寄存器被置为 FFH。
- 字符长度 9 位 (UCnOPT1.UCnEBE = 1)
 - 该寄存器按照 16 位宽度进行读取和写入。
 - 复位后该寄存器被置为 01FFH。

(a) 字符长度 7/8 位(UCnOPT1.UCnEBE = 0)

复位后: FFH R/W 地址: UC0TXL FFFFFFFA08H, UC1TXL FFFFFFFA18H,
UC2TXL FFFFFFFA28H, UC3TXL FFFFFFFA38H,
UC4TXL FFFFFFFA48H

**(b) 字符长度 9 位 (UCnOPT1.UCnEBE = 1)**

复位后: 01FFH R/W 地址: UC0TX FFFFFFFA08H, UC1TX FFFFFFFA18H,
UC2TX FFFFFFFA28H, UC3TX FFFFFFFA38H,
UC4TX FFFFFFFA48H



17.5 中断请求信号

UARTCn 会产生以下两种中断请求信号：

- 接收完成中断请求信号 (INTUCnR)
- 发送使能中断请求信号 (INTUCnT)

这两种中断请求信号的默认优先级为接收完成中断请求信号优先，然后为发送使能中断请求信号。

表 17-3. 中断及其默认优先级

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 (INTUCnR)

接收使能状态下，数据转移至接收移位寄存器并发送至 UCnRX 寄存器时，接收完成中断请求信号输出。

发生接收错误时，接收完成中断请求信号也会继续输出。因此，响应接收完成中断请求信号并读取数据时，读取 UCnSTR 寄存器并检查接收结果是否错误。

在接收禁止状态下，不产生接收完成中断请求信号。

(2) 发送使能中断请求信号 (INTUCnT)

在发送允许状态下，当发送数据从 UCnTX 寄存器传输至 UARTCn 发送移位寄存器，产生发送使能中断请求信号。

17.6 操作

17.6.1 数据格式

执行全双工串行数据的接收和发送。

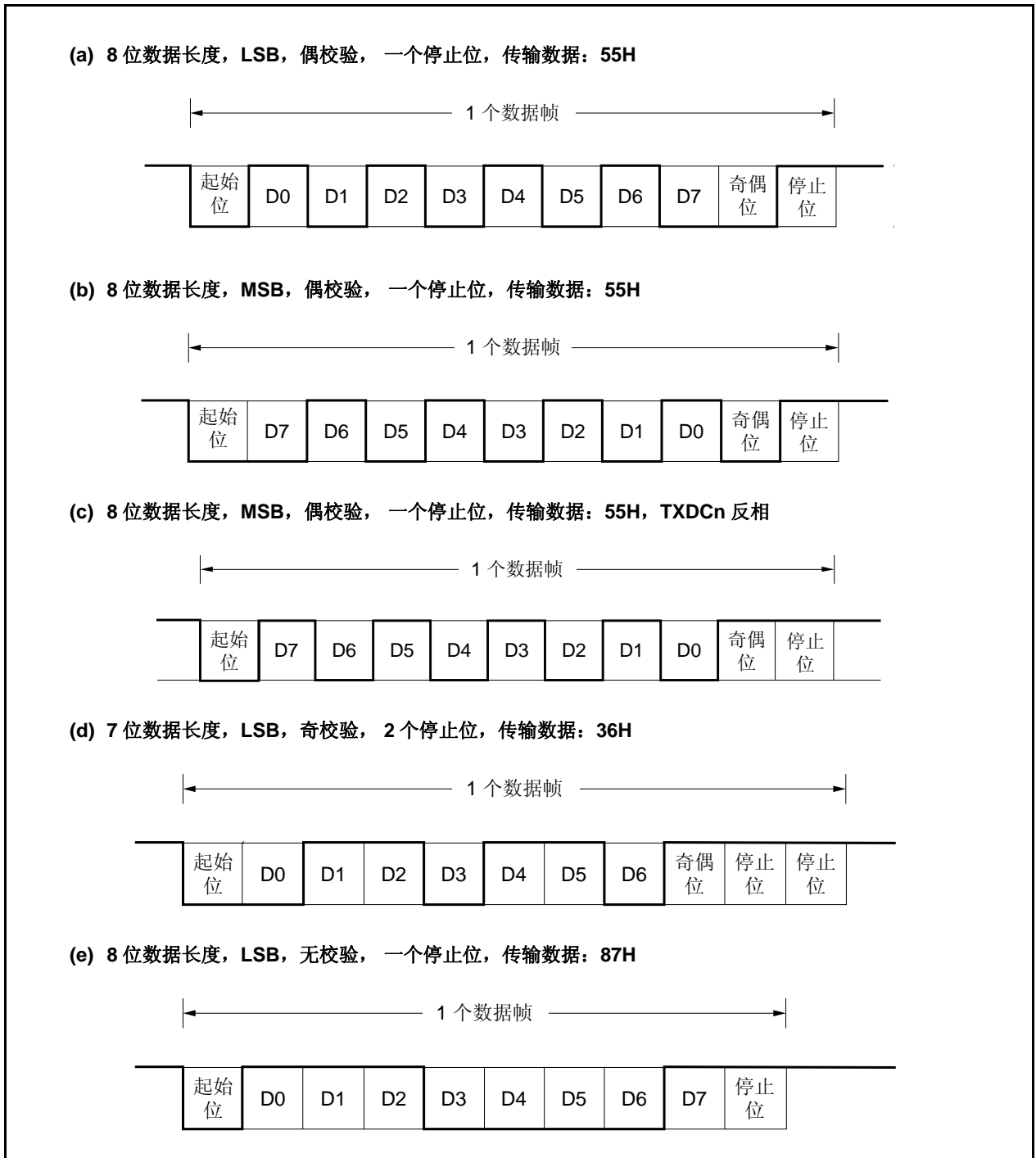
如图 17-7 所示，一个发送/接收数据帧的数据由起始位、字符位、校验位以及停止位组成。

使用 UCnCTL0 寄存器指定一个数据帧内字符位长度、校验段，指定停止位长度，以及 MSB/LSB 传输方式。

另外，UART 中关于 TXDCn 位的输出/反相控制由 UCnOPT0.UCnTDL 位来执行。

- 起始位1 位
- 字符位7 位/8 位
- 奇偶校验位 奇检验/偶校验/0 校验/无校验
- 停止位1 位/2 位

图 17-7. UARTC 发送/接收数据格式



17.6.2 SBF发送/接收格式

V850ES/JG3-U 和 V850ES/JH3-U 具有 SBF (同步间隔场)发送 /接收控制功能，可以支持 LIN 功能的使用。

备注 LIN 表示局域互联网 (Local Interconnect Network)，是一种低速 (1 至 20 kbps) 串行通信协议，有助于降低汽车网络的成本。

LIN 通信为单主机通信，一台主机最多可以连接 15 台从机。

LIN 从机一般是控制开关、传动装置和传感器，且这些装置可通过 LIN 网络连接到 LIN 主机上。

通常，LIN 主机接入诸如 CCN (Controller Area Network) 之类的网络。

另外，LIN 总线使用单线方式且通过遵循 ISO9141 规范的收发器连接至各个节点。

LIN 协议中，主机以某一波特率信息发送一个数据帧，从机将其接收并校正波特率错误。因此，从机波特率错误在±15%范围内，通信可以正常进行。

图 17-8 和 17-9 表明 LIN 的发送和接收的处理。

图 17-8. LIN 发送操作概要

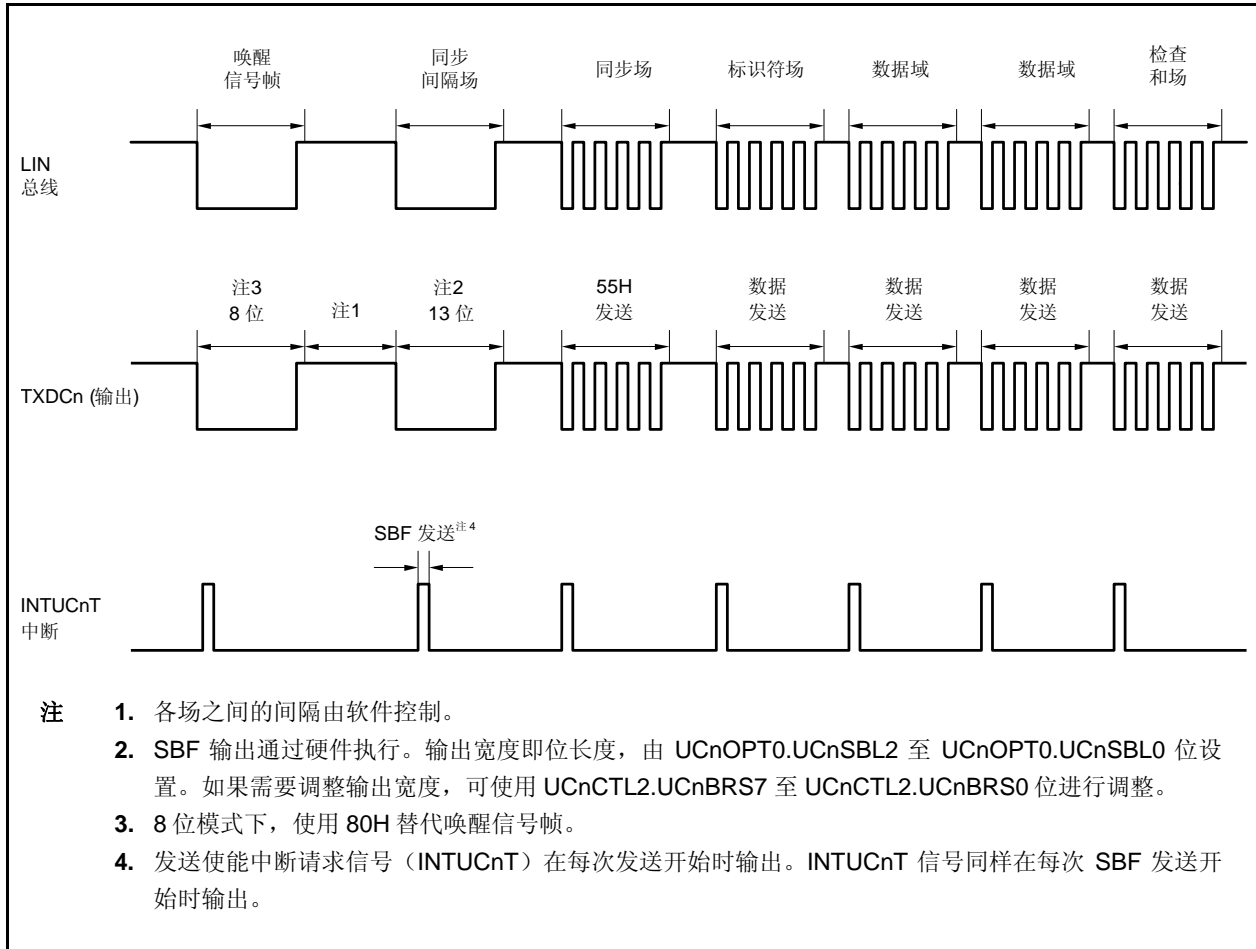
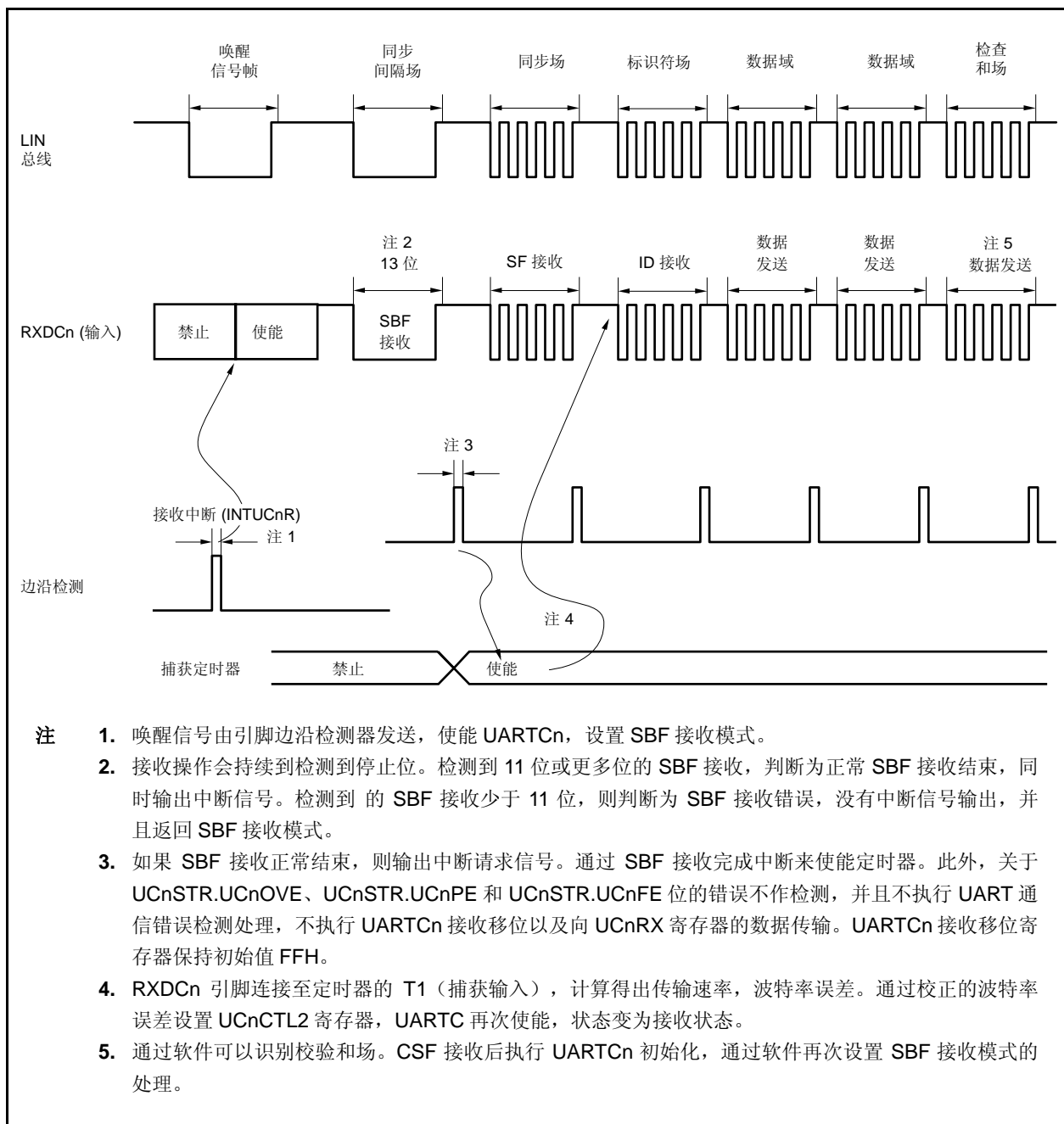


图 17-9. LIN 接收处理概要



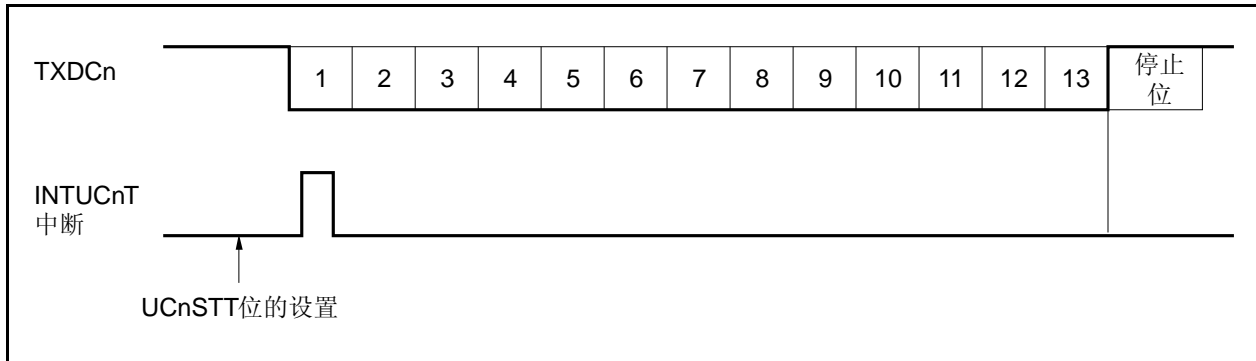
17.6.3 SBF发送

当 UCnCTL0.UCnPWR 位 = UCnCTL0.UCnTXE 位 = 1 时，进入发送使能状态，通过设置 SBF 发送触发信号 (UCnOPT0.UCnSTT 位) 为 1，开始 SBF 发送。

之后，输出由 UCnOPT0.UCnSLS2 至 UCnOPT0.UCnSLS0 位指定的 13 位至 20 位宽度的低电平。SBF 发送开始时，产生发送使能中断请求信号 (INTUCnT)。在 SBF 发送结束后，UCnSTT 位自动清零。之后，恢复 UART 发送模式。

暂停发送，直到下一个待发送数据写入 UCnTX 寄存器，或者直到 SBF 发送触发信号 (UCnSTT 位) 被置位。

图 17-10. SBF 发送



17.6.4 SBF接收

设置 UCnCTL0.UCnPWR 位为 1，然后设置 UCnCTL0.UCnRXE 位为 1，进入接收使能状态。

将 SBF 接收触发 (UCnOPT0.UCnSTR 位) 置 1，来进入 SBF 接收等待状态。

SBF 接收等待状态下，RXDCn 引脚被监控并且开始检测起始位。和 UART 接收等待状态类似。

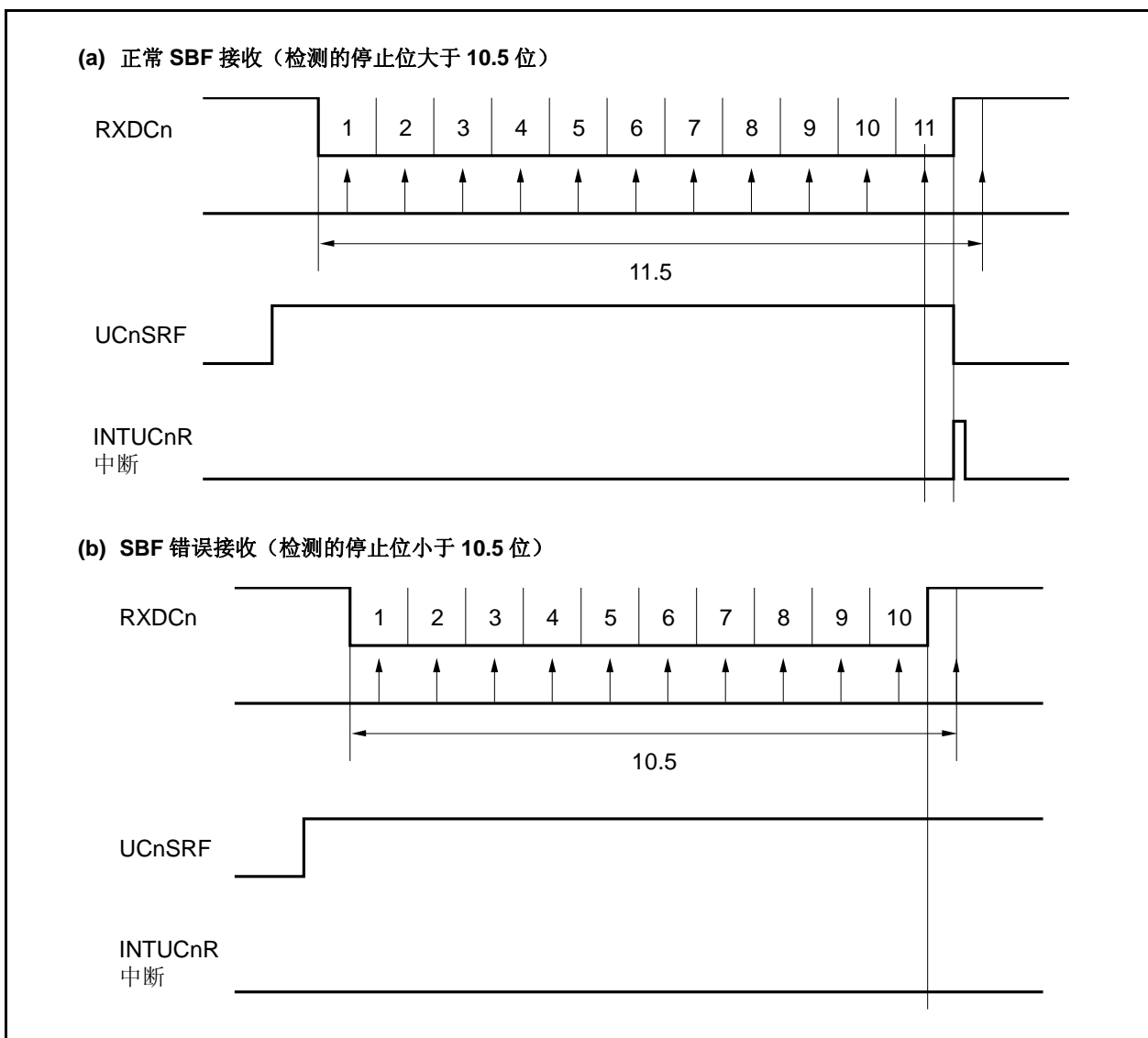
在检测到起始位之后，根据设置的波特率开始接收并且内部计数器开始向上计数。

接收到一个停止位时，如果 SBF 宽度为 11 位或更多位，则判断为正常操作，输出接收完成中断请求信号 (INTUCnR)。UCnOPT0.UCnSRF 位自动清除，SBF 接收结束。对 UCnSTR.UCnOVE、UCnSTR.UCnPE 和 UCnSTR.UCnFE 位的错误不作检测，不执行 UART 通信错误检测处理，不执行 UARTCn 接收移位以及向 UCnRX 寄存器的数据传输，UARTCn 接收移位寄存器保持初始值 FFH。如果 SBF 宽度为 10 位或更少位，则接收终止，作为错误处理且不输出中断，并返回 SBF 接收模式。此时，UCnSRF 位不被清除。

注意事项 1. 如果数据接收期间 SBF 被发送，会发生一个帧错误。

2. SBF 接收 (UCnSRF = 1) 期间，不要将 SBF 接收触发位 (UCnSRT) 和 SBF 发送触发位 (UCnSTT) 设置为 1。

图 17-11. SBF 接收



17.6.5 UART 发送

设置 UCnCTL0.UCnPWR 位为 1，TXDCn 引脚输出高电平。

之后，设置 UCnCTL0.UCnTXE 位为 1，进入发送允许状态，将发送数据写入 UCnTX 寄存器启动发送。起始位、校验位以及停止位自动添加。

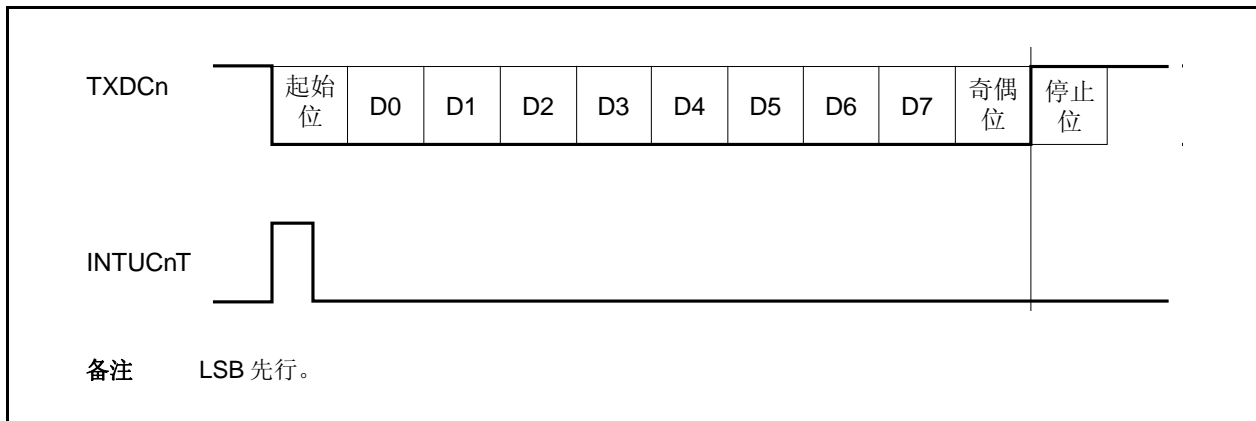
由于 UARTCn 中不提供 CTS (发送允许信号) 输入引脚，所以需要使用一个端口来检查发送目的方已经允许接收。

当发送操作启动之时，UCnTX 寄存器内的数据被传送到 UARTCn 发送移位寄存器。

UCnTX 寄存器的数据发送至 UARTCn 发送移位寄存器的操作完成时，产生发送使能中断请求信号 (INTUCnT)，之后 UARTCn 发送移位寄存器的内容输出至 TXDCn 引脚。

INTUCnT 信号产生后，允许向 UCnTX 寄存器写入下一个发送数据。

图 17-12. UART 发送



17.6.6 连续发送过程

当 UARTCn 发送移位寄存器启动移位操作时，UARTCn 可以写入下一个数据到 UCnTX 寄存器，UARTCn 发送移位寄存器的发送时序可以由发送使能中断请求信号 (INTUCnT) 来判断。

在传输过程中将下一个待发送数据写入 UCnTX 寄存器使得传输效率更高。

注意事项 连续发送执行期间进行发送初始化时，请确保 UCnSTR.UCnTSF 位为 0，然后再执行初始化。UCnTSF 位为 1 时进行初始化，则发送数据无法保证。

图 17-13. 连续发送处理框图

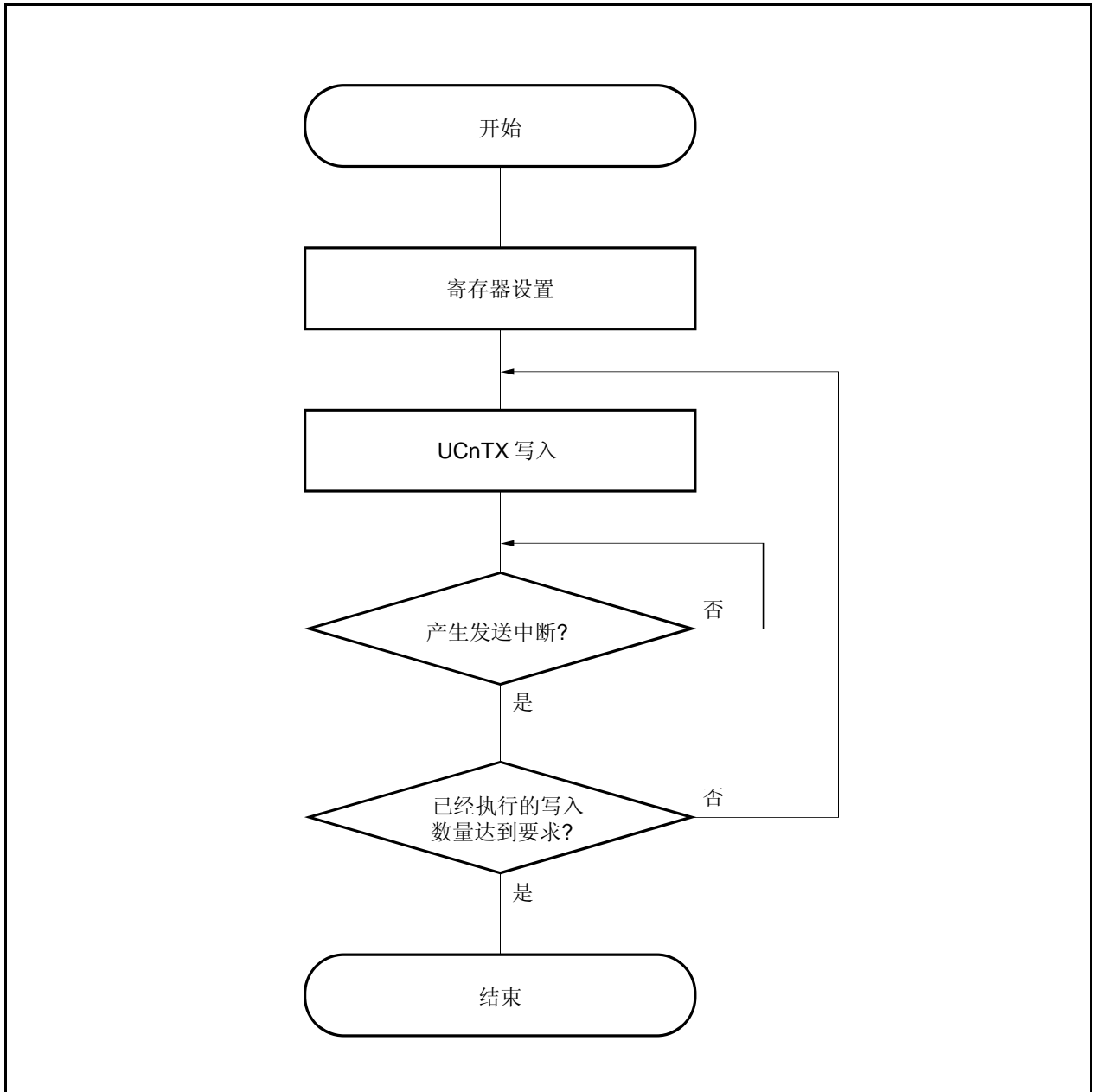
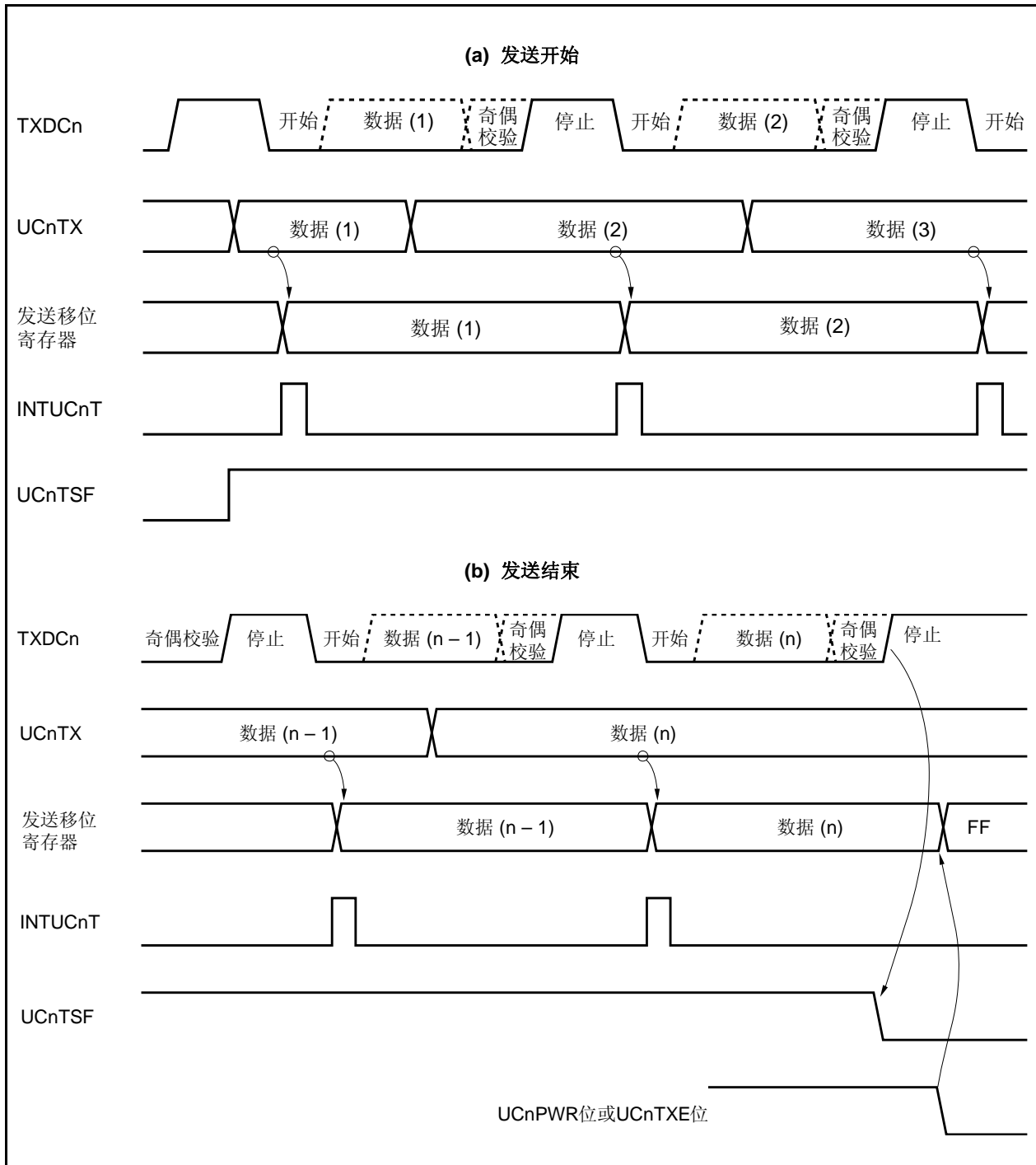


图 17-14. 连续发送操作时序



17.6.7 UART 接收

设置 UCnCTL0.UCnPWR 位为 1，然后设置 UCnCTL0.UCnRXE 位为 1，进入接收等待状态。接收等待状态下，RXDCn 引脚被监测并且进行起始位的检测。

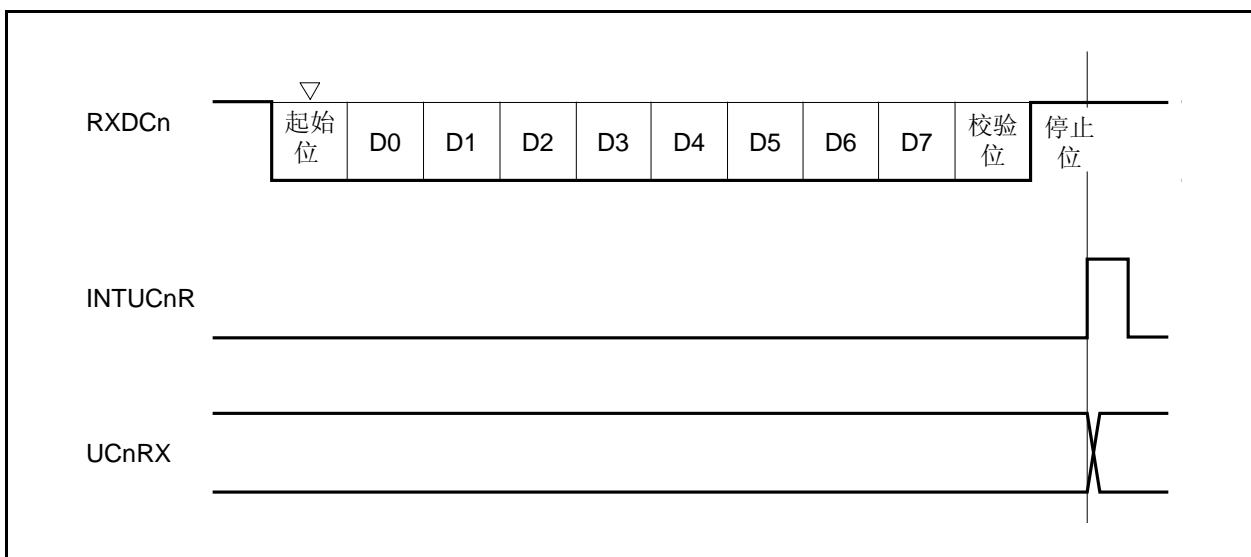
应用两步检测程序来进行起始位的检测。

首先检测到 RXDCn 引脚的上升沿，在下降沿处开始采样。如果 RXDCn 引脚在起始位采样点为低电平，确认是起始位。确认起始位之后，开始接收操作，按照设定的波特率，串行数据被存入 UARTCn 接收移位寄存器。

接收到停止位时，会输出接收完成中断请求信号 (INTUCnR) 时，UARTCn 接收移位寄存器的数据写入 UCnRX 寄存器。但是，如果发生溢出 (UCnSTR.UCnOVE 位) 错误，此时接收到的数据不写入 UCnRX 寄存器，而是被丢弃。

即使接收期间发生奇偶错误 (UCnSTR.UCnPE 位) 或帧错误 (UCnSTR.UCnFE 位)，接收还会继续进行，直到接收到第一个停止位，并在接收完成后输出 INTUCnR 信号。

图 17-15. UART 接收



- 注意事项**
1. 即使发生接收错误，也必须读取 UCnRX 寄存器。如果不读取 UCnRX，则下一个数据接收期间发生溢出错误，并且接收错误会不确定地连续发生。
 2. 认为接收操作期间只有一个停止位。忽略第二个停止位。
 3. 当接收完成时，在产生接收结束中断请求信号 (INTUCnR) 之后读取 UCnRX 寄存器，并对 UCnPWR 位或 UCnRXE 位清 0。如果在 INTUCnR 信号产生之前对 UCnPWR 位或 UCnRXE 位清 0，则无法保证对 UCnRX 寄存器值的正确读取。
 4. 如果 UARTCn 的接收完成处理 (INTUCnR 信号产生) 和 UCnPWR 位 = 0 或 UCnRXE 位 = 0 发生冲突，则即 UCnRX 寄存器内没有存储数据，也可能产生 INTUCnR 信号。
为了完成接收而无需等待 INTUCnR 信号的产生，请确保将中断控制寄存器 (UCnRIC) 的中断屏蔽标志 (UCnRMK) 置位 (1)，清除 UCnPWR 位或 UCnRXE 位 (0)，然后将 UCnRIC 寄存器的中断请求标志 (UCnRIF) 清除 (0)。

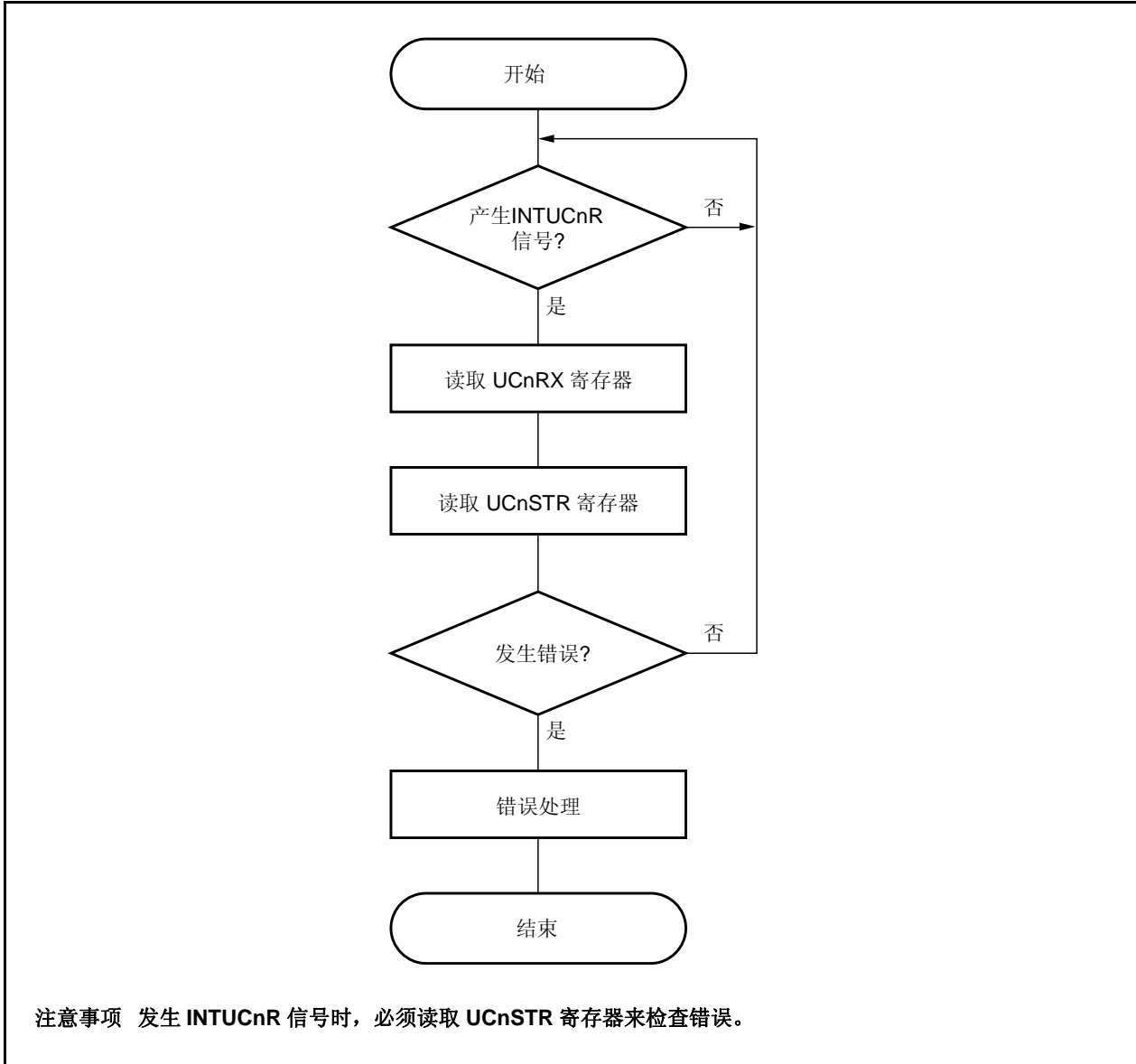
17.6.8 接收错误

接收过程中的错误有 3 种类型：奇偶检验错误、帧错误以及溢出错误。数据接收结果错误标志被设置在 UCnSTR 寄存器中，当错误发生时，输出接收完成中断请求信号 (INTUCnR)。

通过读取 UCnSTR 寄存器的内容，可以确定在接收过程中发生了哪种错误。

读取完成后，写入 0 来清除接收错误标志。

图 17-16. 接收数据读取流程



- 接收错误原因

错误标志	接收错误	原因
UCnPE	奇偶错误	接收到的校验位和设定校验位不匹配
UCnFE	帧错误	未检测到停止位
UCnOVE	溢出错误	在数据从接收缓冲读出之前，下一个数据接收完成

发生接收错误时，根据错误类型执行以下步骤：

- 奇偶错误

如果是因为例如接收线上的噪声问题导致接收到错误数据，则丢弃接收到的数据并重新发送。

- 帧错误

接收端和发送端之间可能发生波特率错误，或者错误的检测到起始位。因为对通信格式来说这是致命错误，确认发送端操作停止，各自进行初始化处理，然后重新开始通信。

- 溢出错误

如果在读出接收数据之前，下一个数据接收完成，1 帧数据会丢弃。如果需要该数据，则进行重新发送。

注意事项 如果在连续接收期间发生接收错误中断，在下一个数据接收完成之前，必须读出 **UCnSTR** 寄存器的内容，然后进行错误处理。

17.6.9 奇偶类型和操作

注意事项 使用 LIN 功能时，将 UCnCTL0.UCnPS1 位和 UCnCTL0.UCnPS0 位固定为 00。

奇偶位用于检测通信数据中的位错误。通常情况下，发送端和接收端应该使用相同的奇偶校验。在使用奇校验和偶校验情况下，可以检测到奇数计数的位错误。而在 0 奇偶校验和无校验情况下，检测不到错误。

(a) 偶校验

(i) 在发送期间

发送数据之中位值为“1”的数量，包括奇偶校验位，控制其数量为偶数。奇偶位的值如下：

- 发送数据中位值为“1”的数量为奇数： 1
- 发送数据中位值为“1”的数量为偶数： 0

(ii) 在接收期间

计算接收数据中位值为“1”的数量，包括校验位，如果为奇数，则输出奇偶校验错误。

(b) 奇数奇偶校验

(i) 在发送期间

与偶校验相对，发送数据中位值为“1”的数量，包括奇偶校验位，控制其数量为奇数。奇偶位的值如下：

- 发送数据中位值为“1”的数量为奇数： 0
- 发送数据中位值为“1”的数量为偶数： 1

(ii) 在接收期间

计算接收数据中位值为“1”的数量，包括校验位，如果为偶数，则输出奇偶校验错误标志。

(c) 0 奇偶校验

在发送期间，奇偶校验位总是“0”，而与发送数据无关。

在接收期间，不执行奇偶校验。因而，无论奇偶位为“0”或“1”，均不会产生奇偶错误。

(d) 无奇偶校验

不向发送数据附加奇偶位。

设定无奇偶位进行接收。因为无奇偶位所以没有奇偶错误发生。

17.6.10 接收数据噪声过滤器

该滤波器使用预分频器输出的基础时钟对 RXDCn 引脚进行采样。

如果读到两次相同的采样值，则匹配检测器输出改变，并且被采样的 RXDCn 信号作为输入数据。所以不超过两个时钟宽度的数据被判定为噪声，不会被送交给内部电路(参见 图 17-18)。有关基础时钟，参见 17.7 (1) (a) 基础时钟。

另外，由于电路如图 17-17 所示，所以，接收操作的处理过程与外部信号状态相比，延迟了 3 个时钟周期。

图 17-17. 噪声过滤电路

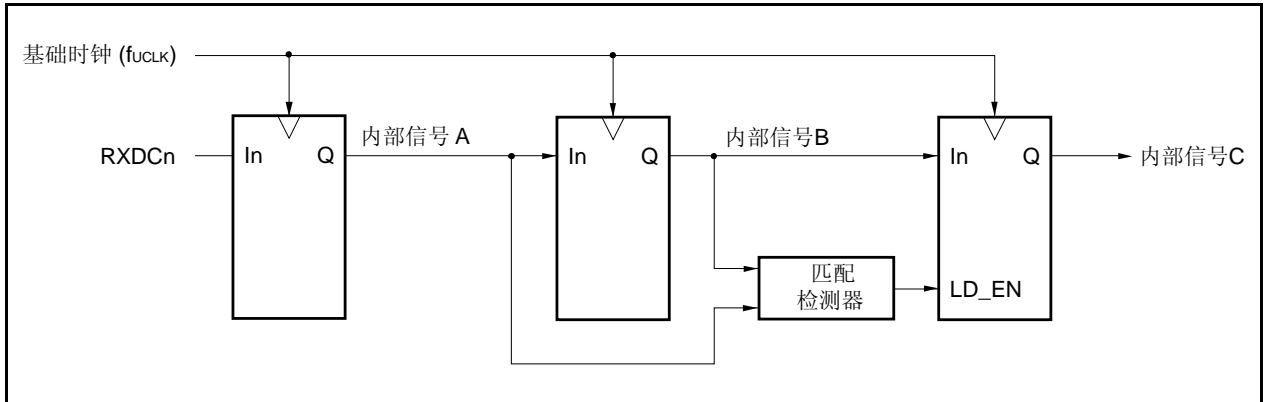
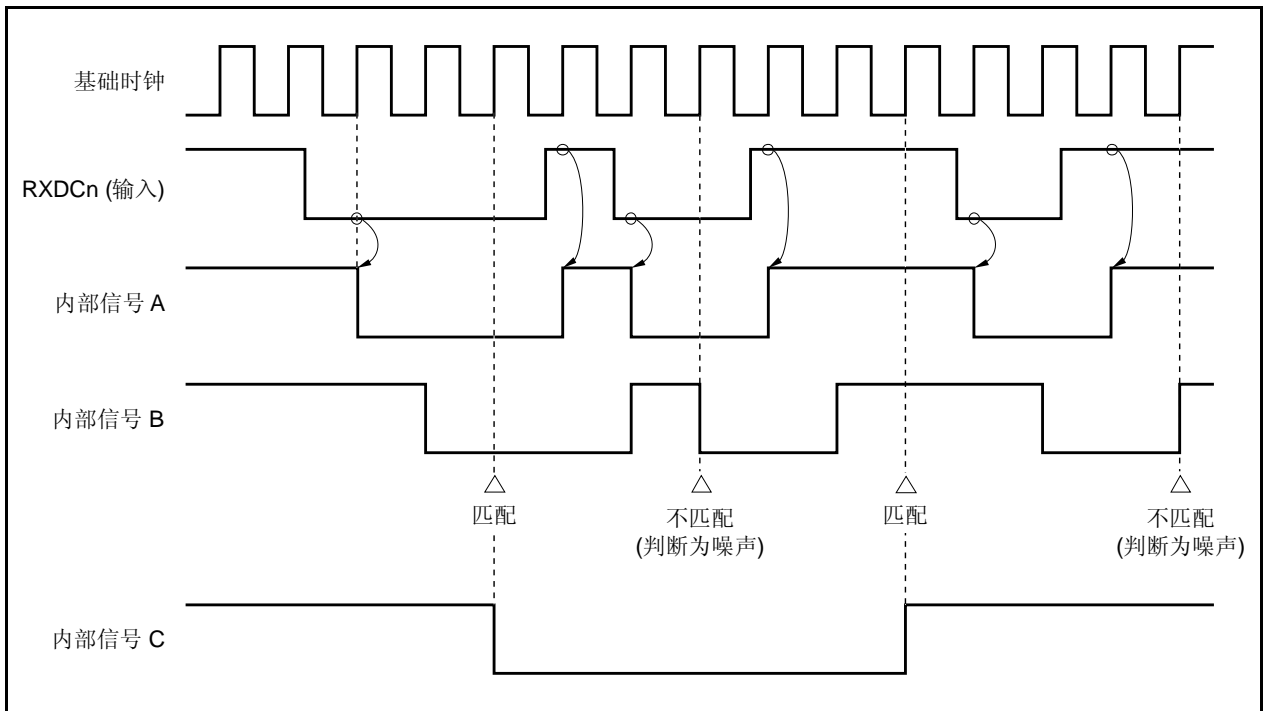


图 17-18. RXDCn 信号判定为噪声的时序



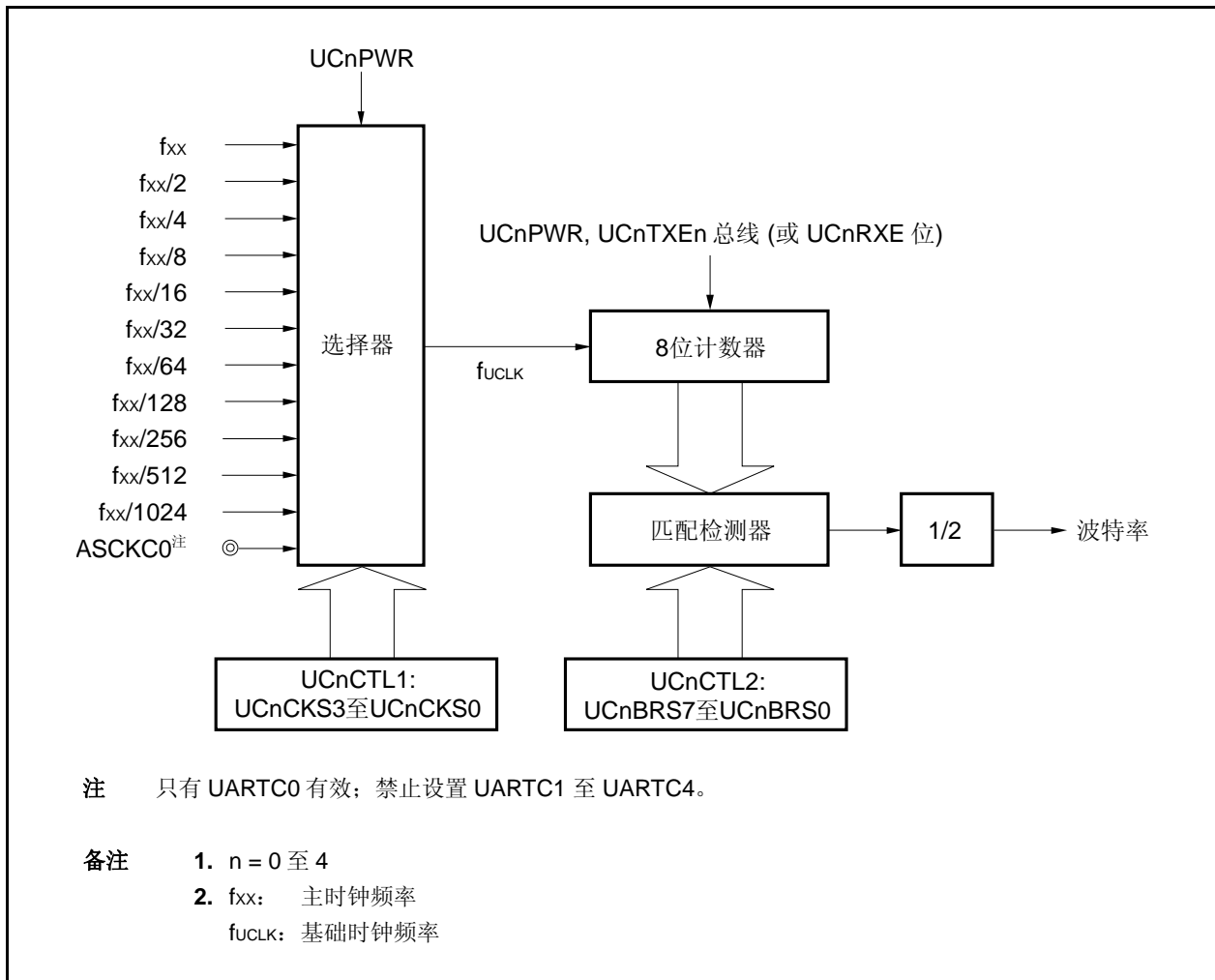
17.7 专用波特率发生器

专用波特率发生器包括一个时钟源选择器和一个 8 位可编程计数器，在 UARTCn 发送/接收期间产生串行时钟。关于串行时钟，对各个通道可以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 17-19. 波特率发生器的配置



(a) 基础时钟

当 UCnCTL0.UCnPWR 位为 1 时，由 UCnCTL1.UCnCKS3 至 UCnCTL1.UCnCKS0 位选择的时钟被提供给 8 位计数器。该时钟称为基础时钟(f_{UCLK})。

(b) 串行时钟的产生

通过设置 UCnCTL1 寄存器和 UCnCTL2 ($n = 0$ 至 4) 寄存器，来产生一个串行时钟。

基础时钟由 UCnCTL1.UCnCKS3 至 UCnCTL1.UCnCKS0 位选择。

8 位计数器的分频值可以使用 UCnCTL2.UCnBRS7 至 UCnCTL2.UCnBRS0 位进行设置。

(3) UARTCn 控制寄存器 2 (UCnCTL2)

UCnCTL2 寄存器是 8 位寄存器，用于选择 UARTCn 的波特率(串行传输速度)。

该寄存器可进行字节读取或写入。

系统复位后，该寄存器被设为 FFH。

注意事项 在改写 UCnCTL2 寄存器之前，将 UCnCTL0.UCnPWR 位清除为 0，或将 UCnTXE 和 UCnRXE 位清除为 0。

复位后: FFH R/W 地址: UC0CTL2 FFFFA02H, UC1CTL2 FFFFA12H,
UC2CTL2 FFFFA22H, UC3CTL2 FFFFA32H,
UC4CTL2 FFFFA42H

7	6	5	4	3	2	1	0
UCnBRS7	UCnBRS6	UCnBRS5	UCnBRS4	UCnBRS3	UCnBRS2	UCnBRS1	UCnBRS0

UCnCTL2 (n = 0 至 4)

UCn BRS7	UCn BRS6	UCn BRS5	UCn BRS4	UCn BRS3	UCn BRS2	UCn BRS1	UCn BRS0	缺省值 (k)	串行 时钟
0	0	0	0	0	0	x	x	x	禁止 设置
0	0	0	0	0	1	0	0	4	f _{UCLK} /4
0	0	0	0	0	1	0	1	5	f _{UCLK} /5
0	0	0	0	0	1	1	0	6	f _{UCLK} /6
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	f _{UCLK} /252
1	1	1	1	1	1	0	1	253	f _{UCLK} /253
1	1	1	1	1	1	1	0	254	f _{UCLK} /254
1	1	1	1	1	1	1	1	255	f _{UCLK} /255

备注 f_{UCLK}: 由 UCnCTL1.UCnCKS3 至 UCnCTL1.UCnCKS0 位选择的
时钟频率

(4) 波特率

波特率误差由下列公式计算。

$$\text{波特率} = \frac{f_{\text{UCLK}}}{2 \times k} [\text{bps}]$$

使用内部时钟时，按以下公式（用 ASCKC0 引脚作为 UARTC0 的时钟时，按上式计算）

$$\text{波特率} = \frac{f_{\text{XX}}}{2^{m+1} \times k} [\text{bps}]$$

备注 f_{UCLK} 是由 UCnCTL1.UCnCKS3 至 UCnCTL1.UCnCKS0 位选择的基础时钟频率
 f_{XX} : 主时钟频率
 m 是由 UCnCTL1.UCnCKS3 至 UCnCTL1.UCnCKS0 位设置的值 ($m = 0$ 至 10)
 k 是由 UCnCTL2.UCnBRS7 至 UCnCTL2.UCnBRS0 位设置的值 ($k = 4$ 至 255)

波特率误差由下列公式计算：

$$\begin{aligned} \text{误差}(\%) &= \left(\frac{\text{实际波特率 (有错误的波特率)}}{\text{目标波特率 (正确的波特率)}} - 1 \right) \times 100 [\%] \\ &= \left(\frac{f_{\text{UCLK}}}{2 \times k \times \text{目标波特率}} - 1 \right) \times 100 [\%] \end{aligned}$$

使用内部时钟时，按以下公式（用 ASCKC0 引脚输入作为 UARTC0 的时钟时，按上式计算波特率误差）

$$\text{误差}(\%) = \left(\frac{f_{\text{XX}}}{2^{m+1} \times k \times \text{目标波特率}} - 1 \right) \times 100 [\%]$$

注意事项 1. 发送过程中，波特率误差必须在接收端的误差容限之内。
 2. 接收时的波特率误差满足一定范围，该范围参见“（5）接收中允许的波特率误差范围”。

要设置波特率，按以下算式计算，设置 UCnCTL1 和 UCnCTL2 寄存器（使用内部时钟时）。

- <1> 设置 k 等于 $f_{xx}/2/(2 \times \text{目标波特率})$ ，m 等于 0。
- <2> 如果 k 等于 256 或更大 ($k \geq 256$)，将 k 减小一半 ($k/2$)，m 加 1 ($m + 1$)。
- <3> 重复第<2>步，直到 k 小于 256 ($k < 256$)。
- <4> 对 k 的第一位小数四舍五入，得到最近的整数。
如果四舍五入后 k 为 256，再次执行第<2>步将 k 设置为 128。
- <5> 将 m 的值设置到 UCnCTL1 寄存器，k 的值设置到 UCnCTL2 寄存器。

示例： 如果 $f_{xx} = 48 \text{ MHz}$ 而目标波特率 = 153,600 bps
 <1> $k = 480,000,000/2/(2 \times 153,600) = 78.125\dots$ ， $m = 0$
 <2>， <3> $k = 78.125\dots < 256$ ， $m = 0$
 <4> 设置 UCnCTL2 寄存器的值： $k = 78 = 4EH$ ，设置 UCnCTL1 寄存器的值： $m = 0$

$$\begin{aligned} \text{实际波特率} &= 48,000,000/2/(2 \times 78) \\ &= 153,846 \text{ [bps]} \end{aligned}$$

$$\begin{aligned} \text{波特率误差} &= \{48,000,000/2/(2 \times 78 \times 153,600) - 1\} \times 100 \\ &= 0.160 \text{ [%]} \end{aligned}$$

以下为波特率设置的典型举例：

表 17-4. 波特率发生器设置数据

波特率 (bps)	$f_{xx} = 48 \text{ MHz}$			$f_{xx} = 32 \text{ MHz}$			$f_{xx} = 24 \text{ MHz}$		
	UCnCTL1	UCnCTL2	ERR (%)	UCnCTL1	UCnCTL2	ERR (%)	UCnCTL1	UCnCTL2	ERR (%)
300	08H	9CH	0.16	07H	D0H	0.16	07H	9CH	-2.3
600	07H	9CH	0.16	06H	D0H	0.16	06H	9CH	0.16
1,200	06H	9CH	0.16	05H	D0H	0.16	05H	9CH	0.16
2,400	05H	9CH	0.16	04H	D0H	0.16	04H	9CH	0.16
4,800	04H	9CH	0.16	03H	D0H	0.16	03H	9CH	0.16
9,600	03H	9CH	0.16	02H	D0H	0.16	02H	9CH	0.16
19,200	02H	9CH	0.16	01H	D0H	0.16	01H	9CH	0.16
31,250	01H	C0H	0.00	01H	80H	0.00	00H	C0H	0.00
38,400	01H	9CH	0.16	00H	D0H	0.16	00H	9CH	0.16
76,800	00H	9CH	0.16	00H	68H	0.16	00H	4EH	0.16
153,600	00H	4EH	0.16	00H	34H	0.16	00H	27H	0.16
312,500	00H	26H	1.05	00H	1AH	-1.54	00H	13H	1.05
625,000	00H	13H	1.05	00H	0DH	-1.54	00H	0AH	-4.00
1,000,000	00H	0CH	0.00	00H	08H	0.00	00H	06H	0.00
1,250,000	00H	0AH	-4.00	禁止设置			00H	05H	-4.00

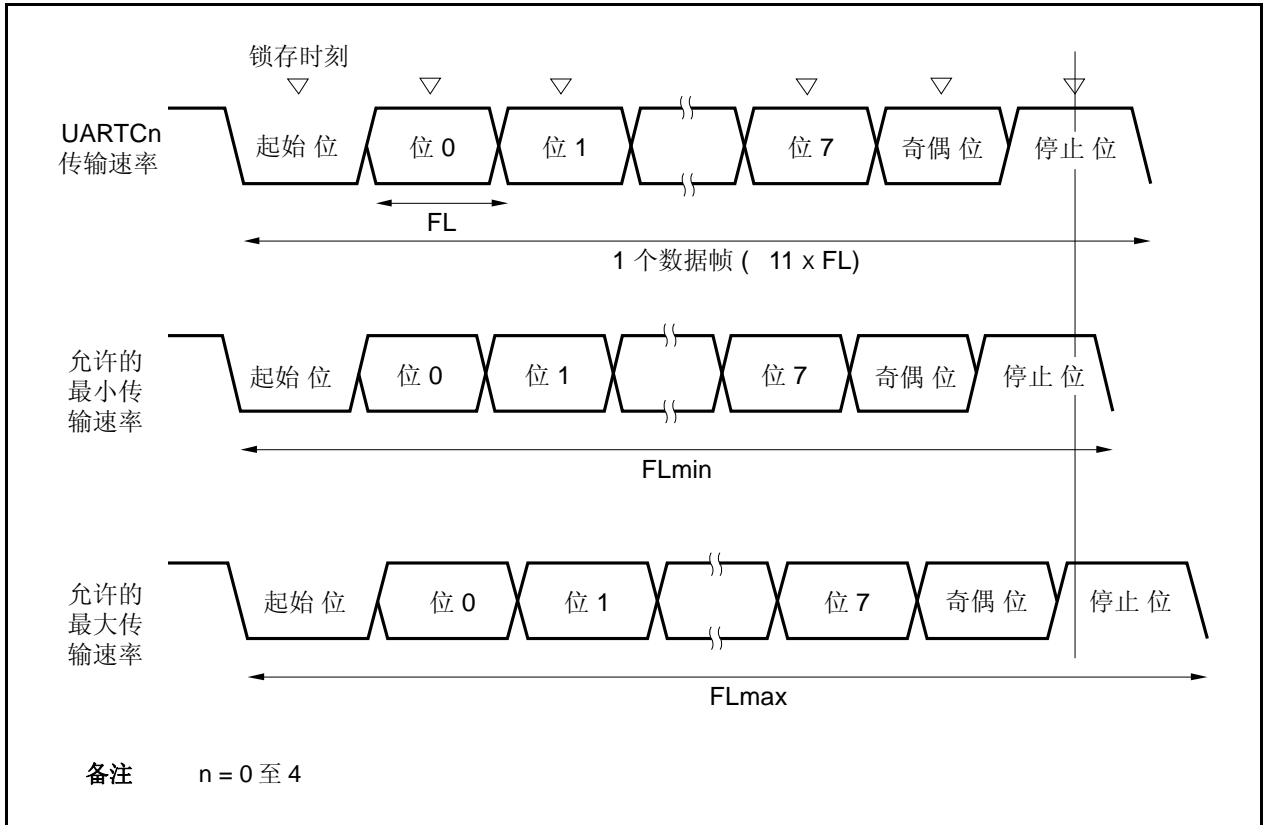
备注 f_{xx} : 主时钟频率
 ERR: 波特率误差 (%)

(5) 在接收期间允许的波特率范围

接收过程中，目的端允许的波特率误差范围如下所示：

注意事项 接收过程中使用以下公式，波特率误差必须设置在允许范围内。

图 17-20. 接收期间允许的波特率范围



如图 17-20 所示，接收数据的锁存时序取决于检测到起始位后 UCnCTL2 寄存器对计数器的设置值。如果在此锁存时序可以及时接收直到最末位数据（停止位），发送数据就可以被正常接收。

当应用于 11 位接收时，以下为理论结果。

$$FL = (\text{波特率})^{-1}$$

波特率：UARTCn 波特率(n = 0 至 4)

k: UCnCTL2.UCnBRS7 至 UCnCTL2.UCnBRS0 位的设定值(n = 0 至 4)。

FL: 1 位数据长度

锁存时间余量：2 个时钟

$$\text{最小允许传输速率: } FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

所以，可以从目的端接收的最大波特率如下所示：

$$BR_{\max} = (FL_{\min}/11)^{-1} = \frac{22k}{21k + 2} \text{ Brate}$$

同理，可以得到下列公式来计算最大允许传输速率：

$$\frac{10}{11} \times FL_{\max} = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FL_{\max} = \frac{21k-2}{20k} FL \times 11$$

所以，可以从目的端接收的最小波特率如下所示：

$$BR_{\min} = (FL_{\max}/11)^{-1} = \frac{20k}{21k - 2} \text{ Brate}$$

由上述公式计算 UARTCn 以及目的端的允许波特率误差，得到的最大和最小波特率值必须遵从以下表格。

表 17-5. 允许的最大/最小波特率误差

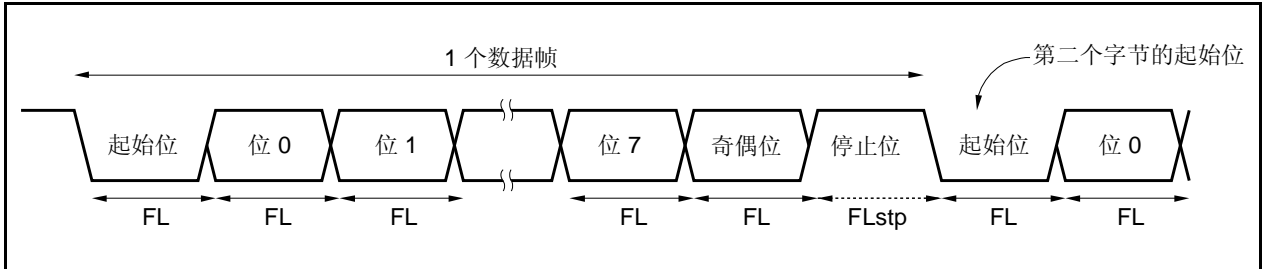
分频比(k)	允许的最大波特率误差	允许的最小波特率误差
4	+2.32%	- 2.43%
8	+3.52%	- 3.61%
20	+4.26%	- 4.30%
50	+4.56%	- 4.58%
100	+4.66%	- 4.67%
255	+4.72%	- 4.72%

- 备注
1. 接收精度依赖于帧内的位数量、输入时钟频率和分频比(k)。输入时钟频率越高且分频比(k)越大，则精度越高。
 2. k: UCnCTL2.UCnBRS7 位至 UCnCTL2.UCnBRS0 位的设定值(n = 0 至 4)

(6) 连续发送期间的传输速率

连续发送期间，从停止位到下一个起始位的传输间隔通常大于 2 个基础时钟。而时序的初始化由接收端通过检测起始位执行，因此，这对传输结果没有影响。

图 17-21. 连续发送期间的传送速率



假定 1 位数据长度为 FL；停止位长度为 FLstp；基础时钟频率为 f_{UCLK} ，可以得到如下表达式：

$$\text{FLstp} = \text{FL} + 2/f_{\text{UCLK}}$$

所以，连续发送期间的传输速率如下。

$$\text{传输速率} = 11 \times \text{FL} + (2/f_{\text{UCLK}})$$

17.8 注意事项

- (1) 当提供给 UARTCn 的时钟停止(比如进入 IDLE1、IDLE2 或 STOP 模式)时,则操作停止,每一个寄存器都保留时钟供应停止前的瞬间值。TXDCn 引脚输出也保持时钟供给停止前的瞬时值。然而,恢复时钟供给后,不能保证正常操作。所以,在时钟恢复之后,应该将 UCnCTL0.UCnPWR 位、UCnCTL0.UCnRXEn 位和 UCnCTL0.UCnTXEn 位设置为 000,对电路进行初始化。
- (2) RXDC1 和 KR7 引脚不能同时使用。使用 RXDC1 引脚时,不要使用 KR7 引脚。若使用 KR7 引脚,就不要使用 RXDC1 引脚(建议将 PFC91 位置为 1,并清除 PFCE91 位为 0)。
- (3) 按以下步骤启动 UARTCn。
 - <1> 将 UCnCTL0.UCnPWR 位置为 1。
 - <2> 设置此端口。
 - <3> 将 UCnCTL0.UCnTXE 位和 UCnCTL0.UCnRXE 位置为 1。
- (4) 按以下步骤停止 UARTCn。
 - <1>将 UCnCTL0.UCnTXE 位和 UCnCTL0.UCnRXE 位清为 0。
 - <2>设置端口,并设置 UCnCTL0.UCnPWR 位为 0(如果端口设置不改变,则不存在该问题)
- (5) 发送模式下(UCnCTL0.UCnPWR 位 = 1 且 UCnCTL0.UCnTXE 位 = 1),因为写入 UCnTX 寄存器会开始发送,所以不要用软件向该寄存器写入相同的值。若连续发送相同值,则重写相同的值。
- (6) 在连续发送中,从停止位到下一个起始位的通信速率要比平常延长 2 个以上基础时钟。但是,接收端通过检测起始位进行时序的初始化,因此接收结果不受影响。

第十八章 3-线可变长度串行 I/O (CSIF)

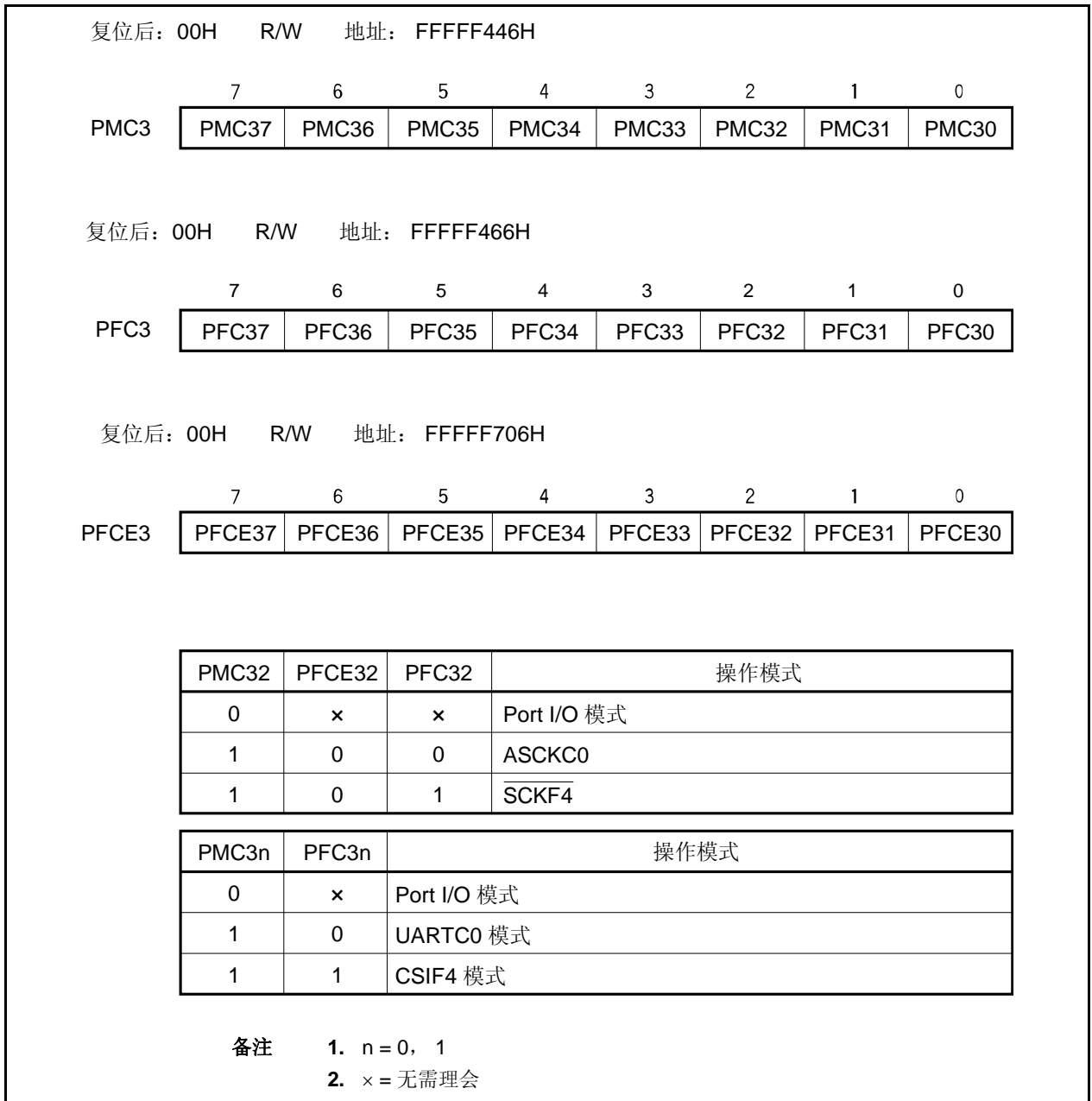
18.1 CSIF和其它串行接口的模式转换

18.1.1 CSIF4 和UARTC0 的模式转换

在 V850ES/JG3-U 和 V850ES/JH3-U 中, CSIF4 和 UARTC0 是相同引脚的复用功能, 因而不能同时使用。使用之前, 用 PMC3、PFC3 和 PFCE3 寄存器预先设置 CSIF4。

注意事项 如果在发送或接收期间切换这些功能, 则 CSIF4 和 UARTC0 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 18-1. CSIF4 和 UARTC0 模式转换设置



18.1.2 CSIF0、UARTC4 和 I²C01 的模式转换

在 V850ES/JG3-U 和 V850ES/JH3-U 中，CSIF0、UARTC4 和 I²C01 是相同引脚的复用功能，因而不能同时使用。使用之前，用 PMC4、PFC4 和 PFCE4 寄存器预先设置在 CSIF0、UARTC4 和 I²C01 之间切换。

注意事项 如果在发送或接收期间切换这些功能，则 CSIF0、UARTC4 和 I²C01 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 18-2. CSIF0、UARTC4 和 I²C01 模式转换设置

复位后: 00H R/W 地址: FFFFF448H

7	6	5	4	3	2	1	0
0	0	0	0	0	PMC42	PMC41	PMC40

复位后: 00H R/W 地址: FFFFF468H

7	6	5	4	3	2	1	0
0	0	0	0	0	PFC42	PFC41	PFC40

复位后: 00H R/W 地址: FFFFF708H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	PFCE41	PFCE40

PMC4n	PFCE4n	PFC4n	操作模式
0	×	×	Port I/O 模式
1	0	0	CSIF0 模式
1	0	1	UARTC4 模式
1	1	0	I ² C01 模式

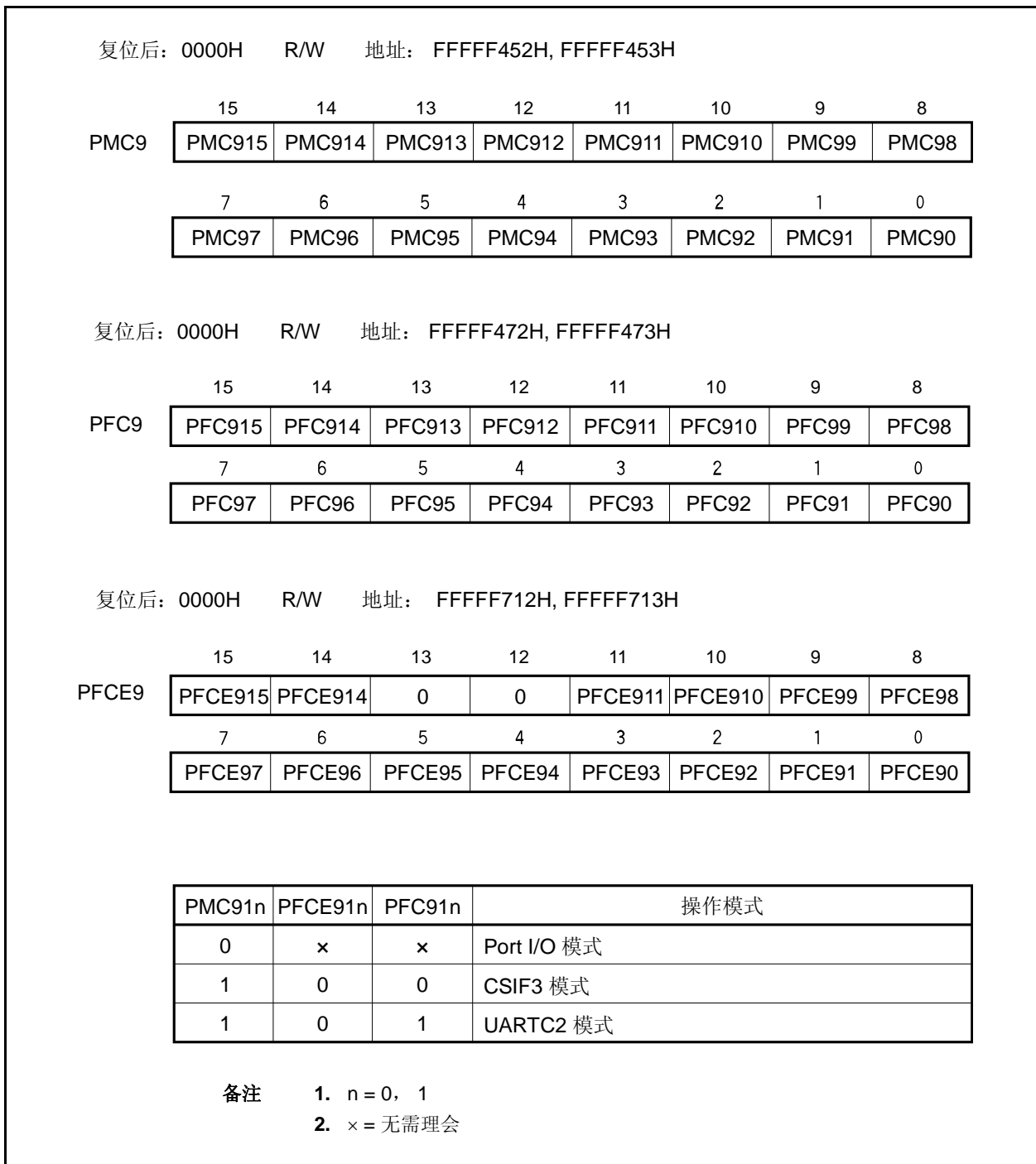
备注 1. n = 0, 1
 2. × = 无需理会

18.1.3 CSIF3 和UARTC2 的模式转换

在 V850ES/JG3-U 和 V850ES/JH3-U 中, CSIF3 和 UARTC2 是相同引脚的复用功能, 因而不能同时使用。使用之前, 用 PMC9、PFC9 和 PFCE9 寄存器预先设置在 CSIF3 和 UARTC2 之间切换。

注意事项 如果在发送或接收期间切换这些功能, 则 CSIF3 和 UARTC2 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 18-3. CSIF3 和 UARTC2 模式转换设置



18.2 特性

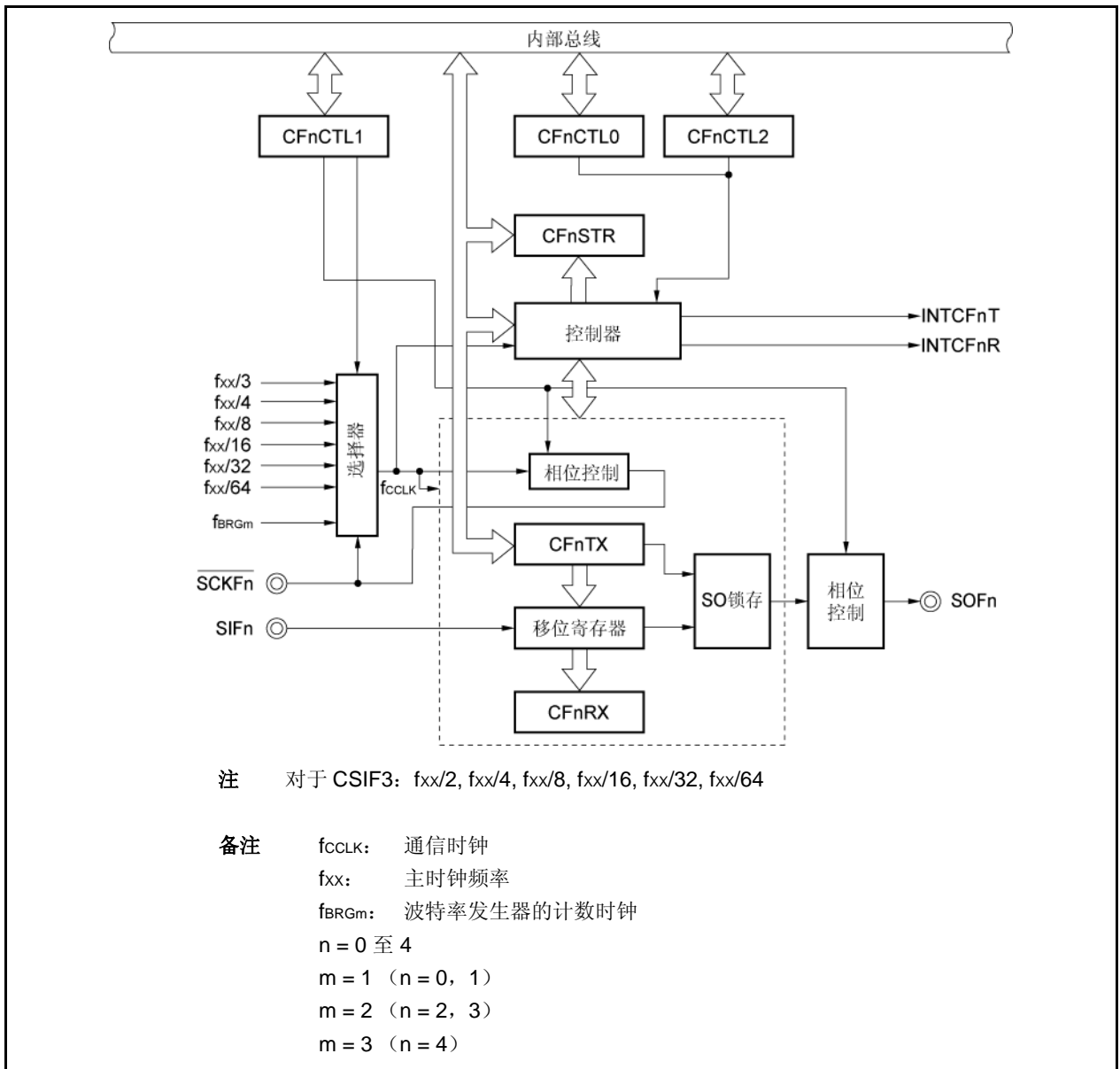
- 传输速率：最大 12 Mbps ($f_{xx} = 48 \text{ MHz}$ ，使用内部时钟，CSIF3)
最大 8 Mbps ($f_{xx} = 48 \text{ MHz}$ ，使用内部时钟，CSIF0 至 CSIF2，CSIF4)
- 主机、从模式可选
- 8-位至 16-位传输，3-线串行接口
- 中断请求信号 (INTCFnT, INTCFnR)
- 串行时钟和数据相位可翻转
- 传输数据长度可以在 8 位至 16 位之间选择任意位长度
- 可切换 MSB/LSB 先行传输数据
- 3-线传输
 - SOFn: 串行数据输出
 - SIFn: 串行数据输入
 - $\overline{\text{SCKFn}}$: 串行时钟输出/输出
- 可指定为发送模式、接收模式以及发送/接收模式

备注 n = 0 至 4

18.3 配置

以下显示 CSIFn 的功能框图。

图 18-4. CSIFn 框图



CSIFn 包括下列硬件：

表 18-1. CSIFn 的配置

项目	配置
寄存器	CSIFn 接收数据寄存器 (CFnRX) CSIFn 发送数据寄存器 (CFnTX)
控制寄存器	CSIFn 控制寄存器 0 (CFnCTL0) CSIFn 控制寄存器 1 (CFnCTL1) CSIFn 控制寄存器 2 (CFnCTL2) CSIFn 状态寄存器 (CFnSTR)

(1) CSIFn 接收数据寄存器 (CFnRX)

CFnRX 是 16-位缓存寄存器，用于存放接收数据。

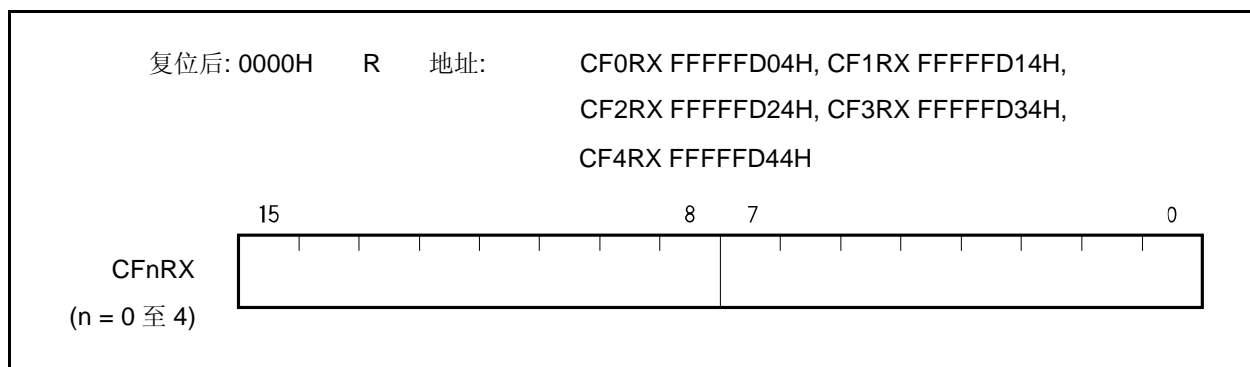
该寄存器是只读寄存器，以 16 位宽度读取。

在接收允许状态下，通过读取 CFnRX 寄存器启动接收操作。

如果传输数据长度为 8 位，该寄存器的低 8 位作为寄存器 CFnRXL，且只能以 8 位宽度进行读取。

系统复位后，该寄存器被设为 0000H。

除复位输入外，清除 CFnCTL0 寄存器的 CFnPWR 位（为 0），也可以对 CFnRX 寄存器进行初始化。



(2) CSIFn 发送数据寄存器 (CFnTX)

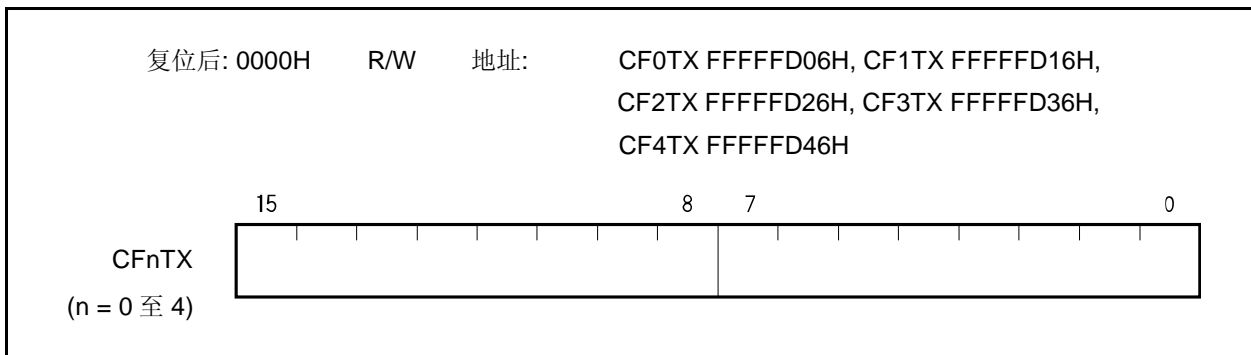
CFnTX 寄存器是 16 位缓冲寄存器，用于写入 CSIFn 的发送数据。

该寄存器可以 16 位宽度进行读取或写入。

在发送允许状态下，通过向 CFnTX 寄存器写入数据启动发送操作。

如果传输数据长度为 8 位，该寄存器的低 8 位作为寄存器 CFnTXL，且只能以 8 位宽度进行读取。

系统复位后，该寄存器被设为 0000H。



备注

通信开始条件如下所示。

发送模式 (CFnTXE 位 = 1, CFnRXE 位 = 0) :

写入 CFnTX 寄存器

发送/接收模式 (CFnTXE 位 = 1, CFnRXE 位 = 1) :

写入 CFnTX 寄存器

接收模式 (CFnTXE 位 = 0, CFnRXE 位 = 1) :

从 CFnRX 寄存器读取数据

18.4 寄存器

以下寄存器用来控制 CSIFn。

- CSIFn 控制寄存器 0 (CFnCTL0)
- CSIFn 控制寄存器 1 (CFnCTL1)
- CSIFn 控制寄存器 2 (CFnCTL2)
- CSIFn 状态寄存器 (CFnSTR)

(1) CSIFn 控制寄存器 0 (CFnCTL0)

CFnCTL0 寄存器用于控制 CSIFn 串行传输操作。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 01H。

(1/3)

复位后: 01H R/W 地址: CF0CTL0 FFFFFFFD00H, CF1CTL0 FFFFFFFD10H,
CF2CTL0 FFFFFFFD20H, CF3CTL0 FFFFFFFD30H,
CF4CTL0 FFFFFFFD40H

	<7>	<6>	<5>	<4>	3	2	1	<0>
CFnCTL0 (n = 0 至 4)	CFnPWR	CFnTXE ^注	CFnRXE ^注	CFnDIR ^注	0	0	CFnTMS ^注	CFnSCE

CFnPWR	CSIFn 操作禁止/使能的说明
0	禁止 CSIFn 操作并且清除 CFnSTR 寄存器
1	使能 CSIFn 操作
<ul style="list-style-type: none"> • CFnPWR 位控制 CSIFn 的操作，并复位内部电路。 	

CFnTXE ^注	发送操作禁止/使能的说明
0	禁止发送操作
1	使能发送操作
<ul style="list-style-type: none"> • CFnTXE 位为 0 时，SOFn 输出低电平。 	

CFnRXE ^注	接收操作禁止/使能的说明
0	禁止接收操作
1	使能接收操作
<ul style="list-style-type: none"> • 即使预置的数据被发送，也不会输出接收完成中断，且接收数据 (CFnRX 寄存器) 也不更新。因为通过清除 CFnRXE 为 0，接收操作被禁止。 	

注 只有当 CFnPWR 位= 0 时，这些位才可重写。可以在重写这些位的同时设置 CFnPWR 位= 1。

注意事项 若要强制挂起发送/接收，将 CFnPWR 位清为 0，而不是清除 CFnRXE 和 CFnTXE 位。
此时，时钟输出停止。

CFnDIR ^注	传输方向模式 (MSB/LSB)的说明
0	MSB先行传输
1	LSB先行传输

CFnTMS ^注	传输模式说明
0	单次传输模式
1	连续发送模式

[单次传输模式]

通信完成后，发生接收完成中断 (INTCFnR)。

即使允许发送 (CFnTXE 位 = 1)，也不产生发送使能中断(INTCFnT)。

如果通信期间(CFnSTR.CFnTSF 位 = 1)写入下一个发送数据，此数据被忽略，且开始下一次通信。同样如果设置为只接收通信(CFnTXE位=0, CFnRXE位=1)，即使通信期间读取接收数据 (CFnSTR.CFnTSF位=1)，也不开始进行下一次通信。

[连续发送模式]

在通信期间 (CFnSTR.CFnTSF位=1) 写入下一个发送数据，使能连续发送。

产生发送使能中断(INTCFnT)后，允许写入下一个发送数据。如果连续传输模式被设置为只接收通信(CFnTXE 位=0, CFnRXE 位 = 1)，产生接收完成中断 (INTCFnR) 后，下一次接收连续开始，而不考虑CFnRX寄存器的读取操作。

因此，从CFnRX寄存器立即读取接收数据。如果此读取操作被延迟，则会发生溢出错误 (CFnOVE位=1)。

注 只有当 CFnPWR 位= 0 时，这些位才可重写。可以在重写这些位的同时设置 CFnPWR 位= 1。

CFnSCE	启动传输禁止/使能的说明
0	通信启动触发无效
1	通信启动触发有效

• 主模式下
该位使能或禁止通信启动触发。

(a) 单次发送或发送/接收模式或连续发送或连续发送/接收模式
当CFnSCE位为1时，将数据写入CFnTX寄存器以启动通信。
设置CFnSCE位为1。

(b) 单次接收模式
因为要通过读取接收数据(CFnRX 寄存器)^{注1}来启动接收操作。所以在
读取最后一个接收数据之前将CFnSCE位清除为0，禁止启动下次接收操作。

(c) 连续接收模式
最后一个数据接收完成前一个通信时钟将CFnSCE位清除为0，使得最后
一个数据接收完成之后^{注2}禁止启动接收。

• 从模式下
该位使能或禁止通信启动触发
设置CFnSCE位为1。

[CFnSCE 位的用法]

• 单次接收模式

<1> 通过INTCFnR 中断服务完成最后数据的接收时，在读取CFnRX寄存器之前
清除CFnSCE 位为0。

<2> 确认 CFnSTR.CFnTSF 位 = 0之后，清除 CFnRXE 位为 0 来禁止接收。
若要继续接收，通过伪读取CFnRX寄存器将 CFnSCE 位设置为 1，来
启动下一次接收。

• 连续接收模式

<1> 最后一个数据接收期间通过INTCFnR中断服务来清除CFnSCE 位为 0。

<2> 读取 CFnRX 寄存器。

<3> 响应 CFnTIR中断后，通过读取CFnRX寄存器来读取最后一个接收数据。

<4> 确认 CFnSTR.CFnTSF 位 = 0之后，清除 CFnRXE 位为 0 来禁止接收。
若要继续接收，通过伪读取CFnRX寄存器将 CFnSCE 位设置为 1，来
启动下一次接收。

- 注
1. 如果其值为 1 时，读取 CFnSCE 位，则开始下一次通信操作。
 2. 最末数据接收结束之前的一个通信时钟，CFnSCE 位不清 0，则下一次通信操作自动开始。

注意事项 请确保将第 3 位和第 2 位清为“0”。

(2) CSIFn 控制寄存器 1 (CFnCTL1)

CFnCTL 是 8 位寄存器，用于控制 CSIFn 串行传输操作。

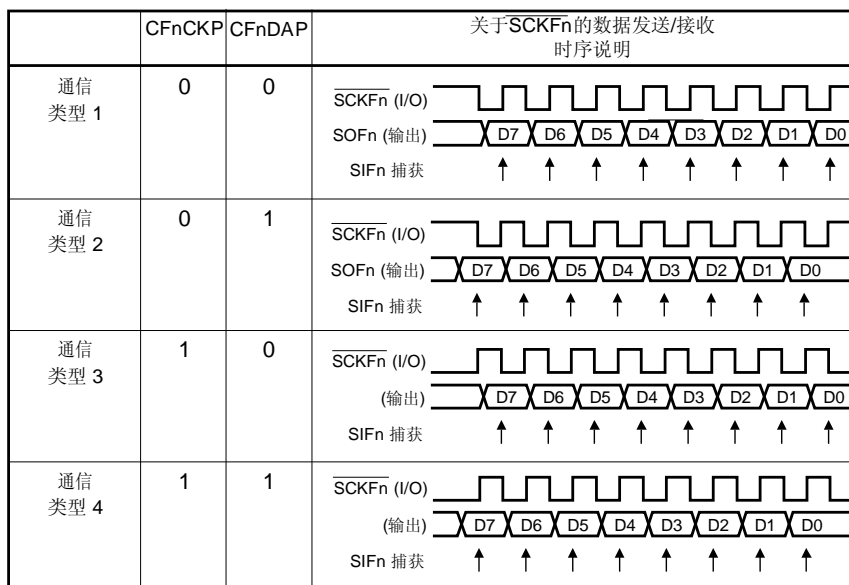
该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

注意事项 仅当 CFnCTL0.CFnPWR 位= 0 时，才可以重写 CFnCTL1 寄存器。

复位后: 00H R/W 地址: CF0CTL1 FFFFFFFD01H, CF1CTL1 FFFFFFFD11H,
CF2CTL1 FFFFFFFD21H, CF3CTL1 FFFFFFFD31H,
CF4CTL1 FFFFFFFD41H

	7	6	5	4	3	2	1	0
CFnCTL1	0	0	0	CFnCKP	CFnDAP	CFnCKS2	CFnCKS1	CFnCKS0
(n = 0 至 4)								



CFnCKS2	CFnCKS1	CFnCKS0	通信时钟 (fcCLK) ^注	模式
0	0	0	f _{xx} /2	主模式
0	0	1	f _{xx} /4	主模式
0	1	0	f _{xx} /8	主模式
0	1	1	f _{xx} /16	主模式
1	0	0	f _{xx} /32	主模式
1	0	1	f _{xx} /64	主模式
1	1	0	f _{BRGm}	主模式
1	1	1	外部时钟 (SCKFn)	从模式

- 注**
1. 设置通信时钟(fcCLK) 为 8 MHz 或更低。
 2. 设置通信时钟(fcCLK) 为 12 MHz 或更低。

备注

当 n = 0, 1, m = 1
 当 n = 2, 3, m = 2
 当 n = 4, m = 3
 关于 f_{BRGm} 详情，参见 18.8 波特率发生器。

(3) CSIFn 控制寄存器 2 (CFnCTL2)

CFnCTL2 是 8 位寄存器，用于控制 CSIFn 串行传输的位数量。

该寄存器可进行字节读取或写入。

系统复位后，该寄存器被设为 00H。

注意事项 仅当 CFnCTL0.CFnPWR 位= 0 或 CFnTXE 和 CFnRXE 位均为 0 时，才可以重写 CFnCTL2 寄存器。

复位后: 00H R/W 地址: CF0CTL2 FFFFFFFD02H, CF1CTL2 FFFFFFFD12H,
CF2CTL2 FFFFFFFD22H, CF3CTL2 FFFFFFFD32H,
CF4CTL2 FFFFFFFD42H

	7	6	5	4	3	2	1	0
CFnCTL2 (n = 0 至 4)	0	0	0	0	CFnCL3	CFnCL2	CFnCL1	CFnCL0

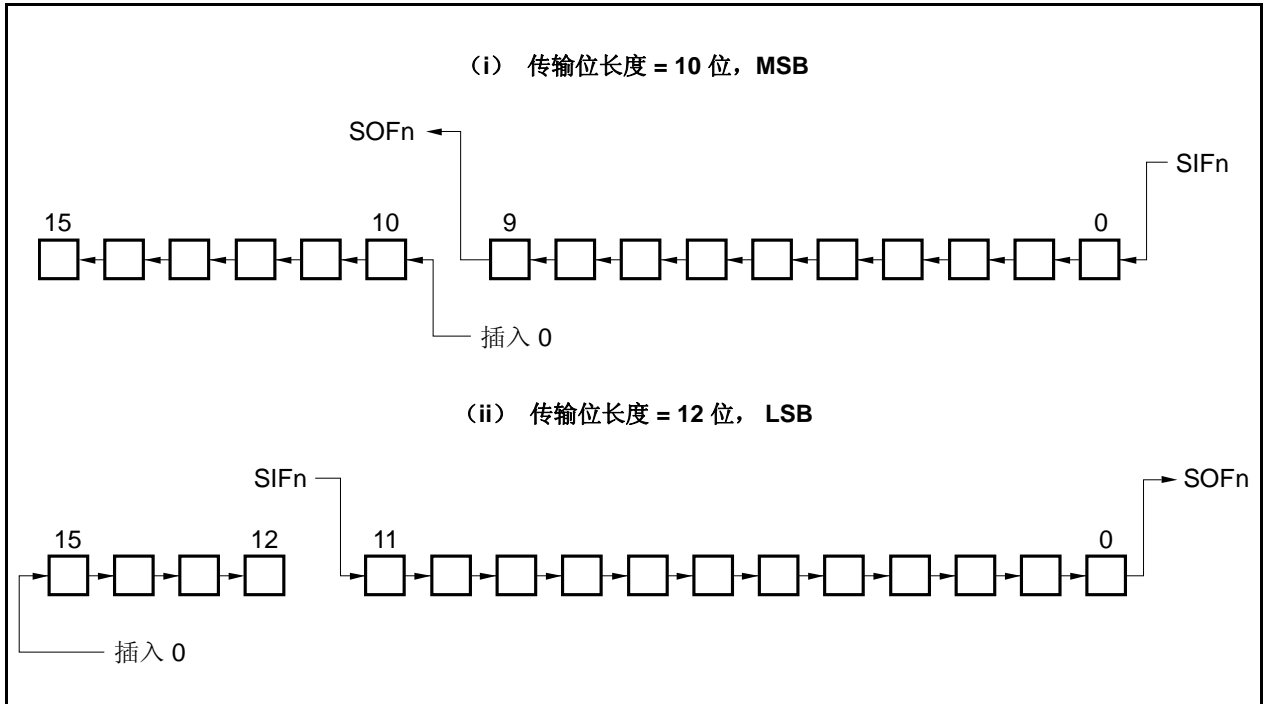
CFnCL3	CFnCL2	CFnCL1	CFnCL0	串行寄存器位的长度
0	0	0	0	8 位
0	0	0	1	9 位
0	0	1	0	10 位
0	0	1	1	11 位
0	1	0	0	12 位
0	1	0	1	13 位
0	1	1	0	14 位
0	1	1	1	15 位
1	x	x	x	16 位

- 备注**
1. 如果传输位数不是 8 位或 16 位，从 CFnTX 和 CFnRX 寄存器的 LSB 起进行填充。
 2. x：无需理会

(a) 传输数据长度改变功能

可以使用 CFnCTL2.CFnCL3 至 CFnCTL2.CFnCL0 位，在 8 位和 16 位之间以位为单位设置 CSIFn 传输的数据长度。

当设置的传输比特长度不是 16 位时，从 LSB 开始向 CFnTX 或 CFnRX 寄存器设置数据，而忽略传输起始位是 MSB 还是 LSB。没有使用的高位可以设置为任意数据，但是，在随后的串行传输中接收的对应数据变为 0。



18.5 中断请求信号

CSIFn 可以产生以下两种中断请求信号：

- 接收完成中断请求信号 (INTCFnR)
- 发送使能中断请求信号 (INTCFnT)

这两种中断请求信号中，默认接收完成中断请求信号有较高优先级，而发送允许中断请求信号优先级较低。

表 18-2. 中断与其默认优先级

中断	优先级
接收完成	高
发送允许	低

(1) 接收完成中断请求信号 (INTCFnR)

接收使能状态下，接收数据传送到 CFnRX 寄存器，产生接收完成中断请求信号。

如果发生溢出错误，同样也会产生中断请求信号。

响应接收完成中断请求信号并读取数据时，读取 CFnSTR 寄存器并检查接收结果是否错误。

单次传输模式中，即使只进行发送，传输完成后也会产生 INTCFnR 中断请求信号。

(2) 发送使能中断请求信号 (INTCFnT)

连续发送或连续发送/接收模式中，发送数据从 CFnTX 寄存器移出，并且一旦允许写入 CFnTX 寄存器，就会产生发送允许中断请求信号。

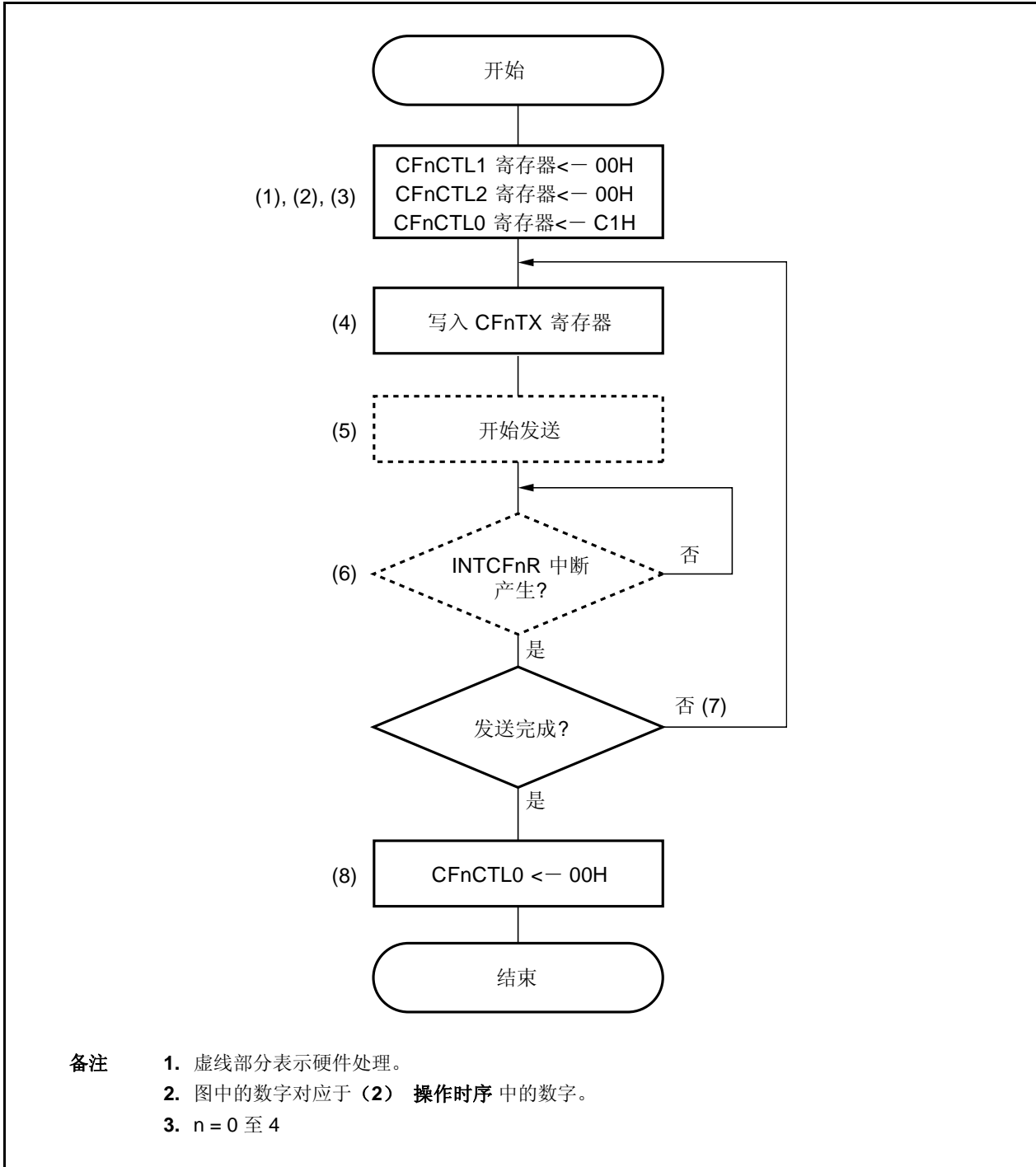
在单次发送和单次发送/接收模式中，不产生 INTCFnT 中断。

18.6 操作

18.6.1 单次传输模式（主模式，发送模式）

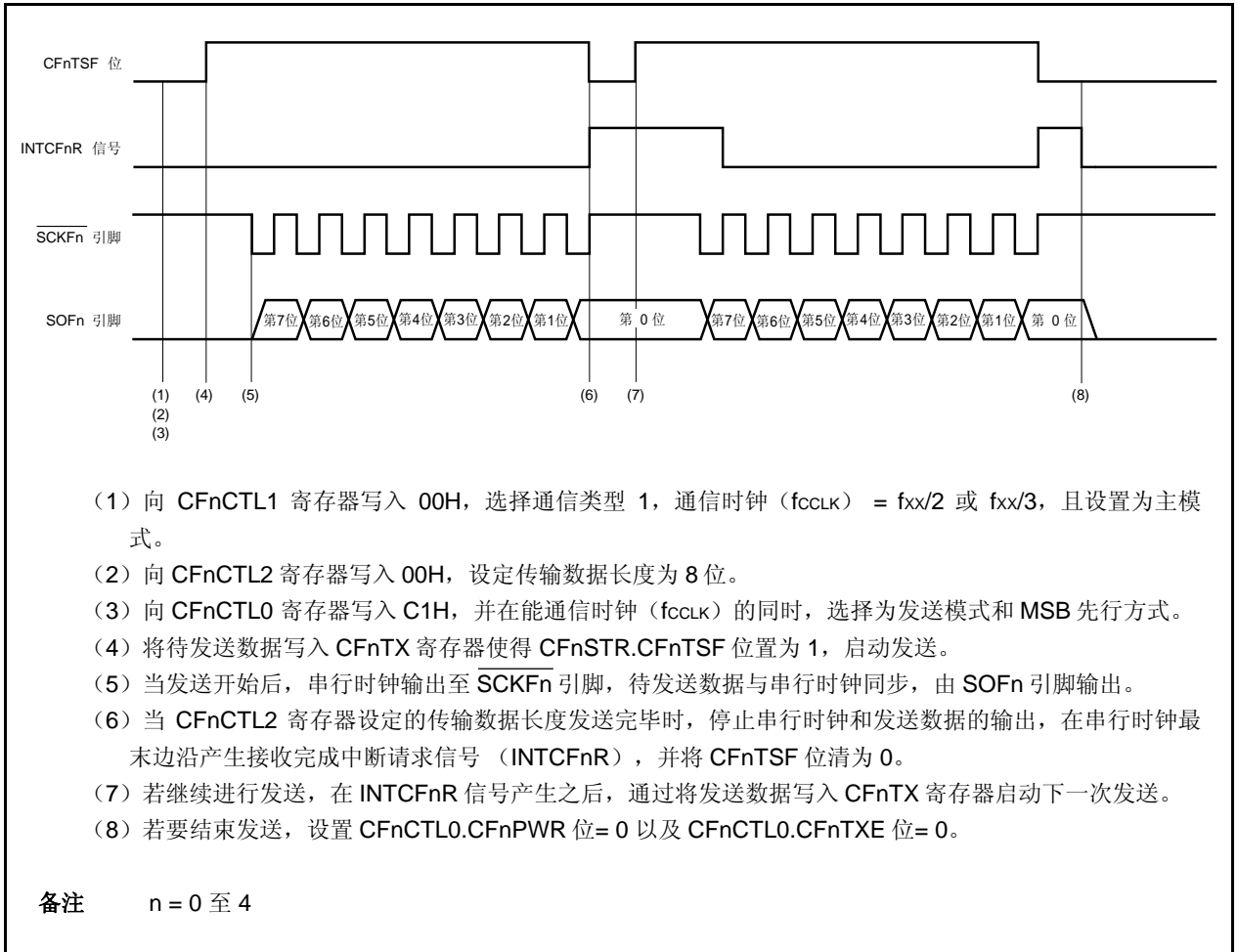
MSB 先行（CFnCTL0.CFnDIR 位= 0），通信类型 1（CFnCTL1.CFnCKP 和 CFnCTL1.CFnDAP 位= 00），通信时钟（f_{CLK}）= f_{xx}/2 或 f_{xx}/3（CFnCTL1.CFnCKS2 至 CFnCTL1.CFnCKS0 位 = 000），传输数据长度= 8 位（CFnCTL2.CFnCL3 至 CFnCTL2.CFnCL0 位= 0000）。

(1) 操作流程



- 备注
1. 虚线部分表示硬件处理。
 2. 图中的数字对应于 (2) 操作时序中的数字。
 3. n = 0 至 4

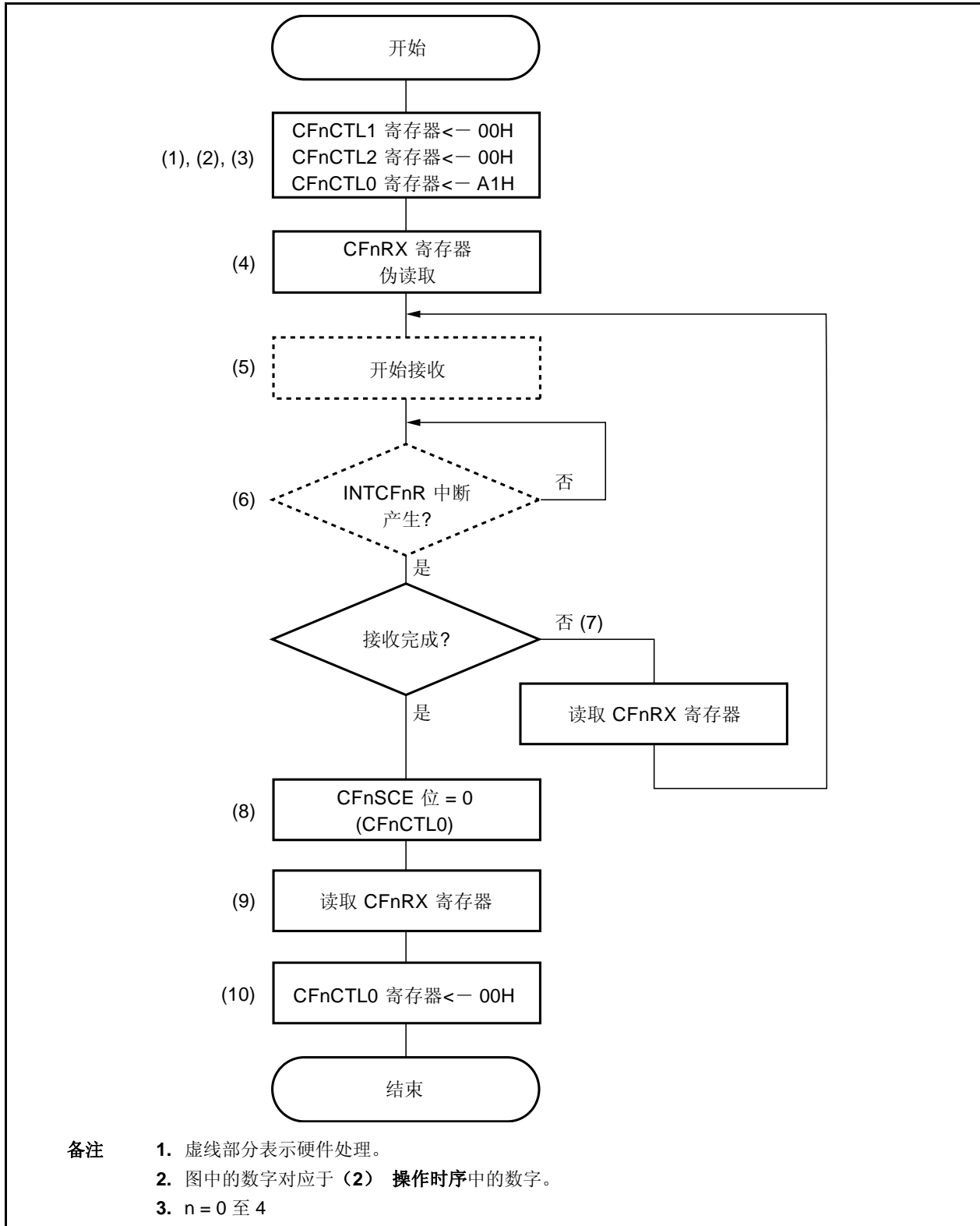
(2) 操作时序



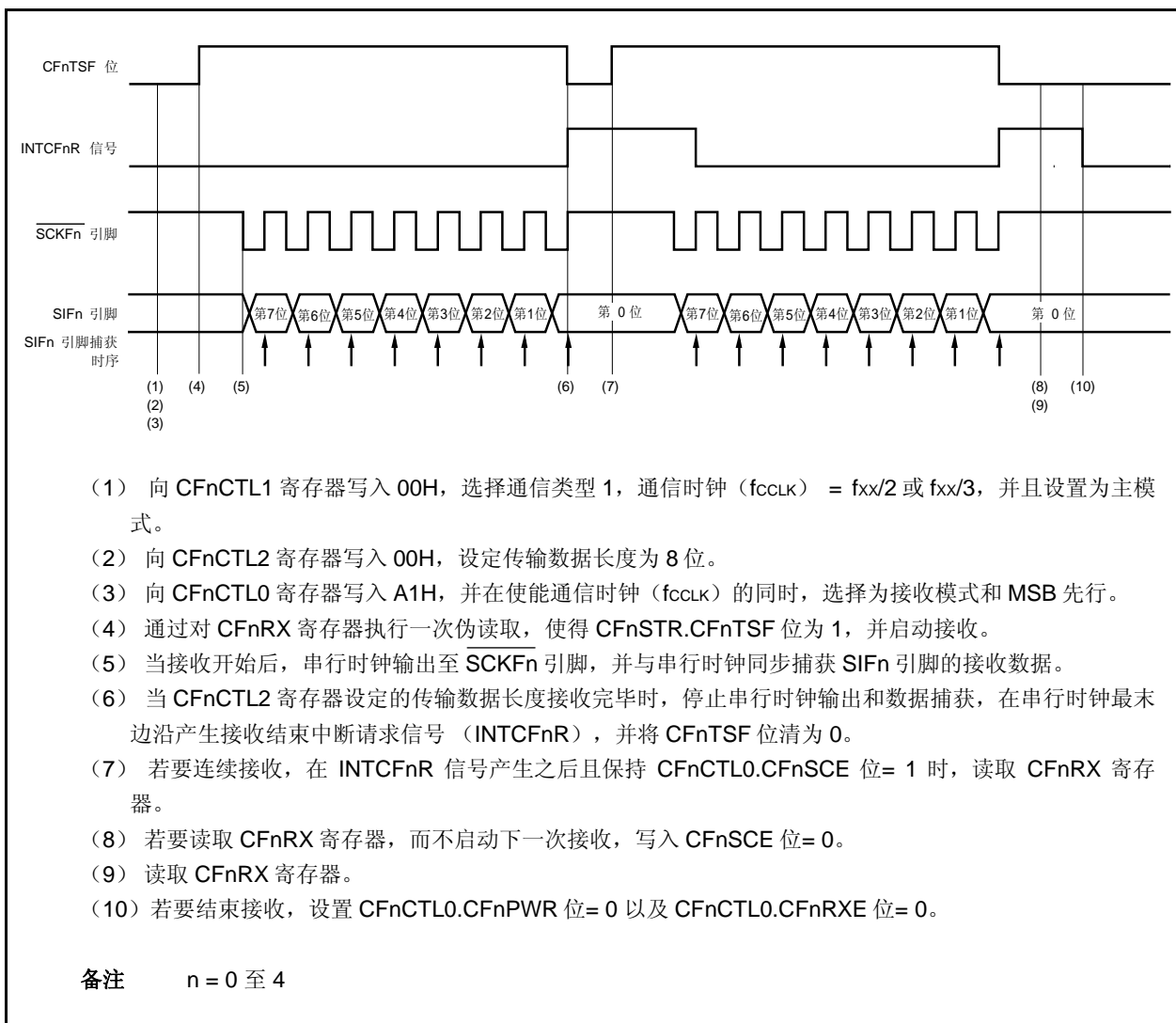
18.6.2 单次传输模式 (主模式, 接收模式)

MSB 先行 (CFnCTL0.CFnDIR 位= 0), 通信类型 1 (CFnCTL1.CFnCKP 和 CFnCTL1.CFnDAP 位= 00), 通信时钟 (f_{CLK}) = f_{xx}/2 或 f_{xx}/3 (CFnCTL1.CFnCKS2 至 CFnCTL1.CFnCKS0 位 = 000), 传输数据长度= 8 位 (CFnCTL2.CFnCL3 至 CFnCTL2.CFnCL0 位= 0000)。

(1) 操作流程



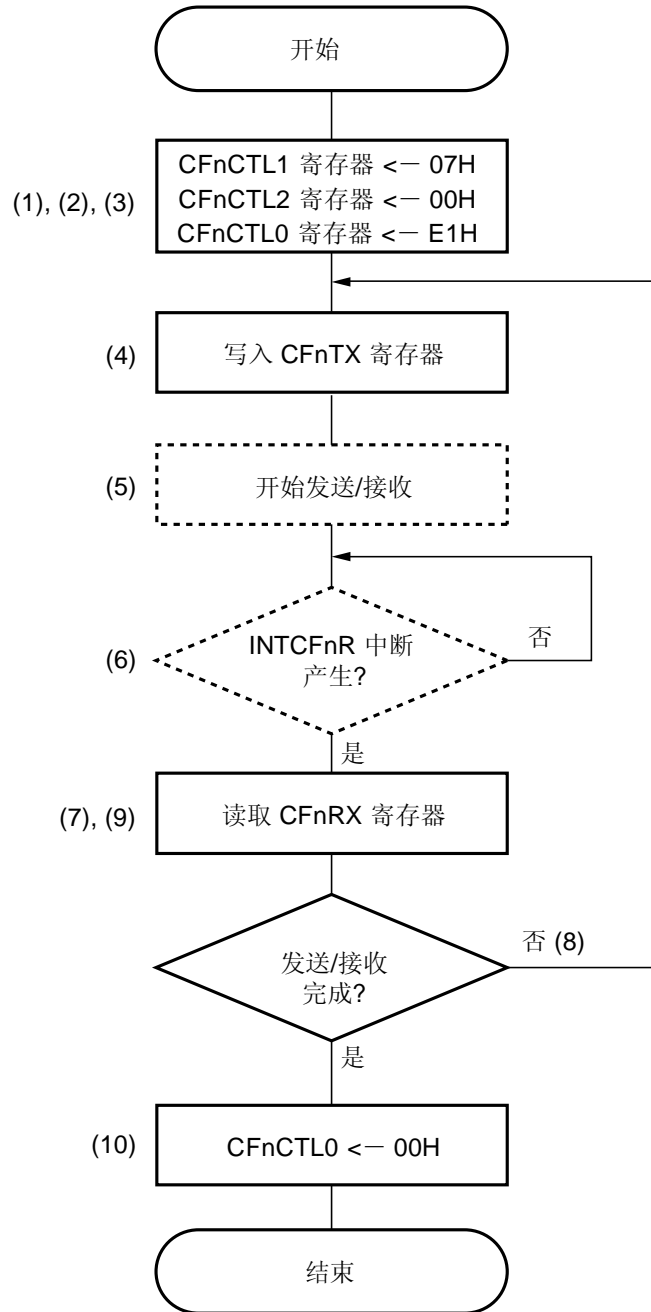
(2) 操作时序



18.6.3 单次传输模式 (主模式, 发送/接收模式)

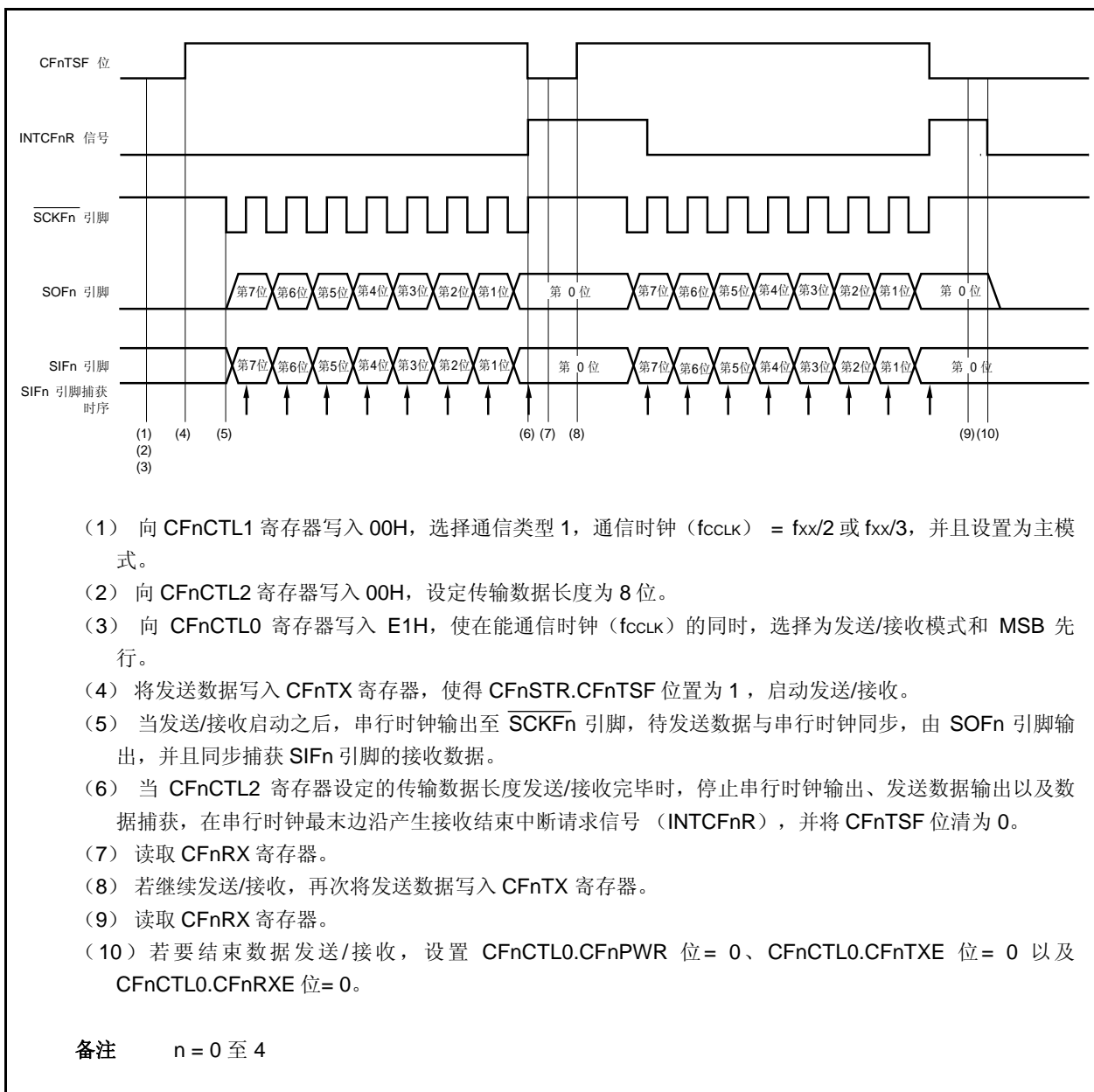
MSB 先行 (CFnCTL0.CFnDIR 位= 0), 通信类型 1 (CFnCTL1.CFnCKP 和 CFnCTL1.CFnDAP 位= 00), 通信时钟 (f_{CLK}) = f_{xx}/2 或 f_{xx}/3 (CFnCTL1.CFnCKS2 至 CFnCTL1.CFnCKS0 位 = 000), 传输数据长度= 8 位 (CFnCTL2.CFnCL3 至 CFnCTL2.CFnCL0 位= 0000)。

(1) 操作流程



- 备注
1. 虚线部分表示硬件处理。
 2. 图中的数字对应于 (2) 操作时序中的数字。
 3. n = 0 至 4

(2) 操作时序



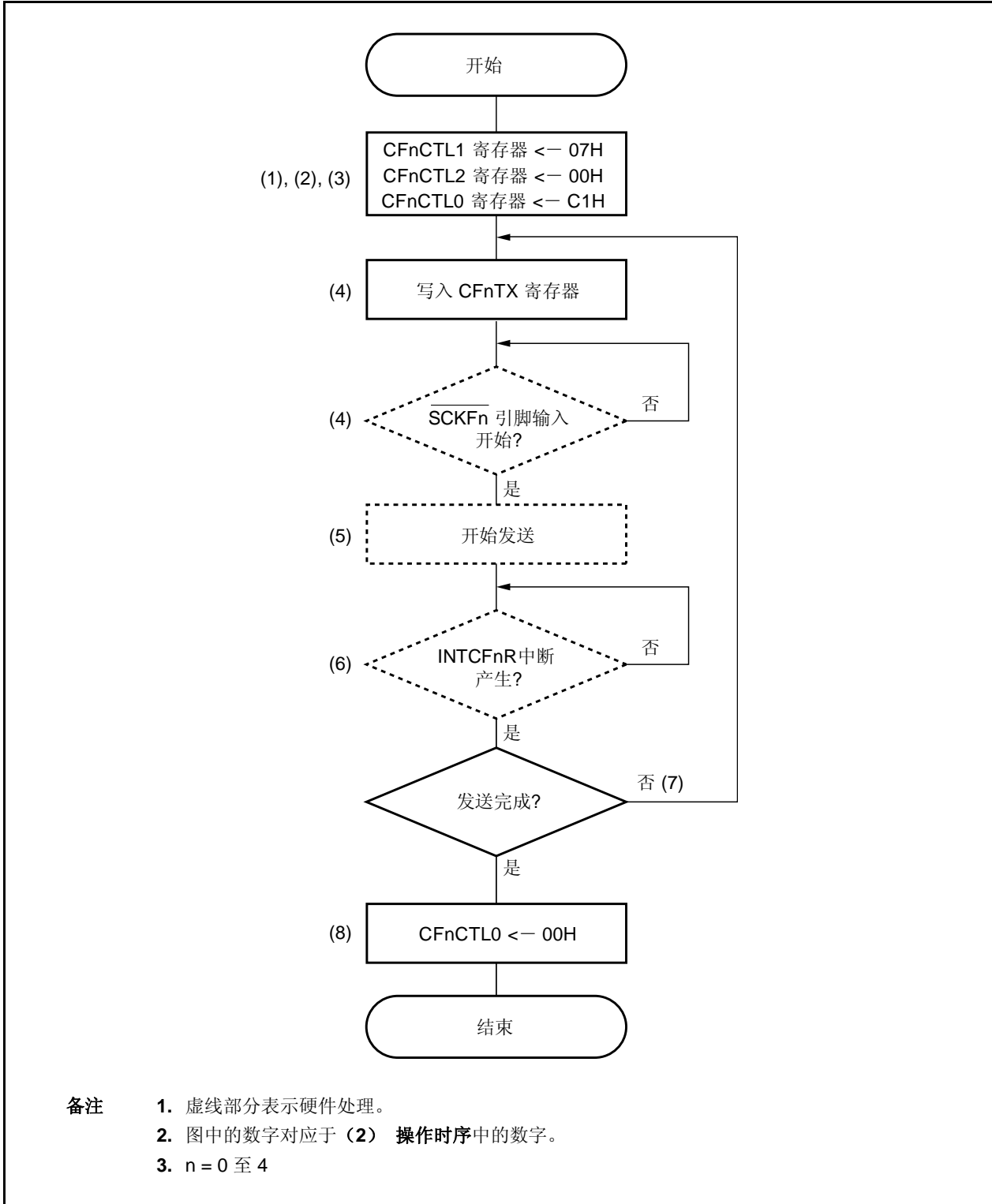
- (1) 向 CFnCTL1 寄存器写入 00H, 选择通信类型 1, 通信时钟 (f_{cCLK}) = f_{xx}/2 或 f_{xx}/3, 并且设置为主模式。
- (2) 向 CFnCTL2 寄存器写入 00H, 设定传输数据长度为 8 位。
- (3) 向 CFnCTL0 寄存器写入 E1H, 使在能通信时钟 (f_{cCLK}) 的同时, 选择为发送/接收模式和 MSB 先行。
- (4) 将发送数据写入 CFnTX 寄存器, 使得 CFnSTR.CFnTSMF 位置为 1, 启动发送/接收。
- (5) 当发送/接收启动之后, 串行时钟输出至 $\overline{\text{SCKFn}}$ 引脚, 待发送数据与串行时钟同步, 由 SOFn 引脚输出, 并且同步捕获 SIFn 引脚的接收数据。
- (6) 当 CFnCTL2 寄存器设定的传输数据长度发送/接收完毕时, 停止串行时钟输出、发送数据输出以及数据捕获, 在串行时钟最末边沿产生接收结束中断请求信号 (INTCFnR), 并将 CFnTSMF 位清为 0。
- (7) 读取 CFnRX 寄存器。
- (8) 若继续发送/接收, 再次将发送数据写入 CFnTX 寄存器。
- (9) 读取 CFnRX 寄存器。
- (10) 若要结束数据发送/接收, 设置 CFnCTL0.CFnPWR 位= 0、CFnCTL0.CFnTXE 位= 0 以及 CFnCTL0.CFnRXE 位= 0。

备注 n = 0 至 4

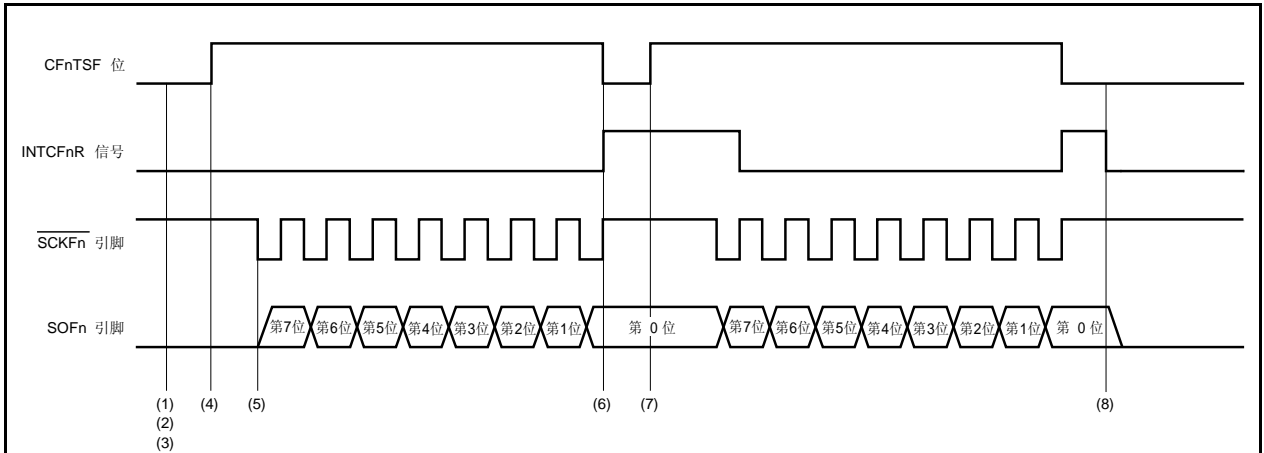
18.6.4 单次传输模式 (从模式, 发送模式)

MSB 先行 (CFnCTL0.CFnDIR 位= 0), 通信类型 1 (CFnCTL1.CFnCKP 和 CFnCTL1.CFnDAP 位= 00), 通信时钟 (f_{clock}) = 外部时钟 ($\overline{\text{SCKFn}}$) (CFnCTL1.CFnCKS2 至 CFnCTL1.CFnCKS0 位 = 111), 传输数据长度= 8 位 (CFnCTL2.CFnCL3 至 CFnCTL2.CFnCL0 位= 0000)。

(1) 操作流程



(2) 操作时序



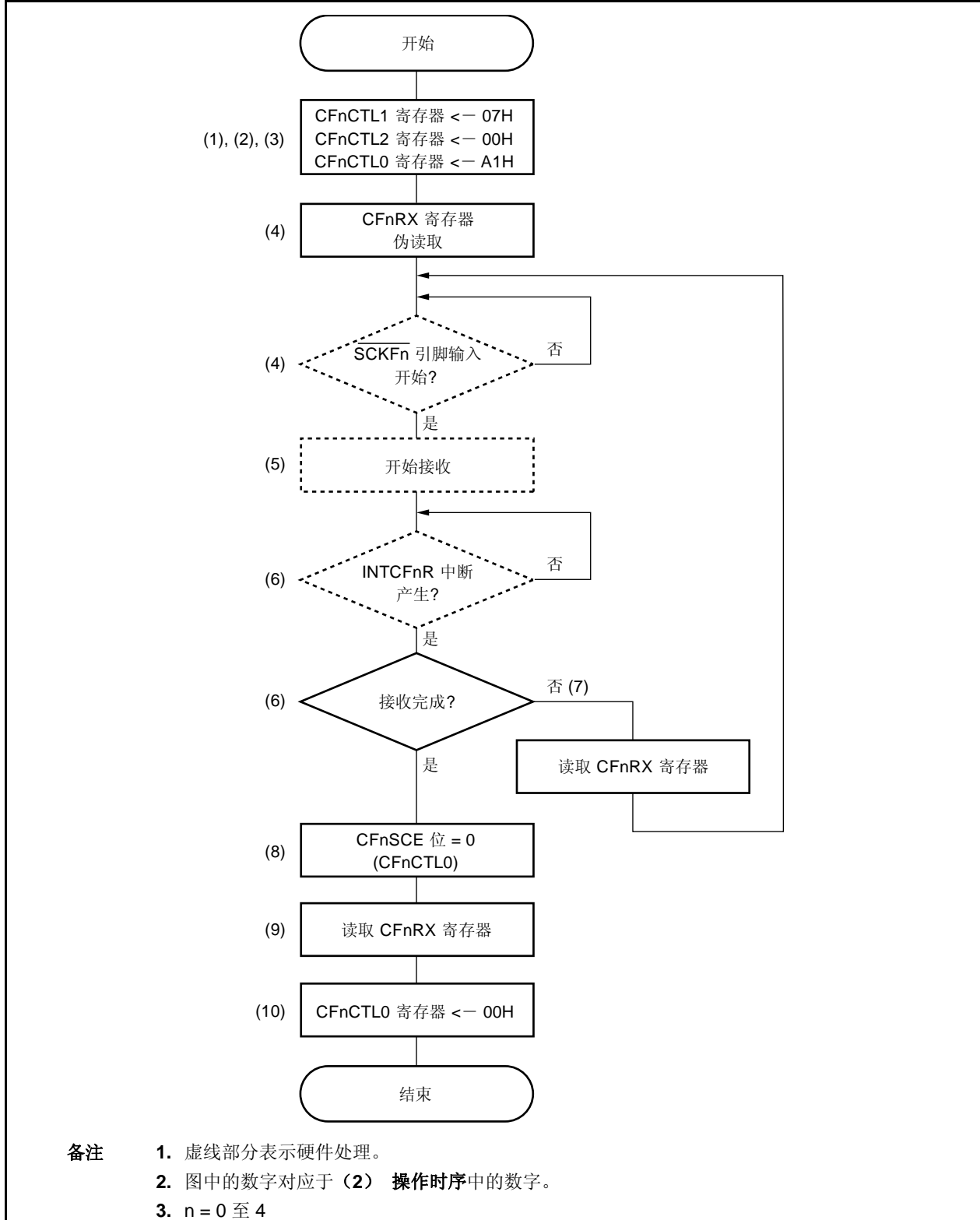
- (1) 向 CFnCTL1 寄存器写入 07H，选择通信类型 1，通信时钟 (fcCLK) = 外部时钟 (SCKFn)，并设置为从模式。
- (2) 向 CFnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。
- (3) 向 CFnCTL0 寄存器写入 C1H，并在使能通信时钟 (fcCLK) 的同时，选择为发送模式和 MSB 先行方式。
- (4) 将发送数据写入 CFnTX 寄存器，使得 CFnSTR.CFnTSF 位为 1，设备等待串行时钟输入。
- (5) 当串行时钟输入时，待发送的数据与串行时钟同步，由 SOFn 引脚输出。
- (6) 当 CFnCTL2 寄存器设定的传输数据长度发送完毕时，停止串行时钟的输入和发送数据的输出，在串行时钟最末边沿产生接收完成中断请求信号 (INTCFnR)，并将 CFnTSF 位清为 0。
- (7) 若要继续进行发送，在 INTCFnR 信号产生后，将发送数据写入 CFnTX 寄存器，并等待串行时钟的输入。
- (8) 若要结束发送，设置 CFnCTL0.CFnPWR 位= 0 以及 CFnCTL0.CFnTXE 位= 0。

备注 n = 0 至 4

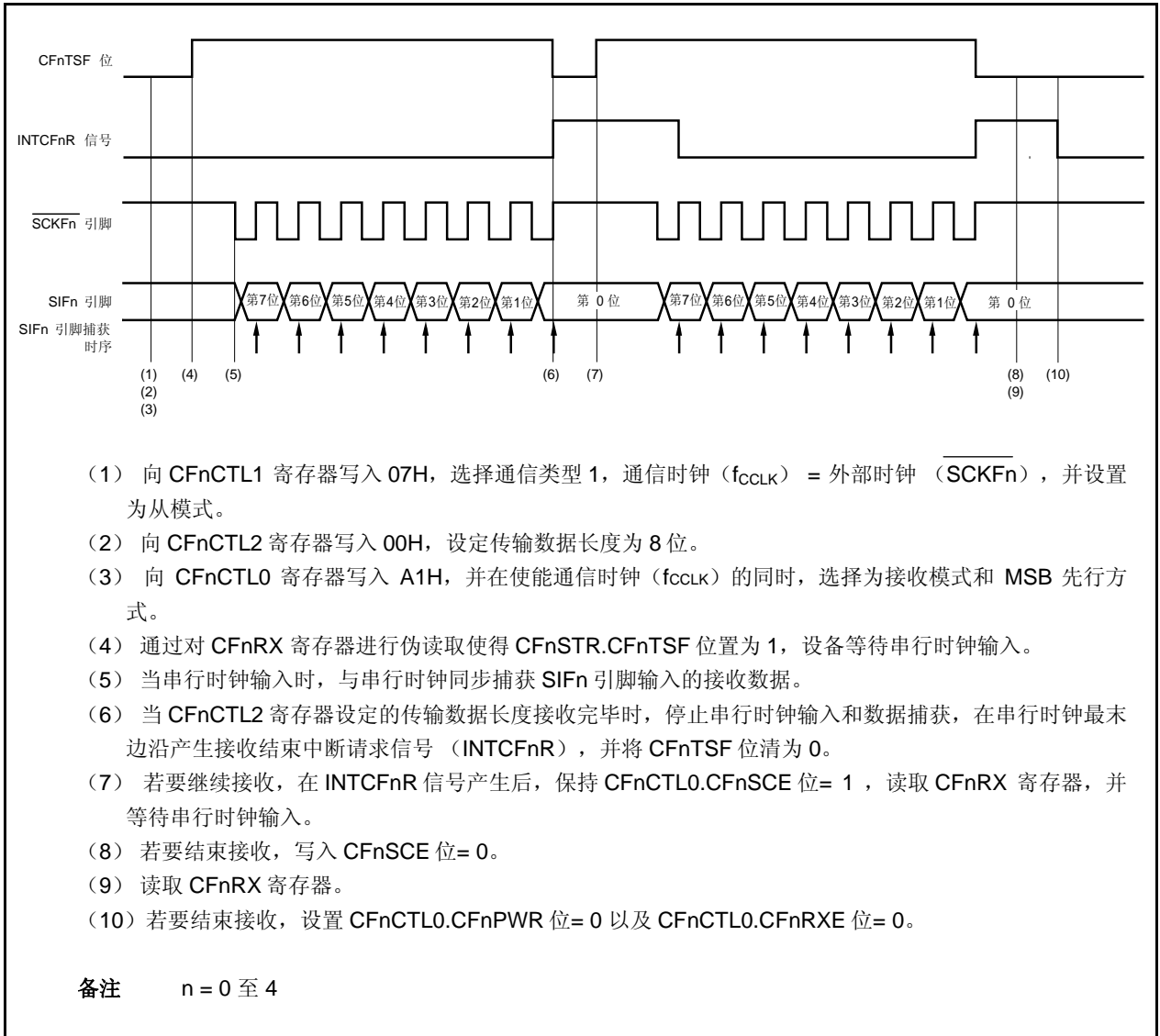
18.6.5 单次传输模式 (从模式, 接收模式)

MSB 先行 (CFnCTL0.CFnDIR 位= 0), 通信类型 1 (CFnCTL1.CFnCKP 和 CFnCTL1.CFnDAP 位= 00), 通信时钟 (f_{CLK}) = 外部时钟 (\overline{SCKFn}) (CFnCTL1.CFnCKS2 至 CFnCTL1.CFnCKS0 位 = 111), 传输数据长度= 8 位 (CFnCTL2.CFnCL3 至 CFnCTL2.CFnCL0 位= 0000)。

(1) 操作流程



(2) 操作时序



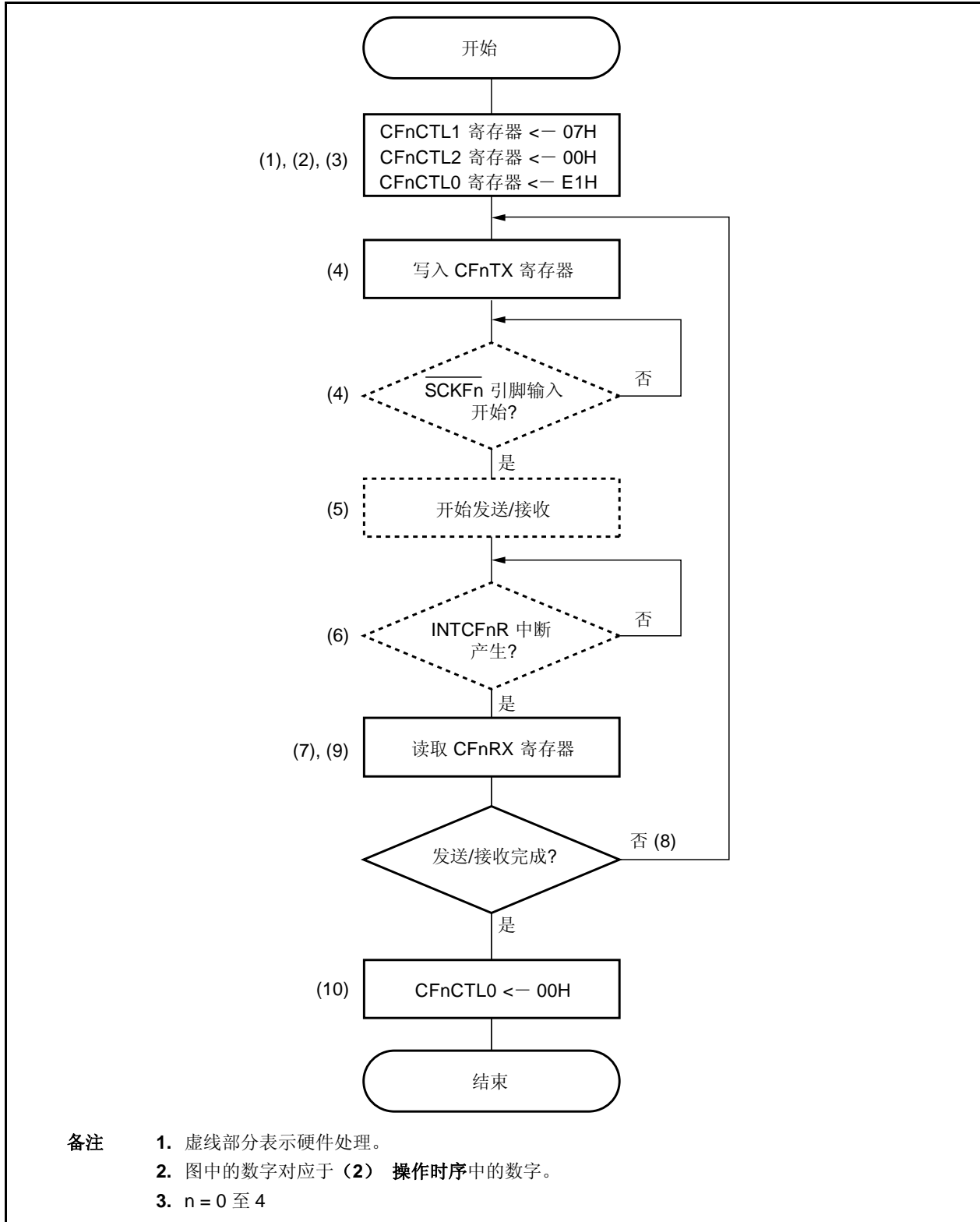
- (1) 向 CFnCTL1 寄存器写入 07H，选择通信类型 1，通信时钟 (f_{CCLK}) = 外部时钟 (SCKFn)，并设置为从模式。
- (2) 向 CFnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。
- (3) 向 CFnCTL0 寄存器写入 A1H，并在使能通信时钟 (f_{CCLK}) 的同时，选择为接收模式和 MSB 先行方式。
- (4) 通过对 CFnRX 寄存器进行伪读取使得 CFnSTR.CFnTSMF 位置为 1，设备等待串行时钟输入。
- (5) 当串行时钟输入时，与串行时钟同步捕获 SIFn 引脚输入的接收数据。
- (6) 当 CFnCTL2 寄存器设定的传输数据长度接收完毕时，停止串行时钟输入和数据捕获，在串行时钟最末边沿产生接收结束中断请求信号 (INTCFnR)，并将 CFnTSMF 位清为 0。
- (7) 若要继续接收，在 INTCFnR 信号产生后，保持 CFnCTL0.CFnSCE 位= 1，读取 CFnRX 寄存器，并等待串行时钟输入。
- (8) 若要结束接收，写入 CFnSCE 位= 0。
- (9) 读取 CFnRX 寄存器。
- (10) 若要结束接收，设置 CFnCTL0.CFnPWR 位= 0 以及 CFnCTL0.CFnRXE 位= 0。

备注 n = 0 至 4

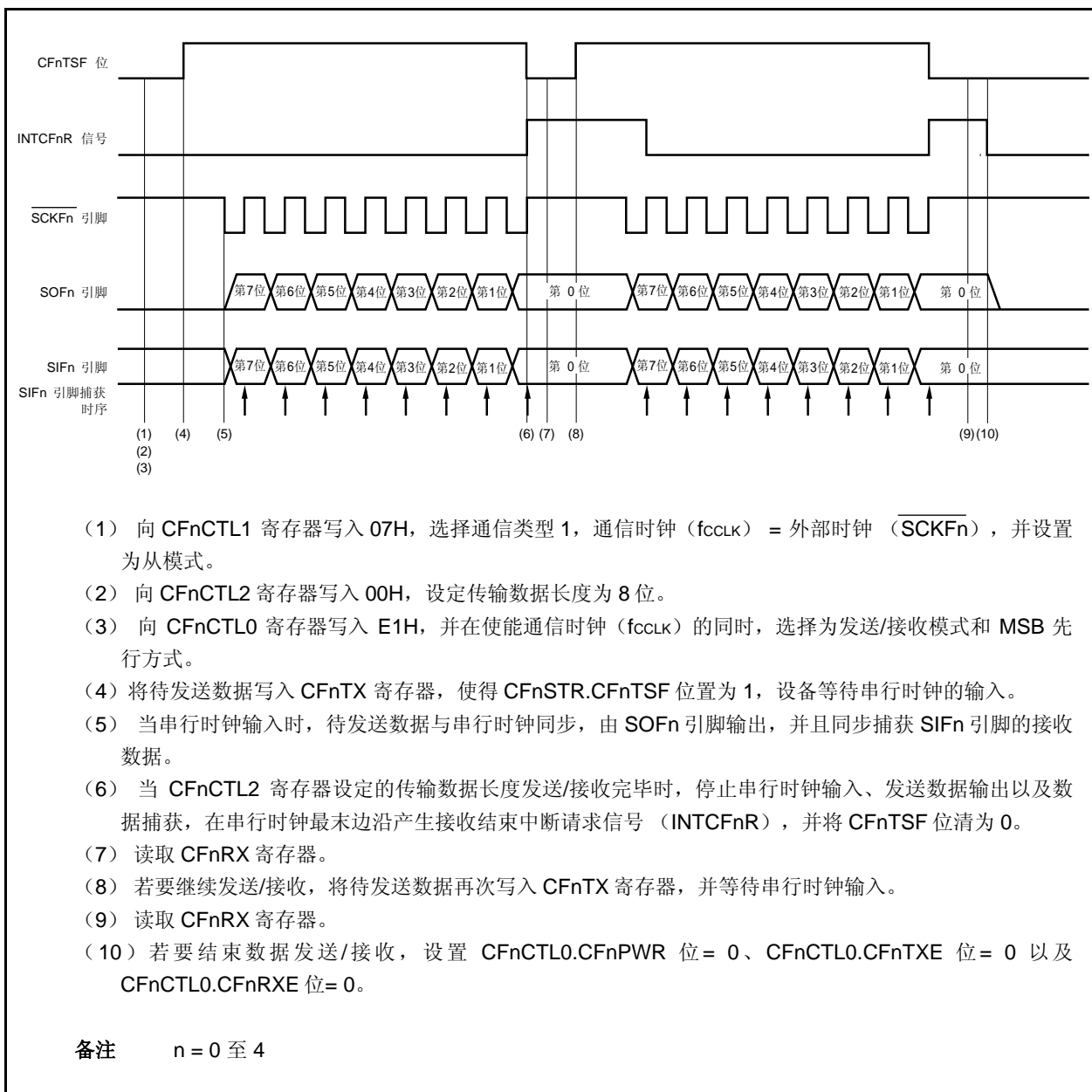
18.6.6 单次传输模式 (从模式, 发送/接收模式)

MSB 先行 (CFnCTL0.CFnDIR 位= 0), 通信类型 1 (CFnCTL1.CFnCKP 和 CFnCTL1.CFnDAP 位= 00), 通信时钟 (f_{clock}) = 外部时钟 (SCKFn) (CFnCTL1.CFnCKS2 至 CFnCTL1.CFnCKS0 位 = 111), 传输数据长度= 8 位 (CFnCTL2.CFnCL3 至 CFnCTL2.CFnCL0 位= 0000)。

(1) 操作流程



(2) 操作时序



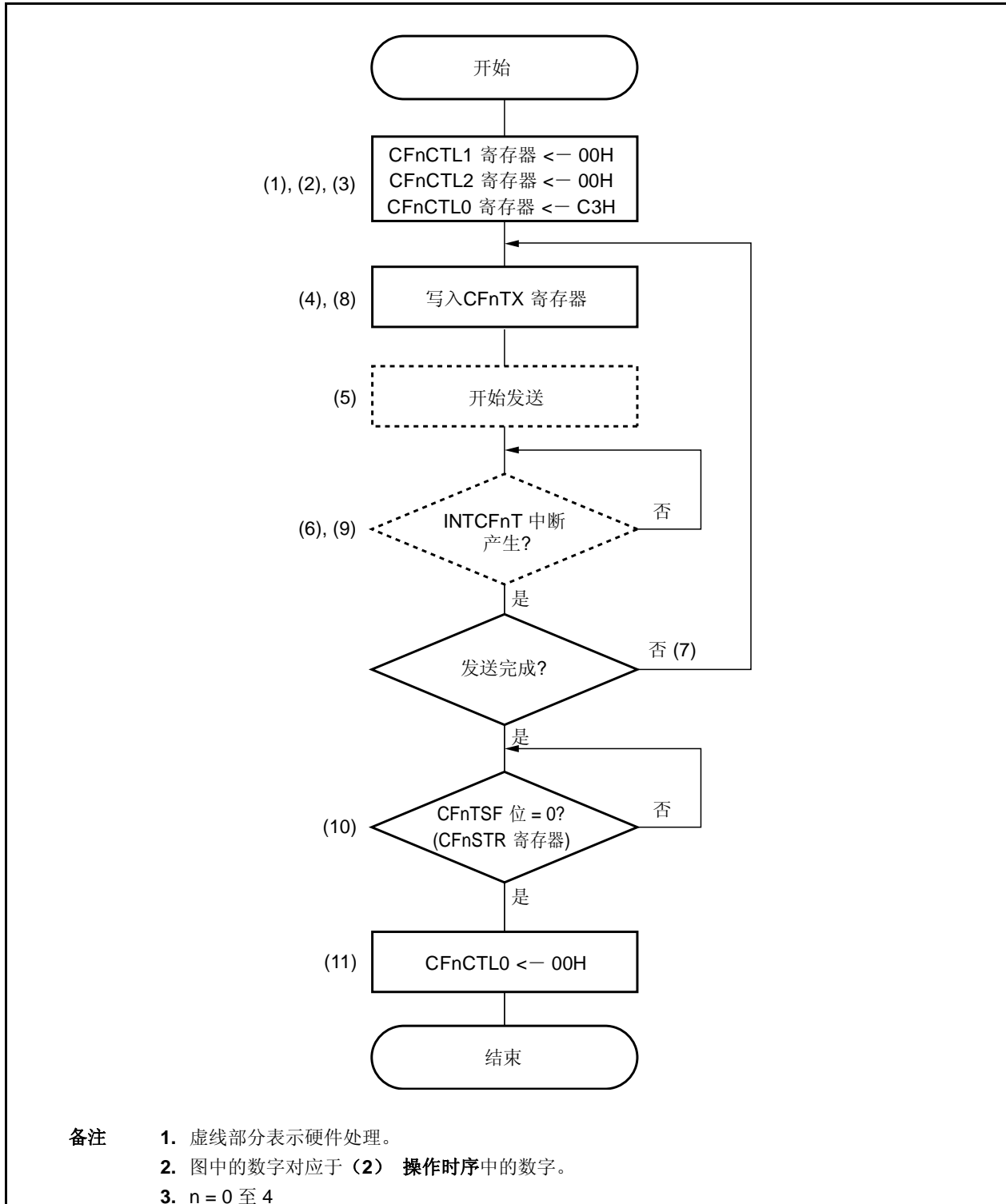
- (1) 向 CFnCTL1 寄存器写入 07H, 选择通信类型 1, 通信时钟 (f_{cCLK}) = 外部时钟 (SCKFn), 并设置为从模式。
- (2) 向 CFnCTL2 寄存器写入 00H, 设定传输数据长度为 8 位。
- (3) 向 CFnCTL0 寄存器写入 E1H, 并在使能通信时钟 (f_{cCLK}) 的同时, 选择为发送/接收模式和 MSB 先行方式。
- (4) 将待发送数据写入 CFnTX 寄存器, 使得 CFnSTR.CFnTSP 位置为 1, 设备等待串行时钟的输入。
- (5) 当串行时钟输入时, 待发送数据与串行时钟同步, 由 SOFn 引脚输出, 并且同步捕获 SIFn 引脚的接收数据。
- (6) 当 CFnCTL2 寄存器设定的传输数据长度发送/接收完毕时, 停止串行时钟输入、发送数据输出以及数据捕获, 在串行时钟最末边沿产生接收结束中断请求信号 (INTCFnR), 并将 CFnTSP 位清为 0。
- (7) 读取 CFnRX 寄存器。
- (8) 若要继续发送/接收, 将待发送数据再次写入 CFnTX 寄存器, 并等待串行时钟输入。
- (9) 读取 CFnRX 寄存器。
- (10) 若要结束数据发送/接收, 设置 CFnCTL0.CFnPWR 位= 0、CFnCTL0.CFnTXE 位= 0 以及 CFnCTL0.CFnRXE 位= 0。

备注 n = 0 至 4

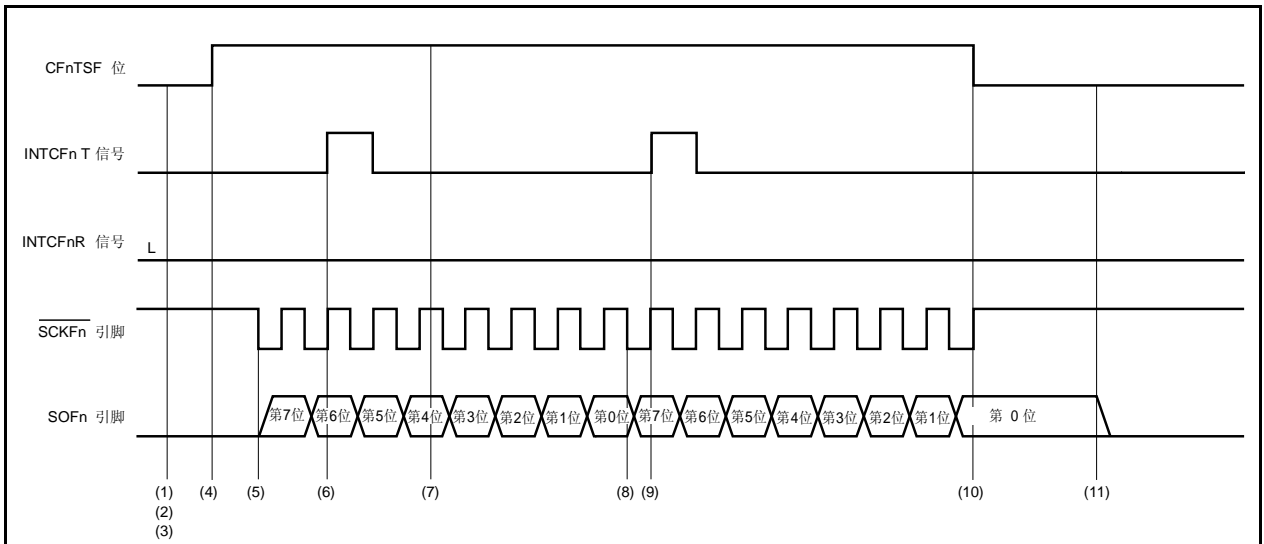
18.6.7 连续传输模式 (主模式, 发送模式)

MSB 先行 (CFnCTL0.CFnDIR 位= 0), 通信类型 1 (CFnCTL1.CFnCKP 和 CFnCTL1.CFnDAP 位= 00), 通信时钟 (f_{CLK}) = f_x/2 或 f_x/3 (CFnCTL1.CFnCKS2 至 CFnCTL1.CFnCKS0 位 = 000), 传输数据长度= 8 位 (CFnCTL2.CFnCL3 至 CFnCTL2.CFnCL0 位= 0000)。

(1) 操作流程



(2) 操作时序



- (1) 向 CFnCTL1 寄存器写入 00H，选择通信类型 1，通信时钟 (f_{cCLK}) = f_{xx}/2 或 f_{xx}/3，并设置为主模式。
- (2) 向 CFnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。
- (3) 向 CFnCTL0 寄存器写入 C3H，在使能通信时钟 (f_{cCLK}) 操作的同时，选择为发送模式、MSB 先行以及连续传输模式。
- (4) 将待发送数据写入 CFnTX 寄存器使得 CFnSTR.CFnTSF 位置为 1，启动发送。
- (5) 当发送开始后，输出串行时钟至 SCKFn 引脚，待发送数据与串行时钟同步，由 SOFn 引脚输出。
- (6) 当发送数据从 CFnTX 寄存器传输至移位寄存器完毕，且允许写入 CFnTX 寄存器时，产生发送允许中断请求信号 (INTCFnT)。
- (7) 若要连续发送，在产生 INTCFnT 信号之后，再次将发送数据写入 CFnTX 寄存器。
- (8) 当通信完成之前，新的发送数据已经写入 CFnTX 寄存器，则在本次通信完成后立即开始下一次通信。
- (9) 从 CFnTX 寄存器传输发送数据至移位寄存器完毕，且产生 INTCFnT 信号。若要结束当前的连续发送传输，不要写入 CFnTX 寄存器。
- (10) 当传输完毕之前下一个发送数据未被写入 CFnTX 寄存器，传输完毕后，则停止输出串行时钟至 SCKFn 引脚，并将 CFnTSF 位清除为 0。
- (11) 若要解除发送使能状态，在检查 CFnTSF 位 = 0 之后，设置 CFnCTL0.CFnPWR 位 = 0 和 CFnCTL0.CFnTXE 位 = 0。

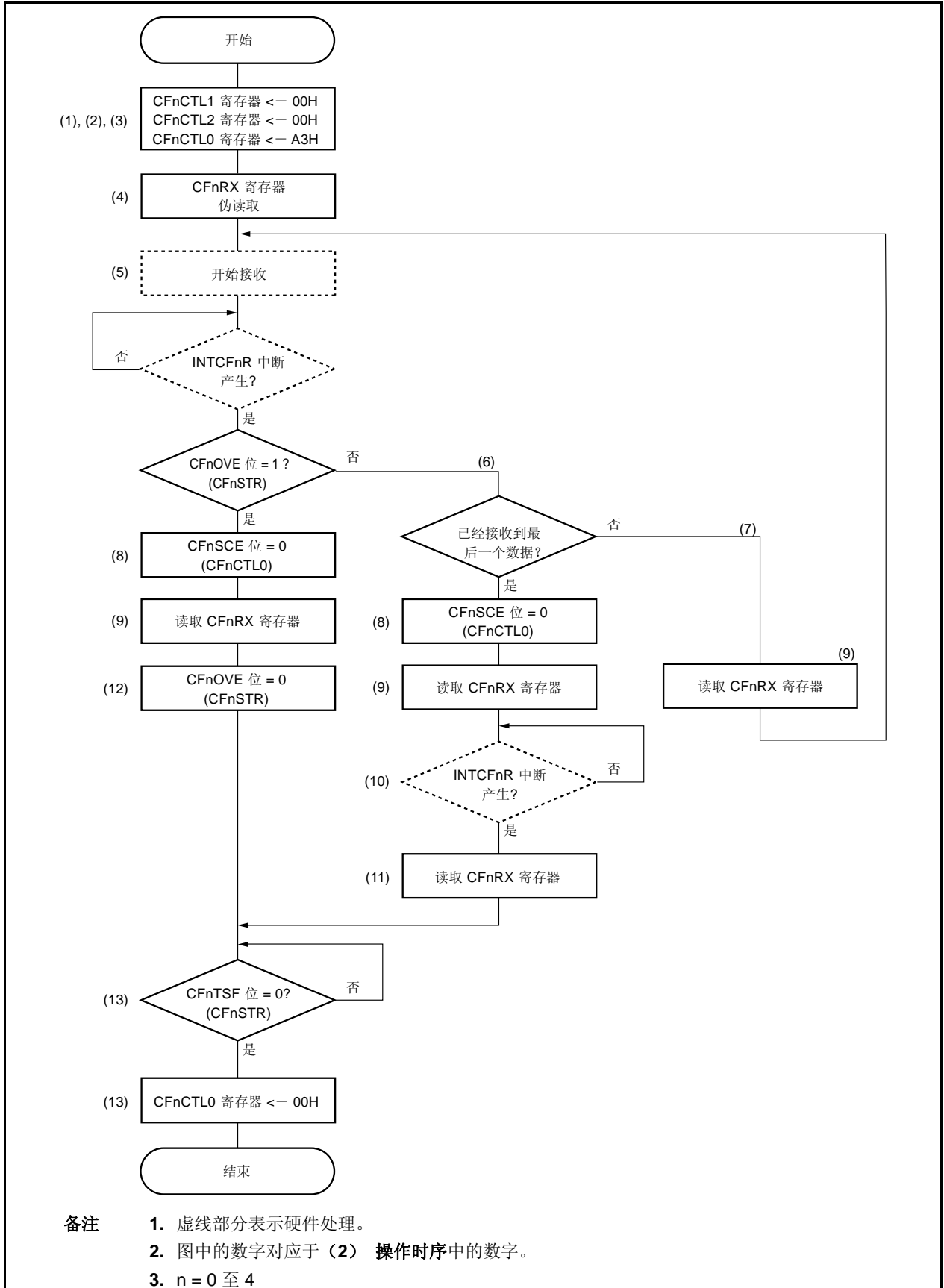
注意事项 连续发送模式下，不产生接收完成中断请求信号 (INTCFnR)。

备注 n = 0 至 4

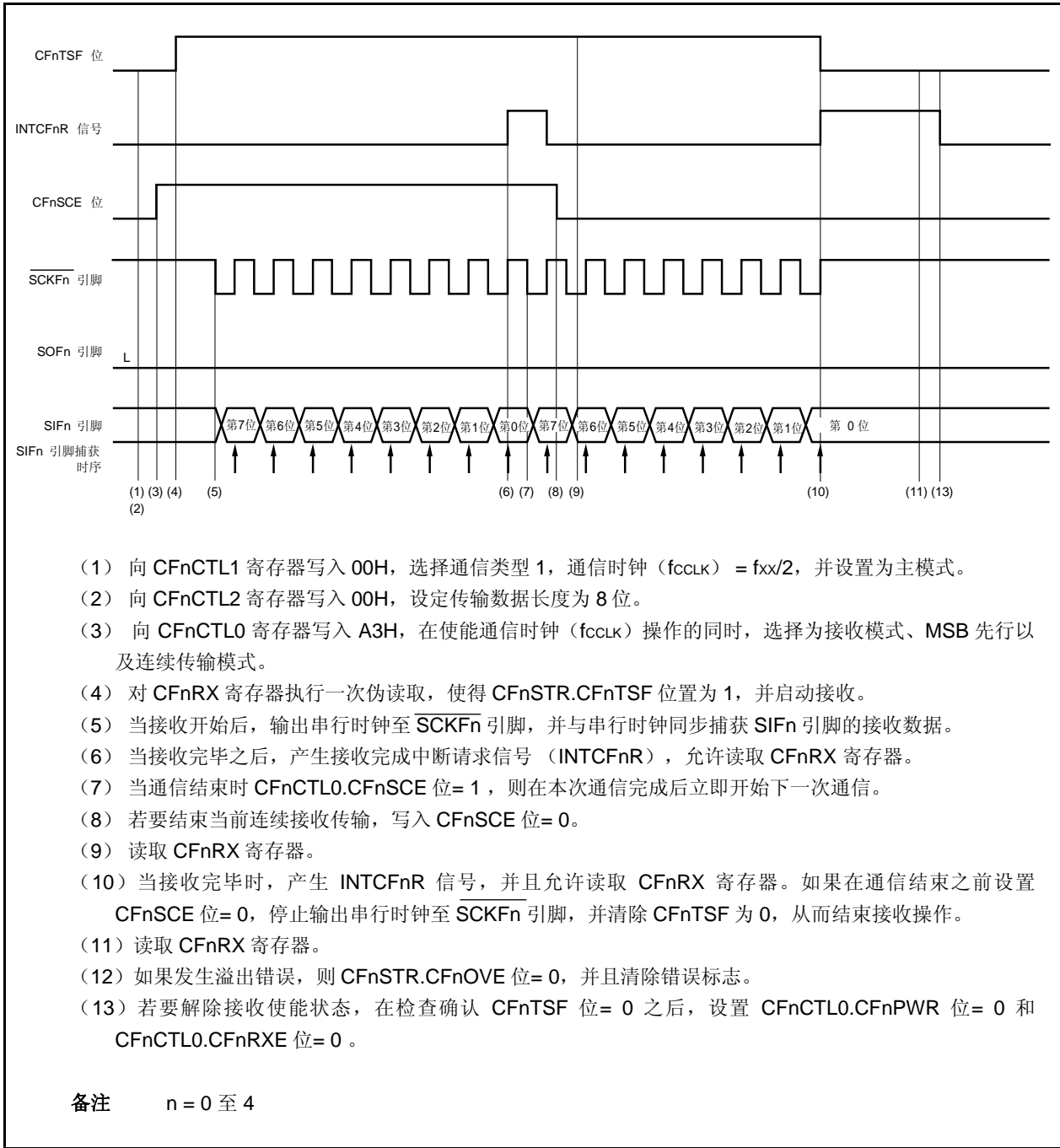
18.6.8 连续传输模式 (主模式, 接收模式)

MSB 先行 (CFnCTL0.CFnDIR 位= 0), 通信类型 1 (CFnCTL1.CFnCKP 和 CFnCTL1.CFnDAP 位= 00), 通信时钟 (f_{CCLK}) = f_{xx}/2 或 f_{xx}/3 (CFnCTL1.CFnCKS2 至 CFnCTL1.CFnCKS0 位 = 000), 传输数据长度= 8 位 (CFnCTL2.CFnCL3 至 CFnCTL2.CFnCL0 位= 0000)。

(1) 操作流程



(2) 操作时序



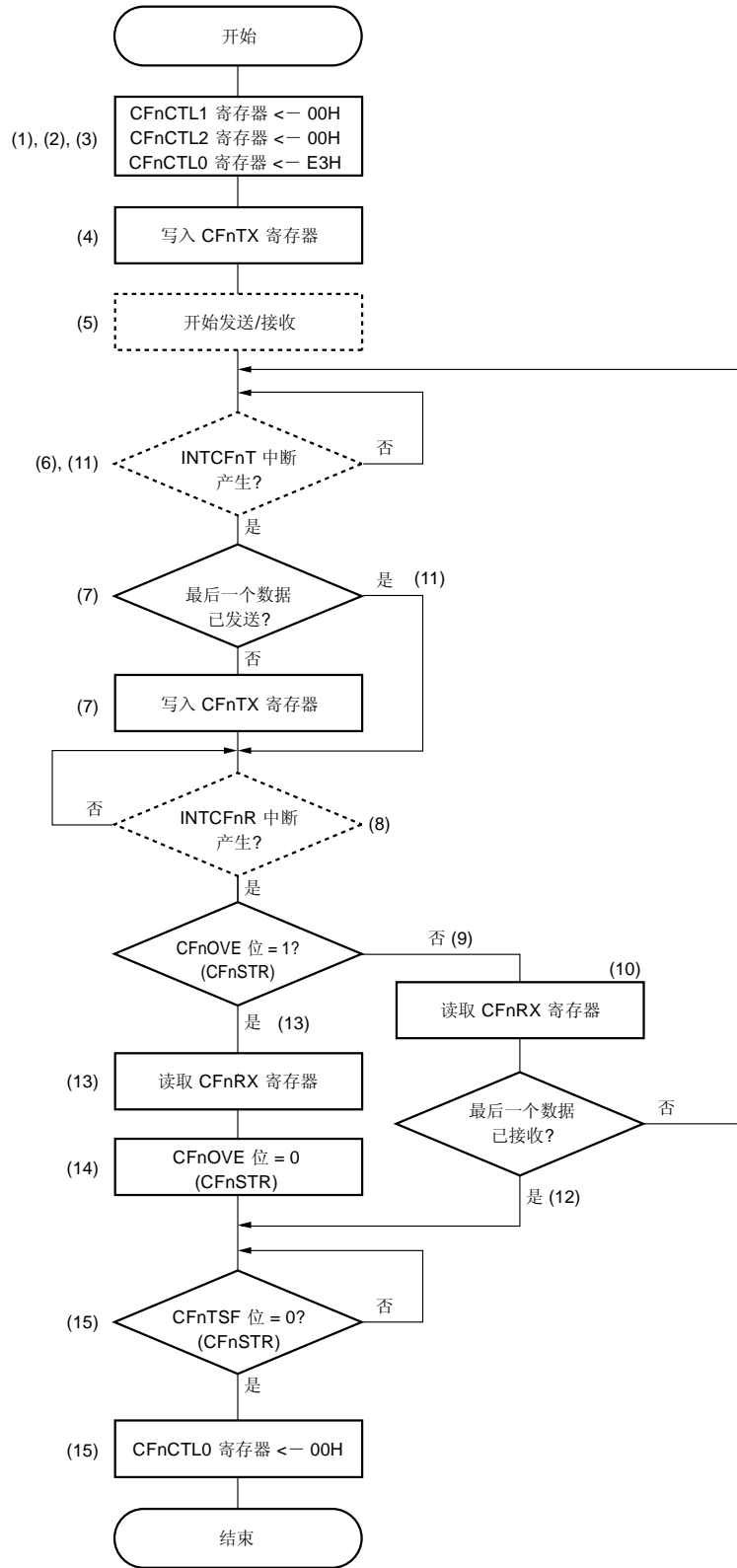
- (1) 向 CFnCTL1 寄存器写入 00H, 选择通信类型 1, 通信时钟 (fcCLK) = fxx/2, 并设置为主模式。
- (2) 向 CFnCTL2 寄存器写入 00H, 设定传输数据长度为 8 位。
- (3) 向 CFnCTL0 寄存器写入 A3H, 在使能通信时钟 (fcCLK) 操作的同时, 选择为接收模式、MSB 先行以及连续传输模式。
- (4) 对 CFnRX 寄存器执行一次伪读取, 使得 CFnSTR.CFnTSF 位置为 1, 并启动接收。
- (5) 当接收开始后, 输出串行时钟至 SCKFn 引脚, 并与串行时钟同步捕获 SIFn 引脚的接收数据。
- (6) 当接收完毕之后, 产生接收完成中断请求信号 (INTCFnR), 允许读取 CFnRX 寄存器。
- (7) 当通信结束时 CFnCTL0.CFnSCE 位= 1, 则在本次通信完成后立即开始下一次通信。
- (8) 若要结束当前连续接收传输, 写入 CFnSCE 位= 0。
- (9) 读取 CFnRX 寄存器。
- (10) 当接收完毕时, 产生 INTCFnR 信号, 并且允许读取 CFnRX 寄存器。如果在通信结束之前设置 CFnSCE 位= 0, 停止输出串行时钟至 SCKFn 引脚, 并清除 CFnTSF 为 0, 从而结束接收操作。
- (11) 读取 CFnRX 寄存器。
- (12) 如果发生溢出错误, 则 CFnSTR.CFnOVE 位= 0, 并且清除错误标志。
- (13) 若要解除接收使能状态, 在检查确认 CFnTSF 位= 0 之后, 设置 CFnCTL0.CFnPWR 位= 0 和 CFnCTL0.CFnRXE 位= 0。

备注 n = 0 至 4

18.6.9 连续传输模式 (主模式, 发送/接收模式)

MSB 先行 (CFnCTL0.CFnDIR 位= 0), 通信类型 1 (CFnCTL1.CFnCKP 和 CFnCTL1.CFnDAP 位= 00), 通信时钟 (f_{CLK}) = f_{xx}/2 或 f_{xx}/3 (CFnCTL1.CFnCKS2 至 CFnCTL1.CFnCKS0 位 = 000), 传输数据长度= 8 位 (CFnCTL2.CFnCL3 至 CFnCTL2.CFnCL0 位= 0000)。

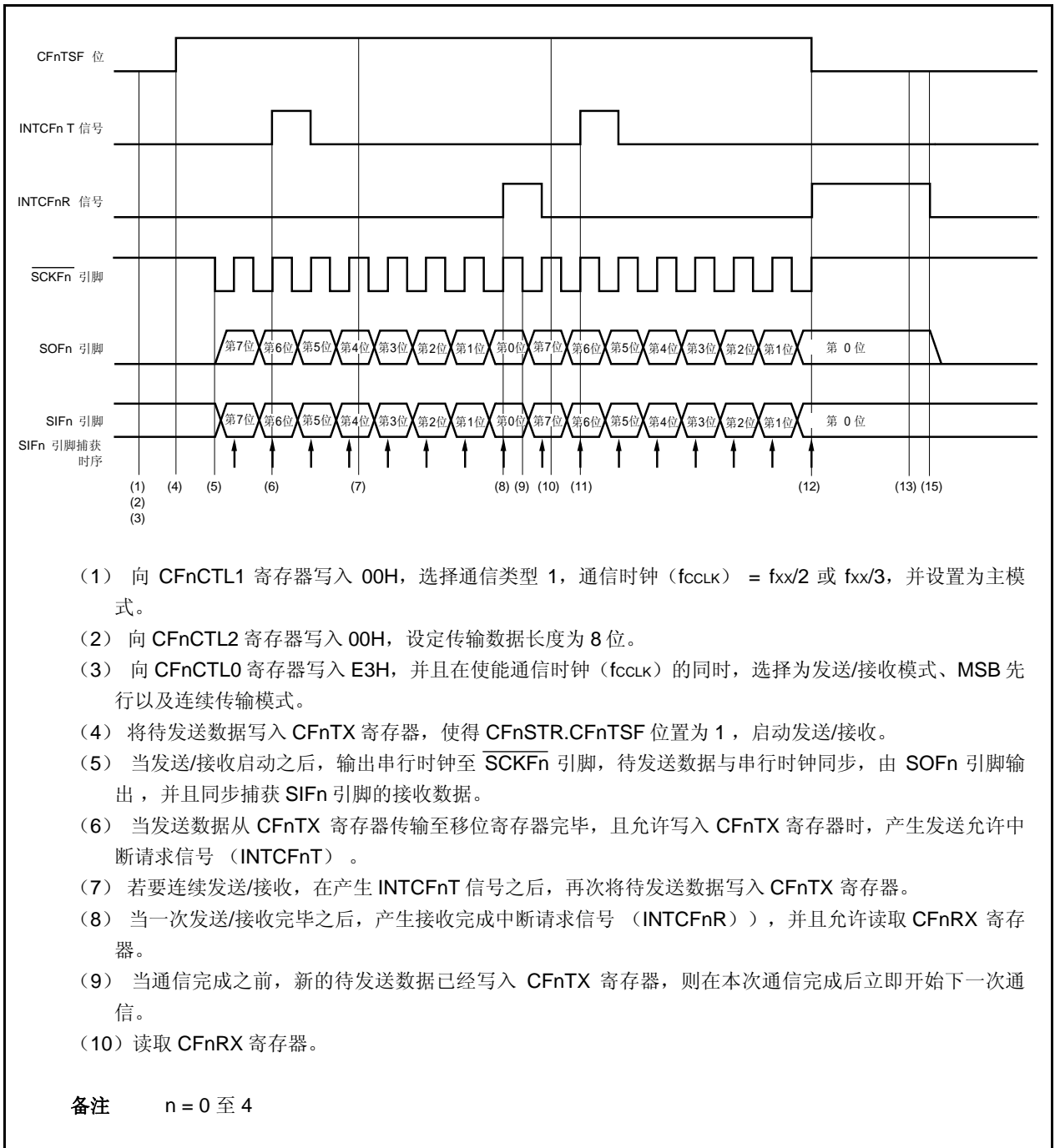
(1) 操作流程



- 备注
1. 虚线部分表示硬件处理。
 2. 图中的数字对应于 (2) 操作时序中的数字。
 3. n = 0 至 4

(2) 操作时序

(1/2)



- (1) 向 CFnCTL1 寄存器写入 00H, 选择通信类型 1, 通信时钟 (fcCLK) = fxx/2 或 fxx/3, 并设置为主模式。
- (2) 向 CFnCTL2 寄存器写入 00H, 设定传输数据长度为 8 位。
- (3) 向 CFnCTL0 寄存器写入 E3H, 并且在使能通信时钟 (fcCLK) 的同时, 选择为发送/接收模式、MSB 先行以及连续传输模式。
- (4) 将待发送数据写入 CFnTX 寄存器, 使得 CFnSTR.CFnTSMF 位置为 1, 启动发送/接收。
- (5) 当发送/接收启动之后, 输出串行时钟至 SCKFn 引脚, 待发送数据与串行时钟同步, 由 SOFn 引脚输出, 并且同步捕获 SIFn 引脚的接收数据。
- (6) 当发送数据从 CFnTX 寄存器传输至移位寄存器完毕, 且允许写入 CFnTX 寄存器时, 产生发送允许中断请求信号 (INTCFnT)。
- (7) 若要连续发送/接收, 在产生 INTCFnT 信号之后, 再次将待发送数据写入 CFnTX 寄存器。
- (8) 当一次发送/接收完毕之后, 产生接收完成中断请求信号 (INTCFnR), 并且允许读取 CFnRX 寄存器。
- (9) 当通信完成之前, 新的待发送数据已经写入 CFnTX 寄存器, 则在本次通信完成后立即开始下一次通信。
- (10) 读取 CFnRX 寄存器。

备注 n = 0 至 4

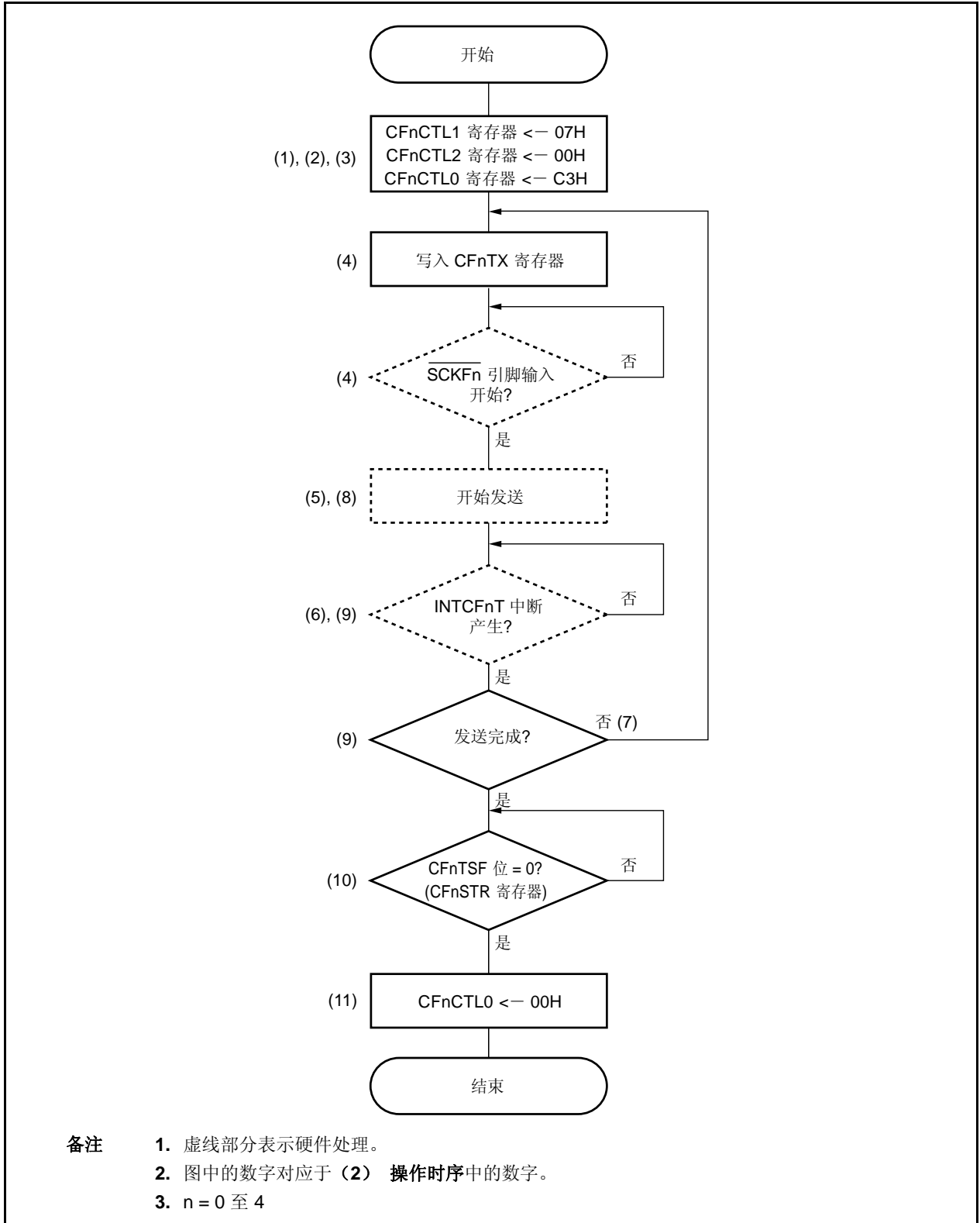
- (11) 从 CFnTX 寄存器传输发送数据至移位寄存器完毕，且产生 INTCFnT 信号。若要结束当前的发送/接收传输，则不写入 CFnTX 寄存器。
- (12) 当传输完毕之前下一个待发送数据未被写入 CFnTX 寄存器，传输完毕后，则停止输出串行时钟至 SCKFn 引脚，并将 CFnTSF 位清除为 0。
- (13) 发生接收错误中断请求信号 (INTCFnR) 时，读取 CFnRX 寄存器。
- (14) 如果发生溢出错误，则 CFnSTR.CFnOVE 位= 0，并且清除错误标志。
- (15) 若要解除发送/接收使能状态，在检查确认 CFnTSF 位= 0 之后，设置 CFnCTL0.CFnPWR 位= 0，CFnCTL0.CFnTXE 位= 0 以及 CFnCTL0.CFnRXE 位= 0。

备注 n = 0 至 4

18.6.10 连续传输模式 (从模式, 发送模式)

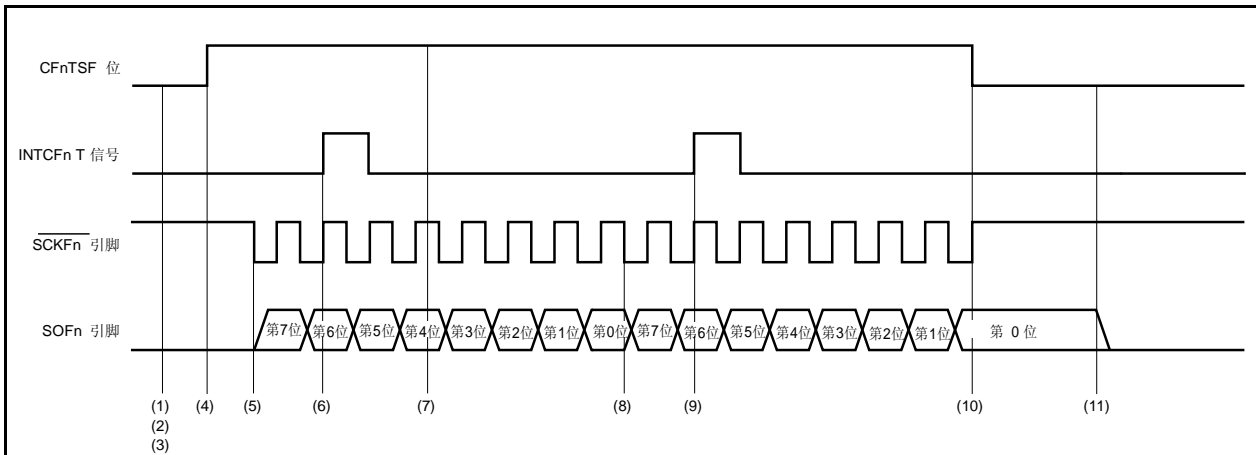
MSB 先行 (CFnCTL0.CFnDIR 位= 0), 通信类型 1 (CFnCTL1.CFnCKP 和 CFnCTL1.CFnDAP 位= 00), 通信时钟 (f_{CLK}) = 外部时钟 (SCKFn) (CFnCTL1.CFnCKS2 至 CFnCTL1.CFnCKS0 位 = 111), 传输数据长度= 8 位 (CFnCTL2.CFnCL3 至 CFnCTL2.CFnCL0 位= 0000)。

(1) 操作流程



- 备注
1. 虚线部分表示硬件处理。
 2. 图中的数字对应于 (2) 操作时序中的数字。
 3. n = 0 至 4

(2) 操作时序



- (1) 向 CFnCTL1 寄存器写入 07H，选择通信类型 1，通信时钟 (f_{CLK}) = 外部时钟 (SCKFn)，并设置为从模式。
- (2) 向 CFnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。
- (3) 向 CFnCTL0 寄存器写入 C3H，在使能通信时钟 (f_{CLK}) 操作的同时，选择为发送模式、MSB 先行以及连续传输模式。
- (4) 将待发送数据写入 CFnTX 寄存器，使得 CFnSTR.CFnTSMF 位为 1，设备等待串行时钟输入。
- (5) 当串行时钟输入时，待发送数据与串行时钟同步，由 SOFn 引脚输出。
- (6) 当发送数据从 CFnTX 寄存器传输至移位寄存器完毕，且允许写入 CFnTX 寄存器时，产生发送允许中断请求信号 (INTCFnT)。
- (7) 若要连续发送，在产生 INTCFnT 信号之后，再次将待发送数据写入 CFnTX 寄存器。
- (8) 当 CFnCTL2 寄存器设定的传输数据长度发送完毕后，有串行时钟输入，则启动连续发送。
- (9) 当发送数据从 CFnTX 寄存器传输至移位寄存器完毕，并且允许写入 CFnTX 寄存器，产生 INTCFnT 信号。若要结束当前的连续发送传输，不要写入 CFnTX 寄存器。
- (10) 当 CFnCTL2 寄存器设定的传输数据长度对应的时钟输入时，而未写入 CFnTX 寄存器，清除 CFnTSMF 位为 0，从而结束发送。
- (11) 若要解除发送使能状态，在检查确认 CFnTSMF 位= 0 之后，设置 CFnCTL0.CFnPWR 位= 0 和 CFnCTL0.CFnTXE 位= 0。

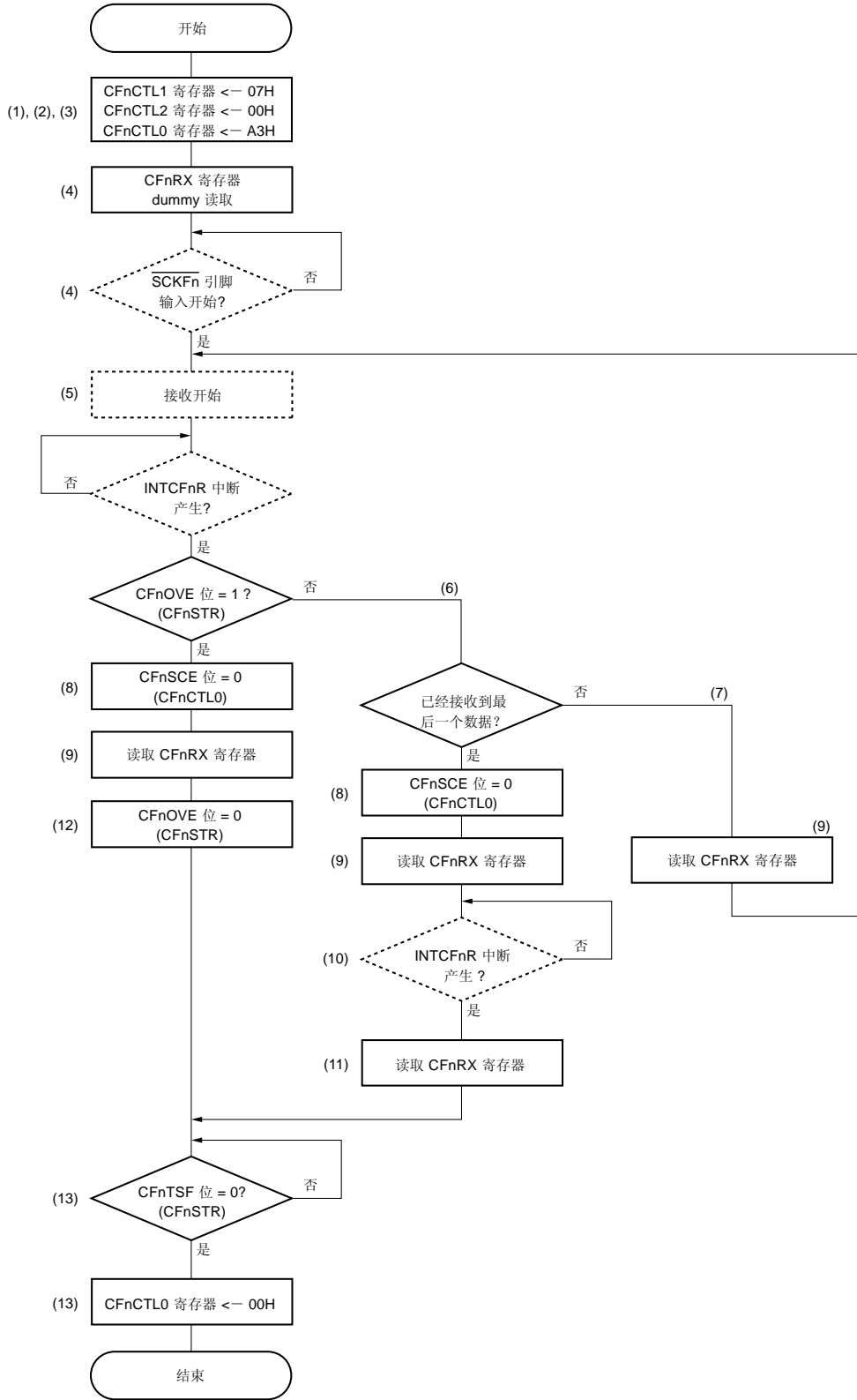
注意事项 连续发送模式下，不产生接收完成中断请求信号 (INTCFnR)。

备注 n = 0 至 4

18.6.11 连续传输模式（从模式，接收模式）

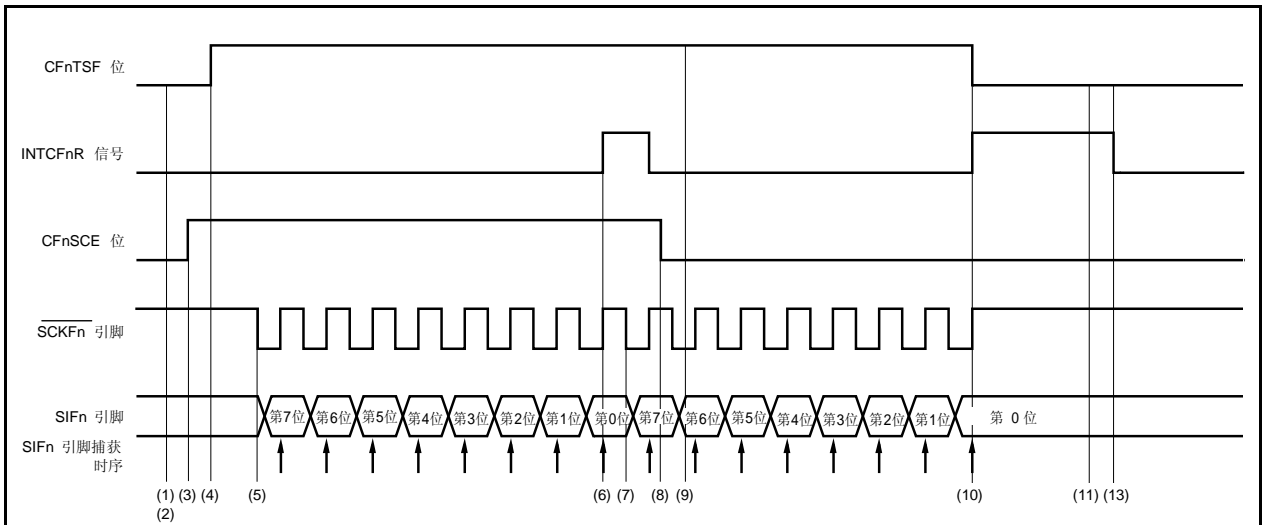
MSB 先行 (CFnCTL0.CFnDIR 位= 0)，通信类型 1 (CFnCTL1.CFnCKP 和 CFnCTL1.CFnDAP 位= 00)，通信时钟 (f_{CCLK}) = 外部时钟 ($\overline{\text{SCKFn}}$) (CFnCTL1.CFnCKS2 至 CFnCTL1.CFnCKS0 位 = 111)，传输数据长度= 8 位 (CFnCTL2.CFnCL3 至 CFnCTL2.CFnCL0 位= 0000)。

(1) 操作流程



- 备注
1. 虚线部分表示硬件处理。
 2. 图中的数字和 (2) 操作时序中的处理数字相对应。
 3. n = 0 至 4

(2) 操作时序



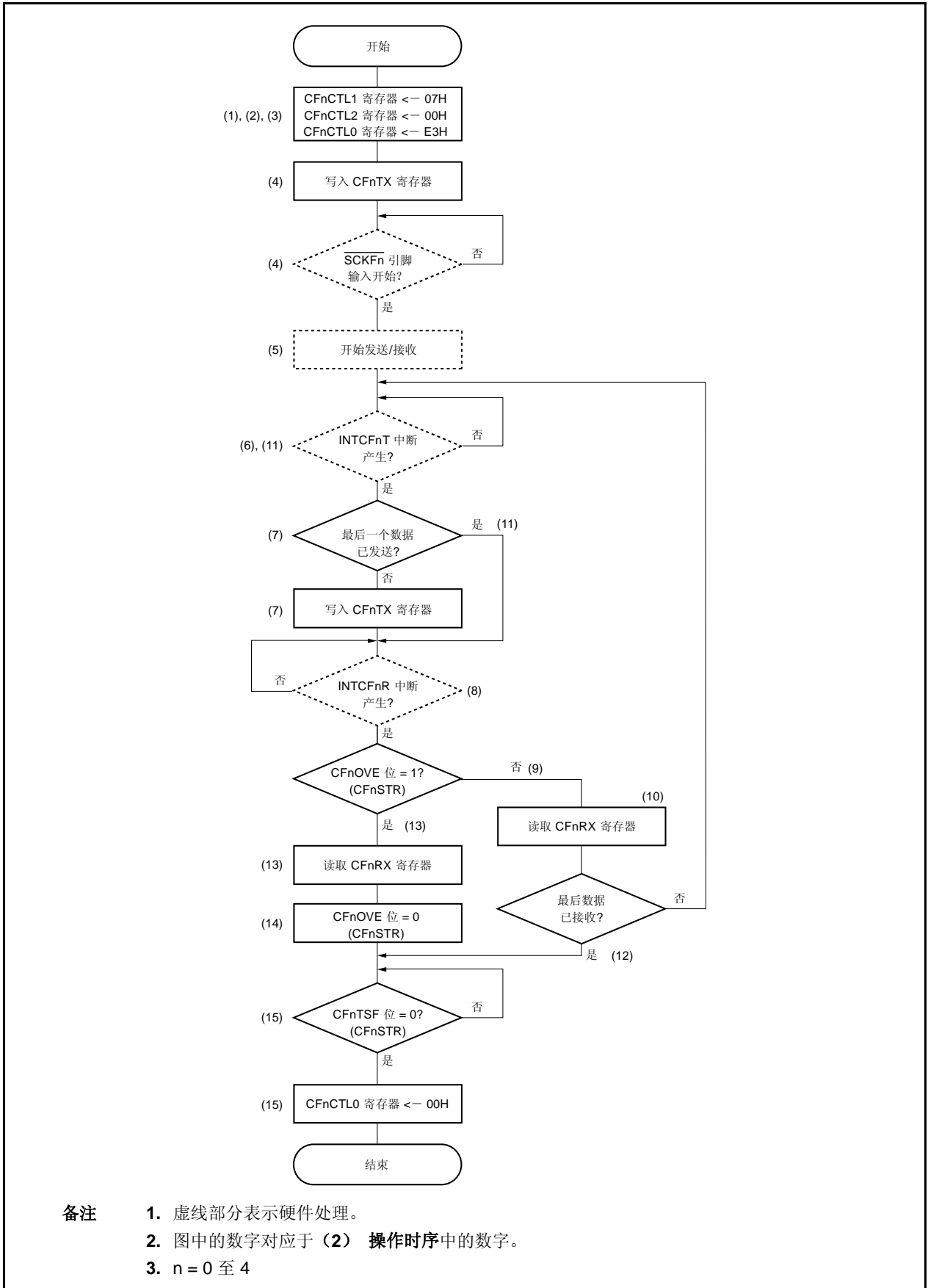
- (1) 向 CFnCTL1 寄存器写入 07H, 选择通信类型 1, 通信时钟 (f_{CLK}) = 外部时钟 (SCKFn), 并设置为从模式。
- (2) 向 CFnCTL2 寄存器写入 00H, 设定传输数据长度为 8 位。
- (3) 向 CFnCTL0 寄存器写入 A3H, 在使能通信时钟 (f_{CLK}) 操作的同时, 选择为接收模式、MSB 先行以及连续传输模式。
- (4) 通过对 CFnRX 寄存器进行伪读取, 使得 CFnSTR.CFnTSMF 位为 1, 设备等待串行时钟输入。
- (5) 当串行时钟输入时, 与串行时钟同步从 SIFn 引脚捕获接收数据。
- (6) 当接收完毕之后, 产生接收完成中断请求信号 (INTCFnR), 并且允许读取 CFnRX 寄存器。
- (7) 在 CFnCTL0.CFnSCE 位= 1 状态下输入串行时钟, 启动连续接收。
- (8) 若要结束当前连续接收传输, 写入 CFnSCE 位= 0。
- (9) 读取 CFnRX 寄存器。
- (10) 当接收完毕时, 产生 INTCFnR 信号, 并且允许读取 CFnRX 寄存器。如果在通信完毕之前设置 CFnSCE 位= 0, 则清除 CFnTSMF 位为 0, 从而结束操作。
- (11) 读取 CFnRX 寄存器。
- (12) 如果发生溢出错误, 则 CFnSTR.CFnOVE 位= 0, 并且清除错误标志。
- (13) 若要解除接收使能状态, 在检查确认 CFnTSMF 位= 0 之后, 设置 CFnCTL0.CFnPWR 位= 0 和 CFnCTL0.CFnRXE 位= 0。

备注 n = 0 至 4

18.6.12 连续传输模式 (从模式, 发送/接收模式)

MSB 先行 (CFnCTL0.CFnDIR 位= 0), 通信类型 1 (CFnCTL1.CFnCKP 和 CFnCTL1.CFnDAP 位= 00), 通信时钟 (f_{clock}) = 外部时钟 ($\overline{\text{SCKFn}}$) (CFnCTL1.CFnCKS2 至 CFnCTL1.CFnCKS0 位 = 111), 传输数据长度= 8 位 (CFnCTL2.CFnCL3 至 CFnCTL2.CFnCL0 位= 0000)。

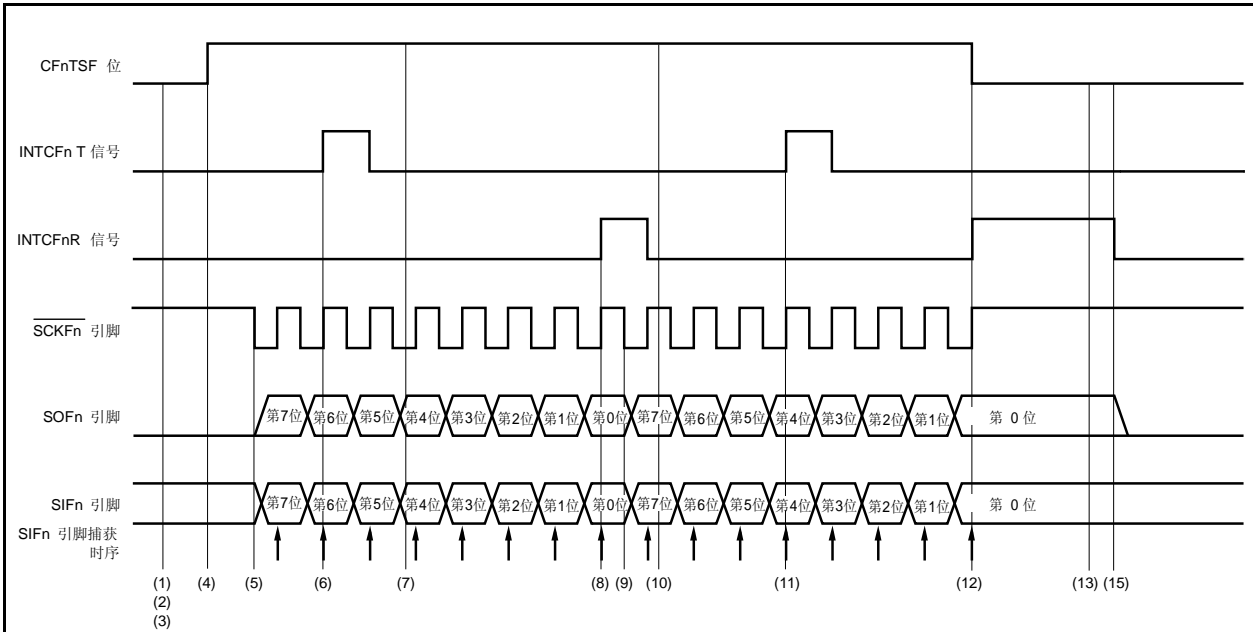
(1) 操作流程



- 备注
1. 虚线部分表示硬件处理。
 2. 图中的数字对应于 (2) 操作时序中的数字。
 3. n = 0 至 4

(2) 操作时序

(1/2)



- (1) 向 CnCTL1 寄存器写入 07H, 选择通信类型 1, 通信时钟 (fcCLK) = 外部时钟 (\overline{SCKFn}), 并设置为从模式。
- (2) 向 CnCTL2 寄存器写入 00H, 设定传输数据长度为 8 位。
- (3) 向 CnCTL0 寄存器写入 E3H, 在使能通信时钟 (fcCLK) 的同时, 选择为发送/接收模式、MSB 先行以及连续传输模式。
- (4) 将待发送数据写入 CnTX 寄存器, 使得 CnSTR.CnTnTSF 位置为 1, 设备等待串行时钟输入。
- (5) 当串行时钟输入时, 与串行时钟同步, 输出发送数据至 SOFn 引脚, 并且捕获 SIFn 引脚的接收数据。
- (6) 当发送数据从 CnTX 寄存器传输至移位寄存器完毕, 且允许写入 CnTX 寄存器时, 产生发送允许中断请求信号 (INTCFnT)。
- (7) 若要连续发送, 在产生 INTCFnT 信号之后, 再次将发送数据写入 CnTX 寄存器。
- (8) 当 CnCTL2 寄存器设定的传输数据长度接收完毕后, 产生接收完成中断请求信号 (INTCFnR), 并且允许读取 CnRX 寄存器。
- (9) 当串行时钟连续输入, 启动连续发送/接收。
- (10) 读取 CnRX 寄存器。
- (11) 当发送数据从 CnTX 寄存器至移位寄存器传输完毕, 并且允许写入 CnTX 寄存器, 产生 INTCFnT 信号。若要结束当前的发送/接收传输, 则不写入 CnTX 寄存器。

备注 n = 0 至 4

- (12) 当 CFnCTL2 设定的传输数据长度相应的时钟输入时，而未写入 CFnTX 寄存器，则产生 INTCFnR 信号。将 CFnTSF 位清除为 0，从而结束发送/接收。
- (13) 产生 INTCFnR 信号时，读取 CFnRX 寄存器。
- (14) 如果发生溢出错误，则 CFnSTR.CFnOVE 位= 0，并且清除错误标志。
- (15) 若要解除发送/接收使能状态，在检查确认 CFnTSF 位= 0 之后，设置 CFnCTL0.CFnPWR 位= 0，CFnCTL0.CFnTXE 位= 0 以及 CFnCTL0.CFnRXE 位= 0。

备注 n = 0 至 4

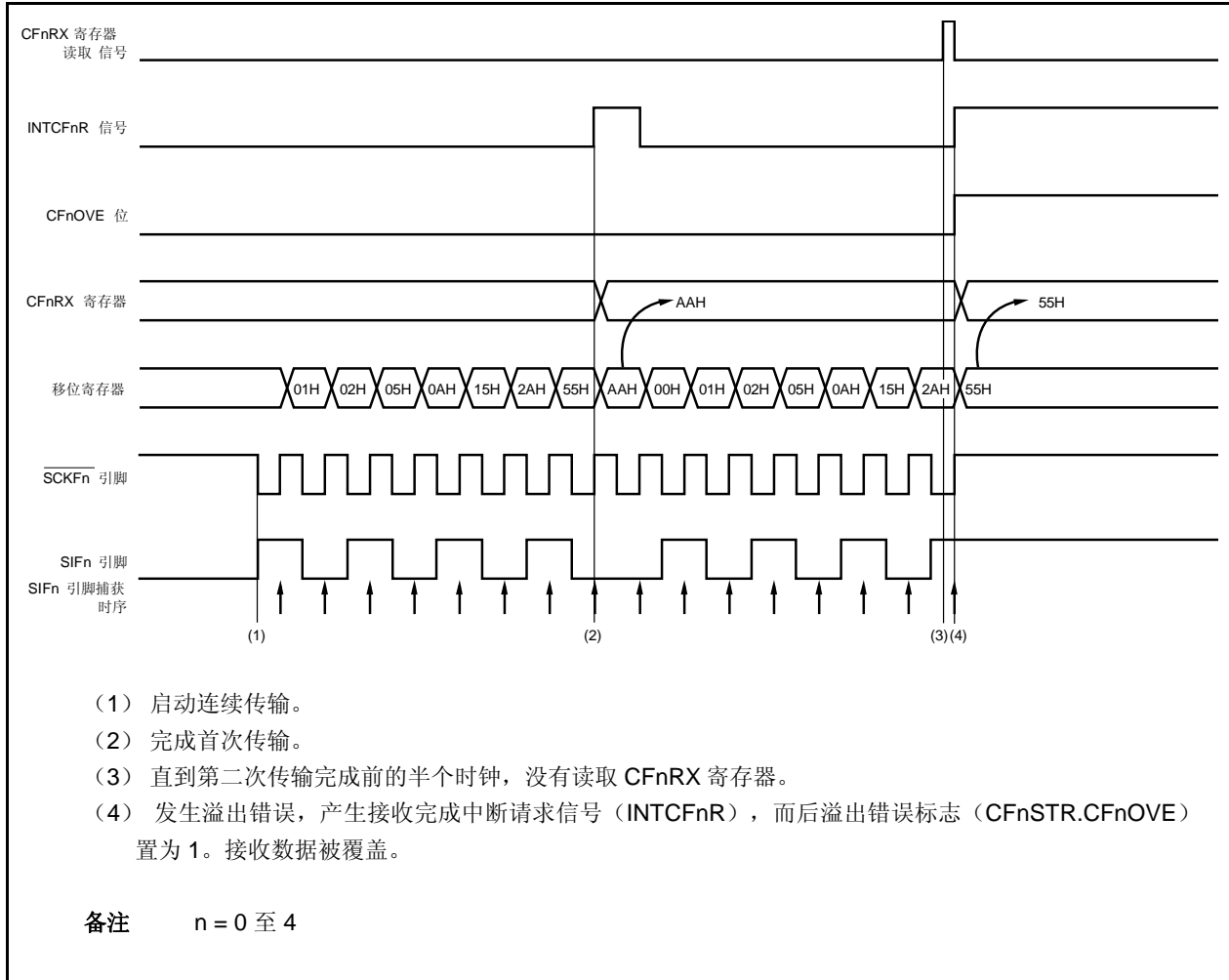
18.6.13 接收错误

连续传输模式下，执行传输时允许接收 (CFnCTL0.CFnRXE 位= 1)，INTCFnR 信号产生后，在读取 CFnRX 寄存器之前完成下一次接收操作时，则再次产生接收完成中断请求信号 (INTCFnR)，并且溢出错误标志 (CFnSTR.CFnOVE) 被置为 1。

即使发生了溢出错误，由于 CFnRX 已经被更新，先前接收到的数据丢失。即使发生了接收错误，如果在读取 CFnRX 寄存器之前下次接收完毕，则再次产生 INTCFnR 信号。

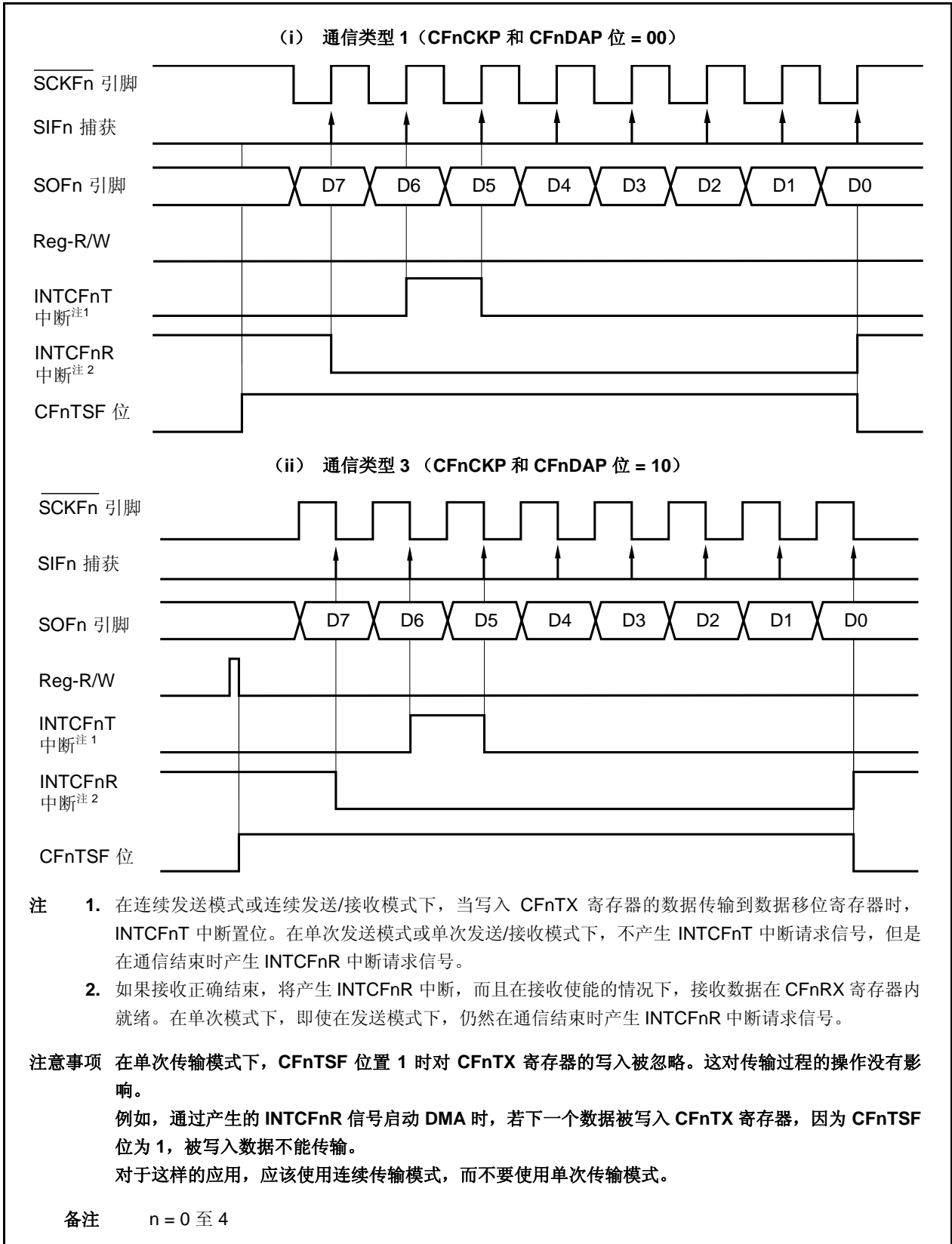
为了避免发生溢出错误，从产生 INTCFnR 信号到采样下次接收最末位数据之前的半个时钟，完成对 CFnRX 寄存器的读取。

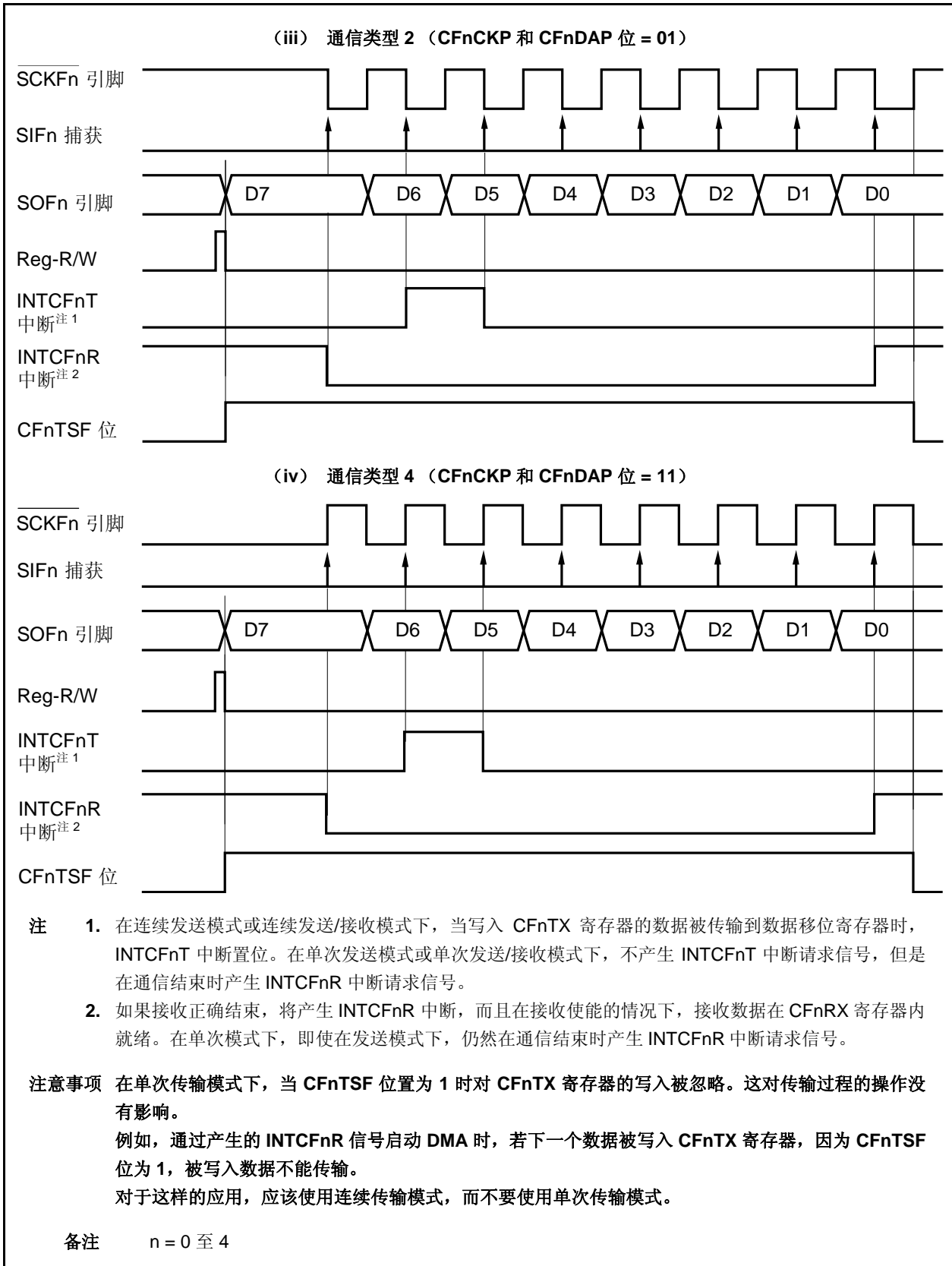
(1) 操作时序



18.6.14 时钟时序

(1/2)





18.7 输出引脚

(1) $\overline{\text{SCKFn}}$ 引脚

当 CSIFn 操作禁止 (CFnCTL0.CFnPWR 位=0) 时, $\overline{\text{SCKFn}}$ 引脚输出状态如下所示。

CFnCKP	CFnCKS2	CFnCKS1	CFnCKS0	$\overline{\text{SCKFn}}$ 引脚输出
0	1	1	1	高阻抗状态
	其它			固定为高电平
1	1	1	1	高阻抗状态
	其它			固定为低电平

- 备注**
1. 如果 CFnCTL1.CFnCKP 和 CFnCKS2 至 CFnCKS0 位中有任何位被重写, $\overline{\text{SCKFn}}$ 引脚的输出电平改变。
 2. n = 0 至 4

(2) SOFn 引脚

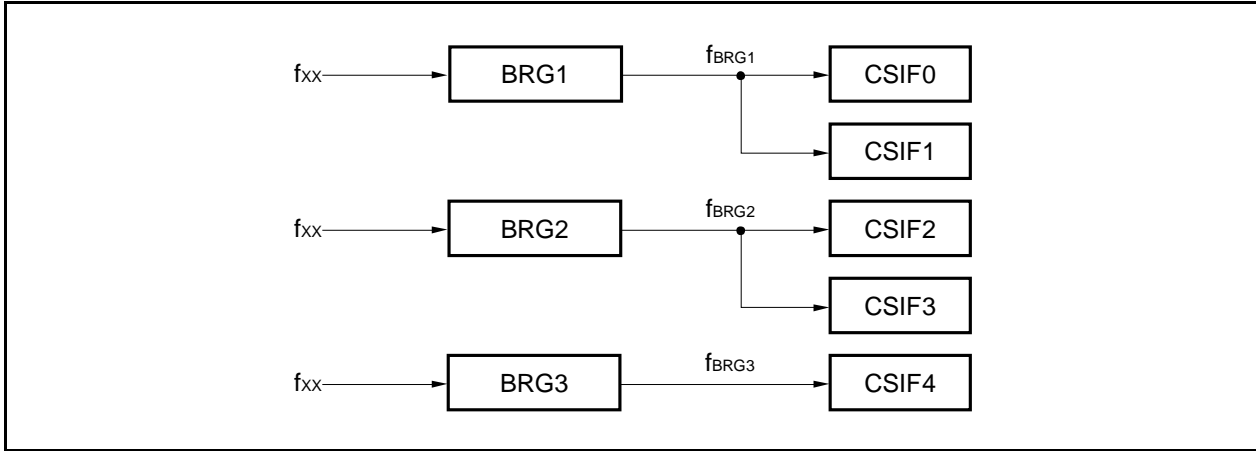
当 CSIFn 操作禁止 (CFnPWR 位=0) 时, SOFn 引脚输出状态如下所示。

CFnTXE	CFnDAP	CFnDIR	SOFn 引脚输出
0	×	×	固定为低电平
1	0	×	SOFn 锁存值 (低电平)。
	1	0	CFnTX0 值 (MSB)
		1	CFnTX0 值 (LSB)

- 备注**
1. 如果 CFnCTL0.CFnTXE 位, CFnCTL0.CFnDIR 位或 CFnCTL1.CFnDAP 位中有任何一位被重写, SOFn 引脚输出会改变。
 2. ×: 无需理会
 3. n = 0 至 4

16.8 波特率发生器

BRG1 至 BRG3 和 CSIF0 至 CSIF4 波特率发生器的连接如下所示：



(1) 预分频模式寄存器 1 至 3 (PRSM1 至 PRSM3)

PRSM1 至 PRSM3 寄存器用于控制 CSIF 波特率信号的发生。

这些寄存器可以按字节或按位进行读写。

系统复位后，这些寄存器被设置为 00H。

复位后：00H R/W 地址： PRSM1 FFFFF320H, PRSM2 FFFFF324H,
PRSM3 FFFFF328H

	7	6	5	<4>	3	2	1	0
PRSMm (m = 1 至 3)	0	0	0	BGCEm	0	0	BGCSm1	BGCSm0

BGCEm	波特率输出
0	禁止
1	使能

BGCSm1	BGCSm0	输入时钟选择 (fBGCSm)	设置值 (k)
0	0	fxx/2	0
0	1	fxx/4	1
1	0	fxx/8	2
1	1	fxx/16	3

- 注意事项**
1. 在操作期间，禁止重写 PRSMm 寄存器。
 2. 在设置 BGCEm 位为 1 之前，设置 PRSMm 寄存器。
 3. 请确保将第 7 位至第 5 位，第 3 位和第 2 位清 “0”。

(2) 预分频比较寄存器 1 至 3 (PRSCM1 至 PRSCM3)

PRSCM1 至 PRSCM3 寄存器是 8 位比较寄存器。

这些寄存器可以按字节进行读取或写入。

系统复位后，这些寄存器被设置为 00H。

**18.8.1 波特率的产生**

通过对主时钟的分频产生发送/接收时钟。从主时钟产生的波特率由以下公式计算获得：

$$f_{BRGm} = \frac{f_{xx}}{2^{k+1} \times N}$$

注意事项 设置 f_{BRGm} 为 8 MHz (CSIF0 至 CSIF2 和 CSIF4) 或更低，12 MHz (CSIF3) 或更低。

备注 f_{BRGm} : BRGm 计数时钟

f_{xx} : 主时钟振荡频率

k: PRSCMm 寄存器设定值 = 0 至 3

N: PRSCMm 寄存器设定值 = 1 至 256

但是仅当 PRSCMm 寄存器设置为 00H 时，N = 256。

m = 1 至 3

18.9 注意事项

- (1) 使用 DMA 传输发送数据和接收数据时，即使串行传输期间发生溢出错误，也不执行错误处理。DMA 传输完成后通过读取 CFnSTR.CFnOVE 位来检查是否发生溢出错误。
- (2) 关于操作期间 (CFnCTL0.CFnPWR 位 = 1) 禁止重写的寄存器，如果操作期间进行了错误的重写，须将 CFnCTL0.CFnPWR 位清 0，然后初始化 CSIFn。

操作期间禁止重写的寄存器列举如下：

- CFnCTL0 寄存器：CFnTXE, CFnRXE, CFnDIR, CFnTMS 位
- CFnCTL1 寄存器：CFnCKP, CFnDAP, CFnCKS2 至 CFnCKS0 位
- CFnCTL2 寄存器：CFnCL3 至 CFnCL0 位

- (3) 通信类型 2 或 4 (CFnCTL1.CFnDAP 位 = 1) 中，在发生接收完成中断 (INTCFnR) 后半个 $\overline{\text{SCKFn}}$ 时钟时，CFnSTR.CFnTSF 位被清除。

单次传输模式下，通信期间 (CFnTSF 位 = 1) 写入的下一个待发送数据被忽略，并且不启动下次通信。同样如果设置为只接收通信 (CFnCTL0.CFnTXE 位 = 0, CFnCTL0.CFnRXE 位 = 1)，即使在通信期间读取接收数据，也不启动下一次通信。

因此，在通信类型 2 或 4 (CBnDAPCFnDAP 位 = 1) 时使用单次传输模式，需特别注意以下情况：

- 若要启动下一次发送，在确认 CFnTSF 位 = 0 之后，将待发送数据写入 CFnTX 寄存器。
- 设置为只接收通信 (CFnTXE 位 = 0, CFnRXE 位 = 1) 时，若要连续执行下一次接收，在确认 CFnTSF 位 = 0 之后，读取 CFnRX 寄存器。

或者使用连续传输模式来代替单次传输模式。尤其在用 DMA 传输时，建议使用连续传输模式。

备注 n = 0 至 4

第十九章 I²C 总线

要使用 I²C 总线功能，需设置 P36/SCL00，P37/SDA00，P40/SDA01，P41/SCL01，P90/SDA02 和 P91/SCL02 引脚作为复用引脚，并且将它们设置为 N 沟道漏极开路输出。

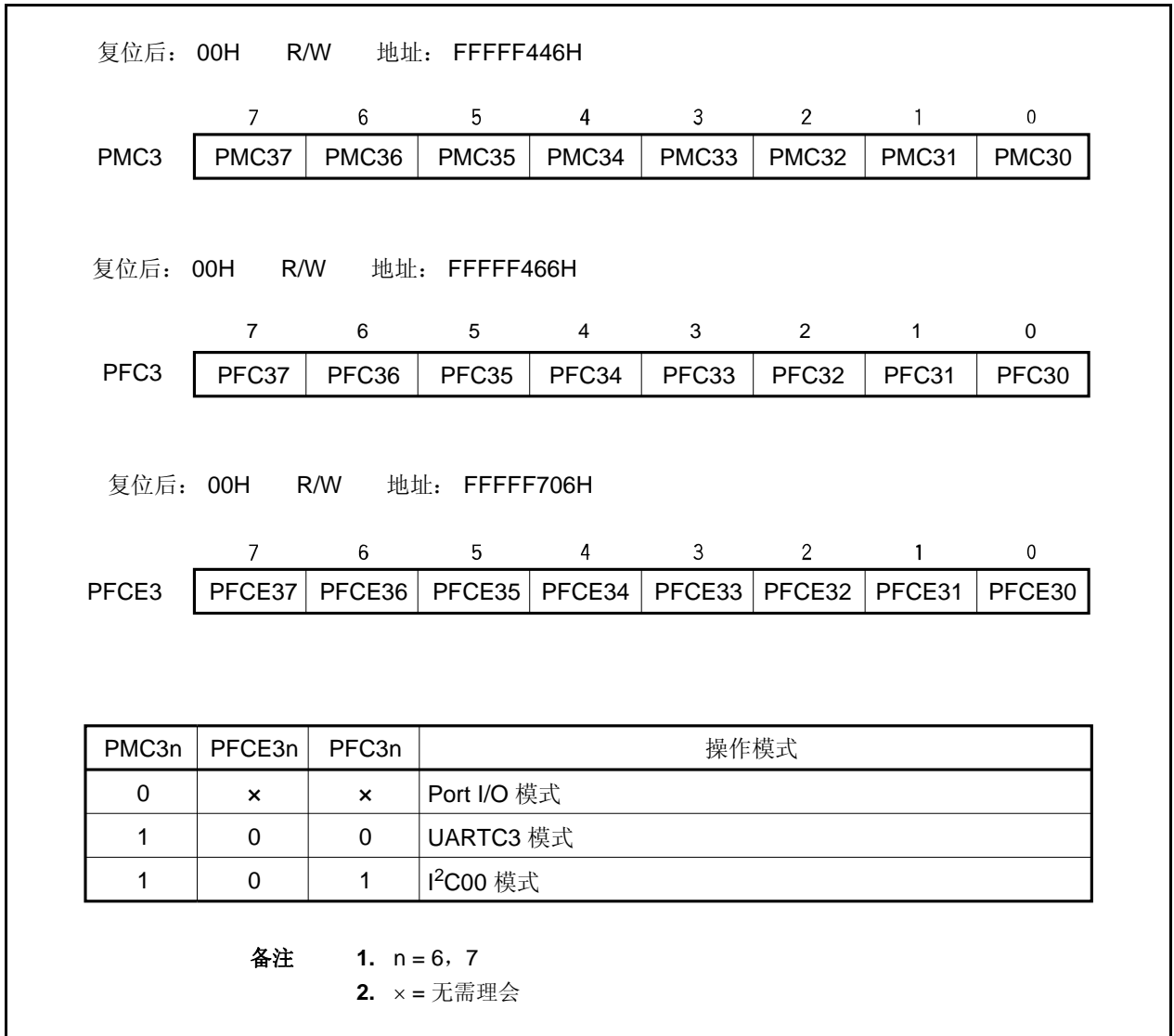
19.1 I²C 总线和其它串行接口的模式转换

19.1.1 UARTC3 和 I²C00 模式转换

V850ES/JG3-U 和 V850ES/JH3-U 单片机中，UARTC3 和 I²C00 是相同引脚的复用功能，因此不能同时使用。使用前，通过 PMC3、PFC3 和 PFCE3 寄存器预先设置 UARTC3 和 I²C00 之间的切换。

注意事项 如果发送或接收期间切换这些功能，则 UARTC3 和 I²C00 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 19-1. UARTC3 和 I²C00 模式转换设置



19.1.2 UARTC4、CSIF0 和 I²C01 模式转换

V850ES/JG3-U 和 V850ES/JH3-U 单片机中，UARTC4、CSIF0 和 I²C01 是相同引脚的复用功能，因此不能同时使用。使用前，通过 PMC4、PFC4 和 PFCE4 寄存器预先设置 UARTC4、CSIF0 和 I²C01 之间的切换。

注意事项 如果发送或接收期间切换这些功能，则 UARTC4、CSIF0 和 I²C01 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 19-2. UARTC4、CSIF0 和 I²C01 模式转换设置

复位后：00H R/W 地址： FFFFFFF448H

7	6	5	4	3	2	1	0
0	0	0	0	0	PMC42	PMC41	PMC40

复位后：00H R/W 地址： FFFFFFF468H

7	6	5	4	3	2	1	0
0	0	0	0	0	PFC42	PFC41	PFC40

复位后：00H R/W 地址： FFFFFFF708H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	PFCE41	PFCE40

PMC4n	PFC4n	PFCE4n	操作模式
0	×	×	Port I/O 模式
1	0	0	CSIF0 模式
1	0	1	I ² C01 模式
1	1	0	UARTC4 模式

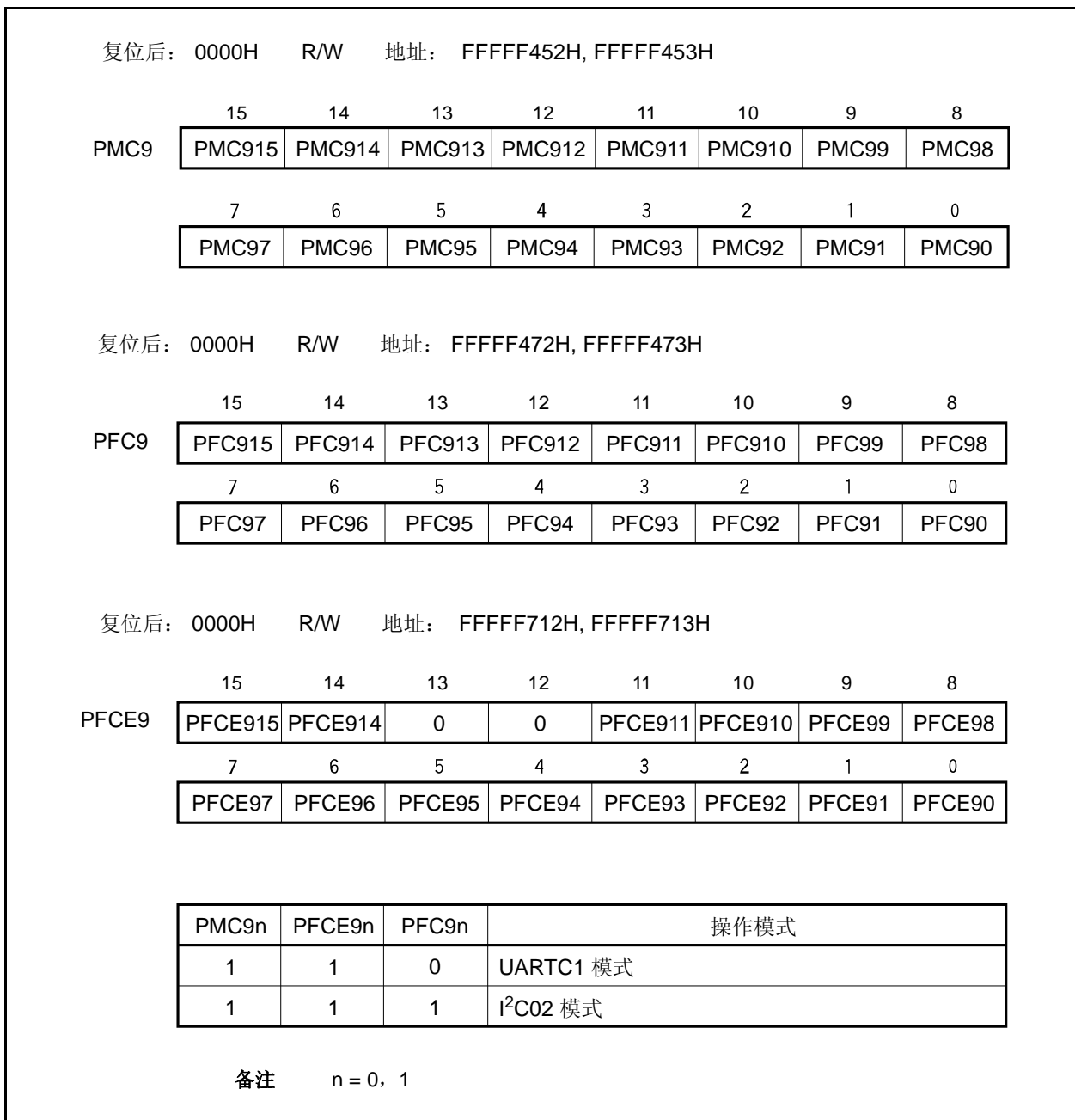
备注 1. n = 0, 1
 2. × = 无需理会

19.1.3 UARTC1 和 I²C02 模式转换

V850ES/JG3-U 和 V850ES/JH3-U 单片机中, UARTC1 和 I²C02 是相同引脚的复用功能, 因此不能同时使用。使用前, 通过 PMC9、PFC9 和 PFCE9 寄存器预先设置 UARTC1 和 I²C02 之间的切换。

注意事项 如果发送或接收期间切换这些功能, 则 UARTC1 和 I²C02 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 19-3. UARTC1 和 I²C02 模式转换设置



19.2 特性

I²C00 至 I²C02 具有下列两种模式：

- 操作停止模式
- I²C (IC 内部) 总线模式 (支持多主设备)

(1) 操作停止模式

在该模式下，不执行串行传输，因此可以降低功耗。

(2) I²C 总线模式 (支持多主设备)

该模式通过两条信号线在多个设备之间进行 8 位数据传输：一个串行时钟引脚 (SCL0n) 和一个串行数据总线引脚 (SDA0n)。

该模式符合 I²C 总线规范格式，主设备可以通过其串行数据总线，为从设备产生“开始条件”、“地址”、“传输方向规范”、“传输”和“停止条件”等数据。从设备通过硬件自动检测接收到的状态和数据。这种功能可以使控制 I²C 总线的应用程序得到简化。

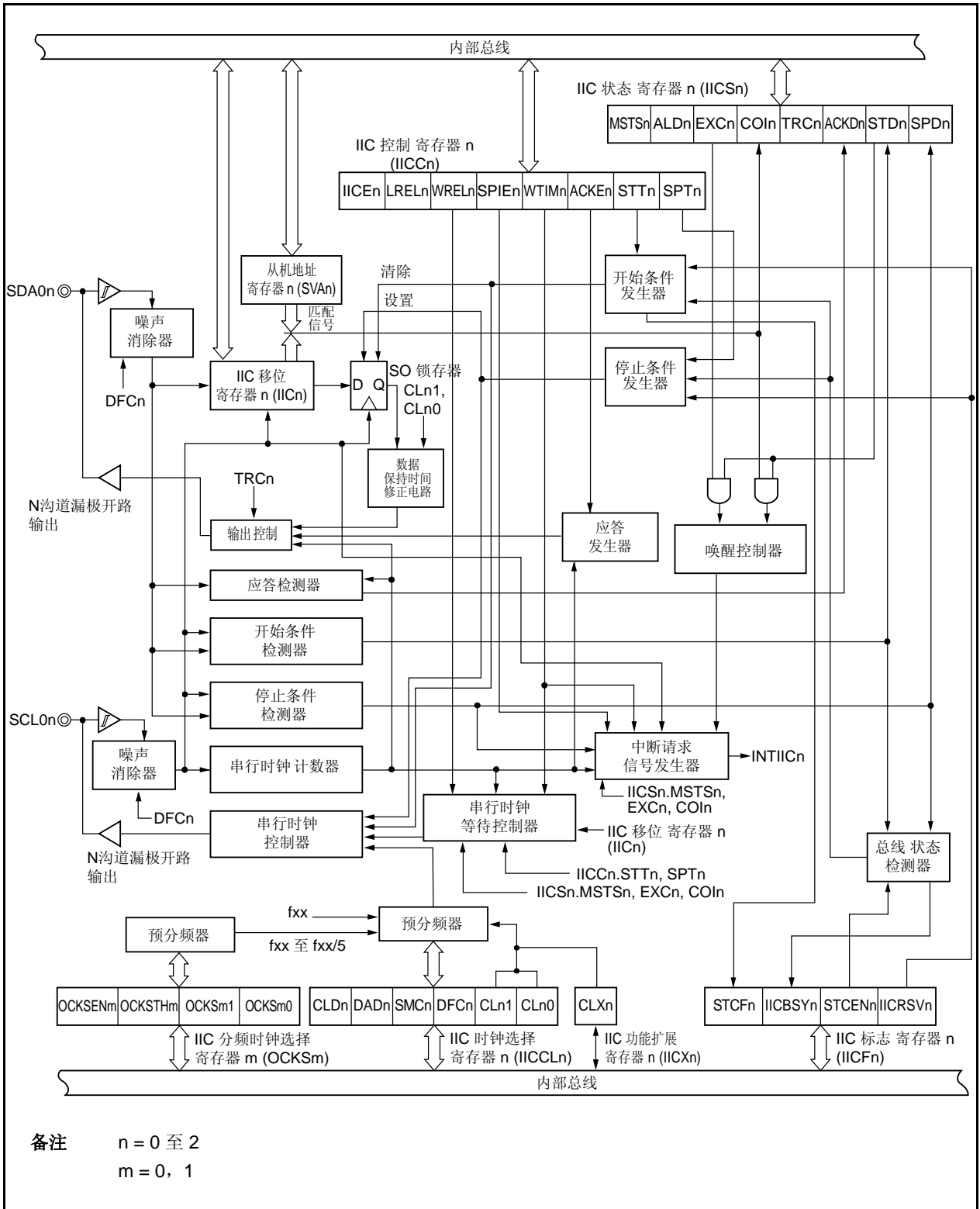
由于 SCL0n 和 SDA0n 引脚被用于 N 沟道漏极开路输出，因此 I²C0n 的串行时钟线和串行数据线都需要上拉电阻。

备注 n = 0 至 2

19.3 配置

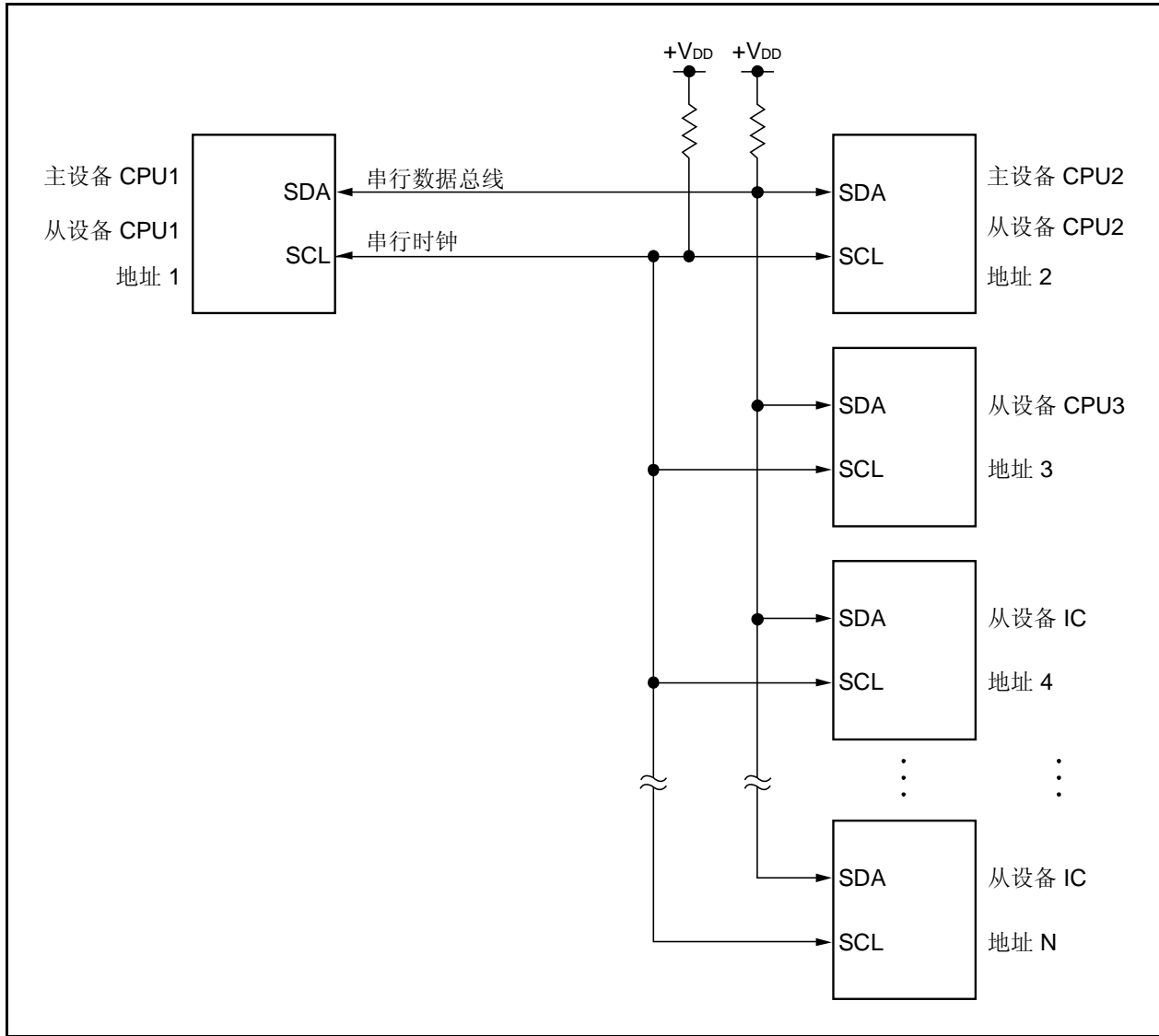
I²C0n 的功能框图如下所示:

图 19-4. I²C0n 的框图



以下展示了一个串行总线结构配置示例。

图 19-5. 使用 I²C 总线的串行总线配置示例



I²C0n 包括下列硬件 (n = 0 至 2)。

表 19-1. I²C0n 的配置

项目	配置
寄存器	IIC 移位寄存器 n (IICn) 从设备地址寄存器 n (SVAn)
控制寄存器	IIC 控制寄存器 n (IICCN) IIC 状态寄存器 n (IICSN) IIC 标志寄存器 n (IICF0n) IIC 时钟选择寄存器 n (IICCLn) IIC 功能扩展寄存器 n (IICXn) IIC 时钟分频选择寄存器 0, 1 (OCKS0, OCKS1)

(1) IIC 移位寄存器 n (IICn)

IICn 寄存器将 8 位串行数据转换为 8 位并行数据，也可将 8 位并行数据转换为 8 为串行数据，可用于发送和接收 (n = 0 至 2)。

对 IICn 寄存器的写入和读取操作被用来控制实际的发送和接收操作。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被置为 00H。

(2) 从设备地址寄存器 n (SVAn)

在从模式下，SVAn 寄存器用于设定本地地址 (n = 0 至 2)。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被置为 00H。

(3) SO 锁存器

SO 锁存器用于保持 SDA0n 引脚的输出电平 (n = 0 至 2)。

(4) 唤醒控制器

该寄存器收到的地址和 SVAn 寄存器设定的地址相匹配时，或接收到扩展码时，该电路就会产生一个中断请求信号 (INTIICn) (n = 0 至 2)。

(5) 预分频器

用于选择使用的采样时钟。

(6) 串行时钟计数器

在发送/接收操作过程中，串行时钟计数器对输出或输入的串行时钟进行计数，且用于验证发送或接收到的 8 位数据。

(7) 中断请求信号发生器

该电路控制中断请求信号 (INTIICn) 的产生。

以下两种触发都可以产生一个 I²C 中断。

- 串行时钟的第 8 个或第 9 个 (由 IICn.WTIMn 位设定) 时钟的下降沿。
- 因为检测到停止条件 (由 IICn.SPIEn 位设定) 而发生中断。

备注 n = 0 至 2

(8) 串行时钟控制器

在主模式下, 该电路从采样时钟生成时钟并经由 SCL0n 引脚输出 (n = 0 至 2)。

(9) 串行时钟等待控制器

该电路控制等待时序。

(10) $\overline{\text{ACK}}$ 发生器、停止条件检测器、开始条件检测器以及 $\overline{\text{ACK}}$ 检测器

这些电路用来产生和检测各种状态。

(11) 数据保持时间修正电路

该电路根据相应的 SCL0n 引脚下降沿, 为数据产生保持时间。

(12) 开始条件发生器

当 IICn.STTn 位被置位时, 该电路产生一个开始条件。

但是, 在通信保留禁止状态 (IICFn.IICRSVn 位 = 1) 下, 如果总线未释放 (IICFn.IICBSYn 位 = 1), 则请求被忽略, 且 IICFn.STCFn 位被置为 1。

(13) 停止条件发生器

当 IICn.SPTn 位被置位时, 该电路产生一个停止条件。

(14) 总线状态检测器

通过检测开始条件和停止条件, 该电路检测总线是否被释放。

但是, 总线状态不能在操作后立即被检测到, 因此使用 IICFn.STCENn 位将总线状态检测器设置为初始状态。

19.4 寄存器

I²C00 至 I²C02 由下列寄存器进行控制：

- IIC 控制寄存器 0 至 2 (IICC0 至 IICC2)
- IIC 状态寄存器 0 至 2 (IICS0 至 IICS2)
- IIC 标志寄存器 0 至 2 (IICF0 至 IICF2)
- IIC 时钟选择寄存器 0 至 2 (IICCL0 至 IICCL2)
- IIC 功能扩展寄存器 0 至 2 (IICX0 至 IICX2)
- IIC 时钟分频选择寄存器 0, 1 (OCKS0, OCKS1)

以下寄存器也可以使用：

- IIC 移位寄存器 0 至 2 (IIC0 至 IIC2)
- 从设备地址寄存器 0 至 2 (SVA0 至 SVA2)

备注 关于复用功能引脚的设置，参见表 4-20 当端口引脚用作复用功能时。

(1) IIC 控制寄存器 0 至 2 (IICC0 至 IICC2)

IICn 寄存器使能/停止 I²C0n 操作，设置等待时间并设置其它 I²C 操作 (n = 0 至 2)。

这些寄存器可以按字节或按位进行读取和写入。但是，当 IICEn 位为 0 时或在等待期间，对 SPIEn、WTIMn 和 ACKEn 位置位。当 IICEn 位由“0”改变为“1”时，这些位也可以被同时置位。

系统复位后，这些寄存器被置为 00H。

复位后：00H R/W 地址：IICC0 FFFFFFFD82H, IICC1 FFFFFFFD92H, IICC2 FFFFFFFDA2H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICn	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

(n = 0 至 2)

IICEn	I ² Cn 操作允许/禁止的说明
0	操作停止。IICSn 寄存器复位 ¹ 。内部操作停止。
1	允许操作。
SCL0n 和 SDA0n 信号线是高电平时，请确保将该位设置为 1。	
清零条件 (IICEn 位 = 0)	置位条件 (IICEn 位 = 1)
<ul style="list-style-type: none"> 由指令清除。 复位后。 	<ul style="list-style-type: none"> 由指令设置。

LRELn ²	退出通信
0	正常操作
1	<p>从当前通信操作退出，并设置为待机模式。在执行之后该设置被自动清除。当接收到本地无关扩展码的情况下也可以使用。</p> <p>SCL0n 和 SDA0n 信号线被设为高阻状态。</p> <p>STTn 和 SPTn 位以及 IICSn 寄存器的 MSTSn, EXCn, COIn, TRCn, ACKDn 和 STDn 位被清零。</p>
从通信中退出之后的待机状态一直保持有效，直到遇到下面的通信入口条件。	
<ul style="list-style-type: none"> 检测到停止条件后，在主模式下重新开始。 在开始条件以后，发生地址匹配或接收到扩展码。 	
清零条件 (LRELn 位 = 0)	置位条件 (LRELn 位 = 1)
<ul style="list-style-type: none"> 执行后自动清除。 复位后。 	<ul style="list-style-type: none"> 由指令设置。

WRELn ²	等待状态取消控制
0	等待状态不能取消。
1	等待状态被取消。等待取消以后，该设置被自动清除。
清零条件 (WRELn 位 = 0)	置位条件 (WRELn 位 = 1)
<ul style="list-style-type: none"> 执行后自动清除。 复位后。 	<ul style="list-style-type: none"> 由指令设置。

注 1. IICSn 寄存器、IICFn.STCFn 和 IICFn.IICBSYn 位、IICCLn.CLDn 位和 IICCLn.DADn 位被重置。
 2. 当 IICEn 位 = 0 时，该标志信号无效。

注意事项 当 SCL0n 信号线为高电平且 SDA0n 信号线为低电平时，如果 I²Cn 操作使能 (IICEn 位 = 1)，则立即检测到开始条件。为了避免这种情况，I²Cn 操作使能以后，立刻使用位处理指令将 LRELn 位设置为 1。

备注 数据被设置后进行读取时，LRELn 和 WRELn 位为 0。

SPIEn [‡]	当检测到停止条件时，使能/禁止产生中断请求	
0	禁止	
1	使能	
清零条件 (SPIEn 位 = 0)		置位条件 (SPIEn 位 = 1)
<ul style="list-style-type: none"> 由指令清除。 复位后。 		<ul style="list-style-type: none"> 由指令设置。

WTIMn [‡]	等待状态和产生中断请求的控制	
0	在第 8 个时钟的下降沿产生中断请求信号。 主模式：在输出 8 个时钟以后，时钟输出被置为低电平，并且设置等待状态。 从模式：在输入 8 个时钟以后，时钟被置为低电平，并为主设备设置等待状态。	
1	在第 9 个时钟信号的下降沿产生中断请求信号。 主模式：在输出 9 个时钟以后，时钟输出被置为低电平，并且设置等待状态。 从模式：在输入 9 个时钟以后，时钟被置为低电平，并为主设备设置等待状态。	
地址传输期间，不论该位如何设置，在第 9 个时钟下降沿处发生中断。地址传输完成后该位的设置生效。主模式下，在地址传输期间的第 9 个时钟下降沿处，插入一个等待状态。对于接收到本地地址的从设备，产生 ACK 信号之后的第 9 个时钟的下降沿处插入一个等待状态。但是，当从设备收到扩展码后，在第 8 个时钟的下降沿插入等待状态。		
清零条件 (WTIMn 位 = 0)		置位条件 (WTIMn 位 = 1)
<ul style="list-style-type: none"> 由指令清除。 复位后。 		<ul style="list-style-type: none"> 由指令设置。

ACKEn [‡]	应答控制	
0	禁止应答。	
1	允许应答。在第 9 个时钟期间，SDA0n 信号线设置为低电平。	
ACKEn 位的设置对从设备的地址接收是无效的，这种情况下，地址匹配时产生 ACK 信号。 但是，在接收扩展码地址时，ACKEn 位的设置有效。在接收到扩展码的系统中设置 ACKEn 位。		
清零条件 (ACKEn 位 = 0)		置位条件 (ACKEn 位 = 1)
<ul style="list-style-type: none"> 由指令清除。 复位后。 		<ul style="list-style-type: none"> 由指令设置。

注 当 IICEn 位 = 0 时，该标志信号无效。

备注 n = 0 至 2

STTn	开始条件触发	
0	不产生开始条件。	
1	当总线被释放（在 STOP 模式下）： 产生一个开始条件（作为主设备启动）。当 SCLn 信号线为高电平时，SDA0n 信号线由高电平变为低电平，这样就可以产生开始条件。经过一段合理的时间之后，SCL0n 信号线变为低电平。 与第三方通信期间： 如果使能通信保留功能（IICFn.IICRSVn 位 = 0）： • 该触发信号作为开始条件保留标志。当设置为 1 时，释放总线然后自动产生一个开始条件。 如果禁止通信保留功能（IICRSVn = 1）： • IICFn.STCFn 位设置为 1，并且 STTn 位的设置信息（1）被清除，该触发信号不能产生开始条件。 处于等待状态（作为主设备）： 等待状态释放后，产生重新开始条件。	
设置时序的有关注意事项： 主设备接收： 传输期间不能设置为 1，只能在 ACKEn 位已经设置 0，且从设备已经得知最末接收时，才可以置为 1。 主设备发送： 在 ACK 期间，通常不会产生开始条件。在第 9 个时钟输出后的等待期间置为 1。 从设备： 即使禁止通信保留功能（IICRSVn 位 = 1），也可以进入通信保留状态。 • 禁止 SPTn 位的同时设置为 1。 • 当 STTn 位设置为 1 时，禁止再次将 STTn 位置 1，直到设置被清 0 为止。		
清零条件（STTn 位 = 0）		置位条件（STTn 位 = 1）
<ul style="list-style-type: none"> • 在通信保留禁止状态，当 STTn 位设置为 1 时。 • 仲裁失败后清除。 • 主设备产生开始条件后清除。 • 当 LRELn 位 = 1 时（通信保存）。 • 当 IICE n 位 = 0（操作停止）时。 • 复位后 		<ul style="list-style-type: none"> • 由指令设置。

备注 1. 如果在数据设置后立即读取 STTn 位，其值为 0。
 2. n = 0 至 2

SPTn	停止条件触发				
0	不产生停止条件。				
1	产生停止条件（主设备传输的终结）。 SDA0 信号线变为低电平之后，将 SCL0n 信号线设置为高电平，或者一直等待直到 SCL0n 引脚变为高电平。然后经过一段合理的时间后，SDA0n 信号线由低电平变为高电平，产生停止条件。				
设置时序的有关注意事项 主设备接收： 传输期间不能被设置为 1，只能在 ACKEn 位已经设置为 0，且从设备已经得知最末接收后的等待期间，才可以被置为 1。 主设备发送： 在 ACK 接收期间，通常不会产生停止条件。在第 9 个时钟输出后的等待期间置为 1。 • 不能和 STTn 位同时置 1。 • 只有在主模式 ^注 下，SPTn 位才可以设置为 1。 • 在 WTIMn 位已经设置 0 时，如果在 8 个时钟输出后的等待期间将 SPTn 位设置为 1，注意，将在第 9 个时钟的高电平期间产生停止条件。 在 8 个时钟输出后的等待期间，WTIMn 位应该从 0 变为 1，且在第 9 个时钟输出后的等待期间应该将 SPTn 位设置为 1。 • 当 SPTn 位设置为 1 时，禁止再次将 SPTn 位置为 1，直到设置被清除为 0。					
<table border="1" style="width: 100%;"> <thead> <tr> <th>清零条件（SPTn 位 = 0）</th> <th>置位条件（SPTn 位 = 1）</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> • 仲裁失败后清除。 • 检测到停止条件后，自动清除。 • 当 LRELn 位 = 1 时清除（通信保存）。 • IICEn 位 = 0（操作停止）时。 • 复位后。 </td> <td> <ul style="list-style-type: none"> • 由指令设置。 </td> </tr> </tbody> </table>		清零条件（SPTn 位 = 0）	置位条件（SPTn 位 = 1）	<ul style="list-style-type: none"> • 仲裁失败后清除。 • 检测到停止条件后，自动清除。 • 当 LRELn 位 = 1 时清除（通信保存）。 • IICEn 位 = 0（操作停止）时。 • 复位后。 	<ul style="list-style-type: none"> • 由指令设置。
清零条件（SPTn 位 = 0）	置位条件（SPTn 位 = 1）				
<ul style="list-style-type: none"> • 仲裁失败后清除。 • 检测到停止条件后，自动清除。 • 当 LRELn 位 = 1 时清除（通信保存）。 • IICEn 位 = 0（操作停止）时。 • 复位后。 	<ul style="list-style-type: none"> • 由指令设置。 				

注 只有在主设备模式下，才可以将 SPTn 位设置为 1。但是，当 IICRSVn 位为 0 时，操作使能之后检测到第一个停止条件之前执行一个主操作，SPTn 位必须设置为 1 并产生一个停止条件。详情参见 19.15 注意事项。

注意事项 当 TRCn 位 = 1 时，如果在第 9 个时钟期间 WRELn 位被置 1 且等待状态被取消，之后 TRCn 位被清除为 0 且 SDA0n 信号线被置为高阻状态。

备注 1. 如果在数据设置以后立即读取 SPTn 位，其值为 0。
2. n = 0 至 2

(2) IIC 状态寄存器 0 至 2 (IICS0 至 IICS2)

IICS_n 寄存器表示 I²C0_n 的状态 (n = 0 至 2)。

这些寄存器只能按字节或按位进行读取。但是, 只有当 IICCN.STTn 位为 1 时或在等待期间才能读取 IICS_n 寄存器。

系统复位后, 这些寄存器被置为 00H。

注意事项 在以下状态时, 禁止访问 IICS_n 寄存器。详情参见 3.4.7 (2) 访问专用片上外设 I/O 寄存器。

- CPU 运行于副时钟且主时钟振荡停止时
- CPU 运行于内部振荡时钟时

(1/3)

复位后: 00H	R	地址: IICS0 FFFFFFFD86H, IICS1 FFFFFFFD96H, IICS2 FFFFFFFDA6H							
		<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS _n		MSTSn	ALDn	EXCn	COLn	TRCn	ACKDn	STDn	SPDn
(n = 0 至 2)									
MSTSn	主设备状态								
0	从设备状态或通信待机状态								
1	主设备通信状态								
清零条件 (MSTSn 位 = 0)					置位条件 (MSTSn 位 = 1)				
<ul style="list-style-type: none"> • 当检测到停止条件时。 • 当 ALDn 位 = 1 时 (仲裁失败)。 • 由 LRELn 位 = 1 清除 (通信保存)。 • 当 IICEn 位由 1 变为 0 (工作停止) 时。 • 复位后。 					<ul style="list-style-type: none"> • 当产生开始条件时。 				
ALDn	仲裁失败检测								
0	该状态意味着不存在总线仲裁, 或仲裁结果为“成功”。								
1	该状态表示仲裁结果为“失败”。MSTSn 位被清除为 0。								
清零条件 (ALDn 位 = 0)					置位条件 (ALDn 位 = 1)				
<ul style="list-style-type: none"> • IICS_n 寄存器被读取^注后, 自动清除。 • 当 IICEn 位由 1 变为 0 (工作停止) 时。 • 复位后。 					<ul style="list-style-type: none"> • 当总线仲裁结果为“失败”。 				
EXCn	扩展码接收的检测								
0	未接收到扩展码。								
1	接收到扩展码。								
清零条件 (EXCn 位 = 0)					置位条件 (EXCn 位 = 1)				
<ul style="list-style-type: none"> • 当检测到开始条件时。 • 当检测到停止条件时。 • 通过 LRELn 位 = 1 清除 (通信保存)。 • 当 IICEn 位由 1 变为 0 时 (操作停止)。 • 复位后。 					<ul style="list-style-type: none"> • 当接收地址数据的高 4 位是“0000”或“1111”时 (在第 8 个时钟的上升沿置位)。 				
<p>注 对 IICS_n 寄存器的其它位进行位操作时, 该位也被清除。</p>									

COIn	匹配地址检测	
0	地址不匹配	
1	地址匹配	
清零条件 (COIn 位 = 0)		置位条件 (COIn 位 = 1)
<ul style="list-style-type: none"> 当检测到开始条件时。 当检测到停止条件时。 通过 LRELn 位 = 1 清除 (通信保存)。 当 IICEn 位由 1 变为 0 (工作停止) 时。 复位后。 		<ul style="list-style-type: none"> 当接收到的地址和本地地址 (SVAn 寄存器) 匹配时 (在第 8 个时钟的上升沿置位)。

TRCn	发送/接收状态检测	
0	接收状态 (除发送状态外)。SDA0n 信号线被置为高阻态。	
1	发送状态。允许 SO 锁存器内的数值从 SDA0n 信号线输出 (在第一个字节的第 9 个时钟下降沿开始生效)。	
清零条件 (TRCn 位 = 0)		置位条件 (TRCn 位 = 1)
<ul style="list-style-type: none"> 当检测到停止条件时。 通过 LRELn 位 = 1 清除 (通信保存)。 当 IICEn 位由 1 变为 0 (工作停止) 时。 通过 IICn.WRELn 位 = 1 清除。 当 ALDn 位从 0 变为 1 (仲裁失败) 时。 复位后。 主设备: <ul style="list-style-type: none"> 当第一个字节的 LSB (传输方向指示位) 输出为“1”时。 从设备: <ul style="list-style-type: none"> 当检测到开始条件时。 当不用于通信时。 		主设备: <ul style="list-style-type: none"> 当产生开始条件时。 当第一个字节的 LSB (传输方向指示位) 输出为“0”时。 从设备: <ul style="list-style-type: none"> 当第一个字节的 LSB (传输方向指示位) 输入为“1”时。

ACKDn	ACK 检测	
0	未检测到 ACK。	
1	检测到 ACK。	
清零条件 (ACKDn 位 = 0)		置位条件 (ACKD 位 = 1)
<ul style="list-style-type: none"> 当检测到停止条件时。 在下一字节第一个时钟的上升沿。 通过 LRELn 位 = 1 清除 (通信从模式)。 当 IICEn 位由 1 变为 0 (工作停止) 时。 复位后。 		<ul style="list-style-type: none"> 在 SCL0n 引脚的第 9 个时钟的上升沿 SDA0n 引脚被设置为低电平后。

注 当 WRELn 位设置为 1，且在第九个时钟通过 TRCn 位 = 1 取消等待状态时，TRCn 被清除为 0，而且 SDA0n 信号线变为高阻态。

备注 n = 0 至 2

STDn	开始条件检测	
0	未检测到开始条件。	
1	检测到开始条件。这表示在地址传输时期有效时段内。	
清零条件 (STDn 位 = 0)		置位条件 (STDn 位 = 1)
<ul style="list-style-type: none"> 当检测到停止条件时。 在地址传输后下一个字节第一个时钟的上升沿。 通过 LRELn 位 = 1 清除 (通信保存)。 当 IICEn 位由 1 变为 0 (工作停止) 时。 复位后。 		<ul style="list-style-type: none"> 当检测到开始条件时。

SPDn	停止条件检测	
0	未检测到停止条件。	
1	检测到停止条件。主设备的通信被终结，并且释放总线。	
清零条件 (SPDn 位 = 0)		置位条件 (SPDn 位 = 1)
<ul style="list-style-type: none"> 在位置该位之后的地址传输第一个时钟上升沿处，以及检测到开始条件时。 当 IICEn 位由 1 变为 0 (工作停止) 时。 复位后。 		<ul style="list-style-type: none"> 当检测到停止条件时。

备注 n = 0 至 2

(3) IIC 标志寄存器 0 至 2 (IICF0 至 IICF2)

IICFn 寄存器设置 I²C0n 操作模式并指示 I²C 总线状态。

这些寄存器可以按字节或按位进行读取和写入。但是，STCFn 和 IICBSYn 位只读。

IICRSVn 使能/禁止通信保留功能（参见 19.14 通信保留）。

IICBSYn 位的初始值通过 STCENn 位设置（参见 19.15 注意事项）。

IICRSVn 和 STCENn 位只能在 I²C0n 操作禁止（IICn.IICEn 位 = 0）时写入。操作使能后，IICFn 可被读取（n = 0 至 2）。

系统复位后，这些寄存器被置为 00H。

复位后: 00H R/W[#] 地址: IICF0 FFFFFFFD8AH, IICF1 FFFFFFFD9AH, IICF2 FFFFFFFDAAH

	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

(n = 0 至 2)

STCFn	STTn 位清除
0	发出开始条件。
1	不发出开始条件, STTn 位被清除。
清零条件 (STCFn 位 = 0)	
<ul style="list-style-type: none"> 通过 IICn.STTn 位 = 1 清除。 当 IICn.IICEn 位 = 0 时。 复位后。 	
置位条件 (STCFn 位 = 1)	
<ul style="list-style-type: none"> 禁止通信保留 (IICRSVn 位 = 1) 期间, 不发出开始条件且 STTn 标志被清除为 0 时。 	

IICBSYn	I ² C0n 总线状态
0	总线释放状态 (STCENn 位 = 1 时的默认通信状态)。
1	总线通信状态 (STCENn 位 = 0 时的默认通信状态)。
清零条件 (IICBSYn 位 = 0)	
<ul style="list-style-type: none"> 当检测到停止条件时。 IICEn 位 = 0 时。 复位后。 	
置位条件 (IICBSYn 位 = 1)	
<ul style="list-style-type: none"> 当检测到开始条件时。 当 STCENn 位 = 0 时, 通过设置 IICEn 位来置位。 	

STCENn	初始化开始使能触发
0	不能产生开始条件, 直到操作使能后 (IICEn 位 = 1) 检测到停止条件为止。
1	产生开始条件, 即使操作使能后 (IICEn 位 = 1) 没有检测到停止条件。
清零条件 (STCENn 位 = 0)	
<ul style="list-style-type: none"> 当检测到开始条件时。 复位后。 	
置位条件 (STCENn 位 = 1)	
<ul style="list-style-type: none"> 由指令设置。 	

IICRSVn	通信保留功能禁止位
0	使能通信保留
1	禁止通信保留
清零条件 (IICRSVn 位 = 0)	
<ul style="list-style-type: none"> 由指令清除。 复位后。 	
置位条件 (IICRSVn 位 = 1)	
<ul style="list-style-type: none"> 由指令设置。 	

注 第 6 位和第 7 位为只读位。

- 注意事项**
1. 仅当操作停止时 (IICEn 位 = 0), 才可以写入 STCENn 位。
 2. 当 STCENn 位 = 1 时, 紧随在 I²Cn 总线操作使能之后, 不管总线的实际状态如何, 总是认为处于总线释放状态 (IICFn.IICBSYn 位 = 0)。因此, 要产生首个开始条件 (STTn 位 = 1), 必需确认总线已经被释放, 这样不会干扰其它通信。
 3. 仅当操作停止时 (IICEn 位 = 0), 才可以写入 IICRSVn 位。

(4) IIC 时钟选择寄存器 0 至 2 (IICCL0 至 IICCL2)

IICCLn 寄存器为 I²C0n 设置传输时钟。

这些寄存器可以按字节或按位进行读取或写入。但是，CLDn 和 DADn 位是只读位。

当 IICn.IICEn 位 = 0 时，设置 IICCLn 寄存器。

组合使用 SMCn, CLn1 和 CLn0 位以及 IICXn.CLXn 位与 OCKSm 寄存器的 OCKSTHm, OCKSm1 和 OCKSm0 位进行设置 (参见 19.4 (6) I²C0n 传输时钟设置方式) (n = 0 至 2, m = 0, 1)。

系统复位后，这些寄存器被置为 00H。

复位后: 00H	R/W [*]	地址: IICCL0 FFFFD84H, IICCL1 FFFFD94H, IICCL2 FFFFD44H							
		7	6	<5>	<4>	3	2	1	0
IICCLn		0	0	CLDn	DADn	SMCn	DFCn	CLn1	CLn0
(n = 0 至 2)									
CLDn	SCL0n 引脚电平的检测 (仅当 IICn.IICEn 位 = 1 时有效)								
0	SCL0n 引脚检测为低电平。								
1	SCL0n 引脚检测为高电平。								
清零条件 (CLDn 位 = 0)						置位条件 (CLDn 位 = 1)			
<ul style="list-style-type: none"> 当 SCL0n 引脚为低电平时。 IICEn 位 = 0 (操作停止) 时。 复位后。 						<ul style="list-style-type: none"> 当 SCL0n 引脚为高电平时。 			
DADn	SDA0n 引脚电平的检测 (仅当 IICEn 位 = 1 时有效)								
0	SDA0n 引脚检测为低电平。								
1	SDA0n 引脚检测为高电平。								
清零条件 (DADn 位 = 0)						置位条件 (DADn 位 = 1)			
<ul style="list-style-type: none"> 当 SDA0n 引脚为低电平时。 IICEn 位 = 0 (操作停止) 时。 复位后。 						<ul style="list-style-type: none"> 当 SDA0n 引脚为高电平时。 			
SMCn	工作模式切换								
0	工作于标准模式。								
1	工作于高速模式。								
DFCn	数字滤波器操作控制								
0	数字滤波器关闭。								
1	数字滤波器开启。								
数字滤波器只能在高速模式下使用。									
在高速模式下，不论 DFCn 位如何设置 (开启/关闭)，传输时钟都不会改变。									
数字滤波器用于消除高速模式下的噪声。									
注 第 4 位和第 5 位是只读位。									
注意事项 请确保将第 7 位和第 6 位清除为“0”。									
备注 当 IICn.IICEn 位 = 0 时，读取 CLDn 和 DADn 位，其值为 0。									

(5) IIC 功能扩展寄存器 0 至 2 (IICX0 至 IICX2)

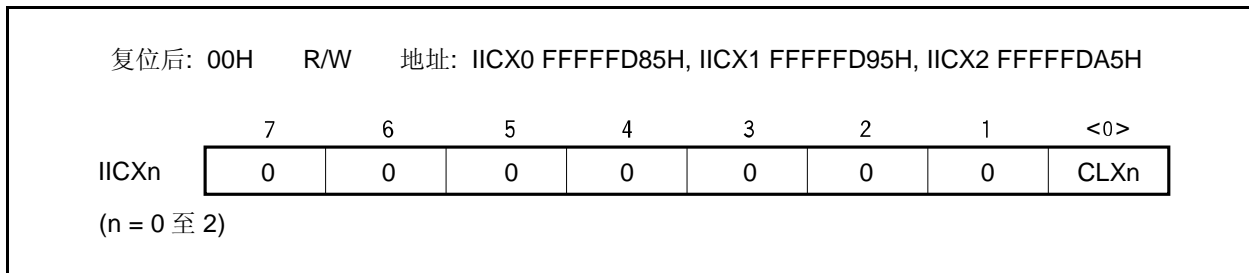
IICXn 寄存器设置 I²C0n 功能扩展 (仅在高速模式下有效)。

这些寄存器可以按字节或按位进行读取或写入。

组合使用 IICCLn 寄存器的 SMCn, CLn1 和 CLn0 位以及 OCKSm 寄存器的 OCKSTHm, OCKSm1 和 OCKSm0 位, 对 CLXn 位进行设置 (参见 19.4 (6) I²C0n 传输时钟设置方式) (m = 0, 1)。

当 IICn.IICEn 位 = 0 时, 设置 IICXn 寄存器。

系统复位后, 这些寄存器被置为 00H。



(6) I²C0n 传输时钟设置方式

使用下列表达式计算 I²C0n 传输时钟频率 (f_{SCL}) (n = 0 至 2)。

$$f_{SCL} = 1 / (m \times T + t_r + t_f)$$

m = 24, 48, 72, 96, 108, 120, 144, 172, 192, 240, 264, 344, 352, 396, 440, 516, 688, 860 (参见表 19-2 时钟设置)。

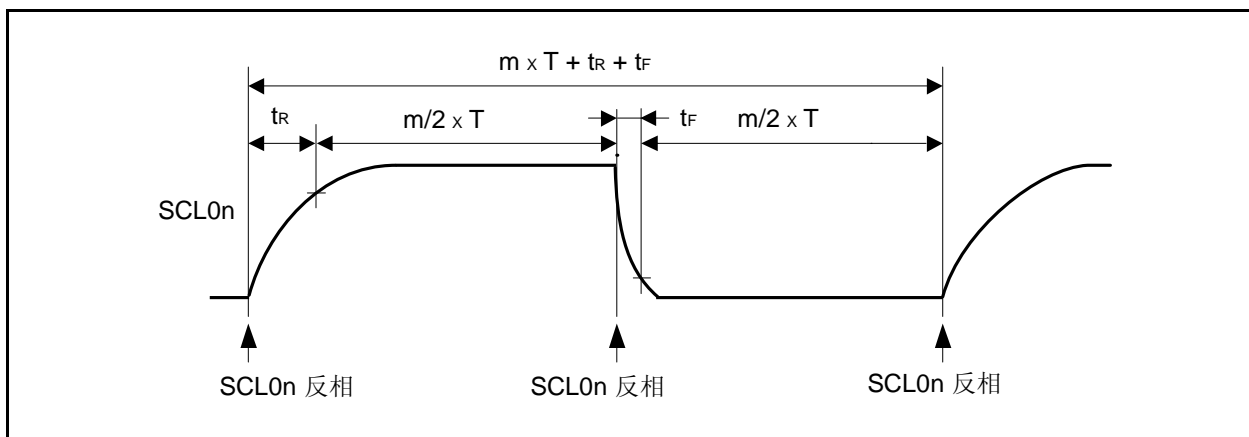
T: 1/f_{xx}

t_r: SCL0n 引脚上升时间

t_f: SCL0n 引脚下降时间

例如, 当 f_{xx} = 19.2 MHz, m = 198, t_r = 200 ns 且 t_f = 50 ns 时, I²C0n 传输时钟频率 (f_{SCL}) 使用下面表达式计算。

$$f_{SCL} = 1 / (198 \times 52 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 94.7 \text{ kHz}$$



组合使用 IICCLn 寄存器的 SMCn, CLn1 和 CLn0 位以及 IICXn 寄存器的 CLXn 位、OCKSm 寄存器的 OCKSTHm, OCKSm1 和 OCKSm0 位来设置选择时钟 (n = 0 至 2, m = 0, 1)。

表 19-2. 时钟设置

IICXn	IICCLn			选择时钟	传输时钟	可设置的主时钟频率 (fxx) 范围	传输速度	操作模式
	CLXn	SMCn	CLn1					
0	0	0	0	fxx/6 (OCKSm = 11H)	fxx/264	24.00 MHz ≤ fxx ≤ 25.14 MHz	90.91 kHz to 95.23 kHz	标准模式 (SMCn = 0)
				fxx/8 (OCKSm = 12H)	fxx/352	24.00 MHz ≤ fxx ≤ 33.52 MHz	68.18 kHz to 95.23 kHz	
				fxx/10 (OCKSm = 13H)	fxx/440	30.00 MHz ≤ fxx ≤ 41.90 MHz	68.18 kHz to 95.23 kHz	
0	0	0	1	fxx/4 (OCKSm = 10H)	fxx/344	24.00 MHz ≤ fxx ≤ 33.52 MHz	48.72 kHz to 97.44 kHz	
				fxx/6 (OCKSm = 11H)	fxx/516	25.14 MHz ≤ fxx ≤ 48.00 MHz	48.72 kHz to 93.02 kHz	
				fxx/8 (OCKSm = 12H)	fxx/688	33.52 MHz ≤ fxx ≤ 48.00 MHz	48.72 kHz to 69.77 kHz	
				fxx/10 (OCKSm = 13H)	fxx/860	41.90 MHz ≤ fxx ≤ 48.00 MHz	48.72 kHz to 55.81 kHz	
0	0	1	1	fxx/4 (OCKSm = 10H)	fxx/264	24.00 MHz ≤ fxx ≤ 25.60 MHz	90.91 kHz to 96.97 kHz	
				fxx/6 (OCKSm = 11H)	fxx/396	38.40 MHz	96.97 kHz	
0	1	0	X	fxx/4 (OCKSm = 10H)	fxx/96	24.00 MHz ≤ fxx ≤ 33.52 MHz	250.00 kHz to 349.17 kHz	
				fxx/6 (OCKSm = 11H)	fxx/144	24.00 MHz ≤ fxx ≤ 48.00 MHz	166.67 kHz to 333.33 kHz	
				fxx/8 (OCKSm = 12H)	fxx/192	32.00 MHz ≤ fxx ≤ 48.00 MHz	166.67 kHz to 250.00 kHz	
				fxx/10 (OCKSm = 13H)	fxx/240	40.00 MHz ≤ fxx ≤ 48.00 MHz	166.67 kHz to 200.00 kHz	
0	1	1	1	fxx/4 (OCKSm = 10H)	fxx/72	24.00 MHz ≤ fxx ≤ 25.60 MHz	333.33 kHz to 355.56 kHz	
				fxx/6 (OCKSm = 11H)	fxx/108	38.40 MHz	355.56 kHz	
1	1	0	X	fxx/6 (OCKSm = 11H)	fxx/72	24.00 MHz ≤ fxx ≤ 25.14 MHz	333.33 kHz to 349.17 kHz	
				fxx/8 (OCKSm = 12H)	fxx/96	32.00 MHz ≤ fxx ≤ 33.52 MHz	333.33 kHz to 349.17 kHz	
				fxx/10 (OCKSm = 13H)	fxx/120	40.00 MHz ≤ fxx ≤ 41.90 MHz	333.33 kHz to 349.17 kHz	
其它				禁止设置	-	-	-	-

备注 1. n = 0 至 2, m = 0 和 1
 2. x: 无需理会

(7) IIC 分频时钟选择寄存器 0, 1 (OCKS0, OCKS1)

OCKSm 寄存器控制 I²C0n 分频时钟 (n = 0 至 2, m = 0, 1)

这些寄存器通过 OCKS0 寄存器控制 I²C00 分频时钟, 通过 OCKS1 寄存器控制 I²C01 和 I²C02 分频时钟。

这些寄存器可以按字节进行读取或写入。

系统复位后, 这些寄存器被置为 00H。

复位后: 00H R/W 地址: OCKS0 FFFFF340H, OCKS1 FFFFF344H

	7	6	5	4	3	2	1	0
OCKSm	0	0	0	OCKSENm	OCKSTHm	0	OCKSm1	OCKSm0

(m = 0, 1)

OCKSENm	I ² C 分频时钟的操作设置
0	禁止 I ² C 分频时钟工作
1	使能 I ² C 分频时钟工作

OCKSTHm	OCKSm1	OCKSm0	I ² C 分频时钟的选择
0	0	0	f _{xx} /4
0	0	1	f _{xx} /6
0	1	0	f _{xx} /8
0	1	1	f _{xx} /10
1	0	0	f _{xx} /2
其它			禁止设置

(8) IIC 移位寄存器 0 至 2 (IIC0 至 IIC2)

IICn 移位寄存器和串行时钟同步, 用于串行发送/接收 (移位操作)。这些移位寄存器可以按字节读取或写入, 但是, 在数据传输期间, 不应该向 IICn 移位寄存器写入数据。

只有在等待期间, 可以访问 (读/写) IICn 移位寄存器。在等待期间以外的其它通信状态时, 禁止访问该寄存器。

但是, 对于主设备, 只有在传输触发位 (IICn.STTn 位) 被置为 1 后, IICn 移位寄存器才可以写入一次。

在等待期间写入 IICn 寄存器, 释放一个等待状态, 且启动数据传输 (n = 0 至 2)。

系统复位后, 这些寄存器被置为 00H。

复位后: 00H R/W 地址: IIC0 FFFFFD80H, IIC1 FFFFFD90H, IIC2 FFFFFDA0H

	7	6	5	4	3	2	1	0
IICn								

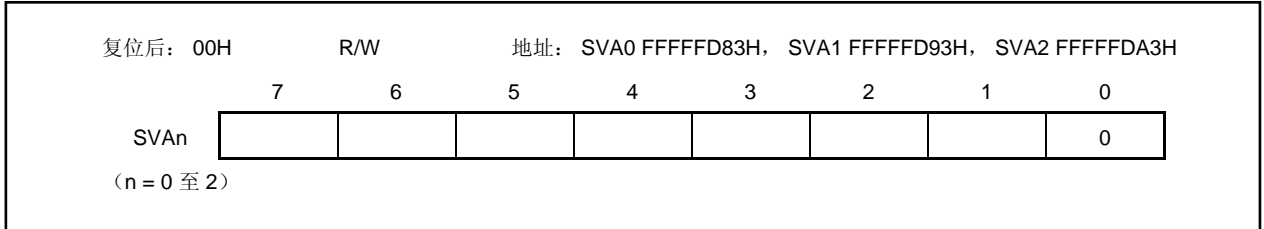
(n = 0 至 2)

(9) 从设备地址寄存器 0 至 2 (SVA0 至 SVA2)

SVAn 寄存器保持 I²C 总线的从地址。

这些移位寄存器可以按字节读取或写入，第 0 位固定为 0。但是当 IICSn.STDn 位 = 1 (开始条件检测) 时，禁止重写这些寄存器。

系统复位后，这些寄存器被置为 00H。



19.5 I²C 总线模式功能

19.5.1 引脚配置

串行时钟引脚 (SCL0n) 和串行数据总线引脚 (SDA0n) 配置如下 (n = 0 至 2)。

SCL0n 该引脚用于串行时钟输入和输出。

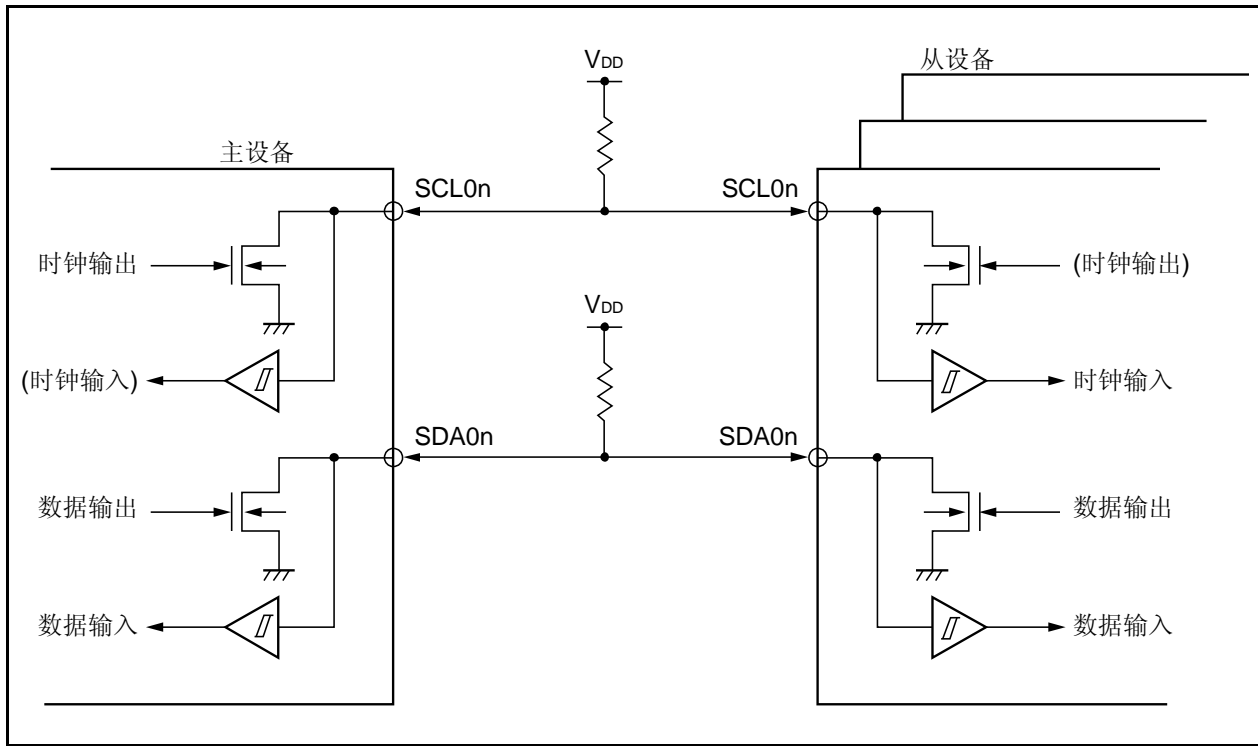
对于主设备和从设备, 该引脚都是 N 沟道漏极开路输出, 输入为施密特输入。

SDA0n 该引脚用于串行数据输入和输出。

对于主设备和从设备, 该引脚都是 N 沟道漏极开路输出, 输入为施密特输入。

由于串行时钟信号线和串行数据总线输出都是 N 沟道漏极开路输出, 所以都需要外接上拉电阻。

图 19-6. 引脚配置图

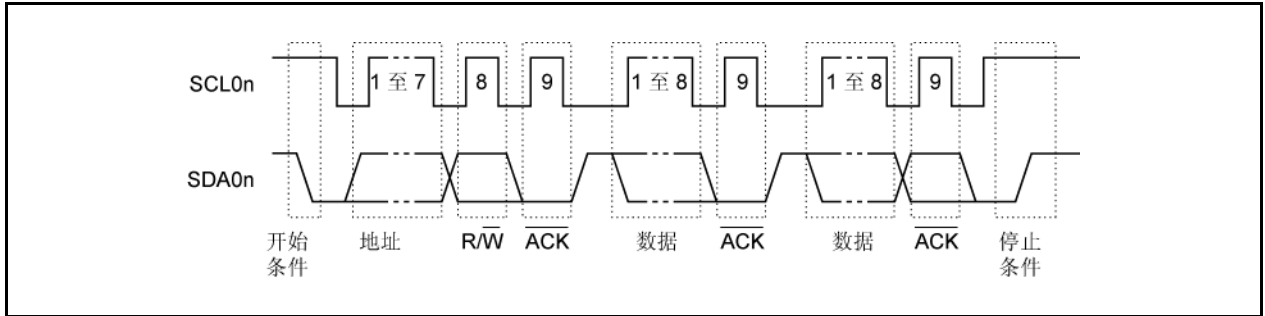


19.6 I²C总线的定义和控制方法

以下章节描述 I²C 总线的串行数据通信格式和 I²C 总线使用的信号。

关于 I²C 总线的串行数据总线产生的“开始条件”、“地址”、“传输方向”、“数据”以及“停止条件”的传输时序，如下所示。

图 19-7. I²C 总线串行数据传输时序



主设备产生开始条件、从设备地址和停止条件。

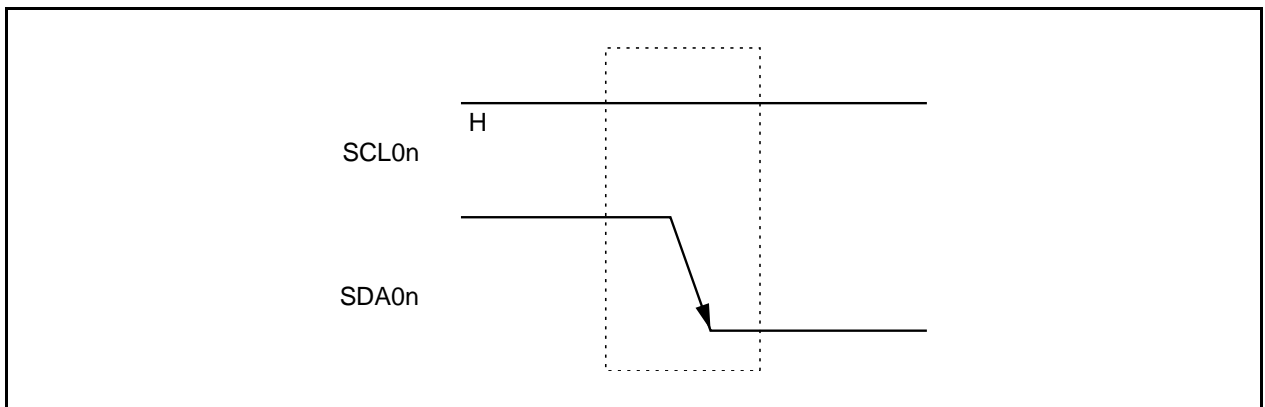
ACK 信号既可由主设备产生，也可由从设备产生。（通常情况下，它由接收 8 位数据的设备产生）。

串行时钟（SCL0n）由主设备连续输出。但是，在从设备中 SCL0n 引脚的低电平时段可以被延长，可以插入一个等待状态（n = 0 至 2）。

19.6.1 开始条件

当 SCL0n 引脚为高电平而且 SDA0n 引脚由高电平变为低电平时，形成一个开始条件。SCL0n 和 SDA0n 引脚的开始条件是启动串行传输时主设备输出至从设备的信号。从设备可以检测开始条件（n = 0 至 2）。

图 19-8. 开始条件



当检测到停止条件（IICSn.SPDn 位 = 1）后，当 IICC0.STT0 位被置位（1）时，输出一个开始条件。当检测到开始条件时，IICSn.STDn 位被置位（1）。（n = 0 至 2）

注意事项 在和其它设备进行通信时，当 V850ES/JG3-U 和 V850ES/JH3-U 的 IICn.IICEn 位被设置为 1 时，根据通信线的状态，可能会检测到开始条件。SCL0n 和 SDA0n 信号线是高电平时，请确保将 IICn.IICEn 位设置为 1。

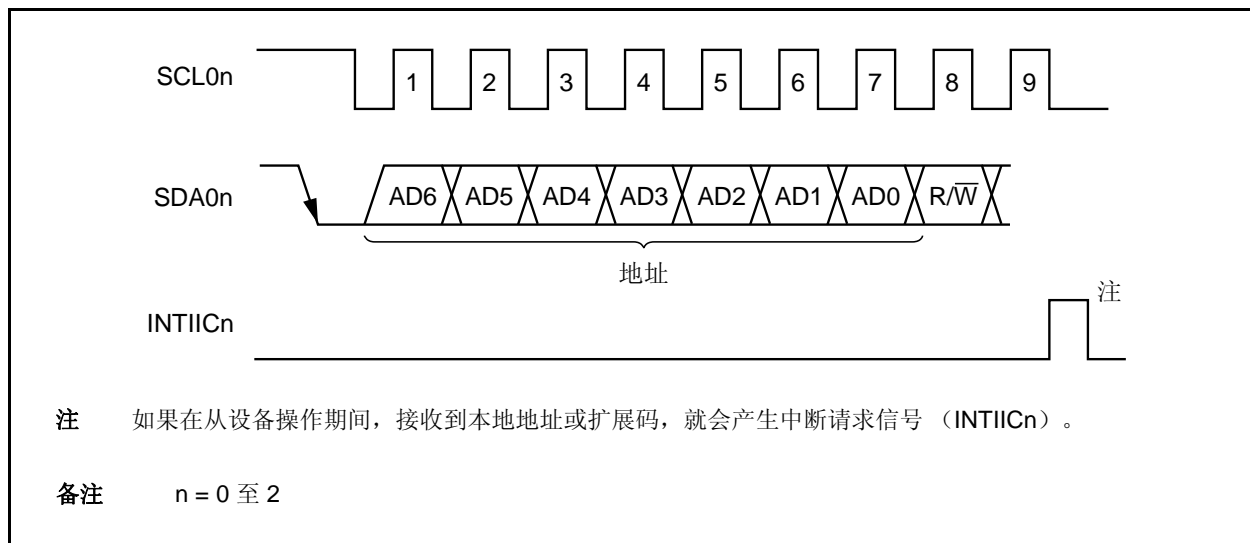
19.6.2 地址

紧随开始条件之后的 7 位数据被定义为地址。

地址是 7 位数据段，主设备通过输出地址来选择连接总线上的其中一个从设备。所以，每一个接入总线的从设备必须具有一个独一无二的地址。

从设备通过检测开始条件，并检查 7 位地址数据是否和 SVAn 寄存器内储存的数值匹配。如果地址数据和 SVAn 寄存器内储存的数据值相匹配，这个从设备就被选择，并且和主设备进行通信，直到主设备产生一个开始条件或结束条件。（n = 0 至 2）

图 19-9. 地址



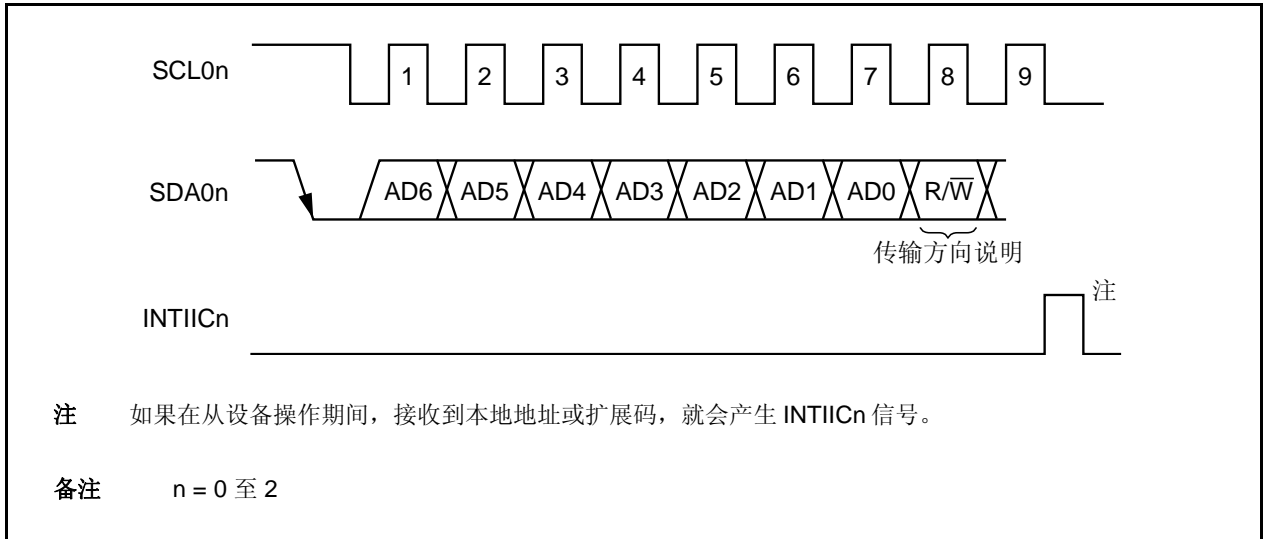
当从设备地址和在 19.6.3 指定传输方向中描述的传输方向一起作为第 8 位数据写入 IIC 移位寄存器 n（IICn）时，地址被输出。接收到的地址数据被写入 IICn 寄存器（n = 0 至 2）。

从设备地址被分配至 IICn 寄存器的高 7 位。

19.6.3 传输方向指示

除了 7 位地址数据之外，主设备发送 1 位数据指示传输方向。当该传输方向指示位为 0 时，表示主设备向从设备传输数据。当该传输方向指示位为 1 时，表示主设备由从设备接收数据。

图 19-10. 传输方向指示



19.6.4 ACK

ACK 用来确认发送设备和接收设备的串行数据传输状态。

每接收到的 8 位数据，接收设备返回一次 ACK 信号。

通常情况下，发送设备发送 8 位数据后，接收到一个 ACK 信号。当 ACK 信号由接收端返回后，接收被判定为正常并且继续进行操作。使用 IICSn.ACKDn 位对 ACK 信号的检测进行确认。

当主设备是接收设备时，收到最后一个数据之后，不返回 ACK 信号且产生停止条件。当从设备为接收设备且不返回 ACK 信号时，主设备产生一个停止条件或一个重新开始条件，然后停止当前的通信传输。返回 ACK 失败的情况可由以下因素导致：

- (a) 接收没有正常执行。
- (b) 接收到最末数据。
- (c) 指定地址上的接收设备（从设备）不存在。

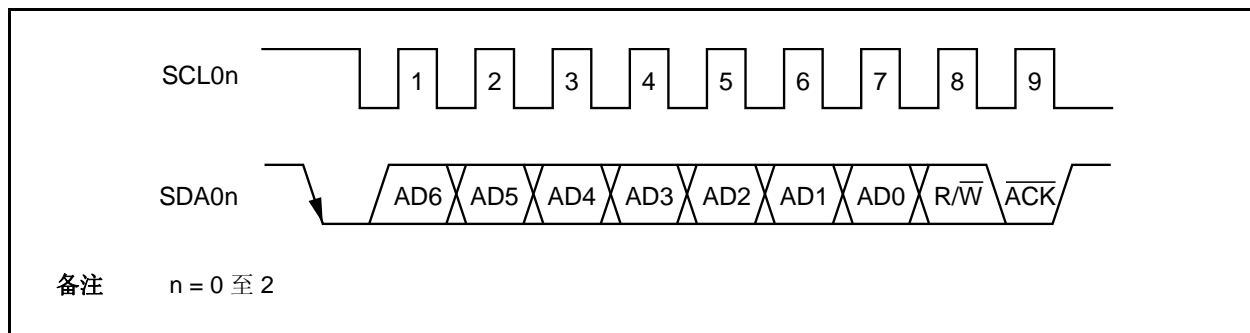
在第 9 个时钟期间，当接收设备将 SDA0n 信号线设置为低电平时，产生 ACK 信号（接收正常）。

当 IICcn.ACKEn 位设置为 1 时，使能自动产生 ACK。发送 7 位地址之后的第 8 位会导致 IICSn.TRcn 位被置位。通常情况下，设置 ACKE0 位为 1 作为接收的应答（TRcn 位 = 0）。

当从设备进行接收（TRcn 位 = 0）时，如果从设备不能接收数据或不再需要接收更多数据，将 ACKEn 位清除为 0，提示主设备，不能再接收后续数据。

与上述相似，当主设备进行接收（TRcn 位 = 0）时，并且不再需要后续数据，将 ACKEn 位清除为 0，来防止 ACK 信号的产生。这样可以通知从设备（发送设备）数据发送结束（发送停止）。

图 19-11. ACK



接收到本地地址后，不管 ACKEn 位的设置值如何，都会自动产生 ACK 信号。如果收到的地址不是本地地址（NACK），不产生 ACK 信号。

接收到扩展码时，在产生 ACK 信号之前，将 ACKEn 位设置为 1。

数据接收期间 ACK 的产生方式取决于等待时序的设置，描述如下：

- 当选择 8 时钟等待时（IICcn.WTIMn 位 = 0）：
如果等待状态取消前将 ACKEn 位设置为 1，ACK 信号在 SCL0n 引脚第 8 个时钟下降沿同步产生。
- 当选择 9 时钟等待时（IICcn.WTIMn 位 = 1）：
如果 ACKEn 位被预先设置为 1，产生 ACK 信号。

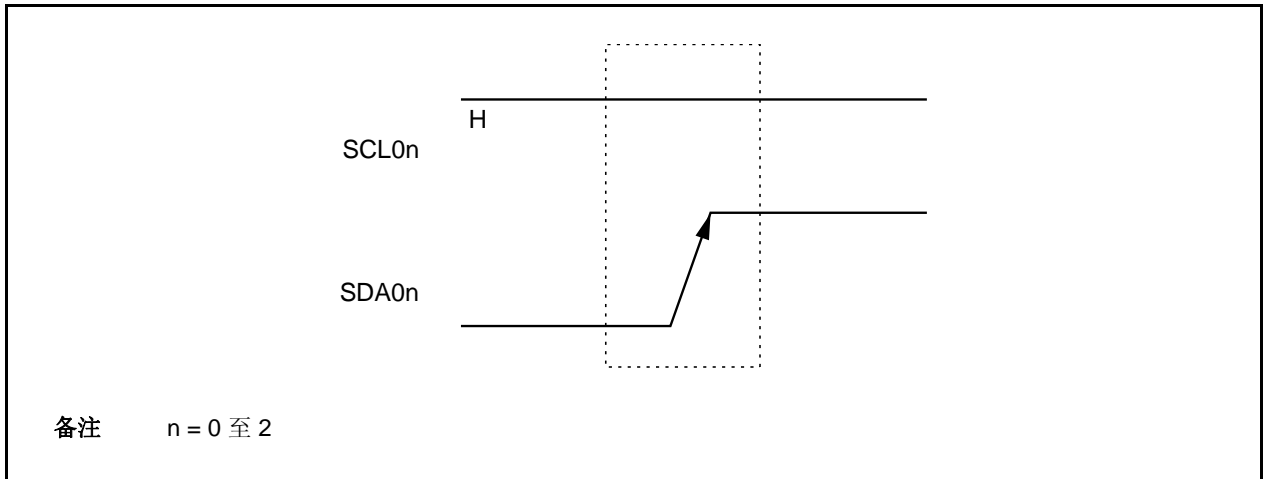
备注 n = 0 至 2

19.6.5 停止条件

当 SCL0n 引脚为高电平时，SDA0n 引脚由低电平变为高电平，即产生一个停止条件（n = 0 至 2）。

当主设备向从设备的串行传输完成之后，产生一个停止条件。当 V850ES/JG3-U 或 V850ES/JH3-U 被用作从设备时，它可以检测开始条件。

图 19-12. 停止条件



当 IICn.SPTn 位被设置为 1 时，发生一个停止条件。当检测到停止条件时，IICn.SPDn 位被置为 1，且当 IICn.SPIEn 位置为 1 时，产生一个中断请求信号（INTIICn）。（n = 0 至 2）

19.6.6 等待状态

等待状态用来通知通信伙伴，设备（主设备或从设备）已经准备就绪可以进行收发数据（即：处于等待状态）。

将 SCL0n 引脚设为低电平，通知通信伙伴本机已经处于等待状态。当主设备和从设备的等待状态都被取消时，下一次传输可以开始。（n = 0 至 2）

图 19-13. 等待状态 (1/2)

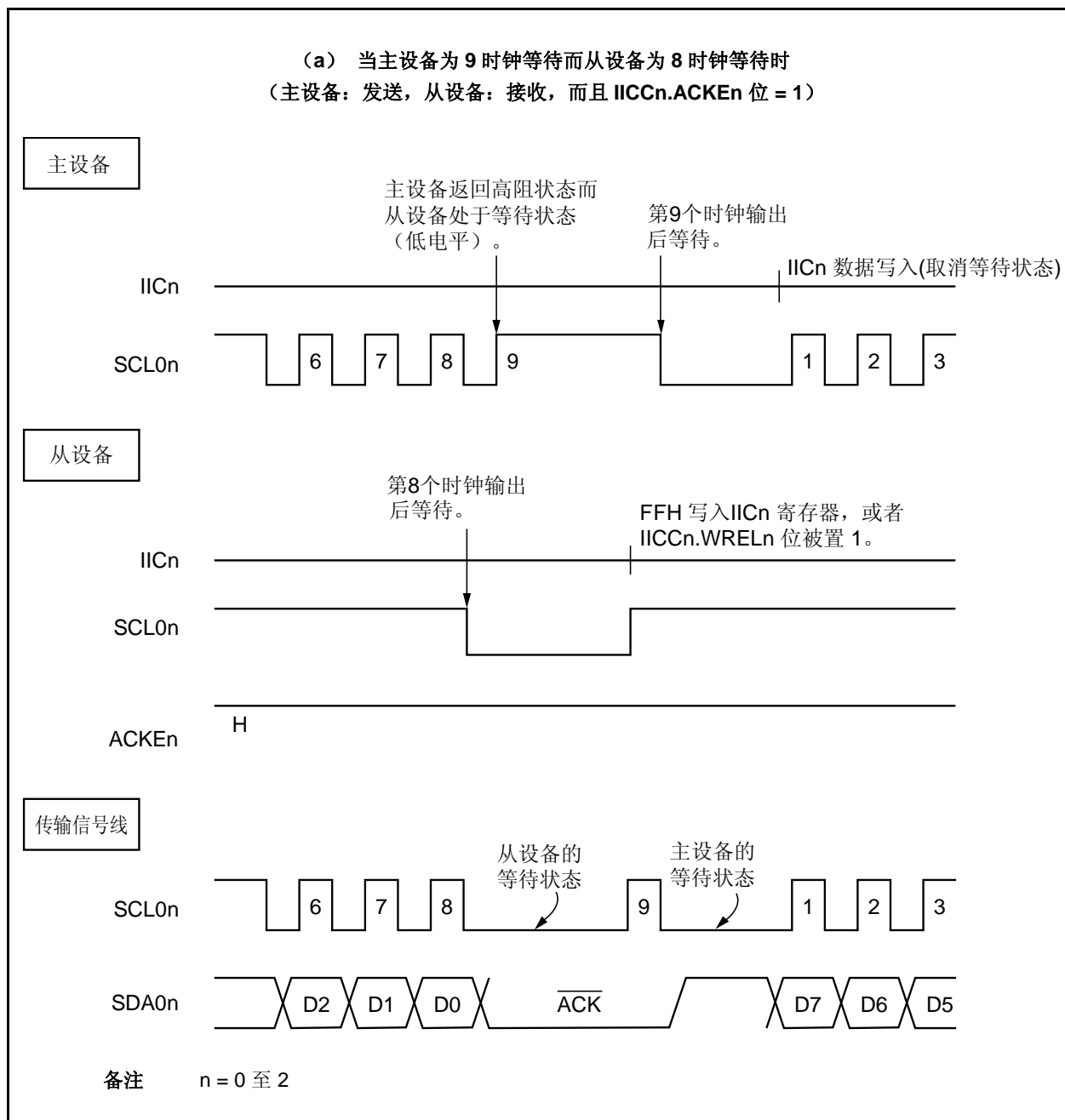
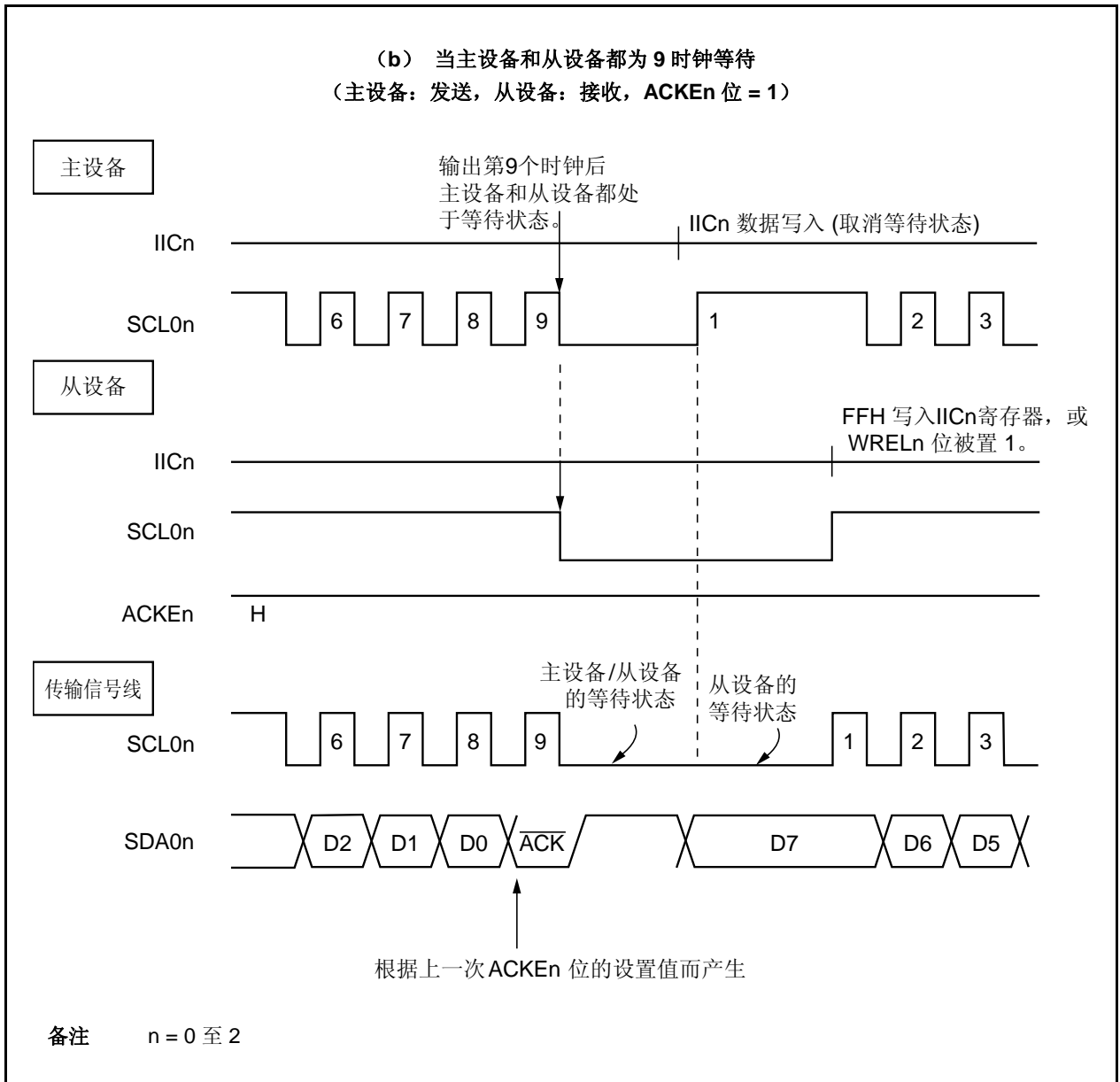


图 19-13. 等待状态 (2/2)



根据 IICn.WTIMn 位的设置, 可能会自动产生等待状态。(n = 0 至 2)

通常情况下, 当 IICn.WRELn 位置为 1, 或 FFH 被写入 IICn 寄存器时, 接收端取消等待状态, 当数据被写入 IICn 寄存器时, 发送端清除等待状态。

主设备还可以通过以下方式取消等待状态。

- 将 IICn.STTn 位置为 1。
- 将 IICn.SPTn 位置为 1。

19.6.7 等待状态取消方法

在使用 I²C0n 的情况下，一般使用以下方式取消等待状态（n = 0 至 2）。

- 向 IICn 寄存器写入数据。
- 将 IICn.WRELn 位置为 1（取消等待状态）。
- 将 IICn.STTn 位置为 1（产生开始条件）。
- 将 IICn.SPTn 位置为 1（产生停止条件）。

如果执行任何一种等待取消动作，I²C0n 将取消等待状态，并重新开始通信。

当取消等待状态并发送数据（包括地址）时，向 IICn 寄存器写入数据。

为了在取消等待状态之后接收数据，或完成数据发送，将 WRELn 位置为 1。

为了在取消等待状态之后产生一个重新开始条件，将 STTn 位设置为 1。

为了在取消等待状态之后产生一个停止条件，将 SPTn 位设置为 1。

对每个等待状态只可执行一次取消操作。

例如，如果将 WRELn 位设置为 1，取消等待状态，然后将数据写入 IICn 寄存器，SDA0n 信号线的时序变化和 IICn 寄存器的写入时序可能发生冲突，可能导致输出到 SDA0n 信号线上的数据不正确。

即使在其它操作情况下，如果通信中途停止，将 IICn.IICEn 位清除为 0 也会停止通信，使能等待状态的取消。

如果 I²C 总线因噪声等原因被死锁，将 IICn.LRELn 位设置为 1，会退出通信操作，使能等待状态的取消。

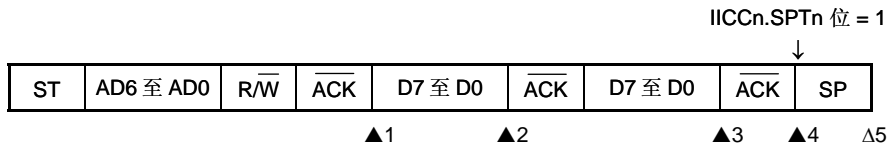
19.7 I²C中断请求信号 (INTIICn)

以下展示了 INTIICn 中断请求信号产生时序和 INTIICn 信号时序处的 IICSn 寄存器的值。(n = 0 至 2)

19.7.1 主设备操作

(1) 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止 (正常发送/接收)

<1> 当 IICn.WTIMn 位 = 0 时

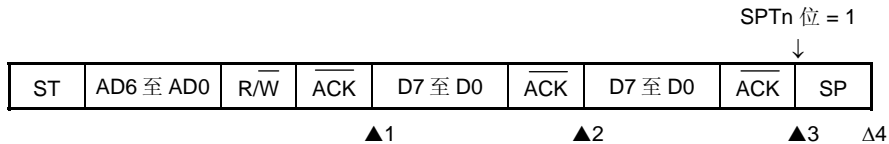


- ▲1: IICSn 寄存器 = 1000X110B
- ▲2: IICSn 寄存器 = 1000X000B
- ▲3: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)
- ▲4: IICSn 寄存器 = 1000XX00B
- Δ5: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



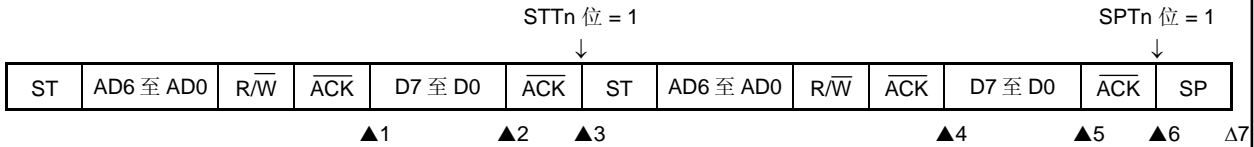
- ▲1: IICSn 寄存器 = 1000X110B
- ▲2: IICSn 寄存器 = 1000X100B
- ▲3: IICSn 寄存器 = 1000XX00B
- Δ4: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

(2) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止 (重新开始)

<1> 当 WTIMn 位 = 0 时

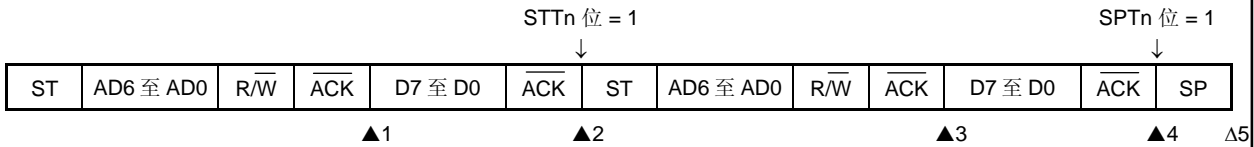


- ▲1: IICSn 寄存器 = 1000X110B
- ▲2: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)
- ▲3: IICSn 寄存器 = 1000XX00B (WTIMn 位 = 0)
- ▲4: IICSn 寄存器 = 1000X110B (WTIMn 位 = 0)
- ▲5: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)
- ▲6: IICSn 寄存器 = 1000XX00B
- Δ7: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



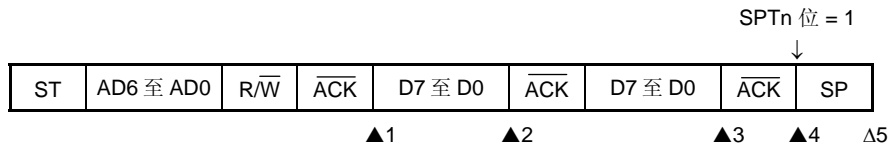
- ▲1: IICSn 寄存器 = 1000X110B
- ▲2: IICSn 寄存器 = 1000XX00B
- ▲3: IICSn 寄存器 = 1000X110B
- ▲4: IICSn 寄存器 = 1000XX00B
- Δ5: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

(3) 开始 ~ 扩展码 ~ 数据 ~ 数据 ~ 停止 (发送扩展码)

<1> 当 WTIMn 位 = 0 时



- ▲1: IICSn 寄存器 = 1010X110B
- ▲2: IICSn 寄存器 = 1010X000B
- ▲3: IICSn 寄存器 = 1010X000B (WTIMn 位 = 1)
- ▲4: IICSn 寄存器 = 1010XX00B
- Δ5: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



- ▲1: IICSn 寄存器 = 1010X110B
- ▲2: IICSn 寄存器 = 1010X100B
- ▲3: IICSn 寄存器 = 1010XX00B
- Δ4: IICSn 寄存器 = 00000001B

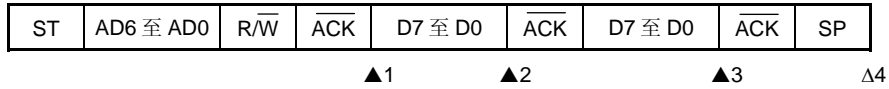
备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

19.7.2 从设备操作（当接收从设备地址数据（地址匹配）时）

(1) 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止

<1> 当 IICn.WTIMn 位 = 0 时



▲1: IICSn 寄存器 = 0001X110B

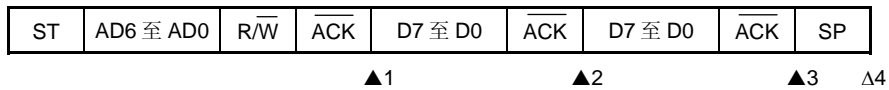
▲2: IICSn 寄存器 = 0001X000B

▲3: IICSn 寄存器 = 0001X000B

▲4: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
 Δ: 仅当 IICn.SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1: IICSn 寄存器 = 0001X110B

▲2: IICSn 寄存器 = 0001X100B

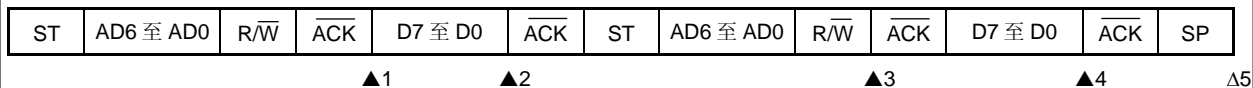
▲3: IICSn 寄存器 = 0001XX00B

▲4: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

(3) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 扩展码 ~ 数据 ~ 停止

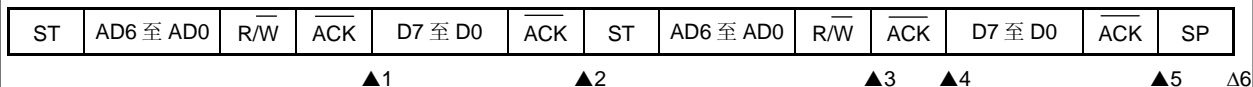
<1> 当 WTIMn 位 = 0 (重新开始后, 扩展码接收) 时



- ▲1: IICSn 寄存器 = 0001X110B
- ▲2: IICSn 寄存器 = 0001X000B
- ▲3: IICSn 寄存器 = 0010X010B
- ▲4: IICSn 寄存器 = 0010X000B
- Δ5: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 (重新开始后, 扩展码接收) 时



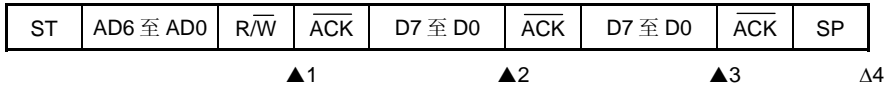
- ▲1: IICSn 寄存器 = 0001X110B
- ▲2: IICSn 寄存器 = 0001XX00B
- ▲3: IICSn 寄存器 = 0010X010B
- ▲4: IICSn 寄存器 = 0010X110B
- ▲5: IICSn 寄存器 = 0010XX00B
- Δ6: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

19.7.3 从设备操作（当接收扩展码时）

(1) 开始 ~ 扩展码 ~ 数据 ~ 数据 ~ 停止

<1> 当 IICn.WTIMn 位 = 0 时



▲1: IICSn 寄存器 = 0010X010B

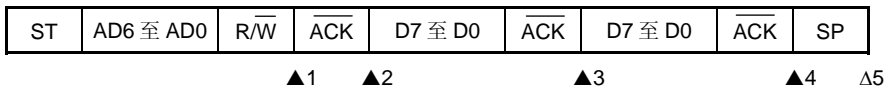
▲2: IICSn 寄存器 = 0010X000B

▲3: IICSn 寄存器 = 0010X000B

Δ4: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
 Δ: 仅当 IICn.SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1: IICSn 寄存器 = 0010X010B

▲2: IICSn 寄存器 = 0010X110B

▲3: IICSn 寄存器 = 0010X100B

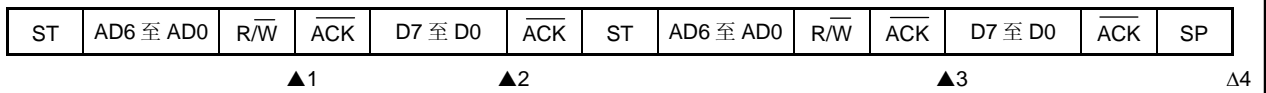
▲4: IICSn 寄存器 = 0010XX00B

Δ5: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

(4) 开始 ~ 扩展码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

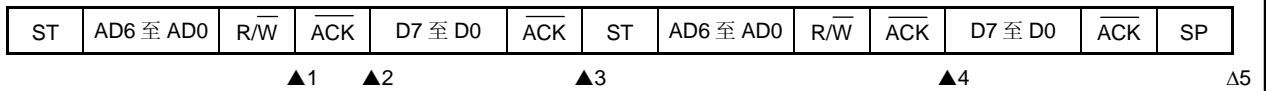
<1> 当 WTIMn 位 = 0 (重新开始后, 地址不匹配 (= 无扩展码)) 时



- ▲1: IICSn 寄存器 = 0010X010B
- ▲2: IICSn 寄存器 = 0010X000B
- ▲3: IICSn 寄存器 = 00000X10B
- Δ4: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 (重新开始后, 地址不匹配 (= 无扩展码)) 时

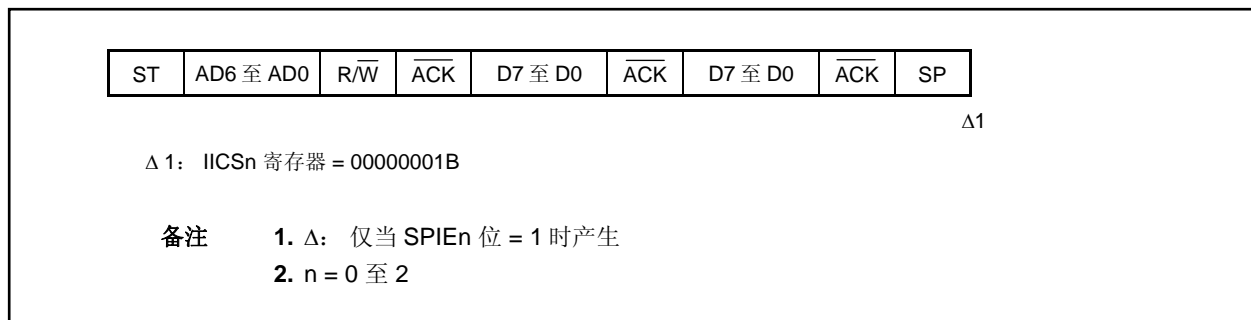


- ▲1: IICSn 寄存器 = 0010X010B
- ▲2: IICSn 寄存器 = 0010X110B
- ▲3: IICSn 寄存器 = 0010XX00B
- ▲4: IICSn 寄存器 = 00000X10B
- Δ5: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

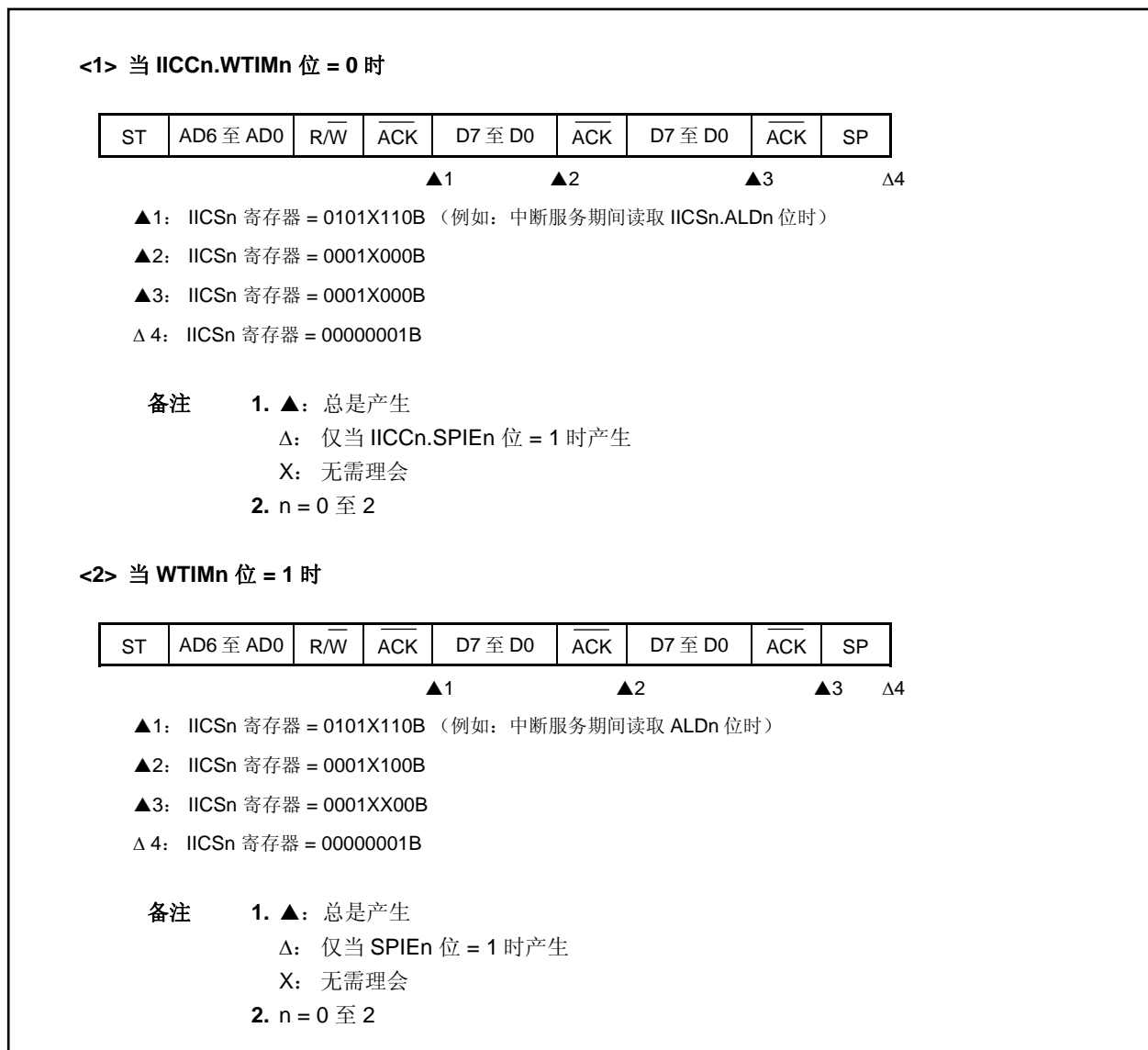
19.7.4 不通信时的操作

(1) 开始 ~ 扩展码 ~ 数据 ~ 数据 ~ 停止



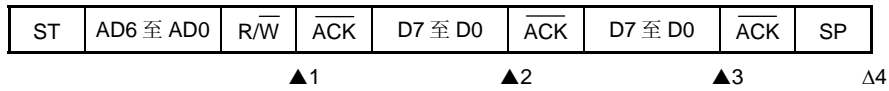
19.7.5 仲裁失败操作 (仲裁失败后作为从设备操作)

(1) 当在从设备地址数据传输期间发生仲裁失败



(2) 在扩展码的传输期间发生仲裁失败

<1> 当 WTIMn 位 = 0 时

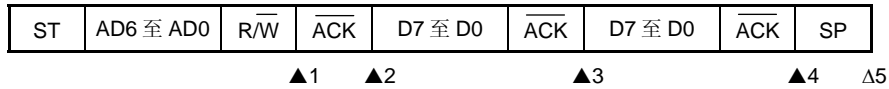


- ▲1: IICSn 寄存器 = 0110X010B (例如: 中断服务期间读取 ALDn 位时)
- ▲2: IICSn 寄存器 = 0010X000B
- ▲3: IICSn 寄存器 = 0010X000B
- Δ4: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



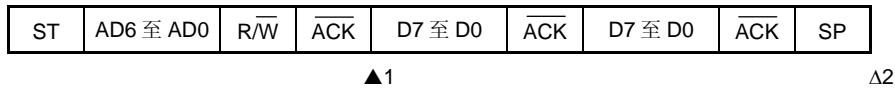
- ▲1: IICSn 寄存器 = 0110X010B (例如: 中断服务期间读取 ALDn 位时)
- ▲2: IICSn 寄存器 = 0010X110B
- ▲3: IICSn 寄存器 = 0010X100B
- ▲4: IICSn 寄存器 = 0010XX00B
- Δ5: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

19.7.6 发生仲裁失败时的操作（在仲裁失败后不进行通信）

(1) 在从设备地址数据传输期间发生仲裁失败

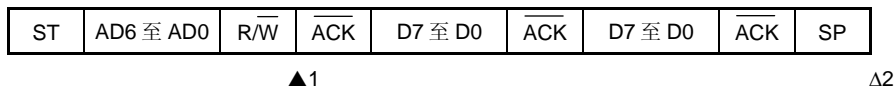


▲1: IICSn 寄存器 = 01000110B（例如：中断服务期间读取 IICSn.ALDn 位时）

Δ 2: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
Δ: 仅当 IICn.SPIEn 位 = 1 时产生
 2. n = 0 至 2

(2) 在扩展码的传输期间发生仲裁失败



▲1: IICSn 寄存器 = 0110X010B（例如：中断服务期间读取 ALDn 位时）

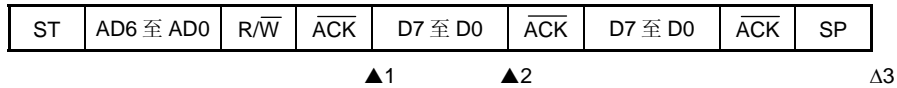
由软件设置 IICn.LRELn 位。

Δ 2: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
Δ: 仅当 SPIEn 位 = 1 时产生
X: 无需理会
 2. n = 0 至 2

(3) 在数据传输期间发生仲裁失败

<1> 当 IICn.WTIMn 位 = 0 时



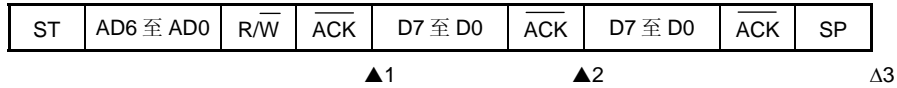
▲1: IICSn 寄存器 = 10001110B

▲2: IICSn 寄存器 = 01000000B (例如: 中断服务期间读取 ALDn 位时)

Δ3: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1: IICSn 寄存器 = 10001110B

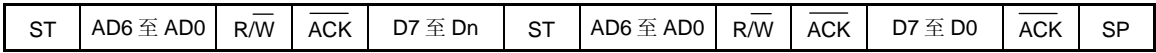
▲2: IICSn 寄存器 = 01000100B (例如: 中断服务期间读取 ALDn 位时)

Δ3: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 2. n = 0 至 2

(4) 在数据传输期间，因为重新开始条件导致发生仲裁失败

<1> 无扩展码（例如：地址不匹配）

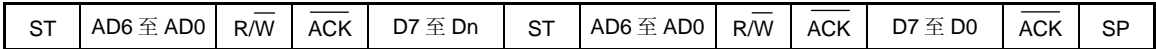


▲1 ▲2 Δ3

- ▲1: IICSn 寄存器 = 1000X110B
- ▲2: IICSn 寄存器 = 01000110B（例如：中断服务期间读取 ALDn 位时）
- Δ3: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. Dn = D6 至 D0
 n = 0 至 2

<2> 扩展代码

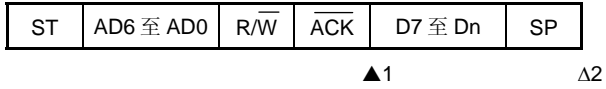


▲1 ▲2 Δ3

- ▲1: IICSn 寄存器 = 1000X110B
- ▲2: IICSn 寄存器 = 0110X010B（例如：中断服务期间读取 ALDn 位时）
 由软件设置 IICn.LRELn 位。
- Δ3: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. Dn = D6 至 D0
 n = 0 至 2

(5) 在数据传输期间，因为停止条件导致发生仲裁失败



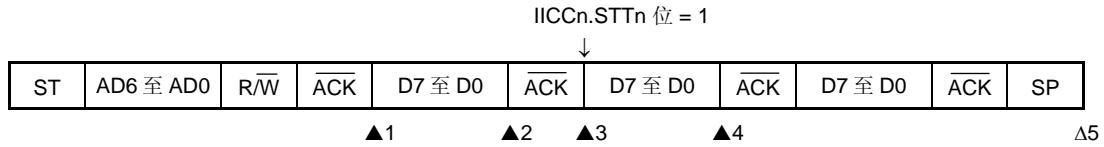
▲1: IICSn 寄存器 = 1000X110B

Δ 2: IICSn 寄存器 = 01000001B

- 备注**
1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. Dn = D6 至 D0
 n = 0 至 2

(6) 当试图产生重新开始条件时，因为 SDA0n 引脚的低电平导致发生仲裁失败

<1> 当 WTIMn 位 = 0 时

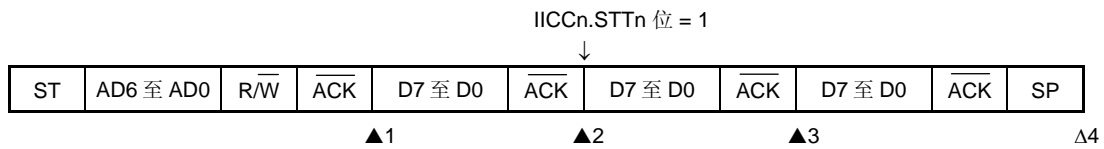


- ▲1: IICSn 寄存器 = 1000X110B
- ▲2: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)
- ▲3: IICSn 寄存器 = 1000XX00B (WTIMn 位 = 0)
- ▲4: IICSn 寄存器 = 01000000B (例如: 中断服务期间读取 ALDn 位时)
- Δ5: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



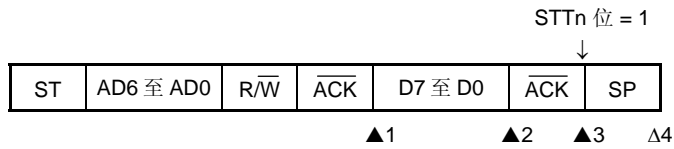
- ▲1: IICSn 寄存器 = 1000X110B
- ▲2: IICSn 寄存器 = 1000XX00B
- ▲3: IICSn 寄存器 = 01000100B (例如: 中断服务期间读取 ALDn 位时)
- Δ4: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

(7) 当试图产生重新开始条件时，由于停止条件导致发生仲裁失败

<1> 当 WTIMn 位 = 0 时

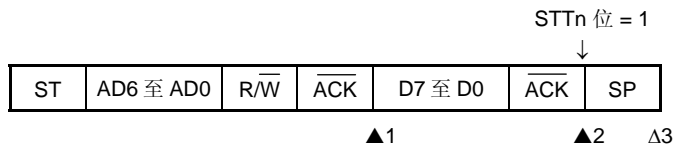


- ▲1: IICSn 寄存器 = 1000X110B
- ▲2: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)
- ▲3: IICSn 寄存器 = 1000XX00B
- Δ4: IICSn 寄存器 = 01000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



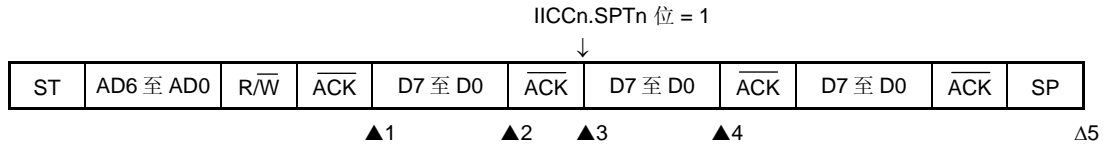
- ▲1: IICSn 寄存器 = 1000X110B
- ▲2: IICSn 寄存器 = 1000XX00B
- Δ3: IICSn 寄存器 = 01000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

(8) 当试图产生停止条件时，由于 SDA0n 引脚为低电平导致发生仲裁失败

<1> 当 WTIMn 位 = 0 时

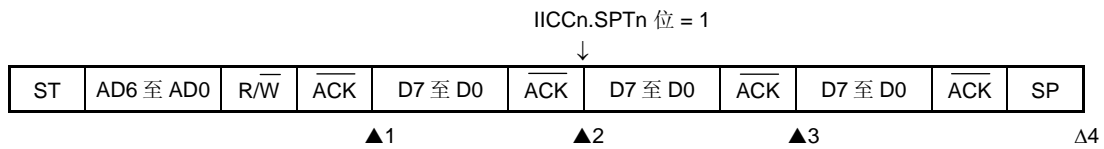


- ▲1: IICSn 寄存器 = 1000X110B
- ▲2: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)
- ▲3: IICSn 寄存器 = 1000XX00B (WTIMn 位 = 0)
- ▲4: IICSn 寄存器 = 01000000B (例如: 中断服务期间读取 ALDn 位时)
- Δ5: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



- ▲1: IICSn 寄存器 = 1000X110B
- ▲2: IICSn 寄存器 = 1000XX00B
- ▲3: IICSn 寄存器 = 01000000B (例如: 中断服务期间读取 ALDn 位时)
- Δ4: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

19.8 中断请求信号 (INTIICn) 的产生时序和等待控制

IICn.WTIMn 位的设置可以决定 INTIICn 寄存器的产生时序和对应的等待控制，如下所示。(n = 0 至 2)

表 19-3. INTIICn 产生时序和等待控制

WTIMn 位	从设备操作期间			主设备操作期间		
	地址	数据接收	数据发送	地址	数据接收	数据发送
0	9 ^{注 1,2}	8 ^{注 2}	8 ^{注 2}	9	8	8
1	9 ^{注 1,2}	9 ^{注 2}	9 ^{注 2}	9	9	9

- 注**
- 仅当接收的地址和 SVAn 寄存器设定的值相匹配时，在第 9 个时钟的下降沿产生从设备的 INTIICn 信号和等待时期。
此刻，不论 IICn.ACKEn 位如何设置，都产生 ACK 信号，对于已经接收到扩展码的从设备，INTIICn 信号在第 8 个时钟的下降沿发生。
当重新开始后地址不匹配时，INTIICn 信号在第 9 个时钟的下降沿发生，但是，不发生等待。
 - 如果接收地址和 SVAn 寄存器的内容不匹配，而且不能收到扩展码，则既不产生 INTIICn 信号，也不产生等待。

- 备注**
- 表中的数量表示串行时钟信号的时钟数目，中断请求信号和等待控制都和这些时钟信号的下降沿同步。
 - n = 0 至 2

(1) 地址发送/接收期间

- 从设备操作：根据上述注 1 和 2 描述的条件来决定中断请求和等待时序时，而不考虑 WTIMn 位的设置。
- 主设备操作：不论 WTIMn 位如何设置，中断请求和等待时序都在第 9 个时钟的下降沿发生。

(2) 在数据接收期间

- 主设备/从设备操作：中断请求和等待时序由 WTIMn 位的设置决定。

(3) 数据发送期间

- 主设备/从设备操作：中断请求和等待时序由 WTIMn 位的设置决定。

(4) 等待取消方法

四种等待取消方法如下所示：

- 将 IICn.WRELn 位置为 1。
- 向 IICn 寄存器写入数据。
- 开始条件的设置 (IICn.STTn 位= 1)^注。
- 停止条件的设置 (IICn.SPTn 位= 1)^注。

注 仅限主设备。

当已经选择 8 时钟等待时 (WTIMn 位 = 0)，在取消等待之前必须确定是否已经产生 $\overline{\text{ACK}}$ 信号。

备注 n = 0 至 2

(5) 停止条件检测

当检测到停止条件时，会产生 INTIICn 信号。

备注 n = 0 至 2

19.9 地址匹配检测方法

在 I²C 总线模式下，通过发送从设备的地址，主设备可以选择特定的从设备。

地址匹配的检测由硬件自动执行。当本地地址被置入 SVAn 寄存器以后，且置入 SVAn 寄存器的本地地址和主设备发送的从设备地址匹配时，或接收到扩展码时，产生一个 INTIICn 信号。（n = 0 至 2）

19.10 错误检测

在 I²C 总线模式下，串行数据总线引脚（SDA0n）在数据传送期间的状态被发送设备的 IICn 寄存器捕获，于是，发送之前 IICn 寄存器内的数据就可以和已发送的 IICn 数据进行比较，从而能够检测传输错误。当比较值不匹配时，判断为发生传输错误。（n = 0 至 2）

19.11 扩展代码

(1) 当接收地址的高 4 位是 0000 或 1111 时，扩展码标志（IICSn.EXCn 位）被置位，表示扩展码接收，且在第 8 个时钟的下降沿处产生一个中断请求信号（INTIICn）。（n = 0 至 2）
存储在 SVAn 寄存器中的本地地址不受影响。

(2) 如果通过 10 位地址传输将 11110xx0 设置到 SVAn 寄存器，且 11110xx0 数据由主设备发出，其结果如下所示。注意，INTIICn 信号在第 8 个时钟的下降沿处产生（n = 0 至 2）。

- 数据匹配的高 4 位： EXCn 位 = 1
- 数据匹配的第 7 位： IICSn.COIn 位 = 1

(3) 根据扩展码之后的数据，中断请求信号发生后的处理也各不相同，这些处理由软件执行。

例如，接收到扩展码以后，当不希望作为从设备操作时，将 IICn.LRELn 位设置为 1，CPU 将进入下一次通信等待状态。

表 19-4. 扩展码位的定义

从设备地址	R/W 位	描述
0000 000	0	通用呼叫地址
1111 0xx	0	10 位从设备地址规范（当地址被授权时）
1111 0xx	1	10 位从设备地址规范（在地址匹配之后读命令发布时）

备注 关于上述之外的扩展码，参见 NXP 发布的 I²C 总线规范。

19.12 仲裁

当多个主设备同时产生开始条件时（在 IICSn.STDn 位被置 1 之前，IICn.STTn 位已被置为 1），调整时钟数目，直到数据不同，多个主设备之间的通信才能执行，这种操作被称为仲裁（n = 0 至 2）。

当一个主设备在仲裁中失败时，在仲裁失败发生时序处将仲裁失败标志（IICSn.ALDn 位）设置为 1，且 SCL0n 信号线和 SDA0n 信号线都被设置为高阻状态，总线被释放。（n = 0 至 2）

仲裁失败的检测基于下次中断请求信号（INTIICn）的时序（当检测到停止条件时，第 8 或第 9 个时钟）和通过软件对 ALDn 位置 1 的设置（n = 0 至 2）。

关于中断请求信号时序的详情，参见 19.7 I²C 中断请求信号（INTIICn）。

图 19-14. 仲裁时序示例

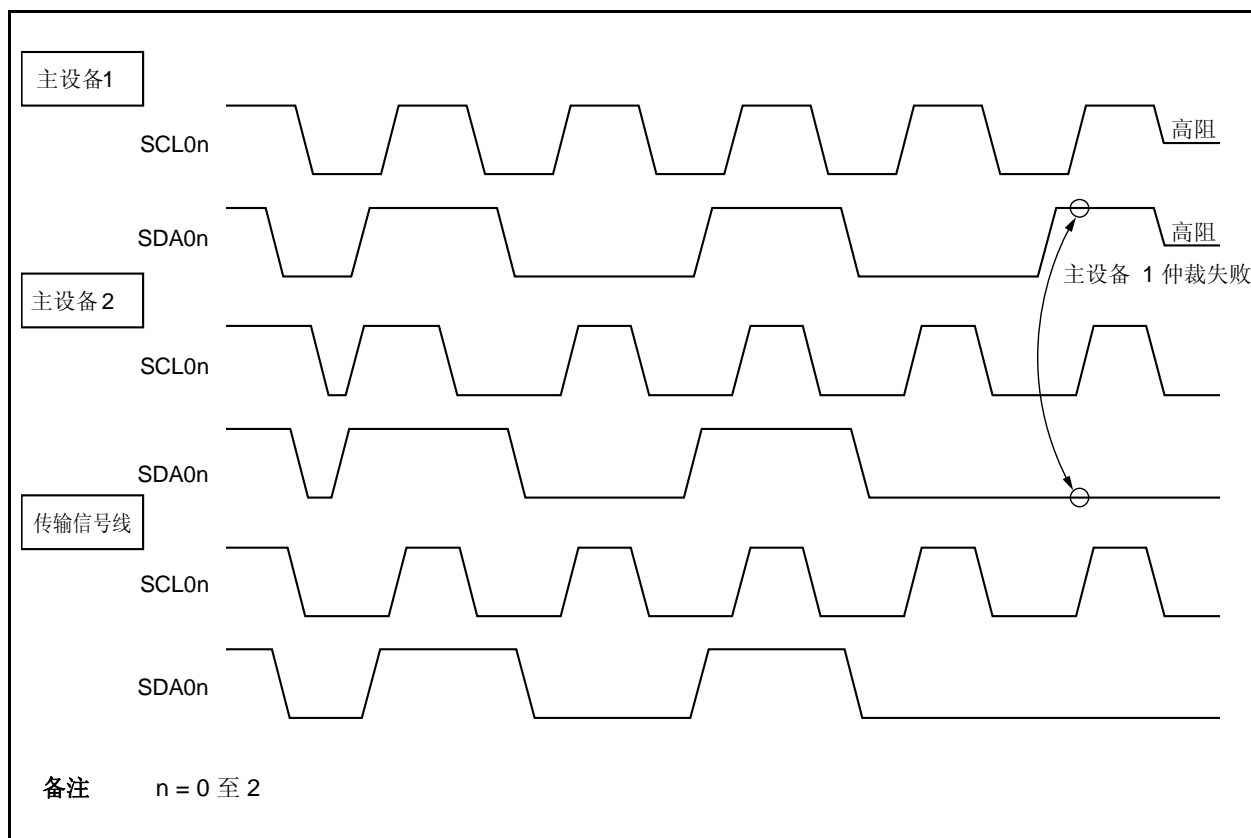


表 19-5. 仲裁期间的状态和中断请求信号发生时序

仲裁期间的状态	中断请求发生时序
发送地址传输	字节传输之后的第 8 个或第 9 个时钟的下降沿 ^{注 1}
地址传输后读出/写入数据	
发送扩展代码	
扩展码传输后读出/写入数据	
发送数据	
数据接收后 $\overline{\text{ACK}}$ 传送期间	
在数据传输期间检测到重新开始条件	
在数据传输期间检测到停止条件	当产生停止条件时（当 IICn.SPIEn 位 = 1 时） ^{注 2}
试图产生重新开始条件时 SDA0n 引脚是低电平	字节传输之后的第 8 个或第 9 个时钟的下降沿 ^{注 1}
试图产生重新开始条件时检测到停止条件	当产生停止条件时（当 IICn.SPIEn 位 = 1 时） ^{注 2}
试图产生停止条件时 SDA0n 引脚是低电平	字节传输之后的第 8 个或第 9 个时钟的下降沿 ^{注 1}
试图产生重新开始条件时 SCL0n 引脚是低电平	

- 注 1. 当 IICn.WTIMn 位 = 1 时，INTIICn 信号在第 9 个时钟的下降沿发生。当 WTIMn 位 = 0 且接收到扩展码的从地址时，INTIICn 信号在第 8 个时钟的下降沿发生（n = 0 至 2）。
2. 当有可能发生仲裁时，在主设备操作要设置 SPIEn 位为 1（n = 0 至 2）。

19.13 唤醒功能

使用 I²C 总线从设备功能接收到本地地址或扩展码时，准备功能是产生一个中断请求信号（INTIICn）的功能。

当地址数据不匹配时，该功能就会阻止不必要的 INTIICn 信号，从而使处理更有效率。

当检测到开始条件后，待机模式被唤醒。仲裁失败可能将主设备（已经产生开始条件）变为从设备，在这种可能情况下发送地址时，待机模式唤醒有效。

但是，当检测到停止条件后，不论唤醒功能如何设置，IICn.SPIEn 位总是置位，这可以决定 INTIICn 信号的使能或禁止（n = 0 至 2）。

19.14 通信保留

19.14.1 使能通信保留功能 (IICFn.IICRSVn 位 = 0) 时

在目前不能使用总线的情况下，开始主设备通信，通信保留功能可以在总线释放后使能开始条件的传输。在两种模式下，总线不能使用。

- 当仲裁结果既不是主设备操作，又不是从设备操作。
- 当接收到扩展码后且从设备操作被禁止时（当 IICFn.LRELn 位被置为 1 时，不返回 $\overline{\text{ACK}}$ 信号且总线释放）（n = 0 至 2）。

如果在不能使用总线时，将 IICFn.STTn 位置 1，将自动产生一个开始条件，并且在总线释放后（检测到停止条件后）设置等待状态。

检测到总线释放（检测到停止条件时）时，写入 IICFn 寄存器导致开始传送主设备地址。此时，IICFn.SPIEn 位应该被设置为 1（n = 0 至 2）。

当 STTn 位被设置为 1 后，根据总线状态决定工作模式（开始条件或通信保留）（n = 0 至 2）。

如果总线已经被释放产生开始条件

如果总线未被释放（待机模式）通信保留

要检测为 STTn 位决定何种工作模式，将 STTn 位设置为 1，待等待时期完毕后，检查 IICFn.MSTSn 位（n = 0 至 2）。

应该由软件设置的等待期间在表 19-6 中列出。

这些等待时期可以通过 IICFn 寄存器的 SMCn, CLn1 和 CLn0 位以及 IICFn.CLXn 位进行设置（n = 0 至 2）。

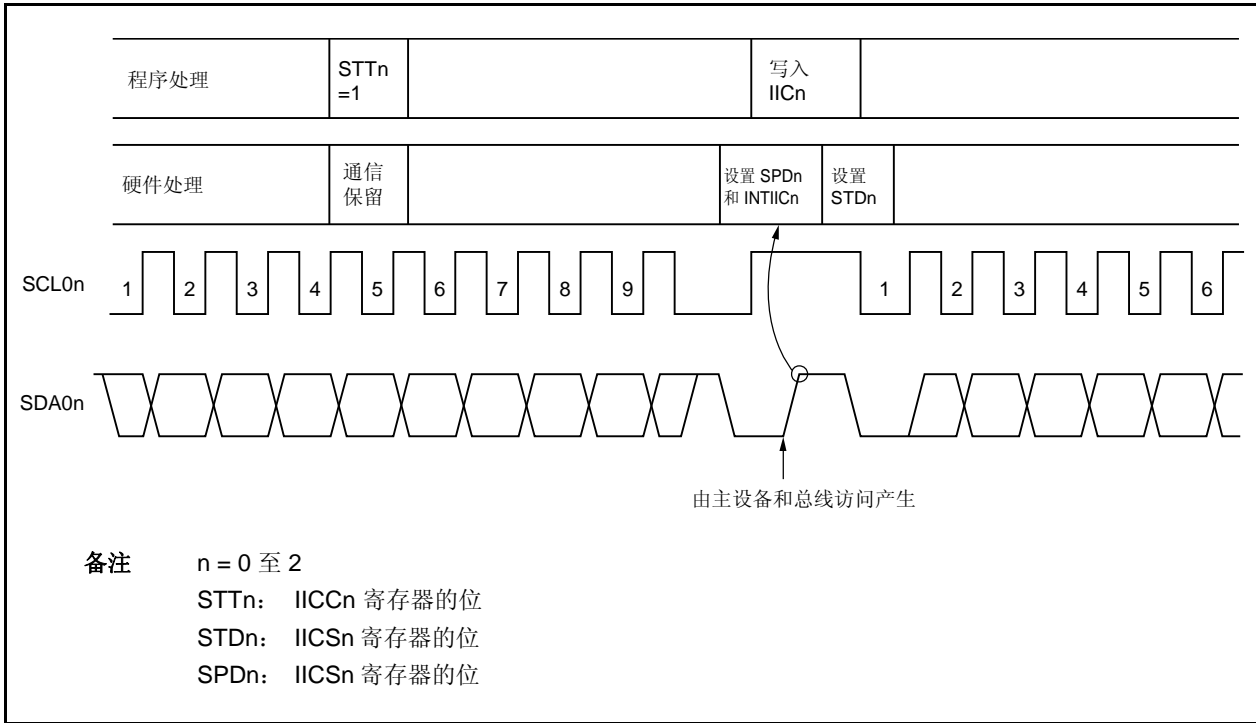
表 19-6. 等待时期

时钟选择	CLXn	SMCn	CLn1	CLn0	等待期
f _{xx} /6 (OCKSm = 11H)	0	0	0	0	156 个时钟
f _{xx} /8 (OCKSm = 12H)	0	0	0	0	208 个时钟
f _{xx} /10 (OCKSm = 13H)	0	0	0	0	260 个时钟
f _{xx} /4 (OCKSm = 10H)	0	0	0	1	188 个时钟
f _{xx} /6 (OCKSm = 11H)	0	0	0	1	282 个时钟
f _{xx} /8 (OCKSm = 12H)	0	0	0	1	376 个时钟
f _{xx} /10 (OCKSm = 13H)	0	0	0	1	470 个时钟
f _{xx} /4 (OCKSm = 10H)	0	0	1	1	148 个时钟
f _{xx} /6 (OCKSm = 11H)	0	0	1	1	222 个时钟
f _{xx} /4 (OCKSm = 10H)	0	1	0	×	64 个时钟
f _{xx} /6 (OCKSm = 11H)	0	1	0	×	96 个时钟
f _{xx} /8 (OCKSm = 12H)	0	1	0	×	128 个时钟
f _{xx} /10 (OCKSm = 13H)	0	1	0	×	160 个时钟
f _{xx} /4 (OCKSm = 10H)	0	1	1	1	52 个时钟
f _{xx} /6 (OCKSm = 11H)	0	1	1	1	78 个时钟
f _{xx} /6 (OCKSm = 11H)	1	1	0	×	60 个时钟
f _{xx} /8 (OCKSm = 12H)	1	1	0	×	80 个时钟
f _{xx} /10 (OCKSm = 13H)	1	1	0	×	100 个时钟

- 备注
1. n = 0 至 2
m = 0 和 1
 2. × = 无需理会

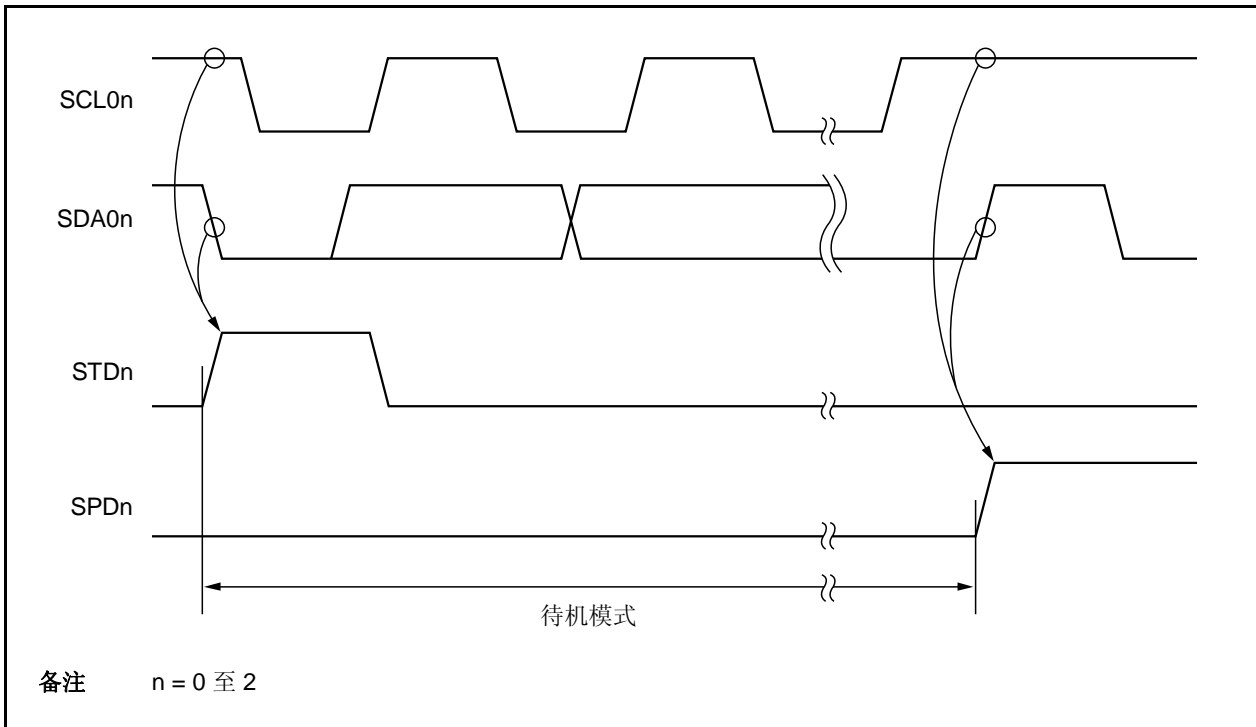
通信保留时序如下所示。

图 19-15. 通信保留时序



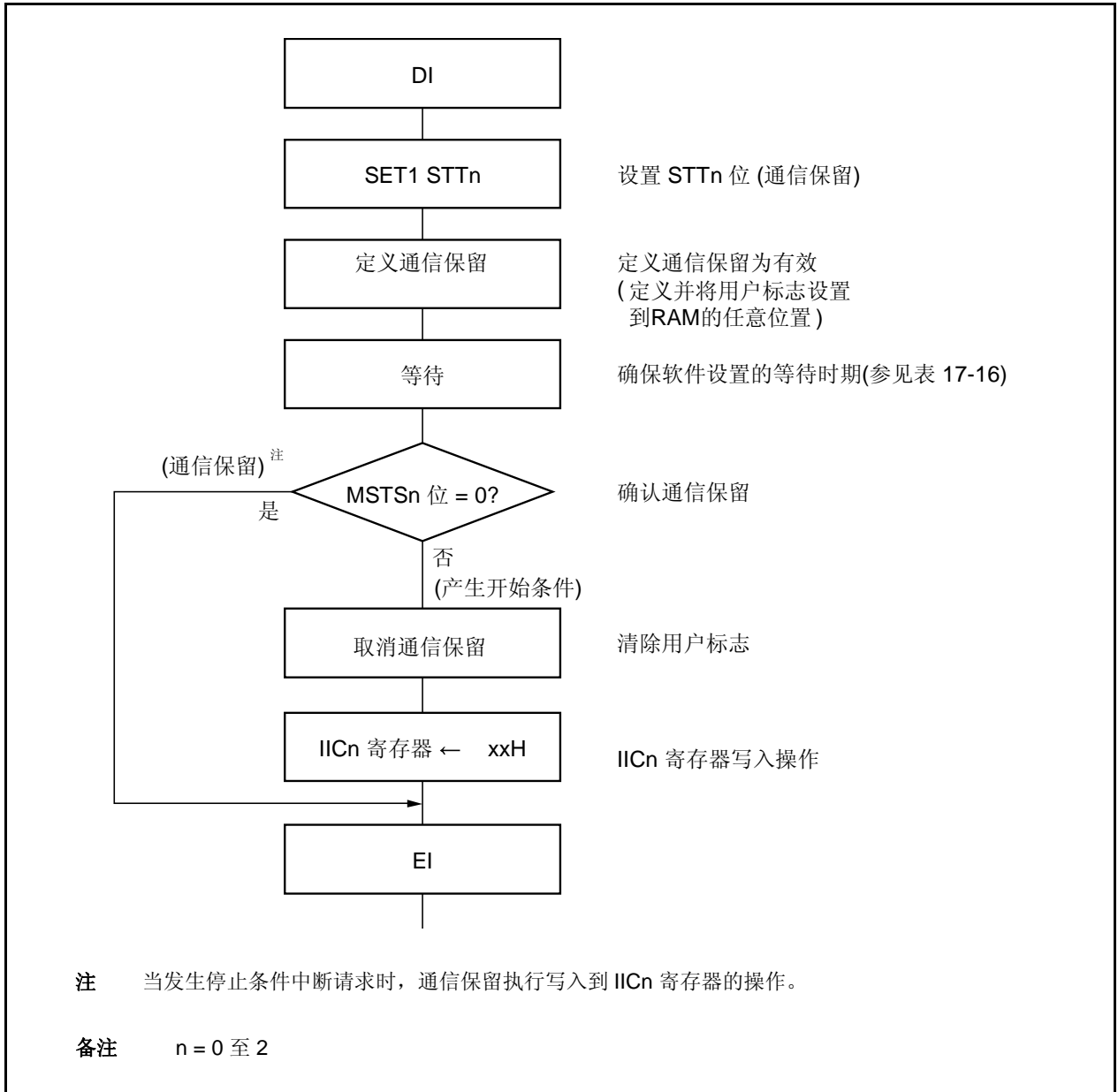
通过以下时序受理通信保留。IICSn.STDn 位设置为 1 后，在检测到停止条件前，设置 IICn.STTn 位为 1 就可以产生一次通信保留 (n = 0 至 2)。

图 19-16. 受理通信保留的时序



通信保留的流程图展示如下。

图 19-17. 通信保留的流程图



19.14.2 禁止通信保留功能时 (IICFn.IICRSVn 位 = 1)

在总线通信中总线没有被使用时，当 IICFn.STTn 位被置位时，通信保留请求被拒绝，且不会产生开始条件。有两种模式下不能使用总线。

- 当仲裁结果既不是主设备操作，又不是从设备操作。
- 当接收到扩展码后且从设备操作被禁止时（当 IICFn.LRELn 位被置为 1 时，不返回 $\overline{\text{ACK}}$ 信号且总线释放）（n = 0 至 2）。

为了确认是产生了开始条件，还是请求遭到拒绝，检查 IICFn.STCFn 标志。在 STTn 位被设置为 1 后，直到 STCFn 标志被置位，需要的时间在表 19-7 中列出。因而，通过软件来保证这段需要的时间。

表 19-7. 等待期

OCKSENm	OCKSm1	OCKSm0	CLn1	CLn0	等待期
1	0	0	0	×	20 个时钟
1	0	1	0	×	30 个时钟
1	1	0	0	×	40 个时钟
1	1	1	0	×	50 个时钟
0	0	0	1	0	10 个时钟

备注

1. ×: 无需理会
2. n = 0 至 2
m = 0, 1

19.15 注意事项

(1) 当 IICFn.STCENn 位 = 0 时

I²C0n 操作使能之后，不管总线的实际状态如何，总是立即认为处于总线通信状态（IICFn.IICBSYn 位 = 1）。要在尚未检测到停止条件的情况下执行主设备通信，先产生一个停止条件，在开始主设备通信之前释放总线。使用以下次序来产生停止条件。

- <1> 设置 IICCLn 寄存器。
- <2> 设置 IICcn.IICEn 位。
- <3> 设置 IICcn.SPTn 位。

(2) 当 IICFn.STCENn 位 = 1 时

I²C0n 操作使能之后，不管总线的实际状态如何，总是立即认为处于总线释放状态（IICFn.IICBSYn 位 = 0）。若要产生首个开始条件（IICcn.STTn 位 = 1），必需确认总线已经被释放，以免干扰其它通信。

(3) 在和其它设备进行通信时，当 V850ES/JG3-U 和 V850ES/JH3-U 的 IICcn.IICEn 位被设置为 1，根据通信线的状态，可能会检测到开始条件。当 SCL0n 和 SDA0n 信号线是高电平时，请确保将 IICcn.IICEn 位设置为 1。

(4) 使能操作（IICcn.IICEn 位 = 1）之前，通过 IICCLn, IICXn 和 OCKSm 寄存器决定工作时钟频率。要改变操作时钟频率，首先要将 IICcn.IICEn 位清 0 一次。

(5) 当 IICcn.STTn 和 IICcn.SPTn 位已经被设置为 1，若不首先将其清 0，则一定不要对其再次置位。

(6) 如果传输已经被保留，设置 IICCN.SPIEn 位为 1，这样，检测到停止条件就会产生一个中断请求信号。当产生一个中断请求信号后，向 I²Cn 写入通信数据将会释放等待状态，然后传输开始。如果检测到停止条件后不产生中断请求信号，由于未产生中断请求，传输将暂停在等待状态。但是，没有必要为了软件对 IICSn.MSTSn 位的检测而将 SPIEn 位置为 1。

备注 n = 0 至 2
 m = 0, 1

19.16 通信操作

以下展示 3 种操作过程和流程图。

(1) 主设备工作于单主系统

当使用 V850ES/JG3-U 和 V850ES/JH3-U 单片机作为单主系统中的主设备时，流程图如下所示。

流程图大体可以分为初始设置和通信处理两部分。在启动时执行初始设置。如果需要和从设备进行通信，进行通信准备后，执行通信处理。

(2) 主设备工作于多主系统

在 I²C_{0n} 总线多主系统中，当总线参与通信之后，不能由 I²C 总线规范判定总线是释放还是被使用。这里，当数据线和时钟线都在一定的时期（1 帧）内保持为高电平，V850ES/JG3-U 和 V850ES/JH3-U 就认为总线已释放并可以进行通信。

操作流程可以大体分为初始设置、通信等待和通信处理。V850ES/JG3-U 和 V850ES/JH3-U 在仲裁中失败并被指定为从设备的处理情况这里予以省略，而只是展示作为主设备的处理操作。启动时执行初始设置以便参与通信。然后，作为主设备等待通信请求或作为从设备等待指定。在通信处理中执行实际的通信，且支持与从设备的发送/接收操作，也支持与其它主设备间的仲裁处理。

(3) 从设备操作

当 V850ES/JG3-U 和 V850ES/JH3-U 用作 I²C_{0n} 总线的从设备时的示例如下。

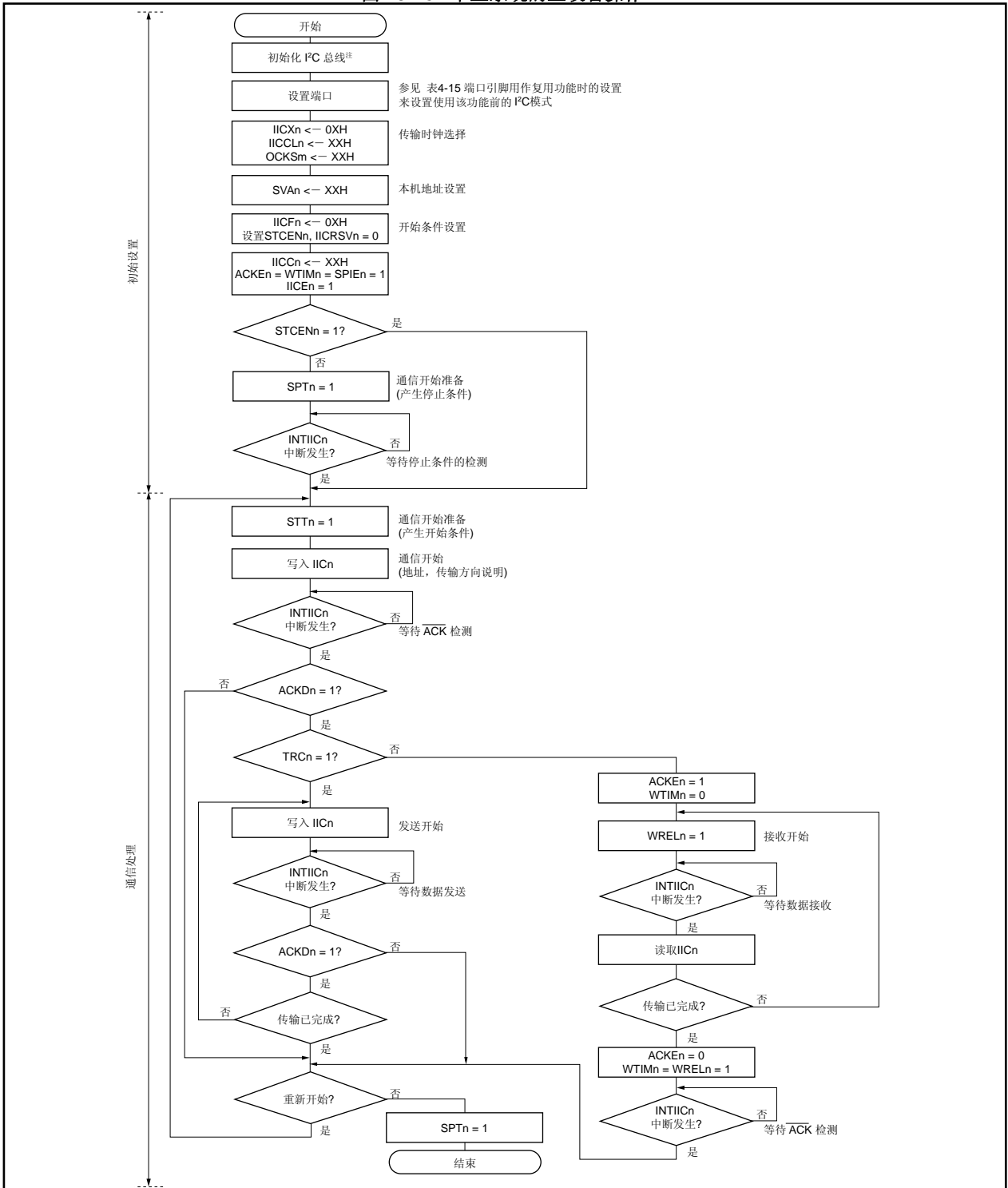
当用作从设备时，由一个中断启动操作。在启动时执行初始设置，然后，等待 INTIIC_n 中断信号的发生（通信等待）。当 INTIIC_n 中断发生时，通信状态被判定，并且判定结果作为一个标志传送给主处理程序。

通过检查该标志，可以执行必需的通信处理。

备注 n = 0 至 2

19.16.1 主设备工作于单主系统

图 19-18. 单主系统的主设备操作



注 释放 I²C_{0n} 总线（SCL_{0n}, SDA_{0n} 引脚 = 高电平），与进行通信的产品规格一致。
例如，当 EEPROMTM 输出低电平到 SDA_{0n} 引脚时，设置 SCL_{0n} 引脚为输出端口并从中输出时钟脉冲，直到 SDA_{0n} 引脚为稳定的高电平。

- 备注**
1. 至于发送格式和接收格式，符合进行通信的产品规格。
 2. n = 0 至 2, m = 0, 1

19.16.2 主设备工作于多主系统

图 19-19. 多主系统中的主设备操作 (1/3)

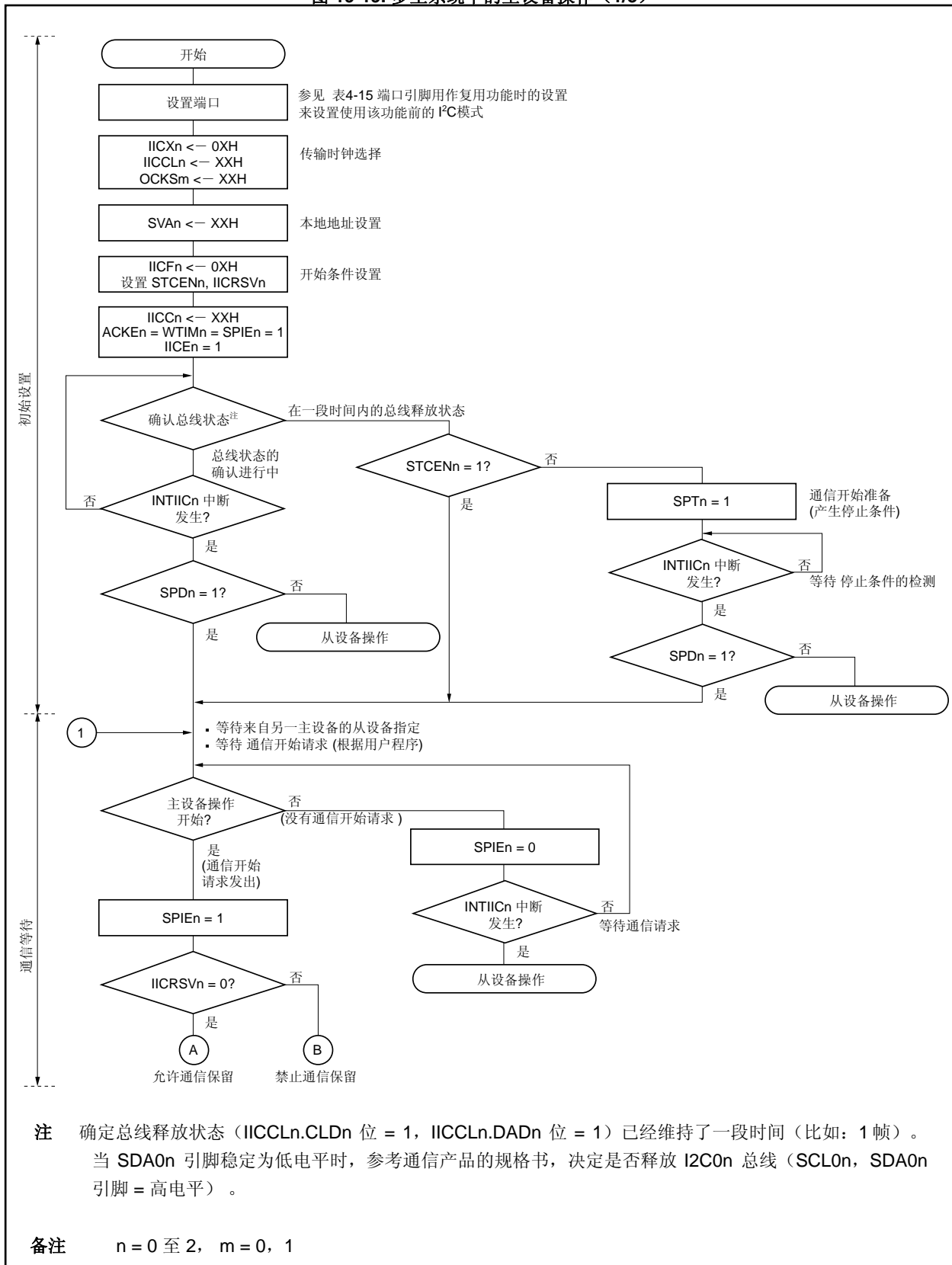


图 19-19. 多主系统中的主设备操作 (2/3)

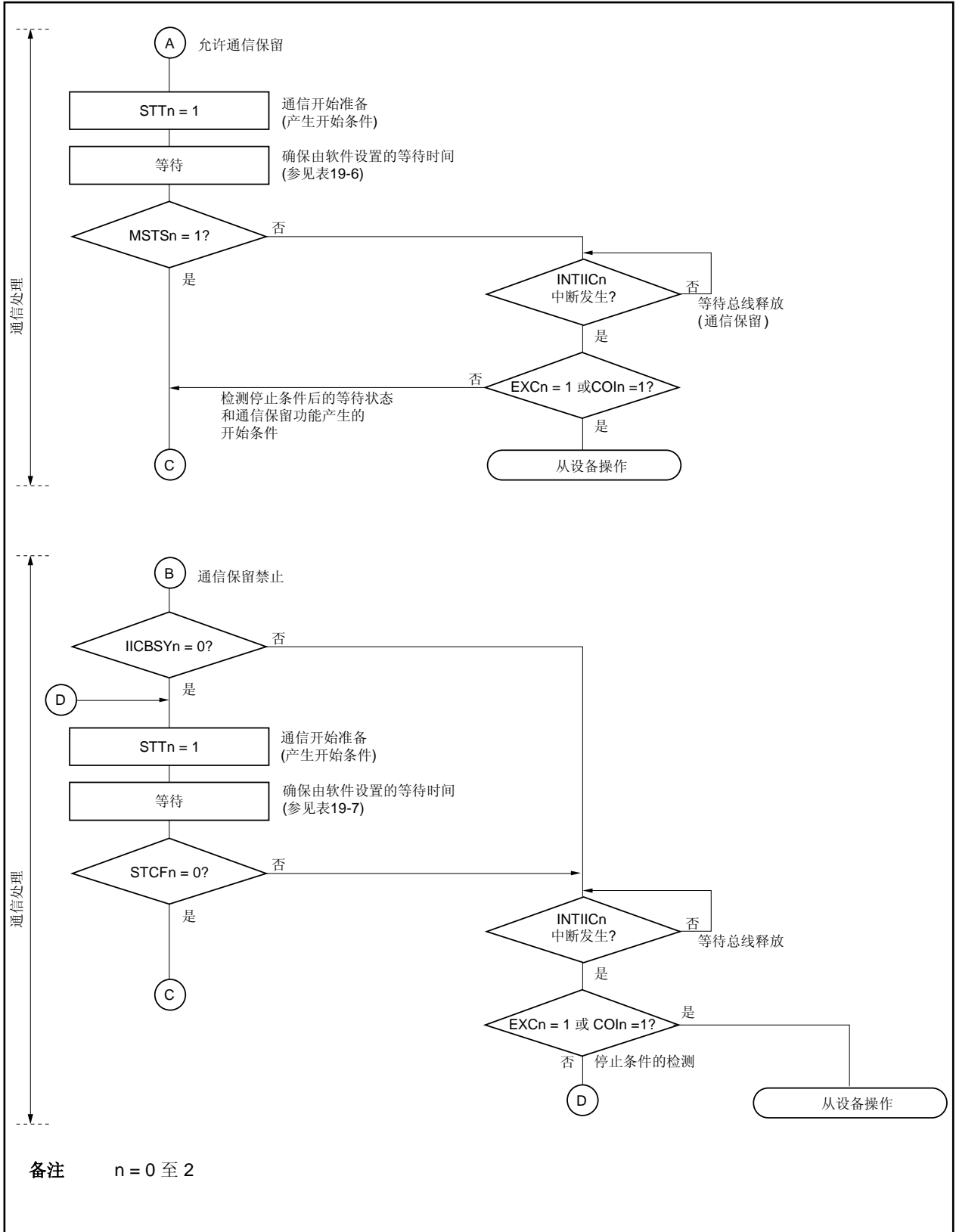
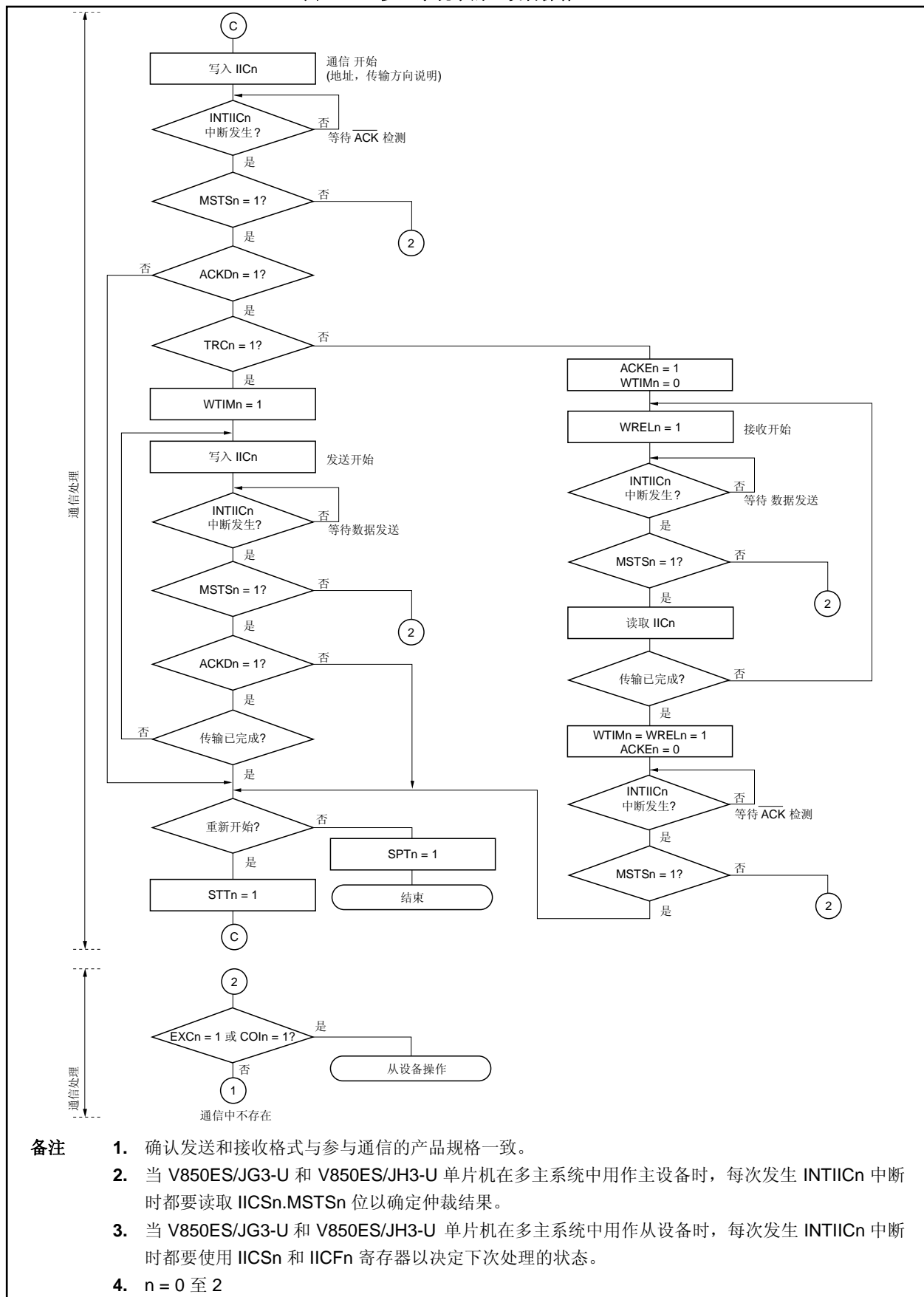


图 19-19. 多主系统中的主设备操作 (3/3)



备注

1. 确认发送和接收格式与参与通信的产品规格一致。
2. 当 V850ES/JG3-U 和 V850ES/JH3-U 单片机在多主系统中用作主设备时，每次发生 INTIICn 中断时都要读取 IICSn.MSTSn 位以确定仲裁结果。
3. 当 V850ES/JG3-U 和 V850ES/JH3-U 单片机在多主系统中用作从设备时，每次发生 INTIICn 中断时都要使用 IICSn 和 IICFn 寄存器以决定下次处理的状态。
4. n = 0 至 2

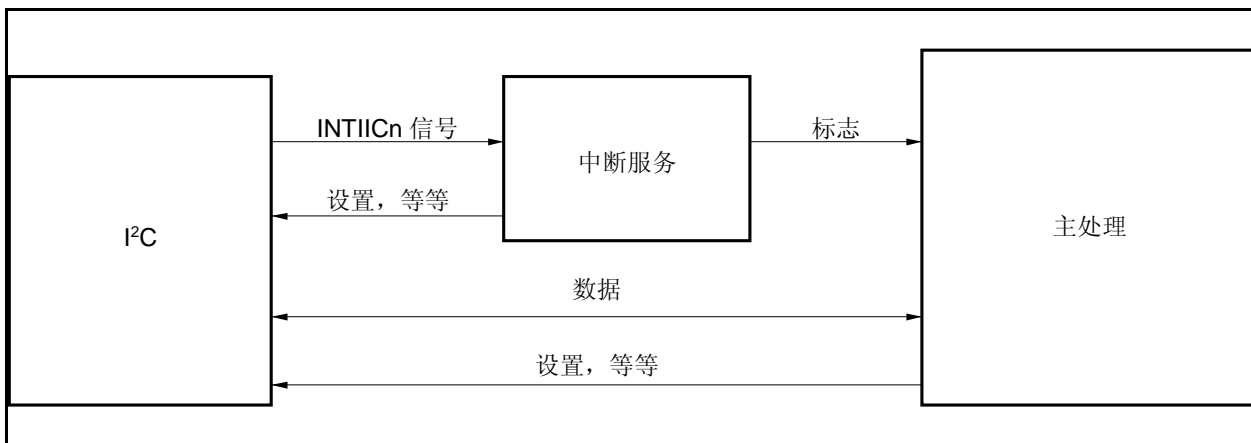
19.16.3 从设备操作

以下展示从设备操作的处理过程。

基本而言，从设备的操作都是由事件驱动的，所以，必需通过 INTIICn 中断进行处理（处理需要操作状态的重大改变，比如，通信过程中检测到停止条件）。

以下描述假定数据通信不支持扩展码，还假设只有在状态改变处理时才执行 INTIICn 中断服务，并且在主程序处理期间执行实际的数据通信。

图 19-20. 从设备操作期间的软件概况



所以，系统准备了以下 3 个标志，以便将其传送给主程序，从而执行数据传输处理，而不使用 INTIICn 信号。

(1) 通信模式标志

该标志表示以下通信状态：

清除模式： 未进行数据通信

通信模式： 正在进行数据通信（检测到有效地址，检测到停止条件，未检测到来自主设备的 $\overline{\text{ACK}}$ 以及地址不匹配）

(2) 就绪标志

该标志表示使能数据通信。这和正常数据传输期间发生 INTIICn 中断的状态相同。该标志在中断处理模块内被置位，在主程序模块中被清除。第一个数据发送的就绪标志并不在中断处理模块内置位，所以，第一个数据发送后不需要清除处理（地址匹配被视为下一个数据的请求）。

(3) 通信方向标志

该标志表示数据通信的方向，和 IICSn.TRCn 位的值相同。

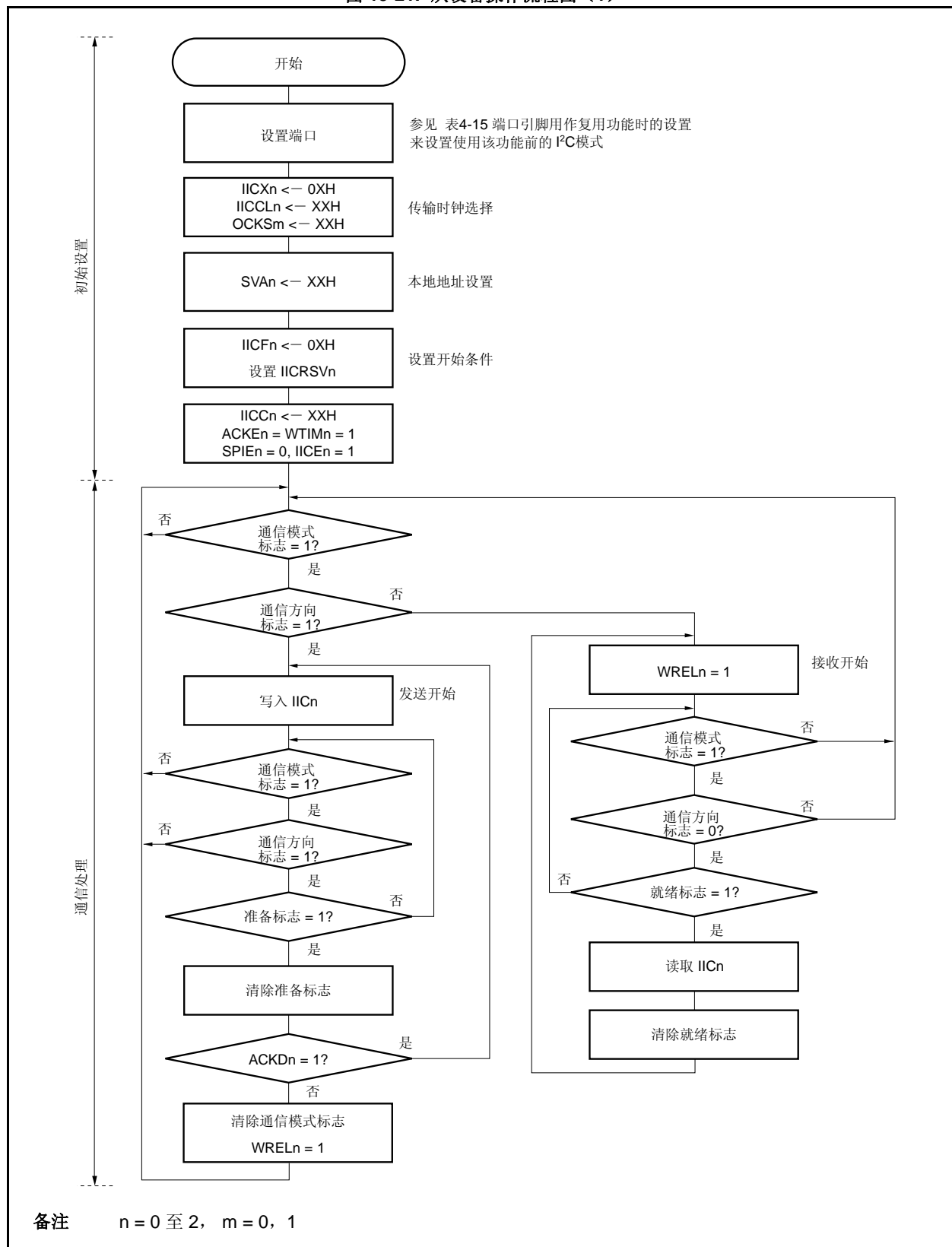
下面展示了从设备操作期间主程序模块的操作。

启动 I²C0n 并且等待通信使能状态。当通信被使能后，使用通信模式标志和就绪标志执行传输（停止条件和开始条件的处理由中断处理执行，条件通过标志位确认）。

对于发送操作，重复发送直到主设备停止返回 $\overline{\text{ACK}}$ 信号，当主设备停止返回 $\overline{\text{ACK}}$ 信号时，传输完毕。

对于接收操作，接收指定数量的数据，且在传输完成后，不为下一个数据立即返回 $\overline{\text{ACK}}$ 信号。此后，主设备产生停止条件或重新开始条件。这将导致退出通信。

图 19-21. 从设备操作流程图中 (1)

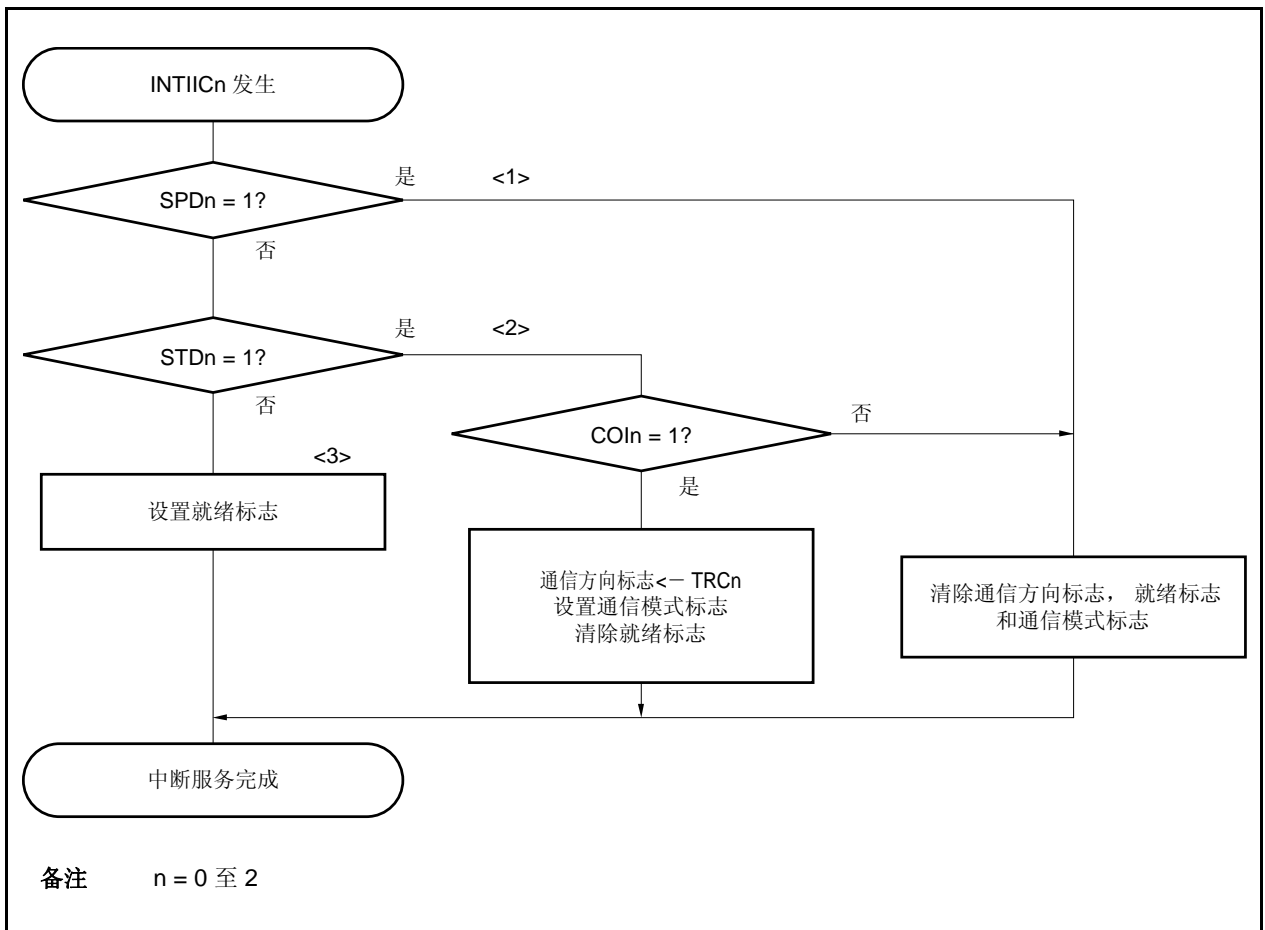


下面展示了从设备使用 INTIICn 中断进行处理的一个示例（这里假定不使用扩展码）。在 INTIICn 中断服务期间，状态得以确认并且执行下列步骤。

- <1> 当检测到停止条件时，通信终止。
- <2> 当检测到开始条件后，进行地址确认。如果地址不匹配，则通信终止。如果地址匹配，设置通信模式并且释放等待状态，从中断中返回操作处理（就绪标志被清除）。
- <3> 对于数据的发送/接收过程，当就绪标志置位时，I²C0n 总线保持为等待状态，从中断中返回操作处理。

备注 上述<1>至<3>步骤对应图 19-22 从设备操作流程（2）的<1>至<3>步骤。

图 19-22. 从设备操作流程（2）



19.17 数据通信的时序

当使用 I²C 总线模式时，主设备通过串行总线输出一个地址，在多个从设备中选择其中一个作为自己的通信伙伴。输出从设备地址以后，主设备发送 IICSn.TRCn 位指定数据传输方向，然后，开始与从设备进行串行通信。

IICn 寄存器的移位操作和串行时钟引脚（SCL0n）信号的下降沿同步。发送数据被传送到 SO 锁存器并经由 SDA0n 引脚输出（MSB 先行）。

由 SDA0n 引脚输入的数据在 SCL0n 引脚信号的上升沿被 IICn 寄存器捕获。
数据通信时序如下所示。

备注 n = 0 至 2

图 19-23. 主设备向从设备传输示例
(当主设备和从设备都选择 9 时钟等待时) (1/3)

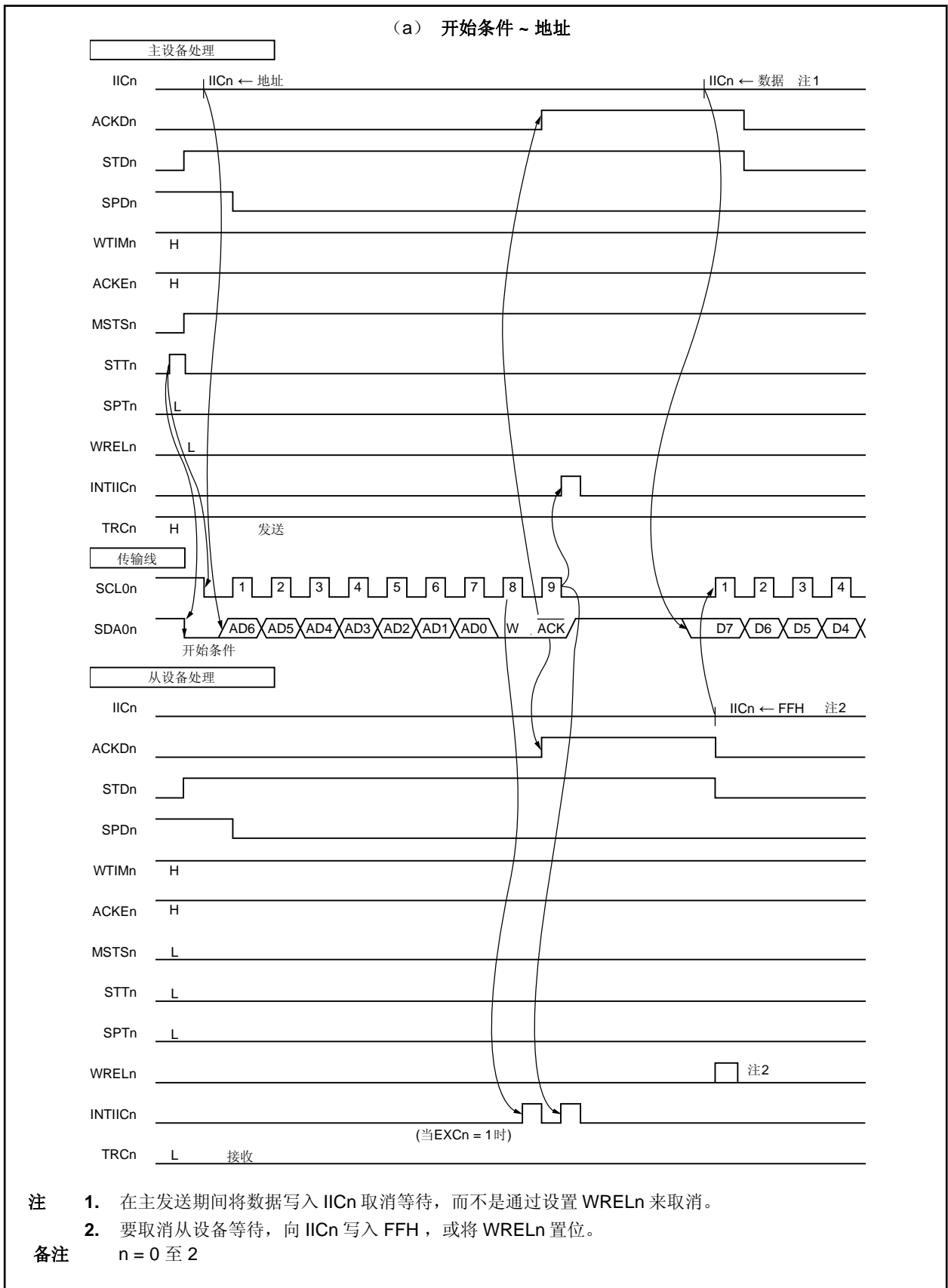


图 19-23. 主设备向从设备传输示例
(当主设备和从设备都选择 9 时钟等待时) (2/3)

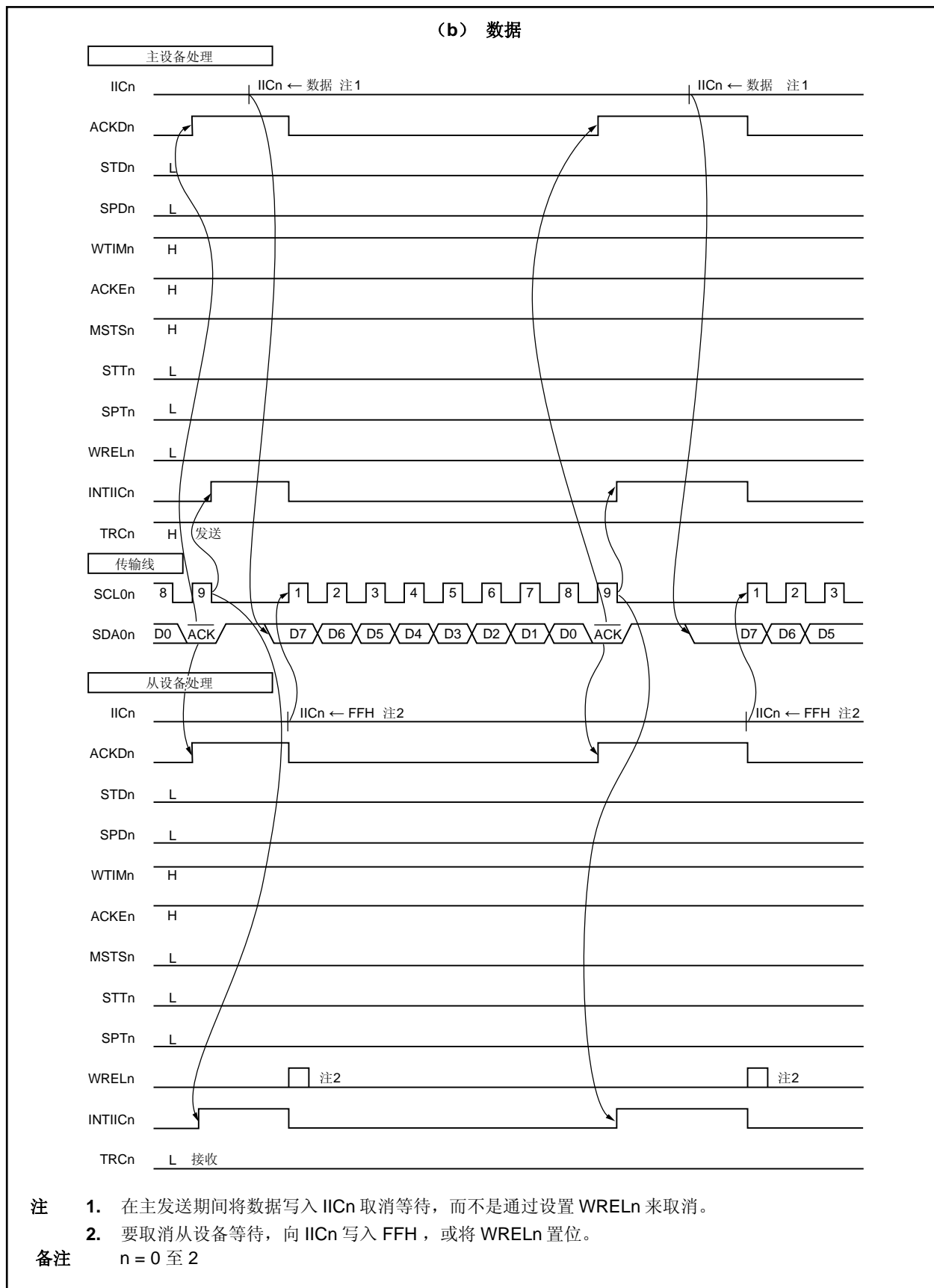


图 19-23. 主设备向从设备传输示例
(当主设备和从设备都选择 9 时钟等待时) (3/3)

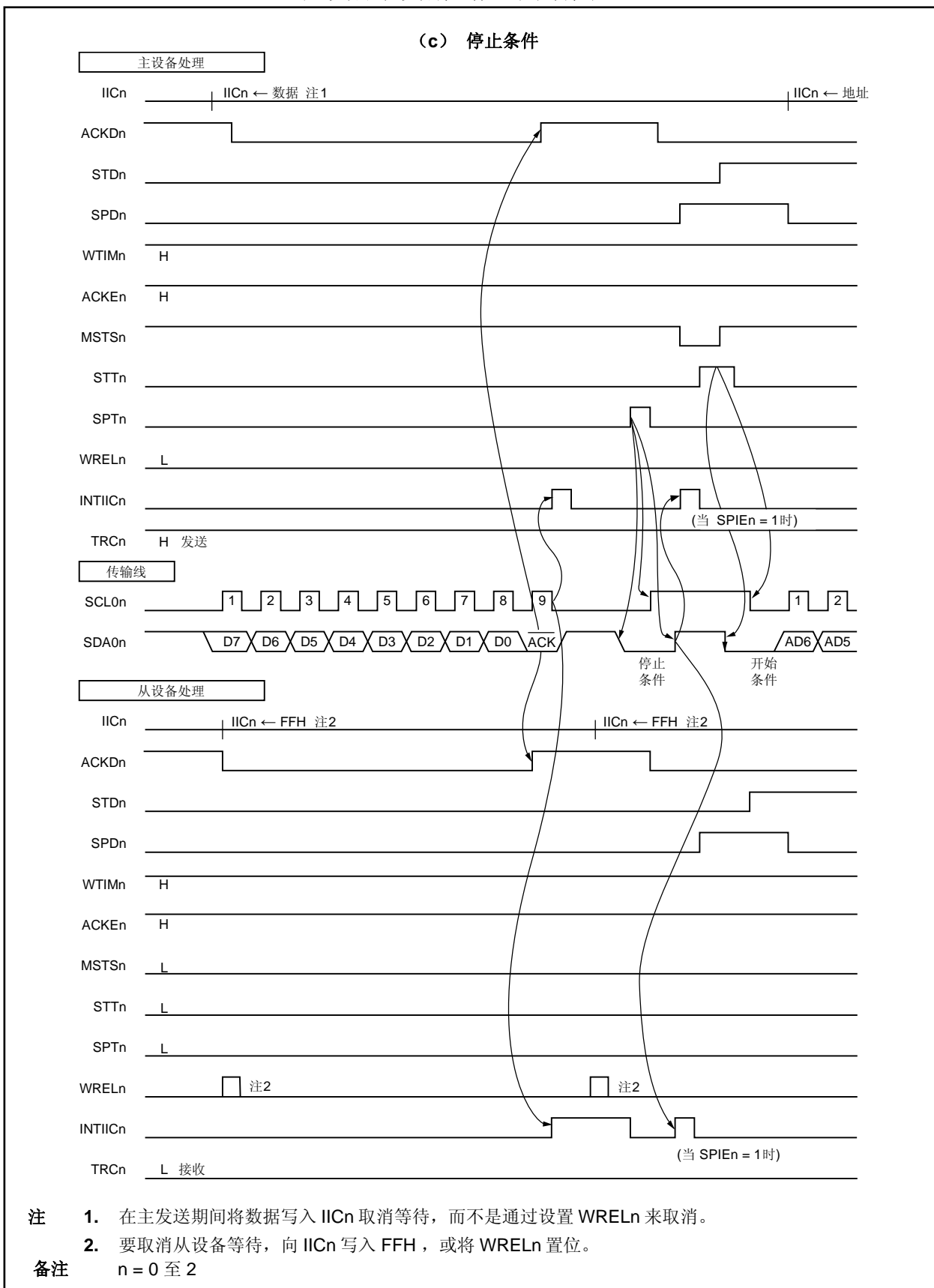


图 19-24. 从设备向主设备传输示例
(选择主设备 8 时钟等待, 从设备 9 时钟等待) (1/3)

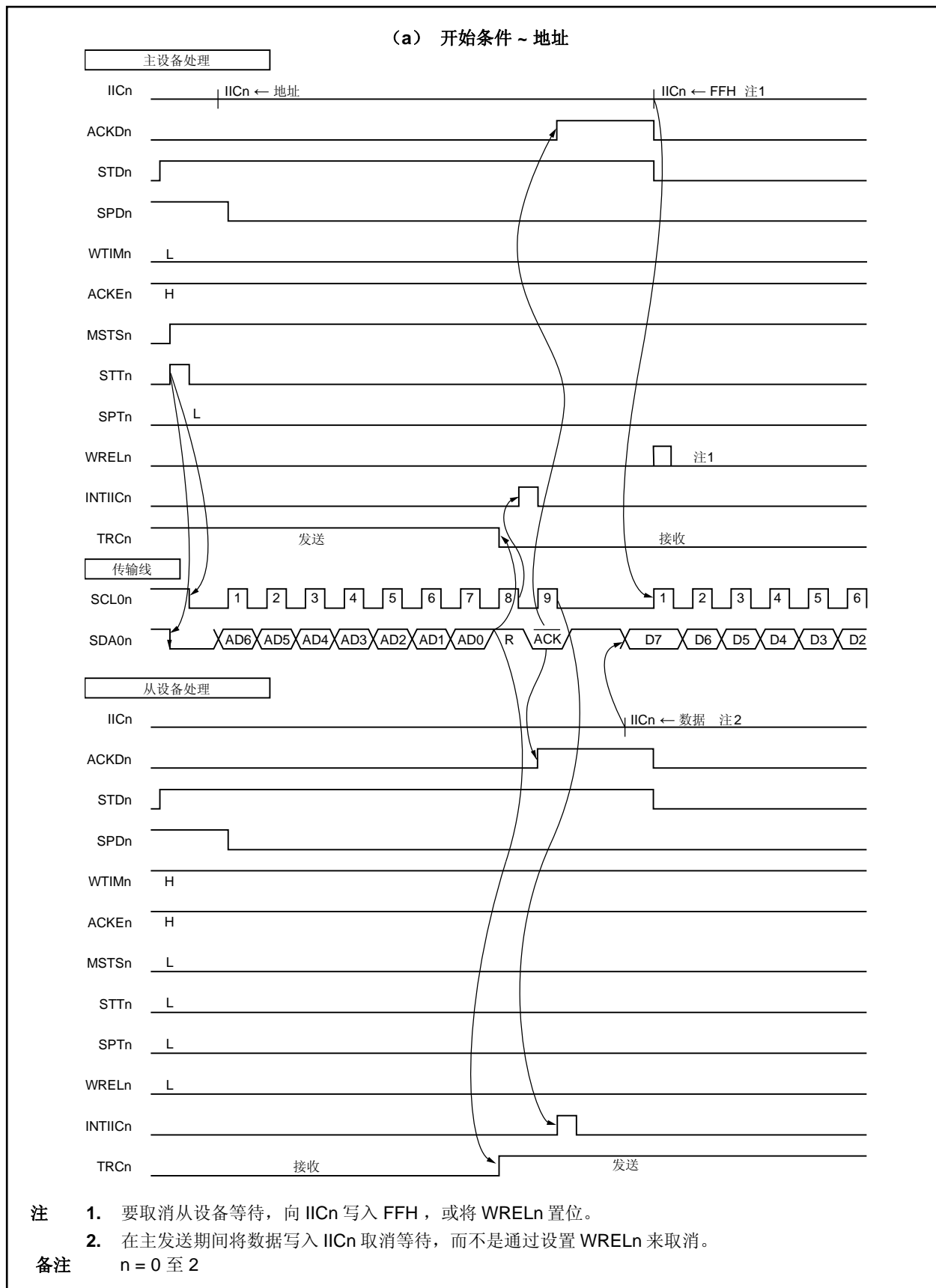


图 19-24. 从设备向主设备传输示例
(选择主设备 8 时钟等待, 从设备 9 时钟等待) (2/3)

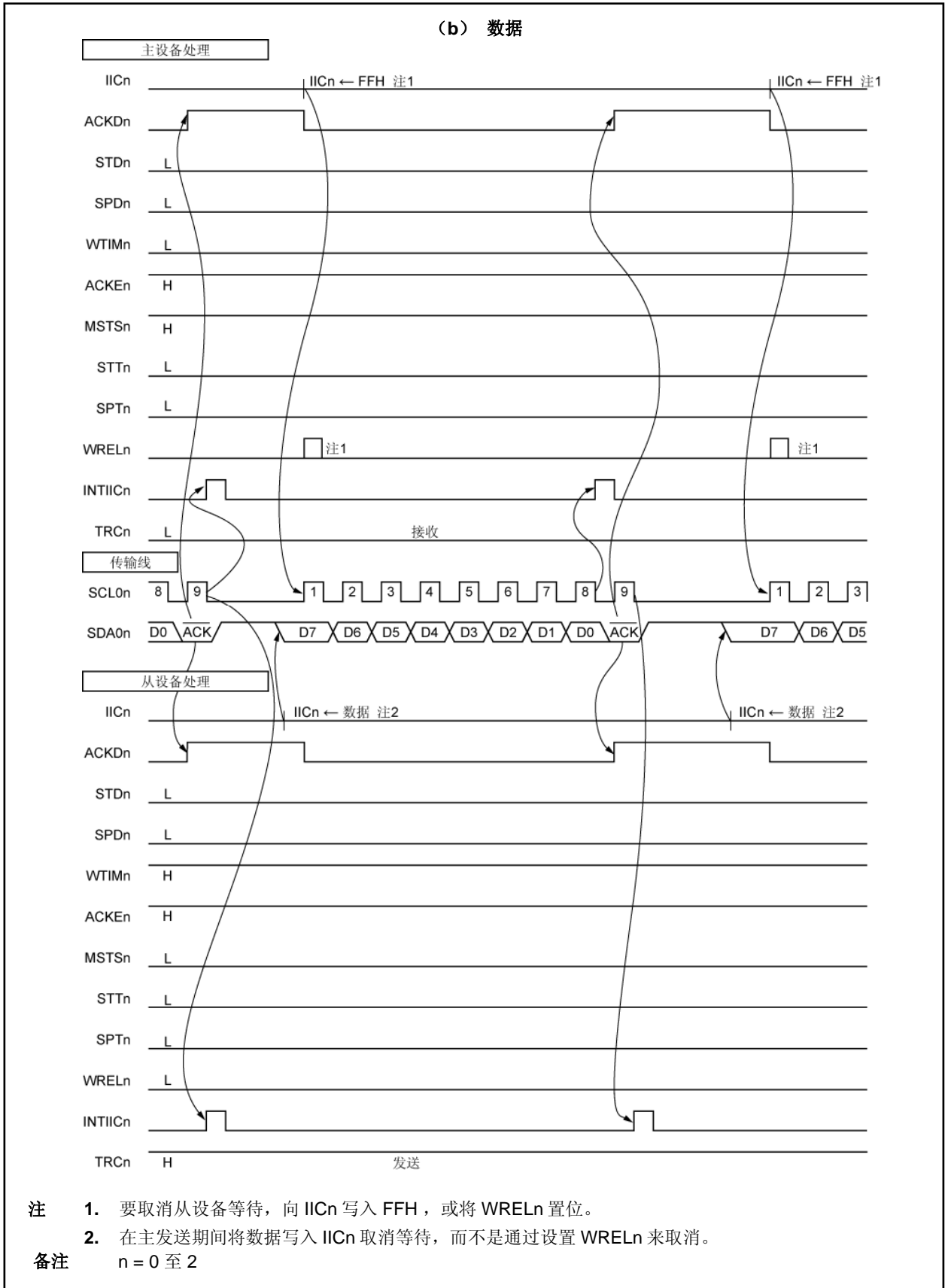
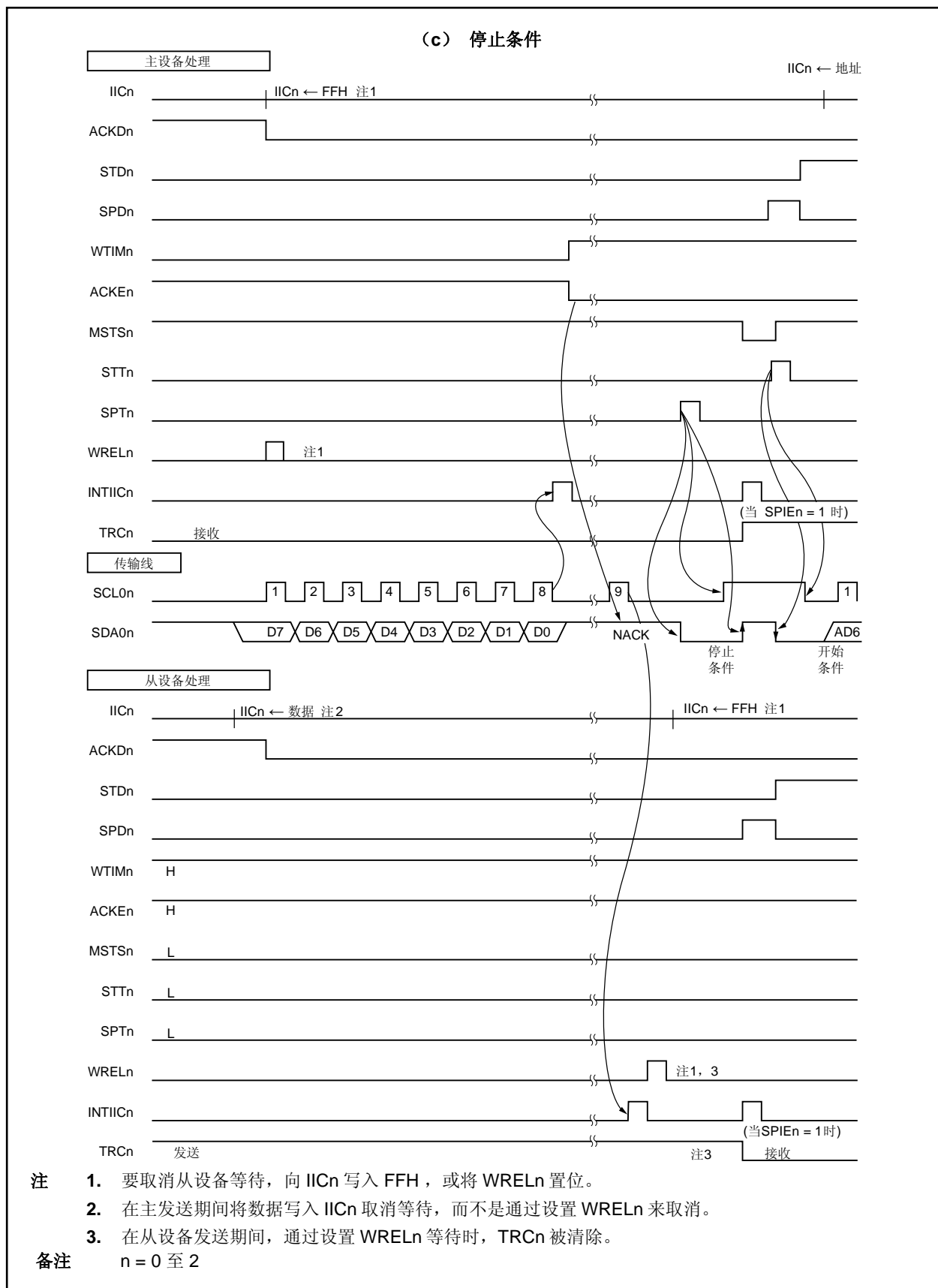


图 19-24. 从设备向主设备传输示例
(选择主设备 8 时钟等待, 从设备 9 时钟等待) (3/3)



第二十章 USB功能控制器USBF

V850ES/JG3-U 和 V850ES/JH3-U 都内置有符合通用串行总线规范的 USB 功能控制器(USBF)。USB 功能控制器和外部主机设备之间使用基于令牌的协议，通过轮询方法进行数据通讯。

20.1 综述

- 符合通用串行总线规范
- 支持 12 Mbps（全速）传输
- 集成有传输端点

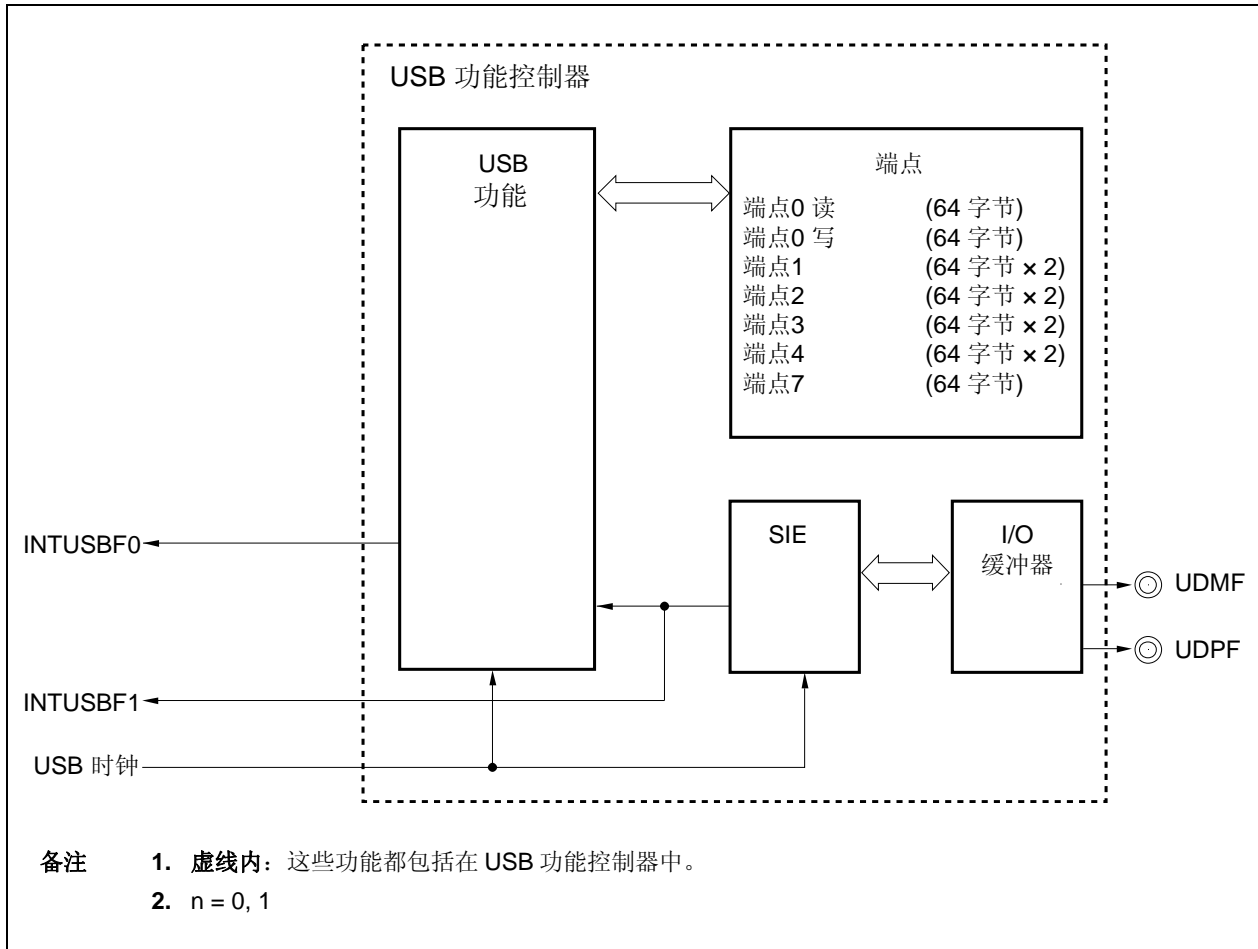
端点名称	FIFO 空间大小（字节）	传输类型	备注
端点 0 读	64	控制传输	-
端点 0 写	64	控制传输	-
端点 1	64 × 2	批量 1 传输（IN）	2 缓冲区配置
端点 2	64 × 2	批量 1 传输（OUT）	2 缓冲区配置
端点 3	64 × 2	批量 2 传输（IN）	2 缓冲区配置
端点 4	64 × 2	批量 2 传输（OUT）	2 缓冲区配置
端点 7	8	中断传输	-

- 批量传输(IN/OUT)可以作为 DMA 传输来执行(2 周期单次传输模式)
- 时钟：可以选择内部时钟(6 MHz 外部时钟 × 内部时钟 8 倍频 = 48 MHz 内部时钟)或外部时钟(外部时钟输入到 UCLK 引脚($f_{USB} = 48 \text{ MHz}$))。

20.2 配置

20.2.1 框图

图 20-1. USB 功能控制器的框图



20.2.2 USB存储器映射

对于 CPU 来说，USB 功能控制器被分配到微控制器内的 CS1 空间。为了方便使用，该存储器空间的划分如下。

表 20-1. CPU 存储器空间的划分

地址	区域	
00200000H 至 00200092H	EPC 控制寄存器区域	
00200100H 至 00200114H	EPC 数据保持寄存器区域	
00200144H 至 002003C4H	EPC 请求数据寄存器区域	
00200400H 至 00200408H	桥寄存器区域	
00200500H 至 0020050EH	DMA 寄存器区域	
00201000H	批量-in 寄存器区域	EP1 (批量-IN1)
00202000H		EP3 (批量-IN2)
00210000H	批量-out 寄存器区域	EP2 (批量-Out1)
00220000H		EP4 (批量-Out2)
00240000H	外设控制寄存器区域	

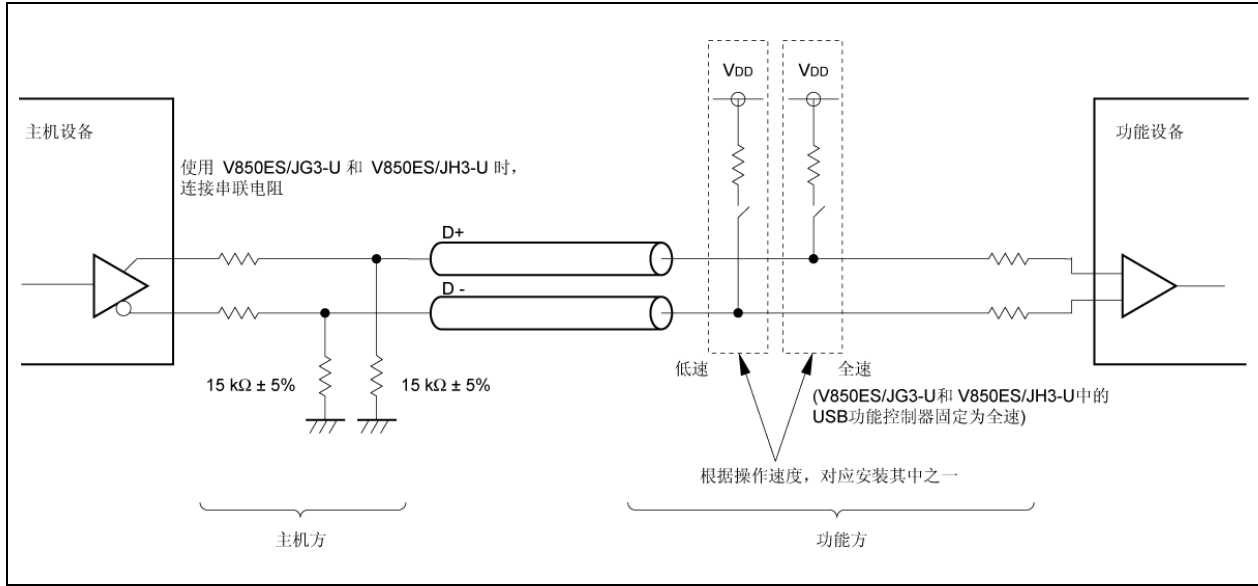
20.3 外部电路配置

20.3.1 概述

在 USB 传输中，当主机控制器和功能控制器互相进行通讯时，USB 信号(D+/D-)必须连接上拉/下拉电阻，以便识别通讯参与方。并且在 V850ES/JG3-U 和 V850ES/JH3-U 中，还必须连接串联电阻。

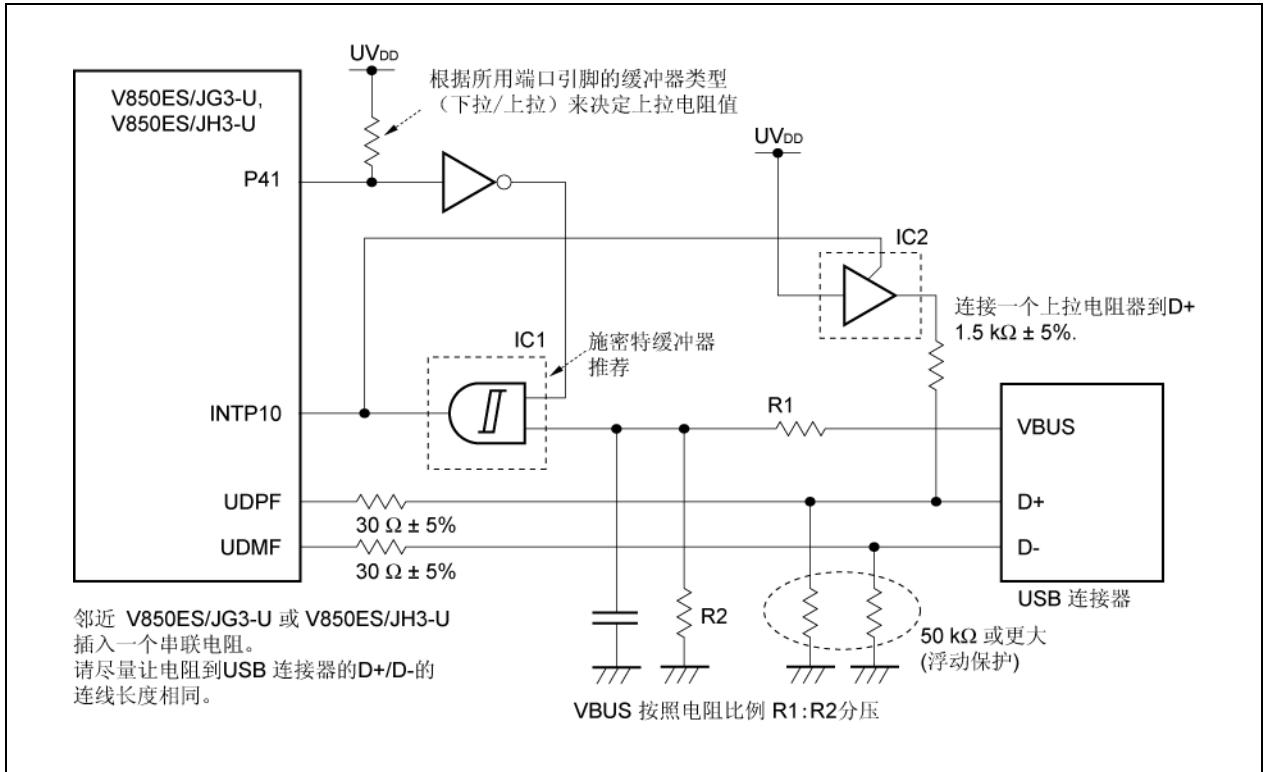
下图展示了 USB 传输线的概要配置。外部配置的详情参见各个章节的描述。

图 20-2. USB 传输线中的上拉，下拉串联电阻概要配置



20.3.2 连接配置

图 20-3. USB 功能控制器连接示例



(1) 连接到 D+/D-的串联电阻

连接 30 Ω ±5%串联电阻到 V850ES/JG3-U 和 V850ES/JH3-U 内置 USB 功能控制器的 D+/D- 引脚 (UFDP, UDFM)。如果没有连接串联电阻, 阻抗值达不到要求, 输出波形可能会受到干扰。

串联电阻的位置要邻近 V850ES/JG3-U 或 V850ES/JH3-U, 各串联电阻和 USB 连接器之间的连线长度相同, 这样使得 D+ 和 D- 的阻抗相等 (推荐使用 90 Ω ±5%的电阻)。

(2) D+的上拉控制

因为 V850ES/JG3-U 和 V850ES/JH3-U 的功能控制器被固定位全速(FS), 请确保通过 1.5 kΩ ±5%将 D+ 引脚 (UFDP)上拉到 UVDD。

为了禁止连接报告(D+ 上拉) 到 USB 主机/HUB (比如在高优先级服务或初始化期间), 在系统中通过一个通用端口来控制 D+的上拉电阻。对于图 20-3 中所示的电路, 通过一个通用端口和 USB 线 VBUS (与"AND"电路)来控制上拉控制信号和 D+引脚的 VBUS 输入信号。在图 20-3 中, 如果通用端口是高电平, D+的上拉被禁止 (在此情况下, 因为通用端口引脚被默认设置为输入端口, 通过上拉保证高电平)。

对于图 20-3 中使用的 IC2, 当系统掉电时, 对该 IC 的电压可以维持。

(3) 检测 USB 线连接/断开

USB 功能控制器(USBF)需要一个 VBUS 输入信号来确认 USB 线是连接或断开, 因为 USBF 的状态由硬件控制。当 USBF 掉电时, USB 线 VBUS 连接到 USB 主机或 HUB, 来自 USB 主机或 HUB 的电压(5 V)被提供作为 VBUS 输入信号。于是, 图 20-3 使用的 IC1, 当系统掉电时, 对该 IC 的电压可以维持。断开图 20-3 中电路的 USB 线, 当 VBUS 电压下降时, 输入到 INTP10 的信号可能会不稳定。因此图 20-3 中的 IC1, 推荐使用施密特缓冲器。

(4) 初始化期间或当 USBF 不使用时的浮动保护

当 USB 功能控制器被初始化或不使用时, 为了防止出现浮动状态, 使用 50 k Ω 或更大的电阻将 D+/D- 引脚下拉。

20.4 注意事项

(1) 时钟准确度

为了操作 USB 功能控制器，USB 时钟必须使用内部时钟 (6 MHz 外部时钟 \times 内部时钟 8 倍频 = 48 MHz 内部时钟) 或外部时钟(输入到 UCLK 引脚的外部时钟 ($f_{\text{USB}} = 48 \text{ MHz}$))。当内部时钟用作 USB 时钟，使用的谐振器准确度 6 MHz ± 500 ppm (最大值)。当使用外部时钟，UCLK 引脚使用的时钟准确度 48 MHz ± 500 ppm (最大值)。如果 USB 时钟的准确度下降，传输数据可能无法满足 USB 规范要求。

(2) HUB 连接

对于 HUB 连接，推荐使用外部时钟(晶体谐振器)。如果使用内部时钟，因为微控制器内部电路上的时钟抖动，可能无法满足 USB 规范要求。当使用内部时钟，推荐按照 1 比 1 基准来连接主机设备。

20.5 请求

USB 标准有请求命令，将来自主机设备的请求报告给功能设备，来执行响应过程。

请求是在控制传输的建立 (SETUP) 阶段接收到的，大部分可以通过 USB 功能控制器(USBF)的硬件自动处理。

20.5.1 自动请求

(1) 译码

下表列出了请求和译码值之间的请求格式以及对应。

表 20-2. 请求格式

偏移量	域名称	
0	bmRequestType	
1	bRequest	
2	wValue	低位端
3		高位端
4	wIndex	低位端
5		高位端
6	wLength	低位端
7		高位端

表 20-3. 请求和译码值之间的对应

请求	偏移量	译码值							响应			数据阶段	
		bmRequestType	bRequest	wValue		wIndex		wLength		Df	Ad		Cf
		0	1	3	2	5	4	7	6				
GET_INTERFACE	81H	0AH	00H	00H	00H	0nH	00H	01H	STALL	STALL	ACK NAK	√	
GET_CONFIGURATION	80H	08H	00H	00H	00H	00H	00H	01H	ACK NAK	ACK NAK	ACK NAK	√	
GET_DESCRIPTOR 设备	80H	06H	01H	00H	00H	00H	XXH	XXH ^{#1}	ACK NAK	ACK NAK	ACK NAK	√	
GET_DESCRIPTOR 配置	80H	06H	02H	00H	00H	00H	XXH	XXH ^{#1}	ACK NAK	ACK NAK	ACK NAK	√	
GET_STATUS 设备	80H	00H	00H	00H	00H	00H	00H	02H	ACK NAK	ACK NAK	ACK NAK	√	
GET_STATUS 端点 0	82H	00H	00H	00H	00H	00H 80H	00H	02H	ACK NAK	ACK NAK	ACK NAK	√	
GET_STATUS 端点 X	82H	00H	00H	00H	00H	\$\$H	00H	02H	STALL	STALL	ACK NAK	√	
CLEAR_FEATURE 设备 ^{#2}	00H	01H	00H	01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×	
CLEAR_FEATURE 端点 0 ^{#2}	02H	01H	00H	00H	00H	00H 80H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×	
CLEAR_FEATURE 端点 X ^{#2}	02H	01H	00H	00H	00H	\$\$H	00H	00H	STALL	STALL	ACK NAK	×	
SET_FEATURE 设备 ^{#3}	00H	03H	00H	01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×	
SET_FEATURE 端点 0 ^{#3}	02H	03H	00H	00H	00H	00H 80H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×	
SET_FEATURE 端点 X ^{#3}	02H	03H	00H	00H	00H	\$\$H	00H	00H	STALL	STALL	ACK NAK	×	
SET_INTERFACE	01H	0BH	00H	0#H	00H	0?H	00H	00H	STALL	STALL	ACK NAK	×	
SET_CONFIGURATION ^{#4}	00H	09H	00H	00H 01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×	
SET_ADDRESS	00H	05H	XXH	XXH	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×	

备注 √: 数据阶段
×: 无数据阶段

- 注
1. 如果 wLength 值比准备的值小, wLength 值被返回; 如果 wLength 值比准备的值大, 准备的值被返回。
 2. 在状态阶段接收到 ACK 时, CLEAR_FEATURE 请求会清除 UF0 设备状态寄存器 L (UF0DSTL) 和 UF0EPn 状态寄存器 L (UF0EnSL) (n = 0 到 4, 7)。

- 注
3. 在状态阶段中接收到 ACK 时, SET_FEATURE 请求会设置 UF0 设备状态寄存器 L (UF0DSTL) 和 UF0 EPn 状态寄存器 L (UF0EnSL) (n = 0 到 4, 7)。在 GET_STATUS 端点 0 请求、SET_FEATURE 端点 0 请求以及由 CPUDEC 中断请求产生的请求以外, 请求的控制传输数据阶段或状态阶段中, 如果在接收到 CLEAR_FEATURE 端点 0 请求之前, UF0E0SL 寄存器的 E0HALT 位被置位, 产生 STALL 响应。对于不支持的请求的 STALL 响应, 不会将 UF0E0SL 寄存器的 E0HALT 位置 1, 并且只要接收到下一个建立 (SETUP) 令牌, STALL 响应立即被清除。
 4. 如果 wValue 不是默认值, 产生一个自动 STALL 响应。

注意事项 1. 通用串行总线规范定义的控制传输的顺序在下列条件下不能满足。在这些条件下, 操作无法保证。

- 如果没有经历 SETUP 阶段, 突然接收到一个 IN / OUT 令牌
 - 如果在 SETUP 阶段的数据阶段发送 DATA PID1
 - 如果接收到 128 个地址或更多地址的令牌
 - 如果在 SETUP 阶段发送的请求数据少于 8 字节
2. 在状态阶段中, 即使主机发送 Null 包以外的数据, 也会产生 ACK 响应。
 3. 如果在 FW 处理的控制传输 (读) 期间, wLength 值是 00H, 对于控制传输 (无数据), 会自动发送一个 Null 包。FW 请求不会自动发送 Null 包。

备注

1. Df: 默认状态, Ad: 地址状态, Cf: 配置状态
2. n = 0 到 4
根据目标的接口编号是否有效, UF0 有效接口编号寄存器 (UF0AIFN) 的设置决定接口编号为 1 到 4 的请求是否被正确响应。
3. \$\$: 包含传输方向的有效端点号
有效的端点由当前置位的复用设置号码决定 (参见 20.6.3(36) UF0 有效复用设置寄存器 (UF0AAS), (38) UF0 端点 1 接口映射寄存器 (UF0E1IM) 至(42) UF0 端点 7 接口映射寄存器 (UF0E7IM))。
4. ? 和 #: 从主机发送的值 (接口编号 0 到 4 的信息)
根据目标的接口编号和复用设置是否有效, UF0 有效接口编号寄存器 (UF0AIFN) 和 UF0 有效复用设置寄存器 (UF0AAS) 可以判定对应每个接口编号的复用设置请求是否被正确响应。

(2) 处理

默认状态、地址状态和配置状态下的自动请求处理如下所示。

备注 默认状态: 该状态下, 操作在默认地址上执行
 地址状态: 地址被分配后的状态
 配置状态: 正确接收到 SET_CONFIGURATION wValue = 1 后的状态

(a) CLEAR_FEATURE() 请求

如果不能清除 CLEAR_FEATURE() 请求, 如果 FEATURE 不存在, 或者如果目标是一个不存在的接口或端点, 在状态阶段产生 STALL 响应。如果 wLength 值非 0, 也会产生 STALL 响应。

- 默认状态: 只有当目标是对设备或端点 0 的请求时, 接收到 CLEAR_FEATURE() 请求, 才会产生正确响应; 否则, 在状态阶段产生 STALL 响应。
- 地址状态: 只有当目标是对设备或端点 0 的请求时, 接收到 CLEAR_FEATURE() 请求, 才会产生正确响应; 否则, 在状态阶段产生 STALL 响应。
- 配置状态: 只有当目标是对一个存在的设备或端点的请求时, 接收到 CLEAR_FEATURE() 请求, 才会产生正确响应; 否则, 在状态阶段产生 STALL 响应。

当 CLEAR_FEATURE() 请求已经被正确处理, UF0CLR 请求寄存器 (UF0CLR) 的对应位被置 1, UF0EPn 状态寄存器 L (UF0EnSL) 的 EnHALT 位被清除为 0, 并且发出一个中断 (n = 0 到 4, 7)。如果在对象是端点时接收到 CLEAR_FEATURE() 请求, 对应端点的双向开关位 (控制 DATA0 和 DATA1 之间的切换) 总是被重置为 DATA0。

(b) GET_CONFIGURATION() 请求

如果 wValue、wIndex 或 wLength 中的任意一个不在表 20-3 列举的范围内, 在数据阶段产生一个 STALL 响应。

- 默认状态: 当接收到 GET_CONFIGURATION() 请求时, 保存在 UF0 配置寄存器 (UF0CNF) 中的值被返回。
- 地址状态: 当接收到 GET_CONFIGURATION() 请求时, 保存在 UF0CNF 寄存器中的值被返回。
- 配置状态: 当接收到 GET_CONFIGURATION() 请求时, 保存在 UF0CNF 寄存器中的值被返回。

(c) GET_DESCRIPTOR() 请求

如果对象描述符的长度是 `wMaxPacketSize` 的倍数，返回一个 `Null` 包表示数据阶段的结束。如果此时描述符的长度小于 `wLength` 值，返回整个描述符；如果描述符的长度大于 `wLength` 值，返回 `wLength` 值长度的描述符。

- 默认状态: 当接收到 `GET_DESCRIPTOR()` 请求时，保存在 `UF0` 设备描述符寄存器 `n` (`UF0DDn`) 和 `UF0` 配置 / 接口 / 端点描述符寄存器 `m` (`UF0CIEm`) (`n = 0` 至 `17`, `m = 0` 至 `255`) 中的值被返回。
- 地址状态: 当接收到 `GET_DESCRIPTOR()` 请求时，保存在 `UF0DDn` 寄存器和 `UF0CIEm` 寄存器中的值被返回。
- 配置状态: 当接收到 `GET_DESCRIPTOR()` 请求时，保存在 `UF0DDn` 寄存器和 `UF0CIEm` 寄存器中的值被返回。

`UF0CIEm` 寄存器中可以存储最大 256 字节的描述符。要返回的描述符多于 256 字节，需要设置 `UF0MODC` 寄存器的 `CDCGDST` 位为 1，并通过 `FW` 来处理 `GET_DESCRIPTOR()` 请求。

在 `UF0` 描述符长度寄存器 (`UF0DSCL`) 中存储 `UF0CIEm` 寄存器设置的描述符的总字节数的值 - 1。传输数据由这个数据的值 + 1 和 `wLength` 控制。

(d) GET_INTERFACE() 请求

如果 `wValue` 和 `wLength` 中有一个不在表 20-3 列举的范围内，或者如果 `wIndex` 不是 `UF0` 有效接口编号寄存器 (`UF0AIFN`) 设置的值，在数据阶段产生一个 `STALL` 响应。

- 默认状态: 当接收到 `GET_INTERFACE()` 请求时，在数据阶段产生一个 `STALL` 响应。
- 地址状态: 当接收到 `GET_INTERFACE()` 请求时，在数据阶段产生一个 `STALL` 响应。
- 配置状态: 当接收到 `GET_INTERFACE()` 请求时，存储在 `UF0` 接口 `n` 寄存器 (`UF0IFn`) 中对应 `wIndex` 值的值被返回 (`n = 0` 到 `4`)。

(e) GET_STATUS() 请求

如果 wValue、wIndex 或 wLength 中的任意一个不在表 20-3 列举的范围内，在数据阶段产生一个 STALL 响应。如果目标是不存在的接口或端点，在数据阶段产生也会一个 STALL 响应。

- 默认状态: 只有当接收到 GET_STATUS()请求，并且该请求是针对设备或端点 0 时，保存在目标状态寄存器^注中的值被返回；否则，在数据阶段产生 STALL 响应。
- 地址状态: 只有当接收到 GET_STATUS()请求，并且该请求是针对设备或端点 0 时，保存在目标状态寄存器^注中的值被返回；否则，在数据阶段产生 STALL 响应。
- 配置状态: 只有当接收到 GET_STATUS()请求，并且该请求是针对一个存在的设备或端点时，保存在目标状态寄存器^注中的值被返回；否则，在数据阶段产生 STALL 响应。

注 目标状态寄存器如下所示。

- 如果目标是一个设备: UF0 设备状态寄存器 L (UF0DSTL)
- 如果目标是端点 0: UF0 EP0 状态寄存器 L (UF0E0SL)
- 如果目标是端点 n: UF0 EPn 状态寄存器 L (UF0EnSL) (n = 1 到 4, 7)

(f) SET_ADDRESS() 请求

如果 wIndex 或 wLength 中有一个不在表 20-3 列举的范围内，在数据阶段产生一个 STALL 响应。如果指定的设备地址大于 127，在数据阶段也会产生一个 STALL 响应。

- 默认状态: 当接收到 SET_ADDRESS()请求时，如果指定的地址不是 0，设备进入地址状态，并且将要输入到 SIE 的 USB 地址值更改为一个指定的地址值。如果指定的地址是 0，设备仍然保持为默认状态。
- 地址状态: 当接收到 SET_ADDRESS()请求时，如果指定的地址是 0，设备进入默认状态，并且将要输入到 SIE 的 USB 地址值返回为默认地址。如果指定的地址不是 0，设备仍然保持为地址状态，并且将要输入 SIE 的 USB 地址值更改为一个指定的新地址值。
- 配置状态: 当接收到 SET_ADDRESS()请求时，如果指定的地址是 0，设备仍然处于配置状态，并且将要输入到 SIE 的 USB 地址值返回为默认地址。在这种情况下，这个非 0 的端点仍然有效，并且对端点 0 以外的端点的控制传输 (IN)、控制传输 (OUT)、批量传输和中断传输也可以响应。如果指定的地址不是 0，设备仍然保持为配置状态，并且将要输入 SIE 的 USB 地址值更改为一个指定的新地址值。

(g) SET_CONFIGURATION() 请求

如果 wValue、wIndex 或 wLength 中的任意一个不在表 20-3 列举的范围内，在数据阶段产生一个 STALL 响应。

- **默认状态:** 当接收到 SET_CONFIGURATION()请求时，如果指定的配置值是 1，UF0 模式状态寄存器 (UF0MODS) 的 CONF 位和 UF0 配置寄存器 (UF0CNF) 被置为 1。如果指定的配置值是 0，UF0MODS 寄存器的 CONF 位和 UF0CNF 寄存器被清为 0。也就是说，设备跳过地址状态转移到配置状态，在配置状态下响应默认地址。
- **地址状态:** 当接收到 SET_CONFIGURATION()请求时，如果指定的配置值是 1，UF0MODS 寄存器的 CONF 位和 UF0CNF 寄存器被置为 1，并且设备进入配置状态。如果指定的配置值是 0，设备仍然保持为地址状态。
- **配置状态:** 当接收到 SET_CONFIGURATION()请求时，如果指定的配置值是 0，UF0MODS 寄存器的 CONF 位和 UF0CNF 寄存器被置为 1，并且设备返回地址状态。如果指定的配置值是 1，设备仍然保持为配置状态。

如果 SET_CONFIGURATION()请求被正确处理，UF0 SET 请求寄存器 (UF0SET) 的目标位被置为 1，并且发出一个中断。即使指定的配置值与当前的配置值相同，在完成 SET_CONFIGURATION()请求之后，所有的暂停特征 (Halt Feature) 也被清除。如果 SET_CONFIGURATION()请求被正确处理，所有端点的数据切换总是再次被初始化为 DATA0 (被定义为默认状态，复用设置 0，从接收到 SET_CONFIGURATION 请求至接收到 SET_INTERFACE 请求期间被设置位默认状态)。

(h) SET_FEATURE() 请求

如果 SET_FEATURE()请求用于不能被设置或不存在的特征，或者如果目标是不存在的接口或端点，在状态阶段产生一个 STALL 响应。如果 wLength 值不是 0，也会产生一个 STALL 响应。

- **默认状态:** 只有当请求是针对一个设备或端点 0 时，接收到 SET_FEATURE()请求，才会产生正确响应；否则，在状态阶段产生一个 STALL 响应。
- **地址状态:** 只有当请求是针对一个设备或端点 0 时，接收到 SET_FEATURE()请求，才会产生正确响应；否则，在状态阶段产生一个 STALL 响应。
- **配置状态:** 只有当请求是针对一个存在的设备或端点时，接收到 SET_FEATURE()请求，才会产生正确响应；否则，在状态阶段产生一个 STALL 响应。

当 SET_FEATURE()请求已经被正确处理时，UF0 SET 请求寄存器 (UF0SET) 的目标位和 UF0 EPn 状态寄存器 L (UF0EnSL) 的 EnHALT 位被置为 1，并且发出一个中断 (n = 0 到 4, 7)。

(i) SET_INTERFACE() 请求

如果 wLength 不在表 20-3 列举的范围内，如果 wIndex 和 UF0 有效接口编号寄存器 (UF0AIFN) 的设置值不同，或者如果 wValue 值和 UF0 有效复用设置寄存器 (UF0AAS) 的设置值不同，在状态阶段产生一个 STALL 响应。

- 默认状态: 当接收到 SET_INTERFACE()请求时，在状态阶段产生一个 STALL 响应。
- 地址状态: 当接收到 SET_INTERFACE()请求时，在状态阶段产生一个 STALL 响应。
- 配置状态: 当接收到 SET_INTERFACE()请求时，在状态阶段发送 Null 包。

当 SET_INTERFACE()请求被正确处理时，发出一个中断。在 SET_INTERFACE()请求被清除后，所有链接连接到目标接口端点的暂停特征 (Halt Feature) 都被清除。所有与目标接口编号有关的端点的数据切换总是再次被初始化为 DATA0。当前选择的复用设置被正确处理的 SET_INTERFACE()请求更改时，受影响的端点的 FIFO 被完全清除，并且所有相关的中断源也被初始化。

当完成 SET_INTERFACE()请求时，连接到目标接口的所有端点的 FIFO 被清除。同时，暂停特征和 Data PID 被初始化，并且相关的 UF0 INT 状态 n 寄存器 (UF0ISn) 被清除为 0 (n = 0 到 4)。(当 SET_CONFIGURATION 请求被完成时，只有暂停特征和 Data PID 被清除。)

如果在 DMA 传输中，SET_INTERFACE()不支持目标端点，DMA 请求信号马上置 0，SET_INTERFACE()请求已经完成时链接的端点的 FIFO 全部被清除。对 FIFO 清除的结果，通过 DMA 的数据传输没有被正确处理。

20.5.2 其它请求

(1) 响应和处理

下表展示了如何响应和处理其它请求。

表 20-4. 其它请求的响应和处理

请求	响应和处理
GET_DESCRIPTOR 字符串	CPUDEC 中断请求的产生
GET_STATUS 接口	自动 STALL 响应
CLEAR_FEATURE 接口	自动 STALL 响应
SET_FEATURE 接口	自动 STALL 响应
所有 SET_DESCRIPTOR	CPUDEC 中断请求的产生
所有其它请求	CPUDEC 中断请求的产生

20.6 寄存器配置

20.6.1 USB控制寄存器

(1) USB 时钟选择寄存器 (UCKSEL)

UCKSEL 寄存器用于选择 USB 控制器的工作时钟。
 UCKSEL 寄存器可以按字节或按位进行读取或写入。
 系统复位后，该寄存器被置为 00H。

复位后: 00H R/W 地址: FFFFFFF40H

符号	7	6	5	4	3	2	1	0
UCKSEL	0	0	0	0	0	0	UUSEL1	0

UUSEL1	USB控制器的工作时钟的选择
0	从 UCLK 引脚输入的外部时钟($f_{USB} = 48 \text{ MHz}$)
1	主时钟($f_{xx} = 48 \text{ MHz}$)

注意事项 请确保设置第7位至第2位，以及第0位为“0”。

(2) USB 功能控制寄存器 (UFCKMSK)

UFCKMSK 寄存器用于控制 USB 功能控制器操作的使能/禁止。
 UFCKMSK 寄存器可以按字节或按位进行读取或写入。
 系统复位后，该寄存器被置为 03H。

复位后: 03H R/W 地址: FFFFFFF41H

符号	7	6	5	4	3	2	1	0
UFCKMSK	0	0	0	0	0	0	UFBUFMSK	UFMSK

UFBUFMSK	UFMSK	USB功能控制器操作的使能/禁止
0	0	操作使能
1	1	操作禁止

(3) USB 功能选择寄存器 (UHCKMSK)

UHCKMSK 寄存器用于控制使用 USB 控制器功能时只存数据 RAM 的操作。

UHCKMSK 寄存器可以按字节或按位进行读取或写入。

系统复位后，该寄存器被置为 03H。

复位后: 03H R/W 地址: FFFFFFF42H

符号	7	6	5	4	3	2	1	0
UHCKMSK	0	0	0	0	0	0	UHBUFMSK	UHMSK

UHBUFMSK	UHMSK	USB控制器功能选择
0	0	USB 主机控制器操作使能
1	0	只存数据 RAM (8KB) 操作使能 (使用 USB 功能)
1	1	使用 USB 主机控制器/只存数据 RAM (8KB) 禁止
其它		禁止设置

注意事项 请确保设置第7位至第2位为“0”。

20.6.2 USB功能控制器寄存器列表

(1) EPC 控制寄存器

(1/2)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
0020000H	UF0 EP0NAK 寄存器	UF0E0N	R/W		√		00H
00200002H	UF0 EP0NAKALL 寄存器	UF0E0NA	R/W		√		00H
00200004H	UF0 EPNAK 寄存器	UF0EN	R/W		√		00H
00200006H	UF0 EPNAK 屏蔽寄存器	UF0ENM	R/W		√		00H
00200008H	UF0 SNDSIE 寄存器	UF0SDS	R/W		√		00H
0020000AH	UF0 CLR 请求寄存器	UF0CLR	R		√		00H
0020000CH	UF0 SET 请求寄存器	UF0SET	R		√		00H
0020000EH	UF0 EP 状态 0 寄存器	UF0EPS0	R		√		00H
00200010H	UF0 EP 状态 1 寄存器	UF0EPS1	R		√		00H
00200012H	UF0 EP 状态 2 寄存器	UF0EPS2	R		√		00H
00200020H	UF0 INT 状态 0 寄存器	UF0IS0	R		√		00H
00200022H	UF0 INT 状态 1 寄存器	UF0IS1	R		√		00H
00200024H	UF0 INT 状态 2 寄存器	UF0IS2	R		√		00H
00200026H	UF0 INT 状态 3 寄存器	UF0IS3	R		√		00H
00200028H	UF0 INT 状态 4 寄存器	UF0IS4	R		√		00H
0020002EH	UF0 INT 屏蔽 0 寄存器	UF0IM0	R/W		√		00H
00200030H	UF0 INT 屏蔽 1 寄存器	UF0IM1	R/W		√		00H
00200032H	UF0 INT 屏蔽 2 寄存器	UF0IM2	R/W		√		00H
00200034H	UF0 INT 屏蔽 3 寄存器	UF0IM3	R/W		√		00H
00200036H	UF0 INT 屏蔽 4 寄存器	UF0IM4	R/W		√		00H
0020003CH	UF0 INT 清除 0 寄存器	UF0IC0	W		√		FFH
0020003EH	UF0 INT 清除 1 寄存器	UF0IC1	W		√		FFH
00200040H	UF0 INT 清除 2 寄存器	UF0IC2	W		√		FFH
00200042H	UF0 INT 清除 3 寄存器	UF0IC3	W		√		FFH
00200044H	UF0 INT 清除 4 寄存器	UF0IC4	W		√		FFH
0020004CH	UF0 INT & DMARQ 寄存器	UF0IDR	R/W		√		00H
0020004EH	UF0 DMA 状态 0 寄存器	UF0DMS0	R		√		00H
00200050H	UF0 DMA 状态 1 寄存器	UF0DMS1	R		√		00H
00200060H	UF0 FIFO 清除 0 寄存器	UF0FIC0	W		√		00H
00200062H	UF0 FIFO 清除 1 寄存器	UF0FIC1	W		√		00H
0020006AH	UF0 数据结束寄存器	UF0DEND	R/W		√		00H
0020006EH	UF0 GPR 寄存器	UF0GPR	W		√		00H
00200074H	UF0 模式控制寄存器	UF0MODC	R/W		√		00H
00200078H	UF0 模式状态寄存器	UF0MODS	R		√		00H

(2/2)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200080H	UF0 有效接口编号寄存器	UF0AIFN	R/W		√		00H
00200082H	UF0 有效复用设置寄存器	UF0AAS	R/W		√		00H
00200084H	UF0 复用设置状态寄存器	UF0ASS	R		√		00H
00200086H	UF0 端点 1 接口映射寄存器	UF0E1IM	R/W		√		00H
00200088H	UF0 端点 2 接口映射寄存器	UF0E2IM	R/W		√		00H
0020008AH	UF0 端点 3 接口映射寄存器	UF0E3IM	R/W		√		00H
0020008CH	UF0 端点 4 接口映射寄存器	UF0E4IM	R/W		√		00H
00200092H	UF0 端点 7 接口映射寄存器	UF0E7IM	R/W		√		00H

(2) EPC 数据保持寄存器

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200100 H	UF0 EP0 读寄存器	UF0E0R	R		√		不确定
00200102H	UF0 EP0 长度寄存器	UF0E0L	R		√		00H
00200104H	UF0 EP0 建立寄存器	UF0E0ST	R		√		00H
00200106H	UF0 EP0 写寄存器	UF0E0W	W		√		不确定
00200108H	UF0 批量-out 1 寄存器	UF0BO1	R		√		不确定
0020010AH	UF0 批量-out 1 长度寄存器	UF0BO1L	R		√		00H
0020010CH	UF0 批量-out 2 寄存器	UF0BO2	R		√		不确定
0020010EH	UF0 批量-out 2 长度寄存器	UF0BO2L	R		√		00H
00200110H	UF0 批量-in 1 寄存器	UF0BI1	W		√		不确定
00200112H	UF0 批量-in 2 寄存器	UF0BI2	W		√		不确定
00200114H	UF0 中断 1 寄存器	UF0INT1	W		√		不确定

(3) EPC 请求数据寄存器

(1/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200144H	UF0 设备状态寄存器 L	UF0DSTL	R/W		√		00H
00200146H	UF0 设备状态寄存器 H	UF0DSTH	R/W		√		00H
00200148H	UF0 接口状态寄存器 L	UF0ISTL	R/W		√		00H
0020014AH	UF0 接口状态寄存器 H	UF0ISTH	R/W		√		00H
0020014CH	UF0 EP0 状态寄存器 L	UF0E0SL	R/W		√		00H
0020014EH	UF0 EP0 状态寄存器 H	UF0E0SH	R/W		√		00H
00200150H	UF0 EP1 状态寄存器 L	UF0E1SL	R/W		√		00H
00200152H	UF0 EP1 状态寄存器 H	UF0E1SH	R/W		√		00H
00200154H	UF0 EP2 状态寄存器 L	UF0E2SL	R/W		√		00H
00200156H	UF0 EP2 状态寄存器 H	UF0E2SH	R/W		√		00H
00200158H	UF0 EP3 状态寄存器 L	UF0E3SL	R/W		√		00H
0020015AH	UF0 EP3 状态寄存器 H	UF0E3SH	R/W		√		00H
0020015CH	UF0 EP4 状态寄存器 L	UF0E4SL	R/W		√		00H
0020015EH	UF0 EP4 状态寄存器 H	UF0E4SH	R/W		√		00H
00200168H	UF0 EP7 状态寄存器 L	UF0E7SL	R/W		√		00H
0020016AH	UF0 EP7 状态寄存器 H	UF0E7SH	R/W		√		00H
00200180H	UF0 地址寄存器	UF0ADRS	R		√		00H
00200182H	UF0 配置寄存器	UF0CNF	R		√		00H
00200184H	UF0 接口 0 寄存器	UF0IF0	R		√		00H
00200186H	UF0 接口 1 寄存器	UF0IF1	R		√		00H
00200188H	UF0 接口 2 寄存器	UF0IF2	R		√		00H
0020018AH	UF0 接口 3 寄存器	UF0IF3	R		√		00H
0020018CH	UF0 接口 4 寄存器	UF0IF4	R		√		00H
002001A0H	UF0 描述符长度寄存器	UF0DSCL	R/W		√		00H
002001A2H	UF0 设备描述符寄存器 0	UF0DD0	R/W		√		不确定
002001A4H	UF0 设备描述符寄存器 1	UF0DD1	R/W		√		不确定
002001A6H	UF0 设备描述符寄存器 2	UF0DD2	R/W		√		不确定
002001A8H	UF0 设备描述符寄存器 3	UF0DD3	R/W		√		不确定
002001AAH	UF0 设备描述符寄存器 4	UF0DD4	R/W		√		不确定
002001ACH	UF0 设备描述符寄存器 5	UF0DD5	R/W		√		不确定
002001AEH	UF0 设备描述符寄存器 6	UF0DD6	R/W		√		不确定
002001B0H	UF0 设备描述符寄存器 7	UF0DD7	R/W		√		不确定
002001B2H	UF0 设备描述符寄存器 8	UF0DD8	R/W		√		不确定
002001B4H	UF0 设备描述符寄存器 9	UF0DD9	R/W		√		不确定
002001B6H	UF0 设备描述符寄存器 10	UF0DD10	R/W		√		不确定
002001B8H	UF0 设备描述符寄存器 11	UF0DD11	R/W		√		不确定
002001BAH	UF0 设备描述符寄存器 12	UF0DD12	R/W		√		不确定
002001BCH	UF0 设备描述符寄存器 13	UF0DD13	R/W		√		不确定
002001BEH	UF0 设备描述符寄存器 14	UF0DD14	R/W		√		不确定
002001C0H	UF0 设备描述符寄存器 15	UF0DD15	R/W		√		不确定
002001C2H	UF0 设备描述符寄存器 16	UF0DD16	R/W		√		不确定
002001C4H	UF0 设备描述符寄存器 17	UF0DD17	R/W		√		不确定

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
002001C6H	UF0 配置/接口/端点描述符寄存器 0	UF0CIE0	R/W		√		不确定
002001C8H	UF0 配置/接口/端点描述符寄存器 1	UF0CIE1	R/W		√		不确定
002001CAH	UF0 配置/接口/端点描述符寄存器 2	UF0CIE2	R/W		√		不确定
002001CCH	UF0 配置/接口/端点描述符寄存器 3	UF0CIE3	R/W		√		不确定
002001CEH	UF0 配置/接口/端点描述符寄存器 4	UF0CIE4	R/W		√		不确定
002001D0H	UF0 配置/接口/端点描述符寄存器 5	UF0CIE5	R/W		√		不确定
002001D2H	UF0 配置/接口/端点描述符寄存器 6	UF0CIE6	R/W		√		不确定
002001D4H	UF0 配置/接口/端点描述符寄存器 7	UF0CIE7	R/W		√		不确定
002001D6H	UF0 配置/接口/端点描述符寄存器 8	UF0CIE8	R/W		√		不确定
002001D8H	UF0 配置/接口/端点描述符寄存器 9	UF0CIE9	R/W		√		不确定
002001DAH	UF0 配置/接口/端点描述符寄存器 10	UF0CIE10	R/W		√		不确定
002001DCH	UF0 配置/接口/端点描述符寄存器 11	UF0CIE11	R/W		√		不确定
002001DEH	UF0 配置/接口/端点描述符寄存器 12	UF0CIE12	R/W		√		不确定
002001E0H	UF0 配置/接口/端点描述符寄存器 13	UF0CIE13	R/W		√		不确定
002001E2H	UF0 配置/接口/端点描述符寄存器 14	UF0CIE14	R/W		√		不确定
002001E4H	UF0 配置/接口/端点描述符寄存器 15	UF0CIE15	R/W		√		不确定
002001E6H	UF0 配置/接口/端点描述符寄存器 16	UF0CIE16	R/W		√		不确定
002001E8H	UF0 配置/接口/端点描述符寄存器 17	UF0CIE17	R/W		√		不确定
002001EAH	UF0 配置/接口/端点描述符寄存器 18	UF0CIE18	R/W		√		不确定
002001ECH	UF0 配置/接口/端点描述符寄存器 19	UF0CIE19	R/W		√		不确定
002001EEH	UF0 配置/接口/端点描述符寄存器 20	UF0CIE20	R/W		√		不确定
002001F0H	UF0 配置/接口/端点描述符寄存器 21	UF0CIE21	R/W		√		不确定

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
002001F2H	UF0 配置/接口/端点描述符 寄存器 22	UF0CIE22	R/W		√		不确定
002001F4H	UF0 配置/接口/端点描述符 寄存器 23	UF0CIE23	R/W		√		不确定
002001F6H	UF0 配置/接口/端点描述符 寄存器 24	UF0CIE24	R/W		√		不确定
002001F8H	UF0 配置/接口/端点描述符 寄存器 25	UF0CIE25	R/W		√		不确定
002001FAH	UF0 配置/接口/端点描述符 寄存器 26	UF0CIE26	R/W		√		不确定
002001FCH	UF0 配置/接口/端点描述符 寄存器 27	UF0CIE27	R/W		√		不确定
002001FEH	UF0 配置/接口/端点描述符 寄存器 28	UF0CIE28	R/W		√		不确定
00200200H	UF0 配置/接口/端点描述符 寄存器 29	UF0CIE29	R/W		√		不确定
00200202H	UF0 配置/接口/端点描述符 寄存器 30	UF0CIE30	R/W		√		不确定
00200204H	UF0 配置/接口/端点描述符 寄存器 31	UF0CIE31	R/W		√		不确定
00200206H	UF0 配置/接口/端点描述符 寄存器 32	UF0CIE32	R/W		√		不确定
00200208H	UF0 配置/接口/端点描述符 寄存器 33	UF0CIE33	R/W		√		不确定
0020020AH	UF0 配置/接口/端点描述符 寄存器 34	UF0CIE34	R/W		√		不确定
0020020CH	UF0 配置/接口/端点描述符 寄存器 35	UF0CIE35	R/W		√		不确定
0020020EH	UF0 配置/接口/端点描述符 寄存器 36	UF0CIE36	R/W		√		不确定
00200210H	UF0 配置/接口/端点描述符 寄存器 37	UF0CIE37	R/W		√		不确定
00200212H	UF0 配置/接口/端点描述符 寄存器 38	UF0CIE38	R/W		√		不确定
00200214H	UF0 配置/接口/端点描述符 寄存器 39	UF0CIE39	R/W		√		不确定
00200216H	UF0 配置/接口/端点描述符 寄存器 40	UF0CIE40	R/W		√		不确定
00200218H	UF0 配置/接口/端点描述符 寄存器 41	UF0CIE41	R/W		√		不确定
0020021AH	UF0 配置/接口/端点描述符 寄存器 42	UF0CIE42	R/W		√		不确定
0020021CH	UF0 配置/接口/端点描述符 寄存器 43	UF0CIE43	R/W		√		不确定

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
0020021EH	UF0 配置/接口/端点描述符 寄存器 44	UF0CIE44	R/W		√		不确定
00200220H	UF0 配置/接口/端点描述符 寄存器 45	UF0CIE45	R/W		√		不确定
00200222H	UF0 配置/接口/端点描述符 寄存器 46	UF0CIE46	R/W		√		不确定
00200224H	UF0 配置/接口/端点描述符 寄存器 47	UF0CIE47	R/W		√		不确定
00200226H	UF0 配置/接口/端点描述符 寄存器 48	UF0CIE48	R/W		√		不确定
00200228H	UF0 配置/接口/端点描述符 寄存器 49	UF0CIE49	R/W		√		不确定
0020022AH	UF0 配置/接口/端点描述符 寄存器 50	UF0CIE50	R/W		√		不确定
0020022CH	UF0 配置/接口/端点描述符 寄存器 51	UF0CIE51	R/W		√		不确定
0020022EH	UF0 配置/接口/端点描述符 寄存器 52	UF0CIE52	R/W		√		不确定
00200230H	UF0 配置/接口/端点描述符 寄存器 53	UF0CIE53	R/W		√		不确定
00200232H	UF0 配置/接口/端点描述符 寄存器 54	UF0CIE54	R/W		√		不确定
00200234H	UF0 配置/接口/端点描述符 寄存器 55	UF0CIE55	R/W		√		不确定
00200236H	UF0 配置/接口/端点描述符 寄存器 56	UF0CIE56	R/W		√		不确定
00200238H	UF0 配置/接口/端点描述符 寄存器 57	UF0CIE57	R/W		√		不确定
0020023AH	UF0 配置/接口/端点描述符 寄存器 58	UF0CIE58	R/W		√		不确定
0020023CH	UF0 配置/接口/端点描述符 寄存器 59	UF0CIE59	R/W		√		不确定
0020023EH	UF0 配置/接口/端点描述符 寄存器 60	UF0CIE60	R/W		√		不确定
00200240H	UF0 配置/接口/端点描述符 寄存器 61	UF0CIE61	R/W		√		不确定
00200242H	UF0 配置/接口/端点描述符 寄存器 62	UF0CIE62	R/W		√		不确定
00200244H	UF0 配置/接口/端点描述符 寄存器 63	UF0CIE63	R/W		√		不确定
00200246H	UF0 配置/接口/端点描述符 寄存器 64	UF0CIE64	R/W		√		不确定
00200248H	UF0 配置/接口/端点描述符 寄存器 65	UF0CIE65	R/W		√		不确定

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
0020024AH	UF0 配置/接口/端点描述符 寄存器 66	UF0CIE66	R/W		√		不确定
0020024CH	UF0 配置/接口/端点描述符 寄存器 67	UF0CIE67	R/W		√		不确定
0020024EH	UF0 配置/接口/端点描述符 寄存器 68	UF0CIE68	R/W		√		不确定
00200250H	UF0 配置/接口/端点描述符 寄存器 69	UF0CIE69	R/W		√		不确定
00200252H	UF0 配置/接口/端点描述符 寄存器 70	UF0CIE70	R/W		√		不确定
00200254H	UF0 配置/接口/端点描述符 寄存器 71	UF0CIE71	R/W		√		不确定
00200256H	UF0 配置/接口/端点描述符 寄存器 72	UF0CIE72	R/W		√		不确定
00200258H	UF0 配置/接口/端点描述符 寄存器 73	UF0CIE73	R/W		√		不确定
0020025AH	UF0 配置/接口/端点描述符 寄存器 74	UF0CIE74	R/W		√		不确定
0020025CH	UF0 配置/接口/端点描述符 寄存器 75	UF0CIE75	R/W		√		不确定
0020025EH	UF0 配置/接口/端点描述符 寄存器 76	UF0CIE76	R/W		√		不确定
00200260H	UF0 配置/接口/端点描述符 寄存器 77	UF0CIE77	R/W		√		不确定
00200262H	UF0 配置/接口/端点描述符 寄存器 78	UF0CIE78	R/W		√		不确定
00200264H	UF0 配置/接口/端点描述符 寄存器 79	UF0CIE79	R/W		√		不确定
00200266H	UF0 配置/接口/端点描述符 寄存器 80	UF0CIE80	R/W		√		不确定
00200268H	UF0 配置/接口/端点描述符 寄存器 81	UF0CIE81	R/W		√		不确定
0020026AH	UF0 配置/接口/端点描述符 寄存器 82	UF0CIE82	R/W		√		不确定
0020026CH	UF0 配置/接口/端点描述符 寄存器 83	UF0CIE83	R/W		√		不确定
0020026EH	UF0 配置/接口/端点描述符 寄存器 84	UF0CIE84	R/W		√		不确定
00200270H	UF0 配置/接口/端点描述符 寄存器 85	UF0CIE85	R/W		√		不确定
00200272H	UF0 配置/接口/端点描述符 寄存器 86	UF0CIE86	R/W		√		不确定
00200274H	UF0 配置/接口/端点描述符 寄存器 87	UF0CIE87	R/W		√		不确定

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200276H	UF0 配置/接口/端点描述符 寄存器 88	UF0CIE88	R/W		√		不确定
00200278H	UF0 配置/接口/端点描述符 寄存器 89	UF0CIE89	R/W		√		不确定
0020027AH	UF0 配置/接口/端点描述符 寄存器 90	UF0CIE90	R/W		√		不确定
0020027CH	UF0 配置/接口/端点描述符 寄存器 91	UF0CIE91	R/W		√		不确定
0020027EH	UF0 配置/接口/端点描述符 寄存器 92	UF0CIE92	R/W		√		不确定
00200280H	UF0 配置/接口/端点描述符 寄存器 93	UF0CIE93	R/W		√		不确定
00200282H	UF0 配置/接口/端点描述符 寄存器 94	UF0CIE94	R/W		√		不确定
00200284H	UF0 配置/接口/端点描述符 寄存器 95	UF0CIE95	R/W		√		不确定
00200286H	UF0 配置/接口/端点描述符 寄存器 96	UF0CIE96	R/W		√		不确定
00200288H	UF0 配置/接口/端点描述符 寄存器 97	UF0CIE97	R/W		√		不确定
0020028AH	UF0 配置/接口/端点描述符 寄存器 98	UF0CIE98	R/W		√		不确定
0020028CH	UF0 配置/接口/端点描述符 寄存器 99	UF0CIE99	R/W		√		不确定
0020028EH	UF0 配置/接口/端点描述符 寄存器 100	UF0CIE100	R/W		√		不确定
00200290H	UF0 配置/接口/端点描述符 寄存器 101	UF0CIE101	R/W		√		不确定
00200292H	UF0 配置/接口/端点描述符 寄存器 102	UF0CIE102	R/W		√		不确定
00200294H	UF0 配置/接口/端点描述符 寄存器 103	UF0CIE103	R/W		√		不确定
00200296H	UF0 配置/接口/端点描述符 寄存器 104	UF0CIE104	R/W		√		不确定
00200298H	UF0 配置/接口/端点描述符 寄存器 105	UF0CIE105	R/W		√		不确定
0020029AH	UF0 配置/接口/端点描述符 寄存器 106	UF0CIE106	R/W		√		不确定
0020029CH	UF0 配置/接口/端点描述符 寄存器 107	UF0CIE107	R/W		√		不确定
0020029EH	UF0 配置/接口/端点描述符 寄存器 108	UF0CIE108	R/W		√		不确定
002002A0H	UF0 配置/接口/端点描述符 寄存器 109	UF0CIE109	R/W		√		不确定

(7/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
002002A2H	UF0 配置/接口/端点描述符 寄存器 110	UF0CIE110	R/W		√		不确定
002002A4H	UF0 配置/接口/端点描述符 寄存器 111	UF0CIE111	R/W		√		不确定
002002A6H	UF0 配置/接口/端点描述符 寄存器 112	UF0CIE112	R/W		√		不确定
002002A8H	UF0 配置/接口/端点描述符 寄存器 113	UF0CIE113	R/W		√		不确定
002002AAH	UF0 配置/接口/端点描述符 寄存器 114	UF0CIE114	R/W		√		不确定
002002ACH	UF0 配置/接口/端点描述符 寄存器 115	UF0CIE115	R/W		√		不确定
002002AEH	UF0 配置/接口/端点描述符 寄存器 116	UF0CIE116	R/W		√		不确定
002002A0H	UF0 配置/接口/端点描述符 寄存器 117	UF0CIE117	R/W		√		不确定
002002A2H	UF0 配置/接口/端点描述符 寄存器 118	UF0CIE118	R/W		√		不确定
002002A4H	UF0 配置/接口/端点描述符 寄存器 119	UF0CIE119	R/W		√		不确定
002002A6H	UF0 配置/接口/端点描述符 寄存器 120	UF0CIE120	R/W		√		不确定
002002B8H	UF0 配置/接口/端点描述符 寄存器 121	UF0CIE121	R/W		√		不确定
002002BAH	UF0 配置/接口/端点描述符 寄存器 122	UF0CIE122	R/W		√		不确定
002002BCH	UF0 配置/接口/端点描述符 寄存器 123	UF0CIE123	R/W		√		不确定
002002BEH	UF0 配置/接口/端点描述符 寄存器 124	UF0CIE124	R/W		√		不确定
002002C0H	UF0 配置/接口/端点描述符 寄存器 125	UF0CIE125	R/W		√		不确定
002002C2H	UF0 配置/接口/端点描述符 寄存器 126	UF0CIE126	R/W		√		不确定
002002C4H	UF0 配置/接口/端点描述符 寄存器 127	UF0CIE127	R/W		√		不确定
002002C6H	UF0 配置/接口/端点描述符 寄存器 128	UF0CIE128	R/W		√		不确定
002002C8H	UF0 配置/接口/端点描述符 寄存器 129	UF0CIE129	R/W		√		不确定
002002CAH	UF0 配置/接口/端点描述符 寄存器 130	UF0CIE130	R/W		√		不确定
002002CCH	UF0 配置/接口/端点描述符 寄存器 131	UF0CIE131	R/W		√		不确定

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
002002CEH	UF0 配置/接口/端点描述符 寄存器 132	UF0CIE132	R/W		√		不确定
002002D0H	UF0 配置/接口/端点描述符 寄存器 133	UF0CIE133	R/W		√		不确定
002002D2H	UF0 配置/接口/端点描述符 寄存器 134	UF0CIE134	R/W		√		不确定
002002D4H	UF0 配置/接口/端点描述符 寄存器 135	UF0CIE135	R/W		√		不确定
002002D6H	UF0 配置/接口/端点描述符 寄存器 136	UF0CIE136	R/W		√		不确定
002002D8H	UF0 配置/接口/端点描述符 寄存器 137	UF0CIE137	R/W		√		不确定
002002DAH	UF0 配置/接口/端点描述符 寄存器 138	UF0CIE138	R/W		√		不确定
002002DCH	UF0 配置/接口/端点描述符 寄存器 139	UF0CIE139	R/W		√		不确定
002002DEH	UF0 配置/接口/端点描述符 寄存器 140	UF0CIE140	R/W		√		不确定
002002E0H	UF0 配置/接口/端点描述符 寄存器 141	UF0CIE141	R/W		√		不确定
002002E2H	UF0 配置/接口/端点描述符 寄存器 142	UF0CIE142	R/W		√		不确定
002002E4H	UF0 配置/接口/端点描述符 寄存器 143	UF0CIE143	R/W		√		不确定
002002E6H	UF0 配置/接口/端点描述符 寄存器 144	UF0CIE144	R/W		√		不确定
002002E8H	UF0 配置/接口/端点描述符 寄存器 145	UF0CIE145	R/W		√		不确定
002002EAH	UF0 配置/接口/端点描述符 寄存器 146	UF0CIE146	R/W		√		不确定
002002ECH	UF0 配置/接口/端点描述符 寄存器 147	UF0CIE147	R/W		√		不确定
002002EEH	UF0 配置/接口/端点描述符 寄存器 148	UF0CIE148	R/W		√		不确定
002002F0H	UF0 配置/接口/端点描述符 寄存器 149	UF0CIE149	R/W		√		不确定
002002F2H	UF0 配置/接口/端点描述符 寄存器 150	UF0CIE150	R/W		√		不确定
002002F4H	UF0 配置/接口/端点描述符 寄存器 151	UF0CIE151	R/W		√		不确定
002002F6H	UF0 配置/接口/端点描述符 寄存器 152	UF0CIE152	R/W		√		不确定
002002F8H	UF0 配置/接口/端点描述符 寄存器 153	UF0CIE153	R/W		√		不确定

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
002002FAH	UF0 配置/接口/端点描述符 寄存器 154	UF0CIE154	R/W		√		不确定
002002FCH	UF0 配置/接口/端点描述符 寄存器 155	UF0CIE155	R/W		√		不确定
002002FEH	UF0 配置/接口/端点描述符 寄存器 156	UF0CIE156	R/W		√		不确定
00200300H	UF0 配置/接口/端点描述符 寄存器 157	UF0CIE157	R/W		√		不确定
00200302H	UF0 配置/接口/端点描述符 寄存器 158	UF0CIE158	R/W		√		不确定
00200304H	UF0 配置/接口/端点描述符 寄存器 159	UF0CIE159	R/W		√		不确定
00200306H	UF0 配置/接口/端点描述符 寄存器 160	UF0CIE160	R/W		√		不确定
00200308H	UF0 配置/接口/端点描述符 寄存器 161	UF0CIE161	R/W		√		不确定
0020030AH	UF0 配置/接口/端点描述符 寄存器 162	UF0CIE162	R/W		√		不确定
0020030CH	UF0 配置/接口/端点描述符 寄存器 163	UF0CIE163	R/W		√		不确定
0020030EH	UF0 配置/接口/端点描述符 寄存器 164	UF0CIE164	R/W		√		不确定
00200310H	UF0 配置/接口/端点描述符 寄存器 165	UF0CIE165	R/W		√		不确定
00200312H	UF0 配置/接口/端点描述符 寄存器 166	UF0CIE166	R/W		√		不确定
00200314H	UF0 配置/接口/端点描述符 寄存器 167	UF0CIE167	R/W		√		不确定
00200316H	UF0 配置/接口/端点描述符 寄存器 168	UF0CIE168	R/W		√		不确定
00200318H	UF0 配置/接口/端点描述符 寄存器 169	UF0CIE169	R/W		√		不确定
0020031AH	UF0 配置/接口/端点描述符 寄存器 170	UF0CIE170	R/W		√		不确定
0020031CH	UF0 配置/接口/端点描述符 寄存器 171	UF0CIE171	R/W		√		不确定
0020031EH	UF0 配置/接口/端点描述符 寄存器 172	UF0CIE172	R/W		√		不确定
00200320H	UF0 配置/接口/端点描述符 寄存器 173	UF0CIE173	R/W		√		不确定
00200322H	UF0 配置/接口/端点描述符 寄存器 174	UF0CIE174	R/W		√		不确定
00200324H	UF0 配置/接口/端点描述符 寄存器 175	UF0CIE175	R/W		√		不确定

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200326H	UF0 配置/接口/端点描述符 寄存器 176	UF0CIE176	R/W		√		不确定
00200328H	UF0 配置/接口/端点描述符 寄存器 177	UF0CIE177	R/W		√		不确定
0020032AH	UF0 配置/接口/端点描述符 寄存器 178	UF0CIE178	R/W		√		不确定
0020032CH	UF0 配置/接口/端点描述符 寄存器 179	UF0CIE179	R/W		√		不确定
0020032EH	UF0 配置/接口/端点描述符 寄存器 180	UF0CIE180	R/W		√		不确定
00200330H	UF0 配置/接口/端点描述符 寄存器 181	UF0CIE181	R/W		√		不确定
00200332H	UF0 配置/接口/端点描述符 寄存器 182	UF0CIE182	R/W		√		不确定
00200334H	UF0 配置/接口/端点描述符 寄存器 183	UF0CIE183	R/W		√		不确定
00200336H	UF0 配置/接口/端点描述符 寄存器 184	UF0CIE184	R/W		√		不确定
00200338H	UF0 配置/接口/端点描述符 寄存器 185	UF0CIE185	R/W		√		不确定
0020033AH	UF0 配置/接口/端点描述符 寄存器 186	UF0CIE186	R/W		√		不确定
0020033CH	UF0 配置/接口/端点描述符 寄存器 187	UF0CIE187	R/W		√		不确定
0020033EH	UF0 配置/接口/端点描述符 寄存器 188	UF0CIE188	R/W		√		不确定
00200340H	UF0 配置/接口/端点描述符 寄存器 189	UF0CIE189	R/W		√		不确定
00200342H	UF0 配置/接口/端点描述符 寄存器 190	UF0CIE190	R/W		√		不确定
00200344H	UF0 配置/接口/端点描述符 寄存器 191	UF0CIE191	R/W		√		不确定
00200346H	UF0 配置/接口/端点描述符 寄存器 192	UF0CIE192	R/W		√		不确定
00200348H	UF0 配置/接口/端点描述符 寄存器 193	UF0CIE193	R/W		√		不确定
0020034AH	UF0 配置/接口/端点描述符 寄存器 194	UF0CIE194	R/W		√		不确定
0020034CH	UF0 配置/接口/端点描述符 寄存器 195	UF0CIE195	R/W		√		不确定
0020034EH	UF0 配置/接口/端点描述符 寄存器 196	UF0CIE196	R/W		√		不确定

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200350H	UF0 配置/接口/端点描述符 寄存器 197	UF0CIE197	R/W		√		不确定
00200352H	UF0 配置/接口/端点描述符 寄存器 198	UF0CIE198	R/W		√		不确定
00200354H	UF0 配置/接口/端点描述符 寄存器 199	UF0CIE199	R/W		√		不确定
00200356H	UF0 配置/接口/端点描述符 寄存器 200	UF0CIE200	R/W		√		不确定
00200358H	UF0 配置/接口/端点描述符 寄存器 201	UF0CIE201	R/W		√		不确定
0020035AH	UF0 配置/接口/端点描述符 寄存器 202	UF0CIE202	R/W		√		不确定
0020035CH	UF0 配置/接口/端点描述符 寄存器 203	UF0CIE203	R/W		√		不确定
0020035EH	UF0 配置/接口/端点描述符 寄存器 204	UF0CIE204	R/W		√		不确定
00200360H	UF0 配置/接口/端点描述符 寄存器 205	UF0CIE205	R/W		√		不确定
00200362H	UF0 配置/接口/端点描述符 寄存器 206	UF0CIE206	R/W		√		不确定
00200364H	UF0 配置/接口/端点描述符 寄存器 207	UF0CIE207	R/W		√		不确定
00200366H	UF0 配置/接口/端点描述符 寄存器 208	UF0CIE208	R/W		√		不确定
00200368H	UF0 配置/接口/端点描述符 寄存器 209	UF0CIE209	R/W		√		不确定
0020036AH	UF0 配置/接口/端点描述符 寄存器 210	UF0CIE210	R/W		√		不确定
0020036CH	UF0 配置/接口/端点描述符 寄存器 211	UF0CIE211	R/W		√		不确定
0020036EH	UF0 配置/接口/端点描述符 寄存器 212	UF0CIE212	R/W		√		不确定
00200370H	UF0 配置/接口/端点描述符 寄存器 213	UF0CIE213	R/W		√		不确定
00200372H	UF0 配置/接口/端点描述符 寄存器 214	UF0CIE214	R/W		√		不确定
00200374H	UF0 配置/接口/端点描述符 寄存器 215	UF0CIE215	R/W		√		不确定
00200376H	UF0 配置/接口/端点描述符 寄存器 216	UF0CIE216	R/W		√		不确定
00200378H	UF0 配置/接口/端点描述符 寄存器 217	UF0CIE217	R/W		√		不确定

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
0020037AH	UF0 配置/接口/端点描述符 寄存器 218	UF0CIE218	R/W		√		不确定
0020037CH	UF0 配置/接口/端点描述符 寄存器 219	UF0CIE219	R/W		√		不确定
0020037EH	UF0 配置/接口/端点描述符 寄存器 220	UF0CIE220	R/W		√		不确定
00200380H	UF0 配置/接口/端点描述符 寄存器 221	UF0CIE221	R/W		√		不确定
00200382H	UF0 配置/接口/端点描述符 寄存器 222	UF0CIE222	R/W		√		不确定
00200384H	UF0 配置/接口/端点描述符 寄存器 223	UF0CIE223	R/W		√		不确定
00200386H	UF0 配置/接口/端点描述符 寄存器 224	UF0CIE224	R/W		√		不确定
00200388H	UF0 配置/接口/端点描述符 寄存器 225	UF0CIE225	R/W		√		不确定
0020038AH	UF0 配置/接口/端点描述符 寄存器 226	UF0CIE226	R/W		√		不确定
0020038CH	UF0 配置/接口/端点描述符 寄存器 227	UF0CIE227	R/W		√		不确定
0020038EH	UF0 配置/接口/端点描述符 寄存器 228	UF0CIE228	R/W		√		不确定
00200390H	UF0 配置/接口/端点描述符 寄存器 229	UF0CIE229	R/W		√		不确定
00200392H	UF0 配置/接口/端点描述符 寄存器 230	UF0CIE230	R/W		√		不确定
00200394H	UF0 配置/接口/端点描述符 寄存器 231	UF0CIE231	R/W		√		不确定
00200396H	UF0 配置/接口/端点描述符 寄存器 232	UF0CIE232	R/W		√		不确定
00200398H	UF0 配置/接口/端点描述符 寄存器 233	UF0CIE233	R/W		√		不确定
0020039AH	UF0 配置/接口/端点描述符 寄存器 234	UF0CIE234	R/W		√		不确定
0020039CH	UF0 配置/接口/端点描述符 寄存器 235	UF0CIE235	R/W		√		不确定
0020039EH	UF0 配置/接口/端点描述符 寄存器 236	UF0CIE236	R/W		√		不确定
002003A0H	UF0 配置/接口/端点描述符 寄存器 237	UF0CIE237	R/W		√		不确定
002003A2H	UF0 配置/接口/端点描述符 寄存器 238	UF0CIE238	R/W		√		不确定

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
002003A4H	UF0 配置/接口/端点描述符 寄存器 239	UF0CIE239	R/W		√		不确定
002003A6H	UF0 配置/接口/端点描述符 寄存器 240	UF0CIE240	R/W		√		不确定
002003A8H	UF0 配置/接口/端点描述符 寄存器 241	UF0CIE241	R/W		√		不确定
002003AAH	UF0 配置/接口/端点描述符 寄存器 242	UF0CIE242	R/W		√		不确定
002003ACH	UF0 配置/接口/端点描述符 寄存器 243	UF0CIE243	R/W		√		不确定
002003AEH	UF0 配置/接口/端点描述符 寄存器 244	UF0CIE244	R/W		√		不确定
002003B0H	UF0 配置/接口/端点描述符 寄存器 245	UF0CIE245	R/W		√		不确定
002003B2H	UF0 配置/接口/端点描述符 寄存器 246	UF0CIE246	R/W		√		不确定
002003B4H	UF0 配置/接口/端点描述符 寄存器 247	UF0CIE247	R/W		√		不确定
002003B6H	UF0 配置/接口/端点描述符 寄存器 248	UF0CIE248	R/W		√		不确定
002003B8H	UF0 配置/接口/端点描述符 寄存器 249	UF0CIE249	R/W		√		不确定
002003BAH	UF0 配置/接口/端点描述符 寄存器 250	UF0CIE250	R/W		√		不确定
002003BCH	UF0 配置/接口/端点描述符 寄存器 251	UF0CIE251	R/W		√		不确定
002003BEH	UF0 配置/接口/端点描述符 寄存器 252	UF0CIE252	R/W		√		不确定
002003C0H	UF0 配置/接口/端点描述符 寄存器 253	UF0CIE253	R/W		√		不确定
002003C2H	UF0 配置/接口/端点描述符 寄存器 254	UF0CIE254	R/W		√		不确定
002003C4H	UF0 配置/接口/端点描述符 寄存器 255	UF0CIE255	R/W		√		不确定

(4) 桥寄存器

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200400H	桥中断控制寄存器	BRGINTT	R/W			√	不确定
00200402H	桥中断使能寄存器	BRGINTE	R/W			√	不确定
00200404H	EPC 模块控制寄存器	EPCCLT	R/W			√	不确定
00200408H	CPU I/F 总线控制寄存器	CPUBCTL	R/W			√	不确定

(5) DMA 寄存器

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200500H	EP1 DMA 控制寄存器 1	UF0E1DC1	R/W			√	0000H
00200502H	EP1 DMA 控制寄存器 2	UF0E1DC2	R/W			√	0000H
00200504H	EP2 DMA 控制寄存器 1	UF0E2DC1	R/W			√	0000H
00200506H	EP2 DMA 控制寄存器 2	UF0E2DC2	R/W			√	0000H
00200508H	EP3 DMA 控制寄存器 1	UF0E3DC1	R/W			√	0000H
0020050AH	EP3 DMA 控制寄存器 2	UF0E3DC2	R/W			√	0000H
0020050CH	EP4 DMA 控制寄存器 1	UF0E4DC1	R/W			√	0000H
0020050EH	EP4 DMA 控制寄存器 2	UF0E4DC2	R/W			√	0000H

(6) 批量-in 寄存器

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00201000H	UF0 EP1 批量-in 传输数据寄存器	UF0EP1BI	R/W			√	0000H
00202000H	UF0 EP3 批量-in 传输数据寄存器	UF0EP3BI	R/W			√	0000H

(7) 批量-out 寄存器

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00210000H	UF0 EP2 批量-out 传输数据寄存器	UF0EP2BO	R			√	0000H
00220000H	UF0 EP4 批量-out 传输数据寄存器	UF0EP4BO	R			√	0000H

(8) 外设控制寄存器

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00240000H	USBF DMA 请求使能寄存器	UFDRQEN	R/W		√	√	0000H

20.6.3 EPC控制寄存器

(1) UF0 EP0NAK 寄存器 (UF0E0N)

该寄存器用于控制端点 0 的 NAK (自动执行的请求除外)。

该寄存器可以按字节进行读取或写入 (但是, 第 0 位只能被读取)。

在 UF0FIC0 和 UF0FIC1 寄存器被置位后, 需要五个 USB 时钟才能在该寄存器上反映出状态。因此, 如果需要正确读取状态, 访问 UF0FIC0 和 UF0FIC1 寄存器的写信号与访问 UF0EPS0、UF0EPS1、UF0EPS2、UF0E0N 和 UF0EN 寄存器的读信号要进行分离, 至少间隔四个 USB 时钟。

当 NAK 被发送到端点 0 读, 端点 2 和端点 4 时, 对 EP0NKR 位的写入访问被忽略。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0N	0	0	0	0	0	0	EP0NKR	EP0NKW	00200000H	00H

位的位置	位名称	功能
1	EP0NKR	该位用于控制到端点 0 的 OUT 令牌的 NAK (自动执行的请求除外)。当端点 0 正确接收到数据时, 它被硬件自动设置为 1。当 UF0E0R 寄存器的数据被 FW 读取时, 它也被硬件清除为 0 (计数器值 = 0)。 <ul style="list-style-type: none"> 1: 发送 NAK。 0: 不发送 NAK (默认值)。 由于某些原因即使 USBF 准备好接收数据, 但不应该从 USB 总线接收数据时, 通过 FW 置该位为 1。在这种情况下, 在该位被 FW 清除为 0 之前 USBF 持续发送 NAK。UF0E0R 寄存器被清除, 则该位也立即被清除为 0。
0	EP0NKW	该位表示如何控制到端点 0 的 IN 令牌的 NAK (自动执行的请求除外)。当端点 0 的数据被发送并且主机正确接收到发送的数据时, 该位由硬件自动清除为 0。在该位被清除之前, UF0E0W 寄存器的数据被保留。因此, 即使在主机不能正确接收数据而产生重发请求情况下, 也不需要重新写入该位。要发送一个短包, 确保设置 UF0DEND 寄存器的 E0DED 位为 1。当 FIFO 满时, 该位被自动设置为 1。UF0DEND 寄存器的 E0DED 位被置为 1, 则 EP0NKW 位被同时立即自动设置为 1。 <ul style="list-style-type: none"> 1: 不发送 NAK。 0: 发送 NAK (默认值)。 在数据阶段不能正确接收 ACK 时, 如果控制传输进入状态阶段, 只要 UF0E0W 寄存器被清除, 则该位立即被清除为 0。当 UF0E0W 由 FW 清除时, 该位也被清除为 0。

下面，使用 IN / OUT 令牌的 SETUP 事务的过程解释如下。

(a) 使用 IN 令牌时（硬件自动执行的请求除外）

在接收 CPUDEC 中断之后，且在从 UF0E0ST 寄存器读取数据之前，应该使用 FW 来清除 UF0IS1 寄存器的 PROT 位为 0。然后，根据请求来执行处理，并且如果需要由 IN 令牌返回数据，向 UF0E0W 寄存器写入数据。在写入完成之后，确认 UF0IS1 寄存器的 PROT 位是 0，并且设置 UF0DEND 寄存器的 E0DED 位为 1。在 EP0NKR 位被置为 1 之后的第一个 IN 令牌时硬件发送数据。如果 UF0IS1 寄存器的 PROT 位为 1，表明在控制传输完成之前已经再次发生 SETUP 事务。在这种情况下，通过清除 UF0IC1 寄存器的 PROTC 位为 0 来清除 UF0IS1 寄存器的 PROT 位为 0，然后再次从 UF0E0ST 寄存器读取数据。随后接收到的请求可以被读取。

(b) 使用 OUT 令牌时（硬件自动执行的请求除外）

在接收 CPUDEC 中断之后，且在从 UF0E0ST 寄存器读取数据之前，应该使用 FW 来清除 UF0IS1 寄存器的 PROT 位。在从 UF0E0R 寄存器读取数据之前，确认 UF0IS1 寄存器的 PROT 位是 0。如果 PROT 是 1，意味着无效数据被保留。通过 FW（EP0NKR 位自动清除为 0）清除 FIFO。如果 UF0IS1 寄存器的 PROT 位为 0，读取 UF0E0L 寄存器的数据长度，并按照设定数量从 UF0E0R 寄存器读取足够的数量。当从 UF0E0R 寄存器读取数据完成时（当 UF0E0R 寄存器的计数器被清除为 0 时），硬件自动清除 EP0NKR 位为 0。

(2) UF0 EP0NAKALL 寄存器 (UF0E0NA)

该寄存器用于控制到端点 0 的所有请求的 NAK。对于自动执行的请求，它也同样有效。

该寄存器可以按字节进行读取或写入。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0NA	0	0	0	0	0	0	0	EP0NKA	00200002H	00H

位的位置	位名称	功能
0	EP0NKA	<p>该位控制到端点 0 的 SETUP 事务以外的 NAK (包含自动执行的请求)。该位由 FW 操作。</p> <p>1: 发送 NAK。 0: 不发送 NAK (默认值)。</p> <p>当用于自动执行的请求的数据被更改时，该寄存器用于防止 FW 的写入访问与从 SIE 读取访问之间的冲突。当执行从 SIE 的访问时，推迟从 FW 的写入访问到该位的反映。在从 FW 重新写入请求数据寄存器之前，确认该位被正确置 1。</p> <p>只有在以下情况，该位置 1 会被反映。</p> <ul style="list-style-type: none"> • USBF 刚复位后，并且从没有接收过 SETUP 令牌 • 总线复位刚接收后，并且从没有接收过 SETUP 令牌 • 检测到 SETUP 令牌的 PID • 阶段被更改为状态阶段 <p>清除该位为 0 会立即被反映，除非正在接收 IN 令牌并且正在发生 NAK 响应。</p> <p>端点 0 传输期间，上述四种情况下，设置 EP0NKA 位为 1 会被反映，但是当端点 0 没有传输数据时，写入该位后会立即被反映。</p>

(3) UF0 EPNAK 寄存器 (UF0EN)

该寄存器用于控制端点 0 以外其它端点的 NAK。

该寄存器可以按字节进行读取或写入（但是，第 5 位，第 4 位，第 1 位和第 0 位只能被读取）。

只有当 UF0ENM 寄存器的 BKO2NKM 位为 1 时，BKO2NK 位才可以被写入。只有当 UF0ENM 寄存器的 BKO1NKM 位为 1 时，BKO1NK 位才可以被写入。

如果 UF0EnIM 寄存器 (n = 1 至 4, 7) 的设置与接口的当前设置不支持各个端点，相关的位无效。

在 UF0FIC0 和 UF0FIC1 寄存器被置位后，需要五个 USB 时钟才能在该寄存器上反映出状态。因此，如果需要正确读取状态，访问 UF0FIC0 和 UF0FIC1 寄存器的写信号与访问 UF0EPS0、UF0EPS1、UF0EPS2、UF0E0N 和 UF0EN 寄存器的读信号要进行分离，至少间隔四个 USB 时钟。

当 NAK 被发送到端点 0 读，端点 2 和端点 4 时，对 BKO1NK 和 BKO2NK 位的写入访问被忽略。

请确保清除第 7 位和第 6 位为“0”。如果这些位被置为 1，操作无法保证。

(1/4)

	7	6	5	4	3	2	1	0	地址	复位后
UF0EN	0	0	0	IT1NK	BKO2NK	BKO1NK	BKI2NK	BKI1NK	00200004H	00H

位的位置	位名称	功能
4	IT1NK	该位用于控制到端点 7 的 NAK（中断 1 传输）。 写入数据到 UF0INT1 寄存器导致该寄存器满，该位被自动置 1 且启动传输。发送一个不会占满 FIFO 的短包，设置 UF0DEND 寄存器的 IT1DEND 位为 1。IT1DEND 位被置为 1，则该位立即被自动置为 1。 1: 不发送 NAK。 0: 发送 NAK（默认值）。 只要 UF0INT1 寄存器被清除，该位也被清除为 0。

位的位置	位名称	功能
3	BKO2NK	<p>该位用于控制到端点 4 的 NAK (批量 2 传输 (OUT))。</p> <p>1: 发送 NAK。 0: 不发送 NAK (默认值)。</p> <p>只有当连接到 UF0BO2 寄存器的 SIE 端的 FIFO (bank 配置的 64 字节 FIFO) 不能接收数据时, 该位才能被设置为 1。当执行一个切换开关操作时, 它被清除为 0。当满足下列条件时, bank 被更改 (切换操作)。</p> <ul style="list-style-type: none"> • 正确接收的数据存储在连接到 SIE 端的 FIFO 中。 • 连接到 CPU 端的 FIFO 计数器的值为 0 (读取完成)。 <p>当接收到 BKO2DT 中断请求时, 应该使用 FW 来读取 UF0BO2L 寄存器的数据长度, 并按照设定数量从 UF0BO2 寄存器读取足够的数。如果由于某些原因不从 USB 总线接收数据, 即使 USBF 已经准备好接收数据, 通过 FW 设置该位为 1。在这种情况下, USBF 持续发送 NAK 直到 FW 清除该位为 0。只要清除 UF0BO2 寄存器, 该位也立即被清除为 0。</p>
2	BKO1NK	<p>该位用于控制到端点 2 的 NAK (批量 1 传输 (OUT))。</p> <p>1: 发送 NAK。 0: 不发送 NAK (默认值)。</p> <p>只有当连接到 UF0BO1 寄存器的 SIE 端的 FIFO (bank 配置的 64 字节 FIFO) 不能接收数据时, 该位才能被设置为 1。当执行一个切换开关操作时, 它被清除为 0。当满足下列条件时, bank 被更改 (切换操作)。</p> <ul style="list-style-type: none"> • 正确接收的数据存储在连接到 SIE 端的 FIFO 中。 • 连接到 CPU 端的 FIFO 计数器的值为 0 (读取完成)。 <p>当接收到 BKO1DT 中断请求时, 应该使用 FW 来读取 UF0BO1L 寄存器的数据长度, 并按照设定数量从 UF0BO1 寄存器读取足够的数。如果由于某些原因不从 USB 总线接收数据, 即使 USBF 已经准备好接收数据, 通过 FW 设置该位为 1。在这种情况下, USBF 持续发送 NAK 直到 FW 清除该位为 0。只要清除 UF0BO1 寄存器, 该位也立即被清除为 0。</p>

- 注意事项**
1. 如果在 PIO 模式下从 UF0BO2 读取数据时, DMA 使能, 立即发出一个 DMA 请求。
 2. 如果在 DMA 传输模式下读取 CPU 侧 FIFO 中的最后一个数据, DMA 请求信号变为无效。
 3. 如果在 DMA 传输模式下接收到 TC 信号, DMA 请求信号变为无效。

位的位置	位名称	功能
1	BKI2NK	<p>该位用于控制到端点 3 的 NAK (批量 2 传输 (IN))。</p> <p>1: 不发送 NAK。 0: 发送 NAK (默认值)。</p> <p>只有当连接到 UF0BI2 寄存器 SIE 端的 FIFO (bank 配置的 64 字节 FIFO) 不能接收数据时, 该位才能被清除为 0。当执行一个切换开关操作 (保留 UF0BI2 寄存器的数据, 直到传输已经正确完成) 时, 它被置为 1。当满足下列条件时, bank 被更改 (切换操作)。</p> <ul style="list-style-type: none"> • 数据被正确写入到连接在 CPU 总线端的 FIFO 中 (写入已经完成, 并且 FIFO 为满或者 UF0DEND 寄存器被置位)。 • 连接在 SIE 端的 FIFO 计数器的值为 0。 <p>写入数据到 FIFO 导致 CPU 端的 FIFO 满并执行 FIFO 切换操作时, 该位被自动置 1 且启动数据传输。但是, 如果在 UF0DEND 寄存器的 BKI2T 位被清除为 0 时, 因为 DMA 写入数据到 CPU 端的 FIFO 导致 FIFO 满, 不执行切换操作, 因为不满足切换操作的条件, 直到 UF0DEND 寄存器的 BKI2DED 位被置为 1。要发送一个不会占满 CPU 端 FIFO 的短包, 在写入数据完成后设置 BKI2DED 位为 1。当 BKI2DED 位被置为 1 的同时, 执行切换开关操作, 则该位被自动置为 1。只要清除 UF0BI2 寄存器, 该位也立即被清除为 0。</p>

- 注意事项**
1. 如果在 PIO 模式下向 UF0BI2 寄存器写入数据时, DMA 使能, 立即发出一个 DMA 请求。
 2. 如果在 DMA 传输模式下写入 64 字节数据, DMA 请求信号变为无效。如果然后 BKI2NK 位被置 1, 数据和 IN 令牌同步发送。在 FIFO 切换时, 只要 DMA 请求没有被屏蔽, DMA 请求信号再次变为有效。如果 BKI2NK 位没有置 1, 即使接收到 IN 令牌, 数据也不会被发送。这种情况下, 设置 UF0DEND 寄存器的 BKI2DED 位为 1。
 3. 如果在 DMA 传输模式下接收到 TC 信号, DMA 请求信号变为无效。同时 DMA 请求被屏蔽。如果 BKI2NK 位没有置 1, 即使接收到 IN 令牌, 数据也不会被发送。当 UF0DEND 寄存器的 BKI2DED 位被 FW 置 1 时, 数据和 IN 令牌同步发送。为了再次执行 DMA 传输, 取消对 DMA 请求的屏蔽。

位的位置	位名称	功能
0	BKI1NK	<p>该位用于控制到端点 1 的 NAK (批量 1 传输 (IN))。</p> <p>1: 不发送 NAK。 0: 发送 NAK (默认值)。</p> <p>只有当连接到 UF0B11 寄存器 SIE 端的 FIFO (bank 配置的 64 字节 FIFO) 不能接收数据时, 该位才能被清除为 0。当执行一个切换开关操作 (保留 UF0B11 寄存器的数据, 直到传输已经正确完成) 时, 它被置为 1。当满足下列条件时, bank 被更改 (切换操作)。</p> <ul style="list-style-type: none"> • 数据被正确写入到连接在 CPU 总线端的 FIFO 中 (写入已经完成, 并且 FIFO 为满或者 UF0DEND 寄存器被置位)。 • 连接在 SIE 端的 FIFO 计数器的值为 0。 <p>写入数据到 FIFO 导致 CPU 端的 FIFO 满并执行 FIFO 切换操作时, 该位被自动置 1 且启动数据传输。但是, 如果在 UF0DEND 寄存器的 BKI1T 位被清除为 0 时, 因为 DMA 写入数据到 CPU 端的 FIFO 导致 FIFO 满, 不执行切换操作, 因为不满足切换操作的条件, 直到 UF0DEND 寄存器的 BKI1DED 位被置为 1。要发送一个不会占满 CPU 端 FIFO 的短包, 在写入数据完成后设置 BKI1DED 位为 1。当 BKI1DED 位被置为 1 的同时, 执行切换开关操作, 则该位被自动置为 1。只要清除 UF0B11 寄存器, 该位也立即被清除为 0。</p>

- 注意事项**
1. 如果在 PIO 模式下向 UF0B11 寄存器写入数据时, DMA 使能, 立即发出一个 DMA 请求。
 2. 如果在 DMA 传输模式下写入 64 字节数据, DMA 请求信号变为无效。如果然后 BKI1NK 位被置 1, 数据和 IN 令牌同步发送。在 FIFO 切换时, 只要 DMA 请求没有被屏蔽, DMA 请求信号再次变为有效。如果 BKI1NK 位没有置 1, 即使接收到 IN 令牌, 数据也不会被发送。这种情况下, 设置 UF0DEND 寄存器的 BKI1DED 位为 1。
 3. 如果在 DMA 传输模式下接收到 TC 信号, DMA 请求信号变为无效。同时 DMA 请求被屏蔽。如果 BKI1NK 位没有置 1, 即使接收到 IN 令牌, 数据也不会被发送。当 UF0DEND 寄存器的 BKI1DED 位被 FW 置 1 时, 数据和 IN 令牌同步发送。为了再次执行 DMA 传输, 取消对 DMA 请求的屏蔽。

(4) UF0 EPNK 屏蔽寄存器 (UF0ENM)

该寄存器用于控制对 UF0EN 寄存器写入访问的屏蔽。

该寄存器可以按字节进行读取或写入。

请确保第 7 位至第 4 位、第 1 位和第 0 位清除为“0”。如果该位被设置为 1，操作无法保证。

	7	6	5	4	3	2	1	0	地址	复位后
UF0ENM	0	0	0	0	BKO2NKM	BKO1NKM	0	0	00200006H	00H

位的位置	位名称	功能
3	BKO2NKM	该位用于指定是否屏蔽对 UF0EN 寄存器的第 3 位 (BKO2NK) 的写入访问。 1: 不屏蔽。 0: 屏蔽 (默认值)。
2	BKO1NKM	该位用于指定是否屏蔽对 UF0EN 寄存器的第 2 位 (BKO1NK) 的写入访问。 1: 不屏蔽。 0: 屏蔽 (默认值)。

(5) UF0 SNDSIE 寄存器 (UF0SDS)

该寄存器执行无握手之类的操作。它可以直接操作 SIE 的引脚。

该寄存器可以按字节进行读取或写入。

确保清除第 2 位。如果该位被设置为 1，操作不被保证。

	7	6	5	4	3	2	1	0	地址	复位后
UF0SDS	0	0	0	0	SNDSTL	0	0	RSUMIN	00200008H	00H

位的位置	位名称	功能
3	SNDSTL	<p>该位使端点 0 发布一个 STALL 握手。如果系统不支持 CPUDEC 处理的请求，导致一个 STALL 握手响应，设置该位为 1。如果 SET_CONFIGURATION 或 SET_INTERFACE 发送一个不支持的 wValue，硬件设置该位为 1。如果由于一个自动执行的请求溢出，端点 0 发生问题，该位也被设置为 1。但是，UF0E0SL 寄存器的 E0HALT 位不被置为 1。</p> <p>1: 用 STALL 握手的响应。 0: 不用 STALL 握手的响应（默认值）。</p> <p>当接收到下一个 SETUP 令牌时，该位被清除为 0，并且总线的握手响应不是 STALL。要通过 FW 将 SNDSTL 位置为 1，不要向 UF0E0W 寄存器写入数据。根据设置该位的时序，STALL 响应不会及时产生，并且可能在 NAK 响应后的下一个传输中产生。当该位被置为 1 时，只有当 FW 执行请求正在执行时，对该位的设置才有效。当接收到下一个 SETUP 令牌时，它被自动清除为 0。</p> <p>备注 SNDSTL 位只对 FW 执行的请求有效。</p>
0	RSUMIN	<p>该位输出恢复 (Resume) 信号到 USB 总线上。除非 UF0DSTL 寄存器的 RMWK 位被置为 1，否则写入该位无效。</p> <p>1: 产生恢复信号 0: 不产生恢复信号（默认值）</p> <p>当该位被设置为 1 时，继续产生恢复信号。在经过指定时间后通过 FW 清除该位为 0。因为信号在内部按照时钟采样，只有在提供 CLK 时，操作才能保证。当系统的 CLK 停止时，必须小心。</p>

(6) UF0 CLR 请求寄存器 (UF0CLR)

该寄存器表明接收到 CLEAR_FEATURE 请求的目标。

该寄存器是只读的，按字节进行读取。

只有产生中断请求时，该寄存器才有意义。在状态阶段完成后，每一位被设置为 1，并且当这个寄存器被读取时，自动清除为 0。

如果 UF0EnIM 寄存器 (n = 1 至 4, 7) 的设置与接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0CLR	CLREP8	CLREP7	CLREP4	CLREP3	CLREP2	CLREP1	CLREP0	CLRDEV	0020000AH	00H

位的位置	位名称	功能
7 至 1	CLREPN	这些位表明接收到 CLEAR_FEATURE 端点 n 请求并且被自动处理。 1: 已自动处理 0: 未自动处理 (默认值)
0	CLRDEV	该位表明接收到 CLEAR_FEATURE Device 请求并被自动处理。 1: 已自动处理 0: 未自动处理 (默认值)

备注 n = 0 至 4, 7

(7) UF0 SET 请求寄存器 (UF0SET)

该寄存器表明自动处理的 SET_XXXX (SET_INTERFACE 除外) 请求的目标。

该寄存器是只读的, 按字节进行读取。

只有产生中断请求时, 该寄存器才有意义。在状态阶段完成后, 每一位被设置为 1, 并且当这个寄存器被读取时, 自动清除为 0。

	7	6	5	4	3	2	1	0	地址	复位后
UF0SET	SETCON	0	0	0	0	SETEP	0	SETDEV	0020000CH	00H

位的位置	位名称	功能
7	SETCON	该位表明接收到 SET_CONFIGURATION 请求并被自动处理。 1: 已自动处理 0: 未自动处理 (默认值)
2	SETEP	该位表明接收到 SET_FEATURE 端点 n 请求 (n = 0 至 4, 7) 并被自动处理。 1: 已自动处理 0: 未自动处理 (默认值)
0	SETDEV	该位表明接收到 SET_FEATURE Device 并被自动处理。 1: 已自动处理 0: 未自动处理 (默认值)

(8) UF0 EP 状态 0 寄存器 (UF0EPS0)

该寄存器表明 USB 总线状态以及寄存器数据的存在或缺席。

该寄存器是只读的，按字节进行读取。

如果 UF0EnIM 寄存器 (n = 1 至 4, 7) 的设置与接口的当前设置不支持各个端点，相关的位无效。

在 UF0FIC0 和 UF0FIC1 寄存器被置位后，需要五个 USB 时钟才能在该寄存器上反映出状态。因此，如果需要正确读取状态，访问 UF0FIC0 和 UF0FIC1 寄存器的写信号与访问 UF0EPS0、UF0EPS1、UF0EPS2、UF0E0N 和 UF0EN 寄存器的读信号要进行分离，至少间隔四个 USB 时钟。

(1/2)

	7	6	5	4	3	2	1	0	地址	复位后
UF0EPS0	IT2	IT1	BKOUT2	BKOUT1	BKIN2	BKIN1	EP0W	EP0R	0020000EH	00H

位的位置	位名称	功能
7, 6	ITn	这些位表示数据是在 UF0INTn 寄存器 (FIFO) 中。通过设置 UF0DEND 寄存器的 ITnDED 位为 1，即使没有数据写入到该寄存器 (Null 数据传输)，也可以创建数据存于 UF0INTn 寄存器的状态。即使 UF0INTn 寄存器的计数器为 0，只要设置 UF0DEND 寄存器的 ITnDED 位为 1，这些位就被硬件置 1。在正确传输之后，这些位被清除为 0。 1: 数据在该寄存器中。 0: 没有数据在该寄存器中 (默认值)。
5, 4	BKOUTn	这些位表明数据是在连接到 CPU 端的 UF0BOn 寄存器 (FIFO) 中。当 FIFO 配置的 UF0BOn 寄存器被切换时，这些位被硬件自动设置为 1。当读取连接到 CPU 端的 UF0BOn 寄存器 (FIFO) 完成时 (计数器值 = 0)，这些位被硬件自动清除为 0。当接收到空数据时 (切换 FIFO 也未发生)，它不被置为 1。 1: 数据在该寄存器中。 0: 没有数据在该寄存器中 (默认值)。
3, 2	BKINn	该位表明数据是在连接到 CPU 端的 UF0Bin 寄存器 (FIFO) 中。通过设置 UF0DEND 寄存器的 BKInDED 位为 1，即使没有数据写入到该寄存器 (Null 数据发送)，也可以创建数据存于 UF0Bin 寄存器的状态。当 UF0Bin 寄存器的计数器为 0 时，只要 UF0DEND 寄存器的 BKInDED 位被置为 1，这些位就被硬件置 1。当执行切换操作时，这些位被清除为 0。 1: 数据在该寄存器中。 0: 没有数据在该寄存器中 (默认值)。

备注 n = 1, 2

位的位置	位名称	功能
1	EP0W	<p>该位表明数据是在 UF0E0W 寄存器 (FIFO) 中。通过设置 UF0DEND 寄存器的 E0DED 位为 1，即使没有数据写入到该寄存器 (Null 数据发送)，也可以创建数据存于 UF0E0W 寄存器的状态。当 UF0E0W 寄存器的计数器为 0 时，只要 UF0DEND 寄存器的 E0DED 位被置为 1，该位就由硬件设置为 1。在正确传输之后，该位被清除为 0。</p> <p>1: 数据在寄存器中。 0: 没有数据在寄存器中 (默认值)。</p>
0	EP0R	<p>该位表明数据是在 UF0E0R 寄存器 (FIFO) 中。当读取 UF0E0R 寄存器 (FIFO) 完成后，它被硬件自动清除为 0。如果接收到 Null 数据，它被置为 1。</p> <p>1: 数据在寄存器中。 0: 没有数据在寄存器中 (默认值)。</p>

(9) UF0 EP 状态 1 寄存器 (UF0EPS1)

该寄存器表明 USB 总线状态以及寄存器数据的存在或缺席。

该寄存器是只读的，按字节进行读取。

		7	6	5	4	3	2	1	0	地址	复位后
UF0EPS1		RSUM	0	0	0	0	0	0	0	00200010H	00H

位的位置	位名称	功能
7	RSUM	该位表明 USB 总线处于恢复 (Resume) 状态。只有在产生中断请求时，该位才有意义。 1: 挂起状态 0: 恢复状态 (默认值) 因为信号在内部按照时钟采样，只有提供 CLK 时，操作才能保证。当系统的 CLK 停止时，必须小心。即使 CLK 被停止，SIE 的 INTRSUM 信号也会工作。因此，通过让中断请求标志寄存器 1L (IF1L) 的 RSUMIF 位变为有效，或者降低 USBF 的 CLK 的频率，可以支持该位。 当该位被读取时，自动清除为 0。

(10) UF0 EP 状态 2 寄存器 (UF0EPS2)

该寄存器表明 USB 总线状态以及寄存器数据的存在或缺席。

该寄存器是只读的，按字节进行读取。

如果 UF0EnIM 寄存器 (n = 1 至 4, 7) 的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0EPS2	0	HALT8	HALT7	HALT4	HALT3	HALT2	HALT1	HALT0	00200012H	00H

位的位置	位名称	功能
6 至 0	HALTn	<p>这些位表明端点 n 当前处于停顿 (stall) 中。当一个停顿条件满足时，例如发生溢出和接收到未定义的请求，这些位被置为 1。这些位由硬件自动置为 1。</p> <p>1: 端点处于停顿中。 0: 端点没有在停顿中 (默认值)。</p> <p>由于发生溢出或接收到未定义的请求，只要 HALT0 位被置为 1，SNDSTL 位立即被置为 1。如果在这个状态下接收到下一个 SETUP 令牌，SNDSTL 位被清除为 0，因此 HALT0 位也被清除为 0。如果端点 0 因为 SET_FEATURE 端点 0 请求而停顿，该位不会被清除为 0，直到接收到 CLEAR_FEATURE 端点 0 请求或暂停特征 (Halt Feature) 被 FW 清除。如果接收到 GET_STATUS 端点 0、CLEAR_FEATURE 端点 0 或 SET_FEATURE 端点 0 请求，或者如果因为 CPUDEC 中断请求而接收到要由 FW 处理的请求，HALT0 位被屏蔽并清除为 0，直到接收到下一个 SETUP 令牌。</p> <p>HALTn 位不被清除为 0，直到端点 n 接收到 CLEAR_FEATURE 端点请求，暂停特征由 SET_INTERFACE 清除，或由端点连接接口的 SET_CONFIGURATION 请求清除，或者暂停特征由 FW 清除。当 SET_INTERFACE 或 SET_CONFIGURATION 请求被正确处理，即使 wValue 与当前设定值相同，除端点 0 外的所有目标端点的暂停特征都会在请求处理后被清除，并且这些位也会被清除为 0。如果端点 0 因为 STALL 响应而被置为暂停特征，它不能被清除，因为 STALL 响应是在响应 SET_INTERFACE 和 SET_CONFIGURATION 请求时产生的。</p>

备注 n = 0 至 4, 7, 8

(11) UF0 INT 状态 0 寄存器 (UF0IS0)

该寄存器表明中断源。如果该寄存器的内容被更改，INT0B 信号变为有效。

该寄存器是只读的，按字节进行读取。

如果由 USBF 产生一个中断请求 (INTUSBF0)，FW 必须读取该寄存器来识别中断源。

向 UF0IC0 寄存器的对应位写入 0 时，该寄存器的各个位被强制清除为 0。

注意事项 在 USBF 中，多个中断源，比如总线复位，恢复，以及 Short，都是内部或操作并作为单个中断请求 (INTUSBF0) 发出。于是，在发生多个中断源的情况下，对它们进行或操作并将结果作为 INTUSBF0 中断请求来发出。

比如，如果发生一个总线复位中断源和一个恢复中断源，这两个源进行或操作并发出 INTUSBF0 中断请求。

在这些条件下，如果总线复位中断源被清除为 0 (UF0IC0.BUSRSTC = 0)，V850ES/JG3-U 或 V850ES/JH3-U 内部 INTUSBF0 中断请求可能保持置 1，因为恢复中断源仍然保持。新的中断请求标志 (US0BIC.US0BIF) 可能不被置 1。

在这种情况下，使用 INTUSBF0 中断服务程序执行对各个中断请求的清除处理之后，再次确认 UF0IS0 和 UF0IS1 寄存器的标志状态，如果还有任何中断源标志置 1，执行标志清除（只有适用的位需要清除（不要执行批量清除））。

(1/2)

	7	6	5	4	3	2	1	0	地址	复位后
UF0IS0	BUSRST	RSUSPD	0	SHORT	DMAED	SETRQ	CLRRQ	EPHALT	00200020H	00H

位的位置	位名称	功能
7	BUSRST	该位表明发生总线复位。 1: 已发生总线复位 (中断请求已产生) 0: 非总线复位状态 (默认值)
6	RSUSPD	该位表明发生恢复或挂起状态。通过 FW 参考 UF0EPS1 寄存器的第 7 位。 1: 已发生恢复或挂起状态 (中断请求已产生) 0: 未发生恢复或挂起状态 (默认值)
4	SHORT	该位表明数据读自 UF0BO1 或 UF0BO2 寄存器的 FIFO，且 USBSPnB 信号 (n = 2, 4) 被有效。只有在 DMA 模式下 FIFO 满，它才有效。 1: USBSPnB 信号被有效 (中断请求已产生) 0: USBSPnB 信号未有效 (默认值) 使用 UF0DMS1 寄存器来识别操作在哪个端点上执行。即使 UF0DMS1 寄存器被 FW 读取，该位也不会被自动清除为 0。

位的位置	位名称	功能
3	DMAED	<p>该位表明对端点 $n(n = 1$ 至 $4, 7)$ 的 DMA 结束 (TC) 信号被有效。</p> <p>1: 对端点 n 的 DMA 结束信号已经输入 (中断请求已产生)</p> <p>0: 对端点 n 的 DMA 结束信号没有输入 (默认值)</p> <p>当该位被置 1 时, 对端点 n 的 DMA 请求信号被有效。对端点 n 的 DMA 请求信号没有被有效, 除非 FW 使能 DMA 传输。</p> <p>使用 UF0DMS0 寄存器来确认在哪个端点上实际执行操作。但是, 即使 UF0DMS0 寄存器被 FW 读取, 该位也不会被自动清除为 0。</p>
2	SETRQ	<p>该位表明已经接收到将要被自动处理的 SET_XXXX 请求, 并且被自动处理 (XXXX = CONFIGURATION 或 FEATURE)。</p> <p>1: 已经接收到将要被自动处理的 SET_XXXX 请求 (中断请求已产生)</p> <p>0: 没有接收到将要被自动处理的 SET_XXXX 请求 (默认值)</p> <p>状态阶段完成后, 该位被置为 1。参考 UF0SET 寄存器来识别请求的目标。即使 UF0SET 寄存器被 FW 读取, 该位也不会被自动清除为 0。</p> <p>当已经接收到 SET_FEATURE 端点请求时, EPHALT 位也被置为 1。</p>
1	CLRRQ	<p>该位表明已经接收到 CLEAR_FEATURE 请求并且被自动处理。</p> <p>1: 已经接收到 CLEAR_FEATURE 请求 (中断请求已产生)</p> <p>0: 没有接收到 CLEAR_FEATURE 请求 (默认值)</p> <p>状态阶段完成后, 该位被置为 1。参考 UF0CLR 寄存器来识别请求的目标。即使 UF0CLR 寄存器被 FW 读取, 该位也不会被自动清除为 0。</p>
0	EPHALT	<p>该位表明一个端点已经停顿。</p> <p>1: 端点已经停顿 (中断请求已产生)</p> <p>0: 端点没有停顿 (默认值)</p> <p>当通过设置 FW 停顿一个端点时, 该位也被设置为 1。</p> <p>通过参考 UF0EPS2 寄存器来识别哪一个端点已经停顿。即使接收到 CLEAR_FEATURE 端点、SET_INTERFACE 或 SET_CONFIGURATION 请求时, 该位也不会被自动清除为 0。如果在端点 0 溢出的情况下接收到下一个 SETUP 令牌, 它也不被自动清除为 0。</p> <p>注意事项 即使端点 0 的暂停特征被设置并且产生这个中断请求, 在接收到 SET_FEATURE 端点 0、CLEAR_FEATURE 端点 0 或 GET_STATUS 端点 0 请求或 FW 处理的请求, 和接收到上述以外的其它 SETUP 令牌之间, UF0EPS2 寄存器的第 0 位被屏蔽并且被清除为 0。</p>

(12) UF0 INT 状态 1 寄存器 (UF0IS1)

该寄存器表明中断源。如果该寄存器的内容被更改，INT0B 信号被有效。

该寄存器是只读的，按字节进行读取。

如果由 USBF 产生一个中断请求 (INTUSBF0)，FW 必须读取该寄存器来识别中断源。

向 UF0IC0 寄存器的对应位写入 0 时，该寄存器的各个位被强制清除为 0。然而，当接收到下一个 SETUP 令牌时，UF0IS1 寄存器的 SUCES 和 STG 位被自动清除为 0。

注意事项 在 USBF 中，多个中断源，比如总线复位，恢复，以及 Short，都是内部或操作并作为单个中断请求 (INTUSBF0) 发出。于是，在发生多个中断源的情况下，对它们进行或操作并将结果作为 INTUSBF0 中断请求来发出。

比如，如果发生一个总线复位中断源和一个恢复中断源，这两个源进行或操作并发出 INTUSBF0 中断请求。

在这些条件下，如果总线复位中断源被清除为 0 (UF0IC0.BUSRSTC = 0)，V850ES/JG3-U 或 V850ES/JH3-U 内部 INTUSBF0 中断请求可能保持置 1，因为恢复中断源仍然保持。新的中断请求标志 (US0BIC.US0BIF) 可能不被置 1。

在这种情况下，使用 INTUSBF0 中断服务程序执行对各个中断请求的清除处理之后，再次确认 UF0IS0 和 UF0IS1 寄存器的标志状态，如果还有任何中断源标志置 1，执行标志清除（只有适用的位需要清除（不要执行批量清除））。

(1/2)

	7	6	5	4	3	2	1	0	地址	复位后
UF0IS1	0	E0IN	E0INDT	E0ODT	SUCES	STG	PROT	CPUDEC	00200022H	00H

位的位置	位名称	功能
6	E0IN	该位表明接收到对端点 0 的 IN 令牌，并且硬件已经自动发送 NAK。 1: 接收到 IN 令牌并且 NAK 已发送（中断请求已产生）。 0: 没有接收到 IN 令牌（默认值）。
5	E0INDT	该位表明数据被正确从 UF0E0W 寄存器发送。 1: 从 UF0E0W 寄存器的发送已经完成（中断请求已产生）。 0: 从 UF0E0W 寄存器发送尚未结束（默认值）。 紧邻在设置 UF0E0N 寄存器的 EPONKW 位为 1 后，数据和 IN 令牌同步发送。当主机正确接收到这个数据时，该位由硬件自动设置为 1。即使数据是 Null 包，它也会被设置为 1。当执行对 UF0E0W 寄存器的第一个写入访问时，该位被硬件自动清除为 0。

位的位置	位名称	功能
4	E0ODT	<p>该位表明数据被正确接收到 UF0E0R 寄存器中。</p> <p>1: 数据是在 UF0E0R 寄存器中 (中断请求已产生)。</p> <p>0: 数据不在 UF0E0R 寄存器中 (默认值)。</p> <p>当正确接收到数据时, 该位被硬件自动设置为 1。同时, UF0EPS0 寄存器的 EPOR 位也被置为 1。如果接收到一个 Null 包, 该位不被置 1。当 FW 读取 UF0E0R 寄存器时, 它被硬件自动清除为 0, 并且 UF0E0L 寄存器的值变为 0。</p>
3	SUCES	<p>该位表明已经接收到一个 FW 处理的请求或硬件处理的请求, 并且已经正确完成状态阶段。</p> <p>1: 控制传输已经被正确处理 (中断请求已产生)。</p> <p>0: 控制传输尚未被正确处理 (默认值)。</p> <p>状态阶段结束时, 该位被置为 1。接收到下一个 SETUP 令牌时, 它被硬件自动清除为 0。</p> <p>在控制传输的状态阶段接收到带有 0 (空数据) 的 Data PID 的数据时, 该位也被设置为 1。</p>
2	STG	<p>当控制传输的阶段进入状态阶段时, 该位被设置为 1。它对 FW 处理的请求和硬件处理的请求都有效。当控制传输 (无数据) 的阶段进入状态阶段时, 该位也被设置为 1。</p> <p>1: 状态阶段 (中断请求已产生)</p> <p>0: 非状态阶段 (默认值)</p> <p>当接收到下一个 SETUP 令牌时, 该位被硬件自动清除为 0。</p> <p>在数据阶段中未能正确接收到 ACK, 当控制传输的阶段进入状态阶段, 它也会被设置为 1。在这种情况下, 如果 FW 正在处理控制传输 (读), 只要 UF0E0W 寄存器被清除, UF0E0N 寄存器的 EP0NKW 位就立即被清除为 0。</p>
1	PROT	<p>该位表明接收到一个 SETUP 令牌。它对 FW 处理的请求和硬件处理的请求都有效。</p> <p>1: 正确接收到 SETUP 令牌 (中断请求已产生)。</p> <p>0: 没有接收到 SETUP 令牌 (默认值)。</p> <p>当数据被正确接收到 UF0E0ST 寄存器中时, 该位被置为 1。当执行对 UF0E0ST 寄存器的第一次读取访问时, 通过 FW 清除该位为 0。如果它没有被 FW 清除为 0, 下一个 SETUP 令牌的接收不能被正确识别。</p> <p>该位用来准确识别控制传输期间被再次执行的 SETUP 事务。如果在控制传输期间 SETUP 事务被重新执行, 并且如果第二个请求由硬件执行, 则 CPUDEC 位不被置为 1, 但是 PROT 位可以用于对重新执行的识别。</p>
0	CPUDEC	<p>该位表明 UF0E0ST 寄存器有一个请求需要 FW 进行译码。</p> <p>1: 数据是在 UF0E0ST 寄存器中 (中断请求已产生)。</p> <p>0: 数据不在 UF0E0ST 寄存器中 (默认值)。</p> <p>当 UF0E0ST 寄存器的所有数据被读取时, 该位由硬件自动清除为 0。</p>

(13) UF0 INT 状态 2 寄存器 (UF0IS2)

该寄存器表明中断源。如果该寄存器的内容被更改，INT1B 信号被有效。

该寄存器是只读的，按字节进行读取。

如果由 USBF 产生一个中断请求 (INTUSBF0)，FW 必须读取该寄存器来识别中断源。

向 UF0IC2 寄存器的对应位写入 0 时，该寄存器的各个位被强制清除为 0。

如果 UF0EnIM (n = 1, 3, 7) 寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IS2	BKI2IN	BKI2DT	BKI1IN	BKI1DT	0	0	IT2DT	IT1DT	00200024H	00H

位的位置	位名称	功能
7, 5	BKInIN	这些位表明 IN 令牌已经接收到 UF0BIn 寄存器 (端点 m) 中，并且已经返回 NAK。 1: 接收到 IN 令牌，并且 NAK 已发送 (中断请求已产生) 0: 没有接收到 IN 令牌 (默认值)
6, 4	BKInDT	这些位表明 UF0B1 寄存器 (端点 1) 的 FIFO 已经被切换。这意味着可以向端点 m 写入数据。 1: FIFO 已经被切换 (中断请求已产生) 0: FIFO 尚未被切换 (默认值) 紧邻在设置 UF0EN 寄存器的 BKInNK 位为 1 后，写入端点 m 的数据和 IN 令牌同步发送。当 FIFO 被切换后，可以从 CPU 写入数据，该位由硬件自动设置为 1。即使数据是 Null 包，当 FIFO 被切换时，该位也被设置为 1。当执行对 UF0B1 寄存器的第一个写入访问时，该位由硬件自动清除为 0。
1, 0	ITnDT	这些位表明数据已经从 UF0INTn 寄存器 (端点 x) 中正确完成接收。 1: 传输已经完成 (中断请求已产生) 0: 传输尚未完成 (默认值) 紧邻在设置 UF0EN 寄存器的 ITnNK 位为 1 后，数据和 IN 令牌同步发送。当主机已经正确接收到该数据后，该位由硬件自动设置为 1。当执行对 UF0INTn 寄存器的第一个写入访问时，该位由硬件自动清除为 0。即使数据是 Null 包，该位也被设置为 1。

备注 n = 1, 2
 当 n = 1 时，m = 1 且 x = 7
 当 n = 2 时，m = 3

(14) UF0 INT 状态 3 寄存器 (UF0IS3)

该寄存器表明中断源。如果该寄存器的内容被更改，INT1B 信号被有效。

该寄存器是只读的，按字节进行读取。

如果由 USBF 产生一个中断请求 (INTUSBF0)，FW 必须读取该寄存器来识别中断源。

向 UF0IC3 寄存器的对应位写入 0 时，该寄存器的各个位被强制清除为 0。

如果 UF0EnIM (n = 2, 4) 寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

(1/2)

	7	6	5	4	3	2	1	0	地址	复位后
UF0IS3	BKO2FL	BKO2NL	BKO2 NAK	BKO2DT	BKO1FL	BKO1NL	BKO1	BKO1DT	00200026H	00H

位的位置	位名称	功能
7, 3	BKOnFL	这些位表明数据已经被正确接收到 UF0BOn 寄存器 (端点 m) 中，并且 CPU 和 SIE 的 FIFO 都保持该数据。 1: 接收的数据是在 UF0BOn 寄存器的两个 FIFO 中 (中断请求已产生) 0: 接收的数据不在 UF0BO1 寄存器的 SIE 端的 FIFO 中 (默认值) 如果数据被保持在 CPU 和 SIE 的 FIFO 中，该位由硬件自动设置为 1。当 FIFO 被切换时，该位由硬件自动清除为 0。
6, 2	BKOnNL	这些位表明一个 Null 包 (长度为 0 的包) 被接收到 UF0BOn 寄存器 (端点 m) 中。 1: Null 包已被接收 (中断请求已产生) 0: Null 包未被接收 (默认值) 当 FIFO 为空时，Null 包接收后，这些位立即被设置为 1。如果数据在 CPU 端的 FIFO 中，当完成对这个 FIFO 的读取时，该位被设置为 1。
5, 1	BKOnNAK	这些位表明 OUT 令牌已经被接收到 UF0BOn 寄存器 (端点 m) 中，并且已经返回 NAK。 1: 接收到 OUT 令牌并且 NAK 已发送 (中断请求已产生)。 0: 没有接收到 OUT 令牌 (默认值)。

备注 n = 1, 2
 当 n = 1 时, m = 2
 当 n = 2 时, m = 4

位的位置	位名称	功能
4, 0	BKOnDT	<p>这些位表明在 UF0BOn 寄存器（端点 m）中数据被正确接收。</p> <p>1: 接收已经正确完成（中断请求已产生）</p> <p>0: 接收尚未完成（默认值）</p> <p>当数据被正确接收并且 FIFO 被切换时，这些位被硬件自动设置为 1。同时，UF0EPS0 寄存器的对应位也被置为 1。如果数据是 Null 包，这些位不被置为 1。当由于 UF0BOn 寄存器被 FW 读取，导致 UF0BOnL 寄存器的值变为 0 时，该位被硬件自动清除为 0。</p> <p>当 CPU 端 FIFO 的所有内容已经被读取时，这些位被自动清除为 0。但是，如果此时数据是在 SIE 端的 FIFO 中，中断请求不会被清除，并且 INTUSBF1 信号不变为无效。如果数据是连续接收的，信号保持有效。</p>

备注 n = 1, 2
 当 n = 1 时, m = 2
 当 n = 2 时, m = 4

(15) UF0 INT 状态 4 寄存器 (UF0IS4)

该寄存器表明中断源。如果该寄存器的内容被更改，INT2B 信号变为有效。

该寄存器是只读的，按字节进行读取。

如果由 USBF 产生一个中断请求 (INTUSBF0)，FW 必须读取该寄存器来识别中断源。

向 UF0IC4 寄存器的对应位写入 0 时，该寄存器的各个位被强制清除为 0。

如果 UF0EnIM (n = 1 至 4, 7) 寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IS4	0	0	SETINT	0	0	0	0	0	00200028H	00H

位的位置	位名称	功能
5	SETINT	该位表明已经接收到 SET_INTERFACE 请求，并且被自动处理。 1: 请求已经被自动处理 (中断请求已产生)。 0: 请求尚未被自动处理 (默认值)。 可以通过读取 UF0ASS 或 UF0IFn 寄存器 (n = 0 至 4) 来识别该位的当前设置。

(16) UF0 INT 屏蔽 0 寄存器 (UF0IM0)

该寄存器用于控制中断源的屏蔽，中断源由 UF0ISO 寄存器表示。

该寄存器可以按字节进行读取或写入。

向该寄存器的对应位写入 1，FW 可以屏蔽来自 USBF 的中断请求 (INTUSBF0) 的发生。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IM0	BUS RSTM	RSU SPDM	0	SHORTM	DMA EDM	SET RQM	CLR RQM	EP HALTM	0020002EH	00H

位的位置	位名称	功能
7	BUSRSTM	该位屏蔽总线复位中断。 1: 屏蔽 0: 不屏蔽 (默认值)
6	RSUSPDM	该位屏蔽恢复/挂起中断。 1: 屏蔽 0: 不屏蔽 (默认值)
4	SHORTM	该位屏蔽 Short 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
3	DMAEDM	该位屏蔽 DMA_END 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
2	SETRQM	该位屏蔽 SET_RQ 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
1	CLRRQM	该位屏蔽 CLR_RQ 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
0	EPHALTM	该位屏蔽 EP_Halt 中断。 1: 屏蔽 0: 不屏蔽 (默认值)

(17) UF0 INT 屏蔽 1 寄存器 (UF0IM1)

该寄存器控制中断源的屏蔽, 中断源由 UF0IS1 寄存器表示。

该寄存器可以按字节进行读取或写入。

向该寄存器的对应位写入 1, FW 可以屏蔽来自 USBF 的中断请求 (INTUSB0B) 的发生。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IM1	0	E0INM	E0 INDTM	E0 ODTM	SUCESM	STGM	PROTM	CPU DECM	00200030H	00H

位的位置	位名称	功能
6	E0INM	该位屏蔽 EP0IN 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
5	E0INDTM	该位屏蔽 EP0INDT 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
4	E0ODTM	该位屏蔽 EP0OUTDT 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
3	SUCESM	该位屏蔽 Success 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
2	STGM	该位屏蔽 Stg 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
1	PROTM	该位屏蔽 Protect 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
0	CPUDECM	该位屏蔽 CPUDEC 中断。 1: 屏蔽 0: 不屏蔽 (默认值)

(18) UF0 INT 屏蔽 2 寄存器 (UF0IM2)

该寄存器控制中断源的屏蔽，中断源由 UF0IS2 寄存器表示。

该寄存器可以按字节进行读取或写入。

向该寄存器的对应位写入 1，FW 可以屏蔽来自 USBF 的中断请求 (INTUSBF0) 的发生。

如果 UF0EnIM 寄存器 (n = 1, 3, 7) 的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IM2	BKI2INM	BKI2DTM	BKI1INM	BKI1DTM	0	0	IT2DTM	IT1DTM	00200032H	00H

位的位置	位名称	功能
7, 5	BKInINM	该位屏蔽 BKInIN 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
6, 4	BKInDTM	该位屏蔽 BKInDT 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
1, 0	ITnDTM	该位屏蔽 INTnDT 中断。 1: 屏蔽 0: 不屏蔽 (默认值)

备注 n = 1, 2

(19) UF0 INT 屏蔽 3 寄存器 (UF0IM3)

该寄存器控制中断源的屏蔽，中断源由 UF0IS3 寄存器表示。

该寄存器可以按字节进行读取或写入。

向该寄存器的对应位写入 1，FW 可以屏蔽来自 USBF 的中断请求 (INTUSBF0) 的发生。

如果 UF0EnIM 寄存器 (n = 2, 4) 的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IM3	BKO2 FLM	BKO2 NLM	BKO2 NAKM	BKO2 DTM	BKO1 FLM	BKO1 NLM	BKO1 NAKM	BKO1 DTM	00200034H	00H

位的位置	位名称	功能
7, 3	BKOnFLM	该位屏蔽 BLKOnFL 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
6, 2	BKOnNLM	该位屏蔽 BLKOnNL 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
5, 1	BKOnNAKM	该位屏蔽 BLKOnNK 中断。 1: 屏蔽 0: 不屏蔽 (默认值)
4, 0	BKOnDTM	该位屏蔽 BLKOnDT 中断。 1: 屏蔽 0: 不屏蔽 (默认值)

备注 n = 1, 2

(20) UF0 INT 屏蔽 4 寄存器 (UF0IM4)

该寄存器控制中断源的屏蔽，中断源由 UF0IS4 寄存器表示。

该寄存器可以按字节进行读取或写入。

向该寄存器的对应位写入 1，FW 可以屏蔽来自 USBF 的中断请求 (INTUSBF0) 的发生。

如果 UF0EnIM 寄存器 (n = 1 至 4, 7) 的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IM4	0	0	SETINTM	0	0	0	0	0	00200036H	00H

位的位置	位名称	功能
5	SETINTM	该位屏蔽 SET_INT 中断。 1: 屏蔽 0: 不屏蔽 (默认值)

(21) UF0 INT 清除 0 寄存器 (UF0IC0)

该寄存器控制中断源的清除，中断源由 UF0IS0 寄存器表示。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 FFH。

向该寄存器的对应位写入 0，FW 可以清除一个中断请求。即使可以由硬件自动清除位 0 的位，在被硬件清除之前也可以由 FW 清除。向该寄存器的某一位写入 0 自动设置该位为 1。写入 1 无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IC0	BUS RSTC	RSU SPDC	1	SHORTC	DMA EDC	SET RQC	CLR RQC	EP HALTC	0020003CH	FFH

位的位置	位名称	功能
7	BUSRSTC	该位清除总线复位中断。 0: 清除
6	RSUSPDC	该位清除恢复/挂起中断。 0: 清除
4	SHORTC	该位清除 Short 中断。 0: 清除
3	DMAEDC	该位清除 DMA_END 中断。 0: 清除
2	SETRQC	该位清除 SET_RQ 中断。 0: 清除
1	CLRRQC	该位清除 CLR_RQ 中断。 0: 清除
0	EPHALTC	该位清除 EP_Halt 中断。 0: 清除

(22) UF0 INT 清除 1 寄存器 (UF0IC1)

该寄存器控制中断源的清除，中断源由 UF0IS1 寄存器表示。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 FFH。

向该寄存器的对应位写入 0，FW 可以清除一个中断请求。即使可以由硬件自动清除位 0 的位，在被硬件清除之前也可以由 FW 清除。向该寄存器的某一位写入 0 自动设置该位为 1。写入 1 无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IC1	1	E0INC	E0 INDTC	E0ODTC	SUCESC	STGC	PROTC	CPU DECC	0020003EH	FFH

位的位置	位名称	功能
6	E0INC	该位清除 EP0IN 中断。 0: 清除
5	E0INDTC	该位清除 EP0INDT 中断。 0: 清除
4	E0ODTC	该位清除 EP0OUTDT 中断。 0: 清除
3	SUCESC	该位清除 Success 中断。 0: 清除
2	STGC	该位清除 Stg 中断。 0: 清除
1	PROTC	该位清除 Protect 中断。 0: 清除
0	CPUDECC	该位清除 CPUDEC 中断。 0: 清除

(23) UF0 INT 清除 2 寄存器 (UF0IC2)

该寄存器控制中断源的清除，中断源由 UF0IS2 寄存器表示。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 FFH。

向该寄存器的对应位写入 0，FW 可以清除一个中断请求。即使可以由硬件自动清除位 0 的位，在被硬件清除之前也可以由 FW 清除。向该寄存器的某一位写入 0 自动设置该位为 1。写入 1 无效。

如果 UF0EnIM (n = 1, 3, 7) 寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IC2	BKl2INC	BKl2 DTC	BKl1INC	BKl1 DTC	1	1	IT2DTC	IT1DTC	00200040H	FFH

位的位置	位名称	功能
7, 5	BKInINC	这些位清除 BLKInIN 中断。 0: 清除
6, 4	BKInDTC	这些位清除 BLKInDT 中断。 0: 清除
1, 0	ITnDTC	这些位清除 INTnDT 中断。 0: 清除

备注 n = 1, 2

(24) UF0 INT 清除 3 寄存器 (UF0IC3)

该寄存器控制中断源的清除，中断源由 UF0IS3 寄存器表示。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 FFH。

向该寄存器的对应位写入 0，FW 可以清除一个中断请求。即使可以由硬件自动清除位 0 的位，在被硬件清除之前也可以由 FW 清除。向该寄存器的某一位写入 0 自动设置该位为 1。写入 1 无效。

如果 UF0EnIM (n = 2, 4) 寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IC3	BKO2 FLC	BKO2 NLC	BKO2 NAKC	BKO2 DTC	BKO1 FLC	BKO1 NLC	BKO1 NAKC	BKO1 DTC	00200042H	FFH

位的位置	位名称	功能
7, 3	BKOnFLC	这些位清除 BLKOnFL 中断。 0: 清除
6, 2	BKOnNLC	这些位清除 BLKOnNL 中断。 0: 清除
5, 1	BKOnNAKC	这些位清除 BLKOnNK 中断。 0: 清除
4, 0	BKOnDTC	这些位清除 BLKOnDT 中断。 0: 清除

备注 n = 1, 2

(25) UF0 INT 清除 4 寄存器 (UF0IC4)

该寄存器控制中断源的清除，中断源由 UF0IS4 寄存器表示。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 FFH。

向该寄存器的对应位写入 0，FW 可以清除一个中断请求。即使可以由硬件自动清除位 0 的位，在被硬件清除之前也可以由 FW 清除。向该寄存器的某一位写入 0 自动设置该位为 1。写入 1 无效。

如果 UF0EnIM (n = 1 至 4, 7) 寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IC4	1	1	SETINTC	1	1	1	1	1	00200044H	FFH

位的位置	位名称	功能
5	SETINTC	该位清除 SET_INT 中断。 0: 清除

(26) UF0 INT & DMARQ 寄存器 (UF0IDR)

该寄存器选择通过中断请求或启动 DMA 来报告。

该寄存器可以按字节进行读取或写入。

如果数据存在于 UF0BO1 或 UF0BO2 寄存器中，或者如果数据可以写入到 UF0BI1 或 UF0BI2 寄存器，该寄存器选择通过中断请求或启动请求的 DMA 来报告给 FW。如果启动了请求的 DMA，可以根据第 0 位和第 1 位的设置来选择 DMA 传输模式。

如果 UF0EnIM (n = 1 至 4) 寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

请确保将第 3 位和第 2 位清除为“0”。如果它们被设置为 1，操作无法保证。

注意事项 如果在 DMA 传输中 SET_INTERFACE 请求不支持目标端点，DMA 请求信号立即变为无效，并且对应位被硬件自动清除为 0。

(1/2)

	7	6	5	4	3	2	1	0	地址	复位后
UF0IDR	DQBI2 MS	DQBI1 MS	DQBO2 MS	DQBO1 MS	0	0	MODE1	MODE0	0020004CH	00H

位的位置	位名称	功能
7, 6	DQBInMS	这些位使能 (屏蔽) 一个向 UF0BI _n 寄存器写入的 DMA 传输请求 (对端点 m 的 DMA 请求信号)。当这些位被置 1，写入数据可以应答时，对端点 m 的 DMA 请求信号被有效。如果输入对端点 m 的 DMA 结束信号 (如果 DMA 控制器发出 TC)，这些位被硬件自动清除为 0。要继续 DMA 传输，通过 FW 重置这些位为 1。 1: 允许有效对端点 m 的 DMA 请求信号 (屏蔽 BKInDT 中断) 0: 禁止有效对端点 m 的 DMA 请求信号 (默认值)
5, 4	DQBO _n MS	这些位使能 (屏蔽) 一个对 UF0BO _n 寄存器读取的 DMA 传输请求 (对端点 x 的 DMA 请求信号)。当这些位被置 1，要数据的数据已经在 UF0BO _n 寄存器中准备好，对端点 x 的 DMA 请求信号被有效。如果输入对端点 x 的 DMA 结束信号 (如果 DMA 控制器发出 TC)，这些位被硬件自动清除为 0。当 USBSPxB 信号被有效时，它们也被清除为 0。要继续 DMA 传输，通过 FW 重置这些位为 1。 1: 允许有效对端点 x 的 DMA 请求信号 (屏蔽 BKOnDT 中断) 0: 禁止有效对端点 x 的 DMA 请求信号 (默认值)

备注 n = 1, 2
 当 n = 1 时, m = 1 且 x = 2
 当 n = 2 时, m = 3 且 x = 4

位的位置	位名称	功能			
1, 0	MODE1, MODE0	这些位选择 DMA 传输模式。			
		MODE1	MODE0	模式	备注
		1	1	禁止设置	操作无法保证
		1	0	需求模式	只要有数据, DMA 请求信号就被有效。如果没有数据, 它就变为无效。
		0	X	单次模式	每次执行 DMA 传输后, DMA 请求信号变为无效。
		备注 <ol style="list-style-type: none"> 1. X: 无须理会 2. 当使用 USB 进行 DMA 传输时, 只有在单步传输模式下可以进行传输。于是, 无论 MODE1 和 MODE0 被设置位需求模式或者单次模式, 操作都不会不同。 			

(27) UF0 DMA 状态 0 寄存器 (UF0DMS0)

该寄存器用于表示端点 1 至端点 4 的 DMA 状态。

该寄存器是只读的，按字节进行读取。

如果 UF0EnIM 寄存器 (n = 1 至 4) 的设置与接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0DMS0	0	0	DQE4	DQE3	DQE2	DQE1	0	0	0020004EH	00H

位的位置	位名称	功能
5	DQE4	该位表明发出一个从端点 4 到存储器的 DMA 读取请求。 1: 发出来自端点 4 的 DMA 读取请求 0: 没有发出来自端点 4 的 DMA 读取请求 (默认值)
4	DQE3	该位表明发出一个从存储器到端点 3 的 DMA 写入请求。 请注意，即使数据是在端点 3 内 (在 BKI2DED 位被置 1 后，FIFO 还未满时)，DMA 请求信号立即被有效，且当 UF0IDR 寄存器的 DQBI2MS 位被置 1 时，启动 DMA 传输。 1: 发出对端点 3 的 DMA 写入请求 0: 没有发出对端点 3 的 DMA 写入请求 (默认值)
3	DQE2	该位表明发出一个从端点 2 到存储器的 DMA 读取请求。 1: 发出来自端点 2 的 DMA 读取请求 0: 没有发出来自端点 2 的 DMA 读取请求 (默认值)
2	DQE1	该位表明发出一个从存储器到端点 1 的 DMA 写入请求。 请注意，即使数据是在端点 1 内 (在 BKI1DED 位被置 1 后，FIFO 还未满时)，DMA 请求信号立即被有效，且当 UF0IDR 寄存器的 DQBI1MS 位被置 1 时，启动 DMA 传输。 1: 发出对端点 1 的 DMA 写入请求 0: 没有发出对端点 1 的 DMA 写入请求 (默认值)

(28) UF0 DMA 状态 1 寄存器 (UF0DMS1)

该寄存器用于表示端点 1 至端点 4 的 DMA 状态。

该寄存器是只读的，按字节进行读取。

如果 UF0EnIM 寄存器 (n = 1 至 4) 的设置与接口的当前设置不支持各个端点，相关的位无效。

当该寄存器被读取时，每位都被自动清除为 0。即使当该寄存器被读取时，但是 UF0ISO 寄存器的第 4 位和第 3 位不被清除为 0，如果 SET_INTERFACE 请求不再支持目标端点，每位都被硬件自动清除为 0（但是，DMA_END 中断请求和 Short 中断请求不被清除）。

	7	6	5	4	3	2	1	0	地址	复位后
UF0DMS1	DEDE4	DSPE4	DEDE3	DEDE2	DSPE2	DEDE1	0	0	00200050H	00H

位的位置	位名称	功能
7, 5, 4, 2	DEDEn	这些位表明对端点 n 的 DMA 结束 (TC) 信号被有效，且发出从端点 n 到存储器的 DMA 读取请求时，DMA 停止。 1: 对端点 n 的 DMA 结束信号是有效的 0: 对端点 n 的 DMA 结束信号是无效的（默认值）
6, 3	DSPEm	这些位表明发出了从端点 m 到存储器的 DMA 读取请求，DMA 已经被停止，因为接收到得数据是一个短包，并且没有其它更多数据需要传输。 1: DMASTOP_EPm 信号是有效的 0: DMASTOP_EPm 信号是无效的（默认值）

备注 n = 1 至 4
 m = 2, 4

(29) UF0 FIFO 清除 0 寄存器 (UF0FIC0)

该寄存器清除每个 FIFO。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 00H。

向该寄存器的对应位写入 1，FW 可以清除目标 FIFO。写入 1 的位被自动清除为 0。向该位写入 0 无效。

如果 UF0EnIM 寄存器 (n = 1, 3, 7) 的设置与接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0FIC0	BKl2SC	BKl2CC	BKl1SC	BKl1CC	ITR2C	ITR1C	EP0WC	EP0RC	00200060H	00H

位的位置	位名称	功能
7, 5	BKInSC	这些位只清除 UF0BIn 寄存器的 SIE 端的 FIFO (复位该计数器)。 1: 清除 BKInNK 位置为 1, 当对端点 m 的 IN 令牌正在被处理时, 写入该位无效。 通过清除 FIFO, BKInNK 位被自动清除为 0。当使用这些位时, 请确保 CPU 端的 FIFO 为空。
6, 4	BKInCC	这些位只清除 UF0BIn 寄存器的 CPU 端的 FIFO (复位该计数器)。 1: 清除
3, 2	ITRnC	这些位只清除 UF0INTn 寄存器 (复位该计数器)。 1: 清除 ITnNK 位置为 1, 当对端点 x 的 IN 令牌正在被处理时, 写入该位无效。 通过清除 FIFO, ITnNK 位被自动清除为 0。
1	EP0WC	该位清除 UF0E0W 寄存器 (复位该计数器)。 1: 清除 EP0NKW 位置为 1, 当对端点 0 的 IN 令牌正在被处理时, 写入该位无效。 通过清除 FIFO, EP0NKW 位被自动清除为 0。
0	EP0RC	该位清除 UF0E0R 寄存器 (复位该计数器)。 1: 清除 当 EP0NKR 位被设置为 1 (除已经被 FW 设置外) 时, 通过清除 FIFO, EP0NKR 位被自动清除为 0。

备注 n = 1, 2
 当 n = 1 时, m = 1 且 x = 7
 当 n = 2 时, m = 3 且 x = 8

(30) UF0 FIFO 清除 1 寄存器 (UF0FIC1)

该寄存器清除每个 FIFO。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 00H。

向该寄存器的对应位写入 1，FW 可以清除目标 FIFO。写入 1 的位被自动清除为 0。向该位写入 0 无效。

如果 UF0EnIM 寄存器 (n = 2, 4) 的设置与接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0FIC1	0	0	0	0	BKO2C	BKO2CC	BKO1C	BKO1CC	00200062H	00H

位的位置	位名称	功能
3, 1	BKOnC	这些位清除 UF0BOn 寄存器的 SIE 端和 CPU 端的 FIFO (复位该计数器)。 1: 清除 当 BKOnNK 位被置为 1 时 (除已经由 FW 设置外), 通过清除 FIFO, BKOnNK 位被自动清除为 0。
2, 0	BKOnCC	这些位只清除 UF0BOn 寄存器的 CPU 端的 FIFO (复位该计数器)。 1: 清除 当 BKOnNK 位被置为 1 时 (除已经由 FW 设置外), 通过清除 FIFO, BKOnNK 位被自动清除为 0。

备注 n = 1, 2

(31) UF0 数据结束寄存器 (UF0DEND)

该寄存器用于报告写入到传输系统的结束。

该寄存器只能写入，可以按字节进行写入（但是，第 7 位和第 6 位可以进行读取和写入）。如果读取该寄存器，读到的值是 00H。

向该寄存器的对应位写入 1，FW 可以清除目标 FIFO。写入 1 的位被自动清除为 0。向该位写入 0 无效。

如果 UF0EnIM 寄存器 (n = 1, 3, 7) 的设置与接口的当前设置不支持各个端点，相关的位无效。

(1/2)

	7	6	5	4	3	2	1	0	地址	复位后
UF0DEND	BKI2T	BKI1T	0	IT2DEND	IT1DEND	BKI2DED	BKI1DED	E0DED	0020006AH	00H

位的位置	位名称	功能
7, 6	BKInT	如果因为 DMA，导致 UF0BIn 寄存器的 CPU 端的 FIFO 装满，这些位指定是否自动执行 FIFO 的切换。 1: 只要 FIFO 装满，立即自动执行 FIFO 的切换操作 0: 即使 FIFO 装满，也不自动执行 FIFO 的切换操作（默认值）
4, 3	ITnDEND	设置该位为 1 来发送 UF0INTn 寄存器的数据。该位被置为 1 时，ITnNK 位被置为 1 并且执行数据传输。 1: 发送一个短包 0: 不发送短包（默认值） 如果 UF0FIC0 寄存器的 ITRnC 位被置为 1，然后这些位被置为 1（UF0INTn 寄存器的计数器 = 0 且 UF0EPS0 寄存器的对应位 = 1），发送一个 Null 包（数据长度为 0）。 如果数据存在于 UF0INTn 寄存器中，并且如果该位被置为 1（UF0INTn 寄存器的计数器 ≠ 0 且 UF0EPS0 寄存器的对应位 = 1），一个短包被发送。 当 FIFO 装满时，这些位被硬件自动控制。

备注 n = 1, 2

位的位置	位名称	功能
2, 1	BKInDED	<p>向 UF0BIn 寄存器写入发送数据完成时，设置这些位为 1。当这些位被置为 1 时，FIFO 被尽快切换，BKInNK 位被设置为 1，并且数据被传输。</p> <p>1: 发送一个短包 0: 不发送短包（默认值）</p> <p>这些位控制 CPU 端的 FIFO。</p> <p>如果 UF0FIC0 寄存器的 BKInCC 位被置为 1，然后这些位被置为 1（UF0BIn 寄存器的计数器 = 0），发送一个 Null 包（数据长度为 0）。</p> <p>如果数据存在于 UF0BIn 寄存器中，且如果这些位被置为 1（UF0BIn 寄存器的计数器 ≠ 0），如果 FIFO 未满，发送一个短包。</p> <p>如果因为 DMA，导致 UF0BIn 寄存器的 CPU 端的 FIFO 装满，且 PIO 或 BKInT 位置 1，即使这些位没有被设置为 1，硬件也会启动数据发送。</p> <p>如果因为 DMA，导致 UF0BIn 寄存器的 CPU 端的 FIFO 装满，且 BKInT 位被清除为 0，请确保设置这些位为 1（参见 20.6.3 (3) UF0 EPNAK 寄存器 (UF0EN)）。</p>
0	E0DED	<p>设置该位为 1 来发送 UF0E0W 寄存器的数据。当该位被置为 1 时，EP0NKW 位被设置为 1 并且执行数据传输。</p> <p>1: 发送一个短包 0: 不发送短包（默认值）</p> <p>如果 UF0FIC0 寄存器的 EP0WC 位被置为 1，然后该位被置为 1（UF0E0W 寄存器的计数器 = 0 且 UF0EPS0 寄存器的第 1 位 = 1），发送一个 Null 包（数据长度为 0）。</p> <p>如果数据存在于 UF0E0W 寄存器中，并且如果该位被设置为 1（UF0E0W 寄存器的计数器 ≠ 0 且 UF0EPS0 寄存器的第 1 位 = 1），如果 FIFO 未满，一个短包被发送。</p>

备注 n = 1, 2

(32) UF0 GPR 寄存器 (UF0GPR)

该寄存器用于控制 USBF 和 USB 接口。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 00H。请确保清除第 7 位至第 1 位为“0”。

向该寄存器的第 0 位写入 1，FW 可以复位 USBF。1 写入该位后，该位被自动清除为 0。向该位写入 0 无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0GPR	0	0	0	0	0	0	0	MRST	0020006EH	00H

位的位置	位名称	功能
0	MRST	设置该位为 1 来复位 USBF。 1: 复位 实际上，在该位被 FW 设置为 1 且写信号变无效后的两个 USB 时钟后，USBF 被复位。当系统时钟工作时由 MRST 位复位 USBF 与通过 RESET 引脚复位（硬件复位）的作用相同（寄存器值回到默认值）。 但是，UF0CS 和 UF0BC 寄存器不被 MRST 位复位。

(33) UF0 模式控制寄存器 (UF0MODC)

该寄存器控制 CPUDEC 处理。

该寄存器可以按字节进行读取或写入。

通过设置该寄存器的每一位，UF0MODS 寄存器的设置可以被更改。只有硬件复位以及 UF0GRP 寄存器的 MRST 位被设置为 1 时，该寄存器的此位才会被自动清除为 0。

即使该寄存器的此位由硬件自动设置为 1，通过 FW 进行的设置会优先作用。

请确保清除第 7 位和第 5 位为“0”。如果这些位被置为 1，操作无法保证。

注意事项 提供该寄存器是用于调试目的。通常情况下，不要设置该寄存器，除非为了检验操作或使用一个特殊模式时。

	7	6	5	4	3	2	1	0	地址	复位后
UF0MODC	0	CDC GDST	0	0	0	0	0	0	00200074H	00H

位的位置	位名称	功能
6	CDCGDST	设置该位为 1，将 GET_DESCRIPTOR 配置请求切换到 CPUDEC 处理。通过设置该位为 1，UF0MODS 寄存器的 CDCGD 位可以被强制置为 1。 1: 将 GET_DESCRIPTOR 配置请求强制更改为 CPUDEC 处理（设置 UF0MODS 寄存器的 CDCGD 位为 1）。 0: 自动处理 GET_DESCRIPTOR 配置请求（默认值）。

(34) UF0 模式状态寄存器 (UF0MODS)

该寄存器用于表示配置状态。

该寄存器是只读的，可以按字节进行读取。

	7	6	5	4	3	2	1	0	地址	复位后
UF0MODS	0	CDCGD	0	MPACK	DFLT	CONF	0	0	00200078H	00H

位的位置	位名称	功能
6	CDCGD	该位指定对 GET_DESCRIPTOR 配置请求是否执行 CPUDEC 处理。 1: 将 GET_DESCRIPTOR 配置请求强制更改为 CPUDEC 处理。 0: 自动处理 GET_DESCRIPTOR 配置请求 (默认值)。
4	MPACK	该位表明端点 0 的发送包大小。 1: 发送一个不是 8 字节的包。 0: 发送一个 8 字节的包 (默认值)。 在 GET_DESCRIPTOR 设备请求被处理后 (状态阶段的正常完成), 该位被硬件自动设置为 1。在 USBF 被复位前 (它不会被总线复位清除为 0), 它不会被清除为 0。 如果该位没有被置为 1, 硬件只按照 8 字节单位传输自动执行的请求。因此, 即使在 GET_DESCRIPTOR 设备请求完成之前, OUT 令牌发送的 8 个以上字节的数据要由 FW 处理, 数据也可以被正确接收。 如果端点 0 的大小是 8 字节, 该位被忽略。
3	DFLT	该位表明默认状态 (DFLT 位 = 1)。 1: 允许响应。 0: 禁止响应 (总是无响应) (默认值)。 该位由总线复位自动设置为 1。在该位被置为 1 之前, 不会响应对所有端点的事务。
2	CONF	该位表明 SET_CONFIGURATION 请求是否已经完成。 1: SET_CONFIGURATION 请求已经完成。 0: SET_CONFIGURATION 请求尚未完成 (默认值)。 当通过 SET_CONFIGURATION 请求接收到配置值 = 1 时, 该位被设置为 1。 除非该位被置为 1, 对端点 0 以外的端点的访问被忽略。 当通过 SET_CONFIGURATION 请求接收到配置值 = 1 时, 该位被清除为 0。当检测到总线复位时, 该位也被清除为 0。

(35) UF0 有效接口编号寄存器 (UF0AIFN)

该寄存器用于设置正确响应 GET/SET_INTERFACE 请求的有效接口编号。因为接口 0 总是有效，接口 1 到 4 可以选择。

该寄存器可以按字节进行读取或写入。

	7	6	5	4	3	2	1	0	地址	复位后
UF0AIFN	ADDIF	0	0	0	0	0	IFNO1	IFNO0	00200080H	00H

位的位置	位名称	功能															
7	ADDIF	该位允许使用 0 以外的接口编号。 1: 支持 IFNO1 和 IFNO0 位指定的接口编号。 0: 仅支持接口 0 (默认值)。 当该位没有被置为 1 时，该寄存器的第 1 位和第 0 位的设置无效。															
1, 0	IFNO1, IFNO0	这些位指定可以支持的接口编号范围。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>IFNO1</th> <th>IFNO0</th> <th>有效接口编号</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>0, 1, 2, 3, 4</td> </tr> <tr> <td>1</td> <td>0</td> <td>0, 1, 2, 3</td> </tr> <tr> <td>0</td> <td>1</td> <td>0, 1, 2</td> </tr> <tr> <td>0</td> <td>0</td> <td>0, 1</td> </tr> </tbody> </table>	IFNO1	IFNO0	有效接口编号	1	1	0, 1, 2, 3, 4	1	0	0, 1, 2, 3	0	1	0, 1, 2	0	0	0, 1
IFNO1	IFNO0	有效接口编号															
1	1	0, 1, 2, 3, 4															
1	0	0, 1, 2, 3															
0	1	0, 1, 2															
0	0	0, 1															

(36) UF0 有效复用设置寄存器 (UF0AAS)

该寄存器用于指定接口编号和复用设置之间的一个链接。

该寄存器可以按字节进行读取或写入。

V850ES/JG3-U 和 V850ES/JH3-U 的 USBF 可以为一个接口设置 5 级复用设置（复用设置 0、1、2、3 和 4 可以定义）和 2 级复用设置（复用设置 0 和 1 可以定义）。

	7	6	5	4	3	2	1	0	地址	复位后
UF0AAS	ALT2	IFAL21	IFAL20	ALT2EN	ALT5	IFAL51	IFAL50	ALT5EN	00200082H	00H

位的位置	位名称	功能															
7, 3	ALn	这些位指定是否有 n 级复用设置与接口 0 链接。当这些位被置为 1 时，IFALn1 和 IFALn0 位的设置无效。 1: n 级复用设置与接口 0 链接。 0: n 级复用设置与接口 0 没有链接（默认值）。															
6, 5, 2, 1	IFALn1, IFALn0	这些位指定要与 n 级复用设置链接的接口编号。如果接口编号超出 UF0AIFN 寄存器指定的范围，n 级复用设置无效（ALnEN 位 = 0）。 <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 15%;">IFALn1</th> <th style="width: 15%;">IFALn0</th> <th style="width: 70%;">将被链接的接口编号</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>链接到接口 4。</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>链接到接口 3。</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>链接到接口 2。</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>链接到接口 1。</td> </tr> </tbody> </table> 对同一个接口编号，不要同时链接 5 级复用设置和 2 级复用设置。	IFALn1	IFALn0	将被链接的接口编号	1	1	链接到接口 4。	1	0	链接到接口 3。	0	1	链接到接口 2。	0	0	链接到接口 1。
IFALn1	IFALn0	将被链接的接口编号															
1	1	链接到接口 4。															
1	0	链接到接口 3。															
0	1	链接到接口 2。															
0	0	链接到接口 1。															
4, 0	ALnEN	这些位使 n 级复用设置有效。除非这些位被置为 1，否则 ALn、IFALn1 和 IFALn0 位的设置无效。 1: n 级复用设置有效。 0: n 级复用设置无效（默认值）。															

备注 n = 2, 5

例如，当 UF0AIFN 寄存器被设置为 82H，且 UF0AAS 寄存器被设置为 15H，接口 0、1、2 和 3 有效。接口 0 和 2 只支持复用设置 0。接口 1 支持复用设置 0 和 1，并且接口 3 支持复用设置 0、1、2、3 和 4。对于这些设置，要求 GET_INTERFACE wIndex = 0/1/2/3、SET_INTERFACE wValue = 0 & wIndex = 0/2、SET_INTERFACE wValue = 0/1 & wIndex = 1 且 SET_INTERFACE wValue = 0/1/2/3/4 & wIndex = 3 被自动响应，并且对于其它 GET/SET_INTERFACE 请求，产生一个 STALL 响应。

(37) UF0 复用设置状态寄存器 (UF0ASS)

该寄存器用于表示复用设置的当前状态。

该寄存器是只读的，按字节进行读取。

当 SET_INT 中断请求已经发出时，检查该寄存器。通过 SET_INTERFACE 请求接收到的值反映在 UF0IFn 寄存器 (n = 0 到 4) 和该寄存器上。

	7	6	5	4	3	2	1	0	地址	复位后
UF0ASS	0	0	0	0	AL5ST3	AL5ST2	AL5ST1	AL2ST	00200084H	00H

位的位置	位名称	功能																								
3 至 1	AL5ST3 至 AL5ST1	<p>这些位表明 5 级复用设置的当前状态。</p> <table border="1"> <thead> <tr> <th>AL5ST3</th> <th>AL5ST2</th> <th>AL5ST1</th> <th>选择的复用设置编号</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0</td> <td>复用设置 4</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>复用设置 3</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>复用设置 2</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>复用设置 1</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>复用设置 0</td> </tr> </tbody> </table>	AL5ST3	AL5ST2	AL5ST1	选择的复用设置编号	1	0	0	复用设置 4	0	1	1	复用设置 3	0	1	0	复用设置 2	0	0	1	复用设置 1	0	0	0	复用设置 0
AL5ST3	AL5ST2	AL5ST1	选择的复用设置编号																							
1	0	0	复用设置 4																							
0	1	1	复用设置 3																							
0	1	0	复用设置 2																							
0	0	1	复用设置 1																							
0	0	0	复用设置 0																							
0	AL2ST	<p>该位表明 2 级复用设置的当前状态 (选择的复用设置编号)。</p> <p>1: 复用设置 1 0: 复用设置 0</p>																								

(38) UF0 端点 1 接口映射寄存器 (UF0E1IM)

该寄存器用于指定端点 1 对哪个接口和复用设置有效。

该寄存器可以按字节进行读取或写入。

该寄存器的设置和通过 SET_INTERFACE 请求选择的复用设置表明端点 1 当前是否有效，并且硬件决定如何响应 GET_STATUS/CLEAR_FEATURE/SET_FEATURE 端点 1 请求和到端点 1 的 IN 事务，以及相关的位是有效或是无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E1IM	E1EN2	E1EN1	E1EN0	E12AL1	E15AL4	E15AL3	E15AL2	E15AL1	00200086H	00H

位的位置	位名称	功能																																				
7 至 5	E1EN2 至 E1EN0	<p>这些位设置一个端点 1 的接口与 2 / 5 级复用设置之间的链接。端点被链接到复用设置 0。链接到复用设置 0 的端点不能从复用设置 1 至 4 中排除。</p> <table border="1" style="width: 100%; border-collapse: collapse; margin: 5px 0;"> <thead> <tr> <th style="width: 10%;">E1EN2</th> <th style="width: 10%;">E1EN1</th> <th style="width: 10%;">E1EN0</th> <th style="width: 70%;">链接状态</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>没有与接口链接</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td></td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>与接口 4 和复用设置 0 链接</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>与接口 3 和复用设置 0 链接</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>与接口 2 和复用设置 0 链接</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>与接口 1 和复用设置 0 链接</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>与接口 0 和复用设置 0 链接</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>没有与接口链接 (默认值)</td> </tr> </tbody> </table> <p>当这些位被设置为 110 或 111 时，即使 E12AL1 位被清除为 0，它们也无效。如果端点被链接，设置 UF0MODS 寄存器的 CONF 位为 1，表明端点 1 有效。</p>	E1EN2	E1EN1	E1EN0	链接状态	1	1	1	没有与接口链接	1	1	0		1	0	1	与接口 4 和复用设置 0 链接	1	0	0	与接口 3 和复用设置 0 链接	0	1	1	与接口 2 和复用设置 0 链接	0	1	0	与接口 1 和复用设置 0 链接	0	0	1	与接口 0 和复用设置 0 链接	0	0	0	没有与接口链接 (默认值)
E1EN2	E1EN1	E1EN0	链接状态																																			
1	1	1	没有与接口链接																																			
1	1	0																																				
1	0	1	与接口 4 和复用设置 0 链接																																			
1	0	0	与接口 3 和复用设置 0 链接																																			
0	1	1	与接口 2 和复用设置 0 链接																																			
0	1	0	与接口 1 和复用设置 0 链接																																			
0	0	1	与接口 0 和复用设置 0 链接																																			
0	0	0	没有与接口链接 (默认值)																																			
4	E12AL1	<p>当 2 级复用设置和链接到接口的复用设置被置为 1 时，该位使端点 1 有效。</p> <p>1: 在设置复用设置 1 时 CONF 位 = 1，使端点有效。</p> <p>0: 即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 (默认值)。</p> <p>当 E15AL4 至 E15AL1 位为 0000 时，该位有效。</p>																																				
3 至 0	E15ALn	<p>当 5 级复用设置和链接到接口的复用设置被置为 n 时，这些位使端点 1 有效。</p> <p>1: 在设置复用设置 1 时 CONF 位 = 1，使端点有效。</p> <p>0: 即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 (默认值)。</p>																																				

备注 n = 1 至 4

(39) UF0 端点 2 接口映射寄存器 (UF0E2IM)

该寄存器用于指定端点 2 对哪个接口和复用设置有效。

该寄存器可以按字节进行读取或写入。

该寄存器的设置和通过 SET_INTERFACE 请求选择的复用设置表明端点 2 当前是否有效，并且硬件决定如何响应 GET_STATUS/CLEAR_FEATURE/SET_FEATURE 端点 2 请求和到端点 2 的 IN 事务，以及相关的位是有效或是无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E2IM	E2EN2	E2EN1	E2EN0	E22AL1	E25AL4	E25AL3	E25AL2	E25AL1	00200088H	00H

位的位置	位名称	功能																																				
7 至 5	E2EN2 至 E2EN0	<p>这些位设置一个端点 2 的接口与 2 / 5 级复用设置之间的链接。端点被链接到复用设置 0。链接到复用设置 0 的端点不能从复用设置 1 至 4 中排除。</p> <table border="1"> <thead> <tr> <th>E2EN2</th> <th>E2EN1</th> <th>E2EN0</th> <th>链接状态</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> <td>没有与接口链接</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>与接口 4 和复用设置 0 链接</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>与接口 3 和复用设置 0 链接</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>与接口 2 和复用设置 0 链接</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>与接口 1 和复用设置 0 链接</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>与接口 0 和复用设置 0 链接</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>没有与接口链接 (默认值)</td> </tr> </tbody> </table> <p>当这些位被设置为 110 或 111 时，即使 E22AL1 位被清除为 0，它们也无效。如果端点被链接，设置 UF0MODS 寄存器的 CONF 位为 1，表明端点 2 有效。</p>	E2EN2	E2EN1	E2EN0	链接状态	1	1	1	没有与接口链接	1	1	0		1	0	1	与接口 4 和复用设置 0 链接	1	0	0	与接口 3 和复用设置 0 链接	0	1	1	与接口 2 和复用设置 0 链接	0	1	0	与接口 1 和复用设置 0 链接	0	0	1	与接口 0 和复用设置 0 链接	0	0	0	没有与接口链接 (默认值)
E2EN2	E2EN1	E2EN0	链接状态																																			
1	1	1	没有与接口链接																																			
1	1	0																																				
1	0	1	与接口 4 和复用设置 0 链接																																			
1	0	0	与接口 3 和复用设置 0 链接																																			
0	1	1	与接口 2 和复用设置 0 链接																																			
0	1	0	与接口 1 和复用设置 0 链接																																			
0	0	1	与接口 0 和复用设置 0 链接																																			
0	0	0	没有与接口链接 (默认值)																																			
4	E22AL1	<p>当 2 级复用设置和链接到接口的复用设置被置为 1 时，该位使端点 2 有效。</p> <p>1: 在设置复用设置 1 时 CONF 位 = 1，使端点有效。</p> <p>0: 即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 (默认值)。</p> <p>当 E25AL4 至 E25AL1 位为 0000 时，该位有效。</p>																																				
3 至 0	E25ALn	<p>当 5 级复用设置和链接到接口的复用设置被设置为 n 时，这些位使端点 2 有效。</p> <p>1: 在设置复用设置 1 时 CONF 位 = 1，使端点有效。</p> <p>0: 即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 (默认值)。</p>																																				

备注 n = 1 至 4

(40) UF0 端点 3 接口映射寄存器 (UF0E3IM)

该寄存器用于指定端点 3 对哪个接口和复用设置有效。

该寄存器可以按字节进行读取或写入。

该寄存器的设置和通过 SET_INTERFACE 请求选择的复用设置表明端点 3 当前是否有效，并且硬件决定如何响应 GET_STATUS/CLEAR_FEATURE/SET_FEATURE 端点 3 请求和到端点 3 的 IN 事务，以及相关的位是有效或是无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E3IM	E3EN2	E3EN1	E3EN0	E32AL1	E35AL4	E35AL3	E35AL2	E35AL1	0020008AH	00H

位的位置	位名称	功能																																				
7 至 5	E3EN2 至 E3EN0	<p>这些位设置一个端点 3 的接口与 2 / 5 级复用设置之间的链接。端点被链接到复用设置 0。链接到复用设置 0 的端点不能从复用设置 1 至 4 中排除。</p> <table border="1" style="width: 100%; border-collapse: collapse; margin: 5px 0;"> <thead> <tr> <th style="width: 10%;">E3EN2</th> <th style="width: 10%;">E3EN1</th> <th style="width: 10%;">E3EN0</th> <th style="width: 70%;">链接状态</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> <td>没有与接口链接</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>与接口 4 和复用设置 0 链接</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>与接口 3 和复用设置 0 链接</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>与接口 2 和复用设置 0 链接</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>与接口 1 和复用设置 0 链接</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>与接口 0 和复用设置 0 链接</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>没有与接口链接 (默认值)</td> </tr> </tbody> </table> <p>当这些位被设置为 110 或 111 时，即使 E32AL1 位被清除为 0，它们也无效。如果端点被链接，设置 UF0MODS 寄存器的 CONF 位为 1，表明端点 3 有效。</p>	E3EN2	E3EN1	E3EN0	链接状态	1	1	1	没有与接口链接	1	1	0		1	0	1	与接口 4 和复用设置 0 链接	1	0	0	与接口 3 和复用设置 0 链接	0	1	1	与接口 2 和复用设置 0 链接	0	1	0	与接口 1 和复用设置 0 链接	0	0	1	与接口 0 和复用设置 0 链接	0	0	0	没有与接口链接 (默认值)
E3EN2	E3EN1	E3EN0	链接状态																																			
1	1	1	没有与接口链接																																			
1	1	0																																				
1	0	1	与接口 4 和复用设置 0 链接																																			
1	0	0	与接口 3 和复用设置 0 链接																																			
0	1	1	与接口 2 和复用设置 0 链接																																			
0	1	0	与接口 1 和复用设置 0 链接																																			
0	0	1	与接口 0 和复用设置 0 链接																																			
0	0	0	没有与接口链接 (默认值)																																			
4	E32AL1	<p>当 2 级复用设置和链接到接口的复用设置被置为 1 时，该位使端点 3 有效。</p> <p>1: 在设置复用设置 1 时 CONF 位 = 1，使端点有效。</p> <p>0: 即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 (默认值)。</p> <p>当 E35AL4 至 E35AL1 位为 0000 时，该位有效。</p>																																				
3 至 0	E35ALn	<p>当 5 级复用设置和链接到接口的复用设置被置为 n 时，这些位使端点 3 有效。</p> <p>1: 在设置复用设置 1 时 CONF 位 = 1，使端点有效。</p> <p>0: 即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 (默认值)。</p>																																				

备注 n = 1 至 4

(41) UF0 端点 4 接口映射寄存器 (UF0E4IM)

该寄存器用于指定端点 4 对哪个接口和复用设置有效。

该寄存器可以按字节进行读取或写入。

该寄存器的设置和通过 SET_INTERFACE 请求选择的复用设置表明端点 4 当前是否有效，并且硬件决定如何响应 GET_STATUS/CLEAR_FEATURE/SET_FEATURE 端点 4 请求和到端点 4 的 IN 事务，以及相关的位是有效或是无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E4IM	E4EN2	E4EN1	E4EN0	E42AL1	E45AL4	E45AL3	E45AL2	E45AL1	0020008CH	00H

位的位置	位名称	功能			
7 至 5	E4EN2 至 E4EN0	这些位设置一个端点 4 的接口与 2 / 5 级复用设置之间的链接。端点被链接到复用设置 0。链接到复用设置 0 的端点不能从复用设置 1 至 4 中排除。			
		E4EN2	E4EN1	E4EN0	链接状态
		1	1	1	没有与接口链接
		1	1	0	
		1	0	1	与接口 4 和复用设置 0 链接
		1	0	0	与接口 3 和复用设置 0 链接
		0	1	1	与接口 2 和复用设置 0 链接
		0	1	0	与接口 1 和复用设置 0 链接
		0	0	1	与接口 0 和复用设置 0 链接
		0	0	0	没有与接口链接 (默认值)
		当这些位被设置为 110 或 111 时，即使 E42AL1 位被清除为 0，它们也无效。如果端点被链接，设置 UF0MODS 寄存器的 CONF 位为 1，表明端点 4 有效。			
4	E42AL1	当 2 级复用设置和链接到接口的复用设置被置为 1 时，该位使端点 4 有效。 1: 在设置复用设置 1 时 CONF 位 = 1，使端点有效。 0: 即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 (默认值)。 当 E45AL4 至 E45AL1 位为 0000 时，该位有效。			
3 至 0	E45ALn	当 5 级复用设置和链接到接口的复用设置被设置为 n 时，这些位使端点 4 有效。 1: 在设置复用设置 1 时 CONF 位 = 1，使端点有效。 0: 即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 (默认值)。			

备注 n = 1 至 4

(42) UF0 端点 7 接口映射寄存器 (UF0E7IM)

该寄存器用于指定端点 7 对哪个接口和复用设置有效。

该寄存器可以按字节进行读取或写入。

该寄存器的设置和通过 SET_INTERFACE 请求选择的复用设置表明端点 7 当前是否有效，并且硬件决定如何响应 GET_STATUS/CLEAR_FEATURE/SET_FEATURE 端点 7 请求和到端点 7 的 IN 事务，以及相关的位是有效或是无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E7IM	E7EN2	E7EN1	E7EN0	E72AL1	E75AL4	E75AL3	E75AL2	E75AL1	0020008CH	00H

位的位置	位名称	功能																																				
7 至 5	E7EN2 至 E7EN0	<p>这些位设置一个端点 7 的接口与 2 / 5 级复用设置之间的链接。端点被链接到复用设置 0。链接到复用设置 0 的端点不能从复用设置 1 至 4 中排除。</p> <table border="1" style="width: 100%; border-collapse: collapse; margin: 5px 0;"> <thead> <tr> <th style="width: 10%;">E7EN2</th> <th style="width: 10%;">E7EN1</th> <th style="width: 10%;">E7EN0</th> <th style="width: 70%;">链接状态</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>没有与接口链接</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td></td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>与接口 4 和复用设置 0 链接</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>与接口 3 和复用设置 0 链接</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>与接口 2 和复用设置 0 链接</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>与接口 1 和复用设置 0 链接</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>与接口 0 和复用设置 0 链接</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>没有与接口链接 (默认值)</td> </tr> </tbody> </table> <p>当这些位被设置为 110 或 111 时，即使 E72AL1 位被清除为 0，它们也无效。如果端点被链接，设置 UF0MODS 寄存器的 CONF 位为 1，表明端点 7 有效。</p>	E7EN2	E7EN1	E7EN0	链接状态	1	1	1	没有与接口链接	1	1	0		1	0	1	与接口 4 和复用设置 0 链接	1	0	0	与接口 3 和复用设置 0 链接	0	1	1	与接口 2 和复用设置 0 链接	0	1	0	与接口 1 和复用设置 0 链接	0	0	1	与接口 0 和复用设置 0 链接	0	0	0	没有与接口链接 (默认值)
E7EN2	E7EN1	E7EN0	链接状态																																			
1	1	1	没有与接口链接																																			
1	1	0																																				
1	0	1	与接口 4 和复用设置 0 链接																																			
1	0	0	与接口 3 和复用设置 0 链接																																			
0	1	1	与接口 2 和复用设置 0 链接																																			
0	1	0	与接口 1 和复用设置 0 链接																																			
0	0	1	与接口 0 和复用设置 0 链接																																			
0	0	0	没有与接口链接 (默认值)																																			
4	E72AL1	<p>当 2 级复用设置和链接到接口的复用设置被置为 1 时，该位使端点 7 有效。</p> <p>1: 在设置复用设置 1 时 CONF 位 = 1，使端点有效。</p> <p>0: 即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 (默认值)。</p> <p>当 E75AL4 至 E75AL1 位为 0000 时，该位有效。</p>																																				
3 至 0	E75ALn	<p>当 5 级复用设置和链接到接口的复用设置被置为 n 时，这些位使端点 7 有效。</p> <p>1: 在设置复用设置 1 时 CONF 位 = 1，使端点有效。</p> <p>0: 即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 (默认值)。</p>																																				

备注 n = 1 至 4

20.6.4 数据保持寄存器

(1) UF0 EP0 读取寄存器 (UF0E0R)

UF0E0R 寄存器是一个 64 字节 FIFO，用于存储到/从端点 0 控制传输的数据阶段中从主机发送的 OUT 数据。该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当从主机接收到数据时，硬件将数据自动传输到 UF0E0R 寄存器。当数据被正确接收后，UF0IS1 寄存器的 E0ODT 位被置为 1。UF0E0L 保持已接收到数据的数量，并且发出中断请求 (INTUSBF0)。当 UF0E0L 寄存器接收数据时，它总是更新已接收到数据的长度。如果最后传输是正确接收，产生中断请求。如果接收异常，UF0E0L 寄存器被清除为 0，并且不产生中断请求。

UF0E0R 寄存器保持的数据必须通过 FW 读取，读取的数量达到 UF0E0L 寄存器数据读取的数量值。使用 UF0EPS0 寄存器的 EP0R 位来检查所有数据已经被读取 (当所有数据已经被读取时，EP0R = 0)。如果 UF0E0L 寄存器的值是 0，UF0E0N 寄存器的 EP0NKR 位被清除为 0，并且 UF0E0R 寄存器准备好接收。当接收到下一个 SETUP 令牌，UF0E0R 寄存器被清除。

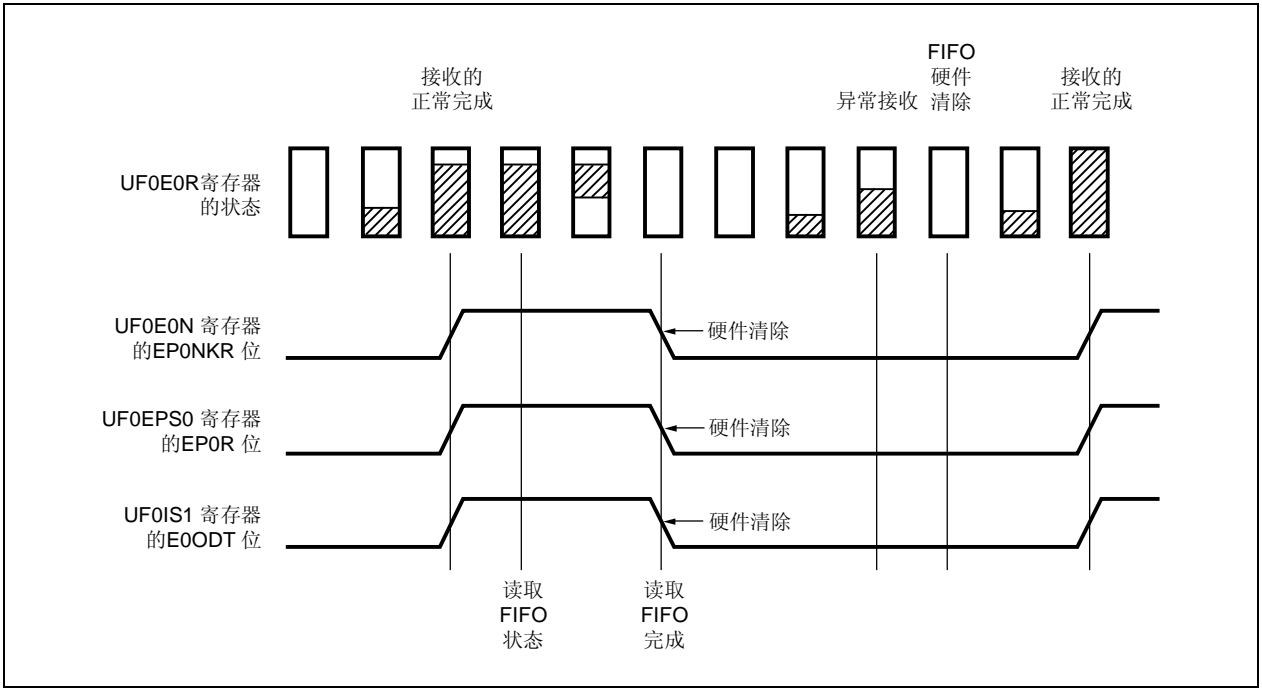
注意事项 读取所有保存的数据。清除 FIFO 来丢弃一些数据。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0R	E0R7	E0R6	E0R5	E0R4	E0R3	E0R2	E0R1	E0R0	00200100H	不确定

位的位置	位名称	功能
7 至 0	E0R7 至 E0R0	这些位存储到/从端点 0 控制传输的数据阶段中从主机发送的 OUT 数据

UF0E0R 寄存器的操作如下所示。

图 20-4. UF0E0R 寄存器的操作



(2) UF0 EP0 长度寄存器 (UF0E0L)

UF0E0L 寄存器用于存储保持在 UF0E0R 寄存器中的数据长度。

该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当接收数据时，UF0E0L 寄存器总是更新已接收到数据的长度。如果最后传输是异常接收，UF0E0L 寄存器被清除为 0，并且不产生中断请求。只有当接收正常，读取 UF0E0L 寄存器的数据长度，并 FW 按照设定数量从 UF0E0R 寄存器读取足够的数，才会产生中断请求。每次 UF0E0R 寄存器被读取后，UF0E0L 寄存器的值递减。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0L	E0L7	E0L6	E0L5	E0L4	E0L3	E0L2	E0L1	E0L0	00200102H	不确定
位的位置	位名称		功能							
7 至 0	E0L7 至 E0L0		这些位存储保持在 UF0E0R 寄存器中的数据长度。							

(3) UF0E0 建立寄存器 (UF0E0ST)

UF0E0ST 寄存器用于保持从主机发送的 SETUP 数据。

该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当接收到 SETUP 事务时，UF0E0ST 寄存器总是写入数据。当已经正确接收 SETUP 事务后，硬件对 UF0IS1 寄存器的 PROT 位置位。在由 FW 处理请求的情况下，对 UF0IS1 寄存器的 CPUDEC 位置位，然后发出中断请求 (INTUSBF0)。在由 FW 处理请求的情况下，确保按照 8 位宽度读取请求。如果不是按照 8 位宽度读取，后续请求不能被正确译码。即使接收到总线复位时，UF0E0ST 寄存器的读取计数器也不会被清除。不管是否接收到总线复位，总是按照 8 位宽度读取这个计数器。

因为 UF0E0ST 寄存器总是允许写入，即使正在读取寄存器数据时接收到一个 SETUP 事务，硬件也会覆盖该寄存器的数据。即使 SETUP 事务未能被正确接收，也不会产生 CPUDEC 中断请求和 Protect 中断请求，但是之前的数据被丢弃。但是，如果接收到的 SETUP 令牌少于 8 字节，接收的 SETUP 令牌被丢弃，并且之前接收的 SETUP 数据被保留。如果当控制传输执行了一次，接收到多次 SETUP 令牌，在以下条件下，确保检查 UF0IS1 寄存器的 PROT 位。如果 PROT 位 = 1，再次读取 UF0E0ST 寄存器，因为 SETUP 事务已经被多次接收。

<1> 如果由 FW 对请求进行译码，且 UF0E0R 寄存器被读取或者 UF0E0W 寄存器被写入

<2> 为该请求准备一个 STALL 响应时，译码结果对其不响应。

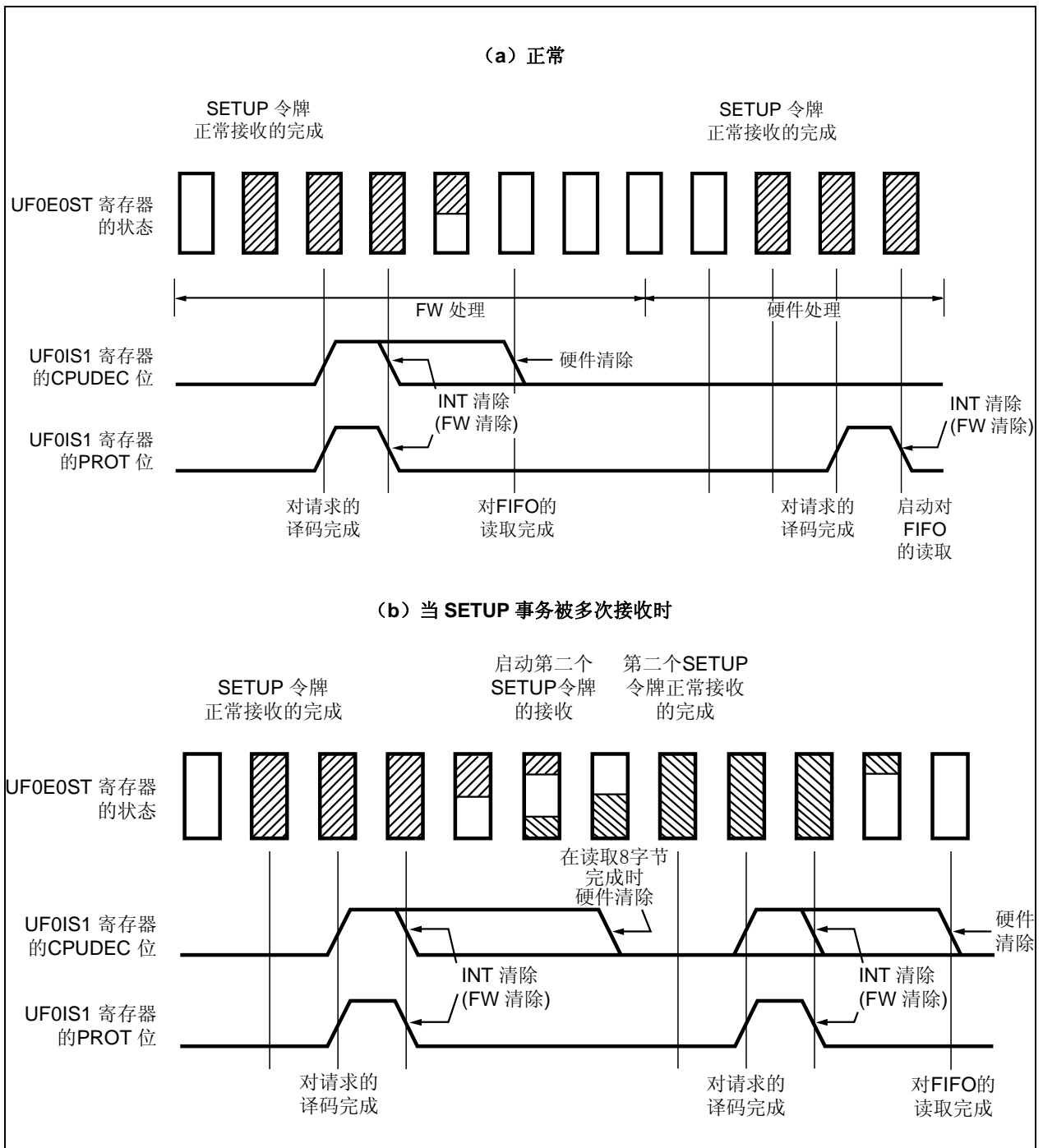
注意事项 确保读取所有保存的数据。UF0E0ST 寄存器总是由 SETUP 事务中的请求来更新。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0ST	E0S7	E0S6	E0S5	E0S4	E0S3	E0S2	E0S1	E0S0	00200104H	00H

位的位置	位名称	功能
7 至 0	E0S7 至 E0S0	这些位保持从主机发送的 SETUP 数据。

UF0E0ST 寄存器的操作如下所示。

图 20-5. UF0E0ST 寄存器的操作



(4) UF0 EP0 写入寄存器 (UF0E0W)

UF0E0W 寄存器是一个 64 字节 FIFO，用于保存到端点 0 的数据阶段中发送到主机的 IN 数据（传递到 SIE）。

该寄存器是只能写入，按字节进行写入。如果读取该寄存器，读到的值是 00H。

只有当 UF0E0N 寄存器的 EP0NKW 位被置为 1 时（NAK 没有被发送时），硬件和 IN 令牌同步，发送数据到 USB 总线。当数据已发送且主机正确接收该数据时，UF0E0N 寄存器的 EP0NKW 位由硬件自动清除为 0。当数据已写入 UF0E0W 寄存器且 UF0DEND 寄存器的 E0DED 位被置为 1（UF0EPS0 寄存器的 EP0W 位 = 1（数据存在））时，发送一个短包。当 UF0E0W 寄存器已清除且 UF0DEND 寄存器的 E0DED 位被置为 1（UF0EPS0 寄存器的 EP0W 位 = 1（数据存在））时，发送一个空包。

发送没尚未完成期间，接收到下一个 SETUP 令牌时，UF0E0W 寄存器被清除为 0。在数据阶段中 ACK 没有被正确接收时，如果控制传输（读）的阶段更改为状态阶段，UF0E0W 寄存器被自动清除为 0。同时，如果 UF0E0N 寄存器的 EP0NKW 位为 1，它也被清除为 0。

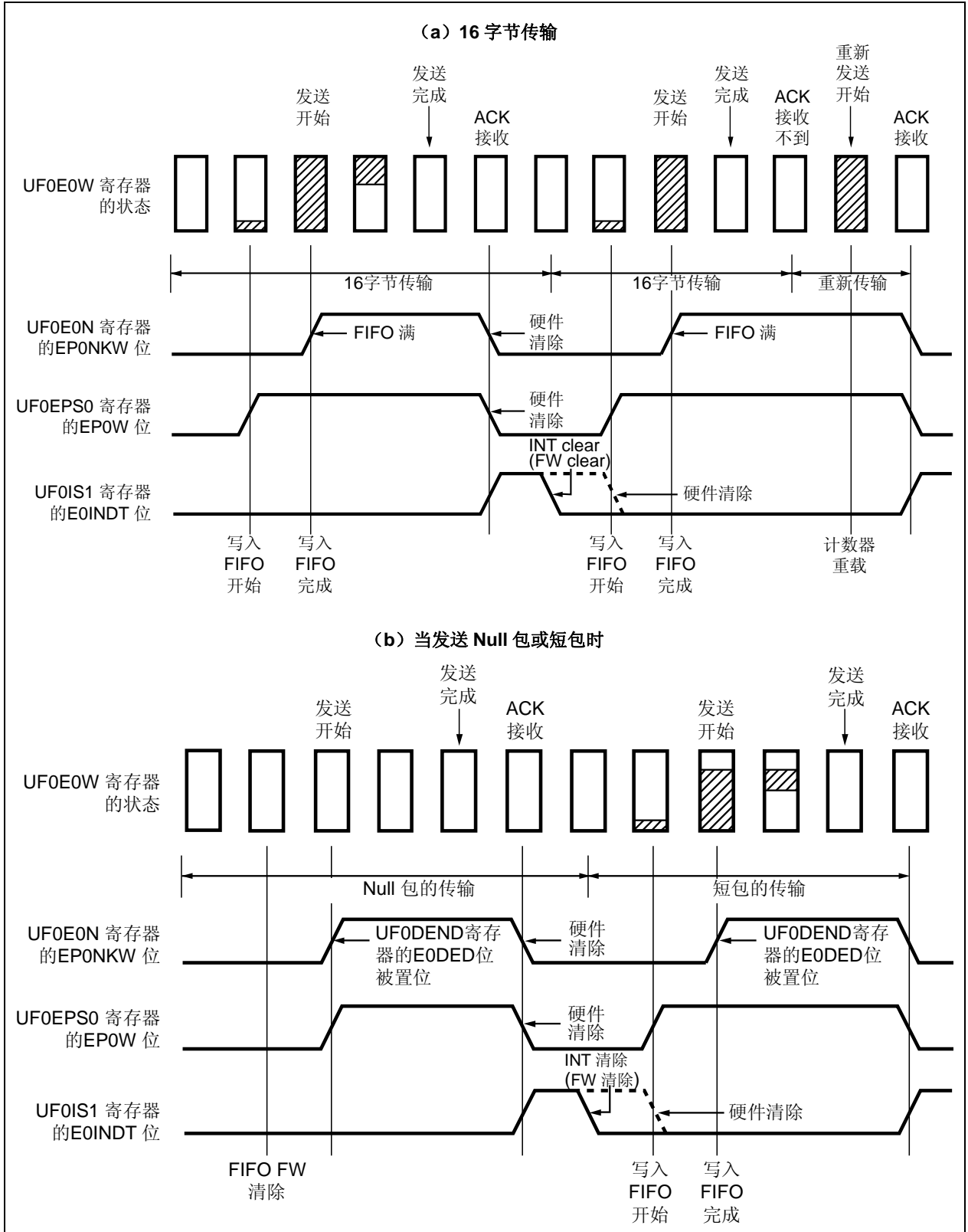
没有数据存于 UF0E0W 寄存器中时，如果被读取，读到的值是 00H。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0W	E0W7	E0W6	E0W5	E0W4	E0W3	E0W2	E0W1	E0W0	00200106H	不确定

位的位置	位名称	功能
7 至 0	E0W7 至 E0W0	这些位保存到端点 0 的数据阶段中发送到主机的 IN 数据。

UF0E0W 寄存器的操作如下所示。

图 20-6. UF0E0W 寄存器的操作



(5) UF0 批量-out 1 寄存器 (UF0BO1)

UF0BO1 寄存器是一个 64 字节× 2 的 FIFO，用于保存进出端点 2 的数据。该寄存器由 64 字节 FIFO 的两个 bank 组成，每个 bank 执行切换操作并在 SIE 和 CPU 端反复连接总线。当数据位于 SIE 端的 FIFO 中且 CPU 端的 FIFO 中没有数据（计数器值 = 0）时，切换操作发生。

该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当硬件从主机接收到端点 2 的数据时，自动将数据传送到 UF0BO1 寄存器。当该寄存器正确接收数据时，发生 FIFO 切换操作。因此，UF0IS3 寄存器的 BKO1DT 位被置为 1，接收到的数据的数量被保持在 UF0BO1L 寄存器中，并且向 CPU 发出中断请求或 DMA 请求。可以通过 UF0IDR 寄存器的 DQBO1MS 位来选择是发出中断请求或发出 DMA 请求。

通过 FW 读取 UF0BO1 寄存器中保持的数据，读取的数据的数量由 UF0BO1L 寄存器决定。当正确接收的数据被连接到 SIE 端的 FIFO 保持，且 UF0BO1L 寄存器的值达到 0 时，发生 FIFO 的切换操作，并且 UF0EN 寄存器的 BKO1NK 位被自动清除为 0。如果读取的数据的数量多于 UF0BO1L 寄存器的值，且满足 FIFO 切换条件，则发生 FIFO 的切换操作。因此，下一个包可能被错误读取。注意，如果不满足切换条件，第一个数据需要重复读取。

数据被保持在连接到 CPU 端的 FIFO 时，如果接收到溢出数据，端点 2 停顿，并且 CPU 端的 FIFO 被清除。

没有数据存于 UF0BO1 寄存器中时，如果被读取，读到的值不确定。

注意事项 确保读取所有保存在该寄存器中的数据。

	7	6	5	4	3	2	1	0	地址	复位后
UF0BO1	BKO17	BKO16	BKO15	BKO14	BKO13	BKO12	BKO11	BKO10	00200108H	不确定

位的位置	位名称	功能
7 至 0	BKO17 至 BKO10	这些位保存进出端点 2 的数据。

UF0BO1 寄存器的操作如下所示。

图 20-7. UF0B01 寄存器的操作 (1/2)

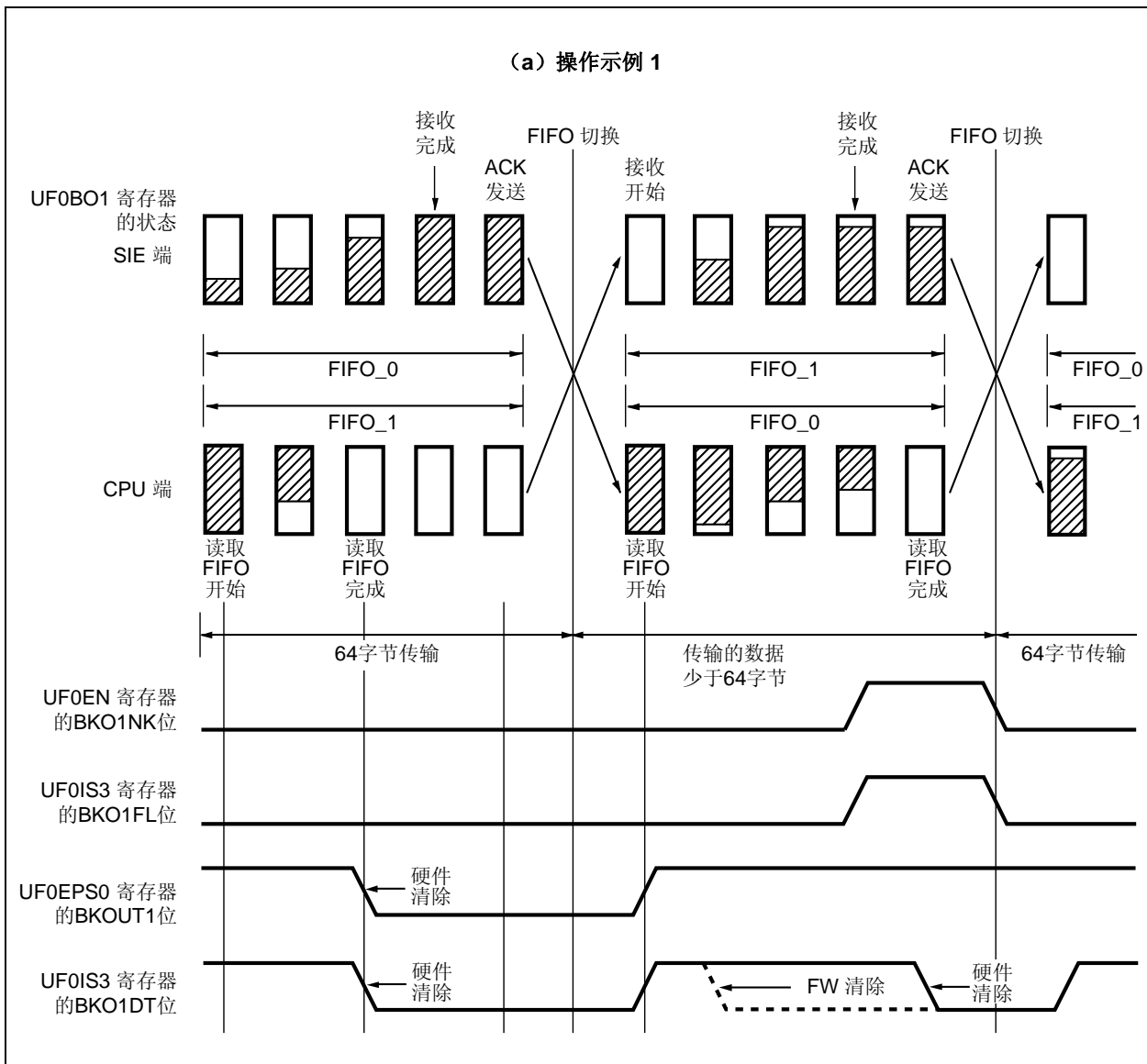
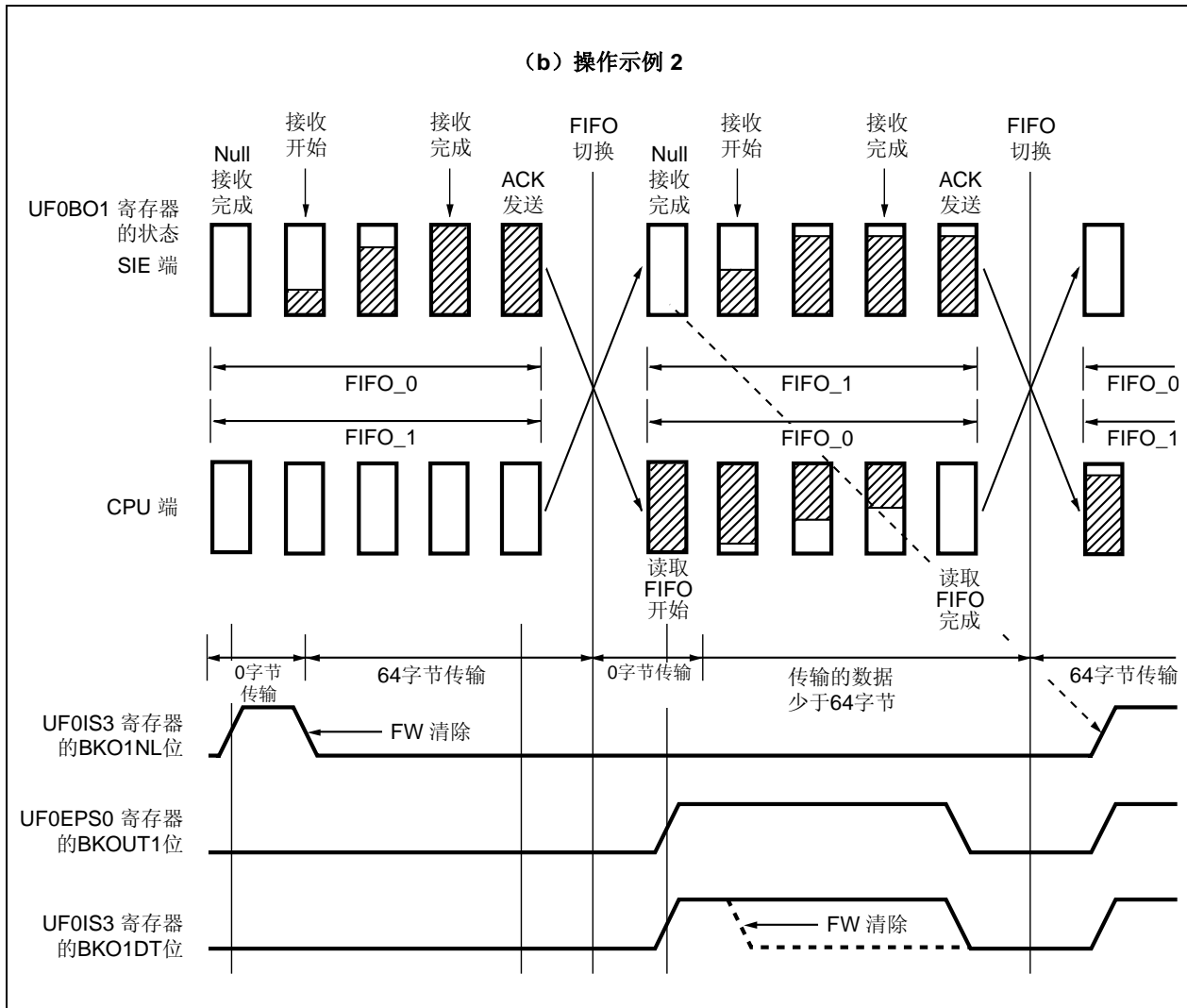


图 20-7. UF0BO1 寄存器的操作 (2/2)



(6) UF0 批量-out 1 长度寄存器 (UF0BO1L)

UF0BO1L 寄存器用于存储保持在 UF0BO1 寄存器中的数据的长度。

该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当接收数据时，UF0BO1L 寄存器总是更新已接收到的数据的长度。如果最后传输是异常接收，UF0BO1L 寄存器被清除为 0 且不产生中断请求。只有当接收正常，才会产生中断请求，且通过 FW 读取 UF0BO1 寄存器中保持的数据，读取的数据的数量由 UF0BO1L 寄存器决定。每次 UF0BO1 寄存器被读取后，UF0BO1L 寄存器的值递减。

	7	6	5	4	3	2	1	0	地址	复位后
UF0BO1L	BKO1L7	BKO1L6	BKO1L5	BKO1L4	BKO1L3	BKO1L2	BKO1L1	BKO1L0	0020010AH	00H

位的位置	位名称	功能
7 至 0	BKO1L7 至 BKO1L0	这些位存储保持在 UF0BO1 寄存器中的数据的长度。

(7) UF0 批量-out 2 寄存器 (UF0BO2)

UF0BO2 寄存器是一个 64 字节× 2 的 FIFO，用于保存进出端点 4 的数据。该寄存器由 64 字节 FIFO 的两个 bank 组成，每个 bank 执行切换操作并在 SIE 和 CPU 端反复连接总线。当数据位于 SIE 端的 FIFO 中且 CPU 端的 FIFO 中没有数据（计数器值 = 0）时，切换操作发生。

该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当硬件从主机接收到端点 4 的数据时，自动将数据传送到 UF0BO2 寄存器。当该寄存器正确接收数据时，发生 FIFO 切换操作。因此，UF0IS3 寄存器的 BKO2DT 位被置为 1，接收到的数据的数量被保持在 UF0BO2L 寄存器中，并且向 CPU 发出中断请求或 DMA 请求。可以通过 UF0IDR 寄存器的 DQBO2MS 位来选择是发出中断请求或发出 DMA 请求。

通过 FW 读取 UF0BO2 寄存器中保持的数据，读取的数据的数量由 UF0BO2L 寄存器决定。当正确接收的数据被连接到 SIE 端的 FIFO 保持，且 UF0BO2L 寄存器的值达到 0 时，发生 FIFO 的切换操作，并且 UF0EN 寄存器的 BKO2NK 位被自动清除为 0。如果读取的数据的数量多于 UF0BO2L 寄存器的值，且满足 FIFO 切换条件，则发生 FIFO 的切换操作。因此，下一个包可能被错误读取。注意，如果不满足切换条件，第一个数据需要重复读取。

数据被保持在连接到 CPU 端的 FIFO 时，如果接收到溢出数据，端点 4 停顿，并且 CPU 端的 FIFO 被清除。

没有数据存于 UF0BO2 寄存器中时，如果被读取，读到的值不确定。

注意事项 确保读取所有保存在该寄存器中的数据。

	7	6	5	4	3	2	1	0	地址	复位后
UF0BO2	BKO27	BKO26	BKO25	BKO24	BKO23	BKO22	BKO21	BKO20	0020010CH	不确定

位的位置	位名称	功能
7 至 0	BKO27 至 BKO20	这些位保存进出端点 4 的数据。

UF0BO2 寄存器的操作如下所示。

图 20-8. UF0B02 寄存器的操作 (1/2)

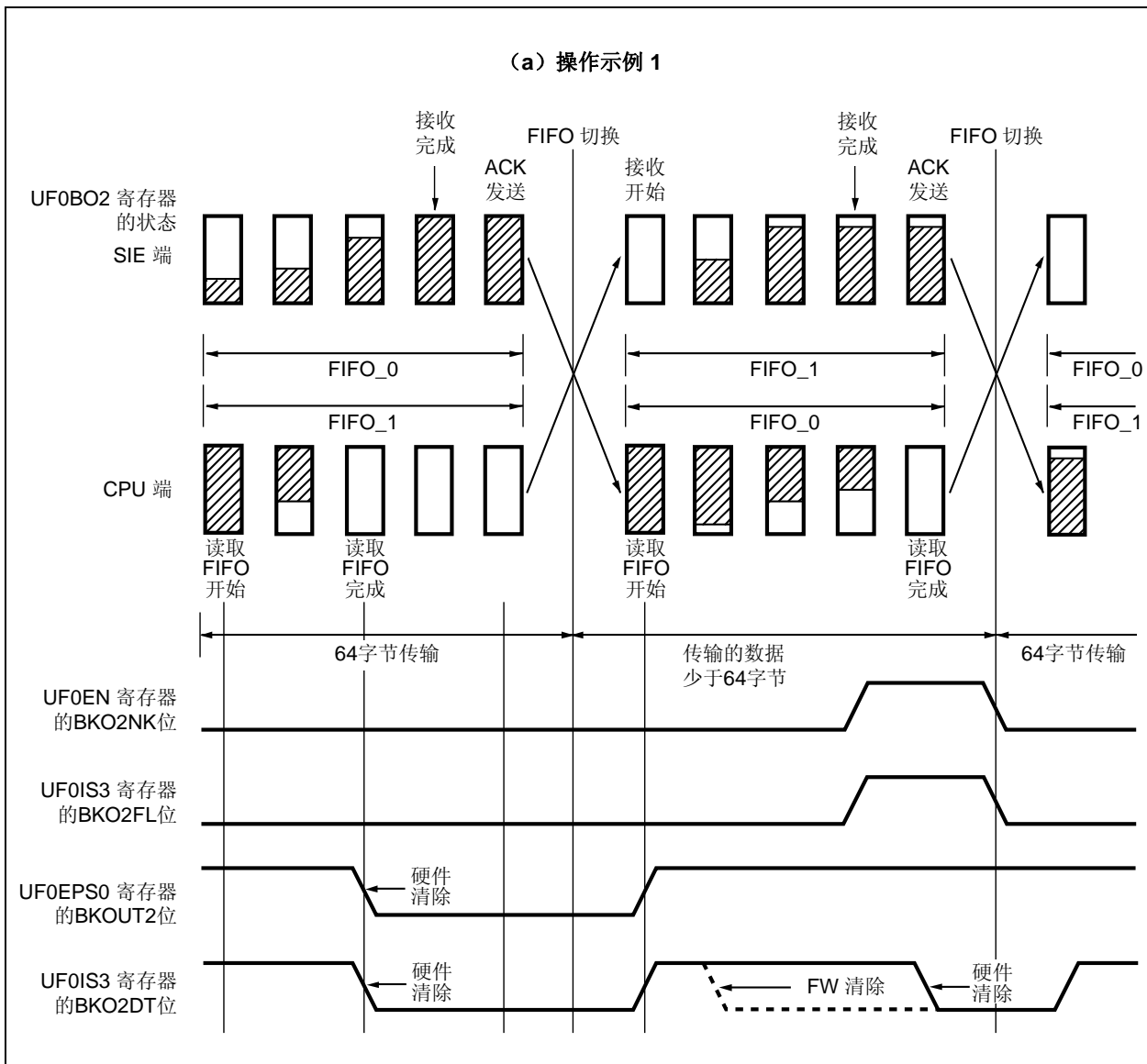
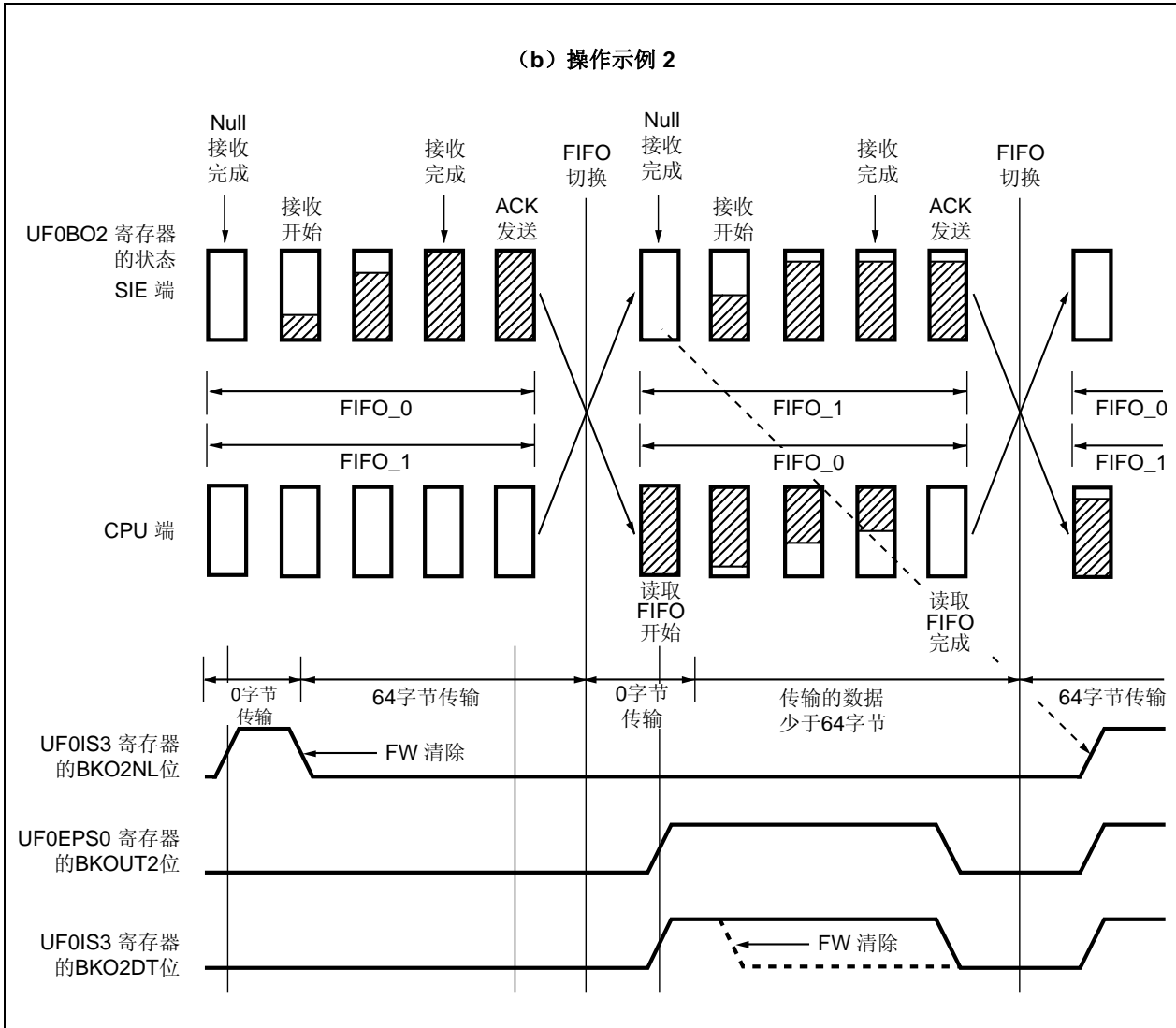


图 20-8. UF0BO2 寄存器的操作 (2/2)



(8) UF0BO2L 批量-out 2 长度寄存器 (UF0BO2L)

UF0BO2L 寄存器用于存储保持在 UF0BO2 寄存器中的数据的长度。

该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当接收数据时，UF0BO2L 寄存器总是更新已接收到的数据的长度。如果最后传输是异常接收，UF0BO2L 寄存器被清除为 0 且不产生中断请求。只有当接收正常，才会产生中断请求，且通过 FW 读取 UF0BO2 寄存器中保持的数据，读取的数据的数量由 UF0BO2L 寄存器决定。每次 UF0BO2 寄存器被读取后，UF0BO2L 寄存器的值递减。

	7	6	5	4	3	2	1	0	地址	复位后
UF0BO2L	BKO2L7	BKO2L6	BKO2L5	BKO2L4	BKO2L3	BKO2L2	BKO2L1	BKO2L0	0020010EH	00H

位的位置	位名称	功能
7 至 0	BKO2L7 至 BKO2L0	这些位存储保持在 UF0BO2 寄存器中的数据的长度。

(9) UF0BI1 批量-in 1 寄存器 (UF0BI1)

UF0BI1 寄存器是一个 64 字节×2 的 FIFO，用于保存进出端点 1 的数据。该寄存器由 64 字节 FIFO 的两个 bank 组成，每个 bank 执行切换操作并在 SIE 和 CPU 端反复连接总线。当 SIE 端的 FIFO 中没有数据（计数器值 = 0）且 CPU 端的 FIFO 被正确写入（FIFO 满或 BKI1DED 位 = 1）时，发生切换操作。

该寄存器是只能写入，按字节进行写入。如果读取该寄存器，读到的值是 00H。

只有当 UF0E0N 寄存器的 BKI1NK 位被置为 1 时（NAK 没有被发送时），硬件和对端点 1 的 IN 令牌同步，发送数据到 USB 总线。被写入或读取的数据地址由硬件管理。因此，只需按顺序将数据写入到 UF0BI1 寄存器，FW 就可以将数据发送到主机。当数据已写入 UF0BI1 寄存器且 UF0DEND 寄存器的 BKI1DED 位被置为 1（UF0EPS0 寄存器的 BKIN1 位 = 1（数据存在））时，发送一个短包。当 UF0BI1 寄存器已清除且 UF0DEND 寄存器的 BKI1DED 位被置为 1（UF0EPS0 寄存器的 BKIN1 位 = 1（数据存在））时，发送一个空包。当数据被正确发送时，发生 FIFO 切换操作。UF0IS2 寄存器的 BKI1DT 位被置为 1，并且向 CPU 发送中断请求。可以通过 UF0IDR 寄存器的 DQB11MS 位来选择是中断请求或 DMA 请求。

	7	6	5	4	3	2	1	0	地址	复位后
UF0BI1	BKI17	BKI16	BKI15	BKI14	BKI13	BKI12	BKI11	BKI10	00200110H	不确定

位的位置	位名称	功能
7 至 0	BKI17 至 BKI10	这些位保存进出端点 1 的数据。

UF0BI1 寄存器的操作如下所示。

图 20-9. UF0B11 寄存器的操作 (1/3)

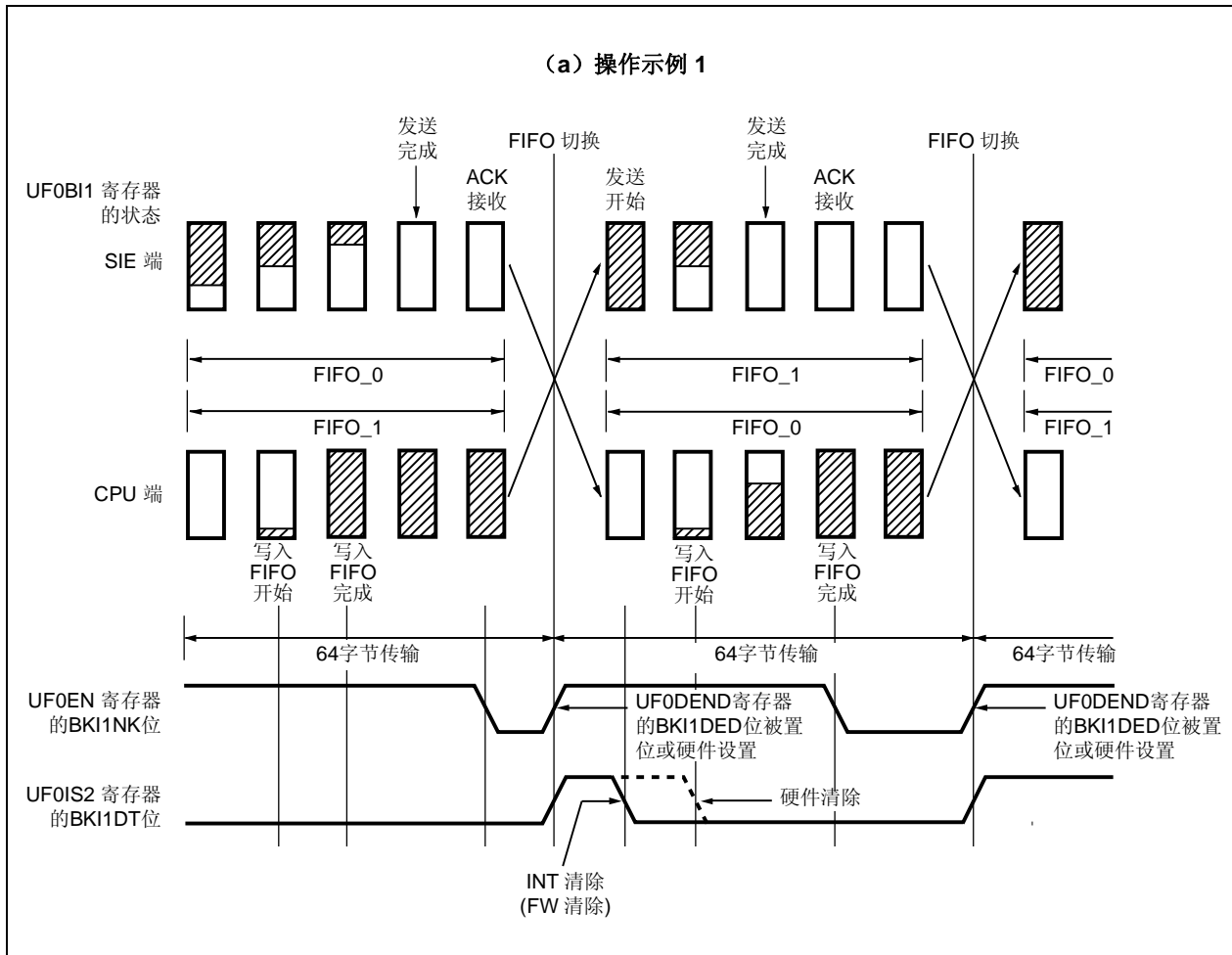


图 20-9. UF0B11 寄存器的操作 (2/3)

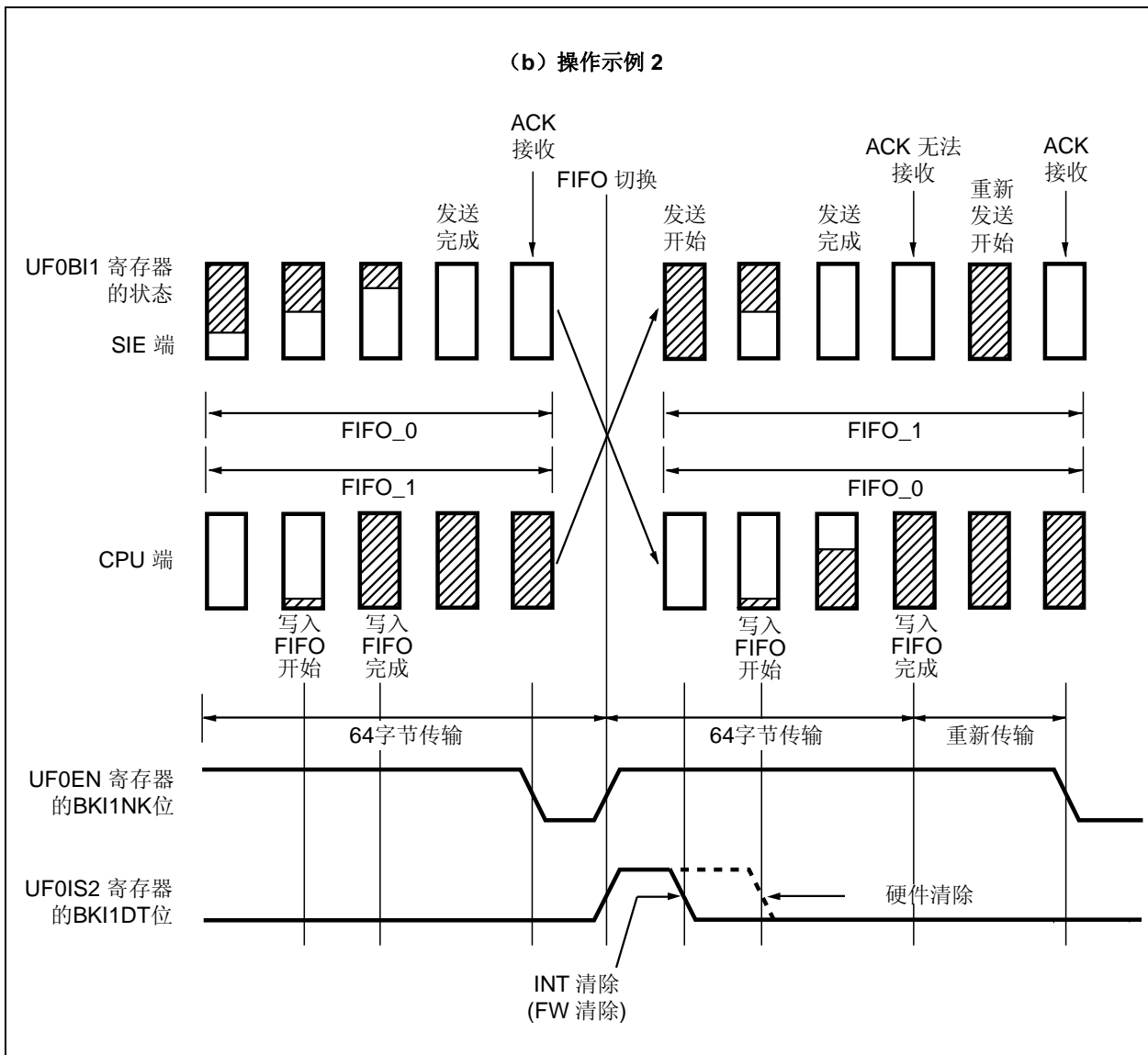
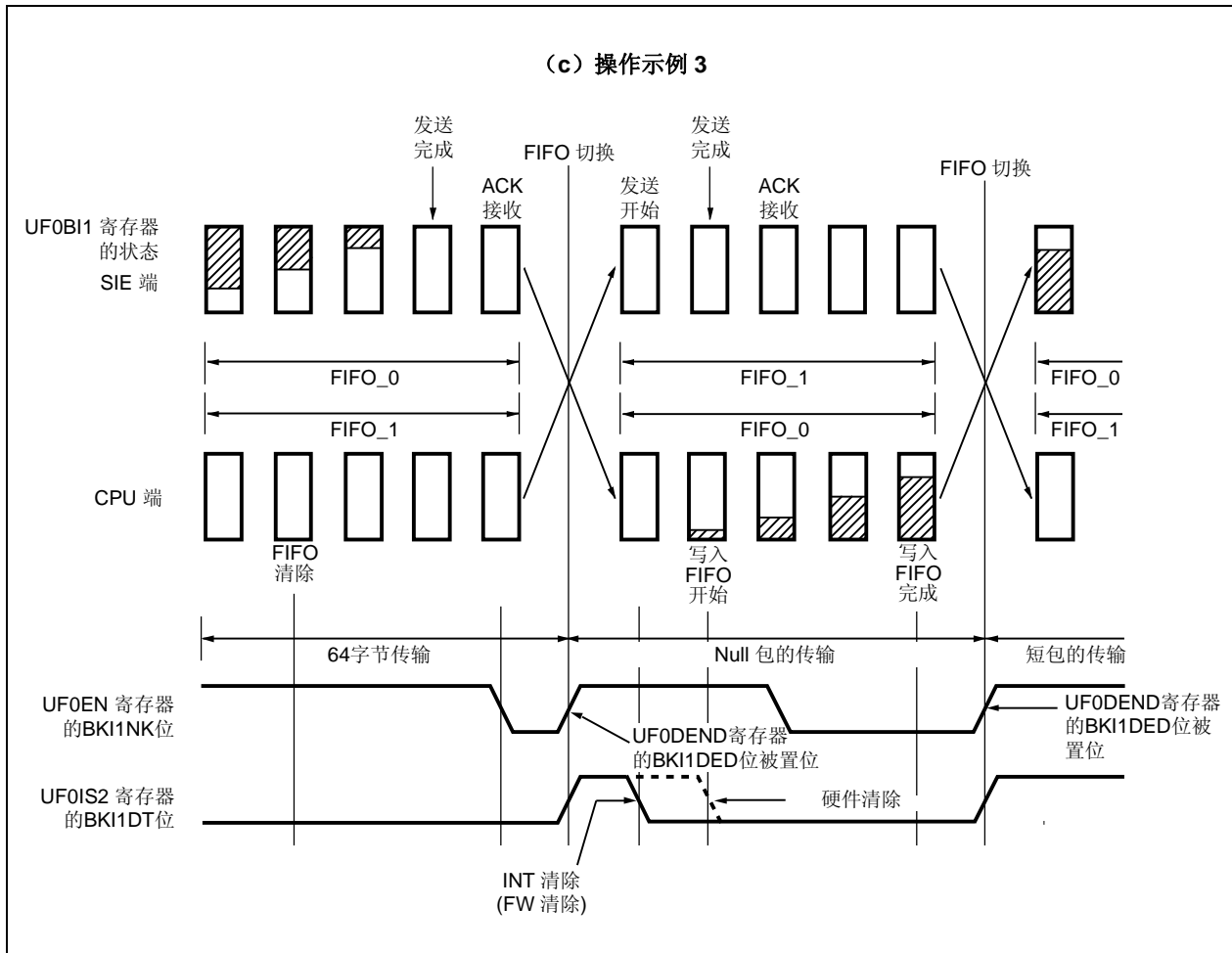


图 20-9. UF0B11 寄存器的操作 (3/3)



(10) UF0 批量-in 2 寄存器 (UF0BI2)

UF0BI2 寄存器是一个 64 字节× 2 的 FIFO，用于保存进出端点 3 的数据。该寄存器由 64 字节 FIFO 的两个 bank 组成，每个 bank 执行切换操作并在 SIE 和 CPU 端反复连接总线。当 SIE 端的 FIFO 中没有数据（计数器值 = 0）且 CPU 端的 FIFO 被正确写入（FIFO 满或 BKI3DED 位 = 1）时，发生切换操作。

该寄存器是只能写入，按字节进行写入。如果读取该寄存器，读到的值是 00H。

只有当 UF0E0N 寄存器的 BKI2NK 位被置为 1 时（NAK 没有被发送时），硬件和对端点 3 的 IN 令牌同步，发送数据到 USB 总线。被写入或读取的数据地址由硬件管理。因此，只需按顺序将数据写入到 UF0BI2 寄存器，FW 就可以将数据发送到主机。当数据已写入 UF0BI2 寄存器且 UF0DEND 寄存器的 BKI2DED 位被置为 1（UF0EPS0 寄存器的 BKIN2 位 = 1（数据存在））时，发送一个短包。当 UF0BI2 寄存器已清除且 UF0DEND 寄存器的 BKI2DED 位被置为 1（UF0EPS0 寄存器的 BKIN2 位 = 1（数据存在））时，发送一个空包。当数据被正确发送时，发生 FIFO 切换操作。UF0IS2 寄存器的 BKI2DT 位被置为 1，并且向 CPU 发送中断请求。可以通过 UF0IDR 寄存器的 DQBI2MS 位来选择是中断请求或 DMA 请求。

	7	6	5	4	3	2	1	0	地址	复位后
UF0BI2	BKI27	BKI26	BKI25	BKI24	BKI23	BKI22	BKI21	BKI20	00200112H	不确定

位的位置	位名称	功能
7 至 0	BKI27 至 BKI20	这些位保存进出端点 3 的数据。

UF0BI2 寄存器的操作如下所示。

图 20-10. UF0BI2 寄存器的操作 (1/3)

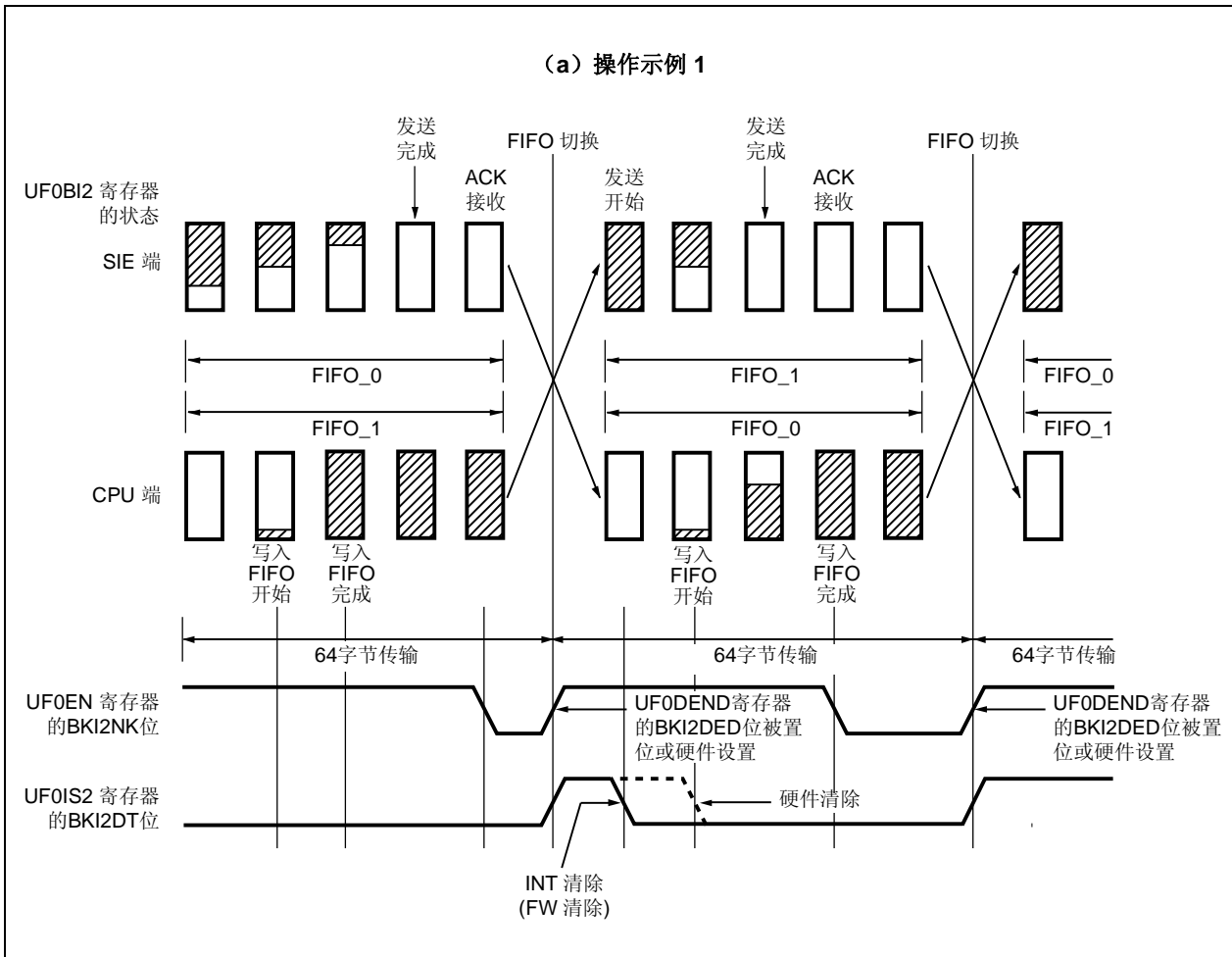


图 20-10. UF0B12 寄存器的操作 (2/3)

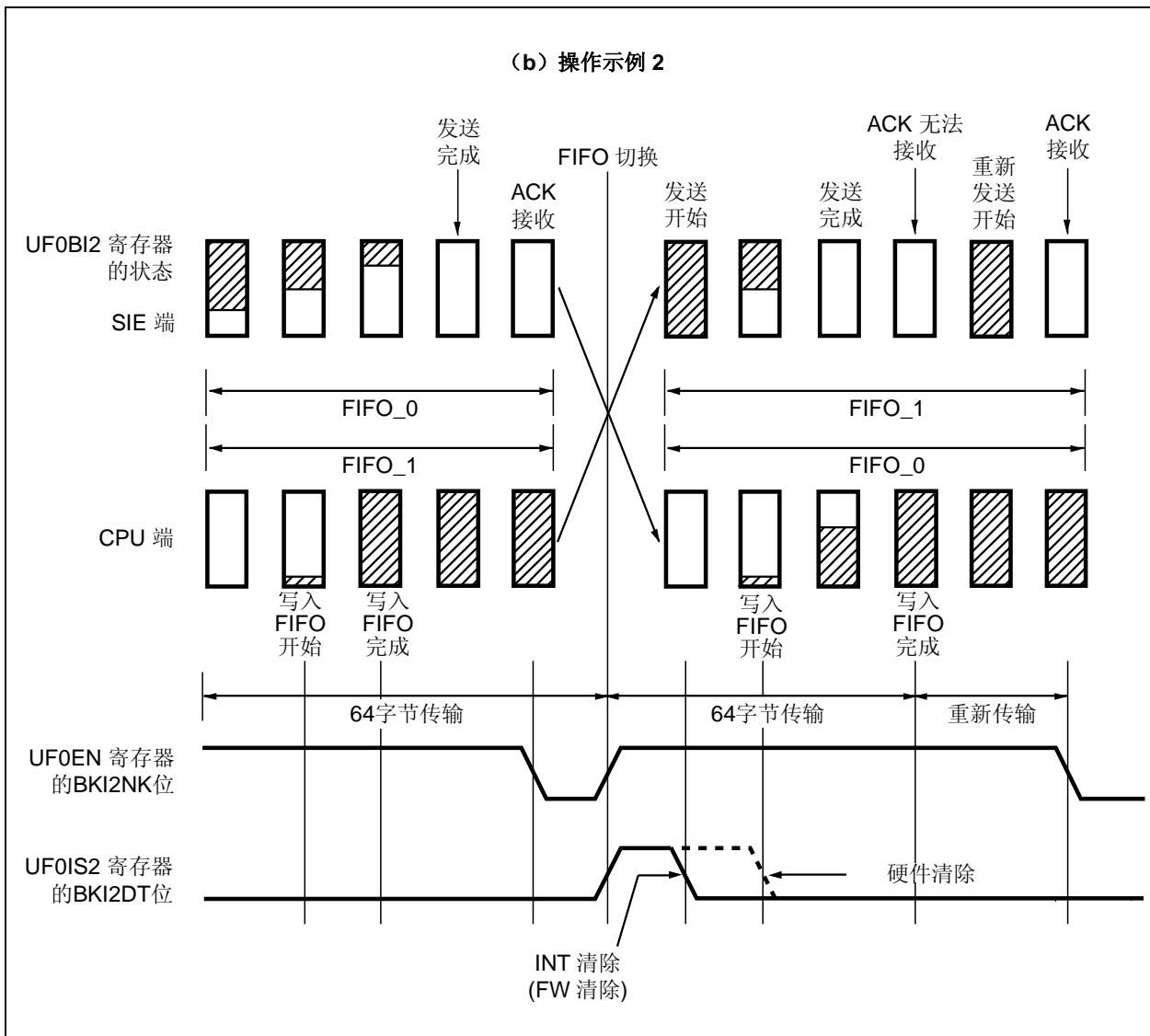
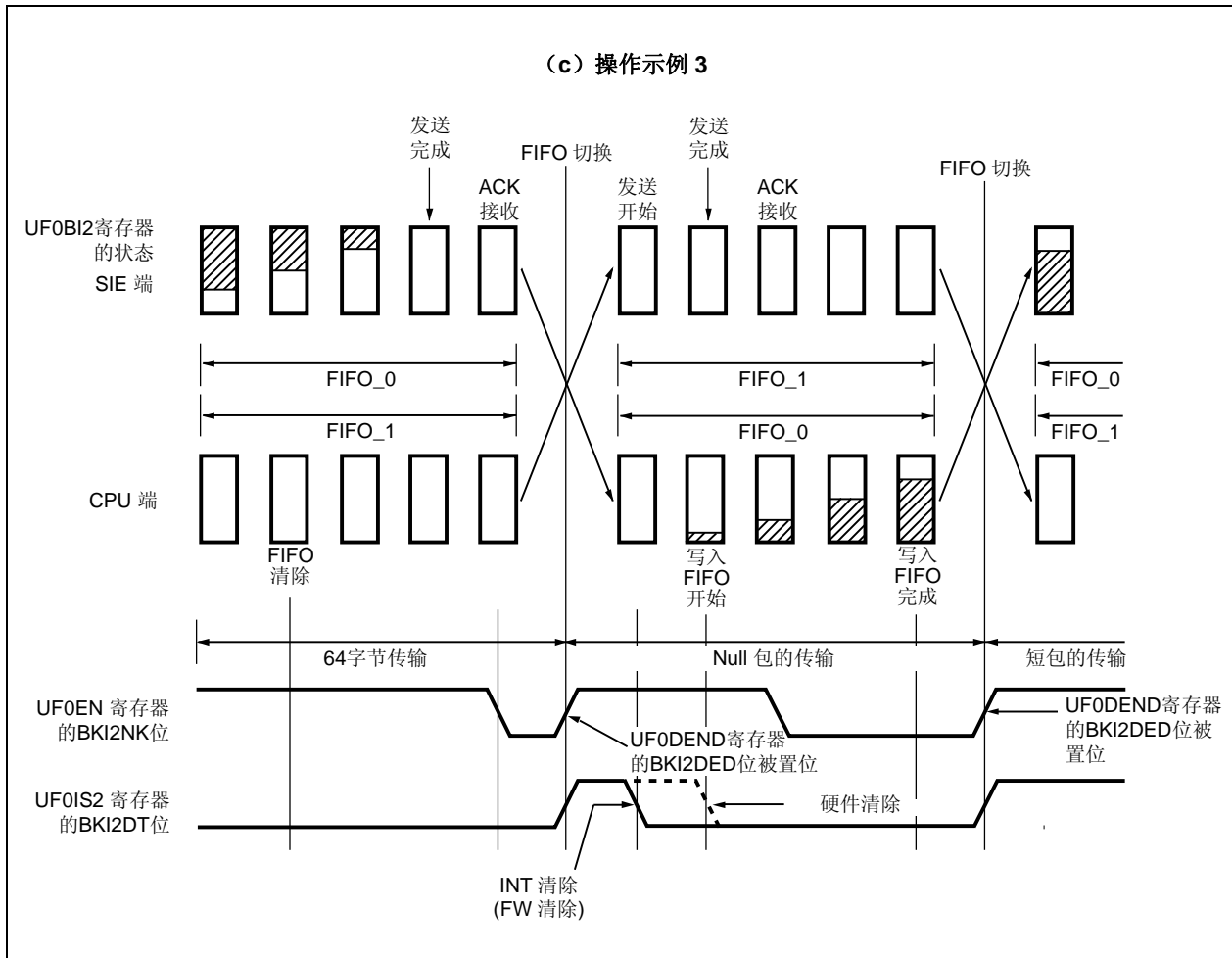


图 20-10. UF0BI2 寄存器的操作 (3/3)



(11) UF0 中断 1 寄存器 (UF0INT1)

UF0INT1 寄存器是一个 8-字节 FIFO，保存进出端点 7 的数据（传递到 SIE）。

该寄存器是只能写入，按字节进行写入。如果读取该寄存器，读到的值是 00H。

只有当 UF0E0N 寄存器的 IT1NK 位被置为 1 时（NAK 没有被发送时），硬件和对端点 7 的 IN 令牌同步，发送数据到 USB 总线。当数据已发送且主机正确接收该数据时，UF0EN 寄存器的 IT1NK 位由硬件自动清除为 0。

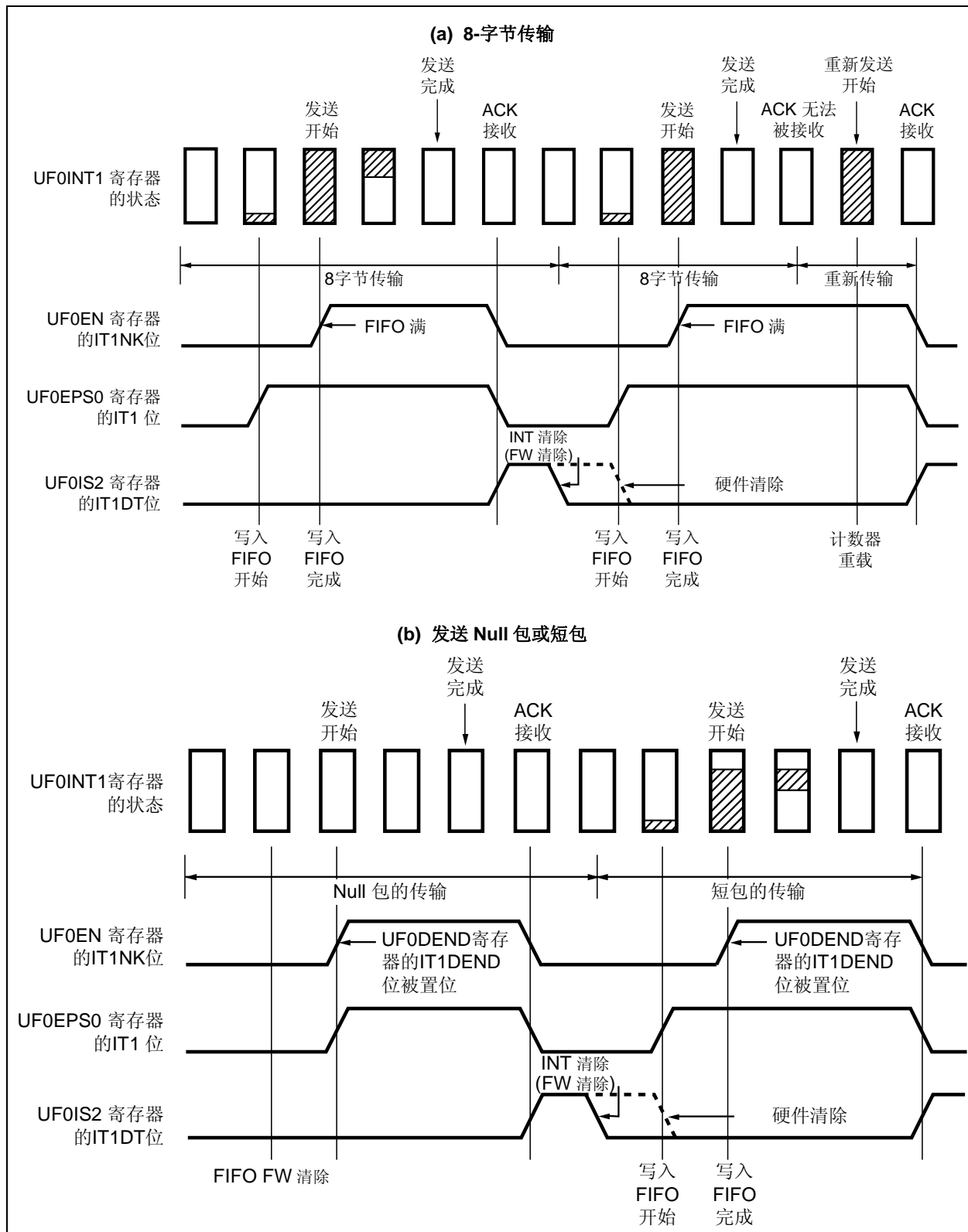
当数据已写入 UF0INT1 寄存器且 UF0DEND 寄存器的 IT1DEND 位被置为 1（UF0EPS0 寄存器的 IT1 位 = 1（数据存在））时，发送一个短包。当 UF0INT1 寄存器已清除且 UF0DEND 寄存器的 IT1DEND 位被置为 1（UF0EPS0 寄存器的 IT1 位 = 1（数据存在））时，发送一个空包。

		7	6	5	4	3	2	1	0	地址	复位后
UF0INT1		IT17	IT16	IT15	IT14	IT13	IT12	IT11	IT10	00200114H	不确定

位的位置	位名称	功能
7 至 0	IT17 至 IT10	这些位保存进出端点 7 的数据。

UF0INT1 寄存器的操作如下所示。

图 20-11. UF0INT1 寄存器的操作



20.6.5 EPC请求数据寄存器

(1) UF0 设备状态寄存器 L (UF0DSTL)

该寄存器保存响应 GET_STATUS 设备请求的返回值。

该寄存器可以按字节进行读取或写入。

当接收到 GET_STATUS 设备请求时，硬件自动将该寄存器的内容发送给主机。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EPONKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0DSTL	0	0	0	0	0	0	RMWK	SFPW	00200144H	00H

位的位置	位名称	功能
1	RMWK	该位指定是否使用设备的远程唤醒功能。 1: 允许 0: 禁止 如果设备支持远程唤醒功能，当接收到 SET_FEATURE 设备请求时，该位由硬件设置为 1，并且当接收到 CLEAR_FEATURE 设备请求时，由硬件清除为 0。如果设备不支持远程唤醒功能，确保主机不会发出 SET_FEATURE 设备请求。
0	SFPW	该位表明设备是自供电还是总线供电。 1: 自供电 0: 总线供电

(2) UF0 EP0 状态寄存器 L (UF0E0SL)

该寄存器保存响应 GET_STATUS 端点 0 请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

如果 USBF 发生一个错误，E0HALT 位被 FW 设置为 1。当正在接收对端点 0 的 USB 端访问时，对该寄存器的写入访问被忽略。

如果之前的控制传输是对于 SET_FEATURE 端点 0、CLEAR_FEATURE 端点 0、GET_STATUS 端点 0 请求或由 FW 处理的请求，当 E0HALT 位被 FW 设置为 1 时，它不会立即被反映出来，直到接收到下一个 SETUP 令牌。

当接收到 GET_STATUS 端点 0 请求时，硬件自动将该寄存器的内容发送给主机。如果端点 0 已经停顿，UF0E0W 和 UF0E0R 寄存器被清除，且 UF0E0N 寄存器的 EP0NKW 和 EP0NKR 位被清除为 0。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0SL	0	0	0	0	0	0	0	E0HALT	0020014CH	00H

位的位置	位名称	功能
0	E0HALT	该位表明端点 0 的状态。 1: 已停顿 (Stalled) 0: 未停顿 当接收到 SET_FEATURE 端点 0 请求时，该位被硬件设置为 1，并且当接收到 CLEAR_FEATURE 端点 0 请求时，该位被硬件清除为 0。DATA PID 被初始化为 DATA0。

(3) UF0 EP1 状态寄存器 L (UF0E1SL)

该寄存器保存响应 GET_STATUS 端点 1 请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

如果端点 1 发生一个错误，E1HALT 位被 FW 设置为 1。当正在接收对端点 1 的 USB 端访问时，对该寄存器的写入访问被忽略。

当接收到 GET_STATUS 端点 1 请求时，硬件自动将该寄存器的内容发送给主机。如果端点 1 已经停顿，UF0B11 寄存器被清除，且 BK11NK 位被清除为 0。

当传输到端点 1 而不是控制传输时，被执行时写入该寄存器总是被屏蔽，确保检查该寄存器来查看数据是否被正确写入。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E1SL	0	0	0	0	0	0	0	E1HALT	00200150H	00H

位的位置	位名称	功能
0	E1HALT	该位表明端点 1 的状态。 1: 已停顿 (Stalled) 0: 未停顿 当接收到 SET_FEATURE 端点 1 请求时，该位被硬件设置为 1，并且当正确接收到 CLEAR_FEATURE 端点 1 请求，SET_CONFIGURATION 请求，或正确接收到对端点 1 链接的接口的 SET_INTERFACE 请求时，该位被硬件清除为 0。DATA PID 被初始化为 DATA0。

(4) UF0 EP2 状态寄存器 L (UF0E2SL)

该寄存器保存响应 GET_STATUS 端点 2 请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

如果端点 2 发生一个错误，E2HALT 位被 FW 设置为 1。当正在接收对端点 2 的 USB 端访问时，对该寄存器的写入访问被忽略。

当接收到 GET_STATUS 端点 2 请求时，硬件自动将该寄存器的内容发送给主机。如果端点 2 已经停顿，UF0BO1 寄存器被清除，且 BKO1NK 位被清除为 0。

当传输到端点 2 而不是控制传输时，被执行时写入该寄存器总是被屏蔽，确保检查该寄存器来查看数据是否被正确写入。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E2SL	0	0	0	0	0	0	0	E2HALT	00200154H	00H

位的位置	位名称	功能
0	E2HALT	该位表明端点 2 的状态。 1: 已停顿 (Stalled) 0: 未停顿 当接收到 SET_FEATURE 端点 2 请求时，该位被硬件设置为 1，并且当正确接收到 CLEAR_FEATURE 端点 2 请求，SET_CONFIGURATION 请求，或正确接收到对端点 2 链接的接口的 SET_INTERFACE 请求时，该位被硬件清除为 0。DATA PID 被初始化为 DATA0。

(5) UF0 EP3 状态寄存器 L (UF0E3SL)

该寄存器保存响应 GET_STATUS 端点 3 请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

如果端点 3 发生一个错误，E3HALT 位被 FW 设置为 1。当正在接收对端点 3 的 USB 端访问时，对该寄存器的写入访问被忽略。

当接收到 GET_STATUS 端点 3 请求时，硬件自动将该寄存器的内容发送给主机。如果端点 3 已经停顿，UF0B12 寄存器被清除，且 BK12NK 位被清除为 0。

当传输到端点 3 而不是控制传输时，被执行时写入该寄存器总是被屏蔽，确保检查该寄存器来查看数据是否被正确写入。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E3SL	0	0	0	0	0	0	0	E3HALT	00200158H	00H

位的位置	位名称	功能
0	E3HALT	该位表明端点 3 的状态。 1: 已停顿 (Stalled) 0: 未停顿 当接收到 SET_FEATURE 端点 3 请求时，该位被硬件设置为 1，并且当正确接收到 CLEAR_FEATURE 端点 3 请求，SET_CONFIGURATION 请求，或正确接收到对端点 3 链接的接口的 SET_INTERFACE 请求时，该位被硬件清除为 0。DATA PID 被初始化为 DATA0。

(6) UF0 EP4 状态寄存器 L (UF0E4SL)

该寄存器保存响应 GET_STATUS 端点 4 请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

如果端点 4 发生一个错误，E4HALT 位被 FW 设置为 1。当正在接收对端点 4 的 USB 端访问时，对该寄存器的写入访问被忽略。

当接收到 GET_STATUS 端点 4 请求时，硬件自动将该寄存器的内容发送给主机。如果端点 4 已经停顿，UF0BO2 寄存器被清除，且 BKO2NK 位被清除为 0。

当传输到端点 4 而不是控制传输时，被执行时写入该寄存器总是被屏蔽，确保检查该寄存器来查看数据是否被正确写入。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E4SL	0	0	0	0	0	0	0	E4HALT	0020015CH	00H

位的位置	位名称	功能
0	E4HALT	该位表明端点 4 的状态。 1: 已停顿 (Stalled) 0: 未停顿 当接收到 SET_FEATURE 端点 4 请求时，该位被硬件设置为 1，并且当正确接收到 CLEAR_FEATURE 端点 4 请求，SET_CONFIGURATION 请求，或正确接收到对端点 4 链接的接口的 SET_INTERFACE 请求时，该位被硬件清除为 0。DATA PID 被初始化为 DATA0。

(7) UF0 EP7 状态寄存器 L (UF0E7SL)

该寄存器保存响应 GET_STATUS 端点 7 请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

如果端点 7 发生一个错误，E7HALT 位被 FW 设置为 1。当正在接收对端点 7 的 USB 端访问时，对该寄存器的写入访问被忽略。

当接收到 GET_STATUS 端点 7 请求时，硬件自动将该寄存器的内容发送给主机。如果端点 7 已经停顿，UF0INT1 寄存器被清除，且 IT1NK 位被清除为 0。

当传输到端点 7 而不是控制传输时，被执行时写入该寄存器总是被屏蔽，确保检查该寄存器来查看数据是否被正确写入。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E7SL	0	0	0	0	0	0	0	E7HALT	00200168H	00H

位的位置	位名称	功能
0	E7HALT	该位表明端点 7 的状态。 1: 已停顿 (Stalled) 0: 未停顿 当接收到 SET_FEATURE 端点 7 请求时，该位被硬件设置为 1，并且当正确接收到 CLEAR_FEATURE 端点 7 请求，SET_CONFIGURATION 请求，或正确接收到对端点 7 链接的接口的 SET_INTERFACE 请求时，该位被硬件清除为 0。DATA PID 被初始化为 DATA0。

(8) UF0 地址寄存器 (UF0ADRS)

该寄存器保存设备地址。

该寄存器是只读的，按字节进行读取。

由 SET_ADDRESS 请求发送的设备地址被分析，且作为结果的值被自动写入该寄存器。如果 SET_ADDRESS 请求由 FW 处理，在状态阶段中接收到 SUCCESS 信号时，该寄存器的值作为设备地址被反映出来。

注意事项 不要对该寄存器执行写入访问。如果该寄存器被写入，操作无法保证。

	7	6	5	4	3	2	1	0	地址	复位后
UF0ADRS	0	ADRS6	ADRS5	ADRS4	ADRS3	ADRS2	ADRS1	ADRS0	00200180H	00H

位的位置	位名称	功能
6 至 0	ADRS6 至 ADRS0	这些位保持 SIE 的设备地址。

(9) UF0 配置寄存器 (UF0CNF)

该寄存器保存响应 GET_CONFIGURATION 请求的返回值。

该寄存器是只读的，按字节进行读取。

当接收到 SET_CONFIGURATION 请求时，wValue 被自动写入该寄存器。

当检测到该寄存器的值从 00H 更改为 00H 以外的其它值时，UF0MODS 寄存器的 CONF 位被设置为 1。如果 SET_CONFIGURATION 请求由 FW 处理，只要数据被写入该寄存器（状态阶段完成之前 CONF 位 = 1），该寄存器的状态立即反映到 UF0MODS 寄存器上。

注意事项 不要对该寄存器执行写入访问。如果该寄存器被写入，操作无法保证。

	7	6	5	4	3	2	1	0	地址	复位后
UF0CNF	0	0	0	0	0	0	CONF1	CONF0	00200182H	00H

位的位置	位名称	功能
1, 0	CONF1, CONF0	这些位保存对 GET_CONFIGURATION 请求的响应的返回数据。

(10) UF0 接口 0 寄存器 (UF0IF0)

该寄存器保存响应 GET_INTERFACE wIndex = 0 请求的返回值。

该寄存器是只读的，按字节进行读取。

当接收到 SET_INTERFACE 请求时，wValue 被自动写入该寄存器。

如果 SET_INTERFACE 请求由 FW 处理，wIndex 和 wValue 被译码，并且端点的设置被自动更改。这时根据设置，将目标端点的状态位和 DPID 自动清除为 0。FIFO 不被自动清除。

注意事项 不要对该寄存器执行写入访问。如果该寄存器被写入，操作无法保证。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IF0	0	0	0	0	0	IF02	IF01	IF00	00200184H	00H

位的位置	位名称	功能
2 至 0	IF02 至 IF00	这些位保存对 GET_INTERFACE wIndex = 0 请求的响应的返回值。

(11) UF0 接口 1 至 4 寄存器 (UF0IF1 至 UF0IF4)

该寄存器保存响应 GET_INTERFACE wIndex = n 请求 (n = 1 至 4) 的返回值。

该寄存器是只读的，按字节进行读取。

当接收到 SET_INTERFACE 请求时，wValue 被自动写入该寄存器。

根据 UF0AIFN 和 UF0AAS 寄存器的设置，这些寄存器无效。

如果 SET_INTERFACE 请求由 FW 处理，wIndex 和 wValue 被译码，并且端点的设置被自动更改。这时根据设置，将目标端点的状态位和 DPID 自动清除为 0。FIFO 不被自动清除。

注意事项 不要对该寄存器执行写入访问。如果该寄存器被写入，操作无法保证。

	7	6	5	4	3	2	1	0		
UF0IF1	0	0	0	0	0	IF12	IF11	IF10	00200186H	复位后 00H
	7	6	5	4	3	2	1	0		
UF0IF2	0	0	0	0	0	IF22	IF21	IF20	00200188H	复位后 00H
	7	6	5	4	3	2	1	0		
UF0IF3	0	0	0	0	0	IF32	IF31	IF30	0020018AH	复位后 00H
	7	6	5	4	3	2	1	0		
UF0IF4	0	0	0	0	0	IF42	IF41	IF40	0020018CH	复位后 00H

位的位置	位名称	功能
2 至 0	IFn2 至 IFn0	这些位保存对 GET_INTERFACE wIndex = n 请求的响应的返回值。

备注 n = 1 至 4

(12) UF0 描述符长度寄存器 (UF0DSCL)

该寄存器保存响应 GET_DESCRIPTOR 配置请求的返回值长度。该寄存器的值是 UF0CIEn 寄存器设置的所有描述符的字节数减 1 (n = 0 至 255)。响应 GET_DESCRIPTOR 配置请求时返回的总描述符长度根据该寄存器的值决定。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

wLength 的处理被自动控制。如果该寄存器被设置为 00H，它意味着将要返回的描述符长度为 1 字节。如果寄存器被设置为 FFH，返回的描述符长度为 256 字节。当使用的描述符长度超过 256 字节时，设置 UF0MODC 寄存器的 CDCG DST 位为 1，且由 FW 处理 GET_DESCRIPTOR 请求（这时，UF0MODS 寄存器的 CDCG D 位也被置为 1）。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

		7	6	5	4	3	2	1	0	地址	复位后
UF0DSCL		DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0	002001A0H	00H

位的位置	位名称	功能
7 至 0	DPL7 至 DPL0	这些位设置响应 GET_DESCRIPTOR 配置请求将要返回的所有描述符的字节数减 1。

(13) UF0 设备描述符寄存器 0 至 17 (UF0DD0 至 UF0DD17)

这些寄存器保存响应 GET_DESCRIPTOR 设备请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EPONKA 位被置为 1 时，数据才能写入到该寄存器。

- 注意事项**
1. 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EPONKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。
 2. 使用 USB 2.0 规范定义的值和最新等级规范作为设置值。

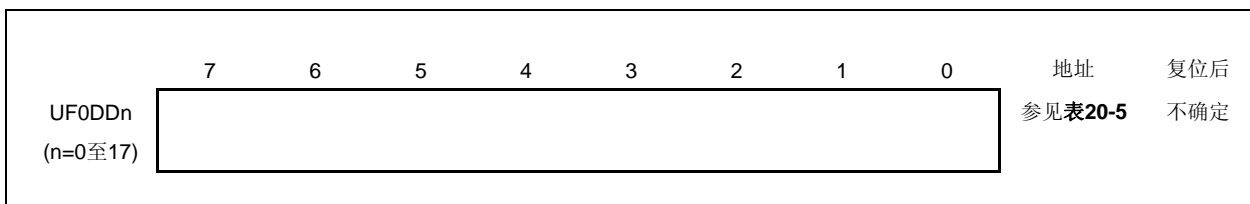


表 20-5. UF0 设备描述符寄存器的映射和数据

符号	地址	域名	内容
UF0DD0	002001A2H	bLength	描述符的大小
UF0DD1	002001A4H	bDescriptorType	设备描述符类型
UF0DD2	002001A6H	bcdUSB	USB 规范的版本号的小数点后面的值
UF0DD3	002001A8H		USB 规范的版本号的小数点前面的值
UF0DD4	002001AAH	bDeviceClass	等级码
UF0DD5	002001ACH	bDeviceSubClass	次等级码
UF0DD6	002001AEH	bDeviceProtocol	协议码
UF0DD7	002001B0H	bMaxPacketSize0	端点 0 的最大包大小
UF0DD8	002001B2H	idVendor	厂商 ID 的低位值
UF0DD9	002001B4H		厂商 ID 的高位值
UF0DD10	002001B6H	idProduct	产品 ID 的低位值
UF0DD11	002001B8H		产品 ID 的高位值
UF0DD12	002001BAH	bcdDevice	设备发布号的低位值
UF0DD13	002001BCH		设备发布号的高位值
UF0DD14	002001BEH	iManufacturer	描述制造商的字符串描述符的索引
UF0DD15	002001C0H	iProduct	描述产品的字符串描述符的索引
UF0DD16	002001C2H	ISerialNumber	描述设备序列号的字符串描述符的索引
UF0DD17	002001C4H	BNumConfigurations	可设置的配置数量

(14) UF0 配置 / 接口 / 端点描述符寄存器 0 至 255 (UF0CIE0 至 UF0CIE255)

这些寄存器保存响应 GET_DESCRIPTOR 配置请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

这些寄存器中可以保存最多 256 字节的描述符信息。按照配置、接口和端点的顺序保存各个描述符（参见表 20-6）。如果有两个或更多接口，在接口描述符之后重复保存数据。

表 20-6. UF0CIE_n 寄存器的映射

地址	保存的描述符
002001C6H	配置描述符 (9 字节)
002001D8H	接口描述符 (9 字节)
002001EAH	端点 1 描述符 (7 字节)
002001F8H	端点 2 描述符 (7 字节)
00200206H	端点 3 描述符 (7 字节)
:	:
002002xxH	接口描述符 (9 字节)
002002xxH+9	端点 1 描述符 (7 字节)
002002xxH+16	端点 2 描述符 (7 字节)
002002xxH+23	端点 3 描述符 (7 字节)
:	:

可以设置到这些寄存器的有效数据的范围根据 UF0DSCLE 寄存器的设置而改变。除了表 20-7 列出的描述符以外，也可以保存特有的等级和厂商描述符。

如果所有的值是固定的，它们可以被存储在 ROM 中。

- 注意事项**
1. 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。
 2. 使用 USB 2.0 规范定义的值和最新等级规范作为设置值。

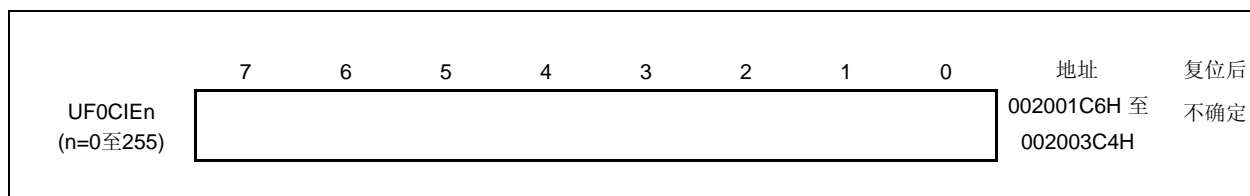


表 20-7. UFGCIEn 寄存器的数据

(a) 配置描述符 (9 字节)

偏移量	域名	内容
0	bLength	该描述符的大小
1	bDescriptorType	描述符类型
2	wTotalLength	配置、所有接口和所有端点描述符的总字节数的低位
3		配置、所有接口和所有端点描述符的总字节数的高位
4	bNumInterface	接口个数
5	bConfigurationValue	选择该配置的值
6	iConfiguration	描述该配置的字符串描述符的索引
7	bmAttributes	该配置的特征 (自供电、无远程唤醒)
8	MaxPower	该配置的最大电源消耗 (单位: mA) ^注

注 该值以 2mA 为单位。(例如: 50 = 100 mA)

(b) 接口描述符 (9 字节)

偏移量	域名	内容
0	bLength	该描述符的大小
1	bDescriptorType	描述符类型
2	bInterfaceNumber	该接口的值
3	bAlternateSetting	选择接口的复用设置的值
4	bNumEndpoints	可用端点的数量
5	bInterfaceClass	等级码
6	bInterfaceSubClass	次等级码
7	bInterfaceProtocol	协议码
8	Interface	描述该接口的字符串描述符的索引

(c) 端点描述符 (7 字节)

偏移量	域名	内容
0	bLength	该描述符的大小
1	bDescriptorType	描述符类型
2	bEndpointAddress	该端点的地址 / 传输方向
3	bmAttributes	传输类型
4	wMaxPaketSize	传输数据的最大数的低位
5		传输数据的最大数的高位
6	bInterval	传输间隔

20.6.6 桥寄存器

(1) 桥中断控制寄存器 (BRGINTT)

BRGINTT 寄存器用于控制中断产生状态的 DMA 传输状态，以及从 EPC 到桥电路的各个端点 (EP1 至 EP4)。

BRGINTT 寄存器可以按 16 位宽度进行读取或写入。

复位后: 0000H R/W 地址: 00200400H

	15	14	13	12	11	10	9	8
BRGINTT	0	0	0	0	EP4INT	EP3INT	EP2INT	EP1INT
	7	6	5	4	3	2	1	0
	0	0	0	0	0	EPCINT2B	EPCINT1B	EPCINT0B

位的位置	位名称	功能
11	EP4INT	在 EP4 中，当 DMA 传输正常结束，或者在 DMA 传输中错误完成，该位被置位。通过写入“1”可以将该位清除位“0”。 0: DMA 传输尚未完成 1: DMA 传输完成
10	EP3INT	在 EP3 中，当 DMA 传输正常结束，或者在 DMA 传输中错误完成，该位被置位。通过写入“1”可以将该位清除位“0”。 0: DMA 传输尚未完成 1: DMA 传输完成
9	EP2INT	在 EP2 中，当 DMA 传输正常结束，或者在 DMA 传输中错误完成，该位被置位。通过写入“1”可以将该位清除位“0”。 0: DMA 传输尚未完成 1: DMA 传输完成
8	EP1INT	在 EP1 中，当 DMA 传输正常结束，或者在 DMA 传输中错误完成，该位被置位。通过写入“1”可以将该位清除位“0”。 0: DMA 传输尚未完成 1: DMA 传输完成
2	EPCINT2B	表示来自 EPC 的中断请求信号“EPC_INT2B”的状态。 清除来自 EPC 寄存器的请求控制。 0: 中断未发出 1: 中断已发出
1	EPCINT1B	表示来自 EPC 的中断请求信号“EPC_INT1B”的状态。 清除来自 EPC 寄存器的请求控制。 0: 中断未发出 1: 中断已发出
0	EPCINT0B	表示来自 EPC 的中断请求信号“EPC_INT0B”的状态。 清除来自 EPC 寄存器的请求控制。 0: 中断未发出 1: 中断已发出

(2) 桥中断使能寄存器 (BRGINTE)

BRGINTE 寄存器用于控制使能或禁止桥电路中的中断发生。

BRGINTE 寄存器可以按 16 位宽度进行读取或写入。

复位后: 0000H R/W 地址: 00200402H

	15	14	13	12	11	10	9	8
BRGINTE	0	0	0	0	EP4INTN	EP3INTN	EP2INTN	EP1INTN
	7	6	5	4	3	2	1	0
	0	0	0	0	0	EPC INT2BEN	EPC INT1BEN	EPC INT0BEN

位的位置	位名称	功能
11	EP4INTN	当 EP4INT 被置位时, 使能或禁止发生中断的设置 0: 禁止中断 1: 使能中断
10	EP3INTN	当 EP3INT 被置位时, 使能或禁止发生中断的设置 0: 禁止中断 1: 使能中断
9	EP2INTN	当 EP2INT 被置位时, 使能或禁止发生中断的设置 0: 禁止中断 1: 使能中断
8	EP1INTN	当 EP1INT 被置位时, 使能或禁止发生中断的设置 0: 禁止中断 1: 使能中断
2	EPCINT2BEN	当 EPCINT2BEN 位被置位时, 使能或禁止发生中断的设置 0: 禁止中断 1: 使能中断
1	EPCINT1BEN	当 EPCINT1BEN 位被置位时, 使能或禁止发生中断的设置 0: 禁止中断 1: 使能中断
0	EPCINT0BEN	当 EPCINT0BEN 位被置位时, 使能或禁止发生中断的设置 0: 禁止中断 1: 使能中断

(3) EPC 模块控制寄存器 (EPCCLT)

EPCCLT 寄存器用于控制到 EPC 模块的复位发生器。

EPCCLT 寄存器可以按 16 位宽度进行读取或写入。

复位后: 0000H R/W 地址: 00200404H

	15	14	13	12	11	10	9	8
EPCCLT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	EPCRST

位的位置	位名称	功能
0	EPCRST	设置 EPC 发生的复位。 0: 复位解除 1: 复位发出

(4) CPU I/F 总线控制寄存器 (CPUBCTL)

CPUBCTL 寄存器用于控制桥电路和 CPU 之间的接口。

CPUBCTL 寄存器可以按 16 位宽度进行读取或写入。

复位后：不确定 R/W 地址：00200408H

	15	14	13	12	11	10	9	8
CPUBCTL	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	BULKWAIT	DATAWAIT	NOWAIT

位的位置	位名称	功能
2	BULKWAIT	当访问批量寄存器时，强行插入 1 等待（批量等待）。 0：未强行插入批量等待*（默认值） 1：强行插入批量等待 注 在写入访问中该设置无效，批量等待被强行插入。
1	DATAWAIT	在 CPU 总线周期后，强行插入 1 等待（数据等待）。 0：未强行插入数据等待（默认值） 1：强行插入数据等待
0	NOWAIT	设置使能/禁止 CPU 总线周期操作的无等待。 0：禁止无等待*（默认值） 1：使能无等待 注 插入 1 等待或更多等待

20.6.7 DMA 寄存器

(1) EPn DMA 控制寄存器 1 (UF0E1DC1 至 UF0E4DC1)

UF0E1DC1 至 UF0E4DC1 寄存器用于控制端点 n (EPn) 的 DMA 传输。(n = 1 至 4)

UF0E1DC1 至 UF0E4DC1 寄存器可以按 16 位宽度进行读取或写入。

(1/2)

复位后: 0000H R/W 地址: 00200500H

	15	14	13	12	11	10	9	8
UF0E1DC1	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	EP1BULK2	EP1BULK1	EP1BULK0	EP1STOP	EP1REQ	EP1DMAEN

复位后: 0000H R/W 地址: 00200504H

	15	14	13	12	11	10	9	8
UF0E2DC1	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	EP2BULK2	EP2BULK1	EP2BULK0	EP2STOP	EP2REQ	EP2DMAEN

复位后: 0000H R/W 地址: 00200508H

	15	14	13	12	11	10	9	8
UF0E3DC1	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	EP3BULK2	EP3BULK1	EP3BULK0	EP3STOP	EP3REQ	EP3DMAEN

复位后: 0000H R/W 地址: 0020050CH

	15	14	13	12	11	10	9	8
UF0E4DC1	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	EP4BULK2	EP4BULK1	EP4BULK0	EP4STOP	EP4REQ	EP4DMAEN

位的位置	位名称	功能																								
5 至 3	EPnBULK2, EPnBULK1, EPnBULK0	<p>表示内部桥的批量传输时状态机“BIN_STATE”的状态</p> <table border="1"> <thead> <tr> <th>EPnBULK2</th> <th>EPnBULK1</th> <th>EPnBULK0</th> <th>“BIN_STATE” 状态</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>BIN_IDLE</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>BIN_CPU</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>BIN_EPC</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>BIN_CMP</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>BIN_END</td> </tr> </tbody> </table>	EPnBULK2	EPnBULK1	EPnBULK0	“BIN_STATE” 状态	0	0	0	BIN_IDLE	0	0	1	BIN_CPU	0	1	0	BIN_EPC	0	1	1	BIN_CMP	1	0	0	BIN_END
EPnBULK2	EPnBULK1	EPnBULK0	“BIN_STATE” 状态																							
0	0	0	BIN_IDLE																							
0	0	1	BIN_CPU																							
0	1	0	BIN_EPC																							
0	1	1	BIN_CMP																							
1	0	0	BIN_END																							
2	EPnSTOP	<p>表示来自 EPC 的 DMA 传输结束状态 (DMA 传输的结束因素)</p> <p>0: 通过 EPn_TCNT 值为“0”结束 DMA 传输</p> <p>1: 通过对“EPC_DMARQ_EPnB”求反结束 DMA 传输</p> <p>设置下一个 EP1_DMAEN 为 “1”，则自动清除(0)。</p>																								
1	EPnREQ	<p>表示来自 EPC 的 “EPC_DMARQ_EPnB”信号的状态</p> <p>0: DMA 请求信号</p> <p>1: 没有 DMA 请求信号</p>																								
0	EPnDMAEN	<p>设置对来自 EPC 的 DMA 请求的控制</p> <p>0: 屏蔽 DMA 请求</p> <p>1: 使能 DMA 请求</p> <p>EPn_TCNT 设置的包的数量传输完成时，或完成对 DMARQ_EPnB 求反数量的 DMA 传输时，自动清除(0)。</p> <p>注意事项 在强行结束中该设置的值无法保证。</p>																								

备注 n = 1 至 4

(2) EPn DMA 控制寄存器 2 (UF0E1DC2 至 UF0E4DC2)

UF0E1DC2 至 UF0E4DC2 寄存器用于控制端点 n (EPn) 的 DMA 传输。(n = 1 至 4)

UF0E1DC2 至 UF0E4DC2 寄存器可以按 16 位宽度进行读取或写入。

(1/2)

复位后: 0000H R/W 地址: 00200502H

	15	14	13	12	11	10	9	8
UF0E1DC2	EP1 TCNT15	EP1 TCNT14	EP1 TCNT13	EP1 TCNT12	EP1 TCNT11	EP1 TCNT10	EP1 TCNT9	EP1 TCNT8
	7	6	5	4	3	2	1	0
	EP1 TCNT7	EP1 TCNT6	EP1 TCNT5	EP1 TCNT4	EP1 TCNT3	EP1 TCNT2	EP1 TCNT1	EP1 TCNT0

复位后: 0000H R/W 地址: 00200506H

	15	14	13	12	11	10	9	8
UF0E2DC2	EP2 TCNT15	EP2 TCNT14	EP2 TCNT13	EP2 TCNT12	EP2 TCNT11	EP2 TCNT10	EP2 TCNT9	EP2 TCNT8
	7	6	5	4	3	2	1	0
	EP2 TCNT7	EP2 TCNT6	EP2 TCNT5	EP2 TCNT4	EP2 TCNT3	EP2 TCNT2	EP2 TCNT1	EP2 TCNT0

复位后: 0000H R/W 地址: 0020050AH

	15	14	13	12	11	10	9	8
UF0E3DC2	EP3 TCNT15	EP3 TCNT14	EP3 TCNT13	EP3 TCNT12	EP3 TCNT11	EP3 TCNT10	EP3 TCNT9	EP3 TCNT8
	7	6	5	4	3	2	1	0
	EP3 TCNT7	EP3 TCNT6	EP3 TCNT5	EP3 TCNT4	EP3 TCNT3	EP3 TCNT2	EP3 TCNT1	EP3 TCNT0

复位后: 0000H R/W 地址: 0020050EH

	15	14	13	12	11	10	9	8
UF0E4DC2	EP4 TCNT15	EP4 TCNT14	EP4 TCNT13	EP4 TCNT12	EP4 TCNT11	EP4 TCNT10	EP4 TCNT9	EP4 TCNT8
	7	6	5	4	3	2	1	0
	EP4 TCNT7	EP4 TCNT6	EP4 TCNT5	EP4 TCNT4	EP4 TCNT3	EP4 TCNT2	EP4 TCNT1	EP4 TCNT0

位的位置	位名称	功能
15 至 0	EPnTCNT15 至 EPnTCNT0	设置 EPn 中 DMA 传输的字节数量。 在每次传输中，EPn_TCNT 的值递减到“0”，则结束 DMA 传输。 注意事项 <ol style="list-style-type: none"> 1. 当 EPn_DMAEN = 0 时，设置该寄存器。 2. 禁止将该寄存器设置为“0”。 请确保为 DMA 传输计数寄存器 DBC0 至 DBC3 设置的值就是该寄存器的值 +1。 3. 该寄存器的设置值可以为桥内的批量传输反映计数器 BIN_TCNT 的值。 当 BIN_TCNT 的值为“0”，EPn_TCNT 也为“0”。 4. 当强行中止时，停止为批量传输更新计数器 BIN_TCNT 的值。

备注 n = 1 至 4

20.6.8 批量-in 寄存器

(1) UF0 EP1 批量-in 传输数据寄存器 (UF0EP1BI)

UF0EP1BI 寄存器用于写入 EP1 的批量-in 传输数据。

UF0EP1BI 寄存器可以按 16 位宽度进行读取或写入。

复位后: 0000H R/W 地址: 00201000H								
	15	14	13	12	11	10	9	8
UF0EP1BI	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	EP1BI7	EP1BI6	EP1BI5	EP1BI4	EP1BI3	EP1BI2	EP1BI1	EP1BI0
位的位置	位名称	功能						
7 至 0	EP1BI7 至 EP1BI0	写入 EP1 的批量-in 传输数据。 将数据写入该寄存器, 数据就会输出到 EPC 模块。 如果使用该寄存器, 将地址 (00201000H) 设置为 DMAC 的 DMA 目的地址寄存器 (DDAn (n = 0 至 3))。						

(2) UF0 EP3 批量-in 传输数据寄存器 (UF0EP3BI)

UF0EP3BI 寄存器用于写入 EP3 的批量-in 传输数据。

UF0EP3BI 寄存器可以按 16 位宽度进行读取或写入。

复位后: 0000H R/W 地址: 00202000H								
	15	14	13	12	11	10	9	8
UF0EP3BI	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	EP3BI7	EP3BI6	EP3BI5	EP3BI4	EP3BI3	EP3BI2	EP3BI1	EP3BI0
位的位置	位名称	功能						
7 至 0	EP3BI7 至 EP3BI0	写入 EP3 的批量-in 传输数据。 将数据写入该寄存器, 数据就会输出到 EPC 模块。 如果使用该寄存器, 将地址 (00202000H) 设置为 DMAC 的 DMA 目的地址寄存器 (DDAn (n = 0 至 3))。						

20.6.9 批量-out寄存器

(1) UF0 EP2 批量-out 传输数据寄存器 (UF0EP2BO)

UF0EP2BO 寄存器用于读取 EP2 的批量-out 传输数据。

UF0EP2BO 寄存器可以按 16 位宽度进行读取。

复位后: 0000H R 地址: 00210000H							
15	14	13	12	11	10	9	8
UF0EP2BO	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
EP2BO7	EP2BO6	EP2BO5	EP2BO4	EP2BO3	EP2BO2	EP2BO1	EP2BO0
位的位置	位名称	功能					
7 至 0	EP2BO7 至 EP2BO0	读取 EP2 的批量-out 传输数据。 从该寄存器读取来自 EPC 模块的输入数据。 如果使用该寄存器, 将地址 (00210000H) 设置为 DMAC 的 DMA 源地址寄存器 (DSAn (n = 0 至 3))。					

(2) UF0 EP4 批量-out 传输数据寄存器 (UF0EP4BO)

UF0EP4BO 寄存器用于读取 EP4 的批量-out 传输数据。

UF0EP4BO 寄存器可以按 16 位宽度进行读取。

复位后: 0000H R 地址: 00220000H							
15	14	13	12	11	10	9	8
UF0EP4BO	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
EP4BO7	EP4BO6	EP4BO5	EP4BO4	EP4BO3	EP4BO2	EP4BO1	EP4BO0
位的位置	位名称	功能					
7 至 0	EP4BO7 至 EP4BO0	读取 EP4 的批量-out 传输数据。 从该寄存器读取来自 EPC 模块的输入数据。 如果使用该寄存器, 将地址 (00220000H) 设置为 DMAC 的 DMA 源地址寄存器 (DSAn (n = 0 至 3))。					

20.6.10 外设控制寄存器

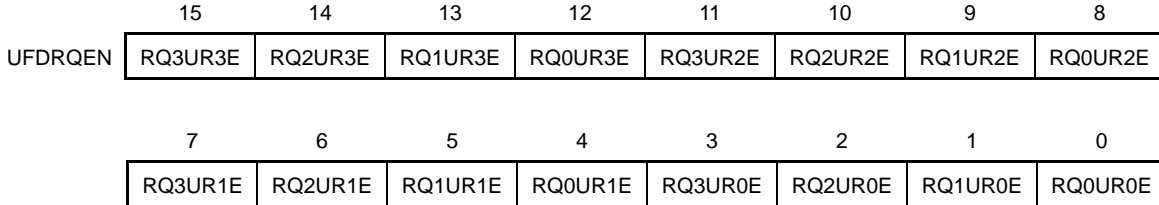
(1) USBF DMA 请求使能寄存器 (UFDRQEN)

UFDRQEN 寄存器用于指定待使用的 DMA 通道和要进行传输的端点。

UFDRQEN 寄存器可以按 8 位或 16 位宽度进行读取或写入。

(1/2)

复位后: 0000H R/W 地址: 00240000H



位的位置	位名称	功能																														
15, 11, 7, 3	RQ3UR3E, RQ3UR2E, RQ3UR1E, RQ3UR0E	指定通过 DMA 通道 3 进行传输的端点 n (EPn)。 (n = 1 至 4) <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>RQ3UR3E</th> <th>RQ3UR2E</th> <th>RQ3UR1E</th> <th>RQ3UR0E</th> <th>通过 DMA3 传输的 EP</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>EP4</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>EP3</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>EP2</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>EP1</td> </tr> <tr> <td colspan="4">其它</td> <td>DMA3 不传输 EPn (不使用 DMA3)</td> </tr> </tbody> </table>	RQ3UR3E	RQ3UR2E	RQ3UR1E	RQ3UR0E	通过 DMA3 传输的 EP	1	0	0	0	EP4	0	1	0	0	EP3	0	0	1	0	EP2	0	0	0	1	EP1	其它				DMA3 不传输 EPn (不使用 DMA3)
RQ3UR3E	RQ3UR2E	RQ3UR1E	RQ3UR0E	通过 DMA3 传输的 EP																												
1	0	0	0	EP4																												
0	1	0	0	EP3																												
0	0	1	0	EP2																												
0	0	0	1	EP1																												
其它				DMA3 不传输 EPn (不使用 DMA3)																												
14, 10, 6, 2	RQ2UR3E, RQ2UR2E, RQ2UR1E, RQ2UR0E	指定通过 DMA 通道 2 进行传输的端点 n (EPn)。 (n = 1 至 4) <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>RQ2UR3E</th> <th>RQ2UR2E</th> <th>RQ2UR1E</th> <th>RQ2UR0E</th> <th>通过 DMA2 传输的 EP</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>EP4</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>EP3</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>EP2</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>EP1</td> </tr> <tr> <td colspan="4">其它</td> <td>DMA2 不传输 EPn (不使用 DMA2)</td> </tr> </tbody> </table>	RQ2UR3E	RQ2UR2E	RQ2UR1E	RQ2UR0E	通过 DMA2 传输的 EP	1	0	0	0	EP4	0	1	0	0	EP3	0	0	1	0	EP2	0	0	0	1	EP1	其它				DMA2 不传输 EPn (不使用 DMA2)
RQ2UR3E	RQ2UR2E	RQ2UR1E	RQ2UR0E	通过 DMA2 传输的 EP																												
1	0	0	0	EP4																												
0	1	0	0	EP3																												
0	0	1	0	EP2																												
0	0	0	1	EP1																												
其它				DMA2 不传输 EPn (不使用 DMA2)																												

位的位置	位名称	功能				
13, 9, 5, 1	RQ1UR3E, RQ1UR2E, RQ1UR1E, RQ1UR0E	指定通过 DMA 通道 1 进行传输的端点 n (EPn)。 (n = 1 至 4)				
		RQ3UR3E	RQ3UR2E	RQ3UR1E	RQ3UR0E	通过 DMA1 传输的 EP
		1	0	0	0	EP4
		0	1	0	0	EP3
		0	0	1	0	EP2
		0	0	0	1	EP1
	其它	DMA1 不传输 EPn (不使用 DMA1)				
12, 8, 4, 0	RQ0UR3E, RQ0UR2E, RQ0UR1E, RQ0UR0E	指定通过 DMA 通道 0 进行传输的端点 n (EPn)。 (n = 1 至 4)				
		RQ3UR3E	RQ3UR2E	RQ3UR1E	RQ3UR0E	通过 DMA0 传输的 EP
		1	0	0	0	EP4
		0	1	0	0	EP3
		0	0	1	0	EP2
		0	0	0	1	EP1
	其它	DMA0 不传输 EPn (不使用 DMA0)				

- 注意事项**
- 禁止对多个 DMA 通道设置同样的 DMA 传输目标，也禁止对同一个 DMA 通道设置多个 DMA 传输目标。
 - 如果使用该寄存器，设置 DMA 触发因素寄存器(DTFRn (n = 0 至 3))为通过中断(00H)禁止 DMA 请求。

下面的流程图展示了主机断开然后重新连接的程序执行，以及上电时的程序执行情况。

图 20-12. 主机断开然后重新连接的程序流程图

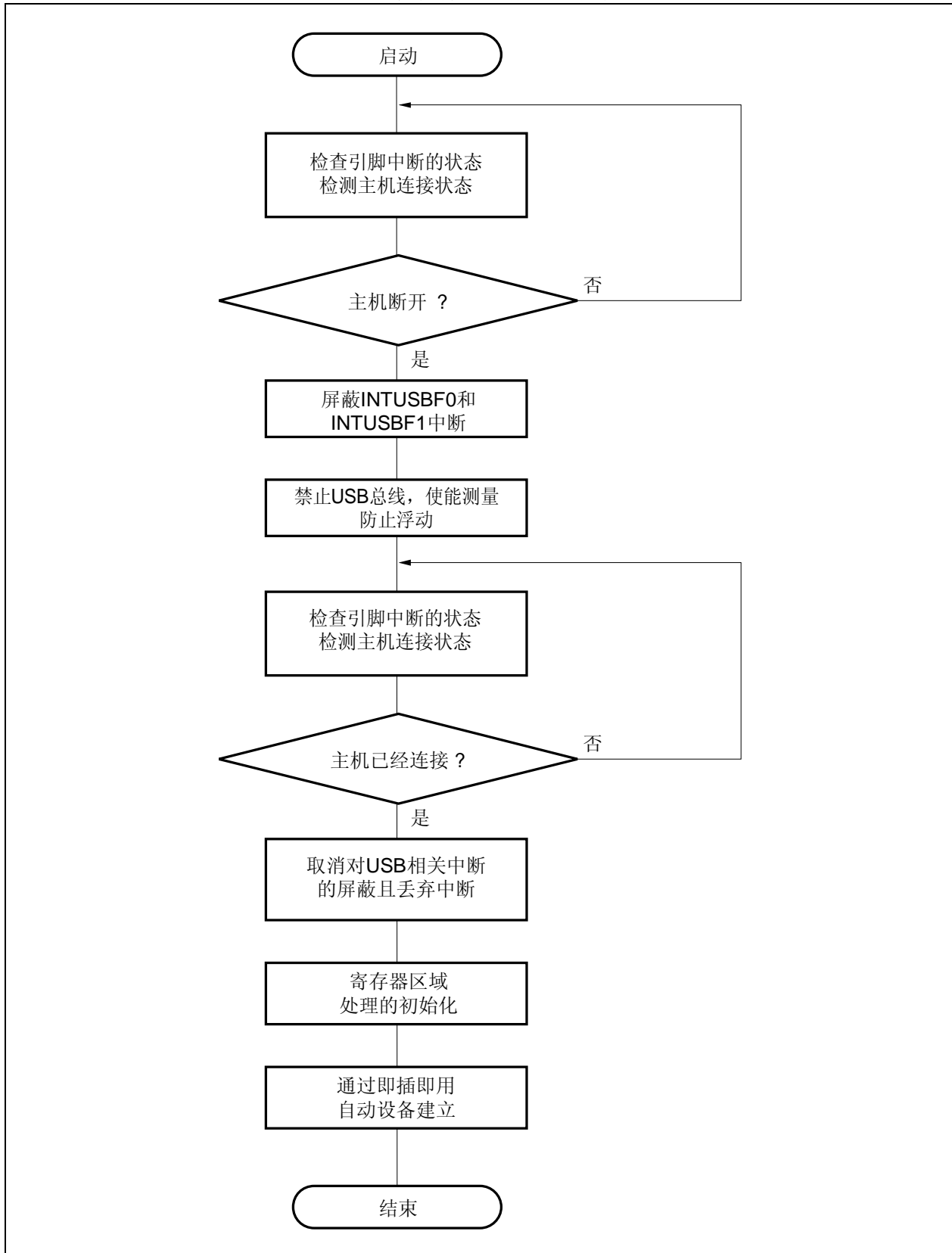
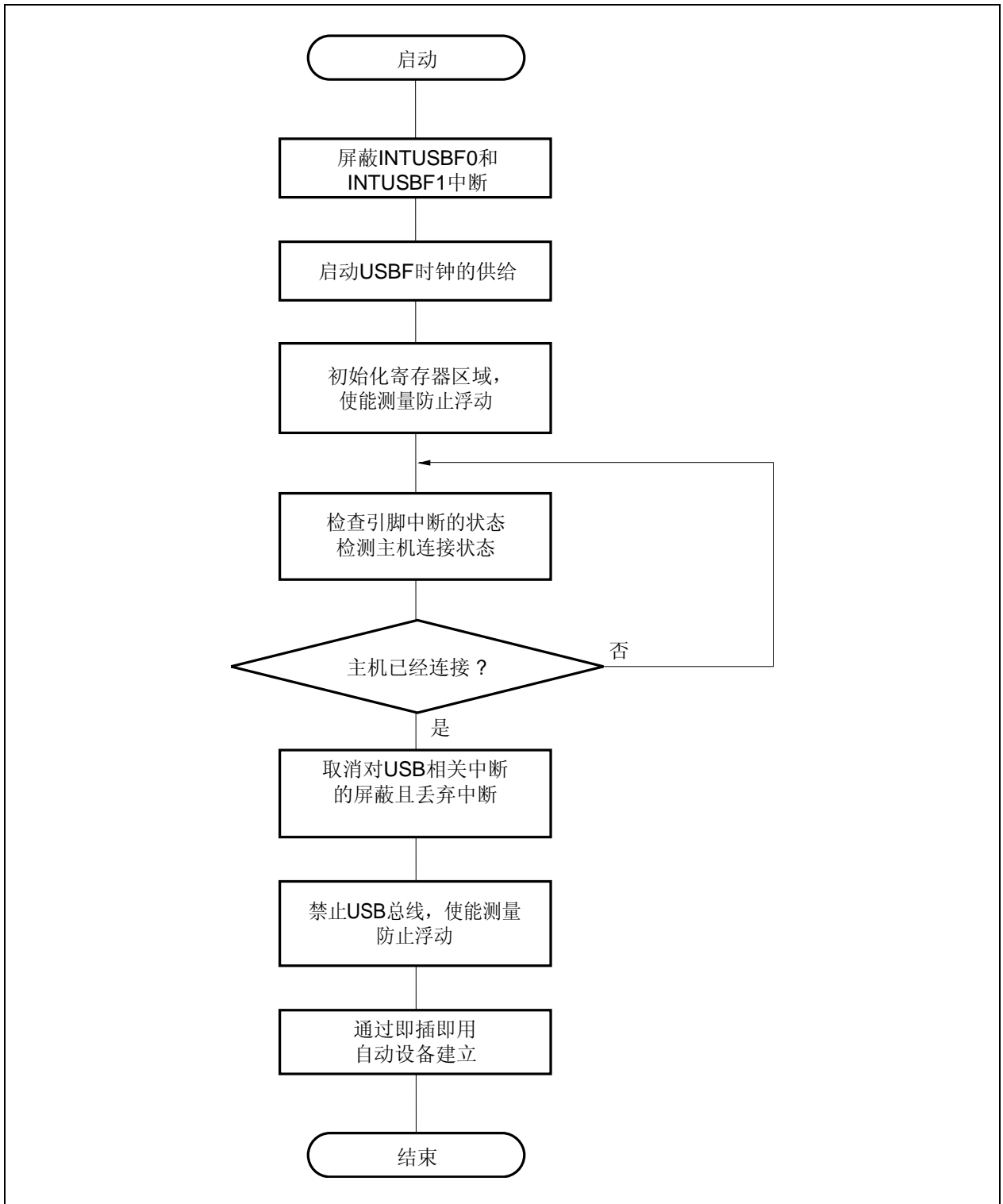


图 20-13. 上电时的程序流程图



20.7 STALL 握手或无握手

对 USBF 错误处理的定义如下。

传输类型	事务	目标包	错误类型	功能响应	处理
控制传输/ 批量传输/ 中断传输	IN/OUT/SETUP	令牌	端点不支持	无响应	无
			端点传输方向不匹配	无响应	无
			CRC 错误	无响应	无
			位填充错误	无响应	无
控制传输/ 批量传输	OUT/SETUP	数据	超时	无响应	无
			PID 检查错误	无响应	无
			不支持的 PID (Data PID 除外)	无响应	无
			CRC 错误	无响应	丢弃接收到的数据
	位填充错误	无响应	丢弃接收到的数据		
OUT	数据	数据 PID 不匹配	ACK	丢弃接收到的数据	
控制传输 (SETUP 阶段)	SETUP	数据	溢出	无响应	丢弃接收到的数据
控制传输 (数据阶段)	OUT	数据	溢出	无响应 ^{注 1}	设置 UF0SDS 寄存器的 SNDSTL 位为 1, 并且丢弃接收到的数据
控制传输 (状态阶段)	OUT	数据	溢出	ACK 或 无响应 ^{注 2}	设置 UF0SDS 寄存器的 SNDSTL 位为 1 并且丢弃接收到的数据
批量传输	OUT	数据	溢出	无响应 ^{注 1}	设置 UF0EnSL 寄存器的 EnHALT 位 (n = 0 至 4, 7, 8) 为 1
控制传输/ 批量传输/ 中断传输	IN	握手	PID 检查错误	-	保持传输的数据并重新传输数据 ^{注 3}
			不支持的 PID (ACK PID 除外)	-	保持传输的数据并重新传输数据 ^{注 3}
			超时	-	保持传输的数据并重新传输数据 ^{注 3}

- 注
1. 主机产生一个 STALL 响应来重新传输。
 2. 如果传输数据小于最大包长度 (MaxPacketSize), 产生一个 ACK 响应并且丢弃状态阶段接收到的数据。如果超过最大包长度, 不产生响应, UF0SDS 寄存器的 SNDSTL 位被设置为 1, 并且丢弃接收到的数据。
 3. 如果在控制传输期间, 接收到一个 OUT 事务, 表明从数据阶段更改到状态阶段, 错误不被处理且认为接收已经正确完成。

- 注意事项
1. 通过当前设置的复用设置号码来判断目标端点是有效还是无效。
 2. 关于包含在控制传输中的到/从端点 0 的响应, 见 20.5 请求。

20.8 特定状态下寄存器的值

表 20-8. 特定状态下寄存器的值 (1/2)

寄存器名	CPU 复位后 (RESET)	总线复位后
UF0E0N 寄存器	00H	值被保持。
UF0E0NA 寄存器	00H	值被保持。
UF0EN 寄存器	00H	值被保持。
UF0ENM 寄存器	00H	值被保持。
UF0SDS 寄存器	00H	值被保持。
UF0CLR 寄存器	00H	值被保持。
UF0SET 寄存器	00H	值被保持。
UF0EPS0 寄存器	00H	值被保持。
UF0EPS1 寄存器	00H	值被保持。
UF0EPS2 寄存器	00H	值被保持。
UF0IS0 寄存器	00H	值被保持。
UF0IS1 寄存器	00H	值被保持。
UF0IS2 寄存器	00H	值被保持。
UF0IS3 寄存器	00H	值被保持。
UF0IS4 寄存器	00H	值被保持。
UF0IM0 寄存器	00H	值被保持。
UF0IM1 寄存器	00H	值被保持。
UF0IM2 寄存器	00H	值被保持。
UF0IM3 寄存器	00H	值被保持。
UF0IM4 寄存器	00H	值被保持。
UF0IC0 寄存器	FFH	值被保持。
UF0IC1 寄存器	FFH	值被保持。
UF0IC2 寄存器	FFH	值被保持。
UF0IC3 寄存器	FFH	值被保持。
UF0IC4 寄存器	FFH	值被保持。
UF0IDR 寄存器	00H	值被保持。
UF0DMS0 寄存器	00H	值被保持。
UF0DMS1 寄存器	00H	值被保持。
UF0FIC0 寄存器	00H	值被保持。
UF0FIC1 寄存器	00H	值被保持。
UF0DEND 寄存器	00H	值被保持。
UF0GPR 寄存器	00H	值被保持。
UF0MODC 寄存器	00H	值被保持。
UF0MODS 寄存器	00H	第 2 位 (CONF) : 清除 (0), 其它位: 值被保持。
UF0AIFN 寄存器	00H	值被保持。
UF0AAS 寄存器	00H	值被保持。
UF0ASS 寄存器	00H	00H
UF0E1IM 寄存器	00H	值被保持。
UF0E2IM 寄存器	00H	值被保持。

表 20-8. 特定状态下寄存器的值 (2/2)

寄存器名	CPU 复位后 (RESET)	总线复位后
UF0E3IM 寄存器	00H	值被保持。
UF0E4IM 寄存器	00H	值被保持。
UF0E7IM 寄存器	00H	值被保持。
UF0E0R 寄存器	不确定 ^{注 1}	值被保持。
UF0E0L 寄存器	00H	值被保持。
UF0E0ST 寄存器	00H	00H
UF0E0W 寄存器	不确定 ^{注 1}	值被保持。
UF0B01 寄存器	不确定 ^{注 1}	值被保持。
UF0B01L 寄存器	00H	值被保持。
UF0B02 寄存器	不确定 ^{注 1}	值被保持。
UF0B02L 寄存器	00H	值被保持。
UF0B11 寄存器	不确定 ^{注 1}	值被保持。
UF0B12 寄存器	不确定 ^{注 1}	值被保持。
UF0INT1 寄存器	不确定	值被保持。
UF0INT2 寄存器	不确定	值被保持。
UF0DSTL 寄存器	00H	00H
UF0E0SL 寄存器	00H	00H
UF0E1SL 寄存器	00H	00H
UF0E2SL 寄存器	00H	00H
UF0E3SL 寄存器	00H	00H
UF0E4SL 寄存器	00H	00H
UF0E7SL 寄存器	00H	00H
UF0ADRS 寄存器	00H	00H
UF0CNF 寄存器	00H	00H
UF0IF0 寄存器	00H	00H
UF0IF1 寄存器	00H	00H
UF0IF2 寄存器	00H	00H
UF0IF3 寄存器	00H	00H
UF0IF4 寄存器	00H	00H
UF0DSCL 寄存器	00H	值被保持。
UF0DDn 寄存器 (n = 0 至 17)	注 2	注 2
UF0CIEn 寄存器 (n = 0 至 255)	注 2	注 2

- 注
1. 该寄存器可以由 RESET 信号清除为 0，因为它的写指针、计数器和读取指针在 RESET 信号变为有效时被清除为 0，与被 UF0FICn 寄存器清除相同，因为该寄存器由 FIFO 控制。
 2. 该寄存器不能被清除为 0。因为可以通过 FW 写入数据，但是，任意值都可以被写入该寄存器（这样做之前，请确保设置 UF0E0NA 寄存器的 EP0NKA 位为 1）。

20.9 FW 处理

执行以下 FW 处理。

- 枚举处理期间，设备端对 SET_CONFIGURATION、SET_INTERFACE、SET_FEATURE 和 CLEAR_FEATURE 请求的设置处理
- XXXXStandard, XXXXClass 和 XXXXVendor 请求的分析和处理不需要自动处理
- 批量传输的 OUT 令牌后，从接收缓冲器读取数据
- 写入响应批量传输 IN 令牌将要返回的数据
- 写入响应中断传输令牌将要返回的数据

下表列出了 FW 支持的请求。

表 20-9. FW 支持的标准请求

请求	接收端	处理 / 频率	说明
CLEAR_FEATURE	接口	自动 STALL 响应	因为没有功能选择器的值，尽管它为 <code>bmRequestType</code> 保留，被认为该请求没有到达接口。 当接收到该请求时，硬件执行自动 STALL 响应。
SET_FEATURE	接口	自动 STALL 响应	因为没有功能选择器的值，尽管它为 <code>bmRequestType</code> 保留，被认为该请求没有到达接口。 当接收到该请求时，硬件执行自动 STALL 响应。
GET_DESCRIPTOR	字符串	FW	返回字符串描述符。 当该请求由 SETUP 令牌接收时，硬件为 FW 产生 CPUDEC 中断请求。FW 从 CPUDEC 中断请求对请求的内容进行译码，并且将要返回给主机的数据写入到 <code>UFOE0W</code> 寄存器。
SET_DESCRIPTOR	设备	FW	重写设备描述符。 当该请求由 SETUP 令牌接收时，硬件为 FW 产生 CPUDEC 中断请求。FW 从 CPUDEC 中断请求对请求的内容进行译码，并且为下一个控制传输 (OUT) 将数据写入 <code>UF0DDn</code> 寄存器 ($n = 0$ 至 17)。
SET_DESCRIPTOR	配置	FW	重写配置描述符。 当该请求由 SETUP 令牌接收时，硬件为 FW 产生 CPUDEC 中断请求。FW 从 CPUDEC 中断请求对请求的内容进行译码，并且为下一个控制传输 (OUT) 将数据写入 <code>UF0CIEn</code> 寄存器 ($n = 0$ 至 255)。
SET_DESCRIPTOR	字符串	FW	重写字符串描述符。 当该请求由 SETUP 令牌接收时，硬件为 FW 产生 CPUDEC 中断请求。FW 从 CPUDEC 中断请求对请求的内容进行译码，并且为下一个控制传输 (OUT) 加载数据。
其它	NA	FW	当该请求由 SETUP 令牌接收时，硬件为 FW 产生 CPUDEC 中断请求。FW 从 CPUDEC 中断请求对请求的内容进行译码，并且执行必需的处理。

20.9.1 初始化处理

初始化处理按照下面两种方式执行。

- 请求数据寄存器的初始化
- 中断的设置

当请求数据寄存器被初始化时，用于 GET_XXXX 请求的数据被写入，对该请求会自动返回一个值，并且一个端点被分配给一个接口。在中断设置中，不是必须检查的中断源可以使用 UF0IMn 寄存器 (n = 0 至 4) 来屏蔽。

以下流程图展示上述处理。

图 20-14. 请求数据寄存器初始化

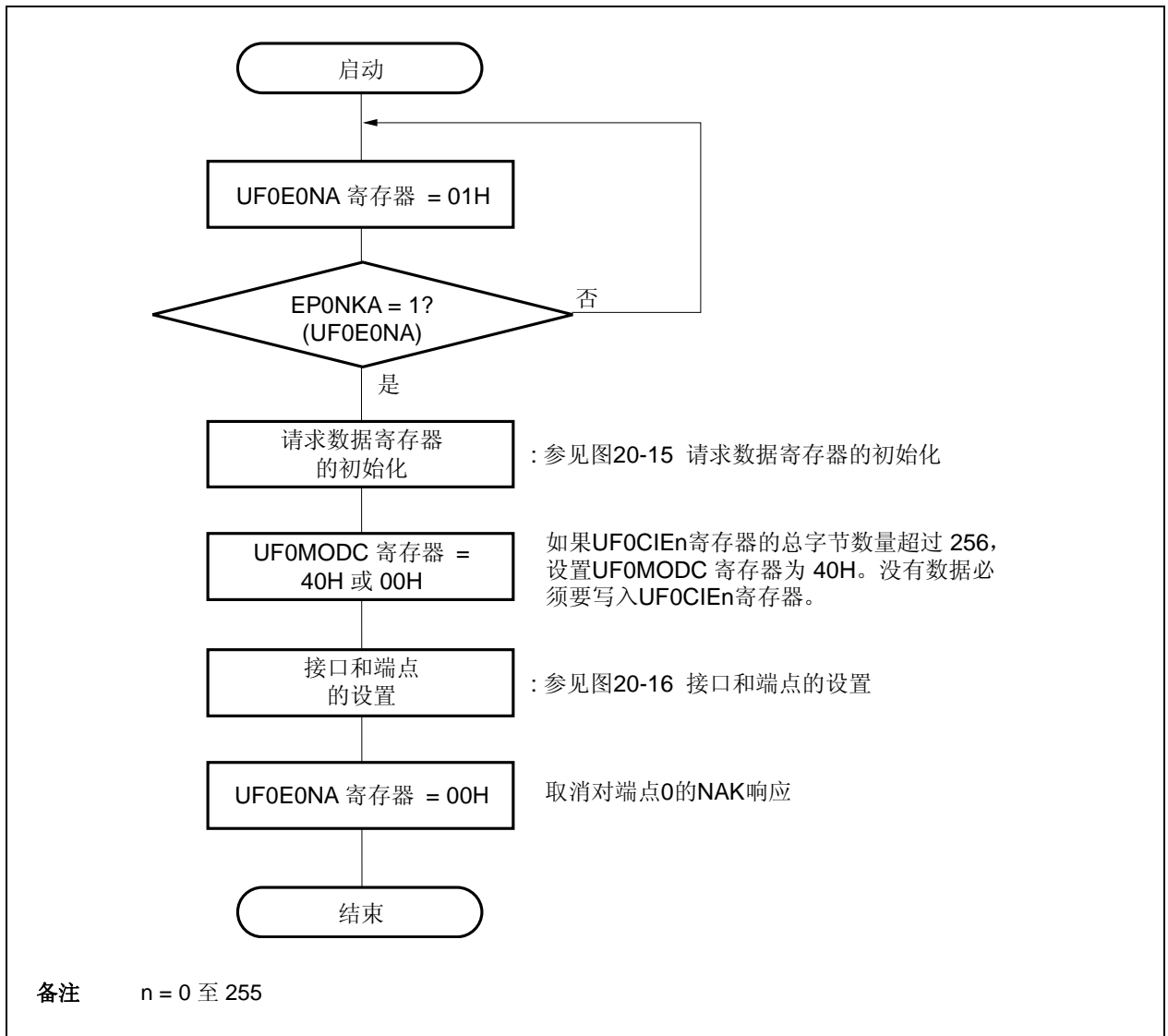


图 20-15. 请求数据寄存器区域的初始化

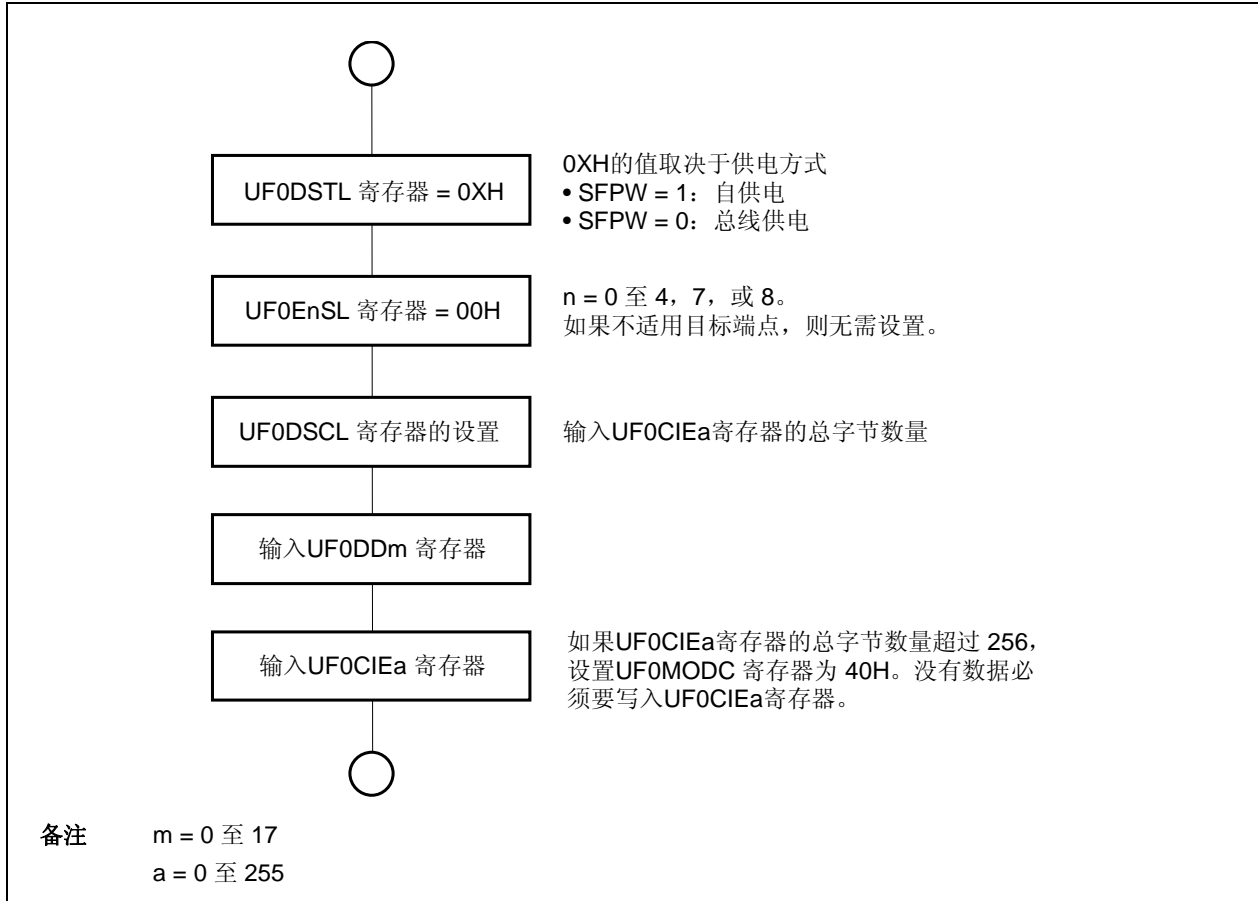


图 20-16. 接口和端点的设置

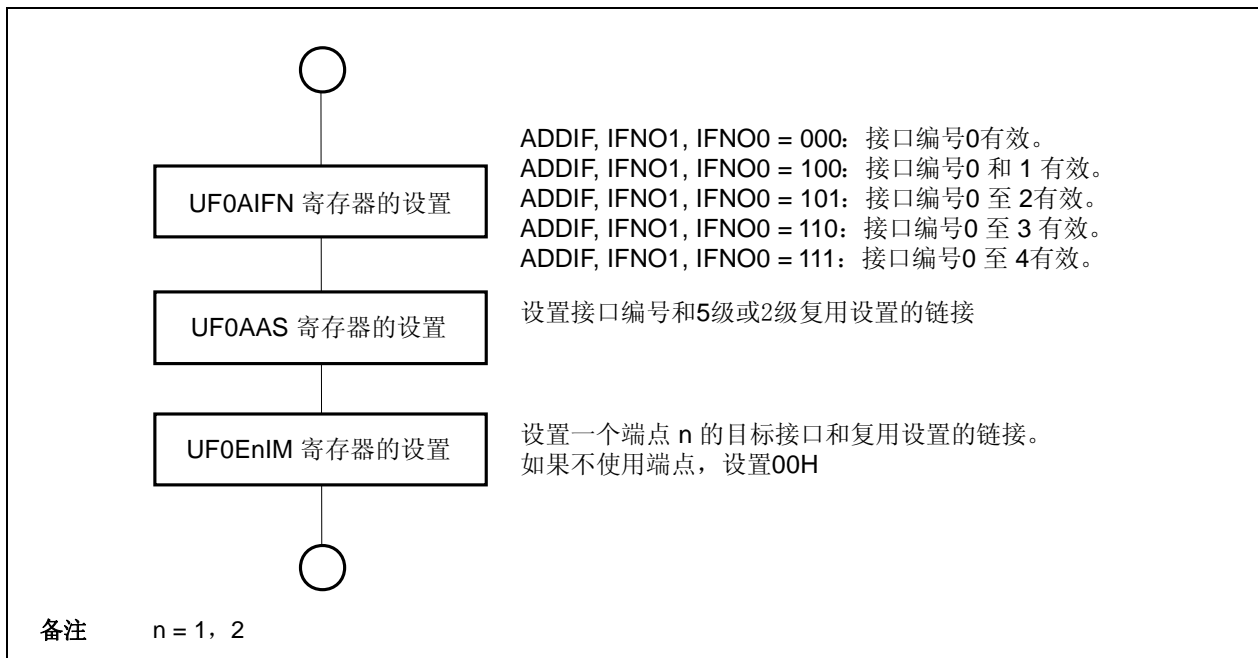
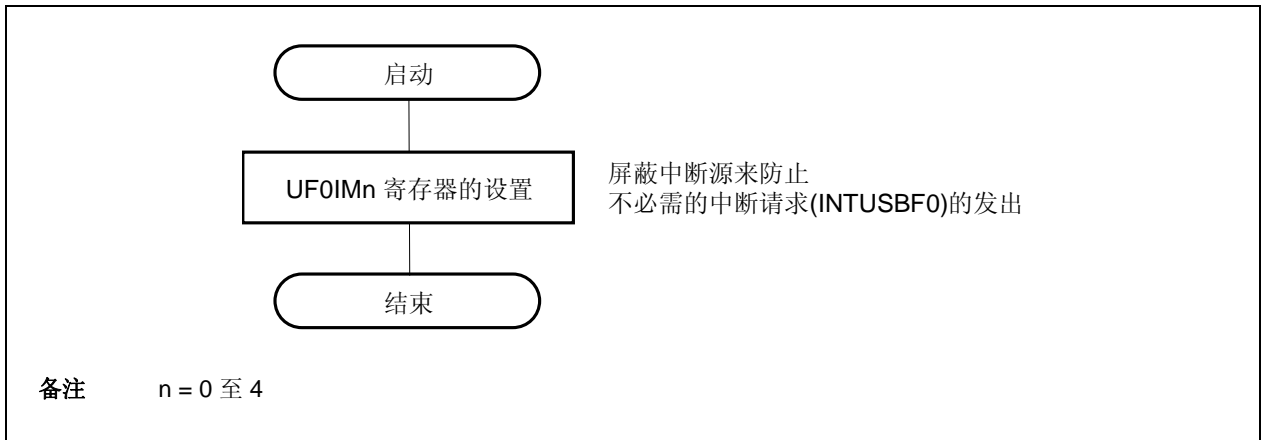


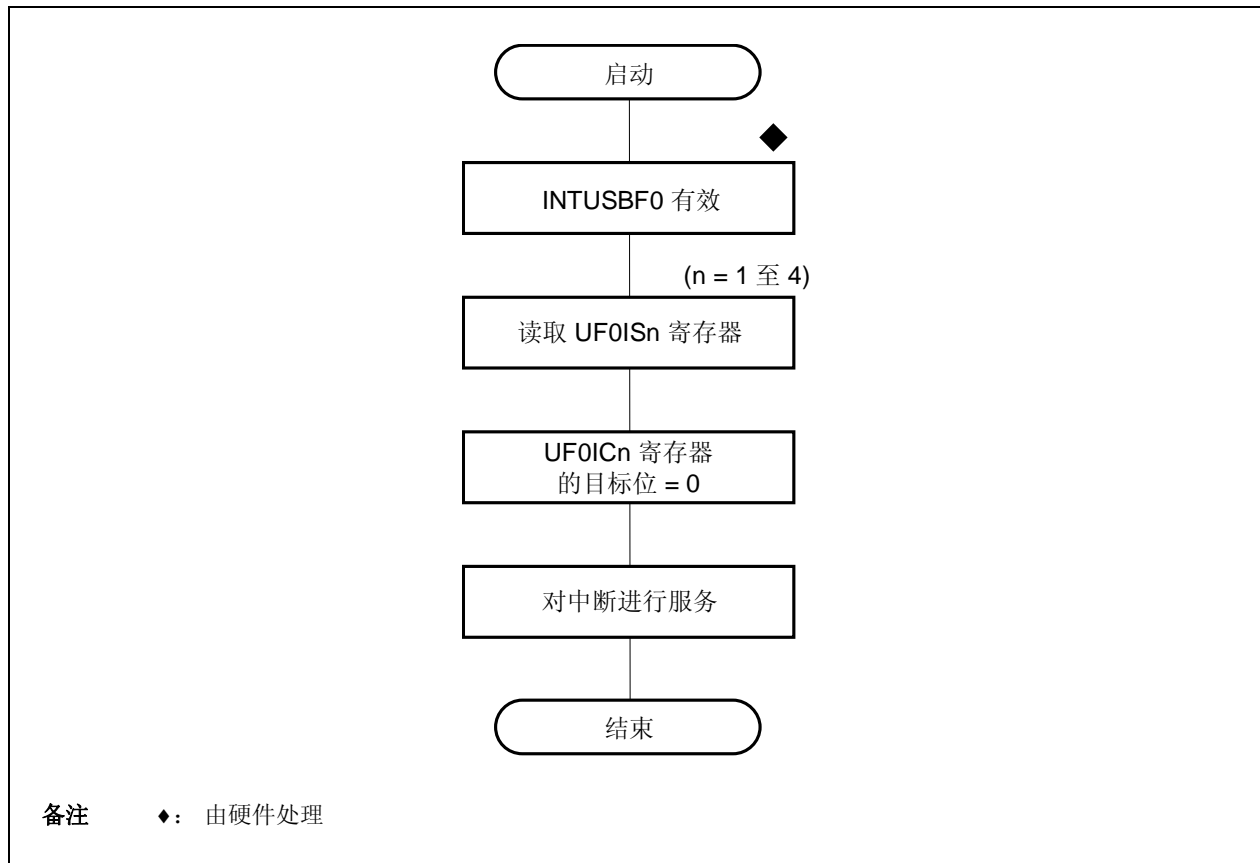
图 20-17. 中断的设置



20.9.2 中断服务

以下流程图展示如何服务一个中断。

图 20-18. 中断服务



当给定的条件满足时，UF0ISn 寄存器下列各位由硬件自动清除（n = 1 至 4）。

- UF0IS1 寄存器的 E0INDT、E0ODT、SUCES、STG 和 CPUDEC 位
- UF0IS2 寄存器的 BKI2DT、BKI1DT、IT2DT 和 IT1DT 位
- UF0IS3 寄存器的 BKO2FL、BKO2DT、BKO1FL 和 BKO1DT 位

因为由 UF0ICn 寄存器清除的中断源优先级比硬件设置的中断源优先级低，根据时序，中断源可能未被清除（n = 0 至 4）。

20.9.3 USB 主处理

USB 主处理涉及 USB 事务的处理。将要被处理的事务类型如下。

- 控制传输中全自动处理的请求
- 控制传输中自动处理的请求
(SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, CLEAR_FEATURE)
- 控制传输的 CPUDEC 请求
- 批量传输 (IN) 的处理
- 批量传输 (OUT) 的处理
- 中断传输 (IN) 的处理

端点 n 的处理涉及数据传输的写入或读取。下面是对 PIO 的流程图。

(1) 控制传输中全自动处理的请求

因为控制传输中全自动处理的请求由硬件执行，不能由 FW 引用。因此，FW 不必为该请求执行任何特殊处理。

(2) 控制传输中自动处理的请求

(SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, CLEAR_FEATURE)

对于控制传输中自动处理的请求，例如 SET_CONFIGURATION、SET_INTERFACE、SET_FEATURE 和 CLEAR_FEATURE 请求，为其写入一个寄存器的处理由硬件自动执行，但是在设备端发出一个中断请求用来识别。如果不需要执行特殊处理，这个处理可能被忽略。

流程图如下所示。

图 20-19. 控制传输中自动处理的请求

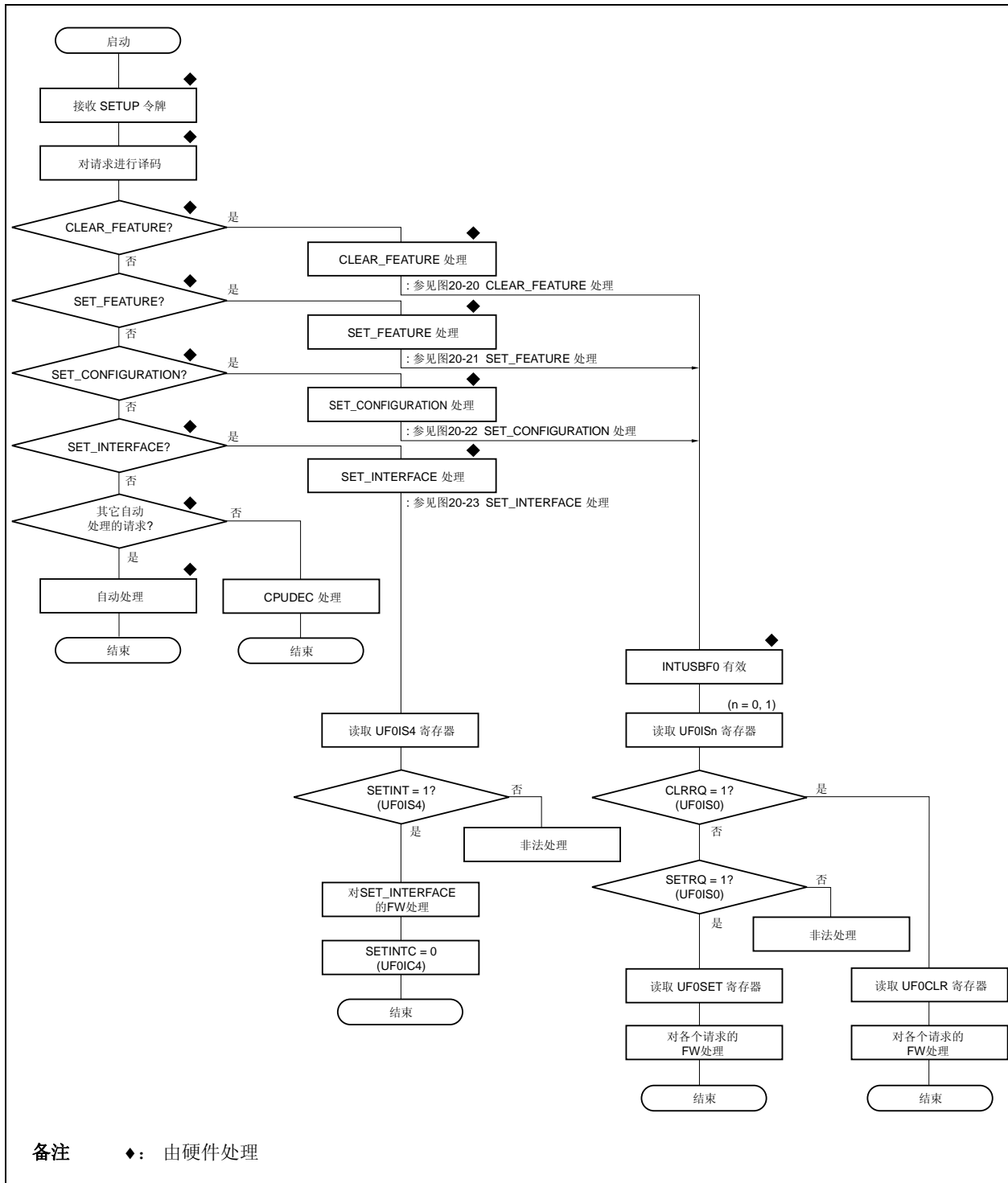


图 20-20. CLEAR_FEATURE 处理

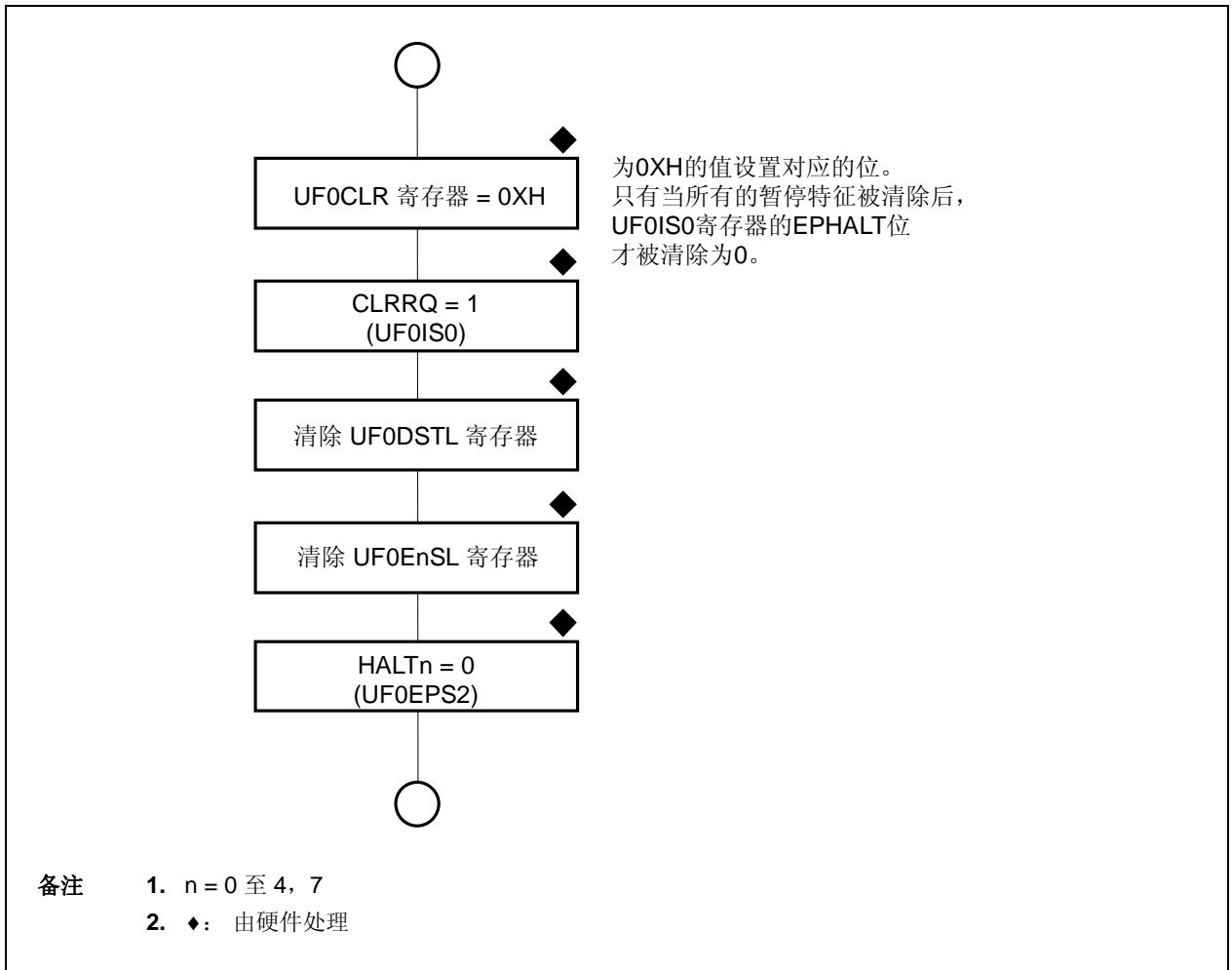


图 20-21. SET_FEATURE 处理

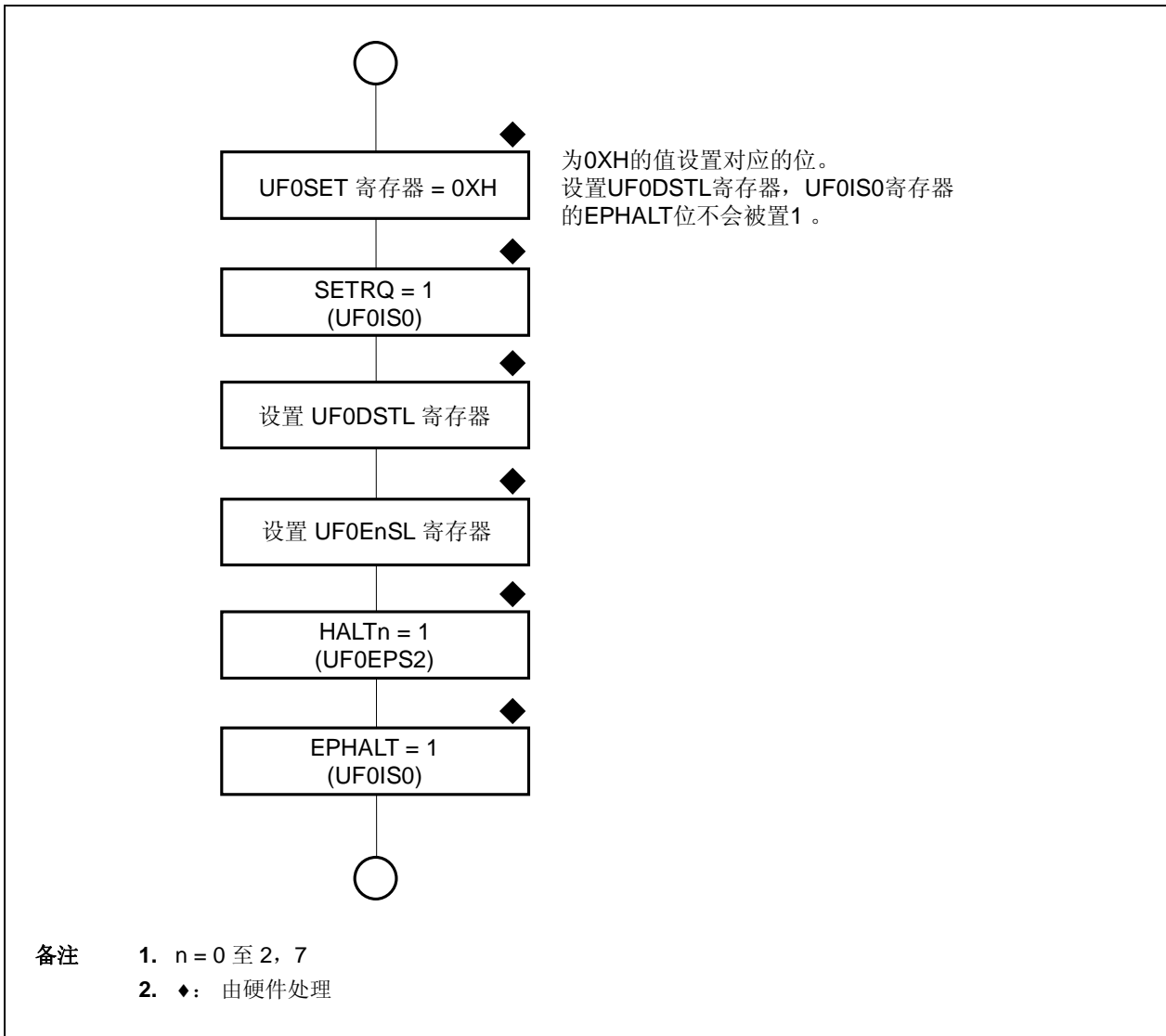


图 20-22. SET_CONFIGURATION 处理

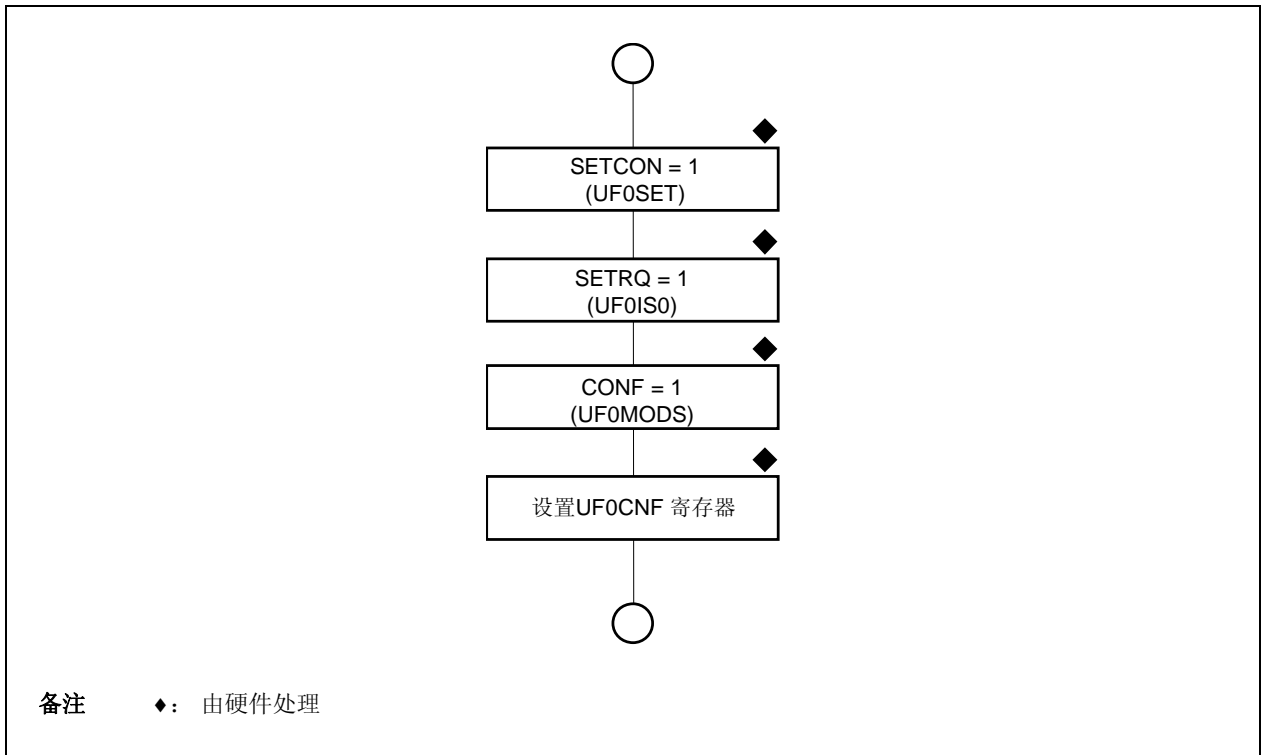
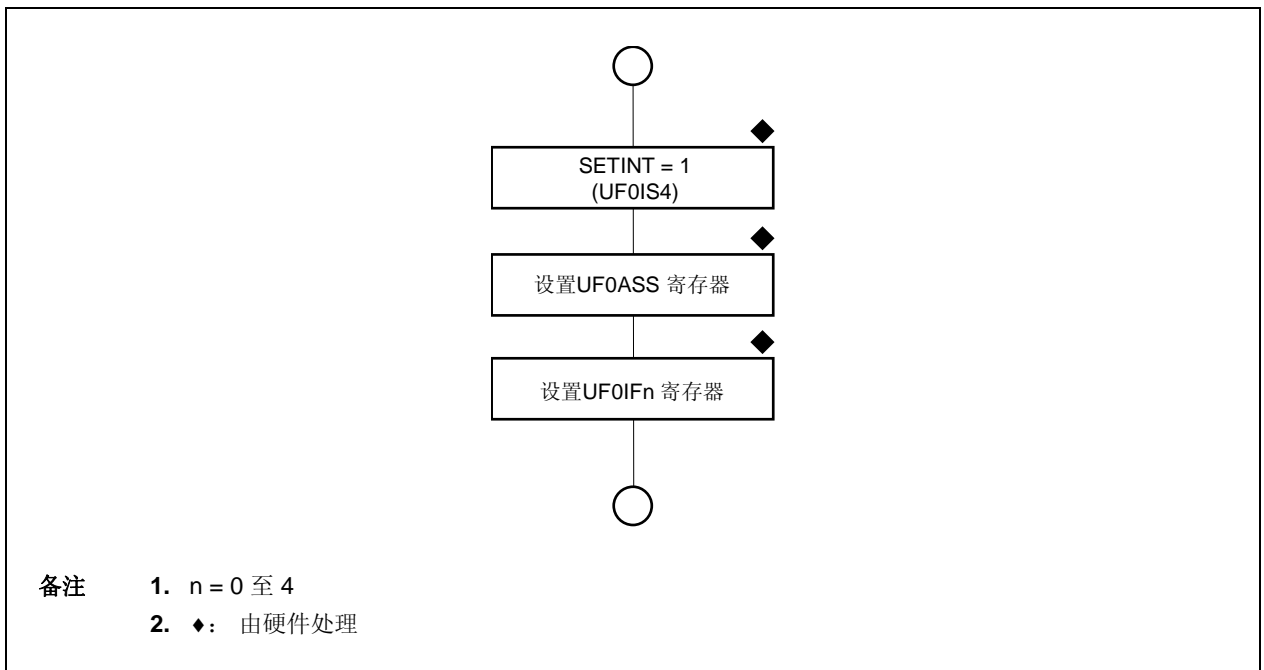


图 20-23. SET_INTERFACE 处理



(3) 控制传输的 CPUDEC 请求

CPUDEC 请求可以被分为三种处理类型：控制传输（写）、控制传输（读）和控制传输（无数据）。控制传输（写）表示一个在数据阶段使用 OUT 事务的请求（例如 SET_DESCRIPTOR）。控制传输（读）表示一个在数据阶段使用 IN 事务的请求（例如 GET_DESCRIPTOR）。控制传输（无数据）表示一个没有数据阶段的请求（例如 SET_CONFIGURATION）。

流程图如下所示。

图 20-24. 控制传输的 CPUDEC 请求 (1/12)

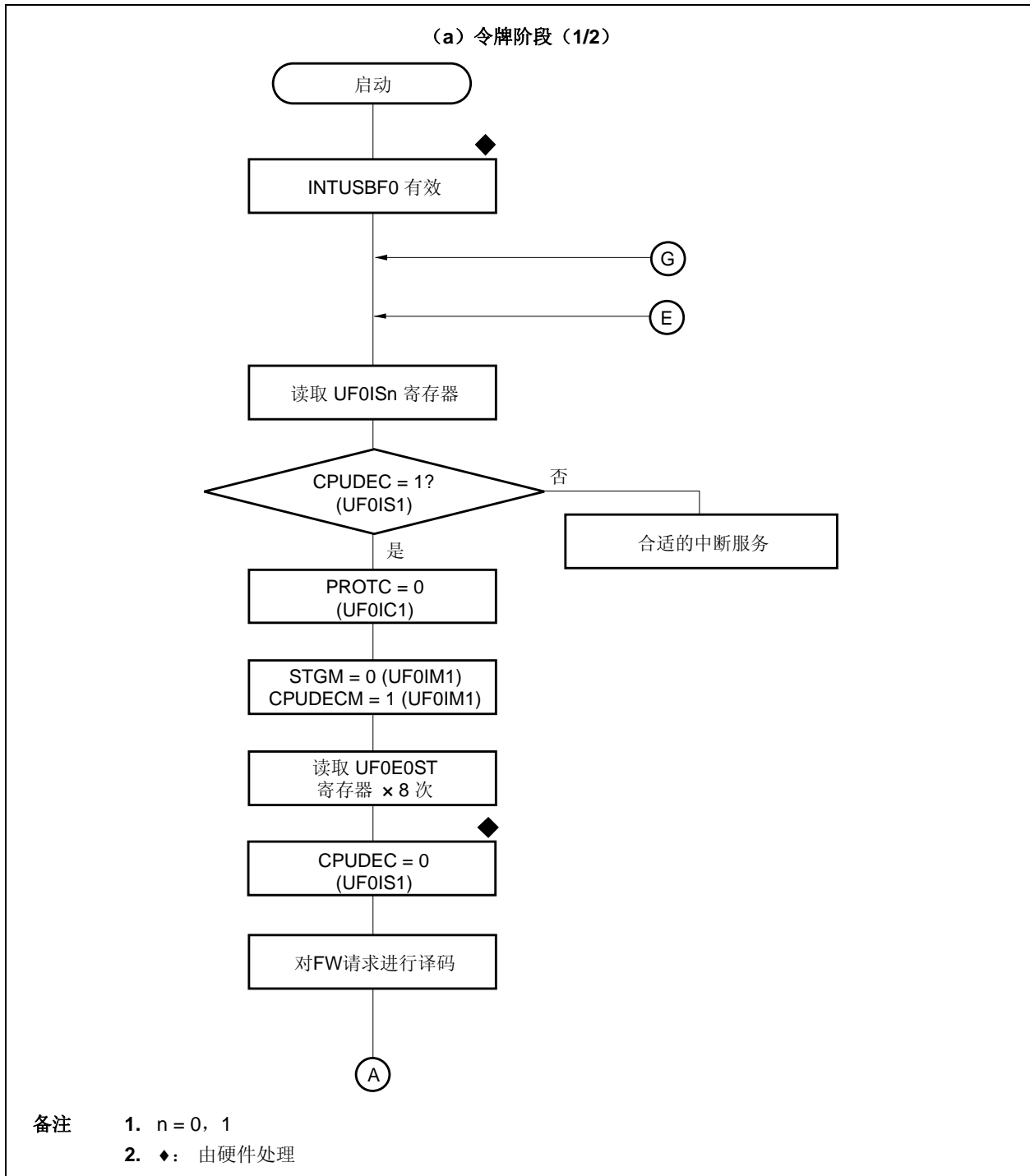


图 20-24. 控制传输的 CPUDEC 请求 (2/12)

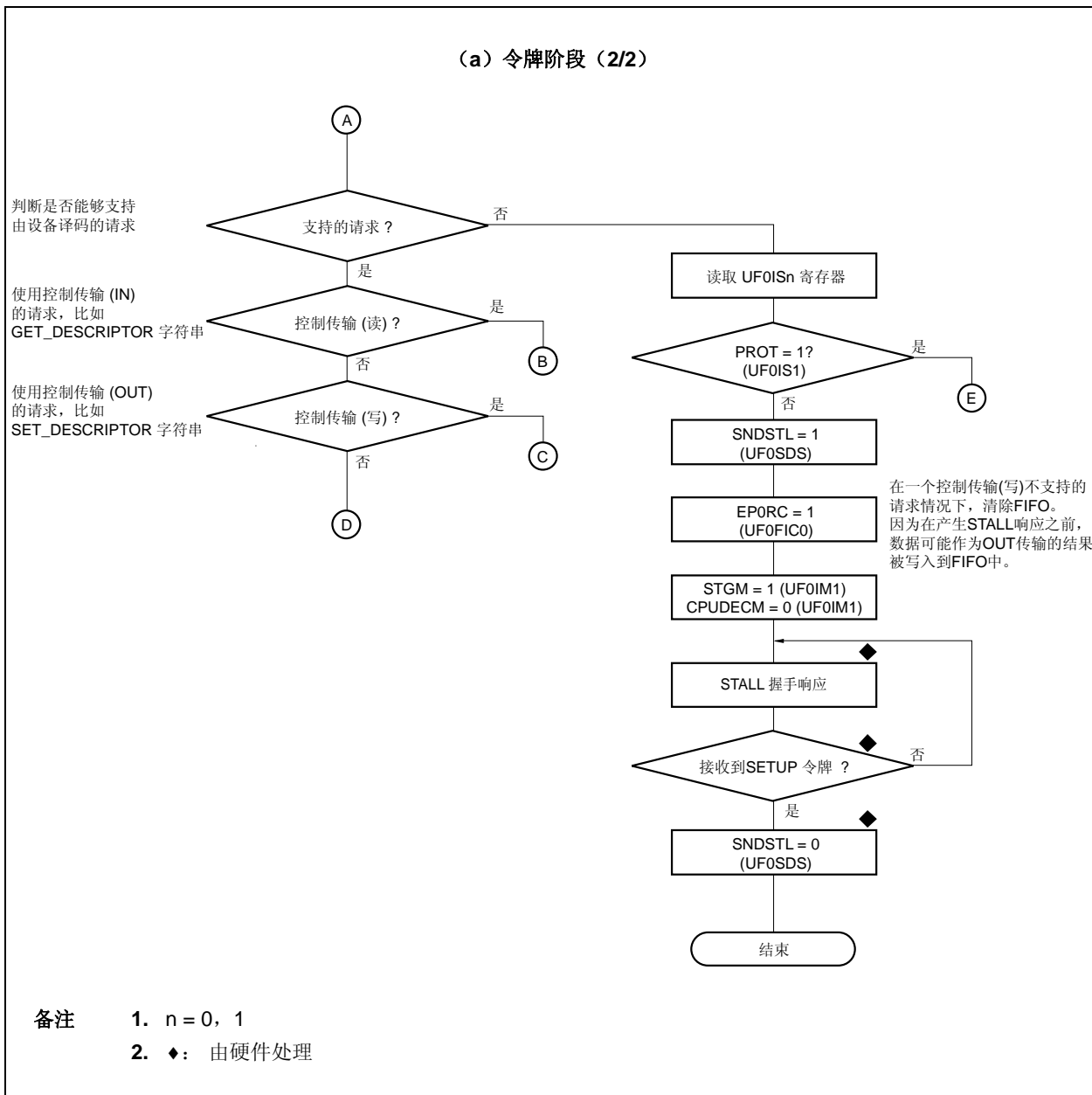


图 20-24. 控制传输的 CPUDEC 请求 (3/12)

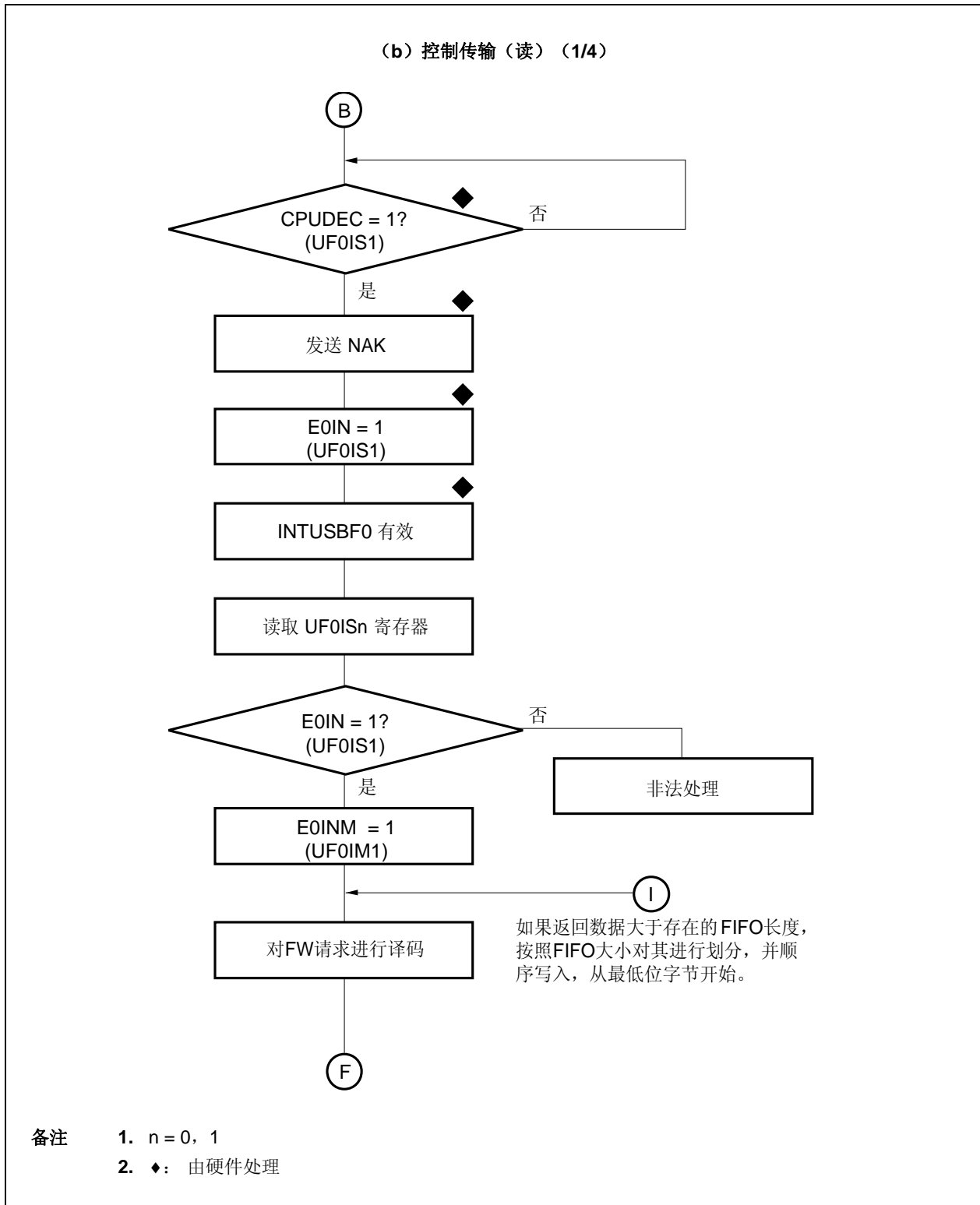


图 20-24. 控制传输的 CPUDEC 请求 (4/12)

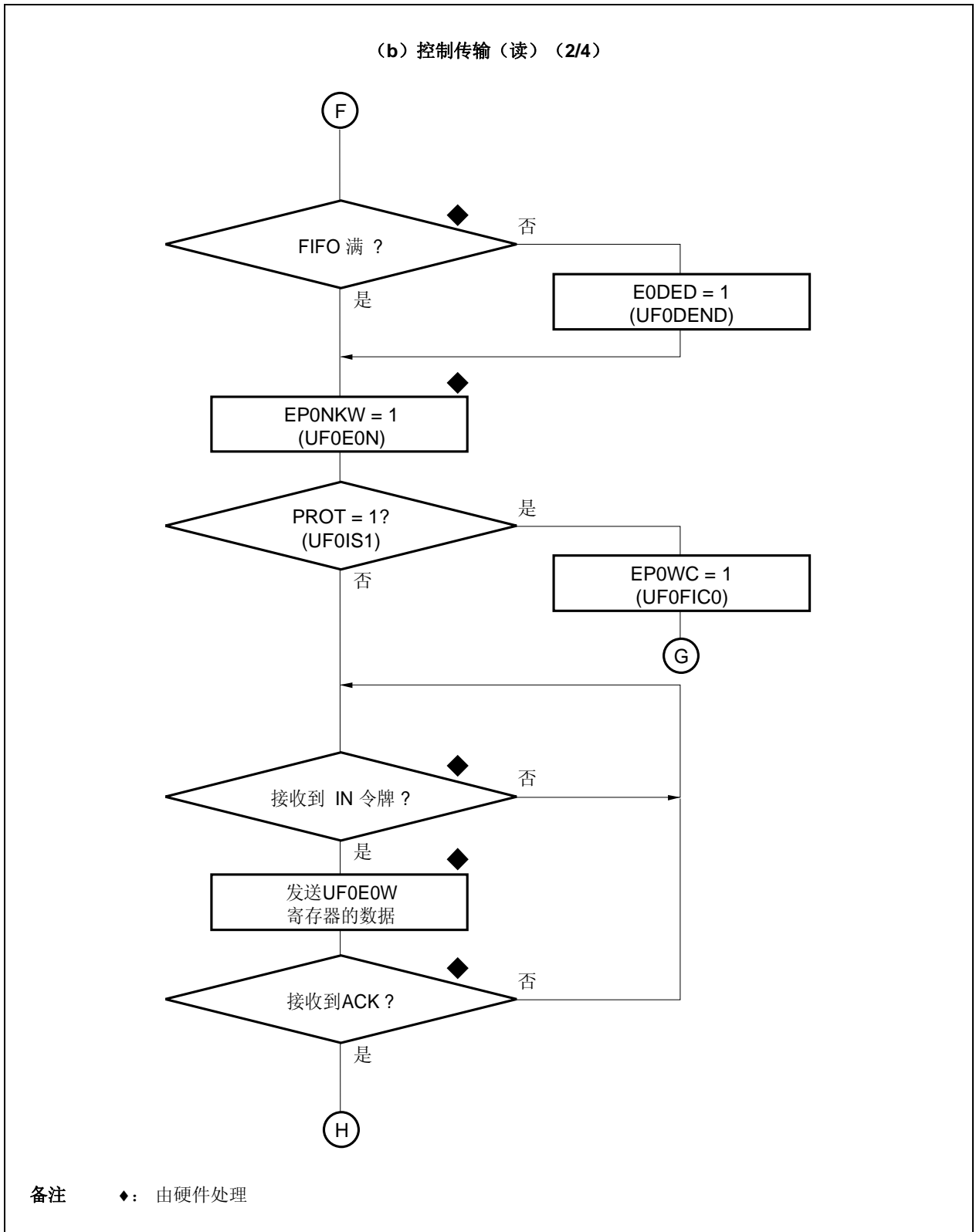


图 20-24. 控制传输的 CPUDEC 请求 (5/12)

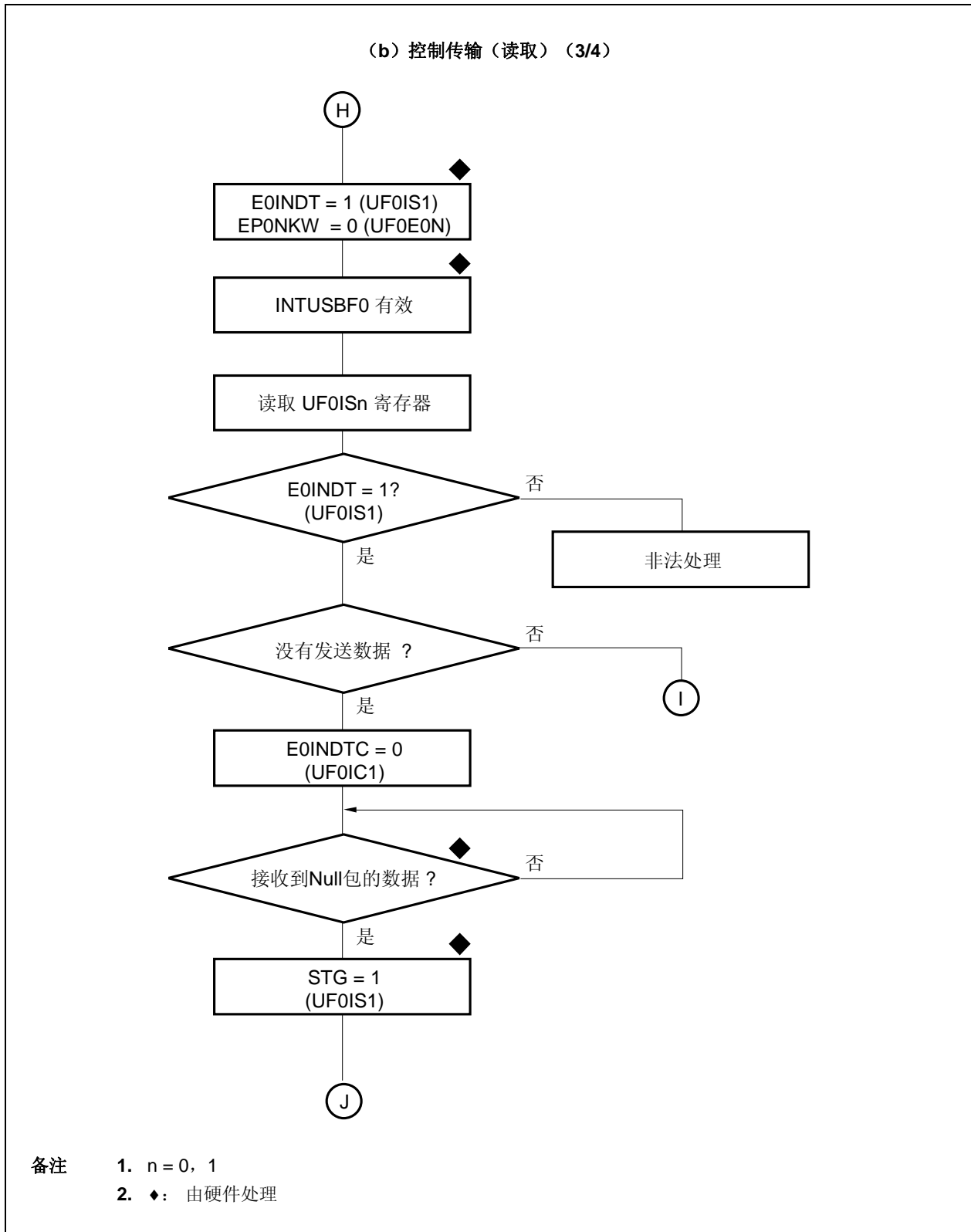


图 20-24. 控制传输的 CPUDEC 请求 (6/12)

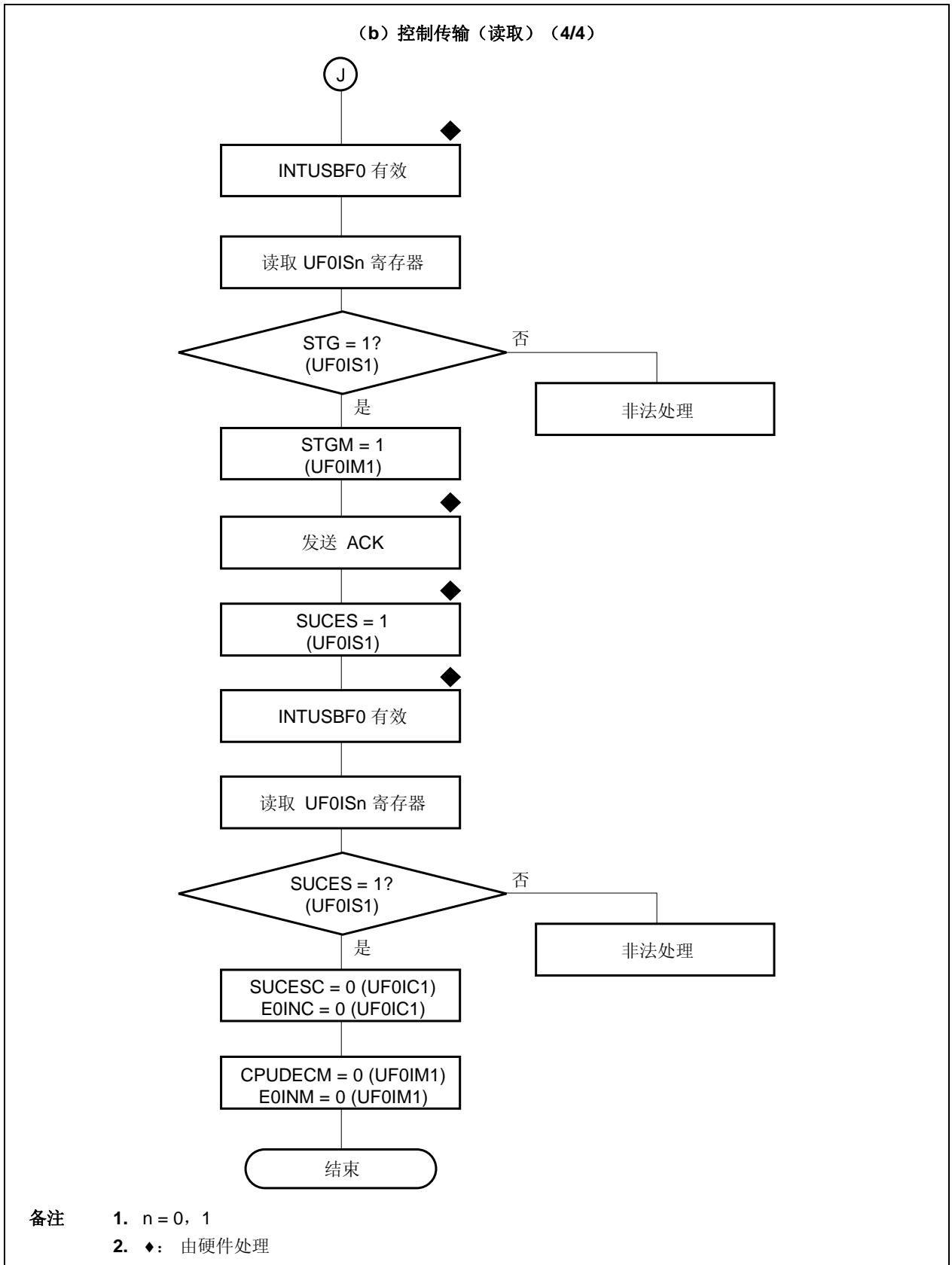


图 20-24. 控制传输的 CPUDEC 请求 (7/12)

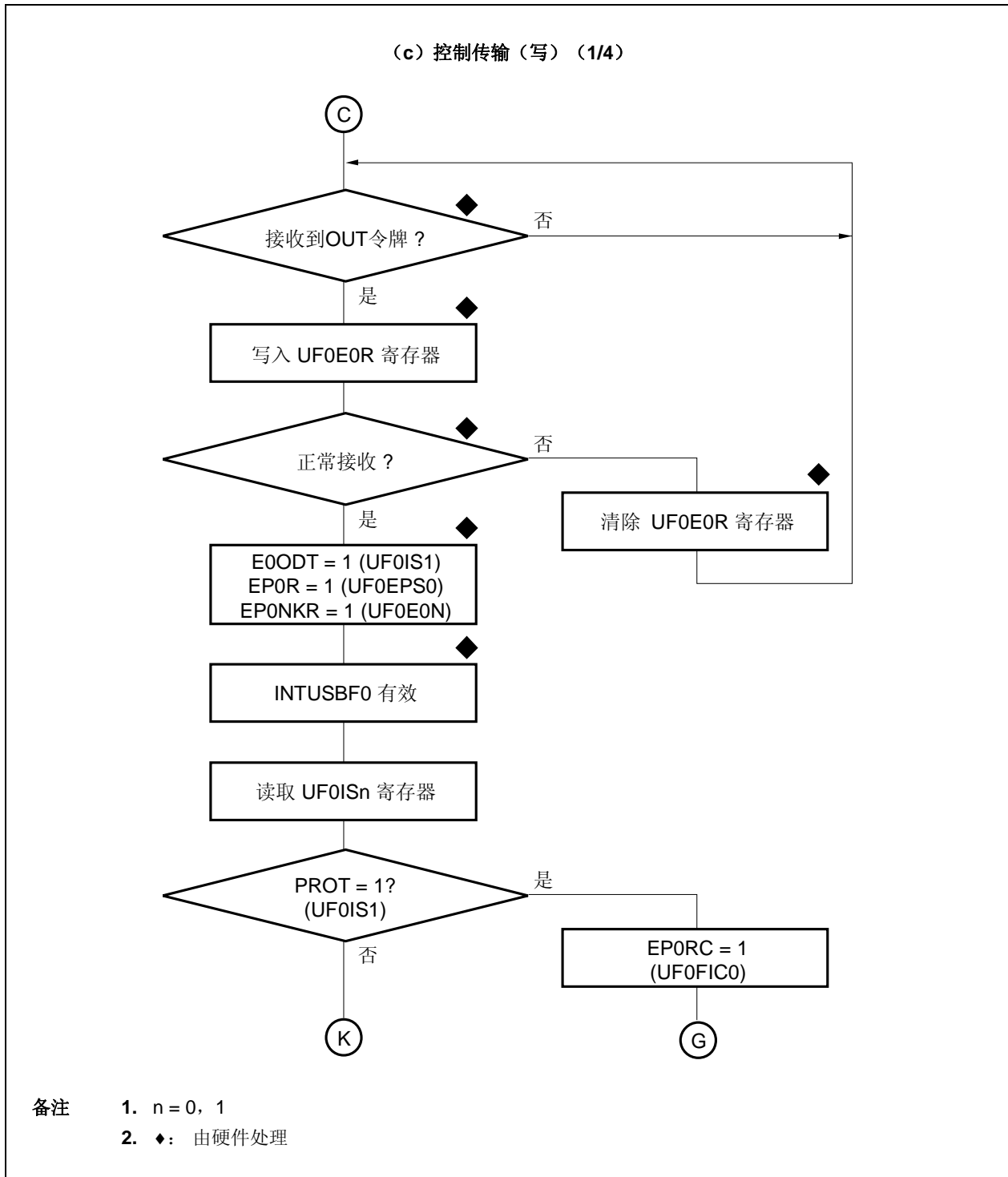


图 20-24. 控制传输的 CPUDEC 请求 (8/12)

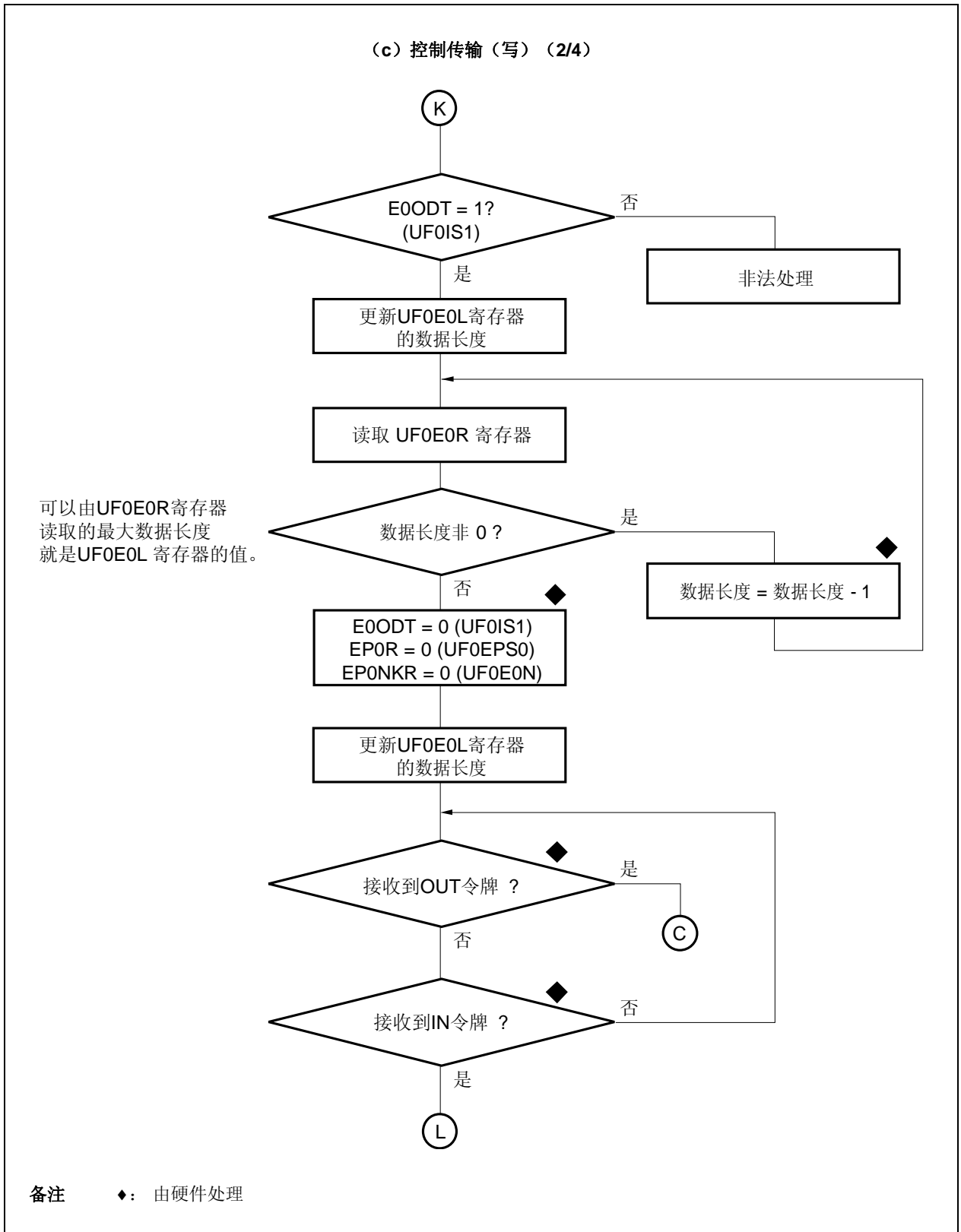


图 20-24. 控制传输的 CPUDEC 请求 (9/12)

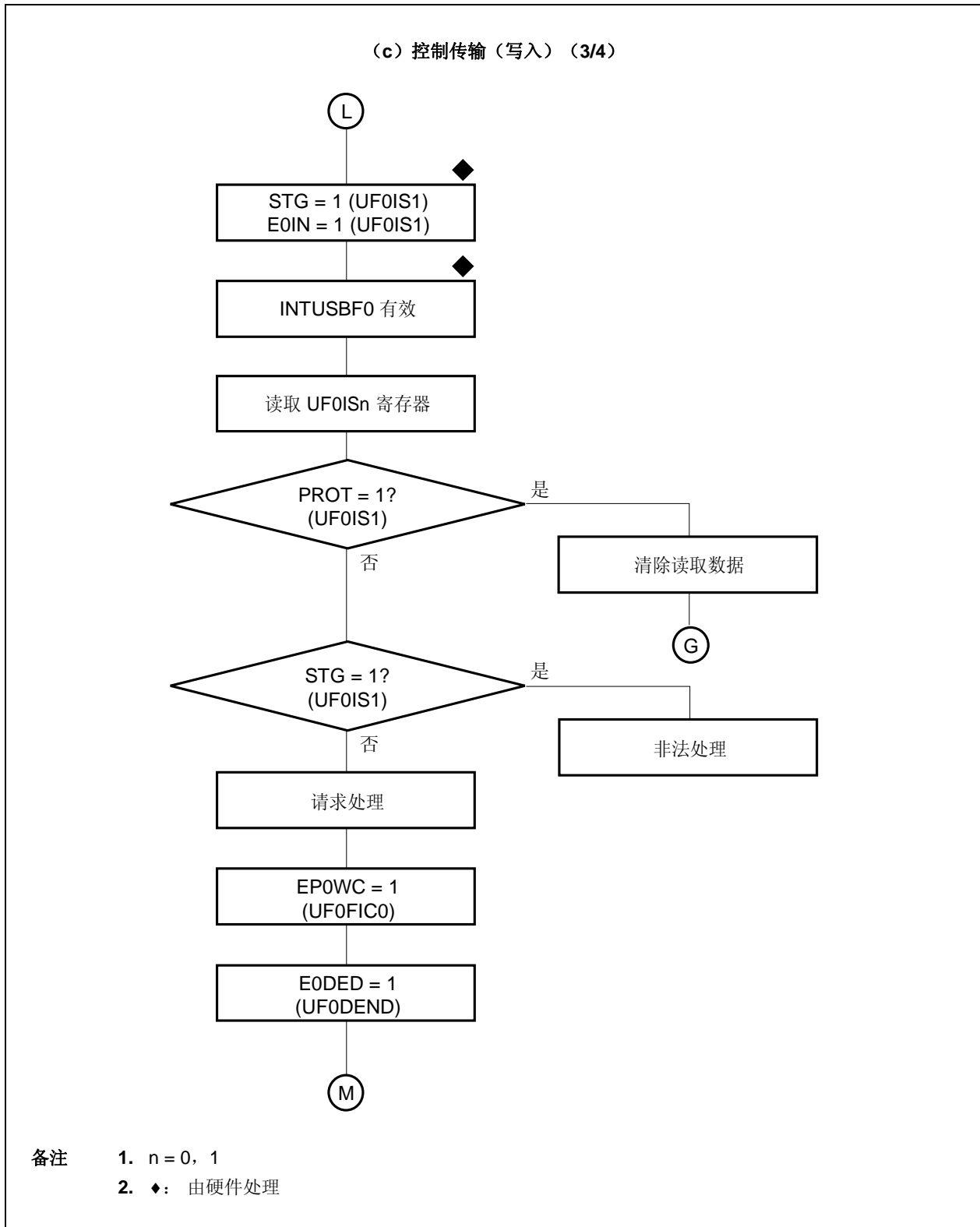


图 20-24. 控制传输的 CPUDEC 请求 (10/12)

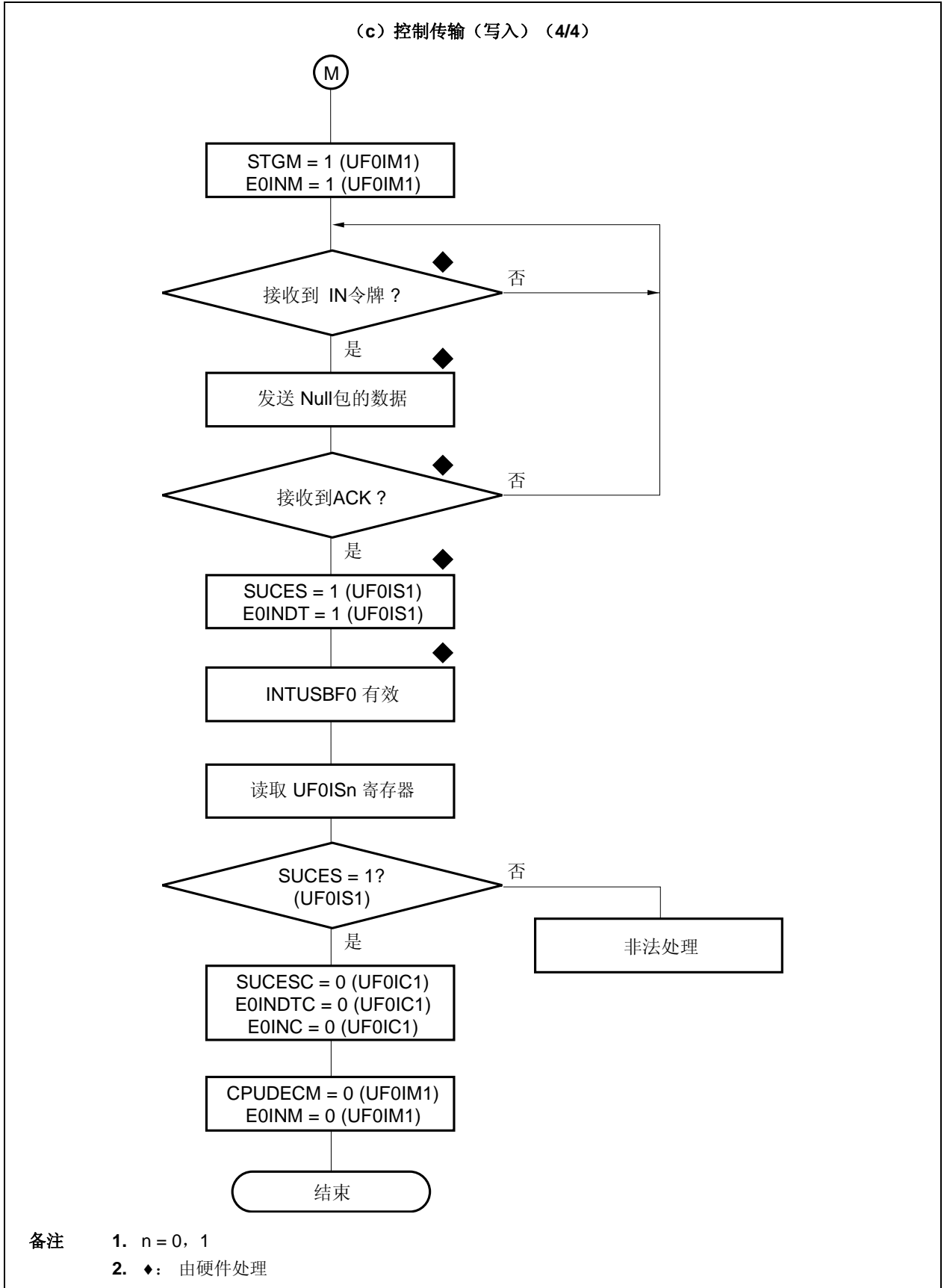


图 20-24. 控制传输的 CPUDEC 请求 (11/12)

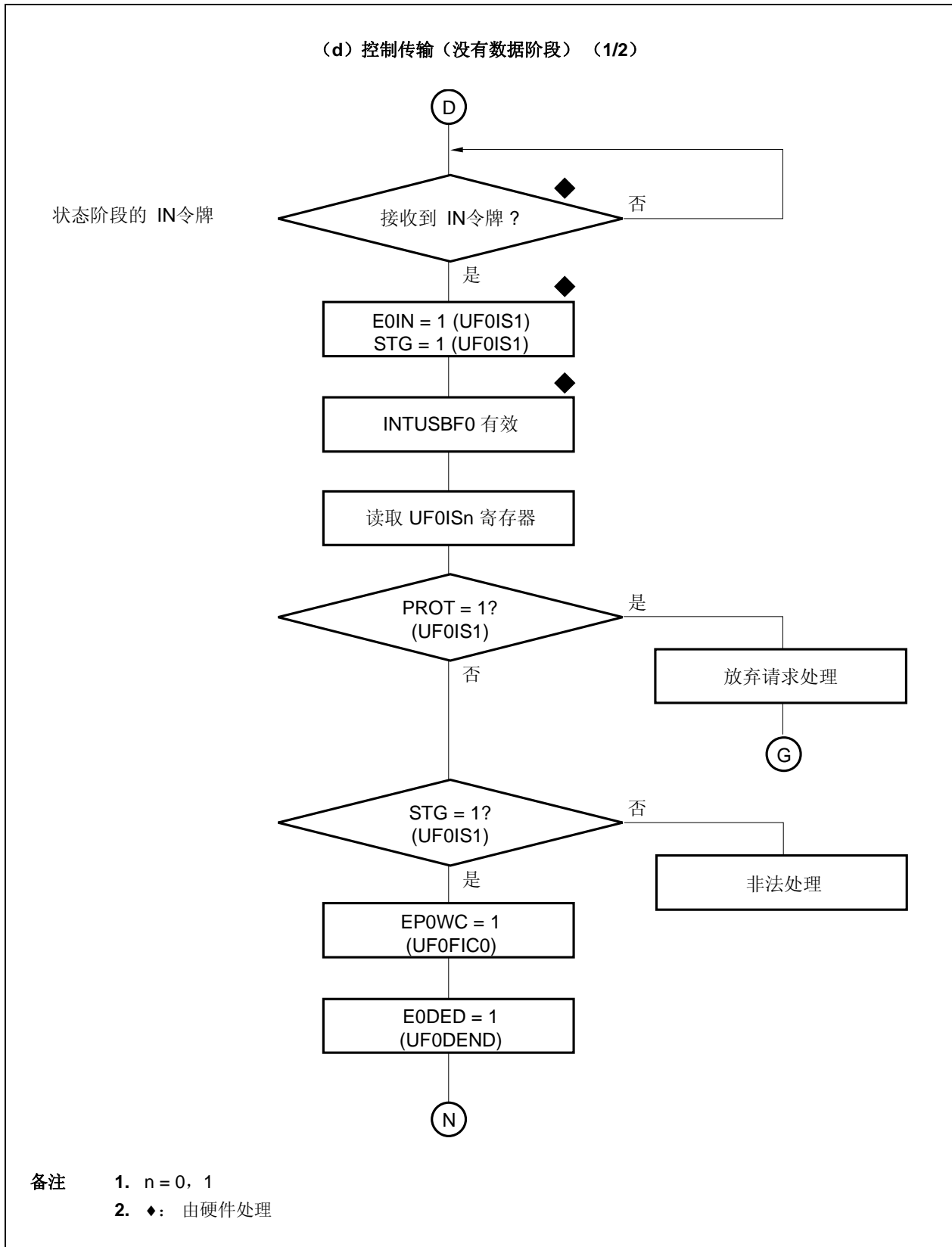
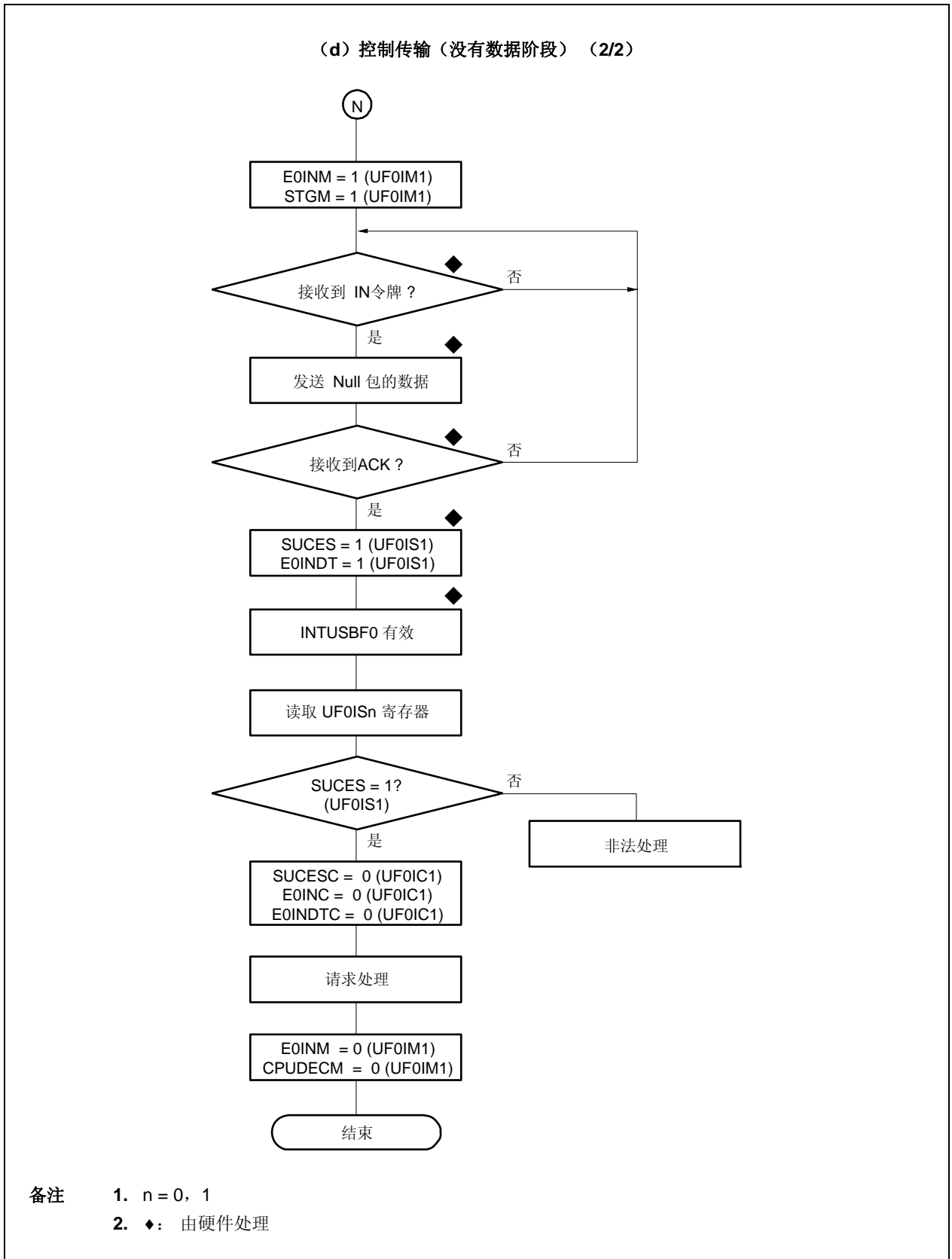


图 20-24. 控制传输的 CPUDEC 请求 (12/12)



(4) 批量传输 (IN) 的处理

批量传输 (IN) 被分配给端点 1 和端点 3。下面的流程图展示了如何控制端点 1。端点 3 可以按照同样的顺序来控制。要使用该流程图作为端点 3 的控制流程，因此，在流程图中读取端点 1 的位名称，正如端点 3 的那些位。

图 20-25. 批量传输 (IN) 的处理 (端点 1)

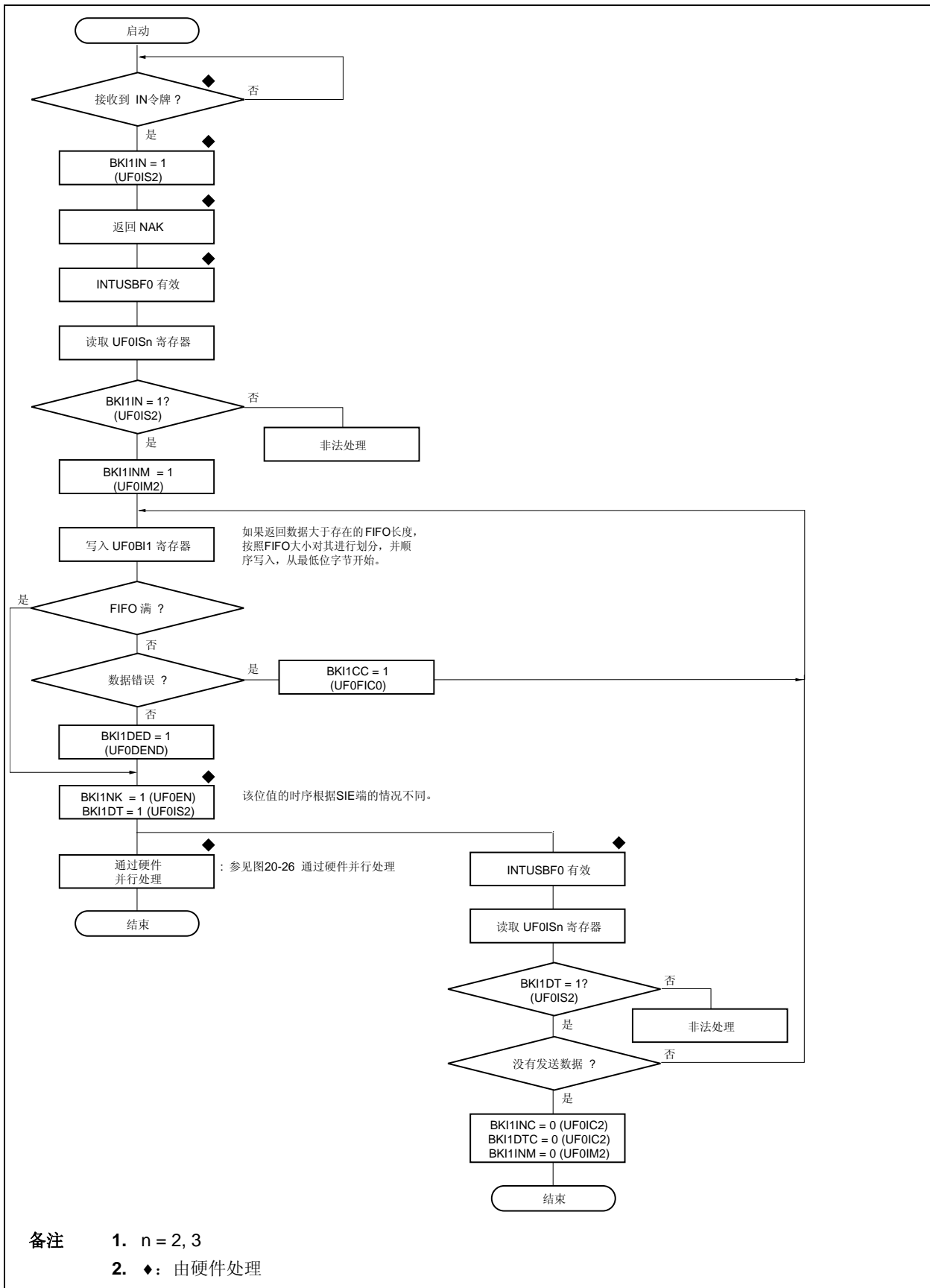
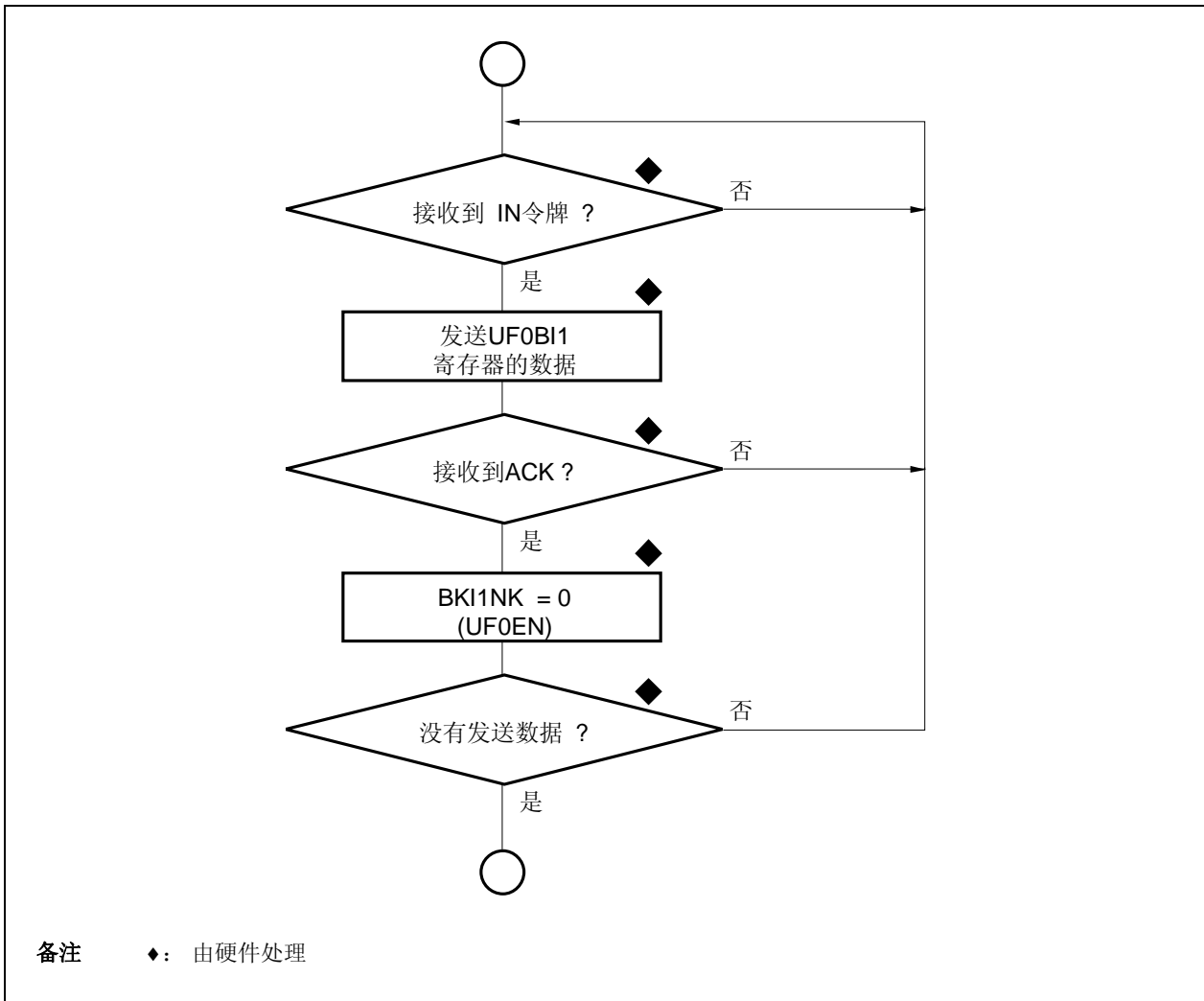


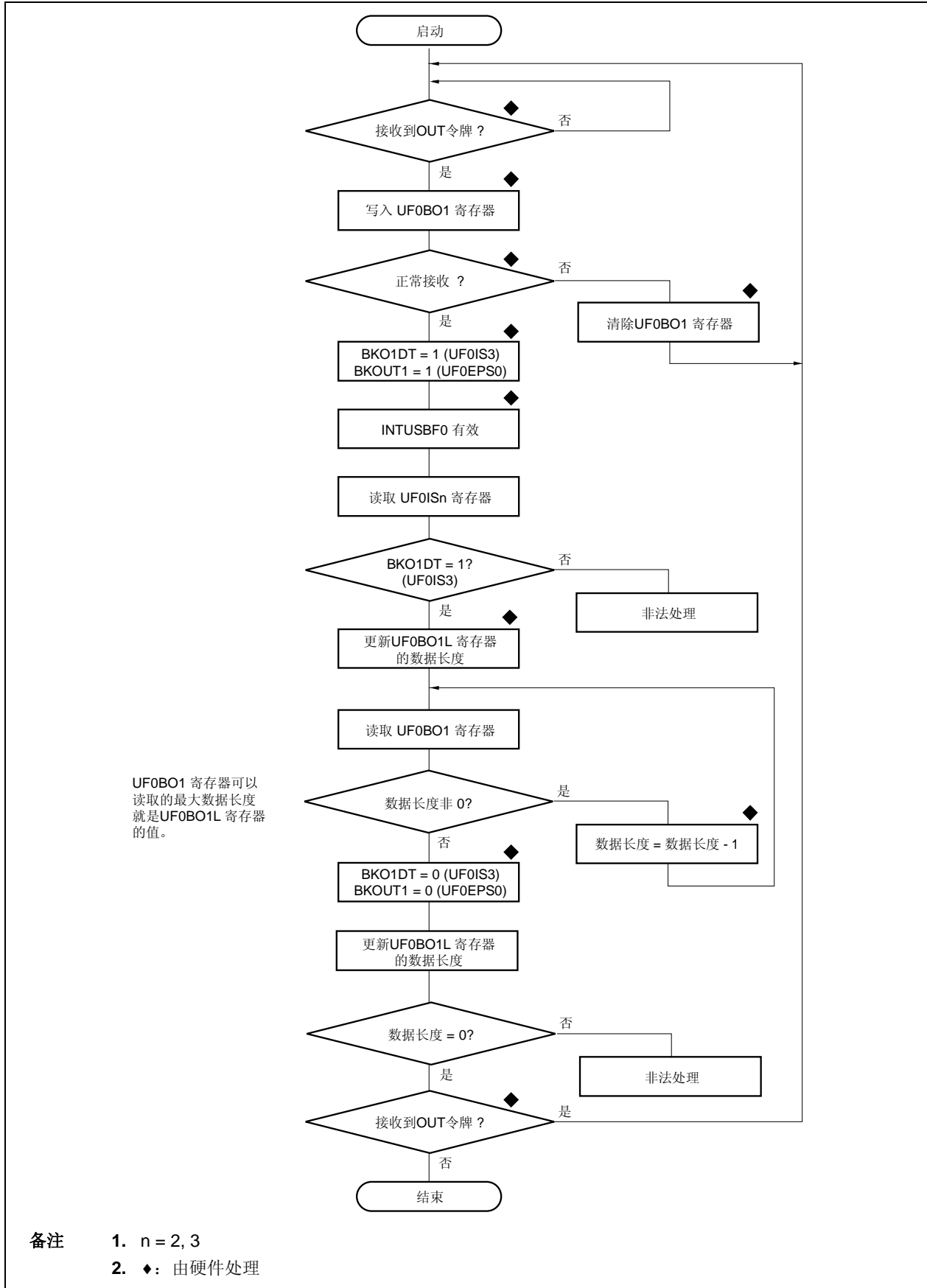
图 20-26. 通过硬件并行处理



(5) 批量传输的处理 (OUT)

批量传输 (OUT) 被分配给端点 2 和端点 4。下面的流程图展示了如何控制端点 2。端点 4 可以按照同样的顺序来控制。要使用该流程图作为端点 4 的控制流程，因此，在流程图中读取端点 2 的位名称，正如端点 4 的那些位。

图 20-27. 批量传输 (OUT) 的正常处理 (端点 2)



批量传输 (OUT) 期间, 从主机发送的数据可能多于系统的期望值。V850ES/JG3-U 和 V850ES/JH3-U 中用于批量传输 (OUT) 的端点 2 和端点 4 由两个 64 字节缓冲区组成, 于是 NAK 响应被尽可能的抑制并且数据可以从 CPU 端读取, 即使随着 USB 总线的传输速率增加, 总线端被访问的频率也增加。因此, 如果主机发送的数据数量多于系统的期望, 在最坏情况下可能自动接收到 128 字节的多余数据。在这种情况下, 当系统期望的数据的数量减少为两个包时, 将控制流程从端点 2 和端点 4 的正常处理更改为下面表示的流程。下面的流程图展示了如何控制端点 2。端点 4 可以按照同样的顺序来控制。要使用该流程图作为端点 4 的控制流程, 因此在流程图中读取端点 2 的位名称, 正如端点 4 的那些位。

图 20-28. 如果发送的数据多于系统期望的处理 (端点 2) (1/2)

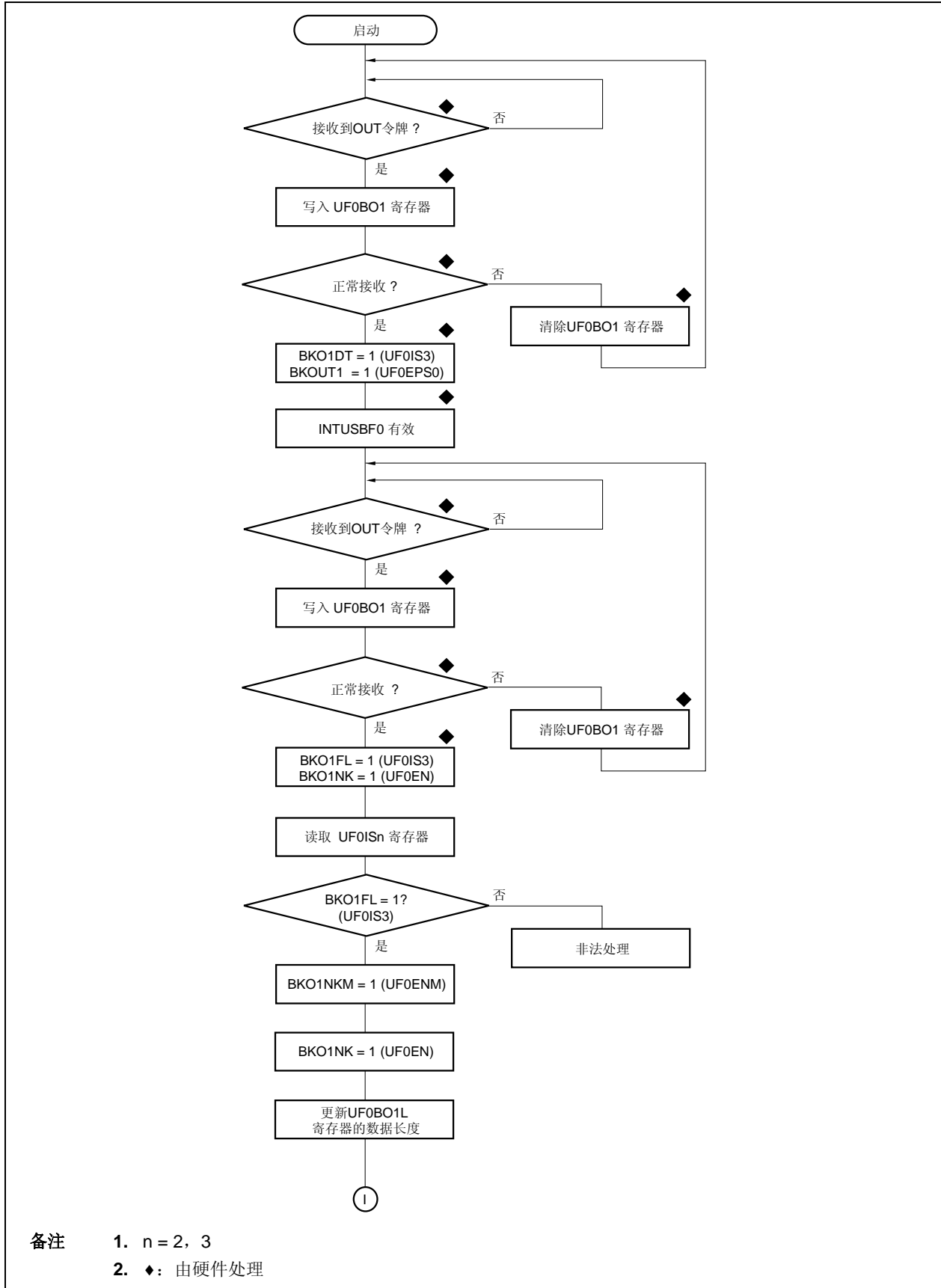
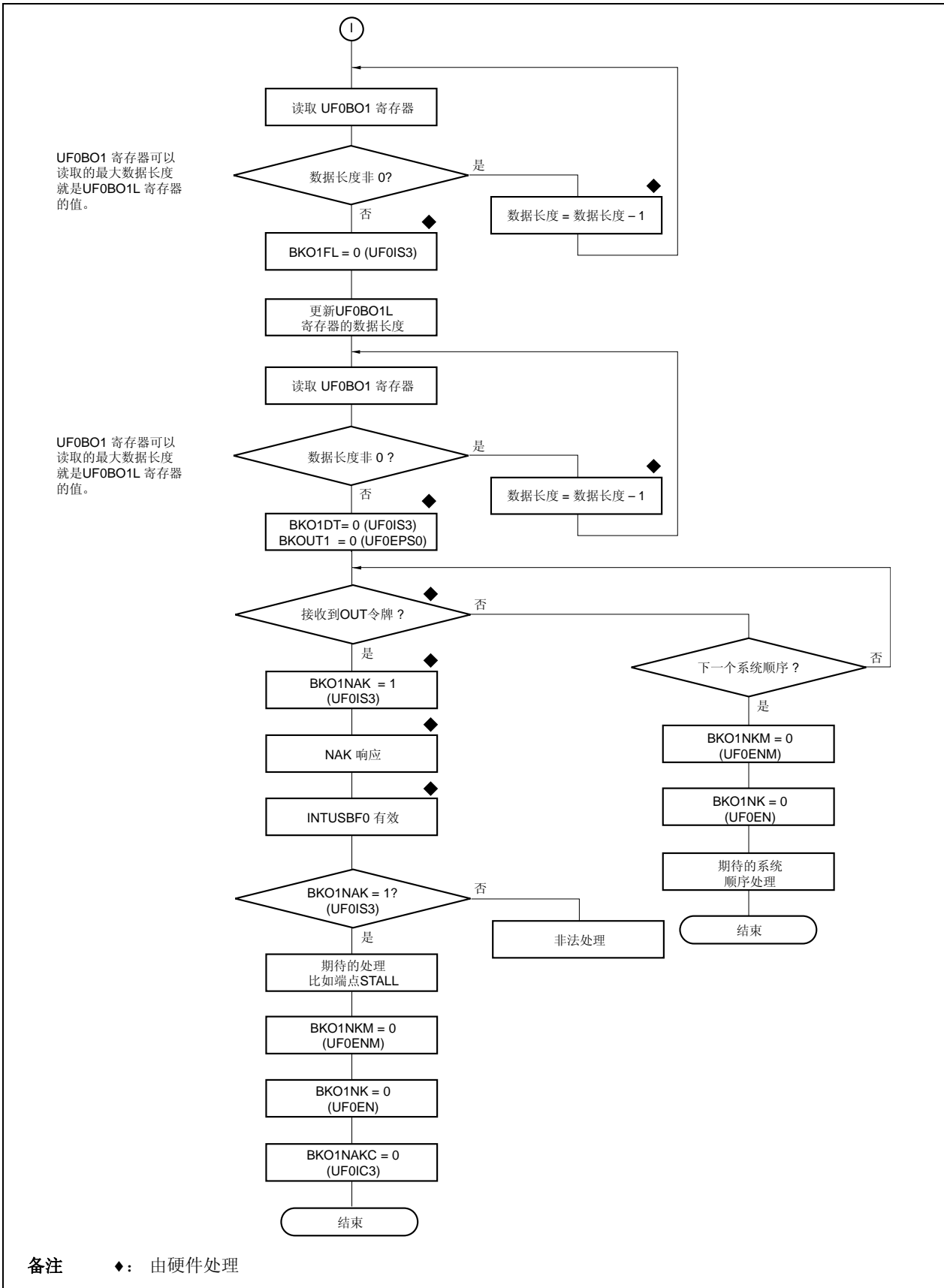


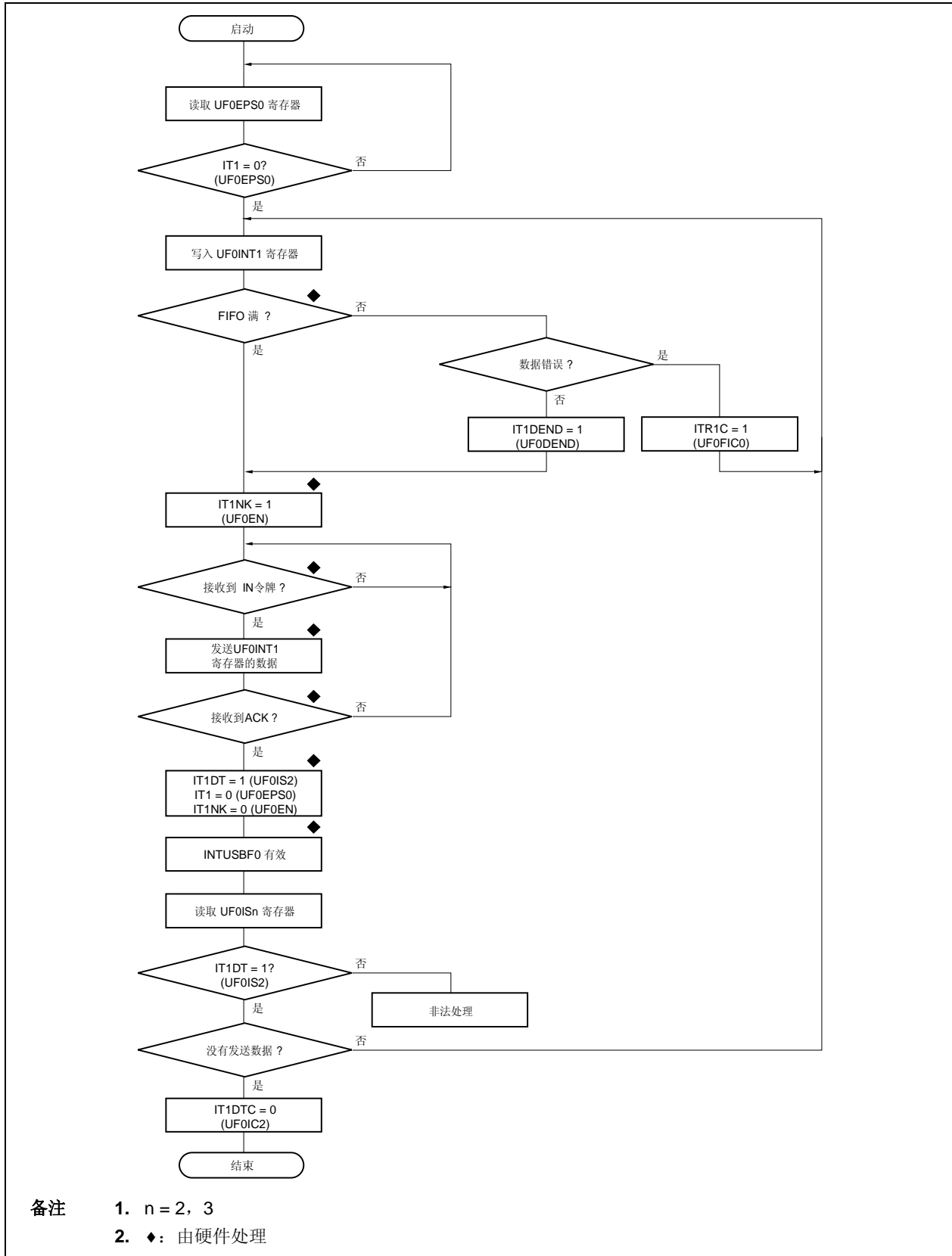
图 20-28. 如果发送的数据多于系统期望的处理 (端点 2) (2/2)



(6) 中断传输(IN)的处理

中断传输(IN)被分配给端点 7。下面展示流程图。

图 20-29. 对中断传输(IN)的处理 (端点 7)



20.9.4 挂起/恢复处理

挂起/恢复处理如何执行依赖于系统的配置。下面给出一个示例。

图 20-30. 挂起/恢复处理示例 (1/3)

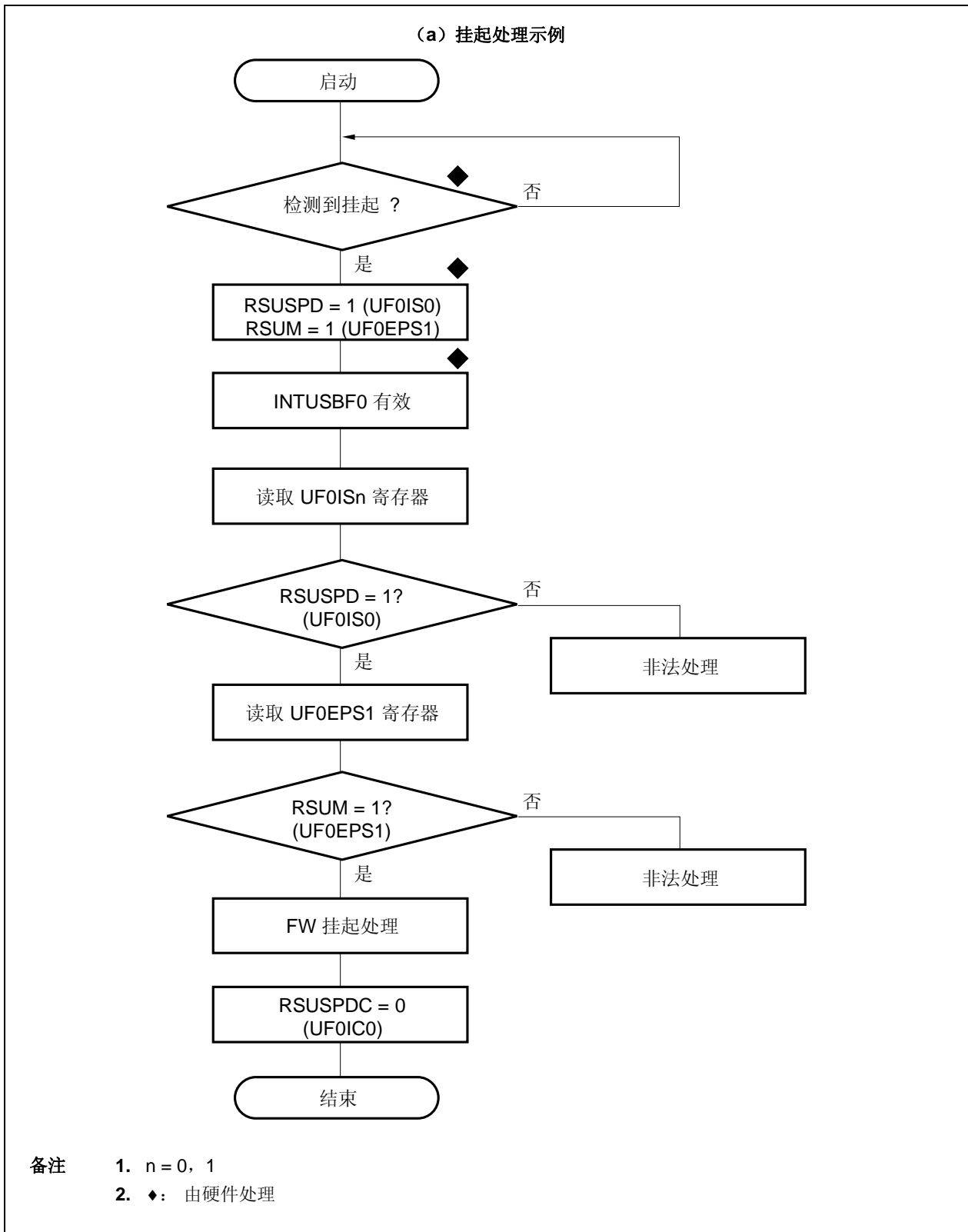


图 20-30. 挂起/恢复处理示例 (2/3)

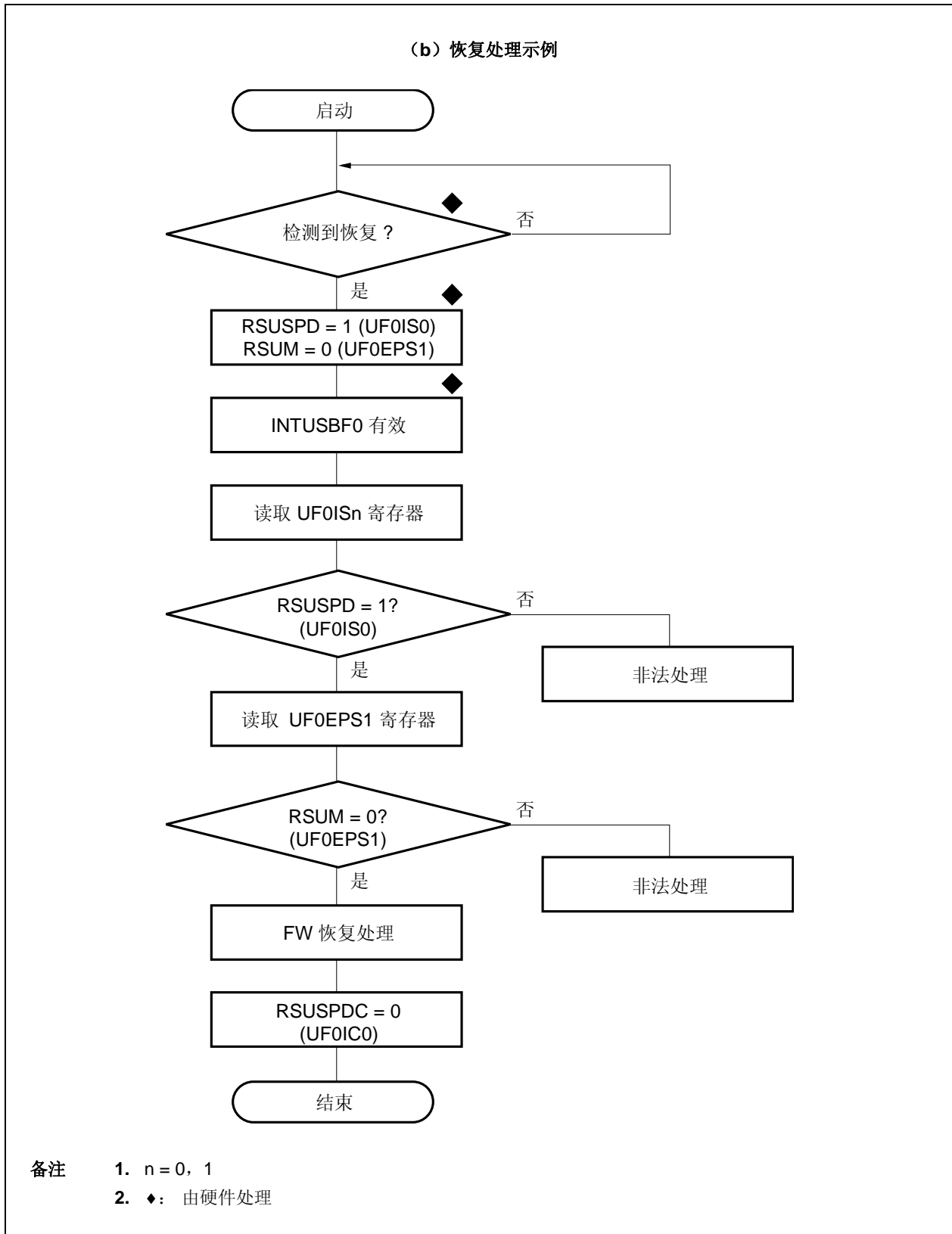
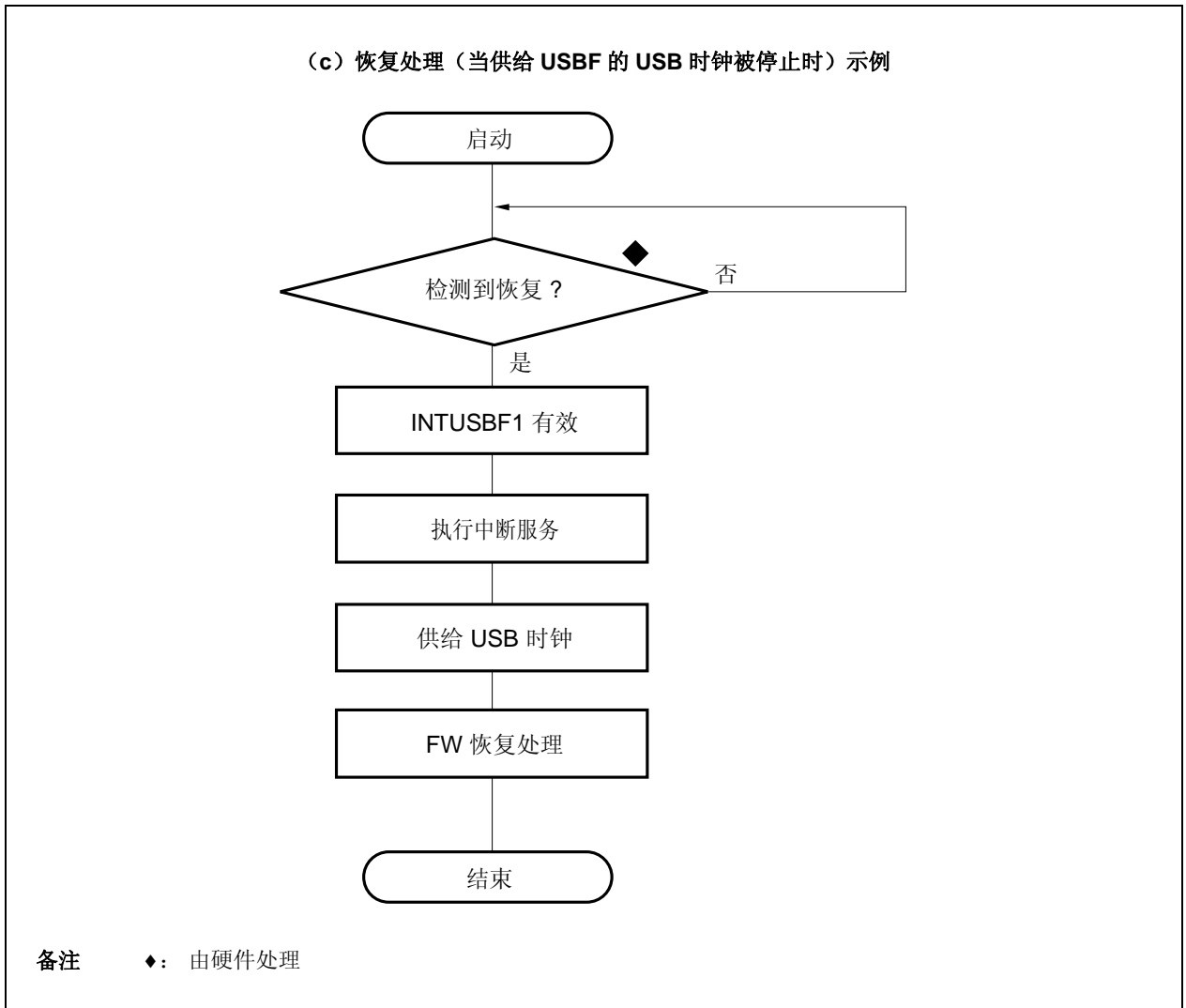


图 20-30. 挂起/恢复处理示例 (3/3)



20.9.5 电源应用后的处理

电源应用后要执行的处理依赖于系统的配置。下面给出一个示例。

图 20-31. 电源应用/电源失败后的处理示例 (1/3)

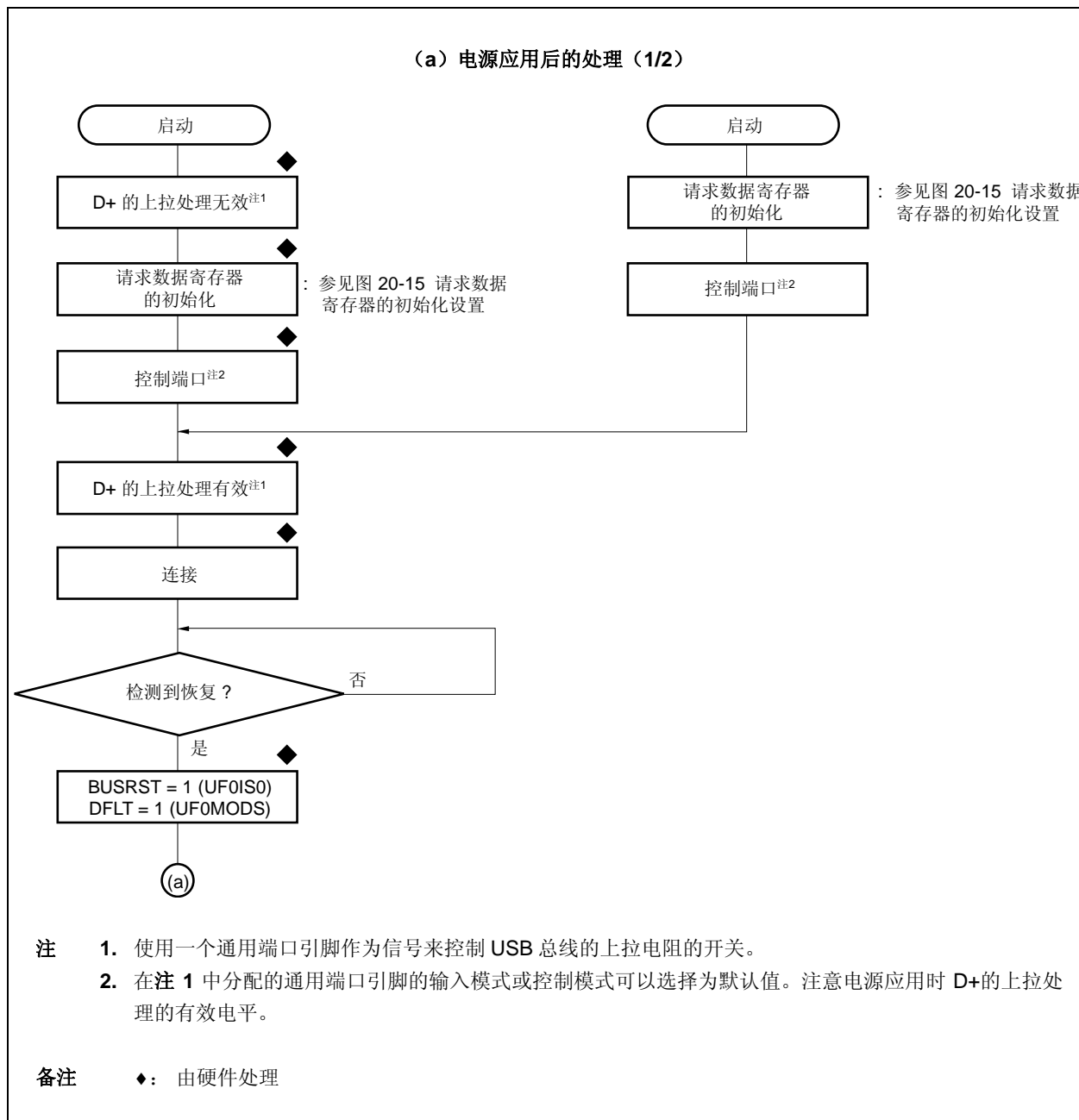


图 20-31. 电源应用/电源失败后的处理示例 (2/3)

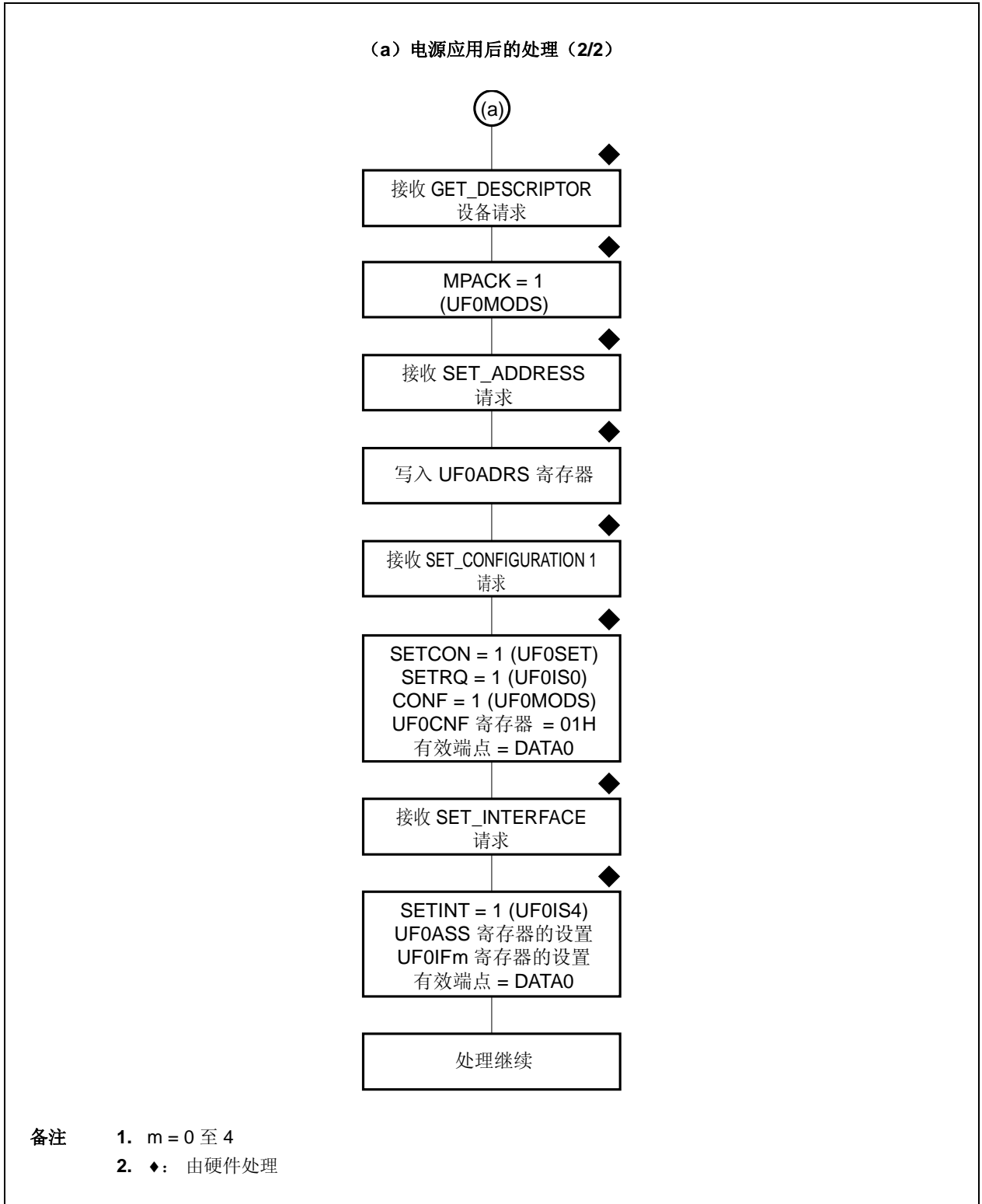
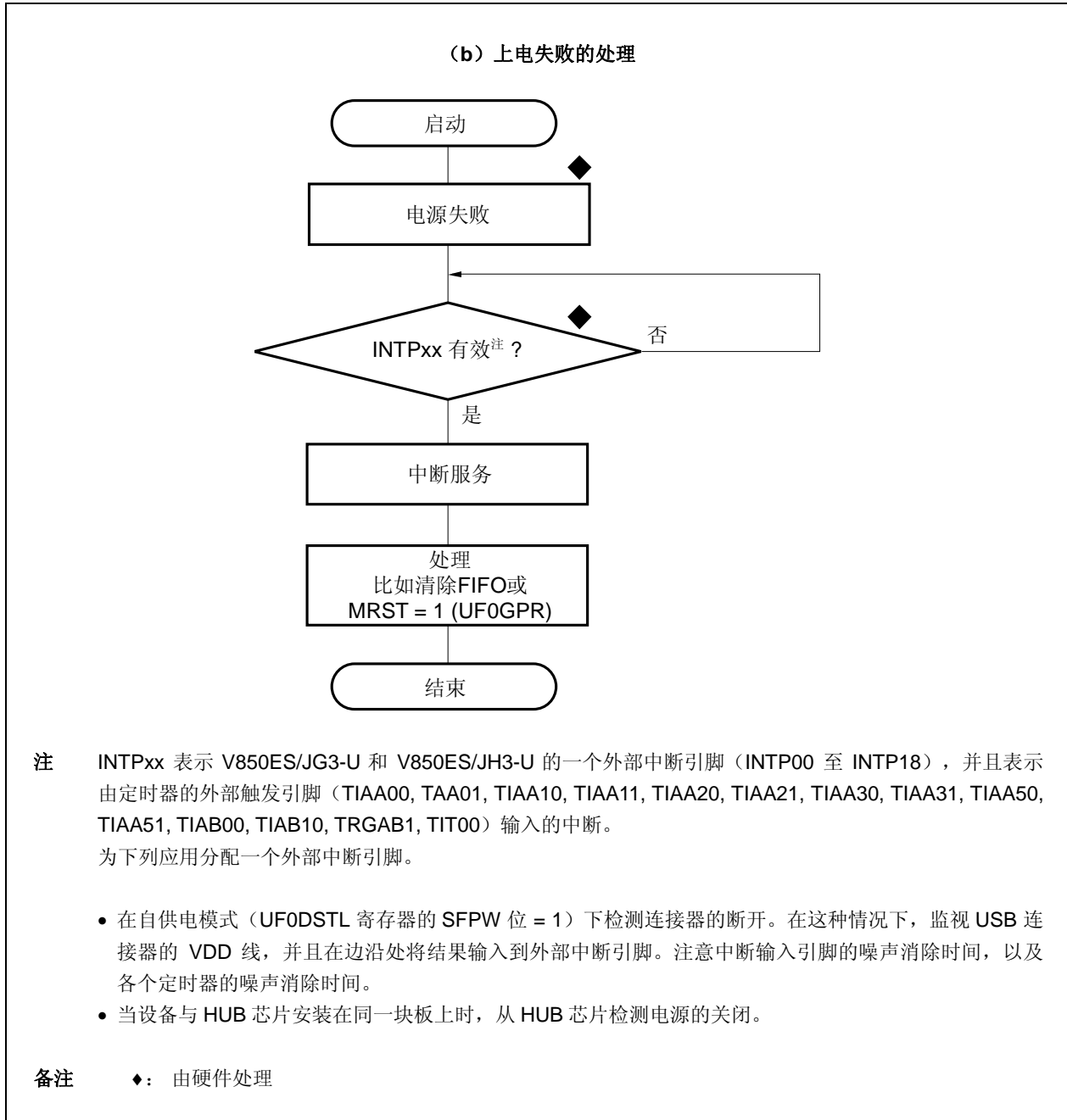


图 20-31. 电源应用/电源失败后的处理示例 (3/3)



20.9.6 在DMA模式下为批量传输 (OUT) 接收数据

批量传输 (OUT) 被分配给端点 2 和端点 4。下面的流程图展示了在使用 DMA 时如何控制端点 2。端点 4 可以按照同样的顺序来控制。要使用该流程图作为端点 4 的控制流程，因此，在流程图中读取端点 2 的位名称，正如端点 4 的那些位。下面的控制流程图展示如何通过 CPU 保持读取数据。

如果通过设置 UF0IDR 寄存器的 DQBO1MS 位为 1，接收到批量传输 (OUT) 的数据，对端点 2 的 DMA 请求变为有效，而不是产生中断请求 (INTUSBF0)。根据 UF0IDR 寄存器的 MODE_n (n = 0, 1) 位的设置，决定这个对端点 2 的 DMA 请求的操作。如果存储在 UF0BO1 寄存器中的所有数据已经被 DMA 读取，对端点 2 的 DMA 请求变为无效。在这种状态下，如果正确接收到下一个批量传输 (OUT) 的数据，对端点 2 的 DMA 请求重新变为有效。如果接收到的批量传输 (OUT) 的数据长度等于或小于 FIFO 大小，只要 DMA 读取数据结束，则立即发布一个 Short 中断请求且 INTUSBF0 (EP2_ENDINT) 信号变为有效。要使用 DMA 再次读取数据，再次设置 DQBO1MS 位为 1。如果因为对端点 2 的 DMA 结束信号完成 DMA 操作，UF0IDR 寄存器的 DQBO1MS 位被清除为 0，且对端点 2 的 DMA 请求变为无效，同时发出 DMA_END 中断请求。如果此时 UF0BO1 寄存器中还保持有数据，再次设置 UF0IDR 寄存器的 DQBO1MS 位，来重新启动 DMA。但是，批量传输 (OUT) 的数据长度总是等于或小于 FIFO 大小。于是，只要 DMA 读取数据，立即发布一个 Short 中断请求且 INTUSBF0 (EP2_ENDINT) 信号变为有效，DQBO1MS 位被清除，对端点 2 的 DMA 请求变为无效。

- 注意事项**
1. 只要还有数据需要传输，在需求模式下 (UF0IDR 寄存器的 MODE1 位和 MODE0 位 = 10)，对端点 n (n = 2, 4) 的 DMA 请求信号就变为有效。
 2. 只要还有数据需要传输，在单次模式下 (UF0IDR 寄存器的 MODE1 位和 MODE0 位 = 0X (X: 无需理会))，对端点 n (n = 2, 4) 的 DMA 请求信号就变为有效。但是每次传输完一个字节后，该信号变为无效。重复这样的操作，直到没有数据需要传输为止。

图 20-32. 通过批量传输 (OUT) 进行 DMA 处理(1/3)

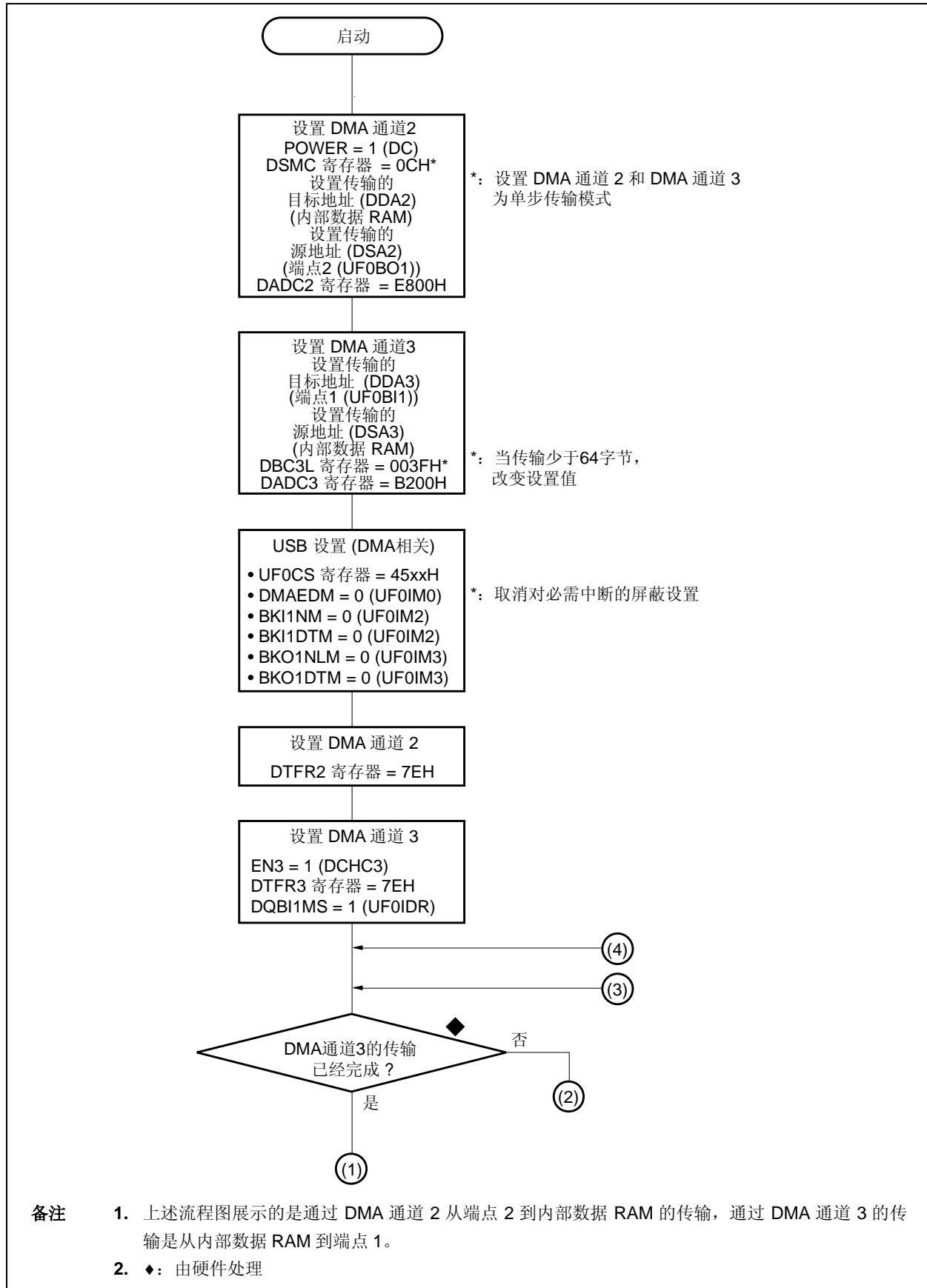


图 20-32. 通过批量传输 (OUT) 进行 DMA 处理(2/3)

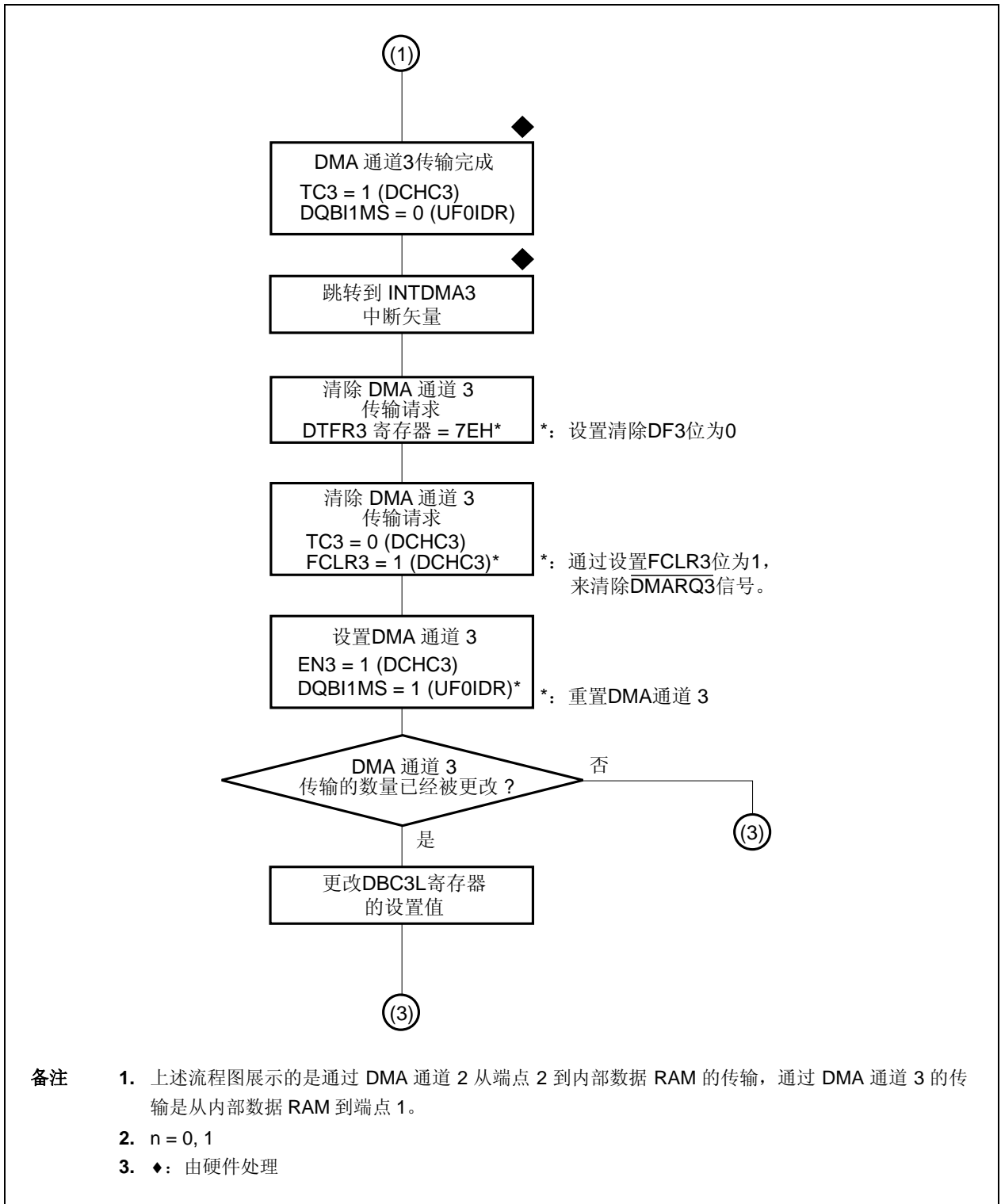
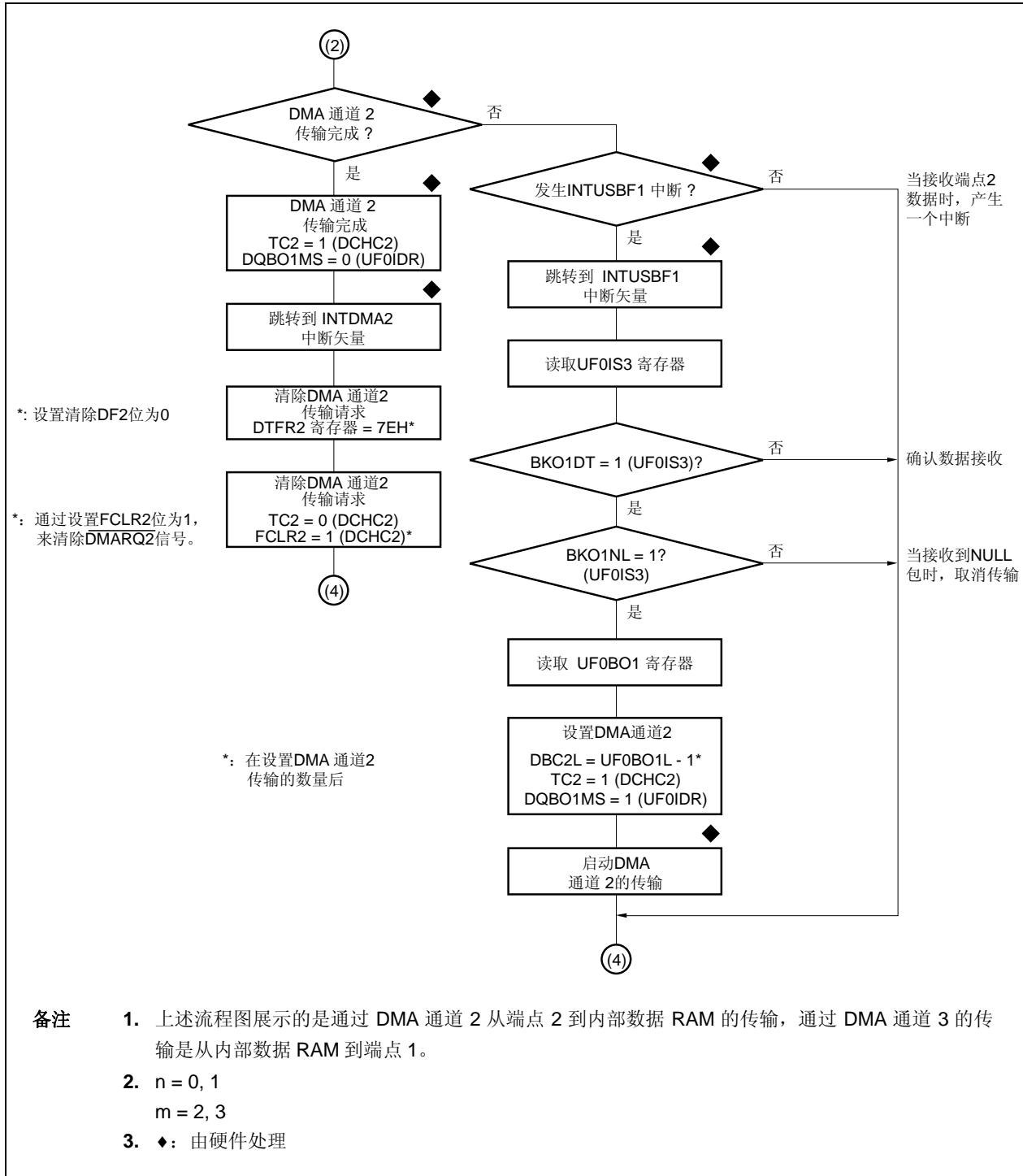


图 20-32. 通过批量传输 (OUT) 进行 DMA 处理(3/3)



20.9.7 在DMA模式下为批量传输 (IN) 发送数据

批量传输 (IN) 被分配给端点 1 和端点 3。下面的流程图展示了在使用 DMA 时如何控制端点 1。端点 3 可以按照同样的顺序来控制。要使用该流程图作为端点 3 的控制流程，因此，在流程图中读取端点 1 的位名称，正如端点 3 的那些位。

如果通过设置 UF0IDR 寄存器的 DQBI1MS 位为 1，可以写入批量传输 (IN) 所需的数据，对端点 1 的 DMA 请求变为有效，而不是产生中断请求 (INTUSBF0)。根据 UF0IDR 寄存器的 MODE_n (n = 0, 1) 位的设置，决定这个对端点 1 的 DMA 请求的操作。如果所有可以写入 UF0BO1 寄存器的数据已经被 DMA 写入，对端点 1 的 DMA 请求变为无效。在这种状态下，进行 FIFO 的切换操作，如果可以写入批量传输 (IN) 所需的数据，对端点 1 的 DMA 请求重新变为有效。如果即使因为 DMA 传输导致 FIFO 占满，却没有执行 FIFO 的自动切换操作，除非 UF0DEND 寄存器的 BKI1T 位被置为 1。因此，请确保设置 UF0DEND 寄存器的 BKI1DED 位为 1 来传输数据。如果因为对端点 1 的 DMA 结束信号完成 DMA 操作，UF0IDR 寄存器的 DQBI1MS 位被清除为 0，且对端点 1 的 DMA 请求变为无效，同时发出 DMA_END 中断请求。要在 FIFO 未满时发送一个短包，设置 UF0DEND 寄存器的 BKI1DED 位为 1。

- 注意事项**
1. 只要还有数据需要传输，在需求模式下 (UF0IDR 寄存器的 MODE1 位和 MODE0 位 = 10)，对端点 n (n = 1, 3) 的 DMA 请求信号就变为有效。
 3. 只要还有数据需要传输，在单次模式下 (UF0IDR 寄存器的 MODE1 位和 MODE0 位 = 0X (X: 无需理会))，对端点 n (n = 1, 3) 的 DMA 请求信号就变为有效。但是每次传输完一个字节后，该信号变为无效。重复这样的操作，直到没有数据需要传输为止。

图 20-33. 通过批量传输 (IN) 进行 DMA 处理 (1/4)

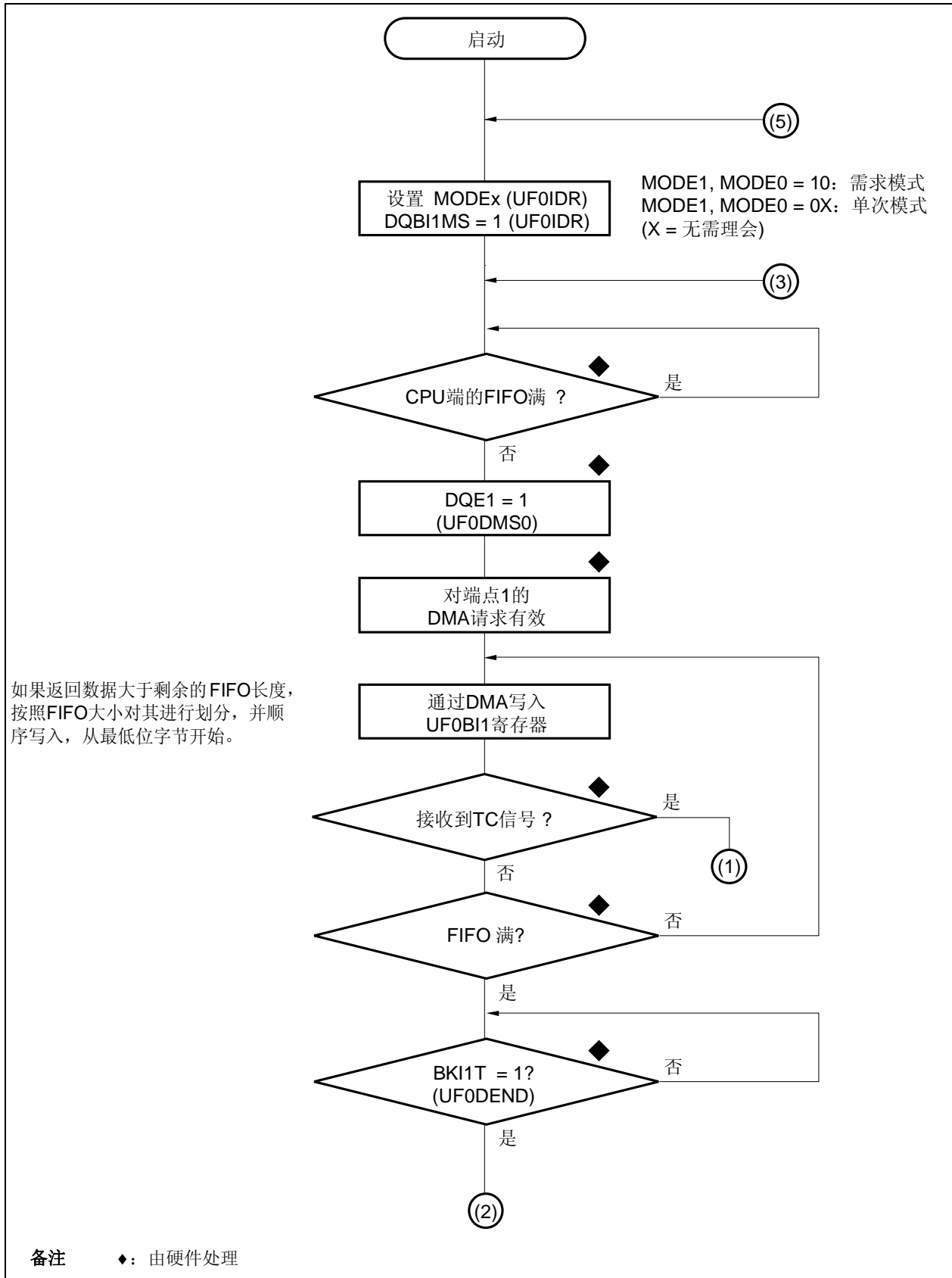


图 20-33. 通过批量传输 (IN) 进行 DMA 处理(2/4)

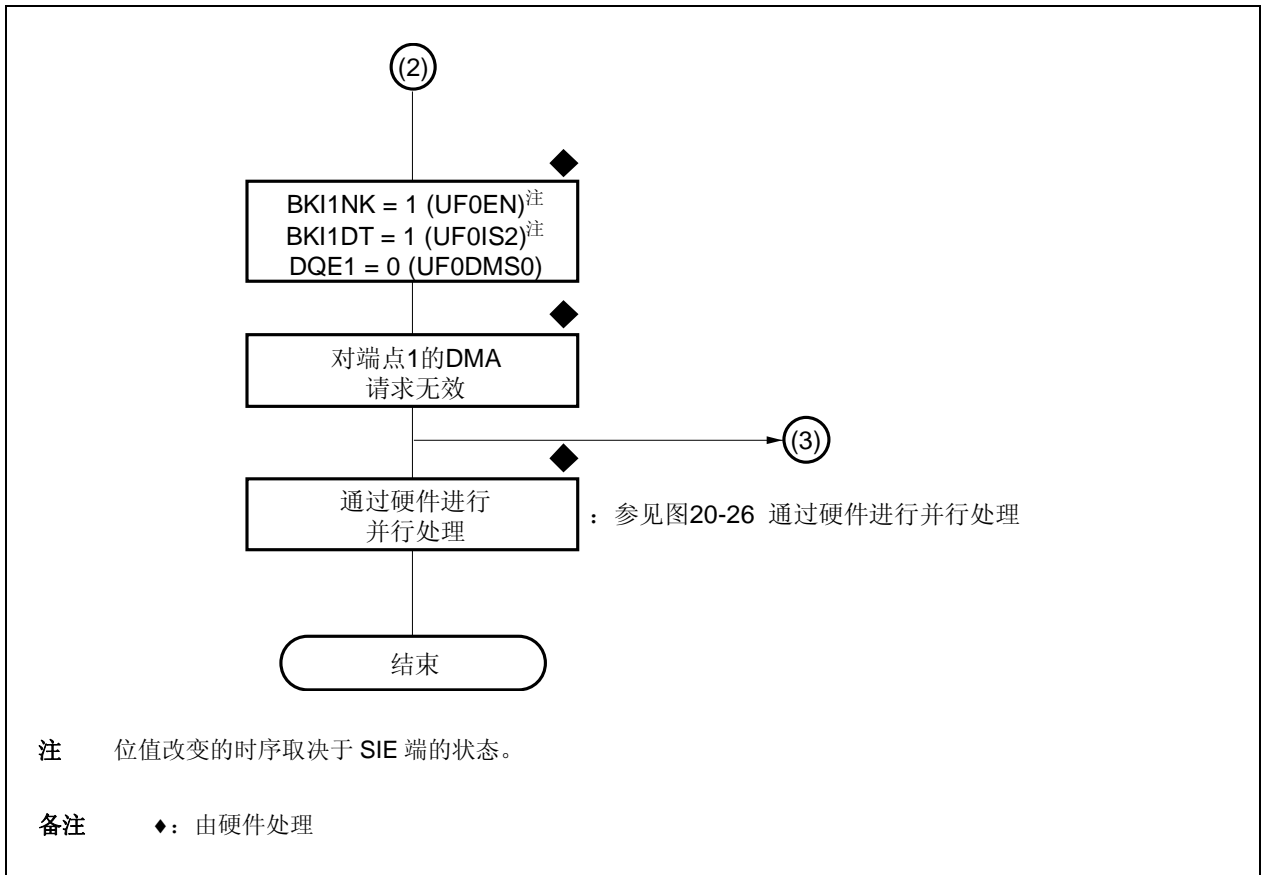


图 20-33. 通过批量传输 (IN) 进行 DMA 处理(3/4)

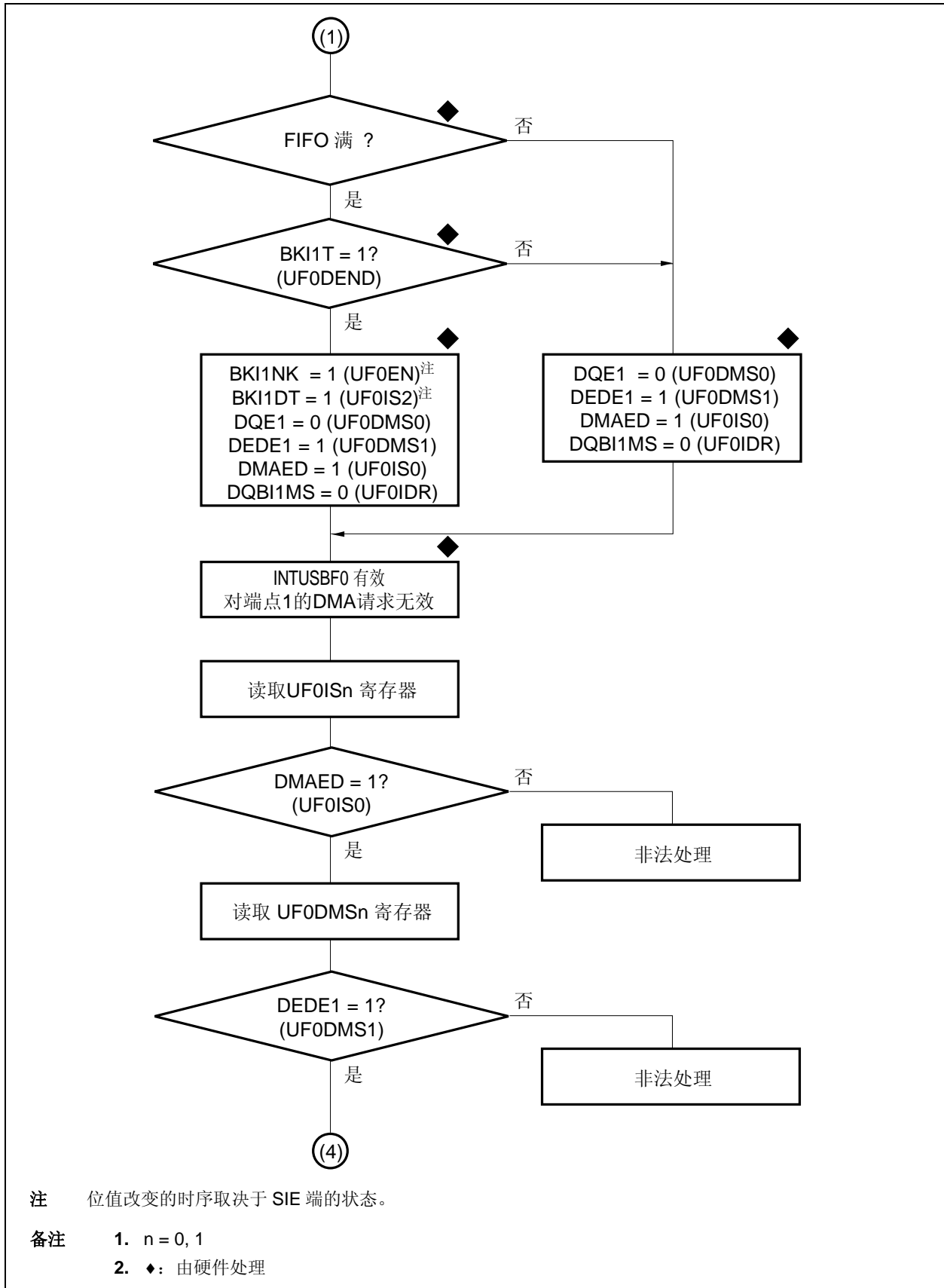
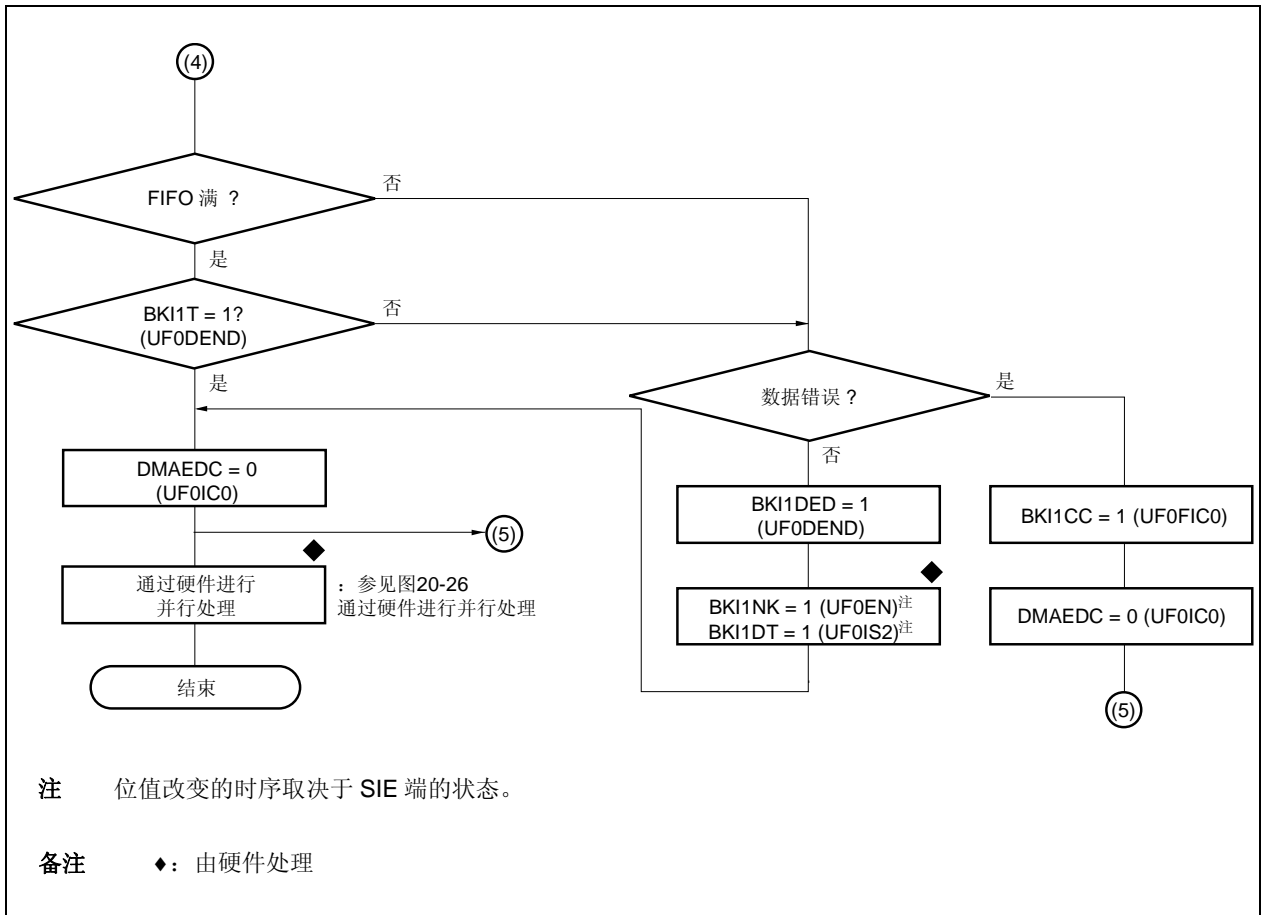


图 20-33. 通过批量传输 (IN) 进行 DMA 处理(4/4)



第二十一章 USB主机控制器 (USBH)

V850ES/JG3-U 和 V850ES/JH3-U 都内置有一个符合通用串行总线规范的 USB 主机控制器 (USBH)，USB 主机控制器和外部设备之间使用基于令牌的协议，通过轮询方法进行数据通讯。

21.1 概述

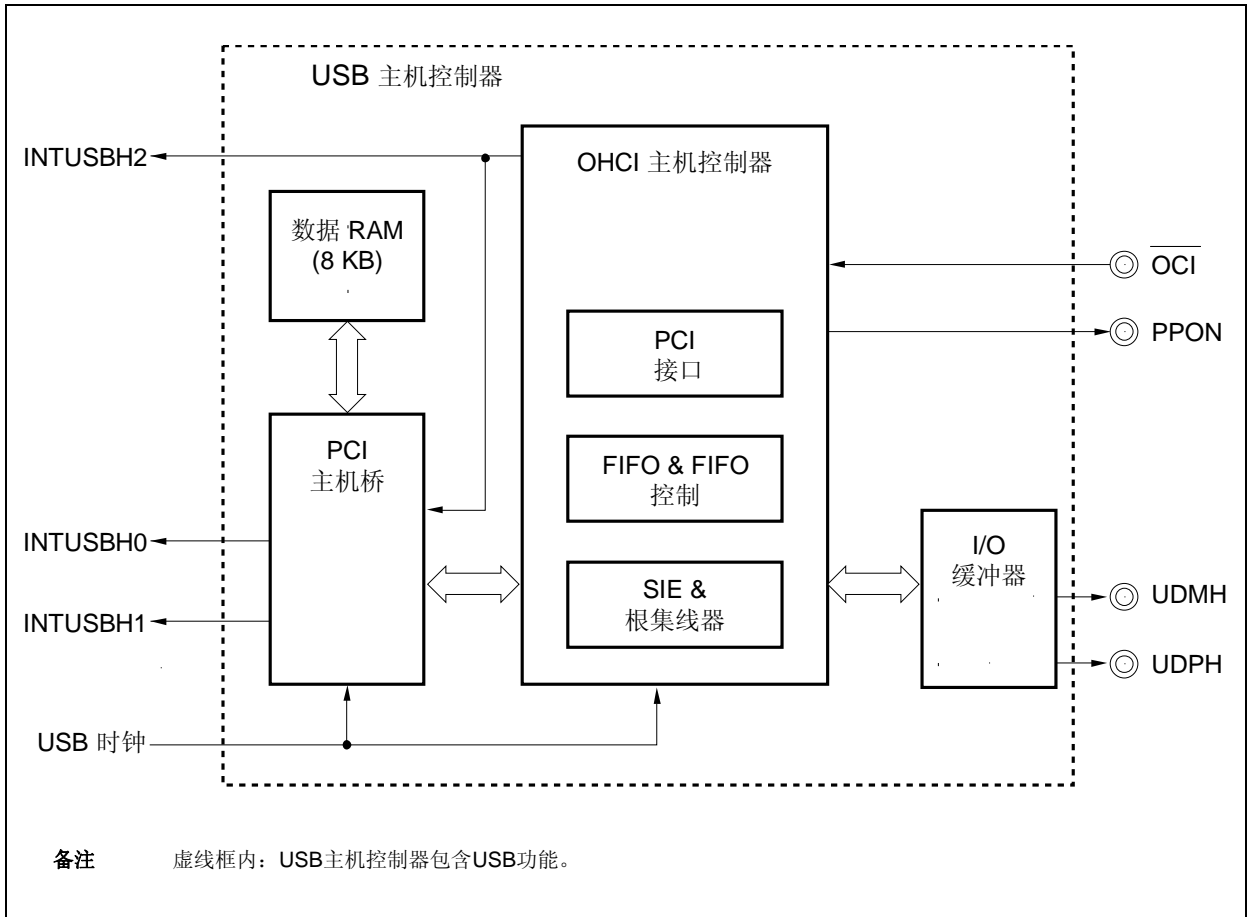
- 符合通用串行总线协议修订版
- 支持 12Mbps (全速) 传输方式
- 支持开放式主机控制器接口 (OHCI) (在 USB 接口状态从禁止到可用时，对控制加以限制)
- 支持控制传输，批量传输，中断传输和同步传输^注
- 内置根集线器 (root hub) 和一个下行端口通道
- 使用 SRAM (8KB) 作为共享存储器
- 时钟：内部时钟 (6 MHz × 8 = 48 MHz) 或者通过 UCLK 引脚输入时钟 (f_{USB} = 48 MHz)

注 同步传输对于系统性能造成很重的负担，因此请确保提前对目标系统的运行状况进行充分评估。

21.2 配置

21.2.1 框图

图 21-1. USB 主机控制器的框图



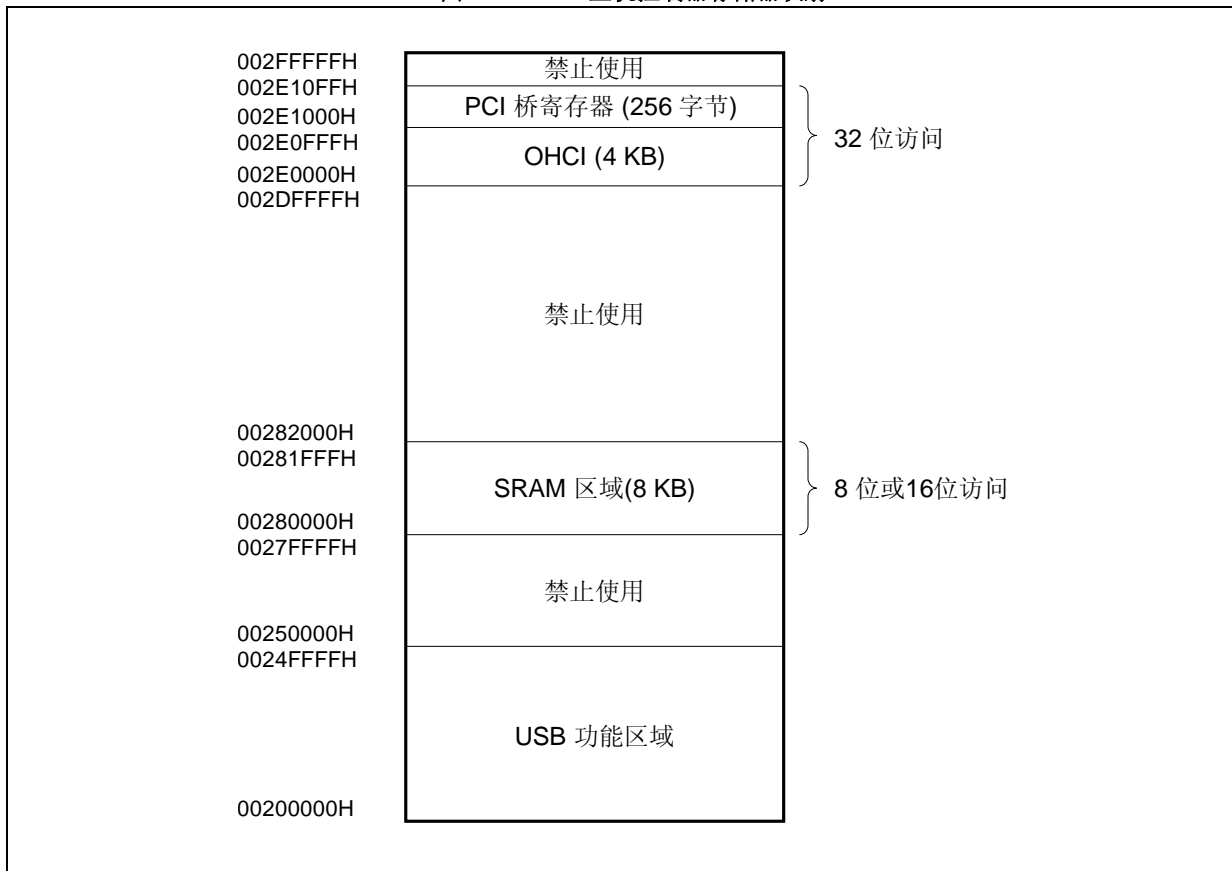
21.2.2 USB 主机控制器存储器映射

USB 主机控制器分配在微处理器内部的 $\overline{CS1}$ 空间 (外部的 $\overline{CS1}$ 引脚不存在)。作为 USB 主机控制器主功能的 OHCI 主机控制器, 可以通过 $\overline{CS1}$ 的 SRAM 接口和内部 PCI 总线桥进行连接。CPU 和内部 PCI 总线桥共享存储器, 且 CPU 的存储空间和 PCI 的存储空间是链接的, 它们之间的关系通过 PCI 主机桥寄存器和可以通过 PCI 主机桥寄存器访问的 OHCI 主机配置寄存器进行定义。

表 21-1. CPU 存储空间分配表

地址	访问宽度	区域空间
00280000H 至 00281FFFH	8 位或16 位	SRAM 区域
002E0000H 至 002E0FFFH	32 位	OHCI 区域
002E1000H 至 002E10FFFH		PCI 总线桥寄存器区域

图 21-2. USB 主机控制器存储器映射



21.2.3 数据访问的注意事项

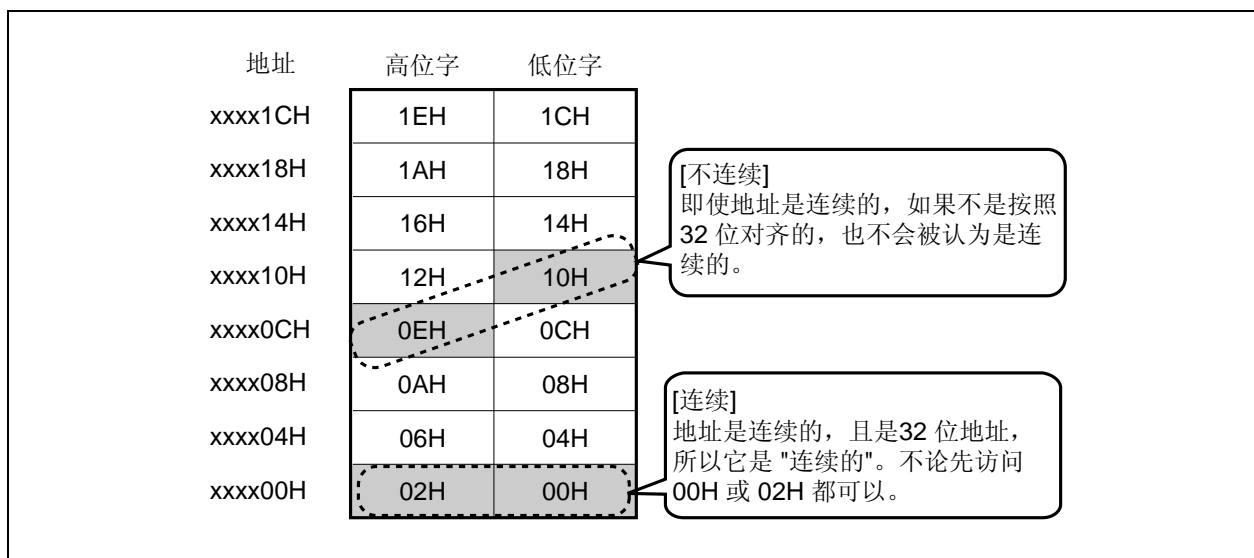
V850ES/JG3-U 和 V850ES/JH3-U 通过 16 位的 PCI 总线桥连接到 USB 主机控制器。OHCI 区域和 PCI 总线桥寄存器按照 32 位宽度进行访问，且 USBH 把从 CPU 收到的 16 位宽度转换为 32 位宽度传给目标设备。

请确保按照表 21-1 的格式访问 32 位的数据单元。数据访问引起的错误会导致 PCI 周期错误 (INTUSBH1)。禁止访问未对齐的地址，因为地址被认为是非连续的。只有访问 32 位对齐的连续地址可以被看做是对连续地址的访问，且正常操作 (参见图 21-3)。

表 21-2. OHCI 和 PCI 总线桥寄存器区域 (32位访问)

	访问	地址		命令	
		第一个周期	第二个周期	第一个周期	第二个周期
<1>	正常	连续		写入	写入
<2>	正常	连续		读取	读取
<3>	错误	不连续		写入	写入
<4>	错误	不连续		读取	读取
<5>	错误	连续或不连续		写入	读取
<6>	错误	连续或不连续		读取	写入
<7>	错误	32位访问	无	写入 (8 位)	
<8>	错误	连续或不连续		写入或读取	写入 (8 位)
<9>	错误	访问禁止区域		-	-

图 21-3. 访问未对齐地址的示例



21.3 外部电路配置

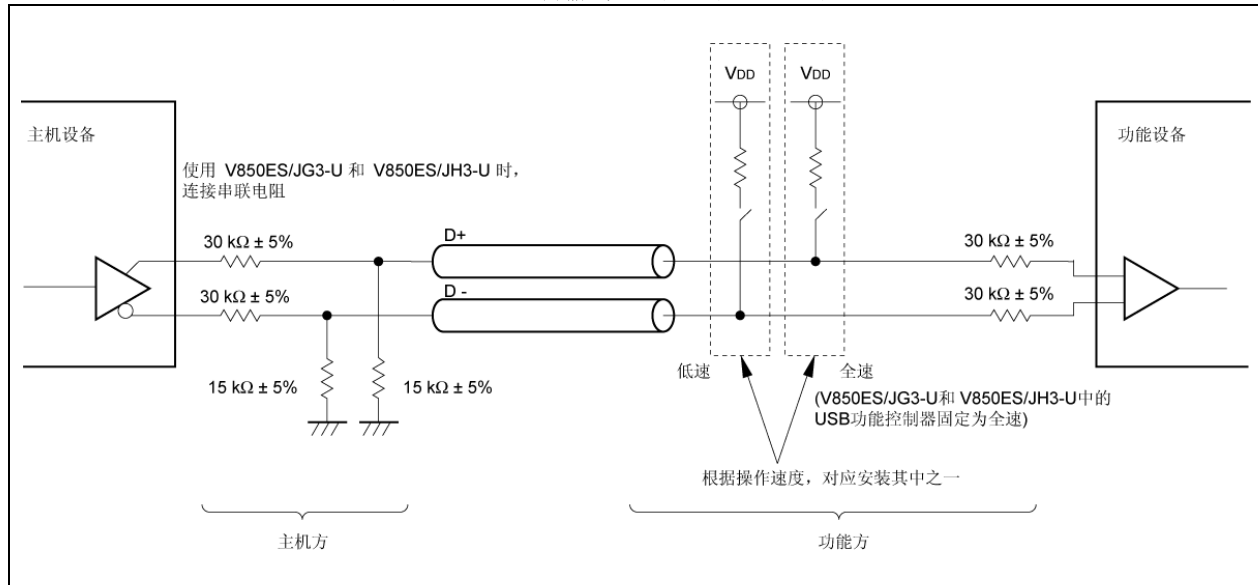
21.3.1 概述

在 USB 传输中,当主机控制器和功能控制器相互进行通讯时,必须对 USB 信号 (D+/D-) 连接上拉/下拉电阻,用来识别通信参与方。另外,在 V850ES/JG3-U 和 V850ES/JH3-U 中,必须连接串联电阻。

因为 V850ES/JG3-U 和 V850ES/JH3-U 内不包含上拉/下拉电阻和串行电阻,请确保在外部进行连接。

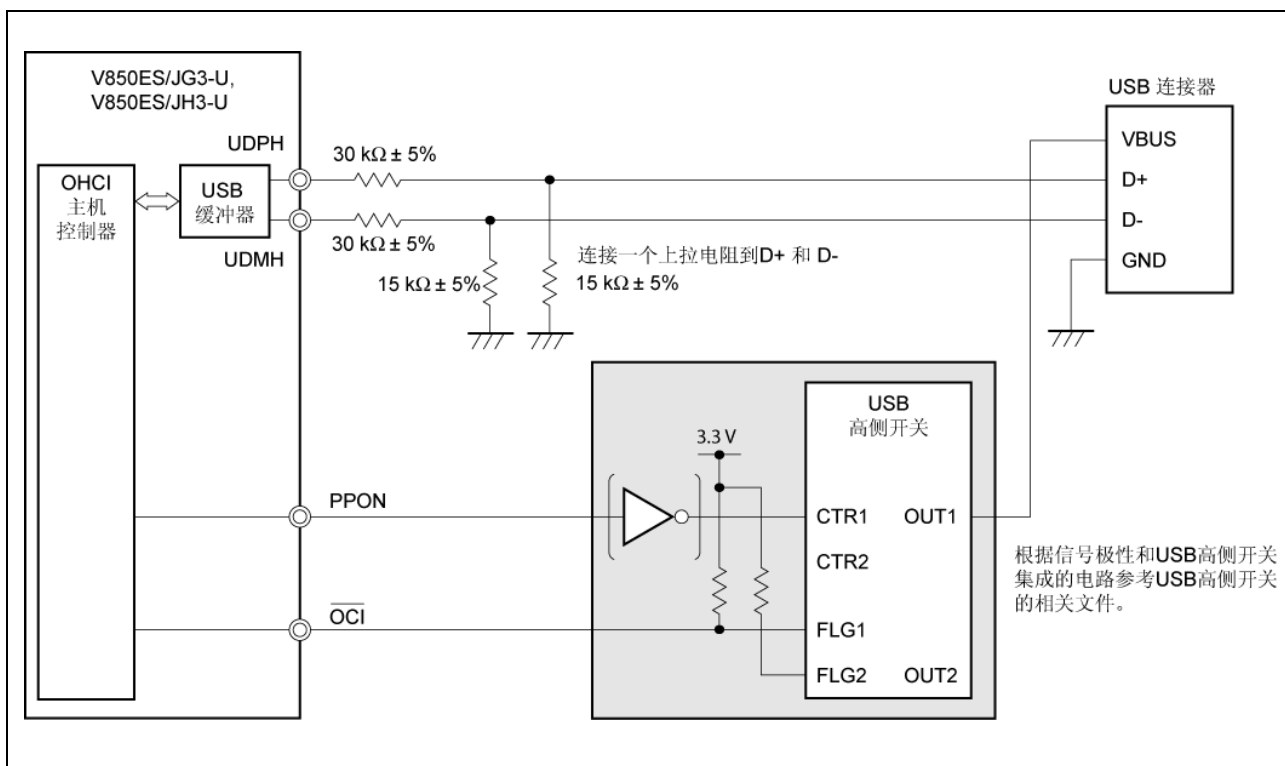
图 21-4 展示了 USB 传输线的电路配置,需要了解详细的外部配置,参见 21.3.2 连接配置。

图 21-4 USB 传输线的上拉/下拉电阻电路配置



21.3.2 连接配置

图 21-5. USB 主机控制器连接示例



(1) 连接到 D+/D- 的串联电阻

在 V850ES/JG3-U 和 V850ES/JH3-U 中，将 $30\Omega \pm 5\%$ 的串联电阻连接到 USB 主机控制器的 D+/D- 引脚（UDPH, UDMH）。如果没有连接电阻，则阻抗率达不到要求，且输出波形会发生混乱。

串联电阻的位置要靠近 V850ES/JG3-U 或 V850ES/JH3-U，串联电阻到 USB 连接器的线缆长度需要相等，这样可以使 D+ 和 D- 的阻抗相同（建议使用 $90\Omega \pm 5\%$ 的电阻）。

(2) D+/D- 的下拉控制

请确保通过 $15\text{ k}\Omega \pm 5\%$ 电阻将 D+/D- 引脚接地，

这种情况下，和功能设备没有连接的配置相同。

21.3.3 USB供电

(1) 过流检测和电源控制

V850ES/JG3-U 和 V850ES/JH3-U 内没有监控过电流的电路，也没有控制向 USB 端口供电的电路。

为了在系统中支持这些功能，需要使用一个类似 USB 高边开关的外部电路来实现这些功能，并且把它连接到 $\overline{\text{OCI}}$ 和 PPON 引脚。

下表展示了用 $\overline{\text{OCI}}$ 和 PPON 引脚来控制外部电路的相关操作，用来实现过流检测或控制向 USB 端口的供电。

表 21-3. $\overline{\text{OCI}}$ 和 PPON 信号

引脚	输入/输出	电平	功能
$\overline{\text{OCI}}$	输入	1	不检测过电流
		0	检测过电流
PPON	输出	1	打开提供给VBUS的电源
		0	关闭提供给VBUS的电源

图 21-5 展示了一个过电流保护和电源 (VBUS) 控制的连接示例。不使用 USB 总线时，停止 VBUS 给端口的供电可以减少电源损耗，虽然他的可用性取决于和 USB 连接器的连接情况。如果已连接的 USB 功能设备是总线供电，建议使用 USB 高边开关，正如连接示例所示。

(2) 根据根集线器寄存器设置条件确定的 VBUS 控制规范

通过 PPS 位进行 VBUS 的控制，取决于 HcRhDescriptorA 和 HcRhDescriptorB (根集线器控制) 寄存器的设置。当控制各个接口的 VBUS 时，HcRhDescriptorA.NPS 和 HcRhDescriptorA.PSM 位必须置为“1”。

21.4 注意事项

(1) 时钟准确度

要操作 USB 主机控制器，必须使用内部时钟（6 MHz 外部时钟 \times 8 内部时钟 = 48 MHz 内部时钟）或者外部时钟（输入到 UCLK 引脚的外部时钟， $f_{\text{USB}} = 48 \text{ MHz}$ ）作为 USB 时钟。当 USB 时钟使用内部时钟时，使用准确度为 $6 \text{ MHz} \pm 500 \text{ ppm}$ 的时钟发生器（最大值）。当 USB 时钟使用外部时钟时，向 UCLK 引脚提供准确度为 $48 \text{ MHz} \pm 500 \text{ ppm}$ （最大值）的时钟。如果 USB 时钟准确度下降，数据传输将无法达到 USB 的传输速率。

(2) 集线器 (HUB) 连接

集线器连接推荐使用外部时钟（晶体振荡器）。使用内部时钟时，可能达不到 USB 传输速率，因为微控制器内部时钟中的时钟抖动。使用内部时钟时，建议和主设备采用一对一的连接方式。

21.5 控制寄存器

21.5.1 USB控制寄存器

(1) USB 时钟选择寄存器 (UCKSEL)

UCKSEL 寄存器用于选择 USB 控制器的操作时钟。

USB 功能控制器也可以使用 UCKSEL 寄存器。详情参见 20.6.1 (1) USB 时钟选择寄存器 (UCKSEL)。

(2) USB 功能选择寄存器 (UHCKMSK)

UHCKMSK 寄存器用于选择 USB 控制器的功能，

USB 功能控制器也可以使用 UHCKMSK 寄存器。详情参见 20.6.1 (2) USB 功能选择寄存器 (UHCKMSK)。

21.6 PCI主机桥

21.6.1 PCI主机桥

PCI 主机桥是一个桥电路，用于连接 CPU 系统和 OHCI 主机控制器的桥电路，具有下列功能。

- **PCI 主周期控制**
对应 CPU 提交的总线访问请求，发布下列 PCI 周期
 - PCI 配置寄存器读/写单周期
 - PCI 存储器读/写单周期

- **PCI 从周期控制**
对应从 PCI 总线到 SDRAM 或者 SRAM 空间访问，响应 PCI 存储器读/写周期（最多突发传输的 8 个双字）

- **PCI 错误处理**
在主设备异常中止、目标异常中止、 $\overline{\text{PERR}}$ 接收和 $\overline{\text{SERR}}$ 接收时，产生错误中断 INTUSBH0（保持发生错误前的瞬时地址）。

- **PCI 地址转换控制**
PCI 窗口基地址寄存器，用来对 CPU 传送来的物理地址进行转换，然后输出到 PCI 总线。

- **控制 V850ES/JG3-U 和 V850ES/JH3-U 内的存储器控制器总线**
通过存储器控制总线从 CPU 发起访问的硬件等待 (WAIT) 来控制总线周期。

- **SRAM 控制**
内置的 8KB SRAM 作为共享存储器，主要用来分配描述符。对由 CPU 和 PCI 总线到 SRAM 的访问进行控制和仲裁。

21.6.2 PCI 主机桥控制器

PCI 桥有以下 PCI 主机桥寄存器。

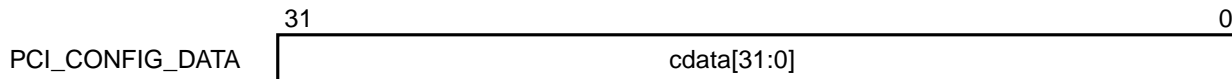
这些寄存器只可以按照 32 位宽度进行访问。

表 21-4. PCI 主机桥寄存器

地址	寄存器名称	标识符	读/写	可操作位宽度				初始值
				1	8	16	32	
002E1000H	PCI 配置数据寄存器	PCI_CONFIG_DATA	R/W				√	不确定
002E1004H	PCI 配置地址寄存器	PCI_CONFIG_ADD	R/W				√	00000000H
002E1008H	PCI 控制寄存器 1	PCI_CONTROL1	R/W				√	07000300H
002E100CH	PCI 控制寄存器 2	PCI_CONTROL2	R/W				√	00000077H
002E1014H	PCI 窗口基地址寄存器	PCI_WIN_BASE	R/W				√	002E0000H
002E1018H	PCI 中断状态寄存器	PCI_INT_STATUS	R/W				√	00000000H
002E101CH	PCI 中断控制寄存器	PCI_INT_CTL	R/W				√	00000000H
002E1020H	PCI 总线错误地址寄存器	PCI_ERR_ADD	R/W				√	00000000H
002E1024H	总线桥控制寄存器	BGR_CTL	R/W				√	00000016H
002E1050H	SRAM_BASE 寄存器	SRAM_BASE	R/W				√	00280000H

(1) PCI 配置数据寄存器 (PCI_CONFIG_DATA)

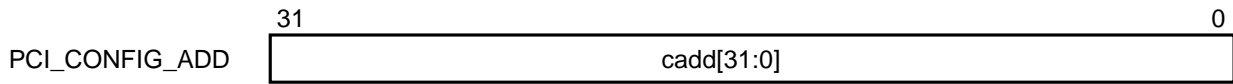
复位后：不确定 R/W 地址：002E1000H



位的位置	位名称	功能
31-0	cdata[31:0]	用来访问 OHCI 主机配置寄存器。 PCI_CONFIG_ADD 寄存器必须提前设置。

(2) PCI 配置地址寄存器 (PCI_CONFIG_ADD)

复位后: 0000 0000H R/W 地址: 002E1004H



位的位置	位名称	功能
31-0	cadd[31:0]	<p>设置将要被访问的OHCI主机配置寄存器的地址 [如何设置 OHCI 配置地址寄存器]</p> <div style="text-align: center;"> </div> <ul style="list-style-type: none"> • 寄存器编号: 指定OHCI主机配置寄存器的数量。 • 功能编号: 指定PCI多功能设备的功能数量。 • IDSEL 规范: 选择将要访问的PCI设备的IDSEL信号。 <p>由于 PCI 主机桥用 AD[31:11] 替代 IDSEL 信号, 设置其中任何一位相当于把 AD 信号连接到 IDSEL 引脚上, 对连接到 PCI 总线上的 PCI 设备执行配置访问。 在 V850ES/JG3-U 和 V850ES/JH3-U 中, AD31 连接到 IDSEL 引脚上。因此, 将 cadd31 位设置为 1, 可以访问 PCI 设备 (OHCI 主机控制器)。</p> <p>[访问 OHCI 主机配置寄存器的方法] 通过以下步骤可以访问 PCI 总线上的配置寄存器</p> <ul style="list-style-type: none"> • PCI_CONFIG_ADD 寄存器设置 在 PCI 配置周期中, 设置将要访问的地址。 • PCI_CONFIG_DATA 寄存器设置 读取 PCI_CONFIG_DATA 寄存器, 可以获得 PCI_CONFIG_ADD 寄存器的设置值。 当数值被写入 PCI_CONFIG_DATA 寄存器时, 可以对相应的 OHCI 主机配置寄存器执行写入访问。

(3) PCI 控制寄存器 1 (PCI_CONTROL1)

复位后: 0700 0300H R/W 地址: 002E1008H

PCI_CONTROL1	31	30	29	28	27	26	25	24
	pci_parkcnt7	pci_parkcnt6	pci_parkcnt5	pci_parkcnt4	pci_parkcnt3	pci_parkcnt2	pci_parkcnt1	pci_parkcnt0
	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	pci_bpmode
	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	pci_req_en1	pci_req_en0
	7	6	5	4	3	2	1	0
	0	0	pci_pchken	pci_reset	sram_en	sdram_en	mem_en	0

位的位置	位名称	功能
31-24	pci_parkcnt [7:0]	PCI 总线驻留定时器 设置切换到总线驻留模式的所需的时间。设置 $\overline{\text{FRAME}}$ 和 $\overline{\text{IRDY}}$ 位为 1, 可以启动计时。 使用该字段的初始设置即可。
16	pci_bpmode	PCI 总线驻留模式。 0: 只有 V850ES/JG3-U 和 V850ES/JH3-U 可以作为总线驻留主方 (初始值)。 1: 总线驻留主方是最后访问 PCI 总线的主方。 使用该字段的初始设置即可。
9, 8	pci_req_en [1:0]	PCI 请求使能。 pci_req_en0 位固定设置为 “1”。 0: 禁止 PCI 请求 (初始值)。 1: 使能 PCI 请求。 当发布 PCI 需求时, 该位置位 (1)。
5	pci_pchken	PCI 奇偶校验使能 0: 禁止奇偶校验 (初始值) 1: 使能既有校验。 使用该位的初始设置即可。
4	pci_reset	PCI 复位 0: PCI 总线为复位状态 (初始值)。 1: PCI 总线复位释放状态。 访问 OHCI 控制器时, 该位置位 (1)。
3	sram_en	SRAM 区域使能 0: 不响应从 PCI 总线到 SRAM 的访问 (初始值)。 1: 响应从 PCI 总线到 SRAM 的访问。 OHCI 控制器访问 SRAM 时, 该位置位 (1)。
2	sdram_en	SDRAM 区域使能 0: 不响应从 PCI 总线到 SRAM 的访问 (初始值)。 1: 响应从 PCI 总线到 SRAM 的访问。 OHCI 控制器访问 SDRAM 时, 该位置位 (1)。
1	mem_en	PCI 存储器区域使能 0: 禁止从 CPU 到 PCI 存储器区域的访问 (初始值)。 1: 使能从 CPU 到 PCI 存储器区域的访问。 访问 OHCI 主机控制器的 OHCI 主机配置寄存器时, 该位置位 (1)。

(4) PCI 控制寄存器 2 (PCI_CONTROL2)

复位后: 0000 0077H R/W 地址: 002E100CH

PCI_CONTROL2	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	wbuf_busy
	7	6	5	4	3	2	1	0
	0	wburst_size2	wburst_size1	wburst_size0	0	rburst_size2	rburst_size1	rburst_size0

位的位置	位名称	功能																																							
8	wbuf_busy	<p>PCI 写入缓冲器忙</p> <p>当写入 PCI 目标 (SDRAM) 时, 表示写入缓冲器中的数据状态。</p> <p>0: 要写入 SDRAM/SRAM 的数据不保持在写缓冲器中。</p> <p>1: 要写入 SDRAM/SRAM 的数据保持在写缓冲器中。</p> <p>该位只读。</p>																																							
6-4	wburst_size [2:0]	<p>PCI 写入突发最大值</p> <p>设置写入 PCI 目标 (SDRAM) 的最大突发长度。</p> <table border="1"> <thead> <tr> <th colspan="3">wburst_size</th> <th rowspan="2">写入PCI目标的最大突发长度。</th> </tr> <tr> <th>2</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>仅支持单次传输</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>2 突发</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>3 突发</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>4 突发</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>5 突发</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>6 突发</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>7 突发</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>8 突发 (初始值)</td> </tr> </tbody> </table> <p>使用该位的初始设置即可。</p>	wburst_size			写入PCI目标的最大突发长度。	2	1	0	0	0	0	仅支持单次传输	0	0	1	2 突发	0	1	0	3 突发	0	1	1	4 突发	1	0	0	5 突发	1	0	1	6 突发	1	1	0	7 突发	1	1	1	8 突发 (初始值)
wburst_size			写入PCI目标的最大突发长度。																																						
2	1	0																																							
0	0	0	仅支持单次传输																																						
0	0	1	2 突发																																						
0	1	0	3 突发																																						
0	1	1	4 突发																																						
1	0	0	5 突发																																						
1	0	1	6 突发																																						
1	1	0	7 突发																																						
1	1	1	8 突发 (初始值)																																						
2-0	rburst_size [2:0]	<p>PCI 读取突发最大值</p> <p>设置从 PCI 目标 (SDRAM) 中读取的最大突发长度。</p> <table border="1"> <thead> <tr> <th colspan="3">rburst_size</th> <th rowspan="2">从PCI目标中读出的最大突发长度。</th> </tr> <tr> <th>2</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>仅支持单次传输</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>2 突发</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>3 突发</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>4 突发</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>5 突发</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>6 突发</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>7 突发</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>8 突发 (初始值)</td> </tr> </tbody> </table> <p>使用该位的初始设置即可。</p>	rburst_size			从PCI目标中读出的最大突发长度。	2	1	0	0	0	0	仅支持单次传输	0	0	1	2 突发	0	1	0	3 突发	0	1	1	4 突发	1	0	0	5 突发	1	0	1	6 突发	1	1	0	7 突发	1	1	1	8 突发 (初始值)
rburst_size			从PCI目标中读出的最大突发长度。																																						
2	1	0																																							
0	0	0	仅支持单次传输																																						
0	0	1	2 突发																																						
0	1	0	3 突发																																						
0	1	1	4 突发																																						
1	0	0	5 突发																																						
1	0	1	6 突发																																						
1	1	0	7 突发																																						
1	1	1	8 突发 (初始值)																																						

(5) PCI 窗口基础地址寄存器 (PCI_WIN_BASE)

复位后: 002E 0000H R/W 地址: 002E1014H

	31	30	29	28	27	26	25	24
PCI_WIN_BASE	p_base23	p_base22	p_base21	p_base20	p_base19	p_base18	p_base17	p_base16
	23	22	21	20	19	18	17	16
	p_base23	p_base22	p_base21	p_base20	p_base19	p_base18	p_base17	p_base16
	15	14	13	12	11	10	9	8
	p_base15	p_base14	p_base13	p_base12	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

位的位置	位名称	功能
31-12	p_base [31:12]	PCI 窗口基础地址寄存器 设置通过 PCI 窗口将要访问的 PCI 空间的最高地址 (PCI 窗口基础地址)。 通过 PCI 窗口访问 PCI 总线存储器空间时, 根据该寄存器的设置, 可以使能对 PCI 总线存储空间任意 4KB 的区域进行访问。

(6) PCI 中断状态寄存器 (PCI_INT_STATUS)

中断状态寄存器用来确认由 PCI 主机桥输出的 INTUSBH0 和 INTUSBH1 中断信号源。

中断源 inta, int_smmi 和 int_pme 的源会被发布中断的单元清除。

中断源 serr, perr, mabort 和 tabort 的源仅在调试期间使用, 在其它操作中无法使用。

如何使能中断源, 参见 6.7.5 PCI_INT_CTL 寄存器。

(1/2)

复位后: 0000 0000H R/W 地址: 002E1018H

PCI_INT_STATUS	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
0	0	0	0	Int_pme	Int_smmi	0	Inta	
7	6	5	4	3	2	1	0	
0	0	0	0	serr	perr	mabort	tabort	

位的位置	位名称	功能
11	int_pme	PCI 中断 PME 状态 0: 没有中断源。 1: 因 PME 产生一个中断。 该中断源被产生 PME 的源清除 (0)。 该位只读。
10	Int_smmi	PCI 中断 SMMI 状态 0: 没有中断源。 1: 因 SMMI 产生一个中断。 该中断源被产生 SMMI 的源清除 (0)。 该位只读。
8	inta	PCI 中断 INTA 状态 0: 没有中断源。 1: 因 INTA 产生一个中断。 该中断源被产生 INTA 的源清除 (0)。 该位只读。
3	serr	PCI 主机桥系统错误中断状态 0: 无中断源 1: 检测到一个系统错误 将“1”写入该位, 即可清除该中断源 (0)。 系统错误只在调试时产生, 在正常的操作时不使用。

位的位置	位名称	功能
2	perr	PCI 主机桥奇偶检验错误中断状态 0: 无中断源 1: 检测到一个奇偶检验错误。 将“1”写入该位, 即可清除该中断源 (0)。 奇偶检验错误只在调试时产生, 在正常的操作时不使用。
1	mabort	PCI 主机桥主异常中断状态 0: 无中断源 1: 接收到主异常中断。 将“1”写入该位, 即可清除该中断源 (0)。 主异常中止只在调试时产生, 在正常的操作时不使用。
0	tabort	PCI 主机桥目标异常中断状态 0: 无中断源 1: 检测到目标异常错误。 将“1”写入该位, 即可清除该中断源 (0)。 目标异常中止只在调试时产生, 在正常的操作时不使用。

(7) PCI 中断控制寄存器 (PCI_INT_CTL)

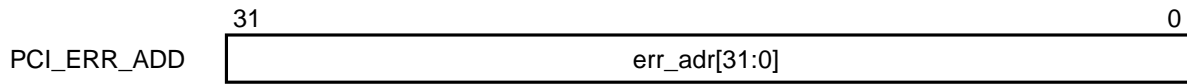
复位后: 0000 0000H R/W 地址: 002E101CH

	31	30	29	28	27	26	25	24
PCI_INT_CTL	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	0	0	0	0	int_pme_en	int_smmi_en	0	inta_en
	7	6	5	4	3	2	1	0
	0	0	0	0	serrint_en	perrint_en	mabortint_en	tabortint_en

位的位置	位名称	功能
11	int_pme_en	PCI 中断 PME 使能 0: 禁止作为中断源 (初始值)。 1: 使能作为中断源。
10	int_smmi_en	PCI 中断 SMMI 使能 0: 禁止作为中断源 (初始值)。 1: 使能作为中断源。
8	inta_en	PCI 中断 INTA 使能 0: 禁止作为中断源 (初始值)。 1: 使能作为中断源。
3	serrint_en	PCI 主机桥系统错误中断使能 0: 禁止作为中断源 (初始值) 1: 使能作为中断源 该中断只在调试时产生, 在正常的操作时不使用。
2	perrint_en	PCI 主机桥奇偶错误中断使能 0: 禁止作为中断源 (初始值) 1: 使能作为中断源 该中断只在调试时产生, 在正常的操作时不使用。
1	mabortint_en	PCI 主机桥异常中断使能 0: 禁止作为中断源 (初始值) 1: 使能作为中断源 该中断只在调试时产生, 在正常的操作时不使用。
0	tabortint_en	PCI 主机桥目标异常中断使能 0: 禁止作为中断源 (初始值) 1: 使能作为中断源 该中断只在调试时产生, 在正常的操作时不使用。

(8) PCI 总线错误地址寄存器 (PCI_ERR_ADD)

复位后: 0000 0000H R/W 地址: 002E1020H



位的位置	位名称	功能
31-0	err_adr [31:0]	<p>PCI 总线错误地址</p> <p>该字段保留发生以下错误源的 PCI 地址。</p> <ul style="list-style-type: none"> • $\overline{\text{SERR}}$ 输入 • 奇偶校验错误 • PCI 总线主异常 • PCI 总线目标异常 <p>被读取时该字段的所有位被清除 (0)。一旦产生总线错误, 有值被写入该寄存器, 此数值将被持续保留, 直到数值被读取或者因为新的总线错误而被更新。该寄存器只在调试时产生, 在正常的操作时不使用。</p>

(9) 总线桥控制寄存器 (BRG_CTL)

复位后: 0000 0000H R/W 地址: 002E1024H

	31	30	29	28	27	26	25	24
BRG_CTL	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	cnv_err
	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	data_wait_en	no_wait_en
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	cnv_err_en

位的位置	位名称	功能
16	cnv_err	<p>16-32 位周期转换错误</p> <p>在从 16 位转换到 32 位期间, 因为接收到异常周期, 产生一个错误, 则该位被置位。该位将报告 INTUSBH1 中断的产生, 该位永远不会被置位, 除非 cnv_err_en 位被置为“1”。</p> <p>0: 没有中断源产生</p> <p>1: 检测到周期转换错误</p> <p>INTUSBH1 (PCI 周期错误) 中断只在调试时产生, 在正常的操作时不使用。</p>
9	data_wait_en	<p>数据等待使能</p> <p>在 CPU 总线周期结尾处强制插入 1 等待 (数据等待)。</p> <p>0: 不强制插入数据等待 (初始值)</p> <p>1: 强制插入数据等待。</p>
8	no_wait_en	<p>无等待使能</p> <p>设置是否使能 CPU 总线周期的无等待操作。</p> <p>0: 禁止无等待操作 (每次至少要插入一个等待) (初始值)</p> <p>1: 使能等待操作 (推荐)</p> <p>该位数值由“1”改变为“0”之前, 推荐对该位读取一次, 然后再写入数据, 这样可以弥补反映改变的时间延迟。此时读取到的值不一定正确, 因为没有正确插入等待, 所以丢弃读到的数据。</p>
0	cnv_err_en	<p>16-32 位周期转换错误使能</p> <p>该位用来设置和清除中断源。当该位被置为“0”时, cnv_err 位被清除, 同时屏蔽中断源。</p> <p>0: 清除中断源 (初始值)</p> <p>1: 使能中断源。</p> <p>INTUSBH1 (PCI 周期错误) 中断只在调试时产生, 在正常的操作时不使用。</p>

(10) SRAM_BASE 寄存器 (SRAM_BASE)

复位后: 0028 0000H R/W 地址: 002E1050H

SRAM_BASE	31	30	29	28	27	26	25	24
	sram_base31	sram_base30	sram_base29	sram_base28	sram_base27	sram_base26	sram_base25	sram_base24
	23	22	21	20	19	18	17	16
	sram_base23	sram_base22	sram_base21	sram_base20	sram_base19	sram_base18	sram_base17	sram_base16
	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

位的位置	位名称	功能
31-16	sram_base [31:16]	SRAM 区域基地址 为访问 SRAM 区域设置基地址。 在 PCI 空间中设置 SRAM 区域的基地址。 当 PCI 目标设备试图访问存储器空间时，地址将会和该寄存器指定的地址进行比较。如果地址匹配，返回相关的响应。

21.7 OHCI主机控制器

21.7.1 OHCI主机控制器功能

OHCI 主机控制器有下列功能。

- 符合 OpenHCI 规范协议 1.0a 版本
- 符合通用串行总线规范修订版 1.1
支持全速 (12Mbps) 传输
- 内置 1 个通道的根集线器, 支持一路下行端口
- USB 时钟: 38MHZ, PCI 时钟: 25 至 33MHZ
- 存储器空间
 - 4KB PCI 存储器区域 (OHCI 操作寄存器) 分配
 - 256 字节主机控制器通信区域 (HCCA) 分配
- 和 CPU 通讯
 - 通过 OHCI 主机控制器的操作寄存器和主机控制器通信区域, 执行通讯操作。

在 CPU 和 OHCI 主机控制器之间, 连接有两条通讯通道。通讯通道包括 OHCI 操作寄存器和作为本次通讯目标 (从) 设备的 OHCI 主机控制器。在 OHCI 主机配置寄存器中的基地址寄存器 (10H) 作为一个指向 OHCI 操作寄存器的指针。

第二条通讯通道是 OHCI 操作寄存器中的主机控制器通信区域 (HCCA), 它作为一个指向共享存储器的指针。OHCI 主机控制器作为本次通讯的主方。

通讯使用的描述符信息由 OHCI 操作寄存器和 HCCA 空间来管理。

21.7.2 OHCI主机配置寄存器

OHCI主机配置寄存器配置256字节的寄存器空间，且包括在OHCI主机控制器内。通过PCI主机桥寄存器（PCI_CONFIG_DATA, PCI_CONFIG_ADD）从CPU系统可以访问这些寄存器。具体地址参见 21.2.2 USB 主机控制器存储器映射。

表22-4. PCI 配置寄存器

地址	位 31									24	23									16	15										8	7							0
00H	设备 ID																厂商 ID																						
04H	状态																命令																						
08H	分类符号																								版本 ID														
0CH	BIST										标题类型										BIST										标题类型								
10H	基地址																																						
14H	保留																																						
18H																																							
1CH																																							
20H																																							
24H																																							
28H	保留																																						
2CH	子系统 ID																子系统厂商 ID																						
30H	保留																																						
34H	保留																								Cap_ptr														
38H	保留																																						
3CH	Max_Latency										Min_Gnt										Max_Latency										Min_Gnt								
40H	PMC																Next_Item_Ptr																PMC						
44H	数据										DMCSR_BSE										数据																		
E0H	保留																																						

(1) 厂商 ID 和设备 ID (偏移地址 00H)



(2) 命令和状态 (偏移地址 04H)

(1/2)

复位后: 0210 0000H R/W

31	30	29	28	27	26	25	24
Detected Parity Error	Signaled System Error	Received Master Abort	Received Target Abort	Signaled Target Abort	Devsel Timing1	Devsel Timing0	Data Parity Detected
23	22	21	20	19	18	17	16
Fast Back to Back Capable	0	0	Capabilities	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	0	0	Fast Back to Back Enable	SERR Enable
7	6	5	4	3	2	1	0
Wait Cycle Control	Parity Error Response	VGA Pallet Snoop	Memory Write and Invalidate	Special Cycle	Bus Master	Memory Space	I/O Space

位的位置	位名称	功能
31	Detected Parity Error	表示奇偶错误状态。当检测到地址或数据奇偶错误时, 该位被置位。通过 PCI 总线写入“1”时, 该位被清除 (0)。
30	Signaled System Error	表示 SERR 状态。在发生系统错误时, 该位被置位。通过 PCI 总线写入“1”时, 该位被清除 (0)。
29	Received Master Abort	表示主设备的主异常中止状态。在主设备操作因为主异常中止被终结时, 该位被置位。通过 PCI 总线写入“1”时, 该位被清除 (0)。
28	Received Target Abort	表示主设备的目标异常中止状态。在主设备操作因为目标异常中止被终结时, 该位被置位。通过 PCI 总线写入“1”时, 该位被清除 (0)。
27	Signaled Target Abort	表示从设备的目标异常中止状态。在从设备操作因为目标异常中止被终结时, 该位被置位。通过 PCI 总线写入“1”时, 该位被清除 (0)。
26-25	Devsel Timing[1:0]	表示 DEVSEL 响应速度。 由于仅支持中等 (Medium) 模式, 该字段被固定为“01”。 这些位为只读属性。
24	Data Parity Detected	主操作期间, 检测到奇偶校验错误时, 该位被置位。通过 PCI 总线写入“1”时, 该位被清除 (0)。当通过奇偶校验错误响应位 (命令寄存器) 禁止奇偶错误响应时, 该位固定为“1”。
23	Fast Back to Back Capable	表示快速背靠背的支持状态。 因为不支持快速背靠背, 该位固定为“1”。 该位只读。
22-21	-	保留 (请确保向这些位写入 “0”)。
20	Capabilities	表示支持电源管理模式。该位固定为“1”。 该位只读。
19-10	-	保留 (请确保向这些位写入 “0”)。
9	Fast Back to Back Enable	快速背靠背使能位。 因为 USB 主机控制器不支持快速背靠背, 该位固定为“0”。 该位只读。

位的位置	位名称	功能
8	SERR Enable	SERR 使能位。 通过 SERR 信号发送系统错误时，该位置为“1”。
7	Wait Cycle Control	等待周期控制使能位。 因为 USB 主机控制器不支持地址/数据的步进，该位固定为“0”。 该位只读。
6	Parity Error Response	奇偶错误响应使能位。 当检查奇偶错误时，该位置为“1”。
5	VGA Pallet Snoop	VGA 调色板窥探使能位 因为 USB 主机控制器不支持 VGA 调色板窥探，该位固定为“0”。 该位只读。
4	Memory write and Invalidate	存储器写和无效使能位 因为 USB 主机控制器不支持存储器写和无效，该位固定为“0”。 该位只读。
3	Special Cycle	特定周期使能位 因为 USB 主机控制器不支持特定周期，该位固定为“0”。 该位只读。
2	Bus Master	总线主设备使能位 使能 PCI 总线的总线主设备访问，在通过系统总线访问 SRAM 前，必须设置为“1”。在主机控制器初始化时，设置该位为“1”。
1	Memory Space	存储器空间访问使能位 根据 PCI 总线协议使能存储器访问，在访问寄存之前必须将该位设置为“1”。在主机控制器初始化时，设置该位为“1”。
0	I/O Space	I/O 空间访问使能位 允许对 PCI 总线协议中定义的 I/O 空间进行访问，因为 USB 主机控制器不使用 I/O 访问，该位固定为“0”。 该位只读。

(3) 版本 ID 和分类符号 (偏移地址 08H)

复位后: 0C03 1042H R

31	30	29	28	27	26	25	24
Base Class7	Base Class6	Base Class5	Base Class4	Base Class3	Base Class2	Base Class1	Base Class0
23	22	21	20	19	18	17	16
Sub Class7	Sub Class6	Sub Class5	Sub Class4	Sub Class3	Sub Class2	Sub Class1	Sub Class0
15	14	13	12	11	10	9	8
Programming I/F7	Programming I/F6	Programming I/F5	Programming I/F4	Programming I/F3	Programming I/F2	Programming I/F1	Programming I/F0
7	6	5	4	3	2	1	0
Revision ID7	Revision ID6	Revision ID5	Revision ID4	Revision ID3	Revision ID2	Revision ID1	Revision ID0

位的位置	位名称	功能
31-24	Base Class [7:0]	表示 PCI 规范中规定的基本类 (分类符号)。 该字段固定为 0CH, 表示串行外设总线控制器。
23-16	Sub Class [7:0]	表示 PCI 规范中规定的子类 (分类符号)。 该字段固定为 03H, 表示 USB 设备。
15-8	Programming I/F [7:0]	表示 PCI 规范中规定的程序接口 (分类符号)。 该字段固定为 10H, 表示 OHCI 规范 USB。。
7-0	Revision ID [7:0]	表示 USB 版本 (分类符号)。 该字段固定为 42H。

(4) 高速缓存线宽度, 时延定时器, 标题类型和 BIST (偏移地址 0CH)

复位后: 0000 0800H R/W

31	30	29	28	27	26	25	24
BIST7	BIST6	BIST5	BIST4	BIST3	BIST2	BIST1	BIST0
23	22	21	20	19	18	17	16
Header Type7	Header Type6	Header Type5	Header Type4	Header Type3	Header Type2	Header Type1	Header Type0
15	14	13	12	11	10	9	8
Latency Timer7	Latency Timer6	Latency Timer5	Latency Timer4	Latency Timer3	Latency Timer2	Latency Timer1	Latency Timer0
7	6	5	4	3	2	1	0
Cache Line Size7	Cache Line Size6	Cache Line Size5	Cache Line Size4	Cache Line Size3	Cache Line Size2	Cache Line Size1	Cache Line Size0

位的位置	位名称	功能
31-24	BIST[7:0]	自检字段。 该字段固定为 00H。 这些位为只读属性。
23-16	Header Type [7:0]	向系统报告标题类型的字段。 因为是 PCI 设备, 该字段固定为 00H。 特别需要说明, 因为不支持多功能, 第 23 位固定为“0”。 这些位为只读属性。
15-8	Latency Timer[7:0]	向系统报告时延定时器的字段。 最低 2 位固定为“00”。 这些位为只读属性。
7-0	Cache Line Size[7:0]	向系统报告高速缓存线宽度的字段。 该字段固定为“00”。 这些位为只读属性。

(5) 基地址 (偏移地址 10H)

复位后: 0000 0000H R/W

31	30	29	28	27	26	25	24
OHCI Base Address27	OHCI Base Address26	OHCI Base Address25	OHCI Base Address24	OHCI Base Address23	OHCI Base Address22	OHCI Base Address21	OHCI Base Address20
23	22	21	20	19	18	17	16
OHCI Base Address19	OHCI Base Address18	OHCI Base Address17	OHCI Base Address16	OHCI Base Address15	OHCI Base Address14	OHCI Base Address13	OHCI Base Address12
15	14	13	12	11	10	9	8
OHCI Base Address11	OHCI Base Address10	OHCI Base Address9	OHCI Base Address8	OHCI Base Address7	OHCI Base Address6	OHCI Base Address5	OHCI Base Address4
7	6	5	4	3	2	1	0
OHCI Base Address3	OHCI Base Address2	OHCI Base Address1	OHCI Base Address0	Prefetchable	Type1	Type0	Memory Space Indicator

位的位置	位名称	功能
31-4	OHCI Base Address [27:0]	通过第 31 位至第 12 位指定操作寄存器地址。 在初始化时设置系统定义的操作寄存器基地址。 第 11 位至第 4 位固定为 00H。 第 11 位至第 4 位为只读属性。
3	Prefetchable	该字段表示由基地址指定的字段是一个存储器空间。 该位固定为 0, 表示禁止从该字段预取。 该位只读。
2-1	Type[1:0]	该字段表示基地址类型。该字段固定为 00H。表示 OHCI 基地址字段指定的地址是 32 位空间的特定位置。 这些位为只读属性。
0	Memory Space Indicator	该字段表示由基地址指定的字段是一个存储器空间。 该位固定为 0。 该位只读。

(6) 子系统厂商 ID 和子系统 ID (偏移地址 2CH)

复位后: 0133 1033H R

31	30	29	28	27	26	25	24
SubSystem ID15	SubSystem ID14	SubSystem ID13	SubSystem ID12	SubSystem ID11	SubSystem ID10	SubSystem ID9	SubSystem ID8
23	22	21	20	19	18	17	16
SubSystem ID7	SubSystem ID6	SubSystem ID5	SubSystem ID4	SubSystem ID3	SubSystem ID2	SubSystem ID1	SubSystem ID0
15	14	13	12	11	10	9	8
SubSystem Vender ID15	SubSystem Vender ID14	SubSystem Vender ID13	SubSystem Vender ID12	SubSystem Vender ID11	SubSystem Vender ID10	SubSystem Vender ID9	SubSystem Vender ID8
7	6	5	4	3	2	1	0
SubSystem Vender ID7	SubSystem Vender ID6	SubSystem Vender ID5	SubSystem Vender ID4	SubSystem Vender ID3	SubSystem Vender ID2	SubSystem Vender ID1	SubSystem Vender ID0

位的位置	位名称	功能
31-16	SubSystem ID[15:0]	表示设备类型。 根据 PCI 规范, 为驱动一个设备选择驱动程序。 该字段固定为 0133H。
15-0	SubSystem Vender ID [15:0]	表示设备厂商。 根据 PCI 规范, 为驱动一个设备选择驱动程序。 该字段固定为 1033H。

(7) 容量指针 (偏移地址 34H)

复位后: 0000 0040H R

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
Capability Pointer7	Capability Pointer6	Capability Pointer5	Capability Pointer4	Capability Pointer3	Capability Pointer2	Capability Pointer1	Capability Pointer0

位的位置	位名称	功能
31-10	-	保留 (请确保向这些位写入“0”)
7-0	Capability Pointer[7:0]	该字段固定为 40H。

(8) 中断线, 中断引脚, Min Gnt, Max Latency (偏移地址 3CH)

复位后: 2A01 0100H R/W

31	30	29	28	27	26	25	24
Max Latency7	Max Latency6	Max Latency5	Max Latency4	Max Latency3	Max Latency2	Max Latency1	Max Latency0
23	22	21	20	19	18	17	16
Min Gnt7	Min Gnt6	Min Gnt5	Min Gnt4	Min Gnt3	Min Gnt2	Min Gnt1	Min Gnt0
15	14	13	12	11	10	9	8
Interrupt Pin7	Interrupt Pin6	Interrupt Pin5	Interrupt Pin4	Interrupt Pin3	Interrupt Pin2	Interrupt Pin1	Interrupt Pin0
7	6	5	4	3	2	1	0
Interrupt Line7	Interrupt Line6	Interrupt Line5	Interrupt Line4	Interrupt Line3	Interrupt Line2	Interrupt Line1	Interrupt Line0

位的位置	位名称	功能
31-24	Max Latency[7:0]	表示最大的滞后时间, 固定为 2AH。 这些位为只读属性。
23-16	Min Gnt[7:0]	表示最小的授权时间, 固定为 01H。 这些位为只读属性。
15-8	Interrupt Pin[7:0]	表示中断输出引脚, 因为使用 INTA, 固定为 01H。 这些位为只读属性。
7-0	Interrupt Line[7:0]	表示中断线, 固定为 00H。

(9) 容量标识符, 下一项指针, 电源管理能力 (偏移地址 40H)

复位后: 7E02 0001H R

31	30	29	28	27	26	25	24
PME Support4	PME Support3	PME Support2	PME Support1	PME Support0	D2 Support	D1 Support	Aux Current2
23	22	21	20	19	18	17	16
Aux Current1	Aux Current0	DSI	0	PME CLK	Version2	Version1	Version0
15	14	13	12	11	10	9	8
Next Item Pointer7	Next Item Pointer6	Next Item Pointer5	Next Item Pointer4	Next Item Pointer3	Next Item Pointer2	Next Item Pointer1	Next Item Pointer0
7	6	5	4	3	2	1	0
Capability Identifier7	Capability Identifier6	Capability Identifier5	Capability Identifier4	Capability Identifier3	Capability Identifier2	Capability Identifier1	Capability Identifier0

位的位置	位名称	功能
31	PME Support[4:0]	表示是否支持 D3 冷状态 (电源管理能力)。 因为不支持 D3 冷状态, 字段固定为“0BH”
30-27		表示在所有的 PCI 电源状态 (D0 到 D3) 下支持 PME (电源管理能力)。 该字段固定为“1111”。
26	D2 Support	表示支持 PCI 电源的 D2 状态 (电源管理能力)。 该位固定为“1”。
25	D1 Support	表示支持 PCI 电源的 D1 状态 (电源管理能力)。 该位固定为“1”。
24-22	Aux Current [2:0]	表示不支持在 D3 冷状态下断言 PME 中断 (电源管理能力)。 该字段因此固定为“000”。
21	DSI	表示使用电源管理时无需特殊初始化 (电源管理能力)。 该位固定为“0”。
19	PME CLK	表示产生 PME 中断时不需要 PCLK (电源管理能力)。 该位固定为“0”。
18-16	Version[2:0]	表示电源管理版本 (电源管理能力)。 根据 USB 主机控制器中的电路配置情况, 这些位固定为“010”。
15-8	Next Item Pointer[7:0]	表示下一项不存在 该字段固定为“00H”。
7-0	Capability Identifier[7:0]	表示电源管理注册 ID。 该字段固定为“01H”。

(10) 电源管理控制/状态, PMCSR 桥支持扩展, 数据 (偏移地址 44H)

(1/2)

复位后: 0000 0000H R/W

31	30	29	28	27	26	25	24
Data7	Data6	Data5	Data4	Data3	Data2	Data1	Data0
23	22	21	20	19	18	17	16
BPCC Enable	B2_B3	0	0	0	0	0	0
15	14	13	12	11	10	9	8
PME Status	Data Scale1	Data Scale0	Data Select3	Data Select2	Data Select1	Data Select0	PME Enable
7	6	5	4	3	2	1	0
0	0	0	0	0	0	Power State1	Power State0

位的位置	位名称	功能
31-24	Data[7:0]	在 PCI 规范中, 该字段为可选字段, 且 USB 主机控制器不支持 (数据)。该字段固定为“00H”。这些位为只读属性。
23	BPCC Enable	桥的位, USB 主机控制器不支持 (PMCSR 桥扩展支持)。该字段固定为“0”。该位只读。
22	B2_B3	桥的位, USB 主机控制器不支持 (PMCSR 桥扩展支持)。该字段固定为“0”。该位只读。
15	PME Status	表示 PME 中断状态 (电源管理控制/状态)。当满足断言 PME 的条件时, 该位被置为“1”。通过 PCI 总线写入“1”时, 该位被清除为“0”。
14-13	Data Scale [1:0]	在 PCI 规范中, 该字段为可选字段, 且 USB 主机控制器不支持 (电源管理控制/状态)。该字段固定为“00”。这些位为只读属性。
12-9	Data Select [3:0]	在 PCI 规范中, 该字段为可选字段, 且 USB 主机控制器不支持 (电源管理控制/状态)。该字段固定为“0000”。这些位为只读属性。
8	PME Enable	该位决定是否使用外部引脚 PME 的设置。该位被置为 1B 时, 从电源管理模式返回, 就回产生一个 PME 中断。

位的位置	位名称	功能		
1-0	Power State[1:0]	表示 PCI 电源状态 (电源管理控制/状态)。 状态变量取决于该字段的设置。		
		电源状态1	电源状态0	字段状况
		0	0	D0 状态
		0	1	D1 状态
		1	0	D2 状态
1	1	D3 热状态		

21.7.3 OHCI 操作寄存器

OHCI 操作寄存器包含在 OHCI 主机控制器中，包括表 21-5 列出的寄存器。详情参见 OpenHCI 规范 1.0a 版本。

在 OpenHCI 规范 1.0a 版本中，按照[1: 端口数量]来定义端口数量，本节也遵照这种方式。例如，端口 1 对应主机通道 0，端口 2 对应主机通道 1。关于偏移地址，参见 21.2.2 USB 主机控制器存储器映射。

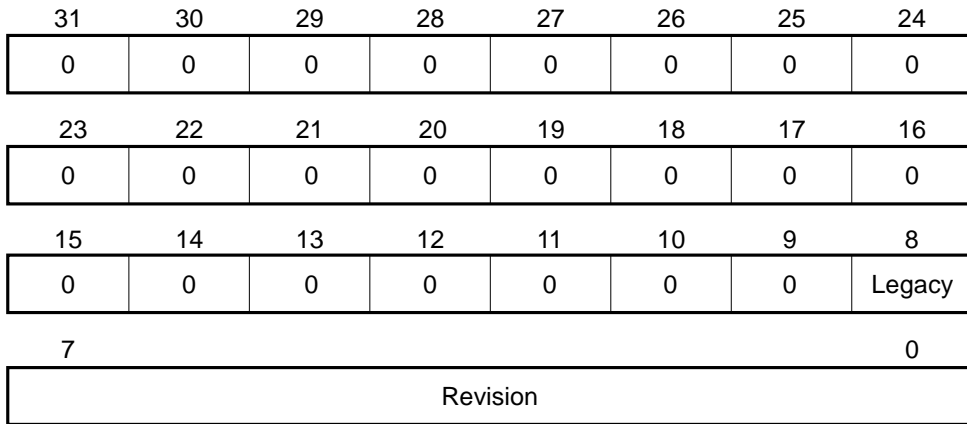
表21-5. OHCI操作寄存器

地址	位	31	24	23	16	15	8	7	0	
00H		HcRevision								
04H		HcControl								
08H		HcCommandStatus								
0CH		HcInterruptStatus								
10H		HcInterruptEnable								
14H		HcInterruptDisable								
18H		HcHCCA								
1CH		HcPeriodCurrentED								
20H		HcControlHeadED								
24H		HcControlCurrentED								
28H		HcBulkHeadED								
2CH		HcBulkCurrentED								
30H		HcDoneHead								
34H		HcFmInterval								
38H		HcFmRemaining								
3CH		HcFmNumber								
40H		HcPeriodicStart								
44H		HcLSThreshold								
48H		HcRhDescriptorA								
4CH		HcRhDescriptorB								
50H		HcRhStatus								
54H		HcRhPortStatus1								
58H		HcRhPortStatus2								
5CH 至 FFH		保留								

- 注
- HC: 主机控制器 (OHCI 主机控制器)
 - HCD: 主机控制器驱动
 - ED: 端点描述符
 - TD: 传输描述符
 - EOP: 包的结尾
 - SOF: 帧的开始

(1) HcRevision 寄存器 (偏移地址 00H)

复位后: 0000 0010H R



位的位置	位名称	功能
31-9	-	保留 (请确保向这些位写入 “0”)。
8	Legacy	表示支持遗存的寄存器是否在 OHCI 主机控制器中完成。 因为 OHCI 主机控制器不支持遗存功能, 该位固定为“0”
7-0	Revision	表示在 OHCI 主机控制器中完成的 OHCI 协议版本。 因为 OHCI 主机控制器遵照 OHCI 规范 1.0a 版本, 该字段固定为 10H。

(2) HcControl 寄存器 (偏移地址 04H)

(1/2)

复位后: 0000 0000H R/W

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	0	RWE	RWC	IR
7	6	5	4	3	2	1	0
HCFS1	HCFS0	BLE	CLE	IE	PLE	CBSR1	CBSR0

位的位置	位名称	功能										
31-11	-	保留 (请确保向这些位写入“0”)										
10	RWE	远程唤醒使能。 设置上行恢复信号的检测。 1: 检测恢复信号作为远程唤醒。 0: 不检测恢复信号作为远程唤醒。										
9	RWC	远程唤醒连接。 表示 OHCI 主机控制器是否支持远程唤醒。 如果系统支持远程唤醒, 在初始化时必须设置该位。 1: 支持远程唤醒。 0: 不支持远程唤醒。										
8	IR	中断路由 表示通过 OHCI 主机控制器输出中断的路由。 设置如何向 HcInterruptStatus 寄存器报告已经产生的中断源。 1: 通过 SMMI 报告中断的产生。 0: 报告 INTA 中断的产生。										
7-6	HCFS[1:0]	主机控制器功能状态 表示 OHCI 主机控制器操作状态。 <table border="1" data-bbox="535 1438 1266 1638"> <thead> <tr> <th>HCFS</th> <th>USB 状态</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>USB 复位</td> </tr> <tr> <td>01</td> <td>USB 恢复</td> </tr> <tr> <td>10</td> <td>USB 操作</td> </tr> <tr> <td>11</td> <td>USB 挂起</td> </tr> </tbody> </table> <p>当 OHCI 主机控制器切换到 USB 操作状态, 将会在 1ms 单位内启动帧的管理。 该状态总是由主机控制器驱动器 (HCD) 来控制, 除非当 USB 挂起时, 通过远程唤醒将 USB 切换到 USB 恢复状态。 在硬件复位后, 该字段被置为“00”, 软件复位后被设置为“11”。</p>	HCFS	USB 状态	00	USB 复位	01	USB 恢复	10	USB 操作	11	USB 挂起
HCFS	USB 状态											
00	USB 复位											
01	USB 恢复											
10	USB 操作											
11	USB 挂起											

位的位置	位名称	功能										
5	BLE	<p>块列表使能 设置是否执行块列表处理。 1: 执行块列表处理。 0: 不执行块列表处理。 从块列表处理的下一帧开始, 该位的设置生效。 当修改块列表时, 该位必须设置为“0”。</p>										
4	CLE	<p>控制列表使能 设置是否执行控制列表处理。 1: 执行控制列表处理。 0: 不执行控制列表处理。 从控制列表处理的下一帧开始, 该位的设置生效。 当修改控制列表时, 该位必须设置为“0”。</p>										
3	IE	<p>同步使能 设置是否执行同步 ED 处理。 在列表处理期间, 当检测到同步 ED 时, 该位被检查, 然后决定是否执行同步 ED 处理。 1: 执行同步传输处理。 0: 不执行同步传输处理。 从同步传输处理的下一帧开始, 该位的设置生效。</p>										
2	PLE	<p>周期列表使能 设置是否执行周期列表处理。 1: 执行周期列表处理。 0: 不执行周期列表处理。 从周期列表处理的下一帧开始, 该位的设置生效。</p>										
1-0	CBSR[1:0]	<p>控制 块服务比例 定义控制传输和批量传输服务的比例。 在执行传输的周期列表处理期间, 该字段中定义的比例保持不变。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>CBSR</th> <th>块ED和控制ED的服务比例</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>1:1</td> </tr> <tr> <td>01</td> <td>2:1</td> </tr> <tr> <td>10</td> <td>3:1</td> </tr> <tr> <td>11</td> <td>4:1</td> </tr> </tbody> </table>	CBSR	块ED和控制ED的服务比例	00	1:1	01	2:1	10	3:1	11	4:1
CBSR	块ED和控制ED的服务比例											
00	1:1											
01	2:1											
10	3:1											
11	4:1											

(3) HcCommandStatus 寄存器 (偏移地址 08H)

(1/2)

复位后: 0000 0000H R/W

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	SOC1	SOC0
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	OCR	BLF	CLF	HCR

位的位置	位名称	功能
31-18	-	保留 (请确保向这些位写入“0”)。
17-16	SOC[1:0]	调度超时计数 对调度超时的数量计数。每次发生超时就向上计数, 即使 HcInterruptStatus 寄存器的 SO 位被置为 1, 也会继续向上计数。 这些位为只读属性。
15-4	-	保留 (请确保向这些位写入“0”)。
3	OCR	所有权变更请求 请求改变 OHCI 主机控制器的控制权限。
2	BLF	块列表填充 表示在块列表中是否存在 TD。 当 TD 被加入到块列表的 ED 时, 该位总是被主机控制器驱动器 (HCD) 置为“1”。 当启动块列表头的处理时, OHCI 主机控制器检查该位。 当该位为“0”时, 如果存在一个 TD, 不会启动列表头处理。如果该位被设置为“1”, OHCI 主机控制器把它设置为“0”, 并且启动控制 ED 处理。如果在块列表中检测到一个 TD, 该位再次被置为“1”, 以便继续块 TD 处理。 在重新配置列表之前, 驱动器必须设置该位, 设置 HcCommand 寄存器的 BLE 位, 且启动列表处理。
1	CLF	控制列表的填充 表示控制列表是否存在。 当 TD 被加入到块列表的 ED 时, 该位总是被主机控制器驱动器 (HCD) 置为“1”。 当启动控制列表头的处理时, OHCI 主机控制器检查该位。 当该位为“0”时, 不会启动列表头处理。如果该位被设置为“1”, OHCI 主机控制器把它设置为“0”, 并且启动控制列表处理。如果在控制列表中检测到一个 TD, 该位再次被置为“1”, 以便继续控制列表的处理。 在重新配置列表之前, 驱动器必须设置该位, 设置 HcCommand 寄存器的 CLE 位, 且启动列表处理。

位的位置	位名称	功能
0	HCR	主机控制器复位 通过 OHCI 主机控制器激活软件复位。 当该位被置位时，无论 OHCI 主机控制器的功能状态如何，进入 USB 挂起状态。 当复位操作完成时，通过 OHCI 主机控制器对该位清除 (0)。

(4) HcInterruptStatus 寄存器 (偏移地址 0CH)

(1/2)

复位后: 0000 0000H R/W

31	30	29	28	27	26	25	24
0	OC	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	RHSC	FNO	UE	RD	SF	WDH	SO

位的位置	位名称	功能
31	-	保留 (请确保向这些位写入“0”)
30	OC	所有权变更 表示一个所有权变更中断的产生。 当 HcCommandStatus 寄存器的 OCR 字段被置“1”时, 该位被置位 (1)。 如果中断源没有被屏蔽, SMMI 中断产生。 1: OC 中断产生 0: OC 中断没有产生 向该位写入“1”, 可以清除中断源。
29-7	-	保留 (请确保向这些位写入“0”)
6	RHSC	根集线器状态改变 表示 HcRhPortStatus 寄存器的设置已经被改变。 由于硬件原因导致 HcRhPortStatus 寄存器的状态发生变化时, 该位置为“1”。 1: RHSC 中断产生 0: RHSC 中断没有产生 向该位写入“1”, 可以清除中断源。
5	FNO	帧编号溢出 表示帧编号的 MSB 已经被改变。 当帧中的 HccaFrameNumber 被更新, 它的帧编号的 MSB 从“0”变为“1”或者从“1”变为“0”之后, 该位被置为“1”。 1: FNO 中断产生 0: FNO 中断没有产生 向该位写入“1”, 可以清除中断源。
4	UE	不可恢复的错误 表示在 PCI 总线上检测到的系统错误和 USB 没有关系。 1: UE 中断产生 0: UE 中断没有产生 向该位写入“1”, 可以清除中断源。

位的位置	位名称	功能
3	RD	<p>恢复检测</p> <p>表示检测到恢复。</p> <p>USB 总线上通过设备检测到恢复信号时，该位被置“1”。如果 USB 恢复是通过主机控制器驱动器 (HCD) 发起的，该位不置位。</p> <p>1: RD 中断产生</p> <p>0: RD 中断没有产生</p> <p>向该位写入“1”，可以清除中断为“0”。</p>
2	SF	<p>帧的开始</p> <p>表示在帧的开始处 HccaFrameNumber 已经被更新。OHCI 主机控制器发送 SOF 包，且更新 HccaFrameNumber。</p> <p>1: SF 中断产生</p> <p>0: SF 中断没有产生</p> <p>向该位写入“1”，可以清除中断为“0”。</p>
1	WDH	<p>回写 Done Head</p> <p>表示 USB 主机控制器更新 HccDoneHead，更新 HccDoneHead 之后，OHCI 主机控制器立即把该位设置为“1”。HccaDoneHead 不会进行更新，直到该位被清除为“0”。</p> <p>1: WDH 中断产生</p> <p>0: WDH 中断没有产生</p> <p>向该位写入“1”，可以清除中断为“0”。</p>
0	SO	<p>调度溢出</p> <p>表示帧中的 USB 调度溢出。在 USB 调度溢出后的帧中，帧的 FrameNumberUpdate 被更新，该位被置为“1”。当该位被设置为“1”时，HcCommandStatus 寄存器的 SOC 字段递增。</p> <p>1: SO 中断产生</p> <p>0: SO 中断没有产生</p> <p>向该位写入“1”，可以清除中断为“0”。</p>

(5) HcInterruptEnable 寄存器 (偏移地址 10H)

(1/2)

复位后: 0000 0000H R/W

31	30	29	28	27	26	25	24
MIE	OCE	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	RHSCE	FNOE	UEE	RDE	SFE	WDHE	SOE

位的位置	位名称	功能
31	MIE	主设备中断使能 指定是否使能由第 30 位至第 6 位设置的中断源为 0。 1: 使能所有指定的中断源 0: 忽略 要清除该位为“0”，向 HcInterruptDisable 寄存器的对应位写入“1”。
30	OCE	所有权变更使能 确定是否使能 OC 作为一个中断源。 1: 使能 OC 作为中断源 0: 忽略 向该位写入“1”，可以设置 OC 作为中断源。 要清除该位为“0”，向 HcInterruptDisable 寄存器对应位写入“1”。
29-7	-	保留 (请确保向这些位写入“0”)
6	RHSCE	根集线器状态改变使能 指定是否使能 RHSC 作为中断源。 1: 使能 RHSC 作为中断源 0: 忽略 向该位写入“1”，可以设置 RHSC 作为中断源。 要清除该位为“0”，向 HcInterruptDisable 寄存器对应位写入“1”。
5	FNOE	帧编号溢出使能 指定是否使能 FNO 作为中断源。 1: 使能 FNO 作为中断源 0: 忽略 向该位写入“1”，可以设置 FNO 作为中断源。 要清除该位为“0”，向 HcInterruptDisable 寄存器对应位写入“1”。
4	UEE	不可恢复的错误使能 指定是否使能 UE 作为中断源。 1: 使能 UE 作为中断源 0: 忽略 向该位写入“1”，可以设置 UE 作为中断源。 要清除该位为“0”，向 HcInterruptDisable 寄存器对应位写入“1”。

位的位置	位名称	功能
3	RDE	<p>恢复检测使能</p> <p>指定是否使能 RD 作为中断源。</p> <p>1: 使能 RD 作为中断源</p> <p>0: 忽略</p> <p>向该位写入“1”，可以设置 RD 作为中断源。</p> <p>要清除该位为“0”，向 HcInterruptDisable 寄存器对应位写入“1”。</p>
2	SFE	<p>开始帧使能</p> <p>指定是否使能 SF 作为中断源。</p> <p>1: 使能 SF 作为中断源</p> <p>0: 忽略</p> <p>向该位写入“1”，可以设置 SF 作为中断源。</p> <p>要清除该位为“0”，向 HcInterruptDisable 寄存器对应位写入“1”。</p>
1	WDHE	<p>回写 Done Head 使能</p> <p>指定是否使能 WDH 作为中断源。</p> <p>1: 使能 WDH 作为中断源</p> <p>0: 忽略</p> <p>向该位写入“1”，可以设置 WDH 作为中断源。</p> <p>要清除该位为“0”，向 HcInterruptDisable 寄存器对应位写入“1”。</p>
0	SOE	<p>调度溢出使能</p> <p>指定是否使能 SO 作为中断源。</p> <p>1: 使能 SO 作为中断源</p> <p>0: 忽略</p> <p>向该位写入“1”，可以设置 SO 作为中断源。</p> <p>要清除该位为“0”，向 HcInterruptDisable 寄存器对应位写入“1”。</p>

(6) HcInterruptDisable 寄存器 (偏移地址 14H)

(1/2)

复位后: 0000 0000H R/W

31	30	29	28	27	26	25	24
MID	OCD	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	RHSCD	FNOD	UED	RDD	SFD	WDHD	SOD

位的位置	位名称	功能
31	MID	主设备中断禁止 设置是否禁止由 HcInterruptEnable 寄存器的第 30 位至第 6 位设置的中断源为 0。 1: 禁止所有指定的中断源 0: 忽略 当读取该位时, HcInterruptEnable 寄存器对应位的值被读取。 要设置该位为“1”, 向 HcInterruptEnable 寄存器的对应位写入“1”。
30	OCD	所有权变更禁止 设置是否从中断源中移除 OC。 1: 禁止 OC 作为中断源 0: 忽略 当读取该位时, HcInterruptEnable 寄存器对应位的值被读取。向该位写入“1”时, 从中断源中清除 OC。 要设置该位为“1”, 向 HcInterruptEnable 寄存器的对应位写入“1”。
29-7	-	保留 (请确保向这些位写入“0”)
6	RHSCD	根集线器状态改变禁止 设置是否从中断源中移除 RHSC。 1: 禁止 RHSC 作为中断源 0: 忽略 当读取该位时, HcInterruptEnable 寄存器对应位的值被读取。向该位写入“1”时, 从中断源中清除 RHSC。 要设置该位为“1”, 向 HcInterruptEnable 寄存器的对应位写入“1”。
5	FNOD	帧的编号溢出禁止 设置是否从中断源中移除 FNO。 1: 禁止 FNO 作为中断源 0: 忽略 当读取该位时, HcInterruptEnable 寄存器对应位的值被读取。向该位写入“1”时, 从中断源中清除 FNO。 要设置该位为“1”, 向 HcInterruptEnable 寄存器的对应位写入“1”。

位的位置	位名称	功能
4	UED	<p>不可恢复的错误禁止 设置是否从中断源中移除 UE。</p> <p>1: 禁止 UE 作为中断源 0: 忽略</p> <p>当读取该位时, HcInterruptEnable 寄存器对应位的值被读取。向该位写入“1”时, 从中断源中清除 UE。</p> <p>要设置该位为“1”, 向 HcInterruptEnable 寄存器的对应位写入“1”。</p>
3	RDD	<p>复位检测禁止 设置是否从中断源中移除 RD。</p> <p>1: 禁止 RD 作为中断源 0: 忽略</p> <p>当读取该位时, HcInterruptEnable 寄存器对应位的值被读取。向该位写入“1”时, 从中断源中清除 RD。</p> <p>要设置该位为“1”, 向 HcInterruptEnable 寄存器的对应位写入“1”。</p>
2	SFD	<p>开始帧禁止 设置是否从中断源中移除 SF。</p> <p>1: 禁止 SF 作为中断源 0: 忽略</p> <p>当读取该位时, HcInterruptEnable 寄存器对应位的值被读取。向该位写入“1”时, 从中断源中清除 SF。</p> <p>要设置该位为“1”, 向 HcInterruptEnable 寄存器的对应位写入“1”。</p>
1	WDHD	<p>回写 Done Head 禁止 设置是否从中断源中移除 WDH。</p> <p>1: 禁止 WDH 作为中断源 0: 忽略</p> <p>当读取该位时, HcInterruptEnable 寄存器对应位的值被读取。向该位写入“1”时, 从中断源中清除 WDH。</p> <p>要设置该位为“1”, 向 HcInterruptEnable 寄存器的对应位写入“1”。</p>
0	SOD	<p>调度溢出禁止 设置是否从中断源中移除 SO。</p> <p>1: 禁止 SO 作为中断源 0: 忽略</p> <p>当读取该位时, HcInterruptEnable 寄存器对应位的值被读取。向该位写入“1”时, 从中断源中清除 SO。</p> <p>要设置该位为“1”, 向 HcInterruptEnable 寄存器的对应位写入“1”。</p>

(7) HcHCCA 寄存器 (偏移地址 18H)

复位后: 0000 0000H R/W

31	30	29	28	27	26	25	24
HcHCCA23	HcHCCA22	HcHCCA21	HcHCCA20	HcHCCA19	HcHCCA18	HcHCCA17	HcHCCA16
23	22	21	20	19	18	17	16
HcHCCA15	HcHCCA14	HcHCCA13	HcHCCA12	HcHCCA11	HcHCCA10	HcHCCA9	HcHCCA8
15	14	13	12	11	10	9	8
HcHCCA7	HcHCCA6	HcHCCA5	HcHCCA4	HcHCCA3	HcHCCA2	HcHCCA1	HcHCCA0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

位的位置	位名称	功能
31-8	HcHCCA [23:0]	设置分配给 OHCI 主机控制器通信空间的 RAM 基地址。 该字段必须在初始化期间设置。OHCI 主机控制器请求从指定基地址开始的 256 字节区域作为 HCCA。

(8) HcPeriodCurrentED 寄存器 (偏移地址 1CH)

复位后: 0000 0000H R

31	30	29	28	27	26	25	24
Period CurrentED27	Period CurrentED26	Period CurrentED25	Period CurrentED24	Period CurrentED23	Period CurrentED22	Period CurrentED21	Period CurrentED20
23	22	21	20	19	18	17	16
Period CurrentED19	Period CurrentED18	Period CurrentED17	Period CurrentED16	Period CurrentED15	Period CurrentED14	Period CurrentED13	Period CurrentED12
15	14	13	12	11	10	9	8
Period CurrentED11	Period CurrentED10	Period CurrentED9	Period CurrentED8	Period CurrentED7	Period CurrentED6	Period CurrentED5	Period CurrentED4
7	6	5	4	3	2	1	0
Period CurrentED3	Period CurrentED2	Period CurrentED1	Period CurrentED0	0	0	0	0

位的位置	位名称	功能
31-4	PeriodCurre -ntED[27:0]	表示周期列表中包含的处理地址。 每次在周期列表处理序列结束时, OHCI 主机控制器更新该字段的值。

(9) HcControlHeadED 寄存器 (偏移地址 20H)

复位后: 0000 0000H R/W

31	30	29	28	27	26	25	24
Control HeadED27	Control HeadED26	Control HeadED25	Control HeadED24	Control HeadED23	Control HeadED22	Control HeadED21	Control HeadED20
23	22	21	20	19	18	17	16
Control HeadED19	Control HeadED18	Control HeadED17	Control HeadED16	Control HeadED15	Control HeadED14	Control HeadED13	Control HeadED12
15	14	13	12	11	10	9	8
Control HeadED11	Control HeadED10	Control HeadED9	Control HeadED8	Control HeadED7	Control HeadED6	Control HeadED5	Control HeadED4
7	6	5	4	3	2	1	0
Control HeadED3	Control HeadED2	Control HeadED1	Control HeadED0	0	0	0	0

位的位置	位名称	功能
31-4	ControlHeadED[27:0]	确定控制传输中列表 ED 的起始地址。 要执行控制传输, 必须在 HcControl 寄存器的 CLE 位设置之前设置该字段 (1)。

(10) HcControlCurrentED 寄存器 (偏移地址 24H)

复位后: 0000 0000H R/W

31	30	29	28	27	26	25	24
Control CurrentED27	Control CurrentED26	Control CurrentED25	Control CurrentED24	Control CurrentED23	Control CurrentED22	Control CurrentED21	Control CurrentED20
23	22	21	20	19	18	17	16
Control CurrentED19	Control CurrentED18	Control CurrentED17	Control CurrentED16	Control CurrentED15	Control CurrentED14	Control CurrentED13	Control CurrentED12
15	14	13	12	11	10	9	8
Control CurrentED11	Control CurrentED10	Control CurrentED9	Control CurrentED8	Control CurrentED7	Control CurrentED6	Control CurrentED5	Control CurrentED4
7	6	5	4	3	2	1	0
Control CurrentED3	Control CurrentED2	Control CurrentED1	Control CurrentED0	0	0	0	0

位的位置	位名称	功能
31-4	ControlCurrentED[27:0]	表示控制列表中包含的处理地址。 每次在控制 ED 处理序列结束时, OHCI 主机控制器更新该字段的值。当配置一个新的列表时, 该字段被设置为 00000000H, 表示该列表的结束。 当挂起或者恢复传输时, 必须保证该字段表示的 ED 是存在的。

(11) HcBulkHeadED 寄存器 (偏移地址 28H)

复位后: 0000 0000H R/W

31	30	29	28	27	26	25	24
BulkHead ED27	BulkHead ED26	BulkHead ED25	BulkHead ED24	BulkHead ED23	BulkHead ED22	BulkHead ED21	BulkHead ED20
23	22	21	20	19	18	17	16
BulkHead ED19	BulkHead ED18	BulkHead ED17	BulkHead ED16	BulkHead ED15	BulkHead ED14	BulkHead ED13	BulkHead ED12
15	14	13	12	11	10	9	8
BulkHead ED11	BulkHead ED10	BulkHead ED9	BulkHead ED8	BulkHead ED7	BulkHead ED6	BulkHead ED5	BulkHead ED4
7	6	5	4	3	2	1	0
BulkHead ED3	BulkHead ED2	BulkHead ED1	BulkHead ED0	0	0	0	0

位的位置	位名称	功能
31-4	BulkHeadED [27:0]	指定批量传输中列表 ED 的起始地址。 要执行批量传输, 必须在 HcControl 寄存器的 BLE 位设置之前设置该字段 (1)。

(12) HcBulkCurrentED 寄存器 (偏移地址 2CH)

复位后: 0000 0000H R/W

31	30	29	28	27	26	25	24
BulkCurrent ED27	BulkCurrent ED26	BulkCurrent ED25	BulkCurrent ED24	BulkCurrent ED23	BulkCurrent ED22	BulkCurrent ED21	BulkCurrent ED20
23	22	21	20	19	18	17	16
BulkCurrent ED19	BulkCurrent ED18	BulkCurrent ED17	BulkCurrent ED16	BulkCurrent ED15	BulkCurrent ED14	BulkCurrent ED13	BulkCurrent ED12
15	14	13	12	11	10	9	8
BulkCurrent ED11	BulkCurrent ED10	BulkCurrent ED9	BulkCurrent ED8	BulkCurrent ED7	BulkCurrent ED6	BulkCurrent ED5	BulkCurrent ED4
7	6	5	4	3	2	1	0
BulkCurrent ED3	BulkCurrent ED2	BulkCurrent ED1	BulkCurrent ED0	0	0	0	0

位的位置	位名称	功能
31-4	BulkCurrent- ED[27:0]	表示正在执行块列表处理的地址。 每次在块 ED 处理序列结束时, OHCI 主机控制器更新本字段的值。当配置一个新的列表, 该字段设置为 00000000H, 表示该列表的结束。 当挂起或者恢复传输时, 必须保证该字段表示的 ED 是存在的。

(13) HcDoneHead 寄存器 (偏移地址 30H)

复位后: 0000 0000H R

31	30	29	28	27	26	25	24
Done Head27	Done Head26	Done Head25	Done Head24	Done Head23	Done Head22	Done Head21	Done Head20
23	22	21	20	19	18	17	16
Done Head19	Done Head18	Done Head17	Done Head16	Done Head15	Done Head14	Done Head13	Done Head12
15	14	13	12	11	10	9	8
Done Head11	Done Head10	Done Head9	Done Head8	Done Head7	Done Head6	Done Head5	Done Head4
7	6	5	4	3	2	1	0
Done Head3	Done Head2	Done Head1	Done Head0	0	0	0	0

位的位置	位名称	功能
31-4	DoneHead [27:0]	表示OHCI主机控制器的HcDoneHead的地址。

(14) HcFmInterval 寄存器 (偏移地址 34H)

复位后: 0000 2EDFH R/W

31	30	29	28	27	26	25	24
FIT	FSMPS14	FSMPS13	FSMPS12	FSMPS11	FSMPS10	FSMPS9	FSMPS8
23	22	21	20	19	18	17	16
FSMPS7	FSMPS6	FSMPS5	FSMPS4	FSMPS3	FSMPS2	FSMPS1	FSMPS0
15	14	13	12	11	10	9	8
0	0	FI13	FI12	FI11	FI10	FI9	FI8
7	6	5	4	3	2	1	0
FI7	FI6	FI5	FI4	FI3	FI2	FI1	FI0

位的位置	位名称	功能
31	FIT	帧间隔切换 用于同步 HCD 和 OHCI 主机控制器的帧设置值。 通过 HCD 向 FI 字段写入数据时, 反转该位的设置。 加载 FI 字段时, OHCI 主机控制器把 FIT 值反映到 HcFmRemaining 寄存器的 FRT 位。 HCD 通过比较写入 FI 字段的 FIT 位的值和从 FRT 位读出的值, 来检测是否有新的 FI 字段值被反映。
30-16	FSMPS [14:0]	FS 最大数据包 设置不会导致调度溢出的最大发送或者接收的数据量。比较当前的帧位置和设置的数值, 判断出可以传送的帧的位置。最终的结果取决于系统总线性能, 因此该数值由 HCD 设置。
15-14	-	保留 (请确保向这些位写入 “0”)
13-0	FI[13:0]	帧间隔 设置全速模式下连续两个 SOF 的间隔时间。 根据 USB 规范的规定, 该字段必须设置为 2EDFH, 以满足一个帧的长度 (=1ms)。

(15) HcFmRemaining 寄存器 (偏移地址 38H)

复位后: 0000 2EDFH R

31	30	29	28	27	26	25	24
FRT	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	FR13	FR12	FR11	FR10	FR9	FR8
7	6	5	4	3	2	1	0
FR7	FR6	FR5	FR4	FR3	FR2	FR1	FR0

位的位置	位名称	功能
31	FRT	帧保留切换 用于同步 HCD 和 OHCI 主机控制器的帧设置值。 当 FR 字段被设置为“0H”且 FI 字段数值被重新加载时, OHCI 主机控制器把 FIT 字段的数值拷贝到该位。 通过比较 FIT 和 FRT 位的数值, HCD 可以检查 FI 字段的设置值是否设置到 FR 字段。
13:0	FR[13:0]	帧保留 表示当前帧数值。 该字段的数值随着时间的流逝而向下减少。 因为当 FI 字段的数值变为“0H”时, 帧数值会被重新加载, FR 字段数值被拷贝到该字段, 且再次开始向下计数。

(16) HcFmNumber 寄存器 (偏移地址 3CH)

复位后: 0000 0000H R

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
Frame Number15	Frame Number14	Frame Number13	Frame Number12	Frame Number11	Frame Number10	Frame Number9	Frame Number8
7	6	5	4	3	2	1	0
Frame Number7	Frame Number6	Frame Number5	Frame Number4	Frame Number3	Frame Number2	Frame Number1	Frame Number0

位的位置	位名称	功能
15-0	FrameNumber [15:0]	表示已发送的帧的数量。 当 FR 字段的数值为“0H”时, 该字段向上计数。

(17) HcPeriodicStart 寄存器 (偏移地址 40H)

复位后: 0000 0000H R/W

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	Periodic Start13	Periodic Start12	Periodic Start11	Periodic Start10	Periodic Start9	Periodic Start8
7	6	5	4	3	2	1	0
Periodic Start7	Periodic Start6	Periodic Start5	Periodic Start4	Periodic Start3	Periodic Start2	Periodic Start1	Periodic Start0

位的位置	位名称	功能
31-14	-	保留 (请确保向这些位写入“0”)
13-0	PeriodicStart [13:0]	用于确定在 PeriodicStart 字段帧中存储的周期传输和异步传输的比例。 该字段必须在 OHCI 主机控制器初始化期间由 HCD 进行设置。 如果 FR 字段的值比该字段的值大, 非周期列表传输将优先于周期列表传输。 OHCI 规范推荐该字段的值设置为 FI 字段值的 10%, 且相关值为 3E67H。

(18) HcLSThreshold 寄存器 (偏移地址 44H)

复位后: 0000 0628H R/W

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	HcLSThres -hold11	HcLSThres -hold10	HcLSThres -hold9	HcLSThres -hold8
7	6	5	4	3	2	1	0
HcLSThres -hold7	HcLSThres -hold6	HcLSThres -hold5	HcLSThres -hold4	HcLSThres -hold3	HcLSThres -hold2	HcLSThres -hold1	HcLSThres -hold0

位的位置	位名称	功能
31-12	-	保留 (请确保向这些位写入“0”)
11-0	HcLSThres- hold[11:0]	LS 传输期间, 在帧的剩余时间内, 指定传输是否可用的阈值。 如果 FR 字段的值大于该字段设置的值, LS 传输可以启动。

(19) HRDA (HcRhDescriptorA) 寄存器 (偏移地址 48H)

(1/2)

复位后: FF00 0902H R/W

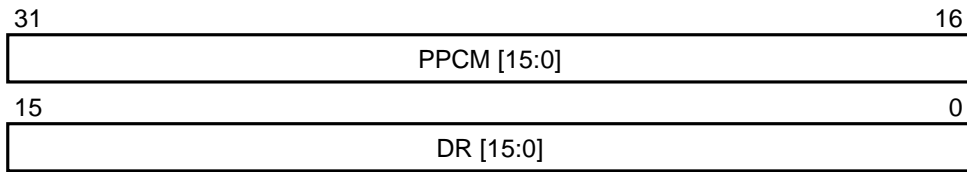
	31	30	29	28	27	26	25	24
HRDA	POTPGT7	POTPGT6	POTPGT5	POTPGT4	POTPGT3	POTPGT2	POTPGT1	POTPGT0
	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	0	0	0	NOCP	OCPM	DT	NPS	PSM
	7	6	5	4	3	2	1	0
	NDP7	NDP6	NDP5	NDP4	NDP3	NDP2	NDP1	NDP0

位的位置	位名称	功能
31-24	POTPGT [7:0]	电源打开到电源稳定的时间 定义上电后主机控制器驱动器 (HCD) 访问根集线器的等待时间 (单位: 2ms)。等待时间为 POTPGT 字段的值 × 2ms。
23-13	-	保留 (请确保向这些位写入 "0")
12	NOCP	无过流保护 定义是否支持根集线器过流保护功能。 1: 不支持过电流保护状态。 0: 支持过电流保护状态。
11	OCPM	过流保护模式 定义如何报告根集线器的过电流状态。 该位设置的模式必须和 PSM 位设置的模式相同。 1: 报告端口单元的过电流状态。 0: 同时报告所有端口的过电流状态。 只有当 NOCP 位被清除为 "0" 时, 该位设置才有效。
10	DT	设备类型 表示根集线器不是一个组合设备。 因为根集线器必须不是一个组合设备, 该字段总是返回 "0"。 该位只读。
9	NPS	无电源开关 定义是否支持电源开关, 或者端口持续供电。 1: 当 OHCI 主机控制器运行时, 一直供电。 0: 对端口的电源可以开关。

位的位置	位名称	功能
8	PSM	<p>电源开关模式</p> <p>定义如何控制根集线器电源开关。</p> <p>1: 端口电源单独控制。</p> <p>0: 所有端口电源同时控制。</p> <p>如果 HcRhDescriptorB 寄存器的 PPCM 字段被置位 (1)，端口只能通过 HcRhPortStatus 寄存器的 SPP 和 CPP 位来控制。如果它被清除为“0”，端口受 SGP 和 CGP 位的控制。只有当 NPS 位被清除为“0”时，该位的设置生效。</p>
7-0	NDP[7:0]	<p>下行端口数量</p> <p>定义 OHCI 主机控制器的根集线器支持的下行端口数量。</p> <p>由于 OHCI 主机控制器内置两个下行端口，该字段固定为 02H。</p> <p>这些位为只读属性。</p>

(20) HcRhDescriptorB 寄存器 (偏移地址 4CH)

复位后: 0006 0000H R/W



位的位置	位名称	功能										
31-16	PPCM [15:0]	<p>端口电源控制屏蔽</p> <p>表示端口是否通过设置/清除全局电源来控制。</p> <p>只有当 PSM 位被置为“1”时，该位的设置有效。</p> <ul style="list-style-type: none"> • 字段 <table border="1"> <thead> <tr> <th>位</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>保留</td> </tr> <tr> <td>1</td> <td>设备连接到端口1的设置</td> </tr> <tr> <td>2</td> <td>设备连接到端口2的设置</td> </tr> <tr> <td>15-3</td> <td>保留</td> </tr> </tbody> </table> <ul style="list-style-type: none"> • 值 <p>1: 端口只能通过设置/清除全局电源来控制。</p> <p>0: 端口通过设置/清除全局电源来控制。</p>	位	描述	0	保留	1	设备连接到端口1的设置	2	设备连接到端口2的设置	15-3	保留
位	描述											
0	保留											
1	设备连接到端口1的设置											
2	设备连接到端口2的设置											
15-3	保留											
15-0	DR [15:0]	<p>设备可移除性</p> <p>表示连接到 USB 主机控制器端口的设备是否可移除。</p> <ul style="list-style-type: none"> • 字段 <table border="1"> <thead> <tr> <th>位</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>保留</td> </tr> <tr> <td>1</td> <td>设备连接到端口1的设置</td> </tr> <tr> <td>2</td> <td>设备连接到端口2的设置</td> </tr> <tr> <td>15-3</td> <td>保留</td> </tr> </tbody> </table> <ul style="list-style-type: none"> • 值 <p>1: 已连接的设备不可移除。</p> <p>0: 已连接的设备可以移除。</p>	位	描述	0	保留	1	设备连接到端口1的设置	2	设备连接到端口2的设置	15-3	保留
位	描述											
0	保留											
1	设备连接到端口1的设置											
2	设备连接到端口2的设置											
15-3	保留											

(21) HcRhStatus 寄存器 (偏移地址 50H)

(1/2)

复位后: 0000 0000H R/W

(a) 当读取时

31	30	29	28	27	26	25	24
CRWE	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	OCIC	LPSC
15	14	13	12	11	10	9	8
DRWE	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	OCI	LPS

(b) 当写入时

31	30	29	28	27	26	25	24
CRWE	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	OCIC	SGP
15	14	13	12	11	10	9	8
SRWE	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	OCI	CGP

位的位置	R/W	位名称	功能
31	-	CRWE	清除远程唤醒使能 清除 DRWE 位为“0”。 向该位写入“1”，会将 DRWE 位清除为 0。向该位写入“0”被忽略。 读取该位，总是获得“0”。 该位只写。
30-18	-	-	保留 (请确保向这些位写入 “0”)
17	-	OCIC	表示过电流改变 报告 OCI 位的改变情况。当 OCI 位的设置改变时，该位置位 (1)。当该位置位 (1) 时，向该位写入“1”，则该位被清除 (0)。 1: 过电流状态已改变 0: 过电流状态没有改变。
16	R	LPSC	本地电源状态改变 因为不支持本地电源状态，该位固定为“0”。
	W	SGP	设置全局电源 用于在全局电源模式下打开所有端口的电源。该位设置为“1”，将会打开所有端口的电源。 当 HcRhDescriptorA 寄存器的 PSM 位设置为“1”时，那些 PPCM 字段被清除 (0) 的端口，其电源被打开。

位的位置	R/W	位名称	功能
15	R	DRWE	设备远程唤醒使能 设置远程唤醒事件中是否包含连接状态改变。 1: 连接状态改变是一个远程唤醒源。 0: 连接状态改变不是一个远程唤醒源。 当该位置位 (1) 时, 如果发生连接状态改变事件, 状态从 USB 挂起改变为 USB 恢复, 且产生检测到恢复的中断。
	W	SRWE	设置远程唤醒使能。 设置 DRWE 位为 1。 向该位写入“1”, 设置 DRWE 位为“1”。向该位写入“0”的操作将被忽略。
14-2	-	-	保留 (请确保向这些位写入 “0”)
1	-	OCI	过电流指示器 报告全局过电流检测模式下的过电流情况。 1: 端口处于过电流状态。 0: 端口处于正常状态。 当端口单元报告过电流时, 该位固定为“0”。 该位只读。
0	R	LPS	本地电源状态 因为不支持本地电源状态, 该位固定为“0”。
	W	CGP	清除全局电源 用于在全局电源模式下关闭所有端口的电源。 设置该位为“1”时, 关闭所有端口的电源。 当 HcRhDescriptorA 寄存器的 PSM 位被设置为“1”时, 那些 PPCM 字段被清除 (0) 的端口, 其电源被关闭。

(22) HcRhPortStatus1/2 寄存器 (偏移地址 54H/58H)

(1/4)

复位后: 0000 0000H R/W

(a) 当读取时

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	PRSC	OCIC	PSSC	PESC	CSC
15	14	13	12	11	10	9	8
0	0	0	0	0	0	LSDA	PPS
7	6	5	4	3	2	1	0
0	0	0	PRS	POCI	PSS	PES	CSC

(b) 当写入时

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	PRSC	OCIC	PSSC	PESC	CSC
15	14	13	12	11	10	9	8
0	0	0	0	0	0	CPP	SPP
7	6	5	4	3	2	1	0
0	0	0	SPR	CSS	SPS	SPE	CPE

位的位置	R/W	位名称	功能
30-21	-	-	保留 (请确保向这些位写入“0”)
20	-	PRSC	端口复位状态改变 表示端口复位完成。 1: 端口复位已经完成。 0: 端口复位状态没有改变。 当 10ms 的硬件复位完成时, 该位置位 (1)。 通过主机控制器驱动器 (HCD) 写入“1”时, 该位被清除 (0)。
19	-	OCIC	过电流表示改变 在端口检测到过电流状态时, 该位置位 (1)。 1: 过电流状态已经改变。 0: 过电流状态没有改变。 通过主机控制器驱动器 (HCD) 写入“1”时, 该位被清除 (0)。
18	-	PSSC	端口挂起状态改变 表示恢复 (RESUME) 序列已经完成。 1: 恢复 (RESUME) 已经完成 0: 端口挂起状态没有改变。 当所有的恢复 (RESUME) 处理通过硬件完成时, 该位被置位。 通过主机控制器驱动器 (HCD) 写入“1”时, 该位被清除为 0。

位的位置	R/W	位名称	功能
17	-	PESC	<p>端口使能状态改变</p> <p>表示 PES 位已经被清除 (0)。</p> <p>1: PES 位的状态已经被改变 (PES 清除)。</p> <p>0: PES 位的状态没有改变。</p> <p>随着 PES 位被清除 (0)，该位被置位 (1)，因为端口状态已经从使能状态变为禁止状态。原因是硬件事件比如过电流，设备断开，掉电或者泡沫错误等。</p> <p>通过主机控制器驱动器 (HCD) 写入“1”时，该位被清除 (0)。</p>
16	-	CSC	<p>连接状态改变</p> <p>表示 CCS 位已经被改变。</p> <p>1: CCS 位状态已经改变</p> <p>0: CCS 位状态没有改变</p> <p>当 CCS 位状态因为 USB 设备的连接或者断开而改变时，该位被置位 (1)。</p> <p>当 USB 设备断开时，发布接口复位、接口挂起和接口使能的请求，该位被置位 (1)，所以驱动可以重新评估设备的连接情况。</p> <p>通过主机控制器驱动器 (HCD) 写入“1”时，该位被清除 (0)。</p>
15-10	-	-	保留 (请确保向这些位写入“0”)
9	R	LSDA	<p>低速设备连接</p> <p>表示连接到 USB 端口的设备速度。</p> <p>1: 低速设备连接</p> <p>0: 全速设备连接</p> <p>只有当 CCS 位被设置时，该位使能。</p>
	W	CPP	<p>清除端口电源</p> <p>关闭端口电源。</p> <p>向该位写入“1”，会关闭端口电源。向该位写入“0”的操作将被忽略。</p>
8	R	PPS	<p>端口电源状态</p> <p>表示端口电源状态。</p> <p>1: 端口电源打开</p> <p>0: 端口电源关闭</p> <p>根据电源转换时间，控制方法不同。</p>
	W	SPP	<p>设置端口电源</p> <p>在端口单元执行电源控制时，打开端口电源。</p> <p>向该位写入“1”，会打开端口电源。向该位写入“0”的操作将被忽略。</p>
7-5	-	-	保留 (请确保向这些位写入“0”)

位的位置	R/W	位名称	功能
4	R	PRS	<p>端口复位状态</p> <p>表示下行端口被复位</p> <p>1: 端口正在被复位</p> <p>0: 端口没有在被复位</p> <p>当端口 10ms 复位结束时, 随着 PRSC 位被置位, 该位被清除 (0)。</p> <p>当 CSC 位被清除 (无设备连接) 时, 该位不能被置位。</p>
	W	SPR	<p>设置端口复位</p> <p>对下行端口进行端口复位。</p> <p>向该位写入“1”, 将启动 10ms 的端口复位。当 CCS 位被清除 (0) 时向该位写入“1”, 则 CSC 位被置位, 且向主机控制器驱动器报告试图对断开的端口进行复位。向该位写入“0”的操作将被忽略。</p>
3	R	POCI	<p>端口过电流指示器</p> <p>表示下行端口的过电流状态。</p> <p>1: 端口处于过电流状态。</p> <p>0: 端口处于正常状态。</p>
	W	CPS	<p>清除端口挂起</p> <p>结束挂起模式, 启动恢复序列。</p> <p>向该位写入“1”, 将启动恢复序列。向该位写入“0”的操作将被忽略。</p> <p>只有当 PPS 位被置位后, 才能启动恢复。</p>
2	R	PSS	<p>端口挂起状态</p> <p>表示端口处于挂起状态, 或者正在执行恢复序列。</p> <p>1: 端口处于挂起状态</p> <p>0: 端口处于正常传输状态</p> <p>当 CCS 位被清除为 0 时 (无设备连接), 该位不能被置位。</p> <p>写入 SPS 位会导致端口进入挂起状态。当恢复结束、端口复位结束或者状态转换为 USB RESUME 状态时, 该位被清除 (0)。</p>
	W	SPS	<p>设置端口挂起</p> <p>端口状态转换为挂起状态。</p> <p>向该位写入“1”时, 端口进入挂起状态, 向该位写入“0”的操作将被忽略。</p> <p>当 CCS 位已经被清除为 0 时, 如果写入该位, CSC 位将被置位, 并且向主机控制器驱动报告试图对一个断开的端口执行挂起。</p>

位的位置	R/W	位名称	功能
1	R	PES	<p>端口使能状态 表示端口使能或禁止</p> <p>1: 端口使能 0: 端口禁止</p> <p>当 CSC 位被清除为 0 时 (无设备连接), 该位不能被置位。 当端口复位完成时, 端口进入使能状态。 当检测到过电流、连接断开、掉电或者泡沫错误时, 该位自动被硬件清除 (0)。</p>
	W	SPE	<p>设置端口使能 设置 PES 位为“1”。</p> <p>向该位写入“0”的操作将被忽略。</p> <p>通过设置 SPR 位来改变端口状态。V850ES/JG3-U 和 V850ES/JH3-U 的 OHCI 主机控制器不支持通过该位来使能端口, 因为虽然 OHCI 规范支持, 但 USB 规范不支持该功能。</p>
0	R	CCS	<p>当前连接状态 表示下行端口处的当前连接状态。</p> <p>1: 有设备连接 0: 无设备连接</p>
	W	CPE	<p>清除端口使能 清除 CPE 位。</p> <p>向该位写入“1”, 则端口进入禁止状态。向该位写入“0”的操作将被忽略。</p>

21.7.4 来自USB主机控制器的中断

USB主机控制器对来自OHCI主机控制器的中断进行收集和分类，并作为下列三种中断类型之一将它们报告给系统。

表 21-6. 来自USB主机控制器的中断

给系统的中断报告信号	OHCI主机控制器产生的中断报告信号
INTUSBH0	USBH 状态中断 (INTA, SMMI或者PME)
INTUSBH1	USBH PCI 周期错误
INTUSBH2	USBH PME 中断

各个中断的详细情况如下所示。

(1) USB 状态中断 (INTUSBH0)

(a) 中断路由

根据 OHCI 操作寄存器中的 HcControl 寄存器 IR 位的设置，OHCI 主机控制器向系统报告 INTA 或 SMMI 中断的产生。复位后，根据 IR 位的初始设置，对 INTA 进行路由配置来报告中断。

除所有权变更外，INTA 和 SMMI 作为中断源。

表 21-7. 中断INTA和SMMI的路由

寄存器的IR位	中断报告信号
0	INTA (初始值)
1	SMMI

为了使用 INTA 和 SMMI 中断，PCI 主机桥寄存器中的 PCI 中断控制寄存器的 inta_en 和 int_smmi_en 位必须置位 (1)。

(b) 中断源

支持 OpenHCI 规范中定义的中断。

使用 HcInterruptEnable 中断寄存器设置报告给系统的中断源。中断通过 IR 位定义的路由路径进行报告。

下表列出了中断源。

表 21-8. 中断INTA和SMMI的源

中断源	描述										
Scheduling Overrun	表示在一帧中发生USB调度溢出。										
Writeback DoneHead	表示USB主机控制器结束TD时发生回写。										
Start Of Frame	表示在一帧开始时对HccaFmNumber进行更新。										
Resume Detected	表示检测到从USB设备发出的恢复信号。										
Unrecoverable Error	表示检测到和USB不相关的错误 (PCI异常中止)。										
Frame Number Overflow	表示寄存器HcFmNumber的FrameNumber15位发生变化 (从0 → 1 或从1 → 0)										
Root Hub Status Change	表示HcRhStatus/HcRhPortStatus状态的变化。 该中断源按照以下事件进行分类。 <table border="1" data-bbox="462 583 1360 793"> <tbody> <tr> <td>OverCurrent Indicate Change</td> <td>表示过电流状态的发生。</td> </tr> <tr> <td>Connect Status Change</td> <td>表示设备和USB总线连接/断开。</td> </tr> <tr> <td>Port Enable Status Change</td> <td>表示由于USB错误, 端口被禁止。</td> </tr> <tr> <td>Port Suspend Status Change</td> <td>表示恢复序列的完成。</td> </tr> <tr> <td>Port Reset Status Change</td> <td>表示USB复位的完成。</td> </tr> </tbody> </table>	OverCurrent Indicate Change	表示过电流状态的发生。	Connect Status Change	表示设备和USB总线连接/断开。	Port Enable Status Change	表示由于USB错误, 端口被禁止。	Port Suspend Status Change	表示恢复序列的完成。	Port Reset Status Change	表示USB复位的完成。
OverCurrent Indicate Change	表示过电流状态的发生。										
Connect Status Change	表示设备和USB总线连接/断开。										
Port Enable Status Change	表示由于USB错误, 端口被禁止。										
Port Suspend Status Change	表示恢复序列的完成。										
Port Reset Status Change	表示USB复位的完成。										
Ownership Change	表示所有权请求的发生 该中断仅报告给SMMI。										

(2) PCI 周期错误中断 (INTUSBH1)

对 OHCI 或者 PCI 总线桥寄存器区域 (32 位访问) 执行数据访问导致错误时, 产生 PCI 周期错误中断 (INTUSBH1)。该中断只用于调试, 正常操作期间不使用。

(3) PME 中断 (INTUSBH2)

PME 是电源管理的中断信号, 当不提供 PCLK 时, 用于向系统报告 USB 总线状态的变化情况。

下表列出了在可能在 USB 总线上发生的事件和对中断的支持情况。

表 21-9. PME中断源

事件	中断
过电流表示	不发生
连接	发生
断开	发生
恢复 (远程唤醒)	发生

要使用 PME 中断, OHCI 主机配置寄存器中的电源管理控制/状态寄存器的 PME 使能位和 PCI 主机桥寄存器中的 PCI 中断控制寄存器的 int_pme_en 位必须提前设置为“1”。

第二十二章 DMA 功能（DMA 控制器）

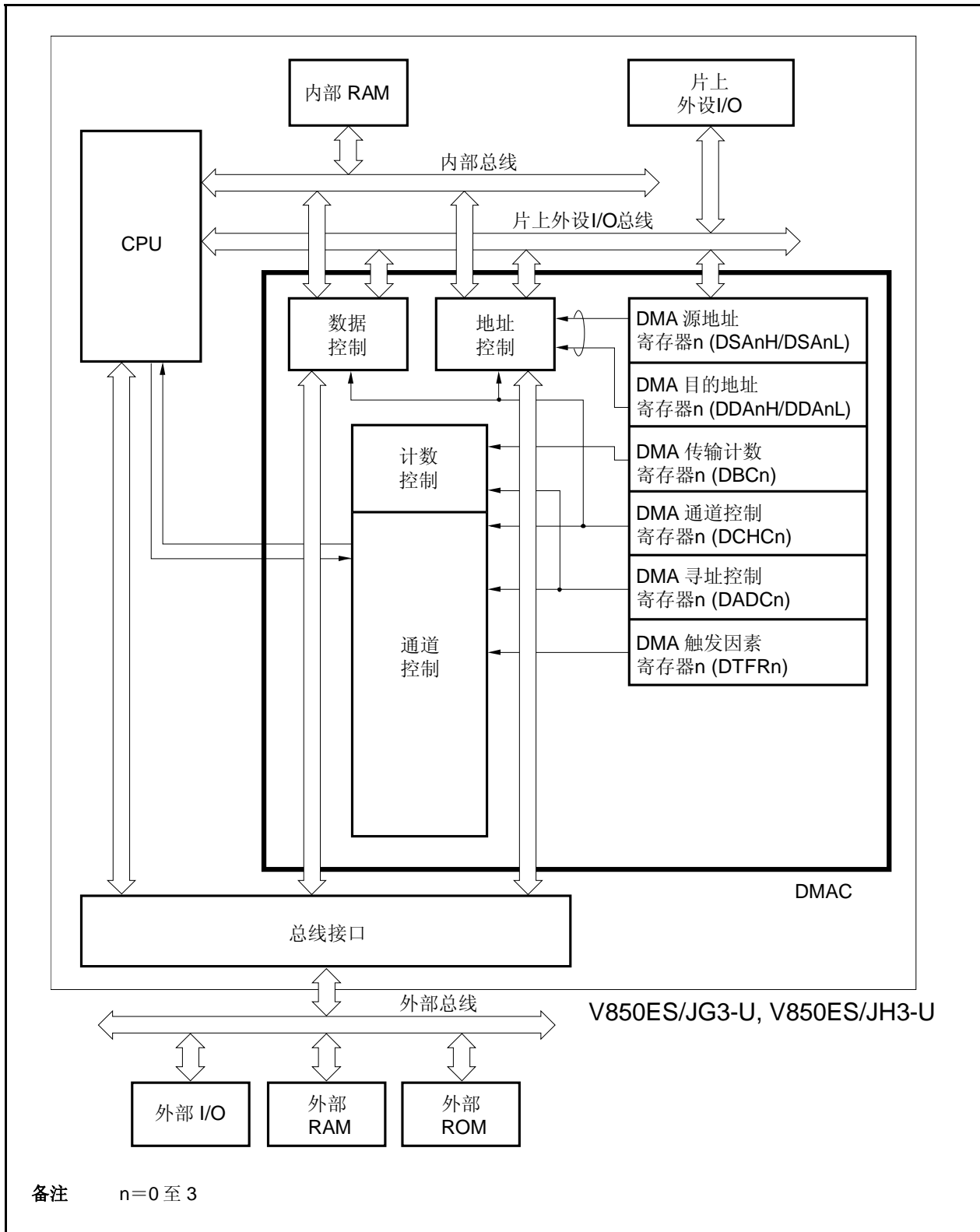
V850ES/JG3-U 和 V850ES/JH3-U 单片机包括一个直接存储器访问（DMA）控制器（DMAC），用于执行和控制 DMA 传输。

根据片上外设 I/O（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发出的 DMA 请求，DMAC 控制存储器与 I/O 之间、存储器之间或 I/O 之间的数据传输。（存储器是指内部 RAM 或外部存储器）

22.1 特性

- 4 路独立 DMA 通道
- 传输单位：8/16 位
- 最大传输计数：65,536 (2^{16})
- 传输类型：双周期传输
- 传输模式：单次传输模式
- 传输请求：
 - 来自片上外设 I/O（串行接口、定时器/计数器和 A/D 转换器）或来自外部输入引脚中断的请求。
 - 来自软件触发的请求。
- 传输对象：
 - 内部 RAM ↔ 外围 I/O
 - 外围 I/O ↔ 外围 I/O
 - 内部 RAM ↔ 外部存储器
 - 外部存储器 ↔ 外围 I/O
 - 外部存储器 ↔ 外部存储器

22.2 配置



22.3 寄存器

(1) DMA 源地址寄存器 0 至 3 (DSA0 至 DSA3)

DSA0 至 DSA3 寄存器用于为 DMA 通道 n 设置 DMA 源地址 (每个 26 位) (n=0 至 3)。

这些寄存器被分为两个 16 位寄存器, 即 DSA_nH 和 DSA_nL。

这些寄存器可以按 16 位宽度读取或写入。

复位后: 不确定 R/W 地址: DSA0H FFFFFFF082H, DSA1H FFFFFFF08AH,
 DSA2H FFFFFFF092H, DSA3H FFFFFFF09AH,
 DSA0L FFFFFFF080H, DSA1L FFFFFFF088H,
 DSA2L FFFFFFF090H, DSA3L FFFFFFF098H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSA _n H (n = 0 至 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 至 3)	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0

IR	DMA传输源分类
0	外部存储或片上外设I/O
1	内部 RAM

SA25 至 SA16	设定DMA传输源地址 (A25 至 A16) (缺省值未定义)。 DMA 传输时, 保持下一个DMA传输源地址。 当DMA传输完成后, 保持最初设定的DMA地址。
-------------	--

SA15 至 SA0	设定DMA传输源地址 (A15 至 A0) (缺省值未定义)。 DMA 传输时, 保持下一个DMA传输源地址。 当DMA传输完成后, 保持最初设定的DMA地址。
------------	---

- 注意事项**
1. 请确保将 DSA_nH 寄存器的第 14 位至第 10 位清为“0”。
 2. 当 DMA 传输禁止 (DCHCn.Enn 位=0) 时, 在以下时序设置 DSA_nH 和 DSA_nL 寄存器。
 - 从复位后到第一次 DMA 传输启动期间
 - 从通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间
 - 从 DMA 传输完成后 (DCHCn.TCn 位=1) 到下一次 DMA 传输启动期间
 3. 当 DSA_n 寄存器的值被读取时, 两个 16 位寄存器, 即 DSA_nH 和 DSA_nL 都被读出。如果读取与更新产生冲突, 则可能是对正在被更新的值进行读取 (请参见 22.13 “注意事项”)。
 4. 系统复位后, 在启动 DMA 发送之前, 设置 DSA_nH, DSA_nL, DDA_nH, DDA_nL 和 DBCn 寄存器。如果没有设置这些寄存器, DMA 传输启动后的操作无法保证。

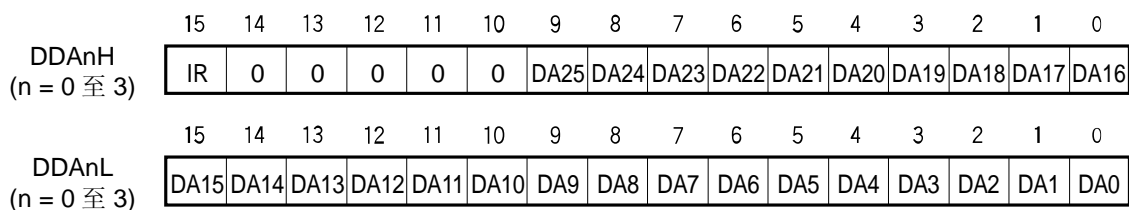
(2) DMA 目的地址寄存器 0 至 3 (DDA0 至 DDA3)

DDA0 至 DDA3 寄存器用于为 DMA 通道 n 设置 DMA 目的地址 (每个 26 位) (n=0 至 3)。

这些寄存器被分为两个 16 位寄存器, 即 DDAnH 和 DDAnL。

这些寄存器可以按 16 位宽度读取或写入。

复位后: 不确定 R/W 地址: DDA0H FFFFF086H, DDA1H FFFFF08EH,
DDA2H FFFFF096H, DDA3H FFFFF09EH,
DDA0L FFFFF084H, DDA1L FFFFF08CH,
DDA2L FFFFF094H, DDA3L FFFFF09CH



IR	DMA传输目的说明
0	外部存储器或片上外设 I/O
1	内部 RAM

DA25 至 DA16	设定一个DMA传输目的地 (A25 至 A16) (缺省值不确定)。 DMA传输期间, 保持下一个DMA传输目的地址。 当DMA传输完成时, 保持最初设定的DMA地址。
-------------	---

DA15 至 DA0	设定一个DMA传输目的地 (A15 至 A0) (缺省值不确定)。 DMA传输期间, 保持下一个DMA传输目的地址。 当DMA传输完成时, 保持最初设定的DMA地址。
------------	--

- 注意事项
1. 请确保将 DSAnH 寄存器的第 14 位至第 10 位清为“0”。
 2. 当 DMA 传输禁止 (DCHCn.Enn 位=0) 时, 在以下时序设置 DDAnH 和 DDAnL 寄存器。
 - 从复位后到第一次 DMA 传输启动期间。
 - 从通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间。
 - 从 DMA 传输完成后 (DCHCn.TCn 位=1) 到下一次 DMA 传输启动期间。
 3. 当 DDAn 寄存器的值被读取时, 两个 16 位寄存器, 即 DDAnH 和 DDAnL 被读出。如果读取与更新产生冲突, 则可能是对正在被更新的值进行读取 (请参见 22.13 “注意事项”)。
 4. 系统复位后, 在开始 DMA 发送之前, 设置 DSAnH, DSAnL, DDAnH, DDAnL 和 DBCn 寄存器。如果没有设置这些寄存器, DMA 传输启动后的操作无法保证。

(3) DMA 传输计数寄存器 0 至 3 (DBC0 至 DBC3)

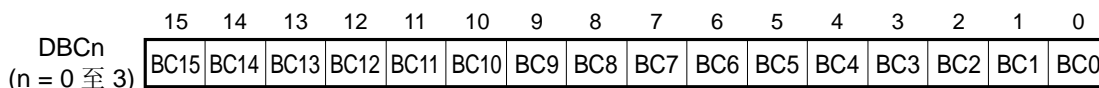
DBC0 至 DBC3 寄存器是 16 位寄存器，用于设置 DMA 通道 n 的传输计数 (n=0 至 3)。

在 DMA 传输过程中，这些寄存器保存剩余的传输计数。

无论传输数据的单位 (8/16 位)，每传输一次，这些寄存器就递减 1，当发生借位时，传输终止。

这些寄存器可以按 16 位宽度读取或写入。

复位后: 不确定 R/W 地址: DBC0 FFFFF0C0H, DBC1 FFFFF0C2H,
DBC2 FFFFF0C4H, DBC3 FFFFF0C6H



BC15至 BC0	传输计数设置 或DMA传输期间保持传输计数
0000H	传输计数1或保持传输计数
0001H	传输计数2或保持传输计数
:	:
FFFFH	传输计数 65,536 (2 ¹⁶), 或保持传输计数
DMA传输完成时, 保存最初设定的传输数据数目。	

注意事项 1. 当 DMA 传输禁止 (DCHCn.Enn 位=0) 时，在下列时序设置 DBCn 寄存器。

- 从复位后到第一次 DMA 传输启动期间
 - 从通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间
 - 从 DMA 传输完成后 (DCHCn.TCn 位=1) 到下一次 DMA 传输启动期间
- 2.** 系统复位后，在开始 DMA 发送之前，设置 DSAnH, DSAnL, DDAnH, DDAnL 和 DBCn 寄存器。如果没有设置这些寄存器，DMA 传输启动之后的操作无法保证。

(4) DMA 寻址控制寄存器 0 至 3 (DADC0 至 DADC3)

DADC0 至 DADC3 寄存器是 16 位寄存器，用于控制 DMA 通道 n 的 DMA 传输模式 (n=0 至 3)。

这些寄存器可以按 16 位宽度读取或写入。

系统复位后，这些寄存器被设置为 0000H。

复位后: 0000H R/W 地址: DADC0 FFFFF0D0H, DADC1 FFFFF0D2H,
DADC2 FFFFF0D4H, DADC3 FFFFF0D6H

DADCn (n = 0 至 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	传输数据宽度的设置
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
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
1	1	禁止设置

- 注意事项**
1. 请确保将 DADCn 寄存器的第 15 位、第 13 位至第 8 位，以及第 3 位至第 0 位清为“0”。
 2. 当 DMA 传输禁止 (DCHCn.Enn 位=0) 时，在下列时序设置 DADCn 寄存器。
 - 从复位后到第一次 DMA 传输启动期间。
 - 从通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间。
 - 从 DMA 传输完成后 (DCHCn.TCn 位=1) 到下一次 DMA 传输启动期间。
 3. DS0 位指定传输数据的宽度，但是无法控制总线的宽度。如果设置了 8 位数据长度 (DS0 位=0)，则不使用低位数据总线。
 4. 如果传输数据宽度被设置为 16 位 (DS0 位=1)，则不能从奇地址开始传输。启动传输的低位地址的第一位总是为 0。
 5. 如果 DMA 传输是在一个片上外设 I/O 寄存器 (作为传输源或目的地) 上执行的，请确保指定的传输宽度与寄存器宽度相同。例如，要在一个 8 位寄存器上执行 DMA 传输，务必指定 8 位宽度传输方式。

(5) DMA 通道控制寄存器 0 至 3 (DCHC0 至 DCHC3)

DCHC0 至 DCHC3 寄存器是 8 位寄存器，用于控制 DMA 通道 n 的 DMA 传输操作模式。

该寄存器可进行字节读写或按位读写。(但是，第 7 位是只读的，而第 1 位和第 2 位是只写的。如果读取第 1 位或第 2 位，则读出的值将始终是 0。)

系统复位后，该寄存器被设为 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至3)	TCn ^{注1}	0	0	0	0	INITn ^{注2}	STGn ^{注2}	Enn

TCn ^{注1}	状态标志指示DMA通道n的DMA传输是否完成
0	DMA传输未完成。
1	DMA传输已完成。
DMA传输的最后设定为1，当读取时清为0。	

INITn ^{注2}	DMA传输禁止(Enn位= 0)时，如果INITn位设定为1，可以初始化DMA传输状态。 在DMA传输完成(TCn位设定为1)之前，重新设定DMA传输状态(重新设定DDAnH, DDAnL, DSAnH, DSAnL, DBCn, 以及DADCn寄存器)，确保初始化DMA通道。 当对DMA控制器进行初始化时，务必阅读22.13 注意事项中的步骤描述。
---------------------	--

STGn ^{注2}	这是DMA传输的软件启动触发信号。 如果在DMA传输使能状态(TCn 位 = 0, Enn 位 = 1)时，该位被置为1，DMA传输启动。
--------------------	--

Enn	设置是否允许DMA通道n的DMA传输
0	DMA传输禁止
1	DMA传输使能
当Enn位被设置为1时，DMA传输使能。 当DMA传输完成时(产生一个终端计数)，该位自动清为0。 通过将Enn位清为0，中断DMA传输。如要恢复，则重新将Enn位设置为1。 当终止或恢复DMA传输时，严格遵守22.13 注意事项 中的步骤描述。	

- 注
1. TCn 位只读。
 2. INITn 和 STGn 位只写。

- 注意事项
1. 请确保将 DCHCn 寄存器的第 6 位至第 3 位清为“0”。
 2. DMA 传输完成后(当产生末尾计数时)，Enn 位被清零，然后 TCn 位被置 1。如果正在更新 DCHCn 寄存器同时发生读取操作，可能会读出一个值表示“传输未完成且传输被禁止”(TCn 位= 0 且 Enn 位=0)。

(6) DMA 触发因素寄存器 0 至 3 (DTFR0 至 DTFR3)

DTFR0 至 DTFR3 寄存器是 8 位寄存器，控制片上外设 I/O 的中断请求信号产生的 DMA 传输启动触发源。

这些寄存器设置的中断请求信号用作 DMA 传输启动因素。

这些寄存器可以按字节读取或写入。但是，DFn 位可以按位进行读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H R/W 地址: DTFR0 FFFFF810H, DTFR1 FFFFF812H,
DTFR2 FFFFF814H, DTFR3 FFFFF816H

	<7>	6	5	4	3	2	1	0
DTFRn	DFn	0	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0

(n = 0 至 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 不要使用软件将 DFn 位置 1。当 DMA 传输禁止时，如果发生了一个被指定作为启动 DMA 传输的触发原因，则将 0 写入该位，以清除 DMA 传输请求。

注意事项 1. 当 DMA 传输禁止 (DCHCn.Enn 位=0) 时，在下列时序设置 IFCn5 至 IFCn0 位。

- 从复位后到第一次 DMA 传输启动期间。
 - 从通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间。
 - 从 DMA 传输完成后 (DCHCn.TCn 位=1) 到下一次 DMA 传输启动期间。
- 2.** 在待机模式 (IDEL1, IDLE2, STOP 或 sub-IDLE 模式) 中产生的中断请求信号不能启动 DMA 传输周期 (DFn 位也不会置 1)。
- 3.** 通过 IFCn5 至 IFCn0 位选定 DMA 启动因素，当选定的片上外设 I/O 产生中断时，不论 DMA 发送操作是允许还是禁止，DFn 位都会置 1。如果这种状态下 DMA 是使能的，则 DMA 传输立即启动。

备注 关于 IFCn5 至 IFCn0 位，请参见表 22-1 DMA 启动因素。

表 22-1. DMA 启动因素 (1/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	中断源
0	0	0	0	0	0	中断禁止的 DMA 请求
0	0	0	0	0	1	INTP02
0	0	0	0	1	0	INTP05
0	0	0	0	1	1	INTP09
0	0	0	1	0	0	INTP10
0	0	0	1	0	1	INTP13
0	0	0	1	1	0	INTP16
0	0	0	1	1	1	INTTAB0OV
0	0	1	0	0	0	INTTAB0CC0
0	0	1	0	0	1	INTTAB0CC1
0	0	1	0	1	0	INTTAB0CC2
0	0	1	0	1	1	INTTAB0CC3
0	0	1	1	0	0	INTTAB1OV_BASE ^注
0	0	1	1	0	1	INTTAB1CC0
0	0	1	1	1	0	INTTAB1CC1
0	0	1	1	1	1	INTTAB1CC2
0	1	0	0	0	0	INTTAB1CC3
0	1	0	0	0	1	INTTT0OV
0	1	0	0	1	0	INTTT0CC0
0	1	0	0	1	1	INTTT0CC1
0	1	0	1	0	0	INTTAA0OV
0	1	0	1	0	1	INTTAA0CC0
0	1	0	1	1	0	INTTAA0CC1
0	1	0	1	1	1	INTTAA1OV
0	1	1	0	0	0	INTTAA1CC0
0	1	1	0	0	1	INTTAA1CC1
0	1	1	0	1	0	INTTAA2CC0
0	1	1	0	1	1	INTTAA2CC1
0	1	1	1	0	0	INTTAA3CC0
0	1	1	1	0	1	INTTAA3CC1
0	1	1	1	1	0	INTTAA4CC0
0	1	1	1	1	1	INTTAA4CC1
1	0	0	0	0	0	INTTAA5CC0
1	0	0	0	0	1	INTTAA5CC1
1	0	0	0	1	0	INTTM0EQ0
1	0	0	0	1	1	INTTM1EQ0
1	0	0	1	0	0	INTTM2EQ0
1	0	0	1	0	1	INTTM3EQ0
1	0	0	1	1	0	INTCF0R/INTIIC1
1	0	0	1	1	1	INTCF0T
1	0	1	0	0	0	INTCF1R
1	0	1	0	0	1	INTCF1T

注 INTTAB1OV_BASE 是 TAB1 的溢出中断被 TMQOP 分选前的中断信号。

表 22-1. DMA 启动因素 (2/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	中断源
1	0	1	0	1	0	INTCF2R
1	0	1	0	1	1	INTCF2T
1	0	1	1	0	0	INTCF3R
1	0	1	1	0	1	INTCF3T
1	0	1	1	1	0	INTCF4R
1	0	1	1	1	1	INTCF4T
1	1	0	0	0	0	INTUC0R
1	1	0	0	0	1	INTUC0T
1	1	0	0	1	0	INTUC1R/INTIIC2
1	1	0	0	1	1	INTUC1T
1	1	0	1	0	0	INTUC2R
1	1	0	1	0	1	INTUC2T
1	1	0	1	1	0	INTUC3R/INTIIC0
1	1	0	1	1	1	INTUC3T
1	1	1	0	0	0	INTUC4R
1	1	1	0	0	1	INTUC4T
1	1	1	0	1	0	INTAD
1	1	1	0	1	1	INTKR
1	1	1	1	0	0	INTRTC1

备注 n=0 至 3

(7) 外部 DMA 请求使能寄存器(EXDRQEN)

当使用 $\overline{\text{UDMARQm}}/\overline{\text{UDMAAKm}}$ 引脚连接外部 USB 设备时, EXDRQEN 寄存器向各个 DMA 通道设置 DMA 请求 ($m = 0, 1$)。

这些寄存器可以按字节读取或写入。

系统复位后, 该寄存器被设为 00H。

复位后: 00H R/W 地址: FFFFFFF60H

	7	6	5	4	3	2	1	0
EXDRQEN	RQ3EX1E	RQ2EX1E	RQ1EX1E	RQ0EX1E	RQ3EX0E	RQ2EX0E	RQ1EX0E	RQ0EX0E

RQnEX1E	DMA 通道n 的指定 (n = 0 至 3)
0	不为 $\overline{\text{UDMARQ1}}/\overline{\text{UDMAAK1}}$ 引脚指定DMA通道 n
1	为 $\overline{\text{UDMARQ1}}/\overline{\text{UDMAAK1}}$ 引脚指定DMA通道 n

RQnEX0E	DMA 通道n 的指定 (n = 0 至 3)
0	不为 $\overline{\text{UDMARQ0}}/\overline{\text{UDMAAK0}}$ 引脚指定DMA通道 n
1	为 $\overline{\text{UDMARQ0}}/\overline{\text{UDMAAK0}}$ 引脚指定DMA通道 n

- 注意事项
- 禁止向 $\overline{\text{UDMARQ1}}/\overline{\text{UDMAAK1}}$ 引脚指定多个 DMA 通道。
(禁止为 $\overline{\text{UDMARQ1}}/\overline{\text{UDMAAK1}}$ 引脚同时将 RQ3EX1E, RQ2EX1E, RQ1EX1E 和 RQ0EX1E 位置位)。
 - 禁止向 $\overline{\text{UDMARQ0}}/\overline{\text{UDMAAK0}}$ 引脚指定多个 DMA 通道。
(禁止为 $\overline{\text{UDMARQ0}}/\overline{\text{UDMAAK0}}$ 引脚同时将 RQ3EX0E, RQ2EX0E, RQ1EX0E 和 RQ0EX0E 位置位)。
 - 禁止为 $\overline{\text{UDMARQ1}}/\overline{\text{UDMAAK1}}$ 引脚和 $\overline{\text{UDMARQ0}}/\overline{\text{UDMAAK0}}$ 引脚指定相同的 DMA 通道 (禁止同时对 RQ3EX1E 和 RQ3EX0E, RQ2EX1E 和 RQ2EX0E, RQ1EX1E 和 RQ1EX0E, 以及 RQ0EX1E 和 RQ0EX0E 位分别置位)。
 - 通过 EXDRQEN 寄存器设置外部源, 当使用来自外部源的 DMA 请求时, 设置 DTFRn.IFCn5-IFCn0 位为 000000 (通过一个中断禁止 DMA 请求)。
详情参见 22.3 (6) DMA 触发因素寄存器 0 至 3 (DTFR0 至 DTFR3)。

22.4 传输对象

表 22-2 说明了传输对象之间的关系 (√: 允许传输, ×: 禁止传输)。

表 22-2 传输对象间的关系

		传输目的地			
		内部 ROM	片上外设 I/O	内部 RAM	外部存储器
传输源	片上外设 I/O	×	√	√	√
	内部 RAM	×	√	×	√
	外部存储器	×	√	√	√
	内部 ROM	×	×	×	×

注意事项 在表 22-2 中, 用“×”标记的传输目的地和传输源组合, 其操作将无法保证。

22.5 传输模式

支持单次传输模式。

在单次传输模式下, 在每次字节/半字传输时总线都被释放。如果有后续的 DMA 传输请求, 则再执行一次传输。该操作一直进行下去, 直到发生一个终止计数。

当 DMAC 释放总线后, 如果发生另一个更高优先级的 DMA 传输请求, 则较高优先级的 DMA 请求将被优先处理。

在一个传输周期中, 如果产生了同一通道的新的传输请求和另一个低优先级通道的传输请求, 则当总线释放给 CPU 后, 执行具有低优先级的通道的 DMA 传输。(同一通道的新的传输请求在传输周期中被忽略)

22.6 传输类型

作为传输模式，可以支持双周期传输。

在双周期传输中，数据传输在两个周期中完成，即读周期和写周期。

在读周期中，输出传输源地址，并执行从源地址向 DMAC 的读操作。在写周期中，输出传输目的地址，并执行从 DMAC 向目的地的写操作。

在读周期和写周期之间总是插入一个时钟的空闲周期。在进行双周期的 DMA 传输时，如果传输源与目的地的数据总线宽度不同，则执行以下操作。

<16 位数据传输>

<1> 从 32 位总线 → 16 位总线传输

先产生一个读周期（高 16 位处于高阻状态），接着产生一个写周期（16 位）。

<2> 从 16/32 位总线向 8 位总线传输

产生一次 16 位读周期，然后产生两次 8 位写周期。

<3> 从 8 位总线向 16/32 位总线传输

先产生两次 8 位读周期，然后产生一次 16 位写周期。

<4> 16 位总线与 32 位总线之间传输

产生一次 16 位读周期，然后产生一次 16 位写周期。

对于在片上外设 I/O 寄存器（传输源或目的地）上执行的 DMA 传输，请确保指定的传输宽度与寄存器宽度相同。例如，要在一个 8 位寄存器上执行 DMA 传输，务必指定 8 位宽度传输方式。

备注 每个传输目标（传输源/目的地）的总线宽度如下：

- 片上外设 I/O: 16 位总线宽度
- 内部 RAM: 32 位总线宽度
- 外部存储器: 8 位或 16 位总线宽度

22.7 DMA通道优先级

DMA 通道优先级固定如下所示:

DMA 通道 0 > DMA 通道 1 > DMA 通道 2 > DMA 通道 3

每个传输周期都要检查这些优先级。

22.8 与DMA传输有关的时间

响应 DMA 请求所需的时间、DMA 传输需要的最少时钟数如下所示:

单次传输: DMA 响应时间 (<1>) + 传输源存储器访问 (<2>) + 1^{#1} + 传输目的地存储器访问 (<2>)

DMA 周期		最少执行时钟数
<1> DMA 请求响应时间		4 时钟 (最小值) + 噪声消除时间 ^{#2}
<2> 存储器访问	外部存储器访问	取决于连接的存储器
	内部 RAM 访问	2 时钟 ^{#3}
	外设 I/O 寄存器访问	3 时钟 + VSWC 寄存器设定的等待周期数 ^{#4}

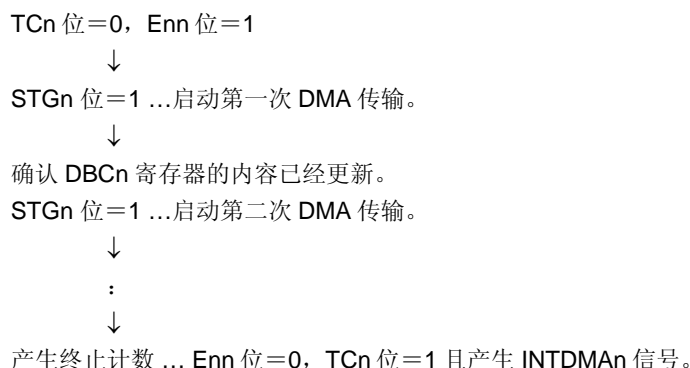
- 注
1. DMA 传输时, 在读周期和写周期之间总是插入一个时钟。
 2. 如果指定一个外部中断 (INTPn) 作为启动 DMA 传输的触发信号, 则增加噪声消除时间 (n=00 至 18)。
 3. 一个 DMA 周期需要两个时钟。
 4. 访问某个特定的外设 I/O 寄存器, 需要更多等待周期 (详细情况, 请参见 3.4.9 (2))。

22.9 DMA 传输启动因素

DMA 传输启动因素有两种，介绍如下：

(1) 软件请求

当 $DCHCn.TCn$ 位=1 且 Enn 位= 1（允许 DMA 传输）时，如果将 $STGn$ 位设置为 1，则启动 DMA 传输。若在此之后立刻请求下一个 DMA 传输周期，则应使用 $DBCn$ 寄存器，确认前一个 DMA 传输周期已经完成，并再次将 $STGn$ 位设置为 1（ $n=0$ 至 3）。



(2) 片上外设 I/O 请求

当 $DCHCn.TCn$ 位=0 且 Enn 位=1（允许 DMA 传输）时，如果从 $DTFRn$ 寄存器设置的片上外设 I/O 产生一个中断请求，则 DMA 传输启动。

- 注意事项**
1. 两个启动因素（软件触发和硬件触发）不能用于一个 DMA 通道。如果对一个 DMA 通道同时产生两个启动因素，则只有其中一个有效。且不能确认有效启动因素。
 2. 在前一个 DMA 传输请求产生后或在前一个 DMA 传输周期中，产生的新传输请求将被忽略（清除）。
 3. 同一个 DMA 通道的传输请求时间间隔取决于 DMA 传输周期总线等待的设置、其他通道的开始状态或外部总线保持请求也会影响该间隔。特别如注意事项 2 所述，在 DMA 传输周期前或在 DMA 传输周期中，同一个通道产生的新的传输请求将被忽略。因此，同一个 DMA 通道的传输请求时间间隔必须足够长。当使用软件触发时，可以通过更新 $DBCn$ 寄存器来检查之前的 DMA 传输周期完成情况。

22.10 DMA 中止因素

如果发生总线保持, 则 DMA 传输将中止。

如果在内部存储器/片上外设 I/O 和内部存储器//片上外设 I/O 之间传输, 同样会发生 DMA 传输中止。

当总线保持被清除, 则 DMA 传输将恢复。

22.11 DMA 传输结束

当 DMA 传输已经完成 DBCn 寄存器设置的次数, 当 DCHCn.Enn 位被清零、TCn 位被置 1 后, 中断控制器 (INTC) 将会发现一个 DMA 传输结束中断请求信号 (INTDMA_n) (n=0 至 3)。

V850ES/JG3-U 和 V850ES/JH3-U 单片机不向外部器件输出终止计数信号。因此, 通过 DMA 传输结束中断或检测 TCn 位来确认 DMA 传输的完成情况。

22.12 操作时序

图 22-1 至图 22-4 显示了 DMA 的操作时序。

图 22-1. DMA 的优先级 (1)

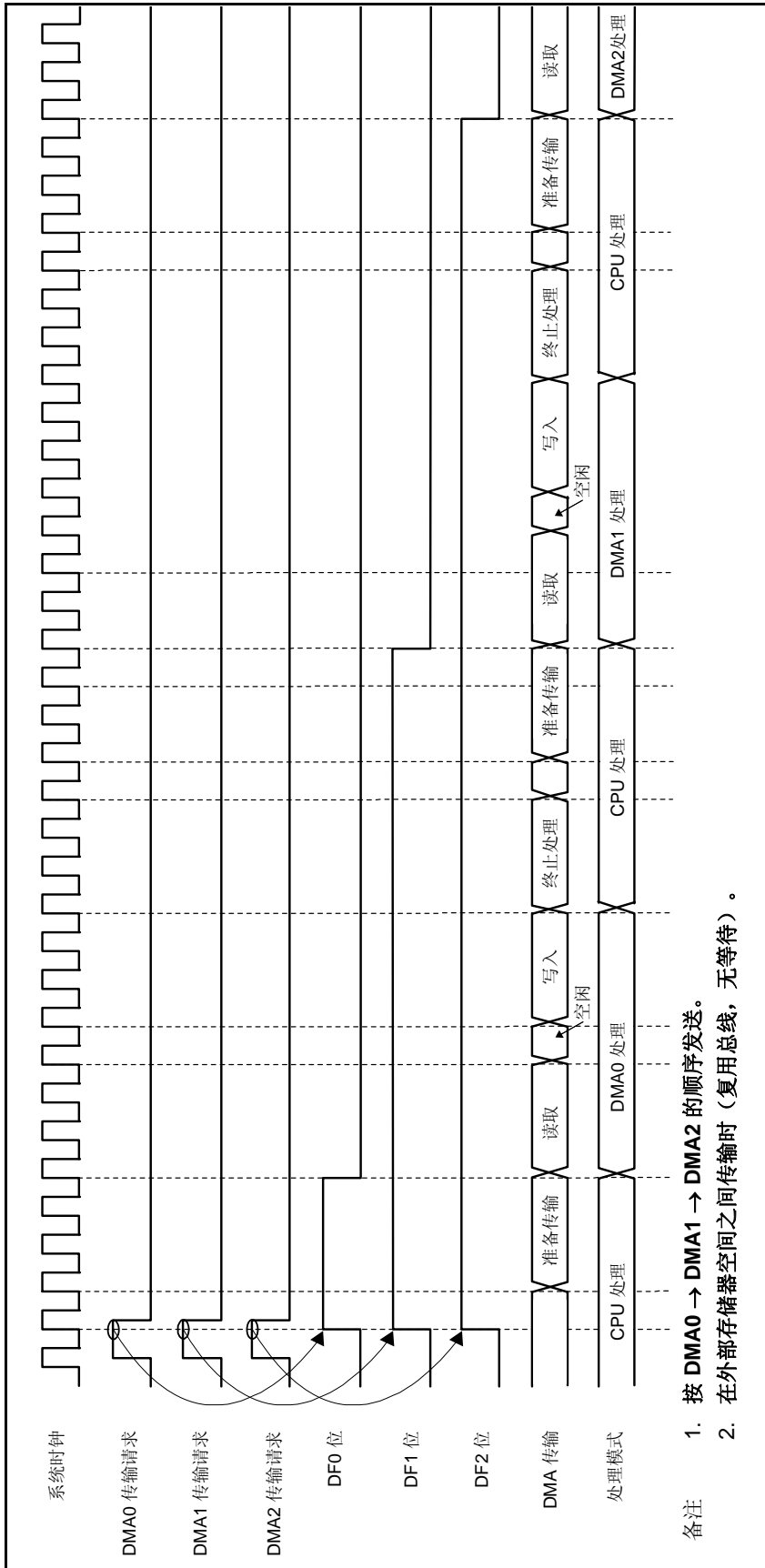


图 22-2. DMA 的优先级 (2)

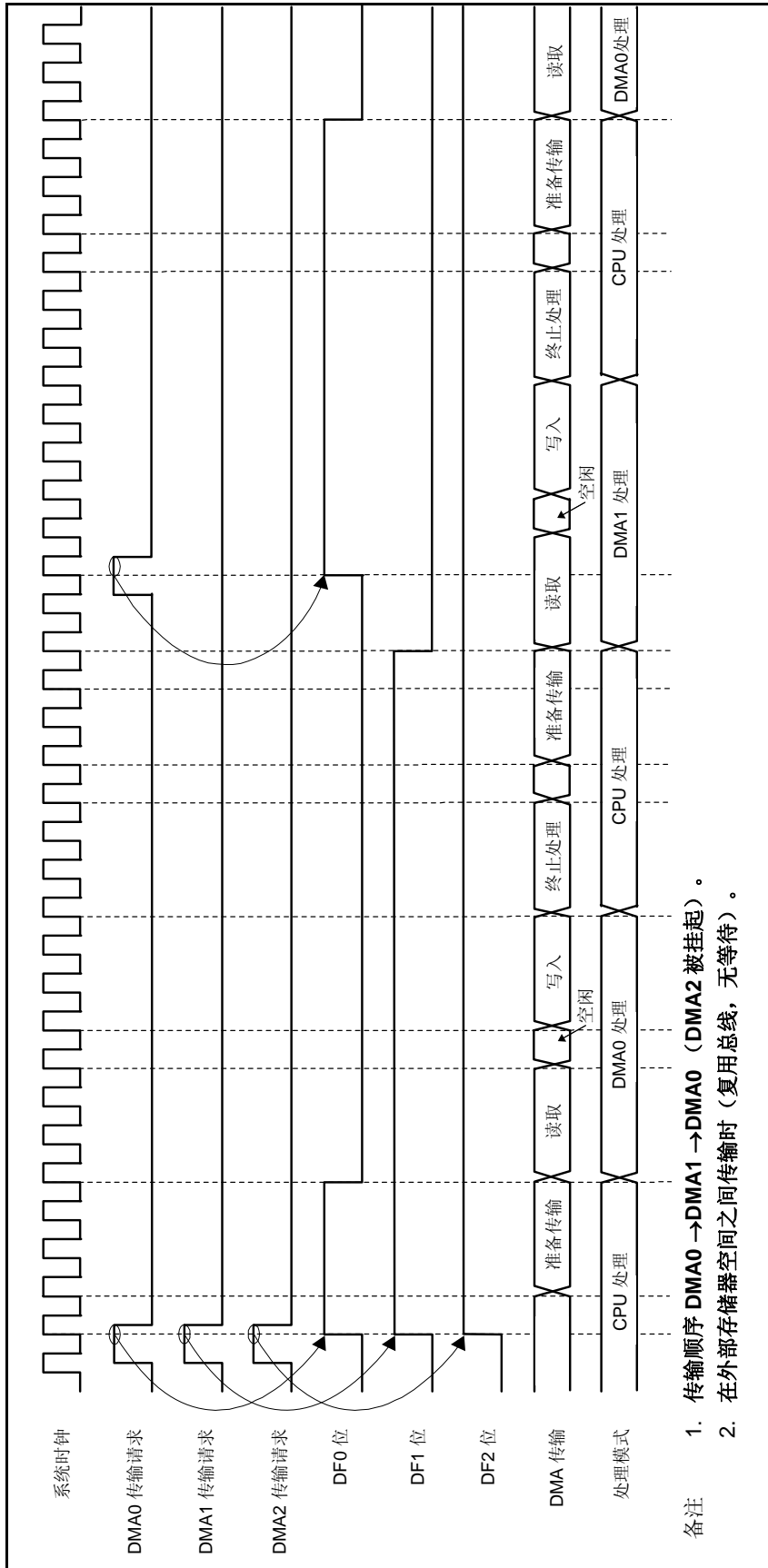


图 22-3 DMA 传输请求被忽略的周期 (1)

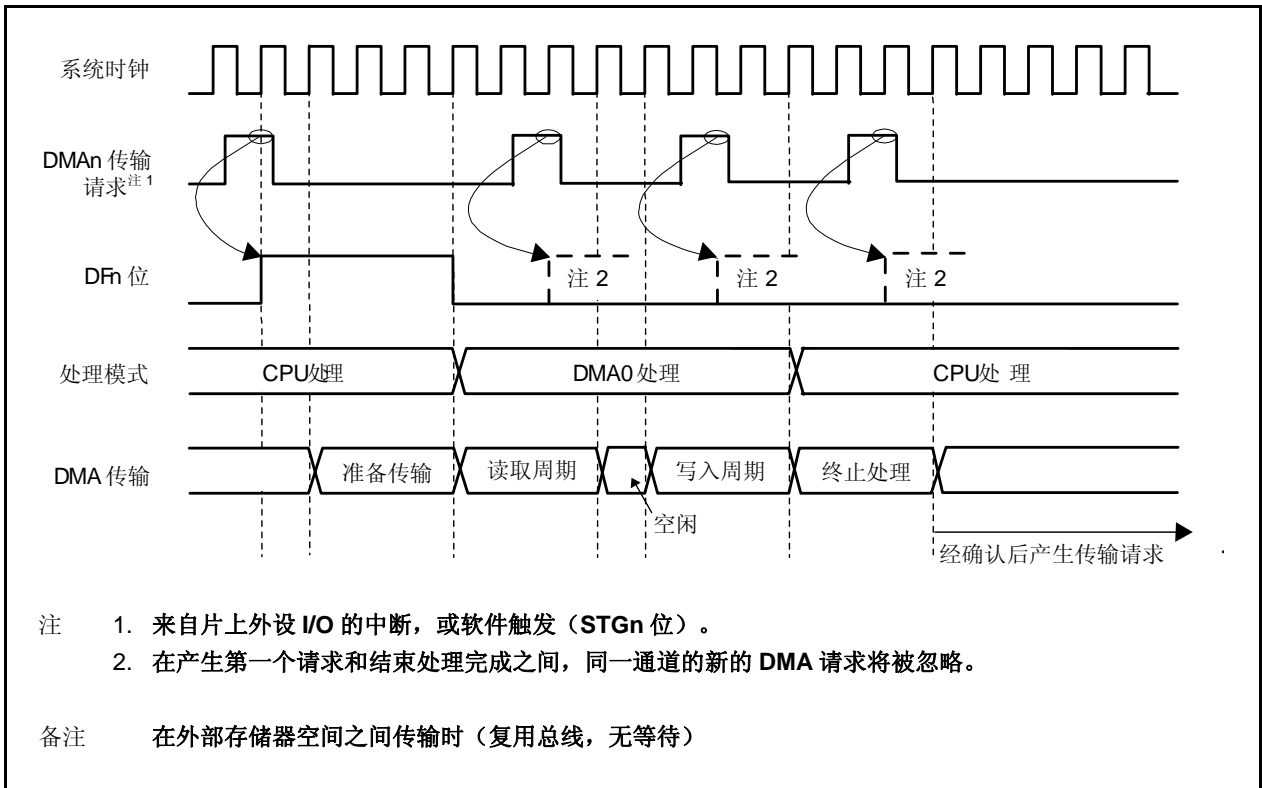
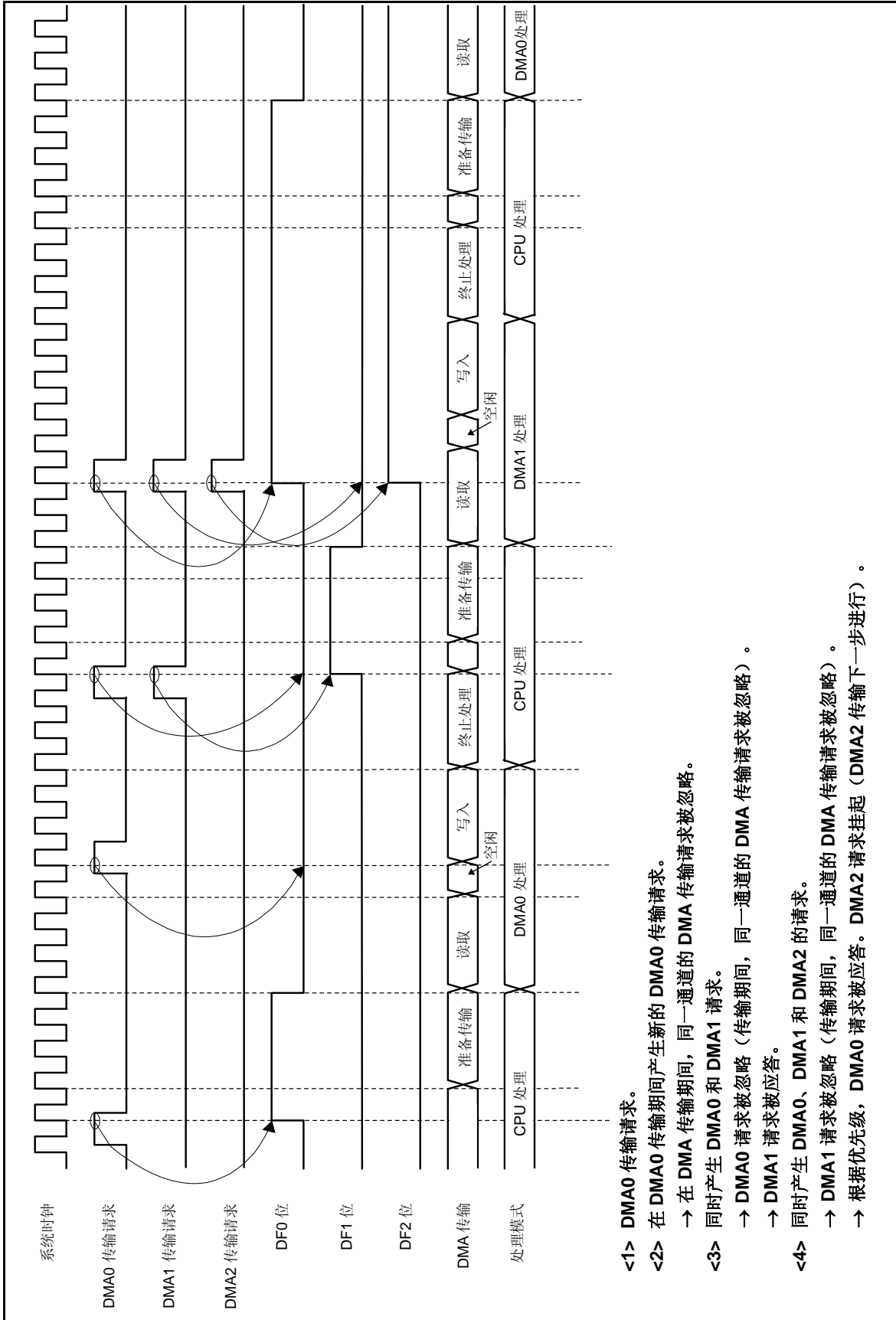


图 22-4 DMA 传输请求被忽略的周期 (2)



- <1> DMA0 传输请求。
- <2> 在 DMA0 传输期间产生新的 DMA0 传输请求。
→ 在 DMA 传输期间，同一通道的 DMA 传输请求被忽略。
- <3> 同时产生 DMA0 和 DMA1 请求。
→ DMA0 请求被忽略（传输期间，同一通道的 DMA 传输请求被忽略）。
→ DMA1 请求被应答。
- <4> 同时产生 DMA0、DMA1 和 DMA2 的请求。
→ DMA1 请求被忽略（传输期间，同一通道的 DMA 传输请求被忽略）。
→ 根据优先级，DMA0 请求被应答。DMA2 请求挂起（DMA2 传输下一步进行）。

22.13 注意事项

(1) VSWC 寄存器注意事项

在使用 DMAC 时，确保根据工作频率为 VSWC 寄存器设置一个合适的值。

如果使用了 VSWC 寄存器的默认值 (77H)，或者给 VSWC 寄存器设置了一个不适当的值，则操作将不能正确执行 (关于寄存器的详细情况，请参见 3.4.8 (1) (a) 系统等待控制寄存器 (VSWC))。

(2) 在内部 RAM 中执行 DMA 传输的注意事项

在内部 RAM 中执行以下指令时，切勿执行与内部 RAM (传输源/目的地) 之间进行数据交互的 DMA 传输，因为 CPU 随后可能会发生错误。

- 位于内部 RAM 中的位操作指令 (SET1, CLR1, 或 NOT1)。
- 访问内部 RAM 中非对齐地址的数据访问指令。

反之，当 DMA 传输的执行与内部 RAM (传输源/目的地) 之间数据交互时，切勿执行以上两条指令。

(3) 读 DCHCn.TCn 位 (n=0 至 3) 的注意事项

当读取 TCn 位时，TCn 位被清零。但是它不会自动清零，即使在一个特定时刻被读取也不会清零。要准确清零 TCn 位，请增加以下处理操作：

(a) 通过轮询 TCn 位等待 DMA 传输完成时

确认 TCn 位已经被设置为 1 (在读取 TCn 位=1 后)，然后再读 TCn 位三次以上。

(b) 在断服务例程中的读取 TCn 位时

执行 TCn 位的读取操作三次。

(4) DMA 传输初始化方法 (将 DCHCn.INITn 位设置为 1)

当执行 DMA 传输的通道要进行初始化时, 即使 INITn 位被置为 1, 通道可能也无法初始化。要准确地初始化通道, 执行以下两种方法之一。

<R> (a) 暂时停止所有 DMA 通道的传输

使用以下<1>至<7>步, 对执行 DMA 传输的通道进行初始化。

但是, 请注意, 当执行第<5>步时, TCn 位将被清零。确保其他处理程序不需要 TCn 位为 1。

<1> 禁止中断 (DI)。

<2> 除了被强制终止的 DMA 通道之外, 读取所有通道的 DCHCn.Enn 位, 然后将值传输到通用寄存器。

<3> 清除使用的 DMA 通道 (包括被强制终止的通道) 的 Enn 位为 0。执行两次清除指令来清除最后一个 DMA 通道的 Enn 位。如果 DMA 传输目标 (传输源/目的地) 涉及到内部 RAM, 则执行三次清除指令。

举例: 如果通道 0, 1, 2 被使用, 则按以下顺序执行指令 (如果传输目标不是内部 RAM)。

- 清除 DCHC0.E00 位为 0
- 清除 DCHC1.E1 位为 0
- 清除 DCHC2.E22 位为 0
- 再次清除 DCHC2.E22 位为 0

<4> 对强制终止的通道, 将其 DCHCn.INITn 位置 1。

<5> 读取所有未被强制终止的通道的 TCn 位。如果 TCn 位和第<2>步读取的 Enn 位均为 1 (逻辑与 (AND) 结果是 1), 将已保存的 Enn 位清零。

<6> 第<5>步的操作完成后, 将 Enn 位的值写入 DCHCn 寄存器。

<7> 使能中断 (EI)。

注意事项 确保执行上述步骤中的第<5>步, 以防止对某通道 Enn 位的非法设置, 该通道的 DMA 传输在第<2>步和第<3>步之间正常完成。

(b) 重复执行 INITn 位设置，直到传输被正确地强制终止

- <1> 如果该通道将要被强制终止（停止片上外设 I/O 的操作），则对来自该通道 DMA 请求源请求不作响应。
- <2> 使用 DTFRn.DFn 位，确认待强制终止的通道 DMA 传输请求未处于挂起状态。如果 DMA 传输请求被挂起，等待，直到挂起请求执行完成。
- <3> 已经确认待强制终止的通道 DMA 传输请求未处于挂起状态，则将 Enn 位清零。
- <4> 再次将待强制终止的通道 Enn 位清零。
如果待强制终止的通道传输目标（传输源/目的地）是内部 RAM，则再次执行该操作。
- <5> 将最初设定给待强制终止的通道初始传输数目拷贝到通用寄存器。
- <6> 将待强制终止的通道 INITn 位置为 1。
- <7> 读取待强制终止的通道 DBCn 寄存器的值，然后将其与在第<5>步中的拷贝值进行比较。如果两个值不匹配，重复操作第<6>和<7>步。

- 备注**
- 1. 在第<7>步中读取 DBCn 寄存器的值时，如果正确完成强制终止，则读取获得初始传输数目。否则，读取到的值是传输的剩余数目。
 - 2. 注意，如果频繁地使用待强制终止的 DMA 通道之外的通道进行 DMA 传输，则方法 (b) 会消耗较长时间。

(5) 暂时停止 DMA 传输的步骤（清除 Enn 位）

使用以下步骤对操作状态下的 DMA 传输进行停止和恢复：

- <1> 对来自 DMA 请求源的传输请求（停止片上外设 I/O 的操作）不作响应。
- <2> 使用 DFn 位（检查 DFn 位是否等于 0），确认 DMA 传输请求未处于挂起状态。
如果请求处于挂起状态，等待，直到挂起 DMA 传输的请求执行完成。
- <3> 如果确认没有 DMA 传输请求处于挂起状态，将 Enn 位清零（该操作停止 DMA 传输）。
- <4> 将 Enn 位设置为 1，以恢复 DMA 传输。
- <5> 对已经停止的 DMA 请求源的操作进行恢复（开始片上外设 I/O 的操作）。

(6) 存储边界

在 DMA 传输过程中，如果传输源或目的地的地址超出了 DMA 对象（外部存储器、内部 RAM 或片上外设 I/O）的区域，则操作将无法保证。

(7) 传输非对齐的数据

不支持采用 16 位总线宽度进行非对齐数据的 DMA 传输。

如果一个奇地址被指定为传输源或目的地，则地址的最低有效位被强制假定为 0。

(8) CPU 的总线仲裁

因为 DMA 控制器拥有比 CPU 更高优先级的总线权限，在 DMA 传输期间，发生的 CPU 访问将保持挂起状态直到 DMA 传输周期完成，总线被释放给 CPU 为止。

但是，CPU 可以访问未执行 DMA 传输的外部存储器，内部外设 I/O 和内部 RAM。

- 在外部存储器与片上外设 I/O 之间执行 DMA 传输时，CPU 可以访问内部 ROM 和内部 RAM。
- 当外部存储器之间正在执行 DMA 传输时，CPU 可以访问内部 ROM 和内部外设 I/O。

(9) DMA 操作时不可重写的寄存器/位

当未进行 DMA 操作时，按以下时序设置下列寄存器。

[寄存器]

- DSAnH, DSAnL, DDAAnH, DDAAnL, DBCn 和 DADCn 寄存器。
- DTFRn.IFCn5 至 DTFRn.IFCn0 位。

[设置时序]

- 从复位后到第一次 DMA 传输启动期间。
- 从通道初始化后到 DMA 传输启动期间。
- 从 DMA 传输完成后 (TCn 位=1) 到下一次 DMA 传输启动期间。

(10) 确保将以下寄存器的指定位设置为 0

- DSAnH 寄存器的第 14 位至第 10 位。
- DDAAnH 寄存器的第 14 位至第 10 位。
- DADCn 寄存器的第 15 位，第 13 位至第 8 位和第 3 位至第 0 位。
- DCHCn 寄存器的第 6 位至第 3 位。

(11) DMA 启动因素

不要使用相同启动因素开启两个或更多 DMA 通道。如果使用相同的启动因素开启了两个或更多的 DMA 通道，那么已经设定的 DMA 通道会再启动，或具有较低优先级的 DMA 通道将在具有较高优先级的 DMA 通道之前被应答，操作无法保证。

(12) 读取 DSA_n 和 DDA_n 寄存器的值

在 DMA 传输过程中, 可能对 DSA_n 和 DDA_n 寄存器中正在更新的值 (n=0 至 3) 进行读取。

例如, 当 DMA 传输源地址 (DSA_n 寄存器) 为 0000FFFFH, 且计数方向是递增 (DADC_n.SAD1 和 DADC_n.SAD0 位=00) 时, 如果先读取 DSA_nH 寄存器, 然后再读取 DSA_nL 寄存器, 则根据是否在读取 DSA_nH 寄存器之后立即执行 DMA 传输, DSA_nL 寄存器的值会有所不同, 描述如下。

(a) 在读取 DSA_n 寄存器时未执行 DMA 传输

<1> 读取 DSA_nH 寄存器的值: DSA_nH=0000H

<2> 读取 DSA_nL 寄存器的值: DSA_nL=FFFFH

(b) 在读取 DSA_n 寄存器时发生 DMA 传输

<1> 读取 DSA_nH 寄存器的值: DSA_nH=0000H

<2> 发生 DMA 传输

<3> DSA_n 寄存器递增: DSA_n=00100000H

<4> 读取 DSA_nL 寄存器的值: DSA_nL=0000H

第二十三章 中断/异常处理功能

V850ES/JG3-U 和 V850ES/JH3-U 为中断服务提供专用的中断控制器（INTC），可以处理总共 87 至 92 个中断请求。

所谓中断，是独立于程序执行而发生的事件，而异常则是与程序执行相关的事件。

V850ES/JG3-U 和 V850ES/JH3-U 能够处理来自于片上外设硬件设备和外部源的中断请求信号。不仅如此，还可以通过 TRAP 指令（软件异常）或产生一个异常事件（比如，取非法操作码）（异常陷阱）来启动异常处理。

23.1 特性

○ 中断

表 23-1. V850ES/JG3-U 和 V850ES/JH3-U 的中断

		内部			外部		
		不可屏蔽中断	可屏蔽中断	总共	不可屏蔽中断	可屏蔽中断	总共
V850ES/JG3-U	μ PD70F3763	1	71	72	1	14	15
	μ PD70F3764	1	71	72	1	14	15
V850ES/JH3-U	μ PD70F3768	1	71	72	1	19	20
	μ PD70F3769	1	71	72	1	19	20

- 8 级可编程优先级（可屏蔽中断）
- 依照优先级控制多重中断
- 可以对每一个可屏蔽中断请求进行屏蔽
- 噪声消除、边沿检测以及为外部中断请求信号指定有效边沿

○ 异常

- 软件异常： 32 个中断源
- 异常陷阱： 2 中断源（非法操作码异常）

V850ES/JG3-U 和 V850ES/JH3-U 的中断/异常发生源分别在表 23-2 和表 23-3 中列出。

表 23-2. V850ES/JG3-U 的中断源 (1/4)

类型	类别	默认优先级	名称	触发	产生单元	异常码	句柄地址	被恢复的 PC	中断控制寄存器
复位	中断	-	RESET	RESET 引脚输入 由内部源复位输入	RESET	0000H	00000000H	不确定	-
不可屏蔽	中断	-	NMI	NMI 引脚有效沿输入	引脚	0010H	00000010H	下一 PC 值	-
		-	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 PC 值	-
		-	TRAP1n ^{注 2}	TRAP 指令	-	005nH ^{注 2}	00000050H	下一 PC 值	-
异常陷阱	异常	-	ILGOP/ DBG0	非法操作码 DBTRAP 指令	-	0060H	00000060H	下一 PC 值	-
可屏蔽	中断	0	INTLVI	低电压检测	POCLVI	0080H	00000080H	下一 PC 值	LVIIC
		3	INTP02	外部中断引脚输入边沿检测 (INTP02)	引脚	00B0H	000000B0H	下一 PC 值	PIC02
		6	INTP05	外部中断引脚输入边沿检测 (INTP05)	引脚	00E0H	000000E0H	下一 PC 值	PIC05
		8	INTP07	外部中断引脚输入边沿检测 (INTP07)	引脚	0100H	00000100H	下一 PC 值	PIC07
		9	INTP08	外部中断引脚输入边沿检测 (INTP08)	引脚	0110H	00000110H	下一 PC 值	PIC08
		10	INTP09	外部中断引脚输入边沿检测 (INTP09)	引脚	0120H	00000120H	下一 PC 值	PIC09
		11	INTP10	外部中断引脚输入边沿检测 (INTP10)	引脚	0130H	00000130H	下一 PC 值	PIC10
		12	INTP11	外部中断引脚输入边沿检测 (INTP11)	引脚	0140H	00000140H	下一 PC 值	PIC11
		13	INTP12	外部中断引脚输入边沿检测 (INTP12)	引脚	0150H	00000150H	下一 PC 值	PIC12
		14	INTP13	外部中断引脚输入边沿检测 (INTP16)	引脚	0160H	00000160H	下一 PC 值	PIC13
		15	INTP14	外部中断引脚输入边沿检测 (INTP14)	引脚	0170H	00000170H	下一 PC 值	PIC14
		16	INTP15	外部中断引脚输入边沿检测 (INTP15)	引脚	0180H	00000180H	下一 PC 值	PIC15
		17	INTP16	外部中断引脚输入边沿检测 (INTP16)	引脚	0190H	00000190H	下一 PC 值	PIC16

注 1. 关于 INTWDT2 的恢复情况, 参见 23.2.2 (2) 来自 INTWDT2 的信号。

2. n = 0 至 FH

表 23-2. V850ES/JG3-U 的中断源 (2/4)

类型	类别	默认优先级	名称	触发	产生单元	异常码	句柄地址	被恢复的 PC	中断控制寄存器
可屏蔽	中断	18	INTP17	外部中断引脚输入边沿检测 (INTP17)	引脚	01A0H	00001A0H	下一 PC 值	PIC17
		19	INTP18	外部中断引脚输入边沿检测 (INTP18)	引脚	01B0H	00001B0H	下一 PC 值	PIC18
		20	INTTAB0OV	TAB0 溢出	TAB0	01C0H	00001C0H	下一 PC 值	TAB0VIC
		21	INTTAB0CC0	TAB0 捕获 0/比较 0 匹配	TAB0	01D0H	00001D0H	下一 PC 值	TAB0CCIC0
		22	INTTAB0CC1	TAB0 捕获 1/比较 1 匹配	TAB0	01E0H	00001E0H	下一 PC 值	TAB0CCIC1
		23	INTTAB0CC2	TAB0 捕获 2/比较 2 匹配	TAB0	01F0H	00001F0H	下一 PC 值	TAB0CCIC2
		24	INTTAB0CC3	TAB0 捕获 3/比较 3 匹配	TAB0	0200H	0000200H	下一 PC 值	TAB0CCIC3
		25	INTTAB1OV ^{※1}	TAB1 溢出	TAB1	0210H	0000210H	下一 PC 值	TAB1VIC
		26	INTTAB1CC0 ^{※2}	TAB1 捕获 0/比较 0 匹配	TAB1	0220H	0000220H	下一 PC 值	TAB1CCIC0
		27	INTTAB1CC1	TAB1 捕获 1/比较 1 匹配	TAB1	0230H	0000230H	下一 PC 值	TAB1CCIC1
		28	INTTAB1CC2	TAB1 捕获 2/比较 2 匹配	TAB1	0240H	0000240H	下一 PC 值	TAB1CCIC2
		29	INTTAB1CC3	TAB1 捕获 3/比较 3 匹配	TAB1	0250H	0000250H	下一 PC 值	TAB1CCIC3
		30	INTTT0OV	TMT0 溢出	TMT0	0260H	0000260H	下一 PC 值	TT0VIC
		31	INTTT0CC0	TMT0 捕获 0/比较 0 匹配	TMT0	0270H	0000270H	下一 PC 值	TT0CCIC0
		32	INTTT0CC1	TMT0 捕获 1/比较 1 匹配	TMT0	0280H	0000280H	下一 PC 值	TT0CCIC1
		33	INTTT0EC	TMT0 编码器输入	TMT0	0290H	0000290H	下一 PC 值	TT0ECIC
		34	INTTAA0OV	TAA0 溢出	TAA0	02A0H	00002A0H	下一 PC 值	TAA0VIC
		35	INTTAA0CC0	TAA0 捕获 0/比较 0 匹配	TAA0	02B0H	00002B0H	下一 PC 值	TAA0CCIC0
		36	INTTAA0CC1	TAA0 捕获 1/比较 1 匹配	TAA0	02C0H	00002C0H	下一 PC 值	TAA0CCIC1
		37	INTTAA1OV	TAA1 溢出	TAA1	02D0H	00002D0H	下一 PC 值	TAA1VIC
38	INTTAA1CC0	TAA1 捕获 0/比较 0 匹配	TAA1	02E0H	00002E0H	下一 PC 值	TAA1CCIC0		
39	INTTAA1CC1	TAA1 捕获 1/比较 1 匹配	TAA1	02F0H	00002F0H	下一 PC 值	TAA1CCIC1		
40	INTTAA2OV	TAA2 溢出	TAA2	0300H	0000300H	下一 PC 值	TAA2VIC		

- 注
1. 在 6 相 PWM 输出模式下使用 TAB1 时, 作为来自 TMQOP 的过零匹配中断 (TAB1TIOD) 请求。
 2. 在 6 相 PWM 输出模式下使用 TAB1 时, 作为来自 TMQOP 的比较匹配中断 (TAB1TICD0)。

表 23-2. V850ES/JG3-U 的中断源 (3/4)

类型	类别	默认优先级	名称	触发	产生单元	异常码	句柄地址	被恢复的 PC	中断控制寄存器
可屏蔽	中断	41	INTTAA2CC0	TAA2 捕获 0/比较 0 匹配	TAA2	0310H	00000310H	下一 PC 值	TAA2CCIC0
		42	INTTAA2CC1	TAA2 捕获 1/比较 1 匹配	TAA2	0320H	00000320H	下一 PC 值	TAA2CCIC1
		43	INTTAA3OV	TAA3 溢出	TAA3	0330H	00000330H	下一 PC 值	TAA3OVIC
		44	INTTAA3CC0	TAA3 捕获 0/比较 0 匹配	TAA3	0340H	00000340H	下一 PC 值	TAA3CCIC0
		45	INTTAA3CC1	TAA3 捕获 1/比较 1 匹配	TAA3	0350H	00000350H	下一 PC 值	TAA3CCIC1
		46	INTTAA4OV	TAA4 溢出	TAA4	0360H	00000360H	下一 PC 值	TAA4OVIC
		47	INTTAA4CC0	TAA4 比较 0 匹配	TAA4	0370H	00000370H	下一 PC 值	TAA4CCIC0
		48	INTTAA4CC1	TAA4 比较 1 匹配	TAA4	0380H	00000380H	下一 PC 值	TAA4CCIC1
		49	INTTAA5OV	TAA5 溢出	TAA5	0390H	00000390H	下一 PC 值	TAA5OVIC
		50	INTTAA5CC0	TAA5 捕获 0/比较 0 匹配	TAA5	03A0H	000003A0H	下一 PC 值	TAA5CCIC0
		51	INTTAA5CC1	TAA5 捕获 1/比较 1 匹配	TAA5	03B0H	000003B0H	下一 PC 值	TAA5CCIC1
		52	INTTM0EQ0	TMM0 比较 匹配	TMM0	03C0H	000003C0H	下一 PC 值	TM0EQIC0
		53	INTTM1EQ0	TMM1 比较 匹配	TMM1	03D0H	000003D0H	下一 PC 值	TM1EQIC0
		54	INTTM2EQ0	TMM2 比较 匹配	TMM2	03E0H	000003E0H	下一 PC 值	TM2EQIC0
		55	INTTM3EQ0	TMM3 比较 匹配	TMM3	03F0H	000003F0H	下一 PC 值	TM3EQIC0
		56	INTCF0R /INTIIC1	CSIF0 接收完成/ CSIF0 接收错误/ IIC1 传输完成	CSIF0/ IIC1	0400H	00000400H	下一 PC 值	CF0RIC/ IICIC1
		57	INTCF0T	CSIF0 连续发送写入使能	CSIF0	0410H	00000410H	下一 PC 值	CF0TIC
		58	INTCF1R	CSIF1 接收完成/ CSIF1 接收错误	CSIF1	0420H	00000420H	下一 PC 值	CF1RIC
		59	INTCF1T	CSIF1 连续发送写入使能	CSIF1	0430H	00000430H	下一 PC 值	CF1TIC
		60	INTCF2R	CSIF2 接收完成/ CSIF2 接收错误	CSIF2	0440H	00000440H	下一 PC 值	CF2RIC
		61	INTCF2T	CSIF2 连续发送写入使能	CSIF2	0450H	00000450H	下一 PC 值	CF2TIC
		62	INTCF3R	CSIF3 接收完成/ CSIF3 接收错误	CSIF3	0460H	00000460H	下一 PC 值	CF3RIC
		63	INTCF3T	CSIF3 连续发送写入使能	CSIF3	0470H	00000470H	下一 PC 值	CF3TIC
		64	INTCF4R	CSIF4 接收完成/ CSIF4 接收错误	CSIF4	0480H	00000480H	下一 PC 值	CF4RIC
		65	INTCF4T	CSIF4 连续发送写入使能	CSIF4	0490H	00000490H	下一 PC 值	CF4TIC

表 23-2. V850ES/JG3-U 的中断源 (4/4)

类型	类别	默认优先级	名称	触发	产生单元	异常码	句柄地址	被恢复的 PC	中断控制寄存器
可屏蔽	中断	66	INTUC0R	UARTC0 接收完成/UARTC0 接收错误	UARTC0	04A0H	000004A0H	下一 PC 值	UC0RIC
		67	INTUC0T	UARTC0 连续发送使能	UARTC0	04B0H	000004B0H	下一 PC 值	UC0TIC
		68	INTUC1R/ INTIIC2	UARTC1 接收完成/UARTC1 接收错误/IIC2 传输完成	UARTC1/ IIC2	04C0H	000004C0H	下一 PC 值	UC1RIC/ IICIC2
		69	INTUC1T	UARTC1 连续发送使能	UARTC1	04D0H	000004D0H	下一 PC 值	UC1TIC
		70	INTUC2R	UARTC2 接收完成/UARTC2 接收错误	UARTC2	04E0H	000004E0H	下一 PC 值	UC2RIC
		71	INTUC2T	UARTC2 连续发送使能	UARTC2	04F0H	000004F0H	下一 PC 值	UC2TIC
		72	INTUC3R/ INTIIC0	UARTC3 接收完成/UARTC0 接收错误/IIC0 传输完成	UARTC3/ IIC0	0500H	00000500H	下一 PC 值	UC3RIC/ IICIC0
		73	INTUC3T	UARTC3 连续发送使能	UARTC3	0510H	00000510H	下一 PC 值	UC3TIC
		74	INTUC4R	UARTC4 接收完成/UARTC4 接收错误	UARTC4	0520H	00000520H	下一 PC 值	UC4RIC
		75	INTUC4T	UARTC4 连续发送使能	UARTC4	0530H	00000530H	下一 PC 值	UC4TIC
		76	INTAD	A/D 转换完成	A/D	0540H	00000540H	下一 PC 值	ADIC
		77	INTDMA0	DMA0 传输完成	DMA	0550H	00000550H	下一 PC 值	DMAIC0
		78	INTDMA1	DMA1 传输完成	DMA	0560H	00000560H	下一 PC 值	DMAIC1
		79	INTDMA2	DMA2 传输完成	DMA	0570H	00000570H	下一 PC 值	DMAIC2
		80	INTDMA3	DMA3 传输完成	DMA	0580H	00000580H	下一 PC 值	DMAIC3
		81	INTKR	按键返回中断	KR	0590H	00000590H	下一 PC 值	KRIC
		82	INTRTC0	RTC 固定周期信号	RTC	05A0H	000005A0H	下一 PC 值	RTC0IC
		83	INTRTC1	RTC 闹钟匹配	RTC	05B0H	000005B0H	下一 PC 值	RTC1IC
		84	INTRTC2	RTC 间隔信号	RTC	05C0H	000005C0H	下一 PC 值	RTC2IC
		89	INTUSBH0	USBH 状态中断	USBH	0610H	00000610H	下一 PC 值	UHIC0
		90	INTUSBH1	USBH PCI 周期错误	USBH	0620H	00000620H	下一 PC 值	UHIC1
91	INTUSBH2	USBH PME 中断	USBH	0630H	00000630H	下一 PC 值	UHIC2		
92	INTUSBF0	USBF 中断	USBF	0640H	00000640H	下一 PC 值	UFIC0		
93	INTUSBF1	USBF 恢复中断	USBF	0650H	00000650H	下一 PC 值	UFIC1		

- 备注**
- 1. 默认优先级:** 当两个或两个以上可屏蔽中断请求同时发生时，系统指定的优先级次序。最高优先级是 0。

不可屏蔽中断的优先级次序是：INTWDT2>NMI。

被恢复的 PC: 当中断服务开始时，程序计数器（PC）的值被存入 EIPC、FEPC 或 DBPC。然而，需要注意的是：在执行以下任意一条指令时，如果程序响应一个不可屏蔽中断或可屏蔽中断，则被恢复的 PC 不是下一 PC 值（下一 PC 值）。（如果在中断执行期间响应一个中断，执行停止，之后在中断服务完成后恢复执行。）

 - 载入指令（SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W）。
 - 除法指令（DIV, DIVH, DIVU, DIVHU）。
 - PREPARE, DISPOSE 指令（仅当在堆栈指针被更新前产生中断）。

下一 PC 值: 在中断/异常处理之后，由此开始进行处理的 PC 值。
 - 2. 当发生非法操作码异常时，非法指令的执行地址计算方法为：（被恢复的 PC-4）。**

表 23-3. V850ES/JH3-U 的中断源 (1/4)

类型	类别	默认优先级	名称	触发	产生单元	异常码	句柄地址	被恢复的 PC	中断控制寄存器
复位	中断	-	RESET	RESET 引脚输入 由内部源复位输入	RESET	0000H	00000000H	Undefined	-
不可屏蔽	中断	-	NMI	NMI 引脚有效沿输入	引脚	0010H	00000010H	下一 PC 值	-
		-	INTWDT2	WDT2 溢出	WDT2	0020H	00000020H	注 1	-
软件异常	异常	-	TRAP0n ^{#2}	TRAP 指令	-	004nH ^{#2}	00000040H	下一 PC 值	-
		-	TRAP1n ^{#2}	TRAP 指令	-	005nH ^{#2}	00000050H	下一 PC 值	-
异常陷阱	异常	-	ILGOP/DBG0	非法操作码 DBTRAP 指令	-	0060H	00000060H	下一 PC 值	-
可屏蔽	中断	0	INTLVI	低电压检测	POCLVI	0080H	00000080H	下一 PC 值	LVIIC
		1	INTP00	外部中断引脚输入边沿检测 (INTP00)	引脚	0090H	00000090H	下一 PC 值	PIC00
		2	INTP01	外部中断引脚输入边沿检测 (INTP01)	引脚	00A0H	000000A0H	下一 PC 值	PIC01
		3	INTP02	外部中断引脚输入边沿检测 (INTP02)	引脚	00B0H	000000B0H	下一 PC 值	PIC02
		4	INTP03	外部中断引脚输入边沿检测 (INTP03)	引脚	00C0H	000000C0H	下一 PC 值	PIC03
		5	INTP04	外部中断引脚输入边沿检测 (INTP04)	引脚	00D0H	000000D0H	下一 PC 值	PIC04
		6	INTP05	外部中断引脚输入边沿检测 (INTP05)	引脚	00E0H	000000E0H	下一 PC 值	PIC05
		7	INTP06	外部中断引脚输入边沿检测 (INTP06)	引脚	00F0H	000000F0H	下一 PC 值	PIC06
		8	INTP07	外部中断引脚输入边沿检测 (INTP07)	引脚	0100H	00000100H	下一 PC 值	PIC07
		9	INTP08	外部中断引脚输入边沿检测 (INTP08)	引脚	0110H	00000110H	下一 PC 值	PIC08
		10	INTP09	外部中断引脚输入边沿检测 (INTP09)	引脚	0120H	00000120H	下一 PC 值	PIC09
		11	INTP10	外部中断引脚输入边沿检测 (INTP10)	引脚	0130H	00000130H	下一 PC 值	PIC10
		12	INTP11	外部中断引脚输入边沿检测 (INTP11)	引脚	0140H	00000140H	下一 PC 值	PIC11
		13	INTP12	外部中断引脚输入边沿检测 (INTP12)	引脚	0150H	00000150H	下一 PC 值	PIC12
		14	INTP13	外部中断引脚输入边沿检测 (INTP13)	引脚	0160H	00000160H	下一 PC 值	PIC13
15	INTP14	外部中断引脚输入边沿检测 (INTP14)	引脚	0170H	00000170H	下一 PC 值	PIC14		

- 注 1. 关于 INTWDT2 的恢复情况, 参见 23.2.2 (2) 来自 INTWDT2 的信号。
2. n = 0 至 FH

表 23-3. V850ES/JH3-U 的中断源 (2/4)

类型	类别	默认优先级	名称	触发	产生单元	异常码	句柄地址	被恢复的 PC	中断控制寄存器
可屏蔽	中断	16	INTP15	外部中断引脚输入边沿检测 (INTP15)	引脚	0180H	00000180H	下一 PC 值	PIC15
		17	INTP16	外部中断引脚输入边沿检测 (INTP16)	引脚	0190H	00000190H	下一 PC 值	PIC16
		18	INTP17	外部中断引脚输入边沿检测 (INTP17)	引脚	01A0H	000001A0H	下一 PC 值	PIC17
		19	INTP18	外部中断引脚输入边沿检测 (INTP18)	引脚	01B0H	000001B0H	下一 PC 值	PIC18
		20	INTTAB0OV	TAB0 溢出	TAB0	01C0H	000001C0H	下一 PC 值	TAB0OVIC
		21	INTTAB0CC0	TAB0 捕获 0/比较 0 匹配	TAB0	01D0H	000001D0H	下一 PC 值	TAB0CCIC0
		22	INTTAB0CC1	TAB0 捕获 1/比较 1 匹配	TAB0	01E0H	000001E0H	下一 PC 值	TAB0CCIC1
		23	INTTAB0CC2	TAB0 捕获 2/比较 2 匹配	TAB0	01F0H	000001F0H	下一 PC 值	TAB0CCIC2
		24	INTTAB0CC3	TAB0 捕获 3/比较 3 匹配	TAB0	0200H	00000200H	下一 PC 值	TAB0CCIC3
		25	INTTAB1OV ^{#1}	TAB1 溢出	TAB1	0210H	00000210H	下一 PC 值	TAB1OVIC
		26	INTTAB1CC0 ^{#2}	TAB1 捕获 0/比较 0 匹配	TAB1	0220H	00000220H	下一 PC 值	TAB1CCIC0
		27	INTTAB1CC1	TAB1 捕获 1/比较 1 匹配	TAB1	0230H	00000230H	下一 PC 值	TAB1CCIC1
		28	INTTAB1CC2	TAB1 捕获 2/比较 2 匹配	TAB1	0240H	00000240H	下一 PC 值	TAB1CCIC2
		29	INTTAB1CC3	TAB1 捕获 3/比较 3 匹配	TAB1	0250H	00000250H	下一 PC 值	TAB1CCIC3
		30	INTTT0OV	TMT0 溢出	TMT0	0260H	00000260H	下一 PC 值	TT0OVIC
		31	INTTT0CC0	TMT0 捕获 0/比较 0 匹配	TMT0	0270H	00000270H	下一 PC 值	TT0CCIC0
		32	INTTT0CC1	TMT0 捕获 1/比较 1 匹配	TMT0	0280H	00000280H	下一 PC 值	TT0CCIC1
		33	INTTT0EC	TMT0 编码器输入	TMT0	0290H	00000290H	下一 PC 值	TT0ECIC
		34	INTTAA0OV	TAA0 溢出	TAA0	02A0H	000002A0H	下一 PC 值	TAA0OVIC
		35	INTTAA0CC0	TAA0 捕获 0/比较 0 匹配	TAA0	02B0H	000002B0H	下一 PC 值	TAA0CCIC0
		36	INTTAA0CC1	TAA0 捕获 1/比较 1 匹配	TAA0	02C0H	000002C0H	下一 PC 值	TAA0CCIC1
		37	INTTAA1OV	TAA1 溢出	TAA1	02D0H	000002D0H	下一 PC 值	TAA1OVIC

- 注
1. 在 6 相 PWM 输出模式下使用 TAB1 时，作为来自 TMQOP 的过零匹配中断 (TAB1TOD) 请求。
 2. 在 6 相 PWM 输出模式下使用 TAB1 时，作为来自 TMQOP 的比较匹配中断 (TAB1TICD0)。

表 23-3. V850ES/JH3-U 的中断源 (3/4)

类型	类别	默认优先级	名称	触发	产生单元	异常码	句柄地址	被恢复的 PC	中断控制寄存器
可屏蔽	中断	38	INTTAA1CC0	TAA1 捕获 0/比较 0 匹配	TAA1	02E0H	000002E0H	下一 PC 值	TAA1CCIC0
		39	INTTAA1CC1	TAA1 捕获 1/比较 1 匹配	TAA1	02F0H	000002F0H	下一 PC 值	TAA1CCIC1
		40	INTTAA2OV	TAA2 溢出	TAA2	0300H	00000300H	下一 PC 值	TAA2OVIC
		41	INTTAA2CC0	TAA2 捕获 0/比较 0 匹配	TAA2	0310H	00000310H	下一 PC 值	TAA2CCIC0
		42	INTTAA2CC1	TAA2 捕获 1/比较 1 匹配	TAA2	0320H	00000320H	下一 PC 值	TAA2CCIC1
		43	INTTAA3OV	TAA3 溢出	TAA3	0330H	00000330H	下一 PC 值	TAA3OVIC
		44	INTTAA3CC0	TAA3 捕获 0/比较 0 匹配	TAA3	0340H	00000340H	下一 PC 值	TAA3CCIC0
		45	INTTAA3CC1	TAA3 捕获 1/比较 1 匹配	TAA3	0350H	00000350H	下一 PC 值	TAA3CCIC1
		46	INTTAA4OV	TAA4 溢出	TAA4	0360H	00000360H	下一 PC 值	TAA4OVIC
		47	INTTAA4CC0	TAA4 比较 0 匹配	TAA4	0370H	00000370H	下一 PC 值	TAA4CCIC0
		48	INTTAA4CC1	TAA4 比较 1 匹配	TAA4	0380H	00000380H	下一 PC 值	TAA4CCIC1
		49	INTTAA5OV	TAA5 溢出	TAA5	0390H	00000390H	下一 PC 值	TAA5OVIC
		50	INTTAA5CC0	TAA5 捕获 0/比较 0 匹配	TAA5	03A0H	000003A0H	下一 PC 值	TAA5CCIC0
		51	INTTAA5CC1	TAA5 捕获 1/比较 1 匹配	TAA5	03B0H	000003B0H	下一 PC 值	TAA5CCIC1
		52	INTTM0EQ0	TMM0 比较匹配	TMM0	03C0H	000003C0H	下一 PC 值	TM0EQIC0
		53	INTTM1EQ0	TMM1 比较匹配	TMM1	03D0H	000003D0H	下一 PC 值	TM1EQIC0
		54	INTTM2EQ0	TMM2 比较匹配	TMM2	03E0H	000003E0H	下一 PC 值	TM2EQIC0
		55	INTTM3EQ0	TMM3 比较匹配	TMM3	03F0H	000003F0H	下一 PC 值	TM3EQIC0
		56	INTCF0R/ INTIIC1	CSIF0 接收完成/ CSIF0 接收错误/ IIC1 传输完成	CSIF0/ IIC1	0400H	00000400H	下一 PC 值	CF0RIC/ IIC1C1
		57	INTCF0T	CSIF0 连续发送写入使能	CSIF0	0410H	00000410H	下一 PC 值	CF0TIC
		58	INTCF1R	CSIF1 接收完成/ CSIF1 接收错误	CSIF1	0420H	00000420H	下一 PC 值	CF1RIC
		59	INTCF1T	CSIF1 连续发送写入使能	CSIF1	0430H	00000430H	下一 PC 值	CF1TIC
60	INTCF2R	CSIF2 接收完成/ CSIF2 接收错误	CSIF2	0440H	00000440H	下一 PC 值	CF2RIC		
61	INTCF2T	CSIF2 连续发送写入使能	CSIF2	0450H	00000450H	下一 PC 值	CF2TIC		
62	INTCF3R	CSIF3 接收完成/ CSIF3 接收错误	CSIF3	0460H	00000460H	下一 PC 值	CF3RIC		

表 23-3. V850ES/JH3-U 的中断源 (4/4)

类型	类别	默认优先级	名称	触发	产生单元	异常码	句柄地址	被恢复的 PC	中断控制寄存器
可屏蔽	中断	63	INTCF3T	CSIF3 连续发送写入使能	CSIF3	0470H	00000470H	下一 PC 值	CF3TIC
		64	INTCF4R	CSIF4 接收完成/ CSIF4 接收错误	CSIF4	0480H	00000480H	下一 PC 值	CF4RIC
		65	INTCF4T	CSIF4 连续发送写入使能	CSIF4	0490H	00000490H	下一 PC 值	CF4TIC
		66	INTUC0R	UARTC0 接收完成/UARTC0 接收错误	UARTC0	04A0H	000004A0H	下一 PC 值	UC0RIC
		67	INTUC0T	UARTC0 连续发送使能	UARTC0	04B0H	000004B0H	下一 PC 值	UC0TIC
		68	INTUC1R /INTIIC2	UARTC1 接收完成/UARTC1 接收错误/IIC2 传输完成	UARTC1/ IIC2	04C0H	000004C0H	下一 PC 值	UC1RIC/ IICIC2
		69	INTUC1T	UARTC1 连续发送使能	UARTC1	04D0H	000004D0H	下一 PC 值	UC1TIC
		70	INTUC2R	UARTC2 接收完成/UARTC2 接收错误	UARTC2	04E0H	000004E0H	下一 PC 值	UC2RIC
		71	INTUC2T	UARTC2 连续发送使能	UARTC2	04F0H	000004F0H	下一 PC 值	UC2TIC
		72	INTUC3R/ INTIIC0	UARTC3 接收完成/UARTC0 接收错误/IIC0 传输完成	UARTC3/ IIC0	0500H	00000500H	下一 PC 值	UC3RIC/ IICIC0
		73	INTUC3T	UARTC3 连续发送使能	UARTC3	0510H	00000510H	下一 PC 值	UC3TIC
		74	INTUC4R	UARTC4 接收完成/UARTC4 接收错误	UARTC4	0520H	00000520H	下一 PC 值	UC4RIC
		75	INTUC4T	UARTC4 连续发送使能	UARTC4	0530H	00000530H	下一 PC 值	UC4TIC
		76	INTAD	A/D 转换完成	A/D	0540H	00000540H	下一 PC 值	ADIC
		77	INTDMA0	DMA0 传输完成	DMA	0550H	00000550H	下一 PC 值	DMAIC0
		78	INTDMA1	DMA1 传输完成	DMA	0560H	00000560H	下一 PC 值	DMAIC1
		79	INTDMA2	DMA2 传输完成	DMA	0570H	00000570H	下一 PC 值	DMAIC2
		80	INTDMA3	DMA3 传输完成	DMA	0580H	00000580H	下一 PC 值	DMAIC3
		81	INTKR	按键返回中断	KR	0590H	00000590H	下一 PC 值	KRIC
		82	INTRTC0	RTC 固定周期信号	RTC	05A0H	000005A0H	下一 PC 值	RTC0IC
		83	INTRTC1	RTC 闹钟匹配	RTC	05B0H	000005B0H	下一 PC 值	RTC1IC
		84	INTRTC2	RTC 间隔信号	RTC	05C0H	000005C0H	下一 PC 值	RTC2IC
		89	INTUSBH0	USBH 状态中断	USBH	0610H	00000610H	下一 PC 值	UHIC0
		90	INTUSBH1	USBH PCI 周期错误	USBH	0620H	00000620H	下一 PC 值	UHIC1
		91	INTUSBH2	USBH PME 中断	USBH	0630H	00000630H	下一 PC 值	UHIC2
		92	INTUSBF0	USBF 中断	USBF	0640H	00000640H	下一 PC 值	UFIC0
93	INTUSBF1	USBF 恢复中断	USBF	0650H	00000650H	下一 PC 值	UFIC1		

- 备注**
- 1. 默认优先级:** 当两个或两个以上可屏蔽中断请求同时发生时，系统指定的优先级次序。最高优先级是 0。

不可屏蔽中断的优先级次序是：INTWDT2>NMI。

被恢复的 PC: 当中断服务开始时，程序计数器（PC）的值被存入 EIPC、FEPC 或 DBPC。然而，需要注意的是：在执行以下任意一条指令时，如果程序响应一个不可屏蔽中断或可屏蔽中断，则被恢复的 PC 不是下一 PC 值（下一 PC 值）。（如果在中断执行期间响应一个中断，执行停止，之后在中断服务完成后恢复执行。）

 - 载入指令（SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W）。
 - 除法指令（DIV, DIVH, DIVU, DIVHU）。
 - PREPARE, DISPOSE 指令（仅当在堆栈指针被更新前产生中断）。

下一 PC 值: 在中断/异常处理之后，由此开始进行处理的 PC 值。
 - 2. 当发生非法操作码异常时，非法指令的执行地址计算方法为：（被恢复的 PC-4）。**

23.2 不可屏蔽中断

即便处于中断禁止（DI）状态，不可屏蔽中断请求信号也可以无条件地得到响应。不可屏蔽中断（NMI）并不遵从优先级控制，而且，比所有其他中断请求信号更优先。

该产品有以下两种不可屏蔽中断请求信号。

- NMI 引脚输入（NMI）。
- 看门狗定时器溢出产生的不可屏蔽中断请求信号（INTWDT2）。

NMI 引脚的有效沿可以选择为以下 4 种类型：“上升沿”、“下降沿”、“双边沿”和“无边沿检测”。

当 WDTM2.WDM21 和 WDTM2.WDM20 位被设置为“01”时，看门狗定时器 2（INTWDT2）的溢出会产生不可屏蔽中断请求信号。

当两个或两个以上不可屏蔽中断请求信号同时发生时，较高优先级的中断得到相应（低优先级的中断请求信号被忽略），如下所示：

INTWDT2 > NMI

在 NMI 正在被处理时，发生新的 NMI 或 INTWDT2 请求信号，则按以下情况处理。

（1）如果在 NMI 处理时，发生新的 NMI 请求信号

不管 PSW.NP 位的值如何，新的 NMI 请求信号都会被挂起。挂起的 NMI 请求信号在当前正在处理的 NMI 请求完毕后得到响应（在 RETI 指令执行完成之后）。

（2）如果在 NMI 处理时，发生 INTWDT2 请求信号

在 NMI 处理期间，如果 NP 位保持置位状态（1），INTWDT2 请求信号被挂起。挂起的 INTWDT2 请求信号在当前正在处理的 NMI 请求完毕后得到响应（在 RETI 指令执行完成之后）。

在 NMI 请求处理期间，如果 NP 位被清除为 0，执行新产生的 INTWDT2 请求信号（NMI 处理服务停止）。

注意事项 关于对不可屏蔽中断请求信号（INTWDT2）执行的不可屏蔽中断服务，参见 23.2.2（2）来自 INTWDT2 信号。

图 23-1. 不可屏蔽中断请求信号的应答操作（1/2）

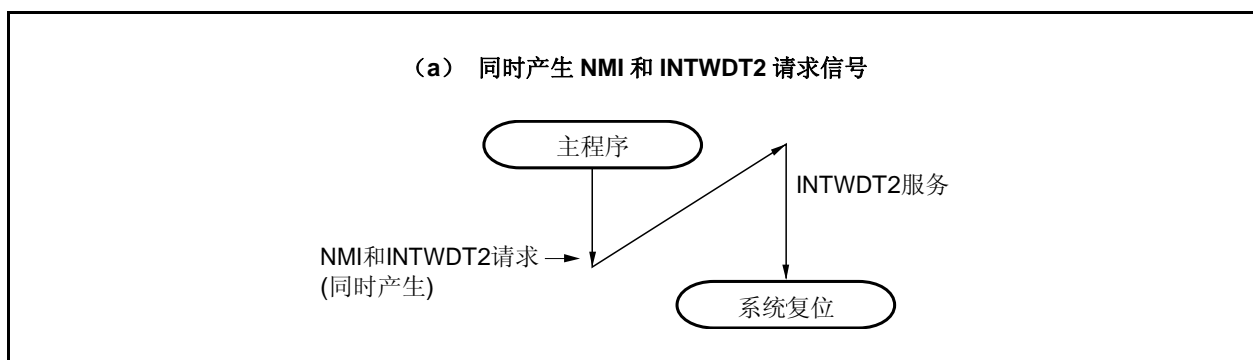
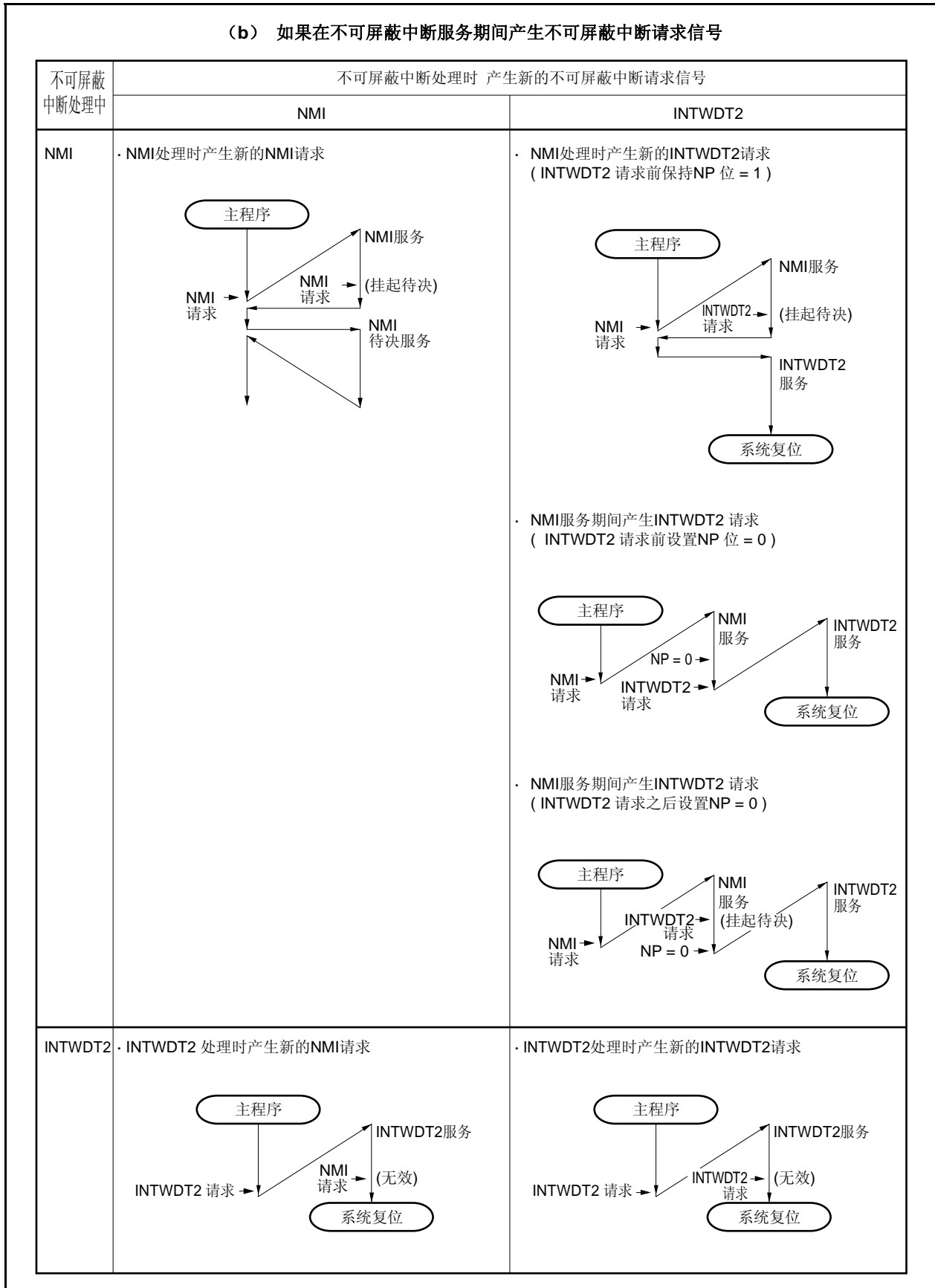


图 23-1. 不可屏蔽中断请求信号的应答操作 (2/2)



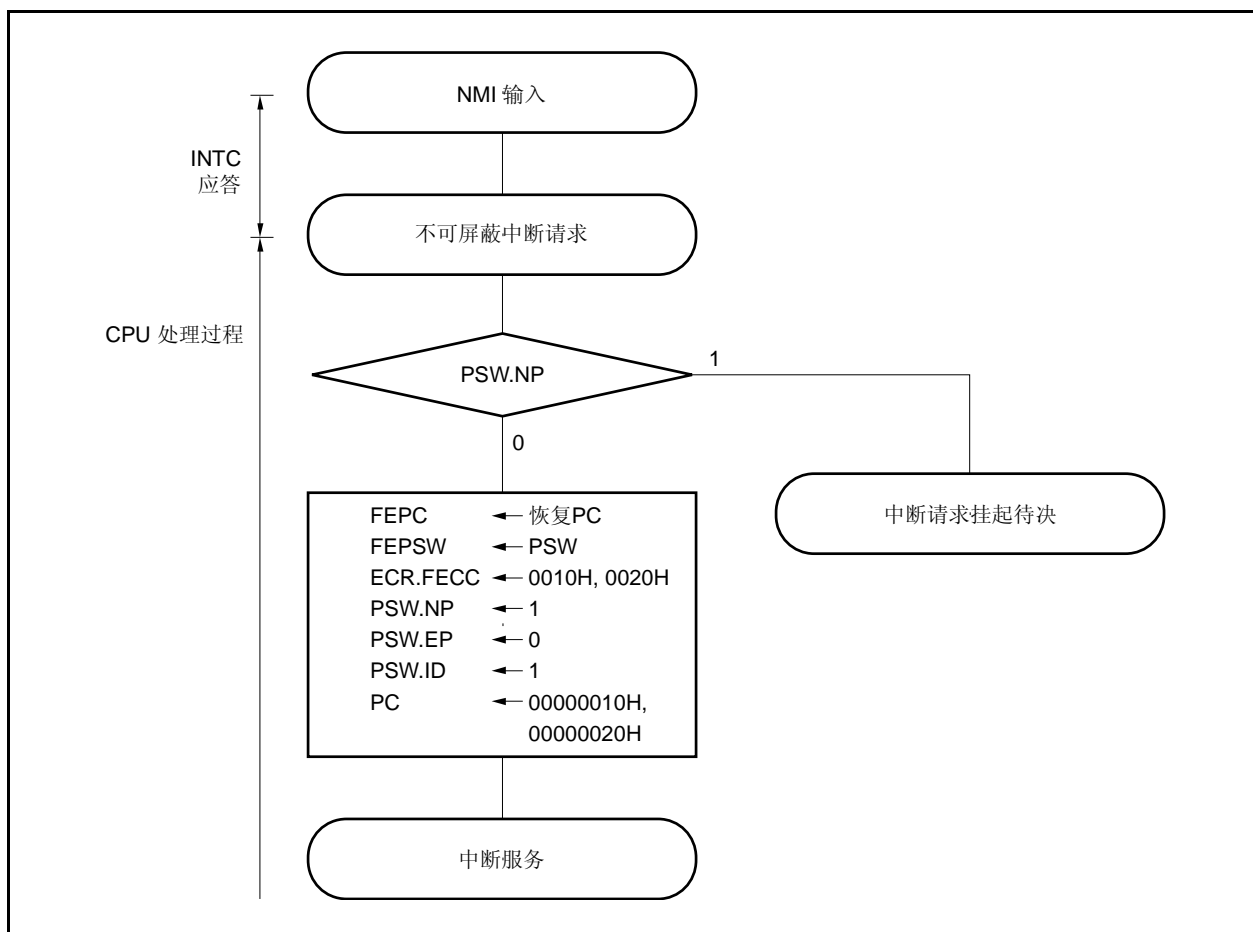
23.2.1 操作

如果产生一个不可屏蔽中断请求信号，则 CPU 执行以下处理过程，并将控制权转移至一个句柄程序。

- <1> 将待恢复的 PC 保存至 FEPC。
- <2> 将当前 PSW 保存至 FEPSW。
- <3> 将异常码（0010H, 0020H）写入 ECR 的高半字（FECC）。
- <4> 将 PSW.NP 和 PSW.ID 位置为 1，并将 PSW.EP 位清为 0。
- <5> 将不可屏蔽中断相应的句柄地址（00000010H, 00000020H）载入 PC，并转移控制。

不可屏蔽中断的服务流程如下所示：

图 23-2. 不可屏蔽中断服务流程



23.2.2 恢复

(1) 来自 NMI 引脚的输入

通过 RETI 指令，从 NMI 服务的恢复处理。

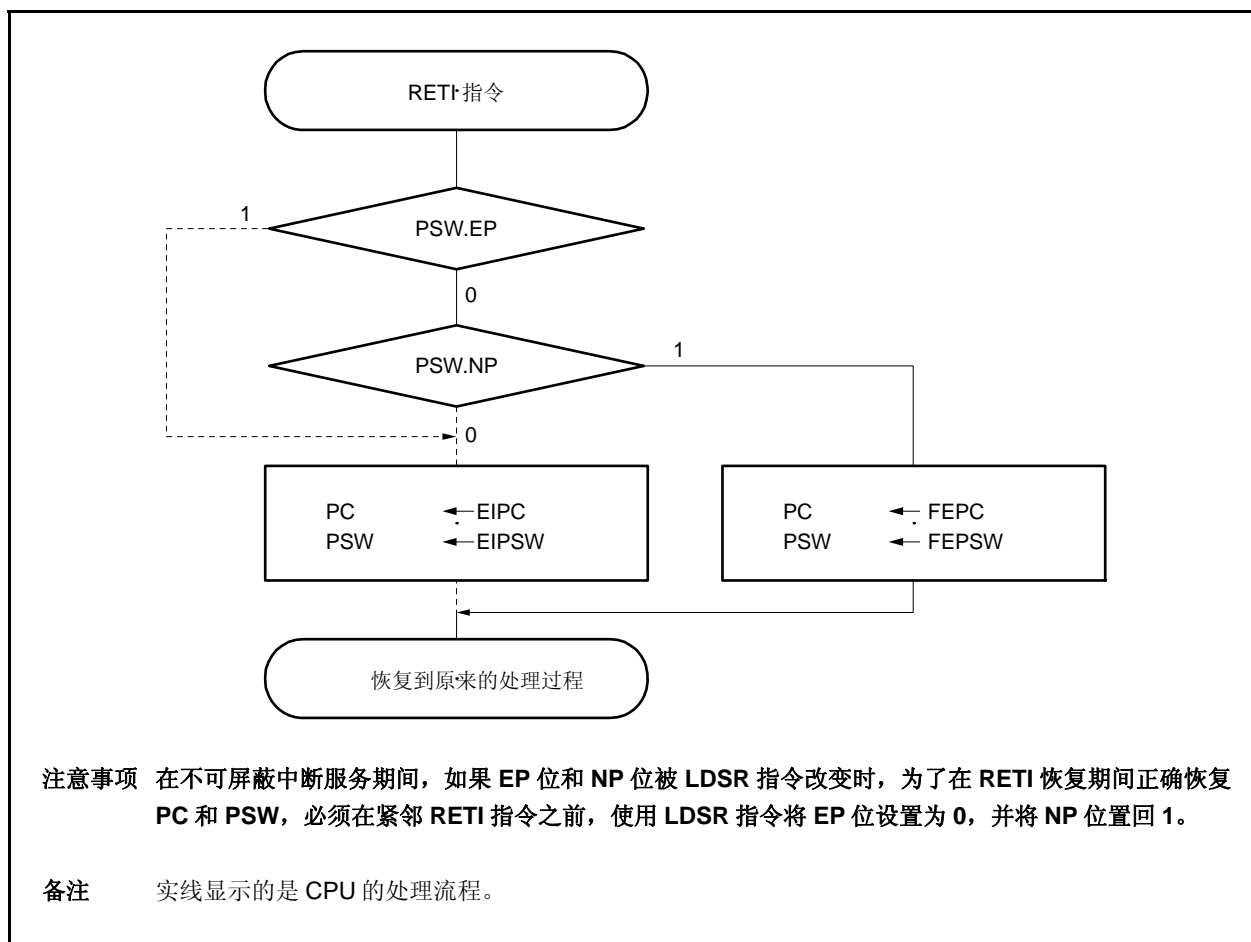
当执行 RETI 指令时，CPU 执行以下处理过程，并且控制转移到被恢复的 PC 地址。

<1> 因为 PSW.EP 位是 0 且 PSW.NP 位是 1，所以，由 FEPC 和 FEPSW 分别载入相应的 PC 和 PSW 的恢复值。

<2> 控制转移返回到被恢复的 PC 和 PSW 的地址。

RETI 指令的处理过程如下所示：

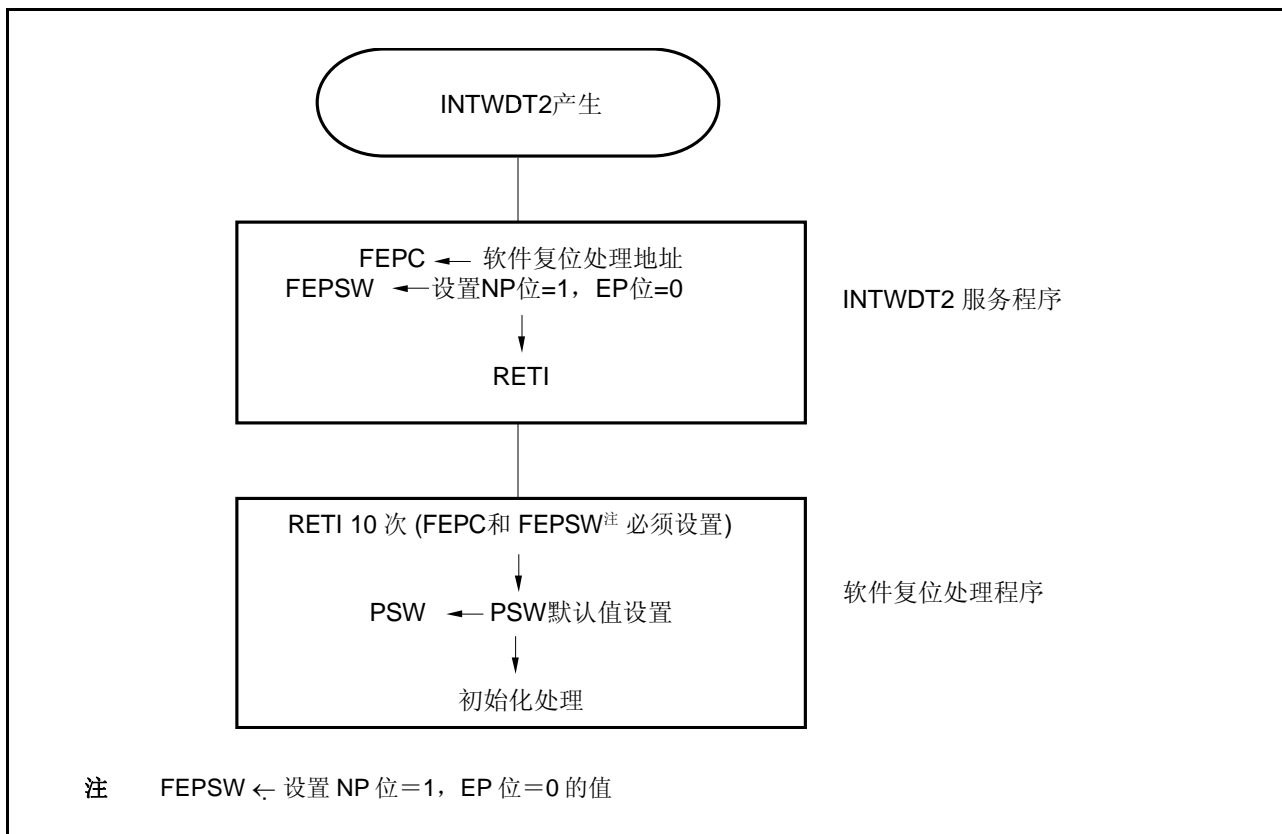
图 23-3. RETI 指令处理过程



(2) 来自 INTWDT2 的信号

对于不可屏蔽中断请求 (INTWDT2) 执行的不可屏蔽中断服务，其恢复时禁止使用 RETI 指令。执行下面的软件复位处理过程。

图 23-4. 软件复位处理过程



23.2.3 NP标志

NP 标志是一个状态标志，用于指示不可屏蔽中断服务正在进行。

当不可屏蔽中断请求信号被响应之后，该标志置位，并且，屏蔽不可屏蔽中断请求，禁止响应多重中断。

复位后: 00000020H

	31		8	7	6	5	4	3	2	1	0
PSW	0			NP	EP	ID	SAT	CY	OV	S	Z

NP	NMI 中断服务状态
0	无 NMI中断服务
1	NMI中断正在处理中

23.3 可屏蔽中断

可屏蔽中断请求信号可以由中断控制寄存器进行屏蔽。V850ES/JG3-U 和 V850ES/JH3-U 单片机有 85 和 90 个可屏蔽中断源。

如果同时发生两个或更多的可屏蔽中断请求信号，根据默认的优先级进行响应。除了默认优先级之外，使用中断控制寄存器（可编程优先级控制）可以指定 8 个优先等级。

当一个中断请求信号已经得到响应，则禁止响应其他可屏蔽中断请求信号，并且中断禁止（DI）状态置位。

当在中断服务程序中执行 EI 指令，中断使能（EI）状态置位，可以使能那些比正在处理的中断请求信号（由中断控制寄存器指定）具有更高的优先级的中断服务。注意，只有具有较高优先级的中断才有这种能力；具有相同优先级的中断不能于嵌套。

但是，为了允许多重中断（中断嵌套），在执行 EI 指令之前有必要将 EIPC 和 EIPSW 保存到存储器或通用寄存器中，并在使用 RETI 指令之前执行 DI 指令，以恢复 EIPC 和 EIPSW 原始值。

23.3.1 操作

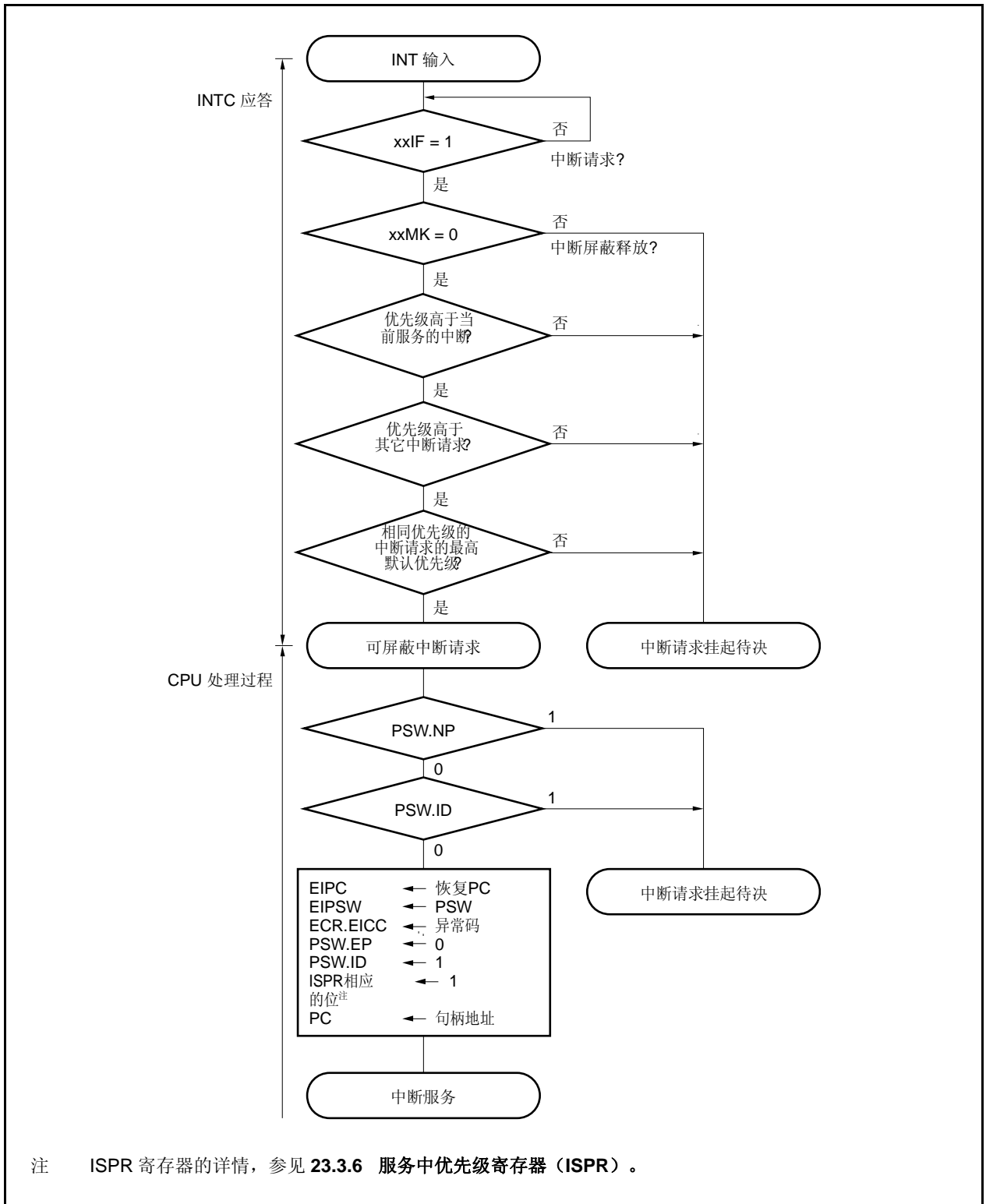
如果发生可屏蔽中断，CPU 执行以下处理过程，并控制转移至句柄程序。

- <1> 将 PC 保存至 EIPC。
- <2> 将 PSW 保存至 EIPSW。
- <3> 将异常码写入 ECR（EICC）的低半字。
- <4> 对 PSW.ID 位置 1，并对 PSW.EP 位清 0。
- <5> 将各个中断对应的处理程序地址设置到 PC 指针，并进行控制转移。

由中断控制寄存器（INTC）屏蔽的可屏蔽中断请求信号，和在另外的中断服务处理时（当 PSW.NP 位=1 或 ID 位=1）产生的可屏蔽中断请求信号，都被保持挂起状态存放在 INTC 内。这种情况下，如果取消对可屏蔽中断的屏蔽或使用 RETI 或 LDSR 指令将 NP 或 ID 清除为 0，则根据被挂起的可屏蔽中断请求的优先级判断，来启动一个新的可屏蔽中断服务。

下图显示了可屏蔽中断的服务过程：

图 23-5. 可屏蔽中断服务过程



23.3.2 恢复

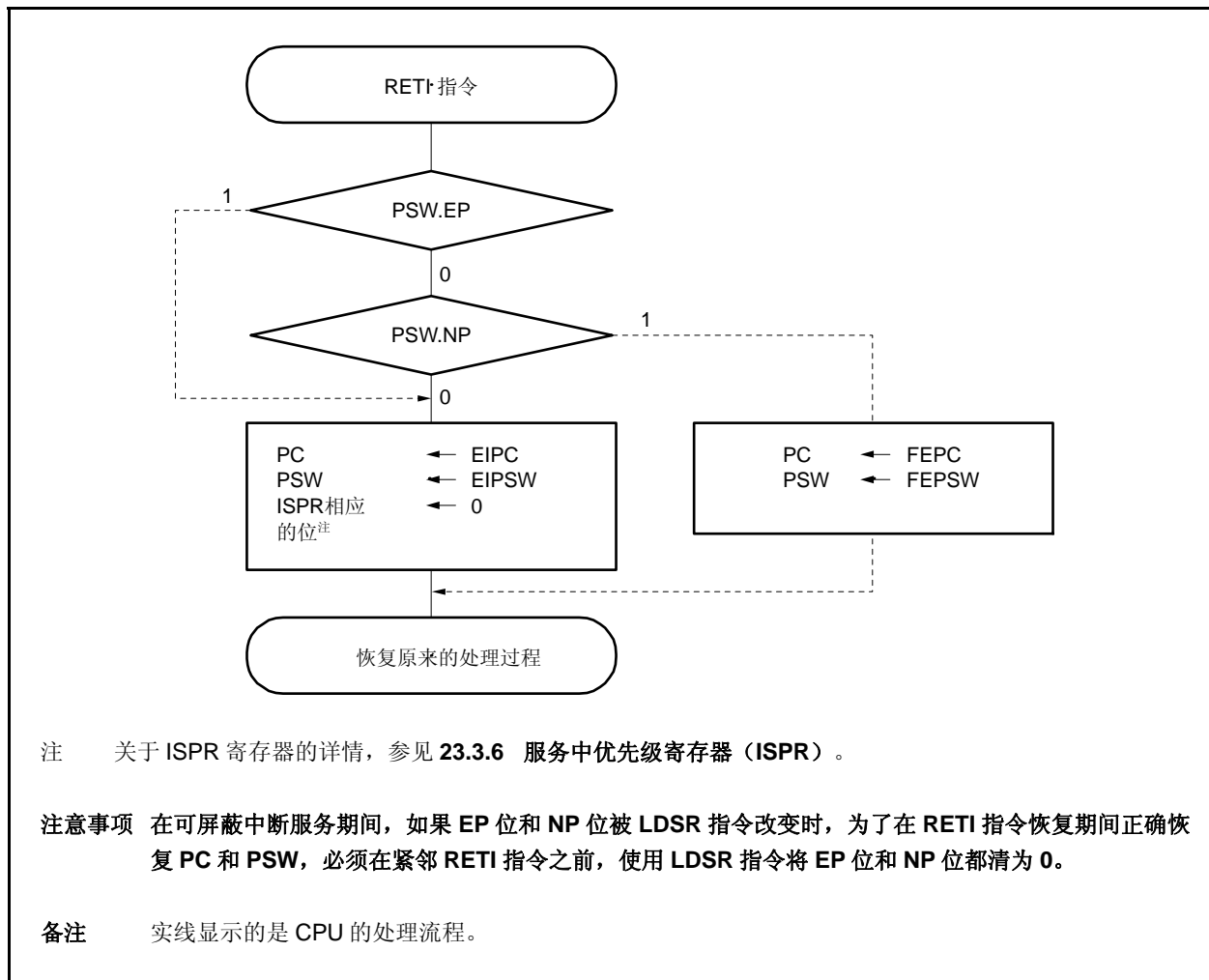
执行 RETI 指令，可以从可屏蔽中断服务中恢复。

当执行 RETI 指令时，CPU 执行以下处理过程，并且控制转移到被恢复的 PC 地址。

- <1> 因为 PSW.EP 位为 0 且 PSW.NP 位为 0，所以由 EIPC 和 EIPSW 载入 PC 和 PSW 的恢复值。
- <2> 控制转移返回到被恢复的 PC 和 PSW 的地址。

RETI 指令的处理过程如下所示：

图 23-6. RETI 指令处理过程



23.3.3 可屏蔽中断的优先级

INTC 执行多重中断服务，也就是在一个中断正在服务处理时响应另一个中断。多重中断可以通过优先等级进行控制。

有两种类型的优先级控制方式：基于默认优先级的控制，以及基于中断控制寄存器（xxICn）的中断优先级指定位（xxPRn）确定的可编程优先级控制。当两个或两个以上中断同时发生，且 xxPRn 位指定的优先级都相同时，按照每个中断请求类型预先设定（默认优先级）的优先级次序进行处理。更多信息参见表 23-2 和表 23-3。通过设置优先级标志，可编程优先级控制将中断请求信号划分为 8 个等级。

注意，当中断请求信号被响应时，PSW.ID 标志自动被置 1。所以，当使用多重中断时，应该预先将 ID 标志清为 0（比如，在中断服务程序中插入一条 EI 指令）来设置多重中断允许模式。

备注 xx: 每一个外设单元的标志名称（参见表 23-4 中断控制寄存器（xxICn））。
 n : 外设单元编号（参见表 23-4 中断控制寄存器（xxICn））。

图 23-7. 当一个中断服务时发生另一个中断请求的处理示例 (1/2)

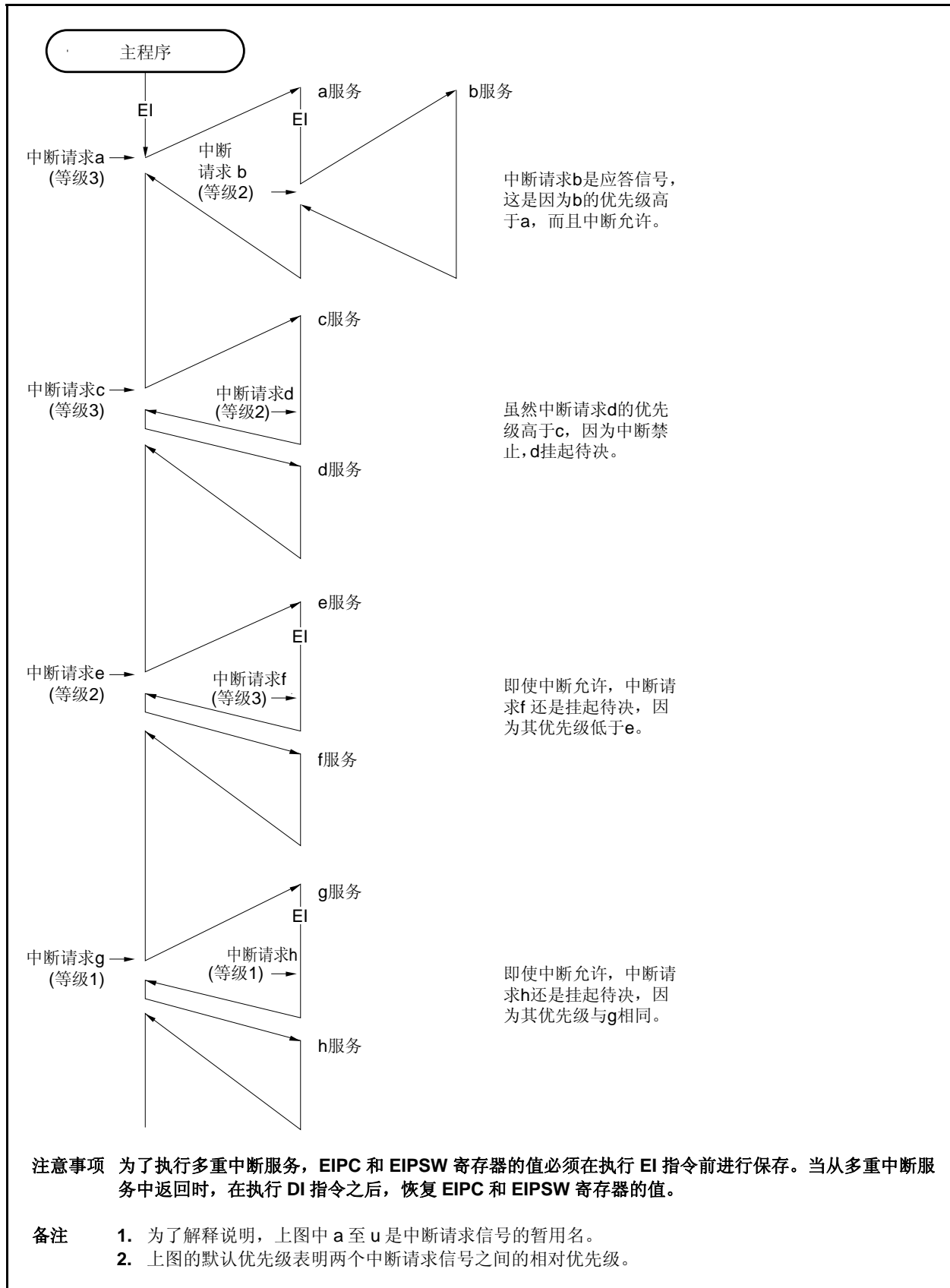


图 23-7. 一个中断服务当中而另一个中断请求发生的处理举例 (2/2)

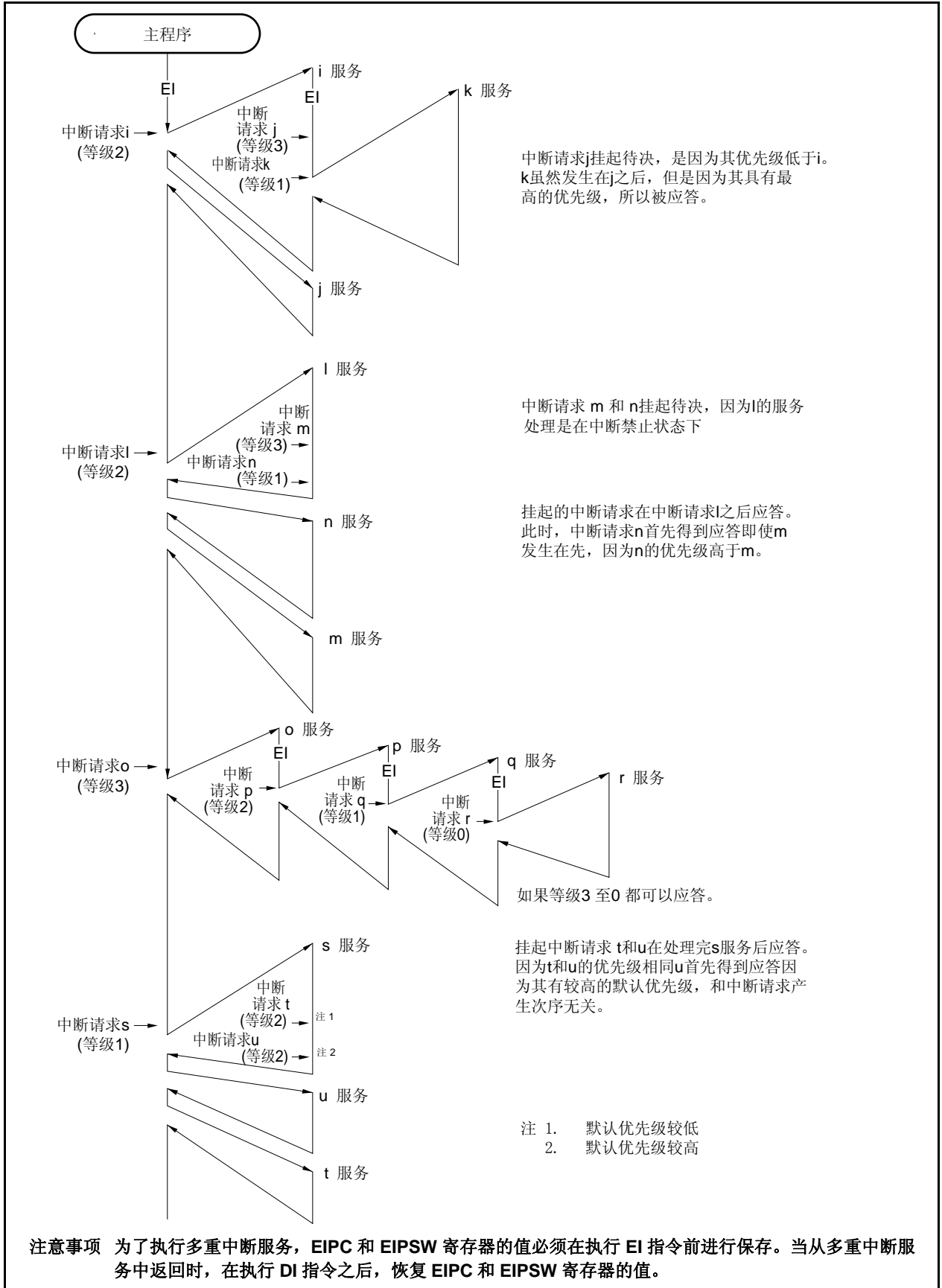
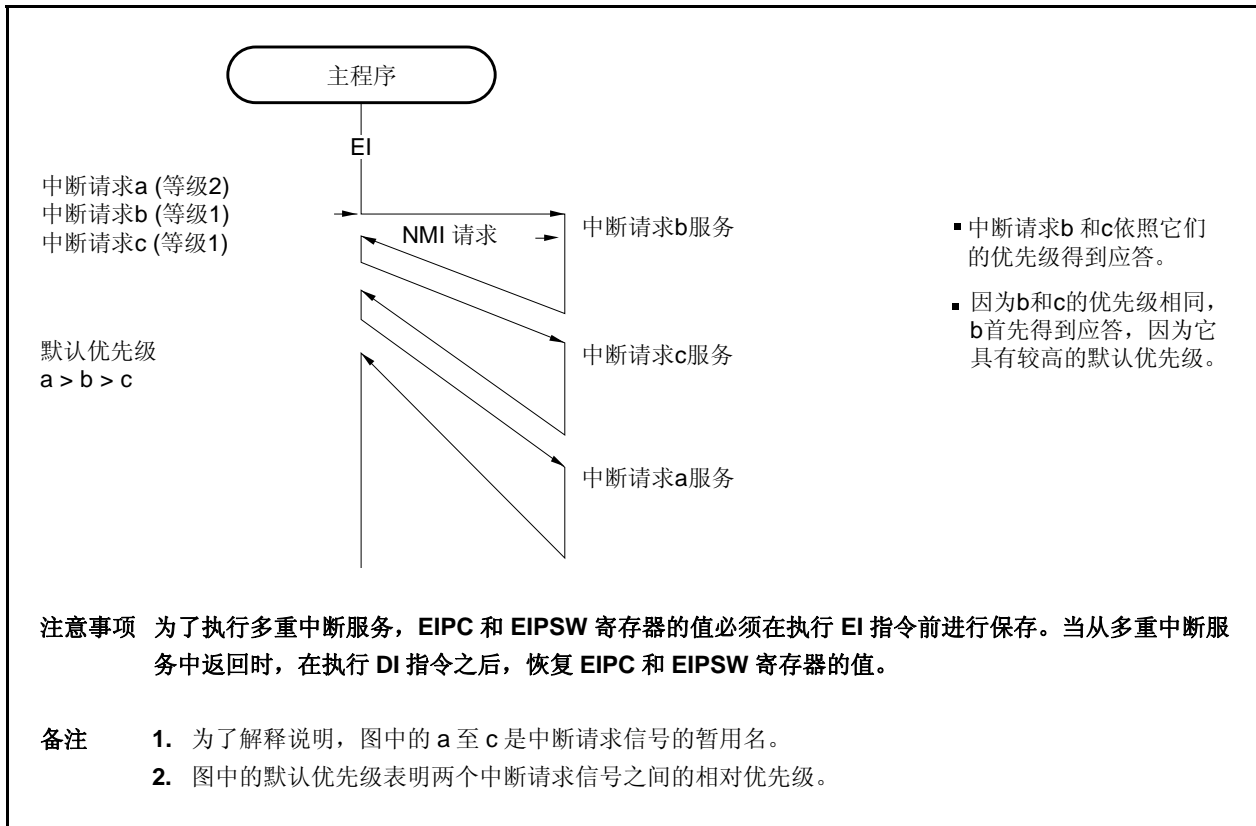


图 23-8. 同时发生的中断请求信号服务举例



23.3.4 中断控制寄存器 (xxICn)

每一个可屏蔽中断请求信号（可屏蔽中断）都被指定一个 xxICn 寄存器，用来设置每一个可屏蔽中断请求的控制条件。

该寄存器可以按字节或按位进行读写。

系统复位后，寄存器被置为 47H。

注意事项 读取 xxICn.xxIFn 位时，禁止中断（DI）或屏蔽中断。如果在中断允许（EI）时或在中断未屏蔽的情况下读取 xxIFn 位，当响应一个中断与读取该位的时序发生冲突时，可能读不出正确值。

复位后: 47H R/W 地址: FFFFF112H 至 FFFFF184H								
xxICn	<7>	<6>	5	4	3	2	1	0
	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0
xxIFn		中断请求标志 ^注						
0		不产生中断请求						
1		产生中断请求						
xxMKn		中断屏蔽标志						
0		中断服务允许						
1		中断服务禁止 (挂起待决)						
xxPRn2	xxPRn1	xxPRn0	中断优先级指定位					
0	0	0	指定等级0 (最高)					
0	0	1	指定等级1					
0	1	0	指定等级2					
0	1	1	指定等级3					
1	0	0	指定等级4					
1	0	1	指定等级5					
1	1	0	指定等级6					
1	1	1	指定等级7 (最低)					

注 如果一个中断请求信号得到响应，xxIFn 标志由硬件自动复位。

备注 xx: 每一个外设单元的标志名称（参见表 23-4 中断控制寄存器 (xxICn)）。

 n : 外设单元编号（参见表 23-4 中断控制寄存器 (xxICn)）。

中断控制寄存器的地址及位如下所示：

表 23-4. 中断控制寄存器 (xxICn) (1/3)

地址	寄存器	位							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	LVIIC	LVIIIF	LVIMK	0	0	0	LVIPR2	LVIPR1	LVIPR0
FFFFF112H	PIC0 [※]	PIF00	PMK00	0	0	0	PPR002	PPR001	PPR000
FFFFF114H	PIC01 [※]	PIF01	PMK01	0	0	0	PPR012	PPR011	PPR010
FFFFF116H	PIC02	PIF02	PMK02	0	0	0	PPR022	PPR021	PPR020
FFFFF118H	PIC03 [※]	PIF03	PMK03	0	0	0	PPR032	PPR031	PPR030
FFFFF11AH	PIC04 [※]	PIF04	PMK04	0	0	0	PPR042	PPR041	PPR040
FFFFF11CH	PIC05	PIF05	PMK05	0	0	0	PPR052	PPR051	PPR050
FFFFF11EH	PIC06 [※]	PIF06	PMK06	0	0	0	PPR062	PPR061	PPR060
FFFFF120H	PIC07	PIF07	PMK07	0	0	0	PPR072	PPR071	PPR070
FFFFF122H	PIC08	PIF08	PMK08	0	0	0	PPR082	PPR081	PPR080
FFFFF124H	PIC09	PIF09	PMK09	0	0	0	PPR092	PPR091	PPR090
FFFFF126H	PIC10	PIF10	PMK10	0	0	0	PPR102	PPR101	PPR100
FFFFF128H	PIC11	PIF11	PMK11	0	0	0	PPR112	PPR111	PPR110
FFFFF12AH	PIC12	PIF12	PMK12	0	0	0	PPR122	PPR121	PPR120
FFFFF12CH	PIC13	PIF13	PMK13	0	0	0	PPR132	PPR131	PPR130
FFFFF12EH	PIC14	PIF14	PMK14	0	0	0	PPR142	PPR141	PPR140
FFFFF130H	PIC15	PIF15	PMK15	0	0	0	PPR152	PPR151	PPR150
FFFFF132H	PIC16	PIF16	PMK16	0	0	0	PPR162	PPR161	PPR160
FFFFF134H	PIC17	PIF17	PMK17	0	0	0	PPR172	PPR171	PPR170
FFFFF136H	PIC18	PIF18	PMK18	0	0	0	PPR182	PPR181	PPR180
FFFFF138H	TAB0OVIC	TAB0OVIF	TAB0OVMK	0	0	0	TAB0OVPPR2	TAB0OVPPR1	TAB0OVPPR0
FFFFF13AH	TAB0CCIC0	TAB0CCIF0	TAB0CCMK0	0	0	0	TAB0CCPPR2	TAB0CCPPR1	TAB0CCPPR0
FFFFF13CH	TAB0CCIC1	TAB0CCIF1	TAB0CCMK1	0	0	0	TAB0CCPPR12	TAB0CCPPR11	TAB0CCPPR10
FFFFF13EH	TAB0CCIC2	TAB0CCIF2	TAB0CCMK2	0	0	0	TAB0CCPPR22	TAB0CCPPR21	TAB0CCPPR20
FFFFF140H	TAB0CCIC3	TAB0CCIF3	TAB0CCMK3	0	0	0	TAB0CCPPR32	TAB0CCPPR31	TAB0CCPPR30
FFFFF142H	TAB1OVIC	TAB1OVIF	TAB1OVMK	0	0	0	TAB1OVPPR2	TAB1OVPPR1	TAB1OVPPR0
FFFFF144H	TAB1CCIC0	TAB1CCIF0	TAB1CCMK0	0	0	0	TAB1CCPPR2	TAB1CCPPR1	TAB1CCPPR0
FFFFF146H	TAB1CCIC1	TAB1CCIF1	TAB1CCMK1	0	0	0	TAB1CCPPR12	TAB1CCPPR11	TAB1CCPPR10
FFFFF148H	TAB1CCIC2	TAB1CCIF2	TAB1CCMK2	0	0	0	TAB1CCPPR22	TAB1CCPPR21	TAB1CCPPR20
FFFFF14AH	TAB1CCIC3	TAB1CCIF3	TAB1CCMK3	0	0	0	TAB1CCPPR32	TAB1CCPPR31	TAB1CCPPR30
FFFFF14CH	TT0OVIC	TT0OVIF	TT0OVMK	0	0	0	TT0OVPPR2	TT0OVPPR1	TT0OVPPR0
FFFFF14EH	TT0CCIC0	TT0CCIF0	TT0CCMK0	0	0	0	TT0CCPPR2	TT0CCPPR1	TT0CCPPR0
FFFFF150H	TT0CCIC1	TT0CCIF1	TT0CCMK1	0	0	0	TT0CCPPR12	TT0CCPPR11	TT0CCPPR10
FFFFF152H	TT0IECIC	TT0IECIF	TT0IECMK	0	0	0	TT0IECPPR2	TT0IECPPR1	TT0IECPPR0
FFFFF154H	TAA0OVIC	TAA0OVIF	TAA0OVMK	0	0	0	TAA0OVPPR2	TAA0OVPPR1	TAA0OVPPR0
FFFFF156H	TAA0CCIC0	TAA0CCIF0	TAA0CCMK0	0	0	0	TAA0CCPPR2	TAA0CCPPR1	TAA0CCPPR0
FFFFF158H	TAA0CCIC1	TAA0CCIF1	TAA0CCMK1	0	0	0	TAA0CCPPR12	TAA0CCPPR11	TAA0CCPPR10
FFFFF15AH	TAA1OVIC	TAA1OVIF	TAA1OVMK	0	0	0	TAA1OVPPR2	TAA1OVPPR1	TAA1OVPPR0
FFFFF15CH	TAA1CCIC0	TAA1CCIF0	TAA1CCMK0	0	0	0	TAA1CCPPR2	TAA1CCPPR1	TAA1CCPPR0
FFFFF15EH	TAA1CCIC1	TAA1CCIF1	TAA1CCMK1	0	0	0	TAA1CCPPR12	TAA1CCPPR11	TAA1CCPPR10

注 仅限 V850ES/JH3-U

表 23-4. 中断控制寄存器 (xxICn) (2/3)

地址	寄存器	位							
		<7>	<6>	5	4	3	2	1	0
FFFFF160H	TAA2OVIC	TAA2OVIF	TAA2OVMK	0	0	0	TAA2OVPPR2	TAA2OVPPR1	TAA2OVPPR0
FFFFF162H	TAA2CCIC0	TAA2CCIF0	TAA2CCMK0	0	0	0	TAA2CCPPR02	TAA2CCPPR01	TAA2CCPPR00
FFFFF164H	TAA2CCIC1	TAA2CCIF1	TAA2CCMK1	0	0	0	TAA2CCPPR12	TAA2CCPPR11	TAA2CCPPR10
FFFFF166H	TAA3OVIC	TAA3OVIF	TAA3OVMK	0	0	0	TAA3OVPPR2	TAA3OVPPR1	TAA3OVPPR0
FFFFF168H	TAA3CCIC0	TAA3CCIF0	TAA3CCMK0	0	0	0	TAA3CCPPR02	TAA3CCPPR01	TAA3CCPPR00
FFFFF16AH	TAA3CCIC1	TAA3CCIF1	TAA3CCMK1	0	0	0	TAA3CCPPR12	TAA3CCPPR11	TAA3CCPPR10
FFFFF16CH	TAA4OVIC	TAA4OVIF	TAA4OVMK	0	0	0	TAA4OVPPR2	TAA4OVPPR1	TAA4OVPPR0
FFFFF16EH	TAA4CCIC0	TAA4CCIF0	TAA4CCMK0	0	0	0	TAA4CCPPR02	TAA4CCPPR01	TAA4CCPPR00
FFFFF170H	TAA4CCIC1	TAA4CCIF1	TAA4CCMK1	0	0	0	TAA4CCPPR12	TAA4CCPPR11	TAA4CCPPR10
FFFFF172H	TAA5OVIC	TAA5OVIF	TAA5OVMK	0	0	0	TAA5OVPPR2	TAA5OVPPR1	TAA5OVPPR0
FFFFF174H	TAA5CCIC0	TAA5CCIF0	TAA5CCMK0	0	0	0	TAA5CCPPR02	TAA5CCPPR01	TAA5CCPPR00
FFFFF176H	TAA5CCIC1	TAA5CCIF1	TAA5CCMK1	0	0	0	TAA5CCPPR12	TAA5CCPPR11	TAA5CCPPR10
FFFFF178H	TM0EQIC0	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR02	TM0EQPR01	TM0EQPR00
FFFFF17AH	TM1EQIC0	TM1EQIF0	TM1EQMK0	0	0	0	TM1EQPR02	TM1EQPR01	TM1EQPR00
FFFFF17CH	TM2EQIC0	TM2EQIF0	TM2EQMK0	0	0	0	TM2EQPR02	TM2EQPR01	TM2EQPR00
FFFFF17EH	TM3EQIC0	TM3EQIF0	TM3EQMK0	0	0	0	TM3EQPR02	TM3EQPR01	TM3EQPR00
FFFFF180H	CF0RIC/ IICIC1	CF0RIF/ IICIF1	CF0RMK/ IICMK1	0	0	0	CF0RPPR2/ IICPPR12	CF0RPPR1/ IICPPR11	CF0RPPR0/ IICPPR10
FFFFF182H	CF0TIC	CF0TIF	CF0TMK	0	0	0	CF0TPPR2	CF0TPPR1	CF0TPPR0
FFFFF184H	CF1RIC	CF1RIF	CF1RMK	0	0	0	CF1RPPR2	CF1RPPR1	CF1RPPR0
FFFFF186H	CF1TIC	CF1TIF	CF1TMK	0	0	0	CF1TPPR2	CF1TPPR1	CF1TPPR0
FFFFF188H	CF2RIC	CF2RIF	CF2RMK	0	0	0	CF2RPPR2	CF2RPPR1	CF2RPPR0
FFFFF18AH	CF2TIC	CF2TIF	CF2TMK	0	0	0	CF2TPPR2	CF2TPPR1	CF2TPPR0
FFFFF18CH	CF3RIC	CF3RIF	CF3RMK	0	0	0	CF3RPPR2	CF3RPPR1	CF3RPPR0
FFFFF18EH	CF3TIC	CF3TIF	CF3TMK	0	0	0	CF3TPPR2	CF3TPPR1	CF3TPPR0
FFFFF190H	CF4RIC	CF3RIF	CF3RMK	0	0	0	CF3RPPR2	CF3RPPR1	CF3RPPR0
FFFFF192H	CF4TIC	CF3TIF	CF3TMK	0	0	0	CF3TPPR2	CF3TPPR1	CF3TPPR0
FFFFF194H	UC0RIC	UC0RIF	UC0RMK	0	0	0	UC0RPPR2	UC0RPPR1	UC0RPPR0
FFFFF196H	UC0TIC	UC0TIF	UC0TMK	0	0	0	UC0TPPR2	UC0TPPR1	UC0TPPR0
FFFFF198H	UC1RIC/ IICIC2	UC1RIF/ IICIF2	UC1RMK/ IICMK2	0	0	0	UC1RPPR2/ IICPPR22	UC1RPPR1/ IICPPR21	UC1RPPR0/ IICPPR20
FFFFF19AH	UC1TIC	UC1TIF	UC1TMK	0	0	0	UC1TPPR2	UC1TPPR1	UC1TPPR0
FFFFF19CH	UC2RIC	UC2RIF	UC2RMK	0	0	0	UC2RPPR2	UC2RPPR1	UC2RPPR0
FFFFF19EH	UC2TIC	UC2TIF	UC2TMK	0	0	0	UC2TPPR2	UC2TPPR1	UC2TPPR0
FFFFF1A0H	UC3RIC/ IICIC0	UC3RIF/ IICIF0	UC3RMK/ IICMK0	0	0	0	UC3RPPR2/ IICPPR02	UC3RPPR1/ IICPPR01	UC3RPPR0/ IICPPR00
FFFFF1A2H	UC3TIC	UC3TIF	UC3TMK	0	0	0	UC3TPPR2	UC3TPPR1	UC3TPPR0
FFFFF1A4H	UC4RIC	UC4RIF	UC4RMK	0	0	0	UC4RPPR2	UC4RPPR1	UC4RPPR0
FFFFF1A6H	UC4TIC	UC4TIF	UC4TMK	0	0	0	UC4TPPR2	UC4TPPR1	UC4TPPR0
FFFFF1A8H	ADIC	ADIF	ADMK	0	0	0	ADPPR2	ADPPR1	ADPPR0
FFFFF1AAH	DMAIC0	DMAIC0	DMAMK0	0	0	0	DMAPPR02	DMAPPR01	DMAPPR00
FFFFF1ACH	DMAIC1	DMAIC1	DMAMK1	0	0	0	DMAPPR12	DMAPPR11	DMAPPR10

表 23-4. 中断控制寄存器 (xxICn) (3/3)

地址	寄存器	位							
		<7>	<6>	5	4	3	2	1	0
FFFF1AEH	DMAIC2	DMAIC2	DMAMK2	0	0	0	DMAPPR22	DMAPPR21	DMAPPR20
FFFF1B0H	DMAIC3	DMAIC3	DMAMK3	0	0	0	DMAPPR32	DMAPPR31	DMAPPR30
FFFF1B2H	KRIC	KRIF	KRMK	0	0	0	KRPPR2	KRPPR1	KRPPR0
FFFF1B4H	RTC0IC	RTC0IF	RTC0MK	0	0	0	RTC0PPR2	RTC0PPR1	RTC0PPR0
FFFF1B6H	RTC1IC	RTC1IF	RTC1MK	0	0	0	RTC1PPR2	RTC1PPR1	RTC1PPR0
FFFF1B8H	RTC2IC	RTC2IF	RTC2MK	0	0	0	RTC2PPR2	RTC2PPR1	RTC2PPR0
FFFF1C2H	UHIC0	UHIC0	UHMK0	0	0	0	UHPPR02	UHPPR01	UHPPR00
FFFF1C4H	UHIC1	UHIC1	UHMK1	0	0	0	UHPPR12	UHPPR11	UHPPR10
FFFF1C6H	UHIC2	UHIC2	UHMK2	0	0	0	UHPPR22	UHPPR21	UHPPR20
FFFF1C8H	UFIC0	UFIC0	UFMK0	0	0	0	UFPPR02	UFPPR01	UFPPR00
FFFF1CAH	UFIC1	UFIC1	UFMK1	0	0	0	UFPPR12	UFPPR11	UFPPR10

23.3.5 中断屏蔽寄存器 0 至 5 (IMR0 至IMR5)

IMR0 至 IMR5 寄存器用于设置可屏蔽中断的中断屏蔽状态。IMR0 至 IMR5 寄存器的 xxMKn 位等同于 xxCn.xxMKn 位。

IMRm 寄存器可以按 16 位宽度进行读取或写入。

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，则这些寄存器可以按字节或按位进行读写 (m=0 至 5)。

系统复位后，这些寄存器被设置为 FFFFH。

注意事项 设备文件将 xxCn.xxMKn 位定义为保留字。如果使用 xxMKn 进行位操作，xxCn 寄存器的内容而不是 IMRm 寄存器被改写 (结果，IMRm 寄存器的内容也被改写)。

(1/2)

复位后: FFFFH		R/W	地址: IMR5 FFFFF10AH, IMR5L FFFFF10AH, IMR5H FFFFF10BH					
	15	14	13	12	11	10	9	8
IMR5 (IMR5H ^注)	1	1	UFMK1	UFMK0	UHMK2	UHMK1	UHMK0	1
	7	6	5	4	3	2	1	0
IMR5L	1	1	1	RTC2MK	RTC1MK	RTC0MK	KRMK	DMAMK3
复位后: FFFFH		R/W	地址: IMR4 FFFFF108H, IMR4L FFFFF108H, IMR4H FFFFF109H					
	15	14	13	12	11	10	9	8
IMR4 (IMR4H ^注)	DMAMK2	DMAMK1	DMAMK0	ADMK	UC4TMK	UC4RMK	UC3TMK	UC3RMK/ IICMK0
	7	6	5	4	3	2	1	0
IMR4L	UC2TMK	UC2RMK	UC1TMK	UC1RMK/ IICMK2	UC0TMK	UC0RMK	CF3TMK	CF3RMK
复位后: FFFFH		R/W	地址: IMR3 FFFFF106H, IMR3L FFFFF106H, IMR3H FFFFF107H					
	15	14	13	12	11	10	9	8
IMR3 (IMR3H ^注)	CF3TMK	CF3RMK	CF2TMK	CF2RMK	CF1TMK	CF1RMK	CF0TMK	CF0RMK/ IICMK1
	7	6	5	4	3	2	1	0
IMR3L	TM3EQMK0	TM2EQMK0	TM1EQMK0	TM0EQMK0	TAA5CCMK1	TAA5CCMK0	TAA5OVMK	TAA4CCMK1
注	若要按字节或按位读取 IMR3 至 IMR5 寄存器的第 8 位至第 15 位，需指定这些位作为 IMR3H 至 IMR5H 寄存器的第 0 位至第 7 位。							
注意事项	将 IMR5 寄存器的第 5 位至第 8 位，以及第 14 位和第 15 位设置为 1。如果改变这些位的设置，操作将无法保证。							
备注	xx:	每一个外设单元的标志名称 (参见表 23-4 中断控制寄存器 (xxICn))。						
	n :	外设单元编号 (参见表 23-4 中断控制寄存器 (xxICn))。						

复位后: FFFFH R/W 地址: IMR2 FFFFF104H,
IMR2L FFFFF104H, IMR2H FFFFF105H

	15	14	13	12	11	10	9	8
IMR2 (IMR2H ^{注1})	TAA4CCMK0	TAA4OVMK	TAA3CCMK1	TAA3CCMK0	TAA3OVMK	TAA2CCMK1	TAA2CCMK0	TAA2OVMK
	7	6	5	4	3	2	1	0
IMR2L	TAA1CCMK1	TAA1CCMK0	TAA1OVMK	TAA0CCMK1	TAA0CCMK0	TAA0OVMK	TMTIECMK	TMT0CCMK1

复位后: FFFFH R/W 地址: IMR1 FFFFF102H,
IMR1L FFFFF102H, IMR1H FFFFF103H

	15	14	13	12	11	10	9	8
IMR1 (IMR1H ^{注1})	TT0CCMK0	TT0OVMK	TAB1CCMK3	TAB1CCMK2	TAB1CCMK1	TAB1CCMK0	TAB1OVMK	TAB0CCMK3
	7	6	5	4	3	2	1	0
IMR1L	TAB0CCMK2	TAB0CCMK1	TAB0CCMK0	TAB0OVMK	PMK18	PMK17	PMK16	PMK15

复位后: FFFFH R/W 地址: IMR0 FFFFF100H,
IMR0L FFFFF100H, IMR0H FFFFF101H

	15	14	13	12	11	10	9	8
IMR0 (IMR0H ^{注1})	PMK14	PMK13	PMK12	PMK11	PMK10	PMK09	PMK08	PMK07
	7	6	5	4	3	2	1	0
IMR0L	PMK06 ^{注2}	PMK05	PMK04 ^{注2}	PMK03 ^{注2}	PMK02	PMK01 ^{注2}	PMK00 ^{注2}	LVIMK

xxMKn	中断屏蔽标志的设置
0	中断服务使能
1	中断服务禁止

- 注**
- 若要按字节或按位读取 IMR0 至 IMR2 寄存器的第 8 位至第 15 位, 需指定这些位作为 IMR0H 至 IMR2H 寄存器的第 0 位至第 7 位。
 - 仅限 V850ES/JH3-U

备注

xx: 每一个外设单元的标志名称 (参见表 23-4 中断控制寄存器 (xxICn))。

n: 外设单元编号 (参见表 23-4 中断控制寄存器 (xxICn))。

23.3.6 正在服务中优先级寄存器 (ISPR)

ISPR 寄存器保存当前响应的可屏蔽中断的优先级。当响应一个中断请求信号时，该寄存器中与中断请求信号优先级对应的位被置为 1，并且在中断服务期间一直保持置位。

当执行 RETI 指令时，对应最高优先级中断请求信号的位被硬件自动清 0。但是，从不可屏蔽中断服务或异常处理中返回时，并不被复位为 0。

该寄存器为只读寄存器，按字节或按位读取。

系统复位后，该寄存器被设为 00H。

注意事项 中断允许 (EI) 状态下，如果在读取 ISPR 寄存器的同时响应一个中断，读取的 ISPR 寄存器值可能是响应中断后其对应位被置位后的值。为了在响应中断前准确读出 ISPR 寄存器的值，应在中断禁止 (DI) 的状态下读取。

复位后: 00H	R	地址: FFFFF1FAH						
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0
	ISPRn	当前正在应答中断的优先级						
	0	没有响应优先级为n的中断请求信号						
	1	正在响应优先级为n的中断请求信号						
备注	n=0 至 7 (优先级等级)							

23.3.7 ID标志

该标志用于控制可屏蔽中断的操作状态，并且存储有关使能或禁止中断请求信号的控制信息。中断禁止标志（ID）位于 PSW 内。

系统复位后，该标志被置为 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	可屏蔽中断服务说明 ^注
0	可屏蔽中断请求信号应答使能
1	可屏蔽中断请求信号应答禁止（挂起）

注 中断禁止标志（ID）功能
 该位由 DI 指令置位（为 1），由 EI 指令清除为 0。其值也可由 RETI 指令或 PSW 相关的 LDSR 指令来改变。
 不管该标志如何，不可屏蔽中断请求信号和异常事件都可以得到响应。当一个可屏蔽中断请求信号被响应时，ID 标志由硬件自动置 1。
 当 xxICn.xxIFn 位置为 1 时，在中断响应禁止期间（ID 标志=1）产生的中断请求信号可以得到响应，并且 ID 标志被清为 0。

23.3.8 看门狗定时器模式寄存器 2（WDTM2）

该寄存器可以按字节或按位进行读写（详情参见 第十三章 看门狗定时器 2 功能）。

系统复位后，该寄存器被设置为 67H。

复位后: 67H R/W 地址: FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	0	0	0	0	0

WDM21	WDM20	看门狗定时器操作模式选择
0	0	停止操作
0	1	不可屏蔽中断请求模式
1	x	复位模式 (初始值)

23.4 软件异常

当 CPU 执行 TRAP 指令时，产生软件异常，并且总是能够得到响应。

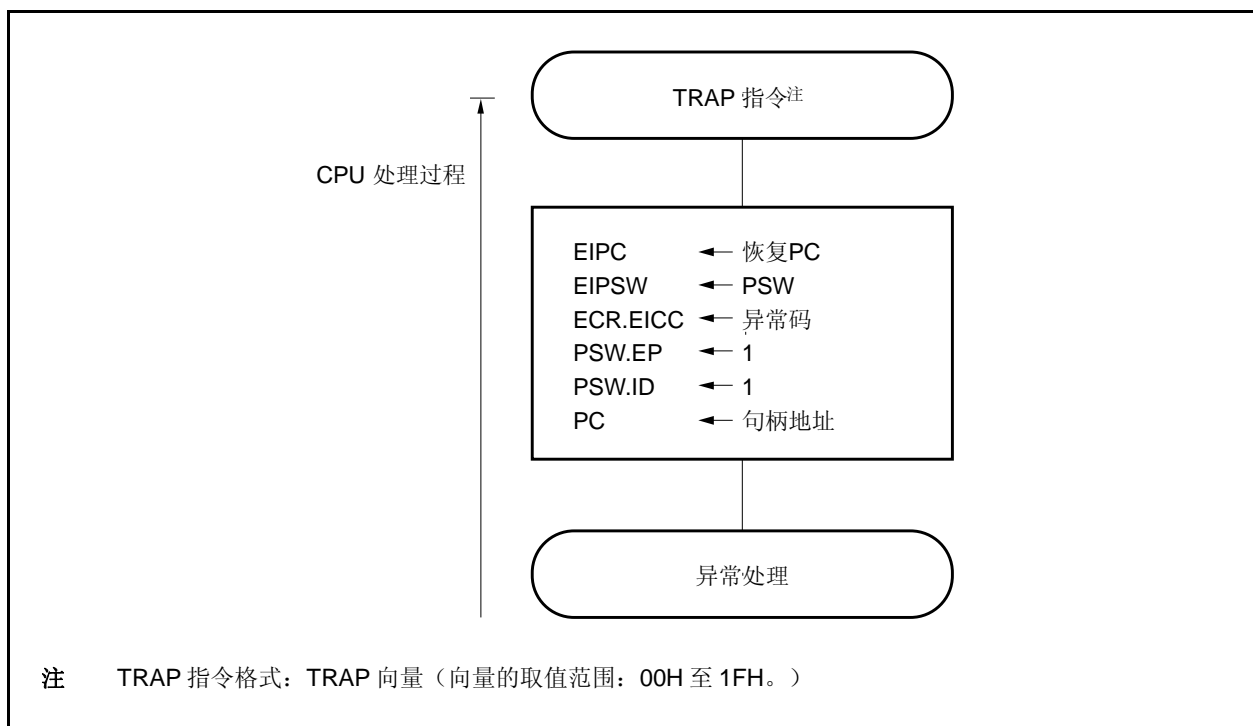
23.4.1 操作

如果发生软件异常，CPU 执行以下处理，并控制转移到句柄程序。

- <1> 将 PC 保存至 EIPC。
- <2> 将 PSW 保存至 EIPSW。
- <3> 写入异常码到 ECR（中断源）的低 16 位（EICC）。
- <4> 将 PSW.EP 和 PSW.ID 位置为 1。
- <5> 将软件异常对应的句柄地址（00000040H 或 00000050H）设置到 PC 指针，并进行控制转移。

软件异常的处理过程如下所示：

图 23-9. 软件异常的处理过程



句柄地址由 TRAP 指令的操作数（向量）决定。如果向量为 00H 至 0FH，句柄地址是 00000040H，而如果向量为 10H 至 1FH，句柄地址是 00000050H。

23.4.2 恢复

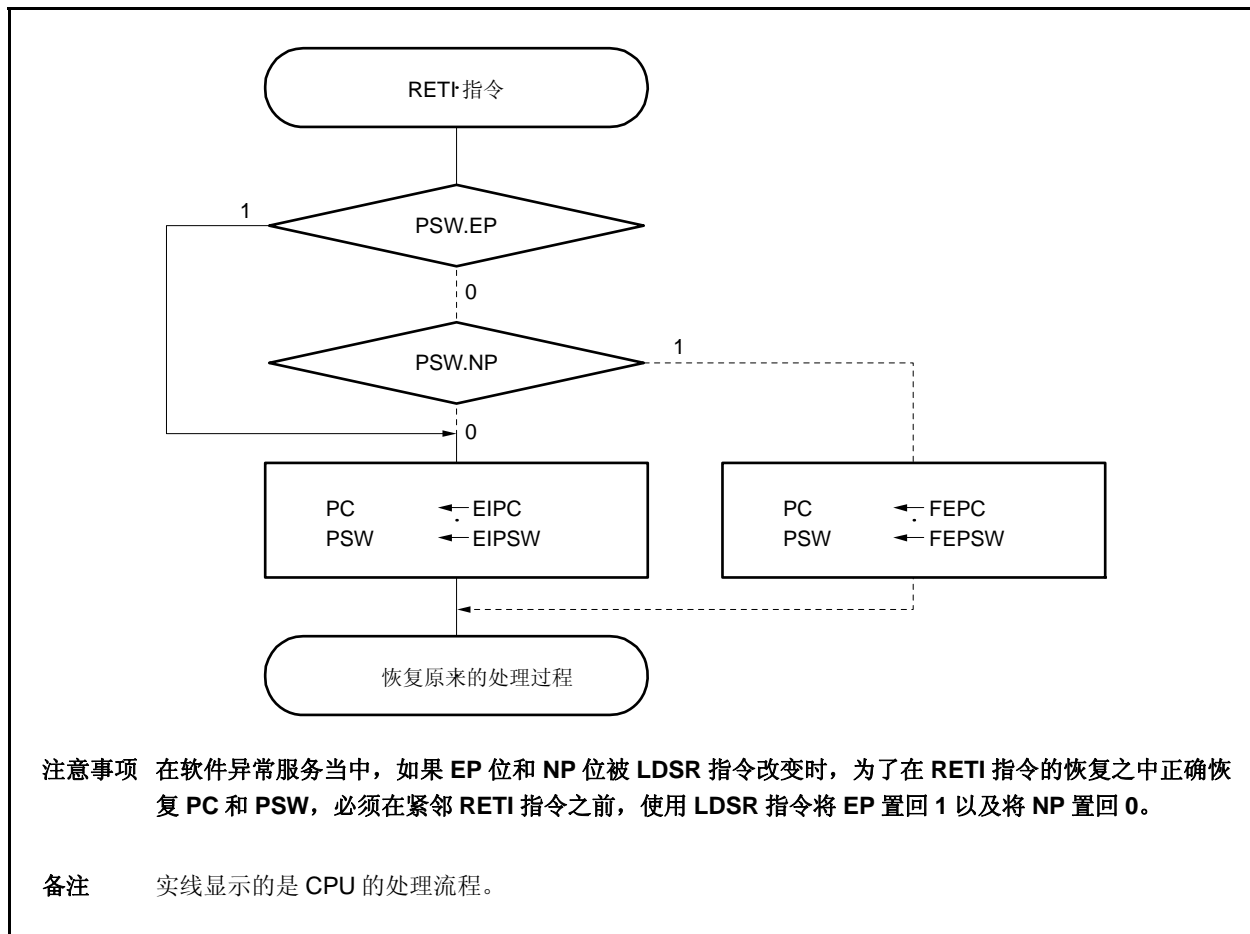
执行 RETI 指令，可以从软件异常处理中恢复。

通过执行 RETI 指令，CPU 执行以下处理并将控制转移至被恢复的 PC 地址。

- <1> 由于 PSW.EP 位为 1，由 EIPC 和 EIPSW 载入被恢复的 PC 和 PSW。
- <2> 控制转移至被恢复的 PC 和 PSW 的地址。

RETI 指令的处理过程如下所示：

图 23-10. RETI 指令处理过程



23.4.3 EP标志

EP 是一个状态标志，用于指示异常处理正在处理中。当异常发生时该位被置位。

复位后: 00000020H

	31		8	7	6	5	4	3	2	1	0
PSW	0			NP	EP	ID	SAT	CY	OV	S	Z

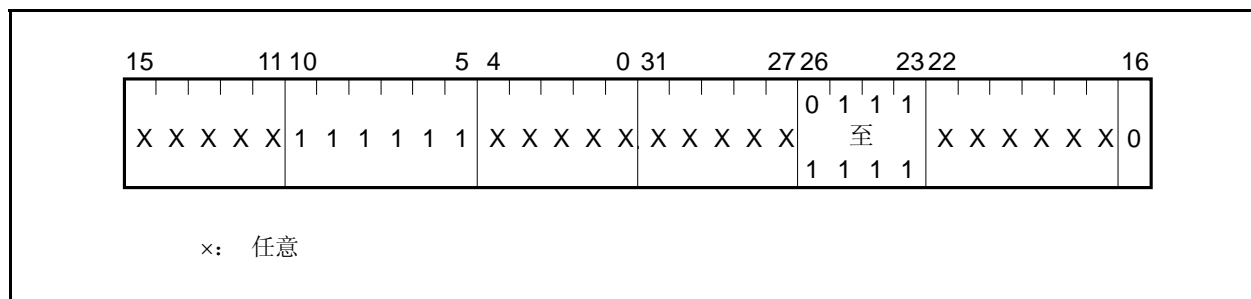
EP	异常处理状态
0	没有进行异常处理。
1	正在进行异常处理。

23.5 异常陷阱

异常陷阱是一个中断，当非法执行指令时产生异常陷阱中断请求。在 V850ES/JG3-U 和 V850ES/JH3-U 单片机中，非法操作码陷阱（ILGOP：非法操作码陷阱）被认为是一个异常陷阱。

23.5.1 非法操作码

非法操作码被定义为一条指令，其指令操作码（第 10 位至第 5 位）=111111B，副操作码（第 26 位至第 23 位）=0111B 至 1111B，另一个副操作码（第 16 位）=0B。当执行这样一条指令时，就会产生一个异常陷阱。



注意事项 建议不要使用非法操作码，因为这些指令将来可能被指定为新用途。

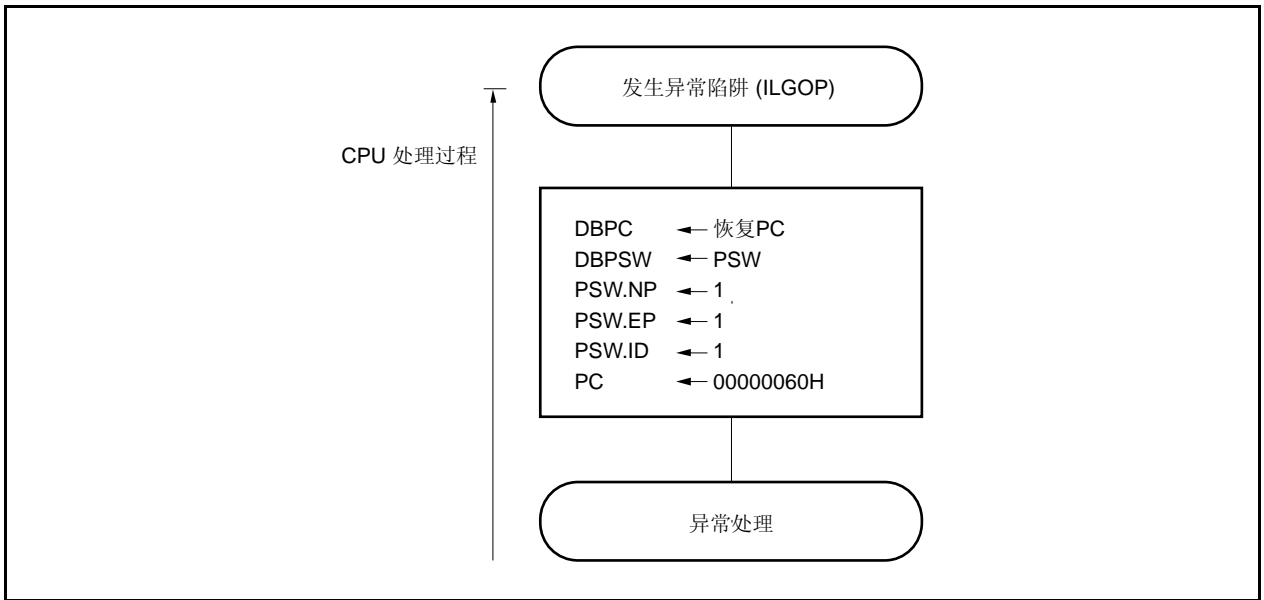
(1) 操作

当发生异常陷阱时，CPU 执行以下处理，并将控制转移到句柄程序：

- <1> 将待恢复的 PC 保存至 DBPC。
- <2> 将当前 PSW 保存至 DBPSW。
- <3> 将 PSW.NP, PSW.EP 和 PSW.ID 位置为 1。
- <4> 将异常陷阱对应的处理程序地址（00000060H）设置到 PC 指针，并进行控制转移。

异常陷阱的处理过程如下所示：

图 23-11. 异常陷阱处理过程



(2) 恢复

执行 DBRET 指令，可以从异常陷阱中恢复。

通过执行 DBRET 指令，CPU 进行以下处理并将控制转移至被恢复的 PC 地址。

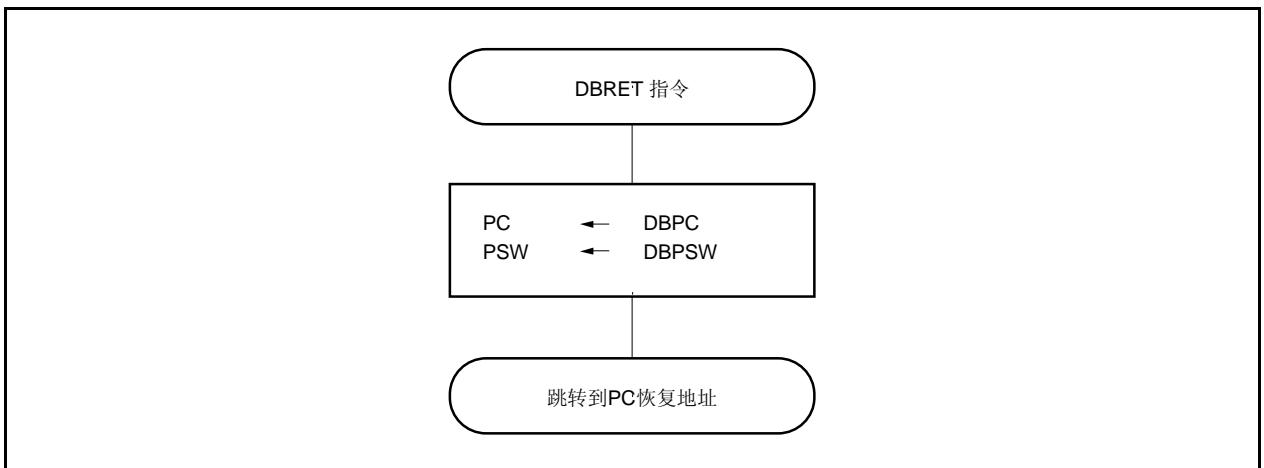
<1> 从 DBPC 和 DBPSW 中载入被恢复的 PC 和 PSW。

<2> 控制转移至被恢复的 PC 和 PSW 所指定的地址。

注意事项 只有在执行非法操作码和 DBRET 指令之间的间隔内，才可以访问 DBPC 和 DBPSW。

从异常陷阱中恢复的处理过程如下所示：

图 23-12. 异常陷阱恢复处理过程



23.5.2 调试陷阱

调试陷阱是一个异常，在执行 DBTRAP 指令时产生，总是可以得到响应。

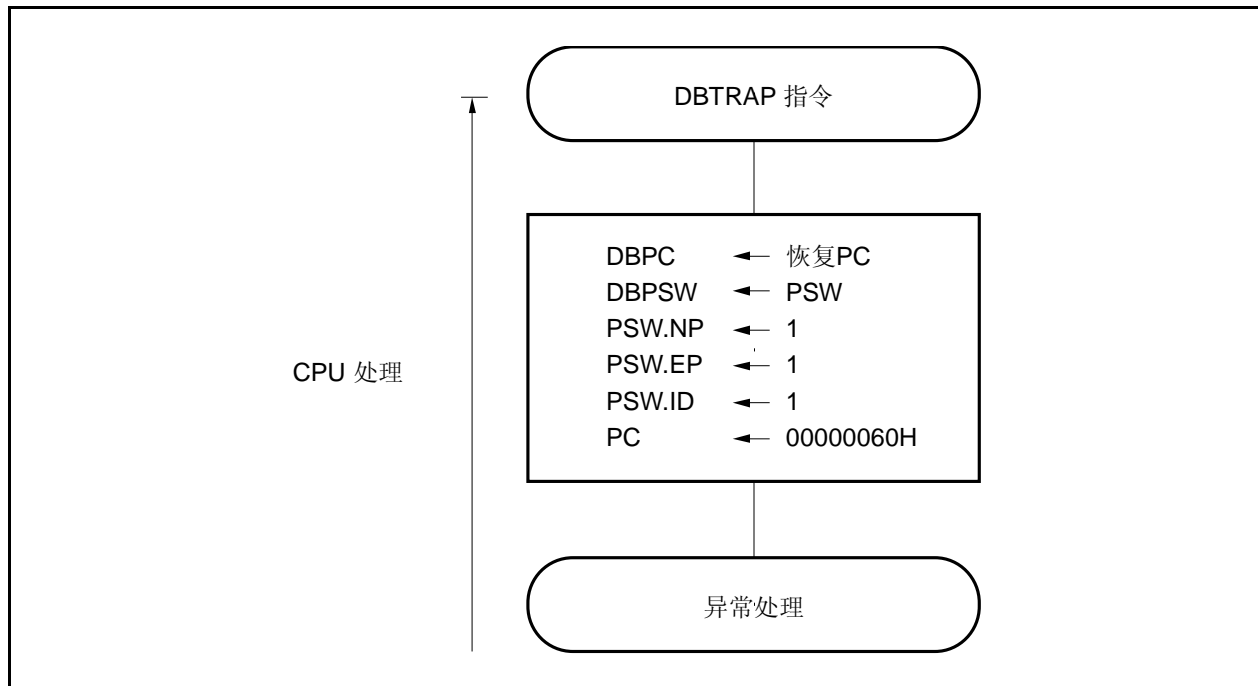
(1) 操作

当发生调试陷阱时，CPU 执行以下处理过程。

- <1> 将待恢复的 PC 保存至 DBPC。
- <2> 将当前 PSW 保存至 DBPSW。
- <3> 将 PSW.NP, PSW.EP 和 PSW.ID 位置为 1。
- <4> 将调试陷阱的句柄地址 (00000060H) 设置到 PC 指针，并进行控制转移。

调试陷阱的处理过程如下所示：

图 23-13. 调试陷阱处理过程



(2) 恢复

执行 DBRET 指令，可以从异常陷阱中进行恢复。

当执行 DBRET 指令时，CPU 执行以下处理过程并将控制转移到恢复 PC 地址。

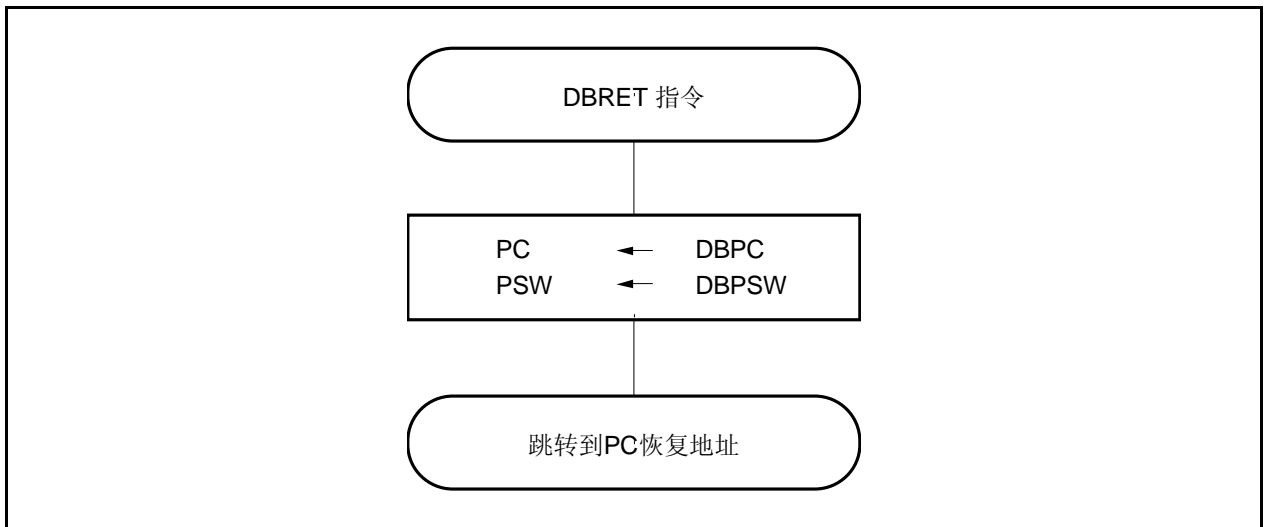
<1> 从 DBPC 和 DBPSW 中读取被恢复的 PC 和 PSW。

<2> 控制转移返回到被恢复的 PC 和 PSW 的地址。

注意事项 只有在执行 DBTRAP 和 DBRET 指令之间的间隔内，才可以访问 DBPC 和 DBPSW。

从调试陷阱中恢复的处理过程如下所示：

图 23-14. 从调试陷阱中恢复的处理流程



23.6 外部中断请求输入引脚（NMI和INTP00至INTP18）

23.6.1 噪声消除

（1）NMI 引脚的噪声消除

NMI 引脚具有模拟延迟的内部噪声消除电路。因此，输入 NMI 引脚的电平需要保持特定的或更长的时间，否则将不会被检测为边沿。因此，经过一定的时间之后，才能检测到边沿。

NMI 引脚能用来解除 STOP 模式。在 STOP 模式下，使用系统时钟，所以噪声无法消除，因为此时内部系统时钟是停止的。

（2）INTP00, INTP01 和 INTP03 至 INTP18 引脚的噪声消除

INTP00, INTP01 和 INTP03 至 INTP18 引脚具有模拟延迟的内部噪声消除电路。因此，输入 NMI 引脚的电平需要保持特定的或更长的时间，否则将不会被检测为边沿。因此，经过一定的时间之后，才能检测到边沿。

（3）INTP02 引脚的噪声消除

INTP02 引脚具有模拟延迟的内部噪声消除电路和内部数字噪声消除电路。使用噪声消除控制寄存器(INTNFC)可以选择其中任意一个（参见 23.6.2 (7)）。

23.6.2 边沿检测

NMI 和 INTP00 至 INTP18 各个引脚输入信号的有效边沿可以从以下 4 种类型选择：

- 上升沿
- 下降沿
- 上升和下降双边沿
- 无边沿检测

在复位后，不检测 NMI 引脚的边沿。所以，除非使用 INTF0 和 INTRO 寄存器指定一种有效边沿，否则不响应中断请求信号（NMI 引脚作为正常端口引脚功能）。

(1) 外部中断下降沿和上升沿指定寄存器 0 (INTF0, INTR0)

INTF0 和 INTR0 寄存器是 8 位寄存器，第 2 位用于指定 NMI 引脚的边沿检测，通过第 0 位、第 1 位以及第 3 位至第 5 位来指定外部中断引脚 (INTP00 至 INTP04) 的下降沿和上升沿检测。

这些寄存器可以按字节或按位进行读写。

系统复位后，这些寄存器被设置为 00H。

注意事项 当引脚功能由外部中断功能（复用功能）改变为端口功能时，可能会检测到一个边沿。所以，将 INTF0n 和 INTR0n 位清为 00，然后再设置端口模式。

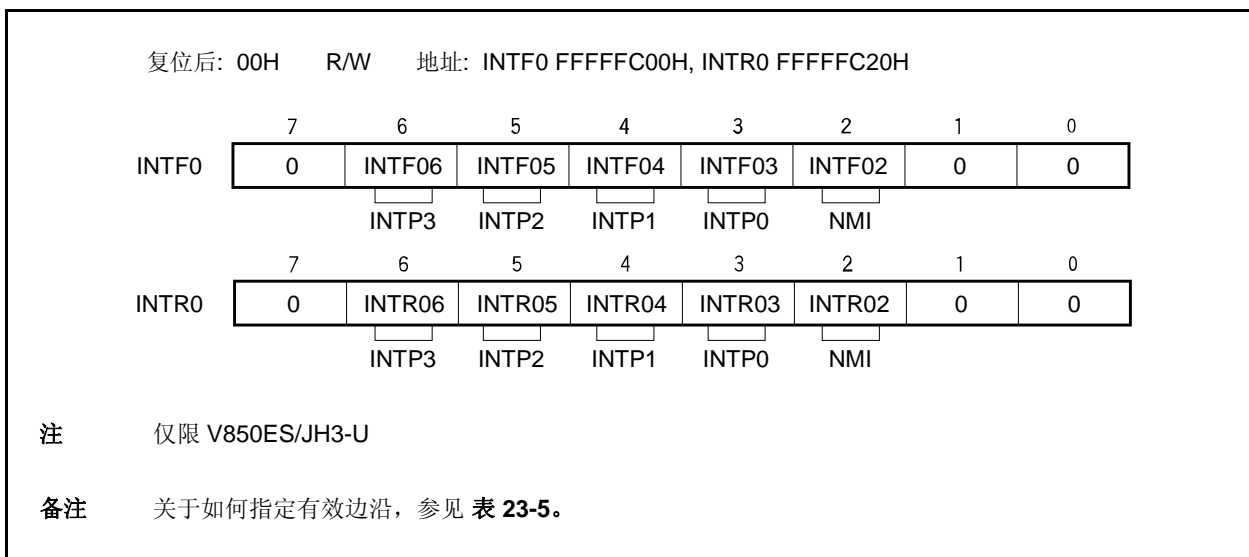


表 23-5. 有效边沿设定

INTF0n	INTR0n	有效边沿设定 (n=0 至 5)
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升和下降双边沿

注意事项 当这些寄存器不用于设置 NMI 或 INTP00 至 INTP04 引脚时，请确保将 INTF0n 和 INTR0n 位清为 00。

备注 n=0, 1: INTP00 至 INTP01 引脚的控制
 n=2: NMI 引脚的控制
 n=3 至 5: INTP02 至 INTP04 引脚的控制

(2) 外部中断下降沿和上升沿指定寄存器 2 (INTF2, INTR2) (仅限 V850ES/JH3-U)

INTF2 和 INTR2 寄存器是 8 位寄存器，用于指定外部中断引脚 (INTP05 和 INTP06) 的下降沿和上升沿检测。这些寄存器可以按字节或按位进行读写。系统复位后，这些寄存器被设置为 00H。

注意事项 当引脚功能由外部中断功能 (复用功能) 改变为端口功能时，可能会检测到一个边沿。所以，将 INTF2n 和 INTR2n 位清为 00，然后再设置端口模式。

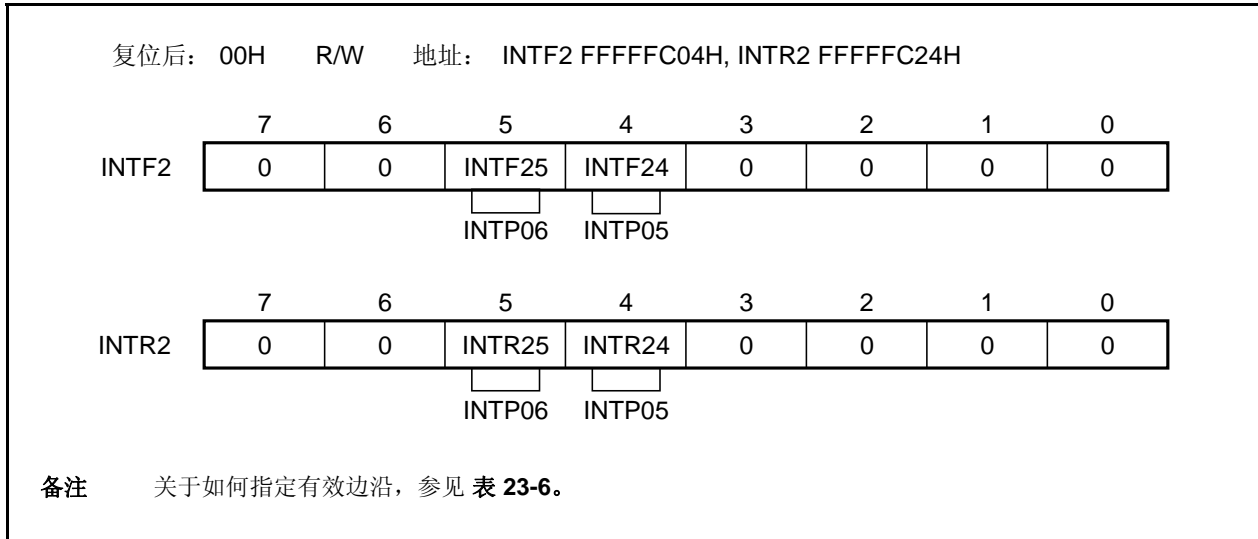


表 23-6. 有效边沿设定

INTF2n	INTR2n	有效边沿指定
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升和下降双边沿

注意事项 当这些寄存器不用于设置 INTP05 和 INTP06 引脚时，请确保将 INTF2n 和 INTR2n 位清为 00。

备注 n = 4, 5: INTP05 和 INTP06 引脚的控制

(3) 外部中断下降沿和上升沿指定寄存器 3 (INTF3, INTR3)

INTF3 和 INTR3 寄存器是 8 位寄存器，用于指定外部中断引脚 (INTP07 至 INTP09) 的下降沿和上升沿检测。

这些寄存器可以按字节或按位进行读写。

系统复位后，这些寄存器被设置为 00H。

注意事项 当引脚功能由外部中断功能（复用功能）改变为端口功能时，可能会检测到一个边沿。所以，将 **INTF3n** 和 **INTR3n** 位清为 **00**，然后再设置端口模式。

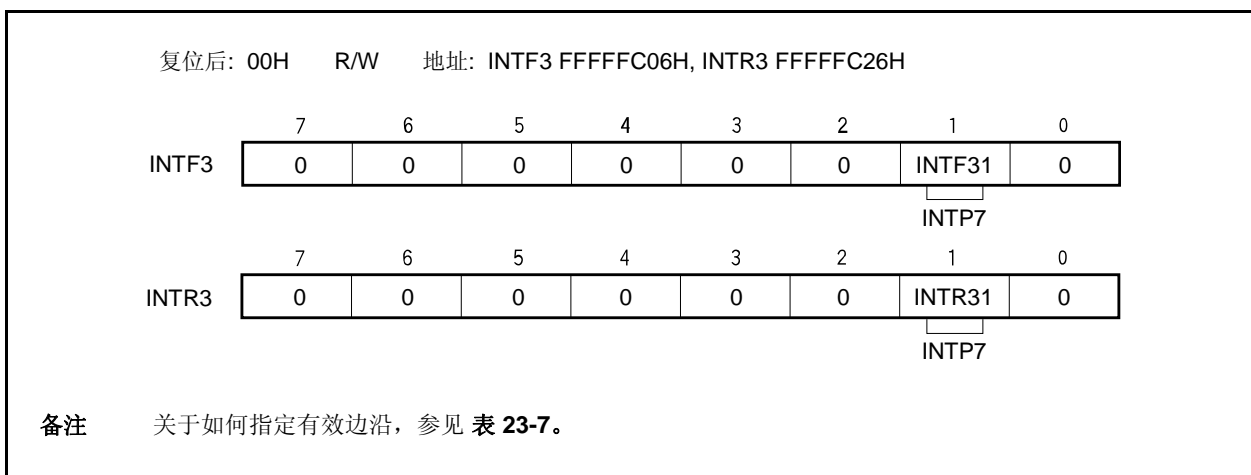


表 23-7. 有效边沿设定

INTF3n	INTR3n	有效边沿指定
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升和下降双边沿

注意事项 当这些寄存器不用于设置 INTP07 至 INTP09 引脚时，请确保将 INTF3n 和 INTR3n 位清为 00。

备注 **n = 0, 1, 4: INTP07 至 INTP09 引脚的控制**

(4) 外部中断下降沿和上升沿指定寄存器 4 (INTF4, INTR4)

INTF4 和 INTR4 寄存器是 8 位寄存器，用于指定外部中断引脚 (INTP10) 的下降沿和上升沿检测。这些寄存器可以按字节或按位进行读写。系统复位后，这些寄存器被设置为 00H。

注意事项 当引脚功能由外部中断功能（复用功能）改变为端口功能时，可能会检测到一个边沿。所以，将 INTF42 和 INTR42 位清为 00，然后再设置端口模式。

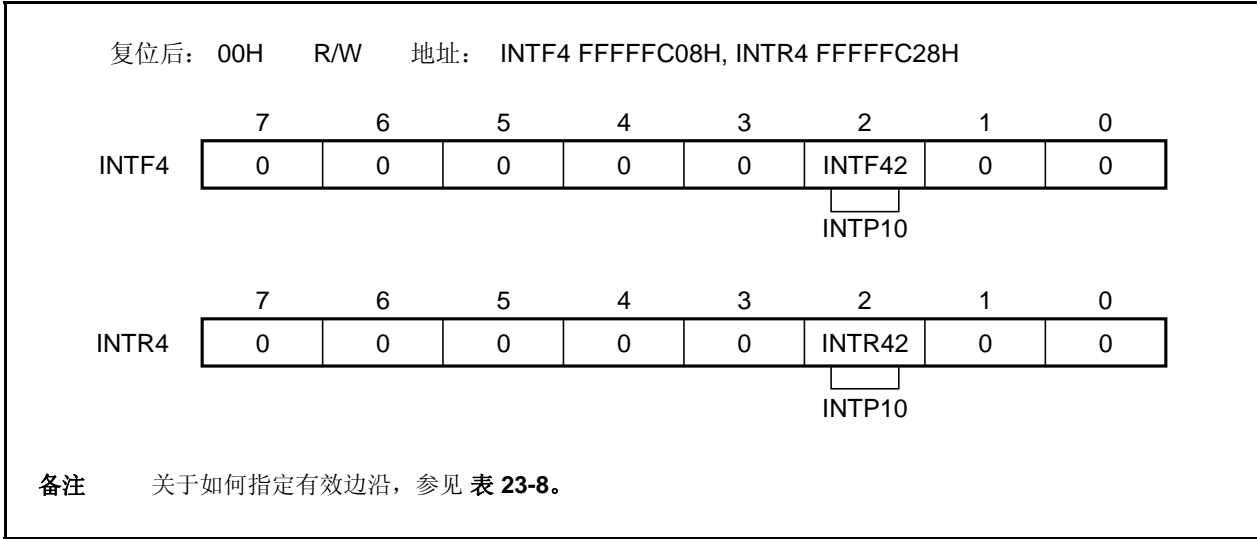


表 23-8. 有效边沿设定

INTF42	INTR42	有效边沿指定
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升和下降双边沿

注意事项 当这些寄存器不用于设置 INTP10 引脚时，请确保将 INTF42 和 INTR42 位清为 00。

(5) 外部中断下降沿和上升沿指定寄存器 5 (INTF5, INTR5) (仅限 V850ES/JH3-U)

INTF5 和 INTR5 寄存器是 8 位寄存器，用于指定外部中断引脚 (INTP05) 的下降沿和上升沿检测。

这些寄存器可以按字节或按位进行读写。

系统复位后，这些寄存器被设置为 00H。

注意事项 当引脚功能由外部中断功能 (复用功能) 改变为端口功能时，可能会检测到一个边沿。所以，将 **INTF56** 和 **INTR56** 位清为 **00**，然后再设置端口模式。

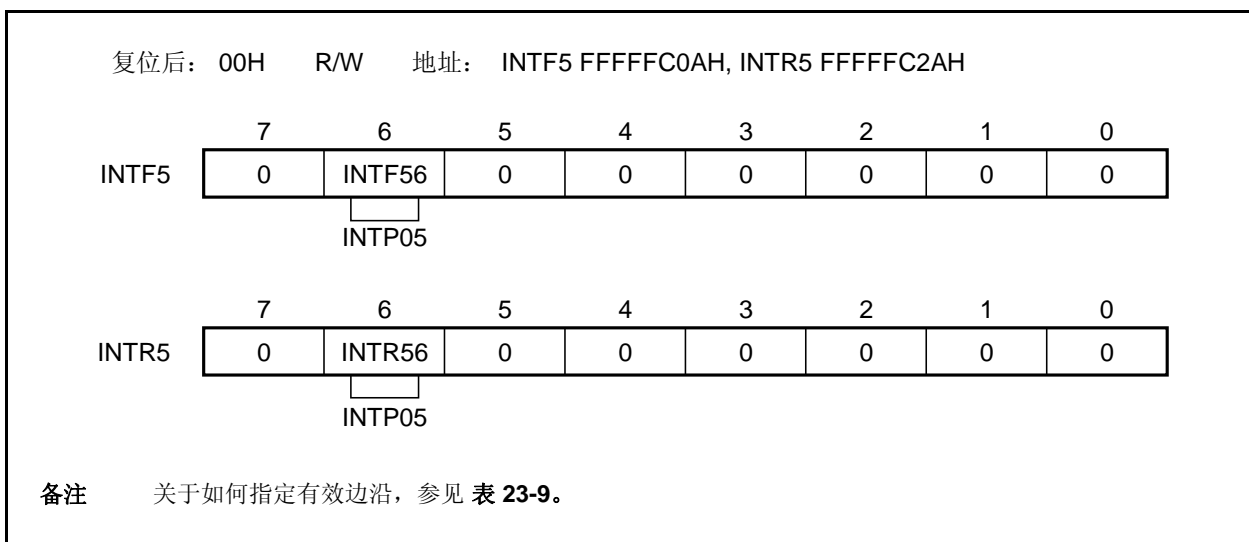


表 23-9. 有效边沿设定

INTF56	INTR56	有效边沿指定
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升和下降双边沿

注意事项 当这些寄存器不用于设置 INTP05 引脚时，请确保将 INTF56 和 INTR56 位清为 00。

(6) 外部中断下降沿和上升沿指定寄存器 9H (INTF9H, INTR9H)

INTF9H 和 INTR9H 寄存器是 8 位寄存器，用于指定外部中断引脚 (INTP11 至 INTP18) 的下降沿和上升沿检测。

这些寄存器可以按字节或按位进行读写。

系统复位后，这些寄存器被设置为 00H。

注意事项 当引脚功能由外部中断功能 (复用功能) 改变为端口功能，可能会检测到一个边沿。所以，将 INTF9n 和 INTR9n 位清为 00，然后再设置端口模式。

复位后: 00H R/W 地址: INTF9 FFFFC12H, INTF9H FFFFC12H, INTF9L FFFFC13H

	15	14	13	12	11	10	9	8
INTF9 (INTF9H ^注)	INTF915	INTF914	INTF913	0	INTF911	INTF910	INTF99	INTF98
	INTP18	INTP17	INTP16		INTP15	INTP14	INTP13	INTP12
(INTF9L)	7	6	5	4	3	2	1	0
	0	INTF96	0	0	0	0	0	0
		INTP11						

复位后: 00H R/W 地址: INTR9 FFFFC32H, INTR9H FFFFC32H, INTR9L FFFFC33H

	15	14	13	12	11	10	9	8
INTR9 (INTR9H ^注)	INTR915	INTR914	INTF913	0	INTR911	INTR910	INTR99	INTR98
	INTP18	INTP17	INTP16		INTP15	INTP14	INTP13	INTP12
(INTR9L)	7	6	5	4	3	2	1	0
	0	INTR96	0	0	0	0	0	0
		INTP11						

注 若要按字节或按位读取 INTF9 至 INTR9 寄存器的第 8 位至第 15 位，需指定这些位作为 INTF9H 和 INTR9H 寄存器的第 0 位至第 7 位。

备注 关于如何指定有效边沿，参见表 23-10。

表 23-10. 有效边沿设定

INTF9n	INTR9n	有效边沿设定 (n=11 至 18)
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升和下降双边沿

注意事项 当这些寄存器不用于设置 INTP11 至 INTP18 引脚时，请确保将 INTF9n 和 INTR9n 位清为 00。

备注 n=6, 8 至 11, 13 至 15: INTP11 至 INTP18 引脚的控制

(7) 噪声消除控制寄存器 (INTNFC)

INTP02 引脚可以选择模拟噪声消除和数字噪声消除。通过使用 INTNFC 寄存器来执行噪声消除的设置。

当选择模拟噪声消除时，引脚的输入电平需要维持一段特定的时间或更久，用来检测作为边沿。

当选择数字噪声消除时，数字采样的采样时钟可从以下范围选择： $f_{xx}/64$ ， $f_{xx}/128$ ， $f_{xx}/256$ ， $f_{xx}/512$ ， $f_{xx}/1,024$ 和 f_{xt} 。执行 3 次采样。

即使选择数字噪声消除，使用 f_{xt} 作为采样时钟，也可以使用 INTP3 中断请求信号来释放 IDLE1，IDLE2 和 STOP 模式。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被设为 00H。

注意事项 在采样时钟改变后，需要 3 个采样时钟来初始化数字噪声消除器。因此，如果在采样时钟改变后，有 INTP02 有效沿在这 3 个采样时钟内输入，可能会产生一个中断请求信号。因此，当使用中断和 DMA 功能时，注意以下几点。

- 当使用中断功能时，在 3 个采样时钟过后，对中断请求标志 (PIC2.PIF2 位) 清零后再使能中断。
- 当使用 DMA 功能 (由 INTP02 启动) 时，在 3 个采样时钟过后使能 DMA。

复位后: 00H R/W 地址: FFFFF728H

	7	6	5	4	3	2	1	0
INTNFC	INTNFEN	0	0	0	0	INTNFC2	INTNFC1	INTNFC0

INTNFEN	INTP02 引脚噪声消除的设置
0	模拟噪声消除 (60 ns (典型值))
1	数字噪声消除

INTNFC2	INTNFC1	INTNFC0	数字采样时钟
0	0	0	$f_{xx}/64$
0	0	1	$f_{xx}/128$
0	1	0	$f_{xx}/256$
0	1	1	$f_{xx}/512$
1	0	0	$f_{xx}/1,024$
1	0	1	f_{xt} (副时钟)
其它			禁止设置

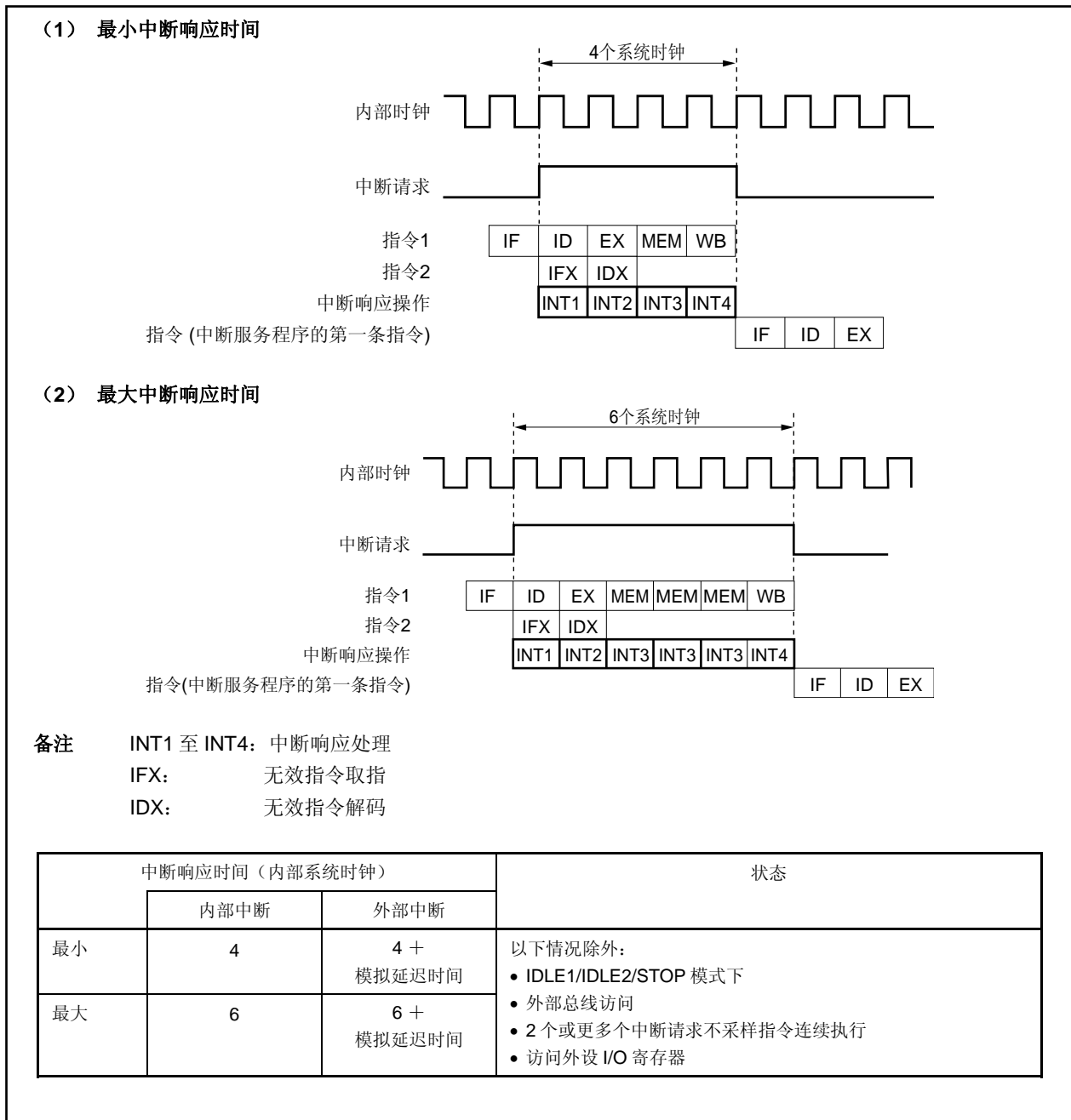
- 备注**
1. 因为采样执行了 3 次，则可靠消除噪音需要 2 个采样时钟宽度。
 2. 当噪声宽度小于 2 个采样时钟时，如果噪声与采样时钟的输入同步，就会产生一个中断请求信号。

23.7 CPU中断响应时间

除了下列情况外，CPU 的中断响应时间最小为 4 个时钟周期。若要连续输入中断请求信号，至少需要在前一中断之后 5 个时钟周期才能输入下一中断请求信号。

- IDLE1/IDLE2/STOP 模式下
- 当访问外部总线时
- 当中断请求不采样指令被连续执行时（参见 23.8 中断不被 CPU 响应期间）。
- 当访问中断控制寄存器时

图 23-15. 中断请求信号响应的流水线操作（概述）



23.8 中断不被CPU响应期间

当指令正在执行时，CPU 可以响应中断。但是，在中断请求不采样指令和下一条指令之间不会响应中断（中断挂起）。

中断请求不采样指令如下所示：

- EI 指令
- DI 指令
- LDSR reg2, 0x5 指令（对于 PSW）
- PRCMD 寄存器的存储指令
- 对下列寄存器的存储指令和 SET1, NOT1, CLR1 操作指令
 - 中断相关的寄存器：
 - 中断控制寄存器（xxICn），中断屏蔽寄存器 0 至 5（IMR0 至 IMR5）
 - 节电控制寄存器（PSC）
 - 片上调试模式寄存器（OCDM）

备注 xx: 每一个外设单元的标志名称（参见 表 23-4 中断控制寄存器（xxICn））。

 n: 外设单元编号（参见 表 23-4 中断控制寄存器（xxICn））。

23.9 注意事项

NMI 引脚复用功能作为 P02 引脚功能，而且在复位后作为正常端口引脚功能。要使能 NMI 引脚，通过 PMC0 寄存器使 NMI 引脚有效。NMI 引脚的初始设置是“无边沿检测”。使用 INTF0 和 INTR0 寄存器选择 NMI 引脚有效边沿。

第二十四章 按键中断功能

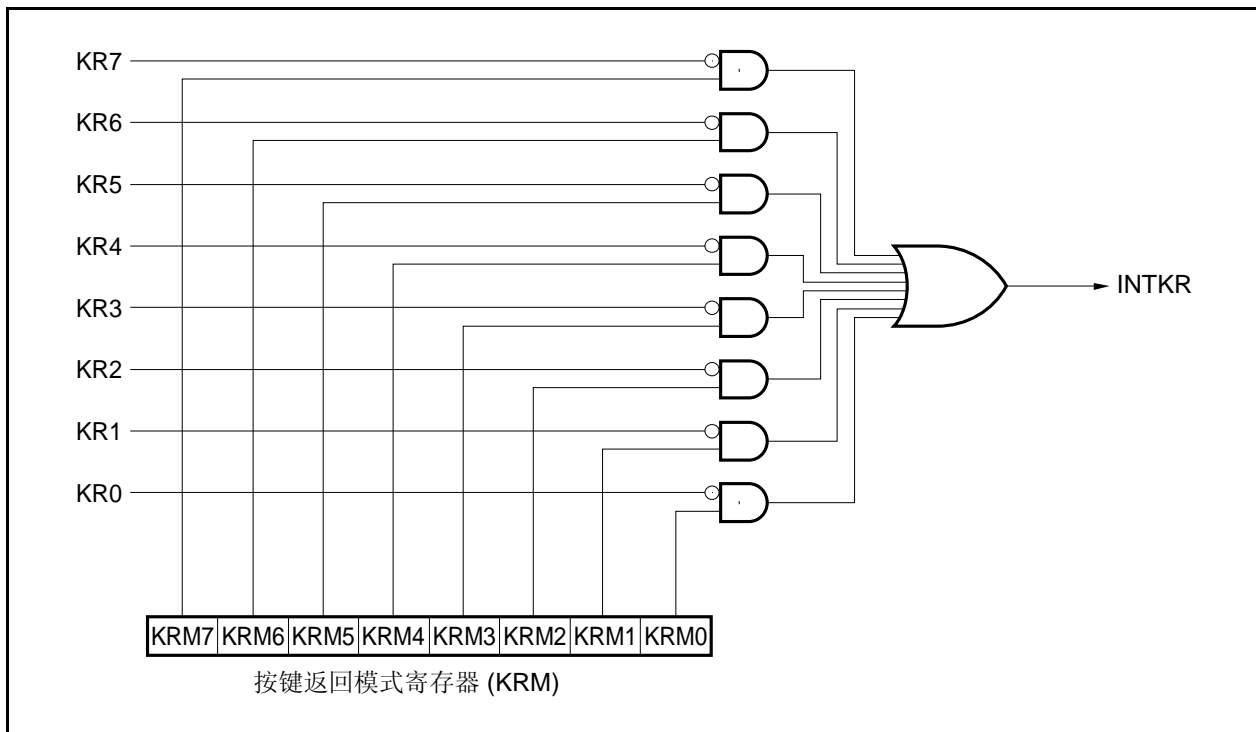
24.1 功能

通过设置 KRM 寄存器，向八个按键输入引脚（KR0 至 KR7）输入一个下降沿，可以产生一个按键中断请求信号（INTKR）。

表 24-1. 按键返回检测引脚的分配

标志	引脚说明
KRM0	用 1 位单元控制 KR0 信号
KRM1	用 1 位单元控制 KR1 信号
KRM2	用 1 位单元控制 KR2 信号
KRM3	用 1 位单元控制 KR3 信号
KRM4	用 1 位单元控制 KR4 信号
KRM5	用 1 位单元控制 KR5 信号
KRM6	用 1 位单元控制 KR6 信号
KRM7	用 1 位单元控制 KR7 信号

图 24-1. 按键返回框图



24.2 寄存器

(1) 按键返回模式寄存器 (KRM)

KRM 寄存器控制 KRM0 至 KRM7 位来使用 KR0 至 KR7 信号。

该寄存器可以按字节或按位读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H	R/W	地址: FFFFF300H					
7	6	5	4	3	2	1	0
KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0
KRMn	按键返回模式控制						
0	禁止检测按键返回信号						
1	检测按键返回信号						

注意事项 将 KRM 寄存器进行一次清零 (00H) 后，再进行重写操作。

备注 关于复用功能引脚的设置，参见表 4-20 当端口引脚用作复用功能时。

24.3 注意事项

- (1) 如果 KR0 至 KR7 中有任意某个引脚输入低电平，即使有下降沿输入其它引脚，也不会产生 INTKR 信号。
- (2) RXDC1 和 KR7 引脚不能同时使用。使用 RXDC1 引脚时，不要使用 KR7 引脚。若使用 KR7 引脚，就不要使用 RXDC1 引脚（建议将 PFC91 位设置为 1，并清除 PFCE91 位为 0）。
- (3) 如果改变 KRM 寄存器，就可能产生一个中断请求信号 (INTKR)。为防止此种情况，在禁止中断 (DI) 或屏蔽之后再改变 KRM 寄存器，然后，将中断请求标志 (KRIC.KRIF 位) 清零后，允许中断 (EI) 或取消屏蔽。
- (4) 若要使用按键中断功能，需确保将端口引脚设置为按键返回引脚，然后允许 KRM 寄存器操作。若要从按键返回引脚切换至端口引脚，需要禁止 KRM 寄存器操作，然后再设置端口引脚。

第二十五章 待机功能

25.1 概述

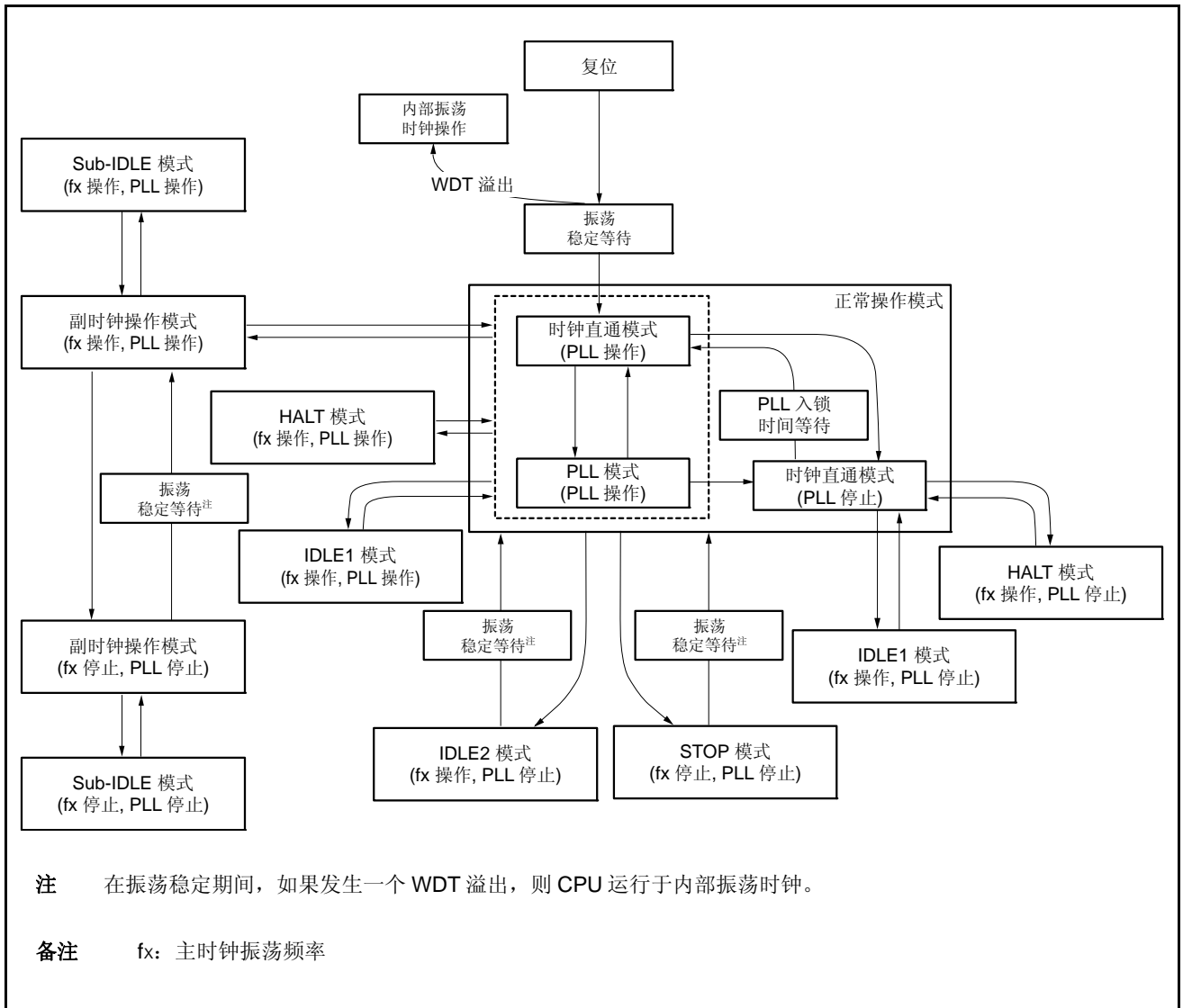
通过结合运用待机模式并为应用程序选择恰当的模式，可以有效减少系统的功耗。表 25-1 列出了可用待机模式：

表 25-1. 待机模式

模式	功能概述
HALT 模式	此模式下，仅停止 CPU 的工作时钟。
IDLE1 模式	此模式下，除振荡器，PLL 和 flash 存储器之外，停止其它所有内部电路的操作。
IDLE2 模式	此模式下，除振荡器外，停止其它所有内部电路的操作。
STOP 模式	此模式下，除副时钟振荡器外，停止其它所有内部电路的操作。
副时钟工作模式	此模式下，副时钟用作内部系统时钟。
Sub-IDLE 模式	在副时钟工作模式下，除振荡器之外，停止其它所有内部电路的操作。

注 PLL 保持模式之前的操作状态。

图 25-1. 状态转换



注 在振荡稳定期间，如果发生一个 WDT 溢出，则 CPU 运行于内部振荡时钟。

备注 fx: 主时钟振荡频率

25.2 寄存器

(1) 节电控制寄存器 (PSC)

PSC 寄存器是 8 位寄存器，用于控制待机功能。该寄存器的 STP 位用于指定 STOP 模式。该寄存器是专用寄存器，只能通过特定的序列组合来写入（参见 3.4.7 专用寄存器）。

该寄存器可以按字节或位进行读写。

系统复位后，该寄存器被设置为 00H。

复位后：00H R/W 地址：FFFFFF1FEH

	7	<6>	<5>	<4>	3	2	<1>	0
PSC	0	NMI1M	NMI0M	INTM	0	0	STP	0

NMI1M	INTWDT2信号发生时解除待机模式的控制
0	允许通过INTWDT2信号解除待机模式
1	禁止通过INTWDT2信号解除待机模式

NMI0M	通过 NMI 引脚输入控制待机模式解除
0	NMI引脚输入有效时解除待机模式
1	NMI引脚输入无效时解除待机模式

INTM	通过可屏蔽中断请求信号控制待机模式的解除
0	可屏蔽中断请求信号有效时解除待机模式
1	可屏蔽中断请求信号无效时解除待机模式

STP	待机模式 ^注 设置
0	正常模式
1	待机模式

注 可以由 STP 位设置的待机模式有 IDLE1, IDLE2, STOP 或 sub-IDLE 模式。

- 注意事项
1. 设置 IDLE1, IDLE2, STOP, 或 sub-IDLE 模式之前, 设置 PSMR.PSM1 和 PSMR.PSM0 位, 然后再设置 STP 位。
 2. 当 HALT 模式解除后, NMI1M, NMI0M, 和 INTM 位的设置无效。
 3. 如果在 NMI1M, NMI0M, 或 INTM 位设为 1 的同时把 STP 位也设为 1, 则对 NMI1M, NMI0M, 或 INTM 位的设置无效。如果在设置 IDLE1/IDLE2/STOP 模式时有未屏蔽中断请求信号处于挂起状态, 则应设置中断请求信号对应的位 (NMI1M, NMI0M, 或 INTM) 为 1, 然后再设置 STP 位为 1。

(2) 节电模式寄存器 (PSMR)

PSMR 寄存器是 8 位寄存器，用于控制节电模式下的操作状态及时钟操作。

该寄存器可按字节或按位进行读写。

复位输入将该寄存器设置为 00H。

复位后: 00H R/W 地址: FFFFF820H

	7	6	5	4	3	2	<1>	<0>
PSMR	0	0	0	0	0	0	PSM1	PSM0

PSM1	PSM0	软件待机模式时的操作说明
0	0	IDLE1, sub-IDLE 模式
0	1	STOP 模式
1	0	IDLE2, sub-IDLE 模式
1	1	STOP 模式

注意事项 1. 请确保将第 2 位至第 7 位清为“0”。

2. 仅当 PSC.STP 位为 1 时，PSM0 和 PSM1 位才有效。

备注

IDLE1: 此模式下，除振荡器和其它一些电路（flash 存储器和 PLL）之外，停止所有的操作。在解除 IDLE1 模式后，无需经过振荡稳定保证时间，就可以恢复正常操作模式。这和 HALT 模式一样。

IDLE2: 此模式下，除振荡器之外，所有的操作停止。解除 IDLE2 模式后，在经过 OSTS 寄存器指定的振荡稳定时间后，恢复正常操作模式。

STOP: 此模式下，除副时钟振荡器之外，所有的操作停止。解除 STOP 模式后，在经过 OSTS 寄存器指定的振荡稳定时间后，恢复正常操作模式。

Sub-IDLE: 此模式下，除振荡器之外，暂停所有的操作。在中断请求信号解除 IDLE 模式后，经过 12 个副时钟周期的保证时间后，恢复副时钟操作模式。

(3) 振荡稳定时间选择寄存器 (OSTS)

在解除 STOP 模式之后至振荡稳定之前需要等待的时间，或在解除 IDLE2 模式之后至内部 flash 存储器稳定之前需要等待的时间，由 OSTS 寄存器控制。

该寄存器可以按字节进行读写。

复位输入将寄存器设置为 06H。

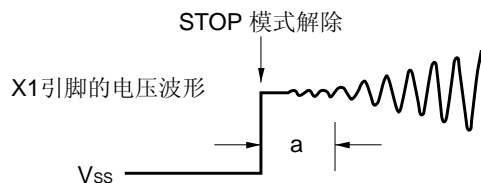
复位后：06H R/W 地址：FFFFF6C0H

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

OSTS2	OSTS1	OSTS0	振荡稳定时间/建立时间的选择 ^注	fx	
				3 MHz	6 MHz
				0	0
0	0	1	$2^{11}/f_x$	0.683 ms	0.341 ms
0	1	0	$2^{12}/f_x$	1.365 ms	0.683 ms
0	1	1	$2^{13}/f_x$	2.730 ms	1.365 ms
1	0	0	$2^{14}/f_x$	5.461 ms	2.731 ms
1	0	1	$2^{15}/f_x$	10.923 ms	5.461 ms
1	1	0	$2^{16}/f_x$	21.85 ms	10.92 ms
1	1	1	禁止设置		

注 当解除 STOP 模式和 IDLE2 模式时，需要不同的振荡稳定时间和建立时间。

注意事项 1. 不管是因为复位输入还是因为中断请求信号解除了 STOP 模式，在解除 STOP 模式之后的等待时间不包括时钟开始振荡之前（下图的“a”）的那段时间。



2. 请确保将第 3 位至第 7 位清为“0”。
3. 复位解除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 = 主时钟振荡器频率

25.3 HALT模式

25.3.1 设置和操作状态

在正常操作模式下，执行一条专用指令（HALT），可以设置为 HALT 模式。

在 HALT 模式下，时钟振荡器继续工作。只是停止 CPU 的时钟供给；其它片上外设功能的时钟不受影响。

结果，程序停止执行，内部 RAM 保持设置为 HALT 模式之前的内容。独立于 CPU 指令处理的片上外设功能继续工作。

表 25-3 显示了 HALT 模式下的操作状态。

通过 HALT 模式和正常操作模式的间歇结合操作，可以减少系统的平均电流消耗。

注意事项 1. 在 HALT 指令后，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信号被挂起，则状态切换至 HALT 模式，但是 HALT 模式立即被该中断请求解除。

25.3.2 解除HALT模式

在 HALT 模式下，有多种方式可以解除 HALT 模式。包括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（INTP00 至 INTP18 引脚输入）、来自 HALT 模式下可操作外设的未屏蔽的内部中断请求信号，或复位信号（由 RESET 引脚输入的复位、WDT2RES 信号，低电压检测器（LVI），或时钟监视器（CLM））。

HALT 模式释放后，恢复为正常操作模式。

(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解除 HALT 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号，可以解除 HALT 模式，且与中断请求信号的优先级无关。但是，如果在一个中断服务程序中设置 HALT 模式，则之后发生的中断请求信号的服务如下：

- (a) 如果发出的中断请求信号优先级低于或等于当前正在服务的中断请求，则仅解除 HALT 模式，并且对该中断请求信号不作响应。该中断请求信号被保持。
- (b) 如果发出的中断请求信号优先级高于当前正在服务的中断请求（包括不可屏蔽中断请求信号），则 HALT 模式解除，且响应该中断请求信号。

表 25-2. 用中断请求信号解除 HALT 模式后的操作

解除源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	执行对句柄地址的跳转。	
可屏蔽中断请求信号	执行对句柄地址的跳转或执行下一条指令。	执行下一条指令。

(2) 通过复位解除 HALT 模式

执行的操作与正常的复位操作相同。

表 25-3. HALT 模式下的操作状态

项目		HALT 模式的设置	运行状态	
			不使用副时钟时	使用副时钟时
主时钟振荡器(f _x)			振荡使能	
副时钟振荡器(f _{xT})			-	振荡使能
内部振荡器(f _R)			振荡使能	
PLL			可操作	
CPU			停止操作	
DMA 控制器			可操作	
中断控制器			可操作	
定时器	TAA0 至 TAA5		可操作	
	TAB0, TAB1		可操作	
	TMM0 至 TMM3		当选择 f _{xT} 之外的时钟为计数时钟时，可操作	可操作
	TMT0		可操作	
实时计数器 (RTC)			当 f _x (BRG 分频) 选择为计数时钟时，可操作	可操作
看门狗定时器 (WDT2)			当选择 f _{xT} 之外的时钟为计数时钟时，可操作	可操作
串行接口	CSIF0 至 CSIF4		可操作	
	I ² C00 至 I ² C02		可操作	
	UARTC0 至 UARTC4		可操作	
A/D 转换器			可操作	
D/A 转换器			可操作	
实时输出功能 (RTO)			可操作	
按键中断功能 (KR)			可操作	
CRC 操作电路			可操作 (因为 CPU 被停止，没有数据输入到 CRCIN 寄存器)	
外部总线接口			参见 第五章 总线控制功能	
端口功能			保持设置为 HALT 模式前的状态。	
内部数据			CPU 寄存器、状态、数据及其它内部数据，例如内部 RAM 的内容，被保持为设置 HALT 模式之前的状态。	
USB 功能			可操作	
USB 主机			可操作	

25.4 IDLE1 模式

25.4.1 设置和操作状态

在正常操作模式下，通过将 PSMR.PSM1 和 PSMR.PSM0 位清 00，并且将 PSC.STP 设置为 1，可以设置 IDLE1 模式。

在 IDLE1 模式下，时钟振荡器、PLL、以及 flash 存储器继续操作，但停止 CPU 和其它片上外设功能的时钟供给。

结果，程序停止执行，且内部 RAM 保持与设置为 IDLE1 模式前的内容。CPU 及其它片上外设功能停止工作。但是，可以在副时钟或外部时钟下工作的片上外设功能继续工作。

表 25-5 显示了 IDLE1 模式下的操作状态。

与 HALT 模式相比，IDLE1 模式可以减少更多功耗，因为停止了片上外设功能的运行。主时钟振荡器不停止，所以在解除 IDLE1 模式后，无需等待振荡稳定时间就可以恢复到正常操作模式，HALT 模式解除时也是如此。

- 注意事项**
1. 为了设置 IDLE1 模式，需要将数据存储在 PSC 寄存器，在存储指令之后，应该插入五个或更多 NOP 指令。
 2. 如果在设置 IDLE1 模式时，有一个未屏蔽的中断请求信号被挂起，则 IDLE1 模式立即被这个等待响应的中断请求解除。

25.4.2 解除IDLE1 模式

在 IDLE1 模式下，有多种方式可以解除 IDLE1 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽外部中断请求信号（INTP00 至 INTP18 引脚输入）、来自 IDLE1 模式下可操作的外设功能的未屏蔽的内部中断请求信号，或复位信号（由 $\overline{\text{RESET}}$ 引脚输入复位、WDT2RES 信号，低电压检测器（LVI），或时钟监视器（CLM））。

IDLE1 模式释放后，恢复为正常操作模式。

(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来解除 IDLE1 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号，可以解除 IDLE1 模式，且与中断请求信号的优先级无关。如果在一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的服务如下：

- (a) 如果发出的中断请求信号优先级低于当前正在服务的中断请求，则仅解除 IDLE1 模式，并且对该中断请求信号不作响应。该中断请求信号被保持。
- (b) 如果发出的中断请求信号优先级高于当前正在服务的中断请求（包括不可屏蔽中断请求信号），则解除 IDLE1 模式，且响应该中断请求信号。

注意事项 通过 PSC.NMI1M, PSC.NMI0M, 和 PSC.INTM 位设置为 1 被禁止的中断请求信号变为无效，并且 IDLE1 模式不会被解除。

表 25-4.通过中断请求信号解除 IDLE1 模式后的操作

解除源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	在确保经过预先设置的建立时间之后，执行对句柄地址的跳转。	
可屏蔽中断请求信号	在确保经过预先设置的建立时间之后，执行对句柄地址的跳转或执行下一条指令。	在确保经过预先设置的建立时间之后，执行下一条指令。

(2) 通过复位解除 IDLE1 模式

执行的操作与正常的复位操作相同。

表 25-5. IDLE1 模式下的操作状态

IDLE1 模式的设置		运行状态	
		不使用副时钟时	使用副时钟时
主时钟振荡器(fx)		振荡使能	
副时钟振荡器(fxt)		-	振荡使能
内部振荡器(fr)		振荡使能	
PLL		可操作	
CPU		停止操作	
DMA 控制器		停止操作	
中断控制器		停止操作（但是可能解除待机模式）	
定时器	TAA0 至 TAA5	停止操作	
	TAB0, TAB1	停止操作	
	TMM0 至 TMM3	当 fr/8 选择为计数时钟时，可操作	当 fr/8 或 fxt 选择为计数时钟时，可操作
	TMT0	停止操作	
实时计数器 (RTC)		当 fx (BRG 分频) 选择为计数时钟时，可操作	可操作
看门狗定时器 2		当 fr 选择为计数时钟时，可操作	当 fr 或 fxt 选择为计数时钟时，可操作
串行接口	CSIF0 至 CSIF4	当 \overline{SCKFn} 输入时钟被选为计数时钟时，可操作 (n=0 至 4)	
	I ² C00 至 I ² C02	停止操作	
	UARTC0 至 UARTC4	停止操作（但是当选择 ASCKC0 引脚时钟时，UARTC0 可操作）	
A/D 转换器		保持操作（转换结果保持） [※]	
D/A 转换器		保持操作（输出保持 [※] ）	
实时输出功能 (RTO)		停止操作（输出保持）	
按键中断功能 (KR)		可操作	
CRC 操作电路		停止操作	
外部总线接口		参见 第五章 总线控制功能	
端口功能		保持设置为 IDLE1 模式前的状态	
内部数据		CPU 寄存器、状态位、数据及其它内部数据，例如内部 RAM 的内容，都保持为设置 IDLE1 模式之前的状态。	

注 为了降低功率损耗，在切换到 IDLE1 模式之前，停止 A/D 转换器和 D/A 转换器。

25.5 IDLE2 模式

25.5.1 设置和操作状态

在正常操作模式下，通过将 PSMR.PSM1 和 PSMR.PSM0 位设置为 10，并且将 PSC.STP 设置为 1，可以设置为 IDLE2 模式。

在 IDLE2 模式下，时钟振荡器继续操作，但停止对 CPU、PLL、flash 存储器及其它片上外设功能的时钟供给。

结果，程序停止执行，且内部 RAM 保持与设置为 IDLE2 模式前的内容。CPU、PLL 以及其它片上外设功能停止工作。但是，可以在副时钟或外部时钟下工作的片上外设功能继续工作。

表 25-7 显示了 IDLE2 模式下的操作状态。

与 IDLE1 模式相比，IDLE2 模式可以减少更多功耗，因为停止了片上外设功能、PLL 和 flash 存储器的运行。但是，因为 PLL 和 flash 存储器停止，当解除 IDLE2 模式时，PLL 和 flash 存储器需要一段建立时间。

- 注意事项**
1. 为了设置 IDLE2 模式，需要将数据存储到 PSC 寄存器，在存储指令之后，应该插入五个或更多 NOP 指令。
 2. 如果在设置 IDLE2 模式时，有一个未屏蔽的中断请求信号被挂起，则 IDLE2 模式立即被这个等待响应的中断请求解除。

25.5.2 解除IDLE2 模式

在 IDLE2 模式下，有多种方式可以解除 IDLE2 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽外部中断请求信号（INTP00 至 INTP18 引脚输入）、来自 IDLE2 模式下可操作的外设功能的未屏蔽的内部中断请求信号，或复位信号（由 $\overline{\text{RESET}}$ 引脚输入复位、WDT2RES 信号，低电压检测器（LVI），或时钟监视器（CLM））。PLL 返回到设置为 IDLE2 模式前的操作状态。

IDLE2 模式释放后，恢复为正常操作模式。

(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来解除 IDLE2 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号，可以解除 IDLE2 模式，且与中断请求信号的优先级无关。如果在一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的服务如下：

- (a) 如果发出的中断请求信号优先级低于当前正在服务的中断请求，则仅解除 IDLE2 模式，并且对该中断请求信号不作响应。该中断请求信号被保持。
- (b) 如果发出的中断请求信号优先级高于当前正在服务的中断请求（包括不可屏蔽中断请求信号），则解除 IDLE2 模式，且响应该中断请求信号。

注意事项 通过 PSC.NMI1M，PSC.NMI0M，和 PSC.INTM 位设置为 1 被禁止的中断请求信号变为无效，并且 IDLE2 模式不会被解除。

表 25-6. 通过中断请求信号解除 IDLE2 模式后的操作

解除源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	在确保的建立时间过后，执行对句柄地址的跳转。	
可屏蔽中断请求信号	在确保的建立时间之后，执行对句柄地址的跳转或执行下一条指令。	在确保的建立时间之后执行下一条指令。

(2) 通过复位解除 IDLE2 模式

执行的操作与正常的复位操作相同。

表 25-7. IDLE2 模式下的操作状态

IDLE2 模式的设置		运行状态	
		不使用副时钟时	使用副时钟时
主时钟振荡器(f _x)		振荡使能	
副时钟振荡器(f _{xT})		-	振荡使能
内部振荡器(f _R)		振荡使能	
PLL		停止操作	
CPU		停止操作	
DMA 控制器		停止操作	
中断控制器		停止操作（但是可能解除待机模式）	
定时器	TAA0 至 TAA5	停止操作	
	TAB0, TAB1	停止操作	
	TMM0 至 TMM3	当 f _R /8 选择为计数时钟时，可操作	当 f _R /8 或 f _{xT} 选择为计数时钟时，可操作
	TMT0	停止操作	
实时计数器 (RTC)		当 f _x (BRG 分频) 选择为计数时钟时，可操作	可操作
看门狗定时器 2		当 f _R 选择为计数时钟时，可操作	当 f _R 或 f _{xT} 选择为计数时钟时，可操作
串行接口	CSIF0 至 CSIF4	当 $\overline{\text{SCKFn}}$ 输入时钟被选为计数时钟时，可操作 (n=0 至 4)	
	I ² C00 至 I ² C02	停止操作	
	UARTC0 至 UARTC4	停止操作（但是当选择 ASCKC0 引脚时钟时，UARTC0 可操作）	
A/D 转换器		保持操作（转换结果保持） [※]	
D/A 转换器		保持操作（输出保持 [※] ）	
实时输出功能 (RTO)		停止操作（输出保持）	
按键中断功能 (KR)		可操作	
CRC 操作电路		停止操作	
外部总线接口		参见 第五章 总线控制功能	
端口功能		保持设置为 IDLE2 模式前的状态	
内部数据		CPU 寄存器、状态位、数据及其它内部数据，例如内部 RAM 的内容，都保持为设置 IDLE2 模式之前的状态。	
USB 功能		可操作	
USB 主机		可操作	

注 为了降低功率损耗，在切换到 IDLE2 模式之前，停止 A/D 转换器和 D/A 转换器。

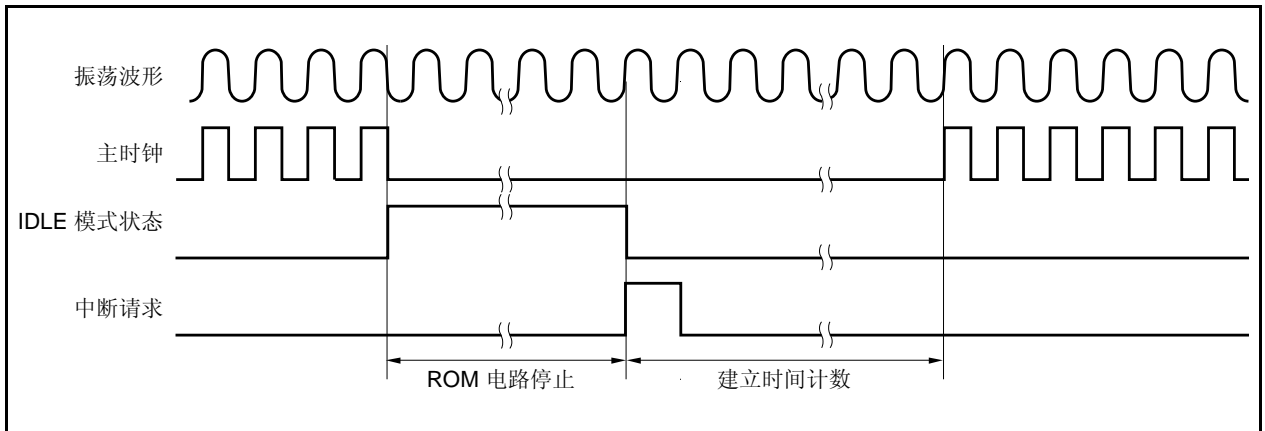
25.5.3 解除IDLE2 模式时确保建立时间

在解除 IDLE2 模式之后，要确保 flash 存储器的建立时间，这是因为在设置 IDLE2 模式后，除了主时钟振荡器之外的模块都停止操作。

(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来解除 IDLE2 模式

设置 OSTS 寄存器，确保指定的建立时间。

当产生释放源时，依据 OSTS 寄存器的设置，专用内部定时器开始计数。当其溢出时，恢复正常工作模式。



(2) 通过复位解除 ($\overline{\text{RESET}}$ 引脚输入, WDT2RES 产生)

此操作与正常复位的操作相同。

振荡稳定时间是 OSTS 寄存器的初始值, $2^{16}/f_x$ 。

25.6 STOP模式

25.6.1 设置和操作状态

在正常操作模式下，将 PSMR.PSM1 和 PSMR.PSM0 位设为 01 或 11，并将 PSC.STP 设为 1，可以设置为 STOP 模式。

在 STOP 模式下，副时钟振荡器继续工作，但主时钟振荡器停止。停止对 CPU 和片上外设功能的时钟供给。

结果，程序停止执行，且内部 RAM 保持与设置为 STOP 模式前的内容。通过副时钟振荡器或外部时钟提供时钟的片上外设功能继续工作。

表 25-9 显示了在 STOP 模式下的操作状态。

因为 STOP 模式使主时钟振荡器的运行停止，所以可将功耗降低到低于 IDLE2 模式的水平。若不使用副时钟振荡器、内部振荡器和外部时钟，功耗达到最小化，仅有漏电流通过。

注意事项 1，为了设置 STOP 模式，需要将数据存储到 PSC 寄存器，在存储指令之后，应该插入五个或更多 NOP 指令。

2。如果在设置 STOP 模式时，有一个未屏蔽的中断请求信号被挂起，则 STOP 模式立即被这个等待响应的中断请求解除。

25.6.2 解除STOP模式

有多种方式可以解除 STOP 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽外部中断请求信号（INTP00 至 INTP18 引脚输入）、来自 STOP 模式下可操作的外设功能的未屏蔽的内部中断请求信号，或复位信号（由 RESET 引脚输入复位、WDT2RES 信号，低电压检测器（LVI））。

在 STOP 模式解除后，在保证的振荡稳定时间后恢复为正常操作模式。

(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来解除 STOP 模式

通过不可屏蔽中断请求信号或未屏蔽中断请求信号，都可以解除 STOP 模式，且与中断请求信号的优先级无关。但是，如果在一个中断服务程序中设置 STOP 模式，则之后发出的中断请求信号的服务如下：

- (a) 如果发出的中断请求信号优先级低于当前正在服务的中断请求，则解除 STOP 模式，并且对该中断请求信号不作响应。该中断请求信号被保持。
- (b) 如果发出的中断请求信号优先级高于当前正在服务的中断请求（包括不可屏蔽中断请求信号），则解除 STOP 模式，且响应该中断请求信号。

注意事项 通过 PSC.NMI1M，PSC.NMI0M，和 PSC.INTM 位设置为 1 被禁止的中断请求信号变为无效，并且 STOP 模式不会被解除。

表 25-8. 通过中断请求信号解除 STOP 模式的操作

解除源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	在确保经过预先设置的建立时间之后，执行对句柄地址的跳转。	
可屏蔽中断请求信号	在确保经过预先设置的建立时间之后，执行对句柄地址的跳转或执行下一条指令。	在确保经过预先设置的建立时间之后，执行下一条指令。

(2) 通过复位解除 STOP 模式

执行的操作与正常的复位操作相同。

表 25-9. STOP 模式下的操作状态

STOP 模式的设置		运行状态	
		不使用副时钟时	使用副时钟时
主时钟振荡器(f_x)		停止振荡	
副时钟振荡器(f_{XT})		-	振荡使能
内部振荡器(f_R)		振荡使能	
PLL		停止操作	
CPU		停止操作	
DMA 控制器		停止操作	
中断控制器		停止操作	
定时器	TAA0 至 TAA5	停止操作	
	TAB0, TAB1	停止操作	
	TMM0 至 TMM3	当 $f_R/8$ 选择为计数时钟时, 可操作	当 $f_R/8$ 或 f_{XT} 选择为计数时钟时, 可操作
	TMT0	停止操作	
实时计数器 (RTC)		停止操作	当 f_{XT} 选择为计数时钟时, 可操作
看门狗定时器 2		当 f_R 选择为计数时钟时, 可操作	当 f_R 或 f_{XT} 选择为计数时钟时, 可操作
串行接口	CSIF0 至 CSIF4	当 \overline{SCKFn} 输入时钟被选为计数时钟时, 可操作 ($n=0$ 至 4)	
	I ² C00 至 I ² C02	停止操作	
	UARTC0 至 UARTC4	停止操作 (但是当选择 ASCKC0 引脚时钟时, UARTC0 可操作)	
A/D 转换器		停止操作 (转换结果不确定) ^{1, 2}	
D/A 转换器		停止操作 ^{3, 4} (高阻抗输出)	
实时输出功能 (RTO)		停止操作 (输出保持)	
按键中断功能 (KR)		可操作	
CRC 操作电路		停止操作	
外部总线接口		参见 第五章 总线控制功能	
端口功能		保持设置为 STOP 模式前的状态	
内部数据		CPU 寄存器、状态位、数据及其它内部数据, 例如内部 RAM 的内容, 都保持为设置 STOP 模式之前的状态。	
USB 功能		停止操作	
USB 主机		停止操作	

- 注
- 如果在 A/D 转换器操作时设置 STOP 模式, 则 A/D 转换器自动停止, 并且在 STOP 模式释放后再次开始操作。但是这种情况下, STOP 模式释放后的 A/D 转换结果无效。STOP 模式设置前的所有 A/D 转换值都无效。
 - 即使在 A/D 转换器操作时设置 STOP 模式, 功耗也同样减少, 和设置 STOP 模式前停止 A/D 转换器的功耗相等。
 - 如果在 D/A 转换器操作时设置 STOP 模式, 那么 D/A 转换器自动停止并且引脚状态变为高阻。STOP 模式解除后, 恢复 D/A 转换, 设定的时间过后, 返回到设置 STOP 模式之前的输出电平状态。
 - 即使在 D/A 转换器操作时设置 STOP 模式, 功耗也同样减少, 和设置 STOP 模式前停止 D/A 转换器的功耗相等。

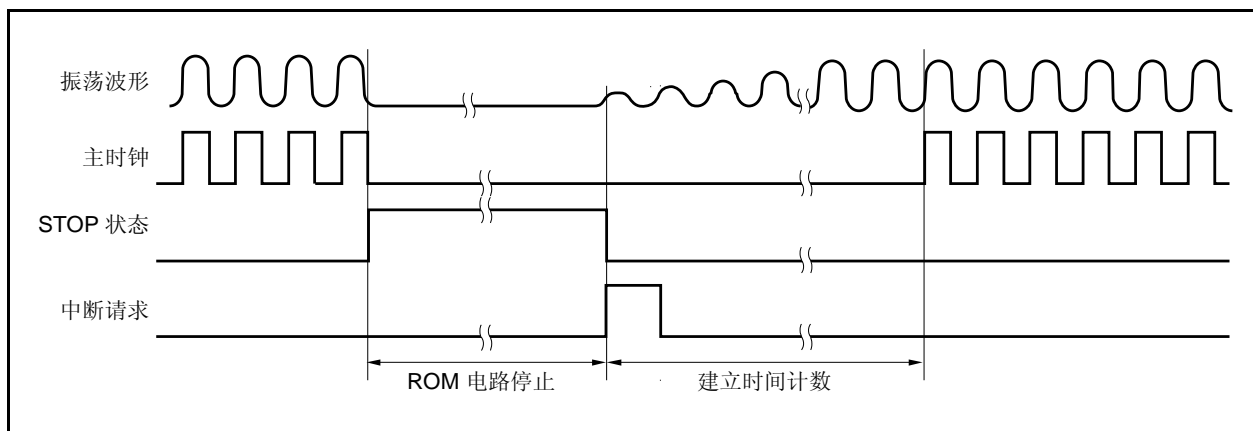
25.6.3 解除STOP模式时保障振荡稳定时间

为了主时钟振荡器，在解除 STOP 模式后需要一段时间来确保主时钟振荡器的振荡稳定，因为在设置 STOP 模式之后，主时钟振荡器的操作停止。

(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来解除 STOP 模式

设置 OSTS 寄存器，确保振荡稳定时间。

当产生释放源时，依据 OSTS 寄存器的设置，专用内部定时器开始计数。当其溢出时，恢复正常工作模式。



(2) 通过复位解除

执行的操作与正常的复位操作相同。

振荡稳定时间是 OSTS 寄存器的初始值， $2^{16}/f_x$ 。

25.7 副时钟操作模式

25.7.1 设置和操作状态

在正常操作模式下，把 PCC.CK3 位设置为 1，可以设置为副时钟工作模式。

当设置为副时钟工作模式时，内部系统时钟从主时钟变为副时钟。使用 PCC.CLS 位来检查时钟是否已经转换。

当 PCC.MCK 位设置为 1 时，主时钟振荡器停止运行。结果，系统工作于副时钟。

在副时钟工作模式下，功耗可以降至低于正常操作模式的级别，因为副时钟用作内部系统时钟。此外，停止主时钟振荡器的运行，功耗可进一步降低至 STOP 模式的级别。

表 25-10 显示了在副时钟操作模式下的操作状态。

- 注意事项**
- 在对 CK3 位进行操作时，不要改变 PCC.CK2 至 PCC.CK0 位的设定值（建议用位操作指令对此位进行操作）。有关 PCC 寄存器的详情，参见 6.3（1）处理器时钟控制寄存器（PCC）。
 - 若下列条件不满足，则改变 CK2 至 CK0 位使条件能够满足，并设置为副时钟工作模式。
内部系统时钟 (f_{CLK}) > 副时钟 (f_{XT}=32.768kHz) × 4

备注 内部系统时钟 (f_{CLK})： 根据 CK2 至 CK0 位的设置，由主时钟 (f_{XX}) 产生的时钟。

25.7.2 解除副时钟操作模式

当 CK3 位被设置为 0 时，通过复位信号（RESET 引脚的输入、WDT2RES 信号、低电压检测器 (LVI) 或时钟监视器 (CLM) 复位）来解除副时钟操作模式。

如果主时钟停止 (MCK 位 = 1)，将 MCK 位置为 1，用软件保证主时钟的振荡稳定时间，并将 CK3 位清 0。

当副时钟工作模式解除时，恢复为正常操作模式。

- 注意事项** 在操作 CK3 位时，不要改变 CK2 至 CK0 位的设置值（建议用位操作指令对此位进行操作）。有关 PCC 寄存器的详情，参见 6.3（1）处理器时钟控制寄存器（PCC）。

表 25-10. 副时钟工作模式的操作状态

副时钟工作模式的设置		运行状态	
		主时钟振荡时	主时钟停止时
副时钟振荡器 (f _{XT})		振荡使能	
内部振荡器 (f _R)		振荡使能	
PLL		可操作	停止操作 ^注
CPU		可操作	
DMA 控制器		可操作	
中断控制器		可操作	
定时器	TAA0 至 TAA5	可操作	停止操作
	TAB0, TAB1	可操作	停止操作
	TMM0 至 TMM3	可操作	当 f _R /8 或 f _{XT} 选择为计数时钟时, 可操作
	TMT0		
实时计数器 (RTC)		可操作	当 f _{XT} 选择为计数时钟时, 可操作
看门狗定时器 (WDT2)		可操作	当 f _R 或 f _{XT} 选择为计数时钟时, 可操作
串行接口	CSIF0 至 CSIF4	可操作	当 \overline{SCKFn} 输入时钟被选为计数时钟时, 可操作 (n=0 至 4)
	I ² C00 至 I ² C02	可操作	停止操作
	UARTC0 至 UARTC4	可操作	停止操作 (但是当选择 ASCKC0 引脚时钟时, UARTC0 可操作)
A/D 转换器		可操作	停止操作
D/A 转换器		可操作	
实时输出功能 (RTO)		可操作	停止操作 (输出保持)
按键中断功能 (KR)		可操作	
CRC 操作电路		可操作	
外部总线接口		参见 第五章 总线控制功能	
端口功能		可设置	
内部数据		可设置	
USB 功能		停止操作	
USB 主机		停止操作	

注 在停止主时钟前, 请确保停止锁相环 (PLLCTL.PLLON 位=0)。

注意事项 当 CPU 运行于副时钟而且主时钟振荡停止时, 在访问寄存器禁止发生等待。如果已经产生等待, 只能通过复位解除 (参见 3.4.8 (2))。

25.8 sub-IDLE模式

25.8.1 设置和操作状态

在副时钟工作模式下，将 PSMR.PSM1 和 PSMR.PSM0 位设置为 00 或 10 并将 PSC.STP 位设置为 1，可以设置为 Sub-IDLE 模式。

该模式下，时钟振荡器继续工作，但停止对 CPU、Flash 存储器及其它片上外设功能的时钟供给。

结果，程序停止执行，且内部 RAM 保持与设置为 sub-IDLE 模式前的内容。CPU 及其它片上外设功能停止。然而，运行于副时钟或外部时钟的片上外设功能继续工作。

因为 Sub-IDLE 模式停止了 CPU、flash 存储器及其它片上外设功能的操作，所以可以比副时钟工作模式降低更多功耗。如果在主时钟停止后设置为 Sub-IDLE 模式，则该模式下的电流消耗可减少至与 STOP 模式同样低的水平。

表 25-12 显示了在 sub-IDLE 模式下的操作状态。

注意事项 1. 为了设置 Sub-IDLE 模式，需要将数据存储在 PSC 寄存器，在存储指令之后，应该插入五个或更多 NOP 指令。

2. 如果在设置 sub-IDLE 模式时，有一个未屏蔽的中断请求信号被挂起，则 sub-IDLE 模式立即被这个等待响应的中断请求解除。

25.8.2 解除sub-IDLE模式

有多种方式可以解除 sub-IDLE 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽外部中断请求信号（INTP0 至 INTP7 引脚输入）、在 sub-IDLE 模式下可操作的外设功能的未屏蔽的内部中断请求信号，或复位信号（由 RESET 引脚输入复位、WDT2RES 信号，低电压检测器（LVI），或时钟监视器（CLM））。PLL 返回到设置 sub-IDLE 模式之前的操作状态。

通过中断请求信号解除 Sub-IDLE 模式时，设置为副时钟工作模式。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号解除 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，都可以解除 sub-IDLE 模式，且与中断请求信号的优先级无关。

但是，如果在一个中断服务程序中设置 sub-IDLE 模式，则之后发出的中断请求信号的服务如下：

(a) 如果发出的中断请求信号优先级低于当前正在服务的中断请求，则 sub-IDLE 模式解除，并且对该中断请求信号不作响应。该中断请求信号被保持。

(b) 如果发出的中断请求信号优先级高于当前正在服务的中断请求（包括不可屏蔽中断请求信号），则 sub-IDLE 模式解除，且响应该中断请求信号。

注意事项 1. 通过 PSC.NMI1M, PSC.NMI0M, 和 PSC.INTM 位设置为 1 被禁止的中断请求信号变为无效，并且 sub-IDLE 模式不会被解除。

2. 当 sub-IDLE 模式解除时，从产生解除 sub-IDLE 模式的中断请求信号至模式被解除需要经过副时钟的 12 个周期（大约 366 (s)）。

表 25-11. 通过中断请求信号解除 Sub-IDLE 模式后的操作

解除源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	执行对句柄地址的跳转。	
可屏蔽中断请求信号	执行对句柄地址的跳转或执行下一条指令。	执行下一条指令。

(2) 通过复位解除 sub-IDLE 模式

执行的操作与正常的复位操作相同。

表 25-12. Sub-IDLE 模式的操作状态

Sub-IDLE 模式的设置		运行状态	
		主时钟振荡时	主时钟停止时
副时钟振荡器		振荡使能	
内部振荡器		振荡使能	
PLL		可操作	停止操作 ^{注1}
CPU		停止操作	
DMA 控制器		停止操作	
中断控制器		停止操作	
定时器	停止操作	停止操作	
	停止操作	停止操作	
	TMM0 至 TMM3	当 fr/8 或 fxt 选择为计数时钟时，可操作	
	TMT0	停止操作	
实时计数器 (RTC)		可操作	当 fxt 选择为计数时钟时，可操作
看门狗定时器 (WDT2)		当 fr 或 fxt 选择为计数时钟时，可操作	
串行接口	CSIF0 至 CSIF4	当 \overline{SCKFn} 输入时钟被选为计数时钟时，可操作 (n=0 至 4)	
	I ² C00 至 I ² C02	停止操作	
	UARTC0 至 UARTC4	停止操作 (但是当选择 ASCKC0 引脚时钟时，UARTC0 可操作)	
A/D 转换器		保持操作 (转换结果保持) ^{注2}	
D/A 转换器		保持操作 (输出保持) ^{注2}	
实时输出功能 (RTO)		停止操作 (输出保持)	
按键中断功能 (KR)		可操作	
CRC 操作电路		停止操作	
外部总线接口		参见 第五章 总线控制功能	
端口功能		保持设置为 Sub-IDLE 模式前的状态	
内部数据		CPU 寄存器、状态位、数据及其它内部数据，例如内部 RAM 的内容，都保持为设置 Sub-IDLE 模式之前的状态。	
USB 功能		停止操作	
USB 主机		停止操作	

- 注 1. 在停止主时钟前，请确保停止锁相环 (PLLCTL.PLLON 位=0)。
- 注 2. 为了降低功率损耗，在切换到 sub-IDLE 模式前，停止 A/D 转换器和 D/A 转换器。

26.1 概述

可以使用以下复位功能：

(1) 4 种类型的复位源

- 通过 $\overline{\text{RESET}}$ 引脚的外部复位输入
- 通过看门狗定时器 2(WDT2)的溢出复位(WDT2RES)
- 通过比较低电压检测器(LVI)供电电压和被检测电压而系统复位
- 通过时钟监控器(CLM) 检测振荡停止而系统复位

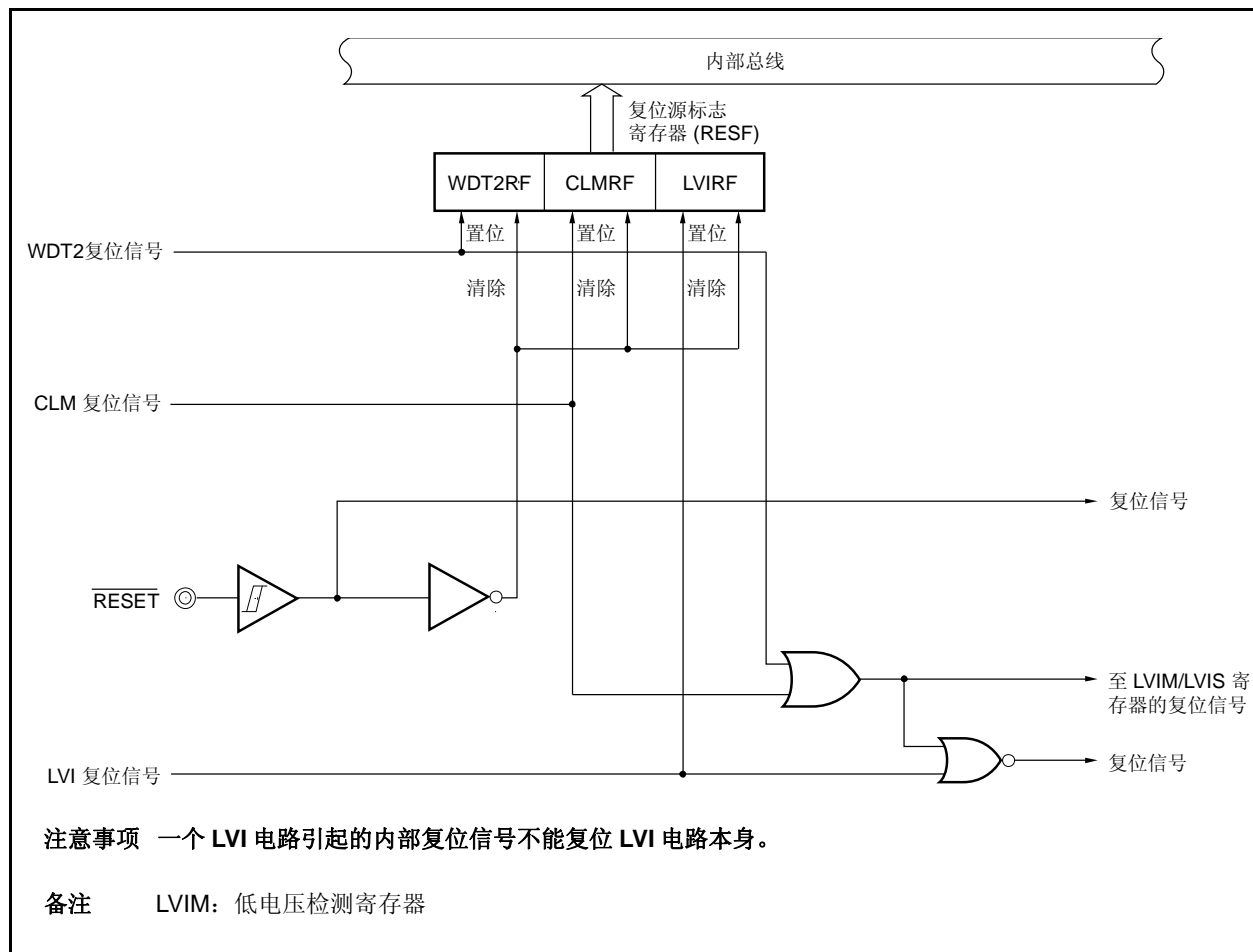
复位解除后，可以通过复位源标志寄存器(RESF)确定复位源。

(2) 紧急操作模式

如果 WDT2 在复位后插入的主时钟振荡稳定期间内溢出，则主时钟振荡被判断为异常，且 CPU 开始运行于内部振荡时钟。

注意事项 紧急操作模式下，除了那些能够在内部时钟振荡工作的模块的寄存器，诸如“中断功能、端口功能、WDT2 或定时器 M”之类，不要访问其它片上外设 I/O 寄存器。此外，在该模式下禁止向 CSIF0 至 CSIF4 以及 UARTC0 提供外部时钟。

图 26-1. 复位功能框图



26.2 检查复位源的寄存器

V850ES/JG3-U 和 V850ES/JH3-U 单片机有 4 种类型的复位源。复位解除后，可以由复位源标志寄存器(RESF)确认已经发生的复位源。

(1) 复位源标志寄存器 (RESF)

RESF 寄存器是一个专用寄存器，它只能通过一个特定序列的组合来写入(参见 3.4.7 专用寄存器)

RESF 寄存器表明复位信号由哪个源产生。

该寄存器可按字节或按位进行读取或写入。

RESET 引脚输入将该寄存器清为 00H。如果复位源不是 $\overline{\text{RESET}}$ 引脚的复位信号，其初始值将不同。

复位后: 00H^注 R/W 地址: FFFFF888H

	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	未产生
1	已产生

CLMRF	来自CLM的复位信号
0	未产生
1	已产生

LVIRF	来自LVI的复位信号
0	未产生
1	已产生

注 当通过 $\overline{\text{RESET}}$ 引脚执行一个复位时，RESF 寄存器的值被清为 00H。当通过看门狗定时器 2 (WDT2)、低电压检测器 (LVI)或时钟监视器(CLM)执行一个复位时，该寄存器的复位标志(WDT2RF 位、CLMRF 位和 LVIRF 位)被置位。而其它复位源时保持不变。

注意事项 只有“0”可以被写到该寄存器的每位。如果写入“0”和置位标志（发生复位时）发生冲突，则置位标志优先。

26.3 操作

26.3.1 通过 $\overline{\text{RESET}}$ 引脚的复位操作

当 $\overline{\text{RESET}}$ 引脚输入低电平时，系统复位，各个硬件单元被初始化。

当 $\overline{\text{RESET}}$ 引脚的电平由低变为高时，复位状态被解除。

表 26-1. $\overline{\text{RESET}}$ 引脚输入的硬件状态

项目	复位期间	复位后
主时钟振荡器 (fx)	振荡停止	振荡开始
副时钟振荡器 (fxT)	振荡继续	
内部 振荡器	振荡停止	振荡开始
外部时钟 (fx 至 fx/1,024)	操作停止	在确保振荡稳定时间之后开始操作
内部系统时钟 (fCLK), CPU 时钟(fCPU)	操作停止	在确保振荡稳定时间之后开始操作(初始化至 fx/8)
CPU	初始化	在确保振荡稳定时间之后程序开始执行
看门狗定时器 2	操作停止 (初始化为 0)	用内部振荡时钟作源时钟，从 0 开始往上计数
内部 RAM	在加电复位时或 CPU 存取和复位输入冲突（数据被损坏）时，状态不确定；否则，保留复位后的瞬时值。	
I/O 线 (端口/复用功能引脚)	高阻 [※]	
片上外设 I/O 寄存器	初始化到指定状态， OCDM 寄存器设为 (01H)。	
其它片上外设功能	操作停止	在确保振荡稳定时间之后开始操作

注 当电源开启时，即使在复位期间，下列引脚也可能会临时输出一个不确定的电平。

- P10/ANO0 引脚
- P11/ANO1 引脚
- DDO 引脚（仅限 V850ES/JH3-U）
- P53/SIF2/TIAB00/KR3/TOAB00/RTP03/DDO 引脚（仅限 V850ES/JG3-U）

注意事项 OCDM 寄存器通过 $\overline{\text{RESET}}$ 引脚的输入来初始化。因此，必须注意的是：在复位解除之后 OCDM.OCDM0 位被清除之前，如果有高电平输入 P56/INTP05/DRST 引脚，可能会进入片上调试模式。详情参见第四章 端口功能。

图 26-2. RESET 引脚输入引起的复位操作时序

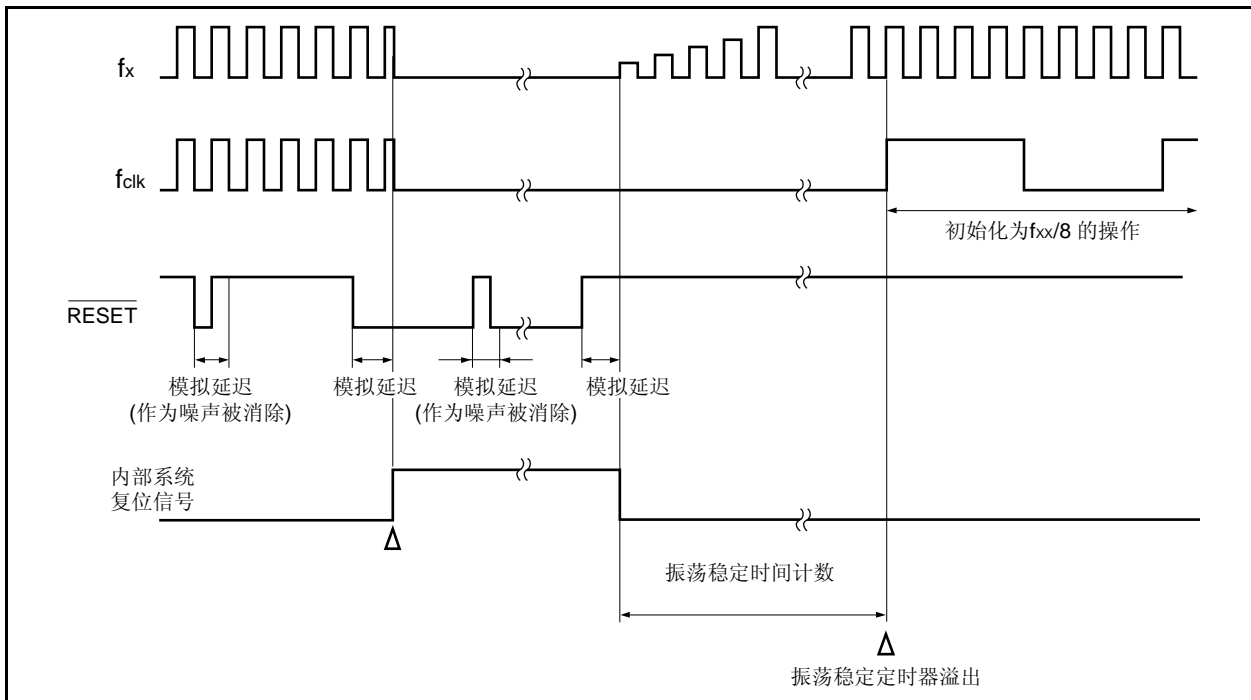
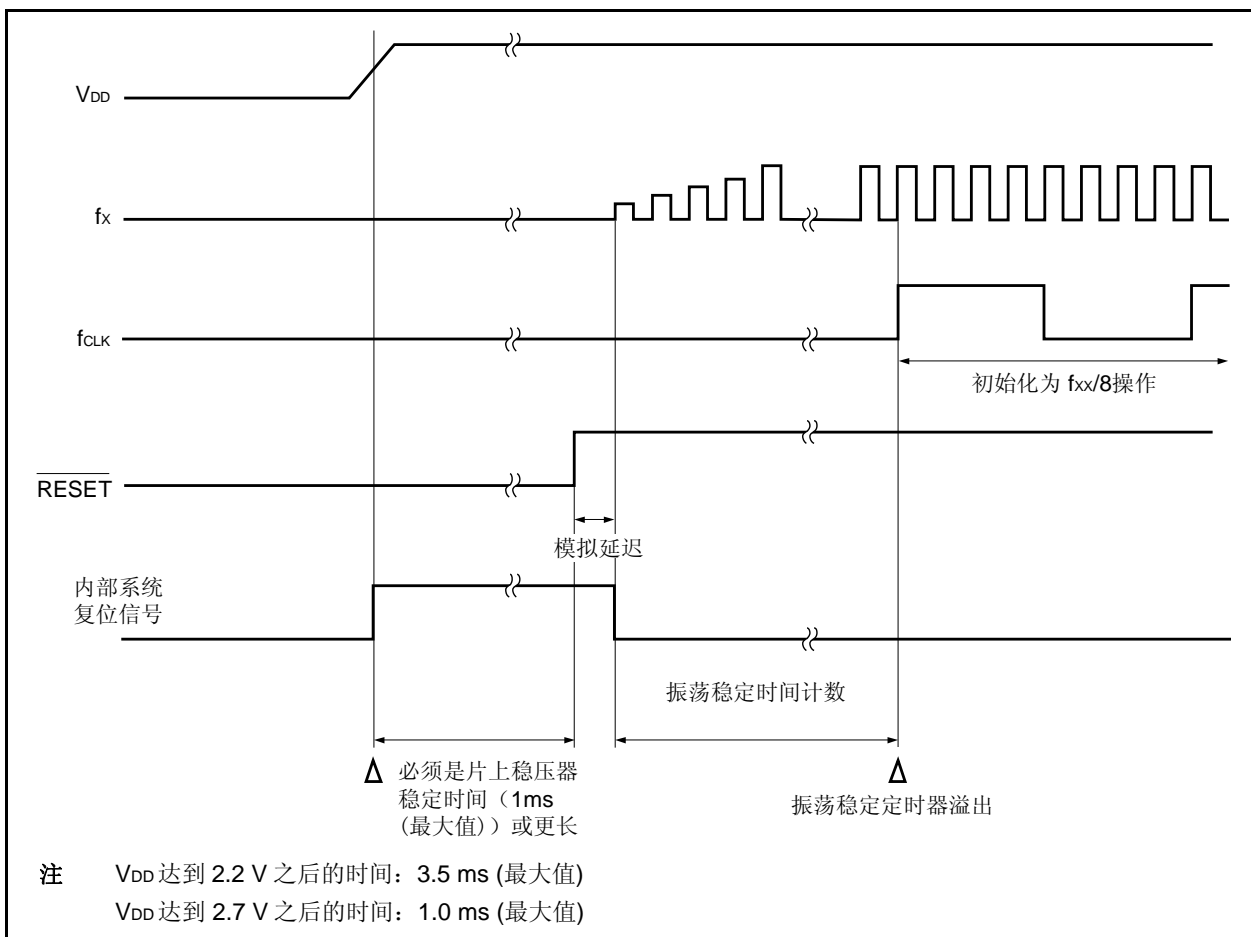


图 26-3. 上电复位操作的时序



26.3.2 通过看门狗定时器 2 的复位操作

若看门狗定时器 2 被设置为溢出复位工作模式，当看门狗定时器 2 发生溢出时（产生 WDT2RES 信号），则执行系统复位，并且硬件被初始化为初始状态。

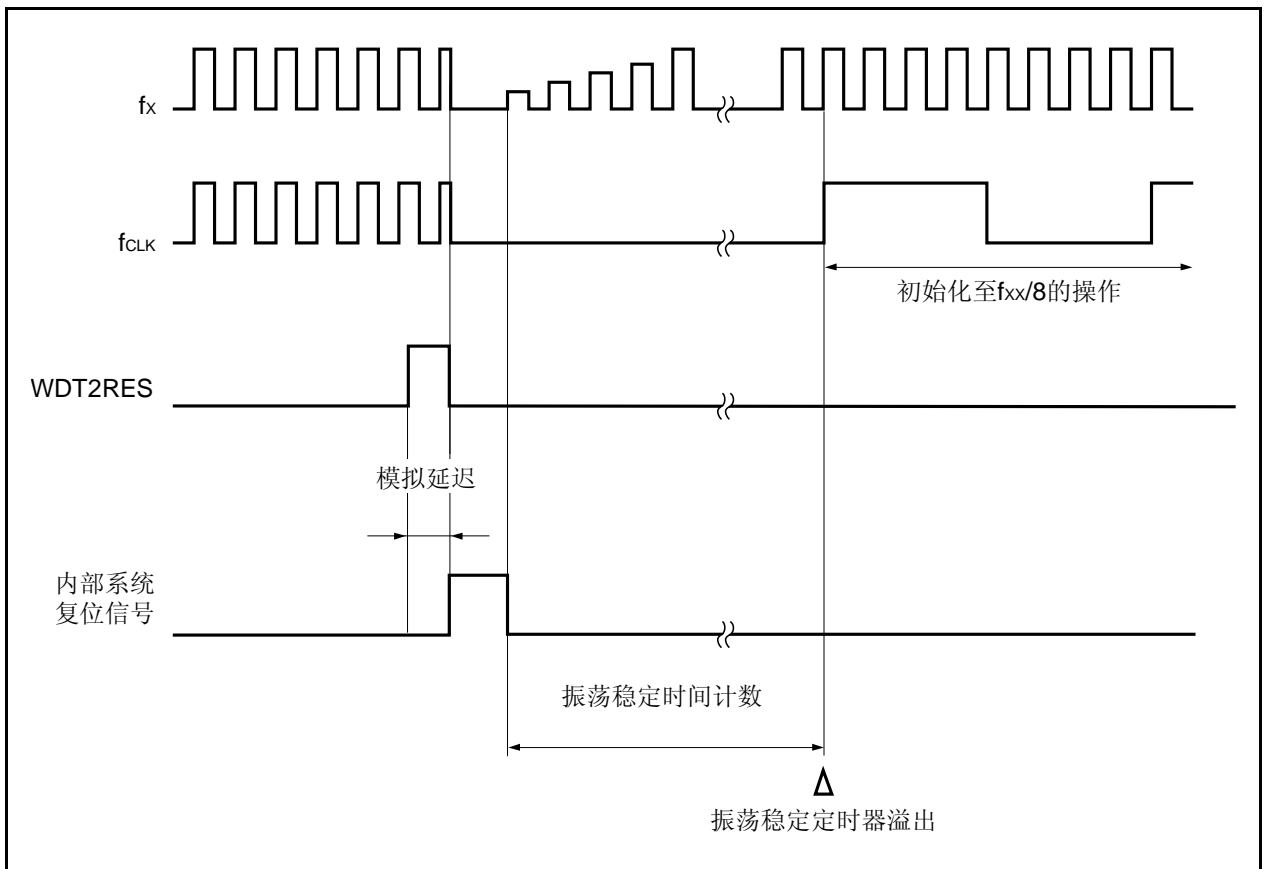
在看门狗定时器 2 溢出后，进入复位状态并持续预定时间(模拟延迟)，并且，复位状态随后被自动解除。

在复位期间主时钟振荡器停止。

表 26-2. 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 (fx)	振荡停止	振荡开始
副时钟振荡器 (fxr)	振荡继续	
内部振荡器	振荡停止	振荡开始
外部时钟(fxx 至 fxx/1,024)	操作停止	在确保振荡稳定时间之后开始操作
内部系统时钟(fxx), CPU 时钟(fcpu)	操作停止	在确保振荡稳定时间之后开始操作(初始化为 fxx/8)
CPU	初始化	在确保振荡稳定时间之后程序开始执行
看门狗定时器 2	操作停止 (初始化为 0)	内部振荡时钟作为源时钟，从 0 开始往上计数。
内部 RAM	在加电复位时或当 CPU 存取和复位输入冲突（数据被损坏）时，状态不确定，否则，保留复位后的瞬时值。	
I/O 线(端口/复用功能引脚)	高阻	
片上外设 I/O 寄存器	初始化为特定状态，保留 OCDM 寄存器的值。	
其它片上外设功能	操作停止	在确保振荡稳定时间之后开始操作。

图 26-4. 通过 WDT2RES 信号产生的复位操作时序



26.3.3 通过低电压检测器的复位操作

当 LVI 操作使能时，如果供电电压下降到低电压检测器的检测电压以下，执行一次系统复位（当 LVIM.LVIMD 位被设为 1 时），且硬件被初始化到初始状态。

从检测到供电电压下降一直持续到供电电压上升超过 LVI 检测电压，一直都保持在复位状态。

在复位期间主时钟振荡器停止。

当 LVIMD 位 = 0 时，如果检测到一个低电压则产生一个中断请求信号(INTLVI)。

表 26-3. 由低电压检测器复位操作时的硬件状态

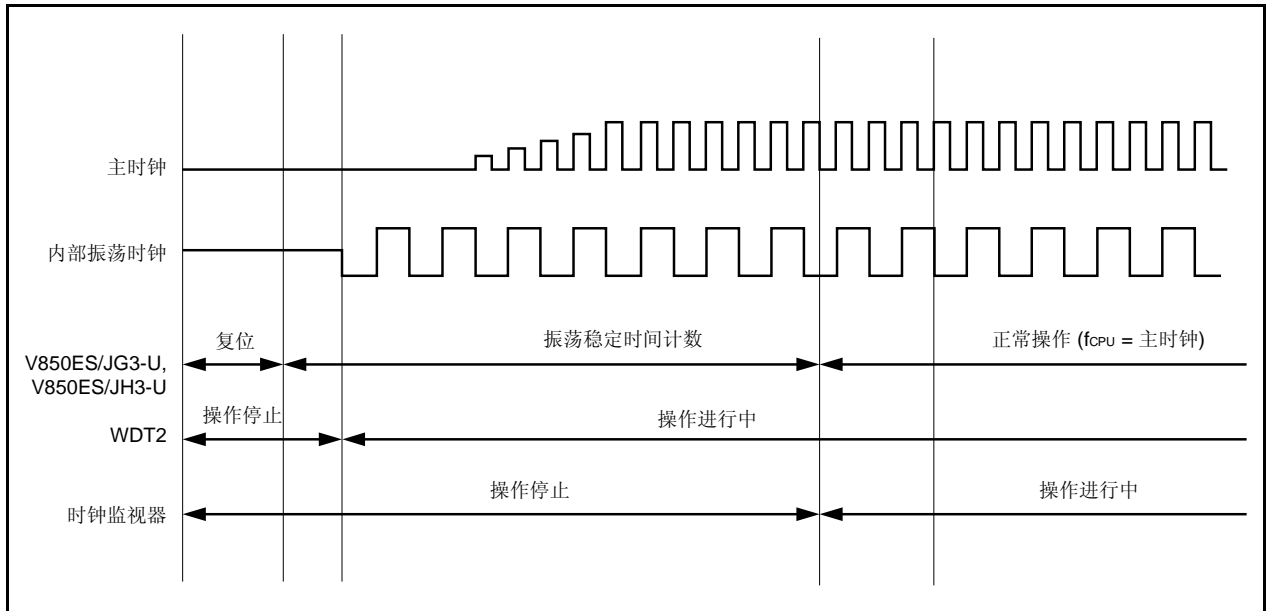
项目	复位期间	复位后
主时钟振荡器(fx)	振荡停止	振荡开始
副时钟振荡器(fxt)	振荡继续	
内部振荡器 r	振荡停止	振荡开始
外部时钟(fx 至 fx/1,024)	操作停止	在确保振荡稳定时间之后开始操作
内部系统 时钟(fxx), CPU 时钟(fCPU)	操作停止	在确保振荡稳定时间之后开始操作(初始化为 fxx/8)
CPU	初始化	在确保振荡稳定时间之后程序开始执行
看门狗定时器 2	操作停止(初始化为 0)	内部振荡时钟作为源时钟，从 0 开始往上计数。
内部 RAM	在加电复位时或当 CPU 访问和复位输入冲突（数据被损坏）时，则会状态不确定。否则，保留复位后的瞬时值	
I/O 线(端口/复用功能引脚)	高阻	
片上外设 I/O 寄存器	初始化到指定状态，保留 OCDM 寄存器的值。	
LVI	操作停止	
其它片上外围功能	操作停止	在确保振荡稳定时间之后开始操作。

备注 关于低电压检测器的复位时序，参见 第二十八章 低电压检测器 (LVI)。

26.3.4 复位解除后的操作

复位解除后，主时钟开始振荡且确保经过振荡稳定时间（OSTS 寄存器初始值： $2^{16}/f_x$ ）之后，CPU 开始执行程序。使用内部振荡时钟作为源时钟时，在复位解除之后 WDT2 立刻开始运行。

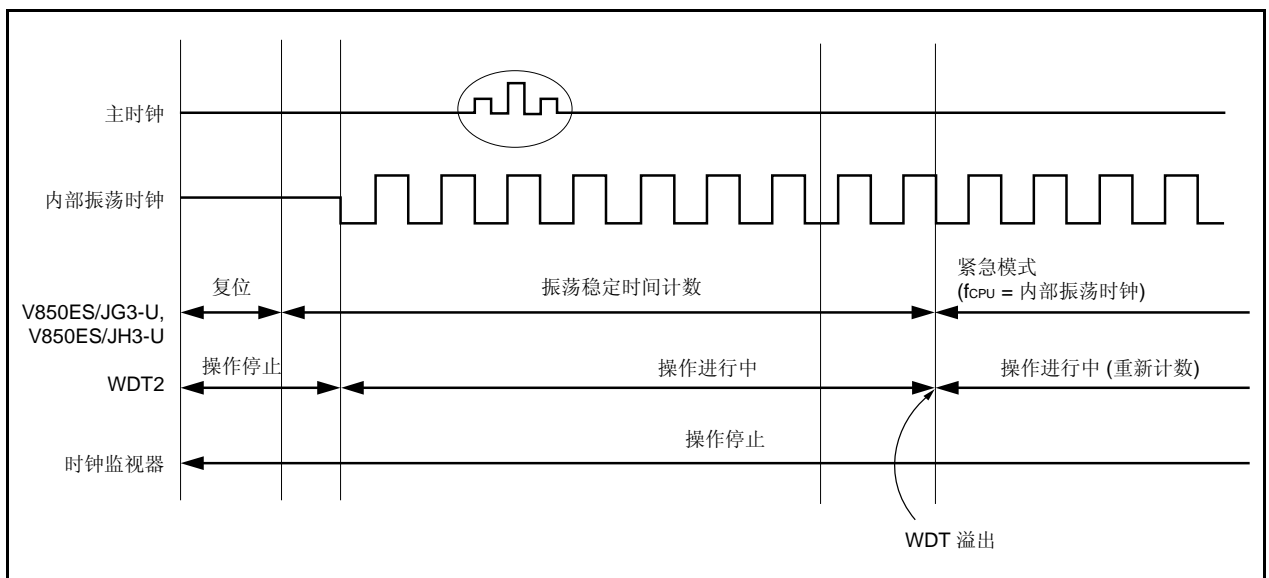
图 26-5. 复位解除后的操作



(1) 紧急操作模式

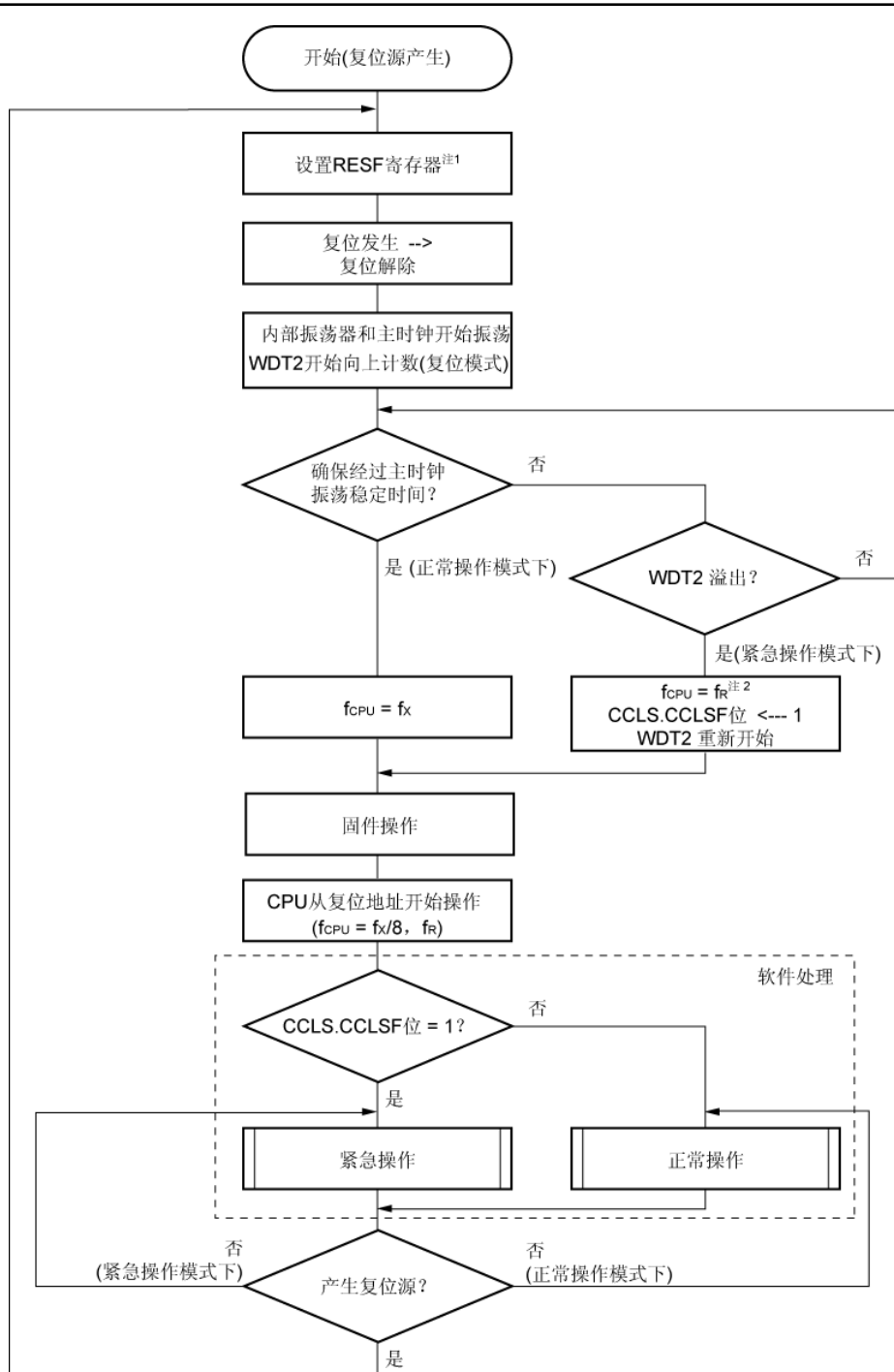
如果在振荡稳定时间完成之前主时钟发生异常，则在执行 CPU 程序之前，WDT2 就已经溢出。此时，将内部振荡时钟作为源时钟，CPU 开始执行程序。

图 26-6. 复位解除后的操作



可以使用 CPU 操作时钟状态寄存器(CCLS)来检测 CPU 操作时钟的状态。

26.3.5 复位功能操作流程



注 1. 根据复位源对不同的位置 1。

复位源	WDT2RF 位	CRMRF 位	LVIRF 位
RESET 引脚	0	0	0
WDT2	1	保留复位前的值。	保留复位前的值。
CLM	保留复位前的值。	1	保留复位前的值。
LVI	保留复位前的值。	保留复位前的值。	1

2. 内部振荡器不能停止。

第二十七章 时钟监控器

27.1 功能

时钟监控器使用内部振荡时钟，对主时钟进行采样，当主时钟振荡停止时，产生复位请求信号。

一旦通过操作使能标志允许时钟监控器的操作，则除了复位之外，没有办法可以使之清 0。

当时钟监控器引发复位时，RESF.CLMRF 位置位。关于 RESF 寄存器的细节，参见 26.2 检查复位源寄存器。

在下列条件时，时钟监控器自动停止：

- 解除 STOP 模式后的振荡稳定期间
- 主时钟停止时（从副时钟工作期内 PCC.MCK 位 = 1 时开始，直至主时钟工作期内 PCC.CLS 位 = 0）
- 采样时钟（内部振荡时钟）停止时
- 当 CPU 运行于内部振荡时钟工作时

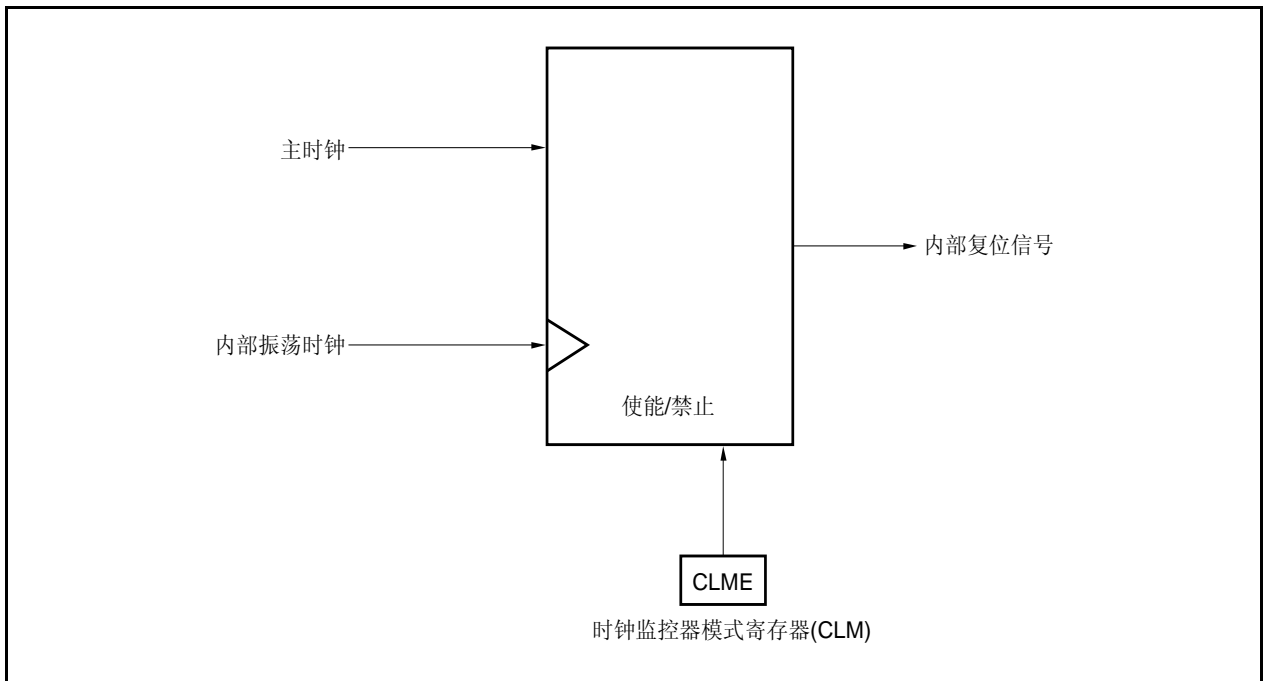
27.2 配置

时钟监控器包含以下硬件。

表 27-1. 时钟监控器的配置

项目	配置
控制寄存器	时钟监控模式寄存器 (CLM)

图 27-1. 通过 RESET 引脚输入进行复位操作的时序



27.3 寄存器

时钟监控器由时钟监控器模式寄存器(CLM)控制。

(1) 时钟监控器模式寄存器(CLM)

CLM 寄存器是专用寄存器。只有通过特定序列的组合才能写入该寄存器（见 3.4.7 专用寄存器）。

该寄存器用于设置时钟监控器的操作模式。

该寄存器可以按字节或按位进行读写。

系统复位后，设置该寄存器为 00H。

复位后: 00H	R/W	地址: FFFF870H						
	7	6	5	4	3	2	1	<0>
CLM	0	0	0	0	0	0	0	CLME

CLME	时钟监控器操作使能或禁止
0	禁止时钟监控器操作。
1	允许时钟监控器操作。

- 注意事项**
1. 一旦 **CLME** 位设置为 **1**，除非复位，不能以任何其它方式将其清 **0**。
 2. 当时钟监控器引发复位时，**CLME** 位被清为 **0**，而 **RESF.CLMRF** 位被置为 **1**。
 3. 请确保设置第 7 位至第 1 位为“**0**”。

27.4 操作

本节介绍时钟监控器的功能。启动条件和停止条件如下：

<启动条件>

将 CLM.CLME 位设为 1，允许操作。

<停止条件>

- 解除 STOP 模式后对振荡稳定时间进行计数时
- 当主时钟停止时（从副时钟工作期内 PCC.MCK 位 = 1 时开始，直到主时钟工作期内 PCC.CLS 位 = 0）
- 当 采样时钟(内部振荡时钟)停止时
- 当 CPU 运行于内部振荡器时钟时

表 27-2. 时钟监控器操作状态

（内部振荡器时钟工作期间，CLM.CLME 位 = 1 时）

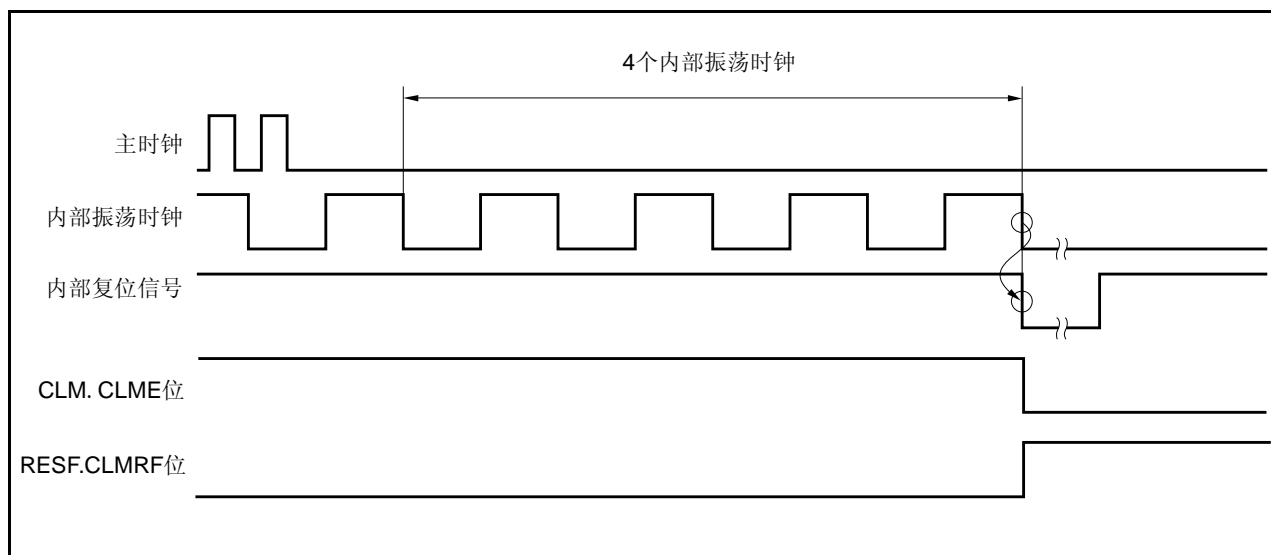
CPU 工作时钟	操作模式	主时钟状态	内部振荡时钟状态	时钟监控器状态
主时钟	HALT 模式	振荡	振荡 ^{注1}	工作 ^{注2}
	IDLE1, IDLE2 模式	振荡	振荡 ^{注1}	工作 ^{注2}
	STOP 模式	停止	振荡 ^{注1}	停止
副时钟 (PCC.MCK = 0)	Sub-IDLE 模式	振荡	振荡 ^{注1}	工作 ^{注2}
副时钟(PCC.MCK= 1)	Sub-IDLE 模式	停止	振荡 ^{注1}	停止
内部振荡时钟	—	停止	振荡 ^{注3}	停止
复位期间	—	停止	停止	停止

- 注
1. 通过设置 RCM.RSTOP 位为 1，可以停止内部振荡器。
 2. 当内部振荡器停止时，时钟监控器也同样停止。
 3. 内部振荡器不能通过软件停止。

(1) 主时钟振荡停止时的操作 (CLME 位=1)

如果在 CLME 位 = 1 时主时钟振荡停止, 则产生一个内部复位信号, 如图 27-2 所示。

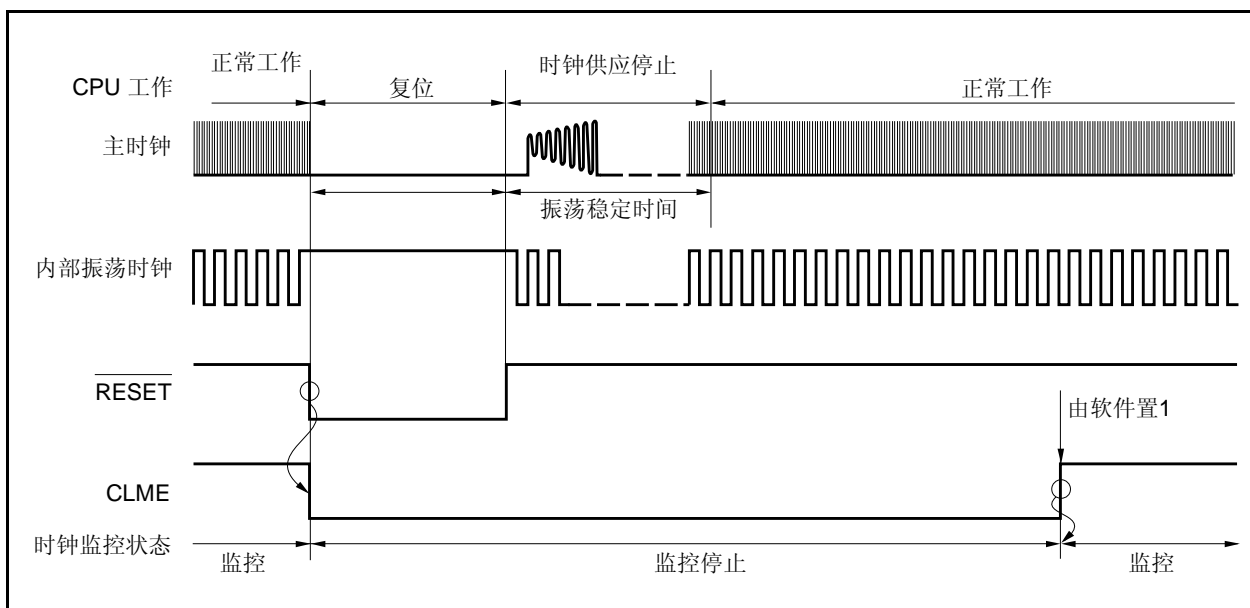
图 27-2. 因主时钟振荡器停止造成的复位时间



(2) $\overline{\text{RESET}}$ 输入后时钟监控器的状态

$\overline{\text{RESET}}$ 输入将 CLM.CLME 位清为 0, 并停止时钟监控器的操作。在主时钟振荡稳定时间结束后, 由软件将 CLME 位设为 1, 则开始监控。

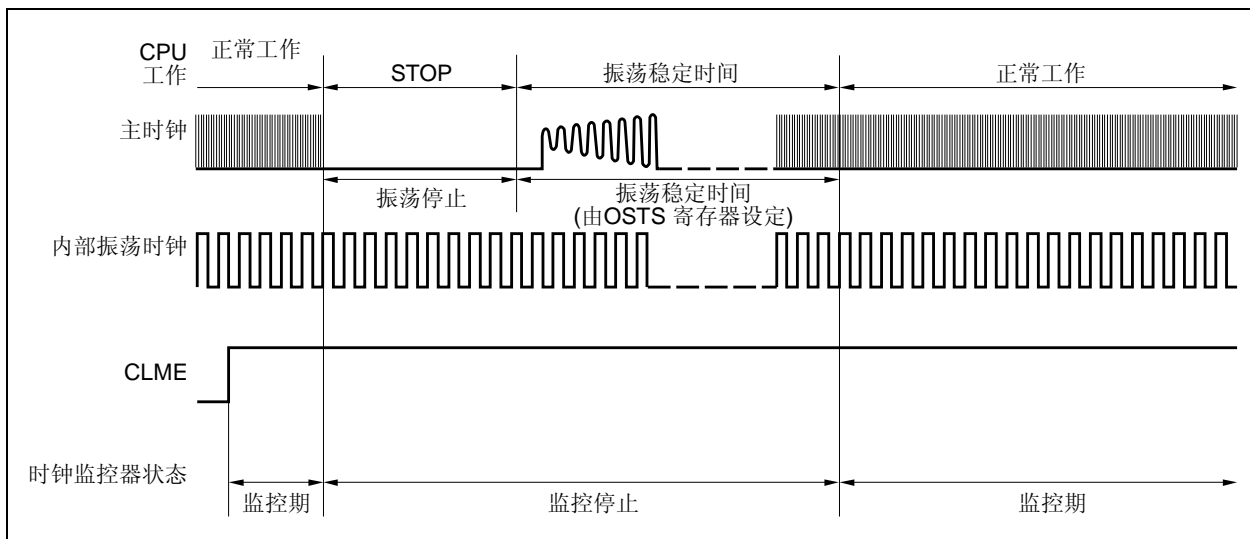
图 27-3. $\overline{\text{RESET}}$ 输入后的时钟监控器状态
($\overline{\text{RESET}}$ 输入且在主时钟振荡稳定时间结束后, 设置 CLM.CLME 位 = 1)



(3) 在 STOP 模式下或 STOP 模式解除后的操作

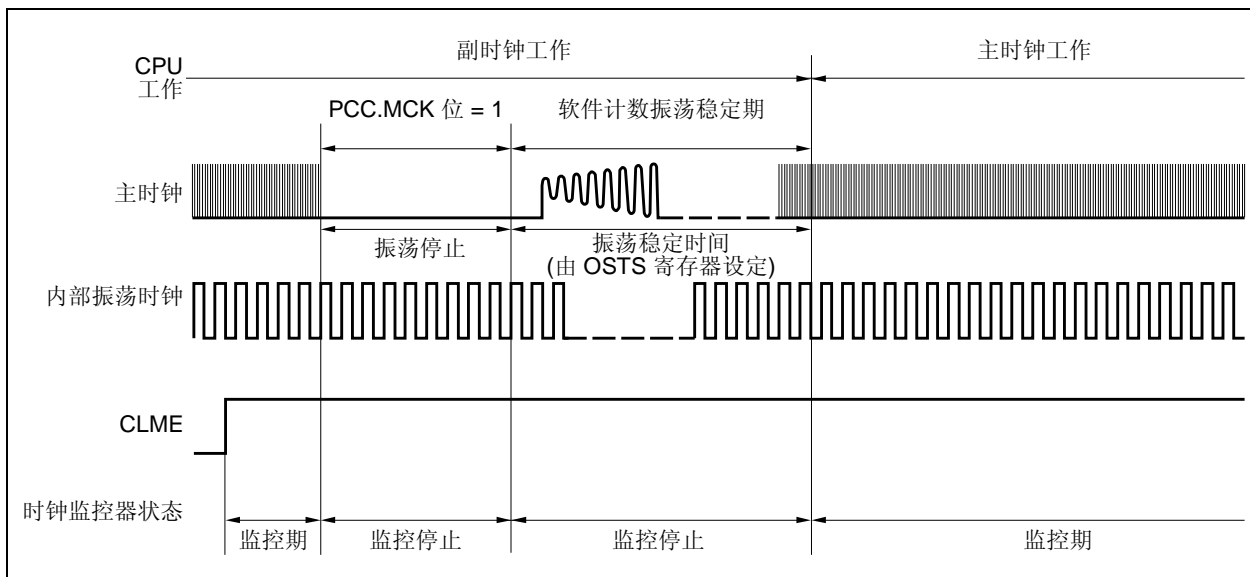
如果在 CLME 位 = 1 时设置 STOP 模式，那么监控器会在 STOP 模式下停止工作，在振荡稳定时间计数期间也停止工作。振荡稳定时间过后，监控器将会自动开始工作。

图 27-4. 在 STOP 模式下或 STOP 模式解除后的操作

**(4) 主时钟停止(任意)时的操作**

在副时钟操作 (PCC.CLS 位 = 1) 期间，或通过设置 PCC.MCK 为 1 使主时钟停止工作时，监控器操作停止，直到主时钟开始工作 (PCC.CLS 位 = 0)。当主时钟开始工作时，监控器自动开始工作。

图 27-5. 主时钟停止(任意)时的操作

**(5) 当 CPU 运行于内部振荡时钟(CCLS.CCLS F 位= 1)时的操作**

当 CCLS F 位为 1 时，即使 CLME 位被置为 1，监控器也不会停止工作。

第二十八章 低电压检测器 (LVI)

28.1 功能

低电压检测器(LVI) 具有以下功能。

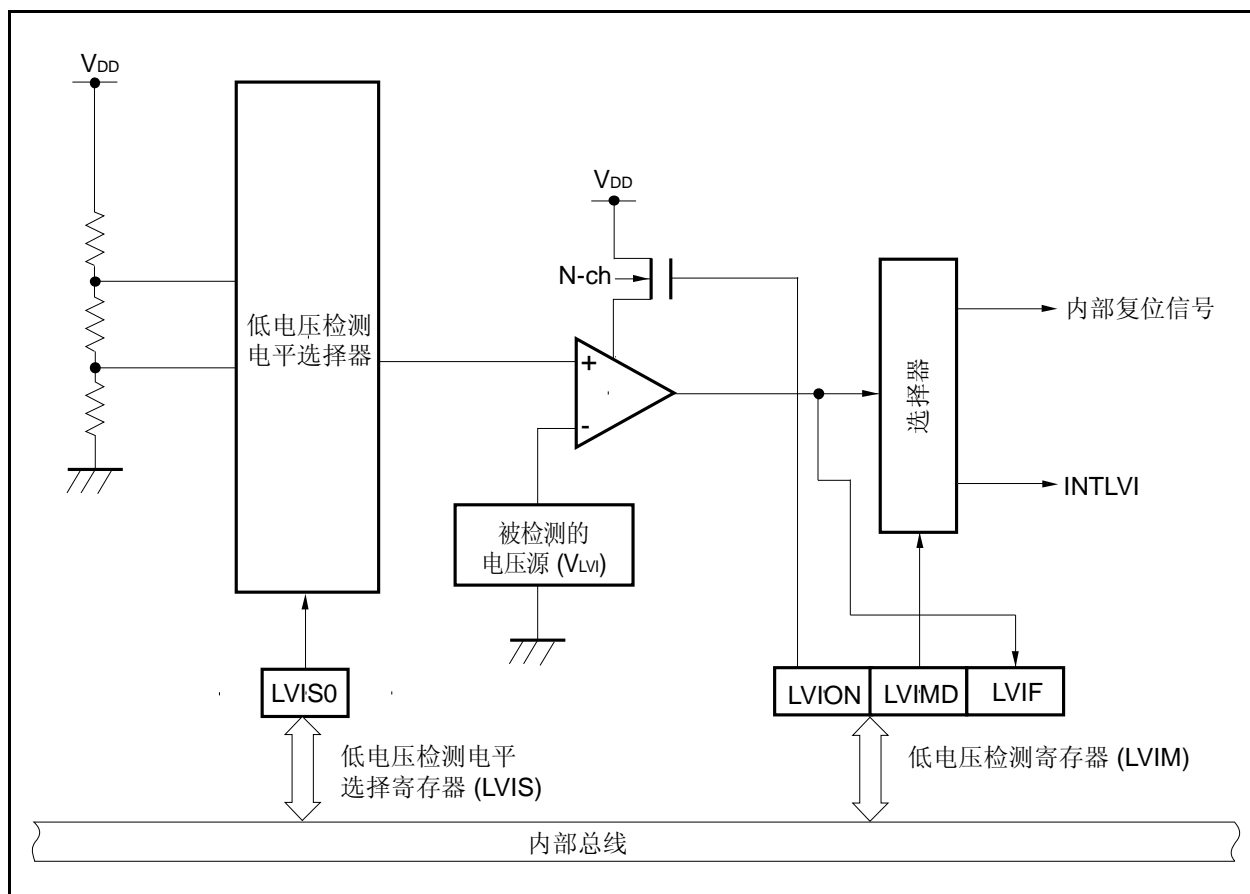
- 如果在使用低电压检测后发生中断，低电压检测器比较供电电压(V_{DD}) 和检测电压 (V_{LVI})，并且，在供电电压下降或上升经过检测电压时，产生一个内部中断信号。
- 如果在使用低电压检测后发生复位，当供电电压 (V_{DD}) 下降经过检测电压 (V_{LVI})时，低电压检测器产生一个内部复位信号。
- 可由软件选择为中断或复位信号。
- 可以在 STOP 模式下工作。

低电压检测器用来产生复位信号，当复位信号产生时 RESF.LVIRF 位被设置为 1。关于 RESF 寄存器的详情，参见 26.2 复位源检查寄存器。

28.2 配置

低电压检测器的框图如下所示。

图 28-1. 低电压寄存器框图



28.3 寄存器

低电压检测器由下列寄存器控制。

- 低电压检测寄存器 (LVIM)
- 内部 RAM 数据状态寄存器 (RAMS)

(1) 低电压检测寄存器 (LVIM)

LVIM 是一个专用寄存器。只有通过特定序列的组合才能写入该寄存器（参见 3.4.7 专用寄存器）。

LVIM 寄存器用于使能或禁止低电压检测，并且设置低电压检测器的工作模式。

该寄存器可以按字节或按位进行读取或写入，但是，LVIF 位为只读位。

复位后: 00H ^{注1}	R/W	地址: FFFFF890H						
	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF
	LVION	使能或禁止低电压检测操作						
	0	禁止工作。						
	1	使能工作。						
	LVIMD	选择低电压检测的工作模式						
	0	当供电电压下降或上升经过被检测电压时，产生中断信号 INTLVI。						
	1	当供电电压下降经过被检测电压时，产生内部复位信号 LVIRES。						
	LVIF ^{注2}	低电压检测标志						
	0	当供电电压 > 检测电压，或操作禁止时。						
	1	当前接入电源的供电电压 < 检测电压时。						

- 注
1. 由低电压检测复位: 82H
由其它复位源复位: 00H
 2. 在 LVI 开始工作 (LVION 位 = 1) 或发生 INTLVI 后，使用 LVIF 位来确认供电电压的状态。

- 注意事项
1. 当 LVION 和 LVIMD 位为 1 时，低电压检测器不能停止，除非低电压检测之外的其他复位源产生复位信号。
 2. 当 LVION 位设置为 1 时，LVI 电路内的比较器开始工作。在 LVION 位置位后，使用 LVIF 位检测电压前，应该由软件控制等待 0.2 ms 或更长的时间。
 3. 请确保将第 6 位至第 2 位清为“0”。

(2) 内部 RAM 数据状态寄存器 (RAMS)

RAMS 寄存器是个专用寄存器，只能按照特殊的组合顺序来写入（参见 3.4.7 专用寄存器）。

该寄存器是标志寄存器，表示内部 RAM 是否有效。

该寄存器能够按字节或按位进行读取或写入。

RAMF 位的置位/清除条件如下。

- 置位条件： 检测到电压低于指定电平
 通过指令置位
- 清除条件： 按照特定顺序写入 0

复位后：01H [*]	R/W	地址：FFFF892H							
		7	6	5	4	3	2	1	0
RAMS		0	0	0	0	0	0	0	RAMF
	RAMF	内部 RAM 电压检测							
	0	未检测到电压低于 RAM 保持电压							
	1	检测到电压低于 RAM 保持电压							

注 只有当检测到一个低于 RAM 保持电压的电压时，该寄存器被重置。

28.4 操作

由 LVIM.VIMD 位的设置来决定，产生中断信号 (INTLVI)或内部复位信号。
如何指定每个操作的描述和时序图如下。

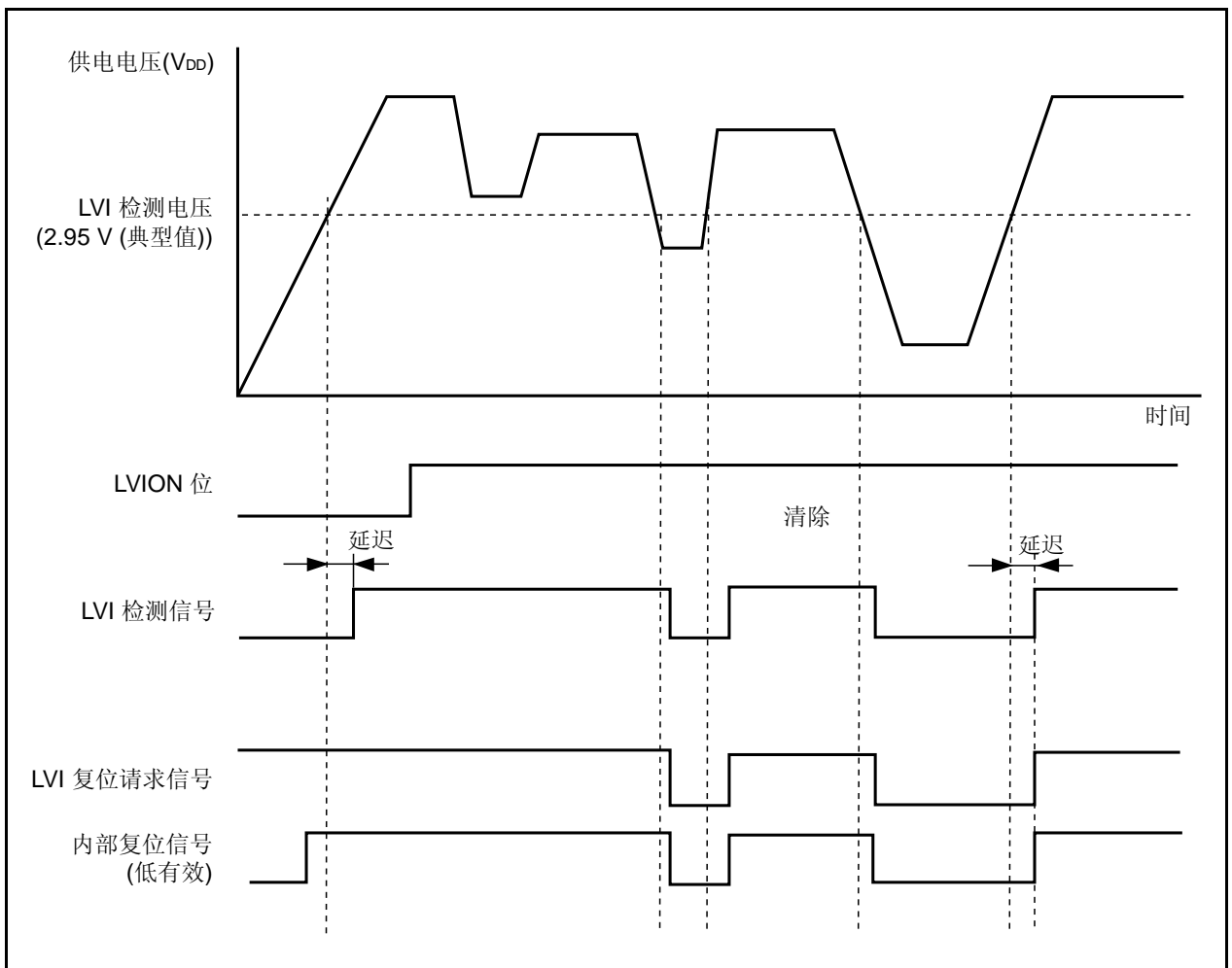
28.4.1 使用内部复位信号

<开始操作>

- <1> 屏蔽 LVI 中断。
- <2> 将 LVIM.LVION 位设为 1 (使能操作)。
- <3> 通过软件插入一个 0.2 ms(最大值)或更长的等待周期。
- <4> 通过 LVIM.LVIF 位检测是否 供电电压 > 检测电压。
- <5> 将 LVIMD 位设为 1 (产生一个内部复位信号)。

注意事项 如果 LVIMD 位被设为 1，LVIM 寄存器的内容不能改变，除非产生一个 LVI 之外的复位请求。

图 28-2. 低电压检测器操作时序(LVIMD 位= 1)



28.4.2 用作中断

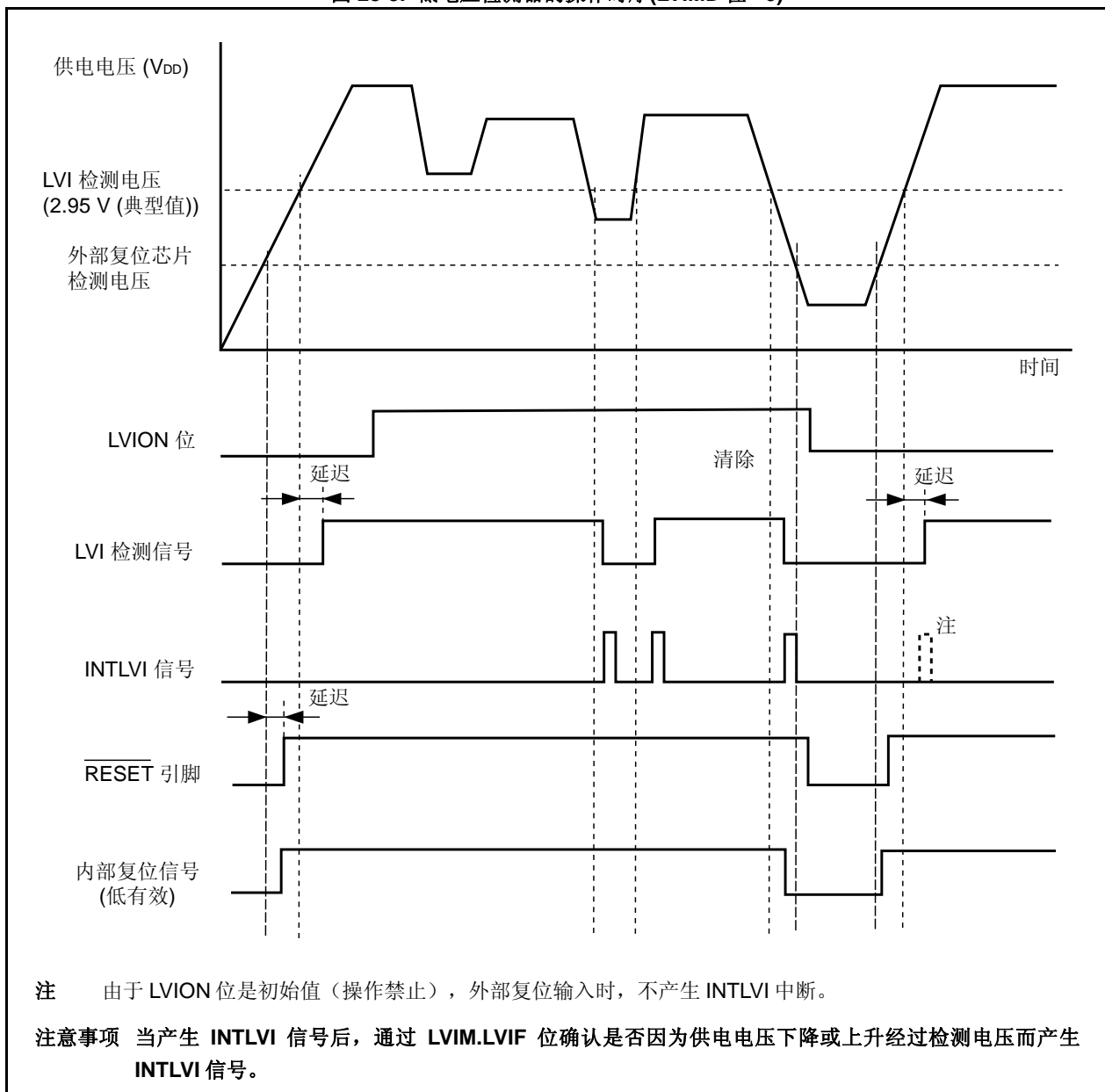
<开始操作>

- <1> 屏蔽 LVI 的中断。
- <2> 将 LVIM.LVION 位置为 1 (使能操作)。
- <3> 通过软件插入一个 0.2 ms(最大值)或更长的等待周期。
- <4> 通过 LVIM.LVIF 位检测是否 供电电压 > 检测电压。
- <5> 清除 LVI 的中断请求标志。
- <6> 解除 LVI 的中断屏蔽。

<停止操作>

将 LVION 位清除为 0。

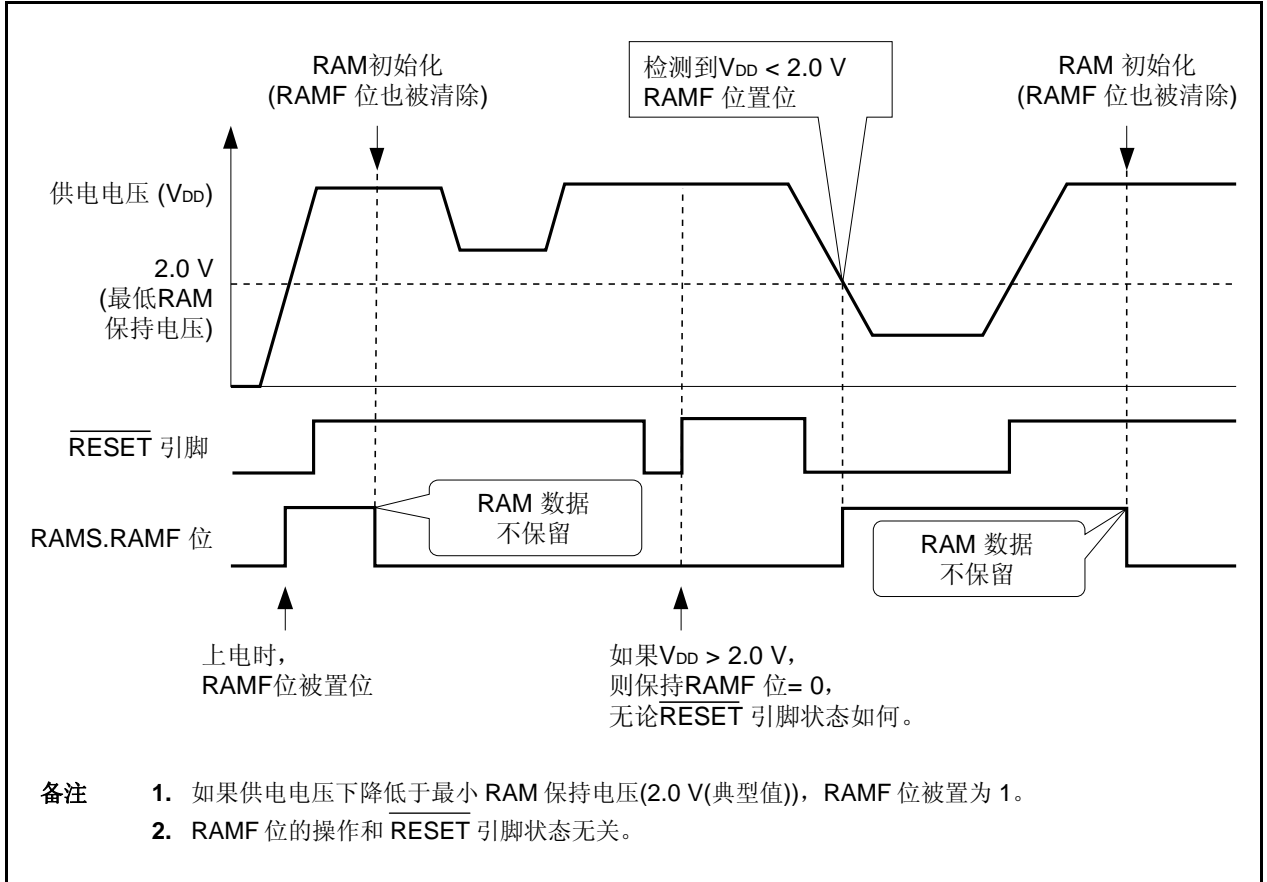
图 28-3. 低电压检测器的操作时序(LVIMD 位= 0)



28.5 RAM 保持电压检测操作

将供电电压和检测电压进行比较，当供电电压降低低于检测电压（包括电源上电）时，则 RAMS.RAMF 位被置为 1。

图 28-4. RAM 保持电压检测功能的操作时序



第二十九章 CRC 功能

29.1 功能

- 使用 CRC 功能电路进行数据块的错误检测。
- 使用 CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$)多项式以 8 位为单元对任何长度的数据块产生 16 位 CRC 校验码。
- 初始值设置到 CRCD 寄存器之后，每当 1 个字节数据被传送到 CRCIN 寄存器，CRC 码就被置入 CRCD 数据寄存器。

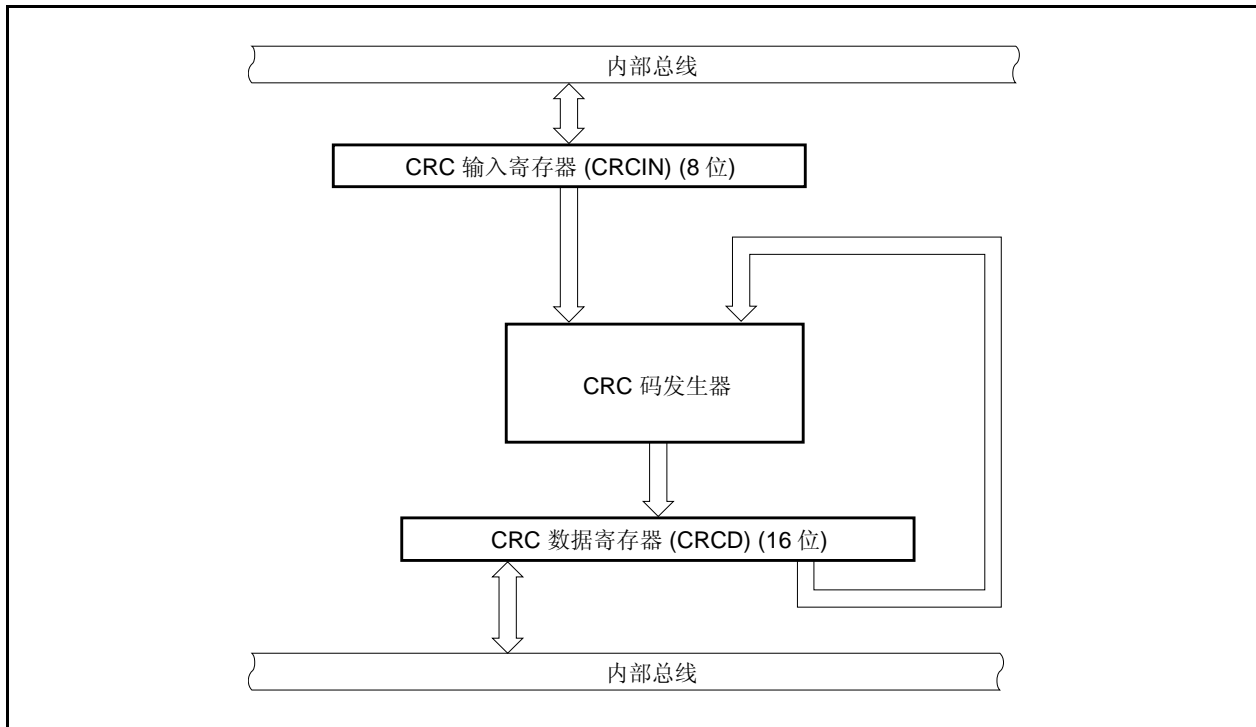
29.2 配置

CRC 控制功能包含下列硬件：

表 29-1. CRC 配置

项目	配置
控制寄存器	CRC 输入寄存器 (CRCIN) CRC 数据寄存器 (CRCD)

图 29-1. CRC 寄存器的功能框图



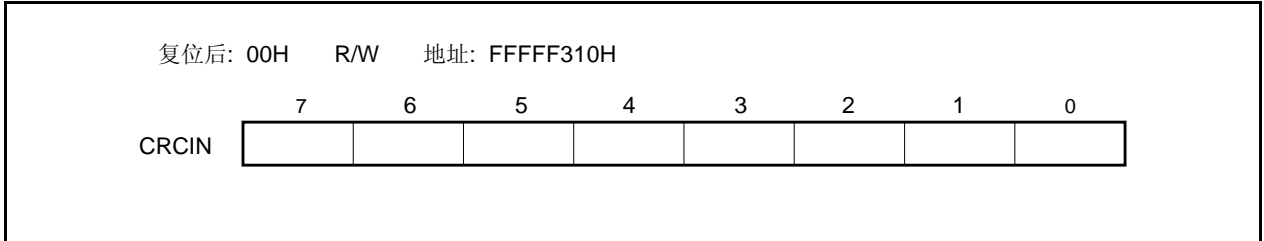
29.3 寄存器

(1) CRC 输入寄存器 (CRCIN)

CRC 输入寄存器是 8 位寄存器，用于设置数据。

该寄存器可按字节进行读写。

系统复位后，该寄存器被设为 00H。



(2) CRC 数据寄存器 (CRCIN)

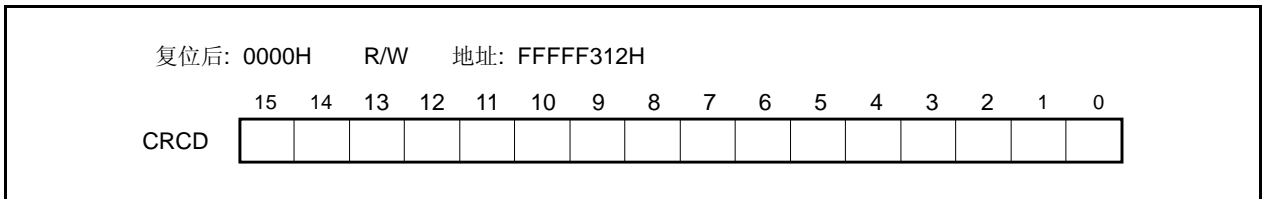
CRCD 寄存器为 16 位寄存器，用来存储 CRC-CCITT 运行结果。

该寄存器可按 16 位宽度进行读写。

系统复位后，该寄存器被清为 0000H。

注意事项 在以下状态禁止访问 CRCD 寄存器。详情参见 3.4.8 (2) 访问片上专用外设 I/O 寄存器。

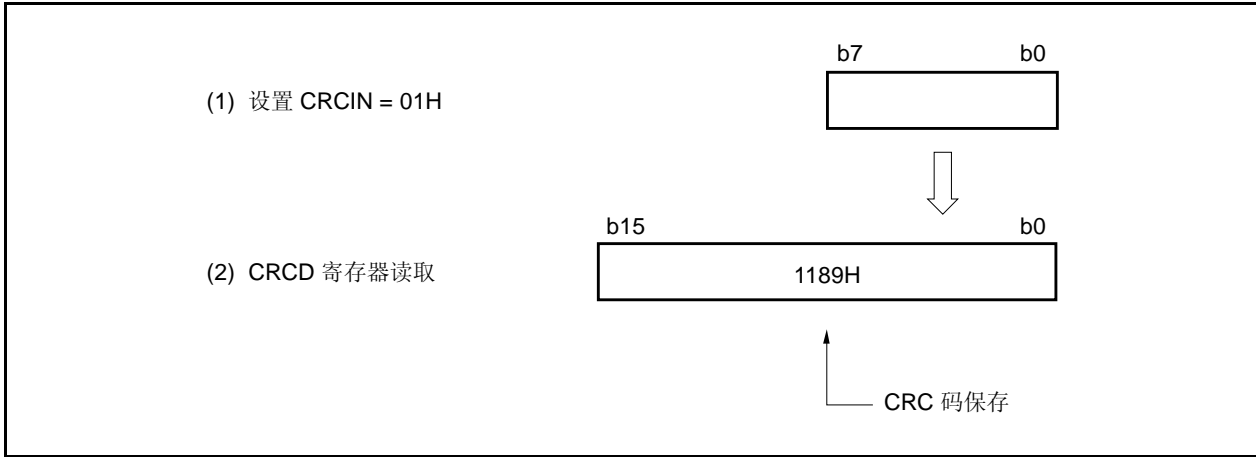
- 当 CPU 运行于副时钟时，而主时钟振荡停止。
- 当 CPU 运行于内部振荡器时钟工作时。



29.4 操作

CRC 工作电路的举例如下。

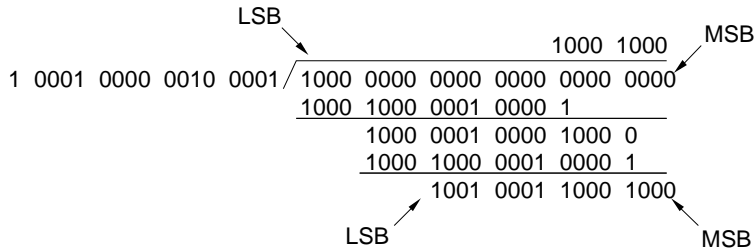
图 29-2. CRC 工作电路操作示例 (LSB 先行)



当 LSB 先行时, 01H 码为(1000 0000)。所以, 当使用 modulo-2 操作公式, 令 $(1000\ 0000) \times X^{16}$ 除以 $(1\ 0001\ 0000\ 0010\ 0001)$ 时, 由生成多项式 $X^{16} + X^{12} + X^5 + 1$ 得到的 CRC 码为其余数。

modulo-2 操作基于以下公式进行。

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 0
- 1 = 1

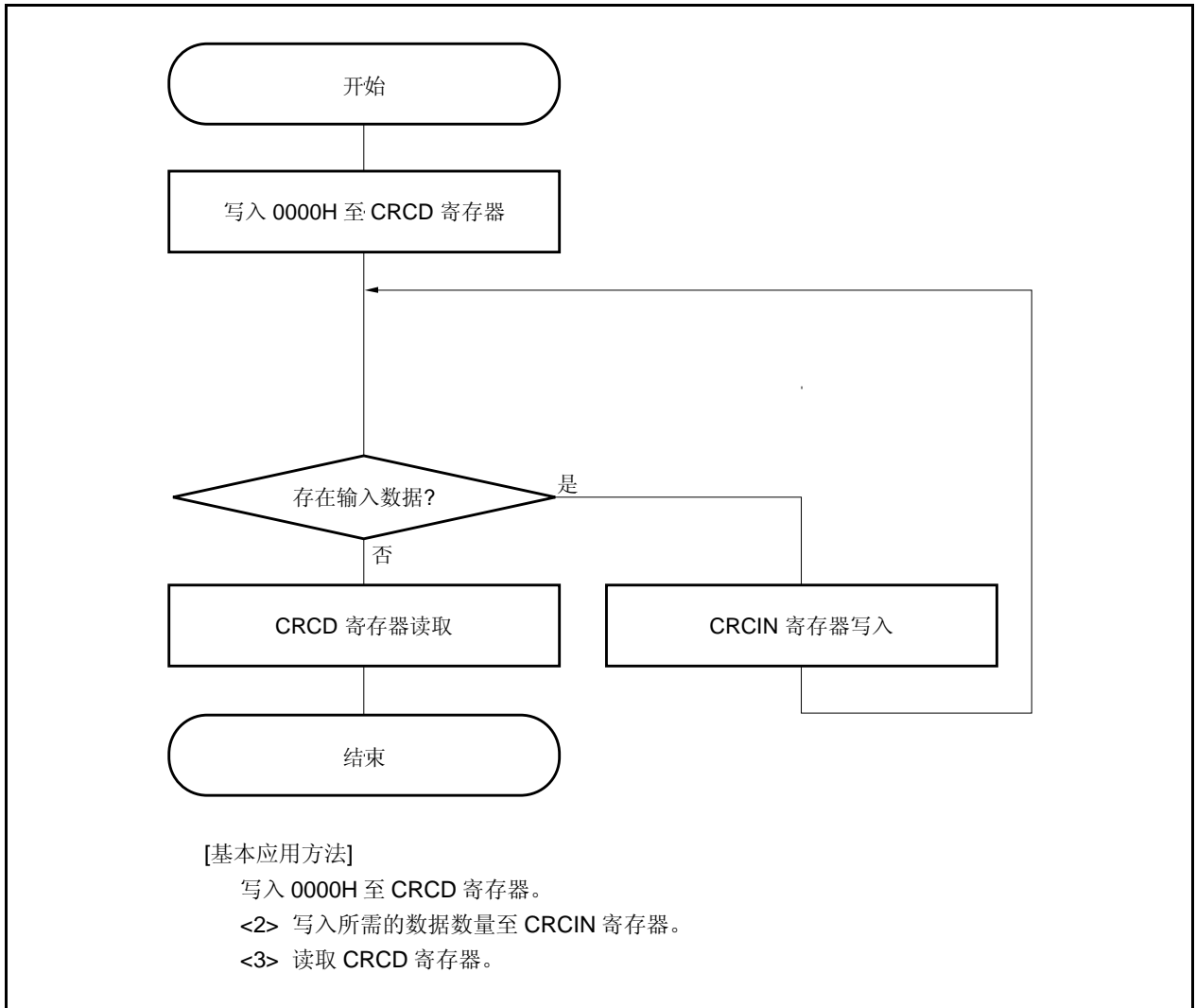


所以, CRC 码成为 $\overbrace{1001}^9 \overbrace{0001}^8 \overbrace{1000}^1 \overbrace{1000}^1$ 。由于使用 LSB 先行方式, 则对应 16 进制表示的 1189H。

29.5 使用方法

以下描述如何使用 CRC 逻辑电路。

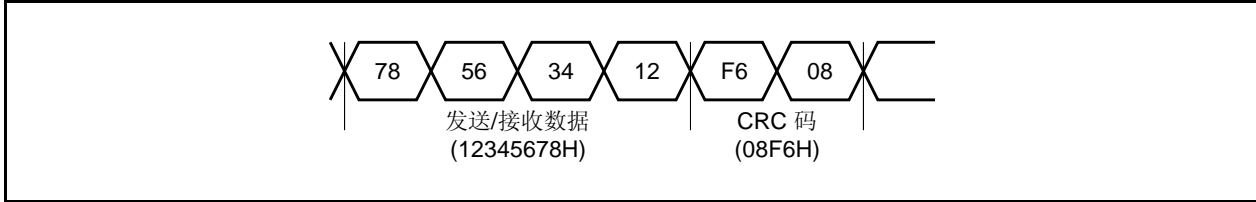
图 29-3. CRC 操作流程



当发送/接收数据由多个字节组成，如果同发送/接收数据一起发送/接收其 CRC 码，通信错误可以很容易得被检测出来。

下面以发送 12345678H (0001 0010 0011 0100 0101 0110 0111 1000B)且 LSB 先行为例，进行解释说明。

图 29-4. CRC 发送举例



发送端的设置过程。

- <1> 写入初始值 0000H 至 CRCD 寄存器。
- <2> 写入待发送的 1 字节数据至发送缓存寄存器。(与此同时，相同数据被写入 CRCIN 寄存器)
- <3> 当发送多字节数据时，每次发送数据被写入发送缓存寄存器时，写入相同数据至 CRCIN 寄存器。
- <4> 在所有数据已发送后，将 CRCD 寄存器(CRC 码)的内容写入发送缓存寄存器并进行发送。(由于是 LSB 先行发送，由低字节开始发送数据，然后才是高字节。)

接收端的设置过程。

- <1> 写入初始值 0000H 至 CRCD 寄存器。
- <2> 当第一个字节接收完毕时，将接收数据写入 CRCIN 寄存器。
- <3> 如果接收多个字节数据，每次接收完毕时都将接收数据写入 CRCIN 寄存器。(正常接收情况下，当所有接收数据都已写入 CRCIN 寄存器，接收端的 CRCD 寄存器的内容应该和发送端 CRCD 寄存器的内容相同。)
- <4> 然后，CRC 码从发送端发出，与接收数据相似，将该数据写入 CRCIN 寄存器。
- <5> 当接收完包括 CRC 码在内的所有数据后，如果 CRCD 寄存器的内容为 0000H，则接收正常。如果 CRCD 寄存器的内容不是 0000H，这表示发生通信错误，所以，向发送端发送重发请求。

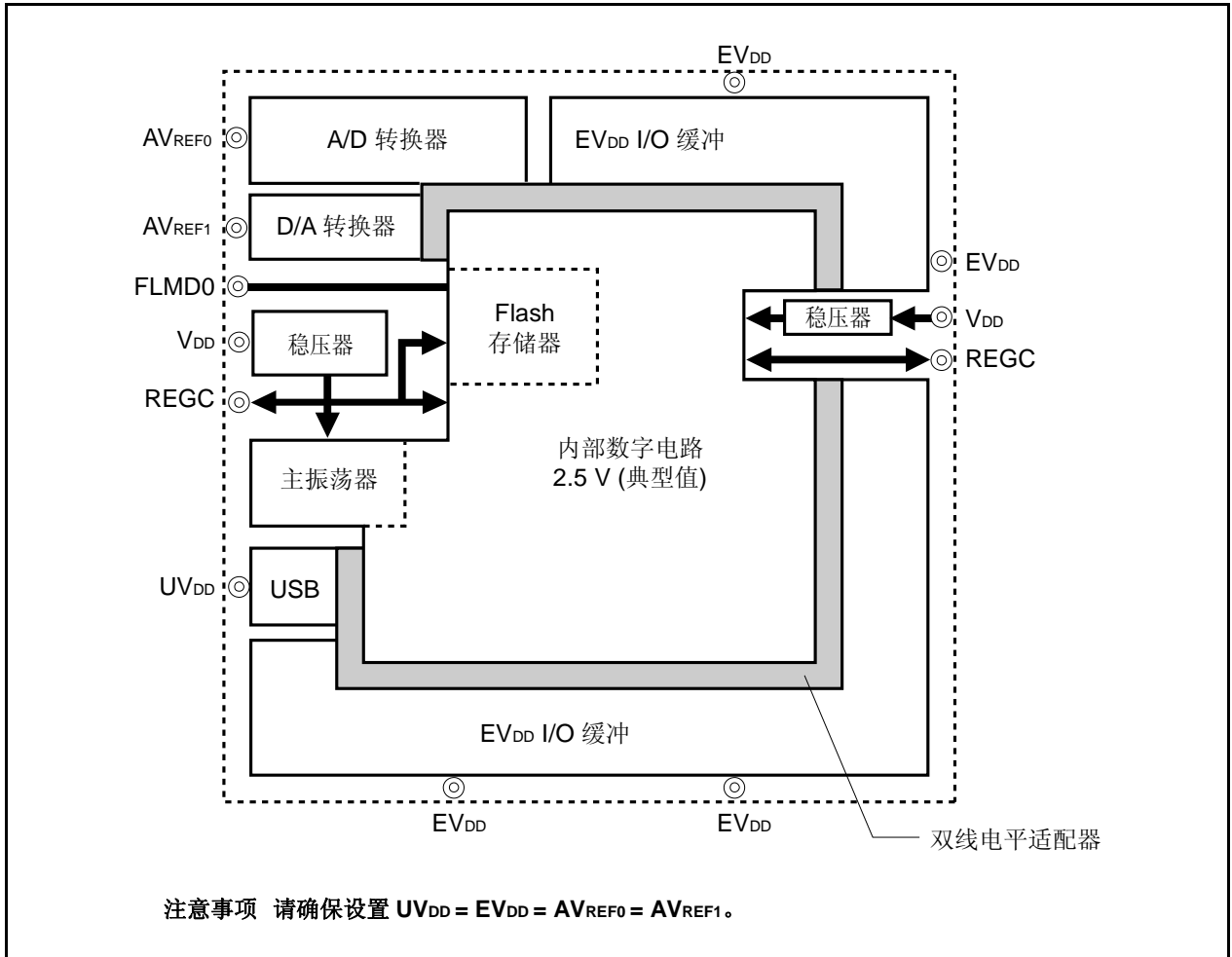
第三十章 稳压器

30.1 概述

V850ES/JG3-U 和 V850ES/JH3-U 包括一个稳压器，用于减少功率消耗和降低噪声。

该稳压器为振荡器模块和内部逻辑电路（除了 A/D 变换器，D/A 变换器和输出缓存器）提供一个阶梯下降的 V_{DD} 电源供电电压。

图 30-1. 稳压器



30.2 操作

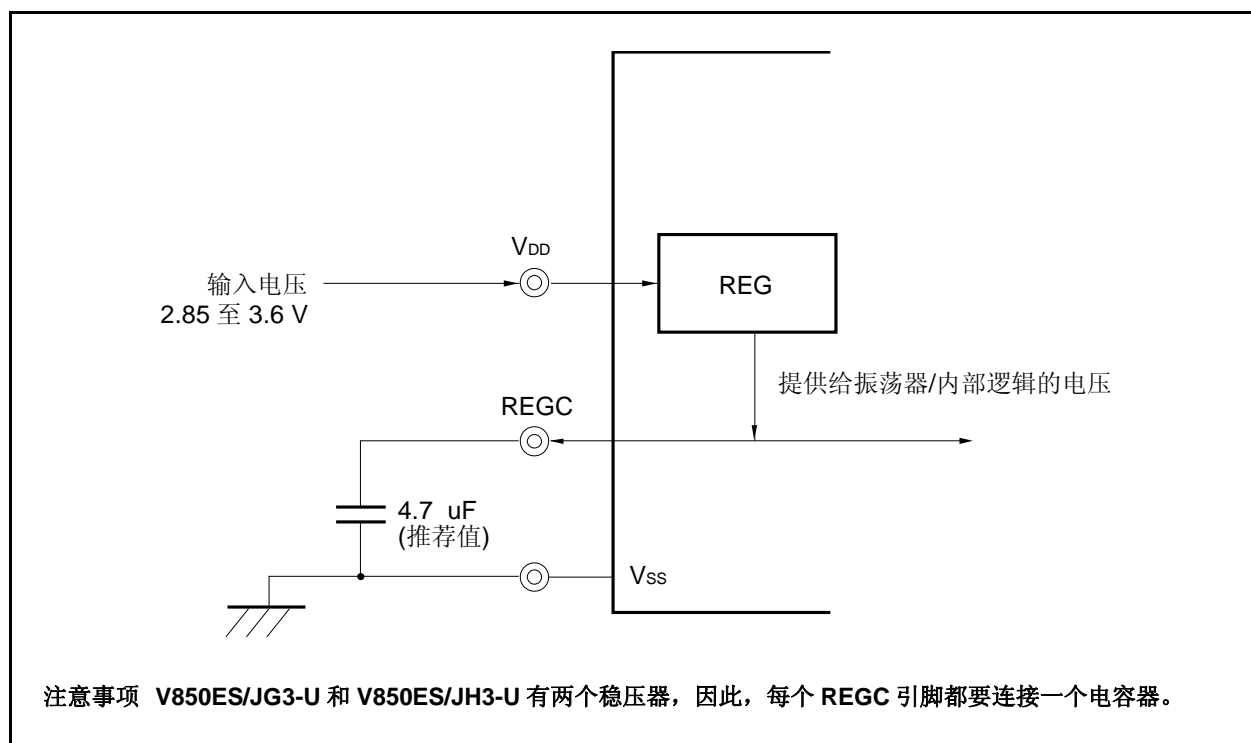
V850ES/JG3-U 和 V850ES/JH3-U 的稳压器能在任何模式(比如正常工作模式、HALT 模式、IDLE1 模式、IDLE2 模式、STOP 模式、副时钟运行模式、Sub-IDLE 模式或者复位期间)下保持运行。

请确保在 REGC 引脚^①接一个电容 (4.7 μ F (推荐值)) 来稳定稳压器的输出。

稳压器引脚的连接方式如下图所示。

注 有两个 REGC 引脚

图 30-2. REGC 引脚连接



第三十一章 FLASH存储器

V850ES/JG3-U 和 V850ES/JH3-U 单片机内置有一个 flash 存储器。

- μ PD70F3763, 70F3768: 384 KB flash 存储器
- μ PD70F3764, 70F3769: 512 KB flash 存储器

Flash 存储器版本为开发环境和批量生产应用提供以下的优势:

- V850ES/JG3-U 和 V850ES/JH3-U 被焊接到目标系统之后改换软件。
- 开始批量生产时的数据调整。
- 根据不同模式的小规模生产式样提供不同的软件。
- 便于库存管理。
- 出厂后更新软件。

31.1 特性

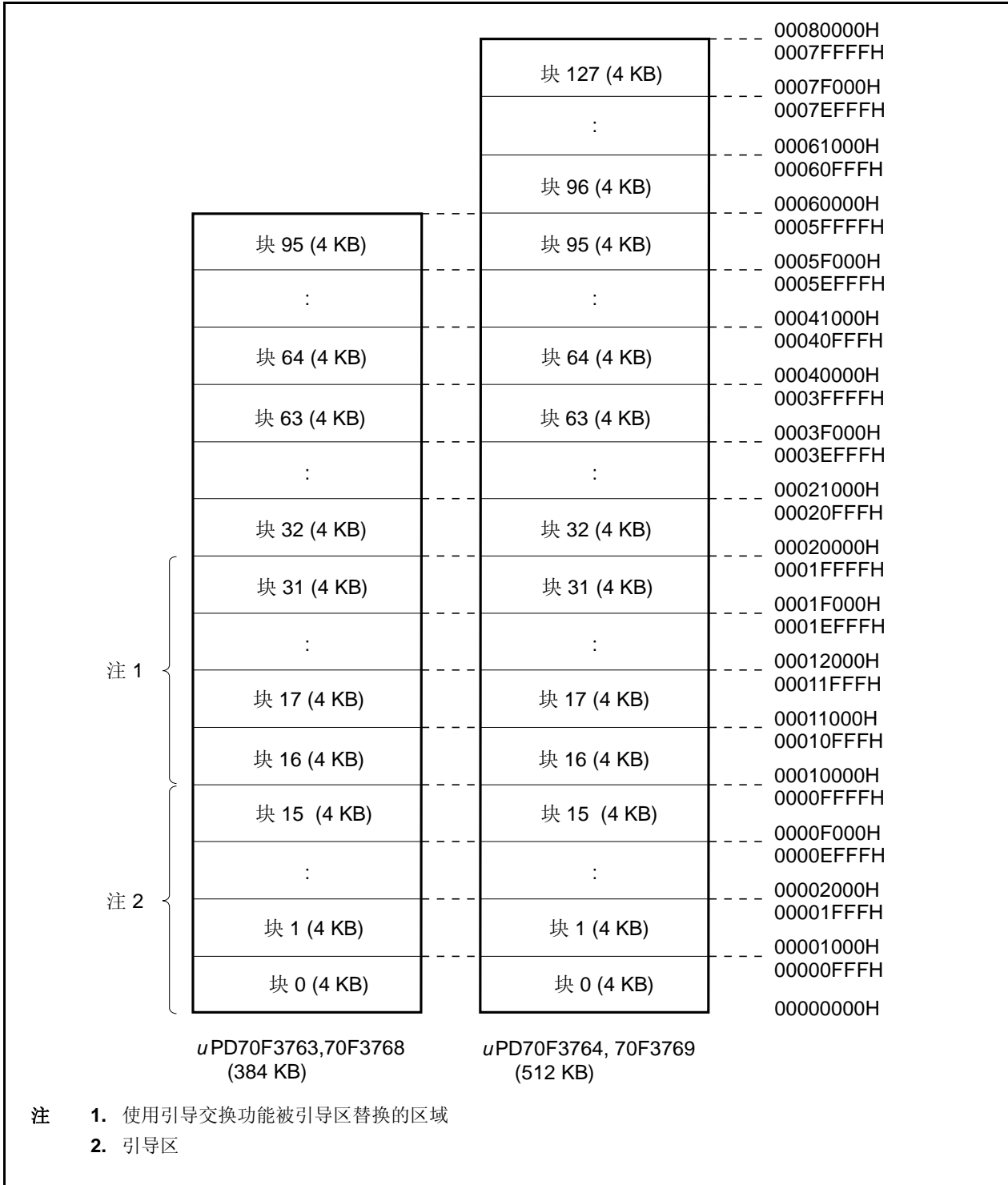
- 4-字节/1-时钟访问 (当读取指令时)
- 容量: 384/512 KB
- 重写电压: 用一个单供电电压进行擦除/写入
- 重写方式:
 - 通过串行接口使用 flash 存储器编程器进行通信重写(在线/离线编程)。
 - 通过用户程序重写 flash 存储器 (自编程)。
- 支持 flash 存储器禁止重写功能(安全功能)。
- 使用引导交换功能通过自编程方式对整个 flash 存储区安全重写。
- 自编程期间可以响应中断。

31.2 存储器配置

V850ES/JG3-U 和 V850ES/JH3-U 单片机内部 flash 存储区划分为 96 或 128 个块，且可以按块为单位进行编程/擦除。也可以一次性擦除所有的模块。

当使用引导交换功能时，块地址为 0 至 15 的物理存储区被块地址为 16 至 31 的物理存储区替换。关于引导交换功能的详情，参见 31.5 通过自编程重写。

图 31-1. Flash 存储器映射



31.3 功能概述

不管 V850ES/JG3-U 和 V850ES/JH3-U 是否被焊接到目标系统上（离线/在线编程），V850ES/JG3-U 和 V850ES/JH3-U 的内部 flash 存储器都可通过 flash 存储器编程器的重写功能进行重写。

此外，还支持安全功能，可以禁止对已经写入内部 flash 存储器的用户程序进行重写，因此，程序不能被未经授权者改变。

假设目标系统生产/出厂后改变程序，则使用用户程序（自编程）的重写功能是理想的选择。同样还支持安全重写整个 flash 存储器的引导交换功能。此外，自编程期间还支持中断服务，所以 flash 存储器可以在多种条件下进行重写，比如和外部设备通信时。

表 31-1. 重写方式

重写方式	功能概述	工作模式
在线编程	设备安装到目标系统之后，flash 存储器可使用 flash 存储器编程器重写。	Flash 存储器编程模式
离线编程	在设备安装到目标系统之前，flash 存储器可通过专用 flash 存储器编程器和专用程序适配板(FA 系列)重写。	
自编程	Flash 存储器可通过执行用户程序来重写，用户程序通过离线/在线编程预先写入 flash 存储器。（在自编程时，不能从内部 flash 存储器区取指令，也不能对内部 flash 存储器区进行数据存取。因此，重写程序必须预先传送到内部 RAM 或外部存储器。）	正常操作模式

备注 FA 系列为 Naito Densei Machida Mfg. Co., Ltd 的产品。

表 31-2. 基本功能

功能	功能概述	支持 (√: 可支持, ×: 不支持)	
		在线/离线编程	自编程
块擦除	指定存储块的内容被擦除。	√	√
片擦除	一次全部擦除整个存储器的内容。	√	×
编程	执行到指定地址的写入, 并校验检查写入的等级是否安全	√	√
校验/检查和	将从 flash 存储器读出的数据和由 flash 编程器发送的数据进行比较。	√	× (可由用户程序读取)
空白检测	检查整个存储器的擦除状态。	√	√
安全设置	禁止使用片擦除命令、块擦除命令、编程命令以及读取指令, 且禁止重写引导区。	√	× (仅当设置由使能变为禁止时, 可以支持)

下表列出了安全功能。出厂之后, 默认禁止片擦除命令、禁止块擦除命令以及禁止编程命令功能, 且可通过在线/离线编程进行重写来完成安全设置。各个安全功能可以同时与其它功能组合使用。

表 31-3. 安全功能

功能	功能概述
禁止块擦除命令	在所有块上禁止执行块擦除指令。关于禁止的设置可以通过执行片擦除指令来初始化。
禁止片擦除命令	在所有块上禁止执行片擦除命令和块擦除命令。一旦设定为禁止之后, 所有的关于禁止的设置都无法重新初始化设定, 因为片擦除命令不能执行。
禁止编程命令	所有模块上禁止执行编程命令和块擦除命令。关于禁止的设置可以通过执行片擦除命令来初始化。
禁止读取命令	所有块上禁止执行读取命令。关于禁止的设置可以通过执行片擦除命令来初始化。
禁止重写引导区	在引导区禁止执行块擦除命令和片擦除命令。一旦设定之后, 禁止重写引导区的设置无法重新初始化。

表 31-4. 安全设置

功能	设置各个安全性能后的擦除、写入、读取操作 (√: 可执行, ×: 不可执行, -: 不支持)		安全设置的注意事项	
	在线/离线编程	自编程	在线/离线编程	自编程
禁止块擦除命令	块擦除命令: × 片擦除命令: √ 编程命令: √ 读取命令: √	块擦除 (Flash 块擦除): √ 片擦除: - 写入(Flash 字写入): √ 读取(Flash 字读取): √	关于禁止的设置无法被初始化	仅当设置由使能变为禁止时, 可以支持自编程。
禁止片擦除命令	块擦除命令: × 片擦除命令: × 编程命令: √ ^{注1} 读取命令: √	块擦除 (Flash 块擦除): √ 片擦除: - 写入(Flash 字写入): √ 读取(Flash 字读取): √	关于禁止的设置可以通过片擦除命令进行初始化	
禁止编程命令	块擦除命令: × 片擦除命令: √ 编程命令: × 读取命令: √	块擦除 (Flash 块擦除): √ 片擦除: - 写入(Flash 字写入): √ 读取(Flash 字读取): √	关于禁止的设置可以通过片擦除命令进行初始化	
禁止读取命令	块擦除命令: √ 片擦除命令: √ 编程命令: √ 读取命令: ×	块擦除 (Flash 块擦除): √ 片擦除: - 写入(Flash 字写入): √ 读取(Flash 字读取): √	关于禁止的设置可以通过片擦除命令进行初始化	
禁止重写引导区	块擦除命令: × ^{注2} 片擦除命令: × 编程命令: × ^{注2} 读取命令: √	块擦除 (Flash 块擦除): × ^{注2} 片擦除: - 写入(Flash 字写入): √ ^{注2} 读取(Flash 字读取): √	关于禁止的设置无法被初始化	

- 注
1. 在这种情况下, 因为擦除命令无效, 与已写入数据不同的数据将无法被写入 flash 存储器。
 2. 在引导区之外可以执行。
 3. 在 reset 输入后, 引导区重写禁止功能开始生效。

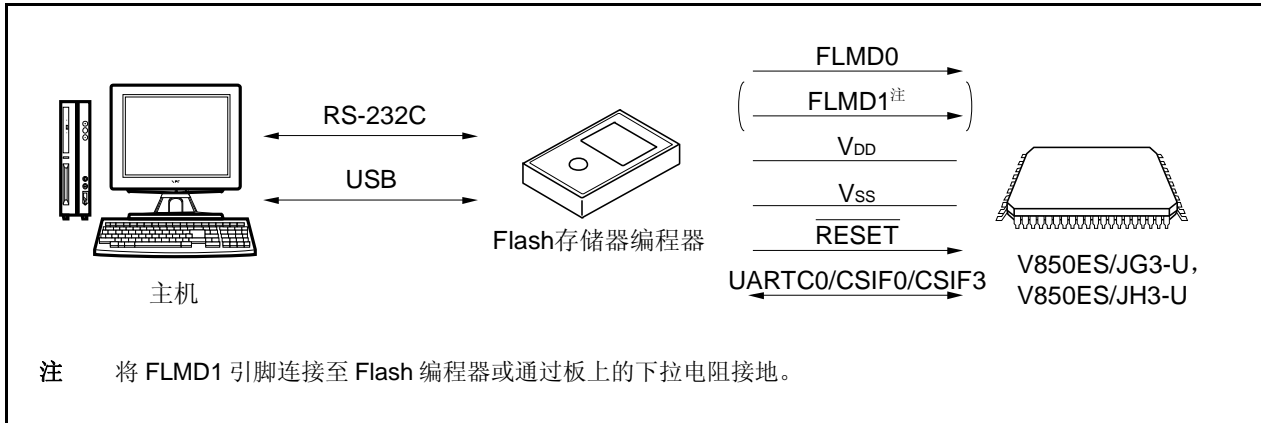
31.4 通过flash存储器编程器进行重写

V850ES/JG3-U 和 V850ES/JH3-U 被安装到目标系统（在线编程）之后，flash 存储器可通过 flash 存储器编程器进行重写。Flash 存储器也可以使用专用的编程适配器（FA 系列）在芯片安装到目标系统前进行重写（离线编程）。

31.4.1 编程环境

下面显示了将程序写入 V850ES/JG3-U 和 V850ES/JH3-U 的 flash 存储器所需的环境。

图 31-2. 程序写入 flash 存储器所需的环境



需要一台主机用来控制 flash 存储器编程器。

UARTC0, CSIF0 或 CSIF3 被用作 flash 存储器编程器和 V850ES/JG3-U 和 V850ES/JH3-U 之间的接口，执行写入、擦除等操作。离线编程需要一个专用编程适配器 (FA 系列)。

备注 FA 系列为 Naito Densai Machida Mfg. Co., Ltd 的产品。

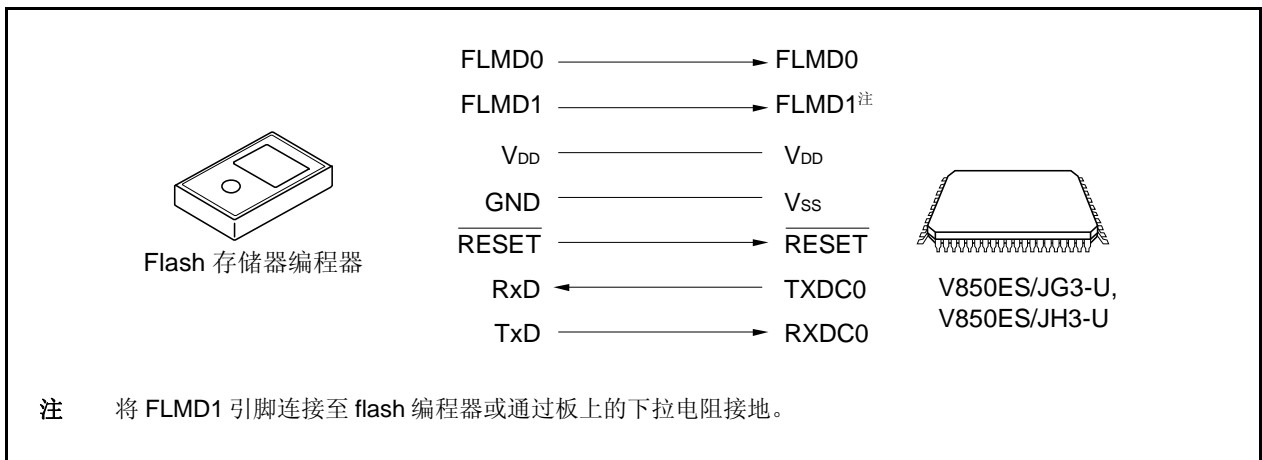
31.4.2 通信模式

flash 存储器编程器和 V850ES/JG3-U 和 V850ES/JH3-U 的通信方式是串行通信，使用 V850ES/JG3-U 和 V850ES/JH3-U 的 UARTC0，CSIF0 或 CSIF3 接口。

(1) UARTC0

传输速率： 9,600 至 153,600 bps

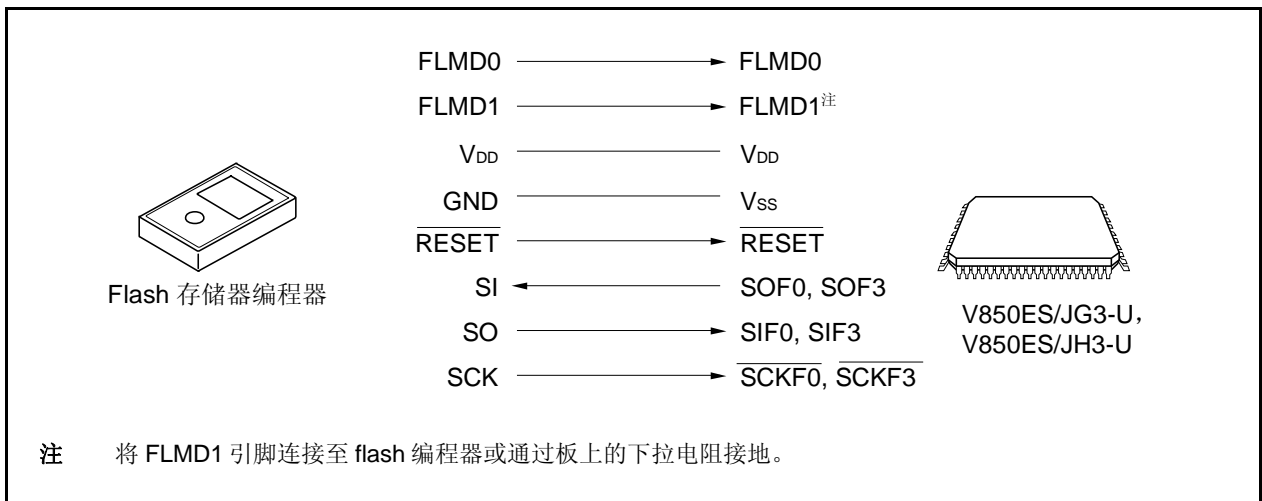
图 31-3. 和 flash 存储器编程器通信(UARTC0)



(2) CSIF0, CSIF3

串行时钟： 5 MHz 或更低 (MSB 先行)

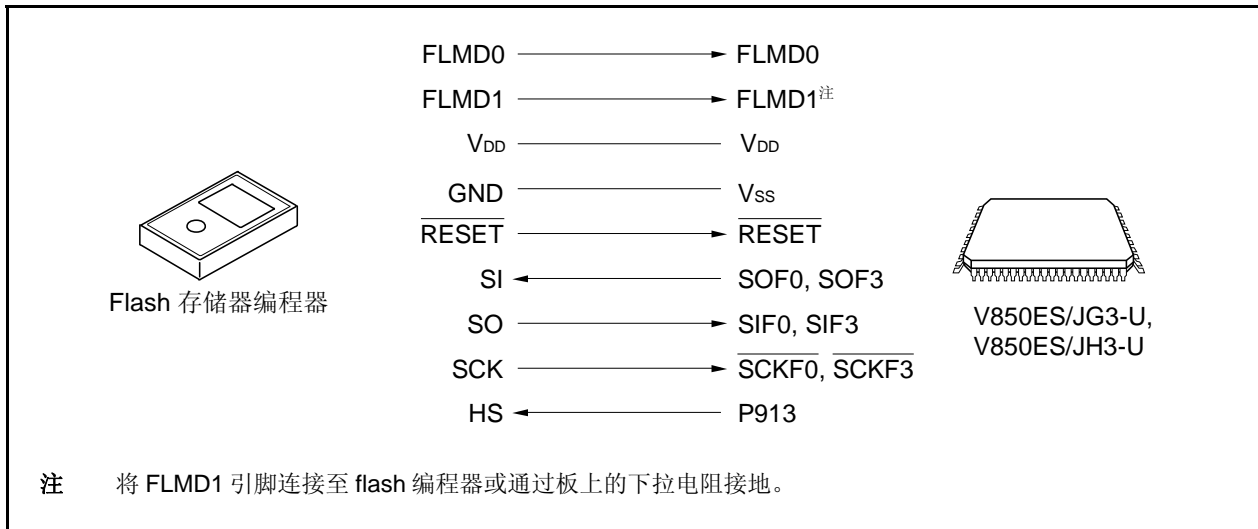
图 31-4. 和 flash 存储器编程器通信(CSIF0, CSIF3)



(3) CSIF0 + HS, CSIF3 + HS

串行时钟: 5 MHz 或更低 (MSB 先行)

图 31-5. 和 flash 存储器编程器通信(CSIF0 + HS, CSIF3 + HS)



flash 存储器编程器输出串行时钟, V850ES/JG3-U 和 V850ES/JH3-U 作为从机工作。

当 flash 存储器编程器使用 PG-FP5 时, 会产生下面的信号到 V850ES/JG3-U 和 V850ES/JH3-U。详情参见 PG-FP5 用户手册(U18865E)。

表 31-5. flash 存储器编程器的信号连接 (PG-FP5)

PG-FP5			V850ES/JG3-U 和 V850ES/JH3-U	连接处理		
信号名称	I/O	引脚功能	引脚名称	UARTC0	CSIF0, CSIF3	CSIF0 + HS, CSIF3 + HS
FLMD0	输出	写入使能/禁止	FLMD0	○	○	○
FLMD1	输出	写入使能/禁止	FLMD1	○ ^{注1}	○ ^{注1}	○ ^{注1}
VDD	-	V _{DD} 电压产生/电压监控	V _{DD}	○	○	○
GND	-	地	V _{SS}	○	○	○
CLK	输出	输出至 V850ES/JG3-U 和 V850ES/JH3-U 的时钟	X1, X2	x ^{注2}	x ^{注2}	x ^{注2}
RESET	输出	复位信号	RESET	○	○	○
SI/RxD	输入	接收信号	SOF0, SOF3/TXDC0	○	○	○
SO/TxD	输出	发送信号	SIF0, SIF3/RXDC0	○	○	○
SCK	输出	传输时钟	SCKF0, SCKF3	x	○	○
HS	输入	CSIF0 + HS, CSIF3 + HS 通信的握手信号	P913	x	x	○

- 注**
1. 将这些引脚按图 31-6 和图 31-7 所示进行连接，或通过板上的下拉电阻将其接地。
 2. 不能通过 flash 存储器编程器的 CLK 引脚提供时钟。在电路板上安装一个振荡器来提供时钟。

备注

○：必须进行连接。
x：不必进行连接。

表 31-6. V850ES/JG3-U Flash 写入适配器 (FA-100GC-UEU-B) 的连线 (1/2)

Flash 存储器编程器(FG-FP5)连接 引脚			FA 板的引脚 名称	使用 CSIF0 + HS		使用 CSIF0		使用 UARTC0	
信号名称	输入/输出	引脚功能		引脚名称	引脚编号	引脚名称	引脚编号	引脚名称	引脚编号
SI/RxD	输入	接收信号	SI	P41/SOF0/ SCL01	23	P41/SOF0/ SCL01	23	P30/TXDC0/ SOF4	25
SO/TxD	输出	发送信号	SO	P40/SIF0/ SDA01	22	P40/SIF0/ SDA01	22	P31/RXDC0/ INTP07/SIF4	26
SCK	输出	传输时钟	SCK	P42/ $\overline{\text{SCKF0}}$	24	P42/ $\overline{\text{SCKF0}}$	24	不需要	-
CLK	输出	至 V850ES /JG3-U 的时钟	X1	不需要	-	不需要	-	不需要	-
			X2	不需要	-	不需要	-	不需要	-
/RESET	输出	复位信号	/RESET	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14
FLMD0	输出	写入电压	FLMD0	FLMD0	8	FLMD0	8	FLMD0	8
FLMD1	输出	写入电压	FLMD1	PDL5/AD5/ FLMD1	76	PDL5/AD5/ FLMD1	76	PDL5/AD5/ FLMD1	76
HS	输入	CSI0 + HS 通信 的握手信号	RESERVE/ HS	P913	55	不需要	-	不需要	-
VDD	-	产生 VDD 电压/ 电压监控	VDD	VDD	9, 60	VDD	9, 60	VDD	9, 60
				EVDD	34, 63, 80	EVDD	34, 63, 80	EVDD	34, 63, 80
				UVDD	19	UVDD	19	UVDD	19
				AVREF0	1	AVREF0	1	AVREF0	1
				AVREF1	5	AVREF1	5	AVREF1	5
GND	-	地	GND	VSS	11, 33, 62, 7	VSS	11, 33, 62, 79	VSS	11, 33, 62, 79
				AVSS	2	AVSS	2	AVSS	2

- 注意事项 1. 请确保通过 4.7 μF (推荐值) 电容器将 REGC 引脚接地。
2. 不能通过 flash 存储器编程器的 CLK 引脚提供时钟。在电路板上安装一个振荡器来提供时钟。

表 31-6. V850ES/JG3-U Flash 写入适配器 (FA-100GC-UEU-B) 的连线 (2/2)

Flash 存储器编程器(FG-FP5)连接引脚			FA 板的引脚 名称	使用 CSIF3 + HS		使用 CSIF3	
信号名称	输入/输出	引脚功能		引脚名称	引脚编号	引脚名称	引脚编号
SI/RxD	输入	接收信号	SI	P911/A11/SOF3	53	P911/A11/SOF3	53
SO/TxD	输出	发送信号	SO	P910/A10/SIF3	52	P910/A10/SIF3	52
SCK	输出	传输时钟	SCK	P912/A12/ $\overline{\text{SCKF3}}$	54	P912/A12/ $\overline{\text{SCKF3}}$	54
CLK	输出	至 V850ES /JG3-U 的时钟	X1	不需要	-	不需要	-
			X2	不需要	-	不需要	-
/RESET	输出	复位信号	/RESET	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14
FLMD0	输出	写入电压	FLMD0	FLMD0	8	FLMD0	8
FLMD1	输出	写入电压	FLMD1	PDL5/AD5/FLMD1	76	PDL5/AD5/FLMD1	76
HS	输入	CSI0 + HS 通信 的握手信号	RESERVE/HS	P913	55	不需要	-
VDD	-	产生 VDD 电压/ 电压监控	VDD	V _{DD}	9, 60	V _{DD}	9, 60
				E _{VDD}	34, 63, 80	E _{VDD}	34, 63, 80
				U _{VDD}	19	U _{VDD}	19
				A _{VREF0}	1	A _{VREF0}	1
				A _{VREF1}	5	A _{VREF1}	5
GND	-	地	GND	V _{SS}	11, 33, 62, 79	V _{SS}	11, 33, 62, 79
				A _{VSS}	2	A _{VSS}	2

- 注意事项 1. 请确保通过 4.7 μ F (推荐值) 电容器将 REGC 引脚接地。
2. 不能通过 flash 存储器编程器的 CLK 引脚提供时钟。在电路板上安装一个振荡器来提供时钟。

图 31-6. V850ES/JG3-U Flash 写入适配器(FA-100GC-UEU-B)的连线示例
(CSIF3 + HS 模式) (1/2)

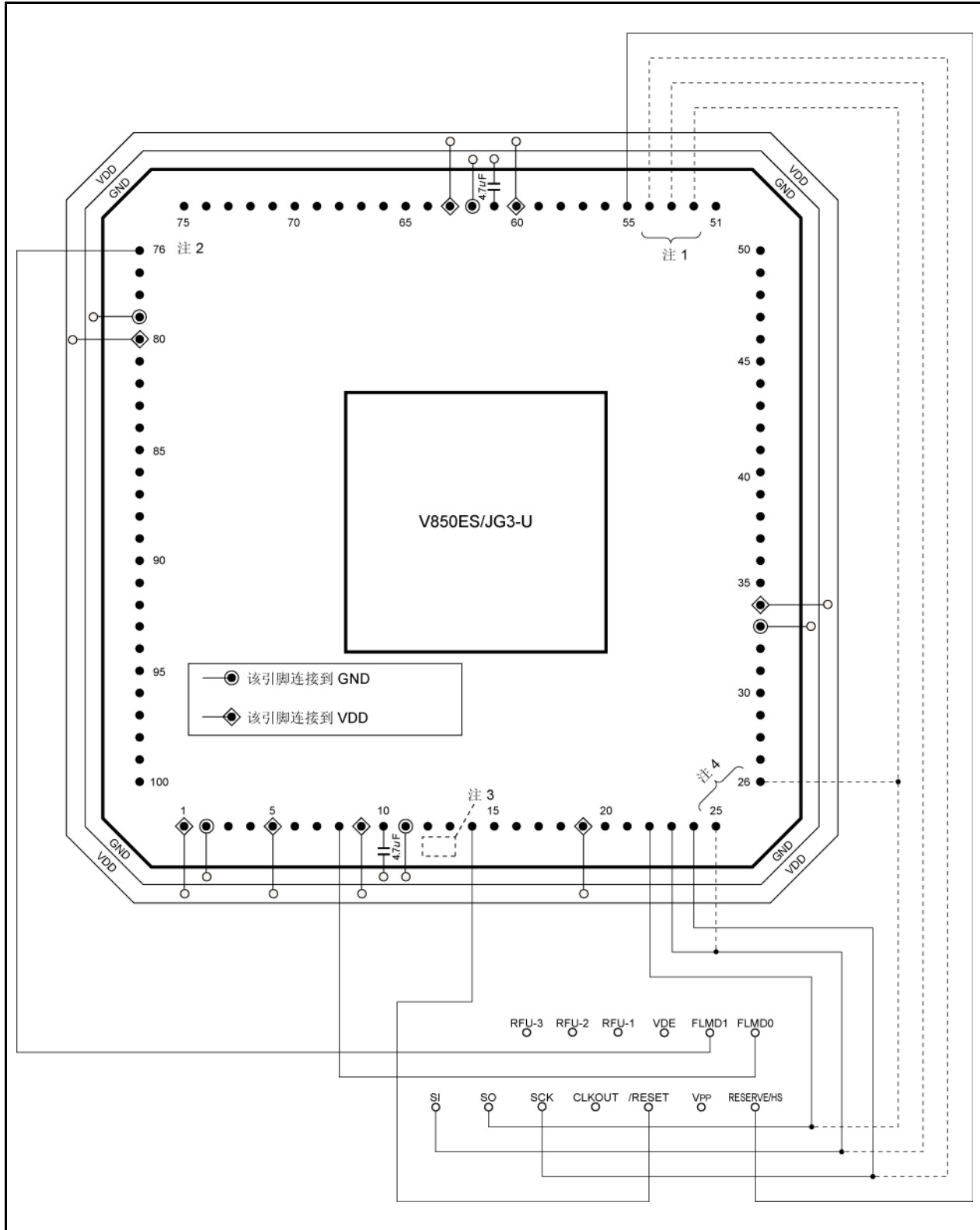
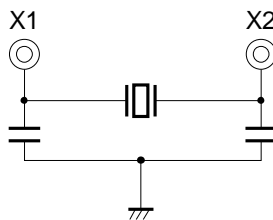


图 31-6. V850ES/JG3-U Flash 写入适配器(FA-100GC-UEU-B)的连线示例
(CSIF3 + HS 模式) (2/2)

- 注
1. 当使用 CSIF3 时对应的引脚。
 2. FLMD1 引脚的接线如下所示，或通过板上的下拉电阻将其接地。
 3. 在 flash 写入适配器上安装一个振荡器 (如虚线所示)并提供时钟。这里给出一个振荡器的例子。

示例:



4. 对应使用 UARTC0 时的引脚。

注意事项 不要输入高电平到 $\overline{\text{DRST}}$ 引脚。

- 备注
1. 对于未列出的引脚的处理，按照未使用引脚的处理方法。(参见 2.3 引脚 I/O 电路类型、I/O 缓冲器电源以及未使用引脚的连接)。
 2. 该适配器适用于 100-针塑封 LQFP。

表 31-7. V850ES/JH3-U Flash 写入适配器 (FA-128GF-GAT-B) 的连线 (1/2)

Flash 存储器编程器(FG-FP5)连接 引脚			FA 板的引脚 名称	使用 CSIF0 + HS		使用 CSIF0		使用 UARTC0	
信号名称	输入/输出	引脚功能		引脚名称	引脚编号	引脚名称	引脚编号	引脚名称	引脚编号
SI/RxD	输入	接收信号	SI	P41/SOF0/ SCL01	30	P41/SOF0/ SCL01	30	P30/TXDC0/ SOF4	37
SO/TxD	输出	发送信号	SO	P40/SIF0/ SDA01	29	P40/SIF0/ SDA01	29	P31/RXDC0/I NTP07/SIF4	38
SCK	输出	传输时钟	SCK	P42/ $\overline{\text{SCKF0}}$	31	P42/ $\overline{\text{SCKF0}}$	31	不需要	-
CLK	输出	至 V850ES /JH3-U 的时钟	X1	不需要	-	不需要	-	不需要	-
			X2	不需要	-	不需要	-	不需要	-
/RESET	输出	复位信号	/RESET	RESET	18	RESET	18	$\overline{\text{RESET}}$	18
FLMD0	输出	写入电压	FLMD0	FLMD0	12	FLMD0	12	FLMD0	12
FLMD1	输出	写入电压	FLMD1	PDL5/AD5/ FLMD1	103	PDL5/AD5/ FLMD1	103	PDL5/AD5/ FLMD1	103
HS	输入	CSIF0 + HS 通信 的握手信号	RESERVE/ HS	P913	69	不需要	-	不需要	-
VDD	-	产生 VDD 电压/ 电压监控	VDD	VDD	13, 82	VDD	13, 82	VDD	13, 82
				EVDD	46, 61, 85, 107	EVDD	46, 61, 85, 107	EVDD	46, 61, 85, 107
				UVDD	23	UVDD	23	UVDD	23
				AVREF0	1	AVREF0	1	AVREF0	1
				AVREF1	5	AVREF1	5	AVREF1	5
GND	-	地	GND	VSS	15, 45, 60, 84, 106	VSS	15, 45, 60, 84, 106	VSS	15, 45, 60, 84, 106
				AVSS	2	AVSS	2	AVSS	2

- 注意事项 1. 请确保通过 4.7 μF (推荐值) 电容器将 REGC 引脚接地。
2. 不能通过 flash 存储器编程器的 CLK 引脚提供时钟。在电路板上安装一个振荡器来提供时钟。

表 31-7. V850ES/JH3-U Flash 写入适配器 (FA-128GF-GAT-B) 的连线 (2/2)

Flash 存储器编程器(FG-FP5)连接引脚			FA 板的引脚 名称	使用 CSIF3 + HS		使用 CSIF3	
信号名称	输入/输出	引脚功能		引脚名称	引脚编号	引脚名称	引脚编号
SI/RxD	输入	接收信号	SI	P911/A11/SOF3	67	P911/A11/SOF3	67
SO/TxD	输出	发送信号	SO	P910/A10/SIF3	66	P910/A10/SIF3	66
SCK	输出	传输时钟	SCK	P912/A12/SCKF3	68	P912/A12/SCKF3	68
CLK	输出	至 V850ES /JH3-U 的时钟	X1	不需要	-	不需要	-
			X2	不需要	-	不需要	-
/RESET	输出	复位信号	/RESET	RESET	16	RESET	16
FLMD0	输出	写入电压	FLMD0	FLMD0	10	FLMD0	10
FLMD1	输出	写入电压	FLMD1	PDL5/AD5/FLMD1	103	PDL5/AD5/FLMD1	78
HS	输入	CSI0 + HS 通信 的握手信号	RESERVE/HS	P913	69	不需要	-
VDD	-	产生 VDD 电压/ 电压监控	VDD	V _{DD}	13, 82	V _{DD}	13, 82
				E _{VDD}	46, 61, 85, 107	E _{VDD}	46, 61, 85, 107
				U _{VDD}	23	U _{VDD}	23
				A _{VREF0}	1	A _{VREF0}	1
				A _{VREF1}	5	A _{VREF1}	5
GND	-	地	GND	V _{SS}	15, 45, 60, 84, 106	V _{SS}	15, 45, 60, 84, 106
				A _{VSS}	2	A _{VSS}	2

- 注意事项 1. 请确保通过 4.7 μ F (推荐值) 电容器将 REGC 引脚接地。
2. 不能通过 flash 存储器编程器的 CLK 引脚提供时钟。在电路板上安装一个振荡器来提供时钟。

图 31-7. V850ES/JH3-U Flash 写入适配器(FA-128GF-GAT-B)的连线示例
(CSIF0 + HS 模式) (1/2)

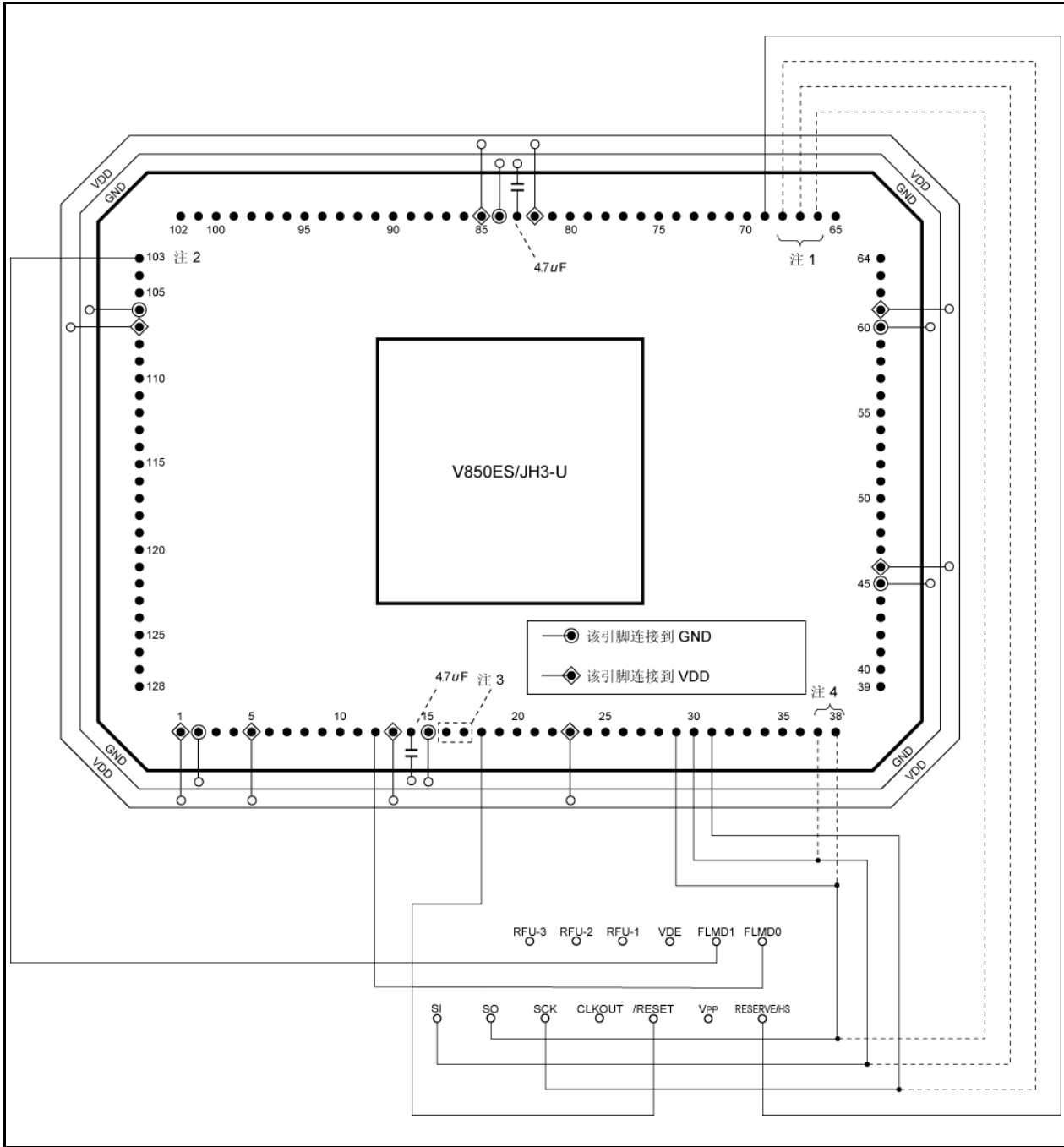
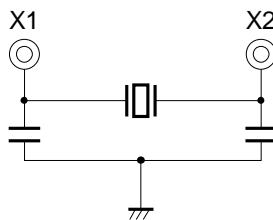


图 31-7. V850ES/JH3-U Flash 写入适配器(FA-128GF-GAT-B)的连线示例
(CSIF0 + HS 模式) (2/2)

- 注
1. 当使用 CSIF3 时对应的引脚。
 2. FLMD1 引脚的接线如下所示，或通过板上的下拉电阻将其接地。
 3. 在 flash 写入适配器上安装一个振荡器 (如虚线所示)并提供时钟。这里给出一个振荡器的例子。

示例:



4. 对应使用 UARTC0 时的引脚。

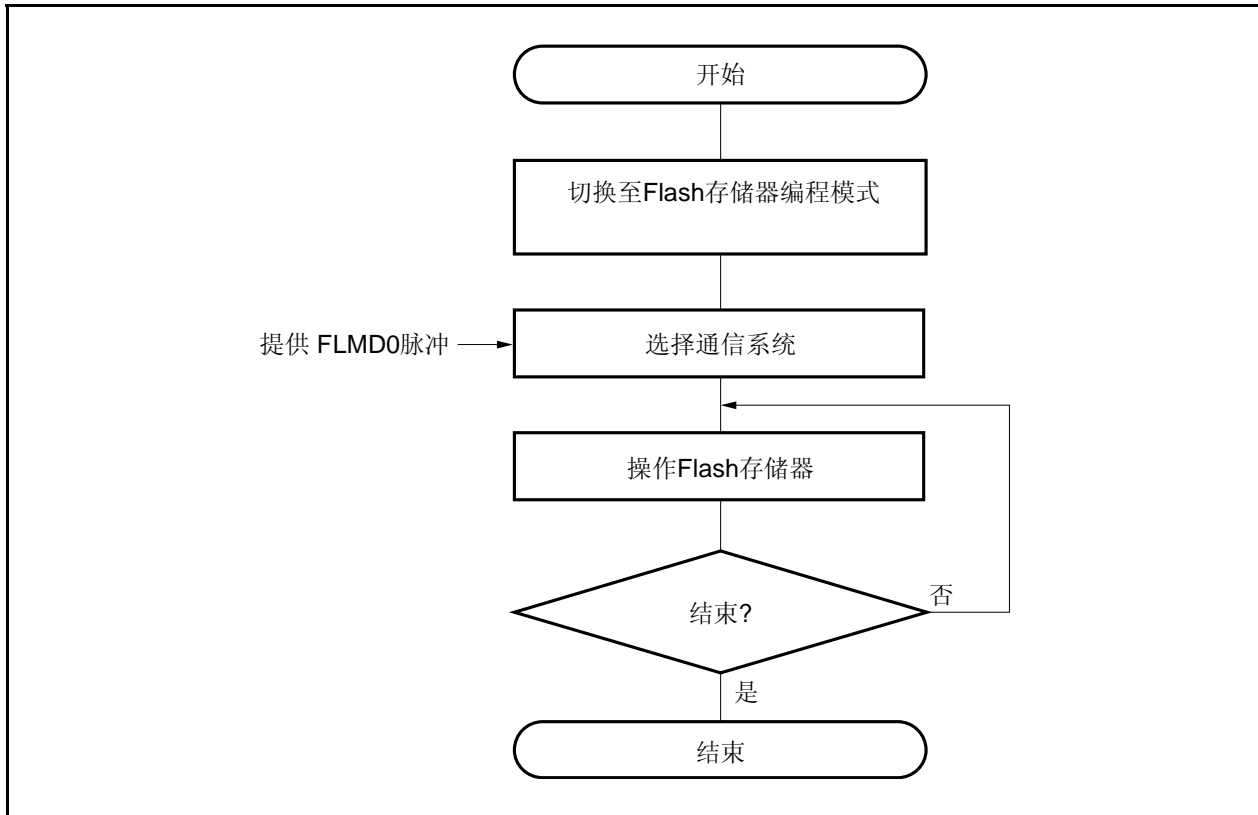
注意事项 不要输入高电平到 $\overline{\text{DRST}}$ 引脚。

- 备注
1. 对于未列出的引脚的处理，按照未使用引脚的处理方法。(参见 2.3 引脚 I/O 电路类型、I/O 缓冲器电源以及未使用引脚的连接)。
 2. 该适配器适用于 128-针塑封 LQFP。

31.4.3 Flash存储器控制

以下显示了操作 flash 存储器的步骤。

图 31-8. 操作 flash 存储器的步骤

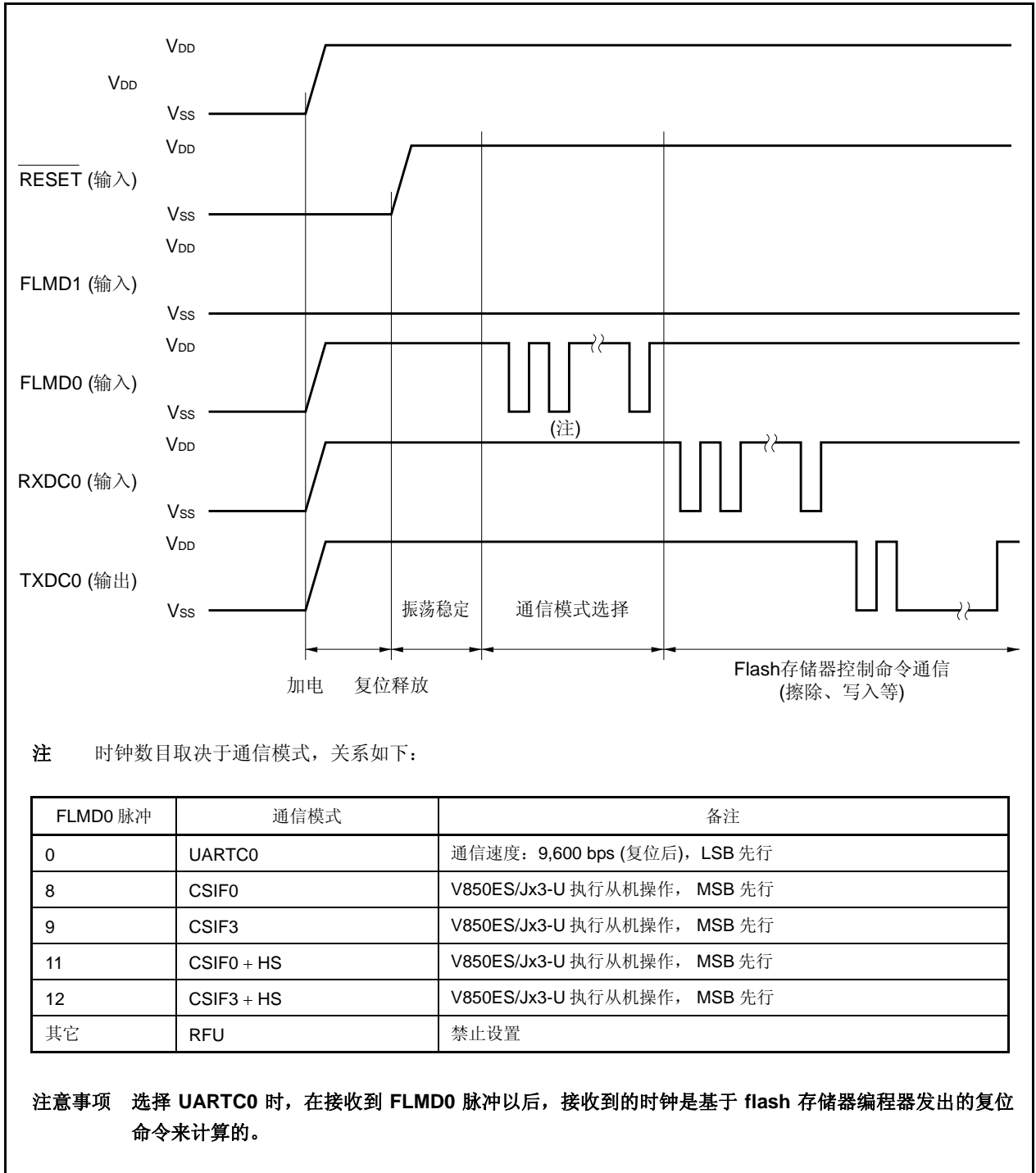


31.4.4 通信模式的选择

在 V850ES/JG3-U 和 V850ES/JH3-U 单片机中，切换至 flash 存储器编程模式后，通过输入脉冲（最多 12 个脉冲）至 FLMD0 引脚来选择通信模式。FLMD0 脉冲由 flash 存储器编程器产生。

下面显示了脉冲数目和通信模式之间的关系。

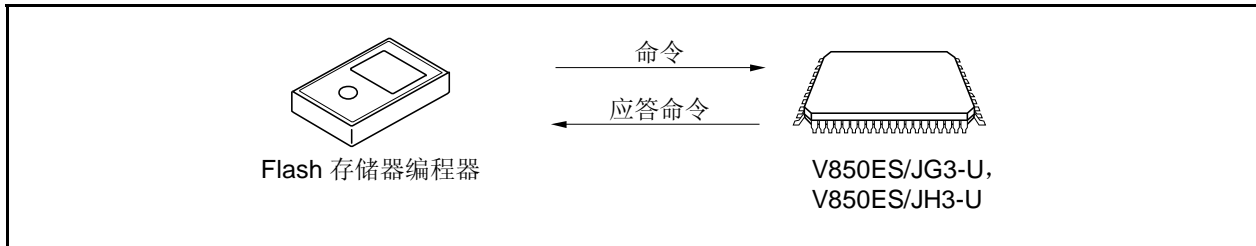
图 31-9. 通信模式的选择



31.4.5 通信命令

V850ES/JG3-U 和 V850ES/JH3-U 通过指令的方式与 flash 存储器编程器进行通信。从 flash 存储器编程器发至 V850ES/JG3-U 和 V850ES/JH3-U 的信号称为“命令”。从 V850ES/JG3-U 和 V850ES/JH3-U 发出至 flash 存储器编程器的应答信号称为“应答命令”。

图 31-10. 通信命令



下面显示了 V850ES/JG3-U 和 V850ES/JH3-U 用来控制 flash 存储器的命令。所有的这些命令都由 flash 存储器编程器产生，而 V850ES/JG3-U 和 V850ES/JH3-U 执行与命令相应的处理。

表 31-7. Flash 存储器控制命令

类别	命令名称	支持			功能
		CSIF0, CSIF3	CSIF0 + HS, CSIF3 + HS	UARTC0	
空白检查	块空白检查命令	√	√	√	检查指定块的存储内容是否已被正确擦除。
擦除	片擦除命令	√	√	√	擦除整个存储器的内容。
	块擦除命令	√	√	√	擦除指定块的存储内容。
写入	编程命令	√	√	√	写入指定的地址范围，且执行内容检验检查。
校验	校验命令	√	√	√	比较存储器中指定地址范围的内容和 flash 编程器传送的数据
	检查和命令	√	√	√	读取指定地址范围内的检查和。
系统设置控制	硅签字命令	√	√	√	读取硅签字信息。
	安全设置命令	√	√	√	禁止片擦除命令、块擦除命令、编程命令、读取命令和引导区重写。

31.4.6 引脚连接

当执行在线写入时，将目标系统上安装一个连接器用以连接 flash 存储器编程器。另外，电路板上还要具有将正常操作模式切换至 flash 存储器编程模式的功能。

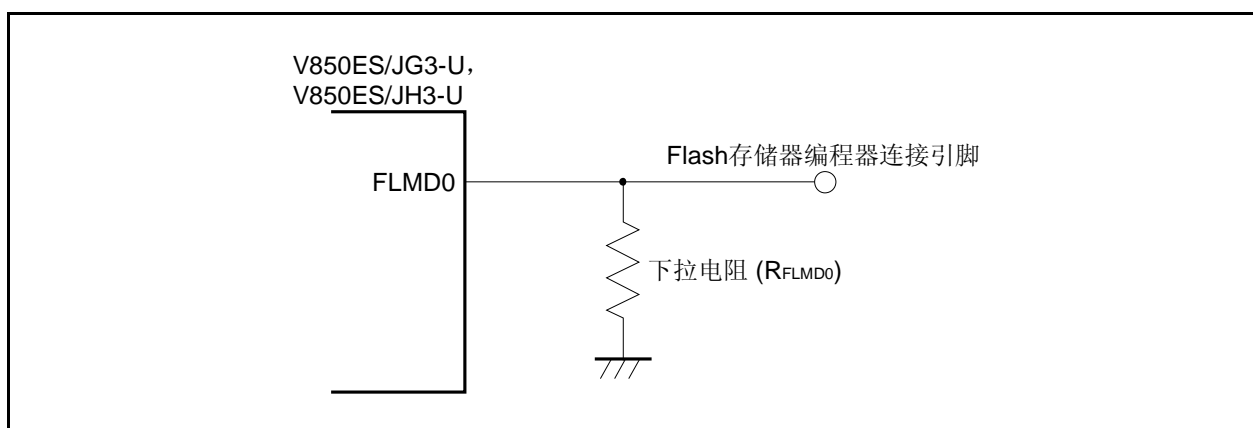
在 flash 存储器编程模式下，所有在 flash 存储器编程过程中未使用的引脚，其状态与复位后的瞬时状态相同。因此，当外部设备不能识别复位后的瞬时状态时，需要对引脚进行处理。

(1) FLMD0 引脚

在正常操作模式下，将电压值为 V_{SS} 的电平输入至 FLMD0 引脚。在 flash 存储器编程模式，为 FLMD0 引脚提供电平为 V_{DD} 的写入电压。

因为在自编程模式时，FLMD0 引脚被用作写入保护引脚，因此在写入 flash 存储器之前，必须通过端口控制等手段给 FLMD0 引脚提供一个电平为 V_{DD} 的电压。详情参见 31.5.5 (1) FLMD0 引脚。

图 31-11. FLMD0 引脚连接示例



(2) FLMD1 引脚

当 FLMD0 引脚输入 0V 时，FLMD1 引脚不起作用。当给 FLMD0 引脚提供 V_{DD} 电平时，进入 flash 存储器编程模式，因此必须给 FLMD1 引脚输入 0V。下面是 FLMD1 引脚的一个连接示例。

图 31-12. FLMD1 引脚连接示例

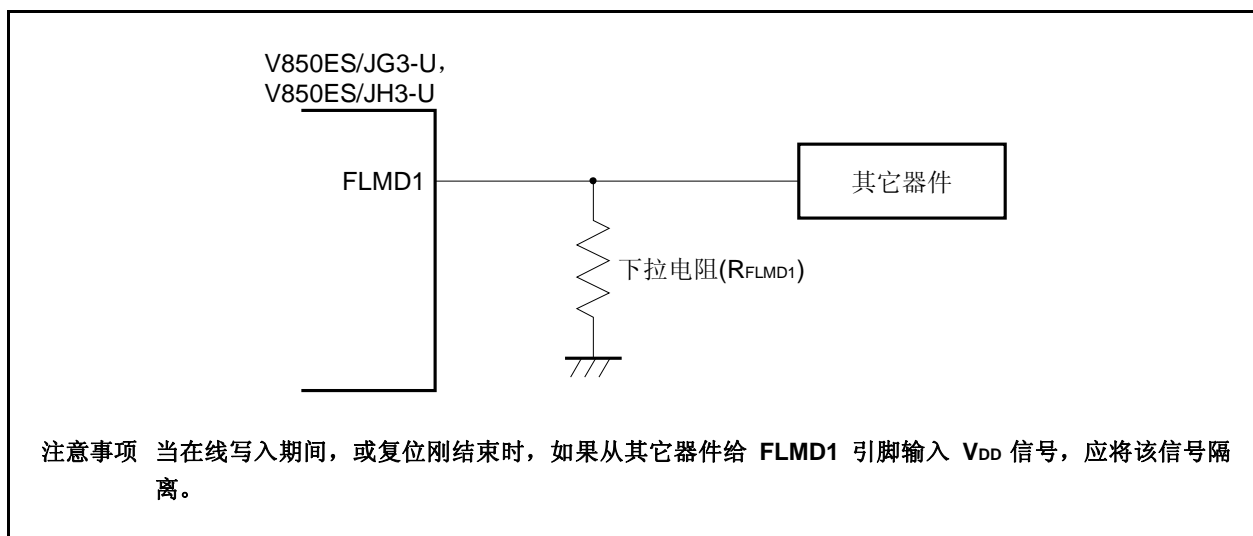


表 31-8. 复位解除后 FLMD0 和 FLMD1 引脚与操作模式之间的关系

FLMD0	FLMD1	操作模式
0	无须理会	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

下面给出各个串行接口使用的引脚。

表 31-9. 串行接口所用引脚

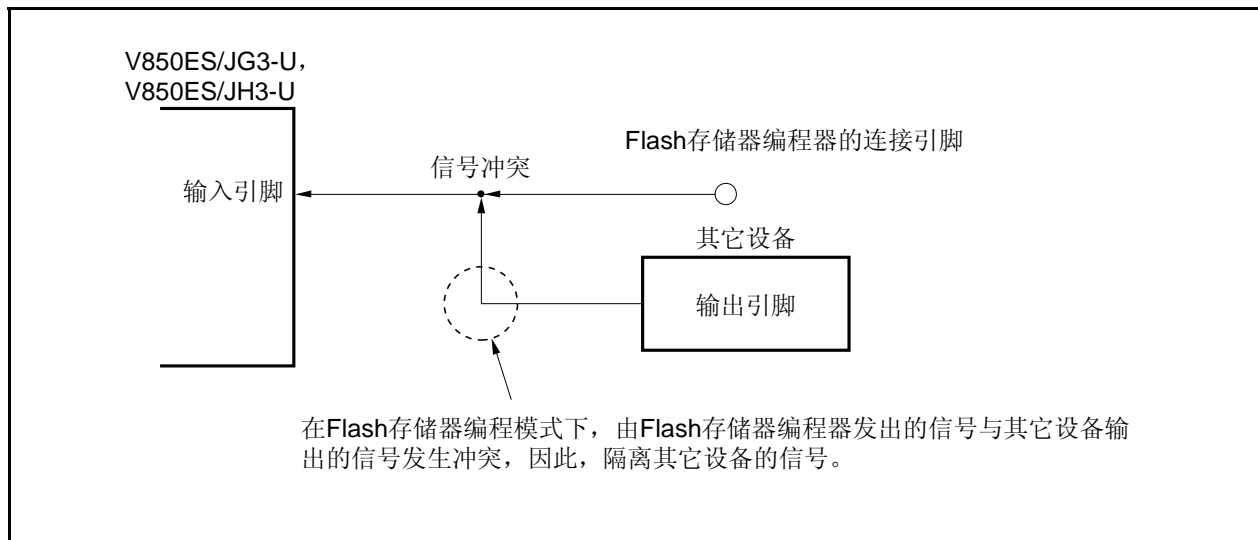
串行接口	所用引脚
UARTC0	TXDC0, RXDC0
CSIF0	SOF0, SIF0, $\overline{\text{SCKF0}}$
CSIF3	SOF3, SIF3, $\overline{\text{SCKF3}}$
CSIF0 + HS	SOF0, SIF0, $\overline{\text{SCKF0}}$, P913
CSIF3 + HS	SOF3, SIF3, $\overline{\text{SCKF3}}$, P913

当将 flash 存储器编程器接到某个串行接口引脚，该接口已经与板上其它设备连接时，应注意避免信号冲突或其它设备的误操作。

(a) 信号冲突

当 flash 存储器编程器（输出）连接到某个串行接口引脚（输入），该接口已经与板上其它设备（输出）连接时，会发生信号冲突。为避免信号冲突，隔离其与其它设备的连接或将其它设备设为输出高阻状态。

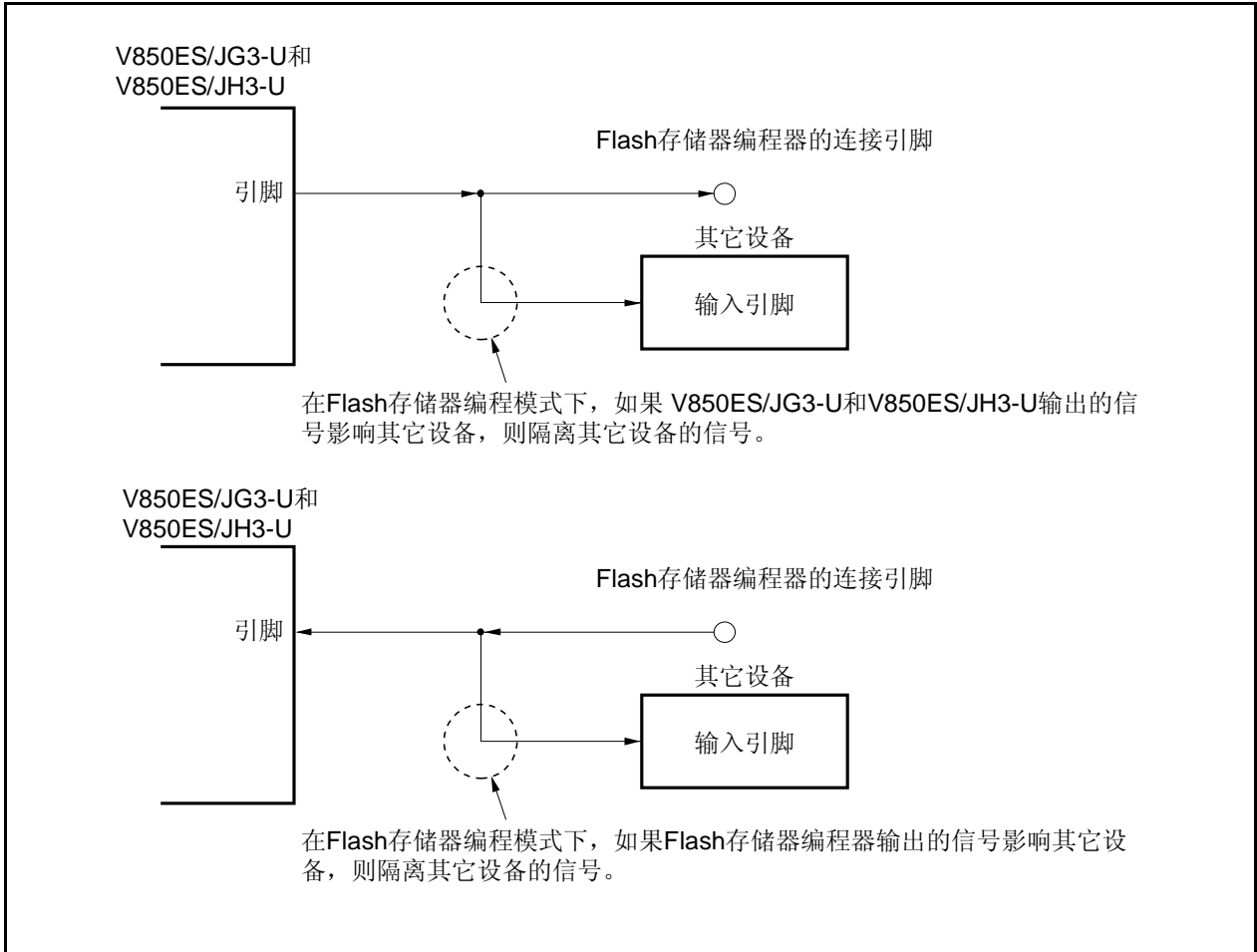
图 31-13. 信号冲突 (串行接口输入引脚)



(b) 其它设备的误操作

当 flash 存储器编程器（输出或输入）接到某个串行接口引脚（输入或输出），该接口已经与板上其它设备（输入）连接时，信号输出到其它设备，导致设备的误操作。为避免这种情况。隔离与其它设备的连接。

图 31-14. 其它设备的误操作

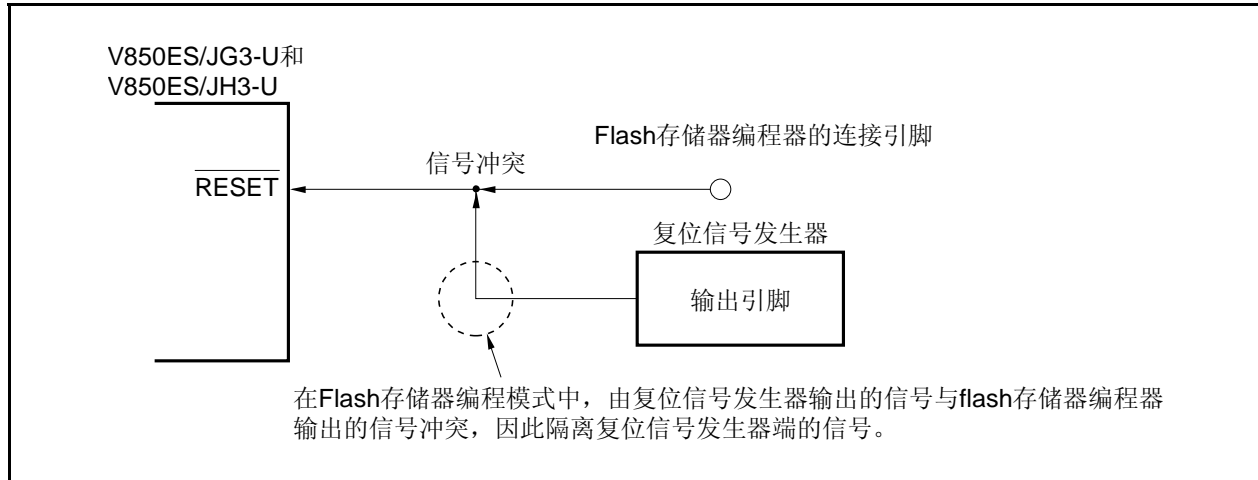


(4) RESET 引脚

当 flash 存储器编程器的复位信号接到 $\overline{\text{RESET}}$ 引脚，该引脚已经和板上的复位信号发生器相连时，会发生信号冲突。为避免信号间的冲突，隔离其它复位信号发生器的连接。

在 Flash 存储器编程模式下，当复位信号由用户系统输入时，编程操作将不能正确执行。所以，不要输入任何 flash 存储器编程器复位信号之外的信号。

图 31-15. 信号冲突 ($\overline{\text{RESET}}$ 引脚)

**(5) 端口引脚 (包括 NMI)**

当系统切换至 flash 存储器编程模式时，所有在 flash 存储器编程模式中不使用的引脚处于与复位后瞬时状态相同的状态。如果连接至各个端口的的外部器件无法识别端口复位后的瞬时状态，引脚就需要适当的处理，如通过电阻接至 V_{DD} 或 V_{SS} 。

(6) 其它信号引脚

X1、X2、XT1、XT2 和 REGC 的连接状态，与正常操作模式相同。

flash 存储器编程期间，向 $\overline{\text{DRST}}$ 引脚输入低电平或开路断开。不要输入高电平。

(7) 供电

提供的电源和正常操作模式的电压相等 (V_{DD} 、 V_{SS} 、 EV_{DD} 、 UV_{DD} 、 AV_{REF0} 、 AV_{REF1} 、 AV_{SS})。

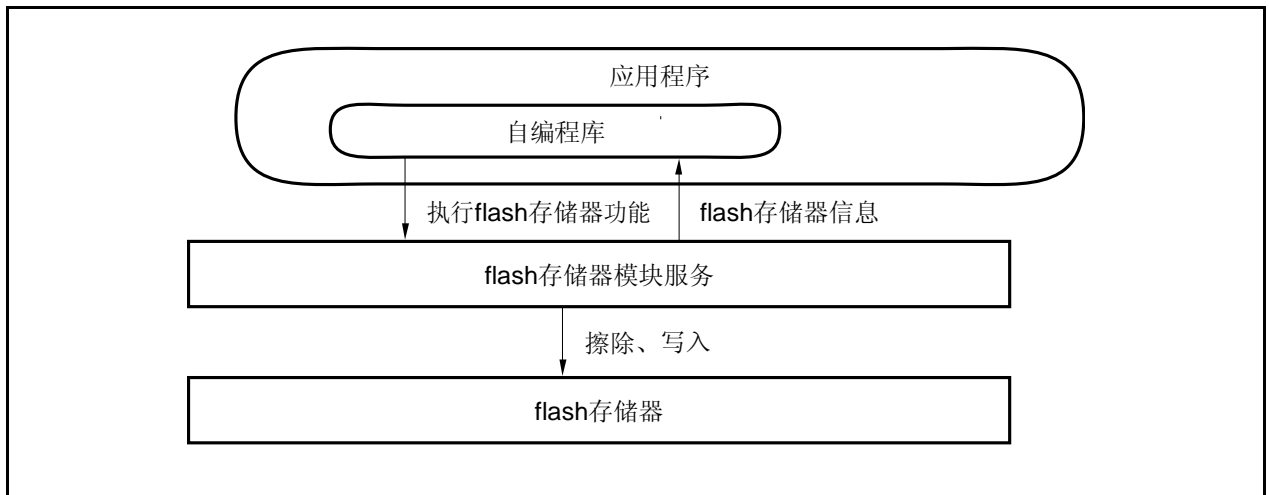
31.5 通过自编程进行重写

31.5.1 概述

V850ES/JG3-U 和 V850ES/JH3-U 支持 flash 模块服务，该服务允许用户程序自行重写内部 flash 存储器。通过用户应用程序对该接口和一个自编程库的使用来重写 flash 存储器，flash 存储器可被预先传送到内部 RAM 或外部存储器的用户程序重写。从而可以更新用户程序并在其区域内改写常量数据^注。

注 请确保不要将程序代码和重写目标的常量数据分配在同一个块中。块配置参见 31.2 存储器配置。

图 31-16. 自编程的概念

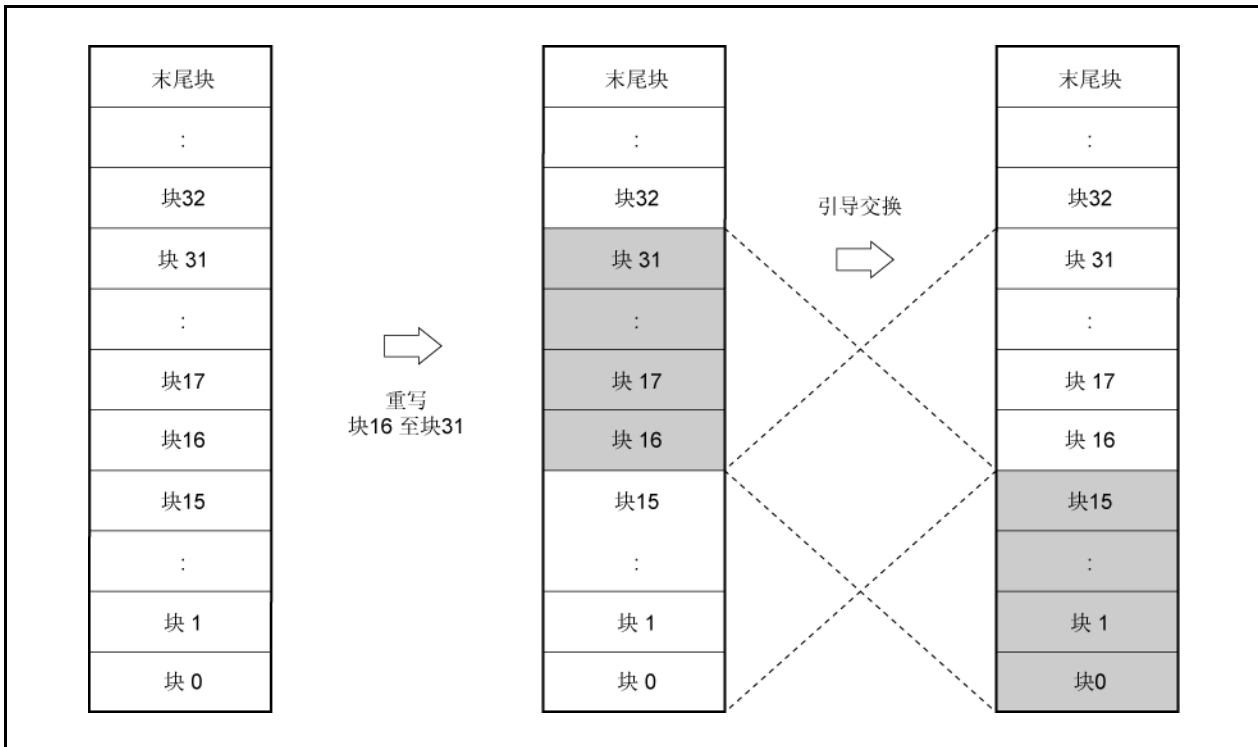


31.5.2 特性

(1) 安全自编程(引导交换功能)

V850ES/JG3-U 和 V850ES/JH3-U 支持引导交换功能，可将物理存储器的块 0 至 15 与块 16 至 31 互换。预先写入起始程序，该程序将要被写入块 16 至块 31，然后交换物理存储器，即使在重写期间发生断电，整个区域也可以安全地重新写入，因为正确用户程序始终存在于块 0 至块 15 中。

图 31-17. 重写整个存储区 (引导交换)



(2) 中断支持

自编程期间，不能从 flash 存储器提取指令。因此即使发生一个中断，写入到 flash 存储器的用户处理程序也不能使用。

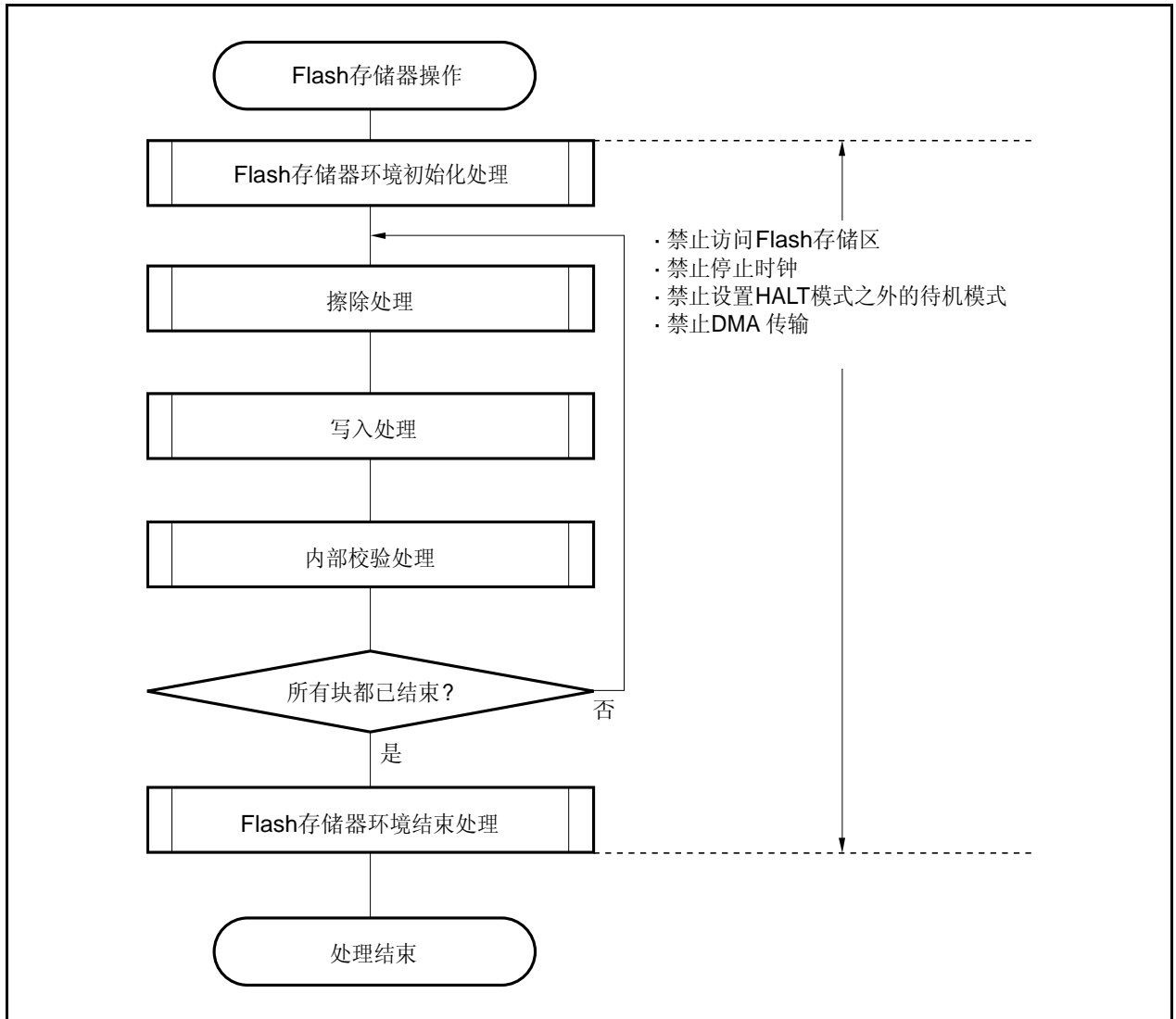
因此，为了在 V850ES/JG3-U 和 V850ES/JH3-U 自编程期间使用中断，处理转移到内部 RAM 的指定地址^注。安排一条跳转指令，将程序处理转移到内部 RAM 中指定地址^注上的用户中断服务程序。

注 NMI 中断： 内部 RAM 的起始地址。
可屏蔽中断： 内部 RAM 起始地址 + 4 的地址。

31.5.3 标准自编程流程

通过 flash 自编程对 flash 存储器重写的整个过程图示如下。

图 31-18. 标准自编程流程



31.5.4 Flash功能

表 31-10. Flash 功能列表

功能名称	概述	支持
FlashInit	自编程库初始化	√
FlashEnv	Flash 环境启动/停止	√
FlashFLMDCheck	FLMD 引脚检查	√
FlashStatusCheck	硬件处理执行状态检查	√
FlashBlockErase	块擦除	√
FlashWordWrite	数据写入	√
FlashBlockIVerify	块的内部校验	√
FlashBlockBlankCheck	块空白检查	√
FlashSetInfo	Flash 存储器信息设置	√
FlashGetInfo	Flash 存储器信息获取	√
FlashBootSwap	执行引导交换	√

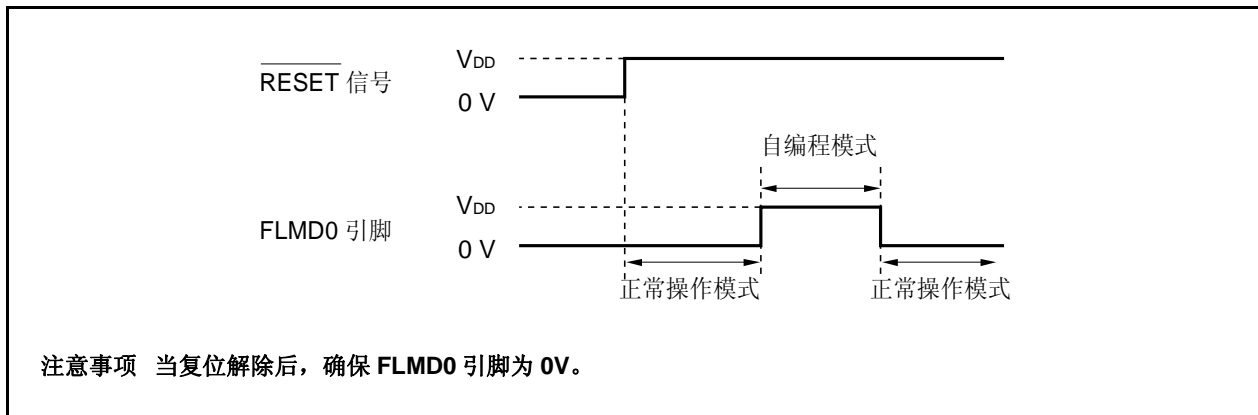
31.5.5 引脚处理

(1) FLMD0 引脚

FLMD0 引脚用于在复位解除后设置操作模式，并在自重写期间保护 flash 存储器不被写入。因此有必要在复位解除后和执行正常操作时保持 FLMD0 引脚的电压为 0V。在重新写入存储器之前，通过端口控制自编程模式期间有电平为 V_{DD} 的电压输入 FLMD0 引脚。

当完成自编程时，FLMD0 引脚电压值必须返回 0V。

图 31-19. 模式变换时序



31.5.6 用到的内部资源

下表列出了自编程需要使用的内部资源。这些内部资源也可以自由地用于除了自编程之外的用途。

表 31-11. 用到的内部资源

资源名称	说明
堆栈区	用户使用堆栈的扩展，被库使用 (内部 RAM 和外部 RAM 中都可以使用)。
库代码 ^注	库的程序实体 (可以用于被操作的 flash 存储器模块之外的任何空间)。
应用程序	作为用户应用程序执行。 调用 flash 存储器功能。
可屏蔽中断	可在用户应用程序的执行状态或自编程状态下使用。因为处理过程转移到内部 RAM 起始地址+4 的地址，为了在自编程状态中使用该中断，预先安排一条跳转指令，将程序处理转移到内部 RAM 起始地址+4 的地址上的用户中断服务程序。
NMI 中断	可在用户应用程序执行状态或自编程状态下使用。因为处理过程被转移到内部 RAM 起始地址，为了在自编程状态时使用该中断，预先安排一条跳转指令，将程序处理转移到内部 RAM 起始地址上的用户中断服务程序。

注 关于用到的资源，参见 **Flash 存储器自编程库用户手册**。

第三十二章 片上调试功能

V850ES/JG3-U 和 V850ES/JH3-U 单片机的片上调试功能可由下面两种方法实现。

- 使用 DCU (调试控制单元)
V850ES/JG3-U 和 V850ES/JH3-U 中，使用 $\overline{\text{DRST}}$ 、DCK、DMS、DDI 以及 DDO 引脚作为调试接口引脚来实现片上调试功能。
- 不使用 DCU
通过 MINICUBE2 或类似的设备来实现片上调试功能，使用用户资源来替代 DCU。

下表显示了两种片上调试功能的特性。

表 32-1. 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"> • 当使用 UARTC0 时 RXDC0, TXDC0 • 使用 CSIF0 时 SIF0、SOF0、$\overline{\text{SCKF0}}$、HS (P913) • 使用 CSIF3 时 SIF3、SOF3、$\overline{\text{SCKF3}}$、HS (P913)
用户资源保障		不需要	必需
硬件中断功能		2 点	2 点
软件中断功能	4 点	4 点	4 点
	2000 点	2000 点	2000 点
实时 RAM 监控功能 ^{注1}		可用	可用
动态内存修改(DMM)功能 ^{注2}		可用	可用
屏蔽功能		$\overline{\text{Reset}}$ 、NMI、INTWDT2、 $\overline{\text{HLDRQ}}$ 、WAIT	$\overline{\text{RESET}}$ 引脚
ROM 安全功能		10-字节 ID 码鉴定	10-字节 ID 码鉴定
使用硬件		MINICUBE 等.	MINICUBE2 等
跟踪功能		不支持	不支持
调试中断接口功能(DBINT)		不支持	不支持

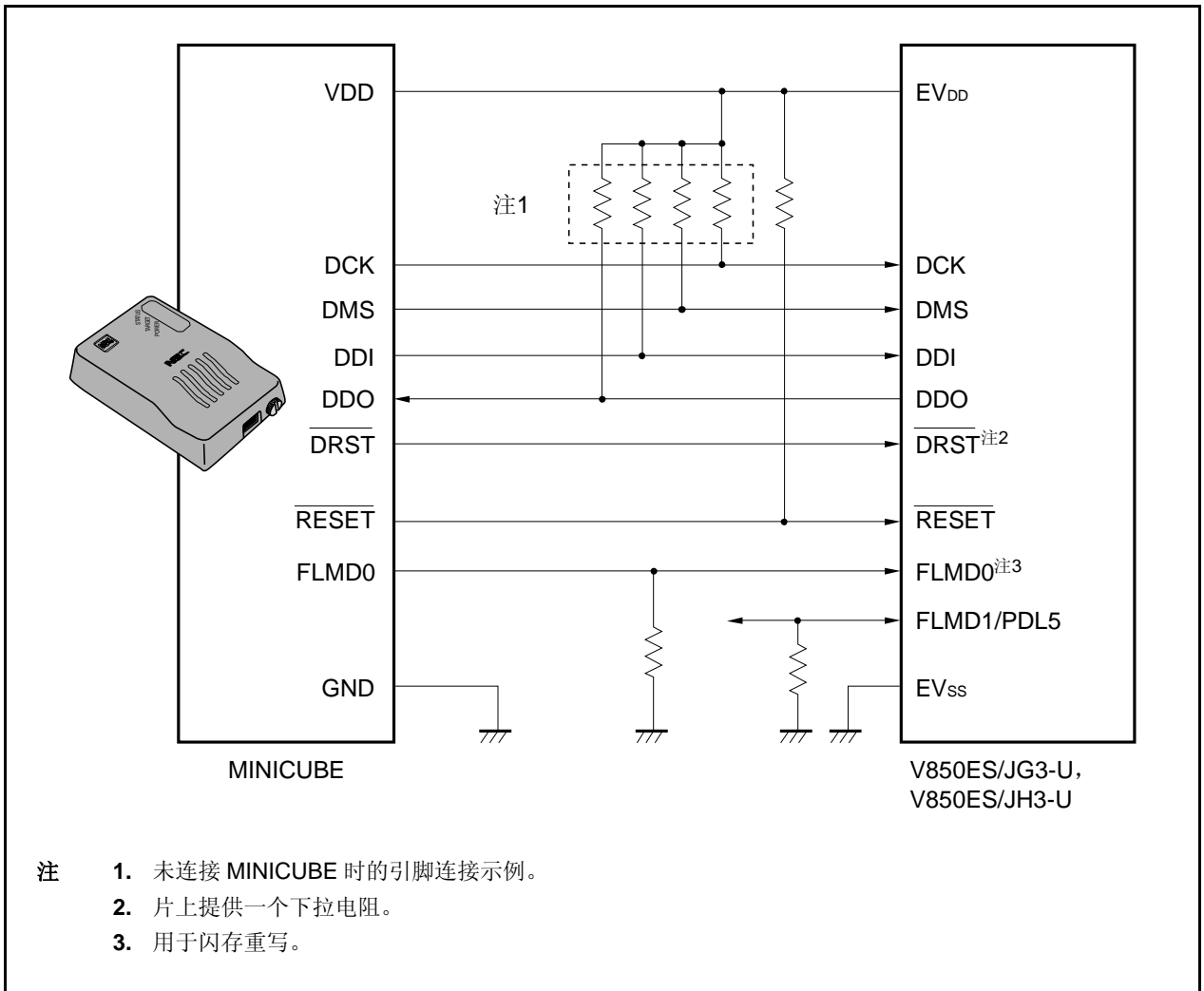
- 注
1. 这是在程序执行时读出存储器内容的功能。
 2. 这是在程序执行时重写 RAM 内容的功能。

32.1 用DCU调试

通过调试接口引脚 ($\overline{\text{DRST}}$, DCK, DMS, DDI 和 DDO) 连接片上调试仿真器(MINICUBE)来调试程序。

32.1.1 连接电路示例

图 32-1. 调试接口用作通信接口时的电路连接示例



32.1.2 接口信号

接口信号的描述如下：

(1) $\overline{\text{DRST}}$

它是片上调试单元的复位输入信号。它是负逻辑信号，异步初始化调试单元。

集成调试器启动后，当 MINICUBE 检测到目标系统的 V_{DD} 时使得 $\overline{\text{DRST}}$ 上升，并且启动设备上的片上调试单元。

当 $\overline{\text{DRST}}$ 信号上升为高后，CPU 产生一个复位信号。

在启动集成调试器开始调试后，总是会产生一个 CPU 复位。

(2) DCK

它是时钟输入信号。由 MINICUBE 为其提供 20 MHz 或 10 MHz 的时钟。在片上调试单元，在 DCK 信号的上升沿对 DMS 和 DDI 信号进行采样，而在其下降沿输出数据 DDO。

(3) DMS

它是传输模式选择信号。调试单元的传输状态随 DMS 信号的电平改变。

(4) DDI

这是数据输入信号。在片上调试单元内在 DCK 的上升沿对其进行采样。

(5) DDO

这是数据输出信号。在 DCK 信号的下降沿由片上调试单元输出该信号。

(6) EV_{DD}

该信号用于检测目标系统的 VDD。如果没有检测到目标系统的 VDD，则由 MINICUBE 输出该信号 ($\overline{\text{DRST}}$, DCK, DMS, DDI, FLMD0 和 RESET)，并进入高阻状态。

(7) FLMD0

Flash 的自编程功能用于通过集成调试器把数据下载到闪存。在 flash 存储器自编程期间，FLMD0 引脚必须保持为高。此外，在 FLMD0 引脚接一个下拉电阻。

FLMD0 引脚可通过下面两种方式进行控制：

<1> 由 MINICUBE 控制：

将 MINICUBE 的 FLMD0 信号连接到 FLMD0 引脚。

在正常模式下，MINICUBE 不作任何驱动（高阻）。

在中断期间，当集成调试器执行下载功能时 MINICUBE 将 FLMD0 引脚上升为高电平。

<2> 由端口控制：

将设备的任一端口连接到 FLMD0 引脚。

可能会用到由用户程序实现闪存自编程功能的同一个端口。

在集成调试器的控制端，在执行下载功能之前，对端口引脚进行设置使端口引脚为高电平，或在执行下载功能之后，降低端口引脚的电平。

详情参见 ID850QB Ver.3.40 集成调试器操作用户手册(U18604E)。

(8) $\overline{\text{RESET}}$

它是系统复位输入引脚。如果通过用户程序设置 OCDM 寄存器 OCDM0 位的值，使得 $\overline{\text{DRST}}$ 引脚无效，那么将无法执行片上调试。因此，通过 $\overline{\text{RESET}}$ 引脚，由 MINICUBE 引发复位，可以使得 $\overline{\text{DRST}}$ 引脚生效（初始化）。

32.1.3 可屏蔽功能

Reset、NMI、INTWDT2、 $\overline{\text{WAIT}}$ 和 $\overline{\text{HLDRQ}}$ 信号都可以被屏蔽。

调试器 (ID850QB) 的可屏蔽功能以及相应的 V850ES/JG3-U 和 V850ES/JH3-U 功能列举如下：

表 32-2. 可屏蔽功能

ID850QB 的可屏蔽功能	相应的 V850ES/JG3-U 和 V850ES/JH3-U 功能
NMI0	NMI 引脚输入
NMI2	产生不可屏蔽中断请求信号 (INTWDT2)
STOP	-
HOLD	$\overline{\text{HLDRQ}}$ 引脚输入
RESET	由 $\overline{\text{RESET}}$ 引脚输入、低电压检测器、时钟监控或看门狗定时器 (WDT2) 溢出，产生复位信号
WAIT	$\overline{\text{WAIT}}$ 引脚输入

32.1.4 寄存器

(1) 片上调试模式寄存器(OCDM) (仅限 V850ES/JG3-U)

OCDM 寄存器用于选择正常操作模式或片上调试模式。该寄存器是一个专用寄存器，只能通过一个特定序列的组合来写入(参见 3.4.7 专用寄存器)。

OCDM 寄存器用于确定片上调试功能的引脚是作为片上调试引脚，还是作为普通的端口/外设引脚。它也可以用来断开 P56/INTP05/ $\overline{\text{DRST}}$ 引脚的内部下拉电阻。

只有在 $\overline{\text{DRST}}$ 引脚输入低电平时，才能够写入 OCDM 寄存器。

该寄存器可按字节或按位进行读写。

复位后: 01H^注 R/W 地址: FFFFF9FCH

	7	6	5	4	3	2	1	<0>
OCDM	0	0	0	0	0	0	0	OCDM0

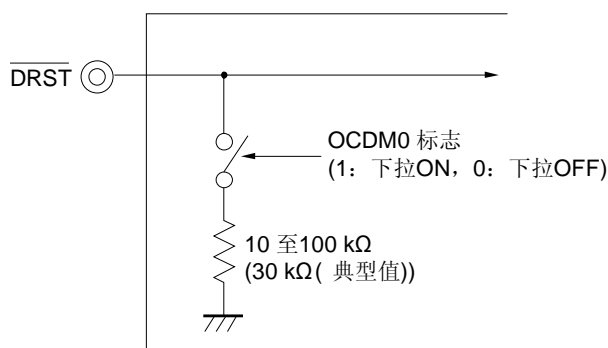
OCDM0	操作模式
0	选择正常操作模式 (可以被复用为片上调试功能引脚的引脚用作端口/外设功能引脚) 并且断开P56/INTP05/DRST引脚的片上下拉电阻。
1	当 $\overline{\text{DRST}}$ 引脚为低电平时: 正常操作模式(可以被复用功能为片上调试功能引脚的引脚用作端口/外设功能引脚) 当 $\overline{\text{DRST}}$ 引脚为高电平时: 片上调试模式 (可以被复用功能为片上调试功能引脚的引脚用作片上调试模式引脚)

注 $\overline{\text{RESET}}$ 输入将该寄存器设为 01H。由 WDT2RES 信号、时钟监控(CLM)或低电压检测器(LVI)引发复位后, OCDM 寄存器的值仍然保持。

注意事项 1. 外部复位后, 当 DDI、DDO、DCK 以及 DMS 引脚不作为片上调试引脚而作为端口引脚时, 必须执行下列所有操作。

- 给 P56/INTP05/DRST 引脚输入一个低电平。
- 将 OCDM0 位置位。这种情况下, 进行下列操作。
 - <1> 将 OCDM0 位清 0。
 - <2> 将 P56/INTP05/DRST 引脚固定为低电平, 直到<1>完成。

2. $\overline{\text{DRST}}$ 引脚有一个下拉电阻。当 OCDM0 标志被清 0 时, 断开该电阻。



32.1.5 操作

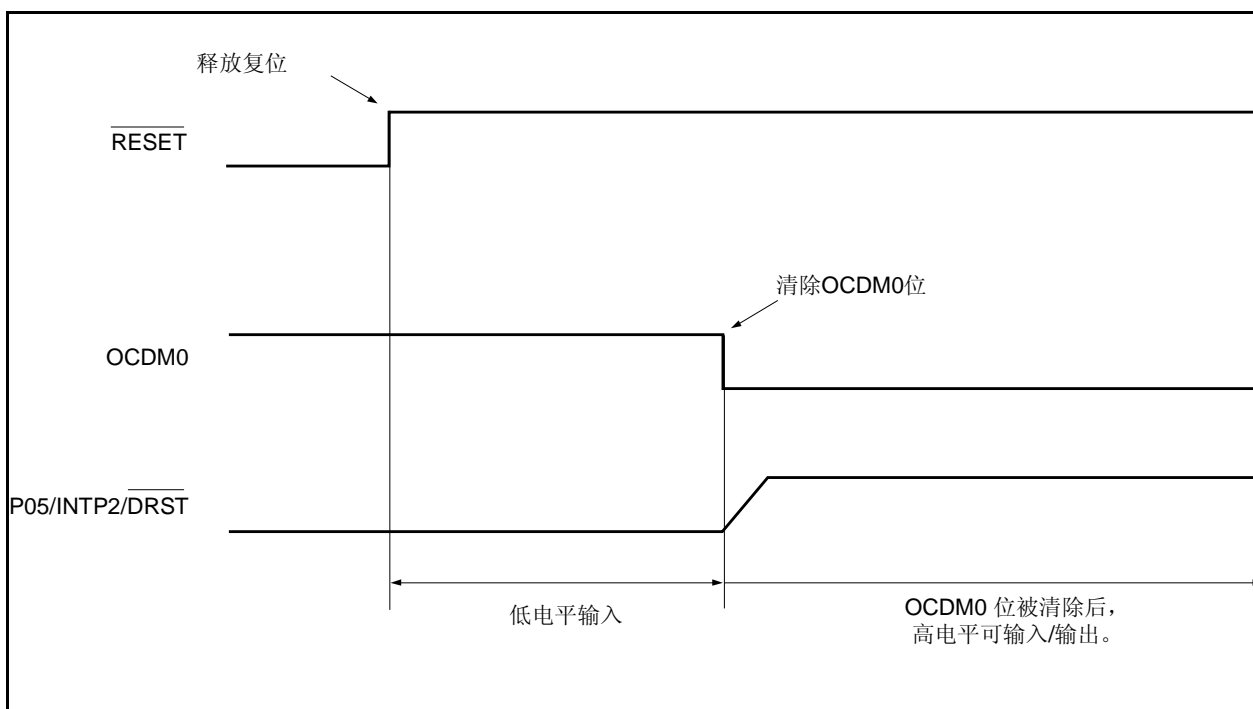
在下表条件时，片上调试功能无效。

当不使用该功能时，在 OCDM0.OCDM0 标志清 0 之前， $\overline{\text{DRST}}$ 引脚一直保持为低电平。

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 32-2. 不使用片上调试功能时的时序



32.1.6 注意事项

- (1) 如果在 RUN（程序执行）时输入一个复位信号（来自目标系统或来自内部复位源的复位信号），则中断功能可能会误操作。
- (2) 即使复位信号被屏蔽功能所屏蔽，如果在引脚输入一个复位信号也可能使得 I/O 缓存器(端口引脚)复位。
- (3) 中断时 reset 引脚被屏蔽，且 CPU 和外设 I/O 不复位。如果在执行用户程序时，flash 存储器由 DMM 重写或由 RAM 监控功能读取之后立刻产生引脚复位或内部复位，那么 CPU 和外设 I/O 可能无法正确复位。
- (4) 在片上调试模式下，DDO 引脚被强制设为高电平输出。

32.2 不使用DCU调试

下面描述如何不用 DCU 而使用 MINICUBE2 来实现片上调试功能。使用 MINICUBE2 的 UARTC0(RXDC0 和 TXDC0)引脚, CSIF0(SIF0、SOF0、 $\overline{\text{SCKF0}}$ 以及 HS(P913))引脚或 CSIF3(SIF3、SOF3、 $\overline{\text{SCKF3}}$ 以及 HS(P913))引脚作为调试接口

32.2.1 电路连接示例

图 32-3. UARTC0/CSIF0/CSIF3 用作通信接口时的电路连接示例

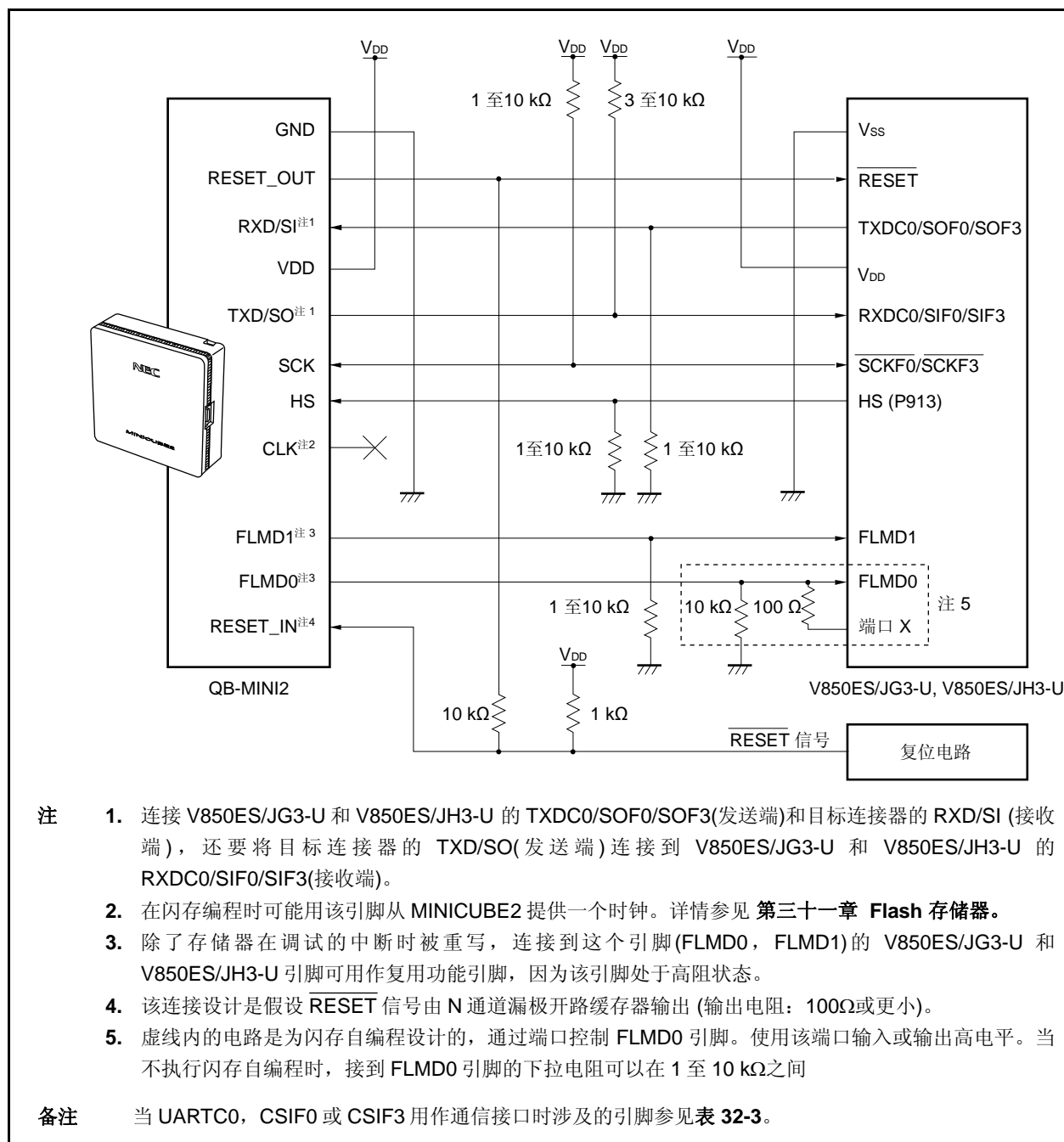


表 32-3. V850ES/JG3-U 和 MINICUBE2 之间的接线

MINICUBE2 (QB-MINI2)的引脚配置			使用 CSIF0-HS		使用 CSIF3-HS		使用 UARTC0	
信号名称	输入/输出	引脚功能	引脚名称	引脚编号	引脚名称	引脚编号	引脚名称	引脚编号
SI/RxD	输入	该引脚用于接收来自 V850ES/JG3-U 的指令和数据	P41/SOF0	23	P911/SOF3	53	P30/TXDC0	25
SO/TxD	输出	该引脚用于发送指令和数据至 V850ES/JG3-U	P40/SIF0	22	P910/SIF3	52	P31/RXDC0	26
SCK	输出	该引脚用于 3-线串行通信的时钟输出	P42/SCKF0	24	P912/SCKF3	54	不需要	-
CLK ^注	输出	到 V850ES/JG3-U 的时钟输出引脚	不需要 ^注	-	不需要 ^注	-	不需要 ^注	-
RESET_OUT	输出	到 V850ES/JG3-U 的复位输出引脚	RESET	14	RESET	14	RESET	14
FLMD0	输出	该输出引脚用于将 V850ES/JG3-U 设置为调试模式或编程模式	FLMD0	8	FLMD0	8	FLMD0	8
FLMD1	输出	该输出引脚用于设置编程模式	PDL5/FLMD1	76	PDL5/FLMD1	76	PDL5/FLMD1	76
HS	输入	CSIO + HS 通信的握手信号	P913	85	P913	85	不需要	-
GND	-	接地	V _{SS}	11, 33, 62, 79	V _{SS}	11, 33, 62, 79	V _{SS}	11, 33, 62, 79
			AV _{SS}	2	AV _{SS}	2	AV _{SS}	2
RESET_IN	输入	目标系统的复位输入						

注 它用作 MINICUBE2 型 flash 编程器的时钟输出。详情参见 第三十一章 FLASH 存储器。

表 32-4. V850ES/JH3-U 和 MINICUBE2 之间的接线

MINICUBE2 (QB-MINI2)的引脚配置			使用 CSIF0-HS		使用 CSIF3-HS		使用 UARTC0	
信号名称	输入/输出	引脚功能	引脚名称	信号名称	输入/输出	引脚功能	引脚名称	信号名称
SI/RXD	输入	该引脚用于接收来自 V850ES/JH3-U 的指令和数据	P41/SOF0	30	P911/SOF3	67	P30/TXDC0	37
SO/TXD	输出	该引脚用于发送指令和数据至 V850ES/JH3-U	P40/SIF0	29	P910/SIF3	66	P31/RXDC0	31
SCK	输出	该引脚用于 3-线串行通信的时钟输出	P42/SCKF0	31	P912/SCKF3	68	不需要	-
CLK ^注	输出	到 V850ES/JH3-U 的时钟输出引脚	不需要 ^注	-	不需要 ^注	-	不需要 ^注	-
RESET_OUT	输出	到 V850ES/JH3-U 的复位输出引脚	RESET $\bar{}$	18	RESET $\bar{}$	18	RESET $\bar{}$	18
FLMD0	输出	该输出引脚用于将 V850ES/JH3-U 设置为调试模式或编程模式	FLMD0	12	FLMD0	12	FLMD0	12
FLMD1	输出	该输出引脚用于设置编程模式	PDL5/FLMD1	103	PDL5/FLMD1	103	PDL5/FLMD1	103
HS	输入	CSIO + HS 通信的握手信号	P913	69	P913	69	不需要	-
GND	-	接地	Vss	15, 45, 84, 106	Vss	15, 45, 84, 106	Vss	15, 45, 84, 106
			AVss	2	AVss	2	AVss	2
RESET_IN	输入	目标系统的复位输入						

注 它用作 MINICUBE2 型 flash 编程器的时钟输出。详情参见 第三十一章 FLASH 存储器。

32.2.2 可屏蔽功能

只有复位信号可以被屏蔽。

在调试器(ID850QB)中可以屏蔽的功能以及相应的 V850ES/JG3-U 和 V850ES/JH3-U 的功能列举如下：

表 32-4. 可屏蔽功能

ID850QB 中可以屏蔽的功能	相应的 V850ES/JG3-U 和 V850ES/JH3-U 功能
NMI0	-
NMI1	-
NMI2	-
STOP	-
HOLD	-
RESET	由 RESET 引脚输入产生复位信号
WAIT	-

32.2.3 用户资源的保留

用户必须准备下列资源来实现 MINICUBE2 和目标器件间的通信功能，并完成各调试功能。这些内容需要在用户程序中设置或使用编译器选项设置。

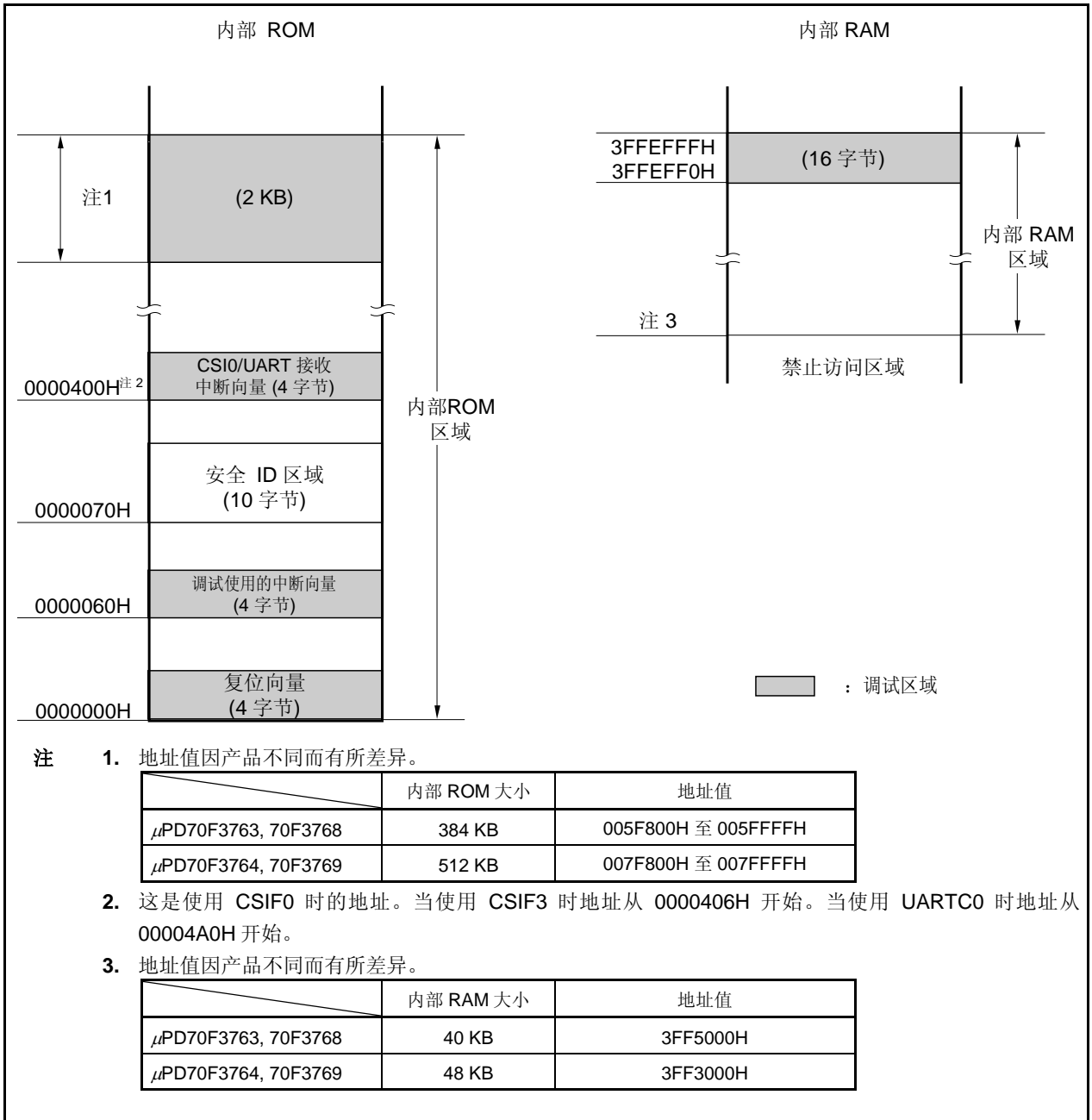
(1) 存储器空间的保留

图 32-4 中的阴影部分是预留区，用于放置调试监控程序，因此用户程序和数据不能分配在这些空间。必须保证这些空间不被用户程序使用。

(2) 安全 ID 设置

ID 码必须嵌入到图 32-4 中 0000070H 至 0000079H 的区域，以防止存储内容被未经授权读取。详情参见 **32.3 ROM 安全功能**。

图 32-4. 调试监控程序的内存空间分配



(3) 复位向量

复位向量包括调试监控程序所需的转移指令。

[如何保留区域]

没有必要刻意保护该区域。但是，当下载一个程序时，调试器根据下列情况重写复位向量。若重写的模式与下列情况不符，则调试器产生一个错误（使用 ID850QB 时为 F0C34）。

(a) 当从 0 地址开始连续放置两个空指令时

重写前		重写后
0x0 nop	→	跳转到 0x0 处的调试监控程序
0x2 nop		0x4 xxxx
0x4 xxxx		

(b) 当从 0 地址开始连续放置两个 0xFFFF 时(已擦除器件)

重写前		重写后
0x0 0xFFFF	→	跳转到 0x0 处的调试监控程序
0x2 0xFFFF		0x4 xxxx
0x4 xxxx		

(c) jr 指令放在 0 地址（使用 CA850 时）

重写前		重写后
0x0 jr disp22	→	跳转到 0x0 处的调试监控程序
		0x4 jr disp22 - 4

(d) 从 0 地址开始连续放置 mov32 和 jmp（使用 IAR 编译 ICCV850 时）

重写前		重写后
0x0 mov imm32,reg1	→	跳转到 0x0 处的调试监控程序
0x6 jmp [reg1]		0x4 mov imm32,reg1
		0xa jmp [reg1]

(e) 在 0 地址处放置用于跳转到调试监控程序的指令

重写前		重写后
跳转到 0x0 处的调试监控程序	→	没有改变

(4) 调试监控程序的保留区域

图 32-4 中的阴影部分是分配给调试监控程序的区域。监控程序对调试通信接口和 CPU 的 RUN 或中断处理过程执行初始化。内部 ROM 区必须填充为 0xFF。该区域禁止被用户程序重写。

[如何保留区域]

若用户程序不使用该区域，则没有必要进行保护。

但是，为了避免在调试器启动时可能出现的问题，建议预先使用编译器保留本区域。

下面给出保留区域的示例，使用 NEC 电子的编译器 CA850。加上汇编源文件和连接伪指令代码，如下所示。

- 汇编源文件(添加下列代码作为汇编源文件。)

```
-- Secures 2 KB space for monitor ROM section
.section "MonitorROM", const
.space 0x800, 0xff

-- Secures interrupt vector for debugging
.section "DBG0"
.space 4, 0xff

-- Secures interrupt vector for serial communication
-- Change the section name according to the serial communication mode used
.section "INTCF0R"
.space 4, 0xff

-- Secures 16-byte space for monitor RAM section
.section "MonitorRAM", bss
.lcomm monitorramsym, 16, 4 -- defines symbol monitorramsym
```

- 连接伪指令(将下列代码添加到连接伪指令文件)

下面给出一个例子，它的内部 ROM 为 512 KB(末地址为 007FFFFH)且内部 RAM 为 56KB (末地址为 3FFEFF H)。

```
MROMSEG : !LOAD ?R V0x07f800{
        MonitorROM = $PROGBITS ?A MonitorROM;
};

MRAMSEG : !LOAD ?RW V0x03ffeff0{
        MonitorRAM = $NOBITS ?AW MonitorRAM;
};
```

(5) 通信串行接口的保留

UARTC0, CSIF0 或 CSIF3 用于 MINICUBE2 和目标系统之间的通信。与串行接口模式相关的设置通过调试监控程序来完成，但如果设置被用户程序改变，则可能发生通信错误。

为了防止发生这样的问题，必须在用户程序中对通信串行接口进行保留。

[如何保留通信串行接口]

- 片上调试模式寄存器(OCDM)

使用 UARTC0, CSIF0 或 CSIF3 的片上调试功能时，设置 OCDM 寄存器功能为正常模式。务必按照下面步骤进行设置。

- P05/INTP02/DRST 引脚输入低电平。
- 将 OCDM0 位设置如下：
 - <1> 将 OCDM0 位清 0。
 - <2> 在<1>完成之前将 P05/INTP02/DRST 引脚输入固定为低电平。

- 串行接口寄存器

在用户程序中不要设置 CSIF0, CSIF3 或 UARTC0 相关的寄存器。

- 中断屏蔽寄存器

使用 CSIF0 时，不要屏蔽发送结束中断(INTCF0R)。使用 CSIF3 时，不要屏蔽发送结束中断(INTCF3R)。使用 UARTC0 时，不要屏蔽接收结束中断 (INTUA0R)。

(a) 使用 CSIF0 时

	7	6	5	4	3	2	1	0
CF0RIC	x	0	x	x	x	x	x	x

(b) 使用 CSIF3 时

	7	6	5	4	3	2	1	0
CF3RIC	x	0	x	x	x	x	x	x

(c) 使用 UARTC0 时

	7	6	5	4	3	2	1	0
UC0RIC	x	0	x	x	x	x	x	x

备注 x: 无须理会

- 使用 UARTC0 时的端口寄存器

当使用 UARTC0 时，通过调试监控程序对端口寄存器进行设置，使 TXDC0 和 RXDC0 引脚有效。在调试期间用户程序不要改变下列寄存器的设置。(同样的值可被覆盖写入)

	7	6	5	4	3	2	1	0
PFC3	x	x	x	x	x	x	0	0
	7	6	5	4	3	2	1	0
PFCE3	x	x	x	x	x	x	0	0
	7	6	5	4	3	2	1	0
PMC3	x	x	x	x	x	x	1	1

备注 x: 无需理会

- 使用 CSIF0 的端口寄存器

使用 CSIF0 时，通过调试监控程序对端口寄存器进行设置，使 SIF0、SOF0、 $\overline{\text{SCKF0}}$ 和 HS(P913)引脚生效。在调试期间用户程序不要改变下列寄存器的设置。(同样的值可被覆盖写入)

(a) SIF0、SOF0 和 $\overline{\text{SCKF0}}$ 设置

	7	6	5	4	3	2	1	0
PMC4	x	x	x	x	x	1	1	1
	7	6	5	4	3	2	1	0
PFC4	x	x	x	x	x	x	0	0
	7	6	5	4	3	2	1	0
PFCE4	x	x	x	x	x	x	0	0

(b) HS (P913 引脚)设置

	15	14	13	12	11	10	9	8
PM9H	x	x	0	x	x	x	x	x
	15	14	13	12	11	10	9	8
P9H	x	x	注	x	x	x	x	x

注 禁止写入该位。
对应 HS 引脚的端口值根据调试器的状态由监控程序改变。为了按照字节进行端口寄存器的设置，用户程序通常会使用读取—修改—写入方法。但是，如果在写入之前发生一个调试中断，就可能执行一个意外的操作。

备注 x: 无需理会

- 使用 CSIF3 的端口寄存器

使用 CSIF3 时，通过调试监控程序对端口寄存器进行设置，使 SIF3、SOF3、 $\overline{\text{SCKF3}}$ 和 HS(P913)引脚生效。在调试期间用户程序不要改变下列寄存器的设置。(同样的值可被覆盖写入)

(a) SIF3、SOF3 和 $\overline{\text{SCKF3}}$ 设置

	15	14	13	12	11	10	9	8
PMC9H	x	x	x	1	1	1	x	x

	15	14	13	12	11	10	9	8
PFC9H	x	x	x	x	0	0	x	x

	15	14	13	12	11	10	9	8
PFCE9H	x	x	x	x	0	0	x	x

(b) HS (P913 引脚)设置

	15	14	13	12	11	10	9	8
PM9H	x	x	0	x	x	x	x	x

	15	14	13	12	11	10	9	8
P9H	x	x	注	x	x	x	x	x

注 禁止写入该位。

对应 HS 引脚的端口值根据调试器的状态由监控程序改变。为了按照字节进行端口寄存器的设置，用户程序通常会使用读取—修改—写入方法。但是，如果在写入之前发生一个调试中断，就可能会执行一个意外的操作。

备注 x: 无需理会

32.2.4 注意事项

(1) 被调试设备的处理

不要将用于调试的设备安装在批量生产的产品上，因为在调试过程中闪存被重写过，所以不能保证闪存的重写次数。此外，不要将调试监控程序嵌入到批量生产的产品中。

(2) 当中断不能执行时

如果满足下列条件之一，则强制中断无法执行。

- 中断禁止(DI)。
- 串行接口发出的中断被屏蔽，该中断用于 MINICUBE2 和目标器件之间的通信。
- 禁止在通过可屏蔽中断解除待机时进入待机模式。
- MINICUBE2 和目标器件之间的通信模式为 UARTC0，且主时钟已被停止。

(3) 当不执行伪实时 RAM 监控(RRM) 功能和 DMM 功能时

如果满足下列条件之一，将 RRM 功能和 DMM 功能不能操作。

- 中断禁止(DI)。
- 串行接口发出的中断被屏蔽，该中断用于 MINICUBE2 和目标器件之间的通信。
- 禁止在通过可屏蔽中断解除待机时进入待机模式。
- MINICUBE2 和目标器件之间的通信模式为 UARTC0，且主时钟已被停止。
- MINICUBE2 和目标器件之间的通信模式为 UARTC0，并且用于通信的时钟和调试器中指定的时钟不同。

(4) 允许由伪 RRM 和 DMM 功能解除待机模式。

如果满足下列条件之一，则可以由伪 RRM 功能和 DMM 功能来解除待机模式。

- MINICUBE2 和目标器件之间的通信模式为 CSIF0 或 CSIF3。
- MINICUBE2 和目标器件之间的通信模式为 UARTC0，且已提供主时钟

(5) 使用 DMM 功能写入需要特定序列重写的外设 I/O 寄存器。

需要特定序列的外设 I/O 寄存器，不能用 DMM 功能写入。

(6) Flash 自编程

如果分配给调试监控程序的空间被闪存自编程所重写，那么调试器将不能再正常运行。

32.3 ROM安全功能

32.3.1 安全ID

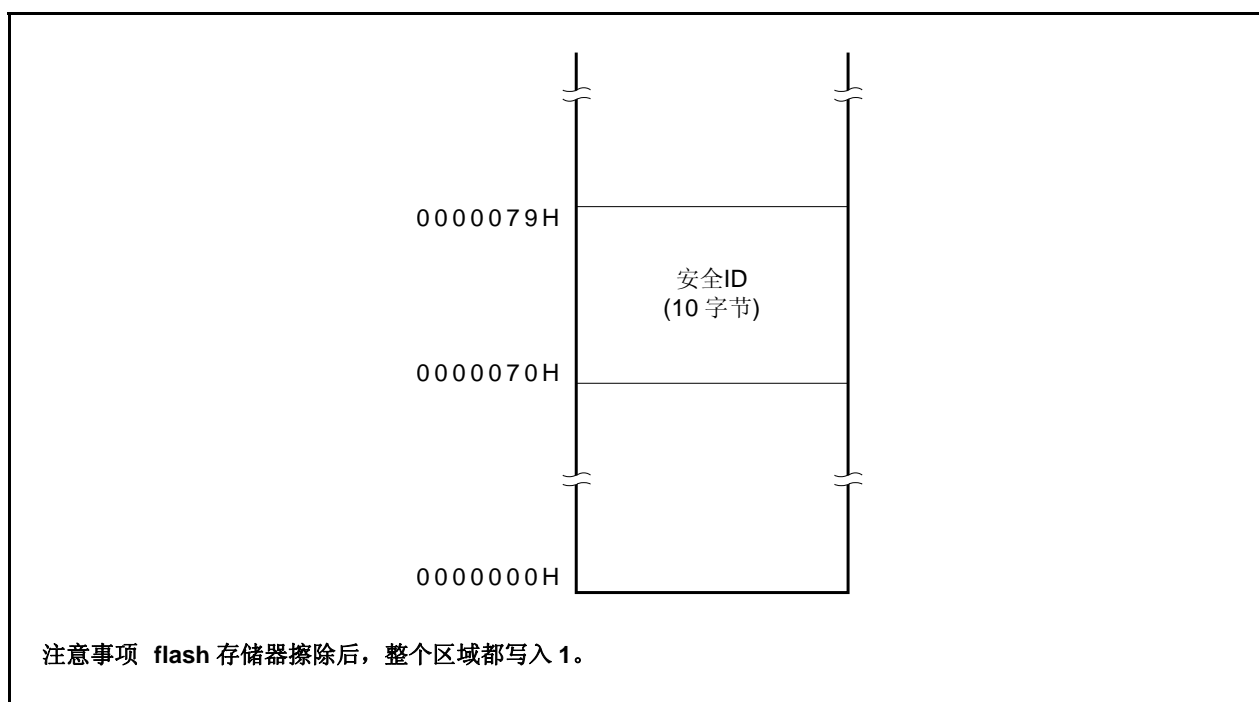
V850ES/JG3-U 和 V850ES/JH3-U 的闪存版本执行鉴定，通过使用一个 10 字节的 ID 码，防止闪存中的内容在由片上仿真器进行片上调试时被未经授权读取。

在 10 字节的片上 flash 存储区地址从 0000070H 至 0000079H 设置 ID 码，以便允许调试器执行 ID 鉴定。

如果 ID 匹配，则安全保护解除，且允许对闪存进行读取，也允许使用片上调试仿真器。

- 在 0000070H 至 0000079H 之间设置 10 字节的 ID 码。
- 0000079H 的第 7 位是片上调试仿真器的使能标志。
(0: 禁用, 1: 使能)
- 当启动片上调试仿真器后，调试器要求输入 ID。当输入的 ID 和 0000070H 至 0000079H 间设置的 ID 码匹配时，调试器启动。
- 即使 ID 码匹配，当片上调试仿真器的使能标志为 0 时，也不能进行调试。

图 32-5. 安全 ID 区



32.3.2 设置

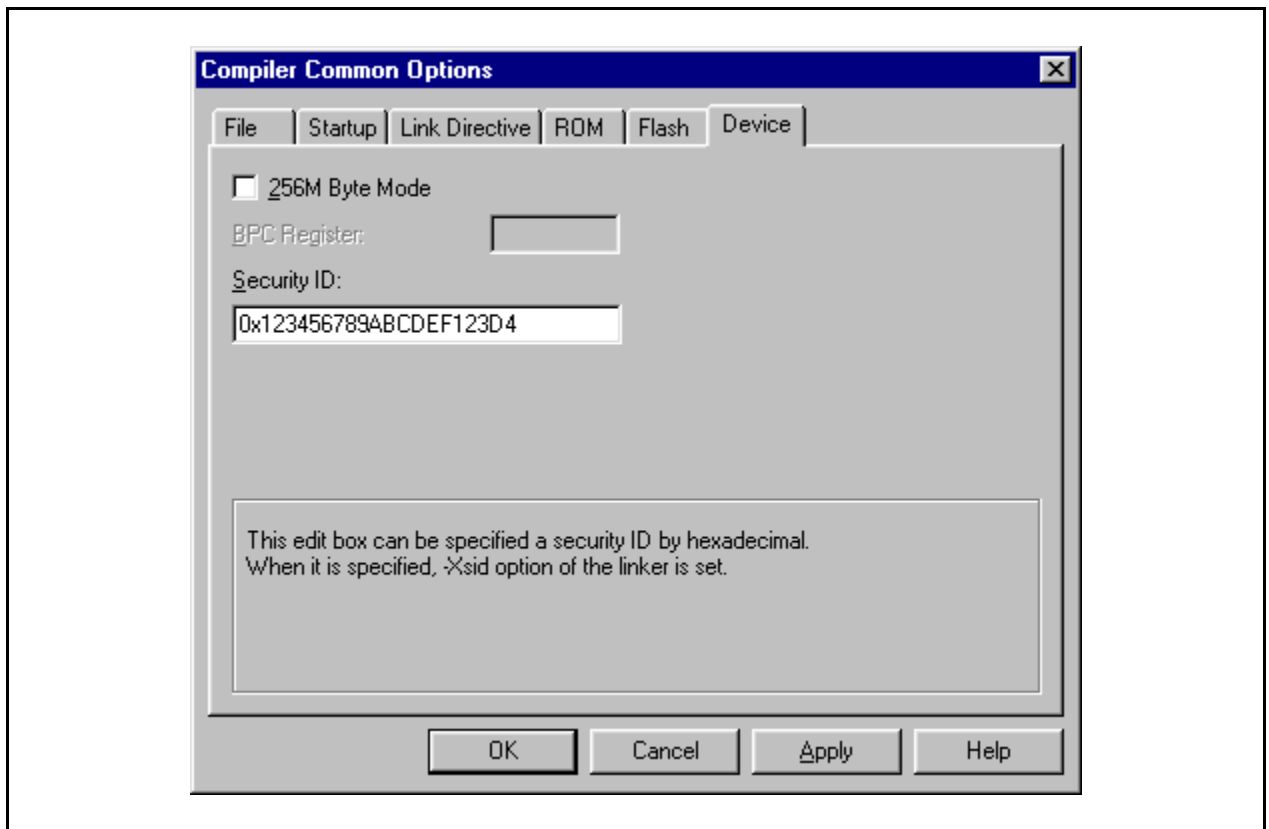
下面在表 32-6 中显示如何设置 ID 码。

当 ID 码按照表 32-6 进行设置时，在 ID850QB 的配置对话框中的 ID 码输入为 “123456789ABCDEF123D4”(ID 码不区分大小写)。

表 32-6. 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

如果设备文件支持 CA850 Ver. 3.10 及更高版本，并且使用了安全 ID，则 ID 码可以在 PM+ 中的编译器通用选项对话框来指定。



[程序示例 (当使用 CA850 Ver. 3.10 或更高版本时)]

```
#-----  
# SECURITYID  
#-----  
    .section  "SECURITY_ID"  --Interrupt handler address 0x70  
    .word     0x78563412     --0-3 byte code  
    .word     0xF1DEBC9A     --4-7 byte code  
    .hword    0xD423         --8-9 byte code
```

备注 将上述程序示例添加到启动文件中。

33.1 最大额定值

(T_A = 25° C) (1/2)

参数	符号	条件	额定值范围	单位
供应电压	V _{DD}	V _{DD} = EV _{DD} = UV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 至 +4.6	V
	EV _{DD}	V _{DD} = EV _{DD} = UV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 至 +4.6	V
	UV _{DD}	V _{DD} = EV _{DD} = UV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 至 +4.6	V
	AV _{REF0}	V _{DD} = EV _{DD} = UV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 至 +4.6	V
	AV _{REF1}	V _{DD} = EV _{DD} = UV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 至 +4.6	V
	V _{SS}	V _{SS} = AV _{SS}	-0.5 至 +0.5	V
	AV _{SS}	V _{SS} = AV _{SS}	-0.5 至 +0.5	V
输入电压	V _{I1}	P60至P65, P90至P915, PCM0, PCM1, PCS0, PCS2, PCS3, PCT0, PCT1, PCT4, PCT6, PDL0至PDL15, PDH0至PDH7, RESET, FLMD0	-0.5 至 EV _{DD} + 0.5 ^{注 1}	V
	V _{I2}	UDMF, UDPF, UDMH, UDPH	-0.5 至 UV _{DD} + 0.5 ^{注 1}	V
	V _{I3}	P10, P11	-0.5 至 AV _{REF1} + 0.5 ^{注 1}	V
	V _{I4}	X1, X2, XT1, XT2	-0.5 至 V _{RO} ^{注 2} + 0.5 ^{注 1}	V
	V _{I5}	P00至P05, P20至P25, P30至P37, P40至P42 P50至P56, PCM2, PCM3	-0.5 至 +6.0	V
模拟输入电压	V _{IAN}	P70 至 P711	-0.5 至 AV _{REF0} + 0.5 ^{注 1}	V

- 注 1. 确保所有供电电压都不超过最大额定值（最大值）。
2. 片上稳压器输出电压

- 注意事项 1. 不要将芯片的输出（或输入/输出）引脚直接互连，或直接连接至 V_{DD}、V_{CC} 和 GND。但是，漏极开路或集电极开路引脚可以直接互连。
如果输出引脚能够设置为高阻状态，并且外部电路的输出时序设计时已经考虑避免输出冲突，芯片和外部电路之间的输出引脚才可以直接相连。
2. 任何参数，哪怕瞬间超过最大额定值，都可能使产品性能受损。就是说，产品工作在最大额定值时处于物理损坏的边缘，因此，产品必须在不超出最大额定值的条件下使用。DC 和 AC 特性的额定值和工作条件代表在正常工作期间的性能保障范围。

- 注 1. 请确保不要超过各路供电电压的最大额定值（最大值）。
2. 片上稳压器输出电压（2.5 V (典型值)）

备注 除非另外说明，复用功能引脚的特性与端口引脚的特性相同。

最大额定值 ($T_A = 25^\circ\text{C}$) (2/2)

参数	符号	条件	额定值范围	单位			
输出电流, 低	I _{OL}	P00 至 P05, P20 至 P25, P30 至 P37, P40 至 P42, P50 至 P56, P60 至 P65, P90 至 P915, PCM0 至 PCM3, PCS0, PCS2, PCS3, PCT0, PCT1, PCT4, PCT6, PDL0 至 PDL15, PDH0 至 PDH7	每个引脚	4	mA		
			全部引脚总和	100	mA		
		UDMF, UDPF, UDMH, UDPH	每个引脚	4	mA		
			全部引脚总和	16	mA		
		P10, P11	每个引脚	4	mA		
			全部引脚总和	8	mA		
		P70 至 P711	每个引脚	4	mA		
			全部引脚总和	20	mA		
		输出电流, 高	I _{OH}	P00 至 P05, P20 至 P25, P30 至 P37, P40 至 P42, P50 至 P56, P60 至 P65, P90 至 P915, PCM0 至 PCM3, PCS0, PCS2, PCS3, PCT0, PCT1, PCT4, PCT6, PDL0 至 PDL15, PDH0 至 PDH7	每个引脚	-4	mA
					全部引脚总和	-100	mA
UDMF, UDPF, UDMH, UDPH	每个引脚			-4	mA		
	全部引脚总和			-16	mA		
P10, P11	每个引脚			-4	mA		
	全部引脚总和			-8	mA		
P70 至 P711	每个引脚			-4	mA		
	全部引脚总和			-20	mA		
工作环境温度	T _A			正常工作模式	-40 至+85	°C	
				Flash 存储器编程模式	-40 至+85	°C	
储存温度	T _{stg}		-40 至+125	°C			

注意事项 1. 不要将芯片的输出（或输入/输出）引脚直接互连，或直接连接至 V_{DD}、V_{CC} 和 GND。但是，漏极开路或集电极开路引脚可以直接互连。

如果输出引脚能够设置为高阻状态，并且外部电路的输出时序设计时已经考虑避免输出冲突，芯片和外部电路之间的输出引脚才可以直接相连。

- 2.** 任何参数，哪怕瞬间超过最大额定值，都可能使产品性能受损。就是说，产品工作在最大额定值时处于物理损坏的边缘，因此，产品必须在不出最大额定值的条件下使用。DC 和 AC 特性的额定值和工作条件代表在正常工作期间的性能保障范围。

备注 除非另外说明，复用功能引脚的特性与端口引脚的特性相同。

33.2 电容

(TA = 25°C, VDD = EVDD = UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V)

参数	符号	条件	最小值	典型值	最大值	单位
I/O 电容	C _{IO}	f _x = 1 MHz 测量的引脚返回至0 V			10	pF

33.3 工作条件

(TA = -40 至 +85°C, VDD = EVDD = UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V, CL = 50 pF)

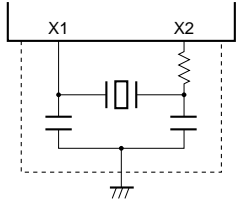
内部系统时钟频率	条件	供应电压				单位
		V _{DD}	EV _{DD}	UV _{DD}	AV _{REF0} , AV _{REF1}	
f _{XX} = 3 至 6 MHz (时钟直通操作) f _{XX} = 24 至 48 MHz (PLL 操作期间)	C = 4.7 μF, A/D 转换器停止, D/A 转换器停止, USB 停止	2.85 至 3.6	2.85 至 3.6	2.85 至 3.6	2.85 至 3.6	V
	C = 4.7 μF, A/D 转换器工作, D/A 转换器工作, USB 工作	3.0 至 3.6	3.0 至 3.6	3.0 至 3.6	3.0 至 3.6	V
f _{XT} = 32.768 kHz	C = 4.7 μF, 注	2.85 至 3.6	2.85 至 3.6	2.85 至 3.6	2.85 至 3.6	V

注 当系统运行于副时钟 (f_{XT} = 32.768 kHz) 时, A/D 转换器, D/A 转换器和 USB 控制器不工作。

33.4 振荡器特性

33.4.1 主时钟振荡器特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

谐振器	电路举例	参数	条件	最小值	典型值	最大值	单位
陶瓷谐振器 /晶体谐振器		振荡频率 (fx) ^{注 1}		3		6	MHz
		振荡稳定时间 ^{注 2}	复位解除后		$2^{16}/f_x$		s
			STOP 模式解除后		注 3		ms
	IDLE2 模式解除后		注 3		μs		

- 注 1. 以上所示的振荡频率仅表示振荡器特性。使用 V850ES/JG3-U 和 V850ES/JH3-U，内部工作条件不要超出 **AC 特性**和 **DC 特性**的额定值。
2. 从振荡开始到振荡电路稳定所需的时间。
3. 该值会根据 OSTS 寄存器的设置而不同。

注意事项 1. 使用 USB 控制器时，如果选择内部时钟作为 USB 时钟，则请确保所用的陶瓷谐振器或晶体谐振器精度为 **6 MHz \pm 500 ppm** 或更低。

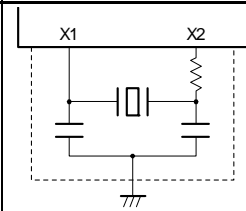
使用从 UCLK 引脚输入的外部时钟时，请确保提供的时钟精度为 **48 MHz \pm 500 ppm** 或更低。

如果 USB 时钟精度降低，则发送/接收数据无法满足 USB 规范。

2. 在使用主时钟振荡器时，上图中虚线框部分的连线应遵从如下的连接方式，避免线间电容的负面影响。
- 连线要保证尽可能短。
 - 振荡器连线不要和其它的信号线互相交叉。
 - 布线时不要将振荡器连线靠近通过大波动电流的信号线
 - 保持振荡器电容的地电位和 V_{SS} 相同。
 - 不要将电容的接地点连接在大电流通过的地模式。
 - 不要从振荡器取信号。
3. 当主时钟停止，设备运行于副时钟时，在由副时钟切换回主时钟之前，应该由程序来实现振荡稳定时间的等待。

33.4.2 副时钟振荡器特性

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

谐振器	电路举例	参数	条件	最小值	典型值	最大值	单位
晶体谐振器		振荡频率 (f_{XT}) ^{注 1}		32	32.768	35	kHz
		振荡稳定时间 ^{注 2}				10	s

- 注
- 以上所示的振荡频率仅表示振荡器特性。使用 V850ES/JG3-U 和 V850ES/JH3-U，内部工作条件不要超出 **AC 特性** 和 **DC 特性** 的额定值。
 - 从 V_{DD} 达到振荡电压范围 (2.85 V (最小值)) 起，到晶体振荡器稳定所需的时间。

注意事项 1. 在使用副时钟振荡器时，上图中虚线框部分的连线应遵从如下的连接方式，避免线间电容的负面影响。

- 连线要保证尽可能短。
 - 振荡器连线不要和其它的信号线互相交叉。
 - 布线时不要将振荡器连线靠近通过大波动电流的信号线
 - 保持振荡器电容的地电位和 V_{SS} 相同。
 - 不要将电容的接地点连接在大电流通过的地模式。
 - 不要从振荡器取信号。
2. 为了降低功耗，副时钟振荡器被设计为低振幅电路，而比起主时钟振荡器更容易受由于噪声引起的干扰。所以，在使用副时钟时需要特别注意布线的方式。
 3. 有关谐振器的选择和振荡器常量，需要用户自行评估振荡特性或者向谐振器生产商申请对谐振器的评估。

33.4.3 PLL 特性

(TA = -40 至 +85°C, VDD = EVDD= UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
输入频率	f _x		2.5		5	MHz
输出频率	f _{xx}	时钟直通模式	3		6	MHz
		PLL 模式 (×8)	24		48	MHz
入锁时间	t _{PLL}				800	μs

33.4.4 内部振荡器特性

(TA = -40 至 +85°C, VDD = EVDD= UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
输出频率	f _r		100	220	400	kHz

33.5 DC 特性

33.5.1 输入/输出电平

(TA = -40 至 +85°C, VDD = EVDD = UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
输入电压, 高	V _{IH1}	RESET, FLMD0, P60 至 P65, P90 至 P915	0.8EV _{DD}		EV _{DD}	V
	V _{IH2}	P00 至 P05, P20 至 P25, P30 至 P35, P42, P50 至 P56	0.8EV _{DD}		5.5	V
	V _{IH3}	P36, P37, P40, P41, PCM2, PCM3	0.7EV _{DD}		5.5	V
	V _{IH4}	PDL0 至 PDL15, PDH0 至 PDH7, PCM0, PCM1, PCS0, PCS2, PCS3, PCT0, PCT1, PCT4, PCT6	0.7EV _{DD}		EV _{DD}	V
	V _{IH5}	UDPF, UDMF, UDPH, UDMH	2.0		UV _{DD}	V
	V _{IH6}	P70 至 P711	0.7AV _{REF0}		AV _{REF0}	V
	V _{IH7}	P10, P11	0.7AV _{REF1}		AV _{REF1}	V
输入电压, 低	V _{IL1}	RESET, FLMD0, P60 至 P65, P90 至 P915	V _{SS}		0.2EV _{DD}	V
	V _{IL2}	P00 至 P05, P20 至 P25, P30 至 P35, P42, P50 至 P56	V _{SS}		0.2EV _{DD}	V
	V _{IL3}	P36, P37, P40, P41, PCM2, PCM3	V _{SS}		0.3EV _{DD}	V
	V _{IL4}	PDL0 至 PDL15, PDH0 至 PDH7, PCM0, PCM1, PCS0, PCS2, PCS3, PCT0, PCT1, PCT4, PCT6	V _{SS}		0.3EV _{DD}	V
	V _{IL5}	UDPF, UDMF, UDPH, UDMH	V _{SS}		0.8	V
	V _{IL6}	P70 至 P711	AV _{SS}		0.3AV _{REF0}	V
	V _{IL7}	P10, P11	AV _{SS}		0.3AV _{REF1}	V
输入漏电流, 高	I _{LIH}	V _I = V _{DD} = EV _{DD} = UV _{DD} = AV _{REF0} = AV _{REF1}			5	μA
输入漏电流, 低	I _{LIL}	V _I = 0 V			-5	μA
输出漏电流, 高	I _{LOH}	V _O = V _{DD} = EV _{DD} = UV _{DD} = AV _{REF0} = AV _{REF1}			5	μA
输出漏电流, 低	I _{LOL}	V _O = 0 V			-5	μA

备注 除非另外说明, 复用功能引脚的特性与端口引脚的特性相同。

(TA = -40 至+85°C, VDD = EVDD = UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件		最小值	典型值	最大值	单位
输出电压, 高	VOH1	注 1	每个引脚 I _{OH} = -1.0 mA	EV _{DD} - 1.0		EV _{DD}	V
			每个引脚 I _{OH} = -100 μA	EV _{DD} - 0.5		EV _{DD}	V
	VOH2	P70 至 P711	每个引脚 I _{OH} = -0.4 mA	AV _{REF0} - 1.0		AV _{REF0}	V
			每个引脚 I _{OH} = -100 μA	AV _{REF0} - 0.5		AV _{REF0}	V
	VOH3	P10, P11	每个引脚 I _{OH} = -0.4 mA	AV _{REF1} - 1.0		AV _{REF1}	V
			每个引脚 I _{OH} = -100 μA	AV _{REF1} - 0.5		AV _{REF1}	V
	VOH4	UDPF, UDMF, UDPH, UDMH	PL = 1.5 kΩ (V _{SS} 已连接)	2.8			V
			每个引脚 I _{OH} = -100 μA	UV _{DD} - 0.5		UV _{DD}	V
输出电压, 低	VOL1	注 2	每个引脚 I _{OL} = 1.0 mA	0		0.4	V
	VOL2	P36, P37, P40, P41, P90, P91	每个引脚 I _{OL} = 3.0 mA	0		0.4	V
	VOL3	P70 至 P711	每个引脚 I _{OL} = 1.0 mA	0		0.4	V
	VOL4	P10, P11	每个引脚 I _{OL} = 0.4 mA	0		0.4	V
	VOL5	UDPF, UDMF, UDPH, UDMH	PL = 1.5 kΩ (UV _{DD} 已连接)	0		0.3	V
软件下拉电阻	R1	P56	V _I = V _{DD}	10	30	100	kΩ

- 注
1. P00 至 P05, P20 至 P25, P30 至 P37, P40 至 P42, P50 至 P56, P60 至 P65, P90 至 P915, PCM0 至 PCM3, PCS0, PCS2, PCS3
PCT0, PCT1, PCT4, PCT6, PDH0 至 PDH7, PDL0 至 PDL15
 2. P00 至 P05, P20 至 P25, P30 至 P35, P42, P50 至 P56, P60 至 P65, P92 至 P915, PCM0 至 PCM3, PCS0, PCS2, PCS3, PCT0, PCT1, PCT4, PCT6, PDH0 至 PDH7, PDL0 至 PDL15

- 备注
1. 除非另外说明, 复用功能引脚的特性与端口引脚的特性相同。
 2. 当 I_{OH} 和 I_{OL} 条件不满足单个引脚的要求, 但是满足所有引脚的总值时, 则只有该引脚不满足 DC 特性。

33.5.2 供电电流

(T_A = -40 至 +85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号	条件		最小值	典型值	最大值	单位
供电电流 ^{1, 2}	I _{DD1}	正常操作	f _{XX} = 48 MHz (f _X = 6 MHz) 外设功能工作		60	120	mA
	I _{DD2}	HALT 模式	f _{XX} = 48 MHz (f _X = 6 MHz) 外设功能工作		42	84	mA
	I _{DD3}	IDLE1 模式	f _{XX} = 48 MHz (f _X = 6 MHz), PLL 打开		4	10	mA
	I _{DD4}	IDLE2 模式	f _{XX} = 6 MHz (f _X = 6 MHz), PLL 关闭		0.5	1	mA
	I _{DD5}	副时钟操作模式	f _{XT} = 32.768 kHz, 主时钟停止, 内部振荡器停止		120	600	μA
	I _{DD6}	Sub-IDLE 模式	f _{XT} = 32.768 kHz, 主时钟停止, 内部振荡器停止		13	150	μA
	I _{DD7}	STOP 模式	副时钟停止, 内部振荡器停止		10	130	μA
			副时钟工作, 内部振荡器停止		13	150	μA
I _{DD8}	Flash 存储器编程 模式	f _{XX} = 48 MHz (f _X = 6 MHz)		65	130	mA	

- 注 1. V_{DD}, EV_{DD} 和 UV_{DD} 电流的总和。不包括流经输出缓冲器、A/D 转换器、D/A 转换器和片上下拉电阻的电流。
2. 典型值的 V_{DD} 是 3.3 V。

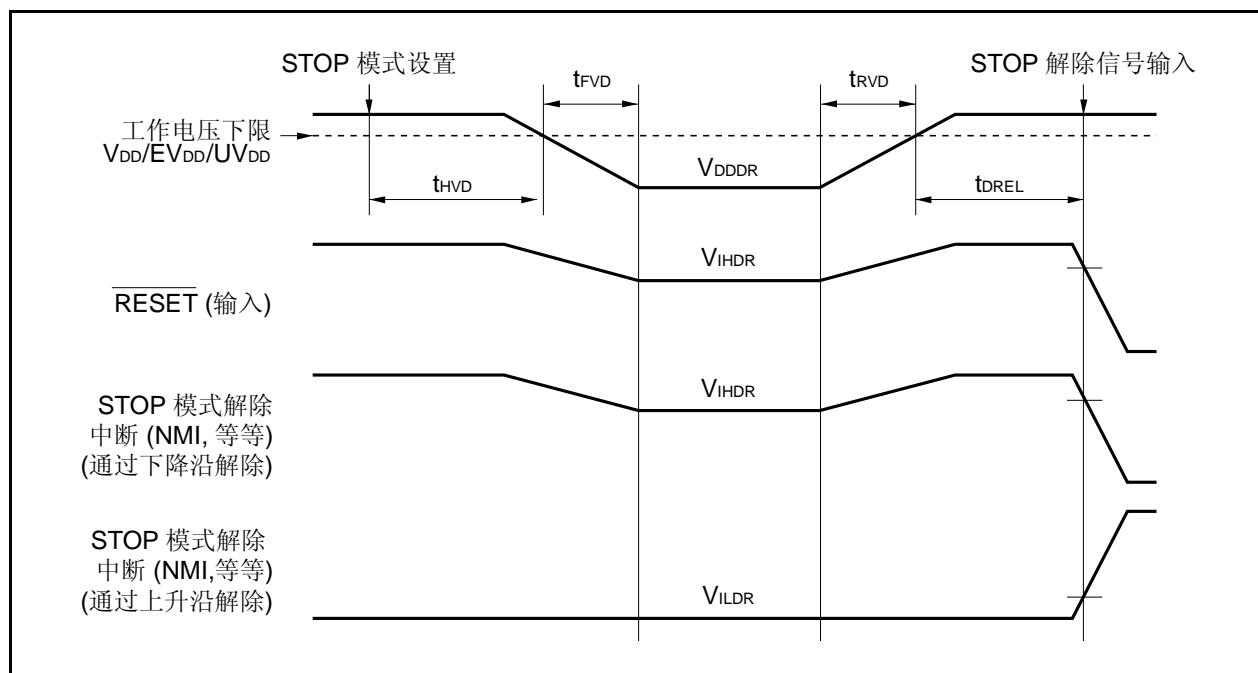
33.6 数据保持特性

(1) STOP 模式下

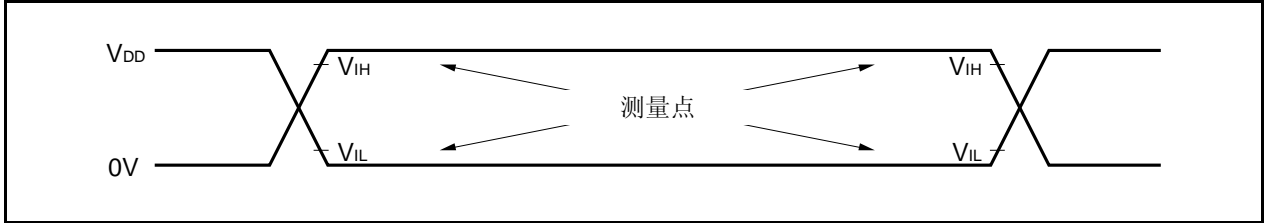
(TA = -40 至 +85°C, VDD = EVDD = UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
数据保持电压	VDDDR	STOP 模式 (所有功能停止)	1.9		3.6	V
数据保持电流	IDDDR	STOP 模式 (所有功能停止), VDDDR = 2.0 V		10	130	μA
供应电压上升时间	trVD		200			μs
供应电压下降时间	tFVD		200			μs
供应电压保持时间	tHVD	设置STOP 模式后	0			ms
STOP 解除信号输入时间	tDREL	VDD 达到2.85 V (最小值) 之后	0			ms
数据保持输入电压, 高	VIHDR	VDD = EVDD = UVDD = VDDDR	0.9VDDDR		VDDDR	V
数据保持输入电压, 低	VILDR	VDD = EVDD = UVDD = VDDDR	0		0.1VDDDR	V

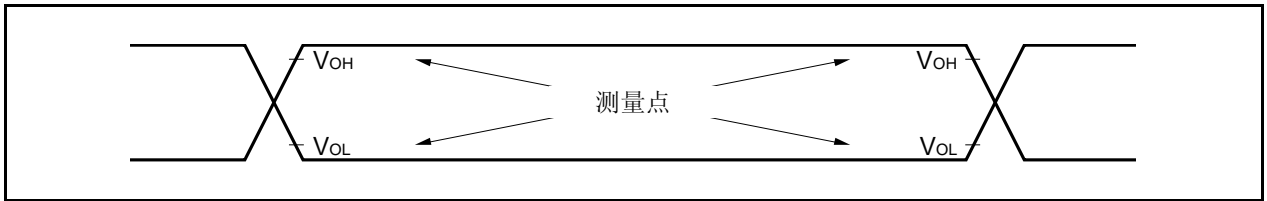
注意事项 转换到 STOP 模式以及从 STOP 模式中恢复都必须在额定工作范围内进行。



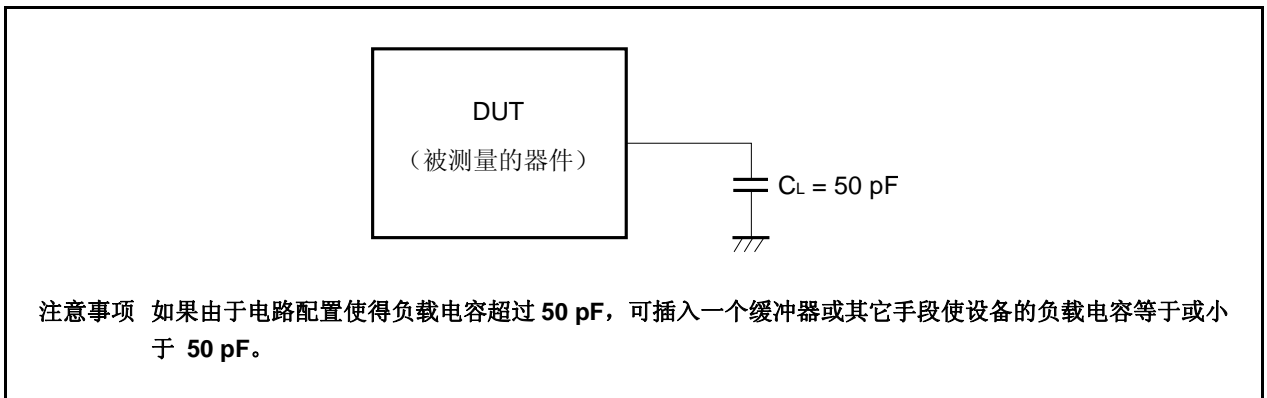
33.7 AC特性

(1) AC 测试输入测量点 (V_{DD} , AV_{REF0} , AV_{REF1} , EV_{DD})

(2) AC 测试输出测量点



(3) 负载条件

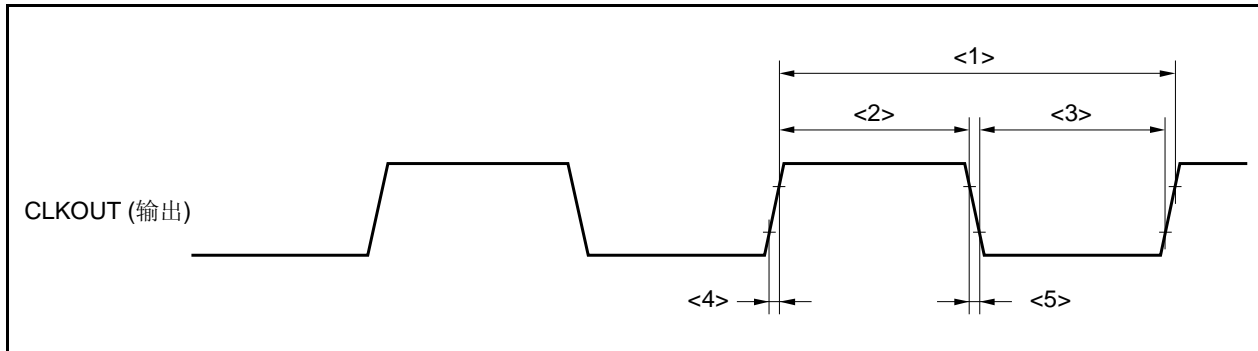


33.7.1 CLKOUT 输出时序

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	最大值	单位
输出周期	t_{CYK}	<1>	20.83 ns	31.25 μs	
高电平宽度	t_{WKH}	<2>	$t_{CYK}/2 - 6$		ns
低电平宽度	t_{WKL}	<3>	$t_{CYK}/2 - 6$		ns
上升时间	t_{KR}	<4>		6	ns
下降时间	t_{KF}	<5>		6	ns

时钟时序



33.7.2 总线时序

(1) 复用总线模式/分离总线模式

(a) 读/写周期 (CLKOUT 异步)

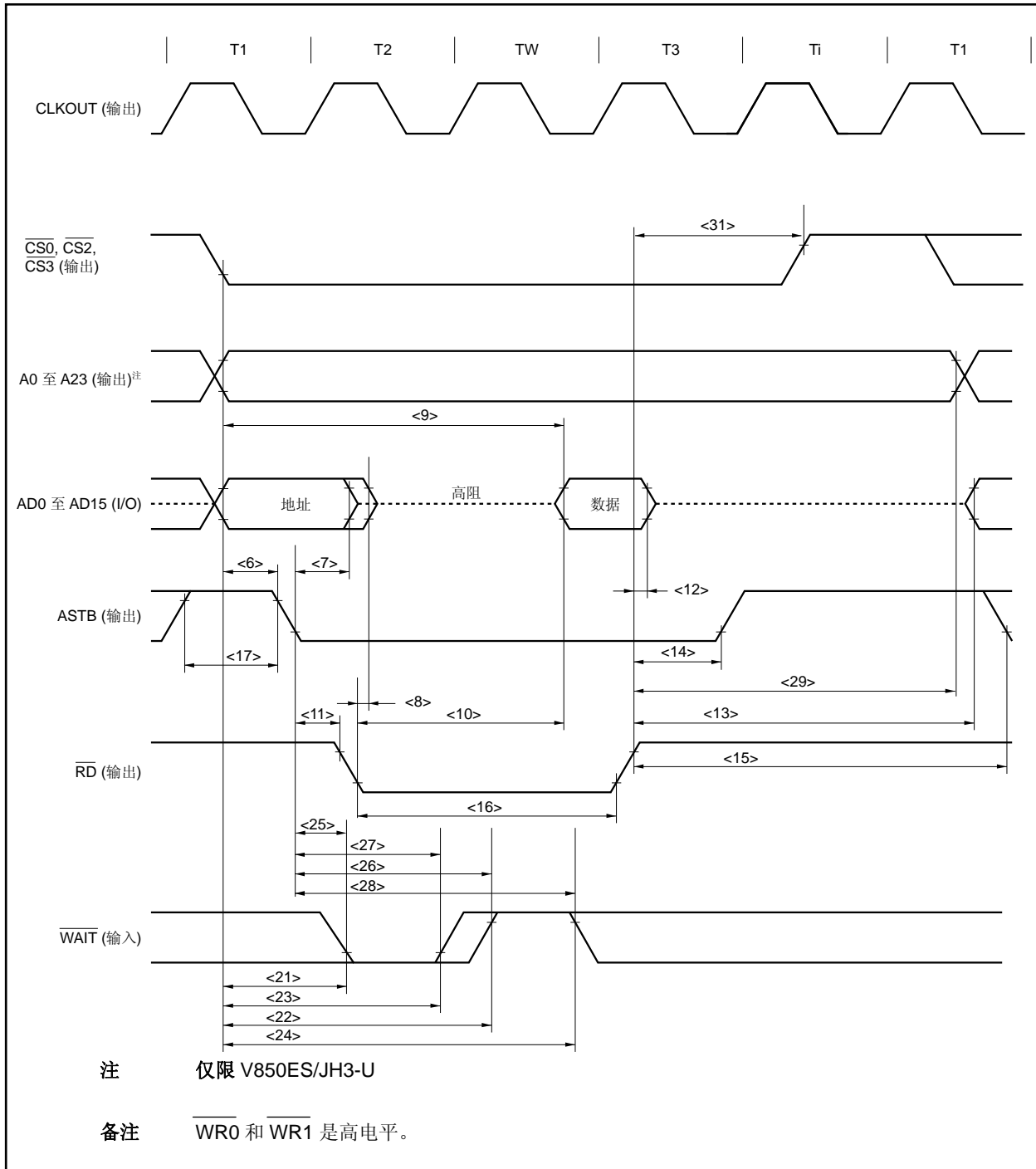
($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	最大值	单位
地址建立时间 (至 $ASTB\downarrow$)	t_{DAST}	<6>	$(0.5 + t_{ASw})T - 9$		ns
地址保持时间 (从 $ASTB\downarrow$ 起)	t_{HSTA}	<7>	$(0.5 + t_{AHw})T - 8$		ns
自 $RD\downarrow$ 至地址悬空的延迟时间	t_{FRDA}	<8>		5	ns
自地址有效到数据输入建立时间	t_{DAID}	<9>		$(2 + n + t_{ASw} + t_{AHw})T - 25$	ns
自 $RD\downarrow$ 到数据输入建立时间	t_{DRDID2}	<10>		$(1 + n)T - 15$	ns
从 $ASTB\downarrow$ 到 $RD\downarrow$ 的延迟时间	t_{DSTRD}	<11>	$(0.5 + t_{AHw})T - 4$		ns
从 $ASTB\downarrow$ 到 $WRm\downarrow$ 的延迟时间	t_{DSTWR}				
数据输入保持时间 (自 $RD\uparrow$ 起)	t_{HRDID}	<12>	0		ns
自 $RD\uparrow$ 到地址输出延迟时间	t_{DRDOD}	<13>	$(1 + i)T - 3$		ns
从 $RD\uparrow$ 至 $ASTB\uparrow$ 的延迟时间	t_{DRDST}	<14>	0.5T - 5		ns
从 $WRm\uparrow$ 到 $ASTB\uparrow$ 的延迟时间	t_{DWRST}				
从 $RD\uparrow$ 到 $ASTB\downarrow$ 的延迟时间	t_{DRDST}	<15>	$(1.5 + i + t_{ASw})T - 4$		ns
RD 低电平宽度	t_{WRDL}	<16>	$(1 + n)T - 10$		ns
WRm 低电平宽度	t_{WWRL}				
$ASTB$ 高电平宽度	t_{WSTH}	<17>	$(1 + i + t_{ASw})T - 10$		ns
自 $WRm\downarrow$ 起的数据输出延迟时间	t_{DWROD}	<18>		9	ns
数据输出延迟时间 (自 $WRm\uparrow$)	t_{DODWR}	<19>	$(1 + n)T - 11$		ns
数据输出保持时间 (自 $WRm\uparrow$ 起)	t_{HWROD}	<20>	T - 3		ns
$WAIT$ 建立时间 (至地址有效)	t_{SAWT1}	<21>	$n \geq 1$	$(1.5 + t_{ASw} + t_{AHw})T - 25$	ns
	t_{SAWT2}	<22>		$(1.5 + n + t_{ASw} + t_{AHw})T - 25$	ns
$WAIT$ 保持时间 (自地址有效起)	t_{HAWT1}	<23>	$n \geq 1$	$(0.5 + n + t_{ASw} + t_{AHw})T$	ns
	t_{HAWT2}	<24>		$(1.5 + n + t_{ASw} + t_{AHw})T$	ns
$WAIT$ 建立时间 (至 $ASTB\downarrow$)	t_{SSTWT1}	<25>	$n \geq 1$	$(1 + t_{AHw})T - 15$	ns
	t_{SSTWT2}	<26>		$(1 + n + t_{AHw})T - 15$	ns
$WAIT$ 保持时间 (自 $ASTB\downarrow$ 起)	t_{HSTWT1}	<27>	$n \geq 1$	$(n + t_{AHw})T - 2$	ns
	t_{HSTWT2}	<28>		$(1 + n + t_{AHw})T - 2$	ns
自 $RD\uparrow$ 到地址保持时间	t_{HRDA2}	<29>		$(1 + i)T - 5$	ns
自 $WRm\uparrow$ 到地址保持时间	t_{HWRA2}	<30>		T - 5	ns
从 $RD\uparrow$ 到 CSn 的保持时间	t_{HRDC2}	<31>	$i \geq 1$	T - 5	ns
从 $WRm\uparrow$ 到 CSn 的保持时间	t_{HWRC2}	<32>		T - 5	ns

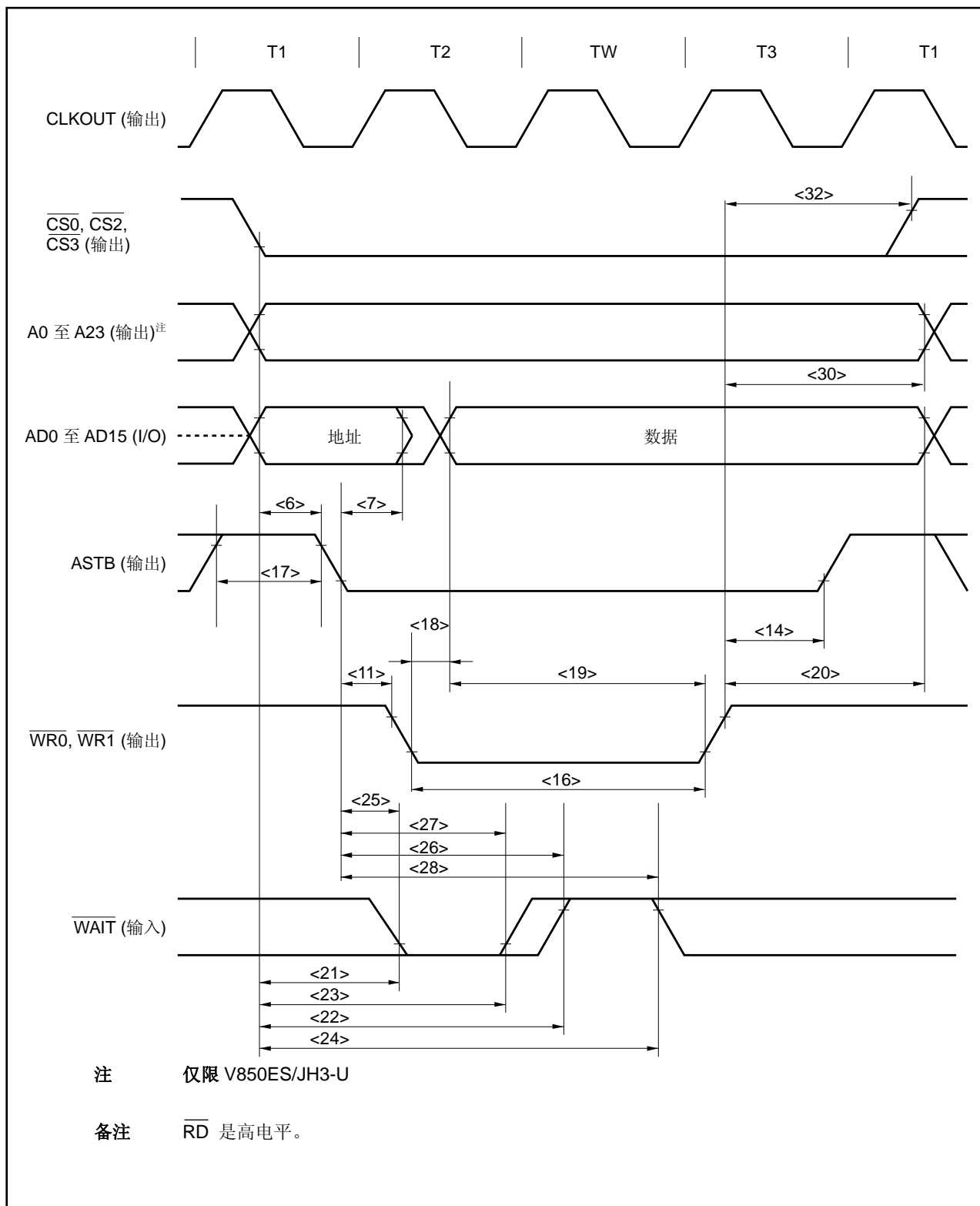
备注

1. t_{ASw} : 地址建立等待的时钟数目。
 t_{AHw} : 地址保持等待的时钟数目。
2. $T = 1/f_{CPU}$ (f_{CPU} : CPU 工作时钟频率)
3. n : 在总线周期中插入的等待时钟数目
当插入可编程等待时, 采样时序改变。
4. $m = 0, 1$
5. i : 读周期后插入的空闲状态数目 (0 或 1)
6. 在上述说明中的值是当占空比为 1: 1 的时钟从 X1 输入时的值。

读周期 (CLKOUT 异步)：复用总线模式/分离总线模式



写周期 (CLKOUT 异步)：复用总线模式/分离总线模式



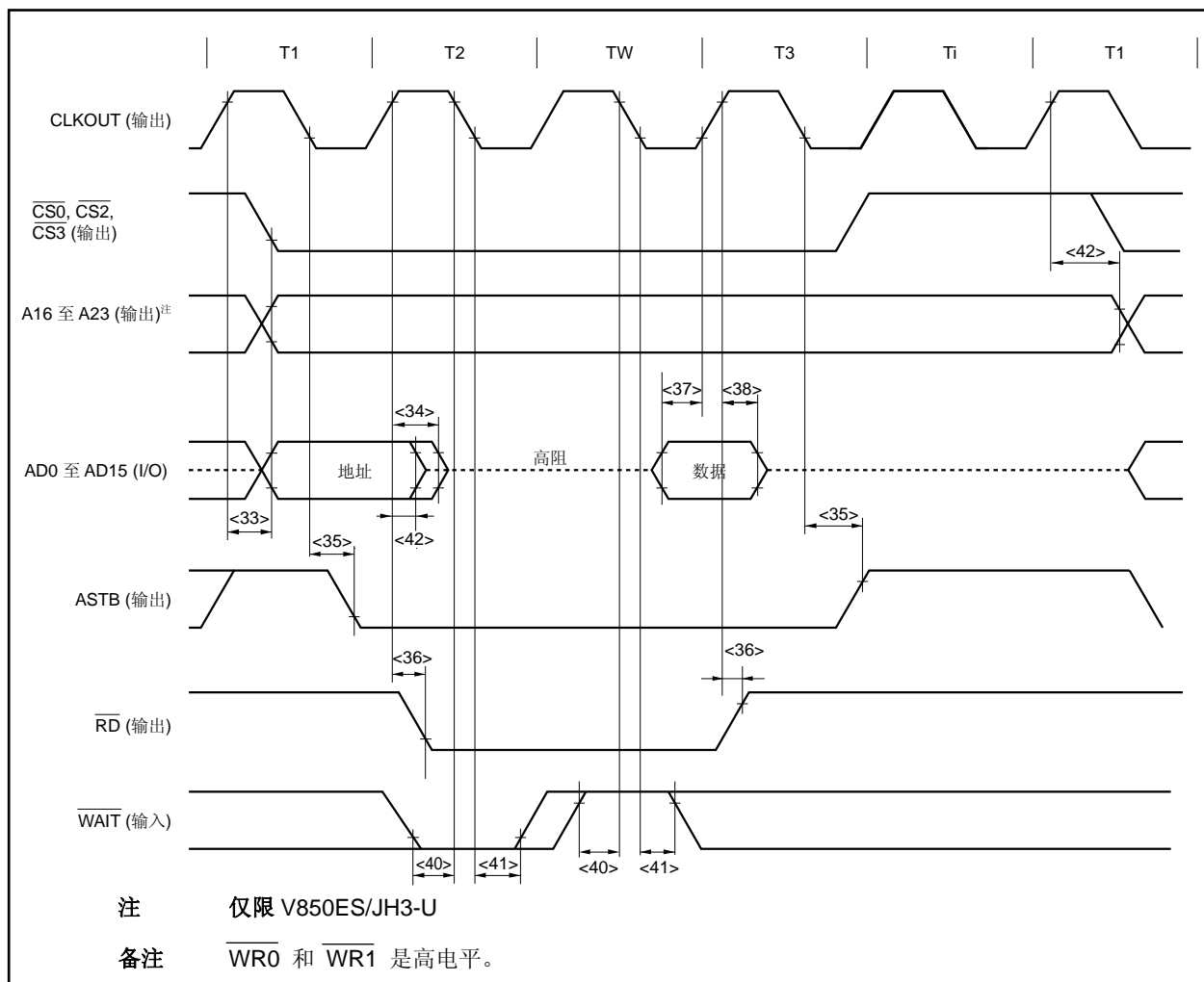
(b) 读/写周期 (CLKOUT 同步): 复用总线模式/分离总线模式

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

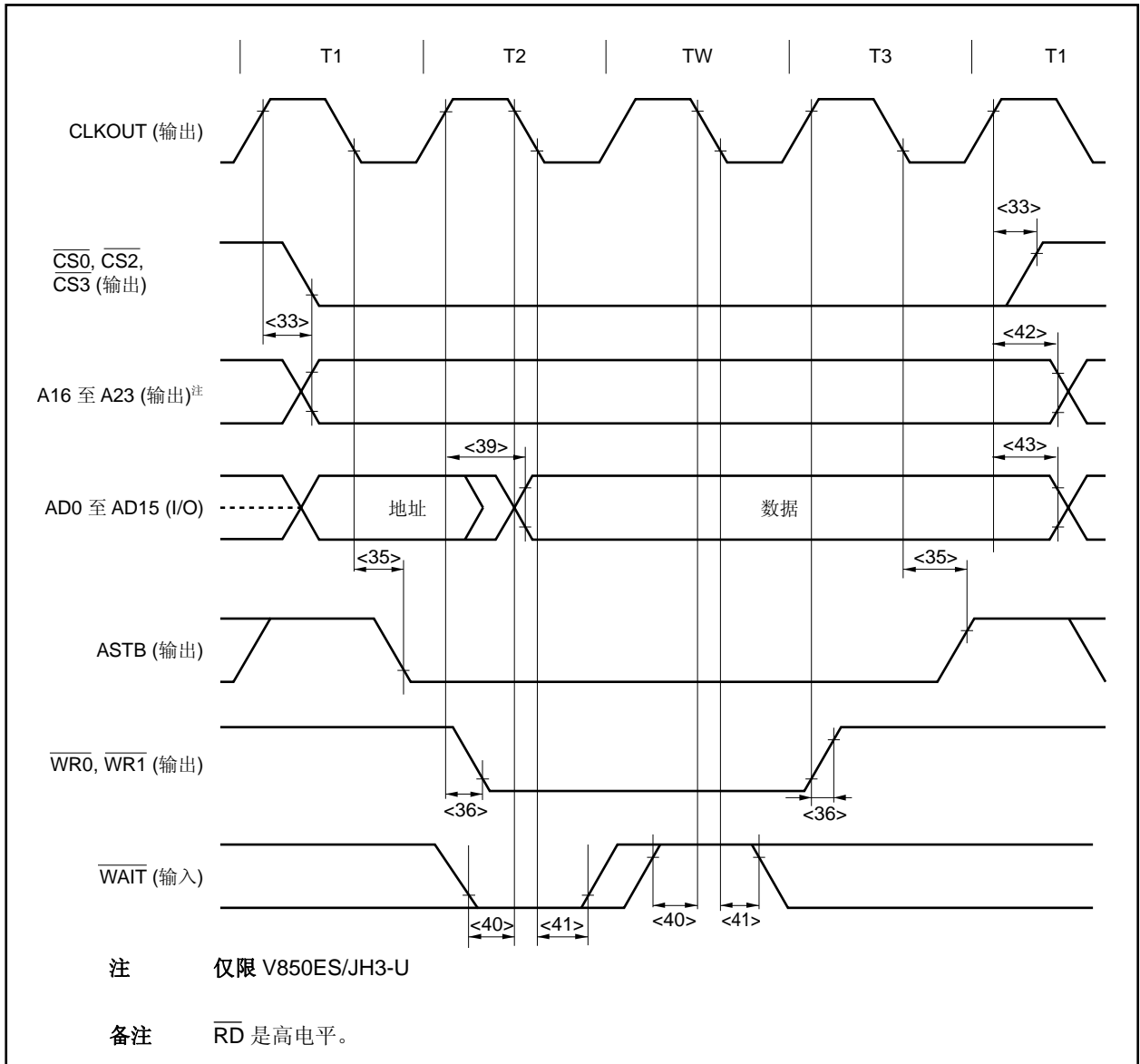
参数	符号	条件	最小值	最大值	单位
自 CLKOUT↑到地址有效的延迟时间	t_{DKA}	<33>	0	17	ns
自 CLKOUT↑到地址悬空延迟时间	t_{FKA}	<34>	0	15	ns
自 CLKOUT↓到 ASTB 的延迟时间	t_{DKST}	<35>	0	12	ns
自 CLKOUT↑到 \overline{RD} 、 \overline{WR}_m 的延迟时间	t_{DKRDWR}	<36>	0	12	ns
数据输入建立时间 (至 CLKOUT↑)	t_{SIDK}	<37>	16		ns
数据输入保持时间 (自 CLKOUT↑)	t_{HKID}	<38>	0		ns
自 CLKOUT↑起数据输出延迟时间	t_{DKOD}	<39>		17	ns
\overline{WAIT} 建立时间 (至 CLKOUT↓)	t_{SWTK}	<40>	16		ns
\overline{WAIT} 保持时间 (自 CLKOUT↓)	t_{HKWT}	<41>	0		ns
自 CLKOUT↑到地址保持时间	t_{HKA2}	<42>	0		ns
自 CLKOUT↑到数据输出保持时间	t_{HKDW}	<43>	0		ns

- 备注
1. $m = 0, 1$
 2. 在上述说明中的值是当占空比为 1: 1 的时钟从 X1 输入时的值。

读周期 (CLKOUT 同步): 复用总线模式/分离总线模式



写周期 (CLKOUT 同步): 复用总线模式/分离总线模式



(2) 总线保持期间 (仅限 V850ES/JH3-U)

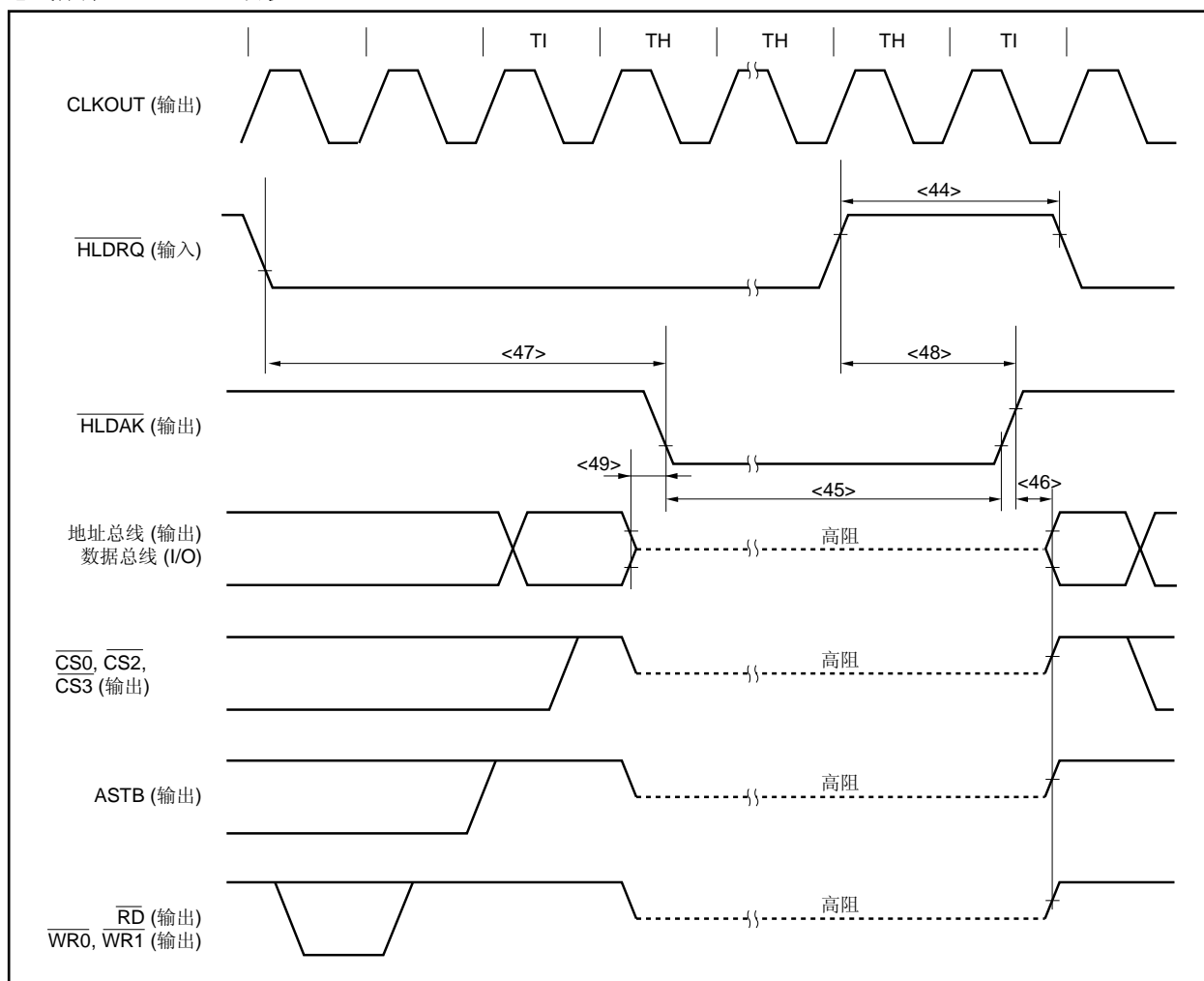
(a) CLKOUT 异步

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	最大值	单位
$\overline{\text{HLDRQ}}$ 高电平宽度	t_{WHQH}	<44>	$T + 16$		ns
$\overline{\text{HLDAK}}$ 低电平宽度	t_{WHAL}	<45>	$T - 10$		ns
自 $\overline{\text{HLDAK}}\uparrow$ 到总线输出的延时时间	t_{DHAC}	<46>	-7		ns
自 $\overline{\text{HLDRQ}}\downarrow$ 到 $\overline{\text{HLDAK}}\downarrow$ 的延时时间	t_{DHQHA1}	<47>	$2.5T$		ns
自 $\overline{\text{HLDRQ}}\uparrow$ 到 $\overline{\text{HLDAK}}\uparrow$ 的延时时间	t_{DHQHA2}	<48>	$0.5T + 17$	$1.5T + 31$	ns
从总线悬空到 $\overline{\text{HLDAK}}\downarrow$ 的延时时间	t_{DFHA}	<49>	0		ns

- 备注
1. $T = 1/f_{\text{CPU}}$ (f_{CPU} : CPU 工作时钟频率)
 2. n: 在总线周期中插入的等待时钟数目
当插入可编程等待时, 采样时序改变。
 3. 在上述说明中的值是当占空比为 1: 1 的时钟从 X1 输入时的值。

总线保持 (CLKOUT 异步)



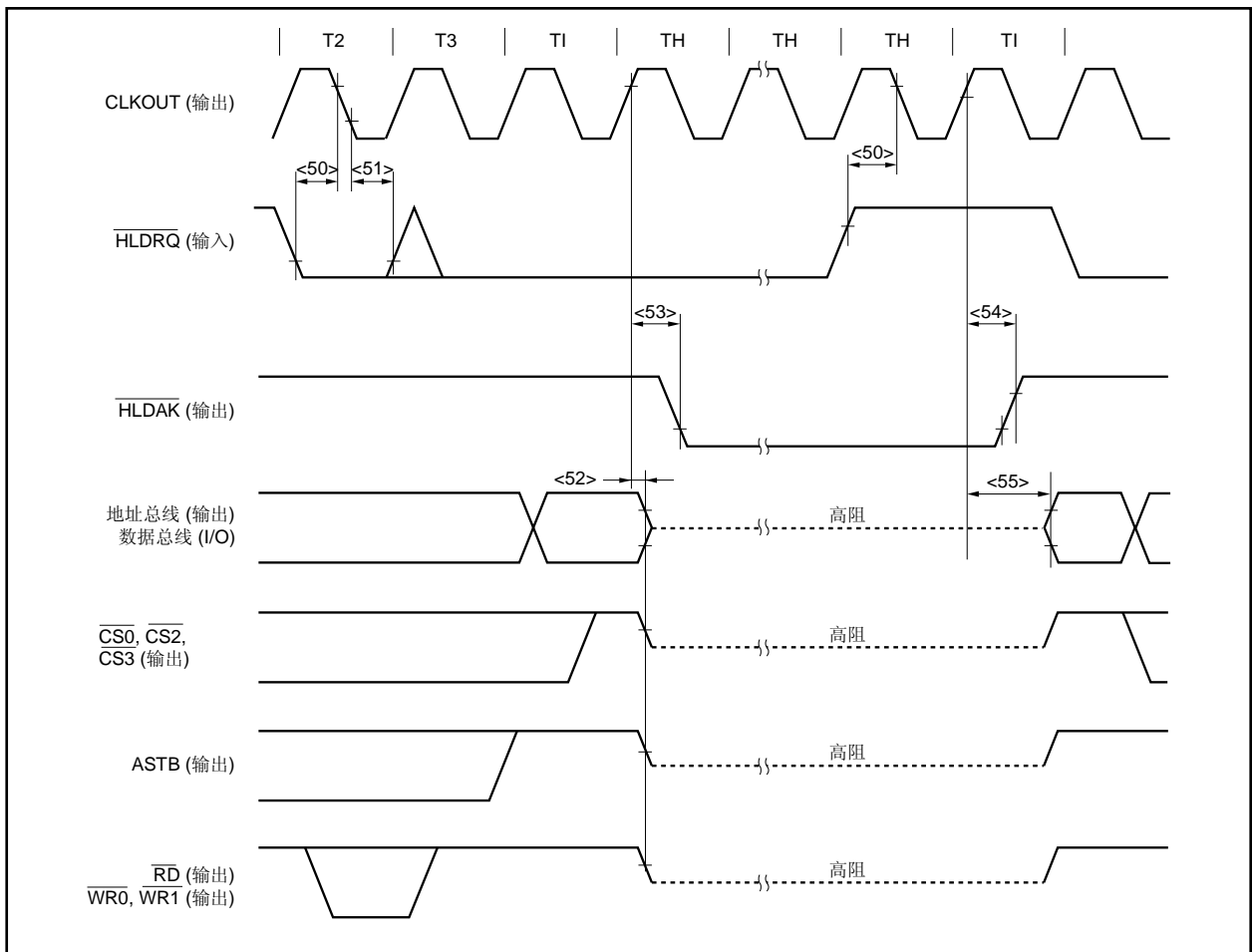
(b) CLKOUT 同步

(TA = -40 至 +85°C, VDD = EVDD = UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
HLD $\overline{\text{RQ}}$ 建立时间 (至 CLKOUT \downarrow)	t _{SHQK}	<50>	16		ns
HLD $\overline{\text{RQ}}$ 保持时间 (自 CLKOUT \downarrow)	t _{HKHQ}	<51>	5		ns
自 CLKOUT \uparrow 至总线悬空的延时时间	t _{DKF}	<52>	0	15	ns
自 CLKOUT \uparrow 至 HLD $\overline{\text{AK}}$ 的延时时间	t _{DKHA1}	<53>	1	15	ns
从 CLKOUT \uparrow 到 HLD $\overline{\text{AK}}$ \uparrow 的延时时间	t _{DKHA2}	<54>	1	15	ns
从 CLKOUT \uparrow 到数据输出的延时时间	t _{DKBO}	<55>	1	17	ns

备注 在上述说明中的值是当占空比为 1:1 的时钟从 X1 输入时的值。

总线保持 (CLKOUT 同步)



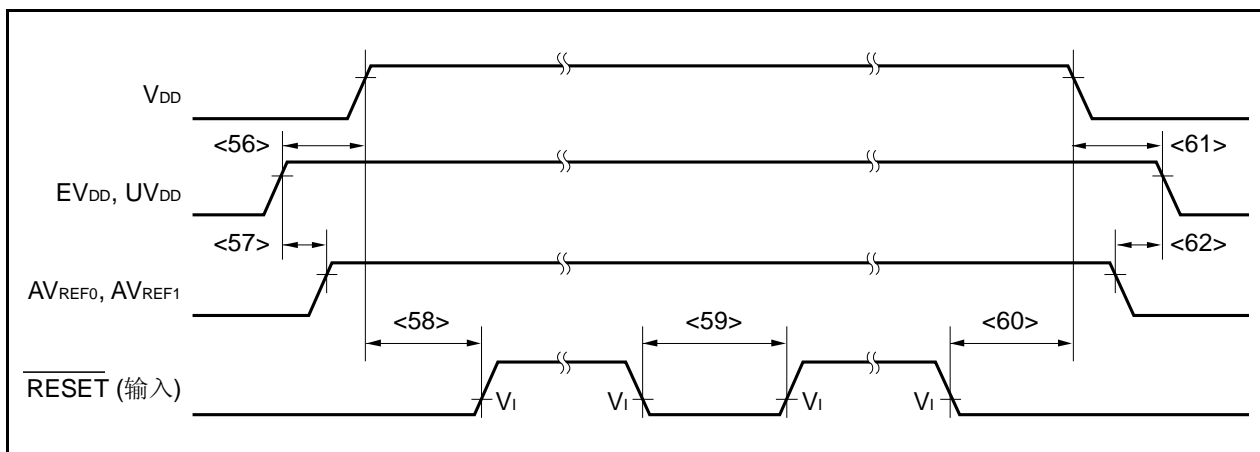
33.8 基本操作

(1) 电源开启/关闭/复位时序

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	最大值	单位
从 EV_{DD} , UV_{DD} 上升到 V_{DD} 的时间	t_{REL} <56>		0		ns
从 EV_{DD} , UV_{DD} 上升到 AV_{REF0} , AV_{REF1} 的时间	t_{REA} <57>		0	t_{REL}	ns
从 V_{DD} 上升到 \overline{RESET} 的时间	t_{RER} <58>		$500 + t_{REG}^{*}$		ns
\overline{RESET} 低电平宽度	t_{WRSL} <59>	模拟噪声消除 (flash 擦除/写入期间)	500		ns
		模拟噪声消除	500		ns
从 \overline{RESET} 下降到 V_{DD} 的时间	t_{FRE} <60>		500		ns
从 V_{DD} 下降到 EV_{DD} , UV_{DD} 的时间	t_{FEL} <61>		0		ns
从 AV_{REF0} , AV_{REF1} 降至 EV_{DD} , UV_{DD} 的时间	t_{FEA} <62>		0	t_{FEL}	ns

注 取决于片上稳压器的特性。



(2) 复位, 中断, FLMD0 引脚时序

(TA = -40 至 +85°C, VDD = EVDD = UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
RESET输入低电平宽度	tWRSL		500		ns
NMI 高电平宽度	tWNIH	模拟噪声消除	500		ns
NMI 低电平宽度	tWNIL	模拟噪声消除	500		ns
INTPn 高电平宽度	tWITH	n = 0 至18 (模拟噪声消除)	500		ns
		n = 2 (数字噪声消除)	3T _{SMP} + 20		ns
INTPn 低电平宽度	tWITL	n = 0 至18 (模拟噪声消除)	500		ns
		n = 2 (数字噪声消除)	3T _{SMP} + 20		ns

备注 T_{SMP}: 通过噪声消除控制寄存器 (INTNFC) 设置。可以选择 fxx/64, fxx/128, fxx/256, fxx/512 和 fxx/1024。

(3) 按键返回

(TA = -40 至 +85°C, VDD = EVDD = UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
KRn 高电平宽度	tWKRH	模拟噪声消除	500		ns
KRn 低电平宽度	tWKRL	模拟噪声消除	500		ns

备注 n = 0 至 7

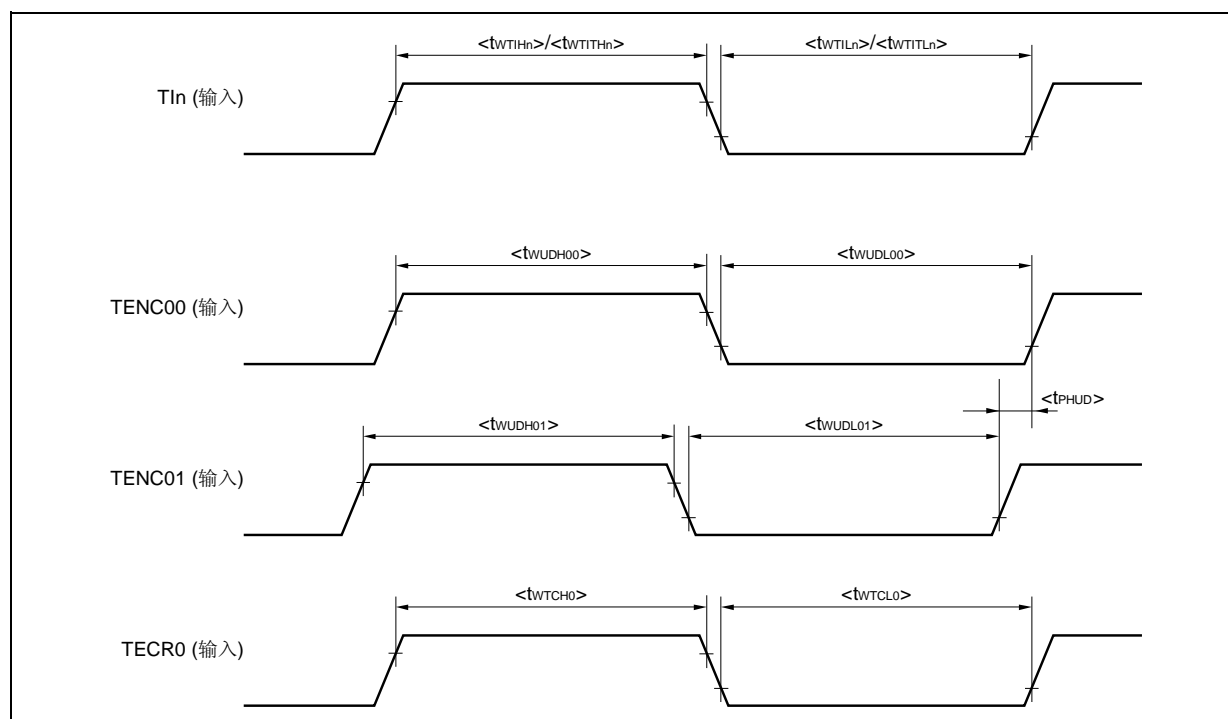
(4) 定时器时序

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	最大值	单位
TI 高电平宽度	t_{TIH}	TAB00 至 TAB03, TAB10 至 TAB13, EVTAB1, TRGAB1	$12T + 20$		ns
		TIAA00, TIAA01, TIAA10, TIAA11, TIAA20, TIAA21, TIAA30, TIAA31, TIAA50, TIAA51,	$3T_{SMP1} + 20$		ns
TI 低电平宽度	t_{TIL}	TAB00 至 TAB03, TAB10 至 TAB13, EVTAB1, TRGAB1	$12T + 20$		ns
		TIAA00, TIAA01, TIAA10, TIAA11, TIAA20, TIAA21, TIAA30, TIAA31, TIAA50, TIAA51,	$3T_{SMP1} + 20$		ns
TENCn 高电平宽度	t_{WENCHn}	$n = 0, 1$	$3T_{SMP2} + 20$		ns
TENCn 低电平宽度	t_{WENCLn}	$n = 0, 1$	$3T_{SMP2} + 20$		ns
TECR0 高电平宽度	t_{WCRH0}		$3T_{SMP2} + 20$		ns
TECR0 低电平宽度	t_{WCRL0}		$3T_{SMP2} + 20$		ns
TITn 高电平宽度	t_{WTITHn}	$n = 0, 1$	$3T_{SMP2} + 20$		ns
TITn 低电平宽度	t_{WTITLn}	$n = 0, 1$	$3T_{SMP2} + 20$		ns
EVTT0 高电平宽度	t_{WTITH0}		$3T_{SMP2} + 20$		ns
EVTT0 低电平宽度	t_{WTITL0}		$3T_{SMP2} + 20$		ns
TENCn 输入时间差别	t_{PHUD}	$n = 0, 1$	$3T_{SMP2} + 20$		ns

备注

1. $T = 1/f_{XX}$
2. T_{SMP1} : 通过噪声消除控制寄存器(TANFC)来设置。可以选择 f_{XX} 和 $f_{XX}/4$ 。
3. T_{SMP2} : 通过噪声消除控制寄存器(TTNFC)来设置。可以选择 $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$ 和 $f_{XX}/64$ 。
4. 上述规范表明可以作为有效边沿被准确检测的脉冲宽度。因此, 即使输入的脉冲宽度短于上述规范, 也可能被检测认为是一个有效边沿。



(5) UARTC 时序

(TA = -40 至 +85°C, VDD = EVDD = UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
发送速率				1.25	Mbps
ASCK0 周期时间				10	MHz

(6) CSIF 时序

(a) 主模式

[使用 CS10 至 CSIF2, 或 CSIF4 时]

(TA = -40 至 +85°C, VDD = EVDD = UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V, CL = 50 pF)

参数	符号		条件	最小值	最大值	单位
$\overline{\text{SCKFn}}$ 周期时间	t _{KCY1}	<63>		125		ns
$\overline{\text{SCKFn}}$ 高电平宽度	t _{KH1}	<64>		t _{KCY1} /2 - 8		ns
$\overline{\text{SCKFn}}$ 低电平宽度			t _{KL1}		t _{KCY1} /2 - 8	
SIFn 建立时间 (至 $\overline{\text{SCKFn}}\uparrow$)	t _{SIK1}	<65>		27		ns
SIFn 建立时间 (至 $\overline{\text{SCKFn}}\downarrow$)				27		ns
SIFn 保持时间 (自 $\overline{\text{SCKFn}}\uparrow$)	t _{SI1}	<66>		27		ns
SIFn 保持时间 (自 $\overline{\text{SCKFn}}\downarrow$)				27		ns
SOFn 输出延迟时间 (自 $\overline{\text{SCKFn}}\uparrow$)	t _{KSO1}	<67>			27	ns
SOFn 输出延迟时间 (自 $\overline{\text{SCKFn}}\downarrow$)						27
SOFn 输出保持时间 (自 $\overline{\text{SCKFn}}\uparrow$)	t _{HSO1}	<68>		t _{KCY1} /2 - 10		ns
SOFn 输出保持时间 (自 $\overline{\text{SCKFn}}\downarrow$)				t _{KCY1} /2 - 10		ns

备注 0 至 2, 4

[使用 CS13 时]

(TA = -40 至 +85°C, VDD = EVDD = UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V, CL = 50 pF)

参数	符号		条件	最小值	最大值	单位
$\overline{\text{SCKF3}}$ 周期时间	t _{KCYM}	<63>		83.3		ns
$\overline{\text{SCKF3}}$ 高电平宽度	t _{KHM}	<64>		t _{KCYM} /2 - 8		ns
$\overline{\text{SCKF3}}$ 低电平宽度				t _{KCYM} /2 - 8		ns
SIF3 建立时间 (至 $\overline{\text{SCKF3}}\uparrow$)	t _{SIKM}	<65>		16		ns
SIF3 建立时间 (至 $\overline{\text{SCKF3}}\downarrow$)				16		ns
SIF3 保持时间 (自 $\overline{\text{SCKF3}}\uparrow$)	t _{SIM}	<66>		16		ns
SIF3 保持时间 (自 $\overline{\text{SCKF3}}\downarrow$)				16		ns
SOF3 输出延迟时间 (自 $\overline{\text{SCKF3}}\uparrow$)	t _{SOM}	<67>			16	ns
SOF3 输出延迟时间 (自 $\overline{\text{SCKF3}}\downarrow$)						16
SOF3 输出保持时间 (自 $\overline{\text{SCKF3}}\uparrow$)	t _{HSOM}	<68>		t _{KCYM} /2 - 10		ns
SOF3 输出保持时间 (自 $\overline{\text{SCKF3}}\downarrow$)				t _{KCYM} /2 - 10		ns

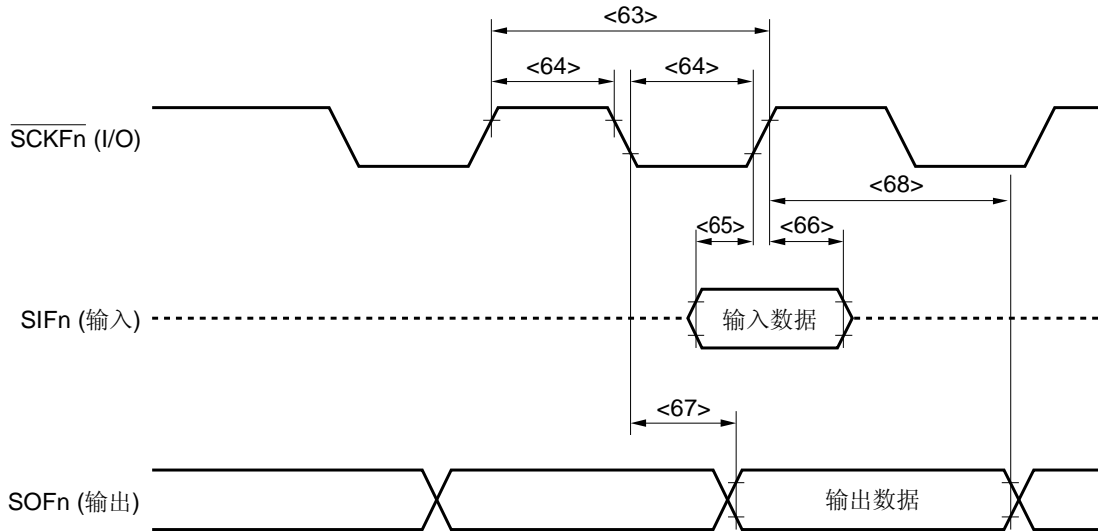
(b) 从模式

(T_A = -40 至 +85°C, V_{DD} = E_{VDD} = U_{VDD} = A_{VREF0} = A_{VREF1}, V_{SS} = A_{VSS} = 0 V, C_L = 50 pF)

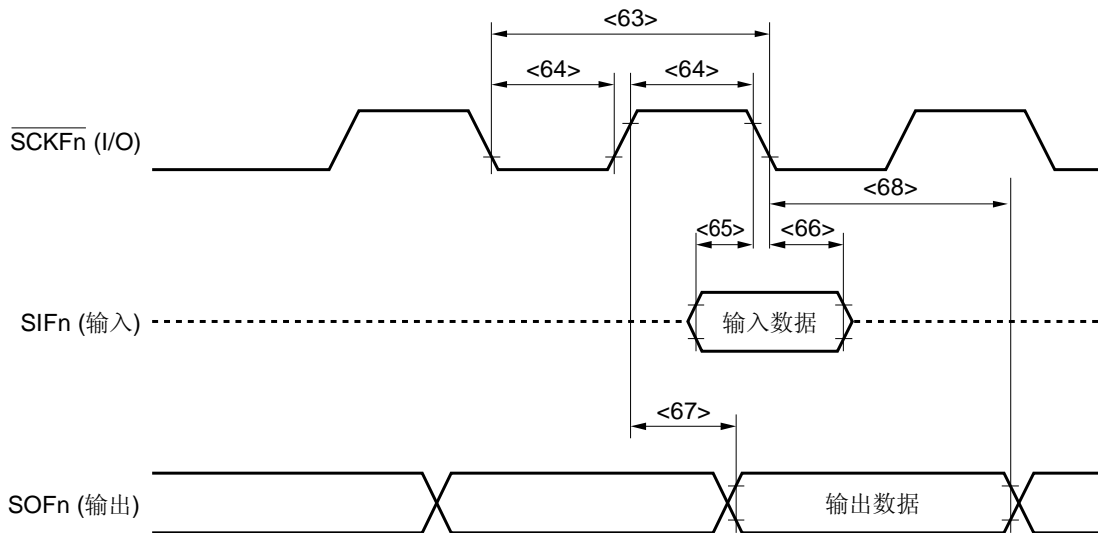
参数	符号	条件	最小值	最大值	单位
SCKFn 周期时间	t _{KCY2}	<63>	125		ns
SCKFn 高电平宽度	t _{KH2}	<64>	t _{KCYn} /2 - 8		ns
SCKFn 低电平宽度	t _{KL2}		t _{KCYn} /2 - 8		ns
SIFn 建立时间 (至 SCKFn↑)	t _{SIK2}	<65>	27		ns
SIFn 建立时间 (自 SCKFn↓)			27		ns
SIFn 保持时间 (至 SCKFn↑)	t _{SI2}	<66>	27		ns
SIFn 保持时间 (自 SCKFn↓)			27		ns
SOFn 输出延迟时间 (至 SCKFn↑)	t _{KSO2}	<67>		27	ns
SOFn 输出延迟时间 (自 SCKFn↓)				27	ns
SOFn 输出延迟时间 (至 SCKFn↑)	t _{HSO2}	<68>	t _{KCYn} /2 - 10		ns
SOFn 输出延迟时间 (自 SCKFn↓)			t _{KCYn} /2 - 10		ns

备注 n = 0 至 4

(a) CFnCTL1.CFnCKP, CFnDAP 位 = 00 or 11



(b) CFnCTL1.CFnCKP, CFnDAP 位 = 10 or 01



备注 n = 0 至 4

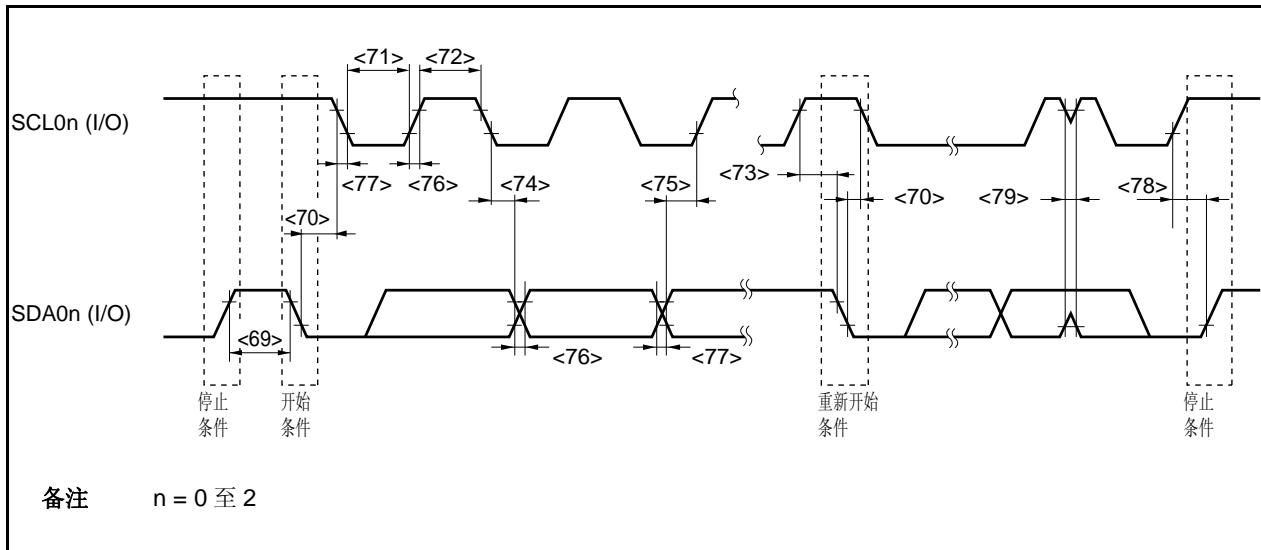
(7) I²C 总线模式(T_A = -40 至 +85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = AV_{SS} = 0 V)

参数	符号		普通模式		高速模式		单位
			最小值	最大值	最小值	最大值	
SCL0n 时钟频率	f _{CLK}		0	100	0	400	kHz
总线空闲时间 (开始条件和停止条件之间)	t _{BUF}	<69>	4.7	–	1.3	–	μs
保持时间 ^{#1}	t _{HD: STA}	<70>	4.0	–	0.6	–	μs
SCL0n 时钟低电平宽度	t _{LOW}	<71>	4.7	–	1.3	–	μs
SCL0n 时钟高电平宽度	t _{HIGH}	<72>	4.0	–	0.6	–	μs
开始/重新开始条件的建立时间	t _{SU: STA}	<73>	4.7	–	0.6	–	μs
数据保持时间	CBUS 兼容主机	t _{HD: DAT}	<74>	5.0	–	–	μs
	I ² C 模式			0 ^{#2}	–	0 ^{#2}	0.9 ^{#3}
数据建立时间	t _{SU: DAT}	<75>	250	–	100 ^{#4}	–	ns
SDA0n 和 SCL0n 信号上升时间	t _R	<76>	–	1000	20 + 0.1Cb ^{#5}	300	ns
SDA0n 和 SCL0n 信号下降时间	t _F	<77>	–	300	20 + 0.1Cb ^{#5}	300	ns
停止条件建立时间	t _{SU: STO}	<78>	4.0	–	0.6	–	μs
由输入滤波器抑制的毛刺脉冲宽度	t _{SP}	<79>	–	–	0	50	ns
每个信号总线的容性负载	Cb		–	400	–	400	pF

- 注
1. 满足开始条件时，第一个时钟脉冲在保持时间后产生。
 2. 系统内部 SDA0n（在 SCL0n 信号的 V_{IHmin} 时）信号需要最短 300 ns 的保持时间，以占用 SCL0n 信号下降沿时的不确定区域。
 3. 如果系统不延长 SCL0n 信号低电平保持时间（t_{LOW}），则只需要满足最大数据保持时间（t_{HD: DAT}）。
 4. 在正常模式 I²C 总线系统中可以适用高速模式 I²C 总线。这种情况下，设置高速模式 I²C 总线使其满足以下条件。
 - 如果系统不延长 SCL0n 信号低电平保持时间：
t_{SU: DAT} ≥ 250 ns
 - 如果系统延长 SCL0n 信号低电平保持时间：
释放 SCL0n 信号线之前，将下一个数据输出到 SDA0n 信号线（t_{Rmax} + t_{SU: DAT} = 1,000 + 250 = 1,250 ns：正常模式 I²C 总线规范）。
 5. Cb：每条总线的总电容（单位：pF）。

备注 n = 0 至 2

I²C 总线模式



(8) 高阻抗控制时序

(TA = -40 至 +85°C, VDD = EVDD = UVDD = AVREF0 = AVREF1, VSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
从振荡器停止到定时器输出高阻抗的时间	tCLM	时钟监视器工作		65	μs
从 TOAB1OFF 输入→ 定时器输出高阻抗的时间	tHTQn			300	ns
从 TOAA1OFF 输入→ 定时器输出高阻抗的时间	tHTP2			300	ns

(9) A/D 转换器

(TA = -40 至 +85°C, VDD = EVDD = UVDD = AVREF0 = AVREF1, 3.0 V ≤ AVREF0 ≤ 3.6 V, VSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
分辨率					10	位
总体误差 ^注		3.0 V ≤ AVREF0 ≤ 3.6 V			±0.6	%FSR
转换时间	tCONV		2.17		24	μs
零标称误差					±0.5	%FSR
满量程误差					±0.5	%FSR
非线性误差					±4.0	LSB
微分线性误差					±4.0	LSB
模拟输入电压	VIAN		AVSS		AVREF0	V
参考电压	AVREF0		3.0		3.6	V
AVREF0 电流	AIREF0	正常转换模式下		3	6.5	mA
		高速转换模式下		4	10	mA
		未使用 A/D 转换器时			5	μA

注 量化误差除外 (±0.05% FSR)。

注意事项 在 A/D 转换期间, 不要设置 (读/写) 复用功能端口, 否则转换分辨率可能会下降。

备注 LSB: 最低有效位
FSR: 满量程范围

(10) D/A 转换器

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}$, $3.0\text{ V} \leq AV_{REF1} \leq 3.6\text{ V}$, $V_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	典型值	最大值	单位
分辨率					8	位
总误差 ^{注1}		$R = 2\text{ M}\Omega$			± 1.2	%FSR
设置时间		$C = 20\text{ pF}$			3	μs
寄存器输出	R_o	输出数据 55H		6.42		$\text{k}\Omega$
参考电压	AV_{REF1}		3.0		3.6	V
AV_{REF1} 电流 ^{注2}	AI_{REF1}	D/A 转换工作		1	2.5	mA
		D/A 转换停止			5	μA

注 1. 量化误差除外 ($\pm 0.05\%$ FSR)。

2. D/A 转换器 1 通道的值

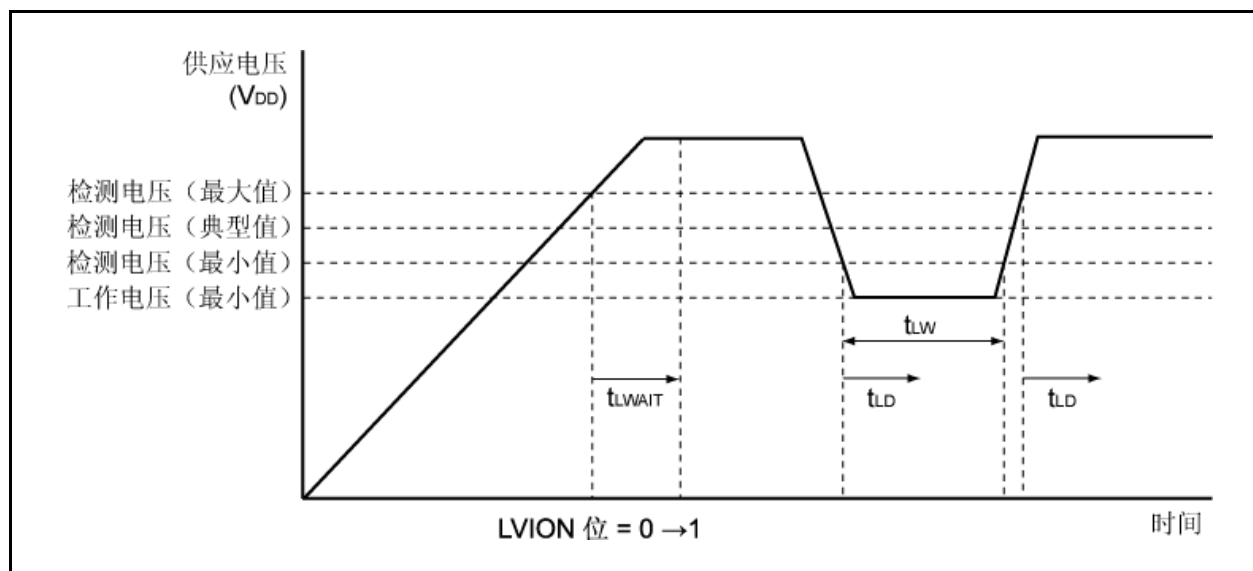
备注 R 是输出引脚负载电阻, C 是输出引脚负载电容。

(11) LVI 电路特性

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V_{LV10}		2.85	2.95	3.05	V
响应时间 ^注	t_{LD}	V_{DD} 达到 V_{LV10} (最大值) 之后, 或 V_{DD} 下降至 V_{LV10} (最大值) 之后		0.2	2.0	ms
最小脉冲宽度	t_{LW}		0.2			ms
参考电压稳定等待时间	t_{LWAIT}	V_{DD} 达到 2.85V (最小值) 之后		0.1	0.2	ms

注 检测到检测电压和输出一个中断或复位信号所需的时间。

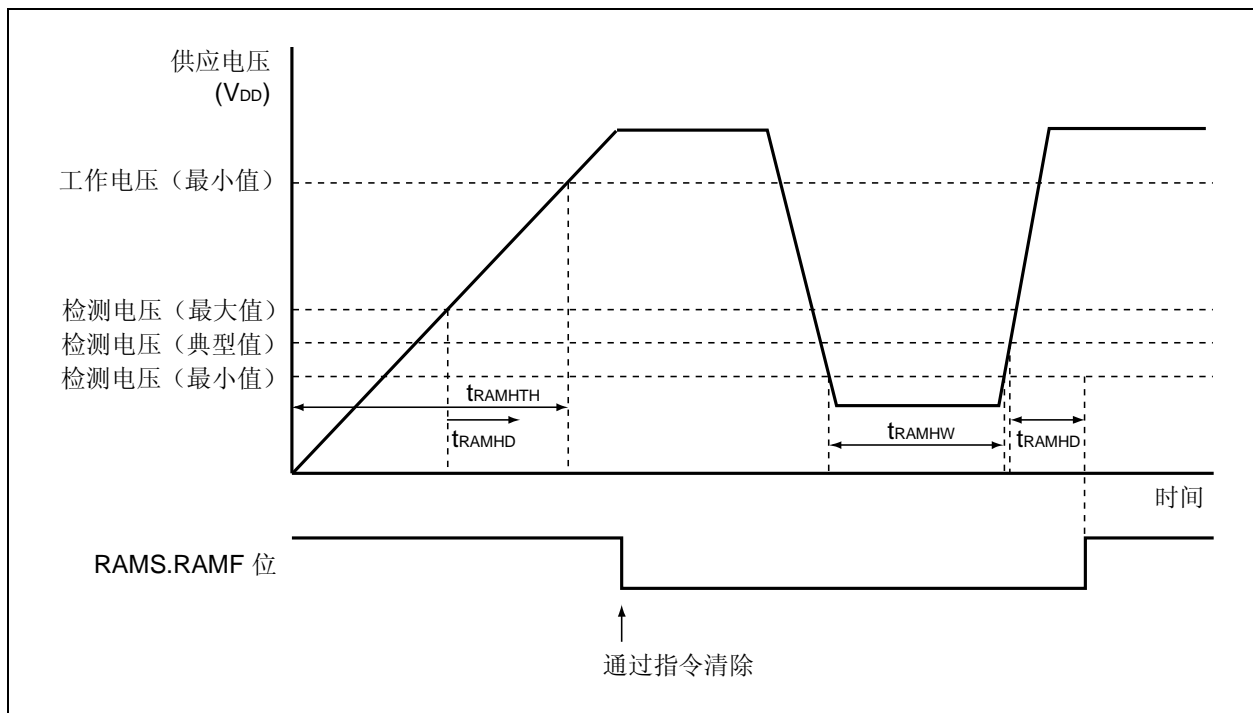


(12) RAM 保持检测

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V_{RAMH}		1.9	2.0	2.1	V
供电电压上升时间	t_{RAMHTH}	$V_{DD} = 0$ 至 2.85 V	0.002			ms
响应时间 ^注	t_{RAMHD}	V_{DD} 达到 2.1 V 之后		0.2	3.0	ms
最小脉冲宽度	t_{RAMHW}		0.2			ms

注 检测到检测电压和设置 RAMS.RAMF 位所需的时间。



33.9 Flash存储器编程特性

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

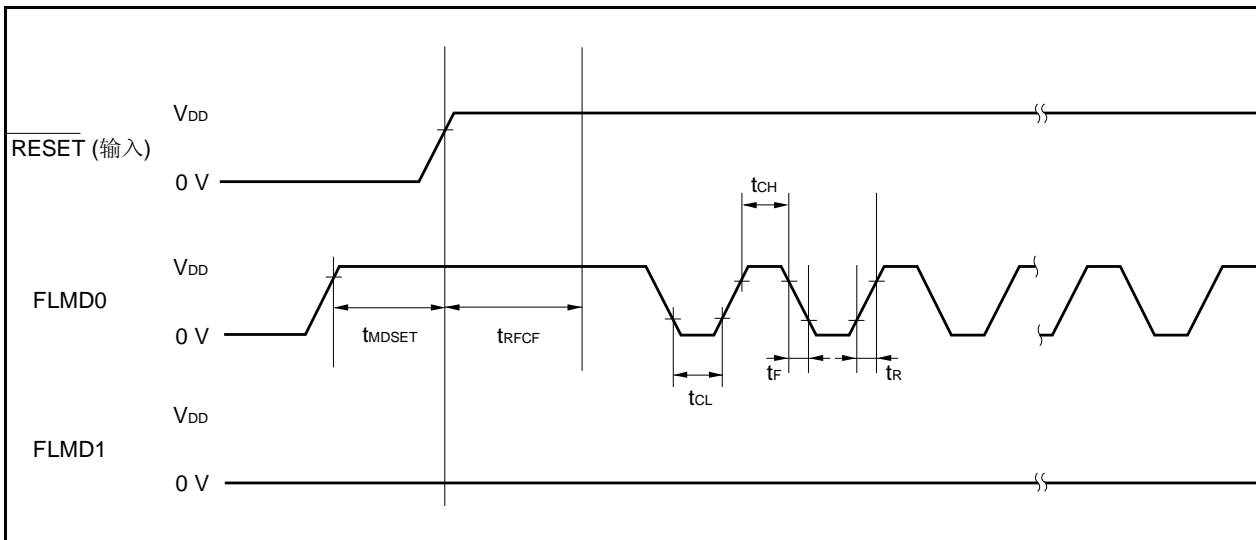
(1) 基本特性

参数	符号	条件	最小值	典型值	最大值	单位
工作频率	f_{CPU}		24		48	MHz
供电电压	V_{DD}		2.85		3.6	V
重写次数	C_{WRT}				100	次
编程温度	t_{PRG}		-40		+85	$^\circ\text{C}$

(2) 串行写入操作特性

参数	符号	条件	最小值	典型值	最大值	单位
FLMD0, FLMD1建立时间	t_{MDSET}		2		3000	ms
自 RESET \uparrow 到FLMD0 计数启动时间	t_{RFCF}	$f_x = 3$ 至 6 MHz	800			μs
FLMD0 计数器 高电平宽度/ 低电平宽度	$t_{\text{CH}}/t_{\text{CL}}$		10		100	μs
FLMD0 计数器上升时间/下降时间	$t_{\text{R}}/t_{\text{F}}$				1	μs

Flash 写入模式设置时序



(3) 编程特性

参数	符号	条件	最小值	典型值	最大值	单位
片擦除时间		f _{xx} = 48 MHz, 成批擦除		105		ms
每 256 字节的写入时间		f _{xx} = 48 MHz		2.0		ms
块内部验证时间		f _{xx} = 48 MHz		10		ms
块空白检查时间		f _{xx} = 48 MHz		0.5		ms
Flash 存储器信息设置时间		f _{xx} = 48 MHz		30		ms

注意事项 当首次对出厂产品进行写入时，不论是进行“擦除-写入”操作，还是“仅写入”操作，都被当作是一次重写。

举例 (P: 写入, E: 擦除):

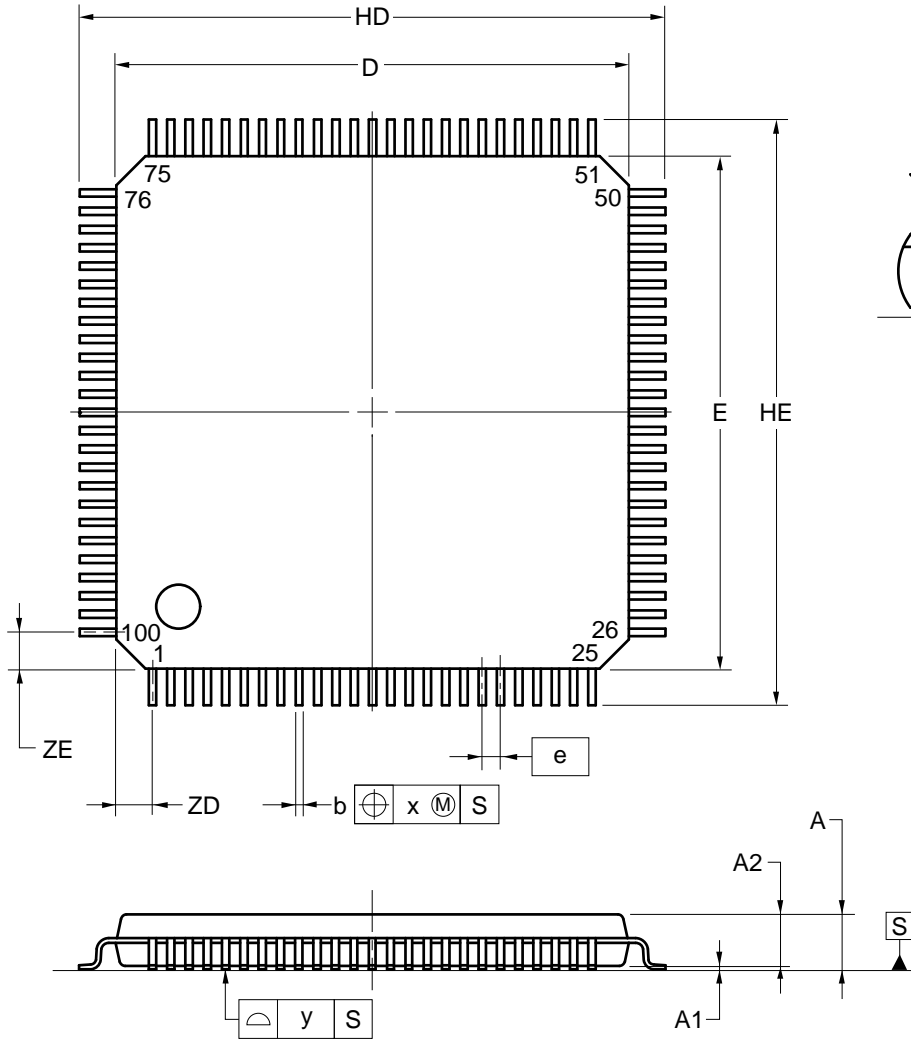
出厂产品 → P → E → P → E → P: 3 次重写

出厂产品 → E → P → E → P → E → P: 3 次重写

备注 每块的大小= 4 KB

第三十四章 封装图

100-针塑封 LQFP (密间距) (14x14)



引脚端的细节图

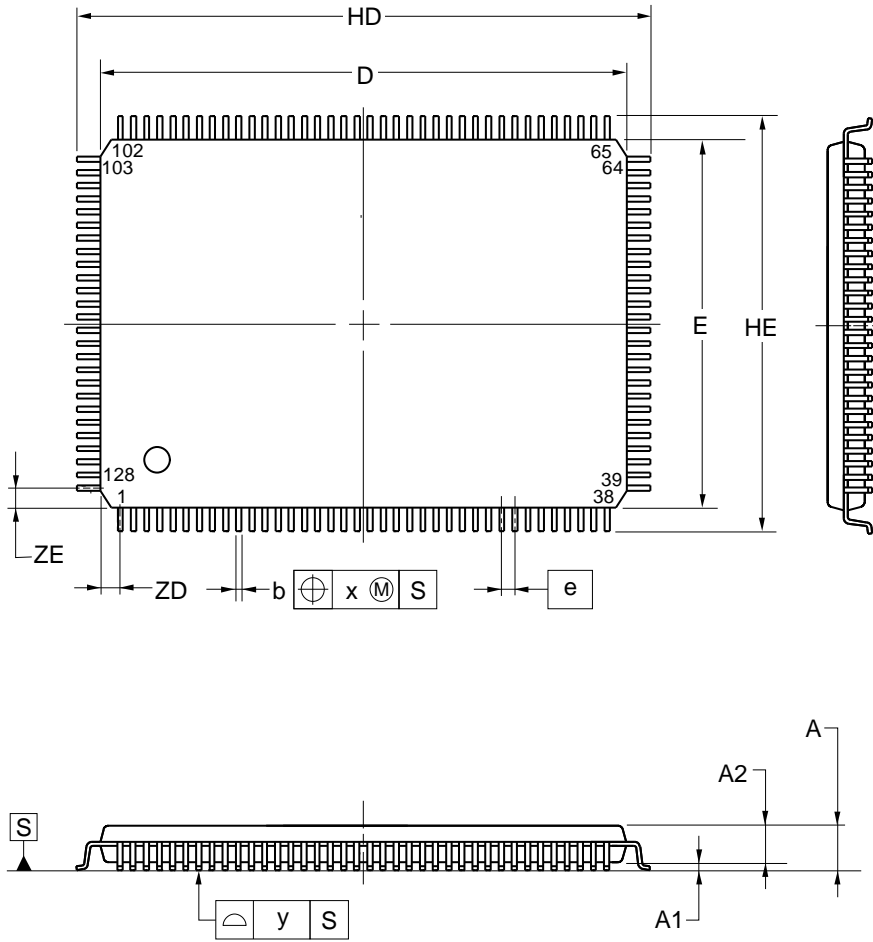
(单位:mm)

项目	尺寸
D	14.00±0.20
E	14.00±0.20
HD	16.00±0.20
HE	16.00±0.20
A	1.60 MAX
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.20 ^{+0.07} _{-0.03}
c	0.125 ^{+0.075} _{-0.025}
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
q	3° ^{+5°} _{-3°}
e	0.50
x	0.08
y	0.08
ZD	1.00
ZE	1.00

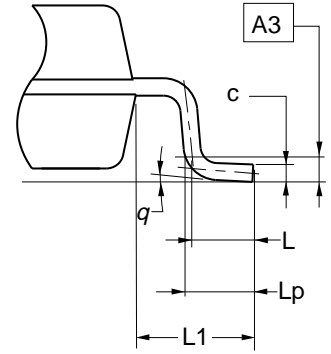
P100GC-50-UEU-1

© NEC Electronics Corporation 2007

128-针塑封 LQFP (密间距) (14x20)



引脚端的细节图



(单位: mm)

项目	尺寸
D	20.00±0.20
E	14.00±0.20
HD	22.00±0.20
HE	16.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.20 ^{+0.07} _{-0.03}
c	0.125 ^{+0.075} _{-0.025}
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
q	3° ^{+5°} _{-3°}
e	0.50
x	0.08
y	0.08
ZD	0.75
ZE	0.75

P128GF-50-GAT

注
在最大公差条件下，每个引脚的中心线位于其真实位置的 0.08 mm 范围之内。

© NEC Electronics Corporation 2006

附录 A 开发工具

以下开发工具可用来开发使用 V850ES/JG3-U 或 V850ES/JH3-U 单片机的系统。
图 A-1 显示了开发工具的配置。

- **支持 PC98-NX 系列**

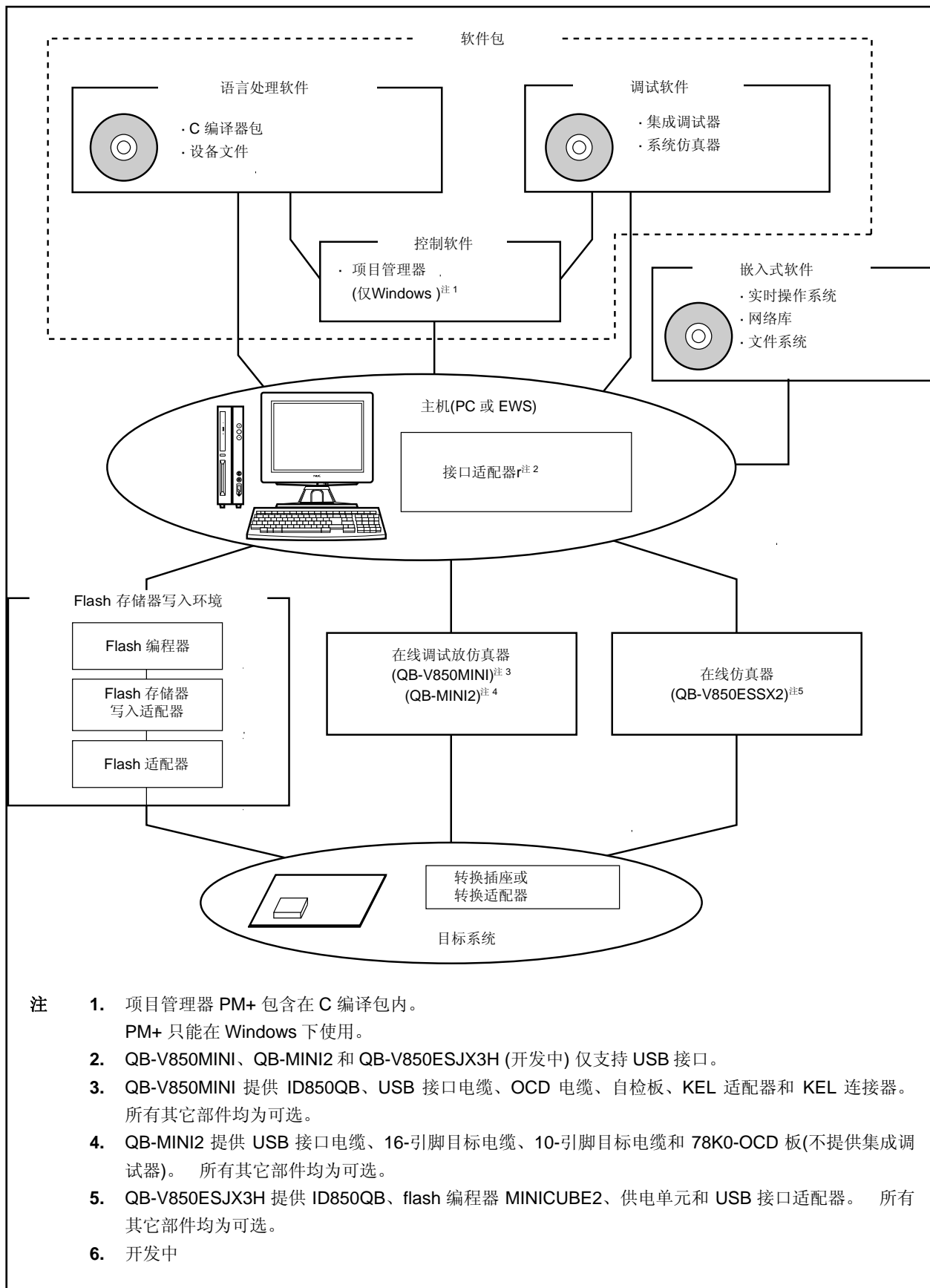
除非另行说明，IBM PC/AT™ 支持兼容的产品兼容于 PC98-NX 系列计算机。当使用 PC98-NX 系列计算机时，可参见 IBM PC/AT 兼容机说明。

- **Windows®**

除非另行说明，“Windows”指以下操作系统。

- Windows 98, 2000
- Windows Me
- Windows XP
- Windows NT® Ver. 4.0

图 A-1. 开发工具配置



A.1 软件包

SP850 V850 微控制器的软件包	本软件包包括用于 V850 微控制器的通用开发工具(软件) 型号: μ SxxxxSP850
------------------------	---

备注: xxxx 型号随使用的主机和操作系统的不同而异。

μ SxxxxSP850

xxxx	主机	OS	提供媒介
AB17	PC-9800 系列, IBM PC/AT 兼容型	Windows (日语版)	CD-ROM
BB17		Windows (英语版)	

A.2 语言处理软件

CA850 C 编译包	本编译器将用 C 语言写的程序转换为微控制器执行的目标代码。 本编译器由项目管理器 PM+ 内启动。 型号: μ SxxxxCA703000
DF703771 设备文件	本文件包含器件的特有信息。 该设备文件应和各开发工具(CA850 或 ID850QB)配合使用。 相应的主机和操作系统随使用的工具不同而异。

备注: xxxx 型号随使用的主机和操作系统的不同而异。

μ SxxxxCA703000

xxxx	主机	OS	提供媒介
AB17	PC-9800 系列, IBM PC/AT 兼容型	Windows (日语版)	CD-ROM
BB17		Windows (英语版)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4SunOS™ (Rel. 2.5.1)	

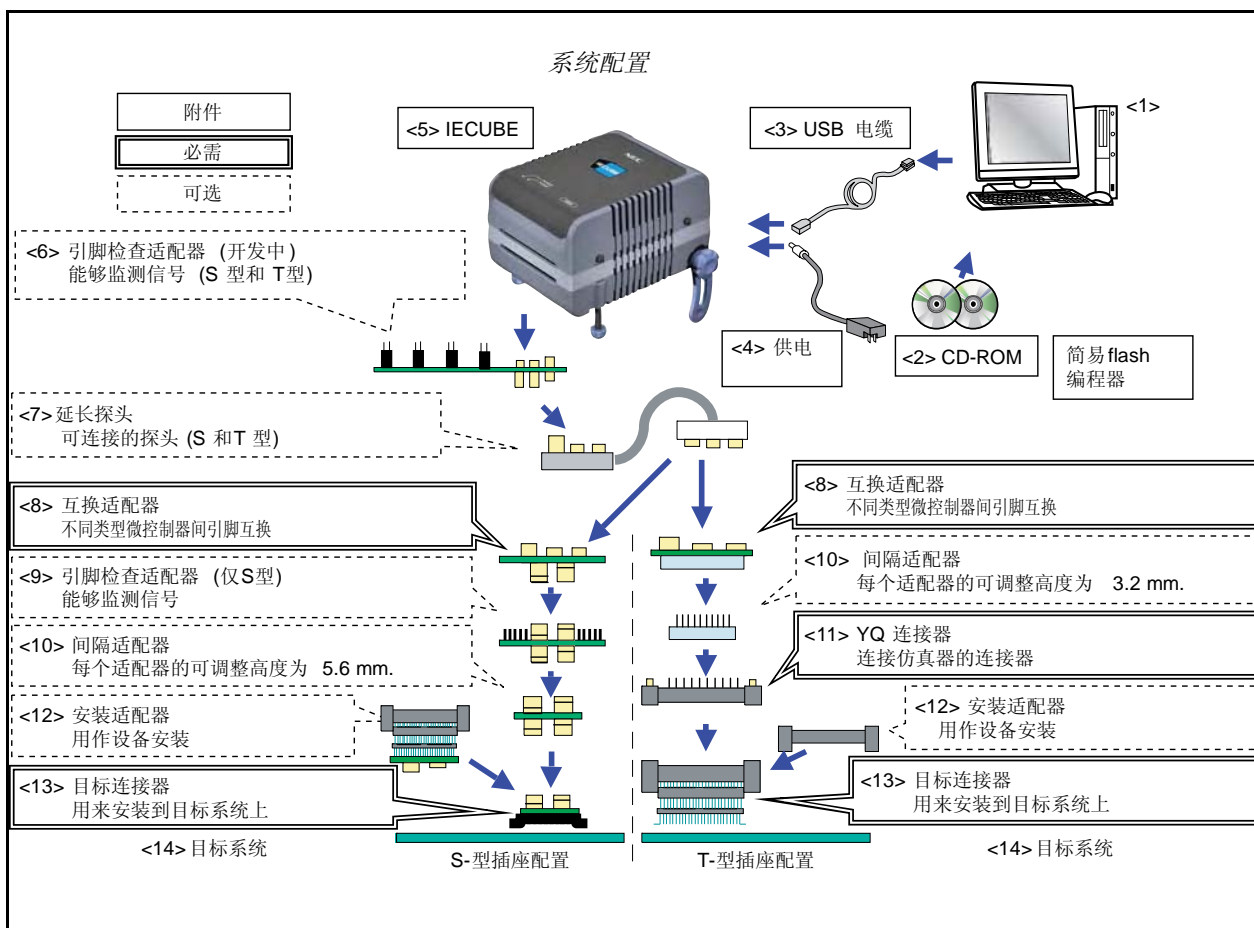
A.3 控制软件

PM+ 项目管理器	这是一个控制软件, 设计使得用户可以在 Windows 环境下有效地开发用户程序。 所有用户开发过程中的操作, 诸如启动编辑器、构建以及进行调试等, 都可以在 PM+ 中进行。 <注意事项> PM+ 包含在 C 编译包 CA850 之中。 它仅能在 Windows 下使用。
--------------	---

A.4 调试工具(硬件)

A.4.1 使用IECUBE QB-V850ESJX3H[※]

当将 QB-V850ESJX3H[※]连接至主机(PC-9821 系列, PC/AT 兼容机)时的系统配置如下所示。即使不备有可选部件,照样可以完成连接。

图 A-2. 系统配置 (使用 QB-V850ESJX3H[※]时) (1/2)

注: 开发中

图 A-2. 系统配置 (使用 QB-V850ESJX3H^{#1}) (2/2)

- <1> 主机(PC-9821 系列, IBM-PC/AT 兼容机)
- <2> 调试器、USB 驱动器、手册等等。(ID850QB 光碟, 附属光碟^{#2})
- <3> USB 接口电缆
- <4> AC 适配器
- <5> 片上仿真器 (QB-V850ESJX3H^{#1})
- <6> 引脚检查适配器(S 和 T 型) (QB-144-CA-01^{#3}) (可选)
- <7> 延长探头 (flexible 型)(S 和 T 型) (QB-144-EP-01S) (可选)
- <8> 延长探头(coaxial 型号)(S 和 T 型) (QB-144-EP-01S) (可选)
- <9> 互换适配器^{#4} (S 型: QB-100GC-EA-04S (GC 包装), QB-128GF-EA-01S (GF 包装), T 型: QB-100GC-EA-05T (GC 包装), QB-128GF-EA-02T (GF 包装))
- <10> 引脚检查适配器^{#5} (仅 S 型: QB-100-CA-01S (GC 包装), QB-128GF-CA-01S (GF 包装)) (可选的)
- <11> 间隔适配器^{#5} (S 型: QB-100-CA-01S (GC 包装), QB-128GF-CA-01S (GF 包装)) (可选的) QB-100GC-YS-01T (GC 包装), QB-128GF-YS-01T (GF 包装) (可选)
- <12> YQ 连接器^{#4} (仅 T 型) (QB-100GC-YQ-01T) (GC 包装), QB-128GF-YQ-01T (GF 包装)
- <13> 焊装适配器 (S 型: QB-100GC-MA-01S (GC 包装), QB-128GF-MA-01S (GF 包装), T 型: QB-100GC-HQ-01T (GC 包装), QB-128GF-HQ-01T (GF 包装)) (可选的)
- <14> 目标连接器^{#4} (S 型: QB-100GC-TC-01S (GC 包装), QB-128GF-TC-01S (GF 包装), T 型: QB-100GC-NQ-01T (GC 包装), QB-128GF-NQ-01T (GF 包装))
- <15> 目标系统

- 注
1. 开发中
 2. 从 NEC 电子的网站上下载设备文件。
<http://www.necel.com/micro/ods/eng/>
 3. 开发中
 4. 根据订货码决定设备供应。
 - 当订购 QB-V850ESJX3H-ZZZ 时
不提供互换适配器和目标连接器。
 - 当订购 QB-V850ESJX3H-S100GC 时
提供 QB-80GC-EA-04S 和 QB-80GC-TC-01S
 - 当订购 QB-V850ESJX3H-S128GF 时
提供 QB-128GF-EA-01S 和 QB-128GF-TC-01S
 - 当订购 QB-V850ESJX3H-T100GC 时
提供 QB-100GC-EA-05T, QB-100GC-YQ-01T 和 QB-100GC-NQ-01T
 - 当订购 QB-V850ESJX3H-T128GF 时
提供 QB-1028GF-EA-02T, QB-128GF-YQ-01T 和 QB-128GF-NQ-01T
 5. 当使用 <9> 和 <10>二者时, 不必考虑 <9> 和<10> 之间次序。

<5> QB-V850ESJX3H [‡] 片上仿真器	当开发使用 V850ES/JF3-L 的应用系统时, 使用该片上调试仿真器调试其硬件和软件。它支持集成调试器 ID850QB。该仿真器应与一个供电单元和仿真探头配合使用。使用 USB 接口电缆将该仿真器与主机连接。
<3> USB 接口电缆	连接主机和 QB-V850ESJX3H 的电缆。
<4> AC 适配器	替代插头, 可 100 至 240 V 供电。
<9> QB-100GC-EA-04S QB-128GF-EA-01S QB-100GC-EA-05T QB-128GF-EA-02T 互换适配器	进行引脚转换的适配器。 <ul style="list-style-type: none"> • QB-100GC-EA-04S: 100 引脚塑封 LQFP (GC-UEU 型) • QB-128GF-EA-01S: 128 引脚塑封 LQFP (GF-GAT 型) • QB-100GC-EA-05T: 100 引脚塑封 LQFP (GC-UEU 型) • QB-128GF-EA-02T: 128 引脚塑封 LQFP (GF-GAT 型)
<10> QB-100-CA-01S QB-128-CA-01S (仅 S 型) 引脚检查适配器	使用示波器检测波形等用途的适配器。 <ul style="list-style-type: none"> • QB-100-CA-01S: 100 引脚塑封 LQFP (GC-UEU 型) • QB-128-CA-01S: 128 引脚塑封 LQFP (GF-GAT 型)
<11> QB-100-SA-01S QB-144-SA-01S QB-100GC-YS-01T QB-128GF-YS-01T 间隔适配器	调整高度的适配器。 <ul style="list-style-type: none"> • QB-100-SA-01S: 100 引脚塑封 LQFP (GC-UEU 型) • QB-144-SA-01S: 128 引脚塑封 LQFP (GF-GAT 型) • QB-100GC-YS-01T: 100 引脚塑封 LQFP (GC-UEU 型) • QB-128GF-YS-01T: 128 引脚塑封 LQFP (GF-GAT 型)
<12> QB-100GC-YQ-01T QB-128GF-YQ-01T (T type only) YQ 连接器	连接目标连接器和互换适配器的转换适配器 <ul style="list-style-type: none"> • QB-100GC-YQ-01T: 100 引脚塑封 LQFP (GC-UEU 型) • QB-128GF-YQ-01T: 128 引脚塑封 LQFP (GF-GAT 型)
<13> QB-100GC-MA-01S QB-128GF-MA-01S QB-100GC-HQ-01T QB-128GF-HQ-01T 焊装适配器	使用插座安装 V850ES/JG3-U 或 V850ES/JH3-U 的适配器。 <ul style="list-style-type: none"> • QB-100GC-MA-01S: 100 引脚塑封 LQFP (GC-UEU 型) • QB-128GF-MA-01S: 128 引脚塑封 LQFP (GF-GAT 型) • QB-100GC-HQ-01T: 100 引脚塑封 LQFP (GC-UEU 型) • QB-128GF-HQ-01T: 128 引脚塑封 LQFP (GF-GAT 型)
<14> QB-100GC-TC-01S QB-128GF-TC-01S QB-100GC-NQ-01T QB-128GF-NQ-01T 目标连接器	焊接到目标系统上的连接器 <ul style="list-style-type: none"> • QB-100GC-TC-01S: 100 引脚塑封 LQFP (GC-UEU 型) • QB-128GF-TC-01S: 128 引脚塑封 LQFP (GF-GAT 型) • QB-100GC-NQ-01T: 100 引脚塑封 LQFP (GC-UEU 型) • QB-128GF-NQ-01T: 128 引脚塑封 LQFP (GF-GAT 型)

注: QB-V850ESSX2 提供一个供电单元、USB 接口电缆和 flash 编程器(MINICUBE2)。它还提供集成调试器 ID850QB 作为控制软件。

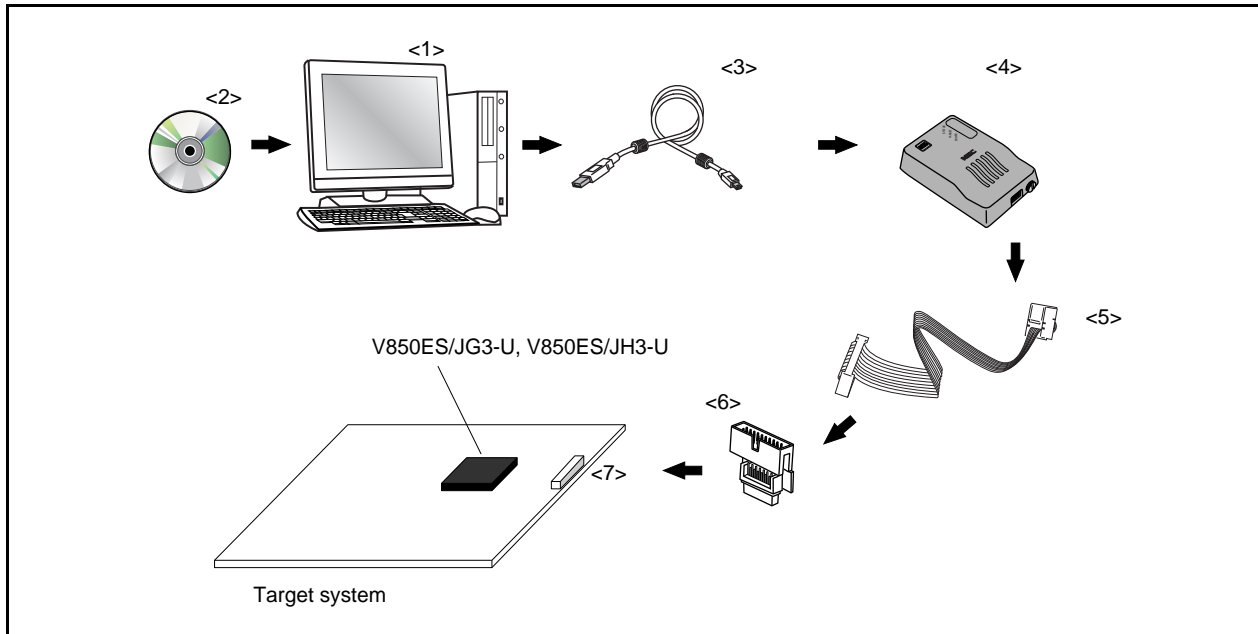
备注: 尖括号中的数字对应于图 A-2 中的数字。

A.4.2 使用MINICUBE QB-V850MINI

(1) 使用 MINICUBE 在线仿真

将 MINICUBE 连接到主机(PC-9821 系列, PC/AT 兼容机)时的系统配置如下所示

图 A-3. 在线仿真系统配置



<1> 主机	配有 USB 接口的 PC
<2> CD-ROM ^{Note 1}	CD-ROM 内有诸如集成调试器 ID850QB, N-Wire 检查器, 设备驱动和文档。它和 MINICUBE 一并提供。
<3> USB 接口电缆	连接主机和 MINICUBE 的 USB 电缆。它和 MINICUBE 一并提供。电缆长度大约为 2m。
<4> MINICUBE 片上调试仿真器	当开发使用 V850ES/JG3-U or V850ES/JH3-U 的应用系统时, 使用该在线调试仿真器调试其硬件和软件, 它支持集成调试器 ID850QB。
<5> OCD 线	连接 MINICUBE 和目标系统的电缆。它和 MINICUBE 一并提供。电缆长度大约为 20 cm。
<6> 连接器转换板 KEL 适配器	该转换板与 MINICUBE 一并提供。
<7> MINICUBE 连接器 KEL 连接器 ^{Note 2}	8830E-026-170S (与 MINICUBE 一并提供) 8830E-026-170L ((单独售卖)

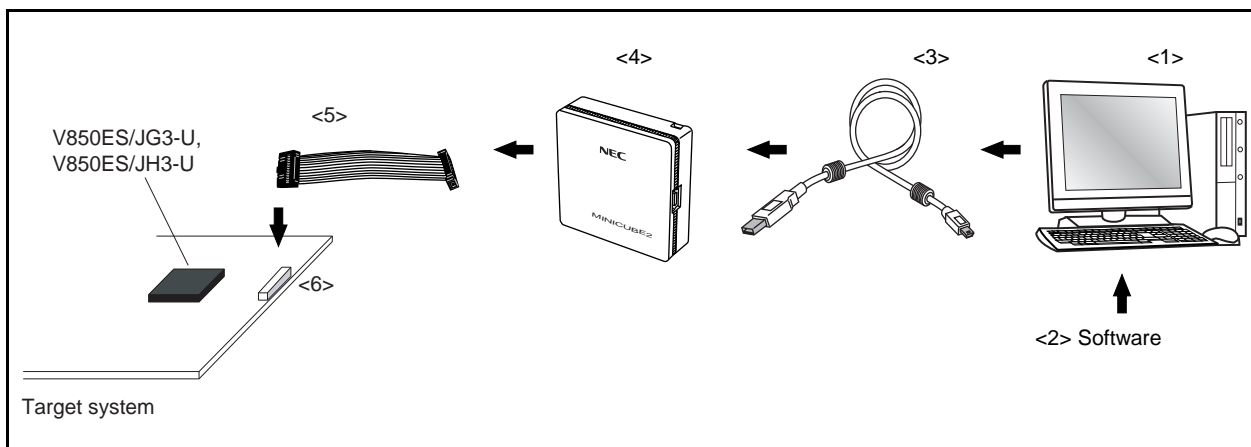
- 注
1. 从 NEC 电子的网站下载设备文件。
<http://www.necel.com/micro/ods/eng/>
 2. KEL 公司的产品

备注: 尖括号中的数字对应于图 A-3 中的数字。

A.4.3 使用MINICUBE2 QB-MINI2

将 MINICUBE2 连接至主机(PC-9821 系列, PC/AT 兼容机)时的系统配置如下所示

图 A-4. 在线仿真系统配置



<1> 主机	配有 USB 接口的 PC
<2> 软件	集成调试器 ID850QB, 设备文件等。从 NEC 电子网站上下载设备文件。 http://www.necel.com/micro/ods/eng/
<3> USB 接口电缆	连接主机和 MINICUBE 的 USB 电缆。它和 MINICUBE 一并提供。电缆长度大约为 2 m。
<4> MINICUBE2 片上调试仿真器	当开发使用 V850ES/JG3-U or V850ES/JH3-U 的应用系统时, 使用该在线调试仿真器调试其硬件和软件, 它支持集成调试器 ID850QB。
<5> 16 引脚目标电缆	连接 MINICUBE2 和目标系统的电缆 它和 MINICUBE 一并提供。电缆长度大约为 15 cm。
<6> 目标连接器 (单独售卖)	使用一个 2.54 mm 间距的 16 引脚通用连接器。

备注: 尖括号中的数字对应于图 A-4 中的数字。

A.5 调试工具(软件)

ID850QB 综合调试器	本调试器支持 V850 微控制器的在线仿真。ID850QB 为基于 Windows 操作系统的软件。 它具有改进的 C-兼容调试功能，并且，使用一个集成窗口显示源程序追踪执行结果，以及相关追踪结果的源文件、反汇编显示、存储显示 它应与设备文件配合使用。 型号: μ Sxxxx ID703000-QB (ID850QB)
SM+ for V850ES/Jx2 (开发中) 系统仿真器	本调试器支持 V850 微控制器的在线仿真。SM+ for V850ES/Jx2 为基于 Windows 操作系统的软件。 当在主机上仿真目标系统时，用于执行在 C 编译器或汇编器的操作。 SM+ for V850ES/Jx2 允许从硬件开发中在非独立基础上执行应用逻辑测试和能力测试,因此，提供更高的开发效率和软件质量。 SM+ 应与设备文件配合使用（单独销售）。 型号: μ SxxxxSM703724-B

备注: xxxx 型号随使用的主机和操作系统的不同而异。

μ SxxxxID703000-QB

μ SxxxxSM703724-B

xxxx	主机	OS	提供媒介
AB17	PC-9800 系列,	Windows (日语版)	CD-ROM
BB17	IBM PC/AT 兼容型	Windows (英语版)	

A.6 嵌入式软件

RX850, RX850 Pro 实时操作系统	RX850 和 RX850 Pro 为实时操作系统, 遵从 μ TRON 3.0 规范。 提供一个工具 (配置器), 用来产生多重信息表。 RX850 Pro 比 RX850 具有更多功能。
	型号: μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$ (RX850) μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$ (RX850 Pro)
Applilet [®] (开发中)	该驱动配置器可以自动为 V850ES/JG3-U 和 V850ES/JH3-U 单片机生成例子程序。
RX-FS850 (文件系统)	这是一个 FAT 文件系统功能。 该文件系统支持 CD-ROM 文件系统功能。 该文件系统被实时操作系统 RX850 Pro 使用。

注意事项: 欲购买 RX850 或 RX850 Pro 实时操作系统, 首先填写购买申请表并签署授权许可文件。

备注: xxxx 和 $\Delta\Delta\Delta\Delta$ 型号随使用的主机和操作系统的不同而异。

μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$

μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$

$\Delta\Delta\Delta\Delta$	产品概要	用于量产的最大数目
001	评估产品	不要用于量产
100K	量产产品	0.1 百万单位
001M		1 百万单位
010M		10 百万单位
S01	源程序	用于量产的源程序

xxxx	主机	OS	提供媒介
AB17	PC-9800 系列, IBM PC/AT 兼容型	Windows (日语版)	CD-ROM
BB17		Windows (英语版)	
3K17	SPARCstation	SunOS™ (Rel. 2.5.1)	

A.7 Flash 存储器烧写工具

Flashpro IV (型号: PG-FP4) Flashpro V (型号: PG-FP5) Flash 存储器编程器	Flash 编程器专用于具备片上 Flash 存储器的微控制器编程。
QB-MINI2 (MINICUBE2)	具有编程功能的片上调试仿真器。
FA-100GC-UEU-B FA-128GF-GAT-B Flash 存储器烧写适配器	Flash 存储器烧写适配器, 用来连接 Flashpro IV、Flashpro V 等。(未连接)FA-100GC-UEU-B: 100 引脚塑封 LQFP (GC-UEU 型) • FA-128GF-GAT-B: 100 引脚塑封 LQFP (GF-GAT 型)

备注: FA-80GC-GAD-B、FA-80GK-GAK-B、FA-70F3736GC-GAD-RX 和 FA-70F3736GK-GAK-RX 为 Naito Densai Machida Mfg. Co., Ltd 的产品。

TEL: +81-42-750-4172

附录B V850ES/Jx3-U 和 V850ES/Jx3 的主要区别

表 B-1. V850ES/Jx3-U 和 V850ES/Jx3 的主要区别

主要区别	V850ES/Jx3-U	V850ES/Jx3
最短指令执行时间	20.8 ns (48 MHz 操作)	31.25 ns (32 MHz 操作)
16 位定时器	TAA (TMP 的高性能型)	TMP
	TAB (TMQ 的高性能型)	TMQ
	TMT (编码器定时器)	无
监控定时器功能	RTC (含硬件计数器)	WT (不含硬件计数器)
电机控制功能	有	无
USB 接口	功能	无
异步串行接口	UARTC (UARTA 的高性能型)	UARTA
封装	100-pin LQFP	100-pin LQFP
	128-pin LQFP	144-pin LQFP

附录C 寄存器索引

(1/14)

符号	名称	单元	页码
ADA0CR0	A/D 转换结果寄存器 0	ADC	693
ADA0CR0H	A/D 转换结果寄存器 0H	ADC	693
ADA0CR1	A/D 转换结果寄存器 1	ADC	693
ADA0CR1H	A/D 转换结果寄存器 1H	ADC	693
ADA0CR2	A/D 转换结果寄存器 2	ADC	693
ADA0CR2H	A/D 转换结果寄存器 2H	ADC	693
ADA0CR3	A/D 转换结果寄存器 3	ADC	693
ADA0CR3H	A/D 转换结果寄存器 3H	ADC	693
ADA0CR4	A/D 转换结果寄存器 4	ADC	693
ADA0CR4H	A/D 转换结果寄存器 4H	ADC	693
ADA0CR5	A/D 转换结果寄存器 5	ADC	693
ADA0CR5H	A/D 转换结果寄存器 5H	ADC	693
ADA0CR6	A/D 转换结果寄存器 6	ADC	693
ADA0CR6H	A/D 转换结果寄存器 6H	ADC	693
ADA0CR7	A/D 转换结果寄存器 7	ADC	693
ADA0CR7H	A/D 转换结果寄存器 7H	ADC	693
ADA0CR8	A/D 转换结果寄存器 8	ADC	693
ADA0CR8H	A/D 转换结果寄存器 8H	ADC	693
ADA0CR9	A/D 转换结果寄存器 9	ADC	693
ADA0CR9H	A/D 转换结果寄存器 9H	ADC	693
ADA0CR10	A/D 转换结果寄存器 10	ADC	693
ADA0CR10H	A/D 转换结果寄存器 10H	ADC	693
ADA0CR11	A/D 转换结果寄存器 11	ADC	693
ADA0CR11H	A/D 转换结果寄存器 11H	ADC	693
ADA0M0	A/D 转换模式寄存器 0	ADC	686
ADA0M1	A/D 转换模式寄存器 1	ADC	688
ADA0M2	A/D 转换模式寄存器 2	ADC	691
ADA0PFM	掉电比较模式寄存器	ADC	695
ADA0PFT	掉电比较门限值寄存器	ADC	696
ADA0S	模拟输入通道指定寄存器	ADC	692
ADIC	中断控制寄存器	INTC	1195
AWC	地址等待控制寄存器	BCU	191
BCC	总线周期控制寄存器	BCU	192
BSC	总线宽度配置寄存器	BCU	180
CCLS	CPU操作时钟状态寄存器	CG	206
CF0CTL0	CSIF0 控制寄存器 0	CSIF	772
CF0CTL1	CSIF0 控制寄存器 1	CSIF	775
CF0CTL2	CSIF0 控制寄存器 2	CSIF	776
CF0RIC	中断控制寄存器	INTC	1195

符号	名称	单元	页码
CF0RX	CSIF0 接收数据寄存器	CSIF	770
CF0RXL	CSIF0 接收数据寄存器 L	CSIF	770
CF0STR	CSIF0 状态寄存器	CSIF	778
CF0TIC	中断控制寄存器	INTC	1195
CF0TX	CSIF0 发送数据寄存器	CSIF	771
CF0TXL	CSIF0 发送数据寄存器 L	CSIF	771
CF1CTL0	CSIF1 控制寄存器 0	CSIF	772
CF1CTL1	CSIF1 控制寄存器 1	CSIF	775
CF1CTL2	CSIF1 控制寄存器 2	CSIF	776
CF1RIC	中断控制寄存器	INTC	1195
CF1RX	CSIF1 接收数据寄存器	CSIF	770
CF1RXL	CSIF1 接收数据寄存器 L	CSIF	770
CF1STR	CSIF1 状态寄存器	CSIF	778
CF1TIC	中断控制寄存器	INTC	1195
CF1TX	CSIF1 发送数据寄存器	CSIF	771
CF1TXL	CSIF1 发送数据寄存器 L	CSIF	771
CF2CTL0	CSIF2 控制寄存器 0	CSIF	772
CF2CTL1	CSIF2 控制寄存器 1	CSIF	775
CF2CTL2	CSIF2 控制寄存器 2	CSIF	776
CF2RIC	中断控制寄存器	INTC	1195
CF2RX	CSIF2 接收数据寄存器	CSIF	770
CF2RXL	CSIF2 接收数据寄存器 L	CSIF	770
CF2STR	CSIF2 状态寄存器	CSIF	778
CF2TIC	中断控制寄存器	INTC	1195
CF2TX	CSIF2 发送数据寄存器	CSIF	771
CF2TXL	CSIF2 发送数据寄存器 L	CSIF	771
CF3CTL0	CSIF3 控制寄存器 0	CSIF	772
CF3CTL1	CSIF3 控制寄存器 1	CSIF	775
CF3CTL2	CSIF3 控制寄存器 2	CSIF	776
CF3RIC	中断控制寄存器	INTC	1195
CF3RX	CSIF3 接收数据寄存器	CSIF	770
CF3RXL	CSIF3 接收数据寄存器 L	CSIF	770
CF3STR	CSIF3 状态寄存器	CSIF	778
CF3TIC	中断控制寄存器	INTC	1195
CF3TX	CSIF3 发送数据寄存器	CSIF	771
CF3TXL	CSIF3 发送数据寄存器 L	CSIF	771
CF4CTL0	CSIF4 控制寄存器 0	CSIF	772
CF4CTL1	CSIF4 控制寄存器 1	CSIF	775
CF4CTL2	CSIF4 控制寄存器 2	CSIF	776
CF4RIC	中断控制寄存器	INTC	1195
CF4RX	CSIF4 接收数据寄存器	CSIF	770
CF4RXL	CSIF4 接收数据寄存器 L	CSIF	770
CF4STR	CSIF4 状态寄存器	CSIF	778
CF4TIC	中断控制寄存器	INTC	1195

符号	名称	单元	页码
CF4TX	CSIF4 发送数据寄存器	CSIF	771
CF4TXL	CSIF4 发送数据寄存器 L	CSIF	771
CKC	Clock 控制寄存器	CG	209
CLM	时钟监视模式寄存器	CLM	1254
CRCD	CRC 数据寄存器	CRC	1265
CRCIN	CRC 输入寄存器	CRC	1265
CTBP	CALLT基指针	CPU	64
CTPC	CALLT 执行状态保存寄存器	CPU	63
CTPSW	CALLT 执行状态保存寄存器	CPU	63
DA0CS0	D/A 转换值设定寄存器 0	DAC	720
DA0CS1	D/A 转换值设定寄存器 1	DAC	720
DA0M	D/A 转换模式寄存器	DAC	719
DADC0	DMA 地址控制寄存器 0	DMAC	1152
DADC1	DMA 地址控制寄存器 1	DMAC	1152
DADC2	DMA 地址控制寄存器 2	DMAC	1152
DADC3	DMA 地址控制寄存器 3	DMAC	1152
DBC0	DMA 字节寄存器 0	DMAC	1151
DBC1	DMA 字节寄存器 1	DMAC	1151
DBC2	DMA 字节寄存器 2	DMAC	1151
DBC3	DMA 字节寄存器 3	DMAC	1151
DBPC	异常/调试陷阱状态保存寄存器	CPU	64
DBPSW	异常/调试陷阱状态保存寄存器	CPU	64
DCHC0	DMA 通道控制寄存器 0	DMAC	1153
DCHC1	DMA 通道控制寄存器 1	DMAC	1153
DCHC2	DMA 通道控制寄存器 2	DMAC	1153
DCHC3	DMA 通道控制寄存器 3	DMAC	1153
DDA0H	DMA 目标地址寄存器 0H	DMAC	1150
DDA0L	DMA 目标地址寄存器 0L	DMAC	1150
DDA1H	DMA 目标地址寄存器 1H	DMAC	1150
DDA1L	DMA 目标地址寄存器 1L	DMAC	1150
DDA2H	DMA 目标地址寄存器 2H	DMAC	1150
DDA2L	DMA 目标地址寄存器 2L	DMAC	1150
DDA3H	DMA 目标地址寄存器 3H	DMAC	1150
DDA3L	DMA 目标地址寄存器 3L	DMAC	1150
DMAIC0	中断控制寄存器	INTC	1195
DMAIC1	中断控制寄存器	INTC	1195
DMAIC2	中断控制寄存器	INTC	1195
DMAIC3	中断控制寄存器	INTC	1195
DSA0H	DMA 源地址寄存器 0H	DMAC	1149
DSA0L	DMA 源地址寄存器 0L	DMAC	1149
DSA1H	DMA 源地址寄存器 1H	DMAC	1149
DSA1L	DMA 源地址寄存器 1L	DMAC	1149
DSA2H	DMA 源地址寄存器 2H	DMAC	1149
DSA2L	DMA 源地址寄存器 2L	DMAC	1149

符号	名称	单元	页码
DSA3H	DMA 源地址寄存器 3H	DMAC	1149
DSA3L	DMA 源地址寄存器 3L	DMAC	1149
DTFR0	DMA 触发因子寄存器 0	DMAC	1154
DTFR1	DMA 触发因子寄存器 1	DMAC	1154
DTFR2	DMA 触发因子寄存器 2	DMAC	1154
DTFR3	DMA 触发因子寄存器 3	DMAC	1154
DWC0	数据等待控制寄存器 0	BCU	188
ECR	中断源寄存器	CPU	61
EIPC	中断状态保存寄存器	CPU	60
EIPSW	中断状态保存寄存器	CPU	60
EXDRQEN	外部 DMA 请求使能寄存器	DMA	1157
FEPC	NMI 状态保存寄存器	CPU	61
FEPSW	NMI 状态保存寄存器	CPU	61
HZA0CTL0	高阻输出控制寄存器 0	Motor	589
HZA0CTL1	高阻输出控制寄存器 1	Motor	589
IIC0	IIC 移位寄存器 0	I ² C	838
IIC1	IIC 移位寄存器 1	I ² C	838
IIC2	IIC 移位寄存器 2	I ² C	838
IICC0	IIC 控制寄存器 0	I ² C	826
IICC1	IIC 控制寄存器 1	I ² C	826
IICC2	IIC 控制寄存器 2	I ² C	826
IICCL0	IIC 时钟选择寄存器 0	I ² C	835
IICCL1	IIC 时钟选择寄存器 1	I ² C	835
IICCL2	IIC 时钟选择寄存器 2	I ² C	835
IICF0	IIC 标志寄存器 0	I ² C	834
IICF1	IIC 标志寄存器 1	I ² C	834
IICF2	IIC 标志寄存器 2	I ² C	834
IICIC0	中断控制寄存器	INTC	1195
IICIC1	中断控制寄存器	INTC	1195
IICIC2	中断控制寄存器	INTC	1195
IICS0	IIC 状态寄存器 0	I ² C	830
IICS1	IIC 状态寄存器 1	I ² C	830
IICS2	IIC 状态寄存器 2	I ² C	830
IICX0	IIC 函数扩展寄存器 0	I ² C	836
IICX1	IIC 函数扩展寄存器 1	I ² C	836
IICX2	IIC 函数扩展寄存器 2	I ² C	836
IMR0	中断屏蔽寄存器 0	INTC	1199
IMR0H	中断屏蔽寄存器 0H	INTC	1199
IMR0L	中断屏蔽寄存器 0L	INTC	1199
IMR1	中断屏蔽寄存器 1	INTC	1199
IMR1H	中断屏蔽寄存器 1H	INTC	1199
IMR1L	中断屏蔽寄存器 1L	INTC	1199
IMR2	中断屏蔽寄存器 2	INTC	1199
IMR2H	中断屏蔽寄存器 2H	INTC	1199

符号	名称	单元	页码
IMR2L	中断屏蔽寄存器 2L	INTC	1199
IMR3	中断屏蔽寄存器 3	INTC	1199
IMR3H	中断屏蔽寄存器 3H	INTC	1199
IMR3L	中断屏蔽寄存器 3L	INTC	1199
IMR4	中断屏蔽寄存器 4	INTC	1199
IMR4H	中断屏蔽寄存器 4H	INTC	1199
IMR4L	中断屏蔽寄存器 4L	INTC	1199
IMR5	中断屏蔽寄存器 5	INTC	1199
IMR5H	中断屏蔽寄存器 5H	INTC	1199
IMR5L	中断屏蔽寄存器 5L	INTC	1199
INTF0	外部下降沿指定寄存器 0	INTC	1211
INTF2	外部下降沿指定寄存器 2	INTC	1212
INTF3	外部下降沿指定寄存器 3	INTC	1213
INTF4	外部下降沿指定寄存器 4	INTC	1214
INTF5	外部下降沿指定寄存器 5	INTC	1215
INTF9	外部下降沿指定寄存器 9	INTC	1216
INTF9H	外部下降沿指定寄存器 9H	INTC	1216
INTF9L	外部下降沿指定寄存器 9L	INTC	1216
INTNFC	噪音消除 控制寄存器	INTC	1217
INTR0	外部上升沿指定寄存器 0	INTC	1211
INTR2	外部上升沿指定寄存器 2	INTC	1212
INTR3	外部上升沿指定寄存器 3	INTC	1213
INTR4	外部上升沿指定寄存器 4	INTC	1214
INTR5	外部上升沿指定寄存器 5	INTC	1215
INTR9	外部上升沿指定寄存器 9	INTC	1216
INTR9H	外部上升沿指定寄存器 9H	INTC	1216
INTR9L	外部上升沿指定寄存器 9L	INTC	1216
ISPR	服务中优先寄存器	INTC	1201
KRIC	中断控制寄存器	INTC	1195
KRM	按键返回模式寄存器	KR	1221
LOCKR	锁定寄存器	CG	210
LVIIC	中断控制寄存器	INTC	1195
LVIM	低压检测寄存器	LVI	1259
OCDM	在线调试模式寄存器	DCU	1303
OCKS0	IIC 时钟分频选择寄存器 0	I ² C	838
OCKS1	IIC 时钟分频选择寄存器 1	I ² C	838
OSTS	振荡器稳定时间选择寄存器	Standby	1226
P0	端口 0 寄存器	端口	107
P1	端口 1 寄存器	端口	112
P2	端口 2 寄存器	端口	113
P3	端口 3 寄存器	端口	117
P4	端口 4 寄存器	端口	122
P5	端口 5 寄存器	端口	126
P6	端口 6 寄存器	端口	132

符号	名称	单元	页码
P7H	端口 7 寄存器 H	端口	136
P7L	端口 7 寄存器 L	端口	136
P9	端口 9 寄存器	端口	139
P9H	端口 9 寄存器 H	端口	139
P9L	端口 9 寄存器 L	端口	139
PC	程序计数器	CPU	58
PCC	执行时钟控制寄存器	CG	202
PCM	端口 CM 寄存器	端口	148
PCS	端口 CS 寄存器	端口	151
PCT	端口 CT 寄存器	端口	153
PDH	端口 DH 寄存器	端口	157
PDL	端口 DL 寄存器	端口	159
PDLH	端口 DL 寄存器 H	端口	159
PDLL	端口 DL 寄存器 L	端口	159
PF0	端口 0 函数寄存器	端口	111
PF2	端口 2 函数寄存器	端口	116
PF3	端口 3 函数寄存器	端口	121
PF4	端口 4 函数寄存器	端口	124
PF5	端口 5 函数寄存器	端口	131
PF9	端口 9 函数寄存器	端口	147
PF9L	端口 9 函数寄存器 L	端口	147
PFC0	端口 0 函数控制寄存器	端口	110
PFC2	端口 2 函数控制寄存器	端口	114
PFC3	端口 3 函数控制寄存器	端口	119
PFC4	端口 4 函数控制寄存器	端口	123
PFC5	端口 5 函数控制寄存器	端口	129
PFC6	端口 6 函数控制寄存器	端口	134
PFC9	端口 9 函数控制寄存器	端口	142
PFC9H	端口 9 函数控制寄存器 H	端口	142
PFC9L	端口 9 函数控制寄存器 L	端口	142
PFCE0	端口 0 函数控制扩展寄存器	端口	111
PFCE2	端口 2 函数控制扩展寄存器	端口	115
PFCE3	端口 3 函数控制扩展寄存器	端口	119
PFCE4	端口 4 函数控制扩展寄存器	端口	123
PFCE5	端口 5 函数控制扩展寄存器	端口	129
PFCE6	端口 6 函数控制扩展寄存器	端口	134
PFCE9	端口 9 函数控制扩展寄存器	端口	143
PFCE9H	端口 9 函数控制扩展寄存器 H	端口	143
PFCE9L	端口 9 函数控制扩展寄存器 L	端口	143
PIC00	中断控制寄存器	INTC	1195
PIC01	中断控制寄存器	INTC	1195
PIC02	中断控制寄存器	INTC	1195
PIC03	中断控制寄存器	INTC	1195
PIC04	中断控制寄存器	INTC	1195

符号	名称	单元	页码
PIC05	中断控制寄存器	INTC	1195
PIC06	中断控制寄存器	INTC	1195
PIC07	中断控制寄存器	INTC	1195
PIC08	中断控制寄存器	INTC	1195
PIC09	中断控制寄存器	INTC	1195
PIC10	中断控制寄存器	INTC	1195
PIC11	中断控制寄存器	INTC	1195
PIC12	中断控制寄存器	INTC	1195
PIC13	中断控制寄存器	INTC	1195
PIC14	中断控制寄存器	INTC	1195
PIC15	中断控制寄存器	INTC	1195
PIC16	中断控制寄存器	INTC	1195
PIC17	中断控制寄存器	INTC	1195
PIC18	中断控制寄存器	INTC	1195
PLLCTL	PLL 控制寄存器	CG	208
PLLS	PLL 锁存时间指定寄存器	CG	211
PM0	端口 0 模式寄存器	端口	108
PM1	端口 1 模式寄存器	端口	112
PM2	端口 2 模式寄存器	端口	113
PM3	端口 3 模式寄存器	端口	117
PM4	端口 4 模式寄存器	端口	122
PM5	端口 5 模式寄存器	端口	127
PM6	端口 6 模式寄存器	端口	133
PM7H	端口 7 模式寄存器 H	端口	137
PM7L	端口 7 模式寄存器 L	端口	137
PM9	端口 9 模式寄存器	端口	139
PM9L	端口 9 模式寄存器 L	端口	139
PMC0	端口 0 模式控制寄存器	端口	109
PMC2	端口 2 模式控制寄存器	端口	114
PMC3	端口 3 模式控制寄存器	端口	118
PMC4	端口 4 模式控制寄存器	端口	123
PMC5	端口 5 模式控制寄存器	端口	128
PMC6	端口 6 模式控制寄存器	端口	133
PMC9	端口 9 模式控制寄存器	端口	140
PMC9H	端口 9 模式控制寄存器 H	端口	140
PMC9L	端口 9 模式控制寄存器 L	端口	140
PMCCM	端口 CM 模式控制寄存器	端口	150
PMCCS	端口 CS 模式控制寄存器	端口	152
PMCCT	端口 CT 模式控制寄存器	端口	155
PMCDH	端口 DH 模式控制寄存器	端口	157
PMCDL	端口 DL 模式控制寄存器	端口	160
PMCDLH	端口 DL 模式控制寄存器 H	端口	160
PMCDLL	端口 DL 模式控制寄存器 L	端口	160

符号	名称	单元	页码
PMCM	端口 CM 模式寄存器	端口	149
PMCS	端口 CS 模式寄存器	端口	151
PMCT	端口 CT 模式寄存器	端口	154
PMDH	端口 DH 模式寄存器	端口	157
PMDL	端口 DL 模式寄存器	端口	159
PMDLH	端口 DL 模式寄存器 H	端口	159
PMDLL	端口 DL 模式寄存器 L	端口	159
PRCMD	命令寄存器	CPU	93
PRSCM0	预分频比较寄存器 0	BRG	657
PRSCM1	预分频比较寄存器 1	BRG	815
PRSCM2	预分频比较寄存器 2	BRG	815
PRSCM3	预分频比较寄存器 3	BRG	815
PRSM0	预分频模式寄存器 0	BRG	656
PRSM1	预分频模式寄存器 1	BRG	814
PRSM2	预分频模式寄存器 2	BRG	814
PRSM3	预分频模式寄存器 3	BRG	814
PSC	上电保存控制寄存器	CG	1224
PSMR	上电保存模式寄存器	CG	1225
PSW	程序状态字	CPU	62
r0-r31	通用寄存器	CPU	58
RAMS	内置RAM数据状态寄存器	LVI	1260
RC1ALH	警报小时设置寄存器	RTC	654
RC1ALM	警报分钟设置寄存器	RTC	654
RC1ALW	警报星期天数设置寄存器	RTC	655
RC1CC0	实时计数器控制寄存器 0	RTC	643
RC1CC1	实时计数器控制寄存器 1	RTC	644
RC1CC2	实时计数器控制寄存器 2	RTC	645
RC1CC3	实时计数器控制寄存器 3	RTC	646
RC1DAY	天计数器寄存器	RTC	650
RC1HOUR	小时计数寄存器	RTC	648
RC1MIN	分钟计数寄存器	RTC	648
RC1MONTH	月计数寄存器	RTC	652
RC1SEC	秒计数寄存器	RTC	647
RC1SUBC	副计数寄存器	RTC	647
RC1SUBU	错误检测寄存器	RTC	653
RC1WEEK	星期天数计数寄存器	RTC	651
RC1YEAR	年计数寄存器	RTC	652
RCM	内置振荡器模式寄存器	CG	206
RESF	复位源标志寄存器	Reset	1245
RTBH0	实时输出缓冲寄存器 0H	RTO	677
RTBL0	实时输出缓冲寄存器 0L	RTO	677
RTC0IC	中断控制寄存器	INTC	1195
RTC1IC	中断控制寄存器	INTC	1195
RTC2IC	中断控制寄存器	INTC	1195

符号	名称	单元	页码
RTPC0	实时输出端口控制寄存器 0	RTO	679
RTPM0	实时输出端口模式寄存器 0	RTO	678
SELCNT0	选择操作控制寄存器 0	Timer	328
SVA0	从地址寄存器 0	I ² C	839
SVA1	从地址寄存器 1	I ² C	839
SVA2	从地址寄存器 2	I ² C	839
SYS	系统状态寄存器	CPU	94
TAA0CCIC0	中断控制寄存器	INTC	1195
TAA0CCIC1	中断控制寄存器	INTC	1195
TAA0CCR0	TAA0 捕捉/比较寄存器 0	Timer	226
TAA0CCR1	TAA0 捕捉/比较寄存器 1	Timer	228
TAA0CNT	TAA0 计数器读缓冲寄存器	Timer	230
TAA0CTL0	TAA0 控制寄存器 0	Timer	217
TAA0CTL1	TAA0 控制寄存器 1	Timer	218
TAA0IOC0	TAA0 I/O 控制寄存器 0	Timer	220
TAA0IOC1	TAA0 I/O 控制寄存器 1	Timer	221
TAA0IOC2	TAA0 I/O 控制寄存器 2	Timer	222
TAA0IOC4	TAA0 I/O 控制寄存器 4	Timer	223
TAA0OPT0	TAA0 选择寄存器 0	Timer	224
TAA0OPT1	TAA0 选择寄存器 1	Timer	225
TAA0OVIC	中断控制寄存器	INTC	1195
TAA1CCIC0	中断控制寄存器	INTC	1195
TAA1CCIC1	中断控制寄存器	INTC	1195
TAA1CCR0	TAA1 捕捉/比较寄存器 0	Timer	226
TAA1CCR1	TAA1 捕捉/比较寄存器 1	Timer	228
TAA1CNT	TAA1 计数器读缓冲寄存器	Timer	230
TAA1CTL0	TAA1 控制寄存器 0	Timer	217
TAA1CTL1	TAA1 控制寄存器 1	Timer	218
TAA1IOC0	TAA1 I/O 控制寄存器 0	Timer	220
TAA1IOC1	TAA1 I/O 控制寄存器 1	Timer	221
TAA1IOC2	TAA1 I/O 控制寄存器 2	Timer	222
TAA1IOC4	TAA1 I/O 控制寄存器 4	Timer	223
TAA1OPT0	TAA1 选择寄存器 0	Timer	224
TAA1OVIC	中断控制寄存器	INTC	1195
TAA2CCIC0	中断控制寄存器	INTC	1195
TAA2CCIC1	中断控制寄存器	INTC	1195
TAA2CCR0	TAA2 捕捉/比较寄存器 0	Timer	226
TAA2CCR1	TAA2 捕捉/比较寄存器 1	Timer	228
TAA2CNT	TAA2 计数器读缓冲寄存器	Timer	230
TAA2CTL0	TAA2 控制寄存器 0	Timer	217
TAA2CTL1	TAA2 控制寄存器 1	Timer	218
TAA2IOC0	TAA2 I/O 控制寄存器 0	Timer	220
TAA2IOC1	TAA2 I/O 控制寄存器 1	Timer	221
TAA2IOC2	TAA2 I/O 控制寄存器 2	Timer	222

符号	名称	单元	页码
TAA2IOC4	TAA2 I/O 控制寄存器 4	Timer	223
TAA2OPT0	TAA2 选择寄存器 0	Timer	224
TAA2OPT1	TAA2 选择寄存器 1	Timer	225
TAA2OVIC	中断控制寄存器	INTC	1195
TAA3CCIC0	中断控制寄存器	INTC	1195
TAA3CCIC1	中断控制寄存器	INTC	1195
TAA3CCR0	TAA3 捕捉/比较寄存器 0	Timer	226
TAA3CCR1	TAA3 捕捉/比较寄存器 1	Timer	228
TAA3CNT	TAA3 计数器读缓冲寄存器	Timer	230
TAA3CTL0	TAA3 控制寄存器 0	Timer	217
TAA3CTL1	TAA3 控制寄存器 1	Timer	218
TAA3IOC0	TAA3 I/O 控制寄存器 0	Timer	220
TAA3IOC1	TAA3 I/O 控制寄存器 1	Timer	221
TAA3IOC2	TAA3 I/O 控制寄存器 2	Timer	222
TAA3IOC4	TAA3 I/O 控制寄存器 4	Timer	223
TAA3OPT0	TAA3 选择寄存器 0	Timer	224
TAA3OVIC	中断控制寄存器	INTC	1195
TAA4CCIC0	中断控制寄存器	INTC	1195
TAA4CCIC1	中断控制寄存器	INTC	1195
TAA4CCR0	TAA4 捕捉/比较寄存器 0	Timer	226
TAA4CCR1	TAA4 捕捉/比较寄存器 1	Timer	228
TAA4CNT	TAA4 计数器读缓冲寄存器	Timer	230
TAA4CTL0	TAA4 控制寄存器 0	Timer	217
TAA4CTL1	TAA4 控制寄存器 1	Timer	218
TAA4OVIC	中断控制寄存器	INTC	1195
TAA5CCIC0	中断控制寄存器	INTC	1195
TAA5CCIC1	中断控制寄存器	INTC	1195
TAA5CCR0	TAA5 捕捉/比较寄存器 0	Timer	226
TAA5CCR1	TAA5 捕捉/比较寄存器 1	Timer	228
TAA5CNT	TAA5 计数器读缓冲寄存器	Timer	230
TAA5CTL0	TAA5 控制寄存器 0	Timer	217
TAA5CTL1	TAA5 控制寄存器 1	Timer	218
TAA5IOC0	TAA5 I/O 控制寄存器 0	Timer	220
TAA5IOC1	TAA5 I/O 控制寄存器 1	Timer	221
TAA5IOC2	TAA5 I/O 控制寄存器 2	Timer	222
TAA5IOC4	TAA5 I/O 控制寄存器 4	Timer	223
TAA5OPT0	TAA5 选择寄存器 0	Timer	224
TAA5OVIC	中断控制寄存器	INTC	1195
TAB0CCIC0	中断控制寄存器	INTC	1195
TAB0CCIC1	中断控制寄存器	INTC	1195
TAB0CCIC2	中断控制寄存器	INTC	1195
TAB0CCIC3	中断控制寄存器	INTC	1195
TAB0CCR0	TAB0 捕捉/比较寄存器 0	Timer	342
TAB0CCR1	TAB0 捕捉/比较寄存器 1	Timer	344

符号	名称	单元	页码
TAB0CCR2	TAB0 捕捉/比较寄存器 2	Timer	346
TAB0CCR3	TAB0 捕捉/比较寄存器 3	Timer	348
TAB0CNT	TAB0 计数器读缓冲寄存器	Timer	350
TAB0CTL0	TAB0 控制寄存器 0	Timer	335
TAB0CTL1	TAB0 控制寄存器 1	Timer	336
TAB0IOC0	TAB0 I/O 控制寄存器 0	Timer	337
TAB0IOC1	TAB0 I/O 控制寄存器 1	Timer	338
TAB0IOC2	TAB0 I/O 控制寄存器 2	Timer	339
TAB0OIC4	TAB0 I/O 控制寄存器 4	Timer	340
TAB0OPT0	TAB0 选择寄存器 0	Timer	341
TAB0OVIC	中断控制寄存器	INTC	1195
TAB1CCIC0	中断控制寄存器	INTC	1195
TAB1CCIC1	中断控制寄存器	INTC	1195
TAB1CCIC2	中断控制寄存器	INTC	1195
TAB1CCIC3	中断控制寄存器	INTC	1195
TAB1CCR0	TAB1 捕捉/比较寄存器 0	Timer	342
TAB1CCR1	TAB1 捕捉/比较寄存器 1	Timer	344
TAB1CCR2	TAB1 捕捉/比较寄存器 2	Timer	346
TAB1CCR3	TAB1 捕捉/比较寄存器 3	Timer	348
TAB1CNT	TAB1 计数器读缓冲寄存器	Timer	350
TAB1CTL0	TAB1 控制寄存器 0	Timer	335
TAB1CTL1	TAB1 控制寄存器 1	Timer	336
TAB1DTC	TAB1 死区比较寄存器 1	Timer	583
TAB1IOC0	TAB1 I/O 控制寄存器 0	Timer	337
TAB1IOC1	TAB1 I/O 控制寄存器 1	Timer	338
TAB1IOC2	TAB1 I/O 控制寄存器 2	Timer	339
TAB1IOC3	TAB1 I/O 控制寄存器 3	Timer	587
TAB1OIC4	TAB1 I/O 控制寄存器 4	Timer	340
TAB1OPT0	TAB1 选择寄存器 0	Timer	341
TAB1OPT1	TAB1 选择寄存器 1	Timer	584
TAB1OPT2	TAB1 选择寄存器 2	Timer	585
TAB1OVIC	中断控制寄存器	INTC	1195
TANFC	噪音消除控制寄存器	Timer	231
TM0CMP0	TMM0 比较寄存器 0	Timer	571
TM0CTL0	TMM0 控制寄存器 0	Timer	572
TM0EQIC0	中断控制寄存器	INTC	1195
TM1CMP0	TMM1 比较寄存器 0	Timer	571
TM1CTL0	TMM1 控制寄存器 0	Timer	572
TM1EQIC0	中断控制寄存器	INTC	1195
TM2CMP0	TMM2 比较寄存器 0	Timer	571
TM2CTL0	TMM2 控制寄存器 0	Timer	572
TM2EQIC0	中断控制寄存器	INTC	1195
TM3CMP0	TMM3 比较寄存器 0	Timer	571
TM3CTL0	TMM3 控制寄存器 0	Timer	572

符号	名称	单元	页码
TM3EQIC0	中断控制寄存器	INTC	1195
TTNFC	噪音消除控制寄存器	Timer	458
TT0CCIC0	中断控制寄存器	INTC	1195
TT0CCIC1	中断控制寄存器	INTC	1195
TT0CCR0	TMT0 捕捉/比较寄存器 0	Timer	453
TT0CCR1	TMT0 捕捉/比较寄存器 1	Timer	455
TT0CNT	TMT0 计数器读缓冲寄存器	Timer	457
TT0CTL0	TMT0 控制寄存器 0	Timer	439
TT0CTL1	TMT0 控制寄存器 1	Timer	440
TT0CTL2	TMT0 控制寄存器 2	Timer	442
TT0IECIC	中断控制寄存器	INTC	1195
TT0IOC0	TMT0 I/O 控制寄存器 0	Timer	444
TT0IOC1	TMT0 I/O 控制寄存器 1	Timer	446
TT0IOC2	TMT0 I/O 控制寄存器 2	Timer	447
TT0IOC3	TMT0 I/O 控制寄存器 3	Timer	448
TT0OPT0	TMT0 选择寄存器 0	Timer	450
TT0OPT1	TMT0 选择寄存器 1	Timer	451
TT0OVIC	中断控制寄存器	INTC	1195
TT0TCW	TMT0 计数写入寄存器	Timer	457
UC0CTL0	UARTC0 控制寄存器 0	UARTC	731
UC0CTL1	UARTC0 控制寄存器 1	UARTC	757
UC0CTL2	UARTC0 控制寄存器 2	UARTC	758
UC0OPT0	UARTC0 选择控制寄存器 0	UARTC	733
UC0OPT1	UARTC0 选择控制寄存器 1	UARTC	735
UC0RIC	中断控制寄存器	INTC	1195
UC0RX	UARTC0 接收数据寄存器	UARTC	739
UC0RXL	UARTC0 接收数据寄存器 L	UARTC	739
UC0STR	UARTC0 状态寄存器	UARTC	737
UC0TIC	中断控制寄存器	INTC	1195
UC0TX	UARTC0 发送数据寄存器	UARTC	740
UC0TXL	UARTC0 发送数据寄存器 L	UARTC	740
UC1CTL0	UARTC1 控制寄存器 0	UARTC	731
UC1CTL1	UARTC1 控制寄存器 1	UARTC	757
UC1CTL2	UARTC1 控制寄存器 2	UARTC	758
UC1OPT0	UARTC1 选择控制寄存器 0	UARTC	733
UC1OPT1	UARTC1 选择控制寄存器 1	UARTC	735
UC1RIC	中断控制寄存器	INTC	1195
UC1RX	UARTC1 接收数据寄存器	UARTC	739
UC1RXL	UARTC1 接收数据寄存器 L	UARTC	739
UC1STR	UARTC1 状态寄存器	UARTC	737
UC1TIC	中断控制寄存器	INTC	1195
UC1TX	UARTC1 发送数据寄存器	UARTC	740
UC1TXL	UARTC1 发送数据寄存器 L	UARTC	740

符号	名称	单元	页码
UC2CTL0	UARTC2 控制寄存器 0	UARTC	731
UC2CTL1	UARTC2 控制寄存器 1	UARTC	757
UC2CTL2	UARTC2 控制寄存器 2	UARTC	758
UC2OPT0	UARTC2 选择控制寄存器 0	UARTC	733
UC2OPT1	UARTC2 选择控制寄存器 1	UARTC	735
UC2RIC	中断控制寄存器	INTC	1195
UC2RX	UARTC2 接收数据寄存器	UARTC	739
UC2RXL	UARTC2 接收数据寄存器 L	UARTC	739
UC2STR	UARTC2 状态寄存器	UARTC	737
UC2TIC	中断控制寄存器	INTC	1195
UC2TX	UARTC2 发送数据寄存器	UARTC	740
UC2TXL	UARTC2 发送数据寄存器 L	UARTC	740
UC3CTL0	UARTC3 控制寄存器 0	UARTC	731
UC3CTL1	UARTC3 控制寄存器 1	UARTC	757
UC3CTL2	UARTC3 控制寄存器 2	UARTC	758
UC3OPT0	UARTC3 选择控制寄存器 0	UARTC	733
UC3OPT1	UARTC3 选择控制寄存器 1	UARTC	735
UC3RIC	中断控制寄存器	INTC	1195
UC3RX	UARTC3 接收数据寄存器	UARTC	739
UC3RXL	UARTC3 接收数据寄存器 L	UARTC	739
UC3STR	UARTC3 状态寄存器	UARTC	737
UC3TIC	中断控制寄存器	INTC	1195
UC3TX	UARTC3 发送数据寄存器	UARTC	740
UC3TXL	UARTC3 发送数据寄存器 L	UARTC	740
UC4CTL0	UARTC4 控制寄存器 0	UARTC	731
UC4CTL1	UARTC4 控制寄存器 1	UARTC	757
UC4CTL2	UARTC4 控制寄存器 2	UARTC	758
UC4OPT0	UARTC4 选择控制寄存器 0	UARTC	733
UC4OPT1	UARTC4 选择控制寄存器 1	UARTC	735
UC4RIC	中断控制寄存器	INTC	1195
UC4RX	UARTC4 接收数据寄存器	UARTC	739
UC4RXL	UARTC4 接收数据寄存器 L	UARTC	739
UC4STR	UARTC4 状态寄存器	UARTC	737
UC4TIC	中断控制寄存器	INTC	1195
UC4TX	UARTC4 发送数据寄存器	UARTC	740
UC4TXL	UARTC4 发送数据寄存器 L	UARTC	740
UCKSEL	USB 时钟选择寄存器	USB	910
UFCKMSK	USB 函数控制寄存器	USB	910
UFIC0	中断控制寄存器	INTC	1195
UFIC1	中断控制寄存器	INTC	1195
UHCKMSK	USB 函数选择寄存器	USB	911
UHIC0	中断寄存器	INTC	1195
UHIC1	中断寄存器	INTC	1195
UHIC2	中断寄存器	INTC	1195

符号	名称	单元	页码
VSWC	系统等待控制寄存器	CPU	95
WDTE	看门狗定时器使能寄存器	WDT	674
WDTM2	看门狗定时器模式寄存器 2	WDT	672

附录D 指令集列表

D.1 规则

(1) 用于描述操作数的寄存器符号

寄存器符号	说明
reg1	通用寄存器:用作源寄存器.
reg2	通用寄存器:主要用作目的寄存器,在某指令里也用作源寄存器
reg3	通用寄存器:主要用作存储除法结果的余数,以及乘法结果的高32位。
bit#3	指明位号的3位数据。
immX	X位立即数。
dispX	X位置换数据。
regID	系统寄存器号。
vector	用来指明陷阱向量的5位数据(00H至1FH)。
cccc	用来显示状态码的4位数据。
sp	堆栈指针(r3)。
ep	元指针(r30)。
listX	X项寄存器列表。

(2) 用于描述操作码的寄存器符号

寄存器符号	说明
R	指定 reg1 或 regID 的 1 位代码数据。
r	指定 reg2 的 1 位代码数据。
w	指定 reg3 的 1 位代码数据。
d	1 位置换数据。
l	1 位立即数 (指示立即数的高位)。
i	1 位立即数。
cccc	显示条件码的 4 位数据。
CCCC	显示 Bcond 指令条件码的 4 位数据。
bbb	指定位号的 3 位数据。
L	在寄存器列表中指示编程寄存器的 1 位数据。

(3) 用于操作的寄存器符号

寄存器符号	说明
←	为之输入
GR []	通用寄存器
SR []	系统寄存器
zero-extend (n)	用零扩展 n 直至满字长。
sign-extend (n)	用符号扩展 n 直至满字长。
load-memory (a, b)	从地址 a 起读取存储长度为 b 的数据。
store-memory (a, b, c)	向地址 c 起存储长度为 a 的字节写入数据 b。
load-memory-bit (a, b)	从地址 a 载入位 b。
store-memory-bit (a, b, c)	写入 c 至地址 a 的 b 位。
saturated (n)	执行 n 的饱和运算 (n 为 2 的补码)。 如此的话, 结果如下处理: n ≥ 7FFFFFFFH, 令其为 7FFFFFFFH。 n ≤ 80000000H, 令其为 80000000H。
result	反映标志的结果。
Byte	字节 (8 位)
Halfword	半字 (16 位)
Word	字 (32 位)
+	加
-	减
	位并拼
×	乘
÷	除
%	除法结果的余数
AND	逻辑乘
OR	逻辑加
XOR	逻辑异或
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) 用于执行时钟的寄存器符号

寄存器符号	说明
i	如果执行第一条指令后紧接着执行另一条指令 (流式执行)。
r	如果执行第一条指令后紧接着重复执行相同指令 (重复执行)。
l	如果紧接之后的指令执行使用当前指令执行结果 (等待执行)。

(5) 用于标志操作的寄存器符号

标识符	说明
(Blank)	无改变。
0	清零。
X	根据结果置位或清 0。
R	恢复先前的保存值。

(6) 条件码

条件码 (cccc)	条件公式	说明
0 0 0 0	$OV = 1$	溢出
1 0 0 0	$OV = 0$	无溢出
0 0 0 1	$CY = 1$	进位 小(相比小)
1 0 0 1	$CY = 0$	无进位 不小(相比大于或等于)
0 0 1 0	$Z = 1$	0
1 0 1 0	$Z = 0$	非 0
0 0 1 1	$(CY \text{ or } Z) = 1$	不高(相比小于或等于 1)
1 0 1 1	$(CY \text{ or } Z) = 0$	高(相比大于)
0 1 0 0	$S = 1$	负
1 1 0 0	$S = 0$	正
0 1 0 1	–	恒为(无条件的)
1 1 0 1	$SAT = 1$	饱和
0 1 1 0	$(S \text{ xor } OV) = 1$	小于有符号数。
1 1 1 0	$(S \text{ xor } OV) = 0$	大于或等于有符号数。
0 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 1$	小于或等于有符号数。
1 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 0$	大于有符号数。

D.2 指令集 (按字母先后次序)

(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	x	x	x	x	
	imm5,reg2	rrrrr010010iiii	GR[reg2]←GR[reg2]+sign-extend(imm5)	1	1	1	x	x	x	x	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1	x	x	x	x	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]	1	1	1		0	x	x	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)	1	1	1		0	x	x	
Bcond	disp9	dddd1011dddcccc 注 1	如果条件满足 则: PC←PC+sign-extend(disp9)	如果条件满足	2	2	2				
			如果条件不满足	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 www01101000010	GR[reg3]←GR[reg2] (23 : 16) GR[reg2] (31 : 24) GR[reg2] (7 : 0) GR[reg2] (15 : 8)	1	1	1	x	0	x	x	
BSW	reg2,reg3	rrrrr11111100000 www01101000000	GR[reg3]←GR[reg2] (7 : 0) GR[reg2] (15 : 8) GR [reg2] (23 : 16) GR[reg2] (31 : 24)	1	1	1	x	0	x	x	
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				x	
	reg2,[reg1]	rrrrr111111RRRRR 000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)	3	3	3				x	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiii www011000cccc0	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	rrrrr111111RRRR www011001cccc0	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	x	x	x	x	
	imm5,reg2	rrrrr010011iiii	result←GR[reg2]-sign-extend(imm5)	1	1	1	x	x	x	x	
CTRET		000001111100000 0000000101000100	PC←CTPC PSW←CTPSW	3	3	3	R	R	R	R	R
DBRET		000001111100000 0000000101000110	PC←DBPC PSW←DBPSW	3	3	3	R	R	R	R	R

助记符	操作数	操作码	操作	执行时钟			标志					
				i	r	l	CY	OV	S	Z	SAT	
DBTRAP		1111100001000000	DBPC←PC+2 (restored PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3						
DI		0000011111000000 0000000101100000	PSW.ID←1	1	1	1						
DISPOSE	imm5,list12	0000011001iiiiL LLLLLLLLLLLL000000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 重复以上两个步骤直到表 12 中所有寄存器被载入。	n+1 注4	n+1 注4	n+1 注4						
	imm5,list12,[reg1]	0000011001iiiiL LLLLLLLLLLLLRRRRR 注5	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 重复以上两个步骤直到表 12 中所有寄存器载入 PC←GR[reg1]	n+3 注4	n+3 注4	n+3 注4						
DIV	reg1,reg2,reg3	rrrrr11111RRRRR wwwww01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		x	x	x		
DIVH	reg1,reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1] ^{注6}	35	35	35		x	x	x		
	reg1,reg2,reg3	rrrrr11111RRRRR wwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1] ^{注6} GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		x	x	x		
DIVHU	reg1,reg2,reg3	rrrrr11111RRRRR wwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1] ^{注6} GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		x	x	x		
DIVU	reg1,reg2,reg3	rrrrr11111RRRRR wwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		x	x	x		
EI		1000011111000000 0000000101100000	PSW.ID←0	1	1	1						
HALT		0000011111000000 0000000100100000	停止	1	1	1						
HSW	reg2,reg3	rrrrr111111000000 wwwww01101000100	GR[reg3]←GR[reg2]((15:0) GR[reg2] (31:16))	1	1	1	x	0	x	x		
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	rrrrr111000RRRRR 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 dddddddddddddd0 注 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	注 11						
LDSR	reg2,regID	rrrrr111111RRRRR 000000000100000 注 12	SR[regID]←GR[reg2]	除非 regID = PSW	1	1	1					
			regID = PSW		1	1	1	x	x	x	x	x
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR dddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	注 11						
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR dddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)	1	1	注 11						
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]	1	1	1						
	imm5,reg2	rrrrr010000iiii	GR[reg2]←sign-extend(imm5)	1	1	1						
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii iiiiiiiiiiiiiiii	GR[reg1]←imm32	2	2	2						
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1						
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 0 ¹⁶)	1	1	1						
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000 注 14	GR[reg3] GR[reg2]←GR[reg2]xGR[reg1]	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 注 14	GR[reg3] GR[reg2]←GR[reg2]xGR[reg1]	1	4	5						
	imm9,reg2,reg3	rrrrr111111iiii wwwww0100111110 注 13	GR[reg3] GR[reg2]←GR[reg2]xzero-extend(imm9)	1	4	5						
NOP		0000000000000000	至少一个时钟周期内不作任何操作。	1	1	1						
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])	1	1	1		0	x	x		
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) 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 000000011100010	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 iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiL LLLLLLLLLLLL00001	Store-memory(sp-4,GR[reg in list12],Word) sp←sp-4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 注4	n+1 注4	n+1 注4					
	list12,imm5, sp/imm ^{※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 0000000101000000	如果 PSW.EP=1 那么 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
SAR	reg1,reg2	rrrrr111111RRRRR 000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr1111110cccc 0000001000000000	如果条件满足 then GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H else GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1					
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5,reg2	rrrrr010001iiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5))	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]-GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]-sign-extend(imm16))	1	1	1	×	×	×	×	×
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]-GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc,reg2	rrrrr1111110cccc 0000000000000000	If conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1					

助记符	操作数	操作码	操作	执行时钟			标志					
				i	r	l	CY	OV	S	Z	SAT	
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR ddddddddddddddd	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					x	
	reg2,[reg1]	rrrrr111111RRRRR 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					x	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	x	0	x	x		
	imm5,reg2	rrrrr010110iiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	x	0	x	x		
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	x	0	x	x		
	imm5,reg2	rrrrr010100iiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	x	0	x	x		
SLD.B	disp7[ep],reg2	rrrrr0110ddddd	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	rrrrr1000ddddd 注 19	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	注9						
SLD.HU	disp5[ep],reg2	rrrrr0000111ddd 注 s 18, 20	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	注9						
SLD.W	disp8[ep],reg2	rrrrr1010ddddd0 注 21	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	注9						
SST.B	reg2,disp7[ep]	rrrrr01111ddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1						
SST.H	reg2,disp8[ep]	rrrrr1001ddddd 注 19	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Halfword)	1	1	1						
SST.W	reg2,disp8[ep]	rrrrr1010ddddd1 注 21	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1						
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1						
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd0 注 8	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Halfword)	1	1	1						
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Word)	1	1	1						
STSR	regID,reg2	rrrrr111111RRRRR 000000001000000	GR[reg2]←SR[regID]	1	1	1						

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]-GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]-GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	0000000010RRRRR	adr←-(PC+2) + (GR [reg1] logically shift left by 1) PC←-(PC+2) + (sign-extend (Load-memory (adr,Halfword)) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←-sign-extend (GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←-sign-extend (GR[reg1] (15 : 0))	1	1	1					
TRAP	vector	000001111111iiii 0000000100000000	EIPC ←-PC+4 (Restored PC) EIPSW ←-PSW ECR.EICC ←-Interrupt code PSW.EP ←-1 PSW.ID ←-1 PC ←-00000040H (when vector is 00H to 0FH) 00000050H (when vector is 10H to 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1	0	×	×		
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR ddddddddddddddd	adr←-GR[reg1]+sign-extend(disp16) Z flag←-Not (Load-memory-bit (adr,bit#3))	3 注3	3 注3	3 注3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←-GR[reg1] Z flag←-Not (Load-memory-bit (adr,reg2))	3 注3	3 注3	3 注3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←-GR[reg2] XOR GR[reg1]	1	1	1	0	×	×		
XORI	imm16,reg1,reg2	rrrrr110101RRRRR iiiiiiiiiiiiiiii	GR[reg2]←-GR[reg1] XOR zero-extend (imm16)	1	1	1	0	×	×		
ZXB	reg1	00000000100RRRRR	GR[reg1]←-zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←-zero-extend (GR[reg1] (15 : 0))	1	1	1					

- 注
1. ddddddd: disp9 的高 8 位。
 2. 如果之前紧邻一条指令改写 PSW 的内容, 则为 3。
 3. 如果没有等待状态 (3 + 读取访问等待状态数)。
 4. n 为列表 12 中加载寄存器的总数。(根据等待状态数目, n 为列表 12 中加载寄存器的总数。如果 n = 0, 和 n = 1 时的操作相同)
 5. RRRRR: 除了 00000 之外。
 6. 只有低半字有效。
 7. ddddddddddddddddddd: disp22 的高 21 位。
 8. ddddddddddddddd: disp16 的高 15 位。
 9. 依照等待状态的数目(如果没有等待状态, 则为 1)。
 10. b: disp16 的位 0。
 11. 依照等待状态的数目(如果没有等待状态, 则为 2)。

- 注 12. 本指令中，为了助记符描述方便，使源寄存器为 `reg2`，而操作码位置上使用 `reg1` 寄存器。所以，助记符描述中寄存器的含义和其它指令有所不同。
- `rrrrr` = 指定 `regID`
`RRRRR` = 指定 `reg2`
13. `iiii`: `imm9` 的低 5 位。
`IIII`: `imm9` 的高 4 位。
14. 不要将为通用寄存器 `reg1` 和 `reg3` 指定同一寄存器。
15. `sp/imm`: 由次操作码的位 19 和位 20 指定。
16. `ff = 00`: 将 `sp` 载入 `ep`。
01: 将符号扩展的 16 位立即数(位 47 至 32)载入 `ep`。
10: 将经过 16 位逻辑左移的 16 位立即数 (位 47 至 32) 载入 `ep`。
11: 将 32 位立即数 (位 63 至 32) 载入 `ep`。
17. If `imm = imm32`, `n + 3` 时钟。
18. `rrrrr`: 非 00000。
19. `dddddd`: `disp8` 的高 8 位。
20. `dddd`: `disp5` 的高 4 位。
21. `dddddd`: `disp8` 的高 6 位。

详细信息请联系:

中国区

MCU 技术支持热线:

电话: +86-400-700-0606 (普通话)

服务时间: 9:00-12:00, 13:00-17:00 (不含法定节假日)

网址:

<http://www.cn.necel.com/> (中文)

<http://www.necel.com/> (英文)

[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

[成都]

日电电子(中国)有限公司成都分公司

四川省成都市二环路南三段 15 号

天华大厦 608 室

电话: (+86)28-8512-5224

传真: (+86)28-8512-5334

[长春]

日电电子(中国)有限公司长春分公司

吉林省长春市朝阳区

西安大路 727 号中银大厦 A 座 1609 室

电话: (+86)431-8859-7533 / 8859-8533

传真: (+86)431-8680-2944

[大连]

日电电子(中国)有限公司长春分公司

大连市中山路 88 号天安国际大厦 2701 室

电话: (+86)411-8230-8815 / 8230-8825

传真: (+86)411-8230-8835