

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

V850ES/JG2

32 位单片微控制器

硬件

μ PD70F3715

μ PD70F3716

μ PD70F3717

μ PD70F3718

μ PD70F3719

[备忘录]

① 输入引脚处的电压波形

输入噪音或一个反射波引起的波形失真可能导致错误发生。如果由于噪音等的影响使CMOS设备的输入电压范围保持在 V_{IL} (MAX) 和 V_{IH} (MIN) 之间, 设备可能发生错误。在输入电平固定时以及输入电平从 V_{IL} (MAX) 过渡到 V_{IH} (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 V_{DD} 或GND。如果有可能尽量定义为输出引脚。对未使用引脚的处理因设备而异, 必须遵循与设备相关的规定和说明。

③ ESD防护措施

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

④ 初始化之前的状态

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

⑤ 电源开关顺序

在一个设备的内部操作和外部接口使用不同的电源的情况下, 按照规定, 应先在接通内部电源之后再

接通外部电源。当关闭电源时, 按照规定, 先关闭外部电源再关闭内部电源。如果电源开关顺序颠倒, 可能会导致设备的内部组件过电压, 产生异常电流, 从而引起内部组件的误操作和性能的退化。

对于每个设备电源的正确开关顺序必须依据设备的规范说明分别进行判断。

⑥ 电源关闭状态下的输入信号

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

每个设备电源关闭时的信号输入必须依据设备的规范说明分别进行判断。

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

MINICUBE 是 NEC Electronics Corporation 在日本和德国的注册商标以及在美国的商标。

EEPROM 是 NEC Electronics Corporation 的商标。

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 **Windows NT** 是 Microsoft Corporation 在美国及其他国家的注册商标或商标。

PC/AT 是 International Business Machines Corporation 的商标。

SPARCstation 是 SPARC International, Inc 的商标。

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

TRON 是 The Realtime Operation System Nucleus 的缩写。

ITRON 是 Industrial TRON 的缩写。

- 本文档信息发布于 2006 年 8 月。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。本文件所登载内容的错误，本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的，对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或其他人所有的专利、版权以及其它知识产权作出任何明示或默示的许可及授权。
- 本文件中的电路、软件以及相关信息仅用以说明半导体产品的运作和应用实例。用户如在设备设计中应用本文件中的电路、软件以及相关信息，应自行负责。对于用户或其他人因使用了上述电路、软件以及相关信息而引起的任何损失，本公司概不负责。
- 虽然本公司致力于提高半导体产品的质量及可靠性，但用户应同意并知晓，我们仍然无法完全消除出现产品缺陷的可能。为了最大限度地减少因本公司半导体产品故障而引起的对人身、财产造成损害（包括死亡）的危险，用户务必在其设计中采用必要的安全措施，如冗余度、防火和防故障等安全设计。
- 本公司产品质量分为：

“标准等级”、“专业等级”以及“特殊等级”三种质量等级。

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

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

“专业等级”： 运输设备（汽车、火车、船舶等），交通用信号控制设备，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或为日本电气电子株式会社（定义如上）开发或制造的产品。

引言

读者对象

本手册适用于那些希望了解 V850ES/JG2 产品的功能并准备利用其进行应用系统开发的用户。

目的

本手册用于帮助用户了解下面**组件**中描述的功能。

组件

V850ES/JG2 产品手册主要分为两个部分: 硬件(本手册)和架构(**V850ES Architecture User's Manual**)。

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

手册使用方法

在阅读本手册前, 读者应掌握电子工程、逻辑电路和微控制器等方面的一般知识。

如何获悉 V850ES/JG2 的全部功能

→ 请依照**目录**阅读本手册。

如何获取某寄存器的详细信息

→ 请参考 **附录 B 寄存器索引**

寄存器格式

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

如何理解具体的指令功能

→ 请参考 **V850ES Architecture User's Manual**。

如何了解 V850ES/JG2 产品的电气指标

→ 请参考第 28 章 **ELECTRICAL SPECIFICATIONS**。

如何了解指令的详细功能

→ 请参阅 **V850ES Architecture User's Manual**。

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

关键的修改都用记号<R>标识。只需要把一个<R>复制到 PDF 文件的“Find What: ”区域里面就可以方便地找到关键修改。

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

相关文档

本手册中指出的相关文档包括了最初的版本，但未注明。

V850ES/JG2 产品相关的文档

文档名称	文档编号
V850ES 架构用户手册	U15943E
V850ES/JG2 硬件用户手册	本手册

开发工具相关的文档

文档名称		文档编号
QB-V850ESSX2 在线仿真器		U17091E
QB—V850MINI 片上调试仿真器		U17638E
QB—MINI2 带编程功能的片上调试仿真器		准备中
CA850 Ver. 3.00 C 编译器包	操作	U17293E
	C 语言	U17291E
	汇编语言	U17292E
	链接伪指令	U17294E
PM+ Ver. 6.20 工程管理器		U17990E
ID850QB Ver. 3.20 集成调试器	操作	U17964E
SM850 Ver. 2.50 系统软仿真器	操作	U16218E
SM850 Ver. 2.00 或更高版系统软仿真器	开放式外部用户接口规范	U14873E
SM+ 系统软仿真器	操作	U17246E
	开放式用户接口	U17247E
RX850 Ver. 3.20 实时操作系统	基础	U13430E
	安装	U17419E
	技术	U13431E
	任务调试器	U17420E
RX850 Pro Ver. 3.20 实时操作系统	基础	U13773E
	安装	U17421E
	技术	U13772E
	任务调试器	U17422E
AZ850 Ver. 3.30 系统性能分析器		U17423E
PG-FP4 Flash 存储器编程器		U15260E

目 录

第一章 介绍	18
1.1 概要	18
1.2 特性	21
1.3 应用领域	22
1.4 订货信息	22
1.5 引脚图 (俯视图)	23
1.6 功能模块配置	26
1.6.1 内部结构框图	26
1.6.2 内部单元	27
第二章 引脚功能	30
2.1 引脚功能列表	30
2.2 引脚状态	40
2.3 I/O引脚电路类型, I/O缓冲供电电平, 和不使用引脚的连接	41
2.4 注意事项	45
第三章 CPU功能	46
3.1 特点	46
3.2 CPU 寄存器组	47
3.2.1 程序寄存器组	48
3.2.2 系统寄存器组	49
3.3 操作模式	55
3.3.1 指定操作模式	55
3.4 地址空间	56
3.4.1 CPU 地址空间	56
3.4.2 CPU 地址空间的绕回(Wraparound)	57
3.4.3 存储器映射	58
3.4.4 区域	60
3.4.5 地址空间的推荐用法	67
3.4.6 周边 I/O 寄存器	70
3.4.7 特殊寄存器	80
3.4.8 注意事项	84
第四章 端口功能	88
4.1 特点	88
4.2 端口的基本配置	88
4.3 端口配置	89
4.3.1 端口 0	94
4.3.2 端口 1	97
4.3.3 端口 3	98
4.3.4 端口 4	104
4.3.5 端口 5	106
4.3.6 端口 7	110
4.3.7 端口 9	112
4.3.8 端口 CM	120
4.3.9 端口 CT	122
4.3.10 端口 DH	124
4.3.11 端口 DL	126
4.4 框图	129

4.5	当使用复用功能时的端口寄存器设置	159
4.6	注意事项	167
4.6.1	设置端口引脚的注意事项	167
4.6.2	端口 n 寄存器 (Pn) 位操作指令的注意事项	170
4.6.3	片上调试引脚的注意事项	171
4.6.4	P05/INTP2/DRST 引脚的注意事项	171
4.6.5	当电源打开时P10, P11, 和 P53 引脚的注意事项	171
4.6.6	滞后性	171
第五章	总线控制功能	172
5.1	特点	172
5.2	总线控制引脚	173
5.2.1	当访问内置ROM, 内置RAM或片上周边I/O时的引脚状态	173
5.2.2	各操作模式下的引脚状态	173
5.3	存储器块(block)功能	174
5.4	外部总线接口模式控制功能	175
5.5	总线访问	176
5.5.1	访问所需的时钟数	176
5.5.2	总线宽度设置功能	176
5.5.3	访问的总线宽度	177
5.6	等待功能	184
5.6.1	可编程等待功能	184
5.6.2	外部等待功能	185
5.6.3	可编程等待和外部等待之间的关系	186
5.6.4	可编程地址等待功能	187
5.7	空闲状态插入功能	188
5.8	总线保持功能	189
5.8.1	功能概述	189
5.8.2	总线保持流程	190
5.8.3	节能模式下的操作	190
5.9	总线优先级	191
5.10	总线时序	192
第六章	时钟产生功能	198
6.1	概述	198
6.2	结构	199
6.3	寄存器	201
6.4	操作	206
6.4.1	各时钟的操作	206
6.4.2	时钟输出功能	206
6.5.1	概述	207
6.5.2	寄存器	207
6.5.3	使用方法	210
第七章	16 位定时器/事件计数器P (TMP)	211
7.1	概述	211
7.2	功能	211
7.3	配置	212
7.4	寄存器	214
7.5	操作	226
7.5.1	间隔定时器模式 (TPnMD2 到 TPnMD0 位 = 000)	227
7.5.2	外部事件计数模式 (TPnMD2 到 TPnMD0 位 = 001)	237

7.5.3	外部触发脉冲输出模式 (TPnMD2 到 TPnMD0 位 = 010)	245
7.5.4	单脉冲输出模式 (TPnMD2 到 TPnMD0 位 = 011)	257
7.5.5	PWM 输出模式 (TPnMD2 到 TPnMD0 位 = 100)	264
7.5.6	自由运行定时器模式 (TPnMD2 到 TPnMD0 位 = 101)	273
7.5.7	脉宽测量模式 (TPnMD2 到 TPnMD0 位 = 110)	290
7.5.8	定时器输出操作	296
7.6	选择器功能	297
7.7	注意事项	298
第八章 16 位定时器/事件计数器Q (TMQ)		299
8.1	概述	299
8.2	功能	299
8.3	配置	300
8.4	寄存器	301
8.5	操作	318
8.5.1	间隔定时器模式 (TQ0MD2 至 TQ0MD0 位 = 000)	319
8.5.2	外部事件计数模式 (TQ0MD2 至 TQ0MD0 位 = 001)	328
8.5.3	外部触发脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 010)	337
8.5.4	单次脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 011)	350
8.5.5	PWM 输出模式 (TQ0MD2 至 TQ0MD0 位 = 100)	359
8.5.6	自由运行定时器模式 (TQ0MD2 至 TQ0MD0 位 = 101)	370
8.5.7	脉宽测量模式 (TQ0MD2 至 TQ0MD0 位 = 110)	390
8.5.8	定时器输出操作	396
8.6	注意事项	397
第九章 16 位间隔定时器M (TMM)		398
9.1	概述	398
9.2	配置	399
9.3	寄存器	400
9.4	操作	401
9.4.1	间隔定时器模式	401
9.4.2	注意事项	405
第十章 钟表定时器功能		406
10.1	功能	406
10.2	配置	407
10.3	控制寄存器	409
10.4	操作	413
10.4.1	钟表定时器的操作	413
10.4.2	间隔定时器的操作	414
10.4.3	注意事项	415
第十一章 看门狗定时器 2 的功能		416
11.1	功能	416
11.2	配置	417
11.3	寄存器	418
11.4	操作	420
第十二章 实时输出功能 (RTO)		421
12.1	功能	421
12.2	配置	422
12.3	寄存器	424

12.4	操作	426
12.5	用法	427
12.6	注意事项	427
第十三章	A/D 转换器	428
13.1	概览	428
13.2	功能	428
13.3	配置	429
13.4	寄存器	432
13.5	操作	443
13.5.1	基本操作	443
13.5.2	转换操作时序	444
13.5.3	触发模式	445
13.5.4	操作模式	447
13.5.5	电压不足比较模式	451
13.6	注意事项	456
13.7	A/D转换器特征表的阅读方法	460
第十四章	D/A转换器	464
14.1	概述	464
14.2	配置	464
14.3	寄存器	465
14.4	操作	467
14.4.1	正常模式下的操作	467
14.4.2	实时输出模式下的操作	467
14.4.3	注意事项	468
第十五章	异步串行接口A (UARTA)	469
15.1	UARTA和其它串行接口的模式转换	469
15.1.1	CSIB4 和UARTA0 模式转换	469
15.1.2	UARTA2 和I ² C00 模式转换	470
15.1.3	UARTA1 和 I ² C02 模式转换	471
15.2	特性	472
15.3	配置	473
15.4	寄存器	475
15.5	中断请求信号	481
15.6	操作	482
15.6.1	数据格式	482
15.6.2	SBF 发送/接收格式	484
15.6.3	SBF发送	486
15.6.4	SBF接收	487
15.6.5	UART发送	488
15.6.6	连续发送	489
15.6.7	UART接收	491
15.6.8	接收错误	492
15.6.9	校验类型和操作	494
15.6.10	接收数据的噪声过滤器	495
15.7	专用波特率发生器	496
15.8	注意事项	504
第十六章	3 线长度可变串行 I/O (CSIB)	505
16.1	CSIB和其它串行接口的模式转换	505
16.1.1	CSIB4 和 UARTA0 模式转换	505

16.1.2	CSIB0 和 I ² C01 模式转换	506
16.2	特征	507
16.3	配置	508
16.4	寄存器	510
16.5	中断请求信号	517
16.6	操作	518
16.6.1	单独传送模式 (主机模式, 发送模式)	518
16.6.2	单独传送模式 (主机模式, 接收模式)	520
16.6.3	单独传送模式 (主机模式, 发送/接收模式)	522
16.6.4	单独传送模式 (从机模式, 发送模式)	524
16.6.5	单独传送模式 (从机模式, 接收模式)	526
16.6.6	单独传送模式 (从机模式, 发送/接收模式)	528
16.6.7	连续传送模式 (主机模式, 发送模式)	530
16.6.8	连续传送模式 (主机模式, 接收模式)	532
16.6.9	连续传送模式 (主机模式, 发送/接收模式)	535
16.6.10	连续传送模式 (从机模式, 发送模式)	539
16.6.11	连续传送模式 (从机模式, 接收模式)	541
16.6.12	连续传送模式 (从机模式, 发送/接收模式)	544
16.6.13	接收错误	548
16.6.14	时钟时序	549
16.7	输出引脚	551
16.8	波特率发生器	552
16.8.1	波特率	553
16.9	注意事项	554
第十七章	I ² C 总线	555
17.1	I ² C 总线和其它串行总线的模式转换	555
17.1.1	UARTA2 和 I ² C00 模式的模式转换	555
17.1.2	CSIB0 和 I ² C01 的模式转换	556
17.1.3	UARTA1 和 I ² C02 的模式转换	557
17.2	特征	558
17.3	配置	559
17.4	寄存器	563
17.5	I ² C 总线模式功能	579
17.5.1	引脚配置	579
17.6	I ² C 总线定义和控制方法	580
17.6.1	开始条件	580
17.6.2	地址	581
17.6.3	传送方向规定	582
17.6.4	ACK	583
17.6.5	停止条件	584
17.6.6	等待状态	585
17.6.7	等待状态取消方法	587
17.7	I ² C 中断请求信号 (INTIICn)	588
17.7.1	主设备操作	588
17.7.2	从设备操作 (当接收从地址数据时 (地址匹配))	591
17.7.3	从设备操作 (当接收扩展码时)	595
17.7.4	不通信时操作	599
17.7.5	仲裁失败操作 (仲裁失败后作为从设备操作)	599
17.7.6	仲裁失败时的操作 (仲裁失败后无通信)	601
17.8	中断请求信号 (INTIICn) 的产生时序和等待控制	608
17.9	地址匹配监测方法	610

17.10	错误监测	610
17.11	扩展码	610
17.12	仲裁	611
17.13	唤醒功能	612
17.14	通信保留	613
17.14.1	通信保留功能允许时(IICFn.IICRSVn 位 = 0)	613
17.14.2	通信保留功能禁止时 (IICFn.IICRSVn 位 = 1)	617
17.15	注意事项	618
17.16	通信操作	619
17.16.2	在多主系统里的主操作	621
17.16.3	从设备操作	624
17.17	数据通信时序	627
第十八章 DMA功能 (DMA控制器)		634
18.1	特性	634
18.2	配置	635
18.3	寄存器	636
18.4	传送目标	643
18.5	传送模式	643
18.6	传送类型	644
18.7	DMA 信道优先级	645
18.8	与DMA 传送相关的时间	645
18.9	DMA传送开始的因素	646
18.10	DMA 中止因素	647
18.11	DMA传送结束	647
18.12	操作时序	647
18.13	注意事项	652
第十九章 中断/异常处理功能		657
19.1	特性	657
19.2	非屏蔽中断	661
19.2.1	操作	663
19.2.2	还原	664
19.2.3	NP标志	665
19.3	可屏蔽中断	666
19.3.1	操作	666
19.3.2	恢复	668
19.3.3	可屏蔽中断的优先级	669
19.3.4	中断控制寄存器 (xxICn)	673
19.3.5	中断屏蔽寄存器 0 到 3 (IMR0 到 IMR3)	675
19.3.6	正在进行服务优先级寄存器(ISPR)	677
19.3.7	ID 标志	678
19.3.8	看门狗定时器模式寄存器 2(WDTM2)	678
19.4	软件异常	679
19.4.1	操作	679
19.4.2	恢复	680
19.4.3	EP标志	681
19.5	异常陷阱	682
19.5.1	非法的操作代码定义	682
19.5.2	调试陷阱	684
19.6	外部中断请求输入引脚(NMI和INTP0 到INTP7)	686
19.6.1	噪声消除	686

19.6.2	边沿检测	686
19.7	CPU的中断确认时间	691
19.8	CPU不对中断进行确认的时期	692
19.9	注意事项	692
第二十章	按键中断功能	693
20.1	功能	693
20.2	寄存器	694
20.3	注意事项	694
第二十一章	待机功能	695
21.1	概述	695
21.2	寄存器	697
21.3	HALT 模式	700
21.3.1	设置和操作状态	700
21.3.2	释放HALT模式	700
21.4	IDLE1 模式	702
21.4.1	设置和操作状态	702
21.4.2	释放IDLE1 模式	702
21.5	IDLE2 模式	704
21.5.1	设置和操作状态	704
21.5.2	释放IDLE2 模式	704
21.5.3	当释放IDLE2 模式时保证设置时间	706
21.6	STOP模式	707
21.6.1	设置和操作状态	707
21.6.2	释放STOP模式	707
21.6.3	当释放STOP模式时保证振荡稳定时间	710
21.7	副时钟操作模式	711
21.7.1	设置和操作状态	711
21.7.2	释放副时钟操作模式	711
21.8	副IDLE模式	713
21.8.1	设置和操作状态	713
21.8.2	释放副IDLE模式	713
第二十二章	复位功能	715
22.1	概述	715
22.2	检测复位源的寄存器	716
22.3	操作	717
22.3.1	由RESET引脚引起的复位操作	717
22.3.2	由看门狗定时器 2 引起的复位操作	719
22.3.3	低电压检测器引起的复位操作	721
22.3.4	复位释放后的操作	722
22.3.5	复位功能操作流程	725
第二十三章	时钟监控器	726
23.1	功能	726
23.2	结构	726
23.3	寄存器	727
23.4	操作	728
第二十四章	低电压检测器 (LVI)	731
24.1	功能	731

24.2	配置	731
24.3	寄存器	732
24.4	操作	734
24.4.1	用作内部复位信号	734
24.4.2	用作中断	735
24.5	RAM 保持电压检测操作	736
24.6	仿真功能	737
第二十五章 调节器		738
25.1	概论	738
25.2	操作	739
第二十六章 FLASH 存储器		740
26.1	特性	740
26.2	存储器配置	741
26.3	功能概述	742
26.4	通过专用Flash编程器重写	745
26.4.1	编程环境	745
26.4.2	通信模式	746
26.4.3	Flash存储器控制	754
26.4.4	通信模式的选择	755
26.4.5	通信命令	756
26.4.6	引脚连接	757
26.5	自编程重写	761
26.5.1	概要	761
26.5.2	特性	762
26.5.3	标准自编程流程	763
26.5.4	Flash 功能	764
26.5.5	引脚处理	764
26.5.6	使用的内部资源	765
第二十七章 片上调试功能		766
27.1	使用DCU调试	767
27.1.1	连接电路示例	767
27.1.2	接口信号	767
27.1.3	可屏蔽功能	769
27.1.4	寄存器	769
27.1.5	寄存器 操作	771
27.1.6	注意事项	771
27.2	不使用DCU调试	773
27.2.1	电路连接示例	773
27.2.2	可屏蔽功能	774
27.2.3	用户资源的保证	775
27.2.4	注意事项	781
27.3	ROM 安全功能	783
27.3.1	安全 ID	783
27.3.2	设置	784
第二十八章 电气特性		786
第二十九章 封装图		821

第三十章 推荐焊接条件	823
附录 A 开发工具	824
A.1 软件包	826
A.2 语言处理软件	826
A.3 控制软件	826
A.4 调试工具 (硬件)	827
A.4.1 使用IECUBE QB-V850ESSX2	827
A.4.2 使用MINICUBE QB-V850MINI	830
A.4.3 使用MINICUBE2 QB-MINI2	831
A.5 调试工具 (软件)	832
A.6 嵌入式软件	833
A.7 Flash 存储器写入工具	834
附录 B 寄存器索引	835
附录 C 指令集列表	845
C.1 常规指令	845
C.2 指令集 (按字母顺序)	848
附录 D 注意事项表	855
附录 E 修订历史	891
E.1 本版中主要修订之处	891

第一章 介绍

V850ES/JG2 是日电子公司(NEC Electronics)的 V850 系列单片微控制器产品之一，主要应用在实时控制应用中的低功耗操作。

1.1 概要

V850ES/JG2 是一款 32 位单片微控制器，它包含了 V850ES CPU 内核以及诸如 ROM/RAM、定时器/计数器、串行接口、A/D 转换器和 D/A 转换器等外围设备功能。

除了实时响应特性和基本指令单时钟执行特点(1-clock-pitch)之外，V850ES/JG2 还具有由硬件乘法器来实现的乘法指令、饱和运算指令、位操作指令等为数字伺服控制应用而优化的指令。此外在实时控制系统中，V850ES/JG2 可以以极高的性能-功耗比实现那些需要低功耗的应用，例说家用音响、打印机和数字家电。

表 1-1 列出了 V850ES/JG2 系列的所有产品。

同时我们还提供在 V850ES/JG2 基础上扩展了 I/O、定时器/计数器、串行接口等功能的 V850ES/JJ2 产品。参照表 1-2 V850ES/JJ2 产品列表。

表 1-1. V850ES/JG2 产品列表

产品名称		μPD70F3715	μPD70F3716	μPD70F3717	μPD70F3718	μPD70F3719
内部存储器	Flash 存储器	128KB	256KB	384KB	512KB	640KB
	RAM	12KB	24KB	32KB	40KB	48KB
存储器容量	逻辑空间	64MB				
	外部存储区	16MB				
外部总线接口		地址总线: 22 位 数据总线: 8/16 位 多路总线模式/独立总线模式				
通用寄存器		32 位×32 寄存器				
主时钟 (振荡频率)		陶瓷/水晶/外部时钟 (在 PLL 模式下: $f_x=2.5\text{ MHz}$ (倍率为 4) 或者 $f_x=2.5\text{ MHz}$ (倍率为 8), 在 clock through 模式下: $f_x=2.5 \sim 10\text{ MHz}$)				
子时钟 (振荡频率)		水晶/外部时钟 ($f_{XT}=32.768\text{ KHz}$)				
内部晶振		$f_R=200\text{ KHz}$ (TYP)				
最小指令执行时间		50ns (主时钟 (f_{xx}) =20MHz)				
DSP 功能		32×32 = 64: 200 ~ 250 ns (在20 MHz时) 32×32 + 32 = 32: 300 ns (在 20 MHz时) 16×16 = 32: 50 ~ 100 ns (在20 MHz时) 16×16 + 32 = 32: 150 ns (在 20 MHz时)				
I/O 端口		I/O: 84 (5 V 容限/N 沟道开漏输出可选: 40)				
定时器		16 位定时/事件计数器 P: 6 通道 16 位定时/事件计数器 Q: 1 通道 16 位间隔定时器 M: 1 通道 钟表定时器: 1 通道 看门狗定时器: 1 通道				
实时输出端口		6 位×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				
串行接口		UART/CSI: 1通道 UART/ I ² C bus: 2通道 CSI: 3通道 CSI/ I ² C bus: 1通道				
DMA 控制		4 通道 (转移目标: 在片外围 I/O, 内部 RAM, 外部存储器)				
中断源		外部 9 (9) [※] , 内部 48				
节能功能		HALT/IDLE1/IDLE2/STOP/子时钟/sub-DLE 模式				
重置		RESET 引脚输入, 看门狗定时器2 (WDT2), 时钟监视器 (CLM), 低电压检测 (LVI)				
DCU		提供 (RUN/break)				
工作电压范围		2.85 ~ 3.6V				
工作温度范围		-40°C ~ 85°C				
封装		100引脚塑料LQFP (细间距) (14×14 mm)				
		100引脚塑料 QFP (14×20 mm)				

注 括号内数字表明可以释放 STOP 模式的外部中断数。

表 1-2. V850ES/JJ2 产品列表

产品名称		μ PD70F3720	μ PD70F3721	μ PD70F3722	μ PD70F3723	μ PD70F3724
内部存储器	Flash 存储器	128KB	256KB	384KB	512KB	640KB
	RAM	12KB	24KB	32KB	40KB	48KB
存储器容量	逻辑空间	64MB				
	外部存储区	16MB				
外部总线接口		地址总线: 24 位 数据总线: 8/16 位 多路总线模式/独立总线模式 片选信号: 4				
通用寄存器		32 位 \times 32 寄存器				
主时钟 (振荡频率)		陶瓷/水晶/外部时钟 (在 PLL 模式下: $f_x=2.5 \sim 5\text{MHz}$ (倍率为 4) 或者 $f_x=2.5\text{MHz}$ (倍率为 8), 在 clock through 模式下: $f_x=2.5 \sim 10\text{MHz}$)				
子时钟 (振荡频率)		水晶/外部时钟 ($f_{XT}=32.768\text{KHz}$)				
内部晶振		$f_R=200\text{KHz}$ (TYP)				
最小指令执行时间		50ns (主时钟 (f_{xx}) =20MHz)				
DSP 功能		32 \times 32 = 64: 200 ~ 250 ns (在20 MHz时) 32 \times 32 + 32 = 32: 300 ns (在 20 MHz时) 16 \times 16 = 32: 50 ~ 100 ns (在20 MHz时) 16 \times 16 + 32 = 32: 150 ns (在 20 MHz时)				
I/O 端口		I/O: 128 (5 V 容限/N 沟道开漏输出可选: 60)				
定时器		16 位定时/事件计数器 P: 9 通道 16 位定时/事件计数器 Q: 1 通道 16 位间隔定时器 M: 1 通道 钟表定时器: 1 通道 看门狗定时器: 1 通道				
实时输出端口		6 位 \times 2 通道				
A/D 转换器		10 位分辨率 \times 16 通道				
D/A 转换器		8 位分辨率 \times 2 通道				
串行接口		UART: 1通道 UART/CSI: 1通道 UART/I ² C bus: 2通道 CSI: 4通道 CSI/I ² C bus: 1通道				
DMA 控制		4 通道 (转移目标: 在片外围 I/O, 内部 RAM, 外部存储器)				
中断源		外部 10 (10) [※] , 内部 61				
节能功能		HALT/IDLE1/IDLE2/STOP/子时钟/sub-IDLE 模式				
复位		RESET \bar 引脚输入, 看门狗定时器 2 (WDT2), 时钟监视器 (CLM), 低电压检测 (LVI)				
DCU		提供 (RUN/break)				
工作电压范围		2.85 ~ 3.6V				
工作温度范围		-40 $^{\circ}\text{C}$ ~ 85 $^{\circ}\text{C}$				
封装		144引脚塑料LQFP (细间距) (20 \times 20mm)				

注 括号内数字表明可以释放 STOP 模式的外部中断数。

1.2 特性

- 最小指令执行时间: 50 ns (工作在 20 MHz 主时钟(fxx)下)
- 通用寄存器: 32 位 × 32 个
- CPU 特性:
 - 带符号乘法 (16 × 16 → 32): 1 至 2 时钟周期
 - 带符号乘法 (32 × 32 → 64): 1 至 5 时钟周期
 - 饱和操作 (包含上溢和下溢的检测功能)
 - 32 位移位指令: 1 时钟周期
 - 位操作指令
 - 区分长短格式的读取/存储指令
- 存储器空间: 为程序和数据提供 64 MB 的线性地址空间
 - 外部扩展: 多达 16 MB (包含 1 MB 用于内部 ROM/RAM)
 - 内置存储器:
 - RAM: 12/24/32/40/48 KB (见 表 1-1)
 - Flash 存储器: 128/256/384/512/640 KB (见 表 1-1)
 - 外部总线接口:
 - 可选择分离总线或多路总线输出
 - 可选择 8/16 位数据总线宽度
 - 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:
 - 不可屏蔽中断: 2 个
 - 可屏蔽中断: 55 个
 - 软件异常: 32 个
 - 异常陷阱: 2 个
- I/O 线: I/O 端口: 84
- 定时器功能:
 - 16 位间隔定时器 M (TMM): 1 通道
 - 16 位定时器/事件计数器 P (TMP): 6 通道
 - 16 位定时器/事件计数器 Q (TMQ): 1 通道
 - 钟表定时器: 1 通道
 - 看门狗定时器: 1 通道
- 实时输出端口: 6 位 × 1 通道
- 串行接口:
 - 异步串行接口 A (UARTA)
 - 3 线可变长串行接口 B (CSIB)
 - I²C 总线接口(I²C)
 - UARTA/CSIB: 1 通道
 - UARTA/I²C: 2 通道
 - CSIB/I²C: 1 通道
 - CSIB: 3 通道
- A/D 转换器: 10 位精度: 12 通道
- D/A 转换器: 8 位精度: 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:
 - 在主时钟或子时钟操作过程中
 - 可选择 7 个级别的 CPU 时钟 (fxx, fxx/2, fxx/4, fxx/8, fxx/16, fxx/32, fxt)
 - 时钟直连模式/PLL 模式可选

- 内置振荡时钟: 200 kHz (TYP.)
- 节能模式: HALT/IDLE1/IDLE2/STOP/子时钟/sub-IDLE 模式
- 封装: 100-pin 塑料 QFP (14 × 20) (仅限μPD70F3715, 70F3716, 70F3717)
100-pin 塑料 LQFP (细间距) (14 × 14)

1.3 应用领域

家用音响, 打印机, 数字家电和其他消费电子设备

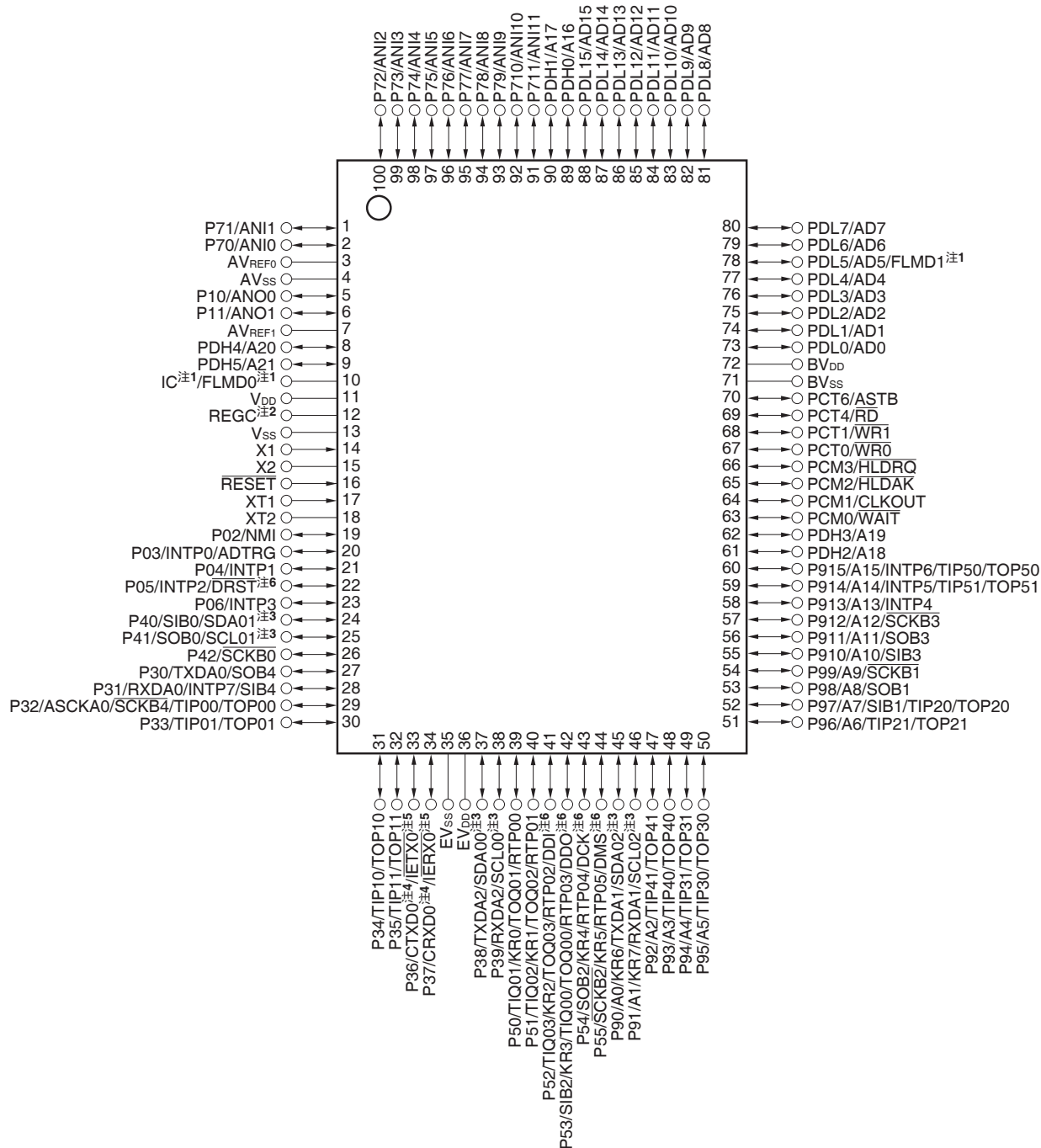
1.4 订货信息

产品代号	封装	内置 Flash ROM
μPD70F3715GFJBT-A	100-pin 塑料 QFP (14 × 20)	128 KB
μPD70F3715GC-8EA-A	100-pin 塑料 LQFP (细间距) (14 × 14)	128KB
μPD70F3716GF-JBT-A	100-pin 塑料 QFP (14 × 20)	256 KB
μPD70F3716GC -8EA-A	100-pin 塑料 LQFP (细间距) (14 × 14)	256 KB
μPD70F3717GF- JBT-A	100-pin 塑料 QFP (14 × 20)	384 KB
μPD70F3717GC -8EA-A	100-pin 塑料 LQFP (细间距) (14 × 14)	384 KB
μPD70F3718GC-8EA-A	100-pin 塑料 LQFP(细间距) (14 × 14)	512 KB
μPD70F3719GC- 8EA-A	100-pin 塑料 LQFP (细间距) (14 × 14)	640KB

备注 末尾带有-A 标号的是无铅产品。

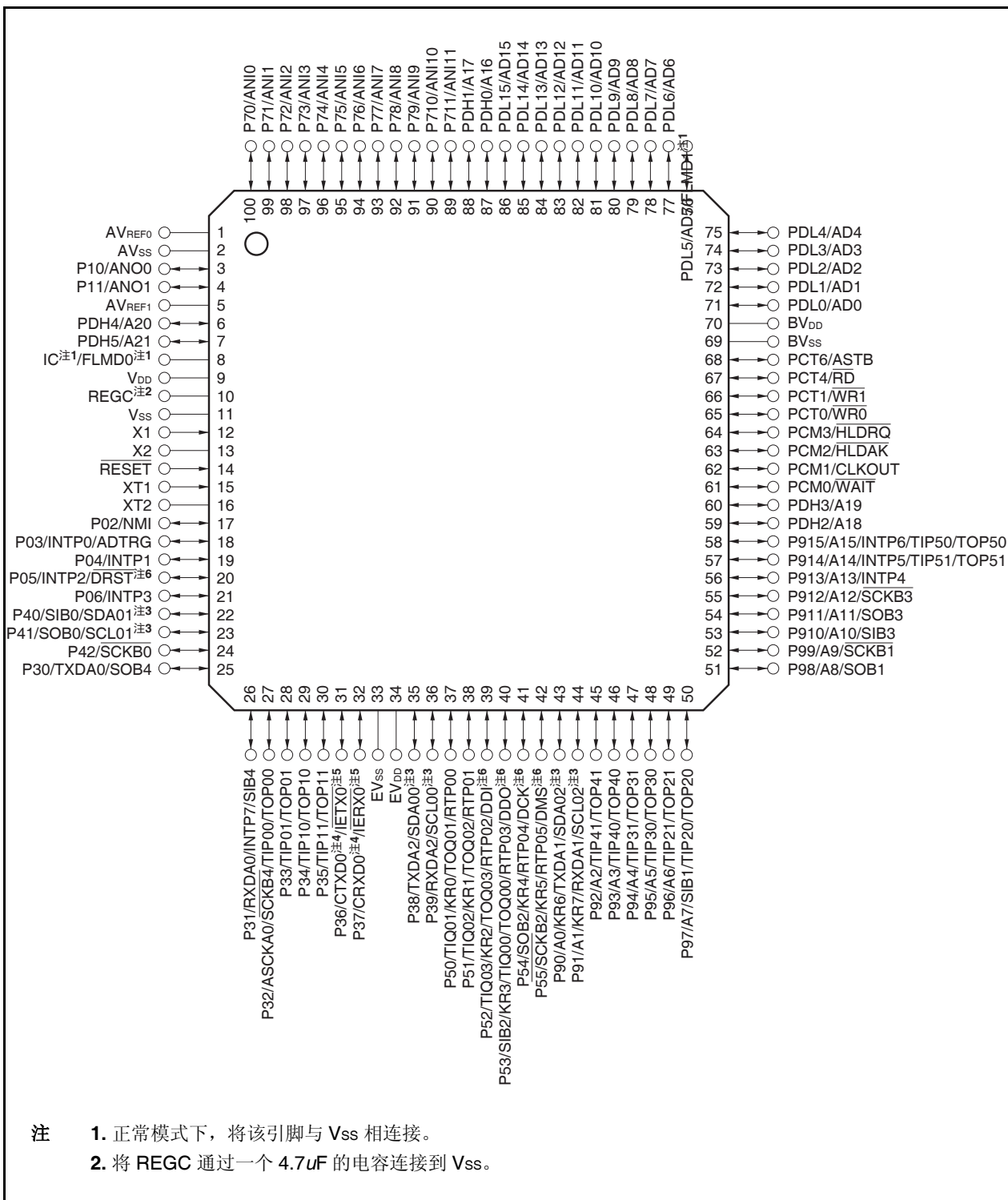
1.5 引脚图 (俯视图)

100 引脚塑料 QFP (14 × 20)

 μ PD70F3715GF -JBT-A μ PD70F3716GF -JBT-A μ PD70F3717GF-JBT-A^注

- 注
1. 正常模式下, 将该引脚与 V_{SS} 相连接。
 2. 将 REGC 通过一个 4.7 μ F 的电容器连接到 V_{SS}。

100 引脚塑料 LQFP (细间距) (14 × 14)

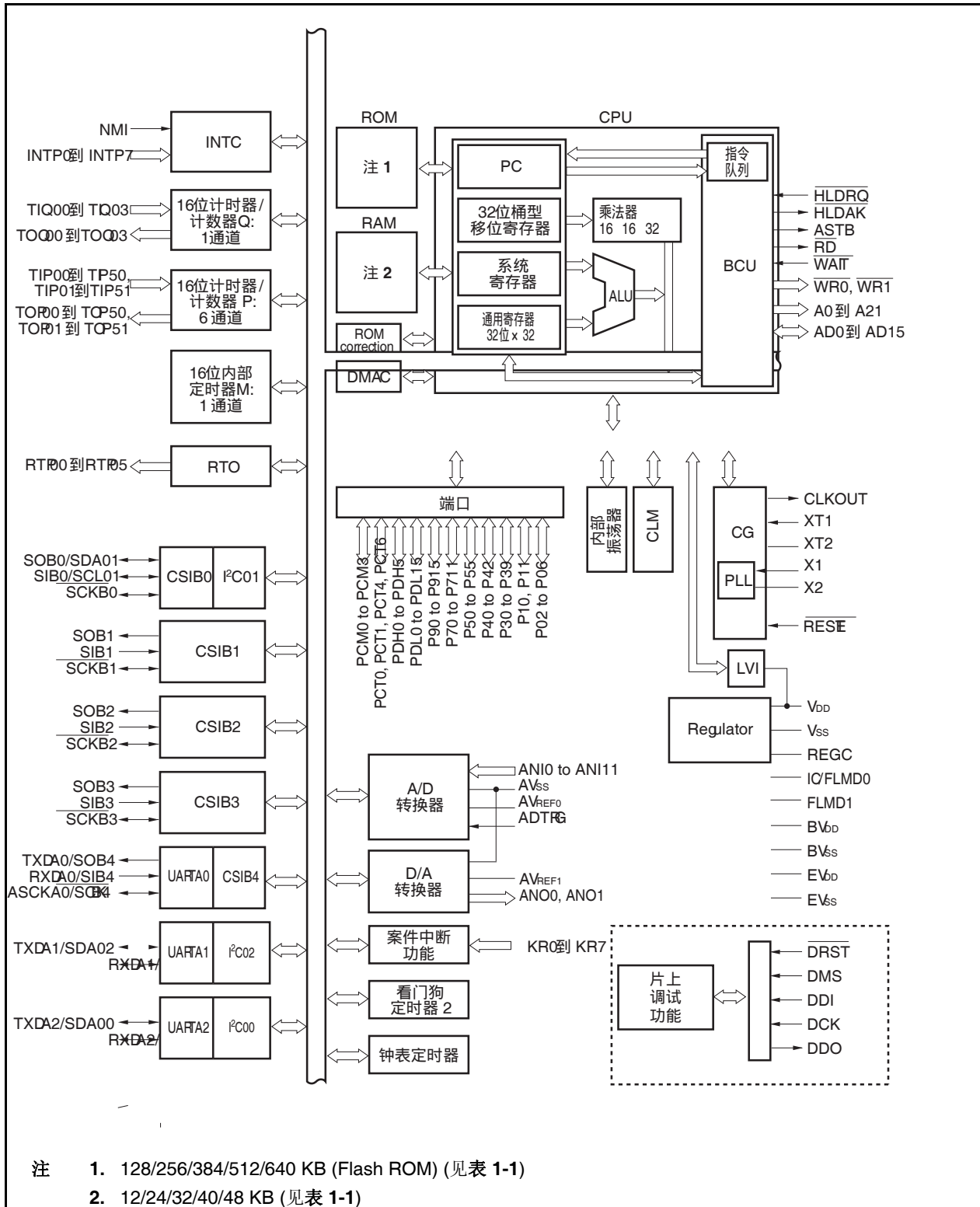
 μ PD70F3715GC-8EA-A μ PD70F3717GC-8EA-A μ PD70F3719GC-8EA-A μ PD70F3716GC-8EA-A μ PD70F3718GC-8EA-A

引脚名称

A0 到 A21:	地址总线	PDH0 到 PDH5:	端口 DH
AD0 到 AD15:	地址/数据总线	PDL0 到 PDL15:	端口 DL
ADTRG:	A/D 触发信号输入	$\overline{\text{RD}}$:	读取选通
ANI0 到 ANI11:	模拟信号输入	REGC:	整流器控制
ANO0, ANO1:	模拟信号输出	$\overline{\text{RESET}}$:	复位
ASCKA0:	异步串行时钟	RTP00 到 RTP05:	实时输出端口
ASTB:	地址选通	RXDA0 到 RXDA2:	接收数据
AVREF0, AVREF1:	模拟参考电压	SCKB0 到 SCKB4:	串行时钟
AVSS:	模拟 Vss	SCL00 到 SCL02:	串行时钟
BVDD:	总线接口的供电引脚	SDA00 到 SDA02:	串行数据
BVSS:	总线接口的地电平引脚	SIB0 到 SIB4:	串行输入
CLKOUT:	时钟输出	SOB0 到 SOB4:	串行输出
DCK:	调试时钟	TIP00, TIP01,	
DDI:	调试数据输入	TIP10, TIP11,	
DDO:	调试数据输出	TIP20, TIP21,	
DMS:	调试模式选择	TIP30, TIP31,	
$\overline{\text{DRST}}$:	调试复位	TIP40, TIP41,	
EVDD:	端口的供电引脚	TIP50, TIP51,	
$\overline{\text{EVSS}}$:	端口的地电平引脚	TIQ00 到 TIQ03:	定时器输入
FLMD0, FLMD1:	Flash 编程模式	TOP00, TOP01,	
$\overline{\text{HLDK}}$:	总线保持响应	TOP10, TOP11,	
$\overline{\text{HLDK}}$:	总线保持请求	TOP20, TOP21,	
INTP0 到 INTP7:	外部中断输入	TOP30, TOP31,	
KR0 到 KR7:	按键中断	TOP40, TOP41,	
NMI:	不可屏蔽中断请求	TOP50, TOP51,	
P02 到 P06:	端口 0	TOQ00 到 TOQ03:	定时器输出
P10, P11:	端口 1	TXDA0 到 TXDA2:	发送数据
P30 到 P39:	端口 3	VDD:	供电电源引脚
P40 到 P42:	端口 4	VSS:	地电平引脚
P50 到 P55:	端口 5	$\overline{\text{WAIT}}$:	等待
P70 到 P711:	端口 7	$\overline{\text{WR0}}$:	低字节写入选通
P90 到 P915:	端口 9	$\overline{\text{WR1}}$:	高字节写入选通
PCM0 到 PCM3:	端口 CM	X1, X2:	主时钟晶振
PCT0, PCT1,		XT1, XT2:	子时钟晶振
PCT4, PCT6:	端口 CT		

1.6 功能模块配置

1.6.1 内部结构框图



1.6.2 内部单元

(1) CPU

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

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

(2) 总线控制单元(BCU)

BCU 可以根据 CPU 所获取的物理地址值来启动外部总线周期。当产生从外部存储空间取指令的操作而 CPU 又没有发出启动总线周期的请求时，BCU 会生成一个预取指地址并进行指令代码的预取指操作。预取指的指令代码被存储在指令队列里。

(3) ROM

该产品提供 640/512/384/256/128 KB 的 Flash 存储器，映射地址为 0000000H 到 009FFFFH/0000000H 到 007FFFFH/0000000H 到 005FFFFH/0000000H 到 003FFFFH/0000000H 到 001FFFFH。

在取指令过程中，该存储器可由 CPU 在一个时钟周期内访问。

(4) RAM

该产品提供 48/40/32/24/12 KB 的 RAM，映射地址为 3FF3000H 到 3FFEFFFH/3FF5000H 到 3FFEFFFH/3FF7000H 到 3FFEFFFH/3FF9000H 到 3FFEFFFH/3FFC000H 到 3FFEFFFH。

在数据访问过程中，该 RAM 可由 CPU 在一个时钟周期内访问。

(5) 中断控制器 (INTC)

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

(6) 时钟发生器 (CG)

该产品提供一个主时钟振荡器和一个子时钟振荡器分别用于产生主时钟振荡频率(fx)和子时钟振荡频率(fxt)。共有两种模式可供选择：时钟直连模式，fx 被直接用作主时钟频率 (fxx)；PLL 模式，fx 在倍频 4 或 8 倍后被用作主时钟频率。

CPU 时钟频率 (fcpu) 可以从 fxx, fxx/2, fxx/4, fxx/8, fxx/16, fxx/32 和 fxt 频率中进行选择。

(7) 内部振荡器

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

(8) 定时器/计数器

片上集成了 6 通道 16 位定时器/事件计数器 P (TMP)，单通道 16 位定时器/事件计数器 Q (TMQ)和单通道 16 位间隔定时器 M (TMM)。

(9) 钟表定时器

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

(10) 看门狗定时器 2

V850ES/JG2 产品提供一个片上看门狗定时器，该定时器用于检测程序死锁，系统异常等错误状态。

看门狗定时器的时钟源可在内置振荡时钟、主时钟以及子时钟之间自由选择。

看门狗定时器可在产生溢出时生成一个不可屏蔽中断请求信号(INTWDT2) 或一个系统复位信号(WDT2RES)。

(11) 串行接口

V850ES/JG2 具有 3 种形式的串行接口: 异步串行接口 A (UARTA)，三线可变长串行接口 B (CSIB) 和一个 I²C 总线接口(I²C)。

使用 UARTA 时，数据通过 TXDA0 到 TXDA2 引脚以及 RXDA0 到 RXDA2 引脚传输。

使用 CSIB 时，数据通过 SOB0 到 SOB4 引脚，SIB0 到 SIB4 引脚以及 SCKB0 到 SCKB4 引脚传输。

使用 I²C 时，数据通过 SDA00 到 SDA02 以及 SCL00 到 SCL02 引脚传输。

(12) A/D 转换器

该 A/D 转换器是一个具有 12 个模拟输入引脚的 10 位 A/D 转换器，采用逐次比较法进行转换。

(13) D/A 转换器

V850ES/JG2 产品提供一个 2 通道 8 位精度的 D/A 转换器，该转换器采用 R-2R 梯形转换法进行转换。

(14) DMA 控制器

V850ES/JG2 产品提供一个 4 通道的 DMA 控制器。该控制器根据片上周边 I/O 器件发出的中断请求在内置 RAM 和片上周边 I/O 设备间传输数据。

(15) 按键中断功能

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

(16) 实时输出功能

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

(17) DCU 调试控制单元

V850ES/JG2 产品提供了符合 JTAG(Joint Test Action Group)通信标准的片上调试功能，通过控制引脚的输入电平和片上调试模式设置寄存器(OCDM)可以实现正常端口功能和片上调试功能间的切换。

(18) 端口

可使用以下通用端口功能和控制引脚功能。

端口	I/O	复用功能
P0	5 位 I/O	NMI、外部中断、A/D 转换器触发信号、调试复位
P1	2 位 I/O	D/A 转换器模拟输出
P3	10 位 I/O	外部中断、串行接口、定时器 I/O、CAN 数据 I/O、IEBus 数据 I/O
P4	3 位 I/O	串行接口
P5	6 位 I/O	定时器 I/O、实时输出、按键中断输入、串行接口，调试 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
PCT	4 位 I/O	外部控制信号
PDH	6 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

V850ES/JG2 的引脚名称和功能如下所示。

有四种类型的引脚 I/O 缓冲式供电电源： AV_{REF0} , AV_{REF1} , BV_{DD} , 和 EV_{DD} 。这些供电电源和引脚间的关系如下所示。

表 2-1. 引脚 I/O 缓冲式供电电源

供电电源	对应引脚
AV_{REF0}	端口 7
AV_{REF1}	端口 1
BV_{DD}	端口 CM, CT, DH (第 0 位~第 3 位), DL
EV_{DD}	\overline{RESET} , 端口 0, 3 ~ 5, 9, DH (第 4, 5 位)

(1) 端口引脚

(1/3)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
P02	19	17	I/O	端口 0 5 位 I/O 端口 可以位选输入/输出模式。 可以位选 N 沟道开漏输出。 5 V 耐压	NMI
P03	20	18			INTP0/ADTRG
P04	21	19			INTP1
P05 [※]	22	20			INTP2/ $\overline{\text{DRST}}$
P06	23	21			INTP3
P10	5	3	I/O	端口 1 2 位 I/O 端口 可以位选输入/输出模式。	ANO0
P11	6	4			ANO1
P30	27	25	I/O	端口 3 10 位 I/O 端口 可以位选输入/输出模式。 可以位选 N 沟道开漏输出。 5 V 耐压	TXDA0/SOB4
P31	28	26			RXDA0/INTP7/SIB4
P32	29	27			ASCKA0/ $\overline{\text{SCKB4}}$ /TIP00/TOP00
P33	30	28			TIP01/TOP01
P34	31	29			TIP10/TOP10
P35	32	30			TIP11/TOP11
P36	33	31			—
P37	34	32			—
P38	37	35			TXDA2/SDA00
P39	38	36			RXDA2/SCL00
P40	24	22	I/O	端口 4 3 位 I/O 端口 可以位选输入/输出模式。 可以位选 N 沟道开漏输出。 5 V 耐压	SIB0/SDA01
P41	25	23			SOB0/SCL01
P42	26	24			$\overline{\text{SCKB0}}$
P50	39	37	I/O	端口 5 6 位 I/O 端口 可以位选输入/输出模式。 可以位选 N 沟道开漏输出。 5 V 耐压	TIQ01/KR0/TOQ01/RTP00
P51	40	38			TIQ02/KR1/TOQ02/RTP01
P52	41	39			TIQ03/KR2/TOQ03/RTP02/DDI
P53	42	40			SIB2/KR3/TIQ00/TOQ00/RTP03/DDO
P54	43	41			SOB2/KR4/RTP04/DCK
P55	44	42			$\overline{\text{SCKB2}}$ /KR5/RTP05/DMS

注 连接一个下拉电阻。可通过设置 OCDM.OCDM0 为 0 断开连接。

备注 GF: 100 引脚塑料 QFP (14 × 20)
GC: 100 引脚塑料 LQFP (密间距) (14 × 14)

(2/3)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
P70	2	100	I/O	端口 7 12 位 I/O 端口 可以位选输入/输出模式。	ANI0
P71	1	99			ANI1
P72	100	98			ANI2
P73	99	97			ANI3
P74	98	96			ANI4
P75	97	95			ANI5
P76	96	94			ANI6
P77	95	93			ANI7
P78	94	92			ANI8
P79	93	91			ANI9
P710	92	90			ANI10
P711	91	89			ANI11
P90	45	43	I/O	端口 9 16 位 I/O 端口 可以位选输入/输出模式。 可以位选 N 沟道开漏输出。 5 V 耐压	A0/KR6/TXDA1/SDA02
P91	46	44			A1/KR7/RXDA1/SCL02
P92	47	45			A2/TIP41/TOP41
P93	48	46			A3/TIP40/TOP40
P94	49	47			A4/TIP31/TOP31
P95	50	48			A5/TIP30/TOP30
P96	51	49			A6/TIP21/TOP21
P97	52	50			A7/SIB1/TIP20/TOP20
P98	53	51			A8/SOB1
P99	54	52			A9/SCKB1
P910	55	53			A10/SIB3
P911	56	54			A11/SOB3
P912	57	55			A12/SCKB3
P913	58	56			A13/INTP4
P914	59	57			A14/INTP5/TIP51/TOP51
P915	60	58			A15/INTP6/TIP50/TOP50
PCM0	63	61	I/O	端口 CM 4 位 I/O 端口 可以位选输入/输出模式。	WAIT
PCM1	64	62			CLKOUT
PCM2	65	63			HLDK
PCM3	66	64			HLDRQ
PCT0	67	65	I/O	端口 CT 4 位 I/O 端口 可以位选输入/输出模式。	WR0
PCT1	68	66			WR1
PCT4	69	67			RD
PCT6	70	68			ASTB

备注 GF: 100 引脚塑料 QFP (14 × 20)
GC: 100 引脚塑料 LQFP (密间距) (14 × 14)

(3/3)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
PDH0	89	87	I/O	端口 DH 6 位 I/O 端口 可以位选输入/输出模式。	A16
PDH1	90	88			A17
PDH2	61	59			A18
PDH3	62	60			A19
PDH4	8	6			A20
PDH5	9	7			A21
PDL0	73	71	I/O	端口 DL 16 位 I/O 端口 可以位选输入/输出模式。	AD0
PDL1	74	72			AD1
PDL2	75	73			AD2
PDL3	76	74			AD3
PDL4	77	75			AD4
PDL5	78	76			AD5/FLMD1
PDL6	79	77			AD6
PDL7	80	78			AD7
PDL8	81	79			AD8
PDL9	82	80			AD9
PDL10	83	81			AD10
PDL11	84	82			AD11
PDL12	85	83			AD12
PDL13	86	84			AD13
PDL14	87	85			AD14
PDL15	88	86			AD15

备注 GF: 100 引脚塑料 QFP (14 × 20)
GC: 100 引脚塑料 LQFP (密间距) (14 × 14)

(2) 非端口引脚

(1/6)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
A0	45	43	输出	外部存储器地址总线 (当使用分立总线) 可以选择 N 沟道开漏输出。 5 V 耐压	P90/KR6/TXDA1/SDA02
A1	46	44			P91/KR7/RXDA1/SCL02
A2	47	45			P92/TIP41/TOP41
A3	48	46			P93/TIP40/TOP40
A4	49	47			P94/TIP31/TOP31
A5	50	48			P95/TIP30/TOP30
A6	51	49			P96/TIP21/TOP21
A7	52	50			P97/SIB1/TIP20/TOP20
A8	53	51			P98/SOB1
A9	54	52			P99/SCKB $\bar{1}$
A10	55	53			P910/SIB3
A11	56	54			P911/SOB3
A12	57	55			P912/SCKB $\bar{3}$
A13	58	56			P913/INTP4
A14	59	57			P914/INTP5/TIP51/TOP51
A15	60	58			P915/INTP6/TIP50/TOP50
A16	89	87	输出	外部存储器地址总线	PDH0
A17	90	88			PDH1
A18	61	59			PDH2
A19	62	60			PDH3
A20	8	6			PDH4
A21	9	7			PDH5
AD0	73	71	I/O	外部存储器地址总线/数据总线	PDL0
AD1	74	72			PDL1
AD2	75	73			PDL2
AD3	76	74			PDL3
AD4	77	75			PDL4
AD5	78	76			PDL5/FLMD1
AD6	79	77			PDL6
AD7	80	78			PDL7
AD8	81	79			PDL8
AD9	82	80			PDL9
AD10	83	81			PDL10
AD11	84	82			PDL11
AD12	85	83			PDL12
AD13	86	84			PDL13
AD14	87	85			PDL14
AD15	88	86			PDL15

备注 GF: 100 引脚塑料 QFP (14 × 20)
GC: 100 引脚塑料 LQFP (密间距) (14 × 14)

(2/6)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
ADTRG	20	18	输入	A/D 转换器外部触发输入。5 V 耐压	P03/INTP0
ANI0	2	100	输入	A/D 转换器的模拟电压输入	P70
ANI1	1	99			P71
ANI2	100	98			P72
ANI3	99	97			P73
ANI4	98	96			P74
ANI5	97	95			P75
ANI6	96	94			P76
ANI7	95	93			P77
ANI8	94	92			P78
ANI9	93	91			P79
ANI10	92	90			P710
ANI11	91	89			P711
ANO0	5	3	输出	D/A 转换器的模拟电压输出	P10
ANO1	6	4			P11
ASCKA0	29	27	输入	UARTA0 波特率时钟输入。5 V 耐压	P32/SCKB4/TIP00/TOP00
ASTB	70	68	输出	外部存储器的地址触发信号输出	PCT6
AV _{REF0}	3	1	—	A/D 转换器的参考电压输入/端口 7 的正电源提供	—
AV _{REF1}	7	5		D/A 转换器的参考电压输入/端口 1 的正电源提供	—
AV _{SS}	4	2	—	A/D 地电位和 D/A 转换器 (与 V _{SS} 电压相同)	—
BV _{DD}	72	70	—	总线接口和复用功能端口的供电引脚	—
BV _{SS}	71	69	—	总线接口和复用功能端口的地电平	—
CLKOUT	64	62	输出	内部系统时钟输出	PCM1
DCK	43	41	输入	调试时钟输入。5 V 耐压	P54/SOB2/KR4/RTP04
DDI	41	39	输入	调试数据输入。5 V 耐压	P52/TIQ03/KR2/TOQ03/RTP02
DDO [※]	42	40	输出	调试数据输出。可选 N 沟道开漏输出。 5 V 耐压	P53/SIB2/KR3/TIQ00/TOQ00/ RTP03
DMS	44	42	输入	调试模式选择输入。5 V 耐压	P55/SCKB2/KR5/RTP05
DRST	22	20	输入	调试复位输入。5 V 耐压	P05/INTP2
EV _{DD}	36	34	—	外部供电电源(与 V _{DD} 电压相同)	—
EV _{SS}	35	33	—	外部地电平(与 V _{SS} 电压相同)	—
FLMD0	10	8	输入	flash 存储器编程模式设置引脚	—
FLMD1	78	76			PDL5/AD5
HLD _{AK}	65	63	输出	总线保持应答输出	PCM2
HLD _{RQ}	66	64	输入	总线保持请求输入	PCM3

注 在片上调试模式状态下，强制设置为高电平输出。

备注 GF: 100 引脚塑料 QFP (14 × 20)
GC: 100 引脚塑料 LQFP (密间距) (14 × 14)

(3/6)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
INTP0	20	18	输入	外部中断请求输入 (可屏蔽, 模拟噪声消除电路)。引脚 INTP3 可选模拟噪声消除电路或数字噪声消除电路。 5 V 耐压	P03/ADTRG
INTP1	21	19			P04
INTP2	22	20			P05/ $\overline{\text{DRST}}$
INTP3	23	21			P06
INTP4	58	56			P913/A13
INTP5	59	57			P914/A14/TIP51/TOP51
INTP6	60	58			P915/A15/TIP50/TOP50
INTP7	28	26			P31/RXDA0/SIB4
KR0 ^{注1}	39	37	输入	按键中断输入 (片上模拟噪声消除电路)。 5 V 耐压	P50/TIQ01/TOQ01/RTP00
KR1 ^{注1}	40	38			P51/TIQ02/TOQ02/RTP01
KR2 ^{注1}	41	39			P52/TIQ03/TOQ03/RTP02/DDI
KR3 ^{注1}	42	40			P53/SIB2/TIQ00/TOQ00/ RTP03/DDO
KR4 ^{注1}	43	41			P54/SOB2/RTP04/DCK
KR5 ^{注1}	44	42			P55/ $\overline{\text{SCKB2}}$ /RTP05/DMS
KR6 ^{注1}	45	43			P90/A0/TXDA1/SDA02
KR7 ^{注1}	46	44			P91/A1/RXDA1/SCL02
NMI ^{注2}	19	17	输入	外部中断输入 (不可屏蔽, 模拟噪声消除电路)。5 V 耐压	P02
$\overline{\text{RD}}$	69	67	输出	外部存储器的读取触发信号输出	PCT4
REGC	12	10	—	调节器输出稳定电容的连接(4.7 μF)	—
$\overline{\text{RESET}}$	16	14	输入	系统复位输入	—
RTP00	39	37	输出	实时输出端口。 可选 N 沟道开漏输出。 5 V 耐压	P50/TIQ01/KR0/TOQ01
RTP01	40	38			P51/TIQ02/KR1/TOQ02
RTP02	41	39			P52/TIQ03/KR2/TOQ03/DDI
RTP03	42	40			P53/SIB2/KR3/TIQ00/TOQ00/ DDO
RTP04	43	41			P54/SOB2/KR4/DCK
RTP05	44	42			P55/ $\overline{\text{SCKB2}}$ /KR5/DMS
RXDA0	28	26	输入	串行接收数据输入 (UARTA0 ~ UARTA2) 5 V 耐压	P31/INTP7/SIB4
RXDA1	46	44			P91/A1/KR7/SCL02
RXDA2	38	36			P39/SCL00

- 注
1. 外部连接上拉电阻
 2. 引脚 NMI 和 P02 为复用功能引脚。复位后, 引脚功能为 P02。要使能引脚 NMI, 设置 PMC0.PMC02 为 1。引脚 NMI 的初始设置为“无边沿检测”。通过寄存器 INTF0 和 INTRO 选择引脚 NMI 的有效边沿。

备注

GF: 100 引脚塑料 QFP (14 × 20)

GC: 100 引脚塑料 LQFP (密间距) (14 × 14)

(4/6)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
SCKB0	26	24	I/O	串行时钟 I/O (CSIB0 ~ CSIB4) 可选 N 沟道开漏输出。 5 V 耐压	P42
SCKB1	54	52			P99/A9
SCKB2	44	42			P55/KR5/RTP05/DMS
SCKB3	57	55			P912/A12
SCKB4	29	27			P32/ASCKA0/TIP00/TOP00
SCL00	38	36	I/O	串行时钟 I/O (I ² C00 ~ I ² C02) 可选 N 沟道开漏输出。 5 V 耐压	P39/RXDA2
SCL01	25	23			P41/SOB0
SCL02	46	44			P91/A1/KR7/RXDA1
SDA00	37	35	I/O	串行发送/接收数据 I/O (I ² C00 ~ I ² C02) 可选 N 沟道开漏输出。 5 V 耐压	P38/TXDA2
SDA01	24	22			P40/SIB0
SDA02	45	43			P90/A0/KR6/TXDA1
SIB0	24	22	输入	串行接收数据输入 (CSIB0 ~ CSIB4) 5 V 耐压	P40/SDA01
SIB1	52	50			P97/A7/TIP20/TOP20
SIB2	42	40			P53/KR3/TIQ00/TOQ00/ RTP03/DDO
SIB3	55	53			P910/A10
SIB4	28	26			P31/RXDA0/INTP7
SOB0	25	23	输出	串行发送数据输出 (CSIB0 ~ CSIB4) 可选 N 沟道开漏输出。 5 V 耐压	P41/SCL01
SOB1	53	51			P98/A8
SOB2	43	41			P54/KR4/RTP04/DCK
SOB3	56	54			P911/A11
SOB4	27	25			P30/TXDA0

备注 GF: 100 引脚塑料 QFP (14 × 20)
GC: 100 引脚塑料 LQFP (密间距) (14 × 14)

(5/6)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
TIP00	29	27	输入	外部事件计数输入/捕捉触发输入/外部触发输入 (TMP0)。 5 V 耐压	P32/ASCKA0/SCKB4/TOP00
TIP01	30	28		捕捉触发输入 (TMP0)。5 V 耐压	P33/TOP01
TIP10	31	29		外部事件计数输入/捕捉触发输入/外部触发输入 (TMP1)。 5 V 耐压	P34/TOP10
TIP11	32	30		捕捉触发输入 (TMP1)。5 V 耐压	P35/TOP11
TIP20	52	50		外部事件计数输入/捕捉触发输入/外部触发输入 (TMP2)。 5 V 耐压	P97/A7/SIB1/TOP20
TIP21	51	49		捕捉触发输入 (TMP2)。5 V 耐压	P96/A6/TOP21
TIP30	50	48		外部事件计数输入/捕捉触发输入/外部触发输入 (TMP3)。 5 V 耐压	P95/A5/TOP30
TIP31	49	47		捕捉触发输入 (TMP3)。5 V 耐压	P94/A4/TOP31
TIP40	48	46		外部事件计数输入/捕捉触发输入/外部触发输入 (TMP4)。 5 V 耐压	P93/A3/TOP40
TIP41	47	45		捕捉触发输入 (TMP4)。5 V 耐压	P92/A2/TOP41
TIP50	60	58		外部事件计数输入/捕捉触发输入/外部触发输入 (TMP5)。 5 V 耐压	P915/A15/INTP6/TOP50
TIP51	59	57		捕捉触发输入 (TMP5)。5 V 耐压	P914/A14/INTP5/TOP51
TIQ00	42	40		外部事件计数输入/捕捉触发输入/外部触发输入 (TMQ0)。 5 V 耐压	P53/SIB2/KR3/TOQ00/RTP03/ DDO
TIQ01	39	37		捕捉触发输入 (TMQ0)。5 V 耐压	P50/KR0/TOQ01/RTP00
TIQ02	40	38			P51/KR1/TOQ02/RTP01
TIQ03	41	39			P52/KR2/TOQ03/RTP02/ DDI

备注 GF: 100 引脚塑料 QFP (14 × 20)
GC: 100 引脚塑料 LQFP (密间距) (14 × 14)

(6/6)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
TOP00	29	27	输出	定时器输出 (TMP0) 可选 N 沟道开漏输出。5 V 耐压	P32/ASCKA0/SCKB4/TIP00
TOP01	30	28			P33/TIP01
TOP10	31	29		定时器输出 (TMP1) 可选 N 沟道开漏输出。5 V 耐压	P34/TIP10
TOP11	32	30			P35/TIP11
TOP20	52	50		定时器输出 (TMP2) 可选 N 沟道开漏输出。5 V 耐压	P97/A7/SIB1/TIP20
TOP21	51	49			P96/A6/TIP21
TOP30	50	48		定时器输出 (TMP3) 可选 N 沟道开漏输出。5 V 耐压	P95/A5/TIP30
TOP31	49	47			P94/A4/TIP31
TOP40	48	46		定时器输出 (TMP4) 可选 N 沟道开漏输出。5 V 耐压	P93/A3/TIP40
TOP41	47	45			P92/A2/TIP41
TOP50	60	58		定时器输出 (TMP5) 可选 N 沟道开漏输出。5 V 耐压	P915/A15/INTP6/TIP50
TOP51	59	57			P914/A14/INTP5/TIP51
TOQ00	42	40	输出	定时器输出 (TMQ0) 可选 N 沟道开漏输出。5 V 耐压	P53/SIB2/KR3/TIQ00/RTP03/ DDO
TOQ01	39	37			P50/TIQ01/KR0/RTP00
TOQ02	40	38			P51/TIQ02/KR1/RTP01
TOQ03	41	39			P52/TIQ03/KR2/RTP02/DDI
TXDA0	27	25	输出	串行发送数据输出 (UARTA0 ~ UARTA2) 可选 N 沟道开漏输出。 5 V 耐压	P30/SOB4
TXDA1	45	43			P90/A0/KR6/SDA02
TXDA2	37	35			P38/SDA00
V _{DD}	11	9	—	内部供电电源引脚	—
V _{SS}	13	11	—	内部地电平	—
WAIT	63	61	输入	外部等待输入	PCM0
WR0	67	65	输出	外部存储器写触发(低 8 位)	PCT0
WR1	68	66		外部存储器写触发(高 8 位)	PCT1
X1	14	12	输入	连接主时钟振荡器	—
X2	15	13	—		—
XT1	17	15	输入	连接子时钟振荡器	—
XT2	18	16	—		—

备注 GF: 100 引脚塑料 QFP (14 × 20)
GC: 100 引脚塑料 LQFP (密间距) (14 × 14)

2.2 引脚状态

如下为各种模式下的引脚操作状态。

表 2-2. 各种模式下的引脚操作状态

引脚名称	上电后 ^{註 1}	复位期间(除去上电时)	HALT 模式 ^{註 2}	IDLE1, IDLE2, Sub-IDLE 模式 ^{註 2}	STOP 模式 ^{註 2}	Idle 模式 ^{註 3}	总线保持
P05/ $\overline{\text{DRST}}$	置低	置低 ^{註 4}	保持	保持	保持	保持	保持
P10/ ANO0 , P11/ ANO1	不确定	Hi-Z	保持	保持	Hi-Z	保持	保持
P53/DDO		Hi-Z ^{註 5}	保持	保持	保持	保持	保持
AD0 ~ AD15	Hi-Z ^{註 6}	Hi-Z ^{註 6}	註 7, 8	Hi-Z	Hi-Z	保持	Hi-Z
A0 ~ A15			不确定 ^{註 7, 9}				
A16 ~ A21			不确定 ^{註 7}				
$\overline{\text{WAIT}}$			—	—	—	—	—
CLKOUT			操作	L	L	操作	操作
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$			H ^{註 7}	H	H	H	Hi-Z
$\overline{\text{RD}}$							
ASTB							
$\overline{\text{HLDK}}$							
$\overline{\text{HLDRQ}}$			操作 ^{註 7}				L
					—	—	—
其他端口引脚	Hi-Z	Hi-Z	保持	保持	保持	保持	保持

- 注
1. 上电后，当电压达到操作所需供电电压范围(下限)后持续 1 ms 的时间。
 2. 使用复用功能时的操作。
 3. 分立总线模式下，在显示 T2 状态后空闲状态下的引脚状态被插入。复用总线模式下，在显示 T3 状态后空闲状态下的引脚状态被插入。
 4. 外部复位期间置低。由看门狗定时器，时钟监测器，等引发的内部复位期间，引脚状态根据 OCDM.OCDM0 位的设置不同而不同。
 5. DDO 输出定义于片上调试模式下。
 6. 总线控制引脚功能复用为端口引脚，所以被初始化为输入模式(普通端口模式)。
 7. HALT 模式下继续操作，在 DMA 操作期间。
 8. 分立总线模式: Hi-Z
复用总线模式: 不确定
 9. 分立总线模式

备注

Hi-Z: 高阻态

保持: 在之前的外部总线周期期间状态保持。

L: 低电平输出

H: 高电平输出

—: 无采样输入 (无应答)

2.3 I/O 引脚电路类型，I/O 缓冲供电电平，和不使用引脚的连接

(1/3)

引脚	复用功能	引脚编号		I/O 电路类型	推荐连接
		GF	GC		
P02	NMI	19	17	10-D	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 输出： 保持开路。
P03	INTP0/ADTRG	20	18		
P04	INTP1	21	19		
P05	INTP2/ $\overline{\text{DRST}}$	22	20	10-N	输入： 通过一个电阻单独连接到 EV _{SS} 。禁止固定到 V _{DD} 电平。 输出： 保持开路。 RESET 引脚复位后内部置低。
P06	INTP3	23	21	10-D	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 输出： 保持开路。
P10, P11	ANO0, ANO1	5, 6	3, 4	12-D	输入： 通过一个电阻单独连接到 AV _{REF1} 或 AV _{SS} 。 输出： 保持开路。
P30	TXDA0/SOB4	27	25	10-G	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 输出： 保持开路。
P31	RXDA0/INTP7/SIB4	28	26	10-D	
P32	ASCKA0/ $\overline{\text{SCKB4}}$ /TIP00	29	27		
P33	TIP01/TOP01	30	28		
P34	TIP10/TOP10	31	29		
P35	TIP11/TOP11	32	30		
P36	—	33	31	10-G	
P37	—	34	32		
P38	TXDA2/SDA00	37	35	10-D	
P39	RXDA2/SCL00	38	36		
P40	SIB0/SDA01	24	22		
P41	SOB0/SCL01	25	23		
P42	$\overline{\text{SCKB0}}$	26	24		
P50	TIQ01/KR0/TOQ01/RTP00	39	37		
P51	TIQ02/KR1/TOQ02/RTP01	40	38		
P52	TIQ03/KR2/TOQ03/RTP02/DDI	41	39		
P53	SIB2/KR3/TIQ00/TOQ00/RTP03/DDO	42	40		
P54	SOB2/KR4/RTP04/DCK	43	41	10-D	
P55	$\overline{\text{SCKB2}}$ /KR5/RTP05/DMS	44	42		
P70 ~ P711	ANI0 ~ ANI11	2, 1, 100-91	100-89	11-G	输入： 通过一个电阻单独连接到 AV _{REF0} 或 AV _{SS} 。 输出： 保持开路。

备注 GF: 100 引脚塑料 QFP (14 × 20)
GC: 100 引脚塑料 LQFP (密间距) (14 × 14)

(2/3)

引脚	复用功能	引脚编号		I/O 电路类型	推荐连接
		GF	GC		
P90	A0/KR6/TXDA1/SDA02	45	43	10-D	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 输出： 保持开路。
P91	A1/KR7/RXDA1/SCL02	46	44		
P92	A2/TIP41/TOP41	47	45		
P93	A3/TIP40/TOP40	48	46		
P94	A4/TIP31/TOP31	49	47		
P95	A5/TIP30/TOP30	50	48		
P96	A6/TIP21/TOP21	51	49		
P97	A7/SIB1/TIP20/TOP20	52	50		
P98	A8/SOB1	53	51	10-G	
P99	A9/SCKB1	54	52	10-D	
P910	A10/SIB3	55	53	10-D	
P911	A11/SOB3	56	54		
P912	A12/SCKB3	57	55		
P913	A13/INTP4	58	56		
P914	A14/INTP5/TIP51/TOP51	59	57		
P915	A15/INTP6/TIP50/TOP50	60	58		
PCM0	WAIT	63	61	5	输入： 通过一个电阻单独连接到 BV _{DD} 或 BV _{SS} 。 输出： 保持开路。
PCM1	CLKOUT	64	62		
PCM2	HLDAK	65	63		
PCM3	HLDRQ	66	64		
PCT0, PCT1	WR0, WR1	67, 68	65, 66		
PCT4	RD	69	67		
PCT6	ASTB	70	68		
PDH0 ~ PDH3	A16 ~ A19	89, 90 61, 62	87, 88 59, 60		
PDH4, PDH5	A20, A21	8, 9	6, 7		
PDL0 ~ PDL4	AD0 ~ AD4	73-77	71-75		输入： 通过一个电阻单独连接到 BV _{DD} 或 BV _{SS} 。 输出： 保持开路。
PDL5	AD5/FLMD1	78	76		
PDL6 ~ PDL15	AD6 ~ AD15	79-88	77-86		

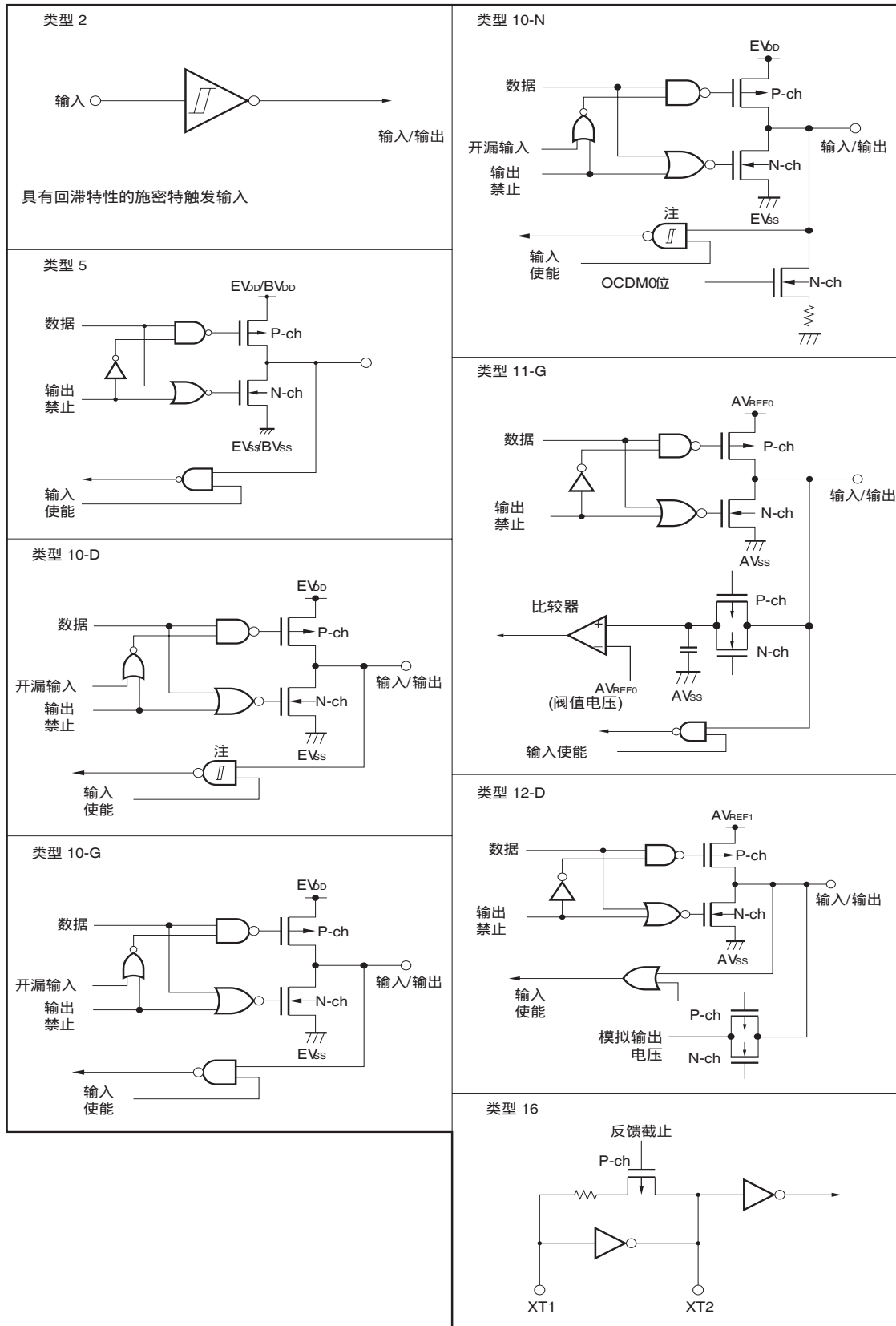
备注 GF: 100 引脚塑料 QFP (14 × 20)
GC: 100 引脚塑料 LQFP (密间距) (14 × 14)

(3/3)

引脚	复用功能	引脚编号		I/O 电路类型	推荐连接
		GF	GC		
AV _{REF0}	—	3	1	—	直接连接到 V _{DD} 并持续提供电平。
AV _{REF1}	—	7	5	—	直接连接到 V _{DD} 并持续提供电平。
AV _{SS}	—	4	2	—	直接连接到 V _{SS} 并持续提供电平。
BV _{DD}	—	72	70	—	直接连接到 V _{DD} 并持续提供电平。
BV _{SS}	—	71	69	—	直接连接到 V _{SS} 并持续提供电平。
EV _{DD}	—	36	34	—	—
EV _{SS}	—	35	33	—	—
FLMD0	—	10	8	—	在非 flash 存储器编程模式下, 直接连接到 V _{SS} 。
REGC	—	12	10	—	连接调整器输出稳定电容。(4.7 μ F (初步值))。
RESET	—	16	14	2	—
V _{DD}	—	11	9	—	—
V _{SS}	—	13	11	—	—
X1	—	14	12	—	—
X2	—	15	13	—	—
XT1	—	17	15	16	连接到 V _{SS} 。
XT2	—	18	16	16	保持开路。

备注 GF: 100 引脚塑料 QFP (14 × 20)
GC: 100 引脚塑料 LQFP (密间距) (14 × 14)

图 2-1. I/O 引脚电路



注 普通端口模式不产生滞后现象。

2.4 注意事项

上电后，下述引脚可能在复位期间输出不确定电平。

- P10/ANO0 引脚
- P11/ANO1 引脚
- P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 引脚

第三章 CPU 功能

V850ES/JG2 系列产品的 CPU 是基于 RISC 结构设计的，具备 5 级流水线控制，几乎所有指令均可在一个时钟周期内完成。

3.1 特点

- 最小指令执行时间: 50 ns (工作于 20MHz 时钟时)
30.5 ns (工作于子时钟(f_{XT}) = 32.768 kHz 时)
- 存储器空间 程序(物理地址)空间: 64 MB 线性
 数据(逻辑地址)空间: 4 GB 线性
- 通用寄存器: 32 位 × 32 个
- 内部 32 位结构
- 5 级流水线控制
- 乘/除法指令
- 饱和运算指令
- 32 位移位指令: 单时钟
- 长/短格式的载入/存储 (load/store)指令
- 4 种位操作指令
 - SET1
 - CLR1
 - NOT1
 - TST1

3.2 CPU 寄存器组

CPU 寄存器可分为两组: 通用程序寄存器组和专用系统寄存器组。所有的寄存器的宽度均为 32 位。
详情请参考 **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	(汇编器保留寄存器)	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基指针)
r18			
r19			
r20			
r21			
r22			
r23			
r24			
r25			
r26			
r27			
r28			
r29			
r30	(元素指针 (EP))		
r31	(链接指针 (LP))		
31	0		
PC	(程序计数器)		

3.2.1 程序寄存器组

程序寄存器组包含通用寄存器和程序计数器。

(1) 通用寄存器 (r0 to r31)

共有 32 个通用寄存器(r0 到 r31)可供选择。这些寄存器均可用于存放数据变量或地址变量。
但是，r0 和 r30 是被指令隐含使用的，因此要慎用这两个寄存器。寄存器 r0 始终保持 0 值，用于使用了 0 值的操作和偏移量为 0 的寻址操作。寄存器 r30 被指令 SLD 和 SST 所使用，在访问存储器时被用作基指针。r1, r3 到 r5 以及 r31 被汇编器和 C 编译器隐含使用。因此，在使用这些寄存器之前要对它们的内容进行保存，以免发生数据丢失。使用之后，要对寄存器的值进行恢复。寄存器 r2 在一些情况下会被实时操作系统所使用。当它没有被实时操作系统使用时可用来存放数据或地址变量。

表 3-1. 程序寄存器

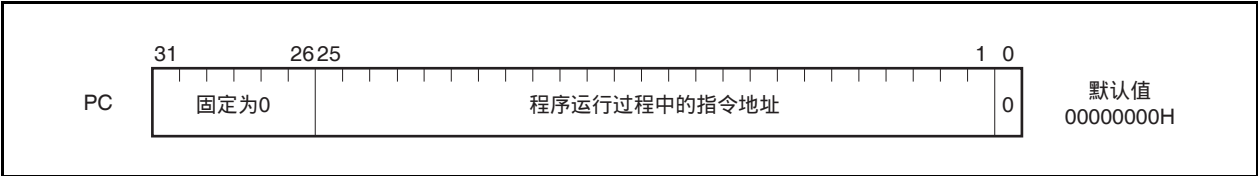
名称	用途	操作
r0	零寄存器	始终保持为零
r1	汇编器保留寄存器	用于生成 32 位立即数据的工作寄存器
r2	地址/数据寄存器(不被实时操作系统所使用时)	
r3	堆栈指针	当函数被调用时用于生成堆栈结构
r4	全局指针	用于访问数据区域内的全局变量
r5	文本指针	用于表示文本区域(放置程序程序代码的区域)的开始部分的寄存器
r6 到 r29	地址/数据变量寄存器	
r30	元素指针	访问存储器时用于生成地址的基指针
r31	连接指针	用于编译器调用函数时
PC	程序计数器	程序执行时存放指令地址

备注 汇编器和 C 编译器所使用的 r1, r3 到 r5, 以及 r31 的详细情况，请参考 CA850 (C 编译器包) 汇编语言用户手册。

(2) 程序计数器 (PC)

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

第 0 位固定为 0，因此，向奇地址的跳转将不会被执行。



3.2.2 系统寄存器组

系统寄存器控制 CPU 的状态并且保存中断信息。

使用载入/存储(load/store)指令(LDSR 或 STSR)可以对这些系统寄存器进行读/写操作，系统寄存器的编号如下表所示。

表 3-2. 系统寄存器编号

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

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

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

备注

√: 允许访问

×: 禁止访问

(1) 中断状态保存寄存器 (EIPC 和 EIPSW)

EIPC 和 EIPSW 用来在可屏蔽中断发生时保存当前的运行状态。

当软件异常或可屏蔽中断发生时，程序指针 PC 的值被保存在 EIPC 中，程序状态字 PSW 的值被保存在 EIPSW 中；当不可屏蔽中断发生时，PC 和 PSW 的值则会被分别保存在 NMI(不可屏蔽中断)状态保存寄存器 FEPC 和 FEPSW 中。

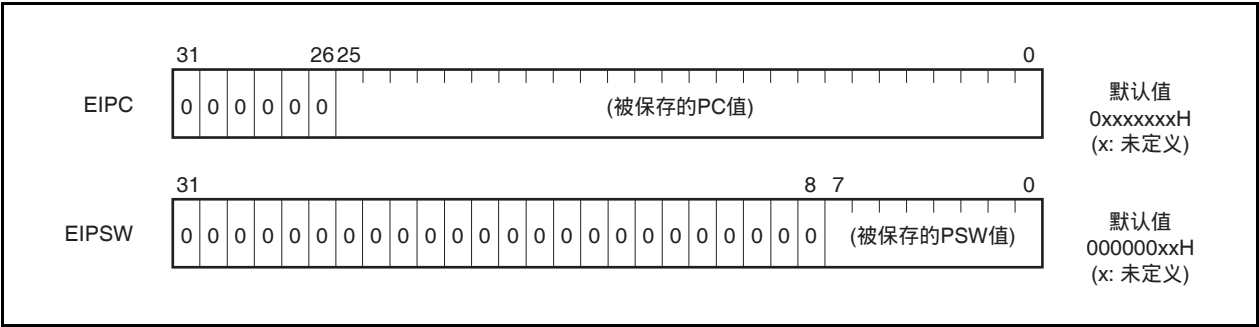
软件异常和可屏蔽中断发生时，当前指令的下一条指令的地址会被保存到 EIPC 中(一些指令除外，详情请参阅 19.8 CPU 不对中断进行确认的时期)。

当前的 PSW 值被保存在 EIPSW 中。

由于中断状态保存寄存器只能对一组中断状态进行保存，因此当多重中断发生时，系统寄存器的内容需要由程序自行保存。

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

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



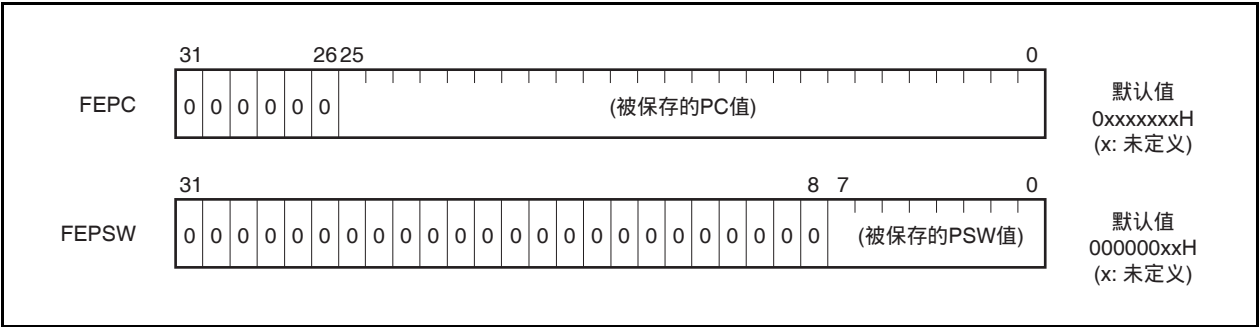
FEPC 和 FEPSW 用来在不可屏蔽中断(NMI)发生时

当不可屏蔽中断发生时，PC 的值被保存在 FEPC 中，程序状态字 PSW 的值被保存在 FEPSW 中。

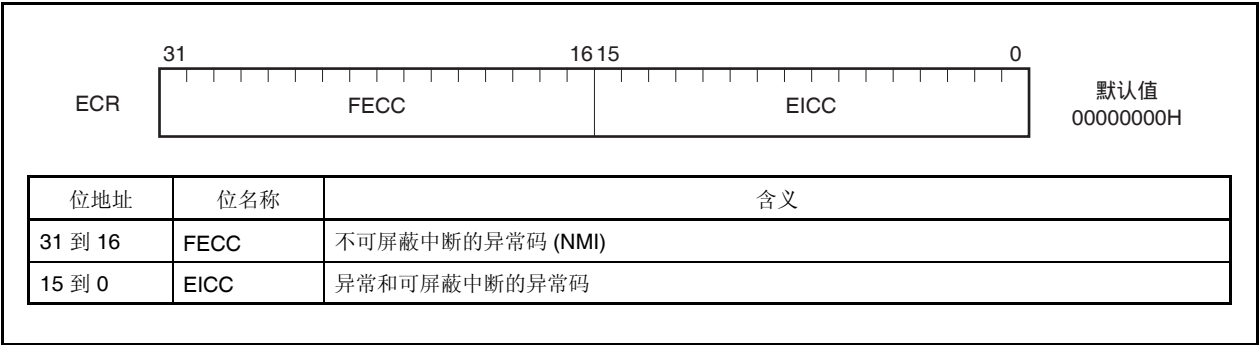
当前的 PSW 值被保存在 FEPSW 中。

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

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



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



(4) 程序状态字 (PSW)

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

如果使用 LDSR 指令更改其中一位标志, 那么新的状态将在 LDSR 指令执行后立即生效。

如果 ID 标志被置 1, 当执行 LDSR 指令对 PSW 进行操作时, 中断请求响应始终保持无响应状态。

寄存器的第 8 到 31 位为扩展功能保留(始终为 0)。

(1/2)



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

备注 请阅读下页中的注。

(2/2)

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

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

(5) CALLT 执行状态保存寄存器 (CTPC 和 CTPSW)

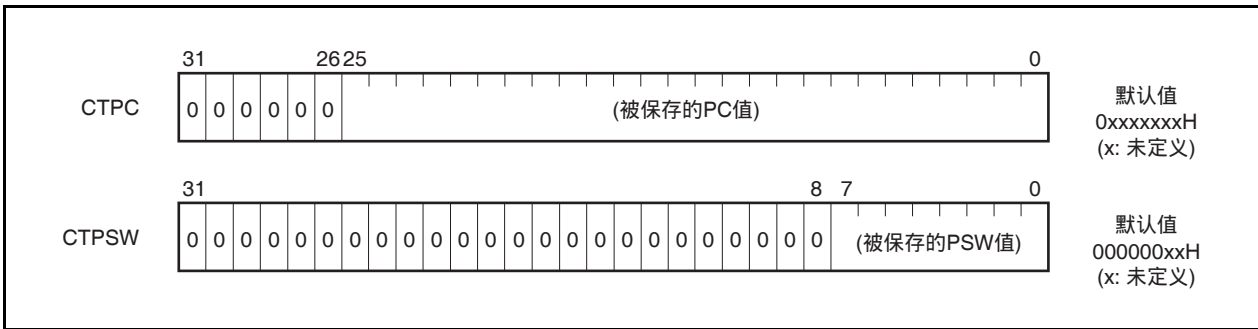
CTPC 和 CTPSW 是 CALLT 执行状态保存寄存器。

当 CALLT 指令执行时，PC 值被保存在 CTPC 中，PSW 值被保存在 CTPSW 中。

被保存在 CTPC 中的值是 CALLT 下一条指令的地址。

保存在 CTPSW 中的值是当前的 PSW 值。

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



(6) 异常/调试陷阱状态保存寄存器 (DBPC 和 DBPSW)

DBPC 和 DBPSW 是异常/调试陷阱状态保存寄存器。

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

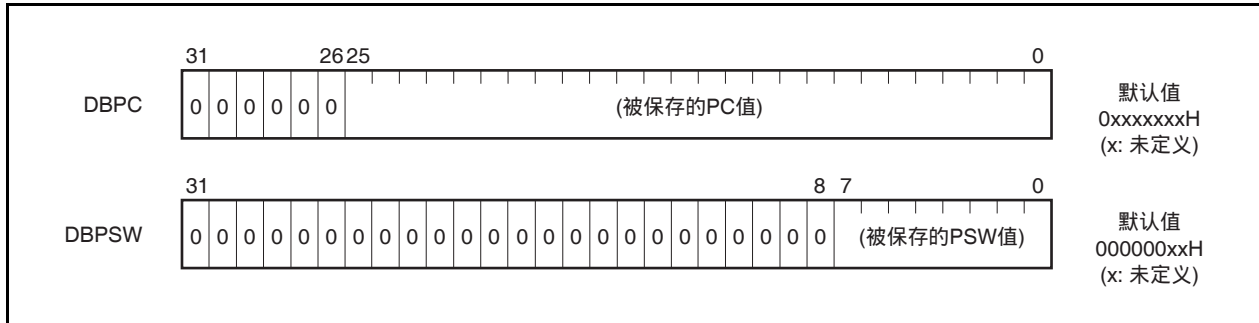
被保存在 DBPC 中的值是异常陷阱或调试陷阱发生时正在执行的指令的下一条指令的地址。

保存在 DBPSW 中的值是当前的 PSW 值。

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

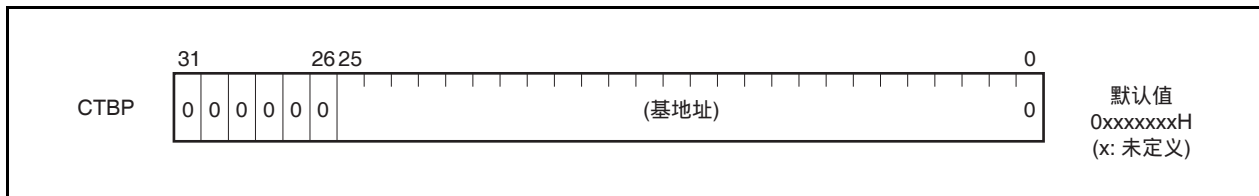
DBPC 的第 26 到 31 位和 DBPSW 的第 8 到 31 位为扩展功能保留(始终为 0)。

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

**(7) CALLT 基址指针 (CTBP)**

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

第 26 到 31 位为扩展功能保留(始终为 0)。



3.3 操作模式

V850ES/JG2 具有以下几种操作模式。

(1) 正常操作模式

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

(2) 闪存编程模式

在这个模式下，内部闪存可以用闪存烧写器进行烧写。
以下产品是 V850ES/JG2 中带有闪存的版本。

(3) 片上调试(On-chip debug)模式

V850ES/JG2 采用专用的通信接口规范来支持 JTAG(Joint Test Action Group)格式的片上调试功能
更多细节详见第 27 章 片上调试功能。

3.3.1 指定操作模式

使用引脚 FLMD0 和 FLMD1 可对器件进行操作模式的指定。

正常模式下，要确保在系统复位释放后有低电平信号输入 FLMD0 引脚。

闪存编程模式下，如果连接有闪存编程器，编程器应该向 FLMD0 脚输入高电平信号，但是在自编程模式下这个信号必须通过一个外部电路输入。

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

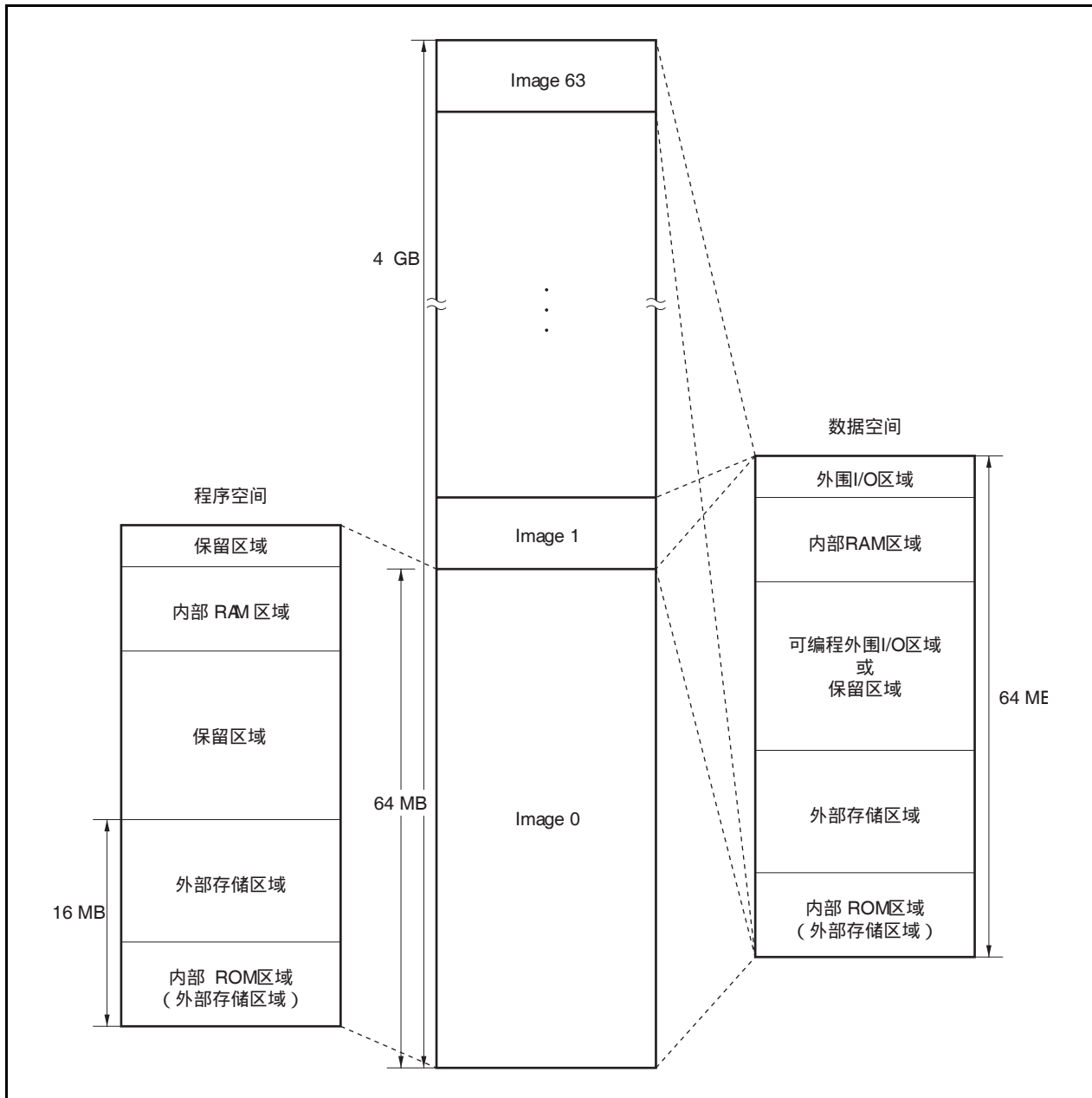
备注 L: 低电平输入
 H: 高电平输入
 ×: 不关心

3.4 地址空间

3.4.1 CPU 地址空间

对于指令地址，最大支持 64MB 的线性地址空间(程序空间)，其中包括外部存储空间和内部 ROM 的总共 16MB 存储空间和内部 RAM 空间。对于操作数地址（数据存取），最大支持 4GB 的线形地址空间。这个 4GB 的线性地址空间可视为 64 个 64MB 的物理地址空间镜像。这意味着无论地址位的第 26 到 31 位为何值，系统总是在同样的 64MB 的物理地址空间中进行寻址。

图 3-1. 地址空间图

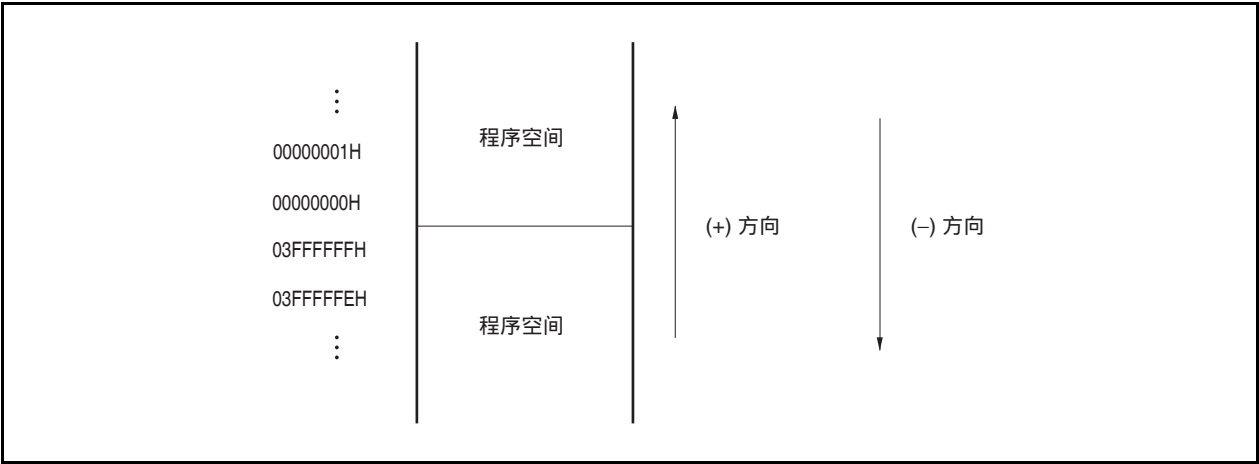


3.4.2 CPU 地址空间的绕回

(1) 程序空间

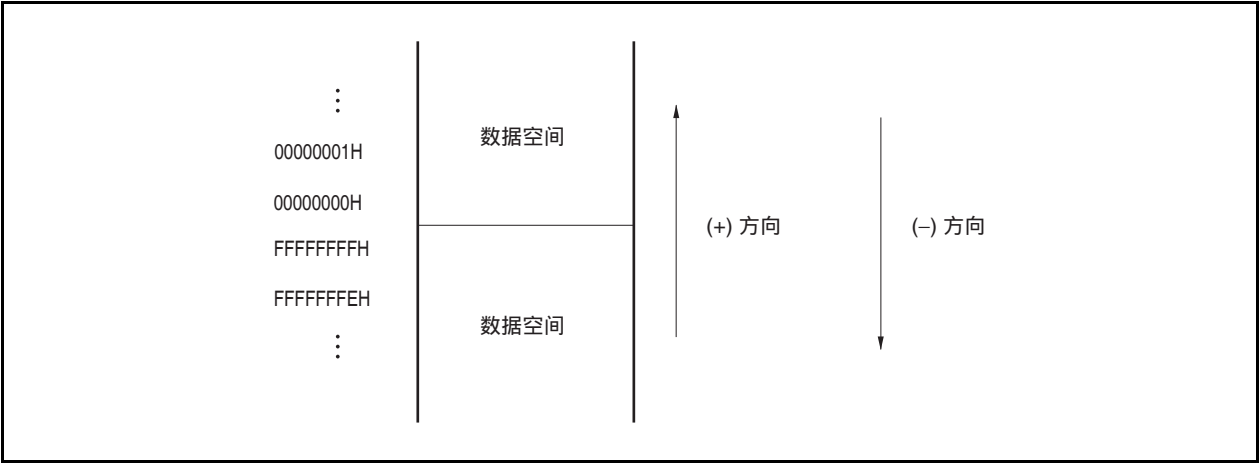
PC(程序计数器)的 32 位中，高 6 位固定为 0，只有低 26 位有效。即使分支地址计算过程中产生第 25 到 26 位的进位或借位，也会被高 6 位忽略。
因此程序空间的地址下限(地址 00000000H)和地址上限(地址 03FFFFFFH)成为连续的地址。所谓绕回，指的就是这种地址下限和地址上限连续的情况。

注意事项 由于从地址 03FF0000H 到 03FFFFFFH 的 4KB 范围是片上外围 I/O 设备区，这个区域不能进行取指令操作。因此，要避免执行那些分支地址计算结果可能会影响此 4KB 区域的操作。



(2) 数据空间

结果超过 32 位的操作数地址计算操作将被忽略。
因此，数据空间的最低地址 0000000H 和最高地址 FFFFFFFFH 之间是相连的，在这两个地址边界附近会发生绕回现象。



3.4.3 存储器映射

V850ES/JG2 存储器的保留区域如下所示。

图 3-2. 数据存储器映射 (物理地址)

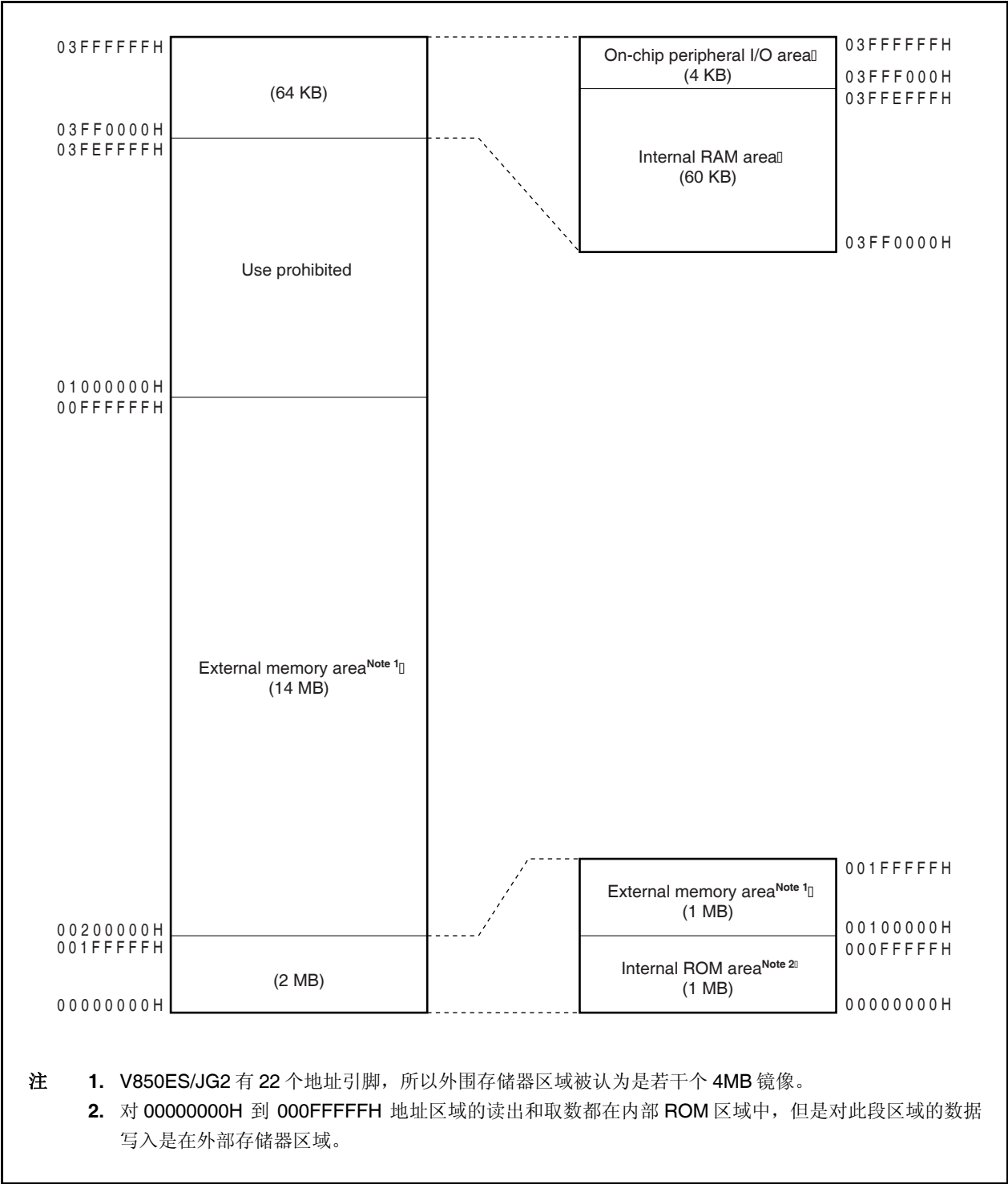
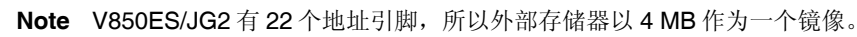


图 3-3. 程序存储器映射图



3.4.4 区域

(1) 内部 ROM 区域

1MB 的空间被保留为内部 ROM 区域。

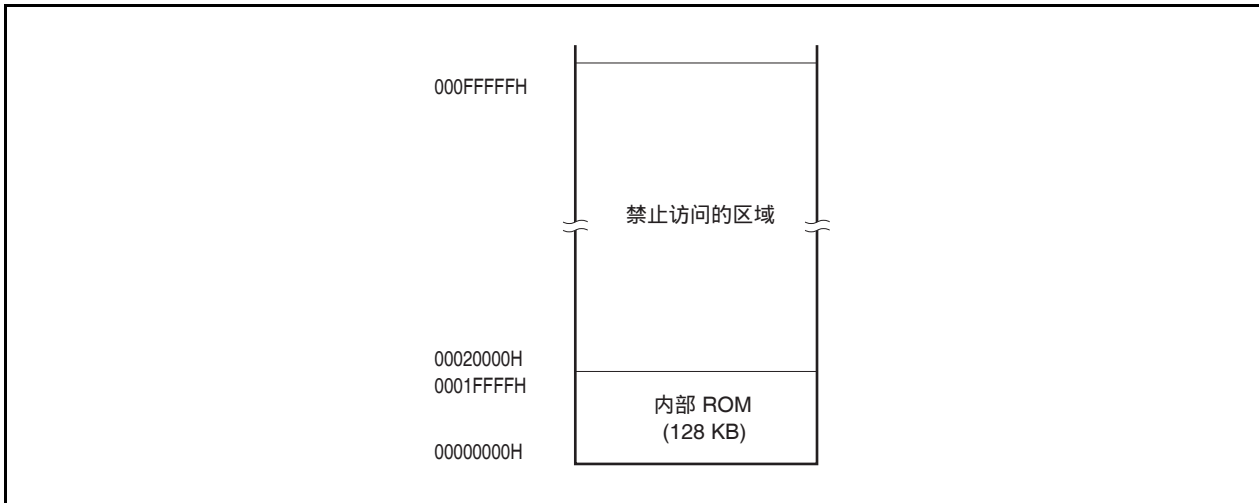
(a) 内部 ROM (128 KB)

μ PD70F3715 芯片的 128KB 内部 ROM 区域地址为 00000000H 到 0001FFFFH。

对地址 00020000H 到 000FFFFFFH 的操作是被禁止的。

Up to 1 MB is reserved as an internal ROM area.

图 3-4. 内部 ROM 区域 (128 KB)

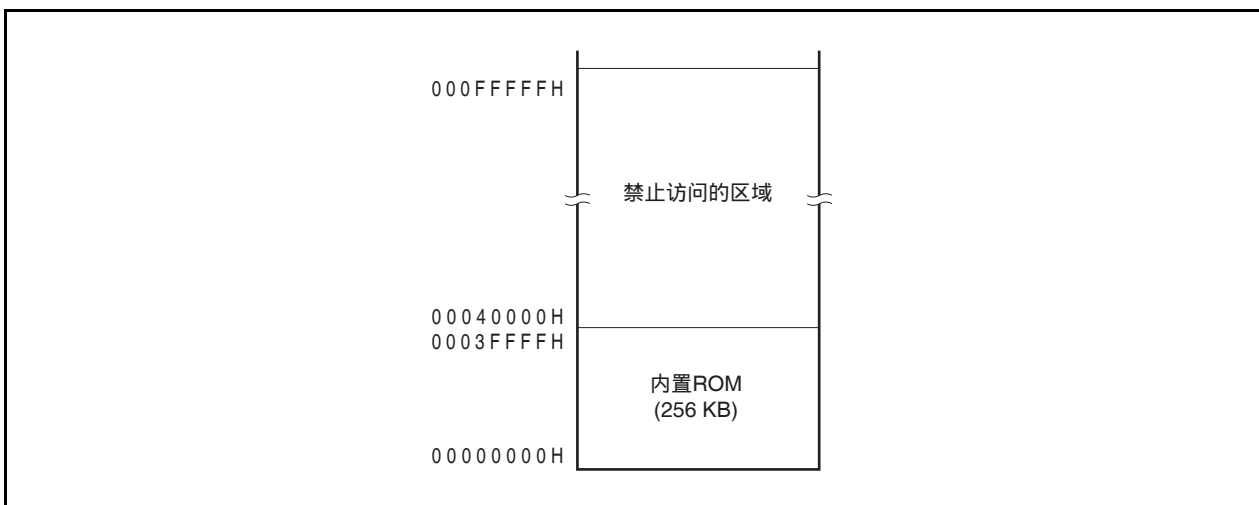


(b) 内部 ROM (256 KB)

μ PD70F3716 芯片的 256KB 内部 ROM 区域地址为 00000000H 到 0003FFFFH。

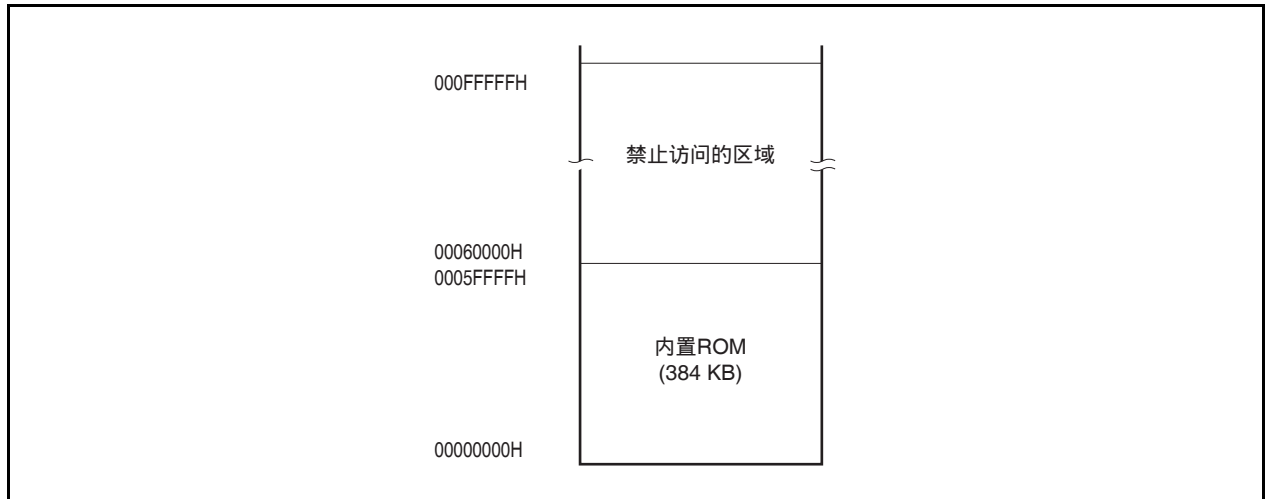
对地址 00040000H 到 000FFFFFFH 的操作是被禁止的。

图 3-5. 内部 ROM 区域 (256KB)

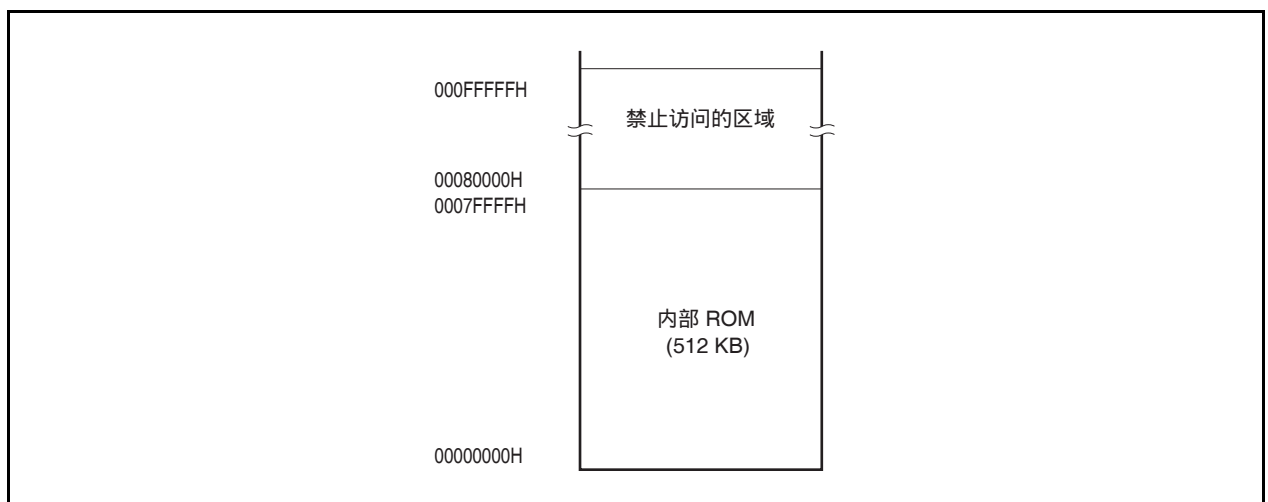


(c) 内部 ROM (384 KB)

μ PD70F3717 芯片的 384KB 内部 ROM 区域地址为 00000000H 到 0005FFFFH。
对地址 00060000H 到 000FFFFFFH 的操作是被禁止的。

图 3-6. 内部 ROM 区域 (384KB)**(d) 内部 ROM (512 KB)**

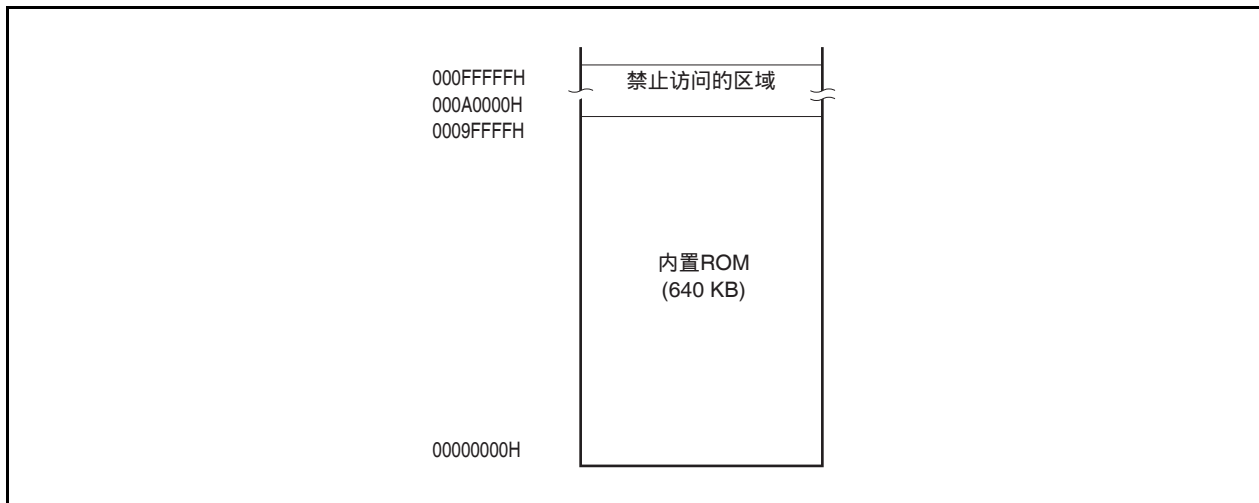
μ PD70F3718 芯片的 512KB 内部 ROM 区域地址为 00000000H 到 0007FFFFH。
对地址 00080000H 到 000FFFFFFH 的操作是被禁止的。

图 3-7. 内部 ROM 区域 (512KB)

(e) 内部 ROM (640 KB)

μPD70F3719 芯片的 640KB 内部 ROM 区域地址为 00000000H 到 0009FFFFH。

对地址 000A0000H 到 000FFFFFFH 的操作是被禁止的。

图 3-8. 内部 ROM 区域 (640KB)

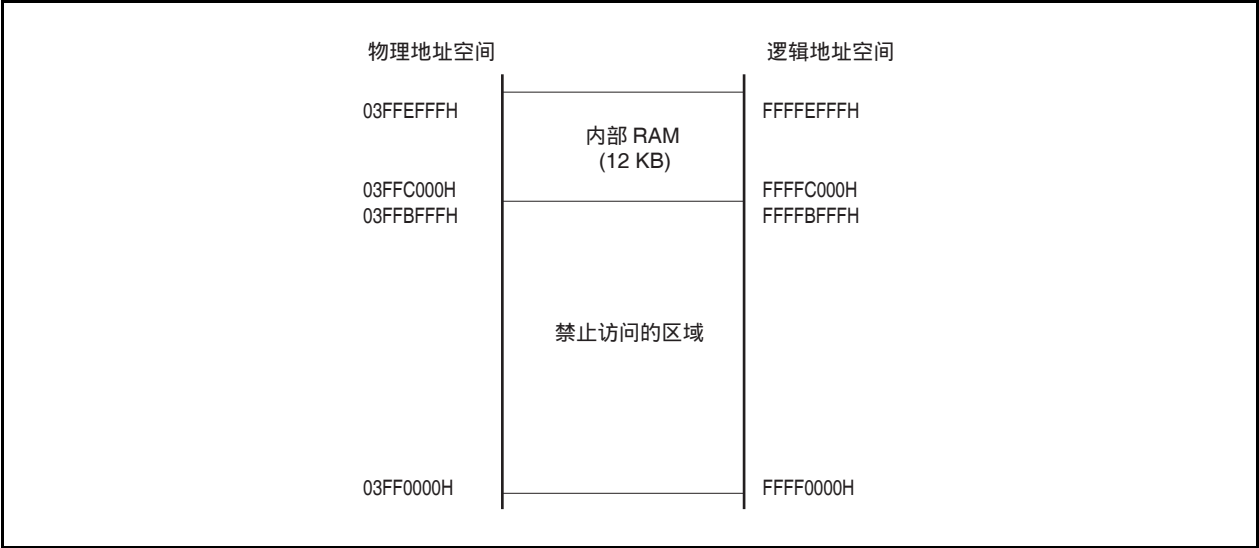
(2) 内部 RAM 区域

60KB 的空间被保留为内部 RAM 区域。

(a) 内部 RAM (12 KB)

μ PD70F3715 芯片的 12KB 内部 RAM 区域地址为 03FFC000H 到 03FFEFFFH。
对地址 03FF0000H 到 03FFBFFFH 的操作是被禁止的。

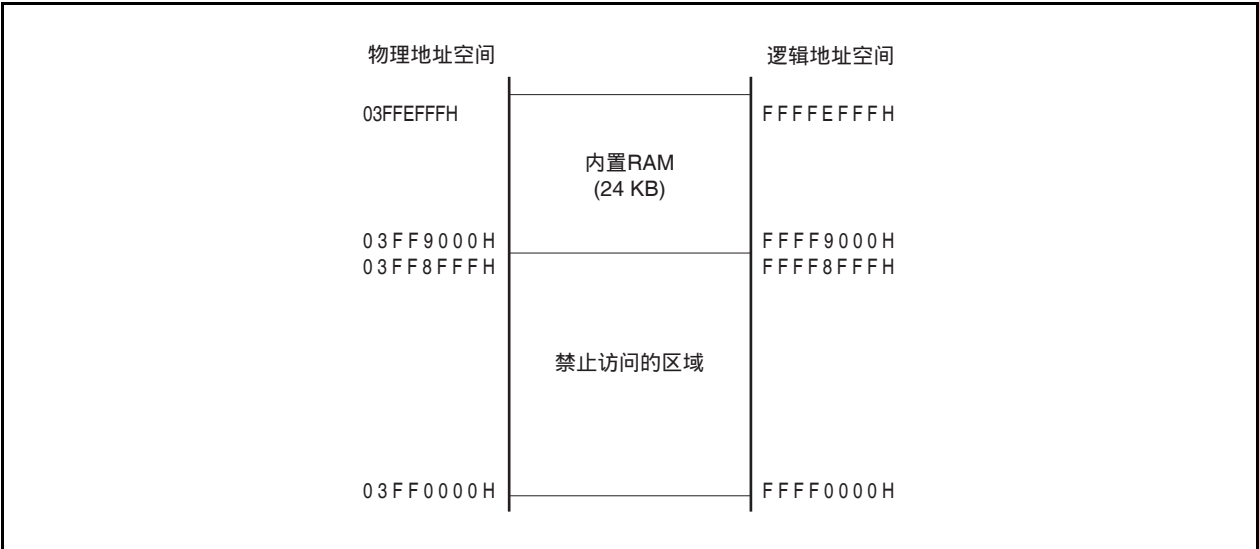
图 3-9. 内部 RAM 区域 (12 KB)



(b) 内部 RAM (24 KB)

μ PD70F3716 芯片的 24KB 内部 RAM 区域地址为 03FF9000H 到 03FFEFFFH。
对地址 03FF0000H 到 03FF8FFFH 的操作是被禁止的。

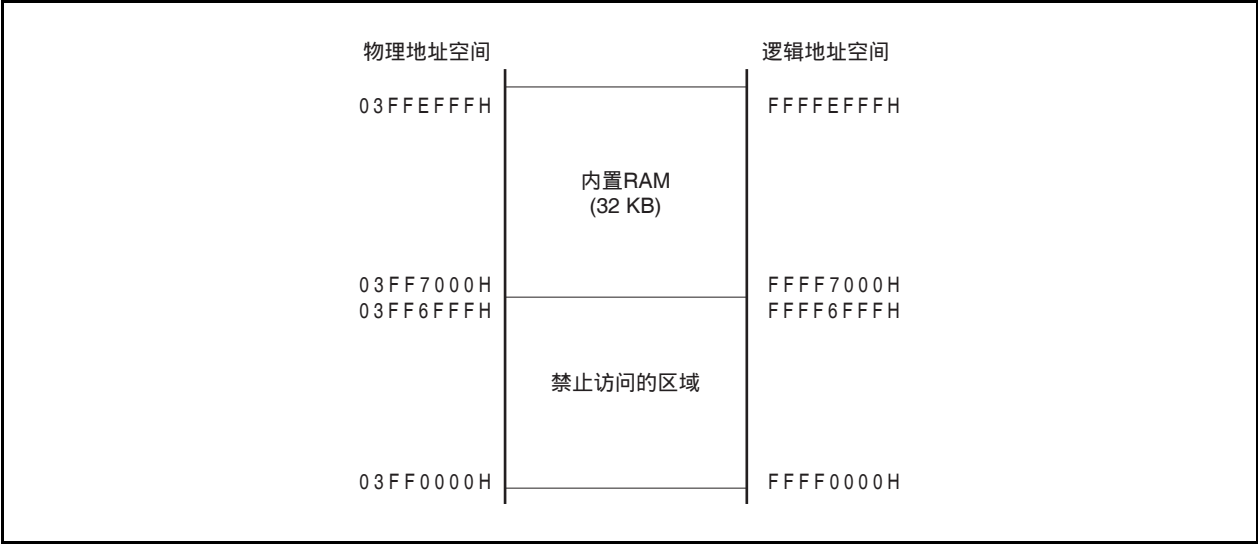
图 3-10. 内部 RAM 区域 (24 KB)



(c) 内部 RAM (32 KB)

μPD70F3717 芯片的 32KB 内部 RAM 区域地址为 03FF7000H 到 03FF6FFFH。
对地址 03FF0000H 到 03FF6FFFH 的操作是被禁止的。

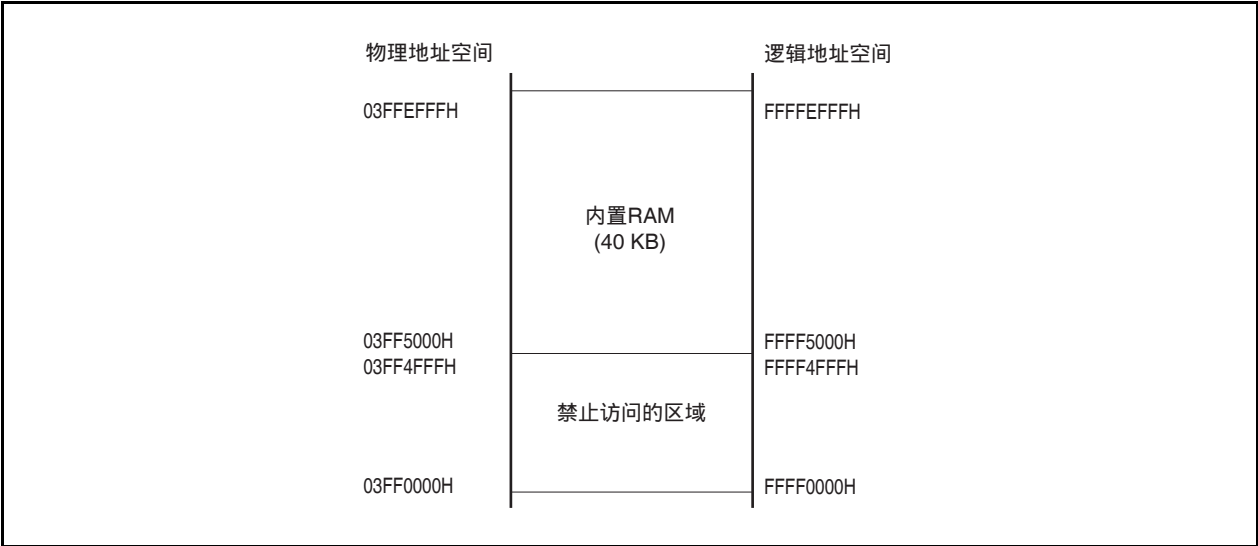
图 3-11. 内部 RAM 区域 (32 KB)



(d) 内部 RAM (40 KB)

μPD70F3718 芯片的 40KB 内部 RAM 区域地址为 03FF5000H 到 03FF6FFFH。
对地址 03FF0000H 到 03FF4FFFH 的操作是被禁止的。

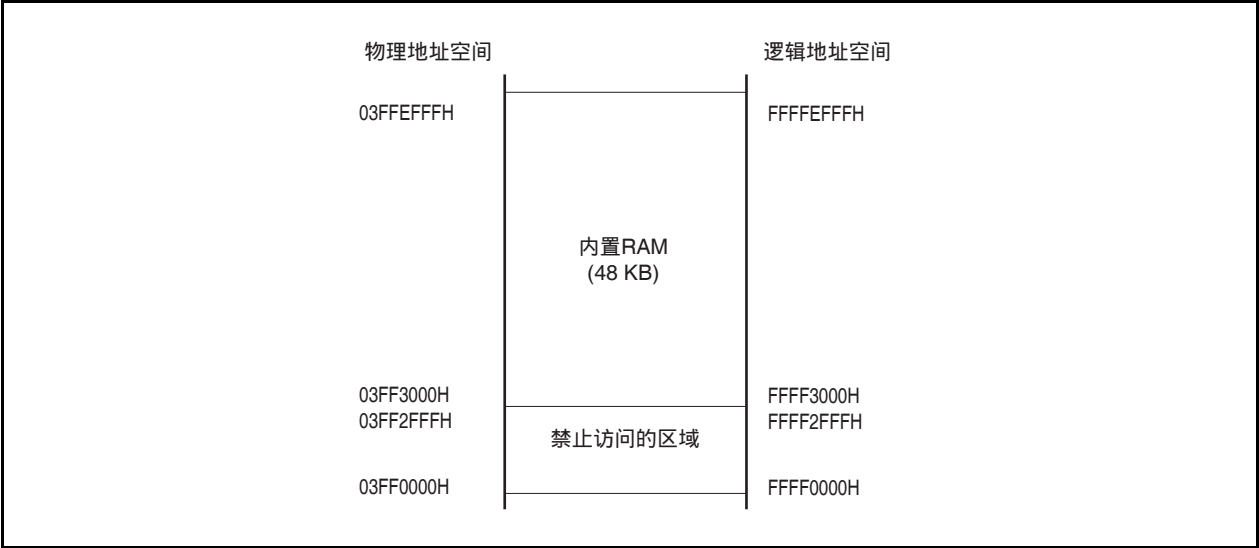
图 3-12. 内部 RAM 区域 (40 KB)



(e) 内部 RAM (48 KB)

μ PD70F3719 芯片的 48KB 内部 RAM 区域地址为 03FF3000H 到 03FFFEFFH。
对地址 03FF0000H 到 03FF2FFFH 的操作是被禁止的。

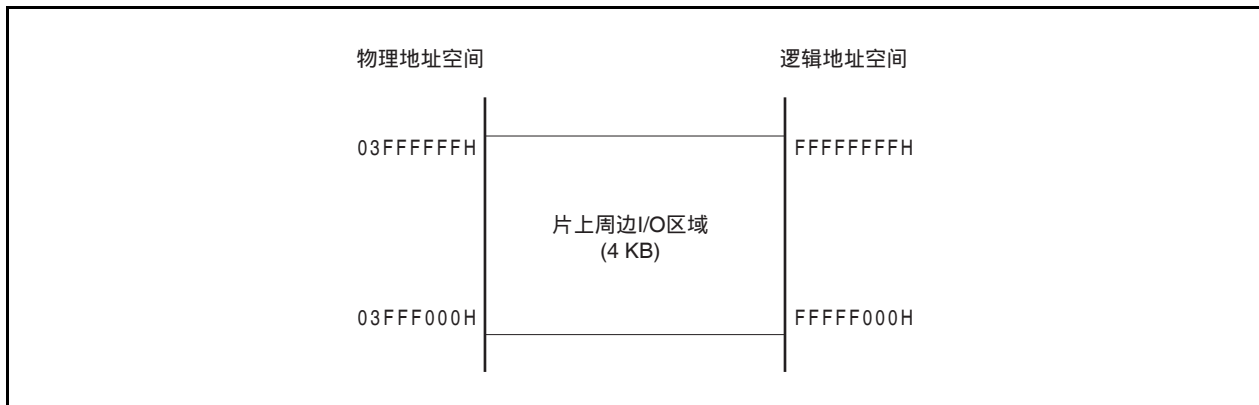
图 3-13. 内部 RAM 区域 (48 KB)



(3) 片上周边 I/O 区域

地址 03FFF000H 到 03FFFFFFH 的 4KB 空间是片上周边 I/O 区域。

图 3-14. 片上周边 I/O 区域



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

- 注意事项**
1. 当寄存器被一个字操作指令读写时，一个字的区域被分为低 16 位和高 16 位分两次字节操作先后读写，其中地址的最低 2 位被忽略。
 2. 若某寄存器仅支持字节操作，那么对其进行半字长(halfword)操作，读取时的高 8 位数据无效，写入时的低 8 位数据被写入寄存器。
 3. 未被指定为寄存器的地址为未来扩展保留。对这些地址的操作是未定的，结果的有效性也是无法保证的。

(4) 外部存储器区域

地址 0100000H 到 0FFFFFFH 的 15MB 空间分配给外部存储区域。具体细节详见 第 5 章 总线控制功能。

注意事项 V850ES/JG2 具有 22 个地址引脚(A0 到 A21)，因此外部存储空间可视为一个个重复的 4MB 的镜象。分离总线模式下或者 A20 和 A21 引脚被占用时，一定要保证 $EV_{DD} = BV_{DD} = V_{DD}$ 。

3.4.5 地址空间的推荐用法

V850ES/JG2 的结构要求在对数据空间内的操作数进行存取时，要确保有一个寄存器可以作为地址指针来使用。这个指针基址的 $\pm 32\text{KB}$ 地址范围内的操作数可以被指令直接访问。由于能够用来充当指针的通用寄存器数量有限，当指针值改变引起地址计算时，为了不影响系统的性能，要指定尽可能多的通用寄存器来保存变量值，这样也可以减少程序所占的空间。

(1) 程序空间

PC 的高 6 位始终为 0，只有低 26 位有效，因此，从地址 00000000H 开始的 64MB 的连续空间被无条件分配给程序空间。

为了在内部 RAM 区域存放程序，RAM 空间的地址如下表所列。

注意事项 如果分支指令发生在内部 RAM 区间的上沿地址处，那么跨越片上外围 I/O 区间的预取址(无效取址) 动作将不会进行。

RAM 大小	访问地址
48 KB	03FF3000H to 03FFEFFFFH
40 KB	03FF5000H to 03FFEFFFFH
32 KB	03FF7000H to 03FFEFFFFH
24 KB	03FF9000H to 03FFEFFFFH
12 KB	03FFC000H to 03FFEFFFFH

(2) 数据空间

在 V850ES/JG2 系列器件中，4GB 的 CPU 地址空间可被看作 64 个 64MB 的地址空间，因此 26 位地址的最低有效位(bit 25)是作为符号扩展位被扩展至 32 位，被用作地址值。

(a) 绕回式内存空间的应用举例

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

零寄存器(r0)是一个由硬件指定的寄存器，其值固定为零，它可以有效地减少专用于指针的寄存器需求量。

例: $\mu PD70F3717$

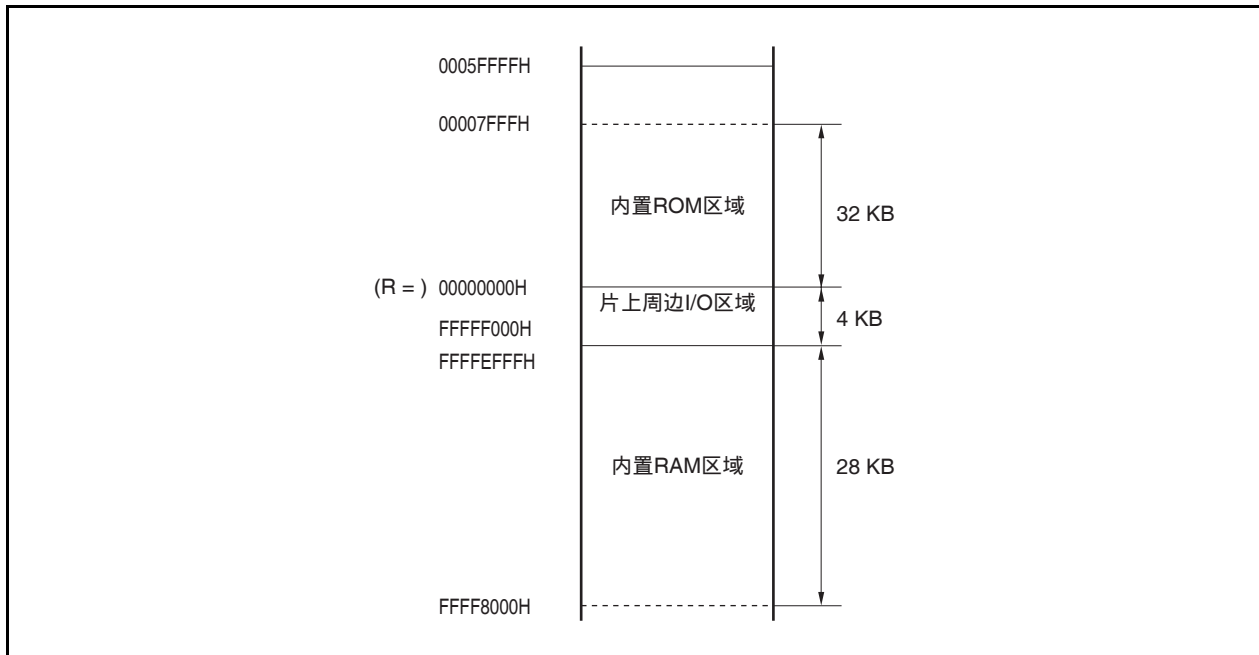
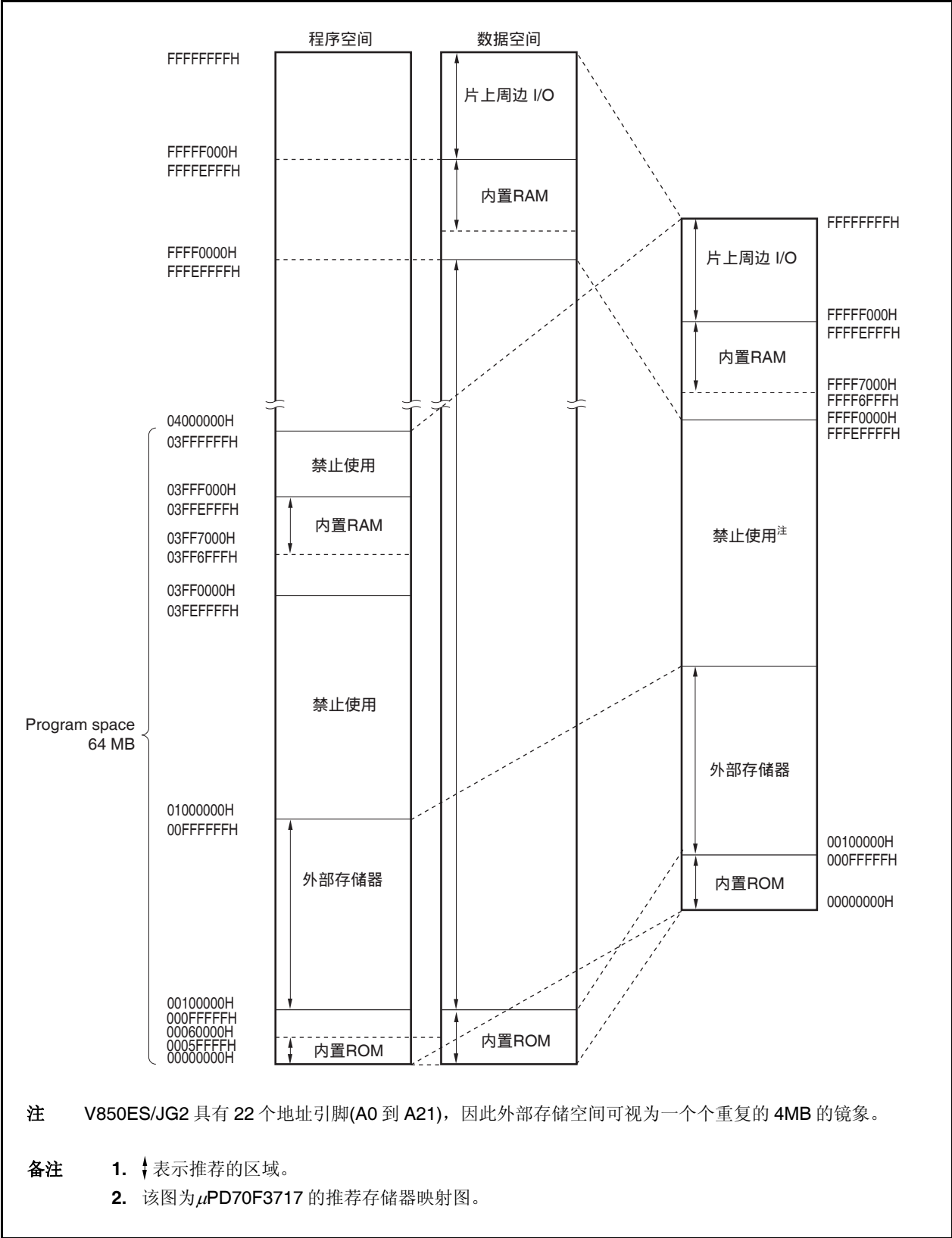


图 3-15. 推荐的存储器映射图



3.4.6 周边 I/O 寄存器

(1/10)

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
FFFFF004H	端口 DL 寄存器	PDL	R/W			√	0000H ^{※1}
FFFFF004H	端口 DL 寄存器 L	PDLL		√	√		00H ^{※1}
FFFFF005H	端口 DL 寄存器 H	PDLH		√	√		00H ^{※1}
FFFFF006H	端口 DH 寄存器	PDH		√	√		00H ^{※1}
FFFFF00AH	端口 CT 寄存器	PCT		√	√		00H ^{※1}
FFFFF00CH	端口 CM 寄存器	PCM		√	√		00H ^{※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	PMDH		√	√		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	PMCDH		√	√		00H
FFFFF04AH	端口 CT 模式控制寄存器	PM CCT		√	√		00H
FFFFF04CH	端口 CM 模式控制寄存器	PMCCM		√	√		00H
FFFFF066H	总线宽度设置寄存器	BSC				√	5555H
FFFFF06EH	系统等待控制寄存器	VSWC			√		77H
FFFFF080H	DMA 源地址寄存器 0L	DSA0L				√	未定义
FFFFF082H	DMA 源地址寄存器 0H	DSA0H				√	未定义
FFFFF084H	DMA 目的地址寄存器 0L	DDA0L				√	未定义
FFFFF086H	DMA 目的地址寄存器 0H	DDA0H				√	未定义
FFFFF088H	DMA 源地址寄存器 1L	DSA1L				√	未定义
FFFFF08AH	DMA 源地址寄存器 1H	DSA1H				√	未定义
FFFFF08CH	DMA 目的地址寄存器 1L	DDA1L				√	未定义
FFFFF08EH	DMA 目的地址寄存器 1H	DDA1H				√	未定义
FFFFF090H	DMA 源地址寄存器 2L	DSA2L				√	未定义
FFFFF092H	DMA 源地址寄存器 2H	DSA2H				√	未定义
FFFFF094H	DMA 目的地址寄存器 2L	DDA2L				√	未定义
FFFFF096H	DMA 目的地址寄存器 2H	DDA2H				√	未定义
FFFFF098H	DMA 源地址寄存器 3L	DSA3L				√	未定义
FFFFF09AH	DMA 源地址寄存器 3H	DSA3H				√	未定义
FFFFF09CH	DMA 目的地址寄存器 3L	DDA3L				√	未定义
FFFFF09EH	DMA 目的地址寄存器 3H	DDA3H				√	未定义
FFFFF0C0H	DMA 传输计数寄存器 0	DBC0				√	未定义
FFFFF0C2H	DMA 传输计数寄存器 1	DBC1				√	未定义
FFFFF0C4H	DMA 传输计数寄存器 2	DBC2				√	未定义
FFFFF0C6H	DMA 传输计数寄存器 3	DBC3				√	未定义
FFFFF0D0H	DMA 地址控制寄存器 0	DADC0				√	0000H

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

(2/10)

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
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
FFFFF110H	中断控制寄存器	LVIIIC		√	√		47H
FFFFF112H	中断控制寄存器	PIC0		√	√		47H
FFFFF114H	中断控制寄存器	PIC1		√	√		47H
FFFFF116H	中断控制寄存器	PIC2		√	√		47H
FFFFF118H	中断控制寄存器	PIC3		√	√		47H
FFFFF11AH	中断控制寄存器	PIC4		√	√		47H
FFFFF11CH	中断控制寄存器	PIC5		√	√		47H
FFFFF11EH	中断控制寄存器	PIC6		√	√		47H
FFFFF120H	中断控制寄存器	PIC7		√	√		47H
FFFFF122H	中断控制寄存器	TQ0OVIC		√	√		47H
FFFFF124H	中断控制寄存器	TQ0CCIC0		√	√		47H
FFFFF126H	中断控制寄存器	TQ0CCIC1		√	√		47H
FFFFF128H	中断控制寄存器	TQ0CCIC2		√	√		47H
FFFFF12AH	中断控制寄存器	TQ0CCIC3		√	√		47H
FFFFF12CH	中断控制寄存器	TP0OVIC		√	√		47H
FFFFF12EH	中断控制寄存器	TP0CCIC0		√	√		47H
FFFFF130H	中断控制寄存器	TP0CCIC1		√	√		47H
FFFFF132H	中断控制寄存器	TP1OVIC		√	√		47H
FFFFF134H	中断控制寄存器	TP1CCIC0		√	√		47H
FFFFF136H	中断控制寄存器	TP1CCIC1		√	√		47H
FFFFF138H	中断控制寄存器	TP2OVIC		√	√		47H
FFFFF13AH	中断控制寄存器	TP2CCIC0		√	√		47H
FFFFF13CH	中断控制寄存器	TP2CCIC1		√	√		47H
FFFFF13EH	中断控制寄存器	TP3OVIC		√	√		47H

(3/10)

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
FFFFF140H	中断控制寄存器	TP3CCIC0		√	√		47H
FFFFF142H	中断控制寄存器	TP3CCIC1		√	√		47H
FFFFF144H	中断控制寄存器	TP4OVIC		√	√		47H
FFFFF146H	中断控制寄存器	TP4CCIC0		√	√		47H
FFFFF148H	中断控制寄存器	TP4CCIC1		√	√		47H
FFFFF14AH	中断控制寄存器	TP5OVIC		√	√		47H
FFFFF14CH	中断控制寄存器	TP5CCIC0		√	√		47H
FFFFF14EH	中断控制寄存器	TP5CCIC1		√	√		47H
FFFFF150H	中断控制寄存器	TM0EQIC0		√	√		47H
FFFFF152H	中断控制寄存器	CB0RIC/IICIC1 ^{※1}		√	√		47H
FFFFF154H	中断控制寄存器	CB0TIC		√	√		47H
FFFFF156H	中断控制寄存器	CB1RIC		√	√		47H
FFFFF158H	中断控制寄存器	CB1TIC		√	√		47H
FFFFF15AH	中断控制寄存器	CB2RIC		√	√		47H
FFFFF15CH	中断控制寄存器	CB2TIC		√	√		47H
FFFFF15EH	中断控制寄存器	CB3RIC		√	√		47H
FFFFF160H	中断控制寄存器	CB3TIC		√	√		47H
FFFFF162H	中断控制寄存器	UA0RIC/CB4RIC		√	√		47H
FFFFF164H	中断控制寄存器	UA0TIC/CB4TIC		√	√		47H
FFFFF166H	中断控制寄存器	UA1RIC/IICIC2		√	√		47H
FFFFF168H	中断控制寄存器	UA1TIC		√	√		47H
FFFFF16AH	中断控制寄存器	UA2RIC/IICIC0		√	√		47H
FFFFF16CH	中断控制寄存器	UA2TIC		√	√		47H
FFFFF16EH	中断控制寄存器	ADIC		√	√		47H
FFFFF170H	中断控制寄存器	DMAIC0		√	√		47H
FFFFF172H	中断控制寄存器	DMAIC1		√	√		47H
FFFFF174H	中断控制寄存器	DMAIC2		√	√		47H
FFFFF176H	中断控制寄存器	DMAIC3		√	√		47H
FFFFF178H	中断控制寄存器	KRIC		√	√		47H
FFFFF17AH	中断控制寄存器	WTIIC		√	√		47H
FFFFF17CH	中断控制寄存器	WTIC		√	√		47H
FFFFF1FAH	当前服务优先级寄存器	ISPR	R	√	√		00H
FFFFF1FCH	命令寄存器	PRCMD	W		√		Undefined
FFFFF1FEH	节能控制寄存器	PSC	R/W	√	√		00H
FFFFF200H	A/D 转换器模式寄存器 0	ADA0M0		√	√		00H
FFFFF201H	A/D 转换器模式寄存器 1	ADA0M1		√	√		00H
FFFFF202H	A/D 转换器通道指定寄存器	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 转换器结果寄存器 0	ADA0CR0				√	Undefined
FFFFF211H	A/D 转换器结果寄存器 0H	ADA0CR0H	R		√		Undefined
FFFFF212H	A/D 转换器结果寄存器 1	ADA0CR1				√	Undefined
FFFFF213H	A/D 转换器结果寄存器 1H	ADA0CR1H			√		Undefined

(4/10)

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
FFFFF214H	A/D 转换器结果寄存器 2	ADA0CR2				√	Undefined
FFFFF215H	A/D 转换器结果寄存器 2H	ADA0CR2H			√		Undefined
FFFFF216H	A/D 转换器结果寄存器 3	ADA0CR3				√	Undefined
FFFFF217H	A/D 转换器结果寄存器 3H	ADA0CR3H			√		Undefined
FFFFF218H	A/D 转换器结果寄存器 4	ADA0CR4				√	Undefined
FFFFF219H	A/D 转换器结果寄存器 4H	ADA0CR4H			√		Undefined
FFFFF21AH	A/D 转换器结果寄存器 5	ADA0CR5				√	Undefined
FFFFF21BH	A/D 转换器结果寄存器 5H	ADA0CR5H			√		Undefined
FFFFF21CH	A/D 转换器结果寄存器 6	ADA0CR6				√	Undefined
FFFFF21DH	A/D 转换器结果寄存器 6H	ADA0CR6H			√		Undefined
FFFFF21EH	A/D 转换器结果寄存器 7	ADA0CR7				√	Undefined
FFFFF21FH	A/D 转换器结果寄存器 7H	ADA0CR7H			√		Undefined
FFFFF220H	A/D 转换器结果寄存器 8	ADA0CR8				√	Undefined
FFFFF221H	A/D 转换器结果寄存器 8H	ADA0CR8H			√		Undefined
FFFFF222H	A/D 转换器结果寄存器 9	ADA0CR9				√	Undefined
FFFFF223H	A/D 转换器结果寄存器 9H	ADA0CR9H			√		Undefined
FFFFF224H	A/D 转换器结果寄存器 10	ADA0CR10				√	Undefined
FFFFF225H	A/D 转换器结果寄存器 10H	ADA0CR10H			√		Undefined
FFFFF226H	A/D 转换器结果寄存器 11	ADA0CR11				√	Undefined
FFFFF227H	A/D 转换器结果寄存器 11H	ADA0CR11H			√		Undefined
FFFFF280H	D/A 转换器转换值设置寄存器 0	DA0CS0	R/W		√		00H
FFFFF281H	D/A 转换器转换值设置寄存器 1	DA0CS1			√		00H
FFFFF282H	D/A 转换器模式寄存器	DA0M		√	√		00H
FFFFF300H	按键中断模式寄存器	KRM		√	√		00H
FFFFF308H	选择操作控制寄存器	SELCNT0		√	√		00H
FFFFF318H	噪声消除控制寄存器	NFC			√		00H
FFFFF320H	BRG1 预分频器模式寄存器	PRSM1		√	√		00H
FFFFF321H	BRG1 预分频器比较寄存器	PRSCM1			√		00H
FFFFF324H	BRG2 预分频器模式寄存器	PRSM2		√	√		00H
FFFFF325H	BRG2 预分频器比较寄存器	PRSCM2			√		00H
FFFFF328H	BRG3 预分频器模式寄存器	PRSM3		√	√		00H
FFFFF329H	BRG3 预分频器比较寄存器	PRSCM3			√		00H
FFFFF340H	IIC 分频时钟选择寄存器	OCKS0			√		00H
FFFFF344H	IIC 分频时钟选择寄存器	OCKS1			√		00H
FFFFF400H	端口 0 寄存器	P0		√	√		00H ^{注 3}
FFFFF402H	端口 1 寄存器	P1		√	√		00H ^{注 3}
FFFFF406H	端口 3 寄存器	P3				√	0000H ^{注 3}
FFFFF406H	端口 3 寄存器 L	P3L		√	√		00H ^{Note}
FFFFF407H	端口 3 寄存器 H	P3H		√	√		00H ^{Note}
FFFFF408H	端口 4 寄存器	P4		√	√		00H ^{注 3}
FFFFF40AH	端口 5 寄存器	P5		√	√		00H ^{注 3}
FFFFF40EH	端口 7 寄存器 L	P7L		√	√		00H ^{注 3}
FFFFF40FH	端口 7 寄存器 H	P7H		√	√		00H ^{注 3}

Note The output latch is 00H or 0000H. When these registers are input, the pin statuses are read.

(5/10)

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
FFFFF412H	端口 9 寄存器	P9				√	0000H ^{註3}
FFFFF412H	端口 9 寄存器 L	P9L		√	√		00H ^{註3}
FFFFF413H	端口 9 寄存器 H	P9H		√	√		00H ^{註3}
FFFFF420H	端口 0 模式寄存器	PM0		√	√		FFH
FFFFF422H	端口 1 模式寄存器	PM1		√	√		FFH
FFFFF426H	端口 3 模式寄存器	PM3				√	FFFFH
FFFFF426H	端口 3 模式寄存器 L	PM3L		√	√		FFH
FFFFF427H	端口 3 模式寄存器 H	PM3H		√	√		FFH
FFFFF428H	端口 4 模式寄存器	PM4	R/W	√	√		FFH
FFFFF42AH	端口 5 模式寄存器	PM5		√	√		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
FFFFF446H	端口 3 模式控制寄存器	PMC3				√	0000H
FFFFF446H	端口 3 模式控制寄存器 L	PMC3L		√	√		00H
FFFFF447H	端口 3 模式控制寄存器 H	PMC3H		√	√		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
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
FFFFF466H	端口 3 功能控制寄存器	PFC3				√	0000H
FFFFF466H	端口 3 功能控制寄存器 L	PFC3L		√	√		00H
FFFFF467H	端口 3 功能控制寄存器 H	PFC3H		√	√		00H
FFFFF468H	端口 4 功能控制寄存器	PFC4		√	√		00H
FFFFF46AH	端口 5 功能控制寄存器	PFC5		√	√		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	TMQ0 控制寄存器 0	TQ0CTL0		√	√		00H
FFFFF541H	TMQ0 控制寄存器 1	TQ0CTL1		√	√		00H
FFFFF542H	TMQ0 I/O 控制寄存器 0	TQ0IOC0		√	√		00H
FFFFF543H	TMQ0 I/O 控制寄存器 1	TQ0IOC1		√	√		00H
FFFFF544H	TMQ0 I/O 控制寄存器 2	TQ0IOC2		√	√		00H

Note The output latch is 00H or 0000H. When these registers are input, the pin statuses are read.

(6/10)

Address	Function Register Name	Symbol	R/W	Manipula 表 Bits			Default Value
				1	8	16	
FFFFF545H	TMQ0 选项寄存器 0	TQ0OPT0		√	√		00H
FFFFF546H	TMQ0 捕捉/比较寄存器 0	TQ0CCR0				√	0000H
FFFFF548H	TMQ0 捕捉/比较寄存器 1	TQ0CCR1				√	0000H
FFFFF54AH	TMQ0 捕捉/比较寄存器 2	TQ0CCR2				√	0000H
FFFFF54CH	TMQ0 捕捉/比较寄存器 3	TQ0CCR3				√	0000H
FFFFF54EH	TMQ0 计数器读取缓存寄存器	TQ0CNT	R			√	0000H
FFFFF590H	TMP0 控制寄存器 0	TP0CTL0	R/W	√	√		00H
FFFFF591H	TMP0 控制寄存器 1	TP0CTL1		√	√		00H
FFFFF592H	TMP0I/O 控制寄存器 0	TP0IOC0		√	√		00H
FFFFF593H	TMP0I/O 控制寄存器 1	TP0IOC1		√	√		00H
FFFFF594H	TMP0I/O 控制寄存器 2	TP0IOC2		√	√		00H
FFFFF595H	TMP0 选项寄存器 0	TP0OPT0		√	√		00H
FFFFF596H	TMP0 捕捉/比较寄存器 0	TP0CCR0				√	0000H
FFFFF598H	TMP0 捕捉/比较寄存器 1	TP0CCR1				√	0000H
FFFFF59AH	TMP0 计数器读取缓存寄存器	TP0CNT	R			√	0000H
FFFFF5A0H	TMP1 控制寄存器 0	TP1CTL0	R/W	√	√		00H
FFFFF5A1H	TMP1 控制寄存器 1	TP1CTL1		√	√		00H
FFFFF5A2H	TMP1I/O 控制寄存器 0	TP1IOC0		√	√		00H
FFFFF5A3H	TMP1I/O 控制寄存器 1	TP1IOC1		√	√		00H
FFFFF5A4H	TMP1I/O 控制寄存器 2	TP1IOC2		√	√		00H
FFFFF5A5H	TMP1 选项寄存器 0	TP1OPT0		√	√		00H
FFFFF5A6H	TMP1 捕捉/比较寄存器 0	TP1CCR0				√	0000H
FFFFF5A8H	TMP1 捕捉/比较寄存器 1	TP1CCR1				√	0000H
FFFFF5AAH	TMP1 计数器读取缓存寄存器	TP1CNT	R			√	0000H
FFFFF5B0H	TMP2 控制寄存器 0	TP2CTL0	R/W	√	√		00H
FFFFF5B1H	TMP2 控制寄存器 1	TP2CTL1		√	√		00H
FFFFF5B2H	TMP2I/O 控制寄存器 0	TP2IOC0		√	√		00H
FFFFF5B3H	TMP2I/O 控制寄存器 1	TP2IOC1		√	√		00H
FFFFF5B4H	TMP2I/O 控制寄存器 2	TP2IOC2		√	√		00H
FFFFF5B5H	TMP2 选项寄存器 0	TP2OPT0		√	√		00H
FFFFF5B6H	TMP2 捕捉/比较寄存器 0	TP2CCR0				√	0000H
FFFFF5B8H	TMP2 捕捉/比较寄存器 1	TP2CCR1				√	0000H
FFFFF5BAH	TMP2 计数器读取缓存寄存器	TP2CNT	R			√	0000H
FFFFF5C0H	TMP3 控制寄存器 0	TP3CTL0	R/W	√	√		00H
FFFFF5C1H	TMP3 控制寄存器 1	TP3CTL1		√	√		00H
FFFFF5C2H	TMP3I/O 控制寄存器 0	TP3IOC0		√	√		00H
FFFFF5C3H	TMP3I/O 控制寄存器 1	TP3IOC1		√	√		00H
FFFFF5C4H	TMP3I/O 控制寄存器 2	TP3IOC2		√	√		00H
FFFFF5C5H	TMP3 选项寄存器 0	TP3OPT0		√	√		00H
FFFFF5C6H	TMP3 捕捉/比较寄存器 0	TP3CCR0				√	0000H
FFFFF5C8H	TMP3 捕捉/比较寄存器 1	TP3CCR1				√	0000H
FFFFF5CAH	TMP3 计数器读取缓存寄存器	TP3CNT	R			√	0000H
FFFFF5D0H	TMP4 控制寄存器 0	TP4CTL0	R/W	√	√		00H
FFFFF5D1H	TMP4 控制寄存器 1	TP4CTL1		√	√		00H

(7/10)

地址	寄存器功能名	符号	R/W	可使用操作位数				默认值
				1	8	16	32	
FFFFF5D3H	TMP4I/O 控制寄存器 1	TP4IOC1		√	√			00H
FFFFF5D4H	TMP4I/O 控制寄存器 2	TP4IOC2		√	√			00H
FFFFF5D5H	TMP4 选项寄存器 0	TP4OPT0		√	√			00H
FFFFF5D6H	TMP4 捕捉/比较寄存器 0	TP4CCR0				√		0000H
FFFFF5D8H	TMP4 捕捉/比较寄存器 1	TP4CCR1				√		0000H
FFFFF5DAH	TMP4 计数器读取缓存 寄存器	TP4CNT	R			√		
FFFFF5E0H	TMP5 控制寄存器 0	TP5CTL0	R/W	√	√			
FFFFF5E1H	TMP5 控制寄存器 1	TP5CTL1		√	√			
FFFFF5E2H	TMP5I/O 控制寄存器 0	TP5IOC0		√	√			
FFFFF5E3H	TMP5I/O 控制寄存器 1	TP5IOC1		√	√			
FFFFF5E4H	TMP5I/O 控制寄存器 2	TP5IOC2		√	√			
FFFFF5E5H	TMP5 选项寄存器 0	TP5OPT0		√	√			
FFFFF5E6H	TMP5 捕捉/比较寄存器 0	TP5CCR0				√		
FFFFF5E8H	TMP5 捕捉/比较寄存器 1	TP5CCR1				√		
FFFFF5EAH	TMP5 计数器读取缓存寄存器	TP5CNT	R			√		
FFFFF680H	钟表定时器操作模式寄存器	WTM	R/W	√	√			
FFFFF690H	TMM0 控制寄存器 0	TM0CTL0		√	√			
FFFFF694H	TMM0 比较寄存器 0	TM0CMP0				√		
FFFFF6C0H	振荡稳定时间选择寄存器	OSTS			√			
FFFFF6C1H	PLL 锁止时间指定寄存器	PLLS			√			
FFFFF6D0H	看门狗定时器模式寄存器 2	WDTM2			√			
FFFFF6D1H	看门狗定时器允许寄存器	WDTE			√			
FFFFF6E0H	实时输出缓存寄存器 0L	RTBL0		√	√			
FFFFF6E2H	实时输出缓存寄存器 0H	RTBH0		√	√			
FFFFF6E4H	实时输出端口模式寄存器 0	RTPM0		√	√			
FFFFF6E5H	实时输出端口控制寄存器 0	RTPC0		√	√			
FFFFF706H	端口 3 功能控制扩展寄存器 L	PFCE3L		√	√			
FFFFF70AH	端口 5 功能控制扩展寄存器	PFCE5		√	√			
FFFFF712H	端口 9 功能控制扩展寄存器	PFCE9				√		
FFFFF712H	端口 9 功能控制扩展寄存器 L	PFCE9L		√	√			00H
FFFFF713H	端口 9 功能控制扩展寄存器 H	PFCE9H		√	√			00H
FFFFF802H	系统状态寄存器	SYS		√	√			
FFFFF80CH	内部振荡模式寄存器	RCM		√	√			
FFFFF810H	DMA 触发源寄存器 0	DTFR0		√	√			
FFFFF812H	DMA 触发源寄存器 1	DTFR1		√	√			
FFFFF814H	DMA 触发源寄存器 2	DTFR2		√	√			
FFFFF816H	DMA 触发源寄存器 3	DTFR3		√	√			
FFFFF820H	节电模式寄存器	PSMR		√	√			
FFFFF822H	时钟控制寄存器	CKC		√	√			
FFFFF824H	锁相环锁止状态寄存器	LOCKR	R	√	√			
FFFFF828H	处理器时钟控制寄存器	PCC	R/W	√	√			
FFFFF82CH	PLL 控制寄存器	PLLCTL		√	√			

(8/10)

地址	寄存器功能名	符号	R/W	可使用操作位数				默认值
				1	8	16	32	
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		√	√			
FFFFF888H	复位源标志寄存器	RESF		√	√			
FFFFF890H	低电压检测寄存器	LVIM		√	√			
FFFFF891H	低电压检测电压选择寄存器	LVIS			√			
FFFFF892H	内部 RAM 数据状态寄存器	RAMS		√	√			
FFFFF8B0H	预分频模式寄存器	PRSM0		√	√			
FFFFF8B1H	预分频器比较寄存器	PRSCM0			√			
FFFFF9FCH	片上调试模式寄存器	OCDM		√	√			
FFFFF9FEH	周边仿真寄存器 1	PEMU1 ^註		√	√			
FFFFFA00H	UARTA0 控制寄存器 0	UA0CTL0		√	√			
FFFFFA01H	UARTA0 控制寄存器 1	UA0CTL1			√			
FFFFFA02H	UARTA0 控制寄存器 2	UA0CTL2			√			
FFFFFA03H	UARTA0 选项控制寄存器 0	UA0OPT0		√	√			
FFFFFA04H	UARTA0 状态寄存器	UA0STR		√	√			
FFFFFA06H	UARTA0 接收数据寄存器	UA0RX	R		√			
FFFFFA07H	UARTA0 发送数据寄存器	UA0TX	R/W		√			
FFFFFA10H	UARTA1 控制寄存器 0	UA1CTL0		√	√			
FFFFFA11H	UARTA1 控制寄存器 1	UA1CTL1			√			
FFFFFA12H	UARTA1 控制寄存器 2	UA1CTL2			√			
FFFFFA13H	UARTA1 选项控制寄存器 0	UA1OPT0		√	√			
FFFFFA14H	UARTA1 状态寄存器	UA1STR		√	√			
FFFFFA16H	UARTA1 接收数据寄存器	UA1RX	R		√			
FFFFFA17H	UARTA1 发送数据寄存器	UA1TX	R/W		√			
FFFFFA20H	UARTA2 控制寄存器 0	UA2CTL0		√	√			
FFFFFA21H	UARTA2 控制寄存器 1	UA2CTL1			√			
FFFFFA22H	UARTA2 控制寄存器 2	UA2CTL2			√			
FFFFFA23H	UARTA2 选项控制寄存器 0	UA2OPT0		√	√			
FFFFFA24H	UARTA2 状态寄存器	UA2STR		√	√			
FFFFFA26H	UARTA2 接收数据寄存器	UA2RX	R		√			
FFFFFA27H	UARTA2 发送数据寄存器	UA2TX	R/W		√			
FFFFFC00H	外部中断下降沿指定寄存器 0	INTF0		√	√			
FFFFFC06H	外部中断下降沿指定寄存器 3	INTF3		√	√			
FFFFFC13H	外部中断下降沿指定寄存器 9H	INTF9H		√	√			00H
FFFFFC20H	外部中断上升沿指定寄存器 0	INTR0		√	√			00H
FFFFFC26H	外部中断上升沿指定寄存器 3	INTR3		√	√			00H
FFFFFC33H	外部中断上升沿指定寄存器 9H	INTR9H		√	√			00H
FFFFFC60H	端口 0 功能控制寄存器	PF0		√	√			00H
FFFFFC66H	端口 3 功能控制寄存器	PF3				√		0000H
FFFFFC66H	端口 3 功能控制寄存器 L	PF3L		√	√			00H
FFFFFC67H	端口 3 功能控制寄存器 H	PF3H		√	√			00H
FFFFFC68H	端口 4 功能控制寄存器	PF4		√	√			00H

Note Only during emulation

(9/10)

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
FFFFFC6AH	端口 5 功能控制寄存器	PF5		√	√		00H
FFFFFC72H	端口 9 功能控制寄存器	PF9				√	0000H
FFFFFC72H	端口 9 功能控制寄存器 L	PF9L		√	√		00H
FFFFFC73H	端口 9 功能控制寄存器 H	PF9H		√	√		00H
FFFFFD00H	CSIB0 控制寄存器 0	CB0CTL0		√	√		01H
FFFFFD01H	CSIB0 控制寄存器 1	CB0CTL1		√	√		00H
FFFFFD02H	CSIB0 控制寄存器 2	CB0CTL2			√		00H
FFFFFD03H	CSIB0 状态寄存器	CB0STR		√	√		00H
FFFFFD04H	CSIB0 接收数据寄存器	CB0RX	R			√	0000H
FFFFFD04H	CSIB0 接收数据寄存器 L	CB0RXL			√		00H
FFFFFD06H	CSIB0 发送数据寄存器	CB0TX	R/W			√	0000H
FFFFFD06H	CSIB0 发送数据寄存器 L	CB0TXL			√		00H
FFFFFD10H	CSIB1 控制寄存器 0	CB1CTL0		√	√		01H
FFFFFD11H	CSIB1 控制寄存器 1	CB1CTL1		√	√		00H
FFFFFD12H	CSIB1 控制寄存器 2	CB1CTL2			√		00H
FFFFFD13H	CSIB1 状态寄存器	CB1STR		√	√		00H
FFFFFD14H	CSIB1 接收数据寄存器	CB1RX				√	0000H
FFFFFD14H	CSIB1 接收数据寄存器 L	CB1RXL			√		00H
FFFFFD16H	CSIB1 发送数据寄存器	CB1TX				√	0000H
FFFFFD16H	CSIB1 发送数据寄存器 L	CB1TXL			√		00H
FFFFFD20H	CSIB2 控制寄存器 0	CB2CTL0		√	√		01H
FFFFFD21H	CSIB2 控制寄存器 1	CB2CTL1		√	√		00H
FFFFFD22H	CSIB2 控制寄存器 2	CB2CTL2			√		00H
FFFFFD23H	CSIB2 状态寄存器	CB2STR		√	√		00H
FFFFFD24H	CSIB2 接收数据寄存器	CB2RX				√	0000H
FFFFFD24H	CSIB2 接收数据寄存器 L	CB2RXL			√		00H
FFFFFD26H	CSIB2 发送数据寄存器	CB2TX				√	0000H
FFFFFD26H	CSIB2 发送数据寄存器 L	CB2TXL			√		00H
FFFFFD30H	CSIB3 控制寄存器 0	CB3CTL0		√	√		01H
FFFFFD31H	CSIB3 控制寄存器 1	CB3CTL1		√	√		00H
FFFFFD32H	CSIB3 控制寄存器 2	CB3CTL2			√		00H
FFFFFD33H	CSIB3 状态寄存器	CB3STR		√	√		00H
FFFFFD34H	CSIB3 接收数据寄存器	CB3RX				√	0000H
FFFFFD34H	CSIB3 接收数据寄存器 L	CB3RXL			√		00H
FFFFFD36H	CSIB3 发送数据寄存器	CB3TX				√	0000H
FFFFFD36H	CSIB3 发送数据寄存器 L	CB3TXL			√		00H
FFFFFD40H	CSIB4 控制寄存器 0	CB4CTL0		√	√		01H
FFFFFD41H	CSIB4 控制寄存器 1	CB4CTL1		√	√		00H
FFFFFD42H	CSIB4 控制寄存器 2	CB4CTL2			√		00H
FFFFFD43H	CSIB4 状态寄存器	CB4STR		√	√		00H
FFFFFD44H	CSIB4 接收数据寄存器	CB4RX				√	0000H
FFFFFD44H	CSIB4 接收数据寄存器 L	CB4RXL			√		00H

(10/10)

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
FFFFFD46H	CSIB4 发送数据寄存器	CB4TX	R/W			√	0000H
FFFFFD46H	CSIB4 发送数据寄存器 L	CB4TXL			√		00H
FFFFFD80H	IIC 移位寄存器 0	IIC0			√		00H
FFFFFD82H	IIC 控制寄存器 0	IICC0		√	√		00H
FFFFFD83H	从设备地址寄存器 0	SVA0			√		00H
FFFFFD84H	IIC 时钟选择寄存器 0	IICCL0		√	√		00H
FFFFFD85H	IIC 功能扩展寄存器 0	IICX0		√	√		00H
FFFFFD86H	IIC 状态寄存器 0	IICS0	R	√	√		00H
FFFFFD8AH	IIC 标志寄存器 0	IICF0	R/W	√	√		00H
FFFFFD90H	IIC 移位寄存器 1	IIC1			√		00H
FFFFFD92H	IIC 控制寄存器 1	IICC1		√	√		00H
FFFFFD93H	从设备地址寄存器 1	SVA1			√		00H
FFFFFD94H	IIC 时钟选择寄存器 1	IICCL1		√	√		00H
FFFFFD95H	IIC 功能扩展寄存器 1	IICX1		√	√		00H
FFFFFD96H	IIC 状态寄存器 1	IICS1	R	√	√		00H
FFFFFD9AH	IIC 标志寄存器 1	IICF1	R/W	√	√		00H
FFFFFDA0H	IIC 移位寄存器 2	IIC2			√		00H
FFFFFDA2H	IIC 控制寄存器 2	IICC2		√	√		00H
FFFFFDA3H	IIC 从设备地址寄存器 2	SVA2			√		00H
FFFFFDA4H	IIC 时钟选择寄存器 2	IICCL2		√	√		00H
FFFFFDA5H	IIC 功能扩展寄存器 2	IICX2		√	√		00H
FFFFFDA6H	IIC 状态寄存器 2	IICS2	R	√	√		00H
FFFFDAAH	IIC 标志寄存器 2	IICF2	R/W	√	√		00H
FFFFDBEH	外部总线接口模式控制寄存器	EXIMC		√	√		00H

3.4.7 特殊寄存器

特殊寄存器是为了防止由程序挂起引起的数据非法写入而受到保护的寄存器。V850ES/JG2 中共有如下 8 个特殊寄存器。

- 节能控制寄存器 (PSC)
- 时钟控制寄存器 (CKC)
- 处理器时钟控制寄存器 (PCC)
- 时钟监视器模式寄存器 (CLM)
- 复位源标志寄存器 (RESF)
- 低电压检测寄存器 (LVIM)
- 内部 RAM 数据状态寄存器 (RAMS)
- 片上调试模式设置寄存器 (OCDM)

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

(1) 向特殊寄存器中写入数据

请按照以下流程向特殊寄存器中写入数据。

- <1> 禁止 DMA 操作
- <2> 将要向特殊寄存器中写入的数据储存在通用寄存器中
- <3> 将步骤<2>中存储的数据写入 PRCMD 寄存器
- <4> 向特殊寄存器写入数据(使用以下指令):
 - 存指令(ST/STT 指令)
 - 位操作指令(SET1/CLR1/NOT1 指令)
- (<5> 到 <9> 插入 NOP 指令 (5 条)。)[‡]
- <10> 依照实际需要, 使能 DMA 操作

[例] 写 PSC 寄存器 (设置待机模式)

```

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

```

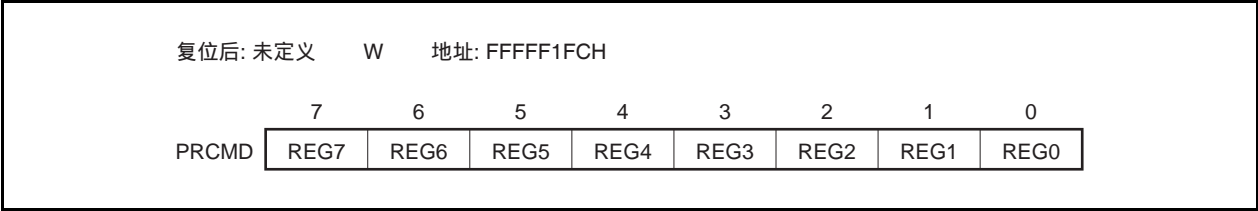
读取特殊寄存器时不需经过特定时序流程。

注 在设置 IDLE1, IDLE2 或 STOP 模式(通过将 PSC.STP 位置 1)后, 必须要紧跟 5 条 NOP 指令。

- 注意事项**
1. 当对命令寄存器进行写操作时, 系统将不响应中断。存储指令应该是按上述的第<3>和<4>步骤连续执行的。若在步骤<3>和<4>之间执行了其他指令, 而这个指令又进行了中断的响应, 则将打乱上述的顺序, 导致设备误操作
 2. 尽管对 PRCMD 写入的数据为虚数据, 但还是请使用同一个通用寄存器, 来设置例子中的第<4>步中的特殊寄存器来向 PRCMD 寄存器(例子中的<3>)写入数据。这个过程也同样可以用于使用通用寄存器进行寻址的操作中。

(2) 命令寄存器 (PRCMD)

命令寄存器(PRCMD)用于保护那些会严重影响系统运行状态的寄存器不被轻易写入，从而使应用系统不会因为程序挂起而停止工作。除非数据先被写入 PRCMD，否则向特殊寄存器中写入的数据是无效的。数据只有通过特定的流程才可能被写入特殊寄存器，这样就能够防止对特殊寄存器的非法写操作。
只能对 PRCMD 进行 8 位写操作。(若对其进行读取，则所读数据不确定)



(3) 系统状态寄存器 (SYS)

该寄存器中包含了用于指定整个系统的操作状态的状态标志位。
可以使用 8 位或 1 位的读或写操作对该寄存器进行访问。
该寄存器复位后被置为 00H。

复位后: 00H		R/W	地址: FFFFF802H					
SYS	7	6	5	4	3	2	1	<0>
	0	0	0	0	0	0	0	PRERR
PRERR		检测保护错误						
0		未发生保护错误						
1		发生了保护错误						

PRERR 标志会在以下情况下改变。

(a) 置位条件 (PRERR 标志 = 1)

- (i) 当向特殊寄存器写入数据之前没有对 PRCMD 寄存器进行写入时 (在执行 3.4.8 (1) 向特殊寄存器中写入数据中的步骤<4>之前没有执行步骤<3>)
- (ii) 当对 PRCMD 寄存器进行写入操作后，没有进行对特殊寄存器的写入，而是写入了片上周边 I/O 寄存器时 (如果 3.4.8 (1) 向特殊寄存器中写入数据中的步骤<4>不用于设置特殊寄存器)。

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

(b) 清零条件 (PRERR 标志 = 0)

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

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

3.4.8 注意事项

(1) 需要首先设置的寄存器

在使用 V850ES/JG2 器件时，请务必首先对以下寄存器进行设置。

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

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

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

(a) 系统等待寄存器 (VSWC)

VSWC 寄存器用于控制总线访问片上外围 I/O 寄存器的等待时间。

访问一个片上周边 I/O 寄存器需要 3 个时钟周期的时间(不包括一个等待周期)。V850ES/JG2 需要根据不同的操作频率插入不同的等待周期数。因此，请根据所使用的时钟，将以下对应的数值设置到 VSWC 寄存器中。

可以对 VSWC 寄存器进行 8 位的读写操作。(地址: FFFFF06EH, 默认值: 77H).

操作频率 (fCLK)	VSWC 的值	等待周期数
$32 \text{ kHz} \leq \text{fCLK} < 16.6 \text{ MHz}$	00H	0 (无等待周期)
$16.6 \text{ MHz} \leq \text{fCLK} \leq 20 \text{ MHz}$	01H	1

(b) 片上调试模式寄存器 (OCDM)

详见第 27 章 片上调试功能。

(c) 看门狗定时器模式寄存器 2 (WDTM2)

WDTM2 寄存器用于设置看门狗定时器 2 的溢出时间和操作时钟。

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

详见第 11 章 看门狗定时器 2 的功能。

(2) 访问特定的片上周边 I/O 寄存器

本产品具有两种内部系统总线。

一种是 CPU 总线，另一种是面向低速周边硬件的周边总线。

CPU 总线和周边总线的时钟是异步的。如果发生 CPU 访问和周边硬件访问的冲突，那么总线上传输的数据就可能是错误的非法数据。因此，在有可能发生总线冲突的情况下，对周边硬件进行访问时，访问 CPU 的时钟周期数量会发生改变，这样就保证了访问所传输数据的正确性。所以，CPU 不会进行对下一条指令的处理，而是进入等待状态。如果产生了这种等待状态，执行一条指令的所需的时间就会因插入了等待周期(如下所示)而增加。

对于那些对实时处理要求较高的应用来说，一定要考虑上述的情况。

当访问片上周边 I/O 寄存器时，插入的等待周期数可能要多于 VSWC 寄存器所设置的数量。

访问的动作内容和如何计算所插入的等待周期数量(CPU 时钟数)的方法如下所示。

周边功能	寄存器名称	访问	k
16 位定时器/事件计数器 P (TMP) (n = 0 到 5)	TPnCNT	读	1 或 2
	TPnCCR0, TPnCCR1	写	<ul style="list-style-type: none"> 第一次访问: 无等待 连续写入: 3 或 4
		读	1 或 2
16 位定时器/事件计数器 Q (TMQ)	TQ0CNT	读	1 或 2
	TQ0CCR0 到 TQ0CCR3	写	<ul style="list-style-type: none"> 第一次访问: 无等待 连续写入: 3 或 4
		读	1 或 2
看门狗定时器 2 (WDT2)	WDTM2	写 (WDT2 工作时)	3
实时输出功能 (RTO)	RTBL0	写 (RTPC0.RTPOE0 位 = 0)	1
	RTBH0	写 (RTPC0.RTPOE0 位 = 0)	1
A/D 转换器	ADA0M0	读	1 或 2
	ADA0CR0 到 ADA0CR11	读	1 或 2
	ADA0CR0H 到 ADA0CR11H	读	1 或 2
I ² C00 到 I ² C02 ^{※1}	IICS0 到 IICS2	读	1

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

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

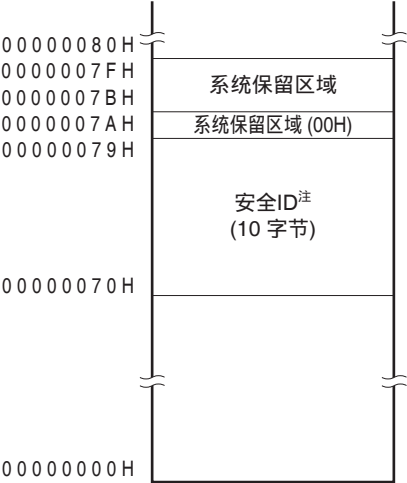
当主时钟振荡停止，系统工作于子时钟时。

当 CPU 工作于内部振荡器时钟时。

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

(3) 系统保留区域

V850ES/JG2 产品中，地址 0000007AH 到 0000007FH 是用于功能扩展的系统保留区域，因此建议不要使用该区域。



注 安全 ID 的详情请参考 27.3.1 安全 ID.

注意事项 如果 flash 存储器中的数据被擦除，那么所有的数据位都将被置 1。

(4) sld 指令与中断请求的冲突限制

(a) 描述

如果在指令<2>之后紧接着下述的指令<1>，在指令<1>操作执行完成之前，有中断请求和指令<2>的解码操作发生冲突，那么指令<1>的执行结果将有可能无法保存到寄存器中。

指令<1>

- ld 指令: ld.b, ld.h, ld.w, ld.bu, ld.hu
- sld 指令: sld.b, sld.h, sld.w, sld.bu, sld.hu
- 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

<例>

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

(b) 解决方法

<1> 当使用编译器(CA850)时

请使用 CA850 Ver. 2.61 或更高版本，因为这些版本的编译器可以自动避免生成上述可能冲突的指令序列。

<2> 当使用汇编器时

在指令<ii>之后紧接着执行 sld 指令时，可以通过以下的两种方式避免上述的问题。

- 在 sld 指令之前插入一条 nop 指令
- 在 sld 之前的上述指令<ii>中，不要使用 sld 指令中出现的目的寄存器。

第四章 端口功能

4.1 特点

- I/O 端口: 84
 - 5 V 容限/N 沟道开漏输出可选: 40 (端口 0, 3 至 5, 9)
- 输入或者输出可以以位为单位指定

4.2 端口的基本配置

V850ES/JG2 由端口 0, 1, 3 至 5, 7, 9, CM, CT, DH,以及 DL 共 84 个 I/O 端口组成。端口配置情况如下图所示

图 4-1. 端口配置图

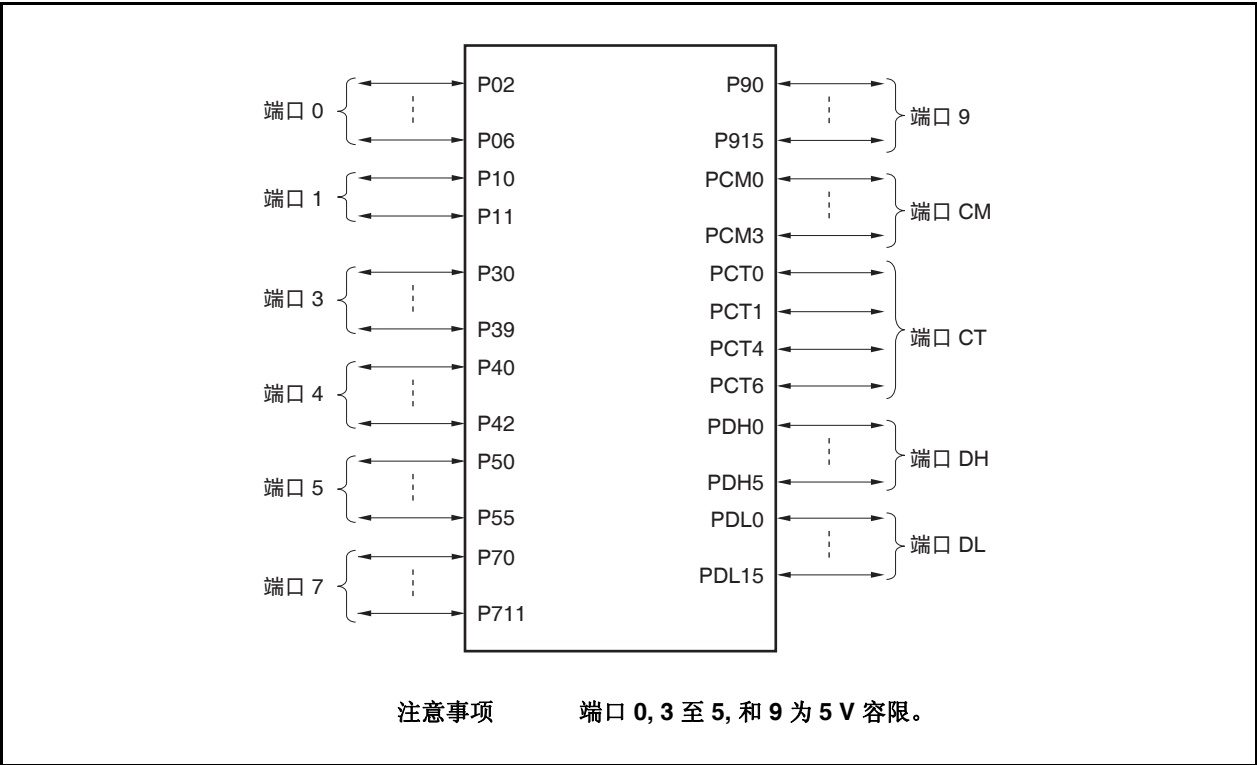


表 4-1. 引脚的 I/O 缓冲器电源

电源	对应引脚
AVREF0	端口 7
AVREF1	端口 1
BVDD	端口 CM, CT, DH (位 0 至 3), DL
EVDD	RESET, 端口 0, 3 至 5, 9, DH (位 4, 5)

4.3 端口配置

表 4-2. 端口配置

项目	配置
控制寄存器	端口 n 模式寄存器 (PMn: n = 0, 1, 3 至 5, 7, 9, CD, CM, CT, DH, DL) 端口 n 模式控制寄存器 (PMCn: n = 0, 3 至 5, 9, CM, CT, DH, DL) 端口 n 功能控制寄存器 (PFCn: n = 0, 3 至 5, 9) 端口 n 功能控制扩展寄存器 (PFCEn: n = 3, 5, 9) 端口 n 功能寄存器 (PFn: n = 0, 3 至 5, 9)
端口	I/O: 84

(1) 端口 n 寄存器 (Pn)

通过读/写 Pn 寄存器，将数据从外部设备输入或输出到外部设备。

Pn 寄存器由保持输出数据的端口锁存器和读引脚状态的电路组成。

Pn 寄存器的每一位都与端口 n 的一个引脚对应。该寄存器可以以位为单位读取或写入。

复位后: 00H 输出锁存)								R/W
Pn	7	6	5	7	3	2	1	0
	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0

Pnm	输出数据的控制 (输出模式下)
0	输出 0.
1	输出 1.

无论 PMCn 寄存器的设置如何，从 Pn 寄存器读写数据如下

表 4-3. 读取/写入 Pn 寄存器

PMn 寄存器的设置	向 Pn 寄存器写入	从 Pn 寄存器读取
输出模式 (PMnm = 0)	数据写入输出锁存器 [※] 。 在端口模式下(PMCn = 0)，输出锁存器的内容从引脚输出。	读取输出锁存的值。
输入模式 (PMnm = 1)	数据写入输出锁存器。引脚状态不受影响 [※] 。	读取引脚状态。

注 除非有新的值写入，否则写到输出锁存的值会一直保持。

(2) 端口 n 模式寄存器 (PMn)

PMn 寄存器指定对应端口引脚的输入或输出模式。
此寄存器的每一位都与端口 n 的一个引脚对应，可以以位为单位指定输入输出模式。

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

(3) 端口 n 模式控制寄存器 (PMCn)

PMCn 寄存器指定端口模式或复用功能。
此寄存器的每一位都与端口 n 的一个引脚对应，可以以位为单位指定端口模式。

复位后: 00H R/W								
	7	6	5	4	3	2	1	0
PMCn	PMCn7	PMCn6	PMCn5	PMCn4	PMCn3	PMCn2	PMCn1	PMCn0
PMCnm	指定操作模式							
0	端口模式							
1	复用功能模式							

(4) 端口 n 功能控制寄存器 (PFCn)

如果引脚有两个复用功能，PFCn 寄存器指定要使用引脚的复用功能。

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

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

(5) 端口 n 功能控制扩展寄存器 (PFCEn)

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

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

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

(6) 端口 n 功能寄存器 (PFn)

PFn 寄存器指定正常输出或 N 沟道开漏输出。
此寄存器的每一位都与端口 n 的一个引脚对应，可以以位为单位指定端口引脚的输出模式。

复位后: 00H R/W

	7	6	5	4	3	2	1	0
PFn	PFn7	PFn6	PFn5	PFn4	PFn3	PFn2	PFn1	PFn0

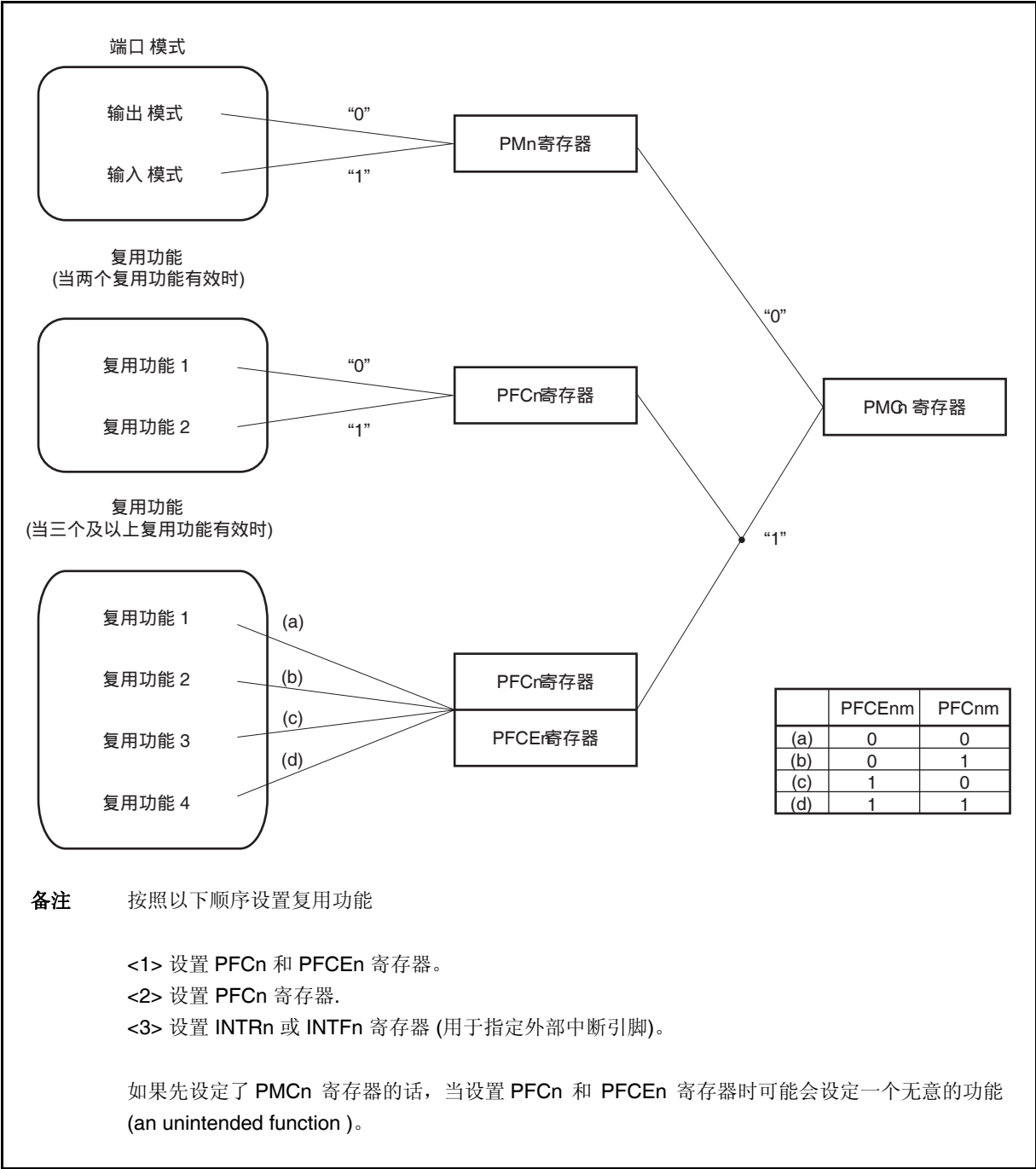
PFnm ^注	正常输出/开漏输出控制
0	正常输出 (CMOS输出)
1	N沟开漏输出

注 PFn 寄存器的 PFnm 位只有在端口模式下 (PMCnm 位 = 0) PMn 寄存器的 PMnm 位为 0 (指定为输出模式) 时有效。当 PMnm 位为 1 (指定为输入模式) 时，对 PFn 寄存器设置的值无效。

(7) 端口设置

按照下图设置端口

图 4-2. 每个寄存器和引脚功能的设置



4.3.1 端口 0

端口 0 是一个可以以位为单位控制 I/O 设置的 5 位端口。
端口 0 包含以下复用功能引脚。

表 4-4. 端口 0 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
P02	19	17	NMI	输入	可选作 N 沟道开漏输出	L-1
P03	20	18	INTP0/ADTRG	输入		N-1
P04	21	19	INTP1	输入		L-1
P05	22	20	INTP2/DRST ^注	输入		AA-1
P06	23	21	INTP3	输入		L-1

注 DRST 引脚用作片上调试。
如果不使用片上调试，从 RESET 引脚复位释放到 OCDM.OCDM0 位清零期间，需要将 P05/INTP2/DRST 引脚固定为低电平。
详细内容请参考 4.6.3 片上调试引脚的注意事项。

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

备注 GF: 100 引脚塑封 QFP (14 × 20)
GC: 100 引脚塑封 LQFP (密间距) (14 × 14)

(1) 端口 0 寄存器 (P0)

复位后: 00H (output latch) R/W 地址: FFFFF400H

	7	6	5	4	3	2	1	0
P0	0	P06	P05	P04	P03	P02	0	0

P0n	输出数据控制 (输出模式) (n = 2到6)
0	输出 0
1	输出 1

(2) 端口 0 模式寄存器 (PM0)

复位后: FFH R/W 地址: FFFFF420H

	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	1	1

PM0n	I/O模式控制 (n = 2到6)
0	输出模式
1	输入模式

(3) 端口 0 模式控制寄存器 (PMC0)

复位后: 00H R/W 地址: FFFFF440H

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

PMC06	指定P06引脚的操作模式
0	I/O端口
1	INTP3输入

PMC05	指定P05引脚的操作模式
0	I/O端口
1	INTP2输入

PMC04	指定P04引脚的操作模式
0	I/O端口
1	INTP1输入

PMC03	指定P03引脚的操作模式
0	I/O端口
1	INTP0输入/ADTRG输入

PMC02	指定P02引脚的操作模式
0	I/O端口
1	NMI输入

注意事项 当 OCDM 时无论 PMC05 位的值如何, P05/INTP2/ $\overline{\text{DRST}}$ 引脚都会变为 $\overline{\text{DRST}}$ 引脚。
OCDM0 位 = 1。

(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

PFC03	指定P03引脚的复用功能
0	INTP0输入
1	ADTRG输入

(5) 端口 0 功能寄存器 (PF0)

复位后: 00H R/W 地址: FFFFFC60H

	7	6	5	4	3	2	1	0
PF0	0	PF06	PF05	PF04	PF03	PF02	0	0

PF0n	正常输出/N沟开漏输出控制 (n = 2到6)
0	正常输出 (CMOS 输出)
1	N沟开漏输出

注意事项：如果一个输出引脚被拉高至 **EV_{DD}** 或更高，请务必设置 **PF0n** 位为 **1**。

4.3.2 端口 1

端口 1 是一个可以以位为单位控制 I/O 设置的 2 位端口。

端口 1 包含以下复用功能引脚。

表 4-5. 端口 1 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
P10	5	3	ANO0	输出	—	A-2
P11	6	4	ANO1	输出	—	A-2

注意事项 当电源打开时，即使在复位期间，P10 和 P11 引脚也可能会暂时输出不定的电平。

备注 GF: 100 引脚塑封 QFP (14 × 20)

GC: 100 引脚塑封 QFP (密间距) (14 × 14)

(1) 端口 1 寄存器 (P1)

复位后: 00H (output latch) R/W 地址: FFFFF402H

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

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

注意事项 在 D/A 转换期间不要对 P1 寄存器进行读或写的操作（参考 14.4.3 注意事项）。

(2) 端口 1 模式寄存器 (PM1)

复位后: FFH R/W 地址: FFFFF422H

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

PM1n	I/O模式控制 (n = 0, 1)
0	输出模式
1	输入模式

- 注意事项**
1. 当使用 P1n 作为复用功能 (ANOn 引脚输出) 时，需要将 PM1n 位置 1。
 2. 当使用 P10 和 P11 其中一个引脚作为 I/O 端口，另一个作为 D/A 输出引脚时，当进行 AD 输出的时候不能改变 I/O 端口的输出电平。

4.3.3 端口 3

端口 3 为可以以位为单位控制 I/O 设置的 10 位 端口。

端口 3 包含以下复用功能引脚。

表 4-6. 端口 3 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
P30	27	25	TXDA0/SOB4	输出	可选作 N 沟道开漏输出	G-3
P31	28	26	RXDA0/INTP7/SIB4	输入		N-3
P32	29	27	ASCKA0/SCKB4/TIP00/TOP00	I/O		U-1
P33	30	28	TIP01/TOP01	I/O		G-1
P34	31	29	TIP10/TOP10	I/O		G-1
P35	32	30	TIP11/TOP11	I/O		G-1
P36	33	31	-	-		C-1
P37	34	32	-	-		C-1
P38	37	35	TXDA2/SDA00	I/O		G-12
P39	38	36	RXDA2/SCL00	I/O		G-6

注意事项 P31 至 P35，P38 和 P39 引脚在复用功能的输入模式具有滞后性（**hysteresis characteristics**），但在端口模式没有滞后性。

备注 GF: 100 引脚塑封 QFP (14 × 20)
GC: 100 引脚塑封 LQFP (密间距) (14 × 14)

(1) 端口 3 寄存器 (P3)

复位后: 0000H (output latch)		R/W	地址:		P3 FFFFF406H, P3L FFFFF406H, P3H FFFFF407H			
	15	14	13	12	11	10	9	8
P3 (P3H)	0	0	0	0	0	0	P39	P38
	7	6	5	4	3	2	1	0
(P3L)	P37	P36	P35	P34	P33	P32	P31	P30
P3n	输出数据控制 (输出模式下) (n = 0到9)							
0	输出 0							
1	输出 1							

- 备注**
1. 可以以 16 位为单位读写 P3 寄存器。
此外当使用 P3 寄存器的高 8 位作为 P3H 寄存器，低 8 位作为 P3L 寄存器时，P3 可以以 8 位或 1 位为单位读/写。
 2. 要以 8 位 或 1 位为单位读/写 P3 寄存器的位 8 至 15，需要将他们指定为 P3H 寄存器的位 0 至 7。

(2) 端口 3 模式寄存器 (PM3)

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

- 备注**
1. 可以以 16 位为单位读/写 PM3 寄存器。
此外当使用 PM3 寄存器的高 8 位作为 PM3H 寄存器，低 8 位作为 PM3L 寄存器时，PM3 可以以 8 位或 1 位为单位读/写。
 2. 要以 8 位 或 1 位为单位读/写 PM3 寄存器的位 8 至 15，需要将他们指定为 PM3H 寄存器的位 0 至 7。

(3) 端口 3 模式控制寄存器 (PMC3)

复位后: 0000H R/W 地址: PMC3FFFF446H

注意事项 务必把第 15-10, 7, 和 6 位清零

备注

1. 可以以 16 位为单位读/写 PMC3 寄存器。
此外当使用 PMC3 寄存器的高 8 位作为 PMC3H 寄存器, 低 8 位作为 PMC3L 寄存器时, PMC3 可以以 8 位或 1 位为单位读/写。
2. 要以 8 位 或 1 位为单位读/写 PMC3 寄存器的位 8 至 15 , 需要将他们指定为 PMC3H 寄存器的位 0 至 7。

(4) 端口 3 功能控制寄存器 (PFC3)

复位后: 0000H		R/W		地址: PFC 3FFFFFF466H PFC3L FFFFFFF466H PFC3H FFFFFFF467				
PFC3 (PFC 3H)	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	PFC 39	PFC 38
(PFC 3L)	7	6	5	4	3	2	1	0
	0	0	PFC 35	PFC 34	PFC 33	PFC 32	PFC 31	PFC 30

- 备注**
1. 有关设定复用功能的详细信息请参考 **4.3.3 (6) 端口 3 复用功能设定**。
 2. 可以以 16 位为单位读/写 PFC3 寄存器。
此外当使用 PFC3 寄存器的高 8 位作为 PFC3H 寄存器，低 8 位作为 PFC3L 寄存器时，PFC3 可以以 8 位或 1 位为单位读/写。
 3. 要以 8 位 或 1 位为单位读/写 PFC3 寄存器的位 8 至 15，需要将他们指定为 PFC3H 寄存器的位 0 至 7。

(5) 端口 3 功能控制扩展寄存器 L (PFCE3L)

复位后: 00H		R/W		地址: FFFFFFF706H				
PFCE3L	7	6	5	4	3	2	1	0
	0	0	0	0	0	PFCE32	0	0

- 备注** 有关设定复用功能的详细信息请参考 **4.3.3 (6) 端口 3 复用功能设定**。。

(6) 端口 3 复用功能设定

PFC39	P39 引脚复用功能的设定
0	RXDA2 输入
1	SCL00 I/O

PFC38	P38 引脚复用功能的设定
0	TXDA2 输出
1	SDA00 I/O

PFC35	P35 引脚复用功能的设定
0	TIP11 输入
1	TOP11 输出

PFC34	P34 引脚复用功能的设定
0	TIP10 输入
1	TOP10 输出

PFC33	P33 引脚复用功能的设定
0	TIP01 输入
1	TOP01 输出

PFCE32	PFC32	P32 引脚复用功能的设定
0	0	ASCKA0 输入
0	1	SCKB4 I/O
1	0	TIP00 输入
1	1	TOP00 输出

PFC31	P31 引脚复用功能的设定
0	RXDA0 输入/INTP7 [※] 输入
1	SIB4 输入

PFC30	P30 引脚复用功能的设定
0	TXDA0 输出
1	SOB4 输出

注 INTP7 引脚和 RXDA0 引脚为一组复用功能引脚。当用作 RXDA0 引脚时，需要禁止 INTP7 复用功能引脚的边沿检测 (将 INTF3.INTF31 位和 INTR3.INTR31 位清 0)。当用作 INTP7 引脚时，需要停止 UARTA0 接收 (将 UA0CTL0.UA0RXE 位清 0)。

(7) 端口 3 功能寄存器 (PF3)

复位后: 0000H		R/W	地址:		PF3 FFFFFFFC66H, PF3L FFFFFFFC66H, PF3H FFFFFFFC67H			
PF3 (PF3H)	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	PF39	PF38
(PF3L)	7	6	5	4	3	2	1	0
	PF37	PF36	PF35	PF34	PF33	PF32	PF31	PF30
PF3n	正常输出/N沟开漏输出控制 (n = 0到9)							
0	正常输出 (CMOS输出)							
1	N沟开漏输出							

注意事项 如果一个输出引脚被拉高至 EV_{DD} 或更高，请务必设置 **PF3n** 位为 1

- 备注**
1. 可以以 16 位为单位读/写 PF3 寄存器。
此外，当使用 PF3 寄存器的高 8 位作为 PF3H 寄存器，低 8 位作为 PF3L 寄存器时，PF3 可以以 8 位或 1 位为单位读/写
 2. 要以 8 位或 1 位为单位读/写 PF3 寄存器的位 8 至 15，需要将他们指定为 PF3H 寄存器的位 0 至 7。

4.3.4 端口 4

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

表 4-7. 端口 4 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
P40	24	22	SIB0/SDA01	I/O	可选作 N 沟道开漏输出	G-6
P41	25	23	SOB0/SCL01	I/O		G-12
P42	26	24	SCKB0	I/O		E-3

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

备注 GF: 100 引脚塑封 QFP (14 × 20)
GC: 100 引脚塑封 LQFP (密间距) (14 × 14)

(1) 端口 4 寄存器 (P4)

复位后: 00H (输出锁存) R/W 地址: FFFF408H								
P4	7	6	5	4	3	2	1	0
	0	0	0	0	0	P42	P41	P40
P4n	输出数据控制 (输出模式下) (n = 0到2)							
0	输出 0							
1	输出 1							

(2) 端口 4 模式寄存器 (PM4)

复位后: FFH R/W 地址: FFFFF428H								
PM4	7	6	5	4	3	2	1	0
	1	1	1	1	1	PM42	PM41	PM40
PM4n	I/O模式控制 (n = 0到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	指定P42引脚的操作模式
0	I/O 端口
1	SCKB0 I/O

PMC41	指定P41引脚的操作模式
0	I/O 端口
1	SOB0 输出/SCL01 I/O

PMC40	指定P40引脚的操作模式
0	I/O 端口
1	SIB0 输入/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	0	PFC41	PFC40

PFC41	指定P41引脚的复用功能
0	SOB0 输出
1	SCL01 I/O

PFC40	指定P40引脚的复用功能
0	SIB0 输入
1	SDA01 I/O

(5) 端口 4 功能寄存器 (PF4)

复位后: 00H R/W 地址: FFFFFC68H

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

PF4n	正常输出/N沟开漏输出控制 (n = 0到2)
0	正常输出 (CMOS 输出)
1	N沟开漏输出

注意事项 如果一个输出引脚被拉高至 EV_{DD} 或更高, 请务必设置 PF4n 位为 1

4.3.5 端口 5

端口 5 为可以以位为单位控制 I/O 设置的 6 位 端口。

端口 5 包含以下复用功能引脚。

表 4-8. 端口 5 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
P50	39	37	TIQ01/KR0/TOQ01/RTP00	I/O	可选作 N 沟道开漏输出	U-5
P51	40	38	TIQ02/KR1/TOQ02/RTP01	I/O		U-5
P52	41	39	TIQ03/KR2/TOQ03/RTP02/DDI ^注	I/O		U-6
P53	42	40	SIB2/KR3/TIQ00/TOQ00/RTP03/DDO ^注	I/O		U-7
P54	43	41	SOB2/KR4/RTP04/DCK ^注	I/O		U-8
P55	44	42	SCKB2/KR5/RTP05/DMS ^注	I/O		U-9

注 DDI, DDO, DCK, 和 DMS 引脚用作片上调试。

如果不使用片上调试功能，从 RESET 引脚复位释放到 OCDM.OCDM0 位清 (0)期间，需要将 P05/INTP2/DRST 引脚固定为低电平。

详细信息请参考 4.6.3 片上调试引脚的注意事项。

注意事项 1. 当电源打开时，即使在复位期间，P53 引脚也可能会暂时输出不定的电平。

2. P50 至 P55 引脚在复用功能的输入模式具有滞后性（hysteresis characteristics），但在端口模式没有滞后性。

备注 GF: 100 引脚塑封 QFP (14 × 20)

GC: 100 引脚塑封 LQFP (密间距) (14 × 14)

(1) 端口 5 寄存器 (P5)

复位后: 00H (output latch) R/W 地址: FFFFF40AH

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

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

(2) 端口 5 模式寄存器 (PM5)

复位后: FFH R/W 地址: FFFFF42AH

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

PM5n	I/O模式控制 (n = 0到5)
0	输出模式
1	输入模式

(3) 端口 5 模式控制寄存器 (PMC5)

复位后: 00H R/W 地址: FFFFF44AH

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

PMC55	指定P55引脚的操作模式
0	I/O 端口
1	SCKB2 I/O/KR5 输入/RTP05 输出

PMC54	指定P54引脚的操作模式
0	I/O 端口
1	SOB2 输出/KR4 输入/RTP04 输出

PMC53	指定P53引脚的操作模式
0	I/O 端口
1	SIB2 输入/KR3 输入/TIQ00 输入/TOQ00 输出/RTP03 输出

PMC52	指定P52引脚的操作模式
0	I/O 端口
1	TIQ03 输入/KR2 输入/TOQ03 输出/RTP02 输出

PMC51	指定P51引脚的操作模式
0	I/O 端口
1	TIQ02 输入/KR1 输入/TOQ02 输出/RTP01 输出

PMC50	指定P50引脚的操作模式
0	I/O 端口
1	TIQ01 输入/KR0 输入/TOQ01 输出/RTP00 输出

(4) 端口 5 功能控制寄存器 (PFC5)

复位后: 00H R/W 地址: FFFFF46AH

	7	6	5	4	3	2	1	0
PFC5	0	0	PFC55	PFC54	PFC53	PFC52	PFC51	PFC50

备注 有关指定复用功能的详细内容, 请参考 4.3.5 (6) 端口 5 复用功能指定。

(5) 端口 5 功能控制扩展寄存器 (PFCE5)

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

备注 有关指定复用功能的详细内容, 请参考 4.3.5 (6) 端口 5 复用功能指定。

(6) 端口 5 复用功能指定

PFCE55	PFC55	P55 引脚复用功能的指定
0	0	SCKB2 I/O
0	1	KR5 输入
1	0	禁止设置
1	1	RTP05 输出

PFCE54	PFC54	P54 引脚复用功能的指定
0	0	SOB2 输出
0	1	KR4 输入
1	0	禁止设置
1	1	RTP04 输出

PFCE53	PFC53	P53 引脚复用功能的指定
0	0	SIB2 输入
0	1	TIQ00 输入/KR3 [※] 输入
1	0	TOQ00 输出
1	1	RTP03 输出

PFCE52	PFC52	P52 引脚复用功能的指定
0	0	禁止设置
0	1	TIQ03 输入/KR2 [※] 输入
1	0	TOQ03 输入
1	1	RTP02 输出

PFCE51	PFC51	P51 引脚复用功能的指定
0	0	禁止设置
0	1	TIQ02 输入/KR1 [※] 输入
1	0	TOQ02 输出
1	1	RTP01 输出

PFCE50	PFC50	P50 引脚复用功能
0	0	禁止设置
0	1	TIQ01 输入/KR0 [※] 输入
1	0	TOQ01 输出
1	1	RTP00 输出

注 KRn 引脚和 TIQ0m 引脚为一组复用功能引脚。当作为 TIQ0m 引脚使用时，需要禁止复用功能 KRn 引脚的按键返回检测 (将 KRM.KRMn 位清 0)，当作为 KRn 引脚使用时，需要禁止复用功能 TIQ0m 引脚的边沿检测(n = 0 至 3, m = 0 至 3)。

引脚名称	用作 TIQ0m 引脚	用作 KRn 引脚
KR0/TIQ01	KRM.KRM0 位 = 0	TQ0IOC1. TQ0TIG2, TQ0IOC1. TQ0TIG3 位 = 0
KR1/TIQ02	KRM.KRM1 位 = 0	TQ0IOC1.TQ0TIG4, TQ0IOC1.TQ0TIG5 位 = 0
KR2/TIQ03	KRM.KRM2 位 = 0	TQ0IOC1.TQ0TIG6, TQ0IOC1.TQ0TIG7 位 = 0
KR3/TIQ00	KRM.KRM3 位 = 0	TQ0IOC1.TQ0TIG0, TQ0IOC1.TQ0TIG1 位 = 0 TQ0IOC2.TQ0EES0, TQ0IOC2.TQ0EES1 位 = 0 TQ0IOC2.TQ0ETS0, TQ0IOC2.TQ0ETS1 位 = 0

(7) 端口 5 功能寄存器 (PF5)

复位后: 00H R/W 地址: FFFFFC6AH

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

PF5n	正常输出/N沟开漏输出控制 (n = 0到5)
0	正常输出(CMOS输出)
1	N沟开漏输出

注意事项 如果一个输出引脚被拉高至 EV_{DD} 或更高，请务必设置 PF5n 位为 1

4.3.6 端口 7

端口 7 为可以以位为单位控制 I/O 设置的 12 位端口。
端口 7 包含以下复用功能引脚.

表 4-9. 端口 7 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
P70	2	100	ANIO	输入	—	A-1
P71	1	99	ANI1	输入		A-1
P72	100	98	ANI2	输入		A-1
P73	99	97	ANI3	输入		A-1
P74	98	96	ANI4	输入		A-1
P75	97	95	ANI5	输入		A-1
P76	96	94	ANI6	输入		A-1
P77	95	93	ANI7	输入		A-1
P78	94	92	ANI8	输入		A-1
P79	93	91	ANI9	输入		A-1
P710	92	90	ANI10	输入		A-1
P711	91	89	ANI11	输入		A-1

备注 GF: 100 引脚塑封 QFP (14 × 20)
GC: 100 引脚塑封 LQFP (密间距) (14 × 14)

(1) 端口 7 寄存器 H, 端口 7 寄存器 L (P7H, P7L)

复位后: 00H (output latch) R 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到11)
0	输出 0
1	输出 1

注意事项 在 A/D 转换期间不要读取 P7H 和 P7L 寄存器。(参考 13.6(4) I/O 复用)**备注** 不能以 16 位为单位访问 P7 寄存器。只能作为 P7H 和 P7L 寄存器以 8 位或 1 位为单位读/写。

(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	I/O模式控制 (n = 0到11)
0	输出模式
1	输入模式

注意事项 当使用 P7n 引脚的复用功能(ANIn pin)时, 需要将 PM7n 位置 1。**备注** 不能以 16 位为单位访问 PM7 寄存器。只能作为 PM7H 和 PM7L 寄存器以 8 位或 1 位为单位读/写。

4.3.7 端口 9

端口 9 为可以以位为单位控制 I/O 设置的 16 位端口。

端口 9 包含以下复用功能引脚。

表 4-10. 端口 9 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
P90	45	43	A0/KR6/TXDA1/SDA02	I/O	可选作 N 沟道开漏输出	U-10
P91	46	44	A1/KR7/RXDA1/SCL02	I/O		U-11
P92	47	45	A2/TIP41/TOP41	I/O		U-12
P93	48	46	A3/TIP40/TOP40	I/O		U-12
P94	49	47	A4/TIP31/TOP31	I/O		U-12
P95	50	48	A5/TIP30/TOP30	I/O		U-12
P96	51	49	A6/TIP21/TOP21	I/O		U-13
P97	52	50	A7/SIB1/TIP20/TOP20	I/O		U-14
P98	53	51	A8/SOB1	输出		G-3
P99	54	52	A9/ $\overline{\text{SCKB1}}$	I/O		G-5
P910	55	53	A10/SIB3	I/O		G-2
P911	56	54	A11/SOB3	输出		G-3
P912	57	55	A12/ $\overline{\text{SCKB3}}$	I/O		G-5
P913	58	56	A13/INTP4	I/O		N-2
P914	59	57	A14/INTP5/TIP51/TOP51	I/O		U-15
P915	60	58	A15/INTP6/TIP50/TOP50	I/O		U-15

注意事项 P90 至 P915 引脚在复用功能的输入模式具有滞后性（**hysteresis characteristics**），但在端口模式没有滞后性。

备注 GF: 100 引脚塑封 QFP (14 × 20)
GC: 100 引脚塑封 LQFP (密间距) (14 × 14)

(1) 端口 9 寄存器 (P9)

复位后: 0000H (output latch)		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) (n = 0到15)						
0		输出 0						
1		输出 1						

- 备注**
1. 可以以 16 位为单位读/写 P9 寄存器。
此外, 当使用 P9 寄存器的高 8 位作为 P9H 寄存器, 低 8 位作为 P9L 寄存器时, P9 可以以 8 位或 1 位为单位读/写。
 2. 要以 8 位 或 1 位为单位读/写 P9 寄存器的位 8 至 15 , 需要将他们指定为 P9H 寄存器的位 0 至 7。

(2) 端口 9 模式寄存器 (PM9)

复位后: FFFFFH		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	I/O模式控制 (n = 0到15)						
	0	输出模式						
	1	输入模式						

- 备注**
1. 可以以 16 位为单位读/写 PM9 寄存器。
此外, 当使用 PM9 寄存器的高 8 位作为 PM9H 寄存器, 低 8 位作为 PM9L 寄存器时, PM9 可以以 8 位或 1 位为单位读/写。
 2. 要以 8 位 或 1 位为单位读/写 PM9 寄存器的位 8 至 15 , 需要将他们指定为 PM9H 寄存器的位 0 至 7。

(1/2)

[illegible]

	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	指定P915引脚的操作模式
0	I/O 端口
1	A15 输出/INTP6 输入/TIP50 输入/TOP50 输出

PMC914	指定P914引脚的操作模式
0	I/O 端口
1	A14 输出/INTP5 输入/TIP51 输入/TOP51 输出

PMC913	指定P913引脚的操作模式
0	I/O 端口
1	A13 输出/INTP4 输入

PMC912	指定P912引脚的操作模式
0	I/O 端口
1	A12 输出/SCKB3 I/O

PMC911	指定P911引脚的操作模式
0	I/O 端口
1	A11 输出/SOB3 输出

PMC910	指定P910引脚的操作模式
0	I/O 端口
1	A10 输出/SIB3 输入

PMC99	指定P99引脚的操作模式
0	I/O 端口
1	A9 输出/ $\overline{\text{SCKB1}}$ I/O

备注 1. 可以以 16 位为单位读/写 PMC9 寄存器。

- 此外, 当使用 PMC9 寄存器的高 8 位作为 PMC9H 寄存器, 低 8 位作为 PMC9L 寄存器时, PMC9 可以以 8 位或 1 位为单位读/写。
2. 要以 8 位或 1 位为单位读/写 PMC9 寄存器的位 8 至 15, 需要将他们指定为 PMC9H 寄存器的位 0 至 7。

PMC98	指定P98引脚的操作模式
0	I/O 端口
1	A8 输出/SOB1 输出
PMC97	指定P97引脚的操作模式
0	I/O 端口
1	A7 输出/SIB1 输入/TIP20 输入/TOP20 输出
PMC96	指定P96引脚的操作模式
0	I/O 端口
1	A6 输出/TIP21 输入/TOP21 输出
PMC95	指定P95引脚的操作模式
0	I/O 端口
1	A5 输出/TIP30 输入/TOP30 输出
PMC94	指定P94引脚的操作模式
0	I/O 端口
1	A4 输出/TIP31 输入/TOP31 输出
PMC93	指定P93引脚的操作模式
0	I/O 端口
1	A3 输出/TIP40 输入/TOP40 输出
PMC92	指定P92引脚的操作模式
0	I/O 端口
1	A2 输出/TIP41 输入/TOP41 输出
PMC91	指定P91引脚的操作模式
0	I/O 端口
1	A1 输出/KR7 输入/RXDA1 输入/SCL02 I/O
PMC90	指定P90引脚的操作模式
0	I/O 端口
1	A0 输出/KR6 输入/TXDA1 输出/SDA02 I/O

注意事项 只有当 **P90** 至 **P915** 引脚全部用作复用功能 **A0** 至 **A15** 引脚时，需要立刻将 **PMC9** 寄存器的全部 **16** 位设置为 **FFFFH**。

(4) 端口 9 功能控制寄存器 (PFC9)

注意事项 当执行单独的地址总线输出 (A0 至 A15) 时, 需要在将 PFC9 寄存器清除为 0000H 后立刻将 PFC9 寄存器设置为 FFFFH。

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. 有关指定复用功能的详细内容, 请参考 4.3.7 (6) 端口 9 复用功能指定。
 2. 可以以 16 位为单位读/写 PFC9 寄存器。
此外, 当使用 PFC9 寄存器的高 8 位作为 PFC9H 寄存器, 低 8 位作为 PFC9L 寄存器时, PFC9 可以以 8 位或 1 位为单位读/写。
 3. 要以 8 位或 1 位为单位读/写 PFC9 寄存器的位 8 至 15, 需要将他们指定为 PFC9H 寄存器的位 0 至 7。

(5) 端口 9 功能控制扩展寄存器 (PFCE9)

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	0	0	0	0			
			7	6	5	4	3	2	1	0			
(PFCE9L)			PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90			

- 备注**
1. 有关指定复用功能的详细内容, 请参考 4.3.7 (6) 端口 9 复用功能指定。
 2. 可以以 16 位为单位读/写 PFCE9 寄存器。
此外, 当使用 PFCE9 寄存器的高 8 位作为 PFCE9H 寄存器, 低 8 位作为 PFCE9L 寄存器时, PFCE9 可以以 8 位或 1 位为单位读/写。
 3. 要以 8 位或 1 位为单位读/写 PFCE9 寄存器的位 8 至 15, 需要将他们指定为 PFCE9H 寄存器的位 0 至 7。

(6) 端口 9 复用功能指定

PFCE915	PFC915	P915 引脚复用功能的指定
0	0	A15 输出
0	1	INTP6 输入
1	0	TIP50 输入
1	1	TOP50 输出

PFCE914	PFC914	P914 引脚复用功能的指定
0	0	A14 输出
0	1	INTP5 输入
1	0	TIP51 输入
1	1	TOP51 输出

PFC913	P913 引脚复用功能的指定
0	A13 输出
1	INTP4 输入

PFC912	P912 引脚复用功能的指定
0	A12 输出
1	SCKB3 I/O

PFC911	P911 引脚复用功能的指定
0	A11 输出
1	SOB3 输出

PFC910	P910 引脚复用功能的指定
0	A10 输出
1	SIB3 输入

PFC99	P99 引脚复用功能的指定
0	A9 输出
1	SCKB1 I/O

PFC98	P98 引脚复用功能的指定
0	A8 输出
1	SOB1 输出

PFCE97	PFC97	P97 引脚复用功能的指定
0	0	A7 输出
0	1	SIB1 输入
1	0	TIP20 输入
1	1	TOP20 输出

PFCE96	PFC96	P96 引脚复用功能的指定
0	0	A6 输出
0	1	禁止设置
1	0	TIP21 输入
1	1	TOP21 输出

PFCE95	PFC95	P95 引脚复用功能的指定
0	0	A5 输出
0	1	TIP30 输入
1	0	TOP30 输出
1	1	禁止设置

PFCE94	PFC94	P94 引脚复用功能的指定
0	0	A4 输出
0	1	TIP31 输入
1	0	TOP31 输出
1	1	禁止设置

PFCE93	PFC93	P93 引脚复用功能的指定
0	0	A3 输出
0	1	TIP40 输入
1	0	TOP40 输出
1	1	禁止设置

PFCE92	PFC92	P92 引脚复用功能的指定
0	0	A2 输出
0	1	TIP41 输入
1	0	TOP41 输出
1	1	禁止设置

PFCE91	PFC91	P91 引脚复用功能的指定
0	0	A1 输出
0	1	KR7 输入
1	0	RXDA1 输入/KR7 输入 ^注
1	1	SCL02 I/O

PFCE90	PFC90	P90 引脚复用功能的指定
0	0	A0 输出
0	1	KR6 输入
1	0	TXDA1 输出
1	1	SDA02 I/O

注 RXDA1 和 KR7 引脚切勿同时使用。使用 RXDA1 引脚就不要使用 KR7 引脚。当使用 KR7 引脚时，不要使用 RXDA1 引脚 (推荐将 PFC91 位置 1 以及将 PFCE91 位清 0)。

(7) 端口 9 功能寄存器 (PF9)

复位后: 0000H		R/W	地址:		PF3 FFFFFFFC72H, PF9L FFFFFFFC72H, PF9H FFFFFFFC73H			
PF9 (PF9H)	15	14	13	12	11	10	9	8
	PF915	PF914	PF913	PF912	PF911	PF910	PF99	PF98
(PF9L)	7	6	5	4	3	2	1	0
	PF97	PF96	PF95	PF94	PF93	PF92	PF91	PF90
PF9n		正常输出/N沟开漏输出控制 (n = 0到15)						
0		正常输出 (CMOS输出)						
1		N沟开漏输出						

注意事项 如果一个输出引脚被拉高 至 EV_{DD}或更高，请务必设置 PF9n 位为 1

- 备注
- 可以以 16 位为单位读/写 PF9 寄存器。
此外，当使用 PF9 寄存器的高 8 位作为 PF9H 寄存器，低 8 位作为 PF9L 寄存器时，PF9 可以以 8 位或 1 位为单位读/写。
 - 要以 8 位或 1 位为单位读/写 PF9 寄存器的位 8 至 15，需要将他们指定为 PF9H 寄存器的位 0 至 7。

4.3.8 端口 CM

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

表 4-11. 端口 CM 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
PCM0	63	61	WAIT	输入	-	D-1
PCM1	64	62	CLKOUT	输出		D-2
PCM2	65	63	HLD $\overline{\text{AK}}$	输出		D-2
PCM3	66	64	HLD $\overline{\text{RQ}}$	输入		D-1

备注 GF: 100 引脚塑封 QFP (14 × 20)
 GC: 100 引脚塑封 LQFP (密间距) (14 × 14)

(1) 端口 CM 寄存器 (PCM)

复位后: 00H (output latch)									R/W	地址: FFFFF00CH								
PCM	7	6	5	4	3	2	1	0										
	0	0	0	0	PCM3	PCM2	PCM1	PCM0										
PCMn		输出数据控制 (输出模式下) (n = 0到3)																
0		输出 0																
1		输出 1																

(2) 端口 CM 模式寄存器 (PMCM)

复位后: FFH R/W 地址: FFFF02CH								
PMCM	7	6	5	4	3	2	1	0
	1	1	1	1	PMCM3	PMCM2	PMCM1	PMCM0
PMCMn		I/O模式控制(n = 0到3)						
0		输出模式						
1		输入模式						

(3) 端口 CM 模式控制寄存器 (PMCCM)

复位后: 00H R/W 地址: FFFFF04CH

	7	6	5	4	3	2	1	0
PMCCM	0	0	0	0	PMCCM3	PMCCM2	PMCCM1	PMCCM0
PMCCM3	指定PCM3引脚的操作模式							
0	I/O 端口							
1	HLDRQ 输入							
PMCCM2	指定PCM2引脚的操作模式							
0	I/O 端口							
1	HLDAK 输出							
PMCCM1	指定PCM1引脚的操作模式							
0	I/O 端口							
1	CLKOUT 输出							
PMCCM0	指定PCM0引脚的操作模式							
0	I/O 端口							
1	WAIT 输入							

4.3.9 端口 CT

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

表 4-12. 端口 CT 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
PCT0	67	65	$\overline{\text{WR0}}$	输出	—	D-2
PCT1	68	66	$\overline{\text{WR1}}$	输出		D-2
PCT4	69	67	$\overline{\text{RD}}$	输出		D-2
PCT6	70	68	ASTB	输出		D-2

备注 GF: 100 引脚塑封 QFP (14 × 20)
 GC: 100 引脚塑封 LQFP (密间距) (14 × 14)

(1) 端口 CT 寄存器 (PCT)

复位后: 00H (output latch) R/W 地址: FFFFF00AH

	7	6	5	4	3	2	1	0
PCT	0	PCT6	0	PCT4	0	0	PCT1	PCT0

PCTn	输出数据控制 (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)

复位后: FFH R/W 地址: FFFFF02AH

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

PMCTn	I/O模式控制 (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)

复位后: 00H R/W 地址: FFFFF04AH

	7	6	5	4	3	2	1	0
PMCCT	0	PMCCT6	0	PMCCT4	0	0	PMCCT1	PMCCT0

PMCCT6	指定PCT6引脚的操作模式						
0	I/O 端口						
1	ASTB 输出						

PMCCT4	指定PCT4引脚的操作模式						
0	I/O 端口						
1	$\overline{\text{RD}}$ 输出						

PMCCT1	指定PCT1引脚的操作模式						
0	I/O 端口						
1	$\overline{\text{WR1}}$ 输出						

PMCCT0	指定PCT0引脚的操作模式						
0	I/O 端口						
1	$\overline{\text{WR0}}$ 输出						

4.3.10 端口 DH

端口 DH 为可以以位为单位控制 I/O 设置的 6 位端口。
端口 DH 包含以下复用功能引脚。

表 4-13. 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	I/O	备注	模块类型
	GF	GC				
PDH0	89	87	A16	输出	—	D-2
PDH1	90	88	A17	输出		D-2
PDH2	61	59	A18	输出		D-2
PDH3	62	60	A19	输出		D-2
PDH4	8	6	A20	输出		D-2
PDH5	9	7	A21	输出		D-2

备注 GF: 100 引脚塑封 QFP (14 × 20)
 GC: 100 引脚塑封 LQFP (密间距) (14 × 14)

(1) 端口 DH 寄存器 (PDH)

复位后: 00H (output latch) R/W 地址: FFFFF006H

	7	6	5	4	3	2	1	0
PDH	0	0	PDH5	PDH4	PDH3	PDH2	PDH1	PDH0

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

(2) 端口 DH 模式寄存器 (PMDH)

复位后: FFH R/W 地址: FFFFF026H								
PMDH	7	6	5	4	3	2	1	0
	1	1	PMDH5	PMDH4	PMDH3	PMDH2	PMDH1	PMDH0
PMDHn	I/O模式控制 (n = 0到5)							
0	输出模式							
1	输入模式							

PMCDHn	指定PDHn引脚的操作模式 (n = 0到5)
0	I/O 端口
1	Am 输出 (地址总线输出) (m = 16到21)

4.3.11 端口 DL

端口 DL 为可以以位为单位控制 I/O 设置的 16 位端口

端口 DL 包含以下复用功能引脚。

表 4-14. 端口 DL 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
PDL0	73	71	AD0	I/O	—	D-3
PDL1	74	72	AD1	I/O		D-3
PDL2	75	73	AD2	I/O		D-3
PDL3	76	74	AD3	I/O		D-3
PDL4	77	75	AD4	I/O		D-3
PDL5	78	76	AD5/FLMD1 ^注	I/O		D-3
PDL6	79	77	AD6	I/O		D-3
PDL7	80	78	AD7	I/O		D-3
PDL8	81	79	AD8	I/O		D-3
PDLDL	82	80	AD9	I/O		D-3
PDL10	83	81	AD10	I/O		D-3
PDL11	84	82	AD11	I/O		D-3
PDL12	85	83	AD12	I/O		D-3
PDL13	86	84	AD13	I/O		D-3
PDL14	87	85	AD14	I/O		D-3
PDL15	88	86	AD15	I/O		D-3

注 由于只需要在闪存编程模式下设置此引脚，因此不需要操作端口控制寄存器。详细信息请参考 **第 30 章 FLASH 存储器**。

备注 GF: 100 引脚塑封 QFP (14 × 20)
GC: 100 引脚塑封 LQFP (密间距) (14 × 14)

(1) 端口 DL 寄存器 (PDL)

复位后: 0000H (输出锁存)

R/W

地址:

PDL 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
(PDLH)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

PDLn	输出地址控制 (输出模式下) (n = 0到15)
0	输出 0
1	输出 1

- 备注**
1. 可以以 16 位为单位读/写 PDL 寄存器。
此外, 当使用 PDL 寄存器的高 8 位作为 PDLH 寄存器, 低 8 位作为 PDLH 寄存器时, PDL 可以以 8 位或 1 位为单位读/写。
 2. 要以 8 位或 1 位为单位读/写 PDL 寄存器的位 8 至 15 , 需要将他们指定为 PDLH 寄存器的位 0 至 7。

(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	I/O模式控制 (n = 0到15)						
	0	输出模式						
	1	输入模式						

- 备注**
1. 可以以 16 位为单位读/写 PMDL 寄存器。
此外, 当使用 PMDL 寄存器的高 8 位作为 PMDLH 寄存器, 低 8 位作为 PMDLH 寄存器时, PMDL 可以以 8 位或 1 位为单位读/写。
 2. 要以 8 位或 1 位为单位读/写 PMDL 寄存器的位 8 至 15 , 需要将他们指定为 PMDLH 寄存器的位 0 至 7。

(3) 端口 DL 模式控制寄存器 (PMCDL)

复位后: 0000H R/W 地址: PMCDL FFFFF044H,
 PMCDLL FFFFF044H, PMCDLH FFFFF045H

	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	指定PDLn引脚的操作模式 (n = 0到15)						
	0	I/O 端口						
	1	ADn I/O (地址/数据总线 I/O)						

注意事项 当 EXIMC 寄存器的 **SMSEL** 位 = 1 (单独模式) 并且 **BSC** 寄存器的 **BS30** 至 **BS00** 位 = 0 (8 位总线带宽), 不要指定 **AD8** 至 **AD15** 引脚。

备注

1. 可以以 16 位为单位读/写 **PMCDL** 寄存器。
此外, 当使用 **PMCDL** 寄存器的高 8 位作为 **PMCDLH** 寄存器, 低 8 位作为 **PMCDLL** 寄存器时, **PMCDL** 可以以 8 位或 1 位为单位读/写。
2. 要以 8 位或 1 位为单位读/写 **PMCDL** 寄存器的位 8 至 15 , 需要将他们指定为 **PMCDLH** 寄存器的位 0 至 7。

4.4 框图

图 4-3. A-1 类型的框图

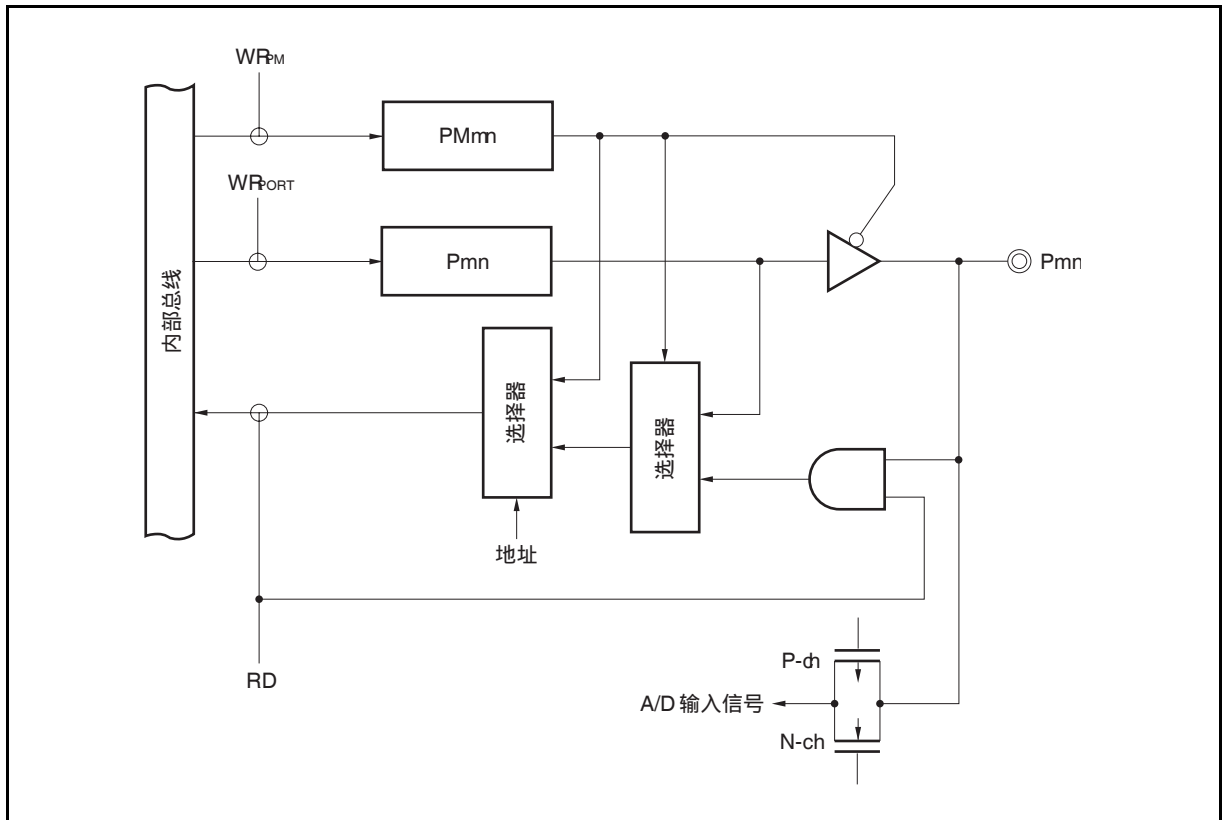


图 4-4. A-2 类型的框图

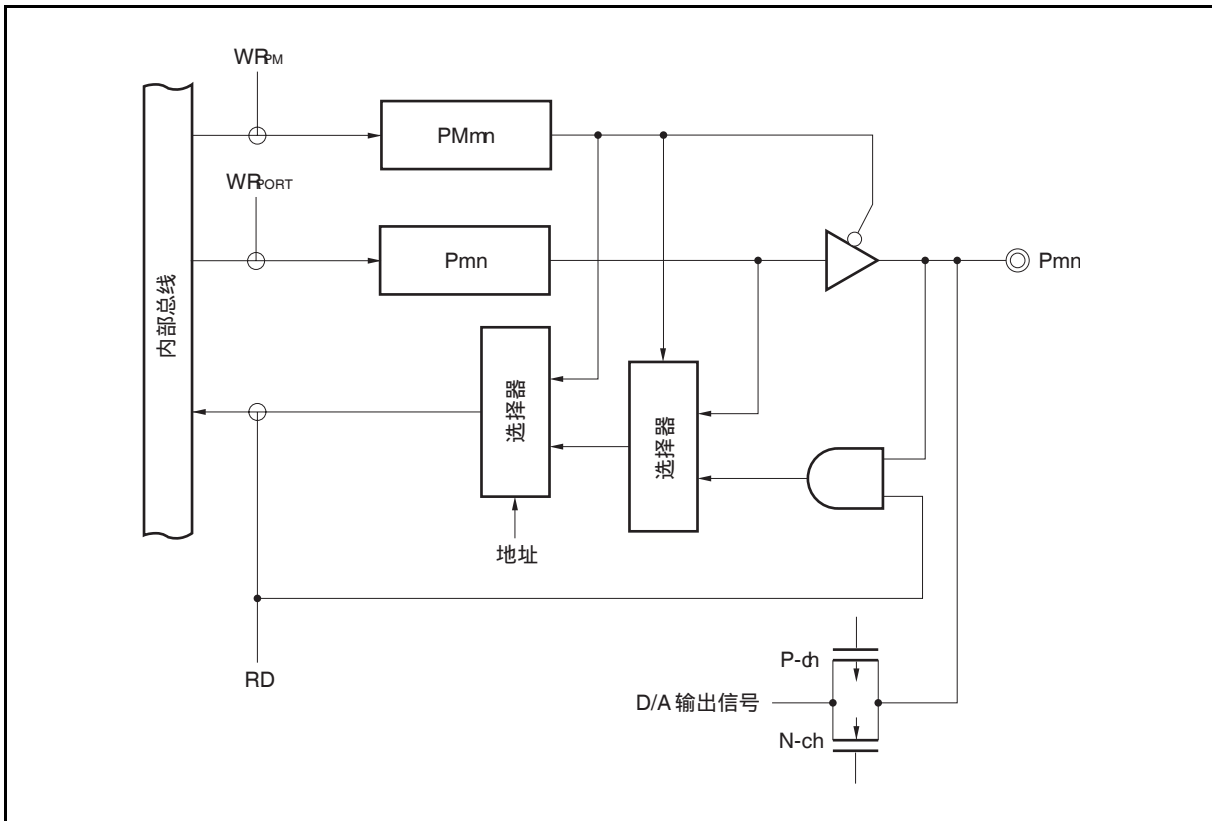


图 4-5. C-1 类型的框图

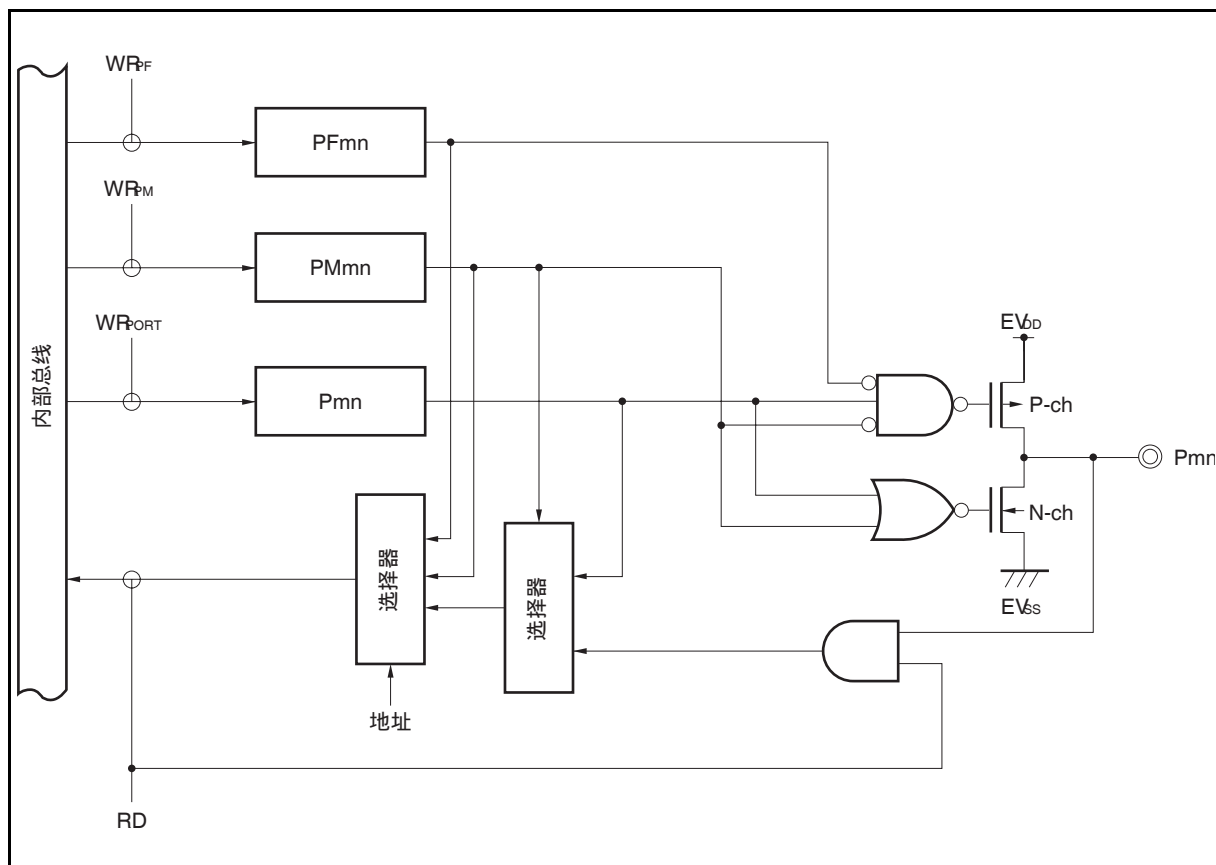


图 4-6. D-1 类型的框图

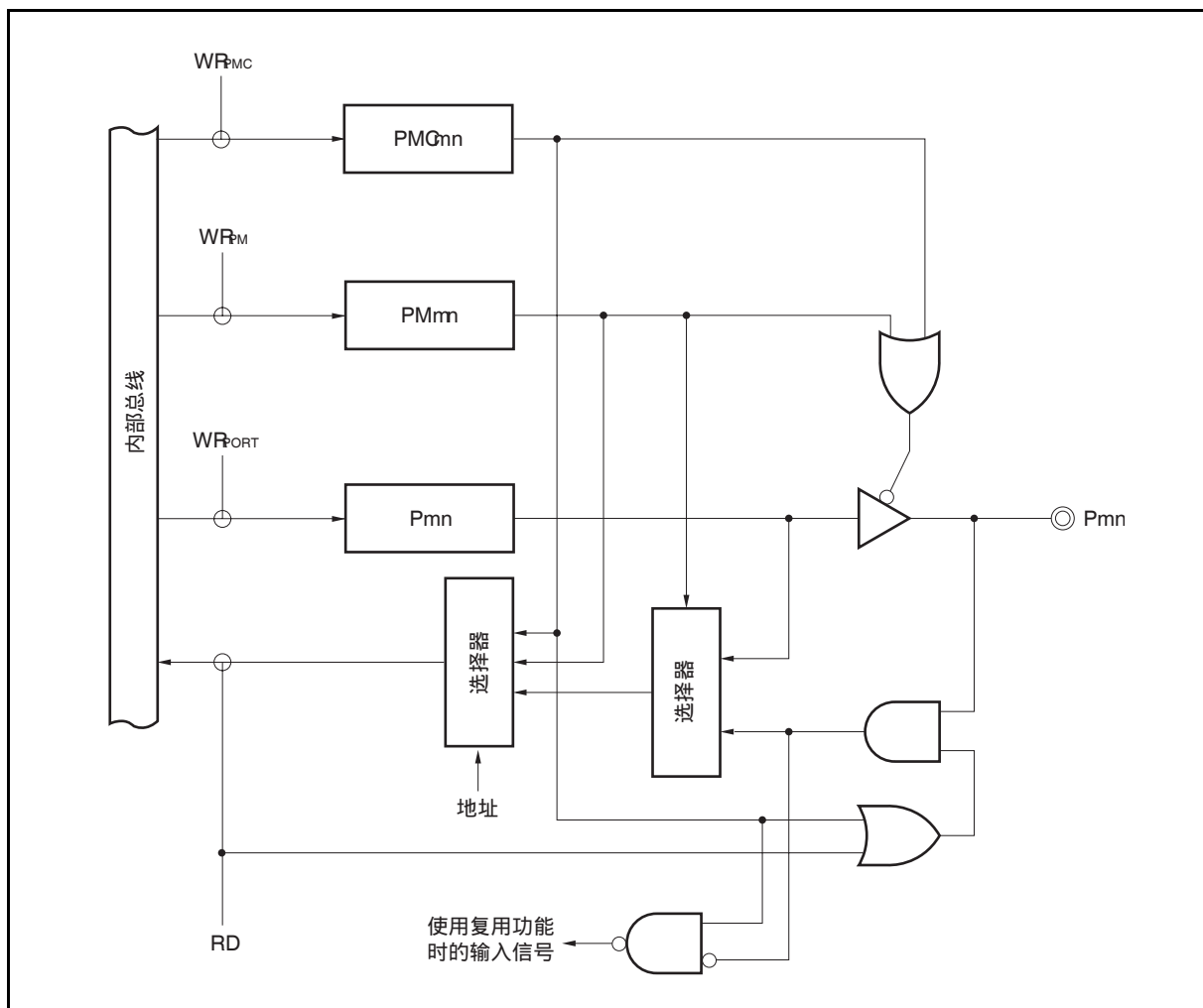


图 4-7. D-2 类型的框图

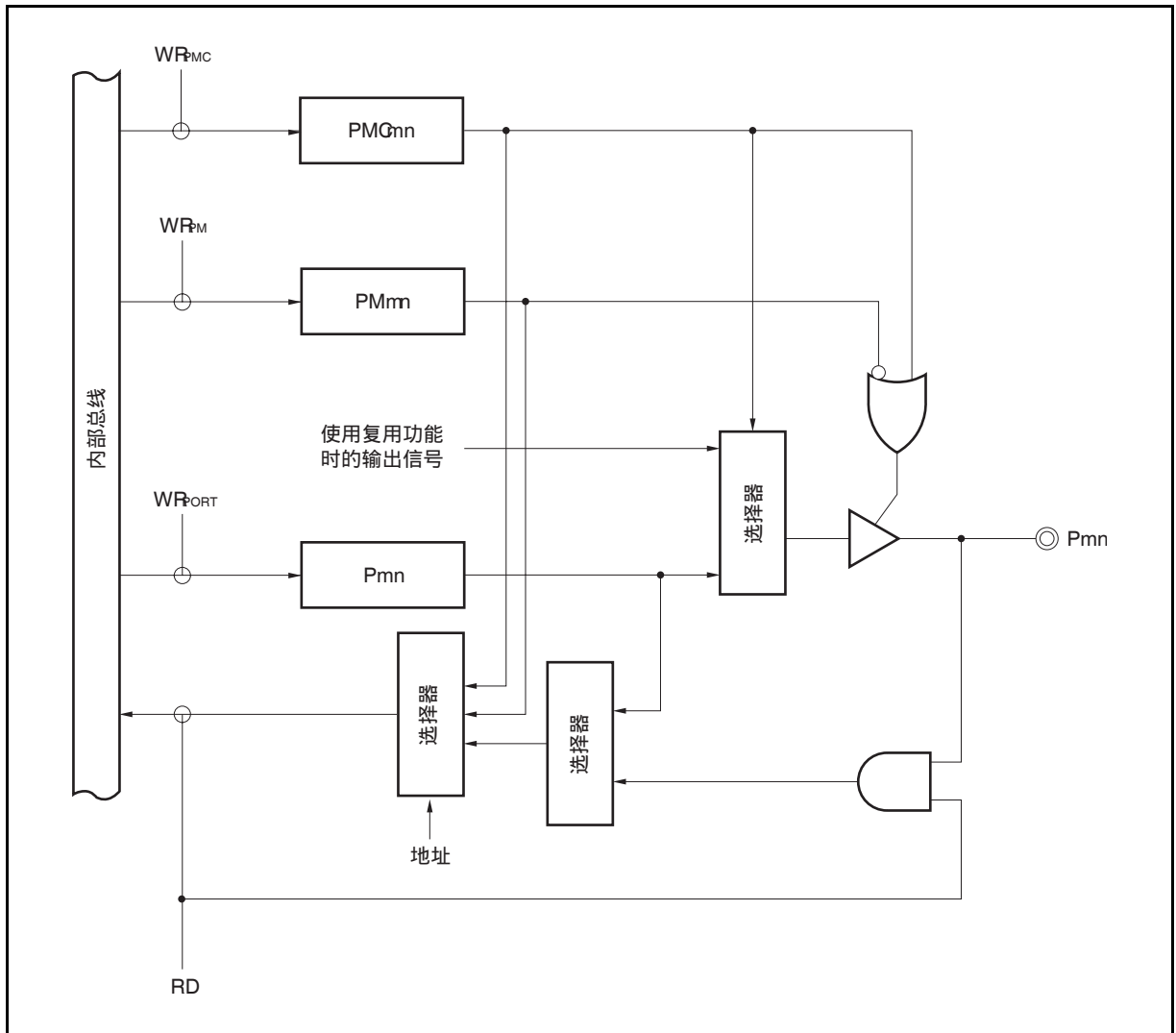


图 4-9. E-3 类型的框图

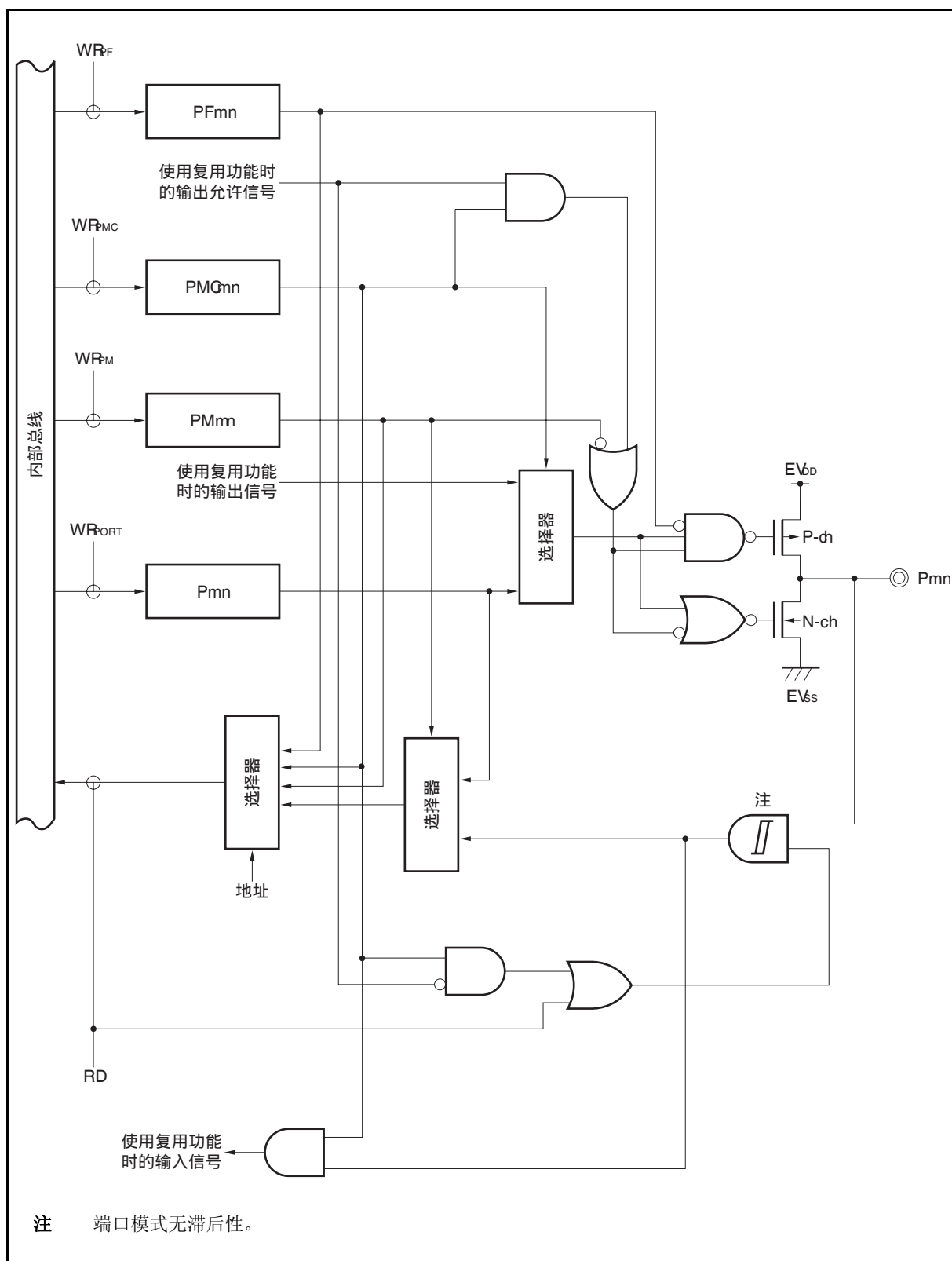


图 4-11. G-2 类型的框图

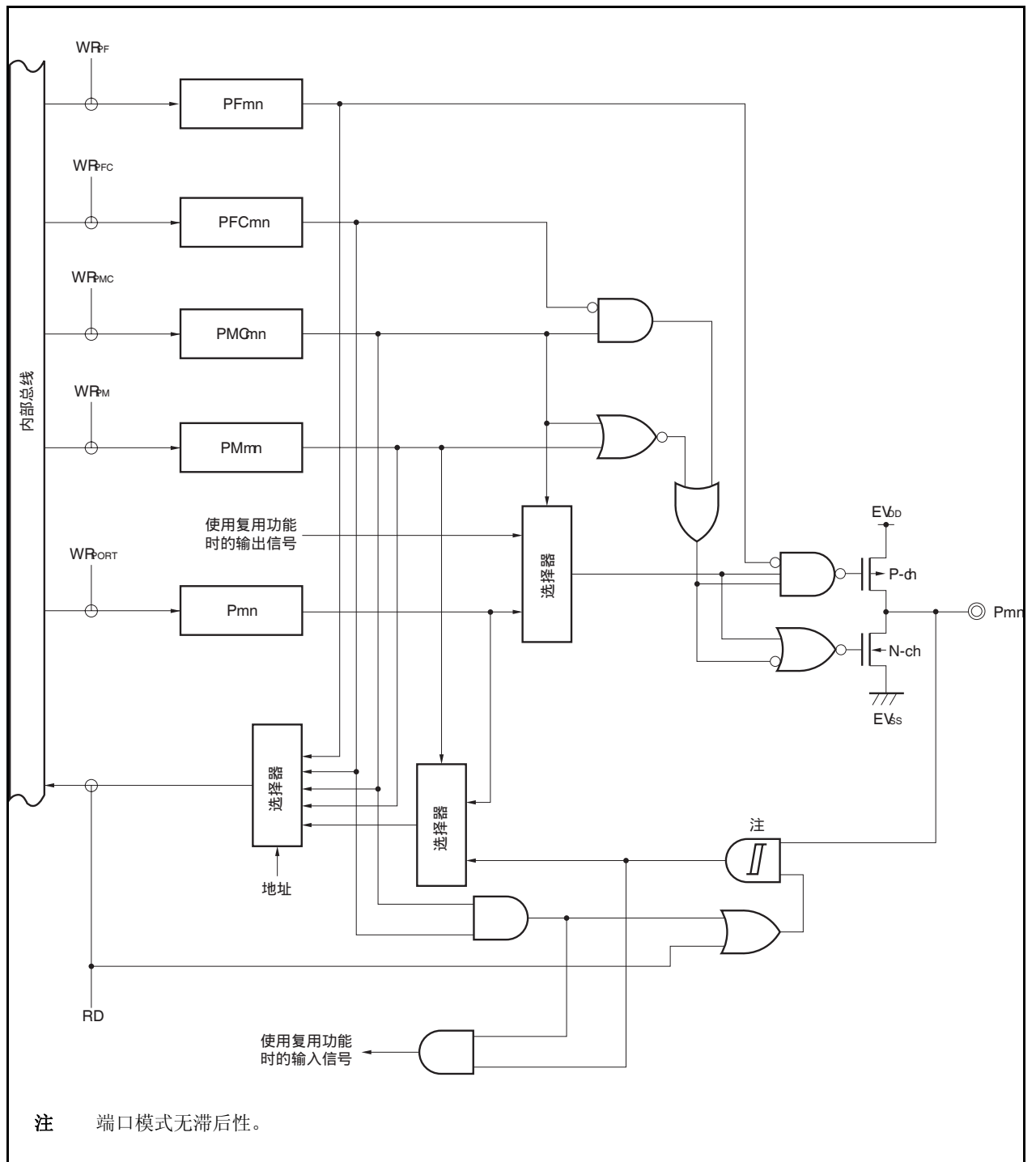


图 4-12. G-3 类型的框图

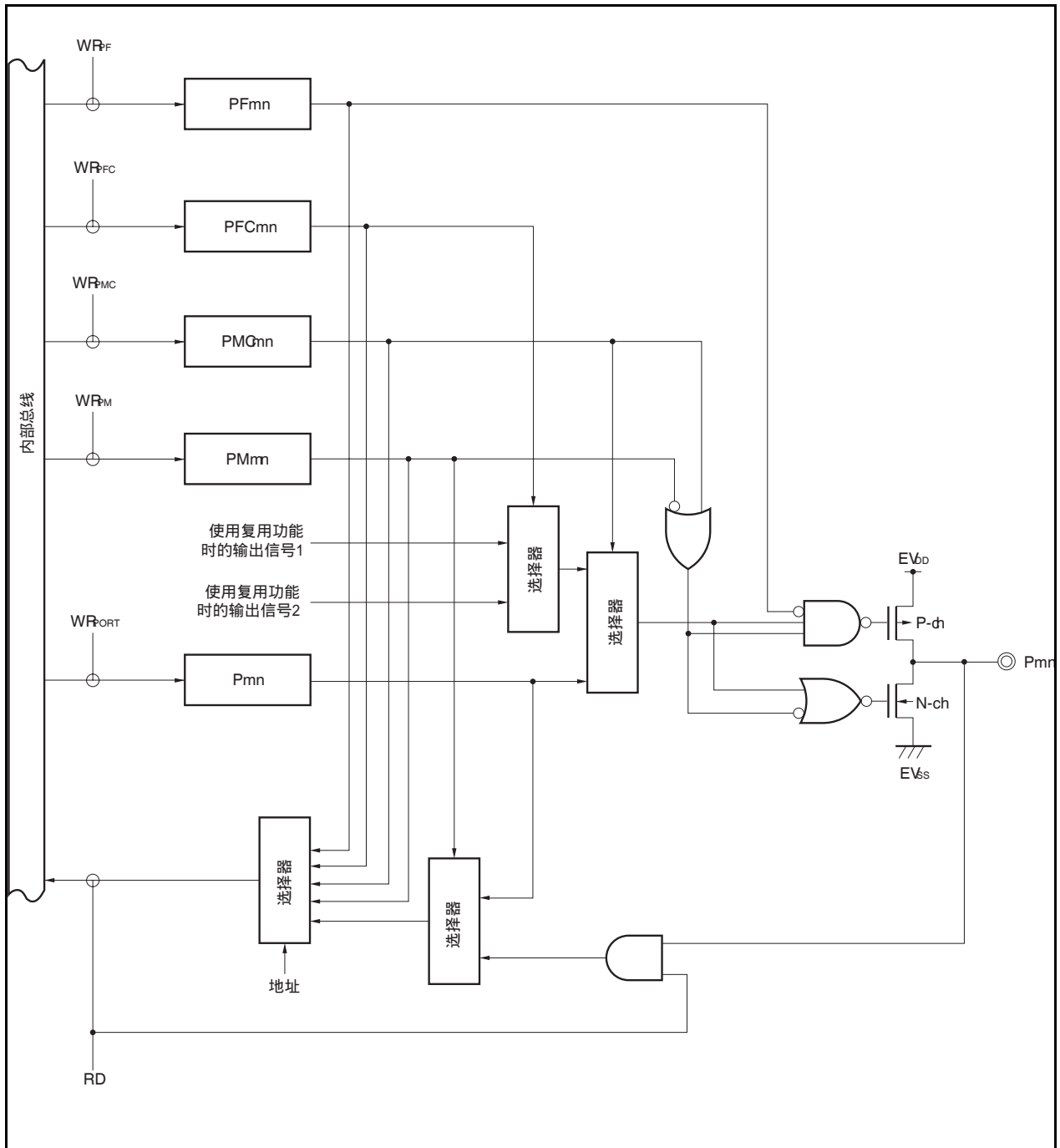


图 4-13. G-5 类型的框图

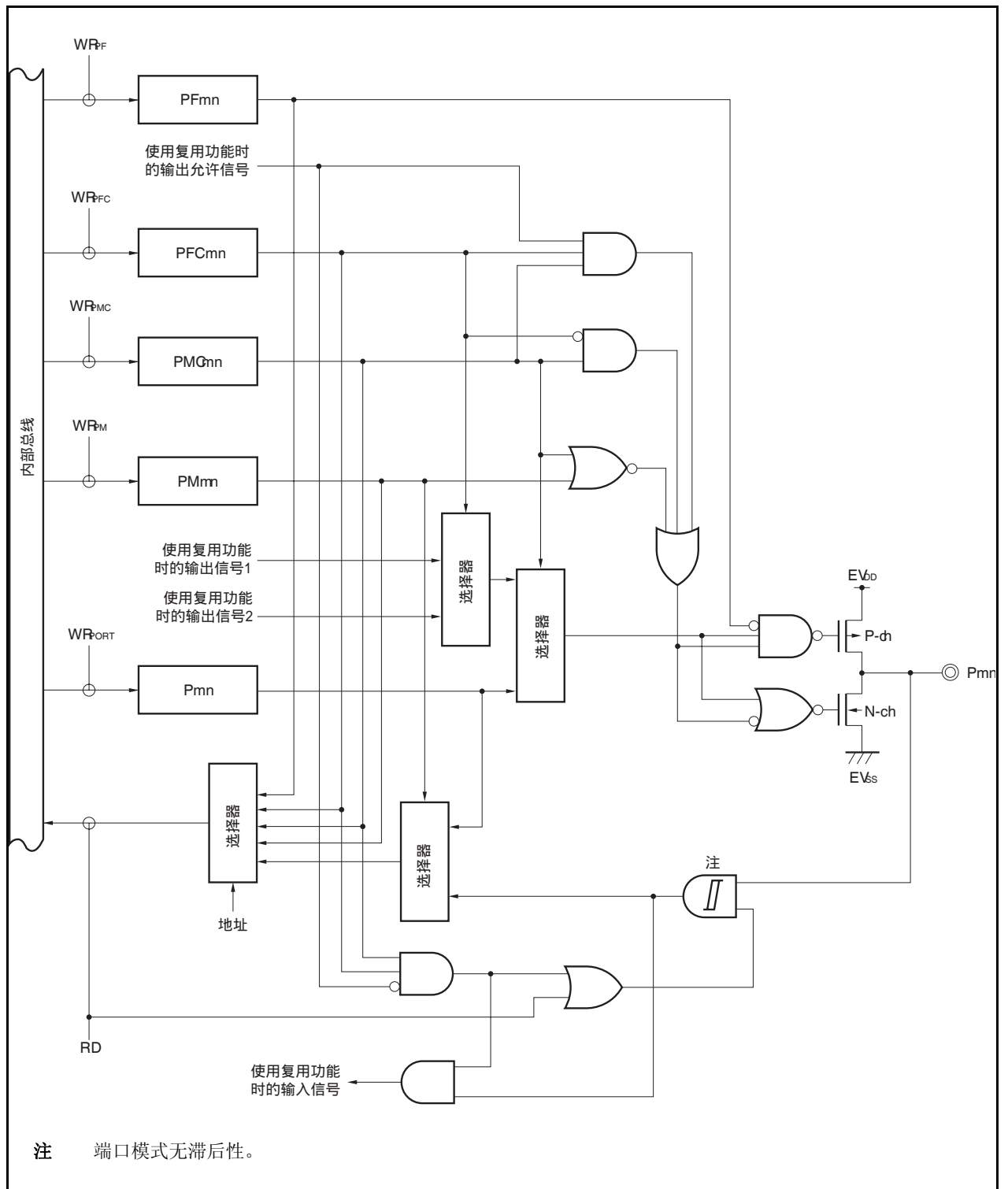




图 4-16. L-1 类型的框图

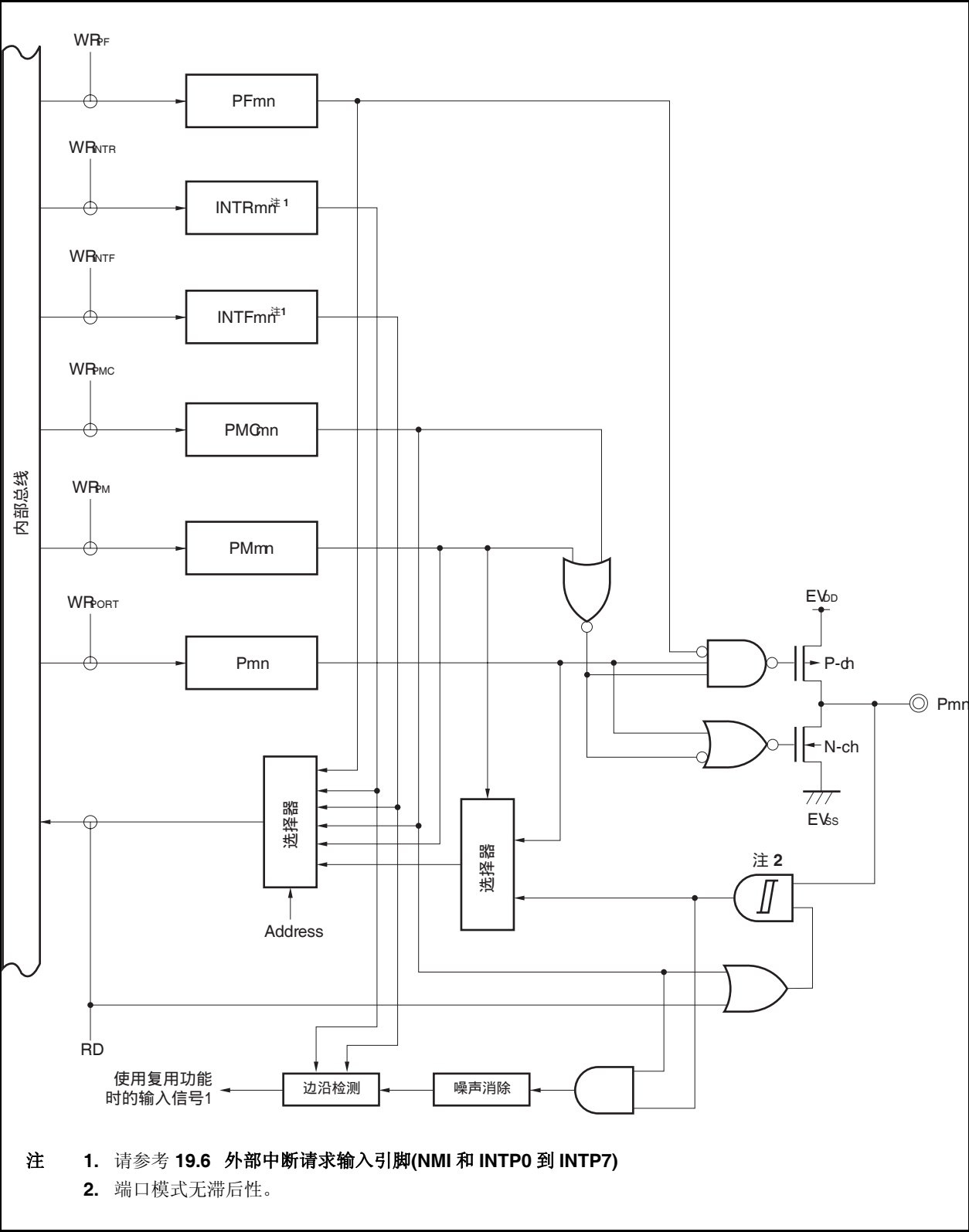


图 4-17. N-1 类型的框图

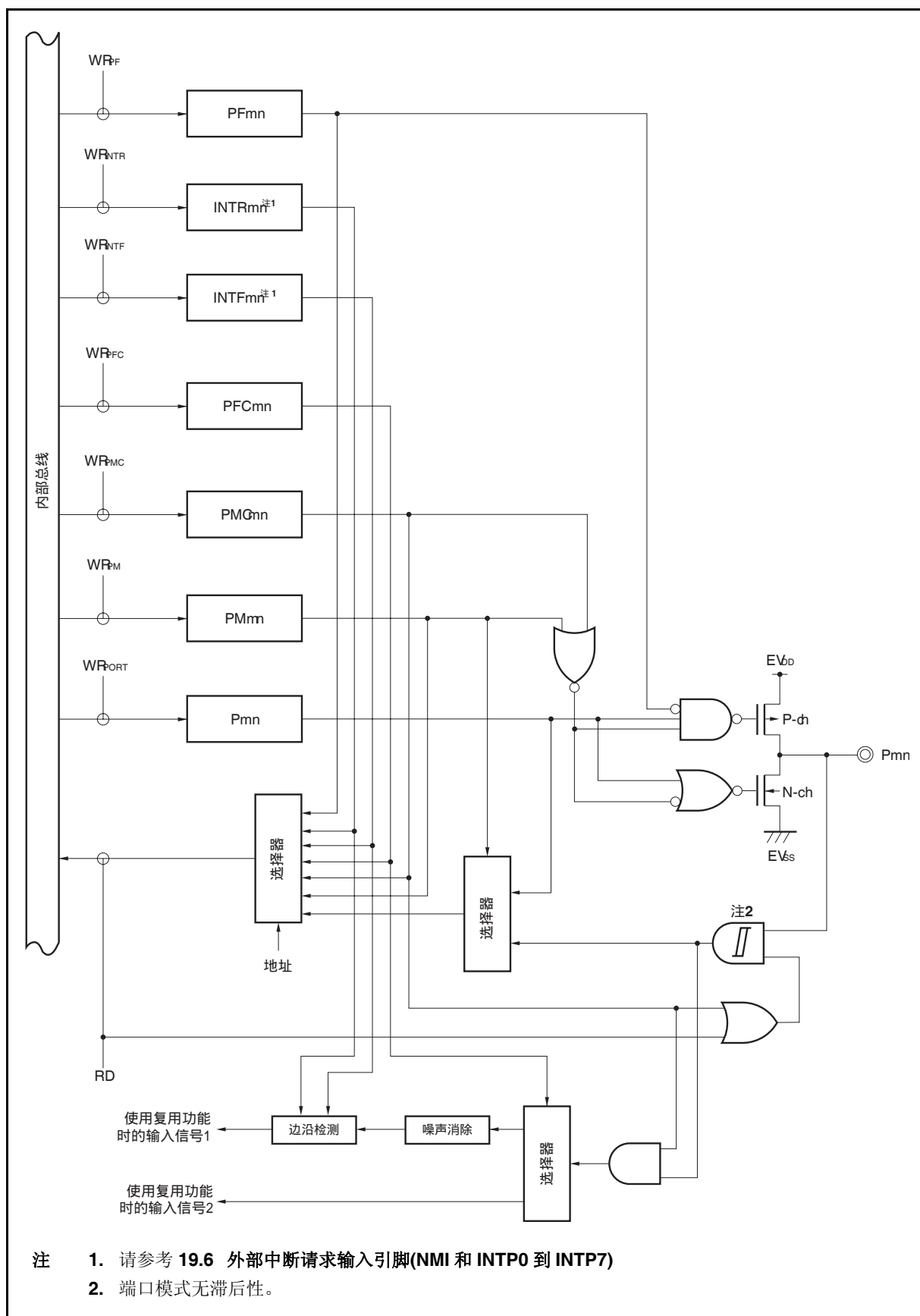


图 4-18. N-2 类型的框图

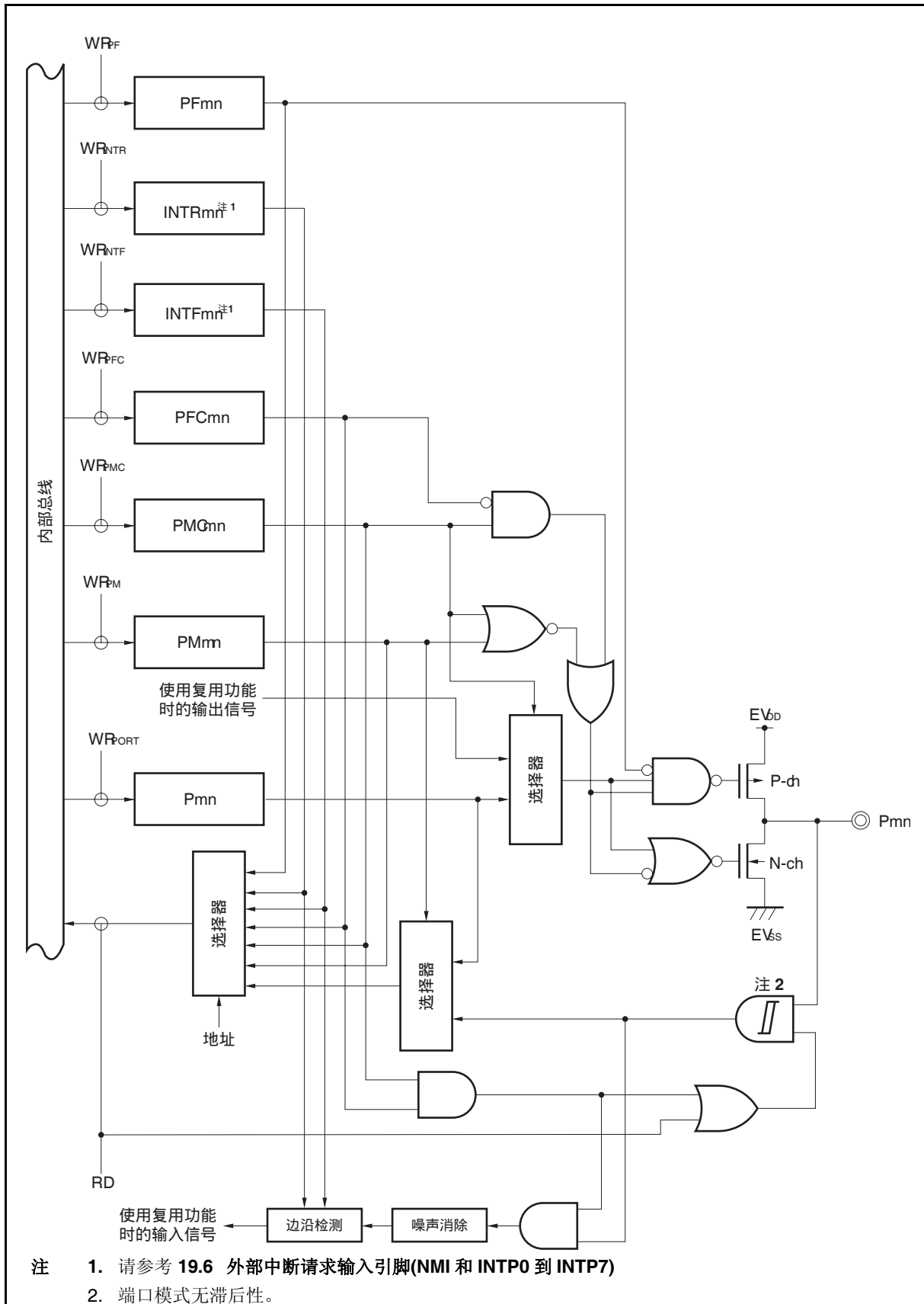


图 4-19. N-3 类型的框图

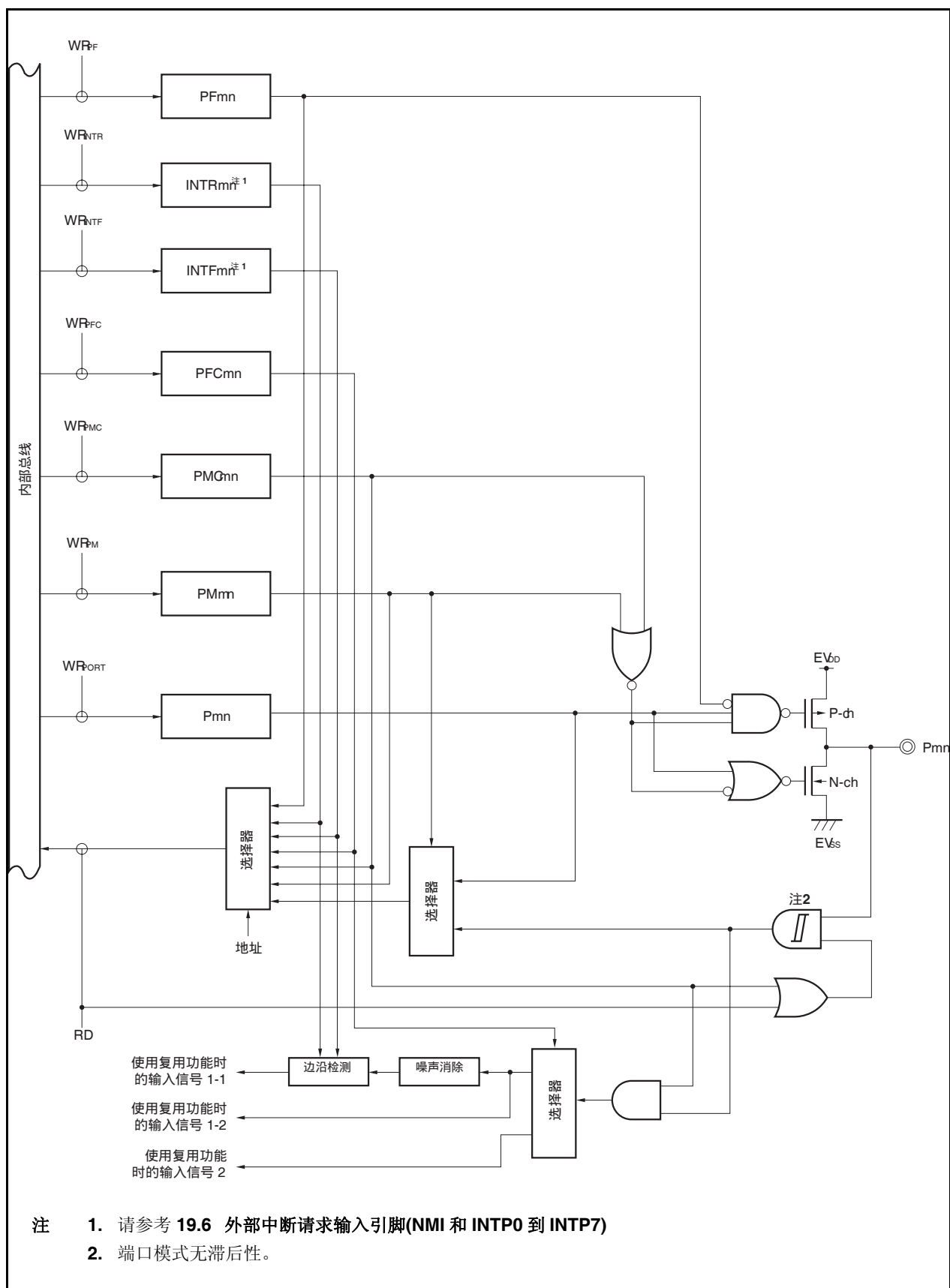


图 4-20. U-1 类型的框图

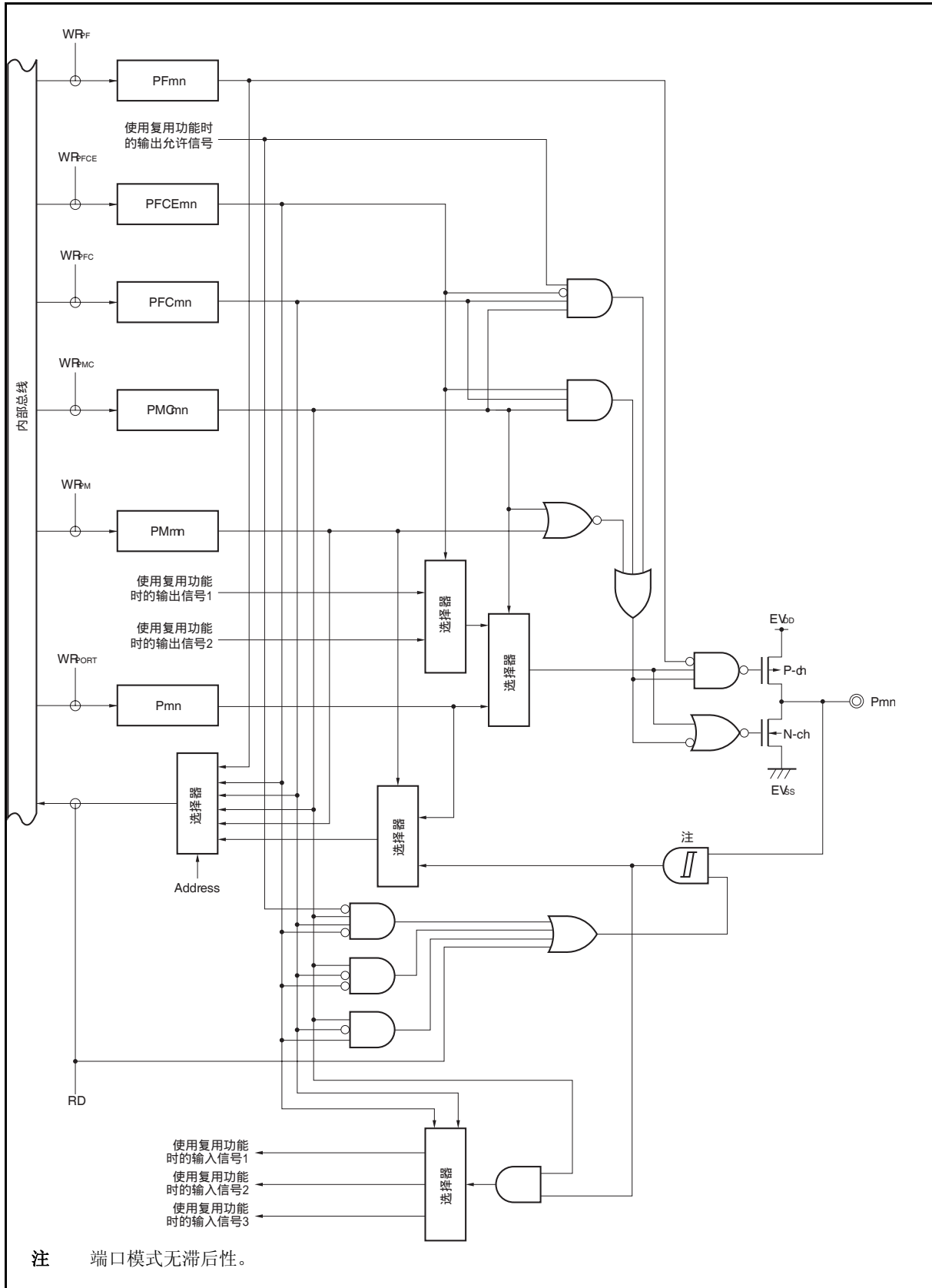


图 4-21. U-5 类型的框图

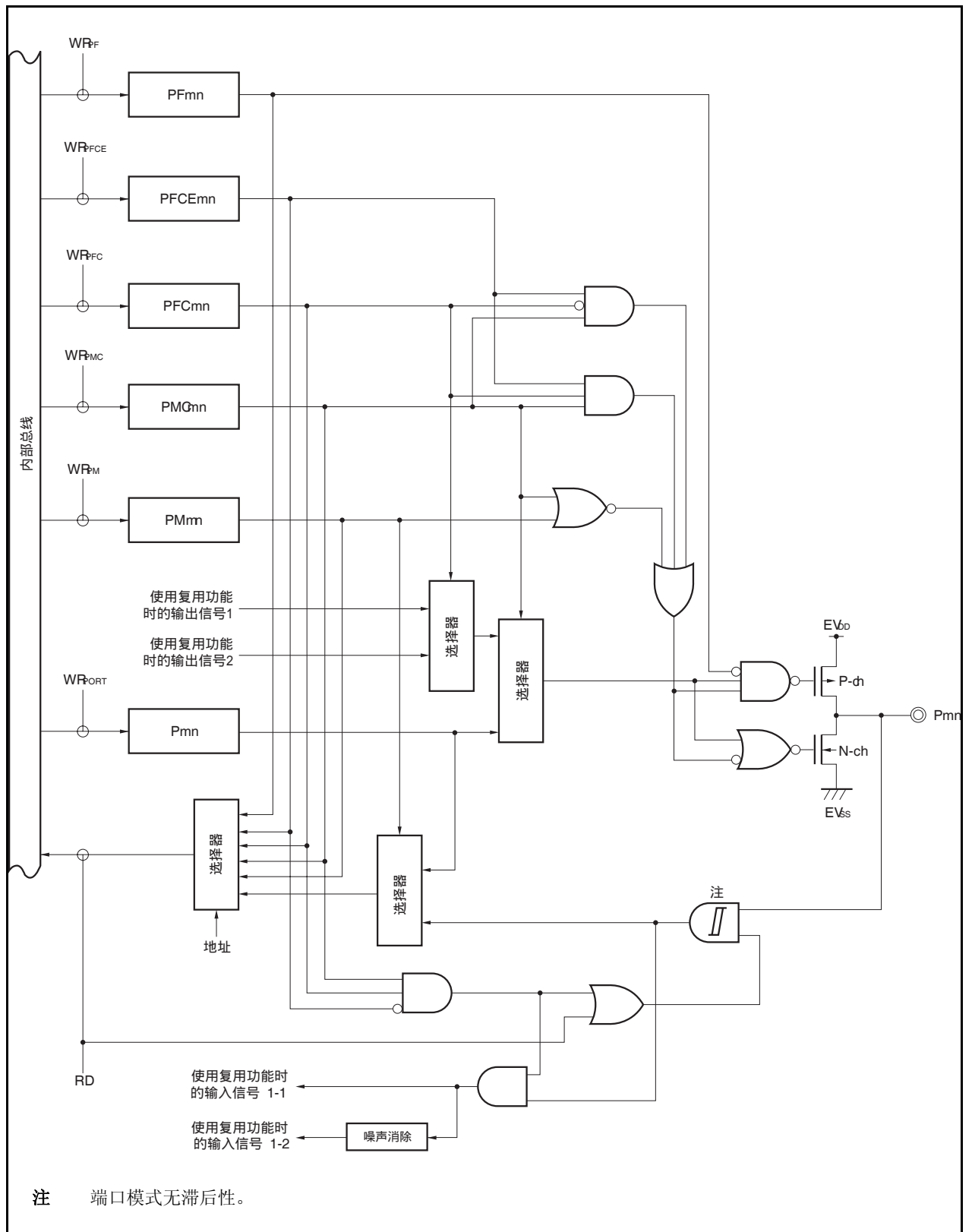


图 4-22. U-6 类型的框图

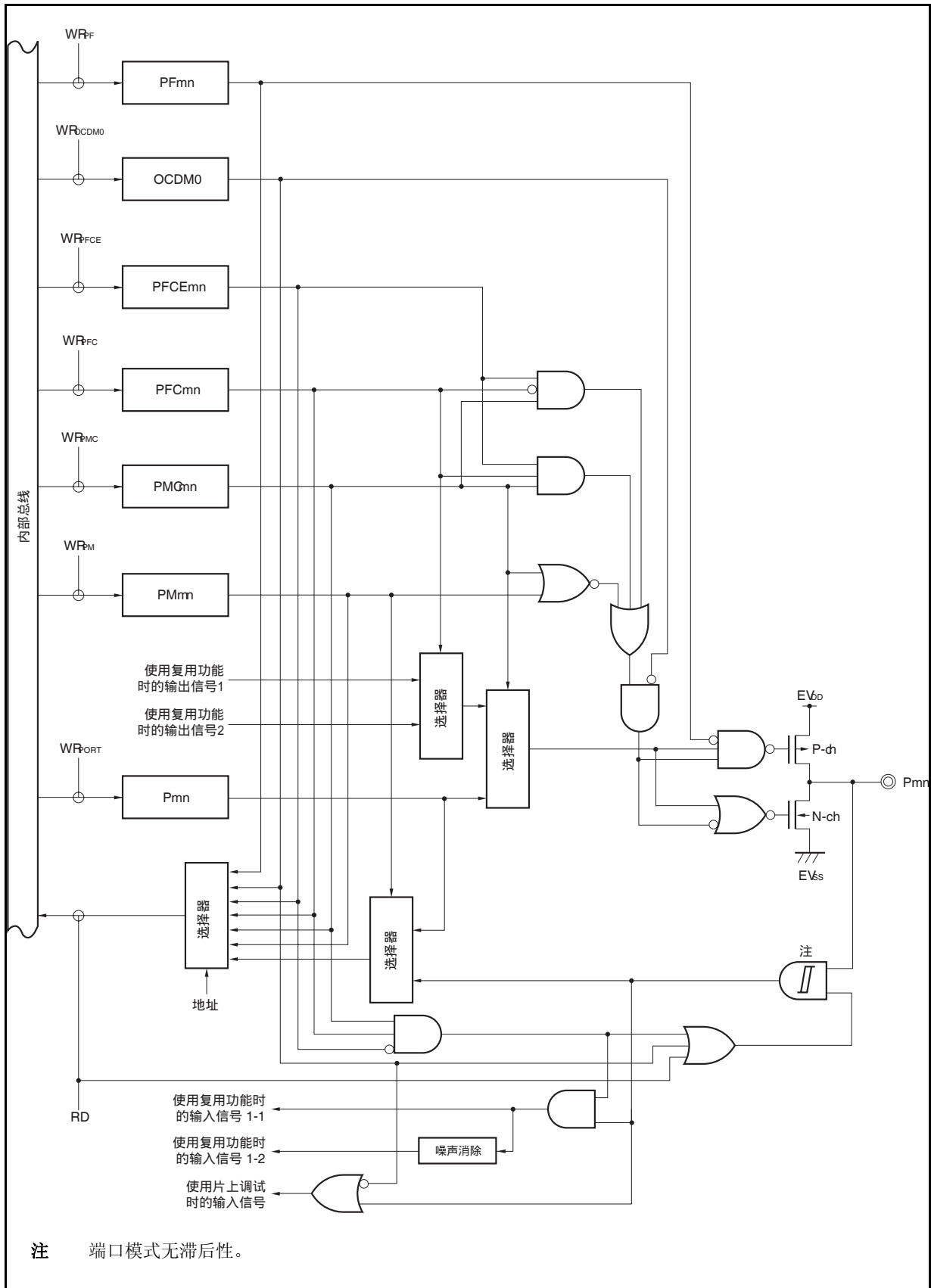
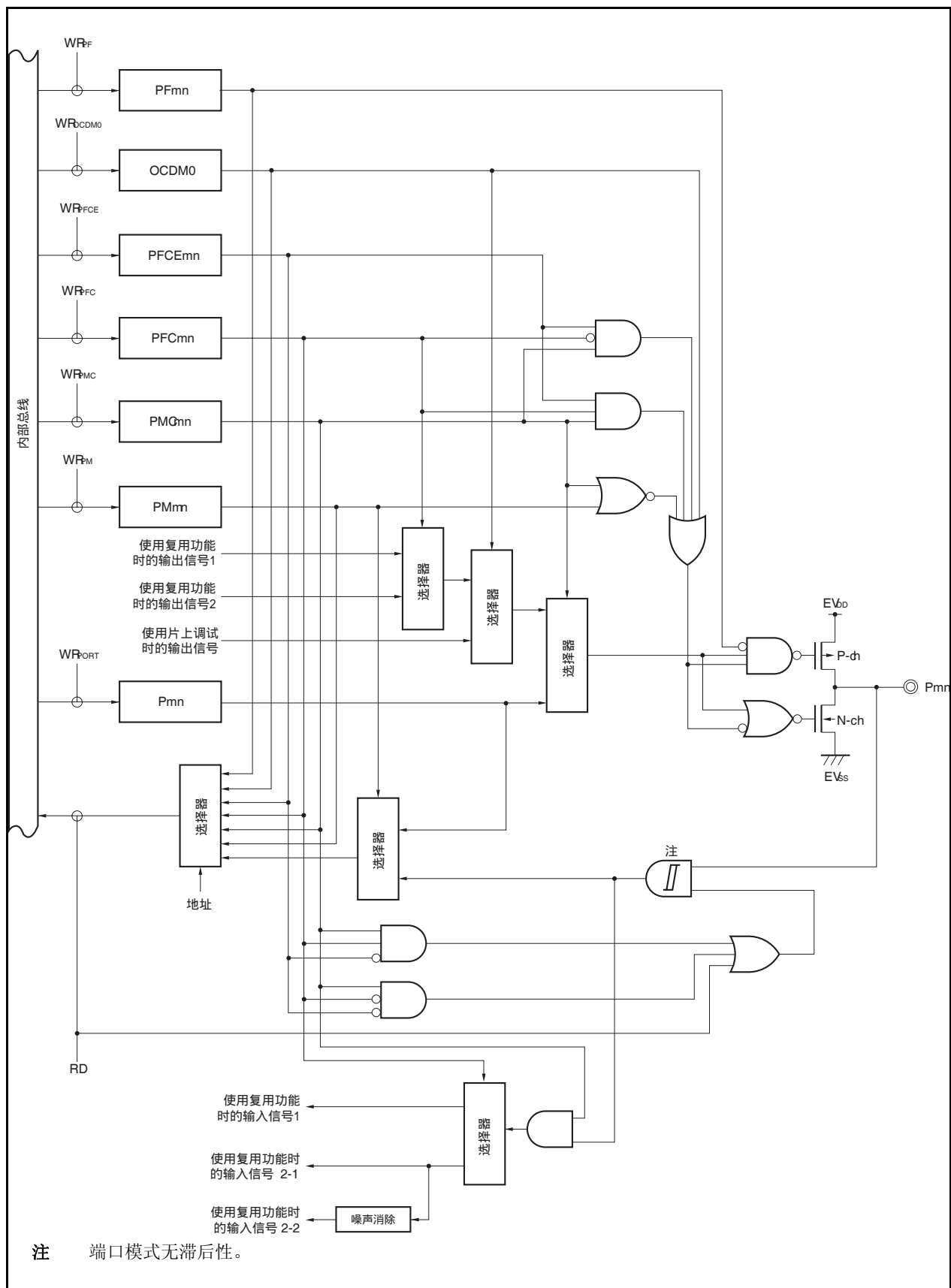


图 4-23. U-7 类型的框图





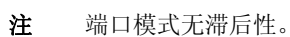


图 4-27. U-11 类型的框图

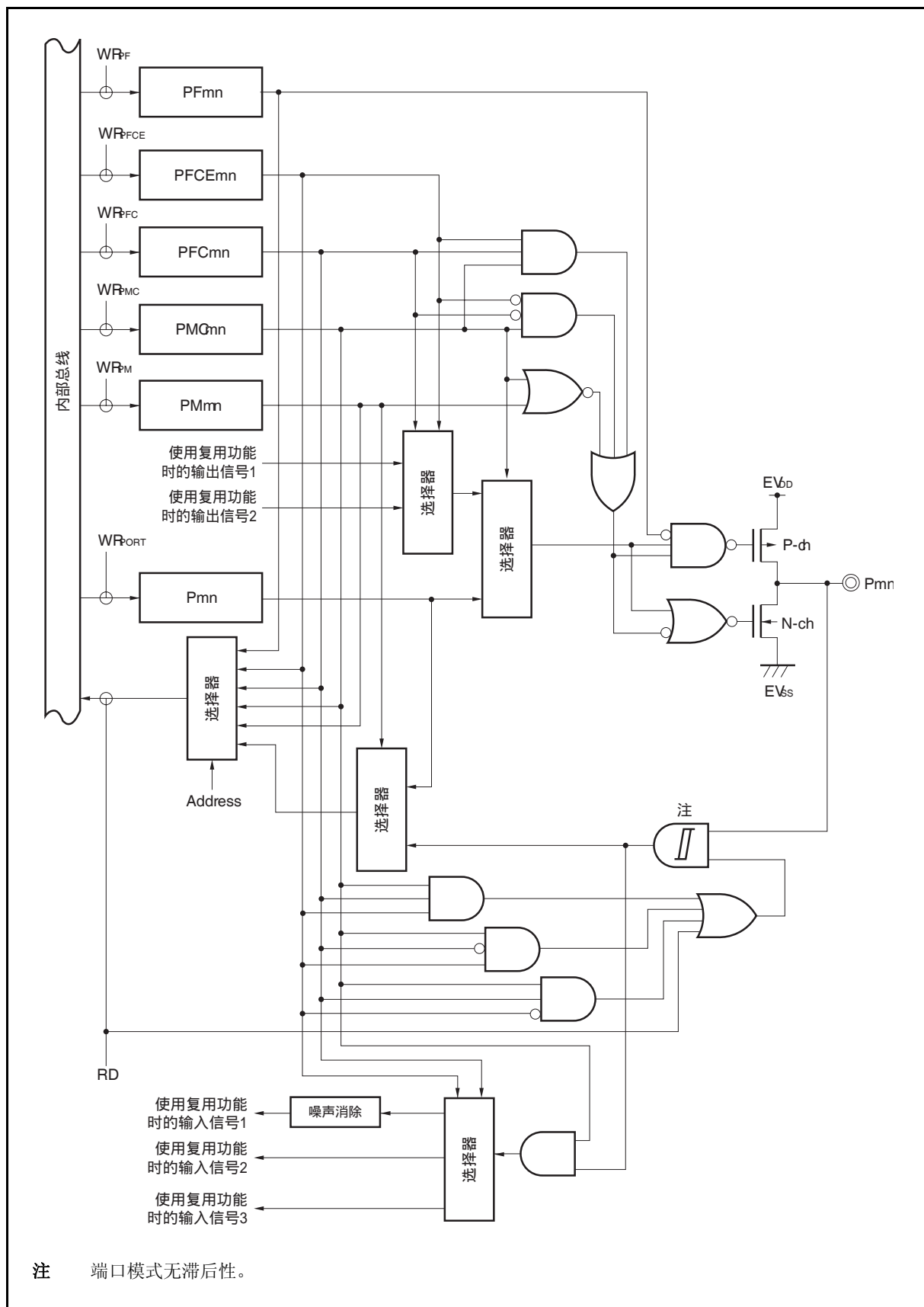


图 4-28. U-12 类型的框图

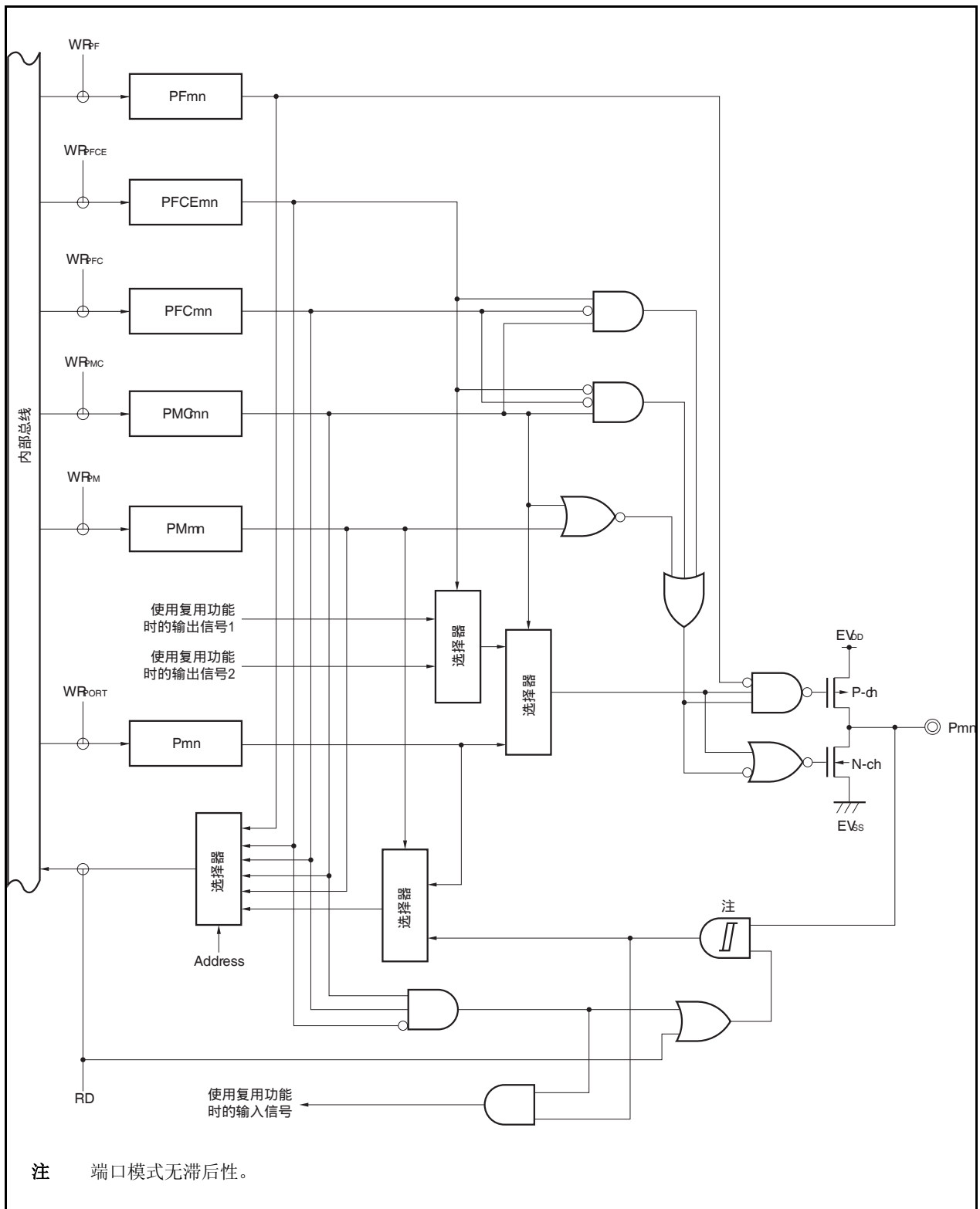


图 4-29. U-13 类型的框图

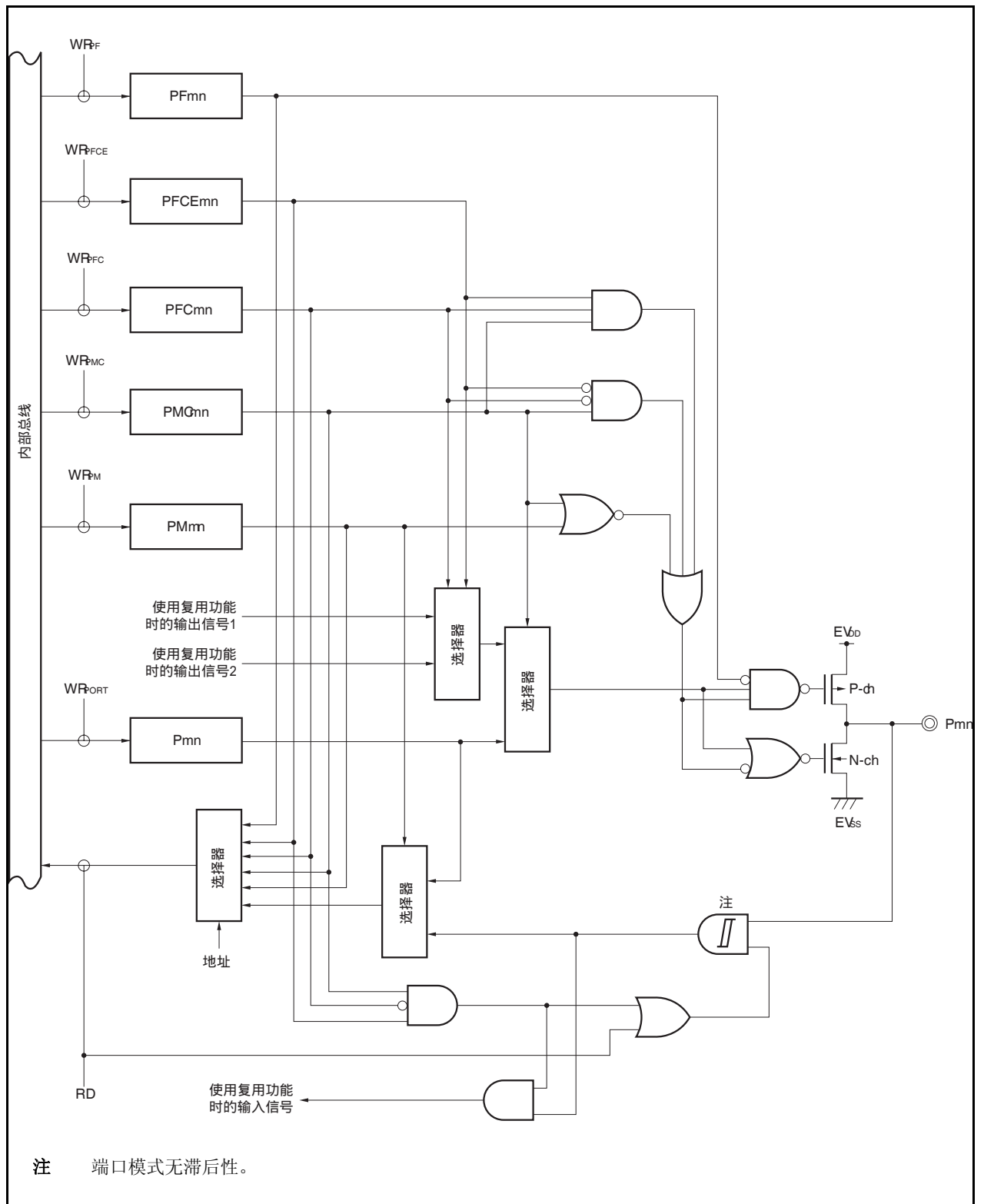


图 4-30. U-14 类型的框图

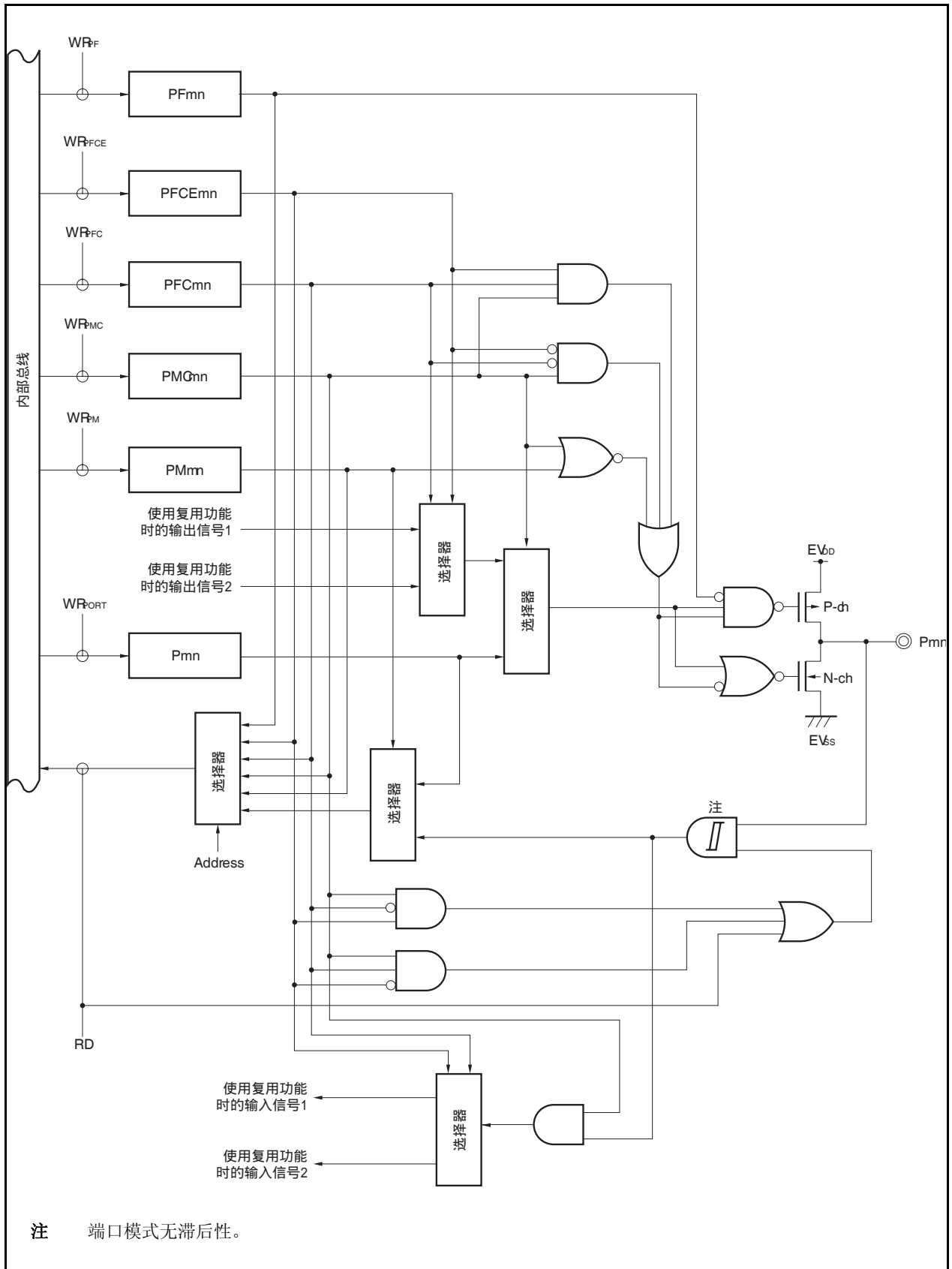


图 4-31. U-15 类型的框图

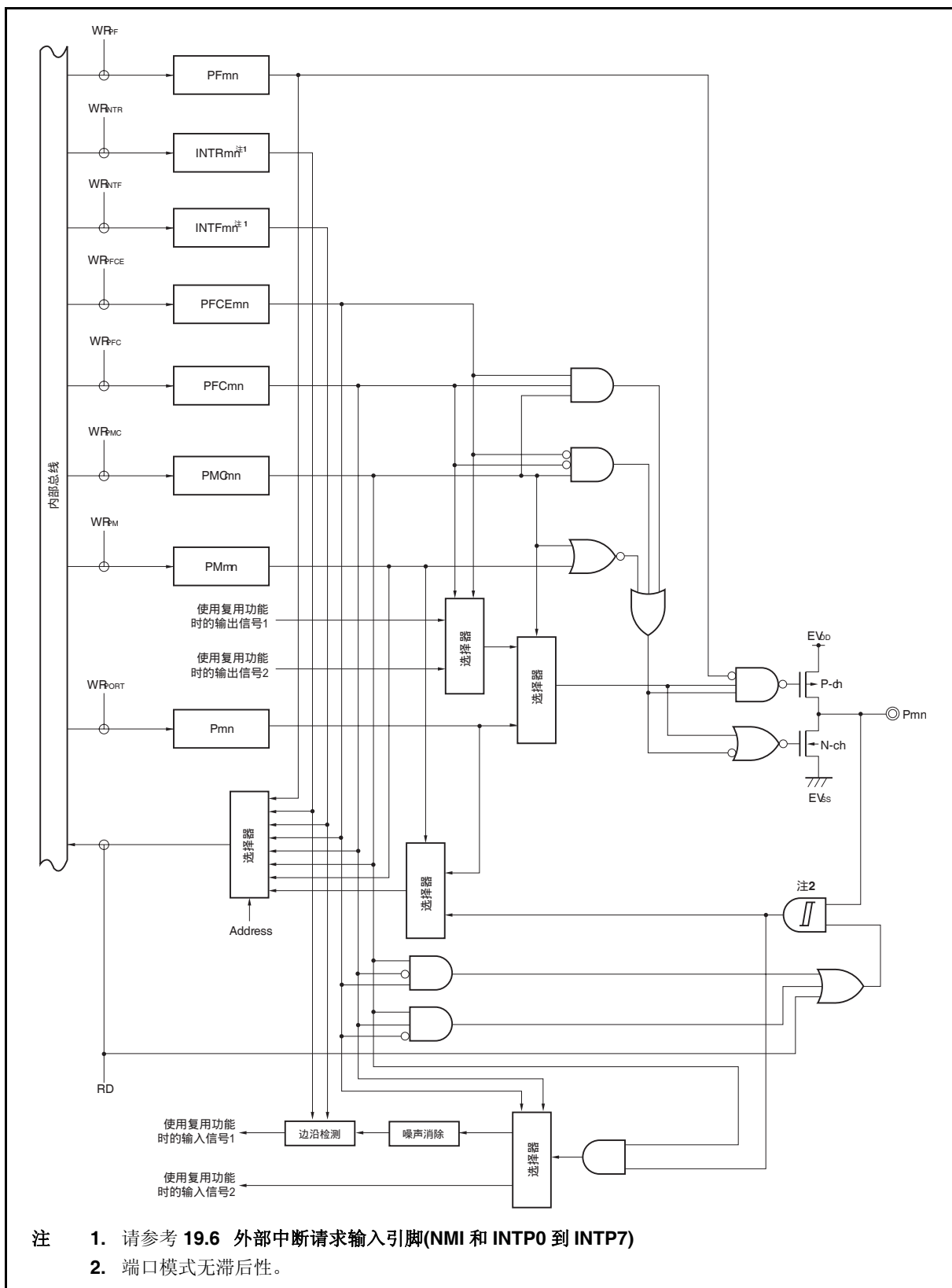
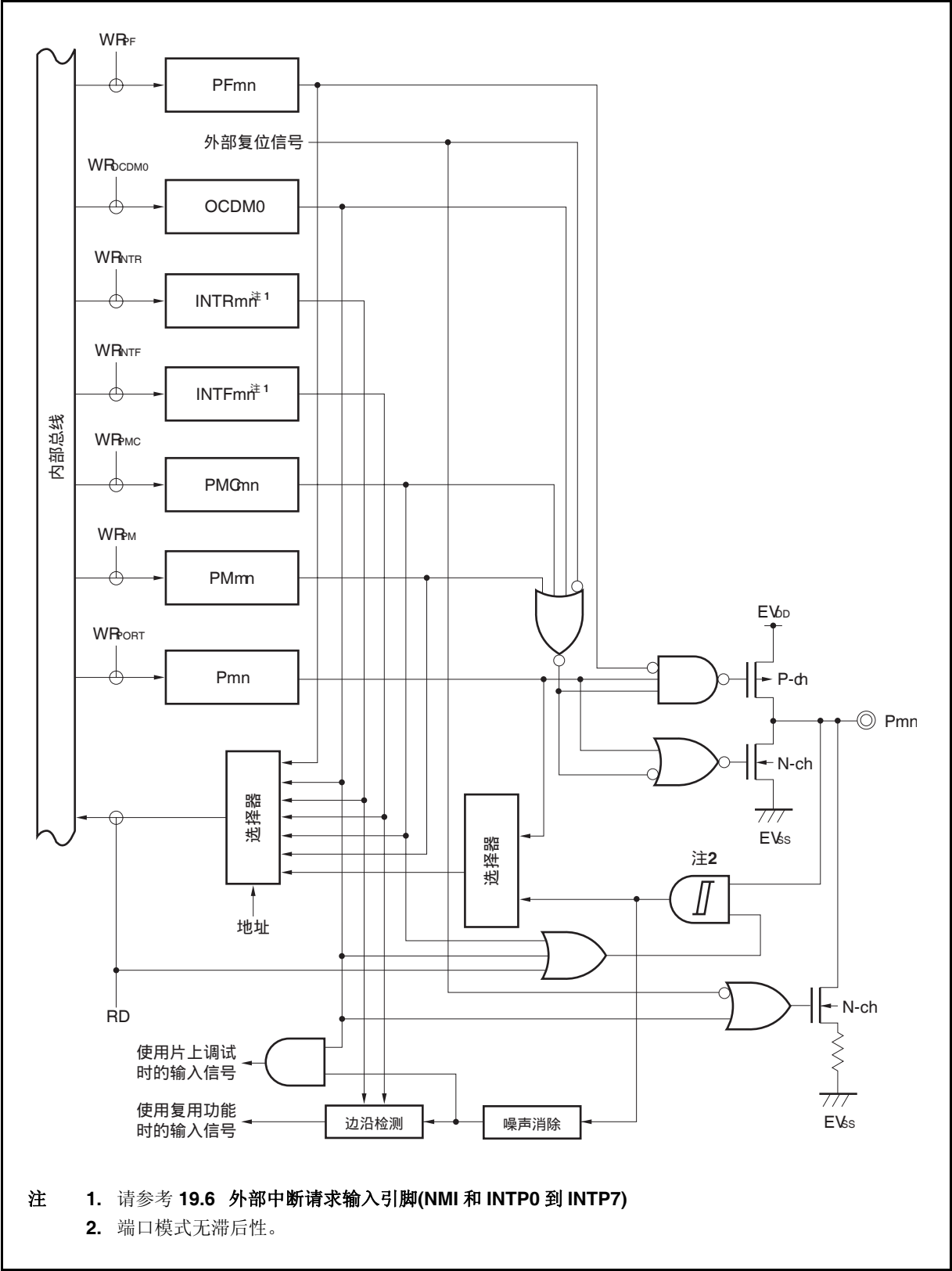


图 4-32. AA-1 类型的框图



4.5 当使用复用功能时的端口寄存器设置

表 4-15 为每个端口用作复用功能时的端口寄存器设置。当端口引脚用作复用功能引脚时，请参考每个引脚的说明。

表 4-15. 端口引脚用作复用功能引脚 (1/7)

引脚名称	复用功能		Pn寄存器的 Pnx位	PMn寄存器的 PMnx位	PMG寄存器的 PMGx位	PFCE寄存器的 PFCEn位	PFC寄存器的 PFCn位	其他位 (寄存器)
	名称	I/O						
P02	NIMI	输入	P02 = 无须设置	PM02 = 无须设置	PM02 = 1			
P03	INTP0	输入	P03 = 无须设置	PM03 = 无须设置	PM03 = 1		PFC03 = 0	
	ADT0	输入	P03 = 无须设置	PM03 = 无须设置	PM03 = 1		PFC03 = 1	
	INTP1	输入	P04 = 无须设置	PM04 = 无须设置	PM04 = 1			
P05	INTP2	输入	P05 = 无须设置	PM05 = 无须设置	PM05 = 1			
	DRST	输入	P05 = 无须设置	PM05 = 无须设置	PM05 = 无须设置			OCDM0 (OCDM) = 1
	INTP3	输入	P06 = 无须设置	PM06 = 无须设置	PM06 = 1			
P10	ANO0	输出	P10 = 无须设置	PM10 = 1				
P11	ANO1	输出	P11 = 无须设置	PM11 = 1				
P30	TXDA0	输出	P30 = 无须设置	PM30 = 无须设置	PM30 = 1		PFC30 = 0	
	SOB4	输出	P30 = 无须设置	PM30 = 无须设置	PM30 = 1		PFC30 = 1	
	RXDA0	输入	P31 = 无须设置	PM31 = 无须设置	PM31 = 1		Note: PFC31 = 0	
	INTP7	输入	P31 = 无须设置	PM31 = 无须设置	PM31 = 1		Note: PFC31 = 0	
	SIB4	输入	P31 = 无须设置	PM31 = 无须设置	PM31 = 1		PFC31 = 1	
P32	ASCKA0	输入	P32 = 无须设置	PM32 = 无须设置	PM32 = 1	PFCE32 = 0	PFC32 = 0	
	SCKB4	I/O	P32 = 无须设置	PM32 = 无须设置	PM32 = 1	PFCE32 = 0	PFC32 = 1	
	TIP00	输入	P32 = 无须设置	PM32 = 无须设置	PM32 = 1	PFCE32 = 1	PFC32 = 0	
	TOR0	输出	P32 = 无须设置	PM32 = 无须设置	PM32 = 1	PFCE32 = 1	PFC32 = 1	
	TIP01	输入	P33 = 无须设置	PM33 = 无须设置	PM33 = 1		PFC33 = 0	
P33	TOR1	输出	P33 = 无须设置	PM33 = 无须设置	PM33 = 1		PFC33 = 1	

注 INTP7 引脚和 RXDA0 引脚为一组复用功能引脚。当用作 RXDA0 引脚时，需要禁止 INTP7 复用功能引脚的边沿检测 (将 INTF3.INTF31 位和 INTR3.INTR31 位清 0)。当用作 INTP7 引脚时，需要停止 UARTA0 接收 (将 UA0CTL0.UA0RXE 位清 0)。

注意事项 当使用 P10 和 P11 其中一个引脚作为 I/O 端口，另一个作为 D/A 输出引脚时(ANO0, ANO1)，确保 D/A 输出期间不改变 I/O 端口电平。

表 4-15. 端口引脚用作复用功能引脚 (2/7)

引脚名称	复用功能		Pn寄存器的 Pnx位	PMn寄存器的 PMnx位	PMC寄存器的 PMGx位	PFCE寄存器的 PFCEn位	PFC寄存器的 PFCn位	其他位 (寄存器)
	名称	I/O						
P34	TIP10	输入	P34 = 无须设置	PM34 = 无须设置	PMC34 = 1		PFC34 = 0	
	TOP10	输出	P34 = 无须设置	PM34 = 无须设置	PMC34 = 1		PFC34 = 1	
P35	TIP11	输入	P35 = 无须设置	PM35 = 无须设置	PMC35 = 1		PFC35 = 0	
	TOP11	输出	P35 = 无须设置	PM35 = 无须设置	PMC35 = 1		PFC35 = 1	
P38	TXDA2	输出	P38 = 无须设置	PM38 = 无须设置	PMC38 = 1		PFC38 = 0	
	SDA00	I/O	P38 = 无须设置	PM38 = 无须设置	PMC38 = 1		PFC38 = 1	PF38 (PF3) = 1
P39	RXDA2	输入	P39 = 无须设置	PM39 = 无须设置	PMC39 = 1		PFC39 = 0	
	SCL00	I/O	P39 = 无须设置	PM39 = 无须设置	PMC39 = 1		PFC39 = 1	PF39 (PF3) = 1
P40	SIB0	输入	P40 = 无须设置	PM40 = 无须设置	PMC40 = 1		PFC40 = 0	
	SDA01	I/O	P40 = 无须设置	PM40 = 无须设置	PMC40 = 1		PFC40 = 1	PF40 (PF4) = 1
P41	SOB0	输出	P41 = 无须设置	PM41 = 无须设置	PMC41 = 1		PFC41 = 0	
	SCL01	I/O	P41 = 无须设置	PM41 = 无须设置	PMC41 = 1		PFC41 = 1	PF41 (PF4) = 1
P42	SCKB0	I/O	P42 = 无须设置	PM42 = 无须设置	PMC42 = 1			
P50	TIQ01	输入	P50 = 无须设置	PM50 = 无须设置	PMC50 = 1	PFCE50 = 0	PFC50 = 1	KRM0 (KRM) = 0
	KR0	输入	P50 = 无须设置	PM50 = 无须设置	PMC50 = 1	PFCE50 = 0	PFC50 = 1	TQ0TIG2, TQ0TIG3 (TQ0IOC1) = 0
	TOQ01	输出	P50 = 无须设置	PM50 = 无须设置	PMC50 = 1	PFCE50 = 1	PFC50 = 0	
	RTP00	输出	P50 = 无须设置	PM50 = 无须设置	PMC50 = 1	PFCE50 = 1	PFC50 = 1	

表 4-15. 端口引脚用作复用功能引脚 (3/7)

引脚名称	复用功能		Pn寄存器的 Pnx位	PMn寄存器的 PMnx位	PFC寄存器的 PFCn位	PFC寄存器的 PFCn位	其他位 (寄存器)
	名称	I/O					
P51	TIQ02	输入	PM3 = 无须设置	PM3 = 1	PFC51 = 0	PFC51 = 1	KRM1 (KRM) = 0
	KR1	输入	PM3 = 无须设置	PM3 = 1	PFC51 = 0	PFC51 = 1	TQ0TIG4, TQ0TIG5 (TQ0IOC1) = 0
	TOQ02	输出	PM3 = 无须设置	PM3 = 1	PFC51 = 1	PFC51 = 0	
	RTP01	输出	PM3 = 无须设置	PM3 = 1	PFC51 = 1	PFC51 = 1	
	TIQ03	输入	PM2 = 无须设置	PM2 = 1	PFC52 = 0	PFC52 = 1	KRM2 (KRM) = 0
P52	KR2	输入	PM2 = 无须设置	PM2 = 1	PFC52 = 0	PFC52 = 1	TQ0TIG6, TQ0TIG7 (TQ0IOC1) = 0
	TOQ03	输出	PM2 = 无须设置	PM2 = 1	PFC52 = 1	PFC52 = 0	
	RTP02	输出	PM2 = 无须设置	PM2 = 1	PFC52 = 1	PFC52 = 1	
	DDI	输入	PM2 = 无须设置	PM2 = 无须设置	PFC52 无须设置	PFC52 无须设置	OCDM0 (OCDM) = 1
	SIB2	输入	PM3 = 无须设置	PM3 = 1	PFC53 = 0	PFC53 = 0	
P53	TIQ00	输入	PM3 = 无须设置	PM3 = 1	PFC53 = 0	PFC53 = 1	KRM3 (KRM) = 0
	KR3	输入	PM3 = 无须设置	PM3 = 1	PFC53 = 0	PFC53 = 1	TQ0TIG0, TQ0TIG1 (TQ0IOC1) = 0, TQ0EES0, TQ0EES1 (TQ0IOC2) = 0, TQ0ETS0, TQ0ETS1 (TQ0IOC2) = 0
	TOQ00	输出	PM3 = 无须设置	PM3 = 1	PFC53 = 1	PFC53 = 0	
	RTP03	输出	PM3 = 无须设置	PM3 = 1	PFC53 = 1	PFC53 = 1	
	DDO	输出	PM3 = 无须设置	PM3 = 无须设置	PFC53 无须设置	PFC53 无须设置	OCDM0 (OCDM) = 1
P54	SOB2	输出	PM3 = 无须设置	PM3 = 1	PFC54 = 0	PFC54 = 0	
	KR4	输入	PM3 = 无须设置	PM3 = 1	PFC54 = 0	PFC54 = 1	
	RTP04	输出	PM3 = 无须设置	PM3 = 1	PFC54 = 1	PFC54 = 1	
	CK	输入	PM3 = 无须设置	PM3 = 1	PFC54 = 1	PFC54 = 1	OCDM0 (OCDM) = 1
	SCKB2	I/O	PM5 = 无须设置	PM5 = 1	PFC55 = 0	PFC55 = 0	
P55	KR5	输入	PM5 = 无须设置	PM5 = 1	PFC55 = 0	PFC55 = 1	
	RTP05	输出	PM5 = 无须设置	PM5 = 1	PFC55 = 1	PFC55 = 1	
	DMS	输入	PM5 = 无须设置	PM5 = 无须设置	PFC55 无须设置	PFC55 无须设置	OCDM0 (OCDM) = 1

表 4-15. 端口引脚用作复用功能引脚 (4/7)

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 PMGnx位	PFCE寄存器的 PFCEn位	PFCn寄存器的 PFCn位	其他位 (寄存器)
	名称	I/O						
P70	AN10	输入	P70 = 无须设置	PM70 = 1				
P71	AN11	输入	P71 = 无须设置	PM71 = 1				
P72	AN12	输入	P72 = 无须设置	PM72 = 1				
P73	AN13	输入	P73 = 无须设置	PM73 = 1				
P74	AN14	输入	P74 = 无须设置	PM74 = 1				
P75	AN15	输入	P75 = 无须设置	PM75 = 1				
P76	AN16	输入	P76 = 无须设置	PM76 = 1				
P77	AN17	输入	P77 = 无须设置	PM77 = 1				
P78	AN18	输入	P78 = 无须设置	PM78 = 1				
P79	AN19	输入	P79 = 无须设置	PM79 = 1				
P710	AN10	输入	P710 = 无须设置	PM710 = 1				
P711	AN11	输入	P711 = 无须设置	PM711 = 1				
P90	A0	输出	P90 = 无须设置	PMC90 = 1	PFCE90 = 0	PFC90 = 0	注 1	
KR6	KR6	输入	P90 = 无须设置	PMC90 = 1	PFCE90 = 0	PFC90 = 1		
	TXDA1	输出	P90 = 无须设置	PMC90 = 1	PFCE90 = 1	PFC90 = 0		
	SDA02	I/O	P90 = 无须设置	PMC90 = 1	PFCE90 = 1	PFC90 = 1	PF90 (PF9) = 1	
	A1	输出	P91 = 无须设置	PMC91 = 1	PFCE91 = 0	PFC91 = 0	注 1	
P91	KR7	输入	P91 = 无须设置	PMC91 = 1	PFCE91 = 0	PFC91 = 1		
	RXDA1/KR7 ²	输入	P91 = 无须设置	PMC91 = 1	PFCE91 = 1	PFC91 = 0		
	SCL02	I/O	P91 = 无须设置	PMC91 = 1	PFCE91 = 1	PFC91 = 1	PF91 (PF9) = 1	

注 1. 当用作复用功能 A0 至 A15 引脚时，需要立刻将 PMC9 寄存器的全部 16 位设置为 FFFFH。

2. RXDA1 和 KR7 引脚切勿同时使用。使用 RXDA1 引脚就不要使用 KR7 引脚。当使用 KR7 引脚时，不要使用 RXDA1 引脚 (推荐将 PFC91 位置 1 以及将 PFCE91 位清 0)。

表 4-15. 端口引脚用作复用功能引脚 (5/7)

引脚名称	复用功能		Pn寄存器的 Pnx位	PMn寄存器的 PMnx位	PFCE寄存器的 PFCEn位	PFCn寄存器的 PFCn位	其他位 (寄存器)
	名称	I/O					
P92	A2	输出	P92 = 无须设置	PMC92 = 1	PFCE92 = 0	PFC92 = 0	注
	TIP41	输入	P92 = 无须设置	PMC92 = 1	PFCE92 = 0	PFC92 = 1	
	TOP41	输出	P92 = 无须设置	PMC92 = 1	PFCE92 = 1	PFC92 = 0	
P93	A3	输出	P93 = 无须设置	PMC93 = 1	PFCE93 = 0	PFC93 = 0	注
	TIP40	输入	P93 = 无须设置	PMC93 = 1	PFCE93 = 0	PFC93 = 1	
	TOP40	输出	P93 = 无须设置	PMC93 = 1	PFCE93 = 1	PFC93 = 0	
P94	A4	输出	P94 = 无须设置	PMC94 = 1	PFCE94 = 0	PFC94 = 0	注
	TIP31	输入	P94 = 无须设置	PMC94 = 1	PFCE94 = 0	PFC94 = 1	
	TOP31	输出	P94 = 无须设置	PMC94 = 1	PFCE94 = 1	PFC94 = 0	
P95	A5	输出	P95 = 无须设置	PMC95 = 1	PFCE95 = 0	PFC95 = 0	注
	TIP30	输入	P95 = 无须设置	PMC95 = 1	PFCE95 = 0	PFC95 = 1	
	TOP30	输出	P95 = 无须设置	PMC95 = 1	PFCE95 = 1	PFC95 = 0	
P96	A6	输出	P96 = 无须设置	PMC96 = 1	PFCE96 = 0	PFC96 = 0	注
	TIP21	输入	P96 = 无须设置	PMC96 = 1	PFCE96 = 1	PFC96 = 0	
	TOP21	输出	P96 = 无须设置	PMC96 = 1	PFCE96 = 1	PFC96 = 1	
P97	A7	输出	P97 = 无须设置	PMC97 = 1	PFCE97 = 0	PFC97 = 0	注
	SIB1	输入	P97 = 无须设置	PMC97 = 1	PFCE97 = 0	PFC97 = 1	
	TIP20	输入	P97 = 无须设置	PMC97 = 1	PFCE97 = 1	PFC97 = 0	
P98	TOP20	输出	P97 = 无须设置	PMC97 = 1	PFCE97 = 1	PFC97 = 1	
	A8	输出	P98 = 无须设置	PMC98 = 1		PFC98 = 0	注
	SOB1	输出	P98 = 无须设置	PMC98 = 1		PFC98 = 1	
P99	A9	输出	P99 = 无须设置	PMC99 = 1		PFC99 = 0	注
	SCKB1	I/O	P99 = 无须设置	PMC99 = 1		PFC99 = 1	

注 当设置 A0 至 A15 引脚作为复用功能时，需要立刻将 PMC9 寄存器的全部 16 位设置为 FFFFH。

表 4-15. 端口引脚用作复用功能引脚 (6/7)

引脚名称	复用功能		Pn寄存器的 Pnx位	PMn寄存器的 PMnx位	PMC寄存器的 PMCx位	PFCE寄存器的 PFCEn位	PFC寄存器的 PFCn位	其他位 (寄存器)
	名称	I/O						
P910	A10	输出	PM910 = 无须设置	PM910 = 无须设置	PMC910 = 1		PFC910 = 0	注
	SIB3	输入	PM910 = 无须设置	PM910 = 无须设置	PMC910 = 1		PFC910 = 1	
P911	A11	输出	PM911 = 无须设置	PM911 = 无须设置	PMC911 = 1		PFC911 = 0	注
	SOB3	输出	PM911 = 无须设置	PM911 = 无须设置	PMC911 = 1		PFC911 = 1	
P912	A12	输出	PM912 = 无须设置	PM912 = 无须设置	PMC912 = 1		PFC912 = 0	注
	SKCB3	I/O	PM912 = 无须设置	PM912 = 无须设置	PMC912 = 1		PFC912 = 1	
P913	A13	输出	PM913 = 无须设置	PM913 = 无须设置	PMC913 = 1		PFC913 = 0	注
	INTP4	输入	PM913 = 无须设置	PM913 = 无须设置	PMC913 = 1		PFC913 = 1	
P914	A14	输出	PM914 = 无须设置	PM914 = 无须设置	PMC914 = 1	PFCE914 = 0	PFC914 = 0	注
	INTP5	输入	PM914 = 无须设置	PM914 = 无须设置	PMC914 = 1	PFCE914 = 0	PFC914 = 1	
	TIP51	输入	PM914 = 无须设置	PM914 = 无须设置	PMC914 = 1	PFCE914 = 1	PFC914 = 0	
	TOR51	输出	PM914 = 无须设置	PM914 = 无须设置	PMC914 = 1	PFCE914 = 1	PFC914 = 1	
P915	A15	输出	PM915 = 无须设置	PM915 = 无须设置	PMC915 = 1	PFCE915 = 0	PFC915 = 0	注
	INTP6	输入	PM915 = 无须设置	PM915 = 无须设置	PMC915 = 1	PFCE915 = 0	PFC915 = 1	
	TIP50	输入	PM915 = 无须设置	PM915 = 无须设置	PMC915 = 1	PFCE915 = 1	PFC915 = 0	
	TOR50	输出	PM915 = 无须设置	PM915 = 无须设置	PMC915 = 1	PFCE915 = 1	PFC915 = 1	
PCM0	WAIT	输入	PCM0 = 无须设置	PCM0 = 无须设置	PMCCM0 = 1			
PCM1	CLKOUT	输出	PCM1 = 无须设置	PCM1 = 无须设置	PMCCM1 = 1			
PCM2	HLDK	输出	PCM2 = 无须设置	PCM2 = 无须设置	PMCCM2 = 1			
PCM3	HLDKQ	输入	PCM3 = 无须设置	PCM3 = 无须设置	PMCCM3 = 1			
PCT0	WR0	输出	PCT0 = 无须设置	PCT0 = 无须设置	PMCCM0 = 1			
PCT1	WR1	输出	PCT1 = 无须设置	PCT1 = 无须设置	PMCCM1 = 1			
PCT4	RD	输出	PCT4 = 无须设置	PCT4 = 无须设置	PMCCM4 = 1			
PCT6	ASTB	输出	PCT6 = 无须设置	PCT6 = 无须设置	PMCCM6 = 1			

注 当不将引脚作为 INTP4 至 INTP6 引脚使用时，禁止边沿检测 (将 INTF9H 寄存器的 INTF9n 位以及 INTR9H 寄存器的 INTR9n 位清 0 (n = 13 至 15))。

表 4-15. 端口引脚用作复用功能引脚 (7/7)

引脚名称	复用功能		Pn寄存器的 Pnx位	PM寄存器的 PMnx位	PM0寄存器的 PM0x位	PFCE寄存器的 PFCEn位	PFC寄存器的 PFCn位	其他位 (寄存器)
	名称	I/O						
PDH0	A16	输出	PDH0 = 无须设置	PMDH0 = 无须设置	PMDH0 = 1			
PDH1	A17	输出	PDH1 = 无须设置	PMDH1 = 无须设置	PMDH1 = 1			
PDH2	A18	输出	PDH2 = 无须设置	PMDH2 = 无须设置	PMDH2 = 1			
PDH3	A19	输出	PDH3 = 无须设置	PMDH3 = 无须设置	PMDH3 = 1			
PDH4	A20	输出	PDH4 = 无须设置	PMDH4 = 无须设置	PMDH4 = 1			
PDH5	A21	输出	PDH5 = 无须设置	PMDH5 = 无须设置	PMDH5 = 1			
PDL0	AD0	I/O	PDL0 = 无须设置	PMDL0 = 无须设置	PMDL0 = 1			
PDL1	AD1	I/O	PDL1 = 无须设置	PMDL1 = 无须设置	PMDL1 = 1			
PDL2	AD2	I/O	PDL2 = 无须设置	PMDL2 = 无须设置	PMDL2 = 1			
PDL3	AD3	I/O	PDL3 = 无须设置	PMDL3 = 无须设置	PMDL3 = 1			
PDL4	AD4	I/O	PDL4 = 无须设置	PMDL4 = 无须设置	PMDL4 = 1			
PDL5	AD5	I/O	PDL5 = 无须设置	PMDL5 = 无须设置	PMDL5 = 1			
	FLMD [†]	输入	PDL5 = 无须设置	PMDL5 = 无须设置	PMDL5 = 无须设置			
PDL6	AD6	I/O	PDL6 = 无须设置	PMDL6 = 无须设置	PMDL6 = 1			
PDL7	AD7	I/O	PDL7 = 无须设置	PMDL7 = 无须设置	PMDL7 = 1			
PDL8	AD8	I/O	PDL8 = 无须设置	PMDL8 = 无须设置	PMDL8 = 1			
PDL9	AD9	I/O	PDL9 = 无须设置	PMDL9 = 无须设置	PMDL9 = 1			
PDL10	AD10	I/O	PDL10 = 无须设置	PMDL10 = 无须设置	PMDL10 = 1			
PDL11	AD11	I/O	PDL11 = 无须设置	PMDL11 = 无须设置	PMDL11 = 1			
PDL12	AD12	I/O	PDL12 = 无须设置	PMDL12 = 无须设置	PMDL12 = 1			
PDL13	AD13	I/O	PDL13 = 无须设置	PMDL13 = 无须设置	PMDL13 = 1			
PDL14	AD14	I/O	PDL14 = 无须设置	PMDL14 = 无须设置	PMDL14 = 1			
PDL15	AD15	I/O	PDL15 = 无须设置	PMDL15 = 无须设置	PMDL15 = 1			

注 由于只需要在闪存编程模式下设置此引脚，因此不需要操作端口控制寄存器。详细信息请参考 **第 26 章 FLASH 存储器**。

4.6 注意事项

4.6.1 设置端口引脚的注意事项

- (1) 在 V850ES/JG2 系列中，通用端口功能和一些周边功能 I/O 共用一个引脚。通过设置 **PMCn** 寄存器来切换通用端口 (端口模式) 和周边功能 I/O 引脚 (复用功能模式)。关于此寄存器的设置顺序，需要注意以下几点问题。

(a) 从端口模式切换到复用功能模式的注意事项

要从端口模式切换到复用功能模式需要按照以下顺序执行。

- | | |
|--|-----------|
| <1> 设置 PFn 寄存器 [※] ： | N 沟道开漏设置 |
| <2> 设置 PFCn 和 PFCEn 寄存器： | 复用功能选择 |
| <3> 将 PMCn 寄存器对应的位置 1： | 切换到复用功能模式 |

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

以下为具体的举例。

注 针对 N 沟道开漏输出引脚

注意事项 无论是端口模式/复用功能模式，读写 **Pn** 寄存器如下。

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

[例] SCL01 引脚设置样例

SCL01 引脚与 P41/SOB0 引脚是一组复用功能引脚。使用 **PMC4**，**PFC4**，和 **PF4** 寄存器选择引脚有效功能。

PMC41 位	PFC41 位	PF41 位	有效引脚功能
0	不用考虑	1	P41 (输出端口模式， N 沟道开漏输出)
1	0	1	SOB0 输出 (N 沟道开漏输出)
	1	1	SCL01 I/O (N 沟道开漏输出)

按照以下顺序将 P41 引脚切换为 SCL01 引脚可能会发生的故障如下所示。

设置顺序	设定内容	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)	Hi-Z
<2>	PMC41 位 ← 1	SOB0 输出	低电平 (基于 CSIB0 设定为高电平)
<3>	PFC41 位 ← 1	SCL01 I/O	高电平 (CMOS 输出)
<4>	PF41 位 ← 1	SCL01 I/O	Hi-Z (N 沟道开漏输出)

在步骤<2>，由于复用功能 SOB0 输出会输出到此引脚，因此 I²C 通讯可能会受影响。在<2> 或 <3>的 CMOS 输出期间可能会产生多余的电流。

(b) 复用功能模式 (输入)的注意事项

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

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

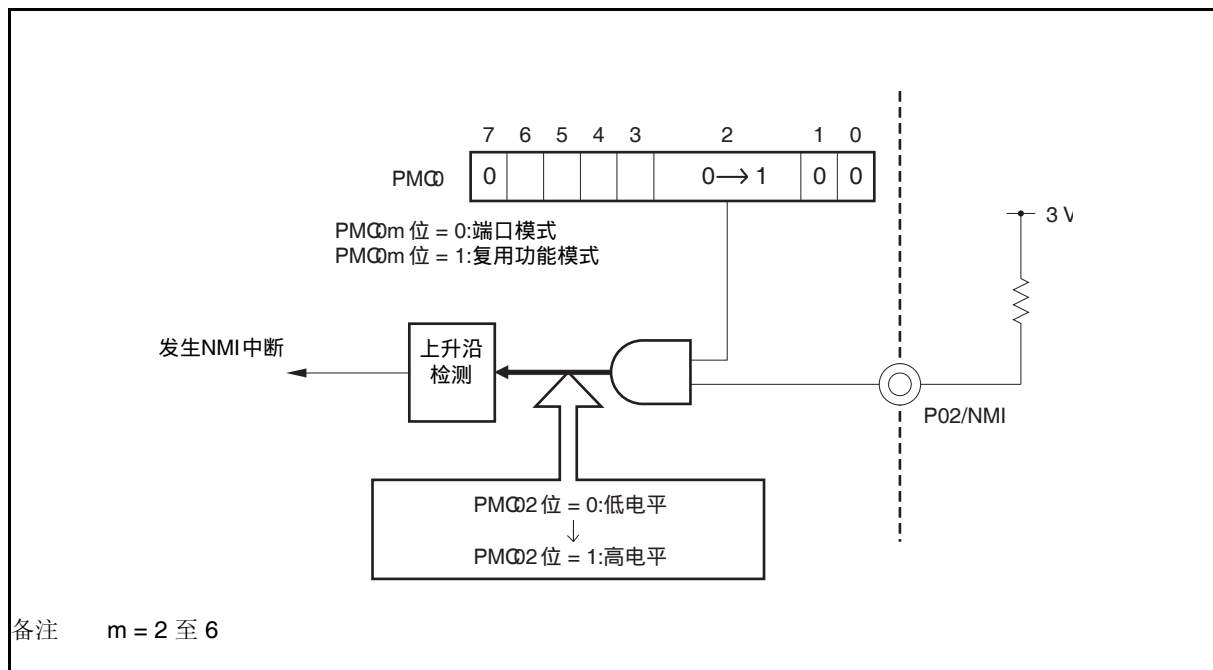
例 1 和例 2 为具体的例子。

[例 1] 从通用端口 (P02) 切换到外部中断引脚 (NMI)

当 P02/NMI 引脚按图 4-33 上拉，在 NMI 引脚边沿检测设置中指定上升沿，尽管在 P02 引脚切换到 NMI 引脚期间(PMC02 位 = 0 → 1)，NMI 引脚连续输入高电平，但仍旧像从低电平变为高电平一样检测为上升沿，由此产生 NMI 中断。

要避免以上情况发生，需要从 P02 引脚切换到 NMI 引脚后再设置 NMI 引脚的有效沿。

图 4-33. 从 P02 切换到 NMI 的例子(不正确)



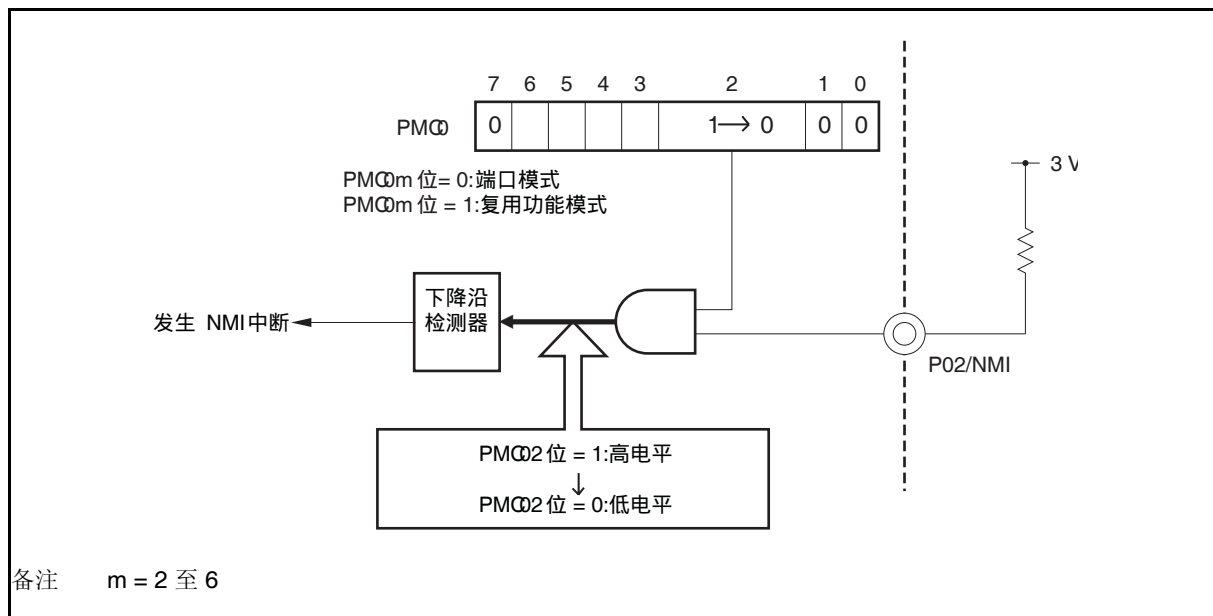
[例 2]

从外部引脚 (NMI) 切换到通用端口 (P02)

当 P02/NMI 引脚按图 4-34 上拉, 在 NMI 引脚边沿检测设置中指定下降沿, 尽管在 NMI 引脚切换到 P02 引脚期间 (PMC02 位 = 1 → 0), NMI 引脚连续输入高电平, 但仍像从高电平变为低电平一样检测为下降沿, 由此产生 NMI 中断。

要避免以上情况发生, 需要在切换到 P02 引脚前将 NMI 引脚边沿检测设置为“无边沿检测”。

图 4-34. 从 NMI 切换到 P02 的例子 (不正确)



- (2) 在端口模式, PFn.PFnm 位只在输出模式下有效 (PMn.PMnm 位 = 0)。在输入模式下 (PMnm 位 = 1), PFn 位的值不会映射到缓冲器中。

4.6.2 端口 n 寄存器 (Pn) 位操作指令的注意事项

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

因此，当端口从输入模式切换到输出模式时，推荐重写输出锁存。

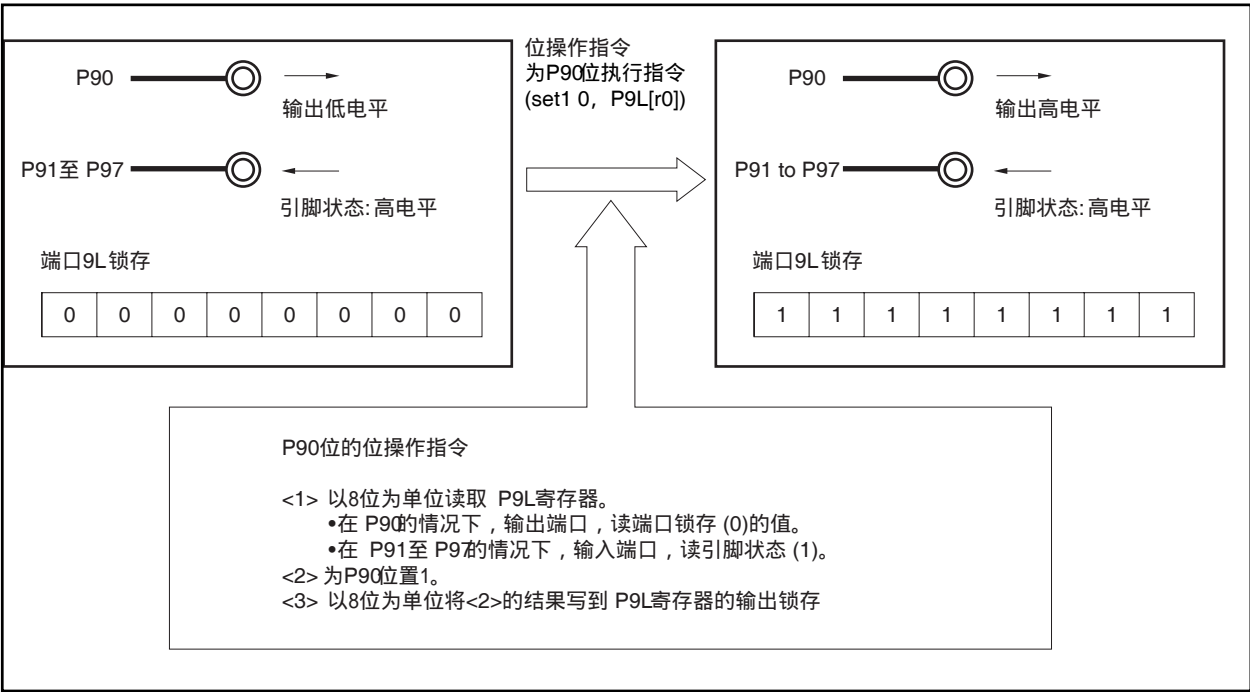
<例> 当 P90 引脚为输出端口，P91 至 P97 引脚为输入端口 (全部引脚状态为高电平)，端口锁存的值为 00H，如果通过位操作指令将 P90 引脚的输出从低电平变为高电平，则端口锁存的值为 FFH。
说明：PMnm 比特 = 1 的端口的 Pn 寄存器的写/读的对象,是各自输出锁/引脚状态。

V850ES/JG2 位操作指令按照以下步骤执行。

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

在步骤 <1>中，当 P91 到 P97 引脚作为输入端口时引脚状态被读取，P90 引脚作为输出端口时，输出锁存 (0) 的值被读取，如果此时 P91 至 P97 引脚的状态为高电平，则读取的值为 FEH。通过步骤 <2>此值变为 FFH。在步骤 <3>，FFH 写到输出锁存。

图 4-35. 位操作指令(P90 引脚)



4.6.3 片上调试引脚的注意事项

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

通过 $\overline{\text{RESET}}$ 引脚复位后, P05/INTP2/ $\overline{\text{DRST}}$ 引脚被初始化为片上调试引脚(DRST)。如果此时向 $\overline{\text{DRST}}$ 引脚输入高电平则被设置为片上调试模式, 因而 DCK, DMS, DDI, 及 DDO 引脚能够被使用。

如果不使用片上调试, 则需要采取以下措施。

- 将 OCDM 寄存器(特殊寄存器)的 OCDM0 位清(0)。

此时, 从通过 $\overline{\text{RESET}}$ 引脚复位释放, 到采取了以上措施期间, 将 P05/INTP2/ $\overline{\text{DRST}}$ 引脚固定为低电平。

采取以上措施之前, 如果将高电平输入到 $\overline{\text{DRST}}$ 引脚, 可能会导致发生故障 (CPU 死锁)。因此在处理 P05 引脚时要非常小心。

注意事项 通过 WDT2RES, 时钟监视器 (CLM), 或低电压检测 (LVI) 信号复位后, P05/INTP2/ $\overline{\text{DRST}}$ 引脚不初始化为片上调试引脚功能 (DRST)。OCDM 寄存器保持当前值。

4.6.4 P05/INTP2/ $\overline{\text{DRST}}$ 引脚的注意事项

P05/INTP2/ $\overline{\text{DRST}}$ 引脚有内部下拉电阻(30 k Ω 典型值)。通过 $\overline{\text{RESET}}$ 引脚复位后, 就会连接下拉电阻。当 OCDM0 位清 (0) 时, 下拉电阻断开。

4.6.5 当电源打开时 P10, P11, 和 P53 引脚的注意事项

当电源打开时, 即使在复位期间以下引脚也可能暂时输出未定义的电平。.

- P10/ANO0 引脚
- P11/ANO1 引脚
- P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 引脚

4.6.6 滞后性

在端口模式, 以下端口引脚没有滞后性。

P02 至 P06

P31 至 P35, P38, P39

P40 至 P42

P50 至 P55

P90 至 P97, P99, P910, P912 至 P915

第五章 总线控制功能

V850ES/JG2 系列产品提供了外部总线接口功能，该功能可以实现设备与外部存储器(ROM 和 RAM)以及外部 I/O 器件之间的连接。

5.1 特点

- 输出可以在一个最小传输时间为 3 个总线周期的多路总线和一个最小传输时间为 2 个总线周期的分离总线之间进行选择。
- 可选择 8 位/16 位数据总线
- 等待功能
 - 最大支持 7 个状态的可编程等待功能
 - 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
- 可连接最大 4MB 物理地址
- 可以使用不同于工作电压的电压来控制总线(例如 $\text{BV}_{\text{DD}} \leq \text{EV}_{\text{DD}} = \text{V}_{\text{DD}}$)。但是，在分离总线模式下或使用了 A20 和 A21 引脚时，必须满足 $\text{BV}_{\text{DD}} = \text{EV}_{\text{DD}} = \text{V}_{\text{DD}}$ 。

5.2 总线控制引脚

用于连接外部设备的引脚如下表所示。

表 5-1. 总线控制引脚 (多路总线)

总线控制引脚	复用引脚	I/O	功能
AD0 到 AD15	PDL0 到 PDL15	I/O	地址/数据总线
A16 到 A21	PDH0 到 PDH5	输出	地址总线
WAIT	PCM0	输入	外部等待控制
CLKOUT	PCM1	输出	内部系统时钟
WR0, WR1	PCT0, PCT1	输出	写入选通信号
RD	PCT4	输出	读取选通信号
ASTB	PCT6	输出	地址选通信号
HLD $\overline{\text{RQ}}$	PCM3	输入	总线保持控制
H $\overline{\text{LDAK}}$	PCM2	输出	

表 5-2. 外部控制引脚 (分离总线)

总线控制引脚	复用引脚	I/O	功能
AD0 到 AD15	PDL0 到 PDL15	I/O	数据总线
A0 到 A15	P90 到 P915	输出	地址总线
A16 到 A21	PDH0 到 PDH5	输出	地址总线
WAIT	PCM0	输入	外部等待控制
CLKOUT	PCM1	输出	内部系统时钟
WR0, WR1	PCT0, PCT1	输出	写入选通信号
RD	PCT4	输出	读取选通信号
HLD $\overline{\text{RQ}}$	PCM3	输入	总线保持控制
H $\overline{\text{LDAK}}$	PCM2	输出	

5.2.1 当访问内置 ROM，内置 RAM 或片上周边 I/O 时的引脚状态

当访问内置 ROM，内置 RAM 或片上周边 I/O 时，各引脚的状态如下表所示。

表 5-3. 当访问内置 ROM，内置 RAM 或片上周边 I/O 时的引脚状态

分离总线模式		多路总线模式	
地址总线 (A21 到 A0)	不确定	地址总线 (A21 到 A16)	不确定
数据总线 (AD15 到 AD0)	Hi-Z	地址/数据总线 (AD15 到 AD0)	不确定
控制信号	不起作用	控制信号	不起作用

注意事项 对内置 ROM 区域进行写访问时地址、数据以及控制信号的激活方式和对外部存储器区域进行访问时的激活方式是相同的。

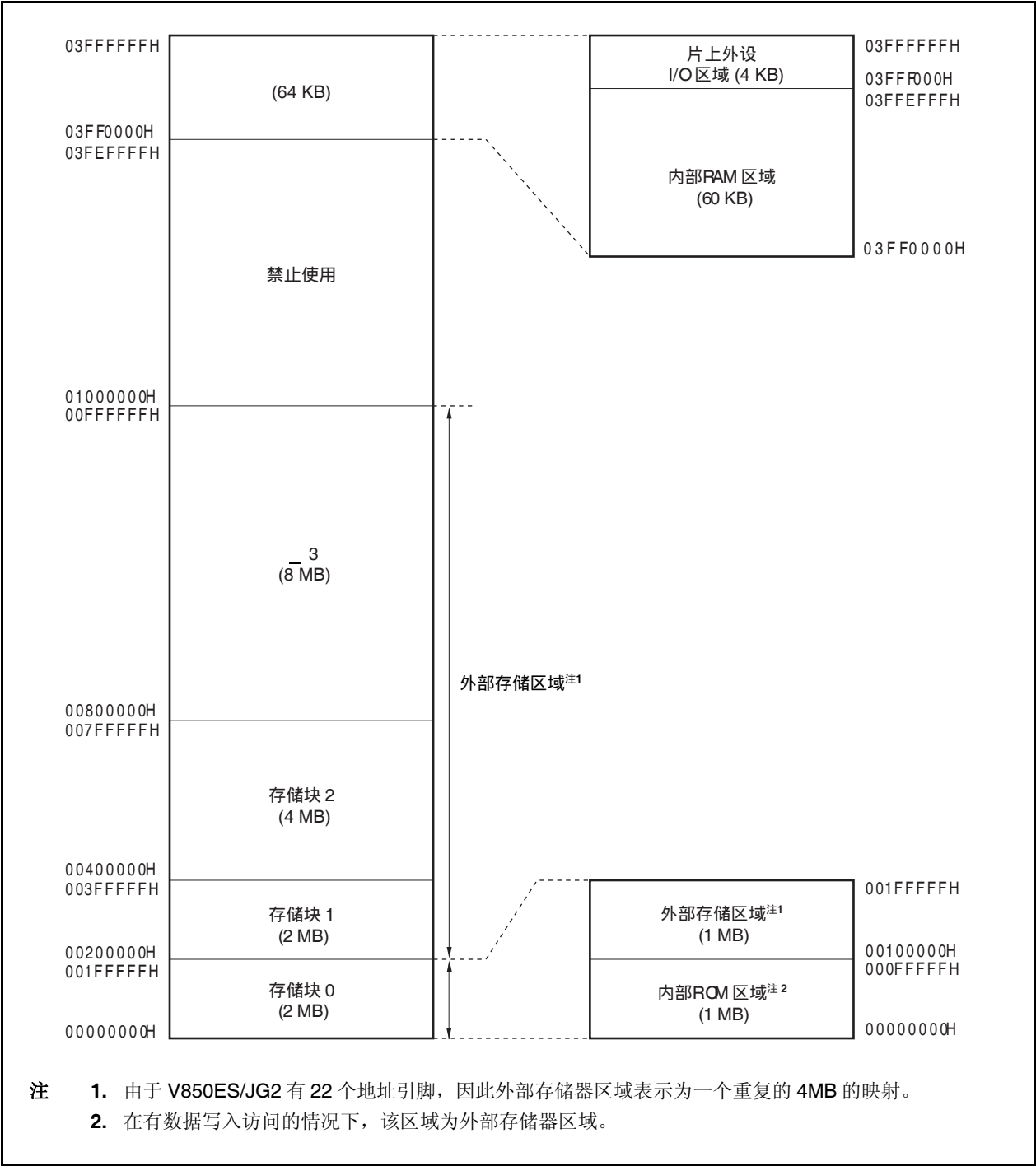
5.2.2 各操作模式下的引脚状态

V850ES/JG2 产品在各操作模式下引脚的状态，可参见 2.2 引脚状态。

5.3 存储器块(block)功能

16MB 的外部存储器空间由低到高被分为 2MB, 2MB, 4MB 和 8MB，共 4 个存储器块。这些存储器块的可编程等待功能和总线周期操作模式可以以块为单位独立控制。

图 5-1. 数据存储器映射: 物理地址



5.4 外部总线接口模式控制功能

V850ES/JG2 系列器件具有以下两种外部总线接口模式。

- 多路总线模式
- 分离总线模式

通过使用 EXIMC 寄存器可以选择两种这。

(1) 外部总线接口模式控制寄存器 (EXIMC)

可以对 EXIMC 寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后: 00H		R/W		地址: FFFFFFFBEH											
7		6		5		4		3		2		1		0	
EXIMC		0		0		0		0		0		0		SMSEL	
SMSEL		模式选择													
0		多路总线模式													
1		分离总线模式													

注意事项 请在对外部存储器进行访问之前从内置 ROM 或内置 RAM 进行对 EXIMC 寄存器的设置。
在设置了 EXIMC 寄存器之后，请务必插入一条 NOP 指令。

5.5 总线访问

5.5.1 访问所需的时钟数

下表中列出了通过总线访问各种设备时所需的基本时钟数。

区域 (总线宽度)	内置 ROM (32 位)	内置 RAM (32 位)	外部存储器 (16 位)
总线周期类型			
取指令 (正常访问)	1	1 ^{#1}	3 + n ^{#2}
取指令 (分支)	2	2 ^{#1}	3 + n ^{#2}
访问操作数	3	1	3 + n ^{#2}

- 注
1. 在与数据访问发生冲突时会增加 1 个时钟周期。
 2. 当使用分离总线模式时所需时钟数为 2 + n (n: 等待状态的数量)。

备注 单位: 时钟/访问

5.5.2 总线宽度设置功能

可以通过 BSC 寄存器对由存储器块 n 选择的各外部存储器区域进行设置。但总线宽度只可以在 8 位和 16 位之间进行选择。

V850ES/JG2 器件的外部存储器区域是由存储器块 0 到 3 选择的。

(1) 总线宽度配置寄存器 (BSC)

可以对 BSC 寄存器进行 16 位的读写操作。

复位输入将把该寄存器设置为 5555H。

注意事项 请在复位后对 BSC 寄存器进行设置，并且不要在设置完成后更改设置值。此外，在对 BSC 寄存器进行初始设置之前，请不要访问外部存储器区域。

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	BS10	0	BS00
	存储块 3		存储块 2		存储块 1		存储块 0	
BSn0	存储块n空间的数据总线宽度 (n = 0 to 3)							
0	8 位							
1	16 位							

注意事项 请务必将第 14, 12, 10 和 8 位设置为 1，将第 15, 13, 11, 9, 7, 5, 3 和 1 位清零。

5.5.3 访问的总线宽度

V850ES/JG2 系列器件以 8 位、16 位或 32 位为单位对片上周边 I/O 和外部存储器进行访问。总线宽度如下所示。

- 片上周边 I/O 的总线宽度固定为 16 位。
- 外部存储器的总线宽度可在 8 位或 16 位之间选择(通过 BSC 寄存器)。

以下将说明访问上述设备时的操作。所有数据的访问都是由低到高顺序进行的。

V850ES/JG2 只支持小端(little-endian)格式。

图 5-2. 以字为单位的小端地址

31	24	23	16	15	8	7	0
000BH	000AH	0009H	0008H				
0007H	0006H	0005H	0004H				
0003H	0002H	0001H	0000H				

(1) 数据空间

V850ES/JG2 具有地址偏移(address misalign)功能。

通过该功能，不论数据以什么格式(以字为单位或半字为单位)存在，都可以将其放置在任意的地址上。但是，如果字数据或半字数据没有和边界对齐，那么访问它们时至少需要产生两次总线周期，导致总线效率降低。

(a) 访问半字长的数据

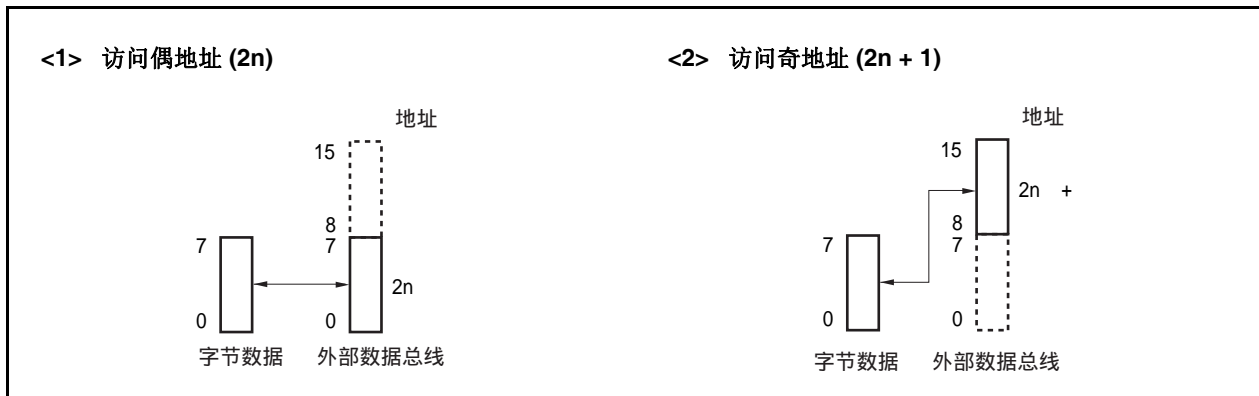
如果该数据地址的最低有效位为 1，那么对其访问时将产生两次字节长的总线周期。

(b) 访问一字长的数据

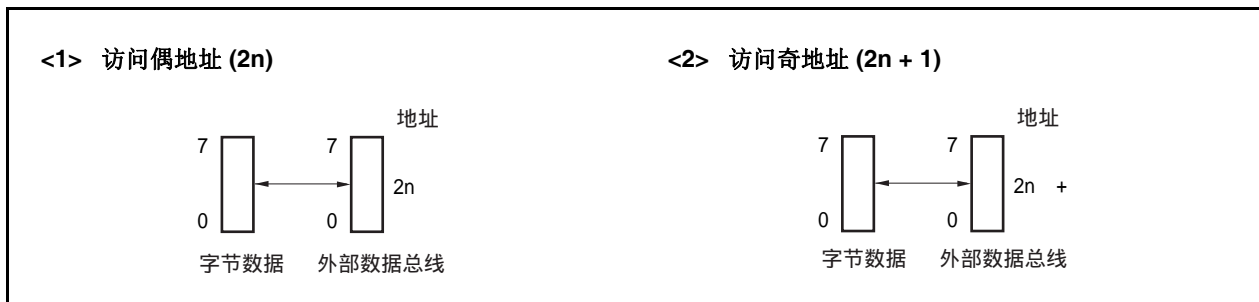
- 如果该数据地址的最低有效位为 1，那么对其访问时将顺序产生一个字节长的总线周期、一个半字长的总线周期和一个字节长的总线周期。
- 如果该数据地址的低 2 位为 10，那么对其访问时将产生两次半字长的总线周期。

(2) 字节访问 (8 位)

(a) 16 位数据总线宽度

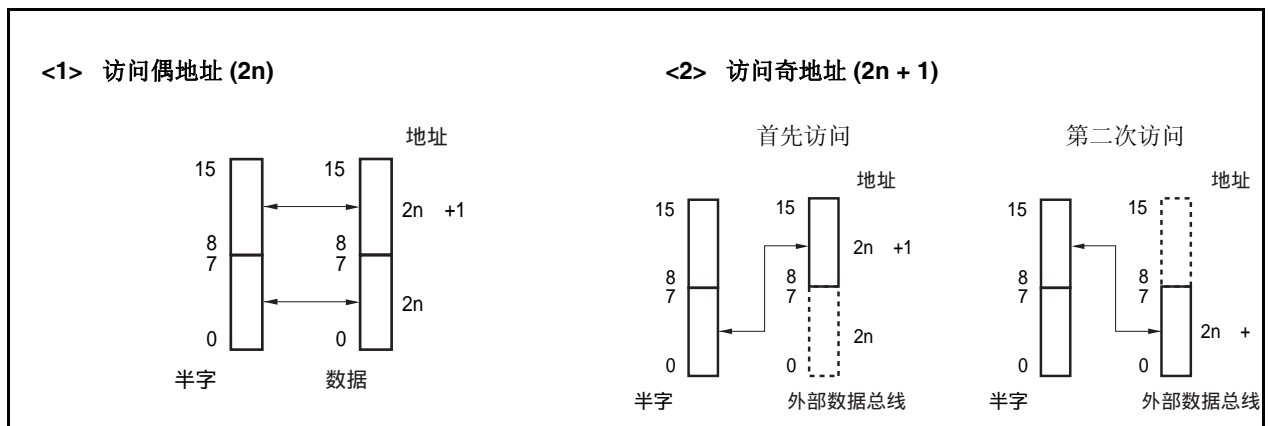


(b) 8 位数据总线宽度

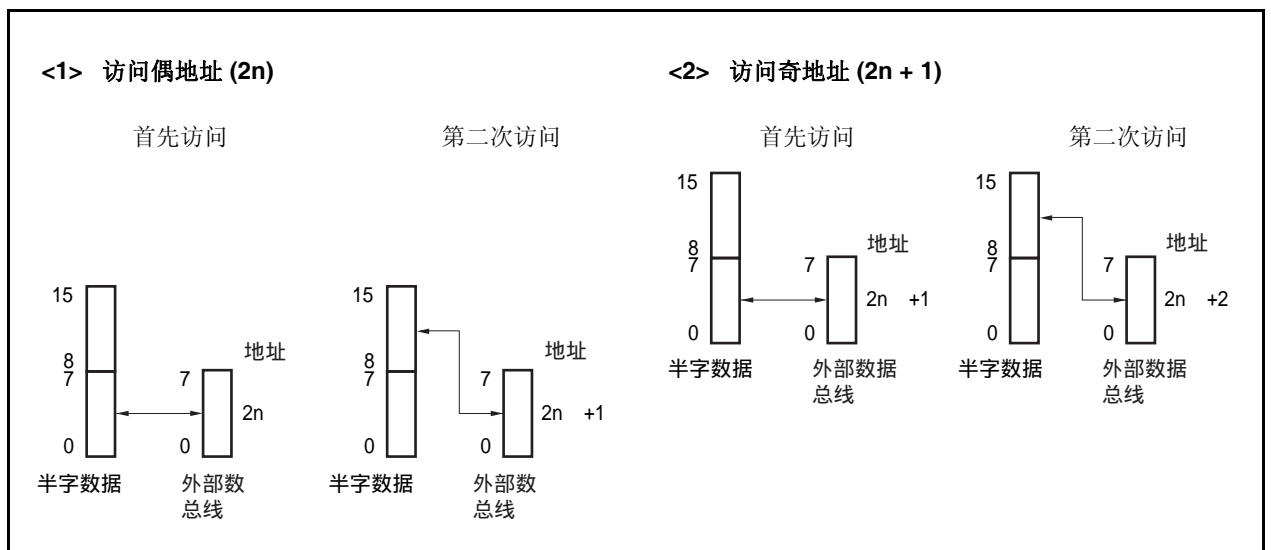


(3) 半字访问 (16 位)

(a) 16 位数据总线宽度

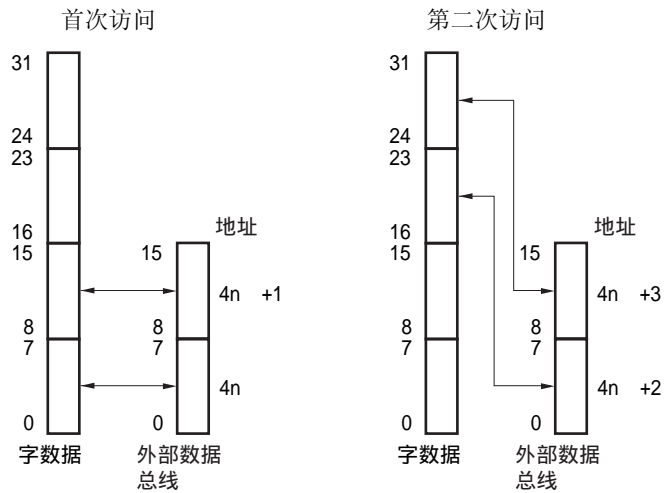
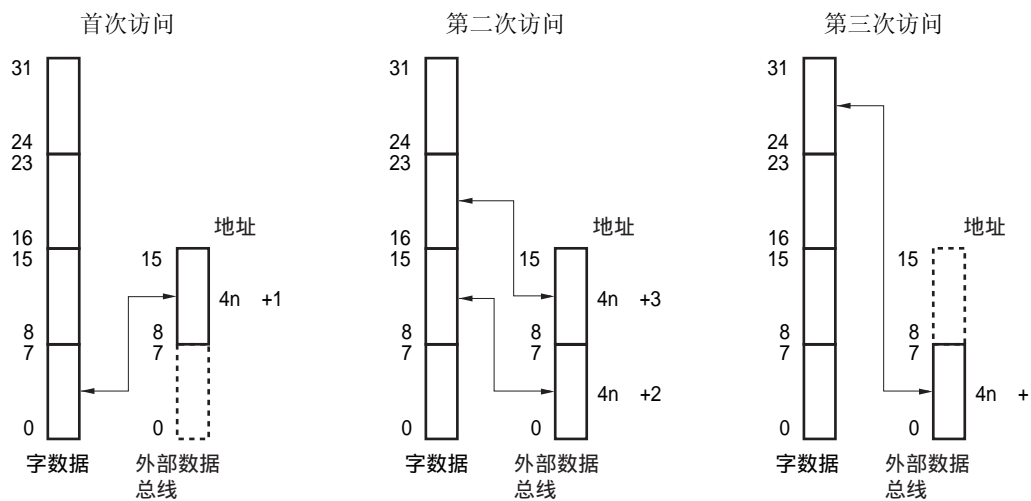


(b) 8 位数据总线宽度

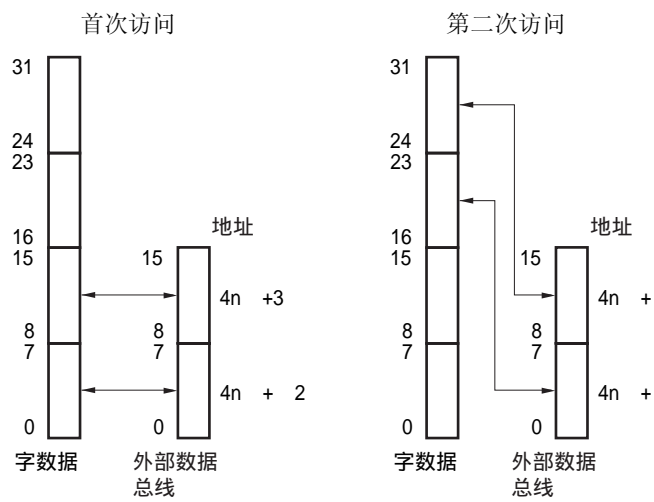
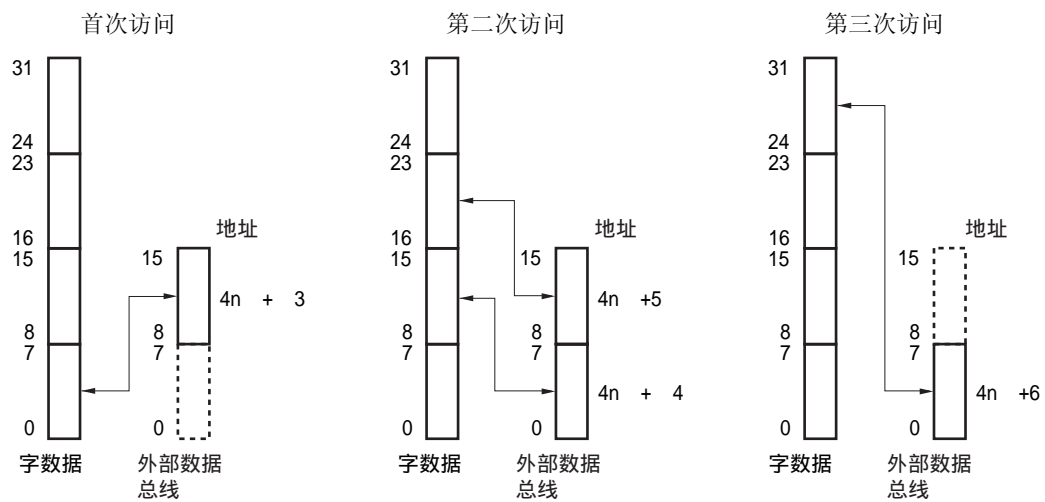


(4) 字访问 (32 bits)

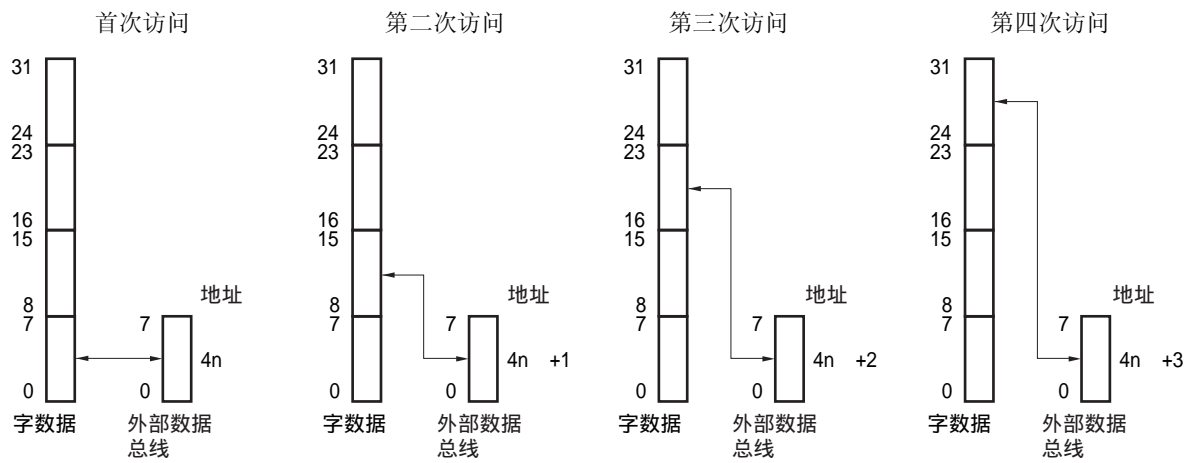
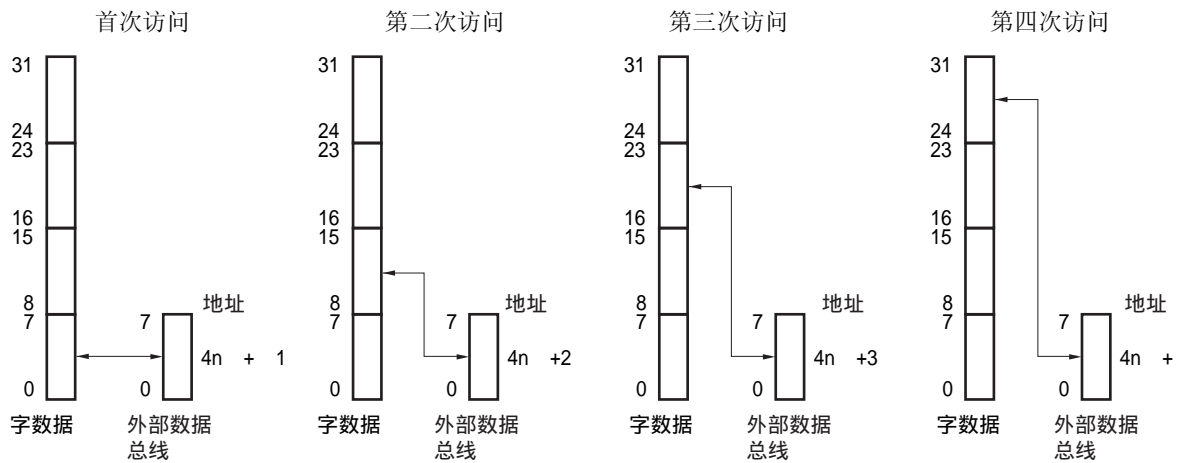
(a) 16 位数据总线宽度 (1/2)

<1> 访问地址 ($4n$)<2> 访问地址 ($4n + 1$)

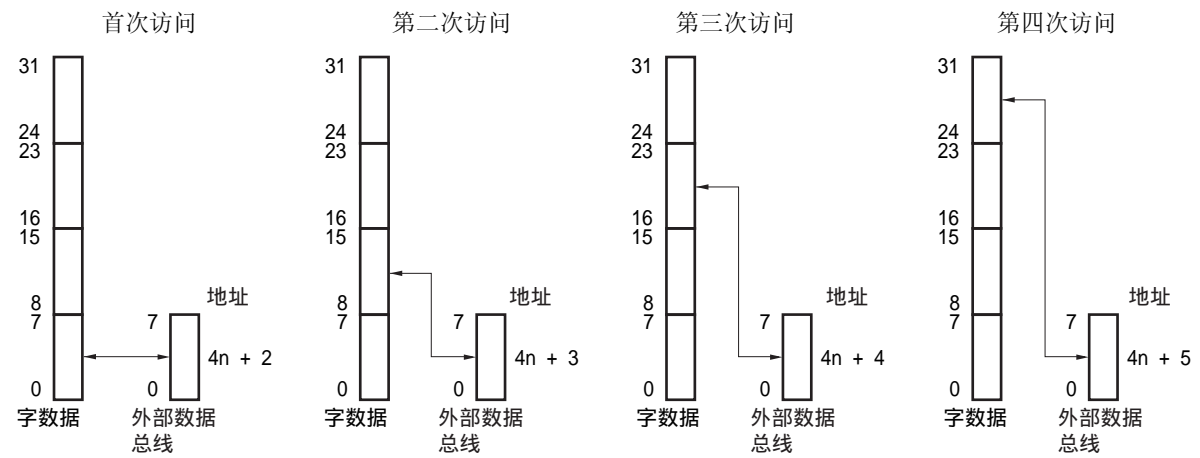
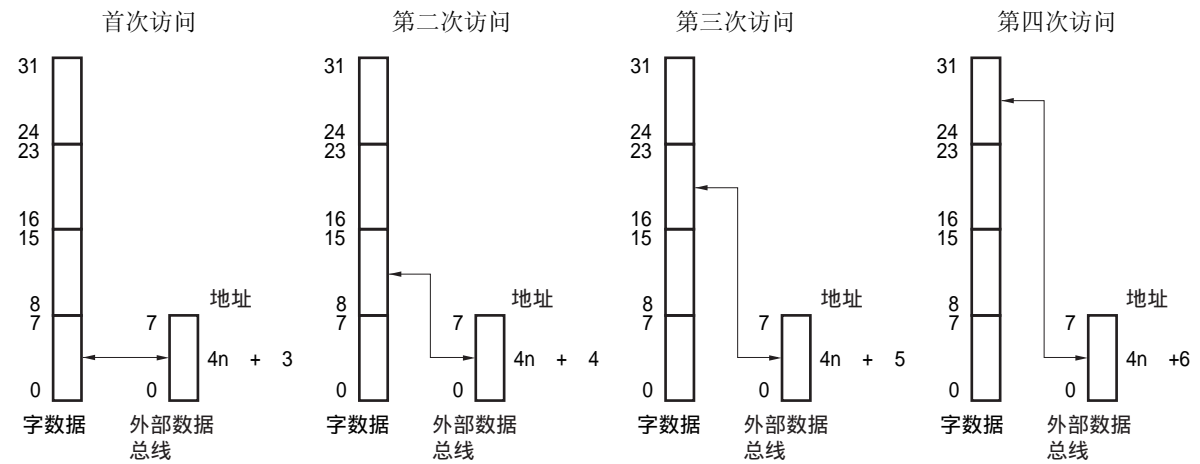
(a) 16 位数据总线宽度 (2/2)

<3> 访问地址 ($4n + 2$)<4> 访问地址 ($4n + 3$)

(b) 8 位数据总线宽度 (1/2)

<1> 访问地址 ($4n$)<2> 访问地址 ($4n + 1$)

(b) 8 位数据总线宽度 (2/2)

<3> 访问地址 ($4n + 2$)<4> 访问地址 ($4n + 3$)

5.6 等待功能

5.6.1 可编程等待功能

(1) 数据等待控制寄存器 0 (DWC0)

为了实现与低速存储器或 I/O 设备的通信，每个存储器块空间可以插入最多 7 个等待状态。

插入的等待状态的个数可由 **DWC0** 寄存器来控制。在系统复位后，各存储器块会被立即插入 7 个等待状态。

可以对 **DWC0** 寄存器进行 16 位的读写操作。

复位输入将把该寄存器设置为 7777H。

- 注意事项**
1. 内置 **ROM** 和内置 **RAM** 不会受可编程等待功能的影响，对它们的访问不会被插入等待状态。片上周边 **I/O** 区域也同样不受可编程等待功能的影响，该区域的访问只会受片上周边 **I/O** 器件的等待控制的影响。
 2. 请在复位后对 **DWC0** 寄存器进行设置，并且不要在设置完成后更改设置值。此外，在对 **DWC0** 寄存器进行初始设置之前，请不要访问外部存储器区域。

复位后: 7777H		R/W		地址: FFFF484H				
DWC0	15	14	13	12	11	10	9	8
	0	DW32	DW31	DW30	0	DW22	DW21	DW20
	存储块 3				存储块 2			
	7	6	5	4	3	2	1	0
	0	DW12	DW11	DW10	0	DW02	DW01	DW00
	存储块 1				存储块 0			

DWn2	DWn1	DWn0	插入到存储块n空间的等待状态的数量 (n = 0 ~ 3)
0	0	0	None
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 请务必将第 15, 11, 7 和 3 位清零。

5.6.2 外部等待功能

为了实现与超低速外部存储器、I/O 器件或异步系统的同步，可以通过外部等待引脚($\overline{\text{WAIT}}$)插入任意数量的等待状态。

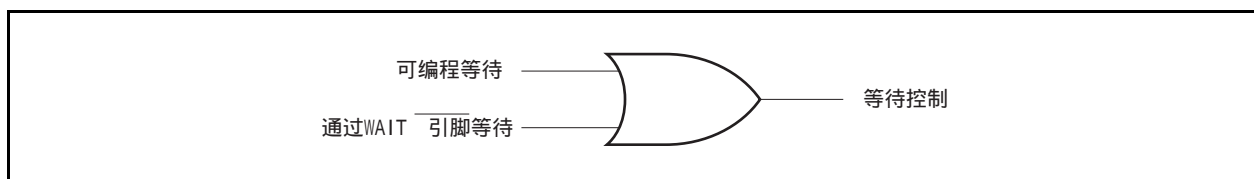
将 **PCMO** 引脚设置为复用功能模式以允许外部等待功能。

和可编程等待功能一样，对内置 **ROM**、内置 **RAM** 以及片上周边 I/O 空间中的任何区域进行访问都不会受外部等待功能的影响。

$\overline{\text{WAIT}}$ 信号可以与 **CLKOUT** 信号异步输入，在多路总线模式下， $\overline{\text{WAIT}}$ 信号在总线周期的 **T2** 和 **TW** 状态时的时钟下降沿处被采样。在分离总线模式下， $\overline{\text{WAIT}}$ 信号会在总线周期的 **T1** 和 **TW** 状态之后的时钟上升沿处被采样。如果没有满足采样时序的建立/保持时间(setup/hold time)，那么在下一状态前可能插入一个等待状态(也可能不插入等待状态)。

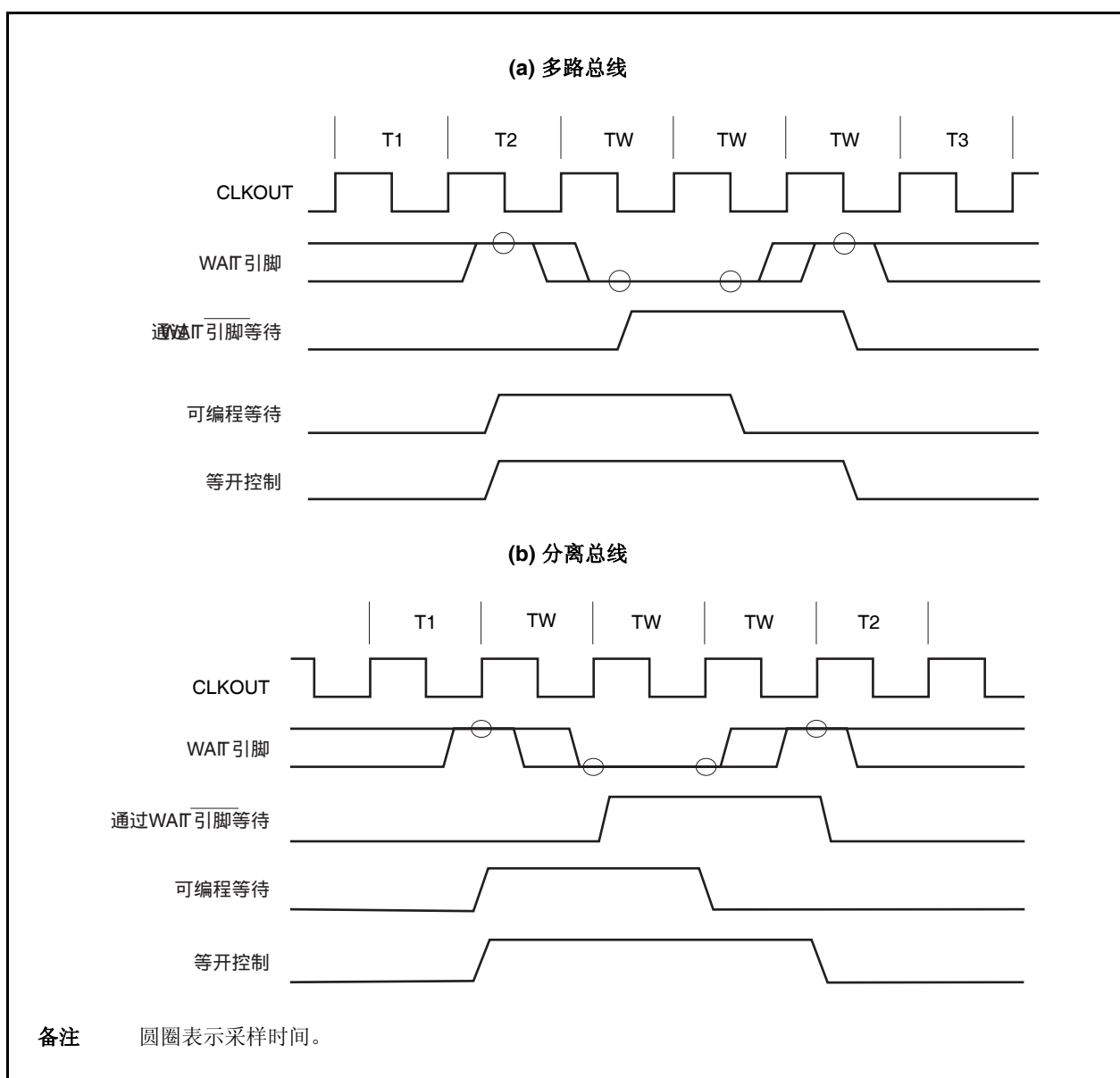
5.6.3 可编程等待和外部等待之间的关系

可编程等待所设置的等待周期数和 **WAIT** 引脚控制的等待周期数以"或"(OR)的方式决定最终插入的等待周期数。



例如，如果可编程等待和 **WAIT** 引脚信号的时序如下图所示，那么将在总线周期中插入 3 个等待状态。

图 5-3. 插入等待举例



5.6.4 可编程地址等待功能

通过 AWC 寄存器可以对向各总线周期插入的地址设立(address-setup)或地址保持(address-hold)等待进行设置。地址等待的插入针对每个存储器块区域(存储器块 0 到 3)进行设置。

插入一个地址设立等待时, T1 状态的时钟高电平将可以看作被延长了 1 个时钟周期。插入一个地址保持等待时, T1 状态的时钟低电平将可以看作被延长了 1 个时钟周期。

(1) 地址等待控制寄存器 (AWC)

可以对 AWC 寄存器进行 16 位的读写操作。

复位输入将把该寄存器设置为 FFFFH。

注意事项 1. 访问内置 ROM 区域, 内置 RAM 区域以及片上周边 I/O 区域时不会插入地址设立和地址保持等待周期。

2. 请在复位后对 AWC 寄存器进行设置, 并且不要在设置完成后更改设置值。此外, 在对 AWC 寄存器进行初始设置之前, 请不要访问外部存储器区域。

复位后：FFFFH		R/W		地址：FFFF488H												
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		AHW1		ASW1		AHW0		ASW0		
		存储块 3				存储块 2				存储块 1				存储块 0		

AHWn	定义地址保持等待的插入 (n = 0 to 3)														
0	不插入														
1	插入														

ASWn	定义地址保等待的插入 (n = 0 to 3)														
0	不插入														
1	插入														

注意事项 请务必将第 15 到第 8 位置 1。

5.7 空闲状态插入功能

为了更方便的与低速外部存储器进行通信，多路地址/数据总线模式下，可以在对每个存储器块空间的访问总线周期中的 **T3** 状态后插入一个空闲状态(**TI**)。分离总线模式下，可在 **T2** 状态后插入一个空闲状态(**TI**)。通过插入空闲状态，可以确保存储器在读取访问期间的数据输出浮动延迟时间(float delay time)。(无法在写入访问时插入空闲状态。)

可以通过 **BCC** 寄存器来设置是否插入空闲状态。

系统复位后会向所有区域插入一个空闲状态。

(1) 总线周期控制寄存器 (BCC)

可以对 **BCC** 寄存器进行 16 位的读写操作。

复位输入将把该寄存器设置为 **AAAAH**。

- 注意事项**
1. 内置 **ROM**，内置 **RAM** 和片上周边 **I/O** 区域不受空闲状态插入的影响。
 2. 请在复位后对 **BCC** 寄存器进行设置，并且不要在设置完成后更改设置值。此外，在对 **BCC** 寄存器进行初始设置之前，请不要访问外部存储器区域。

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	1	0	1	0	1	0
7		6		5		4		3		2		1		0	
BC31		0		BC21		0		BC11		0		BC01		0	
存储块 3				存储块 2				存储块 1				存储块 0			
BCn1		idle状态插入的定义 (n = 0 到 3)													
0		没有插入													
1		插入													

注意事项 请务必将第 15, 13, 11 和 9 位置 1，并将第 14, 12, 10, 8, 6, 4, 2 和 0 位清零。

5.8 总线保持功能

5.8.1 功能概述

将 $\overline{\text{PCM2}}$ 和 $\overline{\text{PCM3}}$ 引脚设置为复用功能可以使 $\overline{\text{HLDRQ}}$ 和 $\overline{\text{HLDK}}$ 功能生效。

若 $\overline{\text{HLDRQ}}$ 引脚变为低电平，则说明有其它总线主设备提出了总线控制请求，此时，总线控制权将被释放，外部地址/数据总线进入高阻状态(总线保持状态)。当总线控制请求被清除且 $\overline{\text{HLDRQ}}$ 引脚恢复高电平时，设备可以重新开始驱动外部总线引脚。

在总线保持期间，内置 ROM 和内置 RAM 上的程序会继续运行，直到产生对片上周边 I/O 寄存器或对外部存储器的访问为止。

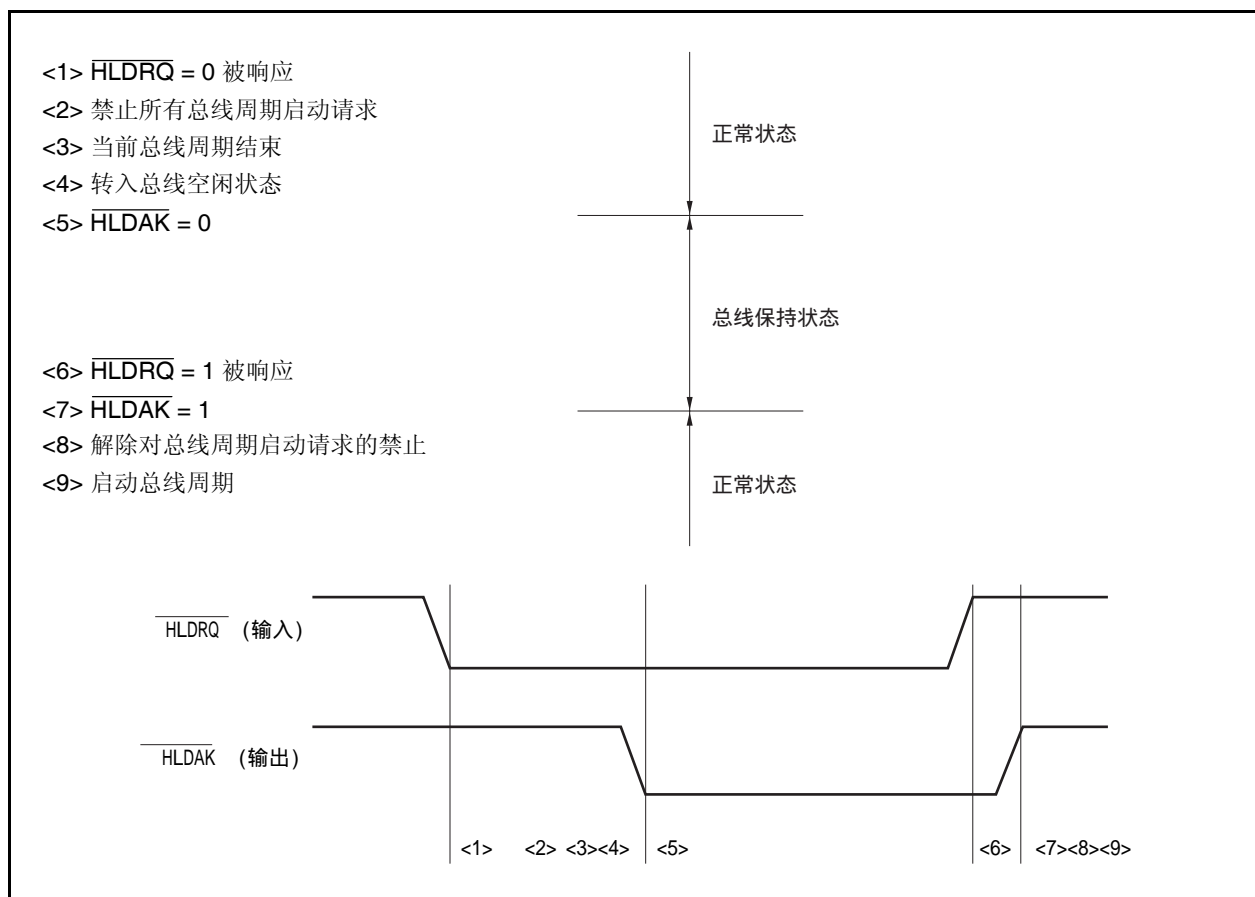
$\overline{\text{HLDK}}$ 引脚的低电平指示总线保持状态。总线保持功能的存在，使具有 2 个或更多总线主设备的多处理器系统配置成为可能。

需要注意的是，总线保持请求在由总线宽度功能(bus sizing function)或位操作指令引起的多重访问周期中不会被响应。

状态	数据总线宽度	访问类型	不响应总线保持请求的时刻
CPU 总线锁定 (CPU bus lock)	16 位	对偶地址进行字访问	第一次和第二次访问之间
		对奇地址进行字访问	第一次和第二次访问之间
			第二次和第三次访问之间
		对奇地址进行半字访问	第一次和第二次访问之间
	8 bits	字访问	第一次和第二次访问之间
			第二次和第三次访问之间
			第三次和第四次访问之间
		半字访问	第一次和第二次访问之间
位操作指令的读取-修改-写入访问	—	—	读取访问和写入访问之间

5.8.2 总线保持流程

总线保持状态转换流程如下所示。



5.8.3 节能模式下的操作

由于在 STOP, IDLE1 和 IDLE2 模式下内部系统时钟停止工作, 所以即使 $\overline{\text{HLD RQ}}$ 引脚为低电平也不会进入总线保持状态。

HALT 模式下, $\overline{\text{HLD A K}}$ 引脚会在 $\overline{\text{HLD RQ}}$ 引脚变为低电平后立刻变为低电平, 同时进入总线保持状态。当 $\overline{\text{HLD RQ}}$ 引脚变为高电平, $\overline{\text{HLD A K}}$ 引脚也会随之变为高电平, 同时退出总线保持状态。

5.9 总线优先级

外部总线周期中执行的操作有总线保持、DMA 传送，操作数访问，取指令(分支)、以及取指令(连续)。
其中，总线保持具有最高的优先级，其次分别为 DMA 传送，操作数访问，取指令(分支)以及取指令(连续)。
在读取-修改-写入访问过程中，读取和写入访问之间可能插入另一条取指令操作。
由于总线宽度的限制，进行了 2 次或更多次访问操作的指令执行过程中不会插入取指令或总线保持操作。

表 5-4. 总线优先级

优先级	外部总线周期	总线主设备
高	总线保持	外部设备
	DMA 传输	DMAC
	操作数访问	CPU
低	取指令(分支)	CPU
	取指令(连续)	CPU

5.10 总线时序

图 5-4. 多路总线读取时序 (总线宽度: 16 位, 16-位访问)

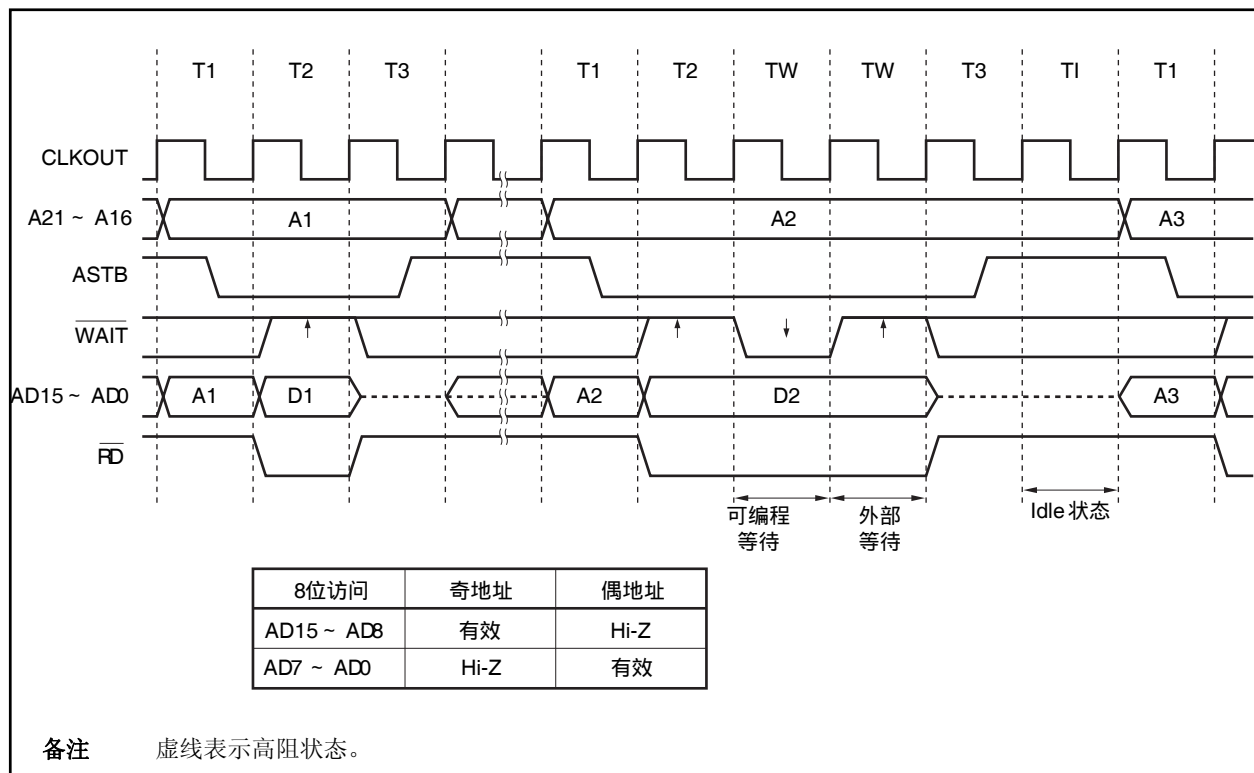


图 5-5. 多路总线读取时序 (总线宽度: 8 位)

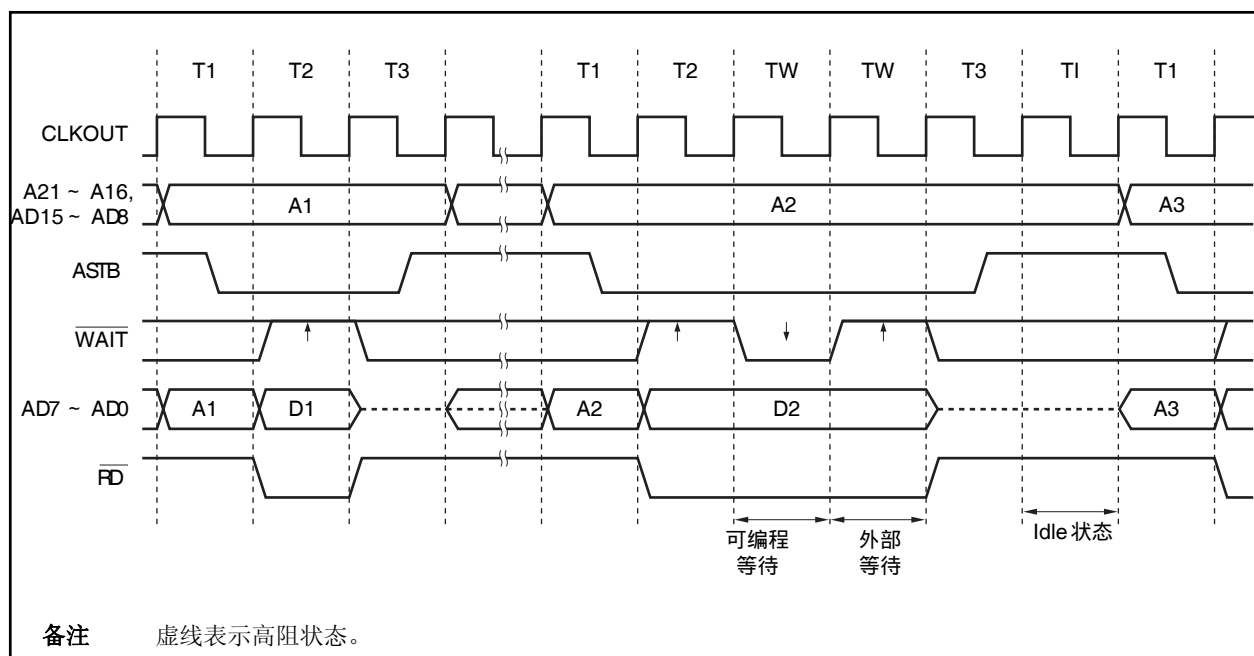


图 5-6. 多路总线写入时序 (总线宽度: 16 位, 16-位访问)

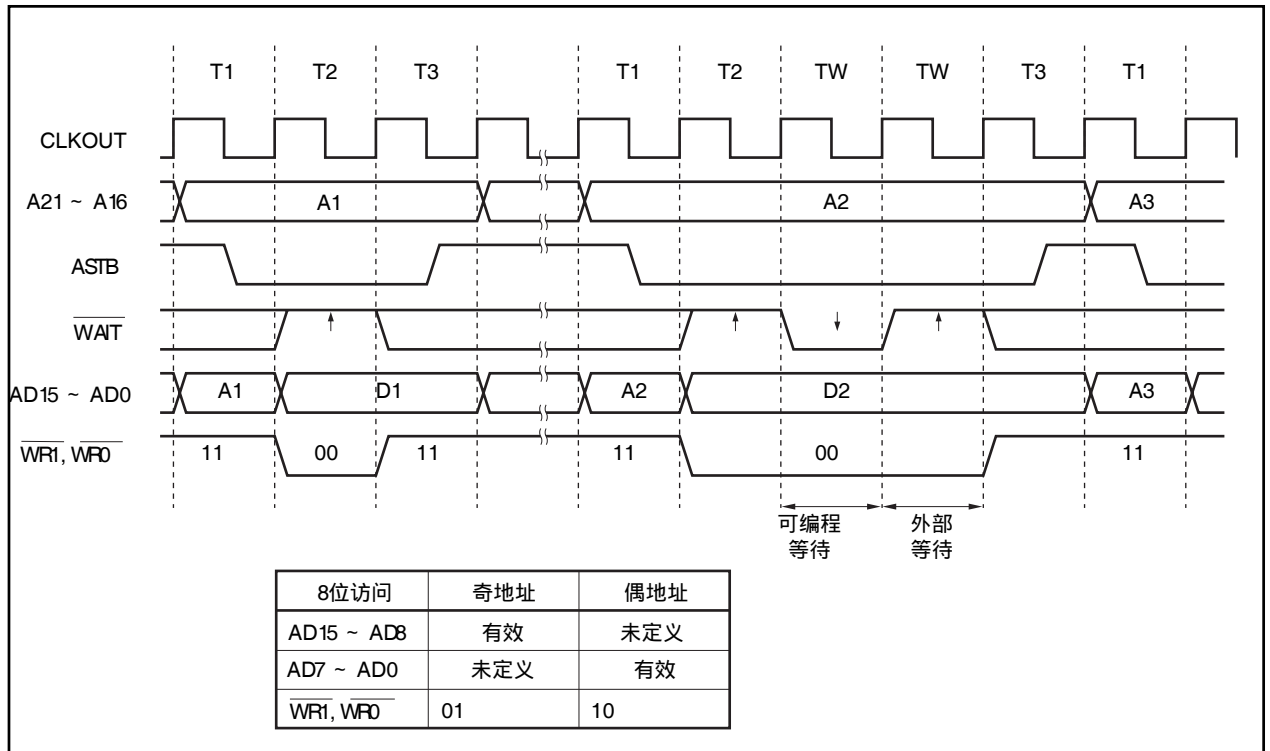


图 5-7. 多路总线写入时序 (总线宽度: 8 位)

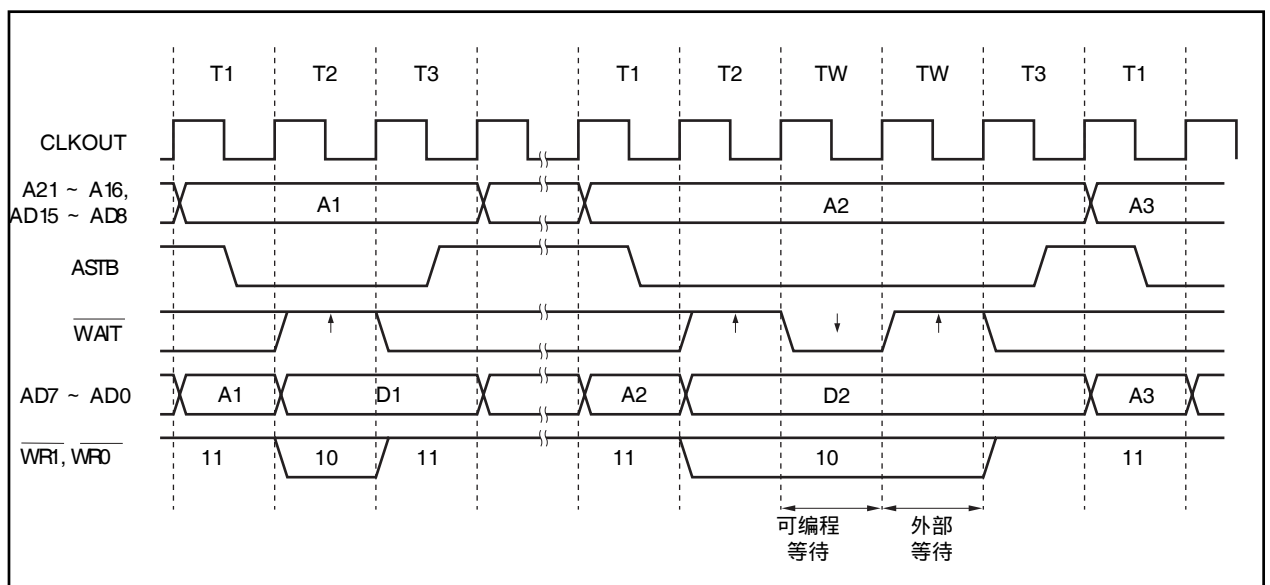


图 5-8. 多路总线保持时序 (总线宽度: 16 位, 16-位访问)

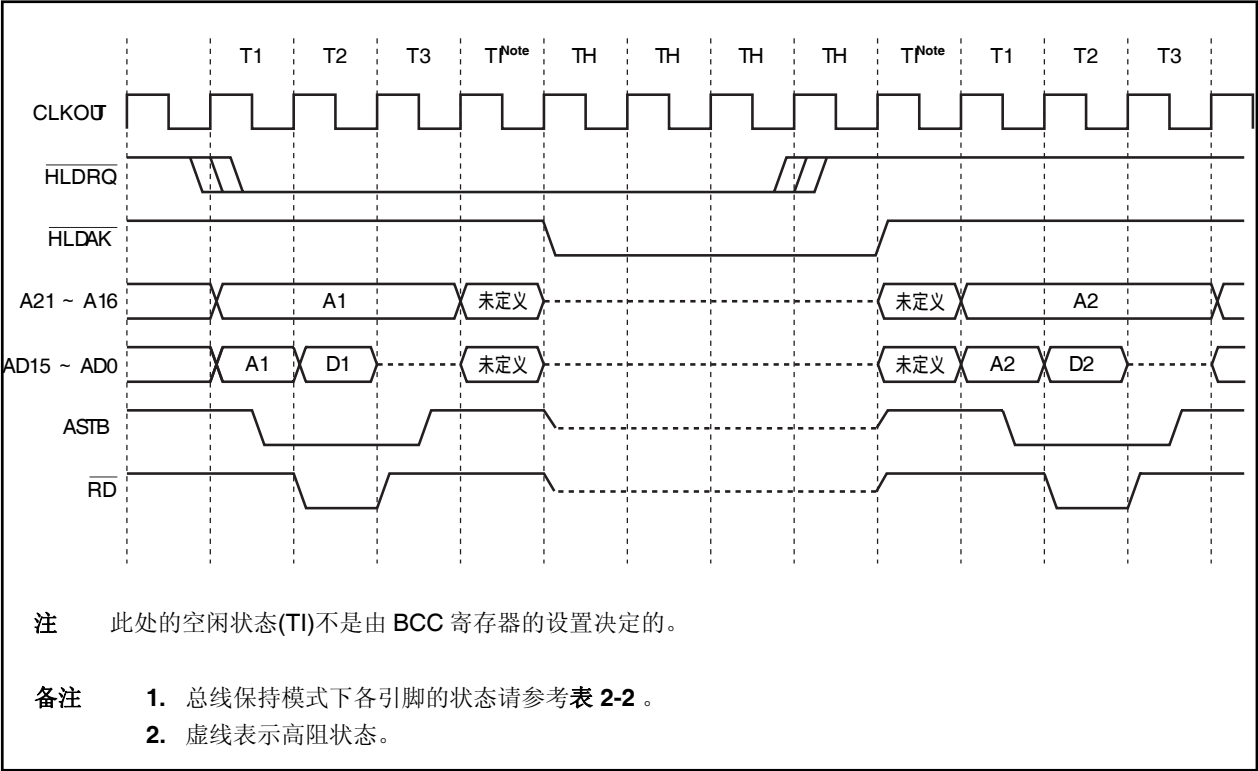


图 5-9. 分离总线读取时序 (总线宽度: 16 位, 16-位访问)

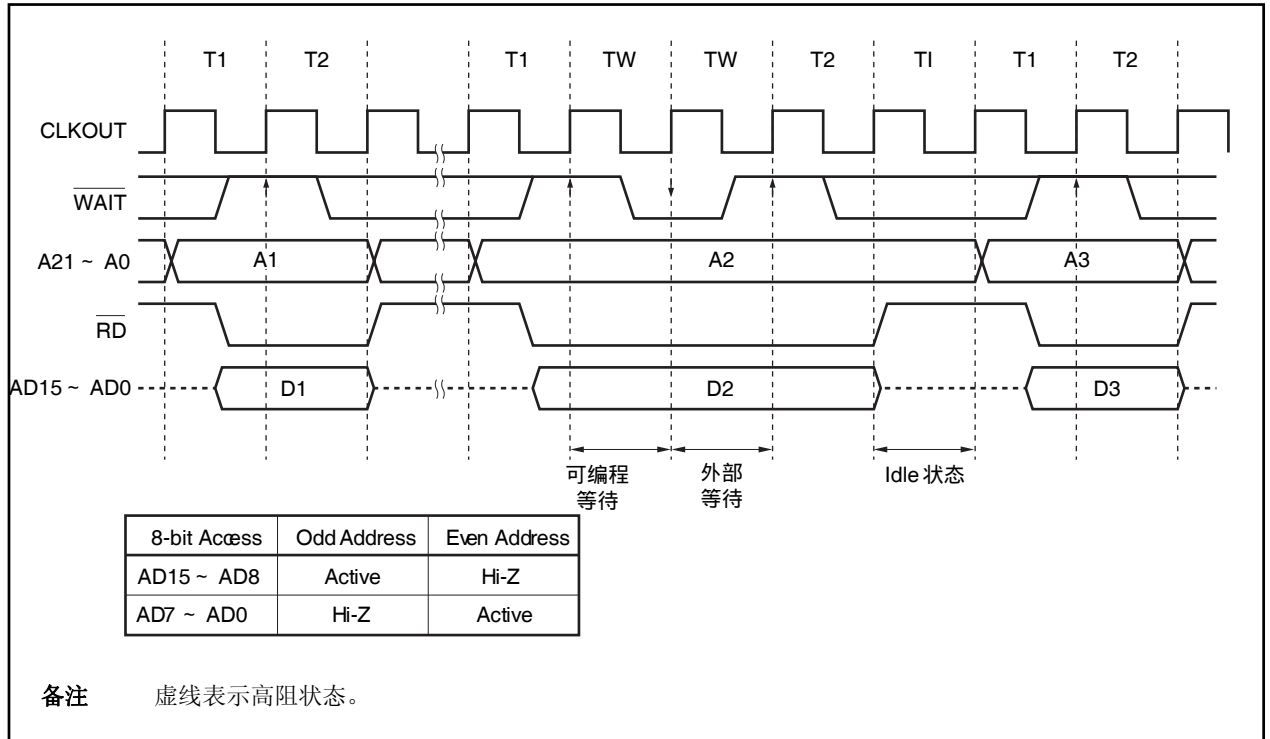


图 5-10. 分离总线读取时序 (总线宽度: 8 位)

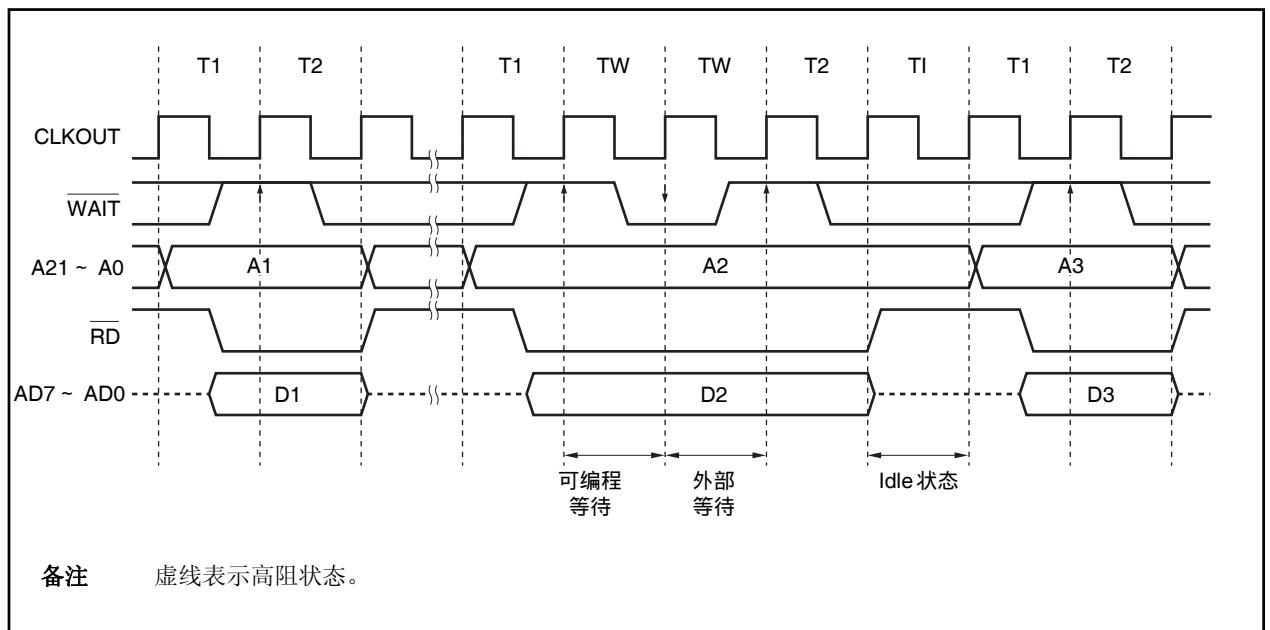
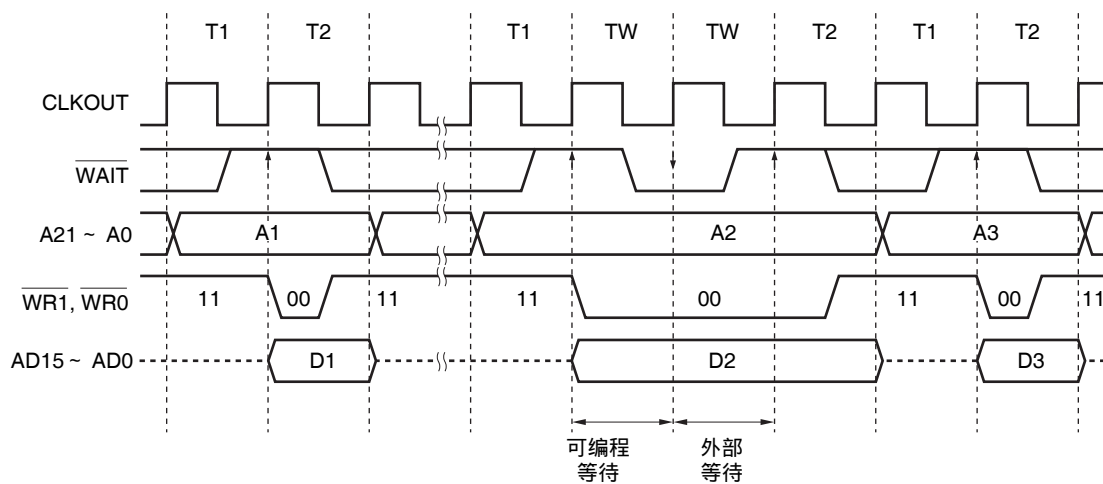


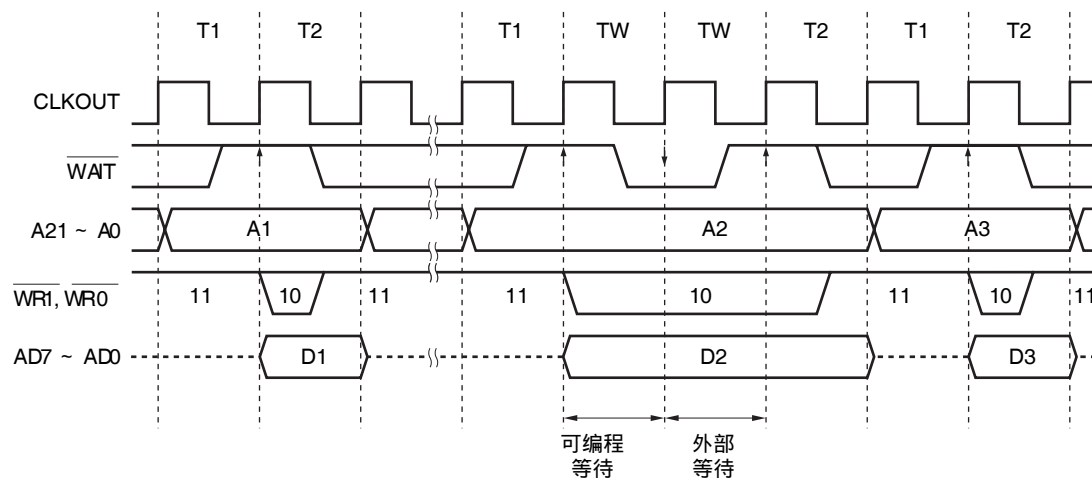
图 5-11. 分离总线写入时序 (总线宽度: 16 位, 16-位访问)



8位访问	奇地址	偶地址
AD15 ~ AD8	有效	未定义
AD7 ~ AD0	未定义	有效
WR1, WR0	01	10

备注 虚线表示高阻状态。

图 5-12. 分离总线写入时序 (总线宽度: 8 位)



备注 虚线表示高阻状态。

图 5-13. 分离总线保持时序 (总线宽度: 8 位, 写入)

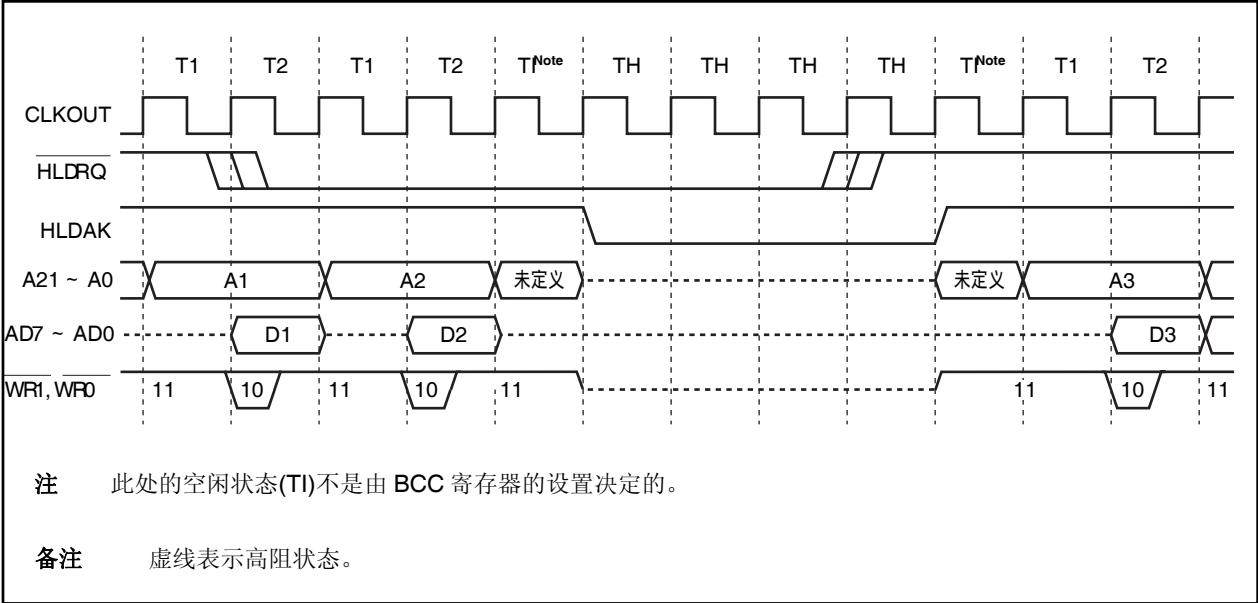
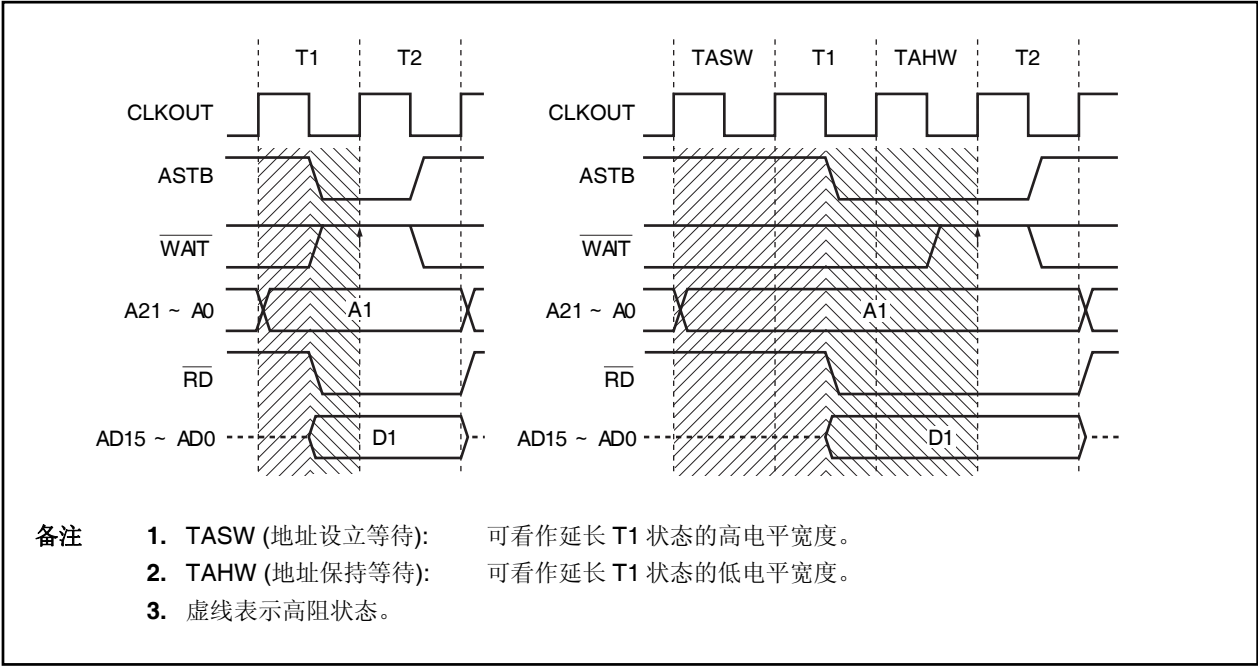


图 5-14. 地址等待时序 (分离总线读取, 总线宽度: 16 位, 16-位访问)



第六章 时钟产生功能

6.1 概述

V850ES/JG2 产品提供以下时钟产生功能。

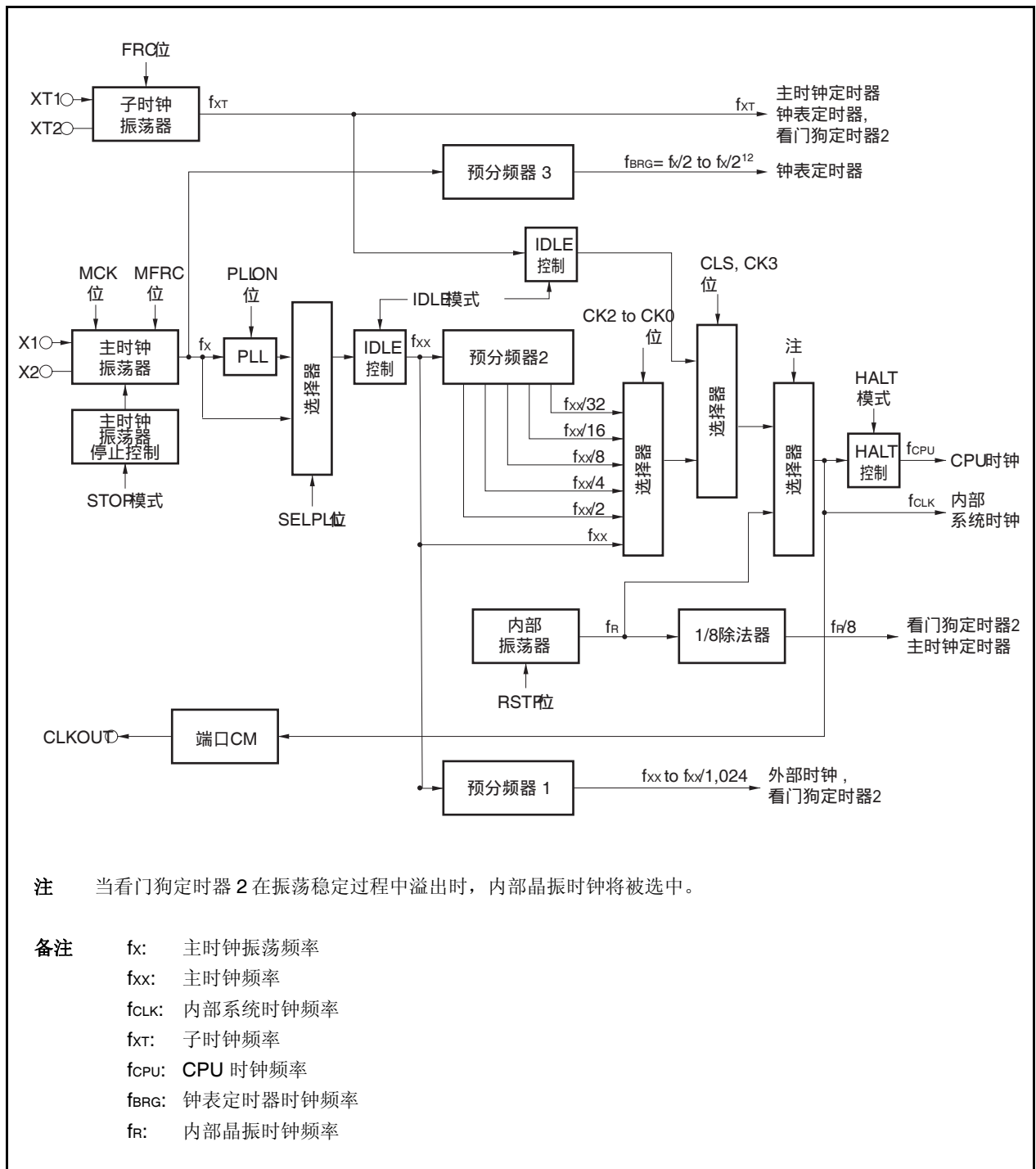
- 主时钟振荡器
 - 时钟直连模式(clock-through mode)
 $f_x = 2.5$ 到 10 MHz ($f_{xx} = 2.5$ 到 10 MHz)
 - PLL 模式
 $f_x = 2.5$ 到 5 MHz ($f_{xx} = 10$ 到 20 MHz)
- 子时钟振荡器
 - $f_{XT} = 32.768\text{ kHz}$
- PLL(锁相环 Phase Locked Loop)倍频 ($\times 4/\times 8$) 功能
 - 可选择时钟直连模式/PLL 模式
- 内部晶振
 - $f_R = 200\text{ kHz}$ (TYP.)
- 内部系统时钟产生
 - 7 级 (f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/8$, $f_{xx}/16$, $f_{xx}/32$, f_{XT})
- 周边时钟产生功能
- 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} : 子时钟频率
- f_R : 内部晶振时钟频率

6.2 结构

图 6-1. 时钟发生器



(1) 主时钟振荡器

主振荡器产生以下振荡频率 (f_x)。

- 时钟直连模式
 $f_x = 2.5$ 到 10 MHz
- PLL 模式
 $f_x = 2.5$ 到 5 MHz

(2) 子时钟振荡器

子振荡器产生频率为 32.768 kHz 的振荡信号(f_{XT})。

(3) 主时钟振荡器停止控制电路

该电路生成一个可停止主时钟振荡器振荡的控制信号。

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

(4) 内部晶振

产生一个频率为 200 kHz (TYP.) 的振荡信号 (f_R)。

(5) 预分频器 1

该预分频器生成的时钟信号(f_{xx} 到 $f_{xx}/1,024$)将提供给以下片上周边器件: TMP0 到 TMP5, TMQ0, TMM0, CSIB0 到 CSIB4, UARTA0 到 UARTA2, I²C00 到 I²C02^{‡1}, ADC, WDT2。

(6) 预分频器 2

该电路对主时钟 (f_{xx}) 进行分频。

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

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

(7) 预分频器 3

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

详见第 10 章 钟表定时器功能。

(8) PLL

该电路对主时钟振荡器产生的时钟信号(f_x)进行 4 或 8 倍频。

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

时钟的倍频系数(4 或 8)由 CKC.CKDIV0 位选择, PLL 工作的开始和停止由 PLLCTL.PLLON 位选择。

6.3 寄存器

(1) 处理器时钟控制寄存器 (PCC)

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

可使用 8 位或 1 位的操作对该寄存器进行读写。

复位输入将把该寄存器置为 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	是否使用副时钟片上反馈电阻							
0	使用							
1	不使用							
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CPU工作与主时钟振荡器时，即使将MCK位置1，主时钟振荡器也不会停止工作。主时钟振荡器只有在CPU时钟更改为副时钟时才可能被停止。• 将MCK位的设置从0更改为1之前，需要将使用主时钟频率工作的周边器件关闭。• 主时钟停止后，器件将工作于副时钟，若将CPU时钟或周边器件的工作时钟切换回主时钟，则需要将MCK位清零并通过软件确保振荡稳定时间。								
MFRC	是否使用主时钟片上反馈电阻							
0	使用							
1	不使用							
CLS ^注	CPU时钟状态 (f _{CPU})							
0	工作于主时钟							
1	工作于副时钟							
CK3	CK2	CK1	CK0	时钟选择 (f _{CLK} /f _{CPU})				
0	0	0	0	f _{xx}				
0	0	0	1	f _{xx} /2				
0	0	1	0	f _{xx} /4				
0	0	1	1	f _{xx} /8				
0	1	0	0	f _{xx} /16				
0	1	0	1	f _{xx} /32				
0	1	1	×	禁止设置				
1	×	×	×	f _{XT}				

注 CLS 位为只读位。

注意事项 1. 在通过 **CLKOUT** 向外输出时钟信号时，不要改变 **CPU** 时钟的频率(通过 **CK3** 到 **CK0** 位)。

2. 对 **CK3** 位操作时，请使用位操作指令。如果使用 8 位操作指令，不要改变 **CK2** 到 **CK0** 位的值。

备注 ×: 不关心

(a) 从主时钟操作切换至副时钟操作的举例

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

- ★ **注意事项** 1. 当停止主时钟工作时，也要同时停止 PLL。另外，工作于主时钟频率的片上周边器件也被停止。
2. 如果不满足下面的条件，那么请改变 CK2 到 CK0 位的设置以满足该条件，然后再切换到子时钟工作模式。
内部系统时钟 (f_{CLK}) > 子时钟 (f_{XT} : 32.768 kHz) \times 4

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

[示例]

```

_DMA_DISABLE:
crl      0, DCHCn[r0]      -- 禁止 DMA 操作。 n = 0 到 3
<1> _SET_SUB_RUN :
st.b     r0, PRCMD[r0]
set1     3, PCC[r0]        -- CK3 位  $\leftarrow$  1
<2> _CHECK_CLS :
tst1     4, PCC[r0]        -- 等待子时钟工作的开始
bz       _CHECK_CLS
<3> _STOP_MAIN_CLOCK :
st.b     r0, PRCMD[r0]
set1     6, PCC[r0]        -- MCK 位  $\leftarrow$  1, 主时钟停止
_DMA_ENABLE:
setl     0, DCHCn[r0]      -- 允许 DMA 操作。 n = 0 到 3

```

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

(b) 从副时钟操作切换至主时钟操作的举例

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

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

[示例]

```

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

```

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

(2) 内置振荡模式寄存器 (RCM)

RCM 寄存器是用于设置内部晶振工作模式的 8 位寄存器。

可以对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器置为 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	内部振荡器的振荡控制
0	内部振荡器振荡
1	内部振荡器停止

注意事项 1. CPU 工作于内部晶振时钟时(CCLS.CCLSF 位 = 1)，不可以停止内部晶振的工作。不要将 RSTOP 位置 1。

2. 即使 RSTOP 位为 1，如果 CCLS.CCLSF 位被置 1 内部晶振也会开始振荡。(振荡稳定时间中产生 WDT 溢出时)此时，RSTOP 位仍为 1。

(3) CPU 操作时钟状态寄存器 (CCLS)

CCLS 寄存器是用于指示 CPU 操作时钟状态的寄存器。

只能对该寄存器进行 8 位或 1 位的读取操作。

复位数据将把该寄存器置为 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	CCLSF

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

6.4 操作

6.4.1 各时钟的操作

下表显示了各时钟的操作状态。

表 6-1. 各时钟的操作状态

寄存器设置 以及操作状态		PCC 寄存器								
		CLS 位= 0, MCK 位= 0					CLS 位 = 1, MCK 位 = 0		CLS 位 = 1, MCK 位 = 1	
		复位 过程中	振荡稳定 时间等待 过程中	HALT 模式	IDLE1, IDLE2 模式	STOP 模式	子时钟 模式	Sub-IDLE 模式	子时钟 模式	Sub-IDLE 模式
目标时钟										
主时钟振荡器 (fx)	×	○	○	○	×	○	○	×	×	
子时钟振荡器 (fxτ)	○	○	○	○	○	○	○	○	○	
CPU 时钟 (fcpu)	×	×	×	×	×	○	×	○	×	
内部系统时钟 (fclk)	×	×	○	×	×	○	×	○	×	
主时钟 (PLL 模式, fxx)	×	○ ^注	○	×	×	○	○	×	×	
周边时钟 (fxx 到 fxx/1,024)	×	×	○	×	×	○	×	×	×	
WT 时钟 (主)	×	○	○	○	×	○	○	×	×	
WT 时钟 (子)	○	○	○	○	○	○	○	○	○	
WDT2 时钟 (内置振荡时钟)	×	○	○	○	○	○	○	○	○	
WDT2 时钟 (主)	×	×	○	×	×	○	×	×	×	
WDT2 时钟 (子)	○	○	○	○	○	○	○	○	○	

注 入锁时间

备注 ○: 可操作
×: 停止

6.4.2 时钟输出功能

时钟输出功能用于将内部系统时钟(fCLK)从 CLKOUT 引脚输出。

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

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

CLKOUT 引脚的状态与表 6-1 中所示的内部系统时钟的状态是一致的，该引脚可以在内部系统时钟为可操作状态时，输出时钟信号。当内部系统时钟处于停止状态时，CLKOUT 引脚输出为低电平。复位释放后，该引脚的默认模式为端口模式(PCM1 引脚: 输入模式)。因此，直到该引脚被设置为输出模式之前，它的引脚状态保持为高阻抗状态 (Hi-Z)

6.5 PLL 功能

6.5.1 概述

在 V850ES/JG2 系列产品中，CPU 和片上周边器件的时钟信号可以选择频率为晶振频率的 4 或 8 倍的 PLL 模式和时钟直连模式。

使用 PLL 功能时: 输入时钟 = 2.5 到 5 MHz (输出: 10 到 20 MHz)

时钟直连模式: 输入时钟 = 2.5 到 10 MHz (输出: 2.5 到 10 MHz)

6.5.2 寄存器

(1) PLL 控制寄存器 (PLLCTL)

PLLCTL 寄存器是用来控制 PLL 功能的 8 位寄存器。

可以对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器置为 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操作控制寄存器
0	PLL停止
1	PLL 工作 (PLL开始工作后，需要一个入锁时间来保证振荡频率的稳定)

SELPLL	CPU操作时钟选择寄存器
0	时钟直通模式
1	PLL模式

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

(2) 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 寄存器用于控制 PLL 模式下的内部系统时钟。

可以对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器置为 0AH。

复位后: 0AH R/W 地址: FFFFF822H

	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	$f_{xx} = 4 \times f_x$ ($f_x = 2.5$ 到 5.0 MHz)
1	$f_{xx} = 8 \times f_x$ ($f_x = 2.5$ MHz)

注意事项 1. $f_x = 5.0 \sim 10.0$ MHz 时, 不可使用 PLL 模式。

2. 在使用 CKC 寄存器在 4 倍和 8 倍间改变倍频系数之前, 要先设置时钟为直连模式并停止 PLL 工作。

3. 请务必将第 3 位和第 1 位置于 1, 将第 7 到 4 和第 2 到 0 位清零。

备注 CPU 时钟和片上周边时钟都会根据 CKC 寄存器的设置倍频, 而只有 CPU 时钟会根据 PCC 寄存器的设置被分频。

(3) 锁定寄存器 (LOCKR)

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

包含 LOCK 位的 LOCKR 寄存器用于反映 PLL 频率稳定状态。

只能对该寄存器进行 8 位或 1 位的读取操作。

复位输入将把该寄存器置为 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	检查PLL锁定状态
0	锁定状态
1	未锁定状态

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

[置位条件]

- 系统复位[※]。
- 在 IDLE2 或 STOP 模式下。
- 设置了 PLL 停止(将 PLLCTL.PLLON 清零)时。
- 主时钟停止，系统工作于子时钟(PCC.CK3 位和 PCC.MCK 位置 1)时。

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

[清零条件]

- 复位释放后振荡稳定时间(OSTS 寄存器的默认时间(见 21.2 (3) 振荡稳定时间选择寄存器 (OSTS))) 溢出时。
- 在 PLL 工作状态下设置了 STOP 模式，STOP 模式解除后，振荡稳定定时器溢出时(时间由 OSTS 寄存器设置)。
- 当 PLLCTL.PLLON 位的设置从 0 改变为 1，PLL 入锁时间定时器产生溢出时(时间由 PLLS 寄存器设置)。
- 在 PLL 工作状态下设置了 IDLE2 模式，IDLE2 模式解除时，所插入的设立时间(setup time)(由 OSTS 寄存器设置的时间)结束后。

(4) PLL 入锁时间指定寄存器 (PLLS)

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

可以对该寄存器进行 8 位的读写操作。

复位输入将把该寄存器设置为 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	选择PLL入锁时间
0	0	$2^{10}/f_x$
0	1	$2^{11}/f_x$
1	0	$2^{12}/f_x$
1	1	$2^{13}/f_x$ (默认值)

注意事项 1. 将入锁时间设置为 800 μ s 或更长。

2. 入锁过程中不要改变 PLLS 寄存器的设置。

6.5.3 使用方法**(1) 使用 PLL 时**

- 复位信号被释放之后, PLL 处于工作状态(PLLCTL.PLLON bit = 1), 但由于默认模式为时钟直连模式(PLLCTL.SELPLL 位 = 0), 为使 PLL 有效, 请选择 PLL 模式(SELPLL 位 = 1)。
 - 允许 PLL 工作: 首先要将 PLLON 位置 1, 然后在 LOCKR.LOCK 位 = 0 之后将 SELPLL 位置 1。
停止 PLL 工作: 首先选择时钟直连模式(SELPLL 位 = 0), 等待 8 个或更多时钟周期, 然后停止 PLL 工作(PLLON 位 = 0)。
 - 系统模式切换至 IDLE2 或 STOP 模式时, PLL 将无条件停止工作, 系统模式从 IDLE2 或 STOP 切换回之前的模式时, PLL 将恢复工作。恢复时间如下所示。
- (a)当从时钟直连模式向 IDLE2 或 STOP 模式切换时。
- STOP 模式: 设置 OSTS 寄存器使得 振荡稳定时间为 (1ms(min.))或更长。
 - IDLE2 模式: 设置 OSTS 寄存器使得 设立时间(Setup time)为 (350 μ s (min.))或更长。

(b)PLL 工作模式中, 向 IDLE2 或 STOP 模式切换时。

STOP 模式: 设置 OSTS 寄存器使得振荡稳定时间为(1ms(min.))或更长

IDLE2 模式: 设置 OSTS 寄存器使得设定时间为 (800 μ s (min.))或更长

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

(2) 不使用 PLL 时

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

第七章 16 位定时器/事件计数器 P (TMP)

定时器 P (TMP) 是一个 16 位的定时器/事件计数器。

V850ES/JG2 具有 6 个定时器/事件计数器通道，TMP0 到 TMP5。

7.1 概述

以下是 TMPn 的概况。

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

备注 n = 0 到 5

7.2 功能

TMPn 具有以下功能。

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

备注 n = 0 到 5

7.3 配置

TMPn 包含以下硬件。

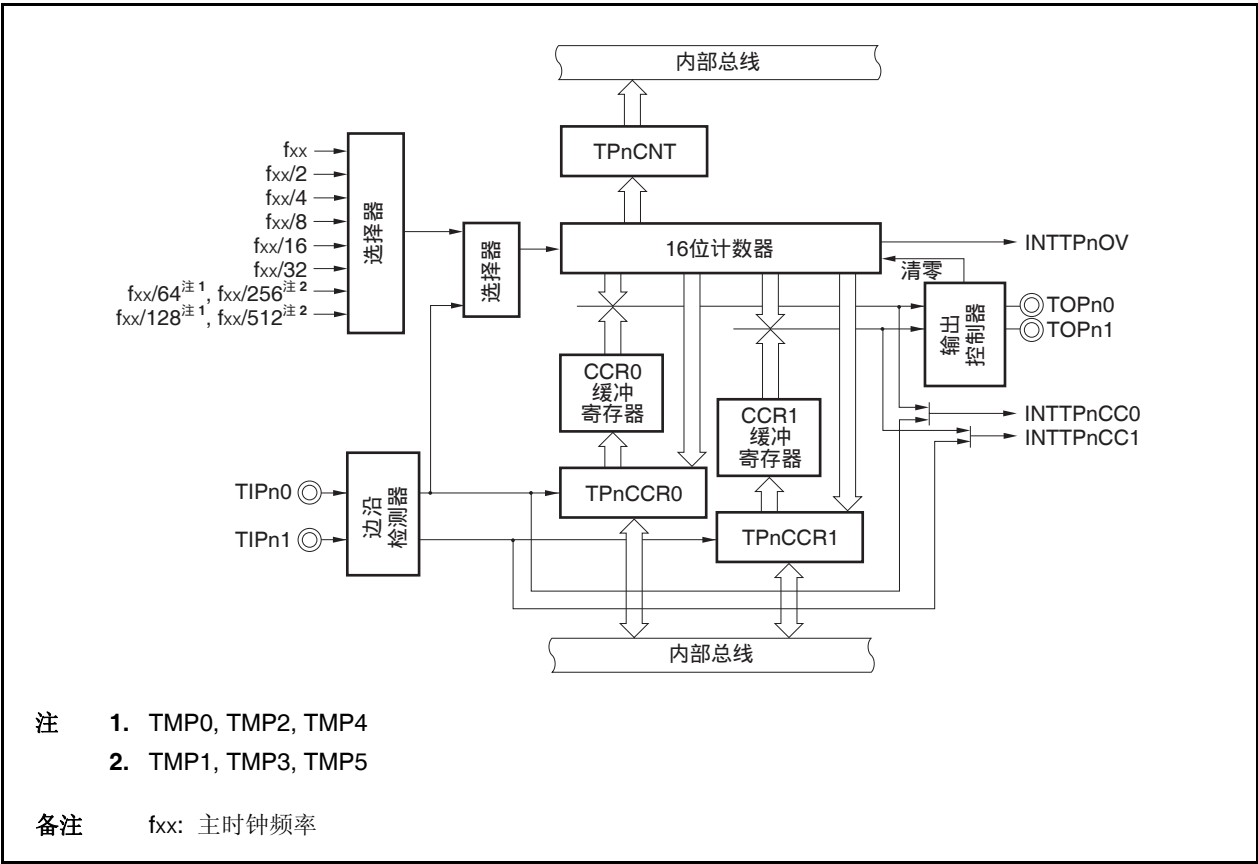
表 7-1. TMPn 的配置

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

- 注
1. TIPn0 引脚的复用功能包括捕捉触发输入信号，外部事件计数输入信号和外部触发输入信号。
 2. 使用 TIPn0, TIPn1, TOPn0 和 TOPn1 引脚功能时，请参照表 4-15 **Settings When Port Pins Are Used for Alternate Functions**。

备注 n = 0 到 5

图 7-1. TMPn 的框图



(1) 16 位计数器

该 16 位计数器可以对内部时钟或外部事件进行计数。

可以使用 TPnCNT 寄存器读取该计数器的计数值。

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

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

(2) CCR0 缓冲寄存器

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

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

CCR0 缓冲寄存器不能被读/写。

复位后 CCR0 缓冲寄存器和 TPnCCR0 寄存器都将被清零。

(3) CCR1 缓冲寄存器

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

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

CCR1 缓冲寄存器不能被读/写。

复位后 CCR0 缓冲寄存器和 TPnCCR0 寄存器都将被清零。

(4) 边沿检测器

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

(5) 输出控制器

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

(6) 选择器

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

7.4 寄存器

用于控制 TMPn 的寄存器如下所述。

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

- 备注
1. 使用 TIPn0, TIPn1, TOPn0 和 TOPn1 引脚功能时, 请参照表 4-15 **Settings When Port Pins Are Used for Alternate Functions**。
 2. n = 0 到 5

(1) TMPn 控制寄存器 0 (TPnCTL0)

TPnCTL0 寄存器是用于控制 TMPn 操作的 8 位寄存器。

可以对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

可使用软件向 TPnCTL0 寄存器写入相同的数值。

复位后: 00H R/W 地址: TP0CTL0 FFFFF590H, TP1CTL0 FFFFF5A0H,
TP2CTL0 FFFFF5B0H, TP3CTL0 FFFFF5C0H,
TP4CTL0 FFFFF5D0H, TP5CTL0 FFFFF5E0H

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

TPnCE	TMPn 操作控制
0	禁止TMPn操作(TMPn异步复位 ^注)。
1	允许TMPn操作。 TMPn开始工作。

TPnCKS2	TPnCKS1	TPnCKS0	内部计数时钟选择	
			n = 0, 2, 4	n = 1, 3, 5
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	f _{xx} /256
1	1	1	f _{xx} /128	f _{xx} /512

注 TPnOPT0.TPnOVF 位, 16 位计数器, 定时器输出 (TOPn0, TOPn1 引脚)

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

备注 f_{xx}: 主时钟频率

(2) TMPn 控制寄存器 1 (TPnCTL1)

TPnCTL1 寄存器是用于控制 TMPn 操作的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后: 00H R/W 地址: TP0CTL1 FFFFF591H, TP1CTL1 FFFFF5A1H,
TP2CTL1 FFFFF5B1H, TP3CTL1 FFFFF5C1H,
TP4CTL1 FFFFF5D1H, TP5CTL1 FFFFF5E1H

	7	<6>	<5>	4	3	2	1	0
TPnCTL1 (n = 0 到 5)	0	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0

TPnEST	软件触发控制
0	—
1	产生一个外部触发输入的有效信号 • 单脉冲输出模式下: 向TPnEST位写入1将触发一个单脉冲的输出。 • 外部触发脉冲输出模式下: 向TPnEST位写入1将触发一个PWM波形的输出。

TPnEEE	计数时钟选择
0	禁止对外部事件计数输入的计数。 (按照TPnCTL0.TPnCK0到TPnCK2位所选择的计数时钟进行计数。)
1	允许对外部事件计数输入的计数。 (按照外部事件计数输入信号的有效沿进行计数。)
TPnEEE位用于对内部计数时钟计数或外部事件计数输入的有效沿计数进行选择。	

TPnMD2	TPnMD1	TPnMD0	定时器模式选择
0	0	0	间隔定时器模式
0	0	1	外部事件计数模式
0	1	0	外部触发脉冲输出模式
0	1	1	单脉冲输出模式
1	0	0	PWM输出模式
1	0	1	自由运行定时器模式
1	1	0	脉宽测量模式
1	1	1	禁止设置

- 注意事项**
1. 对 **TPnEST** 位的设置只有在外部触发脉冲输出模式或单脉冲输出模式才有效。在其它模式下, 对该位的置 1 操作将被忽略。
 2. 在外部事件计数模式下, 无论 **TPnEEE** 位如何设置, 外部事件计数输入都是被选中的。
 3. 在 **TPnCTL0.TPnCE** 位 = 0 时才可对 **TPnEEE** 和 **TPnMD2** 到 **TPnMD0** 位进行设置 (**TPnCE** 位 = 1 时只可以对这些位写入相同的值)。如果在 **TPnCE** 位 = 1 时对这些位进行了改写, 那么操作结果将不能被保证。若由于误操作引起了改写, 则要将 **TPnCE** 位清零然后再次设置这些寄存器位。
 4. 请务必将第 3, 4 和 7 位清零。

(3) TMPn I/O 控制寄存器 0 (TPnIOC0)

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

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后: 00H R/W 地址: TP0IOC0 FFFFF592H, TP1IOC0 FFFFF5A2H,
TP2IOC0 FFFFF5B2H, TP3IOC0 FFFFF5C2H,
TP4IOC0 FFFFF5D2H, TP5IOC0 FFFFF5E2H

	7	6	5	4	3	<2>	1	<0>
TPnIOC0	0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0

(n = 0 to 5)

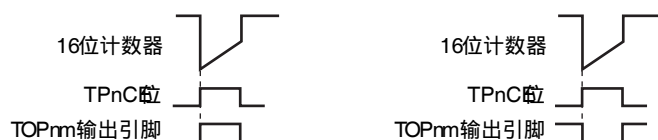
TPnOL1	TOPn1引脚的输出电平设置
0	禁止TOPn1引脚反向输出
1	允许TOPn1引脚反向输出

TPnOE1	TOPn1引脚输出设置
0	禁止定时器输出 • 当TPnOL1位 = 0: TOPn1引脚输出低电平 • 当TPnOL1位 = 1: TOPn1引脚输出高电平
1	允许定时器输出 (TOPn1引脚输出一列方波)。

TPnOL0	TOPn0引脚输出电平设置
0	禁止TOPn0引脚反向输出
1	允许TOPn0引脚反向输出

TPnOE0	TOPn0引脚输出设置
0	禁止定时器输出 <ul style="list-style-type: none"> 当TPnOL0位 = 0: TOPn0引脚输出低电平 当TPnOL0位 = 1: TOPn0引脚输出高电平
1	允许定时器输出 (TOPn0引脚输出一列方波)。

注 TPnOLm 位指定的定时器输出引脚(TOPnm)的输出电平如下所示(m = 0,1)。



注意事项

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

(4) TMPn I/O 控制寄存器 1 (TPnIOC1)

TPnIOC1 寄存器是用于控制捕捉触发输入信号(TIPn0, TIPn1 引脚)有效沿的 8 位寄存器。
可对该寄存器进行 8 位或 1 位的读写操作。
复位输入将把该寄存器设置为 00H。

复位后: 00H R/W 地址: TP0IOC1 FFFFF593H, TP1IOC1 FFFFF5A3H,
 TP2IOC1 FFFFF5B3H, TP3IOC1 FFFFF5C3H,
 TP4IOC1 FFFFF5D3H, TP5IOC1 FFFFF5E3H

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

TPnIS3	TPnIS2	捕捉触发输入信号 (TIPn1引脚)有效边沿设定
0	0	无边沿检测 (不进行捕捉操作)
0	1	上升沿检测
1	0	下降沿检测
1	1	双沿检测

TPnIS1	TPnIS0	捕捉触发输入信号 (TIPn0引脚)有效边沿设定
0	0	无边沿检测 (不进行捕捉操作)
0	1	上升沿检测
1	0	下降沿检测
1	1	双沿检测

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

(5) TMPn I/O 控制寄存器 2 (TPnIOC2)

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

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后: 00H R/W 地址: TP0IOC2 FFFFF594H, TP1IOC2 FFFFF5A4H,
TP2IOC2 FFFFF5B4H, TP3IOC2 FFFFF5C4H,
TP4IOC2 FFFFF5D4H, TP5IOC2 FFFFF5E4H

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

TPnEES1	TPnEES0	外部事件计数输入信号 (TIPn0引脚)有效边沿设定
0	0	无边沿检测 (不进行外部事件计数)
0	1	上升沿检测
1	0	下降沿检测
1	1	双沿检测

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

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

(7) TMPn 捕捉/比较寄存器 0 (TPnCCR0)

根据选择模式的不同，TPnCCR0 可用作捕捉寄存器或比较寄存器。

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

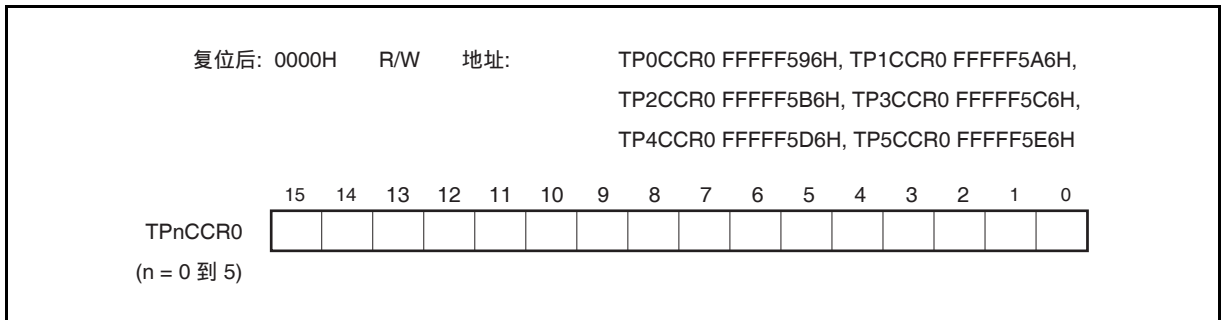
TPnCCR0 寄存器可以在操作过程中进行读写。

可对该寄存器进行 16 位的读写操作。

复位输入将把该寄存器设置为 0000H。

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

- 当 CPU 工作于子时钟且主时钟振荡停止时
- 当 CPU 工作于内置振荡时钟时



(a) 作为比较寄存器时的功能

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

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

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

(b) 作为捕捉寄存器时的功能

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

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

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 7-2. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

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

(8) TMPn 捕捉/比较寄存器 1 (TPnCCR1)

根据选择模式的不同，TPnCCR1 可用作捕捉寄存器或比较寄存器。

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

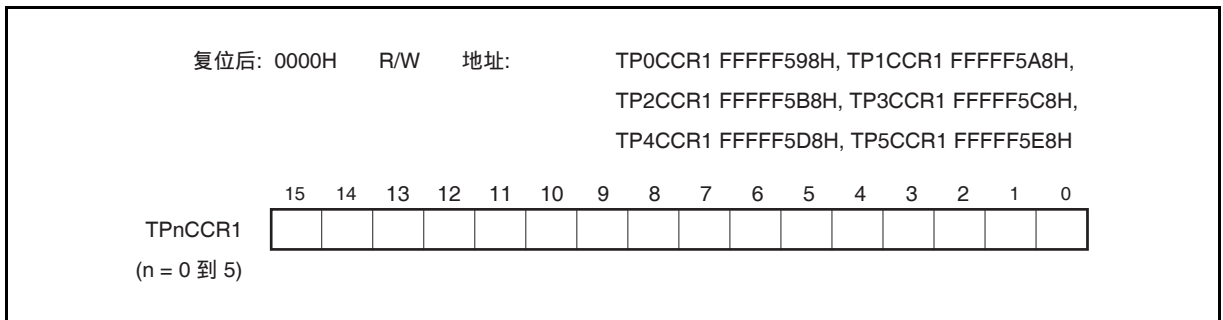
TPnCCR1 寄存器可以在操作过程中进行读写。

可对该寄存器进行 16 位的读写操作。

复位输入将把该寄存器设置为 0000H。

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

- 当 CPU 工作于子时钟且主时钟振荡停止时
- 当 CPU 工作于内置振荡时钟时



(a) 作为比较寄存器时的功能

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

向 TPnCCR1 寄存器中设置的数值将被传送到 CCR1 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 缓冲寄存器中的数值相匹配时，会产生一个比较匹配中断请求信号(INTTPnCC1)。如果允许了 TOPn1 引脚输出功能，那么 TOPn0 引脚输出的电平将被反转。

(b) 作为捕捉寄存器时的功能

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

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

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 7-3. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

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

(9) TMPn 计数器读取缓冲寄存器 (TPnCNT)

可以通过读取缓冲寄存器 TPnCNT 来读取 16 位计数器的计数值。

如果该寄存器在 TPnCTL0.TPnCE 位 = 1 时被读取，那么读取的值就是 16 位计数器的计数值。

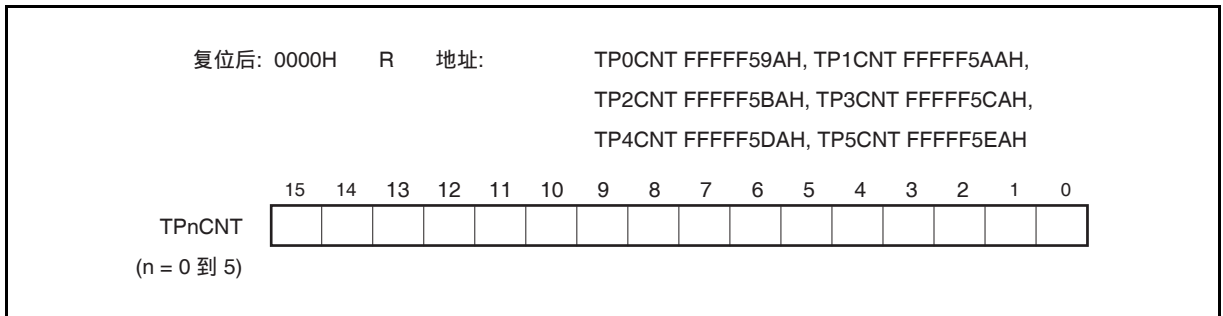
只可以对该寄存器进行 16 位的读取操作。

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

复位后，TPnCE 位被清零的同时 TPnCNT 寄存器的值也被清零。

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

- 当 CPU 工作于子时钟且主时钟振荡停止时
- 当 CPU 工作于内置振荡时钟时



7.5 操作

TMPn 可进行以下操作。

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

- 注

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

备注 n = 0 到 5

7.5.1 间隔定时器模式 (TPnMD2 到 TPnMD0 位 = 000)

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

通常情况下，间隔定时器模式中不使用 TPnCCR1 寄存器。

图 7-2. 间隔定时器的配置图

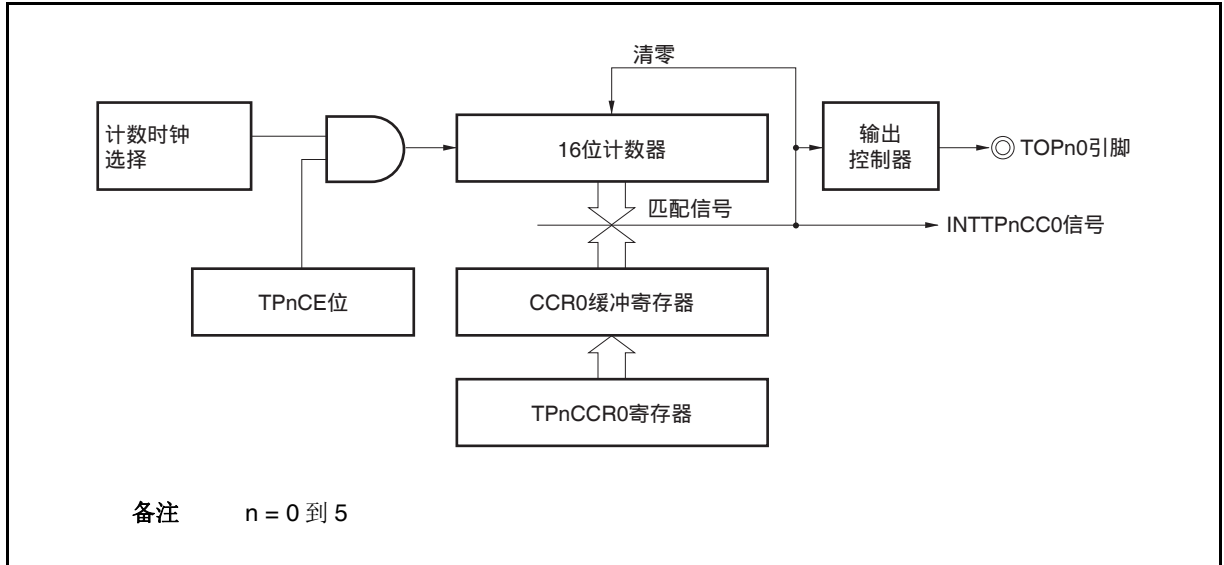
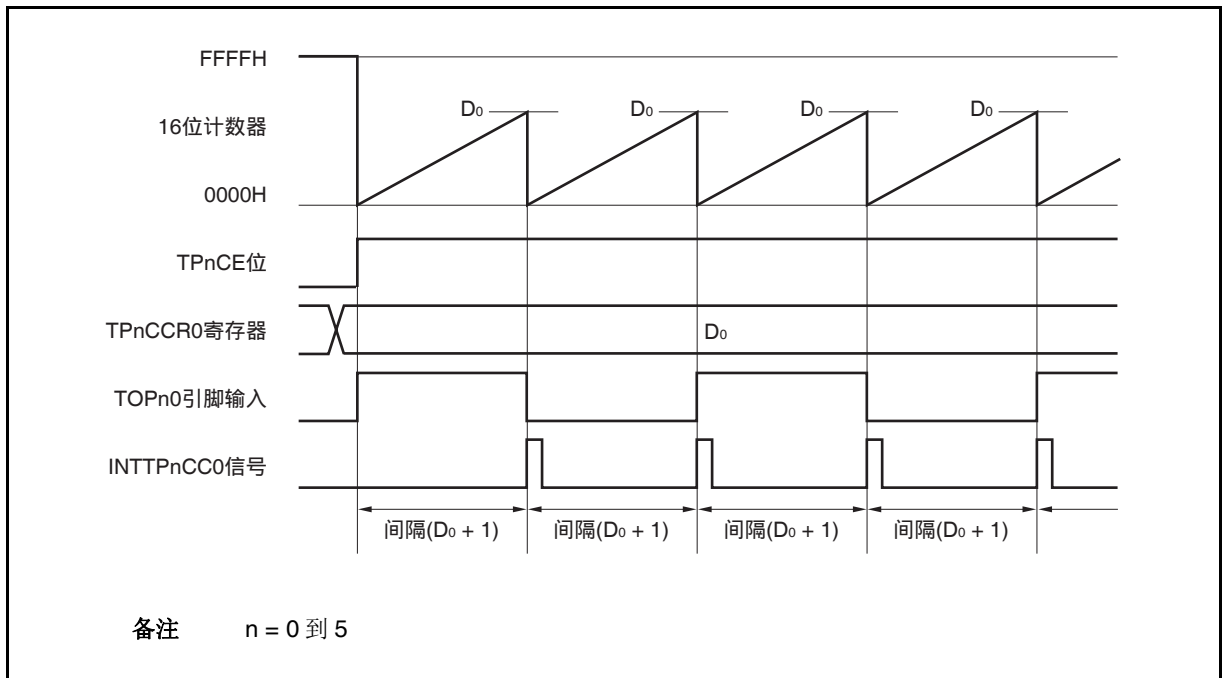


图 7-3. 间隔定时器模式操作的基本时序



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

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

间隔时间可由以下算式求得。

间隔时间 = (TPnCCR0 寄存器的设定值 + 1) × 计数时钟周期

备注 n = 0 到 5

图 7-4. 间隔定时器模式操作的寄存器设置(1/2)

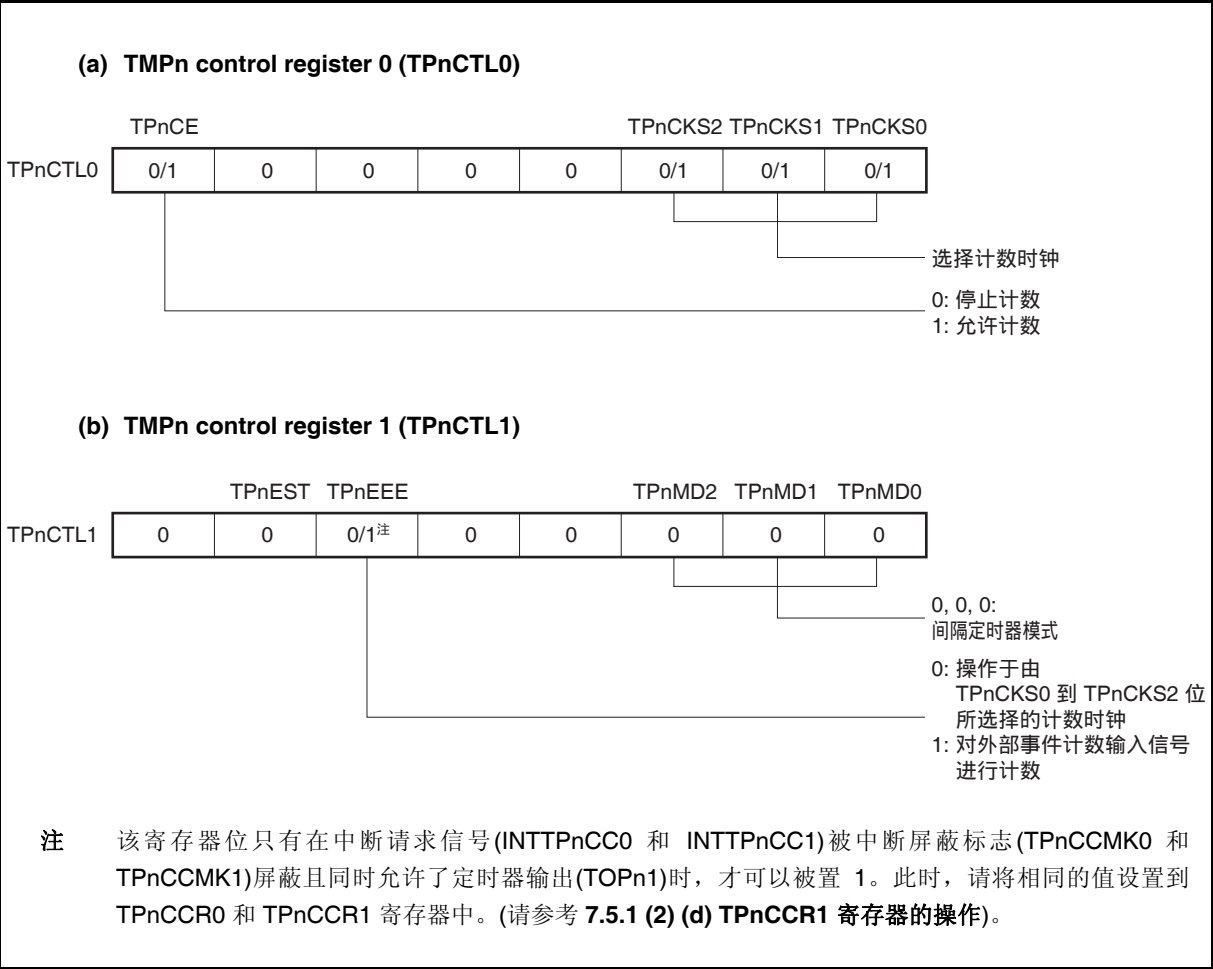
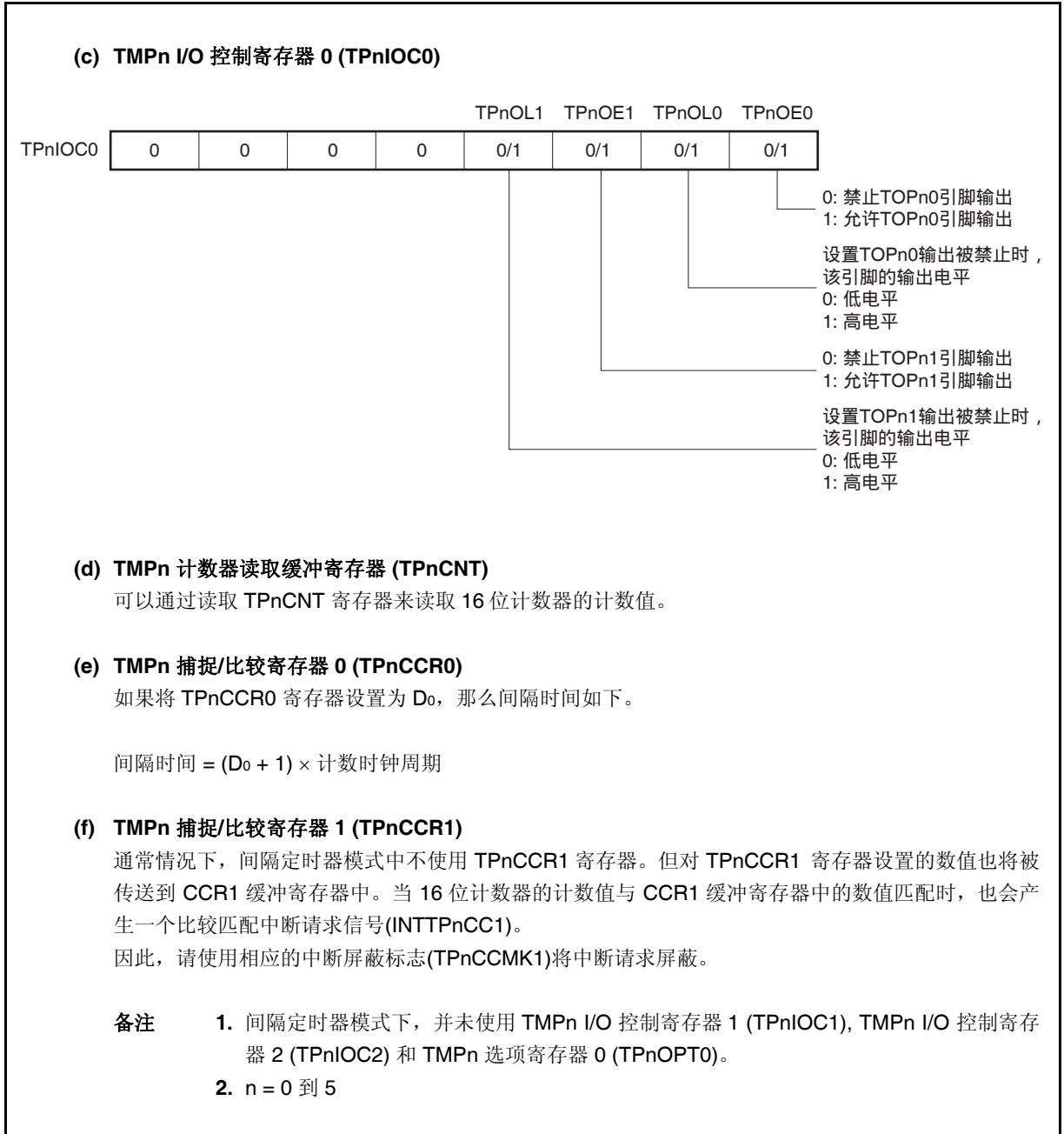
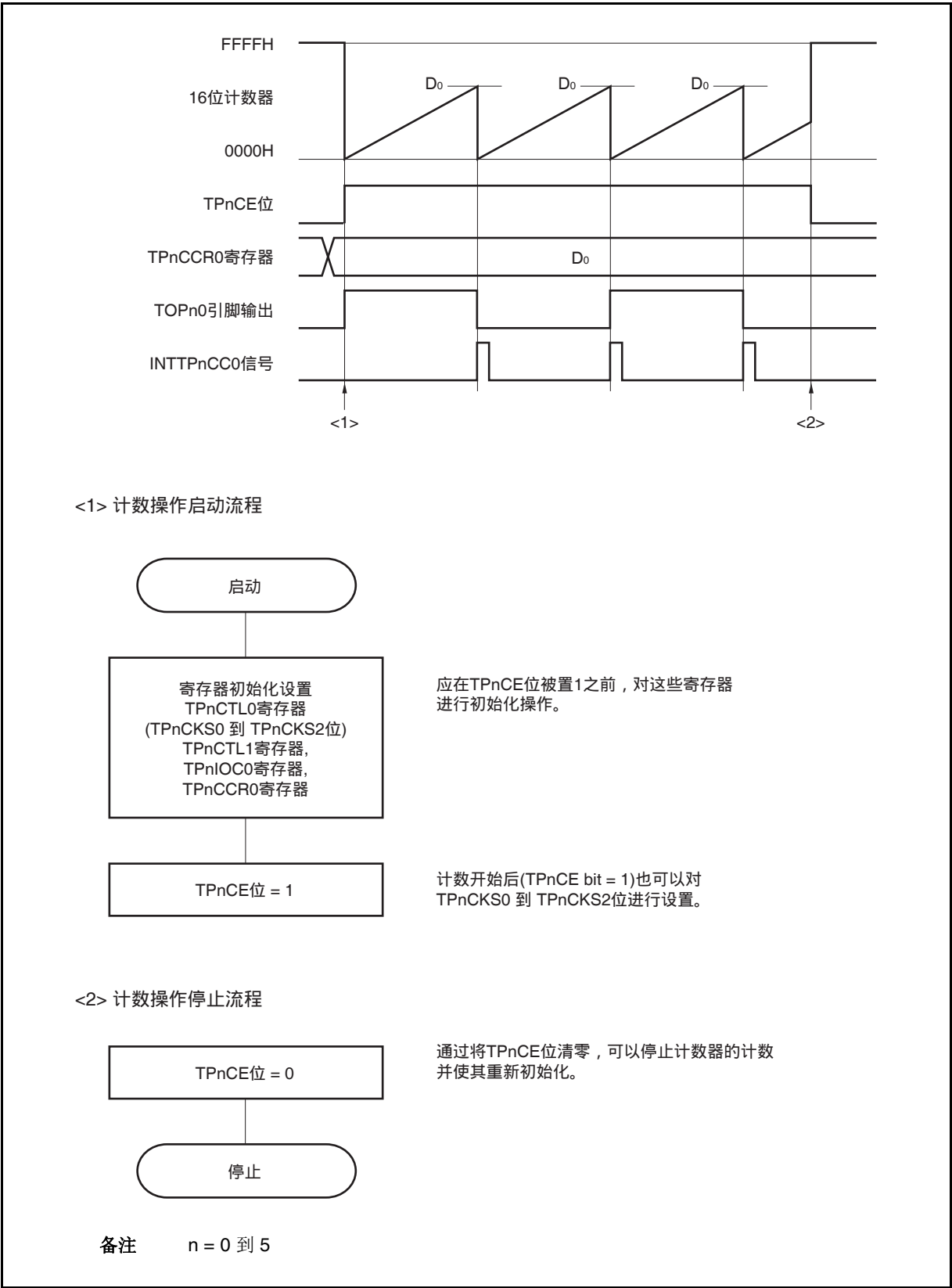


图 7-4. 间隔定时器模式操作的寄存器设置 (2/2)



(1) 间隔定时器模式操作流程

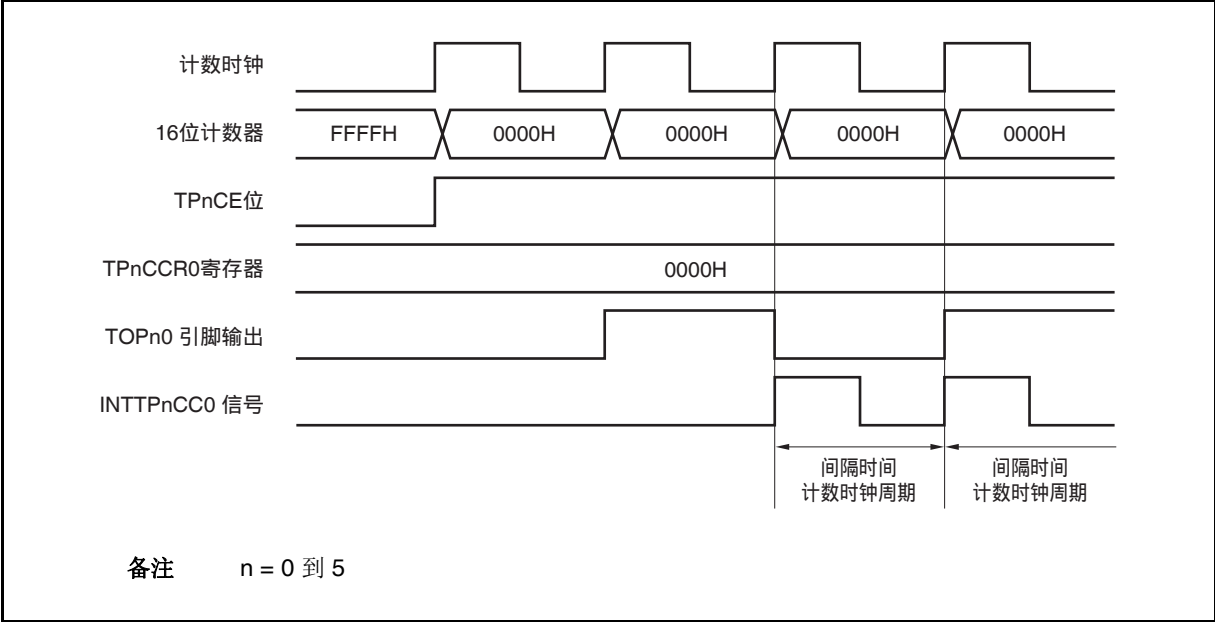
图 7-5. 间隔定时器模式下的软件处理流程



(2) 间隔定时器模式操作时序

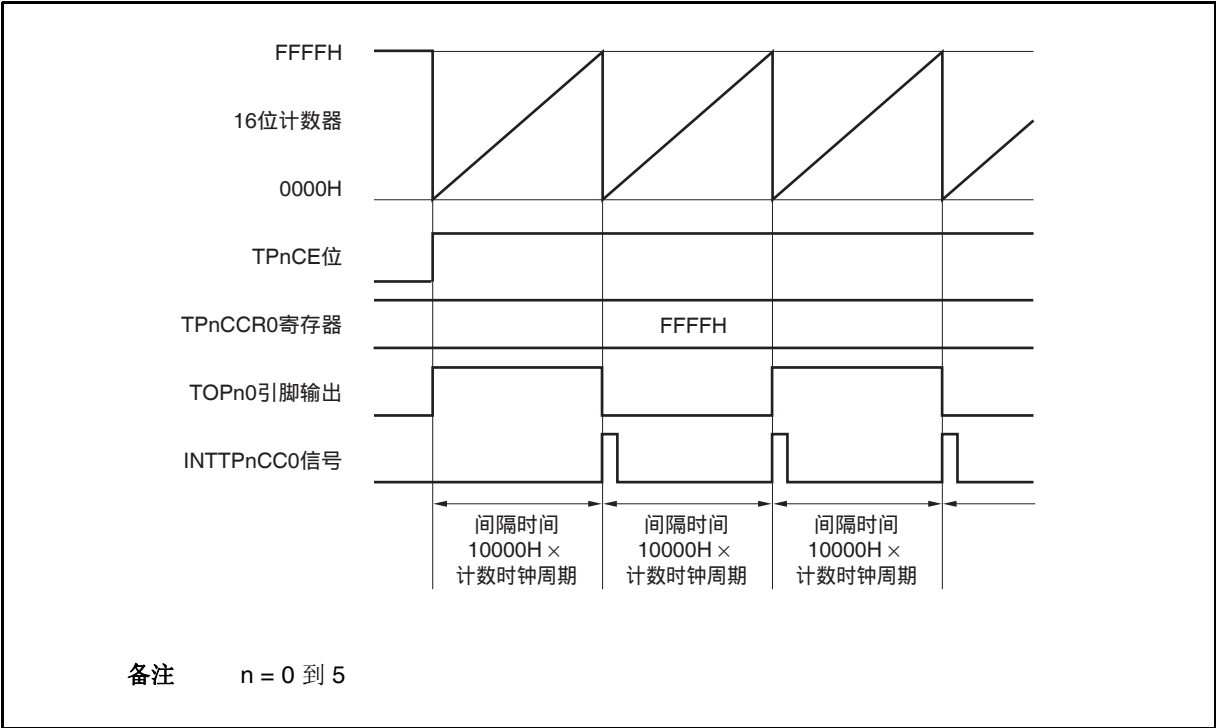
(a) TPnCCR0 寄存器设置为 0000H 时的操作

如果 TPnCCR0 寄存器被设置为 0000H，那么在第二个计数时钟之后(含第二个)每当计数时钟来临，都会产生 INTTPnCC0 信号，并且 TOPn0 引脚的电平都会反转。
16 位计数器的计数值始终保持为 0000H。



(b) TPnCCR0 寄存器被设置为 FFFFH 时的操作

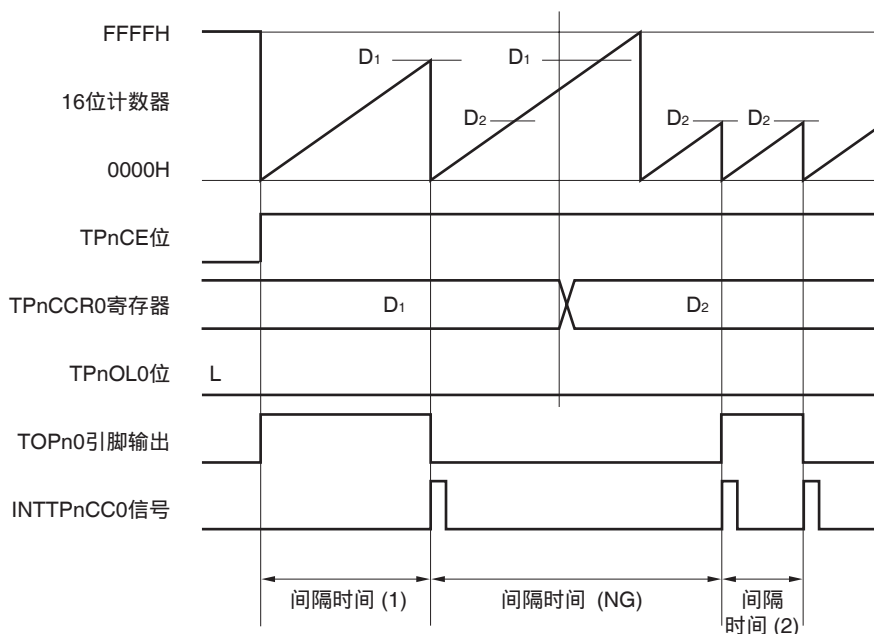
如果 TPnCCR0 寄存器被设置为 FFFFH，那么 16 位计数器将计数到 FFFFH。并在下一个计数时钟时刻同步被清零。同时产生 INTTPnCC0 信号，TOPn0 引脚电平反转。此时，不会产生溢出中断请求信号 (INTTPnOV)，溢出标志位 (TPnOPT0.TPnOVF 位) 也不会被置 1。



(c) 改写 TPnCCR0 寄存器时需要注意的事项

将 TPnCCR0 寄存器的值减小时，要先停止计数，再对设定值进行改变。

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



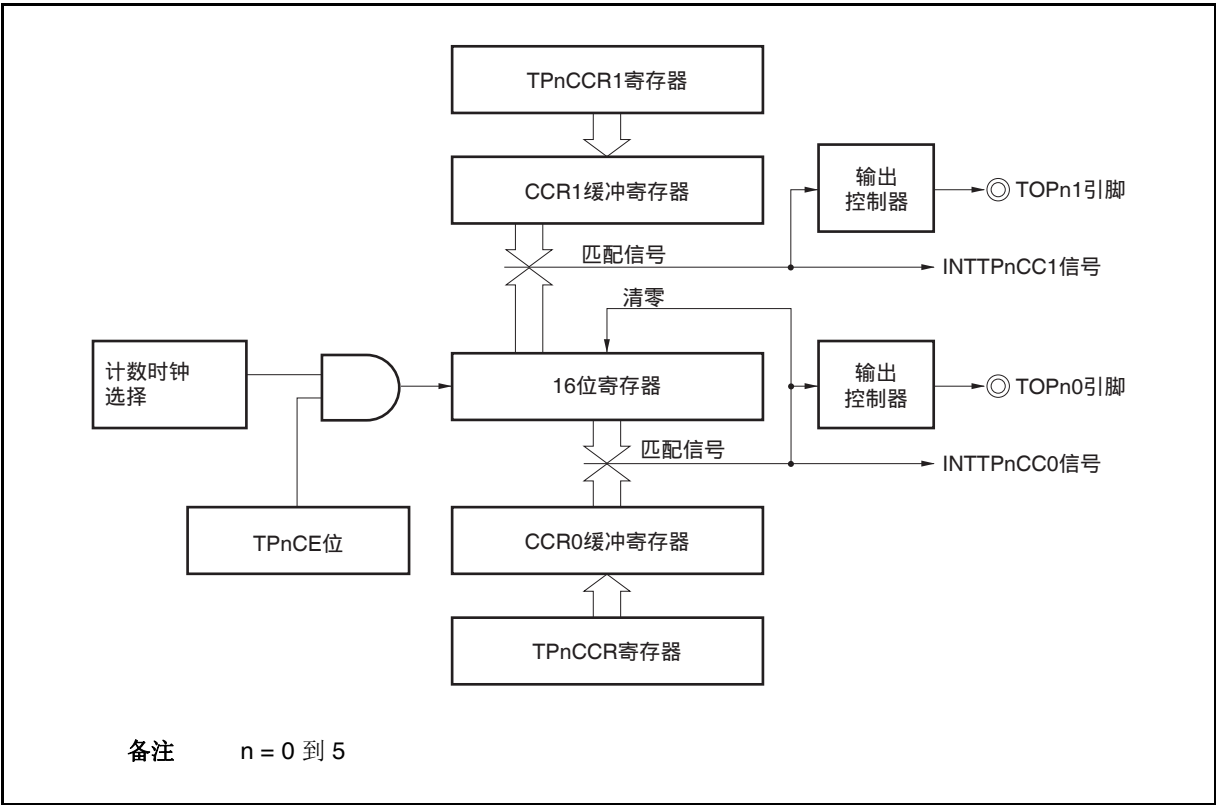
- 备注**
1. 间隔时间 (1): $(D_1 + 1) \times \text{计数时钟周期}$
 间隔时间 (NG): $(10000H + D_2 + 1) \times \text{计数时钟周期}$
 间隔时间 (2): $(D_2 + 1) \times \text{计数时钟周期}$
 2. $n = 0$ 到 5

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

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

(d) TPnCCR1 寄存器的操作

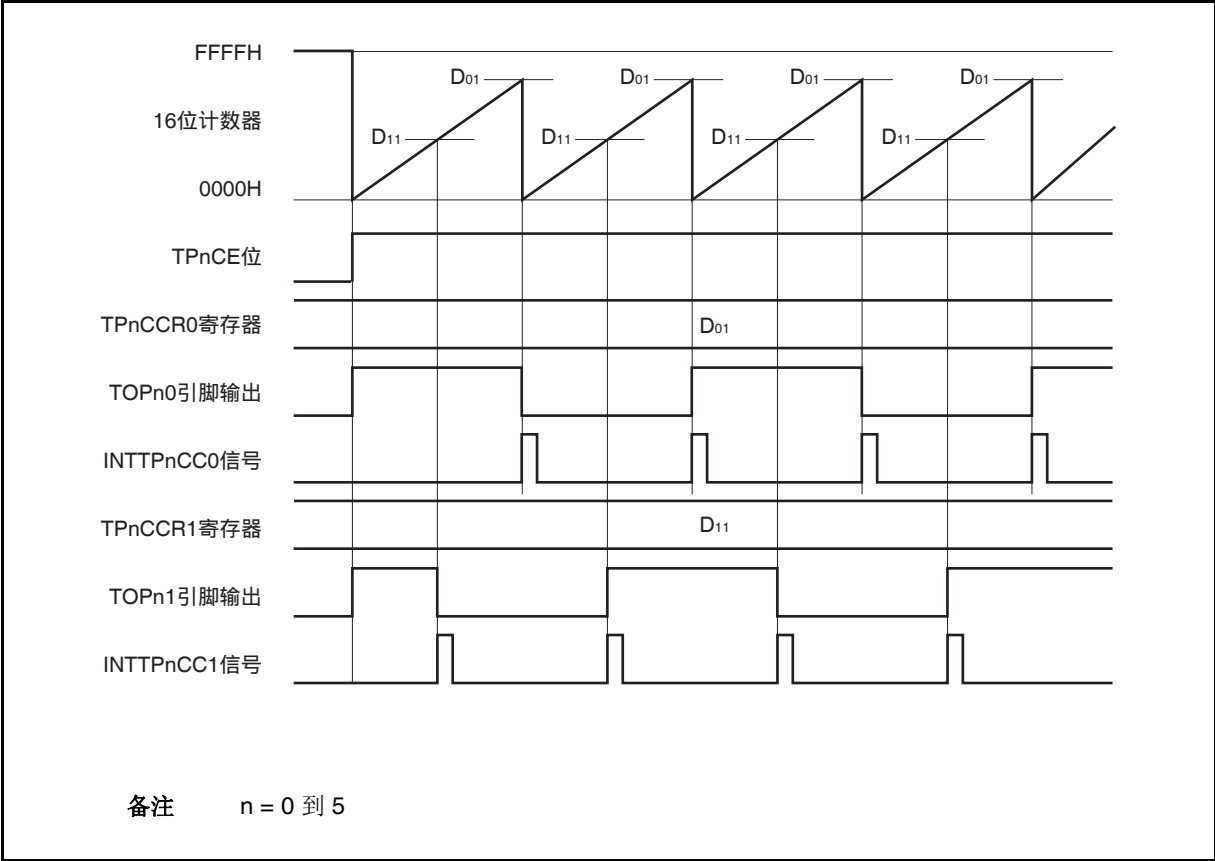
图 7-6. TPnCCR1 寄存器的配置图



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

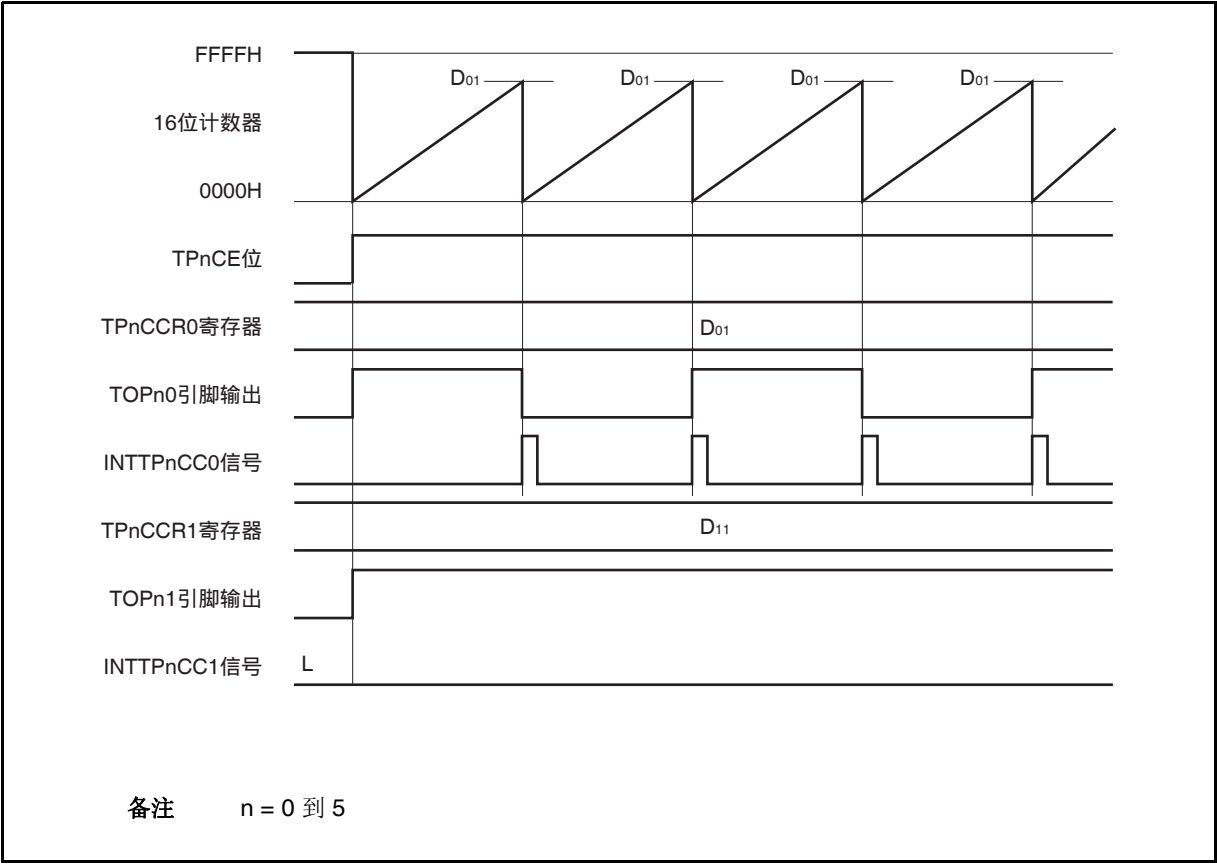
TOPn1 引脚输出的方波周期与 TOPn0 引脚的方波周期相同。

图 7-7. $D_{01} \geq D_{11}$ 时的时序图



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

图 7-8. D₀₁ < D₁₁ 时的时序图



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

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

通常情况下，外部事件计数模式下不使用 TPnCCR1 寄存器。

图 7-9. 外部事件计数模式的配置图

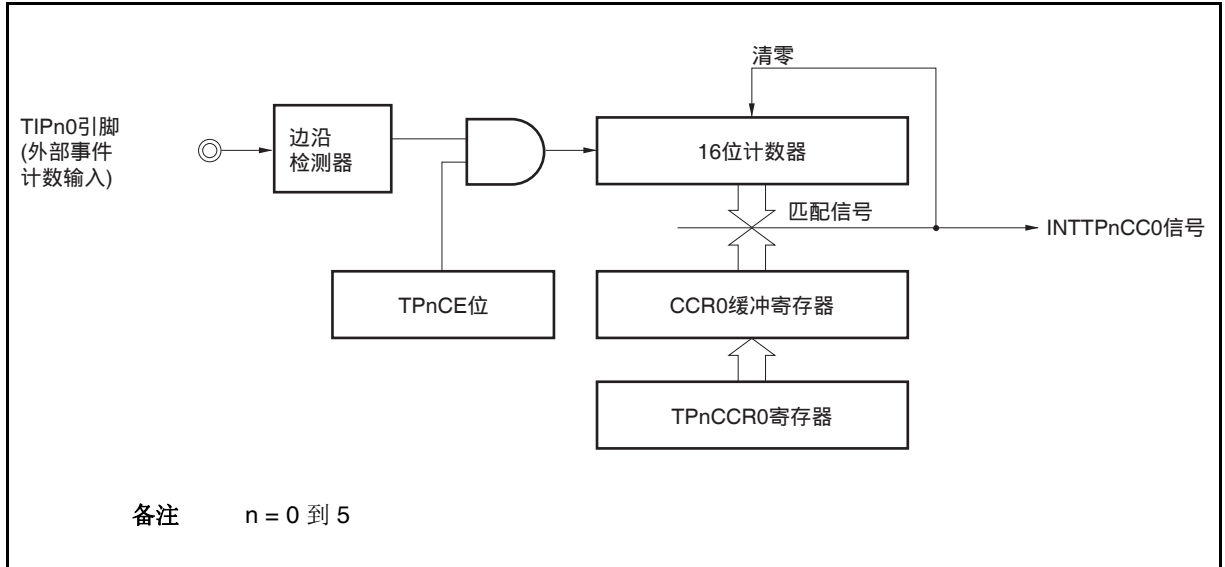
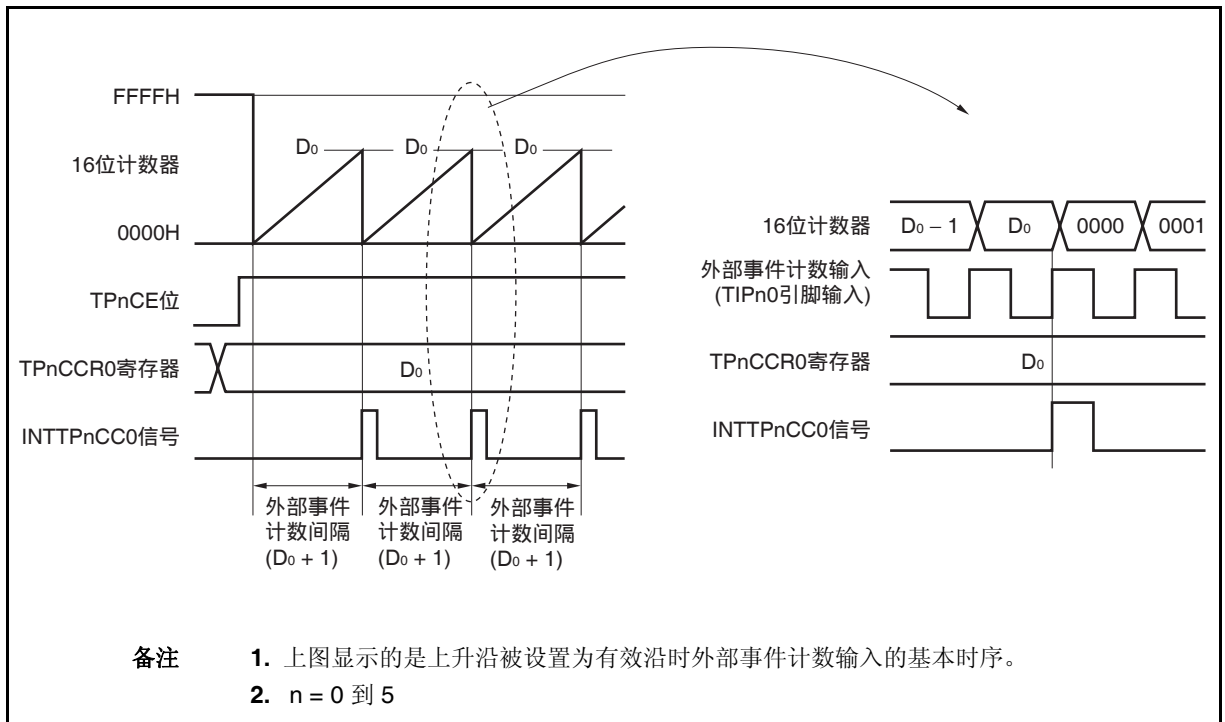


图 7-10. 外部事件计数模式的基本时序



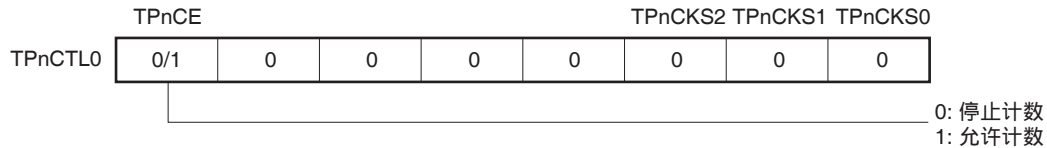
当 TPnCE 位被设置为 1 时，16 位计数器的计数值从 FFFFH 清零为 0000H。每次外部事件计数输入的有效沿被检测时，计数器计数一次。此外，TPnCCR0 寄存器的设定值会被传送到 CCR0 缓冲寄存器中。

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

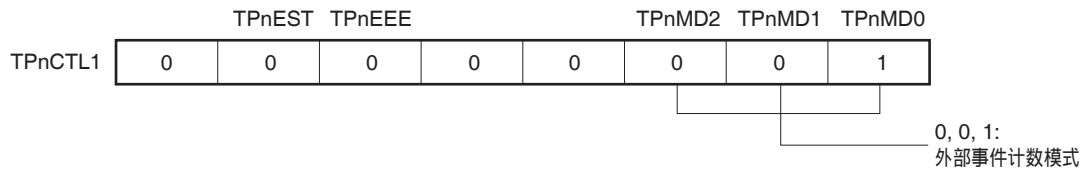
每当检测到(TPnCCR0 寄存器设置值 + 1)次外部事件计数输入的有效沿时，都会产生 INTTPnCC0 信号。

图 7-11. 外部事件计数模式操作的寄存器设置 (1/2)

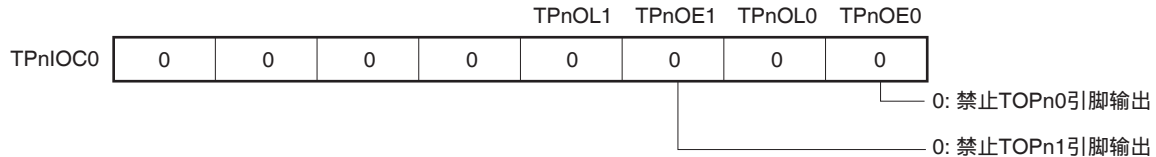
(a) TMPn 控制寄存器 0 (TPnCTL0)



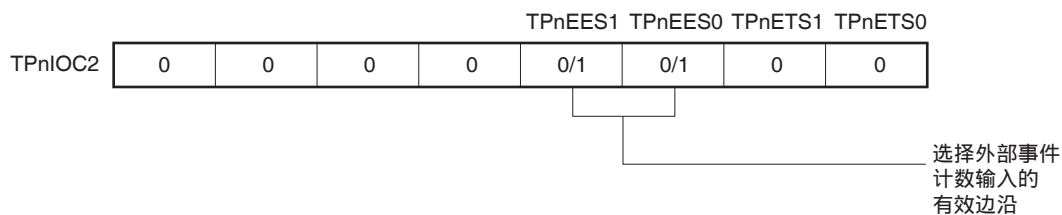
(b) TMPn 控制寄存器 1 (TPnCTL1)



(c) TMPn I/O 控制寄存器 0 (TPnIOC0)



(d) TMPn I/O 控制寄存器 2 (TPnIOC2)



(e) TMPn 计数器读取缓冲寄存器 (TPnCNT)

可以通过读取 TPnCNT 寄存器来读取 16 位计数器的计数值。

图 7-11. 外部事件计数模式操作的寄存器设置 (2/2)

(f) TMPn 捕捉/比较寄存器 0 (TPnCCR0)

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

(g) TMPn 捕捉/比较寄存器 1 (TPnCCR1)

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

因此，请使用相应得中断屏蔽标志 (TPnCCMK1) 将中断请求屏蔽。

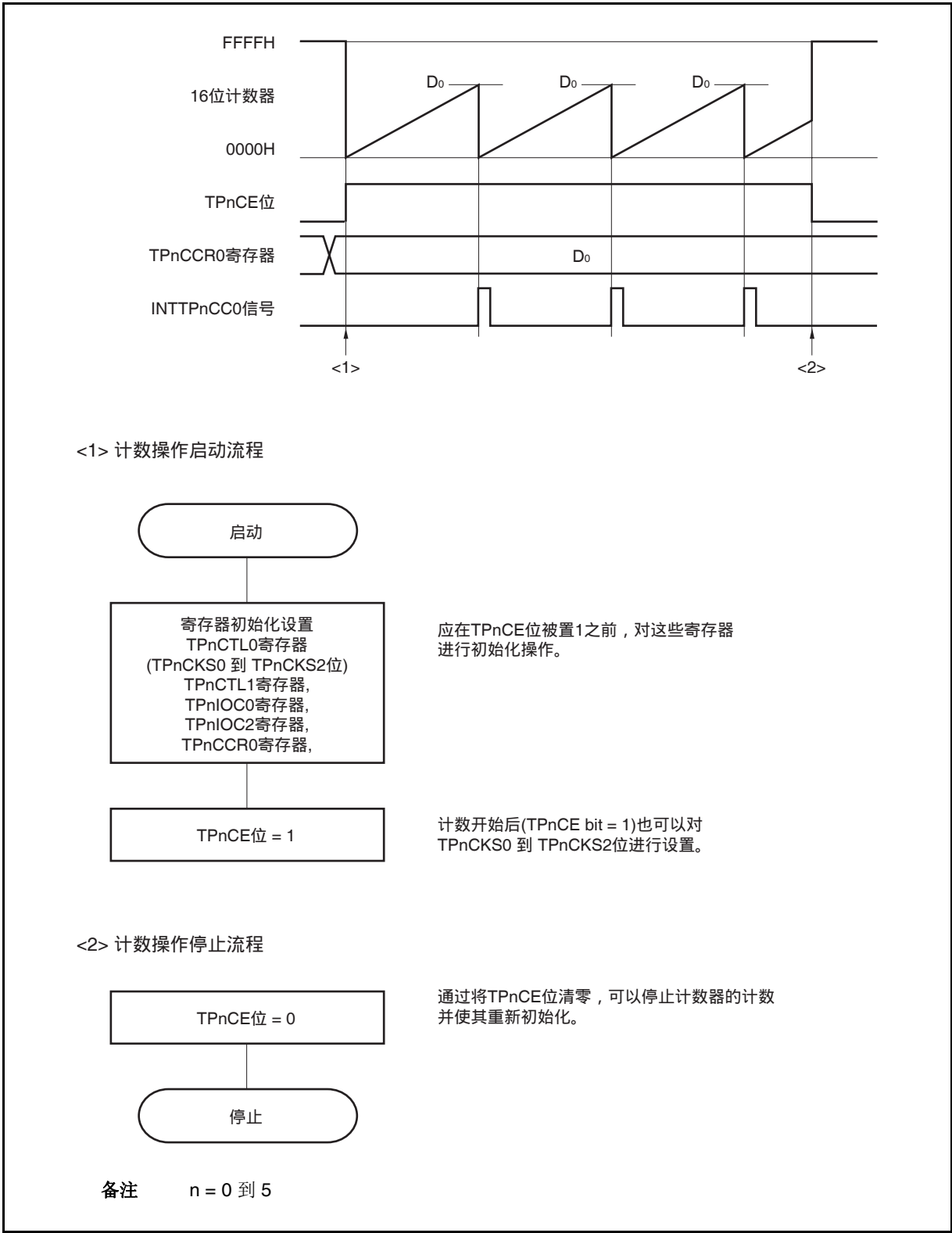
注意事项： 使用外部时钟作为计数时钟时，只能够通过 TIPn0 引脚将外部时钟引入。这时应该将 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位设置为 00 (捕捉触发输入 (TIPn0 引脚)：无边缘检测)。

备注

1. 外部事件计数模式下，并未使用 TMPn I/O 控制寄存器 1 (TPnIOC1) 和 TMPn 选项寄存器 0 (TPnOPT0)。
2. $n = 0$ 到 5

(1) 外部事件计数模式操作流程

图 7-12. 外部事件计数模式的软件处理流程



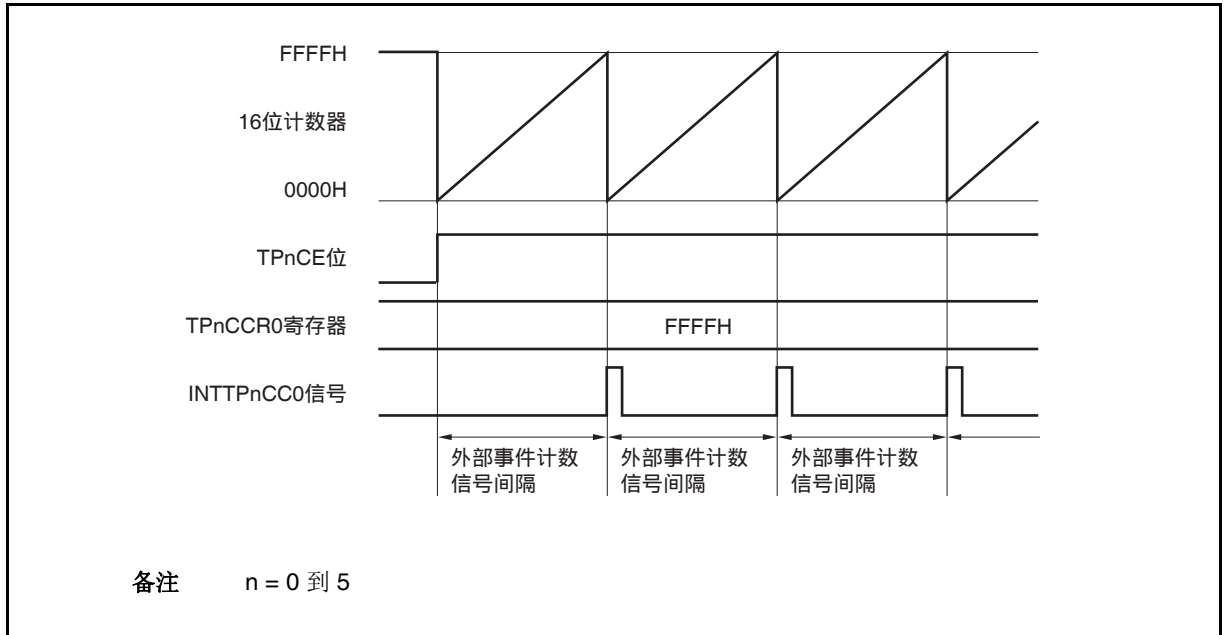
(2) 外部事件计数模式的操作时序

★ 注意事项 1. 在外部事件计数模式中，不要将 TPnCCR0 寄存器设置为 0000H。

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

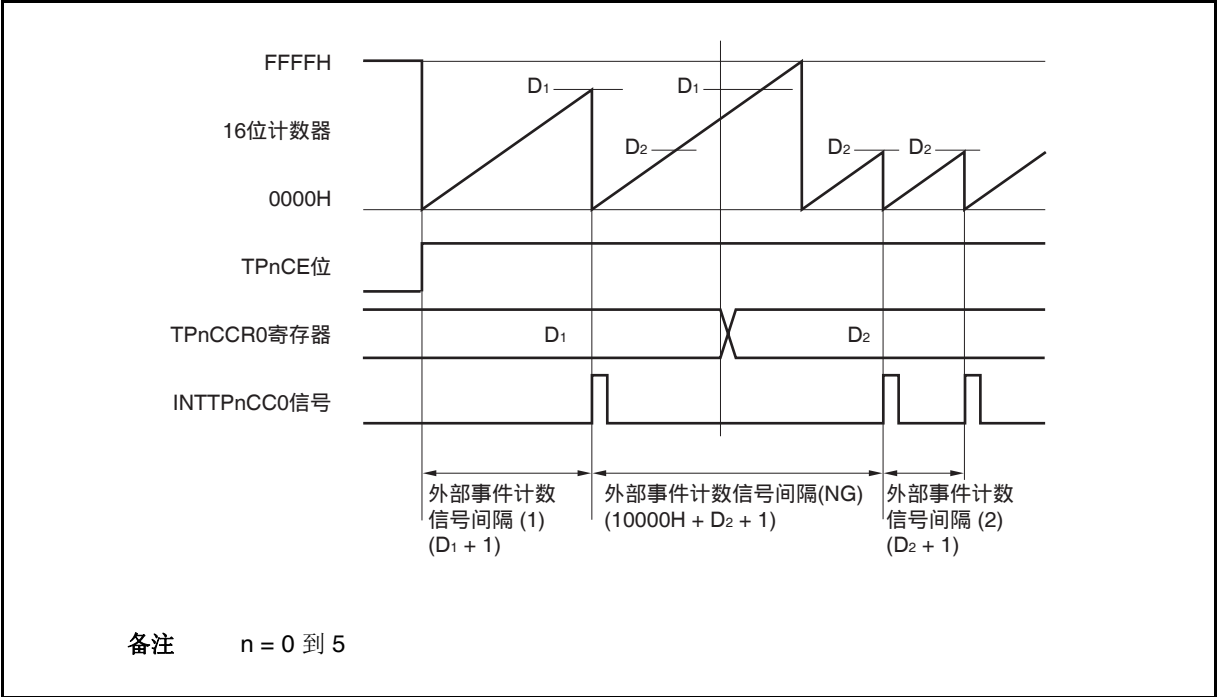
(a) TPnCCR0 寄存器被设置为 FFFFH 时的操作

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



(b) 改写 TPnCCR0 寄存器时需要注意的事项

将 TPnCCR0 寄存器的值减小时，要先停止计数，再对设定值进行改变。
如果在计数过程中将 TPnCCR0 寄存器的值减小，那么 16 位计数器可能产生溢出。

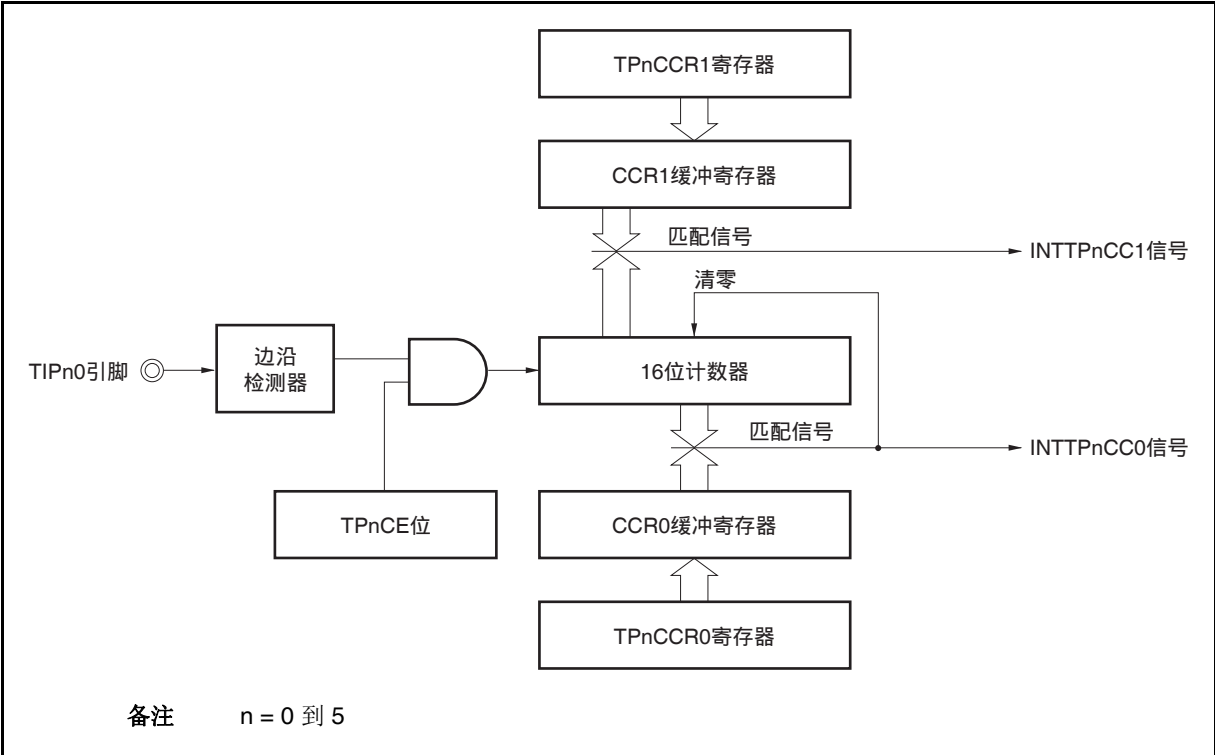


如果当前计数值大于 D₂但小于 D₁，将 TPnCCR0 寄存器的值从 D₁改变为 D₂，新的设定值将在改写后立即从 TPnCCR0 寄存器传送到 CCR0 缓冲寄存器中，那么，16 位计数器的计数值将与 D₂进行比较。由于计数值已经超过了 D₂，因此 16 位计数器会继续计数到 FFFFH 并产生溢出，之后重新从 0000H 开始计数。在重新计数过程中，计数值与 D₂匹配时，才会产生 INTTPnCC0 信号。因此可见，INTTPnCC0 信号不会在预期的事件计数次数“(D₁ + 1) 次”或“(D₂ + 1) 次”时产生，而是在一个“(10000H + D₂ + 1) 次”的计数次数时产生。

★

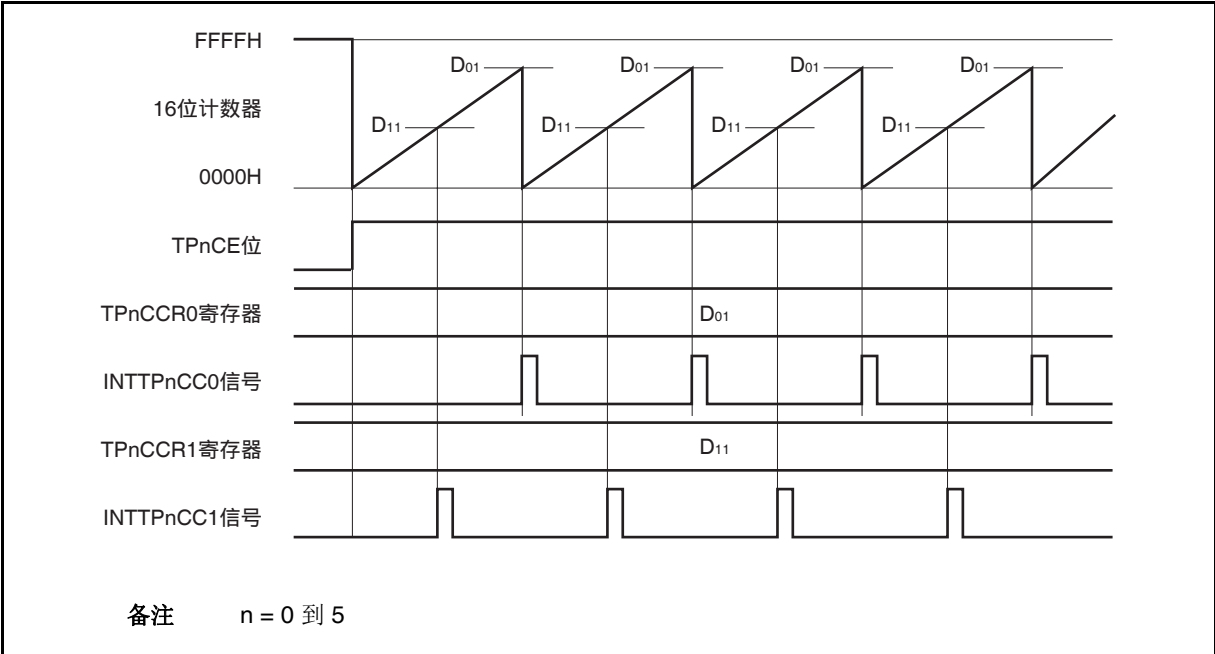
(c) TPnCCR1 寄存器的操作

图 7-13. TPnCCR1 寄存器的配置



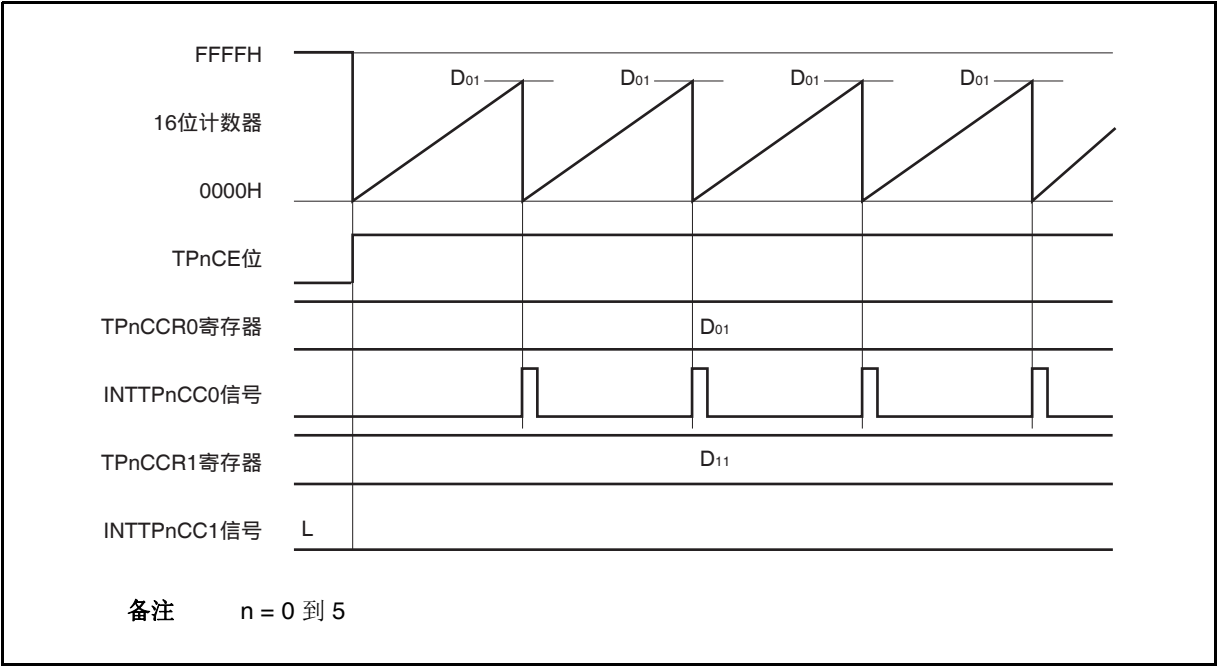
如果 TPnCCR1 寄存器的值小于 TPnCCR0 寄存器的值，那么 INTTPnCC1 信号将每周期产生一次。

图 7-14. $D_{01} \geq D_{11}$ 时的时序图



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

图 7-15. D₀₁ < D₁₁ 时的时序图



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

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

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

图 7-16. 外部触发脉冲输出模式的配置

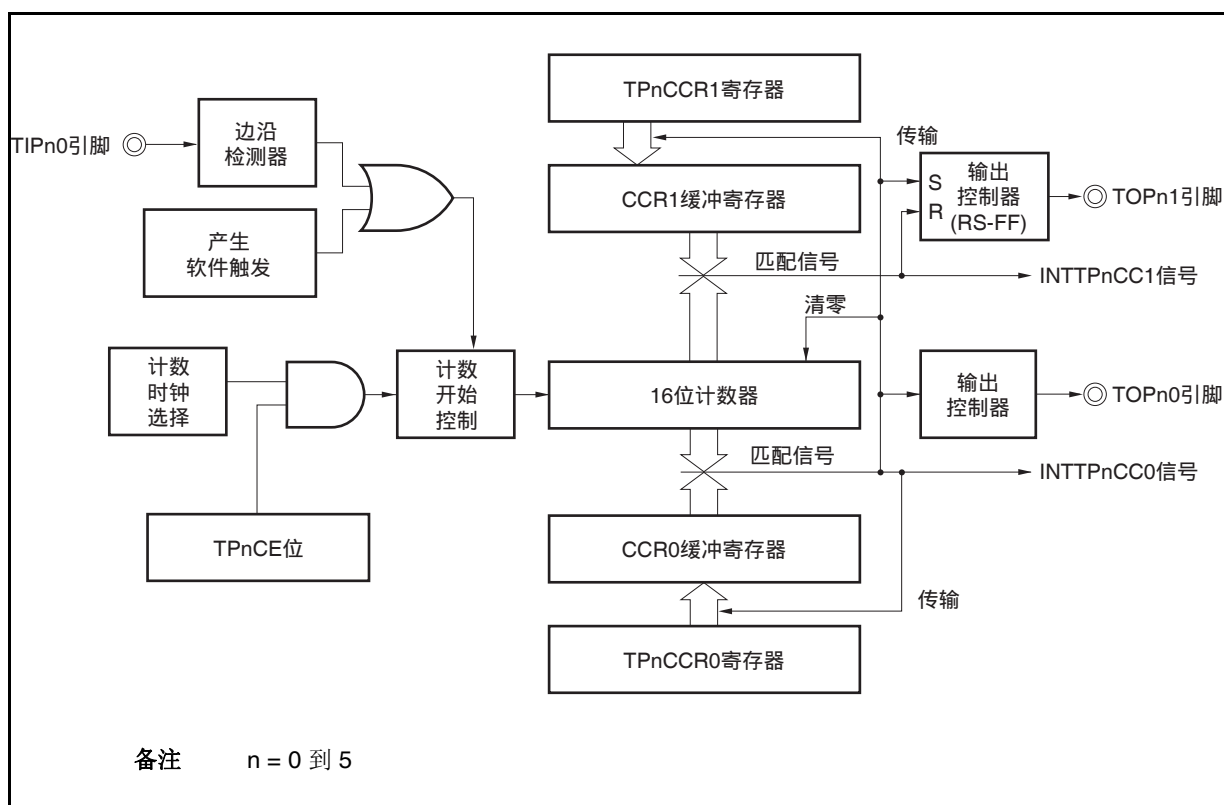
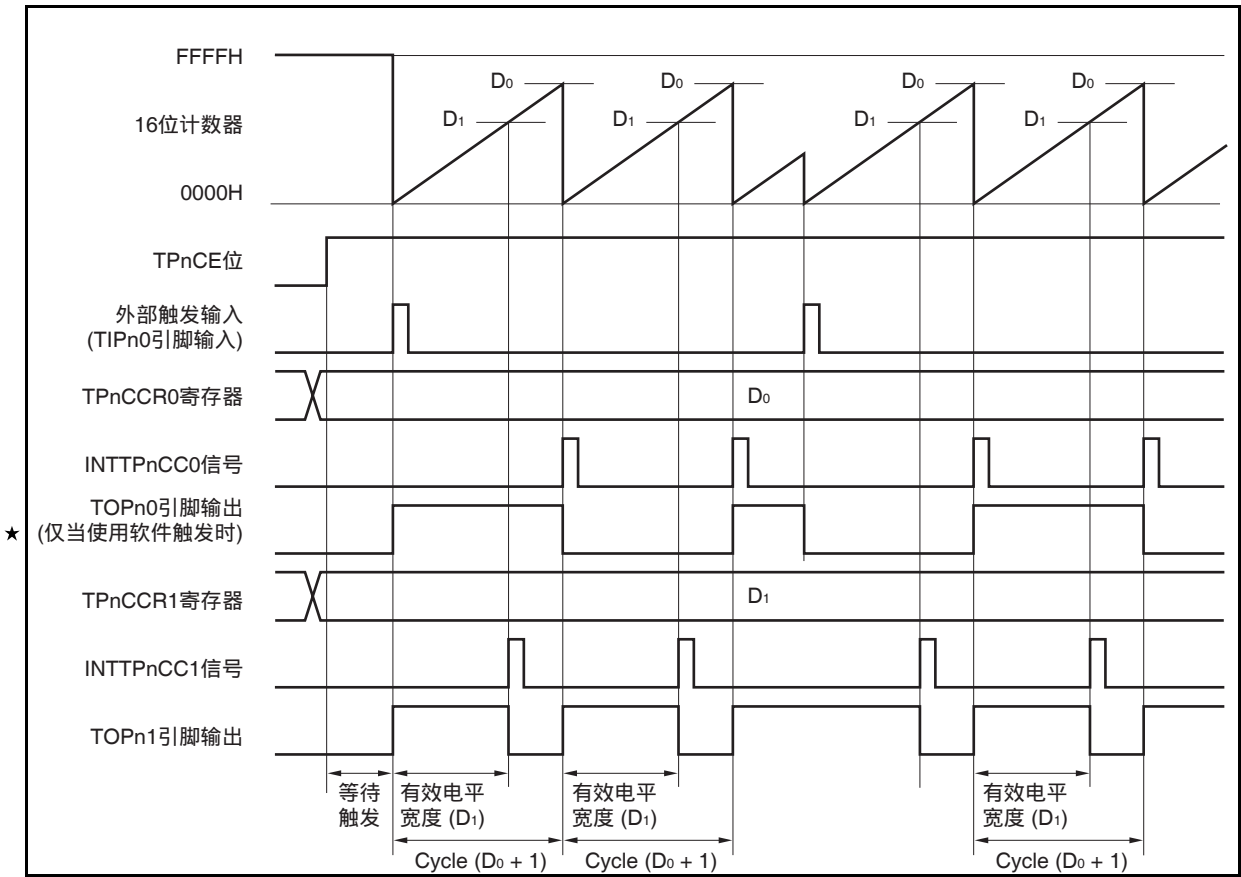


图 7-17. 外部触发脉冲输出模式的基本时序



TPnCTL0.TPnCE 位被置 1 后，16 位定时器/事件计数器 P 进入等待触发信号的状态。触发被检测后，16 位计数器从 FFFFH 清零为 0000H，同时开始计数并从 TOPn1 引脚输出一个 PWM 波形。如果在计数器计数过程中，再次★检测到触发信号，那么计数器将被清零并重新开始计数。(TOPn0 引脚的输出电平反转。TOPn1 在触发产生时输出高电平(无论当前状态为高/低)。)

PWM 波形的有效电平宽度，周期以及占空比系数可由以下的算式得出。

有效电平宽度 = (TPnCCR1 寄存器的设置值) × 计数时钟周期
 周期 = (TPnCCR0 寄存器的值 + 1) × 计数时钟周期
 占空比系数 = (TPnCCR1 寄存器的值)/(TPnCCR0 寄存器的值 + 1)

比较匹配中断请求信号 INTTPnCC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器中的值相匹配后产生，同时 16 位计数器也会被清零。比较匹配中断请求信号 INTTPnCC1 会在 16 位计数器的计数值与 CCR1 缓冲寄存器中的值相匹配时产生。

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

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

备注 n = 0 到 5, m = 0, 1

图 7-18. 外部触发脉冲输出模式的寄存器设置 (1/2)

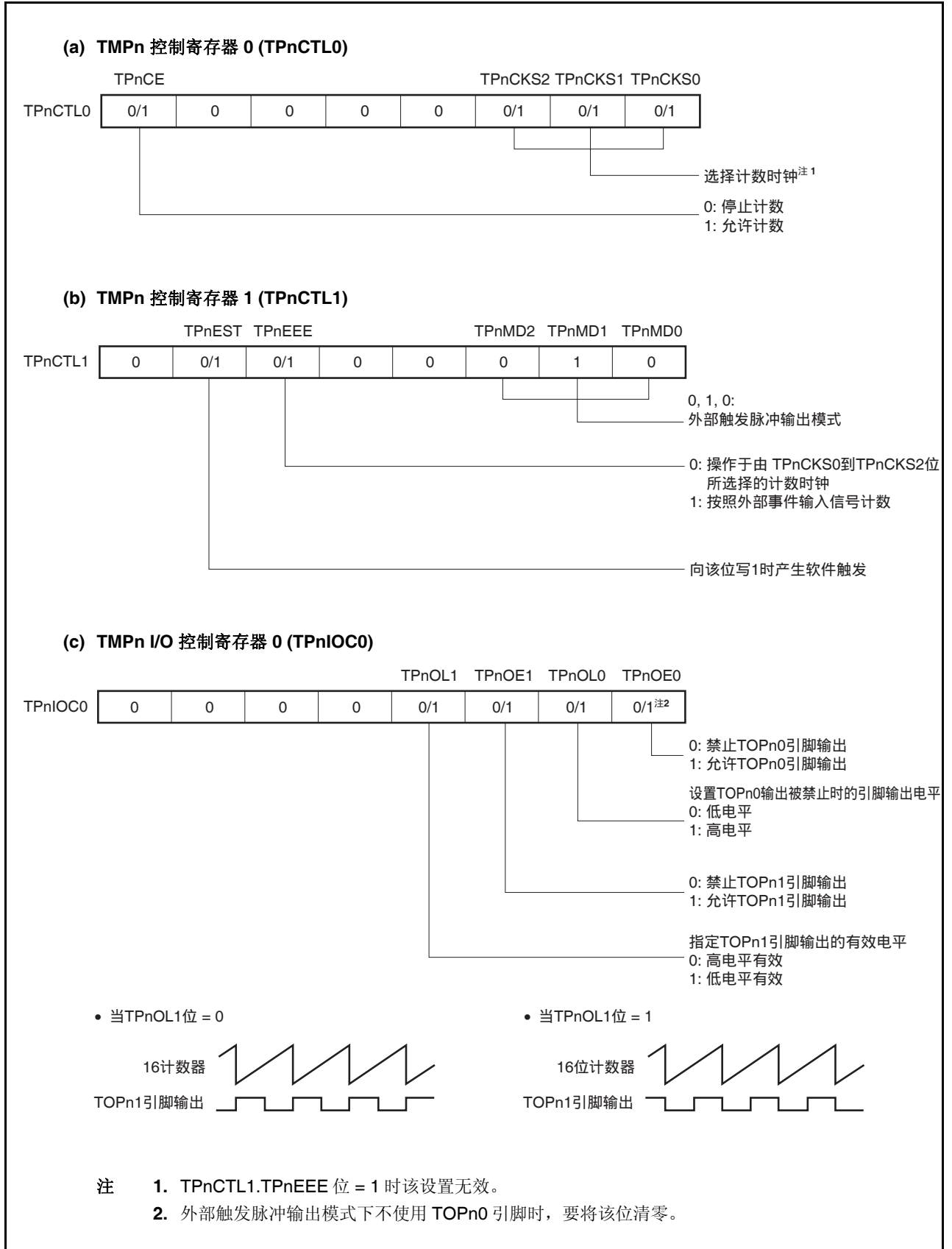
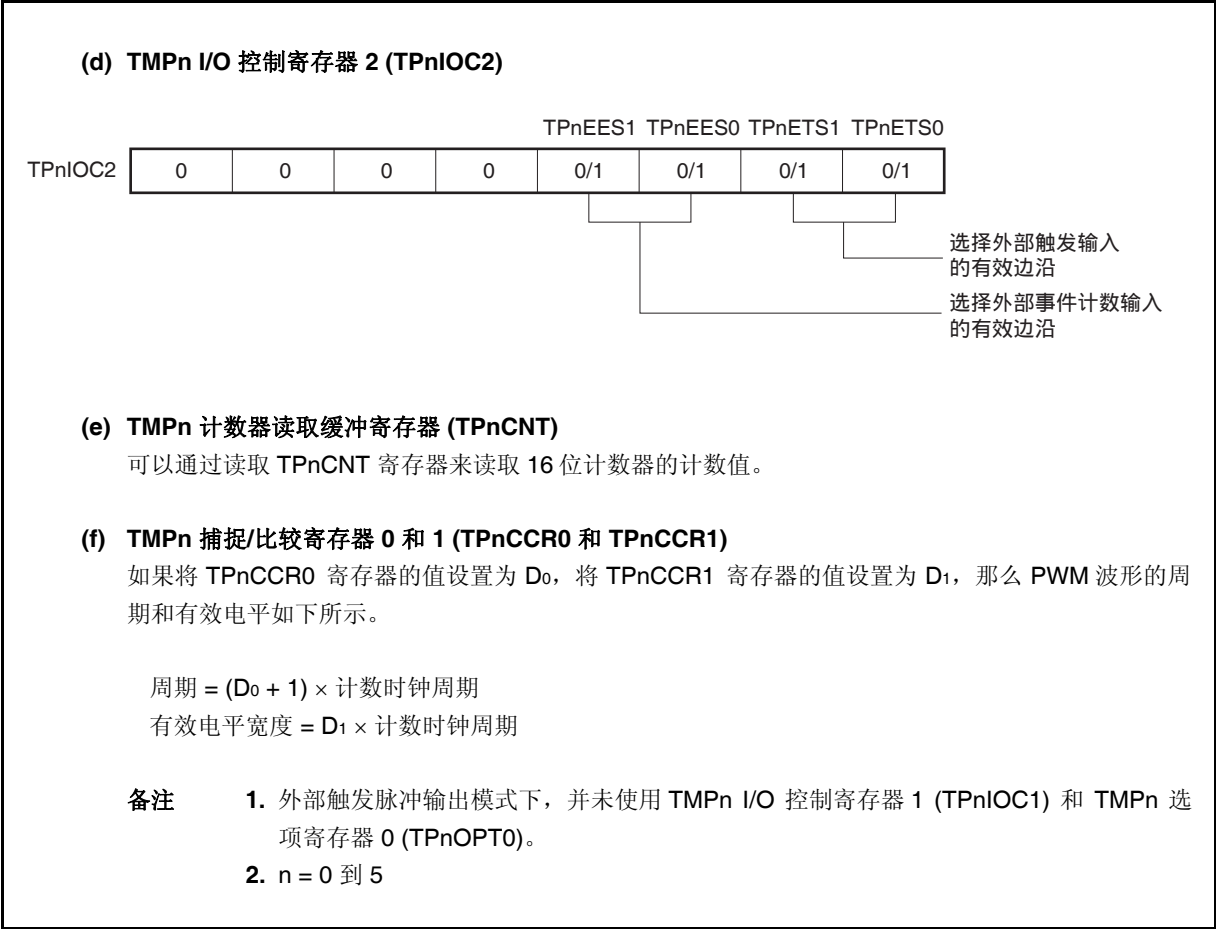


图 7-18. 外部触发脉冲输出模式的寄存器设置 (2/2)



(1) 外部触发脉冲输出模式的操作系统

图 7-19. 外部触发脉冲输出模式的软件处理流程 (1/2)

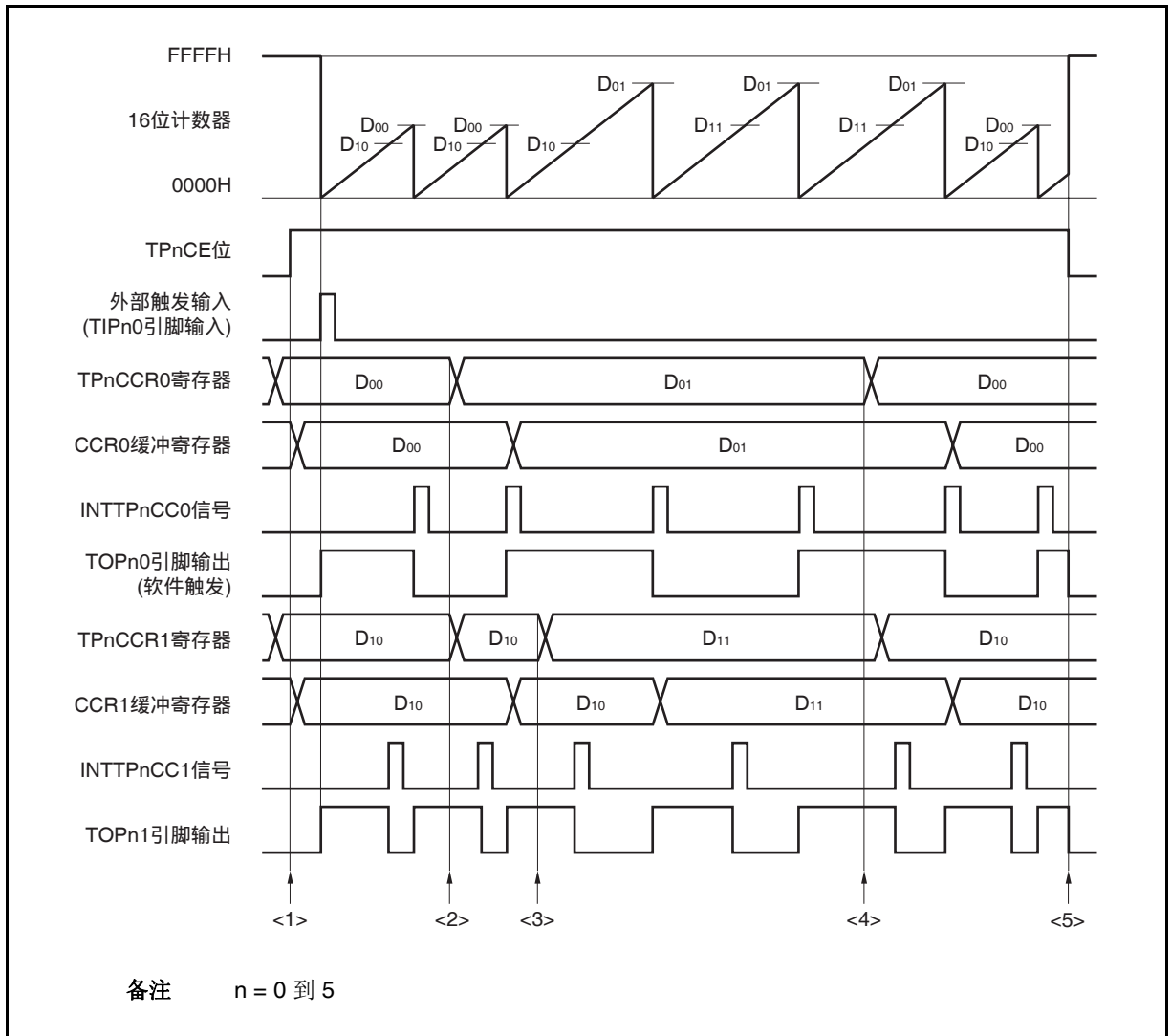
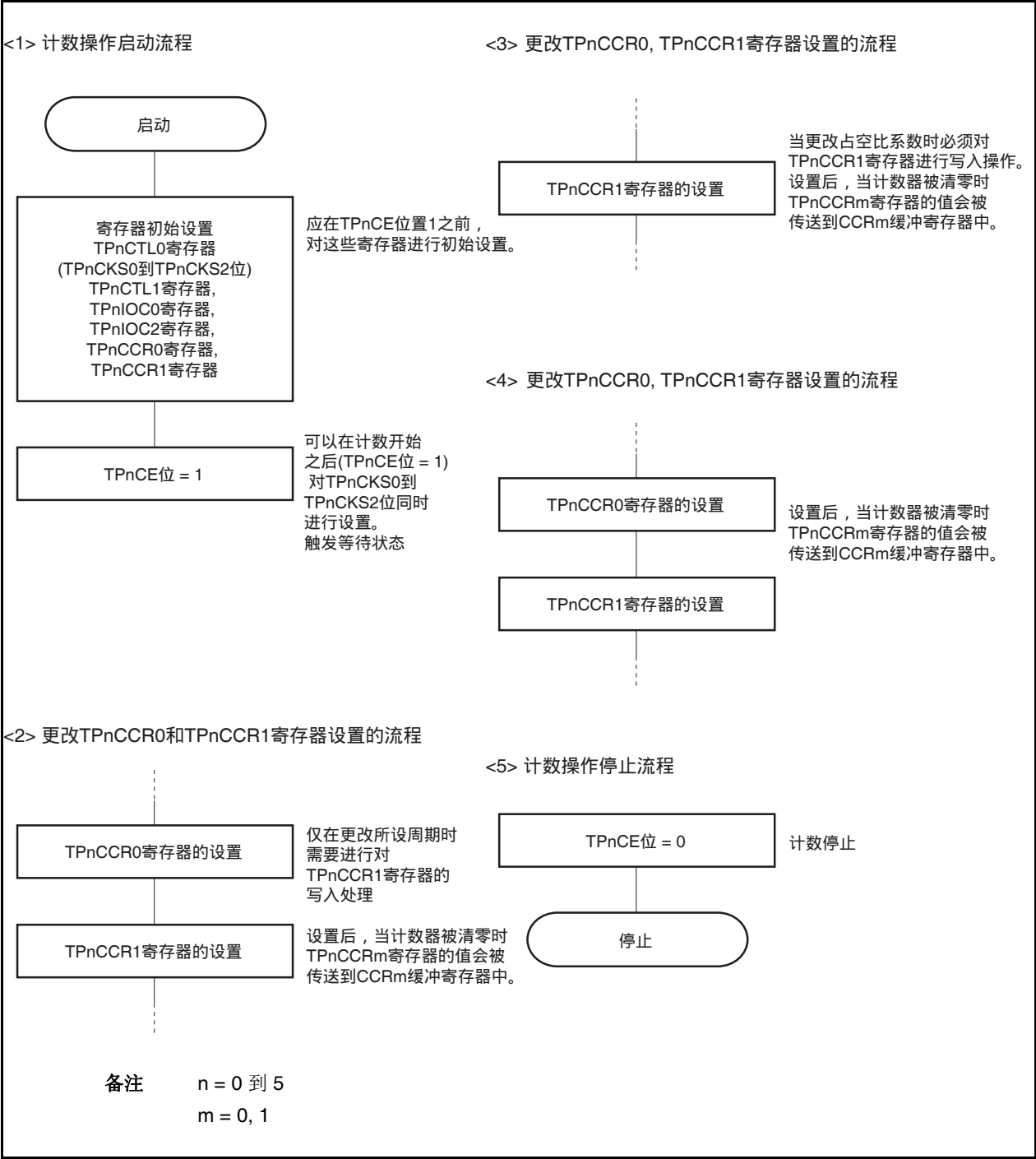


图 7-19. 外部触发脉冲输出模式的软件处理流程 (2/2)

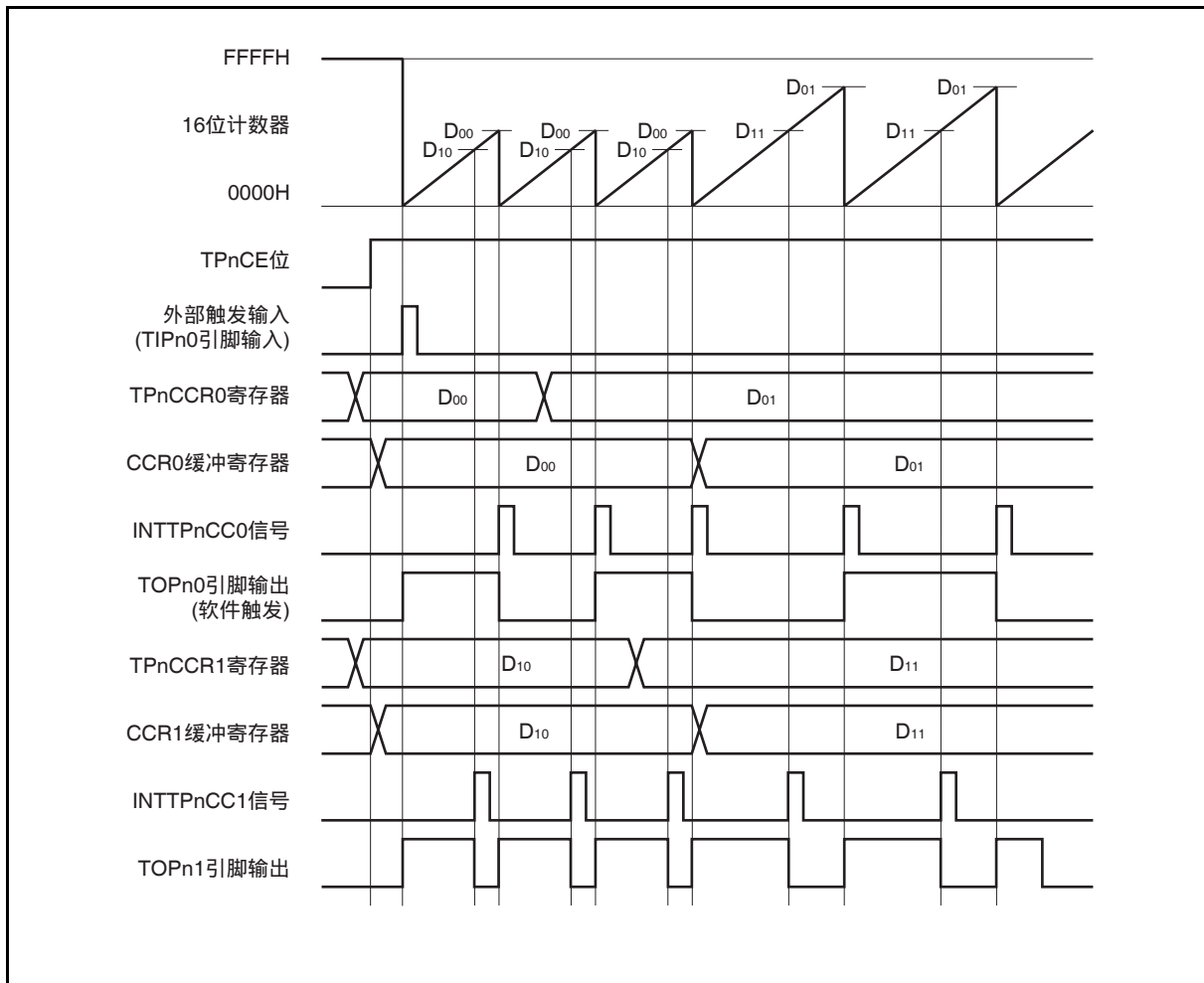


(2) 外部触发脉冲输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

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

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



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

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

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

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

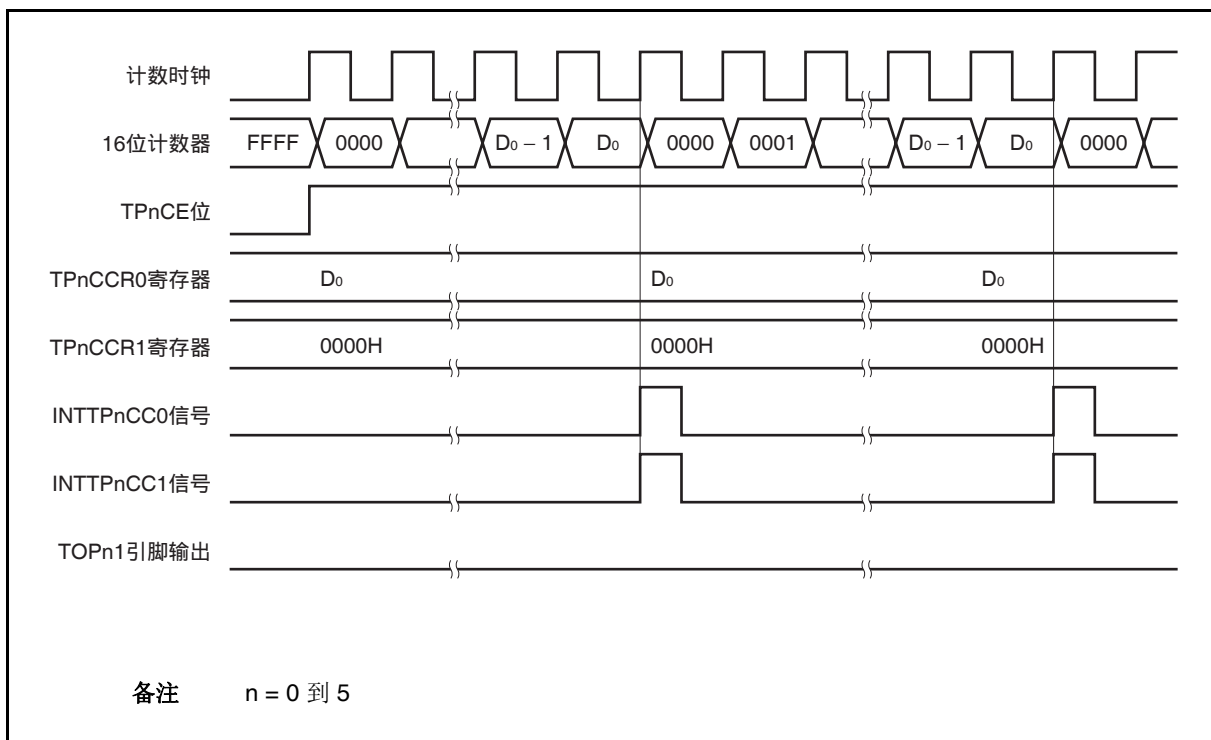
数据被写入 TPnCCR1 寄存器之后，当 16 位计数器被清零时，TPnCCRm 寄存器中的数值才会被传送至 CCRm 缓冲寄存器中。此时 CCRm 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

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

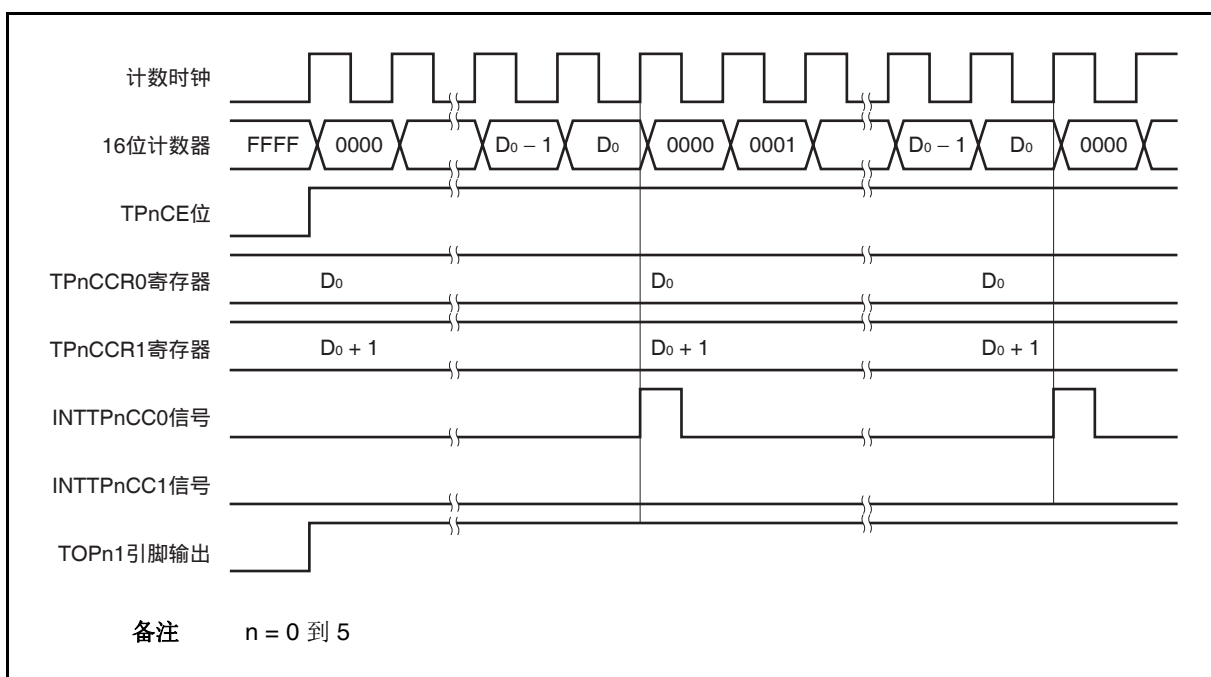
备注 n = 0 到 5
 m = 0, 1

(b) PWM 波形的 0%/100% 输出

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

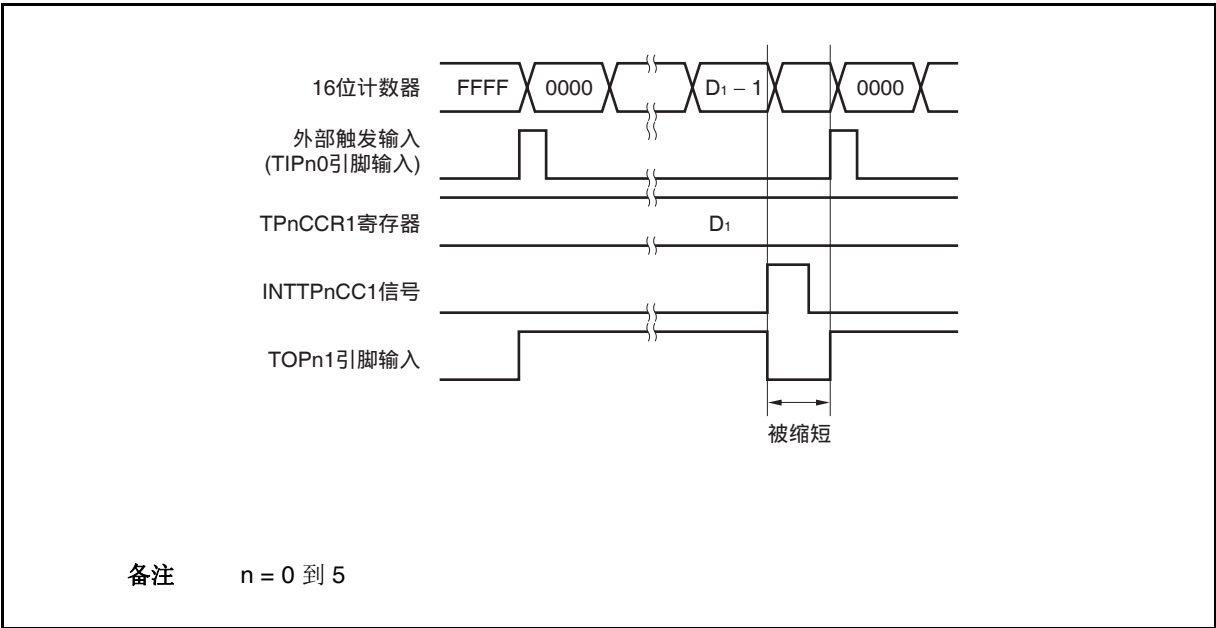


若要输出 100% 波形，则需将 TPnCCR1 寄存器设置为 (TPnCCR0 寄存器值 + 1)。TPnCCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。

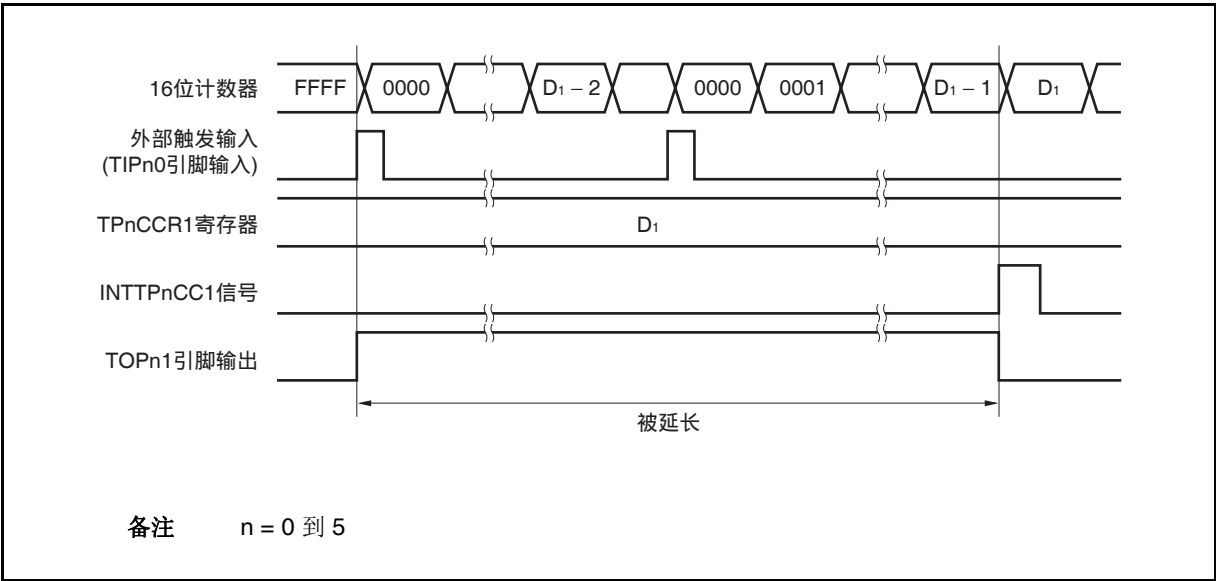


(c) 触发检测和 TPnCCR1 寄存器匹配之间的冲突

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

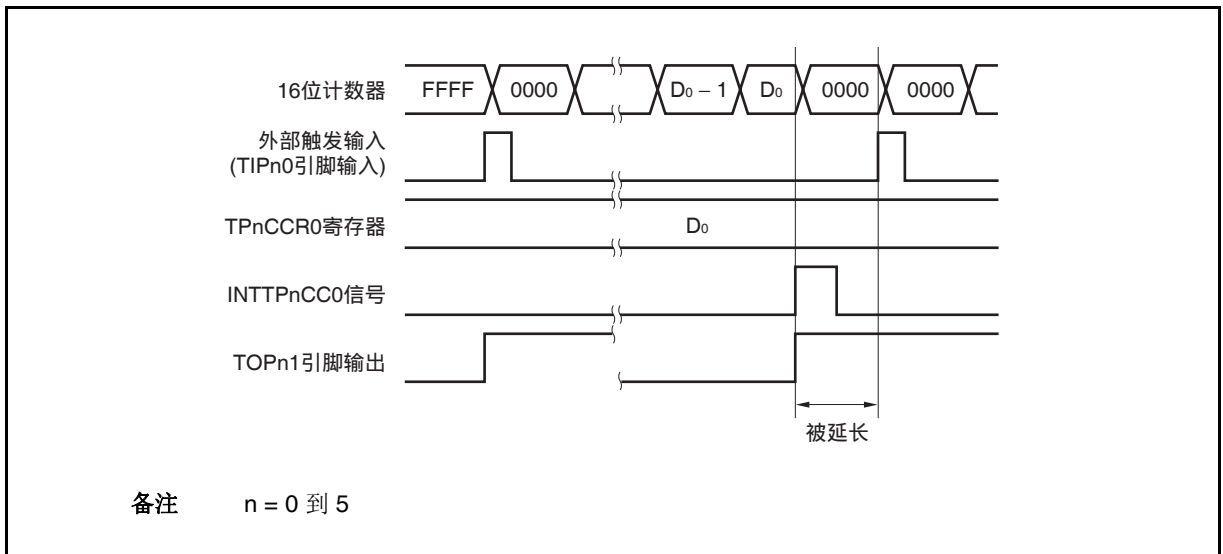


如果在 INTTPnCC1 信号产生之前检测到触发信号，那么 INTTPnCC1 信号将不会产生，此时，16 位计数器被清零并重新开始计数。TOPn1 引脚的输出也会继续保持为有效电平。这种情况下，PWM 波形的有效电平时间就会被延长。

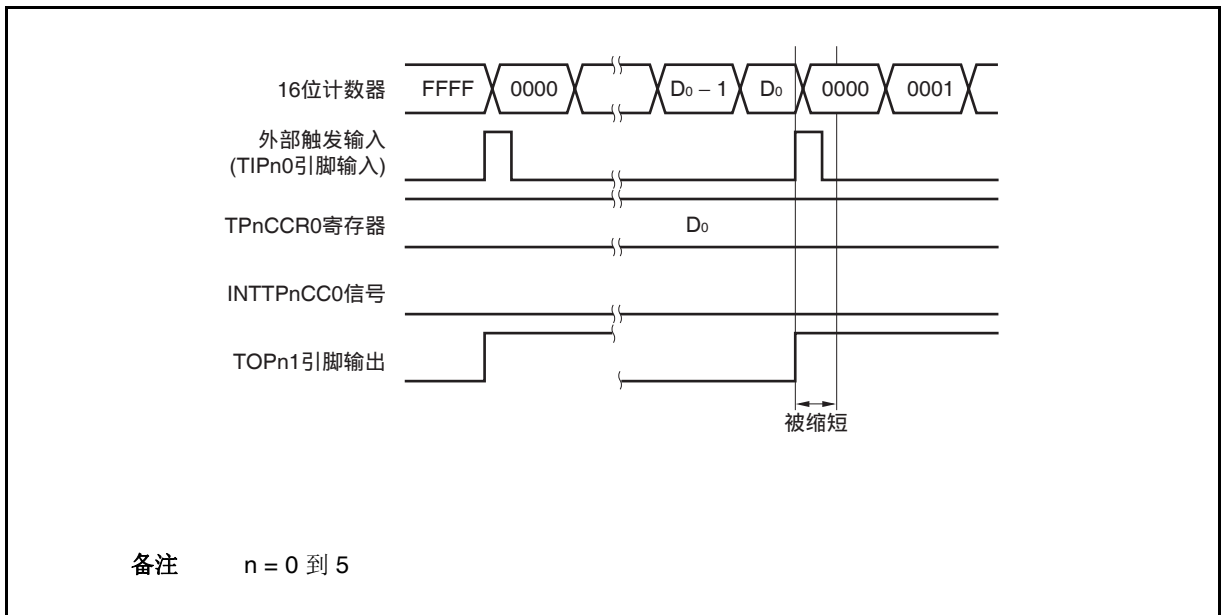


(d) 触发检测和 TPnCCR0 寄存器匹配之间的冲突

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

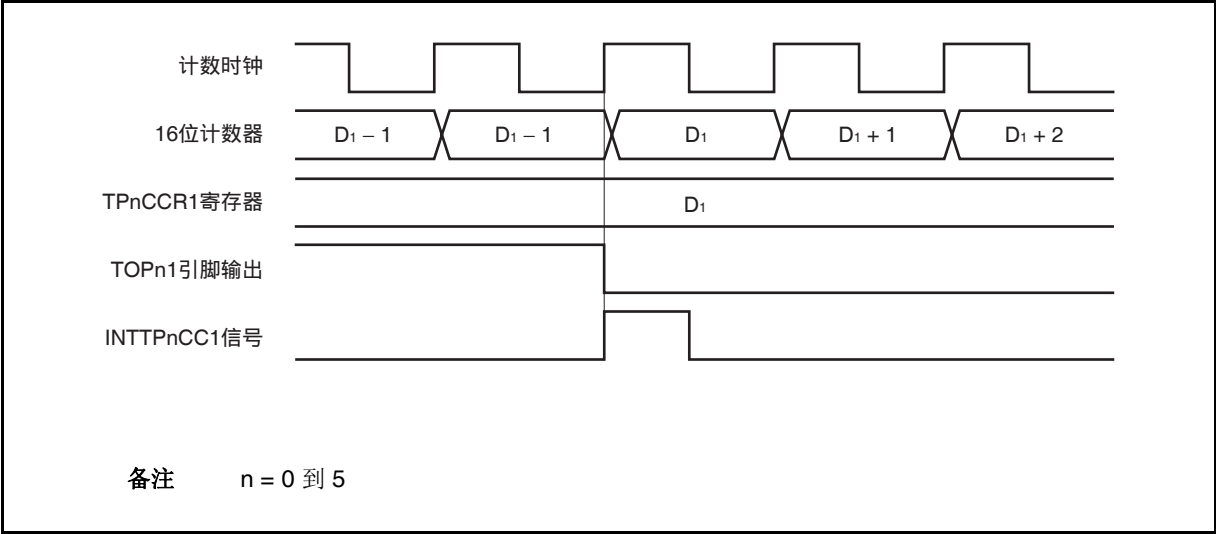


如果在 INTTPnCC0 信号产生之前检测到触发信号，那么 INTTPnCC0 信号将不会产生。16 位计数器将被清零并重新开始计数，TOPn1 引脚输出也会改变为有效电平。这种情况下，PWM 波形的非有效电平时间就会被缩短。



(e) 比较匹配中断请求信号的产生时序 (INTTPnCC1)

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



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

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

7.5.4 单脉冲输出模式 (TPnMD2 到 TPnMD0 位 = 011)

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

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

图 7-20. 单脉冲输出模式的配置图

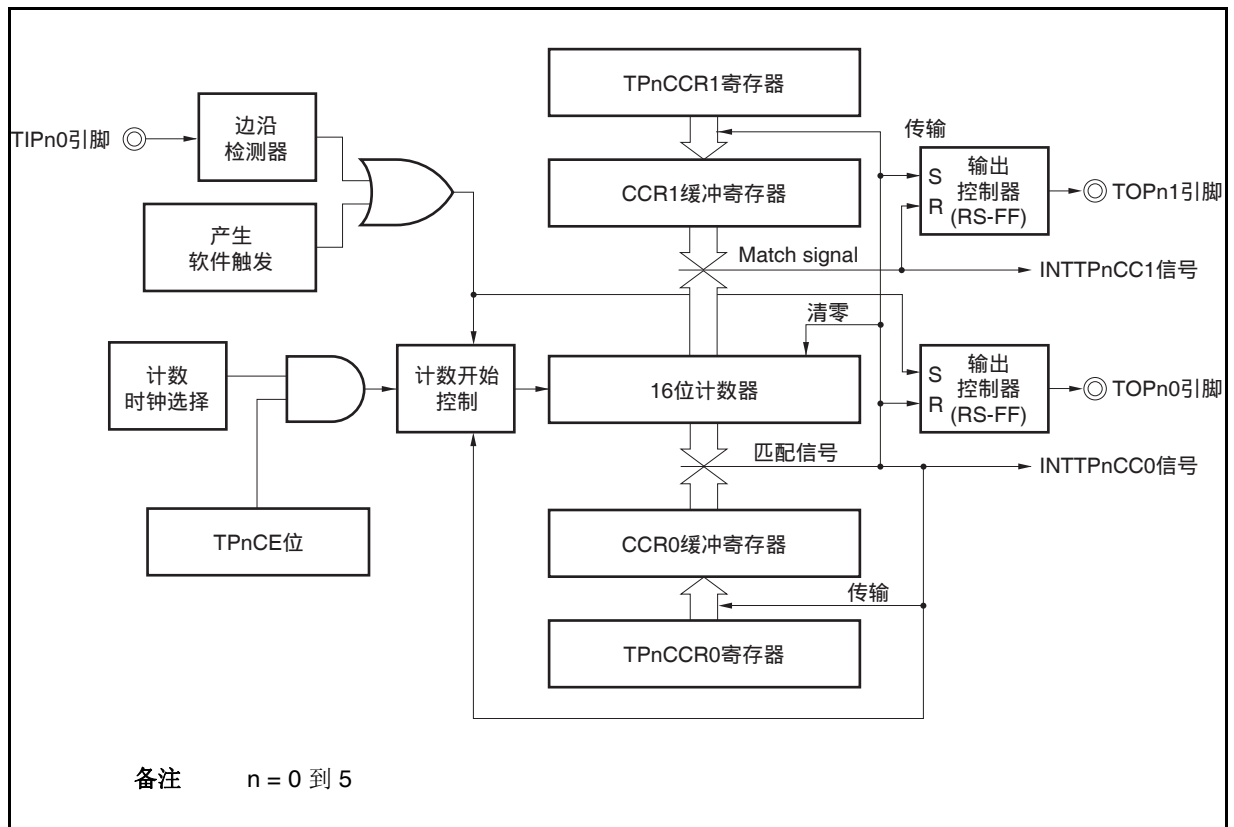
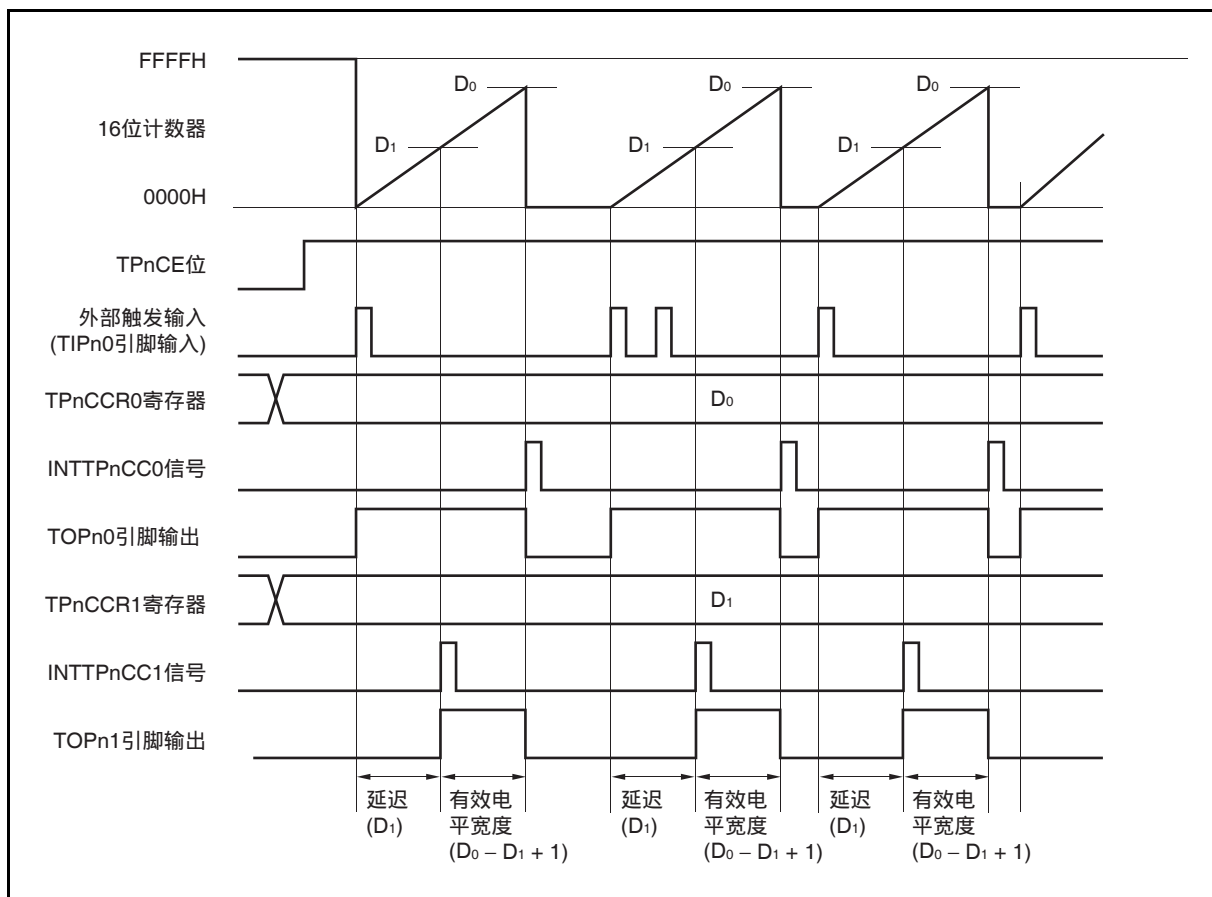


图 7-21. 单脉冲输出模式的基本时序



TPnCTL0.TPnCE 位被置 1 后, 16 位定时器/事件计数器 P 进入等待触发信号的状态。触发被检测后, 16 位计数器从 FFFFH 清零为 0000H, 同时开始计数并从 TOPn1 引脚输出一个单脉冲波形。单脉冲波形输出后, 16 位计数器将被重置为 FFFFH 并停止计数, 16 位定时器/时间计数器 P 重新进入等待触发信号的状态。如果在单脉冲波形输出过程中再次检测到触发信号, 那么这个触发信号将被忽略。

单脉冲波形的输出延迟时间以及有效电平宽度可由以下算式得出。

$$\text{输出延迟时间} = (\text{TPnCCR1 寄存器的设置值}) \times \text{计数时钟周期}$$

$$\text{有效电平宽度} = (\text{TPnCCR0 寄存器的值} - \text{TPnCCR1 寄存器的值} + 1) \times \text{计数时钟周期}$$

比较匹配中断请求信号 **INTTPnCC0** 会在 16 位计数器的计数值与 **CCR0** 缓冲寄存器中的值相匹配时的下一个计数时钟时产生。比较匹配中断请求信号 **INTTPnCC1** 会在 16 位计数器的计数值与 **CCR1** 缓冲寄存器中的值相匹配时产生。

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

备注 $n = 0$ 到 5
 $m = 0, 1$

图 7-22. 单脉冲输出模式的寄存器设置 (1/2)

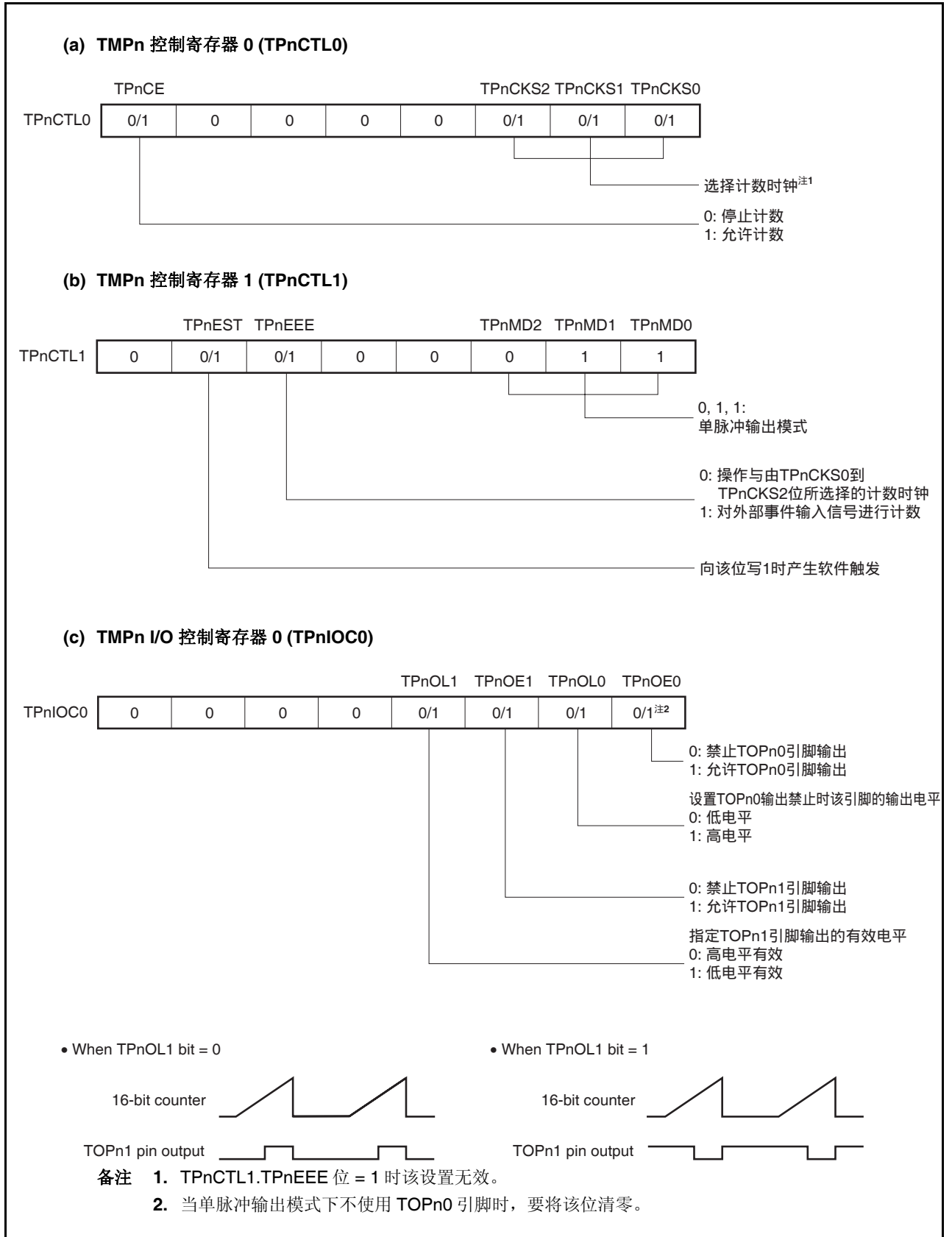
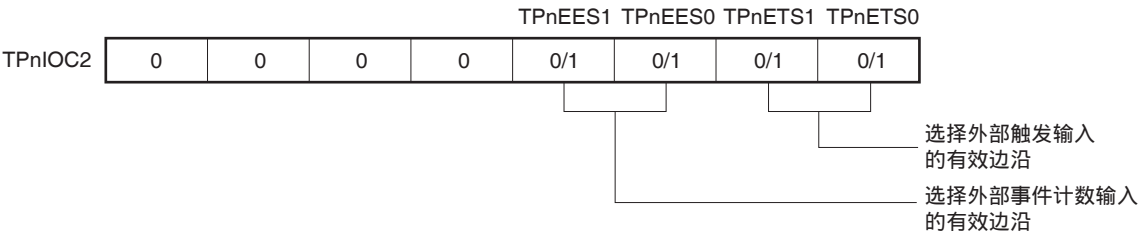


图 7-22. 单脉冲输出模式的寄存器设置 (2/2)

(d) TMPn I/O 控制寄存器 2 (TPnIOC2)



(e) TMPn 计数器读取缓冲寄存器 (TPnCNT)

可以通过读取 TPnCNT 寄存器来读取 16 位计数器的计数值。

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

如果将 TPnCCR0 寄存器的值设置为 D_0 ，将 TPnCCR1 寄存器的值设置为 D_1 ，那么单脉冲波形的有效电平宽度和输出延迟时间如下所示。

有效电平宽度 = $(D_0 - D_1 + 1) \times$ 计数时钟周期

输出延迟时间 = $D_1 \times$ 计数时钟周期

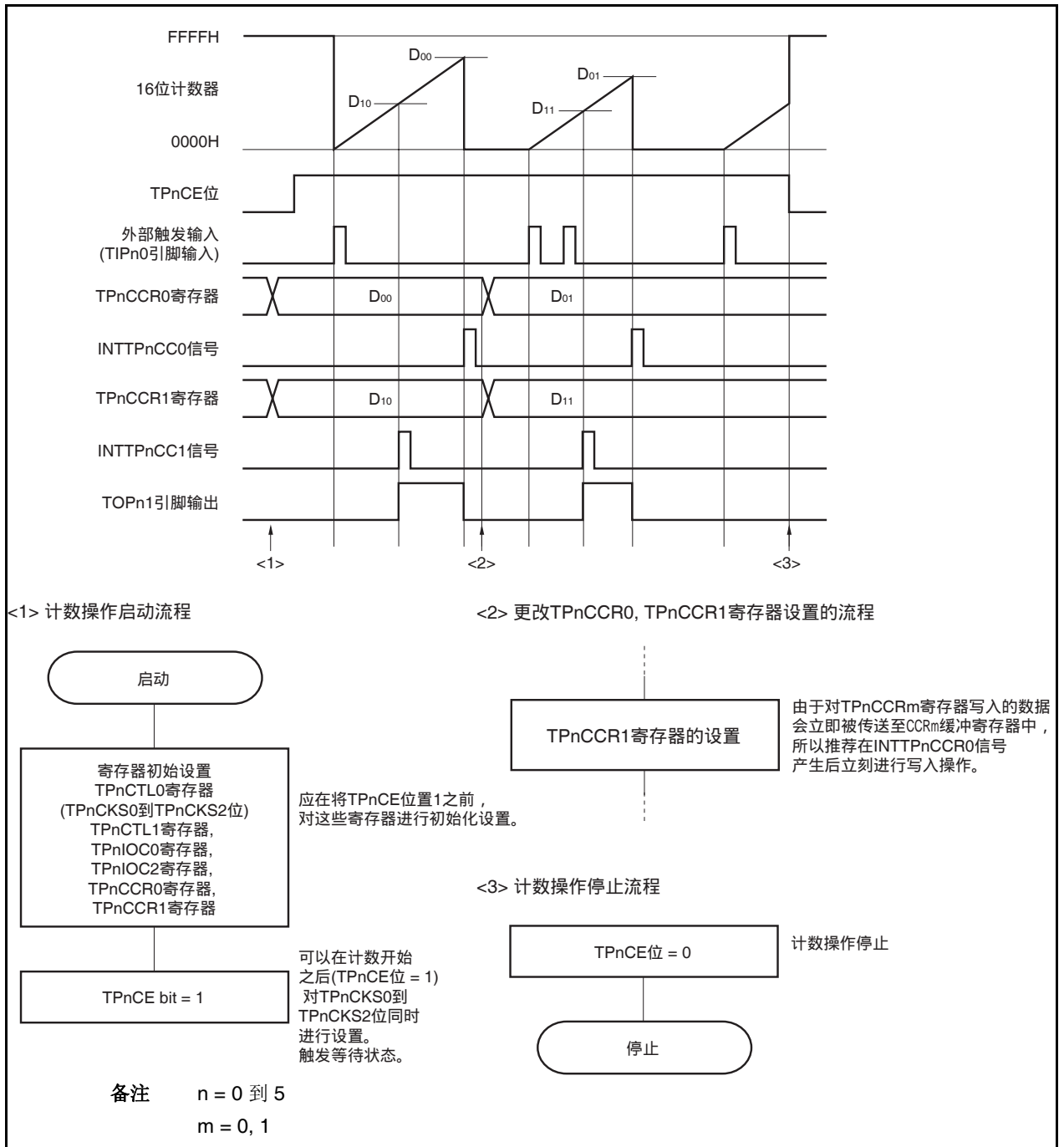
注意事项 如果 TPnCCR1 寄存器的设定值大于 TPnCCR0 中的值，那么单脉冲输出模式下将无法输出单脉冲波形。

备注

- 1. 单脉冲输出模式下，并未使用 TMPn I/O 控制寄存器 1 (TPnIOC1) 和 TMPn 选项寄存器 0 (TPnOPT0)。
- 2. $n = 0$ 到 5

(1) 单脉冲输出模式的操作流程

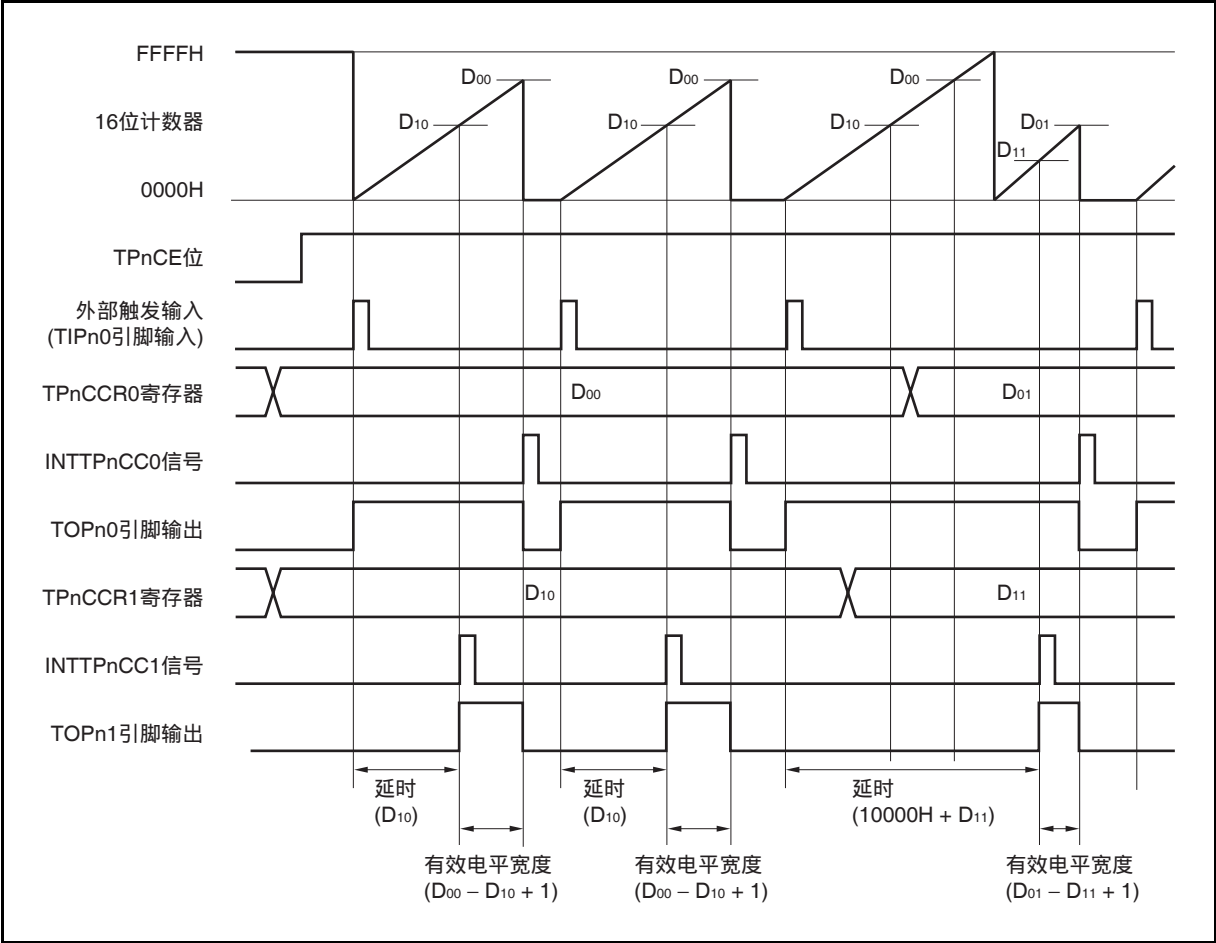
图 7-23. 单脉冲输出模式的软件操作流程



(2) 单脉冲输出模式的操作时序

(a) 改变 TPnCCRm 寄存器时需要注意的事项

将 TPnCCRm 寄存器的值减小时，要先停止计数，再对设定值进行改变。
如果在计数过程中将 TPnCCRm 寄存器的值减小，那么 16 位计数器可能产生溢出。

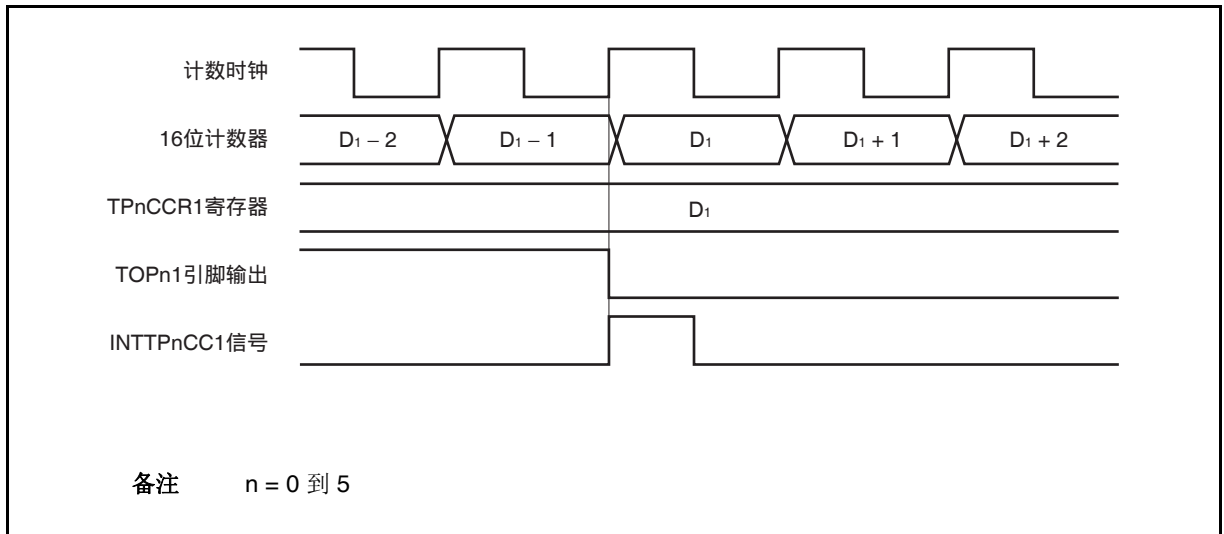


当将 TPnCCR0 寄存器的值从 D₀₀ 改写为 D₀₁，TPnCCR1 寄存器的值从 D₁₀ 改写为 D₁₁ 时，这里假设 D₀₀ > D₀₁ 且 D₁₀ > D₁₁，如果 TPnCCR1 寄存器被改写时，16 位计数器的计数值大于 D₁₁ 且小于 D₁₀，并且 TPnCCR0 寄存器被改写时，计数器的计数值大于 D₀₁ 且小于 D₀₀，两个设定值都将在改写后立即生效并与计数值进行比较。此时计数器会计数到 FFFFH 然后从 0000H 重新开始计数。当计数值与 D₁₁ 匹配时，将产生 INTTPnCC1 信号，同时 TOPn1 引脚电平变为有效电平。当计数值与 D₀₁ 匹配时，将产生 INTTPnCC0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

备注 n = 0 到 5
m = 0, 1

(b) 比较匹配中断请求信号的产生时序 (INTTPnCC1)

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



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

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

备注 $n = 0$ 到 5

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

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

图 7-24. PWM 输出模式的配置图

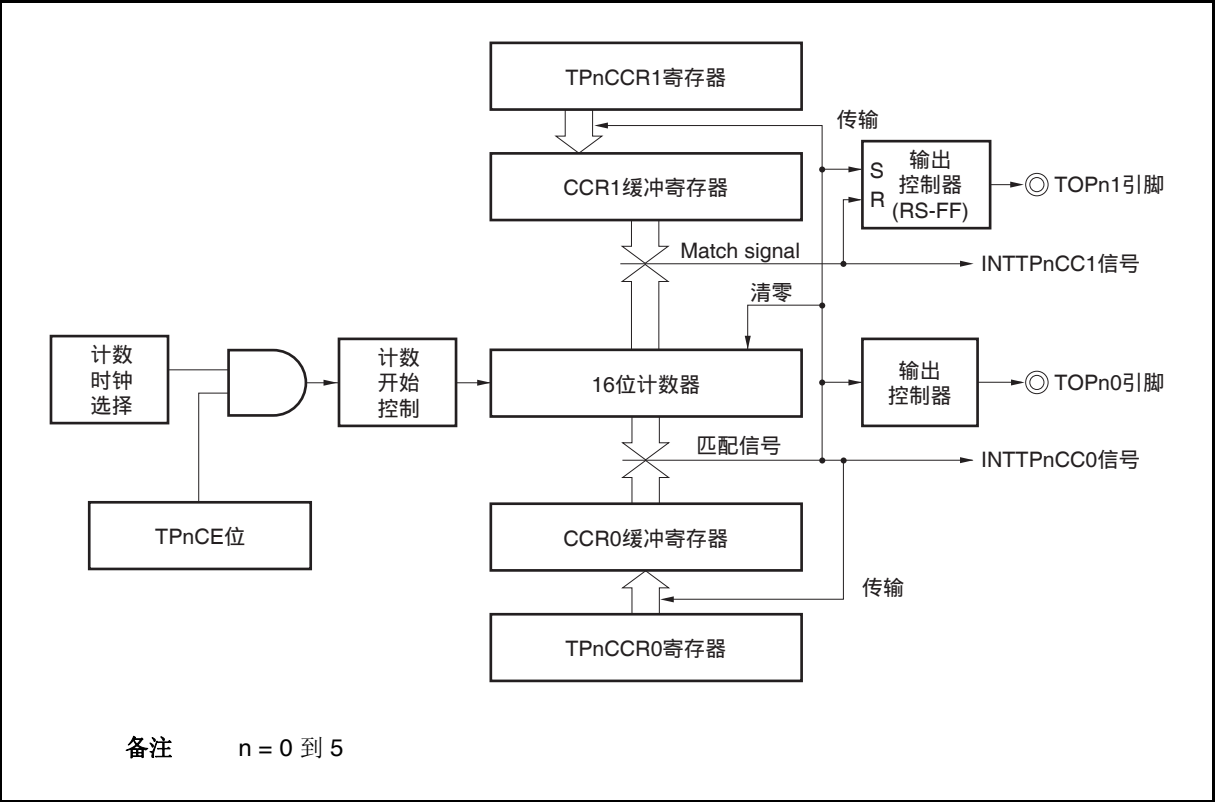
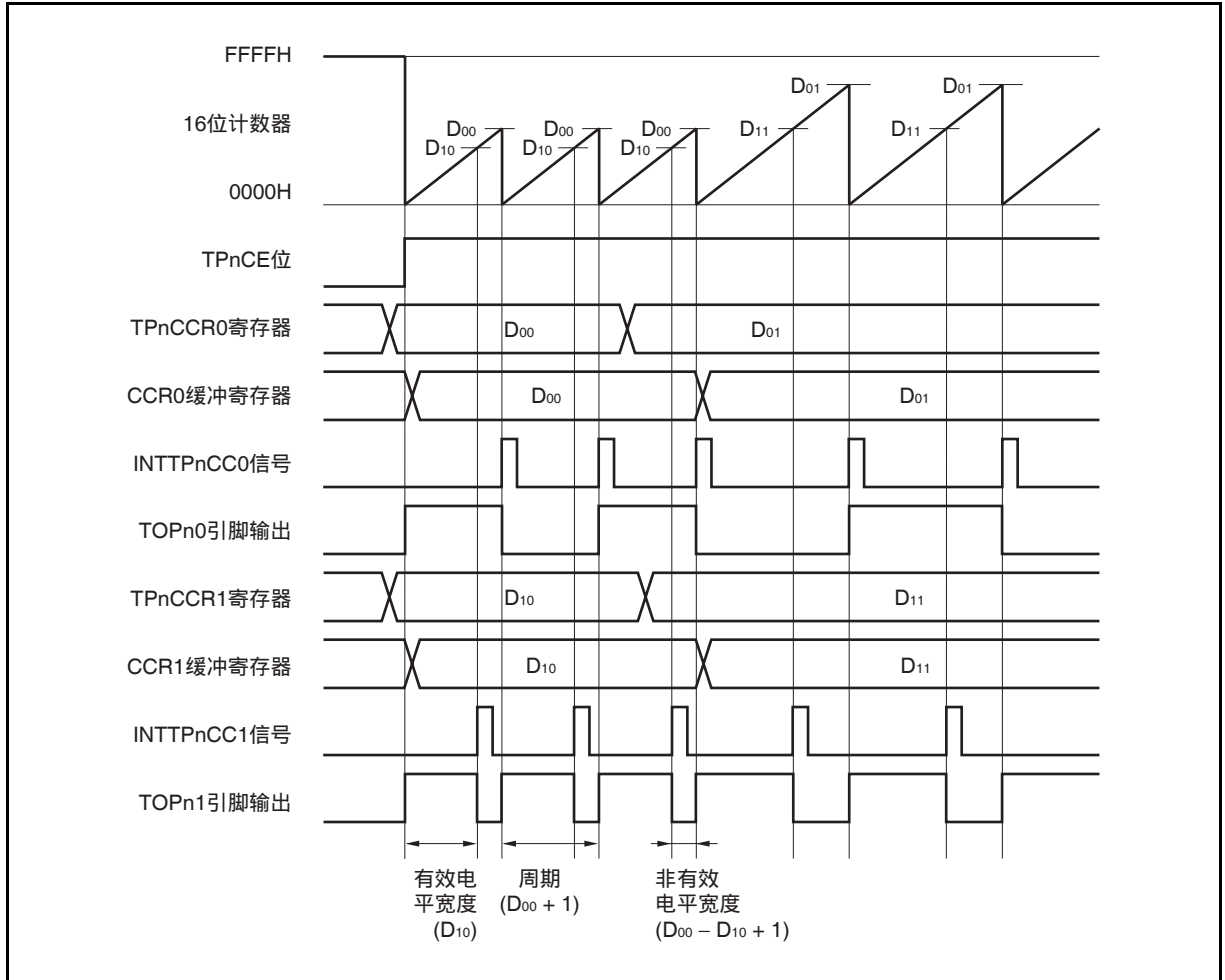


图 7-25. PWM 输出模式的基本时序



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

PWM 波形的有效电平宽度，周期以及占空比系数可由以下算式得出。

有效电平宽度 = (TPnCCR1 寄存器的值) × 计数时钟周期

周期 = (TPnCCR0 寄存器的值 + 1) × 计数时钟周期

占空比系数 = (TPnCCR1 寄存器的值)/(TPnCCR0 寄存器的值 + 1)

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

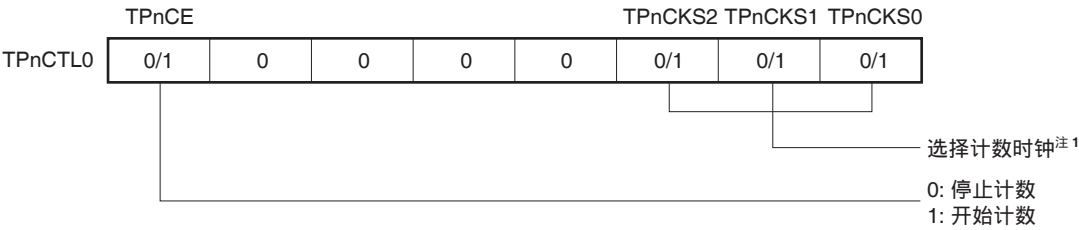
比较匹配中断请求信号 INTTPnCC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器的值相匹配后的下一个计数时钟时产生，同时 16 位计数器将被清零。比较匹配中断请求信号 INTTPnCC1 会在 16 位计数器的计数值与 CCR1 缓冲寄存器的值相匹配时产生。

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

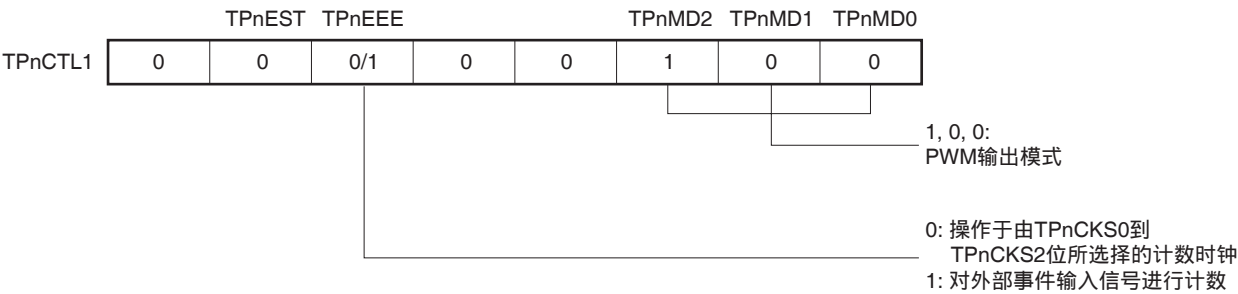
备注 n = 0 到 5, m = 0, 1

图 7-26. PWM 输出模式的寄存器设置 (1/2)

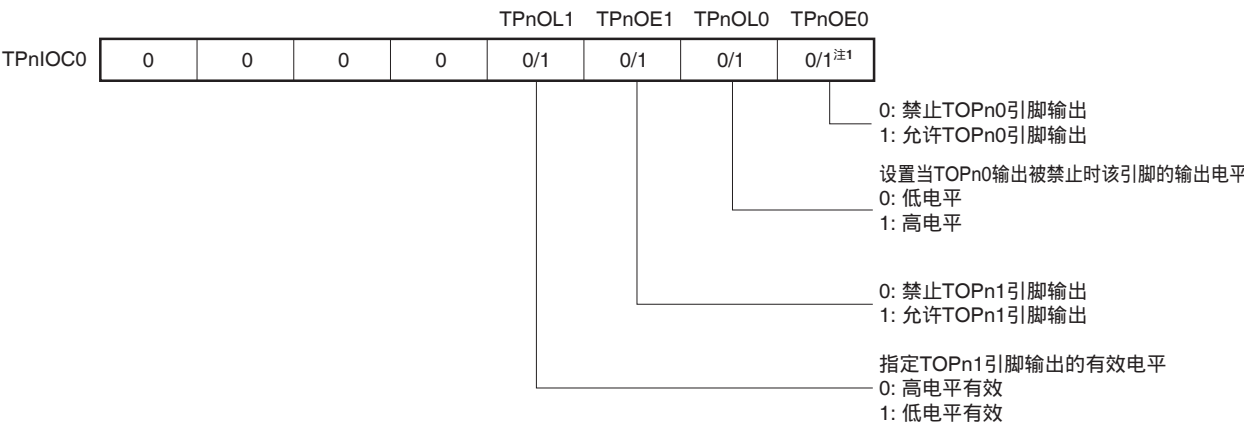
(a) TMPn 控制寄存器 0 (TPnCTL0)



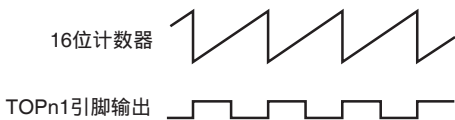
(b) TMPn 控制寄存器 1 (TPnCTL1)



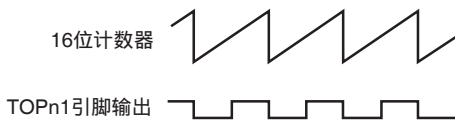
(c) TMPn I/O 控制寄存器 0 (TPnIOC0)



• 当TPnOL1位 = 0



• 当TPnOL1位 = 1



注 1. TPnCTL1.TPnEEE 位 = 1 时该设置无效。
2. 当 PWM 输出模式下不使用 TOPn0 引脚时，要将该位清零。

图 7-26. PWM 输出模式的寄存器设置 (2/2)



(1) PWM 输出模式的操作流程

图 7-27. PWM 输出模式的软件处理流程 (1/2)

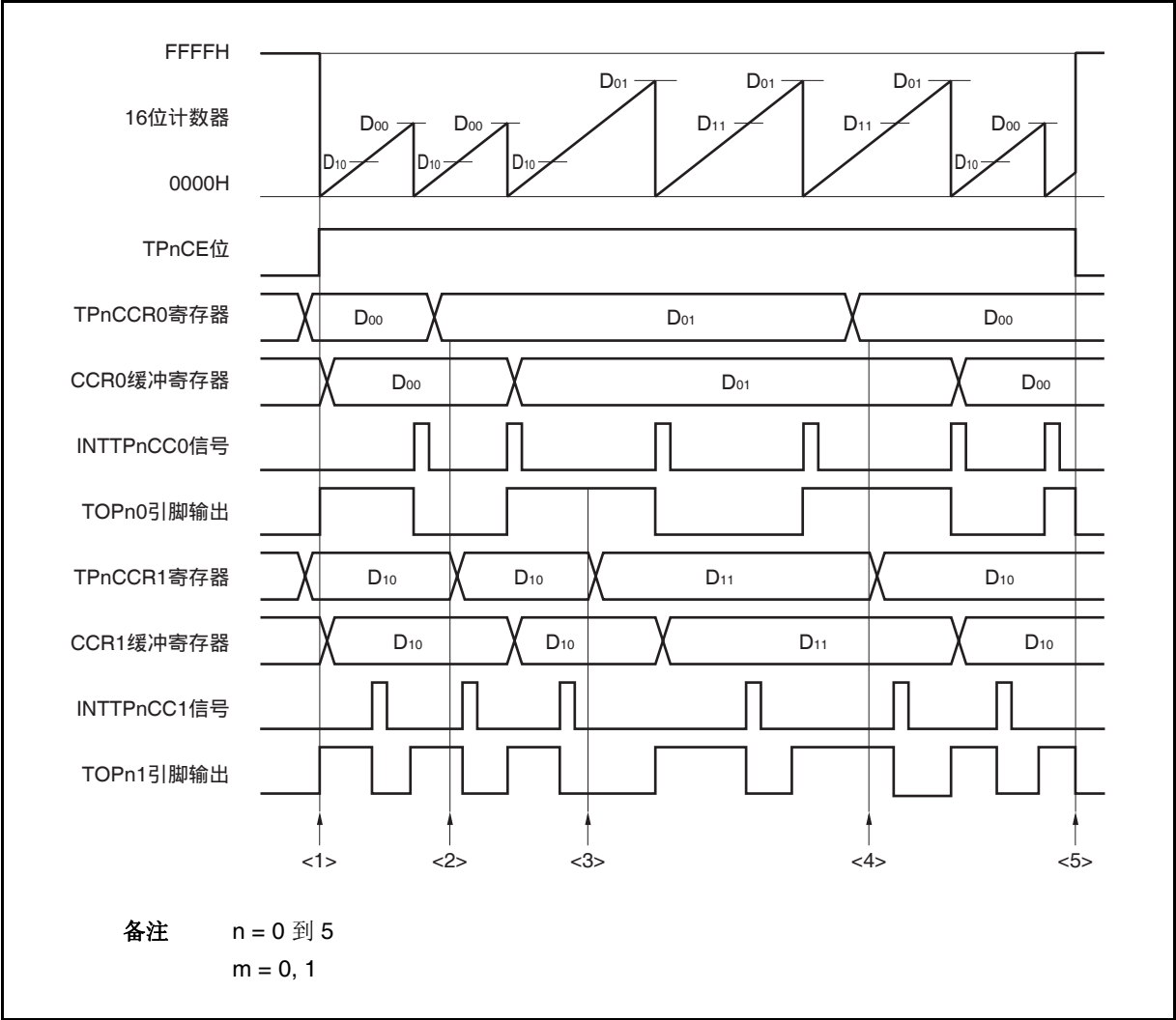
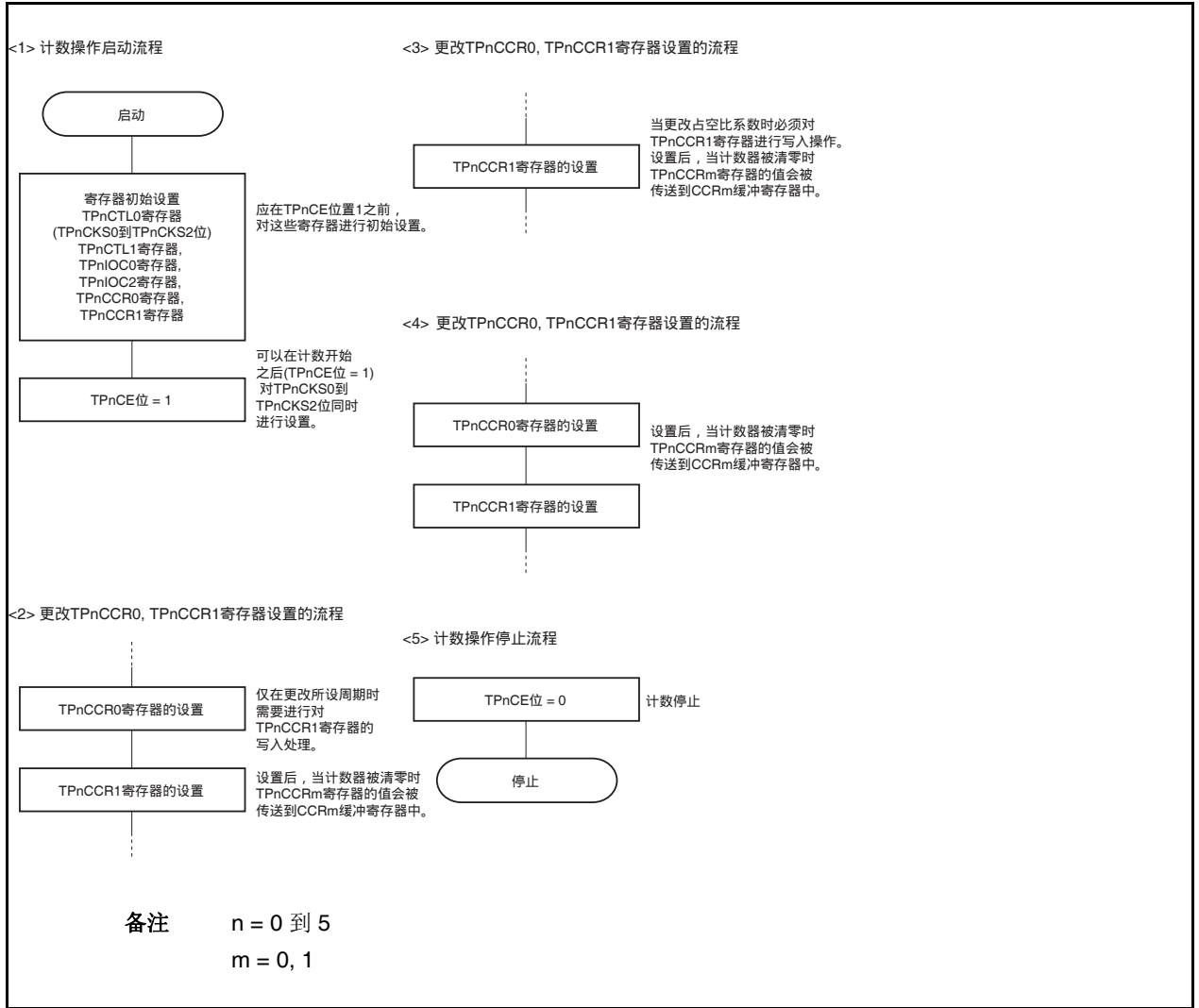


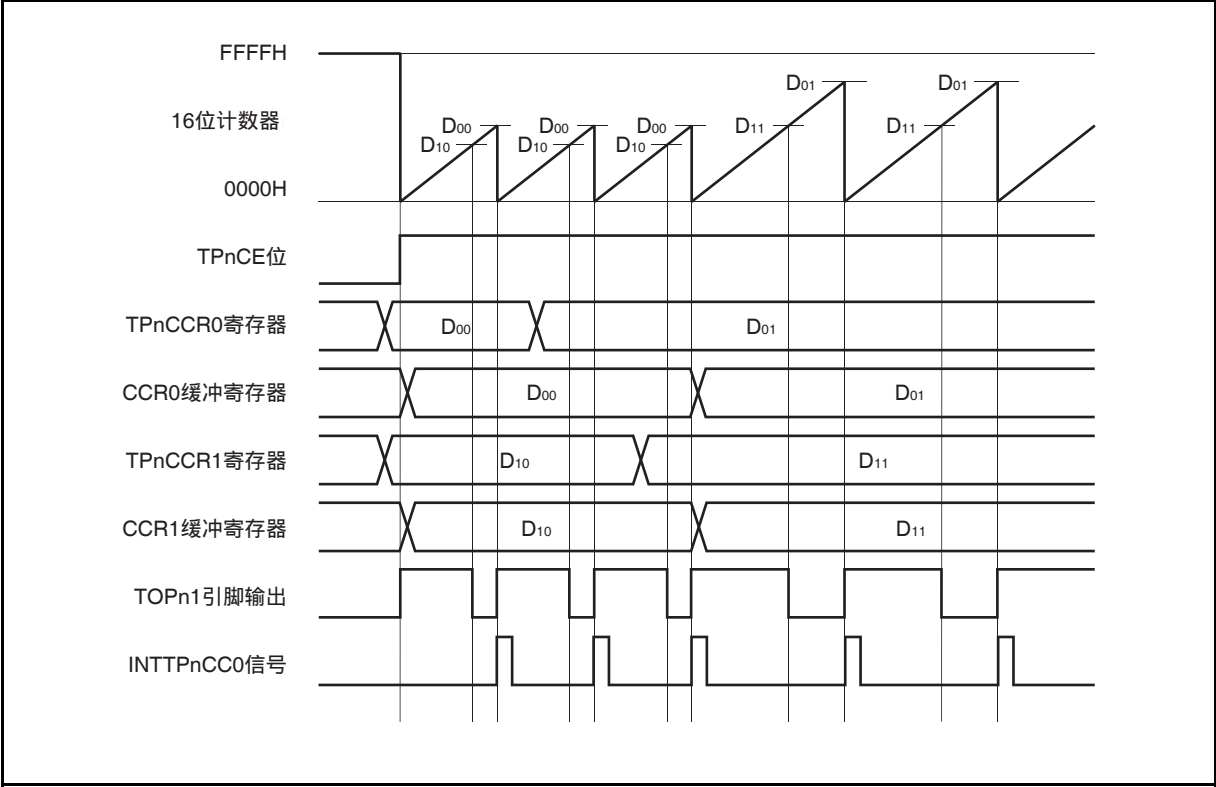
图 7-27. PWM 输出模式的软件处理流程 (2/2)



(2) PWM 输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

计数器计数过程中，若改变 PWM 波形，则要在最后进行对 TPnCCR1 寄存器的写入操作。
对 TPnCCR1 寄存器进行写入操作之后，若需再次更改 TPnCCRm 寄存器的值，须等待下一个 INTTPnCC0 信号被检测到后，再进行相关操作。



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

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

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

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

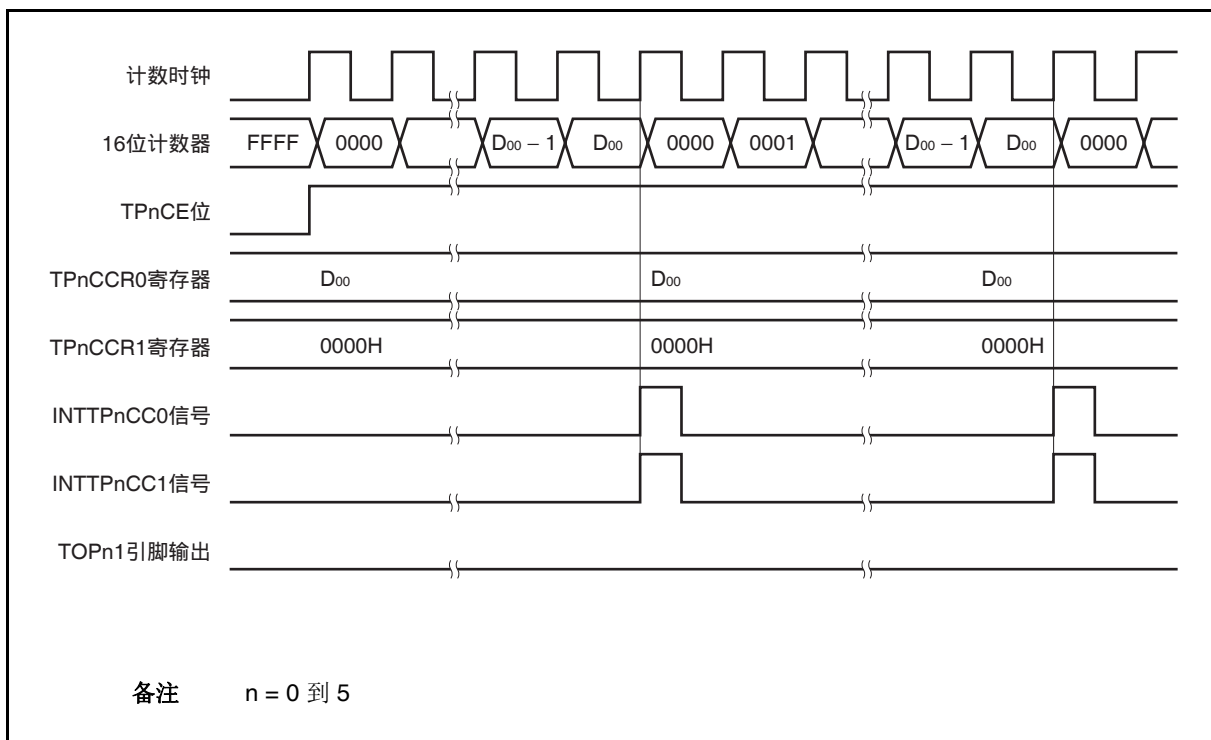
数据被写入 TPnCCR1 寄存器之后，16 位计数器被清零时，TPnCCRm 寄存器中的数值才会被同步传送到 CCRm 缓冲寄存器中，此时 CCRm 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

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

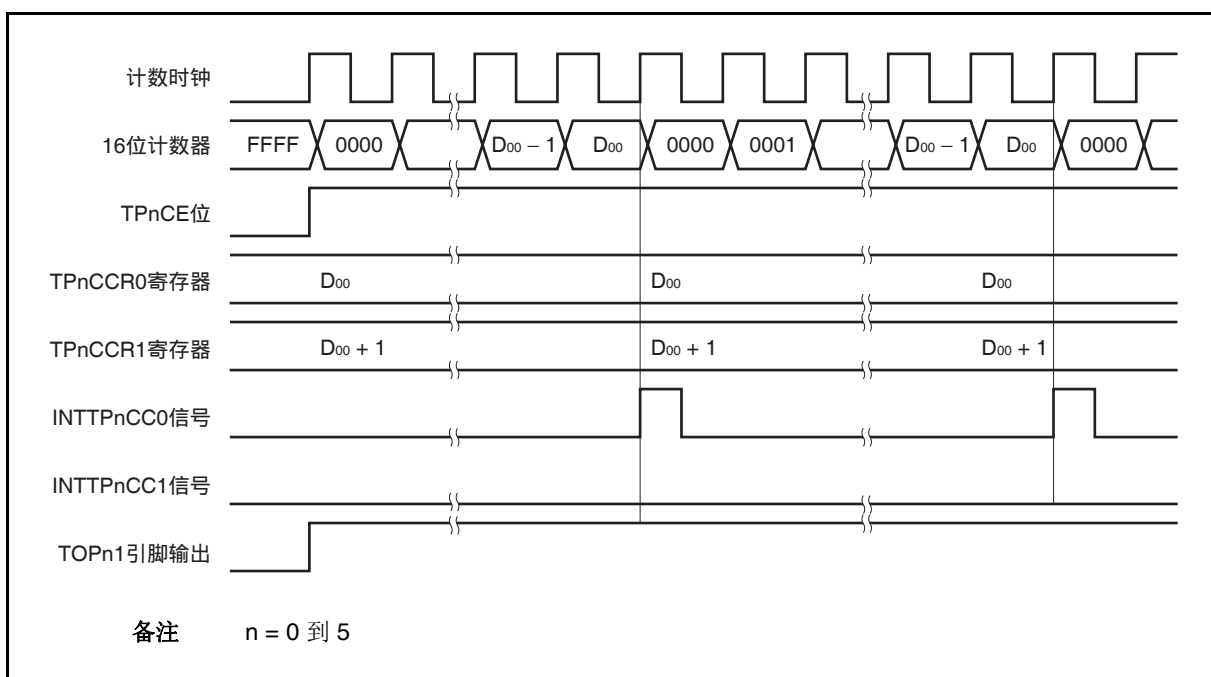
备注 n = 0 到 5, m = 0, 1

(b) PWM 波形的 0%/100% 输出

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

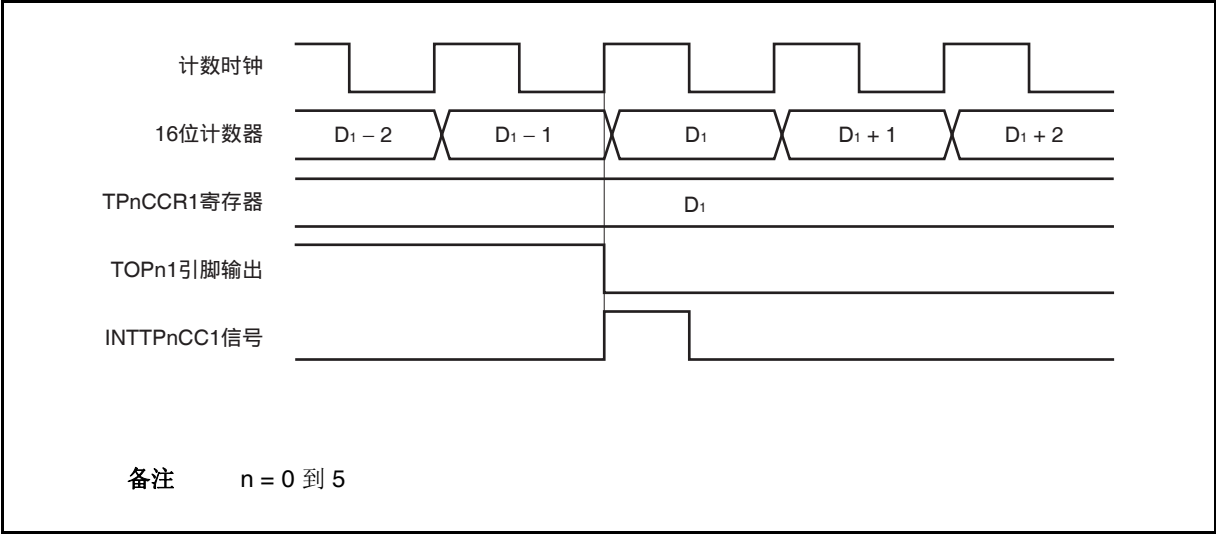


若要输出 100% 波形，则需将 TPnCCR1 寄存器设置为 (TPnCCR0 寄存器值 + 1)。TPnCCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。



(c) 比较匹配中断请求信号的产生时序 (INTTPnCC1)

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



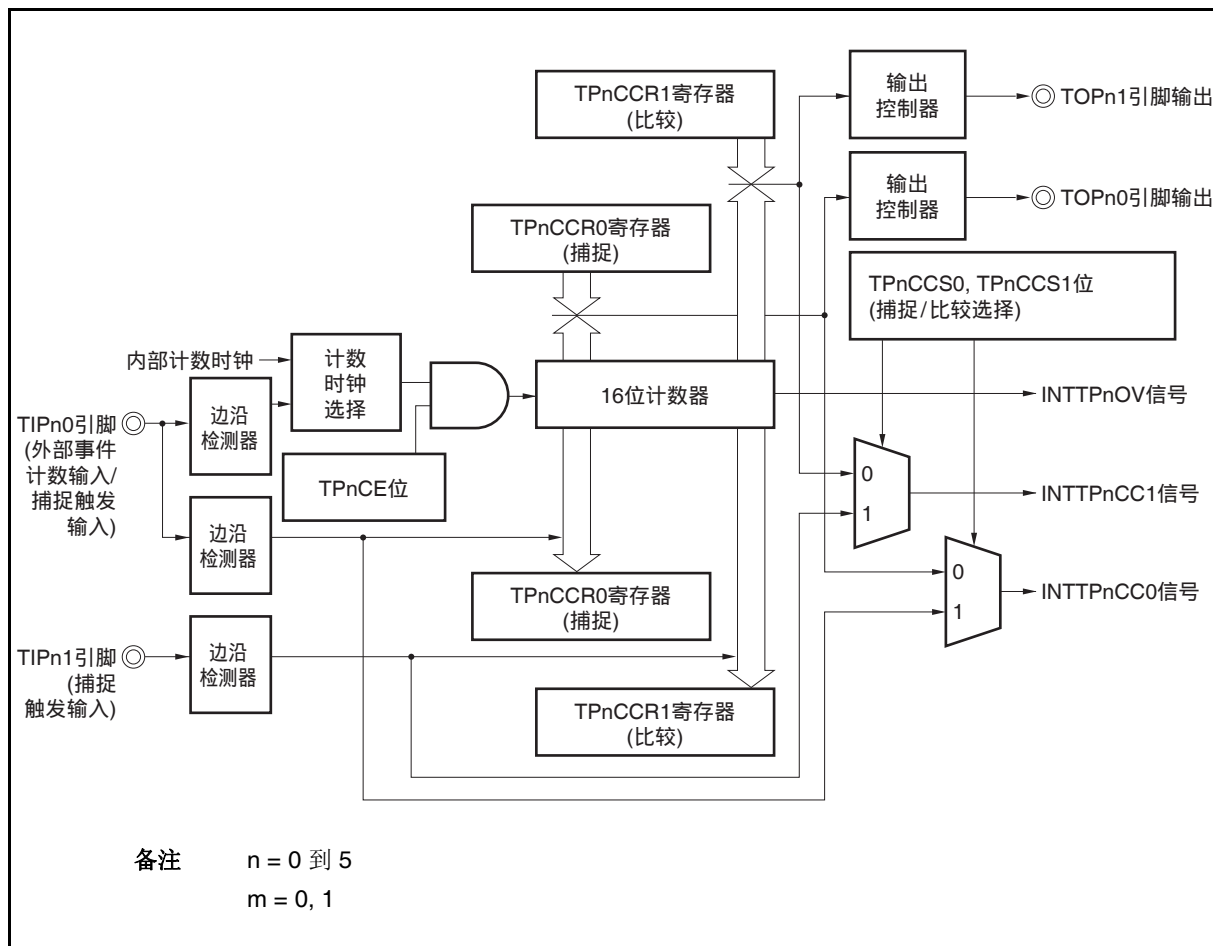
通常情况下，INTTPnCC1 信号会在 16 位计数器计数值与 TPnCCR1 寄存器的值匹配之后的下一个计数时钟同步产生。

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

7.5.6 自由运行定时器模式 (TPnMD2 到 TPnMD0 位 = 101)

在自由运行定时器模式中，TPnCTL0.TPnCE 位被置 1 后，16 位定时器/时间计数器 P 开始计数。此时，根据 TPnOPT0.TPnCCS0 和 TPnOPT0.TPnCCS1 位的设置，TPnCCRm 寄存器的功能可以在比较寄存器和捕捉寄存器之间进行选择。

图 7-28. 自由运行定时器模式的配置图

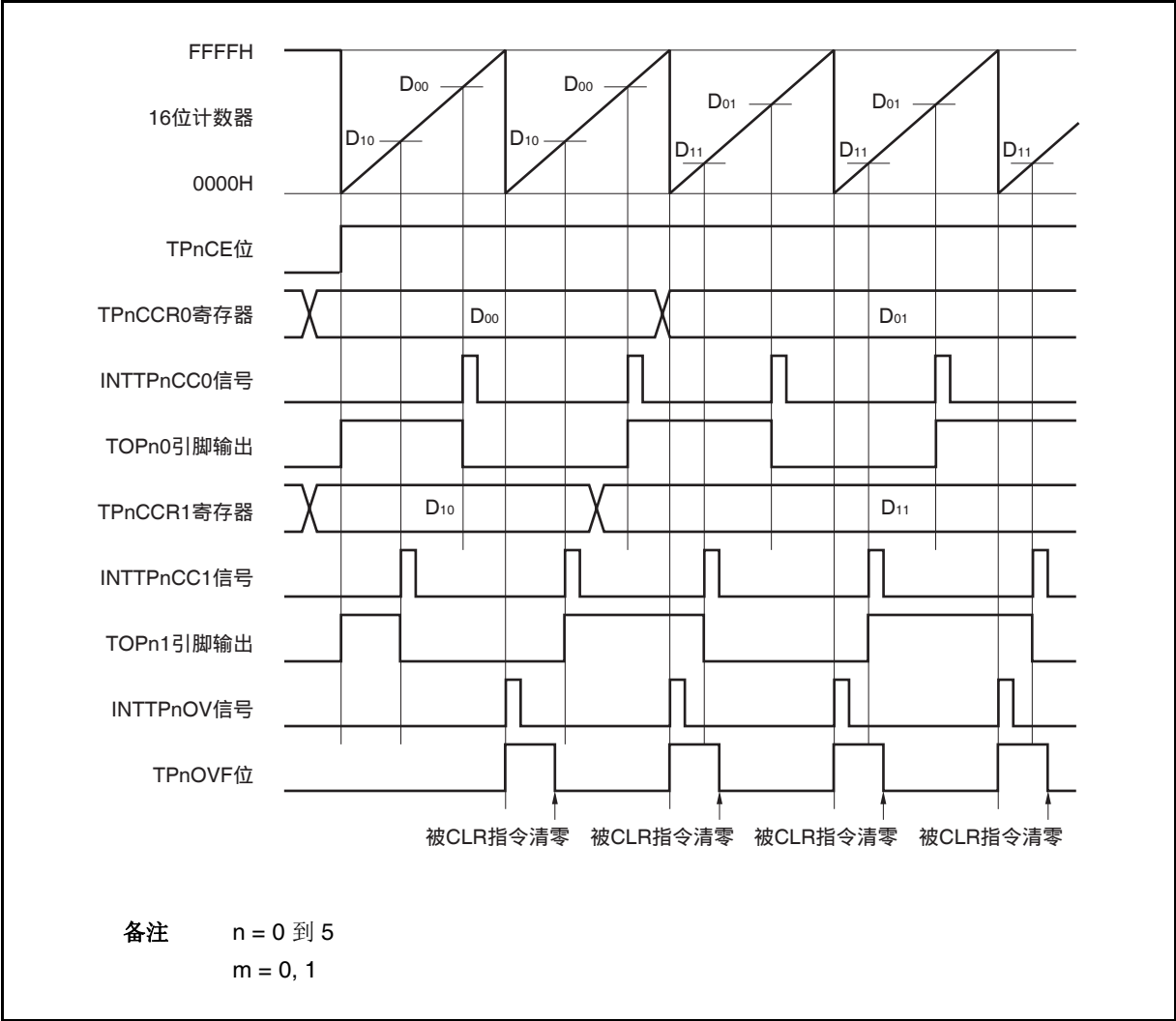


TPnCE 位被置 1 后，16 位计数器开始计数，同时 TOPn0 和 TOPn1 引脚输出信号的电平反转。当 16 位计数器的计数值与 TPnCCRm 寄存器的值匹配时将产生一个比较匹配中断请求信号(INTTPnCCm)，同时 TOPnm 引脚的输出电平将反转。

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

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

图 7-29. 自由运行定时器模式的基本时序 (比较功能)



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

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

图 7-30. 自由运行定时器模式的基本时序 (捕捉功能)

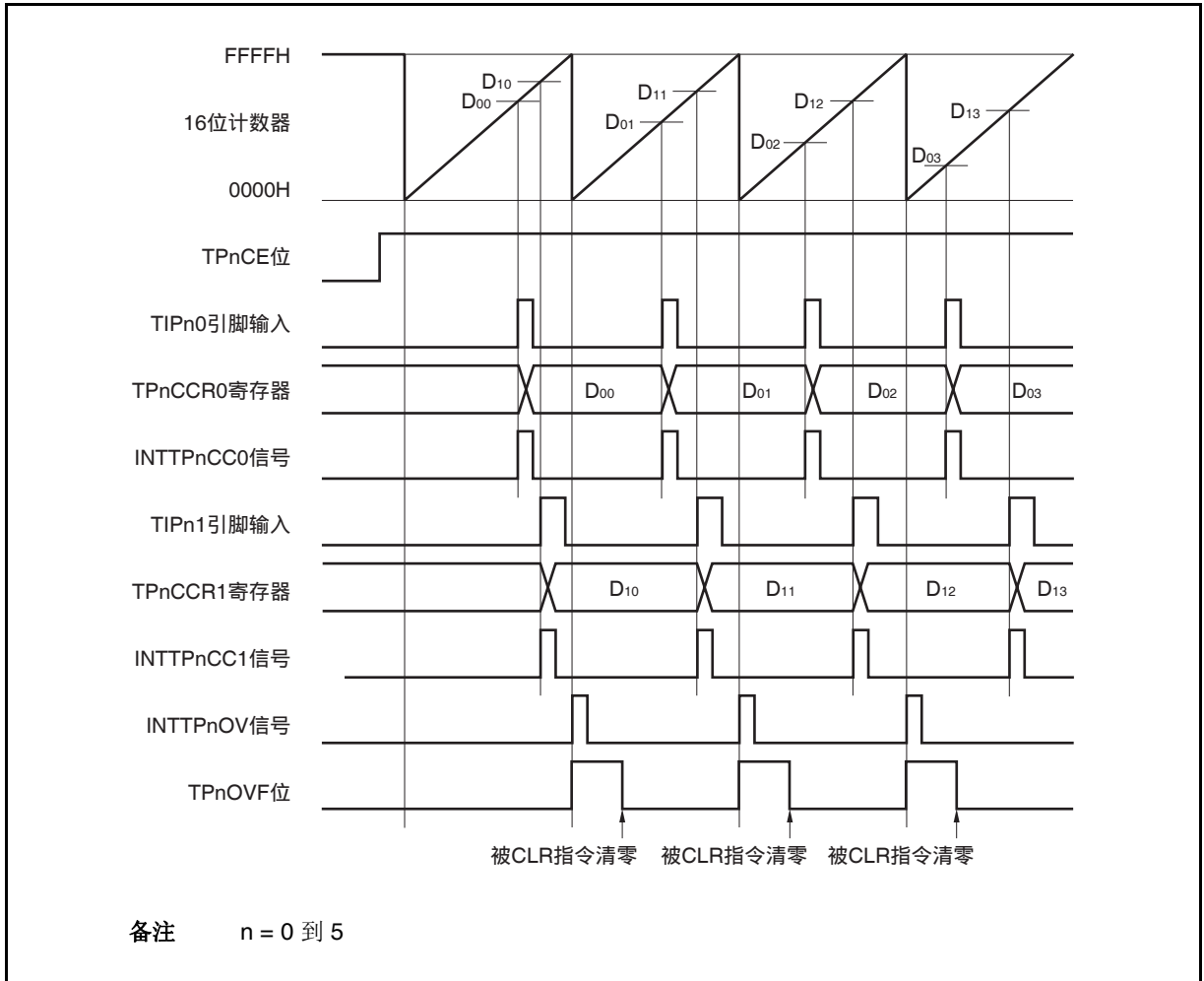
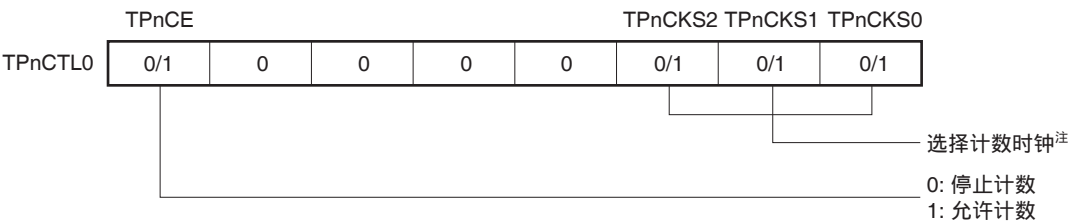


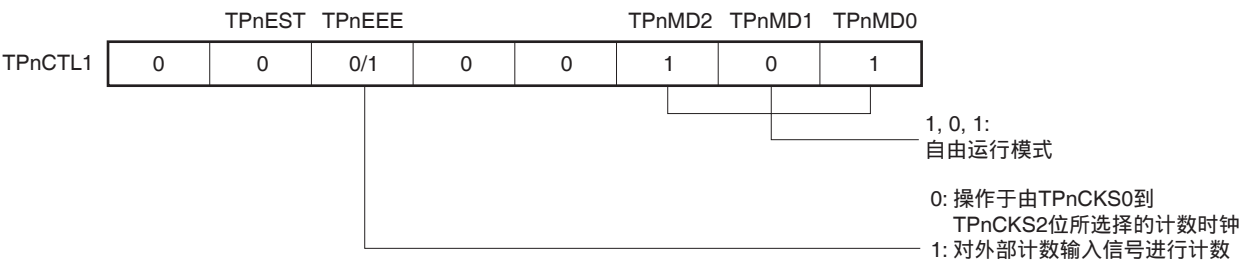
图 7-31. 自由运行定时器模式的寄存器设置 (1/2)

(a) TMPn 控制寄存器 0 (TPnCTL0)



注 TPnCTL1.TPnEEE 位 = 1 时该设置无效。

(b) TMPn 控制寄存器 1 (TPnCTL1)



(c) TMPn I/O 控制寄存器 0 (TPnIOC0)

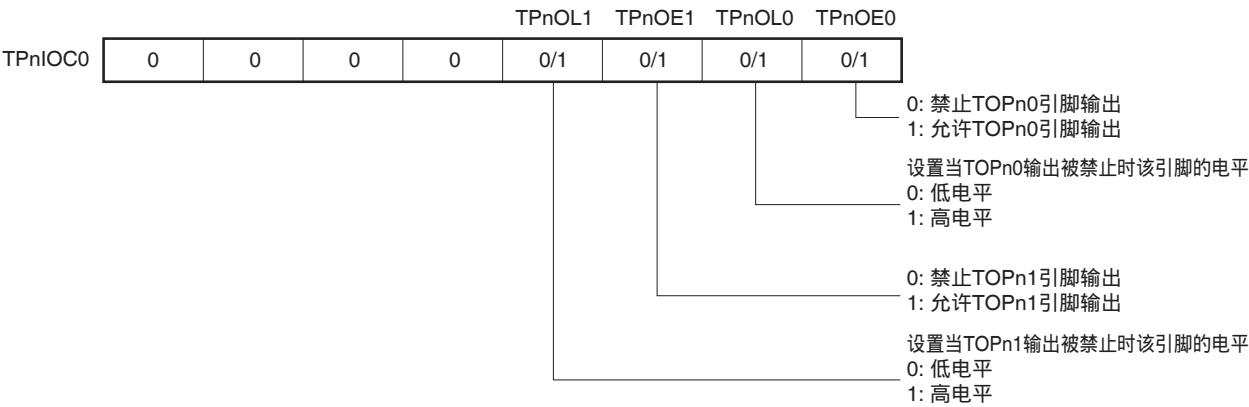
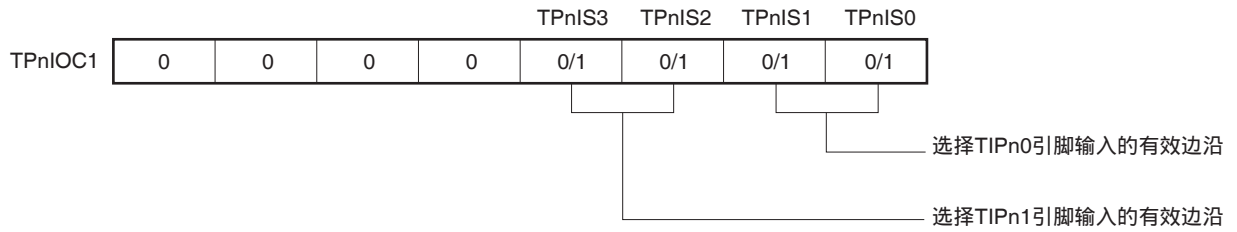
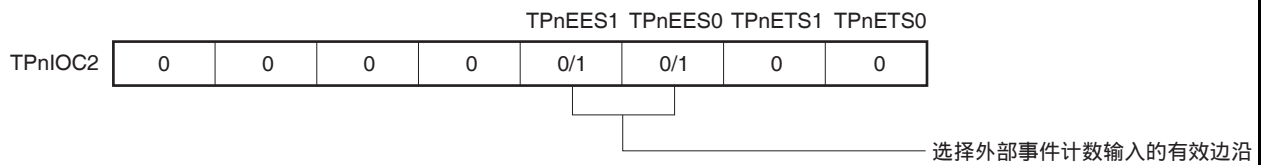


图 7-31. 自由运行定时器模式的寄存器设置 (2/2)

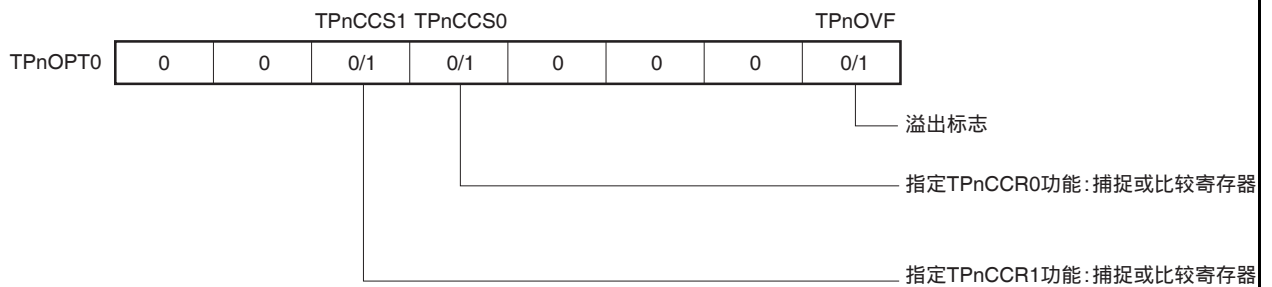
(d) TMPn I/O 控制寄存器 1 (TPnIOC1)



(e) TMPn I/O 控制寄存器 2 (TPnIOC2)



(f) TMPn 选项寄存器 0 (TPnOPT0)



(g) TMPn 计数器读取缓冲寄存器 (TPnCNT)

可以通过读取 TPnCNT 寄存器来读取 16 位计数器的计数值。

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

根据 TPnOPT0.TPnCCSm 位的设置, TPnCCRm 寄存器的功能可以在比较寄存器和捕捉寄存器之间进行选择。

当选择了捕捉寄存器功能时, 若检测到 TIPnm 引脚输入信号的有效沿, 则 16 位计数器此刻的计数值会被保存到 TPnCCRm 寄存器中。

当选择了比较寄存器功能且 TPnCCRm 寄存器的值被设置为 D_m 时, INTTPnCCm 信号将在计数器计数到 $(D_m + 1)$ 时产生, 同时 TOPnm 引脚的输出电平反转。

备注 n = 0 到 5
 m = 0, 1

- (1) 自由运行定时器模式的操作流程
- (a) 当捕捉/比较寄存器用作比较寄存器时

图 7-32. 自由运行定时器模式的软件处理流程(比较功能) (1/2)

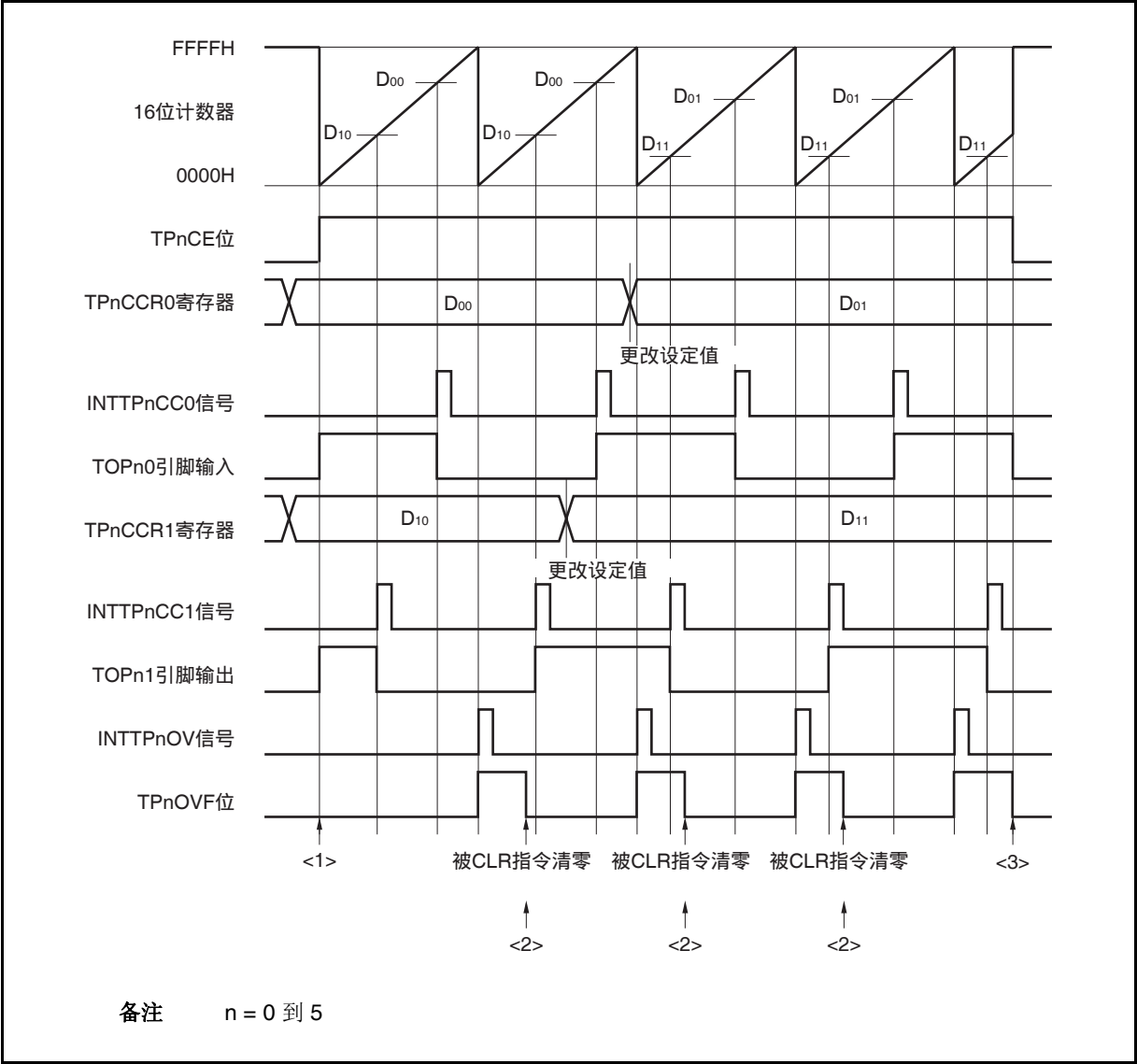
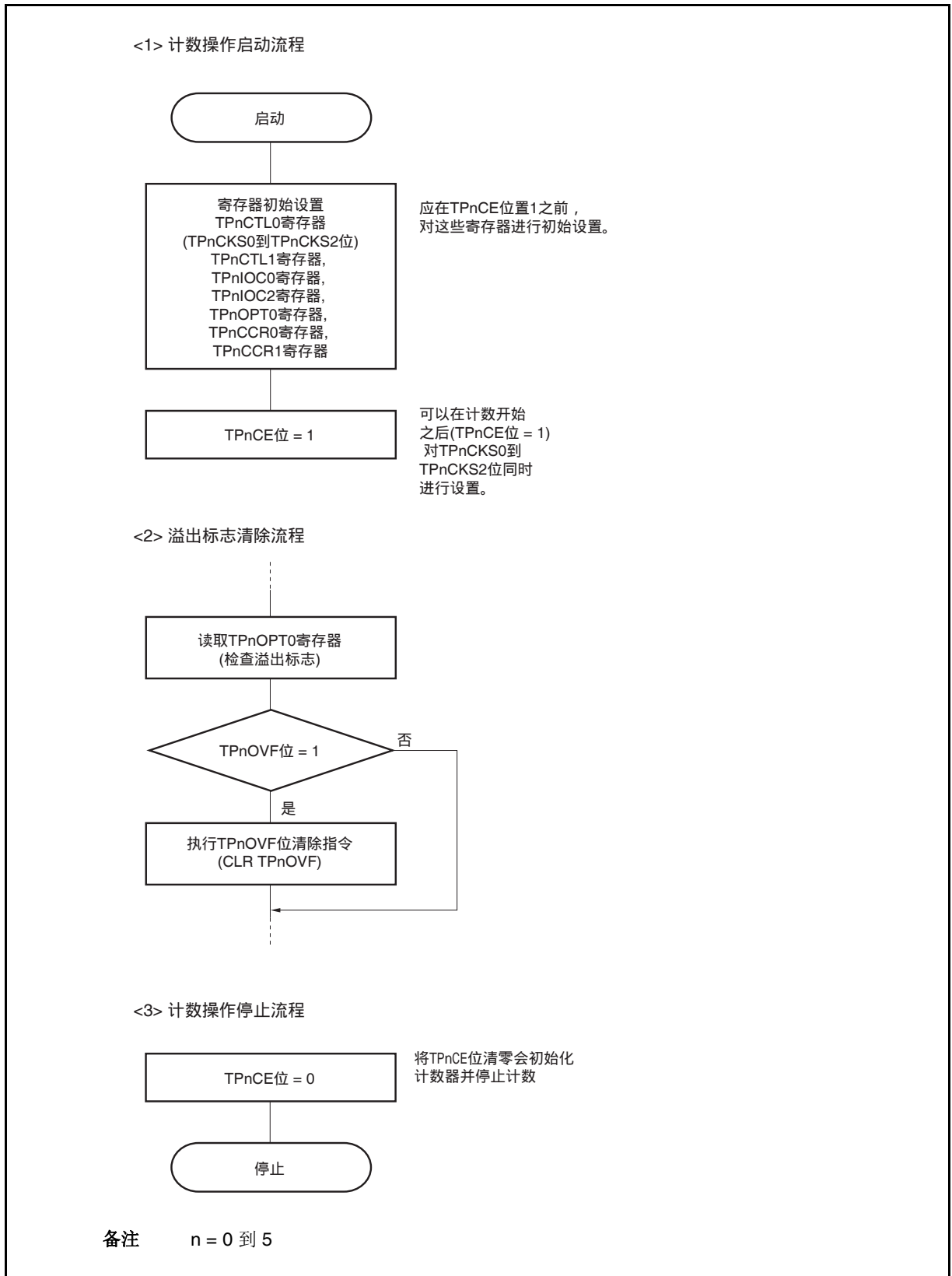


图 7-32. 自由运行定时器模式的软件处理流程 (比较功能) (2/2)



(b) 当捕捉/比较寄存器用作捕捉寄存器时

图 7-33. 自由运行定时器模式的软件处理流程 (捕捉功能) (1/2)

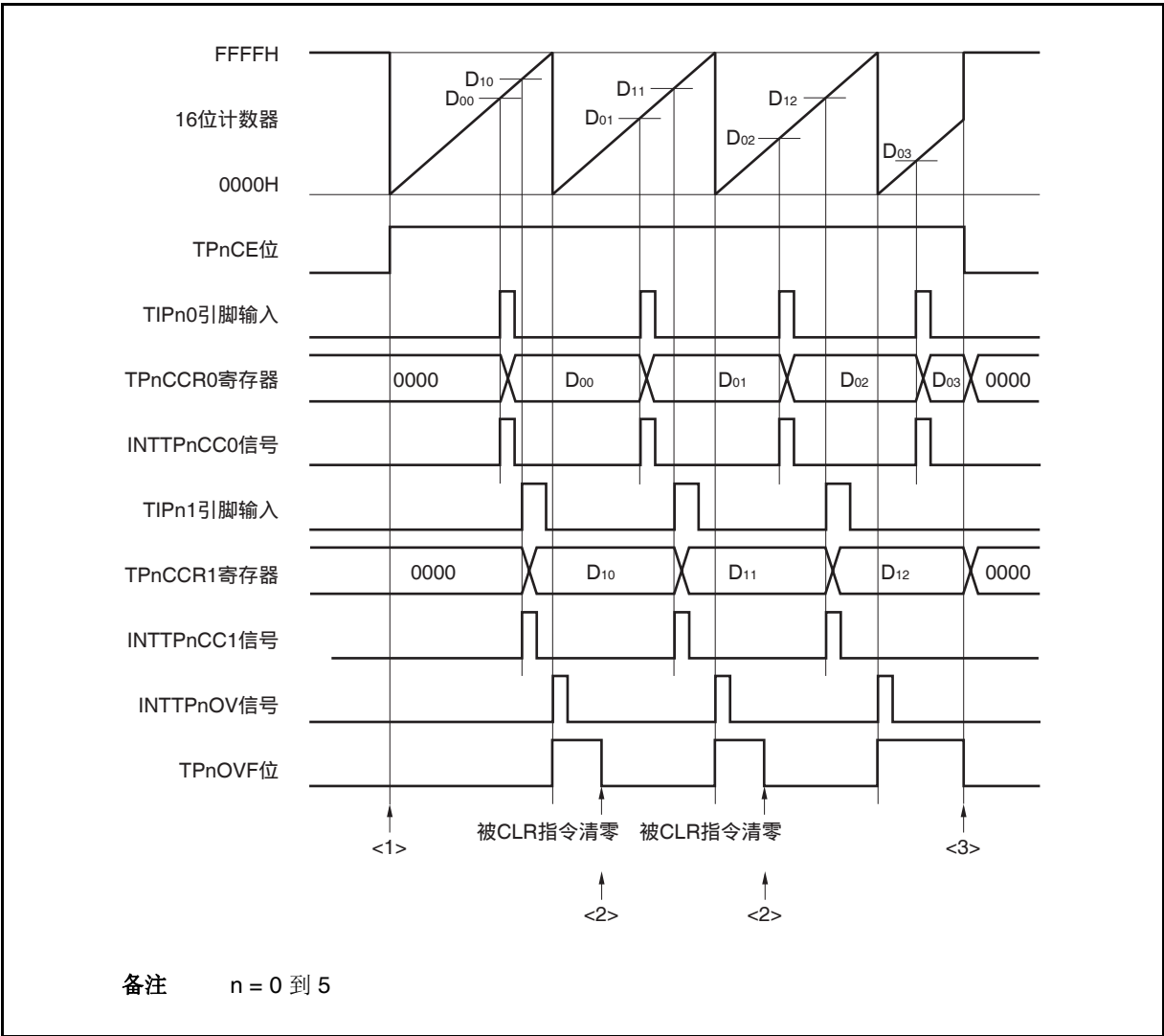
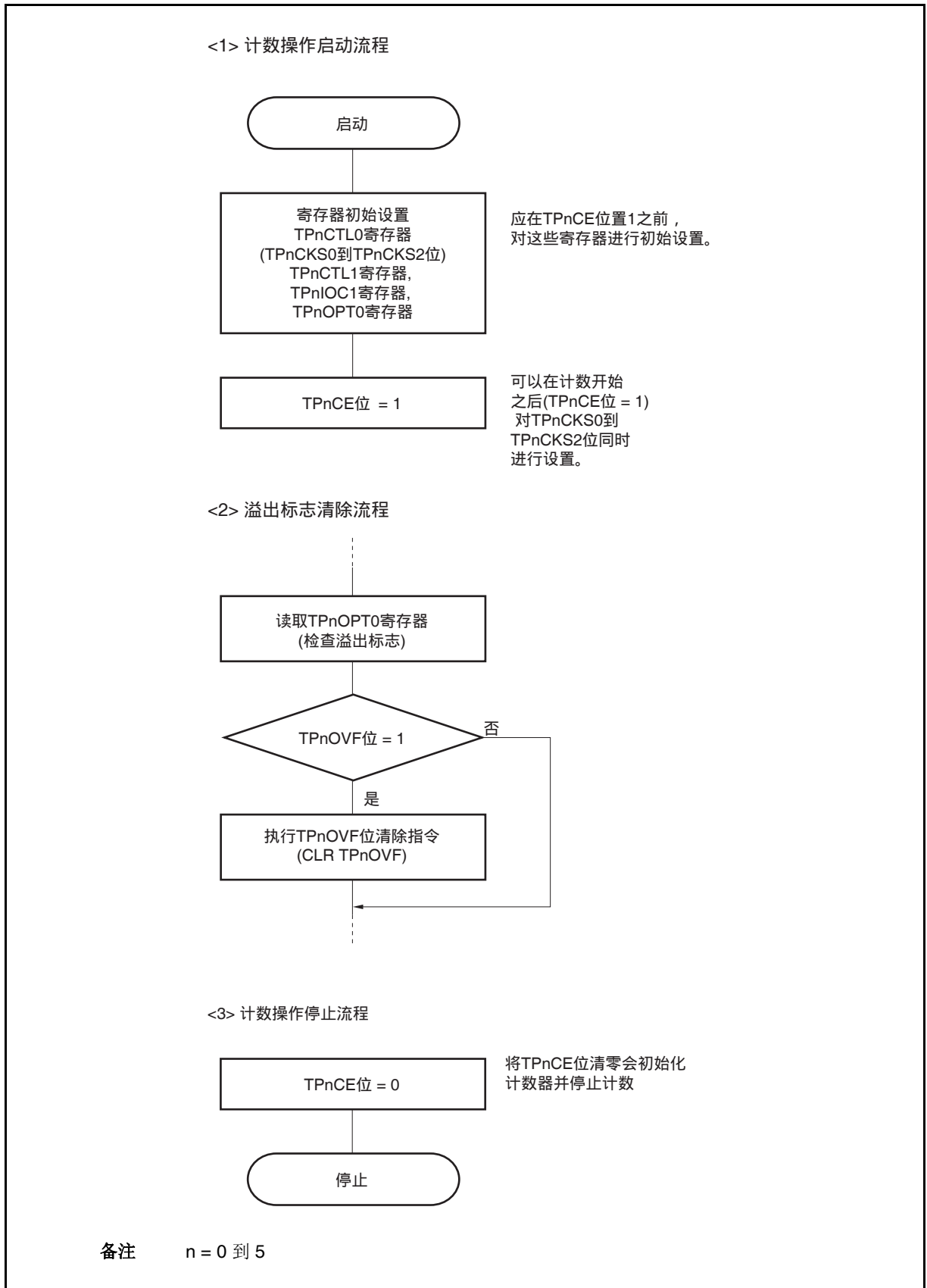


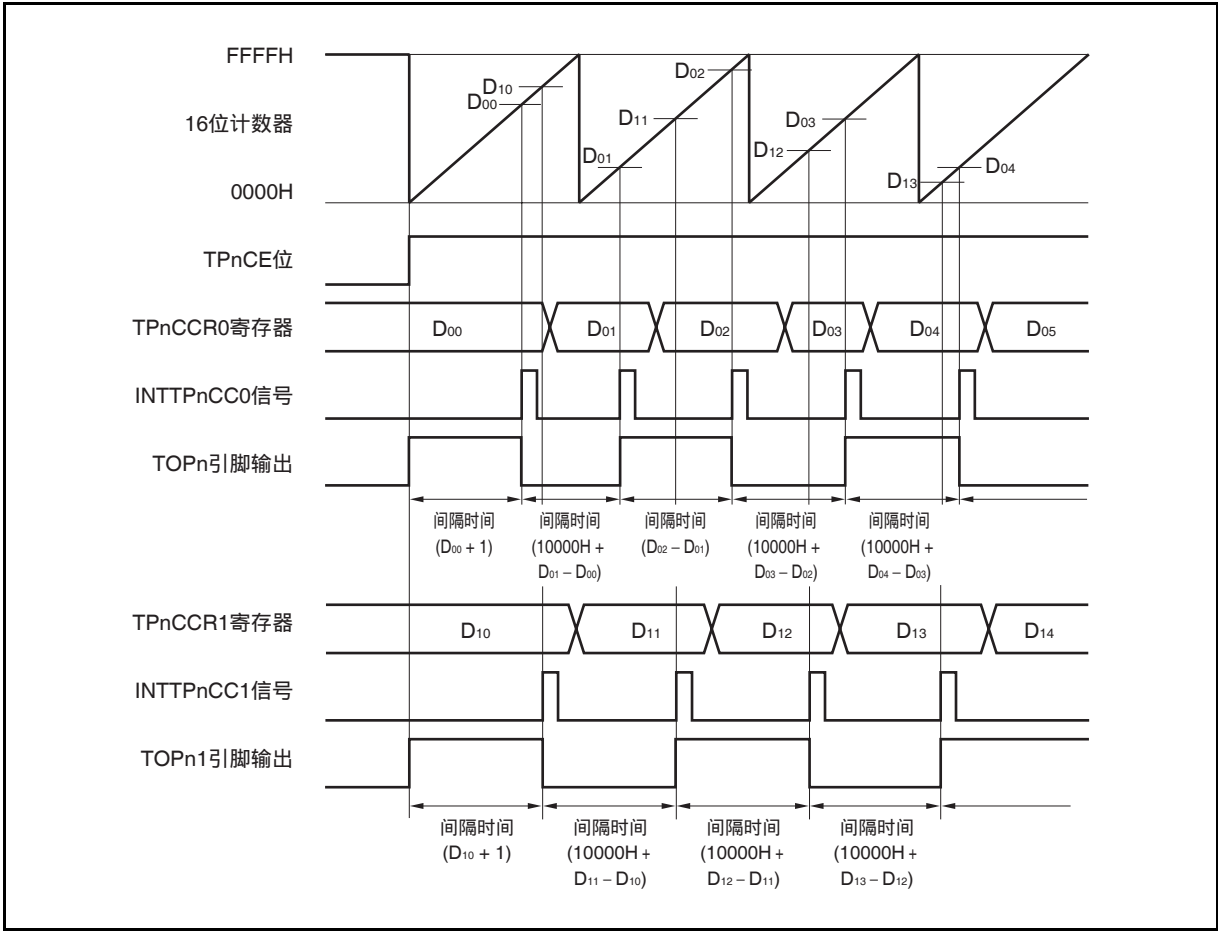
图 7-33. 自由运行定时器模式的软件处理流程 (捕捉功能) (2/2)



(2) 自由运行定时器模式的操作时序

(a) 使用比较寄存器功能的时间间隔操作

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



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

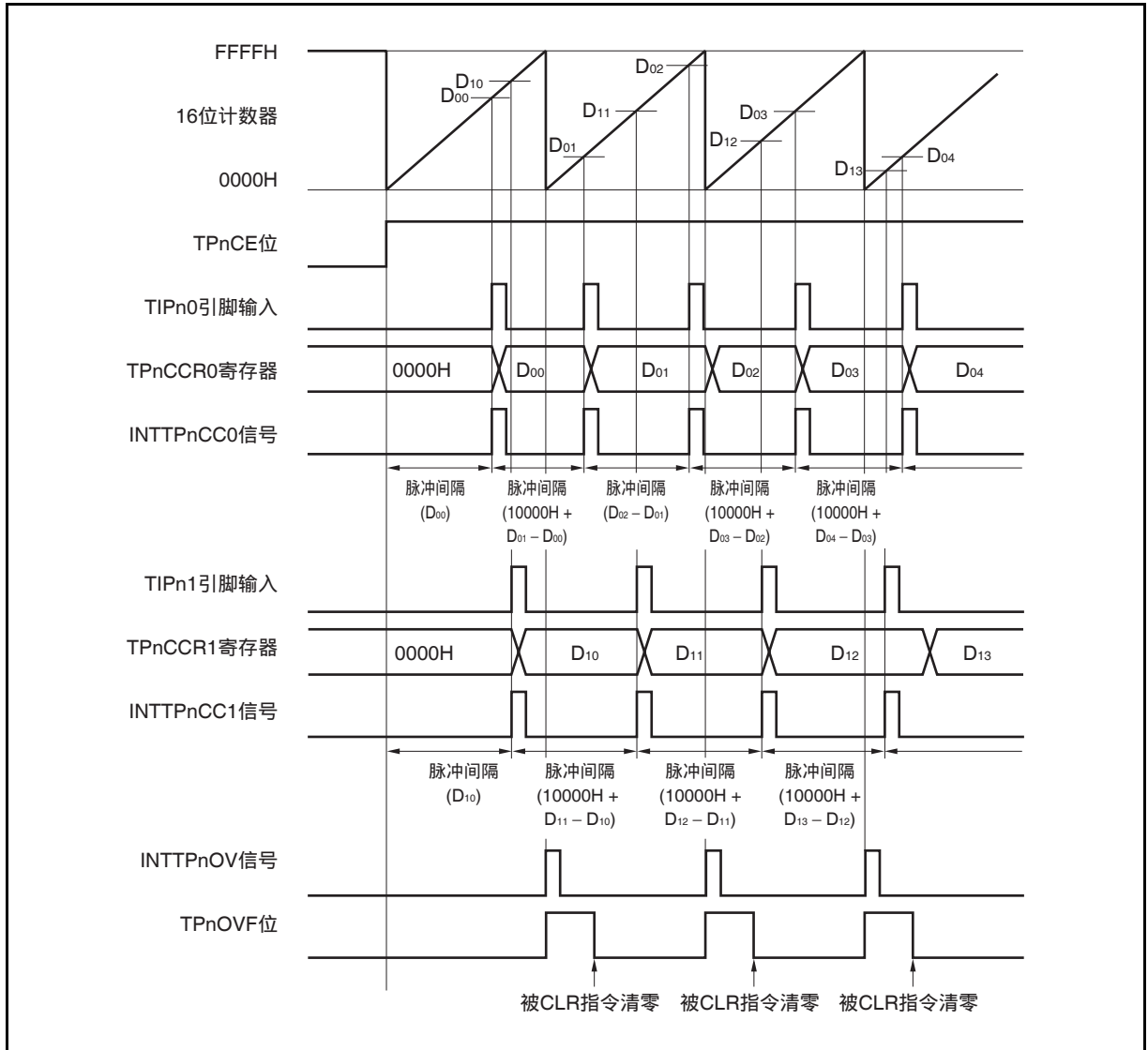
向 TPnCCRm 寄存器中重新设置的值可由以下算式求出，这里假设间隔时间为“Dm”。

- 比较寄存器的最初设置值: $D_m - 1$
- 第二次设置比较寄存器以后(含第二次)的设置值: 前次设置值 + D_m
- (如果计算结果大于 FFFFH，那么将结果减 10000H 后再设置到寄存器中。)

备注 n = 0 到 5
 m = 0, 1

(b) 使用捕捉寄存器进行脉宽测量

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



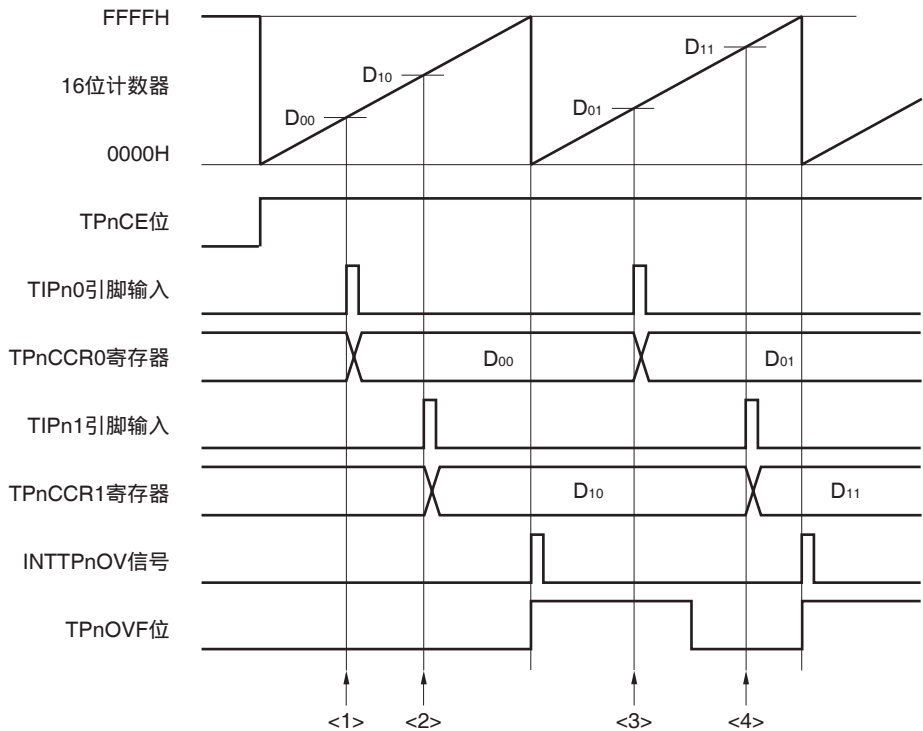
当使用自由运行定时器模式进行脉宽测量操作时，可以在同一通道对两个不同的脉冲宽度进行测量。通过与 INTTPnCCm 信号同步读取 TPnCCRm 寄存器的值并计算该值与上一次读取值的差，就可以求得脉冲的宽度。

备注 n = 0 到 5
 m = 0, 1

(c) 使用两个捕捉寄存器产生溢出时的操作

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

使用两个捕捉寄存器时的错误处理举例



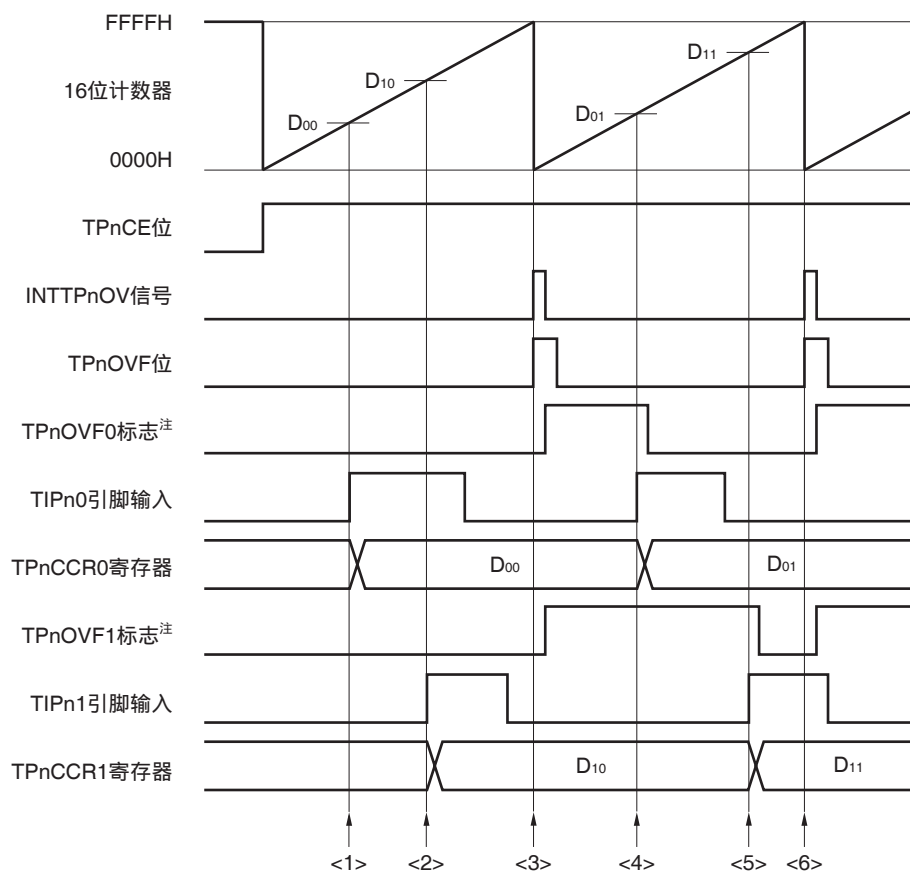
使用自由运行定时器模式对两个脉冲宽度进行测量时，可能出现以下的问题。

- <1> 读取 TPnCCR0 寄存器 (设置为 TIPn0 引脚输入的初始值)。
- <2> 读取 TPnCCR1 寄存器 (设置为 TIPn1 引脚输入的初始值)。
- <3> 读取 TPnCCR0 寄存器。
读取溢出标志。如果溢出标志为 1，那么将其清零。
由于此时溢出标志位为 1，所以脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。
- <4> 读取 TPnCCR1 寄存器
读取溢出标志。由于标志位在步骤<3>中被清零，所以此时读取的值为 0。
由于溢出标志位为 0，所以脉冲宽度按照 $(D_{11} - D_{10})$ (错误)来计算。

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

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

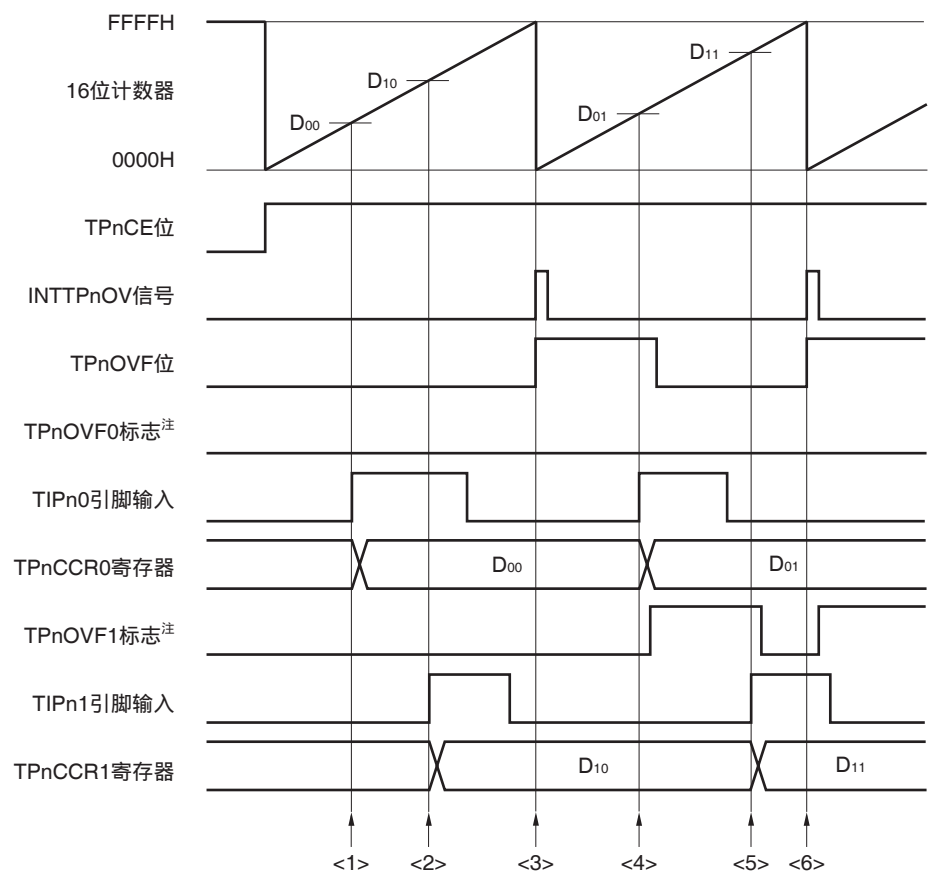
使用两个捕捉寄存器时的处理举例(利用溢出中断)



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

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

使用两个捕捉寄存器时的处理举例(不利用溢出中断)



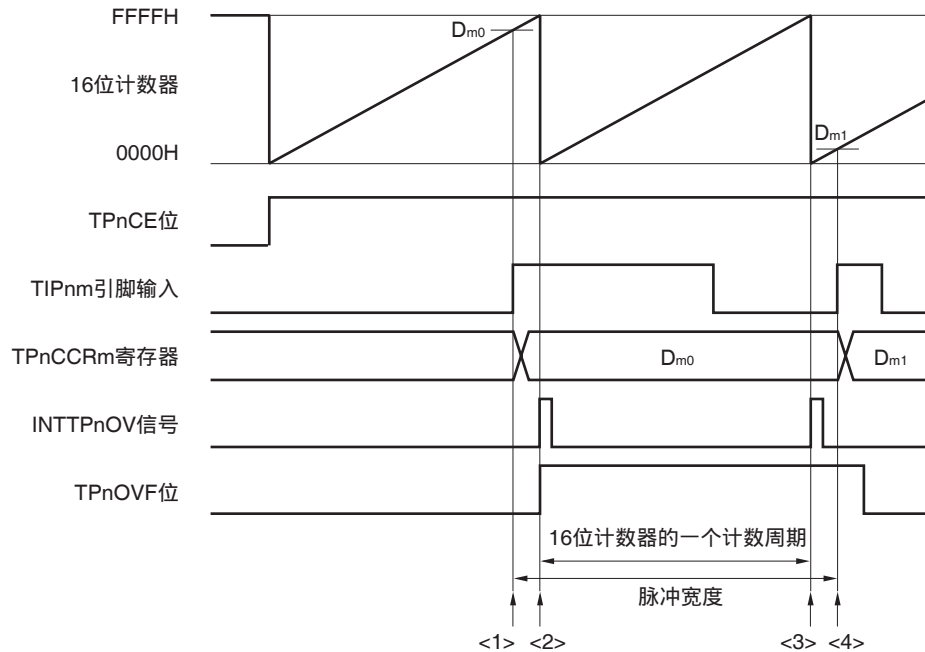
注 这里的 TPnOVF0 和 TPnOVF1 标志是建立于内置 RAM 上的变量，可由软件任意进行设置。

- <1> 读取 TPnCCR0 寄存器 (设置为 TIPn0 引脚输入的初始值)。
- <2> 读取 TPnCCR1 寄存器 (设置为 TIPn1 引脚输入的初始值)。
- <3> 产生了溢出。但不进行任何软件处理。
- <4> 读取 TPnCCR0 寄存器。
读取溢出标志。如果溢出标志为 1，那么只把 TPnOVF1 标志设置为 1，同时将溢出标志清零。
由于溢出标志位为 1，脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。
- <5> 读取 TPnCCR1 寄存器。
读取溢出标志。由于溢出标志位在步骤<4>中被清零，所以读取结果为 0。
读取 TPnOVF1 位。如果 TPnOVF1 标志为 1，那么将其清零。
由于 TPnOVF1 标志为 1，脉冲宽度按照 $(10000H + D_{11} - D_{10})$ (正确)来计算。
- <6> 同步步骤 <3>

(d) 捕捉触发间隔较长时的溢出处理

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

捕捉触发间隔较长时的错误处理举例



使用自由运行定时器模式对较长的脉冲宽度进行测量时，可能出现以下的问题。

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

<2> 产生了溢出。但不进行任何软件处理。

<3> 产生第二次溢出。但不进行任何软件处理。

<4> 读取 TPnCCRM 寄存器。

读取溢出标志位。如果溢出标志为 1，那么将其清零。

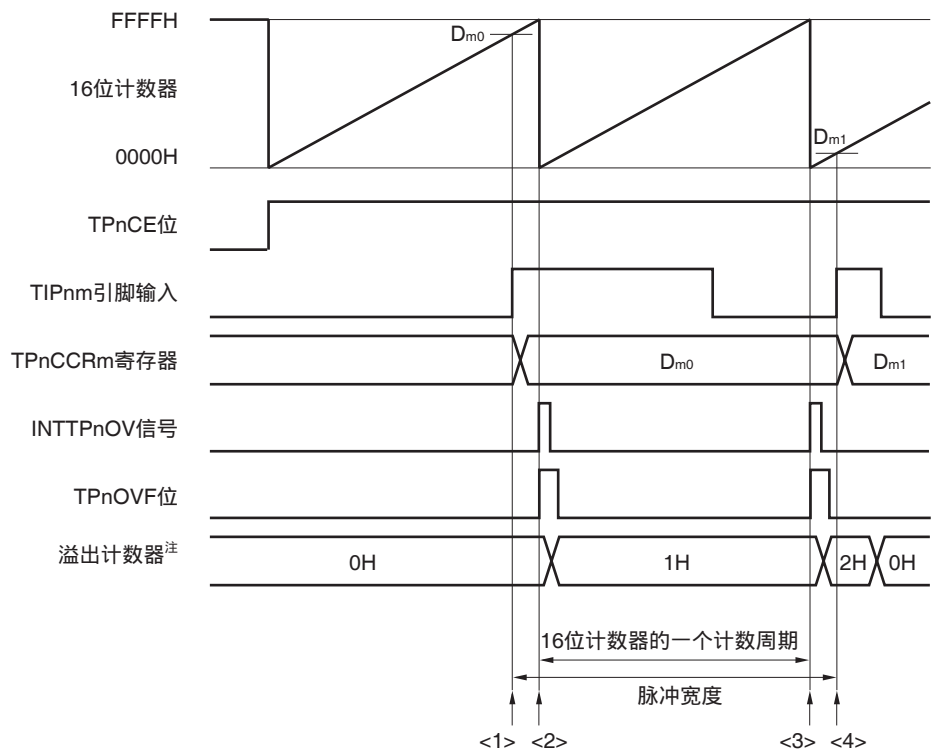
由于溢出标志为 1，脉冲宽度按照 $(10000H + D_{m1} - D_{m0})$ (错误) 来计算。

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

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

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

捕捉触发间隔较长时的处理举例

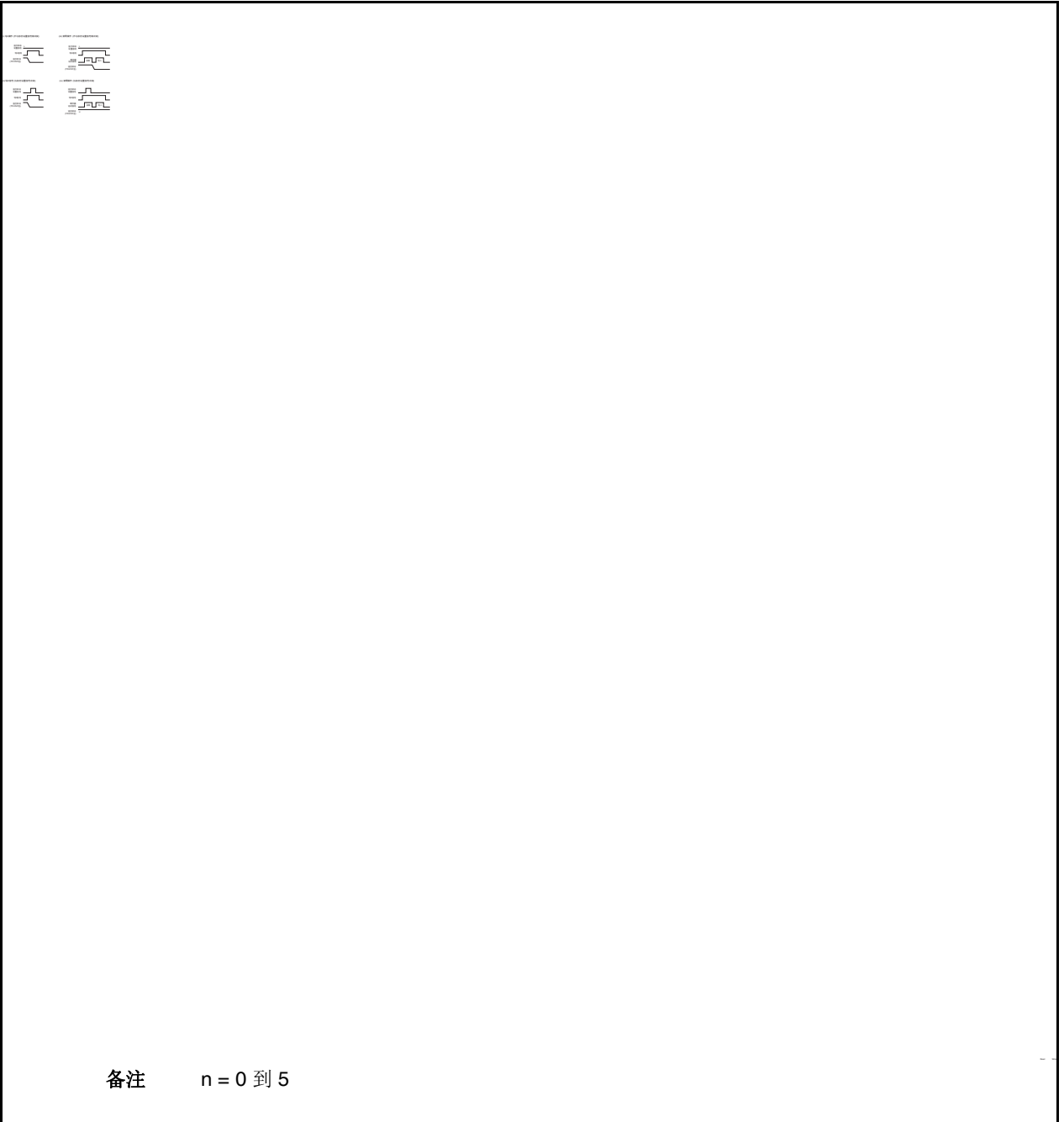


注 这里的溢出次数计数器是建立于内置 RAM 上的变量，可由软件任意进行设置。

- <1> 读取 TPnCCRm 寄存器 (设置为 TIPnm 引脚输入的初始值)。
- <2> 产生了溢出。在溢出中断服务程序中将溢出次数计数器加 1 并将溢出标志清零。
- <3> 产生了第二次溢出。在溢出中断服务程序中将溢出次数计数器加 1 并将溢出标志清零。
- <4> 读取 TPnCCRm 寄存器。
读取溢出次数计数器的值。
→ 假设此时溢出计数器的值为“N”，则脉冲宽度可按 $(N \times 10000H + D_{m1} - D_{m0})$ 计算求得。
在本例中，由于溢出产生了两次，所以脉冲宽度为 $(20000H + D_{m1} - D_{m0})$ 。
将溢出次数计数器清零(0H)。

(e) 将溢出标志位清零

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



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

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

7.5.7 脉宽测量模式 (TPnMD2 到 TPnMD0 位 = 110)

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

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

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

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

图 7-34. 脉宽测量模式的配置图

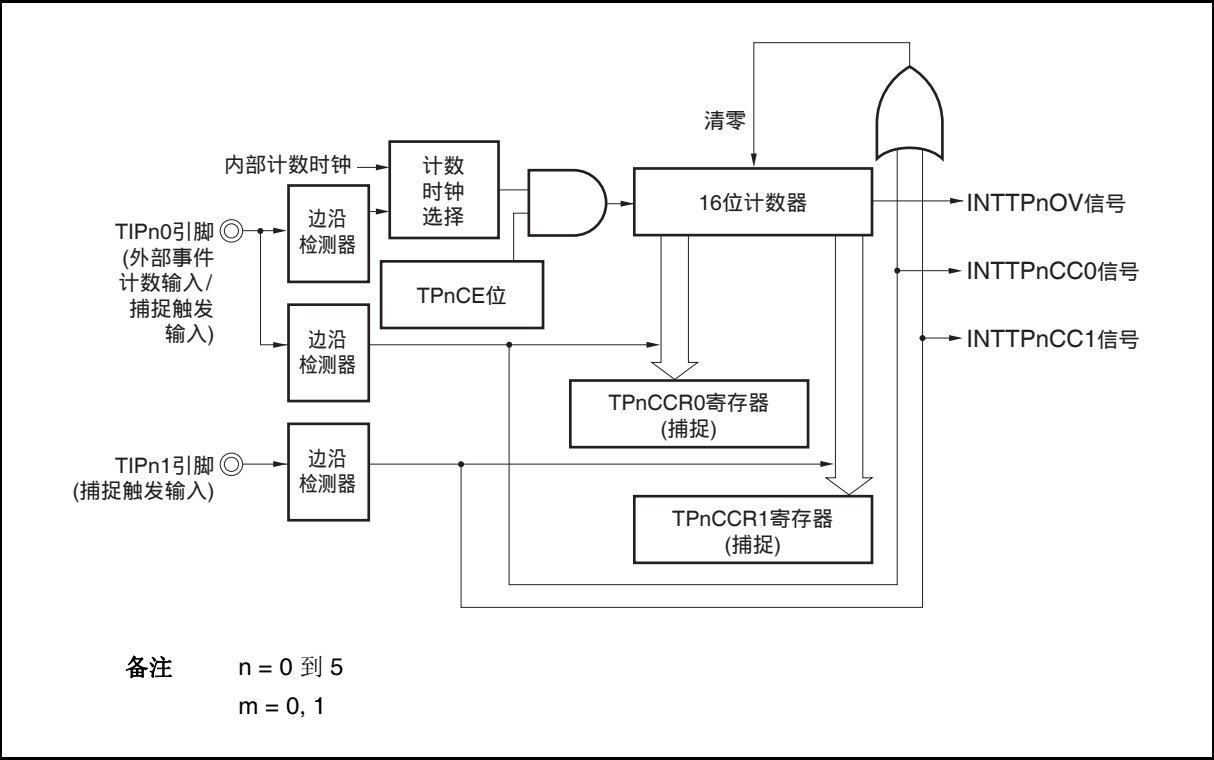
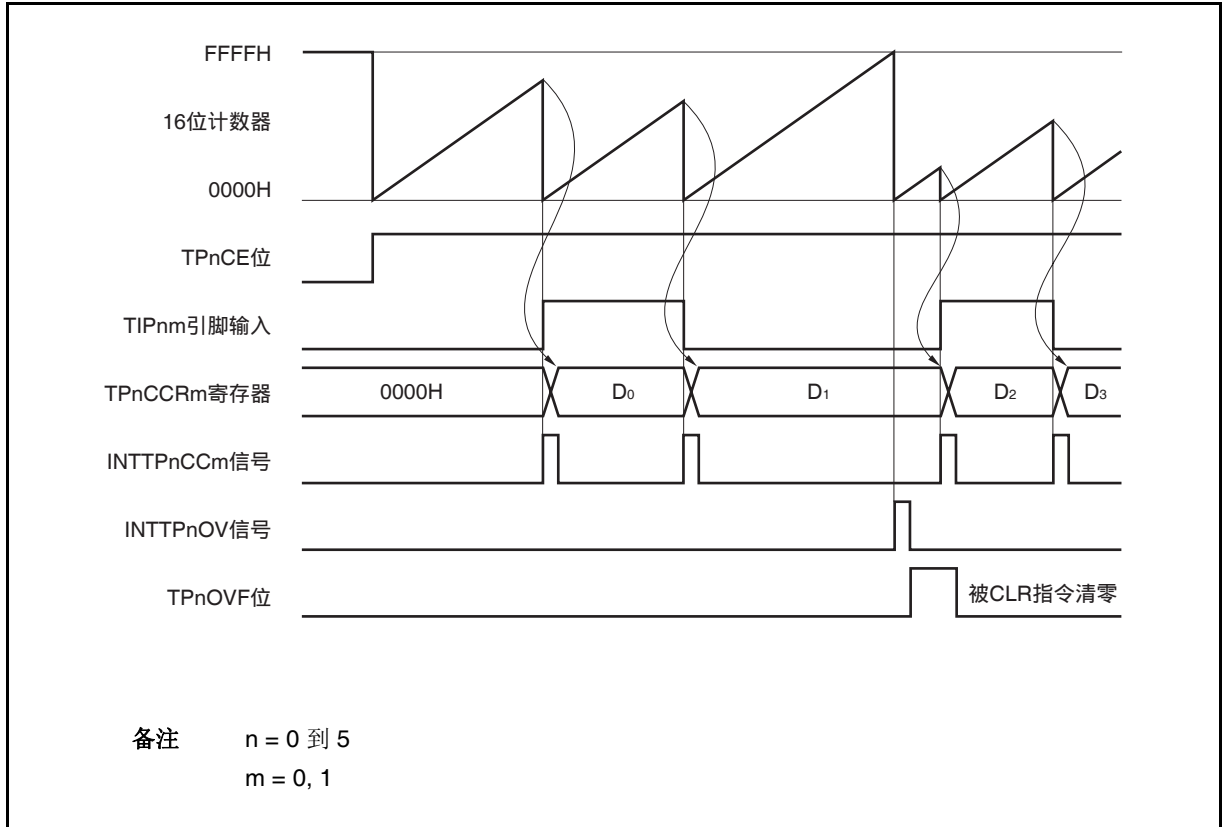


图 7-35. 脉宽测量模式的基本时序



TPnCTL0.TPnCE 位被置 1 后，16 位计数器开始计数。当检测到 TIPnm 引脚的有效沿时，16 位计数器的计数值将被保存到 TPnCCRM 寄存器中，同时计数器被清零并产生一个捕捉中断请求信号(INTTPnCCm)。

脉冲宽度可由下式求得。

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

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

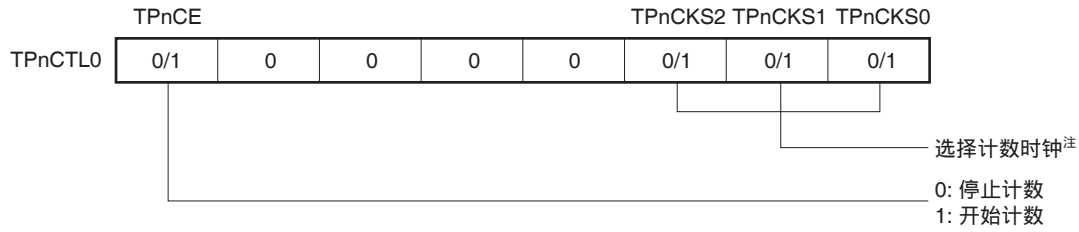
如果溢出标志被设置为 1，那么脉冲宽度可由下式求得。

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

备注 $n = 0$ 到 5
 $m = 0, 1$

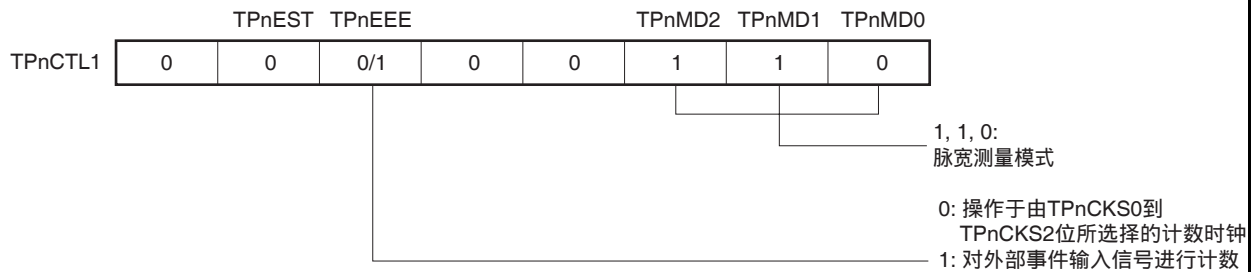
图 7-36. 脉宽测量模式的寄存器设置 (1/2)

(a) TMPn 控制寄存器 0 (TPnCTL0)

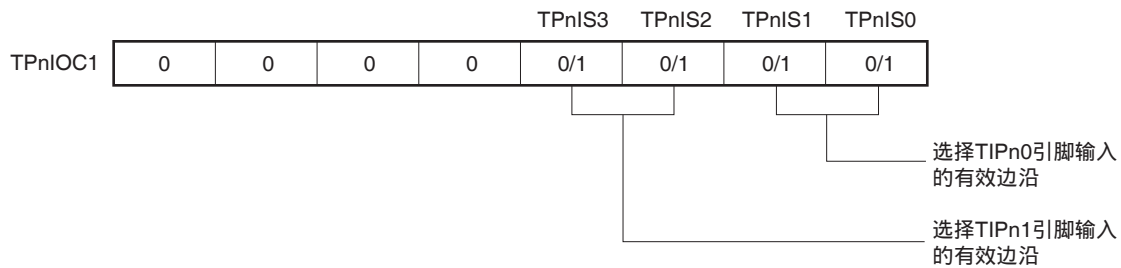


注 TPnEEE 位 = 1 时该设置无效。

(b) TMPn 控制寄存器 1 (TPnCTL1)



(c) TMPn I/O 控制寄存器 1 (TPnIOC1)



(d) TMPn I/O 控制寄存器 2 (TPnIOC2)

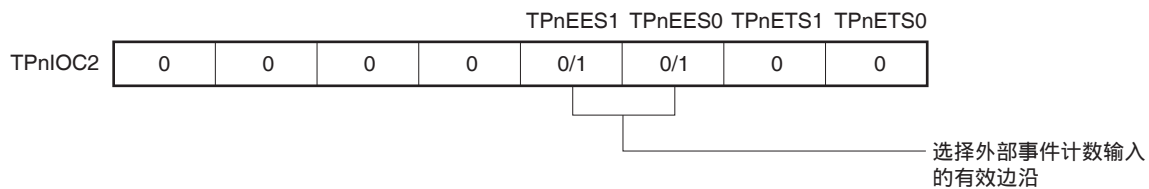
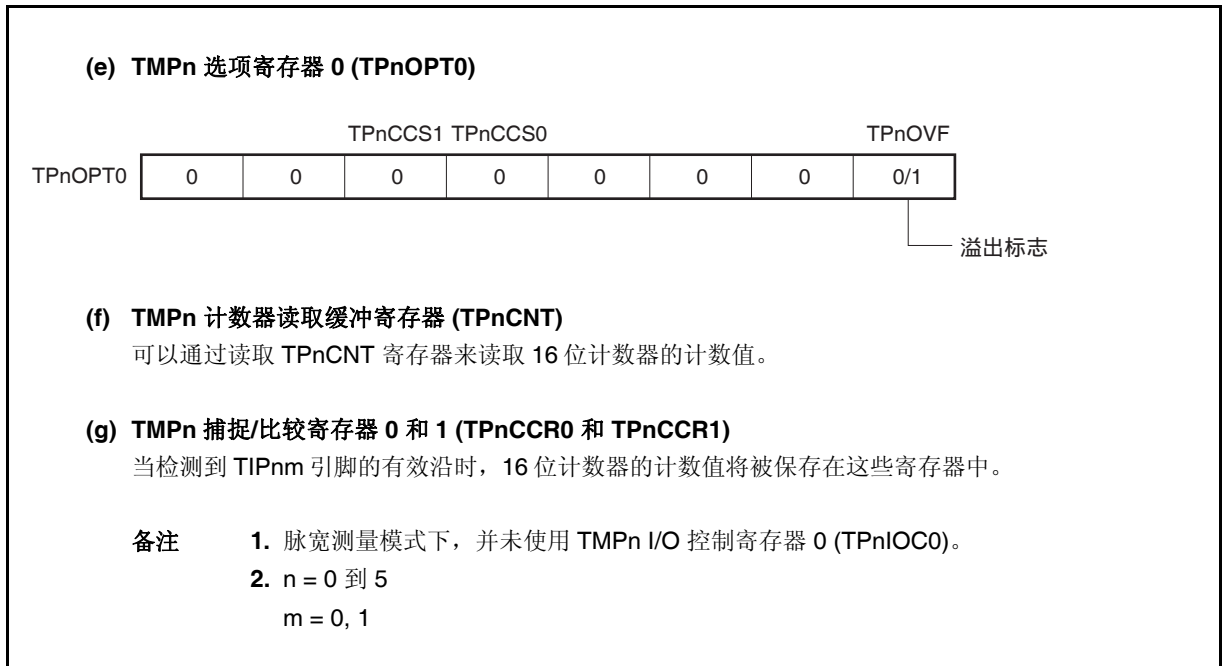
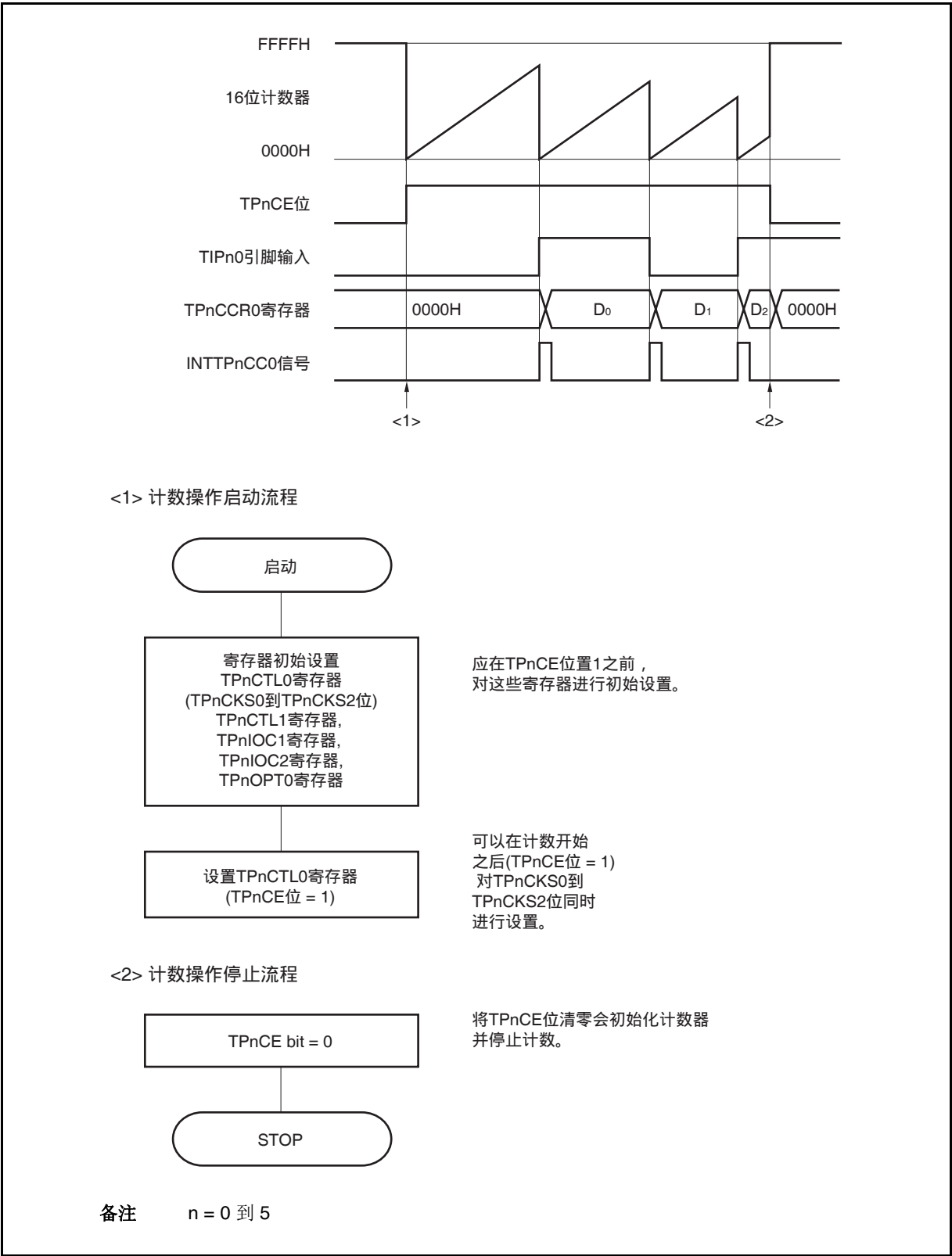


图 7-36. 脉宽测量模式的寄存器设置 (2/2)



(1) 脉宽测量模式的操作流程

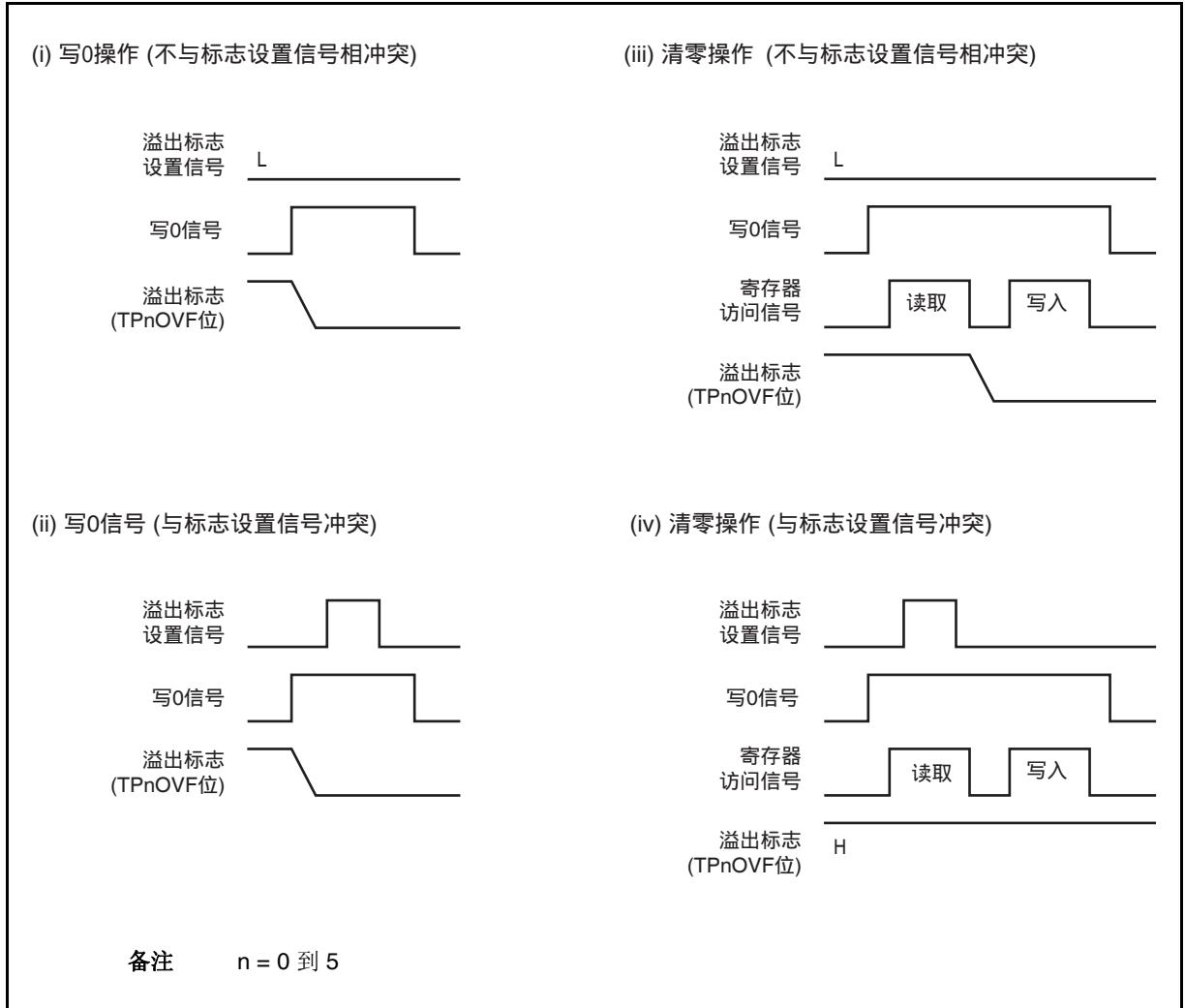
图 7-37. 脉宽测量模式的软件处理流程



(2) 脉宽测量模式的操作时序

(a) 将溢出标志位清零

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



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

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

7.5.8 定时器输出操作

TOPn0 和 TOPn1 引脚的工作模式与输出电平的关系如下表所示。

表 7-4. 各模式下的定时器输入控制

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

备注 n = 0 到 5

表 7-5. TOPn0 和 TOPn1 引脚在定时器输出控制位控制下的真值表

TPnIOC0.TPnOLm 位	TPnIOC0.TPnOEm 位	TPnCTL0.TPnCE 位	TOPnm 引脚的电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前为低电平，计数过程中为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平，计数过程中为低电平

备注 n = 0 到 5
m = 0, 1

7.6 选择器功能

V850ES/JG2 中，可以选择端口/定时器复用功能引脚(TIP10/TIP11)的输入信号和 UARTA 接收复用功能引脚(RXDA0/RXDA1)的输入信号之一作为 TMP1 的捕捉触发输入信号。

用户可以通过这个功能实现以下操作。

- 从端口/定时器复用功能引脚(TIP10/TIP11)的输入信号和 UARTA 接收复用功能引脚(RXDA0/RXDA1)的输入信号之中选择一个作为 TMP1 的 TIP10 和 TIP11 输入信号。
→ 选择 UART0 或 UART1 的 RXDA0 或 RXDA1 信号时，可以计算 UARTA 的 LIN 接收传输速率以及波特率误差。

- 注意事项**
- 使用选择器功能时，要在连接定时器之前将 TMP 的捕捉触发输入设置好。
 - 设置选择器功能之前要先停止相关周边 I/O(TMP 或 UARTA)的工作。

选择器功能的捕捉输入是由下面的寄存器指定的。

(1) 选择器操作控制寄存器 0 (SELCNT0)

SELCNT0 寄存器是用于选择 TMP1 的捕捉触发源的 8 位寄存器。

可以对该寄存器进行 8 位或 1 位的读写操作。

复位后该寄存器的值为 00H。

复位后:00H R/W 地址:FFFF808H

	7	6	5	<4>	<3>	2	1	0
SELCNT0	0	0	0	ISEL4	ISEL3	0	0	0

ISEL4	选择TIP1的输入信号 (TMP1)
0	TIP1引脚输入
1	RXDA1引脚输入

ISEL3	选择TIP1的输入信号 (TMP1)
0	TIP1引脚输入
1	RXDA0引脚输入

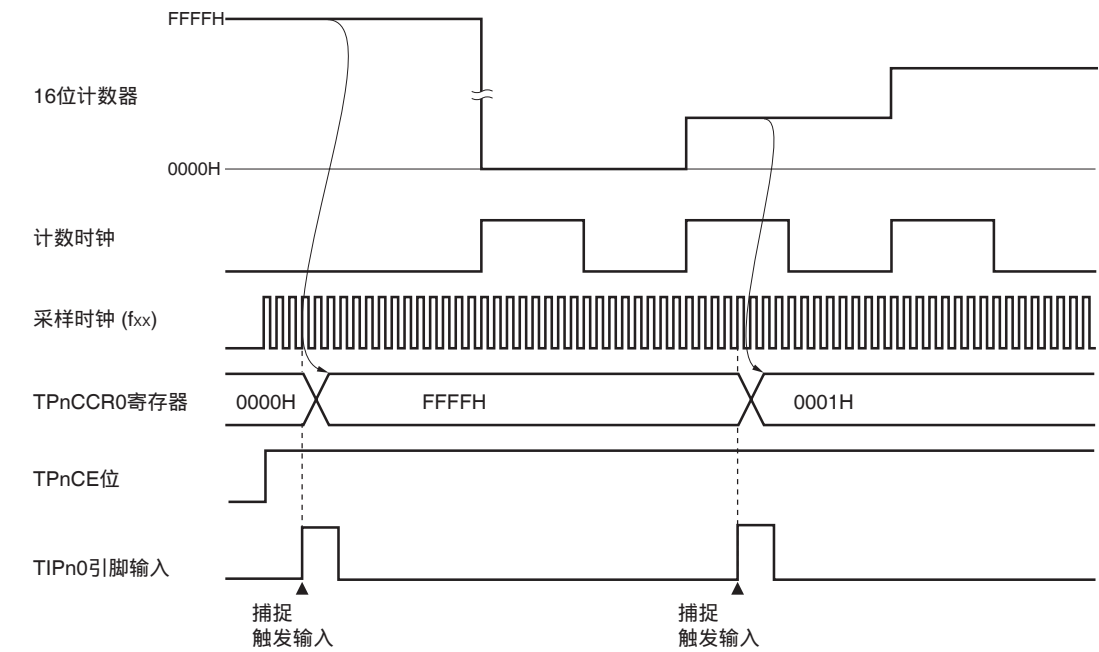
- 注意事项**
- 设置 ISEL3 和 ISEL4 位为 1 之前，要先将对应的引脚设置为捕捉输入模式。
 - 请确保第 5 到 7 位和 2 到 1 位为"0"

7.7 注意事项

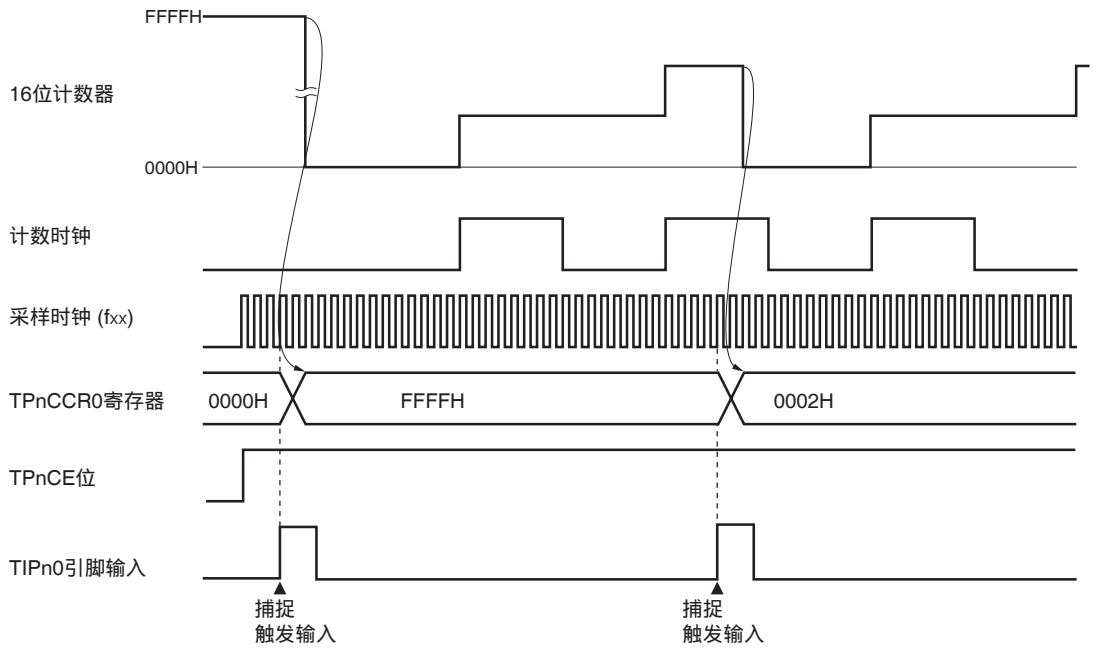
(1) 捕捉操作

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

(a) 自由运行定时器模式



(b) 脉宽测量模式



第八章 16 位定时器/事件计数器 Q (TMQ)

定时器 Q (TMQ) 是一个 16 位的定时器/事件计数器。
V850ES/JG2 具有 TMQ0。

8.1 概述

以下是 TMQ0 的概况。

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

8.2 功能

TMQ0 具有以下功能。

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

8.3 配置

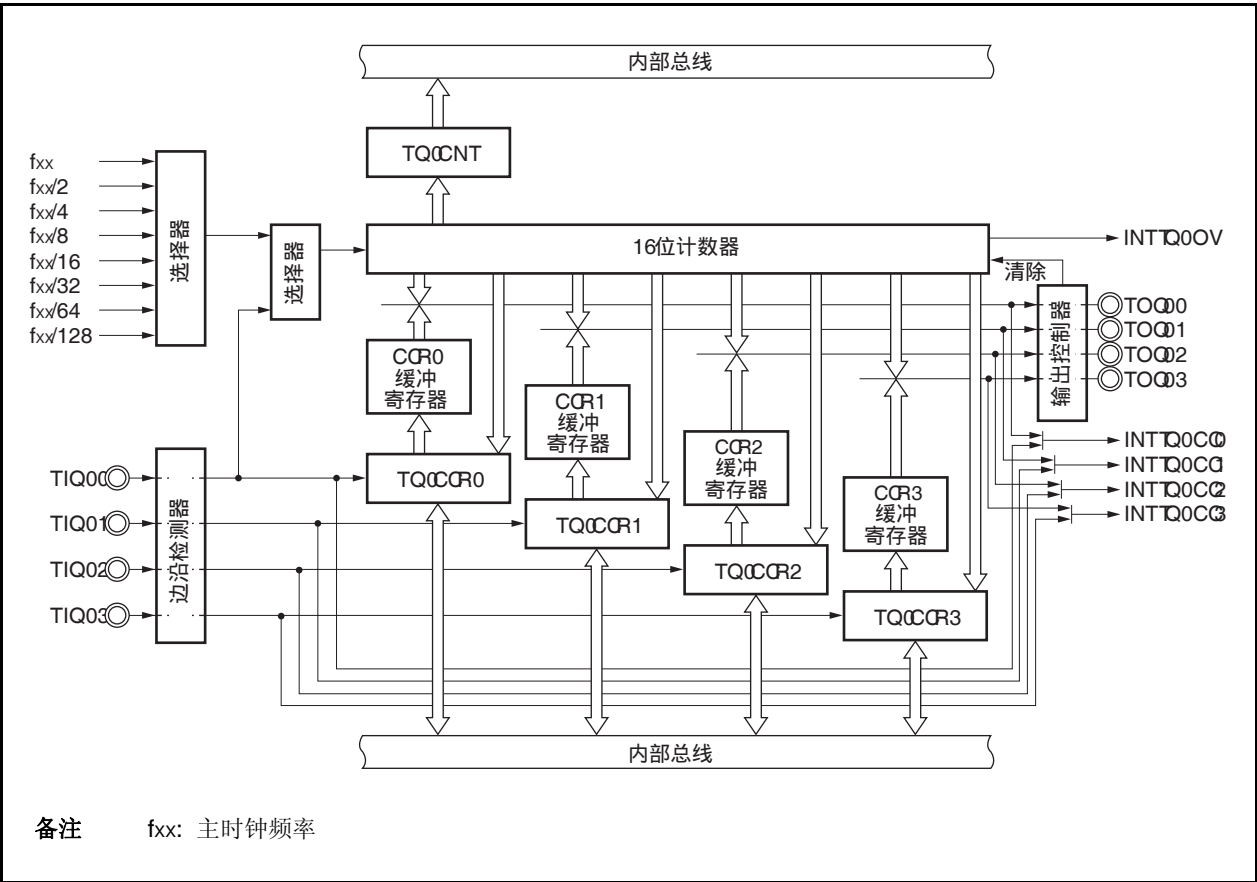
TMQ0 包含以下硬件。

表 8-1. TMQ0 的配置

器件	配置
定时器寄存器	16 位计数器
寄存器	TMQ0 捕捉/比较寄存器 0 至 3 (TQ0CCR0 至 TQ0CCR3) TMQ0 计数器读取缓冲寄存器 (TQ0CNT) CCR0 至 CCR3 缓冲寄存器
定时器输入	4 (TIQ00 ^{#1} 至 TIQ03 引脚)
定时器输出	4 (TOQ00 至 TOQ03 引脚)
控制寄存器 ^{#2}	TMQ0 控制寄存器 0, 1 (TQ0CTL0, TQ0CTL1) TMQ0 I/O 控制寄存器 0 至 2 (TQ0IOC0 至 TQ0IOC2) TMQ0 选项寄存器 0 (TQ0OPT0)

- 注
1. TIQ00 引脚的复用功能包括捕捉触发输入信号，外部事件计数输入信号和外部触发输入信号。
 2. 使用 TIQ00 至 TIQ03 和 TOQ00 至 TOQ03 引脚功能时，请参照表 4-15 端口引脚作为复用引脚使用。

图 8-1. TMQ0 的框图



(1) 16 位计数器

该 16 位计数器可以对内部时钟或外部事件进行计数。

可以使用 TQ0CNT 寄存器读取该计数器的计数值。

当 TQ0CTL0.TQ0CE 位 = 0 时, 16 位计数器的值为 FFFFH。如果此时对 TQ0CNT 寄存器进行读取, 则读取值将为 0000H。

复位输入将把 TQ0CE 位清零。因此, 16 位计数器也会同时被置为 FFFFH。

(2) CCR0 缓冲寄存器

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

当 TQ0CCR0 寄存器被作为比较寄存器使用时, 向 TQ0CCR0 寄存器写入的数据将被传送到 CCR0 缓冲寄存器。如果 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配, 会产生一个比较匹配中断请求信号 (INTTQ0CC0)。

CCR0 缓冲寄存器不能直接被读写。

复位后 CCR0 缓冲寄存器和 TQ0CCR0 寄存器都将被清零。

(3) CCR1 缓冲寄存器

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

当 TQ0CCR1 寄存器被作为比较寄存器使用时, 向 TQ0CCR1 寄存器写入的数据将被传送到 CCR1 缓冲寄存器。如果 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配, 会产生一个比较匹配中断请求信号 (INTTQ0CC1)。

CCR1 缓冲寄存器不能直接被读写。

复位后 CCR1 缓冲寄存器和 TQ0CCR1 寄存器都将被清为 0000H。

(4) CCR2 缓冲寄存器

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

当 TQ0CCR2 寄存器被作为比较寄存器使用时, 向 TQ0CCR2 寄存器写入的数据将被传送到 CCR2 缓冲寄存器。如果 16 位计数器的计数值与 CCR2 缓冲寄存器的值匹配, 会产生一个比较匹配中断请求信号 (INTTQ0CC2)。

CCR2 缓冲寄存器不能直接被读写。

复位后 CCR2 缓冲寄存器和 TQ0CCR2 寄存器都将被清为 0000H。

(5) CCR3 缓冲寄存器

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

当 TQ0CCR3 寄存器被作为比较寄存器使用时, 向 TQ0CCR3 寄存器写入的数据将被传送到 CCR3 缓冲寄存器。如果 16 位计数器的计数值与 CCR3 缓冲寄存器的值匹配, 会产生一个比较匹配中断请求信号 (INTTQ0CC3)。

CCR3 缓冲寄存器不能直接被读写。

复位后 CCR3 缓冲寄存器和 TQ0CCR3 寄存器都将被清为 0000H。

(6) 边沿检测器

该电路用于检测 TIQ00 和 TIQ03 引脚输入的有效边沿。通过 TQ0IOC1 和 TQ0IOC2 寄存器选择无有效沿, 上升沿, 下降沿或双沿有效。

(7) 输出控制器

该电路用于控制 TOQ00 至 TOQ03 引脚的输出。输出控制器是由 TQ0IOC0 寄存器控制的。

(8) 选择器

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

8.4 寄存器

用于控制 TMQ0 的寄存器如下所述。

- TMQ0 控制寄存器 0 (TQ0CTL0)
- TMQ0 控制寄存器 1 (TQ0CTL1)
- TMQ0 I/O 控制寄存器 0 (TQ0IOC0)
- TMQ0 I/O 控制寄存器 1 (TQ0IOC1)
- TMQ0 I/O 控制寄存器 2 (TQ0IOC2)
- TMQ0 选项寄存器 0 (TQ0OPT0)
- TMQ0 捕捉/比较寄存器 0 (TQ0CCR0)
- TMQ0 捕捉/比较寄存器 1 (TQ0CCR1)
- TMQ0 捕捉/比较寄存器 2 (TQ0CCR2)
- TMQ0 捕捉/比较寄存器 3 (TQ0CCR3)
- TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

备注 使用 TIQ00 至 TIQ03 和 TOQ00 至 TOQ03 引脚功能时，请参照表 4-15 端口引脚作为复用引脚使用。

(1) TMQ0 控制寄存器 0 (TQ0CTL0)

TQ0CTL0 寄存器为用于控制 TMQ0 操作的 8 位寄存器。

可以对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

可使用软件向 TQ0CTL0 寄存器写入相同的数值。

复位后: 00H		R/W 地址:		FFFFFF540H						
		<7>		6	5	4	3	2	1	0
TQ0CTL0	TQ0CE	0	0	0	0	0	TQ0CKS2	TQ0CKS1	TQ0CKS0	

TQ0CE	TMQ0操作控制
0	禁止操作TMQ0 (TMQ0异步复位 ^注)。
1	允许操作TMQ0 TMQ0开始工作。

TQ0CKS2	TQ0CKS1	TQ0CKS0	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 TQ0OPT0.TQ0OVF 位, 16 位计数器, 定时器输出 (TOQ00 至 TOQ03 引脚)

- 注意事项
- 在 TQ0CE 位 = 0 时, 设置 TQ0CKS2 至 TQ0CKS0 位。
当将 TQ0CE 位的值从 0 改变为 1 时, 可同时对 TQ0CKS2 至 TQ0CKS0 位进行设置。
 - 请务必将第 3 到第 6 位清零。

备注 f_{xx}: 主时钟频率

(2) TMQ0 控制寄存器 1 (TQ0CTL1)

TQ0CTL1 寄存器是用于控制 TMQ0 操作的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后: 00H R/W 地址: FFFFF541H

	7	<6>	<5>	4	3	2	1	0
TQ0CTL1	0	TQ0EST	TQ0EEE	0	0	TQ0MD2	TQ0MD1	TQ0MD0

TQ0EST	软件触发控制
0	
1	为外部触发输入产生有效信号。 单脉冲输出模式: 通过向 TQ0EST 位写 1 作为触发源来输出单脉冲。 外部触发脉冲输出模式: 通过向 TQ0EST 位写 1 来输出 PWM 波形

TQ0EEE	计数时钟选择
0	禁止通过外部事件计数输入操作。 (通过 TQ0CTL0.TQ0CK 至 TQ0CK2 位选择的计数时钟执行计数)
1	允许通过外部事件计数输入操作。 (在外部事件计数输入信号的有效沿执行计数。)
TQ0EEE 位选择通过内部计数时钟或外部事件计数输入的有效沿执行计数。	

TQ0MD2	TQ0MD1	TQ0MD0	定时器模式选择
0	0	0	间隔定时器模式
0	0	1	外部事件计数模式
0	1	0	外部触发脉冲输出模式
0	1	1	单脉冲输出模式
1	0	0	PWM 输出模式
1	0	1	自由运行定时器模式
1	1	0	脉宽测量模式
1	1	1	禁止设置

- 注意事项**
1. 对 TQ0EST 位的设置只有在外部触发脉冲输出模式或单脉冲输出模式才有效。在其它模式下, 对该位的置 1 操作将被忽略。
 2. 在外部事件计数模式下, 无论 TQ0EEE 位如何设置, 外部事件计数输入都是被选中的。
 3. 只有在 TQ0CTL0.TQ0CE 位 = 0 时才可以对 TQ0EEE 以及 TQ0MD2 至 TQ0MD0 位进行设置(TQ0CE 位 = 1 时, 可以对这些位写入相同的值)。如果在 TQ0CE 位 = 1 时对这些位进行了改写, 那么操作结果将不能被保证。若由于误操作引起了改写, 则要将 TQ0CE 位清零然后再次设置这些寄存器位。
 4. 请务必将第 3,4 和 7 位清零。

(3) TMQ0 I/O 控制寄存器 0 (TQ0IOC0)

TQ0IOC0 寄存器是用于控制定时器输出的 8 位寄存器 (TOQ00 至 TOQ03 引脚)。
可对该寄存器进行 8 位或 1 位的读写操作。
复位输入将把该寄存器设置为 00H。

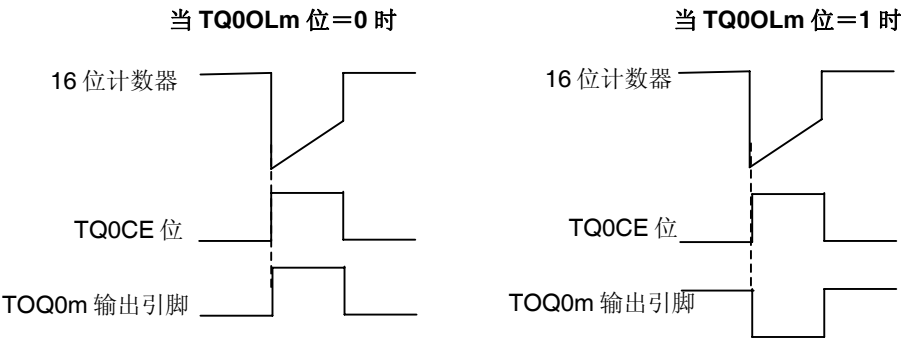
复位后: 00H R/W 地址: FFFFF542H

	7	<6>	5	<4>	3	<2>	1	<0>
TQ0IOC0	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0

TQ0OLm	TO00m 引脚输出电平设置 (m = 0 to 3)
0	禁止TO00m引脚输出反转
1	允许TO00m引脚输出反转

TQ0OEm	TO00m 引脚输出设置 (m = 0 to 3)
0	禁止定时器输出 当 TQ0OLm 位 = 0: TO00m 引脚输出低电平 当 TQ0OLm 位 = 1: TO00m 引脚输出高电平
1	允许定时器输出 (TO00m 引脚输出一个方波).

注：计时器输出引脚(TOQ0m)的输出电平，由下面显示的 TQ0OLm 位决定



- 注意事项
1. 只有在 TQ0CTL0.TQ0CE 位 = 0 时才可以对 TQ0OLm 和 TQ0OEm 位进行改写(当 TQ0CE 位 = 1 时可以向这些位写入相同的值)。若由于误操作引起了改写，则要将 TQ0CE 位清零然后再次设置这些寄存器位。
 2. 当 TQ0CE 和 TQ0OEm 位为 0 时，即使对 TQ0OLm 位进行了操作，TOQ0m 引脚的输出电平也不能确定。

备注 m = 0 至 3

(4) TMQ0 I/O 控制寄存器 1 (TQ0IOC1)

TQ0IOC1 寄存器是用于控制捕捉触发输入信号(TIQ00 至 TIQ03 引脚)有效沿的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后: 00H R/W 地址: FFFFF543H

	7	6	5	4	3	2	1	0
TQ0IOC1	TQ0S7	TQ0S6	TQ0S5	TQ0S4	TQ0S3	TQ0S2	TQ0S1	TQ0S0

TQ0S7	TQ0S6	捕捉触发输入信号 (TIQ03 脚) 有效沿设置
0	0	无边沿检测 (捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	上升下降沿均检测

TQ0S5	TQ0S4	捕捉触发输入信号 (TIQ02 脚) 有效沿设置
0	0	无边沿检测 (捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	上升下降沿均检测

TQ0S3	TQ0S2	捕捉触发输入信号 (TIQ01 脚) 有效沿设置
0	0	无边沿检测 (捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	上升下降沿均检测

TQ0S1	TQ0S0	捕捉触发输入信号 (TIQ00 脚) 有效沿设置
0	0	无边沿检测 (捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	上升下降沿均检测

- 注意事项**
1. 只有在 **TQ0CTL0.TQ0CE** 位 = 0 时才可以对 **TQ0IS7** 至 **TQ0IS0** 进行改写(**TQ0CE** 位 = 1 时只可以向这些位写入相同的值)。若由于误操作引起了改写, 则要将 **TQ0CE** 位清零然后再次设置这些寄存器位。
 2. **TQ0IS7** 至 **TQ0IS0** 位只有在自由运行定时器模式和脉宽测量模式下才有效。其它模式下, 无法进行捕捉操作。

(5) TMQ0 I/O 控制寄存器 2 (TQ0IOC2)

TQ0IOC2 寄存器是用于控制外部事件计数输入信号(TIQ00 引脚)有效沿和外部触发输入信号(TIQ00 引脚)有效沿的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后: 00H	R/W	地址:	FFFFFF544H							
			7	6	5	4	3	2	1	0
TQ0OC2			0	0	0	0	TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0

TQ0EES1	TQ0EES0	外部事件计数输入信号 (TIQ00 引脚) 有效沿设置
0	0	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	上升下降沿均检测

TQ0ETS1	TQ0ETS0	外部触发输入信号 (TIQ00 引脚) 有效沿设置
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	上升下降沿均检测

- 注意事项**
1. 在 TQ0CTL0.TQ0CE 位 = 0 时才可对 TQ0EES1, TQ0EES0, TQ0ETS1 和 TQ0ETS0 位进行改写 (TQ0CE 位 = 1 时只可以向这些位写入相同的值)。若在 TQ0CE 位 = 1 由于误操作引起了改写, 则要将 TQ0CE 位清零然后再次设置这些寄存器位。
 2. TQ0EES1 和 TQ0EES0 位只有在 TQ0CTL1.TQ0EEE 位 = 1 或设置了外部事件计数模式 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 001) 时才有效。
 3. TQ0ETS1 和 TQ0ETS0 位只有在外部触发脉冲输出模式 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 010) 或单脉冲输出模式 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 = 011) 下才有效。

(6) TMQ0 选项寄存器 0 (TQ0OPT0)

TQ0OPT0 寄存器是用于设置捕捉/比较操作和检测溢出的 8 位寄存器。
可对该寄存器进行 8 位或 1 位的读写操作。
复位输入将把该寄存器设置为 00H。

复位后: 00H R/W 地址: FFFFF545H

	7	6	5	4	3	2	1	<0>
TQ0OPT0	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0	0	0	0	TQ0OVF

TQ0CCSm	TQ0CCRm寄存器捕捉/比较选择
0	选择为比较寄存器
1	选择为捕捉寄存器
只有在自由运行定时器模式下 TQ0CCSm位的设置才有效	

TQ0OVF	TMQ0溢出检测
置 (1)	发生溢出
归 (0)	TQ0OVF位写 0 或 TQ0CTL0.Q0CE位 = 0
<ul style="list-style-type: none">• 在自由运行定时器模式下或脉冲宽度测量模式下16位计数器计数值从FFFFH到 0000H溢出时TQ0OVF位复位。• 在自由运行定时器和脉宽测量模式下，TQ0OVF位被置1的同时会产生一个中断请求信号(INTTQ0OV)• TQ0OVF位被置1后，即使对该位或TQ0OPT寄存器进行读取操作，TQ0OVF位也不会被清零。• 可以对TQ0OVF位进行读取和写入操作，但是不能通过软件将TQ0OVF位置1。向其写入1不会对TMQ0的操作造成任何影响。	

注意事项 1. 在 TQ0CTL0.TQ0CE 位 = 0 时才可对 TQ0CCS3 和 TQ0CCS0 位进行改写 (TQ0CE 位 = 1 时只可以向这些位写入相同的值)。
若由于误操作引起了改写，则要将 TQ0CE 位清零然后再次设置这些寄存器位。
2. 请务必将第 1 到 3 位清零。

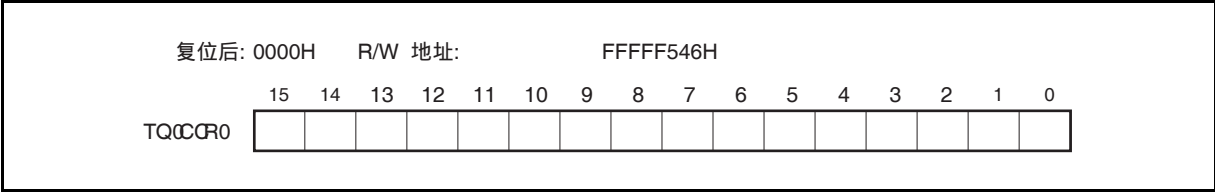
备注 m = 0 至 3

(7) TMQ0 捕捉/比较寄存器 0 (TQ0CCR0)

根据选择模式的不同，TQ0CCR0 寄存器可用作捕捉寄存器或比较寄存器。
该寄存器只有在自由运行模式下根据 TQ0OPT0.TQ0CCS0 位设置的不同，可以用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TQ0CCR0 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。
TQ0CCR0 寄存器可以在操作过程中进行读写。
可对该寄存器进行 16 位的读写操作。
复位输入将把该寄存器设置为 0000H。

- ★
- 注意事项 以下情况下禁止访问 TQ0CCR0 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。
- 当 CPU 工作于子时钟且主时钟振荡停止时

•当 CPU 工作于内置振荡时钟时



(a) 作为比较寄存器时的功能

TQ0CCR0 寄存器可以在 TQ0CTL0.TQ0CE 位 = 1 时被改写。

向 TQ0CCR0 寄存器中设置的数值将被传送到 CCR0 缓冲寄存器中。当 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相匹配时，会产生一个比较匹配中断请求信号(INTTQ0CC0)。如果允许了 TOQ00 引脚输出，那么 TOQ00 引脚输出的电平将被反转。

当 TQ0CCR0 寄存器在间隔定时器模式，外部事件计数模式，外部触发脉冲输出模式，单脉冲输出模式或 PWM 输出模式下被用作周期寄存器时，如果 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相匹配，那么 16 位计数器将被清零(0000H)。

(b) 作为捕捉寄存器时的功能

当 TQ0CCR0 寄存器在自由运行定时器模式下被用作捕捉寄存器时，如果检测到了捕捉触发输入引脚(TIQ00 引脚)的有效边沿，那么 16 位计数器的计数值将被保存到 TQ0CCR0 寄存器中。在脉宽测量模式下，如果检测到了捕捉触发输入引脚(TIQ00 引脚)的有效边沿，那么 16 位计数器的计数值将被保存到 TQ0CCR0 寄存器中并且同时 16 位计数器被清零(0000H)。

即使捕捉操作和读取 TQ0CCR0 寄存器操作冲突，仍可正确读出 TQ0CCR0 寄存器中的值。

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 8-2. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

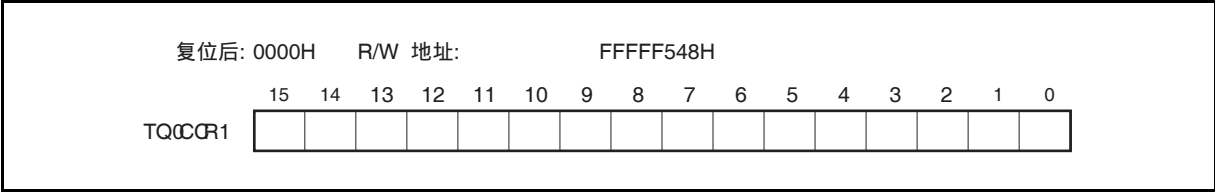
工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	—

(8) TMQ0 捕捉/比较寄存器 1 (TQ0CCR1)

根据选择模式的不同，TQ0CCR1 寄存器可用作捕捉寄存器或比较寄存器。
该寄存器只有在自由运行模式下根据 TQ0OPT0.TQ0CCS1 位设置的不同，可以用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TQ0CCR1 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。
TQ0CCR1 寄存器可以在操作过程中进行读写。
可对该寄存器进行 16 位的读写操作。
复位输入将把该寄存器设置为 0000H。

- ★
- 注意事项 以下情况下禁止访问 TQ0CCR1 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。
- 当 CPU 工作于子时钟且主时钟振荡停止时

•当 CPU 工作于内置振荡时钟时



(a) 作为比较寄存器时的功能

TQ0CCR1 寄存器可以在 TQ0CTL0.TQ0CE 位 = 1 时被改写。

向 TQ0CCR1 寄存器中设置的数值将被传送到 CCR1 缓冲寄存器中。当 16 位计数器的值与 CCR1 缓冲寄存器的值匹配时，产生比较匹配中断请求信号 (INTTQ0CC1)。如果此时允许 TOQ01 引脚输出，则 TOQ01 引脚的输出信号的电平将反转。

(b) 作为捕捉寄存器时的功能

在自由运行定时器模式下当 TQ0CCR1 寄存器用作捕捉寄存器时，如果检测到捕捉触发输入引脚(TIQ01 引脚)的有效沿，那么 16 位计数器的计数值将被保存到 TQ0CCR1 寄存器。在脉宽测量模式下，如果检测到捕捉输入引脚(TIQ01 引脚)的有效沿，那么 16 位计数器的计数值将被保存到 TQ0CCR1 寄存器，同时 16 位计数器清为(0000H)。

即使捕捉操作和读取 TQ0CCR1 寄存器的操作产生冲突，仍可正确读出 TQ0CCR1 寄存器中的值。

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 8-3. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

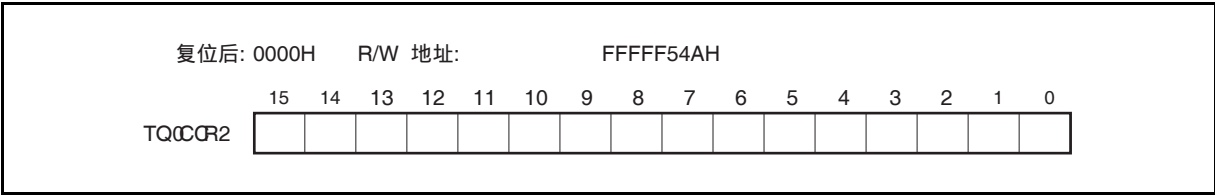
工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	—

(9) TMQ0 捕捉/比较寄存器 2 (TQ0CCR2)

根据选择模式的不同，TQ0CCR2 寄存器可用作捕捉寄存器或比较寄存器。
该寄存器只有在自由运行模式下根据 TQ0OPT0.TQ0CCS2 位设置的不同，可以用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TQ0CCR2 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。
TQ0CCR2 寄存器可以在操作过程中进行读写
可对该寄存器进行 16 位的读写操作。^注
复位输入将把该寄存器设置为 0000H。

- ★
- 注意事项 以下情况下禁止访问 TQ0CCR2 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。
- 当 CPU 工作于子时钟且主时钟振荡停止时

•当 CPU 工作于内置振荡时钟时



(a) 作为比较寄存器时的功能

TQ0CCR2 寄存器可以在 TQ0CTL0.TQ0CE 位 = 1 时被改写。

向 TQ0CCR2 寄存器中设置的数值将被传送到 CCR2 缓冲寄存器中。当 16 位计数器的值与 CCR2 缓冲寄存器的值匹配时，产生比较匹配中断请求信号 (INTTQ0CC2)。如果此时允许 TOQ02 引脚输出，则 TOQ02 引脚的输出信号的电平将反转。

(b) 作为捕捉寄存器时的功能

在自由运行定时器模式下当 TQ0CCR2 寄存器用作捕捉寄存器时，如果检测到捕捉触发输入引脚(TIQ02 引脚)的有效沿，那么 16 位计数器的计数值将被保存到 TQ0CCR2 寄存器。在脉宽测量模式下，如果检测到捕捉输入引脚(TIQ02 引脚)的有效沿，那么 16 位计数器的计数值将被保存到 TQ0CCR2 寄存器，同时 16 位计数器清为(0000H)。

即使捕捉操作和读取 TQ0CCR1 寄存器的操作产生冲突，仍可正确读出 TQ0CCR1 寄存器中的值。

下表列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 8-4. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

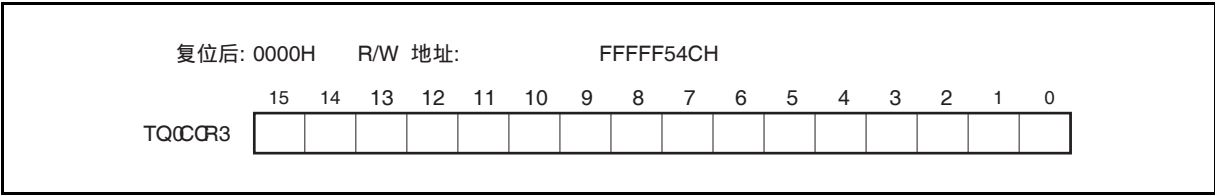
工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	—

(10)TMQ0 捕捉/比较寄存器 3 (TQ0CCR3)

根据选择模式的不同，TQ0CCR3 寄存器可用作捕捉寄存器或比较寄存器。
该寄存器只有在自由运行模式下根据 TQ0OPT0.TQ0CCS3 位设置的不同，可以用作捕捉寄存器或比较寄存器。在脉宽测量模式下，TQ0CCR3 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。
TQ0CCR3 寄存器可以在操作过程中进行读写
可对该寄存器进行 16 位的读写操作。
复位输入将把该寄存器设置为 0000H。

- ★
- 注意事项 以下情况下禁止访问 TQ0CCR3 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。
- 当 CPU 工作于子时钟且主时钟振荡停止时

•当 CPU 工作于内置振荡时钟时



(a) 作为比较寄存器时的功能

TQ0CCR3 寄存器可以在 TQ0CTL0.TQ0CE 位 = 1 时被改写。

向 TQ0CCR3 寄存器中设置的数值将被传送到 CCR3 缓冲寄存器中。当 16 位计数器的值与 CCR3 缓冲寄存器的值匹配时，产生比较匹配中断请求信号 (INTTQ0CC3)。如果此时允许 TOQ03 引脚输出，则 TOQ03 引脚的输出信号的电平将反转。

(b) 作为捕捉寄存器时的功能

在自由运行定时器模式下当 TQ0CCR3 寄存器用作捕捉寄存器时，如果检测到捕捉触发输入引脚(TIQ03 引脚)的有效沿，那么 16 位计数器的计数值将被保存到 TQ0CCR3 寄存器。在脉宽测量模式下，如果检测到捕捉输入引脚(TIQ03 引脚)的有效沿，那么 16 位计数器的计数值将被保存到 TQ0CCR3 寄存器，同时 16 位计数器清为(0000H)。

即使捕捉操作和读取 TQ0CCR3 寄存器的操作产生冲突，仍可正确读出 TQ0CCR3 寄存器中的值。

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

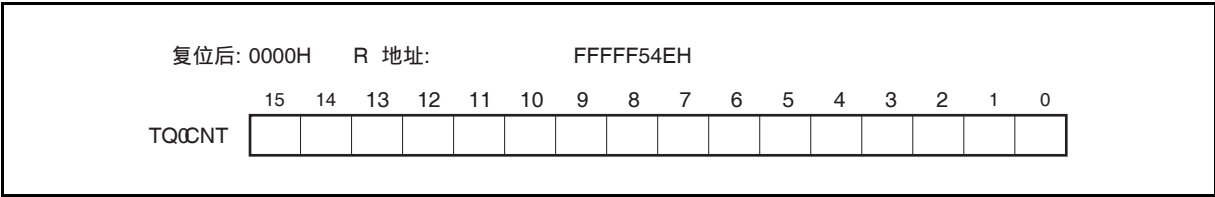
表 8-5. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

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

(11)TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

可以通过读取缓冲寄存器 TQ0CNT 来读取 16 位计数器的计数值。
如果该寄存器在 TQ0CTL0.TQ0CE 位 = 1 时被读取，那么读取的值就是 16 位计数器的计数值。
只可以对该寄存器进行 16 位的读取操作。
当 TQ0CE 位 = 0 时，TQ0CNT 寄存器被清为 0000H。如果此时对 TQ0CNT 寄存器进行读取，读取到的值将是 0000H 而不是 16 位计数器的值(FFFFH)。
复位后，TQ0CE 位被清零的同时 TQ0CNT 寄存器的值也被清零。

- ★ 注意事项 以下情况下禁止访问 TQ0CNT 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。
- 当 CPU 工作于子时钟且主时钟振荡停止时
 - 当 CPU 工作于内置振荡时钟时



8.5 操作

TMQ0 可进行以下操作。

操作	TQ0CTL1.TQ0EST 位 (软件触发位)	TIQ00 引脚 (外部触发输入)	捕捉/比较寄存器 设置	比较寄存器写入
间隔定时器模式	无效	无效	仅比较	可随时写入
外部事件计数模式 ^{注 1}	无效	无效	仅比较	可随时写入
外部触发脉冲输出模式 ^{注 2}	有效	有效	仅比较	同时写入
单脉冲输出模式 ^{注 2}	有效	有效	仅比较	可随时写入
PWM 输出模式	无效	无效	仅比较	同时写入
自由运行定时器模式	无效	无效	可选	可随时写入
脉宽测量模式 ^{注 2}	无效	无效	仅捕捉	不可操作

- 注**
1. 使用外部事件计数模式时，请指定为不检测 TIQ00 引脚捕捉触发输入有效沿(通过将 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位清零)。
 2. 使用外部触发脉冲输出模式，单脉冲输出模式和脉宽测量模式时，请选择内部时钟作为计数时钟(通过将 TQ0CTL1.TQ0EEE 位清零)。

8.5.1 间隔定时器模式 (TQ0MD2 至 TQ0MD0 位 = 000)

在间隔定时器模式下，当 TQ0CTL0.TQ0CE 位 = 1 时，每隔指定间隔时间都会产生一个中断请求信号 (INTTQ0CC0)，同时可以从 TOQ00 引脚输出一个周期为 2 倍间隔时间的方波。

通常情况下，间隔定时器模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。

图 8-2. 间隔定时器的配置图

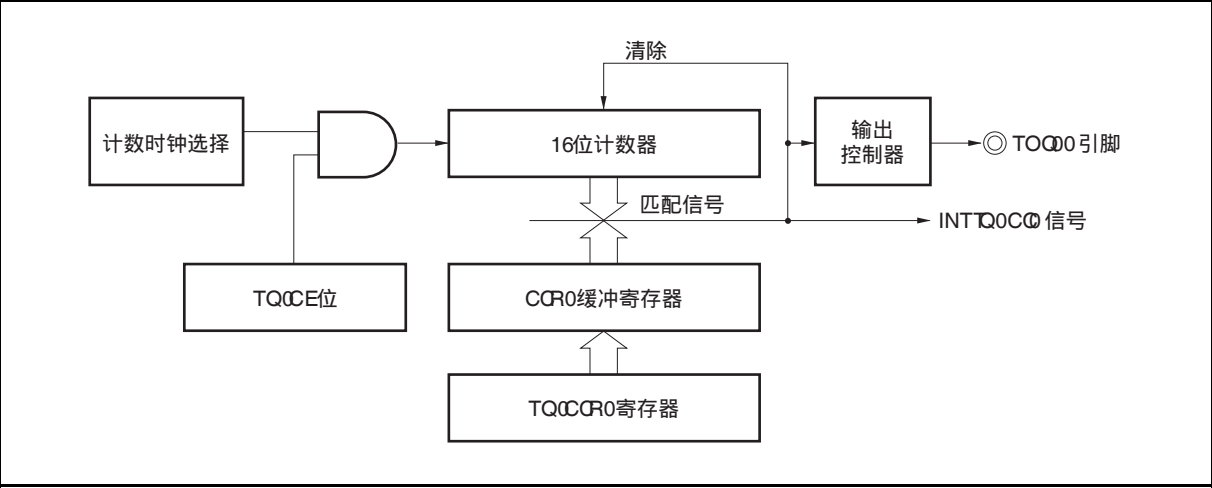
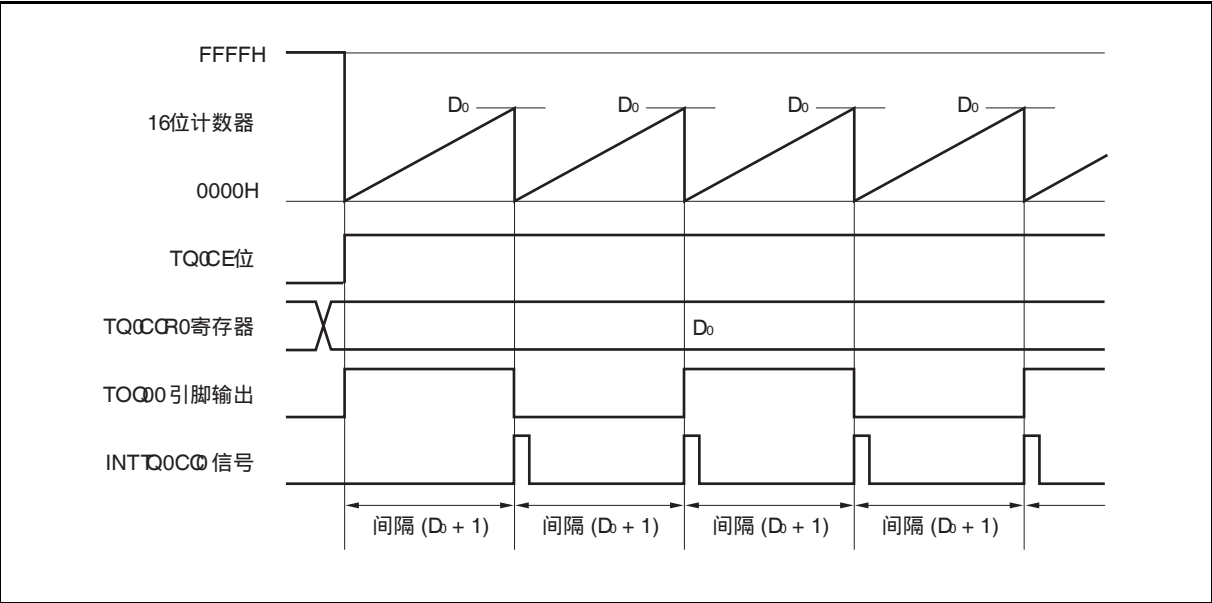


图 8-3. 间隔定时器模式操作的基本时序



16 位计数器将在 TQ0CE 位被置 1 时，与计数时钟同步从 FFFFH 清零为 0000H 并开始计数。此时，TOQ00 引脚的输出电平被反转。另外，TQ0CCR0 寄存器的设置值也会在此时被传送到 CCR0 缓冲寄存器。

当 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值匹配时，16 位计数器清零，TOQ00 引脚输出电平反转并且产生一个比较匹配中断请求信号 (INTTQ0CC0)。

间隔时间可由以下算式求得。

间隔时间 = (TPnCCR0 寄存器的设定值 + 1) × 计数时钟周期

图 8-4. 间隔定时器模式操作的寄存器设置(1/2)

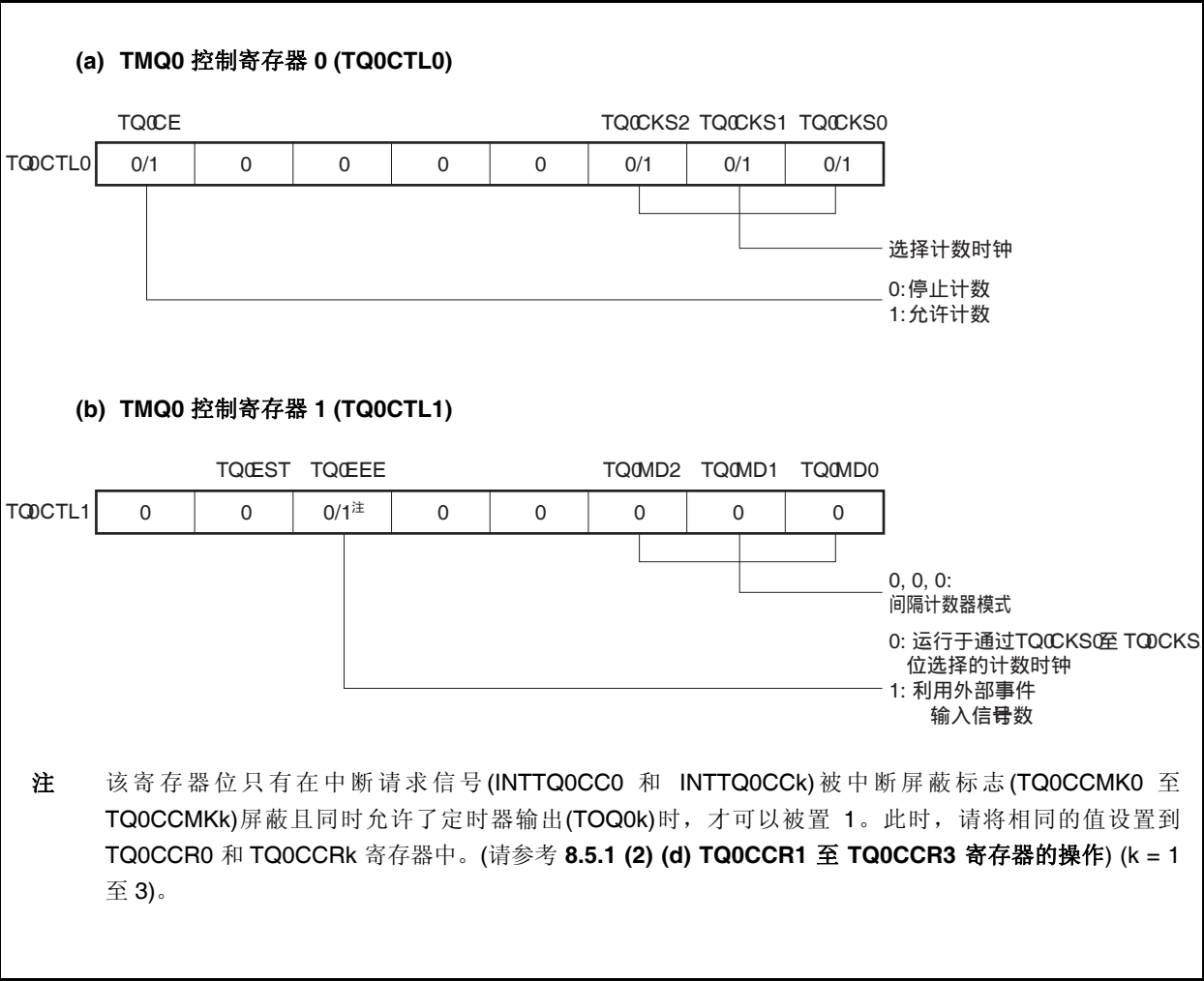
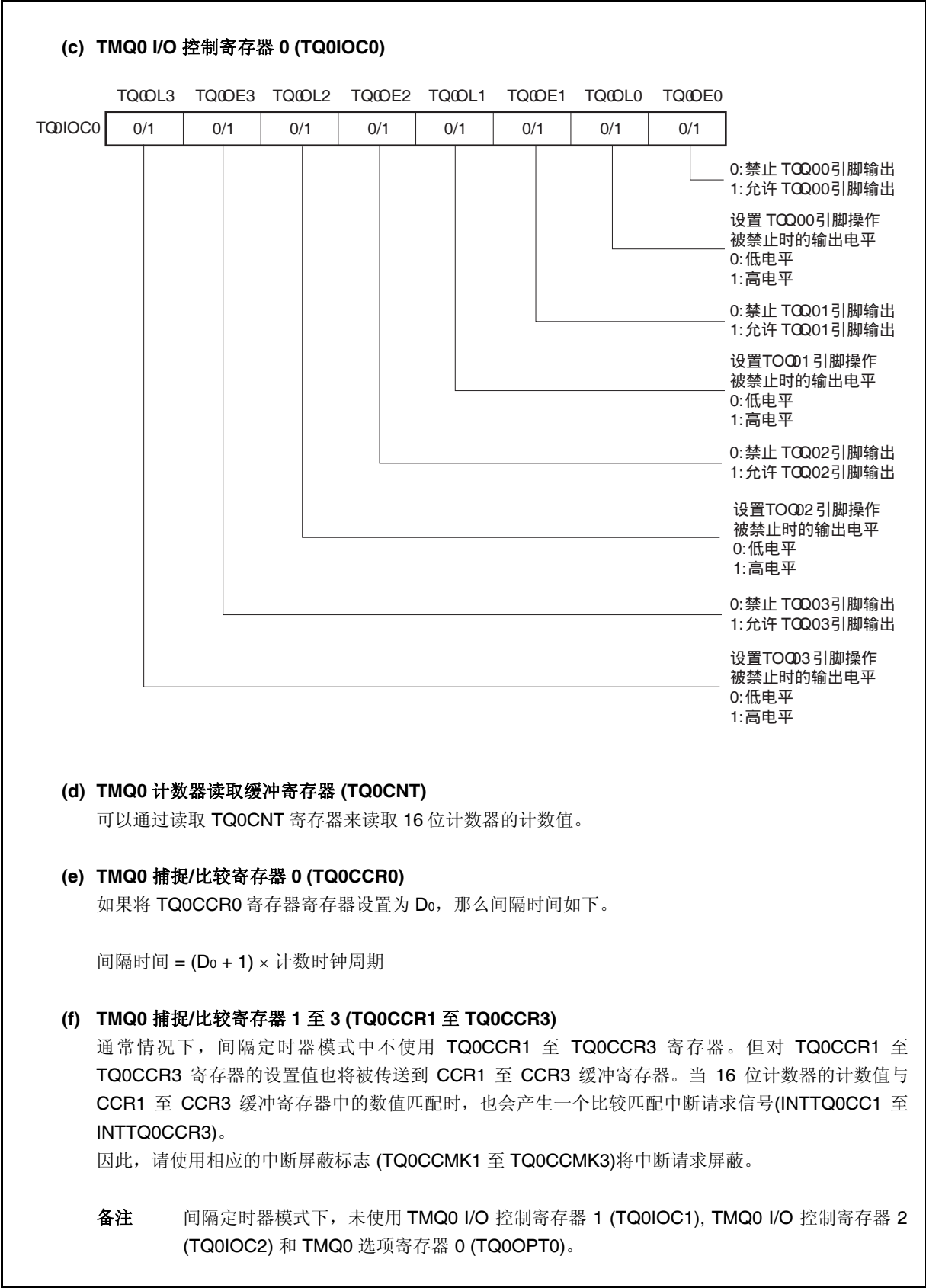
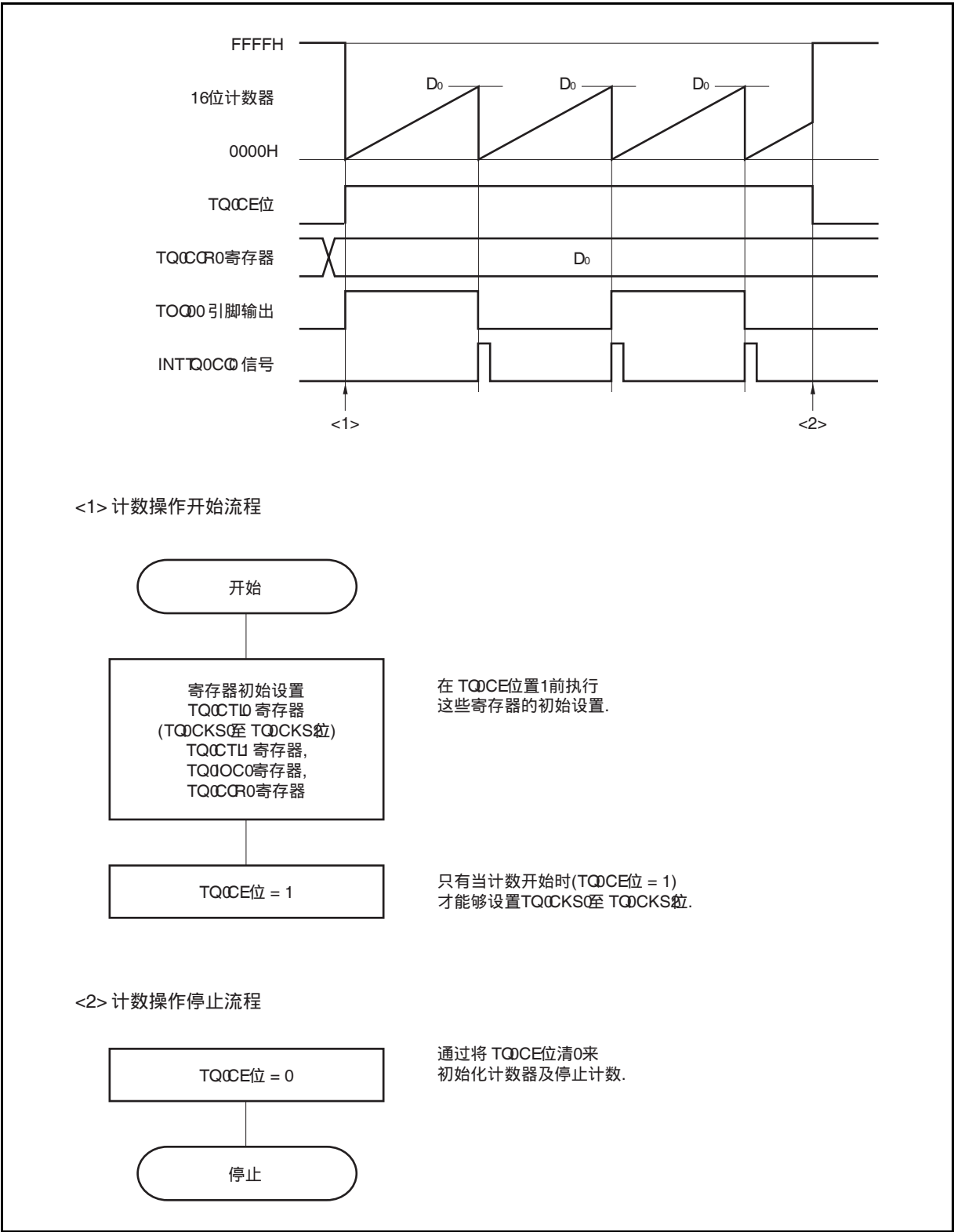


图 8-4. 间隔定时器模式操作的寄存器设置(2/2)



(1) 间隔定时器模式操作流程

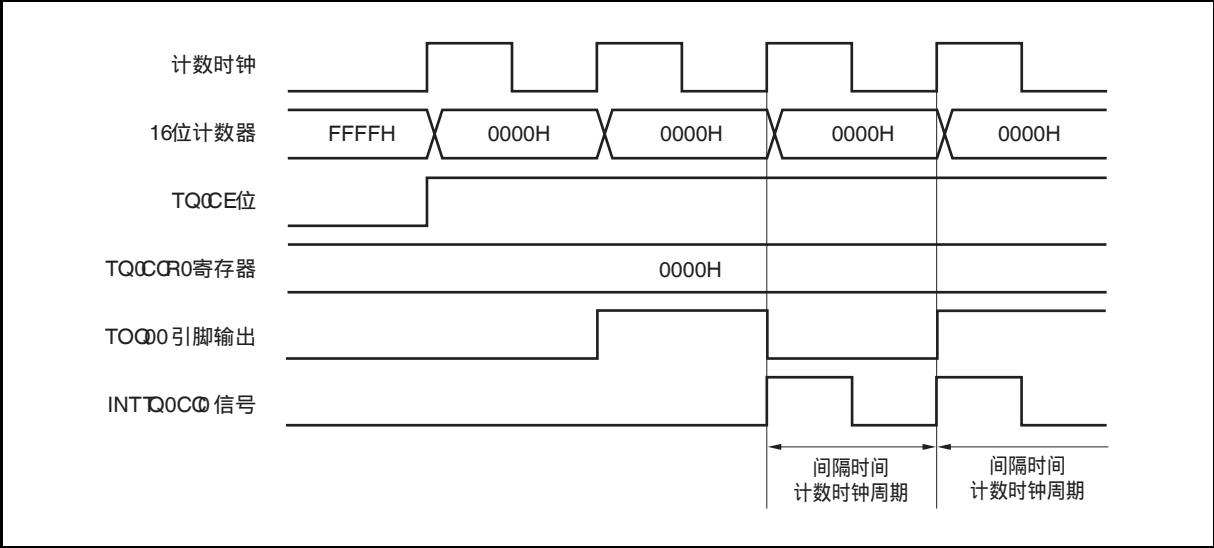
图 8-5. 间隔定时器模式下的软件处理流程



(2) 间隔定时器模式操作时序

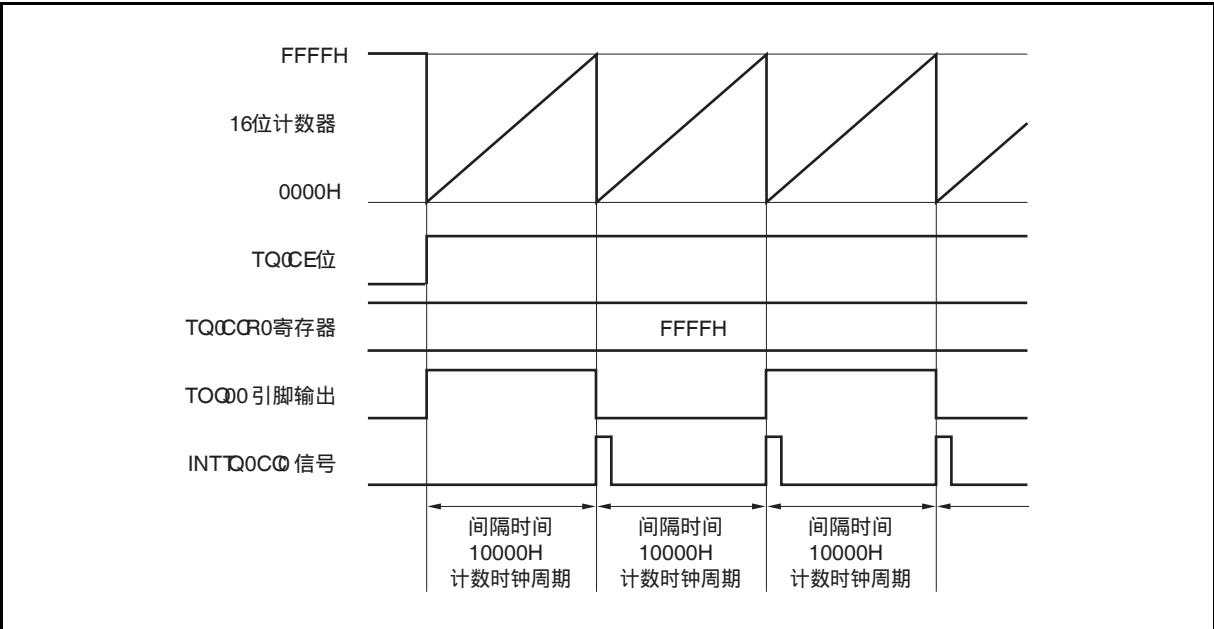
(a) TQ0CCR0 寄存器被设置为 0000H 时的操作

如果 TQ0CCR0 寄存器被设置为 0000H，那么在第二个计数时钟之后(含第二个)，都会产生 INTTQ0CC0 信号，并且 TOQ00 引脚的电平都会反转。
16 位计数器的计数值始终保持为 0000H。



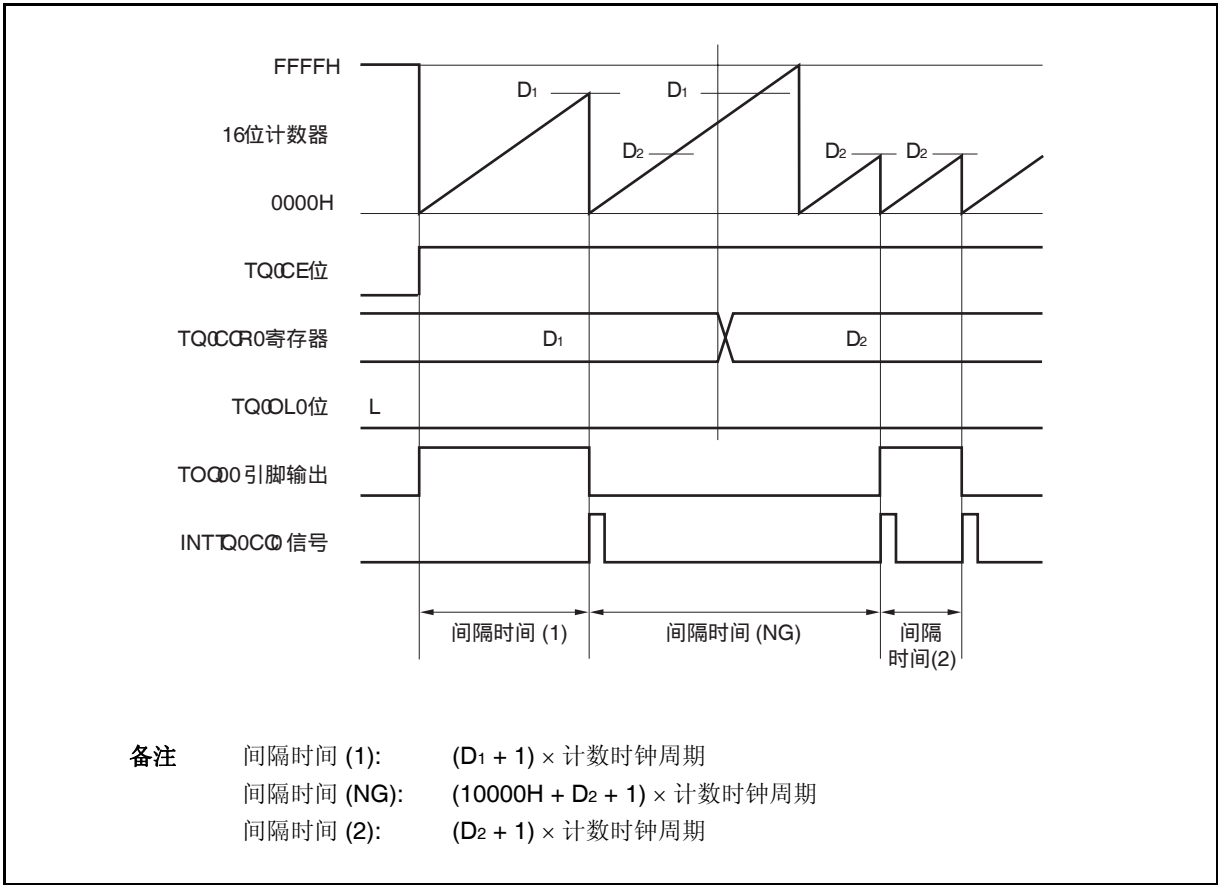
(b) TQ0CCR0 寄存器被设置为 FFFFH 时的操作

如果 TQ0CCR0 寄存器被设置为 FFFFH，那么 16 位计数器将计数到 FFFFH。并在下一个计数时钟时刻同步被清零。同时产生 INTTQ0CC0 信号，TOQ00 引脚电平反转。此时，不会产生溢出中断请求信号(INTTQ0OV)，溢出标志位(TQ0OPT0.TQ0OVF 位)也不会被置 1。



(c) 改写 TQ0CCR0 寄存器时需要注意的事项

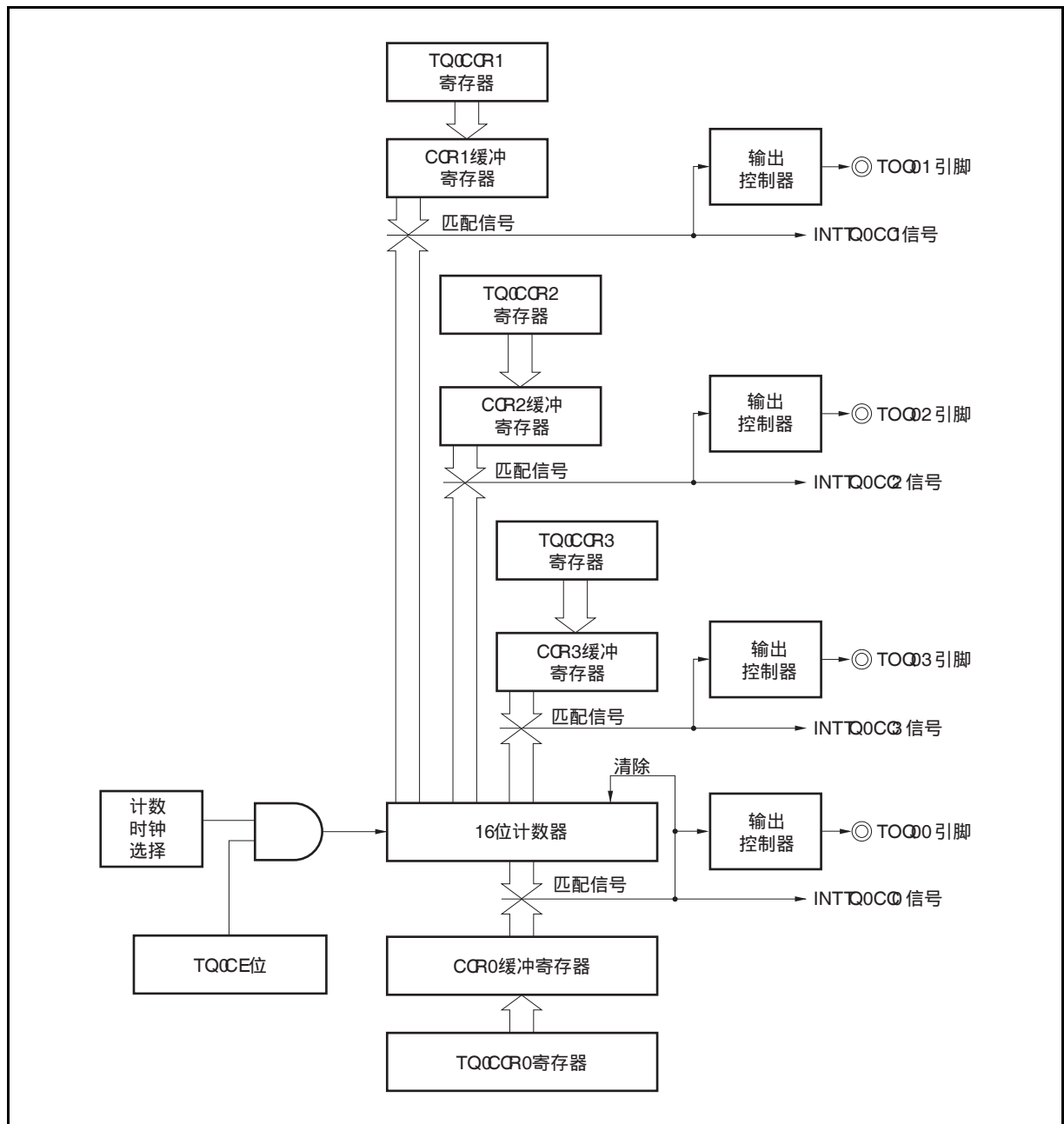
要将 TQ0CCR0 寄存器的值改小，首先停止计数然后再更改设置值。
如果在计数时向 TQ0CCR0 寄存器写入更小的值，16 位计数器可能会溢出。



当计数到大于 D_2 小于 D_1 时如果 TQ0CCR0 寄存器的值从 D_1 变为 D_2 ，改写 TQ0CCR0 寄存器后，此值立即传送到 CCR0 缓冲寄存器。因此，16 位计数器的计数值将与 D_2 进行比较。
因为计数值已经超过了 D_2 ，因此 16 位计数器会继续计数到 FFFFH 并产生溢出，然后从 0000H 开始再次计数。在重新计数过程中，当计数值与 D_2 匹配时，产生 INTTQ0CC0 信号，TOQ00 引脚的输出信号的电平反转。
因此，可能不会在最初预期的“ $(D_1 + 1) \times \text{计数时钟周期}$ ”或“ $(D_2 + 1) \times \text{计数时钟周期}$ ”间隔时间产生 INTTQ0CC0 信号，而是在“ $(10000H + D_2 + 1) \times \text{计数时钟周期}$ ”的间隔时间产生。

(d) 操作 TQ0CCR1 至 TQ0CCR3 寄存器

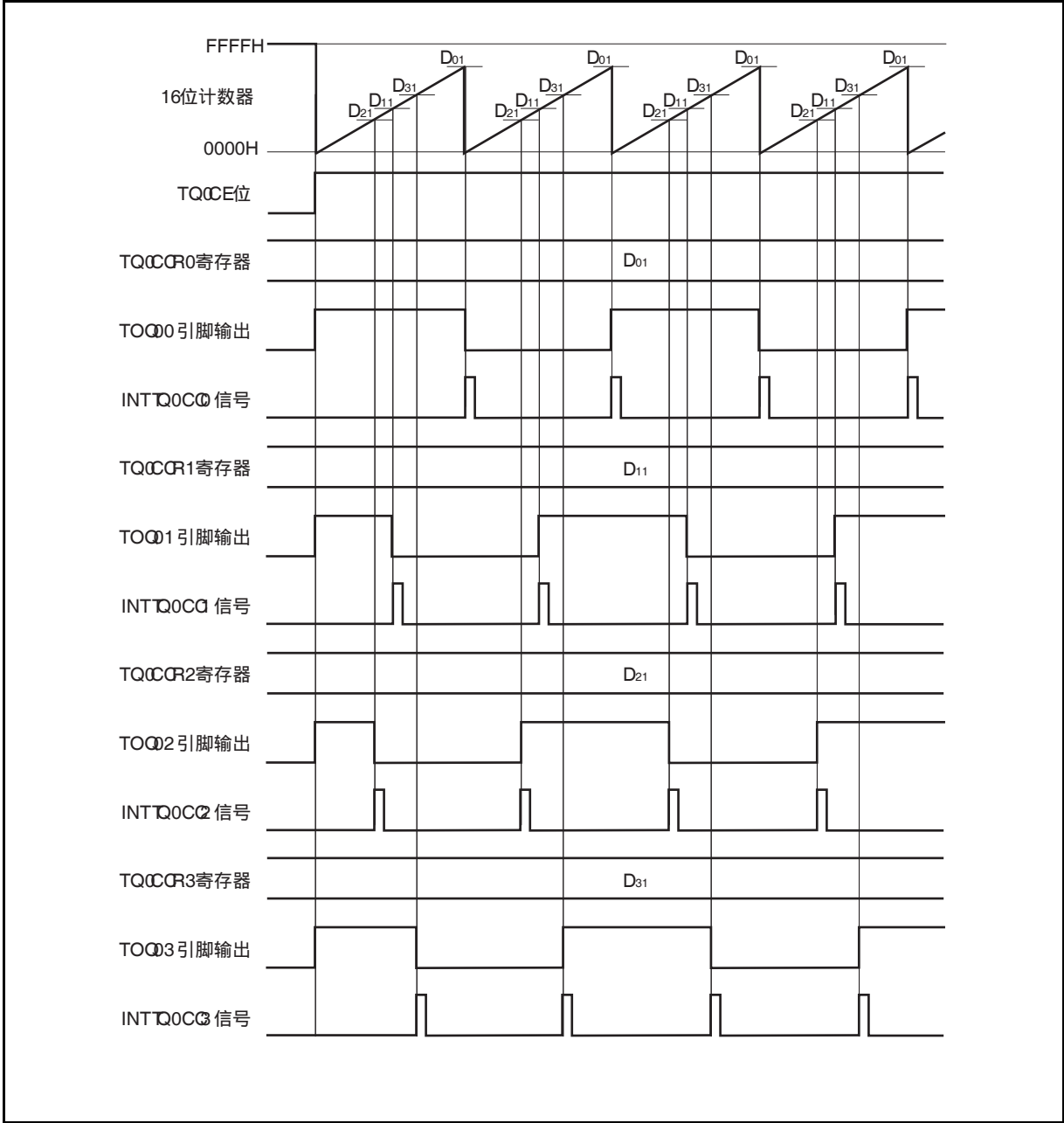
图 8-6. TQ0CCR1 至 TQ0CCR3 寄存器的构成



如果 TQ0CCRk 寄存器的设置值小于 TQ0CCR0 寄存器的设置值，则每个周期产生一次 INTTQ0CCk 信号。同时，TOQ0k 引脚的输出信号的电平反转。
TOQ0k 引脚输出的方波周期与 TOQ00 引脚的方波周期相同。

备注 k = 1 至 3

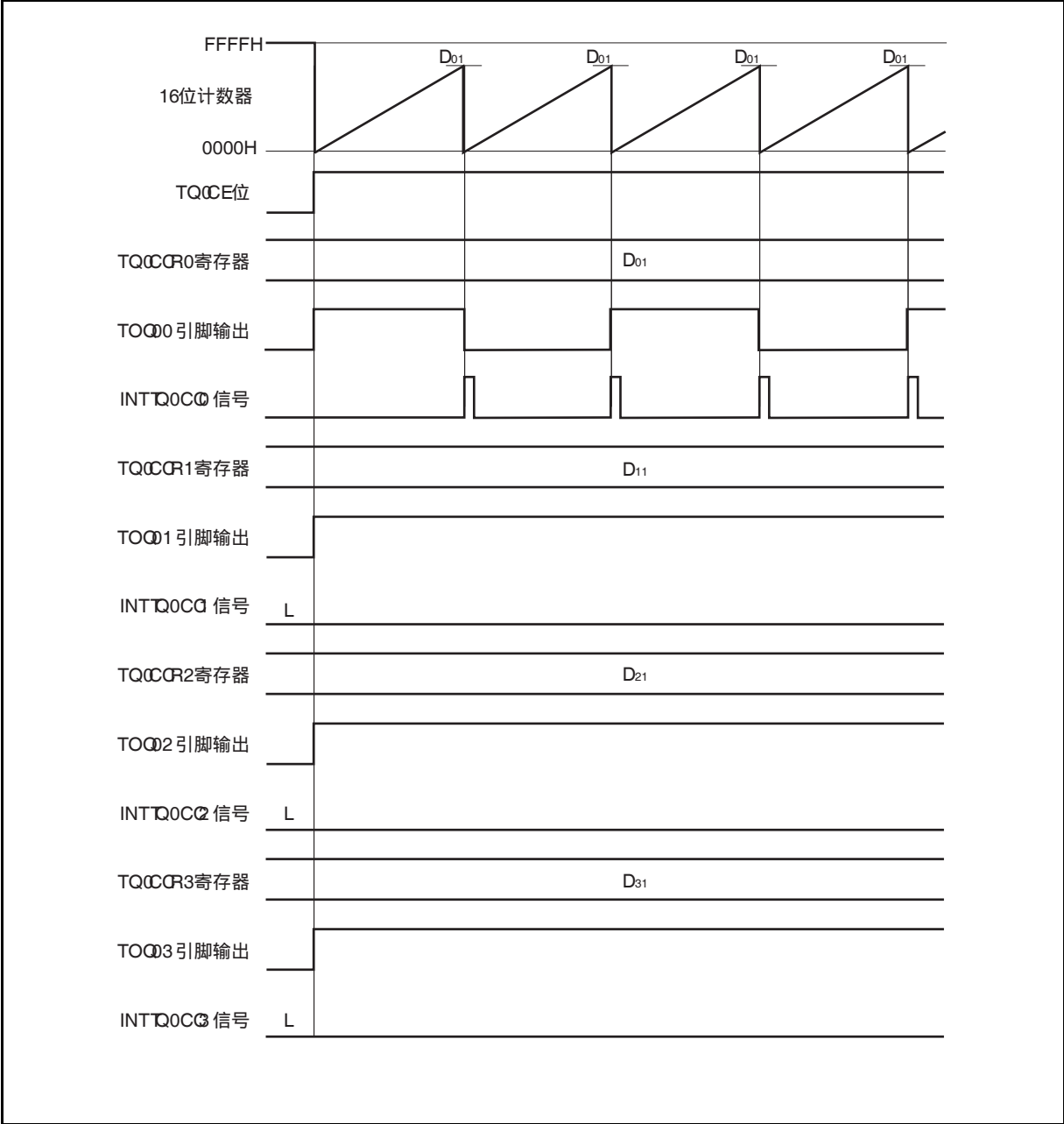
图 8-7. 当 D01 ≥ Dk1 时的时序图



如果 TQ0CCRk 寄存器的设置值大于 TQ0CCR0 寄存器，16 位计数器的计数值不会与 TQ0CCRk 寄存器匹配。因此，不产生 INTTQ0CCk 信号，TOQ0k 引脚的输出也不发生变化。

备注 k = 1 至 3

图 8-8. 当 D01 < Dk1 时的时序图



8.5.2 外部事件计数模式 (TQ0MD2 至 TQ0MD0 位 = 001)

在外部事件计数模式，只有当 TQ0CTL0.TQ0CE 位置 1 时外部事件计数输入的有效沿才有效，每次计到指定数量的有效沿后都会产生中断请求信号 (INTTQ0CC0)。该模式下不能使用 TOQ00 引脚。

通常，TQ0CCR1 至 TQ0CCR3 寄存器不用于外部事件计数模式。

图 8-9. 外部事件计数模式结构

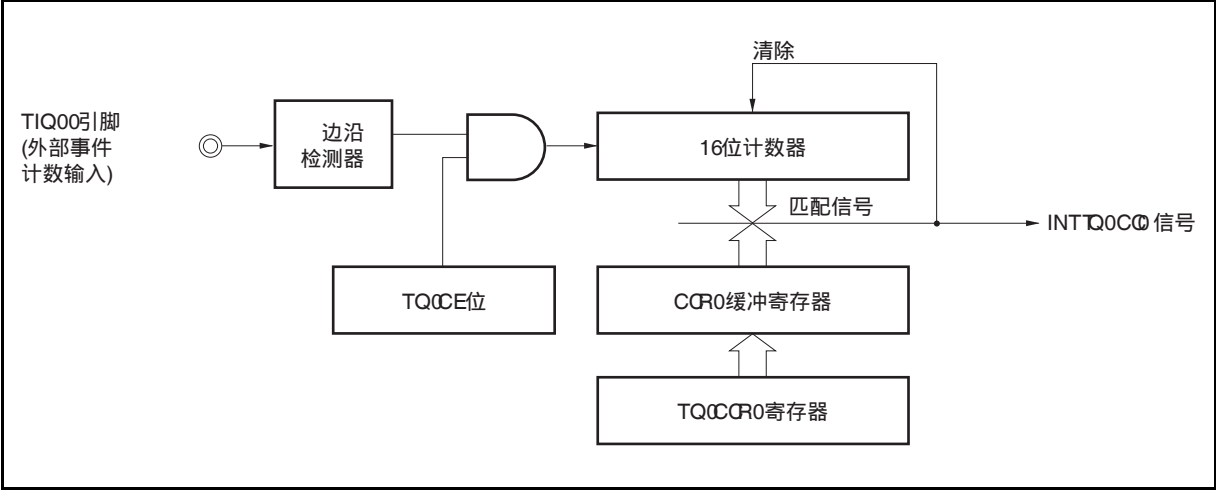
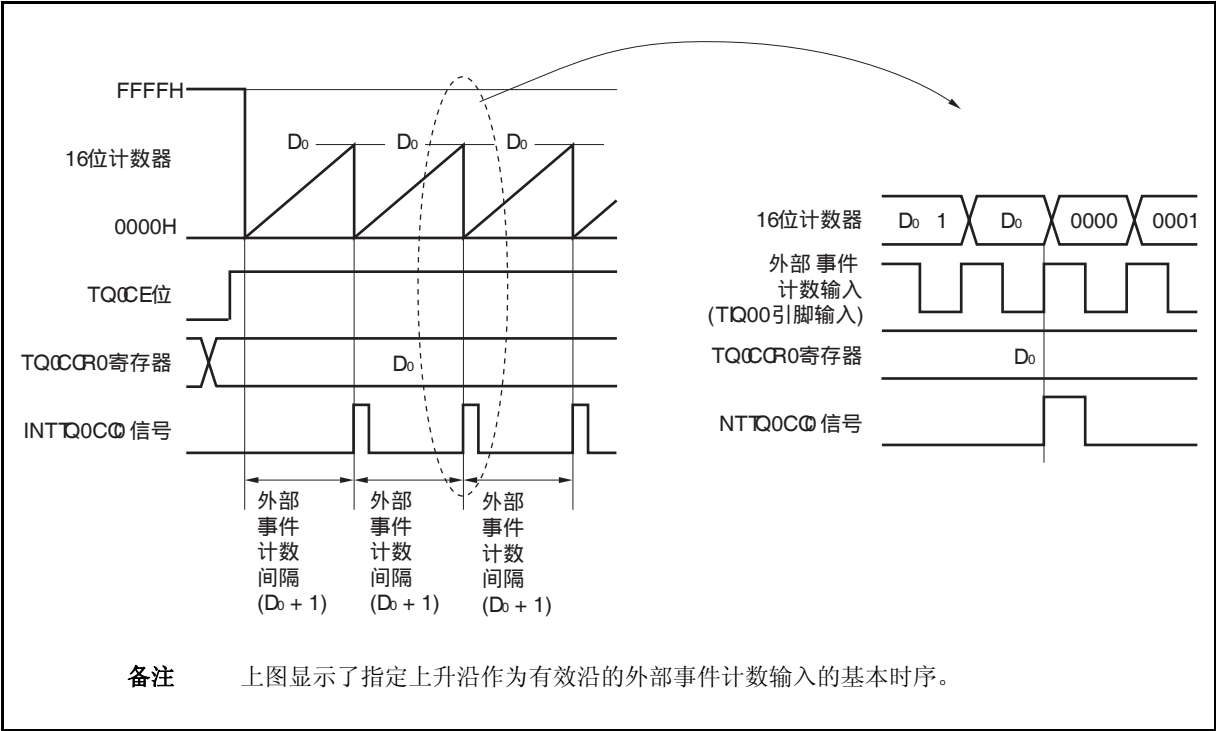


图 8-10. 外部事件计数模式的基本时序



当 TQ0CE 位置 1 时，16 位计数器的值从 FFFFH 清除为 0000H。每次外部事件计数输入的有效沿时计数器计数。此外，TQ0CCR0 寄存器的设置值发送到 CCR0 缓冲寄存器。

当 16 位计数器的计数值与 CCR0 缓冲寄存器匹配时，16 位计数器被清为 0000H，且产生比较匹配中断请求信号 (INTTQ0CC0)。

每次检测到外部事件计数输入的有效沿时(TQ0CCR0 寄存器的设置值+ 1)，都会产生 INTTQ0CC0 信号。

图 8-11. 外部事件计数模式操作的寄存器设置(1/2)

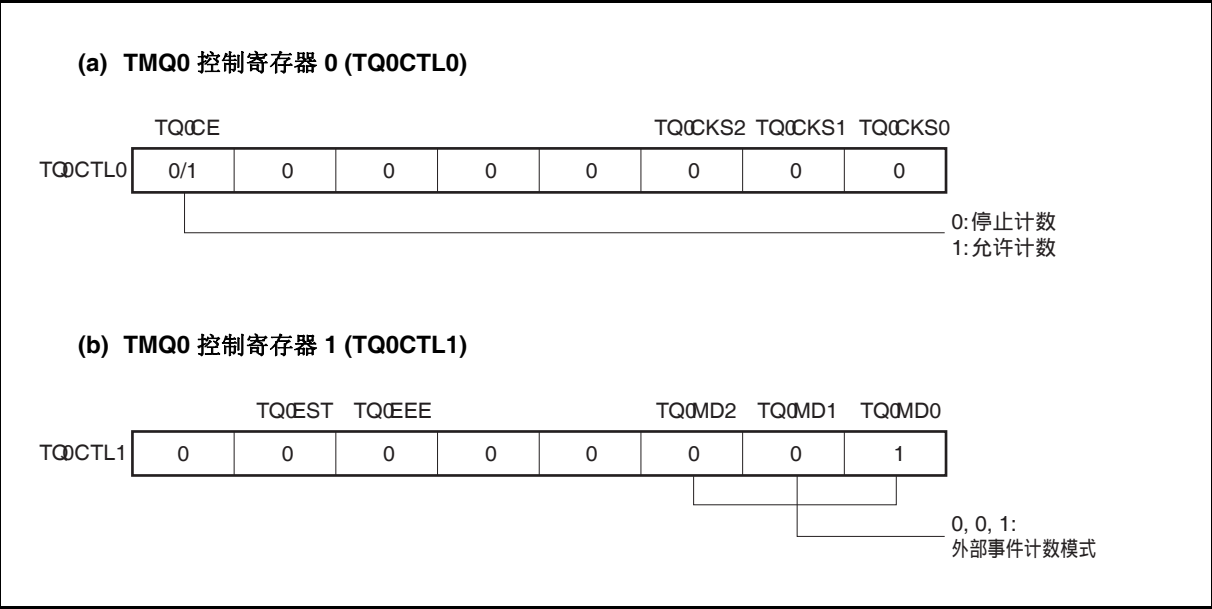
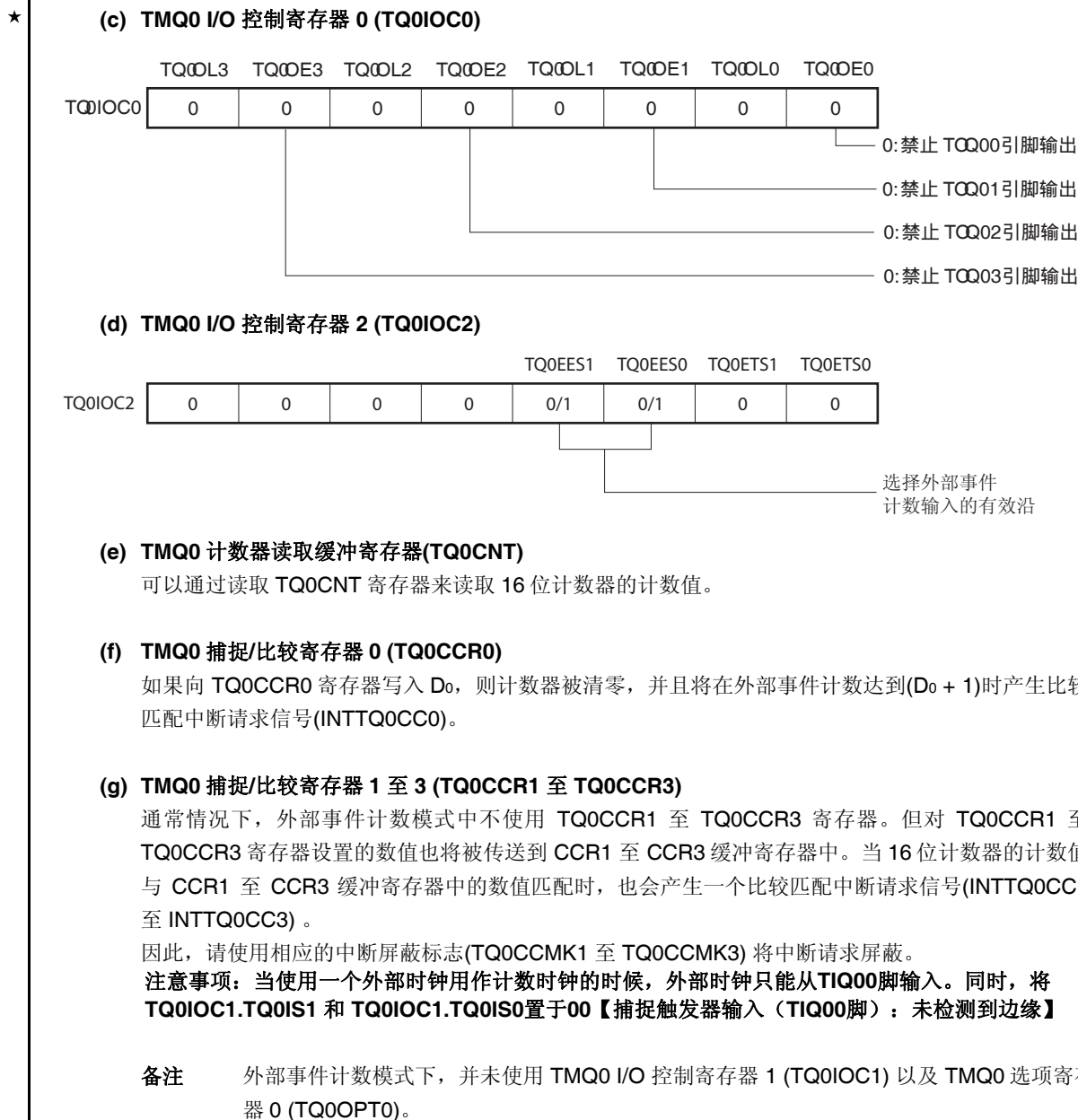
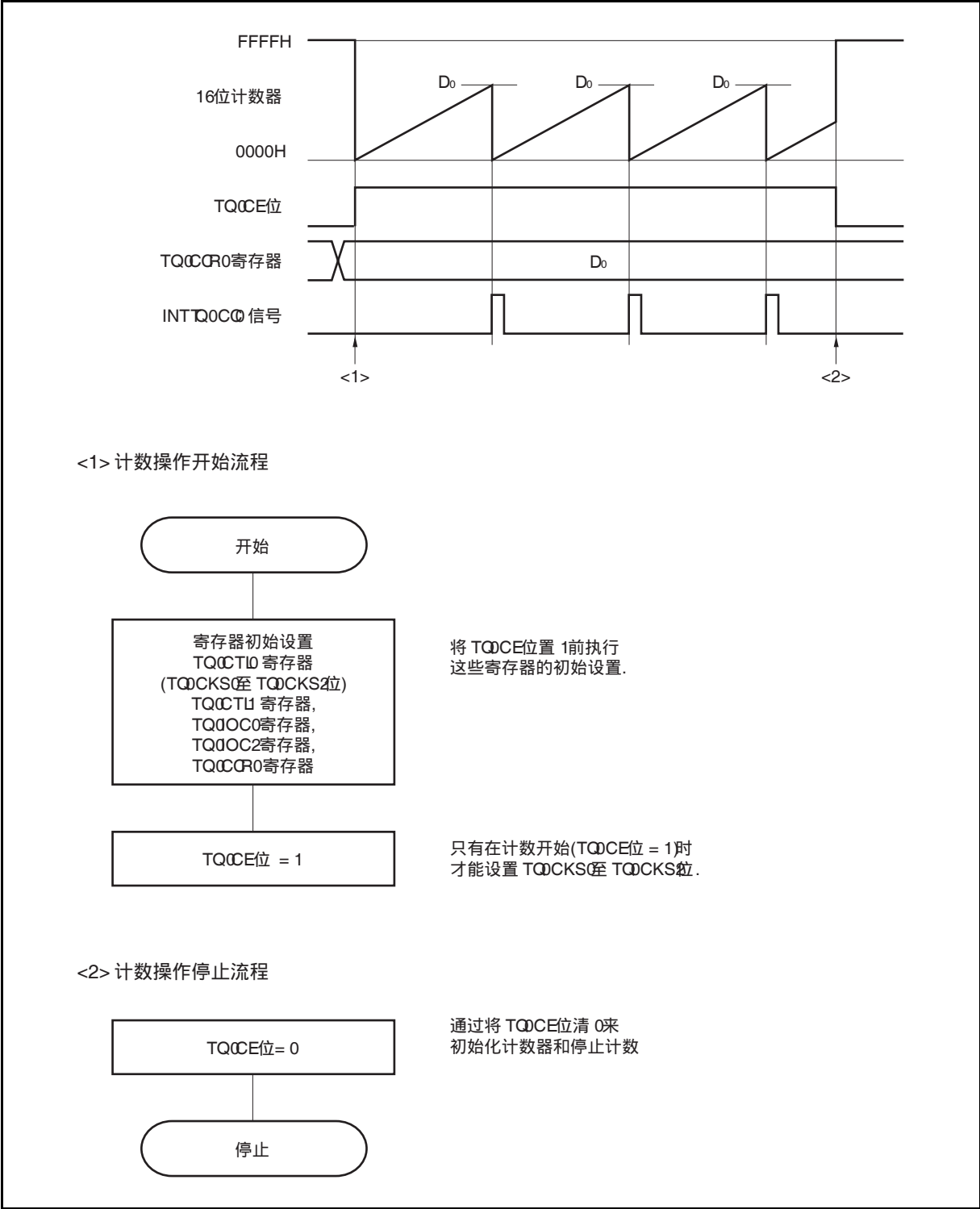


图 8-11. 外部事件计数模式操作的寄存器设置(2/2)



(1) 外部事件计数模式操作流程

图 8-12. 外部事件计数模式的软件处理流程



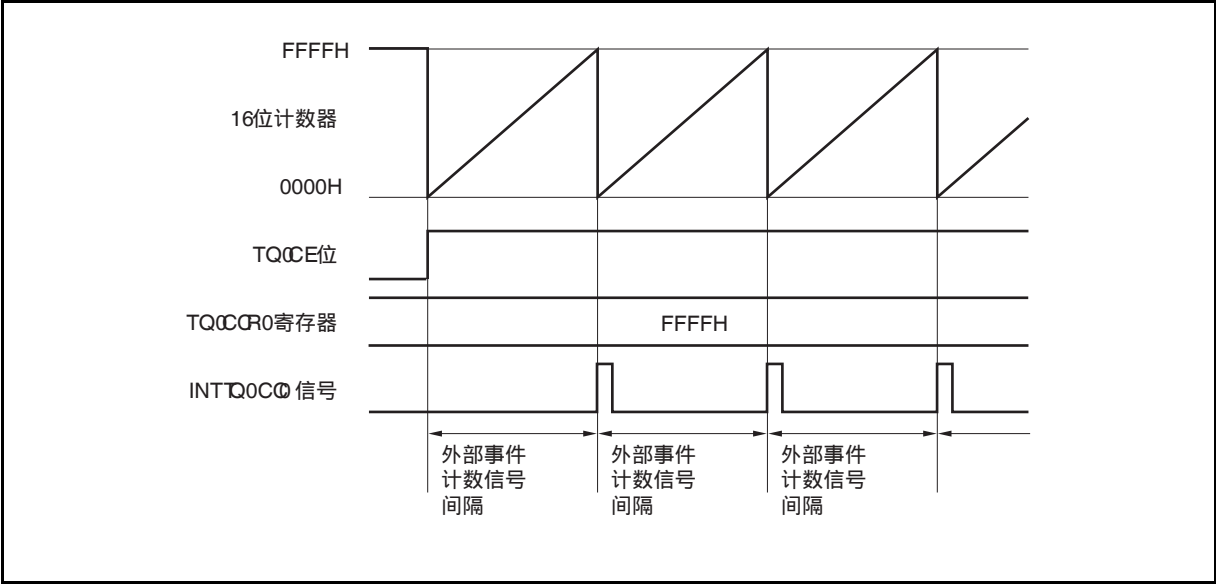
(2) 外部事件计数模式的操作时序

- ★
- 注意事项
1. 在外部事件计数模式中，不要将 TQ0CCR0 寄存器设置为 0000H。

2. 在外部事件计数模式中，禁止使用定时器输出功能。如果要将外部事件计数输入作为定时器输出使用，那么请将工作模式设置为间隔定时器模式并将外部事件计数输入设定为计数时钟 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 000, TQ0CTL1.TQ0EEE 位 = 1)。

(a) TPnCCR0 寄存器被设置为 FFFFH 时的操作

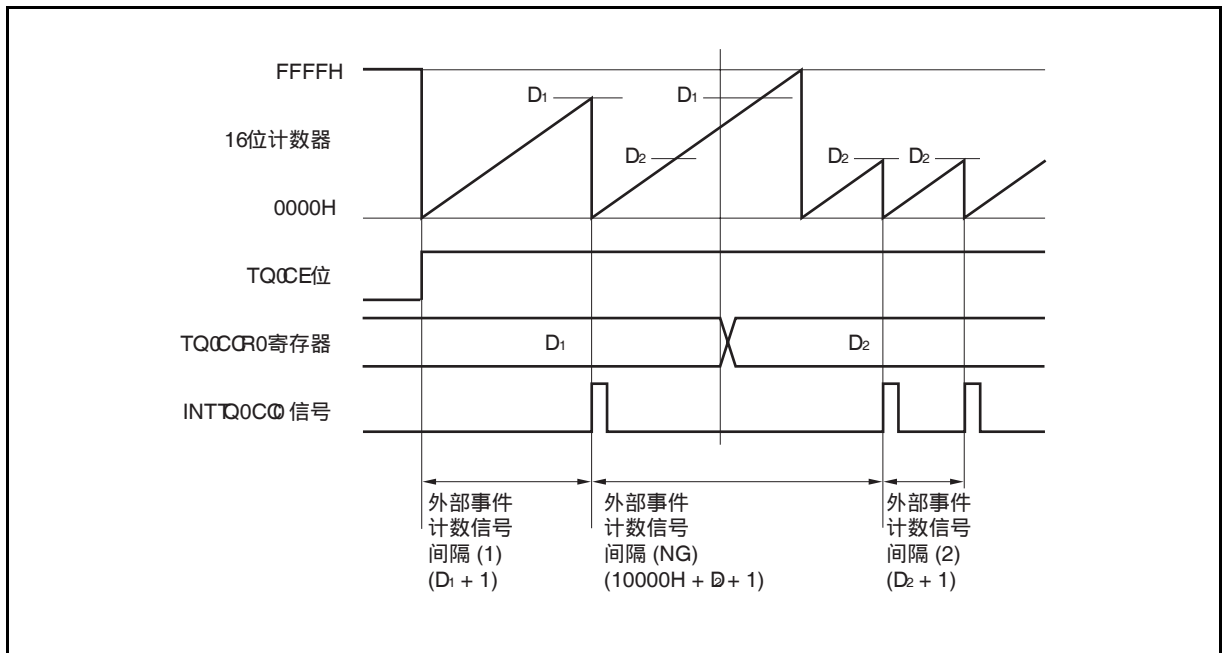
如果 TQ0CCR0 寄存器被设置为 FFFFH，那么 16 位计数器将在每次检测到外部事件计数信号的有效沿处计数，直到 FFFFH。并在下一个计数事件时刻被同步清零，同时产生 INTTQ0CC0 信号。此时，溢出标志位(TQ0OPT0.TQ0OVF 位)不会被置 1。



(b) 改写 TQ0CCR0 寄存器时需要注意的事项

将 TQ0CCR0 寄存器的值改小时，要先停止计数，再对设定值进行改变。

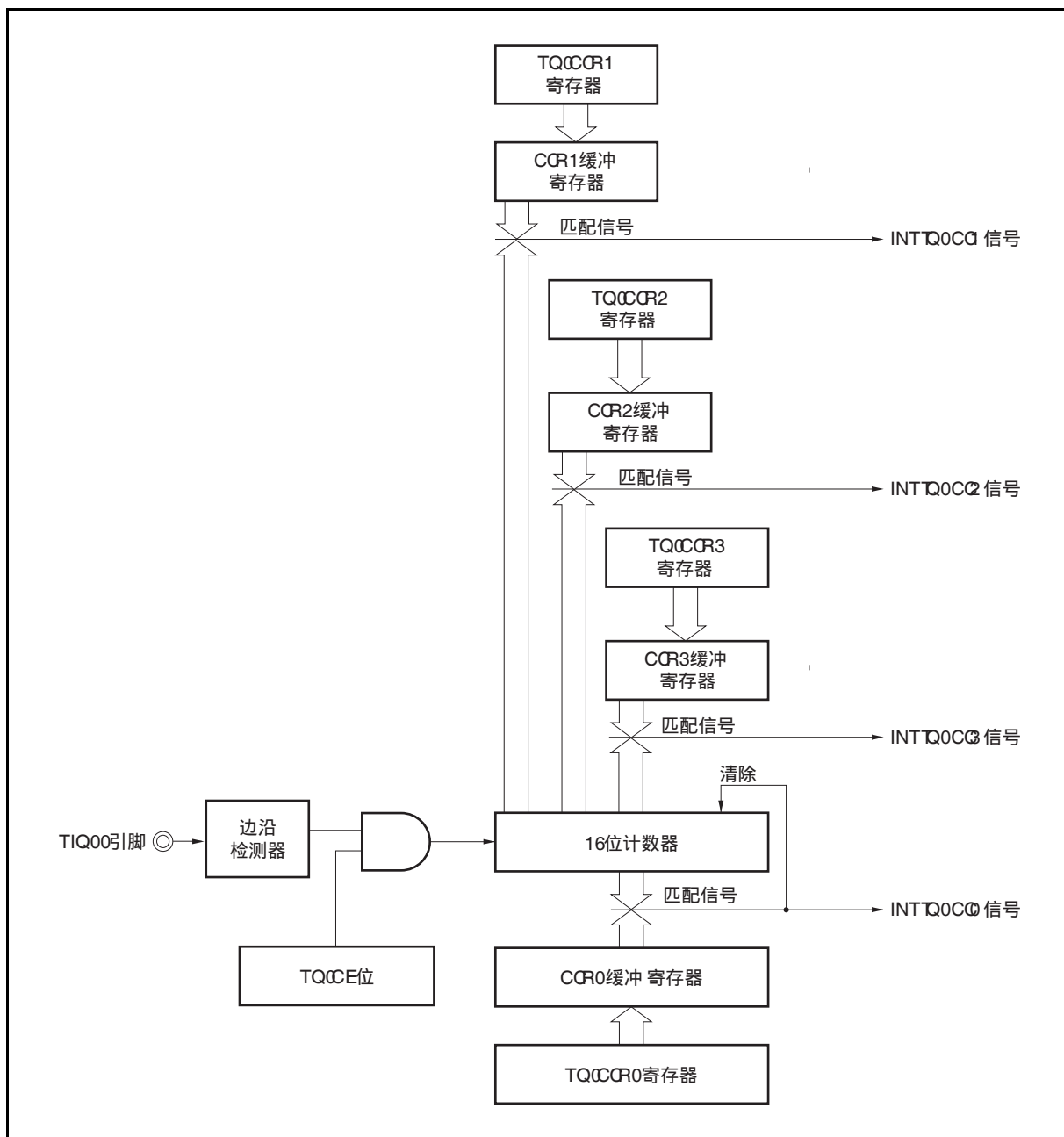
如果在计数过程中将 TQ0CCR0 寄存器的值改小，那么 16 位计数器可能产生溢出。



如果当前计数值大于 D_2 但小于 D_1 ，将 TQ0CCR0 寄存器的值从 D_1 改变为 D_2 ，新的设定值将在改写后立即从 TQ0CCR0 寄存器传送到 CCR0 缓冲寄存器中，那么，16 位计数器的计数值将与 D_2 进行比较。由于计数值已经超过了 D_2 ，因此 16 位计数器会继续计数到 FFFFH 并产生溢出，之后重新从 0000H 开始计数。在重新计数过程中，计数值与 D_2 匹配时，才会产生 INTTQ0CC0 信号。因此可见，INTTQ0CC0 信号不会在预期的事件计数次数 “ $(D_1 + 1)$ 次” 或 “ $(D_2 + 1)$ 次” 时产生，而是在一个 “ $(10000H + D_2 + 1)$ 次” 的计数次数时产生。

★ (c) TQ0CCR1 至 TQ0CCR3 寄存器的操作

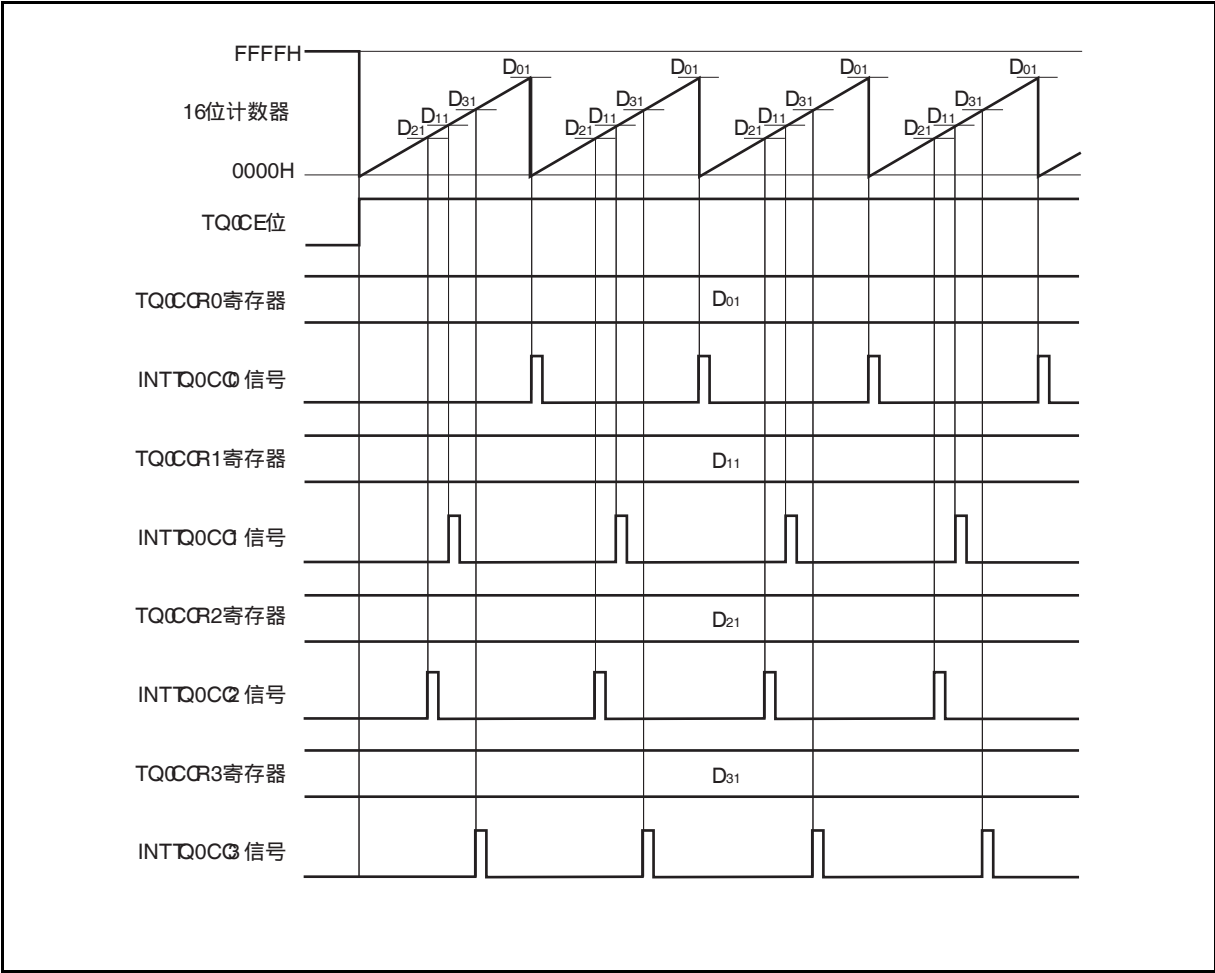
图 8-13. TQ0CCR1 至 TQ0CCR3 寄存器的配置



如果 TQ0CCRk 寄存器的值小于 TQ0CCR0 寄存器的值，那么 INTTQ0CCK 信号将每周期产生一次。

备注 k = 1 至 3

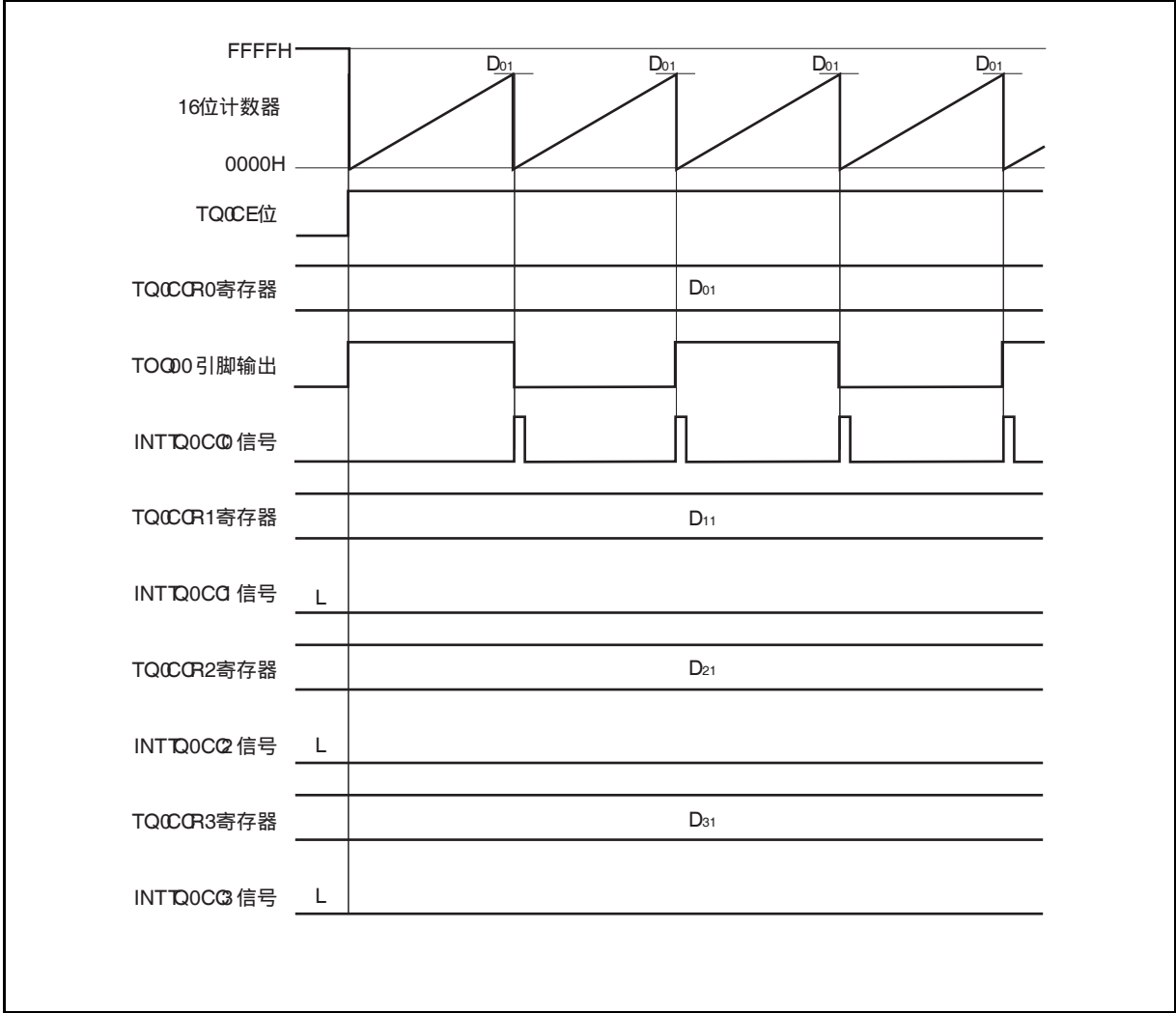
图 8-14. $D_{01} \geq D_{k1}$ 时的时序图



如果 TQ0CCRk 寄存器的值大于 TQ0CCR0 寄存器的值，那么 16 位计数器的计数值不会与 TQ0CCRk 寄存器中的值相匹配。因此，不会产生 INTTQ0CCk 信号。

备注 k = 1 至 3

图 8-15. D₀₁ < D_{k1} 时的时序图



8.5.3 外部触发脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 010)

在外部触发脉冲输出模式中, TQ0CTL0.TQ0CE 位被置 1 后, 16 位定时器/事件计数器 Q 进入等待触发信号的状态。当检测到一个外部触发输入信号的有效沿时, 16 位定时器/事件计数器 Q 开始计数, 并由 TOQ01 至 TOQ03 引脚输出 PWM 波形。

也可以使用软件触发代替外部触发来开始脉冲的输出。当使用软件触发时, 还可由 TOQ00 引脚输出一个以 PWM 波形周期的 2 倍为周期的方波。

图 8-16. 外部触发脉冲输出模式的配置

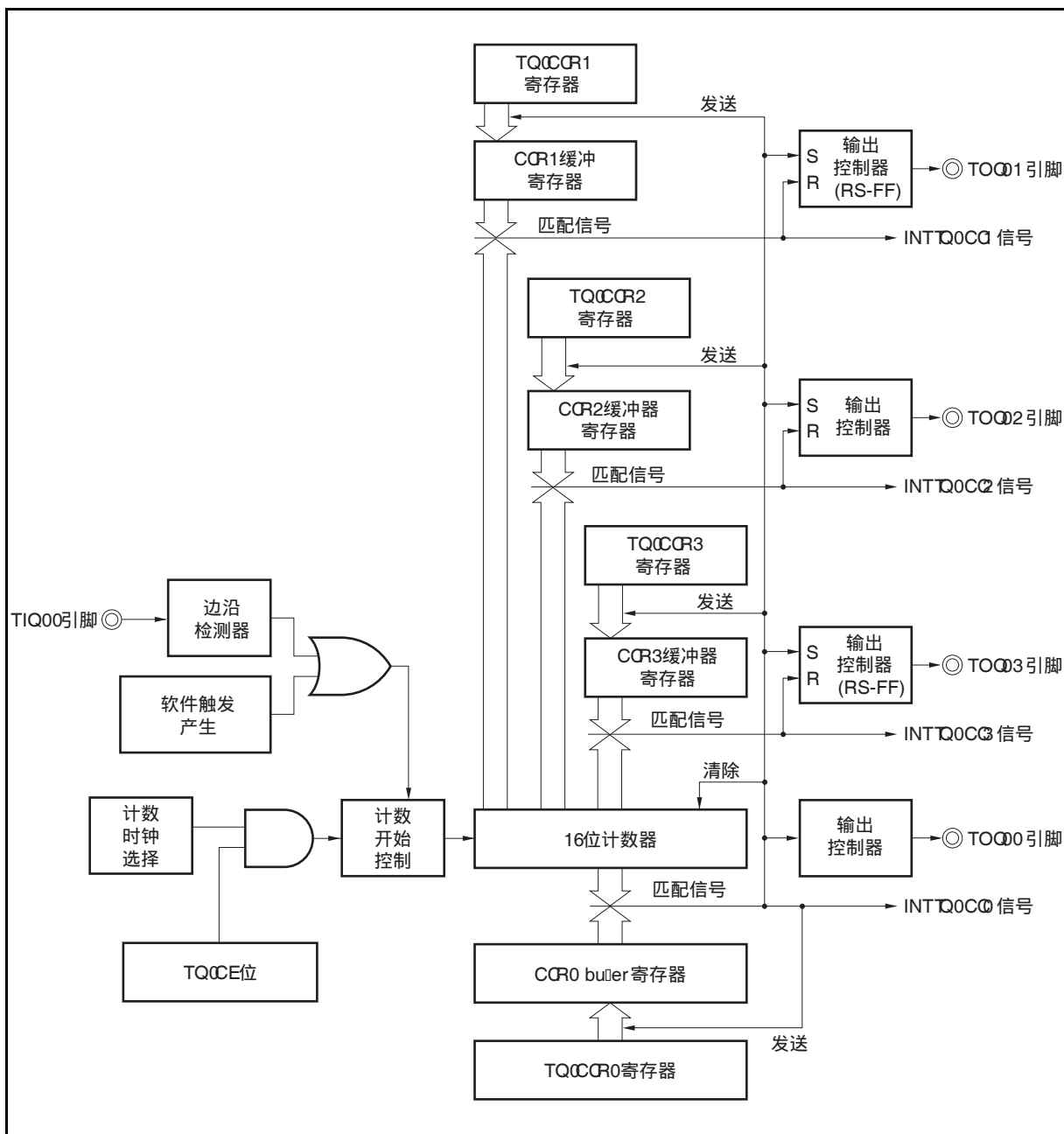
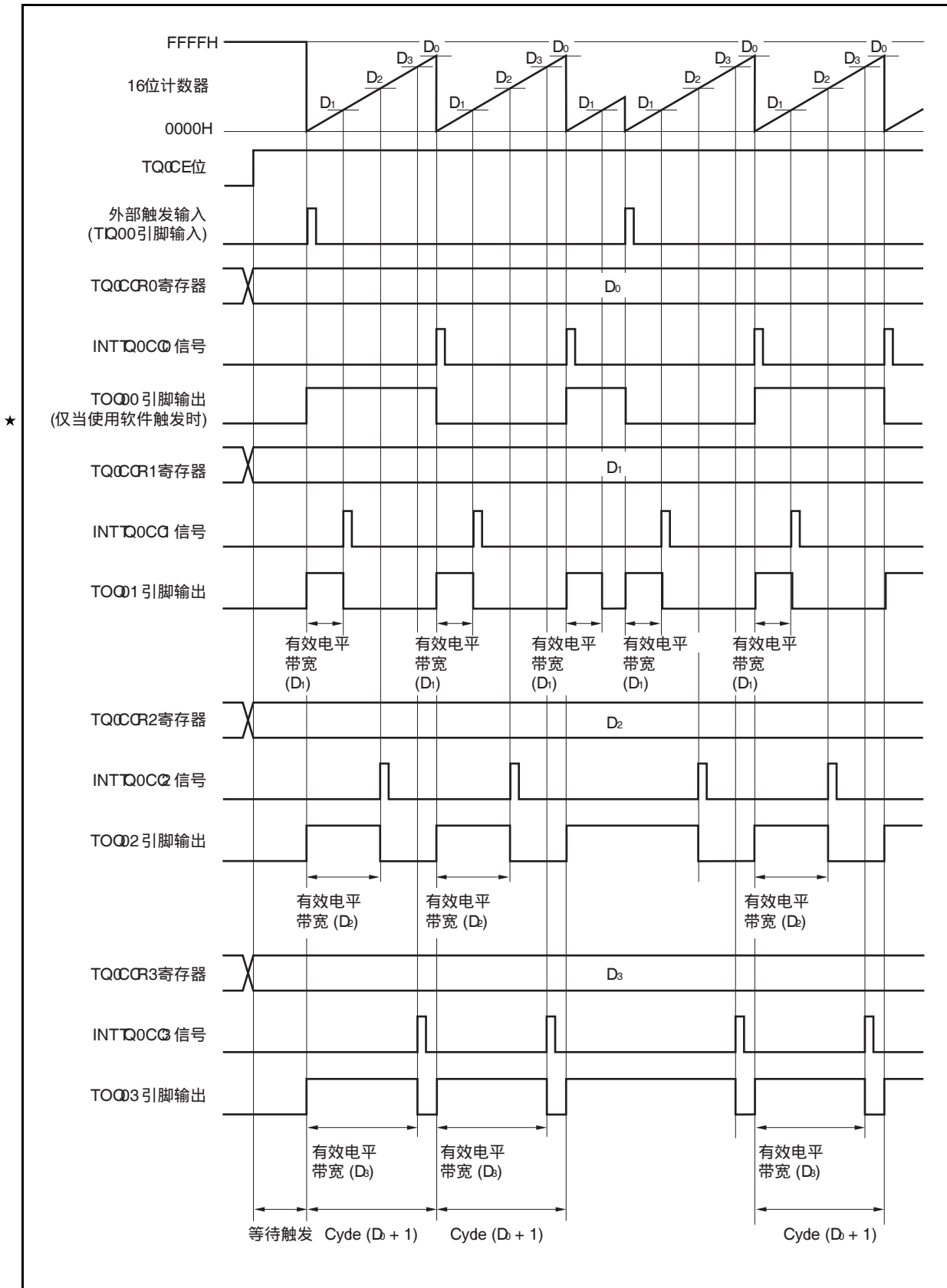


图 8-17. 外部触发脉冲输出模式的寄存器设置



TQ0CE 位被置 1 后, 16 位定时器/事件计数器 Q 进入等待触发信号的状态。触发产生后, 16 位计数器从 FFFFH 清零为 0000H, 同时开始计数并从 TOQ0k 引脚输出一个 PWM 波形。如果在计数器计数过程中, 再次触发信号, 那么计数器将被清为 0000H 并重新开始计数。(TOQ00 引脚的输出电平反转。TOQ0k 在触发产生时输出高电平(无论当前状态为高/低)。)

PWM 波形的有效电平宽度，周期以及占空比系数可由以下的算式得出。

有效电平宽度 = (TQ0CCRk 寄存器的设置值) × 计数时钟周期
周期 = (TQ0CCR0 寄存器的值+ 1) × 计数时钟周期
占空比系数 = (TQ0CCRk 寄存器的值)/(TQ0CCR0 寄存器的值+ 1)

比较匹配中断请求信号 INTTQ0CC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器中的值相匹配后产生。，同时 16 位计数器也会被清零。比较匹配中断请求信号 INTTQ0CCk 会在 16 位计数器的计数值与 CCRk 缓冲寄存器中的值相匹配时产生。

当 16 位计数器的计数值与 CCRm 缓冲寄存器中的值相匹配时，TQ0CCRm 寄存器中设置的值会被传送至 CCRm 缓冲寄存器，同时 16 位计数器被清为 0000H。

触发源可以从外部触发输入信号的有效沿和设置软件触发(TQ0CTL1.TQ0EST 位 = 1)的方式中进行选择。

备注 k = 1 至 3
 m = 0 至 3

图 8-18. 外部触发脉冲输出模式的寄存器设置(1/3)

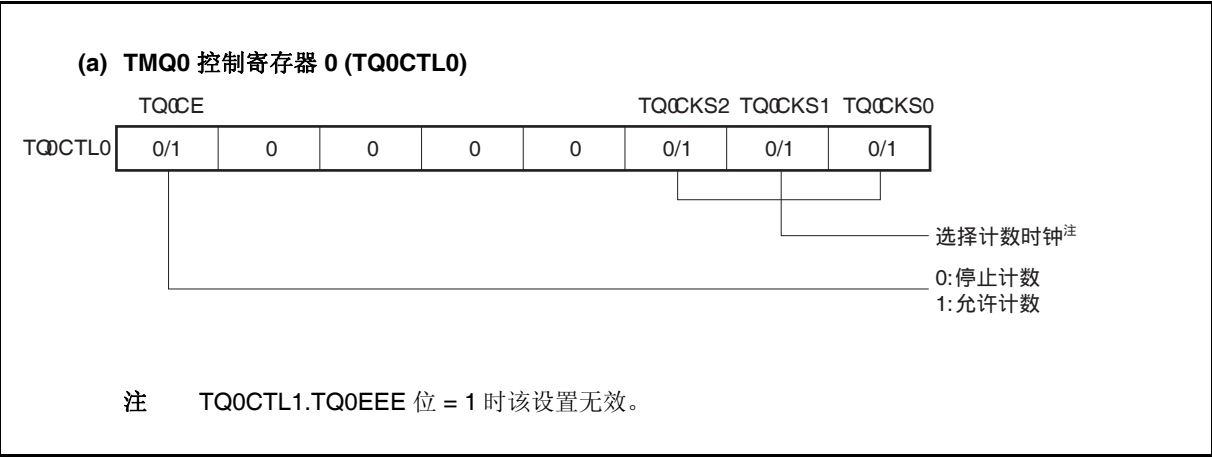
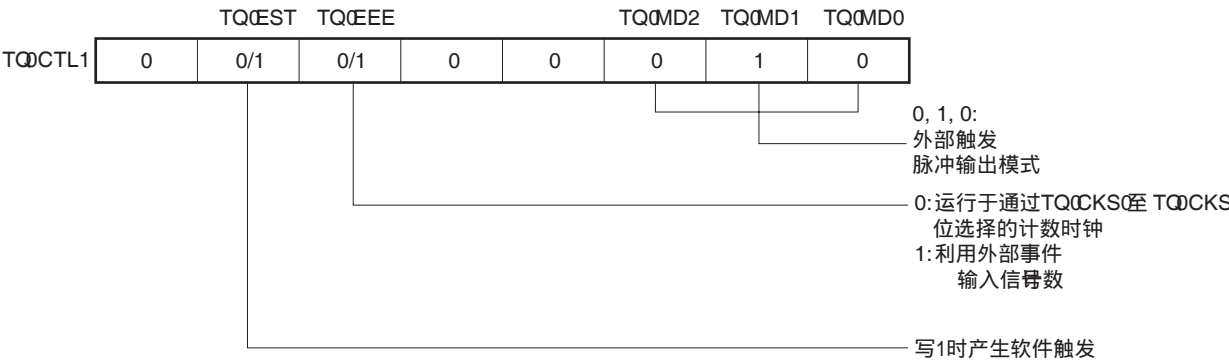
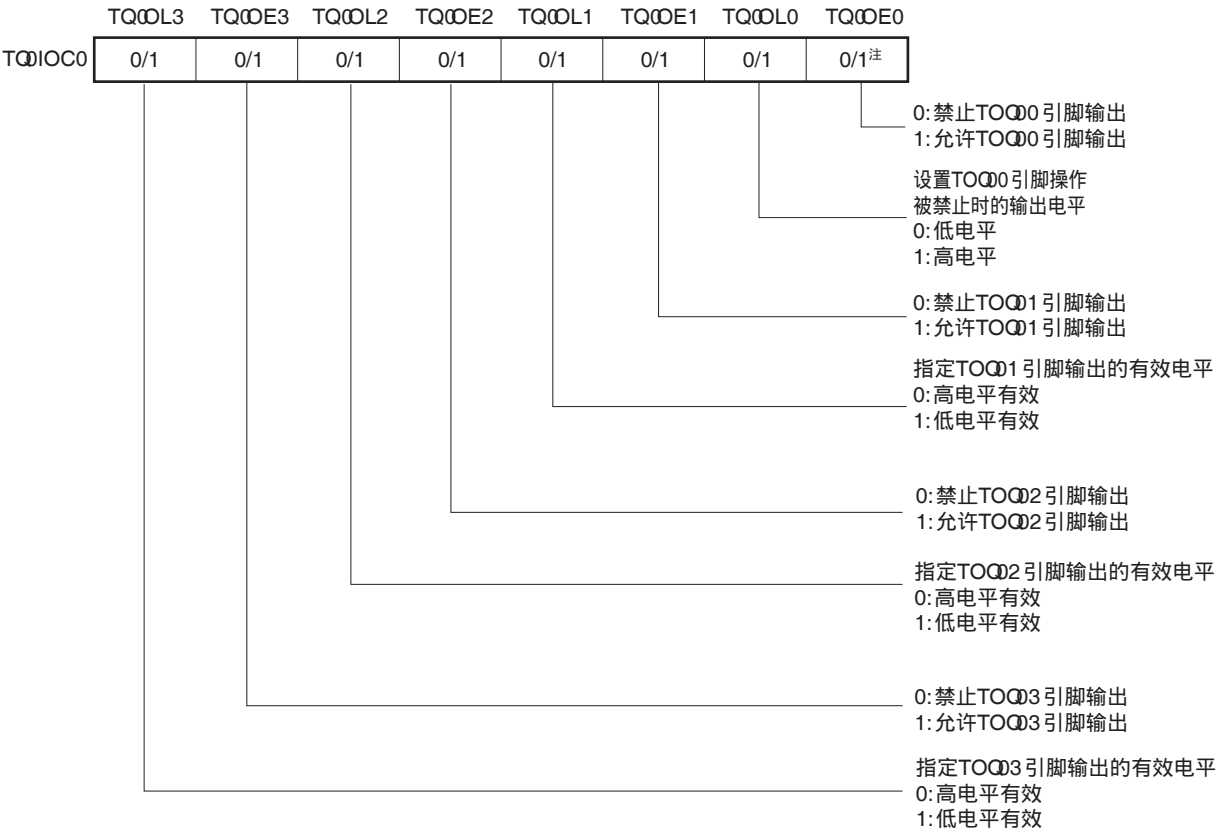


图 8-18. 外部触发脉冲输出模式的寄存器设置(2/3)

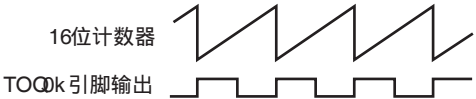
(b) TMQ0 控制寄存器 1 (TQ0CTL1)



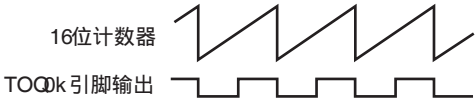
(c) TMQ0 I/O 控制寄存器 0 (TQ0IOC0)



当 TQ0Lk位 = 0时



当 TQ0Lk位 = 1时



★ 注 当 TQ00 引脚不用于外部触发脉冲输出模式时将此位清零。

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)

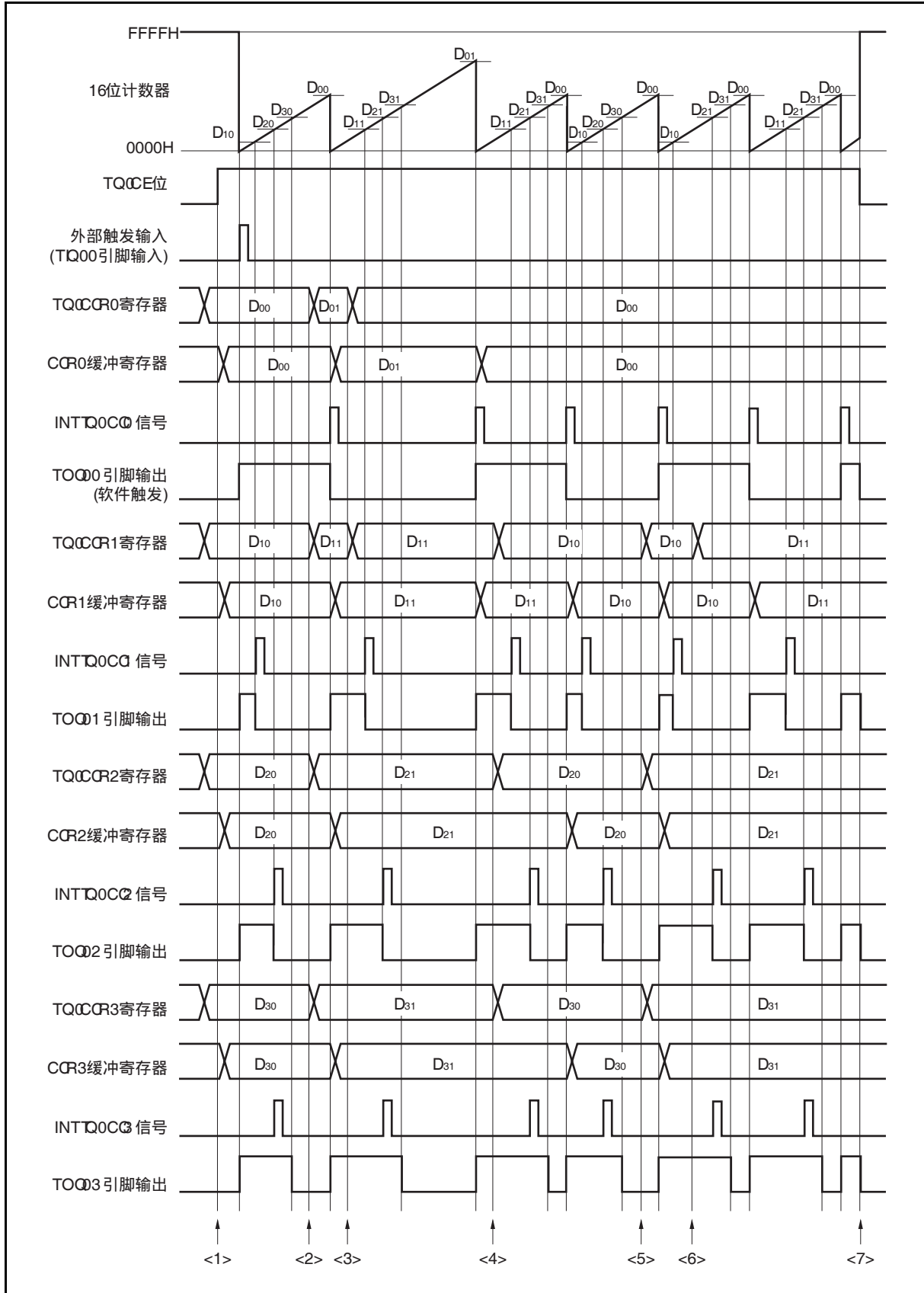
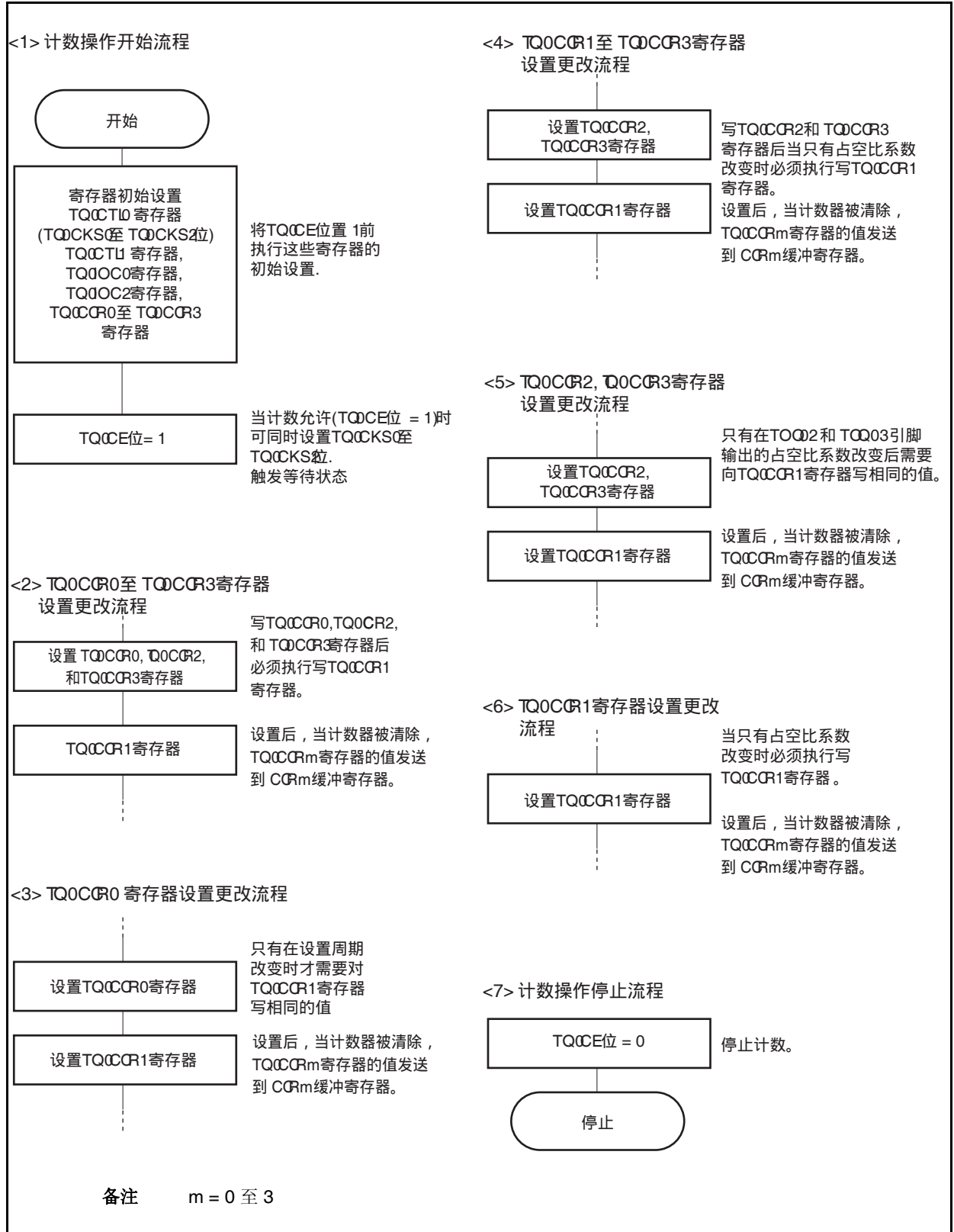


图 8-19. 外部触发脉冲输出模式的软件处理流程(2/2)

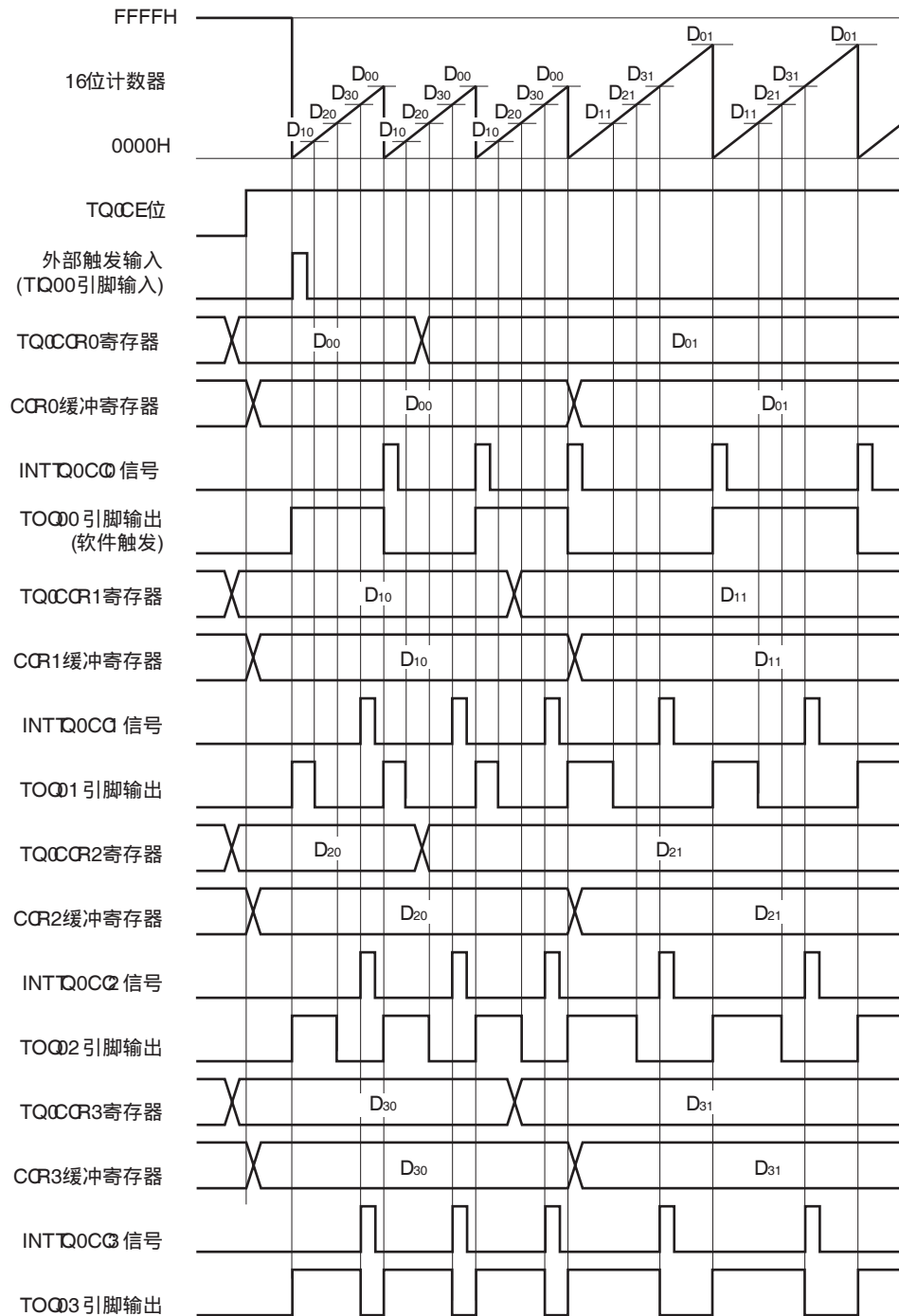


(2) 外部触发脉冲输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

计数器计数过程中，若改变 PWM 波形，则要在最后进行对 TQ0CCR1 寄存器的写入操作。

对 TQ0CCR1 寄存器进行写入操作之后，若需再次更改 TQ0CCRk 寄存器的值，须等待下一个 INTTQ0CC0 信号被检测到后，再进行相关操作。



若要将 TQ0CCRm 寄存器中设置的数据传送至 CCRm 缓冲寄存器，必须对 TQ0CCR1 寄存器进行写入操作。

同时改变 PWM 波形的周期和有效电平时，先将周期值写入 TQ0CCR0 寄存器，再将有效电平宽度值写入 TQ0CCR2 和 TQ0CCR3 寄存器，最后将有效电平写入 TQ0CCR1 寄存器。

只改变 PWM 波形的周期时，先将周期值写入 TQ0CCR0 寄存器，再向 TQ0CCR1 寄存器写入与原来相同的值。

只改变 PWM 波形的有效电平宽度(占空比系数)时，首先需要向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平值，然后再向 TQ0CCR1 寄存器设置有效电平。

若只改变 TOQ01 引脚输出的 PWM 波形的有效电平宽度(占空比系数)，只需要设置 TQ0CCR1 寄存器。

若只改变 TOQ02 和 TOQ03 引脚输出的 PWM 波形的有效电平宽度(占空比系数)，首先需要向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平宽度，然后向 TQ0CCR1 寄存器写入相同的值。

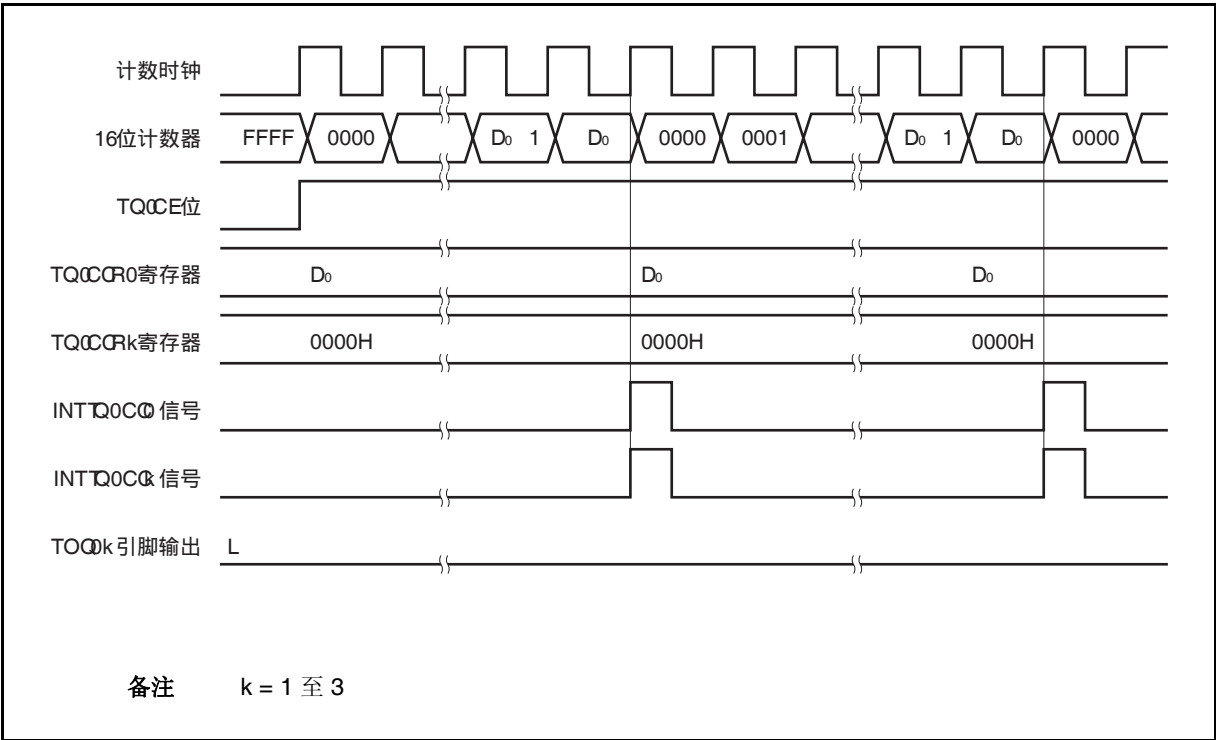
数据被写入 TQ0CCR1 寄存器之后，当 16 位计数器被清零时，TQ0CCRm 寄存器中的数值才会被传送至 CCRm 缓冲寄存器中。此时 CCRm 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

向 TQ0CCR1 寄存器的写入操作完成后，若需马上再次改写 TQ0CCR0 至 TQ0CCR3 寄存器的值，则要在产生 INTTQ0CC0 信号之后再进行改写。否则，由于数据从 TQ0CCRm 寄存器传送至 CCRm 缓冲寄存器的时序与 TQ0CCRm 寄存器的写入时序相冲突，CCrm 缓冲寄存器中所设置的数值可能无法确定。

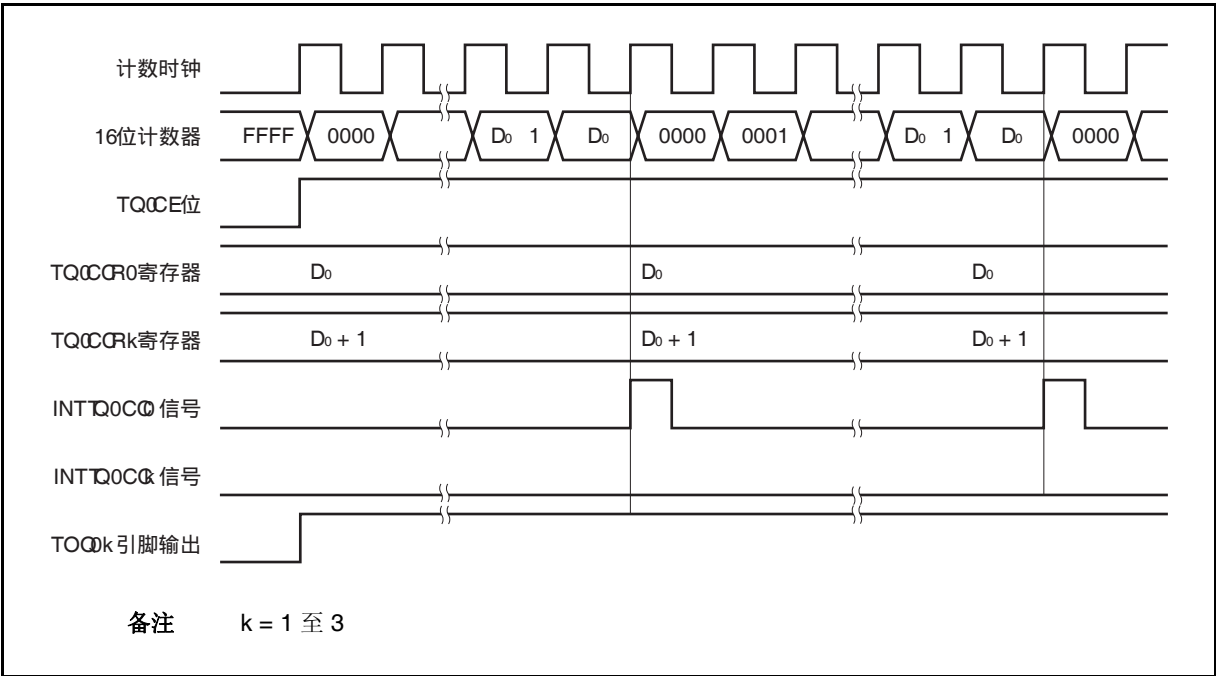
备注 m = 0 至 3

(b) PWM 波形的 0%/100% 输出

若要输出 0% 的波形，则需将 TQ0CCRk 寄存器设置为 0000H。如果 TQ0CCR0 寄存器被设置为 FFFFH，那么 INTTQ0CCK 信号将周期性产生。

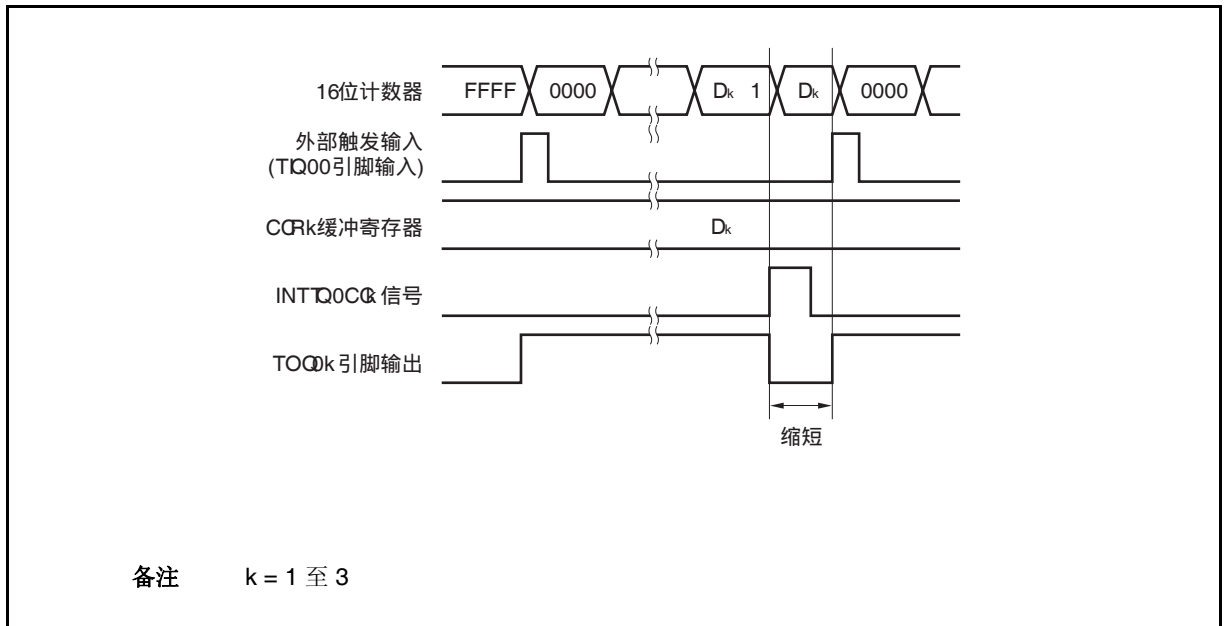


若要输出 100% 波形，则需将 TQ0CCRk 寄存器设置为 (TQ0CCR0 寄存器值 + 1)。TQ0CCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。

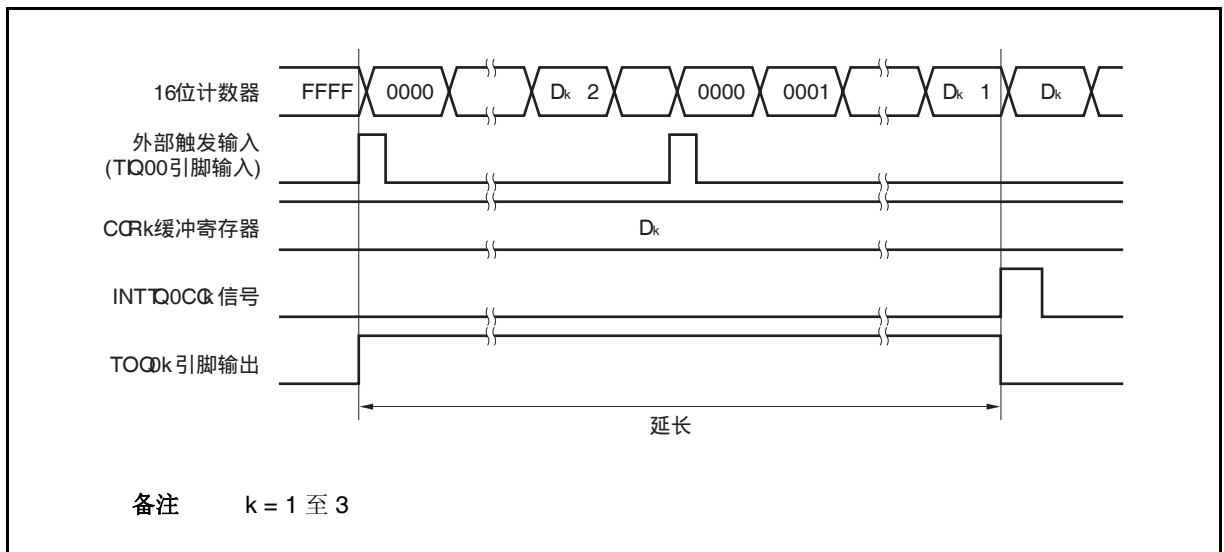


(c) 触发检测和 CCRk 缓冲寄存器匹配之间的冲突

如果触发信号紧随 INTTQ0CCK 中断信号的产生被检测，那么 16 位计数器将被立即清零并重新开始计数，TOQ0k 引脚输出也会改变为有效电平。这种情况下，PWM 波形的非有效电平时间就会被缩短。

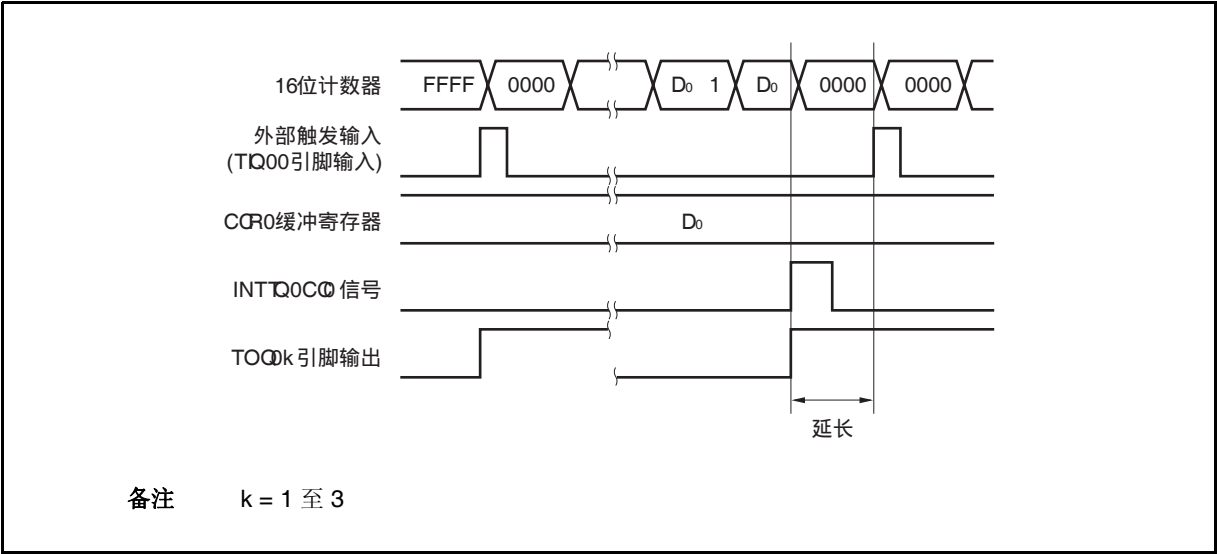


如果在 INTTQ0CCK 信号产生之前检测到触发信号，那么 INTTQ0CCK 信号将不会产生，此时，16 位计数器被清零并重新开始计数。TOQ0k 引脚的输出也会继续保持为有效电平。这种情况下，PWM 波形的有效电平时间就会被延长。

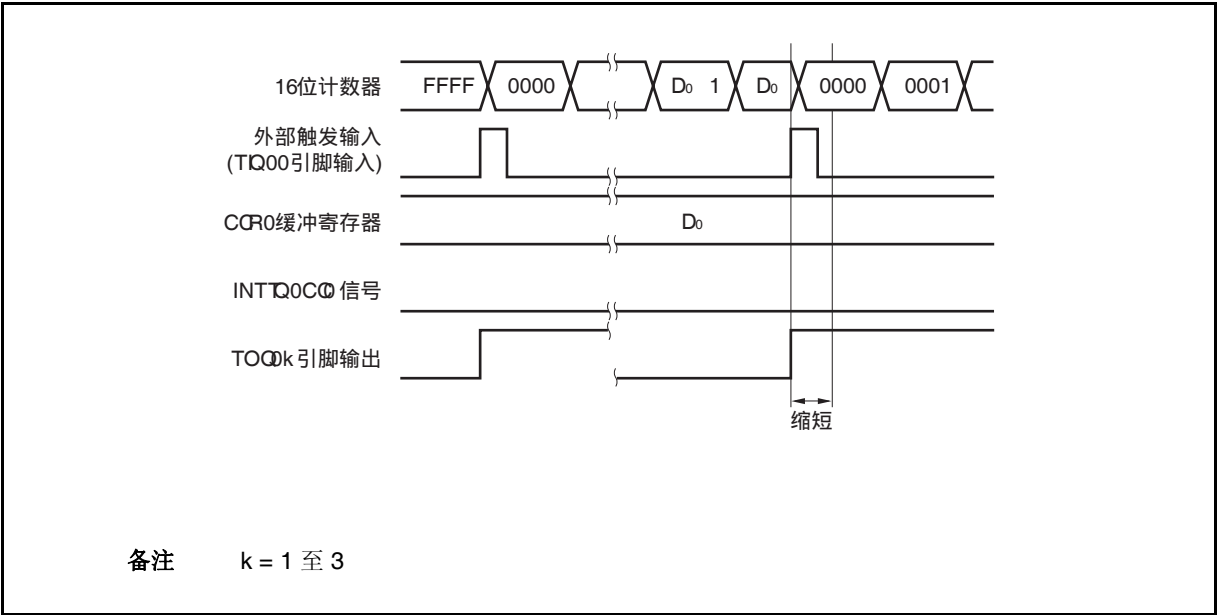


(d) 触发检测和 CCR0 缓冲寄存器匹配之间的冲突

如果触发信号紧随 INTTQ0CC0 中断信号的产生被检测，那么 16 位计数器将被清零并重新开始计数。因此，TOQ0k 引脚输出的有效电平时间就会被延长(从 INTTQ0CC0 信号产生到触发信号被检测的时间)。

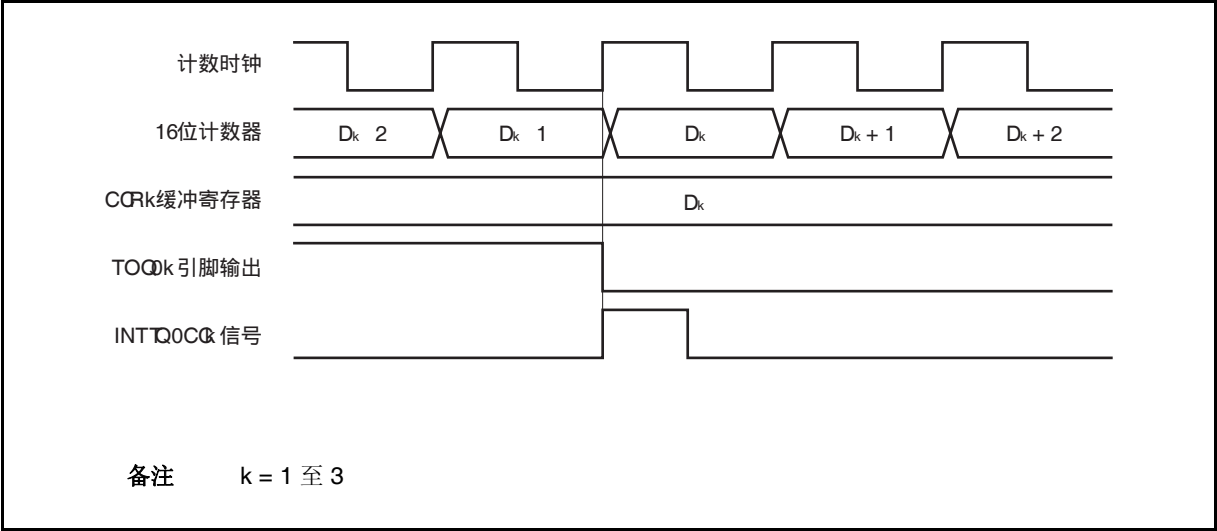


如果在 INTTQ0CC0 信号产生之前检测到触发信号，那么 INTTQ0CC0 信号将不会产生。16 位计数器将被清零并重新开始计数，TOQ0k 引脚输出也会改变为有效电平。这种情况下，PWM 波形的非有效电平时间就会被缩短。



(e) 比较匹配中断请求信号的产生时序(INTTQ0CCK)

外部触发脉冲输出模式下，INTTQ0CCK 信号的产生时序与其它模式下 INTTQ0CCK 信号的产生时序有所不同，这里，INTTQ0CCK 信号将在 16 位计数器的计数值与 CCRk 缓冲寄存器的值相匹配时产生。



通常情况下，INTTQ0CCK 信号会在 16 位计数器计数值与 CCRk 缓冲寄存器的值匹配之后的下一个计数时钟同步产生。

但在外部触发脉冲输出模式下，INTTQ0CCK 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOQ0k 引脚输出信号变化的时序而改变的。

8.5.4 单次脉冲输出模式 (TQ0MD2 到 TQ0MD0 位 = 011)

在单次脉冲输出模式中，TQ0CTL0.TQ0CE 位被置 1 后，16 位定时器/事件计数器 Q 进入等待触发信号的状态。当检测到一个外部触发输入信号的有效沿时，16 位定时器/事件计数器 Q 开始计数，并由 TOQ01 至 TOQ03 引脚输出单脉冲波形。

也可以使用软件触发代替外部触发来开始脉冲的输出。当使用软件触发时，TOQ00 引脚会在 16 位计数器计数时输出有效电平，在计数器停止计数时(等待触发时)输出非有效电平。

图 8-20. 单脉冲输出模式的配置图

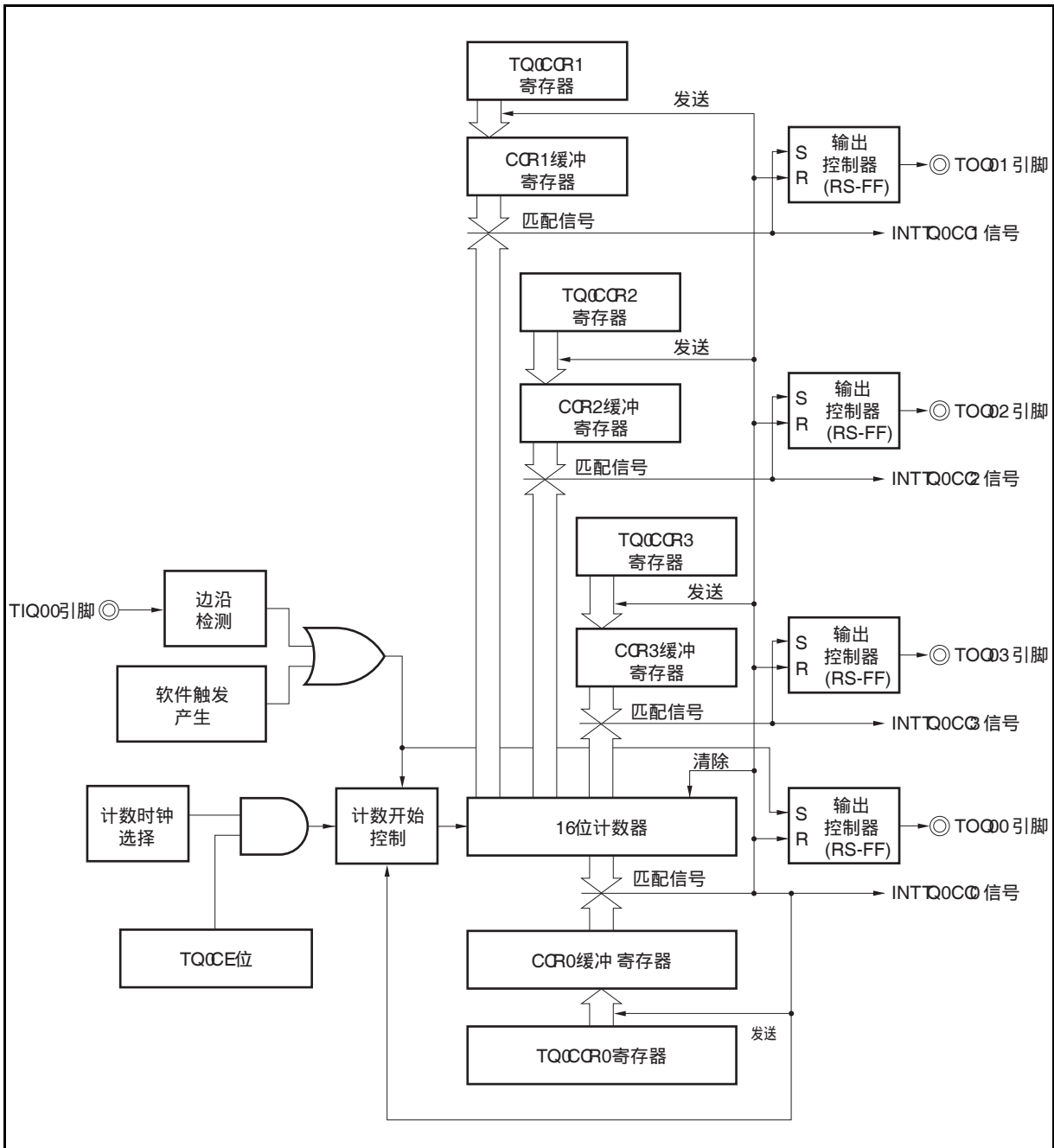
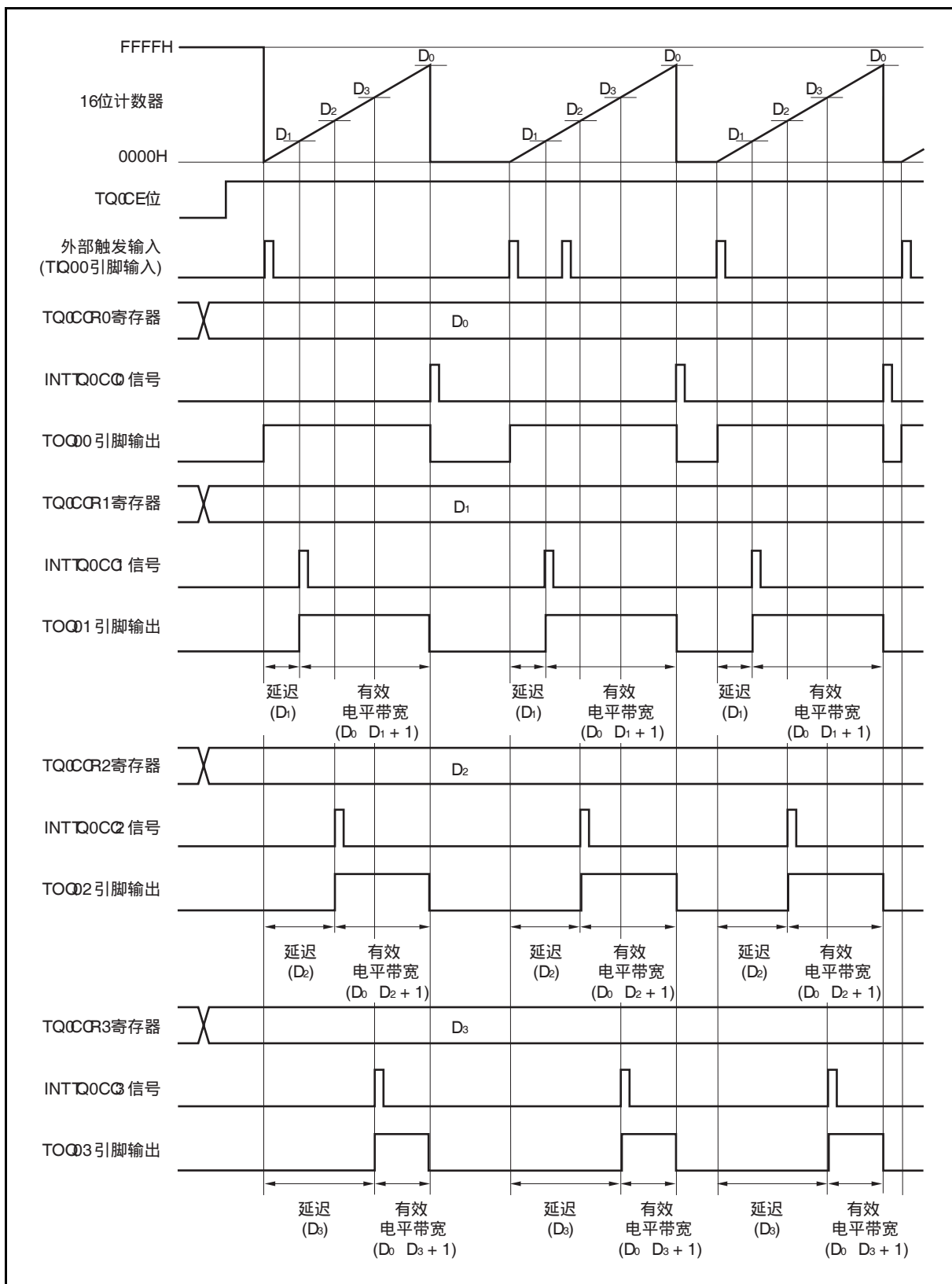


图 8-21. 单脉冲输出模式的基本时序



TQ0CE 位被置 1 后，16 位定时器/事件计数器 Q 进入等待触发信号的状态。触发被检测后，16 位计数器从 FFFFH 清零为 0000H，同时开始计数并从 TOQ0k 引脚输出单脉冲波形。单脉冲波形输出后，16 位计数器将被重置为 FFFFH 并停止计数，16 位定时器/时间计数器 P 重新进入等待触发信号的状态。如果在单脉冲波形输出过程中再次检测到触发信号，那么这个触发信号将被忽略。

单脉冲波形的输出延迟时间以及有效电平宽度可由以下算式得出。

输出延迟时间 = (TQ0CCRk 寄存器的设置值) × 计数时钟周期

有效电平宽度 = (TQ0CCR0 寄存器的值 - TQ0CCRk 寄存器的值 + 1) × 计数时钟周期

比较匹配中断请求信号 INTTQ0CC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器中的值相匹配时的下一个计数时钟时产生。比较匹配中断请求信号 INTTQ0CCk 会在 16 位计数器的计数值与 CCRk 缓冲寄存器中的值相匹配时产生。

触发源可以从外部触发输入信号的有效沿和设置软件触发(TQ0CTL1.TQ0EST 位 = 1)置 1 的方式进行选择。

备注 k = 1 至 3

图 8-22. 单脉冲输出模式的寄存器设置(1/3)

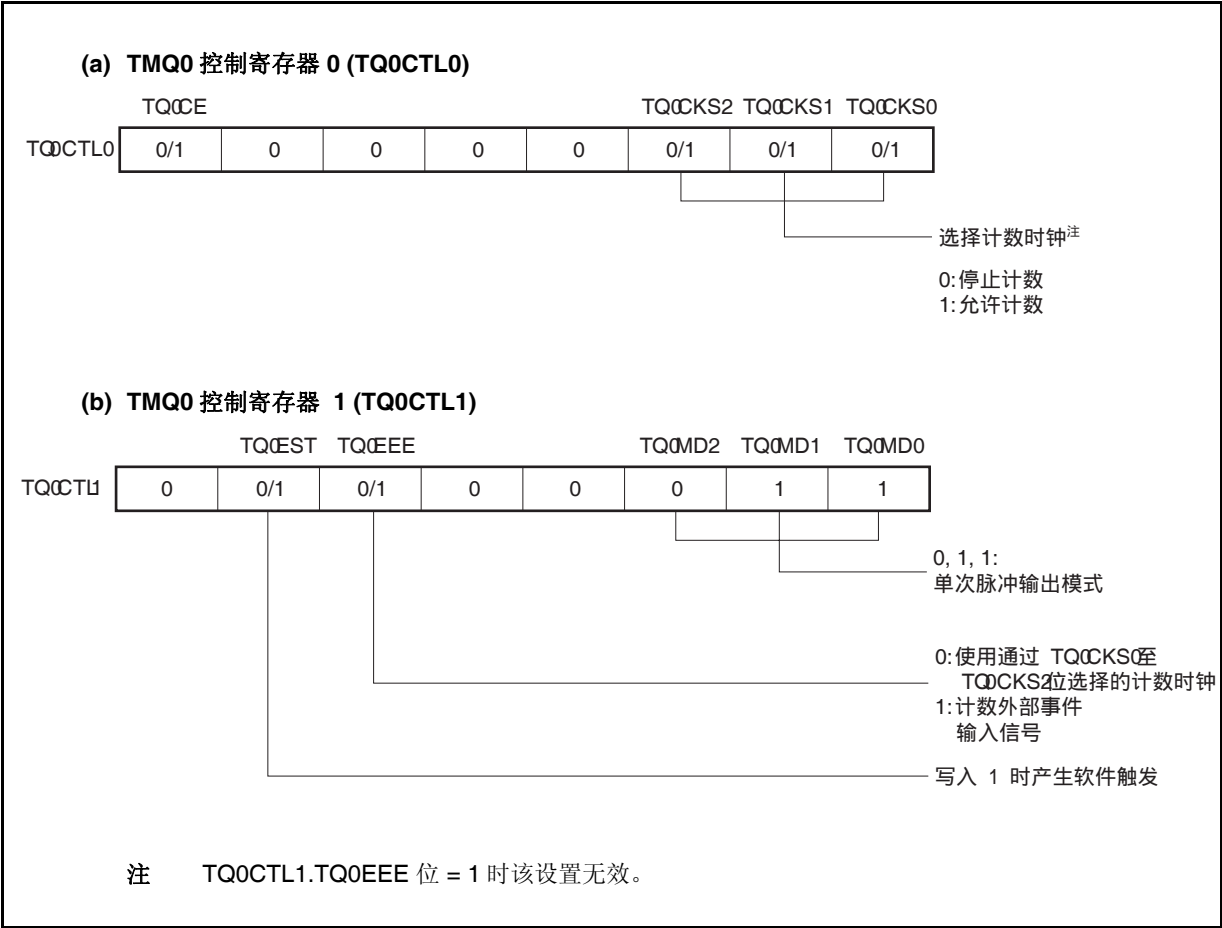


图 8-22. 单脉冲输出模式的寄存器设置(2/3)

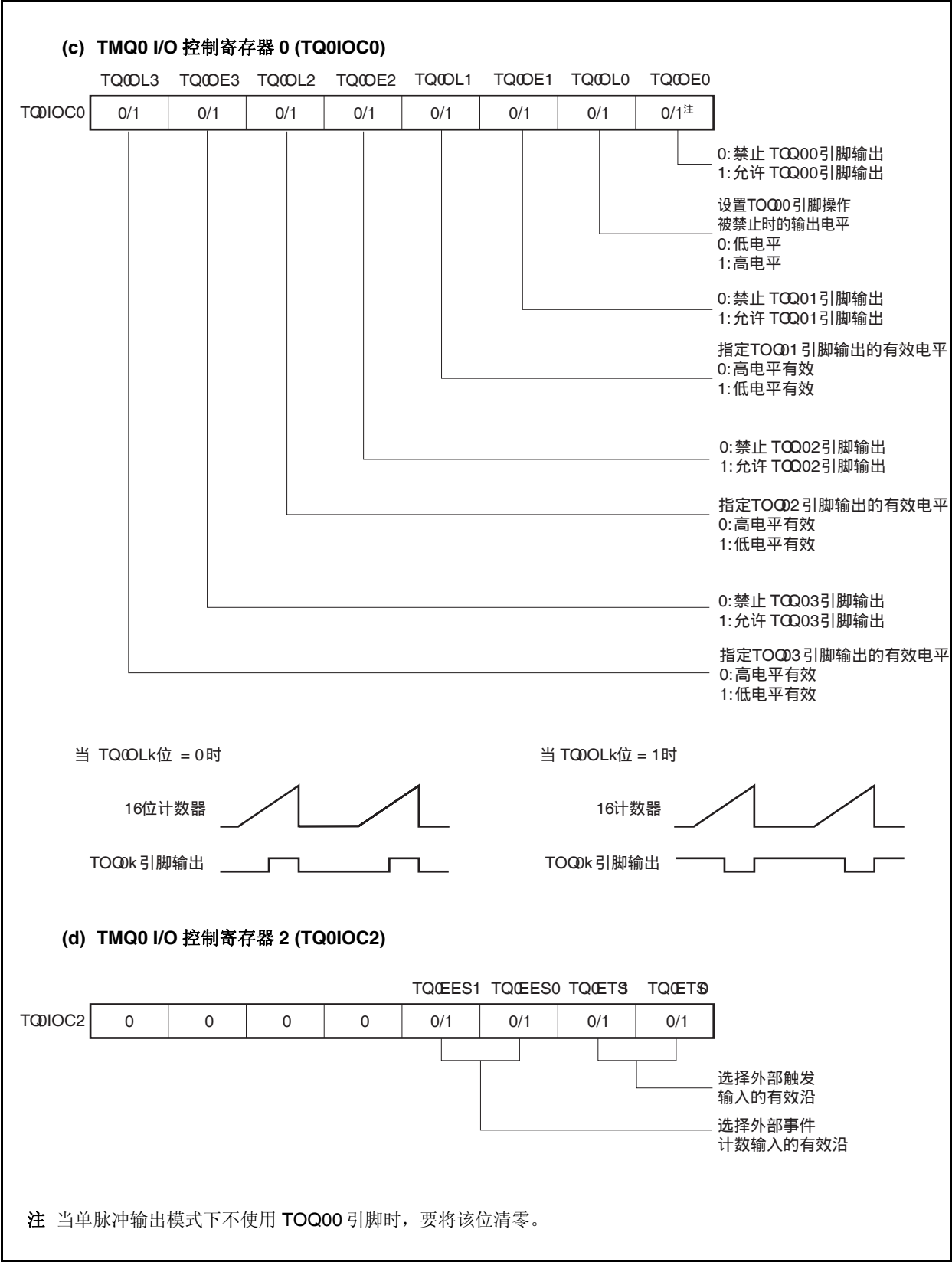


图 8-22. 单脉冲输出模式的寄存器设置(3/3)

(e) TMQ0 计数器读取缓冲寄存器(TQ0CNT)

可以通过读取 TQ0CNT 寄存器来读取 16 位计数器的计数值。

(f) TMQ0 捕捉/比较寄存器 0 至 3 (TQ0CCR0 至 TQ0CCR3)

如果将 TQ0CCR0 寄存器的值设置为 D_0 ，将 TQ0CCR k 寄存器的值设置为 D_k ，那么单脉冲波形的有效电平宽度和输出延迟时间如下所示。

有效电平宽度 = $(D_0 - D_k + 1) \times$ 计数时钟周期

输出延迟时间 = $D_k \times$ 计数时钟周期

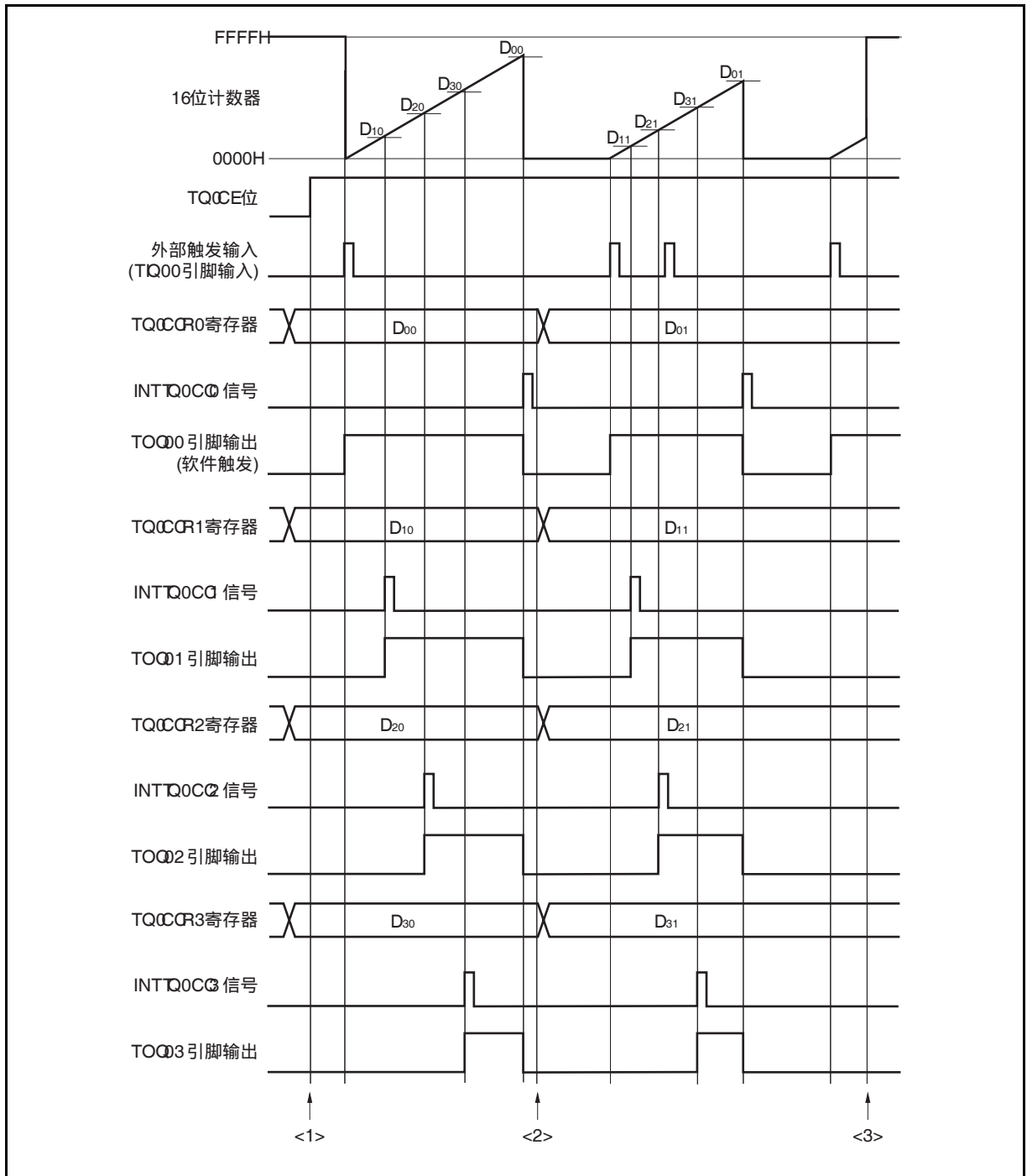
注意事项：如果 TQ0CCR k 触发器设置值大于 TQ0CCR0 寄存器的设置值，在单脉冲输出模式下无法输出单脉冲。

- 备注**
1. 单脉冲输出模式下，并未使用 TMQ0 I/O 控制寄存器 1 (TQ0IOC1) 和 TMQ0 选项寄存器 0 (TQ0OPT0)。
 2. $k = 1$ 至 3

(1) 单脉冲输出模式的操作流程

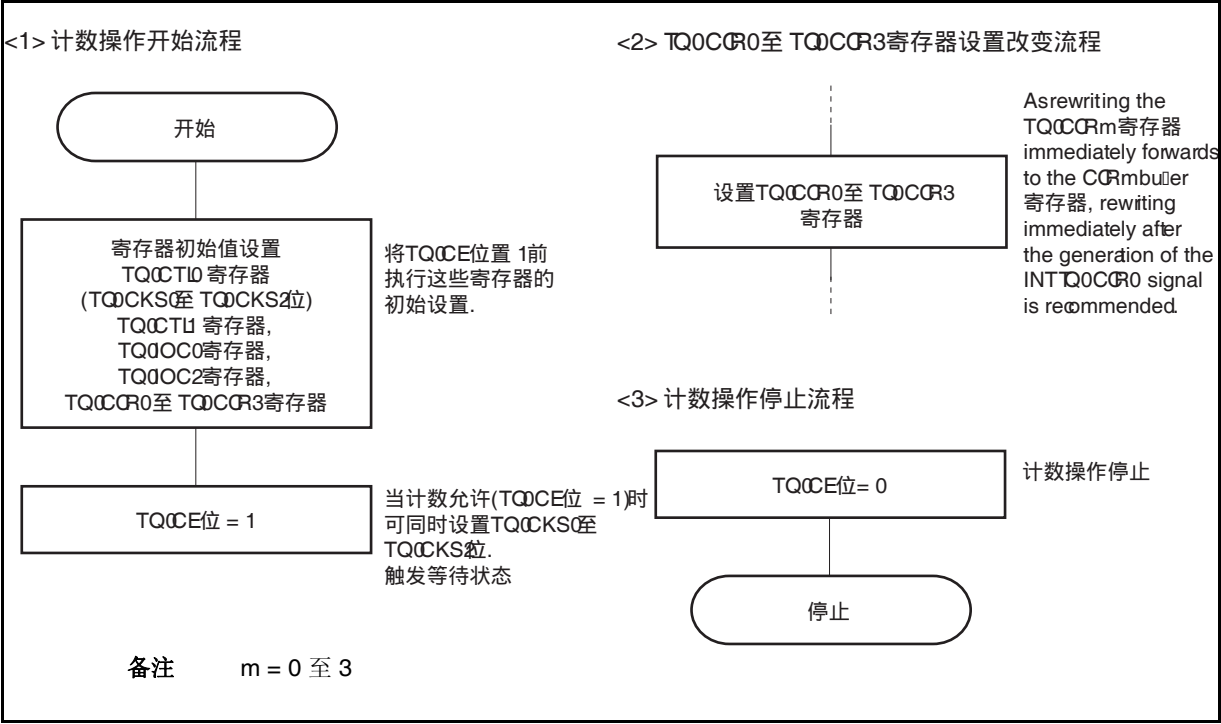
★

图 8-23. 单脉冲输出模式的软件操作流程(1/2)



★

图 8-23. 单脉冲输出模式的软件操作流程(2/2)

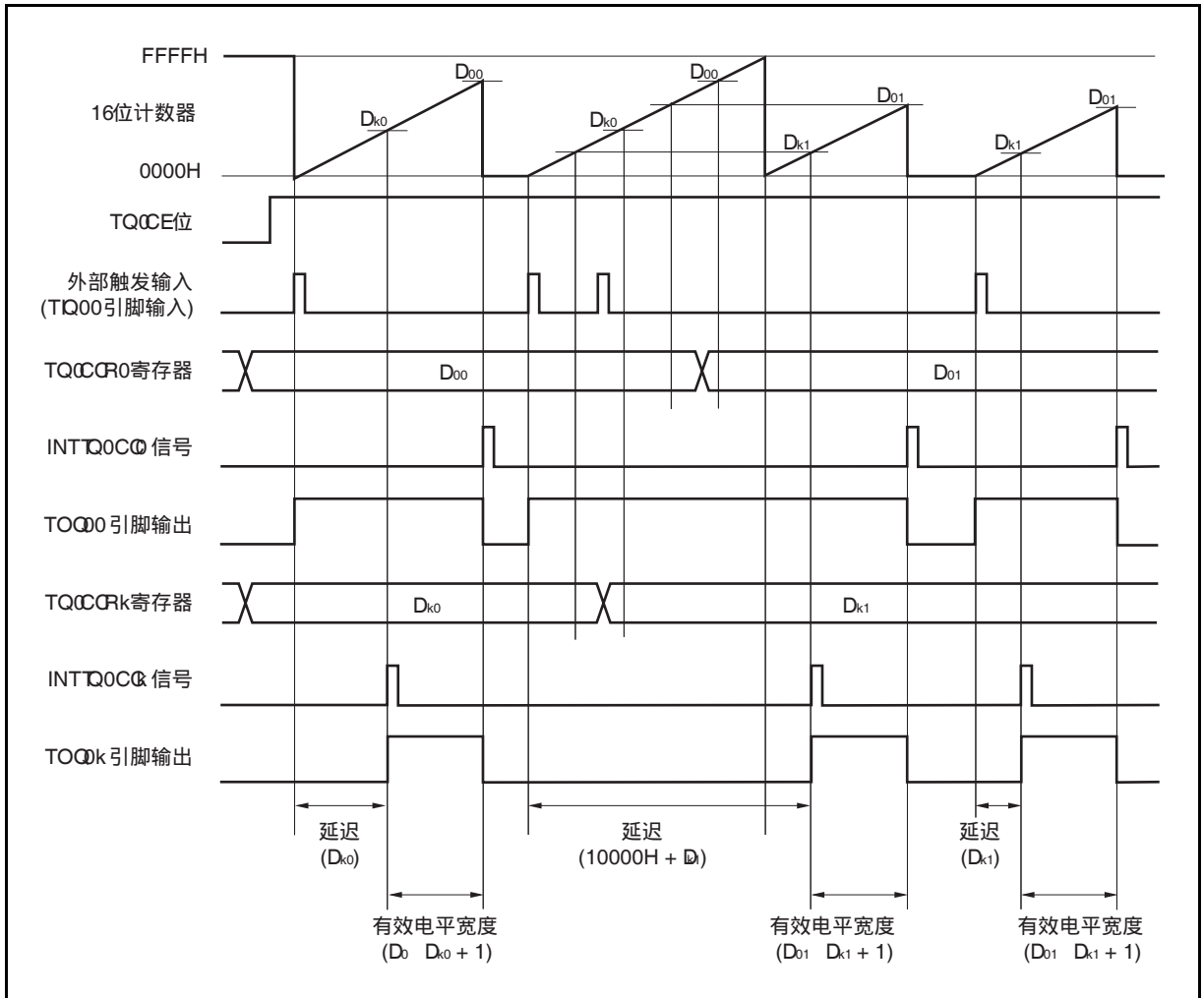


(2) 单脉冲输出模式的操作时序

(a) 改变 TQ0CCRm 寄存器时需要注意的事项

将 TQ0CCRm 寄存器的值减小时，要先停止计数，再对设定值进行改变。

如果在计数过程中将 TQ0CCR0 寄存器的值减小，那么 16 位计数器可能产生溢出。

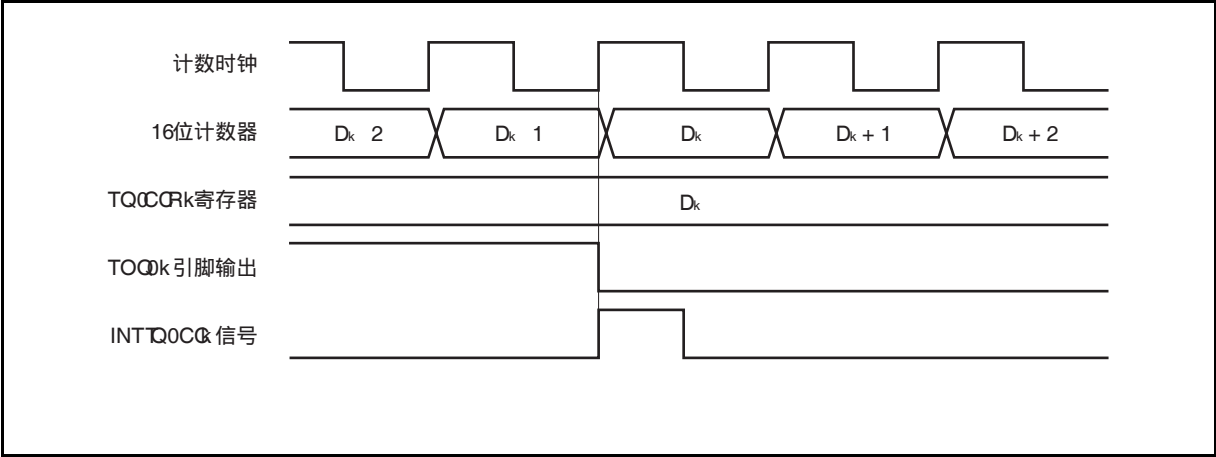


当将 TQ0CCR0 寄存器的值从 D₀₀ 改写为 D₀₁，TQ0CCRk 寄存器的值从 D_{k0} 改写为 D_{k1}，这里假设 D₀₀ > D₀₁ 且 D_{k0} > D_{k1}，如果 TQ0CCRk 寄存器被改写时，16 位计数器的计数值大于 D_{k1} 且小于 D_{k0}，并且 TQ0CCR0 寄存器被改写时，计数器的计数值大于 D₀₁ 且小于 D₀₀，两个设定值都将在改写后立即生效并与计数值进行比较。此时计数器会计数到 FFFFH 然后从 0000H 重新开始计数。当计数值与 D_{k1} 匹配时，将产生 INTTQ0CCk 信号，同时 TO00k 引脚电平变为有效电平。当计数值与 D₀₁ 匹配时，将产生 INTTQ0CC0 信号，同时 TO00k 引脚电平变为非有效电平并且计数器停止计数。因此可见，此时定时器输出的单脉冲波形的延迟时间或有效电平宽度会与预期数值有所差异。

备注 k = 1 至 3

(b) 比较匹配中断请求信号的产生时序 (INTTQ0CCK)

单脉冲输出模式下， INTTQ0CCK 信号的产生时序与其它模式下 INTTQ0CCK 信号的产生时序有所不同。这里， INTTQ0CCK 信号将在 16 位计数器的计数值与 TQ0CCRk 寄存器的值相匹配时产生。



通常情况下， INTTQ0CCK 信号会在 16 位计数器计数值与 TQ0CCRk 寄存器的值匹配之后的下一个计数时钟同步产生。

但在单脉冲输出模式下，INTTQ0CCK 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOQ0k 引脚输出信号变化的时序而改变的。

备注 $k = 1$ 至 3

8.5.5 PWM 输出模式(TQ0MD2 至 TQ0MD0 位 = 100)

PWM 输出模式下, TQ0CTL0.TQ0CE 位被置 1 后, TOQ01 至 TOQ03 引脚将输出 PWM 波形。

另外, TOQ00 引脚将输出一个以 PWM 波形周期的 2 倍为周期的脉冲信号。

图 8-24. PWM 输出模式的配置图

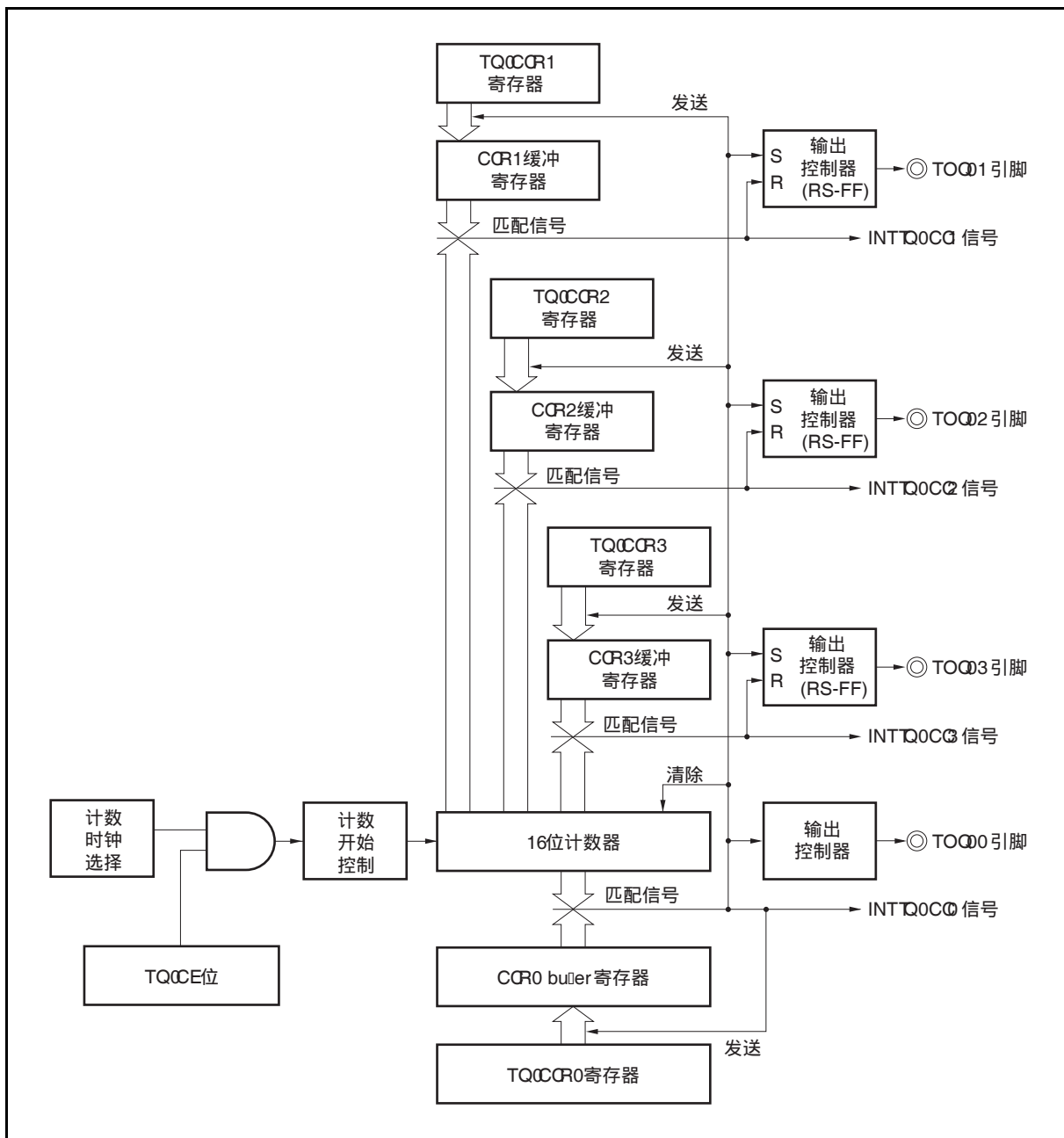
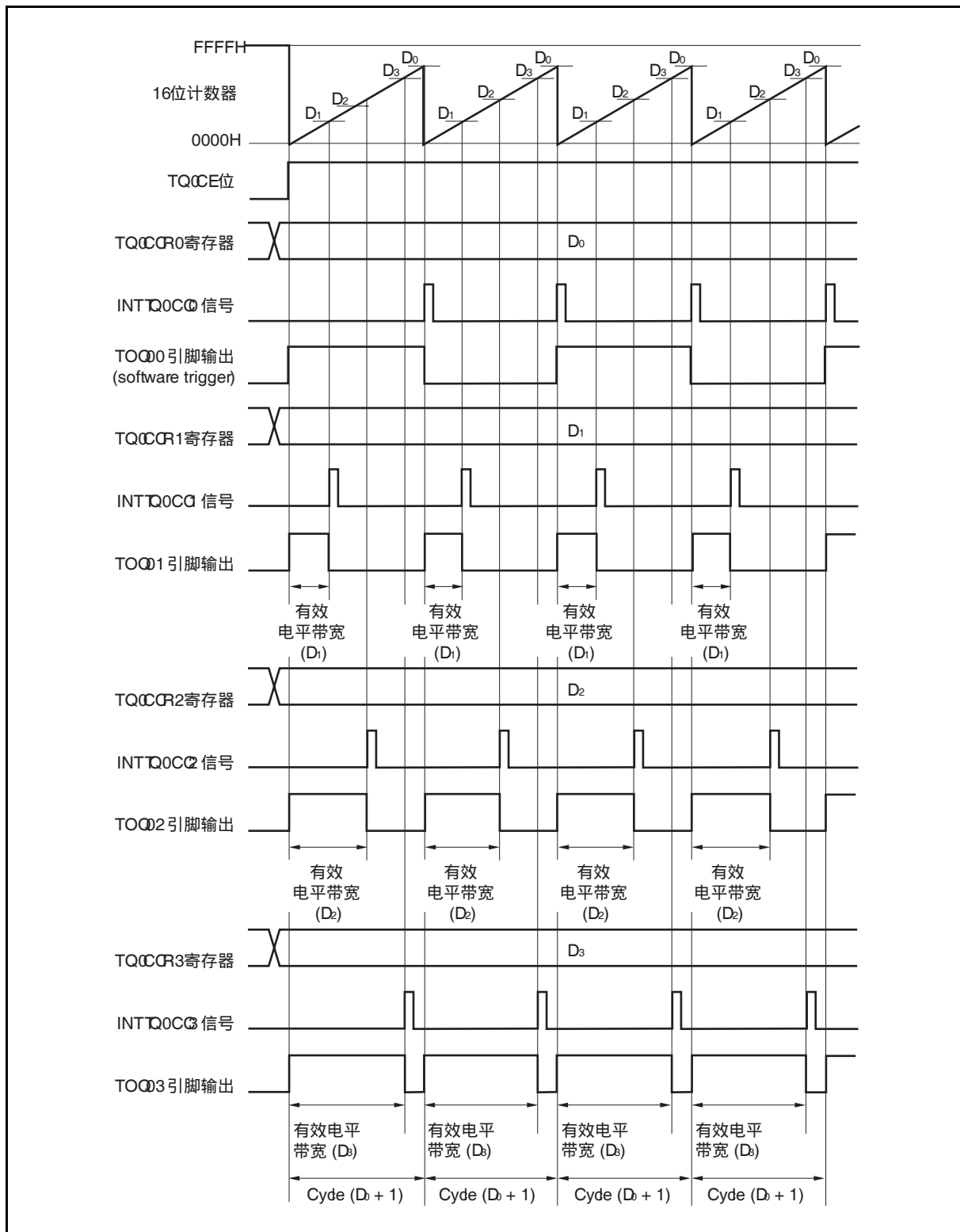


图 8-25. PWM 输出模式的基本时序



TQ0CE 位被置 1 后，16 位计数器从 FFFFH 清零为 0000H 并开始计数，同时从 TOQ0k 引脚输出 PWM 波形。PWM 波形的有效电平宽度，周期以及占空比系数可由以下算式得出。

有效电平宽度 = (TQ0CCRk 寄存器的值) × 计数时钟周期
周期 = (TQ0CCR0 寄存器的值 + 1) × 计数时钟周期
占空比系数 = (TQ0CCRk 寄存器的值)/(TQ0CCR0 寄存器的值 + 1)

可以在计数器计数过程中，通过改写 TQ0CCRm 寄存器的值来改变 PWM 波形的参数。改写后的数值将在 16 位计数器的计数值与当前 CCR0 缓冲寄存器的值匹配之后生效，同时 16 位计数器也会被清为 0000H。
比较匹配中断请求信号 INTTQ0CC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器的值相匹配后的下一个计数时钟时产生，同时 16 位计数器将被清零。比较匹配中断请求信号 INTTQ0CCk 会在 16 位计数器的计数值与 CCRk 缓冲寄存器的值相匹配时产生。

备注 k = 1 至 3
 m = 0 至 3

图 8-26. PWM 输出模式的寄存器设置(1/3)

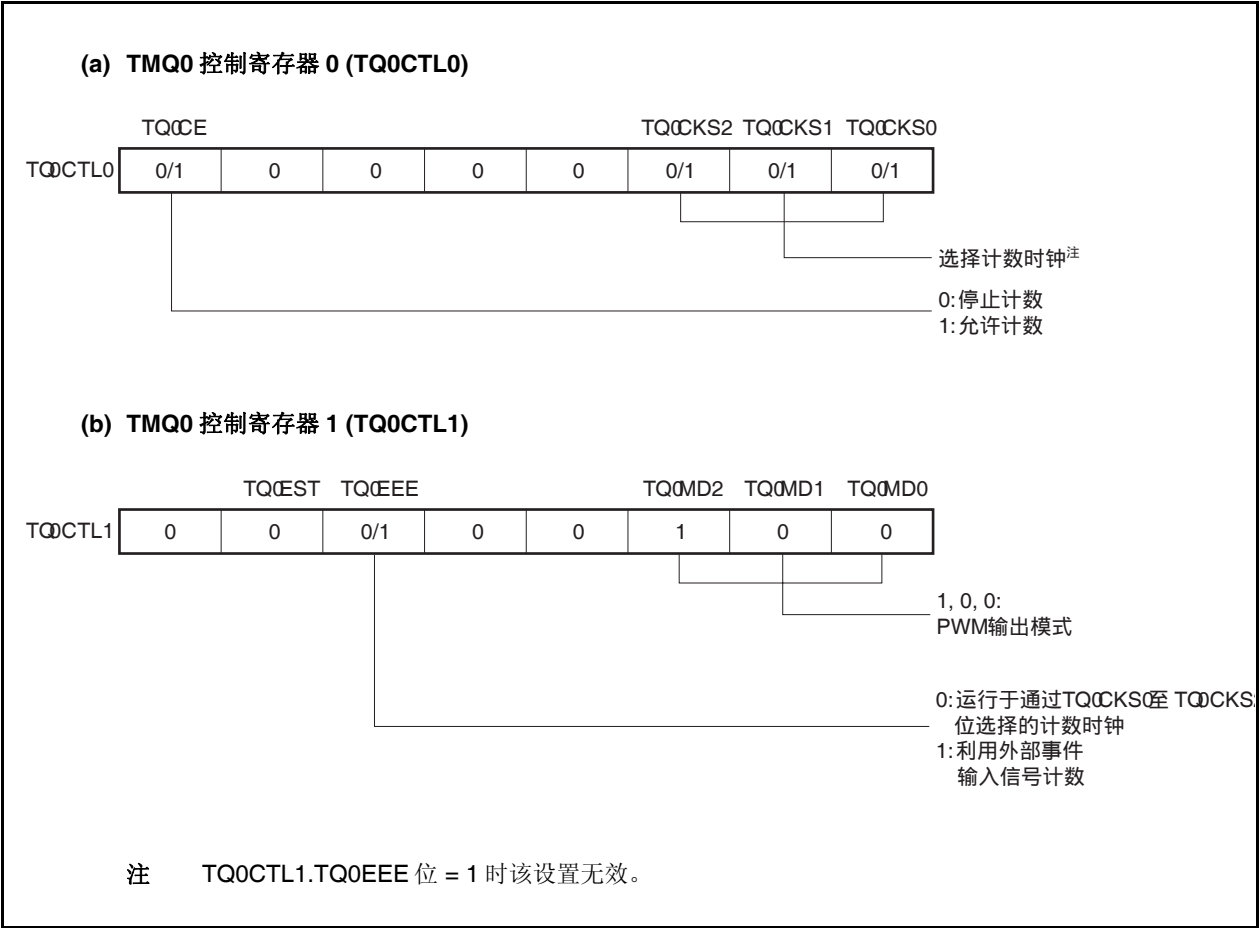


图 8-26. PWM 输出模式的寄存器设置(2/3)

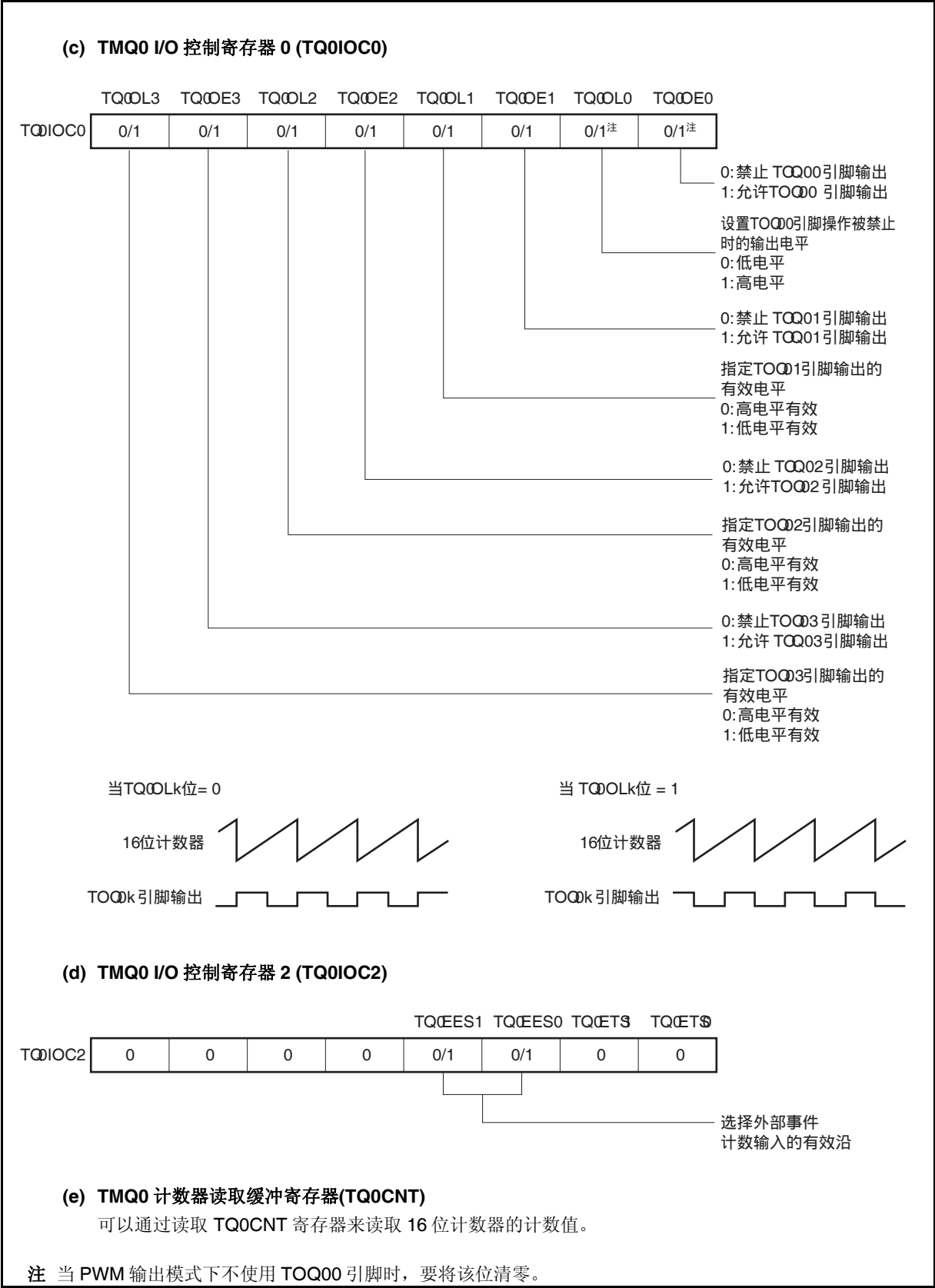


图 8-26. PWM 输出模式的寄存器设置(3/3)

(f) TMQ0 捕捉/比较寄存器 0 至 3 (TQ0CCR0 至 TQ0CCR3)

如果将 TQ0CCR0 寄存器的值设置为 D_0 ，将 TQ0CCR k 寄存器的值设置为 D_k ，那么 PWM 波形的周期和有效电平如下所示。

周期 = $(D_0 + 1) \times$ 计数时钟周期

有效电平宽度 = $D_k \times$ 计数时钟周期

备注

1. PWM 输出模式下，并未使用 TMQ0 I/O 控制寄存器 1 (TQ0IOC1) 和 TMQ0 选项寄存器 0 (TQ0OPT0)。
2. 通过写 TMQ0 捕捉/比较寄存器 1 (TQ0CCR1) 来使更新 TMQ0 捕捉/比较寄存器 2 (TQ0CCR2) 和 TMQ0 捕捉/比较寄存器 3 (TQ0CCR3) 有效。

(1) PWM 输出模式的操作流程

图 8-27. PWM 输出模式的软件处理流程(1/2)

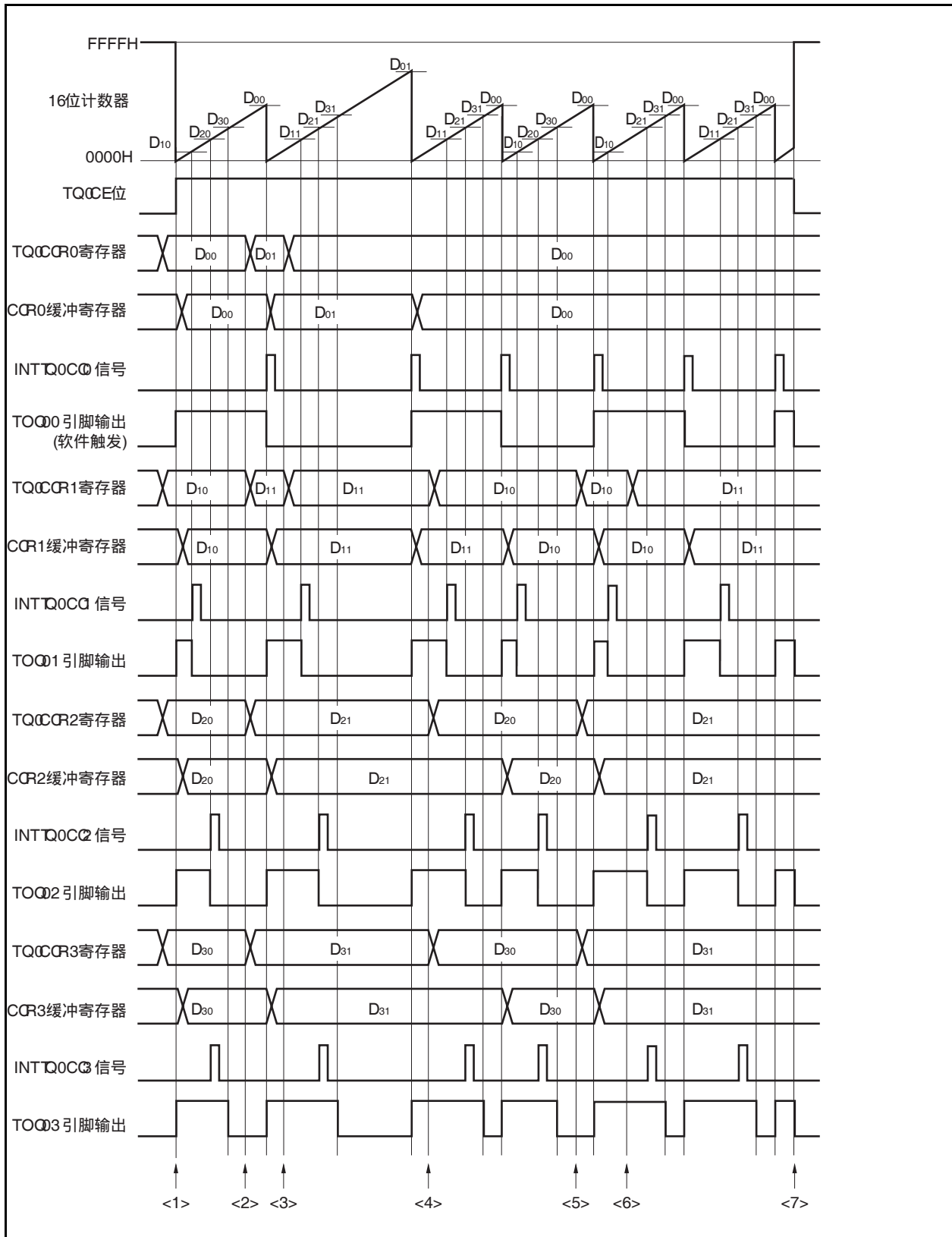
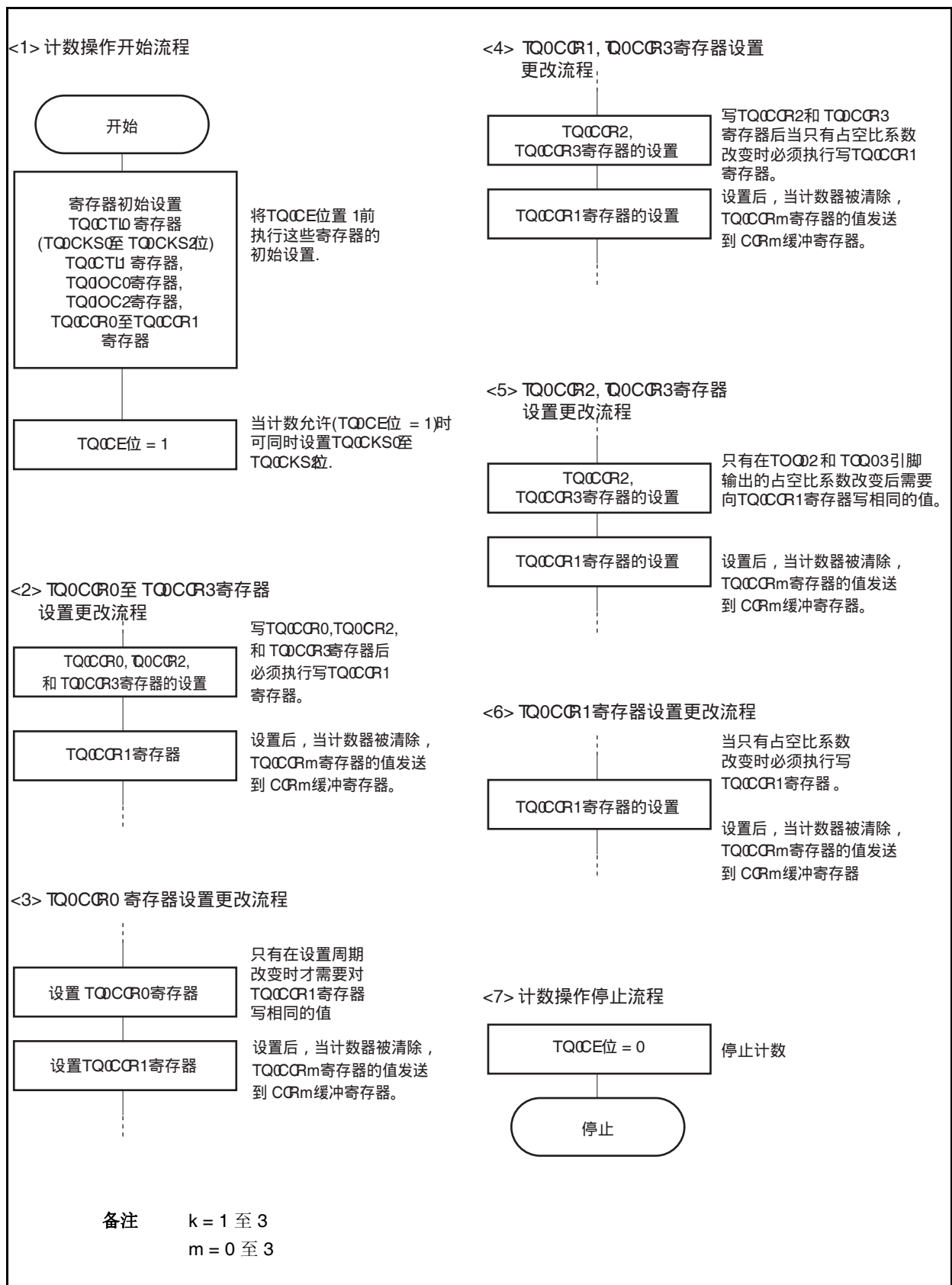


图 8-27. PWM 输出模式的软件处理流程(2/2)

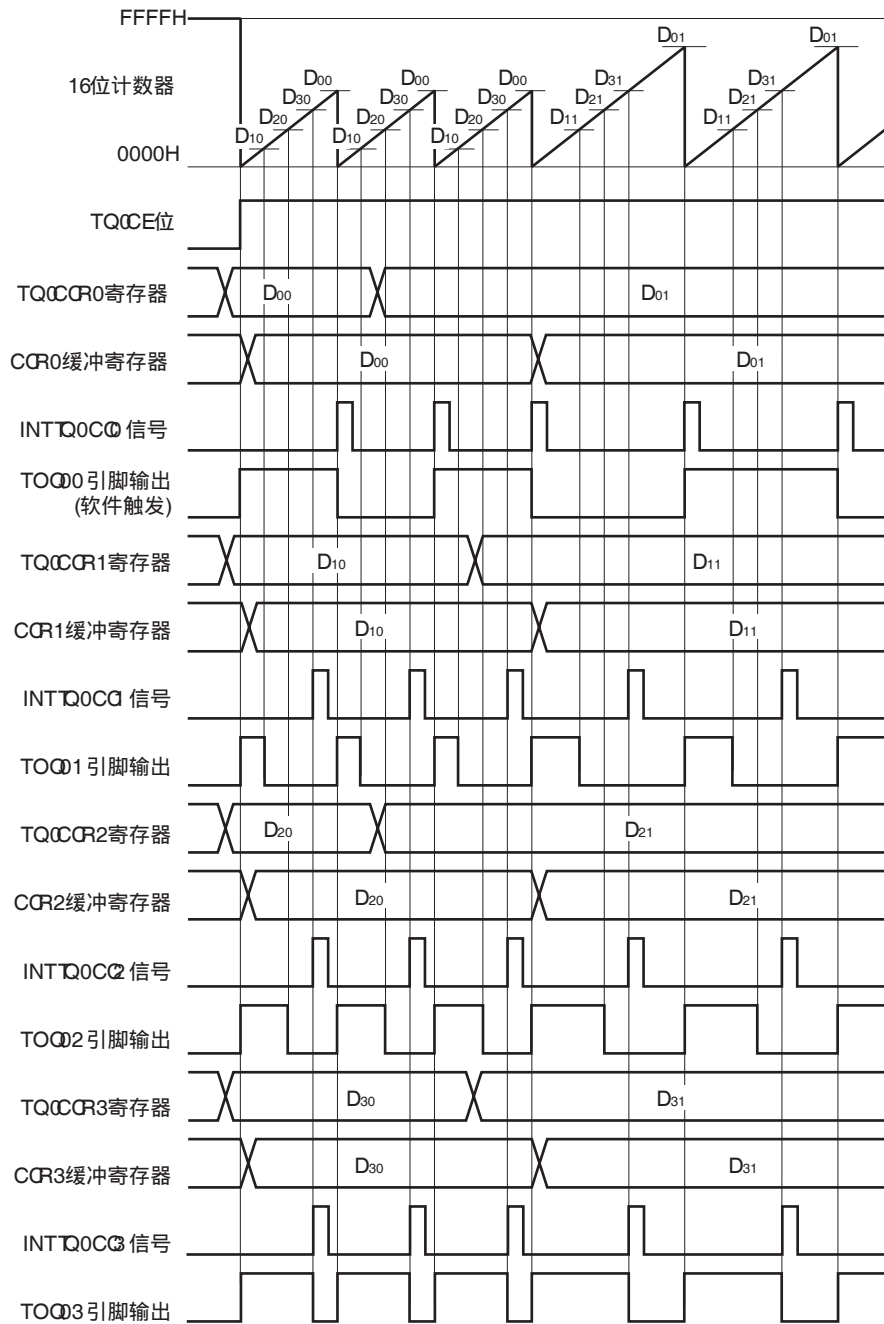


(2) PWM 输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

计数器计数过程中，若改变 PWM 波形，则要在最后进行对 TQ0CCR1 寄存器的写入操作。

对 TQ0CCR1 寄存器进行写入操作之后，若需再次更改 TQ0CCRk 寄存器的值，须等待下一个 INTTQ0CC0 信号被检测到后，再进行相关操作。



若要将 TQ0CCRm 寄存器中设置的数据传送至 CCRm 缓冲寄存器，必须对 TQ0CCR1 寄存器进行写入操作。

同时改变 PWM 波形的周期和有效电平时，先将周期值写入 TQ0CCR0 寄存器，然后向 TQ0CCR2 和 TQ0CCR3 寄存器写入有效电平宽度，最后向 TQ0CCR1 寄存器写入有效电平宽度。

只改变 PWM 波形的有效电平宽度（占空比系数）时，首先向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平然后再向 TQ0CCR1 寄存器设置有效电平。

若只改变通过 TOQ01 引脚输出的 PWM 波形的有效电平宽度（占空比系数）时，只需要设置 TQ0CCR1 寄存器。

若只改变通过 TOQ02 和 TOQ03 引脚输出的 PWM 波形的有效电平宽度（占空比系数）时，先要向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平宽度，然后向 TQ0CCR1 寄存器写入相同的值。

数据被写入 TQ0CCR1 寄存器之后，16 位计数器被清零时，TQ0CCRm 寄存器中的数值才会被同步传送至 CCRm 缓冲寄存器中，此时 CCRm 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

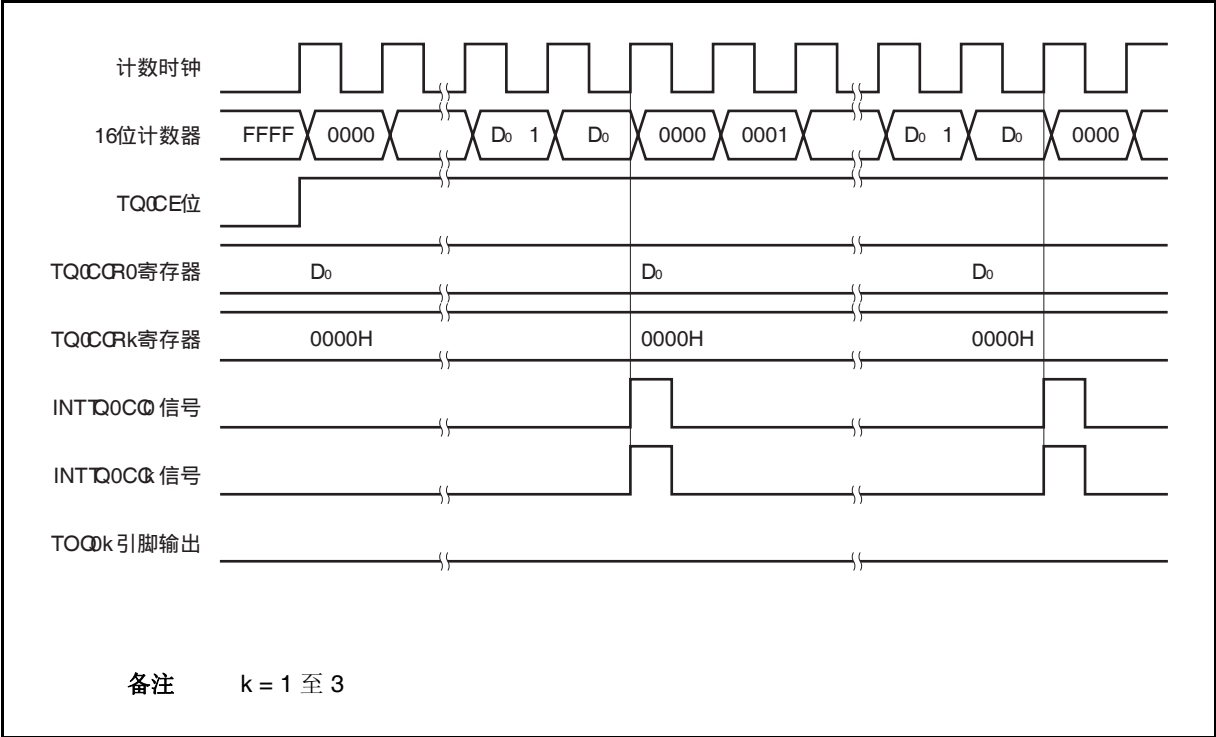
若只改变 PWM 波形的周期，首先要将周期设置于 TQ0CCR0 寄存器，然后向 TQ0CCR1 寄存器写入相同的值。

若要在写入 TQ0CCR1 寄存器后再次写入 TQ0CCR0 至 TQ0CCR3 寄存器，需要在产生 INTTQ0CC0 信号后再进行此步操作。否则，CCRm 缓冲寄存器的值可能无法确定，因为数据从 TQ0CCRm 寄存器传送至 CCRm 缓冲寄存器的时刻与 TQ0CCRm 寄存器的写入时刻相冲突。

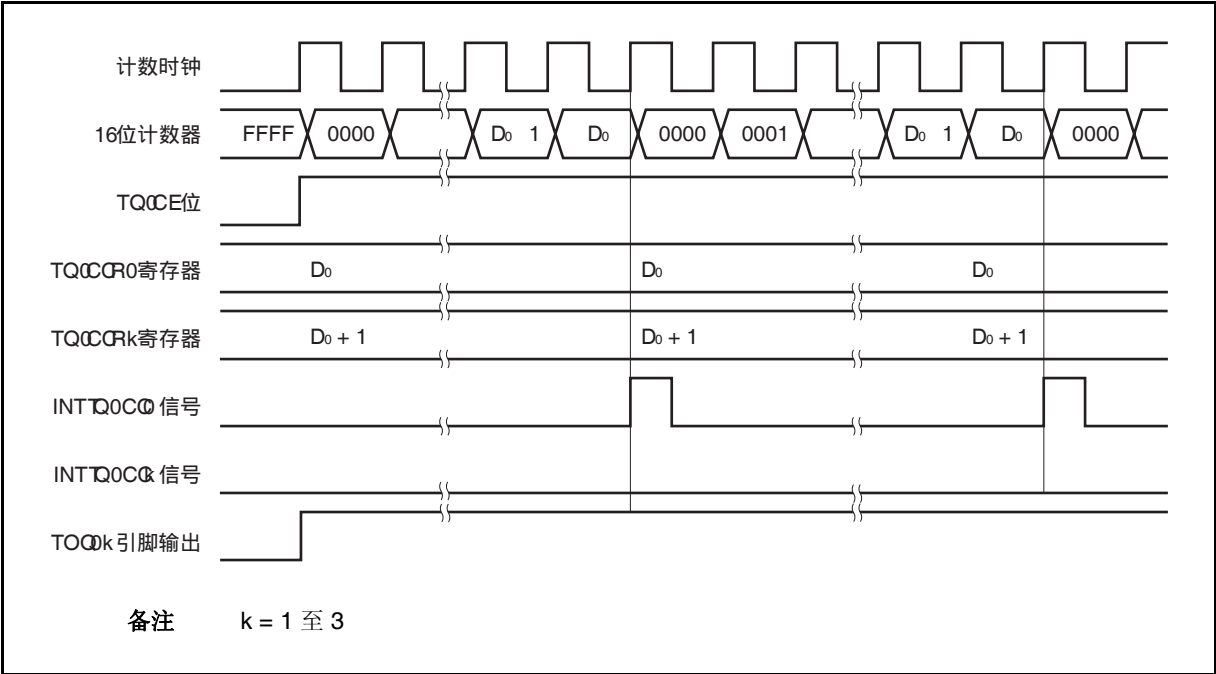
备注 m = 0 至 3

(b) PWM 波形的 0%/100% 输出

若要输出 0% 的波形，则需将 TQ0CCRk 寄存器设置为 0000H。如果 TQ0CCR0 寄存器被设置为 FFFFH，那么 INTTQ0CCK 信号将周期性产生。

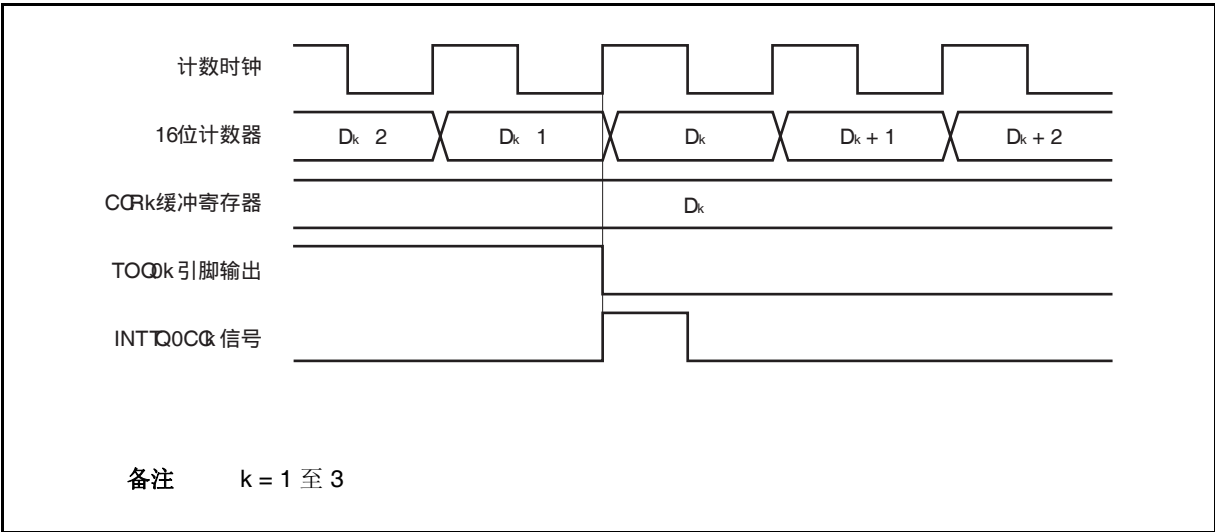


若要输出 100% 波形，则需将 TQ0CCRk 寄存器设置为(TQ0CCR0 寄存器值 + 1)。TQ0CCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。



(c) 比较匹配中断请求信号的产生时序(INTTQ0CCK)

PWM 输出模式下，INTTQ0CCK 信号的产生时序与其它模式下 INTTQ0CCK 信号的产生时序有所不同，这里，INTTQ0CCK 信号将在 16 位计数器的计数值与 TQ0CCRk 寄存器的值相匹配时产生。



通常情况下，INTTQ0CCK 信号会在 16 位计数器计数值与 TQ0CCRk 寄存器的值匹配之后的下一个计数时钟同步产生。

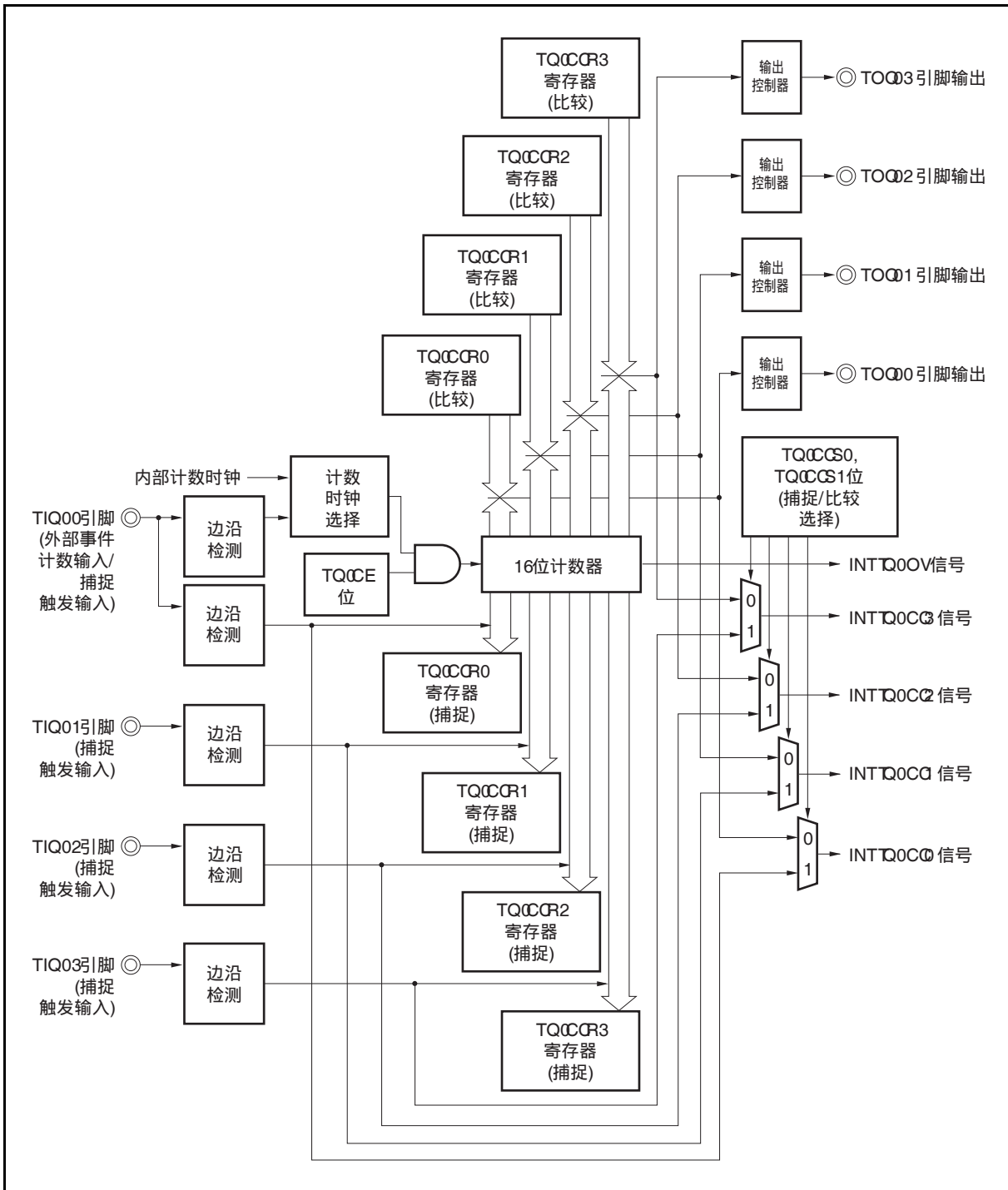
但在 PWM 输出模式下，INTTQ0CCK 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOQ0k 引脚输出信号变化的时序而改变的。

8.5.6 自由运行定时器模式 (TQ0MD2 至 TQ0MD0 位 = 101)

在自由运行定时器模式中, TQ0CTL0.TQ0CE 位被置 1 后, 16 位定时器/时间计数器 Q 开始计数。此时, 根据 TQ0OPT0.TQ0CCS0 和 TQ0OPT0.TQ0CCS1 位的设置 TQ0CCRm 寄存器的功能可以在比较寄存器和捕捉寄存器之间进行选择。

备注 m = 0 至 3

图 8-28. 自由运行定时器模式的配置图

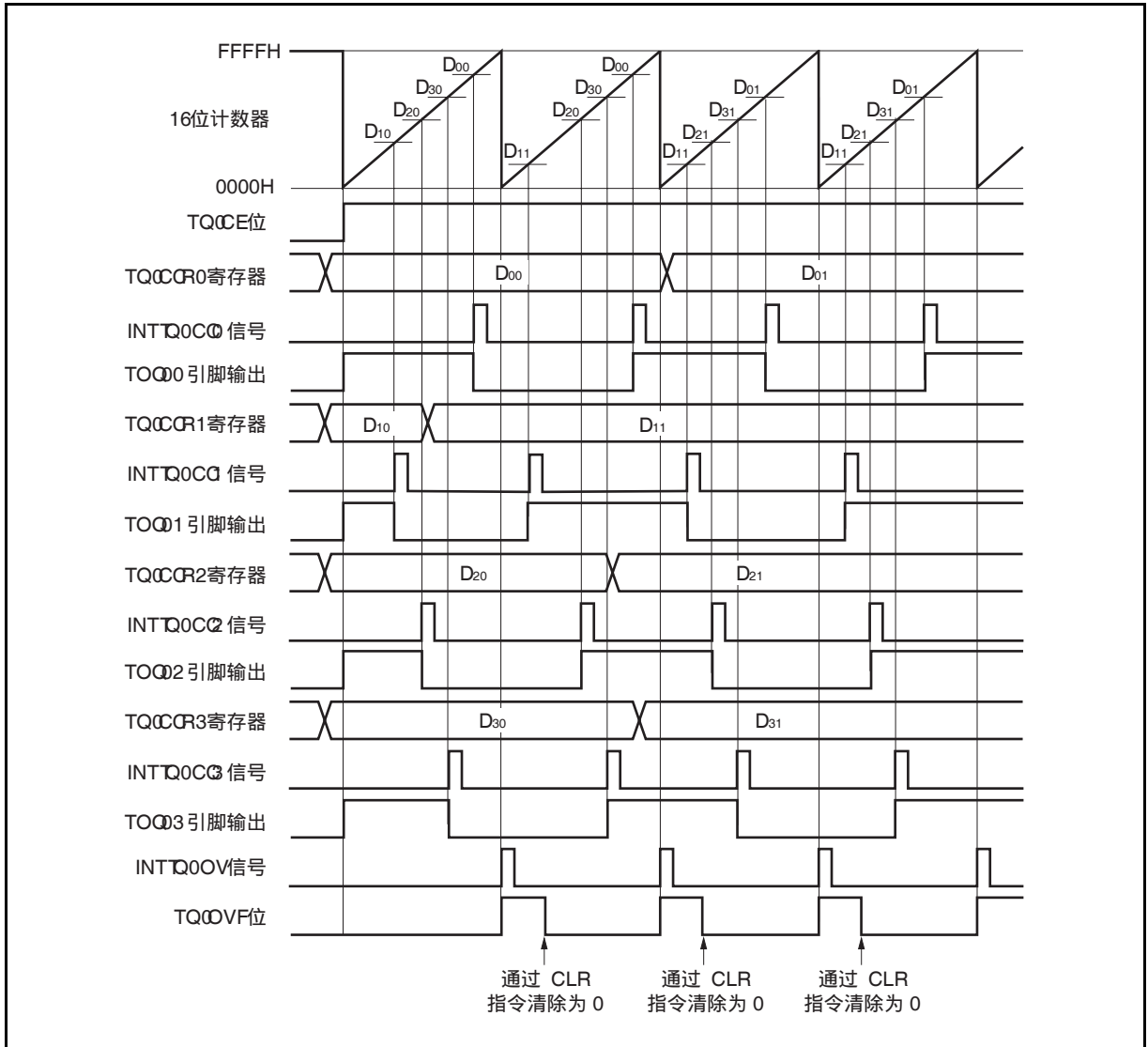


TQ0CE 位被置 1 后, 16 位定时/事件计数器 Q 开始计数, 同时 TOQ00 至 TOQ03 引脚输出信号的电平反转。当 16 位计数器的计数值与 TQ0CCRm 寄存器的值匹配时将产生一个比较匹配中断请求信号(INTTQ0CCm), 同时 TOQ0m 引脚的输出电平将反转。

此后, 16 位计数器将继续与计数时钟同步计数。当它计数到 FFFFH 时, 将在下一个计数时钟时产生一个溢出中断请求信号(INTTQ0OV), 同时计数器被清零并重新开始计数。此时, 溢出标志(TQ0OPT0.TQ0OVF 位) 也会被置 1。请通过软件使用 CLR 指令将溢出标志位清零。

可以在计数器计数过程中改写 TQ0CCRm 寄存器的值。如果对 TQ0CCRm 寄存器的值进行了改写, 那么新的值将立即生效并用于与计数值的比较。

图 8-29. 自由运行定时器模式的基本时序 (比较功能)



TQ0CE 位被置 1 后，16 位计数器开始计数。当检测到 TIQ0m 引脚输入信号的有效沿时，16 位计数器的计数值被保存在 TQ0CCRm 寄存器中，同时产生一个捕捉中断请求信号 (INTTQ0CCm)。

此后，16 位计数器将继续与计数时钟同步计数。当它计数到 FFFFH 时，将在下一个计数时钟时产生一个溢出中断请求信号 (INTTQ0OV)，同时计数器被清零并重新开始计数。此时，溢出标志 (TQ0OVF 位) 也会被置 1。请通过软件使用 CLR 指令将溢出标志位清零。

图 8-30. 自由运行定时器模式的基本时序 (捕捉功能)

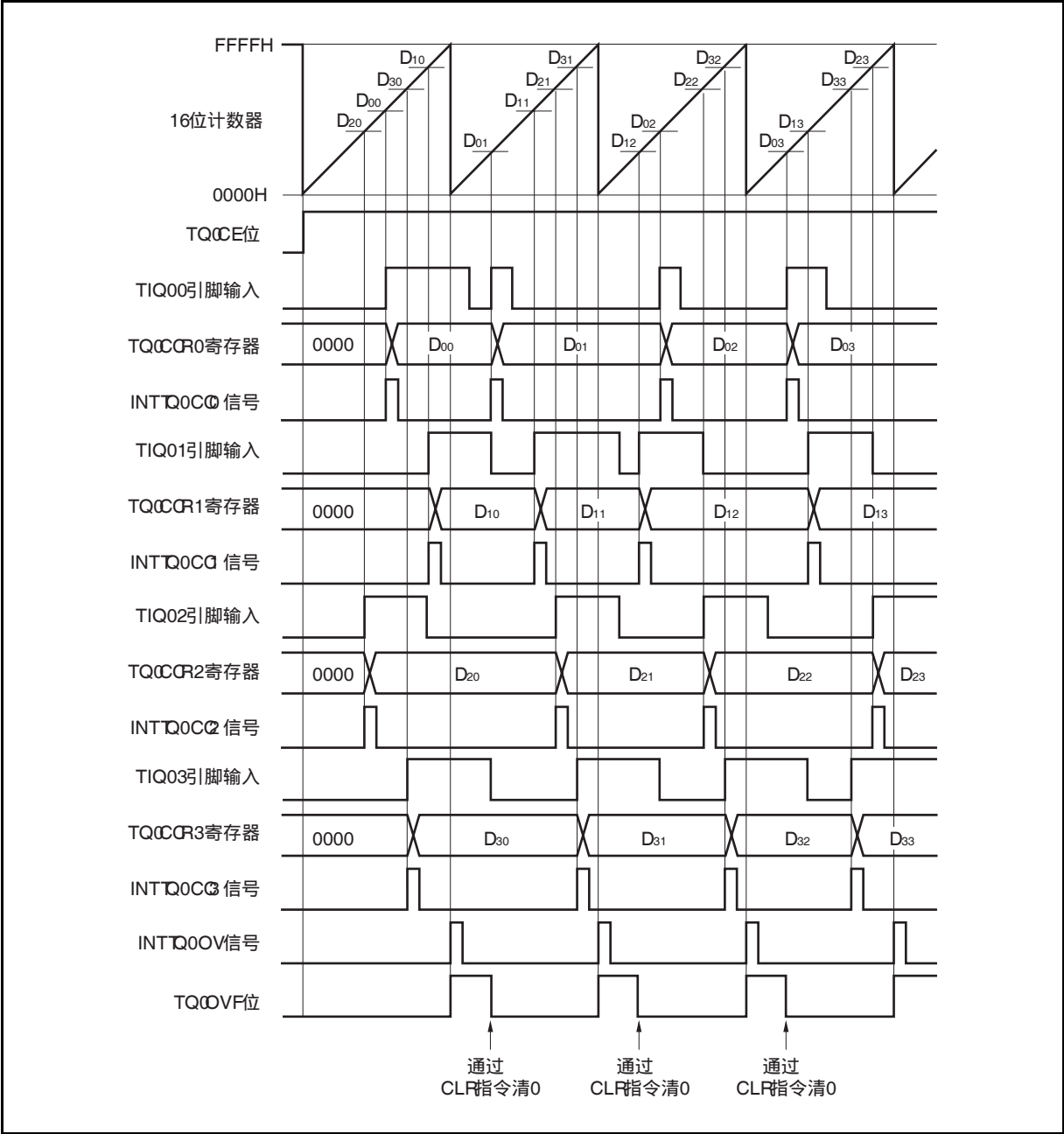


图 8-31. 自由运行定时器模式的寄存器设置(1/3)

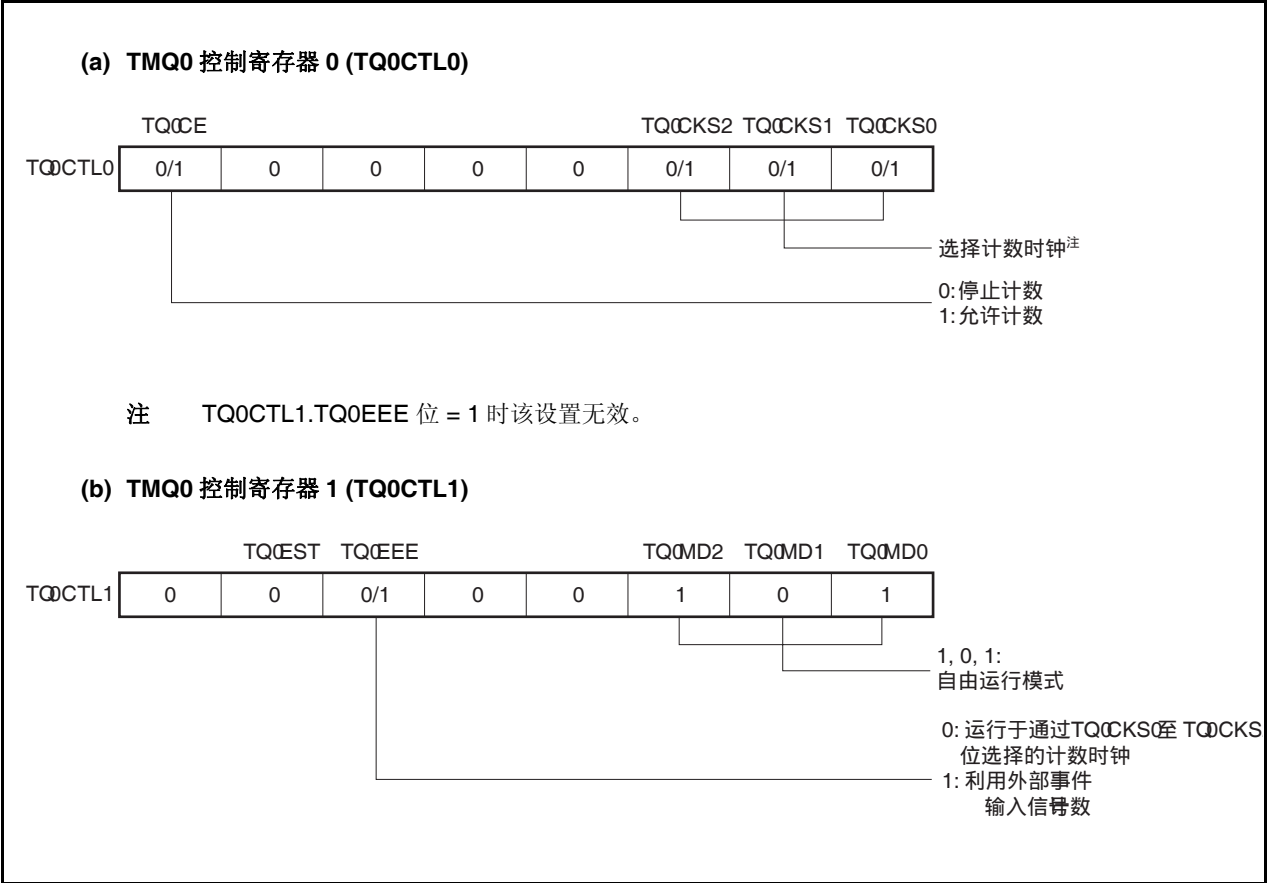


图 8-31. 自由运行定时器模式的寄存器设置 (2/3)

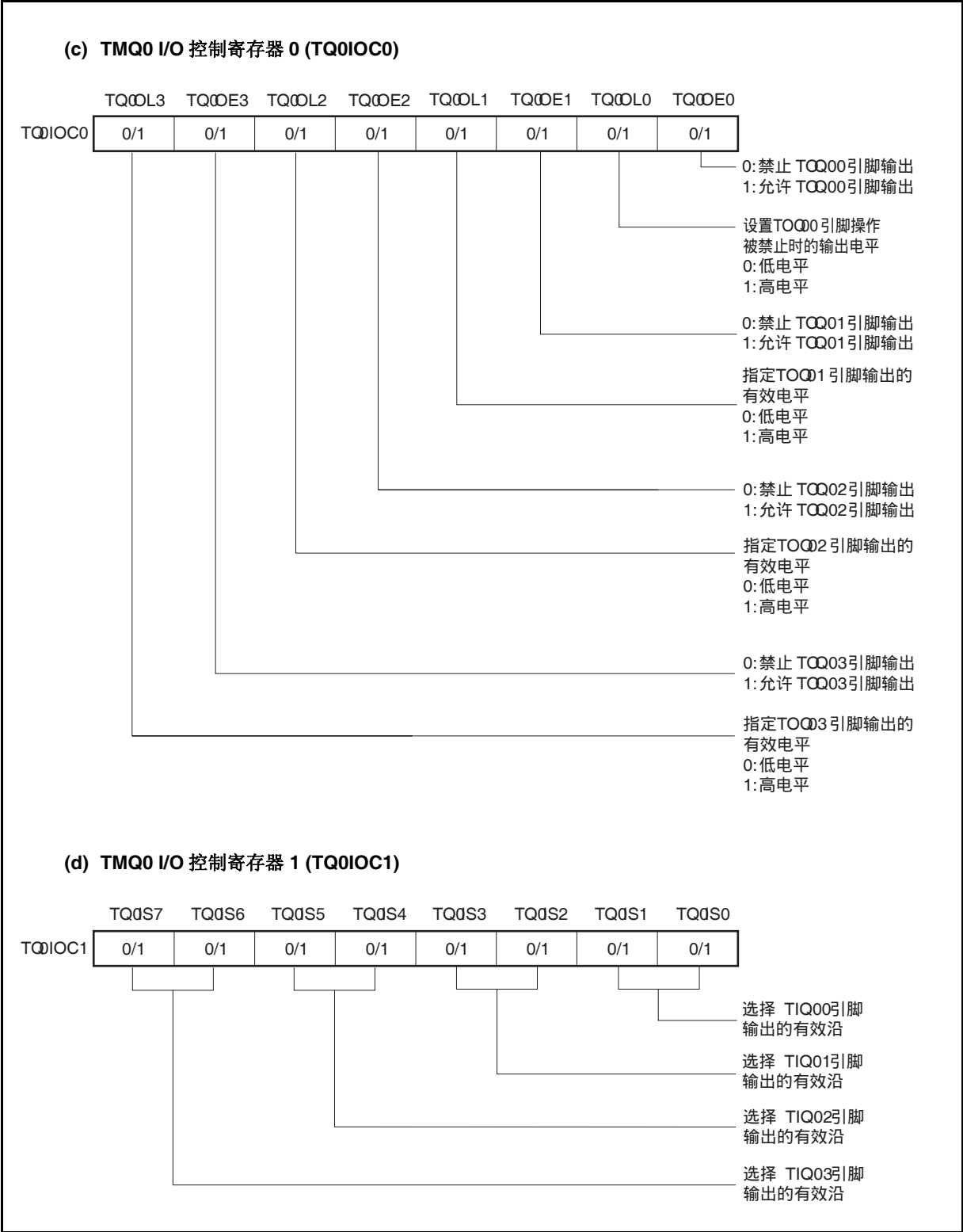
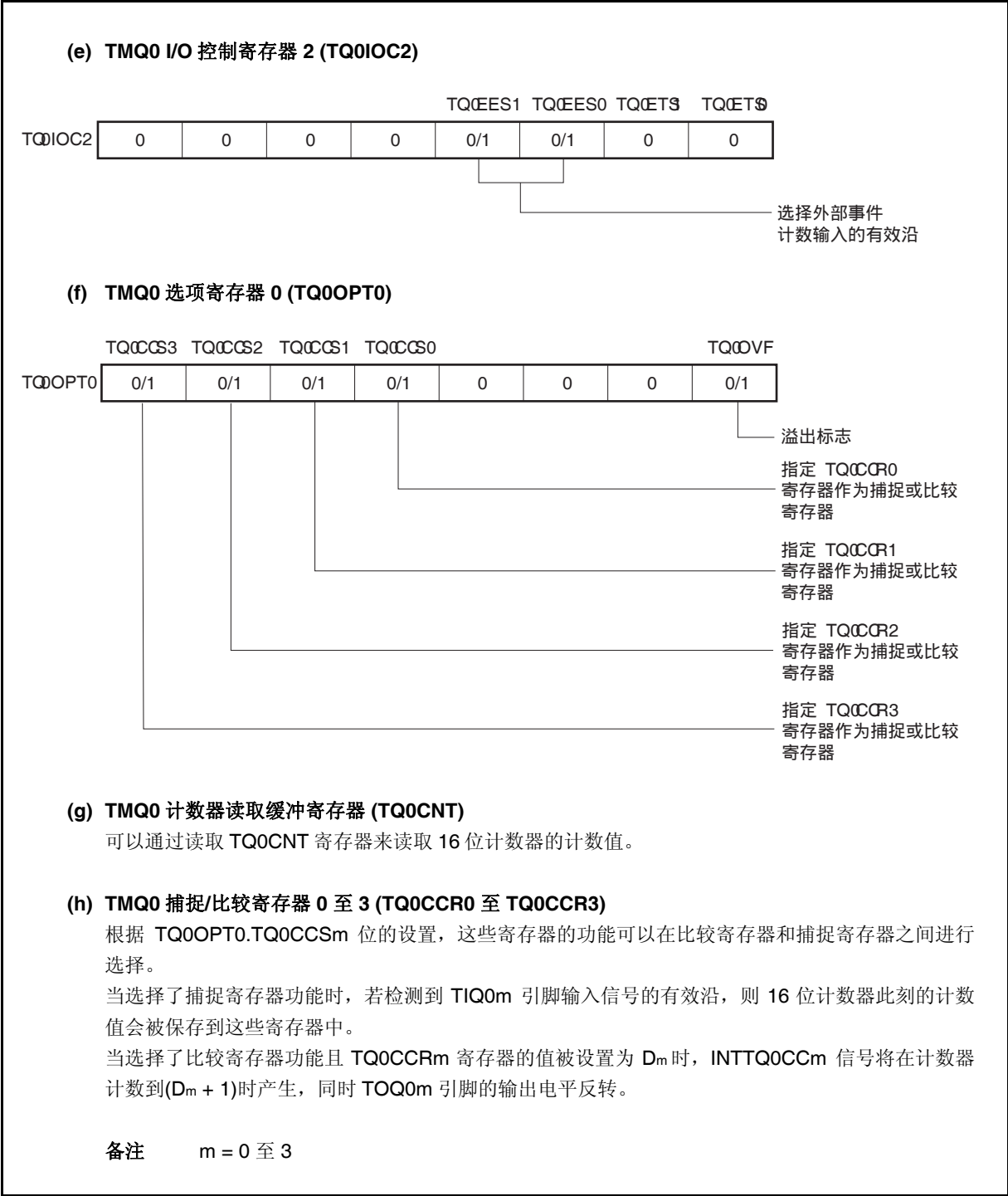


图 8-31. 自由运行定时器模式的寄存器设置 (3/3)



(1) 自由运行定时器模式的操作流程

(a) 当捕捉/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)

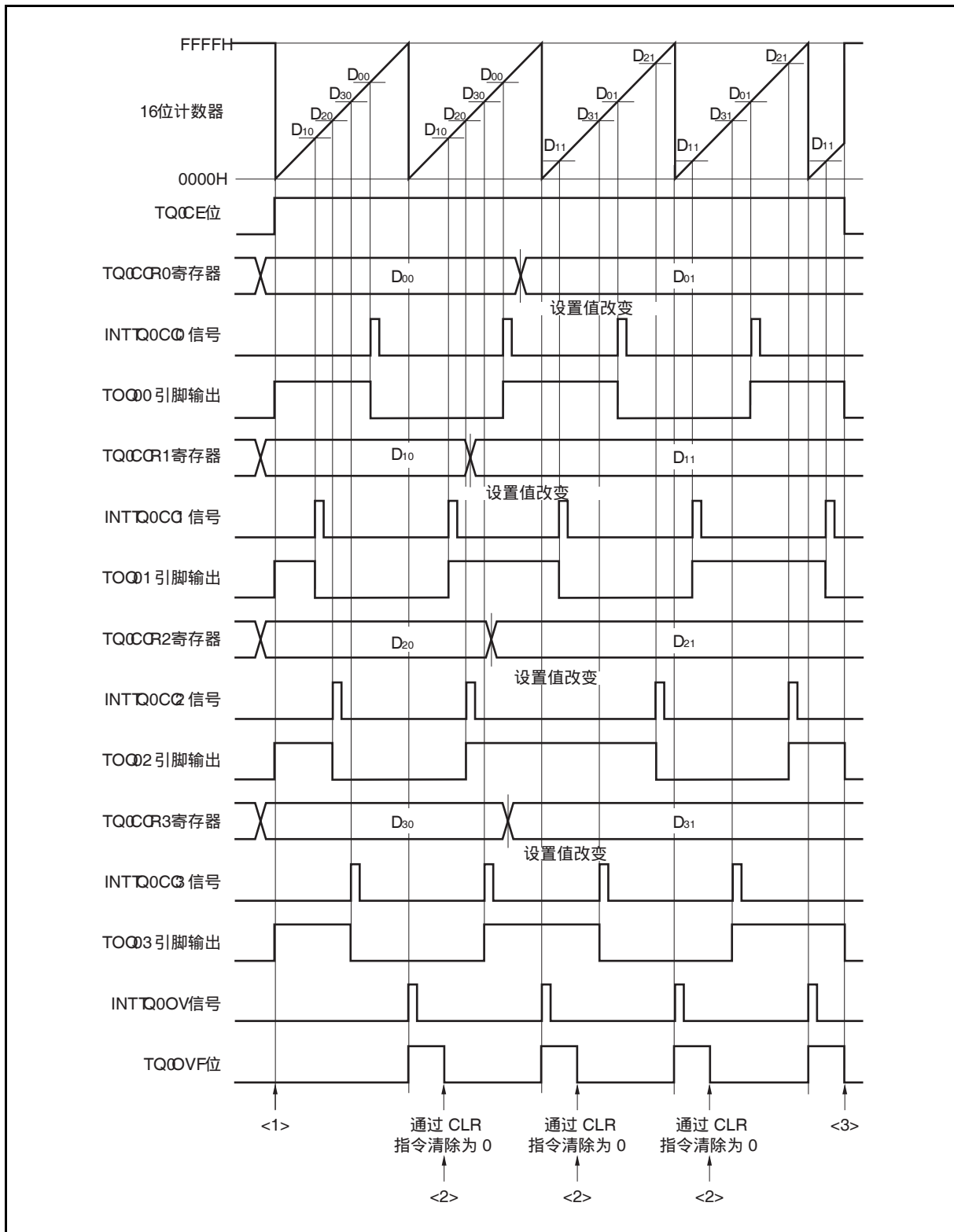
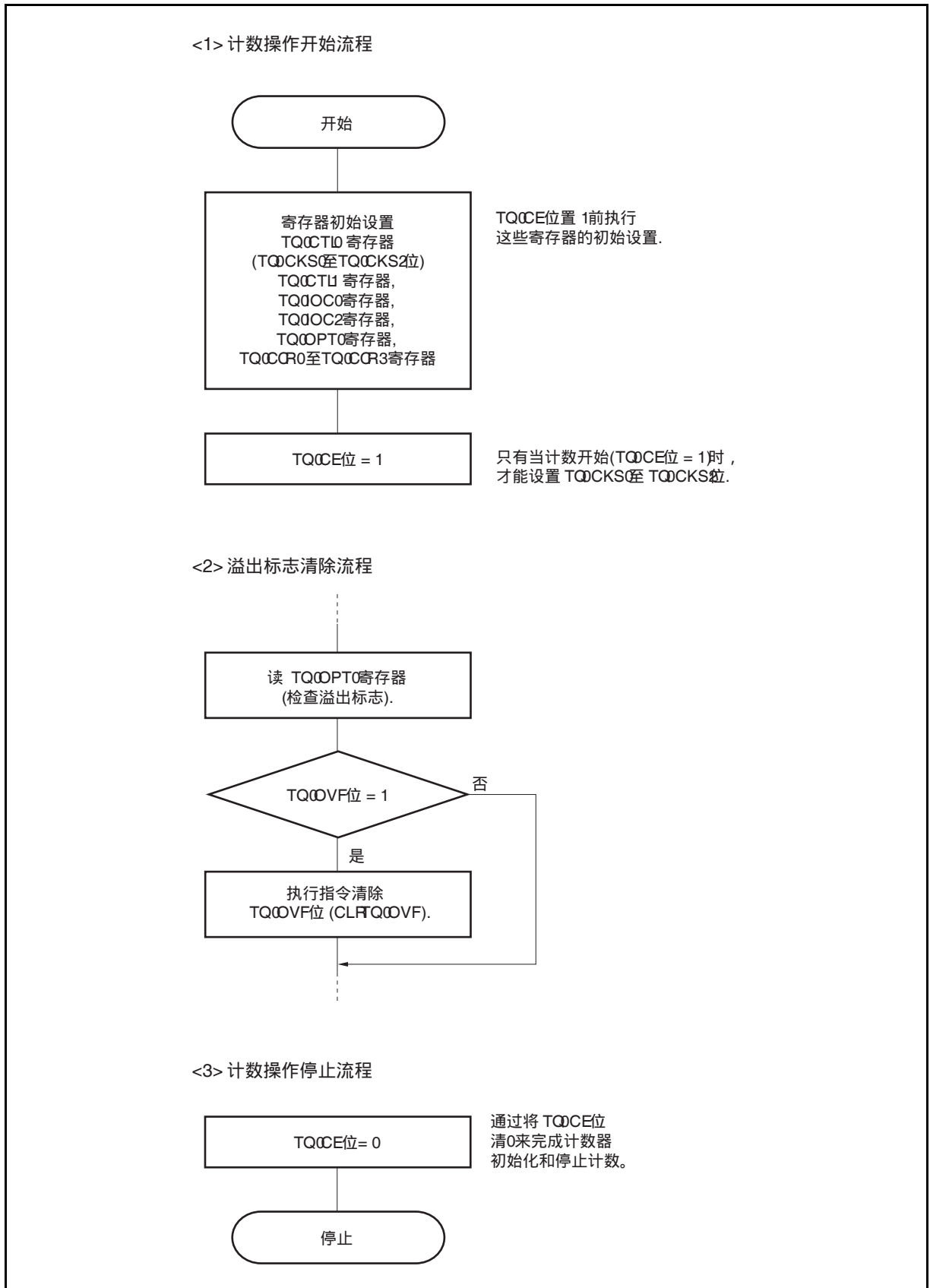


图 8-32. 自由运行定时器模式的软件处理流程(比较功能) (2/2)



(b) 当捕捉/比较寄存器用作捕捉寄存器时

图 8-33. 自由运行定时器模式的软件处理流程 (捕捉功能) (1/2)

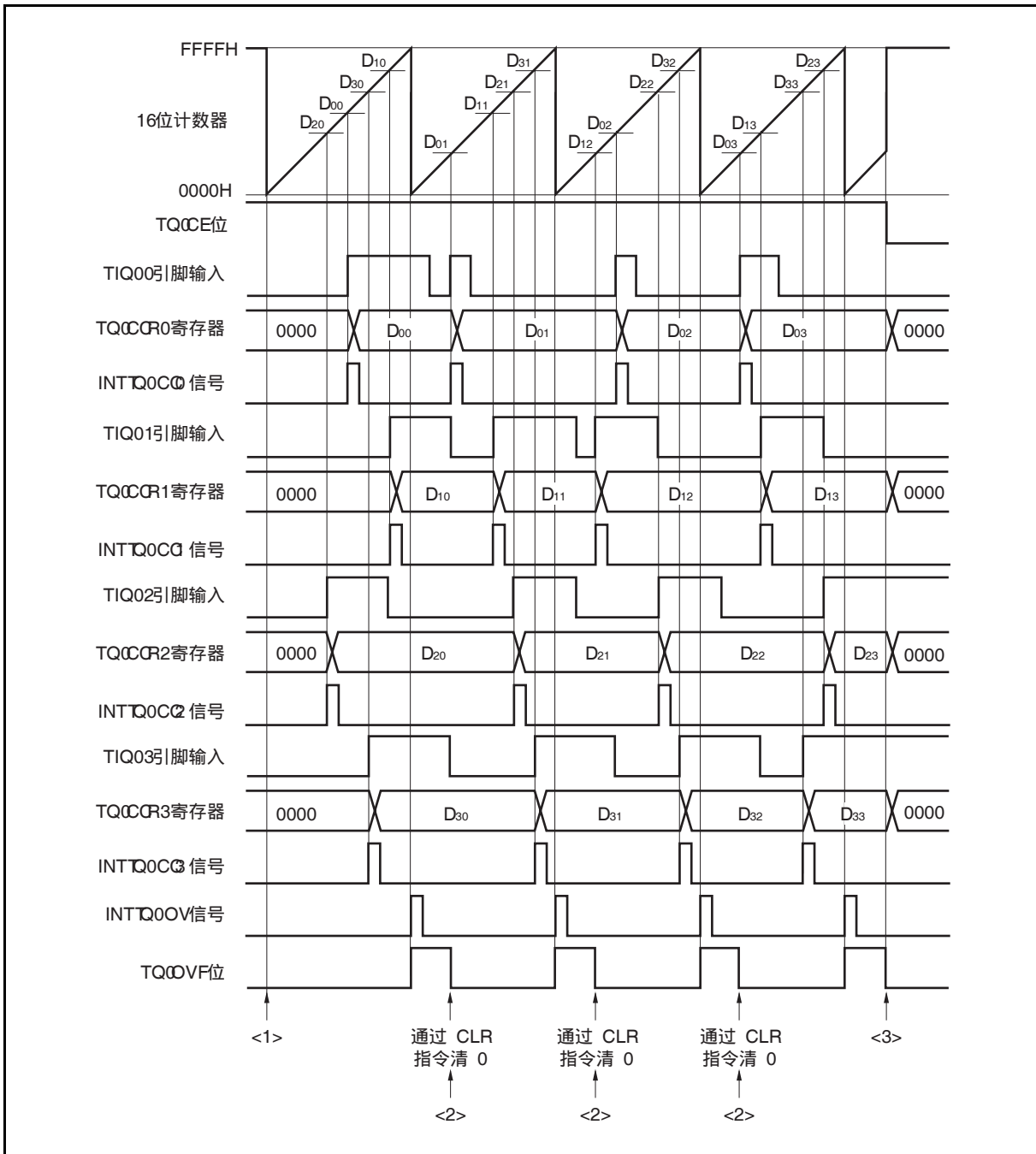
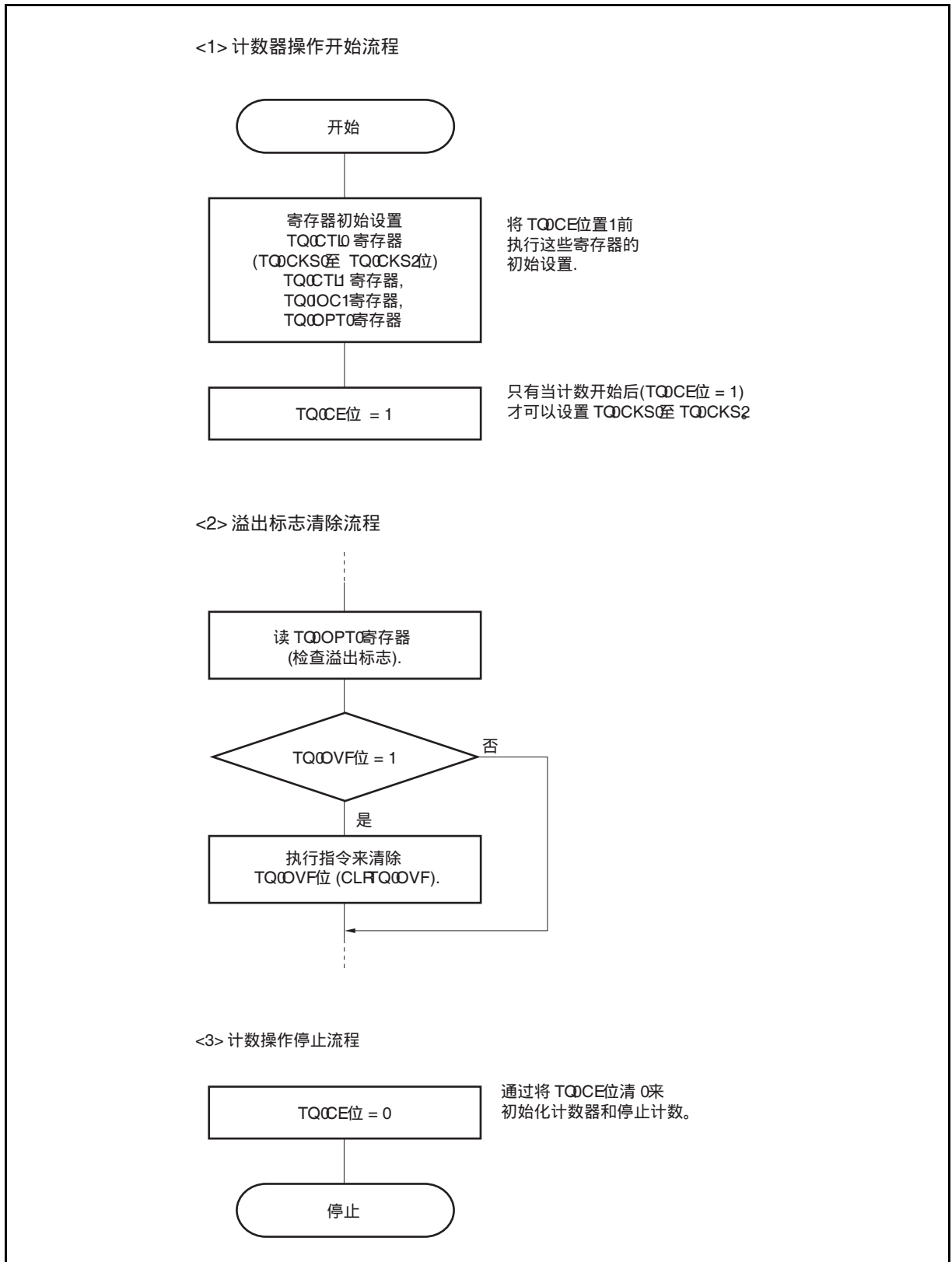


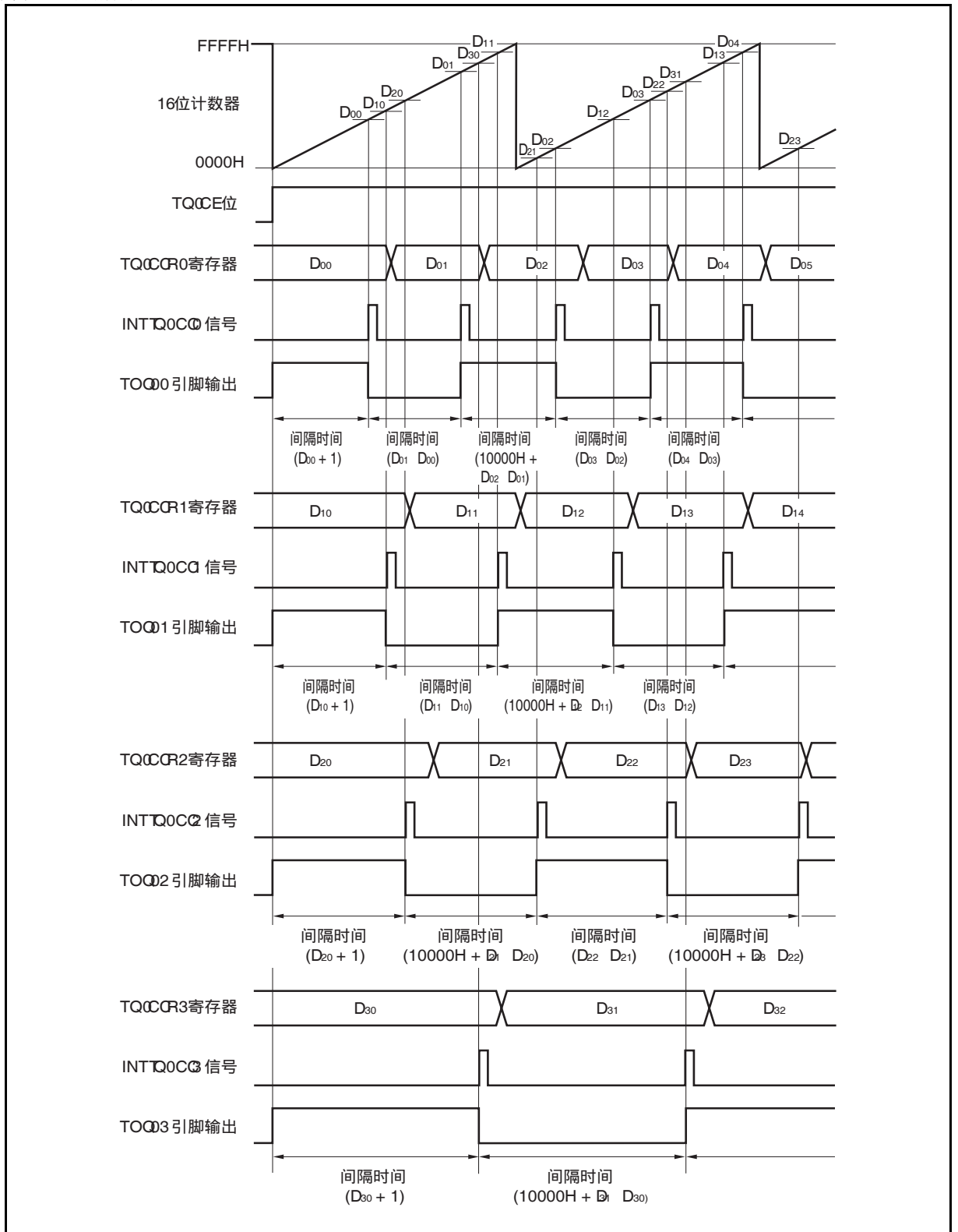
图 8-33. 自由运行定时器模式的软件处理流程 (捕捉功能) (2/2)



(2) 自由运行定时器模式的操作时序

(a) 使用比较寄存器功能的时间间隔操作

当 TQ0CCRm 寄存器被作为比较寄存器使用时, 16 位定时器/事件计数器 Q 可作为间隔定时器使用, 这时, 每当检测到 INTTQ0CCm 信号时, 都需要通过软件处理的方式来设置比较值以指定下一次中断请求产生的间隔时间。



当使用自由运行定时器模式进行间隔时间操作时，可以在同一通道指定四个不同的间隔时间。
进行间隔时间操作时，必须在每次检测到 INTTQ0CCm 信号后执行的中断服务程序中将相应的值设置到 TQ0CCRm 寄存器。
向 TQ0CCRm 寄存器中重新设置的值可由以下算式求出，这里假设间隔时间为“Dm”。

比较寄存器的最初设置值: $D_m - 1$

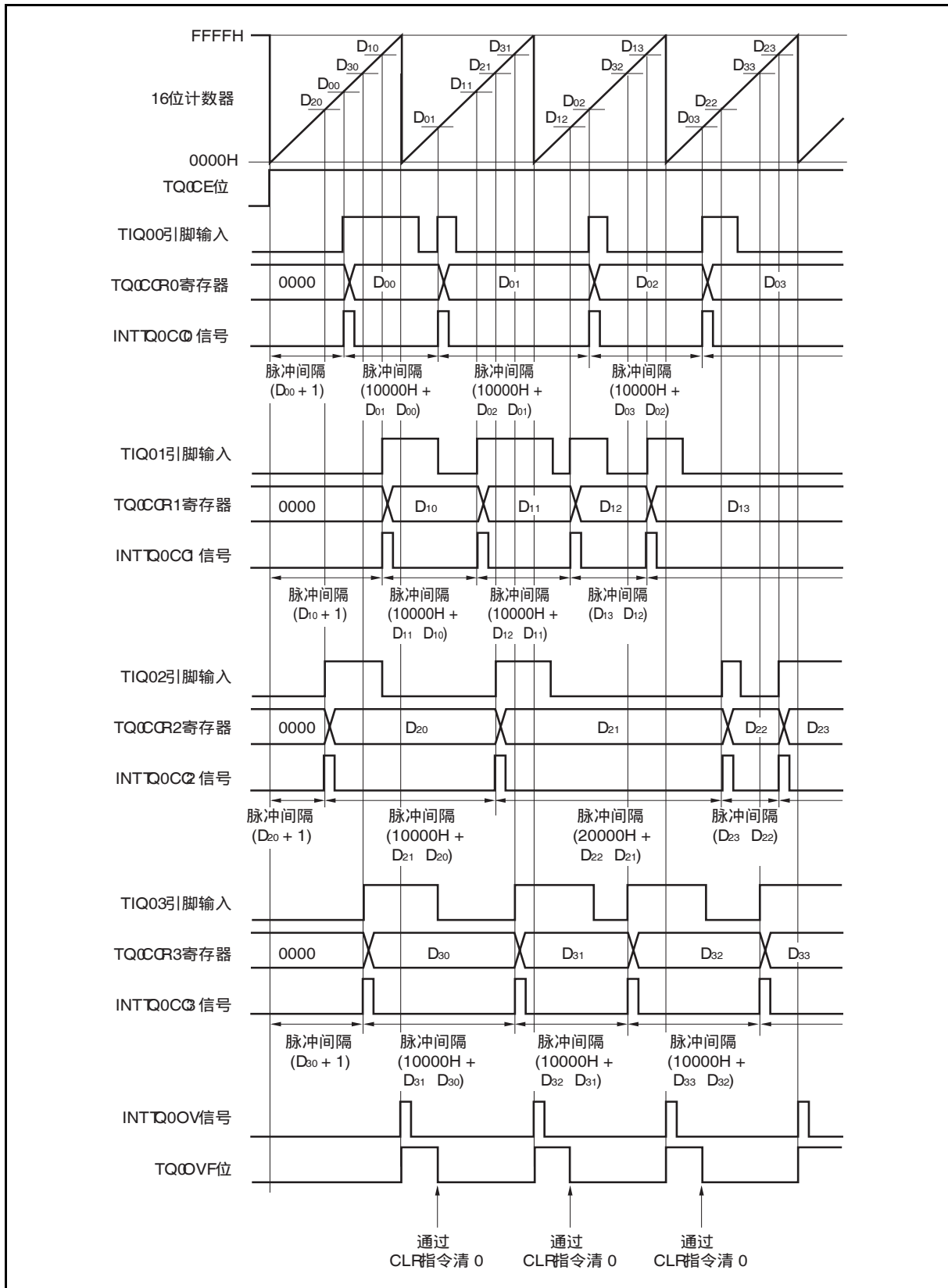
第二次设置比较寄存器以后(含第二次)的设置值: 前次设置值 + D_m

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

备注 m = 0 至 3

(b) 使用捕捉寄存器进行脉宽测量

当 TQ0CCRm 寄存器用作捕捉寄存器进行脉宽测量时，需要通过软件处理在每次检测到中断信号时对捕捉寄存器进行读取并计算间隔时间。



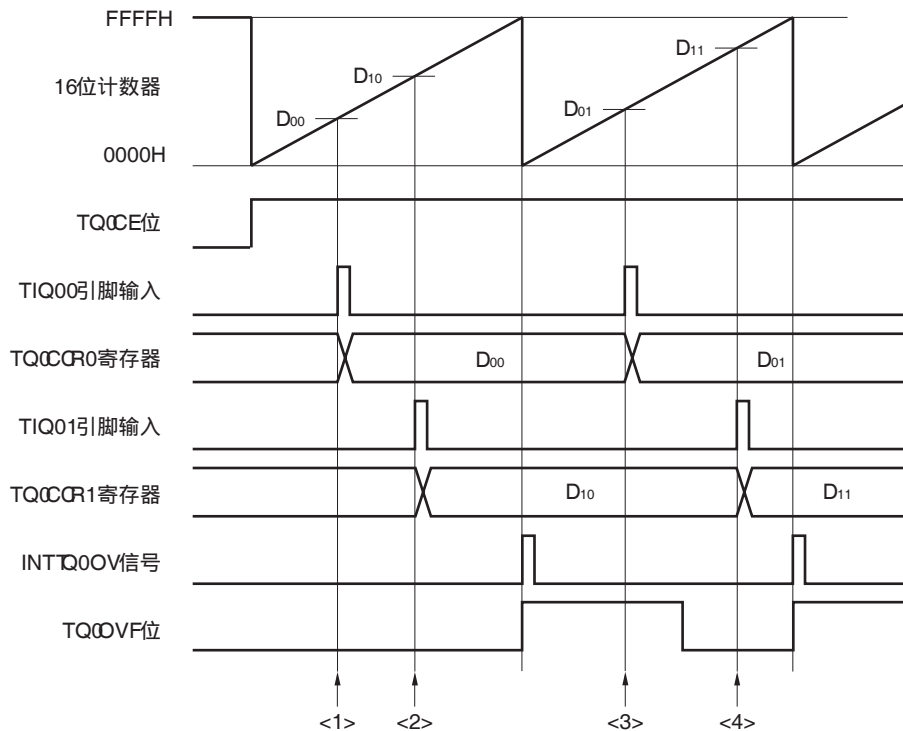
当使用自由运行定时器模式进行脉宽测量操作时，可以在同一通道对四个不同的脉冲宽度进行测量。通过与 $INTTQ0CCm$ 信号同步读取 $TQ0CCRm$ 寄存器的值并计算该值与上一次读取值的差，就可以求得脉冲的宽度。

备注 $m = 0$ 至 3

(c) 使用两个及以上捕捉寄存器产生溢出时的操作

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

使用两个及以上捕捉寄存器时的错误处理举例



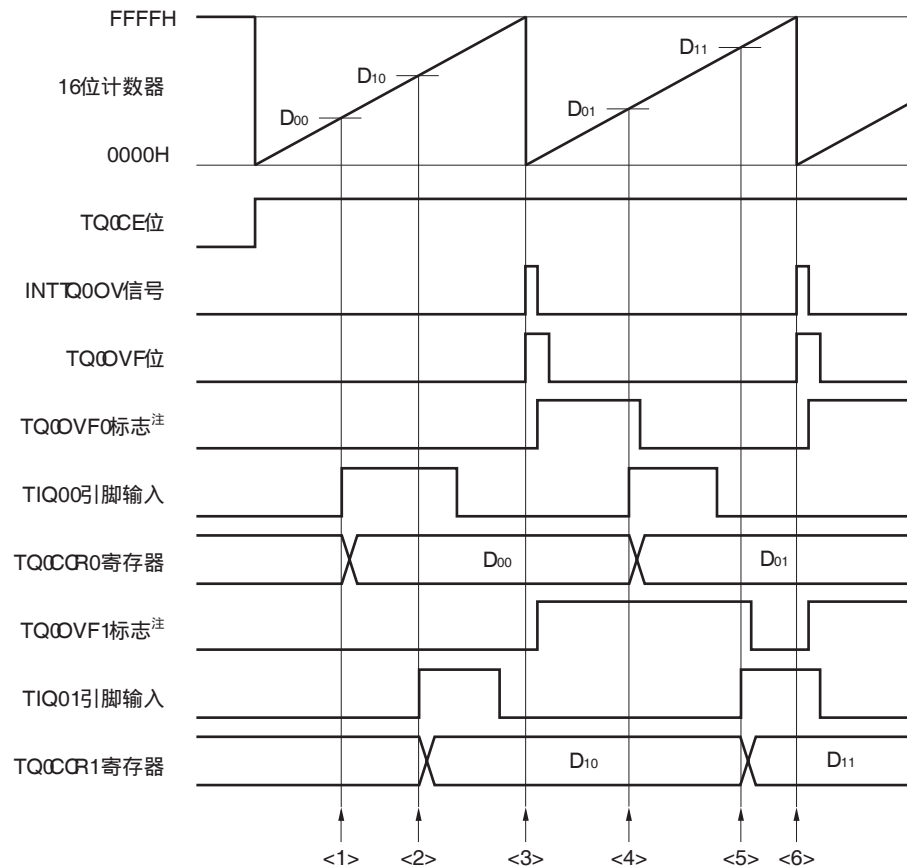
使用自由运行定时器模式对两个脉冲宽度进行测量时，可能出现以下的问题。

- <1> 读取 TQ0CCR0 寄存器 (设置为 TIQ00 引脚输入的初始值)。
- <2> 读取 TQ0CCR1 寄存器 (设置为 TIQ01 引脚输入的初始值)。
- <3> 读取 TQ0CCR0 寄存器。
读取溢出标志。如果溢出标志为 1，那么将其清零。
由于此时溢出标志位为 1，所以脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。
- <4> 读取 TQ0CCR1 寄存器
读取溢出标志。由于标志位在步骤<3>中被清零，所以此时读取的值为 0。
由于溢出标志位为 0，所以脉冲宽度按照 $(D_{11} - D_{10})$ (错误)来计算。

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

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

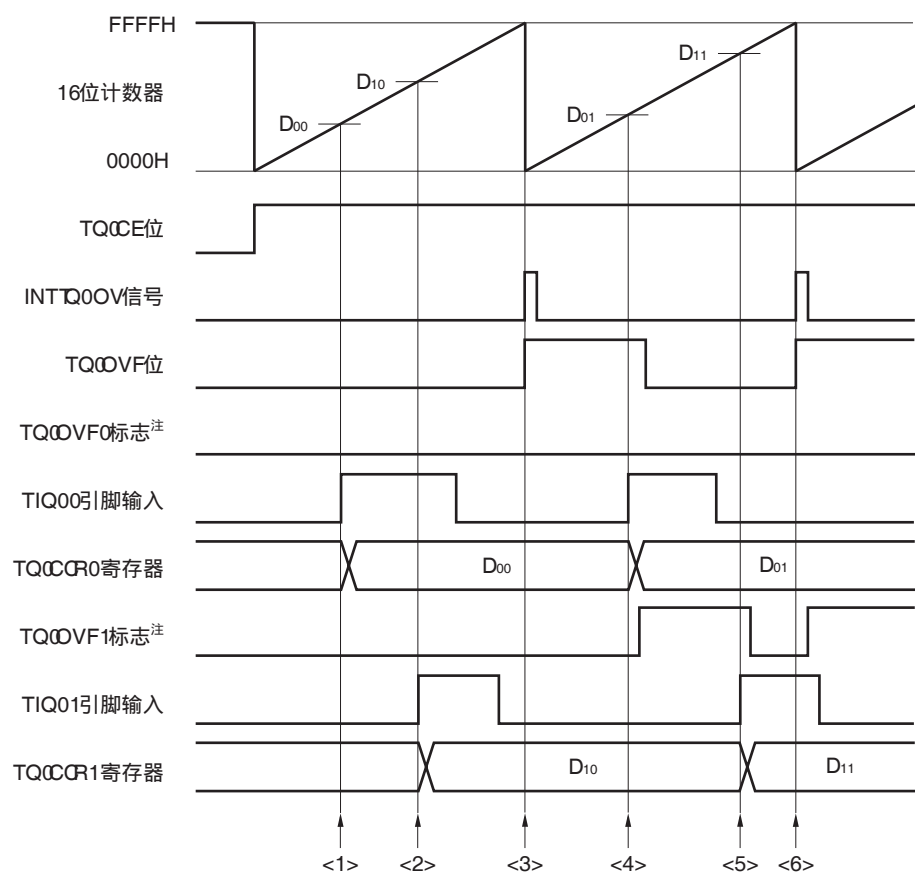
使用两个捕捉寄存器时的处理举例(利用溢出中断)



注 TQ0OVF0 和 TQ0OVF1 标志通过软件设置于内部 RAM。

- <1> 读取 TQ0CCR0 寄存器 (设置为 TIQ00 引脚输入的初始值)。
- <2> 读取 TQ0CCR1 寄存器 (设置为 TIQ01 引脚输入的初始值)。
- <3> 产生了溢出。在溢出中断服务程序中将 TQ0OVF0 和 TQ0OVF1 标志位置 1，同时将硬件溢出标志位清零。
- <4> 读取 TQ0CCR0 寄存器。
读取 TQ0OVF0 标志位。如果该标志位为 1，那么将其清零。
由于 TQ0OVF0 标志位为 1，脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。
- <5> 读取 TQ0CCR1 寄存器。
读取 TQ0OVF1 标志位，如果该标志位为 1，那么将其清零。(步骤<4>中将 TQ0OVF0 位清零但 TQ0OVF1 标志始终保持为 1。)
由于 TQ0OVF1 标志为 1，脉冲宽度按照 $(10000H + D_{11} - D_{10})$ (正确)来计算。
- <6> 同步骤 <3>

使用两个捕捉寄存器时的处理举例(不利用溢出中断)

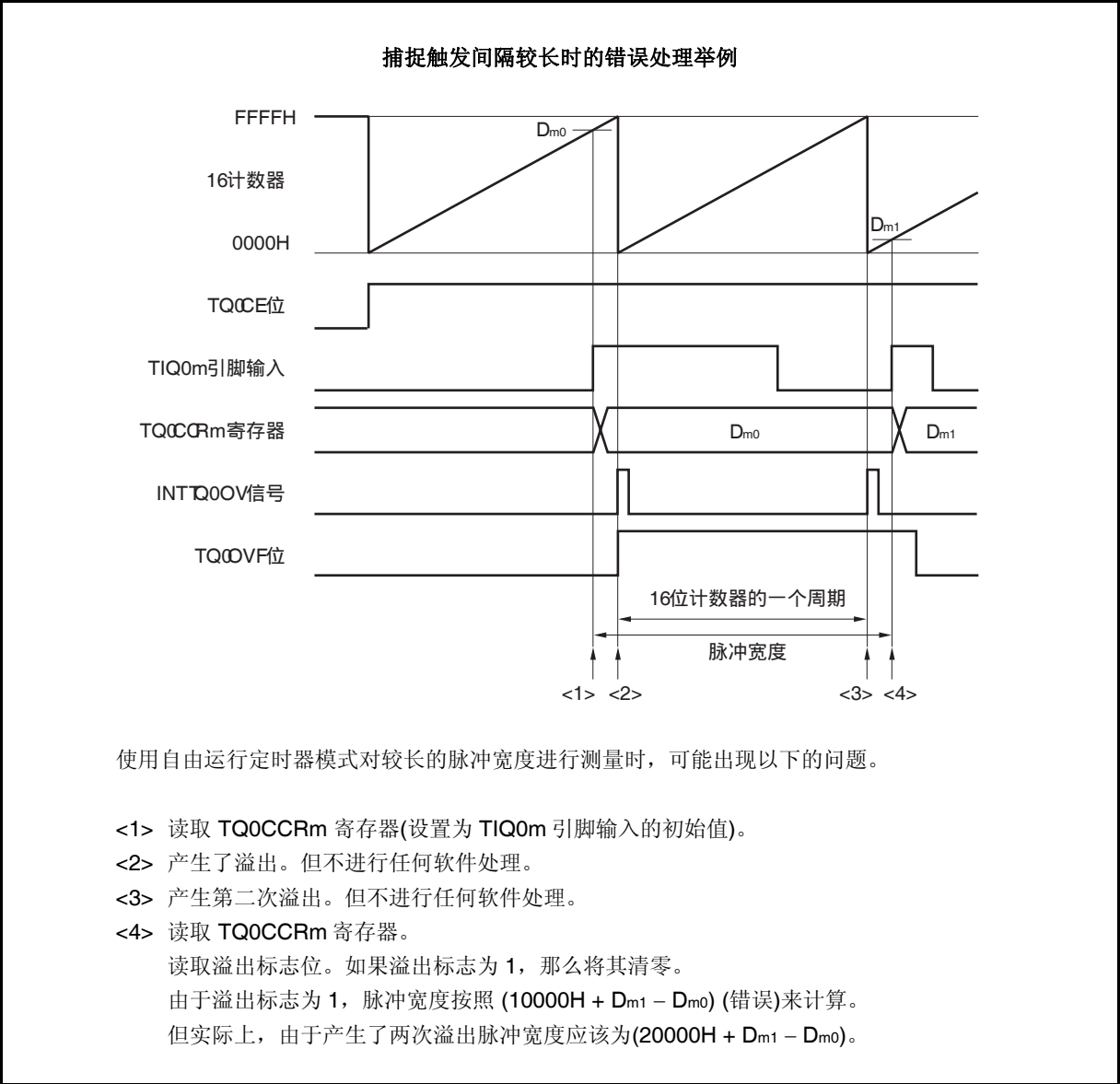


注 TQ0OVF0 和 TQ0OVF1 标志通过软件设置于内部 RAM。

<1> 读取 TQ0CCR0 寄存器 (设置为 TIQ00 引脚输入的初始值)。
<2> 读取 TQ0CCR1 寄存器 (设置为 TIQ01 引脚输入的初始值)。
<3> 产生了溢出。但不进行任何软件处理。
<4> 读取 TQ0CCR0 寄存器。
读取溢出标志。如果溢出标志为 1，那么只把 TQ0OVF1 标志设置为 1，同时将溢出标志清零。
由于 TQ0OVF0 标志位为 1，脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。
<5> 读取 TQ0CCR1 寄存器。
读取溢出标志。由于溢出标志位在步骤<4>中被清零，所以读取结果为 0。
读取 TQ0OVF1 位。如果 TQ0OVF1 标志为 1，那么将其清零。
由于 TQ0OVF1 标志为 1，脉冲宽度按照 $(10000H + D_{11} - D_{10})$ (正确)来计算。
<6> 同步骤 <3>

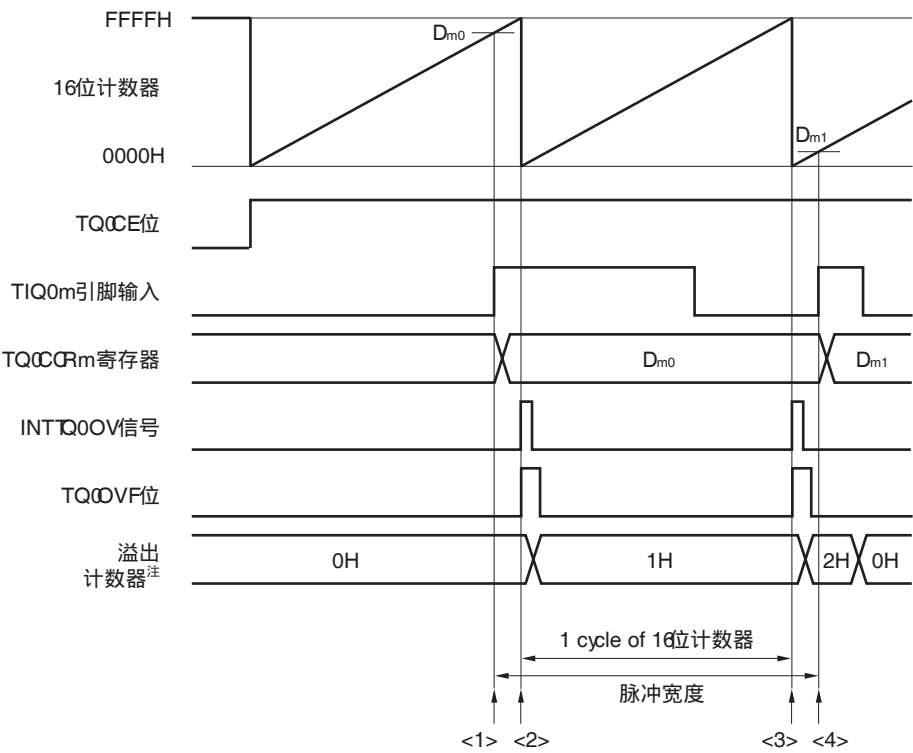
(d) 捕捉触发间隔较长时的溢出处理

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



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

捕捉触发间隔较长时的处理举例

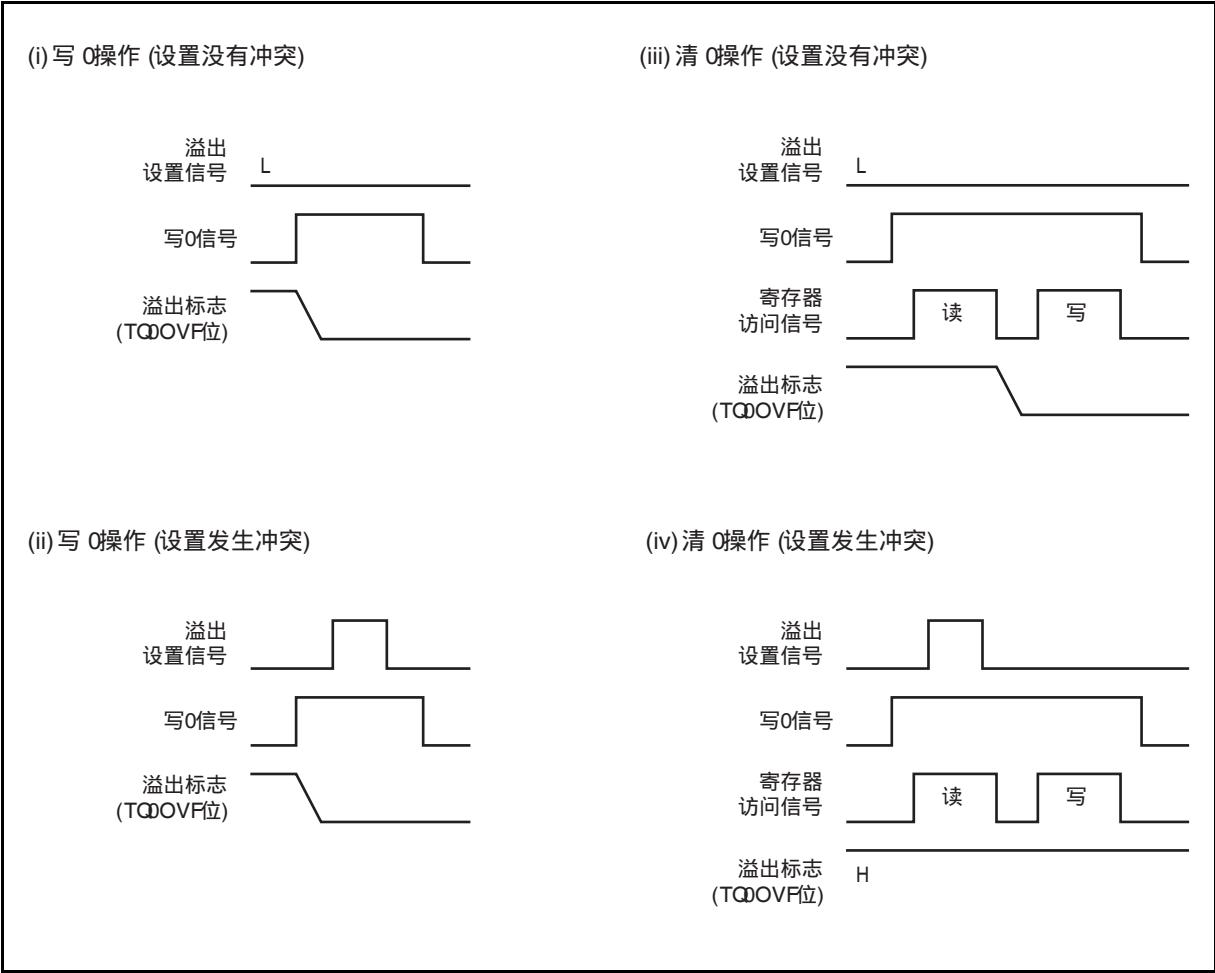


注 这里的溢出次数计数器是建立于内置 RAM 上的变量，可由软件任意进行设置。

- <1> 读取 TQ0CCRm 寄存器 (设置为 TIQ0m 引脚输入的初始值)。
- <2> 产生了溢出。在溢出中断服务程序中将溢出次数计数器加 1 并将溢出标志清零。
- <3> 产生了第二次溢出。在溢出中断服务程序中将溢出次数计数器加 1 并将溢出标志清零。
- <4> 读取 TQ0CCRm 寄存器。
读取溢出次数计数器的值。
→ 假设此时溢出计数器的值为“N”，则脉冲宽度可按 $(N \times 10000H + D_{m1} - D_{m0})$ 计算求得。
在本例中，由于溢出产生了两次，所以脉冲宽度为 $(20000H + D_{m1} - D_{m0})$ 。
将溢出次数计数器清零(0H)。

(e) 将溢出标志位清零

可以通过使用 CLR 指令清零 TQ0OVF 位或向 TQ0OPT0 寄存器写入 8 位数据(第 0 位为 0)的方法将溢出标志位清零。为了能够准确检测到溢出的发生，请务必在每次读取到 TQ0OVF 位为 1 后，立即使用位操作指令将该位清零。



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

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

8.5.7 脉宽测量模式(TQ0MD2 至 TQ0MD0 位 = 110)

在脉宽测量模式中，TQ0CTL0.TQ0CE 位被置 1 后，16 位计数器 Q 开始计数。每当检测到 TIQ0m 引脚输入信号的有效沿时，16 位计数器的计数值就会被保存到 TQ0CCRm 寄存器中，同时计数器清零为 0000H。

两个有效沿之间的间隔时间可以在捕捉中断请求信号 (INTTQ0CCm) 产生后通过读取 TQ0CCRm 寄存器的值来计算。

可以选择 TIQ00 至 TIQ03 引脚中的任意一个作为捕捉触发输入引脚。通过 TQ0IOC1 寄存器将未使用的引脚设置为“无边沿检测”。

当使用外部时钟来测量脉宽时，只能使用 TIQ0k 引脚作为捕捉触发输入引脚，因为外部时钟输入已经固定为 TIQ00 引脚。此时要将 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位清为 00 (捕捉触发输入(TIQ00 引脚): 无边沿检测)。

备注 m = 0 至 3
 k = 1 至 3

图 8-34. 脉宽测量模式的配置图

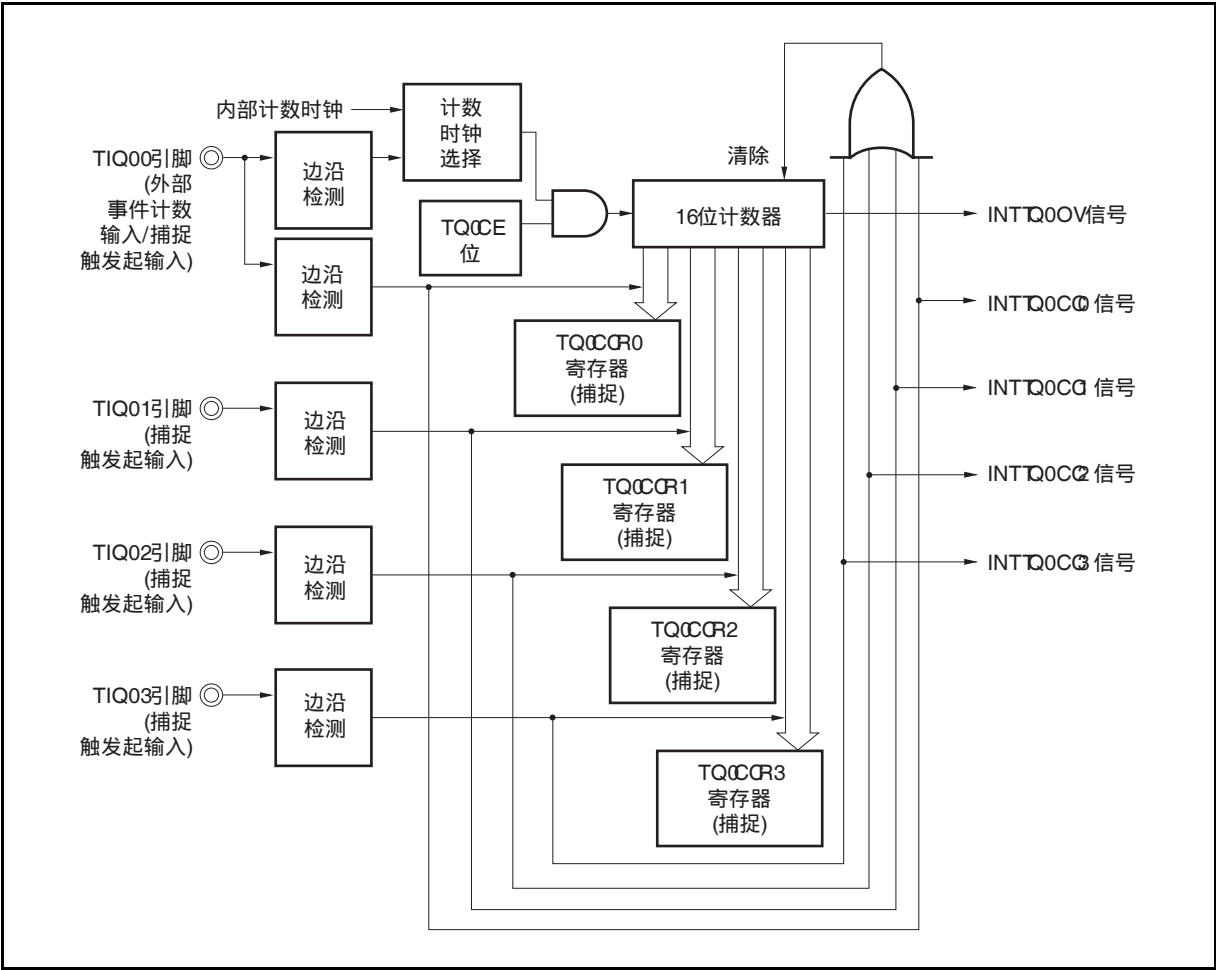
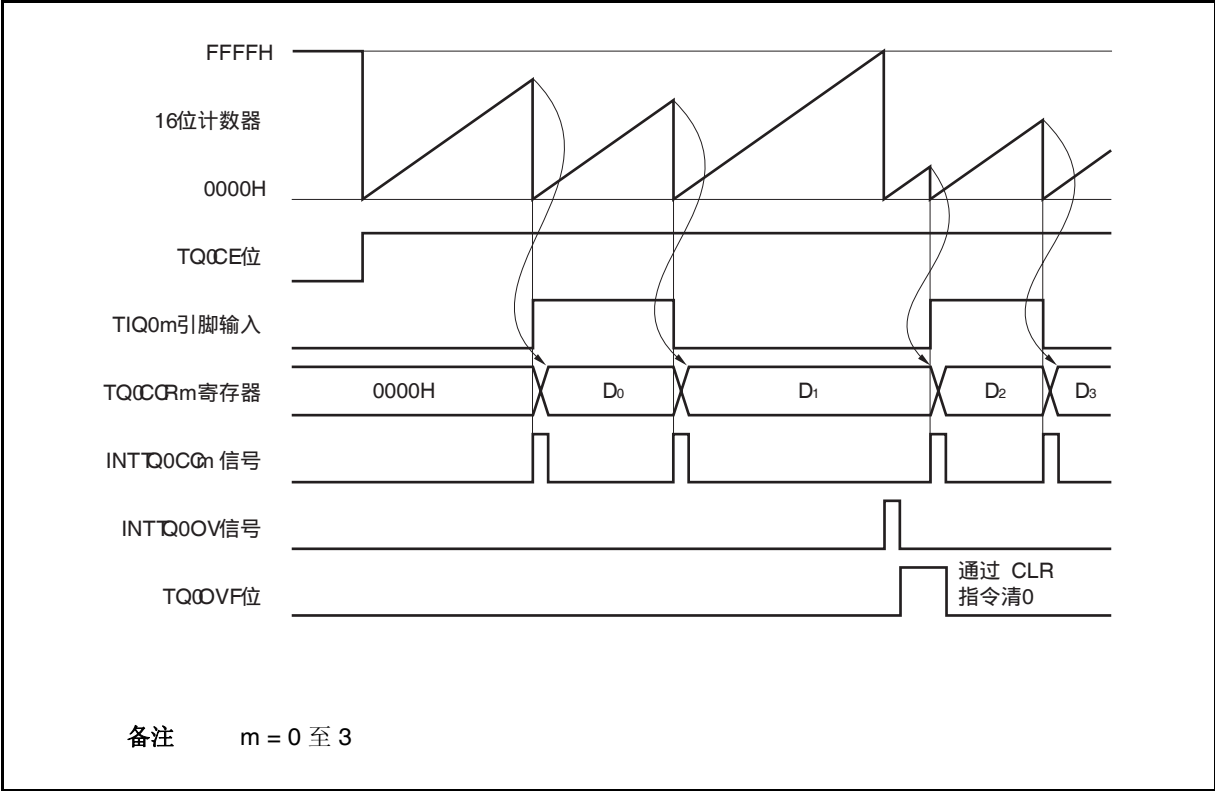


图 8-35. 脉宽测量模式的基本时序



TQ0CE 位被置 1 后，16 位计数器开始计数。当检测到 TIQ0m 引脚的有效沿时，16 位计数器的计数值将被保存到 TQ0CCRm 寄存器中，同时计数器被清零并产生一个捕捉中断请求信号(INTTQ0CCm)。

脉冲宽度可由下式求得。

脉冲宽度 = 捕捉计数值 × 计数时钟周期

如果直到 16 位计数器计数到 FFFFH 仍没有检测到 TIQ0m 引脚的有效沿，那么将在下一个时钟时产生一个溢出中断请求信号 (INTTQ0OV)，同时计数器被清零并重新开始计数。这种情况下，溢出标志位(TQ0OPT0.TQ0OVF 位) 也将被设置为 1。请通过软件使用 CLR 指令将溢出标志清零。

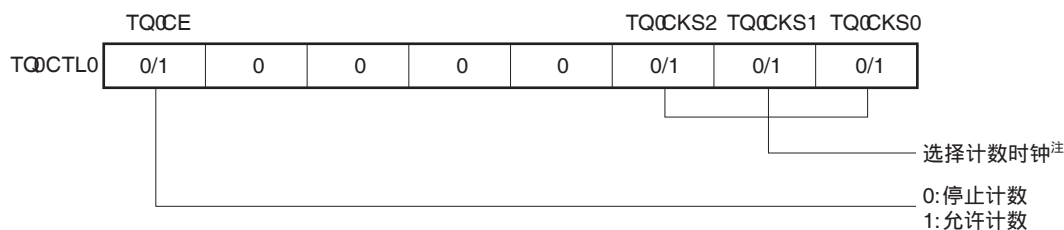
如果溢出标志被设置为 1，那么脉冲宽度可由下式求得。

脉冲宽度 = (10000H × TQ0OVF 位被设置为 1 的次数 + 捕捉计数值) × 计数时钟周期

备注 m = 0 至 3

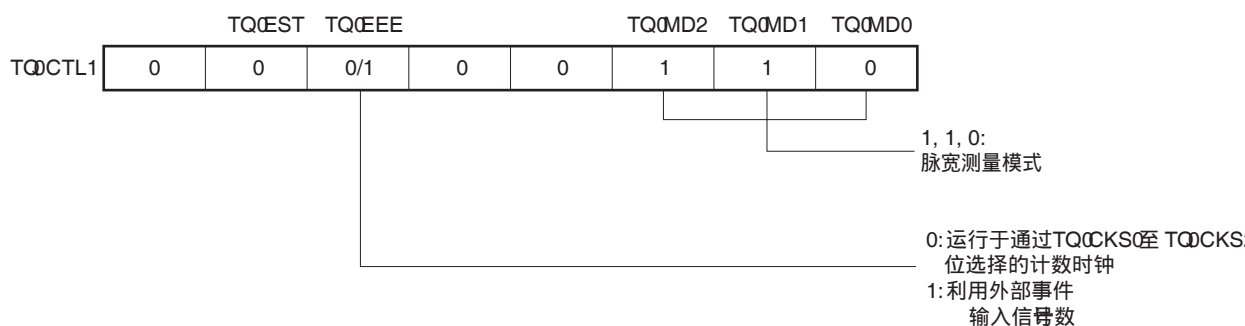
图 8-36. 脉宽测量模式的寄存器设置 (1/2)

(a) TMQ0 控制寄存器 0 (TQ0CTL0)

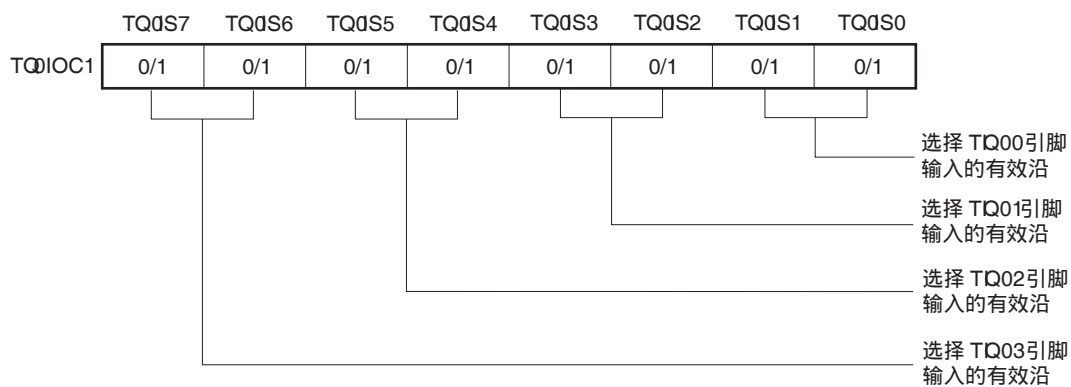


注 TQ0EEE 位 = 1 时该设置无效。

(b) TMQ0 控制寄存器 1 (TQ0CTL1)



(c) TMQ0 I/O 控制寄存器 1 (TQ0IOC1)



(d) TMQ0 I/O 控制寄存器 2 (TQ0IOC2)

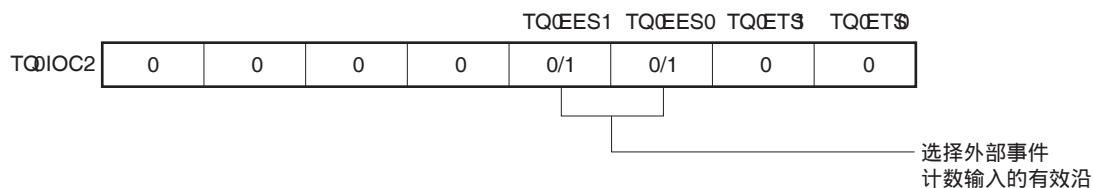
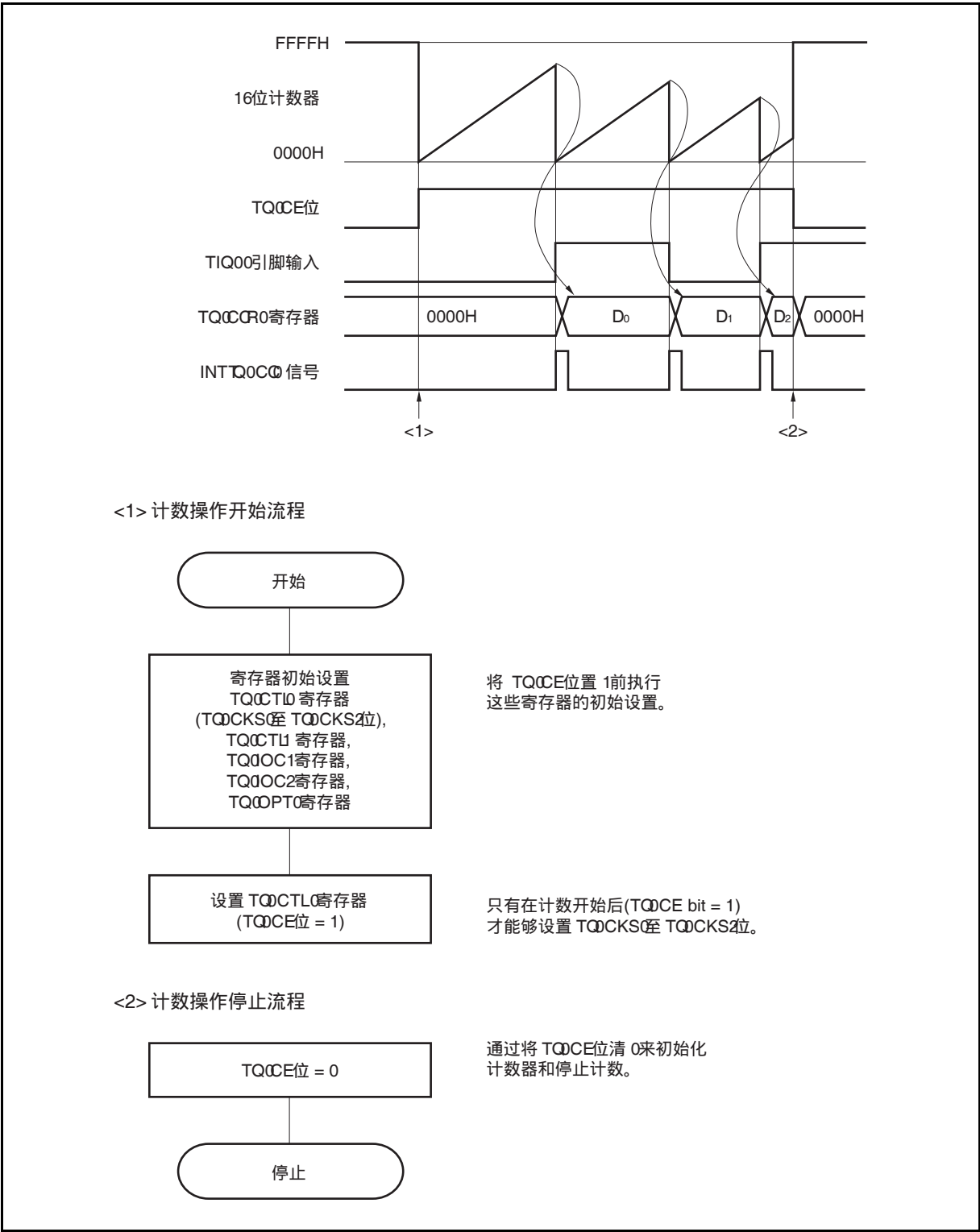


图 8-36. 脉宽测量模式的寄存器设置 (2/2)



(1) 脉宽测量模式的操作流程

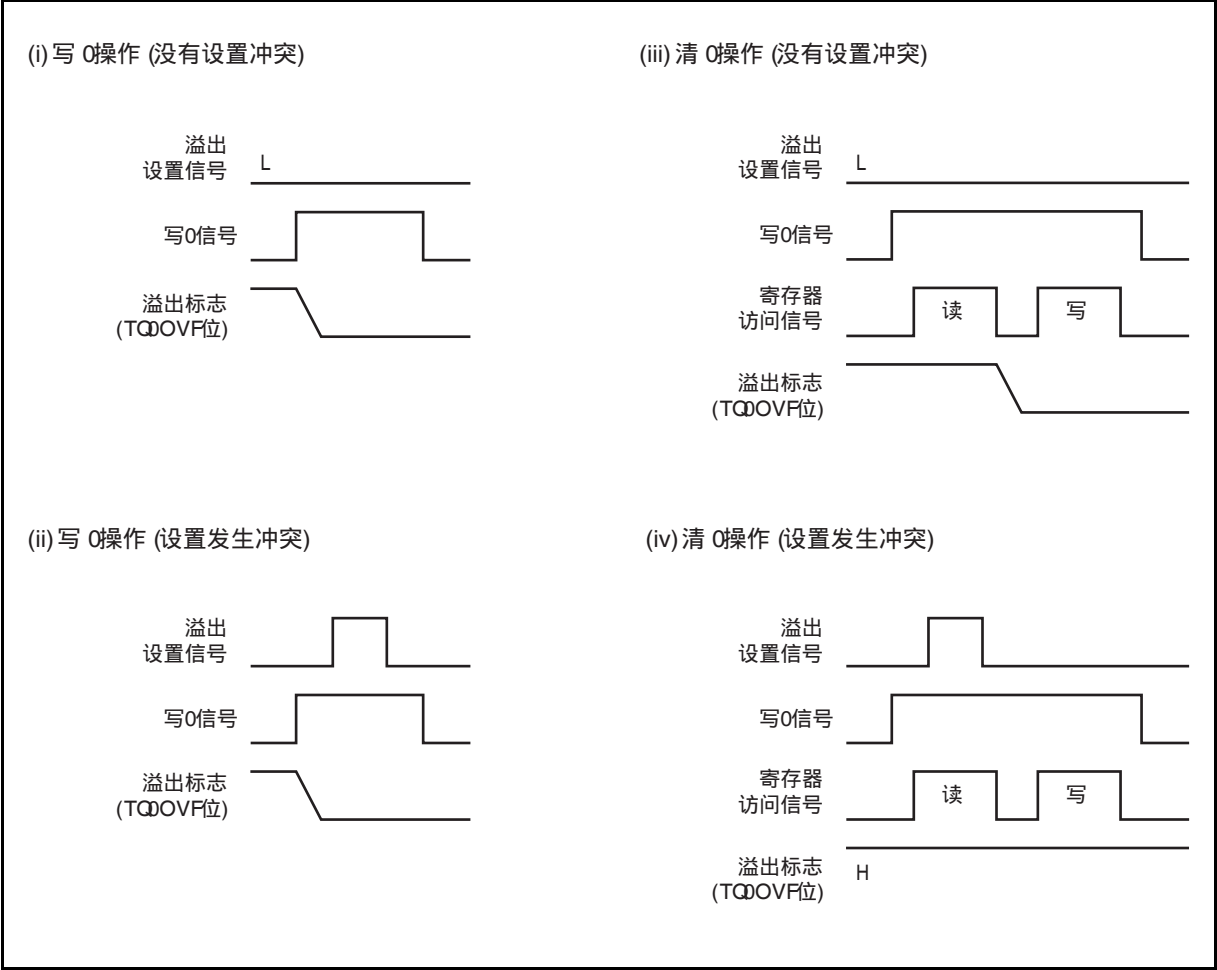
图 8-37. 脉宽测量模式的软件处理流程



(2) 脉宽测量模式的操作时序

(a) 将溢出标志位清零

可以通过使用 CLR 指令清零 TQ0OVF 位或向 TQ0OPT0 寄存器写入 8 位数据(第 0 位为 0)的方法将溢出标志位清零。为了能够准确检测到溢出的发生，请务必在每次读取到 TQ0OVF 位为 1 后，立即使用位操作指令将该位清零。



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

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

8.5.8 定时器输出操作

TOQ00 和 TOQ01 引脚的工作模式与输出电平的关系如下表所示。

表 8-6. 各模式下的定时器输入控制

操作模式	TOQn0 引脚	TOQn1 引脚	TOQn2 引脚	TOQn3 引脚
间隔定时器模式	方波输出			
外部事件计数模式	方波输出	—		
外部触发脉冲输出模式	方波输出	外部触发脉冲输出	外部触发脉冲输出	外部触发脉冲输出
单脉冲输出模式		单脉冲输出	单脉冲输出	单脉冲输出
PWM 输出模式		PWM 输出	PWM 输出	PWM 输出
自由运行定时器模式	方波输出 (仅限使用比较功能时)			
脉宽测量模式	—			

表 8-7. TOQ00 至 TOQ03 引脚在定时器输出控制位控制下的真值表

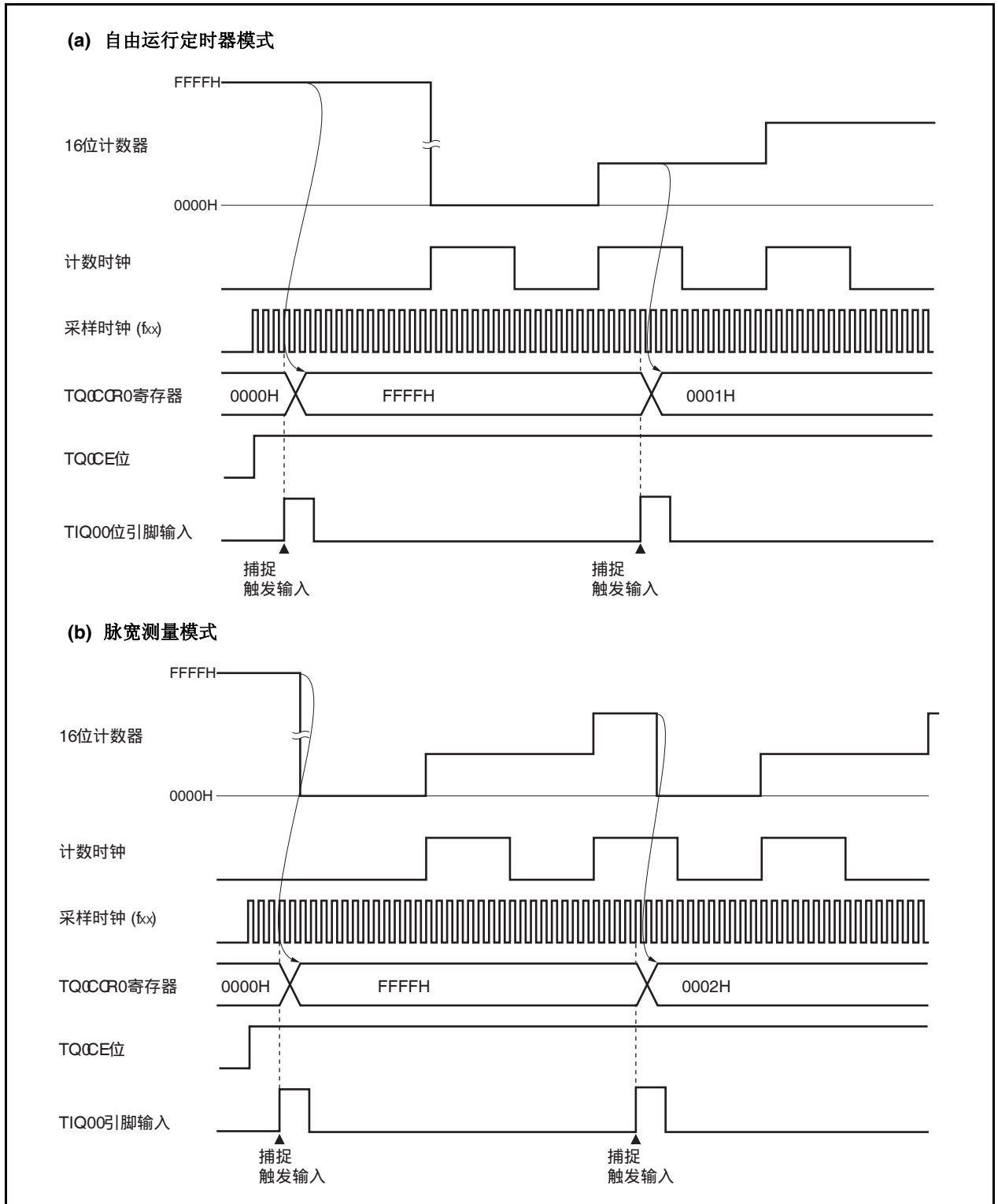
TQ0IOC0.TQ0OLm 位	TQ0IOC0.TQ0OEm 位	TQ0CTL0.TQ0CE 位	TOQ0m 引脚的电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前为低电平，计数过程中为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平，计数过程中为低电平

备注 m = 0 至 3

8.6 注意事项

(1) 捕捉操作

若在捕捉操作时使用了一个较慢的时钟信号作为计数时钟，当 **TQ0CE** 位被设置为 1 后，立刻检测到捕捉触发信号时 **TQ0CCR0**, **TQ0CCR1**, **TQ0CCR2**, 和 **TQ0CCR3** 寄存器中捕捉的计数值将不会是 0000H 而是 FFFFH。



第九章 16 位间隔定时器 M (TMM)

9.1 概述

- 间隔功能
- 8 种时钟可选
- 16 位计数器 × 1
(定时器计数操作过程中不可对 16 位计数器进行读取。)
- 比较寄存器 × 1
(定时器计数操作过程中不可向比较寄存器写入数据。)
- 比较匹配中断 × 1

定时器 M 只支持清除&启动模式(clear & start)。不支持自由运行模式。

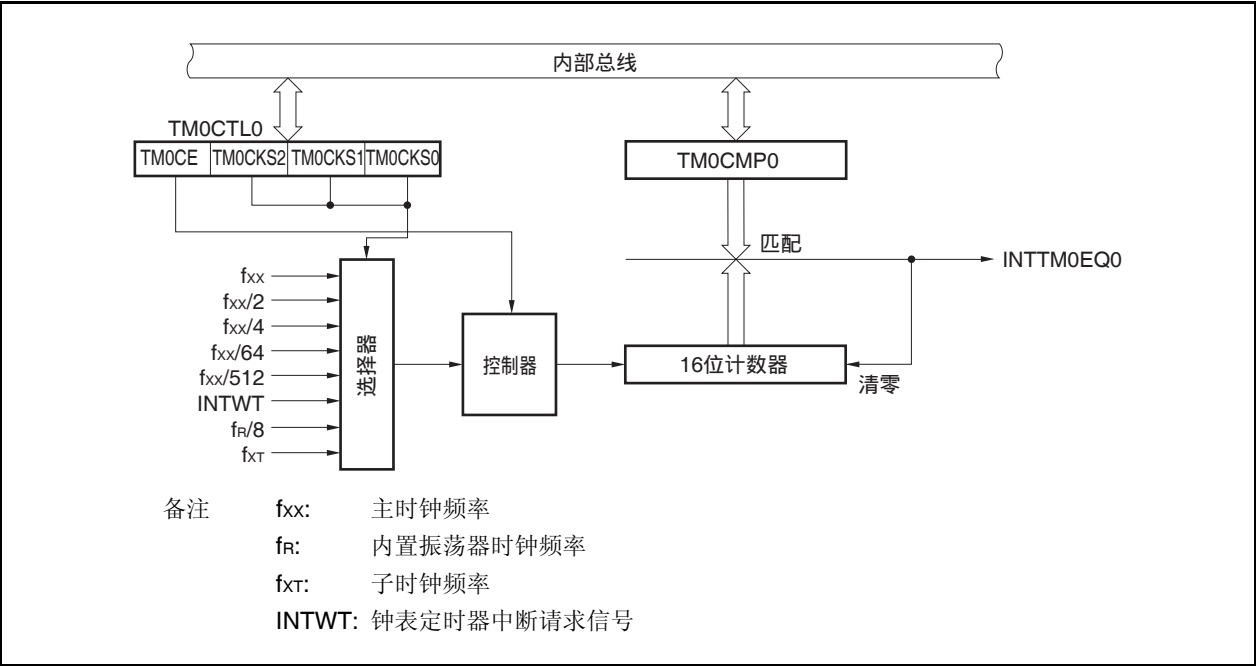
9.2 配置

TMM0 包含以下硬件。

表 9-1. TMM0 的配置

器件	配置
定时器寄存器	16 位计数器
寄存器	TMM0 比较寄存器 0 (TM0CMP0)
控制寄存器	TMM0 控制寄存器 0 (TM0CTL0)

图 9-1. TMM0 的框图

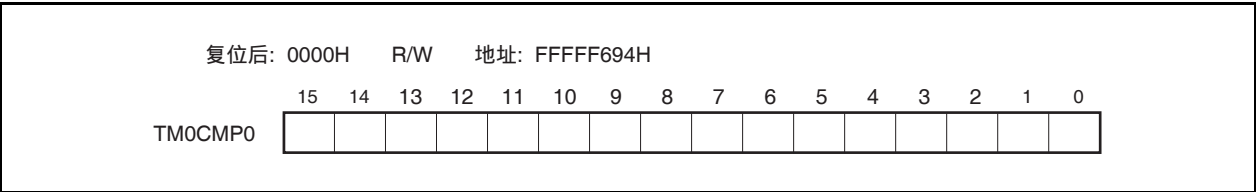


(1) 16 位计数器

该计数器是一个按内部时钟计数的 16 位计数器。
该计数器不可以被读写。

(2) TMM0 比较寄存器 0 (TM0CMP0)

TM0CMP0 寄存器是一个 16 位的比较寄存器。
可以对该寄存器进行 16 位的读写操作。
复位输入将把该寄存器设置为 0000H。
可使用软件向 TM0CMP0 寄存器写入相同的数值。
当 TM0CTL0.TM0CE 位 = 1 时禁止改写 TM0CMP0 寄存器的值。



9.3 寄存器

(1) TMM0 控制寄存器 (TM0CTL0)

TM0CTL0 寄存器是用于控制 TMM0 操作的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

可使用软件向 TM0CTL0 寄存器写入相同的数值。

复位后: 00H R/W 地址: FFFFF690H

	<7>	6	5	4	3	2	1	0
TM0CTL0	TM0CE	0	0	0	0	TM0CKS2	TM0CKS1	TM0CKS0

TM0CE	指定是否允许内部时钟操作
0	禁止TMM0操作 (16位计数器异步复位)。 停止操作时钟应用。
1	允许TMM0操作。 开始操作时钟的应用。TMM0开始工作。
可通过TM0CE位对TMM0的内部时钟控制和内部电路复位进行异步操作。 当TM0CE位被清零时, TMM0的内部时钟被停止(固定为低电平) 同时16位计数器被异步清零。	

TM0CKS2	TM0CKS1	TM0CKS0	计数时钟选择
0	0	0	f _{xx}
0	0	1	f _{xx} /2
0	1	0	f _{xx} /4
0	1	1	f _{xx} /64
1	0	0	f _{xx} /512
1	0	1	INTWT
1	1	0	f _R /8
1	1	1	f _{XT}

- 注意**
1. 在 TM0CE 位 = 0 时设置 TM0CKS2 到 TM0CKS0 位。
当将 TM0CE 的值从 0 改变为 1 时, 不可以同时设置 TM0CKS2 到 TM0CKS0 的值。
 2. 请务必将第 3 位到第 6 位清零。

备注

f_{xx}: 主时钟频率
f_R: 内置振荡时钟频率
f_{XT}: 子时钟频率

9.4 操作

注意事项 不可以将 **TM0CMP0** 寄存器设置为 **FFFFH**。

9.4.1 间隔定时器模式

在间隔定时器模式中，当 **TM0CTL0.TM0CE** 位被置 1 后，中断请求信号(**INTTM0EQ0**)就会按照指定的间隔时间周期性产生。

图 9-2. 间隔定时器的结构图

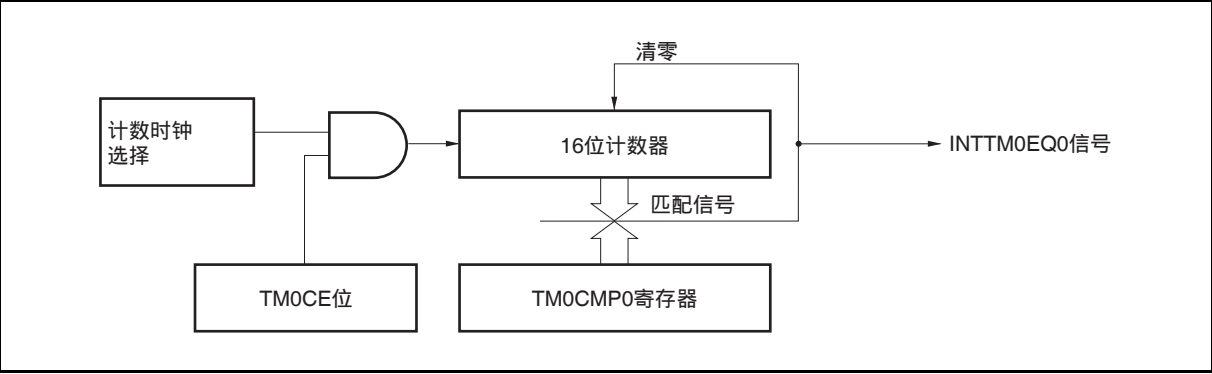
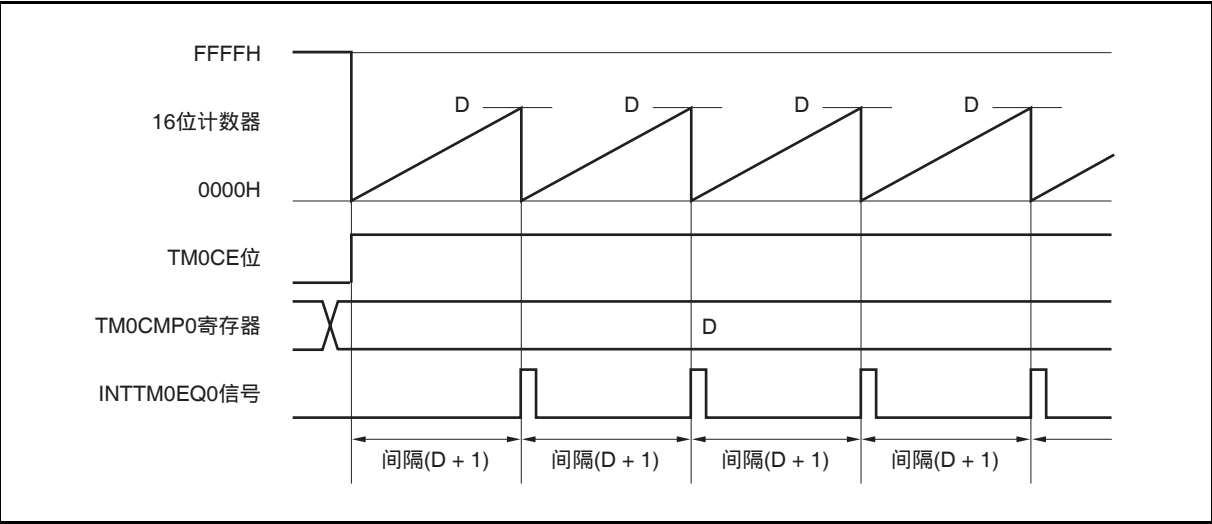


图 9-3. 间隔定时器模式的基本操作时序



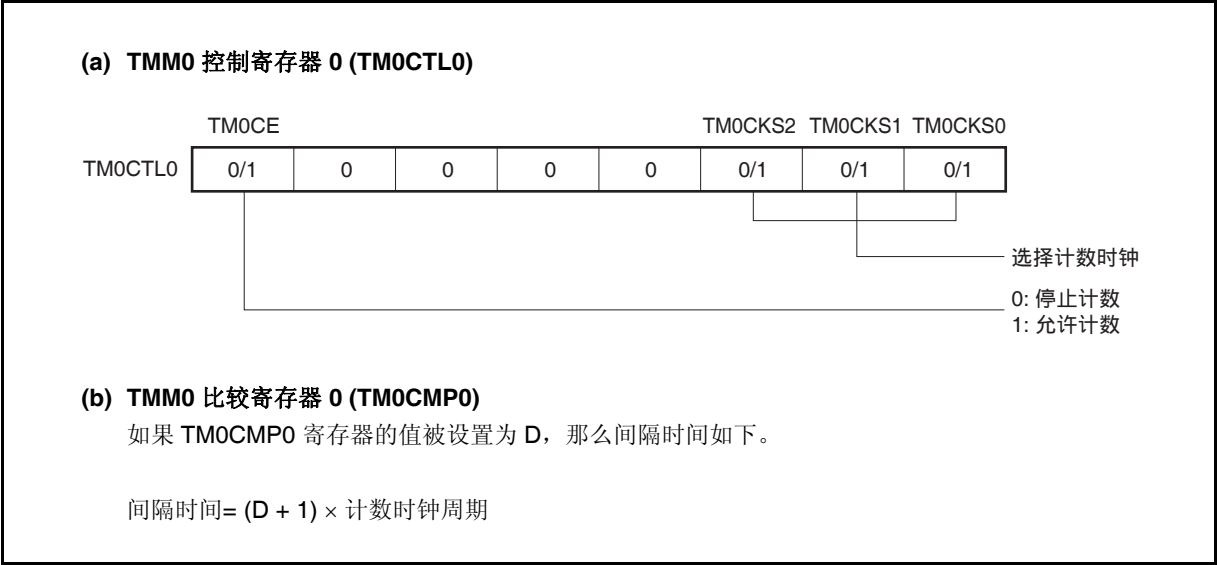
当 **TM0CE** 位被设置为 1，16 位计数器会与计数时钟同步将计数值从 **FFFFH** 清零为 **0000H**，并开始计数。

当 16 位计数器的计数值与 **TM0CMP0** 寄存器中的值相匹配时，16 位计数器被清零为 **0000H** 并同时产生一个比较匹配中断请求信号(**INTTM0EQ0**)。

间隔时间可由下列算式求得。

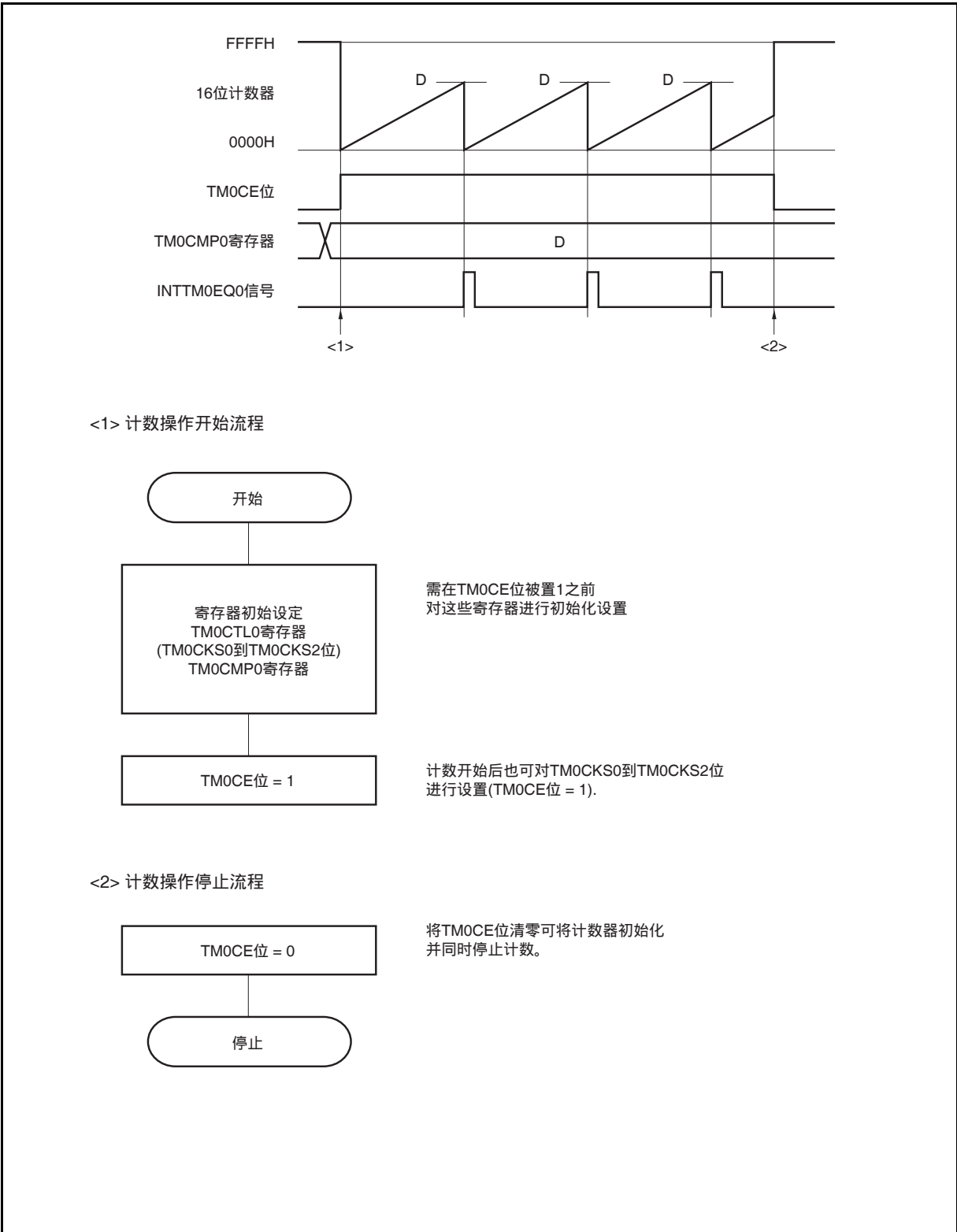
间隔时间 = (TM0CMP0 寄存器的设置值 + 1) × 计数时钟周期

图 9-4. 间隔定时器模式寄存器设置



(1) 间隔定时器模式的操作流程

图 9-5. 间隔定时器模式下的软件处理流程

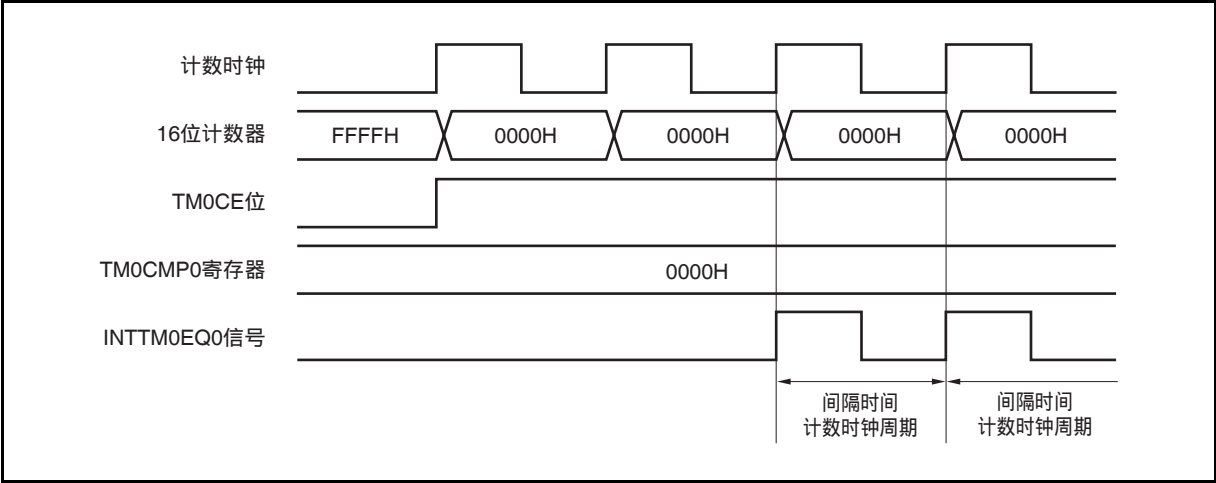


(2) 间隔定时器模式操作时序

注意 不可以将 **TM0CMP0** 寄存器设置为 **FFFFH**。

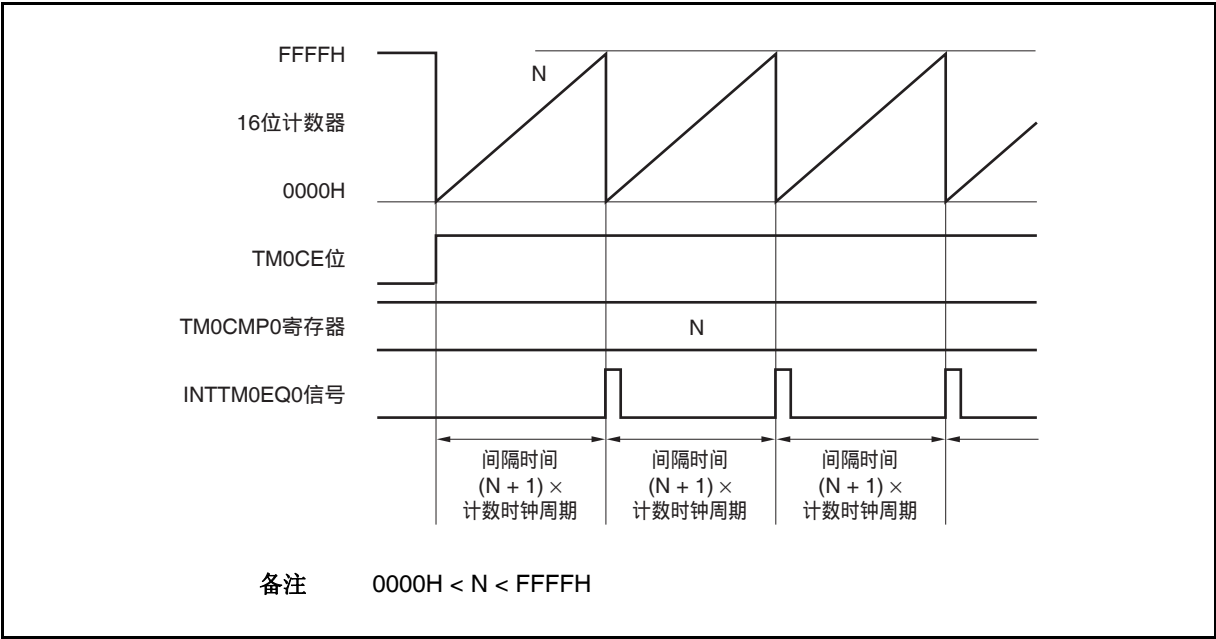
(a) **TM0CMP0** 寄存器被设置为 **0000H** 时的操作

如果 **TM0CMP0** 寄存器被设置为 **0000H**，那么每个计数时钟到来时刻都会产生 **INTTM0EQ0** 信号。
16 位计数器的值始终保持为 **0000H**。



(b) **TM0CMP0** 寄存器被设置为 **N** 时的操作

如果 **TM0CMP0** 寄存器被设置为 **N**，那么 16 位计数器会累加计数到 **N**，然后与下一个计数时钟同步被清零并产生 **INTTM0EQ0** 信号。



9.4.2 注意事项

- (1) 根据选择的计数时钟的不同，从 **TM0CTL0.TM0CE** 位被置 1 到 16 位计数器开始计数的时间间隔如下所示。

所选计数时钟	开始计数前的最大等待时间
f_{xx}	$2/f_{xx}$
$f_{xx}/2$	$6/f_{xx}$
$f_{xx}/4$	$24/f_{xx}$
$f_{xx}/64$	$128/f_{xx}$
$f_{xx}/512$	$1024/f_{xx}$
INTWT	INTWT 信号的第二个上升沿
$f_R/8$	$16/f_R$
f_{XT}	$2/f_{XT}$

- (2) TMM0 工作时，禁止改写 **TM0CMP0** 和 **TM0CTL0** 寄存器的值。

如果在 **TM0CE** 位 = 1 时对这两个寄存器进行改写，那么操作结果将不能被保证。

一旦对这两个寄存器的改写失败，请先将 **TM0CTL0.TM0CE** 位清零，之后重置这些寄存器。

第十章 钟表定时器功能

10.1 功能

钟表定时器具有以下功能。

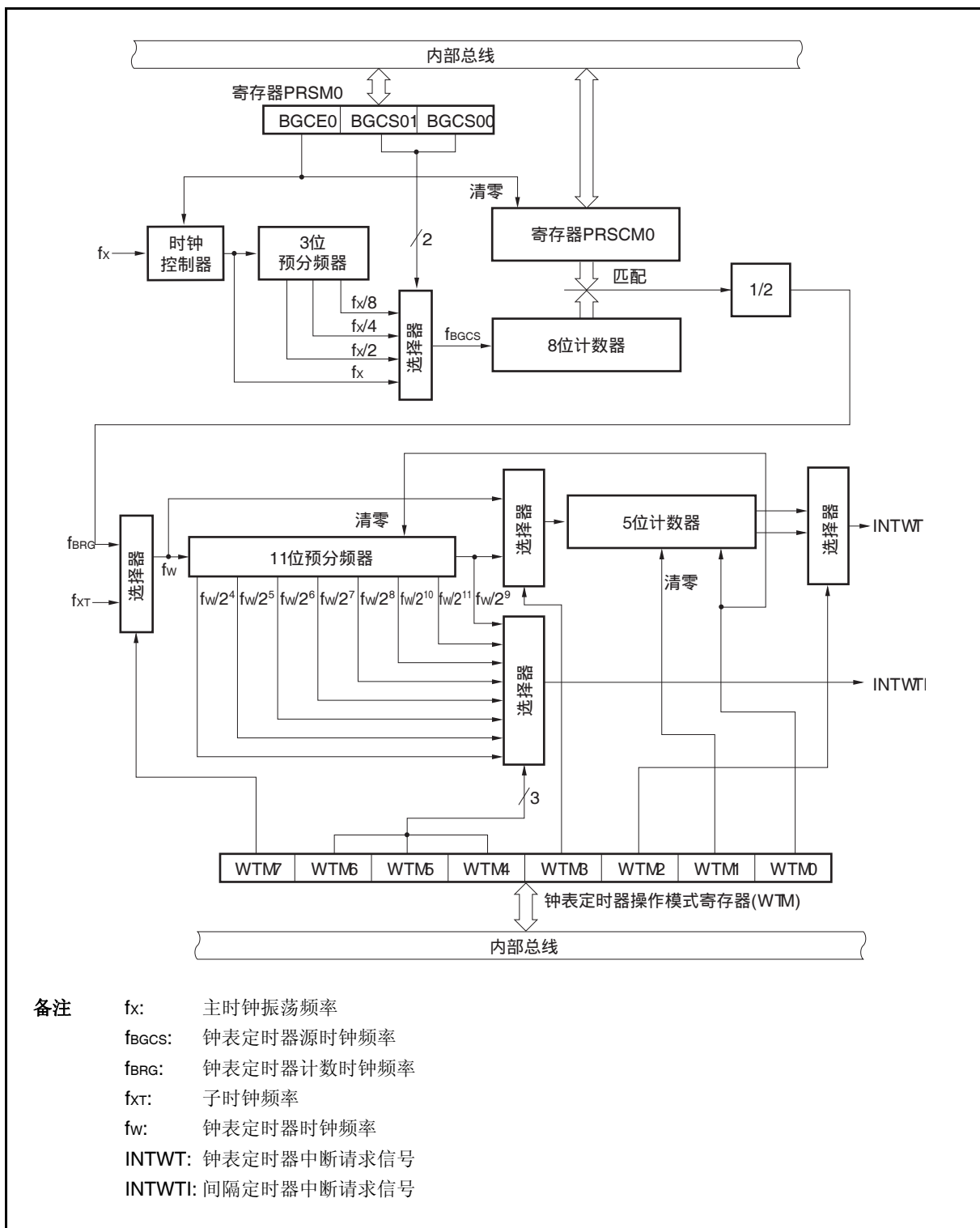
- 钟表定时器: 当使用主时钟或子时钟时, 以 0.5 或 0.25 秒的间隔产生中断请求信号(INTWT)。
- 间隔定时器: 以设定的间隔产生中断请求信号(INTWTI)。

钟表定时器和间隔定时器可以同时使用。

10.2 配置

钟表定时器的框图如下。

图 10-1. 钟表定时器的框图



(1) 时钟控制

当钟表定时器工作于主时钟时，该模块用于控制操作时钟(f_x)的供给和停止。

(2) 3 位预分频器

f_x 被预分频为 $f_x/2$, $f_x/4$, 或 $f_x/8$ 。

(3) 8 位计数器

该 8 位计数器以源时钟(f_{BGS})为计数时钟进行计数。

(4) 11 位预分频器

f_w 被预分频为 $f_w/2^4 \sim f_w/2^{11}$ 。

(5) 5 位计数器

该计数器以 f_w 或 $f_w/2^9$ 为计数时钟进行计数，以间隔 $2^4/f_w$, $2^5/f_w$, $2^{12}/f_w$ 或 $2^{14}/f_w$ 产生一个钟表定时器中断请求信号。

(6) 选择器

钟表定时器有以下 5 种选择器。

- 选择 f_x , $f_x/2$, $f_x/4$, 和 $f_x/8$ 之一为钟表定时器源时钟的选择器
- 选择主时钟(f_x)或子时钟(f_{xT})为钟表定时器时钟的选择器
- 选择 f_w 或 $f_w/2^9$ 为 5 位计数器的计数时钟频率的选择器
- 选择 $2^4/f_w$, $2^{13}/f_w$, $2^5/f_w$, 或 $2^{14}/f_w$ 为信号 $INTWT$ 产生时间间隔的选择器
- 选择 $2^4/f_w$ 或 $2^{11}/f_w$ 为间隔定时器中断请求信号($INTWTI$) 产生时间间隔的选择器

(7) 寄存器 PRSCM

该寄存器是 8 位比较寄存器，用于设置间隔时间。

(8) 寄存器 PRSM

该寄存器控制提供给钟表定时器的时钟。

(9) 寄存器 WTM

该寄存器是 8 位比较寄存器，用于控制钟表定时器/间隔定时器的操作，并设置中断请求信号产生间隔。

10.3 控制寄存器

钟表定时器使用下述寄存器。

- 预分频模式寄存器 0 (PRSM0)
- 预分频比较寄存器 0 (PRSCM0)
- 钟表定时器操作模式寄存器(WTM)

(1) 预分频模式寄存器 0 (PRSM0)

该寄存器控制钟表定时器计数时钟的产生。
可对该寄存器进行 1 位或 8 位的读写操作。
复位信号产生将该寄存器设置为 00H。

复位后: 00H R/W 地址: FFFFF8B0H

	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	钟表定时器时钟源选择 (f _{BGCS})		
			5 MHz	4 MHz
0	0	f _x	200 ns	250 ns
0	1	f _x /2	400 ns	500 ns
1	0	f _x /4	800 ns	1 μs
1	1	f _x /8	1.6 μs	2 μs

- 注意事项
1. 钟表定时器操作期间不要改变的 BGCS00 和 BGCS01 位取值。
 2. 在设置 BGCE0 为 1 前，先设置寄存器 PRSM0。
 3. 根据主时钟频率，设置寄存器 PRSM0 和 PRSCM0 以获得 32.768 kHz 的 f_{BRG} 频率。

(2) 预分频比较寄存器 0 (PRSCM0)

该寄存器是 8 位比较寄存器。
可对该寄存器进行 8 位的读写操作。
复位信号产生将该寄存器设置为 00H。

复位后: 00H R/W 地址: FFFFF8B1H

PRSCM0

7	6	5	4	3	2	1	0
PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00

注意事项

1. 钟表定时器操作期间不要对寄存器 **PRSCM0** 进行写入操作。
2. 在设置 **PRSM0.BGCE0** 为 1 前, 设置寄存器 **PRCM0**。
3. 根据主时钟频率, 设置寄存器 **PRSM0** 和 **PRSCM0** 以获得 32.768 kHz 的 **fBRG** 频率。

fBRG 的计算如下。

$f_{BRG} = f_{BGCS}/2N$

备注 fBGCS: 通过寄存器 **PRSM0** 设定的钟表定时器源时钟频率
N: 设置寄存器 **PRSM0** 的值为 1 到 256
 但是, 当 **PRSM0** 的值设为 00H 时, N = 256。

(3) 钟表定时器操作模式寄存器(WTM)

该寄存器用于允许/禁止计数时钟并设置钟表定时器和间隔定时器的预分频，控制 5 位计数器的操作，设置钟表标志的确立时间。

在设置寄存器 WTM 前，设置寄存器 PRSM0。

可对该寄存器进行 1 位或 8 位的读写操作。

复位信号产生将该寄存器设置为 00H。

(1/2)

复位后: 00H R/W 地址: FFFFF680H

	7	6	5	4	3	2	<1>	<0>
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	WTM6	WTM5	WTM4	预分频内部时钟选择
0	0	0	0	$2^4/f_w$ (488 μ s: $wf = f_{\text{KT}}$)
0	0	0	1	$2^5/f_w$ (977 μ s: $wf = f_{\text{KT}}$)
0	0	1	0	$2^6/f_w$ (1.95 ms: $wf = f_{\text{KT}}$)
0	0	1	1	$2^7/f_w$ (3.91 ms: $wf = f_{\text{KT}}$)
0	1	0	0	$2^8/f_w$ (7.81 ms: $wf = f_{\text{KT}}$)
0	1	0	1	$2^9/f_w$ (15.6 ms: $wf = f_{\text{KT}}$)
0	1	1	0	$2^{10}/f_w$ (31.3 ms: $wf = f_{\text{KT}}$)
0	1	1	1	$2^{11}/f_w$ (62.5 ms: $wf = f_{\text{KT}}$)
1	0	0	0	$2^4/f_w$ (488 μ s: $wf = f_{\text{BRQ}}$)
1	0	0	1	$2^5/f_w$ (977 μ s: $wf = f_{\text{BRQ}}$)
1	0	1	0	$2^6/f_w$ (1.95 ms: $wf = f_{\text{BRQ}}$)
1	0	1	1	$2^7/f_w$ (3.90 ms: $wf = f_{\text{BRQ}}$)
1	1	0	0	$2^8/f_w$ (7.81 ms: $wf = f_{\text{BRQ}}$)
1	1	0	1	$2^9/f_w$ (15.6 ms: $wf = f_{\text{BRQ}}$)
1	1	1	0	$2^{10}/f_w$ (31.2 ms: $wf = f_{\text{BRQ}}$)
1	1	1	1	$2^{11}/f_w$ (62.5 ms: $wf = f_{\text{BRQ}}$)

WTM7	WTM6	WTM5	钟表标志设定时间的选择
0	0	0	$2^{14}/f_w$ (0.5 s: $wf = f_{XT}$)
0	0	1	$2^{13}/f_w$ (0.25 s: $wf = f_{XT}$)
0	1	0	$2^5/f_w$ (977 μ s: $wf = f_{XT}$)
0	1	1	$2^4/f_w$ (488 μ s: $wf = f_{XT}$)
1	0	0	$2^{14}/f_w$ (0.5 s: $wf = f_{BRQ}$)
1	0	1	$2^{13}/f_w$ (0.25 s: $wf = f_{BRQ}$)
1	1	0	$2^5/f_w$ (977 μ s: $wf = f_{BRQ}$)
1	1	1	$2^4/f_w$ (488 μ s: $wf = f_{BRQ}$)

WTM4	5位计数器操作的控制
0	操作停止后清零
1	开始

WTM0	钟表定时器操作使能
0	停止操作(预分频和5位计数器均清零)
1	使能操作

注意事项 当 WTM0 和 WTM1 均为 0 时才可对 WTM2 ~ WTM7 位进行改写操作。

- 备注
- 1. f_w : 钟表定时器时钟频率
 - 2. 括号中的值应用于 $f_w = 32.768$ kHz 时

10.4 操作

10.4.1 钟表定时器的操作

钟表定时器以固定时间间隔产生中断请求信号(INTWT)。使用子时钟(32.768 kHz)或主时钟的 0.25 或 0.5 秒的时间间隔操纵钟表定时器。

当 WTM.WTM1 和 WTM.WTM0 位分别置 1，开始计数操作。当 WTM0 位被清零，11 位预分频器和 5 位计数器被清零，计数操作停止。

当定时器被同时作为钟表定时器和间隔定时器使用时，要通过先清零 WTM1 位再清零 5 位计数器的方法来调整时间。此时，钟表定时器可能产生一个最大 15.6 ms 的误差，而间隔定时器将不受影响。

如果主时钟作为钟表定时器的计数时钟，使用 PRSM0.BGCS01 和 BGCS00 设置计数时钟，使用寄存器 PRSCM0 设置 8 位比较值，以使钟表定时器的计数时钟频率(f_{BRG})为 32.768 kHz。

当 PRSM0.BGCE0 位置 1，f_{BRG} 被用于钟表定时器。

可以使用下面的表达式计算 f_{BRG}。

$$f_{BRG} = f_x / (2^{m+1} \times N)$$

要设置 f_{BRG} 为 32.768 kHz，进行下面计算，设置 BGCS01 和 BGCS00 位和寄存器 PRSCM0。

<1> 设置 N = f_x/65,536。设置 m = 0。

<2> 当 N 的个位的舍入结果为偶数时，在进行舍入之前，请设置 N = N/2，m = m + 1。

<3> 重复步骤 <2> 直到 N 成为奇数或 m = 3。

<4> 将舍入后 N 的值(整数部分)设置到 PRSCM0 寄存器，将 m 设置到 BGCS01 和 BGCS00 位中。

示例: 当 f_x = 4.00 MHz

<1> N = 4,000,000/65,536 = 61.03..., m = 0

<2>, <3> 由于 N (舍入后的整数部分) 为奇数, N = 61, m = 0。

<4> 设置寄存器 PRSCM0 的值: 3DH (61)，设置 BGCS01 和 BGCS00 位: 00

此时，f_{BRG} 的实际频率如下。

$$\begin{aligned} f_{BRG} &= f_x / (2^{m+1} \times N) = 4,000,000 / (2 \times 61) \\ &= 32.787 \text{ kHz} \end{aligned}$$

备注 m: 分频值 (设置 BGCS01 和 BGCS00 位的值) = 0 to 3

N: 设置寄存器 PRSCM0 的值= 1 to 256

但是，当 PRSCM0 的值=00H 时，N = 256。

f_x: 主时钟振荡频率

10.4.2 间隔定时器的操作

钟表定时器可以用作间隔定时器，并按照预设的数值周期性产生中断请求信号(INTWTI)。

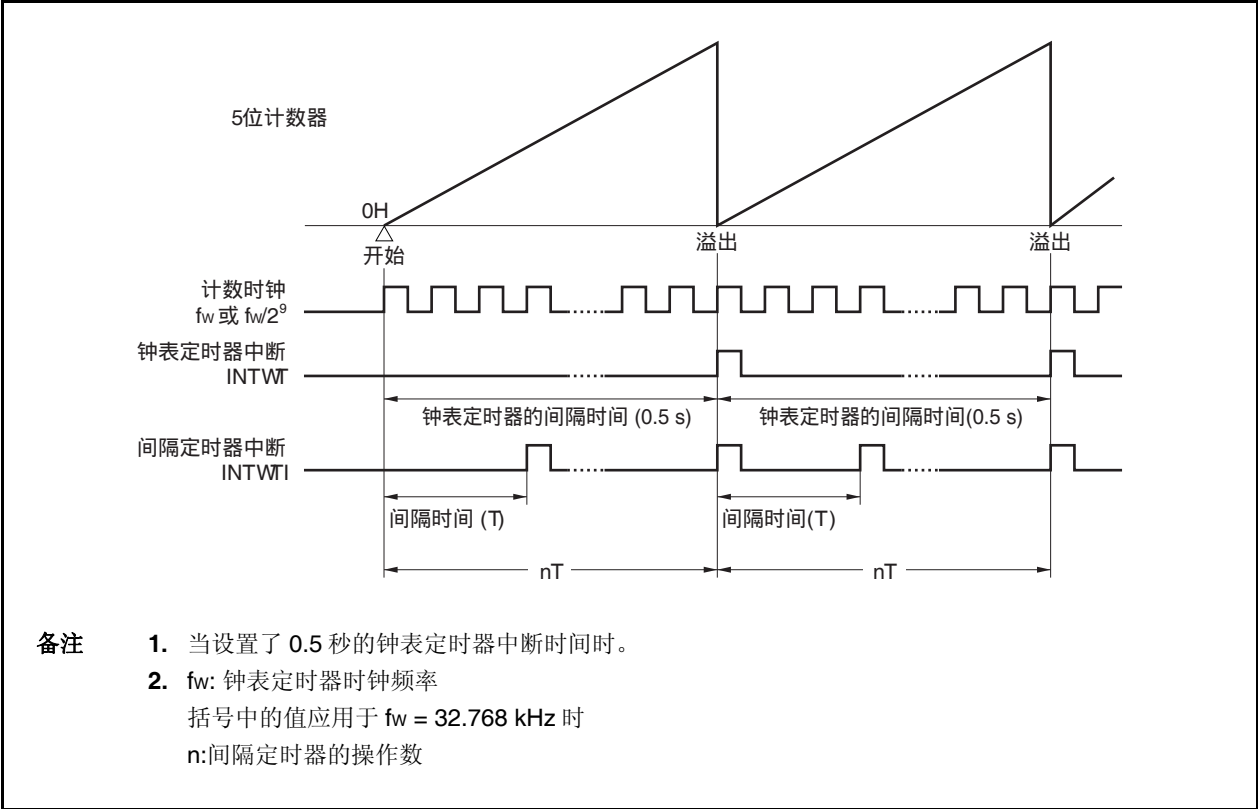
可由寄存器 WTM 的 WTM4 ~ WTM7 位选择间隔时间。

表 10-1. 间隔定时器的间隔时间

WTM7	WTM6	WTM5	WTM4	间隔时间	
0	0	0	0	$2^4 \times 1/f_w$	488 μ s (工作于 $f_w = f_{XT} = 32.768$ kHz)
0	0	0	1	$2^5 \times 1/f_w$	977 μ s (工作于 $f_w = f_{XT} = 32.768$ kHz)
0	0	1	0	$2^6 \times 1/f_w$	1.95 ms (工作于 $f_w = f_{XT} = 32.768$ kHz)
0	0	1	1	$2^7 \times 1/f_w$	3.91 ms (工作于 $f_w = f_{XT} = 32.768$ kHz)
0	1	0	0	$2^8 \times 1/f_w$	7.81 ms (工作于 $f_w = f_{XT} = 32.768$ kHz)
0	1	0	1	$2^9 \times 1/f_w$	15.6 ms (工作于 $f_w = f_{XT} = 32.768$ kHz)
0	1	1	0	$2^{10} \times 1/f_w$	31.3 ms (工作于 $f_w = f_{XT} = 32.768$ kHz)
0	1	1	1	$2^{11} \times 1/f_w$	62.5 ms (工作于 $f_w = f_{XT} = 32.768$ kHz)
1	0	0	0	$2^4 \times 1/f_w$	488 μ s (工作于 $f_w = f_{BRG} = 32.768$ kHz)
1	0	0	1	$2^5 \times 1/f_w$	977 μ s (工作于 $f_w = f_{BRG} = 32.768$ kHz)
1	0	1	0	$2^6 \times 1/f_w$	1.95 ms (工作于 $f_w = f_{BRG} = 32.768$ kHz)
1	0	1	1	$2^7 \times 1/f_w$	3.91 ms (工作于 $f_w = f_{BRG} = 32.768$ kHz)
1	1	0	0	$2^8 \times 1/f_w$	7.81 ms (工作于 $f_w = f_{BRG} = 32.768$ kHz)
1	1	0	1	$2^9 \times 1/f_w$	15.6 ms (工作于 $f_w = f_{BRG} = 32.768$ kHz)
1	1	1	0	$2^{10} \times 1/f_w$	31.3 ms (工作于 $f_w = f_{BRG} = 32.768$ kHz)
1	1	1	1	$2^{11} \times 1/f_w$	62.5 ms (工作于 $f_w = f_{BRG} = 32.768$ kHz)

备注 f_w : 钟表定时器时钟频率

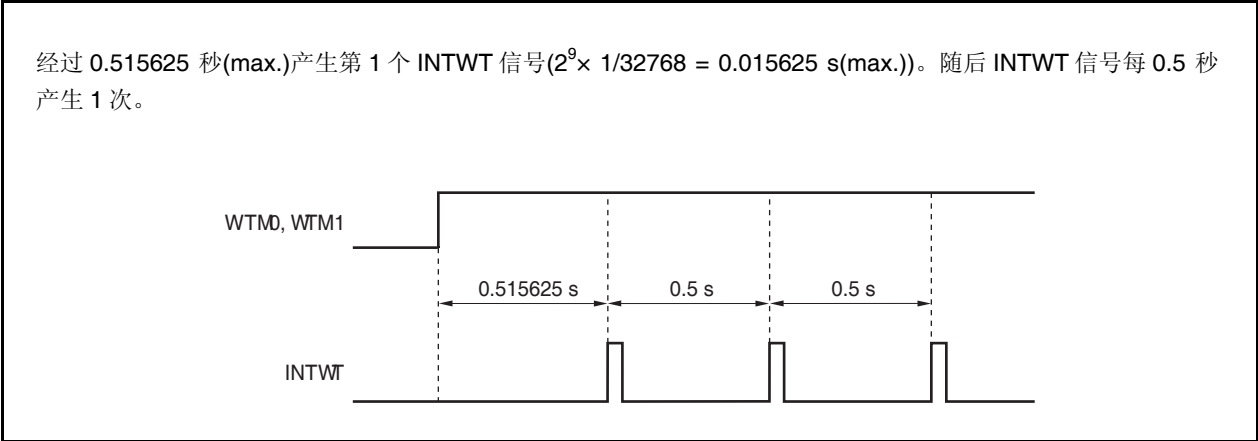
图 10-2. 钟表定时器/间隔定时器的操作时序



10.4.3 注意事项

在允许操作 (WTM.WTM1 和 WTM.WTM0 = 1)后，第一次钟表定时器中断请求信号(INTWT)产生前需要一段准备时间。

图 10-3. 钟表定时器产生中断请求信号(INTWT)的示例(当中断周期= 0.5 秒)



第十一章 看门狗定时器 2 的功能

11.1 功能

看门狗定时器 2 的功能如下。

- 看门狗定时器默认开始工作^{注 1}
 - 复位模式: 看门狗定时器 2 溢出时复位(产生信号 WDT2RES)
 - 不可屏蔽中断请求模式: 看门狗定时器 2 溢出时 NMI 操作(产生信号 INTWDT2)^{注 2}
- 源时钟可选主时钟, 内部振荡时钟和子时钟

注 1. 复位释放后, 看门狗定时器 2 会自动启动。

当不使用看门狗定时器 2, 在复位前通过该功能使其停止工作, 或清除看门狗定时器 2 并在下一次间隔开始前使其停止工作。

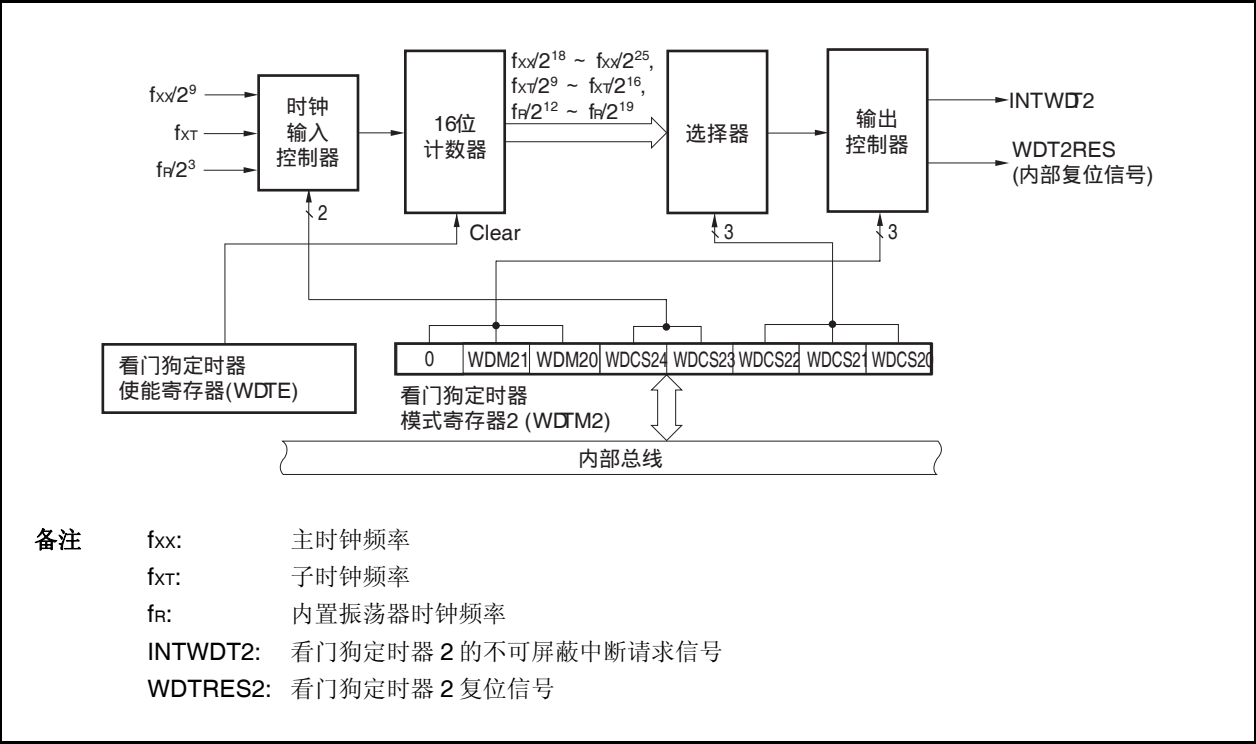
为了确认操作的正确性, 即使在不需要改变默认设置(复位模式, 间隔时间: $f_R/2^{19}$)的情况下, 也要对寄存器 WDTM2 执行一次写操作。

2. 由不可屏蔽中断请求信号(INTWDT2)引发的不可屏蔽中断服务, 可参见 19.2.2 (2) 信号 INTWDT2。

11.2 配置

看门狗定时器 2 的框图如下。

图 11-1. 看门狗定时器 2 的框图



看门狗定时器 2 由以下寄存器控制。

表 11-1. 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)

11.3 寄存器

(1) 看门狗定时器模式寄存器 2 (WDTM2)

该寄存器设置溢出时间和看门狗定时器 2 的操作时钟。
可由 8 位存储器操作指令设置该寄存器。该寄存器可多次读取，但复位释放后只能写入一次。
复位信号产生将该寄存器设置为 67H。

- ★
- 注意事项 下述状态下，禁止访问寄存器 WDTM2。详细情况，参见 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。
- 当 CPU 使用子时钟操作，主时钟振荡停止。
 - 当 CPU 使用内置振荡时钟进行操作

复位后: 67H R/W 地址: FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	WDCS24	WDCS23	WDCS22	WDCS21	WDCS20

WDM21	WDM20	看门狗定时器操作模式选择 2
0	0	停止操作
0	1	不可屏蔽中断请求模式 (产生 INTWDT2 信号)
1	—	复位模式 (产生 WDT2RES 信号)

- 注意事项
- WDCS20 ~ WDCS24 的详细情况，参见表 11-2 看门狗定时器 2 的时钟选择。
 - 尽管停止内置振荡器的操作可以停止看门狗定时器 2，但是，还要使寄存器 WDTM2 清零以确保停止定时器(以避免由于误操作选中主时钟或子时钟)。
 - 如果寄存器 WDTM2 在复位后被复写两次或更多次，那么溢出信号产生，计数器复位。
 - 如需有意产生一个溢出信号，请向寄存器 WDTE 写入“ACH”以外的值一次，或向寄存器 WDTM2 写入数据两次。
 - 要停止看门狗定时器 2 的操作，设置 RCM.RSTP 为 1(以停止内置晶振)并向寄存器 WDTM 写入 00H。如果 RCM.RSTP 不能被置 1，设置 WDCS23 为 1(选中 $2^n/f_{xx}$ ，时钟在 IDLE1, IDLW2, sub-IDLE, 和子时钟模式下可停止工作)。

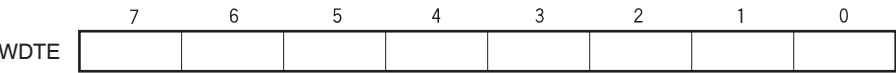
表 11-2. 看门狗定时器 2 的时钟选择

WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	所选择的时钟	100 kHz (MIN.)	200 kHz (TYP.)	400 kHz (MAX.)
0	0	0	0	0	$2^{12}/f_R$	41.0 ms	20.5 ms	10.2 ms
0	0	0	0	1	$2^{13}/f_R$	81.9 ms	41.0 ms	20.5 ms
0	0	0	1	0	$2^{14}/f_R$	163.8 ms	81.9 ms	41.0 ms
0	0	0	1	1	$2^{15}/f_R$	327.7 ms	163.8 ms	81.9 ms
0	0	1	0	0	$2^{16}/f_R$	655.4 ms	327.7 ms	163.8 ms
0	0	1	0	1	$2^{17}/f_R$	1,310.7 ms	655.4 ms	327.7 ms
0	0	1	1	0	$2^{18}/f_R$	2,621.4 ms	1,310.7 ms	655.4 ms
0	0	1	1	1	$2^{19}/f_R$	5,242.9 ms	2,621.47 ms	1,310.7 ms
						$f_{XX} = 20 \text{ MHz}$	$f_{XX} = 16 \text{ MHz}$	$f_{XX} = 10 \text{ MHz}$
0	1	0	0	0	$2^{18}/f_{XX}$	13.1 ms	16.4 ms	26.2 ms
0	1	0	0	1	$2^{19}/f_{XX}$	26.2 ms	32.8 ms	52.4 ms
0	1	0	1	0	$2^{20}/f_{XX}$	52.4 ms	65.5 ms	104.9 ms
0	1	0	1	1	$2^{21}/f_{XX}$	104.9 ms	131.1 ms	209.7 ms
0	1	1	0	0	$2^{22}/f_{XX}$	209.7 ms	262.1 ms	419.4 ms
0	1	1	0	1	$2^{23}/f_{XX}$	419.4 ms	524.3 ms	838.9 ms
0	1	1	1	0	$2^{24}/f_{XX}$	838.9 ms	1,048.6 ms	1,677.7 ms
0	1	1	1	1	$2^{25}/f_{XX}$	1,677.7 ms	2,097.2 ms	3,355.4 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”写入寄存器 WDTE，则看门狗定时器 2 的计数器被清零，计数重新开始。
可由 8 位存储器操作指令设置该寄存器。
复位信号产生将该寄存器设置为 9AH。

复位后: 9AH R/W 地址: FFFFF6D1H



- 注意事项
1. 如果“ACH”以外的数值被写入寄存器 WDTE，必定产生溢出信号。
 2. 当向寄存器 WDTE 执行 1 位存储器操作指令，必定产生溢出信号。
 3. 如需有意产生一个溢出信号，请向寄存器 WDTE 写入“ACH”以外的值一次，或向寄存器 WDTM2 写入数据两次。
然而，当看门狗定时器 2 设为停止工作状态时，即使向寄存器 WDTE 写入“ACH”以外的值一次，或向寄存器 WDTM2 写入数据两次也不会产生溢出信号。
 4. 读取寄存器 WDTE 的值为“9AH”(与写入值“ACH”不同)。

11.4 操作

复位释放后，看门狗定时器 2 自动开始运行。

复位后寄存器 WDTM2 (使用字节访问操作)只可被写入一次。要使用看门狗定时器 2，使用 8 位操作指令，向寄存器 WDTM2 写入操作模式和间隔时间。之后，将无法停止看门狗定时器 2 的操作。

寄存器 WDTM2 的 WDCS24 ~ WDCS20 位被用于选择看门狗定时器 2 循环检测时间间隔。

向寄存器 WDTE 写入 ACH，就会使看门狗定时器 2 的计数器清零，并开始重新计数操作。在计数器开始操作后，循环检测时间间隔内向寄存器 WDTE 写入 ACH。

如果在定时间隔记满时没有对寄存器 WDTE 进行 ACH 写操作，就会根据 WDM21 和 WDTM2.WDM20 的设置值，产生复位信号(WDT2RES)或不可屏蔽中断请求信号(INTWDT2)。

当 WDTM2.WDM21 位设置为 1(复位模式)，如果在复位或待机释放后晶振稳定过程中产生 WDT 溢出，将不产生内部复位且 CPU 时钟将转换成内部振荡时钟。

要不使用看门狗定时器 2，向寄存器 WDTM2 写入 00H。

不可屏蔽中断请求模式下的不可屏蔽中断请求服务可参见 19.2.2 (2)信号 INTWDT2。

第十二章 实时输出功能 (RTO)

12.1 功能

实时输出功能发送预先设置的数据到寄存器 **RTBL0** 和 **RTBH0**，然后在产生定时器中断时由硬件通过输出锁存发送这些数据到外部设备。发送这些数据到外部设备的引脚组成了实时输出功能(RTO)端口。

由于 **RTO** 可以稳定的输出信号，所以适用于控制步进电机。

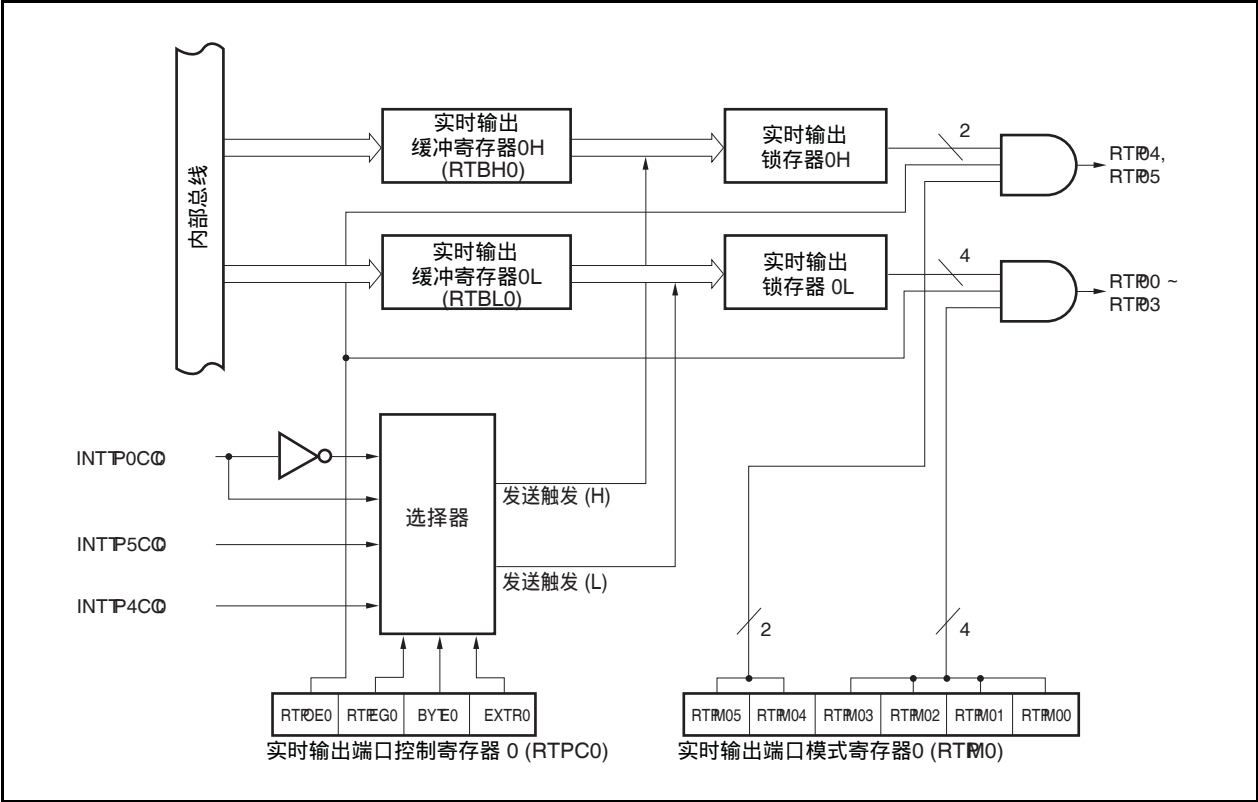
V850ES/JG2 提供一个 **6** 位的实时输出端口通道。

可由 **1** 位存储器操作指令设置实时输出端口工作在普通端口模式或实时输出端口模式。

12.2 配置

RTO 的框图如下。

图 12-1. RTO 的框图



RTO 由以下寄存器控制。

表 12-1. 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)

寄存器 RTBL0 和 RTBH0 是 4 位寄存器，用于保持预设输出数据。
该寄存器被映射到周边 I/O 寄存器区域的独立地址。
可由 1 位或 8 位存储器操作指令设置该寄存器。
复位信号产生将该寄存器设置为 00H。
如果指定了操作模式为 4 位 × 1 通道或 2 位 × 1 通道(RTPC0.BYTE0 = 0)，数据可被分别设置到寄存器 RTBL0 和 RTBH0。通过指定这两个寄存器中的任意一个的地址，这两个寄存器中的数据可被立刻读取。
如果指定了操作模式为 6 位 × 1 通道(BYTE0 = 1)，8 位长的数据可被设置到寄存器 RTBL0 和 RTBH0 中，写入这两个寄存器中的任意一个。另外，通过指定这两个寄存器中的任意一个的地址，这两个寄存器中的数据可被立刻读取。
表 12-2 展示了操作寄存器 RTBL0 和 RTBH0 的过程。

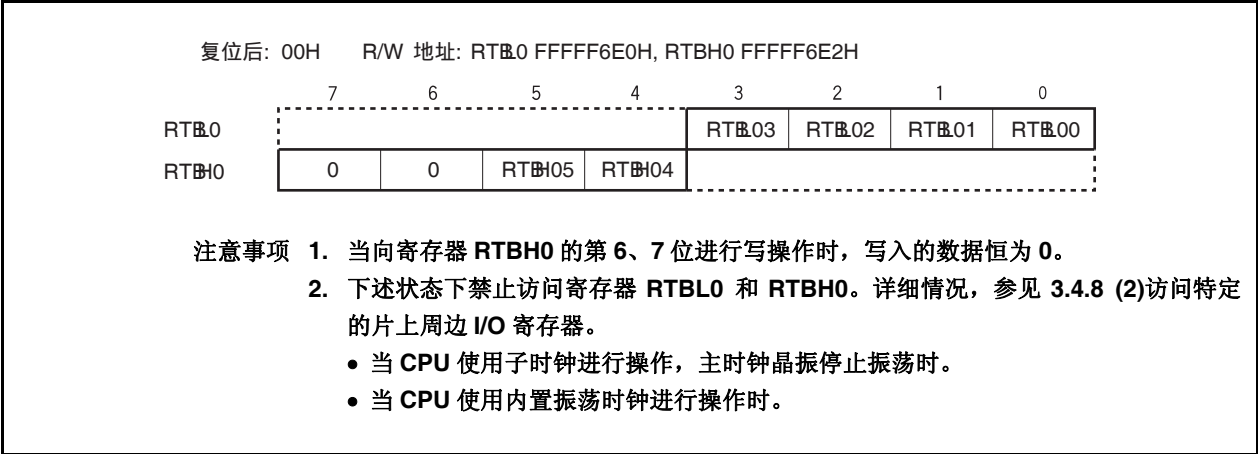


表 12-2. 寄存器 RTBL0 和 RTBH0 的操作

操作模式	被操作的寄存器	读		写 ^注	
		高 4 位	低 4 位	高 4 位	低 4 位
4 位 × 1 通道, 2 位 × 1 通道	RTBL0	RTBH0	RTBL0	Invalid	RTBL0
	RTBH0	RTBH0	RTBL0	RTBH0	Invalid
6 位 × 1 通道	RTBL0	RTBH0	RTBL0	RTBH0	RTBL0
	RTBH0	RTBH0	RTBL0	RTBH0	RTBL0

注 设置实时输出端口后，要在产生实时输出触发之前，将欲输出的数据设置到寄存器 RTBL0 和 RTBH0。

12.3 寄存器

RTO 由下面两个寄存器控制。

- 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)

该寄存器选择实时输出端口模式或普通端口模式，可由 1 位存储器操作指令完成操作。
可由 1 位或 8 位存储器操作指令设置该寄存器。
复位信号产生将该寄存器设置为 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	实时输出端口控制(m = 0 ~ 5)
0	禁止实时输出
1	使能实时输出

- 注意事项
1. 通过允许实时输出操作(RTPC0.RTPOE0 = 1)，RTP00 ~ RTP05 中设置为实时输出的位将进行实时输出操作，而那些设置为普通端口模式的位会输出 0。
 2. 禁止实时输出 (RTPC0.RTPOE0 = 0)时，无论 RTPM0 寄存器如何设置，实时输出引脚 RTP00 ~ RTP05 均输出 0。
 3. 为了将 RTP00 ~ RTP05 引脚作为实时输出引脚使用，需要通过 PMC 和 PFC 寄存器将这些引脚的模式设置为实时输出端口模式。

(2) 实时输出端口控制寄存器 0 (RTPC0)

该寄存器用于设置实时输出端口的操作模式和触发模式。
实时输出端口的操作模式和触发模式的关系如表 12-3 所示。
可由 1 位或 8 位存储器操作指令设置该寄存器。
复位信号产生将该寄存器设置为 00H。

复位后: 00H R/W 地址: FFFFF6E5H

RTPC0

<7>	6	5	4	3	2	1	0
RTPOE0	RTPEG0	BYTE0	EXTR0	0	0	0	0

RTPOE0	实时输出操作控制
0	禁止操作 ^{注1}
1	使能操作

RTPEG0	信号INTTP0CC0有效边沿
0	下降沿 ^{注2}
1	上升沿

BYTE0	实时输出通道设置选项
0	4位×2通道, 2位×2通道
1	6位×2通道

注

1.

禁止实时输出操作(RTPOE0 = 0)时, 实时输出信号的所有位(RTP00 ~ RTP05)输出“0”。

2.

INTTP0CC0 信号的有效宽度与由 TMP0 选择的 1 个计数时钟周期的宽度等宽。

注意事项

只有当 RTPOE0 = 0 时, 设置 RTPEG0, BYTE0 和 EXTR0 位。

表 12-3. 实时输出端口的操作模式和输出触发

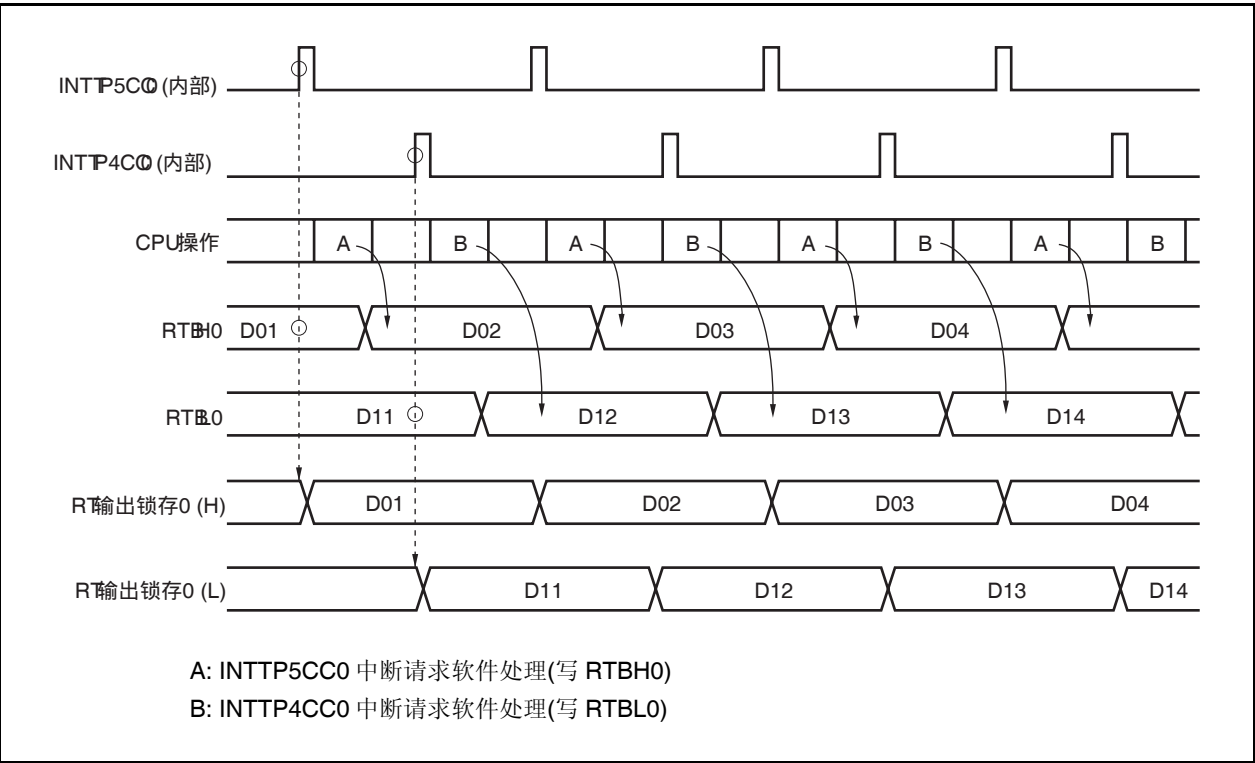
BYTE0	EXTR0	操作模式	RTBH0 (RTP04, RTP05)	RTBL0 (RTP00 to RTP03)
0	0	4 位 × 1 通道,	INTTP5CC0	INTTP4CC0
	1	2 位 × 1 通道	INTTP4CC0	INTTP0CC0
1	0	6 位 × 1 通道	INTTP4CC0	
	1		INTTP0CC0	

12.4 操作

如果设置 RTPC0.RTPOE0 为 1，实时输出操作被允许，寄存器 RTBH0 和 RTBL0 的数据将与产生所选择的发送触发(由 RTPC0.EXTR0 和 RTPC0.BYTE0 设置)同步被发送到实时输出锁存器。对于被发送的数据，只有那些由寄存器 RTPM0 设定的允许实时输出的位才从 RTP00 到 RTP05 位输出。寄存器 RTPM0 设定的禁止实时输出的位输出为 0。

如果通过将 RTPOE0 清零的方法禁止实时输出操作，那么无论寄存器 RTPM0 如何设置，信号 RTP00 ~ RTP05 的输出都为 0。

图 12-2. RTO0 操作时序的示例 (当 EXTR0 = 0, BYTE0 = 0)



备注 待机模式下的操作，参见第 21 章 待机功能

12.5 用法

- (1) 禁止实时输出。
RTPC0.RTPOE0 清零。
- (2) 初始化如下。
 - 设置复用功能引脚端口 5。
设置 PFC5.PFC5m 和 PFCE5.PFCE5m 为 1，然后设置 PMC5.PMC5m 为 1(m = 0 ~ 5)。
 - 可由 1 位存储器操作指令设置实时输出端口模式或普通端口模式。
设置寄存器 RTPM0。
 - 通道配置:选择触发和有效沿。
设置 RTPC0.EXTR0, RTPC0.BYTE0, 和 RTPC0.RTPEG0 位。
 - 设置寄存器 RTBH0 和 RTBL0^{注1}的初始值。
- (3) 允许实时输出。
设置 RTPOE0 = 1。
- (4) 在产生所选择的发送触发^{注2}时，设置寄存器 RTBH0 和 RTBL0 的下一输出值。
- (5) 在产生所选择的发送触发相应的中断服务中，设置寄存器 RTBH0 和 RTBL0 的下一输出值。

注1. 如果当 RTPOE0 = 0 时写寄存器 RTBH0 和 RTBL0，数据被分别发送到实时输出锁存 0H 和 0L。

2. 即使当 RTPOE0 = 1 时写寄存器 RTBH0 和 RTBL0，数据也不被发送到实时输出锁存 0H 和 0L。

12.6 注意事项

- (1) 避免下述软件冲突。
 - 实时输出允许/禁止转换(RTPOE0)和所选择的实时输出触发之间的冲突。
 - 实时输出允许状态下写寄存器 RTBH0 和 RTBL0，与所选择的实时输出触发之间的冲突。
- (2) 在操作初始化前，停止实时输出(RTPOE0 = 0)。
- (3) 一旦实时输出被禁止(RTPOE0 = 0)，确保在再次允许实时输出(RTPOE0 = 0 → 1)前初始化寄存器 RTBH0 和 RTBL0。

第十三章 A/D 转换器

13.1 概览

A/D 转换器用于将模拟输入信号转换为数字信号，具有 10 位分辨率，最多可由 12 个通道(AN10 ~ AN11)组成。

A/D 转换器有以下特点。

- 10 位分辨率
- 12 通道
- 逐次逼近法
- 操作电压: $AV_{REF0} = 3.0$ 到 3.6 V
- 模拟输入电压: 0 V to AV_{REF0}
- 以下为所能提供的操作模式。
 - 连续选择模式
 - 连续扫描模式
 - 单脉冲选择模式
 - 单脉冲扫描模式
- 以下功能为所能提供的触发模式。
 - 软件触发模式
 - 外部触发模式(外部, 1)
 - 定时器触发模式
- 电压不足监测功能(转换结果比较功能)

13.2 功能

(1) 10 位分辨率 A/D 转换

从 AN10 到 AN11 选择一个模拟通道，A/D 转换操作以 10 位分辨率重复进行。每当 A/D 转换结束，都产生一个中断请求信号(INTAD)。

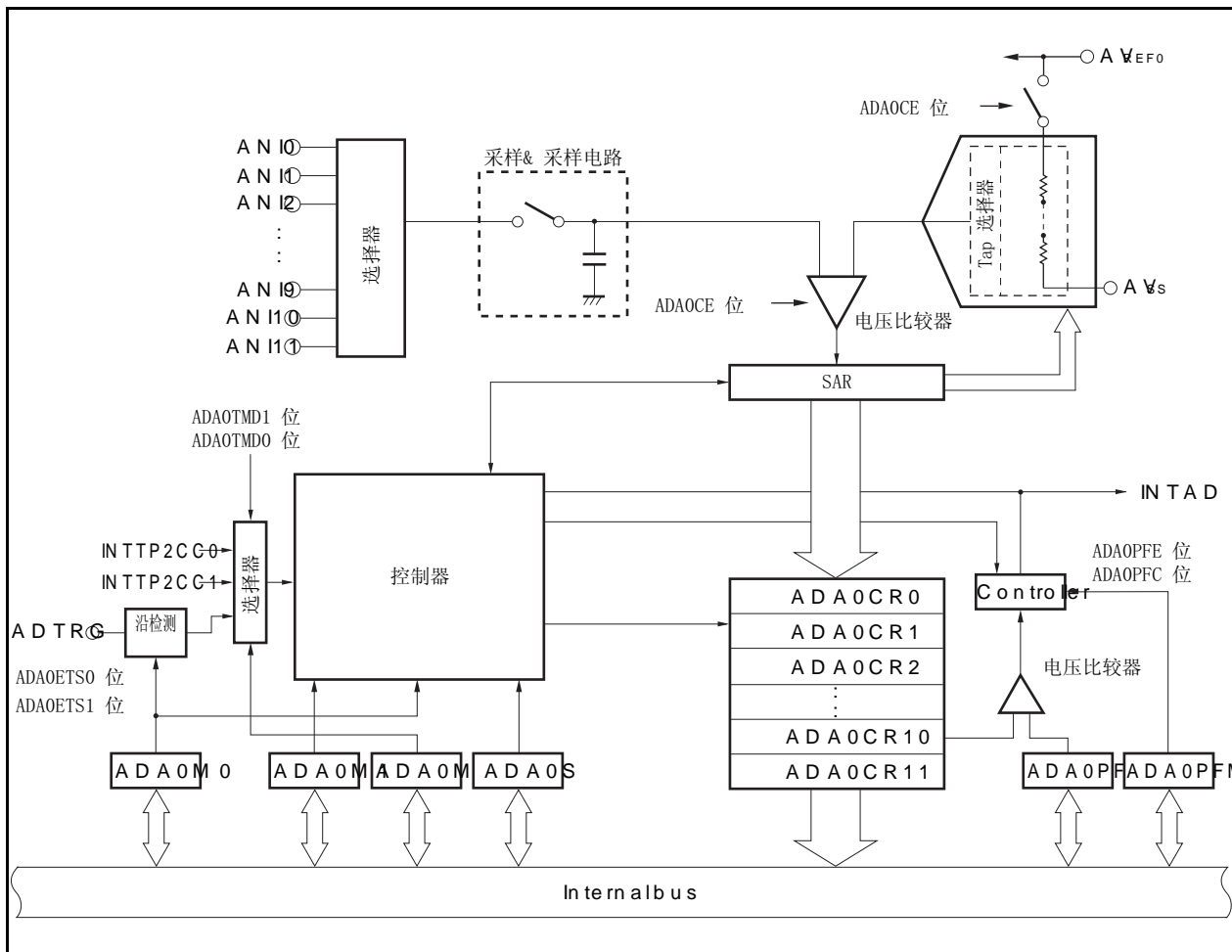
(2) 电压不足检测功能

该功能用于检测电池电压的下降。A/D 转换的结果(寄存器 ADA0CRnH 的值)与寄存器 ADA0PFT 的值进行比较，只有当指定的比较条件($n = 0 \sim 11$)满足时，才产生中断信号 INTAD。

13.3 配置

如下所示为 A/D 转换器的框图。

图 13-1. A/D 转换器的框图



A/D 转换器包括以下硬件。

表 13-1. A/D 转换器的配置

项目	配置
模拟输入	12 通道 (ANIO ~ ANI11 引脚)
寄存器	逐次逼近寄存器 (SAR) A/D 转换结果寄存器 0 ~ 11 (ADA0CR0 ~ ADA0CR11) A/D 转换结果寄存器 0H ~ 11H (ADCR0H ~ ADCR11H): 只能读取高 8 位
控制寄存器	A/D 转换器模式寄存器 0 ~ 2 (ADA0M0 ~ ADA0M2) A/D 转换器通道选择寄存器 0 (ADA0S) 电压不足比较模式寄存器 (ADA0PFM) 电压不足比较阈值寄存器 (ADA0PFT)

(1) 逐次逼近寄存器 (SAR)

通过一串串联电阻，寄存器 SAR 将模拟输入电压值与比较电压值(voltage tap)进行比较，保持比较结果。比较结果从最高有效位(MSB)开始。

当比较结果被致以最低有效位(LSB) (也就是说，当 A/D 转换完全)，寄存器 SAR 的结果被传送到寄存器 ADA0CRn。

备注 $n = 0 \sim 11$

(2) A/D 转换结果寄存器 n (ADA0CRn)，A/D 转换结果寄存器 nH (ADA0CRnH)

寄存器 ADA0CRn 是 16 位寄存器，用于存储 A/D 转换结果。ADA0CRn 由 12 个寄存器组成。根据输入模拟信号，A/D 转换结果存储于 ADA0CRn 的高 10 位。(低 6 位恒为 0。)

(3) A/D 转换器模式寄存器 0 (ADA0M0)

该寄存器设置操作模式并控制 A/D 转换器的转换操作。

(4) A/D 转换器模式寄存器 1 (ADA0M1)

该寄存器设置输入模拟信号被转换的时间。

(5) A/D 转换器模式寄存器 2 (ADA0M2)

该寄存器设置硬件触发模式。

(6) A/D 转换器通道选择寄存器 (ADA0S)

该寄存器设置输入端，该输入端输入被转换的模拟电压。

(7) 电压不足比较模式寄存器 (ADA0PFM)

该寄存器设置电压不足监测模式。

(8) 电压不足比较阈值寄存器 (ADA0PFT)

寄存器 ADA0PFT 设置一个与 A/D 转换结果寄存器 nH (ADA0CRnH)比较的阈值。寄存器 ADA0PFT 中所设置的 8 位数据与 A/D 转换结果寄存器(ADA0CRnH)比较。

(9) 控制器

当 A/D 转换完成或使用电压不足监测功能时，控制器将 A/D 转换结果(ADA0CRnH 的值)与 ADA0PF 的值进行比较，只有当指定的比较条件满足时，才产生中断信号 INTAD。

(10) 采样与保存电路

采用与保存电路采集每个模拟输入信号，发送采样数据到电压比较器。该电路在 A/D 转换期间保持被采样的模拟输入信号的电压。

(11) 电压比较器

电压比较器比较被采样的电压值并使用一串串联电阻保持采样电压值。

(12) 串联电阻串

串联电阻串连接 AV_{REF0} 和 AV_{SS} 并产生一个与模拟输入信号相比较的电压。

(13) $ANI0 \sim ANI11$ 引脚

$ANI0 \sim ANI11$ 是 12 个 A/D 转换器通道的引脚，用于输入待转换成数字信号的模拟信号。没有被寄存器 $ADA0S$ 选定作为模拟输入的引脚可以用于普通输入端口。

注意事项 确保输入 $ANI0 \sim ANI11$ 的电压不超过额定值。特别是，如果大于或等于 AV_{REF0} 的电压输入某个通道，这个通道的转换结果将不确定，其它通道的转换结果也将受到影响。

(14) AV_{REF0} 引脚

该引脚用于输入 A/D 转换器的参考电压。即使没有使用 A/D 转换器，也应保持该引脚电压与 V_{DD} 引脚的一致性。输入到引脚 $ANI0 \sim ANI11$ 的信号基于引脚 AV_{REF0} 和 AV_{SS} 之间的电压被转换成数字信号。

(15) AV_{SS} 引脚

这是 A/D 转换器的信号地引脚。即便没有使用 A/D 转换器，该引脚的电压也应与 V_{SS} 的电压保持一致。

13.4 寄存器

A/D 转换器使用以下寄存器进行控制。

- A/D 转换器模式寄存器 0, 1, 2 (ADA0M0, ADA0M1, ADA0M2)
- A/D 转换器通道选择寄存器 0 (ADA0S)
- 电压不足比较模式寄存器 (ADA0PFM)

还使用下面的寄存器。

- A/D 转换结果寄存器 n (ADA0CRn)
- A/D 转换结果寄存器 nH (ADA0CRnH)
- 电压不足比较阈值寄存器 (ADA0PFT)

(1) A/D 转换器模式寄存器 0 (ADA0M0)

该寄存器是 8 位寄存器，用于选择操作模式并控制 A/D 转换器的转换操作。可由 1 位或 8 位存储器操作指令读写该寄存器。但是，ADA0EF 位是只读的。

复位信号产生将该寄存器清零(00H)。

- ★ **注意事项** 下述情况下禁止访问寄存器 **ADA0M0**。详细情况，参见 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。
- 当 CPU 使用子时钟，且主时钟晶振停止振荡。
 - 当 CPU 使用内部晶振时钟。

(1/2)

复位后: 00H R/W 地址: FFFF00H

	<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	允许 A/D 控制

ADA0MD1	ADA0MD0	A/D 转换操作模式选择
0	0	连续选择模式
0	1	连续扫描模式
1	0	单脉冲选择模式
1	1	单脉冲扫描模式

ADA0ETS1	ADA0ETS0	外部触发 (ADTRG1脚) 输入有效沿选择
0	0	禁止边沿检测
0	1	下降沿检测
1	0	上升沿检测
1	1	兼有上升沿和下降沿检测

ADA0TMD	触发选择模式
0	软件触发模式
1	外部触发/定时器触发模式

ADA0EF	A/D 转换器显示状态
0	A/D 转换停止
1	A/D 转换正在进行

注意事项

- 0 位的写操作将被忽略。
- 当 A/D 转换使能(ADA0CE = 1)后, 禁止改变 ADA0M1.ADA0FR2 ~ ADA0M1.ADA0FR0 位。
- 在如下模式下, 在 A/D 转换停止后(ADA0CE 位 = 0), 向寄存器 ADA0M0, ADA0M2, ADA0S, ADA0PFM, 和 ADA0PFT 进行写操作, 然后再使能 A/D 转换的操作 (ADA0CE 位 = 1)。
 - 正常转换模式
 - 单脉冲选择模式/在高速转换模式下的单脉冲扫描模式

如果寄存器 ADA0M0, ADA0M2, ADA0S, ADA0PFM, 和 ADA0PFT 在 A/D 转换过程中其它模式下(ADA0EF = 1)进行写操作, 将根据模式进行下述操作。

 - 软件触发模式
A/D 转换停止并从头开始。
 - 硬件触发模式
A/D 转换停止并进入触发等待状态。
- 要选择外部触发模式/定时器触发模式(ADA0TMD 位 = 1), 则先设定高速转换模式(ADA0M1.ADA0HS1 位 = 1)。在 A/D 使能转换操作(ADA0CE 位 = 1)后的稳定期间内, 不要立即输入触发。
- 当不使用 A/D 转换器时, 设置 ADA0CE 位 0 停止操作以减少功耗。

(2) A/D 转换器模式寄存器 1 (ADA0M1)

该寄存器是 8 位寄存器，用于选择转换时间。

可由 1 位或 8 位存储器操作指令读写该寄存器。

复位信号产生将该寄存器清零(00H)。

复位后: 00H R/W 地址: FFFF201H

	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(A/D 转换时间)选择
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 = 1)后，禁止改变 ADA0M1 的值。
 2. 要选择外部触发模式/定时器触发模式(ADA0M0.ADA0TMD 位=1)，则先设定高速转换模式(ADA0HS1 位= 1)。在 A/D 使能转换操作(ADA0CE 位 = 1)后的稳定期间内，不要立即输入触发。
 3. 确保第 6 ~ 3 位清零。

备注 A/D 转换时间设置示例，见表 13-2 和 13-3。

表 13-2. 正常转换模式下转换时间的选择 (ADA0HS1 位 = 0)

ADA0FR2 to ADA0FR0 位	A/D Conversion Time				
	稳定时间 + 转换时间 + 等待时间	f _{xx} = 20 MHz	f _{xx} = 16 MHz	f _{xx} = 4 MHz	触发响应时间
000	$13/f_{xx} + 26/f_{xx} + 26/f_{xx}$	禁止设置	禁止设置	16.25 μs	$4/f_{xx}$
001	$26/f_{xx} + 52/f_{xx} + 52/f_{xx}$	6.5 μs	8.125 μs	禁止设置	$5/f_{xx}$
010	$39/f_{xx} + 78/f_{xx} + 78/f_{xx}$	9.75 μs	12.1875 μs	禁止设置	$6/f_{xx}$
011	$50/f_{xx} + 104/f_{xx} + 104/f_{xx}$	12.9 μs	16.125 μs	禁止设置	$7/f_{xx}$
100	$50/f_{xx} + 130/f_{xx} + 130/f_{xx}$	15.5 μs	19.375 μs	禁止设置	$8/f_{xx}$
101	$50/f_{xx} + 156/f_{xx} + 156/f_{xx}$	18.1 μs	22.625 μs	禁止设置	$9/f_{xx}$
110	$50/f_{xx} + 182/f_{xx} + 182/f_{xx}$	20.7 μs	禁止设置	禁止设置	$10/f_{xx}$
111	$50/f_{xx} + 208/f_{xx} + 208/f_{xx}$	23.3 μs	禁止设置	禁止设置	$11/f_{xx}$

备注	稳定时间:	A/D 转换器建立时间 (1 μs 或更长)
	转换时间:	实际 A/D 转换时间 (2.6 ~ 10.4 μs)
	等待时间:	在下次转换前插入的等待时间
	触发响应时间:	如果在稳定时间后产生一个软件触发, 外部触发或时间触发, 此触发将插入转换时间前。

正常转换模式下, 转换在稳定时间结束后开始。设置 ADA0M0.ADA0CE = 1 后, 稳定时间开始。A/D 转换只在转换时间(2.6 ~ 10.4 μs)过程中执行。转换结束后操作停止; 等待时间结束后, 产生中断请求信号(INTAD)。

由于等待时间下转换操作已经停止, 所以工作电流降低。

注意事项 1. 设置 2.6 μs ≤ 转换时间 ≤ 10.4 μs 。

2. A/D 转换期间, 如果对寄存器 ADA0M0, ADA0M2, ADA0S, ADA0PFM, 和 ADA0PFT 进行写操作或输入触发信号, 再次转换将被执行。然而, 如果稳定时间结束时序与向这些寄存器的写操作或者与输入触发信号冲突, 64 个时钟周期的稳定时间再次被插入。如果再次插入的稳定时间结束时序再次产生冲突, 稳定时间再次被插入。因此不要设置触发输入间隔和控制寄存器写间隔小于等于 64 个时钟周期。

表 13-3. 高速转换模式下转换时间的选择 (ADA0HS1 位 = 1)

ADA0FR2~ ADA0FR0 位	A/D 转换时间				
	转换时间 (+稳定时间)	f _{xx} = 20 MHz	f _{xx} = 16 MHz	f _{xx} = 4 MHz	触发响应时间
000	26/f _{xx} (+ 13/f _{xx})	禁止设置	禁止设置	6.5 μ s (+ 3.25 μ s)	4/f _{xx}
001	52/f _{xx} (+ 26/f _{xx})	2.6 μ s (+ 1.3 μ s)	3.25 μ s (+ 1.625 μ s)	禁止设置	5/f _{xx}
010	78/f _{xx} (+ 39/f _{xx})	3.9 μ s (+ 1.95 μ s)	4.875 μ s (+ 2.4375 μ s)	禁止设置	6/f _{xx}
011	104/f _{xx} (+ 50/f _{xx})	5.2 μ s (+ 2.5 μ s)	6.5 μ s (+ 3.125 μ s)	禁止设置	7/f _{xx}
100	130/f _{xx} (+ 50/f _{xx})	6.5 μ s (+ 2.5 μ s)	8.125 μ s (+ 3.125 μ s)	禁止设置	8/f _{xx}
101	156/f _{xx} (+ 50/f _{xx})	7.8 μ s (+ 2.5 μ s)	9.75 μ s (+ 3.125 μ s)	禁止设置	9/f _{xx}
110	182/f _{xx} (+ 50/f _{xx})	9.1 μ s (+ 2.5 μ s)	禁止设置	禁止设置	10/f _{xx}
111	208/f _{xx} (+ 50/f _{xx})	10.4 μ s (+ 2.5 μ s)	禁止设置	禁止设置	11/f _{xx}

备注 转换时间: 实际 A/D 转换时间 (2.6 ~ 10.4 μ s)
 稳定时间: A/D 转换器建立时间 (1 μ s 或更长)
 触发响应时间: 如果在稳定时间后产生一个软件触发, 外部触发或时间触发, 此触发将插入转换时间前。

高速转换模式下, 转换在稳定时间结束后开始。设置 ADA0M0.ADA0CE = 1 后, 稳定时间开始。A/D 转换只在转换时间(2.6 ~ 10.4 μ s)过程中执行。转换结束后操作停止后, 立刻产生中断请求信号(INTAD)。连续转换模式下, 只在第一次转换前插入稳定时间, 在第二次转换后不插入稳定时间(A/D 转换器持续运行)。

注意事项 1. 设置 $2.6 \mu\text{s} \leq \text{转换时间} \leq 10.4 \mu\text{s}$ 。
 2. 高速转换模式下, 禁止在稳定时间内复写寄存器 ADA0M0, ADA0M2, ADA0S, ADA0PFM, ADA0PFT 和输入触发信号。

(3) A/D 转换器模式寄存器 2 (ADA0M2)

该寄存器用于选择硬件触发模式。
可由 1 位或 8 位存储器操作指令读写该寄存器。
复位信号产生将该寄存器清零(00H)。

After reset: 00H R/W Address FFFF203H							
ADA0M2	7	6	5	4	3	2	1 0
	0	0	0	0	0	0	ADA0TMD1 ADA0TMD0

ADA0TMD1	ADA0TMD0	定义硬件触发模式
0	0	外部触发模式 (当检测到ADTRG ₁ 脚有效沿)
0	1	定时器触发模式0 (当INTTP2CC0中断请求发生)
1	0	定时器触发模式 1 (当INTTP2CC1 中断请求发生)
1	1	禁止设置

- 注意事项
- 在如下模式下，在 A/D 转换停止后(ADA0M0.ADA0CE 位 = 0)，向寄存器 ADA0M2 进行写操作，然后再使能 A/D 转换的操作（ADA0CE 位 = 1）。
 - 正常转换模式
 - 单脉冲选择模式/在高速转换模式下的单脉冲扫描模式
 - 确保第 7 ~ 2 位清零。

(4) 模拟输入通道选择寄存器 0 (ADA0S)

该寄存器选择输入模拟电平引脚。该引脚的信号将被转换为数字信号。

可由 1 位或 8 位存储器操作指令读写该寄存器。

复位信号产生将该寄存器清零(00H)。

复位后: 00H R/W 地址: FFFF02H

	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	AN10	AN10
0	0	0	1	AN11	AN10, AN11
0	0	1	0	AN12	AN10 ~ AN12
0	0	1	1	AN13	AN10 ~ AN13
0	1	0	0	AN14	AN10 ~ AN14
0	1	0	1	AN15	AN10 ~ AN15
0	1	1	0	AN16	AN10 ~ AN16
0	1	1	1	AN17	AN10 ~ AN17
1	0	0	0	AN18	AN10 ~ AN18
1	0	0	1	AN19	AN10 ~ AN19
1	0	1	0	AN110	AN10 ~ AN110
1	0	1	1	AN111	AN10 ~ AN111
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. 在如下模式下，在 A/D 转换停止后(ADA0M0.ADA0CE 位 = 0)，向寄存器 ADA0S 进行写操作，然后再使能 A/D 转换的操作 (ADA0CE 位 = 1)。

- 正常转换模式
- 单脉冲选择模式/在高速转换模式下的单脉冲扫描模式

2. 确保第 7 ~ 4 位清零。

(5) A/D 转换结果寄存器 n, nH (ADA0CRn, ADA0CRnH)

ADA0CRn 和 ADA0CRnH 寄存器存储 A/D 转换结果。

该寄存器是只读的，可由 16 位或 8 位存储器操作指令进行读写。但是，由 16 位存储器操作指令访问寄存器 ADA0CRn，8 位存储器操作指令访问寄存器 ADA0CRnH。10 位转换结果存于寄存器 ADA0CRn 的高 10 位，其低 6 位的读取值为 0。转换结果的高 8 位存于 ADA0CRnH。

★

注意事项 下述状态禁止访问寄存器 ADA0CRn 和 ADA0CRnH。

详细情况，参见 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。

- 当 CPU 使用子时钟，且主时钟晶振停止振荡。
- 当 CPU 使用内部晶振时钟。

复位后：未定义 R 地址: ADA0CR0 FFFF210H, ADA0CR1 FFFF212H,
ADA0CR2 FFFF214H, ADA0CR3 FFFF216H,
ADA0CR4 FFFF218H, ADA0CR5 FFFF21AH,
ADA0CR6 FFFF21CH, ADA0CR7 FFFF21EH,
ADA0CR8 FFFF220H, ADA0CR9 FFFF222H,
ADA0CR10 FFFF224H, ADA0CR11 FFFF226H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADA0CRn (n = 0 ~ 11)	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

复位后：未定义 R 地址: ADA0CR0H FFFF211H, ADA0CR1H FFFF213H,
ADA0CR2H FFFF215H, ADA0CR3H FFFF217H,
ADA0CR4H FFFF219H, ADA0CR5H FFFF21BH,
ADA0CR6H FFFF21DH, ADA0CR7H FFFF21FH,
ADA0CR8H FFFF221H, ADA0CR9H FFFF223H,
ADA0CR10H FFFF225H, ADA0CR11H FFFF227H

	7	6	5	4	3	2	1	0
ADA0CRnH (n = 0 ~ 11)	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

注意事项 对寄存器 ADA0M0 和 ADA0S 的写操作可能引起 ADA0CRn 内容的不确定。转换后，在写寄存器 ADA0M0 和 ADA0S 前，读取转换结果。如果不按上述操作进行，正确转换结果可能不能读取。

如下为模拟信号输入引脚(ANI0 to ANI11)和 A/D 转换结果(ADA0CRn)的关系。

$$SAR = \text{INT} \left(\frac{V_{IN}}{AV_{REF0}} \times 1,024 + 0.5 \right)$$

$$ADA0CR^{Note} = SAR \times 64$$

或,

$$(SAR - 0.5) \times \frac{AV_{REF0}}{1,024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{REF0}}{1,024}$$

INT(): 函数, 返回()中的整数值

V_{IN}: 模拟输入电压

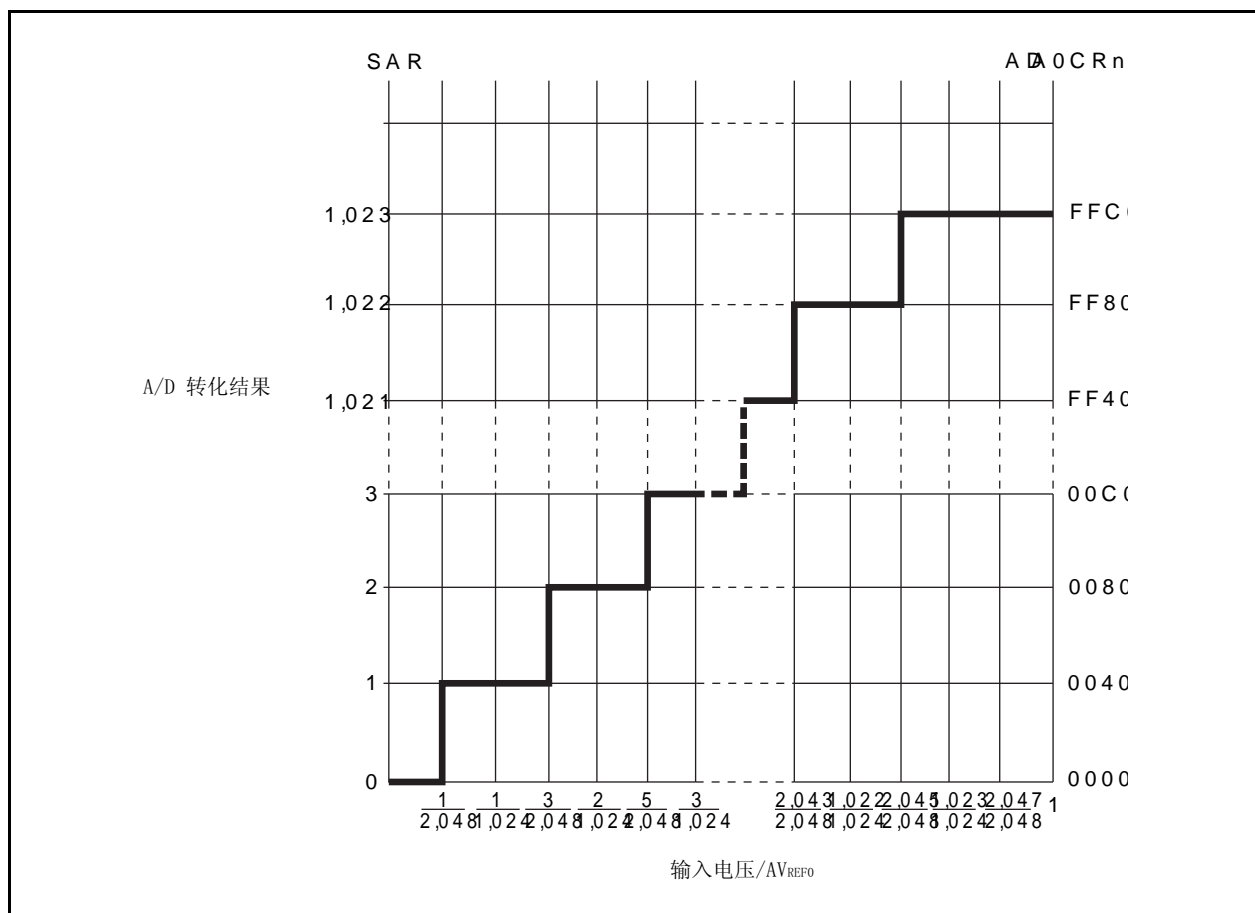
AV_{REF0}: AV_{REF0} 引脚电压

ADA0CR: 寄存器 ADA0CRn 的值

注意事项 ADA0CRn 的低 6 位恒为 0。

如下显示了模拟输入电压和 A/D 转换结果之间的关系。

图 13-2. 模拟输入电压和 A/D 转换结果之间的关系



(6) 电压不足比较模式寄存器 (ADA0PFM)

该寄存器是 8 位寄存器，用于设置电压不足比较模式。

可由 1 位或 8 位存储器操作指令读写该寄存器。

复位信号产生将该寄存器清零(00H)。

复位后:00H R/W 地址:FFFF004H

	<7>	6	5	4	3	2	1	0
ADA0PFM	ADA0PFE	ADA0PFC	0	0	0	0	0	0

ADA0PFE	电压不足比较使能/禁止选择
0	电压不足比较禁止
1	电压不足比较使能

ADA0PFC	电压不足比较模式选择
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. 选择模式下，寄存器 **ADA0PFT** 所设置的 8 位数据与寄存器 **ADA0CRnH** 的值进行比较，**ADA0CRnH** 由 **ADA0S** 进行设置。如果结果与 **ADA0PFC** 位所设置的条件匹配，转换结果存储于寄存器 **ADA0CRn** 并产生中断信号 **INTAD**。如果不匹配，不产生中断信号 **INTAD**。
 2. 扫描模式下，寄存器 **ADA0PFT** 所设置的 8 位数据与寄存器 **ADA0CRnH** 的值进行比较。如果结果与 **ADA0PFC** 位所设置的条件匹配，转换结果存储于寄存器 **ADA0CRn** 并产生中断信号 **INTAD**。如果不匹配，不产生中断信号 **INTAD**。无论比较结果如何，扫描模式继续工作，转换结果存储于寄存器 **ADA0CRn** 中，直到扫描操作完成。但是，扫描操作完成后不产生中断信号 **INTAD**。
 3. 在如下模式下，在 A/D 转换停止后(**ADA0M0.ADA0CE** 位 = 0)，向寄存器 **ADA0PFM** 进行写操作，然后再使能 A/D 转换的操作 (**ADA0CE** 位 = 1)。
 - 正常转换模式
 - 单脉冲选择模式/在高速转换模式下的单脉冲扫描模式

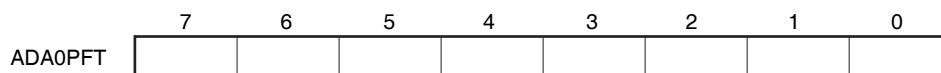
(7) 电压不足比较阈值寄存器 (ADA0PFT)

该寄存器设置电压不足比较模式下的比较值。

可由 1 位或 8 位存储器操作指令读写该寄存器。

复位信号产生将该寄存器清零(00H)。

复位后: 00H R/W 地址: FFFF205H



注意事项 在如下模式下，在 A/D 转换停止后(ADA0M0.ADA0CE 位 = 0)，向寄存器 ADA0PFT 进行写操作，然后再使能 A/D 转换的操作（ADA0CE 位 = 1）。

- 正常转换模式
- 单脉冲选择模式/在高速转换模式下的单脉冲扫描模式

13.5 操作

13.5.1 基本操作

- <1> 使用寄存器 ADA0M0, ADA0M1, ADA0M2, 和 ADA0S 设置操作模式, 触发模式, 和执行 A/D 的转换时间。当寄存器的 ADA0CE 位被设置, 软件触发模式下转换开始, 而外部/时间触发模式下等待触发, 才开始 A/D 转换。
- <2> A/D 转换开始后, 采样和保持电路对被选模拟输入通道的输入电压进行采样。
- <3> 当采样和保持电路采集输入信号到指定状态, 电路进入保持状态, 保持输入电压知道 A/D 转换完成。
- <4> 设置逐次逼近寄存器(SAR)的第 9 位。分接选择器选择(1/2) AV_{REF0} 作为串联电阻串的分接电压。
- <5> 串联电阻串的电压和模拟输入电压的电压差与电压比较器比较。如果模拟输入电压高于(1/2) AV_{REF} , 则 SAR 的 MSB=1。如果模拟输入电压低于(1/2) AV_{REF} , 则 SAR 的 MSB = 0。
- <6> 接下来, SAR 的第 8 位自动置 1, 并进入下一个比较过程。根据第 9 位的预置值选择串联电阻串的分接电压, 具体描述如下。
 - 第 9 位 = 1: (3/4) AV_{REF0}
 - 第 9 位 = 0: (1/4) AV_{REF0}
 比较分接电压与采样电压, 根据结果, 设置 SAR 的第 8 位, 如下所示。
 模拟输入电压 \geq 分接电压: 第 8 位= 1
 模拟输入电压 \leq 分接电压: 第 8 位= 0
- <7> 按此方式继续进行比较, 直至 SAR 的第 0 位。
- <8> 全部 10 位比较完成后, 在 SAR 中保留一个有效的数值结果, 然后将结果传送至 A/D 转换结果寄存器 ADA0CRn 中。同时也会产生 A/D 转换结束中断请求(INTAD)。
- <9> 单脉冲模式下, 转换被停止^注。单脉冲扫描模式下, 扫描一次后转换被停止^注。连续选择模式下, 重复步骤 <2> ~ <8>, 直至 ADA0M0.ADA0CE 被清零。连续扫描模式下, 重复步骤<2> ~ <8>以扫描每个通道。

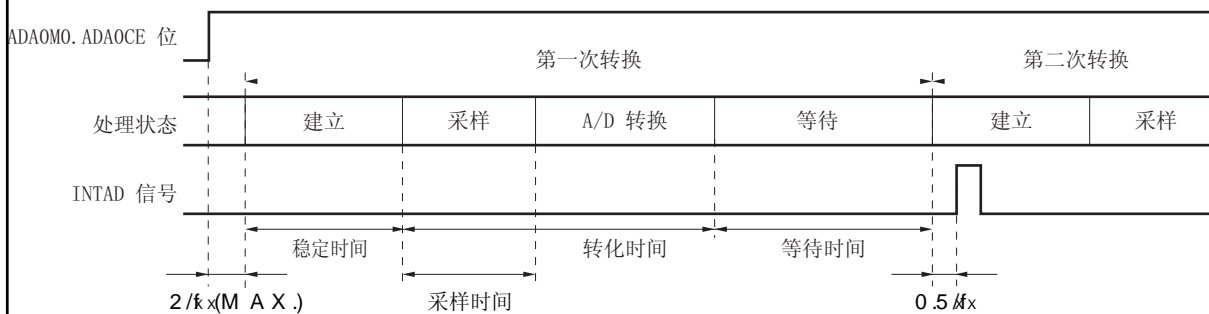
注 外部触发模式/时钟触发模式 0/时钟触发模式 1 下, 进入触发等待状态。

备注 触发等待状态指的是等稳定时间过去后的状态。

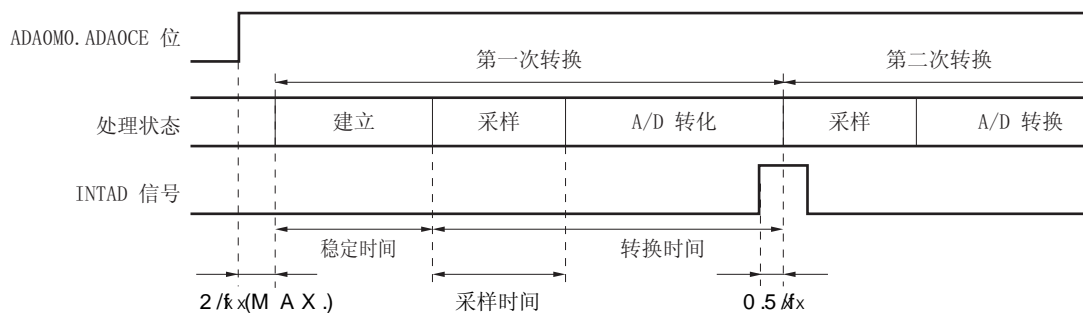
13.5.2 转换操作时序

图 13-3. 转换操作时序(连续转换)

(1) 正常模式下的操作 (ADA0HS1 位 = 0)



(2) 高速转换模式下的操作 (ADA0HS1 位 = 1)



ADA0FR2 ~ ADA0FR0 位	稳定时间	转换时间(采样时间)	等待时间	触发响应时间
000	$13/f_{xx}$	$26/f_{xx} (4/f_{xx})$	$26/f_{xx}$	$4/f_{xx}$
001	$26/f_{xx}$	$52/f_{xx} (8/f_{xx})$	$52/f_{xx}$	$5/f_{xx}$
010	$39/f_{xx}$	$78/f_{xx} (12/f_{xx})$	$78/f_{xx}$	$6/f_{xx}$
011	$50/f_{xx}$	$104/f_{xx} (16/f_{xx})$	$104/f_{xx}$	$7/f_{xx}$
100	$50/f_{xx}$	$130/f_{xx} (20/f_{xx})$	$130/f_{xx}$	$8/f_{xx}$
101	$50/f_{xx}$	$156/f_{xx} (24/f_{xx})$	$156/f_{xx}$	$9/f_{xx}$
110	$50/f_{xx}$	$182/f_{xx} (28/f_{xx})$	$182/f_{xx}$	$10/f_{xx}$
111	$50/f_{xx}$	$208/f_{xx} (32/f_{xx})$	$208/f_{xx}$	$11/f_{xx}$

备注 上述时序适用于稳定时间内产生触发的情况。如果稳定时间过后产生触发，触发响应时间将被插入。

13.5.3 触发模式

通过设置触发模式确定开始转换操作的时序。触发模式包括软件触发模式和硬件触发模式。硬件触发模式包括定时器触发模式 0 和 1，和外部触发模式。ADA0M0.ADA0TMD 用于设置 ADA0M0.ADA0TMD。ADA0M2.ADA0TMD1 和 ADA0M2.ADA0TMD0 位用于设置硬件 ADA0M0.ADA0TMD。

(1) 软件触发模式

当 ADA0M0.ADA0CE = 1，由寄存器 ADA0S 设置的模拟输入引脚(ANI0 ~ ANI11)的信号被转换。当转换完成，结果存储于寄存器 ADA0CRn 中。同时，产生 A/D 转换结束中断请求信号(INTAD)。

如果 ADA0M0.ADA0MD1 和 ADA0M0.ADA0MD0 设置的操作模式是连续选择/扫描模式，下一次转换开始，除非第一次转换完成后设置 ADA0CE = 0。

当转换开始，ADA0M0.ADA0EF = 1(表明转换正在进行)。

如果转换过程中对寄存器 ADA0M0, ADA0M2, ADA0S, ADA0PFM, 或 ADA0PFT 进行写操作，转换中止并重新开始。但是在正常模式以及在高速转换模式中的单次选择模式/但从扫描模式中，是禁止向这些寄存器写操作的。

(2) 外部触发模式

该模式下，当外部触发信号输入到引脚 ADTRG，由寄存器 ADA0S 设置的模拟输入引脚(ANI0 ~ ANI11)的信号开始转换。ADA0M0.ADA0ETS1 和 ADA0M0.ATA0ETS0 用于设置外部触发的检测边缘(即，上升沿，下降沿，或上升和下降沿)。

无论是否设置 ADA0MD1 和 ADA0MD0 的连续选择，连续扫描，单脉冲选择，或单脉冲扫描模式，当转换完成，结果存储于寄存器 ADA0CRn 中。

当转换开始，ADA0EF = 1(表明转换正在进行)。然而，当 A/D 转换器等待触发时，ADA0EF = 0(表明转换停止)。如果在转换过程中输入有效触发，转换中止并重新开始。

如果转换过程中复写寄存器 ADA0M0, ADA0M2, ADA0S, ADA0PFM, 或 ADA0PFT，转换并不中止，而是等待再次触发。然而，在单次选择模式/单次扫描模式中，是禁止向这些寄存器写操作的。

注意事项 为了选择外部触发模式，首先设置高速转换模式。在 A/D 使能转换操作(ADA0M0.ADA0CE 位 = 1)后的稳定期间内，不要立即输入触发。

备注 触发等待状态指的是等稳定时间过去后的状态。

(3) 定时器触发模式

该模式下，当连接到定时器的捕捉/比较寄存器产生比较匹配中断请求信号(INTTP2CC0 或 INTTP2CC1)时，由寄存器 ADA0S 设置的模拟输入引脚(ANI0 ~ ANI11)的信号开始转换。使用 ADA0TMD1 和 ADA0TMD0 选择信号 INTTP2CC0 或 INTTP2CC1，在特定的比较匹配中断请求信号上升沿，转换开始。当 ADA0CE = 1，A/D 转换器等待触发，并在比较匹配中断请求信号输入时开始转换。

无论是否设置 ADA0MD1 和 ADA0MD0 的连续选择，连续扫描，单脉冲选择，或单脉冲扫描模式，当转换完成，结果存储于寄存器 ADA0CRn 中。同时产生 INTAD 信号，A/D 转换器等待再次触发。

当转换开始，ADA0EF = 1(表明转换正在进行)。然而，当 A/D 转换器等待触发时，ADA0EF = 0(表明转换停止)。如果在转换过程中输入有效触发，转换中止并重新开始。

如果转换过程中对寄存器 ADA0M0, ADA0M2, ADA0S, ADA0PFM, 或 ADA0PFT 进行写操作，转换中止，A/D 转换器等待再次触发。However, writing to these registers is prohibited in the one-shot select mode/one-shot scan mode.

注意事项 为了选择外部触发模式，首先设置高速转换模式。Do not input a trigger during stabilization time that is inserted once after the A/D conversion operation is enabled (ADA0M0.ADA0CE bit = 1).在 A/D 使能转换操作(ADA0M0.ADA0CE 位 = 1)后的稳定期间内，不要立即输入触发。

备注 触发等待状态指的是等稳定时间过去后的状态。

13.5.4 操作模式

有 4 种操作可供使用：连续选择模式，连续扫描模式，单脉冲选择模式和单脉冲扫描模式。

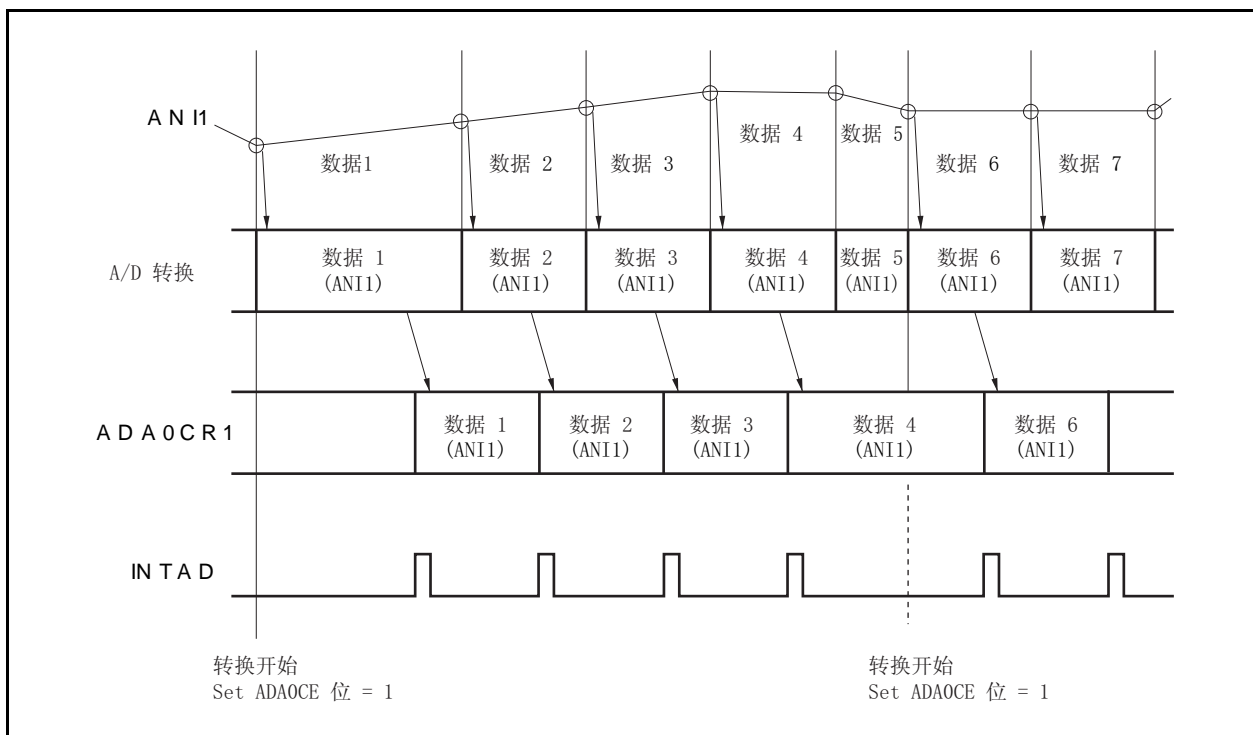
使用 ADA0M0.ADA0MD1 和 ADA0M0.ADA0MD0 设置操作模式。

(1) 连续选择模式

该模式下，由寄存器 ADA0S 选择的某个模拟输入引脚的电压连续转换成数字量。

相应模拟输入引脚的转换结果存储于寄存器 ADA0CRn 中。该模式下，模拟输入引脚与寄存器 ADA0CRn 一一对应。每次 A/D 转换完成，产生 A/D 转换结束中断请求信号 (INTAD)。转换完成后，除非 ADA0M0.ADA0CE = 0 (n = 0 to 11)，否则下一次转换开始。

图 13-4. 连续选择模式的操作时序示例 (寄存器 ADA0S = 01H)



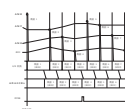
(2) 连续扫描模式

该模式下，模拟输入引脚被依次选择(从引脚 ANI0 到寄存器 ADA0S 所设置的指定引脚)，并转换成数字量。

每次转换的结果存储于与模拟输入引脚对应的寄存器 ADA0CRn 中。当由寄存器 ADA0S 设置的模拟输入引脚的转换完成时，产生 INTAD 信号。除非 ADA0CE = 0 (n = 0 to 11)，否则 A/D 转换再次从引脚 ANI0 开始。

图 13-5. 连续扫描模式的操作时序示例(寄存器 ADA0S = 03H)

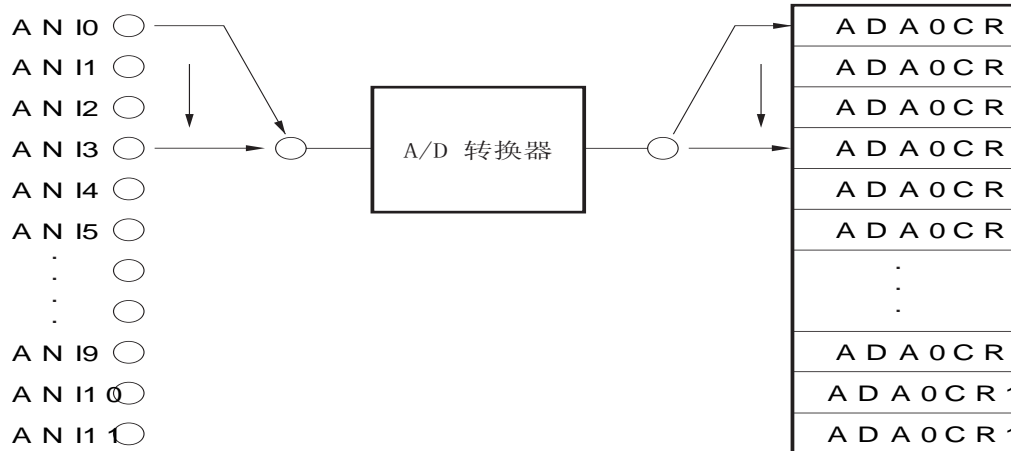
(a) 时序示例



(b) 框图

模拟输入引脚

ADA0CRn 即采取

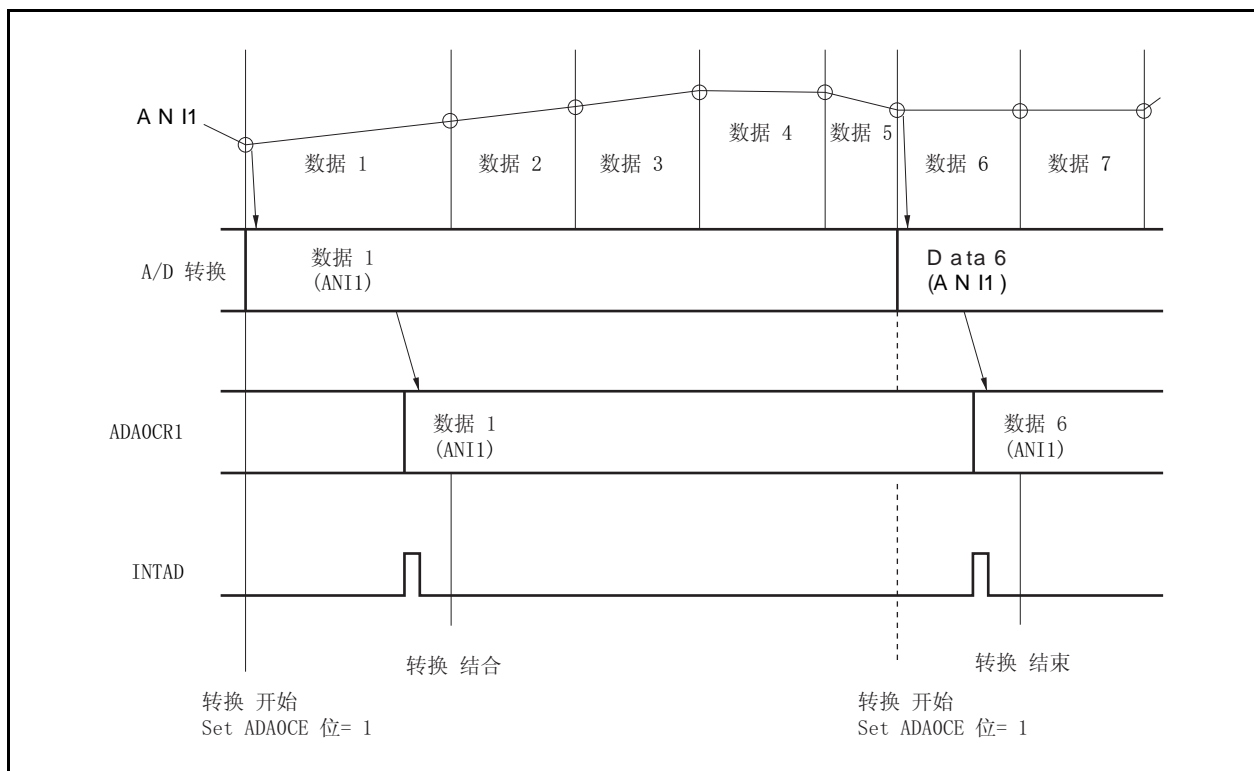


(3) 单脉冲选择模式

该模式下，由寄存器 **ADA0S** 设置的模拟输入引脚的电压转换成数字量，且只转换一次。

相应模拟输入引脚的转换结果存储于寄存器 **ADA0CRn** 中。该模式下，模拟输入引脚与寄存器 **ADA0CRn** 一一对应。每次 A/D 转换完成，产生 A/D 转换结束中断请求信号(**INTAD**)。A/D 转换($n = 0$ to 11)完成后，转换操作停止。

图 13-6. 单脉冲选择模式的操作时序示例(寄存器 **ADA0S** = **01H**)

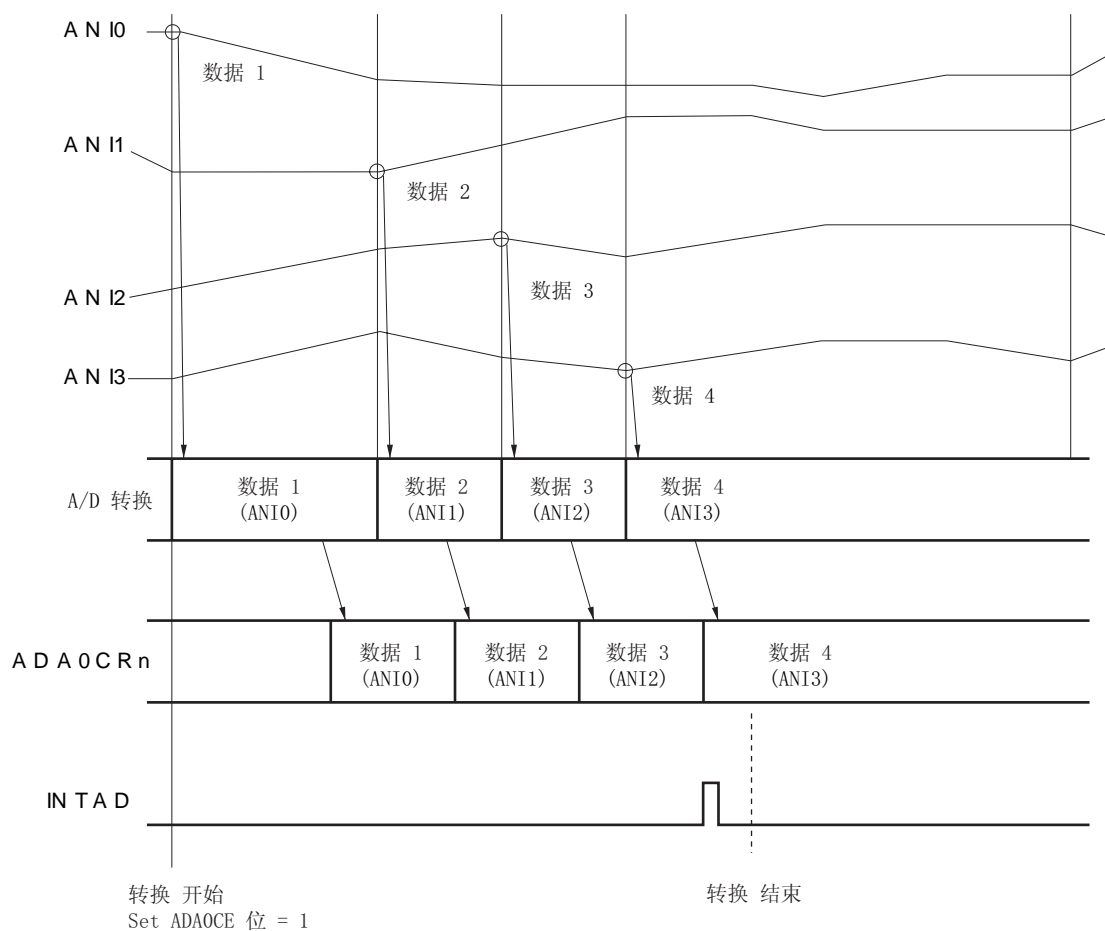
**(4) 单脉冲扫描模式**

该模式下，模拟输入引脚被依次选择(从引脚 **AN10** 到寄存器 **ADA0S** 所设置的指定引脚)，并转换成数字量。

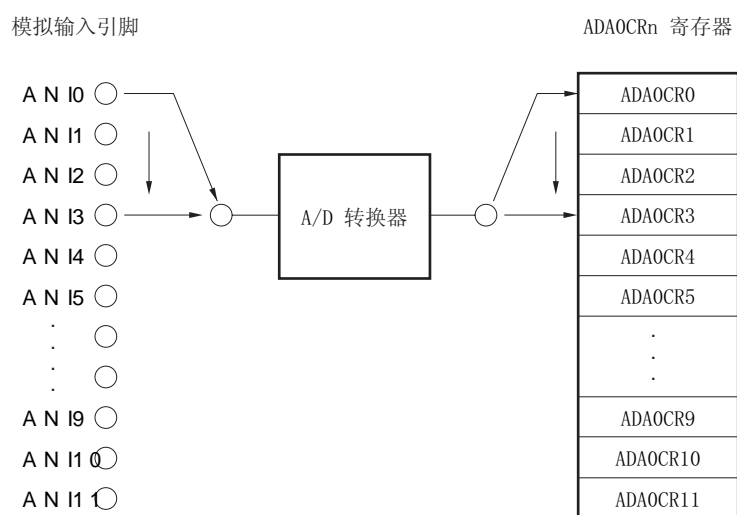
每次转换的结果存储于与模拟输入引脚对应的寄存器 **ADA0CRn** 中。当由寄存器 **ADA0S** 设置的模拟输入引脚的转换完成时，产生 **INTAD** 信号。A/D 转换($n = 0$ to 11)完成后，转换操作停止。

图 13-7. 单脉冲扫描模式的操作时序示例(寄存器 ADA0S = 03H)

(a) 时序示例



(b) 框图



13.5.5 电压不足比较模式

寄存器 ADA0PFM 和 ADA0PFT 用于控制 A/D 转换结束中断请求(INTAD)。

- 当 ADA0PFM.ADA0PFE = 0，每当转换完成时产生信号 INTAD(正常使用 A/D 转换器)。
- 当 ADA0PFE = 1 且 ADA0PFM.ADA0PFC = 0，转换完成后，寄存器 ADA0CRnH 的值与 ADA0PFT 的值比较，只有当 $\text{ADA0CRnH} \geq \text{ADA0PFT}$ 时产生信号 INTAD。
- 当 ADA0PFE = 1 且 ADA0PFC = 1，转换完成后，寄存器 ADA0CRnH 的值与 ADA0PFT 的值比较，只有当 $\text{ADA0CRnH} < \text{ADA0PFT}$ 时产生信号 INTAD。

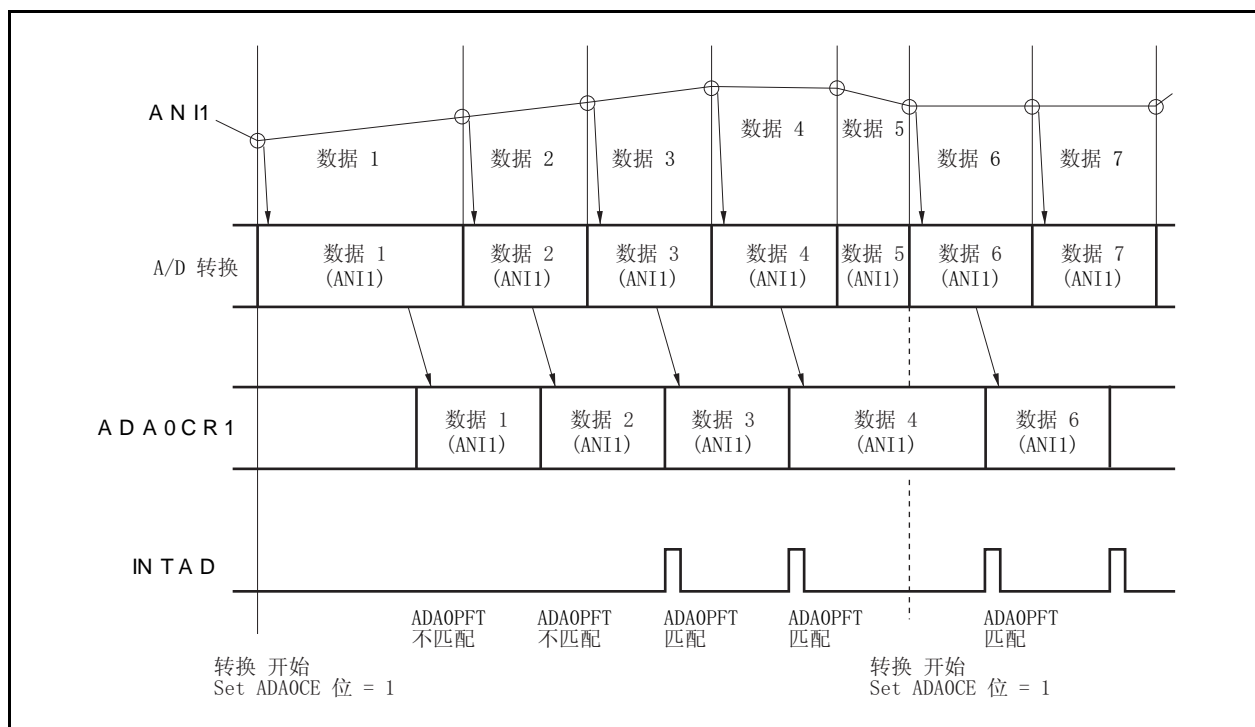
备注 n = 0 to 11

电压不足比较模式下，有 4 种操作可供使用连续选择模式，连续扫描模式，单脉冲选择模式和单脉冲扫描模式。

(1) 连续选择模式

该模式下，由寄存器 **ADA0S** 设置的模拟输入引脚的转换电压结果与寄存器 **ADA0PFT** 的设置值比较。如果电压不足比较的结果与 **ADA0PFC** 设置的条件匹配，转换结果存储于寄存器 **ADA0CRn** 中，产生 **INTAD** 信号。如果不匹配，转换结果存储于寄存器 **ADA0CRn** 中，不产生 **INTAD** 信号。第一次转换完成后，除非 **ADA0M0.ADA0CE = 0** ($n = 0$ to 11)，否则下一次转换开始。

图 13-8. 连续选择模式的操作时序示例
(电压不足比较模式下: **ADA0S = 01H**)

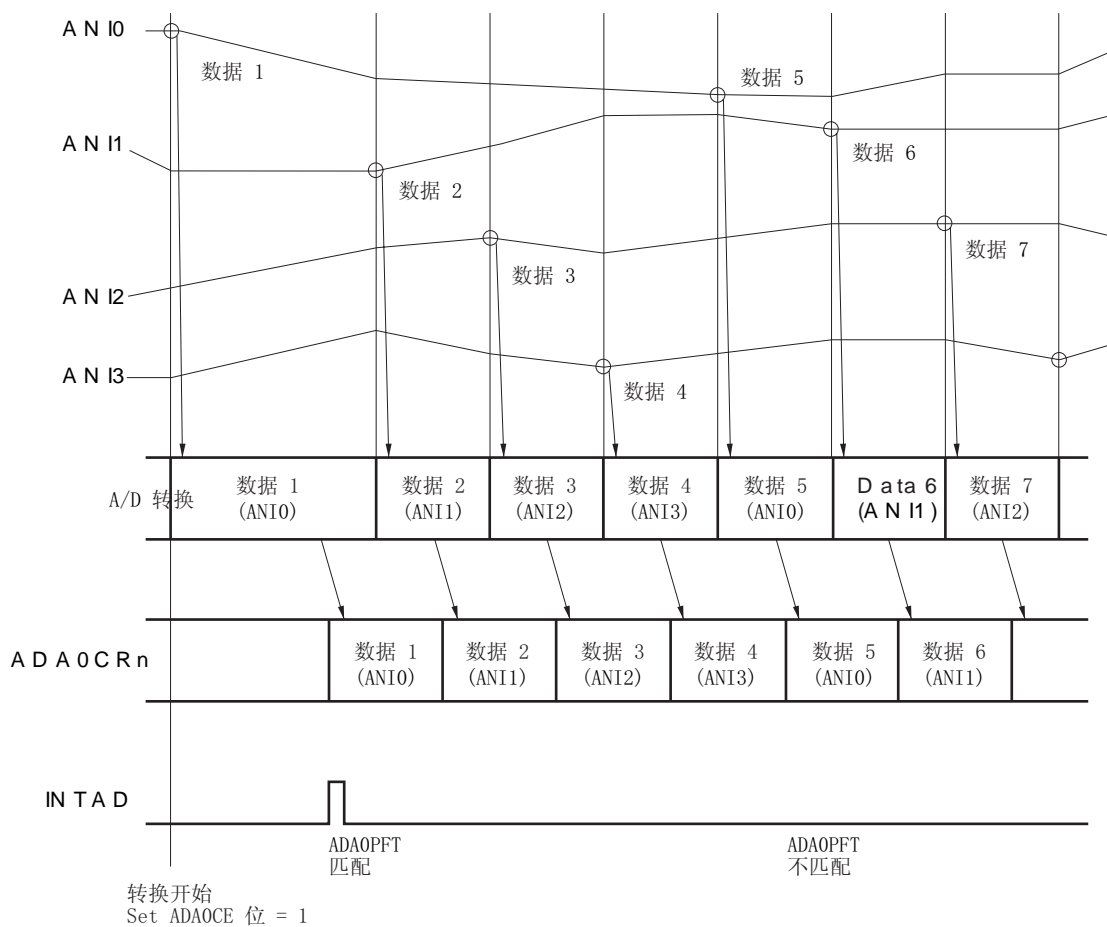
**(2) 连续扫描模式**

该模式下，模拟输入引脚被依次选择(从引脚 **ANIO** 到寄存器 **ADA0S** 所设置的指定引脚)，模拟输入引脚的电压转换结果被存储，寄存器 **ADA0CR0H** 通道 0 的设定值与寄存器 **ADA0PFT** 的值比较。如果电压不足比较的结果与 **ADA0PFC** 设置的条件匹配，转换结果存储于寄存器 **ADA0CRn** 中，产生 **INTAD** 信号。如果不匹配，转换结果存储于寄存器 **ADA0CRn** 中，不产生 **INTAD** 信号。

第一次转换的结果存储于寄存器 **ADA0CR0** 后，模拟输入引脚到寄存器 **ADA0S** 指定引脚的电压连续转换结果连续被存储。转换完成后，除非 **ADA0M0.ADA0CE = 0** ($n = 0$ to 11)，否则下一次转换开始。

图 13-9. 连续扫描模式的操作时序示例
(电压不足比较模式下: ADA0S = 03H)

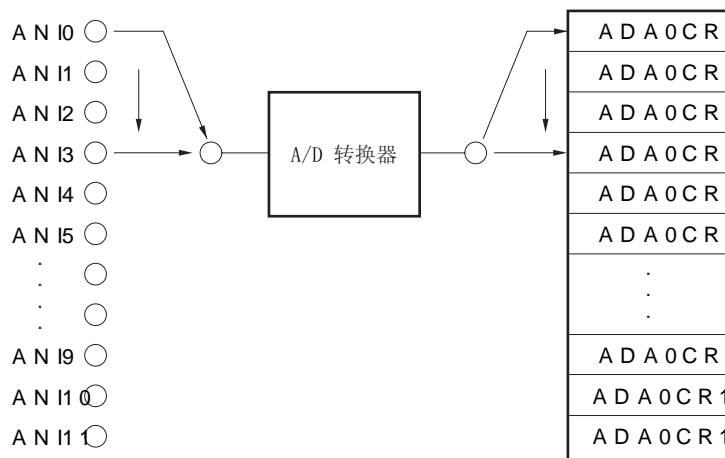
(a) 时序示例



(b) 框图

模拟输入引进

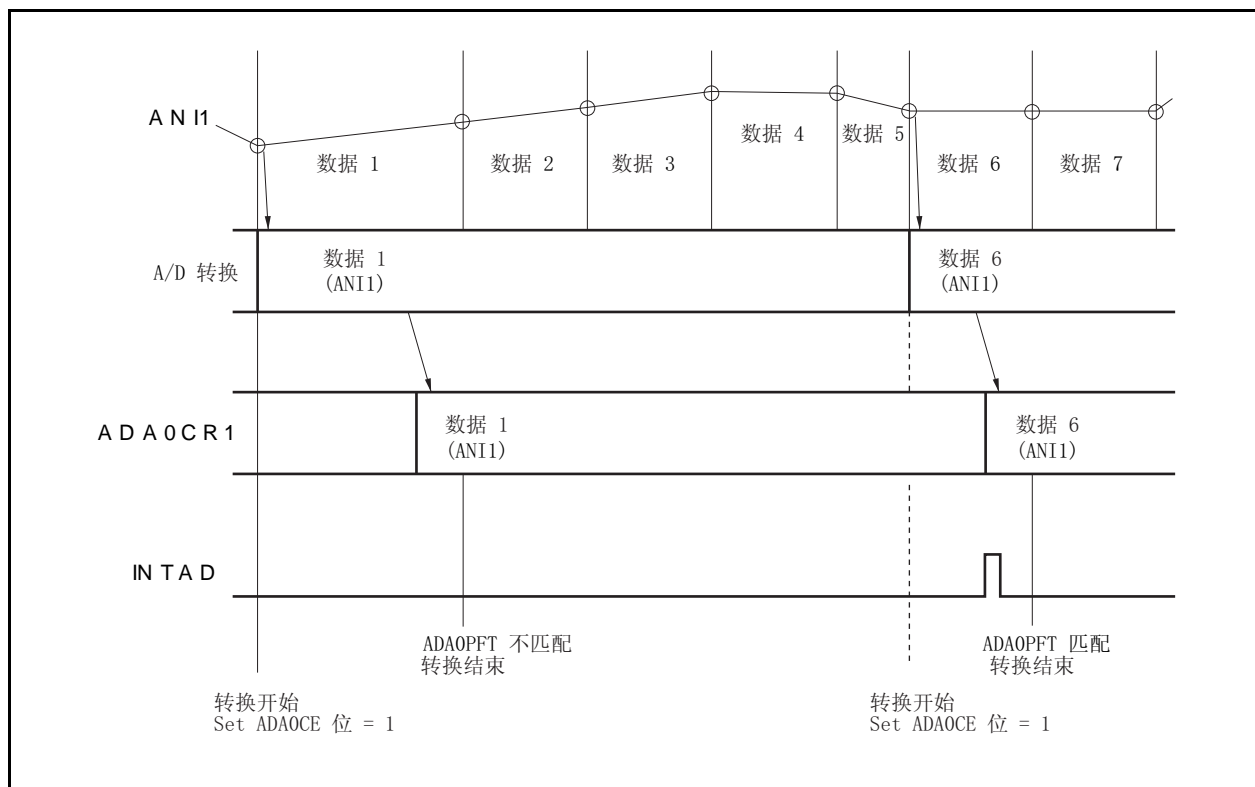
ADA0CRn 寄存器



(3) 单脉冲选择模式

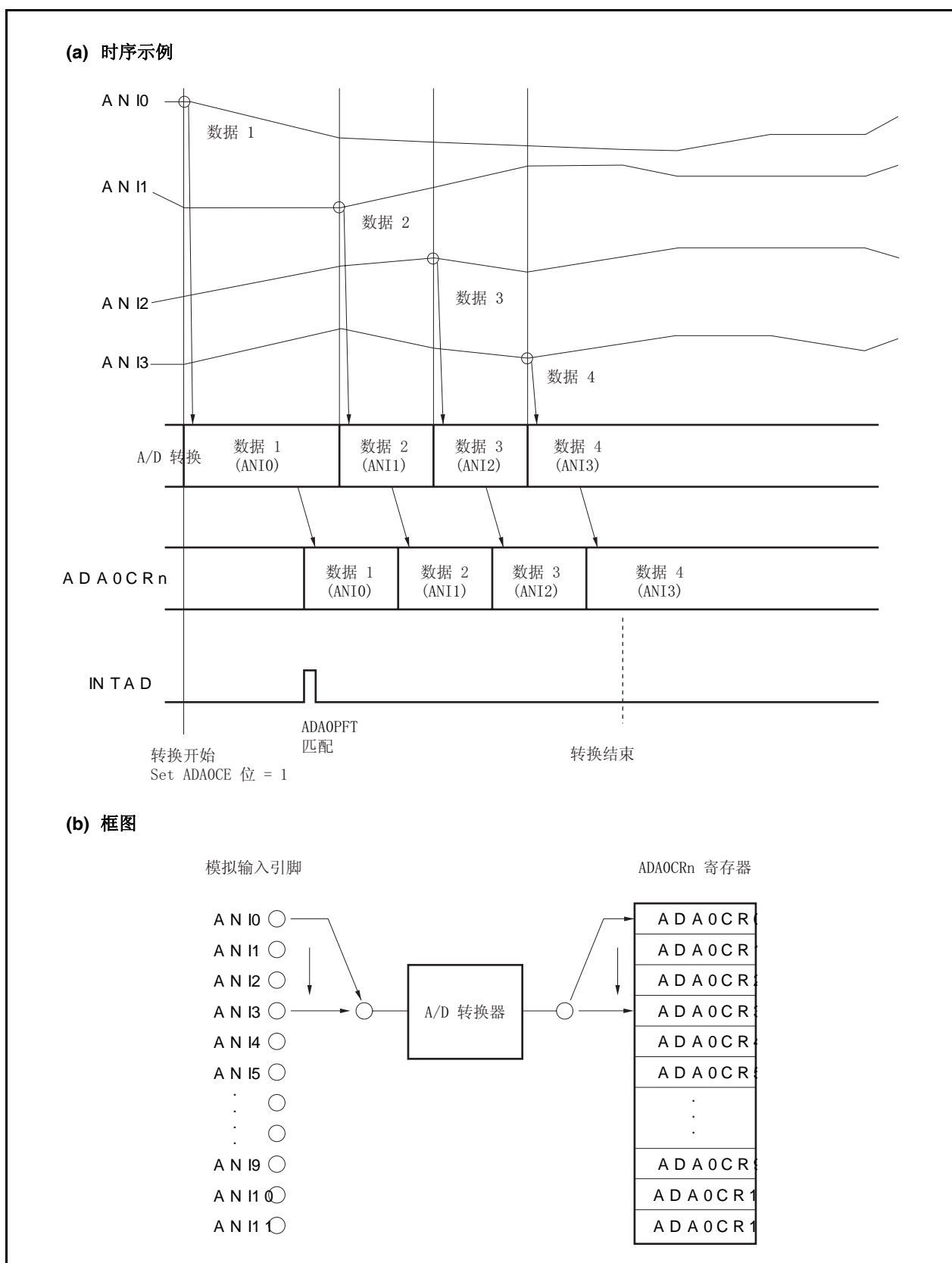
该模式下，由寄存器 **ADA0S** 设置的模拟输入引脚的转换电压结果与寄存器 **ADA0PFT** 的设置值比较。如果电压不足比较的结果与 **ADA0PFC** 设置的条件匹配，转换结果存储于寄存器 **ADA0CRn** 中，产生 **INTAD** 信号。如果不匹配，转换结果存储于寄存器 **ADA0CRn** 中，不产生 **INTAD** 信号。**A/D** 转换完成后，转换操作停止。

图 13-10. 单脉冲选择模式的操作时序示例
(电压不足比较模式下: **ADA0S = 01H**)

**(4) 单脉冲扫描模式**

该模式下，模拟输入引脚被依次选择(从引脚 **ANI0** 到寄存器 **ADA0S** 所设置的指定引脚)，模拟输入引脚的电压转换结果被存储，寄存器 **ADA0CR0H** 通道 0 的设定值与寄存器 **ADA0PFT** 的值比较。如果电压不足比较的结果与 **ADA0PFC** 设置的条件匹配，转换结果存储于寄存器 **ADA0CRn** 中，产生 **INTAD** 信号。如果不匹配，转换结果存储于寄存器 **ADA0CRn** 中，不产生 **INTAD** 信号。第一次转换的结果存储于寄存器 **ADA0CR0** 后，模拟输入引脚到寄存器 **ADA0S** 指定引脚的电压连续转换结果连续被存储。**A/D** 转换完成后，转换操作停止。

图 13-11. 单脉冲扫描模式的操作时序示例
(电压不足比较模式下: ADA0S = 03H)



13.6 注意事项

(1) 不使用 A/D 转换器

当不使用 A/D 转换器时，设置 `ADA0M0.ADA0CE = 0`，使功耗减少。

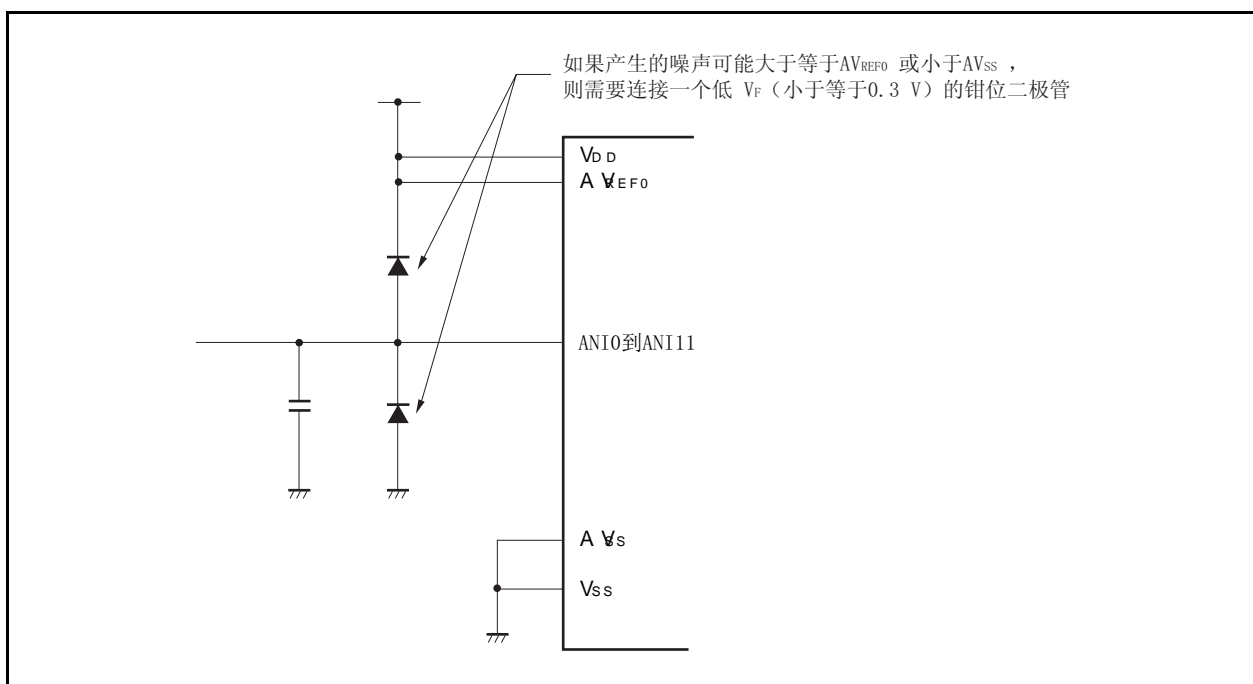
(2) 引脚 ANI0 ~ ANI11 的输入范围

输入指定范围的电压值到引脚 ANI0 ~ ANI11。如果大于等于 AV_{REF0} 或小于等于 AV_{SS} (即使在最大绝对值范围内)的电压被输入到这些引脚，则那个通道的转换值不确定，其它通道的转换值也会受到影响。

(3) 抑制噪声的方法

为了确保 10 位分辨率，引脚 ANI0 ~ ANI11 必须有效抑制噪声。噪声影响随着模拟输入源的输出阻抗的增加而提高。为了降低噪声，推荐连接一个外部电容，如图 13-12 所示。

图 13-12. 模拟输入引脚的处理



(4) I/O 复用

模拟输入引脚(ANI0 ~ ANI11)可功能复用为普通端口引脚。当从引脚 ANI0 ~ ANI11 中选择一个执行 A/D 转换时，不要在转换期间执行读/写端口指令，否则转换分辨率将下降。

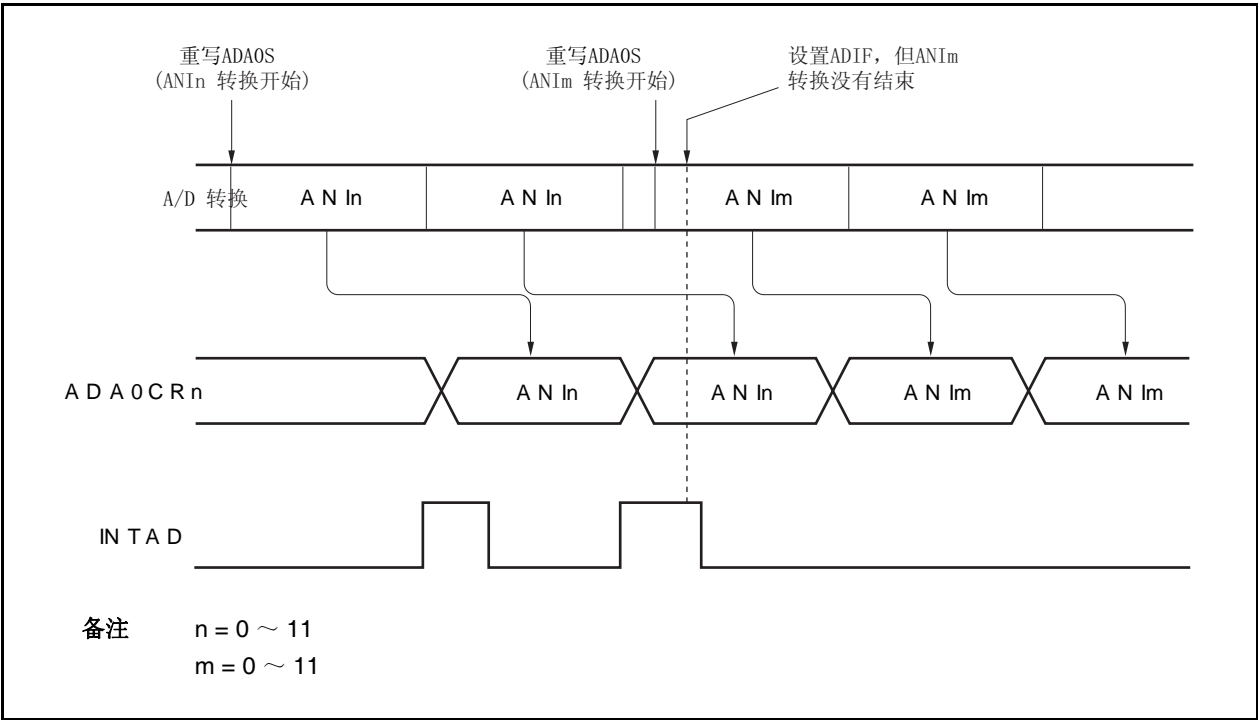
在转换期间设置引脚为输出端口同样会使转换分辨率将下降。其原因是连接端口引脚的外部电路导致输出电流波动。

如果某引脚正在进行 A/D 转换，其临近引脚输入一个数字脉冲，则由于耦合噪声的影响，A/D 转换值可能不准确。因此，在 A/D 转换过程中，确保不使用临近引脚传输脉冲信号。

(5) 中断请求标志 (ADIF)

即使寄存器 ADA0S 的内容改变，中断请求标志 (ADIF)也不清零。因此，如果在 A/D 转换过程中模拟输入引脚改变，上次所选择的模拟输入信号的转换结果可能被存储，换结束中断请求标志可能在寄存器 ADA0S 复写前立刻置 1。如果寄存器 ADA0S 复写后立即读取标志 ADIF，标志 ADIF 可能置 1，即使新选择的模拟输入信号的转换还未完成。当 A/D 转换停止，在再次转换前使标志 ADIF 清零。

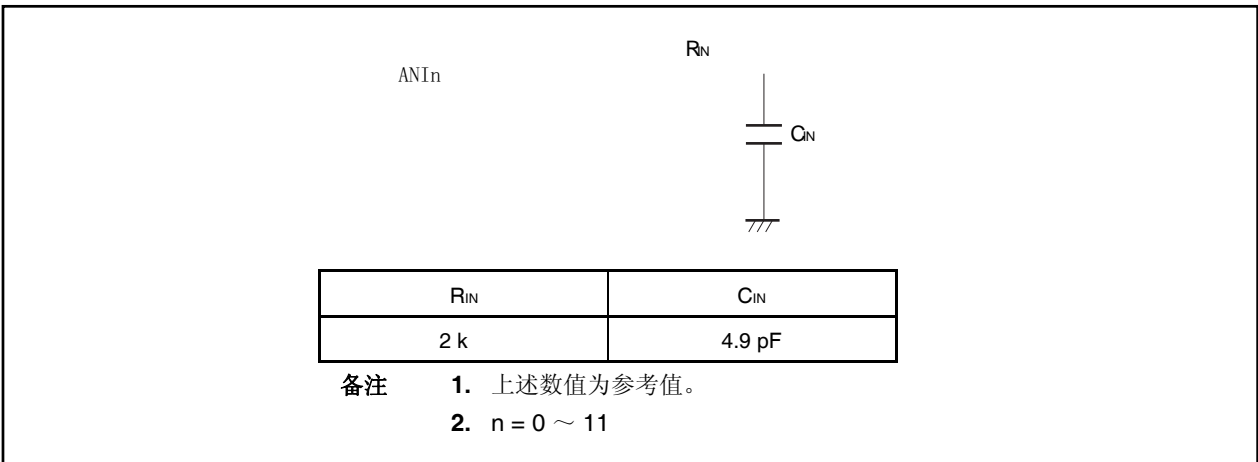
图 13-13. A/D 转换结束中断请求的产生时序



(6) 内部等效电路

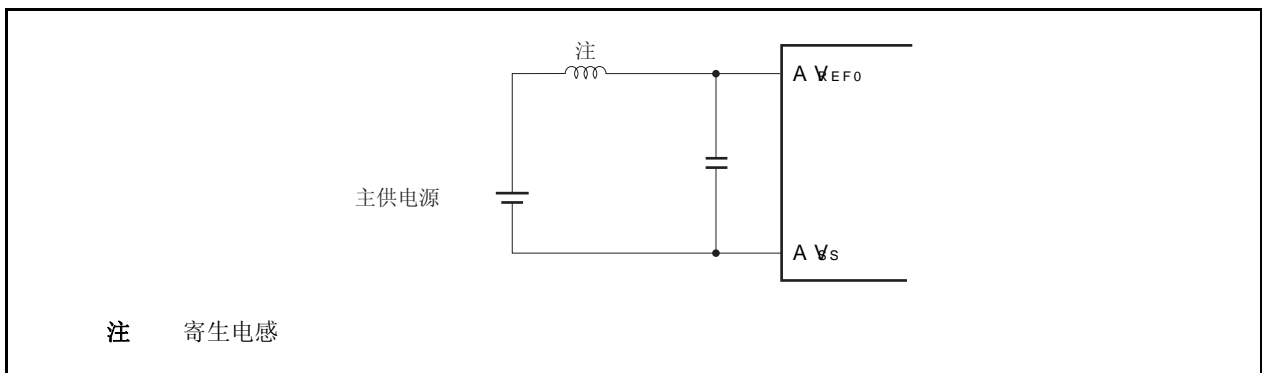
如下展示了模拟输入模块的等效电路。

图 13-14. 引脚 ANIn 的内部等效电路



(7) 引脚 AV_{REF0}

- (a) 引脚 AV_{REF0} 用于 A/D 转换器的供电电压引脚，同样为复用功能端口供电。如图 13-15 所示，当用于备用供电电压时，确保向引脚 AV_{REF0} 提供与 V_{DD} 相同的电压。
- (b) 引脚 AV_{REF0} 也用于 A/D 转换器的参考电压引脚。如果提供给引脚 AV_{REF0} 的源供电电源具有高阻抗，或者如果供电电源低电流负载能力低，参考电压可能被转换中的电流影响(特别是，转换操作使能位 $ADA0CE$ 刚刚置 1 之后)。结果，转换精度会下降。如图 13-15 所示，为了避免上述情况，推荐连接一个电容在引脚 AV_{REF0} 和 AV_{SS} 之间以抑制参考电压的波动。
- (c) 如果提供给引脚 AV_{REF0} 的源供电电源具有高直流阻抗(例如，由于插入一个二极管)，当转换使能时的电压可能比转换停止时的电压低，因为 A/D 转换电流引起电压失真。

图 13-15. 引脚 AV_{REF0} 处理举例(8) 读取 $ADA0CRn$ 寄存器

当执行寄存器 $ADA0M0 \sim ADA0M2$ ， $ADA0S$ ， $ADA0PFM$ 或 $ADA0PFT$ 的写指令，寄存器的 $ADA0CRn$ 内容可能不确定。转换完成后，并在写寄存器 $ADA0M0 \sim ADA0M2$ ， $ADA0S$ ， $ADA0PFM$ 或 $ADA0PFT$ 前，读取转换结果。同样，当响应一个外部/定时器触发时， $ADA0CRn$ 寄存器的内容可能也不确定。在转换结束后并在响应下次的外部/定时器触发之前读取转换结果。在与上述时序不同的时，正确转换结果可能无法读取。

★ (9) 等待模式

因为 A/D 转换器在 STOP 模式下停止操作，转换结果无效，所以功耗可以降低。释放 STOP 模式后，操作恢复，但 STOP 模式释放后的 A/D 转换结果无效。当 STOP 模式释放后使用 A/D 转换器，应在设置 STOP 模式前或释放 STOP 模式后，使 $ADA0M0.ADA0CE = 0$ ，然后在释放 STOP 模式后设置 $ADA0CE = 1$ 。在 IDLE1，IDLE2 或子时钟操作模式下，继续运行。因此，为了减小功耗，应使 $ADA0M0.ADA0CE = 0$ 。在 IDLE1，IDLE2 模式下，由于模拟输入电压值不能保留，IDLE1，IDLE2 模式释放后的 A/D 转换结果无效。进入 IDLE1，IDLE2 模式前的转换结果有效。

(10) 高速转换模式

高速转换模式下，稳定时间内，禁止复写寄存器 $ADA0M0$ ， $ADA0M2$ ， $ADA0S$ ， $ADA0PFM$ ，和 $ADA0PFT$ 并触发输入。

★ (11) A/D 转换时间

A/D 转换时间包括稳定时间、转换时间、等待时间和触发响应时间(详细情况，参见表 13-2 正常转换模式下转换时间的选择 ($ADA0HS1 = 0$) 和表 13-3 高速转换模式下转换时间的选择 ($ADA0HS1 = 1$))。

正常转换模式下，A/D 转换期间，如果对寄存器 $ADA0M0$ ， $ADA0M2$ ， $ADA0S$ ， $ADA0PFM$ ，和 $ADA0PFT$ 进行写操作或者输入触发，开始重新转换。然而，如果稳定时间结束时序与写上述寄存器冲突，或者如果稳定时间结束时序与输入触发冲突，会多插入 64 个时钟的稳定时间。

如果在多插入 64 个时钟的稳定时间结束时，再次发生冲突，稳定时间再次插入。因此，不要设置触发输入间隔和控制写寄存器小于等于 64 个时钟。

★ (12) A/D 转换结果的漂移 variation

由于供电电压的波动，A/D 转换结果可能产生变化，或者可能受到噪声影响。为了减小漂移，使用多次测量求平均值的方法。

★ (13) A/D 转换结果滞后特性

逐次逼近 A/D 转换器维持内部采样和保持电容的模拟输入电压，同时进行 A/D 转换。A/D 转换结束后模拟输入电压保留在内部采样和保持电容中。结果，产生下述现象。

- 当同一个通道用于 A/D 转换，如果电压高于或低于先前 A/D 转换，则出现滞后特征，转换结果受到上次转换值影响。因此，即使同一个模拟输入电压也会产生不同结果。
- 当开关模拟输入通道，可能出现滞后特征，转换结果受到上次转换通道影响。这是因为有一个 A/D 转换器用于 A/D 转换。因此，即使同一个模拟输入电压也会产生不同结果。

13.7 A/D 转换器特征表的阅读方法

以下介绍 A/D 转换器中的专用术语。

(1) 分辨率

这是可识别的最小的模拟输入电压，即每位数字输出的模拟输入电压的百分比，称为 **1 LSB** (最低有效位)。对于满度的 **1 LSB** 的百分比用 **%FSR** (满度范围) 表示。**%FSR** 是以百分比形式表示的一定范围的可变模拟输入电压的比率。与分辨率无关，可如下表示。

$$\begin{aligned} 1\%FSR &= (\text{可变模拟输入电压的最大值} - \text{可变模拟输入电压的最小值})/100 \\ &= (AV_{REF0} - 0)/100 \\ &= AV_{REF0}/100 \end{aligned}$$

当分辨率为 10 位时 **1LSB** 表示如下。

$$\begin{aligned} 1 \text{ LSB} &= 1/2^{10} = 1/1,024 \\ &= 0.098\%FSR \end{aligned}$$

精确度与分辨率无关，而由总误差决定。

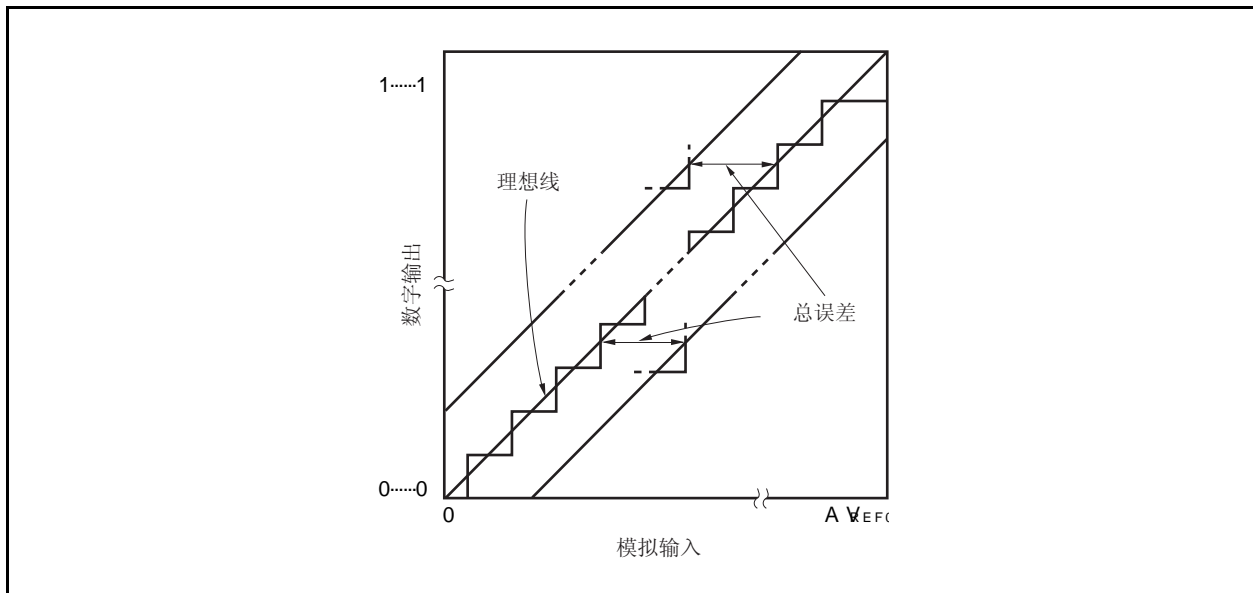
(2) 总误差

总误差是指实际测量值与理论值之间的最大误差。

零度误差、满度误差、积分线性误差和积分非线性误差等组合起来表示总误差。

量化误差不属于特征表中总误差的范围。

图 13-16. 总误差

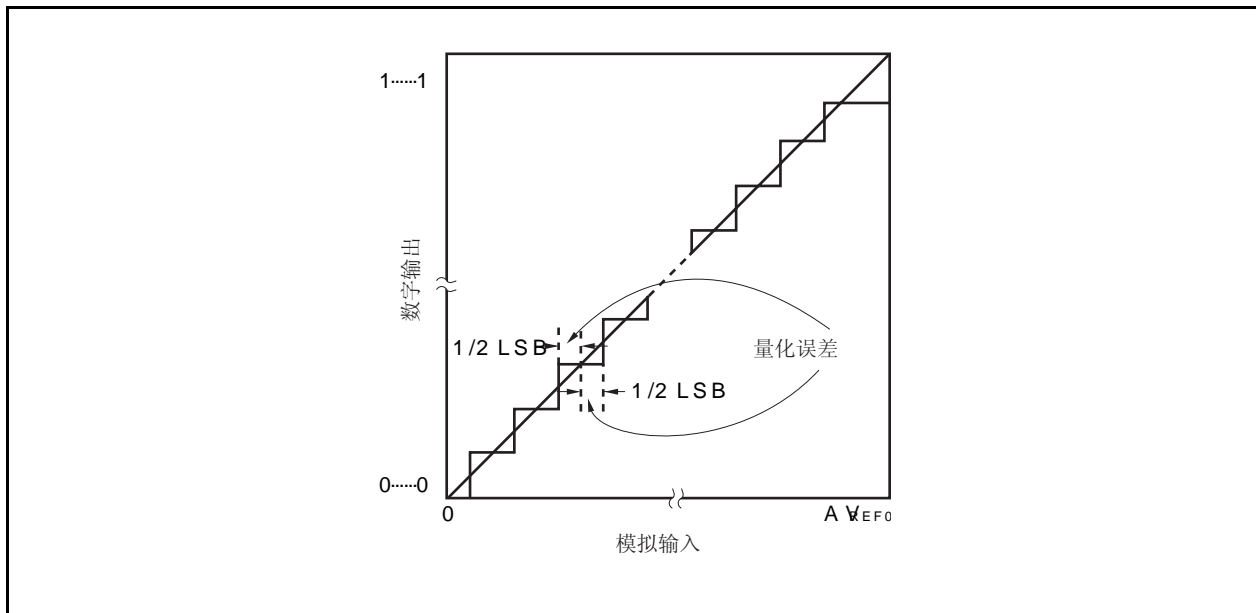


(3) 量化误差

当模拟值转换成数字值时，通常会产生 $\pm 1/2\text{LSB}$ 的误差。在一个 A/D 转换器中，相差 $\pm 1/2\text{LSB}$ 的模拟输入电压被转换成相同的数字代码，因此量化误差不可避免。

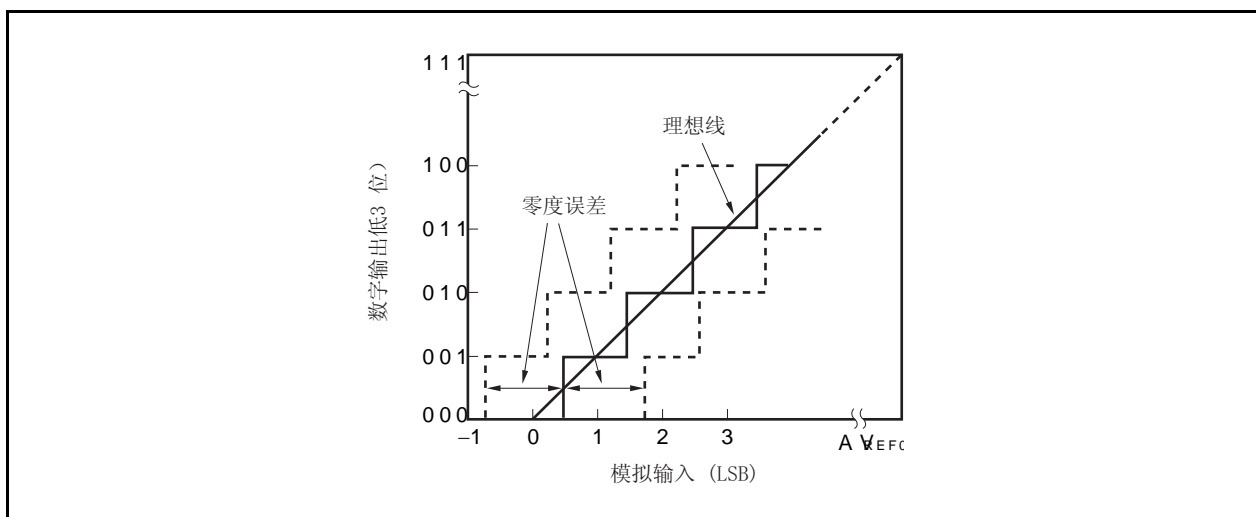
量化误差不属于特征表中总误差、零度误差、满度误差、积分线性误差和微分线性误差的范围。

图 13-17. 量化误差

**(4) 零度误差**

零度误差表示当数字输出范围在 0.....000 ~ 0.....001 之间时模拟输入电压的实际测量值与理论值($1/2\text{LSB}$) 之间的误差。

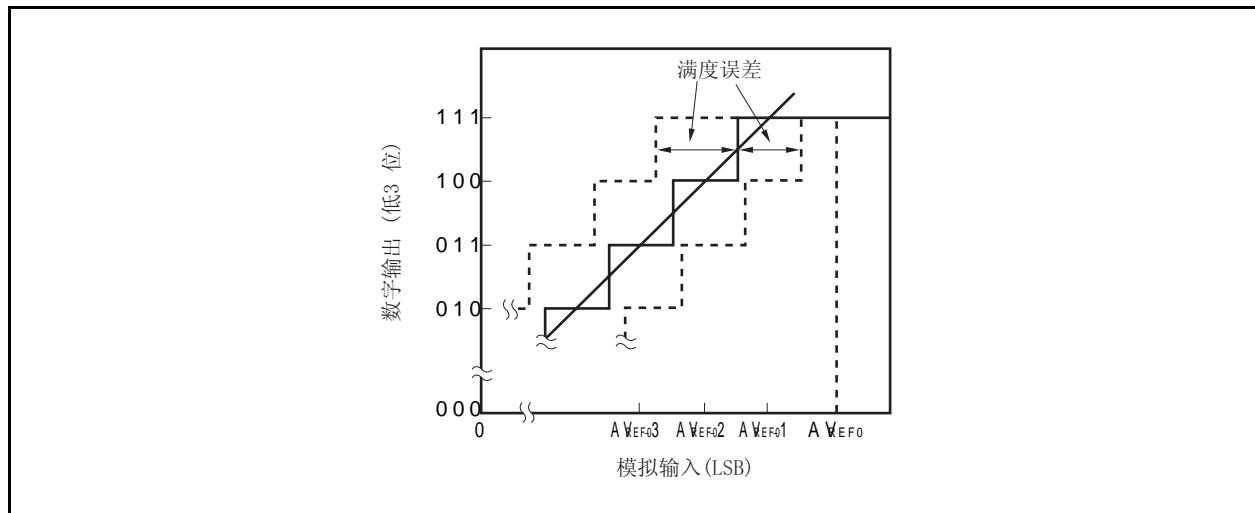
图 13-18. 零度误差



(5) 满度误差

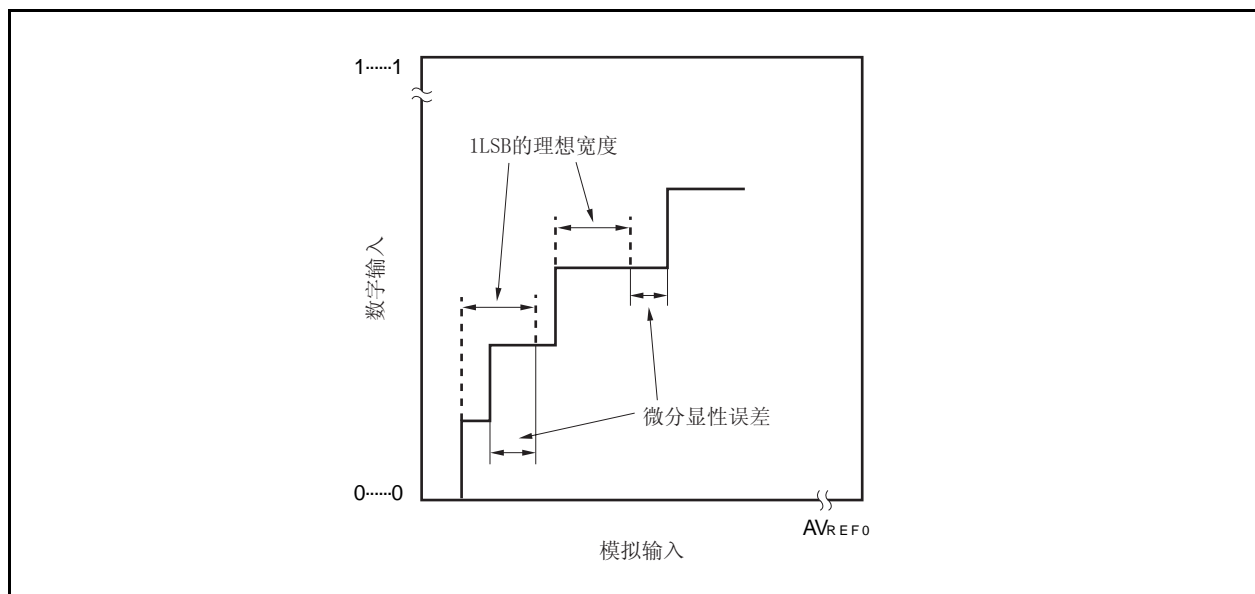
满度误差表示当数字输出范围在 $1\dots\dots110 \sim 1\dots\dots111$ 之间时模拟输入电压的实际测量值与理论值(满度 - $3/2\text{LSB}$)之间的误差。

图 13-19. 满度误差

**(6) 微分线性误差**

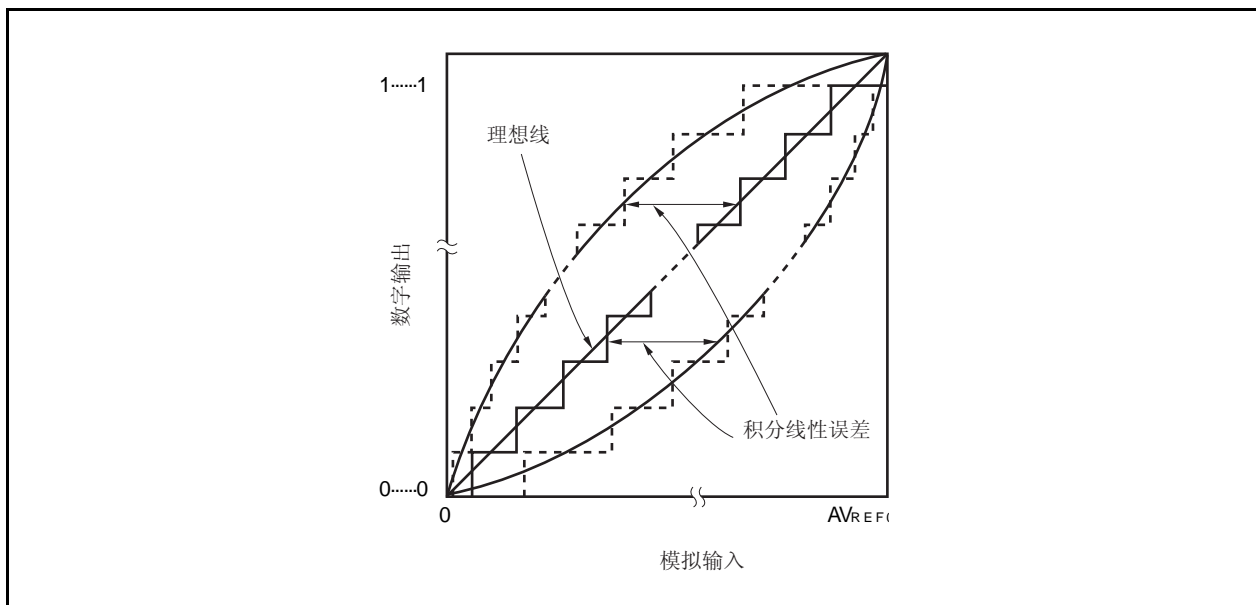
理论上，代码输出的宽度为 1LSB 。当输出一个特定码(specific code)，微分线性误差表示实际测量值与理想值之间的差距。当同一个通道的模拟输入引脚的电压一字节一字节地从 AV_{SS} 到 AV_{REF0} 持续增加，这显示了 A/D 转换的基本特征。当输入电压增加或减小，或两个(包括两个以上)被使用，参见 13.7 (2)总误差。

图 13-20. 微分线性误差



(7) 积分线性误差

积分线性误差说明了转换特征偏离理想线性关系的程度。它表示当零度误差和满度误差均为 0 时实际测量值与理想直线之间误差的最大值。

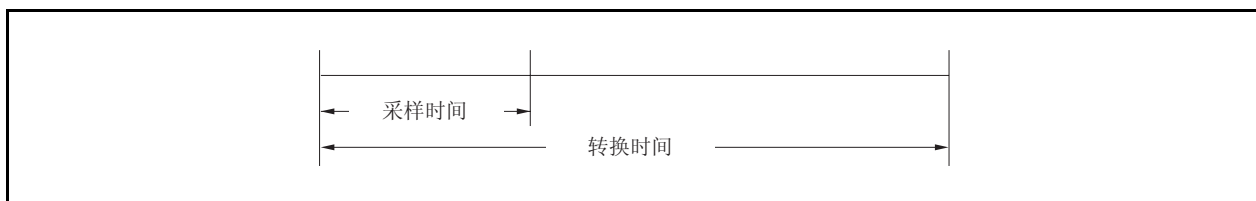
图 13-21. 积分线性误差**(8) 转换时间**

转换时间表示从产生触发到获取数字输出所经历的时间。

采样时间包含在特征表中的转换时间中。

(9) 采样时间

采样时间表示模拟电压的模拟开关被打开到模拟电压被采样&保持电路采样所需的时间。

图 13-22. 采样时间

第十四章 D/A 转换器

14.1 概述

D/A 转换器有以下特点。

D/A 转换器有以下特点。

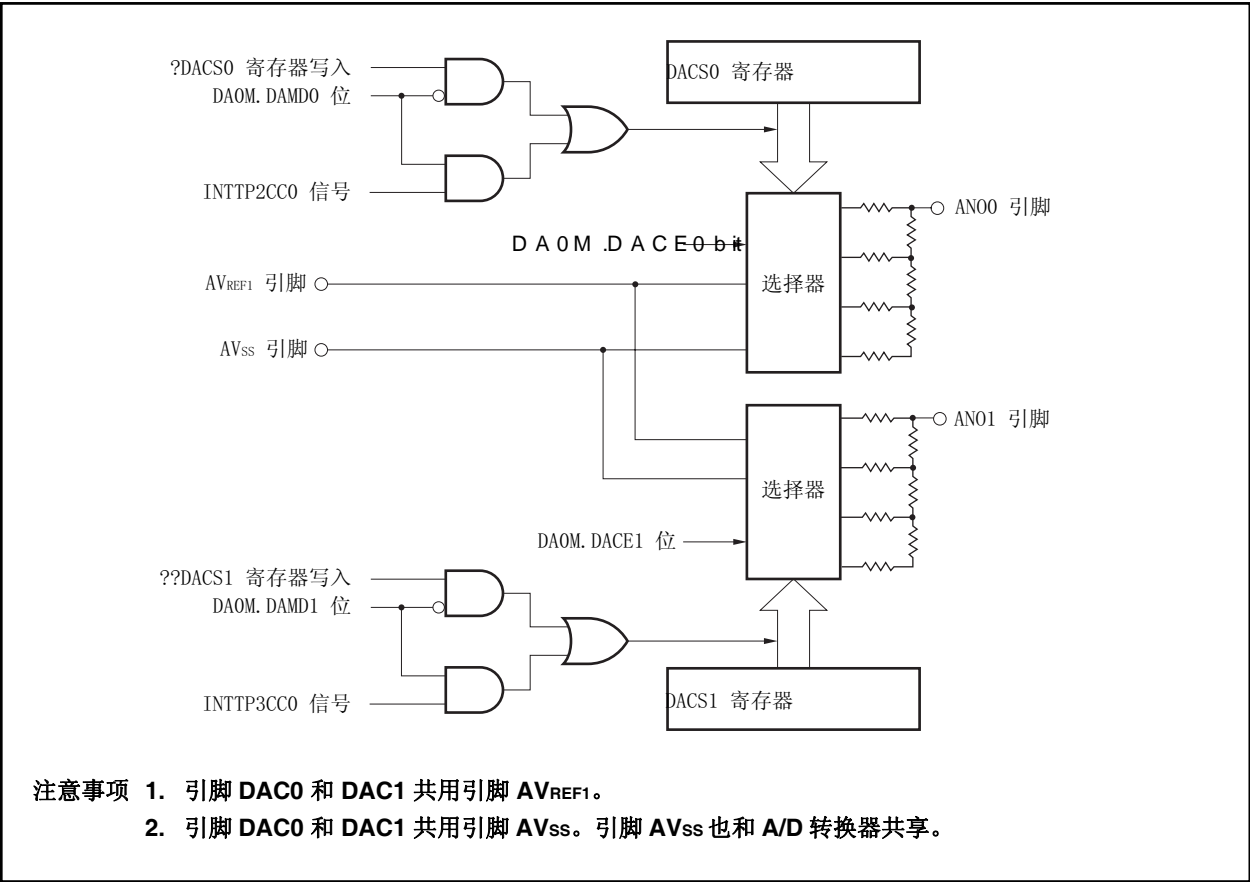
- 8 位分辨率 × 2 通道 (DA0CS0, DA0CS1)
- R-2R 梯形法
- ★ ○ 处理时间: 3 μs max. (当 AVREF1 为 3.0 ~ 3.6 V 且外部负载为 20 pF)
- 模拟输出电压: $AV_{REF1} \times m/256$ ($m = 0$ to 255; 由寄存器 DA0CSn 设置)
- 操作模式: 正常模式、实时输出模式

备注 n = 0, 1

14.2 配置

D/A 转换器的配置如下。

图 14-1. D/A 转换器框图



D/A 转换器包括以下硬件。

表 14-1. 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 转换器值设置寄存器 0, 1 (DA0CS0, DA0CS1)

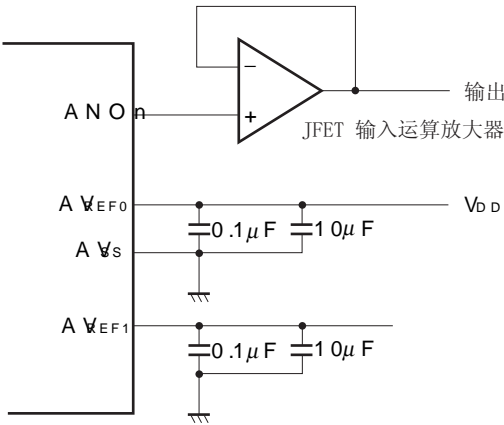
14.3 寄存器

控制 D/A 转换器的寄存器如下所述。

- D/A 转换器模式寄存器 (DA0M)
- D/A 转换器值设置寄存器 0, 1 (DA0CS0, DA0CS1)

(1) D/A 转换器模式寄存器 (DA0M)

该寄存器控制 D/A 转换器的运行。
可由 1 位或 8 位存储器操作指令读写该寄存器。
复位信号产生将该寄存器清零(00H)。



注 实时输出模式(DA0MDn = 1)下，输出触发如下所述。

- 当 n = 0: INTTP2CC0 信号 (参见 第 7 章 16 位定时器/事件计数器 P (TMP))
- 当 n = 1: INTTP3CC0 信号 (参见 第 7 章 16 位定时器/事件计数器 P (TMP))

(2) D/A 转换器值设置寄存器 0, 1 (DA0CS0, DA0CS1)

寄存器 DA0CS0 和 DA0CS1 设置引脚 ANO0 ~ ANO1 的模拟电压输出值。

可由 1 位或 8 位存储器操作指令读写该寄存器。

复位信号产生将该寄存器清零(00H)。

复位后:00H R/W 地址:DA0CS0 FFFFF280H, DA0CS1 FFFFF281H

	7	6	5	4	3	2	1	0
DA0CSn	DA0CSn7	DA0CSn6	DA0CSn5	DA0CSn4	DA0CSn3	DA0CSn2	DA0CSn1	DA0CSn0

注意事项 实时输出模式(DA0M.DA0MDn = 1)下，在信号 INTTP2CC0/INTTP3CC0 产生之前，设置寄存器 DA0CSn。当产生信号 INTTP2CC0/INTTP3CC0，开始 D/A 转换。

备注 n = 0, 1

14.4 操作

14.4.1 正常模式下的操作

对寄存器 DA0CSn 进行写操作触发 D/A 转换。
设置方法如下所述。

- <1> 设置 DA0M.DA0MDn = 0(正常模式)。
- <2> 设置寄存器 DA0CSn 的模拟电压值，该值将输出到引脚 ANOn。
步骤<1> and <2> 组成初始设置。
- <3> 设置 DA0M.DA0CEn = 1(D/A 转换使能)。
设置完成后，开始 D/A 转换。
- <4> 对寄存器 DA0CSn 进行写操作，执行下一次 D/A 转换。
上次的 D/A 转换结果保存到下次 D/A 转换开始。

备注

- 1. 关于引脚复用功能设置，参见表 4-15 使用端口引脚复用功能。
- 2. n = 0, 1

14.4.2 实时输出模式下的操作

TMP2 和 TMP3 的中断请求信号(INTTP2CC0 和 INTTP3CC0) 触发 D/A 转换。
设置方法如下所述。

- <1> 设置 DA0M.DA0MDn = 1(实时输出模式)。
- <2> 设置寄存器 DA0CSn 的模拟电压值，该值将输出到引脚 ANOn。
- <3> 设置 DA0M.DA0CEn = 1(D/A 转换使能)。
步骤<1> and <3> 组成初始设置。
- <4> 操作 TMP2 和 TMP3。
- <5> 当产生信号 INTTP2CC0 和 INTTP3CC0，D/A 转换开始。
- <6> 之后，每次产生信号 INTTP2CC0 和 INTTP3CC0，寄存器 DA0CSn 的值被输出。

备注

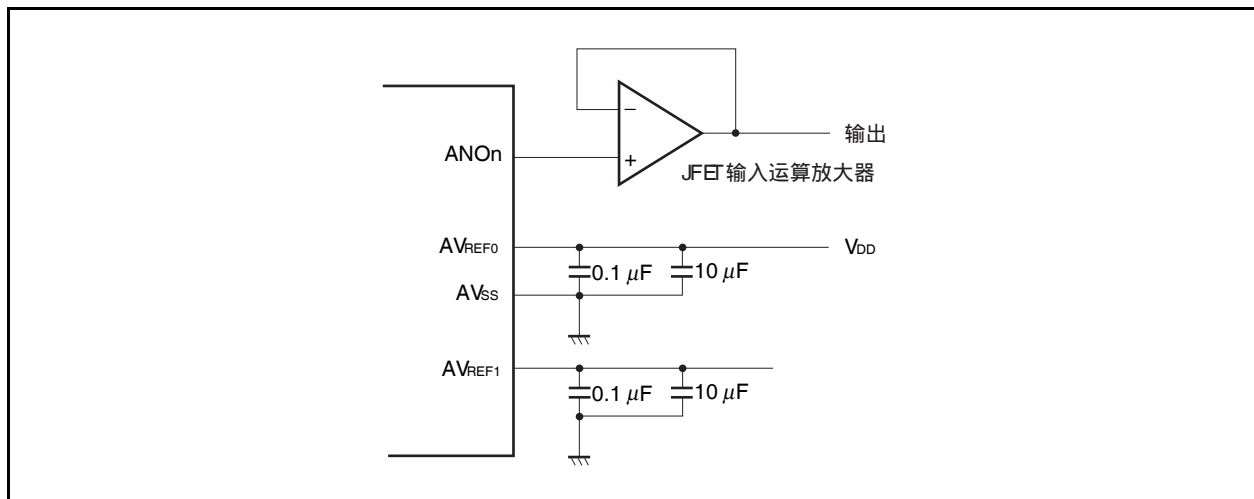
- 1. 引脚 ANOn 和 ANO1 的输出值在步骤<5>之前不确定。
- 2. HALT, IDLE1, IDLE2, 和 STOP 模式下的引脚 ANOn 和 ANO1 的输出值，可参见第 24 章 待机功能。
- 3. 关于引脚复用功能设置，参见表 4-15 使用端口引脚复用功能。

14.4.3 注意事项

当使用 V850ES/JG2 的 D/A 转换器时，注意下述注意事项。

- (1) 实时输出模式下，当响应触发信号后，不要改变寄存器 DA0CSn 的值。
- (2) 在改变操作模式前，确保 DA0M.DA0CEn = 0。
- ★ (3) 当引脚 P10/AN00 和 P11/AN01 其中之一被用于 I/O 端口，而另一个用于 D/A 输出引脚，在 D/A 输出期间不要改变 I/O 端口电平。
- (4) 确保 $AV_{REF0} = V_{DD} = AV_{REF1} = 3.0 \text{ to } 3.6 \text{ V}$ 。如果超出此范围，操作不能保证。
- (5) AV_{REF1} 的电压与 AV_{REF0} 的电压一致。
- (6) 没有电流可以从引脚 ANOn ($n = 0, 1$) 流出，因为 D/A 转换器的输出阻抗很高。当连接一个小于等于 $2 \text{ M}\Omega$ 的电阻时，在电阻和引脚 ANOn 之间插入一个 JFET 输入运算放大器。

图 14-2. 引脚连接示例



- (7) 由于 STOP 模式下，D/A 转换器停止操作，引脚 ANO0 和 ANO1 引入高阻抗状态，功耗可降低。IDLE1, IDLE2, 或子时钟模式下，操作继续。因此，为了降低功耗使 DA0M.DA0CEn = 0。

第十五章 异步串行接口 A (UARTA)

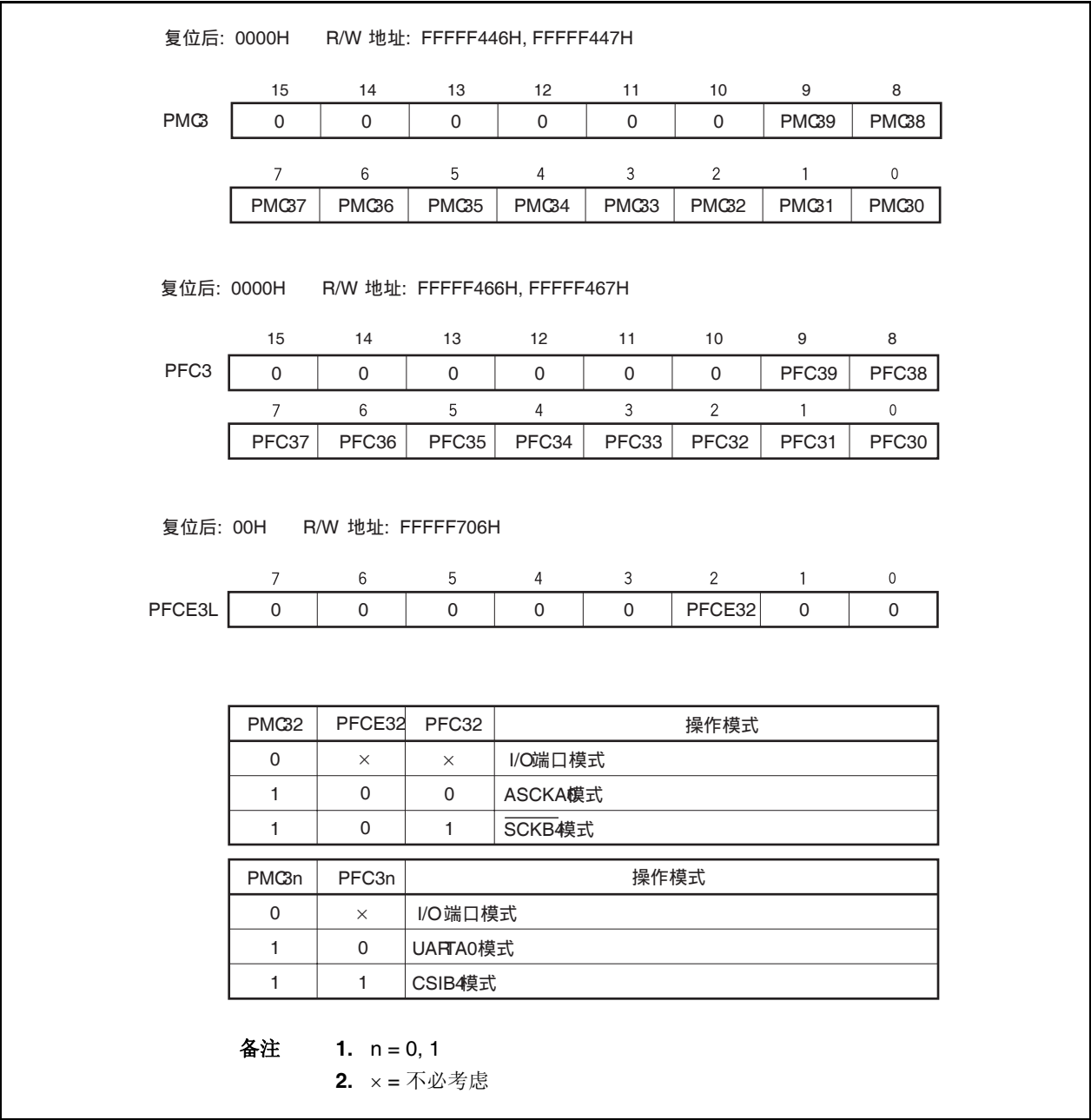
15.1 UARTA 和其它串行接口的模式转换

15.1.1 CSIB4 和 UARTA0 模式转换

在 V850ES/JG2 的 I²C 总线版本(Y 版本)中, CSIB4 和 UARTA0 复用相同引脚, 因此不能同时使用。在使用前, 先用寄存器 PMC3 和 PFC3 设置 UARTA0。

注意事项 如果 CSIB4 和 UARTA0 的发送/接收操作过程中, 进行转换, 这两个功能不能保证。确保不被使用的那个处于非使能态。

图 15-1. CSIB4 和 UARTA0 的模式转换设置

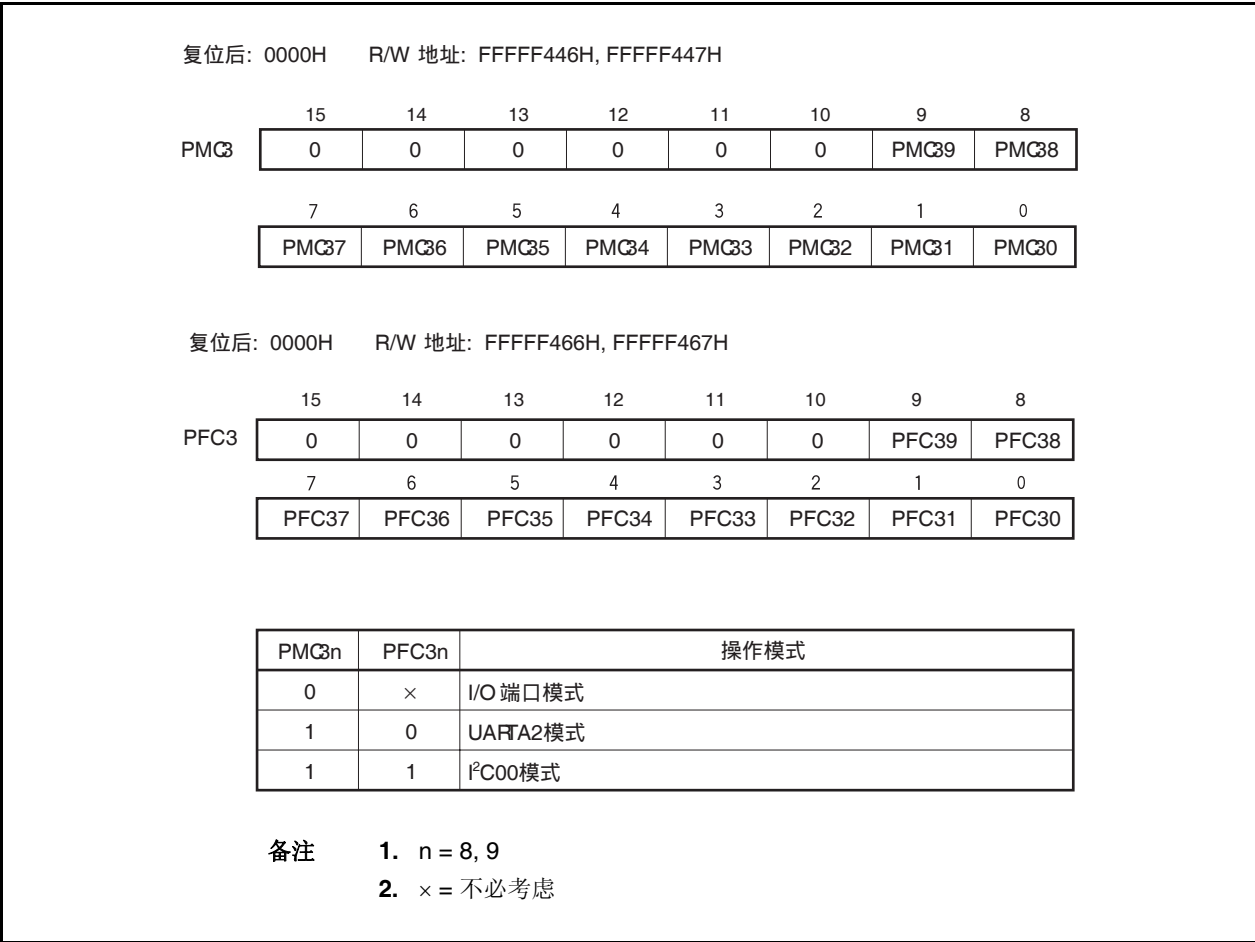


15.1.2 UARTA2 和 I²C00 模式转换

在 V850ES/JG2 的 I²C 总线版本(Y 版本)中, UARTA2 和 I²C00 复用相同引脚, 因此不能同时使用。在使用前, 先用寄存器 PMC3 和 PFC3 设置 UARTA2。

注意事项 如果 UARTA2 和 I²C00 的发送/接收操作过程中, 进行转换, 这两个功能不能保证。确保不被使用的那个处于非使能态。

图 15-2. UARTA2 和 I²C00 的模式转换设置

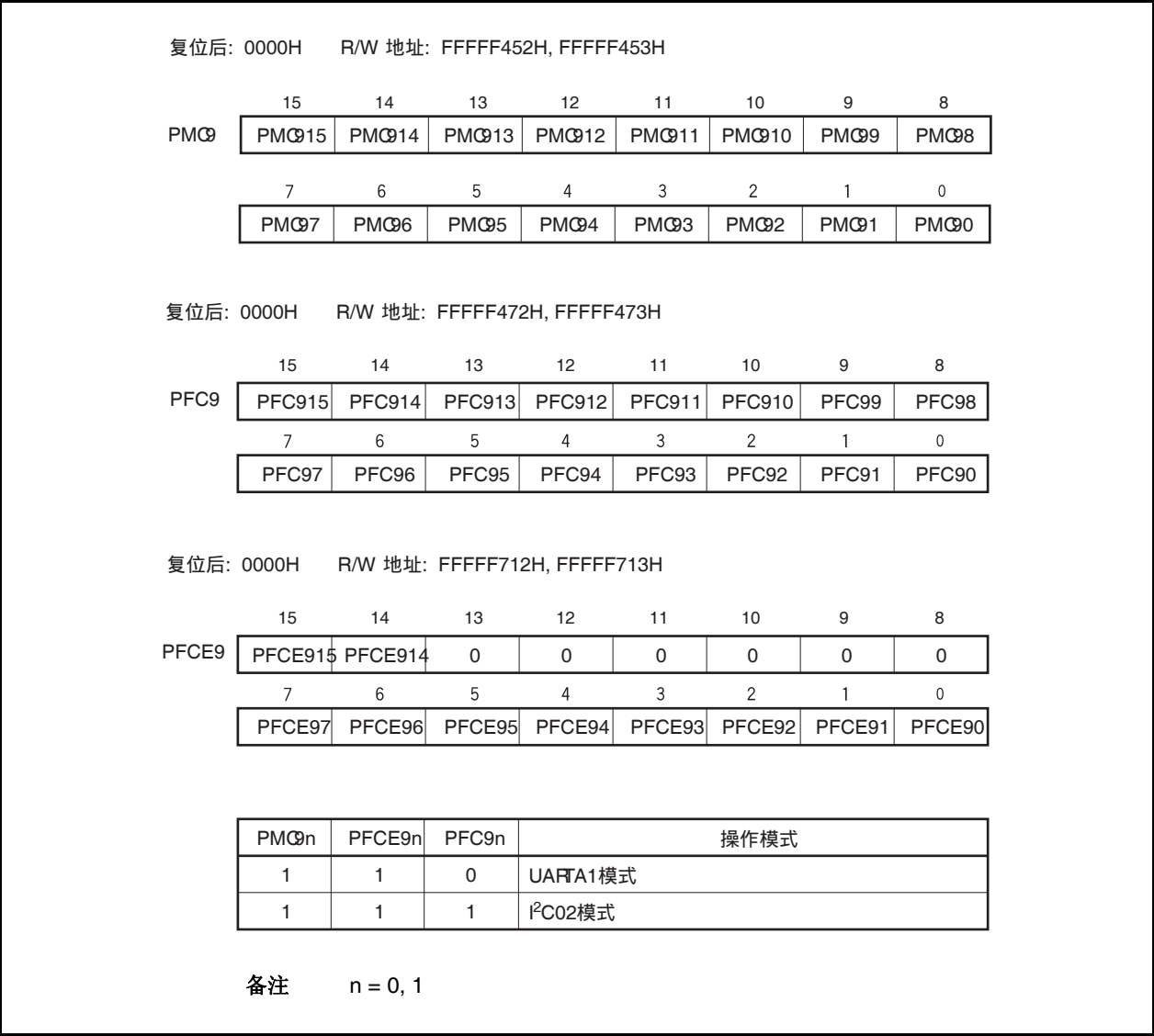


15.1.3 UARTA1 和 I²C02 模式转换

在 V850ES/JG2 的 I²C 总线版本(Y 版本)中, UARTA1 和 I²C02 复用相同引脚, 因此不能同时使用。在使用前, 先设置 UARTA1 的寄存器 PMC9, PFC9 和 PFC E9。

注意事项 如果 UARTA1 和 I²C02 的发送/接收操作过程中, 进行转换, 这两个功能不能保证。确保不被使用的那个处于非使能态。

图 15-3. UARTA1 和 I²C02 模式转换设置



15.2 特性

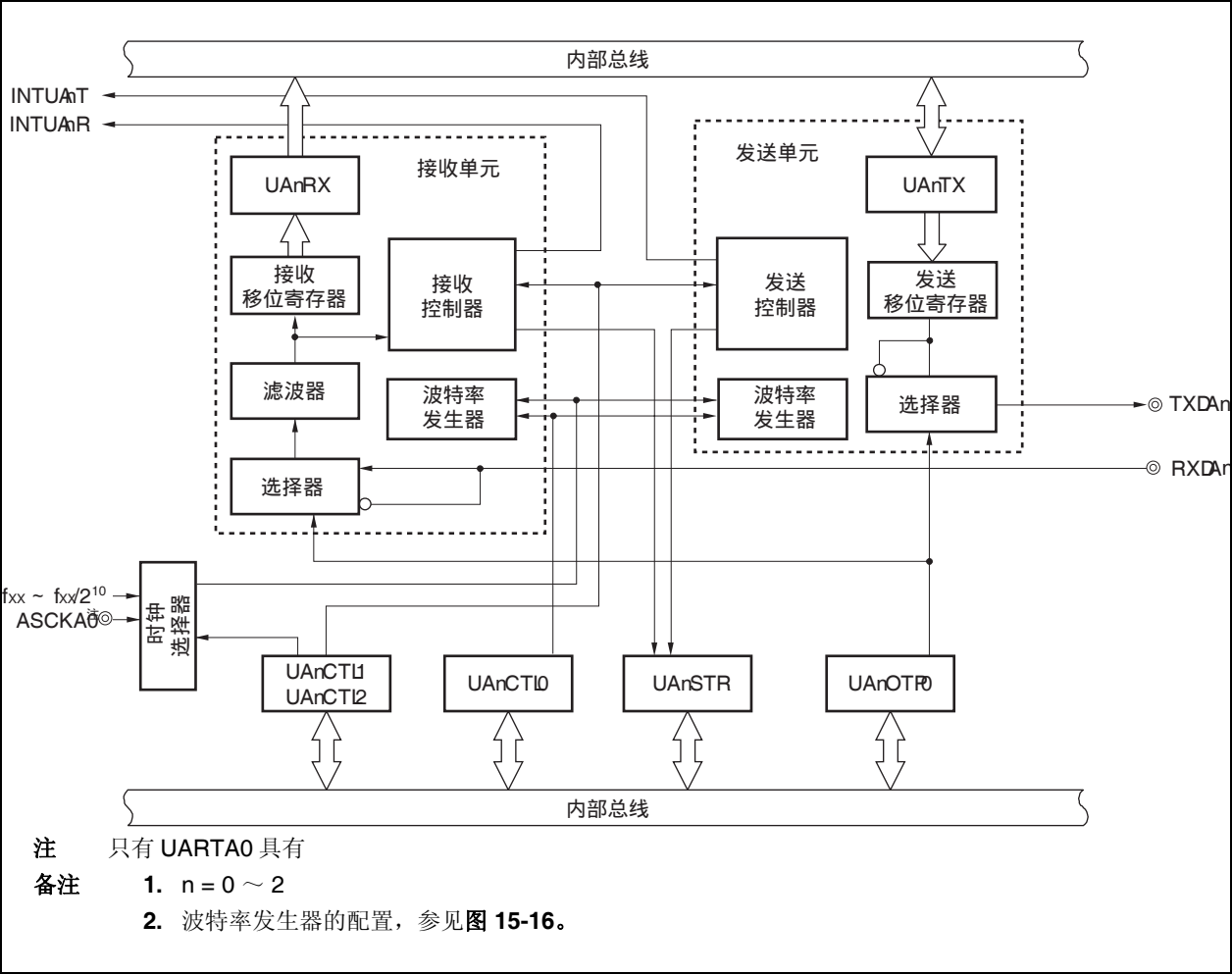
- 传输速率: 300 bps ~ 312.5 kbps(使用 20 MHz 内部系统时钟和专用波特率发生器)
- 全双工通信:
 - 内部 UARTAn 数据接收寄存器(UAnRX)
 - 内部 UARTAn 数据发送寄存器(UAnTX)
- 2 引脚设置:
 - TXDAn: 数据发送输出引脚
 - RXDAn: 数据接收输入引脚
- 接收错误输出功能
 - 校验错误
 - 帧错误
 - 溢出错误
- 中断源: 2
 - 接收完成中断 (INTUAnR):
 - 该中断产生于接收使能状态下，接收到的数据从接收移位寄存器到数据接收寄存器转换完成后。
 - 发送使能中断 (INTUAnT):
 - 该中断产生于发送使能状态下，发送数据从数据发送寄存器到发送移位寄存器的转换过程中。
- 通信数据的宽度: 7, 8 位
- 校验功能: 奇校验, 偶校验, 零校验, 不输出校验位
- 传输停止位: 1, 2 位
- 片上专用波特率发生器
- 可选 MSB-/LSB-first 通信
- 发送/接收数据反向输入/输出操作
- LIN (局部互联网)通信模式下 SBF (同步中断区域)的发送/接收
 - 同步中断区域发送长度为 13 ~ 20
 - 识别 11 位或更多位的 SBF 接收
 - 提供 SBF 接收标志

备注 n = 0 ~ 2

15.3 配置

如下为 UARTAn 的框图。

图 15-4. 异步串行接口 An 的框图



UARTAn 由以下寄存器控制。

表 15-1. UARTAn 的配置

项目	配置
寄存器	UARTAn 控制寄存器 0 (UAnCTL0) UARTAn 控制寄存器 1 (UAnCTL1) UARTAn 控制寄存器 2 (UAnCTL2) UARTAn 选择控制寄存器 0 (UAnOPT0) UARTAn 状态寄存器 (UAnSTR) UARTAn 接收移位寄存器 UARTAn 数据接收寄存器 (UAnRX) UARTAn 发送移位寄存器 UARTAn 数据发送寄存器 (UAnTX)

(1) UARTAn 控制寄存器 0 (UAnCTL0)

该寄存器是 8 位寄存器，用于设定 UARTAn 的操作。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

该寄存器是 8 位寄存器，用于选择 UARTAn 的输入时钟。

(3) UARTAn 控制寄存器 2 (UAnCTL2)

该寄存器是 8 位寄存器，用于控制 UARTAn 的波特率。

(4) UARTAn 选择控制寄存器 0 (UAnOPT0)

该寄存器是 8 位寄存器，用于控制 UARTAn 的串行传输。

(5) UARTAn 状态寄存器(UAnSTR)

该寄存器包含发生接收错误时的错误标志。当发生接收错误时，相应的接收错误标志置 1；当读取寄存器 UAnSTR 时，相应的接收错误标志置 0。

(6) UARTAn 接收移位寄存器

该移位寄存器用于将输入到引脚 RXDAn 的串行数据转换为并行数据。当接收到 1 字节的数据并检测到停止位，接收数据被传送到寄存器 UAnRX。

该寄存器不能直接操作。

(7) UARTAn 数据接收寄存器(UAnRX)

该寄存器是 8 位寄存器，用于保持接收数据。当接收到 7 个位，0 被存储于最高位(当数据以 LSB-first 模式传输)。

在接收使能状态，接收数据以同步方式在每帧移位过程中，从 UARTAn 的接收移位寄存器传送到寄存器 UAnRX。

传输到寄存器 UAnRX 同时引起产生接收完成中断请求信号(INTUAnR)。

(8) UARTAn 发送移位寄存器

该寄存器是移位寄存器，用于将寄存器 UAnTX 传出的并行数据转换成串行数据。

当 1 字节数据从寄存器 UAnTX 传送，移位寄存器数据从引脚 TXDAn 输出。

该寄存器不能直接操作。

(9) UARTAn 数据发送寄存器(UAnTX)

该寄存器是 8 位数据传送缓冲。当要传送的数据写入寄存器 UAnTX，传送开始。当数据可以被写入寄存器 UAnTX(当 1 帧的数据从寄存器 UAnTX 传送到发送移位寄存器 UARTAn)，产生发送使能中断请求信号(INTUAnT)。

15.4 寄存器

(1) UARTAn 控制寄存器 0 (UAnCTL0)

该寄存器是 8 位寄存器，用于控制 UARTAn 的串行发送操作。
可由 1 位或 8 位存储器操作指令设置该寄存器。
复位信号产生将该寄存器设置为 10H。

(1/2)

复位后: 10H		R/W 地址:		UA0CTL0 FFFF800H, UA1CTL0 FFFF810H, UA2CTL0 FFFF820H					
		<7>	<6>	<5>	<4>	3	2	1	0
UAnCTL0		UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL
(n = 0 ~ 2)									
UAnPWR		UARTAn操作控制							
0		禁止 UARTAn操作 (UARTAn复位不同步)							
1		使能 UARTAn操作							
		由UAnPWR位控制UARTAn的操作。UAnPWR位清零 (当UAnOPT0.UATDL = 1时固定于低电平)后， 引脚 TXDAn输出固定在高电平。							
UAnTXE		发送操作使能							
0		禁止发送操作							
1		使能发送操作							
		要开始发送，设置UAnPWR位为1，然后设置 UAnTXE位为1。 要停止发送，UAnTXE位清零，然后 UAnPWR位清零。 要初始化发送单元，UAnTXE位清零，等待2个基本时钟周期， 然后设置UAnTXE位再次为1。否则，初始化可能不被执行(关于基本时钟， 参见15.7 (1) (a)基本时钟)。							
UAnRXE		接收操作使能							
0		禁止接收操作							
1		使能接收操作							
		要开始接收，设置UAnPWR位为1，然后设置 UAnRXE位为1。 要停止接收，UAnRXE位清零，然后 UAnPWR位清零。 要初始化接收单元，UAnRXE位清零，等待2个基本时钟周期， 然后设置UAnRXE位再次为1。否则，初始化可能不被执行(关于基本时钟， 参见15.7 (1) (a)基本时钟)。							

<R>

UAnDIR	发送方向选择
0	MSB-first发送
1	LSB-first发送
只有当UAnPWR位 = 0或UAnTXE位= UAnRX位 = 0时，该寄存器可被改写。	

UAnPS1	UAnPS0	发送期间校验选择	接收期间校验选择
0	0	不输出校验位	无校验接收
0	1	输出零校验	按零校验接收
1	0	输出奇校验	按奇校验判断
1	1	输出偶校验	按偶校验判断
<ul style="list-style-type: none">•只有当UAnPWR位 = 0或UAnTXE位= UAnRX位 = 0时，该寄存器可被改写。•如果接收时被设置为“按零校验接收”，不执行校验判断。因此，UAnSTR.UAPE位不被设置。•LIN格模式下，执行发送和接收，UAnPS1和 UAnPS0位清零。			

<R>

UAnCL	指定每帧发送/接收数据的字符宽度
0	7位
1	8位
只有当UAnPWR位 = 0或UAnTXE位= UAnRX位 = 0时，该寄存器可被改写。	

UAnSL	指定发送数据停止位的个数
0	1位
1	2位
只有当UAnPWR位 = 0或UAnTXE位= UAnRX位 = 0时，该寄存器可被改写。	

备注 详细情况，参见 15.6.9 校验类型和操作。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

详细情况，参见 15.7 (2) UARTAn 控制寄存器 1 (UAnCTL1)。

(3) UARTAn 控制寄存器 2 (UAnCTL2)

详细情况，参见 15.7 (3) UARTAn 控制寄存器 2 (UAnCTL2)。

(4) UARTAn 选择控制寄存器 0 (UAnOPT0)

该寄存器是 8 位寄存器，用于控制 UARTAn 的串行传输。
可由 1 位或 8 位存储器操作指令设置该寄存器。
复位信号产生将该寄存器设置为 14H。

(1/2)

复位后: 14H		R/W 地址:		UA0OPT0 FFFA03H, UA1OPT0 FFFA13H, UA2OPT0 FFFA23H						
		<7>		6	5	4	3	2	1	0
UAnOPT0		UAnSRF		UAnSRT	UAnSTT	UAnSLS2	UAnSLS1	UAnSLS0	UAnTDL	UAnRDL
(n = 0 ~ 2)										
		UAnSRF		SB接收标志						
		0		当UAnCTIO.UAnPWR = UAnCTIO.UAnRXE= 0, 或者SB接收正常结束。						
		1		SB接收操作正在进行中						
		<ul style="list-style-type: none">• SBF(同步中断阈)接收只在LIN通信时被判断。• 当出现SB接收错误时UAnSR位保持1，然后SB接收重新开始。								
		UAnSRT		SB接收触发						
		0								
		1		SB接收触发						
		<ul style="list-style-type: none">• 该位是LIN通信期间的SB接收触发位，读取值恒为0。• 要接收SBF, 设置UAnSRT 为 1)以使能SB接收。• 在设置UAnPWR位 = UAnRXE位 = 1之后，设置UAnSRT								
		UAnSTT		SB发送触发						
		0								
		1		SB发送触发						
		<ul style="list-style-type: none">• 该位是LIN通信期间的SB接收触发位，读取值恒为0。• 在设置UAnPWR位 = UAnTXE位 = 1之后，设置UAnSTT								

★ 注意事项 在 SBF 接收期间(UAnSRF = 1)，不要将 UAnSRT 和 UAnSTT 置 1。

UAnSLS2	UAnSLS1	UAnSLS0	SBF发送宽度控制
1	0	1	13位输出(复位后默认值)
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

该寄存器可在UAnPWR位= 0或在UAnTXE位 = 0进行设置。

UAnTDL	发送数据电平位
0	发送数据正常输出
1	发送数据翻转输出

- 引脚TXDAn输出电平可通过设置UAnTDL位翻转。
- 该寄存器可在UAnPWR位= 0或在UAnTXE位 = 0进行设置。

UAnRDL	接收数据电平位
0	发送数据正常输入
1	发送数据翻转输入

- 引脚RXDAn的输入电平可通过设置UAnRDL位翻转。
- 该寄存器可在UAnPWR位= 0或在UAnRXE位 = 0进行设置。

(5) UARTAn 状态寄存器(UAnSTR)

寄存器是 8 位寄存器，显示了 UARTAn 的传送状态和接收错误原因。

可由 1 位或 8 位存储器操作指令设置该寄存器。UAnTSF 位是只读的。UAnPE, UAnFE 和 UAnOVE 位可以进行读/写操作。只有写入 0 时，这些位才清零；写入 1 时，这些位不能置 1(即使写入 1，它们也保持原值)。

初始条件如下所示。

寄存器/位	初始条件
寄存器 UAnSTR	<ul style="list-style-type: none"> 复位 UAnCTL0.UAnPWR = 0
UAnTSF 位	<ul style="list-style-type: none"> UAnCTL0.UAnTXE = 0
UAnPE, UAnFE, UAnOVE 位	<ul style="list-style-type: none"> 写入 0 UAnCTL0.UAnRXE = 0

复位后: 00H R/W 地址: UA0STR FFFA04H, UA1STR FFFA08H,
 UA2STR FFFA0C4H

	<7>	6	5	4	3	<2>	<1>	<0>
UAnSTR	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

(n = 0 ~ 2)

UAnTSF	发送状态标志位
0	<ul style="list-style-type: none">当设置 UAnPWR = 0或UAnTXE= 0发送完成，没有数据从寄存器UAnTX发送
1	当数据被写入寄存器 UAnTX
连续发送模式下UAnTSF位恒为1。当初始化发送单元，在执行初始化前查看UAnTSF位= 0。当UAnTSF位= 1时执行初始化，不产生发送数据。	

UAnPE	校验错误标志位
0	<ul style="list-style-type: none">当设置UAnPWR位= 0 或 UAnRXE位= 0。当写入0。
1	当校验数据与校验位在接收端不匹配。
<ul style="list-style-type: none">UAnPE位的操作由设置 UAnCTL0.UAnPS 和UAnCTI0.UAnPS位控制。UAnPE位可读写，但只能通过写0操作清零，不能通过写1操作进行设置。当向该位写1，数值不变。	

UAnFE	帧错误标志位
0	<ul style="list-style-type: none">当设置UAnPWR位= 0 或 UAnRXE位= 0。当写入0。
1	当接收端未检测到停止位。
<ul style="list-style-type: none">无论UAnCTI0.UAnSL位如何设置，只检测接收数据的第一个停止位。UAnFE位可读写，但只能通过写0操作清零，不能通过写1操作进行设置。当向该位写1，数值不变。	

UAnOVE	溢出错误标志位
0	<ul style="list-style-type: none">当设置UAnPWR位= 0 或 UAnRXE位= 0。当写入0。
1	当接收到的数据被写入寄存器，在该数据被读取前，下一个接收操作完成
<ul style="list-style-type: none">当产生溢出错误，数据被丢弃，下一个接收到的数据不被写入接收缓存。UAnOVE位可读写，但只能通过写0操作清零，不能通过写1操作进行设置。当向该位写1，数值不变。	

(6) UARTn 数据接收寄存器(UAnRX)

该寄存器是 8 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。

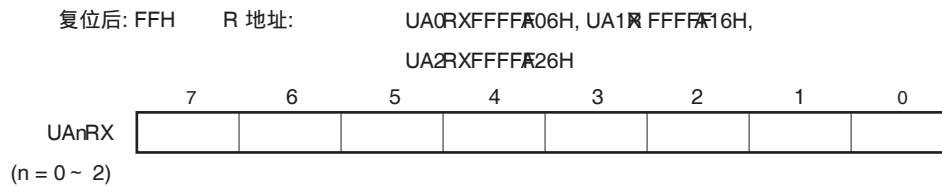
完整接收 1 字节数据后，存储于接收移位寄存器的数据被传送到寄存器 UAnRX。

LSB-first 接收期间，当数据长度设置为 7 位，接收到的数据置于寄存器 UAnRX 的第 0~6 位，第 7 位恒为 0。MSB-first 接收期间，接收到的数据置于寄存器 UAnRX 的第 1~7 位，第 0 位恒为 0。

当产生溢出错误(UAnOVE)，此时接收到的数据不被传送到寄存器 UAnRX 并丢弃。

该寄存器是只读的，可由 8 位存储器操作指令设置该寄存器。

除了复位输入，通过 UAnCTL0.UAnPWR 置 0，寄存器 UAnRX 可置 FFH。

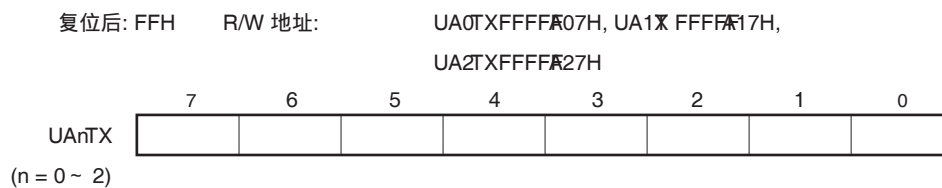


(7) UARTn 数据发送寄存器(UAnTX)

该寄存器是 8 位寄存器，用于设置发送数据。

可由 8 位存储器操作指令读/写该寄存器。

复位信号产生将该寄存器设置为 FFH。



15.5 中断请求信号

由 UARTAn 产生下面两个中断请求信号。

- 接收完成中断请求信号 (INTUAnR)
- 发送使能中断请求信号 (INTUAnT)

这两个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级高，发送使能中断请求信号优先级低。

表 15-2. 中断和其默认优先级

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 (INTUAnR)

接收使能状态下，当数据转移到接收移位寄存器并发送到寄存器 UAnRX 后，输出接收完成中断请求信号。
当接收完成中断请求信号被响应，并且数据被读取，读取寄存器 UAnSTR，判断是否产生接收错误。
接收非使能状态下，不产生接收完成中断请求信号。

(2) 发送使能中断请求信号 (INTUAnT)

发送使能时，如果被发送的数据从寄存器 UAnTX 发送到发送移位寄存器 UARTAn，产生发送使能中断请求信号。

15.6 操作

15.6.1 数据格式

全双工串行数据接收和发送可分别执行。

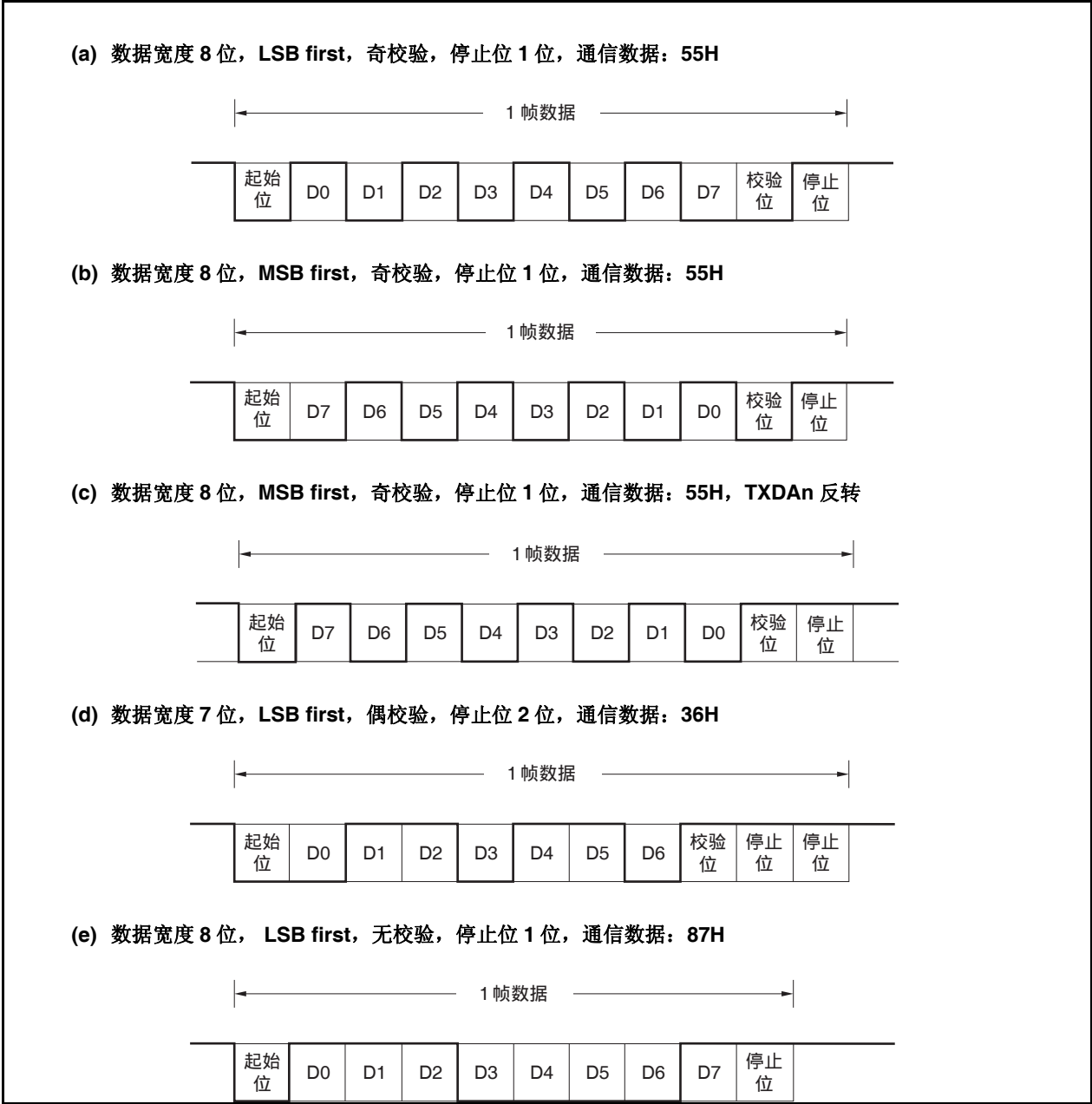
如图 15-5 所示，发送/接收数据的数据帧由起始位，字符位，校验位和停止位组成。

使用寄存器 UAnCTL0 设置每个数据帧的字符位的长度，校验位选择，停止位长度和 MSB/LSB-first 传送。

此外，由 UAnOPT0.UAnTDL 设置 UART 的输出控制和 TXDAn 位的插入输出。

- 起始位 1 位
- 字符位 7 / 8 位
- 校验位 奇校验/偶校验/零校验/无校验
- 停止位 1/2 位

图 15-5. UARTA 发送/接收数据帧格式



15.6.2 SBF 发送/接收格式

V850ES/JG2 具有 SBF(同步中断区域)发送/接收控制功能，用于使能 LIN 功能。

备注 LIN 表示局部互联网，是一个低速(1 ~ 20 kbps)串行通信协议，可用于降低网络费用。
LIN 是一种单主设备(single-master)通信方式，一个主设备上最多可以连接 15 个从设备。
LIN 的从设备用于控制开关、制动器和传感器，这些设备通过 LIN 网与 LIN 主设备相连。
通常，LIN 主设备与一个网络，如 CAN(控制器局域网 Controller Area Network)相连。
此外，LIN 总线采用单线方式，通过收发器(符合 ISO9141)与各节点相连。
在 LIN 协议中，主设备发送带有波特率信息的一帧数据，从设备接收数据并校正波特率误差。因此当从设备端的波特率误差在±15%范围内时，可以进行通信。

图 15-6 和 15-7 概括了 LIN 的发送和接收操作。

图 15-6. LIN 发送操作

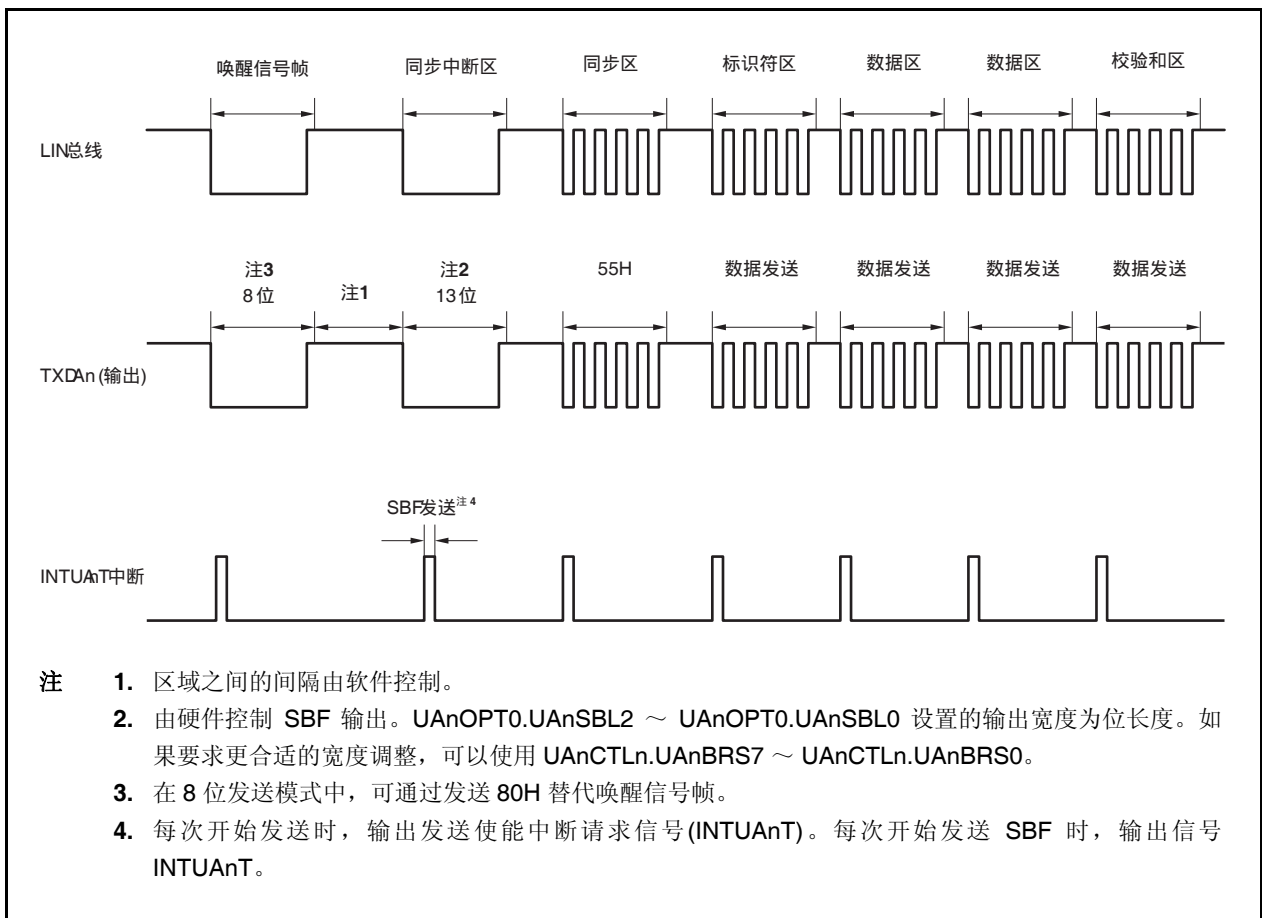
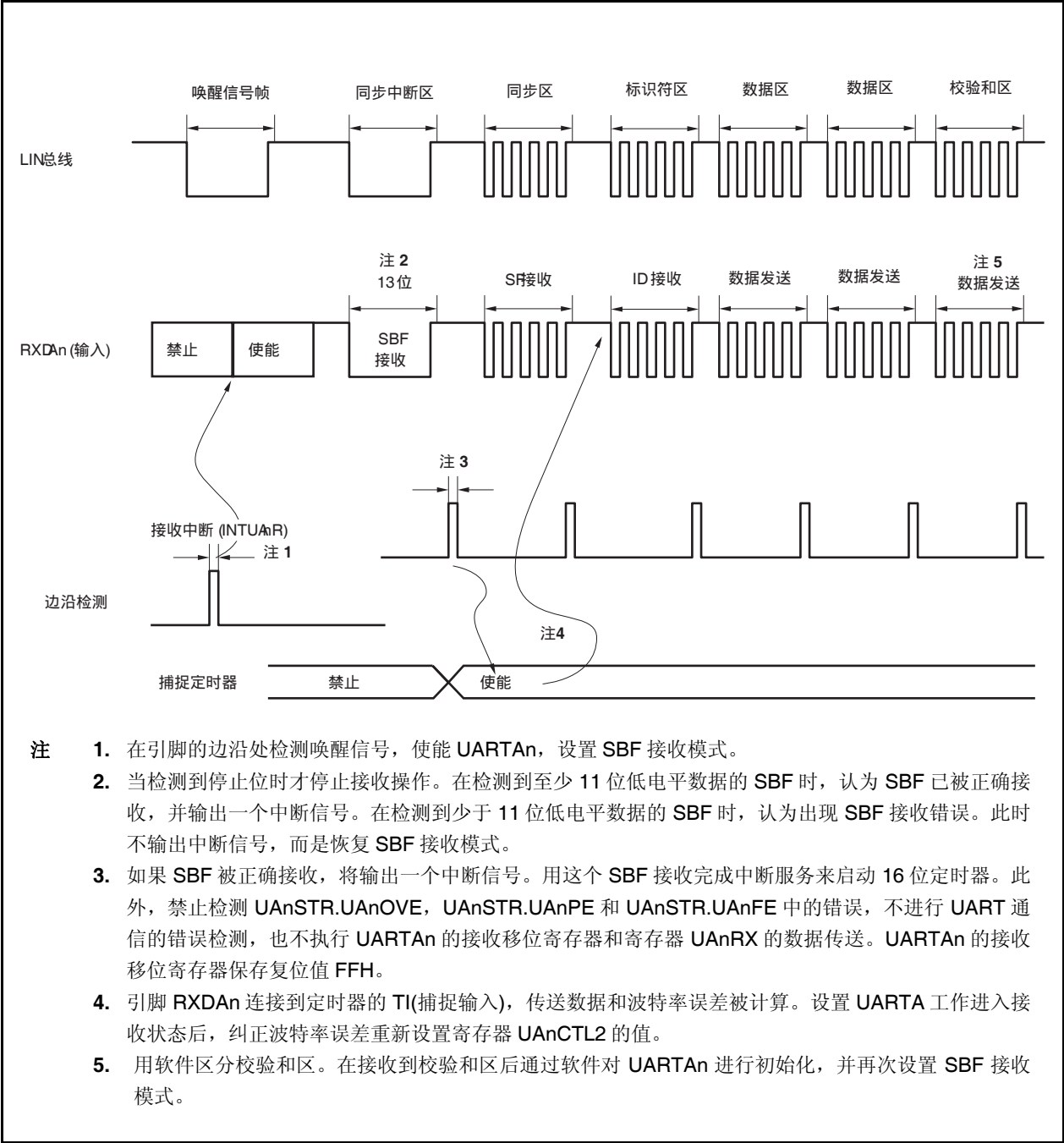


图 15-7. LIN 接收操作



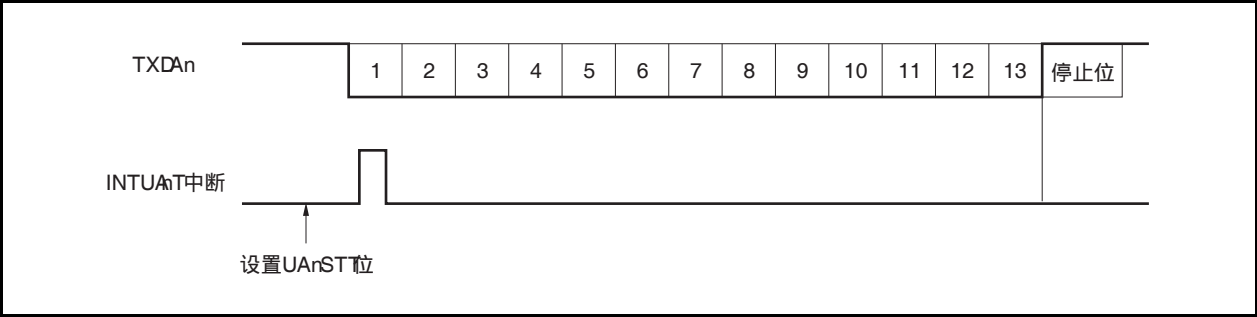
15.6.3 SBF 发送

当 $UAnCTL0.UAnPWR = UAnCTL0.UAnTXE = 1$ ，进入发送使能状态，SBF 发送触发($UAnOPT0.UAnSTT$ 位)置 1 后，SBF 发送开始。

之后，输出 13 ~ 20 个低电平位(由 $UAnOPT0.UAnSLS2 \sim UAnOPT0.UAnSLS0$ 位设置)。SBF 开始发送时，产生发送使能中断请求信号($INTUAnT$)。SBF 发送完之后， $UAnSTT$ 位自动清零。之后，恢复 UART 发送模式。

直到下一个被发送的数据写入寄存器 $UAnTX$ ，或直到设置发送触发($UAnSTT$)为 1，才停止发送。

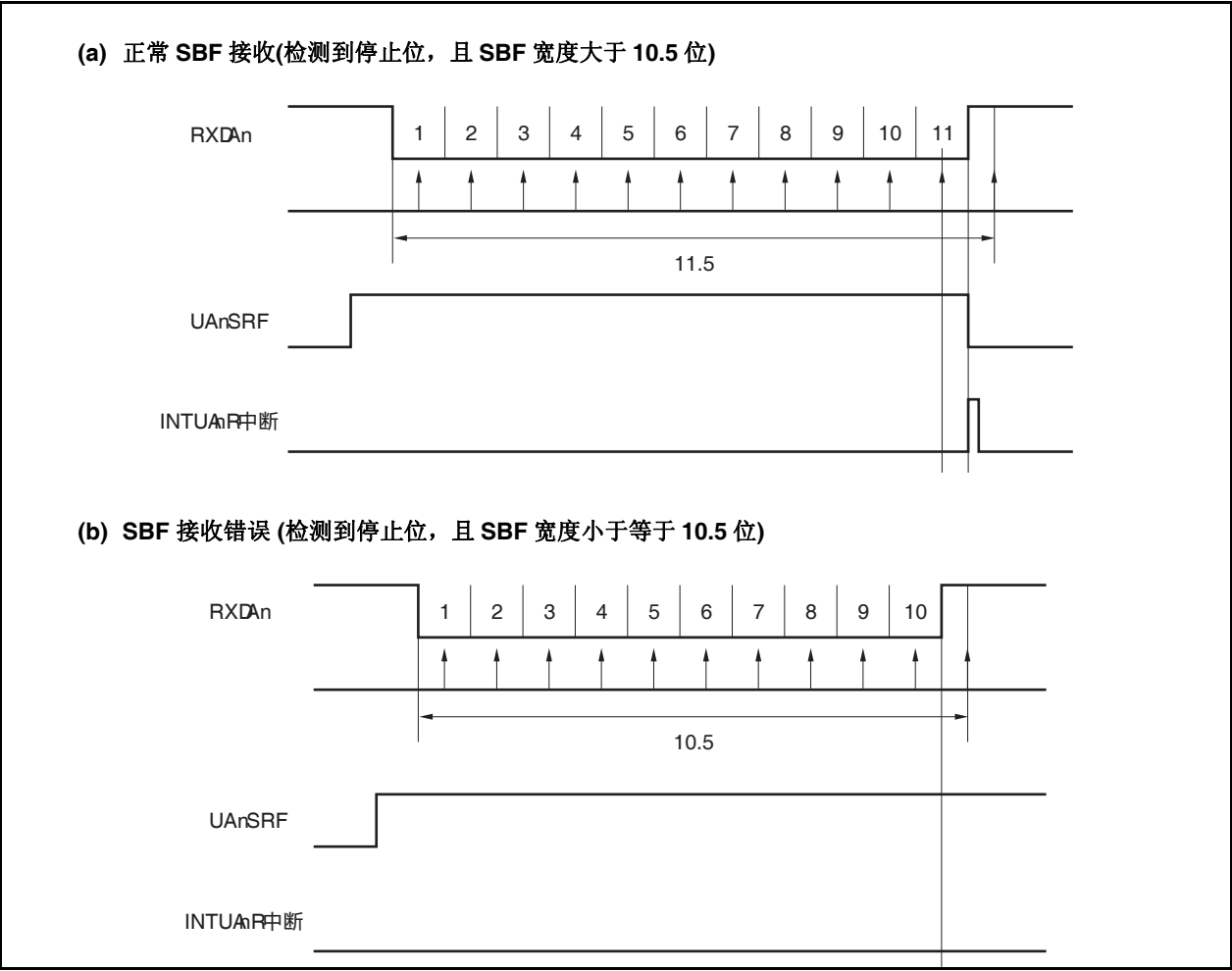
图 15-8. SBF 发送



15.6.4 SBF 接收

当设置 UAnCTL0.UAnPWR = 1，并设置 UAnCTL0.UAnRXE = 1，进入接收使能状态。
由 SBF 接收触发(UAnOPT0.UAnSTR = 1)设置 SBF 接收等待状态。
SBF 接收等待状态下，正如 UART 接收等待状态下一样，引脚 RXDAn 被监测，开始起始位检测。
检测到起始位后，开始接收，内部计数器根据设定的波特率计算开始计数。
当接收到停止位，如果 SBF 的宽度大于等于 11 位，属正常操作，输出接收完成中断请求信号(INTUAnR)。UAnOPT0.UAnSRF 位自动清零，SBF 接收结束。禁止检测 UAnSTR.UAnOVE，UAnSTR.UAnPE 和 UAnSTR.UAnFE 中的错误，不进行 UART 通信的错误检测，也不执行 UARTAn 的接收移位寄存器和寄存器 UAnRX 的数据传送。UARTAn 的接收移位寄存器保存复位值 FFH。如果 SBF 的宽度小于等于 10 位，接收终止，不产生中断，返回 SBF 接收模式。UAnSRF 位此时不清零。

图 15-9. SBF 接收



15.6.5 UART 发送

设置 UAnCTL0.UAnPWR = 1，使引脚 TXDAn 输出高电平。

之后，设置 UAnCTL0.UAnTXE = 1，进入发送使能状态。把要发送的数据写入寄存器 UAnTX，开始发送。起始位、校验位和停止位会自动被添加到数据中。

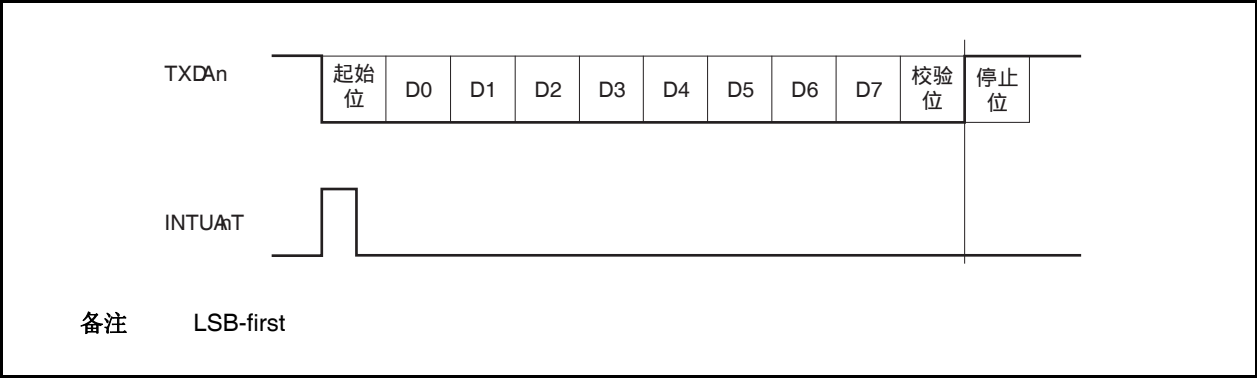
UARTAn 下没有提供 CTS(发送使能信号)输入引脚，所以使用某个端口检测发送终端的接收状态是否使能。

开始发送时，寄存器 UAnTX 中的数据被传送到发送移位寄存器 UARTAn。

完成寄存器 UAnTX 到发送移位寄存器 UARTAn 的数据的发送时，产生发送使能中断请求信号(INTUAnT)，然后，发送移位寄存器 UARTAn 的内容被输出到引脚 TXDAn。

产生信号 INTUAnT 后，可以写入下一个数据到寄存器 UAnTX。

图 15-10. UART 发送



15.6.6 连续发送

UARTAn 可以在 UARTAn 的发送移位寄存器开始移位操作时，将下一个发送数据写入寄存器 UAnTX。发送移位寄存器 UARTAn 的发送时序可通过发送使能中断请求信号(INTUAnT)判断。

通过发送过程中向寄存器 UAnTX 写入下一个被发送的数据，判断有效通信率。

注意事项 在连续发送过程中执行发送初始化，确保 UAnSTR.UAnTSF = 0，然后进行初始化。当 UAnTSF = 1 时初始化数据，其发送过程不能保证。

图 15-11. 连续发送操作流程

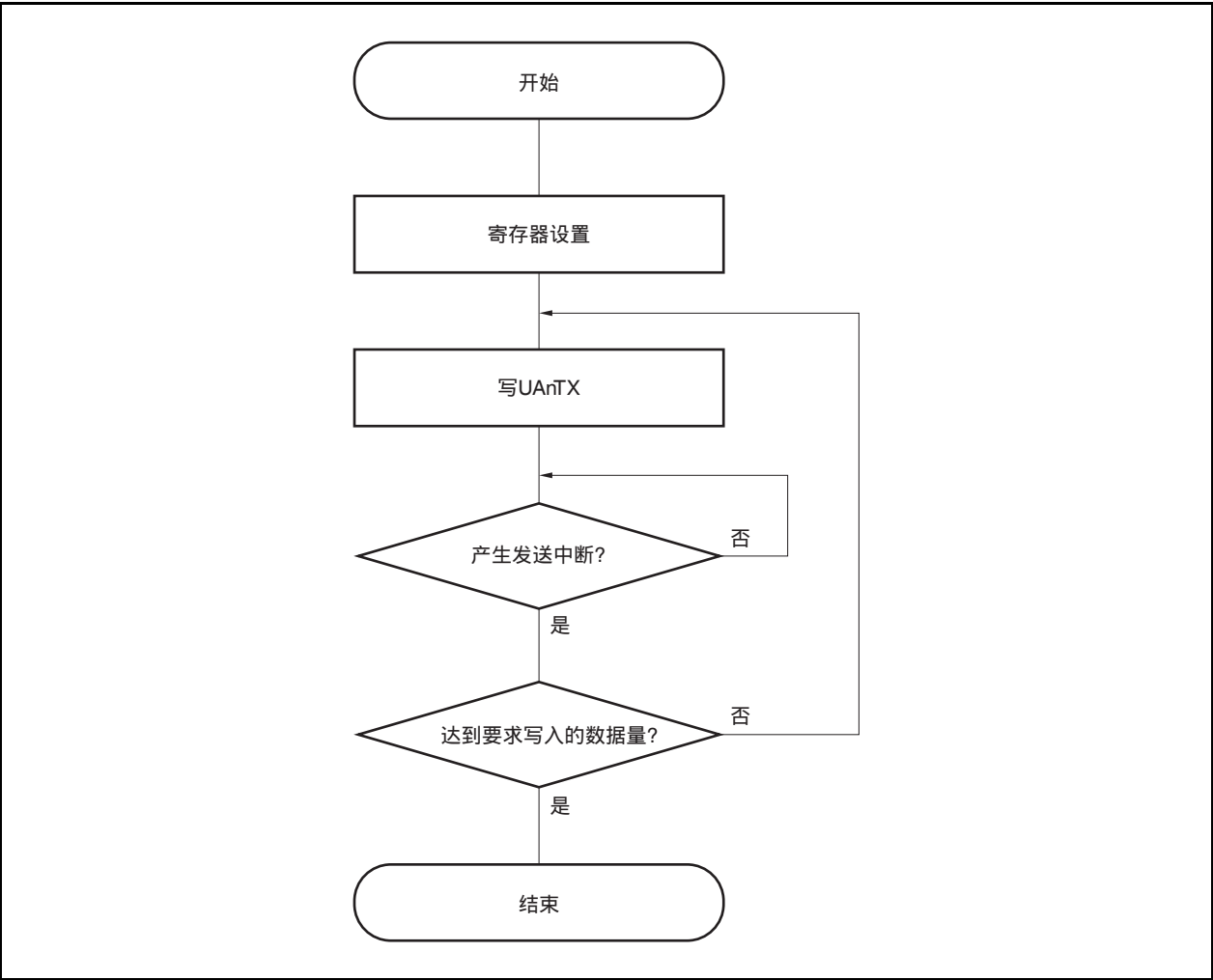
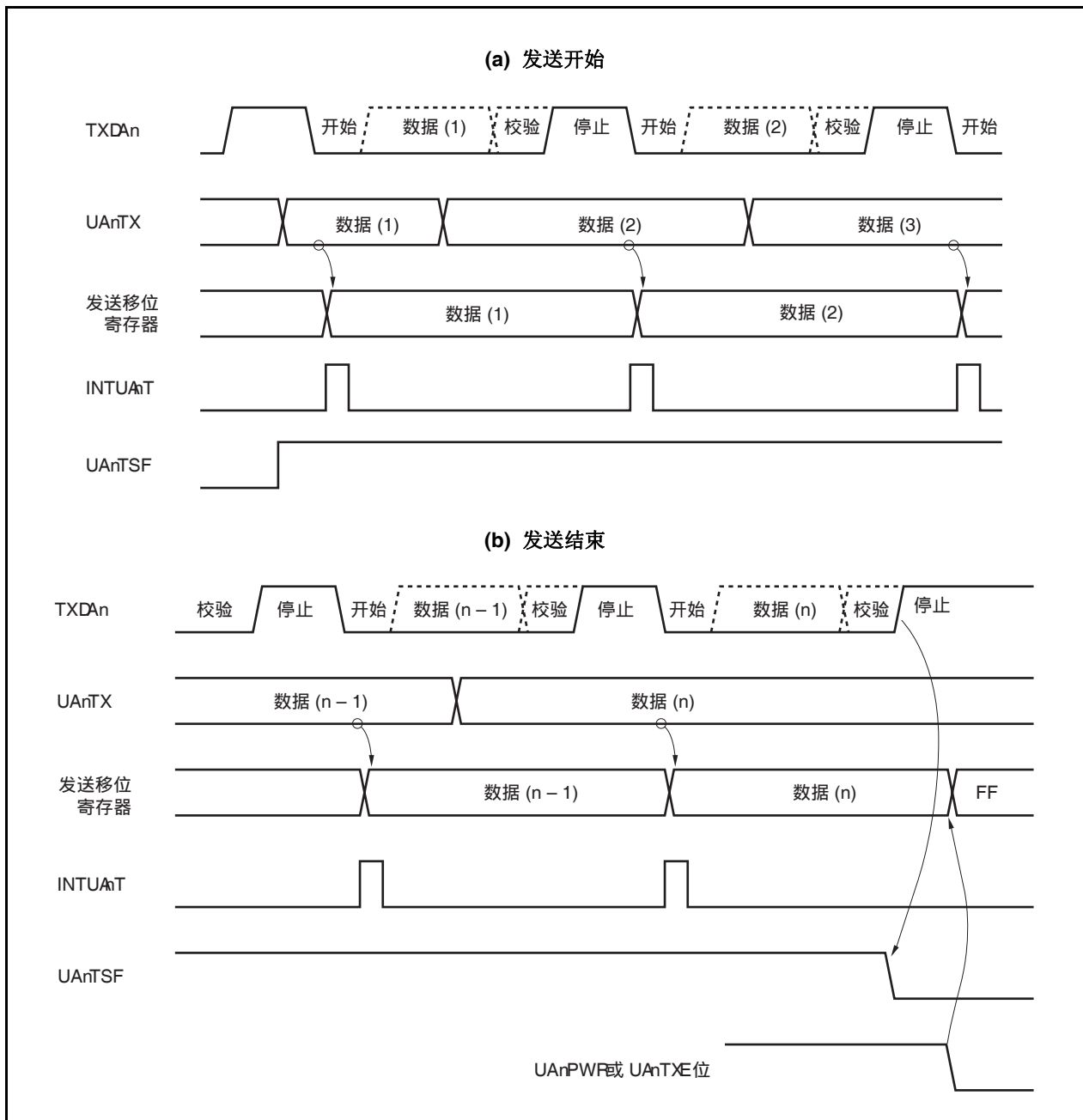


图 15-12. 连续发送操作时序



15.6.7 UART 接收

当设置 $UAnCTL0.UAnPWR = 1$ 且 $UAnCTL0.UAnRXE = 1$ ，进入接收等待状态。接收等待状态下，引脚 $RXDAn$ 被监控，检测起始位。

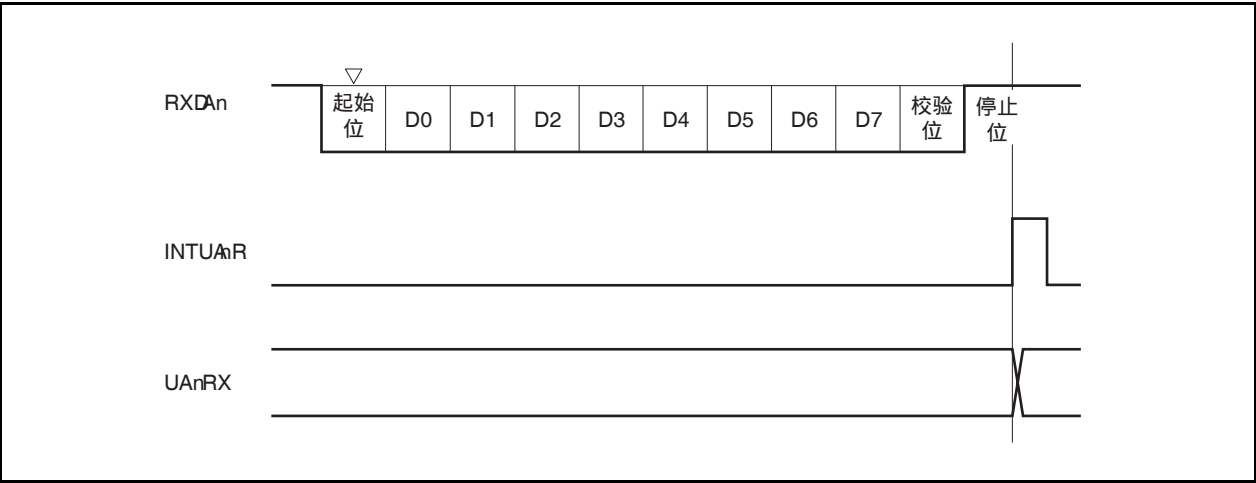
检测起始位使用 2 步检测子程。

首先，检测到引脚 $RXDAn$ 的上升沿，在下降沿开始采样。如果在起始位采样点，引脚 $RXDAn$ 为低电平，认为检测到起始位。检测到起始位后，接收操作开始，串行数据根据设定好的波特率保存到 $UARTAn$ 的接收移位寄存器。

当接收到停止位时，输出接收完成中断请求信号($INTUAnR$)， $UARTAn$ 的接收移位寄存器的数据写入寄存器 $UAnRX$ 。但是，如果产生溢出错误($UAnSTR.UAnOVE$)，此时接收到的数据不被写入寄存器 $UAnRX$ ，而是被丢弃。

即使在接收时产生校验错误($UAnSTR.UAnPE$)或帧错误，继续接收直到接收到第一个停止位，并在接收完全后输出 $INTUAnR$ 。

图 15-13. UART 接收



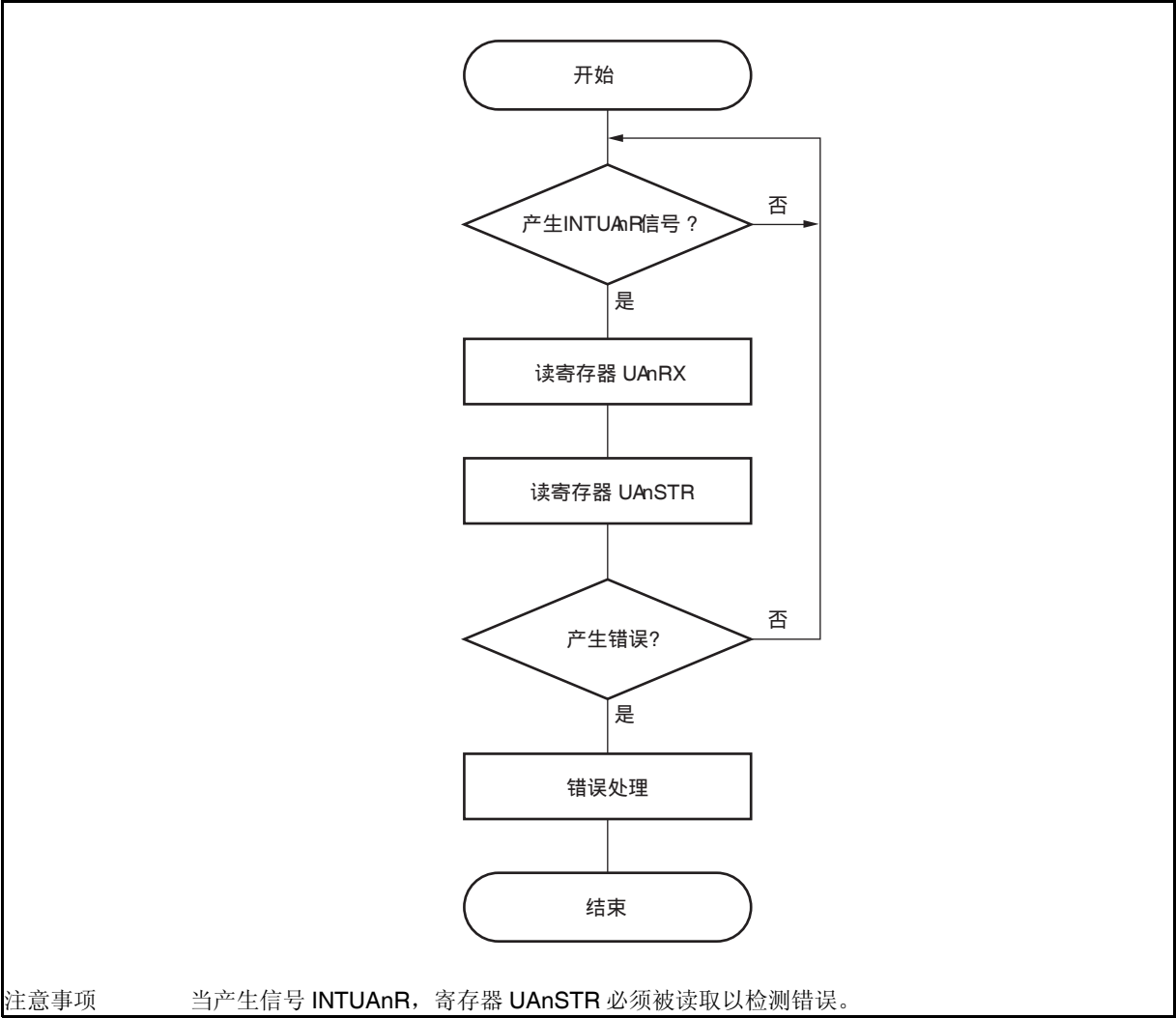
- 注意事项
1. 即使产生接收错误，也要确保读取寄存器 $UAnRX$ 。如果寄存器 $UAnRX$ 不被读取，当接收到下一个数据时会产生溢出错误，而且接收错误状态保持不变。
 2. 接收始终按“停止位的个数 = 1”的情况执行，第 2 个停止位被忽略。
 3. 在接收完成，产生接收完成中断请求信号后，读取寄存器 $UAnRX$ ，使 $UAnPWR$ 或 $UAnRXE$ 清零。如果 $UAnPWR$ 或 $UAnRXE$ 在信号 $INTUAnR$ 产生前被清零，寄存器 $UAnRX$ 的读取值不被保证。
 4. 如果 $UARTAn$ 的接收完成处理过程(产生信号 $INTUAnR$)和 $UAnPWR = 0$ 或 $UAnRXE = 0$ 产生冲突，无论数据是否被存储到寄存器 $UAnRX$ 中都产生信号 $INTUAnR$ 。
为了在不等待 $INTUAnR$ 信号产生的情况下完成接收，设置中断控制寄存器($UAnRIC$)的中断屏蔽标志($UAnRMK$)为 1，且 $UAnPWR = 0$ 或 $UAnRXE = 0$ ，确保寄存器 $UAnRIC$ 的中断请求标志($UAnRIF$)清零。

15.6.8 接收错误

接收过程中的错误有三种：校验错误，帧错误和溢出错误。寄存器 UAnSTR 的数据接收结果错误标志置 1，当产生错误时，输出接收完成中断请求信号(INTUAnR)。

可以通过读取寄存器 UAnSTR 判断接收过程出现何种错误。
在读取接收结果错误标志后，清除该寄存器。

接收数据读取流程



接收错误原因

错误标志	接收错误	原因
UAnPE	校验错误	接收的校验位与设置不匹配
UAnFE	帧错误	没检测到停止位
UAnOVE	溢出错误	在数据被从接收缓冲中读取前，接收到下一个完整数据

当产生接收错误，根据错误类型进行如下的相应处理。

- 校验错误
如果传输过程中由于噪声等原因接收到的错误数据，丢弃该数据并重新发送。
- 帧错误
在发送方和接收方之间可能产生波特率错误，或者起始位可能检测错误。由于帧错误是通信格式的致命错误，检查发送方操作停止，双方重新初始化，然后重新开始通信。
- 溢出错误
由于在读取接收到的数据前，下一次接收完成，1 帧的数据将被丢弃。如果需要，重传该数据。

注意事项 连续接收过程中产生接收错误中断，在下一次接收完成前必须读取寄存器中的内容，并执行错误处理。

15.6.9 校验类型和操作

注意事项 当使用 LIN 功能，固定寄存器 UAnCTL0 的 UAnPS1 和 UAnPS0 位为 00。

通信数据的校验位用于检测位错误。通常，发送方和接收方使用同种校验方式。

奇校验和偶校验模式下，可以检测到奇数位错误。零校验和无校验模式下，不能检测错误。

(a) 偶校验

(i) 发送

控制发送数据，包括校验位，使得数据中“1”的个数为偶数。校验位的取值如下。

- 如果发送数据有奇数个“1”：1
- 如果发送数据有偶数个“1”：0

(ii) 接收

计算接收数据(包括校验位)中“1”的个数。如果“1”的个数为奇数，则产生校验错误。

(b) 奇校验

(i) 发送

与偶校验相反，控制发送数据，包括校验位，使得数据中“1”的个数为奇数。校验位的取值如下。

- 如果发送数据有奇数个“1”：0
- 如果发送数据有偶数个“1”：1

(ii) 接收

计算接收数据(包括校验位)中“1”的个数。如果“1”的个数为偶数，则产生校验错误。

(c) 零校验

无论发送数据为何值，发送时校验位清零。

接收数据时不检测校验位。因此无论校验位为何值，都不会产生校验错误。

(d) 无校验

发送数据中没有校验位。

进行接收操作时认为接收数据中没有校验位。因此也不会产生校验错。

15.6.10 接收数据的噪声过滤器

使用由预分频模块输出的基准时钟采样 **RxDAn** 信号。

如果两次采样值相同，则匹配检测器的输出会发生变化，并把采样的数据作为输入数据。因此，不超过 **2** 个时钟宽度的数据被判断为噪声，不向内部电路发送(参见图 15-15)。参见 15.7 (1) (a) 基本时钟。

此外，如图 15-14 所示的电路结构，接收操作的内部过程与外部信号状态相比延迟三个时钟才执行。

图 15-14. 噪声过滤器电路

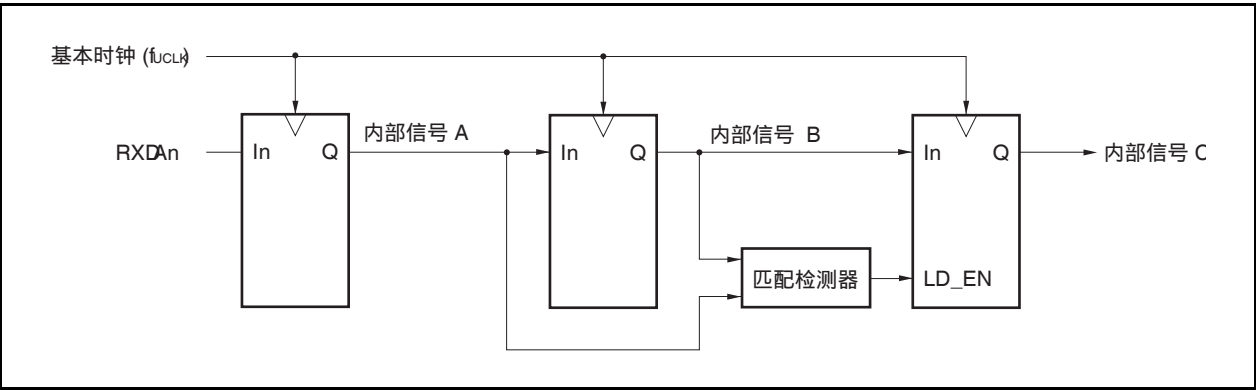
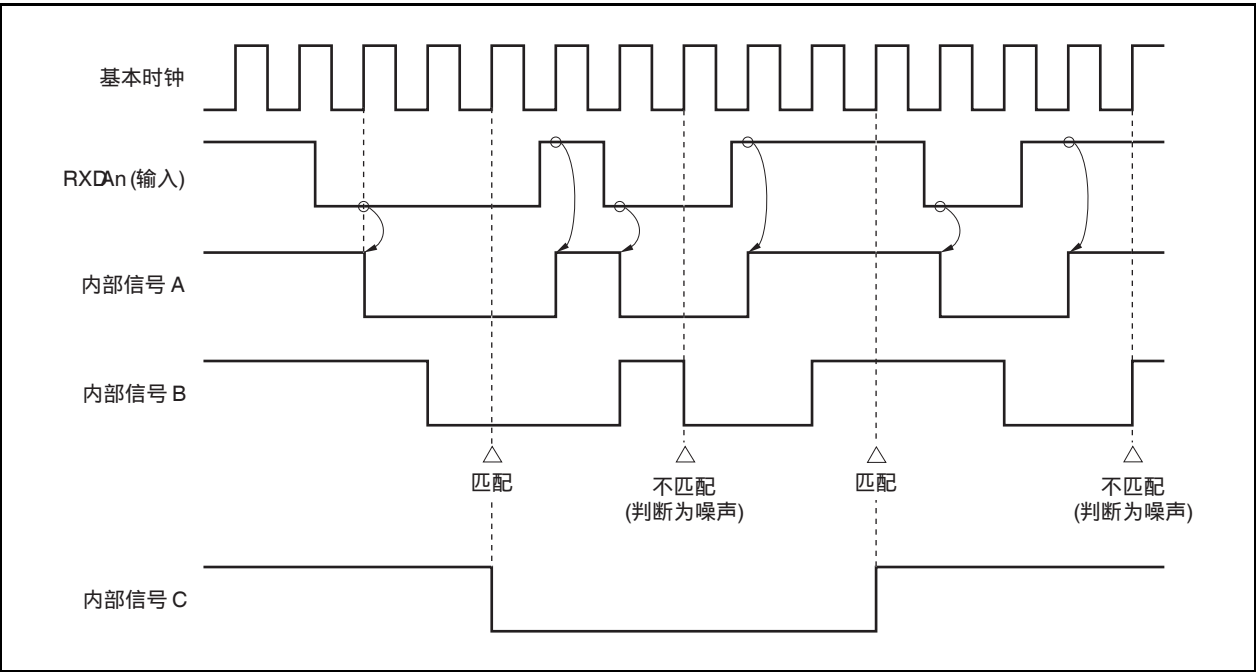


图 15-15. RxDAn 信号判断为噪声的时序



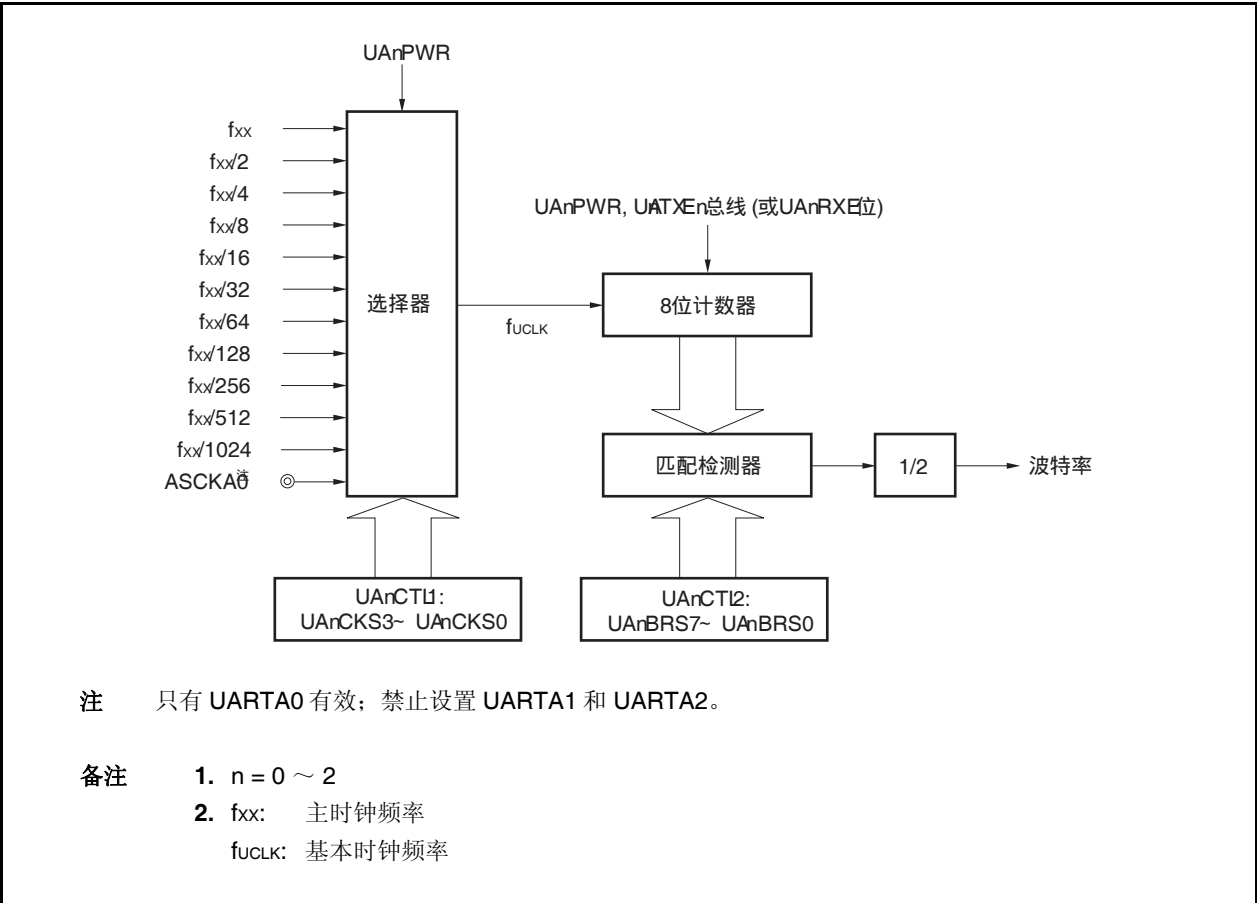
15.7 专用波特率发生器

专用波特率发生器由一个源时钟选择器和一个 8 位可编程计数器组成，用于产生 UARTAn 发送/接收的串行时钟。可选择专用波特率发生器为指定通道提供时钟。
分别使用不同的8位计数器用于发送和接收。

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 15-16. 波特率发生器的配置



(a) 基本时钟

当 UAnCTL0.UAnPWR 置 1，由 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 设置的时钟用作 8 位计数器。该时钟称为基本时钟(f_{UCLK})。

(b) 串行时钟发生器

通过设置寄存器 UAnCTL1 和 UAnCTL2($n = 0 \sim 2$)产生串行时钟。
由 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 设置基本时钟。
由 UAnCTL2.UAnBRS7 ~ UAnCTL2.UAnBRS0 位设置 8 位计数器的分频值。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

该寄存器是 8 位寄存器，用于选择 UARTAn 的基本时钟。
可由 8 位存储器操作指令设置该寄存器。
复位信号产生将该寄存器设置为 00H。

注意事项 在复写寄存器 UAnCTL1 前，UAnCTL0.UAnPWR 位清零。

复位后: 00H R/W 地址: UA0CTL1 FFFF~~A~~01H, UA1CTL1 FFFF~~A~~11H,
UA2CTL1 FFFF~~A~~21H

UAnCTL1
(n = 0 ~ 2)

7	6	5	4	3	2	1	0
0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	基本时钟(f _{CLK}) 选择
0	0	0	0	f _{xx}
0	0	0	1	f _{xx} /2
0	0	1	0	f _{xx} /4
0	0	1	1	f _{xx} /8
0	1	0	0	f _{xx} /16
0	1	0	1	f _{xx} /32
0	1	1	0	f _{xx} /64
0	1	1	1	f _{xx} /128
1	0	0	0	f _{xx} /256
1	0	0	1	f _{xx} /512
1	0	1	0	f _{xx} /1,024
1	0	1	1	外部时钟 ^注 (引脚ASCKA0)
其它情况				禁止设置

注 只有 UARTA0 有效；禁止设置 UARTA1 和 UARTA2。

备注 f_{xx}: 主时钟频率

(3) UARTAn 控制寄存器 2 (UAnCTL2)

该寄存器是 8 位寄存器，用于控制 UARTAn 的波特率。

可由 8 位存储器操作指令设置该寄存器。

复位信号产生将该寄存器设置为 FFH。

注意事项 在复写寄存器 UAnCTL2 前，UAnCTL0.UAnPWR 位或 UAnTXE 和 UAnRXE 位清零。

复位后：FFH R/W 地址： UA0CT12 FFFF02H, UA1CT12 FFFF02H, UA2CT12 FFFF02H

	7	6	5	4	3	2	1	0
UAnCT12	UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0

(n = 0 ~ 2)

UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0	Default (k)	串行时钟
0	0	0	0	0	0				禁止设置
0	0	0	0	0	1	0	0	4	fucLK/4
0	0	0	0	0	1	0	1	5	fucLK/5
0	0	0	0	0	1	1	0	6	fucLK/6
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	fucLK/252
1	1	1	1	1	1	0	1	253	fucLK/253
1	1	1	1	1	1	1	0	254	fucLK/254
1	1	1	1	1	1	1	1	255	fucLK/255

备注 f_{UCLK}: 由 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 位选择时钟频率

(4) 波特率

通过下列公式计算波特率。

$$\text{波特率} = \frac{f_{\text{UCLK}}}{2 \times k} [\text{bps}]$$

当使用外部时钟，公式如下(在 UARTA0 工作时使用引脚 ASCKA0，使用上述公式计算)。

$$\text{波特率} = \frac{f_{\text{xx}}}{2^{m+1} \times k} [\text{bps}]$$

备注 f_{UCLK} = 由 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 设置的基本时钟频率
 f_{xx} : 主时钟频率
 m = 由 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 设置的数值 ($m = 0 \sim 10$)
 k = 由 UAnCTL2.UAnBRS7 ~ UAnCTL2.UAnBRS0 设置的数值 ($k = 4 \sim 255$)

通过下列公式计算波特率误差。

$$\begin{aligned} \text{Error (\%)} &= \left(\frac{\text{实际波特率 (有误差的波特率)}}{\text{预期波特率 (正确的波特率)}} - 1 \right) \times 100 [\%] \\ &= \left(\frac{f_{\text{UCLK}}}{2 \times k \times \text{预期波特率}} - 1 \right) \times 100 [\%] \end{aligned}$$

当使用内部时钟，公式如下(在 UARTA0 工作时使用引脚 ASCKA0，使用上述公式计算)。

$$\text{Error (\%)} = \left(\frac{f_{\text{xx}}}{2^{m+1} \times k \times \text{预期波特率}} - 1 \right) \times 100 [\%]$$

- 注意事项**
1. 发送期间必须保持波特率误差在接收端允许的误差范围内。
 2. 接收期间，波特率误差必须满足“(5) 接收期间允许的波特率范围”中所描述的范围。

为了设置波特率，进行如下计算并设置寄存器 UAnCTL1 和 UAnCTL2(当使用内部时钟)。

- <1> 设置 $k = f_{xx}/(2 \times \text{预期波特率})$ 。设置 $m = 0$ 。
<2> 设置 $k = k/2$ 和 $m = m + 1$ ，其中 $k \geq 256$ 。
<3> 重复步骤 <2> 直到 $k < 256$ 。
<4> k 进行四舍五入，即取整。
如果取整后 $k = 256$ ，再次进行步骤<2>的操作(此时 k 为 128)。
<5> 设置 m 到寄存器 UAnCTL1， k 到寄存器 UAnCTL2。

例如： 当 $f_{xx} = 20 \text{ MHz}$ 且预期波特率 = 153,600 bps
<1> $k = 20,000,000/(2 \times 153,600) = 65.10\dots$, $m = 0$
<2>, <3> $k = 65.10\dots < 256$, $m = 0$
<4> 设置寄存器 UAnCTL2 的值: $k = 65 = 41\text{H}$, 设置寄存器 UAnCTL1 的值: $m = 0$

实际波特率 $= 20,000,000/(2 \times 65)$
 $= 153,846 \text{ [bps]}$

波特率误差 $= \{20,000,000/(2 \times 65 \times 153,600) - 1\} \times 100$
 $= 0.160 \text{ [%]}$

波特率设置的典型示例如下。

表 15-3. 波特率发生器的数据设置

波特率 (bps)	$f_{xx} = 20 \text{ MHz}$			$f_{xx} = 18.874 \text{ MHz}$			$f_{xx} = 16 \text{ MHz}$			$f_{xx} = 10 \text{ MHz}$		
	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)
300	08H	82H	0.16	07H	F6H	-0.10	07H	D0H	0.16	07H	82H	0.16
600	07H	82H	0.16	06H	F6H	-0.10	06H	D0H	0.16	06H	82H	0.16
1,200	06H	82H	0.16	05H	F6H	-0.10	05H	D0H	0.16	05H	82H	0.16
2,400	05H	82H	0.16	04H	F6H	-0.10	04H	D0H	0.16	04H	82H	0.16
4,800	04H	82H	0.16	03H	F6H	-0.10	03H	D0H	0.16	03H	82H	0.16
9,600	03H	82H	0.16	02H	F6H	-0.10	02H	D0H	0.16	02H	82H	0.16
19,200	02H	82H	0.16	01H	F6H	-0.10	01H	D0H	0.16	01H	82H	0.16
31,250	01H	A0H	0	01H	97H	-0.01	01H	80H	0	00H	A0H	0
38,400	01H	82H	0.16	00H	F6H	-0.10	00H	D0H	0.16	00H	82H	0.16
76,800	00H	82H	0.16	00H	7BH	-0.10	00H	68H	0.16	00H	41H	0.16
153,600	00H	41H	0.16	00H	3DH	0.72	00H	34H	0.16	00H	21H	-1.36
312,500	00H	20H	0	00H	1EH	0.66	00H	1AH	-1.54	00H	10H	0

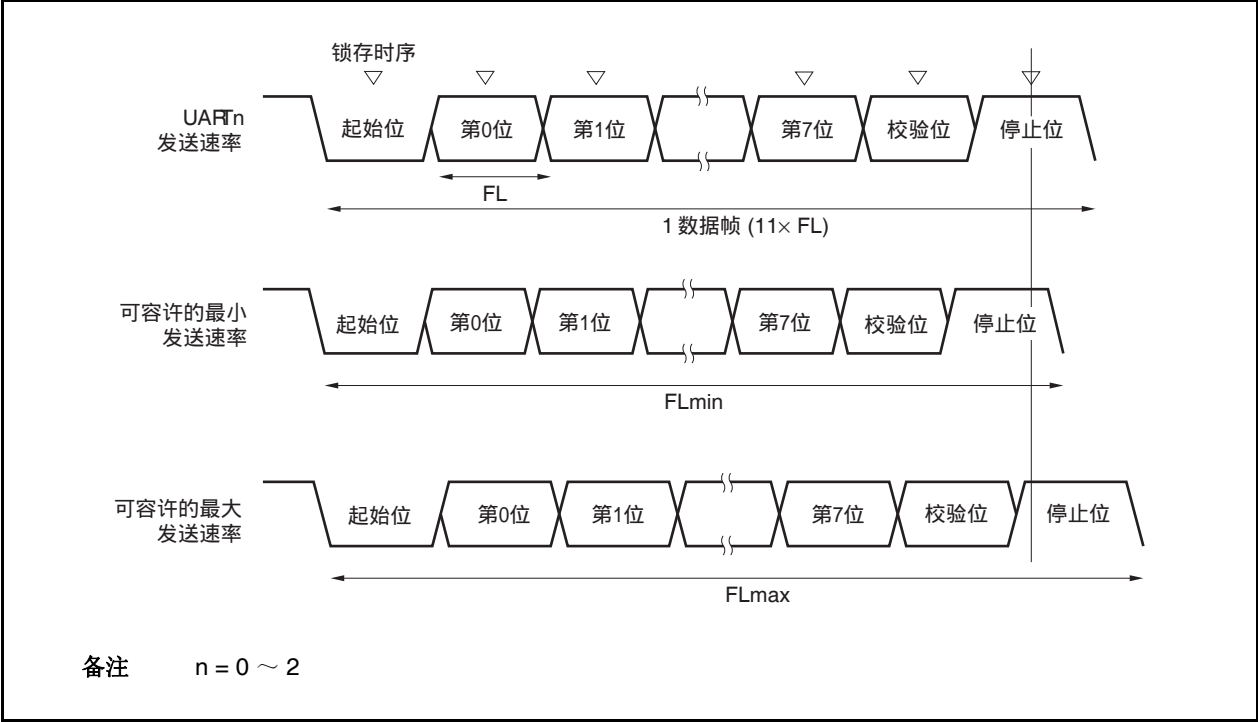
备注 f_{xx} : 主时钟频率
 ERR: 波特率误差 (%)

(5) 接收期间允许的波特率范围

接收期间来自发送端的波特率可允许的误差如下所示。

注意事项 必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。

图 15-17. 接收期间允许的波特率范围



如图 15-17 所示，当检测到起始位后，接收数据的锁存时序由寄存器 UAnCTL2 设置的计数器确定。如果数据的最后一位(停止位)满足该锁存时序，则发送数据可被正确接收。
假定接收数据为 11 位，各项理论值计算如下。

$$FL = (\text{Brate})^{-1}$$

- Brate: UARTAn 波特率(n = 0 ~ 2)
- k: UAnCTL2.UAnBRS7 ~ UAnCTL2.UAnBRS0 位的设置值 (n = 0 ~ 2)
- FL: 1 位数据宽度
- 锁存时序极限: 2 个时钟

可允许的最小数据帧宽度:
$$FL_{min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

因此在发送端设定的可接收的最大波特率如下所示。

$$BR_{max} = (FL_{min}/11)^{-1} = \frac{22k}{21k + 2} \text{ Brate}$$

同样，可计算允许的最大数据帧宽度。

$$\frac{10}{11} \times FL_{max} = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FL_{max} = \frac{21k - 2}{20 k} FL \times 11$$

因此在接收端可接收的最小波特率如下所示。

$$BR_{min} = (FL_{max}/11)^{-1} = \frac{20k}{21k - 2} \text{ Brate}$$

可通过上述最小和最大波特率公式计算在 UARTAn 与接收端之间允许的波特率误差，如下所示。

表 15-4. 允许的最大/最小波特率误差

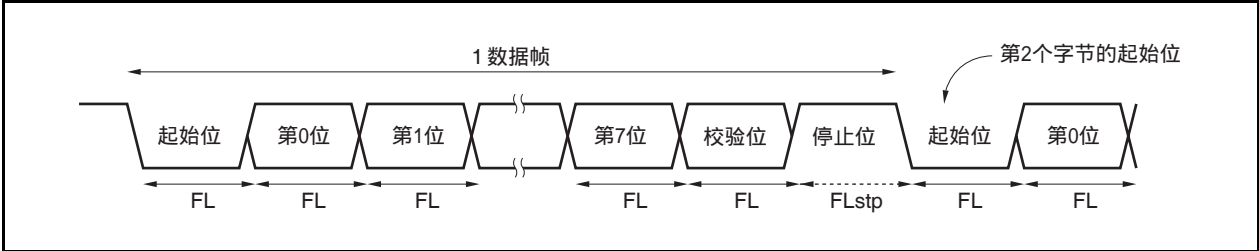
分频比 (k)	允许的最大波特率误差	允许的最小波特率误差
4	+2.32%	-2.43%
8	+3.52%	-3.61%
20	+4.26%	-4.30%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.72%

- 备注
1. 接收准确度取决于每帧的每帧的位数、输入时钟频率和分频比(k)。输入时钟频率和分频比(k)越高，准确度就更高。
 2. k: UAnCTL2.UAnBRS7 ~ UAnCTL2.UAnBRS0 位的设置值(n = 0 至 2)

(6) 连续发送期间的波特率

当连续发送数据时，从停止位到下一个起始位的传送速率一般为 2 个基本时钟。不过，由于在检测到起始位时接收端的时序被初始化，因此通信结果不会受到影响。

图 15-18. 连续发送期间的传送速率



当 1 位数据宽度为 FL、停止位宽度为 FLstp，基本时钟频率为 f_{xCLK} 时，有以下公式成立。

$$FLstp = FL + 2/f_{uCLK}$$

因此，连续发送期间的发送速率为：

$$\text{发送速率} = 11 \times FL + (2/f_{uCLK})$$

15.8 注意事项

- (1) 当 UARTAn 所使用的时钟停止工作(例如, 进入 IDLE1, IDLE2, 或 STOP 模式), 操作停止, 每个寄存器保持时钟停止前的值。引脚 TXDAn 的输出继续保持时钟停止前的值。然而, 时钟恢复后, 操作不被保证。因此, 时钟恢复后, 通过设置 UAnCTL0.UAnPWR, UAnCTL0.UAnRXEn, 和 UAnCTL0.UAnTXEn 为 000, 使电路初始化。
- (2) 引脚 RXDA1 和 KR7 不能同时使用。使用引脚 RXDA1 时, 不要使用引脚 KR7。使用引脚 KR7 时, 不要使用引脚 RXDA1(推荐设置 PFC91 为 1 且 PFCE91 为 0)。
- (3) UARTAn 下, 不产生由于通信错误引起的中断。当使用 DMA 传送接收和发送的数据, 即使传送期间出现错误(校验, 溢出, 帧错误), 也不进行处理。可以在 DMA 传送后读取寄存器 UAnSTR 查看是否出错, 或着在通信过程中读取寄存器 UAnSTR 查看是否出错。
- (4) 按照下述次序启动 UARTAn。
 - <1> 设置 UAnCTL0.UAnPWR 为 1。
 - <2> 设置端口。
 - <3> 设置 UAnCTL0.UAnTXE 为 1, UAnCTL0.UAnRXE 为 1。
- (5) 按照下述次序停止 UARTAn。
 - <1>设置 UAnCTL0.UAnTXE 为 0, UAnCTL0.UAnRXE 为 0。
 - <2>设置端口, 设置 UAnCTL0.UAnPWR 为 0 (如果端口设置不改变则不需要修改)。
- (6) 发送模式下(UAnCTL0.UAnPWR = 1 和 UAnCTL0.UAnTXE = 1), 不要写相同的值到寄存器 UAnTX, 因为写寄存器后传输就开始了。连续发送相同的值时, 写相同的值到寄存器 UAnTX。
- (7) 连续发送模式下, 从停止位到下一个起始位的通信速率比正常情况下的 2 个基本时钟要长。然而, 接收端通过检测起始位初始化时序, 所以结果不受影响。

第十六章 3 线长度可变串行 I/O (CSIB)

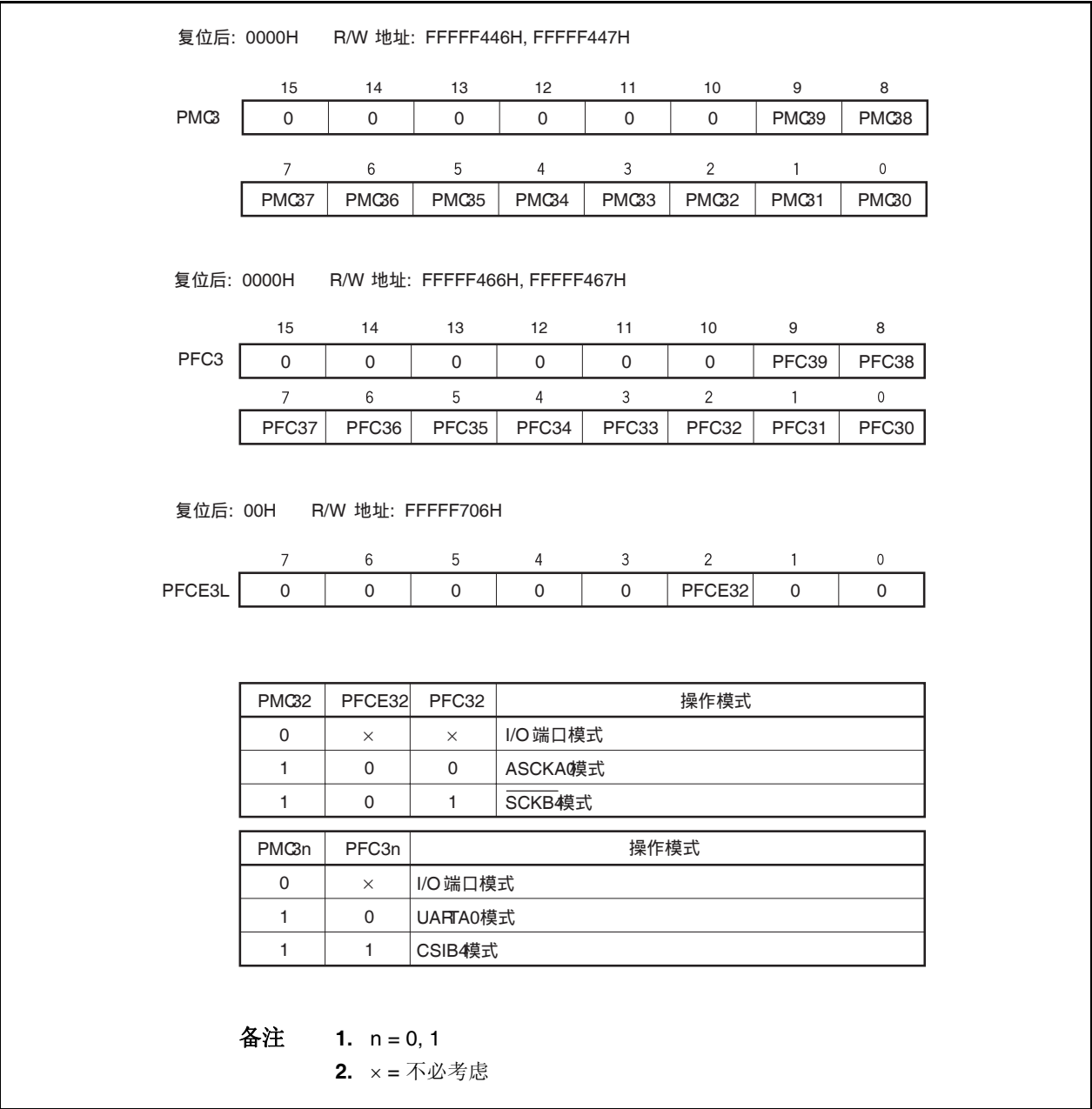
16.1 CSIB 和其它串行接口的模式转换

16.1.1 CSIB4 和 UARTA0 模式转换

V850ES/JG2 中, CSIB4 和 UARTA0 复用相同引脚, 因此不能同时使用。在使用前, 先用寄存器 PMC3 和 PFC3 设置复用相同引脚, 因此不能同时使用。在使用前, 先用寄存器 PMC3 和 PFC3 设置 UARTA0。

注意事项 如果 CSIB4 和 UARTA0 的发送/接收操作过程中, 进行转换, 这两个功能不能保证。确保不被使用的那个处于非使能态。

图 16-1. CSIB4 和 UARTA0 模式转换设置

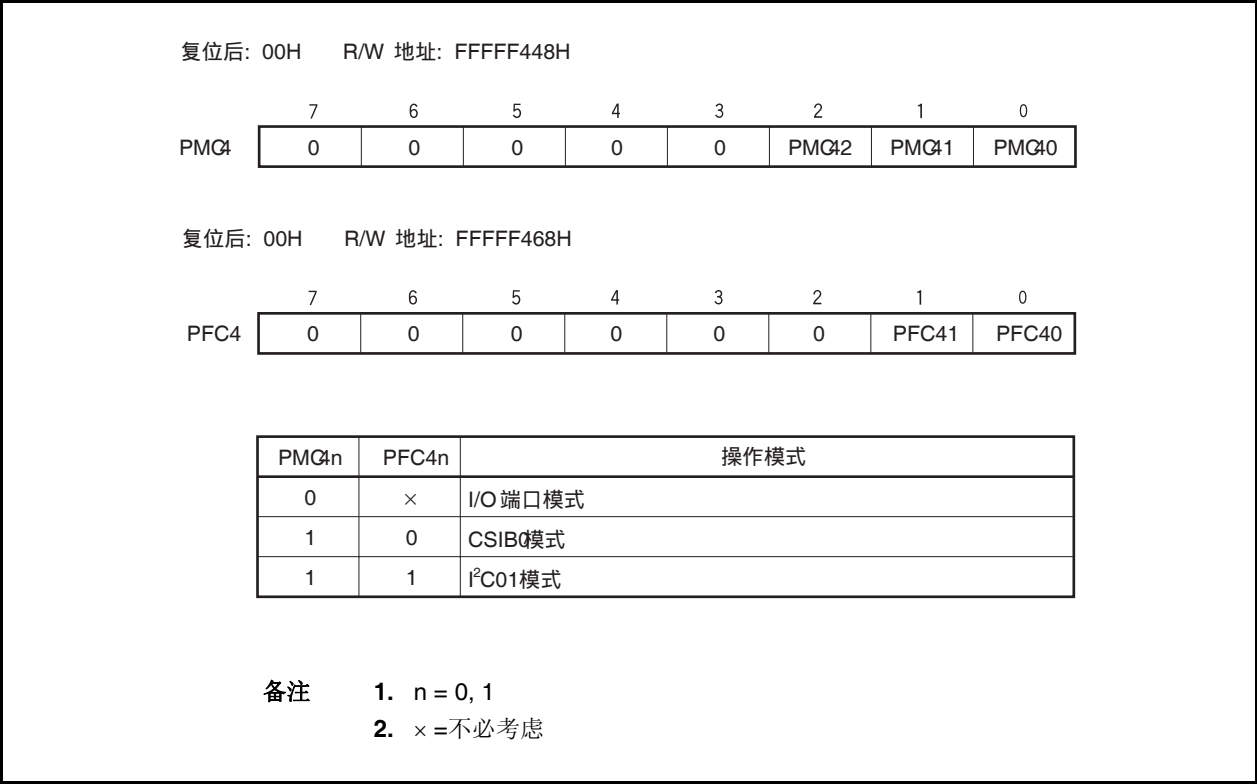


16.1.2 CSIB0 和 I²C01 模式转换

在 V850ES/JG2 的 I²C 总线版本(Y 版本)中, CSIB0 和 I²C01 复用相同引脚, 因此不能同时使用。在使用前, 先用寄存器 PMC4 和 PFC4 设置 CSIB0。

注意事项 如果 CSIB0 和 I²C01 的发送/接收操作过程中, 进行转换, 这两个功能不能保证。确保不被使用的那个处于非使能态。

图 16-2. CSIB0 和 I²C01 的模式转换设置



16.2 特征

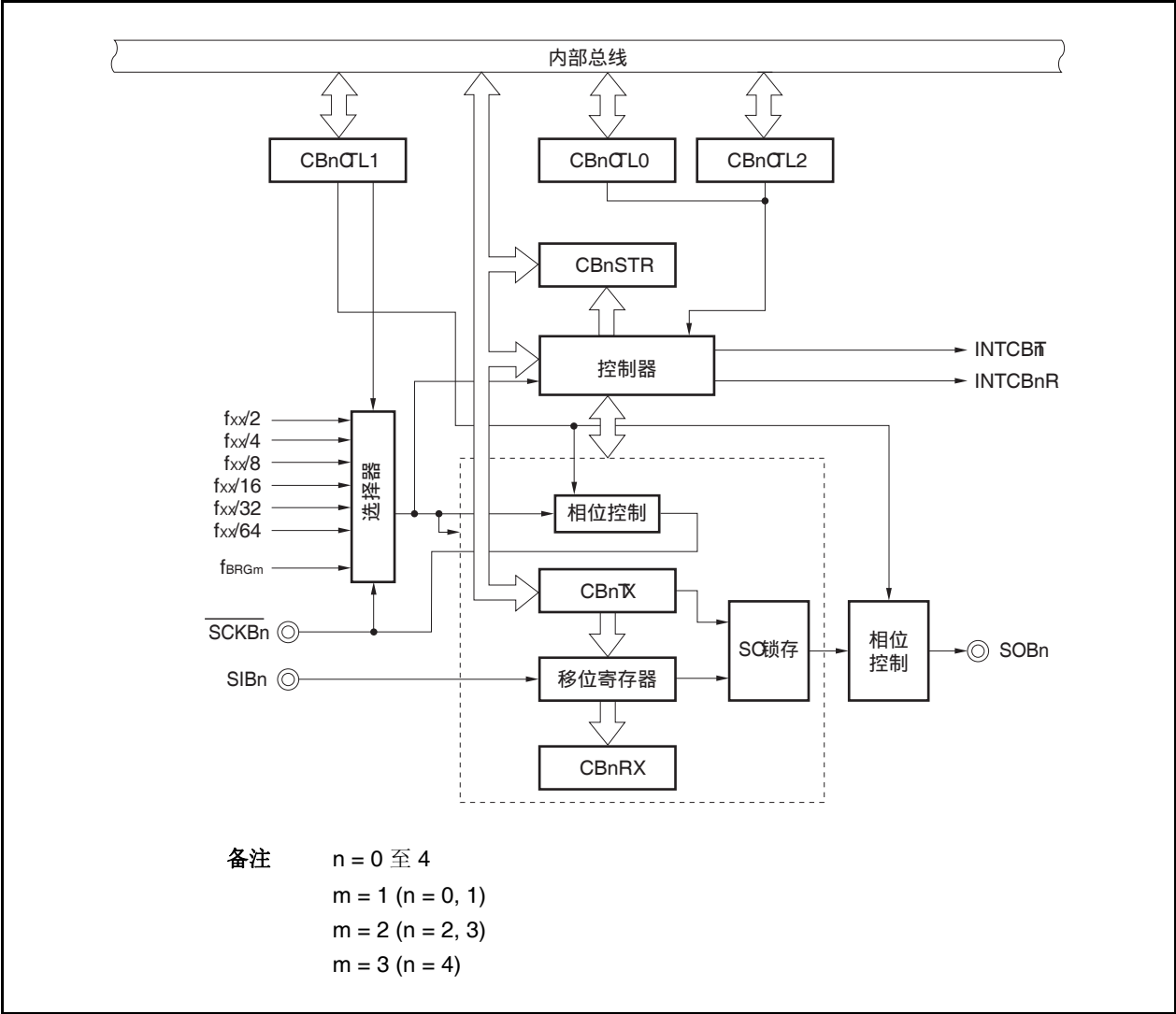
- 传输速率: 8 Mbps ~ 4.9 kbps ($f_{xx} = 20 \text{ MHz}$, 使用内部时钟)
- 主/从模式选择
- 8 位~16 位通信数据宽度, 3 线串行接口
- 中断请求信号(INTCBnT , INTCBnR) $\times 2$
- 可选串行时钟和数据相位
- 可由 1 位操作指令在 8 和 16 位中选择通信数据长度
- 可选 MSB-/LSB-first 通信
- 3 线通信
 - SOBn: 串行数据输出
 - SIBn: 串行数据输入
 - $\overline{\text{SCKBn}}$: 串行时钟输出
- 可选发送模式, 接收模式和发送/接收模式

备注 $n = 0 \text{ 至 } 4$

16.3 配置

CSIBn 的框图如下。

图 16-3. CSIBn 的框图



CSIBn 由以下寄存器控制。

表 16-1. CSIBn 的配置

项目	配置
寄存器	CSIBn 数据接收寄存器(CBnRX) CSIBn 数据发送寄存器(CBnTX)
控制寄存器	CSIBn 控制寄存器 0 (CBnCTL0) CSIBn 控制寄存器 1 (CBnCTL1) CSIBn 控制寄存器 2 (CBnCTL2) CSIBn 状态寄存器(CBnSTR)

该寄存器是 **16 位**缓冲寄存器，用于保持接收数据。

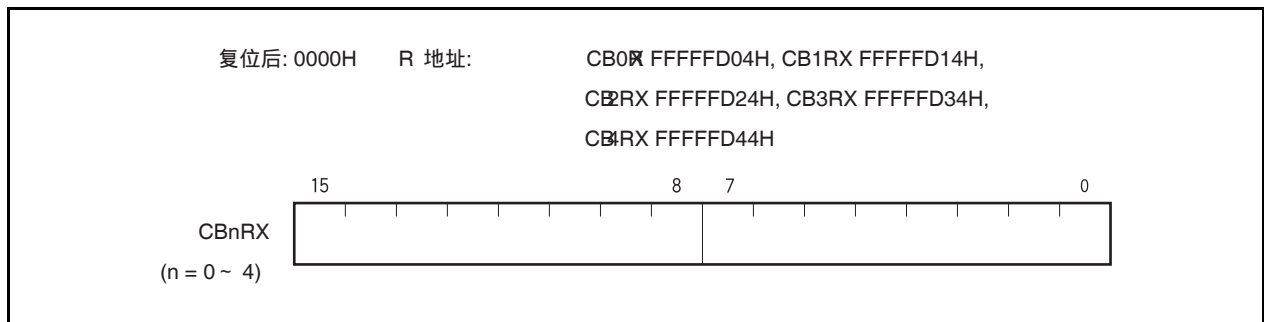
该寄存器是只读的，可由 **16 位**存储器操作指令进行设置。

在接收使能状态下，通过读取寄存器 **CBnRX** 开始接收操作。

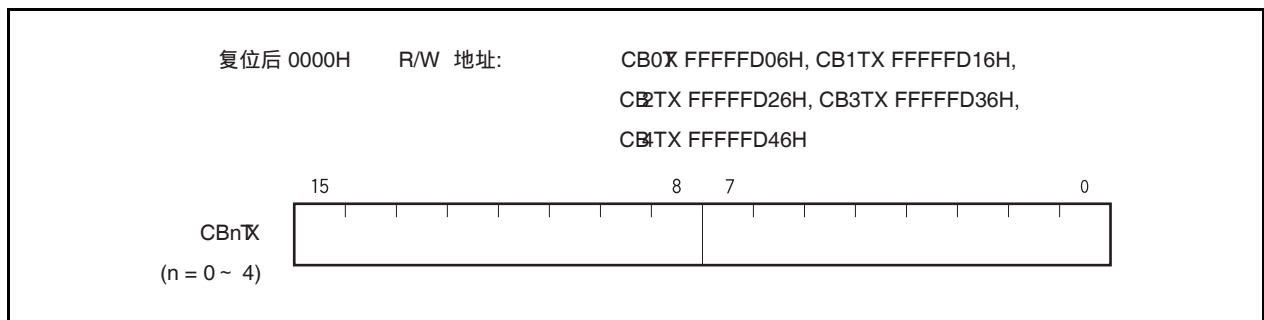
如果通信数据的长度是 **8 位**，该寄存器的低 **8 位**是只读的，由 **8 位**存储器操作指令设置寄存器 **CBnRXL**。

复位信号产生将该寄存器设置为 **0000H**。

除了复位信号，通过清除(置 0)寄存器 **CBnCTL0** 的 **CBnPWR** 位，寄存器 **CBnRX** 可被初始化。



寄存器是 16 位缓冲寄存器，用于写入 CSIBn 发送数据。
该寄存器可以读/写，可由 16 位存储器操作指令进行设置。
在发送使能状态下，通过写数据到寄存器 CBnTX 中开始发送操作。
如果通信数据的长度是 8 位，该寄存器的低 8 位是只读的，由 8 位存储器操作指令设置寄存器 CBnTXL。
复位信号产生将该寄存器设置为 0000H。



16.4 寄存器

下述寄存器用于控制 CSIBn。

- CSIBn 控制寄存器 0 (CBnCTL0)
- CSIBn 控制寄存器 1 (CBnCTL1)
- CSIBn 控制寄存器 2 (CBnCTL2)
- CSIBn 状态寄存器 (CBnSTR)

(1) CSIBn 控制寄存器 0 (CBnCTL0)

该寄存器 CSIBn 控制串行通信操作。

可由 1 位或 8 位存储器操作指令读/写该寄存器。

复位信号产生将该寄存器设置为 01H。

(1/3)

复位后: 01H R/W 地址: CB0CTL0 FFFFFFFD00H, CB1CTL0 FFFFFFFD10H,
CB2CTL0 FFFFFFFD20H, CB3CTL0 FFFFFFFD30H,
CB4CTL0 FFFFFFFD40H

	<7>	<6>	<5>	<4>	3	2	1	<0>
CBnCTL0 (n = 0 ~ 4)	CBnPWR	CBnTxE	CBnRxE	CBnDI	0	0	CBnTMS	CBnSCE

CBnPWR	CSIBn操作禁止/使能控制
0	禁止CSIBn操作并复位寄存器 CBnSTR
1	CSIBn操作使能
• CBnPWR位控制CSIBn操作并复位内部电路。	

CBnTxE	发送操作禁止/使能控制
0	禁止发送操作
1	发送操作使能
• 当CBnTxE位为0时，SOB输出为低电平。	

CBnRxE	接收操作禁止/使能控制
0	禁止接收操作
1	接收操作使能
• 当CBnRxE位被清零，不产生接收完全中断， 即使为禁止接收操作而发送指定数据， 接收数据(寄存器CBnRX也不被更新。	

注 这些位只能在 CBnPWR = 0 时复写。但在复写这些位时可以设置 CBnPWR = 1。

注意事项 为了强行停止发送/接收，清除 CBnPWR 位，而不是 CBnRxE 位。
同时，时钟输出停止。

★

CBnDIR	发送方向模式选择(MSB/LSB)
0	
1	

CBnTMS	发送模式选择
0	单独发送模式
1	连续发送模式

[单独发送模式]
当通信完成，产生接收完全中断 (INTCBnR)
即使发送使能 (CBnTXE= 1)，不产生发送使能中断(INTCBnT)
如果在通信期间(CBnSTR.CBnS位 = 1)写入下一个发送数据，该数据被忽略，且开始下一次通信。同样，如果设置只能接收模式(CBnTXE= 0, CBnRXE= 1)，即使接收数据在通信期间(CBnSTR.CBnS位 = 1)被读取，也不开始下一次通信。

[连续发送模式]
在通信期间(CBnSTR.CBnS位 = 1)写入下一个发送数据，即可使能连续发送模式。产生发送使能中断 (INTCBnT)后，才可写入下一个发送数据。如果在连续发送模式下设置了只能接收模式(CBnTXE= 0, CBnRXE= 1)，无论是否在产生接收完成中断 (INTCBnR)读取寄存器CBnRX，下一次接收都将立即开始。
因此，要立刻读取寄存器CBnR中接收到的数据。如果延迟执行此操作，将产生溢出错误(CBnOVE位= 1)。

注 这些位只能在 CBnPWR = 0 时复写。但在复写这些位时可以设置 CBnPWR = 1。

CBnSCE	开始发送禁止/使能选择
0	通信开始触发无效
1	通信开始触发有效

• 主模式
 该位用于禁止或使能通信开始触发。
 (a) 单独发送或发送/接收模式，或连续发送或连续发送/接收模式
 CbSC位不影响通信操作。
 (b) 单独接收模式
 读取最后接收到的数据前，清零CBnSC位。因为读取接收数据寄存器CBnRX接收已经开始^{注1}。
 (c) 连续接收模式
 在接收到最后一个字节的数据前，清零CBnSC位。从而禁止接收^{注2}。

• 从模式
 该位用于使能或禁止通信开始触发。
 设置CBnSC位为1。

[CBnSC位的使用]

• 单独接收模式
 <1> 当INTCBn中断服务子程序接收到最后一字节数据，在读取寄存器CBnRX前清除CBnSC位。
 <2> 确定 CBnSTR.CBSCF位= 0后，清零 CBnRX位以禁止接收。
 要连续接收，设置CBnSC位为1，以开始下一次接收，通过虚拟读取寄存器CBnRX

• 连续接收模式
 <1> 接收最后一个数据时在INTCBn中断服务子程序里清除CBnSC位。
 <2> 读取寄存器CBnRX
 <3> 响应CBnTI中断后，从寄存器CBnRX读取最后接收到的数据。
 <4> 确定 CBnSTR.CBSCF位= 0后，清零 CBnRX位以禁止接收。
 要连续接收，设置CBnSC位为1，以等待下一次接收，通过虚拟读取寄存器CBnRX

- 注
1. 但 CBnSCE 位为 1 时被读取，下一次通信操作开始。
 2. 如果 CBnSCE 位在最后一个数据接收完成前的一个通信时钟处没有被清零，那么下一次通信操作将自动开始。

注意事项 确保 2 和 3 位为 0。

(2) CSIBn 控制寄存器 1 (CBnCTL1)

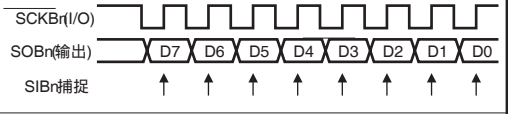
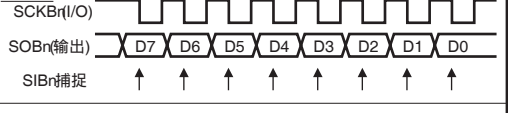
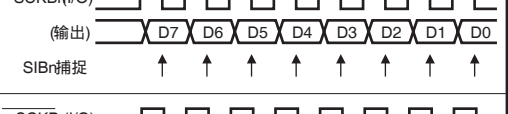
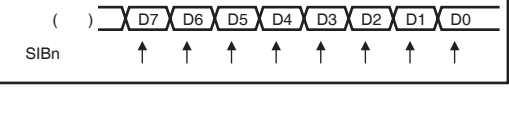
该寄存器是 8 位寄存器，用于控制 CSIBn 串行通信操作。
可由 1 位或 8 位存储器操作指令读/写该寄存器。
复位信号产生将该寄存器设置为 00H。

注意事项 只有当 CBnCTL0.CBnPWR = 0 时，寄存器 CBnCTL1 可被复写。

复位后 00H R/W 地址: CB0CTL1 FFFFFFFD01H, CB1CTL1 FFFFFFFD11H,
CB2CTL1 FFFFFFFD21H, CB3CTL1 FFFFFFFD31H,
CB4CTL1 FFFFFFFD41H

	7	6	5	4	3	2	1	0
CBnCTL1	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

(n = 0 ~ 4)

	CBnCKP	CBnDAP	数据发送/接收时序 SCKB的设置
通信类型 1	0	0	
通信类型 2	0	1	
通信类型 3	1	0	
通信类型 4	1	1	

CBnCKS2	CBnCKS1	CBnCKS0	通信时钟	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(SCKBn)	从模式

备注 当 n = 0, 1, m = 1
 当 n = 2, 3, m = 2
 当 n = 4, m = 3
 关于 fBRGm 的详细信息，参见 16.8 波特率发生器。

(3) CSIBn 控制寄存器 2 (CBnCTL2)

该寄存器是 8 位寄存器，用于控制 CSIBn 串行通信的字节长度。
可由 1 位或 8 位存储器操作指令读/写该寄存器。
复位信号产生将该寄存器设置为 00H。

注意事项 只有当 CBnCTL0.CBnPWR = 0 或者 CBnTXE 和 CBnRXE 都为 0 时，寄存器 CBnCTL2 可被复写。

复位后: 00H R/W 地址: CB0CTL2 FFFFFFFD02H, CB1CTL2 FFFFFFFD12H,
CB2CTL2 FFFFFFFD22H, CB3CTL2 FFFFFFFD32H,
CB4CTL2 FFFFFFFD42H

CBnCTL2 (n = 0 ~ 4)	7	6	5	4	3	2	1	0
	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

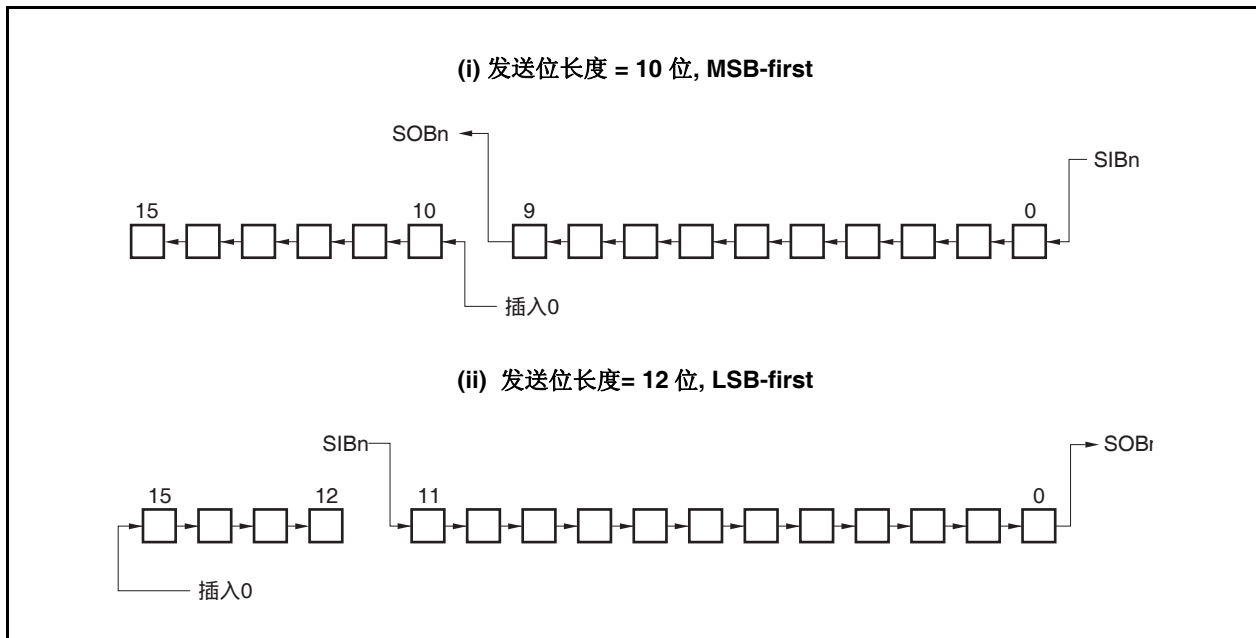
CBnCL3	CBnCL2	CBnCL1	CBnCL0	串行寄存器的位长度
0	0	0	0	8 位
0	0	0	1	9 位
0	0	1	0	10 位
0	0	1	1	11 位
0	1	0	0	12 位
0	1	0	1	13 位
0	1	1	0	14 位
0	1	1	1	15 位
1	×	×	×	16 位

- 备注**
- 1. 如果发送字节数不是 8/16 位，那么要将数据从寄存器 CBnTX 和 CBnRX 的最低有效位处开始存储。
 - 2. ×: 不必考虑

(a) 发送数据长度改变功能

使用 CBnCTL2.CBnCL3 ~ CBnCTL2.CBnCL0, 可由 1 位操作指令设置 CSIBn 发送数据长度(8~16 位)。

当发送位长度设置为非 16 位, 设置寄存器 CBnTX 或 CBnRX 从 LSB 开始, 无论发送起始位是 MSB 或者 LSB。任何数据都可放置到那些不用的高字节位, 但是接收的数据的高字节位在串行发送后都变为 0。



(4) CSIBn 状态寄存器(CBnSTR)

该寄存器是 8 位寄存器，用于显示 CSIBn 的状态。

可由 1 位或 8 位存储器操作指令读/写该寄存器。但标志 CBnTSF 是只读的。

复位信号产生将该寄存器设置为 00H。

除了复位信号，通过清除(置 0)寄存器 CBnCTL0 的 CBnPWR 位，该寄存器可被初始化。

复位后 00H R/W 地址: CB0STR FFFFFFFD03H, CB1STR FFFFFFFD13H,
 CB2STR FFFFFFFD23H, CB3STR FFFFFFFD33H,
 CB4STR FFFFFFFD43H

	<7>	6	5	4	3	2	1	<0>
CBnSTR	CBnTSF	0	0	0	0	0	0	CBnOVE

(n = 0 ~ 4)

CBnTSF	通信状态标志
0	通信停止
1	通信中

•通信过程中，当数据存入寄存器CBnTX中，该寄存器被设置；接收过程中，当虚拟读取寄存器时，该寄存器被设置。
当发送结束，该标志在最后一个时钟边沿被清零。

CBnOVE	溢出错误标志
0	没有溢出
1	溢出

•当接收操作完成后 CPU没有读取接收缓存的值就开始下一次接收时，产生溢出错误。
CBnOVE标志显示产生溢出错误状态。
•单独发送模式下CBnOV位依然有效。因此，只有在发送时注意以下方面。
•不要检查CBnOV标志。
•即使不要求读取接收数据，也要读取该位。
•标志通过写入0来清零。即使写入1也不能被设置。

<R> 16.5 中断请求信号

CSIBn 能够产生下列两个中断信号。

- 接收完成中断请求信号 (INTCBnR)
- 发送使能中断请求信号 (INTCBnT)

对于这两个中断请求信号，接收完成中断请求信号的默认优先级高，而发送使能中断请求信号的默认优先级低。

表 16-2. 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 (INTCBnR)

在接收中断使能时，当接收的数据被送到 CBnRX 寄存器时，则产生接收中断请求信号。
如果溢出错误，也会产生该中断请求信号。
当响应接收完成中断请求信号时，读数据，通过读 CBnSTR 寄存器来检查接收的结果是否正确。
在单发送模式中，在单独发送模式中，即使只有发送执行时，当完成传送时也会产生

(2) 发送使能中断请求信号 (INTCBnT)

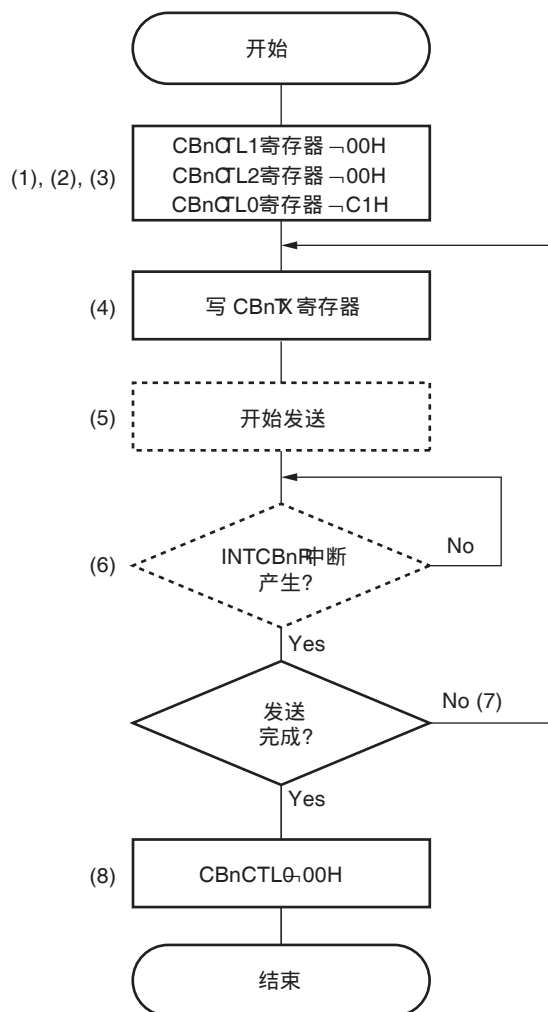
在连续发送或连续发送/接收模式中，一旦使能向 CBnTX 中写数据，数据就会由 CBnTX 寄存器发送出去。
并产生发送使能中断请求信号。
在单独发送和单独发送/接收模式中，不产生 INTCBnT 中断。

<R> 16.6 操作

16.6.1 单独传送模式 (主机模式, 发送模式)

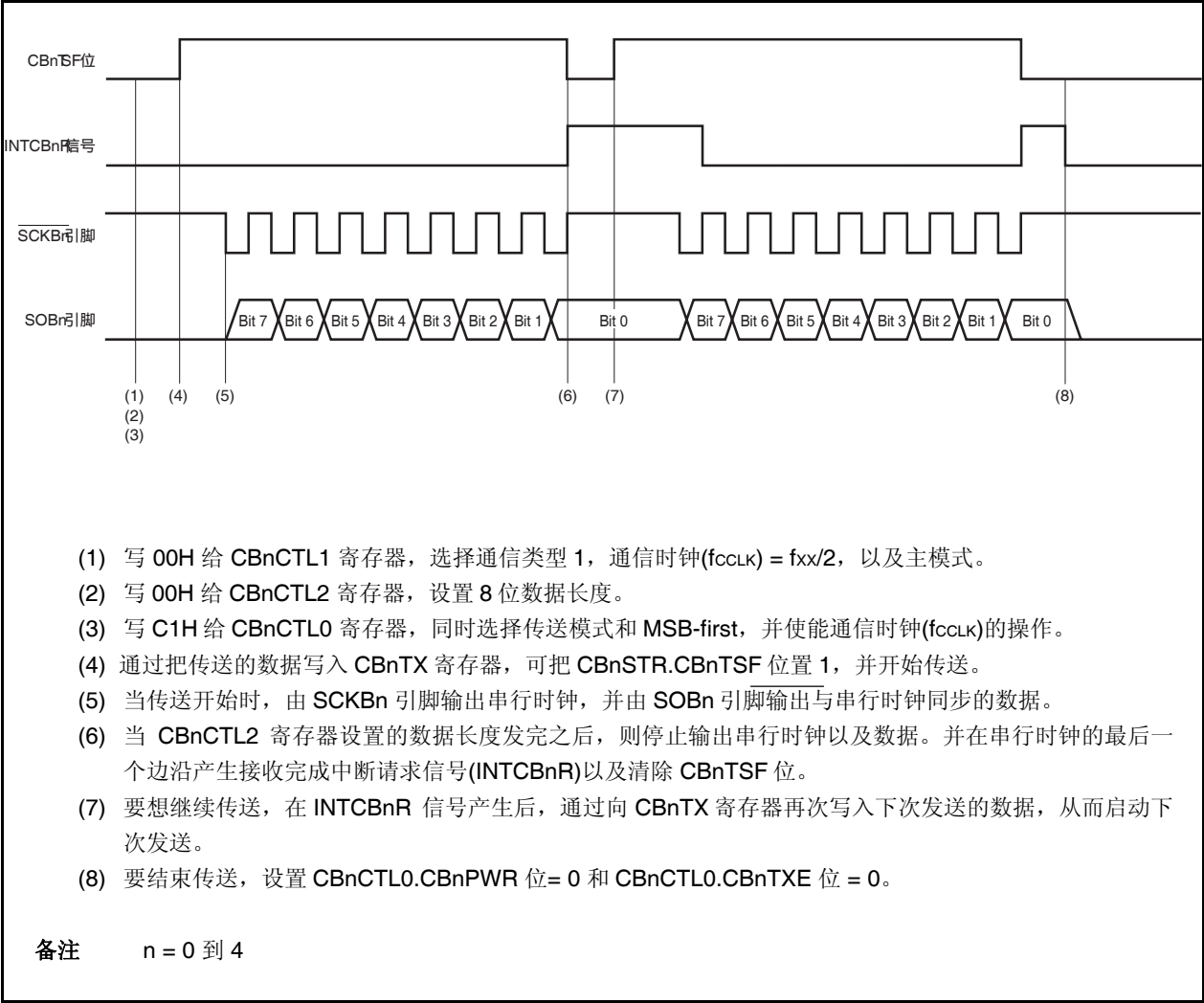
MSB first (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = f_{XX}/2 (CBnCTL1.CBnCKS2 到 CBnCTL1.CBnCKS0 位 = 000), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 到 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



- 备注
1. 虚线表示的是硬件处理过程。
 2. 该图中所对应的处理时序在 (2) 操作时序 中。
 3. n = 0 到 4

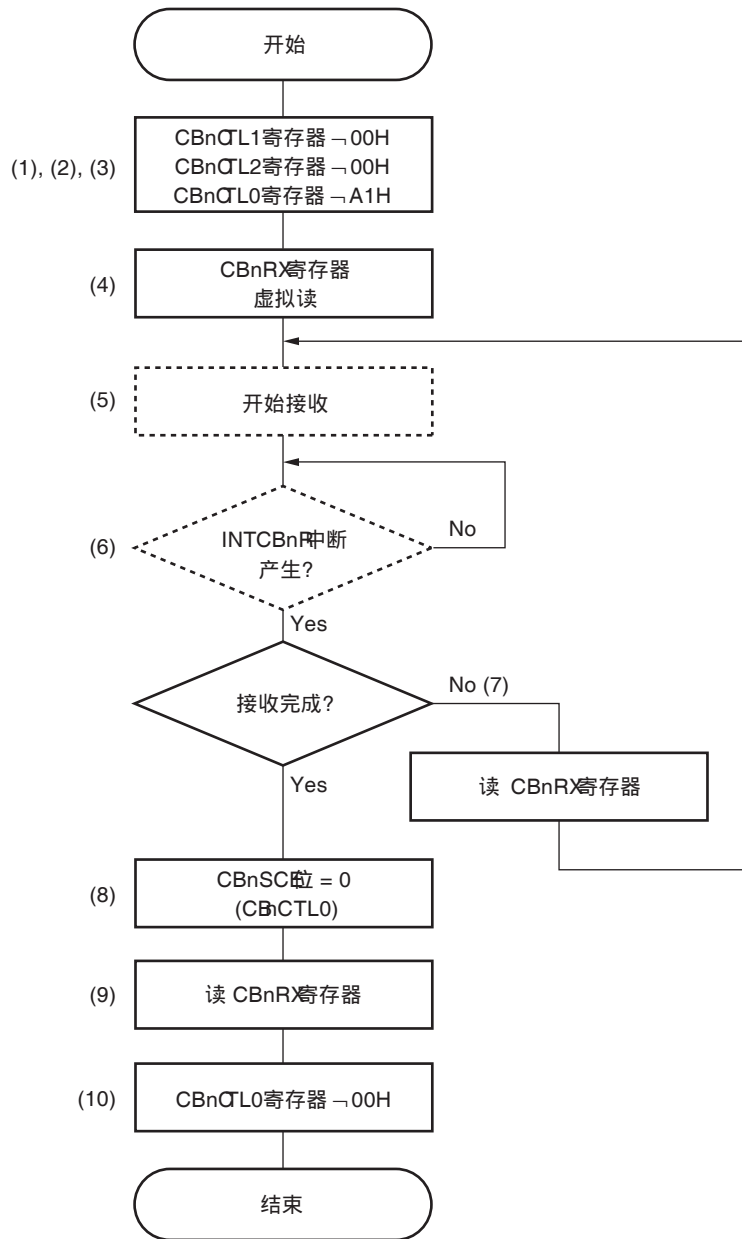
(2) 操作时序



16.6.2 单独传送模式 (主机模式, 接收模式)

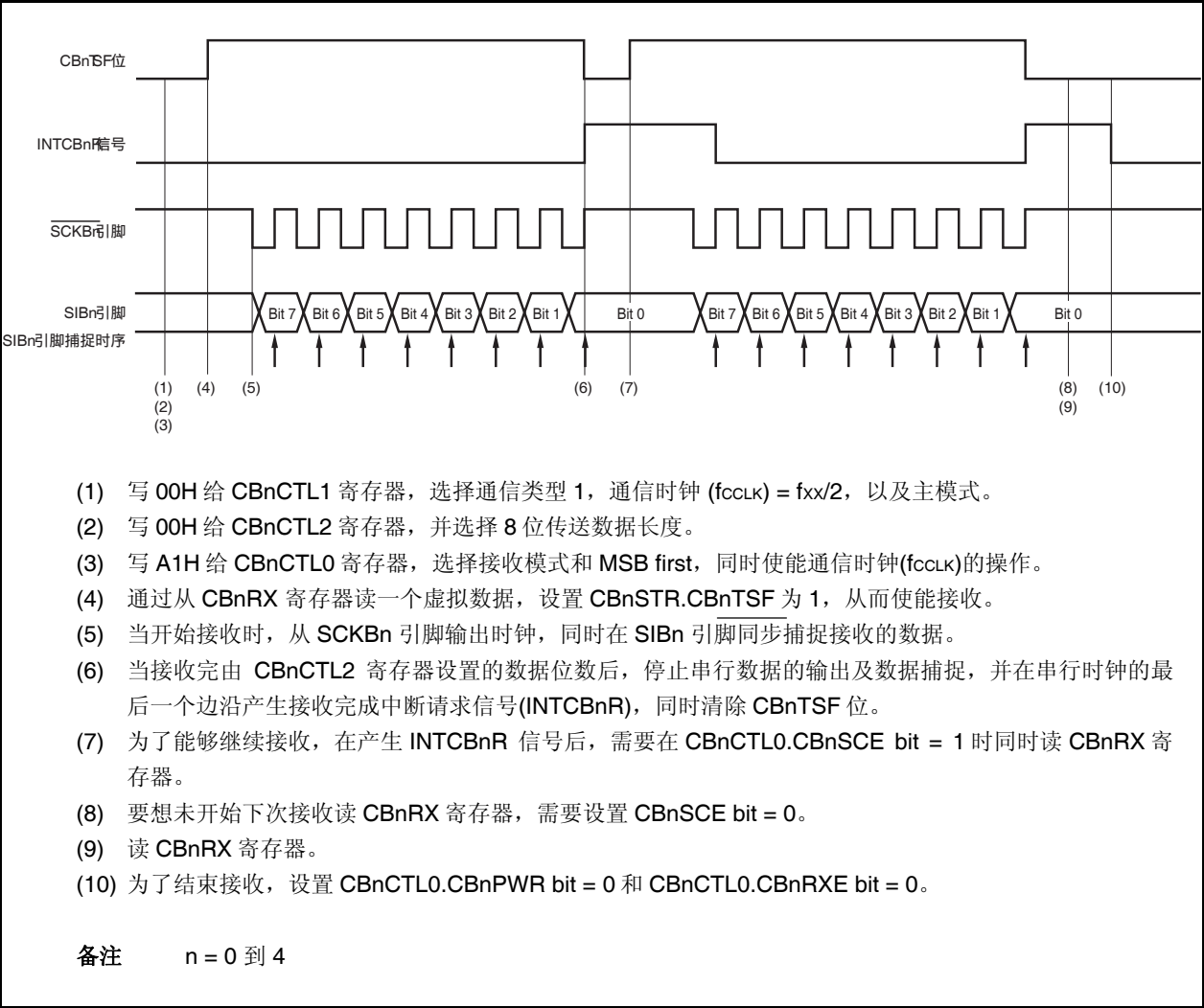
MSB first (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = f_{xx}/2 (CBnCTL1.CBnCKS2 到 CBnCTL1.CBnCKS0 位 = 000), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 到 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



- 备注
1. 虚线表示的是硬件处理过程。
 2. 该图中所对应的处理时序在 (2) 操作时序 中。
 3. n = 0 到 4

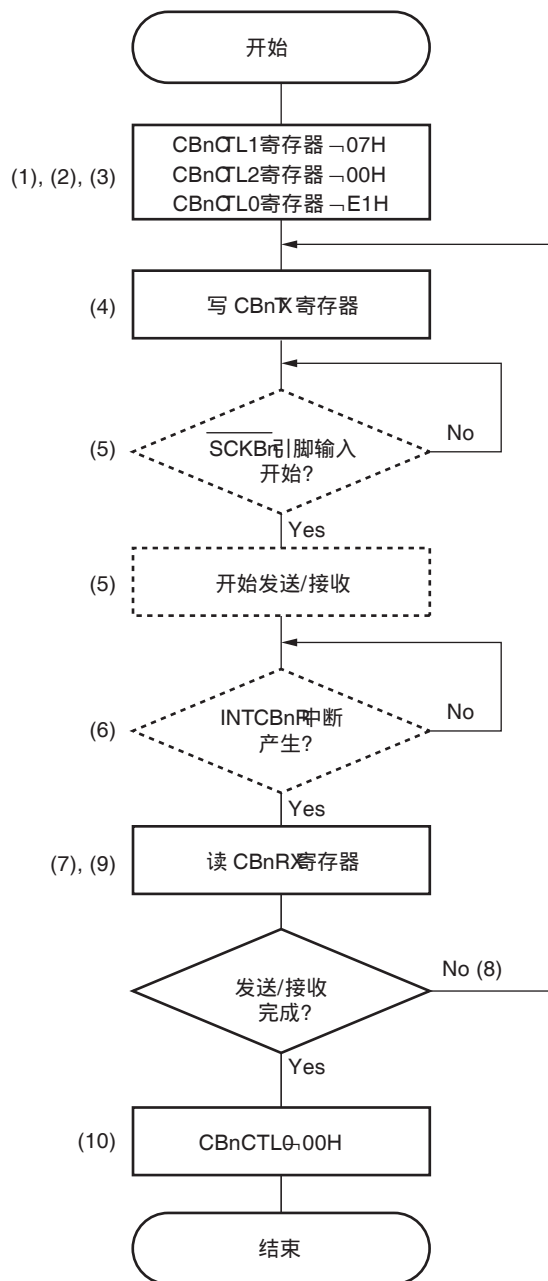
(2) 操作时序



16.6.3 单独传送模式(主机模式，发送/接收模式)

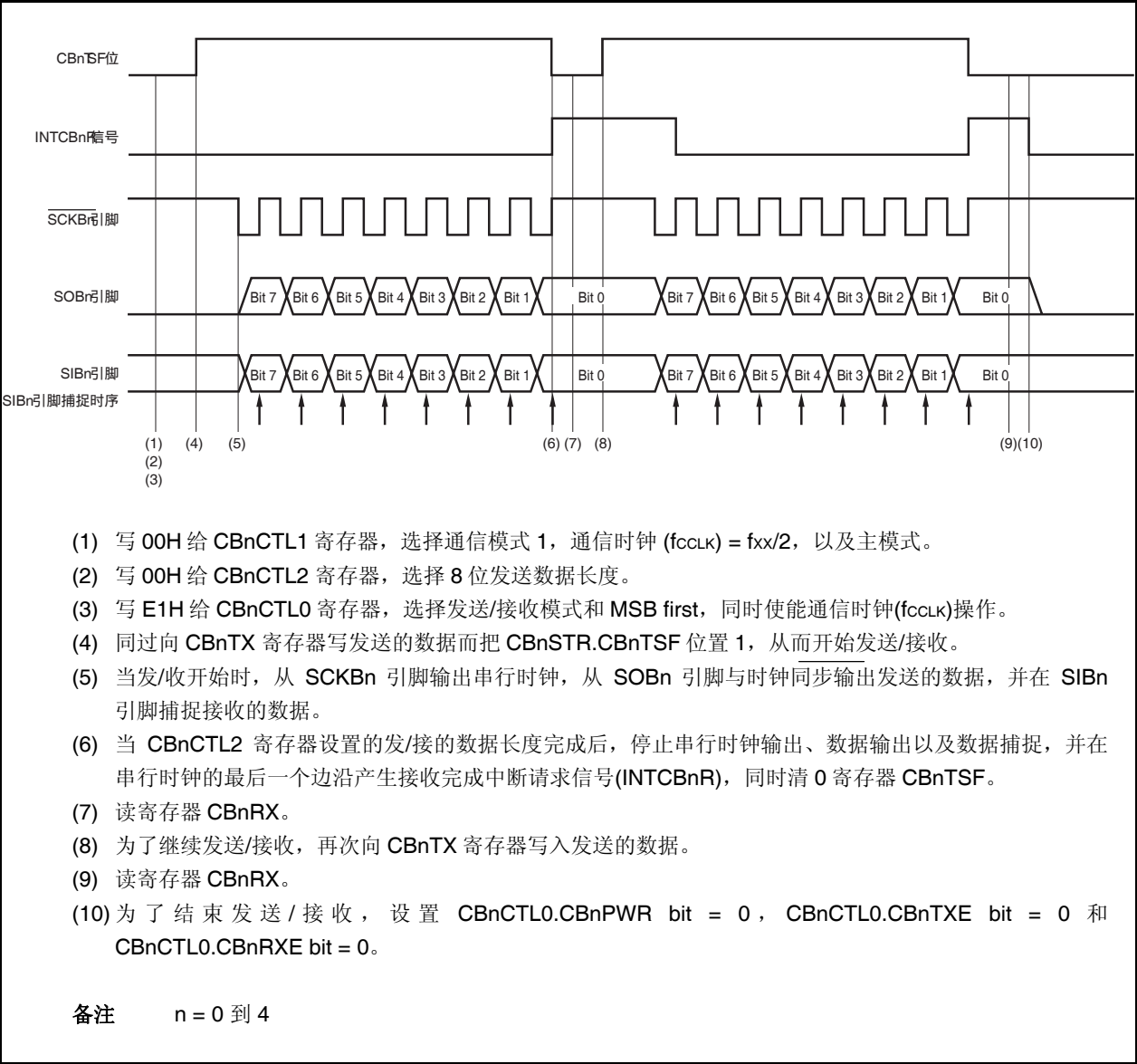
MSB first (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = f_{xx}/2 (CBnCTL1.CBnCKS2 到 CBnCTL1.CBnCKS0 位 = 000), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 到 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



- 备注
1. 虚线表示的是硬件处理过程。
 2. 该图中所对应的处理时序在 (2) 操作时序 中。
 3. n = 0 到 4

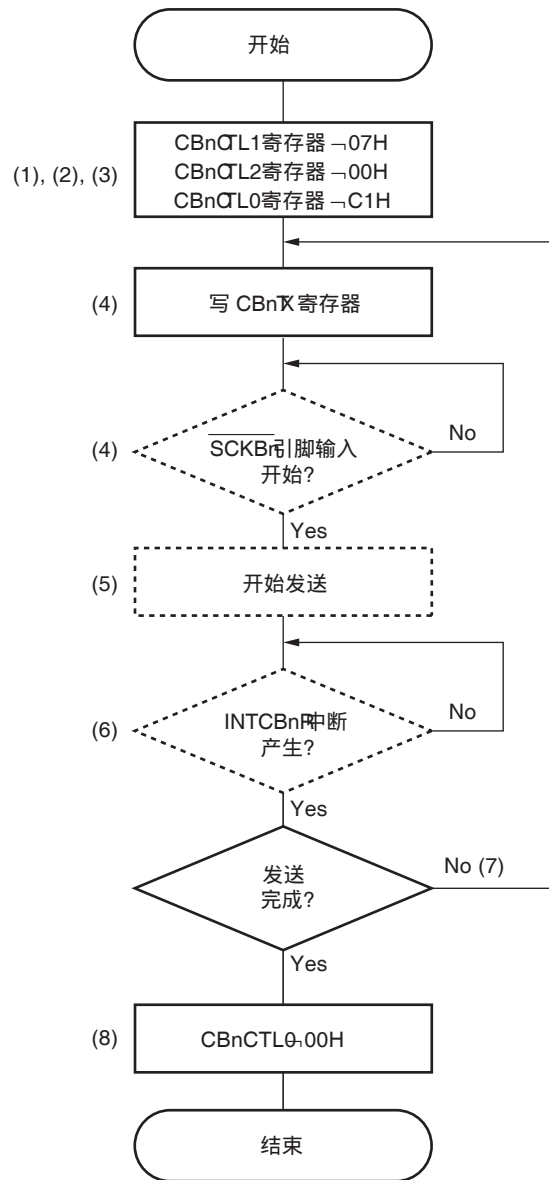
(2) 操作时序



16.6.4 单独传送模式 (从机模式, 发送模式)

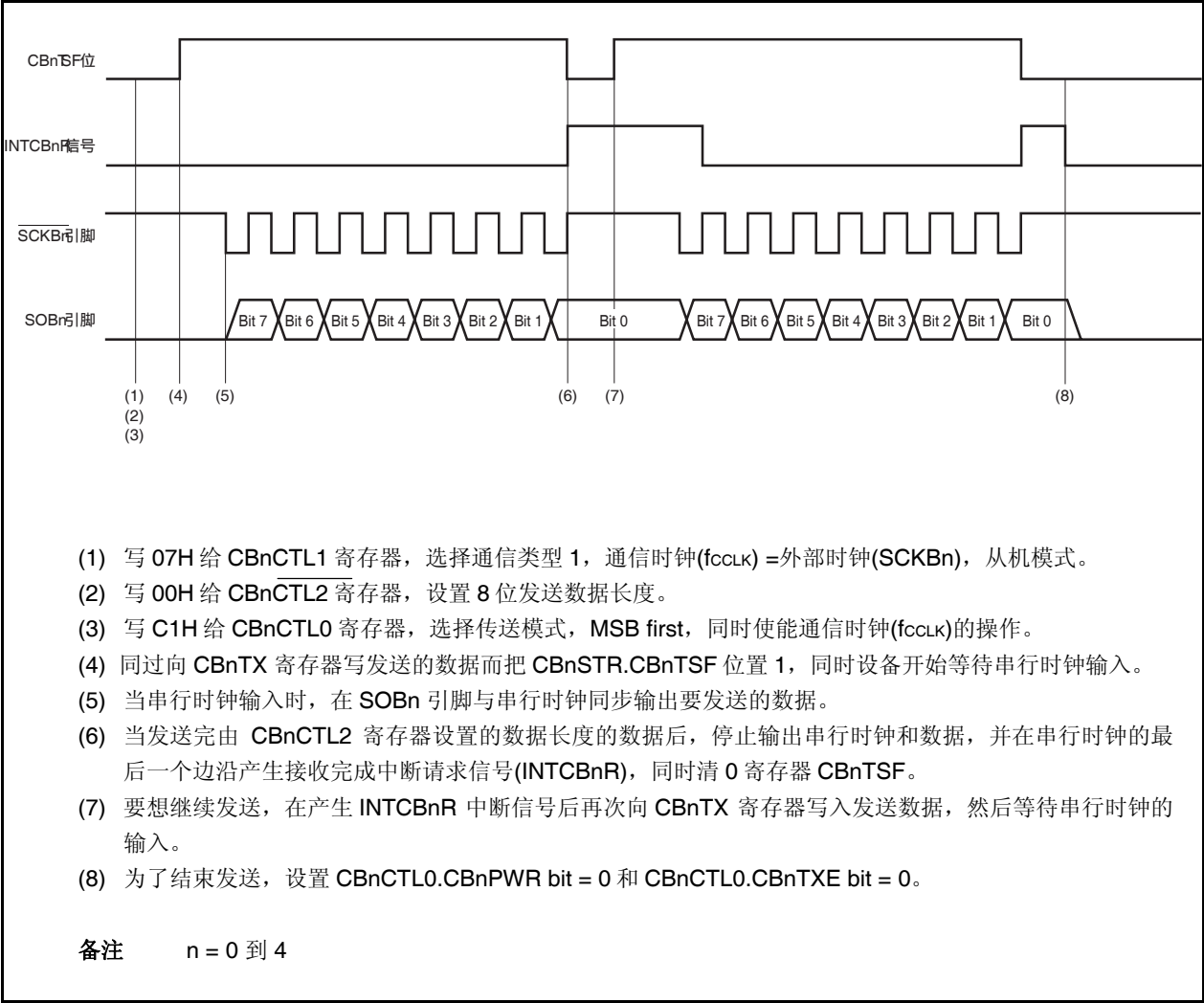
MSB first (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = external clock (SCKBn) (CBnCTL1.CBnCKS2 到 CBnCTL1.CBnCKS0 位 = 111), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 到 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



- 备注**
1. 虚线表示的是硬件处理过程。
 2. 该图中所对应的处理时序在 (2) 操作时序 中。
 3. n = 0 到 4

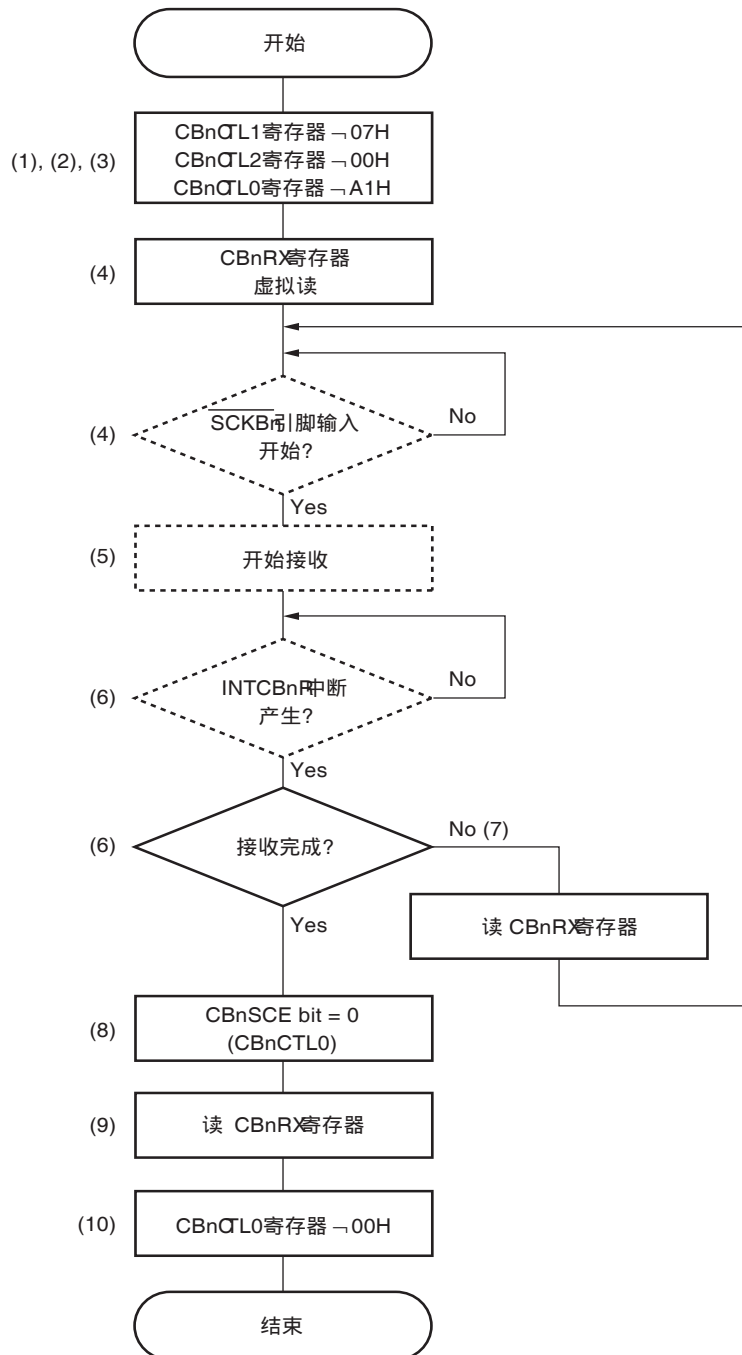
(2) 操作时序



16.6.5 单独传送模式（从机模式，接收模式）

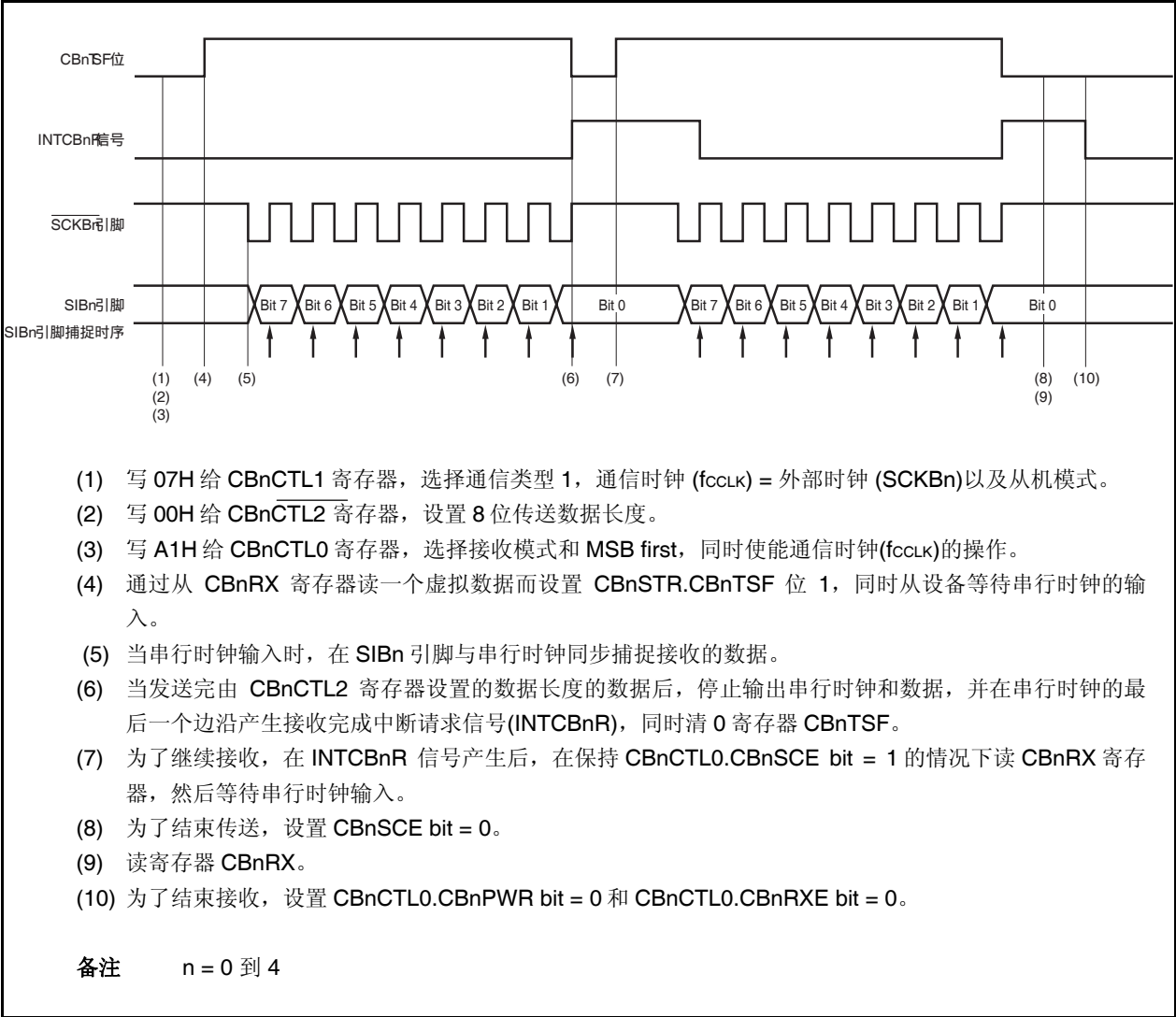
MSB first (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = external clock (SCKBn) (CBnCTL1.CBnCKS2 到 CBnCTL1.CBnCKS0 位 = 111), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 到 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



- 备注
1. 虚线表示的是硬件处理过程。
 2. 该图中所对应的处理时序在 (2) 操作时序 中。
 3. n = 0 到 4

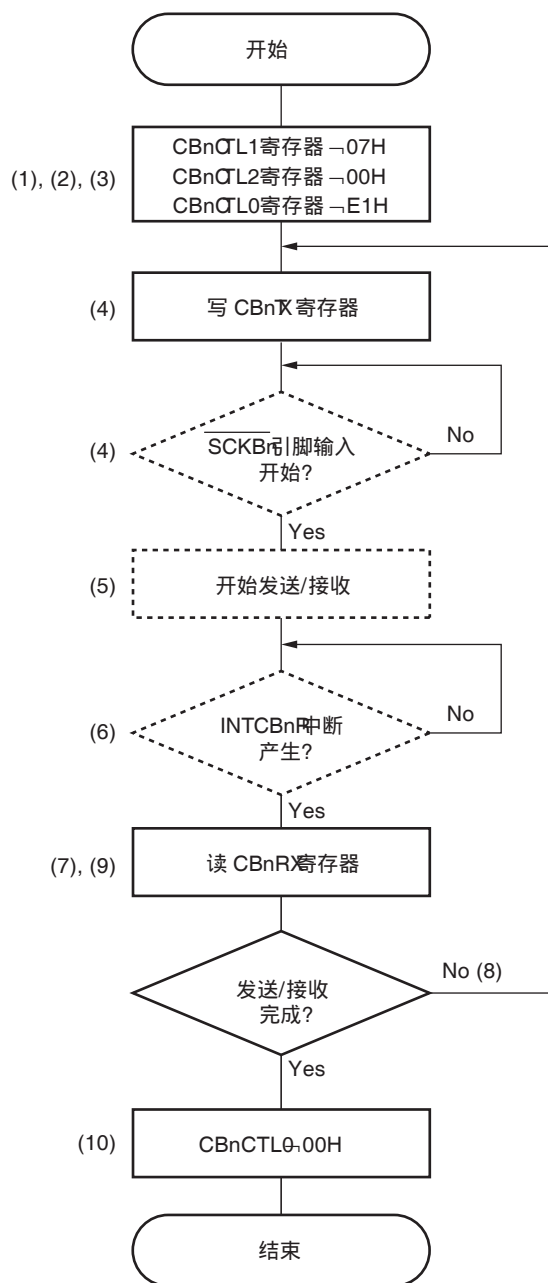
(2) 操作时序



16.6.6 单独传送模式(从机模式，发送/接收模式)

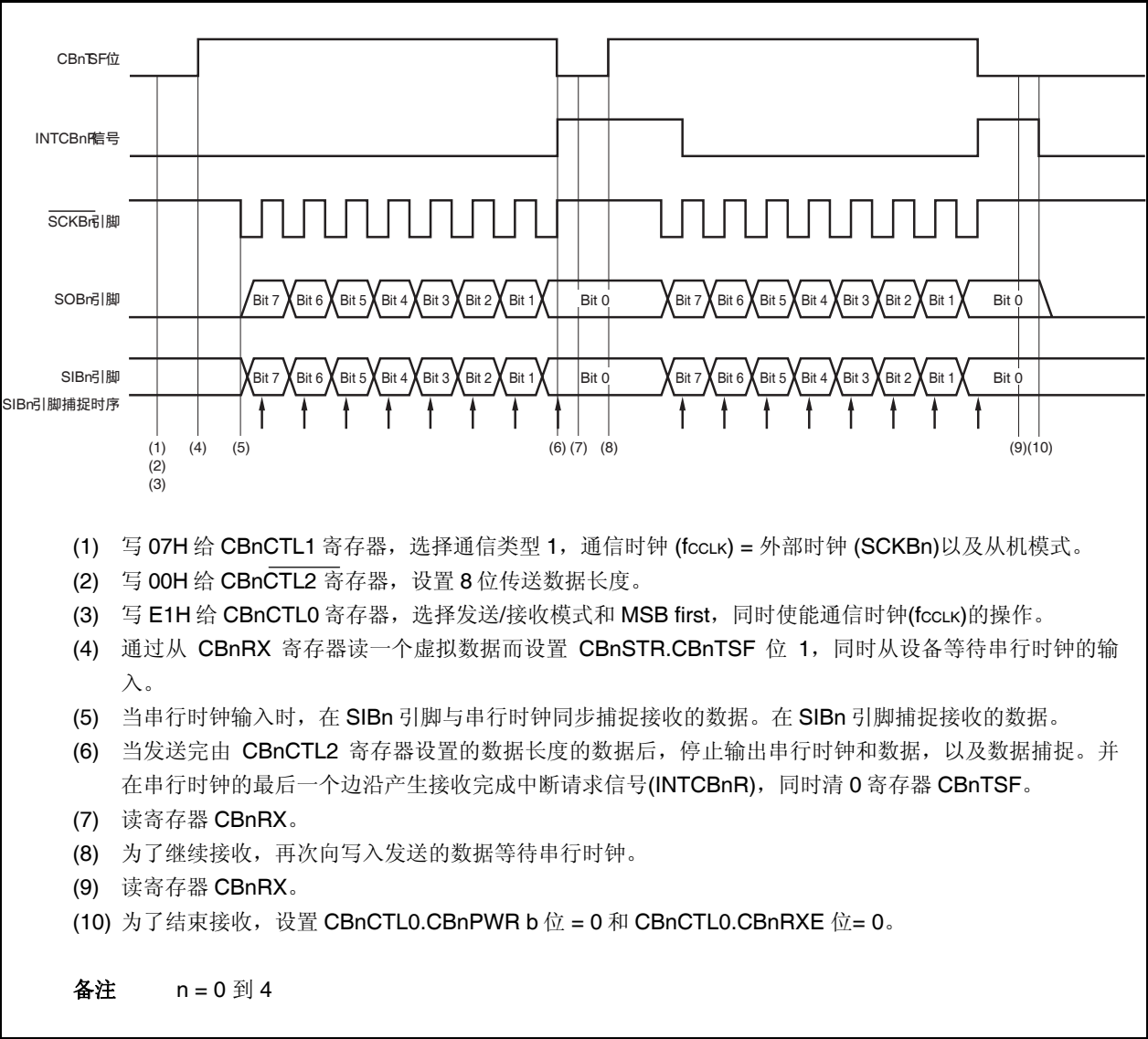
MSB first (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = external clock (SCKBn) (CBnCTL1.CBnCKS2 到 CBnCTL1.CBnCKS0 位 = 111), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 到 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



- 备注
1. 虚线表示的是硬件处理过程。
 2. 该图中所对应的处理时序在 (2) 操作时序 中。
 3. n = 0 到 4

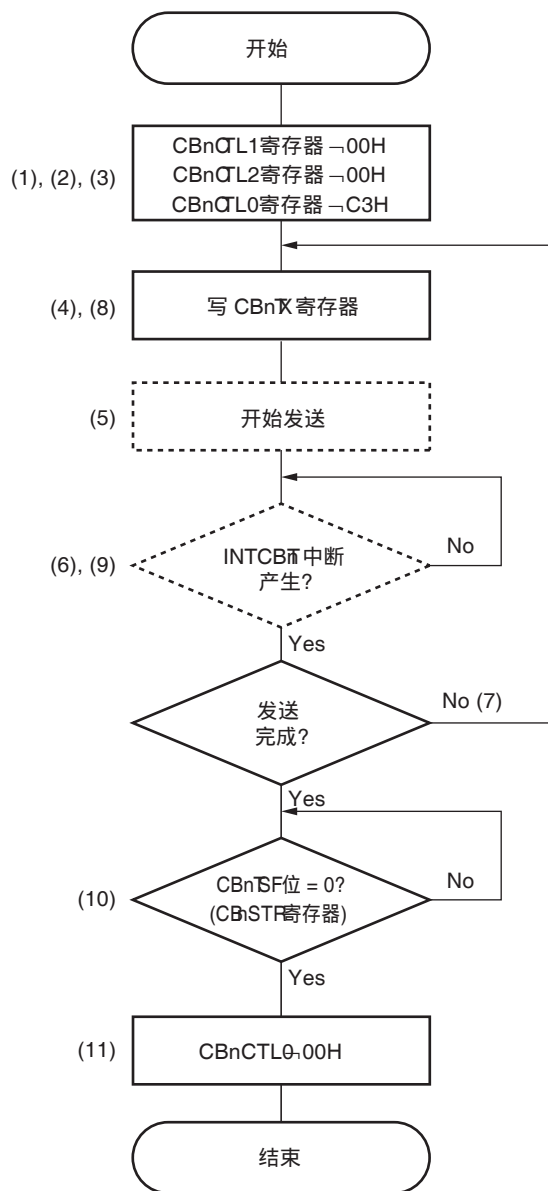
(2) 操作时序



16.6.7 连续传送模式(主机模式, 发送模式)

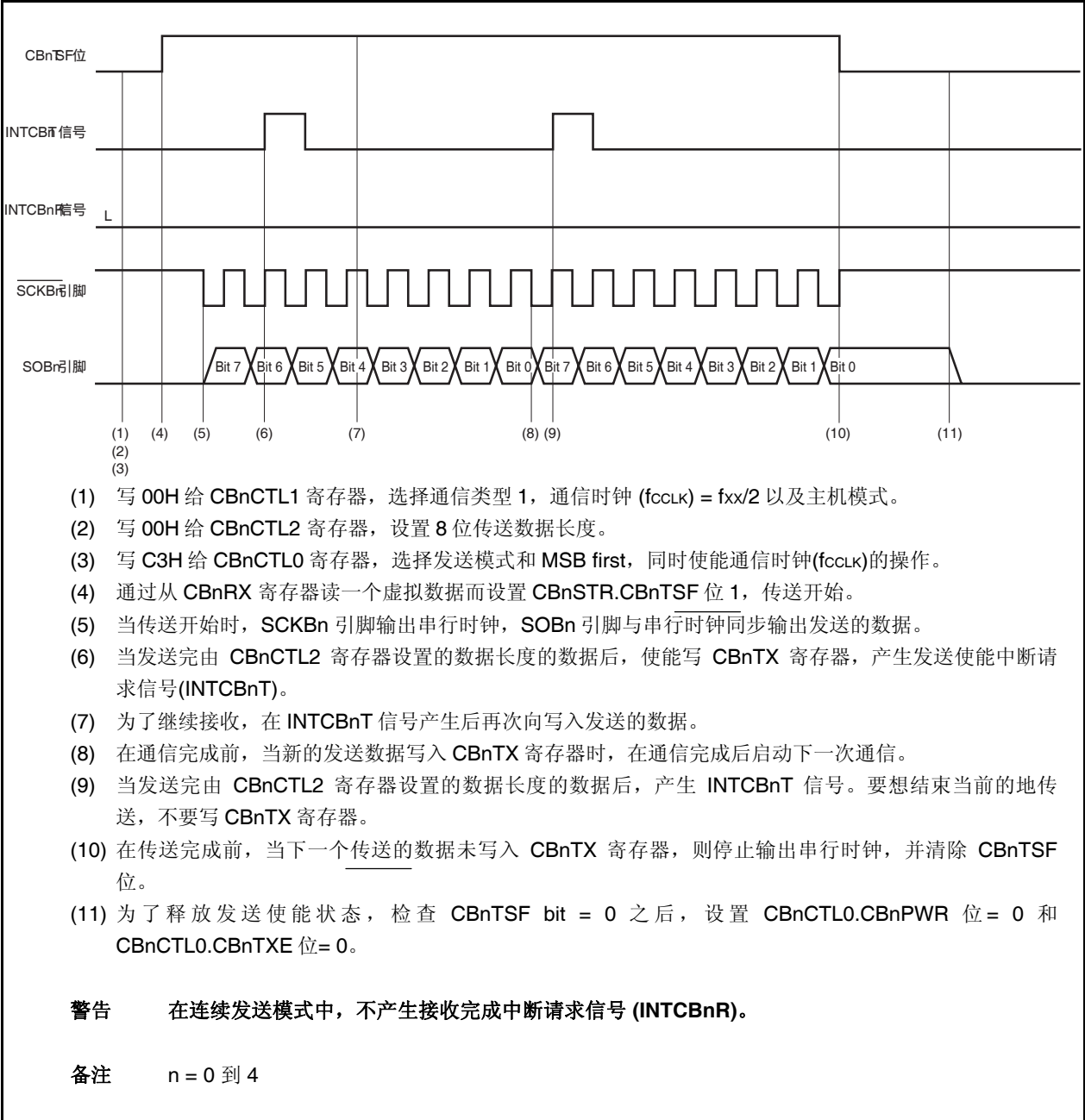
MSB first (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = f_{xx}/2 (CBnCTL1.CBnCKS2 到 CBnCTL1.CBnCKS0 位 = 000), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 到 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



- 备注
1. 虚线表示的是硬件处理过程。
 2. 该图中所对应的处理时序在 (2) 操作时序 中。
 3. n = 0 到 4

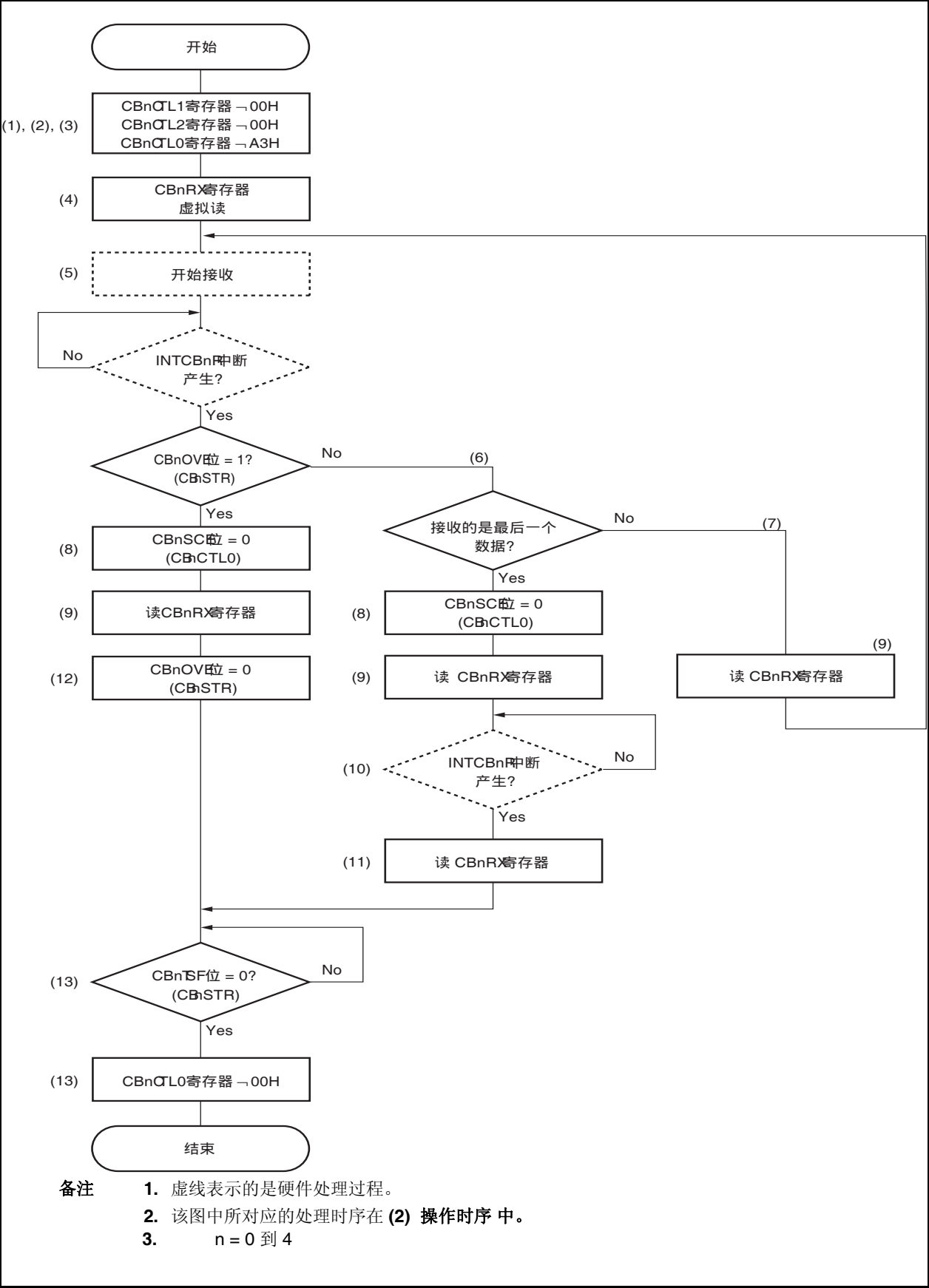
(2) 操作时序



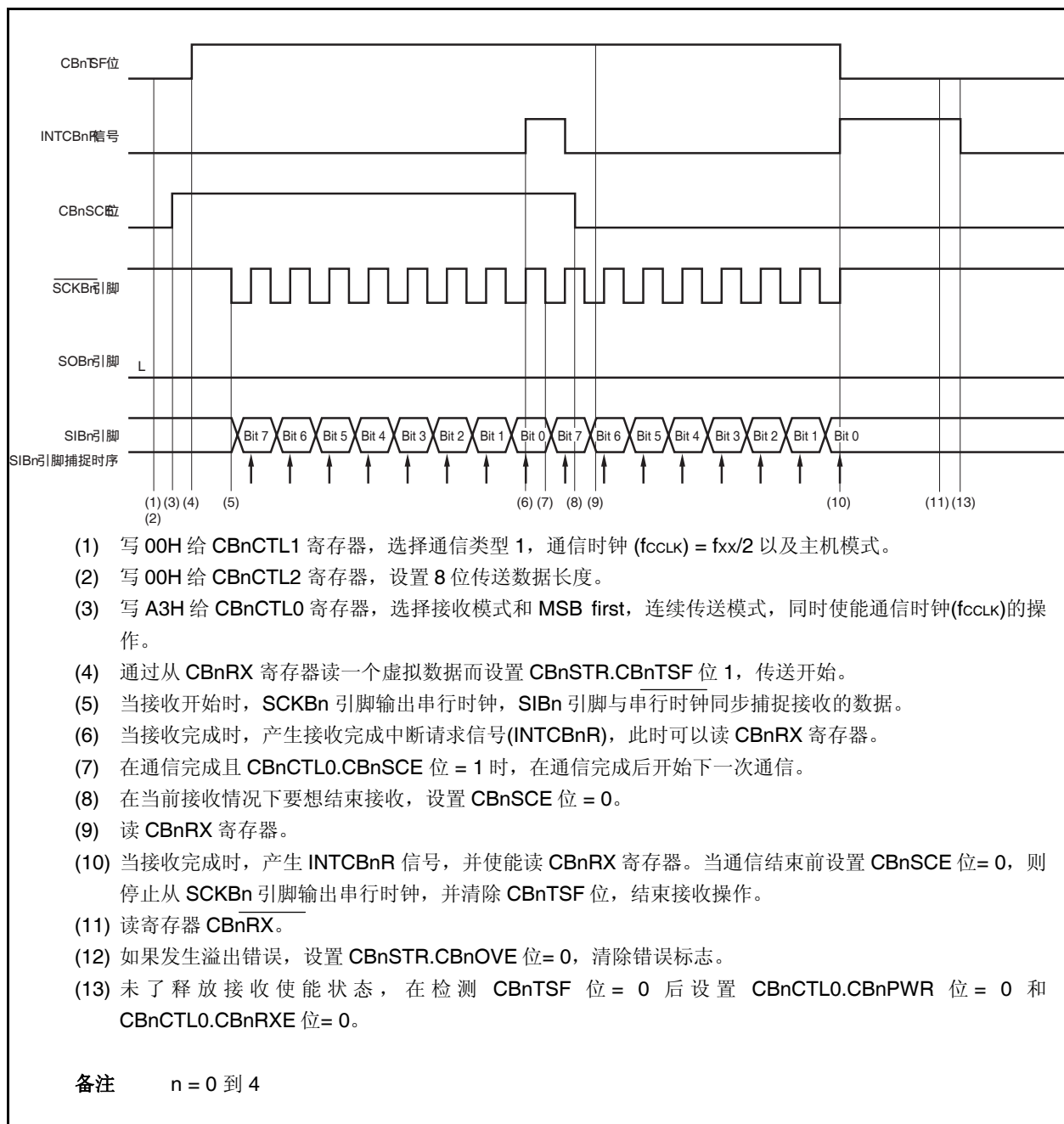
16.6.8 连续传送模式(主机模式，接收模式)

MSB first (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = $f_{\text{xx}}/2$ (CBnCTL1.CBnCKS2 到 CBnCTL1.CBnCKS0 位 = 000), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 到 CBnCTL2.CBnCL0 位 = 0000)。

(1) 操作流程



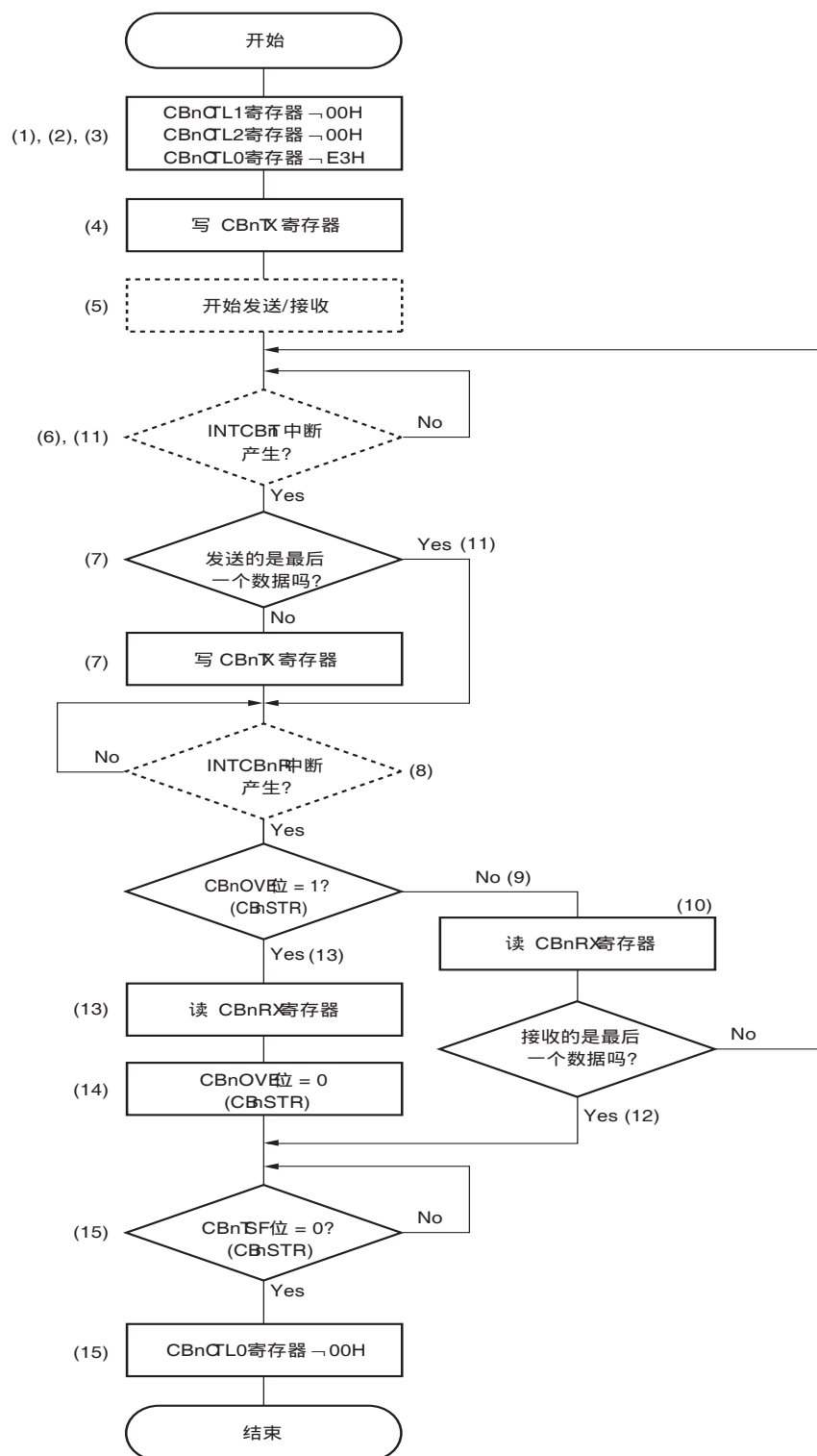
(2) 操作时序



16.6.9 连续传送模式 (主机模式, 发送/接收模式)

MSB first (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = $f_{\text{xx}}/2$ (CBnCTL1.CBnCKS2 到 CBnCTL1.CBnCKS0 位 = 000), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 到 CBnCTL2.CBnCL0 位 = 0000)。

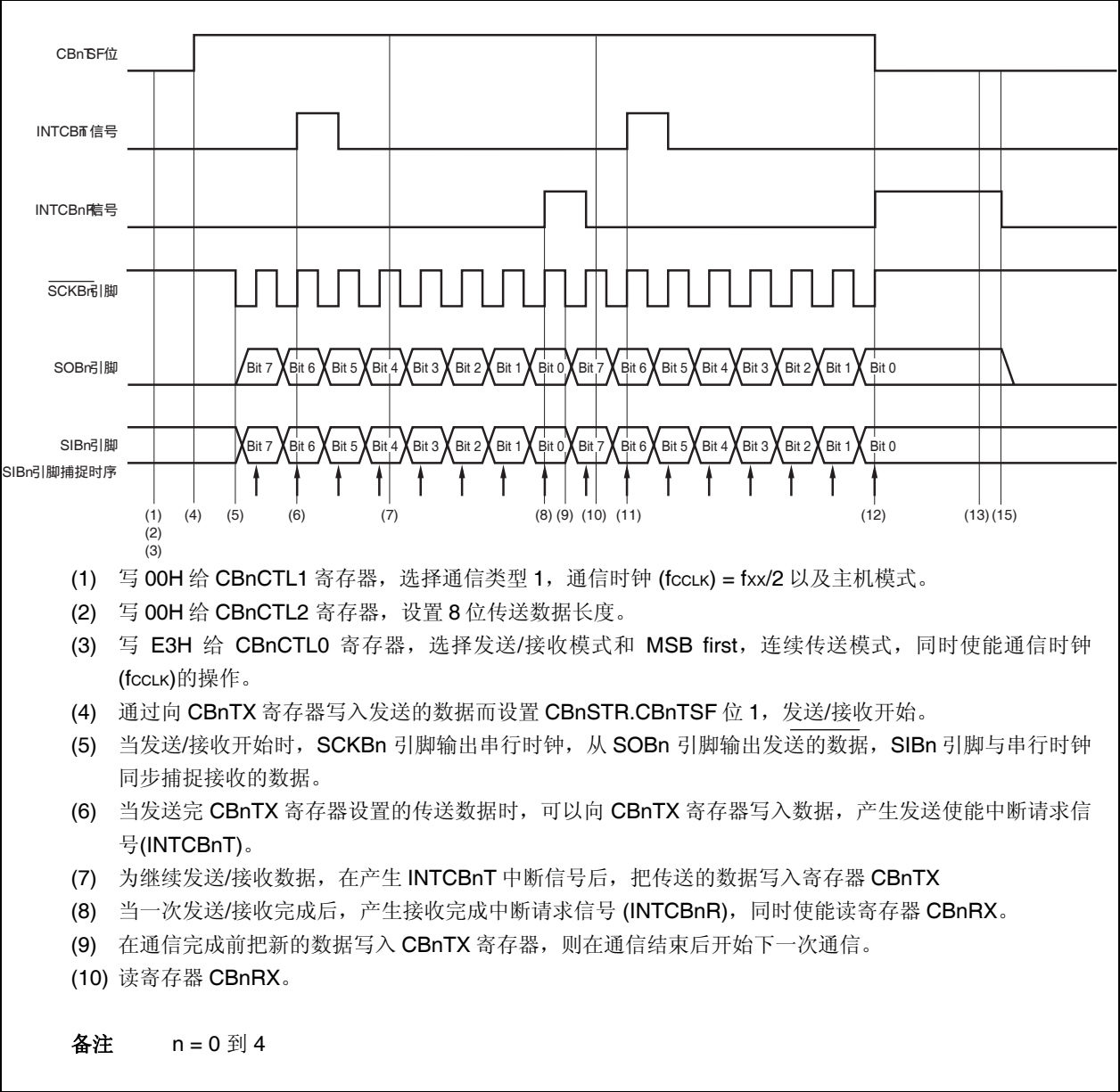
(1) 操作流程



- 备注
1. 虚线表示的是硬件处理过程。
 2. 该图中所对应的处理时序在 (2) 操作时序 中。
 3. $n = 0$ 到 4

(2) 操作时序

(1/2)



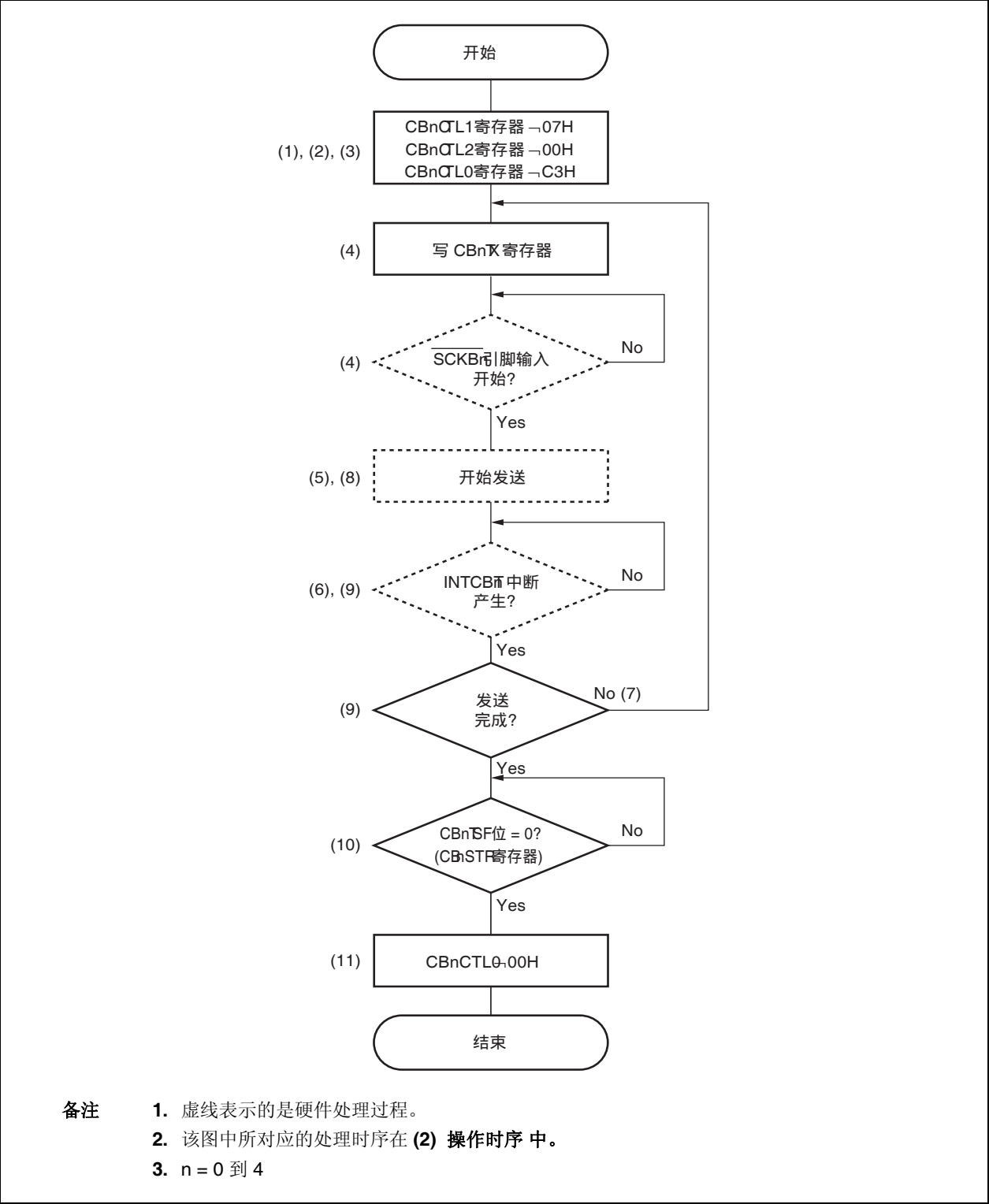
- (11) 当 CBnTX 寄存器中要发送的数据都转移到移位寄存器中时，产生 INTCBnT 信号。为了结束连续发送/接收，不要向 CBnTX 寄存器写入数据。
- (12) 在传送结束前如果没有向 CBnTX 寄存器写入下次发送的数据，这时在传送结束后 SCKBn 引脚就停止输出串行时钟，并清除 CBnTSF 位。
- (13) 当产生接收错误中断请求信号时，读寄存器 CBnRX。
- (14) 如果发生溢出错误，设置 CBnSTR.CBnOVE 位= 0，并清除错误标志。
- (15) 为了释放发送/接收使能状态，在检查 CBnTSF 位= 0 后，设置 CBnCTL0.CBnPWR 位= 0，CBnCTL0.CBnTXE 位= 0，以及 CBnCTL0.CBnRXE 位= 0。

备注 n = 0 到 4

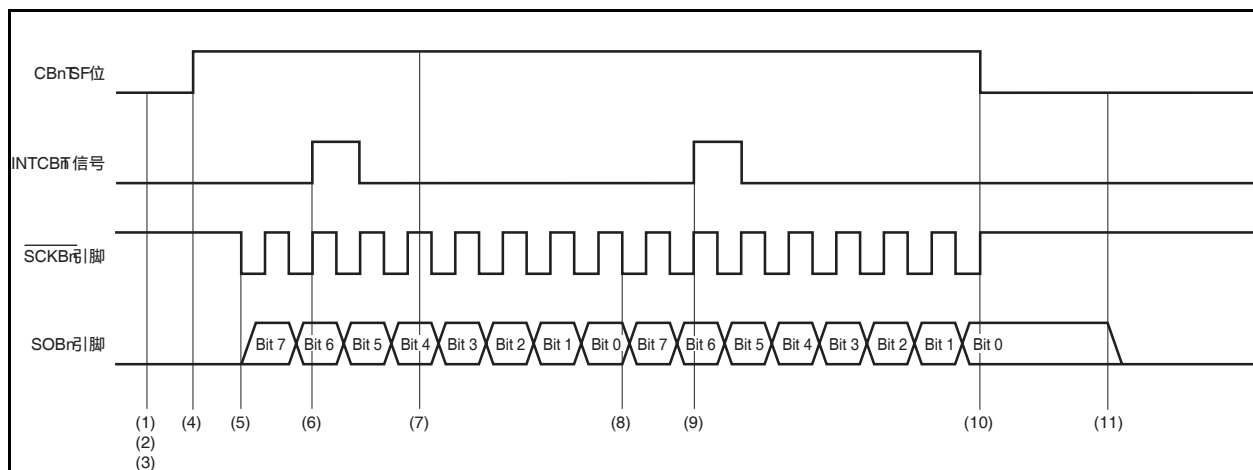
16.9.10 连续传送模式(从机模式,发送模式)

MSB first (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00), 通信时钟 (f_{CLK}) = external clock (SCKBn) (CBnCTL1.CBnCKS2 到 CBnCTL1.CBnCKS0 位 = 111), 发送数据长度= 8 位 (CBnCTL2.CBnCL3 到 CBnCTL2.CBnCL0 位 = 0000)。

(1) 操作流程



(2) 操作时序



- (1) 写 07H 给 CBnCTL1 寄存器，选择通信类型 1，通信时钟 (f_{CCLK}) = external clock (SCKBn)以及从机模式。
- (2) 写 00H 给 CBnCTL2 寄存器，设置 8 位传送数据长度。
- (3) 写 C3H 给 CBnCTL0 寄存器，选择发送模式和 MSB first，连续传送模式，同时使能通信时钟(f_{CCLK})的操作。
- (4) 通过向 CBnTX 寄存器写入发送的数据而设置 CBnSTR.CBnTSF 位 1，从设备等待串行时钟的输入。
- (5) 当串行时钟输入时，从 SOBn 引脚与串行时钟同步输出发送的数据。
- (6) 当发送完 CBnTX 寄存器设置的传送数据时，可以向 CBnTX 寄存器写入数据，产生发送使能中断请求信号(INTCBnT)。
- (7) 为继续发送数据，在产生 INTCBnT 中断信号后，把传送的数据写入寄存器 CBnTX。
- (8) 在 CBnCTL2 寄存器设置的发送数据的长度完成后再输入串行时钟时，则开始连续发送。
- (9) 当 CBnTX 中要传送的数据完全移到移位寄存器中时，使能写 CBnTX 寄存器，并产生 INTCBnT 中断信号。为结束当前发送的连续发送，不要向 CBnTX 寄存器写数据。
- (10) 当未向 CBnTX 寄存器写数据而输入 CBnCTL2 寄存器设置的数据长度的时钟时，清除 CBnTSF 位，以结束发送。
- (11) 为释放发送使能状态，检查 CBnTSF 位= 0 后设置 CBnCTL0.CBnPWR 位= 0 和 CBnCTL0.CBnTXE 位= 0。

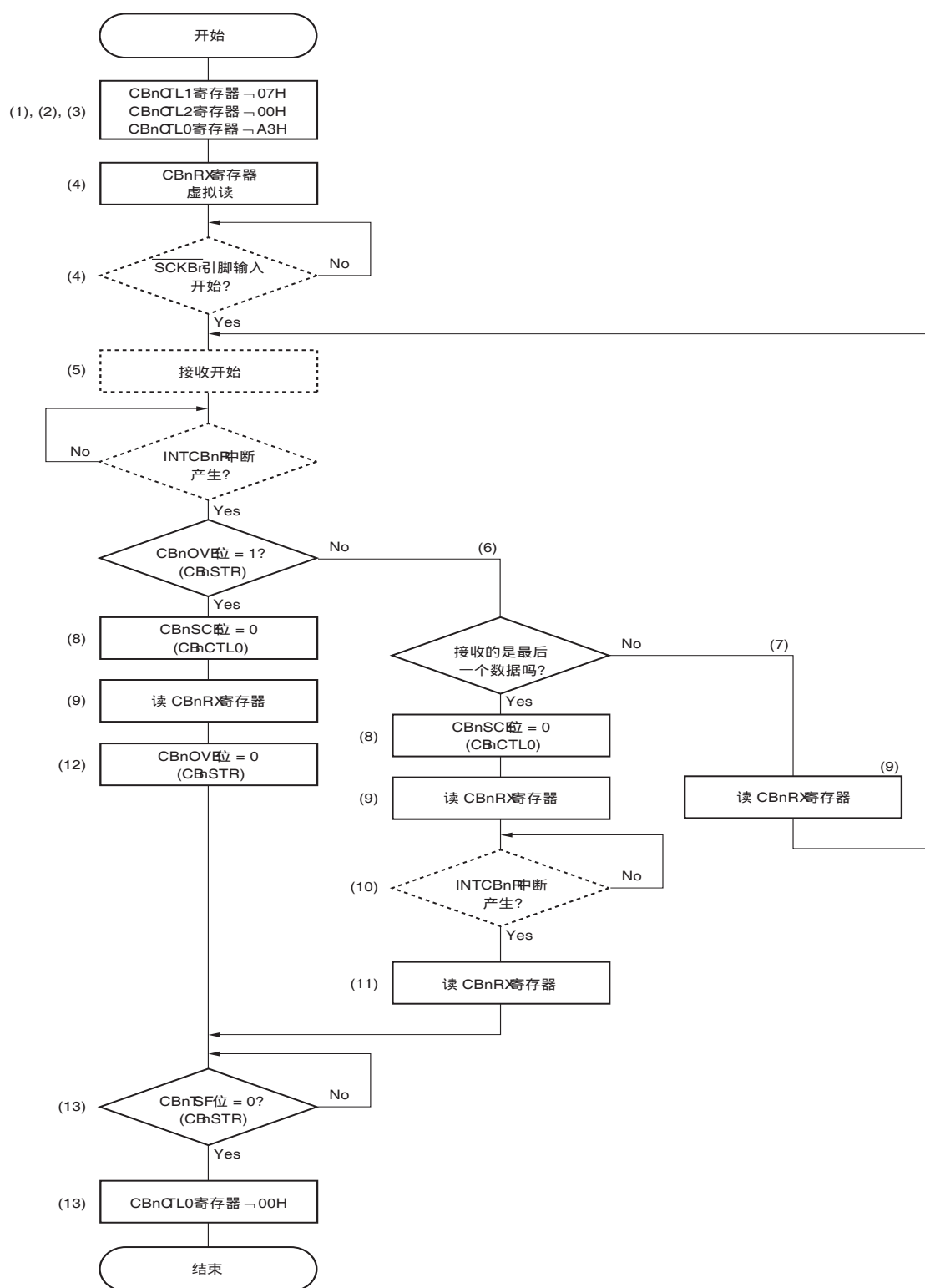
警告 在连续发送模式中，不产生接收完成中断请求信号 (INTCBnR)。

备注 n = 0 到 4

16.6.11 连续传送模式 (从机模式, 接收模式)

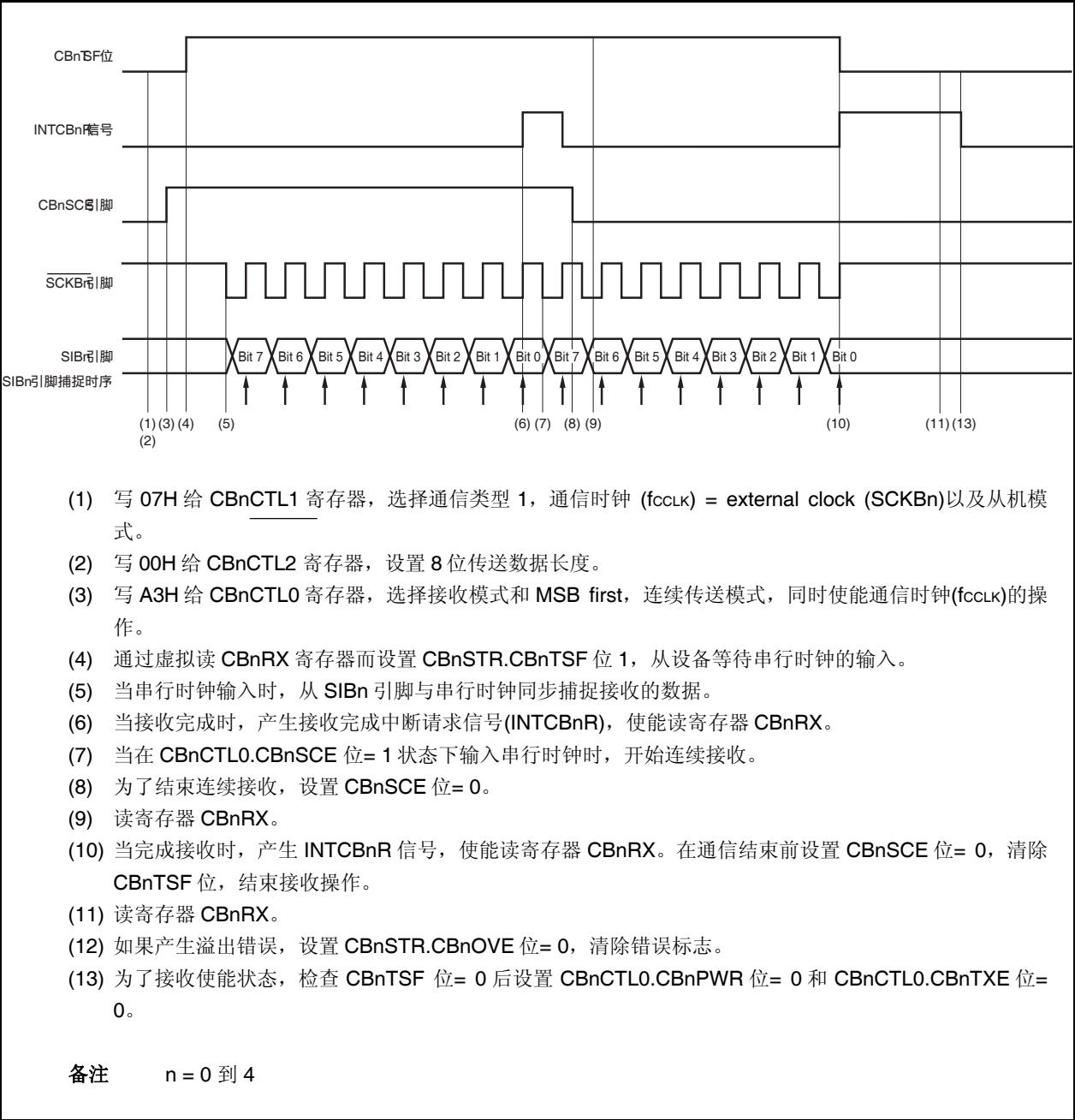
MSB first (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = external clock (SCKBn) (CBnCTL1.CBnCKS2 到 CBnCTL1.CBnCKS0 位 = 111), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 到 CBnCTL2.CBnCL0 位 = 0000)。

(1) 操作流程



- 备注**
1. 虚线表示的是硬件处理过程。
 2. 该图中所对应的处理时序在 (2) 操作时序 中。
 3. $n = 0$ 到 4

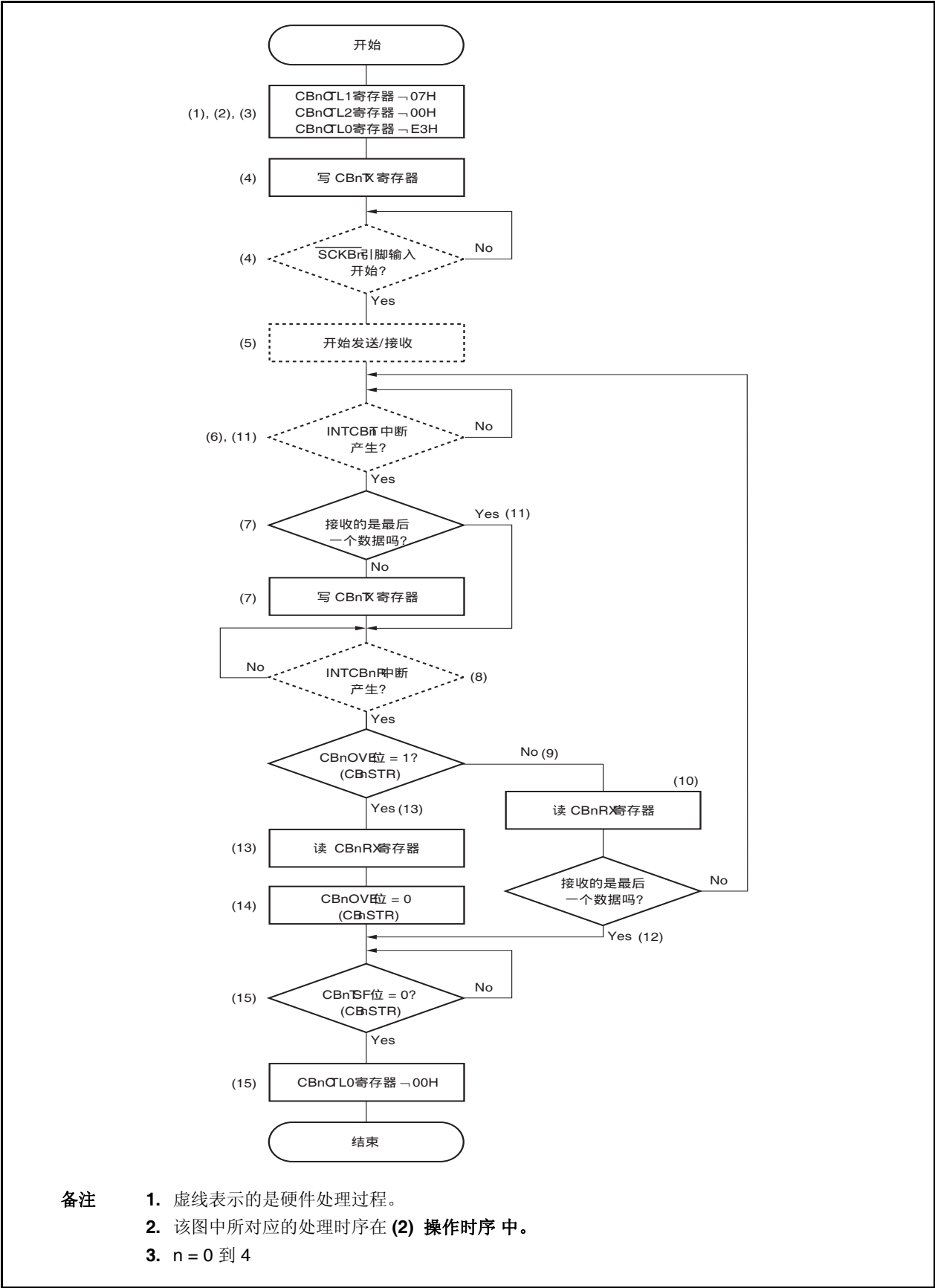
(2) 操作时序



16.6.12 连续传送模式(从机模式，发送/接收模式)

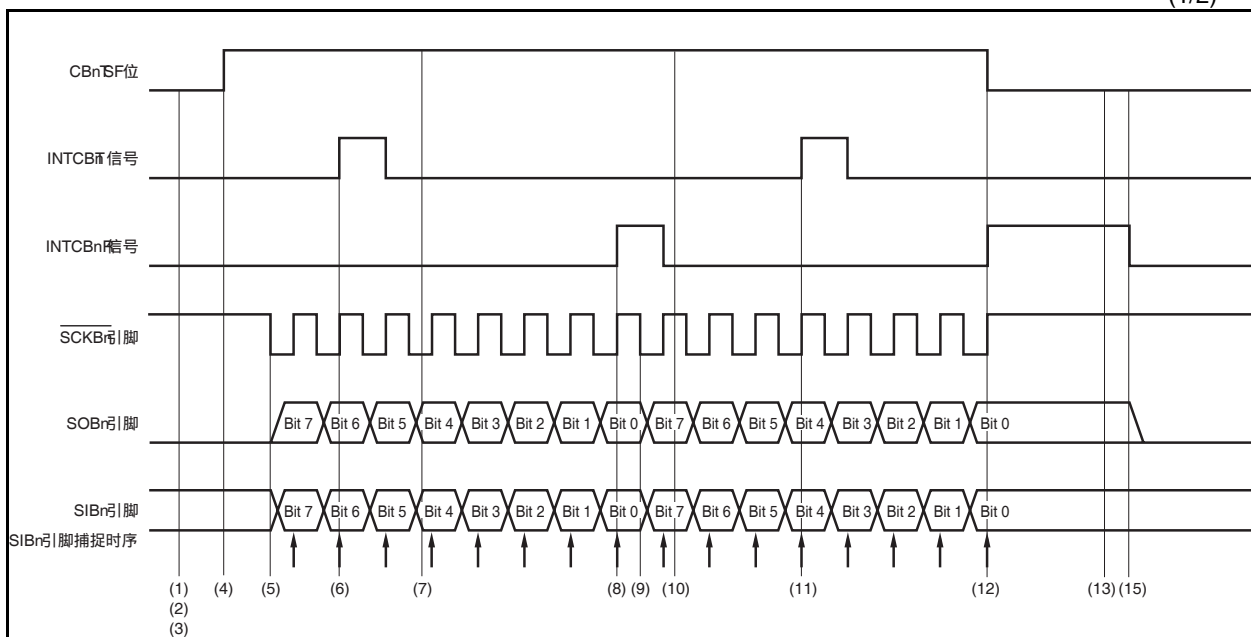
MSB first (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00), 通信时钟 (f_{CCLK}) = external clock (SCKBn) (CBnCTL1.CBnCKS2 到 CBnCTL1.CBnCKS0 位 = 111), 发送数据长度= 8 位 (CBnCTL2.CBnCL3 到 CBnCTL2.CBnCL0 位 = 0000)。

(1) 操作流程



(2) 操作时序

(1/2)



- (1) 写 07H 给 `CBnCTL1` 寄存器，选择通信类型 1，通信时钟 (`fcCLK`) = external clock (`SCKBn`)以及从机模式。
- (2) 写 00H 给 `CBnCTL2` 寄存器，设置 8 位传送数据长度。
- (3) 写 E3H 给 `CBnCTL0` 寄存器，选择发送/模式和 **MSB first**，连续传送模式，同时使能通信时钟(`fcCLK`)的操作。
- (4) 通过写数据给 `CBnTX` 寄存器而设置 `CBnSTR.CBnTSF` 位 1，从设备等待串行时钟的输入。
- (5) 当串行时钟输入时，从 `SOBn` 引脚与串行时钟同步输出发送的数据，并在 `SIBn` 引脚捕捉接收的数据。
- (6) 当 `CBnTX` 寄存器中的数据都转移到移位寄存器中时，使能写寄存器 `CBnTX`，并产生发送使能中断请求信号 (`INTCBnT`)。
- (7) 为了连续发送，在产生 `INTCBnT` 信号后再次把发送的数据写入 `CBnTX` 寄存器。
- (8) 在 `CBnCTL2` 寄存器设置的发送数据的长度完成后，产生接收完成中断请求信号(`INTCBnR`)，并使能读寄存器 `CBnRX`。
- (9) 当串行时钟连续输入时，开始连续发送/接收。
- (10) 读寄存器 `CBnRX`。
- (11) 当寄存器 `CBnTX` 中发送的数据完全送到移位寄存器中时，使能写寄存器 `CBnTX`，产生 `INTCBnT` 中断信号。为了结束连续发送/接收状态，不要写寄存器 `CBnTX`。

备注 $n = 0$ 到 4

- (12) 当未写 CBnTX 寄存器输入了由寄存器 CBnTX 设置的传送数据长度的时钟时，产生 INTCBnR 中断信号。清除 CBnTSF 位以结束发送/接收。
- (13) 当产生 INTCBnR 信号时，如寄存器 CBnRX。
- (14) 如果发生溢出错误，设置 CBnSTR.CBnOVE 位= 0，清除错误标志。
- (15) 为了释放发送/接收状态，检查 CBnTSF 位= 0 后设置 CBnCTL0.CBnPWR 位= 0 和 CBnCTL0.CBnTXE 位= 0。

备注 n = 0 到 4

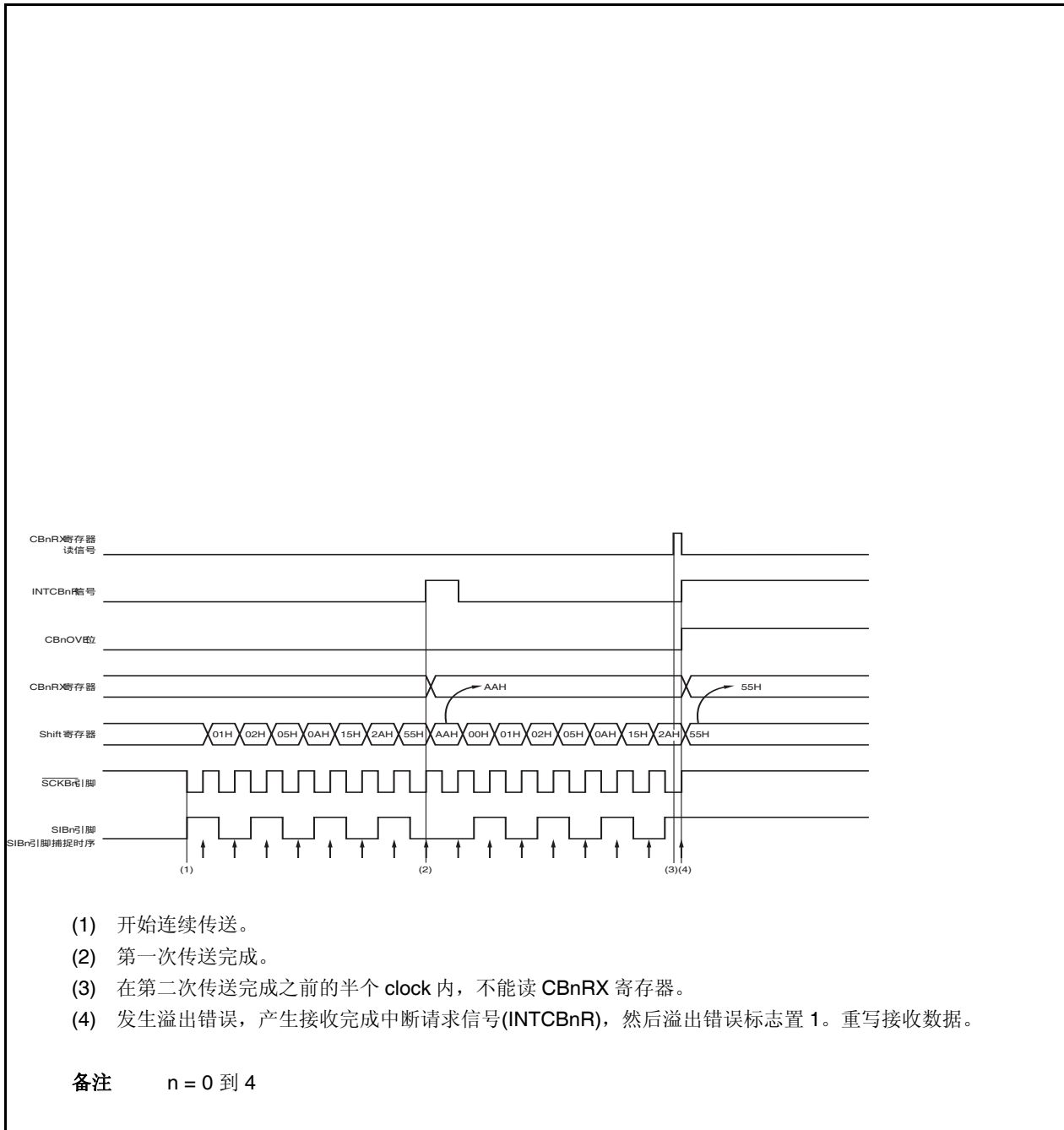
16.6.13 接收错误

在连续传送模式中，在接收使能的情况下执行传送，如果在读 CBnRX 寄存器之前，产生 INTCBnR 中断信号之后，下一次接收操作又完成了，这时会再次产生接收完成中断请求信号。并且溢出错误标志(CBnSTR.CBnOVE)置 1。

如果发生溢出错误，则 CBnRX 寄存器被更新，以前的数据将丢失。如果 CBnRX 寄存器没被读，则在下次接收完成后发生溢出错误，并再次产生中断。

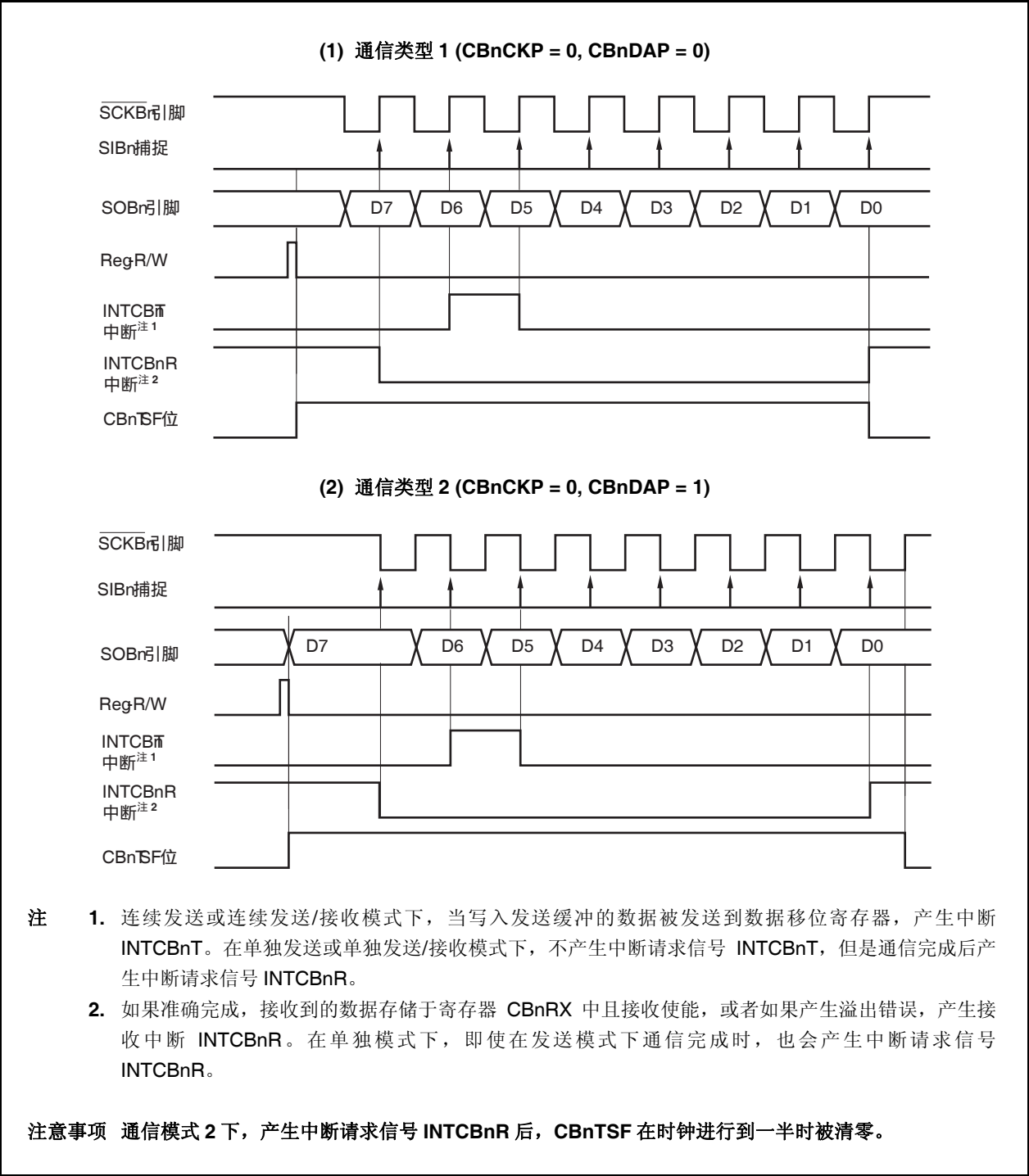
为了避免发生溢出错误，在采样下一个接收数据的最后一位，产生 INTCBnR 中断信号以前的半个时钟之前，要完成对 CBnRX 寄存器的读操作。

(1) 操作时序

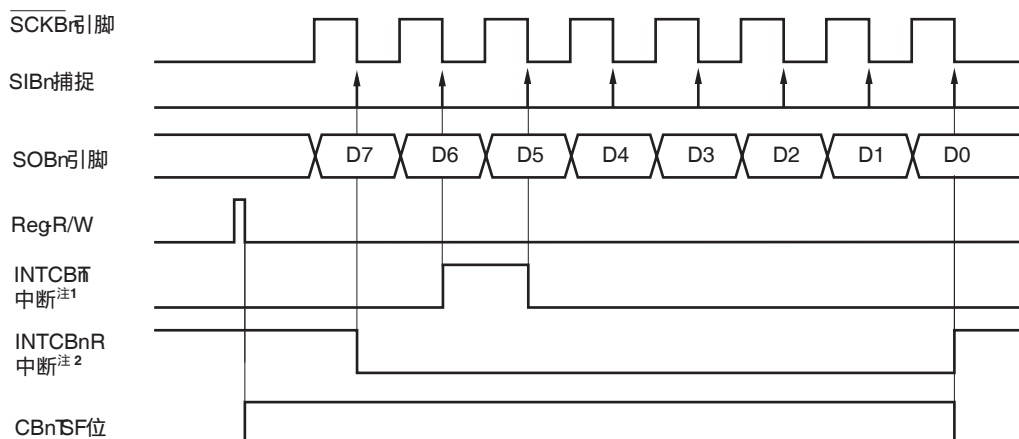


16.6.14 时钟时序

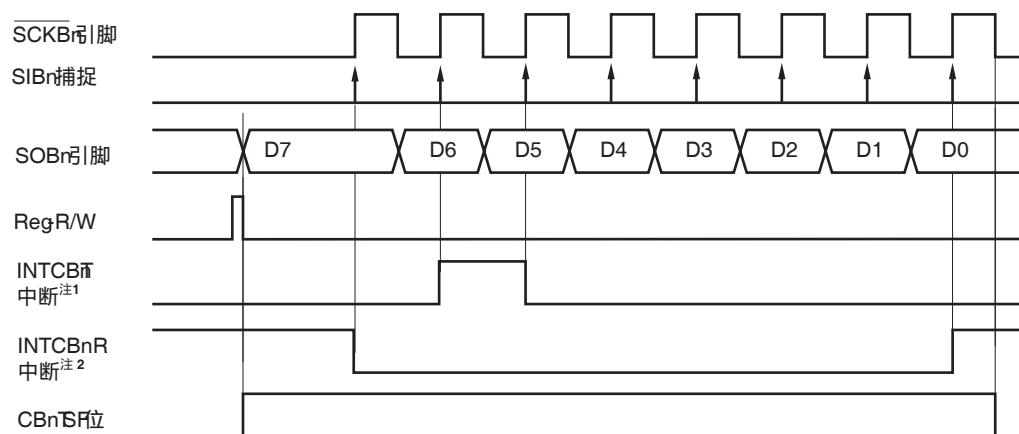
(1/2)



(3) 通信类型 3 (CBnCKP = 1, CBnDAP = 0)



(4) 通信类型 4 (CBnCKP = 1, CBnDAP = 1)



- 注
1. 连续发送或连续发送/接收模式下，当写入发送缓冲的数据被发送到数据移位寄存器，产生中断 INTCBnT。在单独发送或单独发送/接收模式下，不产生中断请求信号 INTCBnT，但是通信完成后产生中断请求信号 INTCBnR。
 2. 如果准确完成，接收到的数据存储在寄存器 CBnRX 中且接收使能，或者如果产生溢出错误，产生接收中断 INTCBnR。在单独模式下，即使在发送模式下通信完成时，也会产生中断请求信号 INTCBnR。

注意事项 通信模式 4 下，产生中断请求信号 INTCBnR 后，CBnTSF 在时钟进行到一半时被清零。

16.7 输出引脚

(1) $\overline{\text{SCKBn}}$ 引脚

当 CSIBn 的操作非使能($\text{CBnCTL0.CBnPWR} = 0$)时, 引脚 $\overline{\text{SCKBn}}$ 的输出状态如下。

CBnCKP	CBnCKS2	CBnCKS1	CBnCKS0	$\overline{\text{SCKBn}}$ 引脚输出
0	1	1	1	高阻态
	除上述情况			固定高电平
1	1	1	1	高阻态
	除上述情况			固定低电平

- 备注**
1. 如果 CBnCTL1.CBnCKP 和 $\text{CBnCKS2} \sim \text{CBnCKS0}$ 被复写, 引脚 $\overline{\text{SCKBn}}$ 的输出电平将改变。
 2. $n = 0$ 至 4
 3. \times : 不用考虑

(2) SOBn 引脚

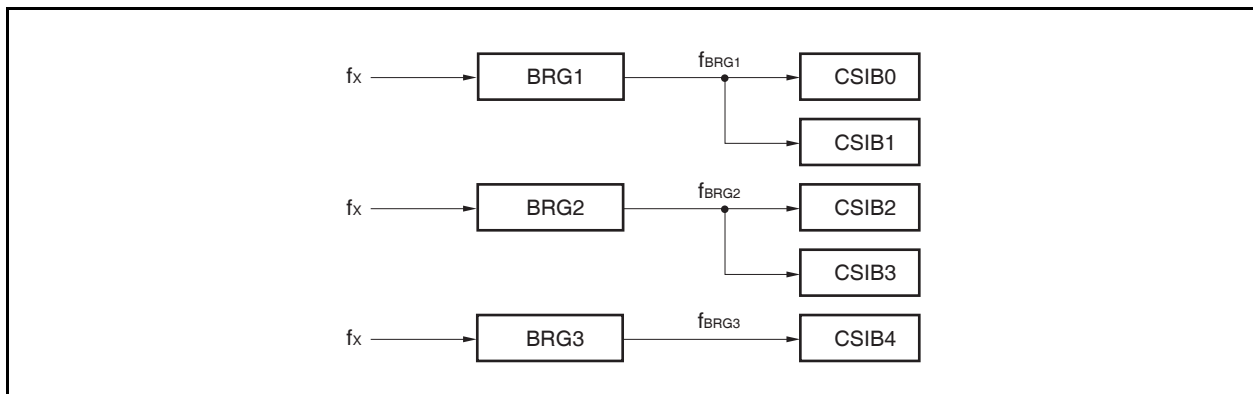
当 CSIBn 的操作非使能($\text{CBnCTL0.CBnPWR} = 0$)时, 引脚 SOBn 输出状态如下。

CBnTXE	CBnDAP	CBnDIR	SOBn 引脚输出
0	\times	\times	固定于低电平
1	0	\times	SOBn 锁存值 (低电平)
	1	0	CBnTX 寄存器值 (MSB)
		1	CBnTX 寄存器值 (LSB)

- 备注**
1. 如果 CBnCTL0.CBnTXE , CBnCTL0.CBnDIR 和 CBnCTL1.CBnDAP 被复写, 引脚 SOBn 的输出电平将改变。
 2. $n = 0$ 至 4
 3. \times : 不用考虑

16.8 波特率发生器

BRG1 ~ BRG3 和 CSIB0 ~ CSIB4 波特率发生器的连接如下面框图所示。



(1) 预分频模式寄存器 1 ~ 3 (PRSM1 ~ PRSM3)

寄存器 PRSM1 ~ PRSM3 控制产生 CSIB 的波特率信号。

可由 1 位或 8 位存储器操作指令读/写该寄存器。

复位信号产生将该寄存器设置为 00H。

复位后: 00H R/W 地址: PRSM1 FFFFF320H, PRSM2 FFFFF324H,
PRSM3 FFFFF328H

PRSMm (m = 1 ~ 3)	7	6	5	<4>	3	2	1	0
	0	0	0	BGCEm	0	0	BGCSm1	BGCSm0

BGCEm	波特率输出
0	禁止
1	使能

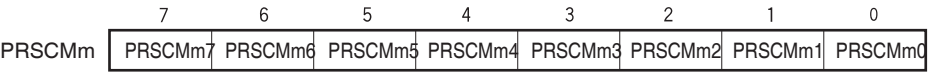
BGCSm1	BGCSm0	输入时钟选择 (f _{BGCSn})	设置值 (k)
0	0	f _{xx}	0
0	1	f _{xx} /2	1
1	0	f _{xx} /4	2
1	1	f _{xx} /8	3

- 注意事项**
- 操作过程中，不要复写寄存器 PRSMm。
 - 在设置 BGCEm = 1 前，设置寄存器 PRSMm。

(2) 预分频比较寄存器 1 ~ 3 (PRSCM1 ~ PRSCM3)

寄存器 PRSCM1 ~ PRSCM3 是 8 位比较寄存器。
可由 1 位或 8 位存储器操作指令读/写该寄存器。
复位信号产生将该寄存器设置为 00H。

复位后: 00H R/W 地址: PRSCM1 FFFF321H, PRSCM2 FFFF325H,
PRSCM3 FFFF329H



- 注意事项
- 1. 操作过程中，不要复写寄存器 PRSCMm。
 - 2. 在设置 BGCEm = 1 前，设置寄存器 PRCMm。

16.8.1 波特率

对主时钟分频产生发送/接收时钟。由主时钟产生波特率通过下列公司计算。

$$f_{BRGm} = \frac{f_{xx}}{2^{k+1} \times N}$$

- 备注
- f_{BRGm}: BRGm 计数时钟
 - f_{xx}: 主时钟频率
 - k: 寄存器 PRSMm, 其值 = 0 至 3
 - N: 寄存器 PRSCMm, 其值= 1 至 256
然而，只有当寄存器 PRSCMm 为 00H 时，N = 256。
 - m = 1 至 3

16.9 注意事项

- (1) 当使用 DMA 传送接收和发送的数据，即使串行传送期间出现溢出错误，也不进行处理。可以在 DMA 传送后读取寄存器 CBnSTR.CBnOVE，查看是否出现溢出错误。
- (2) 考虑到寄存器禁止在操作期间(CBnCTL0.CBnPWR 为 1)复写，如果操作期间由于错误操作而复写寄存器，设置 CBnCTL0.CBnPWR 为 0，然后初始化 CSIBn。

禁止在操作期间复写的寄存器如下。

- 寄存器 CBnCTL0: CBnTXE, CBnRXE, CBnDIR, CBnTMS 位
- 寄存器 CBnCTL1: CBnCKP, CBnDAP, CBnCKS2 至 CBnCKS0 位
- 寄存器 CBnCTL2: CBnCL3 至 CBnCL0 位

- (3) 通信类型 2 或 4 (CBnCTL1.CBnDAP = 1)下，产生接收完成中断(INTCBnR)后， \overline{SCKBn} 时钟执行到一半时 CBnSTR.CBnTSF 被清零。

单独发送模式下，通信过程中(CBnTSF = 1)忽略写入的下一个发送数据，下一次通信不开始。同样，在只能接收的通信中(CBnCTL0.CBnTXE = 0, CBnCTL0.CBnRXE = 1)，如果接收到的数据在通信过程中(CBnTSF = 1)被读取，则下一次通信不开始。

因此，在通信模式 2 或 4(CBnDAP = 1)下的单独发送模式，特别要注意下述问题。

- 要开始下一次发送，确保 CBnTSF = 0，然后向寄存器 CBnTX 写入要发送的数据。
- 只能接收通信模式下(CBnTXE = 0, CBnRXE = 1)，要连续执行下一次接收，确保 CBnTSF = 0，然后读取寄存器 CBnTX。

或者，使用连续发送模式而不用单独发送模式。推荐在使用 DMA 时使用连续发送模式。

备注 n = 0 至 4

第十七章 I²C 总线

为了使用 I²C 总线功能，分别设置 P38/SDA00，P39/SCL00，P40/SDA01，P41/SCL01，P90/SDA02 和 P91/SCL02 引脚为 N-ch 开漏输出，它们分别作为串行收发数据 IO 引脚（SDA00 to SDA02）和串行时钟 IO 引脚（SCL00 to SCL02）。

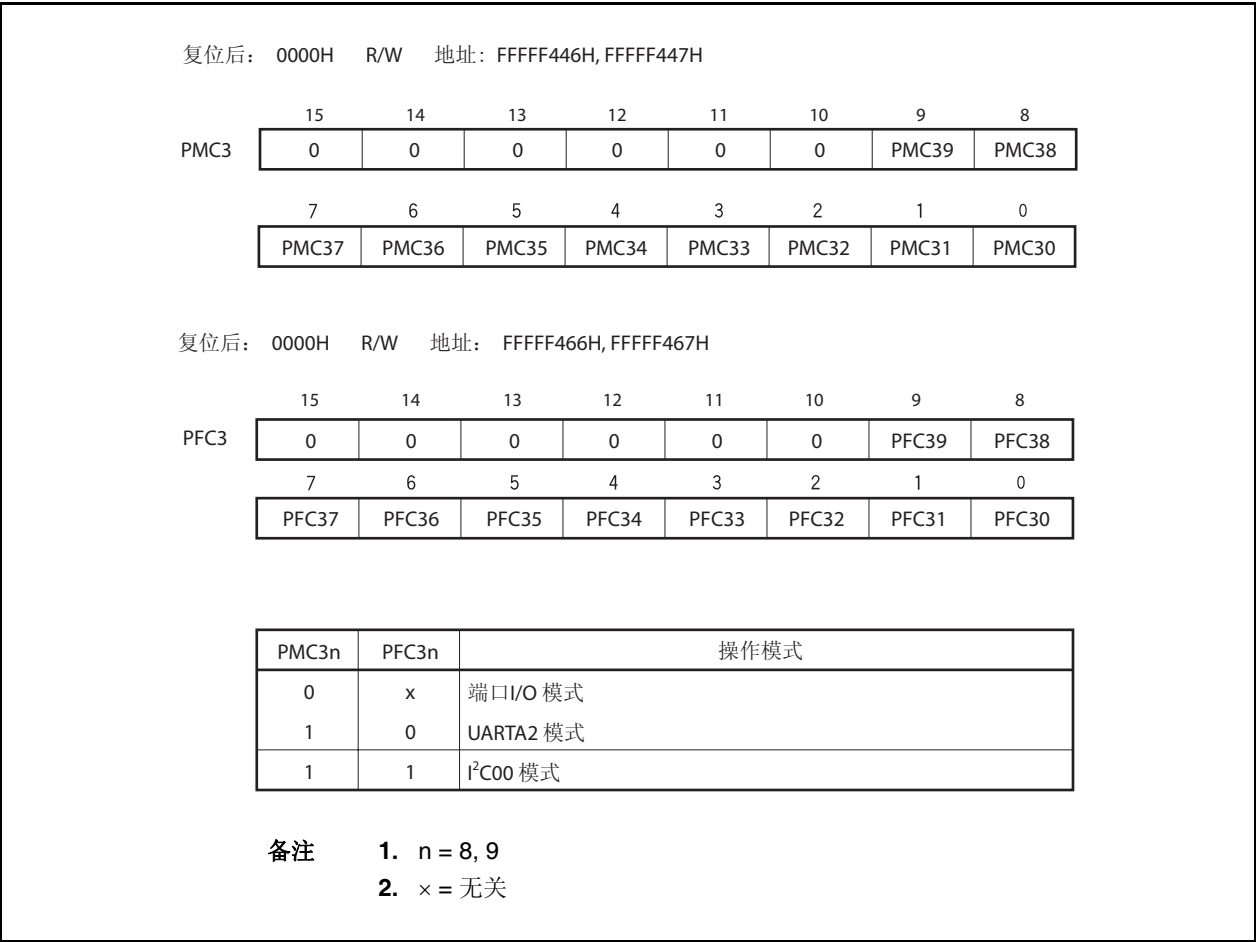
17.1 I²C 总线和其它串行总线的模式转换

17.1.1 UARTA2 和 I²C00 模式的模式转换

V850ES/JG2 中，UARTA2 和 I²C00 是同一个引脚的复用功能，因此不能同时使用。使用之前首先用 PMC3 和 PFC3 寄存器设置 I²C00。

注意事项 在传送或接收过程中，如果功能发生了转换，那么 UARTA2 和 I²C00 的传送/接收操作并不能保证。请务必禁止未使用的那个功能。

图 17-1. UARTA2 和 I²C00 的模式转换设置



17.1.2 CSIB0 和 I²C01 的模式转换

V850ES/JG2 中，CSIB0 和 I²C01 是同一个引脚的复用功能，因此不能同时使用。使用之前首先用 PMC4 和 PFC4 寄存器设置 I²C01。

注意事项 在传送或接收过程中，如果功能发生了转换，那么 CSIB0 和 I²C01 的传送/接收操作并不能保证。请务必禁止未使用的那个功能。

图 17-2. CSIB0 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

PMC4

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

PFC4

0	0	0	0	0	0	PFC41	PFC40
---	---	---	---	---	---	-------	-------

PMC4n	PFC4n	操作模式
0		端口 I/O 模式
1	0	CSIB0 模式
1	1	I ² C01 模式

备注

1. n = 0, 1

2. × = 无关

556

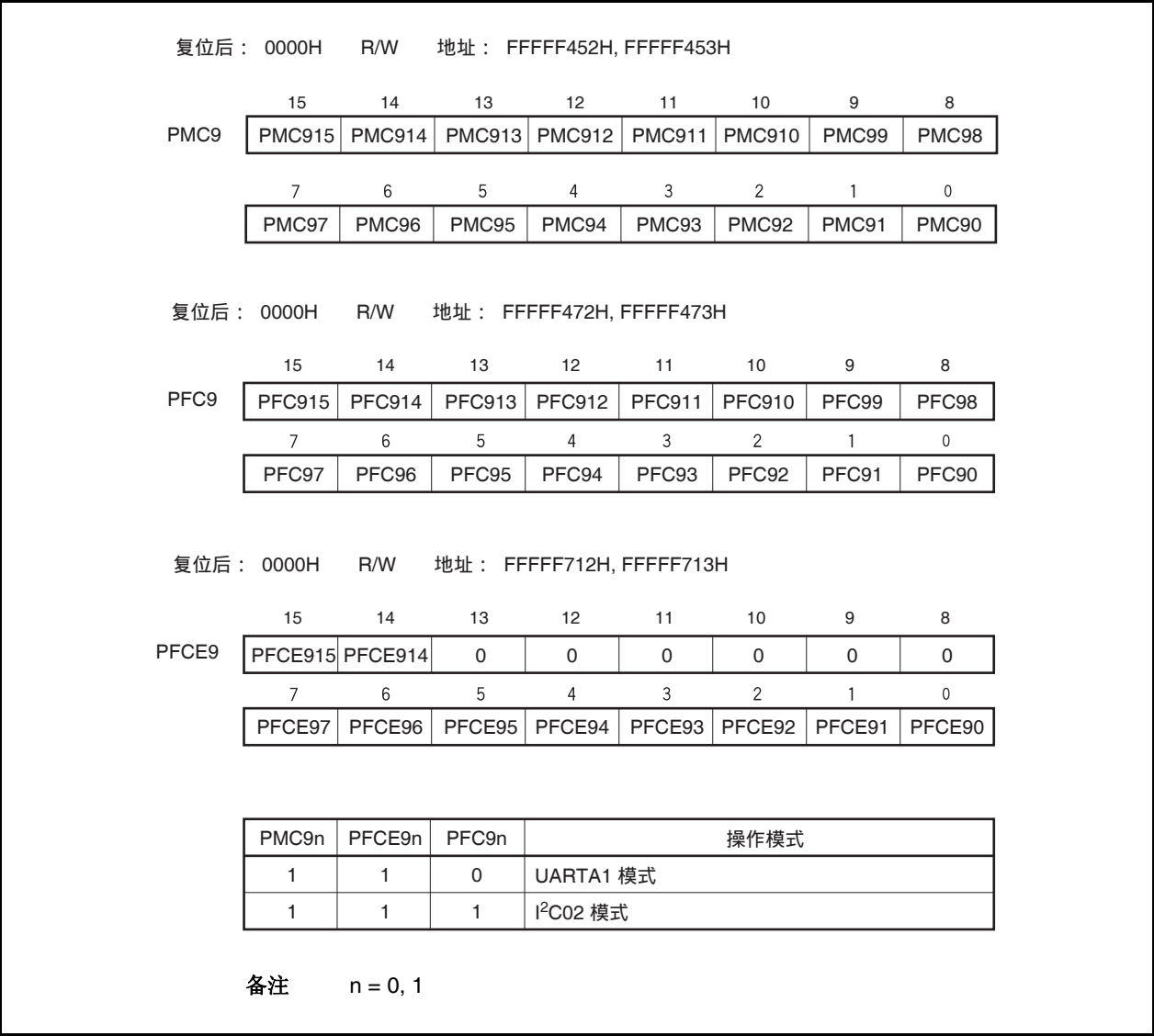
用户手册 U17715CA2V0UD

17.1.3 UARTA1 和 I²C02 的模式转换

V850ES/JG2 中，UARTA1 和 I²C02 是同一个引脚的复用功能，因此不能同时使用。使用之前首先用 PMC9 和 PFC9 以及 PFCE9 寄存器设置 I²C02。

注意事项 在传送或接收过程中，如果功能发生了转换，那么 UARTA1 和 I²C02 的传送/接收操作并不能保证。请务必禁止未使用的那个功能。

图 17-3. UARTA1 和 I²C02 的模式转换设置



17.2 特征

I²C00 到 I²C02 有下列两种模式。

- 操作停止模式
- I²C (Inter IC) 总线模式 (支持多主模式)

(1) 操作停止模式

这种模式下，不进行串行传输，因此能够降低了能耗。

(2) I²C 总线模式 (支持多主模式)

这种模式用于在几个设备之间进行 8 位数据传输，使用两条线：串行时钟引脚(SCL0n)和串行数据总线引脚(SDA0n)

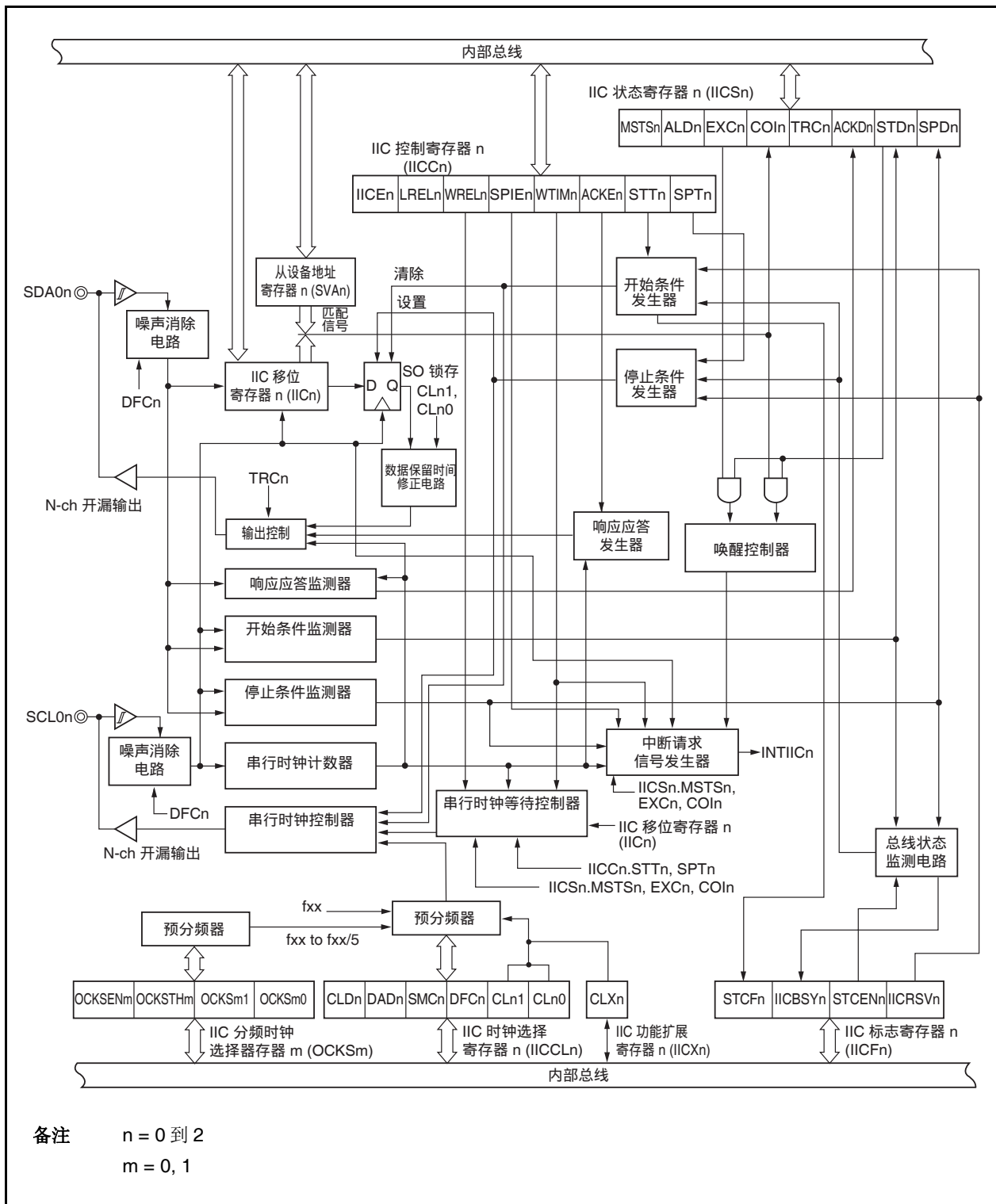
这种模式遵循 I²C 总线格式，并且主设备能产生“开始条件”，“地址”，“传送方向”，“数据”和“停止条件”数据，通过串行数据总线传给从设备。从设备的监测接收状态和数据由硬件自动完成。这个功能可以简化应用程序中 I²C 总线的控制部分。

既然 SCL0n 和 SDA0n 引脚用作 N-ch 开漏输出，那么串行时钟线和串行数据总线需要一个上拉电阻。

备注 **n = 0 到 2**

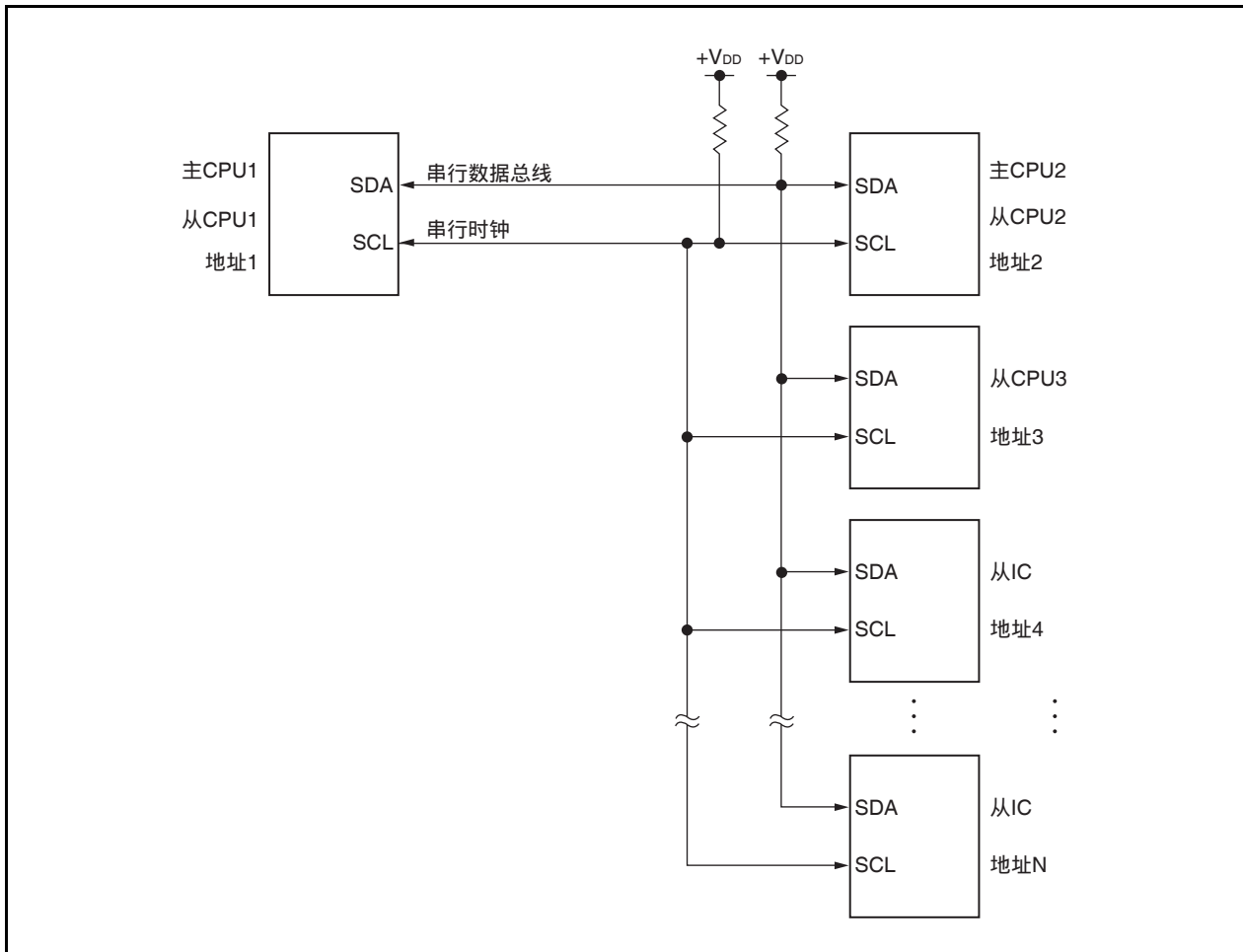
17.3 配置

I²C0n 方框图如下所示。

图 17-4. I²C0n 方框图

一个串行总线配置实例如下所示。

图 17-5. 使用 I²C Bus 的串行总线配置实例



I²C0n 包括下列硬件 (n = 0 到 2)。

表 17-1. I²C0n 的配置

名称	配置
寄存器	IIC 移位寄存器 n (IICn) 从地址寄存器 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 寄存器把 8 位串行数据转换成 8 位并行数据，反之亦然，并且传送和接收中都能使用(n = 0 to 2)。
对 IICn 寄存器的读和写操作用来控制实际的传送和接收操作。
这个寄存器能进行 8 位读或写操作。
复位会将这个寄存器清 00H。

(2) 从地址寄存器 n (SVAn)

SVAn 寄存器从模式时设置本地地址 (n = 0 到 2)。
这个寄存器能进行 8 位读或写操作。
复位会将这个寄存器清 00H。

(3) SO 锁存

SO 锁存用来保存 SDA0n 引脚 的输出电平 (n = 0 到 2)。

(4) 唤醒控制器

当本寄存器接收到的地址和 SVAn 寄存器中的地址值相等时，或者接收到扩展码时，这个电路产生一个中断请求 (INTIICn) (n = 0 to 2)。

(5) 预分频器

选择使用的采样时钟。

(6) 串行时钟计数器

这个计数器在传送和接收操作期间对输出和输入的串行时钟进行计数，并用于对传送出和接收到的 8 位数据进行校验。

(7) 中断请求信号发生器

这个电路控制中断请求信号 (INTIICn) 产生。

当满足下列任意一个触发条件时，有 I²C 中断产生。

- 串行时钟的第 8 个或第 9 个时钟的下降沿(由 IICn.WTIMn 位设置)
- 监测到停止条件时 (由 IICn.SPIEn 位设置)

备注 n = 0 到 2

(8) 串行时钟控制器

主模式下，这个电路从采样时钟产生时钟并通过 SCL0n 引脚输出(n = 0 到 2)。

(9) 串行时钟等待寄存器

这个电路控制等待时间。

(10) \overline{ACK} 发生器， 停止条件监测器， 开始条件监测器， 和 \overline{ACK} 监测器。

这个电路用来产生和监测各种状态。

(11) 数据保持时间修正电路

这个电路根据 SCL0n 引脚的下降沿来确定数据的保持时间。

(12) 开始条件发生器

当 IICn.STTn 置位时产生开始条件。

然而，在通信保留禁止状态 (IICFn.IICRSVn 位 = 1)，这个请求被忽略，并且如果总线不释放 (IICFn.IICBSYn 位 = 1)，那么 IICFn.STCFn 位被置 1。

(13) 停止条件发生器

当 IICn.SPTn 置位时产生停止条件。

(14) 总线状态监测器

通过监测开始条件和停止条件，确定总线是否被释放。

然而，总线状态无法在操作后立即监测，因此使用 IICFn.STCENn 位来将总线状态监测器设置为初始状态。

17.4 寄存器

I²C00 到 I²C02 由下列寄存器控制。

- IIC 控制寄存器 0 到 2 (IICC0 到 IICC2)
- IIC 状态寄存器 0 到 2 (IICS0 到 IICS2)
- IIC 标志寄存器 0 到 2 (IICF0 到 IICF2)
- IIC 时钟选择寄存器 0 到 2 (IICCL0 到 IICCL2)
- IIC 功能扩展寄存器 0 到 2 (IICX0 到 IICX2)
- IIC 分频时钟选择寄存器 0, 1 (OCKS0, OCKS1)

下列寄存器也同样要使用。

- IIC 移位寄存器 0 到 2 (IIC0 到 IIC2)
- 从机地址寄存器 0 到 2 (SVA0 到 SVA2)

备注 复用引脚的设置，请参阅表 4-15 端口引脚用作复用功能。

(1) IIC 控制寄存器 0 到 2 (IICC0 到 IICC2)

IICC0 到 IICC2 寄存器可以允许/停止 I²C0n 操作，设置等待时间或进行其它 I²C 操作 (n = 0 到 2)。

这些寄存器能进行 8 位或 1 位读写操作。然而，当 IICE0 位为 0 或者在等待过程中，需要对 SPIEn、

WTIMn 和 ACKEn 置位。当 IICEn 位从“0”到“1”时，这些位也能同时设置。

复位输入将这些寄存器清 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 到 2)

IICEn	I ² Cn 允许/禁止操作的详细规定
0	操作停止。 IICSn 寄存器复位 ¹ 。 内部操作停止。
1	操作允许。
当 SCL0n 和 SDA0n 线为高电平时, 请确保将此位置 1。	
清零条件 (IICEn 位 = 0)	
置位条件 (IICEn 位 = 1)	
<ul style="list-style-type: none"> 指令清零 复位后 	

LRELn ²	退出通信
0	正常操作
1	退出当前通信状态并且设置为待机模式。这个设置在执行后自动清除。本操作同样适用于接收到本地不相关扩展码的情况。 SCL0n 和 SDA0n 线 设置为高阻态。 IICSn 寄存器的 STTn 和 SPTn 位和 MSTSn, EXCn, COIn, TRCn, ACKDn 和 STDn 位清零。
从通信退出后的待机模式实际上会持续保持, 直到满足下一个通信进入条件为止:	
<ul style="list-style-type: none"> 监测到停止条件时, 重启后进入主模式。 在开始条件之后, 产生地址匹配或者接收到扩展码。 	
清零条件 (LRELn 位 = 0)	
设置的条件 (LRELn 位 = 1)	
<ul style="list-style-type: none"> 执行后自动清除 复位后 	

WRELn ²	等待状态取消控制位
0	不取消等待状态
1	取消等待状态。在等待状态被取消后, 这个设置会自动清除。
清除条件(WRELn 位 = 0)	
设置条件(WRELn 位 = 1)	
<ul style="list-style-type: none"> 执行后自动清除 复位后 	

- 注 1. IICSn 寄存器, IICFn.STCFn 和 IICFn.IICBSYn 位, 和 IICCLn.CLDn 和 IICCLn.DADn 位复位。
2. 当 IICEn 位 = 0 时, 标志信号无效。

注意 当 SCL0n 线是高电平, SDA0n 线是低电平时, 如果允许 I²Cn 操作 (IICEn 位 = 1), 那么会立即监测到开始条件。为了避免这种情况, 在 I²Cn 操作使能之后, 立即用位操作指令将 LRELn 置 1。

备注 数据设置后进行读取时, LRELn 和 WRELn 位清零。

SPIEn ^注	监测到停止条件时使能/禁止中断请求产生	
0	禁止	
1	允许	
清零条件 (SPIEn 位 = 0)		设置条件 (SPIEn 位 = 1)
<ul style="list-style-type: none">• 由指令清除• 复位清除		<ul style="list-style-type: none">• 由指令设置

WTIMn ^注	等待状态和中断请求产生的控制位		
0	第 8 个时钟的下降沿产生中断请求 主模式： 输出 8 个时钟后，时钟输出被置为低电平，并设置等待状态。 从模式： 8 个时钟输入后，时钟设置为低电平，并为主设备设置等待状态。		
1	第 9 个时钟下降沿产生中断请求。 主模式： 9 个时钟输出后，时钟输出被置为低电平，并设置等待状态。 从模式： 9 个时钟输入后，时钟设置为低电平，并为主设备设置等待状态。		
在地址传送过程中，不管此位如何设置，在第 9 个时钟下降沿都会产生中断。地址传送完毕后，对此位的设置才有效。主模式下，地址传送过程中，在第 9 个时钟下降沿插入等待状态。对于接收本地地址的从设备来说， \overline{ACK} 产生后的第 9 个时钟下降沿插入等待状态。然而，当从设备接收到扩展码时， 等待状态插在第 8 个时钟下降沿处。			
清零条件 (WTIMn 位 = 0)		设置条件 (WTIMn 位 = 1)	
● 指令清除 ● 复位清除		● 指令设置	

ACKEn ^注	响应控制
0	禁止响应
1	允许响应。在第 9 个时钟期间，SDA0n 线为低电平。
对于从设备的地址接收来说，ACKEn 位设置无效。这种情况下，当地址匹配时产生 <u>ACK</u> 信号。然而，对扩展码的接收来说，此位设置有效。在接收扩展码的系统中，ACKEn 置位。	
清零条件(ACKEn 位= 0)	设置条件 (ACKEn 位= 1)
<ul style="list-style-type: none">• 由指令清除• 复位清除	<ul style="list-style-type: none">• 由指令设置

注 当 IICEn 位 = 0 时，标志信号无效。

备注 n = 0 到 2

STTn	开始条件触发
0	不产生开始条件。
1	<p>总线释放时 (STOP 模式下):</p> <p>(作为主设备开始), 当 SCLn 线为高电平, SDA0n 线从高电平变为低电平时产生开始条件。接着在一段时间间隔后, SCL0 线变为低电平。</p> <p>和第三方通信期间:</p> <p>假如通信保留功能使能(IICFn.IICRSVn 位 = 0)</p> <ul style="list-style-type: none">• 此触发器作为开始条件保留标志。当设置为 1 时, 释放总线, 然后自动产生开始条件。 <p>假如通信保留功能禁止 (IICRSVn = 1)</p> <ul style="list-style-type: none">• IICFn.STCFn 为设置为 1, 并且 STTn 位的信息设置 (1) 清除。 触发器不产生开始条件。 <p>等待状态 (当主设备时):</p> <p>等待状态释放后, 重启条件产生。</p>
<p>有关设置时序的注意事项</p> <p>对于主接收操作: 传送期间不能设置为 1。只有当 ACKEn 位设置为 0, 并且告知从设备接收终止时, 设置为 1。</p> <p>对于主传送操作: 在 ACK 期间开始条件不能正常产生。第 9 个时钟输出后的等待过程中设置为 1。</p> <p>对于从设备: 即使当通信保留功能禁止(IICRSVn 位 = 1)时, 也进入通信保留状态。</p> <ul style="list-style-type: none">• 当 SPTn 位禁止的同时, 设置为 1。• 当 STTn 位设置为 1 时, 禁止再次设置 STTn 位为 1, 直到设置清除为止。	
清零条件 (STTn 位 = 0)	设置条件 (STTn 位 = 1)
<ul style="list-style-type: none">• 在通信保留禁止状态下, 当 STTn 位设置为 1 时• 仲裁失败则被清除• 开始条件产生后, 由主设备清除• LRELn 位 = 1 (通信除外)• IICEEn 位 = 0 (操作停止)• 复位后	<ul style="list-style-type: none">• 由指令设置

备注 1. 数据设置后, 如果立即读数据, STTn 位为 0。
 2. n = 0 到 2

★

SPTn	停止条件触发
0	不产生停止条件
1	产生停止条件 (主设备传送终止)。 当 SDA0n 线进入低电平, 将 SCL0n 设置为高电平, 或者插入等待状态直到 SCLn0 引脚变为高电平。 在固定的时间间隔后, SDAAn0 线从低电平变为高电平, 停止条件产生。
有关设置时序的注意事项	
主接收: 传送时不能设置为 1。 只有当 ACKEn 位设置为 0, 并且在从设备被告知接收终止后的等待过程中, 才可以设置为 1。	
主传送: 在 <u>ACK</u> 期间开始条件不能正常产生。第 9 个时钟输出后的等待过程中, 设置为 1。	
<ul style="list-style-type: none">不能和 STTn 位同时设置为 1。只有处于主模式[✱]时, SPTn 位才可以设置为 1。当 WTIMn 位设置为 0 时, 假如输出 8 个时钟后的等待过程中, SPTn 位设置为 1, 注意在第 9 个时钟的高电平期间将产生停止条件。 WTIMn 位在输出 8 个时钟后的等待期间应当从 0 变成 1, SPTn 位在第 9 个时钟输出后的等待期间应当置 1。当 SPTn 位设置为 1 时, 禁止将 SPTn 位再次设置为 1, 直到设置清零为止。	
清除条件(SPTn 位 = 0)	设置条件 (SPTn 位 = 1)
<ul style="list-style-type: none">由仲裁失败清除监测到停止条件后自动清除当 LRELn 位 = 1 (通信除外)当 IICEn 位 = 0 (停止操作)复位后	<ul style="list-style-type: none">由指令设置

注 只有在主模式下, SPTn 位才可以置为 1。但是当 IICRSVn 位为 0 时, SPTn 位必须设置为 1, 并且切换到操作允许状态后, 监测到第一个停止条件之前产生停止条件。具体细节敬请查阅 17.15 注意事项。

注意事项 当 TRCn 位 = 1 时, 在第 9 个时钟和等待状态取消期间, WRELn 位设置为 1, 之后 TRCn 位清零, 并且 SDA0n 线置为高阻状态。

备注 1. 数据设置后, 假如立即进行读操作, SPTn 位为 0。
 2. n = 0 到 2

(2) IIC 状态寄存器 0 到 2 (IICS0 到 IICS2)

IICS0 到 IICS2 指示 I²C0n 总线的状态 (n = 0 到 2)。

这些寄存器为只读寄存器，支持 8 位字节操作和位操作。但只有 IICn.STTn 位为 1 或者在等待期间才能读取。

复位输入将这些寄存器清零 00H。

注意事项 下列几种状态禁止访问 IICSn 寄存器。详细情况，参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。

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

(1/3)

复位后: 00H R 地址: IICS0 FFFFFD86H, IICS1 FFFFFD96H, IICS2 FFFFFDA6H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTS _n	ALD _n	EXC _n	COL _n	TRC _n	ACKD _n	STD _n	SPD _n

(n = 0 到 2)

MSTS _n	主设备状态	
0	从设备状态或者通信待机状态	
1	主设备通信状态	
清除条件(MSTS _n 位 = 0)		设置条件 (MSTS _n 位 = 1)
<ul style="list-style-type: none">• 当监测到停止条件时• 当 ALD_n 位 = 1 (仲裁失败)• 由 LREL_n 位 = 1 清除(通信除外)• 当 IICE_n 位从 1 变为 0 时 (停止操作)• 复位后		<ul style="list-style-type: none">• 开始条件产生时

ALD _n	监测仲裁失利	
0	此状态表明没有仲裁或者仲裁结果是“成功”。	
1	此状态表明仲裁结果是“失败”。MSTS _n 位清零。	
清零条件 (ALD _n 位 = 0)		设置条件 (ALD _n 位 = 1)
<ul style="list-style-type: none">• 在 IICS_n 寄存器被读取^注后自动清除• 当 IICE_n 位从 1 变为 0 (停止操作)• 复位后		<ul style="list-style-type: none">• 当仲裁结果是“失败”。

EXC _n	扩展码的监测	
0	没有接收到扩展码。	
1	接收到扩展码。	
清除条件 (EXC _n 位 = 0)		清除条件 (EXC _n 位 = 1)
<ul style="list-style-type: none">• 监测到开始条件• 监测到停止条件• LREL_n 位 = 1 则清除 (通信除外)• 当 IICE_n 位从 1 变为 0 (操作停止)• 复位后		<ul style="list-style-type: none">• 当接收到的地址数据的高四位是“0000”或者“1111”（在第 8 个时钟的上升沿设置）。

注 当对 IICS_n 寄存器之中的其他位进行位操作时，本位也清零。

★

COIn	监测匹配地址
0	地址不匹配。
1	地址匹配。
清零条件 (COIn 位 = 0)	
<ul style="list-style-type: none"> 监测到开始条件 监测到停止条件 由 LRELn 位 = 1 清除 (通信除外) 当 IICEn 位从 1 变成 0 (停止操作) 复位后 	
设置条件 (COIn 位 = 1)	
<ul style="list-style-type: none"> 当接收到的地址与本地地址匹配时 (SVAn 寄存器) (在第 8 个时钟的上升沿设置)。 	

TRCn	监测传送/接收状态
0	接收状态 (非传送状态)。SDA0n 线设置为高阻状态。
1	传送状态。SO 锁存器中的数值允许输出到 SDA0n 线 (第 1 个字节的第 9 个时钟下降沿处开始有效)。
清零状态 (TRCn 位 = 0)	
<ul style="list-style-type: none"> 监测到停止状态 由 LRELn 位 = 1 清除 (通信除外) 当 IICEn 位从 1 变为 0 时 (停止操作) 由 IICn.WRELn 位 = 1 清除 当 ALDn 位从 0 变为 1 (仲裁失败) 复位 	
设置条件 (TRCn 位 = 1)	
主模式 <ul style="list-style-type: none"> 产生开始条件时 当“0”输出给第一个字节的 LSB (传送方向规定位) 从模式 <ul style="list-style-type: none"> 当第一个字节的 LSB (传送方向规定位) 输入“1” 	
主模式 <ul style="list-style-type: none"> 当“1”输出给第一个字节的 LSB (传送方向规定位) 从模式 <ul style="list-style-type: none"> 监测到开始条件 不用于通信时	

ACKDn	ACK 监测
0	不监测 $\overline{\text{ACK}}$ 。
1	监测 $\overline{\text{ACK}}$ 。
清零条件 (ACKDn 位 = 0)	
<ul style="list-style-type: none"> 监测到停止条件时 在下一个字节的第一个时钟的上升沿 由 LRELn 位 = 1 清除 (通信除外) 当 IICEn 位从 1 变为 0 (停止操作) 复位后 	
设置条件 (ACKD 位 = 1)	
<ul style="list-style-type: none"> SCL0n 引脚的第九个时钟的上升沿时, SDA0n 位设置为低电平之后 	

注 当 WRELn 位设置为 1, 并且在第 9 个时钟通过 TRCn = 1 将等待状态取消为 0 时, TRCn 位清零, SDA0n 线为高阻态。

备注 n = 0 到 2

STDn	监测开始条件
0	不监测开始条件。
1	监测开始条件。用来标明地址传送正在进行。
清零状态 (STDn 位 = 0)	
设置状态 (STDn 位 = 1)	
<ul style="list-style-type: none">• 监测到停止条件• 地址传送后下一个字节的第一个时钟的上升沿时• 由 LRELn 位 = 1 (通信除外)• 当 IICEn 位从 1 变为 0 (操作停止)• 复位后	<ul style="list-style-type: none">• 监测到开始条件。

SPDn	停止条件监测
0	不监测停止条件。
1	监测停止状态。主设备通信终止，总线释放。
清零条件 (SPDn 位 = 0)	
设置条件 (SPDn 位 = 1)	
<ul style="list-style-type: none">• 此位设置之后的地址传送字节的第一个时钟上升沿，监测到开始条件• 当 IICEn 位从 1 变为 0 (操作停止)• 复位后	<ul style="list-style-type: none">• 监测到停止条件

备注 n = 0 到 2

(3) IIC 标志寄存器 0 到 2 (IICF0 到 IICF2)

IICF0 到 IICF2 寄存器设置 I²C0n 操作模式，并且指示 I²C 总线状态。

这些寄存器支持 8 位或 1 位读写操作。但是 STCFn 和 IICBSYn 位只读。

IICRSVn 允许/禁止通信保留功能 (请看 17.14 通信保留)。

IICBSYn 位的初始值用 STCENn 位设置 (请看 17.15 注意事项)。

只有当 I²C0n 操作禁止(IICCN.IICEn 位 = 0)时，IICRSVn 和 STCENn 位可写。操作使能后，IICFn 可读 (n = 0 到 2)。

复位输入将这些寄存器清零 00H。

复位后: 00H R/W^注 地址: IICF0 FFFFFD8AH, IICF1 FFFFFD9AH, IICF2 FFFFFDAAH

	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

(n = 0 到 2)

STCFn	STTn 位清除
0	发出开始条件
1	开始条件无法发出, STTn 位清除
清零条件 (STCFn 位 = 0)	
设置条件 (STCFn 位 = 1)	
<ul style="list-style-type: none"> • IICn.STTn 位 = 1 清除 • 当 IICn.IICEn 位 = 0 时 • 复位后 	<ul style="list-style-type: none"> • 在通信保留禁止 (IICRSVn 位 = 1) 期间, 开始条件不发出, 并且 STTn 标志清零。

IICBSYn	I ² Cn 总线状态
0	总线释放状态 (当 STCENn 位 = 1 时, 默认通信状态)
1	总线通信状态 (当 STCENn 位 = 0 时, 默认通信状态)
清零条件 (IICBSYn 位 = 0)	
设置条件 (IICBSYn 位 = 1)	
<ul style="list-style-type: none"> • 监测到停止条件 • 当 IICEn 位 = 0 时 • 复位后 	<ul style="list-style-type: none"> • 监测到开始条件 • 当 STCENn 位 = 0, 通过设置 IICEn 位

STCENn	初始开始使能触发
0	操作允许(IICEn 位 = 1)后, 监测到停止条件才产生开始条件, 。
1	操作允许(IICEn 位 = 1)后, 即使没有监测到停止条件, 也产生开始条件。
清零条件 (STCENn 位 = 0)	
设置条件 (STCENn 位 = 1)	
<ul style="list-style-type: none"> • 监测到开始条件 • 复位后 	<ul style="list-style-type: none"> • 由指令设置

IICRSVn	通信保留功能禁止位
0	通信保留允许
1	通信保留禁止
清零条件 (IICRSVn 位 = 0)	
设置条件 (IICRSVn 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 位 6 和位 7 只读。

- 注意事项**
1. 只有当操作停止(IICEn 位 = 0)时, 对 STCENn 位写入。
 2. 当 STCENn 位 = 1 时, I²Cn 总线操作允许后, 立即识别总线释放状态位(IICBSYn = 0), 不管实际总线状态如何。因此, 为了发出第一个开始条件(STTn = 1), 有必要确认总线释放, 以免干扰其它通信。
 3. 只有当操作停止(IICEn 位 = 0)时, 对 IICRSVn 位写入。

(4) IIC 时钟选择寄存器 0 到 2 (IICCL0 到 IICCL2)

IICCL0 到 IICCL2 寄存器设置 I²C0n 总线的传送时钟。

这些寄存器能进行 8 位或 1 位读写。然而，CLDn 和 DADn 位只读。

当 IICn.IICEn 位 = 0 时，设置 IICCLn 寄存器。

SMCn, CLn1 和 CLn0 位和 IICXn.CLXn 位以及 OCKSm 寄存器的 OCKSTHm, OCKSm1 和 OCKSm0 位的设置一起进行。(请看 17.4 (6) I²C0n 传送时钟设置方法) (n = 0 到 2, m = 0, 1)。

复位输入将这些寄存器清零 00H。

复位: 00H R/W[※] 地址: IICCL0 FFFFD84H, IICCL1 FFFFD94H, IICCL2 FFFFDA4H

	7	6	<5>	<4>	3	2	1	0
IICCLn	0	0	CLDn	DADn	SMCn	DFCn	CLn1	CLn0

(n = 0 到 2)

CLDn	SCL0n 引脚电平的监测 (只有当 IICn.IICEn 位 = 1 时有效)
0	SCL0n 引脚在低电平时监测。
1	SCL0n 引脚在高电平时监测。
清零条件 (CLDn 位 = 0)	
<ul style="list-style-type: none"> • 当 SCL0n 引脚为低电平时 • 当 IICEn 位 = 0 (操作停止) • 复位后 	
设置条件 (CLDn 位 = 1)	
<ul style="list-style-type: none"> • 当 SCL0n 引脚为高电平 	

DADn	SDA0n 引脚电平的监测 (只有当 IICEn 位 = 1 时有效)
0	SDA0n 引脚在低电平时监测
1	SDA0n 引脚在高电平时监测
清零条件 (DADn 位 = 0)	
<ul style="list-style-type: none"> • SDA0n 引脚为低电平时 • 当 IICEn 位 = 0 (停止操作) • 复位后 	
设置条件 (DAD0n 位 = 1)	
<ul style="list-style-type: none"> • 当 SDA0n 引脚为高电平 	

SMCn	操作模式切换
0	标准模式操作。
1	高速模式操作。

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。	
高速模式下，不管 DFCn 位设置为何 (开/关)，传送时钟并不变化。	
数字滤波器用来在高速模式下除掉噪声。	

注 IICCLn 的位 4 和位 5 为只读位。

注意事项 一定要将 IICCLn 的位 7 和位 6 清零。

备注 IICn.IICEn 位 = 0 时，当读 CLDn 和 DADn 位时，读取的数值为 0。

(5) IIC 功能扩展寄存器 0 到 2 (IICX0 到 IICX2)

IICX0 到 IICS2 寄存器设置 I²C0n 功能扩展 (只在高速模式下有效)。

这些寄存器可以 8 位或 1 位读写操作。

CLXn 位的设置和 IICCLn 寄存器的 SMCn, CLn1, CLn0 位以及 OCKSm 寄存器的 OCKSTHm, OCKSm1, OCKSm0 位一起进行 (请看 17.4 (6) I²C0n 传送时钟设置方法) (m = 0, 1)。

当 IICCN.IICE n 位 = 0 时设置 IICXn 寄存器。

复位输入将这些寄存器清零 00H。

After reset: 00H R/W Address: IICX0 FFFFD85H, IICX1 FFFFD95H, IICX2 FFFFDA5H

	7	6	5	4	3	2	1	<0>
IICXn	0	0	0	0	0	0	0	CLXn

(n = 0 to 2)

(6) I²C0n 传送时钟设置方法

I²C0n 传送时钟频率 (f_{SCL}) 用下列表达式计算 (n = 0 到 2)。

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

m = 12, 18, 24, 36, 44, 48, 54, 60, 66, 72, 86, 88, 96, 132, 172, 176, 198, 220, 258, 344 (参见 表 17-2 时钟设置)。

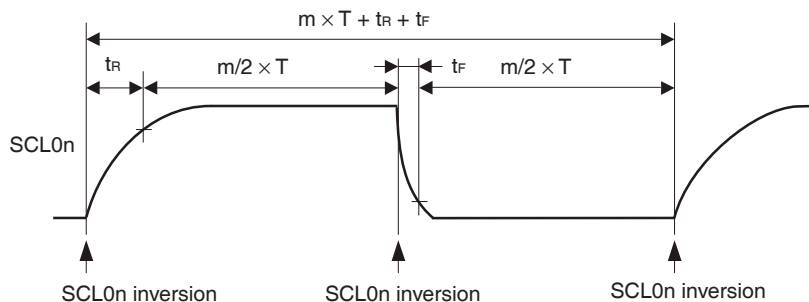
T: 1/f_{xx}

T_R: SCL0n 引脚上升时间

T_F: SCL0n 引脚下降时间

例如, the I²C0n transfer clock frequency (f_{SCL}) 当 f_{xx} = 19.2 MHz, m = 198, t_R = 200 ns, 和 t_F = 50 ns 时 I²C0n 传送时钟频率(f_{SCL})用下式计算。

$$f_{SCL} = 1/(198 \times 52 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 94.7 \text{ kHz}$$



选择的时钟可以由 IICCLn 寄存器的 SMCn, CLn1 和 CLn0 位, IICXn 寄存器的 CLXn 位以及 OCKSm 寄存器的 OCKSTHm, OCKSm1 和 OCKSm0 位共同设置 (n = 0 到 2, m = 0, 1)。

表 17-2. 时钟设置 (1/2)

IICX0	IICCL0			选择时钟	传送时钟	可选择的主时钟 频率 (fxx) 范围	操作模式
位 0	位 3	位 1	位 0				
CLX0	SMC0	CL01	CL00				
0	0	0	0	fxx (当 OCKS0 = 18H 时)	fxx/44	2.00 MHz ≤ fxx ≤ 4.19 MHz	标准模式 (SMC0 位 = 0)
				fxx/2 (当 OCKS0 = 10H 时)	fxx/88	4.00 MHz ≤ fxx ≤ 8.38 MHz	
				fxx/3 (当 OCKS0 = 11H 时)	fxx/132	6.00 MHz ≤ fxx ≤ 12.57 MHz	
				fxx/4 (当 OCKS0 = 12H 时)	fxx/176	8.00 MHz ≤ fxx ≤ 16.76 MHz	
				fxx/5 (当 OCKS0 = 13H 时)	fxx/220	10.00 MHz ≤ fxx ≤ 20.00 MHz	
0	0	0	1	fxx (当 OCKS0 = 18H 时)	fxx/86	4.19 MHz ≤ fxx ≤ 8.38 MHz	
				fxx/2 (当 OCKS0 = 10H 时)	fxx/172	8.38 MHz ≤ fxx ≤ 16.76 MHz	
				fxx/3 (当 OCKS0 = 11H 时)	fxx/258	12.57 MHz ≤ fxx ≤ 20.00 MHz	
				fxx/4 (当 OCKS0 = 12H 时)	fxx/344	16.76 MHz ≤ fxx ≤ 20.00 MHz	
0	0	1	0	Fxx [※]	fxx/86	4.19 MHz ≤ fxx ≤ 8.38 MHz	
0	0	1	1	fxx (当 OCKS0 = 18H 时)	fxx/66	6.40 MHz	
				fxx/2 (当 OCKS0 = 10H 时)	fxx/132	12.80 MHz	
				fxx/3 (当 OCKS0 = 11H 时)	fxx/198	19.20 MHz	
0	1	0	×	fxx (当 OCKS0 = 18H 时)	fxx/24	4.19 MHz ≤ fxx ≤ 8.38 MHz	高速模式 (SMC0 位 = 1)
				fxx/2 (当 OCKS0 = 10H 时)	fxx/48	8.00 MHz ≤ fxx ≤ 16.76 MHz	
				fxx/3 (当 OCKS0 = 11H 时)	fxx/72	12.00 MHz ≤ fxx ≤ 20.00 MHz	
				fxx/4 (当 OCKS0 = 12H 时)	fxx/96	16.00 MHz ≤ fxx ≤ 20.00 MHz	
0	1	1	0	Fxx [※]	fxx/24	4.00 MHz ≤ fxx ≤ 8.38 MHz	
0	1	1	1	fxx (当 OCKS0 = 18H 时)	fxx/18	6.40 MHz	
				fxx/2 (当 OCKS0 = 10H 时)	fxx/36	12.80 MHz	
				fxx/3 (当 OCKS0 = 11H 时)	fxx/54	19.20 MHz	
1	1	0	×	fxx (当 OCKS0 = 18H 时)	fxx/12	4.00 MHz ≤ fxx ≤ 4.19 MHz	
				fxx/2 (当 OCKS0 = 10H 时)	fxx/24	8.00 MHz ≤ fxx ≤ 8.38 MHz	
				fxx/3 (当 OCKS0 = 11H 时)	fxx/36	12.00 MHz ≤ fxx ≤ 12.57 MHz	
				fxx/4 (当 OCKS0 = 12H 时)	fxx/48	16.00 MHz ≤ fxx ≤ 16.67 MHz	
				fxx/5 (当 OCKS0 = 13H 时)	fxx/60	20.00 MHz	
1	1	1	0	Fxx [※]	fxx/12	4.00 MHz ≤ fxx ≤ 4.19 MHz	
其余				禁止设置	—	—	—

注 既然时钟选择为 f_{xx}，而忽略 OCKS0 寄存器中的数值，OCKS0 寄存器清 00H (I²C 分频时钟停止状态)。

备注 ×: 无关

表 17-2. 时钟设置 (2/2)

IICXm	IICCLm			选择时钟	传送时钟	可设置的主时钟 频率 (f _{xx}) 范围	操作模式
位 0	位 3	位 1	位 0				
CLXm	SMCm	CLm1	CLm0				
0	0	0	0	f _{xx} (当 OCKS1 = 18H 时)	f _{xx} /44	2.00 MHz ≤ f _{xx} ≤ 4.19 MHz	标准模式 (SMCm bit = 0)
				f _{xx} /2 (当 OCKS1 = 10H 时)	f _{xx} /88	4.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
				f _{xx} /3 (当 OCKS1 = 11H 时)	f _{xx} /132	6.00 MHz ≤ f _{xx} ≤ 12.57 MHz	
				f _{xx} /4 (当 OCKS1 = 12H 时)	f _{xx} /176	8.00 MHz ≤ f _{xx} ≤ 16.76 MHz	
				f _{xx} /5 (当 OCKS1 = 13H 时)	f _{xx} /220	10.00 MHz ≤ f _{xx} ≤ 20.00 MHz	
0	0	0	1	f _{xx} (当 OCKS1 = 18H 时)	f _{xx} /86	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	
				f _{xx} /2 (当 OCKS1 = 10H 时)	f _{xx} /172	8.38 MHz ≤ f _{xx} ≤ 16.76 MHz	
				f _{xx} /3 (当 OCKS1 = 11H 时)	f _{xx} /258	12.57 MHz ≤ f _{xx} ≤ 20.00 MHz	
				f _{xx} /4 (当 OCKS1 = 12H 时)	f _{xx} /344	16.76 MHz ≤ f _{xx} ≤ 20.00 MHz	
0	0	1	0	F _{xx} [※]	f _{xx} /86	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	
0	0	1	1	f _{xx} (当 OCKS1 = 18H 时)	f _{xx} /66	6.40 MHz	
				f _{xx} /2 (当 OCKS1 = 10H 时)	f _{xx} /132	12.80 MHz	
				f _{xx} /3 (当 OCKS1 = 11H 时)	f _{xx} /198	19.20 MHz	
0	1	0	×	f _{xx} (当 OCKS1 = 18H 时)	f _{xx} /24	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	高速模式 (SMCm 位 = 1)
				f _{xx} /2 (当 OCKS1 = 10H 时)	f _{xx} /48	8.00 MHz ≤ f _{xx} ≤ 16.76 MHz	
				f _{xx} /3 (当 OCKS1 = 11H 时)	f _{xx} /72	12.00 MHz ≤ f _{xx} ≤ 20.00 MHz	
				f _{xx} /4 (当 OCKS1 = 12H 时)	f _{xx} /96	16.00 MHz ≤ f _{xx} ≤ 20.00 MHz	
0	1	1	0	F _{xx} [※]	f _{xx} /24	4.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
0	1	1	1	f _{xx} (当 OCKS1 = 18H 时)	f _{xx} /18	6.40 MHz	
				f _{xx} /2 (当 OCKS1 = 10H 时)	f _{xx} /36	12.80 MHz	
				f _{xx} /3 (当 OCKS1 = 11H 时)	f _{xx} /54	19.20 MHz	
1	1	0	×	f _{xx} (当 OCKS1 = 18H 时)	f _{xx} /12	4.00 MHz ≤ f _{xx} ≤ 4.19 MHz	
				f _{xx} /2 (当 OCKS1 = 10H 时)	f _{xx} /24	8.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
				f _{xx} /3 (当 OCKS1 = 11H 时)	f _{xx} /36	12.00 MHz ≤ f _{xx} ≤ 12.57 MHz	
				f _{xx} /4 (当 OCKS1 = 12H 时)	f _{xx} /48	16.00 MHz ≤ f _{xx} ≤ 16.67 MHz	
				f _{xx} /5 (当 OCKS1 = 13H 时)	f _{xx} /60	20.00 MHz	
1	1	1	0	F _{xx} [※]	f _{xx} /12	4.00 MHz ≤ f _{xx} ≤ 4.19 MHz	
其余				禁止设置	—	—	—

注 既然时钟选择为 f_{xx}，而不管 OCKS1 寄存器的数值为何，OCKS1 寄存器清 00H (I²C 分频时钟停止状态)。

备注 1. m = 1, 2
2. ×: 无关

(7) IIC 分频时钟选择寄存器 0, 1 (OCS0, OCS1)

OCS0 和 OCS1 寄存器控制 I²C0n 分频时钟 (n = 0 到 2)。

这些寄存器通过 OCS0 寄存器控制 I²C00 分频时钟，通过 OCS1 寄存器控制 I²C01 和 I²C02 分频时钟。

这些寄存器能进行 8 位的读写操作。

复位输入将这些寄存器清 00H。

After reset: 00H R/W Address: 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	Operation setting of I ² C division clock						
0	Disable I ² C division clock operation						
1	Enable I ² C division clock operation						

OCKSTHm	OCKSm1	OCKSm0	Selection of I ² C division clock				
0	0	0	f _{xx} /2				
0	0	1	f _{xx} /3				
0	1	0	f _{xx} /4				
0	1	1	f _{xx} /5				
1	0	0	f _{xx}				
Other than above			Setting prohibited				

(8) IIC 移位寄存器 0 到 2 (IIC0 到 IIC2)

IIC0 到 IIC2 寄存器用于使串行传送/接收（移位操作）与串行时钟同步。这些寄存器能进行 8 位读写，但是数据传送期间不能写入 IICn 寄存器。

只在等待期间才可以访问（读/写）IICn 寄存器。除了等待期间之外，在通信状态时禁止访问。对于主设备来说，只有在传送触发位(IICn.STTn 位)置 1 之后，IICn 寄存器才能写入。

等待期间写入 IICn 寄存器会释放一个等待状态，数据传送开始(n = 0 到 2)。

复位输入将这些寄存器清 00H。

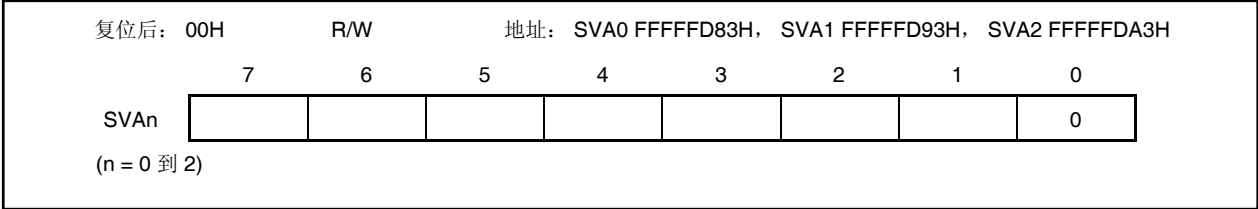
复位后: 00H R/W 地址: IIC0 FFFFFD80H, IIC1 FFFFFD90H, IIC2 FFFFFDA0H							
	7	6	5	4	3	2	1 0
IICn							
(n = 0 到 2)							

(9) 从地址寄存器 0 到 2 (SVA0 到 SVA2)

SVA_n 寄存器保存 I²C 总线的从地址。

这些寄存器能进行 8 位读写，但是最低位恒为 0。当 IICSn.STD_n 位 = 1（监测开始状态）时禁止重写寄存器。

复位输入将这些寄存器清 00H。



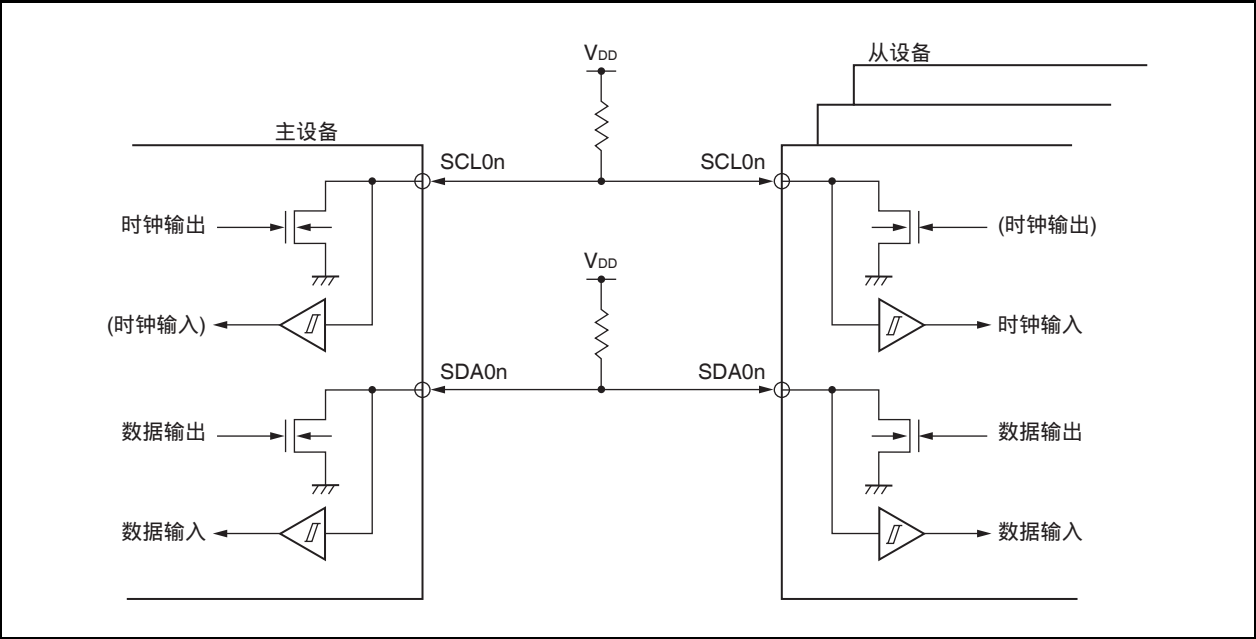
17.5 I²C 总线模式功能

17.5.1 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 到 2)。

- SCL0n 此引脚用于串行时钟输入和输出。
此引脚对主从设备均为 N-ch 开漏输出。输入为施密特输入。
 - SDA0n 此引脚用于串行数据输入和输出。
此引脚对主从设备均为 N-ch 开漏输出。输入为施密特输入。
- 既然串行时钟线和串行数据线的输出均为 N-ch 开漏输出，所以需要外部上拉电阻。

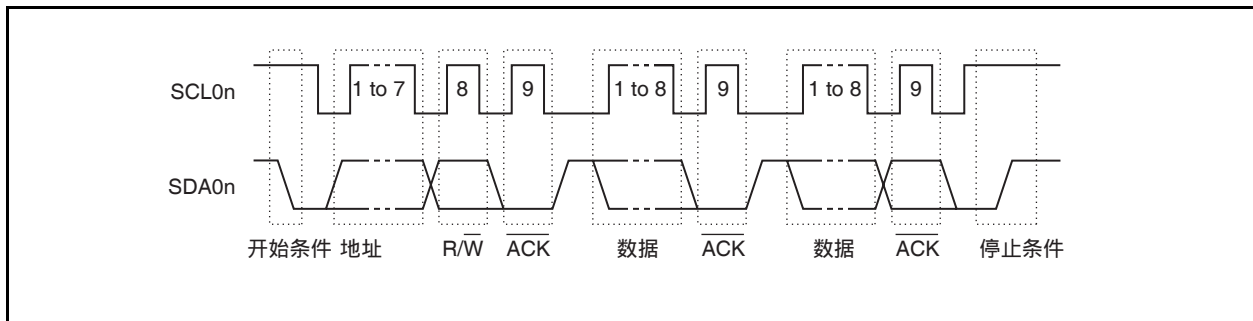
图 17-6. 引脚配置图



17.6 I²C 总线定义和控制方法

下面章节描述了 I²C 总线串行数据通信格式和 I²C 总线使用的信号。I²C 总线串行数据线上产生的“开始条件”，“地址”，“传送方向规定”，“数据”和“停止条件”分别如下所示。

图 17-7. I²C 串行数据传送时序



主设备产生开始条件，从地址和停止条件。

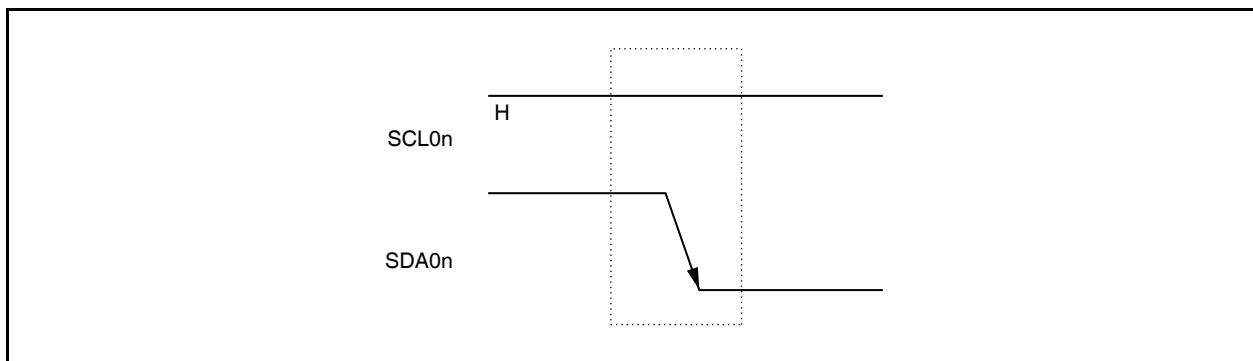
主设备和从设备都可以产生 ACK。(它通常由接收 8 位数据的设备产生)。

串行时钟 (SCL0n) 由主设备连续输出。然而，在从设备方面，SCL0n 引脚的低电平时间可延长，方便插入等待状态。(n = 0 到 2)。

17.6.1 开始条件

当 SCL0n 引脚为高电平，同时 SDA0n 引脚从高电平变成低电平时，开始条件满足。开始条件是当开始串行传送时，主设备由 SCL0n 和 SDA0n 引脚输出给从设备的信号。从设备能够检测到开始条件 (n = 0 到 2)。

图 17-8. 开始条件



监测到停止条件(IICSn.SPDn 位 = 1)后，当 IICn.STTn 位置 1 时，输出开始条件。监测到开始条件时，IICSn.STDn 位置 1 (n = 0 到 2)。

注意事项 同其它设备的通信进行过程中，当 V850ES/JG2 的 IICn.IICEn 位设置为 1 时，监测通信线的状态就可能找到开始条件。当 SCL0n 和 SDA0n 线为高电平时，一定要把 IICn.IICEn 位设置为 1。

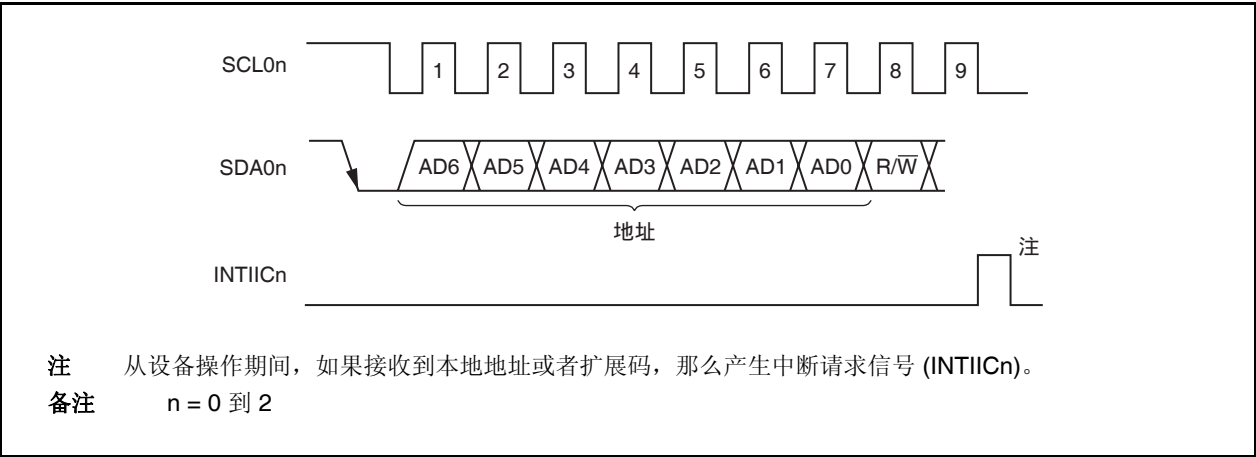
17.6.2 地址

开始条件之后紧跟的 7 位数据为地址。

地址通过总线选择连接到主设备的一个 7 位数据段，用来选择从设备。因此，每一个连接到总线上的从设备必须有唯一的地址。

从设备内的硬件可以监测开始条件，并检验接收到的 7 位地址数据是否与存储在 SVAn 寄存器里的地址数据匹配。如果地址数据与 SVAn 寄存器里的数值匹配，那么这个从设备被选中，与主设备的通信由主设备发来的开始条件或停止条件(n = 0 到 2)决定。

图 17-9. 地址



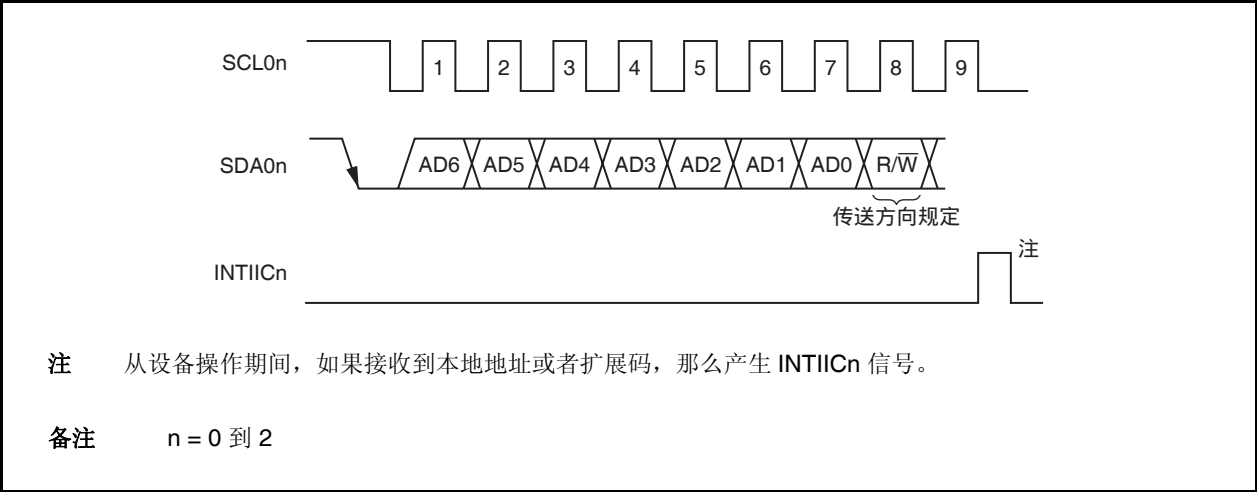
从地址和第 8 位，定义了传送方向，如下在 17.6.3 传送方向规定所描述的，一起写入 IIC 移位寄存器 n (IICn)，并且输出。接收到的地址写入 IICn 寄存器 (n = 0 到 2)。

从地址被指定给 IICn 寄存器的高 7 位。

17.6.3 传送方向规定

除了 7 位地址数据，主设备发送一个数据位规定传送方向。当此传送方向规定位为 0 时，表明主设备正在给从设备传送数据。当为 1 时，表明主设备正在接收从设备来的数据。

图 17-10. 传送方向规定



17.6.4 $\overline{\text{ACK}}$

$\overline{\text{ACK}}$ 用于确认传送和接收设备的串行数据状态。

接收设备每接收到 8 位数据就返回 $\overline{\text{ACK}}$ 。

传送设备通常在传送 8 位数据后，接收 $\overline{\text{ACK}}$ 。当接收设备返回 $\overline{\text{ACK}}$ 时，认为正常接收，接着继续运行。 $\overline{\text{ACK}}$ 的监测通过 IICSn.ACKDn 位确认

当主设备是接收设备时，接收到终止数据后，并不返回 $\overline{\text{ACK}}$ ，而是产生停止条件。当从接收设备是设备，并且未返回 $\overline{\text{ACK}}$ 时，主设备既不产生停止条件，也不产生重启条件，接着停止当前数据传送。返回 $\overline{\text{ACK}}$ 失败由下列原因引起。

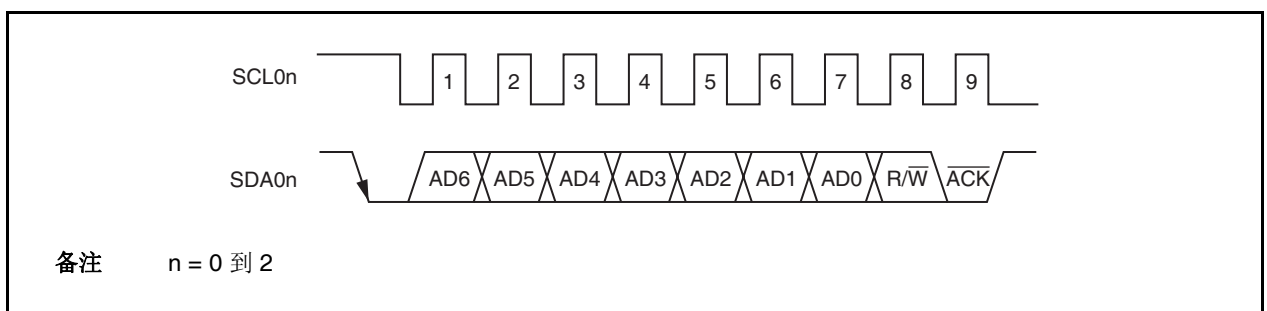
- (a) 接收不正常进行。
- (b) 接收到终止数据。
- (c) 特定的地址的接收设备（从设备）不存在。

第 9 个时钟期间，当接收设备设置 SDA0n 线为低电平时，产生 $\overline{\text{ACK}}$ （正常接收）。

当 IICSn.ACKEn 位置 1 时，允许自动产生 $\overline{\text{ACK}}$ 。紧跟着 7 位地址数据的第 8 位数据的传送能使得 IICSn.TRCn 置位。通常，设置 ACKEn 位为 1 进行接收 (TRCn 位 = 0)。

当从设备在接收时 (当 TRCn 位 = 0)，假如从设备不能接收数据或者不需要再接收数据了，那么清 ACKEn 位为 0，告知主设备，不再继续接收数据了。

与此类似，当主设备接收 (当 TRCn 位 = 0) 时，不再需要后续数据时，对 ACKEn 位清零，禁止 $\overline{\text{ACK}}$ 产生。这就告知从设备（传送设备）数据传送结束 (停止传送)。

图 17-11. $\overline{\text{ACK}}$ 

当接收到本地地址时，不管 ACKEn 位为何值，自动产生 $\overline{\text{ACK}}$ 。如果接收到的地址不是本地地址，不产生 $\overline{\text{ACK}}$ (NACK)。

当接收扩展码时，预先设置 ACKEn 位为 1，以产生 $\overline{\text{ACK}}$ 。

在数据接收期间， $\overline{\text{ACK}}$ 的产生方法依据等待时序的设置，如下所描述。

- 选择 8 个时钟等待 (IICSn.WTIMn 位 = 0):
在等待状态取消前，如果 ACKEn 位设置为 1，那么在 SCL0n 引脚的第 8 个时钟下降沿产生 $\overline{\text{ACK}}$ 。
- 选择 9 个时钟等待 (IICSn.WTIMn 位 = 1):
如果 ACKEn 位预先设置为 1，产生 $\overline{\text{ACK}}$ 。

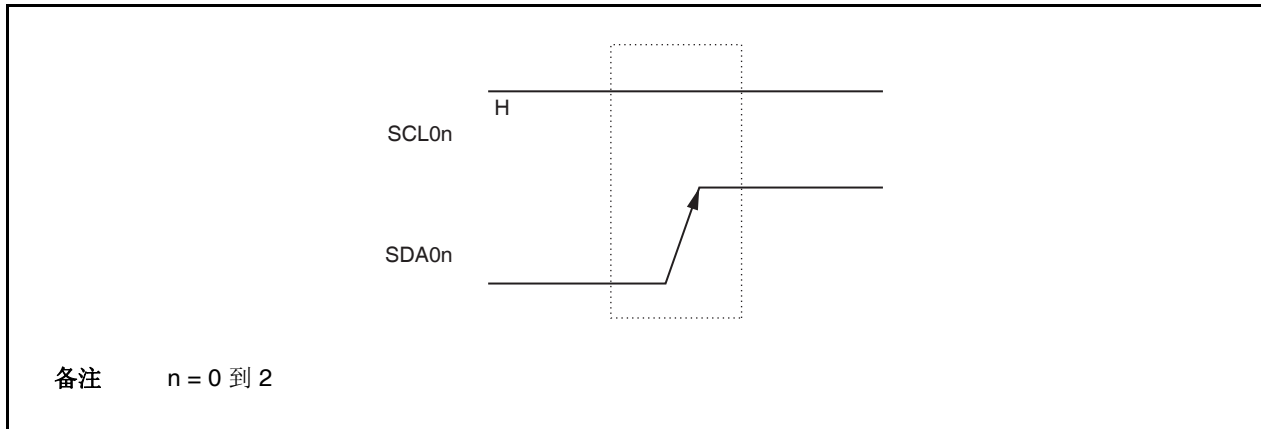
备注 n = 0 到 2

17.6.5 停止条件

当 SCL0n 引脚是高电平时，SDA0n 引脚从低电平变成高电平，那么就产生停止条件。(n = 0 到 2)。

当串行传送完成时，主设备向从设备输出停止条件。当用作从设备时，可以监测到开始条件。

图 17-12. 停止条件



当 IICCn.SPTn 位设置为 1 时，停止条件产生。当监测到停止条件时，如果 IICCn.SPIEn 位设置为 1 (n = 0 到 2)，则 IICSn.SPDn 位置 1，并产生中断请求信号 (INTIICn)。

17.6.6 等待状态

等待状态用来告知通信双方，一方（主设备或从设备）正在准备传送或接收数据(也就是说，处于等待状态)。
设置 SCL0n 引脚为低电平，这就告知通信双方处于等待状态。当双方的等待状态取消时，就可以开始下一个数据传送了(n = 0 到 2)。

图 17-13. 等待状态 (1/2)

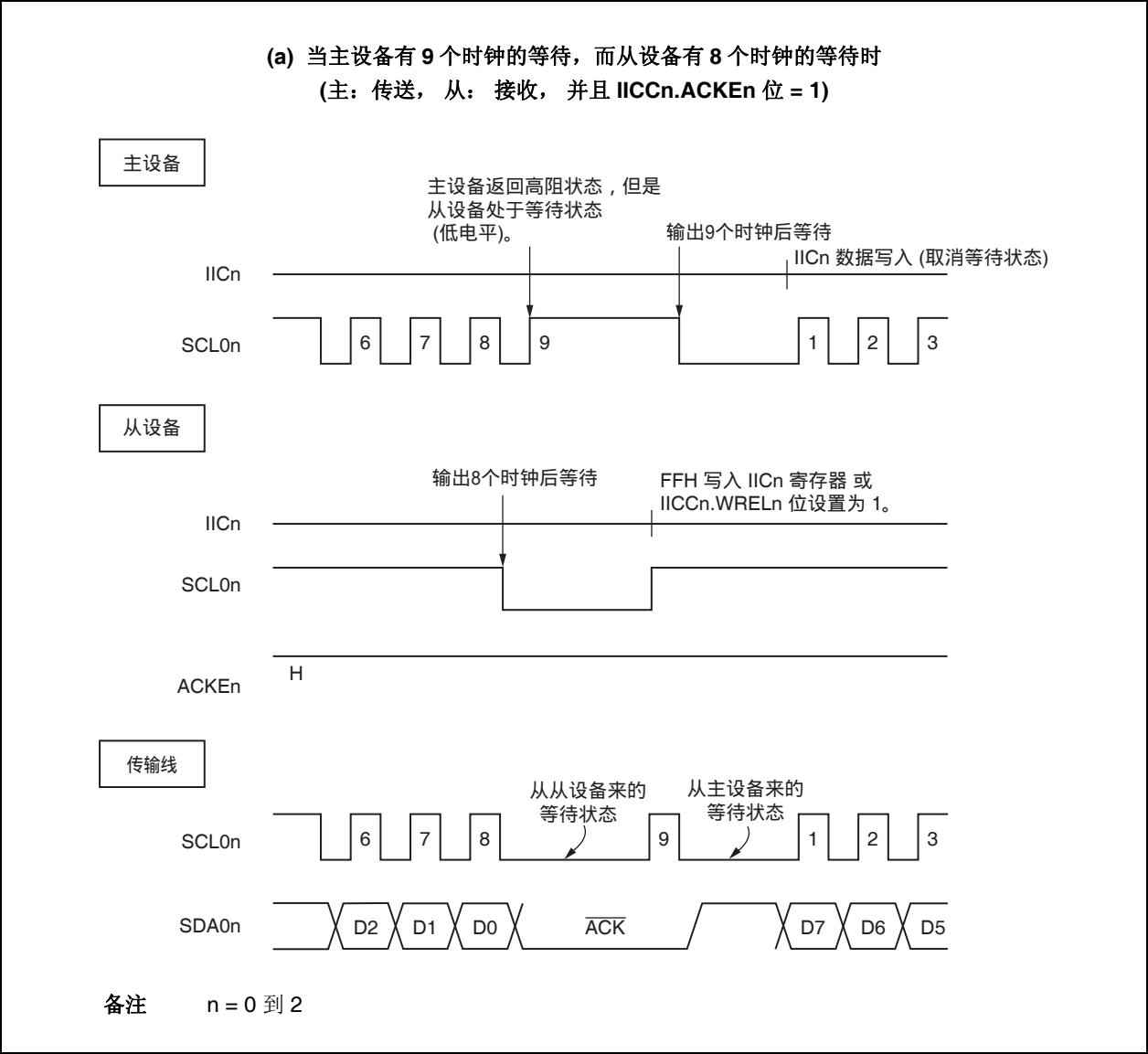
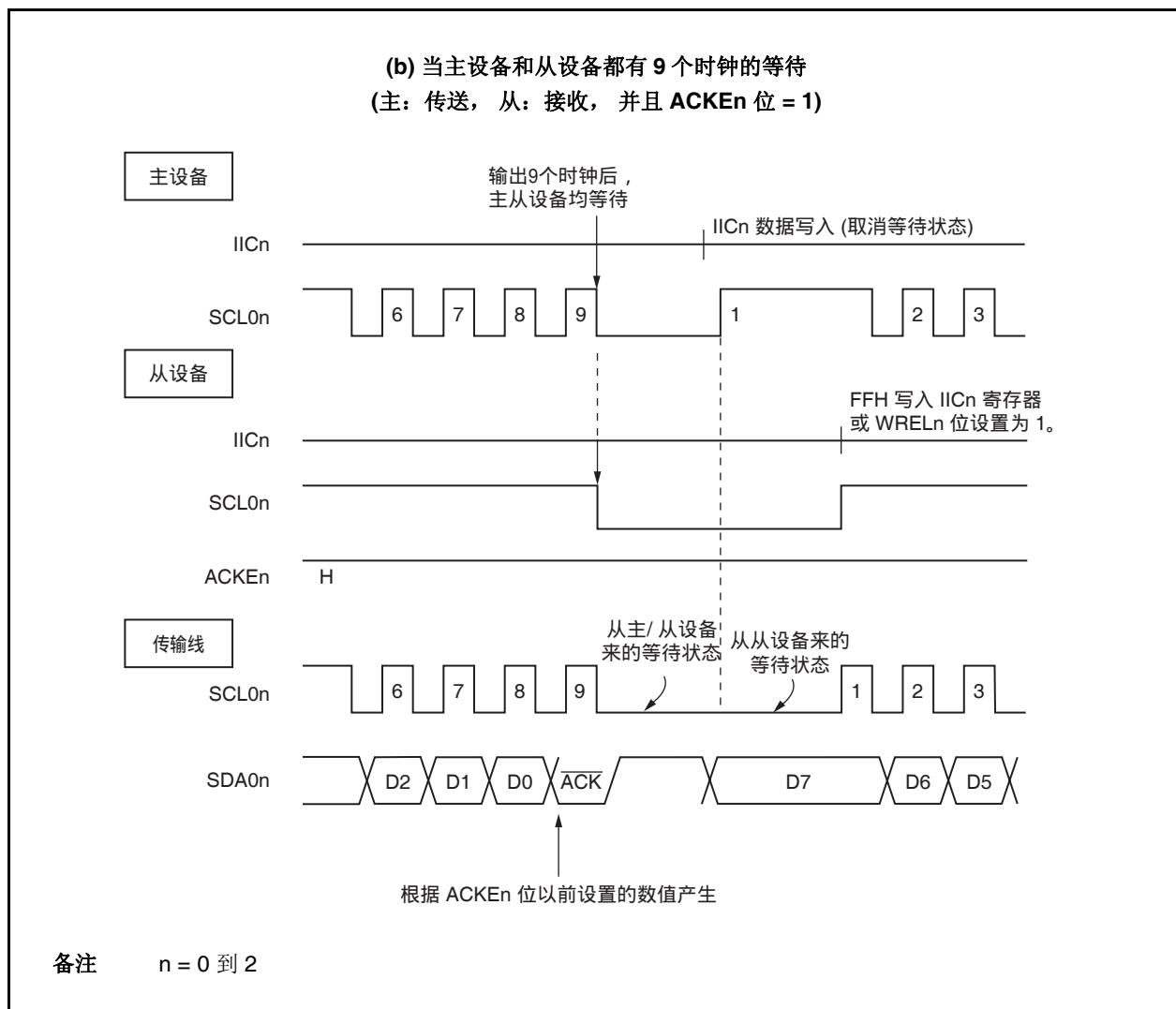


图 17-13. 等待状态 (2/2)



依据 IICn.WTIMn 位 (n = 0 到 2) 的设置, 等待状态可能自动产生。

通常, 在接收方, 当 IICn.WRELn 位置 1, 或者当 FFH 写入 IICn 寄存器时, 等待状态取消, 传送方写入数据到 IICn 寄存器来取消等待状态。

主设备也可以通过下列两者之一来取消等待状态。

- 设置 IICn.STTn 位为 1
- 设置 IICn.SPTn 位为 1

17.6.7 等待状态取消方法

在 I²C0n 情况下，等待状态可以通过下列方式正常取消 (n = 0 到 2)。

- 写入数据给 IICn 寄存器
- 设置 IICCn.WRELn 位为 1 (等待状态取消)
- 设置 IICCn.STTn 位为 1 (开始状态产生)
- 设置 IICCn.SPTn 位为 1 (停止状态产生)

如果进行了任何一个等待状态取消的动作，那么 I²C0n 将取消等待状态，并重启通信。

当取消等待状态，发送数据(包括地址)时，写数据给 IICn 寄存器。

取消等待状态后，为了接收数据，或者完成数据传送，设置 WRELn 位为 1。

取消等待状态后，为了产生重启条件，设置 STTn 位为 1。

取消等待状态后，为了产生停止条件，设置 SPTn 位为 1。

对于每个等待状态只能执行一次取消。

例如，通过设置 WRELn 位为 1 将等待状态取消后，如果有数据写入 IICn 寄存器，那么 SDA_n 线改变时序和 IICn 寄存器写入时序之间的冲突可能导致输出给 SDA_n 线的数据不正确。

即使在其它操作的情况下，如果通信中途停止，对 IICCn.IICEn 位的清零将停止通信，从而使等待状态可以被取消。

如果 I²C 总线因为噪声等原因而死锁，设置 IICCn.LRELn 位为 1 将引起通信操作退出，从而允许等待状态取消。

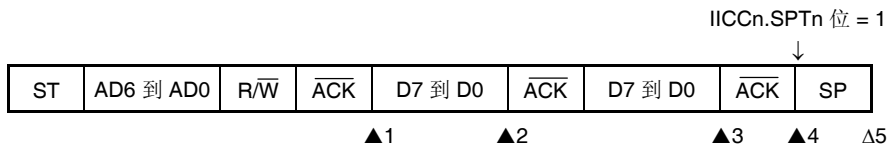
17.7 I²C 中断请求信号 (INTIICn)

下面显示在 INTIICn 中断请求信号产生时序和 INTIICn 信号时序(n = 0 到 2)中, IICSn 寄存器的数值。

17.7.1 主设备操作

(1) 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止 (正常传送/接收)

<1> 当 IICn.WTIMn 位 = 0



▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 1000X000B

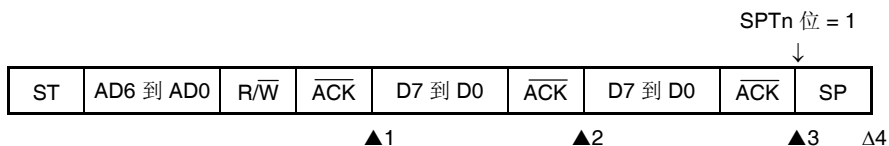
▲3: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)

▲4: IICSn 寄存器 = 1000XX00B

Δ5: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 一直产生
 Δ: 只有当 SPIEn 位 = 1 时产生
 X: 无关
 2. n = 0 到 2

<2> 当 WTIMn 位 = 1



▲1: IICSn 寄存器 = 1000X110B

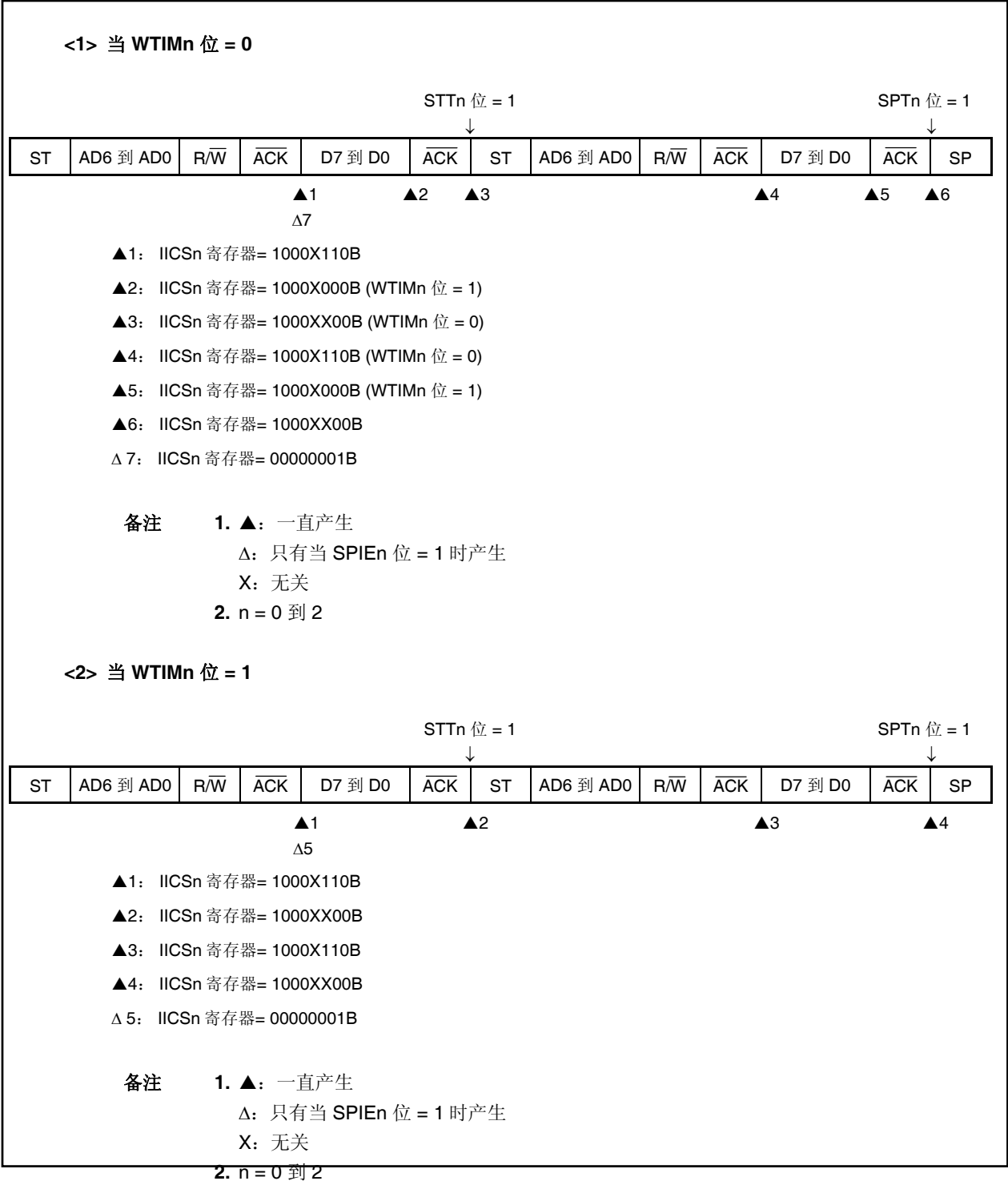
▲2: IICSn 寄存器 = 1000X100B

▲3: IICSn 寄存器 = 1000XX00B

Δ4: IICSn 寄存器 = 00000001B

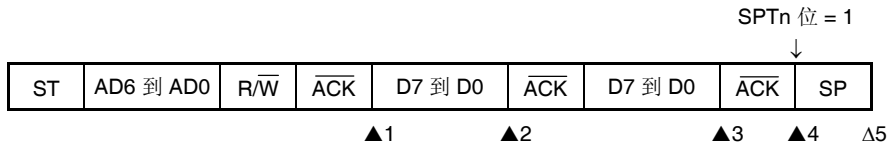
- 备注**
1. ▲: 一直产生
 Δ: 只有当 SPIEn 位 = 1 时产生
 X: 无关
 2. n = 0 到 2

(2) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止 (重启)



(3) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止 (扩展码传送)

<1> 当 WTIMn 位 = 0



▲1: IICSn 寄存器= 1010X110B

▲2: IICSn 寄存器= 1010X000B

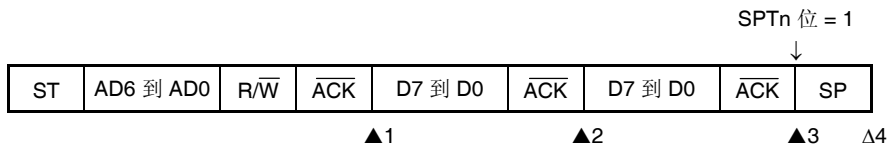
▲3: IICSn 寄存器= 1010X000B (WTIMn 位 = 1)

▲4: IICSn 寄存器= 1010XX00B

Δ5: IICSn 寄存器= 00000001B

- 备注**
1. ▲: 一直产生
 Δ: 只有当 SPIEn 位 = 1 时产生
 X: 无关
 2. n = 0 到 2

<2> 当 WTIMn 位 = 1



▲1: IICSn 寄存器= 1010X110B

▲2: IICSn 寄存器= 1010X100B

▲3: IICSn 寄存器= 1010XX00B

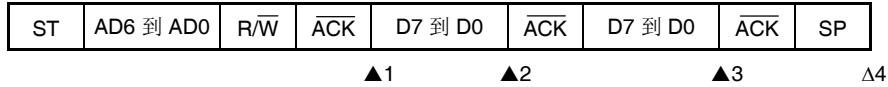
Δ4: IICSn 寄存器= 00000001B

- 备注**
1. ▲: 一直产生
 Δ: 只有当 SPIEn 位 = 1 时产生
 X: 无关
 2. n = 0 到 2

17.7.2 从设备操作 (当接收从地址数据时 (地址匹配))

(1) 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止

<1> 当 IICn.WTIMn 位 = 0



▲1: IICSn 寄存器= 0001X110B

▲2: IICSn 寄存器= 0001X000B

▲3: IICSn 寄存器= 0001X000B

Δ 4: IICSn 寄存器= 00000001B

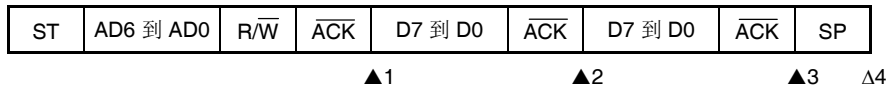
备注 1. ▲: 一直产生

Δ: 只有当 IICn.SPIEn 位 = 1 时产生

X: 无关

2. n = 0 到 2

<2> 当 WTIMn 位 = 1



▲1: IICSn 寄存器= 0001X110B

▲2: IICSn 寄存器= 0001X100B

▲3: IICSn 寄存器= 0001XX00B

Δ 4: IICSn 寄存器= 00000001B

备注 1. ▲: 一直产生

Δ: 只有当 SPIEn 位 = 1 时产生

X: 无关

2. n = 0 到 2

(2) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0 (重启后, 地址匹配)

ST	AD6 到 AD0	R/W	$\overline{\text{ACK}}$	D7 到 D0	$\overline{\text{ACK}}$	ST	AD6 到 AD0	R/W	$\overline{\text{ACK}}$	D7 到 D0	$\overline{\text{ACK}}$	SP
----	-----------	-----	-------------------------	---------	-------------------------	----	-----------	-----	-------------------------	---------	-------------------------	----

▲1

▲2

▲3

▲4

Δ5

▲1: IICSn 寄存器= 0001X110B

▲2: IICSn 寄存器= 0001X000B

▲3: IICSn 寄存器= 0001X110B

▲4: IICSn 寄存器= 0001X000B

Δ 5: IICSn 寄存器= 00000001B

备注

1. ▲: 一直产生

Δ: 只有当 SPIEn 位 = 1 时产生

X: 无关

2. n = 0 到 2

<2> 当 WTIMn 位 = 1 (重启后, 地址匹配)

ST	AD6 到 AD0	R/W	$\overline{\text{ACK}}$	D7 到 D0	$\overline{\text{ACK}}$	ST	AD6 到 AD0	R/W	$\overline{\text{ACK}}$	D7 到 D0	$\overline{\text{ACK}}$	SP
----	-----------	-----	-------------------------	---------	-------------------------	----	-----------	-----	-------------------------	---------	-------------------------	----

▲1

▲2

▲3

▲4

Δ5

▲1: IICSn 寄存器= 0001X110B

▲2: IICSn 寄存器= 0001XX00B

▲3: IICSn 寄存器= 0001X110B

▲4: IICSn 寄存器= 0001XX00B

Δ 5: IICSn 寄存器= 00000001B

备注

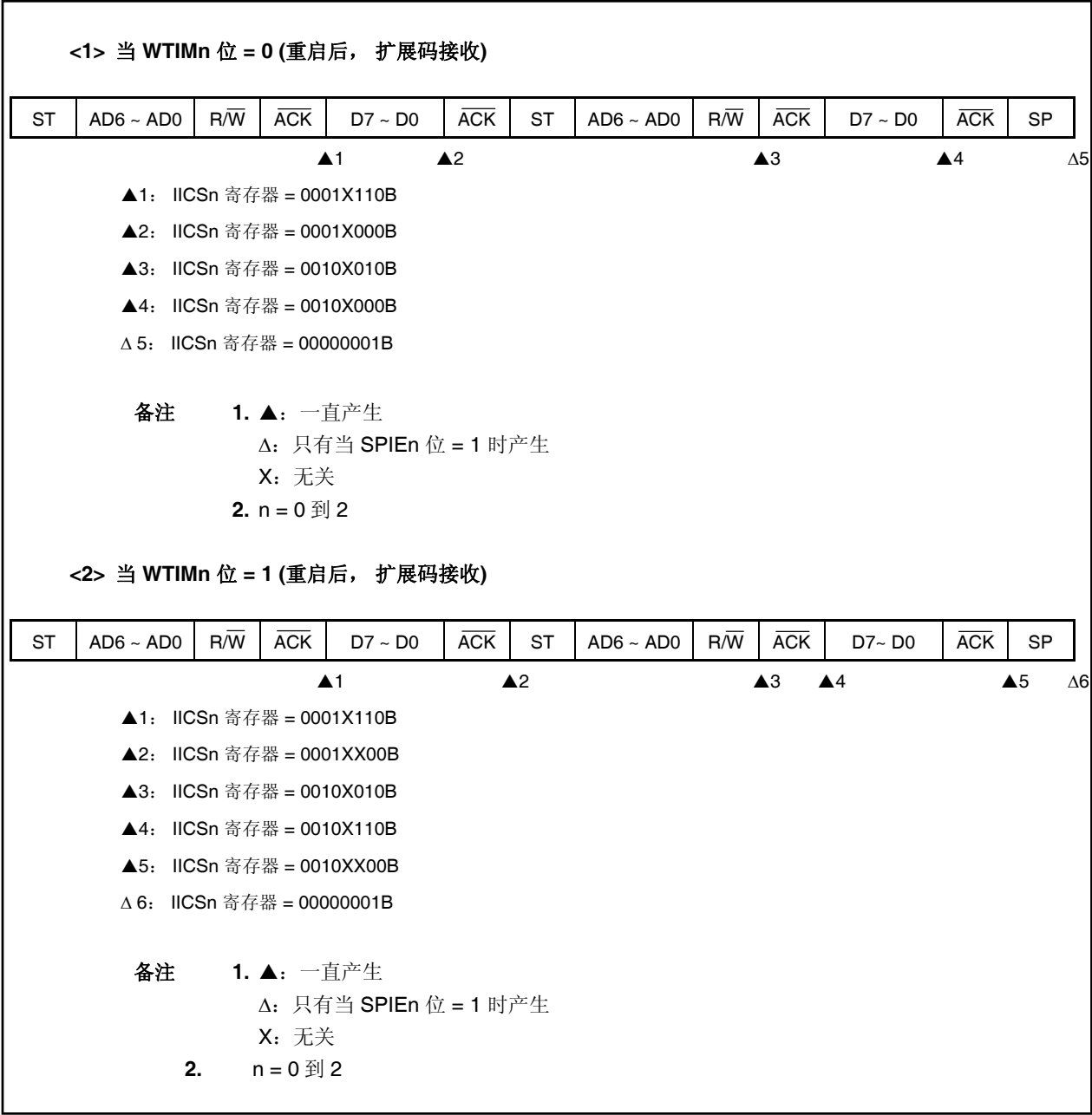
1. ▲: 一直产生

Δ: 只有当 SPIEn 位 = 1 时产生

X: 无关

2. n = 0 到 2

(3) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 代码 ~ 数据 ~ 停止



(4) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0 (重启后, 地址不匹配 (= 非扩展码))

ST	AD6 ~ AD0	R/W	$\overline{\text{ACK}}$	D7 ~ D0	$\overline{\text{ACK}}$	ST	AD6 ~ AD0	R/W	$\overline{\text{ACK}}$	D7 ~ D0	$\overline{\text{ACK}}$	SP
----	-----------	-----	-------------------------	---------	-------------------------	----	-----------	-----	-------------------------	---------	-------------------------	----

▲1

▲2

▲3

Δ4

▲1: IICSn 寄存器 = 0001X110B

▲2: IICSn 寄存器 = 0001X000B

▲3: IICSn 寄存器 = 00000X10B

Δ4: IICSn 寄存器 = 00000001B

备注

1. ▲: 一直产生

Δ: 只有当 SPIEn 位 = 1 时产生

X: 无关

2. n = 0 到 2

<2> 当 WTIMn 位 = 1 (重启后, 地址不匹配 (= 非扩展码))

ST	AD6 ~ AD0	R/W	$\overline{\text{ACK}}$	D7 ~ D0	$\overline{\text{ACK}}$	ST	AD6 ~ AD0	R/W	$\overline{\text{ACK}}$	D7 ~ D0	$\overline{\text{ACK}}$	SP
----	-----------	-----	-------------------------	---------	-------------------------	----	-----------	-----	-------------------------	---------	-------------------------	----

▲1

▲2

▲3

Δ4

▲1: IICSn 寄存器 = 0001X110B

▲2: IICSn 寄存器 = 0001XX00B

▲3: IICSn 寄存器 = 00000X10B

Δ4: IICSn 寄存器 = 00000001B

备注

1. ▲: 一直产生

Δ: 只有当 SPIEn 位 = 1 时产生

X: 无关

2. n = 0 到 2

17.7.3 从设备操作 (当接收扩展码时)

(1) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止

<1> 当 IICn.WTIMn 位 = 0

ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	D7 ~ D0	ACK	SP
			▲1		▲2		▲3	Δ4

▲1: IICSn 寄存器 = 0010X010B
▲2: IICSn 寄存器 = 0010X000B
▲3: IICSn 寄存器 = 0010X000B
Δ 4: IICSn 寄存器 = 00000001B

备注 1. ▲: 一直产生
 Δ: 只有当 IICn.SPIEn 时 = 1 产生
 X: 无关
 2. n = 0 到 2

<2> 当 WTIMn 位 = 1

ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	D7 ~ D0	ACK	SP
			▲1	▲2		▲3	▲4	Δ5

▲1: IICSn 寄存器 = 0010X010B
▲2: IICSn 寄存器 = 0010X110B
▲3: IICSn 寄存器 = 0010X100B
▲4: IICSn 寄存器 = 0010XX00B
Δ 5: IICSn 寄存器 = 00000001B

备注 1. ▲: 一直产生
 Δ: 只有当 SPIEn 位 = 1 时产生
 X: 无关
 2. n = 0 到 2

(2) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0 (重启后, 地址匹配)

ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	SP
----	-----------	-----	-----	---------	-----	----	-----------	-----	-----	---------	-----	----

▲1

▲2

▲3

▲4

Δ5

▲1: IICSn 寄存器 = 0010X010B

▲2: IICSn 寄存器 = 0010X000B

▲3: IICSn 寄存器 = 0001X110B

▲4: IICSn 寄存器 = 0001X000B

Δ 5: IICSn 寄存器 = 00000001B

备注

1. ▲: 一直产生

Δ: 只有当 SPIEn 位 = 1 时产生

X: 无关

2. n = 0 到 2

<2> 当 WTIMn 位 = 1 (重启后, 地址匹配)

ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	SP
----	-----------	-----	-----	---------	-----	----	-----------	-----	-----	---------	-----	----

▲1

▲2

▲3

▲4

▲5

Δ6

▲1: IICSn 寄存器 = 0010X010B

▲2: IICSn 寄存器 = 0010X110B

▲3: IICSn 寄存器 = 0010XX00B

▲4: IICSn 寄存器 = 0001X110B

▲5: IICSn 寄存器 = 0001XX00B

Δ 6: IICSn 寄存器 = 00000001B

备注

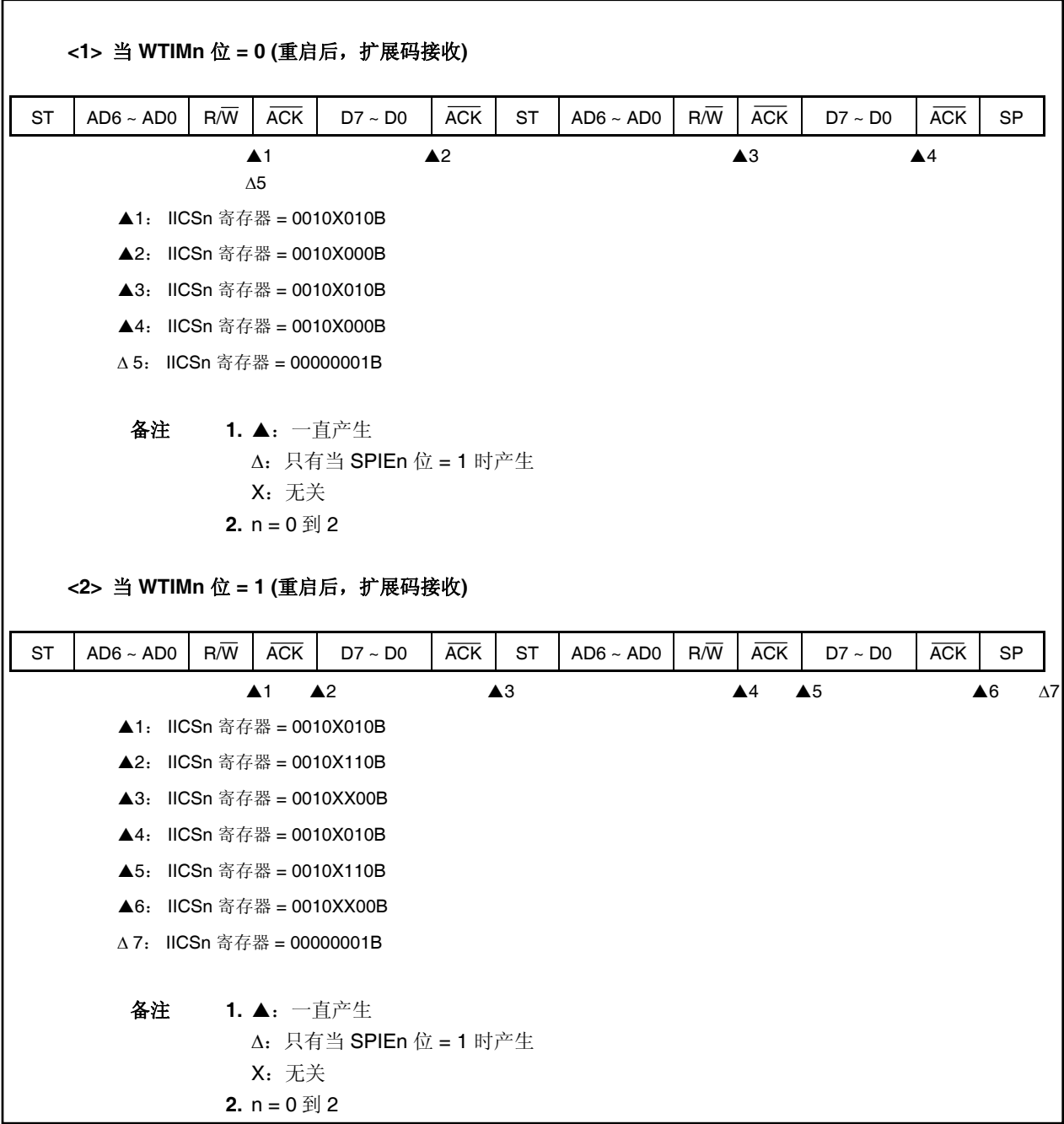
1. ▲: 一直产生

Δ: 只有当 SPIEn 位 = 1 时产生

X: 无关

2. n = 0 到 2

(3) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 代码 ~ 数据 ~ 停止



(4) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0 (重启后, 地址不匹配 (= 非扩展码))

ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	SP
----	-----------	-----	-----	---------	-----	----	-----------	-----	-----	---------	-----	----

▲1

▲2

▲3

Δ4

▲1: 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. ▲: 一直产生

Δ: 只有当 SPIEn 位 = 1 时产生

X: 无关

2. n = 0 到 2

<2> 当 WTIMn 位 = 1 (重启后, 地址不匹配 (= 非扩展码))

ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	SP
----	-----------	-----	-----	---------	-----	----	-----------	-----	-----	---------	-----	----

▲1

▲2

▲3

▲4

Δ5

▲1: 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. ▲: 一直产生

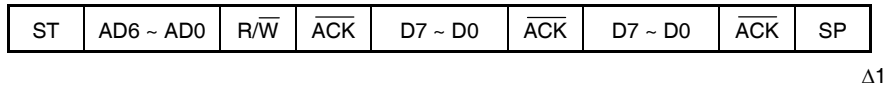
Δ: 只有当 SPIEn 位 = 1 时产生

X: 无关

2. n = 0 到 2

17.7.4 不通信时操作

(1) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止



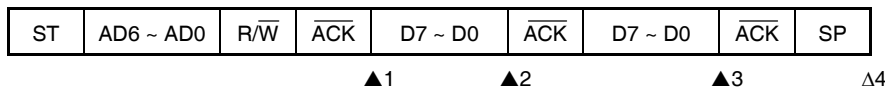
Δ 1: IICSn 寄存器 = 00000001B

备注 1. Δ: 只有当 SPIEn 位 = 1 时产生
 2. n = 0 到 2

17.7.5 仲裁失败操作 (仲裁失败后作为从设备操作)

(1) 在从地址数据传送期间仲裁失败

<1> 当 IICSn.WTIMn 位 = 0



▲1: IICSn 寄存器 = 0101X110B (例如: 中断服务处理期间读 IICSn.ALDn 位)

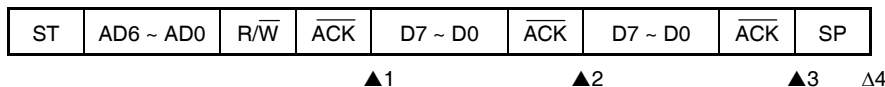
▲2: IICSn 寄存器 = 0001X000B

▲3: IICSn 寄存器 = 0001X000B

Δ 4: IICSn 寄存器 = 00000001B

备注 1. ▲: 一直产生
 Δ: 只有当 IICSn.SPIEn 位 = 1 时产生
 X: 无关
 2. n = 0 到 2

<2> 当 WTIMn 位 = 1



▲1: IICSn 寄存器 = 0101X110B (例如: 中断服务处理期间读 ALDn 位)

▲2: IICSn 寄存器 = 0001X100B

▲3: IICSn 寄存器 = 0001XX00B

Δ 4: IICSn 寄存器 = 00000001B

备注 1. ▲: 一直产生
 Δ: 只有当 SPIEn 位 = 1
 X: 无关
 2. n = 0 到 2

(2) 扩展码传送期间仲裁失败

<1> 当 WTIMn 位 = 0

ST	AD6 ~ AD0	R/W	$\overline{\text{ACK}}$	D7 ~ D0	$\overline{\text{ACK}}$	D7 ~ D0	$\overline{\text{ACK}}$	SP
			▲1		▲2		▲3	Δ4

▲1: IICSn 寄存器 = 0110X010B (例如: 中断服务处理期间读 ALDn 位)

▲2: IICSn 寄存器 = 0010X000B

▲3: IICSn 寄存器 = 0010X000B

Δ4: IICSn 寄存器 = 00000001B

- 备注
1. ▲: 一直产生
Δ: 只有当 SPIEn 位 = 1 时产生
X: 无关
 2. n = 0 到 2

<2> 当 WTIMn 位 = 1

ST	AD6 ~ AD0	R/W	$\overline{\text{ACK}}$	D7 ~ D0	$\overline{\text{ACK}}$	D7 ~ D0	$\overline{\text{ACK}}$	SP
			▲1	▲2		▲3	▲4	Δ5

▲1: IICSn 寄存器 = 0110X010B (例如: 中断服务处理期间读 ALDn 位)

▲2: IICSn 寄存器 = 0010X110B

▲3: IICSn 寄存器 = 0010X100B

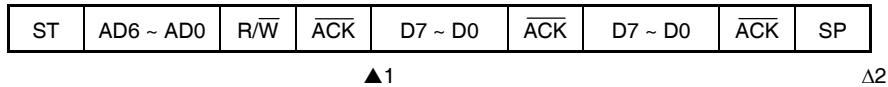
▲4: IICSn 寄存器 = 0010XX00B

Δ5: IICSn 寄存器 = 00000001B

- 备注
1. ▲: 一直产生
Δ: 只有当 SPIEn 位 = 1 时产生
X: 无关
 2. n = 0 到 2

17.7.6 仲裁失败时的操作(仲裁失败后无通信)

(1) 从地址数据传送期间仲裁失败



▲1

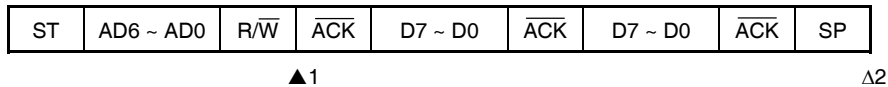
Δ2

▲1: IICSn 寄存器 = 01000110B (例如: 中断服务处理期间读 IICSn.ALDn 位)

Δ 2: IICSn 寄存器 = 00000001B

- 备注
1. ▲: 一直产生
Δ: 只有当 IICn.SPIEn 位 = 1 时产生
 2. n = 0 到 2

(2) 扩展码传送期间仲裁失败



▲1

Δ2

▲1: IICSn 寄存器 = 0110X010B (例如: 中断服务处理期间读 ALDn 位)

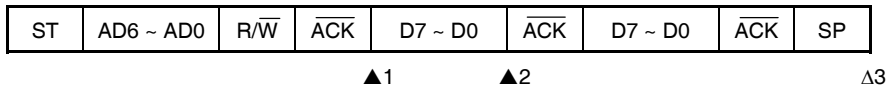
IICn.LRELn 位由软件置 1

Δ 2: IICSn 寄存器 = 00000001B

- 备注
1. ▲: 一直产生
Δ: 只有当 SPIEn 位 = 1 时产生
X: 无关
 2. n = 0 到 2

(3) 数据传送期间仲裁失败

<1> 当 IICn.WTIMn 位 = 0



▲1: IICSn 寄存器 = 10001110B

▲2: IICSn 寄存器 = 01000000B (例如: 中断服务处理期间读 ALDn 位)

Δ 3: IICSn 寄存器 = 00000001B

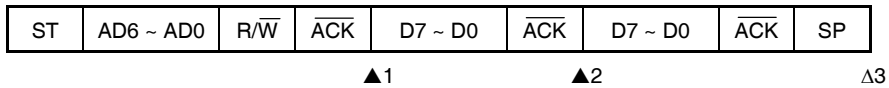
备注

1. ▲: 一直产生

Δ: 只有当 SPIEn 位 = 1 时产生

2. n = 0 到 2

<2> 当 WTIMn 位 = 1



▲1: IICSn 寄存器 = 10001110B

▲2: IICSn 寄存器 = 01000100B (例如: 中断服务处理期间读 ALDn 位)

Δ 3: IICSn 寄存器 = 00000001B

备注

1. ▲: 一直产生

Δ: 只有当 SPIEn 位 = 1 时产生

2. n = 0 到 2

(4) 数据传送期间由于重启条件致使仲裁失败

<1> 非扩展码 (例如： 地址不匹配)

ST	AD6 ~ AD0	R/W	ACK	D7 ~ Dn	ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	SP
----	-----------	-----	-----	---------	----	-----------	-----	-----	---------	-----	----

▲1

▲2

Δ3

▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 01000110B (例如： 中断服务处理期间读 ALDn 位)

Δ 3: IICSn 寄存器 = 00000001B

备注

1. ▲： 一直产生

Δ: 只有当 SPIEn 位 = 1 时产生

X: 无关

2. Dn = D6 ~ D0

n = 0 到 2

<2> 扩展码

ST	AD6 ~ AD0	R/W	ACK	D7 ~ Dn	ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	SP
----	-----------	-----	-----	---------	----	-----------	-----	-----	---------	-----	----

▲1

▲2

Δ3

▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 0110X010B (例如： 中断服务处理期间读 ALDn 位)

IICcn.LRELn 位由软件置 1

Δ 3: IICSn 寄存器 = 00000001B

备注

1. ▲： 一直产生

Δ: 只有当 SPIEn 位 = 1 时产生

X: 无关

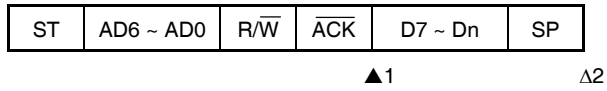
2. Dn = D6 ~ D0

n = 0 到 2

用户手册 U17715CA2V0UD

603

(5) 数据传输期间由于停止条件致使仲裁失败



▲1: IICSn 寄存器 = 1000X110B

Δ 2: IICSn 寄存器 = 01000001B

备注

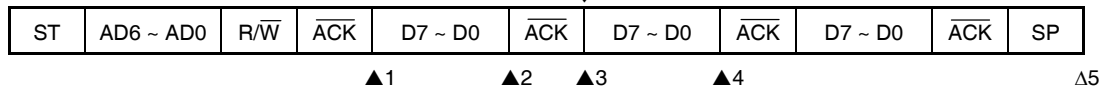
1. ▲: 一直产生
 Δ: 只有当 SPIEn 位 = 1 时产生
 X: 无关
2. Dn = D6 ~ D0
 n = 0 到 2

(6) 当试图产生重启条件时由于 SDA0n 引脚的低电平致使仲裁失败

★

<1> 当 WTIMn 位 = 0

IICn.STTn 位 = 1



▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)

▲3: IICSn 寄存器 = 1000XX00B (WTIMn 位 = 0)

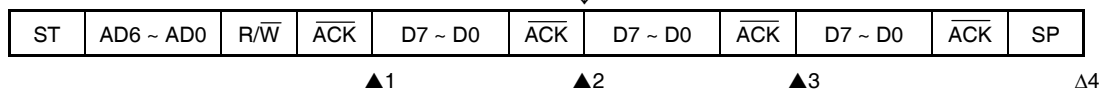
▲4: IICSn 寄存器 = 01000000B (例如: 中断服务处理期间读 ALDn 位)

Δ5: IICSn 寄存器 = 00000001B

- 备注
1. ▲: 一直产生
Δ: 只有当 SPIEn 位 = 1 时产生
X: 无关
 2. n = 0 到 2

<2> 当 WTIMn 位 = 1

IICn.STTn 位 = 1



▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 1000XX00B

▲3: IICSn 寄存器 = 01000100B (例如: 中断服务处理期间读 ALDn 位)

Δ4: IICSn 寄存器 = 00000001B

- 备注
1. ▲: 一直产生
Δ: 只有当 SPIEn 位 = 1 时产生
X: 无关
 2. n = 0 到 2

(7) 试图产生重启条件时由于停止条件致使仲裁失败

★ <1> 当 WTIMn 位 = 0

STTn 位 = 1
↓

ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	SP
----	-----------	-----	-----	---------	-----	----

▲1 ▲2 ▲3 Δ4

▲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. ▲: 一直产生
 Δ: 只有当 SPIEn 位 = 1 时产生
 X: 无关
 2. n = 0 到 2

<2> 当 WTIMn 位 = 1

STTn 位 = 1
↓

ST	AD6 ~ AD0	R/W	ACK	D7 ~ D0	ACK	SP
----	-----------	-----	-----	---------	-----	----

▲1 ▲2 Δ3

▲1: IICSn 寄存器 = 1000X110B
 ▲2: IICSn 寄存器 = 1000XX00B
 Δ3: IICSn 寄存器 = 01000001B

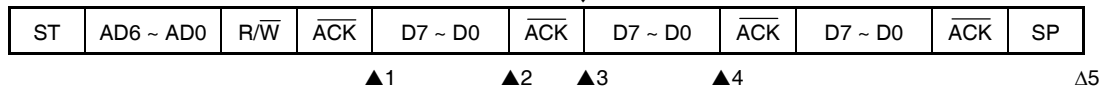
备注 1. ▲: 一直产生
 Δ: 只有当 SPIEn 位 = 1 时产生
 X: 无关
 2. n = 0 到 2

(8) 试图产生停止条件时由于 SDA0n 引脚的低电平致使仲裁失败

★

<1> 当 WTIMn 位 = 0

IICn.SPTn 位 = 1



▲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: 中断服务处理期间读 ALDn 位)

Δ5: IICSn 寄存器 = 00000001B

备注

1. ▲: 一直产生

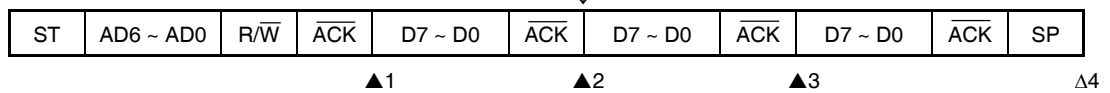
Δ: 只有当 SPIEn 位 = 1 时产生

X: 无关

2. n = 0 到 2

<2> 当 WTIMn 位 = 1

IICn.SPTn 位 = 1



▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 1000XX00B

▲3: IICSn 寄存器 = 01000000B (例如: 中断服务处理期间读 ALDn 位)

Δ4: IICSn 寄存器 = 00000001B

备注

1. ▲: 一直产生

Δ: 只有当 SPIEn 位 = 1

X: 无关

2. n = 0 到 2

17.8 中断请求信号 (INTIICn) 的产生时序和等待控制

INTIICn 寄存器产生和相应等待控制所依据的时序由 IICn.WTIMn 位的设置决定，如下所示(n = 0 to 2)。

表 17-3. 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	数据传送	地址	数据接收	数据传送
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 1** 只有与 SVAn 寄存器设置的地址匹配时，在第 9 个时钟下降沿，从设备的 INTIICn 信号和等待区间才产生。这时，不管 IICn.ACKEn 位的设置为何， $\overline{\text{ACK}}$ 均产生。对于接收扩展码的从设备来说，INTIICn 信号在第 8 个时钟下降沿产生。

重启后地址不匹配时，INTIICn 信号在第 9 个时钟的下降沿产生，但无等待产生。
2. 如果接收到的地址与 SVAn 寄存器的内容不匹配，并且也没有接收到扩展码，那么既无 INTIICn 信号也无等待产生。

- 备注** 1. 表中的数字表示串行时钟的时钟信号的个数。中断请求和等待控制都和这些时钟信号的下降沿同步。

2. n = 0 到 2

(1) 地址传送/接收期间

- 从设备操作：不管 WTIMn 位的设置为何，中断和等待时序一定。
- 主设备操作：不管 WTIMn 位的设置为何，中断和等待时序在第 9 个时钟下降沿产生。

(2) 数据接收期间

- 主/从设备操作：中断和等待时序由 WTIMn 位决定。

(3) 数据传送期间

- 主/从设备操作：中断和等待时序由 WTIMn 位决定。

(4) 等待取消方法

四种等待取消方法如下。

- 通过设置 IICn.WRELn 位为 1
- 通过写 IICn 寄存器
- 通过开始条件设置 (IICn.STTn 位 = 1)[※]
- 通过停止条件设置 (IICn.SPTn 位 = 1)[※]

注 只对主设备设置

当选择 8 个时钟等待(WTIMn 位 = 0)时，ACK 产生与否必须要优先于等待取消决定。

备注 n = 0 到 2

(5) 停止条件监测

当监测到停止条件时，INTIICn 信号产生。

备注 n = 0 到 2

17.9 地址匹配监测方法

I²C 总线模式中，主设备通过通过传送相应的从地址选择特定的从设备。

地址匹配监测由硬件自动完成。当本地地址已经设置入 SVAn 寄存器，并且当设置入 SVAn 寄存器的地址与主设备发送的从地址匹配时，或者接收到扩展码时，INTIICn 信号产生。(n = 0 到 2)。

17.10 错误监测

I²C 总线中，数据传送期间串行数据总线引脚(SDA0n)的状态由传送设备的 IICn 寄存器捕捉，因此先于传送的 IICn 寄存器的数据与传送的 IICn 数据进行比较，从而允许传送错误监测。相比较的数据值不匹配时，判决为发生传送错误。

(n = 0 到 2)。

17.11 扩展码

- (1) 当接收地址的高 4 位为 0000 或者 1111，扩展码标志(IICSn.EXCn 位) 设置为扩展码接收，并且第 8 个时钟下降沿产生中断请求 (INTIICn) (n = 0 到 2)。

存储在 SVAn 寄存器里的本地地址不受影响。

- (2) 如果 SVAn 寄存器通过 10 位地址传送设置为 11110xx0，并且 11110xx0 从主设备传送，结果如下。注意 INTIICn 信号在第 8 个时钟下降沿产生(n = 0 到 2)。

- 数据匹配的高 4 位：EXCn 位 = 1
- 数据匹配的 7 位：IICSn.COIn 位 = 1

- (3) 既然中断请求信号产生后的处理，根据扩展码后面的数据而并不相同，所以由软件来完成这些处理。

例如，接收到扩展码后，不期望作为从设备操作，那么设置 IICn.LRELn 位为 1，CPU 将进入下一个通信等待状态。

表 17-4. 扩展码位定义

从地址	R/W 位	描述
0000 000	0	通用调用地址
0000 000	1	开始字节
0000 001	X	CBUS 地址
0000 010	X	预留给不同总线格式的地址
1111 0xx	X	10 位从地址规定

17.12 仲裁

当几个主设备同时产生开始条件时(当 IICn.STTn 位在 IICSn.STDn 位设置为 1 之前设置为 1)，主设备之间的通信在时钟的个数进行调整直到数据有所不同时为止一直进行。这种操作称为仲裁(n = 0 到 2)。

当主设备之一仲裁失败时，通过产生仲裁失败所依据的时序将仲裁失败标志(IICSn.ALDn 位)置 1，SCL0n 和 SDA0n 线均设置为高阻态，从而释放总线(n = 0 到 2)。

仲裁失败的监测基于下一个中断请求信号(INTIICn)的时序(当监测到停止条件等等，第 8 个或者第 9 个时钟)和 ALDn 位为 1 的设置。这些都由软件设置(n = 0 到 2)。

中断请求信号时序的详细情况，请看 17.7 I²C 中断请求信号(INTIICn)。

图 17-14. 仲裁时序实例

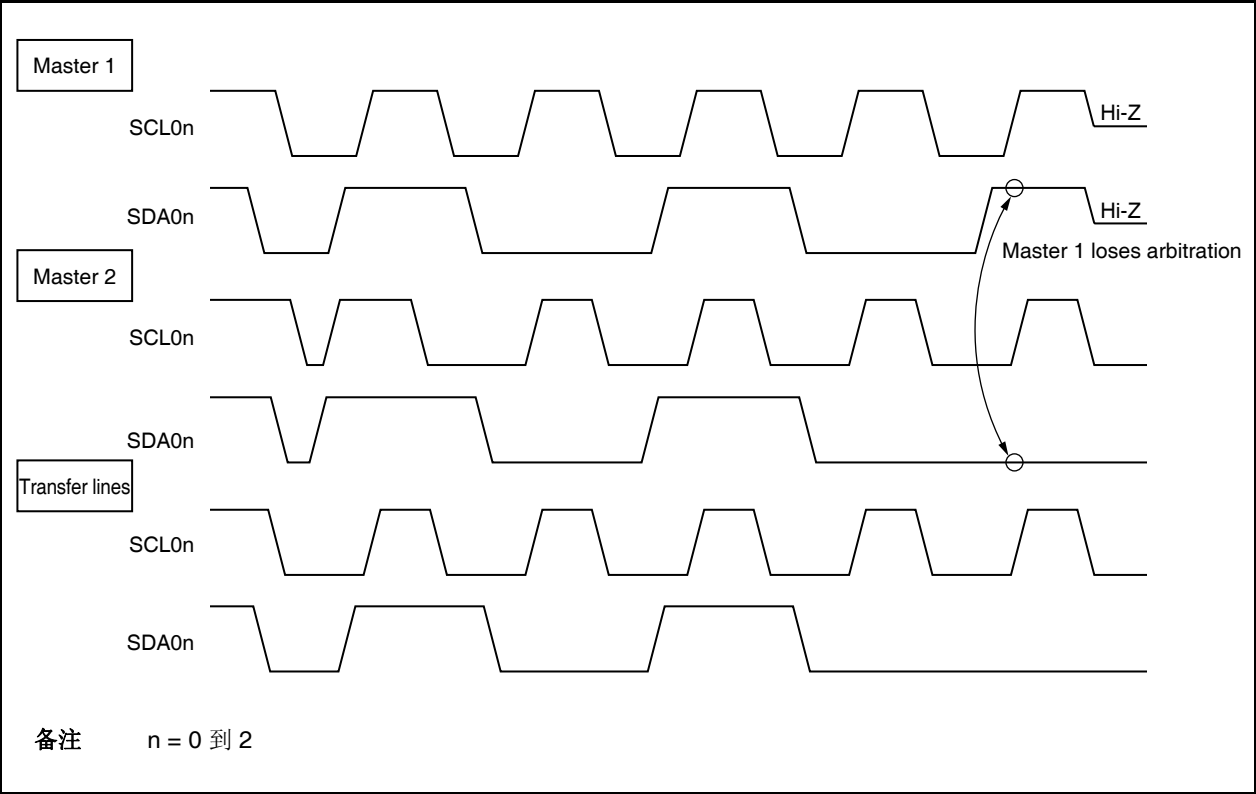


表 17-5. 仲裁和中断请求信号产生时序期间的状态

仲裁期间状态	中断请求产生时序
地址传送中	字节传送后的第 8 个或者第 9 个时钟下降沿 ¹
地址传送后读/写数据	
传送地址码	
扩展码传送后读/写数据	
传送数据	
数据接收后 \overline{ACK} 传送期间	
数据传送期间当监测到重启条件时	
数据传送期间当监测到停止条件时	监测到停止条件 (当 IICn.SPIEn 位 = 1) ²
SDA0n 引脚为低电平时, 试图产生重启条件	字节传送后第 8 个或者第 9 个时钟的下降沿 ¹
监测到停止条件时, 试图产生重启条件	监测到停止条件 (当 IICn.SPIEn 位 = 1) ²
DSA0n 引脚为低电平时, 试图产生停止条件	字节传送后第 8 个或者第 9 个时钟的下降沿 ¹
SCL0n 引脚为低电平时, 试图产生重启条件	

- 注
1. 当 IICn.WTIMn 位 = 1, 在第 9 个时钟的下降沿产生 INTIICn 信号。当 WTIMn 位 = 0, 并且接收到扩展码的从地址时, 在第 9 个时钟的下降沿产生 INTIICn 信号(n = 0 to 2)。
 2. 仲裁可能产生时, 设置 SPIEn 位为 1, 用于主设备操作(n = 0 到 2)。

17.13 唤醒功能

I²C 总线从功能就是当接收到本地地址和扩展码时产生中断请求信号(INTIICn)的功能。

这个功能使处理更有效, 当地址不匹配时, 避免了不必要的 INTIICn 信号产生。

监测到开始条件时, 设置唤醒待机模式。由于可能仲裁失败而致使主设备(已经产生开始条件)变为从设备的情况下的地址传送期间, 唤醒待机模式有效。

然而, 当检测到开始条件时, 不管唤醒功能而设置 IICn.SPIEn 位, 并且这决定了 INTIICn 信号允许或者禁止(n = 0 到 2)。

17.14 通信保留

17.14.1 通信保留功能允许时(IICFn.IICRSVn 位 = 0)

不使用总线期间为了开始主设备通信，可以进行通信保留而允许在总线释放时传送开始条件。总线不时用时有两种模式。

- 当仲裁导致既不进行主操作也不进行从操作。
- 接收到扩展码，并且从操作禁止时(当 IICFn.LRELn 位设置为 1 时，不返回 $\overline{\text{ACK}}$ ，总线释放) (n = 0 到 2)。

当总线不使用时，如果 IICFn.STTn 位设置为 1，总线释放后(监测到停止条件后)，自动产生开始条件，设置等待状态。

检测到总线释放时(监测到停止条件时)，写 IICFn 寄存器使主地址传送开始。这时，IICFn.SPIEn 位应当设置为 1 (n = 0 到 2)。

当 STTn 已经设置为 1 时，操作模式(作为开始条件或者作为通信保留) 根据总线状态决定 (n = 0 到 2)。

如果总线已经释放产生开始条件

如果总线还没有释放 (待机模式)通信保留

为了监测已经为 STTn 位决定了何种操作模式，设置 STTn 位为 1，接着为等待时间，然后检验 IICSn.MSTS_n 位 (n = 0 到 2)。

由软件设置的等待时间，由表 17-6 所示。等待时间由 IICCLn 寄存器的 SMCn，CLn1 和 CLn0 以及 IICXn.CLXn 位设置 (n = 0 到 2)。

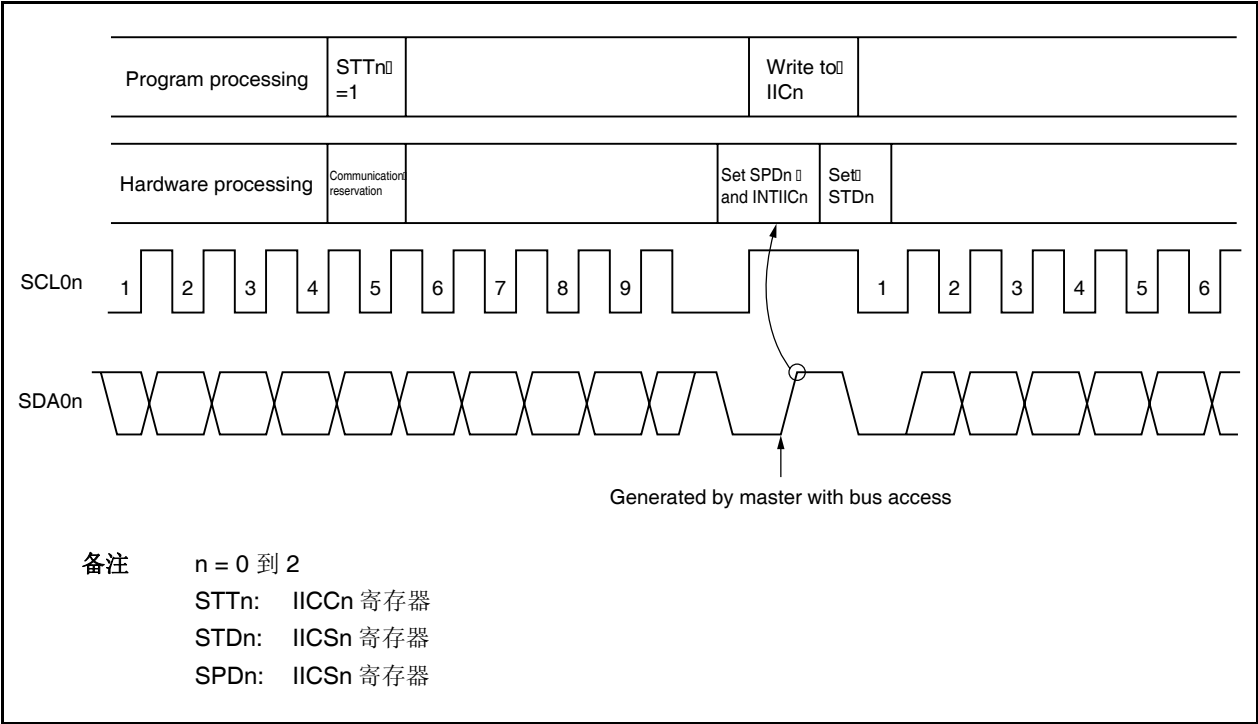
表 17-6. 等待时间

时钟选择	CLXn	SMCn	CLn1	CLn0	等待周期
f _{xx} (当 OCKSm = 18H 时)	0	0	0	0	26 个时钟
f _{xx} /2 (当 OCKSm = 10H 时)	0	0	0	0	52 个时钟
f _{xx} /3 (当 OCKSm = 11H 时)	0	0	0	0	78 个时钟
f _{xx} /4 (当 OCKSm = 12H 时)	0	0	0	0	104 个时钟
f _{xx} /5 (当 OCKSm = 13H 时)	0	0	0	0	130 个时钟
f _{xx} (当 OCKSm = 18H 时)	0	0	0	1	47 个时钟
f _{xx} /2 (当 OCKSm = 10H 时)	0	0	0	1	94 个时钟
f _{xx} /3 (当 OCKSm = 11H 时)	0	0	0	1	141 个时钟
f _{xx} /4 (当 OCKSm = 12H 时)	0	0	0	1	188 个时钟
f _{xx}	0	0	1	0	47 个时钟
<R> f _{xx} (当 OCKSm = 18H 时)	0	0	1	1	37 个时钟
<R> f _{xx} /2 (当 OCKSm = 10H 时)	0	0	1	1	74 个时钟
<R> f _{xx} /3 (当 OCKSm = 11H 时)	0	0	1	1	111 个时钟
f _{xx} (当 OCKSm = 18H 时)	0	1	0	×	16 个时钟
f _{xx} /2 (when OCKSm = 10H 时)	0	1	0	×	32 个时钟
f _{xx} /3 (当 OCKSm = 11H 时)	0	1	0	×	48 个时钟
f _{xx} /4 (当 OCKSm = 12H 时)	0	1	0	×	64 个时钟
f _{xx}	0	1	1	0	16 个时钟
<R> f _{xx} (当 OCKSm = 18H 时)	0	1	1	1	13 个时钟
<R> f _{xx} /2 (当 OCKSm = 10H 时)	0	1	1	1	26 个时钟
<R> f _{xx} /3 (当 OCKSm = 11H 时)	0	1	1	1	39 个时钟
f _{xx} (当 OCKSm = 18H 时)	1	1	0	×	10 个时钟
f _{xx} /2 (当 OCKSm = 10H 时)	1	1	0	×	20 个时钟
f _{xx} /3 (当 OCKSm = 11H 时)	1	1	0	×	30 个时钟
f _{xx} /4 (当 OCKSm = 12H 时)	1	1	0	×	40 个时钟
<R> f _{xx} /5 (当 OCKSm = 13H 时)	1	1	0	×	50 个时钟
f _{xx}	1	1	1	0	10 个时钟

- 备注
1. n = 0 到 2
m = 0, 1
 2. × = 无关

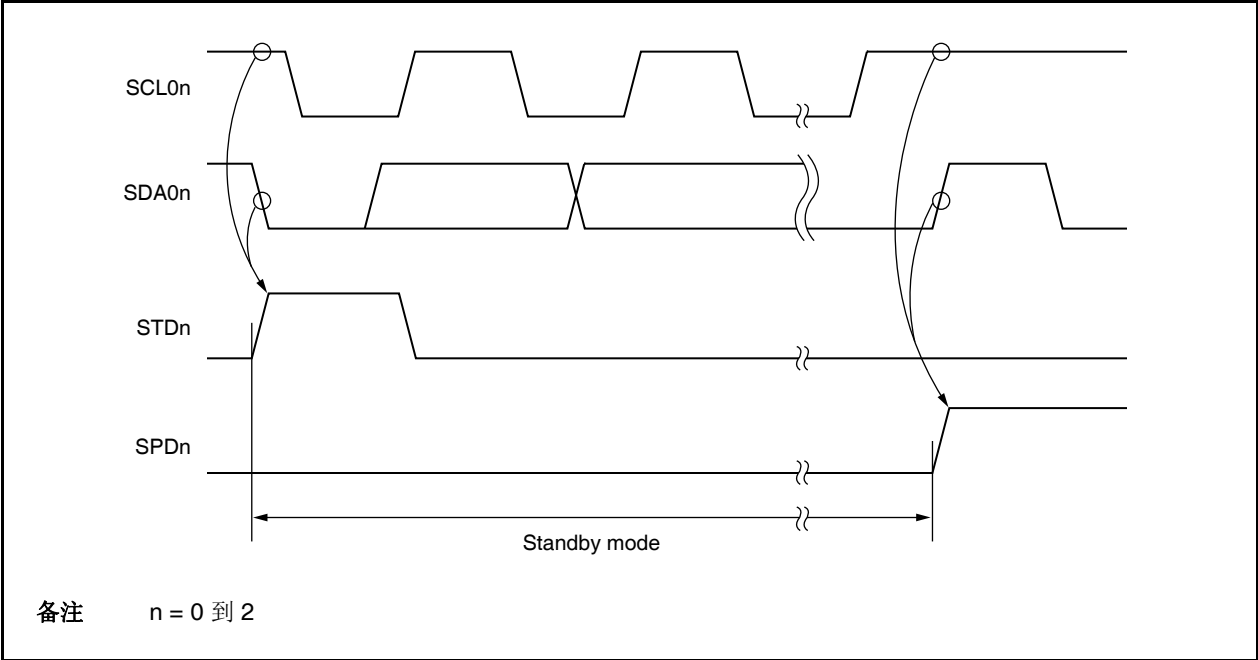
通信保留时序如下所示。

图 17-15. 通信保留时序



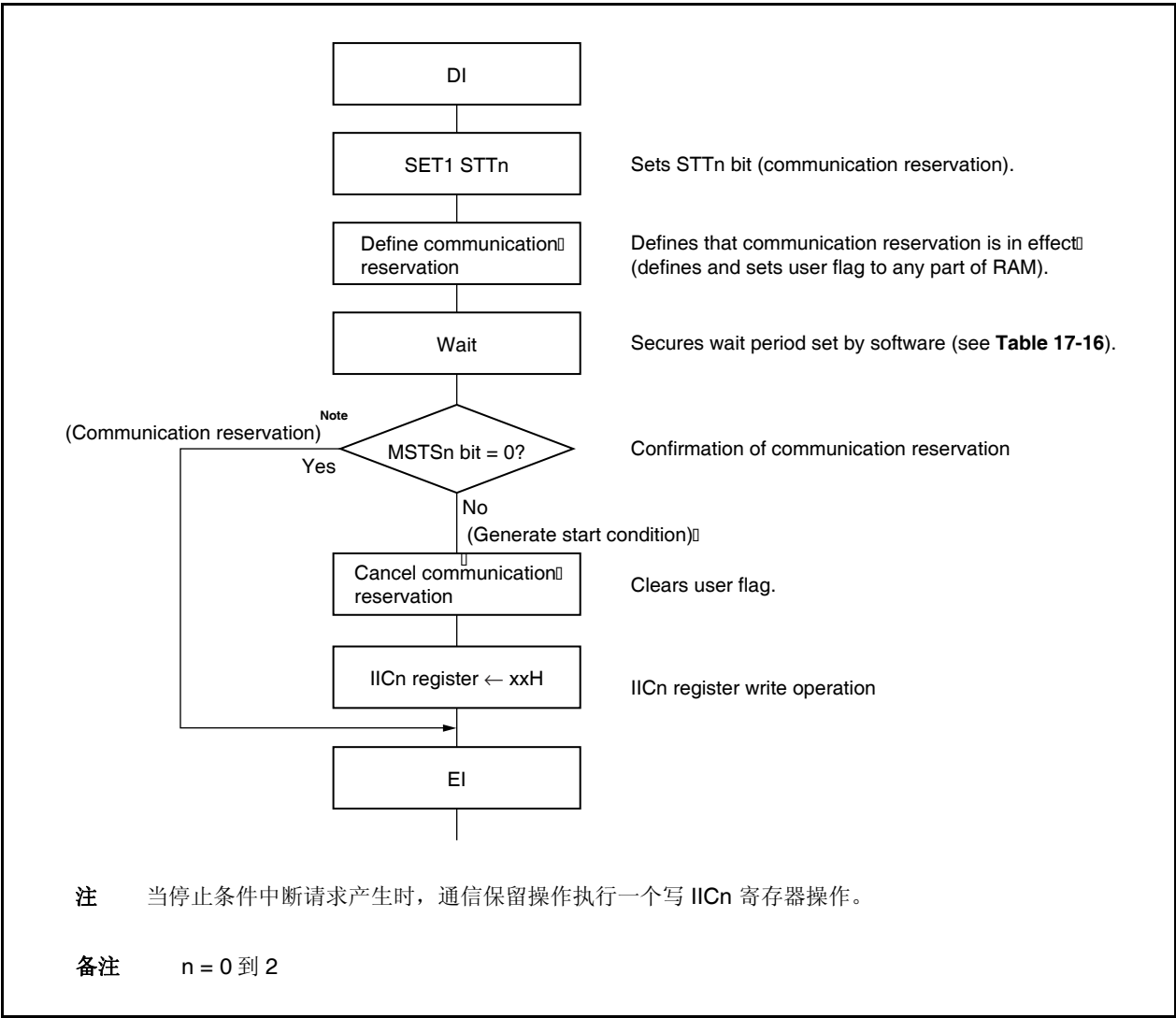
接受通信保留通过下列时序设置。IICSn.STDn 位设置为 1 后，通过在监测到停止条件之前设置 IICn.STTn 位为 1 来进行通信保留设置(n = 0 到 2)。

图 17-16. 接受通信保留时序



通信保留流程图如下阐释。

图 17-17. 通信保留流程图



17.14.2 通信保留功能禁止时 (IICFn.IICRSVn 位 = 1)

总线通信期间，总线不用于通信时，如果 IICFn.STTn 位被设置，则拒绝请求，开始条件也不产生。两种模式下总线不使用。

- 仲裁致使既不进行主操作也不进行从操作。
- 当接收到扩展码，并且从操作禁止时(当 IICFn.LRELn 位设置为 1 时，ACK 不返回，总线释放) (n = 0 到 2)。

为了确认是否产生开始条件，是否拒绝请求，检验 IICFn.STCFn 标志。表 17-7 显示的是 STTn 设置为 1 后直到设置 STCFn 标志之前所需要的时间。因此，通过软件设置时间。

<R>

表 17-7. 等待时间

OCKSEn _m	OCKSm ₁	OCKSm ₀	CLn ₁	CLn ₀	等待时间
1	0	0	0	×	10 个时钟
1	0	1	0	×	15 个时钟
1	1	0	0	×	20 个时钟
1	1	1	0	×	25 个时钟
0	0	0	1	0	5 个时钟

- 备注
- 1. ×: 无关
 - 2. n = 0 到 2

$m = 0, 1$

17.15 注意事项

(1) 当 IICFn.STCENn 位 = 0

I²C0n 操作允许后，不管实际总线状态为何，立即识别总线通信状态(IICFn.IICBSYn 位 = 1)。在还没有监测到停止条件的状态下，执行主设备通信，那么在开始主设备通信之前，停止条件产生，然后总线释放。

依照下列顺序产生停止条件。

- <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

I²C0n 操作允许后，不管实际的总线状态为何，立即识别到总线释放状态(IICBSYn 位 = 0)。为了产生第一个开始条件 (IICCN.STTn 位 = 1)，由必要确认总线已经释放，以免干扰其它通信。

(3) 在与其它设备的通信正在进行过程中，当 V850ES/JG2 的 IICCN.IICEn 位设置为 1 时，依据通信线的状态，开始条件可能监测到。当 SCL0n 和 SDA0n 线为高电平时，一定要设置 IICCN.IICEn 位为 1。

(4) 允许操作(IICCN.IICEn 位 = 1)之前，由 IICCLn， IICXn 和 OCKSm 寄存器决定操作时钟频率。为了改变操作时钟频率，清 IICCN.IICEn 位一次。

(5) IICCN.STTn 和 IICCN.SPTn 位置 1 后，在没有首先清零的情况下，没有必要重置。

(6) 如果通信已经保留,设置 IICCN.SPIEn 位为 1，以便中断请求通过监测停止条件产生。中断请求信号产生后，等待状态通过写通信数据给 I2Cn 释放，然后传送开始。如果中断不通过监测停止条件产生，在等待状态时，传送将停止，因为不产生中断请求。然而，没有必要软件设置 SPIEn 位为 1 监测 IICSn.MSTS n 位。

备注 n = 0 到 2
 m = 0, 1

17.16 通信操作

以下用流程图表示了三种操作方式。

(1) 在单一主系统里的主操作

下面所示是在单一主系统里使用 V850ES/JG2 作为主设备的流程图。

此流程图被整体的分为初始化设置和通信处理两个部分。在启动的时候执行初始化设置。如果需要与从设备进行通信，请做好设置后执行通信处理。

(2) 在多主系统里的主操作

在 I²C0n 总线多主系统里，当 I²C 总线在通信过程中，无法判断总线在占用还是已释放。此时，当数据和时钟保持一定时间的高电平（1 帧），V850ES/JG2 将处于总线释放状态并进入通信。

此流程图被整体的分为初始化设置、通信等待和通信处理三个部分。

这里省略了 V850ES/JG2 仲裁失败并被指定为从设备的处理方法，只显示了作为主设备的处理。在启动的时候执行初始化设置以准备通信。然后，等待通信请求（作为主设备）或者等待指定（作为从设备）。实际的通信工作是在通信处理过程中完成的，它支持与从设备的传送/接收以及与其它主设备的仲裁。

(3) 从操作

如下所示是当 V850ES/JG2 作为 I²C0n 总线的从设备的一个示例。

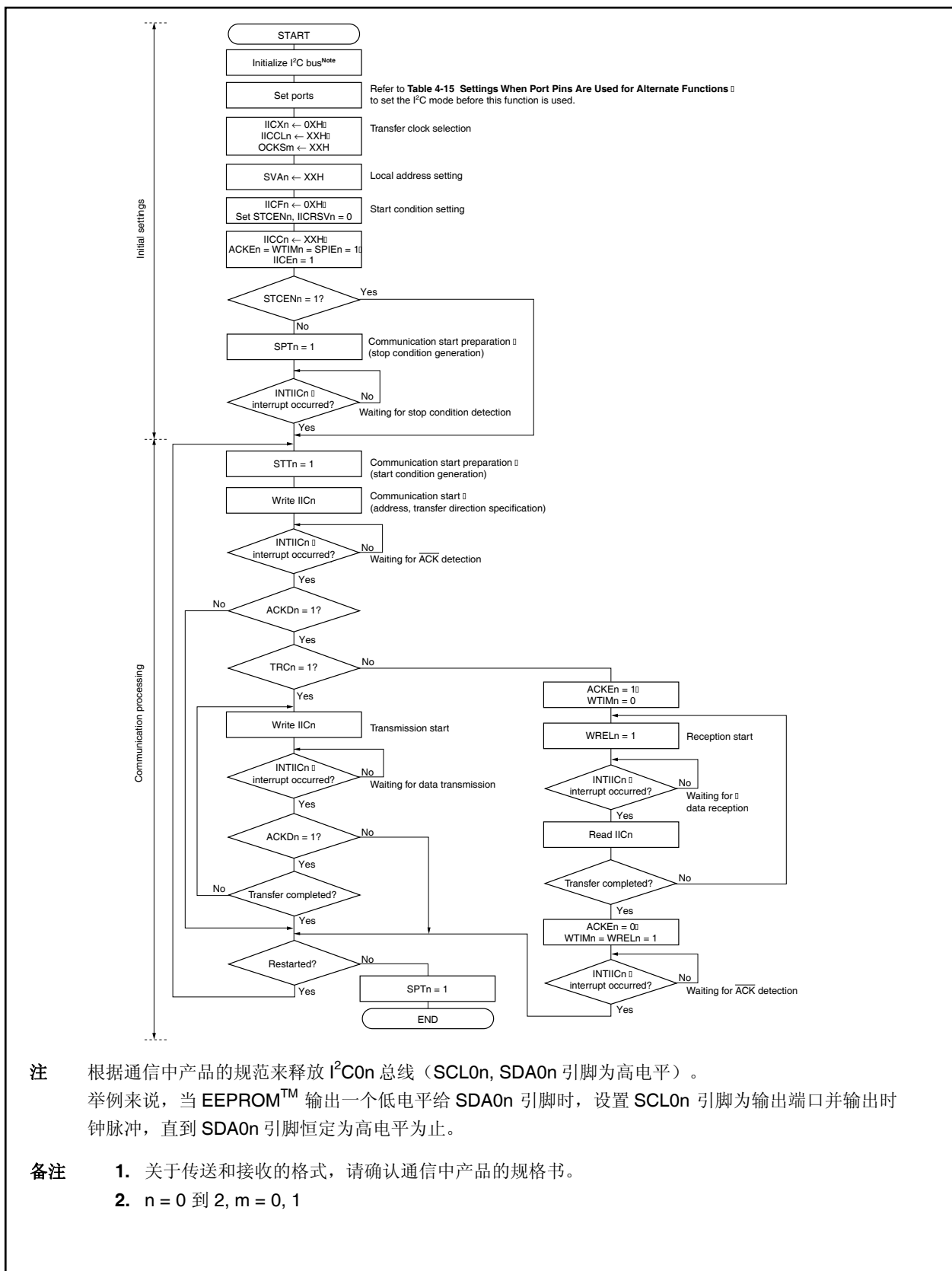
当作为从设备时，操作由中断来启动。在启动的时候执行初始化设置，然后等待 INTIICn 中断的发生（通信等待）。一旦 INTIICn 中断发生，将判断通信状态并将结果作为一个标志传送给主进程。

为了检查这些标志，需要进行必要的通信处理。

备注 n = 0 到 2

17.16.1 在单一主系统里的主操作流程

图 17-18. 在单一主系统里的主操作



17.16.2 在多主系统里的主操作

图 17-19. 在多主系统里的主操作 (1/3)

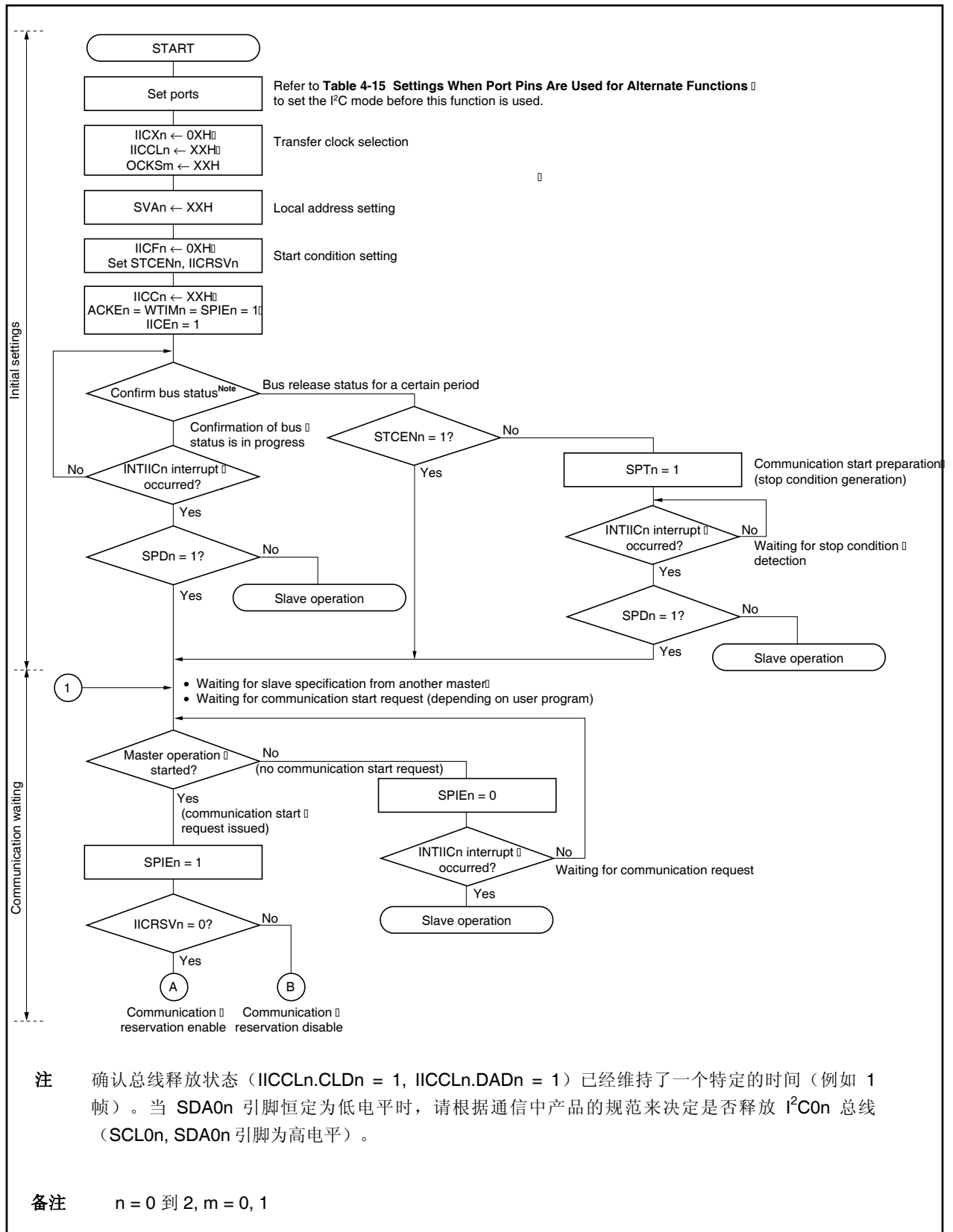


图 17-19. 在多主系统里的主操作 (2/3)

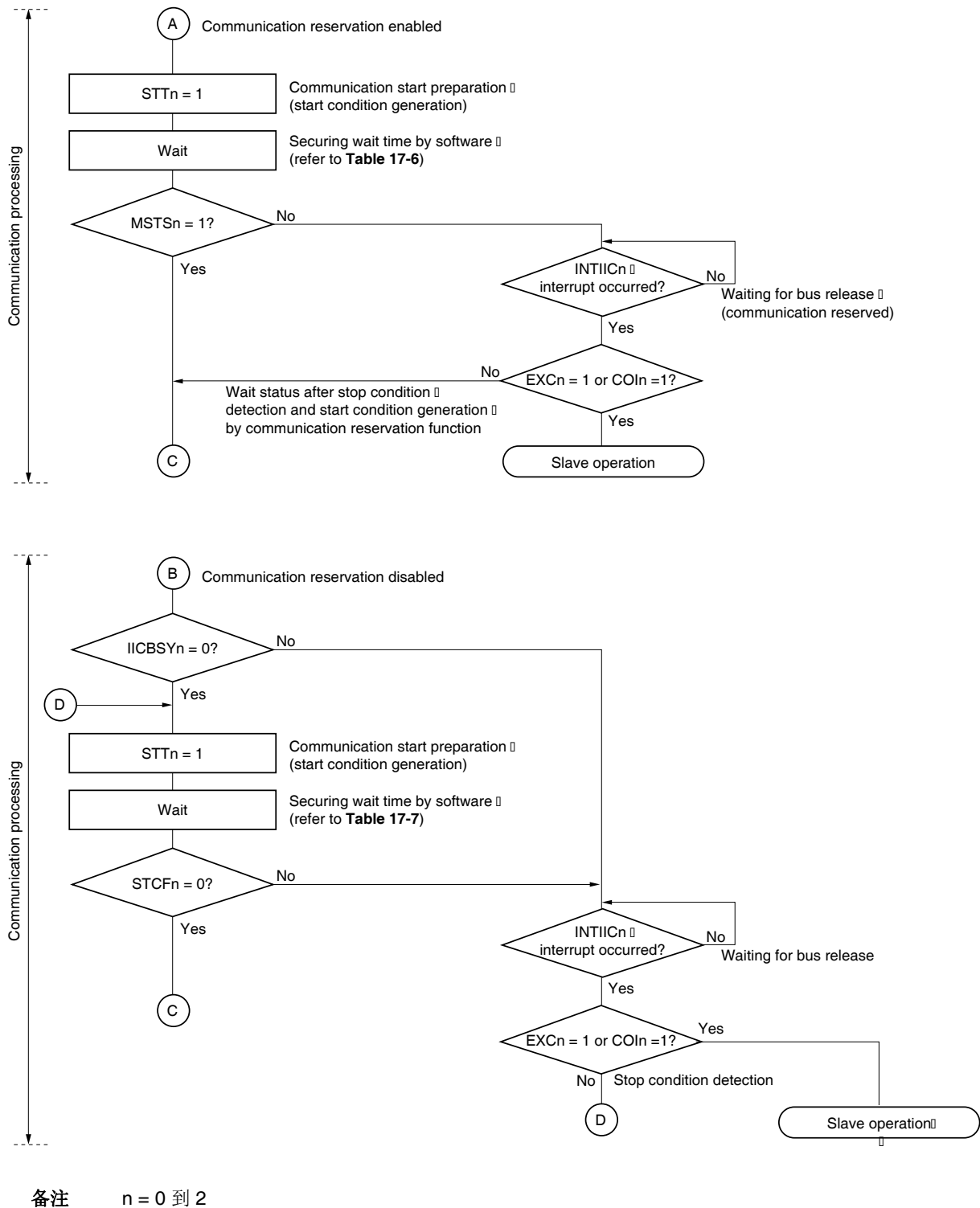
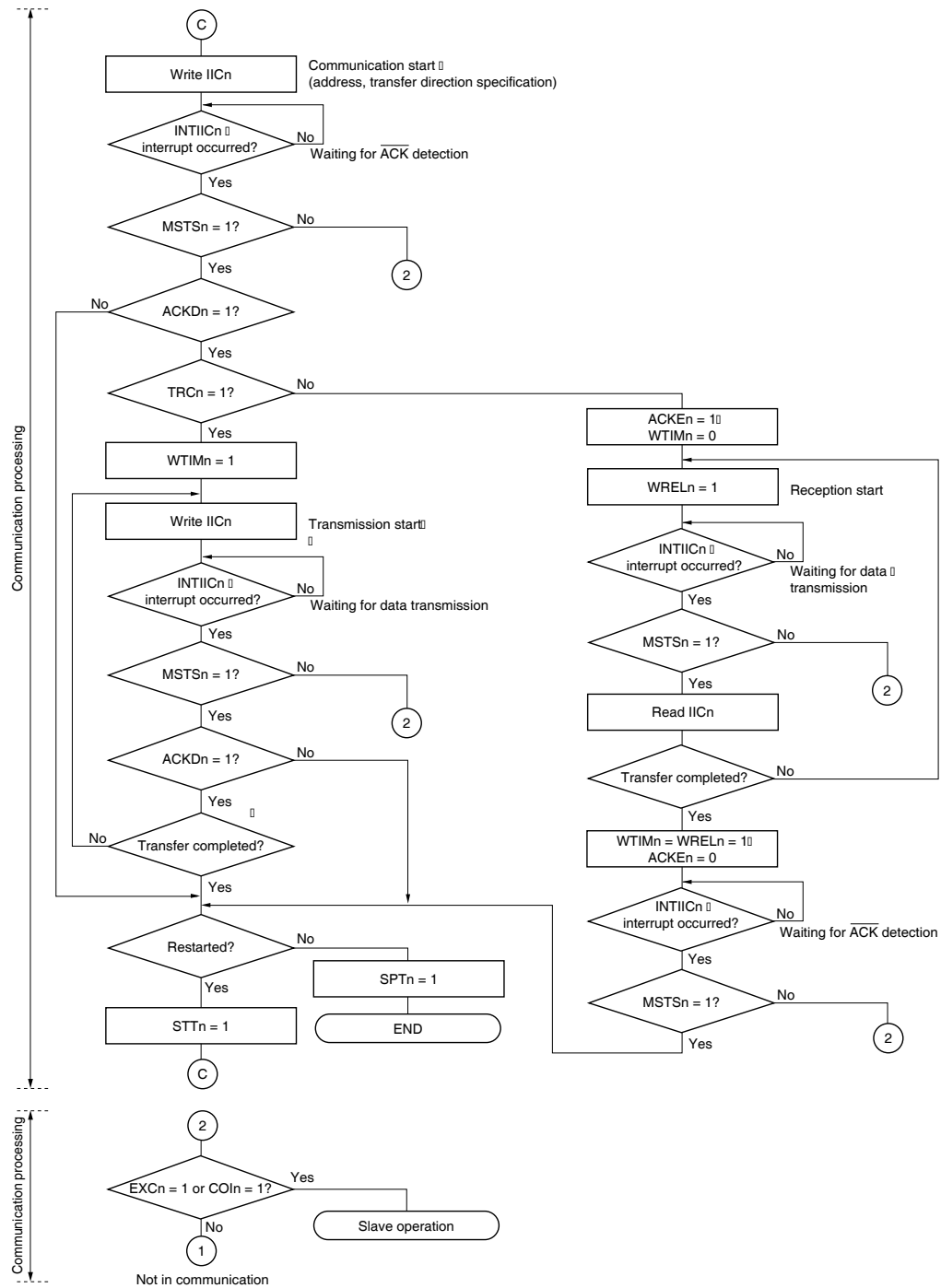


图 17-19. 在多主系统里的主操作 (3/3)



备注

1. 关于传送和接收的格式，请确认通信中产品的规格书。
2. 当使用 V850ES/JG2 作为多主系统里的主设备时，请在每个 INTIIC_n 中断发生时读取 IICSn.MSTS_n 位，以确认仲裁结果。
3. 当使用 V850ES/JG2 作为多主系统里的从设备时，请在每个 INTIIC_n 中断发生时用 IICSn 和 IICFn 寄存器确认状态，以决定下一步处理。
4. n = 0 到 2

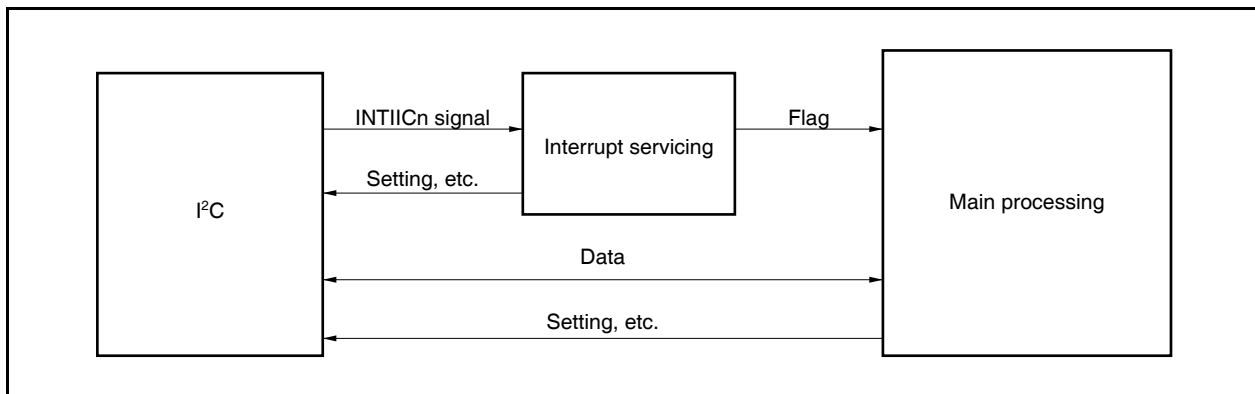
17.16.3 从设备操作

下面所示为从设备操作的处理流程。

从设备操作基本上是事件驱动。因此，有必要由 INTIICn 中断 (处理需要一定的操作状态的改变，例如通信期间停止状态监测) 处理。

下面的描述假定数据通信不支持扩展码。同时，假定 INTIICn 中断服务程序只完成状态改变处理和主程序处理期间只有实际的数据传送。

图 17-20. 从设备操作期间软件架构



因此，准备下列三个标志，以便使数据传送处理通过传送标志给主程序而不是 INTIICn 信号完成。

(1) 通信模式标志

这个标志指示下列通信状态。

清除模式： 不进行数据处理

通信模式： 进行数据通信处理(有效的地址监测，停止条件监测，从主设备来的 $\overline{\text{ACK}}$ 不监测和地址不匹配)

(2) 准备标志

这个标志指示数据通信允许。正常数据传送期间和 INTIICn 中断相同。这个标志在中断服务处理模块设置，在主程序模块处理期间清除。第一个传送数据的准备标志不在中断服务处理模块设置，因此第一个数据在没有清除处理的情况下进行串送(地址匹配认为是下一个数据的请求)。

(3) 通信方向标志

这个标志指示通信的方向，和 IICSn.TRCn 位相同。

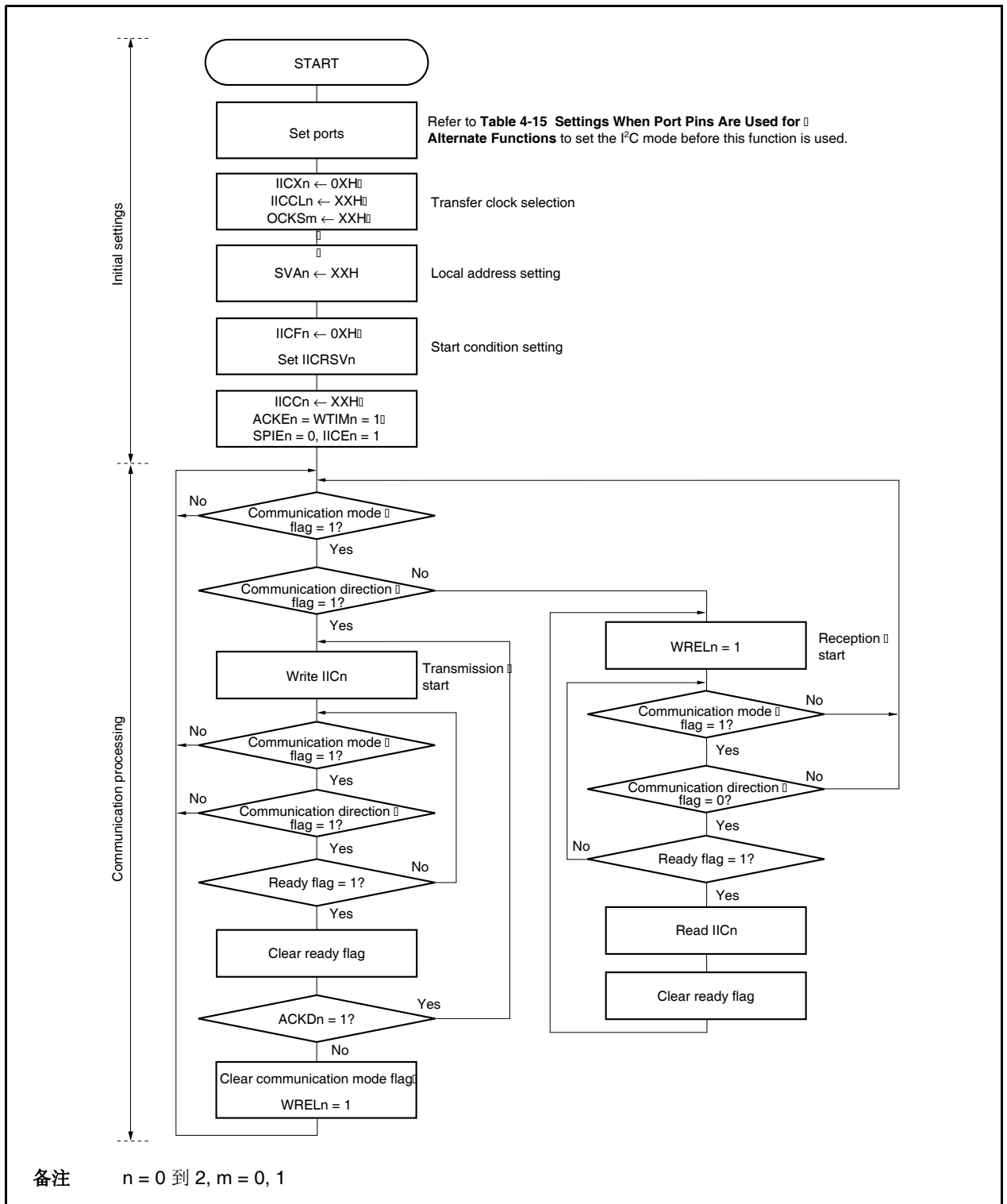
下面显示的是从设备操作期间主程序处理模块的操作。

I²COn 开始，并且等待通信允许状态。当通信允许时，使用通信模式标志和准备标志进行传送(停止条件和开始条件的处理由中断完成，条件的确认由标志来完成)。

对于传送来说，重复传送操作直到主设备停止返回 $\overline{\text{ACK}}$ 。当主设备停止返回 $\overline{\text{ACK}}$ 时，传送完成。

对于接收来说，传送完成后，接收到要求数目的数据，但是并不立即返回 $\overline{\text{ACK}}$ ，为下一个数据准备。这之后，主设备产生停止条件或者重启条件。这将引起退出通信。

图 17-21. 从设备操作流程 (1)

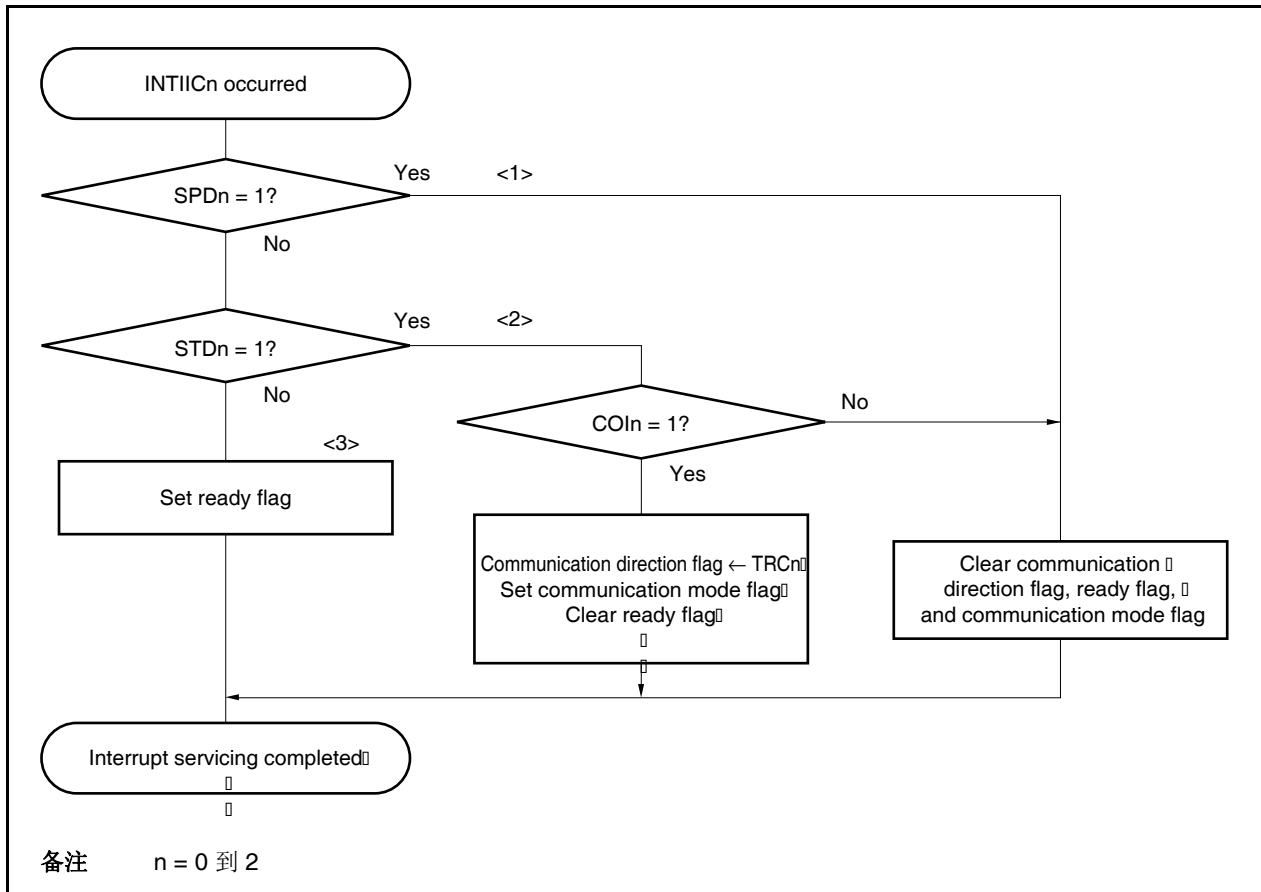


下面显示的实例是由 INTIICn 中断完成的从设备的处理(这里假定不使用扩展码)。INTIICn 中断期间, 状态得到确认, 并且执行接下来的步骤。

- <1> 监测到停止状态时, 通信终止。
- <2> 监测到开始条件时, 确认地址。如果地址不匹配, 通信终止。如果地址匹配, 设定通信模式, 等待释放, 操作从中断返回(清除准备标志)。
- <3> 对于数据传送/接收来说, 准备标志设置时, 在 I²C0n 总线处于等待状态期间, 操作从中断返回。

备注 上面的<1> ~ <3> 对应于图 17-22 从设备操作流程(2) 中的<1> ~ <3> 。

图 17-22. 从设备操作流程 (2)



17.17 数据通信时序

使用 I²C 总线模式时，主设备通过串行总线输出一个地址来从几个从设备里选择一个座为它的通信方。

输出从地址后，主设备传送 IICSn.TRCn 为，它规定了数据传送方向，并且接着与从设备开始串行通信。

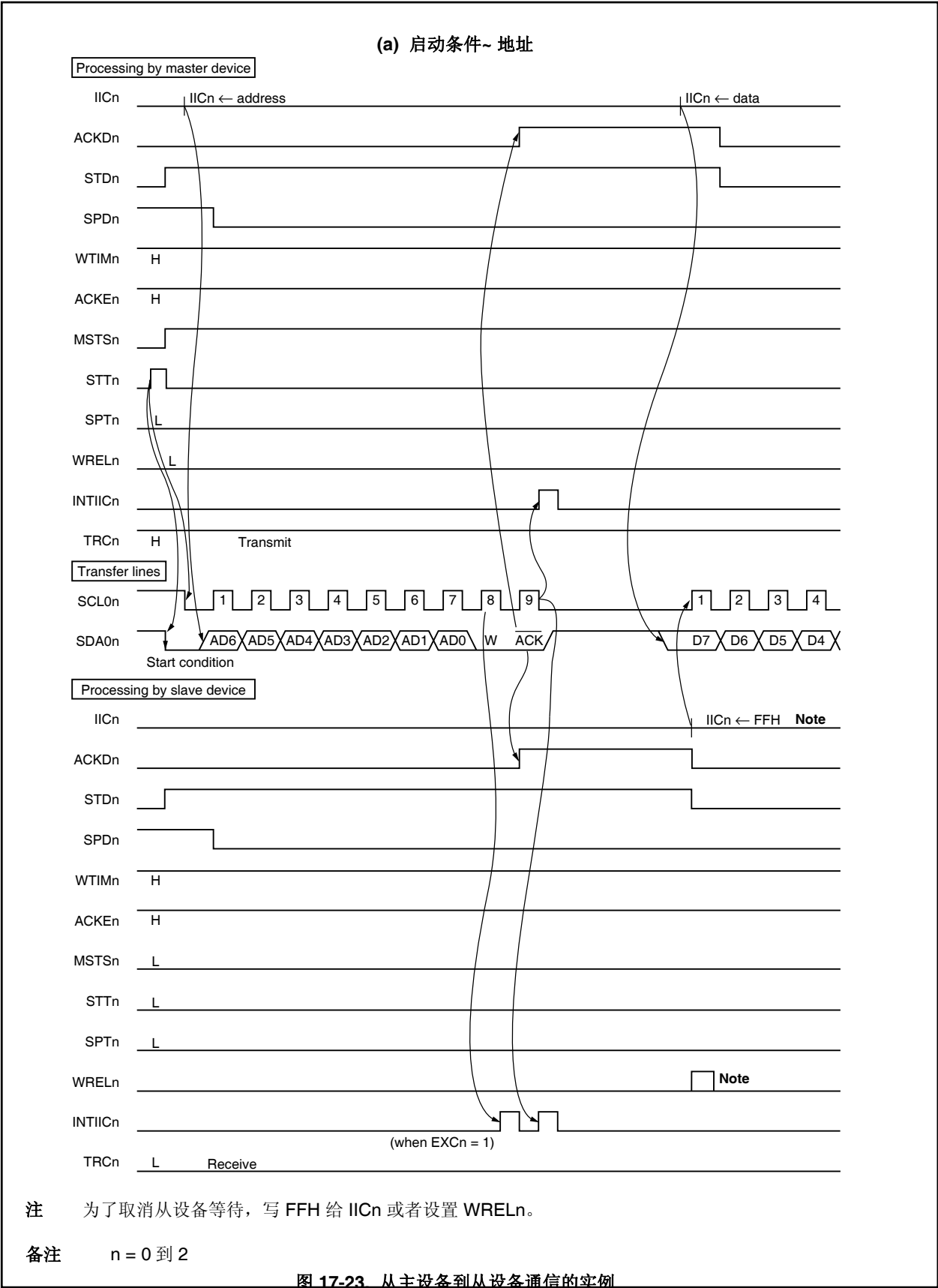
IICn 寄存器的移位操作和串行时钟引脚(SCL0n)的下降沿同步。传送数据传送到 SO 锁存器，并且通过 SDA0n 引脚(MSB 先)输出。

通过 SDA0n 引脚的数据在 SCL0n 引脚的上升沿由 IICn 寄存器捕捉。

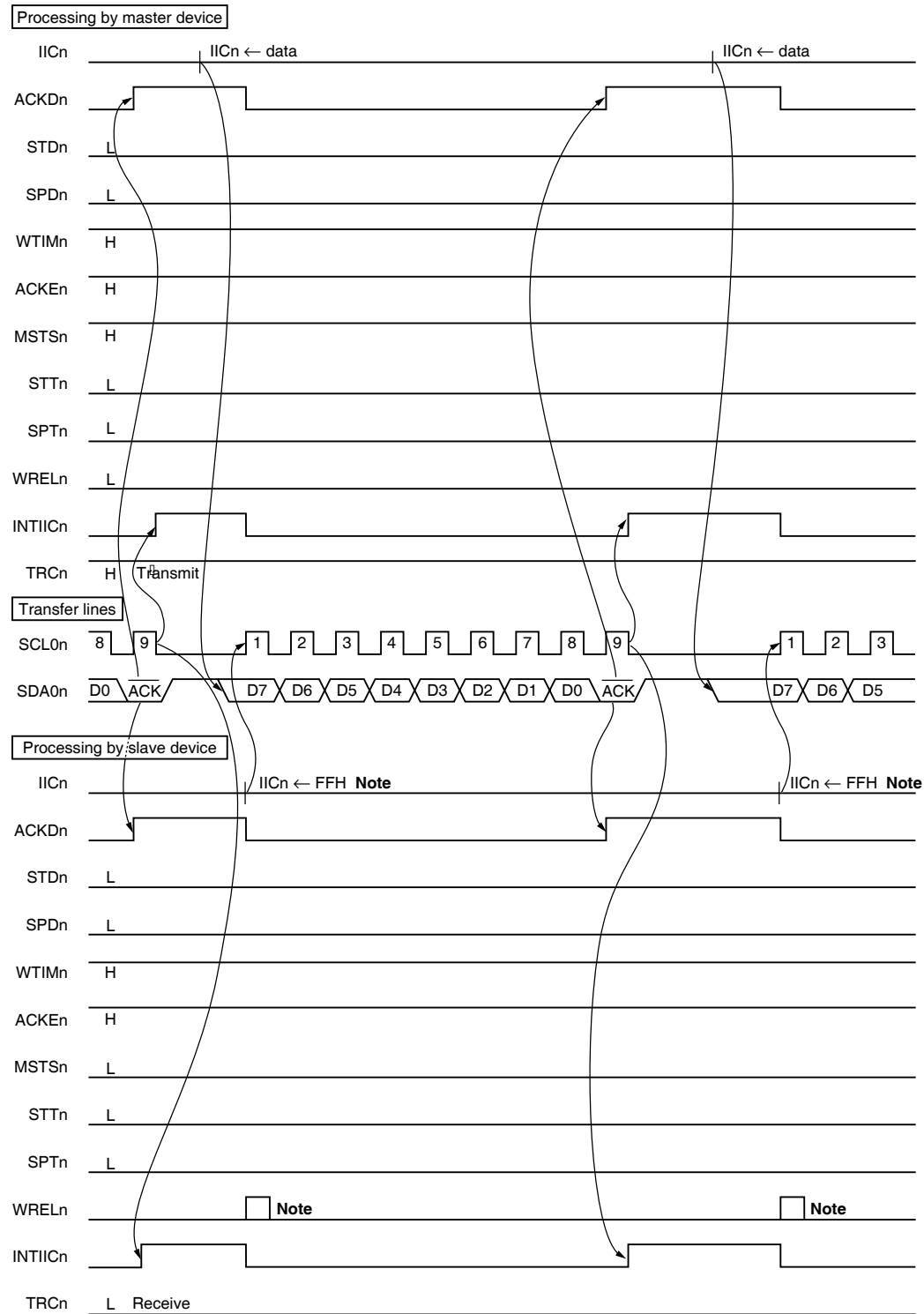
数据通信时序如下所示。

备注 n = 0 到 2

图 17-23. 从主设备到从设备通信的实例
(当主设备和从设备都选择 9 个时钟等待时) (1/3)



(b) 数据



注 为了取消从设备等待，写 FFH 给 IICn 或者设置 WRELn。

备注 n = 0 到 2

图 17-23. 从主设备到从设备通信的实例
(当主设备和从设备都选择 9 个时钟等待时) (3/3)

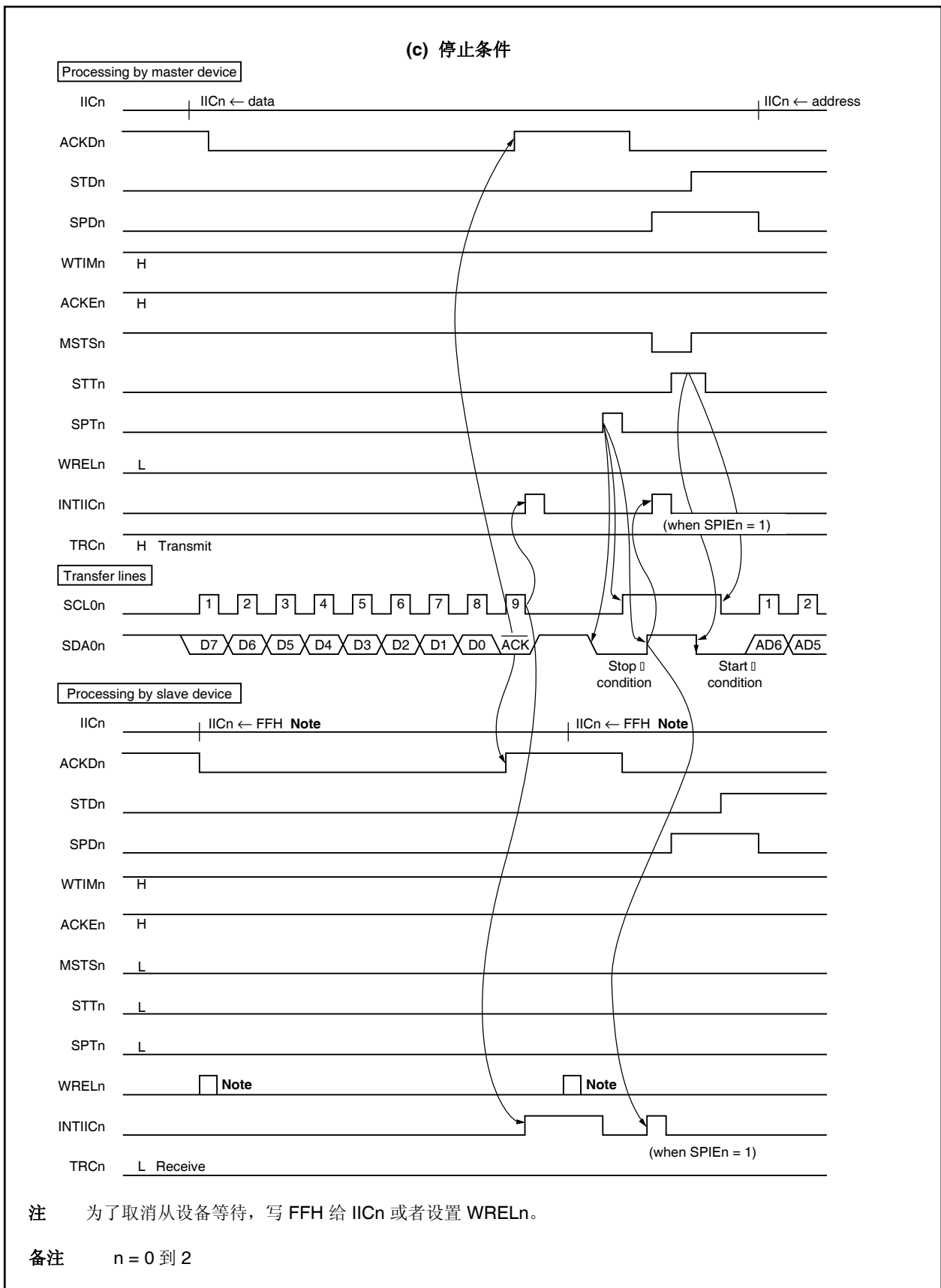


图 17-24. 从从设备到主设备通信的实例
(当主设备和从设备都选择 9 个时钟等待时) (1/3)

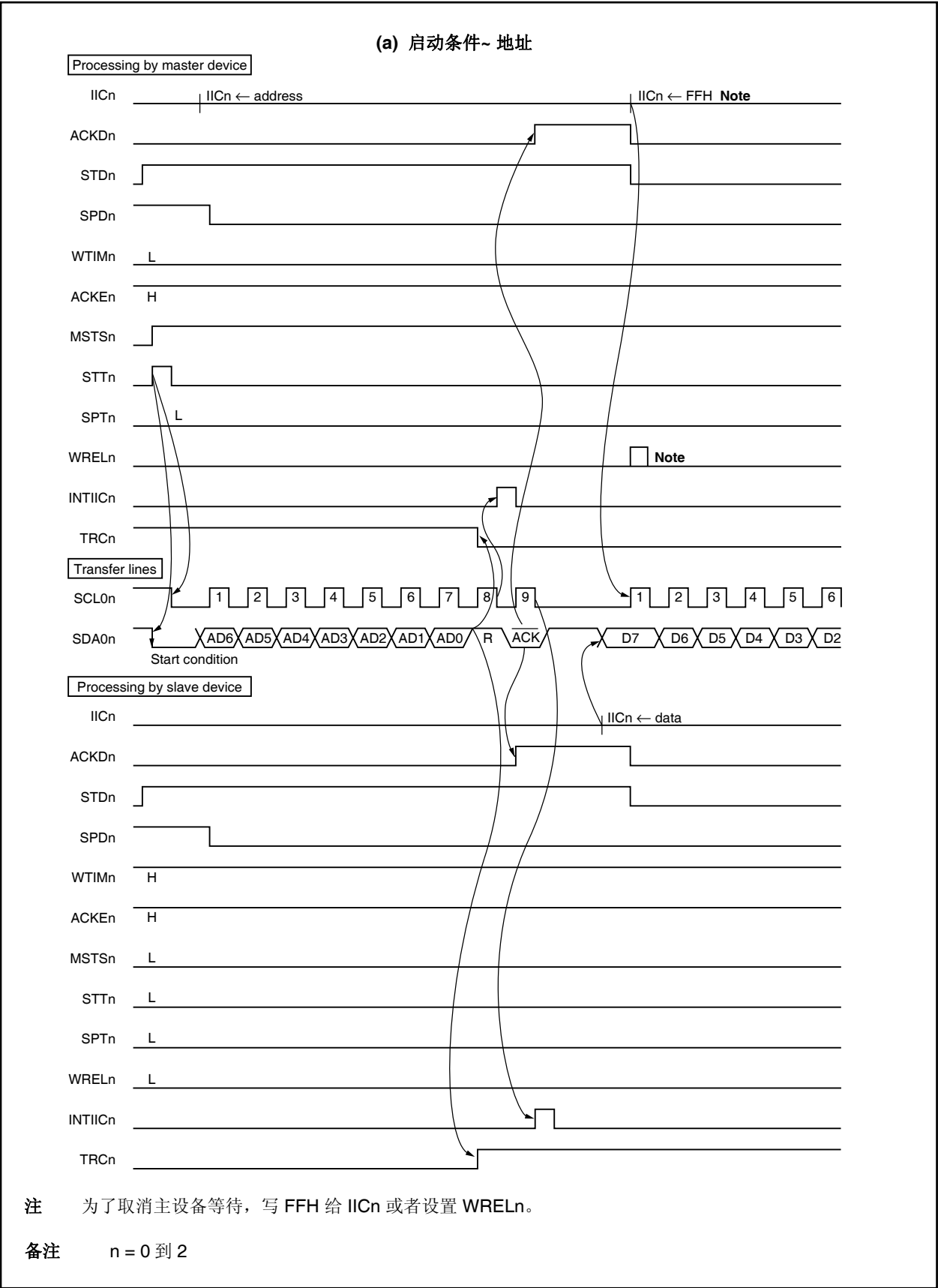
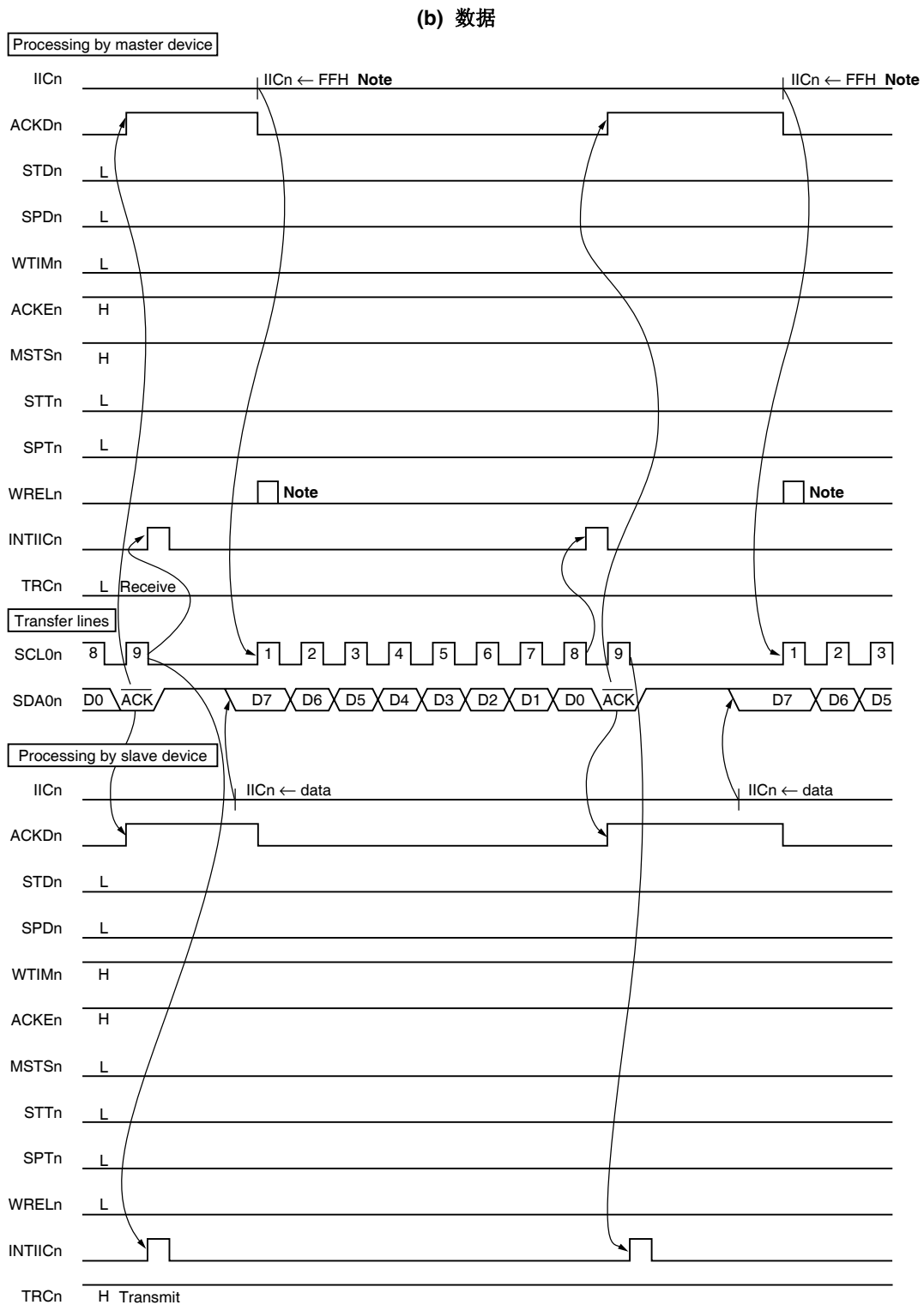


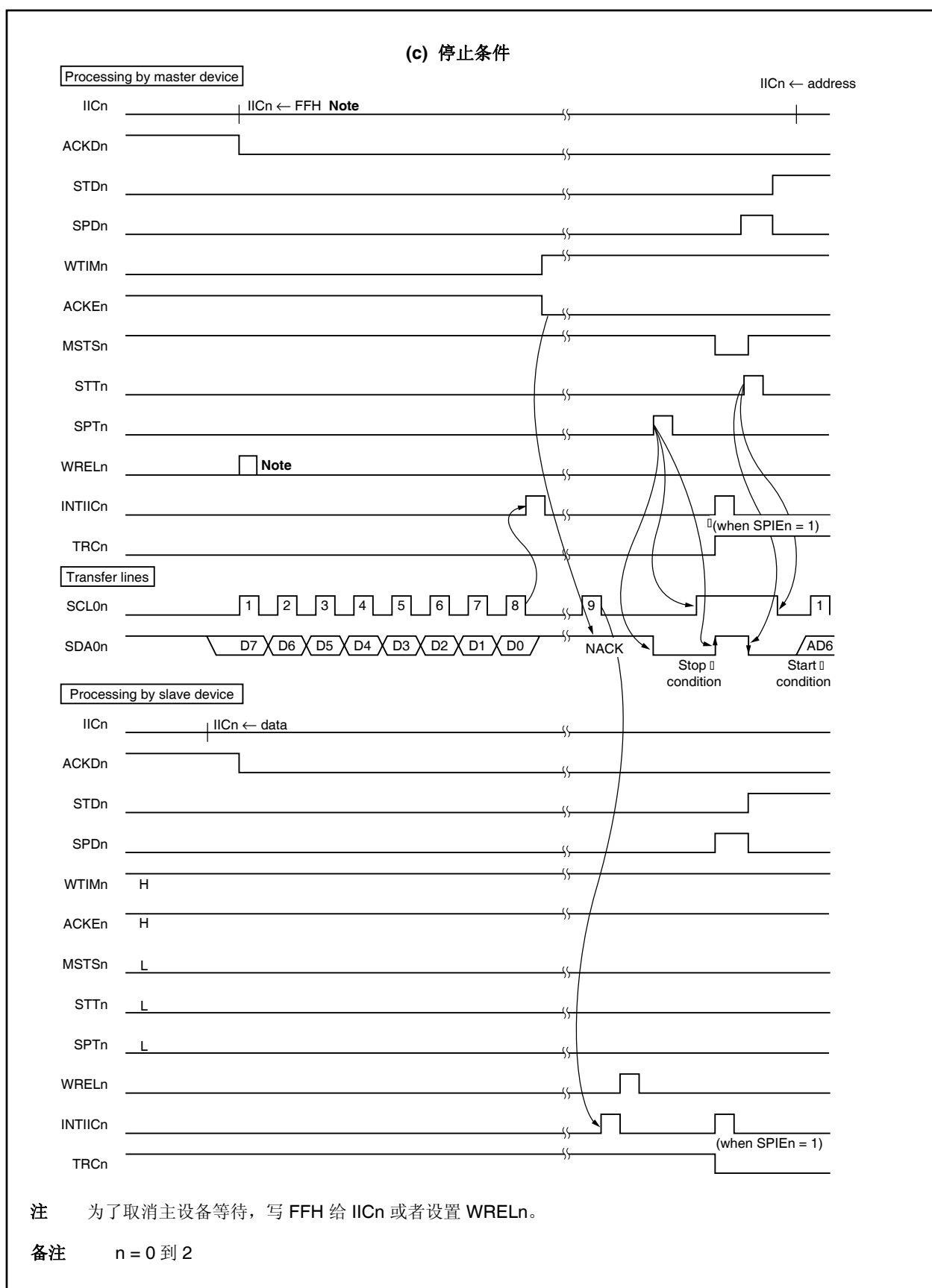
图 17-24. 从从设备到主设备通信的实例
(当主设备和从设备都选择 9 个时钟等待时) (2/3)



注 为了取消主设备等待，写 FFH 给 IICn 或者设置 WRELn。

备注 n = 0 到 2

图 17-24. 从从设备到主设备通信的实例
(当主设备和从设备都选择 9 个时钟等待时) (3/3)



第十八章 DMA 功能 (DMA 控制器)

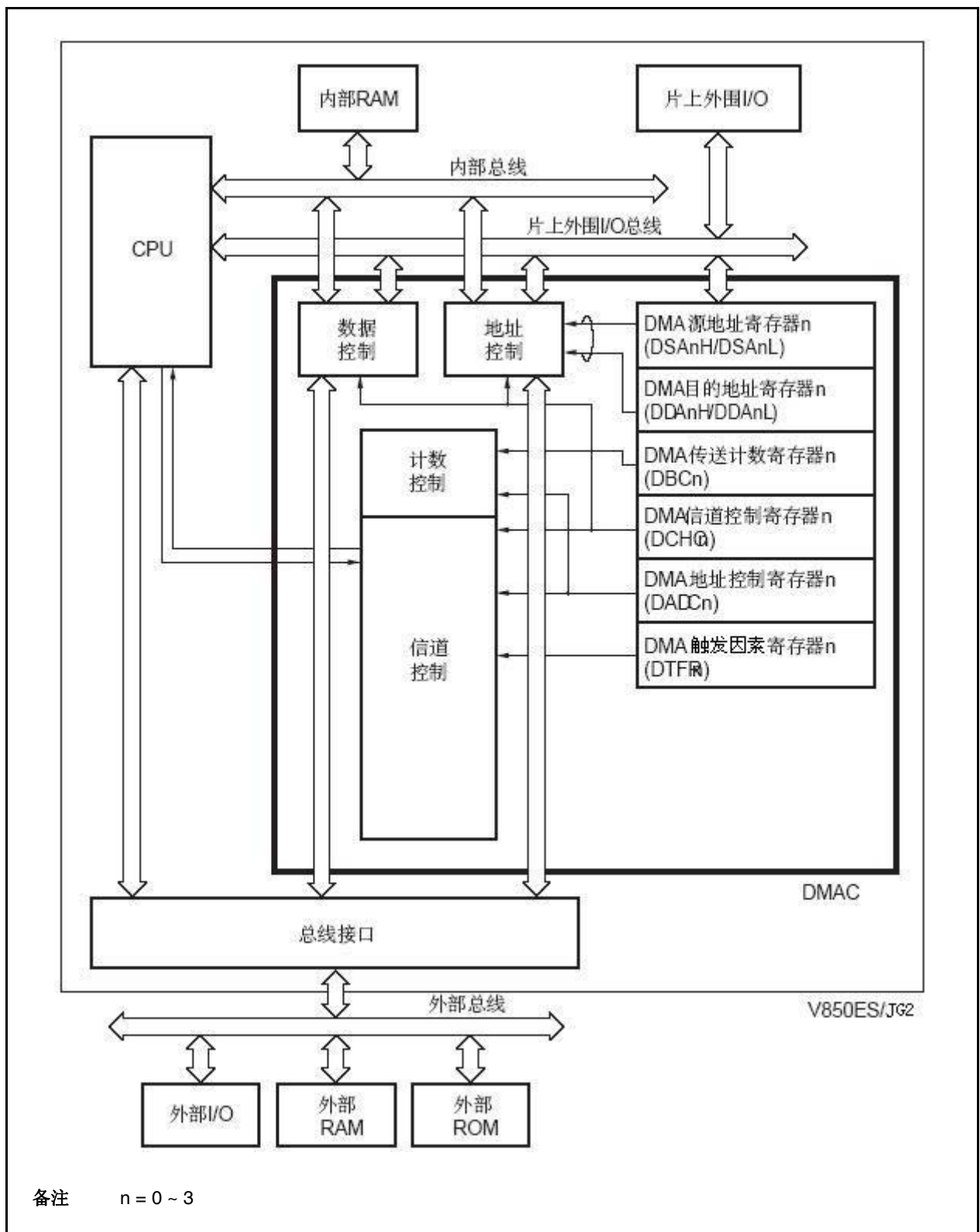
V850ES/JG2 包含了直接存储器访问(DMA)控制器(DMAC)，它用来执行和控制 DMA 传送。

DMAC 控制基于 DMA 请求的存储器和 I/O、存储器和存储器或是 I/O 和 I/O 之间的数据传送，这种请求通常是由片上外围 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，或是软件触发(内部 RAM 或是外部存储器)发出的。

18.1 特性

- 4 路独立的 DMA 信道
- 传送单元：8/16 位
- 最大传送计数值：65,536 (2^{16})
- 传送类型：双周期传送
- 传送模式：单一的传送模式
- 传送请求：
 - 通过片上外围 I/O(串行接口，定时器/计数器和 A/D 转换器)中断或是外部输入引脚的中断而产生的请求。
 - 通过软件触发产生的请求
- 传送目标：
 - 内部 RAM ↔ 外围 I/O
 - 外围 I/O ↔ 外围 I/O
 - 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

18.2 配置



18.3 寄存器

(1) DMA 源地址寄存器 0 到 3 (DSA0 到 DSA3)

DSA0 到 DSA3 寄存器为 DMA 信道 n(n 的取值范围是 0 到 3)设定 DMA 的源地址(每个寄存器 26 位)。

这些寄存器能够被分为 2 个 16 位寄存器，DSAnH 和 DSAnL。

这些寄存器能够以 16 位为单元被读取或是写入。

复位后: 未定义 读/写 地址

DSA0H FFFFF082H, DSA1H FFFFF08AH,

DSA2H FFFFF092H, DSA3H FFFFF09AH,

DSA0L FFFFF080H, DSA1L FFFFF088H,

DSA2L FFFFF090H, DSA3L FFFFF098H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSAnH (n = 0到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
DSAnL (n = 0到3)	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0

IR	指定DMA传送源
0	外部存储器或是片上外围I/O
1	内部RAM

SA25到SA16	设定DMA传送源的地址(A25到A16) (未定义缺省值)。 在DMA传送期间，保存下一个DMA传送的源地址。 当DMA传送完成，保持第一个设定的DMA地址。
-----------	---

SA15到SA0	设定DMA传送源的地址(A15到A0) (未定义缺省值)。 在DMA传送期间，保存下一个DMA传送的源地址。 当DMA传送完成，保持第一个设定的DMA地址。
----------	--

注意事项 1. 一定要将 DSAnH 寄存器的位 14 到 10 清 0。

2. 当 DMA 传送被禁止的时候(DCHCn.Enn 位为 0)，在下列时序下设定 DSAnH 和 DSAnL 寄存器。

- 在复位后开始发送第一个 DMA 传送之前的时期
- 在通过 DCHCn.INITn 位将信道初始化后来开始 DMA 传送的时期
- 在完成一个 DMA 传送(DCHCn.TCn 位为 1)和开始下一个 DMA 传送之间的时期

3. 当 DSAn 寄存器中的数值被读取，两个 16 位寄存器 DSAnH 和 DSAnL 的值被读取。如果读取操作与数据更新发生冲突，将读取正在被更新的数据(参见 18.13 注意事项)。

4. 复位后，在开始 DMA 传送之前要首先设定 DSAnH, DSAnL, DDAnH, DDAnL 和 DBCn 寄存器。否则，当 DMA 发送开始后的操作无法得到保证。

(2) DMA 目的地址寄存器 0 到 3 (DDA0 到 DDA3)

DDA0 到 DDA3 寄存器为 DMA 信道 n(n 的取值范围是 0 到 3)设定 DMA 的目的地址(每个寄存器 26 位)。
这些寄存器可以分为两个 16 位寄存器, DDAnH 和 DDAnL。
这些寄存器可以以 16 位作为单元被读取或是写入。

复位后的值: 未定义 读/写 地址: DDA0H FFFFF086H, DDA0L FFFFF08EH,
DA2H FFFFF096H, DDA2H FFFFF09EH,
DDA0L FFFFF084H, DDA0L FFFFF08CH,
DDA2L FFFFF094H, DDA2L FFFFF09CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDAnH (n = 0 到 3)	IR	0	0	0	0	0	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDAnL (n = 0 到 3)	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

IR	指定DMA传送的目的
0	外部存储器或片上外围I/O
1	内部RAM

DA25到DA16	设定DMA传送目的的地址(A25到A16) (未定义缺省值)。 在DMA传送期间, 保存下一个DMA传送的目的地址。 当DMA传送完成, 保持第一个设定的DMA源地址。
-----------	--

DA15 to DA0	设定DMA传送目的的地址(A15到A0) (未定义缺省值)。 在DMA传送期间, 保存下一个DMA传送的目的地址。 当DMA传送完成, 保持第一个设定的DMA源地址。
-------------	---

- 注意事项**
- 一定要将 DDAnH 寄存器的位 14 到 10 清 0。
 - 当 DMA 传送被禁止的时候(DCHCn.Enn 位为 0), 在下列时序下设定 DDAnH 和 DDAnL 寄存器。
 - 在复位后开始发送第一个 DMA 传送之前的时期
 - 在通过 DCHCn.INITn 位将信道初始化后来开始 DMA 传送的时期
 - 在完成一个 DMA 传送(DCHCn.TCn 位为 1)和开始下一个 DMA 传送之间的时期
 - 当 DDAn 寄存器中的数值被读取, 两个 16 位寄存器 DDAnH 和 DDAnL 的值被读取。如果读取操作与数据更新发生冲突, 将读取正在被更新的数据(参见 18.13 注意事项)。
 - 复位后, 在开始 DMA 传送之前要首先设定 DSAnH, DSAnL, DDAnH, DDAnL 和 DBCn 寄存器。否则, 当 DMA 发送开始后的操作无法得到保证。

(3) DMA 字节计数寄存器 0 到 3 (DBC0 到 DBC3)

DBC0 到 DBC3 寄存器是 16 位寄存器，它用来为 DMA 信道 n (n 的取值范围是 0 到 3) 设定字节传送的值。在 DMA 发送过程中，这些寄存器保存着剩余的发送计数。这些寄存器在每次传送之后会自动减 1，不考虑传送的数据单元(8/16 位)，并且在发生借位时中止发送。这些寄存器可以以 16 位作为单元被读取或是写入。

复位后的值: 未定义 读/写 地址: DBC0 FFFFF0C0H, DBC1 FFFFF0C2H,
DBC2 FFFFF0C4H, DBC3 FFFFF0C6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBCn (n=0到3)	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0

BC15到 BC0	在DMA传送期间字节传送计数器设定 或是剩余的字节传送数
0000H	字节传送计数为1或是剩余的字节传送数为1
0001H	字节传送计数为2或是剩余的字节传送数为2
:	:
FFFFH	字节传送计数为65,536 (2^{16})或是剩余的字节传送数为65,536 (2^{16})
当DMA传送完成后保持第一次设定的传送数据的数量。	

注意事项 1. 当 DMA 传送被禁止的时候(DCHCn.Enn 位为 0)，在下列时序下设定 DBCn 寄存器。

- 在复位后开始发送第一个 DMA 传送之前的时期
- 在通过 DCHCn.INITn 位将信道初始化后来开始 DMA 传送的时期
- 在完成一个 DMA 传送(DCHCn.TCn 位为 1)和开始下一个 DMA 传送之间的时期

2. 复位后，在开始 DMA 传送之前要首先设定 DSA nH, DSA nL, DDA nH, DDA nL 和 DBCn 寄存器。否则，当 DMA 发送开始后的操作无法得到保证。

(4) DMA 寻址控制寄存器 0 到 3 (DADC0 到 DADC3)

DADC0 到 DADC3 寄存器是 16 位寄存器，它控制着 DMA 信道 n (n 的取值范围是 0 到 3) 的 DMA 传送模式。

这些寄存器可以以 16 位作为单元被读取或是写入。

复位后，这些寄存器的值为 0000H。

复位后的值: 0000H 读/写 地址: DADC0 FFFFF0D0H, ADC1 FFFFF0D2H,
DADC2 FFFFF0D4H, ADC3 FFFFF0D6H

DADCn (n=0到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	设定传送数据的单元
0	8 位
1	16 位

SAD1	SAD0	设定传送源地址的计数方式
0	0	递增
0	1	递减
1	0	固定
1	1	禁止设定

DAD1	DAD0	设定传送目的地址的计数方式
0	0	递增
0	1	递减
1	0	固定
1	1	禁止设定

- 注意事项**
- 一定要将 DADCn 寄存器的位 15, 13 到 8, 和 3 到 0 清 0。
 - 当 DMA 传送被禁止的时候(DCHCn.Enn 位为 0)，在下列时序下设定 DADCn 寄存器。
 - 在复位后开始发送第一个 DMA 传送之前的时期
 - 在通过 DCHCn.INITn 位将信道初始化后来开始 DMA 传送的时期
 - 在完成一个 DMA 传送(DCHCn.TCn 位为 1)和开始下一个 DMA 传送之间的时期
 - DS0 位指定了传送数据的大小，但并不能控制总线容量。因此如果设定为 8 位数据(DS0 位为 0)，较低位的数据总线并不经常使用。
 - 如果传送数据设定为 16 位 (DS0 位为 1)，传送不能从奇数地址开始。传送总是从低地址的首位为 0 的地址开始。
 - 如果由片上外围 I/O 寄存器 (无论作为发送的源还是目的) 执行 DMA 传送，一定要按照寄存器的大小来设定传送数据的大小。例如，在一个 8 位寄存器上执行 DMA 传送，一定要指定 8 位传送。

(5) DMA 信道控制寄存器 0 到 3 (DCHC0 到 DCHC3)

DCHC0 到 DCHC3 寄存器是 8 位寄存器，它控制着 DMA 信道 n 的 DMA 发送操作模式。

这些寄存器可以以 8 位或是 1 位为单元读取或是写入。(但是，位 7 是只读位、位 1 和位 2 是只写位。如果读取位 1 和位 2，结果永远为 0。)

复位后，这些寄存器的值为 00H。

复位后的值: 00H 读/写 地址: DCHC0 FFFFF0E0H, DCHC1 FFFFF0E2H,
DCHC2 FFFFF0E4H, DCHC3 FFFFF0E6H

DCHCn	<7>	6	5	4	3	<2>	<1>	<0>
(n = 0到3)	TCn ^{#1}	0	0	0	0	INITn ^{#2}	STGn ^{#2}	Enn

TCn ^{#1}	指示通过DMA信道的DMA传送是否完成的状态标志
0	DMA 传送未完成。
1	DMA 传送完成。
该位在最后一个 DMA 传送时被置1，在读取该位时被清0。	

INITn ^{#2}	如果在禁止DMA传送时将 INITn位置1(Enn位= 0), DMA传送状态能够被初始化。 当在DMA 传送完成之前(TCn位被置1之前)重新设定 DMA 传送状态时(重新设定DDAnH, DDAnL, DSAAnH, DSAAnL, DBCn和DACn寄存器), 一定要初始化 DMA 信道。 但是当初始化 DMA 控制器时, 一定要按照18. 13注意事项中的描述。
---------------------	--

STGn ^{#2}	这是一个软件启动触发的DMA传送。 如果在 DMA 传送允许状态中(TCn位 = 0, Enn位 = 1)该位被置1, 开始DMA传送。
--------------------	---

Enn	设定通过DMA信道的DMA 传送是否被允许
0	禁止DMA 传送
1	允许DMA 传送
当Enn位置1时, 允许DMA 传送。 当DMA 传送完成后(产生最终计数值), 该位自动清0。 为了中止DMA 传送, 通过软件将Enn位清0。如果要恢复, 重新将Enn位置1。 但是当中止或者恢复DMA 传送时, 一定要按照18. 13注意事项中的描述。	

- 注
1. TCn 位为只读位。
 2. INITn 和 STGn 位为只写位。

- 注意事项
1. 一定要将 DCHCn 寄存器的位 6 到 3 清 0。
 2. 当 DMA 传送完成时 (中止记数产生)，Enn 位清 0 然后 TCn 位置 1。如果在其数据位被更新时读取 DCHCn 寄存器，指示“传送未完成和传送被禁止(TCn 位为 0 和 Enn 位为 0)”的数值可以被读取。

(6) DMA 触发因素寄存器 0 到 3 (DTFR0 到 DTFR3)

DTFR0 到 DTFR3 寄存器是 8 位寄存器，它通过片上外围 I/O 的中断请求信号控制着 DMA 传送的起始触发。

由这些寄存器设置的中断请求信号作为 DMA 传送的起始因素。

这些寄存器可以以 8 位为单元进行读取或是写入。但是，DFn 位只能以 1 位的方式读取或是写入。复位后这些寄存器的值为 00H。

复位后的值: 00H	读/写	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到3)										
			DFn ^注	DMA 传送请求标志						
			0	没有DMA 传送请求						
			1	有DMA 传送请求						

<R>

注 如果在 DMA 传送禁止期间，产生了一个被指定作为引起 DMA 传送的中断，对该位写入 0 将清除一个 DMA 发送请求。

- 注意事项 1. 当禁止 DMA 传送时(DCHCn.Enn 位的值为 0)，在以下时序设定 IFCn5 到 IFCn0 位。
- 在复位后开始发送第一个 DMA 传送之前的时期
 - 在通过 DCHCn.INITn 位将信道初始化后来开始 DMA 传送的时期
 - 在完成一个 DMA 传送(DCHCn.TCn 位为 1)和开始下一个 DMA 传送之间的时期
2. 在待机模式 (IDEL1, IDLE2, STOP, 或 sub-IDLE 模式) 下产生的中断请求不会开始 DMA 传送周期 (也不会将 DFn 位置 1)。
3. 如果通过 IFCn5 到 IFCn0 位选择了 DMA 开始因素，当所选择的片上外围 I/O 的中断产生时，DFn 位被置 1，而无论 DMA 传送是否被允许。在这种状态下如果允许 DMA，那么 DMA 发送立即开始。

备注 IFCn5 到 IFCn0 位，参见 表 18-1 DMA 开始因素。

表 18-1. DMA 开始因素

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	中断源
0	0	0	0	0	0	禁止由中断请求的 DMA
0	0	0	0	0	1	INTP0
0	0	0	0	1	0	INTP1
0	0	0	0	1	1	INTP2
0	0	0	1	0	0	INTP3
0	0	0	1	0	1	INTP4
0	0	0	1	1	0	INTP5
0	0	0	1	1	1	INTP6
0	0	1	0	0	0	INTP7
0	0	1	0	0	1	INTTQ0OV
0	0	1	0	1	0	INTTQ0CC0
0	0	1	0	1	1	INTTQ0CC1
0	0	1	1	0	0	INTTQ0CC2
0	0	1	1	0	1	INTTQ0CC3
0	0	1	1	1	0	INTTP0OV
0	0	1	1	1	1	INTTP0CC0
0	1	0	0	0	0	INTTP0CC1
0	1	0	0	0	1	INTTP1OV
0	1	0	0	1	0	INTTP1CC0
0	1	0	0	1	1	INTTP1CC1
0	1	0	1	0	0	INTTP2OV
0	1	0	1	0	1	INTTP2CC0
0	1	0	1	1	0	INTTP2CC1
0	1	0	1	1	1	INTTP3CC0
0	1	1	0	0	0	INTTP3CC1
0	1	1	0	0	1	INTTP4CC0
0	1	1	0	1	0	INTTP4CC1
0	1	1	0	1	1	INTTP5CC0
0	1	1	1	0	0	INTTP5CC1
0	1	1	1	0	1	INTTM0EQ0
0	1	1	1	1	0	INTCB0R/INTIIC1
0	1	1	1	1	1	INTCB0T
1	0	0	0	0	0	INTCB1R
1	0	0	0	0	1	INTCB1T
1	0	0	0	1	0	INTCB2R
1	0	0	0	1	1	INTCB2T
1	0	0	1	0	0	INTCB3R
1	0	0	1	0	1	INTCB3T
1	0	0	1	1	0	INTUA0R/INTCB4R
1	0	0	1	1	1	INTUA0T/INTCB4T
1	0	1	0	0	0	INTUA1R/INTIIC2
1	0	1	0	0	1	INTUA1T
1	0	1	0	1	0	INTUA2R/INTIIC0
1	0	1	0	1	1	INTUA2T
1	0	1	1	0	0	INTAD
1	0	1	1	0	1	INTKR
其他						禁止设定

备注 n = 0 ~ 3

18.4 传送目标

表 18-2 显示了传送目标之间的关系 (√: 允许传送, ×: 禁止传送)。

表 18-2. 传送目标之间的关系

		传送目的地			
		内部 ROM	片上外围 I/O	内部 RAM	外部存储器
源	片上外围 I/O	×	√	√	√
	内部 RAM	×	√	×	√
	外部存储器	×	√	√	√
	内部 ROM	×	×	×	×

注意事项 如果表 18-2 标记为“×”的源和目的地之间执行操作，结果将得不到保证。

18.5 传送模式

单一传送作为一种传送模式被支持。

在单一传送模式中，总线在每一次字节/半字传送后被释放。如果存在后续的 DMA 传送请求，传送立即被再次执行。这种操作直到发生了终止计数才会停止。

当 DMAC 释放总线后，如果发出一个具有更高优先级的 DMA 发送请求，该更高优先级的请求永远居先。

如果在传送周期内同一个信道发出一个新的传送请求同时另外一个信道产生一个具有较低优先级的传送请求，则在 CPU 释放总线之后，执行具有较低优先级信道的 DMA 传送 (在传送周期中同一信道新的传送请求被忽略)。

18.6 传送类型

作为一种传送类型，双周期传送被支持。

在双周期传送中，数据传送在两个周期内执行，一个读取周期和一个写入周期。

在读取周期中，传送的源地址被输出，从源到 **DMAC** 执行读取操作。在写入周期中，传送的目的地址被输出，从 **DMAC** 到目的执行写入操作。

时钟的空闲周期永远插入在读取周期和写入周期之间。如果用于进行双周期 **DMA** 传送的数据总线宽度在传送源和目的之间不一致，操作将按照如下步骤执行。

<16 位数据传送>

<1> 32 位总线 → 16 位总线的传送

在产生一个写入周期(16 位)之后，紧接着产生一个读取周期 (高 16 位处于高阻抗状态)。

<2> 16/32 位总线到 8 位总线的传送

16 位读取周期产生一次，随后产生两次 8 位写入周期。

<3> 8 位总线到 16/32 位总线的传送

8 位读取周期产生两次，随后 16 位写入周期产生一次。

<4> 16 位总线和 32 位总线之间的传送

16 位读取周期产生一次，随后 16 位写入周期产生一次。

对片上外围 I/O 寄存器(传送源/目的) 执行 **DMA** 传送时，一定要指定与寄存器大小相同的传送单元。例如，对一个 8 位寄存器执行 **DMA** 传送时，一定要指定字节(8 位)传送。

备注 每一种传送目的的总线宽度 (传送源/目的) 如下。

- 片上外围 I/O: 16 位总线宽度
- 内部 RAM: 32 位总线宽度
- 外部存储器: 8 位或者 16 位总线宽度

18.7 DMA 信道优先级

DMA 信道优先级固定如下。

DMA 信道 0 > DMA 信道 1 > DMA 信道 2 > DMA 信道 3

在每一个传送周期检测优先级。

18.8 与 DMA 传送相关的时间

响应 DMA 请求所需要的时间和 DMA 传送所需的最小时钟数如下所示。

单一传送: DMA 响应时间 (<1>) + 传送源存储器访问 (<2>) + 1^{注1} + 传送目的存储器访问(<2>)

DMA 周期		最小执行时钟数
<1> DMA 请求响应时间		4 时钟 (最小) + 噪声消除时间 ^{注2}
<2>存储器访问	外部存储器访问	取决于所连接的存储器
	内部 RAM 访问	2 时钟 ^{注3}
	外围 I/O 寄存器访问	3 时钟 + 由 VSWC 寄存器指定的等待周期数 ^{注4}

- 注
1. 在 DMA 传送的读取周期和写入周期之间总是要插入一个时钟。
 2. 如果指定一个外部中断 (INTPn) 作为触发开始 DMA 传送，噪声消除时间将增加 (n 的取值范围是 0 到 7)。
 3. DMA 周期需要两个时钟。
 4. 访问特定的外围 I/O 寄存器需要更多的等待周期 (详见 3.4.8 (2))。

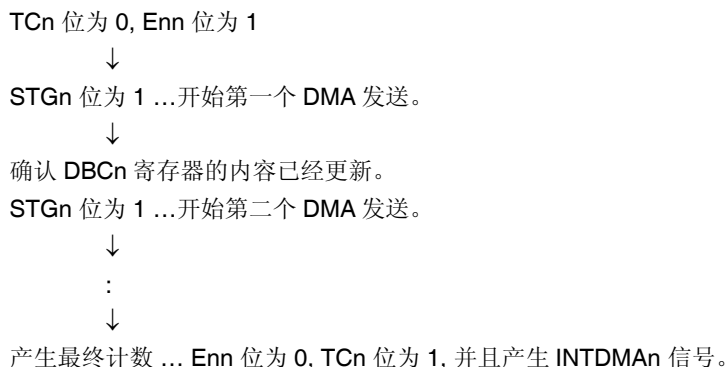
18.9 DMA 传送开始的因素

有两种类型的 DMA 传送开始因素，如下所示。

(1) 通过软件发出请求

如果当 DCHCn.TCn 位为 1 和 Enn 位为 1(DMA 传送允许)时把 STGn 位置 1，DMA 传送开始。

为了在此之后立即请求下一个 DMA 传送周期，通过 DBCn 寄存器确认先前的 DMA 传送周期已经完成，然后再次将 STGn 位置 1 (n 的取值范围是 0 到 3)。



(2) 通过片上外围 I/O 产生请求

如果当 DCHCn.TCn 位为 0 和 Enn 位为 1 (允许 DMA 传送)时，由 DTFRn 寄存器设定的片上外围 I/O 产生中断请求，DMA 发送开始。

- 注意事项**
1. 两种开始因素(软件触发和硬件触发)不能应用在同一个 DMA 信道，否则它们中间只有一个是有效的。有效的开始因素不能被识别。
 2. 在先前的 DMA 发送请求产生后或是在先前的 DMA 传送周期中产生的新的传送请求将被忽略。
 3. 相同的 DMA 信道的传送请求间隔的变化取决于在 DMA 发送周期中总线等待的设定，其它信道的开始状态或是外部总线的保持请求。特殊情况下，如注意事项 2 中的描述，在 DMA 传送周期之前或是 DMA 传送周期中同一信道产生的新的发送请求将被忽略。因此，相同 DMA 信道的传送请求间隔必须通过系统充分分隔。当使用软件触发时，先前产生的 DMA 传送周期是否完成可以通过更新 DBCn 寄存器检测。

18.10 DMA 中止因素

如果发生总线保持，DMA 传送将被中止。

如果传送在内部存储器/片上外围 I/O 与内部存储器/片上外围 I/O 之间执行时，会出现同样的状况。

当清除总线保持后，DMA 传送恢复。

18.11 DMA 传送结束

当 DCHCn.Enn 位清 0 和 TCn 位置 1 后，如果 DMA 传送已经完成了在 DBCn 寄存器中设定的次数，中断控制器(INTC)将产生一个 DMA 传送结束中断请求信号(INTDMAn) (n 的取值范围是 0 到 3)。

V850ES/JG2 不会向外部设备输出终止计数信号。因此，要通过 DMA 传送结束中断或是 TCn 位轮询确认 DMA 传送完成。

18.12 操作时序

图 18-1 到 18-4 显示了 DMA 操作时序。

图 18-1. DMA 的优先级 (1)

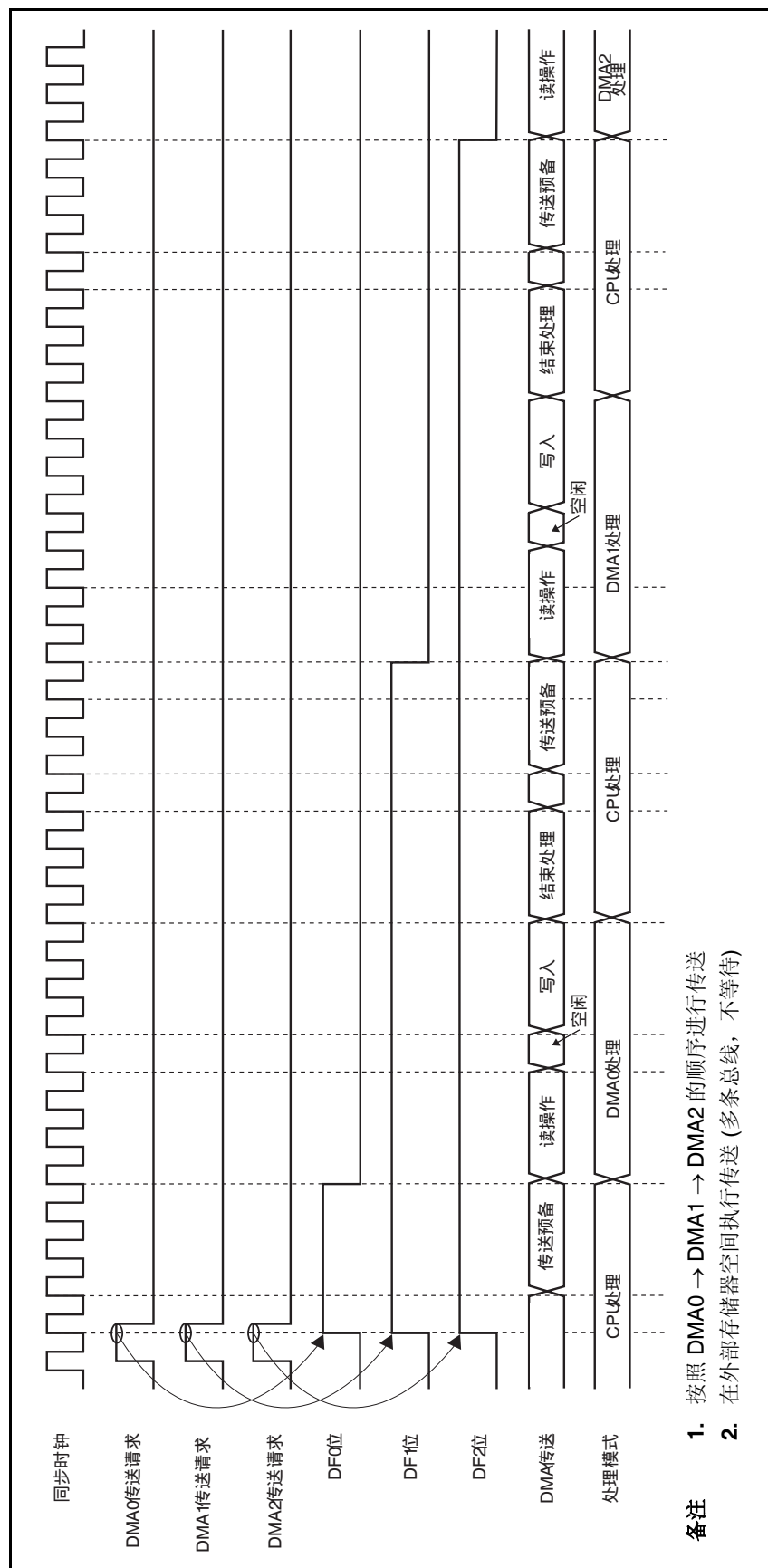


图 18-2. DMA 的优先级(2)

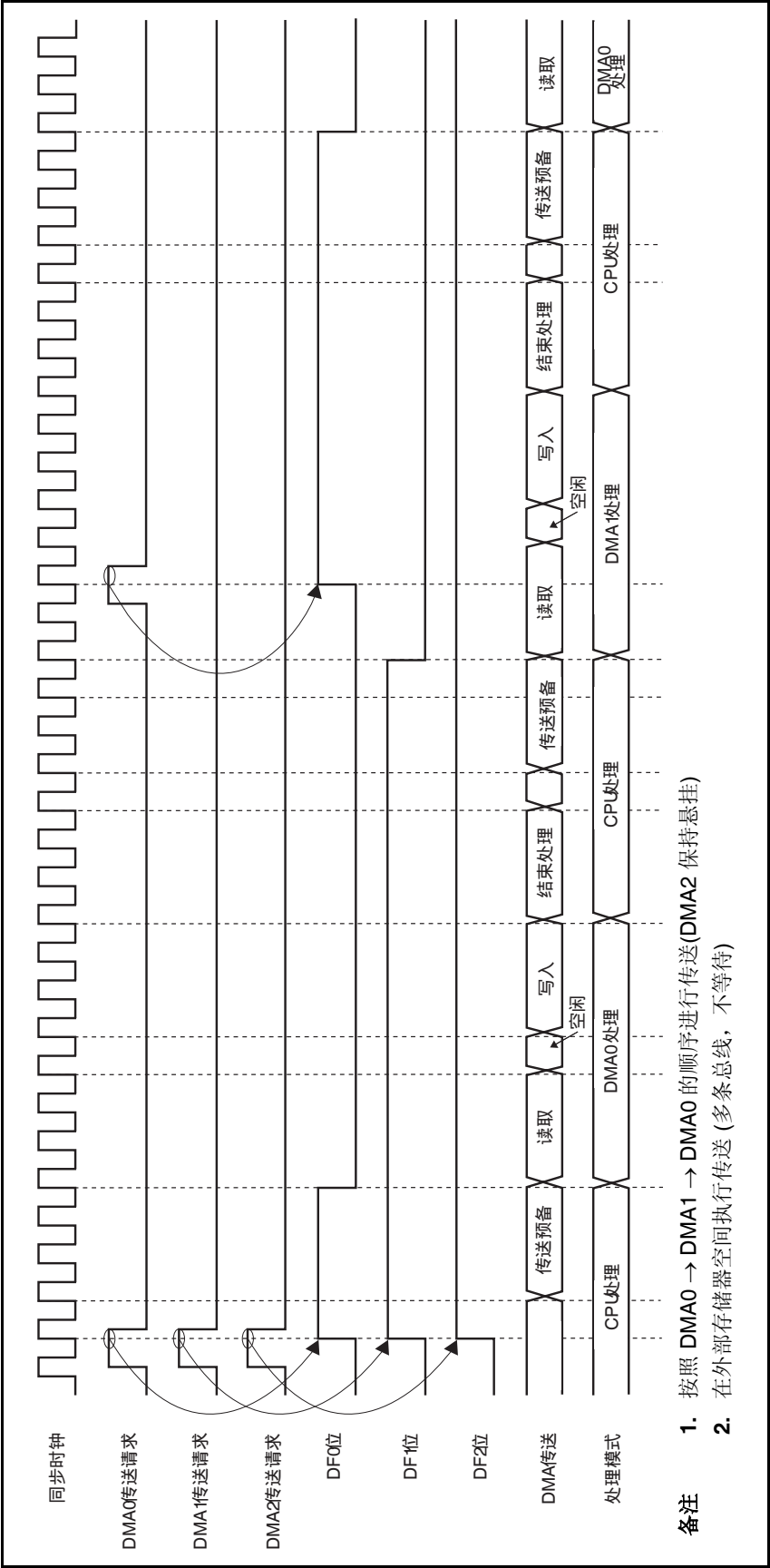


图 18-3. DMA 传送请求被忽略的时期 (1)

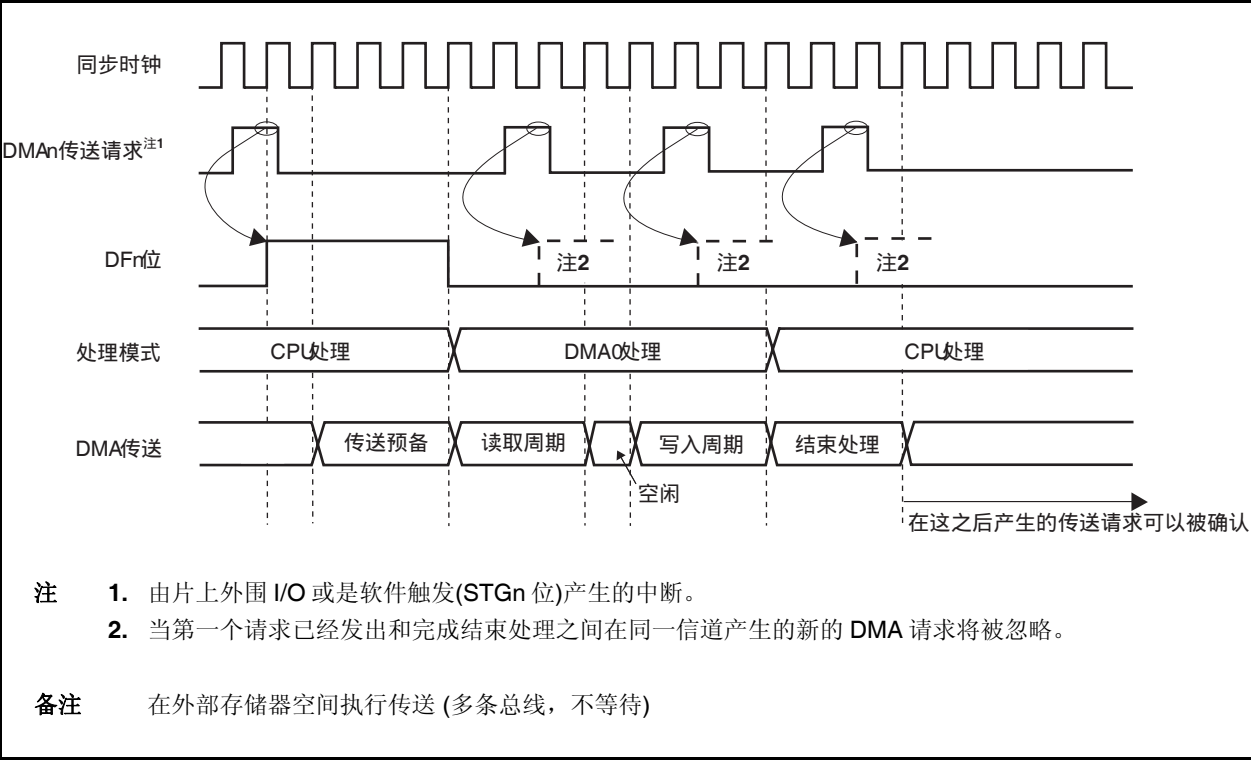
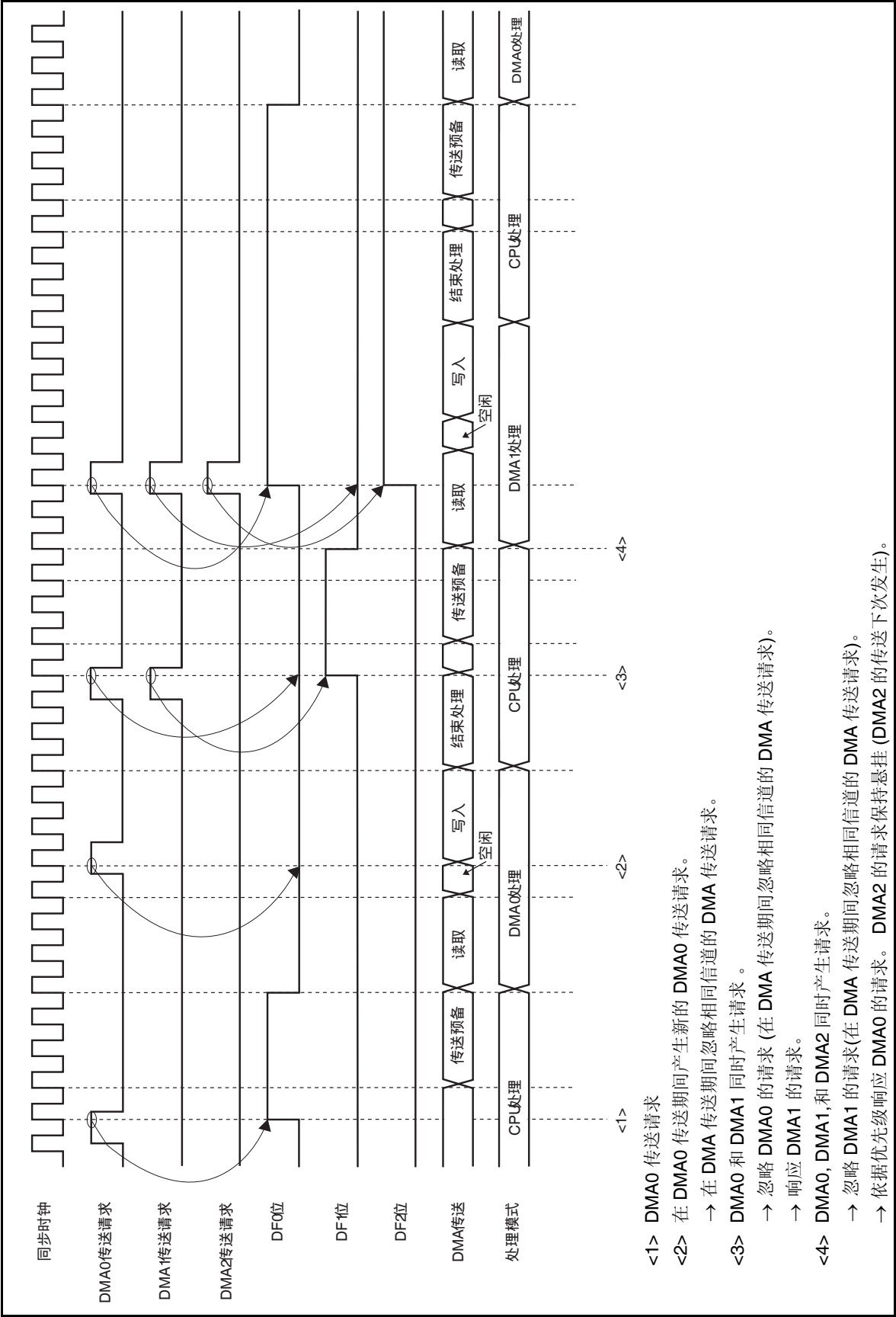


图 18-4. DMA 传送请求被忽略的时期(2)



18.13 注意事项

(1) VSWC 寄存器的注意事项

当使用 DMAC 时，一定要依照操作频率为 VSWC 寄存器设定适当的值。

当使用 VSWC 寄存器的缺省值 (77H)，或者为 VSWC 寄存器设定的值不恰当时，不会执行正确的操作 (VSWC 寄存器的详细信息，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 传送的注意事项

当在内部 RAM 中执行下列指令时，不要执行向/从内部 RAM(无论是源或者目的)中传送数据的 DMA 传送，这是由于 CPU 之后可能不会正确的运行。

- 在内部 RAM 中的位操作指令 (SET1, CLR1, 或是 NOT1)
- 在内部 RAM 中对于非对齐地址的数据访问指令

相反的，当向/从内部 RAM 中(无论是源或者目的)执行传送数据的 DMA 传送时，不要执行上述两种指令。

(3) 读取 DCHCn.TCn 位的注意事项 (n 的取值范围是 0 到 3)

当对 TCn 位进行读取操作时它将被清 0，但是它并不是自动被清 0 即使是在一个特殊的时序下。为了正确地将 TCn 位清 0，要加上下列处理过程。

(a) 当通过轮询 TCn 位的方式等待 DMA 传送完成

确认 TCn 位已经被置 1 (在 TCn 位为 1 被读取后)，然后另外再读取 TCn 位三次。

(b) 当在中断服务程序中读取 TCn 位时

执行读取 TCn 位 3 次。

(4) DMA 传送初始化过程 (将 DCHCn.INITn 位置 1)

当要执行 DMA 传送的信道被初始化时，即使将 INITn 位置 1，该信道也不会被初始化。为了正确的初始化信道，需要执行下列两个过程之一。

(a) 临时中止所有 DMA 信道的传送

初始化正在执行 DMA 传送的信道需要按以下步骤<1>到<7> 执行。

注，但是当执行第<5>步时 TCn 位被清 0。确定其它处理程序不需要 TCn 位为 1。

<1> 禁止中断 (DI)。

<2> 读取被强行中止的 DMA 信道之外的其它 DMA 信道的 DCHCn.Enn 位，然后将这个值传送到通用目的寄存器。

<3> 将正在被使用的 DMA 信道的 Enn 位 (包括被强行中止的信道)清 0。为了将最后一个 DMA 信道的 Enn 位清 0，需要执行两次清除指令。如果 DMA 传送的目的 (传送源/目的) 是内部 RAM，需要将该指令执行三次。

例如： 如果正在使用信道 0，1 和 2 (如果传送的目的不是内部 RAM)，按照以下顺序执行指令。

- 将 DCHC0.E00 位清 0。
- 将 DCHC1.E11 位清 0。
- 将 DCHC2.E22 位清 0。
- 再次将 DCHC2.E22 位清 0。

<4> 将被强行中止的 DMA 信道的 INITn 位置 1。

<5> 读取除被强行中止的信道外的其它各个信道的 TCn 位。如果在第<2>步中读取的 TCn 位和 Enn 位的值都为 1 (逻辑与的结果为 1)，将保存的 Enn 位清 0。

<6> 在第<5>步操作之后，将 Enn 位的值写入 DCHCn 寄存器。

<7> 允许中断 (EI)。

注意事项 一定要执行上述的第<5>步来防止对正常完成第<2>和<3>步之间 DMA 传送信道的 Enn 位的非法设置。

(b) 重复执行 INITn 位的设定直到传送被正确地强行中止

- <1> 抑制要被强行中止的信道的 DMA 请求源发出的请求(停止片上外围 I/O 的操作)。
- <2> 通过使用 DTFRn.DFn 位来检测要被强行中止的信道的 DMA 传送请求未处于悬挂状态。如果一个 DMA 传送请求处于悬挂状态, 需要等待直到执行完该悬挂请求。
- <3> 如果确定要被强行中止的信道的 DMA 请求未处于悬挂状态, 将 Enn 位清 0。
- <4> 再次将要被强行中止的信道的 Enn 位清 0。
如果要被强行中止的信道 (传送源/目的) 的传送的目的是内部 RAM, 再次执行该操作。
- <5> 复制要被强行中止的信道的传送初始值到通用目的寄存器。
- <6> 将要被强行中止的信道的 INITn 位置 1。
- <7> 读取要被强行中止的信道的 DBCn 寄存器的值, 并与第<5>步中的复制值相比较。如果两者不匹配, 重复第<6>步和第<7>步的操作。

- 备注**
- 1. 当在第<7>步中读取了 DBCn 寄存器的值后, 如果正确的完成了强行中止, 那么传送的初始值被读取。否则, 传送的剩余值被读取。
 - 2. 注意如果频繁使用一个被强行中止的信道之外的信道的 DMA 传送, 那么方式(b)可能会占用很长的时间。

(5) 临时停止 DMA 传送的程序 (将 Enn 位清 0)

停止和恢复 DMA 传送需要按以下步骤执行。

- <1> 抑制 DMA 请求源的传送请求 (停止片上外围 I/O 的操作)。
- <2> 使用 DFn 位来检测 DMA 传送请求未处于悬挂状态 (检测 DFn 位是否为 0)。
如果请求被悬挂, 等待直到被悬挂的 DMA 传送请求执行完毕。
- <3> 如果确认没有 DMA 传送请求被悬挂, 将 Enn 位清 0 (该操作停止 DMA 传送)。
- <4> 将 Enn 位置 1 恢复 DMA 传送。
- <5> 恢复被停止的 DMA 请求源的操作 (开始片上外围 I/O 的操作)。

(6) 存储器边界

如果在 DMA 传送期间传送源或者目的地址超出了 DMA 目标的区域, 那么操作将不会得到保证(外部存储器, 内部 RAM, 或是片上外围 I/O)。

(7) 传送非对齐数据

不支持在 16 位总线宽度上非对齐数据的 DMA 传送。

如果一个奇数地址被指定为传送源或者目的, 地址的最低有效位被强制假定为 0。

(8) 对于 CPU 的总线仲裁

由于 DMA 控制器对于总线的控制权优先于 CPU，因此发生在 DMA 传送期间的 CPU 访问被悬挂直到 DMA 传送周期完成且将总线释放给 CPU。

但是，CPU 可以访问那些未执行 DMA 传送的外部存储器，片上外围 I/O 和内部 RAM。

- 当在外部存储器和片上外围 I/O 之间执行 DMA 传送时，CPU 可以访问内部 RAM。
- 当在外部存储器之间执行 DMA 传送时，CPU 可以访问内部 RAM 和片上外围 I/O。

(9) 在 DMA 操作期间寄存器/位不准被重新写入

在 DMA 操作未被执行时在下面的时序中设定下列寄存器。

[寄存器]

- DSA_nH, DSA_nL, DDA_nH, DDA_nL, DBC_n, 和 DADC_n 寄存器
- DTFR_n.IFC_n5 到 DTFR_n.IFC_n0 位

[时序设定]

- 在复位后到开始第一个 DMA 传送之间的时期
- 在信道初始化到开始 DMA 传送的时间
- 在完成一个 DMA 传送 (TC_n 位为 1) 到开始下一个 DMA 发送之间的时期

(10) 一定要将下列寄存器的位清 0。

- DSA_nH 寄存器的位 14 到 10
- DDA_nH 寄存器的位 14 到 10
- DADC_n 寄存器的位 15, 13 到 8, 和 3 到 0
- DCHC_n 寄存器的位 6 到 3

<R>

(11) DMA 开始因素

不要使用相同的开始因素开始两个或以上的 DMA 信道。否则，已被设置的信道的 DMA 传送可能开始或者具有较低优先级的信道会比具有较高优先级的信道更早被响应。操作将得不到保证。

(12) 从 DSAn 和 DDAn 寄存器中读取数值

在 DMA 传送期间，可以在 DSAn 和 DDAn 寄存器中读取更新中的值 (n 的取值范围是 0 到 3)。

例如，如果当 DMA 传送源地址(DSAn 寄存器)为 0000FFFFH 且记数方式为递增(DADCn.SAD1 和 DADCn.SAD0 位为 00)，此时读取 DSAnH 寄存器和 DSAnL 寄存器，DSAn 寄存器的值的区别如下所示，这将取决于在 DSAnH 寄存器被读取之后 DMA 传送是否被立即执行。

(a) 如果在 DSAn 寄存器被读取时 DMA 传送没有发生

- <1> DSAnH 寄存器的读取值为：DSAnH = 0000H
- <2> DSAnL 寄存器的读取值为：DSAnL = FFFFH

(b) 如果在 DSAn 寄存器被读取时发生了 DMA 传送

- <1> DSAnH 寄存器的读取值为：DSAnH = 0000H
- <2> 发生 DMA 传送
- <3> 增加 DSAn 寄存器：DSAn = 00100000H
- <4> DSAnL 寄存器的读取值为：DSAnL = 0000H

第十九章 中断/异常处理功能

V850ES/JG2 为中断服务提供一个专门的中断控制器 (INTC)，它可以处理总共 57 个中断请求。

中断是指不受程序执行影响的事件发生，异常是指依赖于程序执行的事件发生。

V850ES/JG2 可以处理由片上外围硬件和外部事件源引起的中断请求信号。此外，异常处理可以通过 TRAP 指令(软件异常)或者产生一个异常事件(例如：存在非法操作码) (异常陷阱)开始。

19.1 特性

○ 中断

- 非屏蔽中断：2 个中断源
- 可屏蔽中断：外部：8 个中断源，内部：47 个中断源
- 8 个可编程中断优先级级别 (针对可屏蔽中断)
- 依据优先级进行复合中断控制
- 针对可屏蔽中断，可以使用屏蔽选项
- 噪声消除，边沿检测，和外部中断请求信号有效边沿说明。

○ 异常

- 软件异常：32 个异常源
- 异常陷阱：2 个异常源 (非法的操作码异常)

中断/异常源见表 19-1.

表 19-1. 中断源列表 (1/3)

类型	级别	缺省优先级	名称	触发	产生单元	异常代码	处理器地址	还原的 PC	中断控制寄存器
复位	中断	–	RESET	RESET 引脚输入 通过内部源的复位输入	RESET	0000H	00000000H	未定义	–
非屏蔽中断	中断	–	NMI	NMI 引脚有效边沿输入 t	Pin	0010H	00000010H	下一个 PC	–
		–	INTWDT2	WDT2 溢出	WDT2	0020H	00000020H	注 1	–
软件异常	异常	–	TRAP0n ^{注 2}	TRAP 指令	–	004nH ^{注 2}	00000040H	下一个 PC	–
		–	TRAP1n ^{注 2}	TRAP 指令	–	005nH ^{注 2}	00000050H	下一个 PC	–
异常陷阱	异常	–	ILGOP/ DBG0	非法的操作数/ DBTRAP 指令	–	0060H	00000060H	下一个 PC	–
可屏蔽	中断	0	INTLVI	低电压检测	POCLVI	0080H	00000080H	下一个 PC	LVIIC
		1	INTP0	外部中断引脚输入边沿检测 (INTP0)	Pin	0090H	00000090H	下一个 PC	PIC0
		2	INTP1	外部中断引脚输入边沿检测 (INTP1)	Pin	00A0H	000000A0H	下一个 PC	PIC1
		3	INTP2	外部中断引脚输入边沿检测 (INTP2)	Pin	00B0H	000000B0H	下一个 PC	PIC2
		4	INTP3	外部中断引脚输入边沿检测 (INTP3)	Pin	00C0H	000000C0H	下一个 PC	PIC3
		5	INTP4	外部中断引脚输入边沿检测 (INTP4)	Pin	00D0H	000000D0H	下一个 PC	PIC4
		6	INTP5	外部中断引脚输入边沿检测 (INTP5)	Pin	00E0H	000000E0H	下一个 PC	PIC5
		7	INTP6	外部中断引脚输入边沿检测 (INTP6)	Pin	00F0H	000000F0H	下一个 PC	PIC6
		8	INTP7	外部中断引脚输入边沿检测 (INTP7)	Pin	0100H	00000100H	下一个 PC	PIC7
		9	INTTQ0OV	TMQ0 溢出	TMQ0	0110H	00000110H	下一个 PC	TQ0OVIC
		10	INTTQ0CC0	TMQ0 捕获 0/比较 0 匹配	TMQ0	0120H	00000120H	下一个 PC	TQ0CCIC0
		11	INTTQ0CC1	TMQ0 捕获 1/比较 1 匹配	TMQ0	0130H	00000130H	下一个 PC	TQ0CCIC1
		12	INTTQ0CC2	TMQ0 捕获 2/比较 2 匹配	TMQ0	0140H	00000140H	下一个 PC	TQ0CCIC2
		13	INTTQ0CC3	TMQ0 捕获 3/比较 3 匹配	TMQ0	0150H	00000150H	下一个 PC	TQ0CCIC3
		14	INTTP0OV	TMP0 溢出	TMP0	0160H	00000160H	下一个 PC	TP0OVIC
		15	INTTP0CC0	TMP0 捕获 0/比较 0 匹配	TMP0	0170H	00000170H	下一个 PC	TP0CCIC0
		16	INTTP0CC1	TMP0 捕获 1/比较 1 匹配	TMP0	0180H	00000180H	下一个 PC	TP0CCIC1
		17	INTTP1OV	TMP1 溢出	TMP1	0190H	00000190H	下一个 PC	TP1OVIC
		18	INTTP1CC0	TMP1 捕获 0/比较 0 匹配	TMP1	01A0H	000001A0H	下一个 PC	TP1CCIC0
		19	INTTP1CC1	TMP1 捕获 1/比较 1 匹配	TMP1	01B0H	000001B0H	下一个 PC	TP1CCIC1
		20	INTTP2OV	TMP2 溢出	TMP2	01C0H	000001C0H	下一个 PC	TP2OVIC
		21	INTTP2CC0	TMP2 捕获 0/比较 0 匹配	TMP2	01D0H	000001D0H	下一个 PC	TP2CCIC0
		22	INTTP2CC1	TMP2 捕获 1/比较 1 匹配	TMP2	01E0H	000001E0H	下一个 PC	TP2CCIC1

- 注
1. 在 INTWDT2 例子中的还原，见 19.2.2 (2) INTWDT2 信号。
 2. n 的取值范围是 0 到 FH。

表 19-1. 中断源列表 (2/3)

类型	级别	缺省优先级	名称	触发	产生单元	异常代码	处理者地址	还原的 PC	中断控制寄存器
可屏蔽	中断	23	INTTP3OV	TMP3 溢出	TMP3	01F0H	000001F0H	下一个 PC	TP3OVIC
		24	INTTP3CC0	TMP3 捕获 0/比较 0 匹配	TMP3	0200H	00000200H	下一个 PC	TP3CCIC0
		25	INTTP3CC1	TMP3 捕获 1/比较 1 匹配	TMP3	0210H	00000210H	下一个 PC	TP3CCIC1
		26	INTTP4OV	TMP4 溢出	TMP4	0220H	00000220H	下一个 PC	TP4OVIC
		27	INTTP4CC0	TMP4 捕获 0/比较 0 匹配	TMP4	0230H	00000230H	下一个 PC	TP4CCIC0
		28	INTTP4CC1	TMP4 捕获 1/比较 1 匹配	TMP4	0240H	00000240H	下一个 PC	TP4CCIC1
		29	INTTP5OV	TMP5 溢出	TMP5	0250H	00000250H	下一个 PC	TP5OVIC
		30	INTTP5CC0	TMP5 捕获 0/比较 0 匹配	TMP5	0260H	00000260H	下一个 PC	TP5CCIC0
		31	INTTP5CC1	TMP5 捕获 1/比较 1 匹配	TMP5	0270H	00000270H	下一个 PC	TP5CCIC1
		32	INTTM0EQ0	TMM0 比较匹配	TMM0	0280H	00000280H	下一个 PC	TM0EQIC0
		33	INTCB0R/ INTIIC1	CSIB0 接收完成/ CSIB0 接收错误/ IIC1 传送完成	CSIB0/ IIC1	0290H	00000290H	下一个 PC	CB0RIC/ IIC1C1
		34	INTCB0T	CSIB0 连续传送写入允许	CSIB0	02A0H	000002A0H	下一个 PC	CB0TIC
		35	INTCB1R	CSIB1 接收完成/ CSIB1 接收错误	CSIB1	02B0H	000002B0H	下一个 PC	CB1RIC
		36	INTCB1T	CSIB1 连续传送写入允许	CSIB1	02C0H	000002C0H	下一个 PC	CB1TIC
		37	INTCB2R	CSIB2 接收完成/ CSIB2 接收错误	CSIB2	02D0H	000002D0H	下一个 PC	CB2RIC
		38	INTCB2T	CSIB2 连续传送写入允许	CSIB2	02E0H	000002E0H	下一个 PC	CB2TIC
		39	INTCB3R	CSIB3 接收完成/ CSIB3 接收错误	CSIB3	02F0H	000002F0H	下一个 PC	CB3RIC
		40	INTCB3T	CSIB3 连续传送写入允许	CSIB3	0300H	00000300H	下一个 PC	CB3TIC
		41	INTUA0R/ INTCB4R	UARTA0 接收完成/ CSIB4 接收完成/ CSIB4 接收错误	UARTA0/ CSIB4	0310H	00000310H	下一个 PC	UA0RIC/ CB4RIC
		42	INTUA0T/ INTCB4T	UARTA0 连续传送允许/ CSIB4 连续传送写入允许	UARTA0/ CSIB4	0320H	00000320H	下一个 PC	UA0TIC/ CB4TIC
		43	INTUA1R/ INTIIC2	UARTA1 接收完成/ UARTA1 接收错误/ IIC2 传送完成	UARTA1/ IIC2	0330H	00000330H	下一个 PC	UA1RIC/ IIC2C2
		44	INTUA1T	UARTA1 连续传送允许	UARTA1	0340H	00000340H	下一个 PC	UA1TIC
		45	INTUA2R/ INTIIC0	UARTA2 接收完成/ IIC0 传送完成	UARTA/ IIC0	0350H	00000350H	下一个 PC	UA2RIC/ IIC0C0
		46	INTUA2T	UARTA2 连续传送允许	UARTA2	0360H	00000360H	下一个 PC	UA2TIC
		47	INTAD	A/D 转换完成	A/D	0370H	00000370H	下一个 PC	ADIC

表 19-1. 中断源列表 (3/3)

类型	级别	缺省优先级	名称	触发	产生单元	异常代码	处理者地址	还原的 PC	中断控制寄存器
可屏蔽	中断	48	INTDMA0	DMA0 传送完成	DMA	0380H	00000380H	下一个 PC	DMAIC0
		49	INTDMA1	DMA1 传送完成	DMA	0390H	00000390H	下一个 PC	DMAIC1
		50	INTDMA2	DMA2 传送完成	DMA	03A0H	000003A0H	下一个 PC	DMAIC2
		51	INTDMA3	DMA3 传送完成	DMA	03B0H	000003B0H	下一个 PC	DMAIC3
		52	INTKR	按键返回中断	KR	03C0H	000003C0H	下一个 PC	KRIC
		53	INTWTI	钟表定时器间隔	WT	03D0H	000003D0H	下一个 PC	WTIIC
		54	INTWT	钟表定时器参考时间	WT	03E0H	000003E0H	下一个 PC	WTIC

备注 1. 缺省优先级: 表示当两个或更多的可屏蔽中断请求同时发生时, 它们的有限顺序。最高值为 0。

非屏蔽中断的优先级顺序为 INTWDT2 > NMI。

还原 PC: 当中断服务产生时被保存到 EIPC, FEPC, 或是 DBPC 的程序计数器(PC)的值。但是, 当非屏蔽中断或者可屏蔽中断被确认, 且下列指令之一正在被执行时, 还原的 PC 不会成为下一个 PC(如果在中断执行期间确认中断, 则该操作停止并在中断服务结束后还原)。

- 载入指令 (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
- 除法指令 (DIV, DIVH, DIVU, DIVHU)
- 预备, 处理指令(仅限中断在堆栈指针更新之前产生的情况)

下一个 PC: 在中断/异常处理之后的开始处理的 PC 的值。

2. 当一个非法的操作代码异常发生时, 非法指令的执行地址通过保存的 PC 值减去 4 得出。

19.2 非屏蔽中断

非屏蔽中断请求会被无条件的确认，即使是在禁止中断的条件下(DI)。一个非屏蔽中断不会受到优先级的控制，并且相对于其它的中断请求而言具有绝对的优先权。

在本产品中有列两种非屏蔽中断请求信号。

- NMI 引脚输入 (NMI)
- 非屏蔽中断请求信号可以通过看门狗定时器溢出产生(INTWDT2)

NMI 引脚的有效边沿可以选择以下四种方式：“上升沿”，“下降沿”，“双边沿”，和“不检测边沿”。

当 WDTM2.WDM21 位和 WDTM2.WDM20 位设为 01 时，由看门狗定时器 2 溢出而产生的非屏蔽中断请求信号(INTWDT2)生效。

如果两个或是更多的非屏蔽中断请求同时产生，具有较高优先级的中断首先生效，如下所示(具有较低优先级的中断请求信号将被忽略)。

INTWDT2 > NMI

如果在非屏蔽中断正在进行的同时，发出一个新的 NMI 或者 INTWDT2 请求信号，将按照如下方式进行。

(1) 如果在非屏蔽中断正在进行的同时，发出一个新的 NMI 请求信号

新的 NMI 请求信号被保持，忽略 PSW.NP 位的值。而新的 NMI 请求信号会在当前的 NMI 运行完毕后被确认(在 RETI 指令执行后)。

(2) 如果在非屏蔽中断正在进行的同时，发出一个 INTWDT2 请求信号

当存在正在进行的 NMI 时，如果 NP 位的值是(1)，则 INTWDT2 请求信号被保持。被保持的 INTWDT2 请求信号会在当前的 NMI 运行完毕后被确认(在 RETI 指令执行后)。

当存在正在进行的 NMI 时，如果 NP 位为 (0)，则立即执行新产生的 INTWDT2 请求信号 (NMI 服务停止)。

注意事项 由非屏蔽中断请求信号(INTWDT2)执行的非屏蔽中断服务，参考 19.2.2 (2) INTWDT2 信号。

图 19-1. 非屏蔽中断请求信号确认操作 (1/2)

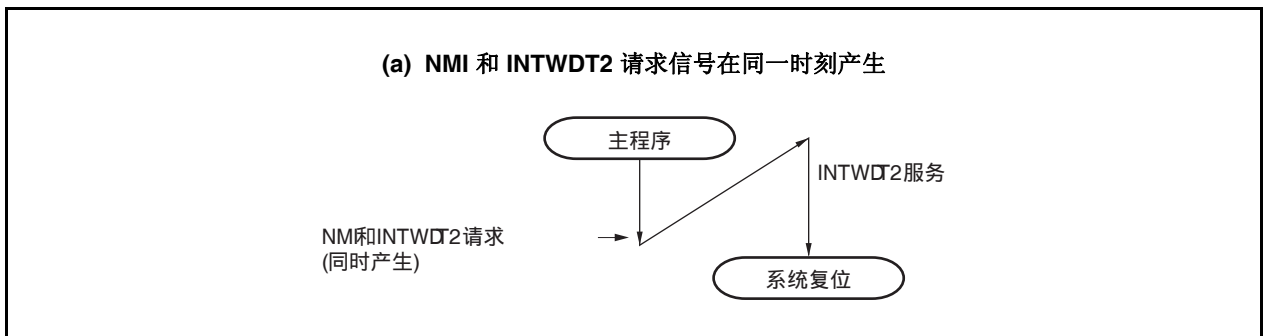
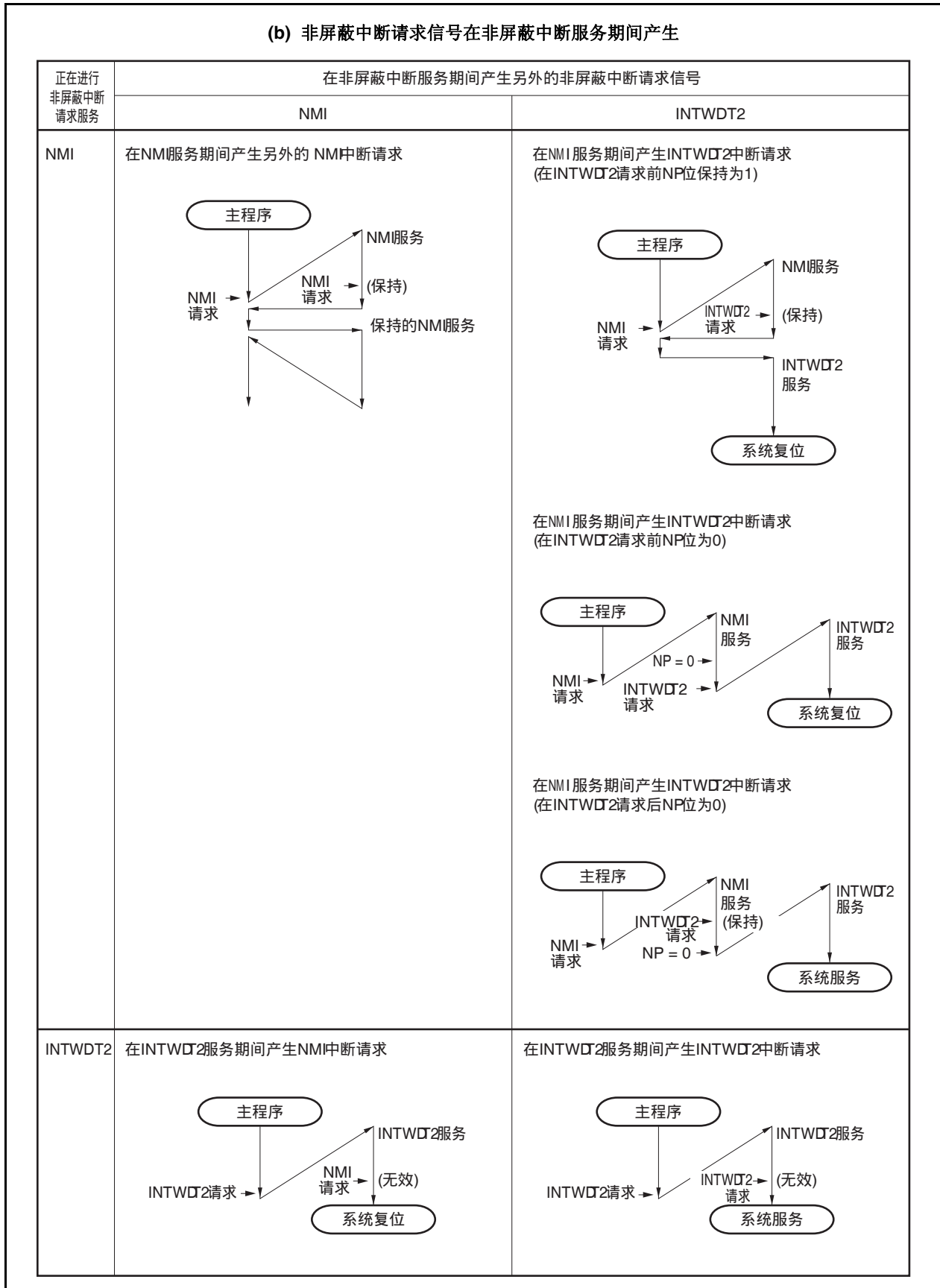


图 19-1. 非屏蔽中断请求信号确认操作(2/2)



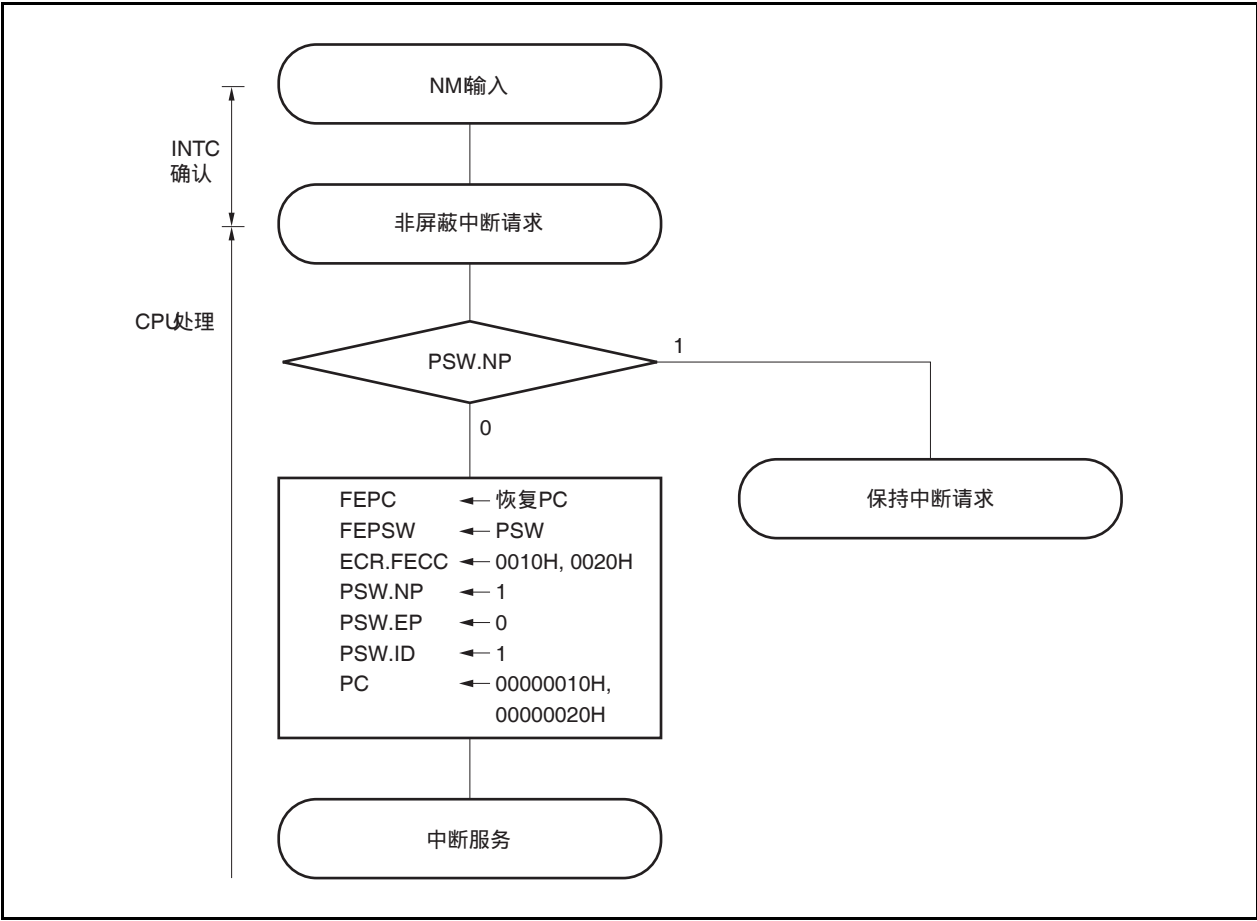
19.2.1 操作

如果产生一个非屏蔽中断请求信号，CPU 执行下列处理，并且将操作权转移给处理例程。

- <1> 将还原 PC 保存的 FEPC。
- <2> 将当前的 PSW 保存到 FEPSW。
- <3> 向 ECR 的高半字(FECC)写入异常代码 (0010H, 0020H)。
- <4> 将 PSW.NP 位和 PSW.ID 位置 1 并将 PSW.EP 位清 0。
- <5> 在 PC 中设置相应的非屏蔽中断的处理地址(00000010H, 00000020H)，然后转移操作权。

非屏蔽中断的服务过程如下图所示。

图 19-2. 非屏蔽中断的服务过程



19.2.2 还原

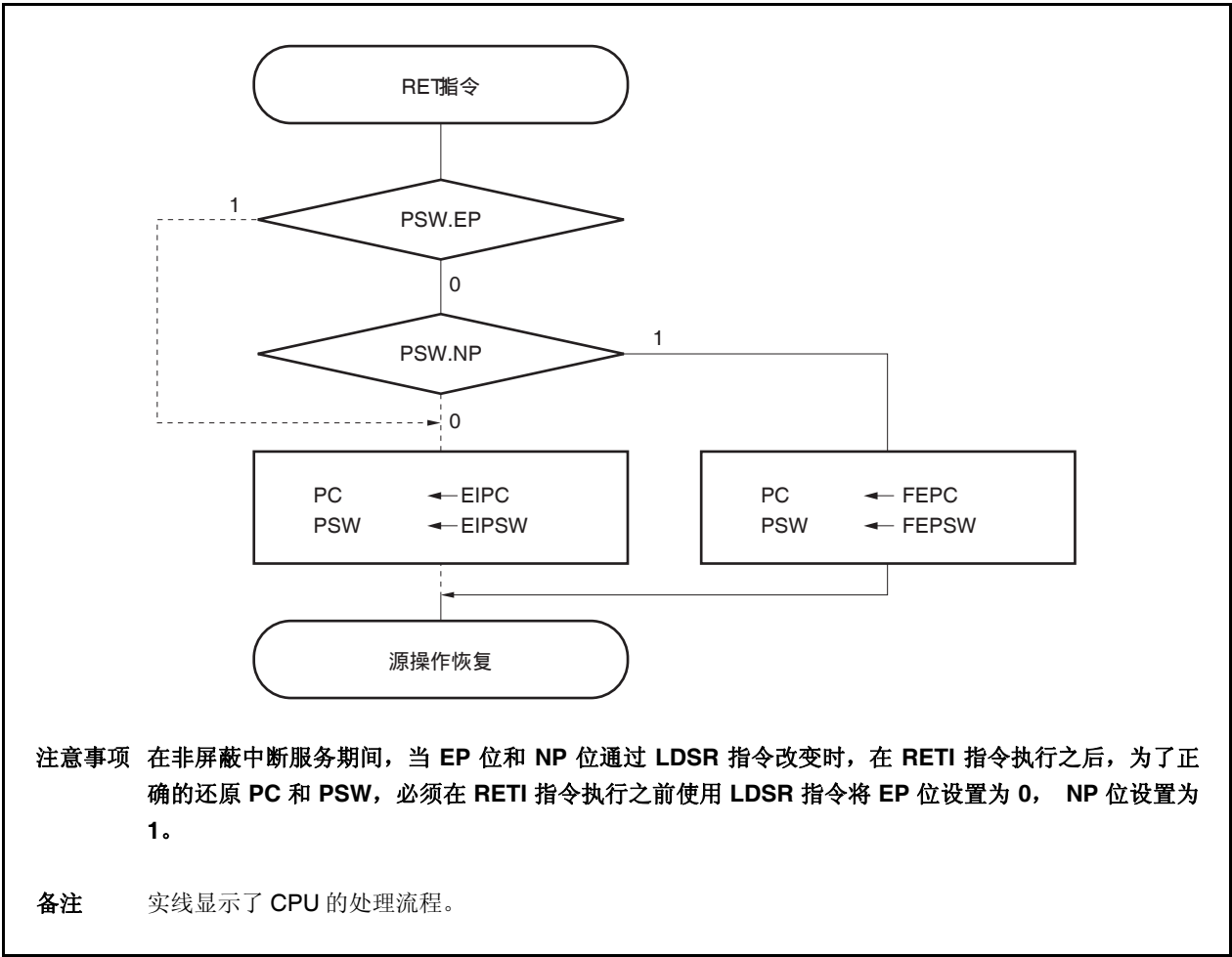
(1) 通过 NMI 引脚输入

通过 RETI 指令从 NMI 服务中执行还原。
当执行 RETI 指令时，CPU 执行下列处理，并且将操作权转移给 PC 存储的地址。

- <1> 分别从 FEPC 和 FEPSW 中加载被保存的 PC 和 PSW 的值，因为 PSW.EP 位的值是 0 且 PSW.NP 的值为 1。
- <2> 操作权转移给还原的 PC 和 PSW 的地址。

下图举例说明了 RETI 指令是如何处理的。

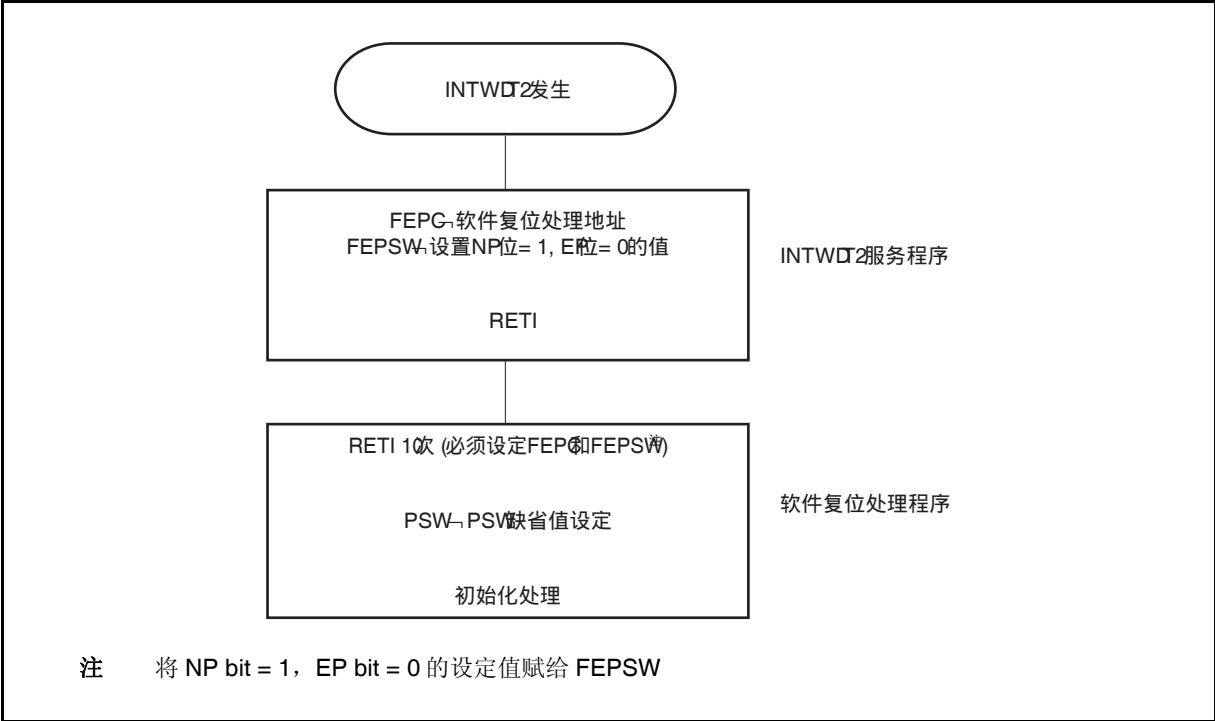
图 19-3. RETI 指令处理过程



(2) 通过 INTWDT2 信号

禁止使用 RETI 指令从非屏蔽中断服务(通过非屏蔽中断请求 INTWDT2 产生)中恢复。执行下列软件复位处理。

图 19-4. 软件复位处理



19.2.3 NP 标志

NP 标志是一个状态标志，它指示了正在执行非屏蔽中断。
当非屏蔽中断请求被确认时会设置该标志，并且还可以保证多重非屏蔽中断的情况出现。

复位后的值: 00000020H

	31																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
--	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

19.3 可屏蔽中断

可屏蔽中断请求信号可以通过中断控制寄存器屏蔽。V850ES/JG2 拥有 55 个可屏蔽中断源。

如果有两个或者更多的可屏蔽中断请求信号同时产生，它们将依照缺省优先级的顺序被确认。除了缺省优先级之外，还有八个优先级等级可以使用，这需要使用中断控制寄存器 (可变成优先级控制)。

如果一个中断请求信号被确认，将禁止对其它可屏蔽中断请求信号的确认，从而进入了中断禁止状态(DI)。

当在中断服务程序中执行了 EI 指令时，进入中断允许状态(EI)，这将允许具有比正在进行处理的当前中断请求信号更高优先级(通过中断控制寄存器设定)的中断。注意只有优先级更高的中断有这种功能，具有相同优先级的中断不能实现嵌套。

但是为了允许多重中断，在执行 EI 指令之前需要将 EIPC 和 EIPSW 保存到内存或是通用目的寄存器当中，并且在 RETI 指令恢复 EIPC 和 EIPSW 的原始值之前执行 DI 指令。

19.3.1 操作

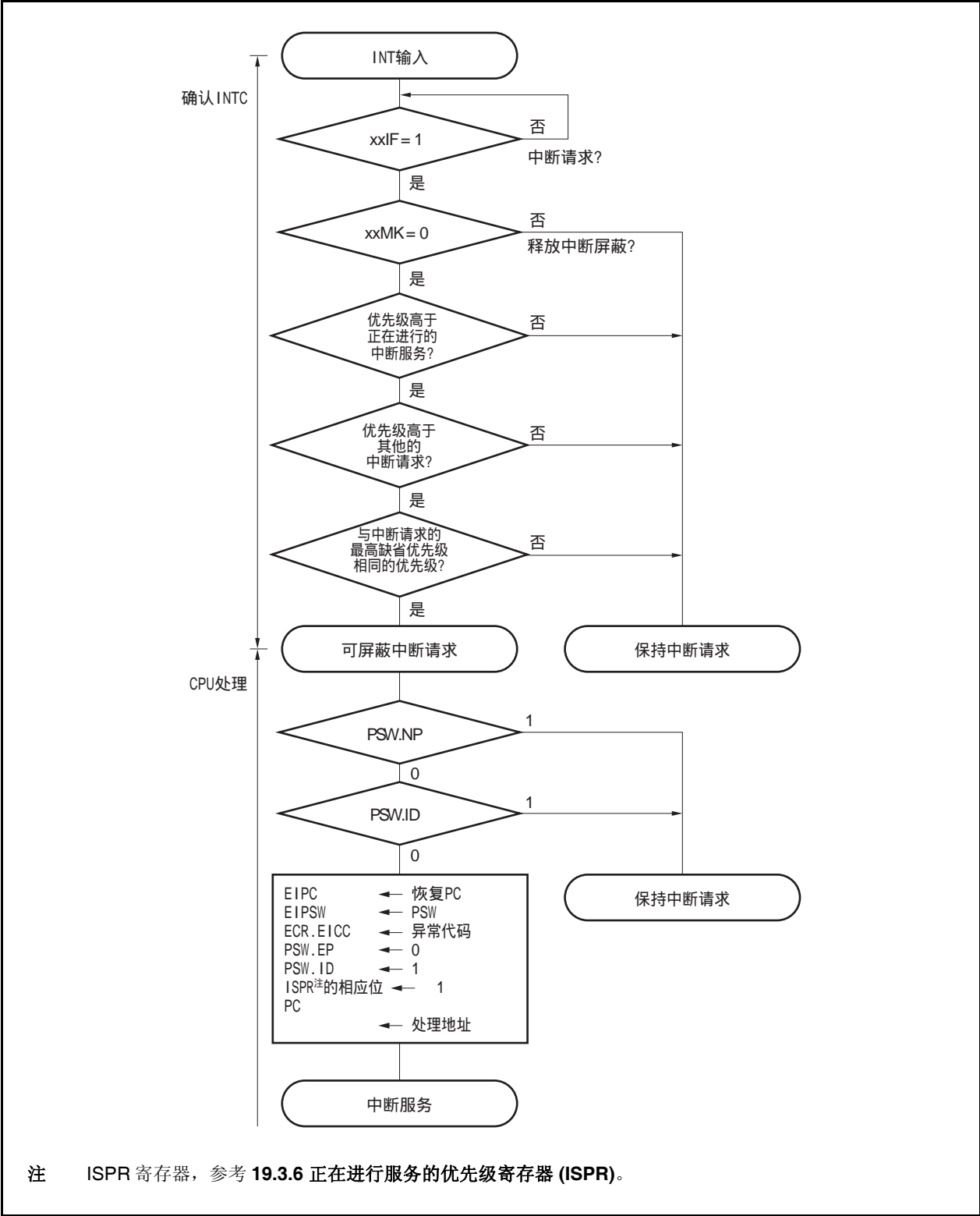
如果产生了可屏蔽中断，CPU 将执行下列处理，并且将操作权转移给处理例程。

- <1> 将恢复的 PC 保存到 EIPC。
- <2> 将当前的 PSW 保存到 EIPSW。
- <3> 向 ECR 的低半字 (EICC)写入一个异常代码。
- <4> 将 PSW.ID 位置 1 且将 PSW.EP 位清 0。
- <5> 将相应中断的处理地址设置给 PC，并且转移控制权。

通过 INTC 屏蔽的可屏蔽中断请求信号和在另一个中断正在执行过程中(当 PSW.NP 位=1 或是 PSW.ID 位=1)产生的可屏蔽中断请求信号在 INTC 中被保持。在这种情况下，一个新的可屏蔽中断服务开始是与可屏蔽中断请求的优先级设定是一致的。这需要可屏蔽中断未被屏蔽，而且没有通过 RETI 指令或是 LDSR 指令将 NP 位和 ID 位清 0。

可屏蔽中断是如何执行的将在下面举例说明。

图 19-5. 可屏蔽中断服务



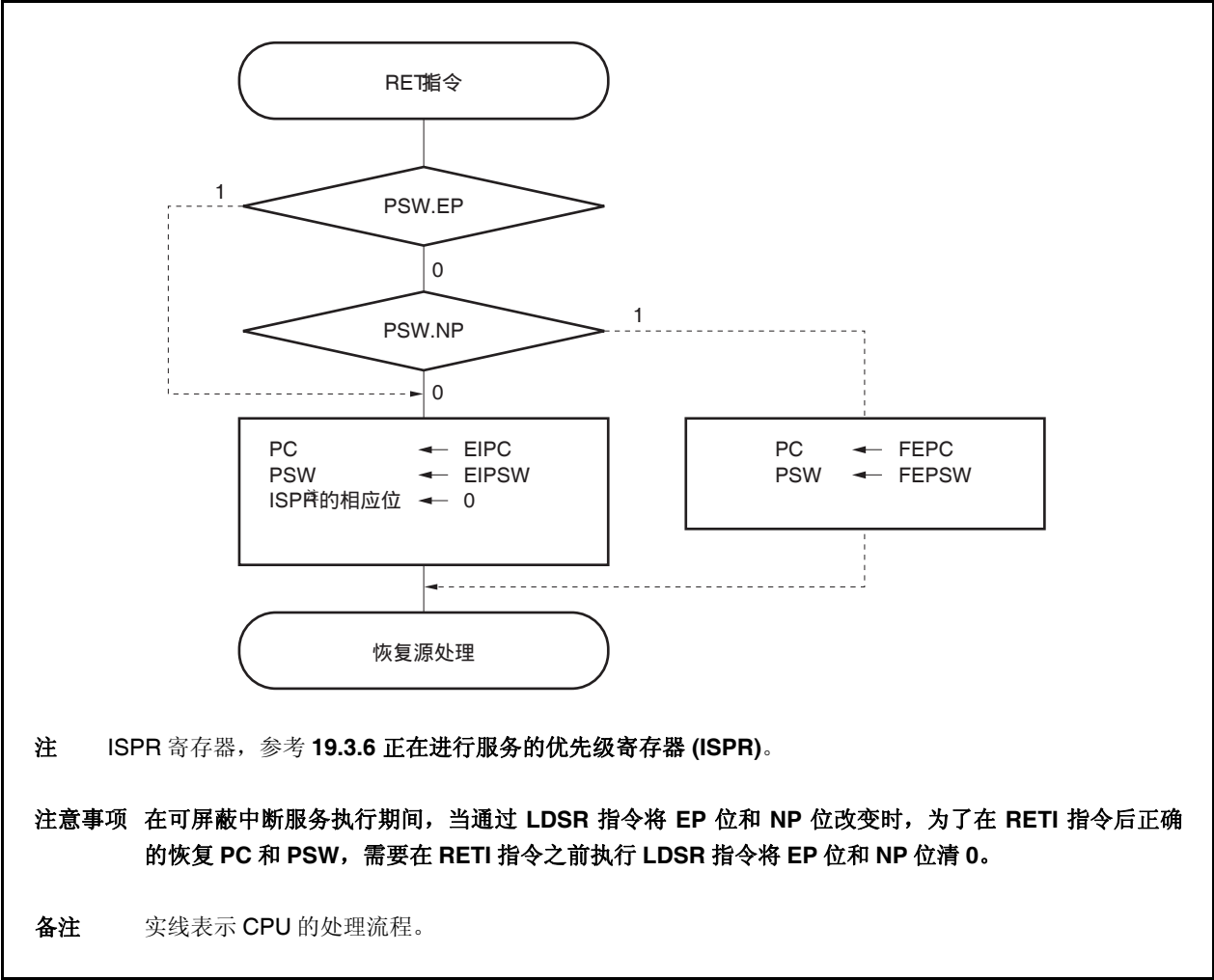
19.3.2 恢复

通过执行 RETI 指令能够从可屏蔽中断服务中恢复。
当 RETI 指令执行后，CPU 将执行下列步骤，且将操作权转移给恢复的 PC 地址。

- <1> 从 EIPC 和 EIPSW 中加载恢复的 PC 和 PSW，这是因为 PSW.EP 位为 0 并且 PSW.NP 位为 0。
- <2> 将操作权转移给恢复 PC 和 PSW 的地址。

下图举例说明 RETI 指令的处理过程。

图 19-6. RETI 指令处理过程



19.3.3 可屏蔽中断的优先级

INTC 执行多重中断服务，这是指当一个中断服务正在进行的时候确认另外一个中断。多重中断可以通过优先级来控制。

一共有两种类型的优先级控制：基于缺省优先级别的控制和基于可编程优先级别的控制，后者是通过在中断控制(**xxICn**)寄存器的中断优先级指定位(**xxPRn**)的设置来实现的。当两个具有相同优先级的中断同时产生时，中断请求信号将依照预先指定给每一种中断类型的优先级别(缺省优先级别)顺序进行中断服务。如需更多的信息，参考**表 19-1 中断/异常源列表**。可编程的优先级通过设定优先级别制定标志分八个等级对用户化的中断请求信号进行控制。

注意当一个中断请求信号被确认后，**PSW.ID** 标志将自动置 1。因此，当使用复合中断时，需要预先将 **ID** 标志清 0 来设定中断允许模式(例如，在中断服务程序中设置 **EI** 指令)。

备注 **xx**: 表示任意的外围单元的名称(参考**表 19-2 中断控制寄存器(xxICn)**)
 n: 外围单元编号(参考**表 19-2 中断控制寄存器(xxICn)**)。

图 19-7. 中断嵌套举例(1/2)

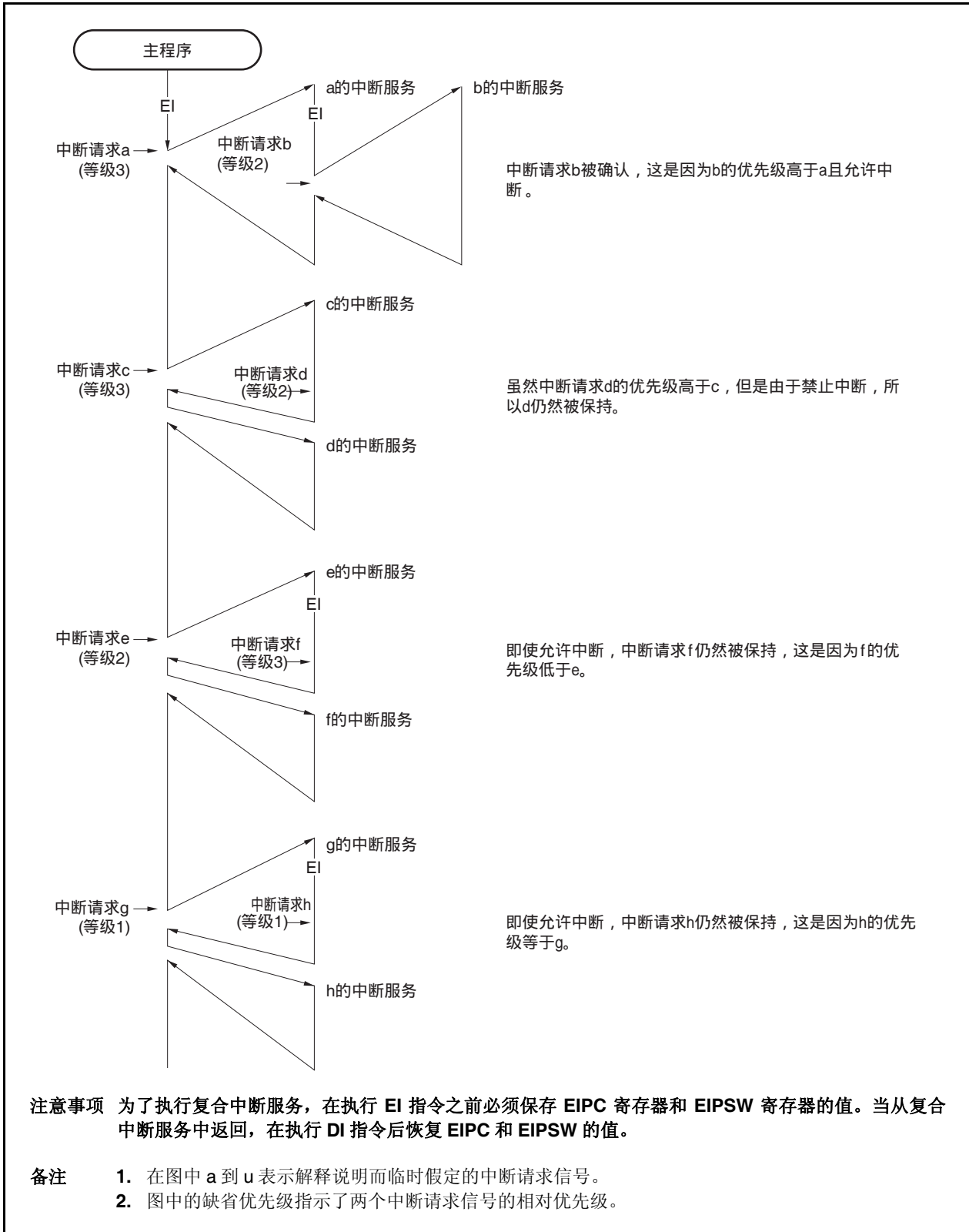


图 19-7. 中断嵌套举例(2/2)

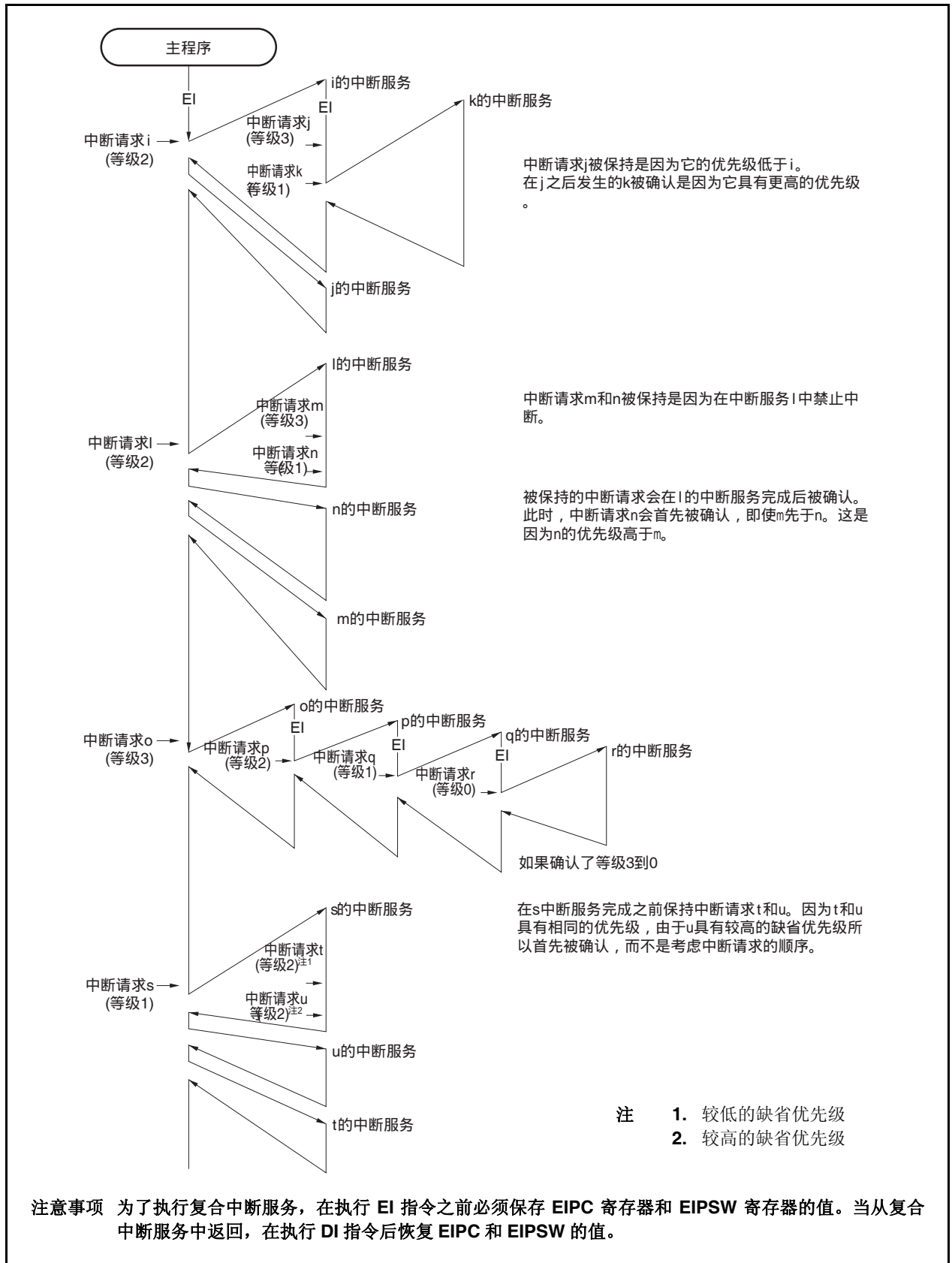
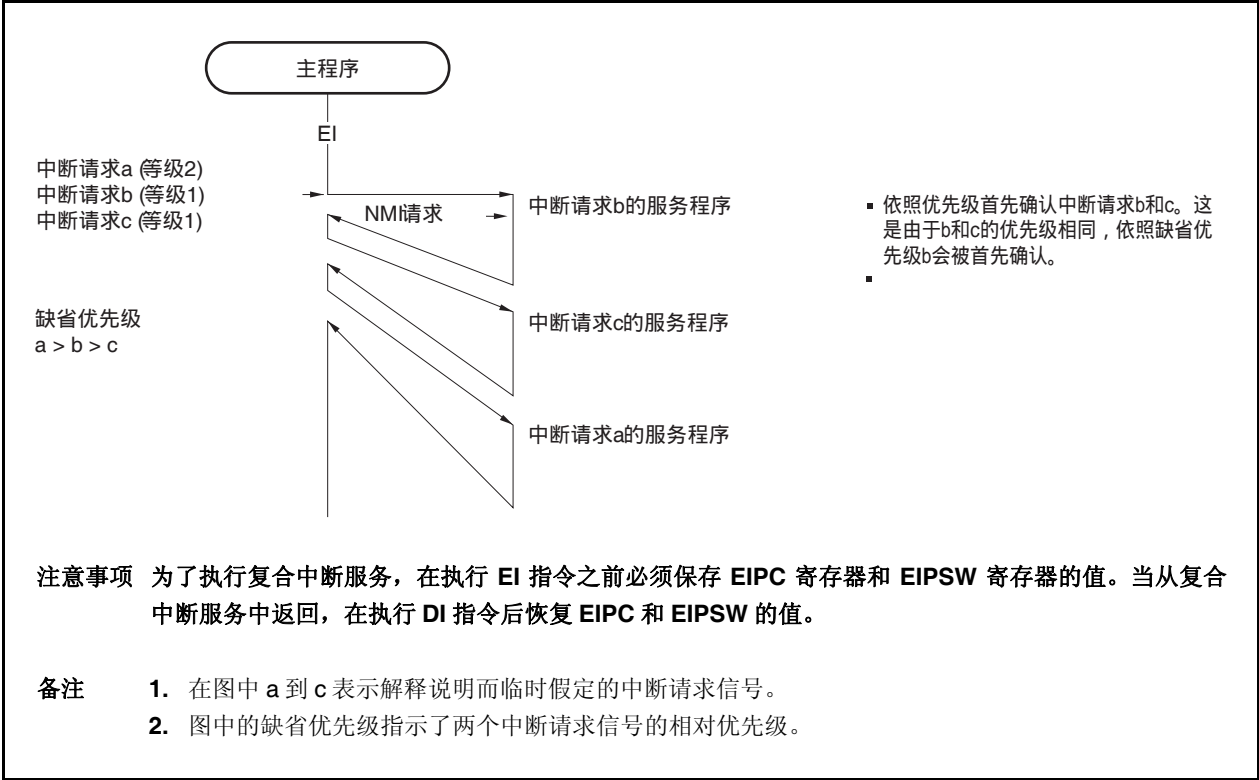


图 19-8. 服务中断请求信号同时发生举例



19.3.4 中断控制寄存器 (xxICn)

xxICn 寄存器被分配给每一个中断请求信号(可屏蔽中断)并且设置每一个可屏蔽中断请求控制条件。
该寄存器可以通过 8 位或是 1 位单元读取或是写入。
复位后该寄存器的值为 47H。

注意事项 禁止中断 (DI)或者屏蔽中断来读取 xxICn.xxIFn 位。如果在允许中断(EI)或是中断未被屏蔽的时候，读取 xxIFn 位，确认中断且读取该位冲突时数据可能有误。

复位后的值: 47H 读/写 地址: FFFFF112H to 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	未发出中断请求
1	发出中断请求

xxMKn	中断屏蔽标志
0	允许中断服务
1	禁止中断服务 (保持)

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 如果确认中断信号后，xxIFn 标志通过硬件自动复位。

备注 xx: 表示任意的外围单元的名称(参考表 19-2 中断控制寄存器(xxICn))
 n: 外围单元编号(参考表 19-2 中断控制寄存器(xxICn))。

中断控制寄存器的位和地址如下所示。

表 19-2. 中断控制寄存器(xxlCn) (1/2)

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	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF114H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF116H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF118H	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
FFFFF11AH	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
FFFFF11CH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
FFFFF11EH	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60
FFFFF120H	PIC7	PIF7	PMK7	0	0	0	PPR72	PPR71	PPR70
FFFFF122H	TQ0OVIC	TQ0OVIF	TQ0OVMK	0	0	0	TQ0OVPR2	TQ0OVPR1	TQ0OVPR0
FFFFF124H	TQ0CCIC0	TQ0CCIF0	TQ0CCMK0	0	0	0	TQ0CCPR02	TQ0CCPR01	TQ0CCPR00
FFFFF126H	TQ0CCIC1	TQ0CCIF1	TQ0CCMK1	0	0	0	TQ0CCPR12	TQ0CCPR11	TQ0CCPR10
FFFFF128H	TQ0CCIC2	TQ0CCIF2	TQ0CCMK2	0	0	0	TQ0CCPR22	TQ0CCPR21	TQ0CCPR20
FFFFF12AH	TQ0CCIC3	TQ0CCIF3	TQ0CCMK3	0	0	0	TQ0CCPR32	TQ0CCPR31	TQ0CCPR30
FFFFF12CH	TP0OVIC	TP0OVIF	TP0OVMK	0	0	0	TP0OVPR2	TP0OVPR1	TP0OVPR0
FFFFF12EH	TP0CCIC0	TP0CCIF0	TP0CCMK0	0	0	0	TP0CCPR02	TP0CCPR01	TP0CCPR00
FFFFF130H	TP0CCIC1	TP0CCIF1	TP0CCMK1	0	0	0	TP0CCPR12	TP0CCPR11	TP0CCPR10
FFFFF132H	TP1OVIC	TP1OVIF	TP1OVMK	0	0	0	TP1OVPR2	TP1OVPR1	TP1OVPR0
FFFFF134H	TP1CCIC0	TP1CCIF0	TP1CCMK0	0	0	0	TP1CCPR02	TP1CCPR01	TP1CCPR00
FFFFF136H	TP1CCIC1	TP1CCIF1	TP1CCMK1	0	0	0	TP1CCPR12	TP1CCPR11	TP1CCPR10
FFFFF138H	TP2OVIC	TP2OVIF	TP2OVMK	0	0	0	TP2OVPR2	TP2OVPR1	TP2OVPR0
FFFFF13AH	TP2CCIC0	TP2CCIF0	TP2CCMK0	0	0	0	TP2CCPR02	TP2CCPR01	TP2CCPR00
FFFFF13CH	TP2CCIC1	TP2CCIF1	TP2CCMK1	0	0	0	TP2CCPR12	TP2CCPR11	TP2CCPR10
FFFFF13EH	TP3OVIC	TP3OVIF	TP3OVMK	0	0	0	TP3OVPR2	TP3OVPR1	TP3OVPR0
FFFFF140H	TP3CCIC0	TP3CCIF0	TP3CCMK0	0	0	0	TP3CCPR02	TP3CCPR01	TP3CCPR00
FFFFF142H	TP3CCIC1	TP3CCIF1	TP3CCMK1	0	0	0	TP3CCPR12	TP3CCPR11	TP3CCPR10
FFFFF144H	TP4OVIC	TP4OVIF	TP4OVMK	0	0	0	TP4OVPR2	TP4OVPR1	TP4OVPR0
FFFFF146H	TP4CCIC0	TP4CCIF0	TP4CCMK0	0	0	0	TP4CCPR02	TP4CCPR01	TP4CCPR00
FFFFF148H	TP4CCIC1	TP4CCIF1	TP4CCMK1	0	0	0	TP4CCPR12	TP4CCPR11	TP4CCPR10
FFFFF14AH	TP5OVIC	TP5OVIF	TP5OVMK	0	0	0	TP5OVPR2	TP5OVPR1	TP5OVPR0
FFFFF14CH	TP5CCIC0	TP5CCIF0	TP5CCMK0	0	0	0	TP5CCPR02	TP5CCPR01	TP5CCPR00
FFFFF14EH	TP5CCIC1	TP5CCIF1	TP5CCMK1	0	0	0	TP5CCPR12	TP5CCPR11	TP5CCPR10
FFFFF150H	TM0EQIC0	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR02	TM0EQPR01	TM0EQPR00
FFFFF152H	CB0RIC/ IICIC1	CB0RIF/ IICIF1	CB0RMK/ IICMK1	0	0	0	CB0RPR2/ IICPR12	CB0RPR1/ IICPR11	CB0RPR0/ IICPR10
FFFFF154H	CB0TIC	CB0TIF	CB0TMK	0	0	0	CB0TPR2	CB0TPR1	CB0TPR0
FFFFF156H	CB1RIC	CB1RIF	CB1RMK	0	0	0	CB1RPR2	CB1RPR1	CB1RPR0
FFFFF158H	CB1TIC	CB1TIF	CB1TMK	0	0	0	CB1TPR2	CB1TPR1	CB1TPR0
FFFFF15AH	CB2RIC	CB2RIF	CB2RMK	0	0	0	CB2RPR2	CB2RPR1	CB2RPR0
FFFFF15CH	CB2TIC	CB2TIF	CB2TMK	0	0	0	CB2TPR2	CB2TPR1	CB2TPR0
FFFFF15EH	CB3RIC	CB3RIF	CB3RMK	0	0	0	CB3RPR2	CB3RPR1	CB3RPR0
FFFFF160H	CB3TIC	CB3TIF	CB3TMK	0	0	0	CB3TPR2	CB3TPR1	CB3TPR0

表 19-2. 中断控制寄存器(xxICn) (2/2)

地址	寄存器	位							
		<7>	<6>	5	4	3	2	1	0
FFFFF162H	UA0RIC/ CB4RIC	UA0RIF/ CB4RIF	UA0RMK/ CB4RMK	0	0	0	UA0RPR2/ CB4RPR2	UA0RPR1/ CB4RPR1	UA0RPR0/ CB4RPR0
FFFFF164H	UA0TIC/ CB4TIC	UA0TIF/ CB4TIF	UA0TMK/ CB4TMK	0	0	0	UA0TPR2/ CB4TPR2	UA0TPR1/ CB4TPR1	UA0TPR0/ CB4TPR0
FFFFF166H	UA1RIC/ IICIC2	UA1RIF/ IICIF2	UA1RMK/ IICMK2	0	0	0	UA1RPR2/ IICPR22	UA1RPR1/ IICPR21	UA1RPR0/ IICPR20
FFFFF168H	UA1TIC	UA1TIF	UA1TMK	0	0	0	UA1TPR2	UA1TPR1	UA1TPR0
FFFFF16AH	UA2RIC/ IICIC0	UA2RIF/ IICIF0	UA2RMK/ IICMK0	0	0	0	UA2RPR2/ IICPR02	UA2RPR1/ IICPR01	UA2RPR0/ IICPR00
FFFFF16CH	UA2TIC	UA2TIF	UA2TMK	0	0	0	UA2TPR2	UA2TPR1	UA2TPR0
FFFFF16EH	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF170H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF172H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF174H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF176H	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF178H	KRIC	KRIF	KRMK	0	0	0	KRPR2	KRPR1	KRPR0
FFFFF17AH	WTIC	WTIF	WTMK	0	0	0	WTIPR2	WTIPR1	WTIPR0
FFFFF17CH	WTIC	WTIF	WTMK	0	0	0	WTPR2	WTPR1	WTPR0

19.3.5 中断屏蔽寄存器 0 到 3 (IMR0 到 IMR3)

IMR0 到 IMR3 寄存器为可屏蔽中断设定中断屏蔽状态。IMR0 到 IMR3 寄存器的 xxMKn 位相当于 xxICn.xxMKn 位。

IMRm 寄存器可以以 16 位为单元读取或是写入(m = 0 到 3)

如果 IMRm 寄存器的高 8 位用作 IMRmH 寄存器，低 8 位用作 IMRmL 寄存器，这些寄存器可以以 8 位或是 1 位为单元进行读取或是写入操作(m = 0 到 3)。

复位后这些寄存器的值为 FFFFH。

注意事项 设备文件定义了 xxICn.xxMKn 位为保留字。如果有其它的位使用 xxMKn 作为名称进行操作，xxICn 寄存器的内容会被写入，而不是 IMRm 寄存器(结果，IMRm 寄存器的内容也会被覆盖)。

复位后的值: FFFFH 读/写 地址: IMR3 FFFFF106H,
IMR3L FFFFF106H, IMR3H FFFFF107H

	15	14	13	12	11	10	9	8
IMR3 (IMR3H)	1	1	1	1	1	TRXMK0/ IEMK2	RECMK0/ IEMK1	WUPMK0/ STMK
	7	6	5	4	3	2	1	0
IMR3L	ERRMK0/ ERRMK	WTMK	WTIMK	KRMK	DMAMK3	DMAMK2	DMAMK1	DMAMK0

复位后的值: FFFFH 读/写 地址: IMR2 FFFFF104H,
IMR2L FFFFF104H, IMR2H FFFFF105H

	15	14	13	12	11	10	9	8
IMR2 (IMR2H)	ADMK	UA2TMK	UA2RMK/ IICMK0	UA1TMK	UA1RMK/ IIC2MK	UA0TMK/ CB4TMK	UA0RMK/ CB4RMK	CB3TMK
	7	6	5	4	3	2	1	0
IMR2L	CB3RMK	CB2TMK	CB2RMK	CB1TMK	CB1RMK	CB0TMK	CB0RMK/ IICMK1	TM0EQMK0

复位后的值: FFFFH 读/写 地址: IMR1 FFFFF102H,
IMR1L FFFFF102H, IMR1H FFFFF103H

	15	14	13	12	11	10	9	8
IMR1 (IMR1H)	TP5CCMK1	TP5CCMK0	TP5QMK	TP4CCMK1	TP4CCMK0	TP4QMK	TP3CCMK1	TP3CCMK0
	7	6	5	4	3	2	1	0
IMR1L	TP3QMK	TP2CCMK1	TP2CCMK0	TP2QMK	TP1CCMK1	TP1CCMK0	TP1QMK	TP0CCMK1

复位后的值: FFFFH 读/写 地址: IMR0 FFFFF100H,
IMR0L FFFFF100H, IMR0H FFFFF101H

	15	14	13	12	11	10	9	8
IMR0 (IMR0H)	TP0CCMK0	TP0QMK	TQ0CCMK3	TQ0CCMK2	TQ0CCMK1	TQ0CCMK0	TQ0QMK	PMK7
	7	6	5	4	3	2	1	0
IMR0L	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

xxMKn	中断屏蔽标志的设定
0	允许中断服务
1	禁止中断服务

注 为了以 8 位或 1 位为单元读取 IMR0 到 IMR3 寄存器的位 8 到位 15，需要将它们设定为 IMR0H 到 IMR3H 寄存器的位 0 到位 7。

注意事项 将 IMR3 寄存器的位 7 到位 15 置 1。如果这些位的设定值改变，操作将不会得到保证。

备注 xx: 表示任意的外围单元的名称(参考表 19-2 中断控制寄存器(xxICn))
n: 外围单元编号(参考表 19-2 中断控制寄存器(xxICn))。

19.3.6 正在进行服务优先级寄存器(ISPR)

ISPR 寄存器当前被确认的可屏蔽中断的优先级别。当一个中断请求被确认时，寄存器中对应该中断请求信号优先级别的位置 1，且在中断服务程序执行期间保持。

当执行 RETI 指令后，对应具有最高优先级的中断请求信号的位通过硬件自动复位为 0。但是，当从非屏蔽中断服务或是异常处理中返回时，该位不会复位为 0。

这是一个只读寄存器，可以以 8 位或是 1 位为单元进行访问。

复位后该寄存器的值为 00H。

注意事项 如果在中断允许状态(EI)对 ISPR 寄存器进行读取时确认一个中断，在寄存器的位通过确认中断的方式被设置之后，可以读取 ISPR 寄存器的值。为了在确认中断之前正确的读取 ISPR 寄存器的值，需要禁止中断(DI)。

复位后的值:00H 只读 地址: FFFFF16H

ISPR

<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	拥有当前优先级的中断是否被确认
0	未确认带有优先级n的中断请求信号
1	已确认带有优先级n的中断请求信号

备注 n = 0 到 7 (优先级别)

19.3.7 ID 标志

这个标志控制可屏蔽中断的操作状态，并且存储有关允许或是禁止中断请求信号的控制信息。中断禁止标志(ID)位于 PSW 之中。

复位后该标志的值为 00000020H。

复位后的值: 00000020H

	31	8	7	6	5	4	3	2	1	0
PSW	0		NP	EP	ID	SAT	CY	OV	S	Z

ID	可屏蔽中断服务 ^注 的说明
0	允许确认可屏蔽中断请求信号
1	禁止确认可屏蔽中断请求信号(保持)

注 中断禁止标志(ID)功能

当执行 **DI** 指令时将这一位置 1，当执行 **EI** 指令时将这一位清 0。当参考 **PSW** 时，也可以通过 **RETI** 指令或是 **LDSR** 指令修改它的值。

非屏蔽中断请求信号和异常的确认不会考虑这个标志的设定。当一个可屏蔽中断请求信号被确认后，ID 标志位会通过硬件自动置 1。

在确认被禁止的时期(ID flag = 1)产生的中断请求信号会在xxICn.xxIFn 位置 1, ID 标志清 0 时被确认。

19.3.8 看门狗定时器模式寄存器 2(WDTM2)

该寄存器可以以 8 位为单元进行读取或是写入(详情参考第 11 章看门狗定时器 2 的功能)。

复位后该寄存器的值为 67H。

复位后的值: 67H 读/写 地址: FFFFFFF6D0H

	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的选择
0	0	停止运行
0	1	非屏蔽中断请求模式
1	×	复位模式(初始值)

19.4 软件异常

当 CPU 执行 TRAP 指令的时候会产生一个软件的异常，且总是会被确认。

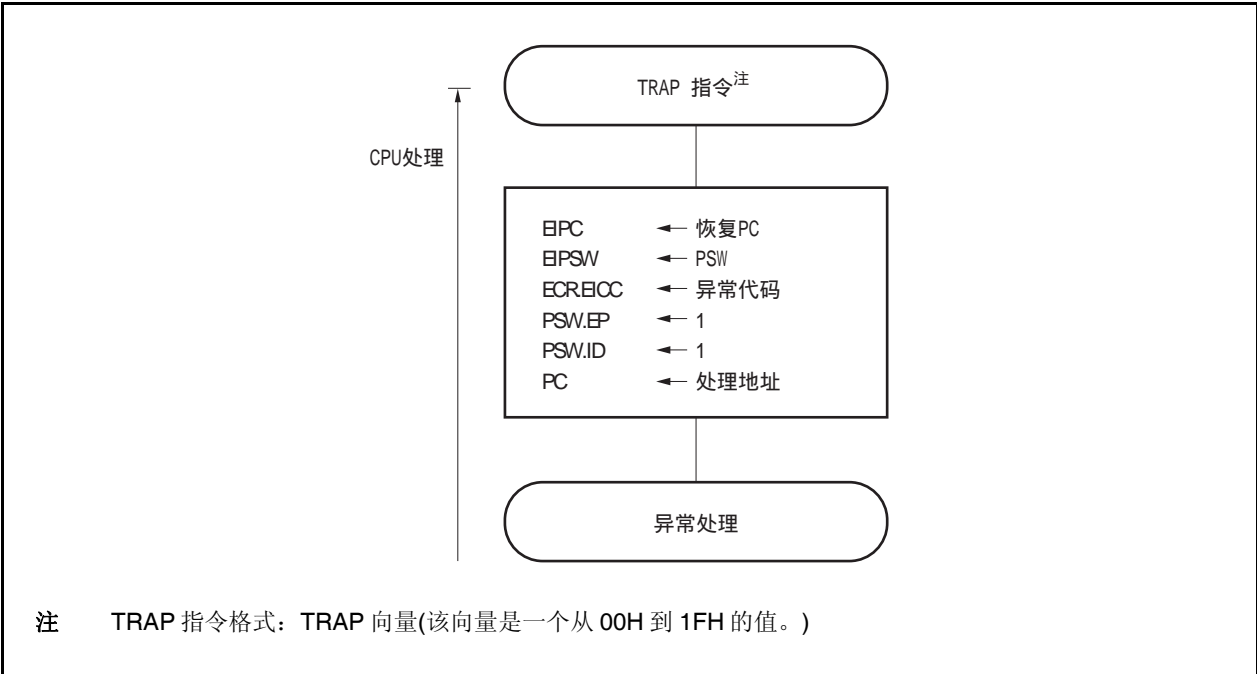
19.4.1 操作

如果一个软件异常产生，CPU 将执行下列处理过程，并且将操作权转移给处理例程。

- <1> 将恢复的 PC 保存在 EIPC 中。
- <2> 将当前的 PSW 保存在 EIPSW 中。
- <3> 在 ECR (中断源)的低 16 位(EICC)中写入一个异常代码。
- <4> 将 PSW.EP 位和 PSW.ID 位置 1。
- <5> 在 PC 中设置附和软件异常的处理地址(00000040H or 00000050H)，并且转移控制权。

图 19-9 举例说明软件异常的处理过程。

图 19-9. 软件异常处理过程



处理地址是通过 TRAP 指令的操作数(向量)决定的。如果该向量是 00H 到 0FH，它将变为 00000040H；而如果向量的值是 10H 到 1FH，它将变为 00000050H。

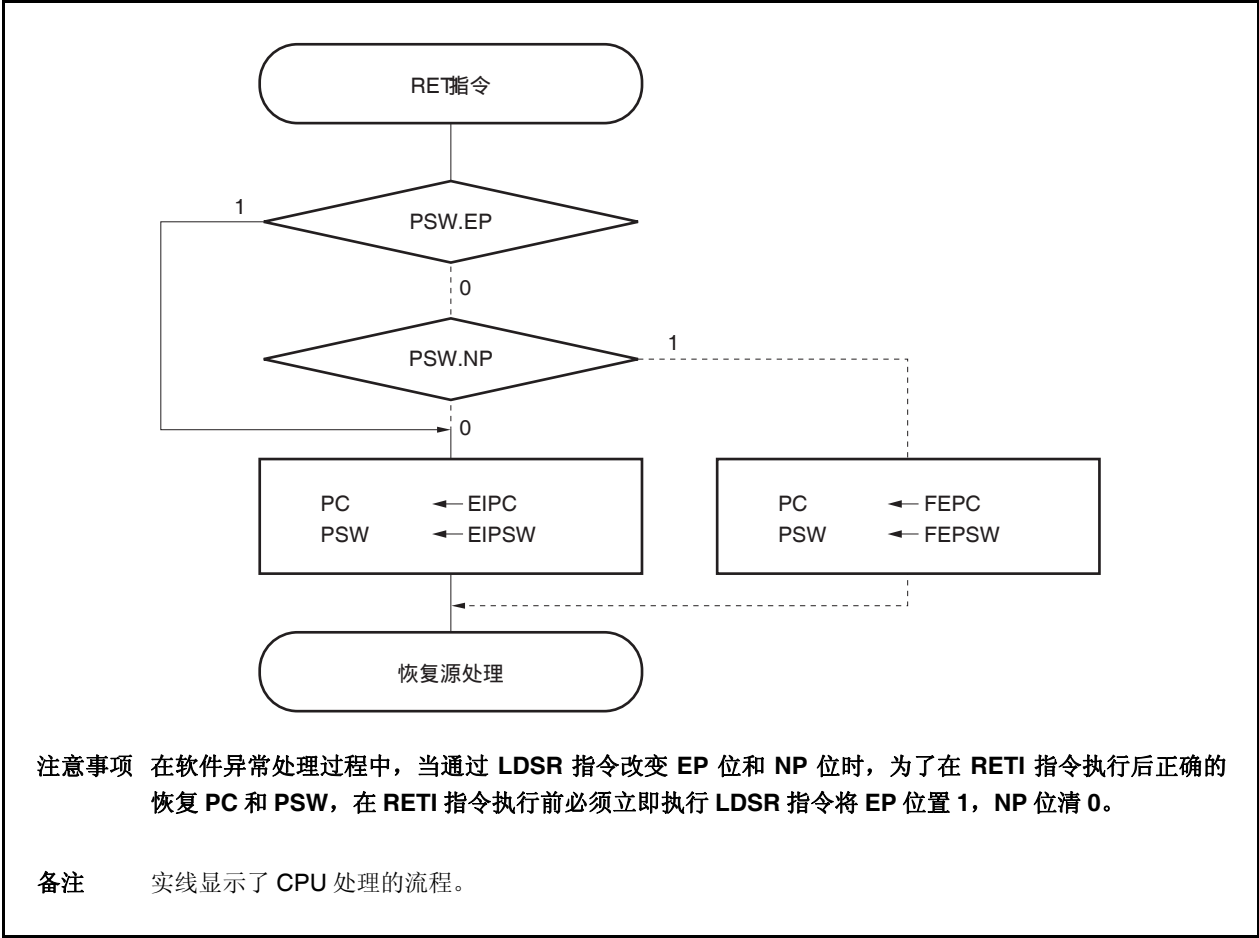
19.4.2 恢复

通过执行 RETI 指令可以从软件异常进程中恢复。
通过执行 RETI 指令，CPU 执行下列处理过程，并且将控制权转移到恢复 PC 的地址。

- <1> 由于 PSW.EP 位为 1，从 EIPC 和 EIPSW 中加载恢复的 PC 和 PSW。
- <2> 将控制权转移到恢复 PC 和 PSW 的地址。

下图举例说明了 RETI 指令的处理过程。

图 19-10. RETI 指令处理过程



注意事项 在软件异常处理过程中，当通过 LDSR 指令改变 EP 位和 NP 位时，为了在 RETI 指令执行后正确的恢复 PC 和 PSW，在 RETI 指令执行前必须立即执行 LDSR 指令将 EP 位置 1，NP 位清 0。

备注 实线显示了 CPU 处理的流程。

19.4.3 EP 标志

EP 标志用于指出异常进程正在进行中。在异常发生时，该位将被设定。

复位后的值：00000020H

PSW

31876543210

0NPEPIDSATCYOVSSZ

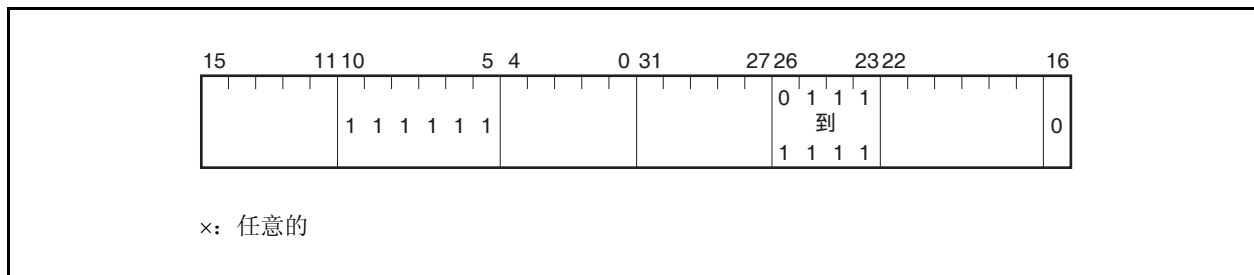
EP	异常处理状态
0	未进行异常处理
1	正在进行异常处理.

19.5 异常陷阱

异常陷阱是一个中断，在指令的非法操作发生时需要使用。在 V850ES/JG2，一个非法的操作代码异常(ILGOP: 非法的操作代码陷阱)被认为是一个异常陷阱。

19.5.1 非法的操作代码定义

非法指令位 10 到位 5 的操作代码是 111111B, 位 26 到位 23 的子操作代码是 0111B 到 1111B, 位 16 的子操作代码是 0B。当指令应用到这个非法指令执行后, 产生一个异常陷阱。



注意事项 由于将来有可能分配该指令到一个非法的操作代码，推荐不要使用它。

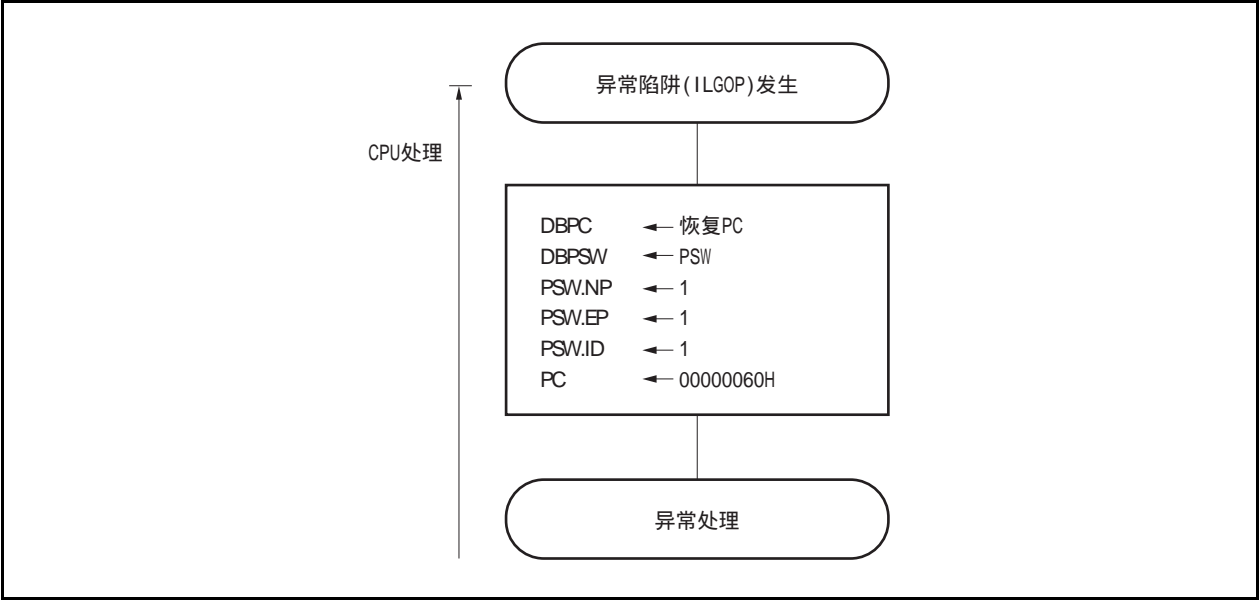
(1) 操作

如果产生一个异常陷阱，CPU 执行下列处理过程，并且将操作权转移给处理例程。

- <1> 将恢复的 PC 保存到 DBPC。
- <2> 将当前的 PSW 保存到 DBPSW。
- <3> 将 PSW.NP 位, PSW.EP 位, 和 PSW.ID 位置 1。
- <4> 在 PC 中设置相应异常陷阱的处理地址(00000060H), 并且转移控制权。

下图举例说明异常陷阱的处理过程。

图 19-11.异常陷阱的处理过程



(2) 恢复

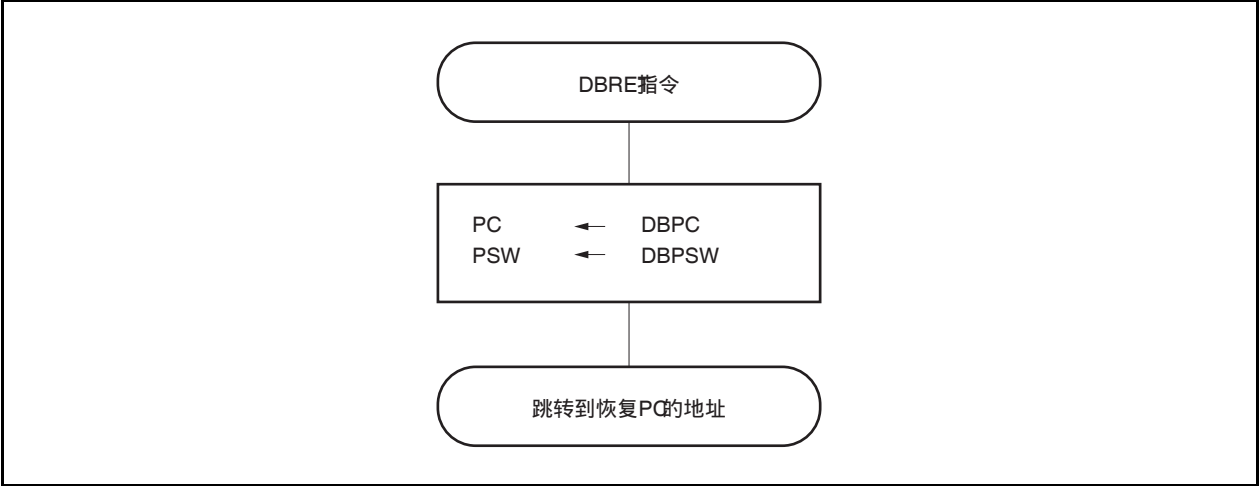
通过执行 DBRET 指令实现从异常陷阱中恢复。通过执行 DBRET 指令，CPU 执行下列处理过程并且控制着恢复 PC 的地址。

- <1> 从 DBPC 和 DBPSW 中加载恢复的 PC 和 PSW。
- <2> 向恢复的 PC 和 PSW 代表的地址转移控制权。

注意事项 仅在执行非法操作代码和 DBRET 指令的间隔内才能够访问 DBPC 和 DBPSW

图 19-12 举例说明从异常陷阱中恢复的处理过程。

图 19-12. 从异常陷阱中恢复的处理过程



19.5.2 调试陷阱

调试陷阱是一个异常，它在 DBTRAP 指令执行时产生，且总是会被确认。

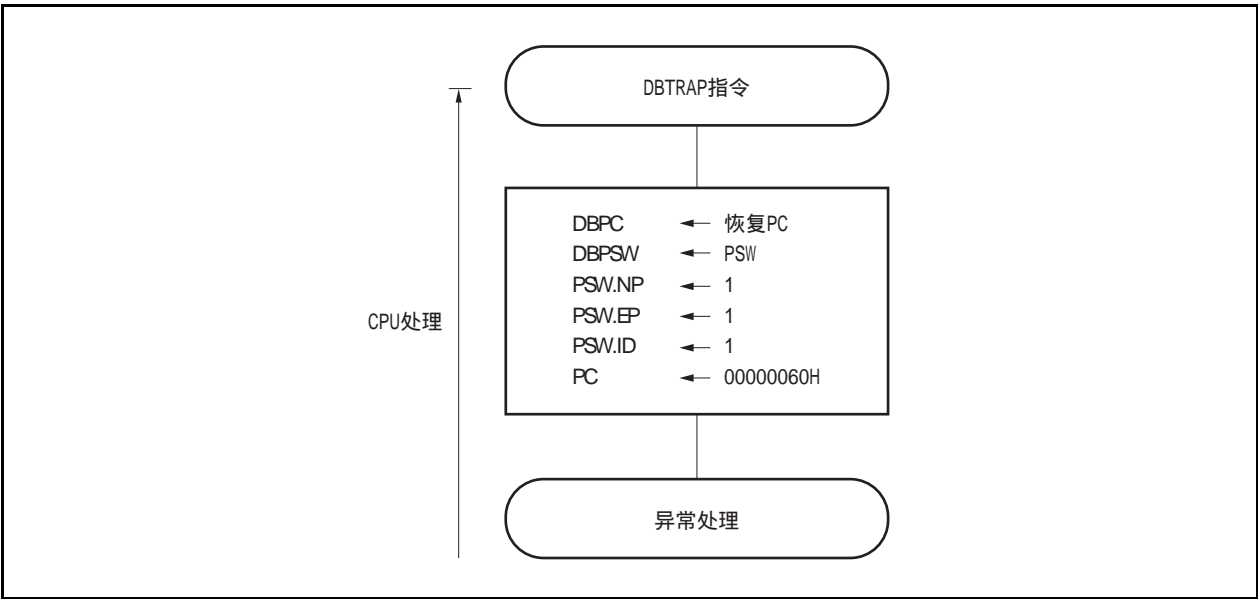
(1) 操作

根据不同的调试陷阱，CPU 执行如下进程。

- <1> 将恢复的 PC 保存到 DBPC。
- <2> 将当前的 PSW 保存到 DBPSW。
- <3> 将 PSW.NP 位，PSW.EP 位，和 PSW.ID 位置 1。
- <4> 在 PC 中设置相应调试陷阱的处理地址(00000060H)，并且转移控制权。

下图显示了调试陷阱的处理格式。

图 19-13. 调试陷阱的处理格式



(2) 恢复

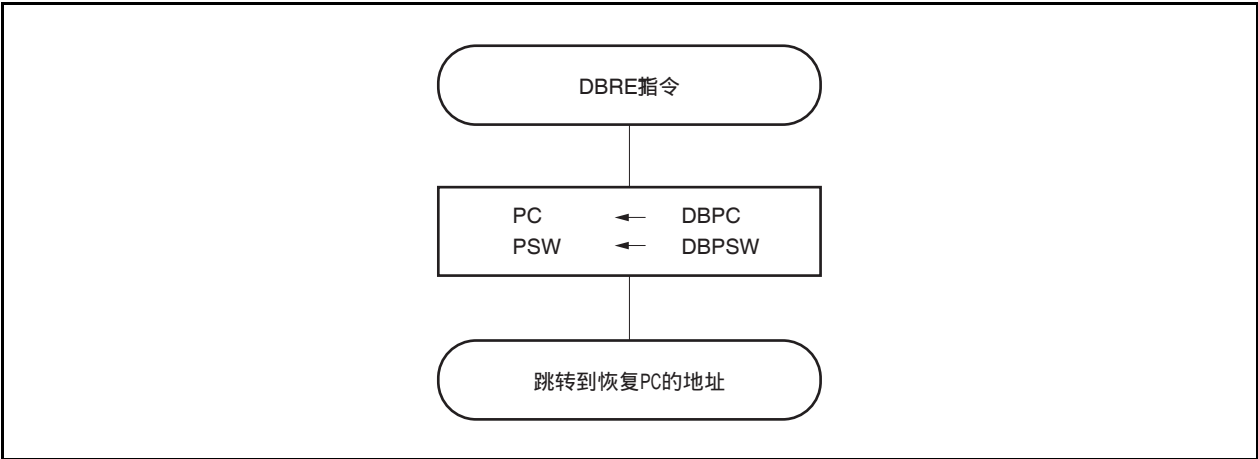
通过执行 DBRET 指令可以从调试陷阱中恢复。
执行 DBRET 指令，CPU 将执行以下步骤，并将控制权转移到恢复 PC 的地址。

- <1> 从 DBPC 和 DBPSW 中加载恢复的 PC 和 PSW。
- <2> 向恢复的 PC 和 PSW 代表的地址转移控制权。

注意事项 在 DBTRAP 指令执行后 DBRET 指令执行之前，可以访问 DBPC 和 DBPSW。

下表显示了从调试陷阱中恢复的处理格式。

图 19-14. 从调试陷阱中恢复的处理格式



19.6 外部中断请求输入引脚(NMI 和 INTP0 到 INTP7)

19.6.1 噪声消除

(1) NMI 引脚的噪声消除

NMI 引脚拥有内部噪声消除电路，它使用模拟延迟。因此，NMI 引脚的输入电平不会作为边沿被检测，除非保持一定的时间或者更长。所以只有在特定的时间才会检测边沿。

NMI 引脚可以用来释放 STOP 模式。在 STOP 模式中，使用系统时钟不会消除噪声，因为内部系统时钟是停止的。

(2) INTP0 到 INTP7 引脚的噪声消除

INTP0 到 INTP7 引脚拥有内部噪声消除电路，它使用模拟延迟。因此，这些引脚的输入电平不会作为边沿被检测，除非保持一定的时间或者更长。所以只有在特定的时间才会检测边沿。

19.6.2 边沿检测

每一个 NMI 和 INTP0 到 INTP7 引脚的有效边沿可以在以下四种方式中选择。

- 上升沿
- 下降沿
- 双边沿
- 不检测边沿

在复位后，NMI 引脚不会检测边沿。因此，中断请求信号不会被确认，除非使用 INTF0 和 INTR0 寄存器允许有效边沿(NMI 引脚功能作为一个端口引脚)。

(1) 外部中断下降，上升沿指定寄存器 0(INTF0, INTR0)

INTF0 和 INTR0 寄存器是 8 位寄存器，它们通过位 2 指定了 NMI 引脚的上升沿和下降沿的检测，通过位 3 到位 6 指定了外部中断引脚(INTP0 to INTP3)的上升沿和下降沿的检测。
这些寄存器可以以 8 位或是 1 位为单元进行读取或是写入。
复位后这些寄存器的值为 00H。

注意事项 当引脚的功能从外部中断功能(被用功能)转变到端口功能时，可以进行边沿检测。因此，首先要将 INTF0n 位和 INTR0n 位设定为 00，然后再设定端口模式。

复位后的值: 00H 读/写 地址: INTF0 FFFFC00H, INTR0 FFFFC20H

INTF0

7	6	5	4	3	2	1	0
0	INTF06	INTF05	INTF04	INTF03	INTF02	0	0
	INTP3	INTP2	INTP1	INTP0	NMI		

INTR0

7	6	5	4	3	2	1	0
0	INTR06	INTR05	INTR04	INTR03	INTR02	0	0
	INTP3	INTP2	INTP1	INTP0	NMI		

备注

如果设定有效边沿参考表 19-3。

表 19-3. 有效边沿设定

INTF0n	INTR0n	有效边沿设定(n = 2 到 6)
0	0	不检测边沿
0	1	上升沿
1	0	下降沿
1	1	双边沿

注意事项 当这些寄存器不被用作 NMI 或 INTP0 到 INTP3 时，一定要将 INTF0n 和 INTR0n 位设定为 00。

备注 n=2: 控制 NMI 引脚
 n=3 到 6: 控制 INTP0 到 INTP3 引脚

(2) 外部中断下降，上升沿指定寄存器 3 (INTF3, INTR3)

INTF3 和 INTR3 寄存器是 8 位寄存器，它指定了外部中断引脚(INTP7)的上升沿和下降沿的检测。
这些寄存器可以以 8 位或是 1 位为单元进行读取或是写入。
复位后这些寄存器的值为 00H。

- 注意事项 1. 当引脚的功能从外部中断功能(被用功能)转变到端口功能时，可以进行边沿检测。因此，首先要将 INTF0n 位和 INTR0n 位设定为 00，然后再设定端口模式。
2. INTP7 引脚和 RXDA0 引脚是复用引脚。当使用 RXDA0 pin 引脚功能时，要禁止 INTP7 复用功能引脚的边沿检测功能(将 INTF3.INTF31 位和 INTR3.INTR31 位清 0)。当使用 INTP7 引脚功能时，要停止 UARTA0 接收(将 UA0CTL0.UA0RXE 位清 0)。

复位后的值: 00H 读/写 地址: INTF3 FFFFFFFC06H, INTR3 FFFFFFFC26H

	7	6	5	4	3	2	1	0
INTF3	0	0	0	0	0	0	INTF31	0
							INTP7	
	7	6	5	4	3	2	1	0
INTR3	0	0	0	0	0	0	INTR31	0
							INTP7	

备注 如何设定有效边沿参考表 19-4。

表 19-4. 有效边沿设定

INTF31	INTR31	有效边沿设定
0	0	不检测边沿
0	1	上升沿
1	0	下降沿
1	1	双边沿

注意事项 当这些寄存器不被用作 NMI 或 INTP7 时，一定要将 INTF31 和 INTR31 位设定为 00。

(3) 外部中断下降，上升沿指定寄存器 9H (INTF9H, INTR9H)

INTF9H 和 INTR9H 寄存器是 8 位寄存器，它们指定了外部中断引脚(INTP4 到 INTP6)的上升沿和下降沿的检测。

这些寄存器可以以 8 位或是 1 位为单元进行读取或是写入。

复位后这些寄存器的值为 00H。

注意事项 当引脚的功能从外部中断功能(被用功能)转变到端口功能时，可以进行边沿检测。因此，首先要将 INTF9n 位和 INTR9n 位设定为 00，然后再设定端口模式。

复位后的值: 00H 读/写 地址: INTF9H FFFFC13H, INTR9H FFFFC33H

INTF9H

15	14	13	12	11	10	9	8
INTF915	INTF914	INTF913	0	0	0	0	0
INTP6	INTP5	INTP4					

INTR9H

15	14	13	12	11	10	9	8
INTR915	INTR914	INTR913	0	0	0	0	0
INTP6	INTP5	INTP4					

备注

如何设定有效边沿参考表 19-5。

表 19-5. 有效边沿设定

INTF9n	INTR9n	有效边沿设定(n = 13 到 15)
0	0	不检测边沿
0	1	上升沿
1	0	下降沿
1	1	双边沿

注意事项 当这些寄存器不被用作 INTP4 或 INTP6 时，一定要将 INTF9n 和 INTR9n 位设定为 00。

备注 n = 13 to 15: Control of INTP4 to INTP6 pins

(4) 噪声消除控制寄存器(NFC)

可以为 INTP3 引脚选择数字噪声消除。噪声消除设定通过使用 NFC 寄存器执行。

当选择了数字噪声消除，数字采样的时钟可以从 f_{xx}/64, f_{xx}/128, f_{xx}/256, f_{xx}/512, f_{xx}/1,024, and f_{XT} 中选择。采样频率是时钟的三倍。

即使选择了数字噪声消除，应用 f_{XT} 作为采样时钟使得通过 INTP3 中断请求信号释放 IDLE1, IDLE2, 和 STOP 模式成为可能。

这些寄存器可以以 8 位为单元进行读取或是写入。

复位后这些寄存器的值为 00H。

注意事项 在改变采样时钟之后，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的 3 个时钟之后 INTP3 输入有效边沿，将产生中断请求。因此，当使用中断和 DMA 功能的时候，一定要注意以下几点。

- 当使用中断功能时，在经过 3 个采样时钟之后，在中断请求标志(PIC3.PIF3 位)被清除后，要允许中断。
- 当使用 DMA 功能时(由 INTP3 始发)，在经过 3 个采样时钟之后允许 DMA。

复位后的值: 00H 读/写 地址: FFFFF318H

	7	6	5	4	3	2	1	0
NFC	NFEN	0	0	0	0	NFC2	NFC1	NFC0

NFEN	INTP3 引脚的噪声消除设定
0	模拟噪声消除(60 ns 典型值))
1	数字噪声消除

NFC2	NFC1	NFC0	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			禁止设定

备注

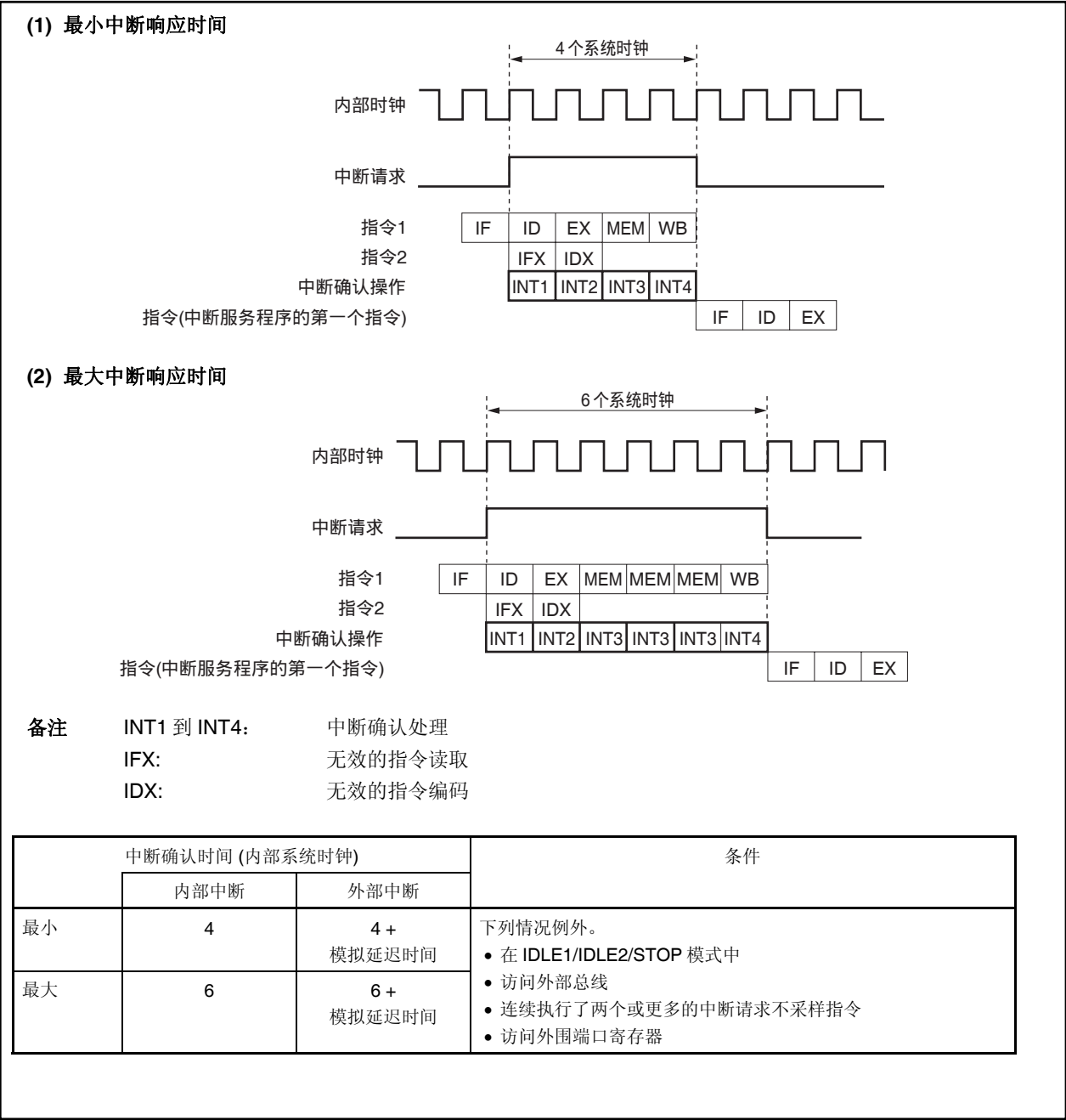
1. 当采样执行 3 次之后，可靠的噪声消除宽度是 2 个采样时钟。
2. 在噪声的宽度小于 2 个采样时钟的情况下，如果有与采样时钟同步的噪声输入将会产生中断请求信号。

19.7 CPU 的中断确认时间

除非出现下列情况，CPU 的中断确认时间最少需要 4 个时钟周期。为了成功的输入中断请求信号，中断输入的间隔最少应该间隔 5 个时钟。

- 在 IDLE1/IDLE2/STOP 模式中
- 当访问外部总线时
- 当成功的执行了中断请求不采样指令(参考 19.8 CPU 不对中断进行确认的时期。)
- 当访问中断控制寄存器时

图 19-15. 在中断请求信号确认过程中的传递操作(概要)



19.8 CPU 不对中断进行确认的时期

当执行指令时，中断会通过 CPU 确认。但是，在两个中断请求不采样指令之间(中断被保持)。中断不采样信号指令如下。

- EI 指令
- DI 指令
- PSW 的 LDSR reg2, 0x5 指令
- PRCMD 寄存器的存储指令
- 以下寄存器的存储，SET1，NOT1，或是 CLR1 指令。
 - 有关中断的寄存器：
中断控制寄存器(xxICn)， 中断屏蔽寄存器 0 到 3(IMR0 到 IMR3)
 - 节能控制寄存器(PSC)
 - 片上调试模式寄存器(OCDM)

备注 **xx:** 表示外围单元的名字(参考表 19-2 中断控制寄存器(xxICn))
 n: 外围单元编号(参考表 19-2 中断控制寄存器(xxICn))。

19.9 注意事项

NMI 引脚和 P02 引脚是功能复用引脚，复位后的功能是一个正常的端口。为了执行 NMI 的功能，在 PMCO 寄存器中确认 NMI 引脚。NMI 引脚的初始设定是不检测边沿。使用 INTF0 寄存器和 INTRO 寄存器设定 NMI 引脚的有效边沿。

第二十章 按键中断功能

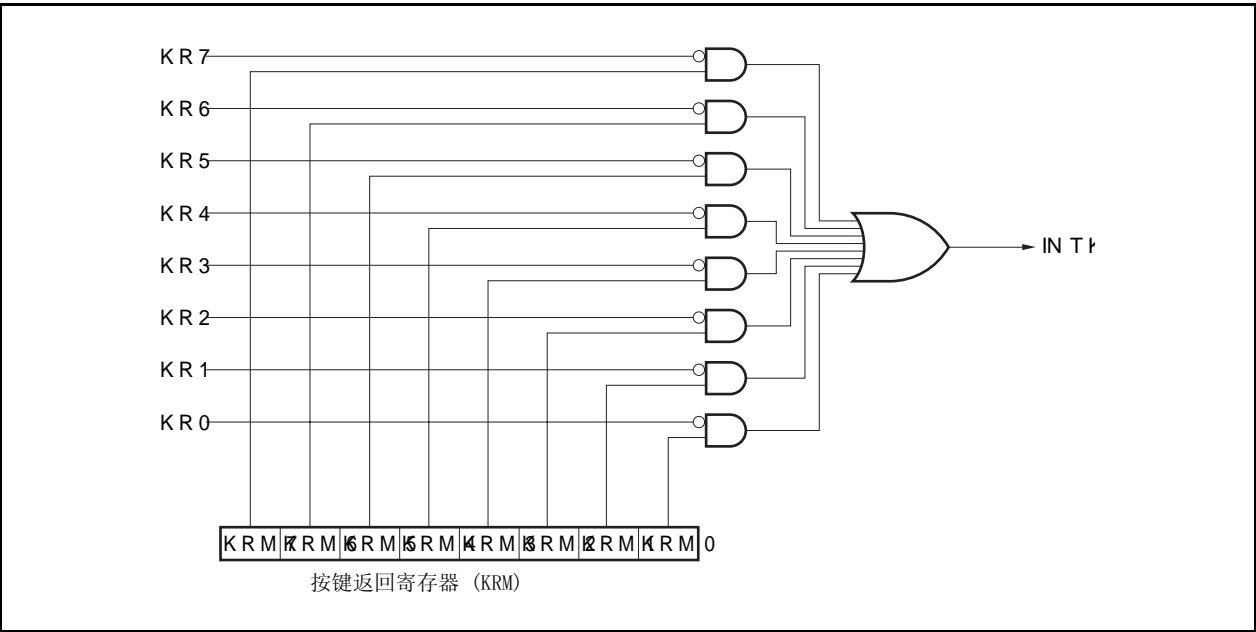
20.1 功能

通过设置 KRM 寄存器，给八个按键输入引脚（KR0 ~ KR7）输入下降沿，可以产生按键中断请求信号（INTKR）。

表 20-1. 按键返回检测引脚的任务

标记	引脚描述
KRM0	位选控制 KR0 信号
KRM1	位选控制 KR1 信号
KRM2	位选控制 KR2 信号
KRM3	位选控制 KR3 信号
KRM4	位选控制 KR4 信号
KRM5	位选控制 KR5 信号
KRM6	位选控制 KR6 信号
KRM7	位选控制 KR7 信号

图 20-1. 按键返回框图



20.2 寄存器

(1) 按键返回模式寄存器（KRM）

KRM 寄存器使用 KR0 ~ KR7 信号控制 KRM0 ~ KRM7 位。
该寄存器可用 8 位或 1 位指令进行读或写。
复位输入将该寄存器清零（00H）。

After reset: 00H		R/W	Address: FFFFF300H					
KRM	7	6	5	4	3	2	1	0
	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0
KRMn		Control of key return mode						
0		Does not detect key return signal						
1		Detects key return signal						

注意事项 在 KRM 寄存器被清零（00H）后，重写 KRM 寄存器。

备注 对于复用功能引脚设置，参见表 4-15 使用端口引脚作为复用功能引脚。

20.3 注意事项

- (1) 如果低电平被输入给 KR0 ~ KR7 中的任何一个，即使其它引脚的下降沿被输入，也不会产生 INTKR 信号。
- (2) 不能同时使用 RXDA1 和 KR7 引脚。为了使用 RXDA1 引脚，就不要使用 KR7 引脚。为了使用 KR7 引脚，就不要使用 RXDA1 引脚（推荐设置 PFC91 位为 1 并且将 PFCE91 位清零）。
- (3) 如果改变 KRM 寄存器，则会产生中断请求信号（INTKR）。为避免这种情况，在禁止中断（DI）或屏蔽中断之后改变 KRM 寄存器，然后将中断请求标志（KRIC、KRIF 位）清零，再允许中断或清除屏蔽。
- (4) 为了使用按键中断功能，确信设置端口引脚为按键返回引脚，然后使用 KRM 寄存器允许操作。为了从按键返回引脚切换到端口引脚，使用 KRM 寄存器禁止操作，然后设置为端口引脚。

第二十一章 待机功能

21.1 概述

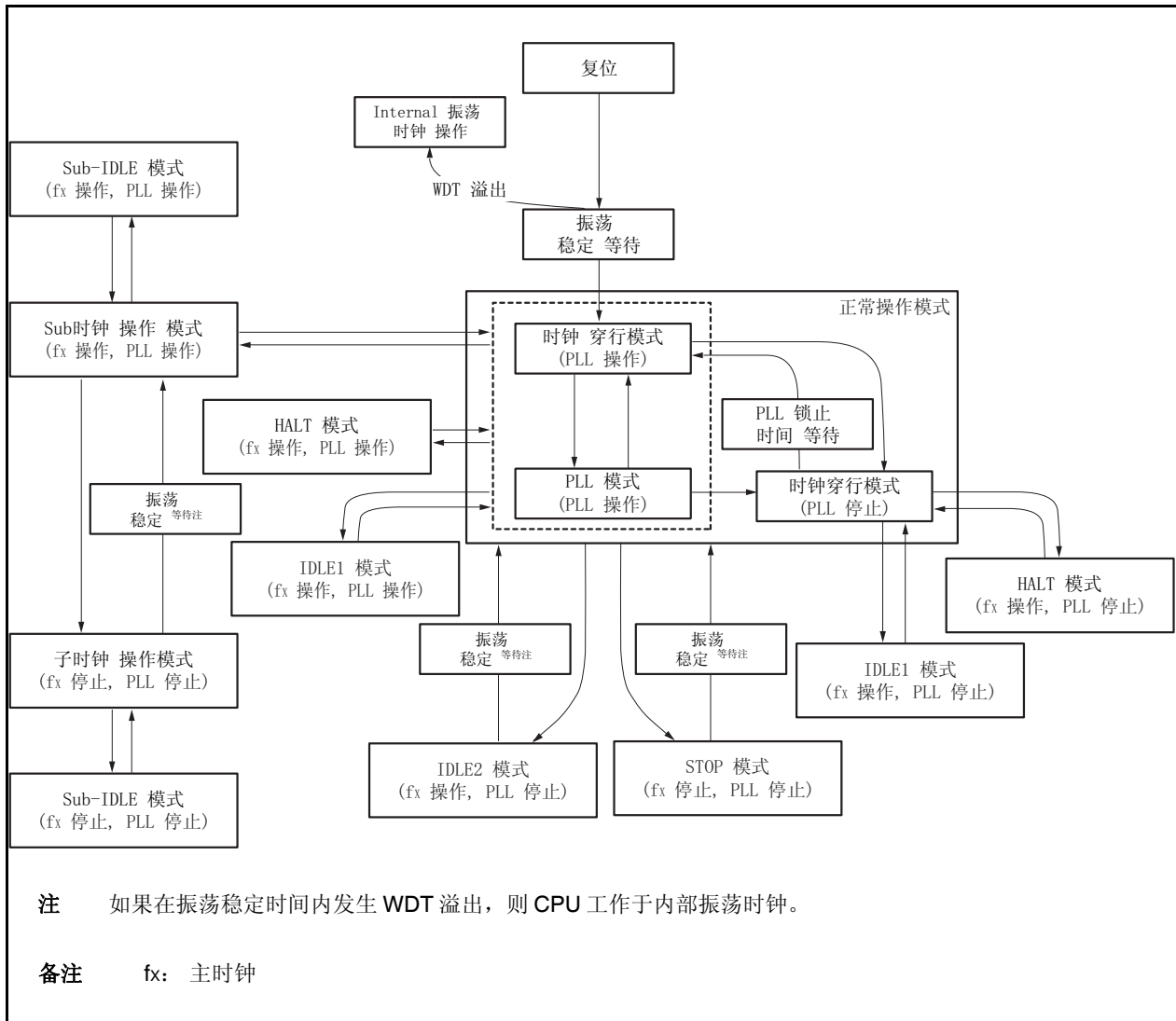
通过结合使用待机模式并选择合适的应用模式，可以有效的降低系统功耗。可用到的待机模式在表 21-1 中列出。

表 21-1. 待机模式

模式	功能概要
HALT 模式	此模式仅停止 CPU 的操作时钟
IDLE1 模式	此模式停止除振荡器之外的全部内部电路，PLL ^注 ，和 flash 存储器的操作
IDLE2 模式	此模式停止除了振荡器之外的所有片内操作
STOP 模式	此模式停止除了副时钟振荡器之外的所有片内操作
副时钟操作模式	此模式副时钟被用作内部系统时钟
副 IDLE 模式	此模式停止除了振荡器之外的全部片内操作，工作在副时钟操作模式下

注 PLL 保持预先操作状态。

图 21-1. 状态转换



21.2 寄存器

(1) 功率节省控制寄存器（PSC）

PSC 寄存器是控制待机功能的 8 位寄存器。此寄存器的 STP 位被用于指定 STOP 模式。此寄存器是仅能使用特殊顺序组合写入的特殊寄存器（参见 3.4.7 特殊寄存器）。

该寄存器可用 8 位或 1 位指令进行读或写。

复位输入将该寄存器清零（00H）。

复位后: 00H R/W 地址: FFFFF1FEH

	7	< 6 >		< 5 >	< 4 >	3	2	< 1 >	0
PSC	0	NMI1M	NMI0M	INTM	0	0	STP	0	

NMI1M	由INTWDT2 信号的发生来释放等待模式
0	允许由INTWDT2 信号释放等待模式
1	禁止由INTWDT2 信号释放等待模式

NMI0M	由NMI引进控制的释放等待模式
0	允许由NMI引脚输入释放等待模式
1	禁止由NMI引脚输入释放等待模式

INTM	等待模式 release control via maskable interrupt request signal
0	允许可屏蔽中断释放等待模式
1	禁止可屏蔽中断释放等待模式

STP	等待 ^注 设置
0	正常模式
1	等待模式

注 待机模式由 STP 位设置: IDLE1, IDLE2, STOP, 或 sub-IDLE 模式

- 注意事项
1. 在设置 IDLE1, IDLE2, STOP 或副 IDLE 模式之前, 先设置 PSMR.PSM1 和 PSMR.PSM0 位, 然后设置 STP 位。
 2. 当释放 HALT 模式时对 NMI1M, NMI0M 和 INTM 位进行设置是非法的。
 3. 如果在对 NMI1M, NMI0M, 或 INTM 位置 1 相同的时刻对 STP 位也置 1, 则对, NMI0M, or INTM 这些位的设置无效。当设置了 IDLE1/IDLE2/STOP 模式后, 有一个挂起的不可屏蔽的中段请求, 则要将与中断请求信号相关的位(NMI1M, NMI0M, 或 INTM)置 1, 然后在将 STP 位置 1。

<R>

(2) 功率节省模式寄存器 (PSMR)

PSMR 寄存器是一个 8 位寄存器，其控制省电模式中的操作状态和时钟操作。
该寄存器可用 8 位或 1 位指令进行读或写。
复位输入将该寄存器清零 (00H)。

复位后:	00H	R/W	地址:	FFFFF820H				
P S M R	7	6	5	4	3	2	<1>	<0>
	0	0	0	0	0	0	P S M 1	P S M 0

P S M 1	P S M 0	在软件等待模式下操作的定义
0	0	IDLE1, sub-IDLE 模式
0	1	STOP mode
1	0	IDLE2, sub-IDLE 模式
1	1	STOP 模式

- 注意事项 1. 确定将第 2 位 ~ 第 7 位清零。
2. 仅当 PSC.STP 位为 1 时，PSM0 和 PSM1 位是有效的。

备注 IDLE1: 在此模式中，除了振荡器之外的所有操作和一些其它的电路（flash 存储器和 PLL）都被停止。
在 IDLE1 模式被释放后，重新恢复为正常操作模式而无须确保振荡稳定时间，就像 HALT 模式一样。

IDLE2: 在此模式中，除了振荡器操作之外的所有操作都被停止。
在 IDLE2 模式被释放后，等待通过 OSTS 寄存器指定的设置时间后恢复为正常操作模式（flash 存储器和 PLL）。

STOP: 在此模式中，除了副时钟振荡器操作之外的所有操作都被停止。
在 STOP 模式被释放后，等待通过 OSTS 寄存器指定的振荡稳定时间后恢复为正常操作模式。

副 IDLE: 在此模式中，除了振荡器操作之外的所有其它操作都被暂停。在通过中断请求信号释放 IDLE 模式之后，在等待 12 个副时钟周期之后重新恢复副时钟操作模式。

(3) 振荡稳定时间选择寄存器（OSTS）

通过 OSTS 寄存器控制 STOP 模式释放后到振荡稳定的等待时间或 IDLE2 模式释放后到内置 flash 存储器稳定的等待时间。

OSTS 寄存器可用 8 位指令进行读或写。

复位输入设置该寄存器为 06H。

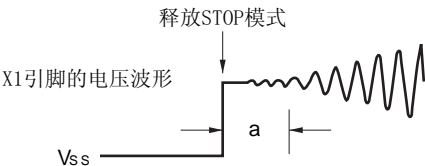
复位后: 06H R/W 地址: FFFF6C0H

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

OSTS2	OSTS1	OSTS0	振荡稳定时间/设置时间 ^注 的选择	fx	
				4 MHz	5 MHz
0	0	0	2 ¹⁰ /fx	0.256 ms	0.205 ms
0	0	1	2 ¹¹ /fx	0.512 ms	0.410 ms
0	1	0	2 ¹² /fx	1.024 ms	0.819 ms
0	1	1	2 ¹³ /fx	2.048 ms	1.638 ms
1	0	0	2 ¹⁴ /fx	4.096 ms	3.277 ms
1	0	1	2 ¹⁵ /fx	8.192 ms	6.554 ms
1	1	0	2 ¹⁶ /fx	16.38 ms	13.107 ms
1	1	1	禁止设置		

注 当 STOP 模式和 IDLE2 模式被释放后分别要求振荡稳定时间和设置时间。

注意事项 1. STOP 模式释放后的等待时间不包括从释放 STOP 模式到时钟振荡启动这段时间（即下图“a”所示的部分），不管通过复位输入或中断请求信号的产生是否释放了 STOP 模式。



- 2. 确定将第 3 位 ~ 第 7 位清零。
- 3. 复位释放后的振荡稳定时间是 2¹⁶/fx（因为 OSTS 寄存器的初始值= 06H）。

备注 fx = 主时钟振荡频率

21.3 HALT 模式

21.3.1 设置和操作状态

当在正常操作模式中执行专用指令（HALT）时，HALT 模式被设置。

在 HALT 模式中，时钟振荡器继续操作。仅仅停止对 CPU 的时钟供应，继续提供给其它内置外围设备功能的时钟供应。

结果，程序停止执行，并且内部 RAM 保持 HALT 模式设置之前的内容。独立于 CPU 指令处理的内置外围设备功能可继续操作。

表 21-3 显示了 HALT 模式中的操作状态。

通过使用 HALT 模式并结合正常操作模式的间歇操作可降低系统的平均电流消耗。

注意事项 1. 在 HALT 指令之后插入五个或更多的 NOP 指令。

2. 如果在执行 HALT 指令时有一个未屏蔽的中断请求被保持等待响应，则此状态被转移到 HALT 模式中，然后通过这个等待响应的中断请求，HALT 模式被立即释放。

21.3.2 释放 HALT 模式

可通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号），未屏蔽的外部中断请求信号（INTP0 ~ INTP7 引脚输入），来自于在 HALT 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，WDT2RES 信号，低电压检测器（LVI）或时钟监视器（CLM）产生的复位）来释放 HALT 模式。

在 HALT 模式释放后，恢复正常操作模式。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 HALT 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 HALT 模式，而不管中断请求信号的优先级。如果在某个中断服务程序中设置 HALT 模式，则之后产生的中断请求信号按如下所述来执行。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 HALT 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 HALT 模式，并且响应此中断请求信号。

表 21-2. 通过中断请求信号释放 HALT 模式后的操作

释放源	中断允许（EI）状态	中断禁止（DI）状态
不可屏蔽中断请求信号	执行对处理地址的跳转	
可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令	执行下一条指令

(2) 通过复位释放 HALT 模式

执行与正常复位操作相同的操作。

表 21-3. HALT 模式中的操作状态

项目 \ HALT 模式设置		操作状态	
		当不适应副时钟时	当使用副时钟时
主时钟振荡器		允许振荡	
副时钟振荡器		—	允许振荡
内部振荡器		允许振荡	
PLL		可操作	
CPU		停止操作	
DMA		可操作	
中断控制器		可操作	
定时器 P (TMP0 ~ TMP5)		可操作	
定时器 Q (TMQ0)		可操作	
定时器 M (TMM0)		当非 f_{XT} 时钟被选择作为计数时钟时可操作	可操作
钟表定时器		当 f_{XT} (BRG 分频) 时钟被选择作为计数时钟时可操作	可操作
看门狗定时器 2		当非 f_{XT} 时钟被选择作为计数时钟时可操作	可操作
串行接口	CSIB0 ~ CSIB1	可操作	
	I ² C00 ~ I ² C02	可操作	
	UARTA0 ~ UARTA2	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	
外部总线接口		参见 2.2 引脚状态	
端口功能		在 HALT 模式被设置之前保持状态	
内部数据		CPU 寄存器, 状态, 数据和全部其它的内部数据, 例如内部 RAM 的内容, 被保持为 HALT 模式设置之前的状态	

21.4 IDLE1 模式

21.4.1 设置和操作状态

通过在正常操作模式中对 PSMR.PSM1 和 PSMR.PSM0 位清零，并设置 PSC.STP 位为 1 设置 IDLE1 模式。

在 IDLE1 模式中，时钟振荡器，PLL 和 flash 存储器继续操作但停止给 CPU 和其它内置外围设备功能的时钟供应。

结果，停止程序执行，并且保持在设置 IDLE1 模式之前的内部 RAM 的内容。CPU 和其它内置外围设备功能停止操作。但是，使用副时钟或外部时钟的内置外围设备功能继续操作。

表 21-5 显示了在 IDLE1 模式中的操作状态。

因为 IDLE1 模式停止了内置外围设备功能的操作，所以 IDLE1 模式比 HALT 模式更能降低功耗。因为不停止主时钟振荡器，所以当用与释放 HALT 模式相同的方法释放 IDLE1 模式时，无须等待振荡稳定时间就能恢复到正常操作模式。

注意事项 1. 在执行往 PSC 寄存器中存入数据来设置 IDLE1 模式的指令之后，插入五个或更多的 NOP 指令。

★ 2. 如果设置 IDLE1 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则 IDLE1 模式立即被此等待响应的中断请求释放。

21.4.2 释放 IDLE1 模式

可通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号），未屏蔽的外部中断请求信号（INTP0 ~ INTP7 引脚输入），来自于在 IDLE1 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，WDT2RES 信号，低电压检测器（LVI）或时钟监视器（CLM）产生的复位）来释放 IDLE1 模式。

在 IDLE1 模式释放后，恢复正常操作模式。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE1 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE1 模式，而不管中断请求信号的优先级。如果在某个中断服务程序中设置 IDLE1 模式，则之后产生的中断请求信号按如下所述来执行。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 IDLE1 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 IDLE1 模式，并且响应此中断请求信号。

表 21-1. 通过中断请求信号释放 IDLE1 模式后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	执行对处理地址的跳转	
可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令	执行下一条指令

(2) 通过复位释放 IDLE1 模式

执行与正常复位操作相同的操作。

表 21-5. IDLE1 模式中的操作状态

IDLE1 模式设置		操作状态	
		当不使用副时钟时	当使用副时钟时
主时钟振荡器		允许振荡	
副时钟振荡器		—	允许振荡
内部振荡器		允许振荡	
PLL		可操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作 (但是允许释放待机模式)	
定时器 P (TMP0 ~ TMP5)		停止操作	
定时器 Q (TMQ0)		停止操作	
定时器 M (TMM0)		当选择 $f_R/8$ 作为计数时钟时可操作	当选择 $f_R/8$ 或 f_{XT} 作为计数时钟时可操作
钟表定时器		当选择 f_X (BRG 分频) 作为计数时钟时可操作	可操作
看门狗定时器 2		当选择 f_R 作为计数时钟时可操作	当选择 f_R 或 f_{XT} 作为计数时钟时可操作
串行接口	CSIB0 ~ CSIB1	当选择 SCKBn 输入时钟作为计数时钟时可操作 ($n = 0 \sim 1$)	
	I ² C00 ~ I ² C02	停止操作	
	UARTA0 ~ UARTA2	停止操作 (但是当选择 ASCKA0 输入时钟时是可操作的)	
A/D 转换器		保持操作 (保持转换结果) ^注	
D/A 转换器		保持操作 (输出保持) ^注	
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	
外部总线接口		参见 2.2 引脚状态	
端口功能		保持设置 IDLE1 模式之前的状态	
内部数据		CPU 寄存器, 状态, 数据和全部其它的内部数据, 例如内部 RAM 的内容, 被保持为 IDLE1 模式设置之前的状态	

注 为实现低功耗, 在设置 IDLE1 模式之前停止 A/D 转换器和 D/A 转换器。

21.5 IDLE2 模式

21.5.1 设置和操作状态

通过在正常操作模式中设置 PSMR.PSM1 和 PSMR.PSM0 位为 10，并设置 PSC.STP 位为 1 设置 IDLE2 模式。

在 IDLE2 模式中，时钟振荡器继续操作但停止给 CPU，PLL，flash 存储器和其它内置外围设备功能的时钟供应。

结果，停止程序执行，并且保持在设置 IDLE2 模式之前的内部 RAM 的内容。CPU，PLL 和其它内置外围设备功能停止操作。但是，使用副时钟或外部时钟的内置外围设备功能继续操作。

表 21-7 显示了在 IDLE2 模式中的操作状态。

因为 IDLE2 模式停止了内置外围设备功能，PLL 和 flash 存储器的操作，所以 IDLE2 模式比 IDLE1 模式更能降低功耗。然而，因为 PLL 和 flash 存储器被停止，所以当释放 IDLE2 模式时，对 PLL 和 flash 存储器要求有一段设置时间。

注意事项 1. 在执行往 PSC 寄存器中存入数据来设置 IDLE2 模式的指令之后，插入五个或更多的 NOP 指令。

★ **2.** 如果设置 IDLE2 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则 IDLE2 模式立即被此等待响应的中断请求释放。

21.5.2 释放 IDLE2 模式

可通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号），未屏蔽的外部中断请求信号（INTP0 ~ INTP7 引脚输入），来自于在 IDLE2 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，WDT2RES 信号，低电压检测器（LVI）或时钟监视器（CLM）产生的复位）来释放 IDLE2 模式。在设置 IDLE2 模式之前，PLL 返回到操作状态。

在 IDLE2 模式释放后，恢复正常操作模式。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE2 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE2 模式，而不管中断请求信号的优先级。如果在某个中断服务程序中设置 IDLE2 模式，则之后产生的中断请求信号按如下所述来执行。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 IDLE2 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 IDLE2 模式，并且响应此中断请求信号。

注意事项 通过设置 PSC.NMI1M，PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号是无效的，IDLE2 模式也不会被释放。

表 21-6. 通过中断请求信号释放 IDLE2 模式后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	在保证规定的设置时间之后执行对处理地址的跳转	
可屏蔽中断请求信号	在保证规定的设置时间之后执行对处理地址的跳转或执行下一条指令	在保证规定的设置时间之后执行下一条指令

(2) 通过复位释放 IDLE2 模式

执行与正常复位操作相同的操作。

表 21-7. IDLE2 模式中的操作状态

IDLE2 模式设置		操作状态	
		当不使用副时钟时	当使用副时钟时
主时钟振荡器		允许振荡	
副时钟振荡器		—	允许振荡
内部振荡器		允许振荡	
PLL		停止操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是允许释放待机模式）	
定时器 P（TMP0 ~ TMP5）		停止操作	
定时器 Q（TMP0）		停止操作	
定时器 M（TMM0）		当选择 $f_R/8$ 作为计数时钟时可操作	当选择 $f_R/8$ 或 f_{XT} 作为计数时钟时可操作
钟表定时器		当选择 f_x （BRG 分频）作为计数时钟时可操作	可操作
看门狗定时器 2		当选择 f_R 作为计数时钟时可操作	当选择 f_R 或 f_{XT} 作为计数时钟时可操作
串行接口	CSIB0 ~ CSIB1	当选择 $SCKB_n$ 输入时钟作为计数时钟时可操作（ $n = 0 \sim 1$ ）	
	I ² C00 ~ I ² C02	停止操作	
	UARTA0 ~ UARTA2	停止操作（但是当选择 ASCKA0 输入时钟时 UARTA0 是可操作的）	
A/D 转换器		保持操作（保持转换结果） ^注	
D/A 转换器		保持操作（输出保持 ^注 ）	
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	
外部总线接口		参见 2.2 引脚状态	
端口功能		保持设置 IDLE2 模式之前的状态	
内部数据		CPU 寄存器，状态，数据和全部其它的内部数据，例如内部 RAM 的内容，被保持为 IDLE2 模式设置之前的状态	

注 为实现低功耗，在设置 IDLE2 模式之前停止 A/D 转换器和 D/A 转换器。

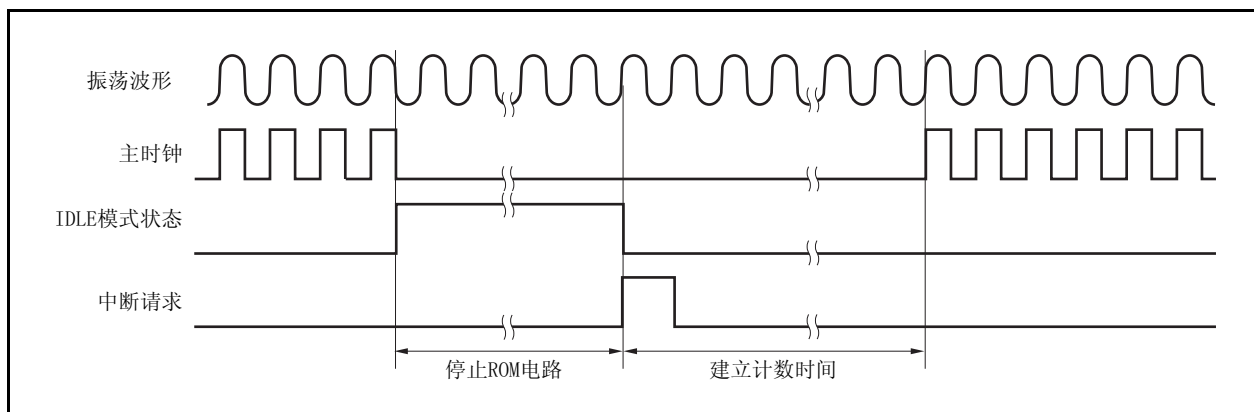
21.5.3 当释放 IDLE2 模式时保证设置时间

在释放 IDLE2 模式之后需保证 flash 存储器的设置时间，这是因为在设置 IDLE2 模式之后，除了主时钟振荡器外，block 的操作也被停止了。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE2 模式

通过设置 OSTS 寄存器保证指定的设置时间。

当产生释放源时，根据 OSTS 寄存器的设置，专用的内部定时器开始计数。当它溢出时，正常操作模式被恢复。



(2) 通过复位 ($\overline{\text{RESET}}$ 引脚输入, WDT2RES 产生) 释放

此操作与正常复位操作相同。

振荡稳定时间是 OSTS 寄存器的初始值, $2^{16}/f_x$ 。

21.6 STOP 模式

21.6.1 设置和操作状态

通过在正常操作模式中设置 PSMR.PSM1 和 PSMR.PSM0 位为 01 或 11，并设置 PSC.STP 位为 1 设置 STOP 模式。

在 STOP 模式中，副时钟振荡器继续操作，主时钟振荡器停止。停止给 CPU 和内置外围设备功能的时钟供应。

结果，停止程序执行，并且保持在设置 STOP 模式之前的内部 RAM 的内容。使用副时钟振荡器或外部时钟的内置外围设备功能继续操作。

表 21-9 显示了在 STOP 模式中的操作状态。

因为 STOP 模式停止了主时钟振荡器的操作，所以 STOP 模式比 IDLE2 模式更能降低功耗。如果不使用副时钟振荡器，内部振荡器和外部时钟，则功耗最低仅为漏电流消耗的功耗。

21.6.2 释放 STOP 模式

可通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号），未屏蔽的外部中断请求信号（INTP0 ~ INTP7 引脚输入），来自于在 STOP 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，WDT2RES 信号或低电压检测器（LVI）产生的复位）来释放 STOP 模式。

在 STOP 模式释放后，在振荡稳定时间被保证之后，恢复正常操作模式。

注意事项 通过设置 PSC.NMI1M，PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号是无效的，STOP 模式也不会被释放。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 STOP 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 STOP 模式，而不管中断请求信号的优先级。如果在某个中断服务程序中设置 STOP 模式，则之后产生的中断请求信号按如下所述来执行。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 STOP 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 STOP 模式，并且响应此中断请求信号。

注意事项 1. 在执行往 PSC 寄存器中存入数据来设置 STOP 模式的指令之后，插入五个或更多的 NOP 指令。

★ 2. 如果设置 STOP 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则 STOP 模式立即被此等待响应的中断请求释放。

表 21-8. 通过中断请求信号释放 STOP 模式后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	在保证振荡稳定时间之后执行对处理地址的跳转	
可屏蔽中断请求信号	在保证振荡稳定时间之后执行对处理地址的跳转或执行下一条指令	在保证振荡稳定时间之后执行下一条指令

(2) 通过复位释放 STOP 模式

执行与正常复位操作相同的操作。

表 21-9. STOP 模式中的操作状态

STOP 模式设置		操作状态	
		当不使用副时钟时	当使用副时钟时
主时钟振荡器		停止振荡	
副时钟振荡器		—	允许振荡
内部振荡器		允许振荡	
PLL		停止操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是允许释放待机模式）	
定时器 P（TMP0 ~ TMP5）		停止操作	
定时器 Q（TMP0）		停止操作	
定时器 M（TMM0）		当选择 $f_{n/8}$ 作为计数时钟时可操作	当选择 $f_{n/8}$ 或 f_{XT} 作为计数时钟时可操作
钟表定时器		停止操作	当选择 f_{XT} 作为计数时钟时可操作
看门狗定时器 2		当选择 f_n 作为计数时钟时可操作	当选择 f_n 或 f_{XT} 作为计数时钟时可操作
串行接口	CSIB0 ~ CSIB1	当选择 $SCKB_n$ 输入时钟作为计数时钟时可操作（ $n = 0 \sim 1$ ）	
	I ² C00 ~ I ² C02	停止操作	
	UARTA0 ~ UARTA2	停止操作（但是当选择 $ASCKA0$ 输入时钟时，UARTA0 是可操作的）	
A/D 转换器		停止操作（转换结果不确定）※s 1, 2	
D/A 转换器		停止操作※s 3, 1（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	
外部总线接口		参见 2.2 引脚状态	
端口功能		保持 STOP 模式设置之前的状态	
内部数据		CPU 寄存器，状态，数据和全部其它的内部数据，例如内部 RAM 的内容，被保持为 STOP 模式设置之前的状态	

- 注
1. 当操作 A/D 转换器时，如果设置 STOP 模式，则 A/D 转换器自动停止操作，并且在 STOP 模式释放后再次启动操作。然而，在这种情况下，释放 STOP 模式后的 A/D 转换结果是无效的。所有设置 STOP 模式之前的 A/D 转换结果也是无效的。
 2. 即使在操作 A/D 转换器时设置 STOP 模式，功耗降低也与在设置 STOP 模式之前停止 A/D 转换器时的一样。
 3. 如果操作 D/A 转换器时设置 STOP 模式，D/A 转换器自动停止并且引脚变为高阻态。在 STOP 模式被释放后，D/A 转换器重新启动，等待设置时间后，状态返回到设置 STOP 模式之前的输出电平。

★

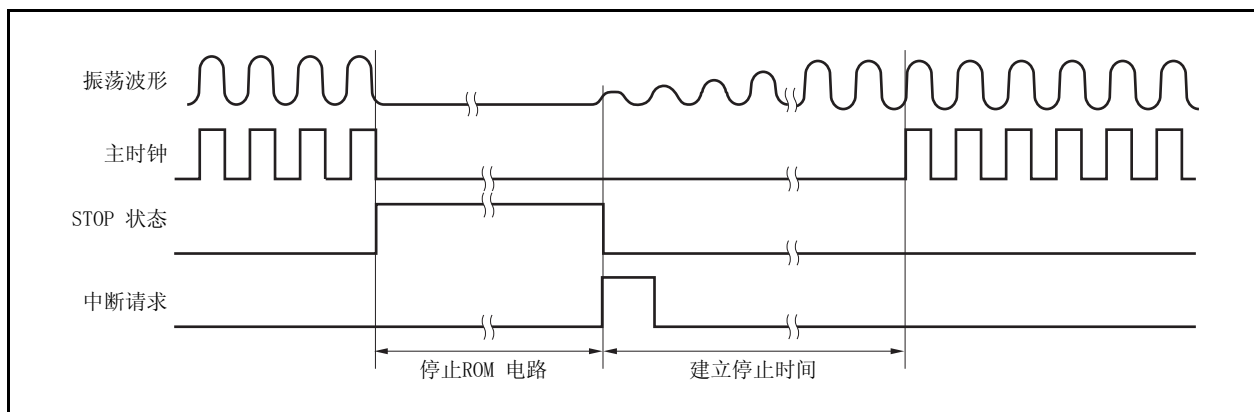
21.6.3 当释放 STOP 模式时保证振荡稳定时间

- ★ 在释放 STOP 模式后需保证主时钟振荡器的振荡稳定时间，这是因为在设置 STOP 模式之后主时钟振荡器操作停止。

(1) 通过不可屏蔽中断请求信号或未屏蔽的中断请求信号释放 STOP 模式

通过设置 OSTS 寄存器保证振荡稳定时间。

当产生释放源时，根据 OSTS 寄存器的设置，专用内部定时器开始计数。当它溢出时，恢复为正常操作模式。



(2) 通过复位释放

此操作与正常复位操作相同。

振荡稳定时间是 OSTS 寄存器的初始值， $2^{16}/f_x$ 。

21.7 副时钟操作模式

21.7.1 设置和操作状态

通过在正常操作模式中设置 PCC.CK3 位 1 来设置副时钟操作模式。

当副时钟操作模式被设置时，内部系统时钟从主时钟转换为副时钟。通过使用 PCC.CLS 位检测是否时钟已被切换。

当 PCC.MCK 位被设置为 1 时，主时钟振荡器的操作被停止。这样，系统仅以副时钟来操作。

在副时钟操作模式中，与正常操作模式相比，功耗降到一个较低的水平上，这是因为副时钟被用作内部系统时钟。除此之外，通过停止主时钟振荡器的操作，功耗能更进一步降低到 STOP 模式的等级。

表 21-10 显示了副时钟操作模式的状态。

注意事项 1. 当操作 CK3 位时，不要改变 PCC.CK2 ~ PCC.CK0 位（推荐使用位操作指令来操作）的设置值。对于 PCC 寄存器的细节描述，参见 6.3 (1) 处理器时钟控制寄存器(PCC)。

2. 如果以下的条件不满足，改变 CK2 ~ CK0 位的设置以便满足条件并且设置副时钟操作模式。

$$\text{主时钟}(f_{\text{xx}}) > \text{副时钟}(f_{\text{XT}} = 32.768 \text{ kHz}) \times 4$$

21.7.2 释放副时钟操作模式

当 CK3 位被清零时，可以通过复位信号（通过 RESET 引脚输入，WDT2RES 信号，低电压检测器（LVI）或时钟检测器（CLM）复位）释放副时钟操作模式。

如果停止主时钟（MCK 位 = 1），设置 MCK 位为 1，通过软件保证主时钟的振荡稳定时间，并将 CK3 位清零。

当副时钟操作模式释放后，恢复为正常操作模式。

注意事项 当操作 CK3 位时，不要改变 CK2 ~ CK0 位（推荐使用位操作指令来操作）的设置值。

对于 PCC 寄存器的细节描述，参见 6.3 (1) 处理器时钟控制寄存器(PCC)。

表 21-10. 副时钟操作模式中的操作状态

副时钟操作模式设置		操作状态	
项目		当主时钟振荡时	当主时钟停止时
副时钟振荡器		允许振荡	
内部振荡器		允许振荡	
PLL		可操作	停止操作 ^注
CPU		可操作	
DMA		可操作	
中断控制器		可操作	
定时器 P (TMP0 ~ TMP5)		可操作	停止操作
定时器 Q (TMP0)		可操作	停止操作
定时器 M (TMM0)		可操作	当选择 $f_{R/8}$ 或 f_{XT} 作为计数时钟时可操作
钟表定时器		可操作	当选择 f_{XT} 作为计数时钟时可操作
看门狗定时器 2		可操作	当选择 f_R 或 f_{XT} 作为计数时钟时可操作
串行接口	CSIB0 ~ CSIB1	可操作	当选择 \overline{SCKBn} 输入时钟作为计数时钟时可操作 ($n = 0 \sim 1$)
	I ² C00 ~ I ² C02	可操作	停止操作
	UARTA0 ~ UARTA2	可操作	停止操作 (但是当选择 ASCKA0 输入时钟时 UARTA0 是可操作的)
A/D 转换器		可操作	停止操作
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	
外部总线接口		参见 2.2 引脚状态	
端口功能		可设置	
内部数据		可设置	

注 在停止主时钟之前，需要先确认先停止 PLL(PLLCTL.PLLON 位 = 0)。

注意事项 当 CPU 工作在副时钟并且主时钟振荡停止时，不能访问发生等待的寄存器。如果产生等待，则其只能通过复位来释放 (见 3.4.8 (2))。

21.8 副 IDLE 模式

21.8.1 设置和操作状态

通过在副时钟操作模式中设置 PSMR.PSM1 和 PSMR.PSM0 位为 00 或 10，并设置 PSC.STP 位为 1 设置副 IDLE 模式。

在此模式中，时钟振荡器继续操作，停止给 CPU，flash 存储器和其它内置外围设备功能的时钟供应。

结果，停止程序执行，并且保持设置副 IDLE 模式之前的内部 RAM 的内容。停止 CPU 和其它内置外围设备功能。然而，可使用副时钟或外部时钟操作的内置外围设备功能继续操作。

因为副 IDLE 模式停止了 CPU，flash 存储器和其它外围设备功能的操作，所以它比副时钟操作模式更能降低功耗。如果在主时钟停止后设置副 IDLE 模式，当前功耗能降低至 STOP 模式的等级。

表 21-12 显示了副 IDLE 模式中的操作状态。

注意事项 1. 在执行往 PSC 寄存器中存入数据来设置副 IDLE 模式的指令之后，插入五个或更多的 NOP 指令。

★ 2. 如果设置副 IDLE 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则副 IDLE 模式立即被此等待响应的中断请求释放。

21.8.2 释放副 IDLE 模式

可通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号），未屏蔽的外部中断请求信号（INTP0 ~ INTP7 引脚输入），来自于在副 IDLE 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，WDT2RES 信号，低电压检测器（LVI）或时钟监视器（CLM）产生的复位）来释放副 IDLE 模式。在设置副 IDLE 模式之前，PLL 返回到操作状态。

当通过中断请求信号释放副 IDLE 模式时，副时钟操作模式被设置。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放副 IDLE 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放副 IDLE 模式，而不管中断请求信号的优先级。如果在某个中断服务程序中设置副 IDLE 模式，则之后产生的中断请求信号按如下所述来执行。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放副 IDLE 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放副 IDLE 模式，并且响应此中断请求信号。

注意事项 1. 通过设置 PSC.NMI1M，PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号是无效的，副 IDLE 模式也不会被释放。

- 2. 当释放副 IDLE 模式时，从产生释放副 IDLE 模式的中断请求信号开始到副 IDLE 模式被释放之间须等待 12 个副时钟周期（大约 366 μ s）。

表 21-11. 通过中断请求信号释放副 IDLE 模式之后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	执行对处理地址的跳转	
可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令	执行下一条指令

(2) 通过复位释放副 IDLE 模式

执行与正常复位操作相同的操作。

表 21-12. 在副 IDLE 模式中的操作状态

副 IDLE 模式设置		操作状态	
		当主时钟振荡时	当主时钟停止时
副时钟振荡器		允许振荡	
内部振荡器		允许振荡	
PLL		可操作	停止操作 ^{※1}
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是允许释放待机模式）	
定时器 P（TMP0 ~ TMP5）		停止操作	
定时器 Q（TMP0）		停止操作	
定时器 M（TMM0）		当选择 f _R /8 或 f _{XT} 作为计数时钟时可操作	
钟表定时器		停止操作	当选择 f _{XT} 作为计数时钟时可操作
看门狗定时器 2		当选择 f _R 或 f _{XT} 作为计数时钟时可操作	
串行接口	CSIB0 ~ CSIB1	当选择 SCKB _n 输入时钟作为计数时钟时可操作（n = 0 ~ 1）	
	I ² C00 ~ I ² C02	停止操作	
	UARTA0 ~ UARTA2	停止操作（但是当选择 ASCKA0 输入时钟时 UARTA0 是可操作的）	
A/D 转换器		保持操作（保持转换结果） ^{※2}	
D/A 转换器		保持操作（输出保持 ^{※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	
外部总线接口		见 2.2 引脚状态（与 IDLE1, IDLE2 模式相同的操作状态）	
端口功能		保持设置副 IDLE 模式之前的状态	
内部数据		CPU 寄存器，状态，数据和全部其它的内部数据，例如内部 RAM 的内容，被保持为副 IDLE 模式设置之前的状态	

- 注
1. 停止主时钟之前确定停止 PLL（PLLCTL.PLLON 位 = 0）。
 2. 为了实现低功耗，在进入副 IDLE 模式之前停止 A/D 和 D/A 转换器。

第二十二章 复位功能

22.1 概述

如下复位功能可用。

(1) 四种复位源

- 由 **RESET** 引脚输入的外部复位
- 由看门狗定时器 2(WDT2)溢出(WDT2RES)引起的复位
- 由于低电压检测器(LVI)比较供电电压和检测电压引起的系统复位
- 由于检测到时钟监控器(CLM)停止振荡引起的系统复位

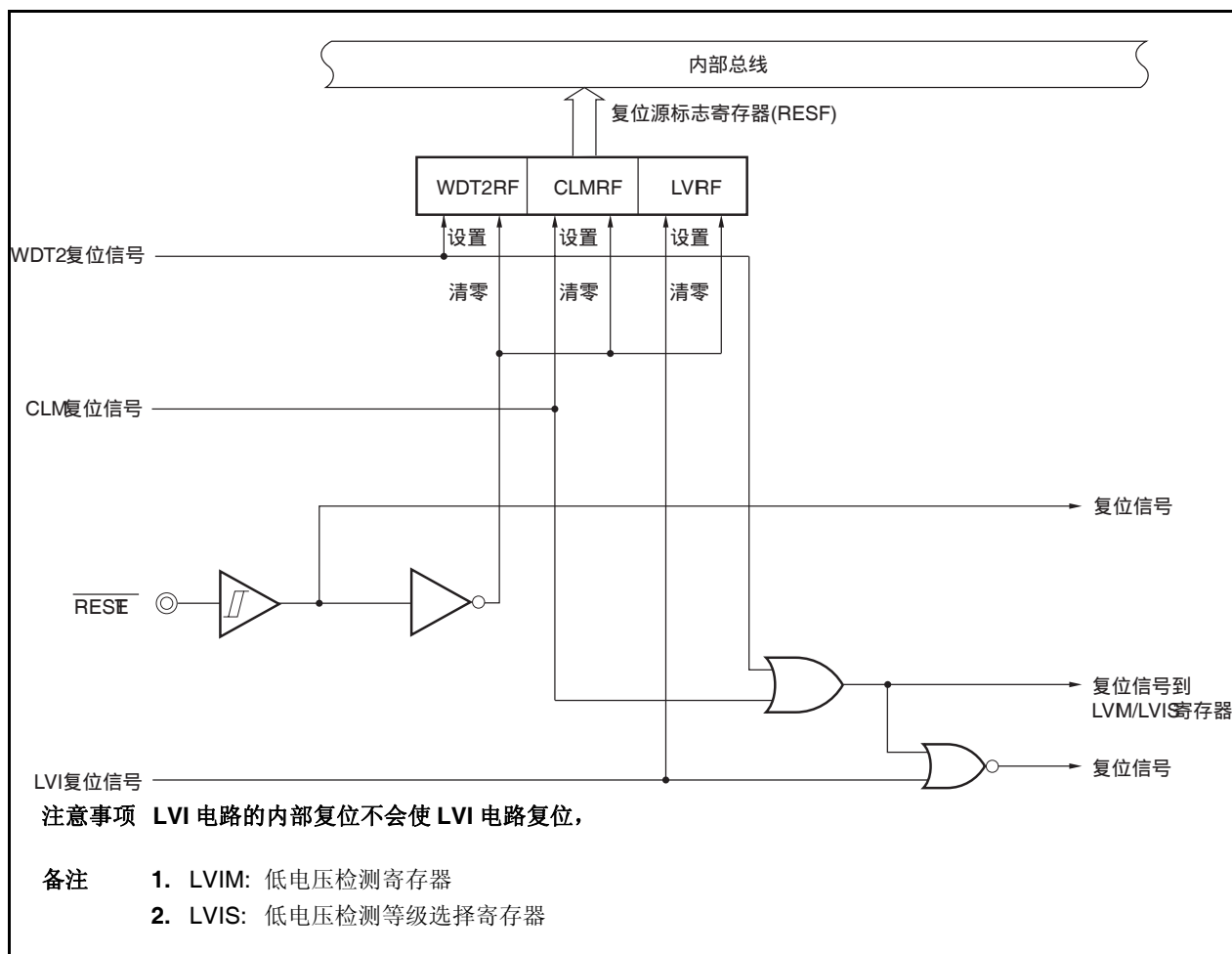
当一个复位信号释放后，复位源通过复位源标志寄存器(RESF)来确认。

(2) 紧急事件操作模式

复位后在主时钟振荡稳定时间内如果 WDT2 溢出，判断为一个不规则的主时钟振荡并且 CPU 在内部振荡时钟下开始操作。

注意事项 当 CPU 在内部振荡时钟下操作，禁止访问会产生等待状态的寄存器。关于会产生等待状态的寄存器，参见 3.4.8 (2) 访问特殊片上外围 I/O 寄存器。

图 22-1 复位功能框图



22.2 检测复位源的寄存器

V850ES/JG2 有 4 个复位源。当一个复位释放后，出现的复位源可以通过复位源标志寄存器(RESF)检测。

(1) 复位源标志寄存器 (RESF)

RESF 寄存器是一个特殊寄存器，只能够通过特殊的顺序来写入 (参见 3.4.7 特殊寄存器)。

RESF 寄存器表明了复位信号产生的源。

这个寄存器可以通过位指令和 8 位指令读取和写入。

$\overline{\text{RESET}}$ 引脚输入清除这些寄存器为 00H。如果复位源不是 $\overline{\text{RESET}}$ 引脚信号，缺省值则有所不同。

复位后: 00H R/W 地址: FFFFF888H

	7	6	5	4	3	2	1	0
RESF	0	0	0	WDT2RF	0	0	CLMRF	LVRF

WDT2RF	从 WDT2产生的复位信号
0	不产生
1	产生

CLMRF	从 CLM产生的复位信号
0	不产生
1	产生

LVRF	从 LVI产生的复位信号
0	不产生
1	产生

注 由 $\overline{\text{RESET}}$ 引脚引起的复位将把 RESF 寄存器的值清为 00H。由看门狗定时器 2(WDT2)，低电压检测器 (LVI)，或者时钟监控器(CLM)引起的复位，寄存器复位标志(WDT2RF 位，CLMRF 位，和 LVRF 位)被设置。但是，其他源保留。

注意事项 寄存器的每一位只能被写入“0”。如果写入“0”和设置标志(复位事件)冲突，设置标志优先。

22.3 操作

22.3.1 由 RESET 引脚引起的复位操作

当 RESET 引脚输入一个低电平，系统复位，并且所有硬件初始化。

当 RESET 引脚的电平由低变为高，复位状态释放。

表 22-1 RESET 引脚输入后的硬件状态

项目	复位期间	复位后
主时钟振荡器(fx)	停止振荡	开始振荡
副时钟振荡器 (fxr)	振荡继续	
内部振荡器	停止振荡	开始振荡
外围时钟(fx to fx/1,024)	停止操作	振荡稳定时间之后开始振荡
内部系统时钟(fCLK), CPU 时钟(fCPU)	停止操作	振荡稳定时间之后开始操作(初始化为 fxx/8)
CPU	初始化	振荡稳定时间之后开始执行程序
看门狗定时器 2	停止操作(初始化到 0)	以内部振荡时钟作为源时钟从 0 开始累加
内部 RAM	如果上电复位或者 CPU 访问和复位输入冲突(数据被破坏)则不确定，其余值在复位后则立即保留 ^{注1}	
I/O 引脚(端口/复用引脚)	高阻抗 ^{注2}	
片上外围 I/O 寄存器	初始化为指定状态，OCDM 寄存器被设置为(01H)。	
其他片上外围功能	停止操作	振荡稳定时间之后可以开始操作

- 注**
1. 因为支持启动交换(boot swap)功能，V850ES/JG2 的固件在内部系统状态被释放后使用了一部分内部 RAM。因此，一些 RAM 区域的内容在上电复位后不能被保持。详细内容参见 22.3.4 复位释放后的操作
 2. 当电源开启，下列引脚可能临时在复位时输出一个不确定的电平
 - P10/ANO0 引脚
 - P11/ANO1 引脚
 - P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 引脚

注意事项 RESET 引脚的输入初始化 OCDM 寄存器。因此，请注意复位释放后在 OCDM.OCDM0 位被清除前如果一个高电平输入到 P05/DRST 引脚，可能会进入片上调试模式。详细内容参见第四章 端口功能。

图 22-2 RESET 引脚输入引起的复位操作时序

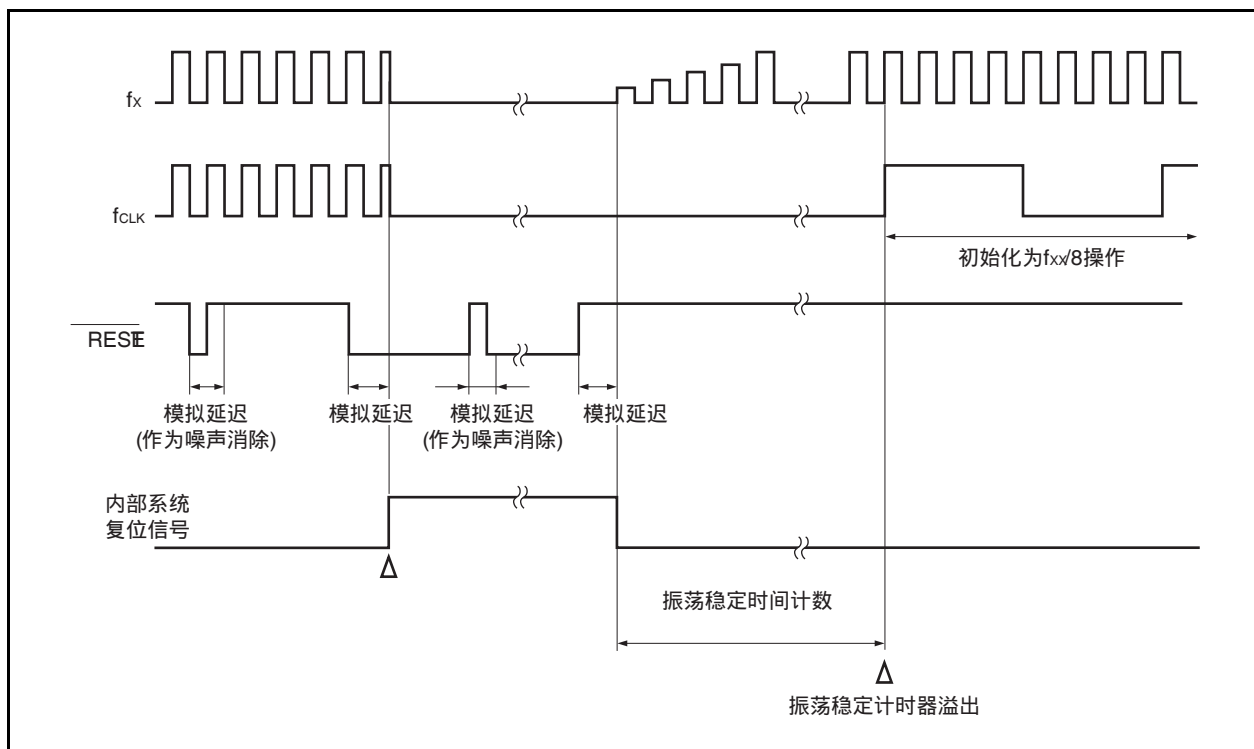
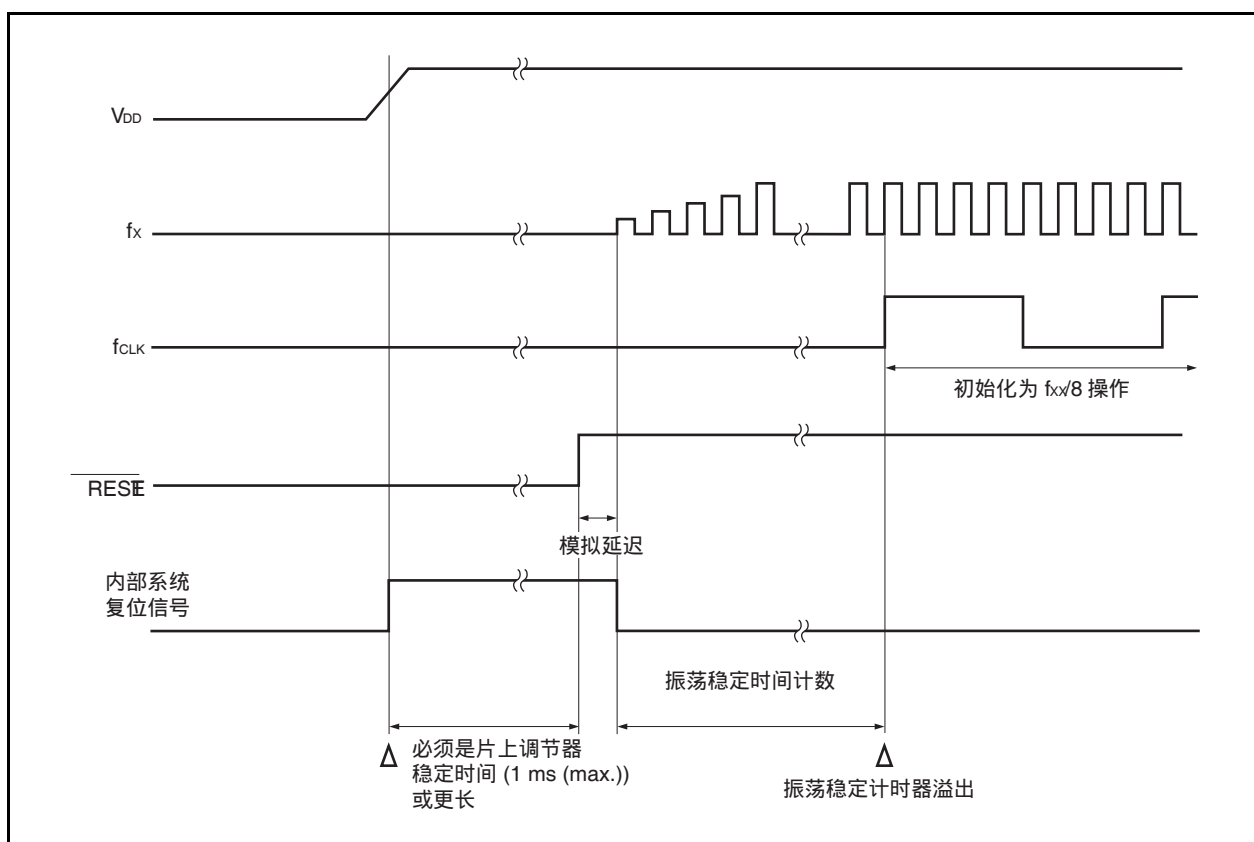


图 22-3 上电复位操作时序



22.3.2 由看门狗定时器 2 引起的复位操作

当看门狗定时器 2 溢出被设置为复位操作模式，在看门狗定时器 2 溢出上(产生 WDT2RES 信号)，执行系统复位并且硬件初始化到初始化状态。

在看门狗定时器 2 溢出后，进入复位状态并且持续到预定时间(模拟延时)，并且复位状态自动释放。

在复位期间主时钟振荡停止。

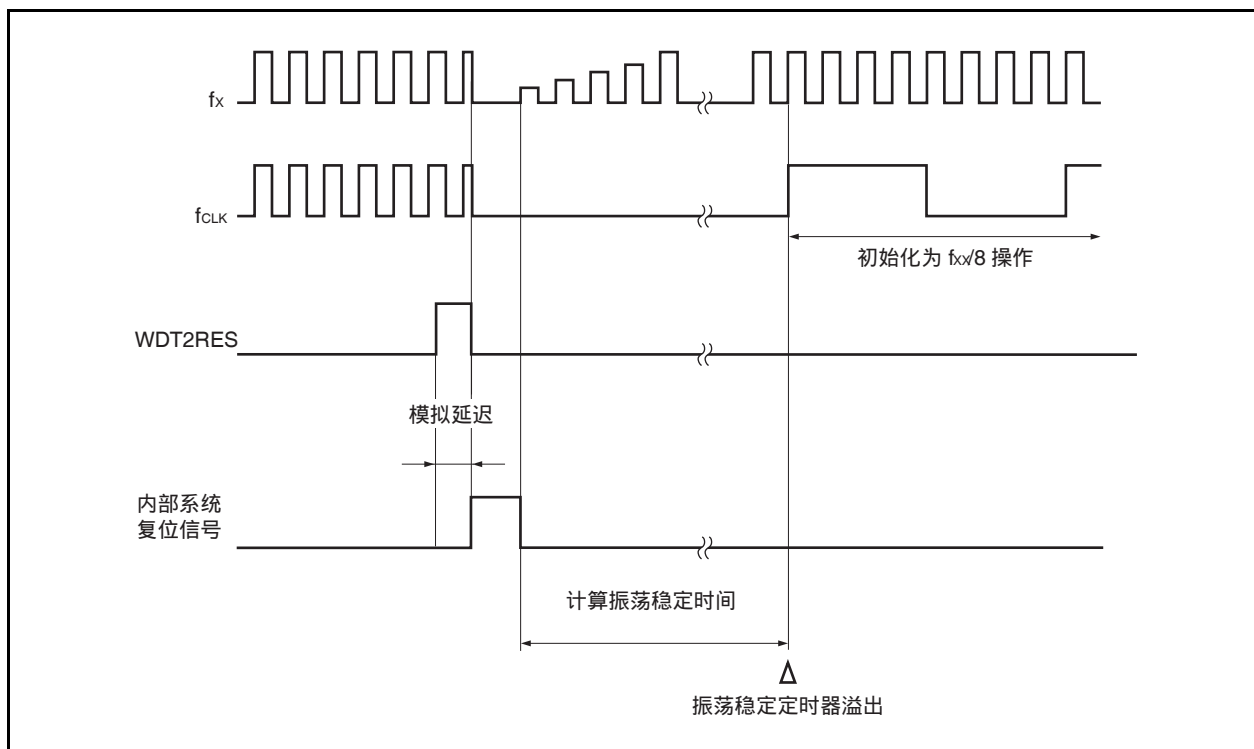
表 22-2 看门狗定时器 2 复位操作时的硬件状态

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
外围时钟(fx to fx/1,024)	操作停止	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	振荡稳定时间之后开始振荡(初始化为 fxx/8)
CPU	初始化	振荡稳定时间之后开始执行程序
看门狗定时器 2	操作停止(初始化为 0)	以内部振荡时钟作为源时钟从 0 开始累加
内部 RAM	如果上电复位或者 CPU 访问和复位输入冲突(数据被破坏)则不确定，其余值在复位后则立即保留 ^注	
I/O 引脚(端口/复用引脚)	高阻抗	
片上外围 I/O 寄存器	初始化为指定状态，OCDM 寄存器值保留。	
其他片上外围功能	操作停止	振荡稳定时间之后可以开始操作

注 因为支持启动交换功能，V850ES/JG2 的固件在内部系统状态被释放后使用了一部分内部 RAM。因此，一些

RAM 区域的内容在上电复位后不能被保持。详细内容参见 22.3.4 复位释放后的操作

图 22-4 WDT2RES 信号引起的复位操作时序



22.3.3 低电压检测器引起的复位操作

当开启 LVI 操作时如果供电电压低于低电压检测器检测的电压，产生一个系统复位(当 LVIM.LVIMD 位被设为 1)，并且硬件初始化到初始化状态。

复位状态从电源掉电持续到供电电压高于低电压检测器检测的电压。

复位期间主时钟振荡器停止振荡。

当 LVIMD 位= 0，如果低电压被检测到，产生一个中断请求信号(INTLVI)。

表 22-3 低电压检测器引起的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
复时钟振荡器 (fxr)	振荡继续	
内部振荡器	振荡停止	振荡开始
外围时钟(fx to fx/1,024)	操作停止	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	振荡稳定时间之后开始振荡(初始化为 fxx/8)
CPU	初始化	振荡稳定时间之后开始执行程序
看门够定时器 2	操作停止(初始化为 0)	以内部振荡时钟作为源时钟从 0 开始累加
内部 RAM	如果上电复位或者 CPU 通路河复位输入冲突(数据被破坏)则不确定，其余值在复位后则立即保留 ^注	
I/O 引脚(端口/复用引脚)	高阻抗	
片上外围 I/O 寄存器	初始化为指定状态，OCDM 寄存器值保留。	
LVI	操作停止	
其他片上外围功能	操作停止	振荡稳定时间之后可以开始操作

注 因为支持启动交换功能，V850ES/JG2 的固件在内部系统状态被释放后使用了一部分内部 RAM。因此，一些

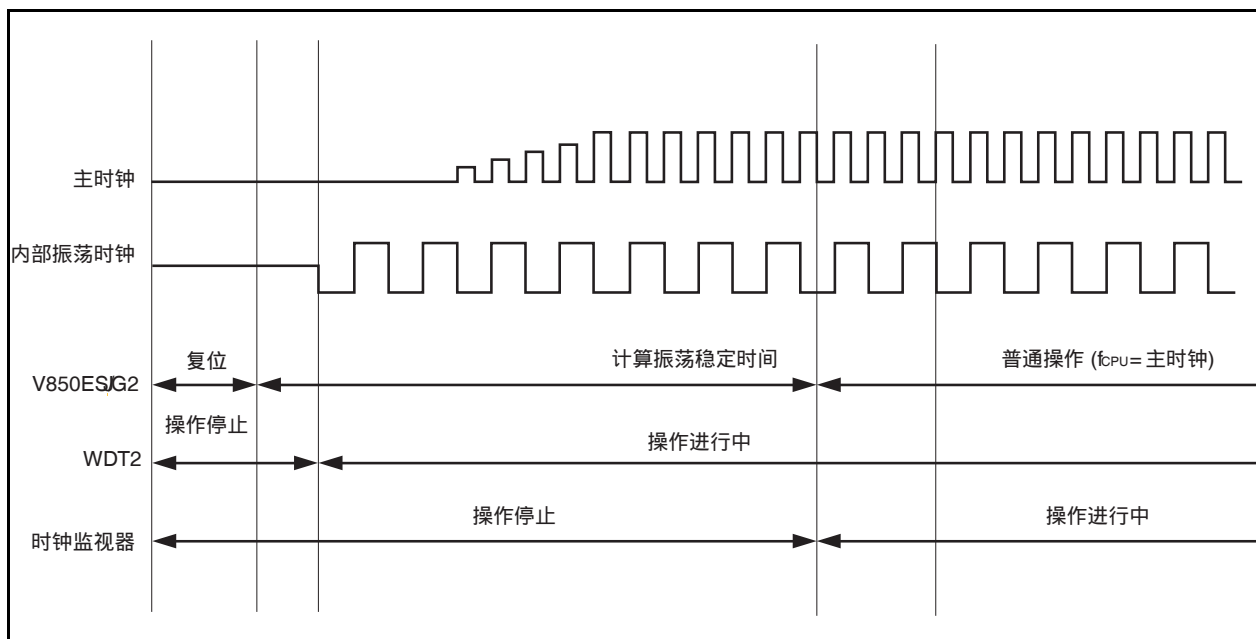
RAM 区域的内容在上电复位后不能被保持。详细内容参见 22.3.4 复位释放后的操作

备注 低电压检测器的复位时序，参见第 24 章 低电压检测器(LVI)

22.3.4 复位释放后的操作

复位释放以后，主时钟开始振荡，经过振荡稳定时间(OSTS 寄存器初始值为： $2^{16}/f_x$)后，CPU 开始执行程序。在复位释放后 WDT2 立即以内部振荡时钟为时钟源开始操作。

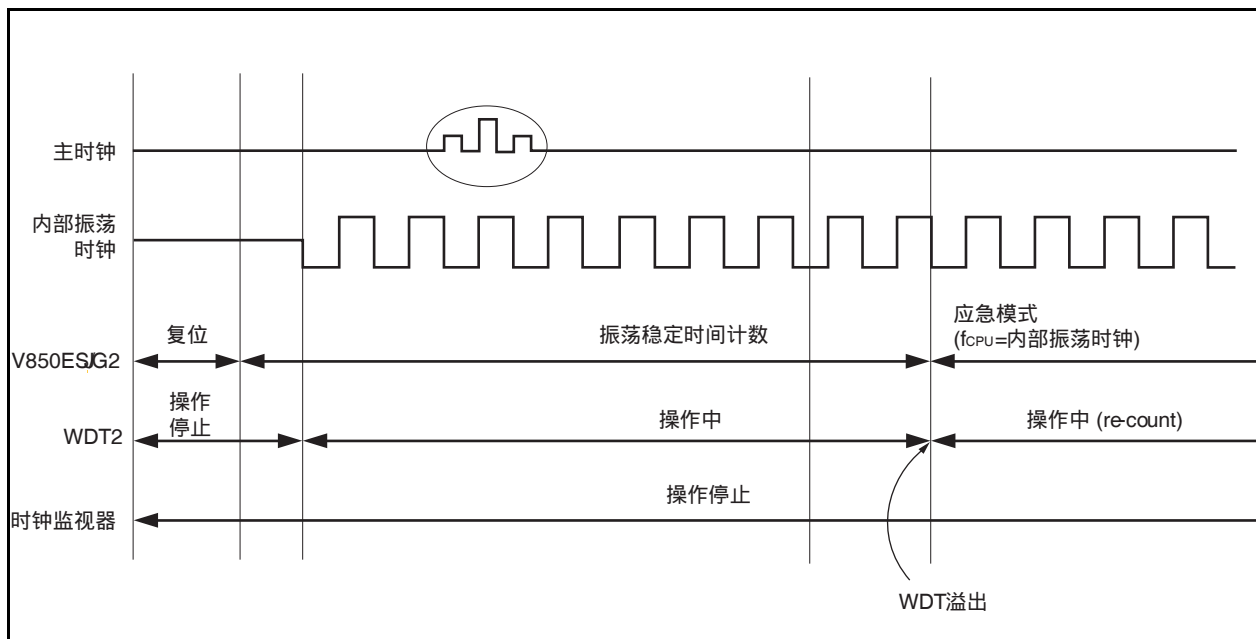
图 22-5 复位后的操作



(1) 紧急操作模式

如果在安全振荡稳定时间之中发生一个主时钟异常。WDT2 在执行 CPU 程序之前溢出。此时，CPU 使用内部振荡时钟作为时钟源开始执行程序。

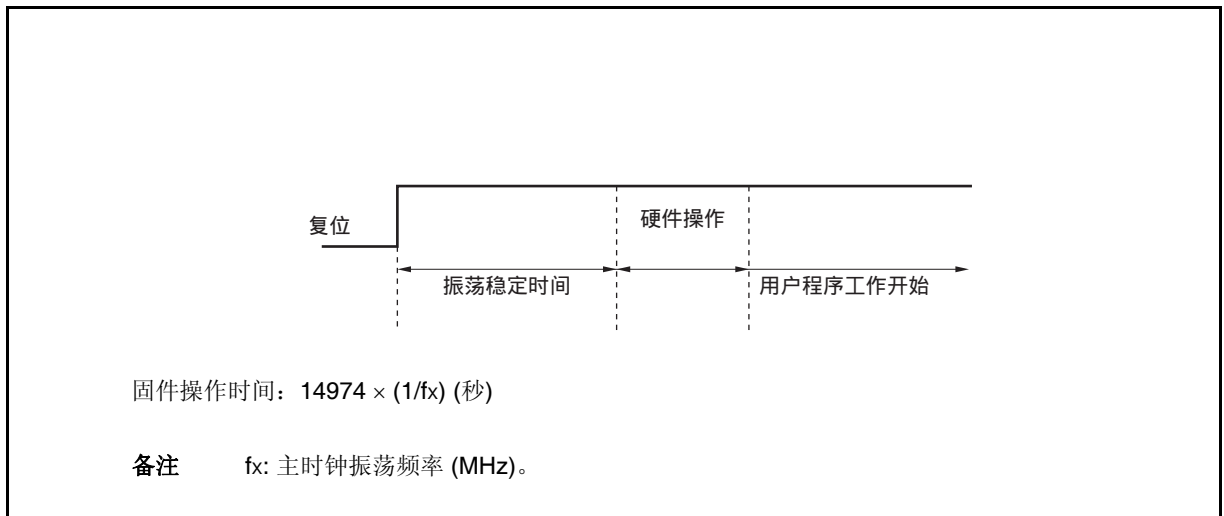
图 22-6 复位释放后的操作



CPU 操作时钟状态可以通过 CPU 操作时钟状态寄存器(CCLS)来检测。

(2) 固件操作

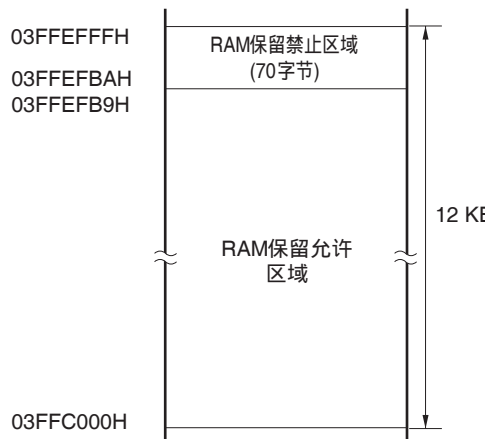
在 V850ES/JG2 里复位状态被释放后，内部固件在用户程序执行前进行支持启动切换(boot swap)的操作。



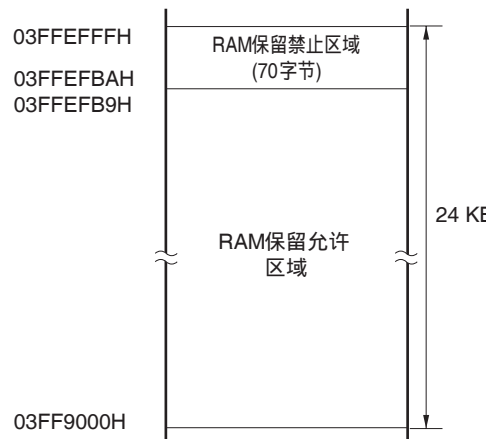
由于固件使用一部分内部 RAM，下列 RAM 区域在上电复位后不被保持。

- 12 KB RAM 版本: 03FFEFBAH 到 03FFEFFFH
- 24 KB RAM 版本: 03FFEFBAH 到 03FFEFFFH
- 32 KB RAM 版本: 03FF7000H 到 03FF7095H, 03FFEFBAH 到 03FFEFFFH
- 40 KB RAM 版本: 03FFEFBAH 到 03FFEFFFH
- 48 KB RAM 版本: 03FF3000H 到 03FF3095H, 03FFEFBAH 到 03FFEFFFH

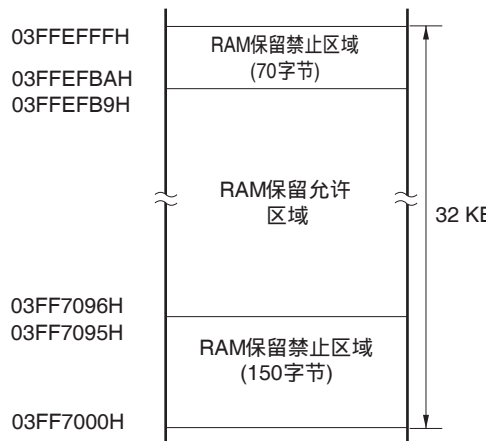
(a) 12 KB RAM 版本



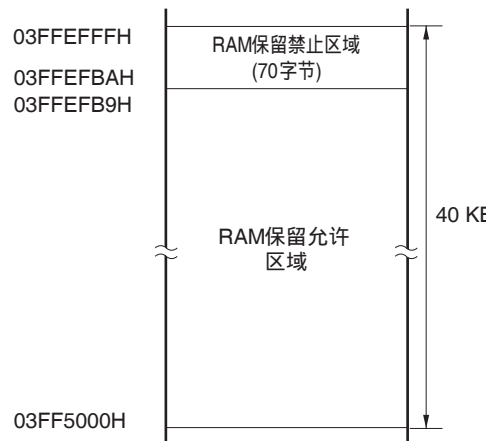
(b) 24 KB RAM 版本



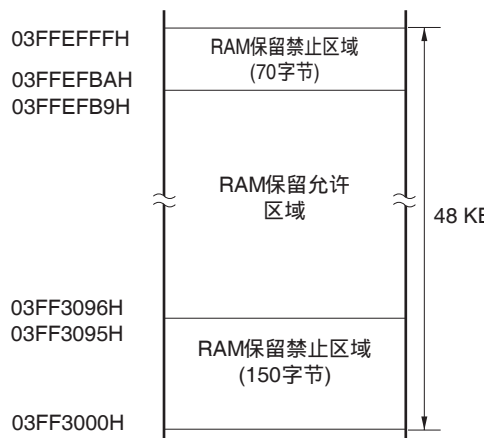
(c) 32 KB RAM 版本



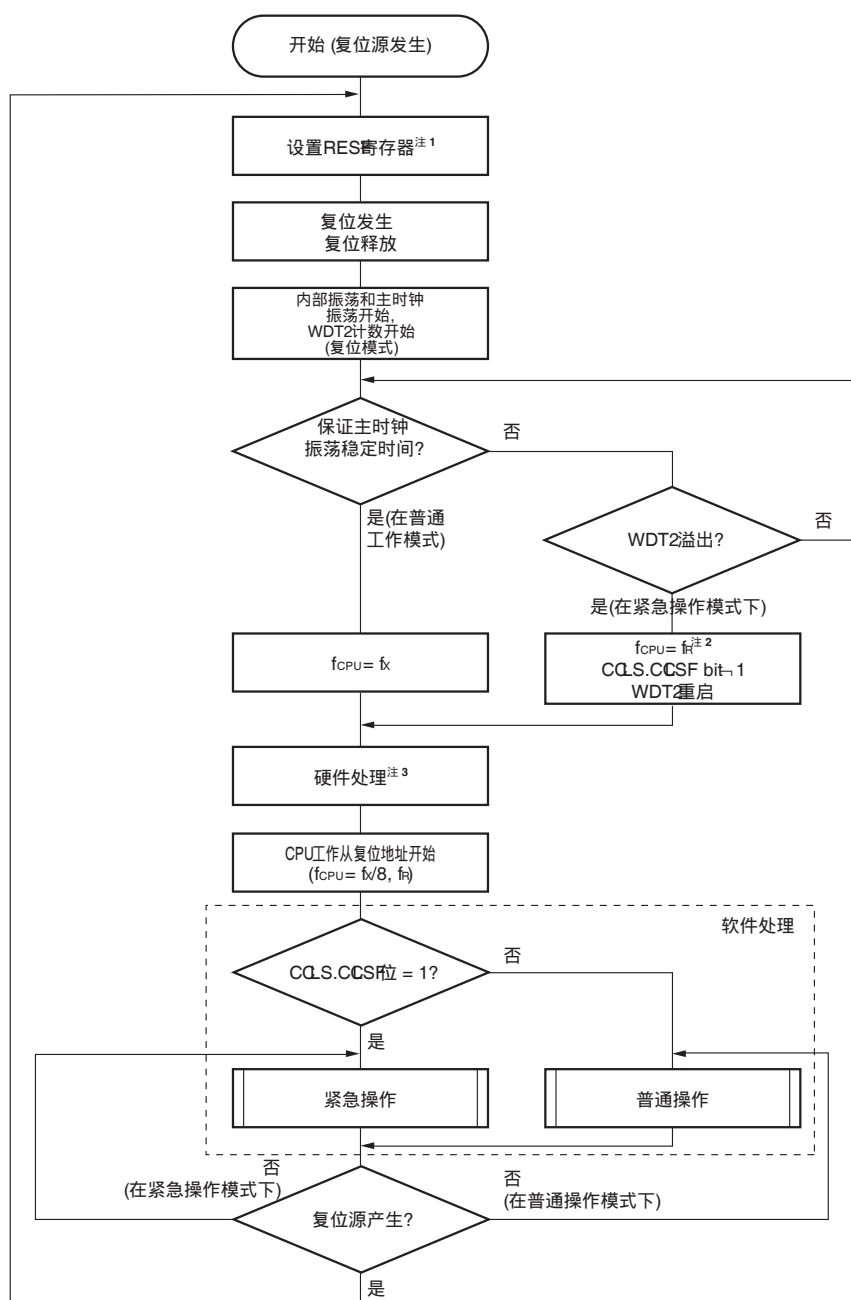
(d) 40 KB RAM 版本



(e) 48 KB RAM 版本



22.3.5 复位功能操作流程



注 1. 各寄存器位的设置根据复位源的不同而不同。

复位源	WDT2RF 位	CRMRF 位	LVIRF 位
RESET pin	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振荡。

第二十三章 时钟监控器

23.1 功能

时钟监控器通过使用内部振荡时钟对主时钟采样，并且在主时钟停止振荡的时候产生一个复位信号。

一旦通过操作允许标志允许时钟监控器的操作，除了复位无法通过其余任何方法来清零。

由时钟监控器产生复位时，RESF.CLMRF 位被设置，关于 RESF 寄存器的详细设置，参见 22.2 检测复位源的寄存器。

时钟监控器在下列条件下自动停止。

- STOP 模式释放后的振荡稳定时间内
- 当主时钟停止时(从副时钟操作 PCC.MCK 位= 1 开始，到主时钟操作 PCC.CLS 位= 0 结束)
- 当采样时钟(内部振荡时钟)停止时
- 当 CPU 使用内部振荡时钟操作时

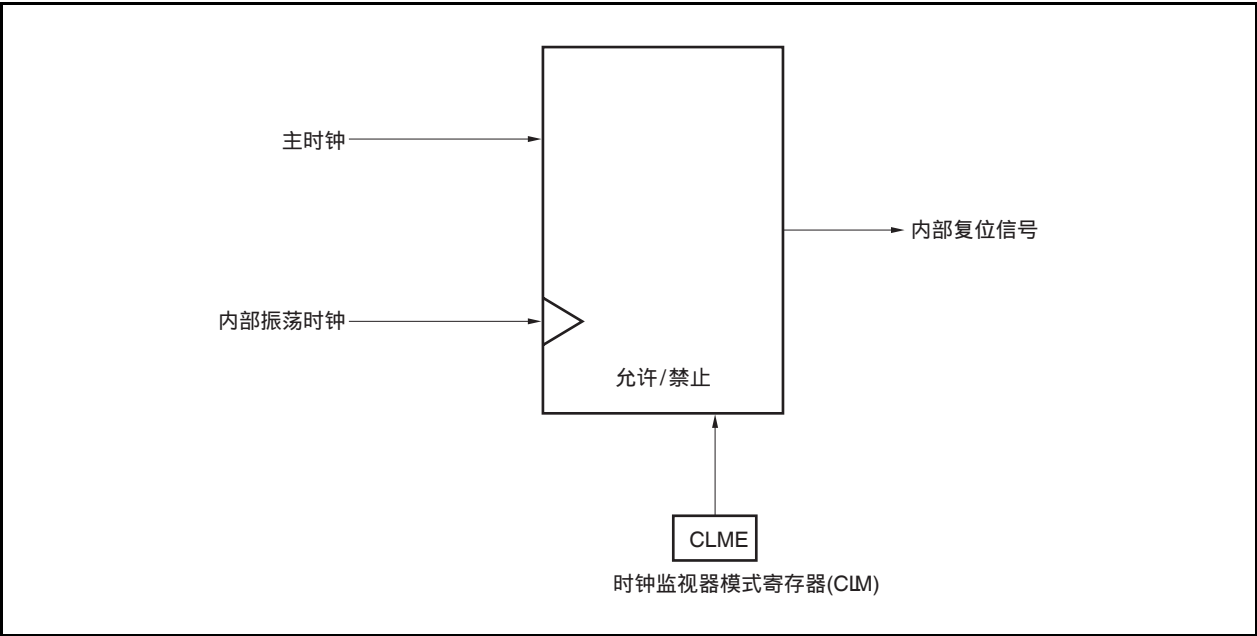
23.2 结构

时钟监控器由下列硬件组成。

表 23-1 时钟监控器的构成

项目	构成
控制寄存器	时钟监控器模式寄存器 (CLM)

图 23-1 RESET 引脚输入引起的复位时序



23.3 寄存器

时钟监控器通过时钟监控器模式寄存器(CLM)来控制。

(1) 时钟监控器模式寄存器(CLM)

CLM 寄存器是一个特殊寄存器。只能够通过特殊的顺序来写入(参见 3.4.7 特殊寄存器)。

这个寄存器用来设置操作时钟监控器的模式。

这个寄存器可以通过位指令和 8 位指令来读取和写入。

复位输入清除这个寄存器为 00H。

复位后: 00H		R/W	地址: FFFFF870H					
	7	6	5	4	3	2	1	<0>
CLM	0	0	0	0	0	0	0	CLME
CLME	时钟监控器操作允许或禁止							
0	禁止时钟监控器操作.							
1	允许时钟监控器操作.							

- 注意事项
- 一旦 CLME 位被设为 1，除了复位其余任何操作都不能将其清为 0。
 - 由时钟监控器引起的一个复位，CLME 位被清为 0 并且 RESF.CLMRF 被设置为 1。

23.4 操作

这个部分解释了时钟监控器的功能。下面是开始和停止条件。

<开始条件>

当 CLM.CLME 位设置为 1 时允许操作。

<停止条件>

- 当 STOP 模式释放后开始计数振荡稳定时间时
- 当主时钟停止(从副时钟操作时 PCC.MCK 位=1 到主时钟操作时 PCC.CLS bit 位=0 为止)时
- 当采样时钟停止(内部振荡时钟)时
- 当 CPU 使用内部振荡时钟操作时

表 23-2 时钟监控器的操作状态
(当 CLM.CLME 位=1, 在内部振荡时钟操作时)

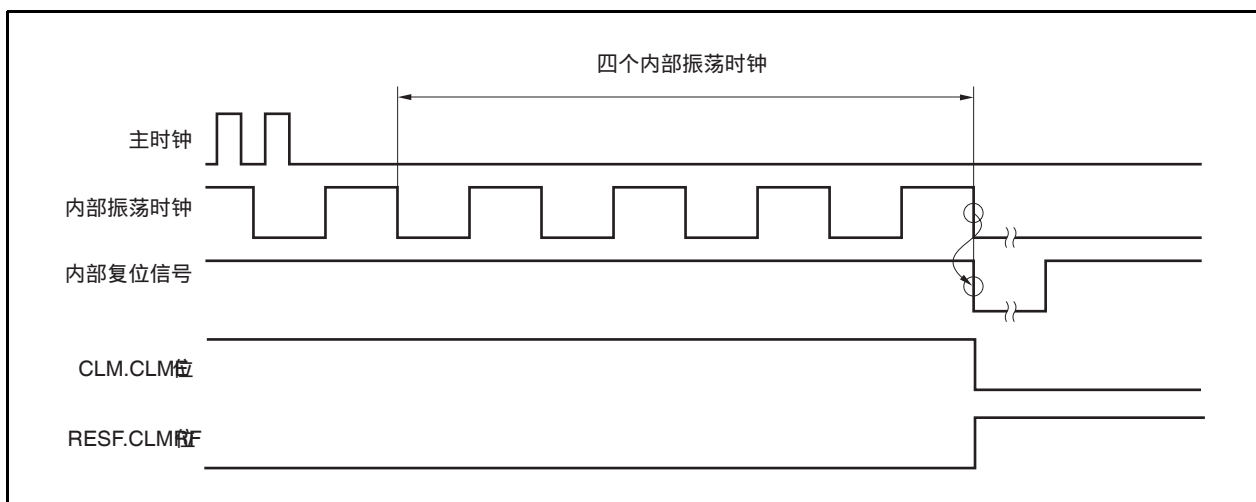
CPU 操作时钟	操作模式	主时钟状态	内部振荡时钟状态	时钟监控器状态
主时钟	HALT 模式	振荡	振荡 ^{注1}	操作 ^{注2}
	IDLE1, IDLE2 模式	振荡	振荡 ^{注1}	操作 ^{注2}
	STOP 模式	停止	振荡 ^{注1}	停止
副时钟 (PCC 寄存器的 MCK 位 = 0)	Sub-IDLE 模式	振荡	振荡 ^{注1}	操作 ^{注2}
副时钟 (PCC 寄存器的 MCK 位 = 1)	Sub-IDLE 模式	停止	振荡 ^{注1}	停止
内部振荡时钟	—	停止	振荡 ^{注3}	停止
复位期间	—	停止	停止	停止

- 注
1. 内部振荡器可以通过设置 RCM.RSTOP 位为 1 来停止
 2. 内部振荡器停止的时候时钟监控器也停止
 3. 内部振荡器不能通过软件来停止

(1) 当主时钟振荡器停止时(CLME 位=1)的操作

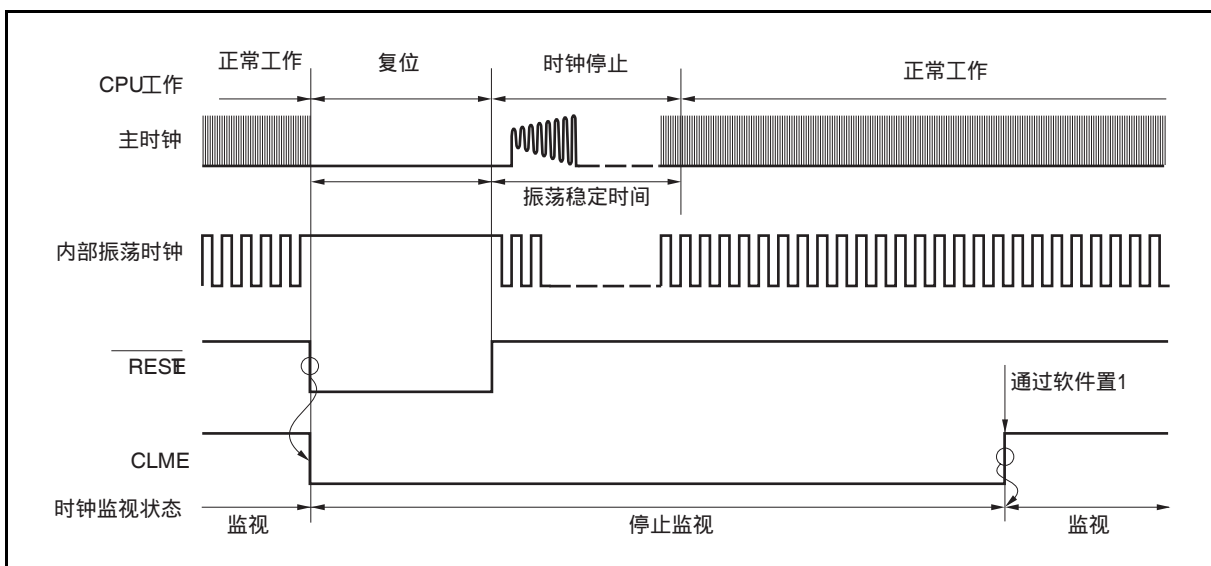
当 CLME 位=1 的时候主时钟的振荡停止，一个如图 23-2 的内部复位信号就会产生。

图 23-2 主时钟振荡停止引起的复位周期

**(2) RESET 输入后的时钟监控器状态**

RESET 输入清除 CLM.CLME 位为 0 并且停止时钟监控器的操作。在主时钟振荡稳定时间后当通过软件把 CLME 位设为 1，监控操作开始。

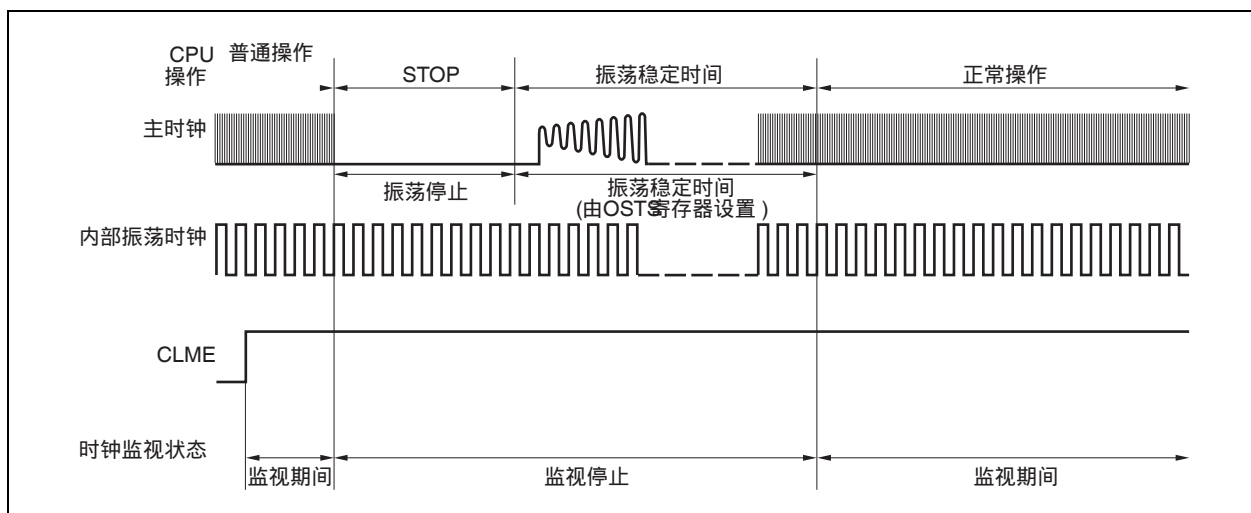
图 23-3 RESET 输入后的时钟监控器状态
(RESET 输入并且在主时钟振荡稳定时间后设置 CLM.CLME 位=1)



(3) STOP 模式和 STOP 模式释放后的操作

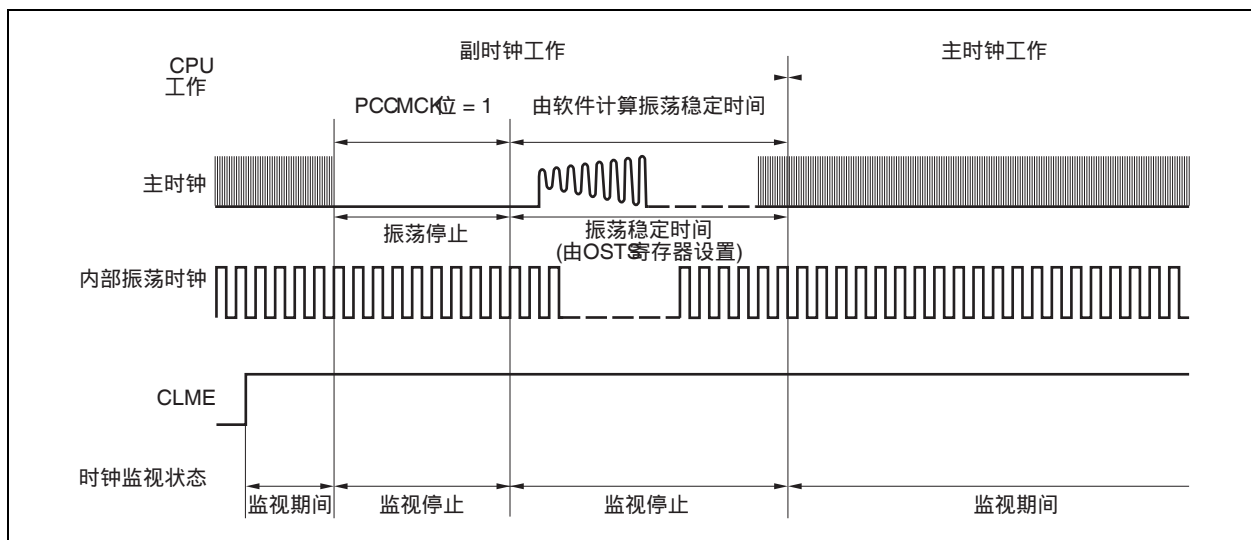
当设置 CLM.CLME 位=1 时进入 STOP 模式，在 STOP 模式和振荡稳定时间计数下监控操作停止。振荡稳定时间过后，监控器自动开始操作。

图 23-4 STOP 模式或 STOP 模式释放后的操作

**(4) 主时钟停止时的操作(任意时刻)**

副时钟操作期间(PCC.CLS 位=1)或者通过设置 PCC.MCK 位为 1 来停止主时钟，到主时钟开始操作之前 (PCC.CLS 位=0)监控器停止操作。直到主时钟开始操作监控器才自动开始操作。

图 23-5 主时钟停止时的操作(任意时刻)

**(5) 当 CPU 使用内部振荡时钟时的操作(CCLS.CCLSIF 位 = 1)**

当 CCLSIF 位设为 1 时监控器的操作不停止，即时 CLME 位被设为 1。

第二十四章 低电压检测器 (LVI)

24.1 功能

低电压检测电路 (LVI) 有以下功能。

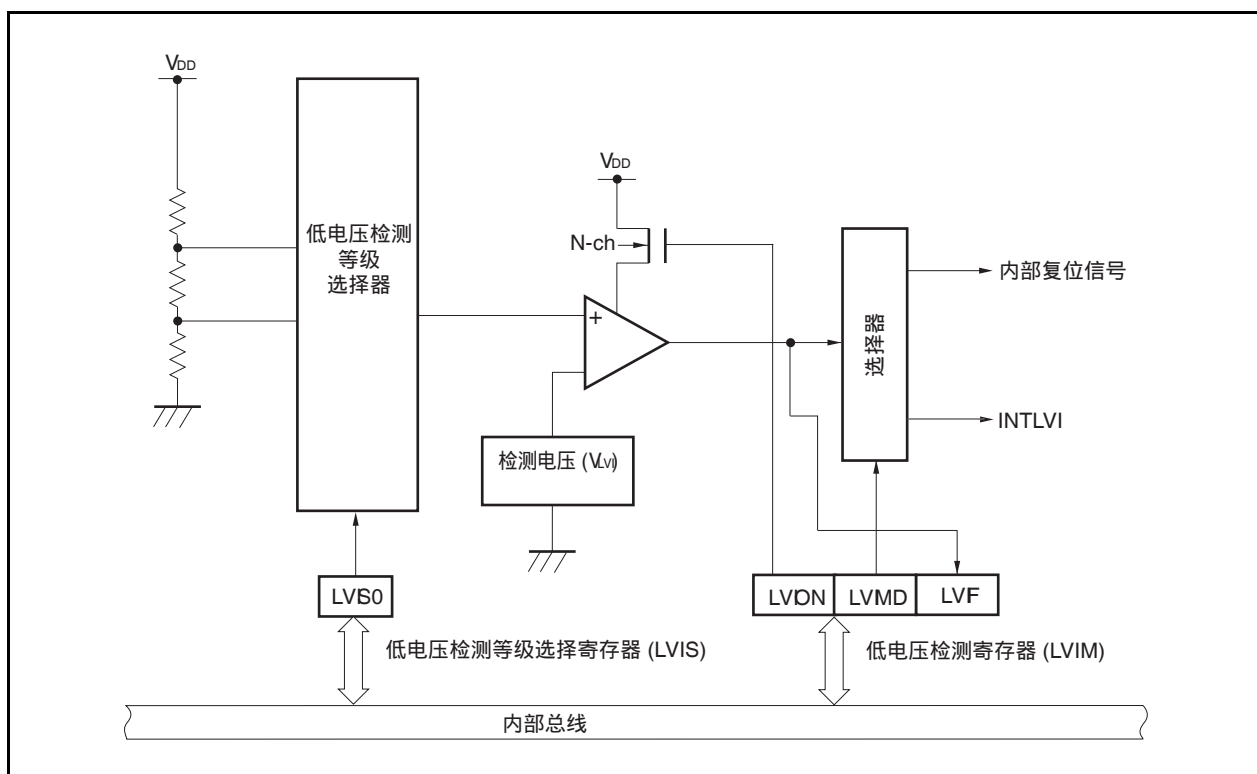
- LVI 电路比较供电电压 (V_{DD}) 和检测电压 (V_{LVI})，当 $V_{DD} < V_{LVI}$ 时产生内部复位信号或内部中断信号。
- 可由软件改变供电电压的检测等级 (2 个等级)。
- 可由软件选择中断或复位功能。
- 在 STOP 模式下可操作。

当低电压检测电路用作产生复位时，如果复位产生，复位控制标志寄存器 (RESF) 的第 0 位 (LVIRF) 被置 1。如欲了解 RESF 的详细信息，请参见 22.2 检测复位源的寄存器。

24.2 配置

低电压检测电路框图下所示。

图 24-1 低电压检测电路框图



24.3 寄存器

低电压检测电路由以下寄存器控制。

- 低电压检测寄存器(LVIM)
- 低电压检测等级选择寄存器(LVIS)

(1) 低电压检测寄存器 (LVIM)

此寄存器是一个特殊寄存器。它只有在特殊的顺序组合时才可以写(请看 **3.4.7 特殊寄存器**)。

它用来允许或禁止低电压检测，并设置其操作模式。

此寄存器可以进行 8 位或 1 位读写。然而，LVIF 位只读。

复位后: 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	低电压检测操作允许或禁止						
	0	禁止操作						
	1	允许操作						
	LVIMD	低电压检测操作模式的选择						
	0	当供电电压 (V_{DD}) < 检测电压 (V_{LVI}) 时产生中断请求信号 INTLVI。						
	1	当供电电压 (V_{DD}) < 检测电压 (V_{LVI}) 时产生内部复位信号 LVIREs。						
	LVIF ^{注 2}	低电压检测标志						
	0	供电电压 (V_{DD}) > 检测电压 (V_{LVI})，或当禁止操作时						
	1	供电电压 (V_{DD}) < 检测电压 (V_{LVI})						

- 注
1. 由低电压检测复位: 82H
由其它源复位: 00H
 2. 当 LVION 位 = 1 和 LVIMD 位 = 0 时, LVIF 标志的值会作为中断请求信号 INTLVI 被输出。

- 注意事项
1. 当 LVION 和 LVIMD 位为 1 时, 除非产生 LVI 复位以外的复位请求时, 否则低电平检测电路不能停止。
 2. 当 LVION=1 时, LVI 电路中的比较器开始操作。从设置 LVION=1 到读取 LVIF 确定电压, 请使用软件等待 0.2 ms 或更长的时间。
 3. 一定要将位 6 ~ 2 清 0。

(2) 低电压检测等级选择寄存器 (LVIS)

该寄存器用于选择低电压检测等级。

此寄存器可以进行 8 位或 1 位读写。

复位后: 注	R/W	地址: FFFFF891H	7	6	5	4	3	2	1	0
LVIS			0	0	0	0	0	0	0	LVIS0

LVIS0	检测等级
0	3.0 V \pm 0.15 V
1	2.85 V \pm 0.15 V (禁止设置)

注 由电压检测引起的复位: 保留
由其它源引起的复位: 00H

- 注意事项 1. 当 LVION 和 LVIMD 位为 1 时, 此寄存器不能写, 直到产生除 LVI 复位以外的复位请求时。
2. 一定要将位 7 ~ 1 清 0。

(3) 内部 RAM 数据状态寄存器 (RAMS)

RAMS 寄存器是一个特殊寄存器。它只有在特殊的顺序组合时才可以写(请看 3.4.7 特殊寄存器)。

此寄存器是一个标志寄存器, 用来指示内部 RAM 是否有效。

此寄存器可以进行 8 位或 1 位读写。

RAMF 位的设置/清除条件如下所示。

- 设置条件:
 - 检测到比设定电压低的电压
 - 由指令设置
 - WDT2 和 CLM 产生复位信号
 - 当访问 RAM 时, 产生复位信号
 - 当访问内部 RAM 时, 由 $\overline{\text{RESET}}$ 引脚产生复位信号
- 清除条件:
 - 以特定顺序写入 0

复位后: 01H [*]	R/W	地址: FFFFF892H	7	6	5	4	3	2	1	<0>
RAMS			0	0	0	0	0	0	0	RAMF

RAMF	内部 RAM 数据有效/无效
0	有效
1	无效

注 此寄存器若在 $\overline{\text{RESET}}$ 引脚输入(只在 RAM 访问时), 看门狗定时器 2 溢出或者时钟监视器引起复位时会被设置为 01H。而由其它源引起的复位, 此寄存器数值保留。

24.4 操作

依据 LVIM.LVIMD 位的设置，产生中断请求信号 (INTLVI) 或者内部复位信号。
下面和时序图一起阐明每一个操作。

24.4.1 用作内部复位信号

<开始操作>

<1> 屏蔽 LVI 中断。

<2> 由 LVIS.LVIS0 位选择检测的电压。

<3> 设置 LVIM.LVION 位为 1 (允许操作)。

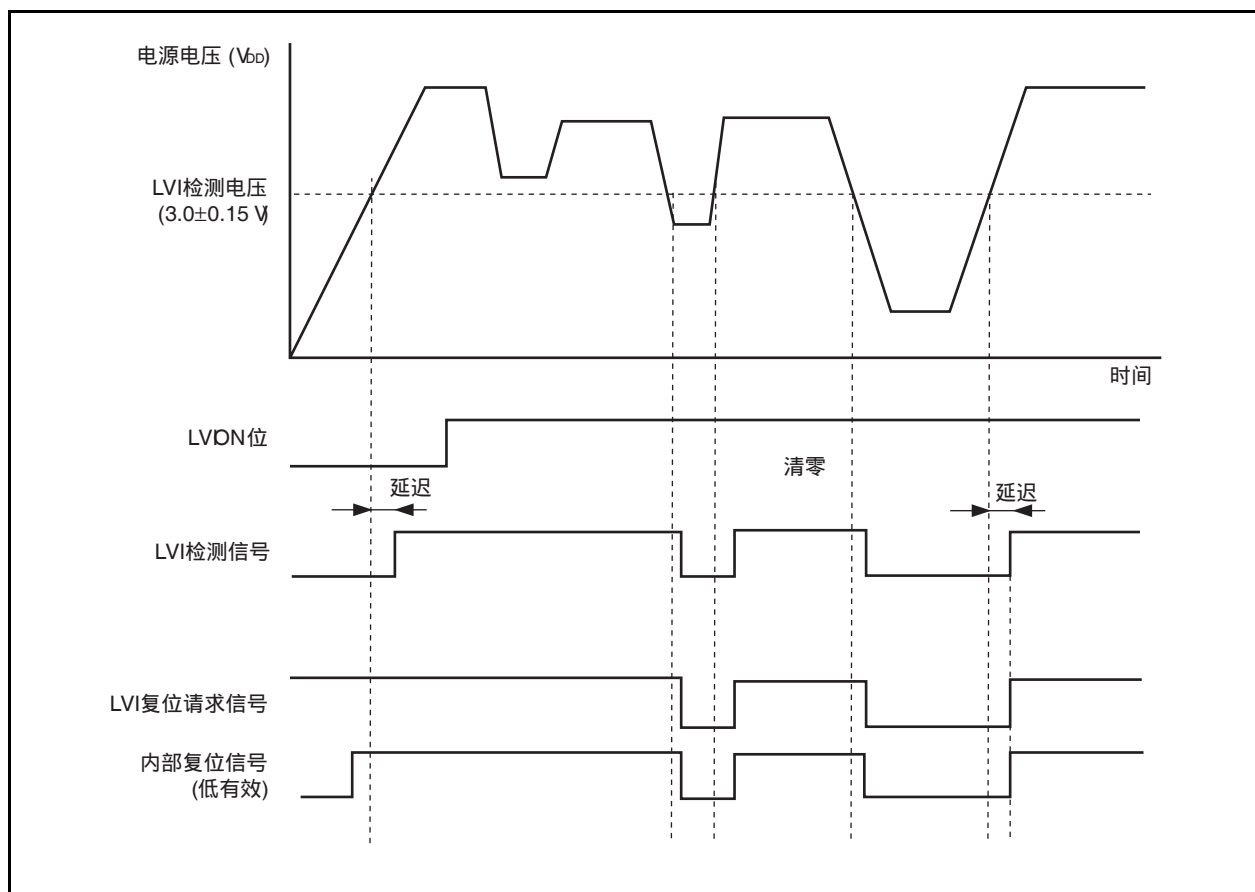
<4> 由软件插入 0.2 ms (max.)或更长的等待周期。

<5> 由 LVIM.LVIF 位，检验是否供电源电压 > 检测电压。

<6> 设置 LVIMD 位为 1 (产生内部复位信号)。

注意事项 如果 LVIMD 位设置为，那么 LVIM 和 LVIS 寄存器的内容不能改变，直到产生 LVI 之外的复位请求。

图 24-2 低电压检测点路的操作时序 (LVIMD 位 = 1)



24.4.2 用作中断

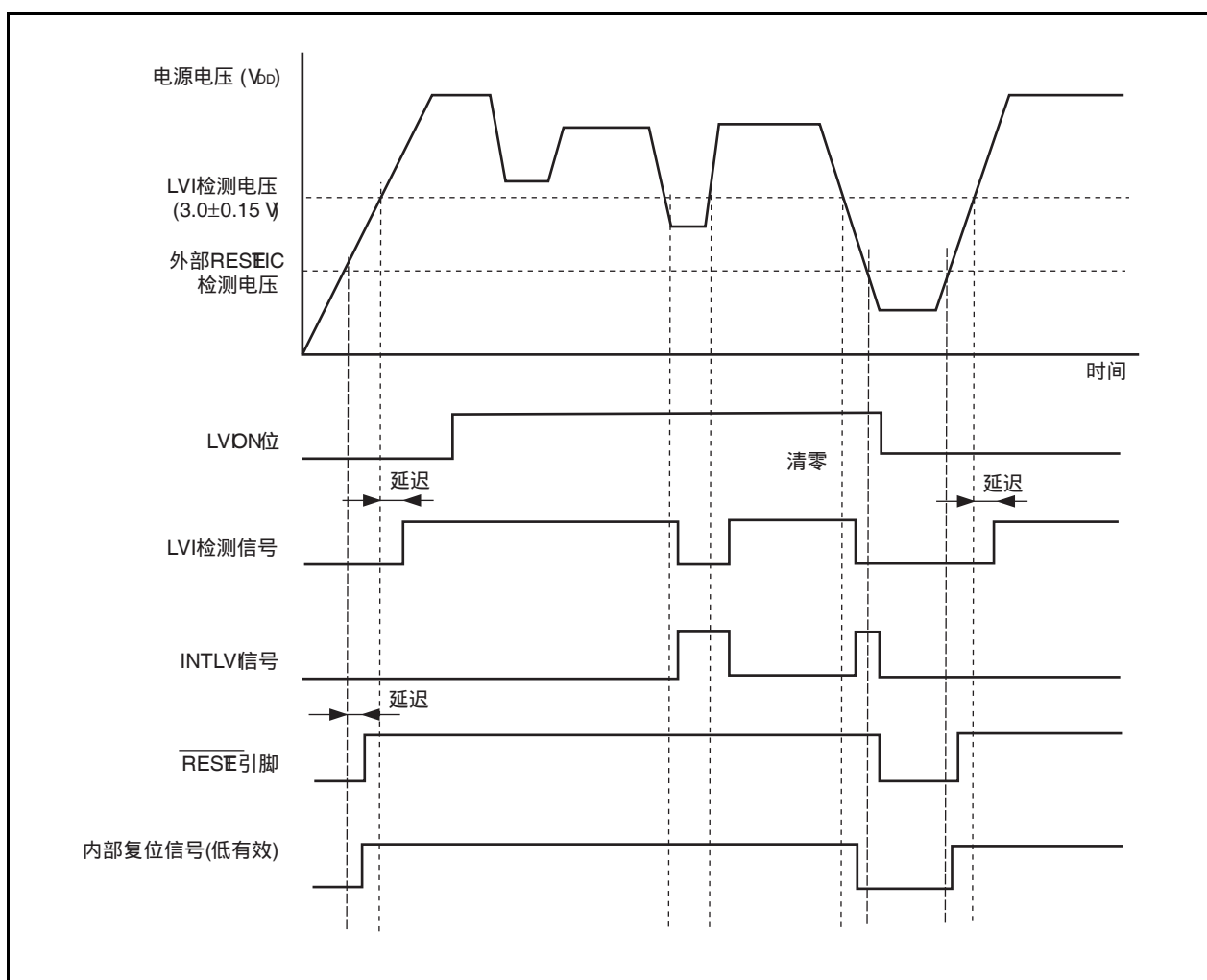
<开始操作>

- <1> 屏蔽 LVI 中断。
- <2> 由 LVIS.LVIS0 位选择检测的电压。
- <3> 设置 LVIM.LVION 位为 1 (允许操作)。
- <4> 由软件插入 0.2 ms (max.)或更长的等待周期。
- <5> 由 LVIM.LVIF 位，检验是否供电电压 > 检测电压。
- <6> 清除 LVI 的中断请求标志。
- <7> 打开 LVI 中断。

<停止操作>

将 LVION 位清 0。

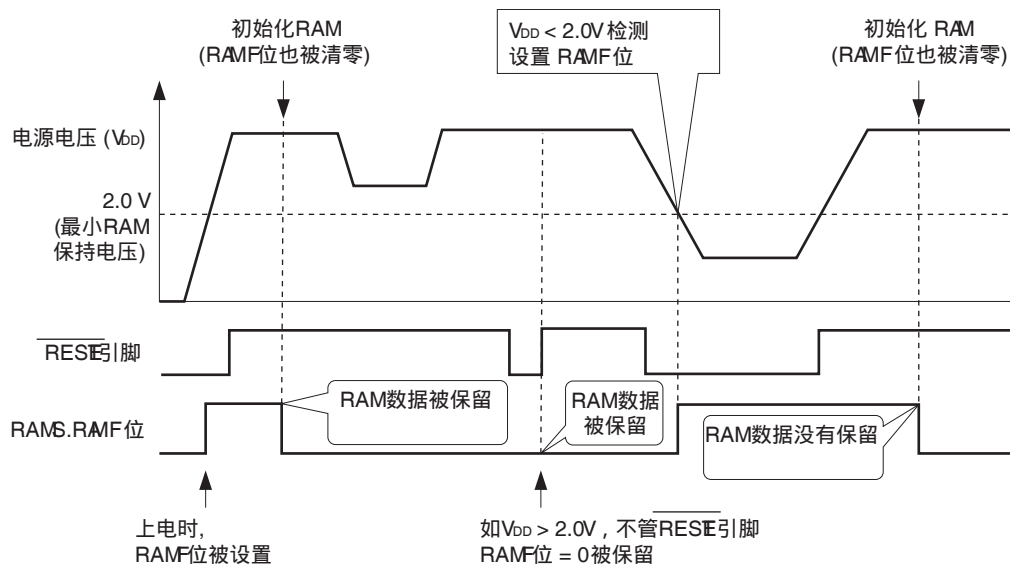
图 24-3 低电压检测电路的操作时序 (LVIM 位 = 0)



24.5 RAM 保持电压检测操作

比较供电电压和检测电压。当其低于检测电压时(包括上电)，RAMS.RAMF 位设置为 1。

图 24-4 RAM 保持电压检测功能的操作时序



- 备注**
1. 如果供电电压将低到小于最小的 RAM 保持电压 (2.0 V (TYP.)), 那么 RAMF 位设置为 1。
 2. RAMF 位的操作不受 RESET 引脚状态的影响。

24.6 仿真功能

当使用在线仿真器时，RAM 保持标志(RAMS.RAMF 位)的操作可以通过操作调试器上的 PEMU1 寄存器进行伪控制 和仿真。
此寄存器只有在仿真模式下有效，正常模式下无效。

(1) 外围仿真寄存器 1 (PEMU1)

复位后：00H

R/W

地址：FFFF9FEH

	7	6	5	4	3	2	1	0
PEMU1	0	0	0	0	0	EVARAMIN	0	0

EVARAMIN	RAM 保持电压检测信号的假定
0	不检测低于 RAM 保持电压的电压。
1	检测低于 RAM 保持电压的电压 (设置 RAMF 标志)。

注意事项 此位不能自动清除。

- [用法]
- 当使用线上仿真器时，RAMF 的伪仿真通过重写调试器上的寄存器实现。
- <1> CPU 停止 (CPU 操作停止)。
 - <2> 通过使用寄存器写命令设置 EVARAMIN 位为 1。
通过设置 EVARAMIN 位为 1，RAMF 位硬件上设置为 1。(内部 RAM 数据无效)。
 - <3> 再次通过寄存器写命令将 EVARAMIN 位清零。
除非进行此操作(清 EVARAMIN 位为 0)，否则 RAMF 位不能由 CPU 操作指令清 0。
 - <4> 运行 CPU 并恢复仿真。

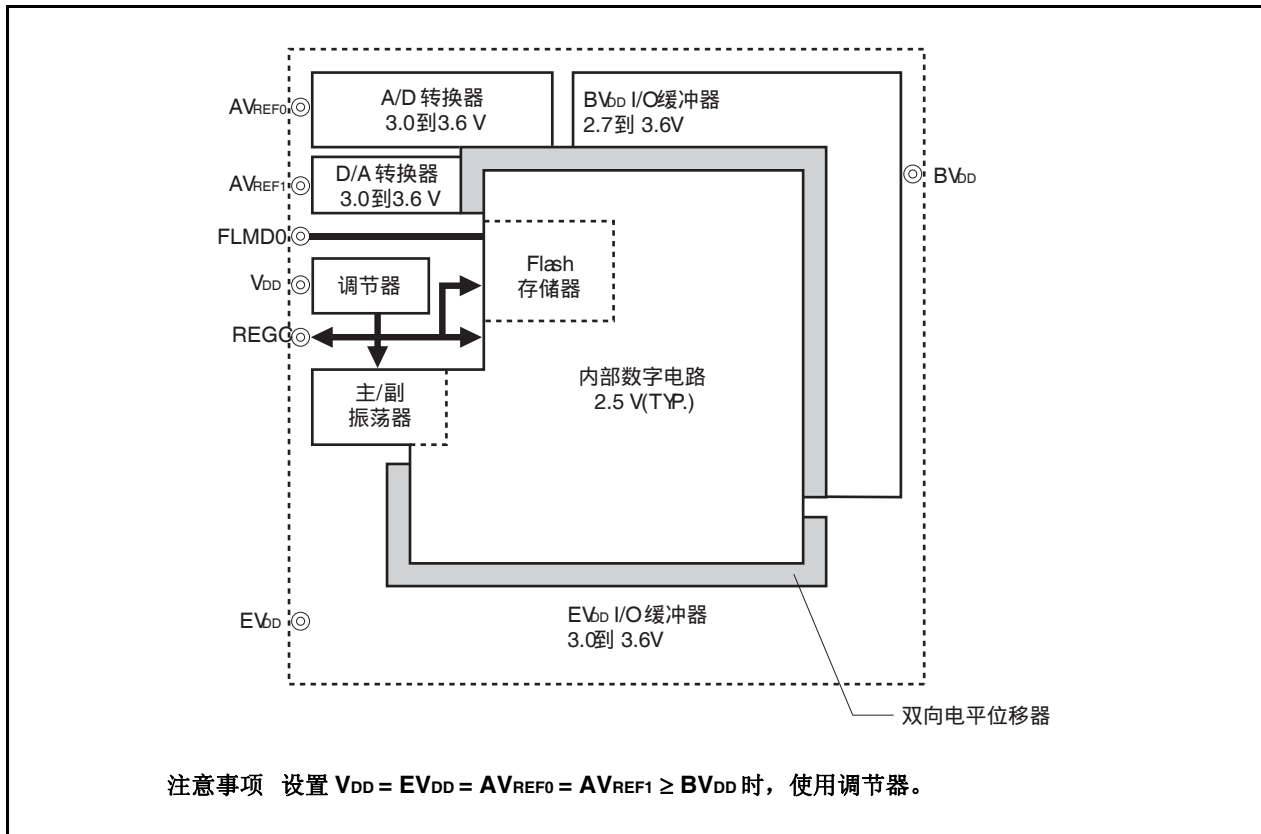
第二十五章 调节器

25.1 概论

V850ES/JG2 包括一个可以降低能耗和噪声的调节器。

此调节器供应一个分阶下降的供电源 V_{DD} 给振荡器模块和内部逻辑电路(除了 A/D 转换器, D/A 转换器和输出缓冲器)。调节器输出电压设置为 2.5 V (TYP.)。

图 25-1 稳压器



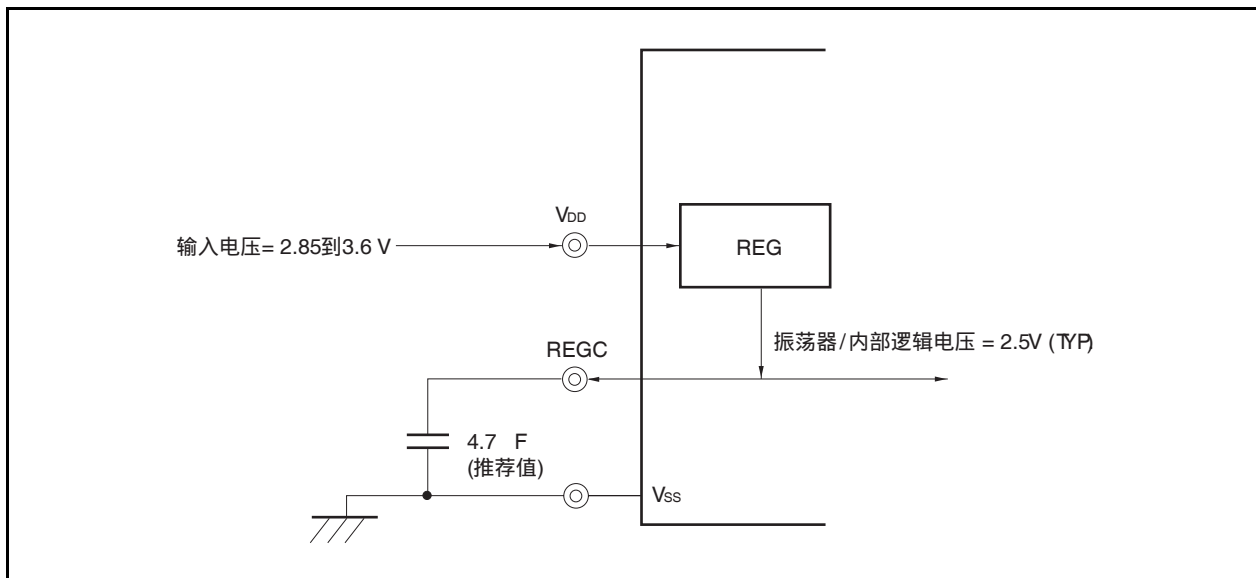
25.2 操作

产品的调节器总是运行于各种模式下(正常操作模式, HALT 模式, IDLE1 模式, IDLE2 模式, STOP 模式或者复位)。

一定要给 REGC 引脚连接一个电容(4.7 μF (推荐值))来稳定调节器输出。

调节器引脚连接方法的方框图如下所示。

图 25-2 REGC 引脚连接



第二十六章 FLASH 存储器

V850ES/JG2 集成了 flash 存储器

- μ PD70F3715: 128 KB flash 存储器
- μ PD70F3716: 256 KB flash 存储器
- μ PD70F3717: 384 KB flash 存储器
- μ PD70F3718: 512 KB flash 存储器
- μ PD70F3719: 640 KB flash 存储器

Flash 存储器版本对于开发环境与批量生产应用来说提供了如下的优势条件。

- 在 V850ES/JG2 被安装到目标系统之后可以对软件进行更改。
- 量产后的数据调整。
- 在不同模式的小批量生产中依照规格区分软件。
- 简化存货管理。
- 出库后的软件更新。

26.1 特性

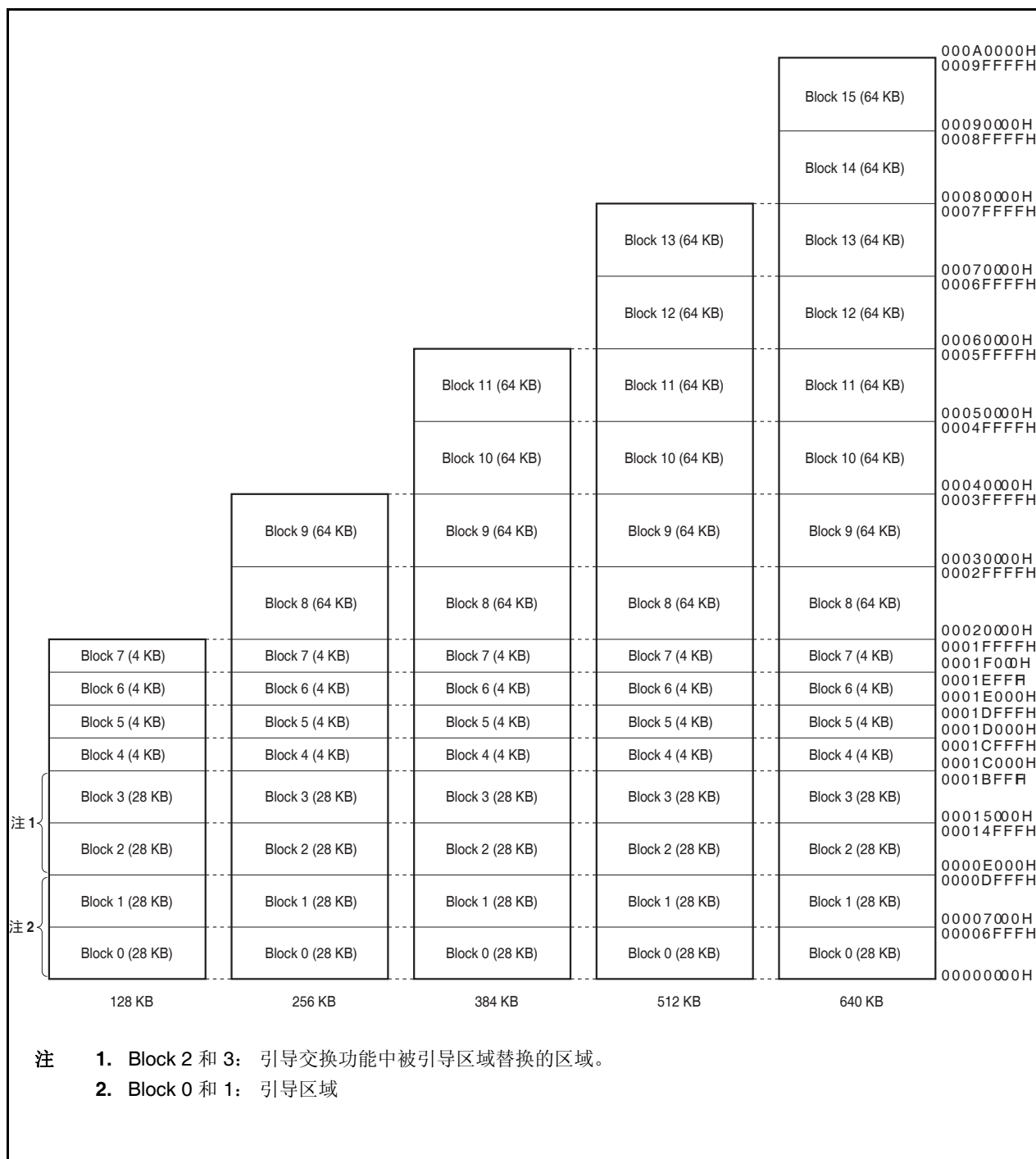
- 4 字节 /1 个时钟访问 (当取指令时)
- 容量: 640/512/384/256/128 KB
- 写入电压: 单电压擦除/写入
- 重写方法
 - 通过与专用 flash 编程器的串行接口进行通信重写 (在板/离板编程)
 - 通过用户程序重写 flash 存储器(自编程)
- 支持 Flash 存储器禁止写入功能 (安全功能)
- 由使用引导交换功能的自编程安全重写整个 flash 存储器区域
- 自编程期间可以响应中断。

26.2 存储器配置

V850ES/JG2 的内部 flash 存储器区域被分割为 16/14/12/10/8 个 block，并且可以 block 为单位进行编程/擦除。所有 block 也可以一次被擦除。

当使用引导交换功能时，物理存储器地址 block 2 和 3 代替 block 0 和 1。引导交换功能的更多细节，参见 26.5 通过自编程重写。

图 26-1. Flash 存储器影射图



26.3 功能概述

V850ES/JG2 的内部 flash 存储器可使用专用 flash 编程器的重写功能重写，不论 V850ES/JG2 是否已经被安装到目标系统 (离板/在板编程)。

另外，也支持禁止对已写入到内部 flash 存储器中的用户程序重写的安全功能，因此程序不能由未经授权的人修改。

使用用户程序（自编程）的重写功能对于那些产品出库后有可能发生改变的目标系统中的应用比较理想。也支持安全重写整个 flash 存储器的引导交换功能。另外，自编程期间支持中断服务，因此 flash 存储器可以在不同的条件下被重写，例如当与外围设备通讯时。

表 26-1. 重写方法

重写方法	功能概述	操作模式
在板编程	安装到目标系统后 Flash 存储器可以被重写，必须使用专用 Flash 编程器。	Flash 存储器编程模式
离板编程	在安装到目标系统之前，可使用专用 Flash 编程器和专用编程适配器 (FA 系列) 重写 Flash 存储器。	
自编程	Flash 存储器可以通过执行用离板/在板编程方法预先写入到 flash 存储器中的用户程序进行重写。(自编程期间，不能取指令且不能访问内部 flash 存储器区域的数据。因此，重写程序必须预先发送到内部 RAM 或外部存储器)。	正常操作模式

备注 FA 系列是 Naito Densei Machida Mfg. Co., Ltd. 的产品。

表 26-2. 基本功能

功能	功能概述	支持 (√: 支持, ×: 不支持)	
		在板/离板编程	自编程
Block 擦除	擦除指定的存储器 block。	√	√
芯片擦除	一次擦除整个存储器区域。	√	×
写入	写入到指定地址, 并且校验检查写入级别是否安全地完成。	√	√
校验/校验和	比较 flash 存储器中读出的数据与从 flash 编程器发送的数据。	√	× (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	使用 block 擦除命令, chip 擦除命令, 编程命令可以被禁止。	√	× (只有当设置由允许改为禁止时才支持)

如下列表为安全功能。出库后默认允许 block 擦除命令禁止, chip 擦除命令禁止, 编程命令禁止功能, 可通过使用在板/离板编程重写进行安全设置。 每个安全功能均可与其他安全功能联合同时使用。

表 26-3. 安全功能

功能	功能概述
Block 擦除命令禁止	所有 block 禁止执行 block 擦除命令。禁止的设置可以通过芯片擦除命令的执行进行初始化。
芯片擦除命令禁止	禁止执行 block 擦除和芯片擦除命令。一旦禁止被设置, 由于芯片擦除命令不能执行因此禁止设置不能被初始化。
编程命令禁止	所有的 block 都禁止编程和 block 擦除命令。禁止的设置可以通过芯片擦除命令的执行进行初始化。
读取命令禁止	不支持 (永久禁止)。
引导区域重写禁止	不支持。

<R>

表 26-4. 安全设置

功能	设置各安全时的擦除，写入，读取操作 (√: 可执行，×: 不可执行，-: 不支持)		安全设置的注意要点	
	在板/离板编程	自编程	在板/离板编程	自编程
Block 擦除命令禁止	Block 擦除命令: × 芯片擦除命令: √ 编程命令: √ 读取命令: ×	Block 擦除 (FlashBlock 擦除): √ 芯片擦除: - 写入 (Flash 字写入): √ 读取 (Flash 字读取): √	禁止的设置可以通过芯片擦除命令的执行进行初始化。	只有当设置由允许改为禁止时才支持
芯片擦除命令禁止	Block 擦除命令: × 芯片擦除命令: × 编程命令: √ ^註 读取命令: ×	Block 擦除 (FlashBlock 擦除): √ 芯片擦除: - 写入 (Flash 字写入): √ 读取 (Flash 字读取): √	禁止设置不能被初始化。	
编程命令禁止	Block 擦除命令: × 芯片擦除命令: √ 编程命令: × 读取命令: ×	Block 擦除 (FlashBlock 擦除): √ 芯片擦除: - 写入 (Flash 字写入): √ 读取 (Flash 字读取): √	禁止的设置可以通过芯片擦除命令的执行进行初始化。	

注 这种情况下，由于擦除命令是无效的，与已经被写入到 flash 存储器中的数据不同的数据不能被写入。

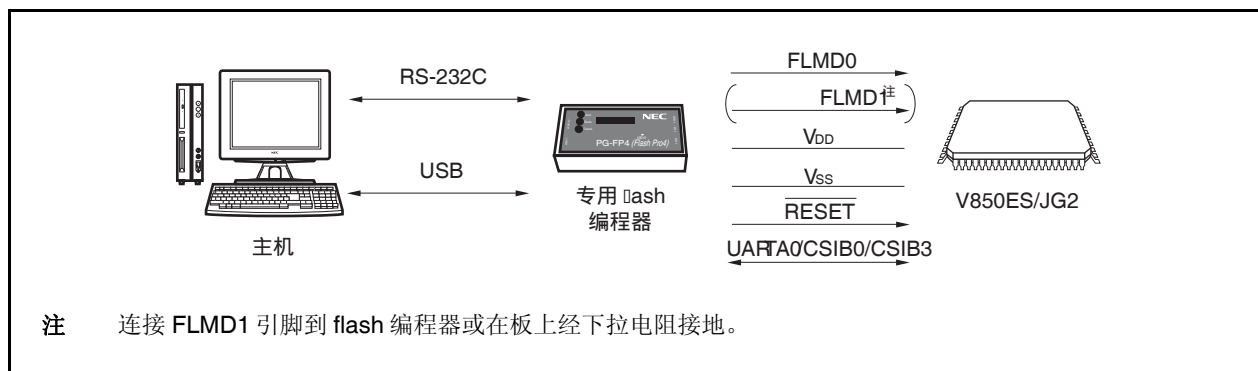
26.4 通过专用 Flash 编程器重写

在 V850ES/JG2 安装到目标系统（在板编程）后可以通过专用 flash 编程器重写 flash 存储器。在设备安装到目标系统（离板编程）前也可以通过专用编程适配器（FA 系列）重写 flash 存储器。

26.4.1 编程环境

如下图所示为写程序到 V850ES/JG2 的 flash 存储器所需要的环境。

图 26-2. 写程序到 Flash 存储器所需要的环境



需要一个用于控制专用 flash 编程器的主机。

UARTA0, CSIB0, 或 CSIB3 作为专用 flash 编程器和 V850ES/JG2 的接口执行写入，擦除等。离板写入需要专用编程适配器（FA 系列）。

- FA-70F3719GC-8EA-MX (GC-8EA type) (已接线)
- FA-70F3717GF-JBT-MX (GF-JBT type) (已接线)
- FA-100GC-8EU-A (GC-8EA type) (未接线: 需要接线)
- FA-100GF-3BA-A (GF-JBT type) (未接线: 需要接线)

备注 FA 系列是 Naito Densei Machida Mfg. Co., Ltd 的产品。

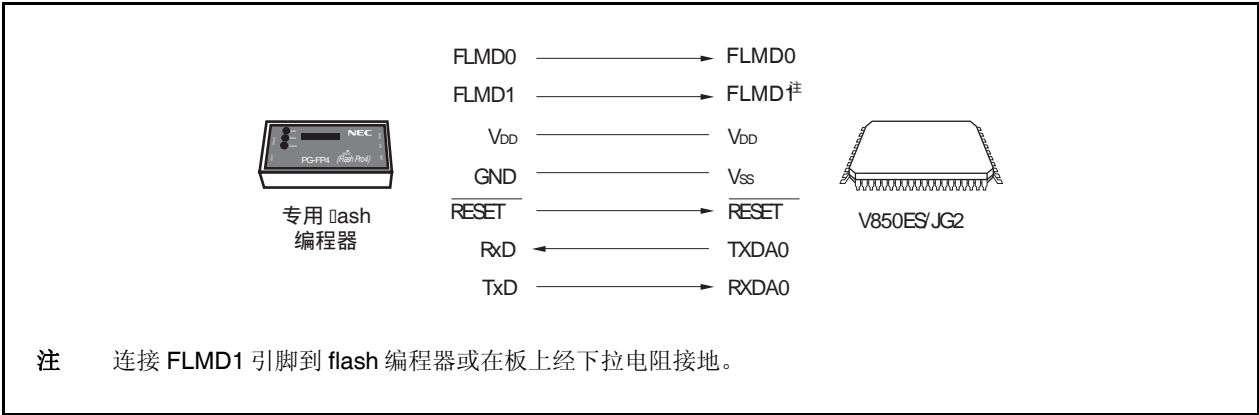
26.4.2 通信模式

V850ES/JG2 与专用 flash 编程器间的通信通过 V850ES/JG2 的串行接口 UARTA0, CSIB0, 或 CSIB3 完成。

(1) UARTA0

<R> 发送速率: 9,600, 19,200, 31,250, 38,400, 76,800, 153,600 bps
(不支持设置 57,600, 115,200, 或者 128,000 bps)

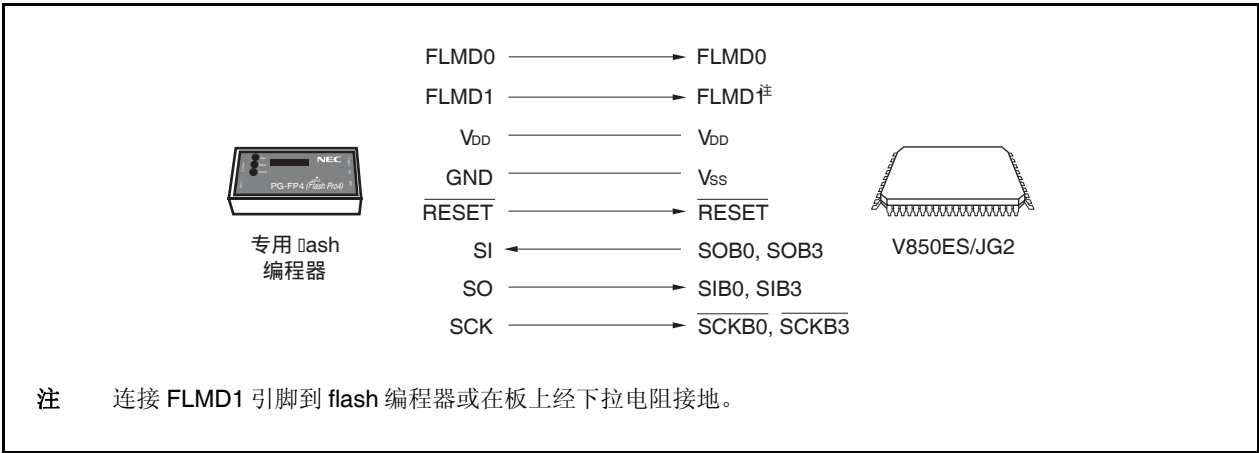
图 26-3. 与专用 flash 编程器通信(UARTA0)



(2) CSIB0, CSIB3

串行时钟: 2.4 kHz ~ 2.5 MHz (最高有效位先行)

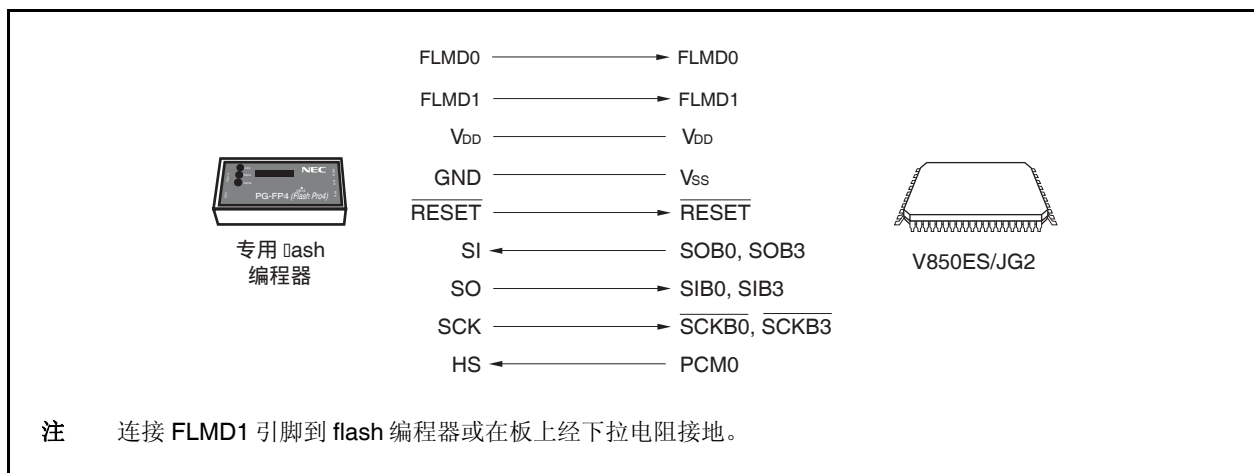
图 26-4. 与专用 flash 编程器通信(CSIB0, CSIB3)



(3) CSIB0 + HS, CSIB3 + HS

串行时钟: 2.4 kHz ~ 2.5 MHz (最高有效位先行)

图 26-5. 与专用 flash 编程器通信(CSIB0 + HS, CSIB3 + HS)



专用 flash 编程器输出发送时钟，V850ES/JG2 作为从设备操作。

当 PG-FP4 用作专用 flash 编程器时，产生如下信号到 V850ES/JG2。更多细节，参见 PG-FP4 用户手册 (U15260E)。

表 26-5. 专用 Flash 编程器的信号连接(PG-FP4)

PG-FP4			V850ES/JG2	连接处理		
信号名称	I/O	引脚功能	引脚名称	UARTA0	CSIB0, CSIB3	CSIB0 + HS, CSIB3 + HS
FLMD0	输出	写入允许/禁止	FLMD0	○	○	○
FLMD1	输出	写入允许/禁止	FLMD1	○ ^{注 1}	○ ^{注 1}	○ ^{注 1}
VDD	—	VDD 电压产生/电压监测	VDD	○	○	○
GND	—	接地	VSS	○	○	○
CLK	输出	时钟输入到 V850ES/JG2	X1, X2	× ^{注 2}	× ^{注 2}	× ^{注 2}
RESET	输出	复位信号	RESET	○	○	○
SI/RxD	输入	接收信号	SOB0, SOB3/ TXDA0	○	○	○
SO/TxD	输出	发送信号	SIB0, SIB3/ RXDA0	○	○	○
SCK	输出	发送时钟	SCKB0, SCKB3	×	○	○
HS	输入	用于 CSIB0 + HS, CSIB3 + HS 通信的握手信号	PCM0	×	×	○

- 注 1. 按图 26-6 和 26-7 中所示连接这些引脚，或在板上经下拉电阻接地。
2. 无法由 flash 编程器的 CLK 引脚提供时钟。板上需要有振荡器并提供时钟。

备注 ○: 必须连接。
×: 不必连接。

表 26-6. V850ES/JG2 Flash 写入适配器接线 (FA-100GF-3BA-A, FA-100GC-8EU-A) (1/2)

Flash 编程器 (PG-FP4) 连接引脚			FA 板引脚 名称	使用 CSIB0 + HS			使用 CSIB0			使用 UARTA0		
信号名 称	I/O	引脚功能		引脚名称	引脚编号		引脚名称	引脚编号		引脚名称	引脚编号	
					GF	GC		GF	GC		GF	GC
SI/RxD	输入	接收信号	SI	P41/SOB0/ SCL01	25	23	P41/SOB0/ SCL01	25	23	P30/TXDA0/ SOB4	27	25
SO/TxD	输出	发送信号	SO	P40/SIB0/ SDA01	24	22	P40/SIB0/ SDA01	24	22	P31/RXDA0/ INTP7/SIB4	28	26
SCK	输出	发送时钟	SCK	P42/SCKB0	26	24	P42/SCKB0	26	24	不需要	—	—
CLK	输出	向 V850ES/JG2 提供时钟	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	16	14	RESET	16	14	RESET	16	14
FLMD0	输出	写入电压	FLMD0	FLMD0	10	8	FLMD0	10	8	FLMD0	10	8
FLMD1	输出	写入电压	FLMD1	PLD5/AD5/ FLMD1	78	76	PLD5/AD5/ FLMD1	78	76	PLD5/AD5/ FLMD1	78	76
HS	输入	用于 CSIO + HS 通信的握手信号	RESERVE/ HS	PCM0/WAIT	63	61	不需要	—	—	不需要	—	—
VDD	—	VDD 电压产生/ 电压监测	VDD	V _{DD}	11	9	V _{DD}	11	9	V _{DD}	11	9
				BV _{DD}	72	70	BV _{DD}	72	70	BV _{DD}	72	70
				EV _{DD}	36	34	EV _{DD}	36	34	EV _{DD}	36	34
				AV _{REF0}	3	1	AV _{REF0}	3	1	AV _{REF0}	3	1
				AV _{REF1}	7	5	AV _{REF1}	7	5	AV _{REF1}	7	5
GND	—	地	GND	V _{SS}	13	11	V _{SS}	13	11	V _{SS}	13	11
				AV _{SS}	4	2	AV _{SS}	4	2	AV _{SS}	4	2
				BV _{SS}	71	69	BV _{SS}	71	69	BV _{SS}	71	69
				EV _{SS}	35	33	EV _{SS}	35	33	EV _{SS}	35	33

- 注意事项 1. 确保 REGC 引脚经 4.7 μF 电容接地。
2. 时钟无法由 flash 编程器的 CLK 引脚提供。
板上需要有振荡器并提供时钟。

备注 GF: 100 引脚塑封 QFP (14 × 20)
GC: 100 引脚塑封 LQFP (密间距) (14 × 14)

表 26-6. V850ES/JG2 Flash 写入适配器接线 (FA-100GF-3BA-A, FA-100GC-8EU-A) (2/2)

Flash 编程器 (PG-FP4) 连接引脚			FA 板引脚名称	使用 CSIB3 + HS			使用 CSIB3		
信号名称	I/O	引脚功能		引脚名称	引脚编号		引脚名称	引脚编号	
					GF	GC		GF	GC
SI/RxD	输入	接收信号	SI	P911/A11/SOB3	56	54	P911/A11/SOB3	56	54
SO/TxD	输出	发送信号	SO	P910/A10/SIB3	55	53	P910/A10/SIB3	55	53
SCK	输出	发送时钟	SCK	P912/A12/SCKB3	57	55	P912/A12/SCKB3	57	55
CLK	输出	向 V850ES/JG2 提供时钟	X1	不需要	—	—	不需要	—	—
			X2	不需要	—	—	不需要	—	—
/RESET	输出	复位信号	/RESET	RESET	16	14	RESET	16	14
FLMD0	输出	写入电压	FLMD0	FLMD0	10	8	FLMD0	10	8
FLMD1	输出	写入电压	FLMD1	PLD5/AD5/FLMD1	78	76	PLD5/AD5/FLMD1	78	76
HS	输入	用于 CSIO + HS 通信的握手信号	RESERVE/HS	PCM0/WAIT	63	61	不需要	—	—
VDD	—	VDD 电压产生/电压监测	VDD	V _{DD}	11	9	V _{DD}	11	9
				BV _{DD}	72	70	BV _{DD}	72	70
				EV _{DD}	36	34	EV _{DD}	36	34
				AV _{REF0}	3	1	AV _{REF0}	3	1
				AV _{REF1}	7	5	AV _{REF1}	7	5
GND	—	地	GND	V _{SS}	13	11	V _{SS}	13	11
				AV _{SS}	4	2	AV _{SS}	4	2
				BV _{SS}	71	69	BV _{SS}	71	69
				EV _{SS}	35	33	EV _{SS}	35	33

- 注意事项
1. 确保 REGC 引脚经 4.7 μF 电容接地。
 2. 时钟无法由 flash 编程器的 CLK 引脚提供。
板上需要有振荡器并提供时钟。

备注

GF: 100 引脚塑封 QFP (14 × 20)

GC: 100 引脚塑封 LQFP (密间距) (14 × 14)

图 26-6. V850ES/JG2 Flash 写入适配器连接示例 (FA-100GF-3BA-A)
(在 CSIB0 + HS 模式下) (1/2)

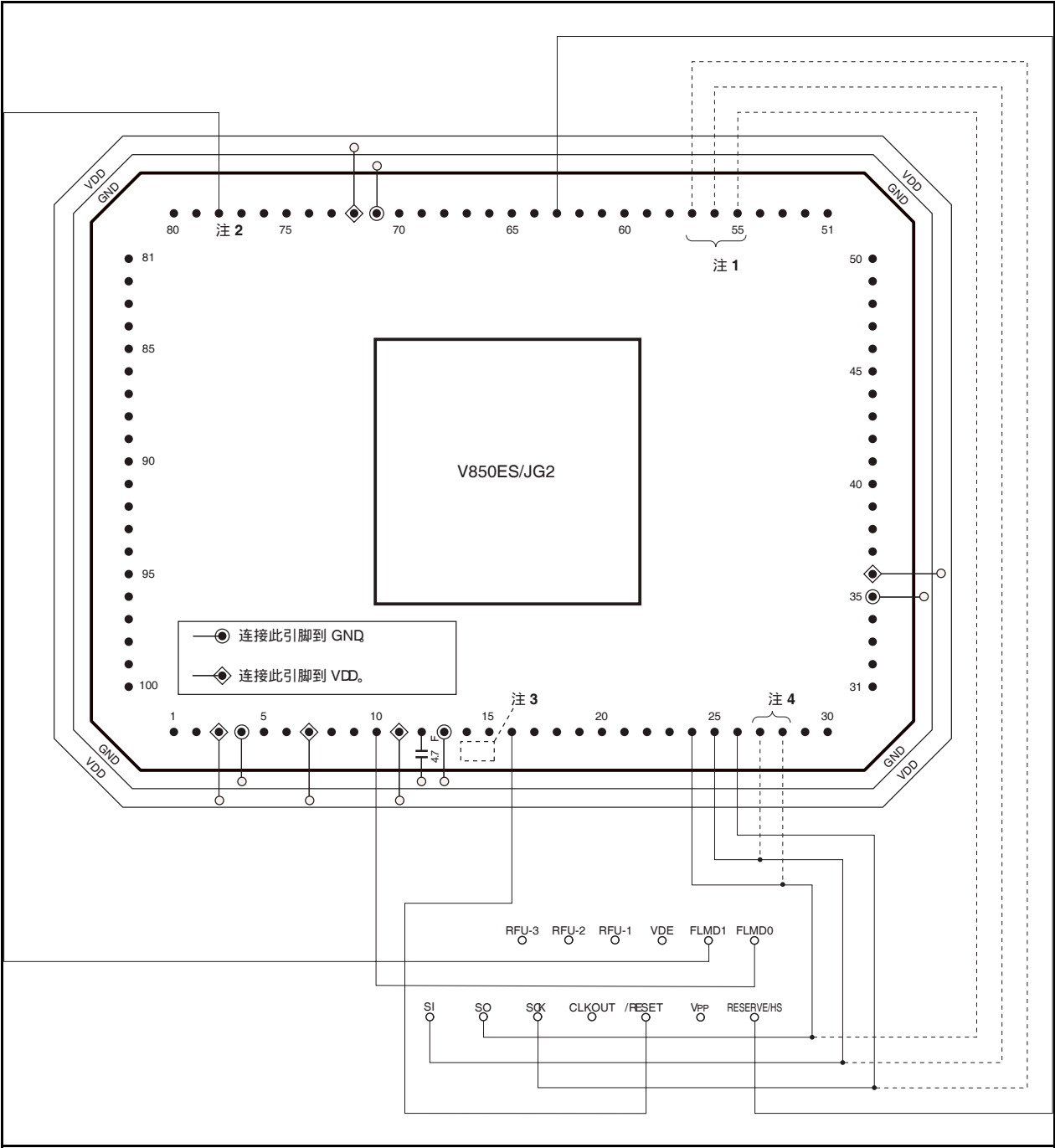
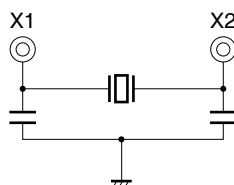


图 26-6. V850ES/JG2 Flash 写入适配器连接示例 (FA-100GF-3BA-A)
(在 CSIB0 + HS 模式下) (2/2)

- 注
1. 当 CSIB3 使用时的相应引脚。
 2. 按照下面所示连接 FLMD1 引脚，或在板上经下拉电阻将其接地。
 3. 在 flash 写入适配器上设置一个振荡器(虚线处所示)并且提供时钟。这有一个振荡器的示例。

示例:



4. 当使用 UARTA0 时的相应引脚。

注意事项 不要输入高电平到 $\overline{\text{DRST}}$ 引脚。

- 备注
1. 没有示意的引脚根据不使用引脚的操作处理(参见 2.3 I/O 引脚电路类型, I/O 缓冲供电电平, 和不使用引脚的连接)。
 2. 此适配器适用于 100 引脚塑封 QFP 封装。

图 26-7. V850ES/JG2 Flash 写入适配器连接示例 (FA-100GC-8EU-A)
(在 CSIB0 + HS 模式下) (1/2)

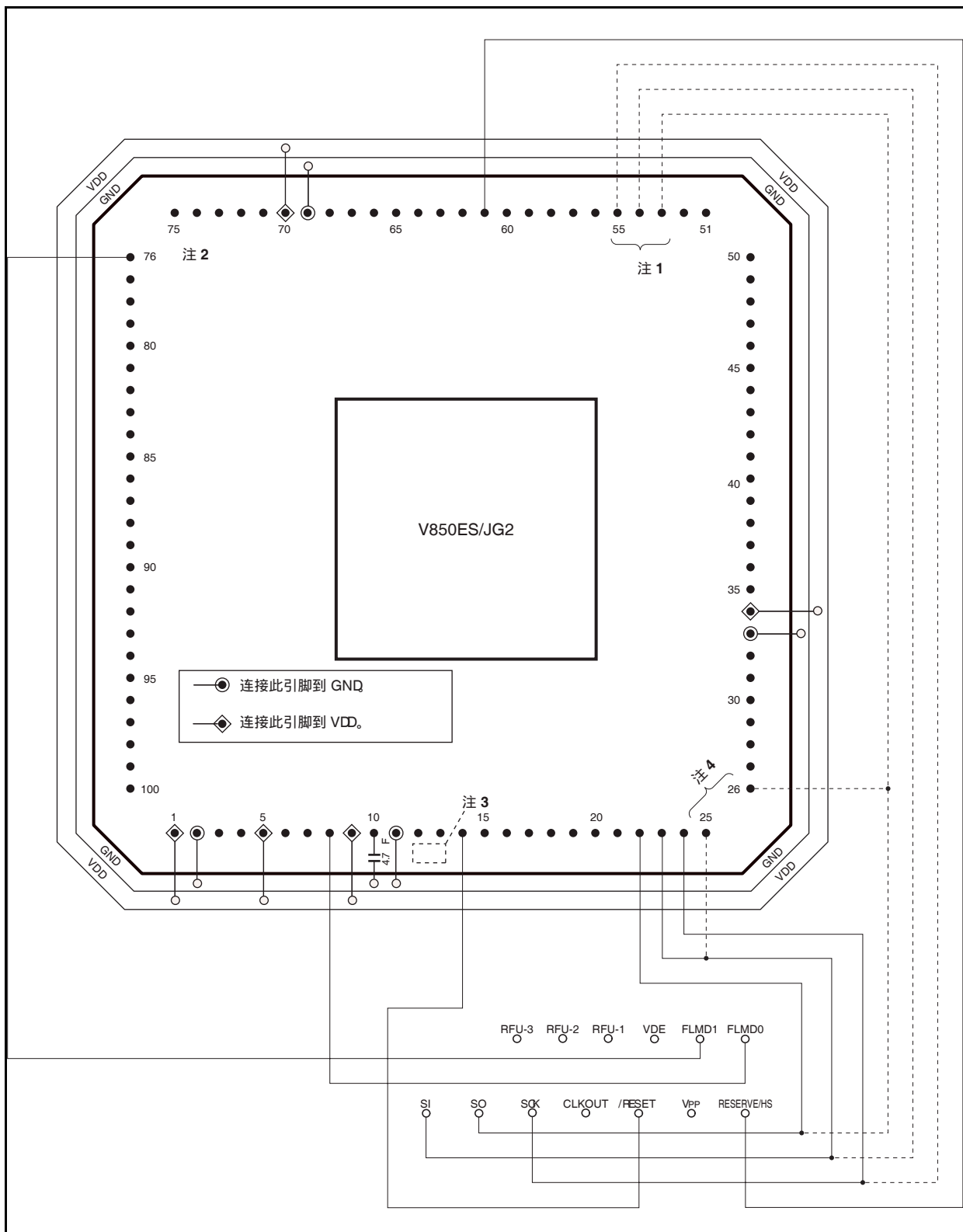
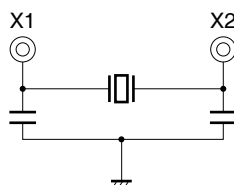


图 26-7. V850ES/JG2 Flash 写入适配器连接示例 (FA-100GC-8EU-A)
(在 CSIB0 + HS 模式下) (2/2)

- 注
1. 当 CSIB3 使用时的相应引脚。
 2. 按照下面所示连接 FLMD1 引脚，或在板上经下拉电阻将其接地。
 3. 在 flash 写入适配器上设置一个振荡器(虚线处所示)并且提供时钟。
这有一个振荡器的示例。

示例:



4. 当使用 UARTA0 时的相应引脚。

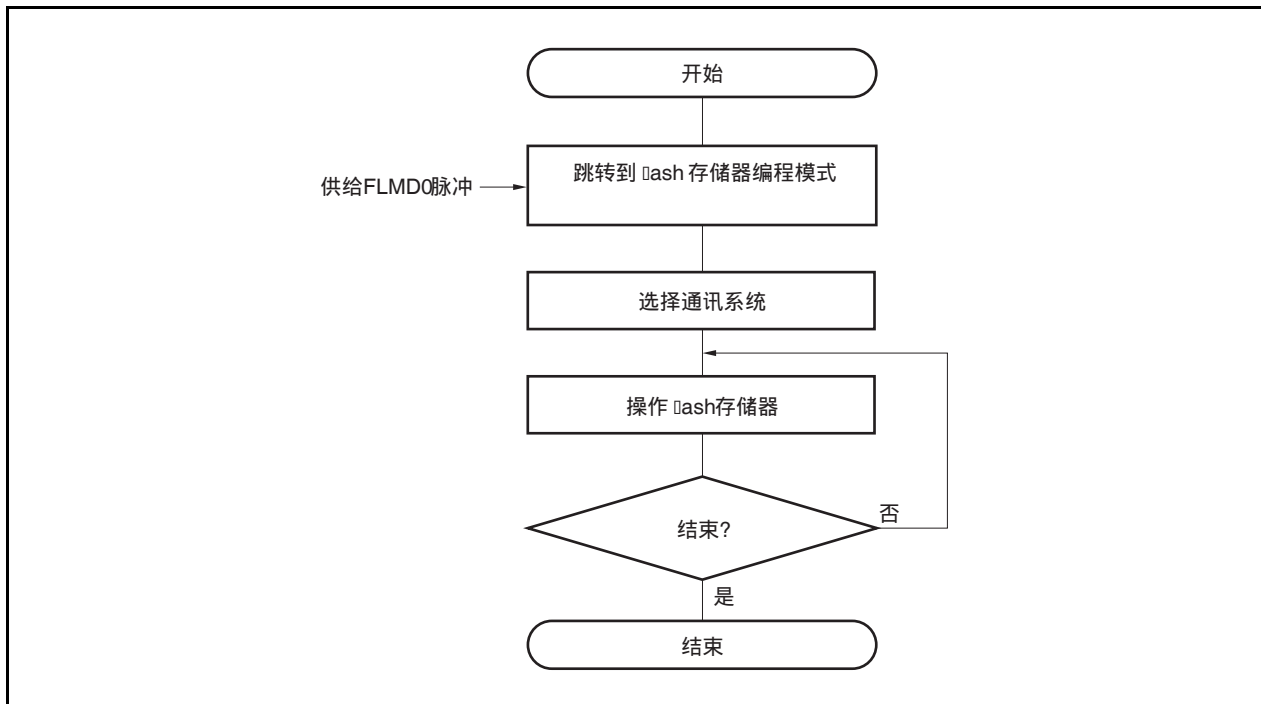
注意事项 不要输入高电平到 $\overline{\text{DRST}}$ 引脚。

- 备注
1. 没有示意的引脚根据不使用引脚的操作处理(参见 2.3 I/O 引脚电路类型, I/O 缓冲供电电平, 和不使用引脚的连接)。
 2. 此适配器适用于 100 引脚塑封 LQFP 封装。

26.4.3 Flash 存储器控制

下图显示了 Flash 存储器操作过程。

图 26-8. Flash 存储器操作过程

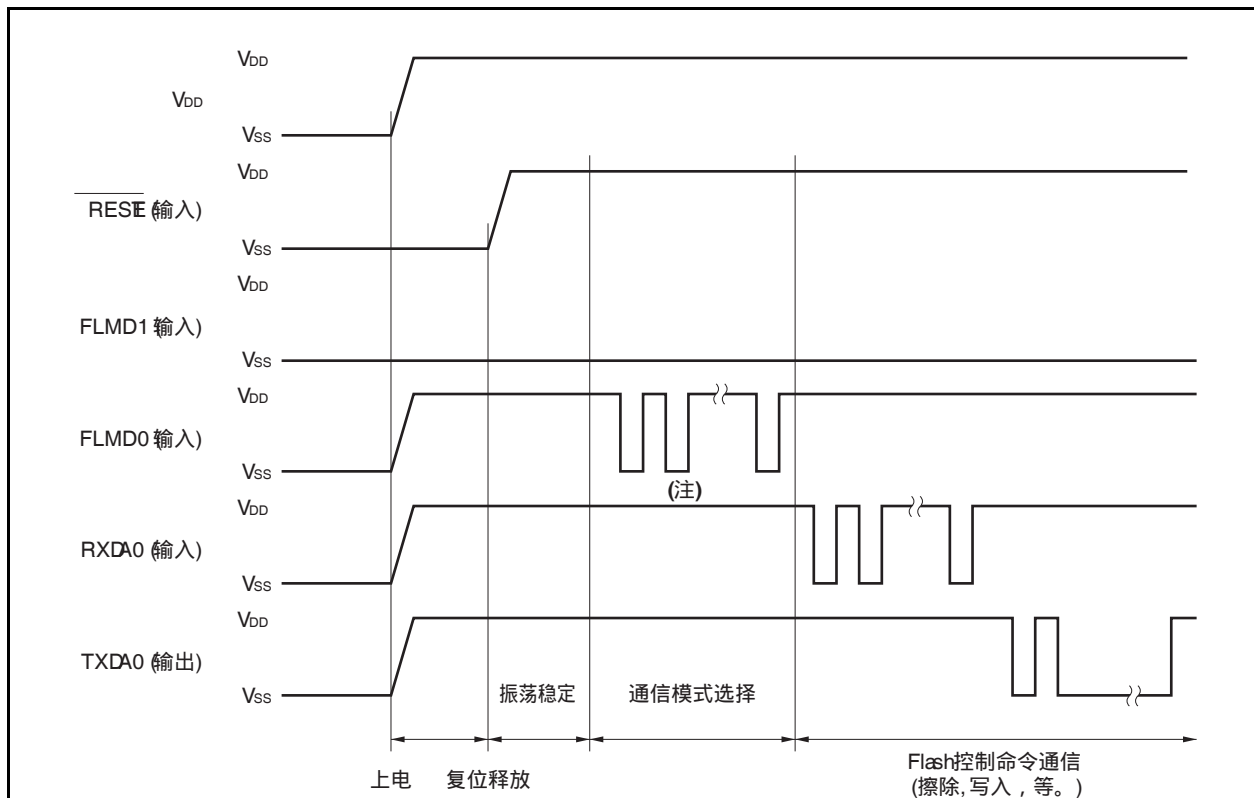


26.4.4 通信模式的选择

在 V850ES/JG2 中，进入 Flash 存储器编程模式后，通过将脉冲（可多达 12 个脉冲）输入到 FLMD0 引脚，来选择通信模式。由专用 Flash 编程器产生这些 FLMD0 脉冲。

下图显示了脉冲个数与通信模式之间的关系。

图 26-9. 通信模式的选择



注 如下时钟个数根据通信模式而不同。

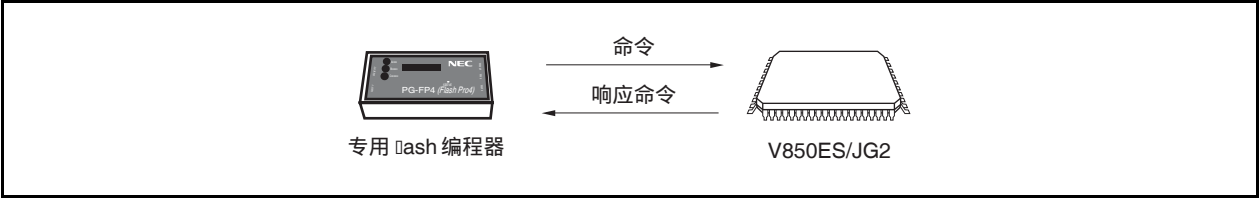
FLMD0 脉冲	通信模式	备注
0	UARTA0	通信速率：9,600 bps (复位后)，最低有效位先行
8	CSIB0	V850ES/JG2 执行从设备操作，最高有效位先行
9	CSIB3	V850ES/JG2 执行从设备操作，最高有效位先行
11	CSIB0 + HS	V850ES/JG2 执行从设备操作，最高有效位先行
12	CSIB3 + HS	V850ES/JG2 执行从设备操作，最高有效位先行
其他	保留给将来使用	设置禁止

注意事项 当选择 UARTA0 时，在接收到 FLMD0 脉冲后，根据专用 Flash 编程器发送的复位命令计算接收时钟。

26.4.5 通信命令

V850ES/JG2 使用命令与专用 Flash 编程器进行通信。从专用 Flash 编程器发往 V850ES/JG2 的信号称为“命令”，从 V850ES/JG2 发往专用 Flash 编程器的响应信号称为“响应命令”。

图 26-10. 通信命令



下面所示为在 V850ES/JG2 中用于 flash 存储器控制的命令。所有的这些命令从专用 flash 编程器发出，V850ES/JG2 根据这些命令执行处理。

表 26-7. Flash 存储器控制命令

类别	命令名称	支持			功能
		CSIB0, CSIB3	CSIB0 + HS, CSIB3 + HS	UARTA0	
空白检测	Block 空白检测命令	√	√	√	检查存储器中指定 block 的内容是否被正确地擦除。
擦除	芯片擦除命令	√	√	√	擦除整个存储器的内容。
	Block 擦除命令	√	√	√	擦除存储器指定 block 的内容。
写入	编程命令	√	√	√	写指定地址范围，并执行内容校验检查。
校验	校验命令	√	√	√	比较存储器指定地址范围的内容与 flash 编程器发送的数据。
	校验和命令	√	√	√	在指定地址范围读检验和。
系统设置，控制	硅标记命令	√	√	√	读硅标记信息。
	安全设置命令	√	√	√	禁止芯片擦除命令，block 擦除命令，和编程命令。

<R>

26.4.6 引脚连接

执行在板写入操作时，目标系统上必须有连接专用 Flash 编程器的连接器。另外，要提供一个在板功能，可以从正常操作模式切换到 Flash 存储器编程模式。

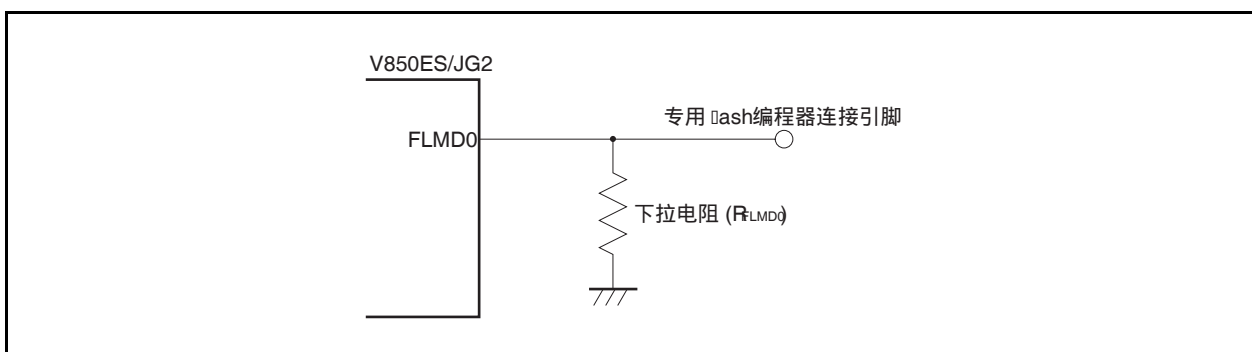
在 flash 存储器编程模式，所有那些不用于 Flash 存储器编程的引脚状态与复位伊始的状态相同。因此如果外部设备没有响应复位伊始的状态，则要求进行引脚处理。

(1) FLMD0 引脚

在正常操作模式下，向 FLMD0 引脚输入与 V_{SS} 相同电平的电压。在 Flash 存储器编程模式中，向 FLMD0 引脚提供与 V_{DD} 相同电平的写入电压。

因为在自编程模式下 FLMD0 引脚作为写入保护引脚，在写入 flash 存储器前，与 V_{DD} 相同电平的电压必须通过端口控制提供给 FLMD0 引脚，等等。更多细节，参见 26.5.5 (1) FLMD0 引脚。

图 26-11. FLMD0 引脚连接示例



(2) FLMD1 引脚

当 0 V 输入到 FLMD0 引脚，FLMD1 引脚不起作用。当 V_{DD} 供给 FLMD0 引脚，flash 存储器编程模式开始，因此 0 V 必须输入到 FLMD1 引脚。FLMD1 引脚的连接示例如下所示。

图 26-12. FLMD1 引脚连接示例

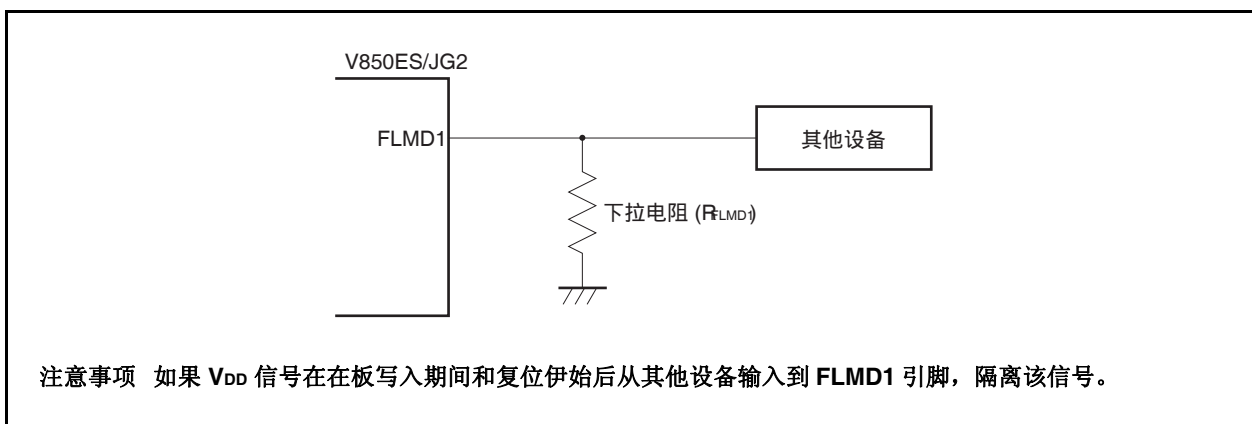


表 26-8. FLMD0 和 FLMD1 引脚的关系和复位释放后的操作模式

FLMD0	FLMD1	操作模式
0	无须关注	正常操作模式
V _{DD}	0	Flash 存储器编程模式
V _{DD}	V _{DD}	设置禁止

(3) 串行接口引脚

串行接口使用的引脚如下所示。

表 26-9. 串行接口使用的引脚

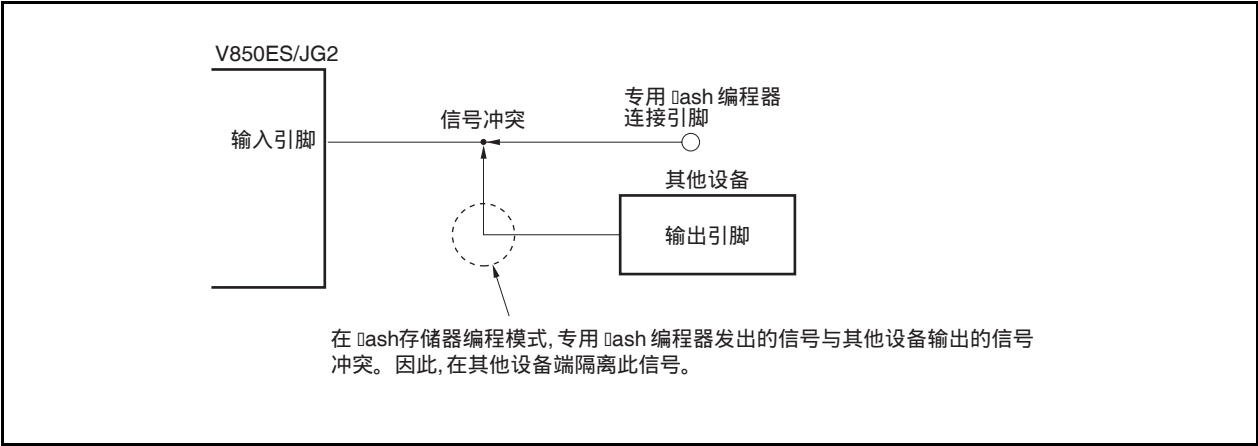
串行接口	使用的引脚
UARTA0	TXDA0, RXDA0
CSIB0	SOB0, SIB0, SCKB0
CSIB3	SOB3, SIB3, SCKB3
CSIB0 + HS	SOB0, SIB0, SCKB0, PCM0
CSIB3 + HS	SOB3, SIB3, SCKB3, PCM0

在将专用 Flash 编程器和串行接口 (已与板上其它设备连接) 的引脚相连时，必须特别注意以避免信号之间发生冲突及另一个设备出现误操作。

(a) 信号冲突

当专用 Flash 编程器 (输出端) 与串行接口 (已连接到另一个设备的输出端) 的一个串行接口引脚 (输入端) 相连时，这时会产生信号冲突。为了避免这种情况，应隔离与另一个设备的连接，或者使另一个设备处于输出高阻抗状态。

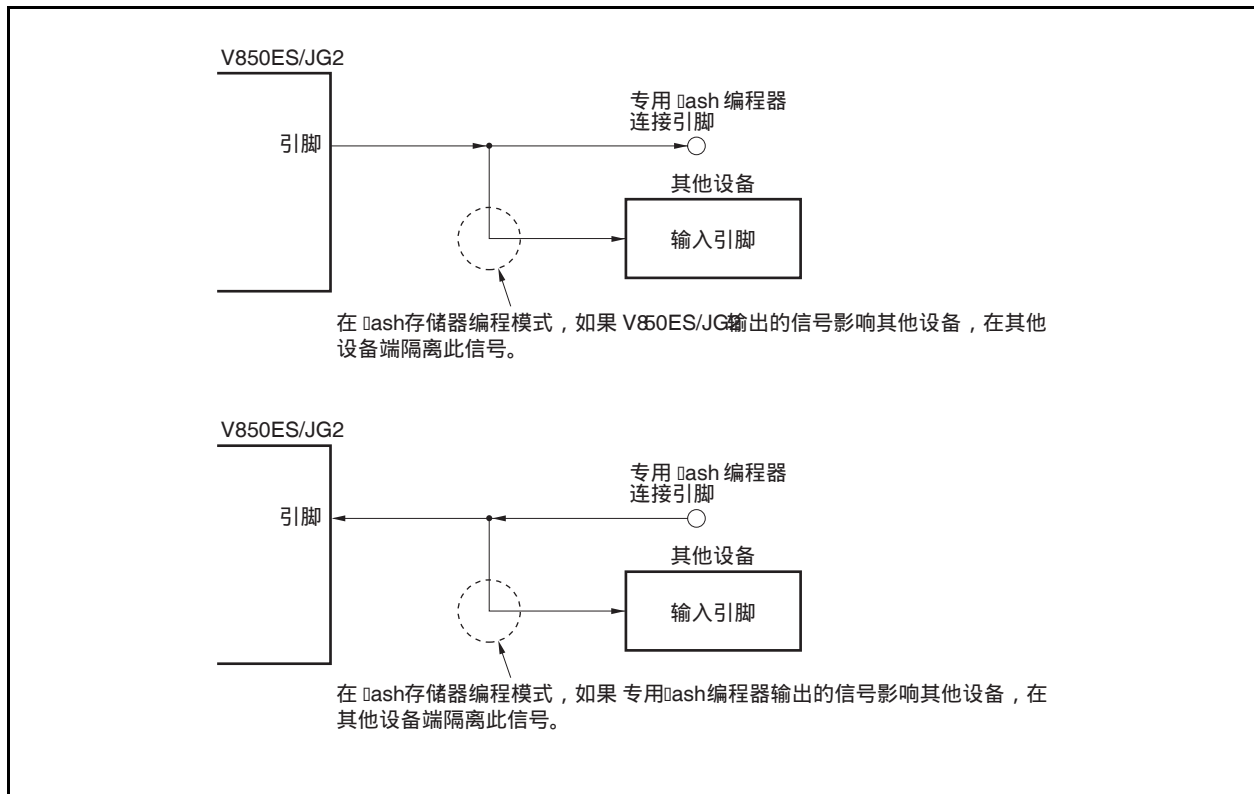
图 26-13. 信号冲突(串行接口输入引脚)



(b) 其他设备的误操作

当专用 Flash 编程器的输出或输入端与串行接口（已连接到另一个设备的输入端）的一个输入或输出引脚相连，则信号将输出到另一个设备，从而引起该设备的误操作。为了避免这种情况，应隔离与该设备的连接。

图 26-14. 其他设备的误操作

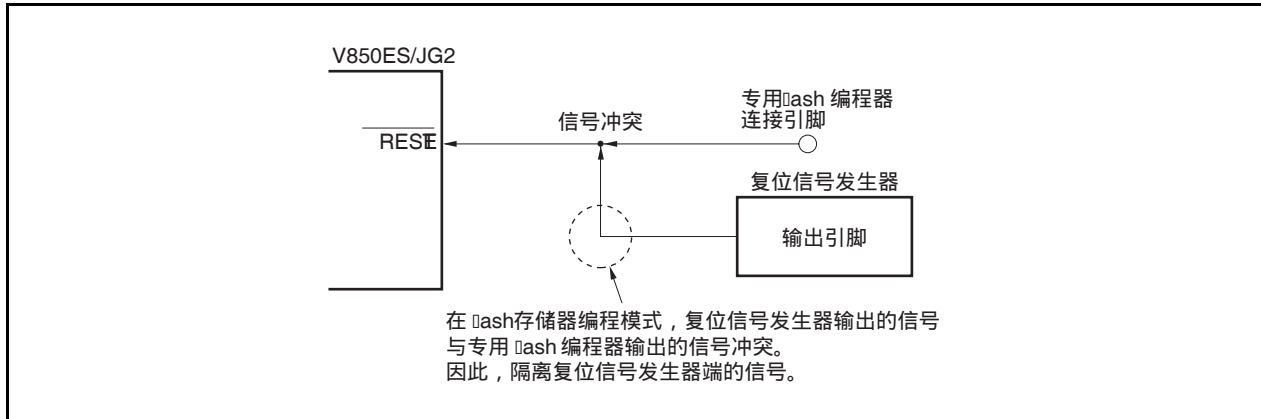


(4) RESET 引脚

当将专用 Flash 编程器的复位信号连接到 RESET 引脚（已连接到板上的复位信号发生器），则会产生信号冲突。为了避免这种情况，应隔离与复位信号发生器的连接。

在 Flash 存储器编程模式下，如果从用户系统输入复位信号，则不能对 Flash 存储器进行正确编程。因此除了专用 Flash 编程器的复位信号外，不要输入其它信号。

图 26-15. 信号冲突(RESET 引脚)

**(5) 端口引脚 (包括 NMI)**

当系统切换到 flash 存储器编程模式，所有不用于 flash 存储器编程的引脚状态与复位初始后的状态相同。

因此，如果与端口连接的外部设备没有识别端口复位初始后的状态，则要求对引脚进行适当地处理，比如通过电阻连接到 VDD 或 VSS。

(6) 其他信号引脚

连接 X1, X2, XT1, XT2, 及 REGC 与在正常操作模式下的状态相同。

在 flash 存储器编程期间，输入低电平到 DRST 引脚或置为开路。不要输入高电平。

(7) 电源

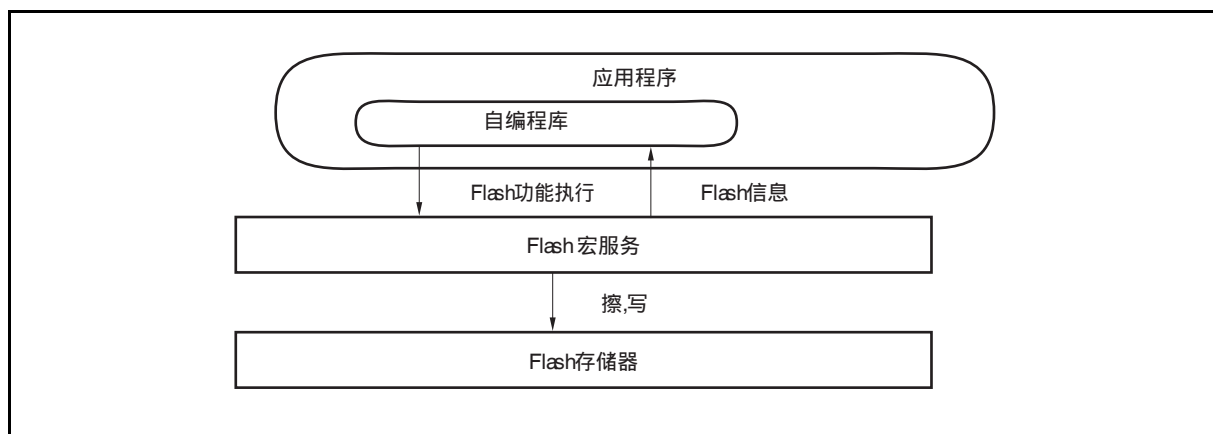
与正常操作模式下电源相同 (VDD, VSS, EVDD, EVSS, BVDD, BVSS, AVREF0, AVREF1, AVSS)。

26.5 自编程重写

26.5.1 概要

V850ES/JG2 支持允许用户程序自己重写内部 flash 存储器的 flash 宏指令服务。通过使用此接口和由用户应用程序实现的用于重写 flash 存储器的自编程库，flash 存储器可以通过预先传送到内部 RAM 或外部存储器中的用户应用程序重写。因此，可以更新用户程序，并且可以重写固定数据。

图 26-16. 自编程的概念

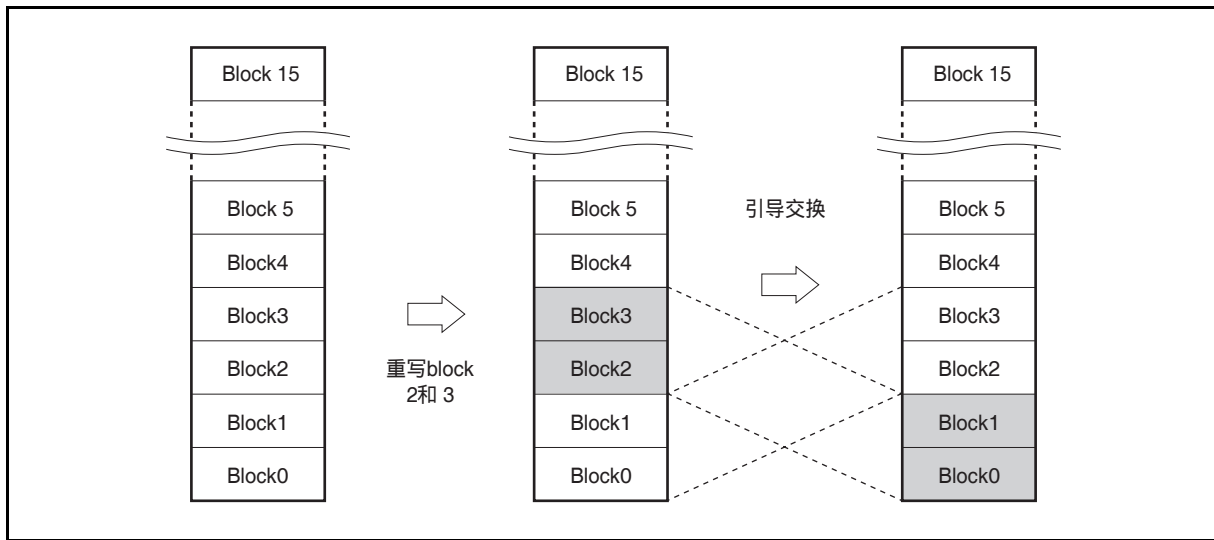


26.5.2 特性

(1) 安全的自编程(引导交换功能)

V850ES/JG2 支持引导交换功能，可以用物理存储器的 block 2 和 3 交换 block 0 和 1。通过预先向 block 2 和 3 写入一个要被重写的启动程序，然后交换物理存储器，整个区域可以被安全的重写，即使在重写过程中发生掉电，因为正确的用户程序永远存在于 block 0 和 1。

图 26-17. 写入整个存储器区域 (引导交换)



(2) 中断支持

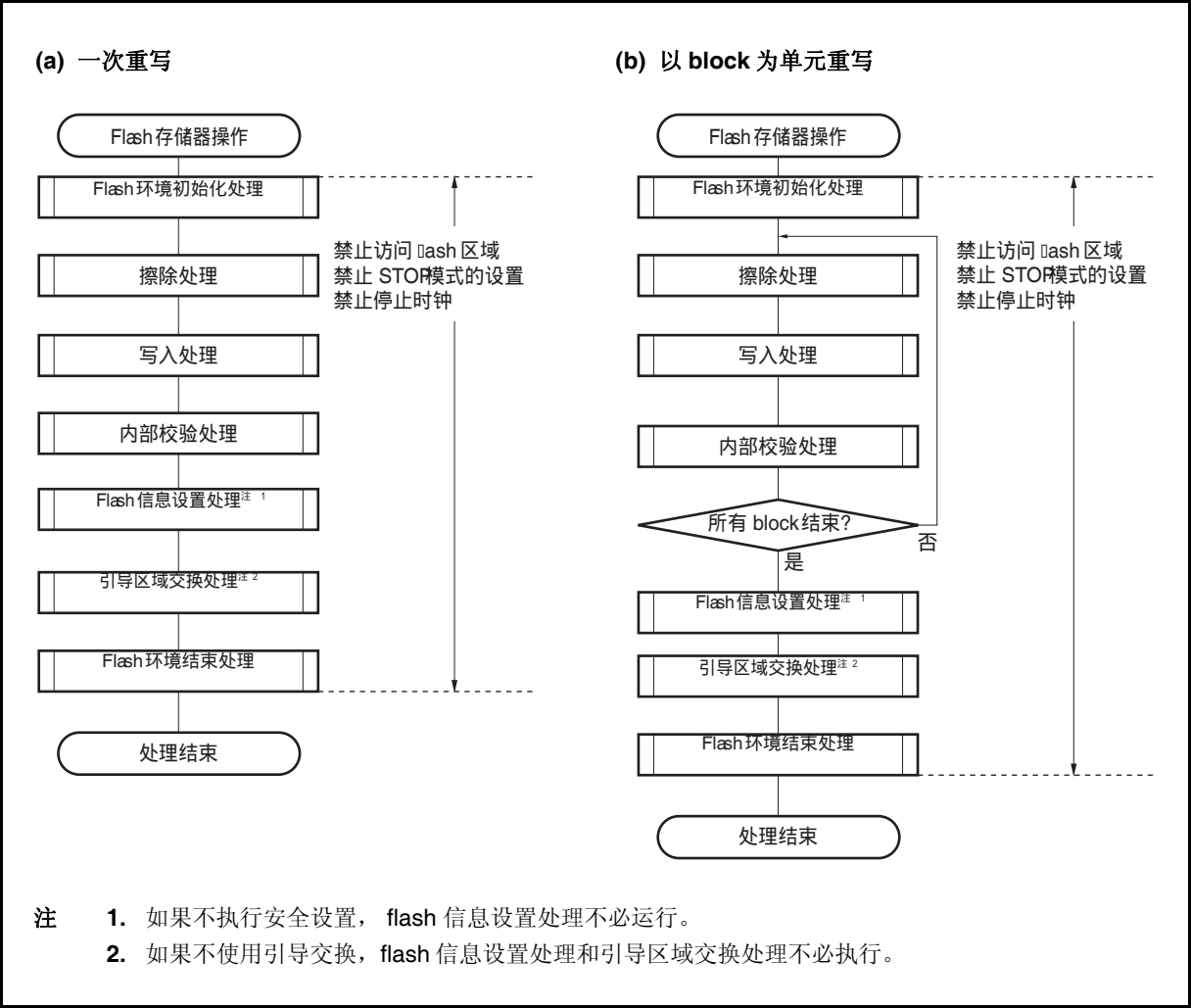
自编程期间不能从 flash 存储器取指令。因此，通常，即使中断发生写入 flash 存储器中的用户处理程序也不能被使用。在 V850ES/JG2 中，用户处理程序可以通过库功能在整个 RAM 区域注册，因此中断服务程序可以通过内部 RAM 或者外部存储器的执行被处理。

26.5.3 标准自编程流程

由 flash 自编程重写 flash 存储器的完整过程如下图所示。

<R>

图 26-18. 标准自编程流程



26.5.4 Flash 功能

表 26-10. Flash 功能列表

功能名称	概要	支持
FlashEnv	Flash 控制宏初始化	√
FlashBlockErase	仅擦除一个指定 block	√
FlashWordWrite	从指定地址写入	√
FlashBlockVerify	指定 block 的内部校验	√
FlashBlockBlankCheck	指定 block 的空白检测	√
FlashFLMDCheck	FLMD 引脚的检测	√
FlashStatusCheck	之前指定操作的状态检查	√
FlashGetInfo	读 flash 信息	√
FlashSetInfo	Flash 信息的设置	√
FlashBootSwap	引导区域交换	√
FlashSetUserHandler	用户中断处理程序注册功能	√

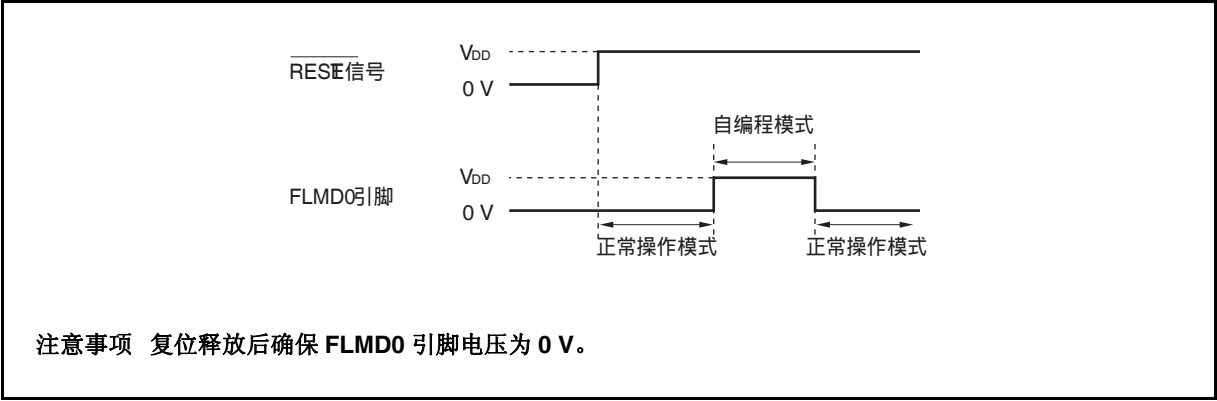
26.5.5 引脚处理

(1) FLMD0 引脚

FLMD0 引脚用于当复位释放时设置操作模式和在自重写时保护 flash 存储器不被写入。因此当复位释放和执行正常操作时必须保持 FLMD0 引脚电压为 0 V。重写存储器前，在自编程模式中也需要经由端口控制向 FLMD0 引脚提供与 V_{DD} 电平相同的电压。

自编程完成后，FLMD0 引脚的电压必须返回 0 V。

图 26-19. 模式改变时序



26.5.6 使用的内部资源

下表所列为用于自编程的内部资源。这些内部资源除了自编程也可以用于其他用途。

表 26-11. 使用的内部资源

资源名称	描述
入口 RAM 区域 (任选内部 RAM/外部 RAM124 字节)	用于 flash 宏指令服务的程序和参数存储在这个区域。通过调用库初始化功能复制入口程序和默认参数。
堆栈区域 (用户堆栈 + 300 字节)	通过库使用的由用户应用的扩展堆栈(内部 RAM 和外部 RAM 均可使用)。
库代码 (1900 字节)	库的程序实体 (除了被操作的 flash 存储器 block 外可以用于任何地方)。
应用程序	作为用户应用程序执行。 调用 flash 功能。
可屏蔽中断	可用于用户应用程序执行状态或自编程状态下。要在自编程状态下使用此中断，必须由注册功能预先注册中断服务起始地址。
NMI 中断	可用于用户应用程序执行状态或自编程状态下。要在自编程状态下使用此中断，必须由注册功能预先注册中断服务起始地址。

第二十七章 片上调试功能

V850ES/J2 具有片上调试功能，可以通过以下的两种方法实现。

- 使用 DCU (调试控制单元)
通过 $\overline{\text{DRST}}$, DCK, DMS, DDI, 以及 DDO 引脚作为调试接口引脚， V850ES/JG2 中的片上 DCU 来实现片上调试功能。
- 不使用 DCU
使用 MINICUBE2 或用户的其它的资源，通过代替 DCU 来实现片上调试的功能。

下表表示两种片上调试功能的特点。

表 27-1. 调试功能特点

		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 UARTA0 RXD0, TXD0 • 当使用 CSIB0 SIB0, SOB0, $\overline{\text{SCKB0}}$, HS (PCM0) • 当使用 CSIB3 SIB3, SOB3, $\overline{\text{SCKB3}}$, HS (PCM0)
Securement of user resources 用户资源保证		不需要	需要
硬件中断功能		2 个断点	2 个断点
软件中断功能	内部 ROM 区域	4 个断点	4 个断点
	内部 RAM 区域	2000 个断点	2000 个断点
实时 RAM 检测功能 ^{注1}		可以	可以
动态存储修改功能 (DMM) ^{注2}		可以	可以
屏蔽功能		$\overline{\text{Reset}}$, NMI, $\overline{\text{INTWDT2}}$, $\overline{\text{HLDRQ}}$, $\overline{\text{WAIT}}$	$\overline{\text{RESET}}$ 引脚
ROM 安全功能		10 字节 ID 代码认证	10 字节 ID 代码认证
使用的硬件		NINICUBE®, 等。	NINICUBE2, 等。
Trace 功能		不支持。	不支持。
调试中断接口功能(DBINT)		不支持。	不支持。

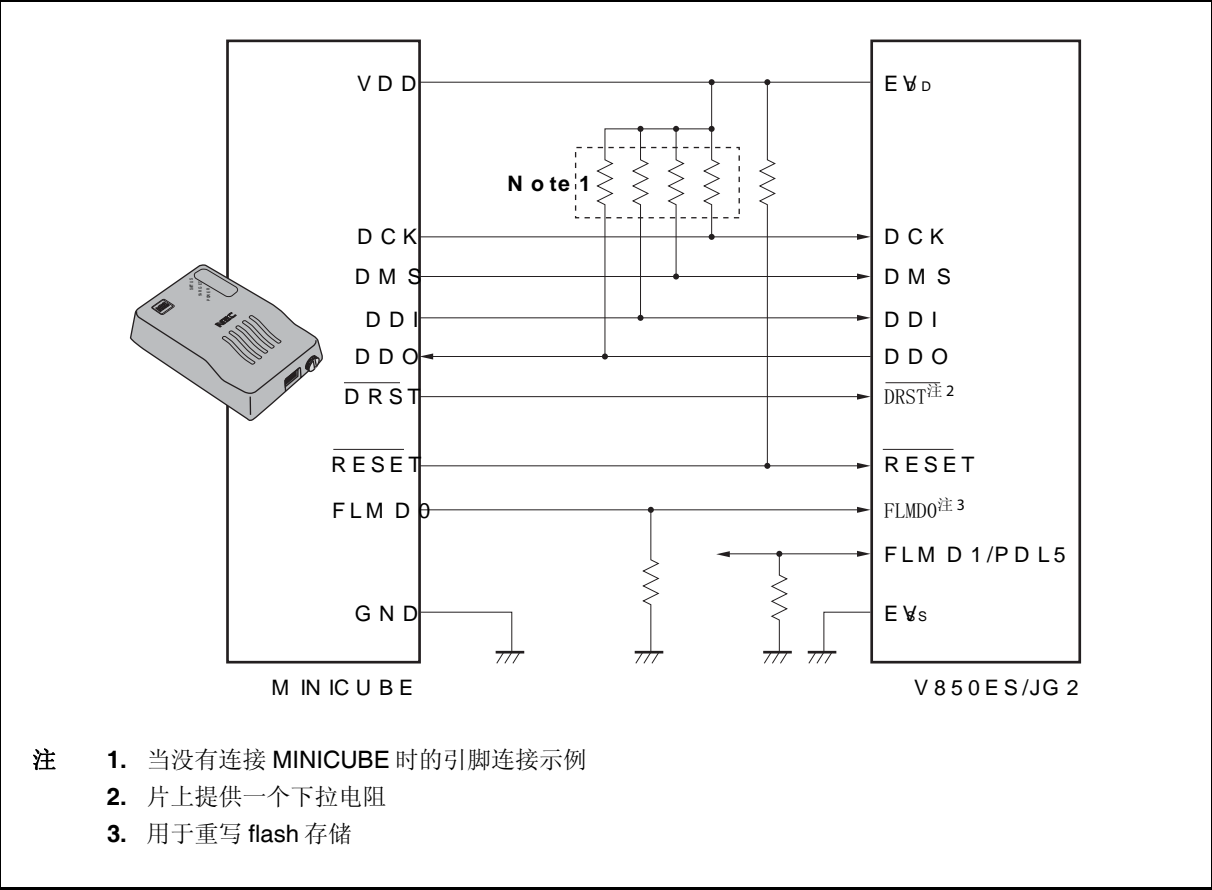
- 注 1. 该功能在程序执行期间读出存储内容。
2. 该功能在程序执行期间重写 RAM 内容。

27.1 使用 DCU 调试

通过调试接口引脚，连接上片上调试仿真器（ \overline{DRST} , DCK, DMS, DDI, 和 DDO），可以调试程序。

27.1.1 连接电路示例

图 27-1. 当调试接口引脚用于通讯接口是的电路连接示例



27.1.2 接口信号

接口信号如下所述。

(1) \overline{DRST}

对于片上调试单元这是一个复位输入信号。它是异步初始化调试控制单元的负逻辑信号。

在集成调试器启动后，当 MINICUBE 检测到目标系统的 VDD 后拉高 \overline{DRST} 信号，并且启动设备的片上调试单元。

当 \overline{DRST} 信号为高时，CPU 也产生一个复位信号。

当通过启动集成调试器开始调试时，CPU 也产生复位信号。

(2) DCK

这是一个时钟输入信号。它由 MINICUBE 提供一个 20 MHz 的时钟。在片上调试单元，当 DCK 信号上升沿时 DMS 和 DDI 信号被采样，在下降沿时数据 DDO 被输出。

(3) DMS

这是发送模式选择信号。测试单元的发送状态根据 DMS 信号的电平改变。

(4) DDI

这是一个数据输入信号。DCK 上升沿时在片上调试单元采样。

(5) DDO

这是一个数据输出信号。DCK 信号下降沿时从片上调试单元输出。

(6) EVDD

此信号用于检测目标系统的 $\overline{\text{VDD}}$ 。如果目标系统的 $\overline{\text{VDD}}$ 没有被检测到，MINICUBE ($\overline{\text{DRST}}$, $\overline{\text{DCK}}$, $\overline{\text{DMS}}$, $\overline{\text{DDI}}$, $\overline{\text{FLMD0}}$, 和 $\overline{\text{RESET}}$) 信号输出为高阻抗状态。

(7) FLMD0

The flash 自编程功能用于通过集成调试器下载数据到 flash 存储器。在 flash 自编程期间， $\overline{\text{FLMD0}}$ 引脚必须保持为高。另外，将下拉电阻连接到 $\overline{\text{FLMD0}}$ 引脚。

$\overline{\text{FLMD0}}$ 引脚也可以用如下两种方式控制。

<1> 通过 MINICUBE 控制

连接 MINICUBE 的 $\overline{\text{FLMD0}}$ 信号到 $\overline{\text{FLMD0}}$ 引脚。

在正常模式下，MINICUBE 不产生任何驱动 (高阻抗)。

中断期间，当集成调试器的下载功能执行时 MINICUBE 将 $\overline{\text{FLMD0}}$ 引脚置高电平。

<2> 通过端口控制

连接设备的任意端口到 $\overline{\text{FLMD0}}$ 引脚。

通过用户程序实现 flash 自编程功能任意端口都可以使用。

在集成调试器的控制台上，执行下载功能前设置端口引脚为高电平，或者执行完下载功能后置端口引脚为低。

更多细节，参见 ID850QB Ver. 3.10 集成调试器操作用户手册 (U17435E)。

(8) $\overline{\text{RESET}}$

这是一个系统复位输入引脚。如果 $\overline{\text{DRST}}$ 引脚由用户程序设置的 OCDM 寄存器的 OCDM0 位的数值而无效，那么不能执行片上调试功能。因此，使用 $\overline{\text{RESET}}$ 引脚，通过 MINICUBE 实现复位，使 $\overline{\text{DRST}}$ 引脚有效(初始化)。

27.1.3 可屏蔽功能

可以屏蔽 Reset, NMI, INTWDT2, $\overline{\text{WAIT}}$, 以及 $\overline{\text{HLD RQ}}$ 这些信号。
调试器（ID850QB）的可屏蔽功能和与之相一致的 V850ES/JG2 的功能如下。

表 27-2. 可屏蔽功能

ID850QB 的可屏蔽功能	V850ES/JG2 与之相一致的功能
NMI0	NMI 引脚输入
NMI2	产生不可屏蔽中断请求信号(INTWDT2)
STOP	—
HOLD	$\overline{\text{HLD RQ}}$ 引脚输入
RESET	复位信号由 $\overline{\text{RESET}}$ 引脚输入，低电压检测，时钟检测或看门狗定时器(WDT2)溢出而产生
WAIT	$\overline{\text{WAIT}}$ 引脚输入

27.1.4 寄存器

(1) 片上调试模式寄存器(OCDM)

OCDM 寄存器用于选择正常操作模式或片上调试模式。该寄存器是特殊寄存器，并且只有在对规定的顺序组合下才能写入 (参见 3.4.7 特殊寄存器)。
此寄存器也用来规定提供片上调试功能的引脚是作为片上调试引脚还是作为一般的端口/外围功能引脚。也可用于断开 P05/INTP2/ $\overline{\text{DRST}}$ 引脚的内部下拉电阻。
OCDM 寄存器仅当 $\overline{\text{DRST}}$ 引脚输入为低电平时可以写入。
该寄存器可用 8 位或 1 位指令读写。

复位后: 01H^注 R/W 地址: FFFF96H

	7	6	5	4	3	2	1	<0>
OCDM	0	0	0	0	0	0	0	OCDM0

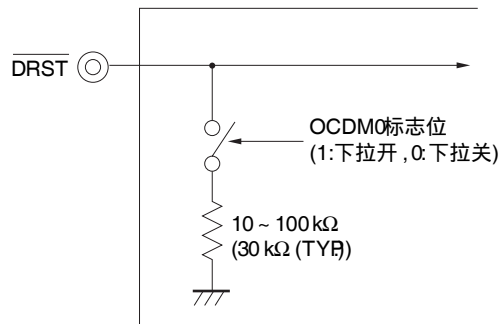
OCDM0	操作模式
0	选择普通工作模式 (在此模式下, 具有片上调试复用功能的引脚, 将不作为该复用功能使用, 而是用作普通的端口), 并且断开P05/INTP2/ <u>DRST</u> 脚的片上下拉电阻。
1	<p><u>DRST</u>脚为低时:</p> <p>普通工作模式 (在此模式下, 具有片上调试复用功能的引脚, 将不作为该复用功能使用, 而是作为普通的端口)</p> <p><u>DRST</u>脚为高时:</p> <p>片上调试模式 (片上调试模式下, 这些引脚才作为片上调试模式)</p>

注 RESET 输入设置此寄存器为 01H。通过 WDT2RES 信号复位后, 时钟监视器(CLM), 或低电压检测器(LVI), OCDM 寄存器的值仍被保留。

注意事项 1. 外部复位后, 当 DDI, DDO, DCK, 和 DMS 引脚不用于片上调试引脚而作为端口引脚时, 可以进行下面的任何一项操作。

- 输入低电压到 P05/INTP2/DRST 引脚。
- 设置 OCDM0 位。在这种情况下, 进行如下操作。
 - <1> OCDM0 位清零。
 - <2> P05/INTP2/DRST 引脚恒为低电平直到<1> 完成。

2. DRST 引脚有片上下拉电阻。这个电阻当 OCDM0 标志清 0 时断开。



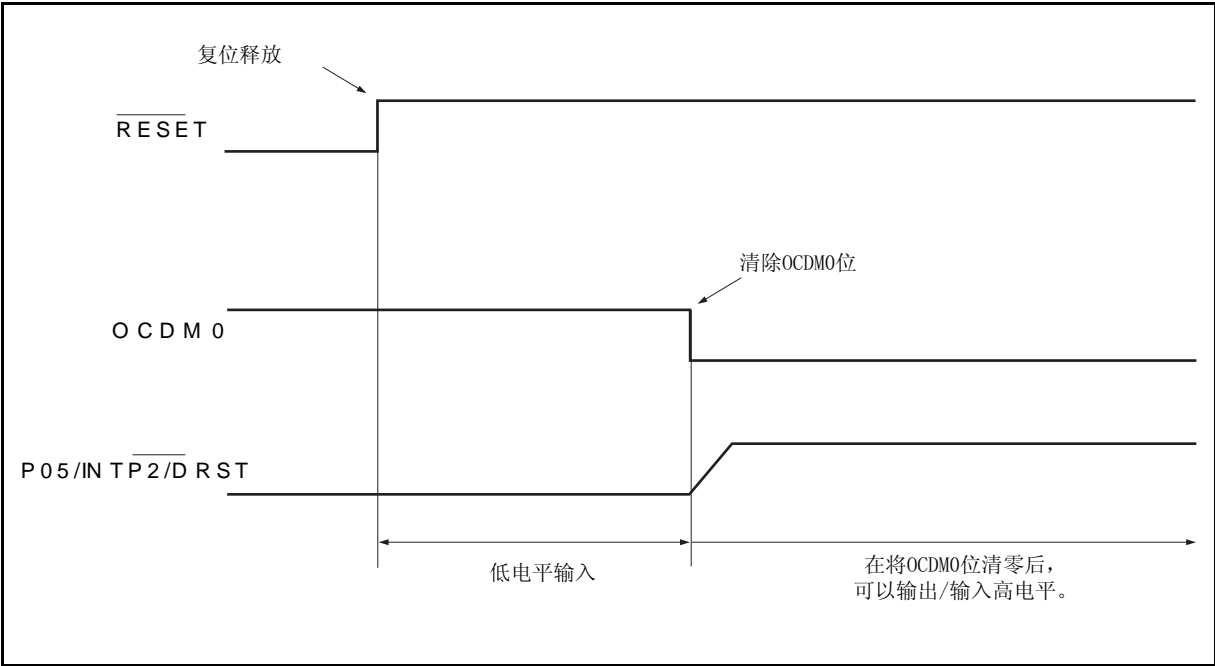
27.1.5 寄存器 操作

在下表中列出的条件下片上调试功能无效。
当不使用此功能时，保持 $\overline{\text{DRST}}$ 引脚为低直到 OCDM.OCDM0 标志被清零。

OCDM0 标志		0	1
$\overline{\text{DRST}}$ 引脚	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 27-2. 不应用片上调试功能时的时序



27.1.6 注意事项

- (1) RUN (程序执行)期间，如果输入复位信号(从目标系统或从内部复位源来的复位信号)，暂停功能就可能出现故障。
- (2) 即使复位信号由屏蔽功能屏蔽，如果复位信号是从引脚输入的话，I/O 缓冲器(端口引脚)也可能复位。
- (3) 因为内部 flash 存储器内的软件断点设置是通过 ROM 修正功能实现的，它可以通过目标复位或由看门狗定时器 2 产生的内部复位强制为无效。当硬件暂停或者强制暂停发生时，断点再次有效。在此期间不产生软件暂停。
- (4) 暂停期间屏蔽引脚复位屏蔽，CPU 和 外围 I/O 不复位。用户程序执行的时候，flash 存储器一旦由 DMA 重写或者由 RAM 监视功能读取就马上产生引脚复位或内部复位的情况下，CPU 和外围 I/O 可能不正确复位。
- (5) 当下列条件(a) 和 (b)满足时，由于暂停等原因仿真器(IECUBE[®], MINICUBE)操作停止的情况下，看门狗定时器 2 并不停止，并且产生复位或不可屏蔽中断。当发生复位时，调试器挂起。
 - (a) 主时钟或副时钟用作看门狗定时器 2 的源时钟。

(b) 内部振荡时钟停止(RCM.RSTOP 位 = 1).

为避免这种情况，可进行如下两种处理。

- 当使用仿真器时，内部振荡时钟用作源时钟。
- 当使用仿真器时，内部振荡器不要停止。

(6) 当下列条件(a) 和 (b)满足时，由于暂停等原因仿真器(IECUBE, MINICUBE)操作停止的情况下，即使外围暂停功能设置为“Break”， TMM 也并不停止。

- (a) INTWT，内部振荡时钟 ($f_R/8$)， 或者选择副时钟用作 TMM 源时钟
- (b) 主时钟停止。

为避免这种情况，可进行如下两种处理。

- 当使用仿真器时，主时钟 (f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/64$, $f_{xx}/512$) 用作源时钟。
- 当使用仿真器时，禁止主时钟振荡。

(7) 在片上调试模式下， DDO 引脚强制为高电平输出。

27.2 不使用 DCU 调试

以下叙述了在不使用 DCU 的情况下, 怎样使用 MINICUBE2 的 UARTA0 (RXDA0 以及 TXDA0)引脚, CSIB0 (SIB0, SOB0, SCKB0,以及 HS (PMC0))引脚, 或 CSIB3 (SIB3, SOB3, SCKB3,以及 HS (PMC0))引脚作为调试接口, 来实现片上调试功能。

27.2.1 电路连接示例

图 27-3. 当使用 UARTA0/CSIB0/CSIB3 作为通讯接口时的电路连接示例

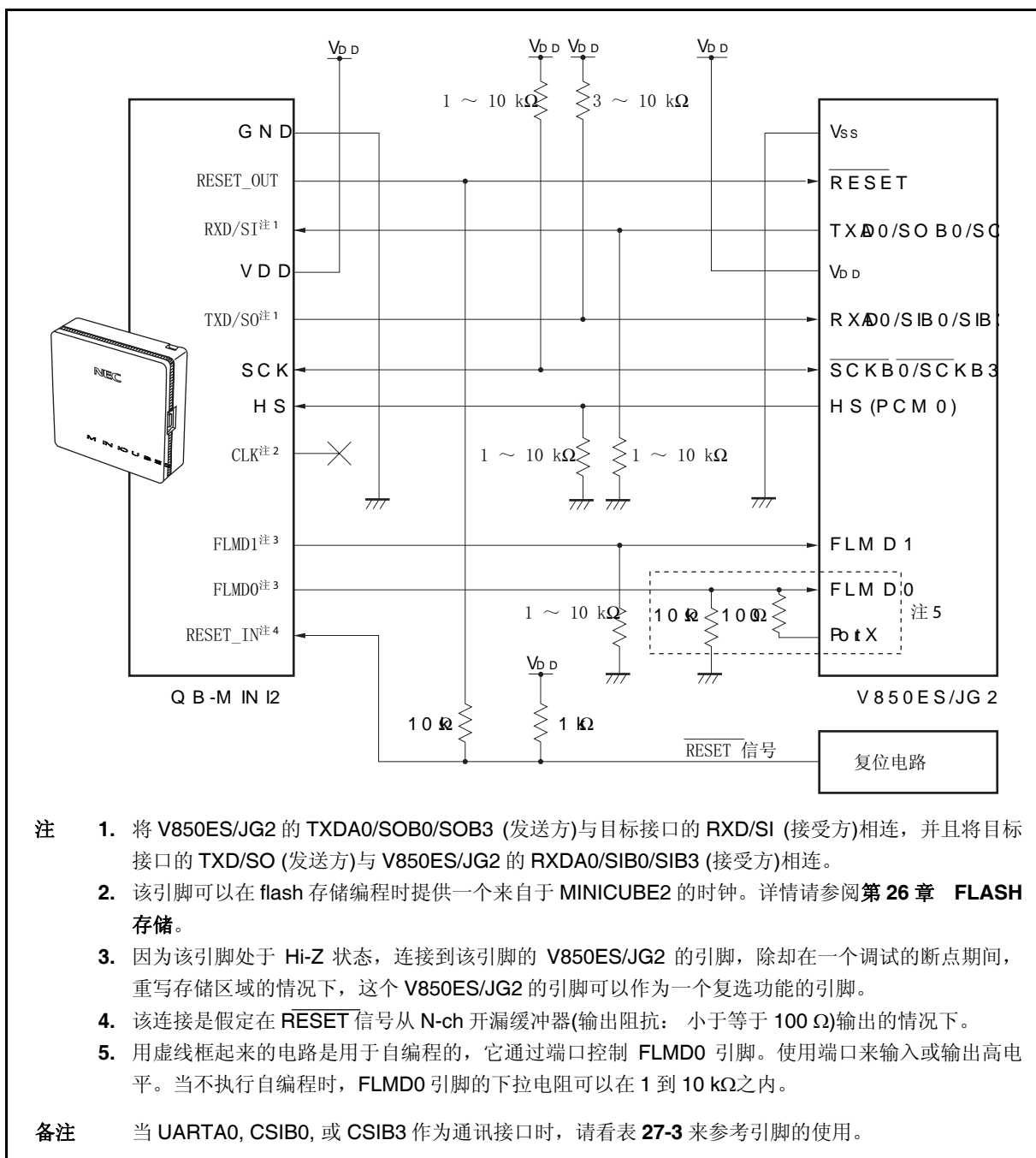


表 27-3. V850ES/JG2 和 MINICUBE2 之间的连线

MINICUBE2 (QB-MINI2)的引脚定义			CSIB0-HS			CSIB3-HS			UARTA0		
信号名称	I/O	引脚功能	引脚名称	引脚 No.		引脚名称	引脚 No.		引脚名称	引脚 No.	
				GC	GF		GC	GF		GC	GF
SI/RxD	输入	从 V850ES/JG2 接收数据和命令的引脚	P41/SOB0	23	25	P911/SOB3	54	56	P30/TXDA0	25	27
SO/TxD	输出	向 V850ES/JG2 发送数据和命令的引脚	P40/SIB0	22	24	P910/SIB3	53	55	P31/RXDA0	26	28
SCK	输出	3 先串行通讯的时钟输出	P42/SCKB0	24	26	P912/SCKB3	55	57	不需要	—	—
CLK ^{Note}	输出	向 V850ES/JG2 的时钟输出引脚	不需要 ^注	—	—	不需要 ^注	—	—	不需要 ^注	—	—
			不需要 ^注	—	—	不需要 ^注	—	—	不需要 ^注	—	—
RESET_OUT	输出	向 V850ES/JG2 的复位输出引脚	RESET	14	16	RESET	14	16	RESET	14	16
FLMD0	输出	输出引脚来设置 V850ES/JG2 的调试模式或编程模式	FLMD0	8	10	FLMD0	8	10	FLMD0	8	10
FLMD1	输出	输出引脚来设置编程模式	PDL5/FLMD1	76	78	PDL5/FLMD1	76	78	PDL5/FLMD1	76	78
HS	输入	CSIO + HS 通讯的握手信号	PCM0/WAIT	61	63	PCM0/WAIT	61	63	不需要	—	—
GND	—	地	V _{SS}	11	13	V _{SS}	11	13	V _{SS}	11	13
			AV _{SS}	2	4	AV _{SS}	2	4	AV _{SS}	2	4
			BV _{SS}	69	71	BV _{SS}	69	71	BV _{SS}	69	71
			EV _{SS}	33	35	EV _{SS}	33	35	EV _{SS}	33	35
RESET_IN	输入	目标系统上的复位输入引脚									

注 它用作 MINICUBE2 的 flash 编程器的时钟输出。详情请参阅 第 26 章 FLASH 存储。

备注 GC: 100-引脚 plastic LQFP (fine pitch) (14 × 14)
GF: 100-引脚 plastic QFP (fine pitch) (14 × 20)

27.2.2 可屏蔽功能

只能屏蔽复位信号。

T 调试器(ID850QB)的屏蔽功能和与之相一致的 V850ES/JG2 的功能如下所列。

表 27-4. 可屏蔽功能

ID850QB 的可屏蔽功能	V850ES/JG2 与之相一致的功能
NMI0	—
NMI1	—
NMI2	—
STOP	—
HOLD	—
RESET	由 RESET 引脚输入产生的复位信号
WAIT	—

27.2.3 用户资源的保证

用户必须有如下准备来执行 MINICUBE2 和目标设备的通讯，以及执行每个调试功能。用户必须在程序中设置或使用编译器选项来取保这些准备。

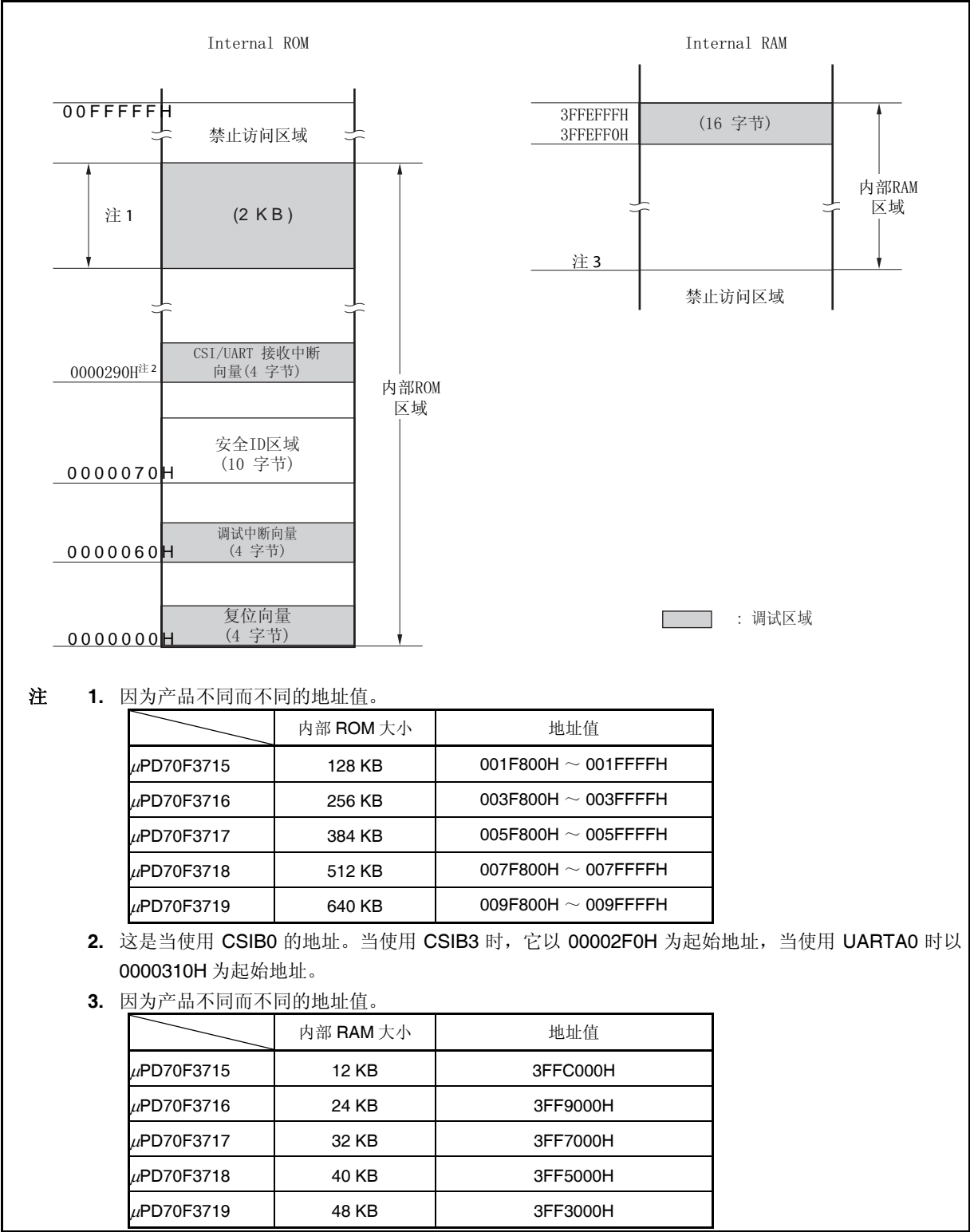
(1) 存储空间的保证

在图 27-4 中的阴影的部分作为保留区域用于放置调试监测程序，因此用户的程序和数据不能放置在这些空间中。

(2) 安全 ID 设置

ID 代码必须写在图 27-4 中区域的 0000070H 到 0000079H 之间，以确保只有相关授权人士才能读取程序。详情请参阅 **27.3 ROM 安全功能**。

图 27-4. 调试监测程序分配的存储空间



(3) 复位向量

复位向量包括用于调试监测程序的跳转指令。

[怎样保证区域]

无需特意去保护这个区域。然而，当下载一个程序时，调试器根据以下事件重写复位向量。如果重写模式与以下事件不能匹配，则调试器会产生一个错误（当使用 ID850QB 时错误信息为 F0C34）。

(a) 从地址 0 开始连续放置两条 nop 指令

重写前		重写后
0x0 nop	→	跳转到在 0x0 的调试监测程序
0x2 nop		0x4 xxxx
0x4 xxxx		

(b) 从地址 0(已经擦除的设备)开始连续放置两个 0xFFFF

重写前		重写后
0x0 0xFFFF	→	跳转到在 0x0 的调试监测程序
0x2 0xFFFF		0x4 xxxx
0x4 xxxx		

(c) 从地址 0（当使用 CA850）开始放置的 jr 指令

重写前		重写后
0x0 jr disp22	→	跳转到在 0x0 的调试监测程序
		0x4 jr disp22 - 4

(d) 从地址 0(当使用 IAR 编译器 ICCV850)开始连续放置的 mov32 和 jmp

重写前		重写后
0x0 mov imm32,reg1	→	跳转到在 0x0 的调试监测程序
0x6 jmp [reg1]		0x4 mov imm32,reg1
		0xa jmp [reg1]

(e) 从地址 0 开始放置的用于调试监测程序的跳转指令

重写前		重写后
跳转到在 0x0 的调试监测程序	→	无变化

(4) 用于调试监测程序的保证区域

图 27-4 中的阴影部分分配的是调试监测程序。该程序执行调试器通讯接口的初始化处理以及 RUN 或 CPU 的中断处理。内部 ROM 区域必须填以 0xFF。该区域不能由用户程序改写。

[怎样保证该区域]

如果用户程序不使用这块区域则不保证该区域。

为了避免在调试器启动过程中可能发生的问题，推荐通过编译器来预先保证这个区域。

以下是使用 NEC 电子编译器 CA850 时，保证该区域的示例。要添加的汇编源文件和链接指令代码则如下所示。

- 汇编程序 (将以下代码作为汇编源程序加入。)

```
-- 保证 2KB 的空间用于监测 ROM 段
.section "MonitorROM", const
.space    0x800, 0xff

-- 保证用于调试器的中断向量
.section "DBG0"
.space   4, 0xff

-- 保证用于串行通讯的中断向量
-- 根据使用的串行通讯模式来改变段名称
.section "INTCB0R"
.space   4, 0xff

-- 保证用于监测 RAM 段的 16 个字节
.section "MonitorRAM", bss
.lcomm   monitorramsym, 16, 4    -- defines symbol monitorramsym
```

- 链接指令 (向链接指令文件加入以下代码。)

以下示例说明了当内部 ROM 为 256 KB(结束地址为 003FFFFH)，内部 RAM 为 24 KB(结束地址为 3FFEFFFH)

```
MROMSEG      : !LOAD ?R V0x03f800{
               MonitorROM    = $PROGBITS    ?A MonitorROM;
};

MRAMSEG       : !LOAD ?RW V0x03ffef0{
               MonitorRAM     = $NOBITS      ?AW MonitorRAM;
};
```


(5) 保证串行端口的通讯

UARTA0, CSIB0, 或 CSIB3 用于 MINICUBE2 与目标系统的通讯。和串行接口模式有关的设置由调试监测程序执行，但如果由用户程序来改变这项设置，则可能发生通讯报错。
为了防止发生这样的问题，必须在用户程序中来保证通讯串行接口。

[如何保证通讯串行接口]

- 片上调试模式寄存器 (OCDM)
对于需要使用 UARTA0, CSIB0, 或 CSIB3 的片上调试功能时，设置 OCDM 寄存器为正常模式。需要按照如下设置。
 - 向 P05/INTP2/ $\overline{\text{DRST}}$ 引脚输入低电平。
 - 按如下设置 OCDM0 位。
 - <1> 将 OCDM0 位清零。
 - <2> 在步骤<1>的处理结束后，将 P05/INTP2/ $\overline{\text{DRST}}$ 引脚输入设为低电平。
- 串行接口寄存器
在用户程序中不要设置和 CSIB0, CSIB3, 或 UARTA0 有关的寄存器。
- 中断屏蔽寄存器
当使用 CSIB0 时，不要屏蔽发送结束中断(INTCB0R)。当使用 CSIB3 时，不要屏蔽发送结束中断(INTCB3R)。当使用 UARTA0 时，不要屏蔽接收结束中断(INTUA0R)。

(a) 当使用 CSIB0 时

	7	6	5	4	3	2	1	0
CB0RIC	×	0	×	×	×	×	×	×

(b) 当使用 CSIB3 时

	7	6	5	4	3	2	1	0
CB3RIC	×	0	×	×	×	×	×	×

(c) 当使用 UARTA0 时

	7	6	5	4	3	2	1	0
UA0RIC	×	0	×	×	×	×	×	×

备注 ×: 无需介意

- 当使用 UARTA0 时的端口寄存器

当使用时 UARTA0，通过调试监测程序，端口寄存器的设置要使 TXDA0 和 RXDA0 引脚有效。在调试期间，不要通过用户程序来改变如下的寄存器设置。（相同的值会被改写）

PFC3	7	6	5	4	3	2	1	0
	×	×	×	×	×	×	0	0
PMC3L	7	6	5	4	3	2	1	0
	×	×	×	×	×	×	1	1
备注 ×: 无需介意								

- 当使用 CSIB0 时的端口寄存器

当使用 CSIB0 时，通过调试监测程序，设置端口寄存器，使 SIB0, SOB0, $\overline{\text{SCKB0}}$, 和 HS (PMC0) 引脚有效。在调试期间，不要通过用户程序来改变如下的寄存器设置。（相同的值会被改写）

(a) SIB0, SOB0, 和 $\overline{\text{SCKB0}}$ 设置								
PMC4	7	6	5	4	3	2	1	0
	×	×	×	×	×	1	1	1
PFC4	7	6	5	4	3	2	1	0
	×	×	×	×	×	×	0	0
(b) HS (PMC0 引脚) 设置								
PMCM	7	6	5	4	3	2	1	0
	×	×	×	×	×	×	×	0
PCM	7	6	5	4	3	2	1	0
	×	×	×	×	×	×	×	注
注 禁止向该位写操作。 由根据调试状态的监测程序来改变与 HS 引脚相一致的端口值。为了以 8 位为单位执行端口寄存器的设置，用户程序通常可以使用读取-修改-写入。如果一个用于调试的中断在写操作前发生，则可能会执行一个不期望的操作。								
备注 ×: 无需介意								

- 当使用 CSIB3 时的端口寄存器
当使用 CSIB0 时，通过调试监测程序，设置端口寄存器，使 SIB3, SOB3, SCKB3, 和 HS (PMC0)引脚有效。在调试期间，不要通过用户程序来改变如下的寄存器设置。（相同的值会被改写）

(a) SIB3, SOB3, 和 SCKB3 设置

76543210

PMC9H

×

×

×

1

1

1

×

×

76543210

PFC9H

×

×

×

1

1

1

×

×

(b) HS (PMC0 引脚) 设置

76543210

PMCM

×

×

×

×

×

×

×

0

76543210

PCM

×

×

×

×

×

×

×

注

注

禁止向该位进行写操作。
由根据调试状态的监测程序来改变与 HS 引脚相一致的端口值。为了以 8 位为单位执行端口寄存器的设置，用户程序通常可以使用读取-修改-写入。如果一个用于调试的中断在写操作前发生，则可能会执行一个不期望的操作。

备注

×: 无需介意

27.2.4 注意事项

(1) 用于调试的设备的处理

不要将用于调试的设备焊接在准备量产的产品上，因为在调试期间重写 flash 存储器并且不能保证对其重写的次数。而且，不要将调试监测程序写入会量产的产品中去。

(2) 当不能执行中断时

如果符合以下之一的条件，则不能执行强制中断。

- 中断禁止(DI)
- 用于 MINICUBE2 和目标设备之间通讯的串行接口的中断被屏蔽。
- 进入了等待模式后，但禁止可以释放等待模式的一个可屏蔽中断
- 和目标设备之间的通讯模式是 UARTA0，且主时钟被 MINICUBE2 停止

用户手册 U17715CA2V0UD

781

(3) 当不执行伪实时 RAM 监测(RRM)功能和 DMM 功能

如果符合如下的条件之一，则不执行 PRM 功能和 DMM 功能

- 中断禁止(DI)
- 用于 MINICUBE2 和目标设备之间通讯的串行接口的中断被屏蔽。
- 进入了等待模式后，但禁止可以释放等待模式的一个可屏蔽中断
- MINICUBE2 和目标设备之间的通讯模式是 UARTA0，且主时钟已停止
- n 和目标设备之间的通讯模式是 UARTA0，而有一个与调试器不同的时钟用于通讯

(4) 允许由伪 RRM 和 DMM 功能来释放等待模式

如符合以下条件之一，则伪 RRM 和 DMM 功能可以释放等待模式

- MINICUBE2 和目标设备之间的通讯模式为 CSIB0 或 CSIB3
- MINICUBE2 和目标设备之间的通讯模式是 UARTA0，且提供主时钟。

(5) 使用 DMM 功能时，对外设 I/O 寄存器的写操作需要一个特殊的时序

需要一个特殊的外设 I/O 寄存器不能用 DMM 功能对其写操作。

(6) 调试器的启动会被减缓的器件

Chip erase and writing of the monitor program for debugging are conducted when the debugger is first started up, but this operation takes about a dozen seconds.当调试器最初启动时，导入用于调试的监测程序对芯片的擦除和写入，但该操作会需要 12 个时钟。

(7) 用于调试器的监测程序的写入

当使用调试器来改变 CPU 操作时钟设置时，调试器重写监测程序。所需要的时间与如上（6）中所要求的一样。对于集成调试器 ID850QB，改变在结构对话框中的时钟设置时，会发生这样的操作。

(8) Flash 自编程

如果一段分配了调试监测程序的空间由 flash 自编程改写，则调试器不再能够正常使用。

27.3 ROM 安全功能

27.3.1 安全 ID

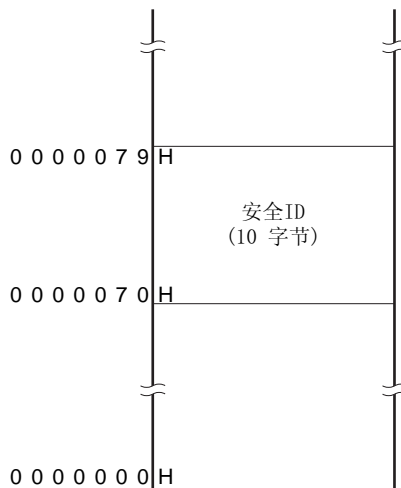
V850ES/JG2 的 flash 存储器版使用 10 个字节的 ID 码来进行加密，以防在片上调试期间 flash 存储器的内容被没有授权的人通过片上调试仿真器读取。

在 10 个字节的片上 flash 从 0000070H 到 0000079H 的存储区域进行设置，对调试器进行 ID 加密。

如果 IDs 一致，那么就释放安全保密功能，读取 flash 存储器和使用片上仿真器均可。

- 在 0000070H ~ 0000079H 之间设置 10 个字节的 ID 号码。
- 0000079H 的位 7 是 片上仿真器的允许标志 (0: 禁止, 1: 允许)
- 当片上仿真器开始时, 调试器要求输入 ID 号码。当输入的 ID 号码与在 0000070H ~ 0000079H 设置的 ID 号码一致时, 调试器开始工作。
- 如果片上调试器允许标志为 0, 即使 ID 号码匹配, 也不能进行调试。

图 27-5. 安全 ID 区域



注意事项 在擦除 flash 存储区域后，在整个区域写入 1。

27.3.2 设置

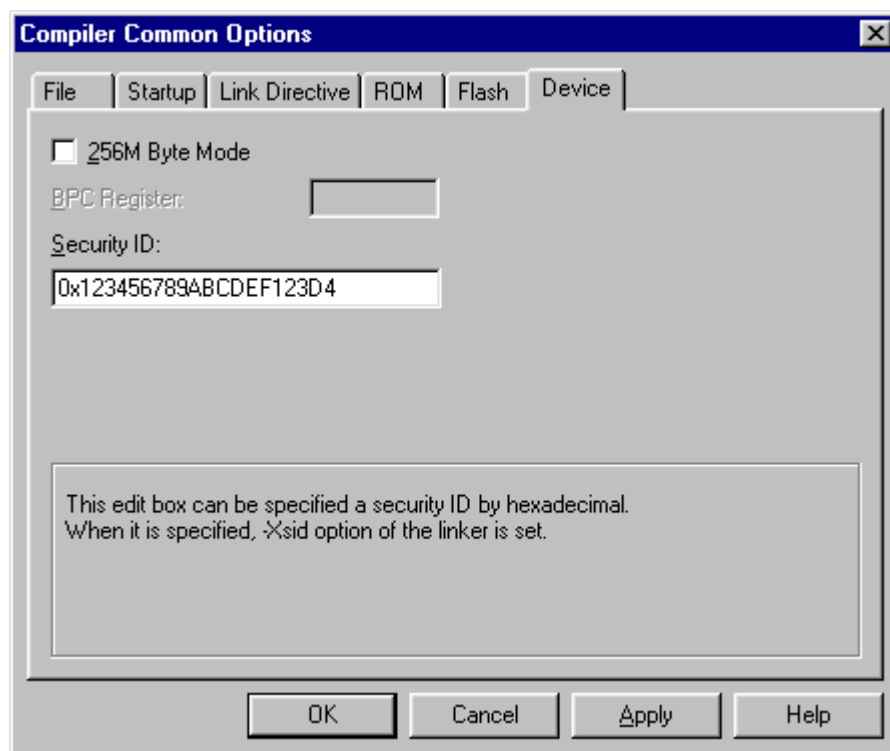
在表 27-5 中表述了如果设置 ID 码。

当 ID 码如表 27-5 中所示时，ID850QB 的结构对话框中的 ID 码输入为“123456789ABCDEF123D4”（ID 码与事件无关）。

表 27-5. ID 码

地址	值
0x70	0x12
0x71	0x34
0x72	0x56
0x73	0x78
0x74	0x9A
0x75	0xBC
0x76	0xDE
0x77	0XF1
0x78	0x23
0x79	0xD4

支持 CA850 Ver. 3.10 或更新版本的设备文件可以定义 ID 码，并且安全 ID 使用 PM+编译器普通选项设置。



[程序示例 (当使用 CA850 Ver. 3.10 或更新的版本)]

#-----
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

向起始文件（startup files）中加入以上程序示例。

第二十八章 电气特性

最大额定值 (T_A = 25°C) (1/2)

参数	符号	条件	额定值	单位
电源电压	V _{DD}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 ~ +4.6	V
	BV _{DD}		-0.5 ~ +4.6	V
	EV _{DD}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 ~ +4.6	V
	AV _{REF0}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 ~ +4.6	V
	AV _{REF1}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 ~ +4.6	V
	V _{SS}	V _{SS} = EV _{SS} = BV _{SS} = AV _{SS}	-0.5 ~ +0.5	V
	AV _{SS}	V _{SS} = EV _{SS} = BV _{SS} = AV _{SS}	-0.5 ~ +0.5	V
	BV _{SS}	V _{SS} = EV _{SS} = BV _{SS} = AV _{SS}	-0.5 ~ +0.5	V
	EV _{SS}	V _{SS} = EV _{SS} = BV _{SS} = AV _{SS}	-0.5 ~ +0.5	V
输入电压	V _{I1}	$\overline{\text{RESET}}$, FLMD0, PDH4, PDH5	-0.5 ~ EV _{DD} + 0.5 ^{※1}	V
	V _{I2}	PCM0 ~ PCM3, PCT0, PCT1, PCT4, PCT6, PDH0 ~ PDH3, PDL0 ~ PDL15	-0.5 ~ BV _{DD} + 0.5 ^{※1}	V
	V _{I3}	P10, P11	-0.5 ~ AV _{REF1} + 0.5 ^{※1}	V
	V _{I4}	X1, X2, XT1, XT2	-0.5 ~ V _{RO} ^{※2} + 0.5 ^{※1}	V
	V _{I5}	P02 ~ P06, P30 ~ P39, P40 ~ P42, P50 ~ P55, P90 ~ P915	-0.5 ~ +6.0	V
模拟输入电压	V _{IAN}	P70 ~ P711	-0.5 ~ AV _{REF0} + 0.5 ^{※1}	V

- 注
1. 确保不要超过每个电源电压的最大额定值 (MAX.值)。
 2. 片上调节器输出电压(2.5 V (TYP.))

备注 除非另外指定，否则复用功能引脚的特性与端口的特性相同。

最大额定值 ($T_A = 25^{\circ}\text{C}$) (2/2)

参数	符号	条件		额定值	单位
输出电流，低	I _{OL}	P02 ~ P06, P30 ~ P39, P40 ~ P42, P50 ~ P55, P90 ~ P915, PDH4, PDH5	每个引脚	4	mA
			全部引脚	50	mA
		PCM0 ~ PCM3, PCT0, PCT1, PCT4, PCT6, PDH0 ~ PDH3, PDL0 ~ PDL15	每个引脚	4	mA
			全部引脚	50	mA
		P10, P11	每个引脚	4	mA
			全部引脚	8	mA
		P70 ~ P711	每个引脚	4	mA
			全部引脚	20	mA
输出电流，高	I _{OH}	P02 ~ P06, P30 ~ P39, P40 ~ P42, P50 ~ P55, P90 ~ P915, PDH4, PDH5	每个引脚	−4	mA
			全部引脚	−50	mA
		PCM0 ~ PCM3, PCT0, PCT1, PCT4, PCT6, PDH0 ~ PDH3, PDL0 ~ PDL15	每个引脚	−4	mA
			全部引脚	−50	mA
		P10, P11	每个引脚	−4	mA
			全部引脚	−8	mA
		P70 ~ P711	每个引脚	−4	mA
			全部引脚	−20	mA
工作环境温度	T _A			−40 ~ +85	°C
存储温度	T _{stg}			−40 ~ +125	°C

- 注
1. 不要直接将 IC 产品的输出 (或 I/O) 引脚连接起来, 或连接到 V_{DD} , V_{CC} , 和 GND。漏极开路引脚或集电极开路引脚, 可以直接互相连接。
如果输出引脚设置为高阻抗状态并且避免与外部电路的输出时序冲突, 可以直接将 IC 产品的输出引脚与外部电路连接起来。
 2. DC 特性和 AC 特性中指出的额定值和条件是正常工作的质量保证。任何一项参数哪怕是在瞬间超过最大额定值, 都会使产品质量受到影响。也就是说, 最大额定值是产品濒临物理损坏的临界点, 因而, 必须保证产品在不超过最大额定值的条件下使用。

备注 除非另外指定, 否则复用功能引脚的特性与端口的特性相同。

电容 ($T_A = -40 \sim +85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

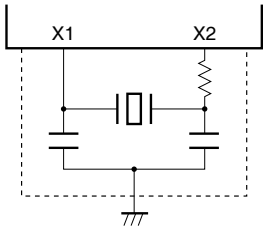
参数	符号	条件	MIN.	TYP.	MAX.	单位
I/O 电容	C_{IO}	$f_x = 1\text{ MHz}$ 不可测量的引脚返回 0 V			10	pF

工 作 条 件

($T_A = -40 \sim +85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

内部系统时钟频率	条件	电源电压				单位
		V_{DD}	EV_{DD}	BV_{DD}	AV_{REF0} , AV_{REF1}	
$f_{XX} = 2.5 \sim 20\text{ MHz}$	$C = 4.7\text{ }\mu\text{F}$, A/D 转换停止, D/A 转换停止	2.85 ~ 3.6	2.85 ~ 3.6	2.7 to 3.6	2.85 ~ 3.6	V
	$C = 4.7\text{ }\mu\text{F}$, A/D 转换操作。 D/A 转换操作	3.0 ~ 3.6	3.0 ~ 3.6	2.7 to 3.6	3.0 ~ 3.6	V
$f_{XT} = 32.768\text{ kHz}$	$C = 4.7\text{ }\mu\text{F}$, A/D 转换停止, D/A 转换停止	2.85 ~ 3.6	2.85 ~ 3.6	2.7 to 3.6	2.85 ~ 3.6	V

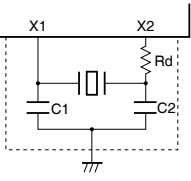
主 时 钟 振 荡 器 特 性
($T_A = -40 \sim +85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

谐振器	电路示例	参数	条件	MIN.	TYP.	MAX.	单位
陶瓷谐振器 / 晶体谐振器		振荡频率(f_x) ^{註1}		2.5		10	MHz
		振荡稳定时间 ^{註2}	复位后释放		$2^{16}/f_x$		s
			STOP模式后释放	$1^{\text{註4}}$	注3		ms
			IDLE2 模式后释放	$350^{\text{註4}}$	注3		μs

- 注 1. 上面所列的振荡频率仅指振荡器特性。使用 V850ES/ JG2，内部工作条件不能超出 **AC 特性**和 **DC 特性**中规定的额定范围。
2. 从振荡开始到谐振稳定的时间。
3. 该值根据 OSTS 寄存器的设置而改变。
4. Flash 存储器启动所需要的时间。使用 OSTS 寄存器的可靠启动时间。

- 注 1. 当使用主时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。
- 连接线越短越好。
 - 连接线不应与其他信号线交叉。
 - 流经的电流变化较大的信号线不要在振荡器周围布线。
 - 要保持振荡器电容器的接地点电压与 V_{SS} 相同。
 - 不要将电容的地信号接入大电流地。
 - 不要从振荡器获取信号。
2. 当主时钟停止副时钟工作时，在转换回主时钟前等待直到振荡稳定时间，由程序保证。

<R> (i) KYOCERA KINSEKI CORPORATION: 晶体谐振器 ($T_A = -40 \sim +85^{\circ}\text{C}$)

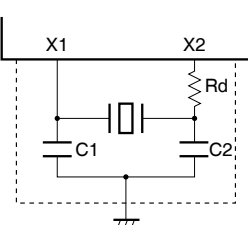
生产商 (零件号码)	电路示例	振荡频率 f_x (kHz)	推荐电路常数			振荡电压范围	
			C1 (pF)	C2 (pF)	Rd (k Ω)	MIN. (V)	MAX. (V)
KYOCERA KINSEKI CORPORATION (HC-49/U-S)		4,000	12	12	1	2.85	3.6
		5,000	10	10	1	2.85	3.6
		8,000	8	8	0	2.85	3.6
		10,000	8	8	0	2.85	3.6
		3,145.72	12	12	1	2.85	3.6
		4,718.592	10	10	0	2.85	3.6
		6,291.456	8	8	0	2.85	3.6

注意事项 该振荡器常数是由振荡器生产商提供的在指定环境下评估得出的参考值。

如果在实际应用中需要优化振荡器特性，请振荡器生产商在实验电路上评价。

振荡器电压和频率仅为振荡器特性。使用 V850ES/JG2，内部工作条件不能超出 DC 特性和 AC 特性中规定的额定范围。

<R> (ii) Murata Mfg. Co. Ltd.: 陶瓷谐振器 ($T_A = -40 \sim +85^{\circ}\text{C}$)

类型	电路示例	编号	振荡频率 f _x (MHz)	推荐电路常数			晶振电压范围	
				C1 (pF)	C2 (pF)	R _d (kΩ)	MIN. (V)	MAX. (V)
表面封装		CSTCR4M00G55-R0	4.000	(39)	(39)	0	2.85	3.60
		CSTCR5M00G55-R0	5.000	(39)	(39)	0	2.85	3.60
		CSTCR6M00G55-R0	6.000	(39)	(39)	0	2.85	3.60
		CSTCE8M00G55-R0	8.000	(33)	(33)	0	2.85	3.60
		CSTCE10M0G55-R0	10.000	(33)	(33)	0	2.85	3.60
Lead		CSTLS4M00G56-B0	4.000	(47)	(47)	0	2.85	3.60
		CSTLS5M00G53-B0	5.000	(15)	(15)	0	2.85	3.60
		CSTLS6M00G53-B0	6.000	(15)	(15)	0	2.85	3.60
		CSTLS8M00G53-B0	8.000	(15)	(15)	0	2.85	3.60
		CSTLS10M0G53-B0	10.000	(15)	(15)	0	2.85	3.60

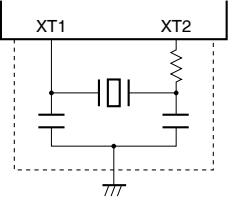
注意事项 该振荡器常数是由振荡器生产商提供的在指定环境下评估得出的参考值。

如果在实际应用中需要优化振荡器特性，请振荡器生产商在实验电路上评价。

振荡器电压和频率仅为振荡器特性。使用 V850ES/JG2，内部工作条件不能超出 DC 特性和 AC 特性中规定的额定范围。

备注 C1 和 C2 列中括号中的数值指的是与晶振匹配的电容。

副 时 钟 振 荡 器 特 性
($T_A = -40 \sim +85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

谐振器	电路示例	参数	条件	MIN.	TYP.	MAX.	单位
晶体谐振器		振荡频率 (f_{XT}) ^{※1}		32	32.768	35	kHz
		振荡稳定时间 ^{※2}				10	s

- 注 1. 上面所示的振荡频率仅为振荡器特性。因此使用 V850ES/JG2 内部工作条件不要超过 **AC 特性**和 **DC 特性**中所示的额定值。
2. 从 V_{DD} 达到振荡电压范围 (2.85 V (MIN.))到晶体谐振器稳定需要时间。
- 注 1. 当使用副时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。
- 连接线越短越好。
 - 连接线不应与其他信号线交叉。
 - 流经的电流变化较大的信号线不要在振荡器周围布线。
 - 要保持振荡器电容器的接地点电压与 V_{SS} 相同。
 - 不要将电容的地信号接入大电流地。
 - 不要从振荡器获取信号。
2. 为了缩减能耗，副时钟振荡器被设计为低幅电路，并且它比主时钟振荡器更容易受到噪声的影响。
- 因此在使用副时钟时更要注意布线方法。
3. 关于谐振器的选择和振荡器常数，请客户自己评价或使用谐振器生产商的评价。

PLL特 性
($T_A = -40 \sim +85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

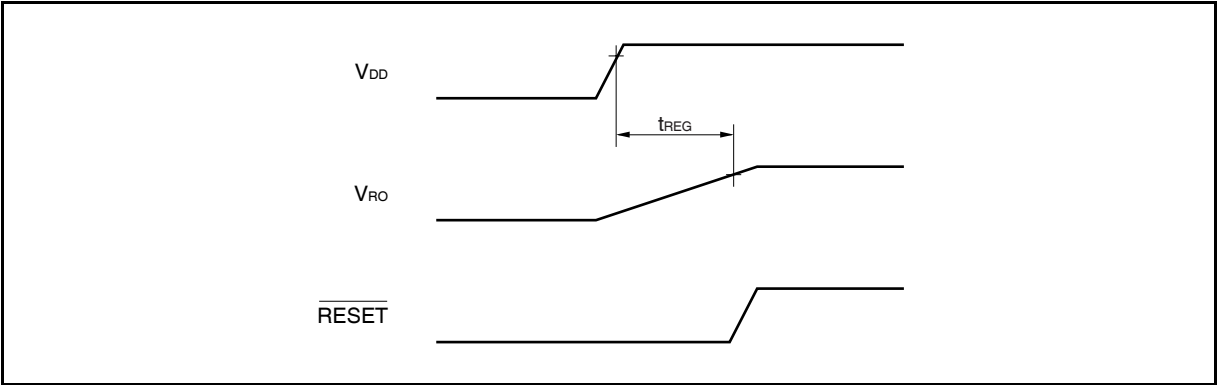
参数	符号	条件	MIN.	TYP.	MAX.	单位
输入频率	f _x	×4 模式	2.5		5	MHz
		×8 模式	2.5		2.5	MHz
输出频率	f _{xx}	×4 模式	10		20	MHz
		×8 模式	20		20	MHz
锁时间	t _{PLL}	V _{DD} 达到2.85 V (MIN.)后			800	μs

内部振荡器特 性
($T_A = -40 \sim +85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
输出频率	f _R		100	200	400	kHz

调节器特 性
($T_A = -40 \sim +85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
输入电压	V _{DD}	f _{xx} = 20 MHz (MAX.)	2.85		3.6	V
输出电压	V _{RO}			2.5		V
调节器输出稳定时间	t _{REG}	V _{DD} 达到2.85 V (MIN.)后, 稳定电容 C = 4.7 μF连接到REGC 引脚			1	ms



DC

特

性

(T_A = -40 ~ +85°C, BV_{DD} ≤ V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V) (1/3)

参数	符号	条件	MIN.	TYP.	MAX.	单位
输入电压, 高	V _{IH1}	PDH4, PDH5	0.7EV _{DD}		EV _{DD}	V
	V _{IH2}	RESET, FLMD0	0.8EV _{DD}		EV _{DD}	V
	V _{IH3}	P02 ~ P06, P30 ~ P37, P42, P50 ~ P55, P92 ~ P915	0.8EV _{DD}		5.5	V
	V _{IH4}	P38, P39, P40, P41, P90, P91	0.7EV _{DD}		5.5	V
	V _{IH5}	PCM0 ~ PCM3, PCT0, PCT1, PCT4, PCT6, PDH0 ~ PDH3, PDL0 ~ PDL15	0.7BV _{DD}		BV _{DD}	V
	V _{IH6}	P70 ~ P711	0.7AV _{REF0}		AV _{REF0}	V
	V _{IH7}	P10, P11	0.7AV _{REF1}		AV _{REF1}	V
输入电压, 低	V _{IL1}	PDH4, PDH5	EV _{SS}		0.3EV _{DD}	V
	V _{IL2}	RESET, FLMD0	EV _{SS}		0.2EV _{DD}	V
	V _{IL3}	P02 ~ P06, P30 ~ P37, P42, P50 ~ P55, P92 ~ P915	EV _{SS}		0.2EV _{DD}	V
	V _{IL4}	P38, P39, P40, P41, P90, P91	EV _{SS}		0.3EV _{DD}	V
	V _{IL5}	PCM0 ~ PCM3, PCT0, PCT1, PCT4, PCT6, PDH0 ~ PDH3, PDL0 ~ PDL15	BV _{SS}		0.3BV _{DD}	V
	V _{IL6}	P70 ~ 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} = BV _{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} = BV _{DD} = AV _{REF0} = AV _{REF1}			5	μA
输出漏电流, 低	I _{LOL}	V _O = 0 V			-5	μA

备注 除非另外指定, 否则复用功能引脚的特性与端口的特性相同。

DC

特

性

(T_A = -40 ~ +85°C, V_{DD} ≤ V_{DD} = E_{VDD} = A_{VREF0} = A_{VREF1}, V_{SS} = E_{VSS} = B_{VSS} = A_{VSS} = 0 V) (2/3)

参数	符号	条件			MIN.	TYP.	MAX.	单位
输出电压, 高	V _{OH1}	P02 ~ P06, P30 ~ P39, P40 ~ P42, P50 ~ P55, P90 ~ P915, PDH4, PDH5	每个引脚 I _{OH} = -1.0 mA	全部引脚 -20 mA	E _{VDD} - 1.0		E _{VDD}	V
			每个引脚 I _{OH} = -100 μA	全部引脚 -6.0 mA	E _{VDD} - 0.5		E _{VDD}	V
	V _{OH2}	PCM0 ~ PCM3, PCT0, PCT1, PCT4, PCT6, PDH0 ~ PDH3, PDL0 ~ PDL15	每个引脚 I _{OH} = -1.0 mA	全部引脚 -20 mA	B _{VDD} - 1.0		B _{VDD}	V
			每个引脚 I _{OH} = -100 μA	全部引脚 -2.8 mA	B _{VDD} - 0.5		B _{VDD}	V
	V _{OH3}	P70 ~ P711	每个引脚 I _{OH} = -0.4 mA	全部引脚 -4.8 mA	A _{VREF0} - 1.0		A _{VREF0}	V
			每个引脚 I _{OH} = -100 μA	全部引脚 -1.2 mA	A _{VREF0} - 0.5		A _{VREF0}	V
	V _{OH4}	P10, P11	每个引脚 I _{OH} = -0.4 mA	全部引脚 -0.8 mA	A _{VREF1} - 1.0		A _{VREF1}	V
			每个引脚 I _{OH} = -100 μA	全部引脚 -0.2 mA	A _{VREF1} - 0.5		A _{VREF1}	V
输出电压, 低	V _{OL1}	P02 ~ P06, P30 ~ P37, P42, P50 ~ P55, P92 ~ P915, PDH4, PDH5	每个引脚 I _{OL} = 1.0 mA	全部引脚 20 mA	0		0.4	V
	V _{OL2}	P38, P39, P40, P41, P90, P91	每个引脚 I _{OL} = 3.0 mA		0		0.4	V
	V _{OL3}	PCM0 ~ PCM3, PCT0, PCT1, PCT4, PCT6, PDH0 ~ PDH3, PDL0 ~ PDL15	每个引脚 I _{OL} = 1.0 mA	全部引脚 20 mA	0		0.4	V
	V _{OL4}	P10, P11, P70 ~ P711	每个引脚 I _{OL} = 0.4 mA	全部引脚 5.6 mA	0		0.4	V
软件下拉电阻	R _I	P05	V _I = V _{DD}		10	30	100	kΩ

备注

1. 除非另外指定, 否则复用功能引脚的特性与端口的特性相同。
2. 当 I_{OH} 和 I_{OL} 条件不满足但所有引脚的总和值满足时, 仅那个引脚不能满足 DC 特性。

DC

特

性

(T_A = -40 ~ +85°C, B_{VDD} ≤ V_{DD} = E_{VDD} = A_{VREF0} = A_{VREF1}, V_{SS} = E_{VSS} = B_{VSS} = A_{VSS} = 0 V) (3/3)

参数	符号	条件			MIN.	TYP.	MAX.	单位
电流 ^{注1}	IDD1	正常操作	fxx = 20 MHz (fx = 5 MHz)	注 2		32	48	mA
				注 3		30	45	mA
	IDD2	HALT 模式	fxx = 20 MHz (fx = 5 MHz)	注 2		17	26	mA
				注 3		16	24	mA
	IDD3	IDLE1 模式	fxx = 5 MHz (fx = 5 MHz), PLL off			0.8	1.6	mA
	IDD4	IDLE2 模式	fxx = 5 MHz (fx = 5 MHz), PLL off			0.3	0.8	mA
	IDD5	副时钟操作模式	fXT = 32.768 kHz, 主时钟, 内部振荡器停止	注 2		300	600	μA
				注 3		200	400	μA
	IDD6	Sub-IDLE 模式	fXT = 32.768 kHz, 主时钟, 内部振荡器停止	注 2		18	100	μA
				注 3		18	80	μA
	IDD7	STOP 模式	副时钟停止, 内部振荡器停止			6	50	μA
			副时钟工作, 内振荡器停止			10	60	μA
			副时钟停止, 内部振荡器工作			10	60	μA
	IDD8	Flash 存储器编程 模式模式	fxx = 20 MHz (fx = 5 MHz)	注 2		35	54	mA
				注 3		33	51	mA

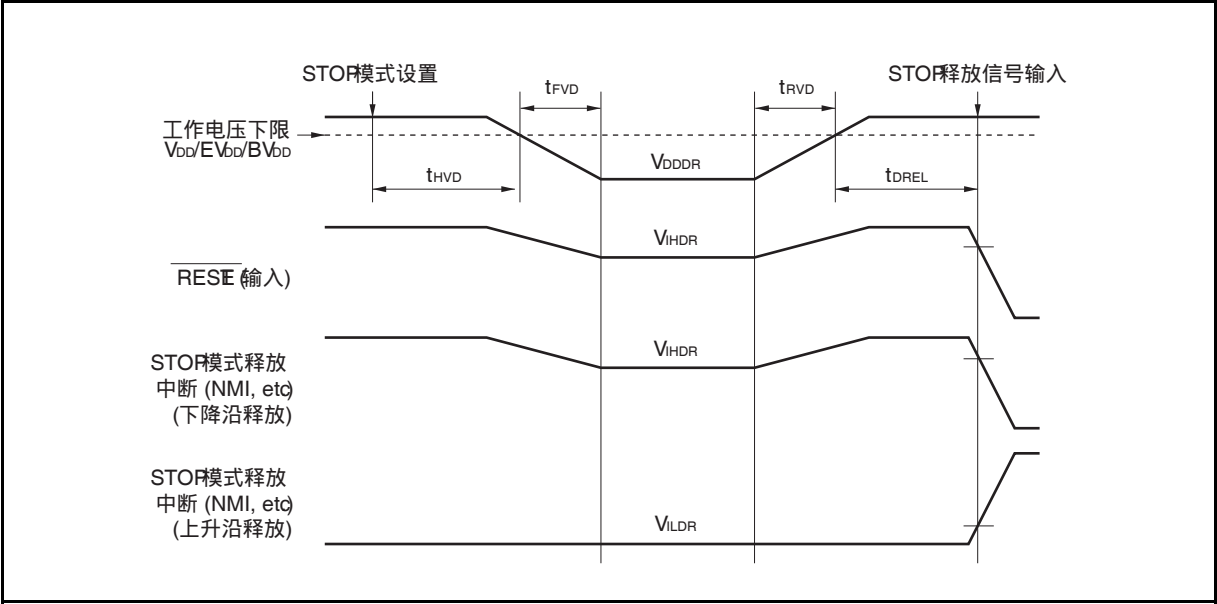
- 注 1. V_{DD}, E_{VDD}, 和 B_{VDD} 电流总和。电流流过输出缓冲器, A/D 转换器, D/A 转换器, 片上下拉电阻不包括在内。
2. μPD70F3718, 70F3719
3. μPD70F3715, 70F3716, 70F3717

数 据 保 持 特 性

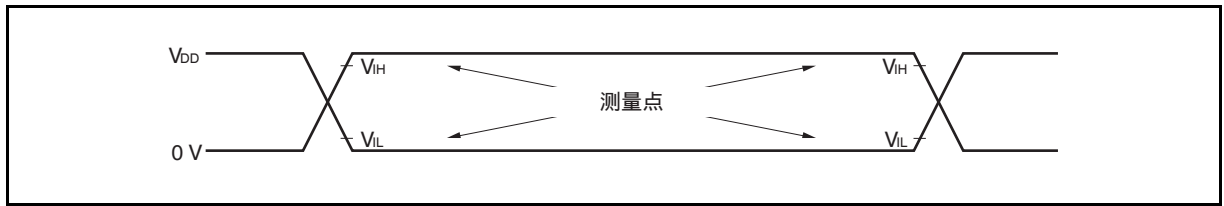
在 **STOP** 模 式
($T_A = -40 \sim +85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
数据保持电压	V_{DDDR}	STOP 模式 (所有功能停止)	1.9		3.6	V
数据保持电流	I_{DDDR}	STOP 模式 (所有功能停止)		7	50	μA
电压上升时间	t_{RVD}		200			μs
电压下降时间	t_{FVD}		200			μs
电压保持时间	t_{HVD}	STOP 模式设置后	0			ms
STOP 释放信号输入时间	t_{DREL}	V_{DD} 达到 2.85 V (MIN.) 后	0			ms
数据保持输入电压, 高	V_{IHDR}	$V_{DD} = EV_{DD} = BV_{DD} = V_{DDDR}$	$0.9V_{DDDR}$		V_{DDDR}	V
数据保持输入电压, 低	V_{ILDR}	$V_{DD} = EV_{DD} = BV_{DD} = V_{DDDR}$	0		$0.1V_{DDDR}$	V

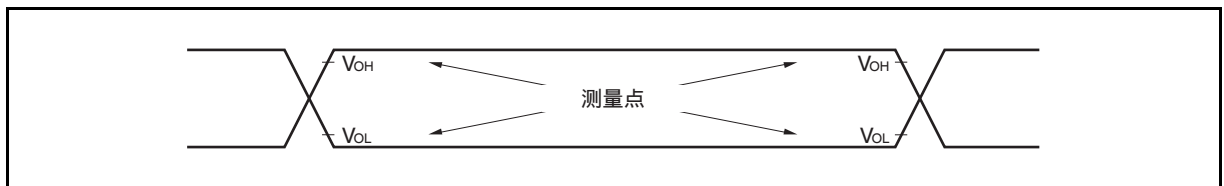
注意事项 转换到 **STOP** 模式和从 **STOP** 模式还原必须在额定的工作范围完成。



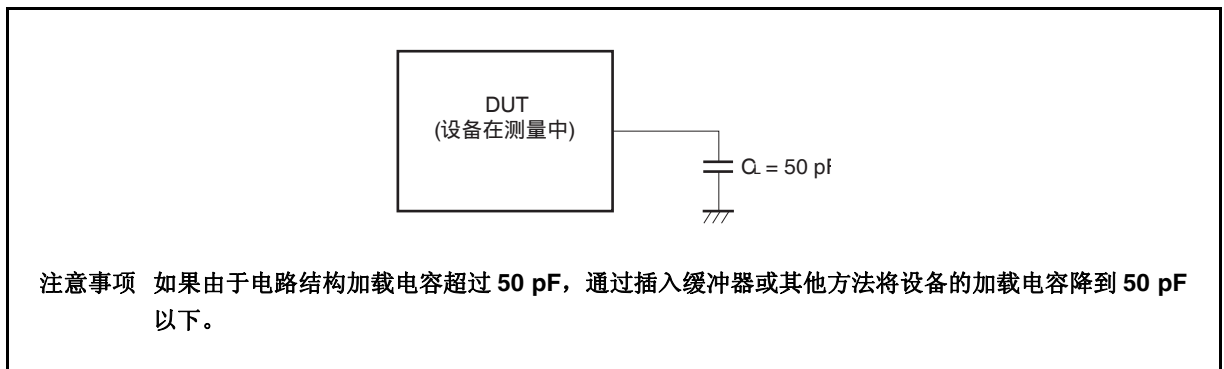
AC 特性

AC 输入测试点 (V_{DD} , AV_{REF0} , AV_{REF1} , EV_{DD} , BV_{DD})

AC 输出测试点



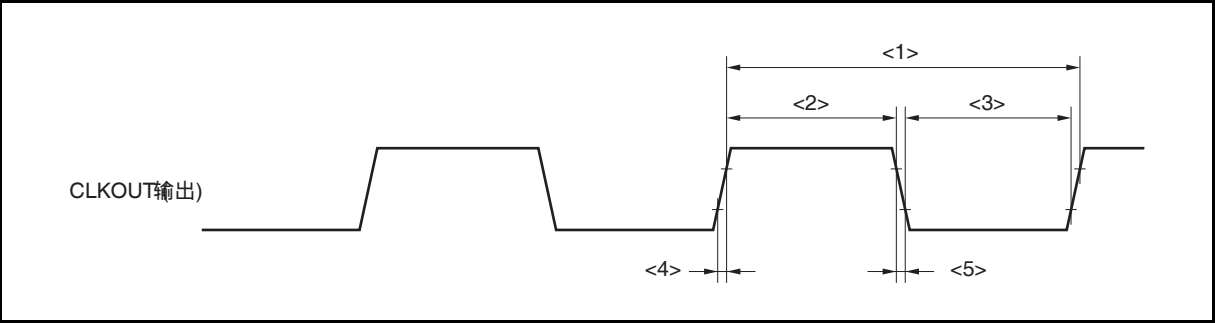
加载条件



CLKOUT 输出时序
(T_A = -40 ~ +85°C, BV_{DD} ≤ V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

参数	符号	条件	MIN.	MAX.	单位
输出周期	t _{cyk}	<1>	50 ns	31.25 μs	
高电平宽度	t _{wKH}	<2>	t _{cyk} /2 - 10		ns
低电平宽度	t _{wKL}	<3>	t _{cyk} /2 - 10		ns
上升时间	t _{kR}	<4>		10	ns
下降时间	t _{kF}	<5>		10	ns

时钟时序



总线时序

(1) 在复用总线模式

(a) 读/写周期 (CLKOUT 异步)

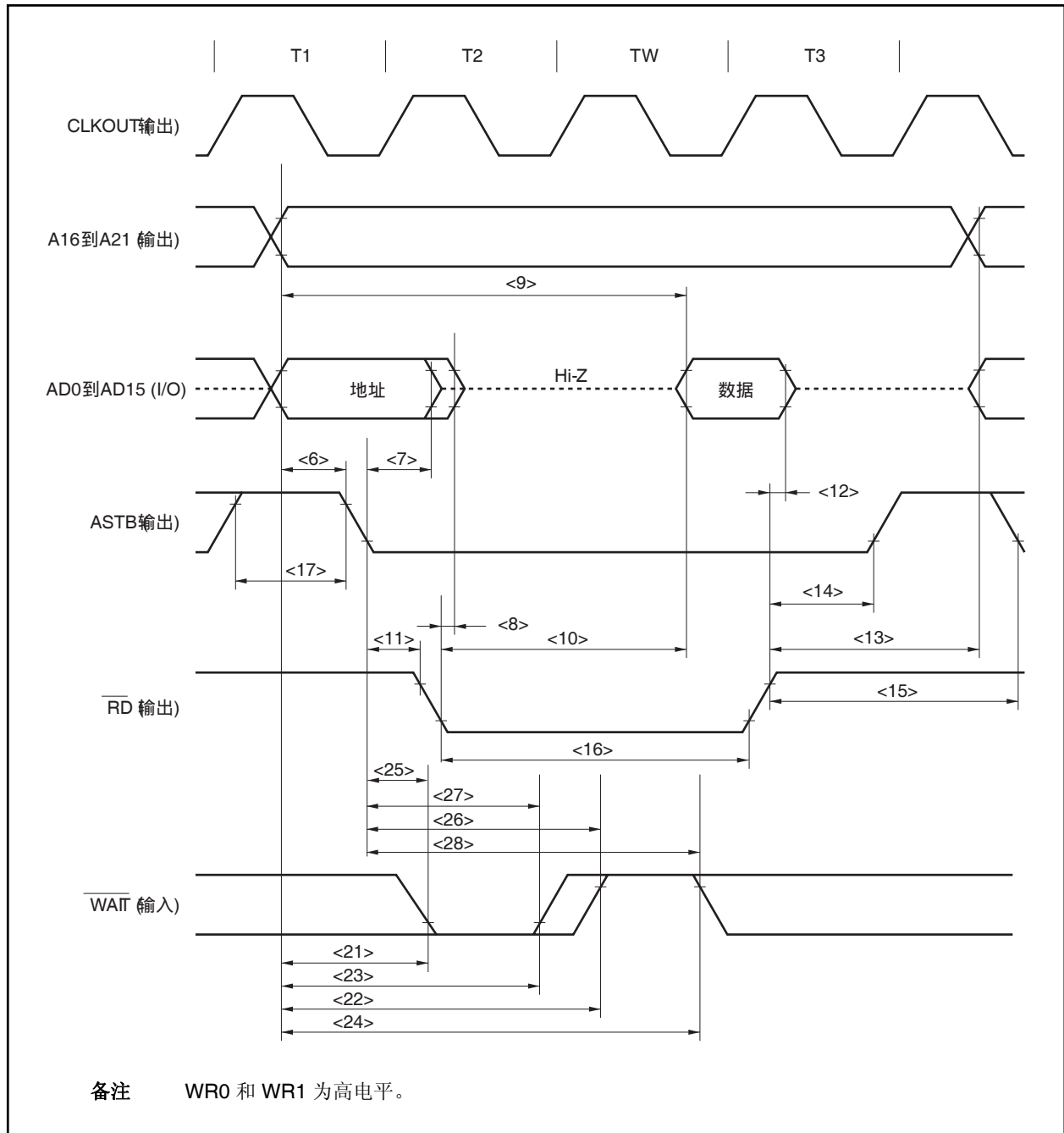
($T_A = -40 \sim +85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
地址建立时间 (到 $ASTB\downarrow$)	t_{SAST}	<6>	$(0.5 + t_{ASW})T - 20$		ns
地址保持时间 (从 $ASTB\downarrow$)	t_{HSTA}	<7>	$(0.5 + t_{AHW})T - 15$		ns
从 $RD\downarrow$ 到地址浮动的延迟时间	t_{FRDA}	<8>		16	ns
从地址数据输入建立时间	t_{SAID}	<9>		$(2 + n + t_{ASW} + t_{AHW})T - 35$	ns
从 $RD\downarrow$ 到数据输入建立时间	t_{SRID}	<10>		$(1 + n)T - 25$	ns
从 $ASTB\downarrow$ 到 RD , $WRm\downarrow$ 的延迟时间	$t_{DSTRDWR}$	<11>	$(0.5 + t_{AHW})T - 15$		ns
数据输入保持时间 (从 $RD\uparrow$)	t_{HRDID}	<12>	0		ns
从 $RD\uparrow$ 地址输出时间	t_{DRDA}	<13>	$(1 + i)T - 15$		ns
从 RD , $WRm\uparrow$ 到 $ASTB\uparrow$ 的延迟时间	$t_{DRDWRST}$	<14>	$0.5T - 15$		ns
从 $RD\uparrow$ 到 $ASTB\downarrow$ 的延迟时间	t_{DRDST}	<15>	$(1.5 + i + t_{ASW})T - 15$		ns
RD , WRm 低电平宽度	t_{WRDWRL}	<16>	$(1 + n)T - 15$		ns
$ASTB$ 高电平宽度	t_{WSTH}	<17>	$(1 + i + t_{ASW})T - 15$		ns
从 $WRm\downarrow$ 数据输出时间	t_{DWROD}	<18>		15	ns
数据输出建立时间 (到 $WRm\uparrow$)	t_{SODWR}	<19>	$(1 + n)T - 20$		ns
数据输出保持时间 (从 $WRm\uparrow$)	t_{HWROD}	<20>	$T - 15$		ns
WAIT 建立时间 (到地址)	t_{SAWT1}	<21> $n \geq 1$		$(1.5 + t_{ASW} + t_{AHW})T - 35$	ns
	t_{SAWT2}	<22>		$(1.5 + n + t_{ASW} + t_{AHW})T - 35$	ns
WAIT 保持时间 (从地址)	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 (到 $ASTB\downarrow$)	t_{SSTWT1}	<25> $n \geq 1$		$(1 + t_{AHW})T - 25$	ns
	t_{SSTWT2}	<26>		$(1 + n + t_{AHW})T - 25$	ns
WAIT 保持时间 (从 $ASTB\downarrow$)	t_{HSTWT1}	<27> $n \geq 1$	$(n + t_{AHW})T$		ns
	t_{HSTWT2}	<28>	$(1 + n + t_{AHW})T$		ns

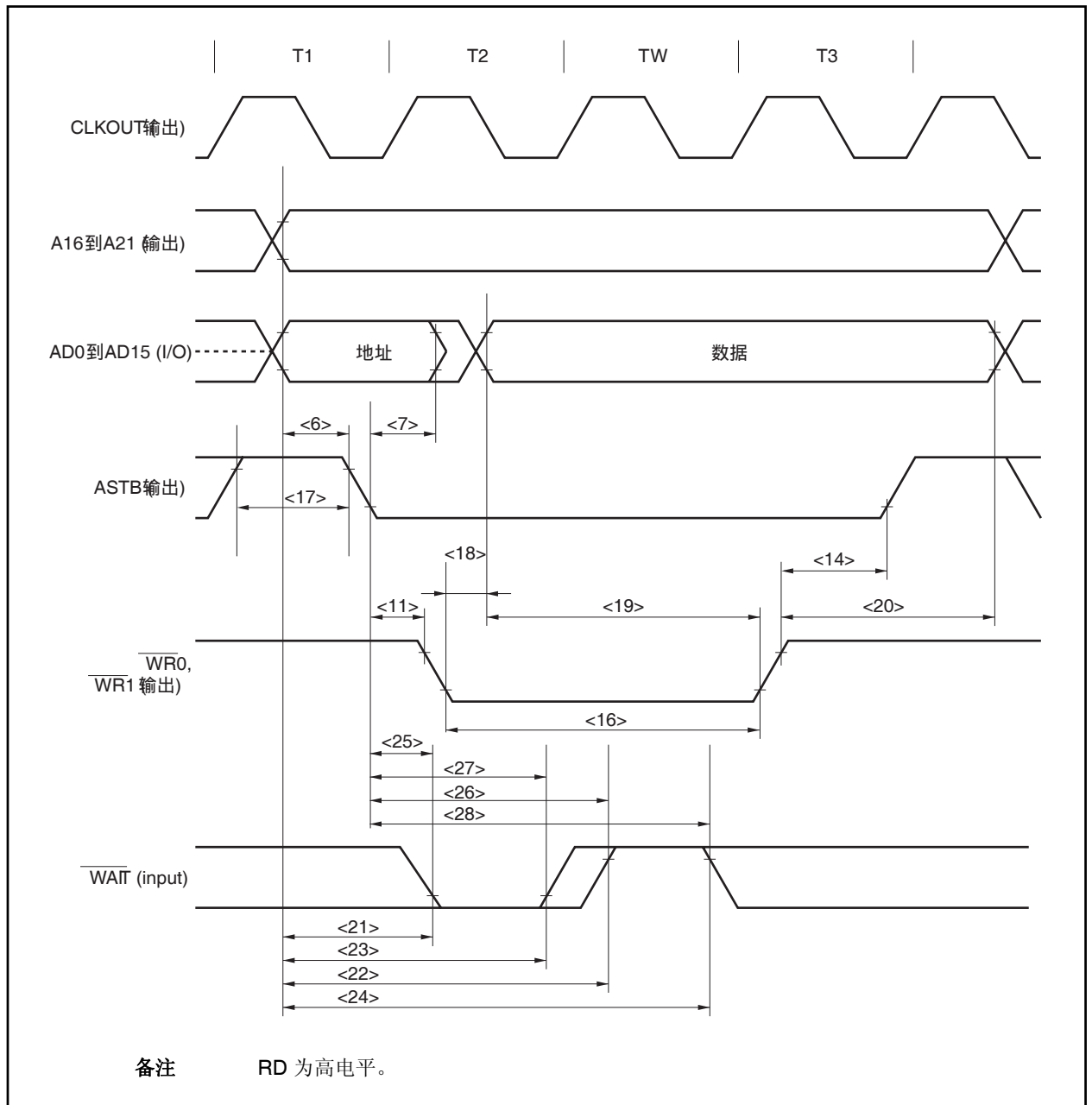
备注

1. t_{ASW} : 地址建立等待时钟的个数
 t_{AHW} : 地址保持等待时钟的个数
2. $T = 1/f_{CPU}$ (f_{CPU} : CPU 工作时钟频率)
3. n : 在总线周期插入的等待时钟个数
 当插入可编程等待时采样时序改变。
4. $m = 0, 1$
5. i : 在一个读周期后插入的闲置状态的序号 (0 或 1)
6. 上面指定的值是当从 X1 输入的时钟占空比为 1:1 时的值。

读周期 (CLKOUT A 异步): 在复用总线模式



写周期 (CLKOUT 异步): 在复用总线模式



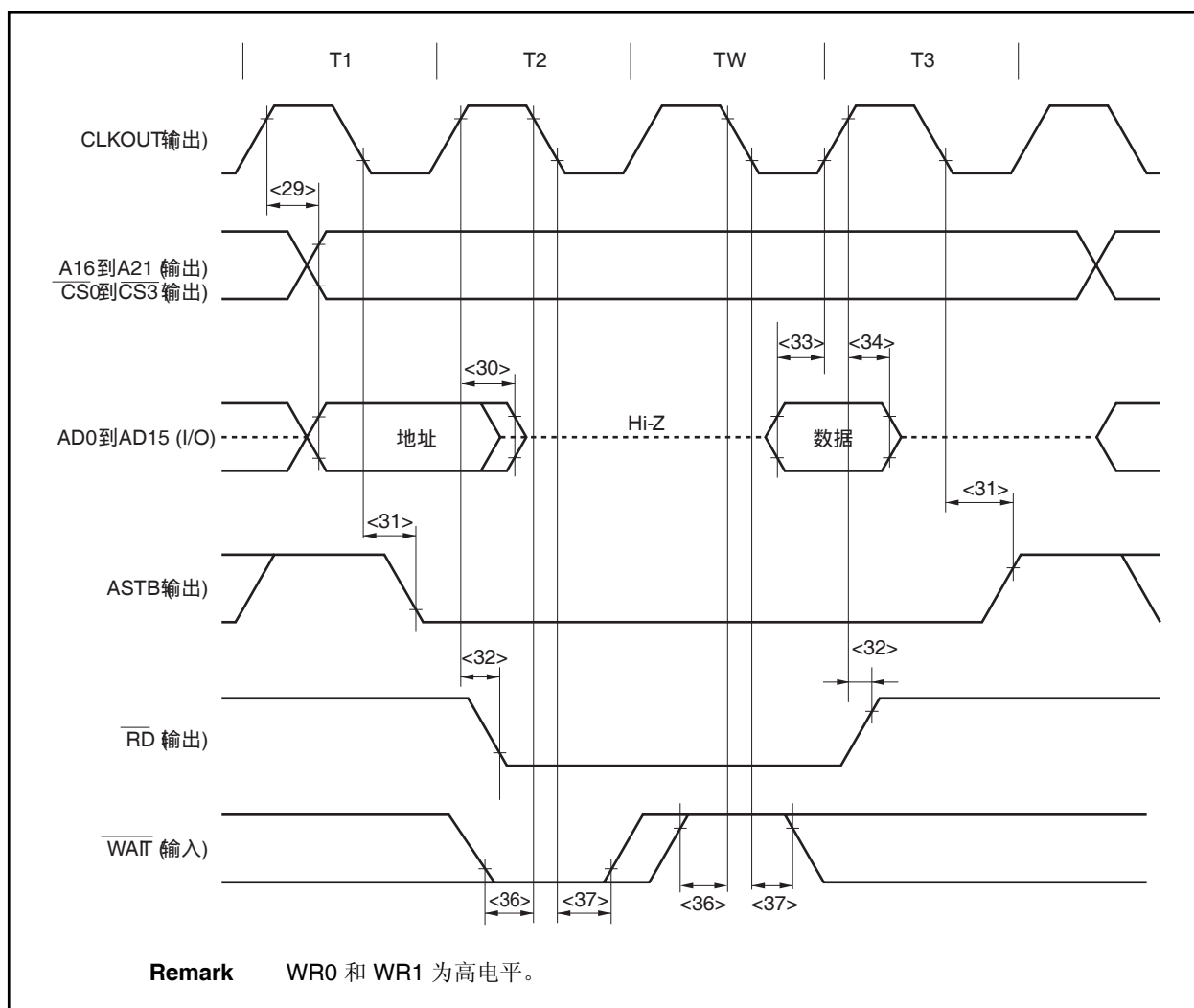
(b) 读/写周期 (CLKOUT 同步): 在复用总线模式

(TA = -40 ~ +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V, CL = 50 pF)

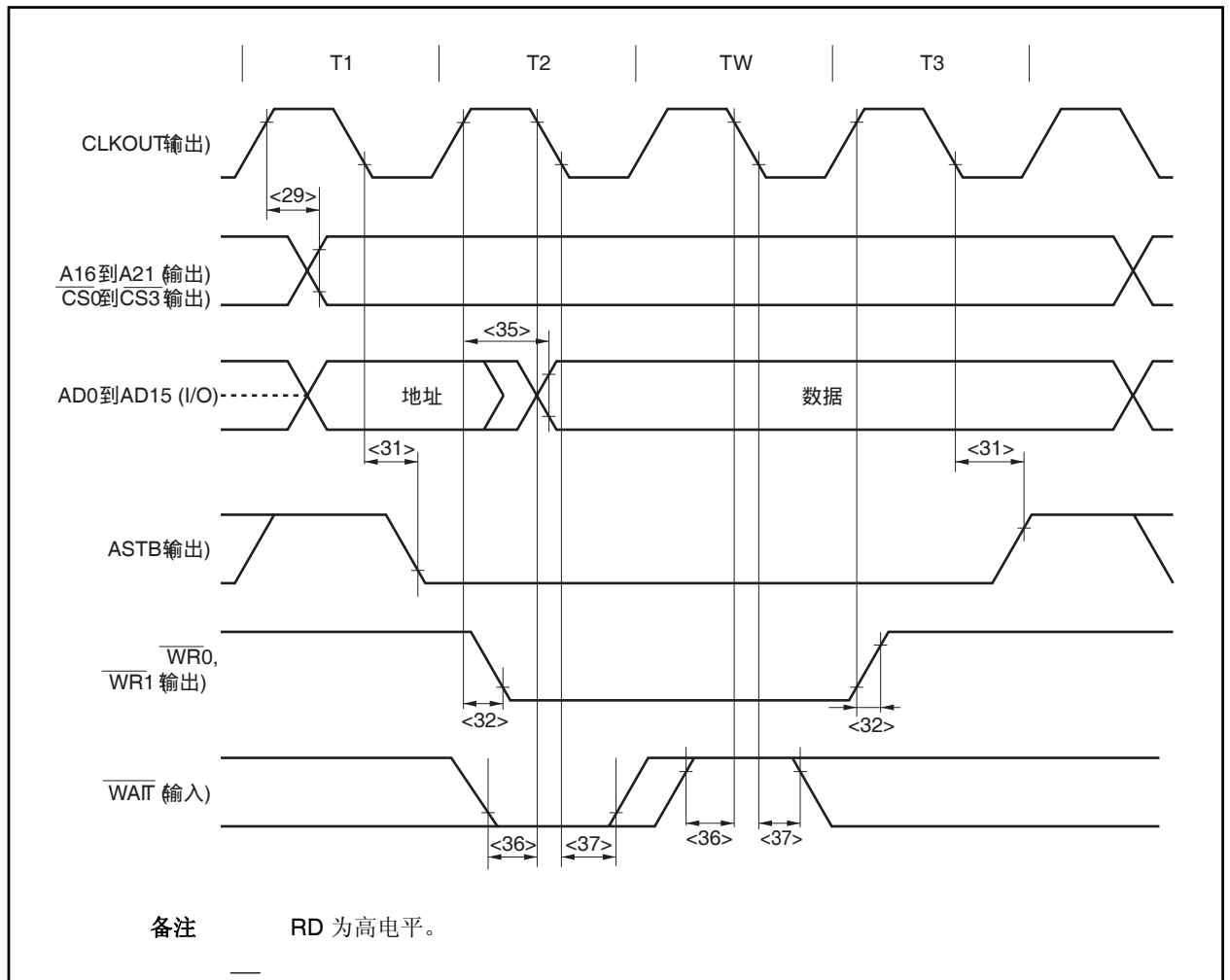
参数	符号	条件	MIN.	MAX.	单位
从 CLKOUT↑ 到 地址的延迟时间	tDKA	<29>	0	25	ns
从 CLKOUT↑ 到地址浮动的延迟时间	tFKA	<30>	0	19	ns
从 CLKOUT↓ 到 ASTB 的延迟时间	tDKST	<31>	-12	7	ns
从 CLKOUT↑ 到 RD, WRm 的延迟时间	tDKRDWR	<32>	-5	14	ns
数据输入建立时间 (到 CLKOUT↑)	tSIDK	<33>	15		ns
数据输入保持时间 (从 CLKOUT↑)	tHKID	<34>	5		ns
从 CLKOUT↑ 数据输出延迟时间	tDKOD	<35>		19	ns
WAIT 建立时间 (到 CLKOUT↓)	tSWTK	<36>	20		ns
WAIT 保持时间 (从 CLKOUT↓)	tHKWT	<37>	5		ns

- 备注
1. m = 0, 1
 2. 上面指定的值是当从 X1 输入的时钟占空比为 1:1 时的值。

读周期 (CLKOUT 同步): 在复用总线模式



写周期 (CLKOUT 同步): 在复用总线模式



(2) 在独立总线模式

(a) 读周期 (CLKOUT 异步): 独立总线模式

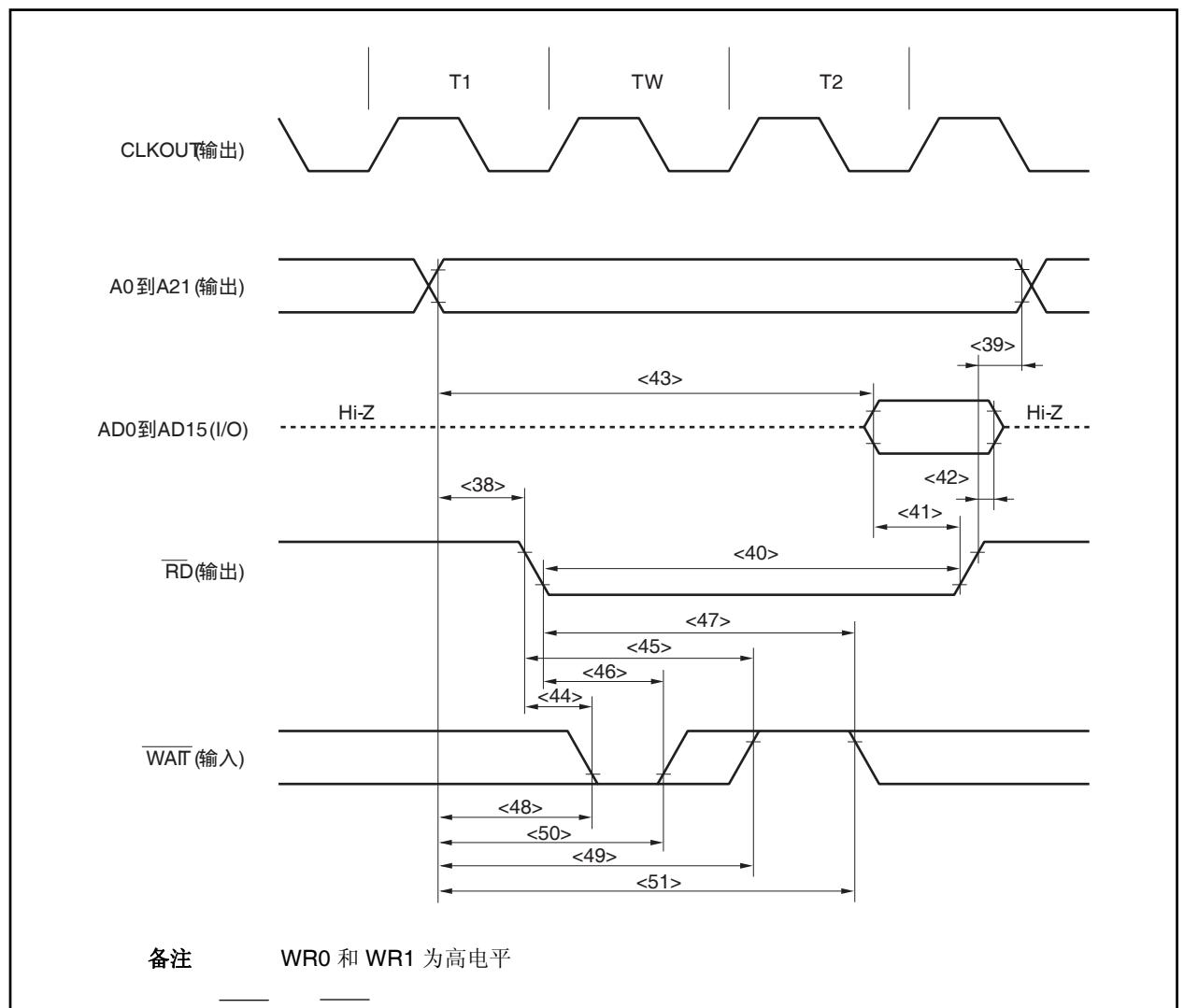
(T_A = -40 ~ +85°C, V_{DD} ≤ V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号	条件	MIN.	MAX.	单位
地址建立时间 (到 RD↓)	t _{SARD}	<38>	(0.5 + t _{ASW})T - 23		ns
地址保持时间 (从 RD↑)	t _{HARD}	<39>	iT + 1		ns
RD 低电平宽度	t _{WRDL}	<40>	(1.5 + n + t _{AHW})T - 10		ns
数据建立时间 (到 RD↑)	t _{SISD}	<41>	23		ns
数据保持时间 (从 RD↑)	t _{HISD}	<42>	0		ns
数据建立时间 (到地址)	t _{SAID}	<43>		(2 + n + t _{ASW} + t _{AHW})T - 40	ns
WAIT 建立时间 (到 RD↓)	t _{SRDWT1}	<44>		(0.5 + t _{AHW})T - 25	ns
	t _{SRDWT2}	<45>		(0.5 + n + t _{AHW})T - 25	ns
WAIT 保持时间 (从 RD↓)	t _{HRDWT1}	<46>	(n - 0.5 + t _{AHW})T		ns
	t _{HRDWT2}	<47>	(n + 0.5 + t _{AHW})T		ns
WAIT 建立时间 (到地址)	t _{SAWT1}	<48>		(1 + t _{ASW} + t _{AHW})T - 45	ns
	t _{SAWT2}	<49>		(1 + n + t _{ASW} + t _{AHW})T - 45	ns
WAIT 保持时间 (从地址)	t _{HAWT1}	<50>	(n + t _{ASW} + t _{AHW})T		ns
	t _{HAWT2}	<51>	(1 + n + t _{ASW} + t _{AHW})T		ns

备注

1. t_{ASW}: 地址建立等待时钟的个数
t_{AHW}: 地址保持等待时钟的个数
2. T = 1/f_{CPU} (f_{CPU}: CPU 工作时钟频率)
3. n: 在总线周期插入的等待时钟个数
当插入可编程等待时采样时序改变。
4. i: 在一个读周期后插入的闲置状态的序号 (0 或 1)。
5. 上面指定的值是当从 X1 输入的时钟占空比为 1:1 时的值。

读周期(CLKOUT 异步): 独立总线模式



(b) 写周期 (CLKOUT 异步): 独立总线模式

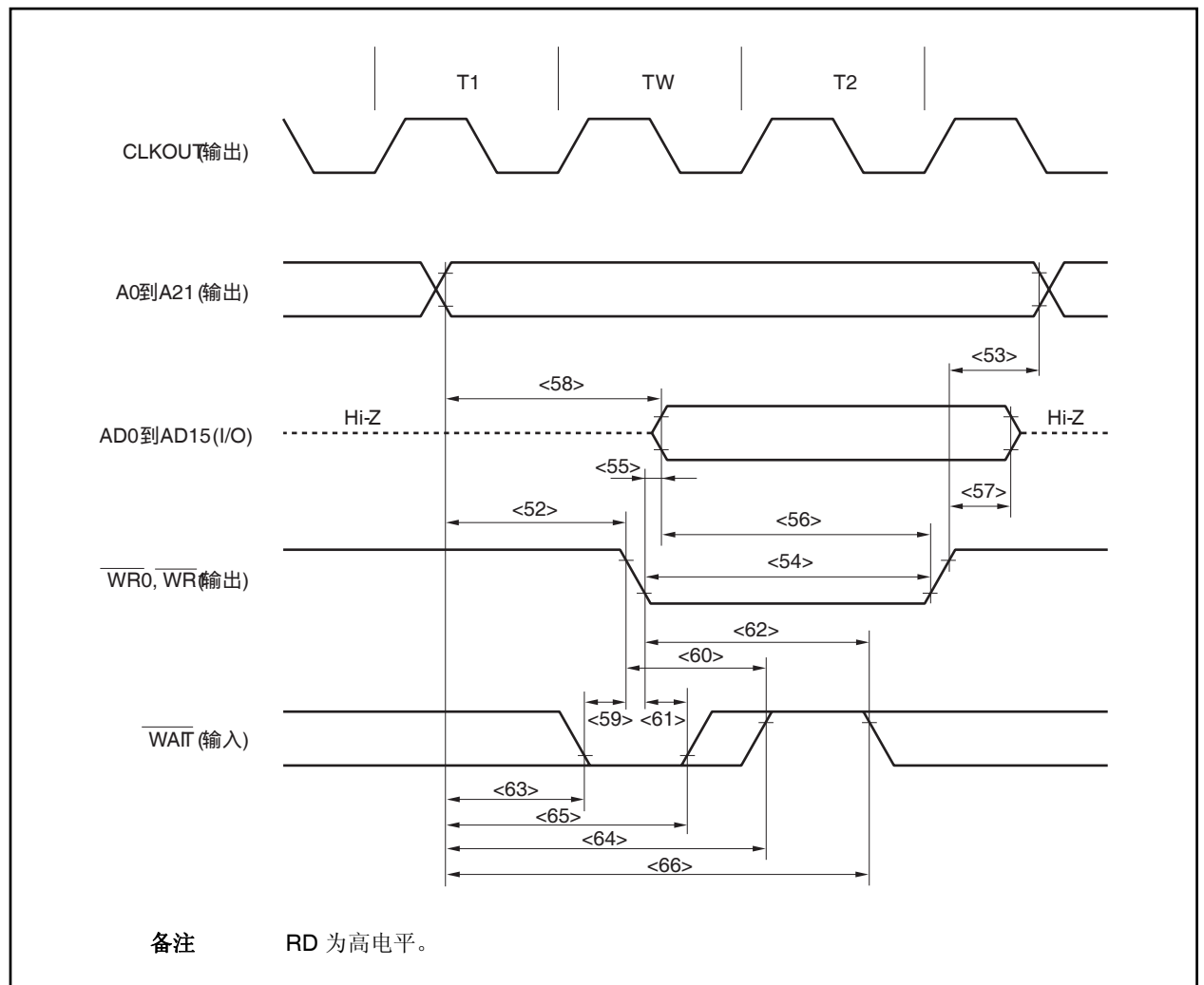
(TA = -40 ~ +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	MIN.	MAX.	单位
地址建立时间 (到 $\overline{WRm}\downarrow$)	tSAWR	<52>	$(1 + t_{ASW} + t_{AHW})T - 23$		ns
地址保持时间 (从 $\overline{WRm}\uparrow$)	tHAWR	<53>	$0.5T - 10$		ns
\overline{WRm} 低电平宽度	tWWRL	<54>	$(0.5 + n)T - 10$		ns
数据输出时间 ($\overline{WRm}\downarrow$)	tDOSDW	<55>	-5		ns
数据建立时间 (到 $\overline{WRm}\uparrow$)	tSOSDW	<56>	$(0.5 + n)T - 20$		ns
数据保持时间 (从 $\overline{WRm}\uparrow$)	tHOSDW	<57>	$0.5T - 10$		ns
数据建立时间 (到地址)	tSAOD	<58>	$(1 + t_{ASW} + t_{AHW})T - 25$		ns
\overline{WAIT} 建立时间 (到 $\overline{WRm}\downarrow$)	tSWRWT1	<59>	22		ns
	tSWRWT2	<60>		$nT - 22$	ns
\overline{WAIT} 保持时间 (从 $\overline{WRm}\downarrow$)	tHWRWT1	<61>	0		ns
	tHWRWT2	<62>	nT		ns
\overline{WAIT} 建立时间 (到地址)	tSAWT1	<63>		$(1 + t_{ASW} + t_{AHW})T - 45$	ns
	tSAWT2	<64>		$(1 + n + t_{ASW} + t_{AHW})T - 45$	ns
\overline{WAIT} 保持时间 (从地址)	tHAWT1	<65>	$(n + t_{ASW} + t_{AHW})T$		ns
	tHAWT2	<66>	$(1 + n + t_{ASW} + t_{AHW})T$		ns

备注

1. $m = 0, 1$
2. t_{ASW} : 地址建立等待时钟的个数
 t_{AHW} : 地址保持等待时钟的个数
3. $T = 1/f_{CPU}$ (f_{CPU} : CPU 工作时钟频率)
4. n : 在总线周期插入的等待时钟个数
当插入可编程等待时采样时序改变。
5. 上面指定的值是当从 X1 输入的时钟占空比为 1:1 时的值。

写周期 (CLKOUT 异步): 独立总线模式



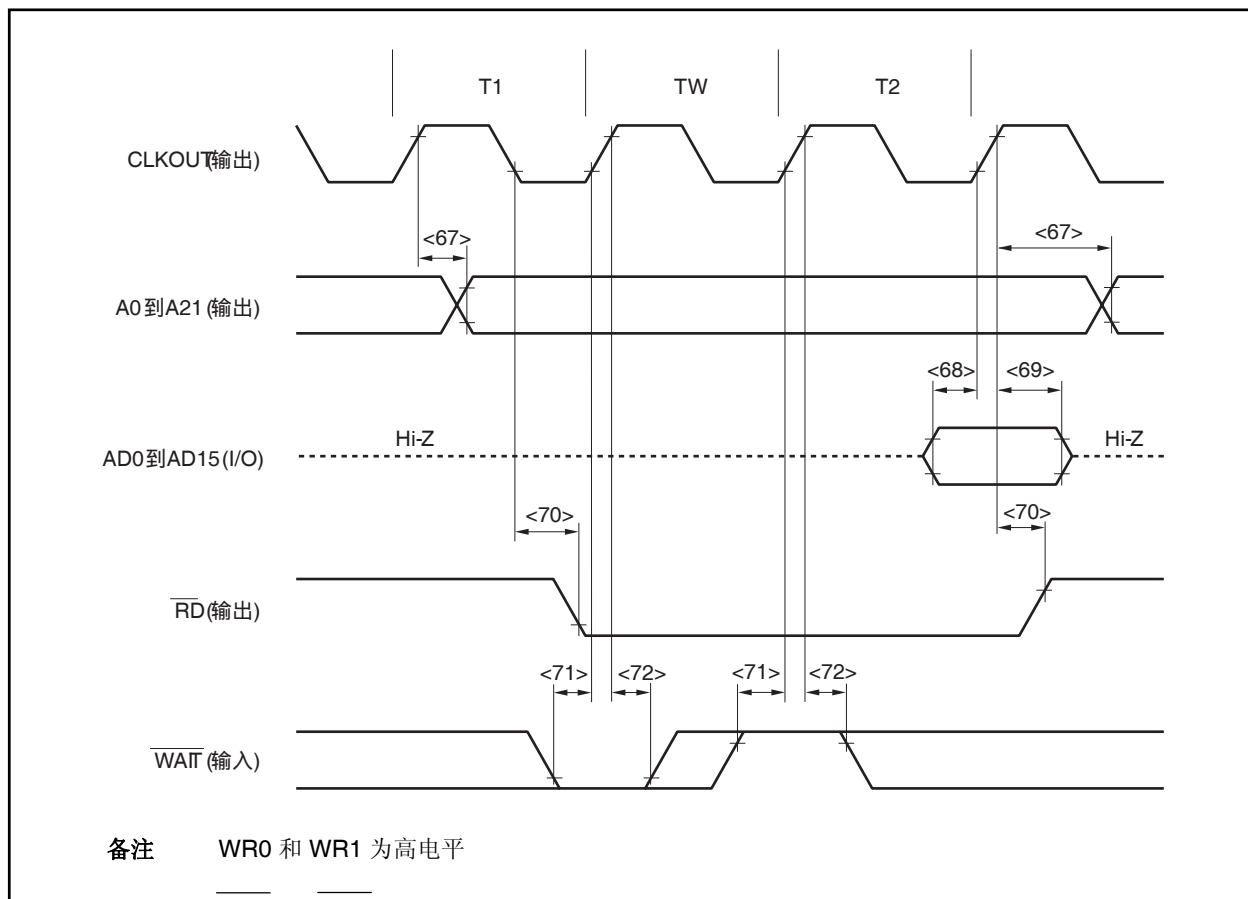
(c) 读周期 (CLKOUT 同步): 独立总线模式

($T_A = -40 \sim +85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
从 CLKOUT↑ 到地址的延迟时间, CS	t_{DKSA}	<67>	2	25	ns
数据输入建立时间 (到 CLKOUT↑)	t_{SISDK}	<68>	20		ns
数据输入保持时间 (从 CLKOUT↑)	t_{HKISD}	<69>	0		ns
从 CLKOUT↓ 到 \overline{RD} 的延迟时间	t_{DKSR}	<70>	-2	12	ns
\overline{WAIT} 建立时间 (到 CLKOUT↑)	t_{SWTK}	<71>	20		ns
\overline{WAIT} 保持时间 (从 CLKOUT↑)	t_{HKWT}	<72>	0		ns

备注 上面指定的值是当从 X1 输入的时钟占空比为 1:1 时的值。

读周期 (CLKOUT 同步, 1 Wait): 独立总线模式



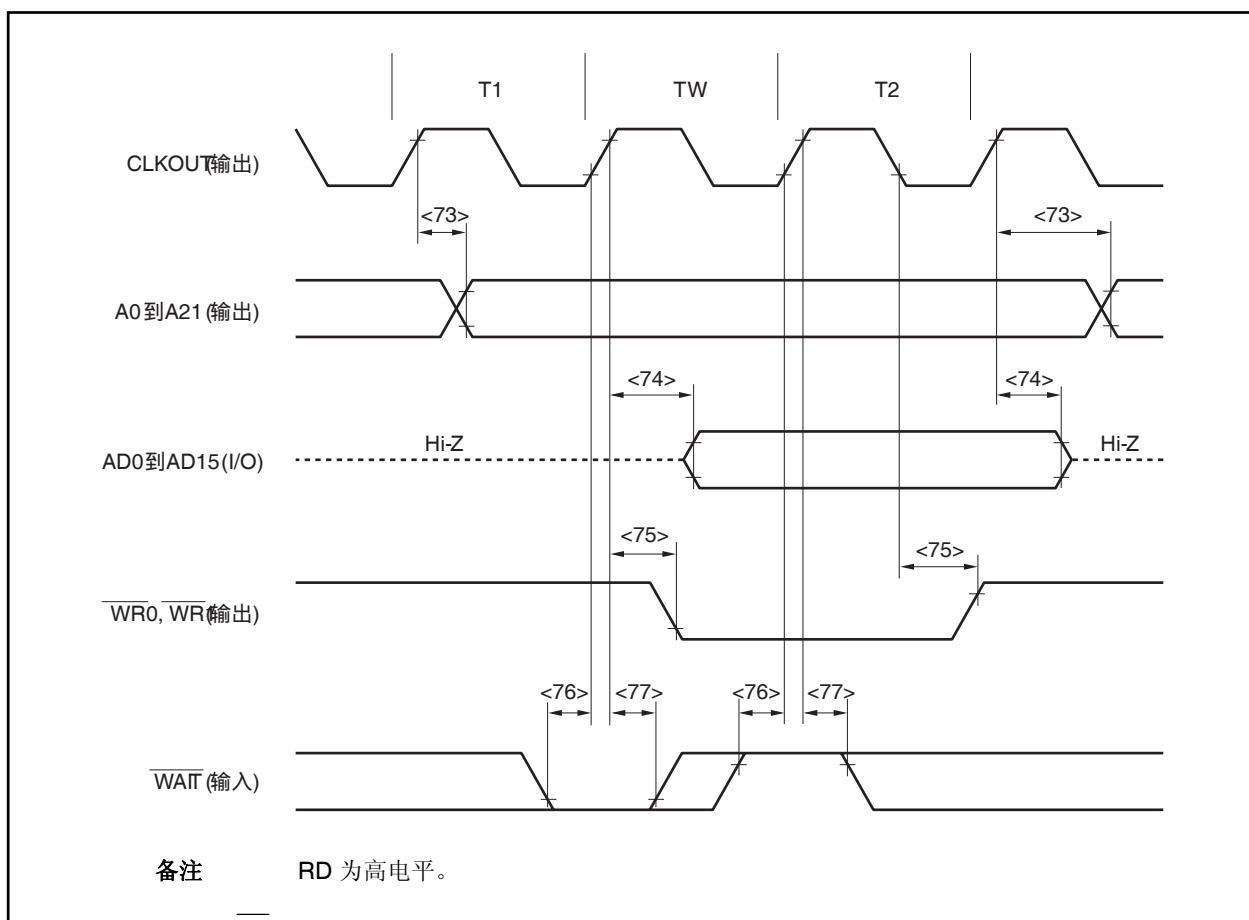
(d) 写周期 (CLKOUT 同步): 独立总线模式

(TA = -40 ~ +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	MIN.	MAX.	单位
从 CLKOUT↑到地址的延迟时间, CS	t _{DKSA}	<73>	2	25	ns
从 CLKOUT↑到数据输出的延迟时间	t _{DKSD}	<74>	2	15	ns
从 CLKOUT↑↓到 \overline{WRm} 的延迟时间	t _{DKSW}	<75>	-2	12	ns
\overline{WAIT} 设置时间 (到 CLKOUT↑)	t _{SWTK}	<76>	20		ns
\overline{WAIT} 保持时间 (从 CLKOUT↑)	t _{HKWT}	<77>	0		ns

- 备注
1. m = 0, 1
 2. 上面指定的值是当从 X1 输入的时钟占空比为 1:1 时的值。

写周期 (CLKOUT 同步): 独立总线模式



(3) 总线保持

(a) CLKOUT 异步

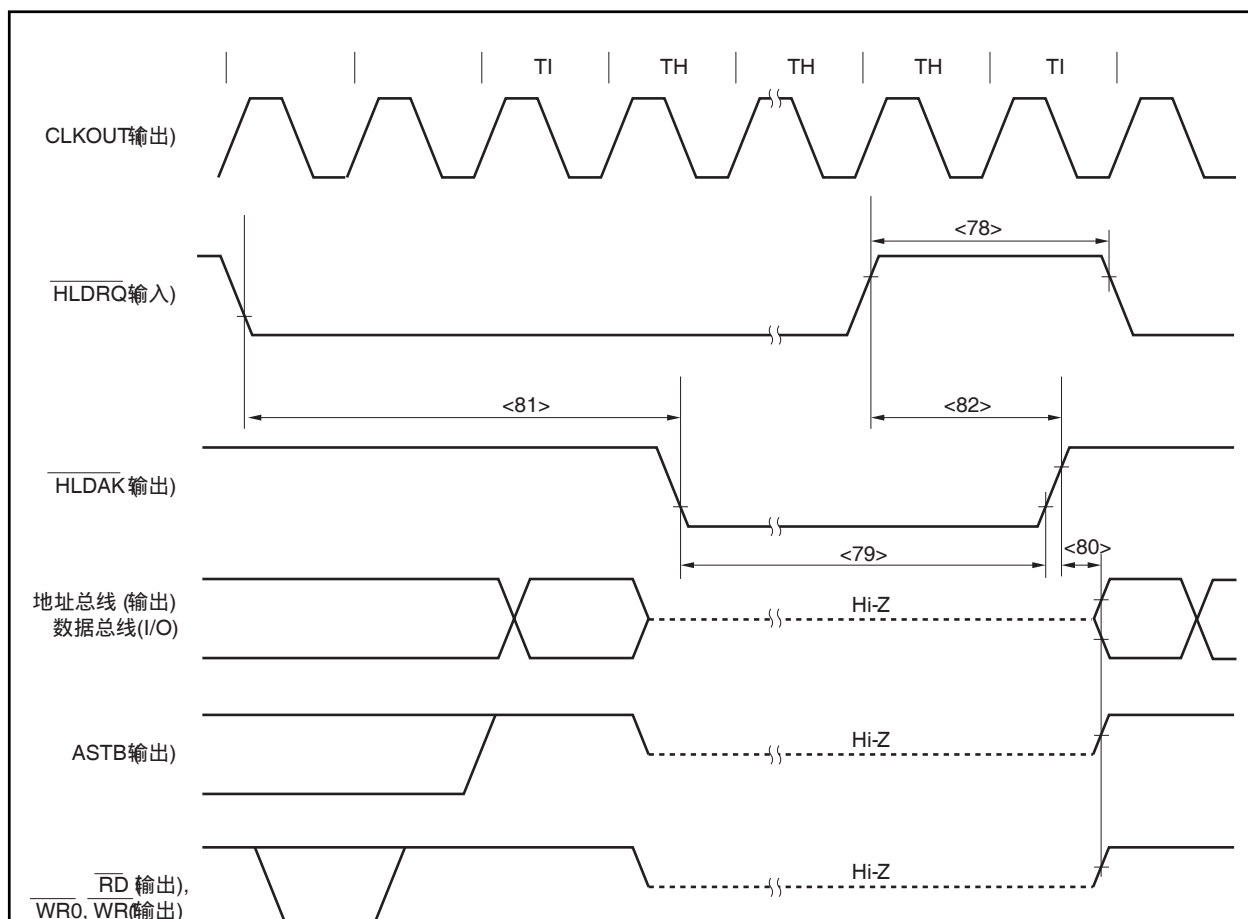
($T_A = -40 \sim +85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
HLD $\overline{\text{RQ}}$ 高电平宽度	t_{WHQH}	<78>	$T + 10$		ns
HLD $\overline{\text{AK}}$ 低电平宽度	t_{WHAL}	<79>	$T - 15$		ns
从 HLD $\overline{\text{AK}}\uparrow$ 到总线输出的延迟时间	t_{DHAC}	<80>	-3		ns
从 HLD $\overline{\text{RQ}}\downarrow$ 到 HLD $\overline{\text{AK}}$ 的延迟时间 \downarrow	t_{DHQA1}	<81>		$(2n + 7.5)T + 25$	ns
从 HLD $\overline{\text{RQ}}\uparrow$ 到 HLD $\overline{\text{AK}}\uparrow$ 的延迟时间	t_{DHQA2}	<82>	$0.5T$	$1.5T + 25$	ns

备注

1. $T = 1/f_{\text{CPU}}$ (f_{CPU} : CPU 工作时钟频率)
2. n : 在总线周期插入的等待时钟个数
当插入可编程等待时采样时序改变。
3. 上面指定的值是当从 X1 输入的时钟占空比为 1:1 时的值。

总线保持 (CLKOUT 异步)



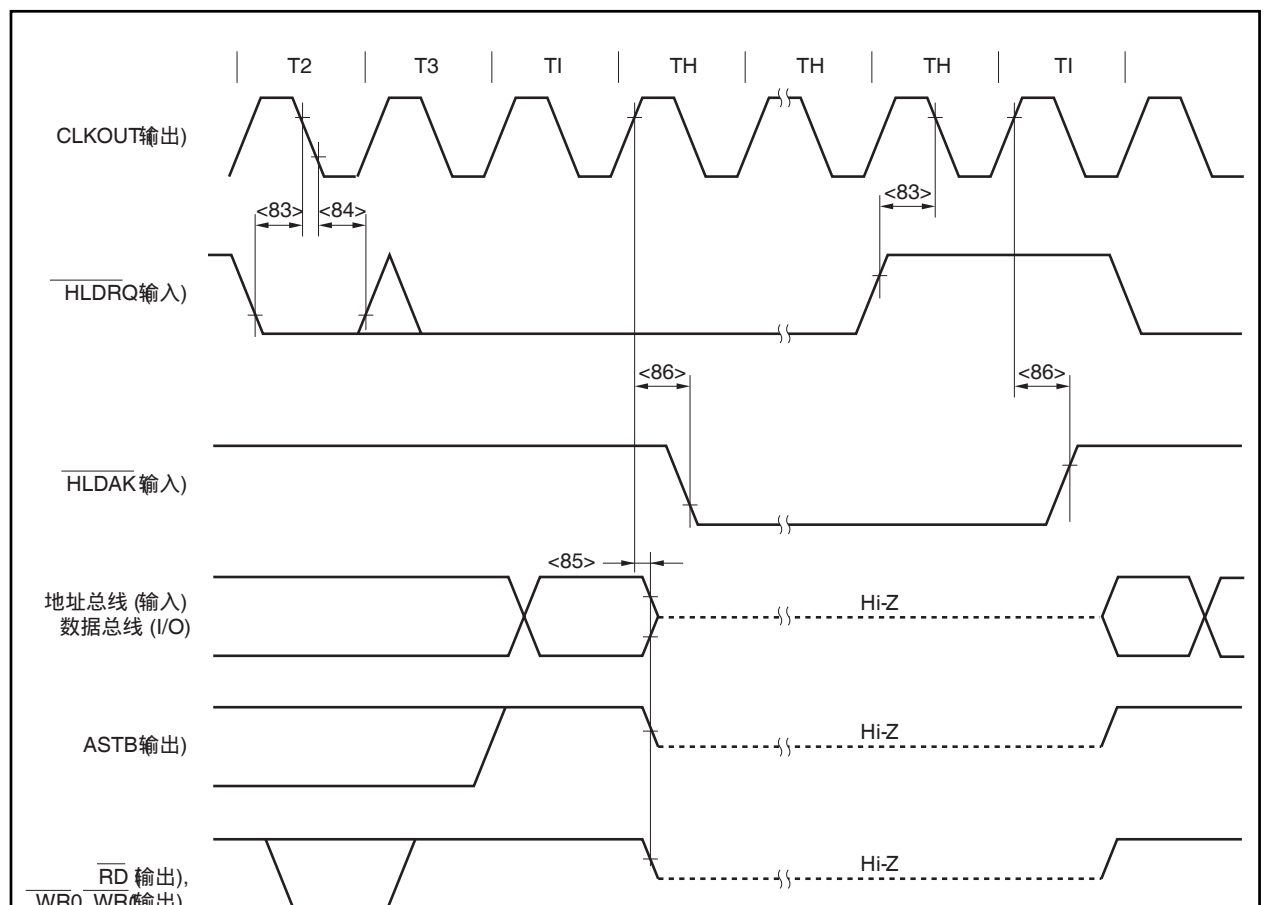
(b) CLKOUT 同步

($T_A = -40 \sim +85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
HLD $\overline{\text{RQ}}$ 建立时间 (到 CLKOUT \downarrow)	t_{SHQK}	<83>	20		ns
HLD $\overline{\text{RQ}}$ 保持时间 (从 CLKOUT \downarrow)	t_{HKHQ}	<84>	5		ns
从 CLKOUT \uparrow 到总线浮动的延迟时间	t_{DKF}	<85>		19	ns
从 CLKOUT \uparrow 到 HLD $\overline{\text{AK}}$ 的延迟时间	t_{DKHA}	<86>		19	ns

备注 上面指定的值是当从 X1 输入的时钟占空比为 1:1 时的值。

总线保持 (CLKOUT 同步)

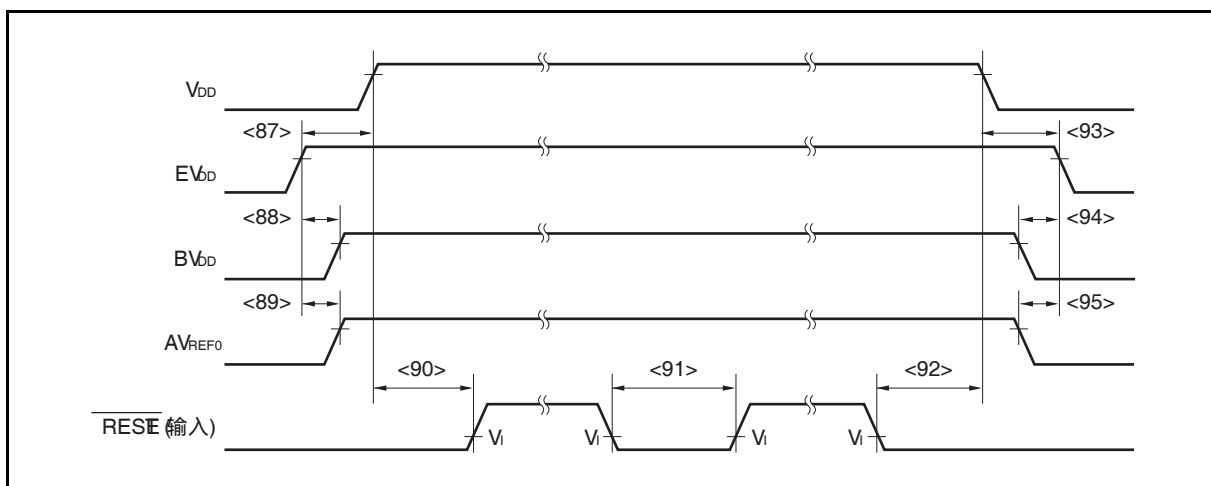


上电/下电/复位时序

(TA = -40 ~ +85°C, VSS = AVSS = BVSS = EVSS = 0 V, CL = 50 pF)

参数	符号	条件	MIN.	MAX.	单位
EVDD↑ → VDD↑	tREL	<87>	0		ns
EVDD↑ → BVDD↑	tREB	<88>	0	tREL	ns
EVDD↑ → AVREF0, AVREF1↑	tREA	<89>	0	tREL	ns
EVDD↑ → RESET↑	tRER	<90>	500 + tREG [#]		ns
RESET 低电平宽度	tWRSL	模拟噪声消除(擦除/写入期间)	500		ns
		模拟噪声消除	500		ns
RESET↓ VDD↓	tFRE	<92>	500		ns
VDD↓ EVDD↓	tFEL	<93>	0		ns
BVDD↓ EVDD↓	tFEB	<94>	0	tFEL	ns
AVREF0↓ EVDD↓	tFEA	<95>	0	tFEL	ns

注 根据片上调节器特性特性。



中断, FLMD0 引脚时序

(TA = -40 ~ +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	MIN.	MAX.	单位
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 ~ 7 (模拟噪声消除)	500		ns
		n = 3 (数字噪声消除)	3T _{SMP} + 20		ns
INTPn 低电平宽度	tWITL	n = 0 ~ 7 (模拟噪声消除)	500		ns
		n = 3 (数字噪声消除)	3T _{SMP} + 20		ns
FLMD0 高电平宽度	tWMDH		500		ns
FLMD0 低电平宽度	tWMDL		500		ns

备注 T_{SMP}: 噪声消除采样时钟周期

按键返回时序

(T_A = -40 ~ +85°C, B_{VDD} ≤ V_{DD} = E_{VDD} = A_{VREF0} = A_{VREF1}, V_{SS} = E_{VSS} = B_{VSS} = A_{VSS} = 0 V, C_L = 50 pF)

参数	符号	条件	MIN.	MAX.	单位
KRn 高电平宽度	t _{WKRH}	模拟噪声消除	500		ns
KRn 低电平宽度	t _{WKRL}	模拟噪声消除	500		ns

备注 n = 0 ~ 7

定时器时序

(T_A = -40 ~ +85°C, B_{VDD} ≤ V_{DD} = E_{VDD} = A_{VREF0} = A_{VREF1}, V_{SS} = E_{VSS} = B_{VSS} = A_{VSS} = 0 V, C_L = 50 pF)

参数	符号	条件	MIN.	MAX.	单位
TI 高电平宽度	t _{TIH}	TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, TIP31, TIP40, TIP41, TIP50, TIP51,	2T + 20		ns
TI 低电平宽度	t _{TIL}	TIQ00 ~ TIQ03	2T + 20		ns

备注 T = 1/f_{xx}

UART 时序

(T_A = -40 ~ +85°C, B_{VDD} ≤ V_{DD} = E_{VDD} = A_{VREF0} = A_{VREF1}, V_{SS} = E_{VSS} = B_{VSS} = A_{VSS} = 0 V, C_L = 50 pF)

参数	符号	条件	MIN.	MAX.	单位
发送速率				312.5	kbps
ASCK0 周期时间				10	MHz

CSIB 时序

(1) 主设备模式

(2) ($T_A = -40 \sim +85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

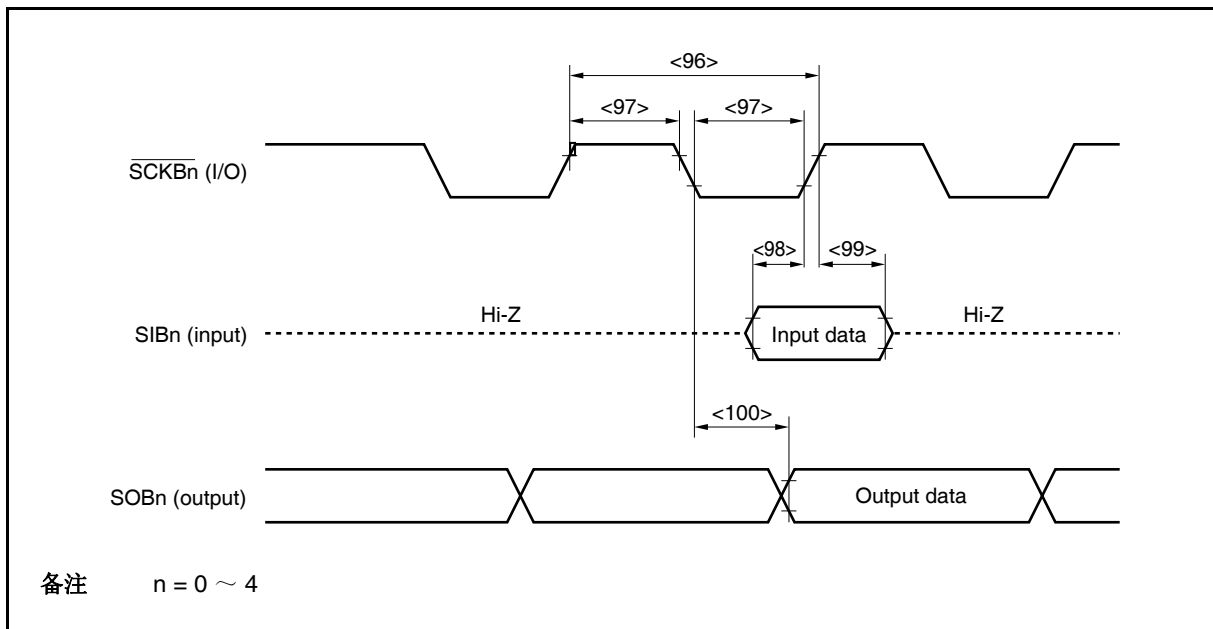
参数	符号	条件	MIN.	MAX.	单位
$\overline{\text{SCKBn}}$ 周期时间	t_{KCY1}	<96>	125		ns
$\overline{\text{SCKBn}}$ 高-/低电平宽度	t_{KH1} , t_{KL1}	<97>	$t_{\text{KCY1}}/2 - 5$		ns
SIBn 建立时间(到 $\overline{\text{SCKBn}}\uparrow$)	t_{SIK1}	<98>	30		ns
SIBn 保持时间(从 $\overline{\text{SCKBn}}\uparrow$)	t_{KSI1}	<99>	30		ns
从 $\overline{\text{SCKBn}}\downarrow$ 到 SOBn 输出的延迟时间	t_{KSO1}	<100>		30	ns

备注 $n = 0 \sim 4$

(3) 从设备模式

(4) ($T_A = -40 \sim +85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
$\overline{\text{SCKBn}}$ 周期时间	t_{KCY2}	<96>	125		ns
$\overline{\text{SCKBn}}$ 高-/低电平宽度	t_{KH2} , t_{KL2}	<97>	57.5		ns
SIBn 建立时间 (到 $\overline{\text{SCKBn}}\uparrow$)	t_{SIK2}	<98>	30		ns
SIBn 保持时间 (从 $\overline{\text{SCKBn}}\uparrow$)	t_{KSI2}	<99>	30		ns
从 $\overline{\text{SCKBn}}\downarrow$ 到 SOBn 输出的延迟时间	t_{KSO2}	<100>		30	ns

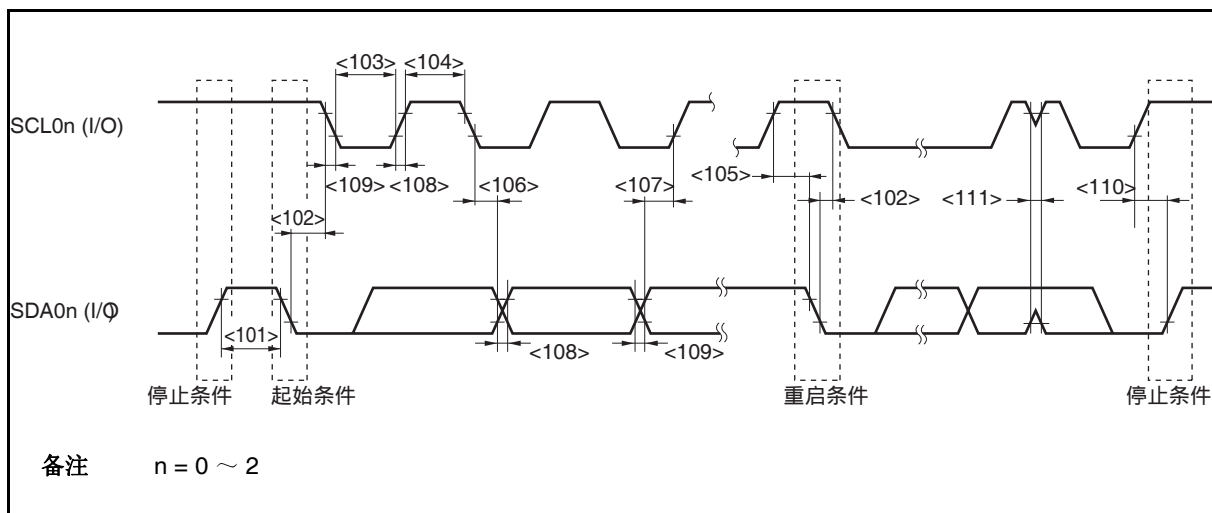
备注 $n = 0 \sim 4$ 

I²C Bus 模式(T_A = -40 ~ +85°C, V_{DD} ≤ V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数		符号		正常模式		高速模式		单位
				MIN.	MAX.	MIN.	MAX.	
SCL0n 时钟频率		fCLK		0	100	0	400	kHz
总线自由时间 (从启动到停止的条件)		tBUF	<101>	4.7	—	1.3	—	μs
保持时间 ^{※1}		tHD:STA	<102>	4.0	—	0.6	—	μs
SCL0n 时钟低电平宽度		t低	<103>	4.7	—	1.3	—	μs
SCL0n 时钟高电平宽度		t高	<104>	4.0	—	0.6	—	μs
启动/重启条件的建立时间		tSU:STA	<105>	4.7	—	0.6	—	μs
数据保持时间	CBUS兼容主设备	tHD:DAT	<106>	5.0	—	—	—	μs
	I ² C 模式			0 ^{※2}	—	0 ^{※2}	0.9 ^{※3}	μs
数据建立时间		tSU:DAT	<107>	250	—	100 ^{※4}	—	ns
SDA0n和 SCL0n 信号上升时间		tR	<108>	—	1000	20 + 0.1Cb ^{※5}	300	ns
SDA0n和 SCL0n 信号下降时间		tF	<109>	—	300	20 + 0.1Cb ^{※5}	300	ns
停止条件建立时间		tSU:STO	<110>	4.0	—	0.6	—	μs
输入滤波器尖峰抑制的脉冲宽度		tSP	<111>	—	—	0	50	ns
每个总线的加载电容		Cb		—	400	—	400	pF

- 注
1. 在起始条件中，第一个时钟脉冲产生在保持时间后。
 2. 系统内 SDA0n(在 SCL0n 信号的 V_{IHmin})信号需要一个最少 300ns 的保持时间，以使在 SCL0n 的下降沿时占用未定义区域。
 3. 如果系统不延长 SCL0n 信号低的保持时间(t_{LOW})，仅需要满足最大数据保持时间(t_{HD:DAT})。
 4. 高速模式 I²C 总线可以用于正常模式 I²C 总线系统。在这种情况下，设定高速模式 I²C 总线，因此会遇到如下条件。
 - 如果系统不延长 SCL0n 信号低的保持时间：
t_{SU:DAT} ≥ 250 ns
 - 如果系统延长 SCL0n 信号低的保持时间：
发送如下数据位到 SDA0n 优先于 SCL0n 释放 (t_{Rmax} + t_{SU:DAT} = 1, 000 + 250 = 1, 250 ns: 正常模式 I²C 总线规范)。
 5. C_b: 一个总线的电容总和(单位: pF)

备注 n = 0 ~ 2

I²C Bus 模式

A/D 转换器

($T_A = -40 \sim +85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $3.0\text{ V} \leq AV_{REF0} \leq 3.6\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
分辨率					10	bit
总误差 [※]		$3.0 \leq AV_{REF0} \leq 3.6\text{ V}$			± 0.6	%FSR
转换时间	t_{CONV}		2.6		24	μs
零度误差					± 0.5	%FSR
满度误差					± 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	V_{IAN}		AV_{SS}		AV_{REF0}	V
参考电压	AV_{REF0}		3.0		3.6	V
AV _{REF0} 电流	AI _{REF0}	正常转换模式		3	6.5	mA
		高速转换模式		4	10	mA
		当A/D转换不用时			5	μA

注 不包括量化误差 ($\pm 0.05\%$ FSR).

注意事项 在 A/D 转换期间不要设置 (读/写) 复用端口; 否则转换分辨率可能降低。

备注 LSB: 最小有效位
FSR: 满度范围

D/A 转换器

($T_A = -40 \sim +85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $3.0\text{ V} \leq AV_{REF1} \leq 3.6\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
分辨率					8	bit
总误差 ^{注1}		$R = 2\text{ M}\Omega$			± 1.2	%FSR
设置时间		$C = 20\text{ pF}$			3	μs
输出电阻	R_o	输出数据 55H		3.5		$\text{k}\Omega$
参考电压	AV_{REF1}		3.0		3.6	V
AV_{REF1} 电流 ^{注2}	AI_{REF1}	D/A 转换操作		1	2.5	mA
		D/A 转换停止			5	μA

注 1. 不包括量化误差($\pm 0.5\text{ LSB}$).

2. D/A 转换器 1 通道的值

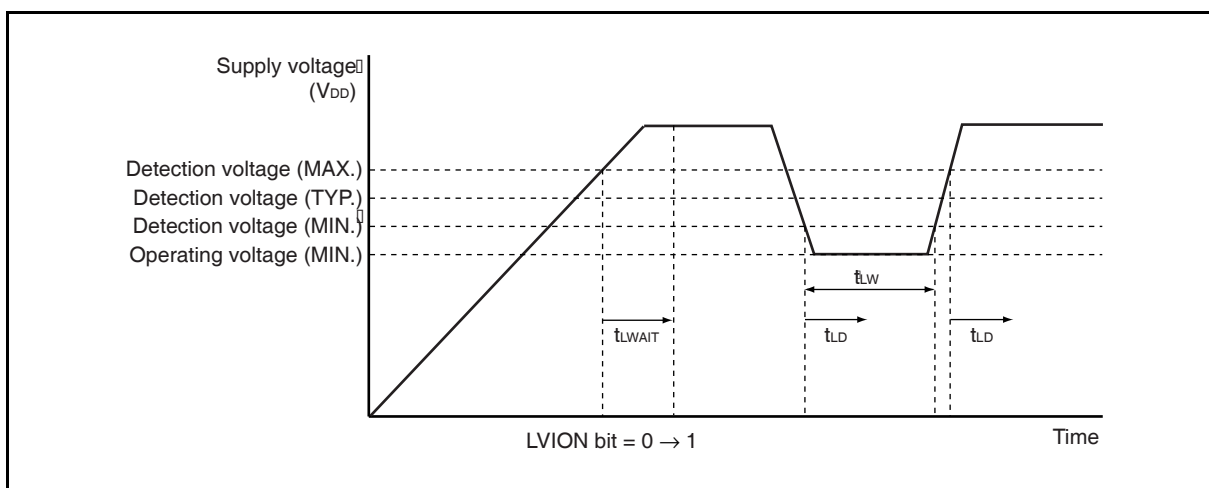
备注 R 为输出引脚加载阻抗, C 为输出引脚加载电容。

LVI 电 路 特 性

($T_A = -40 \sim +85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
检测电压	V_{LVIO}		2.85	3.0	3.15	V
响应时间 ^注	t_{LD}	在 V_{DD} 达到 V_{LVIO}/V_{LVH1} (MAX.) 后, 或在 V_{DD} 下降到 V_{LVIO}/V_{LVH1} (MIN.) 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}	在 V_{DD} 达到 2.85 V (MIN.) 后		0.1	0.2	ms

注 检测检测电压和输出一个中断或复位信号所必须的时间。

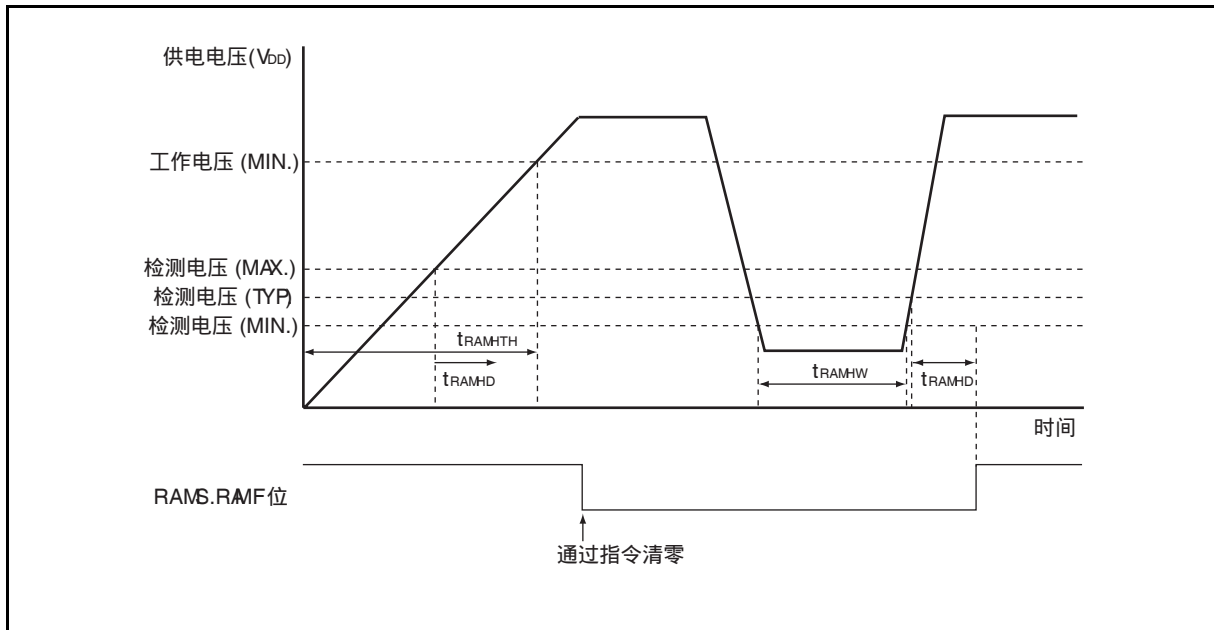


RAM 保存检测

($T_A = -40 \sim +85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
检测电压	V_{RAMH}		1.9	2.0	2.1	V
电源电压上升时间	t_{RAMHTh}	$V_{DD} = 0 \sim 2.85\text{ V}$	0.002			ms
响应时间 [※]	t_{RAMHD}	在 V_{DD} 达到 2.1 V 后		0.2	2.0	ms
最小脉冲宽度	t_{RAMHW}		0.2			ms

注 检测检测电压和设置 RAMS.RAMF 位所必需的时间。



Flash 存储器编程特性

($T_A = -40 \sim +85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

(1) 基本特性

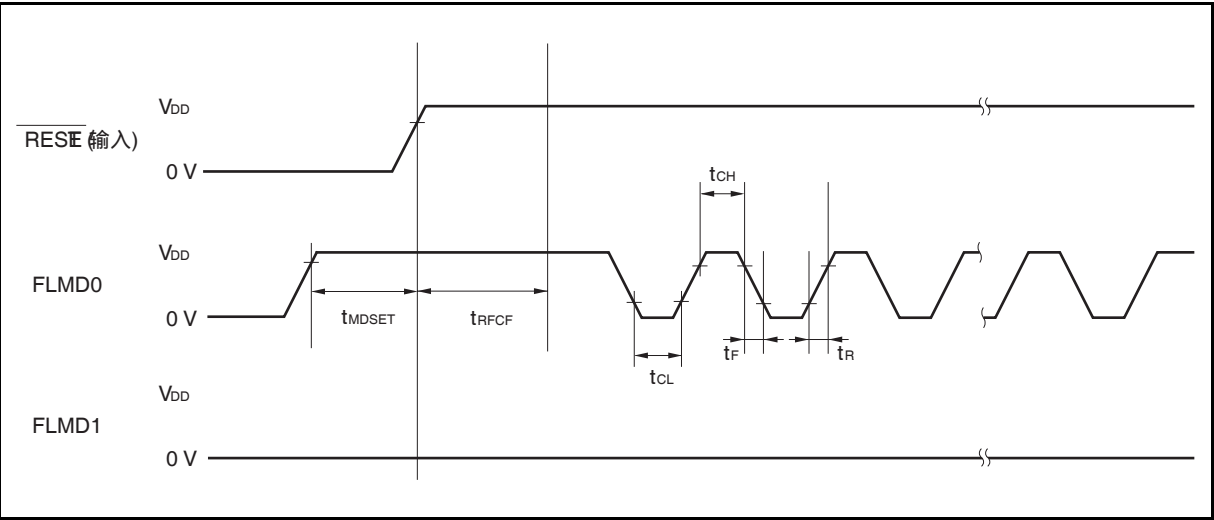
参数	符号	条件	MIN.	TYP.	MAX.	单位
工作频率	f _{CPU}		2.5		20	MHz
电源电压	V _{DD}		2.85		3.6	V
重写次数	C _{WRT}				100	次
编程温度	t _{PRG}		-40		+85	°C

(2) 串行写入操作特性

参数	符号	条件	MIN.	TYP.	MAX.	单位
FLMD0, FLMD1建立时间	t _{MDSET}		2		3000	ms
从 RESET↑ FLMD0计数开始时间	t _{RFCF}	f _x = 2.5 ~ 10 MHz	17855/f _x + α			s
FLMD0 计数器高电平宽度/ 低电平宽度	t _{CH} /t _{CL}		10	100		μs
FLMD0计数器上升时间/下降时间	t _R /t _F				50	ns

备注 α = 振荡稳定时间

Flash 写入模式建立时序



(3) 编程特性

参数	符号	条件	MIN.	TYP.	MAX.	单位
Block 擦除时间		f _{xx} = 20 MHz	注 1	304		ms
			注 2	1405		ms
			注 3	3057		ms
每 256 字节的写入时间		f _{xx} = 20 MHz		8.1		ms
Block 内部校验时间		f _{xx} = 20 MHz	注 1	20		ms
			注 2	141		ms
			注 3	322		ms
Block 空白检查时间		f _{xx} = 20 MHz	注 1	9.2		ms
			注 2	64		ms
			注 3	147		ms
Flash 存储器信息设置时间		f _{xx} = 20 MHz		1.0		ms

- 注
1. Block 大小 = 4 KB
 2. Block 大小 = 28 KB
 3. Block 大小 = 64 KB

注意事项 在出货后第一次对产品进行写操作时，“擦除 → 写”和“只写”作为一次重写。

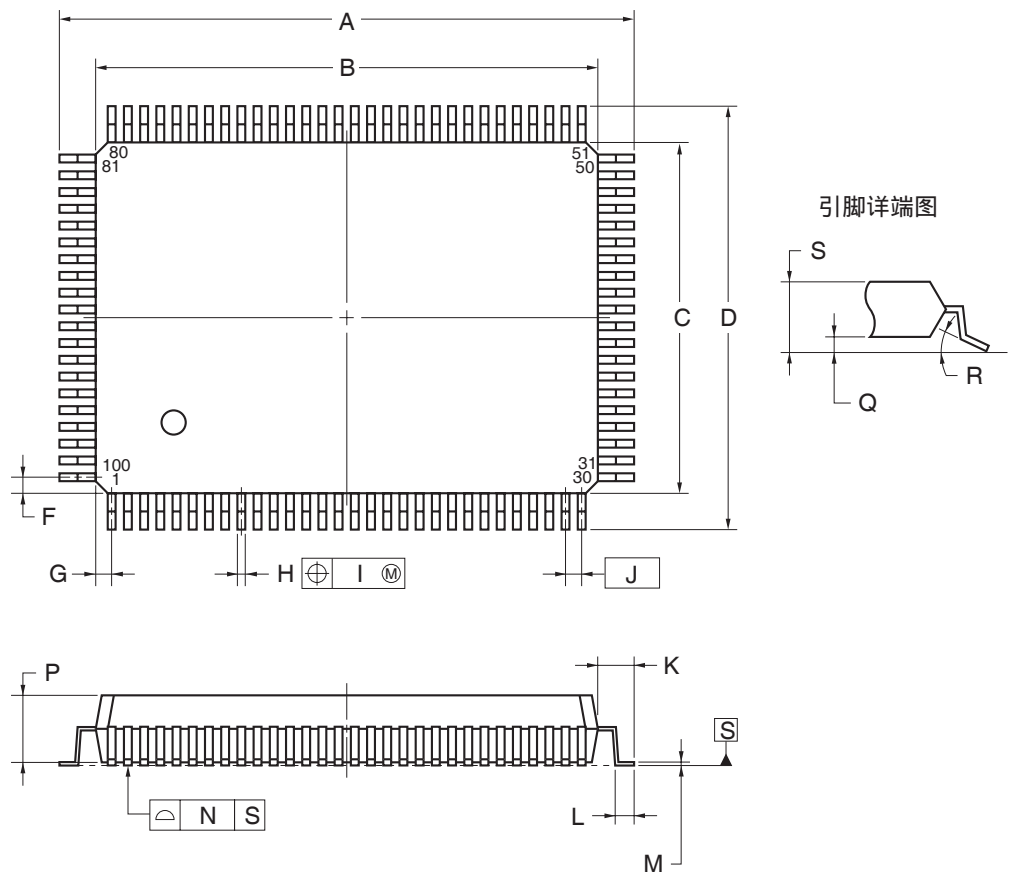
示例(P: 写入, E: 擦除)

出库产品 → P → E → P → E → P: 3 次重写

出库产品 → E → P → E → P → E → P: 3 次重写

第二十九章 封装图

100引脚塑封 QFP (14x20)

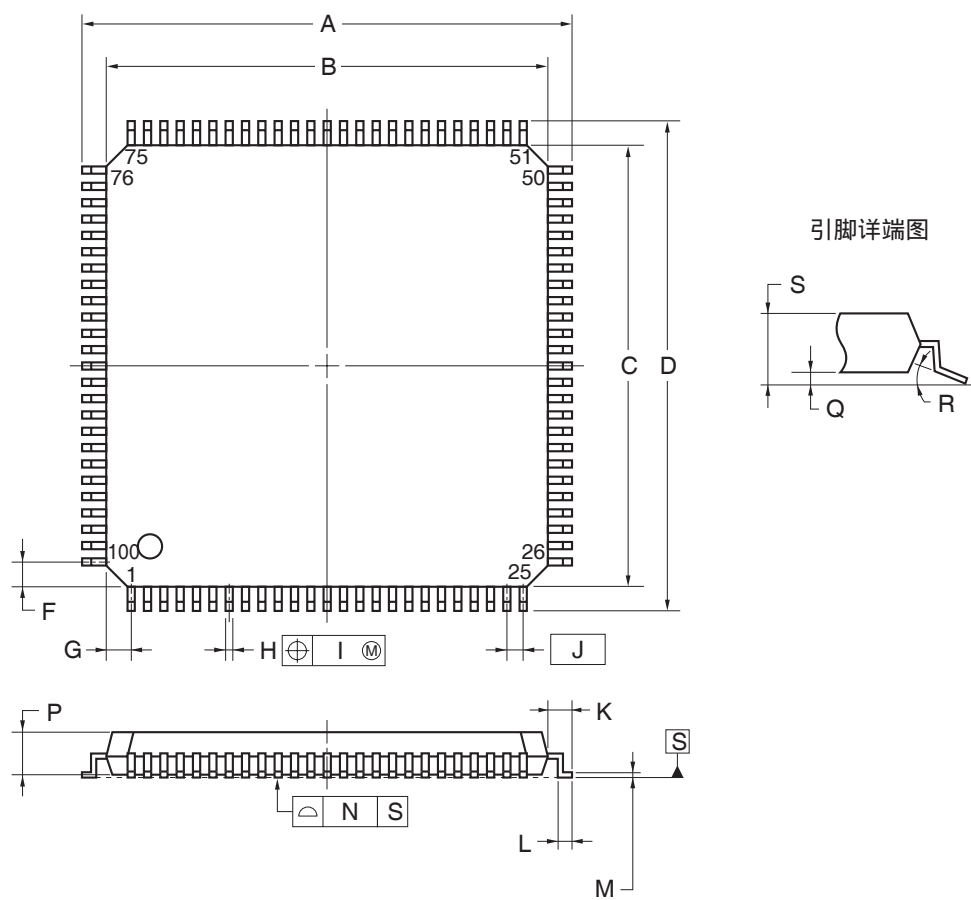


注
该结构在最大使用材料情况下，每条引脚的中心线
位于其实际位置的 0.13mm (TP.)内。

项目	尺寸
A	23.2±0.2
B	20.0±0.2
C	14.0±0.2
D	17.2±0.2
F	0.825
G	0.575
H	0.32 ^{+0.08} _{-0.07}
I	0.13
J	0.65 (TP.)
K	1.6±0.2
L	0.8±0.2
M	0.17 ^{+0.06} _{-0.05}
N	0.10
P	2.7±0.1
Q	0.125±0.075
R	3° ^{+7°} _{-3°}
S	3.0 MAX.

S100GF-65B72

100引脚塑封 LQFP(引脚阵) (14x14)



注
该结构在最大使用材料情况下，每条引脚的中心线位于其实际位置的0.08mm (TP.)内。

项目	尺寸
A	16.00±0.20
B	14.00±0.20
C	14.00±0.20
D	16.00±0.20
F	1.00
G	1.00
H	0.22 ^{+0.05} _{-0.04}
I	0.08
J	0.50 (TP.)
K	1.00±0.20
L	0.50±0.20
M	0.17 ^{+0.03} _{-0.07}
N	0.08
P	1.40±0.05
Q	0.10±0.05
R	3° ^{+7°} _{-3°}
S	1.60 MAX.

S100GC-50-8EU, 82

第三十章 推荐焊接条件

V850ES/JG2 应在如下推荐条件下焊接和装配。

技术信息，参见如下网站。

半导体设备装配手册(<http://www.necel.com/pkg/en/mount/index.html>)

表 30-1 贴装焊接条件

- (1) μ PD70F3715GC-8EA-A: 100 引脚塑封 LQFP (fine pitch) (14 × 14)
 μ PD70F3716GC-8EA-A: 100 引脚塑封 LQFP (fine pitch) (14 × 14)
 μ PD70F3717GC-8EA-A: 100 引脚塑封 LQFP (fine pitch) (14 × 14)
 μ PD70F3718GC-8EA-A: 100 引脚塑封 LQFP (fine pitch) (14 × 14)
 μ PD70F3719GC-8EA-A: 100 引脚塑封 LQFP (fine pitch) (14 × 14)

焊接方法	焊接条件	推荐型号
红外线	最高温度: 260°C, 时间: 最多 60 s (220°C 或更高)。次数: 小于等于 3 次, 暴露限制: 7 天 [※] (之后在 125°C 预烘 20 ~ 72 小时)	IR60-207-3
局部加热	引脚温度: 最大 350°C, 时间: 最大 3 秒 (每引脚排)	—

注 干燥条件下打开后，在允许的存储时间内，存储在低于 25°C 和 65% RH 下。

注意事项 不要一起使用不同的焊接方式(除局部加热外)。

- 备注 1. 零件号结尾带 -A 的产品为无铅产品。
 2. 以上推荐以外的焊接方法和条件，请联系 NEC Electronics 的销售代表。

- (2) μ PD70F3715GF-JBT-A: 100 引脚塑封 QFP (fine pitch) (14 × 20)
 μ PD70F3716GF-JBT-A: 100 引脚塑封 QFP (fine pitch) (14 × 20)
 μ PD70F3717GF-JBT-A: 100 引脚塑封 QFP (fine pitch) (14 × 20)

焊接方法	焊接条件	推荐型号
红外线	最高温度: 260°C, 时间: 最多 60 s (220°C 或更高)。次数: 小于等于 3 次, 暴露限制: 7 天 [※] (之后在 125°C 预烘 20 ~ 72 小时)	IR60-207-3
波峰焊	详细信息，联系 NEC 电子销售代表	—
局部加热	引脚温度: 最大 350°C, 时间: 最大 3 秒 (每引脚排)	—

注 干燥下打开后，在允许的存储时间内，存储在低于 25°C 和 65% RH 下。

注意事项 不要一起使用不同的焊接方式(除局部加热外)。

- 备注 1. 零件号结尾带 -A 的产品为无铅产品。
 2. 以上推荐以外的焊接方法和条件，请联系 NEC Electronics 的销售代表。

附录 A 开发工具

在使用 V850ES/JG2 的系统开发中可应用如下开发工具。

图 A-1 所示为开发工具的组成。

- **支持 PC98-NX 系列**

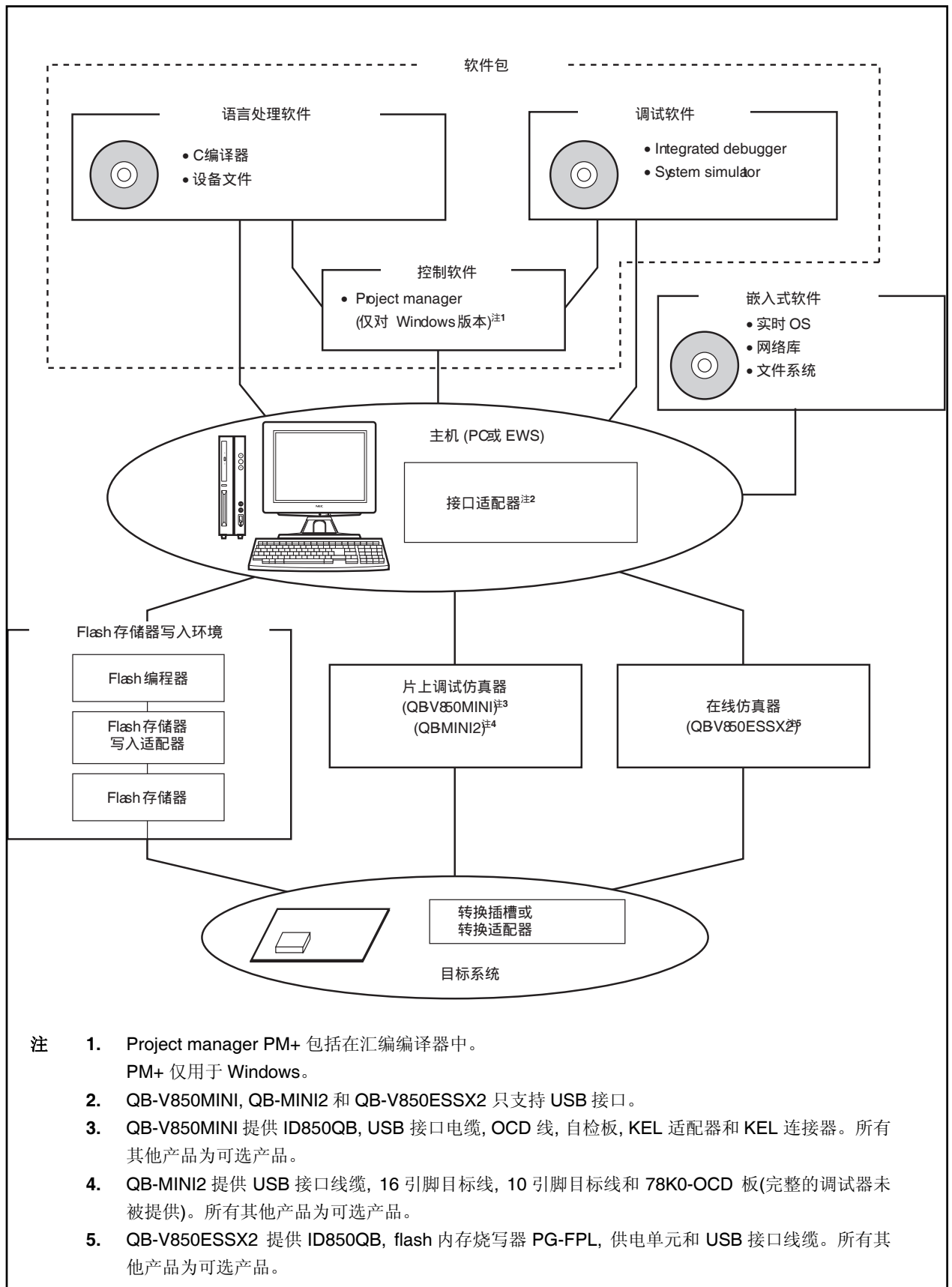
除非特别说明，IBM PC/AT™ 兼容机支持的产品与 PC98-NX 系列计算机是兼容的。当使用 PC98-NX 系列计算机时，参阅 IBM PC/AT 兼容机的使用说明。

- **Windows™**

除非特别说明，“Windows”指以下几种操作系统。

- Windows 98
- Windows 2000
- Windows Me
- Windows XP
- Windows NT™ Ver. 4.0

图 A-1 开发工具的组成



A.1 软件包

SP850 V850 微控制器软件包	软件包包括适用于 V850 微控制器的开发工具（软件）
	产品型号: μ SxxxxSP850

备注 产品型号中的 xxxx 随主机和使用的 OS 而变化。

μ SxxxxSP850

xxxx	主机	OS	存储介质
AB17	PC-9800 系列,	Windows (日文版)	CD-ROM
BB17	IBM PC/AT 兼容机	Windows (英文版)	

A.2 语言处理软件

CA850 C 编译器	此编译器将 C 语言程序转换为微控制器可执行的目标代码。此编译器由 project manager PM+ 启动。
	产品型号: μ SxxxxCA703000
DF703724 设备文件	该文件包含设备特有的信息。 该设备文件应结合工具(CA850, SM+ for V850ES/Jx2 和 ID850)一起使用。 相应的 OS 和主机随使用工具而变化。

备注 产品型号中的 xxxx 随主机和使用的 OS 而变化。

μ SxxxxCA703000

xxxx	主机	OS	存储介质
AB17	PC-9800 系列,	Windows (日文版)	CD-ROM
BB17	IBM PC/AT 兼容机	Windows (英文版)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4), Solaris™ (Rel. 2.5.1)	

A.3 控制软件

PM+ 项目管理器	这是一个控制软件，可使用户在 Windows 环境下有效地进行软件开发。用户程序开发中进行的任何操作，如启动编辑器、构造程序和启动调试器，都可以由 PM+ 执行。 <注意事项> PM+ is 包含在 C 编译器软件包 CA850。 仅在 Windows 下使用。
--------------	--

A.4 调试工具 (硬件)

A.4.1 使用 IECUBE QB-V850ESSX2

当连接 QB-V850ESSX2 到主机(PC-9821 系列, PC/AT 兼容机)时的系统配置如下所示。如果没有准备可选产品, 尽可能连接。

图 A-2 系统配置 (用于 QB-V850ESSX2 时) (1/2)

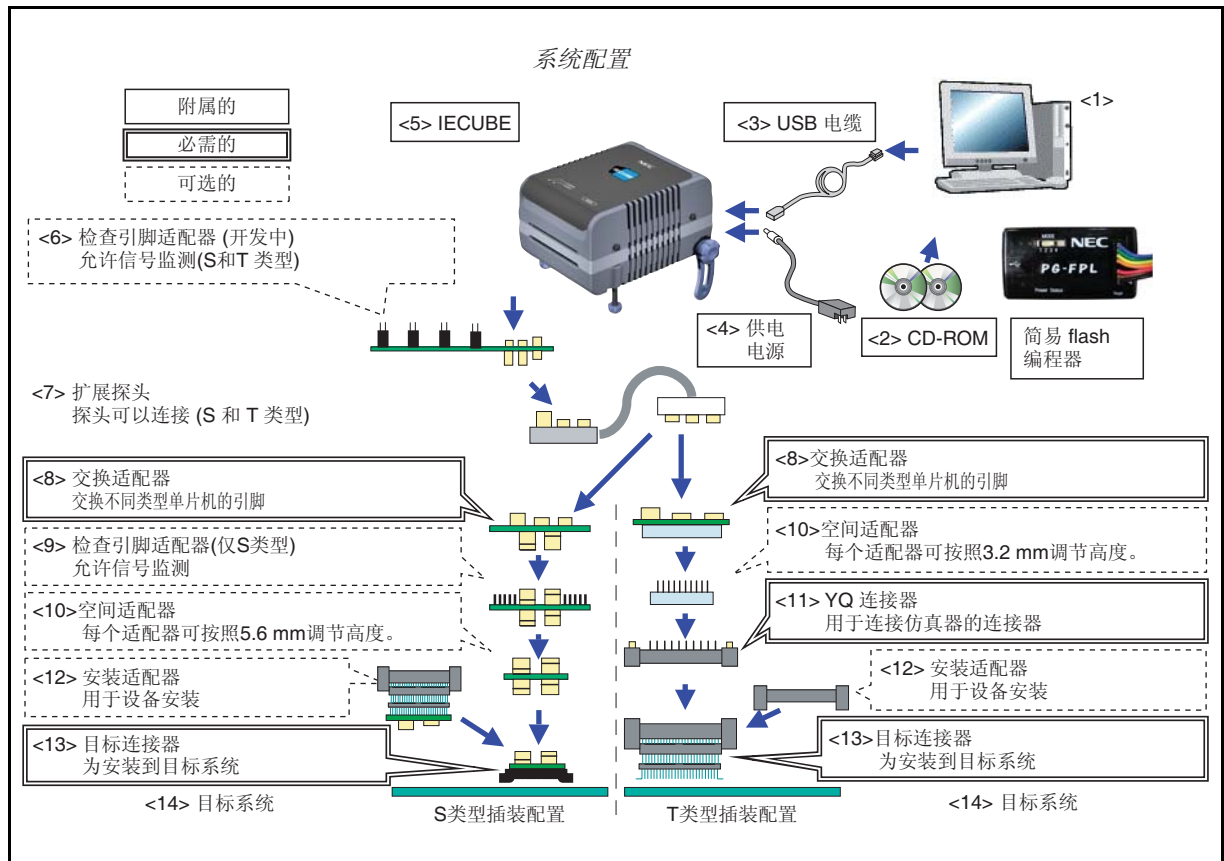


图 A-2 系统配置 (用于 QB-V850ESSX2 时) (2/2)

- <1> 主机(PC-9821 系列, 兼容 IBM-PC/AT)
- <2> 调试器, USB 驱动, 手册等(ID850QB 光盘, 附属光盘^{注意 1})
- <3> USB 接口电缆
- <4> AC 适配器
- <5> 在线仿真器(QB-V850ESSX2)
- <6> 检测引脚适配器(S 和 T 型) (QB-144-CA-01^{注意 2}) (可选)
- <7> 仿真探头(S 和 T 型) (QB-144-EP-01S) (可选)
- <8> 转换适配器^{注意 3} (S 型: QB-100GC-EA-01S (GC 封装), QB-100GF-EA-01S (GF 封装), T 型: QB-100GC-EA-01T (GC 封装), QB-100GF-EA-01T (GF 封装))
- <9> 检测引脚适配器^{注意 4} (仅 S 型) (QB-100-CA-01S) (可选)
- <10> 间隔适配器^{注意 4} (S 型: QB-100-SA-01S (GC/GF 封装), T 型: QB-100GC-YS-01T (GF 封装), QB-100GF-YS-01T (GF 封装) (可选)
- <11> YQ 连接器^{注意 3} (仅 T 型) (QB-100GC-YQ-01T) (GC 封装), QB-100GF-YQ-01T (GF 封装)
- <12> 装配适配器(S 型: QB-100GC-MA-01S (GC 封装), QB-100GF-MA-01S (GF 封装), T 型: QB-100GF-HQ-01T (GC 封装), QB-100GF-HQ-01T (GF 封装)) (可选)
- <13> 目标系统连接器^{注意 3} (S 型: QB-100GC-TC-01S (GC 封装), QB-100GF-TC-01S (GF 封装), T 型: QB-100GC-NQ-01T (GC 封装), QB-100GF-NQ-01T (GF 封装))
- <14> 目标系统

- 注
1. 从 NEC Electronics 网站获取设备文件。
<http://www.necel.com/micro/ods/eng/index.html>
 2. 开发中。
 3. 依照订购的数字提供器件。
 - 订购 QB-V850ESSX2-ZZZ 时
不提供交换适配器和目标连接器。
 - 订购 QB-V850ESSX2-S100GC 时
提供 QB-100GF-EA-01S 和 QB-100GF-TC-01S。
 - 订购 QB-V850ESSX2-S100GF 时
提供 QB-100GF-EA-01S 和 QB-100GF-TC-01S。
 - 订购 QB-V850ESSX2-T100GF 时
提供 QB-100GF-EA-01T, QB-100GF-YQ-01T, 和 QB-100GF-NQ-01T。
 - 订购 QB-V850ESSX2-T100GC 时
提供 QB-100GC-EA-01T, QB-100GC-YQ-01T, 和 QB-100GC-NQ-01T。
 4. 当同时使用<9>和<10>时, 不必关心<9>和<10>之间的顺序。

<5> QB-V850ESSX2 ^注 在线仿真器	当使用 V850ES/JG2 系列产品进行系统开发时，in-circuit 仿真器用来调试硬件和软件。它等同于整合调试工具 ID850QB。该仿真器应与电源和仿真探头结合使用。通过 USB 将仿真器连接到主机上。
<3> USB 接口电缆	电缆连接主机和 QB-V850ESSX2。
<4> AC 适配器	100 to 240 V can be supported by replacing the AC plug.
<8> QB-100GC-EA-01S QB-100GF-EA-01S QB-100GC-EA-01T QB-100GF-EA-01T 交换适配器	用于执行引脚的转换适配器。 <ul style="list-style-type: none"> • QB-100GC-EA-01S: 100-pin plastic LQFP (GC-8EA 类型) • QB-100GF-EA-01S: 100-pin plastic QFP (GF-JBT 类型) • QB-100GC-EA-01T: 100-pin plastic LQFP (GC-8EA 类型) • QB-100GF-EA-01T: 100-pin plastic QFP (GF-JBT 类型)
<9> QB-100-CA-01S (仅 S 型) 引脚检测适配器	在方波监控中使用的适配器需要用到示波境。 <ul style="list-style-type: none"> • QB-100-CA-01S: GC-8EA/GF-JBT 类型
<10> QB-100-SA-01S QB-100GC-YS-01T QB-100GF-YS-01T 空间适配器	用于调整最高点的适配器。 <ul style="list-style-type: none"> • QB-100GF-SA-01S: GC-8EA/GF-JBT 类型 • QB-100GC-YS-01T: 100-pin plastic LQFP (GC-8EA 类型) • QB-100GF-YS-01T: 100-pin plastic QFP (GF-JBT 类型)
<11> QB-100GC-YQ-01T QB-100GF-YQ-01T (仅 T 型) YQ 连接器	连接目标连接器和转换适配器的转换适配器 <ul style="list-style-type: none"> • QB-100GC-YQ-01T: 100-pin plastic LQFP (GC-8EA 类型) • QB-100GF-YQ-01T: 100-pin plastic QFP (GF-JBT 类型)
<12> QB-100GC-MA-01S QB-100GF-MA-01S QB-100GC-HQ-01T QB-100GF-HQ-01T 安装适配器	带有插槽的 V850ES/SJ2 的适配器。 <ul style="list-style-type: none"> • QB-100GC-MA-01S: 100-pin plastic LQFP (GC-8EA 类型) • QB-100GF-MA-01S: 100-pin plastic QFP (GF-JBT 类型) • QB-100GC-HQ-01T: 100-pin plastic LQFP (GC-8EA 类型) • QB-100GF-HQ-01T: 100-pin plastic QFP (GF-JBT 类型)
<13> QB-100GC-TC-01S QB-100GF-TC-01S QB-100GC-NQ-01T QB-100GF-NQ-01T 目标连接器	目标系统焊接的连接器。 <ul style="list-style-type: none"> • QB-100GC-TC-01S: 100-pin plastic LQFP (GC-8EA 类型) • QB-100GF-TC-01S: 100-pin plastic QFP (GF-JBT 类型) • QB-100GC-NQ-01T: 100-pin plastic LQFP (GC-8EA 类型) • QB-100GF-NQ-01T: 100-pin plastic QFP (GF-JBT 类型)

注 QB-V850ESSX2 提供电源单元，USB 接口线缆，和 flash 内存编程器 PG-FPL。同时也提供集成调试器 ID850QB 作为控制工具。

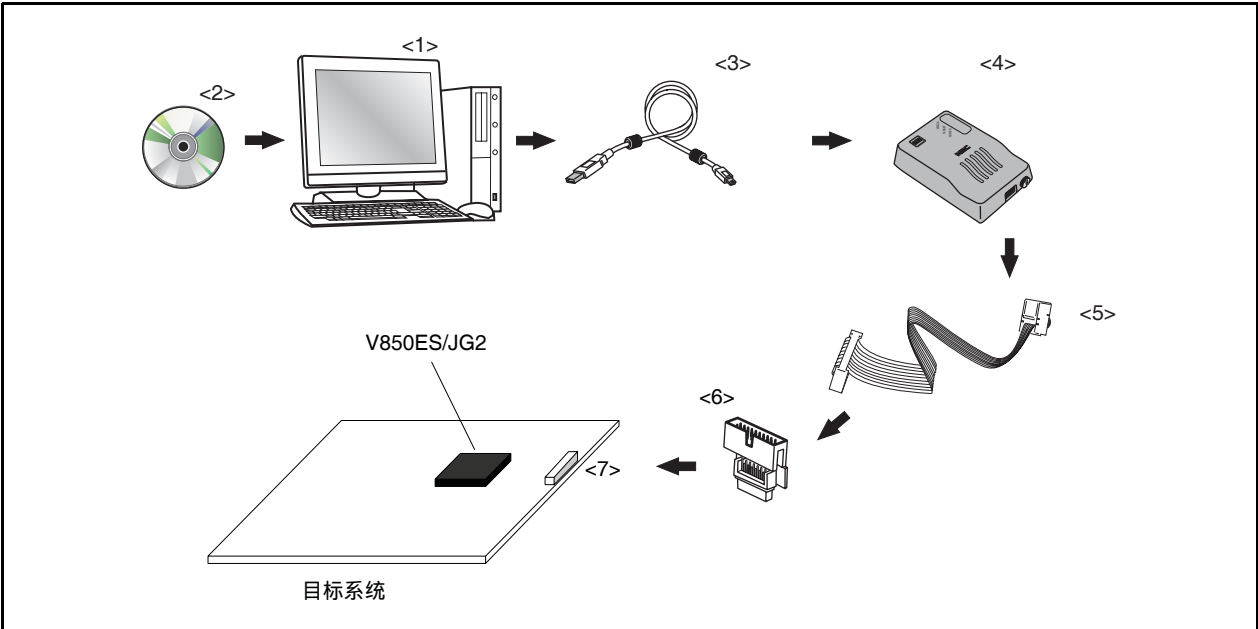
备注 方括号中的数字对应图 A-2 中的数字。

A.4.2 使用 MINICUBE QB-V850MINI

(1) 使用 MINICUBE 片上仿真

当连接 MINICUBE 到主机(PC-9821 系列, PC/AT 兼容机)时的系统配置如下所示。

图 A-3 片上仿真系统构成



<1> 主机	带 USB 口的 PC
<2> CD-ROM ^{注意 1}	CD-ROM 内容有集成调试器 ID850QB, N-Wire 检测器, 设备驱动和文档。和 MINICUBE 一起提供。
<3> USB 接口线	USB 线连接主机和 MINICUBE。和 MINICUBE 一起提供。线长约 2 m.
<4> MINICUBE 片上调试器	本调试器用在使用 V850ES/JG2 开发应用系统时硬件和软件的调试。支持集成调试器 ID850QB.
<5> OCD 线	连接 MINICUBE 和目标系统。 和 MINICUBE 一起提供。线长约 20 cm.
<6> 连接器转换板 KEL 适配器	本转换板和 MINICUBE 一起提供。
<7> MINICUBE 连接器 KEL 连接器 ^{注意 2}	8830E-026-170S (和 MINICUBE 一起提供) 8830E-026-170L (单独销售)

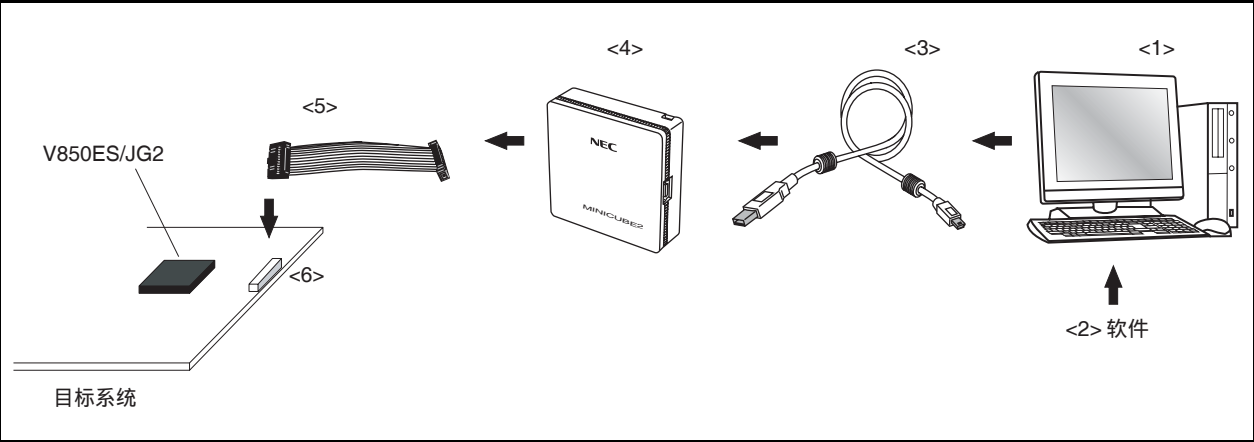
注 1. 从 NEC Electronics 网站下载设备文件。
<http://www.necel.com/micro/ods/eng/index.html>
2. KEL 公司产品

备注 方括号中的数字对应图 A-3 中的数字。

A.4.3 使用 MINICUBE2 QB-MINI2

当连接 MINICUBE2 到主机(PC-9821 系列, PC/AT 兼容机)时的系统配置如下所示。

图 A-4 片上仿真系统构成



<1> 主机	带 USB 口的 PC
<2> 软件	集成调试器 ID850QB, 设备文件等. 从 NEC 电子网站下载设备文件. http://www.necel.com/micro/ods/eng/
<3> USB 接口线	USB 线连接主机和 MINICUBE. 和 MINICUBE 一起提供. 线长约 2 m.
<4> MINICUBE2 片上调试器	本调试器用在使用 V850ES/JG2 开发应用系统时硬件和软件的调试. 支持集成调试器 ID850QB.
<5> 16 引脚目标线	连接 MINICUBE2 和目标系统. 和 MINICUBE 一起提供. 线长约 15 cm.
<6> 目标连接器 (单独销售)	使用 2.54 mm 间距的 16 引脚通用连接器

备注 方括号中的数字对应图 A-4 中的数字。

A.5 调试工具 (软件)

SM+ for V850ES/Jx2 (开发中) 系统模拟器	该系统模拟器支持 V850 微控制器。 SM+ for V850ES/Jx2 是基于 Windows 的软件。当模拟目标系统操作时用于在主机上调试 C 源程序或汇编程序。 通过使用 SM+ for V850ES/Jx2,允许执行逻辑测试应用和基于硬件开发的独立测试，因此提供了很高的开发效率和软件质量。
	产品型号: μSxxxxSM703724-B
ID850QB 集成调试器	此调试器支持 V850 微控制器在线仿真器。ID850QB 是基于 Windows 的软件。它拥有改良的 C 语言编译调试功能，并且能够通过结合源程序的集成窗口功能，分解显示，和存储器显示来显示追踪源程序的结果。 该功能需要使用器件文件。
	产品型号: μSxxxx ID703000-QB (ID850QB)

备注 产品型号中的 xxxx使用的 OS 而变化。

μSxxxxID703000-QB

xxxx	主机	OS	存储介质
AB17	PC-9800 系列, IBM PC/AT 兼容机	Windows (日文版本)	CD-ROM
BB17		Windows (英文版本)	

A.6 嵌入式软件

RX850, RX850 Pro 实时 OS	RX850 和 RX850 Pro 是依照 μ ITRON 3.0 规范的实时操作系统。 提供一个为了产生多个信息表格的工具 (配置器)。 RX850 Pro 比 RX850 具有更多的功能。
	产品型号: μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$ (RX850) μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$ (RX850 Pro)
Applilet [®] 注意	本驱动配置能自动生成 V850ES/JG2 用的例程。
RX-FS850 (文件系统)	这是 FAT 文件系统功能 此文件系统支持 CD-ROM 文件系统功能。 此文件系统用在实时 OS RX850 Pro。

注意 如何获得 Applilet, 请联系 NEC 电子销售代表。

注意事项 为了购买 RX850 或 RX850 Pro, 首先需要填写购买申请表并且签署许可协议。

备注 产品型号中的 xxxx 和 $\Delta\Delta\Delta\Delta$ 随主机和使用的 OS 而变化。

μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$

μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$

$\Delta\Delta\Delta\Delta$	产品概述	用于大规模生产的上限
001	评价目标	不用于批量生产
100K	大规模生产目标	10 万单元
001M		100 万单元
010M		1000 万单元
S01	源程序	量产的目标源程序

xxxx	主机	OS	存储介质
AB17	PC-9800 系列,	Windows (日文版)	CD-ROM
BB17	IBM PC/AT 兼容机	Windows (英文版)	
3K17	SPARCstation	Solaris (Rel. 2.5.1)	

A.7 Flash 存储器写入工具

Flashpro IV (产品型号: PG-FP4) Flash 编程器	Flash 编程器专用于有片上 flash 存储器的微控制器。
QB-MINI2 (MINICUBE2)	带写入功能的片上调试器。
FA-100GC-8UE-A FA-100GF-3BA-A Flash 存储器写入适配器	Flash 存储器写入适配器用于连接 Flashpro IV 等 (未连线). <ul style="list-style-type: none"> FA-100GC-8EU-A: 100-pin plastic LQFP (GC-8EA 类型) FA-100GF-3BA-A: 100-pin plastic QFP (GF-JBT 类型)
FA-70F3719GC-8EA-MX FA-70F3717GF-JBT-MX Flash 存储器写入适配器	Flash 存储器写入适配器用于连接 Flashpro IV 等 (已经连线). <ul style="list-style-type: none"> FA-70F3719GC-8EA-MX: 100-pin plastic LQFP (GC-8EA 类型) FA-70F3717GF-JBT-MX: 100-pin plastic QFP (GF-JBT 类型)

备注 FA-100GC-8EU-A, FA-100GF-3BA-A, FA-70F3719GC-8EA-MX, FA-70F3717GF-JBT-M 是 Naito Densai Machida Mfg. Co., Ltd. 的产品。
TEL: +81-42-750-4172

附录 B 寄存器索引

(1/10)

符号	名称	单元	页码
ADA0CR0	A/D 转换结果寄存器 0	ADC	439
ADA0CR0H	A/D 转换结果寄存器 0H	ADC	439
ADA0CR1	A/D 转换结果寄存器 1	ADC	439
ADA0CR1H	A/D 转换结果寄存器 1H	ADC	439
ADA0CR2	A/D 转换结果寄存器 2	ADC	439
ADA0CR2H	A/D 转换结果寄存器 2H	ADC	439
ADA0CR3	A/D 转换结果寄存器 3	ADC	439
ADA0CR3H	A/D 转换结果寄存器 3H	ADC	439
ADA0CR4	A/D 转换结果寄存器 4	ADC	439
ADA0CR4H	A/D 转换结果寄存器 4H	ADC	439
ADA0CR5	A/D 转换结果寄存器 r 5	ADC	439
ADA0CR5H	A/D 转换结果寄存器 5H	ADC	439
ADA0CR6	A/D 转换结果寄存器 6	ADC	439
ADA0CR6H	A/D 转换结果寄存器 6H	ADC	439
ADA0CR7	A/D 转换结果寄存器 7	ADC	439
ADA0CR7H	A/D 转换结果寄存器 7H	ADC	439
ADA0CR8	A/D 转换结果寄存器 8	ADC	439
ADA0CR8H	A/D 转换结果寄存器 8H	ADC	439
ADA0CR9	A/D 转换结果寄存器 9	ADC	439
ADA0CR9H	A/D 转换结果寄存器 9H	ADC	439
ADA0CR10	A/D 转换结果寄存器 10	ADC	439
ADA0CR10H	A/D 转换结果寄存器 10H	ADC	439
ADA0CR11	A/D 转换结果寄存器 11	ADC	439
ADA0CR11H	A/D 转换结果寄存器 11H	ADC	439
ADA0M0	A/D 转换结果寄存器 0	ADC	432
ADA0M1	A/D 转换结果寄存器 1	ADC	434
ADA0M2	A/D 转换结果寄存器 2	ADC	437
ADA0PFM	电压不足比较模式寄存器	ADC	441
ADA0PFT	电压不足比较阈值寄存器	ADC	442
ADA0S	A/D 转换通道指定寄存器	ADC	438
ADIC	中断控制寄存器	INTC	673
AWC	地址等待控制寄存器	BCU	187
BCC	总线周期控制寄存器	BCU	188
BSC	总线宽度配置寄存器	BCU	176
CB0CTL0	CSIB0 控制寄存器 0	CSIB	510
CB0CTL1	CSIB0 控制寄存器 1	CSIB	513
CB0CTL2	CSIB0 控制寄存器 2	CSIB	514
CB0RIC	中断控制寄存器	INTC	673
CB0RX	CSIB0 接收数据寄存器	CSIB	509
CB0RXL	CSIB0 接收数据寄存器 L	CSIB	509
CB0STR	CSIB0 状态寄存器	CSIB	516
CB0TIC	中断控制寄存器	INTC	673

符号	名称	单元	页码
CB0TX	CSIB0 发送数据寄存器	CSI	509
CB0TXL	CSIB0 发送数据寄存器 L	CSI	509
CB1CTL0	CSIB1 控制寄存器 0	CSI	510
CB1CTL1	CSIB1 控制寄存器 1	CSI	513
CB1CTL2	CSIB1 控制寄存器 2	CSI	514
CB1RIC	中断控制寄存器	INTC	673
CB1RX	CSIB1 接收数据寄存器	CSI	509
CB1RXL	CSIB1 接收数据寄存器 L	CSI	509
CB1STR	CSIB1 状态寄存器	CSI	516
CB1TIC	中断控制寄存器	INTC	673
CB1TX	CSIB1 发送数据寄存器	CSI	509
CB1TXL	CSIB1 发送数据寄存器 L	CSI	509
CB2CTL0	CSIB2 控制寄存器 0	CSI	510
CB2CTL1	CSIB2 控制寄存器 1	CSI	513
CB2CTL2	CSIB2 控制寄存器 2	CSI	514
CB2RIC	中断控制寄存器	INTC	673
CB2RX	CSIB2 接收数据寄存器	CSI	509
CB2RXL	CSIB2 接收数据寄存器 L	CSI	509
CB2STR	CSIB2 状态寄存器	CSI	516
CB2TIC	中断控制寄存器	INTC	673
CB2TX	CSIB2 发送数据寄存器	CSI	509
CB2TXL	CSIB2 发送数据寄存器 L	CSI	509
CB3CTL0	CSIB3 控制寄存器 0	CSI	510
CB3CTL1	CSIB3 控制寄存器 1	CSI	513
CB3CTL2	CSIB3 控制寄存器 2	CSI	514
CB3RIC	中断控制寄存器	INTC	673
CB3RX	CSIB3 接收数据寄存器	CSI	509
CB3RXL	CSIB3 接收数据寄存器 L	CSI	509
CB3STR	CSIB3 状态寄存器	CSI	516
CB3TIC	中断控制寄存器	INTC	673
CB3TX	CSIB3 发送数据寄存器	CSI	509
CB3TXL	CSIB3 发送数据寄存器 L	CSI	509
CB4CTL0	CSIB4 控制寄存器 0	CSI	510
CB4CTL1	CSIB4 控制寄存器 1	CSI	513
CB4CTL2	CSIB4 控制寄存器 2	CSI	514
CB4RIC	中断控制寄存器	INTC	673
CB4RX	CSIB4 接收数据寄存器	CSI	509
CB4RXL	CSIB4 接收数据寄存器 L	CSI	509
CB4STR	CSIB4 状态寄存器	CSI	516
CB4TIC	中断控制寄存器	INTC	673
CB4TX	CSIB4 发送数据寄存器	CSI	509
CB4TXL	CSIB4 发送数据寄存器 L	CSI	509
CCLS	CPU 操作时钟状态寄存器	CG	205
CKC	时钟控制寄存器	CG	208
CLM	时钟监视器模式寄存器	CLM	727

符号	名称	单元	页码
CTBP	CALLT 基址指针	CPU	54
CTPC	CALLT 执行状态保存寄存器	CPU	53
CTPSW	CALLT 执行状态保存寄存器	CPU	53
DA0CS0	D/A 转换值设置寄存器 0	DAC	466
DA0CS1	D/A 转换值设置寄存器 1	DAC	466
DA0M	D/A 转换模式寄存器	DAC	465
DADC0	DMA 地址控制寄存器 0	DMAC	639
DADC1	DMA 地址控制寄存器 1	DMAC	639
DADC2	DMA 地址控制寄存器 2	DMAC	639
DADC3	DMA 地址控制寄存器 3	DMAC	639
DBC0	DMA 发送计数寄存器 0	DMAC	638
DBC1	DMA 发送计数寄存器 1	DMAC	638
DBC2	DMA 发送计数寄存器 2	DMAC	638
DBC3	DMA 发送计数寄存器 3	DMAC	638
DCHC0	DMA 通道控制寄存器 0	DMAC	54
DCHC1	DMA 通道控制寄存器 1	DMAC	54
DCHC2	DMA 通道控制寄存器 2	DMAC	640
DCHC3	DMA 通道控制寄存器 3	DMAC	640
DDA0H	DMA 目的地址寄存器 0H	DMAC	640
DDA0L	DMA 目的地址寄存器 0L	DMAC	640
DDA1H	DMA 目的地址寄存器 1H	DMAC	637
DDA1L	DMA 目的地址寄存器 1L	DMAC	637
DDA2H	DMA 目的地址寄存器 2H	DMAC	637
DDA2L	DMA 目的地址寄存器 2L	DMAC	637
DDA3H	DMA 目的地址寄存器 3H	DMAC	637
DDA3L	DMA 目的地址寄存器 3L	DMAC	637
DMAIC0	中断控制寄存器	INTC	637
DMAIC1	中断控制寄存器	INTC	637
DMAIC2	中断控制寄存器 r	INTC	673
DMAIC3	中断控制寄存器	INTC	673
DSA0H	DMA 源地址寄存器 0H	DMAC	673
DSA0L	DMA 源地址寄存器 0L	DMAC	673
DSA1H	DMA 源地址寄存器 1H	DMAC	636
DSA1L	DMA 源地址寄存器 1L	DMAC	636
DSA2H	DMA 源地址寄存器 2H	DMAC	636
DSA2L	DMA 源地址寄存器 2L	DMAC	636
DSA3H	DMA 源地址寄存器 3H	DMAC	636
DSA3L	DMA 源地址寄存器 3L	DMAC	636
DTFR0	DMA 触发源寄存器 0	DMAC	636
DTFR1	DMA 触发源寄存器 1	DMAC	636
DTFR2	DMA 触发源寄存器 2	DMAC	641
DTFR3	DMA 触发源寄存器 3	DMAC	641
DWC0	数据等待控制寄存器 0	BCU	641
ECR	中断源寄存器	CPU	51

符号	名称	单元	页码
EIPC	中断状态保存寄存器	CPU	50
EIPSW	中断状态保存寄存器	CPU	50
EXIMC	外部总线接口模式控制寄存器	BCU	175
FEPC	NMI 状态保存寄存器	CPU	51
FEPSW	NMI 状态保存寄存器	CPU	51
IIC0	IIC 移位寄存器 0	I ² C	577
IIC1	IIC 移位寄存器 1	I ² C	577
IIC2	IIC 移位寄存器 2	I ² C	577
IICC0	IIC 控制寄存器 0	I ² C	563
IICC1	IIC 控制寄存器 1	I ² C	563
IICC2	IIC 控制寄存器 2	I ² C	563
IICCL0	IIC 时钟选择寄存器 0	I ² C	573
IICCL1	IIC 时钟选择寄存器 1	I ² C	573
IICCL2	IIC 时钟选择寄存器 2	I ² C	573
IICF0	IIC 标志寄存器 0	I ² C	571
IICF1	IIC 标志寄存器 1	I ² C	571
IICF2	IIC 标志寄存器 2	I ² C	571
IICIC0	中断控制寄存器	INTC	673
IICIC1	中断控制寄存器	INTC	673
IICIC2	中断控制寄存器	INTC	673
IICS0	IIC 状态寄存器 0	I ² C	568
IICS1	IIC 状态寄存器 1	I ² C	568
IICS2	IIC 状态寄存器 2	I ² C	568
IICX0	IIC 功能扩展寄存器 0	I ² C	574
IICX1	IIC 功能扩展寄存器 1	I ² C	574
IICX2	IIC 功能扩展寄存器 2	I ² C	574
IMR0	中断掩码寄存器 0	INTC	673
IMR0H	中断掩码寄存器 0H	INTC	673
IMR0L	中断掩码寄存器 0L	INTC	673
IMR1	中断掩码寄存器 1	INTC	673
IMR1H	中断掩码寄存器 1H	INTC	673
IMR1L	中断掩码寄存器 1L	INTC	673
IMR2	中断掩码寄存器 2	INTC	673
IMR2H	中断掩码寄存器 2H	INTC	673
IMR2L	中断掩码寄存器 2L	INTC	673
IMR3	中断掩码寄存器 3	INTC	673
IMR3H	中断掩码寄存器 3H	INTC	673
IMR3L	中断掩码寄存器 3L	INTC	673
INTF0	外部中断下降沿指定寄存器 0	INTC	687
INTF3	外部中断下降沿指定寄存器 3	INTC	688
INTF9H	外部中断下降沿指定寄存器 9H	INTC	689
INTR0	外部中断上升沿指定寄存器 0	INTC	687
INTR3	外部中断上升沿指定寄存器 3	INTC	688
INTR9H	外部中断上升沿指定寄存器 9H	INTC	689
ISPR	当前优先级寄存器	INTC	677

符号	名称	单元	页码
KRIC	中断控制寄存器	INTC	673
KRM	按键返回模式寄存器	KR	694
LOCKR	锁定寄存器	CG	209
LVIIC	中断控制寄存器	INTC	673
LVIM	低电压检测寄存器	LVI	732
LVIS	低电压检测等级选择寄存器	LVI	733
NFC	噪声消除控制寄存器	INTC	690
OCDM	片上调试模式寄存器	调试	769
OCKS0	IIC 分频时钟选择寄存器 0	I ² C	577
OCKS1	IIC 分频时钟选择寄存器 1	I ² C	577
OCKS2	IEBus 时钟选择寄存器	IEBus	699
OSTS	振荡稳定时间选择寄存器	待机	94
P0	端口 0 寄存器	端口	97
P1	端口 1 寄存器	端口	99
P3	端口 3 寄存器	端口	99
P3H	端口 3 寄存器 H	端口	99
P3L	端口 3 寄存器 L	端口	104
P4	端口 4 寄存器	端口	106
P5	端口 5 寄存器	端口	111
P7H	端口 7 寄存器 H	端口	111
P7L	端口 7 寄存器 L	端口	113
P9	端口 9 寄存器	端口	113
P9H	端口 9 寄存器 H	端口	113
P9L	端口 9 寄存器 L	端口	48
PC	程序指针	CPU	201
PCC	处理器时钟控制寄存器	CG	120
PCM	端口 CM 寄存器	端口	122
PCT	端口 CT 寄存器	端口	124
PDH	端口 DH 寄存器	端口	127
PDL	端口 DL 寄存器	端口	127
PDLH	端口 DL 寄存器 H	端口	127
PDLL	端口 DL 寄存器 L	端口	737
PEMU1	外围扩展寄存器 1	CPU	96
PF0	端口 0 功能控制寄存器	端口	103
PF3	端口 3 功能控制寄存器	端口	103
PF3H	端口 3 功能控制寄存器 H	端口	103
PF3L	端口 3 功能控制寄存器 L	端口	105
PF4	端口 4 功能控制寄存器	端口	109
PF5	端口 5 功能控制寄存器	端口	119
PF9	端口 9 功能控制寄存器	端口	119
PF9H	端口 9 功能控制寄存器 H	端口	119
PF9L	端口 9 功能控制寄存器 L	端口	96
PFC0	端口 0 功能控制寄存器	端口	101
PFC3	端口 3 功能控制寄存器	端口	101
PFC3H	端口 3 功能控制寄存器 H	端口	677

符号	名称	单元	页码
PFC3L	端口 3 功能控制寄存器 L	端口	101
PFC4	端口 4 功能控制寄存器	端口	105
PFC5	端口 5 功能控制寄存器	端口	108
PFC9	端口 9 功能控制寄存器	端口	116
PFC9H	端口 9 功能控制寄存器 H	端口	116
PFC9L	端口 9 功能控制寄存器 L	端口	116
PFCE3L	端口 3 功能控制扩展寄存器 L	端口	101
PFCE5	端口 5 功能控制扩展寄存器	端口	108
PFCE9	端口 9 功能控制扩展寄存器	端口	116
PFCE9H	端口 9 功能控制扩展寄存器 H	端口	116
PFCE9L	端口 9 功能控制扩展寄存器 L	端口	116
PIC0	中断控制寄存器	INTC	673
PIC1	中断控制寄存器	INTC	673
PIC2	中断控制寄存器	INTC	673
PIC3	中断控制寄存器	INTC	673
PIC4	中断控制寄存器	INTC	673
PIC5	中断控制寄存器	INTC	673
PIC6	中断控制寄存器	INTC	673
PIC7	中断控制寄存器	INTC	673
PLLCTL	PLL 控制寄存器	CG	207
PLLS	PLL 锁止时间指定寄存器	CG	210
PM0	端口 0 模式寄存器	端口	95
PM1	端口 1 模式寄存器	端口	97
PM3	端口 3 模式寄存器	端口	99
PM3H	端口 3 模式寄存器 H	端口	99
PM3L	端口 3 模式寄存器 L	端口	99
PM4	端口 4 模式寄存器	端口	104
PM5	端口 5 模式寄存器	端口	107
PM7H	端口 7 模式寄存器 H	端口	111
PM7L	端口 7 模式寄存器 L	端口	111
PM9	端口 9 模式寄存器	端口	113
PM9H	端口 9 模式寄存器 H	端口	113
PM9L	端口 9 模式寄存器 L	端口	113
PMC0	端口 0 模式控制寄存器	端口	95
PMC3	端口 3 模式控制寄存器	端口	100
PMC3H	端口 3 模式控制寄存器 H	端口	100
PMC3L	端口 3 模式控制寄存器 L	端口	100
PMC4	端口 4 模式控制寄存器	端口	105
PMC5	端口 5 模式控制寄存器	端口	107
PMC9	端口 9 模式控制寄存器	端口	114
PMC9H	端口 9 模式控制寄存器 H	端口	114
PMC9L	端口 9 模式控制寄存器 L	端口	114
PMCCM	端口 CM 模式控制寄存器	端口	121
PMCCT	端口 CT 模式控制寄存器	端口	123
PMCDH	端口 DH 模式控制寄存器	端口	125

符号	名称	单元	页码
PMCDL	端口 DL 模式控制寄存器	端口	128
PMCDLH	端口 DL 模式控制寄存器 H	端口	128
PMCDLL	端口 DL 模式控制寄存器 L	端口	128
PMCM	端口 CM 模式寄存器	端口	120
PMCT	端口 CT 模式寄存器 r	端口	122
PMDH	端口 DH 模式寄存器	端口	124
PMDL	端口 DL 模式寄存器	端口	127
PMDLH	端口 DL 模式寄存器 H	端口	125
PMDLL	端口 DL 模式寄存器 L	端口	125
PRCMD	命令寄存器	CPU	82
PRSCM0	预分频比较寄存器	WT	410
PRSCM1	BRG1 预分频比较寄存器	BRG	553
PRSCM2	BRG2 预分频比较寄存器	BRG	553
PRSCM3	BRG3 预分频比较寄存器	BRG	553
PRSM0	预分频模式寄存器	WT	409
PRSM1	BRG1 预分频模式寄存器	BRG	552
PRSM2	BRG2 预分频模式寄存器	BRG	552
PRSM3	BRG3 预分频模式寄存器	BRG	552
PSC	Power save control register	CG	697
PSMR	Power save mode register	CG	698
PSW	程序状态字	CPU	52
r0-r31	通用寄存器	CPU	48
RAMS	内部 RAM 数据状态寄存器	CG	733
RCM	内部振荡模式寄存器	CG	205
RESF	复位源标志寄存器	LVI	716
RTBH0	实时输出缓冲寄存器 0H	RTP	423
RTBL0	实时输出缓冲寄存器 0L	RTP	423
RTPC0	实时输出端口模式寄存器 0	RTP	425
RTPM0	实时输出端口模式寄存器 0	RTP	424
SELCNT0	选择器操作控制寄存器 0	Timer	297
SVA0	IIC 从地址寄存器 0	I ² C	578
SVA1	IIC 从地址寄存器 1	I ² C	578
SVA2	IIC 从地址寄存器 2	I ² C	578
SYS	系统状态寄存器 0	CPU	83
TM0CMP0	TMM0 比较寄存器 0	Timer	399
TM0CTL0	TMM0 控制寄存器 0	Timer	400
TM0EQIC0	中断控制寄存器	INTC	673
TP0CCIC0	中断控制寄存器	INTC	673
TP0CCIC1	中断控制寄存器	INTC	673
TP0CCR0	TMP0 捕捉/比较寄存器 0	Timer	221
TP0CCR1	TMP0 捕捉/比较寄存器 1	Timer	223
TP0CNT	TMP0 计数读缓冲寄存器	Timer	225
TP0CTL0	TMP0 控制寄存器 0	Timer	215
TP0CTL1	TMP0 控制寄存器 1	Timer	215
TP0IOC0	TMP0 I/O 控制寄存器 0	Timer	217

符号	名称	单元	页码
TP0IOC1	TMP0 I/O 控制寄存器 1	Timer	218
TP0IOC2	TMP0 I/O 控制寄存器 2	Timer	219
TP0OPT0	TMP0 选项寄存器 0	Timer	220
TP0OVIC	中断控制寄存器	INTC	673
TP1CCIC0	中断控制寄存器	INTC	673
TP1CCIC1	中断控制寄存器	INTC	673
TP1CCR0	TMP1 捕捉/比较寄存器 0	Timer	221
TP1CCR1	TMP1 捕捉/比较寄存器 1	Timer	223
TP1CNT	TMP1 计数读缓冲寄存器	Timer	225
TP1CTL0	TMP1 控制寄存器 0	Timer	215
TP1CTL1	TMP1 控制寄存器 1	Timer	215
TP1IOC0	TMP1 I/O 控制寄存器 0	Timer	217
TP1IOC1	TMP1 I/O 控制寄存器 1	Timer	218
TP1IOC2	TMP1 I/O 控制寄存器 2	Timer	219
TP1OPT0	TMP1 选项寄存器 0	Timer	220
TP1OVIC	中断控制寄存器	INTC	673
TP2CCIC0	中断控制寄存器	INTC	673
TP2CCIC1	中断控制寄存器	INTC	673
TP2CCR0	TMP2 捕捉/比较寄存器 0	Timer	221
TP2CCR1	TMP2 捕捉/比较寄存器 1	Timer	223
TP2CNT	TMP2 计数读缓冲寄存器	Timer	225
TP2CTL0	TMP2 控制寄存器 0	Timer	215
TP2CTL1	TMP2 控制寄存器 1	Timer	215
TP2IOC0	TMP2 I/O 控制寄存器 0	Timer	217
TP2IOC1	TMP2 I/O 控制寄存器 1	Timer	218
TP2IOC2	TMP2 I/O 控制寄存器 2	Timer	219
TP2OPT0	TMP2 选项寄存器 0	Timer	220
TP2OVIC	中断控制寄存器	INTC	673
TP3CCIC0	中断控制寄存器	INTC	673
TP3CCIC1	中断控制寄存器	INTC	673
TP3CCR0	TMP3 捕捉/比较寄存器 0	Timer	221
TP3CCR1	TMP3 捕捉/比较寄存器 1	Timer	223
TP3CNT	TMP3 计数读缓冲寄存器	Timer	225
TP3CTL0	TMP3 控制寄存器 0	Timer	215
TP3CTL1	TMP3 控制寄存器 1	Timer	215
TP3IOC0	TMP3 I/O 控制寄存器 0	Timer	217
TP3IOC1	TMP3 I/O 控制寄存器 1	Timer	218
TP3IOC2	TMP3 I/O 控制寄存器 2	Timer	219
TP3OPT0	TMP3 选项寄存器 0	Timer	220
TP3OVIC	中断控制寄存器	INTC	673
TP4CCIC0	中断控制寄存器	INTC	673
TP4CCIC1	中断控制寄存器	INTC	673
TP4CCR0	TMP4 捕捉/比较寄存器 0	Timer	221
TP4CCR1	TMP4 捕捉/比较寄存器 1	Timer	223
TP4CNT	TMP4 计数读缓冲寄存器	Timer	225

符号	名称	单元	页码
TP4CTL0	TMP4 控制寄存器 0	Timer	215
TP4CTL1	TMP4 控制寄存器 1	Timer	215
TP4IOC0	TMP4 I/O 控制寄存器 0	Timer	217
TP4IOC1	TMP4 I/O 控制寄存器 1	Timer	218
TP4IOC2	TMP4 I/O 控制寄存器 2	Timer	219
TP4OPT0	TMP4 选项寄存器 0	Timer	220
TP4OVIC	中断控制寄存器	INTC	673
TP5CCIC0	中断控制寄存器	INTC	673
TP5CCIC1	中断控制寄存器	INTC	673
TP5CCR0	TMP5 捕捉/比较寄存器 0	Timer	221
TP5CCR1	TMP5 捕捉/比较寄存器 1	Timer	223
TP5CNT	TMP5 计数读缓冲寄存器	Timer	225
TP5CTL0	TMP5 控制寄存器 0	Timer	215
TP5CTL1	TMP5 控制寄存器 1	Timer	215
TP5IOC0	TMP5 I/O 控制寄存器 0	Timer	217
TP5IOC1	TMP5 I/O 控制寄存器 1	Timer	218
TP5IOC2	TMP5 I/O 控制寄存器 2	Timer	219
TP5OPT0	TMP5 选项寄存器 0	Timer	220
TP5OVIC	中断控制寄存器	INTC	673
TQ0CCIC0	中断控制寄存器	INTC	673
TQ0CCIC1	中断控制寄存器	INTC	673
TQ0CCIC2	中断控制寄存器	INTC	673
TQ0CCIC3	中断控制寄存器	INTC	673
TQ0CCR0	TMQ0 捕捉/比较寄存器 0	Timer	309
TQ0CCR1	TMQ0 捕捉/比较寄存器 1	Timer	311
TQ0CCR2	TMQ0 捕捉/比较寄存器 2	Timer	313
TQ0CCR3	TMQ0 捕捉/比较寄存器 3	Timer	315
TQ0CNT	TMQ0 计数读缓冲寄存器	Timer	317
TQ0CTL0	TMQ0 控制寄存器 0	Timer	303
TQ0CTL1	TMQ0 控制寄存器 1	Timer	304
TQ0IOC0	TMQ0 I/O 控制寄存器 0	Timer	305
TQ0IOC1	TMQ0 I/O 控制寄存器 1	Timer	306
TQ0IOC2	TMQ0 I/O 控制寄存器 2	Timer	307
TQ0OPT0	TMQ0 选项寄存器 0	Timer	308
TQ0OVIC	中断控制寄存器	INTC	673
UA0CTL0	UARTA0 控制寄存器 0	UARTA	475
UA0CTL1	UARTA0 控制寄存器 1	UARTA	497
UA0CTL2	UARTA0 控制寄存器 2	UARTA	498
UA0OPT0	UARTA0 选项控制寄存器 0	UARTA	477
UA0RIC	中断控制寄存器	INTC	673
UA0RX	UARTA0 接收数据寄存器	UARTA	480
UA0STR	UARTA0 状态寄存器	UARTA	478
UA0TIC	中断控制寄存器	INTC	673
UA0TX	UARTA0 发送数据寄存器	UARTA	480
UA1CTL0	UARTA1 控制寄存器 0	UARTA	475

符号	名称	单元	页码
UA1CTL1	UARTA1 控制寄存器 1	UARTA	497
UA1CTL2	UARTA1 控制寄存器 2	UARTA	498
UA1OPT0	UARTA1 选项控制寄存器 0	UARTA	477
UA1RIC	中断控制寄存器	INTC	673
UA1RX	UARTA1 接收数据寄存器	UARTA	480
UA1STR	UARTA1 状态寄存器	UARTA	478
UA1TIC	中断控制寄存器	INTC	673
UA1TX	UARTA1 发送数据寄存器	UARTA	480
UA2CTL0	UARTA2 控制寄存器 0	UARTA	475
UA2CTL1	UARTA2 控制寄存器 1	UARTA	497
UA2CTL2	UARTA2 控制寄存器 2	UARTA	498
UA2OPT0	UARTA2 选项控制寄存器 0	UARTA	477
UA2RIC	中断控制寄存器	INTC	673
UA2RX	UARTA2 接收数据寄存器	UARTA	480
UA2STR	UARTA2 状态寄存器	UARTA	478
UA2TIC	中断控制寄存器	INTC	673
UA2TX	UARTA2 发送数据寄存器	UARTA	480
VSWC	系统等待控制寄存器	CPU	84
WDTE	看门狗定时器允许寄存器	WDT	420
WDTM2	看门狗定时器模式寄存器 2	WDT	678
WTIC	中断控制寄存器	INTC	673
WTIIC	中断控制寄存器	INTC	673
WTM	钟表定时器操作模式寄存器	WT	411

附录 C 指令集列表

C.1 常规指令

(1) 用于描述操作数的寄存器符号

寄存器符号	解释
reg1	通用寄存器: 用作源寄存器。
reg2	通用寄存器: 主要用作目的寄存器。在有些指令中也用作源寄存器。
reg3	通用寄存器: 主要用作存放除法运算结果的余数或乘法运算结果的高 32 位。
bit#3	用于指定位编号的 3 位数据
immX	X 位立即数据
dispX	X 位偏移量数据
regID	系统寄存器编号
vector	用于指定陷阱向量的 5 位数据(00H 到 1FH)
cccc	表示条件代码的 4 位数据
sp	堆栈指针 (r3)
ep	元素指针(r30)
listX	X 个寄存器列表

(2) 用于描述操作码的寄存器符号

寄存器符号	解释
R	指定 reg1 或 regID 的 1 位数据码
r	指定 reg2 的 1 位数据码
w	指定 reg3 的 1 位数据码
d	1 位位移数据
l	1 位立即数据 (指示立即数据的高位)
i	1-bit 立即数据
cccc	表示条件代码的 4 位数据
CCCC	表示 Bcond 指令的条件代码的 4 位数据
bbb	用于指定位编号的 3 位数据
L	用于指定在寄存器列表中的程序寄存器的 1 位数据

(3) 用于描述操作数的寄存器符号

寄存器符号	解释
←	输入到
GR []	通用寄存器
SR []	系统寄存器
zero-extend (n)	以 0 填充以扩展至字边界
sign-extend (n)	以符号填充以扩展至字边界
load-memory (a, b)	从地址 a 读数据 b 。
store-memory (a, b, c)	写长度为 c 的数据 b 到地址 a 。
load-memory-bit (a, b)	读地址 a 的 b 位。
store-memory-bit (a, b, c)	写入 c 到 地址 a 的 b 位。
saturated (n)	执行 n 的饱和处理 (n 为补码形式). 结算结果为 若 $n \geq 7FFFFFFFH$, 则 $7FFFFFFFH$. 若 $n \leq 80000000H$, 则 $80000000H$.
result	以标识来反映计算结果
Byte	字节 (8 位)
Halfword	半个字 (16 位)
Word	字 (32 位)
+	加
−	减
	位连接
×	乘法
÷	除法
%	除法结果余数
AND	逻辑与
OR	逻辑或
XOR	异或
NOT	逻辑非
logically shift left by	逻辑左移
logically shift right by	逻辑右移
arithmetically shift right by	算术右移

(4) 用于表示操作时钟的寄存器符号

寄存器符号	解释
i	在指令执行后立即执行另一条指令(issue)。
r	在指令执行后立即重复同一指令(repeat)。
l	在指令执行后在指令中立即使用执行结果(latency)。

(5) 用于描述标志操作的寄存器符号

标识符	解释
(空)	不变
0	清零
X	依据结果设置或清零
R	恢复预先保存的值。

(6) 条件代码

条件代码 (cccc)	条件公式	解释
0 0 0 0	$OV = 1$	溢出
1 0 0 0	$OV = 0$	没有溢出
0 0 0 1	$CY = 1$	进位 小于(小于)
1 0 0 1	$CY = 0$	没有进位 不低于(大于或等于)
0 0 1 0	$Z = 1$	零
1 0 1 0	$Z = 0$	非零
0 0 1 1	$(CY \text{ or } Z) = 1$	不大于(小于或等于)
1 0 1 1	$(CY \text{ or } Z) = 0$	大于(大于)
0 1 0 0	$S = 1$	负
1 1 0 0	$S = 0$	正
0 1 0 1	—	一直(无论什么情况下)
1 1 0 1	$SAT = 1$	饱和
0 1 1 0	$(S \text{ xor } OV) = 1$	有符号小于
1 1 1 0	$(S \text{ xor } OV) = 0$	有符号大于或等于
0 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 1$	有符号小于或等于
1 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 0$	有符号大于

C.2 指令集 (按字母顺序)

(1/6)

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRR R	GR[reg2]←GR[reg2]+GR[reg1]	1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiii	GR[reg2]←GR[reg2]+符号扩展(imm5)	1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRR R iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+符号扩展(imm16)	1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRR R	GR[reg2]←GR[reg2]AND GR[reg1]	1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRR R iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND 零扩展(imm16)	1	1	1		0	×	×	
Bcond	disp9	ddddd1011ddcccc 注 1	如果条件满足	2	2	2					
			当条件满足时	注2	注2	注2					
			当条件不满足时	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) GR[reg2] (31 : 24) GR[reg2] (7 : 0) GR[reg2] (15 : 8)	1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) GR[reg2] (15 : 8) GR [reg2] (23 : 16) GR[reg2] (31 : 24)	1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+零扩展(imm6 逻辑左移 1 位) PC←CTBP+零扩展(装载存储器(adr,Halfword))	4	4	4					
CLR1	bit#3,disp16[reg1]	10bbb111110RRRRR dddddddddddddd	adr←GR[reg1]+符号扩展(disp16) Z flag←Not(装载存储器位(adr,bit#3)) 存储存储器位(adr,bit#3,0)	3 注3	3 注3	3 注3				×	
	reg2,[reg1]	rrrrr111111RRRR R 0000000011100100	adr←GR[reg1] Z flag←Not(装载存储器位(adr,reg2)) 存储存储器位(adr,reg2,0)	3 注3	3 注3	3 注3				×	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiii wwwwww011000cccc0	若条件满足 则 GR[reg3]←sign-extended(imm5) 否则 GR[reg3]←GR[reg2]	1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRR wwwwww011001cccc0	若条件满足 则 GR[reg3]←GR[reg1] 否则 GR[reg3]←GR[reg2]	1	1	1					
CMP	reg1,reg2	rrrrr001111RRRR R	result←GR[reg2]-GR[reg1]	1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010011iiii	result←GR[reg2]-符号扩展(imm5)	1	1	1	×	×	×	×	
CTRET		000001111100000 0000000101000100	PC←CTPC PSW←CTPSW	3	3	3	R	R	R	R	R
DBRET		000001111100000 0000000101000110	PC←DBPC PSW←DBPSW	3	3	3	R	R	R	R	R

(2/6)

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
DBTRAP		1111100001000000	DBPC←PC+2 (restored PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3					
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5,list12	0000011001iiii LLLLLLLLLLLL0000 0	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded	n+1 注4	n+1 注4	n+1 注4					
	imm5,list12,[reg1]	0000011001iiii LLLLLLLLLLLLRRRR R 注5	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded PC←GR[reg1]	n+3 注4	n+3 注4	n+3 注4					
DIV	reg1,reg2,reg3	rrrrr11111RRRR Rwwwwww010110000 00	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVH	reg1,reg2	rrrrr000010RRRR R	GR[reg2]←GR[reg2]÷GR[reg1] ^{注6}	35	35	35		×	×	×	
	reg1,reg2,reg3	rrrrr11111RRRR Rwwwwww010110000 00	GR[reg2]←GR[reg2]÷GR[reg1] ^{注6} GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	rrrrr11111RRRR Rwwwwww010110000 10	GR[reg2]←GR[reg2]÷GR[reg1] ^{注6} GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	rrrrr11111RRRR Rwwwwww010110000 10	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	Stop	1	1	1					
HSW	reg2,reg3	rrrrr11111100000 wwwwww01101000100	GR[reg3]←GR[reg2]:(15:0) GR[reg2] (31:16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110ddddd dddddddddddddd0 注7	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110ddddd dddddddddddddd0 注7	PC←PC+sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	rrrrr111000RRRR R dddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adrs,Byte))	1	1	注11					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRR Rdddddddddddddd1 注8, 10	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adrs,Byte))	1	1	注11					

(3/6)

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
LD.H	disp16[reg1],reg2	rrrrr111001RRRRRddd ddddddddddd0 注 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load- memory(adr,Halfword))	1	1	注 11					
LDSR	reg2,regID	rrrrr111111RRRRR000 0000000100000 注 12	SR[regID]←GR[reg2]]	1	1	1					
			regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRRddd ddddddddddd1 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	rrrrr111001RRRRRddd ddddddddddd1 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	rrrrr110001RRRRRi iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRRi iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 0 ¹⁶)	1	1	1					
MUL	reg1,reg2,reg3	rrrrr111111RRRRRww www01000100000	GR[reg3] GR[reg2]←GR[reg2]xGR[reg1] 注 14	1	4	5					
	imm9,reg2,reg3	rrrrr111111iiiiww www01001111100 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	rrrrr110111RRRRRi iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] ^{#6} ximm16	1	1	2					
MULU	reg1,reg2,reg3	rrrrr111111RRRRRww www01000100010	GR[reg3] GR[reg2]←GR[reg2]xGR[reg1] 注 14	1	4	5					
	imm9,reg2,reg3	rrrrr111111iiiiww www0100111110 注 13	GR[reg3] GR[reg2]←GR[reg2]xzero- extend(imm9)	1	4	5					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.	1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])	1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRRddd ddddddddddd	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				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)	3 注 3	3 注 3	3 注 3				×	

(4/6)

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
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii i	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiL LLLLLLLLLLL00001	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 ^{注15}	0000011110iiiiL LLLLLLLLLLLff011 imm16/imm32 注16	Store-memory(sp-4,GR[reg in list12],Word) sp←sp+4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5) ep←sp/imm	n+2 注4 注17	n+2 注4 注17	n+2 注4 注17					
RETI		000001111100000 000000010100000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R
SAR	reg1,reg2	rrrrr11111RRRRR 000000001010000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr111110cccc 000000100000000	若条件满足 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	rrrrr111110cccc 000000000000000	若条件满足 then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1					

(5/6)

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
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR dddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 注3	3 注3	3 注3				×	
	reg2,[reg1]	rrrrr111111RRRR R000000001110000 0	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 注3	3 注3	3 注3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110dddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	注9					
SLD.BU	disp4[ep],reg2	rrrrr0000110dddd 注 18	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	注9					
SLD.H	disp8[ep],reg2	rrrrr1000dddddd 注 19	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	注9					
SLD.HU	disp5[ep],reg2	rrrrr0000111dddd 注 s 18, 20	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	注9					
SLD.W	disp8[ep],reg2	rrrrr1010dddddd0 注 21	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	注9					
SST.B	reg2,disp7[ep]	rrrrr0111dddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001dddddd 注 19	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Halfword)	1	1	1					
SST.W	reg2,disp8[ep]	rrrrr1010dddddd1 注 21	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1					
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR dddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR dddddddddddddd0 注 8	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Halfword)	1	1	1					
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR dddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Word)	1	1	1					
STSR	regID,reg2	rrrrr111111RRRRR 000000001000000	GR[reg2]←SR[regID]	1	1	1					

(6/6)

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]-GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]-GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	0000000010RRRRR	adr←(PC+2) + (GR [reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr,Halfword)) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15 : 0))	1	1	1					
TRAP	向量	000001111111iiii 0000000100000000	EIPC ←PC+4 (Restored PC) EIPSW ←PSW ECR.EICC ←Interrupt code PSW.EP ←1 PSW.ID ←1 PC ←00000040H (当向量是 00H 到 0FH) 00000050H (当向量是 10H 到 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR dddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr,bit#3))	3 注3	3 注3	3 注3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3 注3	3 注3	3 注3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15 : 0))	1	1	1					

- 注
1. dddddddd: disp9 的高 8 位。
 2. 如果有重写即刻的 PSW 内容的指令，那么时钟数为 3。
 3. 如果没有等待状态(3 + 读取访问的等待状态数)。
 4. n 为 list12 装载寄存器的总数。(由等待状态数而得。如果没有等待状态，那么 n 即为 list12 寄存器的总数。若 n = 0，则与 n = 1 时进行同样的操作)。
 5. RRRRR: 00000 除外。
 6. 仅低半字数据有效。
 7. ddddddddddddddddddd: disp22 的高 21 位。
 8. ddddddddddddddd: disp16 的高 15 位。
 9. 根据等待状态的数目(如果没有等待状态为 1)。
 10. b: disp16 的第 0 位。
 11. 根据等待状态的数目(如果没有等待状态为 2)。

- 注 12. 在该指令中，为了便于助记时式的描述，源寄存器被作为 **reg2** 而 **reg1** 被用作操作码。因此，助记描述中的寄存器指定方法和操作码中的指定方法有所不同。
- rrrrr = regID 指定
RRRRR = reg2 指定
13. iiii: imm9 的低 5 位。
IIII: imm9 的高 4 位。
14. 对于通用寄存器 **reg1** 和 **reg3**，不要指定相同的寄存器。
15. sp/imm: 由副操作码的位 19 和位 20 指定。
16. ff = 00: 将 sp 载入 ep.
01: 将符号扩展 16 位立即数据(47 到 32 位)载入 ep
10: 将逻辑左移 16 位的 16 位立即数据(47 到 32 位)载入 ep
11: 加载 32 位立即数 (bits 63 到 32) in ep.
17. 如果 imm = imm32, n + 3 个时钟。
18. rrrrr: 除 00000 外。
19. ddddddd: disp8 的高 7 位。
20. dddd: disp5 的高 4 位。
21. ddddddd: disp8 的高 6 位。

附录 D 注意事项表

此附录列明了本文档所需的注意事项。

下表的“分类 (硬件/软件)”定义：

硬件： 有关微控制器内部/外部硬件的注意事项。

软件： 有关寄存器设置或编程等软件的事项。

(1/36)

章节	分类	功能	具体功能	注意事项	页码
第 1 章	硬件	介绍	FLMD0	正常模式下，将该引脚与 Vss 相连接。	pp. 23, 24
			REGC	将 REGC 通过一个 4.7uF 的电容连接到 Vss。	pp. 23, 24
第 2 章	软件	引脚功能	P05	连接一个下拉电阻。可通过设置 OCDM.OCDM0 为 0 断开连接。	p. 31
	硬件		DDO	在片上调试模式状态下， 强制设置为高电平输出。	p. 35
	软件		KR0 - KR7	外部连接上拉电阻。	p. 36
			NMI	引脚 NMI 和 P02 为复用功能引脚。复位后，引脚功能为 P02。要使能引脚 NMI，设置 PMC0.PMC02 为 1。引脚 NMI 的初始设置为“无边沿检测”。通过寄存器 INTF0 和 INTRO 选择引脚 NMI 的有效边沿。	p. 36
	硬件		当电源开始供电	当电源开始供电，下述引脚可能在复位期间输出不确定电平。 ● P10/ANO0 引脚 ● P11/ANO1 引脚 ● P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 引脚	p. 45
第 3 章	软件	CPU 功能	EIPC 寄存器, EIPSW 寄存器, FEPC 寄存器, FEPSW 寄存器	由于中断状态保存寄存器只有一组，因此，要实现多重中断需用程序对这组寄存器进行保存。	p. 49
			EIPC, FEPC, CTPC	即使 EIPC, FEPC 或 CTPC 的第 0 位被 LDSR 指令置 1，在中断处理程序执行后程序由 RETI 指令返回时，第 0 位仍然被忽略(这是因为 PC 的第 0 位固定为 0)。因此请使用偶数数值(第 0 位为 0)来设置 EIPC, FEPC 或 CTPC。	p. 49
			程序空间	由于从地址 03FFF000H 到 03FFFFFFH 的 4KB 范围是片上外围 I/O 设备区，这个区域不能进行取指令操作。因此，要避免执行可能使分支地址计算结果为此 4KB 中值的操作。	p. 57
			片上周边 I/O 区域	当寄存器被一个字操作指令读写时，一个字的空间被分为低 16 位和高 16 位分两次被字节操作先后读写，其中最低 2 位地址被忽略	p. 66
				若对只能进行字节操作的寄存器进行半字长(halfword)操作，那么读取时，高 8 位数据无效，写入时，低 8 位数据被写入寄存器。	p. 66
				未被指定为寄存器的地址为未来扩展保留。对这些地址操作结果的有效性是无法保证的	p. 66
			外部存储区域	V850ES/JG2 具有 22 个地址引脚(A0 到 A21)，因此外部存储空间可视为一个重复的 4MB 的镜像。使用此 22 个地址引脚时，一定要保证 EVDD = BVDD = VDD。	p. 66

章节	分类	功能	具体功能	注意事项	页码
第 3 章	软件	CPU 功能	内部 RAM 区域	如果分支指令在内部 RAM 区间的上沿地址处发生, 那么将不会产生跨越片上外围 I/O 区间的预取址(无效取址)。	p. 67
			将数据设置给特殊寄存器	设置 IDLE1, IDLE2 或 STOP 模式(通过将 PSC.STP 位置 1)后, 要立即插入 5 条 NOP 指令	p. 81
				当对命令寄存器进行写操作时, 系统将不响应中断。存储指令应该是按上述的第<3>和<4>步骤连续执行的。若在步骤<3>和<4>之间执行了其他指令, 而这个指令又进行了中断的响应, 则将打乱上述的顺序导致设备误操作。	p. 81
				尽管对 PRCMD 写入的数据为虚数据, 但还是请使用与设置特殊寄存器(例子中的<4>)的通用寄存器相同的寄存器来向 PRCMD 寄存器(例子中的<3>)写入虚数据。这个过程也同样可以用于使用通用寄存器进行寻址的操作中。	p. 81
			SYS 寄存器	如果在对 PRCMD 寄存器进行写入之后, 向 SYS 寄存器(并非特殊寄存器)的 PRERR 位写入 0, 那么 PRERR 位将被清零。(写入命令优先)	p. 83
				如果在对 PRCMD 寄存器进行写入之后, 又执行了对 PRCMD 寄存器(并非特殊寄存器)的写入操作, 那么 PRERR 位将被置 1。	p. 83
			需要首先设置的寄存器	在使用 V850ES/JG2 器件时, 请务必首先对以下寄存器进行设置。 ● 系统等待控制寄存器 (VSWC) ● 片上调试模式寄存器 (OCDM) ● 看门狗定时器模式寄存器 2 (WDTM2)	p. 84
			VSWC 寄存器	访问一个片上周边 I/O 寄存器需要 3 个时钟周期的时间(不包含等待周期)。V850ES/JG2 需要根据不同的操作频率插入不同的等待周期数。因此, 请根据所使用的时钟, 将以下对应的数值设置到 VSWC 寄存器中。	p. 84
			访问特定的片上周边 I/O 寄存器	以下状态中, 禁止访问上述的寄存器。如果这时产生了等待状态, 那么只有复位才可退出等待状态。 ● 当主时钟振荡停止, 系统工作于子时钟时 ● 当 CPU 工作于内置振荡器时钟时	p. 85
	硬件	系统保留区域	V850ES/JG2 产品中, 地址 0000007AH 到 0000007FH 是用于功能扩展的系统保留区域, 因此建议不要使用该区域。	p. 86	
			如果 flash 存储器中的数据被擦除, 那么所有的数据位都将被置 1。	p. 86	
第 4 章	硬件	端口功能	基本端口配置	端口 0, 3 至 5, 和 9 为 5 V 容限。	p. 88
	软件		PFn 寄存器	PFn 寄存器的 PFnm 位只有在端口模式下 (PMCnm 位 = 0) PMn 寄存器的 PMnm 位为 0 (指定为输出模式) 时有效。当 PMnm 位为 1 (指定为输入模式) 时, 对 PFn 寄存器设置的值无效。	p. 92
	硬件, 软件		端口 0	DRST 引脚用于片上调试。 如果不使用片上调试, 从 RESET 引脚复位释放到 OCDM.OCDM0 位清零期间, 需要将 P05/INTP2/DRST 引脚固定为低电平。 详细内容请参考 4.6.3 片上调试引脚的注意事项。	p. 94
				P02 至 P06 引脚在复用功能的输入模式具有滞后性, 但在端口模式没有滞后性。	p. 94
	软件		PMC0 寄存器	当 OCDM.OCDM0 位 = 1 时无论 PMC05 位的值如何, P05/INTP2/DRST 引脚都会变为 DRST 引脚。	p. 95

章节	分类	功能	具体功能	注意事项	页码
第 4 章	软件	端口功能	PF0 寄存器	如果一个输出引脚被拉高 至 EVDD 或更高, 请务必设置 PF0n 位为 1。	p. 96 <input type="checkbox"/>
			端口 1	当电源打开时, 即使在复位期间, P10 和 P11 引脚也可能会暂时输出不定的电平。	p. 97 <input type="checkbox"/>
			P1 寄存器	在 D/A 转换器件不要读写 P1 寄存器 (参看 14.4.3 注意事项)。	p. 97 <input type="checkbox"/>
			PM1 寄存器	当使用 P1n 作为复用功能 (ANOn 引脚输出) 时, 需要将 PM1n 位置 1。	p. 97 <input type="checkbox"/>
	当使用 P10 和 P11 其中一个引脚作为 I/O 端口, 另一个作为 D/A 输出引脚时, 当进行 AD 输出的时候不能改变 I/O 端口的输出电平。			p. 97 <input type="checkbox"/>	
	端口 3		P31 至 P35, P38, 和 P39 引脚在复用功能的输入模式具有滞后性, 但在端口模式没有滞后性。	p. 98 <input type="checkbox"/>	
	P3 寄存器		要以 8 位 或 1 位为单位读/写 P3 寄存器的位 8 至 15 , 需要将他们指定为 P3H 寄存器的位 0 至 7。	p. 99 <input type="checkbox"/>	
	PM3 寄存器		要以 8 位 或 1 位为单位读/写 PM3 寄存器的位 8 至 15 , 需要将他们指定为 PM3H 寄存器的位 0 至 7。	p. 99 <input type="checkbox"/>	
	PMC3 寄存器		务必把第 15-10, 7, 和 6 位清零。	p. 100 <input type="checkbox"/>	
			要以 8 位 或 1 位为单位读/写 PMC3 寄存器的位 8 至 15 , 需要将他们指定为 PMC3H 寄存器的位 0 至 7。	p. 100 <input type="checkbox"/>	
	PFC3 寄存器		要以 8 位 或 1 位为单位读/写 PFC3 寄存器的位 8 至 15 , 需要将他们指定为 PFC3H 寄存器的位 0 至 7。	p. 101 <input type="checkbox"/>	
	端口 3 复用功能 设定		INTP7 引脚和 RXDA0 引脚为一组复用功能引脚。当用作 RXDA0 引脚时, 需要禁止 INTP7 复用功能引脚的边沿检测 (将 INTF3.INTF31 位和 INTR3.INTR31 位清 0)。当用作 INTP7 引脚时, 需要停止 UARTA0 接收 (将 UA0CTL0.UA0RXE 位清 0)。	p. 102 <input type="checkbox"/>	
	PF3 寄存器		如果一个输出引脚被拉高 至 EVDD 或更高, 请务必设置 PF3n 位为 1。	p. 103 <input type="checkbox"/>	
			2. 要以 8 位 或 1 位为单位读/写 PF3 寄存器的位 8 至 15 , 需要将他们指定为 PF3H 寄存器的位 0 至 7。	p. 103 <input type="checkbox"/>	
	端口 4		P40 至 P42 引脚在复用功能的输入模式具有滞后性, 但在端口模式没有滞后性。	p. 104 <input type="checkbox"/>	
	PF4 寄存器		如果一个输出引脚被拉高 至 EVDD 或更高, 请务必设置 PF4n 位为 1。	p. 105 <input type="checkbox"/>	
	硬件, 软件	端口 5	DDI, DDO, DCK, 和 DMS 引脚用作片上调试。 如果不使用片上调试功能, 从 RESET 引脚复位释放到 OCDM.OCDM0 位清 (0) 期间, 需要将 P05/INTP2/DRST 引脚固定为低电平。 详细信息请参考 4.6.3 片上调试引脚的注意事项。	p. 106 <input type="checkbox"/>	
			当电源打开时, 即使在复位期间, P53 引脚也可能会暂时输出不定的电平。	p. 106 <input type="checkbox"/>	
			P50 至 P55 引脚在复用功能的输入模式具有滞后性, 但在端口模式没有滞后性。	p. 106 <input type="checkbox"/>	
	软件	端口 5 模式寄存器 (PM5)	KRn 引脚和 TIQ0m 引脚为一组复用功能引脚。当作为 TIQ0m 引脚使用时, 需要禁止复用功能 KRn 引脚的按键返回检测 (将 KRM.KRMn 位清 0), 当作为 KRn 引脚使用时, 需要禁止复用功能 TIQ0m 引脚的边沿检测。	p. 109 <input type="checkbox"/>	
		PF5 寄存器	如果一个输出引脚被拉高 至 EVDD 或更高, 请务必设置 PF5n 位为 1。	p. 109 <input type="checkbox"/>	

章节	分类	功能	具体功能	注意事项	页码
第 4 章	软件	端口功能	P7H 寄存器, P7L 寄存器	在 A/D 转换期间不要读取 P7H 和 P7L 寄存器 (参看 13.6 (4) 复用 I/O)。	p. 111 <input type="checkbox"/>
			PM7H 寄存器, PM7L 寄存器	当使用 P7n 引脚的复用功能(ANIn 引脚)时, 需要将 PM7n 位置 1。	p. 111 <input type="checkbox"/>
	硬件		端口 9	P90 - P97, P99, P910, 和 P912 至 P915 引脚在复用功能的输入模式具有滞后性, 但在端口模式没有滞后性。	p. 112 <input type="checkbox"/>
			P9 寄存器	要以 8 位 或 1 位为单位读/写 P9 寄存器的位 8 至 15, 需要将他们指定为 P9H 寄存器的位 0 至 7。	p. 113 <input type="checkbox"/>
	软件		PM9 寄存器	要以 8 位 或 1 位为单位读/写 PM9 寄存器的位 8 至 15, 需要将他们指定为 PM9H 寄存器的位 0 至 7。	p. 113 <input type="checkbox"/>
			PMC9 寄存器	要以 8 位 或 1 位为单位读/写 PMC9 寄存器的位 8 至 15, 需要将他们指定为 PMC9H 寄存器的位 0 至 7。	p. 114 <input type="checkbox"/>
				只有当 P90 至 P915 引脚全部用作复用功能 A0 至 A15 引脚时, 需要立刻将 PMC9 寄存器的全部 16 位设置为 FFFFH。	p. 115 <input type="checkbox"/>
			PFC9 寄存器	当执行单独的地址总线输出 (A0 至 A15) 时, 需要在将 PFC9 寄存器清除为 0000H 后立即将 PMC9 寄存器设置为 FFFFH。	p. 116 <input type="checkbox"/>
				要以 8 位或 1 位为单位读/写 PFC9 寄存器的位 8 至 15, 需要将他们指定为 PFC9H 寄存器的位 0 至 7。	p. 116 <input type="checkbox"/>
			PFCE9 寄存器	要以 8 位或 1 位为单位读/写 PFCE9 寄存器的位 8 至 15, 需要将他们指定为 PFCE9H 寄存器的位 0 至 7。	p. 116 <input type="checkbox"/>
			端口 9 复用功能指定	RXDA1 和 KR7 引脚切勿同时使用。使用 RXDA1 引脚就不要使用 KR7 引脚。当使用 KR7 引脚时, 不要使用 RXDA1 引脚 (推荐将 PFC91 位置 1 以及将 PFCE91 位清 0)。	p. 118 <input type="checkbox"/>
			PF9 寄存器	如果一个输出引脚被拉高 至 EVDD 或更高, 请务必设置 PF9n 位为 1。	p. 119 <input type="checkbox"/>
				要以 8 位或 1 位为单位读/写 PF9 寄存器的位 8 至 15, 需要将他们指定为 PF9H 寄存器的位 0 至 7。	p. 119 <input type="checkbox"/>
			PDL 寄存器	要以 8 位或 1 位为单位读/写 PDL 寄存器的位 8 至 15, 需要将他们指定为 PDLH 寄存器的位 0 至 7。	p. 127 <input type="checkbox"/>
			PMDL 寄存器	要以 8 位或 1 位为单位读/写 PMDL 寄存器的位 8 至 15, 需要将他们指定为 PMDLH 寄存器的位 0 至 7。	p. 127 <input type="checkbox"/>
			PMCDL 寄存器	当 EXIMC 寄存器的 SMSEL 位 = 1 (单独模式) 并且 BSC 寄存器的 BS30 至 BS00 位 = 0 (8 位总线带宽), 不要指定 AD8 至 AD15 引脚。	p. 128 <input type="checkbox"/>
				要以 8 位或 1 位为单位读/写 PMCDL 寄存器的位 8 至 15, 需要将他们指定为 PMCDLH 寄存器的位 0 至 7。	p. 128 <input type="checkbox"/>
			使用端口引脚作为复用功能引脚	INTP7 引脚和 RXDA0 引脚为一组复用功能引脚。当用作 RXDA0 引脚时, 需要禁止 INTP7 复用功能引脚的边沿检测 (将 INTF3.INTF31 位和 INTR3.INTR31 位清 0)。当用作 INTP7 引脚时, 需要停止 UAR0A0 接收 (将 UA0CTL0.UA0RXE 位清 0)。	p. 160 <input type="checkbox"/>
				当使用 P10 和 P11 其中一个引脚作为 I/O 端口, 另一个作为 D/A 输出引脚时 (ANO0, ANO1), 在 D/A 输出期间端口 I/O 电平不改变。	p. 160 <input type="checkbox"/>
				只有当用作复用功能 A0 至 A15 引脚时, 需要立刻将 PMC9 寄存器的全部 16 位设置为 FFFFH。	p. 163 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 4 章	软件	端口功能	使用端口引脚作为复用功能引脚	RXDA1 和 KR7 引脚切勿同时使用。使用 RXDA1 引脚就不要使用 KR7 引脚。当使用 KR7 引脚时，不要使用 RXDA1 引脚 (推荐将 PFC91 位置 1 以及将 PFCE91 位清 0)。	p. 163 <input type="checkbox"/>
			端口模式切换到复用功能模式的注意事项	要从端口模式切换到复用功能模式需要按照以下顺序执行。 <1> 设置 PFn 寄存器 [※] : N 沟道开漏设置 <2> 设置 PFCn 和 PFCEn 寄存器: 复用功能选择 <3> 将 PMCn 寄存器对应的位置 1: 切换到复用功能模式 如果首先设置了 PMCn 寄存器，需要注意，再按照 PFn, PFCn, 和 PFCEn 寄存器的设置时或基于这些设置，有可能发生意外的操作。	p. 167 <input type="checkbox"/>
				无论是端口模式/复用功能模式，读写 Pn 寄存器如下。 • Pn 寄存器读: 读取端口输出锁存值 (当 PMn.PMnm 位 = 0 时)，或读取引脚状态(PMn.PMnm 位 = 1)。 • Pn 寄存器写: 写入端口输出锁存。	p. 167 <input type="checkbox"/>
		复用功能模式 (输入) 的注意事项		当 PMCn.PMCnm 位为 0 时，由于 PMcN 寄存器设置值以及引脚电平 AND 输出，因此输入到复用功能模块的信号为低电平。因而，由于端口设置和复用功能操作允许时刻，可能会发生意外的操作。因此，要按照以下顺序从端口模式切换到复用功能模式。 • 要从端口模式切换到复用功能模式 (输入) 使用 PMcN 寄存器将引脚设置为复用功能模式，然后允许复用功能操作。 • 要从复用功能模式 (输入) 切换到端口模式 停止复用功能操作然后将引脚切换到端口模式。	p. 168 <input type="checkbox"/>
			端口模式下的 PFn.PFnm 位	在端口模式，PFn.PFnm 位只在输出模式下有效(PMn.PMnm 位 = 0)。在输入模式下 (PMnm 位 = 1)，PFnm 位的值不会映射到缓冲器中。	p. 169 <input type="checkbox"/>
			端口 n 寄存器 (Pn) 位操作指令的注意事项	当位操作指令执行在一个同时提供输入和输出功能的端口时，不属于操作对象目标位的输入端口的输出锁存的值也可能被写入。 因此，当端口从输入模式切换到输出模式时，推荐重写输出锁存。	p. 170 <input type="checkbox"/>
	硬件, 软件	片上调试引脚的注意事项		如果不使用片上调试，则需要采取以下措施。 • 将 OCDM 寄存器(特殊寄存器)的 OCDM0 位清(0) 此时，从通过 RESET 引脚复位释放，到采取了以上措施期间，将 P05/INTP2/DRST 引脚固定为低电平 采取以上措施之前，如果将高电平输入到 DRST 引脚，可能会导致发生故障 (CPU 死锁)。因此在处理 P05 引脚时要非常小心。	p. 171 <input type="checkbox"/>
				通过 WDT2RES, 时钟监视器 (CLM), 或低电压检测 (LVI) 信号复位后，P05/INTP2/DRST 引脚不初始化为片上调试引脚功能 (DRST)。OCDM 寄存器保持当前值。	p. 171 <input type="checkbox"/>
		P05/INTP2/DRST 引脚的注意事项		P05/INTP2/DRST 引脚有内部下拉电阻(30 kΩ 典型值)。通过 RESET 引脚复位后，就会连接下拉电阻。当 OCDM0 位清 (0) 时，下拉电阻断开。	p. 171 <input type="checkbox"/>
		当电源打开时 P10, P11, 和 P53 引脚的注意事项		当电源打开时，即使在复位期间以下引脚也可能暂时输出未定义的电平。 • P10/ANO0 引脚 • P11/ANO1 引脚 • P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 引脚。	p. 171 <input type="checkbox"/>
	硬件				

章节	分类	功能	具体功能	注意事项	页码
第4章	软件	端口功能	滞后性	在端口模式下, 以下端口引脚没有滞后性。 P02 - P06 P31 - P35, P38, P39 P40 - P42 P50 - P55 P90 - P97, P99, P910, P912 - P915	p. 171 <input type="checkbox"/>
第5章	软件	总线控制功能	在端口模式, 以下端口引脚没有滞后性。	对内置 ROM 区域进行写访问时地址、数据以及控制信号的激活方式和对外部存储器区域进行访问时的激活方式是相同的。	p. 173 <input type="checkbox"/>
			EXIMC 寄存器	请在对外部存储器进行访问之前从内置 ROM 或内置 RAM 进行对 EXIMC 寄存器的设置。 在设置了 EXIMC 寄存器之后, 请务必插入一条 NOP 指令	p. 175 <input type="checkbox"/>
			BSC 寄存器	请在复位后对 BSC 寄存器进行设置, 并且不要在设置完成后更改设置值。此外, 在对 BSC 寄存器进行初始设置之前, 请不要访问外部存储器区域。	p. 176 <input type="checkbox"/>
				请务必将第 14, 12, 10 和 8 位设置为 1, 将第 15, 13, 11, 9, 7, 5, 3 和 1 位清零。	p. 176 <input type="checkbox"/>
			DWC0 寄存器	内置 ROM 和内置 RAM 不会受可编程等待功能的影响, 对它们的访问不会被插入等待状态。片上周边 I/O 区域也同样不受可编程等待功能的影响, 该区域的访问只会受片上周边 I/O 器件的等待控制的影响。	p. 184 <input type="checkbox"/>
				请在复位后对 DWC0 寄存器进行设置, 并且不要在设置完成后更改设置值。此外, 在对 DWC0 寄存器进行初始设置之前, 请不要访问外部存储器区域。	p. 184 <input type="checkbox"/>
				请务必将第 15, 11, 7 和 3 位清零。	p. 184 <input type="checkbox"/>
			AWC 寄存器	访问内置 ROM 区域, 内置 RAM 区域以及片上周边 I/O 区域时不会插入地址设立和地址保持等待周期。	p. 187 <input type="checkbox"/>
				请在复位后对 AWC 寄存器进行设置, 并且不要在设置完成后更改设置值。此外, 在对 AWC 寄存器进行初始设置之前, 请不要访问外部存储器区域。	p. 187 <input type="checkbox"/>
				请务必将第 15 到第 8 位置 1。	p. 187 <input type="checkbox"/>
			BCC 寄存器	内置 ROM, 内置 RAM 和片上周边 I/O 区域不受空闲状态插入的影响。	p. 188 <input type="checkbox"/>
				请在复位后对 BCC 寄存器进行设置, 并且不要在设置完成后更改设置值。此外, 在对 BCC 寄存器进行初始设置之前, 请不要访问外部存储器区域。	p. 188 <input type="checkbox"/>
				请务必将第 15, 13, 11 和 9 位置 1, 并将第 14, 12, 10, 8, 6, 4, 2 和 0 位清零	p. 188 <input type="checkbox"/>
第6章	软件	时钟产生功能	PCC 寄存器	在通过 CLKOUT 向外输出时钟信号时, 不要改变 CPU 时钟的频率(通过 CK3 到 CK0 位)。	p. 202 <input type="checkbox"/>
				对 CK3 位操作时, 请使用位操作指令。如果使用 8 位操作指令, 不要改变 CK2 到 CK0 位的值	p. 202 <input type="checkbox"/>
				当停止主时钟工作时, 也要同时停止 PLL。另外, 工作于主时钟频率的片上周边器件也被停止。	p. 203 <input type="checkbox"/>
				如果不满足下面的条件, 那么请改变 CK2 到 CK0 位的设置以满足该条件, 然后再切换到子时钟工作模式。 内部系统时钟 (fclk) > 子时钟 (fxt: 32.768 kHz) × 4	p. 203 <input type="checkbox"/>

(7/36)

章节	分类	功能	具体功能	注意事项	页码
第 6 章	软件	时钟产生功能	PCC 寄存器	只有在主时钟振荡稳定之后才可以使工作于主时钟频率的片上周边器件开始工作。如果它们在振荡稳定之前就被启动, 则可能引起误操作。	p. 204 <input type="checkbox"/>
			RCM 寄存器	CPU 工作于内置振荡器时钟时(CCLS.CCLSIF 位 = 1), 不可以停止内置振荡器的工作。不要将 RSTOP 位置 1。	p. 205 <input type="checkbox"/>
				即使 RSTOP 位为 1, 如果 CCLS.CCLSIF 位被置 1 内置振荡器也会开始振荡。(振荡稳定时间中产生 WDT 溢出时)此时, RSTOP 位仍为 1。	p. 205 <input type="checkbox"/>
			PLLCTL 寄存器	当 PLLON 位被清零时, SELPLL 位也会被自动清零(进入时钟直连模式)。	p. 207 <input type="checkbox"/>
				只有在 PLL 时钟频率稳定后, SELPLL 位才能够被置 1。如果时钟频率没有稳定(未入锁), 那么无论向该位写入的数据为何值, SELPLL 位都会被写入"0"。	p. 207 <input type="checkbox"/>
			CKC 寄存器	fx = 5.0 ~ 10.0 MHz 时, 不可使用 PLL 模式。	p. 208 <input type="checkbox"/>
				在使用 CKC 寄存器在 4 倍和 8 倍间改变倍频系数之前, 要先设置时钟为直连模式并停止 PLL 工作。	p. 208 <input type="checkbox"/>
				请务必将第 3 位和第 1 位置于 1, 将第 7 到 4 和第 2 到 0 位清零。	p. 208 <input type="checkbox"/>
			LOCKR 寄存器	LOCK 寄存器不能实时反映 PLL 的锁定状态。	p. 209 <input type="checkbox"/>
			PLLS 寄存器	将入锁时间设置为 800 μ s 或更长。	p. 210 <input type="checkbox"/>
				入锁过程中不要改变 PLLS 寄存器的设置。	p. 210 <input type="checkbox"/>
第 7 章	软件	16 位定时器/事件计数器 P (TMP)	TPnCTL0 寄存器	在 TPnCE 位 = 0 时才可对 TPnCKS2 到 TPnCKS0 位进行设置。当将 TPnCE 位的值从 0 改变为 1 时, 可同时对 TPnCKS2 到 TPnCKS0 位进行设置。	p. 215 <input type="checkbox"/>
				请务必将第 3 到第 6 位清零。	p. 215 <input type="checkbox"/>
			TPnCTL1 寄存器	对 TPnEST 位的设置只有在外部触发脉冲输出模式或单脉冲输出模式才有效。在其它模式下, 对该位的置 1 操作将被忽略。	p. 216 <input type="checkbox"/>
				在外部事件计数模式下, 无论 TPnEEE 位如何设置, 外部事件计数输入都是被选中的。	p. 216 <input type="checkbox"/>
				在 TPnCTL0.TPnCE 位 = 0 时才可对 TPnEEE 和 TPnMD2 到 TPnMD0 位进行设置 (TPnCE 位 = 1 时只可以向这些位写入相同的值)。如果在 TPnCE 位 = 1 时对这些位进行了改写, 那么操作结果将不能被保证。若由于误操作引起了改写, 则要将 TPnCE 位清零然后再次设置这些寄存器位。	p. 216 <input type="checkbox"/>
				请务必将第 3, 4 和 7 位清零。	p. 216 <input type="checkbox"/>
			TPnIOC0 寄存器	在 TPnCTL0.TPnCE 位 = 0 时才可对 TPnOL1, TPnOE1, TPnOL0 和 TPnOE0 位进行改写 (TPnCE 位 = 1 时只可以向这些位写入相同的值)。若在 TPnCE 位 = 1 由于误操作引起了改写, 则要将 TPnCE 位清零然后再次设置这些寄存器位。	p. 217 <input type="checkbox"/>
				在 TPnCE 位和 TPnOEm 位为 0 时, 既是对 TPnOLm 位进行了操作, TOPnm 引脚的输出电平也不能确定。(m = 0, 1)。	p. 217 <input type="checkbox"/>
			TPnIOC1 寄存器	在 TPnCTL0.TPnCE 位 = 0 时才可对 TPnIS3 到 TPnIS0 位进行改写 (TPnCE 位 = 1 时只可以向这些位写入相同的值)。若在 TPnCE 位 = 1 由于误操作引起了改写, 则要将 TPnCE 位清零然后再次设置这些寄存器位。	p. 218 <input type="checkbox"/>
				TPnIS3 到 TPnIS0 位只有在自由运行定时器模式和脉宽测量模式下才有效。其它模式下, 无法进行捕捉操作。	p. 218 <input type="checkbox"/>
			TPnIOC2 寄存器	在 TPnCTL0.TPnCE 位 = 0 时才可对 TPnEES1, TPnEES0, TPnETS1 和 TPnETS0 位进行改写 (TPnCE 位 = 1 时只可以向这些位写入相同的值)。	p. 219 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 7 章	软件	16 位定时器/事件计数器 P (TMP)	TPnIOC2 寄存器	TPnEES1 和 TPnEES0 位只有在 TPnCTL1.TPnEEE 位 = 1 或设置了外部事件计数模式(TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0 位 = 001)时才有效。	p. 219 <input type="checkbox"/>
				TPnETS1 和 TPnETS0 位只有在外部触发脉冲输出模式(TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0 位 = 010) 或单脉冲输出模式(TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0 = 011) 下才有效。	p. 219 <input type="checkbox"/>
			TPnOPT0 寄存器	在 TPnCTL0.TPnCE 位 = 0 时才可对 TPnCCS1 和 TPnCCS0 位进行改写 (TPnCE 位 = 1 时只可以向这些位写入相同的值)。若在 TPnCE 位 = 1 由于误操作引起了改写, 则要将 TPnCE 位清零然后再次设置这些寄存器位。	p. 220 <input type="checkbox"/>
				请务必将第 1 到 3 位, 第 6 和第 7 位清零。	p. 220 <input type="checkbox"/>
			TPnCCR0 寄存器	以下情况下禁止访问 TPnCCR0 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器 <ul style="list-style-type: none"> 当 CPU 工作于子时钟且主时钟振荡停止时 当 CPU 工作于内置振荡时钟时 	p. 221 <input type="checkbox"/>
			TPnCCR1 寄存器	以下情况下禁止访问 TPnCCR1 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器 <ul style="list-style-type: none"> 当 CPU 工作于子时钟且主时钟振荡停止时 当 CPU 工作于内置振荡时钟时 	p. 223 <input type="checkbox"/>
			TPnCNT 寄存器	以下情况下禁止访问 TPnCNT 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。 <ul style="list-style-type: none"> 当 CPU 工作于子时钟且主时钟振荡停止时 当 CPU 工作于内置振荡时钟时 	p. 225 <input type="checkbox"/>
			操作	使用外部事件计数模式时, 请关闭对 TIPn0 引脚捕捉触发输入的有效边沿的检测 (通过将 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位清零)。	p. 226 <input type="checkbox"/>
				使用外部触发脉冲输出模式, 单脉冲输出模式和脉宽测量模式时, 请选择内部时钟作为计数时钟(通过将 TPnCTL1.TPnEEE 位清零)。	p. 226 <input type="checkbox"/>
			间隔定时器模式 (TPnMD2 至 TPnMD0 位 = 000)	该寄存器位只有在中断请求信号(INTTPnCC0 和 INTTPnCC1)被中断屏蔽标志 (TPnCCMK0 和 TPnCCMK1)屏蔽且同时允许了定时器输出(TOPn1)时, 才可以被置 1。此时, 请将相同的值设置到 TPnCCR0 和 TPnCCR1 寄存器中。(请参考 7.5.1 (2) (d) TPnCCR1 寄存器的操作)。	p. 228 <input type="checkbox"/>
			改写 TPnCCR0 寄存器时需要注意的事项	将 TPnCCR0 寄存器的值减小时, 要先停止计数, 再对设定值进行改变。 如果在计数过程中将 TPnCCR0 寄存器的值减小, 那么 16 位计数器可能产生溢出。	p. 233 <input type="checkbox"/>
			外部时间计数模式下的寄存器设置	当外部时钟用作计数时钟时, 外部时钟只能使用 TIPn0 引脚输入, 因为外部时钟输入已经固定为 TIPn0 引脚。此时要将 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位清零 (捕捉触发输入(TIPn0 引脚): 无边沿检测)。	p. 239 <input type="checkbox"/>
			外部事件计数模式的操作时序	在外部事件计数模式中, 不要将 TPnCCR0 寄存器设置为 0000H。	p. 241 <input type="checkbox"/>
				在外部事件计数模式中, 禁止使用定时器输出功能。如果要将外部事件计数输入作为定时器输出使用, 那么请将工作模式设置为间隔定时器模式并将外部事件计数输入设定为计数时钟(TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0 位 = 000, TPnCTL1.TPnEEE 位 = 1)。	p. 241 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第7章	软件	16 位定时器/事件计数器 P (TMP)	重写 TPnCCR0 寄存器的注意事项	要将 TPnCCR0 寄存器的值改小, 要先停止计数, 然后再改变设置值。如果在计数期间将 TPnCCR0 寄存器的值改小, 则可能会引起溢出错误。	p. 242 <input type="checkbox"/>
			TPnIOC0, TPnOE0, TPnOL0 位	在外部触发脉冲输出模式下如果未使用 TOPn0 引脚, 则需要将此位清 0。	p. 247 <input type="checkbox"/>
			运行时改变脉宽的注意事项	计数器工作时如果改变 PWM 波形, 要在最后写 TPnCCR1 寄存器。 检测到 INTTPnCC0 信号后写 TPnCCR1 寄存器, 然后改写 TPnCCRm 寄存器	p. 251 <input type="checkbox"/>
			TPnIOC0.TPnOE0, TPnOL0 位	在单次脉冲输出模式下如果未使用 TOPn0 引脚, 需要将此位清 0。	p. 259 <input type="checkbox"/>
			单次脉冲输出模式下寄存器设置	如果设置在 TPnCCR1 寄存器的值大于 TPnCCR0 寄存器的值, 即使处于单次脉冲输出模式下, 也不会输出单次脉冲。	p. 260 <input type="checkbox"/>
			改变 TPnCCRm 寄存器时需要注意的事项	将 TPnCCRm 寄存器的值减小时, 要先停止计数, 再对设定值进行改变 如果在计数过程中将 TPnCCRm 寄存器的值减小, 那么 16 位计数器可能产生溢出	p. 262 <input type="checkbox"/>
			TPnIOC0.TPnOE0, TPnOL0 位	在 PWM 输出模式下, 当不使用 TOPn0 引脚时, 需要将此位清 0。	p. 266 <input type="checkbox"/>
			选择器功能	当使用选择器功能时, 要在连接定时器之前设置 TMP 的捕捉触发输入。	p. 297 <input type="checkbox"/>
				设置选择器功能之前, 要先禁止被连接的周边 I/O 器件(TMP 或 UARTA)。	p. 297 <input type="checkbox"/>
			SELCNT0 寄存器	若将 ISEL0, ISEL3 或 ISEL4 位设置为 1, 则请将这些位所对应的引脚模式设置为捕捉输入模式。	p. 297 <input type="checkbox"/>
				请务必将第 7 到第 5 位, 第 2 位以及第 1 位设置为 0。	p. 297 <input type="checkbox"/>
			捕捉操作	若在捕捉操作时使用了一个较慢的时钟信号作为计数时钟, 当 TPnCE 位被设置为 1 后, 立刻检测到捕捉触发信号时, TPnCCR0 和 TPnCCR1 寄存器中捕捉的计数值将不会是 0000H 而是 FFFFH。	p. 298 <input type="checkbox"/>
第8章	软件	16 位定时器/事件计数器 Q (TMQ)	TQ0CTL0 寄存器	在 TQ0CE 位 = 0 时, 设置 TQ0CKS2 至 TQ0CKS0 位。当将 TQ0CE 位的值从 0 改变为 1 时, 可同时对 TQ0CKS2 至 TQ0CKS0 位进行设置。	p. 303 <input type="checkbox"/>
				请务必将第 3 到第 6 位清零。	p. 303 <input type="checkbox"/>
			TQ0CTL1 寄存器	对 TQ0EST 位的设置只有在外部触发脉冲输出模式或单脉冲输出模式才有效。在其它模式下, 对该位的置 1 操作将被忽略。	p. 304 <input type="checkbox"/>
				在外部事件计数模式下, 无论 TQ0EEE 位如何设置, 外部事件计数输入都是被选中的。	p. 304 <input type="checkbox"/>
				只有在 TQ0CTL0.TQ0CE 位 = 0 时才可以对 TQ0EEE 以及 TQ0MD2 至 TQ0MD0 位进行设置(TQ0CE 位 = 1 时, 可以对这些位写入相同的值)。如果在 TQ0CE 位 = 1 时对这些位进行了改写, 那么操作结果将不能被保证。若由于误操作引起了改写, 则要将 TQ0CE 位清零然后再次设置这些寄存器位。	p. 304 <input type="checkbox"/>
				请务必将第 3,4 和 7 位清零。	p. 304 <input type="checkbox"/>
			TQ0IOC0 寄存器	只有在 TQ0CTL0.TQ0CE 位 = 0 时才可以对 TQ0OLm 和 TQ0OEm 位进行改写(当 TQ0CE 位 = 1 时可以向这些位写入相同的值)。若由于误操作引起了改写,	p. 305 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 8 章	软件	16 位定时器/事件计数器 Q (TMQ)	TQ0IOC0 寄存器	当 TQ0CE 和 TQ0OEm 位为 0 时, 即使对 TQ0OLm 位进行了操作, TQ0Om 引脚的输出电平也不能确定。	p. 305 <input type="checkbox"/>
			TQ0IOC1 寄存器	只有在 TQ0CTL0.TQ0CE 位 = 0 时才可以对 TQ0IS7 至 TQ0IS0 进行改写 (TQ0CE 位 = 1 时只可以向这些位写入相同的值)。若由于误操作引起了改写, 则要将 TQ0CE 位清零然后再次设置这些寄存器位。	p. 306 <input type="checkbox"/>
				TQ0IS7 至 TQ0IS0 位只有在自由运行定时器模式和脉宽测量模式下才有效。其它模式下, 无法进行捕捉操作。	p. 306 <input type="checkbox"/>
			TQ0IOC2 寄存器	在 TQ0CTL0.TQ0CE 位 = 0 时才可对 TQ0EES1, TQ0EES0, TQ0ETS1 和 TQ0ETS0 位进行改写 (TQ0CE 位 = 1 时只可以向这些位写入相同的值)。若在 TQ0CE 位 = 1 由于误操作引起了改写, 则要将 TQ0CE 位清零然后再次设置这些寄存器位。	p. 307 <input type="checkbox"/>
				TQ0EES1 和 TQ0EES0 位只有在 TQ0CTL1.TQ0EEE 位 = 1 或设置了外部事件计数模式(TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 001)时才有效。	p. 307 <input type="checkbox"/>
				TQ0ETS1 和 TQ0ETS0 位只有在外部触发脉冲输出模式 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 010) 或单脉冲输出模式 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 = 011) 下才有效。	p. 307 <input type="checkbox"/>
			TQ0OPT0 寄存器	在 TQ0CTL0.TQ0CE 位 = 0 时才可对 TQ0CCS3 和 TQ0CCS0 位进行改写 (TQ0CE 位 = 1 时只可以向这些位写入相同的值)。若由于误操作引起了改写, 则要将 TQ0CE 位清零然后再次设置这些寄存器位。	p. 308 <input type="checkbox"/>
				请务必将第 1 到 3 位清零。	p. 308 <input type="checkbox"/>
			TQ0CCR0 寄存器	以下情况下禁止访问 TQ0CCR0 寄存器。详细内容请参考 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。 • 当 CPU 工作于子时钟且主时钟振荡停止时 • 当 CPU 工作于内置振荡时钟时	p. 309 <input type="checkbox"/>
			TQ0CCR1 寄存器	以下情况下禁止访问 TQ0CCR1 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。 • 当 CPU 工作于子时钟且主时钟振荡停止时 • 当 CPU 工作于内置振荡时钟时	p. 311 <input type="checkbox"/>
			TQ0CCR2 寄存器	以下情况下禁止访问 TQ0CCR2 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器 • 当 CPU 工作于子时钟且主时钟振荡停止时 • 当 CPU 工作于内置振荡时钟时	p. 313 <input type="checkbox"/>
			TQ0CCR3 寄存器	以下情况下禁止访问 TQ0CCR3 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。 • 当 CPU 工作于子时钟且主时钟振荡停止时 • 当 CPU 工作于内置振荡时钟时	p. 315 <input type="checkbox"/>
			TQ0CNT 寄存器	以下情况下禁止访问 TQ0CNT 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器 • 当 CPU 工作于子时钟且主时钟振荡停止时 • 当 CPU 工作于内置振荡时钟时	p. 317 <input type="checkbox"/>
			外部时间计数模式	使用外部事件计数模式时, 请指定为不检测 TIQ00 引脚捕捉触发输入有效沿(通过将 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位清零)。	p. 318 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第8章	软件	16 位定时器/事件计数器 Q (TMQ)	外部触发脉冲输出模式, 单次脉冲输出模式, 脉冲宽度测量模式	使用外部触发脉冲输出模式, 单脉冲输出模式和脉宽测量模式时, 请选择内部时钟作为计数时钟(通过将 TQ0CTL1.TQ0EEE 位清零)。	p. 318 <input type="checkbox"/>
			TQ0CTL1.TQ0EEE 位	该寄存器位只有在中断请求信号(INTTQ0CC0 和 INTTQ0CCk)被中断屏蔽标志(TQ0CCMK0 至 TQ0CCMKk)屏蔽且同时允许了定时器输出(TQ0Qk)时, 才可以被置 1。此时, 请将相同的值设置到 TQ0CCR0 和 TQ0CCRk 寄存器中。(请参考 8.5.1 (2) (d) TQ0CCR1 至 TQ0CCR3 寄存器的操作) (k = 1 至 3)。	p. 320 <input type="checkbox"/>
			改写 TQ0CCR0 寄存器时需要注意的事项	要将 TQ0CCR0 寄存器的值改小, 首先停止计数然后再更改设置值。 如果在计数时向 TQ0CCR0 寄存器写入更小的值, 16 位计数器可能会溢出	pp. 324, 333 <input type="checkbox"/>
			外部事件计数模式操作的寄存器设置	当使用一个外部时钟用作计数时钟的时候, 外部时钟只能从 TIQ00 脚输入。同时, 将 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 置于 00【捕捉触发器输入 (TIQ00 脚): 未检测到边缘】	p. 330 <input type="checkbox"/>
			外部事件计数模式的操作程序	在外部事件计数模式中, 不要将 TQ0CCR0 寄存器设置为 0000H	p. 332 <input type="checkbox"/>
				在外部事件计数模式中, 禁止使用定时器输出功能。如果要外部事件计数输入作为定时器输出使用, 那么请将工作模式设置为间隔定时器模式并将外部事件计数输入设定为计数时钟(TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 000, TQ0CTL1.TQ0EEE 位 = 1)。	p. 332 <input type="checkbox"/>
			TQ0IOC0.TQ0OE0, TQ0OL0 位	当 TOQ00 引脚不用于外部触发脉冲输出模式时将此位清零。	p. 340 <input type="checkbox"/>
			操作过程中改变脉冲宽度时需要注意的事项	计数器计数过程中, 若改变 PWM 波形, 则要在最后进行对 TQ0CCR1 寄存器的写入操作 对 TQ0CCR1 寄存器进行写入操作之后, 若需再次更改 TQ0CCRk 寄存器的值, 须等待下一个 INTTQ0CC0 信号被检测到后, 再进行相关操作。	p. 344 <input type="checkbox"/>
			TQ0IOC0.TQ0OE0, TQ0OL0 位	当单脉冲输出模式下不使用 TOQ00 引脚时, 要将该位清零。	p. 353 <input type="checkbox"/>
			单脉冲输出模式的寄存器设置	如果 TQ0CCRk 触发器设置值大于 TQ0CCR0 寄存器的设置值, 在单脉冲输出模式下无法输出单脉冲。	p. 354 <input type="checkbox"/>
			改变 TQ0CCRm 寄存器时需要注意的事项	将 TQ0CCRm 寄存器的值减小时, 要先停止计数, 再对设定值进行改变。 如果在计数过程中将 TQ0CCR0 寄存器的值减小, 那么 16 位计数器可能产生溢出。	p. 357 <input type="checkbox"/>
			TQ0IC0.TQ0OE0, TQ0OL0 位	当 PWM 输出模式下不使用 TOQ00 引脚时, 要将该位清零。	p. 362 <input type="checkbox"/>
			捕捉操作	若在捕捉操作时使用了一个较慢的时钟信号作为计数时钟, 当 TQ0CE 位被设置为 1 后, 立刻检测到捕捉触发信号时 TQ0CCR0, TQ0CCR1, TQ0CCR2, 和 TQ0CCR3 寄存器中捕捉的计数值将不会是 0000H 而是 FFFFH。	p. 397 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第9章	软件	16位间隔定时器 M (TMM)	TM0CTL0 寄存器	在 TM0CE 位 = 0 时设置 TM0CKS2 到 TM0CKS0 位。 当将 TM0CE 的值从 0 改变为 1 时, 不可以同时设置 TM0CKS2 到 TM0CKS0 的值。	p. 400 <input type="checkbox"/>
				请务必将第 3 位到第 6 位清零。	p. 400 <input type="checkbox"/>
			间隔定时器下的操作	不可以将 TM0CMP0 寄存器设置为 FFFFH。	pp. 401, 404 <input type="checkbox"/>
			计数开始	从 TM0CTL0.TM0CE 位被置 1 到 16 位计数器开始计数的时间间隔, 根据选择的计数时钟的不同	p. 405 <input type="checkbox"/>
			TM0CMP0, TM0CTL0 寄存器	TMM0 工作时, 禁止改写 TM0CMP0 和 TM0CTL0 寄存器的值。 如果在 TM0CE 位 = 1 时对这两个寄存器进行改写, 那么操作结果将不能被保证。 一旦发生上述情况, 请先将 TM0CTL0.TM0CE 位清零, 之后重置这些寄存器。	p. 405 <input type="checkbox"/>
第10章	软件	钟表定时器功能	PRSM0 寄存器	钟表定时器操作期间不要改变的 BGCS00 和 BGCS01 位取值。	p. 409 <input type="checkbox"/>
				在设置 BGCE0 为 1 前, 设置寄存器 PRSM0。	p. 409 <input type="checkbox"/>
				根据主时钟频率, 设置寄存器 PRSM0 和 PRSCM0 以获得 32.768 kHz 的 fBRG 频率。	p. 409 <input type="checkbox"/>
		PRSCM0 寄存器		钟表定时器操作期间不要对寄存器 PRSCM0 进行写入操作	p. 410 <input type="checkbox"/>
				在设置 PRSM0.BGCE0 为 1 前, 设置寄存器 PRCM0。	p. 410 <input type="checkbox"/>
				根据主时钟频率, 设置寄存器 PRSM0 和 PRSCM0 以获得 32.768 kHz 的 fBRG 频率	p. 410 <input type="checkbox"/>
		WTM 寄存器		当 WTM0 和 WTM1 均为 0 时才可对 WTM2 ~ WTM7 位进行写入操作。	p. 412 <input type="checkbox"/>
	硬件	看门狗定时器 2 的功能	看门狗定时器默认启动	在允许操作 (WTM.WTM1 和 WTM.WTM0 = 1)后, 第一次钟表定时器中断请求信号(INTWT)产生前需要一段准备时间。	p. 415 <input type="checkbox"/>
				经过 0.515625 秒(max.)产生第 1 个 INTWT($2^9 \times 1/32768 = 0.015625 \text{ s(max.)}$)。随后 INTWT 信号每 0.5 秒产生 1 次。	p. 415 <input type="checkbox"/>
				由不可屏蔽中断请求信号(INTWDT2)引发的不可屏蔽中断服务, 可参见 19.2.2 (2) 信号 INTWDT2	p. 416 <input type="checkbox"/>
第11章	软件	看门狗定时器 2 的功能	WDTM2 寄存器	复位释放后, 看门狗定时器 2 会自动启动 当不使用看门狗定时器 2, 在复位前通过该功能使其停止工作, 或清除看门狗定时器 2 并在下一次间隔开始前使其停止工作。 为了确认操作的正确性, 即使在不需要改变默认设置(复位模式, 间隔时间: $f_R/2^{19}$)的情况下, 也要对寄存器 WDTM2 执行一次写操作。	p. 416 <input type="checkbox"/>
				下述状态下, 禁止访问寄存器 WDTM2。详细情况, 参见 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。 • 当 CPU 使用子时钟操作, 主时钟振荡停止。 • 当 CPU 使用内置振荡时钟进行操作	p. 418 <input type="checkbox"/>
				WDCS20 ~ WDCS24 的详细情况, 参见表 11-2 看门狗定时器 2 的时钟选择	p. 418 <input type="checkbox"/>
				尽管停止内置振荡器的操作可以停止看门狗定时器 2, 但是, 还要使寄存器 WDTM2 清零以确保停止定时器(以避免由于误操作选中主时钟或子时钟)。	p. 418 <input type="checkbox"/>

(13/36)

章节	分类	功能	具体功能	注意事项	页码
第 11 章	软件	看门狗定时器 2 的功能	WDTM2 寄存器	如果寄存器 WDTM2 在复位后被复写两次或更多次, 那么溢出信号产生, 计数器复位。	p. 418 <input type="checkbox"/>
				如需有意产生一个溢出信号, 请向寄存器 WDTE 写入“ACH”以外的值一次, 或向寄存器 WDTM2 写入数据两次。然而, 当看门狗定时器 2 设为停止工作状态时, 即使向寄存器 WDTE 写入“ACH”以外的值一次, 或向寄存器 WDTM2 写入数据两次也不会产生溢出信号。	p. 418 <input type="checkbox"/>
				要停止看门狗定时器 2 的操作, 设置 RCM.RSTP 为 1(以停止内置晶振)并向寄存器 WDTM 写入 00H。如果 RCM.RSTP 不能被置 1, 设置 WDSCS23 为 1(选中 2 ⁰ /fxx, 时钟在 IDLE1, IDLW2, sub-IDLE, 和子时钟模式下可停止工作)。	p. 418 <input type="checkbox"/>
			WDTE 寄存器	如果“ACH”以外的数值被写入寄存器 WDTE, 必定产生溢出信号	p. 420 <input type="checkbox"/>
				当向寄存器 WDTE 执行 1 位存储器操作指令, 必定产生溢出信号。	p. 420 <input type="checkbox"/>
				如需有意产生一个溢出信号, 请向寄存器 WDTE 写入“ACH”以外的值一次, 或向寄存器 WDTM2 写入数据两次。 然而, 当看门狗定时器 2 设为停止工作状态时, 即使向寄存器 WDTE 写入“ACH”以外的值一次, 或向寄存器 WDTM2 写入数据两次也不会产生溢出信号。	p. 420 <input type="checkbox"/>
				读取寄存器 WDTE 的值为“9AH”(与写入值“ACH”不同)。	p. 420 <input type="checkbox"/>
第 12 章	软件	实时输出功能 (RTO)	RTBL0, RTBH0 寄存器	当向寄存器 RTBH0 的第 6、7 位进行写操作时, 写入的数据恒为 0	p. 423 <input type="checkbox"/>
				下述状态下禁止访问寄存器 RTBL0 和 RTBH0。详细情况, 参见 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。 • 当 CPU 使用子时钟进行操作, 主时钟晶振停止振荡时。 • 当 CPU 使用内置振荡时钟进行操作时。	p. 423 <input type="checkbox"/>
				设置实时输出端口后, 要在产生实时输出触发之前, 将欲输出的数据设置到寄存器 RTBL0 和 RTBH0。	p. 423 <input type="checkbox"/>
			RTPM0 寄存器	通过允许实时输出操作(RTPC0.RTPOE0 = 1), RTP00 ~ RTP05 中设置为实时输出的位将进行实时输出操作, 而那些设置为普通端口模式的位会输出 0。	p. 424 <input type="checkbox"/>
				禁止实时输出 (RTPC0.RTPOE0 = 0)时, 无论 RTPM0 寄存器如何设置, 实时输出引脚 RTP00 ~ RTP05 均输出 0。	p. 424 <input type="checkbox"/>
				为了将 RTP00 ~ RTP05 引脚作为实时输出引脚使用, 需要通过 PMC 和 PFC 寄存器将这些引脚的模式设置为实时输出端口模式。	p. 424 <input type="checkbox"/>
			RTPC0 寄存器	只有当 RTPOE0 = 0 时, 设置 RTPEG0, BYTE0 和 EXTR0 位	p. 425 <input type="checkbox"/>
			注意事项	避免下述软件冲突。 • 实时输出允许/禁止转换(RTPOE0)和所选择的实时输出触发之间的冲突。 • 实时输出允许状态下写寄存器 RTBH0 和 RTBL0, 与所选择的实时输出触发之间的冲突。	p. 427 <input type="checkbox"/>
				在操作初始化前, 停止实时输出(RTPOE0 = 0)。	p. 427 <input type="checkbox"/>
				一旦实时输出被禁止(RTPOE0 = 0), 确保在再次允许实时输出(RTPOE0 = 0 1)前初始化寄存器 RTBH0 和 RTBL0。	p. 427 <input type="checkbox"/>

(14/36)

章节	分类	功能	具体功能	注意事项	页码
第 13 章	硬件	A/D 转换器	ANI0 - ANI15 引脚	确保输入 ANI0 ~ ANI11 的电压不超过额定值。 特别是，如果大于或等于 AV_{REF0} 的电压输入某个通道，这个通道的转换结果将不确定，其它通道的转换结果也将受到影响。	p. 431
			ADA0M0 寄存器	下述情况下禁止访问寄存器 ADA0M0。详细情况，参见 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。 <ul style="list-style-type: none">当 CPU 使用子时钟，且主时钟晶振停止振荡。当 CPU 使用内置晶振时钟。	p. 432
	0 位的写操作将被忽略。			p. 433	
	当 A/D 转换使能(ADA0CE = 1)后，禁止改变 ADA0M1.ADA0FR2 ~ ADA0M1.ADA0FR0 位。			p. 433	
	在以下模式下，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，然后再允许 A/D 转换 (ADA0CE 位 = 1)。 <ul style="list-style-type: none">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。 如果寄存器 ADA0M0, ADA0M2, ADA0S, ADA0PFM, 和 ADA0PFT 在 A/D 转换过程中(ADA0EF = 1)进行写操作，将根据模式进行下述操作 <ul style="list-style-type: none">软件触发模式 A/D 转换停止并从头开始。硬件触发模式 A/D 转换停止并进入触发等待状态			p. 433	
	要选择外部触发模式/定时器触发模式(ADA0TMD 位 = 1)，需要设置为高速转换模式(ADA0M1.ADA0HS1 位= 1)。允许 A/D 转换操作后不要在稳定时间内输入触发。(ADA0CE 位= 1)。			p. 433	
	当不使用 A/D 转换器时，设置 ADA0CE 位 0 停止操作以减少功耗。			p. 433	
	ADA0M1 寄存器			A/D 转换使能(ADA0M0.ADA0CE = 1)后，禁止改变 ADA0M1 的值。	p. 434
			设置高速转换模式 (ADA0HS1 位 = 1)后，选择外部触发模式/定时器触发模式 (ADA0M0.ADA0TMD bit = 1), 允许 A/D 转换操作后不要在稳定时间内输入触发。(ADA0CE bit = 1)	p. 434	
			确保第 6 ~ 3 位清零。	p. 434	
	软件		正常转换模式下转换时间的选择 (ADA0HS1 位= 0)	设置 $2.6\ \mu s \leq \text{转换时间} \leq 10.4\ \mu s$ 。	p. 435
				A/D 转换期间，如果对寄存器 ADA0M0, ADA0M2, ADA0S, ADA0PFM, 和 ADA0PFT 进行写操作或输入触发信号，再次转换将被执行。然而，如果稳定时间结束时序与向这些寄存器的写操作或者与输入触发信号冲突，64 个时钟周期的稳定时间再次被插入。 如果再次插入的稳定时间结束时序再次产生冲突，稳定时间再次被插入。因此不要设置触发输入间隔和控制寄存器写间隔小于等于 64 个时钟周期。	p. 435

章节	分类	功能	具体功能	注意事项	页码
第 13 章	软件	A/D 转换器	高速转换模式下转换时间的选择 (ADA0HS1 位 = 1)	设置 $2.6 \mu s \leq \text{转换时间} \leq 10.4 \mu s$	p. 436 <input type="checkbox"/>
				高速转换模式下，禁止在稳定时间内复写寄存器 ADA0M0, ADA0M2, ADA0S, ADA0PFM, ADA0PFT 和输入触发信号。	p. 436 <input type="checkbox"/>
			ADA0M2 寄存器	在以下模式下，当 A/D 转换停止(ADA0CE 位 = 0)时将数据写入 ADA0M2 寄存器，然后再允许 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。	p. 437 <input type="checkbox"/>
				确保第 7 ~ 2 位清零。	p. 437 <input type="checkbox"/>
			ADA0S 寄存器	在以下模式下，当 A/D 转换停止(ADA0M0.ADA0CE 位 = 0)时将数据写入 ADA0S 寄存器，然后再允许 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	p. 438 <input type="checkbox"/>
				确保第 7 ~ 4 位清零。	p. 438 <input type="checkbox"/>
			ADA0CRn, ADA0CRnH 寄存器	下述状态禁止访问寄存器 ADA0CRn 和 ADA0CRnH。详细情况，参见 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。 • 当 CPU 使用子时钟，且主时钟晶振停止振荡。 • 当 CPU 使用内部晶振时钟。	p. 439 <input type="checkbox"/>
				寄存器 ADA0M0 和 ADA0S 的写操作可能引起 ADA0CRn 内容的不确定。转换后，在写寄存器 ADA0M0 和 ADA0S 前，读取转换结果。如果不按上述操作进行，正确转换结果可能不能读取。	p. 439 <input type="checkbox"/>
			ADA0PFM 寄存器	选择模式下，寄存器 ADA0PFT 所设置的 8 位数据与寄存器 ADA0CRnH 的值进行比较，ADA0CRnH 由 ADA0S 进行设置。如果结果与 ADA0PFC 位所设置的条件匹配，转换结果存储于寄存器 ADA0CRn 并产生中断信号 INTAD。如果不匹配，不产生中断信号 INTAD。	p. 441 <input type="checkbox"/>
				扫描模式下，寄存器 ADA0PFT 所设置的 8 位数据与寄存器 ADA0CRnH 的值进行比较。如果结果与 ADA0PFC 位所设置的条件匹配，转换结果存储于寄存器 ADA0CRn 并产生中断信号 INTAD。如果不匹配，不产生中断信号 INTAD。无论比较结果如何，扫描模式继续工作，转换结果存储于寄存器 ADA0CRn 中，直到扫描操作完成。但是，扫描操作完成后不产生中断信号 INTAD。	p. 441 <input type="checkbox"/>
				在以下模式下，当 A/D 转换停止(ADA0M0.ADA0CE 位 = 0)时将数据写入 ADA0PFM 寄存器，然后再允许 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	p. 441 <input type="checkbox"/>
			ADA0PFT 寄存器	在以下模式下，当 A/D 转换停止(ADA0M0.ADA0CE 位 = 0)时将数据写入 ADA0PFT 寄存器，然后再允许 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	p. 442 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第13章	软件	A/D 转换器	外部触发模式	为了选择外部触发模式，首先设置高速转换模式。在 A/D 使能转换操作 (ADA0M0.ADA0CE 位 = 1) 后的稳定期间内，不要立即输入触发。	p. 445 <input type="checkbox"/>
			定时器触发模式	为了选择外部触发模式，首先设置高速转换模式。在 A/D 使能转换操作 (ADA0M0.ADA0CE 位 = 1) 后的稳定期间内，不要立即输入触发。	p. 446 <input type="checkbox"/>
			不使用 A/D 转换器	当不使用 A/D 转换器时，设置 ADA0M0.ADA0CE = 0，使功耗减少。	p. 456 <input type="checkbox"/>
			引脚 ANI0 ~ ANI11 的输入范围	输入指定范围的电压值到引脚 ANI0 ~ ANI11。如果大于等于 AVREF0 或小于等于 AVss (即使在最大绝对值范围内) 的电压被输入到这些引脚，则那个通道的转换值不确定，其它通道的转换值也会受到影响。	p. 456 <input type="checkbox"/>
			抑制噪声的方法	为了确保 10 位分辨率，引脚 ANI0 ~ ANI11 必须有效抑制噪声。噪声影响随着模拟输入源的输出阻抗的增加而提高。为了降低噪声，推荐连接一个外部电容，如图 13-12 所示	p. 456 <input type="checkbox"/>
			I/O 复用	模拟输入引脚 (ANI0 ~ ANI11) 可功能复用为普通端口引脚。当从引脚 ANI0 ~ ANI11 中选择一个执行 A/D 转换时，不要在转换期间执行读/写端口指令，否则转换分辨率将下降。 在转换期间设置引脚为输出端口同样会使转换分辨率将下降。其原因是连接端口引脚的外部电路导致输出电流波动。 如果某引脚正在进行 A/D 转换，其临近引脚输入一个数字脉冲，则由于耦合噪声的影响，A/D 转换值可能不准确。因此，在 A/D 转换过程中，确保不使用临近引脚传输脉冲信号。	p. 456 <input type="checkbox"/>
	硬件	AVREF0 引脚	中断请求标志 (ADIF)	即使寄存器 ADA0S 的内容改变，中断请求标志 (ADIF) 也不清零。因此，如果在 A/D 转换过程中模拟输入引脚改变，上次所选择的模拟输入信号的转换结果可能被存储，转换结束中断请求标志可能在寄存器 ADA0S 复写前立刻置 1。如果寄存器 ADA0S 复写后立即读取标志 ADIF，标志 ADIF 可能置 1，即使新选择的模拟输入信号的转换还未完成。当 A/D 转换停止，在再次转换前使标志 ADIF 清零。	p. 457 <input type="checkbox"/>
			AVREF0 引脚	(a) 引脚 AVREF0 用于 A/D 转换器的供电电压引脚，同样为复用功能端口供电。 如图 13-15 所示，当用于备用供电电压时，确保向引脚 AVREF0 提供与 VDD 相同的电压。 (b) 引脚 AVREF0 也用于 A/D 转换器的参考电压引脚。如果提供给引脚 AVREF0 的源供电电源具有高阻抗，或者如果供电电源低电流负载能力低，参考电压可能被转换中的电流影响(特别是，转换操作使能位 ADA0CE 刚刚置 1 之后)。结果，转换精度会下降。如图 13-15 所示，为了避免上述情况，推荐连接一个电容在引脚 AVREF0 和 AVSS 之间以抑制参考电压的波动。 (c) 如果提供给引脚 AVREF0 的源供电电源具有高直流阻抗(例如，由于插入一个二极管)，当转换使能时的电压可能比转换停止时的电压低，因为 A/D 转换电流引起电压失真。	p. 458 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 13 章	软件	A/D 转换器	读取 ADA0CRn 寄存器	当执行寄存器 ADA0M0 ~ ADA0M2 , ADA0S, ADA0PFM 或 ADA0PFT 的写指令, 寄存器的 ADA0CRn 内容可能不确定。转换完成后, 并在写寄存器 ADA0M0 ~ ADA0M2, ADA0S, ADA0PFM 或 ADA0PFT 前, 读取转换结果。同样, 当响应一个外部/定时器触发时, ADA0CRn 寄存器的内容可能也不确定。在转换结束后并在响应下次的外部/定时器触发之前读取转换结果。在与上述时序不同的时, 正确转换结果可能无法读取	p. 458
			等待模式	因为 A/D 转换器在 STOP 模式下停止操作, 转换结果无效, 所以功耗可以降低。释放 STOP 模式后, 操作恢复, 但 STOP 模式释放后的 A/D 转换结果无效。当 STOP 模式释放后使用 A/D 转换器, 应在设置 STOP 模式前或释放 STOP 模式后, 使 ADA0M0.ADA0CE = 0, 然后在释放 STOP 模式后设置 ADA0CE = 1。 在 IDLE1, IDLE2 或子时钟操作模式下, 继续运行。因此, 为了减小功耗, 应使 ADA0M0.ADA0CE = 0。在 IDLE1, IDLE2 模式下, 由于模拟输入电压值不能保留, IDLE1, IDLE2 模式释放后的 A/D 转换结果无效。 进入 IDLE1, IDLE2 模式前的转换结果有效	p. 459
			高速转换模式	高速转换模式下, 稳定时间内, 禁止复写寄存器 ADA0M0, ADA0M2, ADA0S, ADA0PFM, 和 ADA0PFT 并触发输入。	p. 459
			A/D 转换时间	A/D 转换时间包括稳定时间、转换时间、等待时间和触发响应时间(详细情况, 参见表 13-2 正常转换模式下转换时间的选择 (ADA0HS1 = 0) 和表 13-3 高速转换模式下转换时间的选择 (ADA0HS1 = 1))。 正常转换模式下, A/D 转换期间, 如果对寄存器 ADA0M0, ADA0M2, ADA0S, ADA0PFM, 和 ADA0PFT 进行写操作或者输入触发, 开始重新转换。然而, 如果稳定时间结束时时序与写上述寄存器冲突, 或者如果稳定时间结束时时序与输入触发冲突, 会多插入 64 个时钟的稳定时间。 如果在多插入 64 个时钟的稳定时间结束时, 再次发生冲突, 稳定时间再次插入。因此, 不要设置触发输入间隔和控制写寄存器小于等于 64 个时钟。	p. 459
			VA/D 转换结果的漂移	由于供电电压的波动, A/D 转换结果可能产生变化, 或者可能受到噪声影响。为了减小漂移, 使用多次测量求平均值的方法	p. 459
	硬件	A/D 转换结果滞后特性	逐次逼近 A/D 转换器维持内部采样和保持电容的模拟输入电压, 同时进行 A/D 转换。A/D 转换结束后模拟输入电压保留在内部采样和保持电容中。结果, 产生下述现象。 <ul style="list-style-type: none">当同一个通道用于 A/D 转换, 如果电压高于或低于先前 A/D 转换, 则出现滞后特征, 转换结果受到上次转换值影响。因此, 即使同一个模拟输入电压也会产生不同结果。当开关模拟输入通道, 可能出现滞后特征, 转换结果受到上次转换通道影响。这是因为有一个 A/D 转换器用于 A/D 转换。因此, 即使同一个模拟输入电压也会产生不同结果。	p. 459	

(18/36)

章节	分类	功能	具体功能	注意事项	页码
第 14 章	硬件	D/A 转换器	D/A 转换器	DAC0 和 DAC1 共用 AV _{REF1} 引脚。	p. 464
			D/A 转换器	DAC0 和 DAC1 共用 AV _{ss} 引脚。A/D 转换器也共用 AV _{ss} 引脚。.	p. 464
			DA0M 寄存器	在实时输出模式下(DA0MDn 位= 1)输出触发如下： • 当 n = 0 时：INTTP2CC0 信号(参看第 7 章 16 位定时/事件计数器 P (TMP)) • 当 n = 1 时：INTTP3CC0 信号(参看第 7 章 16 位定时/事件计数器 P (TMP))	p. 645
	软件	D/A 转换器	DA0CS0, DA0CS1 寄存器	在实时输出模式下 (DA0M.DA0MDn 位= 1)，在 INTTP2CC0/INTTP3CC0 信号产生前设置 DA0CSn 寄存器。当产生 INTTP2CC0/INTTP3CC0 信号时 D/A 转换开始	p. 466
			注意事项	实时输出模式下，当响应触发信号后，不要改变寄存器 DA0CSn 的值	p. 468
			注意事项	在改变操作模式前，确保 DA0M.DA0CEn = 0。	p. 468
			注意事项	当引脚 P10/AN00 和 P11/AN01 其中之一被用于 I/O 端口，而另一个用于 D/A 输出引脚，在 D/A 输出期间不要改变 I/O 端口电平。	p. 468
			注意事项	确保 AVREF0 = VDD = AVREF1 = 3.0 to 3.6 V。如果超出次范围，操作不能保证。	p. 468
			注意事项	AVREF1 的电压与 AVREF0 的电压一致。	p. 468
			注意事项	没有电流可以从引脚 ANOn (n = 0, 1)流出，因为 D/A 转换器的输出阻抗很高。当连接一个小于等于 2 M Ω 的电阻时，在电阻和引脚 ANOn 之间插入一个 JFET 输入运算放大器。	p. 468
	硬件	D/A 转换器	注意事项	由于 STOP 模式下，D/A 转换器停止操作，引脚 ANO0 和 ANO1 引入高阻抗状态，功耗可降低。 IDLE1, IDLE2,或时钟模式下，操作继续。因此，为了降低功耗使 DA0M.DA0CEn = 0。	p. 468
第 15 章	软件	异步串行接口 A (UARTA)	CSIB4 和 UARTA0 模式转换	如果 CSIB4 和 UARTA0 的发送/接收操作过程中，进行转换，这两个功能不能保证。确保不被使用的那个处于非使能态。	p. 469
			UARTA2 和 I ² C00 模式转换	如果 UARTA2 和 I ² C00 的发送/接收操作过程中，进行转换，这两个功能不能保证。确保不被使用的那个处于非使能态。	p. 470
			UARTA1 和 I ² C02 模式转换	如果 UARTA1 和 I ² C02 的发送/接收操作过程中，进行转换，这两个功能不能保证。确保不被使用的那个处于非使能态。	p. 471
			UAnOPT0 寄存器	在 SBF 接收期间(UAnSRF = 1)，不要将 UAnSRT 和 UAnSTT 置 1。	p. 477
			SBF 接收	在数据接收期间如果发送 SBF，则会产生帧错误。	p. 487
				在 SBF 接收期间(UAnSRF = 1)，不要将 UAnSRT 和 UAnSTT 置 1。	p. 487
			连续发送	在连续发送过程中执行发送初始化，确保 UAnSTR.UAnTSF = 0，然后进行初始化。当 UAnTSF = 1 时初始化数据，其发送过程不能保证	p. 489
			UART 接收	即使产生接收错误，也要确保读取寄存器 UAnRX。如果寄存器 UAnRX 不被读取，当接收到下一个数据时会产生溢出错误，而且接收错误状态保持不变。	p. 491

章节	分类	功能	具体功能	注意事项	页码
第 15 章	软件	异步串行接口 A (UARTA)	UART 接收	接收始终按“停止位的个数 = 1”的情况执行，第 2 个停止位被忽略。	p. 491 <input type="checkbox"/>
				在接收完成，产生接收完成中断请求信号后，读取寄存器 UAnRX，使 UAnPWR 或 UAnRXE 清零。如果 UAnPWR 或 UAnRXE 在信号 INTUAnR 产生前被清零，寄存器 UAnRX 的读取值不被保证。	p. 491 <input type="checkbox"/>
				如果 UARTAn 的接收完成处理过程(产生信号 INTUAnR)和 UAnPWR = 0 或 UAnRXE = 0 产生冲突，无论数据是否被存储到寄存器 UAnRX 中都产生信号 INTUAnR。 为了在不等待 INTUAnR 信号产生的情况下完成接收，设置中断控制寄存器 (UAnRIC)的中断屏蔽标志(UAnRMK)为 1，且 UAnPWR = 0 或 UAnRXE = 0，确保寄存器 UAnRIC 的中断请求标志(UAnRIF)清零。	p. 491 <input type="checkbox"/>
			接收错误	当产生信号 INTUAnR，寄存器 UAnSTR 必须被读取以检测错误。	p. 492 <input type="checkbox"/>
				连续接收过程中产生接收错误中断，在下次接收完成前必须读取寄存器中的内容，并执行错误处理	p. 493 <input type="checkbox"/>
			校验类型和操作	当使用 LIN 功能，固定寄存器 UAnCTL0 的 UAnPS1 和 UAnPS0 位为 00	p. 494 <input type="checkbox"/>
			UAnCTL1 寄存器	在复写寄存器 UAnCTL1 前，UAnCTL0.UAnPWR 位清零	p. 497 <input type="checkbox"/>
			UAnCTL2 寄存器	在复写寄存器 UAnCTL2 前，UAnCTL0.UAnPWR 位或 UAnTXE 和 UAnRXE 位清零。	p. 498 <input type="checkbox"/>
			波特率错误	发送期间必须保持波特率误差在接收端允许的误差范围内。	p. 499 <input type="checkbox"/>
				接收期间，波特率误差必须满足“(5) 接收期间允许的波特率范围”中所描述的范围	p. 499 <input type="checkbox"/>
			接收期间允许的波特率范围	必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。	p. 501 <input type="checkbox"/>
			当 UARTAn 所使用的时钟停止工作	当 UARTAn 所使用的时钟停止工作(例如，进入 IDLE1, IDLE2, 或 STOP 模式)，操作停止，每个寄存器保持时钟停止前的值。引脚 TXDAn 的输出继续保持时钟停止前的值。然而，时钟恢复后，操作不被保证。因此，时钟恢复后，通过设置 UAnCTL0.UAnPWR, UAnCTL0.UAnRXEn, 和 UAnCTL0.UAnTXEn 为 000，使电路初始化。	p. 504 <input type="checkbox"/>
			RXDA1 引脚 KR7 引脚	引脚 RXDA1 和 KR7 不能同时使用。使用引脚 RXDA1 时，不要使用引脚 KR7。使用引脚 KR7 时，不要使用引脚 RXDA1(推荐设置 PFC91 为 1 且 PFCE91 为 0)。	p. 504 <input type="checkbox"/>
			使用 DMA 传送接收和发送的数据	UARTAn 下，不产生由于通信错误引起的中断。当使用 DMA 传送接收和发送的数据，即使传送期间出现错误(校验，溢出，帧错误)，也不进行处理。可以在 DMA 传送后读取寄存器 UAnSTR 查看是否出错，或着在通信过程中读取寄存器 UAnSTR 查看是否出错	p. 504 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 15 章	软件	异步串行接口 A (UART A)	启动 UARTAn	按照下述次序启动 UARTAn。 <1>设置 UAnCTL0.UAnPWR 为 1 <2>设置端口。 <3>设置 UAnCTL0.UAnTXE 为 1, UAnCTL0.UAnRXE 为 1。	p. 504 <input type="checkbox"/>
			停止 UARTAn	按照下述次序停止 UARTAn <1> 设置 UAnCTL0.UAnTXE 为 0, UAnCTL0.UAnRXE 为 0。 <2> 设置端口, 设置 UAnCTL0.UAnPWR 为 0 (如果端口设置不改变则不需要修改)。	p. 504 <input type="checkbox"/>
			发送模式	发送模式下(UAnCTL0.UAnPWR = 1 和 UAnCTL0.UAnTXE = 1), 不要写相同的值到寄存器 UAnTX, 因为写寄存器后传输就开始了。连续发送相同的值时, 写相同的值到寄存器 UAnTX。	p. 504 <input type="checkbox"/>
			连续发送模式	连续发送模式下, 从停止位到下一个起始位的通信速率比正常情况下的 2 个基本时钟要长。然而, 接收端通过检测起始位初始化时序, 所以结果不受影响。	p. 504 <input type="checkbox"/>
第 16 章	软件	3 线长度可变串行 I/O (CSIB)	CSIB4 和 UARTA0 模式转换	如果 CSIB4 和 UARTA0 的发送/接收操作过程中, 进行转换, 这两个功能不能保证。确保不被使用的那个处于非使能态。	p. 505 <input type="checkbox"/>
			CSIB0 和 I ² C01 模式转换	如果 CSIB0 和 I ² C01 的发送/接收操作过程中, 进行转换, 这两个功能不能保证。确保不被使用的那个处于非使能态。	p. 506 <input type="checkbox"/>
			CBnCTL0 寄存器	为了强行停止发送/接收, 清除 CBnPWR 位, 而不是 CBnRxE 和 CBnTXE 位。同时, 时钟输出停止。	p. 510 <input type="checkbox"/>
				确保 2 和 3 位为 0。	p. 512 <input type="checkbox"/>
			CBnCTL1 寄存器	只有当 CBnCTL0.CBnPWR = 0 时, 寄存器 CBnCTL1 可被复写。	p. 513 <input type="checkbox"/>
				将通讯时钟(f _{CCLK})设置为 8 MHz 或更低。	p. 513 <input type="checkbox"/>
			CBnCTL2 寄存器	只有当 CBnCTL0.CBnPWR = 0 或者 CBnTXE 和 CBnRXE 都为 0 时, 寄存器 CBnCTL2 可被复写。	p. 514 <input type="checkbox"/>
			连续发送模式 (主设备模式, 发送模式)	在连续发送模式中, 不产生接收完成中断请求信号 (INTCBnR)。	p. 531 <input type="checkbox"/>
			连续发送模式 (从设备模式, 发送模式)	在连续发送模式中, 不产生接收完成中断请求信号 (INTCBnR)。	p. 540 <input type="checkbox"/>
			时钟时序	I 单独发送模式下, 通信过程中(CBnTSF = 1)忽略写入的下一个发送数据, 下一次通信不开始。使用连续发送模式而不用单独发送模式。推荐在使用 DMA 时使用连续发送模式	p. 549 <input type="checkbox"/>
			PRSM1 至 PRSM3 寄存器	操作过程中, 不要复写寄存器 PRSMm。	p. 552 <input type="checkbox"/>
				在设置 BGCEm = 1 前, 设置寄存器 PRSMm。	p. 552 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 16 章	软件	3 线长度可变串行 I/O (CSIB)	PRSCM1 至 PRSCM3 寄存器	操作过程中，不要复写寄存器 PRSCMm	p. 553 <input type="checkbox"/>
				在设置 BGCEm = 1 前，设置寄存器 PRCMm。	p. 553 <input type="checkbox"/>
			波特率	将 f _{BRGm} 设置为 8 MHz 或更低。	p. 553 <input type="checkbox"/>
			使用 DMA 发送接收数据时	当使用 DMA 传送接收和发送的数据，即使串行传送期间出现溢出错误，也不进行处理。可以在 DMA 传送后读取寄存器 CBnSTR.CBnOVE，查看是否出现溢出错误。	p. 554 <input type="checkbox"/>
			CBnCTL0 寄存器 CBnCTL1 寄存器 CBnCTL2 寄存器	考虑到寄存器禁止在操作期间(CBnCTL0.CBnPWR 为 1)复写，如果操作期间由于错误操作而复写寄存器，设置 CBnCTL0.CBnPWR 为 0，然后初始化 CSIBn。 禁止在操作期间复写的寄存器如下。 <ul style="list-style-type: none"> • CBnCTL0 寄存器: CBnTXE, CBnRXE, CBnDIR, CBnTMS 位 • CBnCTL1 寄存器: CBnCKP, CBnDAP, CBnCKS2 至 CBnCKS0 位 • CBnCTL2 寄存器: CBnCL3 至 CBnCL0 位 	p. 554 <input type="checkbox"/>
			通信类型 2 或 4	通信类型 2 或 4 (CBnCTL1.CBnDAP = 1)下，产生接收完成中断(INTCBnR)后，SCKBn 时钟执行到一半时 CBnSTR.CBnTSF 被清零。 单独发送模式下，通信过程中(CBnTSF = 1)忽略写入的下一个发送数据，下一次通信不开始。同样，在只能接收的通信中(CBnCTL0.CBnTXE = 0, CBnCTL0.CBnRXE = 1)，如果接收到的数据在通信过程中(CBnTSF = 1)被读取，则下一次通信不开始。 因此，在通信模式 2 或 4(CBnDAP = 1)下的单独发送模式，特别要注意下述问题。 <ul style="list-style-type: none"> • 要开始下一次发送，确保 CBnTSF = 0，然后向寄存器 CBnTX 写入要发送的数据。 • 只能接收通信模式下(CBnTXE = 0, CBnRXE = 1)，要连续执行下一次接收，确保 CBnTSF = 0，然后读取寄存器 CBnTX。 或者，使用连续发送模式而不用单独发送模式。推荐在使用 DMA 时使用连续发送模式。	p. 554 <input type="checkbox"/>
第 17 章	软件	I ² C 总线	I ² C 总线	为了使用 I ² C 总线功能，分别设置 P38/SDA00, P39/SCL00, P40/SDA01, P41/SCL01, P90/SDA02 和 P91/SCL02 引脚分别作为发送/接收数据 I/O 引脚(SDA00 - SDA02)和串行时钟 I/O 引脚(SCL00 - SCL02)，且将它们设置为 N-ch 开漏输出。	p. 555 <input type="checkbox"/>
			UARTA2 和 I ² C00 模式转换	在传送或接收过程中，如果转换了这些功能，那么 UARTA2 和 I ² C00 的传送/接收操作并不能确保。一定要禁止那个不用的功能。	p. 555 <input type="checkbox"/>
			CSIB0 和 I ² C01 的模式转换	在传送或接收过程中，如果转换了这些功能，那么 UARTA2 和 I ² C01 的传送/接收操作并不能确保。一定要禁止那个不用的功能。	p. 556 <input type="checkbox"/>
			UARTA1 和 I ² C02 的模式转换	在传送或接收过程中，如果转换了这些功能，那么 UARTA2 和 I ² C02 的传送/接收操作并不能确保。一定要禁止那个不用的功能	p. 557 <input type="checkbox"/>
			IICC0 - IICC2 寄存器	当 SCL0n 线是高电平，SDA0n 线是低电平，I ² Cn 操作允许(IICE _n 位 = 1)时，那么立即监测到开始条件。为了避免这种情况，允许 I ² Cn 操作之后，立即用位操作指令将 LRELn 位置 1。	p. 564 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 17 章	软件	I ² C 总线	IICC0 - IICC2 寄存器	只有在主模式下，设置 SPTn 位为 1。然而，当 IICRSVn 位为 0 时，SPTn 位必须设置为 1，并且切换到操作允许状态后，监测到第一个停止条件之前产生停止条件。详细情况，请看 17.15 注意事项	p. 567 <input type="checkbox"/>
				当 TRCn 位 = 1 时，在第 9 个时钟和等待状态取消期间，WRELn 位设置为 1，之后，此位清零，并且 SDA0n 线设置为高阻状态。	p. 567 <input type="checkbox"/>
			IICS0 - IICS2 寄存器	A 下列几种状态禁止访问 IICSn 寄存器。详细情况，参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。 • 当 CPU 运行于副时钟，并且主时钟振荡停止时 • 当 CPU 运行于内部振荡时钟时	p. 568 <input type="checkbox"/>
				. 当 WRELn 位设置为 1，还有，在第 9 个时钟时，通过 TRCn = 1，将等待状态取消为 0 时，TRCn 位清零，SDA0n 线为高阻态。	p. 569 <input type="checkbox"/>
			IICF0 - IICF2 寄存器	只有当操作停止(IICEn 位 = 0)时，写 STCENn 位。	p. 572 <input type="checkbox"/>
				当 STCENn 位 = 1 时，I2Cn 总线操作允许后，立即识别总线释放状态(IICBSYn 位 = 0)，而不管实际总线状态。因此，为了发出第一个开始条件(STTn 位 = 1)，有必要确认总线释放，以免干扰其它通信	p. 572 <input type="checkbox"/>
				只有当操作停止(IICEn 位 = 0)时，写 IICRSVn 位。	p. 572 <input type="checkbox"/>
			IICCL0 - IICCL2 寄存器	一定要将 IICCLn 的位 7 和为 6 清零。	p. 573 <input type="checkbox"/>
			I ² C0n 传送时钟设置方法	既然选择时钟为 fxx，而不管 OCKS0 寄存器的数值为何，OCKS0 寄存器清 00H (I ² C 分频时钟停止状态)	p. 575 <input type="checkbox"/>
				既然选择时钟为 fxx，而不管 OCKS1 寄存器的数值为何，OCKS1 寄存器清 00H (I ² C 分频时钟停止状态)	p. 576 <input type="checkbox"/>
			开始条件	同其它设备的通信进行过程中，当 V850ES/JG2 的 IICn.IICEn 位设置为 1 时，依据通信线的状态，可能监测到开始条件。当 SCL0n 和 SDA0n 线为高电平时，一定要把 IICn.IICEn 位设置为 1。	p. 580 <input type="checkbox"/>
			仲裁状态和中断请求信号发生时序	当 IICn.WTIMn 位 = 1，在第 9 个时钟的下降沿产生 INTIICn 信号。当 WTIMn 位 = 0，并且接收到扩展码的从地址时，在第 9 个时钟的下降沿产生 INTIICn 信号(n = 0 - 2)。	p. 612 <input type="checkbox"/>
				当 IICn.SPTn 位设置为 1 时，停止条件产生。当 IICn.SPIEn 位设置为 1 (n = 0 到 2)，监测到停止条件时，IICSn.SPDn 位设置为 1，并且产生中断请求信号 (INTIICn)。	p. 612 <input type="checkbox"/>
			当 IICFn.STCENn 位 = 0	I ² C0n 操作允许后，不管实际总线状态为何，立即识别总线通信状态(IICFn.IICBSYn 位 = 1)。在还没有监测到停止条件的状态下，执行主设备通信，那么在开始主设备通信之前，停止条件产生，然后总线释放。 依照下列顺序产生停止条件。 <1>设置 IICCLn 寄存器。 <2>设置 IICn.IICEn 位。 <3>设置 IICn.SPTn 位。	p. 618 <input type="checkbox"/>
			当 IICFn.STCENn 位 = 1	当 STCENn 位 = 1 时，I2Cn 总线操作允许后，立即识别总线释放状态(IICBSYn 位 = 0)，而不管实际总线状态。因此，为了发出第一个开始条件(STTn 位 = 1)，有必要确认总线释放，以免干扰其它通信。	p. 618 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 17 章	软件	I ² C 总线	同其它设备的通信进行过程中	同其它设备的通信进行过程中, 当 V850ES/JG2 的 IICn.IICEn 位设置为 1 时, 依据通信线的状态, 可能监测到开始条件。当 SCL0n 和 SDA0n 线为高电平时, 一定要把 IICn.IICEn 位设置为 1。	p. 618 <input type="checkbox"/>
			允许操作	允许操作(IICn.IICEn 位 = 1)之前, 由 IICLn, IICXn 和 OCKSm 寄存器决定操作时钟频率。为了改变操作时钟频率, 清 IICn.IICEn 位一次	p. 618 <input type="checkbox"/>
			IICn.STTn, SPTn 位	IICn.STTn 和 IICn.SPTn 位置 1 后, 在没有首先清零的情况下, 没有必要重置。	p. 618 <input type="checkbox"/>
			通信保留	如果通信已经保留, 设置 IICn.SPIEn 位为 1, 以便中断请求通过监测停止条件产生。中断请求信号产生后, 等待状态通过写通信数据给 I2Cn 释放, 然后传送开始。如果中断不通过监测停止条件产生, 在等待状态时, 传送将停止, 因为不产生中断请求。然而, 没有必要软件设置 SPIEn 位为 1 监测 IICn.MSTSn 位。	p. 618 <input type="checkbox"/>
			在单一主系统里的主操作	根据通信中产品的规范来释放 I ² C0n 总线 (SCL0n, SDA0n 引脚为高电平)。举例来说, 当 EEPROM TM 输出一个低电平给 SDA0n 引脚时, 设置 SCL0n 引脚为输出端口并输出时钟脉冲, 直到 SDA0n 引脚恒定为高电平为止。	p. 620 <input type="checkbox"/>
			在多主系统里的主操作	确认总线释放状态 (IICLn.CLDn = 1, IICLn.DADn = 1) 已经维持了一个特定的时间 (例如 1 帧)。当 SDA0n 引脚恒定为低电平时, 请根据通信中产品的规范来决定是否释放 I ² C0n 总线 (SCL0n, SDA0n 引脚为高电平)。	p. 621 <input type="checkbox"/>
				关于传送和接收的格式, 请确认通信中产品的规格书。	p. 623 <input type="checkbox"/>
				当使用 V850ES/JG2 作为多主系统里的主设备时, 请在每个 INTIICn 中断发生时读取 IICn.MSTSn 位, 以确认仲裁结果。	p. 623 <input type="checkbox"/>
				当使用 V850ES/JG2 作为多主系统里的从设备时, 请在每个 INTIICn 中断发生时用 IICSn 和 IICFn 寄存器确认状态, 以决定下一步处理。	p. 623 <input type="checkbox"/>
			取消从设备等待	当 IICn.WRELn 位置 1, 或者当 FFH 写入 IICn 寄存器时, 等待状态取消	pp. 628 - 630 <input type="checkbox"/>
			取消主设备等待	当 IICn.WRELn 位置 1, 或者当 FFH 写入 IICn 寄存器时, 等待状态取消	pp. 631 - 633 <input type="checkbox"/>
第 18 章	软件	DMA 功能 (DMA 控制器)	DSA0 - DSA3 寄存器	一定要将 DSA nH 寄存器的位 14 到 10 清 0。	p. 636 <input type="checkbox"/>
				当 DMA 传送被禁止的时候(DCHCn.Enn 位为 0), 在下列时序下设定 DSA nH 和 DSA nL 寄存器。 <ul style="list-style-type: none"> 在复位后开始发送第一个 DMA 传送之前的时期 在通过 DCHCn.INITn 位将信道初始化后来开始 DMA 传送的时期 在完成第一个 DMA 传送(DCHCn.TCn 位为 1)和开始下一个 DMA 传送之间的时期 	p. 636 <input type="checkbox"/>

(24/36)

章节	分类	功能	具体功能	注意事项	页码
第 18 章	软件	DMA 功能 (DMA 控制器)	DSA0 - DSA3 寄存器	当 DSA _n 寄存器中的数值被读取, 两个 16 位寄存器 DSA _n H 和 DSA _n L 的值被读取。如果读取操作与数据更新发生冲突, 将读取正在被更新的数据(参见 18.13 注意事项)。	p. 636 <input type="checkbox"/>
				复位后, 在开始 DMA 传送之前要首先设定 DSA _n H, DSA _n L, DDA _n H, DDA _n L 和 DBC _n 寄存器。否则, 当 DMA 发送开始后的操作无法得到保证。	p. 636
			DDA0 - DDA3 寄存器	一定要将 DDA _n H 寄存器的位 14 到 10 清 0。	p. 637
				当 DMA 传送被禁止的时候(DCHC _n .Enn 位为 0), 在下列时序下设定 DDA _n H 和 DDA _n L 寄存器。 <ul style="list-style-type: none"> 在复位后开始发送第一个 DMA 传送之前的时期 在通过 DCHC_n.INIT_n 位将信道初始化后来开始 DMA 传送的时期 在完成第一个 DMA 传送(DCHC_n.TC_n 位为 1)和开始下一个 DMA 传送之间的时期 	p. 637 <input type="checkbox"/>
				当 DDA _n 寄存器中的数值被读取, 两个 16 位寄存器 DDA _n H 和 DDA _n L 的值被读取。如果读取操作与数据更新发生冲突, 将读取正在被更新的数据(参见 18.13 注意事项)。	p. 637 <input type="checkbox"/>
				复位后, 在开始 DMA 传送之前要首先设定 DSA _n H, DSA _n L, DDA _n H, DDA _n L 和 DBC _n 寄存器。否则, 当 DMA 发送开始后的操作无法得到保证。	p. 637 <input type="checkbox"/>
			DBC0 - DBC3 寄存器	当 DMA 传送被禁止的时候(DCHC _n .Enn 位为 0), 在下列时序下设定 DBC _n 寄存器 <ul style="list-style-type: none"> 在复位后开始发送第一个 DMA 传送之前的时期 在通过 DCHC_n.INIT_n 位将信道初始化后来开始 DMA 传送的时期 在完成第一个 DMA 传送(DCHC_n.TC_n 位为 1)和开始下一个 DMA 传送之间的时期 	p. 638 <input type="checkbox"/>
				复位后, 在开始 DMA 传送之前要首先设定 DSA _n H, DSA _n L, DDA _n H, DDA _n L 和 DBC _n 寄存器。否则, 当 DMA 发送开始后的操作无法得到保证。	p. 638 <input type="checkbox"/>
			DADC0 - DADC3 寄存器	一定要将 DADC _n 寄存器的位 15, 13 到 8, 和 3 清 0。	p. 639 <input type="checkbox"/>
				当 DMA 传送被禁止的时候(DCHC _n .Enn 位为 0), 在下列时序下设定 DADC _n 寄存器。 <ul style="list-style-type: none"> 在复位后开始发送第一个 DMA 传送之前的时期 在通过 DCHC_n.INIT_n 位将信道初始化后来开始 DMA 传送的时期 在完成第一个 DMA 传送(DCHC_n.TC_n 位为 1)和开始下一个 DMA 传送之间的时期 	p. 639 <input type="checkbox"/>
				DS0 位指定了传送数据的大小, 但并不能控制总线容量。因此如果设定为 8 位数据(DS0 位为 0), 较低位的数据总线并不经常使用。	p. 639 <input type="checkbox"/>
				如果传送数据设定为 16 位 (DS0 位为 1), 传送不能从奇数地址开始。传送总是从低地址的首位为 0 的地址开始。	p. 639 <input type="checkbox"/>
				如果由片上外围端口寄存器 (无论作为发送的源还是目的) 执行 DMA 操作, 一定要按照寄存器的大小来设定传送数据的大小。例如, 在一个 8 位寄存器上执行 DMA 传送, 一定要指定 8 位传送。	p. 639 <input type="checkbox"/>
			DCHC0 - DCHC3 寄存器	TC _n 位为只读位。	p. 640 <input type="checkbox"/>
				INIT _n 和 STG _n 位为只写位。	p. 640 <input type="checkbox"/>
				一定要将 DCHC _n 寄存器的位 6 到 3 清 0。	p. 640 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 18 章	软件	DMA 功能 (DMA 控制器)	DCHC0 - DCHC3 寄存器	当 DMA 传送完成时 (中止记数产生), Enn 位清 0 然后 TCn 位置 1。如果在其数据位被更新时读取 DCHCn 寄存器, 指示“传送未完成和传送被禁止(TCn 位为 0 和 Enn 位为 0)”的数值可以被读取。	p. 640 <input type="checkbox"/>
			DTFR0 - DTFR3 寄存器	如果在 DMA 传送禁止期间, 产生了一个被指定作为引起 DMA 传送的中断, 对该位写入 0 将清除一个 DMA 发送请求。	p. 641 <input type="checkbox"/>
				当禁止 DMA 传送时(DCHCn.Enn 位的值为 0), 在以下时序设定 IFCn5 到 IFCn0 位。 <ul style="list-style-type: none"> 在复位后开始发送第一个 DMA 传送之前的时期 在通过 DCHCn.INITn 位将信道初始化后来开始 DMA 传送的时期 在完成第一个 DMA 传送(DCHCn.TCn 位为 1)和开始下一个 DMA 传送之间的时期 	p. 641 <input type="checkbox"/>
				在待机模式 (IDEL1, IDLE2, STOP, 或 sub-IDLE 模式) 下产生的中断请求不会开始 DMA 传送周期 (也不会将 DFn 位置 1)。	p. 641 <input type="checkbox"/>
				如果通过 IFCn5 到 IFCn0 位选择了 DMA 开始因素, 当所选择的片上外围端口的中断产生时, DFn 位被置 1, 而无论 DMA 传送是否被允许。在这种状态下如果允许 DMA, 那么 DMA 发送立即开始。	p. 641 <input type="checkbox"/>
			传送目标之间的关系	如果表 18-2 标记为“x”的源和目的地之间执行操作, 结果将得不到保证。	p. 643 <input type="checkbox"/>
			通过片上外围 I/O 产生请求	两种开始因素(软件触发和硬件触发)不能应用在同一个 DMA 信道, 否则它们中间只有一个是有效的。有效的开始因素不能被识别。	p. 646 <input type="checkbox"/>
				在先前的 DMA 发送请求产生后或是在先前的 DMA 传送周期中产生的新的传送请求将被忽略。	p. 646 <input type="checkbox"/>
				相同的 DMA 信道的传送请求间隔的变化取决于在 DMA 发送周期中总线等待的设定, 其它信道的开始状态或是外部总线的保持请求。特殊情况下, 如注意事项 2 中的描述, 在 DMA 传送周期之前或是 DMA 传送周期中同一信道产生的新的发送请求将被忽略。因此, 相同 DMA 信道的传送请求间隔必须通过系统充分分隔。当使用软件触发时, 先前产生的 DMA 传送周期是否完成可以通过更新 DBCn 寄存器检测。	p. 646 <input type="checkbox"/>
			VSWC 寄存器的注意事项	当使用 DMAC 时, 一定要依照操作频率为 VSWC 寄存器设定适当的值 当使用 VSWC 寄存器的缺省值 (77H), 或者为 VSWC 寄存器设定的值不恰当时, 不会执行正确的操作(VSWC 寄存器的详细信息, 参见 3.4.8 (1) (a) 系统等待控制寄存器 (VSWC))。	p. 652 <input type="checkbox"/>
			在内部 RAM 中执行 DMA 传送的注意事项	当在内部 RAM 中执行下列指令时, 不要执行向/从内部 RAM(无论是源或者目的)中传送数据的 DMA 传送, 这是由于 CPU 之后可能不会正确的运行。 <ul style="list-style-type: none"> 在内部 RAM 中的位操作指令(SET1, CLR1, 或是 NOT1)。 在内部 RAM 中对于非对齐地址的数据访问指令。 相反的, 当向/从内部 RAM 中(无论是源或者目的)执行传送数据的 DMA 传送时, 不要执行上述两种指令。	p. 652 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 18 章	软件	DMA 功能 (DMA 控制器)	读取 DCHCn.TCn 位的注意事项	<p>当对 TCn 位进行读取操作时它将被清 0，但是它并不是自动被清 0 即使是在一个特殊的时序下。为了正确地将 TCn 位清 0，要加上下列处理过程。</p> <p>(a) 当通过轮询 TCn 位的方式等待 DMA 传送完成。 确认 TCn 位已经被置 1 (在 TCn 位为 1 被读取后)，然后另外再读取 TCn 位三次。</p> <p>(b) 当在中断服务程序中读取 TCn 位时。 执行读取 TCn 位 3 次。</p>	p. 652 <input type="checkbox"/>
			DMA 传送初始化过程 (将 DCHCn.INITn 位置 1)	<p>当要执行 DMA 传送的信道被初始化时，即使将 INITn 位置 1，该信道也不会被初始化。为了正确的初始化信道，需要执行下列两个过程之一。</p> <p>(a) 临时中止所有 DMA 信道的传送 初始化正在执行 DMA 传送的信道需要按以下步骤<1>到<7> 执行 注，但是当执行第<5>步时 TCn 位被清 0。确定其它处理程序不需要 TCn 位为 1。</p> <p><1> 禁止中断 (DI)</p> <p><2> 读取被强行中止的 DMA 信道之外的其它 DMA 信道的 DCHCn.Enn 位，然后将这个值传送到通用目的寄存器。</p> <p><3> 将正在被使用的 DMA 信道的 Enn 位 (包括被强行中止的信道)清 0。为了将最后一个 DMA 信道的 Enn 位清 0，需要执行两次清除指令。如果 DMA 传送的目的 (传送源/目的) 是内部 RAM，需要将该指令执行三次。</p> <p>例如： 如果正在使用信道 0，1 和 2 (如果传送的目的不是内部 RAM)，按照以下顺序执行指令</p> <ul style="list-style-type: none"> • 将 DCHC0.E00 位清 0 • 将 DCHC1.E11 位清 0 • 将 DCHC2.E22 位清 0 • 再次将 DCHC2.E22 位清 0 <p><4> 将被强行中止的 DMA 信道的 INITn 位置 1。</p> <p><5> 读取除被强行中止的信道外的其它各个信道的 TCn 位。如果在第<2>步中读取的 TCn 位和 Enn 位的值都为 1 (逻辑与的结果为 1)，将保存的 Enn 位清 0。</p> <p><6> 在第<5>步操作之后，将 Enn 位的值写入 DCHCn 寄存器。</p> <p><7> 允许中断 (EI)。</p>	p. 653 <input type="checkbox"/>
				一定要执行上述的第<5>步来防止对正常完成第<2>和<3>步之间 DMA 传送信道的 Enn 位的非法设置。	p. 653 <input type="checkbox"/>
				<p>(b) 重复执行 INITn 位的设定直到传送被正确地强行中止。</p> <p><1> 抑制要被强行中止的信道的 DMA 请求源发出的请求(停止片上外围端口的操作)。</p> <p><2> 通过使用 DTFRn.DFn 位来检测要被强行中止的信道的 DMA 传送请求未处于悬挂状态。如果一个 DMA 传送请求处于悬挂状态，需要等待直到执行完该悬挂请求。</p> <p><3> 如果确定要被强行中止的信道的 DMA 请求未处于悬挂状态，将 Enn 位清 0。</p> <p><4> 再次将要被强行中止的信道的 Enn 位清 0。 如果要被强行中止的信道 (传送源/目的) 的传送的目的是内部 RAM，再次执行该操作。</p> <p><5> 复制要被强行中止的信道的传送初始值到通用目的寄存器。</p> <p><6> 将要被强行中止的信道的 INITn 位置 1。</p> <p><7> 读取要被强行中止的信道的 DBCn 寄存器的值，并与第<5>步中的复制值相比较。如果两者不匹配，重复第<6>步和第<7>步的操作。</p>	p. 654 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 18 章	软件	DMA 功能 (DMA 控制器)	临时停止 DMA 传送的程序 (将 Enn 位清 0)	<p>停止和恢复 DMA 传送需要按以下步骤执行。</p> <p><1> 抑制 DMA 请求源的传送请求 (停止片上外围端口的操作)。</p> <p><2> 使用 DFn 位来检测 DMA 传送请求未处于悬挂状态 (检测 DFn 位是否为 0)。如果悬挂请求, 等待直到被悬挂的 DMA 传送请求执行完毕。</p> <p><3> 如果确认没有 DMA 传送请求被悬挂, 将 Enn 位清 0 (该操作停止 DMA 传送)。</p> <p><4> 将 Enn 位置 1 恢复 DMA 传送。</p> <p><5> 恢复被停止的 DMA 请求源的操作 (开始片上外围端口的操作)。</p>	p. 654 <input type="checkbox"/>
			内存边界	如果在 DMA 传送期间传送源或者目的地址超出了 DMA 目标的区域, 那么操作将不会得到保证(外部内存, 内部 RAM, 或是片上外围端口)。	p. 654 <input type="checkbox"/>
			传送非对齐数据	不支持在 16 位总线宽度上非对齐数据的 DMA 传送。 如果一个奇数地址被指定为传送源或者目的, 地址的最低有效位被强制假定为 0。	p. 654 <input type="checkbox"/>
			对于 CPU 的总线仲裁	<p>由于 DMA 控制器对于总线的控制权优先于 CPU, 因此发生在 DMA 传送期间的 CPU 访问被悬挂直到 DMA 传送周期完成且将总线释放给 CPU。</p> <p>但是, CPU 可以访问那些未执行 DMA 传送的外部内存, 片上外围端口和内部 RAM。</p> <ul style="list-style-type: none"> 当在外部内存和片上外围端口之间执行 DMA 传送时, CPU 可以访问内部 RAM。 当在外部内存之间执行 DMA 传送时, CPU 可以访问内部 RAM 和片上外围端口。 	p. 655 <input type="checkbox"/>
			在 DMA 操作期间绝不能对寄存器/位进行重新写入	<p>在 DMA 操作未被执行时在下面的时序中设定下列寄存器。</p> <p>[寄存器]</p> <ul style="list-style-type: none"> DSAnH, DSAnL, DDAAnH, DDAAnL, DBCn, 和 DADCn 寄存器 DTFRn.IFCn5 至 DTFRn.IFCn0 位 <p>[时序设定]</p> <ul style="list-style-type: none"> 在复位后到开始第一个 DMA 传送之间的时期 在信道初始化到开始 DMA 传送的时间 在完成第一个 DMA 传送 (TCn 位为 1)到开始下一个 DMA 发送之间的时期 	p. 655 <input type="checkbox"/>
			DSAnH 寄存器 DDAAnH 寄存器 DADCn 寄存器 DCHCn 寄存器	<p>一定要将下列寄存器的位清 0。</p> <ul style="list-style-type: none"> DSAnH 寄存器的位 14 到 10 DDAAnH 寄存器的位 14 到 10 DADCn 寄存器的位 15, 13 到 8, 和 3 到 0 DCHCn 寄存器的位 6 到 3 	p. 655 <input type="checkbox"/>
			DMA 开始因素	不要使用相同的开始因素开始两个或以上的 DMA 信道。否则, 已被设置的信道的 DMA 传送可能开始或者具有较低优先级的信道会比具有较高优先级的信道更早被响应。操作将得不到保证。	p. 655 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 18 章	软件	DMA 功能 (DMA 控制器)	从 DSA _n 和 DDA _n 寄存器中读取数值	<p>在 DMA 传送期间, 可以在 DSA_n 和 DDA_n 寄存器中读取更新中的值 (n 的取值范围是 0 到 3)。</p> <p>例如, 如果当 DMA 传送源地址(DSA_n 寄存器)为 0000FFFFH 且计数方式为递增 (DADC_n.SAD1 和 DADC_n.SAD0 位为 00), 此时读取 DSA_nH 寄存器和 DSA_nL 寄存器, DSA_n 寄存器的值的区别如下所示, 这将取决于在 DSA_nH 寄存器被读取之后 DMA 传送是否被立即执行。</p> <p>(a) 如果在 DSA_n 寄存器被读取时 DMA 传送没有发生</p> <p><1> DSA_nH 寄存器的读取值为: DSA_nH = 0000H</p> <p><2> DSA_nL 寄存器的读取值为: DSA_nL = FFFFH</p> <p>(b) 如果在 DSA_n 寄存器被读取时发生了 DMA 传送</p> <p><1> DSA_nH 寄存器的读取值为: DSA_nH = 0000H</p> <p><2> 发生 DMA 传送</p> <p><3> 增加 DSA_n 寄存器: DSA_n 的取值是 00100000H</p> <p><4> DSA_nL 寄存器的读取值为: DSA_nL = 0000H</p>	p. 656 <input type="checkbox"/>
第 19 章	软件	中断/异常处理功能	非屏蔽中断	<p>由非屏蔽中断请求信号(INTWDT2)执行的非屏蔽中断服务, 参考 19.2.2 (2) INTWDT2 信号。</p> <p>在非屏蔽中断服务期间, 当 EP 位和 NP 位通过 LDSR 指令改变时, 在 RETI 指令执行之后, 为了正确的还原 PC 和 PSW, 必须在 RETI 指令执行之前使用 LDSR 指令将 EP 位设置为 0, NP 位设置为 1。</p>	p. 661 <input type="checkbox"/> p. 664 <input type="checkbox"/>
			可屏蔽中断	在可屏蔽中断服务执行期间, 当通过 LDSR 指令将 EP 位和 NP 位改变时, 为了在 RETI 指令后正确的恢复 PC 和 PSW, 需要在 RETI 指令之前执行 LDSR 指令将 EP 位和 NP 位清 0。	p. 668 <input type="checkbox"/>
			多重中断	为了执行复合中断服务, 在执行 EI 指令之前必须保存 EIPC 寄存器和 EIPSW 寄存器的值。当从复合中断服务中返回, 在执行 DI 指令后恢复 EIPC 和 EIPSW 的值。	pp. 670 - 672 <input type="checkbox"/>
			中断控制寄存器	<p>禁止中断 (DI)或者屏蔽中断来读取 xxIC_n.xxIF_n 位。如果在允许中断(EI)或是中断未被屏蔽的时候, 读取 xxIF_n 位, 确认中断且读取该位冲突时数据可能有误。</p> <p>如果确认中断信号后, xxIF_n 标志通过硬件自动复位。</p>	p. 673 <input type="checkbox"/> p. 673 <input type="checkbox"/>
			IMR0 - IMR3 寄存器	<p>设备文件定义了 xxIC_n.xxMK_n 位为保留字。如果有其它的位使用 xxMK_n 作为名称进行操作, xxIC_n 寄存器的内容会被写入, 而不是 IMR_m 寄存器(结果, IMR_m 寄存器的内容也会被覆盖)。</p> <p>为了以 8 位或 1 位为单元读取 IMR0 到 IMR3 寄存器的位 8 到 15, 需要将它们设定为 IMR0H 到 IMR3H 寄存器的位 0 到 7。</p> <p>将 IMR3 寄存器的位 13 到 15 置 1。如果这些位的设定值改变, 操作将不会得到保证。</p>	p. 675 <input type="checkbox"/> p. 676 <input type="checkbox"/> p. 676 <input type="checkbox"/>
			ISPR 寄存器	如果在中断允许状态(EI)对 ISPR 寄存器进行读取时确认一个中断, 在寄存器的位通过确认中断的方式被设置之后, 可以读取 ISPR 寄存器的值。为了在确认中断之前正确的读取 ISPR 寄存器的值, 需要禁止中断(DI)。	p. 677 <input type="checkbox"/>

(29/36)

章节	分类	功能	具体功能	注意事项	页码
第 19 章	软件	中断/异常处理功能	软件异常进程中恢复	在软件异常处理过程中，当通过 LDSR 指令改变 EP 位和 NP 位时，为了在 RETI 指令执行后正确的恢复 PC 和 PSW，在 RETI 指令执行前必须立即执行 LDSR 指令将 EP 位置 1，NP 位清 0。	p. 680 <input type="checkbox"/>
			非法的操作代码	由于将来有可能分配该指令到一个非法的操作代码，推荐不要使用它	p. 682 <input type="checkbox"/>
			异常陷阱的恢复	仅在执行非法操作代码和 DBRET 指令的间隔内才能够访问 DBPC 和 DBPSW。	p. 683 <input type="checkbox"/>
			从调试陷阱中恢复	在 DBTRAP 指令执行后 DBRET 指令执行之前，可以访问 DBPC 和 DBPSW。	p. 685 <input type="checkbox"/>
			INTF0, INTR0 寄存器	当引脚的功能从外部中断功能(被用功能)转变到端口功能时，可以进行边沿检测。因此，首先要将 INTF0n 位和 INTR0n 位设定为 00，然后再设定端口模式。	p. 687 <input type="checkbox"/>
				当这些寄存器不被用作 NMI 或 INTP0 到 INTP3 时，一定要将 INTF0n 和 INTR0n 位设定为 00。	p. 687 <input type="checkbox"/>
			INTF3, INTR3 寄存器	当引脚的功能从外部中断功能(被用功能)转变到端口功能时，可以进行边沿检测。因此，首先要将 INTF0n 位和 INTR0n 位设定为 00，然后再设定端口模式。	p. 688 <input type="checkbox"/>
				INTP7 引脚和 RXDA0 引脚是复用引脚。当使用 RXDA0 引脚引脚功能时，要禁止 INTP7 复用功能引脚的边沿检测功能(将 INTF3.INTF31 位和 INTR3.INTR31 位清 0)。当使用 INTP7 引脚功能时，要停止 UAR0A0 接收(将 UA0CTL0.UA0RXE 位清 0)。	p. 688 <input type="checkbox"/>
				当这些寄存器不被用作 NMI 或 INTP7 时，一定要将 INTF31 和 INTR31 位设定为 00。	p. 688 <input type="checkbox"/>
			INTF9H, INTR9H 寄存器	当引脚的功能从外部中断功能(被用功能)转变到端口功能时，可以进行边沿检测。因此，首先要将 INTF9n 位和 INTR9n 位设定为 00，然后再设定端口模式。	p. 689 <input type="checkbox"/>
				当这些寄存器不被用作 INTP4 或 INTP6 时，一定要将 INTF9n 和 INTR9n 位设定为 00。	p. 689 <input type="checkbox"/>
			NFC 寄存器	在改变采样时钟之后，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的 3 个时钟之后 INTP3 输入有效边沿，将产生中断请求。因此，当使用中断和 DMA 功能的时候，一定要注意以下几点。 • 当使用中断功能时，在经过 3 个采样时钟之后，在中断请求标志(PIC3.PIF3 位)被清除后，要允许中断。 • 当使用 DMA 功能时(由 INTP3 始发)，在经过 3 个采样时钟之后允许 DMA。	p. 690 <input type="checkbox"/>
			NMI 引脚	NMI 引脚和 P02 引脚是功能复用引脚，复位后的功能是一个正常的端口。为了执行 NMI 的功能，在 PMC0 寄存器中确认 NMI 引脚。NMI 引脚的初始设定是不检测边沿。使用 INTF0 寄存器和 INTR0 寄存器设定 NMI 引脚的有效边沿。	p. 692 <input type="checkbox"/>
第 20 章	软件	按键中断功能	KRM 寄存器	在 KRM 寄存器被清零(00H)后，重写 KRM 寄存器。	p. 694 <input type="checkbox"/>
				如果改变 KRM 寄存器，则会产生中断请求信号(INTKR)。为避免这种情况，在禁止中断(DI)或屏蔽中断之后改变 KRM 寄存器，然后将中断请求标志(KRIC、KRIF 位)清零，再允许中断或清除屏蔽。	p. 694 <input type="checkbox"/>

(30/36)

章节	分类	功能	具体功能	注意事项	页码
第20章	软件	按键中断功能	KR0 - KR7 引脚	如果低电平被输入给 KR0 ~ KR7 中的任何一个, 即使其它引脚的下降沿被输入, 也不会产生 INTKR 信号。	p. 694 <input type="checkbox"/>
			RXDA1 引脚 KR7 引脚	不能同时使用 RXDA1 和 KR7 引脚。为了使用 RXDA1 引脚, 就不要使用 KR7 引脚。为了使用 KR7 引脚, 就不要使用 RXDA1 引脚 (推荐设置 PFCE91 位为 1 并且将 PFCE91 位清零)。	p. 694 <input type="checkbox"/>
			使用按键中断功能	为了使用按键中断功能, 确信设置端口引脚为按键返回引脚, 然后使用 KRM 寄存器允许操作。为了从按键返回引脚切换到端口引脚, 使用 KRM 寄存器禁止操作, 然后设置为端口引脚。	p. 694 <input type="checkbox"/>
第21章	软件	待机功能	PSC 寄存器	在设置 IDLE1, IDLE2, STOP 或副 IDLE 模式之前, 先设置 PSMR.PSM1 和 PSMR.PSM0 位, 然后设置 STP 位。	p. 697 <input type="checkbox"/>
				当释放 HALT 模式时对 NMI1M, NMI0M 和 INTM 位进行设置是非法的。	p. 697 <input type="checkbox"/>
				如果在对 NMI1M, NMI0M, 或 INTM 位置 1 相同的时刻对 STP 位也置 1, 则对, NMI0M, 或 INTM 这些位的设置无效。当设置了 IDLE1/IDLE2/STOP 模式后, 有一个挂起的不可屏蔽的中段请求, 则要与中断请求信号相关的位(NMI1M, NMI0M, 或 INTM)置 1, 然后在将 STP 位置 1。	p. 697 <input type="checkbox"/>
			PSMR 寄存器	确定将第 2 位 ~ 第 7 位清零。	p. 698 <input type="checkbox"/>
				仅当 PSC.STP 位为 1 时, PSM0 和 PSM1 位是有效的。	p. 698 <input type="checkbox"/>
			OSTS 寄存器	STOP 模式释放后的等待时间不包括从释放 STOP 模式到时钟振荡启动这段时间 (即下图“a”所示的部分), 不管通过复位输入或中断请求信号的产生是否释放了 STOP 模式。	p. 699 <input type="checkbox"/>
				确定将第 3 位 ~ 第 7 位清零。	p. 699 <input type="checkbox"/>
				复位释放后的振荡稳定时间是 $2^{16}/f_x$ (因为 OSTS 寄存器的初始值= 06H)	p. 699 <input type="checkbox"/>
			HALT 模式	在 HALT 指令之后插入五个或更多的 NOP 指令	p. 700 <input type="checkbox"/>
				如果在执行 HALT 指令时有一个未屏蔽的中断请求被保持等待响应, 则此状态被转移到 HALT 模式中, 然后通过这个等待响应的中断请求, HALT 模式被立即释放。	p. 700 <input type="checkbox"/>
			IDLE1 模式	在执行往 PSC 寄存器中存入数据来设置 IDLE1 模式的指令之后, 插入五个或更多的 NOP 指令。	p. 702 <input type="checkbox"/>
				如果设置 IDLE1 模式时有一个未被屏蔽的中断请求信号被保持等待响应, 则 IDLE1 模式立即被此等待响应的中断请求释放。	p. 702 <input type="checkbox"/>
			释放 IDLE1 模式	通过设置 PSC.NMI1M, PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号, 这样中断请求信号是无效的, IDLE1 模式也不会被释放。	p. 702 <input type="checkbox"/>
			IDLE2 模式	在执行往 PSC 寄存器中存入数据来设置 IDLE2 模式的指令之后, 插入五个或更多的 NOP 指令。	p. 704 <input type="checkbox"/>
				如果设置 IDLE2 模式时有一个未被屏蔽的中断请求信号被保持等待响应, 则 IDLE2 模式立即被此等待响应的中断请求释放。	p. 704 <input type="checkbox"/>
			释放 IDLE2 模式	通过设置 PSC.NMI1M, PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号, 这样中断请求信号是无效的, IDLE2 模式也不会被释放。	p. 704 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 21 章	软件	待机功能	STOP 模式	在执行往 PSC 寄存器中存入数据来设置 STOP 模式的指令之后，插入五个或更多的 NOP 指令。	p. 707 <input type="checkbox"/>
				如果设置 STOP 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则 STOP 模式立即被此等待响应的中断请求释放。	p. 707 <input type="checkbox"/>
			释放 STOP 模式	通过设置 PSC.NMI1M, PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号是无效的，STOP 模式也不会被释放。	p. 707 <input type="checkbox"/>
			副时钟模式	当操作 CK3 位时，不要改变 PCC.CK2 ~ PCC.CK0 位（推荐使用位操作指令来操作）的设置值。对于 PCC 寄存器的细节描述，参见 6.3 (1) 处理器时钟控制寄存器(PCC)。	p. 711 <input type="checkbox"/>
				如果以下的条件不满足，改变 CK2 ~ CK0 位的设置以便满足条件并且设置副时钟操作模式。 主时钟(f _{xx}) > 副时钟 (f _{XT} = 32.768 kHz) × 4	p. 711 <input type="checkbox"/>
			释放副时钟模式	当操作 CK3 位时，不要改变 CK2 ~ CK0 位（推荐使用位操作指令来操作）的设置值。对于 PCC 寄存器的细节描述，参见 6.3 (1) 处理器时钟控制寄存器(PCC)。	p. 711 <input type="checkbox"/>
				停止主时钟前一定要停止 PLL (PLLCTL.PLLON 位= 0)。	p. 712 <input type="checkbox"/>
				当 CPU 工作在副时钟并且主时钟振荡停止时，不能访问发生等待的寄存器。如果产生等待，则其只能通过复位来释放（见 3.4.8 (2)）。	p. 712 <input type="checkbox"/>
			Sub-IDLE 模式	在执行往 PSC 寄存器中存入数据来设置副 IDLE 模式的指令之后，插入五个或更多的 NOP 指令。	p. 713 <input type="checkbox"/>
				如果设置副 IDLE 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则副 IDLE 模式立即被此等待响应的中断请求释放。	p. 713 <input type="checkbox"/>
			释放 sub-IDLE 模式	通过设置 PSC.NMI1M, PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号是无效的，副 IDLE 模式也不会被释放。	p. 713 <input type="checkbox"/>
				当释放副 IDLE 模式时，从产生释放副 IDLE 模式的中断请求信号开始到副 IDLE 模式被释放之间须等待 12 个副时钟周期（大约 366 μs）	p. 713 <input type="checkbox"/>
				停止主时钟前一定要停止 PLL (PLLCTL.PLLON 位= 0)。	p. 714 <input type="checkbox"/>
第 22 章	软件	复位功能	紧急事件操作模式	当 CPU 在内部振荡时钟下操作，有权使用在等待状态下禁止使用的寄存器。关于等待状态下禁止使用的寄存器，参见 3.4.8 (2) 访问特殊片上外围 I/O 寄存器。	p. 715 <input type="checkbox"/>
			复位功能	LVI 电路的内部复位不会使 LVI 电路复位。	p. 715 <input type="checkbox"/>
			RESF 寄存器	寄存器的每一位只能被写入“0”。如果写入“0”和设置标志(复位事件)冲突，设置标志优先。	p. 716 <input type="checkbox"/>
	硬件		复位后的内部 RAM 状态	因为支持启动交换功能，V850ES/JG2 的固件在内部系统状态被释放后使用了一部分内部 RAM。因此，一些 RAM 区域的内容在上电复位后不能被保持。详细内容参见 22.3.4 复位释放后的操作	pp. 717, 719, 721 <input type="checkbox"/>
			RESET 引脚输入的硬件状态	当电源开启，下列引脚可能临时在复位时输出一个不确定的电平。 • P10/ANO0 引脚 • P11/ANO1 引脚 • P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 引脚	p. 717 <input type="checkbox"/>

(32/36)

章节	分类	功能	具体功能	注意事项	页码
第 22 章	硬件, 软件	复位功能	RESET 引脚输入的硬件状态	RESET 引脚的输入初始化 OCDM 寄存器。因此, 注意注意事项, 复位释放后在 OCDM.OCDM0 位被清除前如果一个高电平输入到 P05/DRST 引脚, 可能会进入片上调试模式。详细内容参见第四章 端口功能。	p. 717 <input type="checkbox"/>
第 23 章	软件	时钟监控器	CLM 寄存器	一旦 CLME 位被设为 1, 除了复位其余任何操作都不能将其清为 0。	p. 727 <input type="checkbox"/>
				由时钟监控器引起的一个复位, CLME 位被清为 0 并且 RESF.CLMRF 被设置为 1。	p. 727 <input type="checkbox"/>
			内部振荡器	内部振荡器可以通过设置 RCM.RSTOP 位为 1 来停止。	p. 728 <input type="checkbox"/>
				内部振荡器停止的时候时钟监控器也停止。	p. 728 <input type="checkbox"/>
				内部振荡器不能通过软件来停止。	p. 728 <input type="checkbox"/>
第 24 章	软件	低电压监测器 (LVI)	LVIM 寄存器	当 LVION 和 LVIMD 位为 1 时, 低电压检测电路不能停止, 直到产生除 LVI 复位以外的复位请求时	p. 732 <input type="checkbox"/>
				当 LVION=1 时, LVI 电路中的比较器开始操作。从设置 LVION=1 到 LVIF 确定电压, 使用软件等待 0.2 ms 或更长	p. 732 <input type="checkbox"/>
				一定要将位 6 ~ 2 清 0	p. 732 <input type="checkbox"/>
			LVIS 寄存器	当 LVION 和 LVIMD 位为 1 时, 此寄存器不能写, 直到产生除 LVI 复位以外的复位请求时。	p. 733 <input type="checkbox"/>
				一定要将位 7 ~ 1 清 0	p. 733 <input type="checkbox"/>
			用作内部复位信号	如果 LVIMD 位设置为 1, 那么 LVIM 和 LVIS 寄存器的内容不能改变, 直到 LVI 之外的复位请求产生。	p. 734 <input type="checkbox"/>
第 25 章	硬件	调节器	稳压器	设置 $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} \geq BV_{DD}$ 时, 使用调节器。	p. 738 <input type="checkbox"/>
第 26 章	硬件	FLASH 存储器	FLMD1 引脚	连接 FLMD1 引脚到 flash 编程器或在板上经下拉电阻接地。	pp. 745 to 747 <input type="checkbox"/>
			PG-FP4	按图 26-6 中所示连接这些引脚, 或在板上经下拉电阻接地。	p. 747 <input type="checkbox"/>
				无法由 flash 编程器的 CLK 引脚提供时钟。板上需要有振荡器并提供时钟。	p. 747 <input type="checkbox"/>
			V850ES/JG2 Flash 写入适配器接线 (FA-100GF-3BA-A, FA-100GC-8EU-A)	确保 REGC 引脚经 4.7 μF 电容接地。	pp. 748, 749 <input type="checkbox"/>
				时钟无法由 flash 编程器的 CLK 引脚提供。时钟无法由 flash 编程器的 CLK 引脚提供。	pp. 748, 749 <input type="checkbox"/>
			V850ES/JG2 Flash 写入适配器连接示例 (FA-100GF-3BA-A)	按照下面所示连接 FLMD1 引脚, 或在板上经下拉电阻将其接地。	p. 751 <input type="checkbox"/>
				在 flash 写入适配器上设置一个振荡器(虚线处所示)并且提供时钟	p. 751 <input type="checkbox"/>
				要输入高电平到 DRST 引脚。	p. 751 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 26 章	硬件	FLASH 存储器	V850ES/JG2 flash 写入适配器连接示例 (FA-100GC-8EU-A)	按照下面所示连接 FLMD1 引脚，或在板上经下拉电阻将其接地。	p. 753 <input type="checkbox"/>
				在 flash 写入适配器上设置一个振荡器(虚线处所示)并且提供时钟	p. 753 <input type="checkbox"/>
				不要输入高电平到 DRST 引脚。	p. 753 <input type="checkbox"/>
			通信模式的选择	当选择 UARTA0 时，在接收到 FLMD0 脉冲后，根据专用 Flash 编程器发送的复位命令计算接收时钟	p. 755 <input type="checkbox"/>
			FLMD1 引脚	如果 VDD 信号在在板写入期间和复位伊始后从其他设备输入到 FLMD1 引脚，隔离该信号。	p. 757 <input type="checkbox"/>
			FLMD0 引脚	当复位释放后，确保 FLMD0 引脚为 0 V。	p. 764 <input type="checkbox"/>
第 27 章	硬件, 软件	片上调试功能	OCDM 寄存器	外部复位后，当 DDI, DDO, DCK, 和 DMS 引脚不用于片上调试引脚而作为端口引脚时，可以进行下面的任何一项操作。 • 输入低电压到 P05/INTP2/DRST 引脚。 • 设置 OCDM0 位。在这种情况下，进行如下操作。 <1> OCDM0 位清零。 <2> P05/INTP2/DRST 引脚恒为低电平直到<1>完成。	p. 770 <input type="checkbox"/>
				DRST 引脚有片上下拉电阻。这个电阻当 OCDM0 标志清 0 时断开。	p. 770 <input type="checkbox"/>
	软件	注意事项 (DUC)		RUN (程序执行)期间，如果输入复位信号(从目标系统或从内部复位源来的复位信号)，暂停功能就可能出现故障。	p. 771 <input type="checkbox"/>
				即使复位信号由屏蔽功能屏蔽，如果复位信号是从引脚输入的话，I/O 缓冲器(端口引脚)也可能复位。	p. 771 <input type="checkbox"/>
				因为内部 flash 存储器内的软件断点设置是通过 ROM 修正功能实现的，它可以通过目标复位或由看门狗定时器 2 产生的内部复位强制为无效。当硬件暂停或者强制暂停发生时，断点再次有效。在此期间不产生软件暂停。	p. 771 <input type="checkbox"/>
				暂停期间屏蔽引脚复位屏蔽，CPU 和 外围 I/O 不复位。用户程序执行的时候，flash 存储器一旦由 DMA 重写或者由 RAM 监视功能读取就马上产生引脚复位或内部复位的情况下，CPU 和外围 I/O 可能不正确复位。	p. 771 <input type="checkbox"/>
				当下列条件(a) 和 (b)满足时，由于暂停等原因仿真器(IECUBE, MINICUBE)操作停止的情况下，看门狗定时器 2 并不停止，并且产生复位或不可屏蔽中断。 当发生复位时，调试器挂起。 (a) 主时钟或副时钟用作看门狗定时器 2 的源时钟。 (b) 内部振荡时钟停止(RCM.RSTOP 位 = 1). 为避免这种情况，可进行如下两种处理。 • 当使用仿真器时，内部振荡时钟用作源时钟。 • 当使用仿真器时，内部振荡器不要停止。	p. 772 <input type="checkbox"/>
				当下列条件(a) 和 (b)满足时，由于暂停等原因仿真器(IECUBE, MINICUBE)操作停止的情况下，TMM 并不停止，即使外围暂停功能设置为“Break”。 (a) INTW, 内部振荡时钟 (fr/8), 或者选择副时钟用作 TMM 源时钟。 (b) 主时钟停止。 为避免这种情况，可进行如下两种处理 • 当使用仿真器时，主时钟 (fxx, fxx/2, fxx/4, fxx/64, fxx/512) 用作源时钟。 • 当使用仿真器时，禁止主时钟振荡。	p. 772 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 27 章	硬件	片上调试功能	注意事项 (DUC)	在片上调试模式下， DDO 引脚强制为高电平输出。	p. 772
			注意事项 (DUC 以外)	不要将用于调试的设备焊接在准备量产的产品上，因为在调试期间重写 flash 存储器并且不能保证对其重写的次数。而且，不要将调试监测程序写入会量产的产品中去。	p. 781
	软件			如果符合以下之一的条件，则不能执行强制中断。 <ul style="list-style-type: none">中断禁止(DI)用于 MINICUBE2 和目标设备之间通讯的串行接口的中断被屏蔽进入了等待模式后，但禁止可以释放等待模式的一个可屏蔽中断和目标设备之间的通讯模式是 UARTA0，且主时钟被 MINICUBE2 停止	p. 781
				如果符合如下的条件之一，则不执行 RRM 功能和 DMM 功能 <ul style="list-style-type: none">中断禁止(DI)用于 MINICUBE2 和目标设备之间通讯的串行接口的中断被屏蔽。进入了等待模式后，但禁止可以释放等待模式的一个可屏蔽中断MINICUBE2 和目标设备之间的通讯模式是 UARTA0，且主时钟已停止MINICUBE2 和目标设备之间的通讯模式是 UARTA0，而有一个与调试器不同的时钟用于通讯	p. 782
				如符合以下条件之一，则伪 RRM 和 DMM 功能可以释放等待模式。 <ul style="list-style-type: none">MINICUBE2 和目标设备之间的通讯模式为 CSIB0 或 CSIB3MINICUBE2 和目标设备之间的通讯模式是 UARTA0，且提供主时钟	p. 782
				需要一个特殊的外设 I/O 寄存器不能用 DMM 功能对其写操作	p. 782
				当调试器最初启动时，导入用于调试的监测程序对芯片的擦除和写入，但该操作会需要 12 个时钟。	p. 782
				当使用调试器来改变 CPU 操作时钟设置时，调试器重写监测程序。所需要的时间与如上（6）中所要求的一样。对于集成调试器 ID850QB，改变在结构对话框中的时钟设置时，会发生这样的操作。	p. 782
				如果一段分配了调试监测程序的空间由 flash 自编程改写，则调试器不再能够正常使用	p. 782
			安全 ID	在擦除 flash 存储区域后，在整个区域写入 1。	p. 783
第 28 章	硬件	电气特性	最大额定值	确保不要超过每个电源电压的最大额定值 (MAX.值)。	p. 786
				不要直接将 IC 产品的输出（或 I/O）引脚连接起来，或连接到 VDD，VCC，和 GND。漏极开路引脚或集电极开路引脚，可以直接互相连接。 如果输出引脚设置为高阻抗状态并且避免与外部电路的输出时序冲突，可以直接将 IC 产品的输出引脚与外部电路连接起来。	p. 787

章节	分类	功能	具体功能	注意事项	页码
第 28 章	硬件	电气特性	主时钟振荡器特性	DC 特性和 AC 特性中指出的额定值和条件是正常工作的质量保证。任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。	p. 787 <input type="checkbox"/>
				上面所列的振荡频率仅指振荡器特性。使用 V850ES/ JG2，内部工作条件不能超出 AC 特性和 DC 特性中规定的额定范围。	p. 789 <input type="checkbox"/>
				Flash 存储器启动所需要的时间。使用 OSTs 寄存器的可靠启动时间。	p. 789 <input type="checkbox"/>
				当使用主时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。 <ul style="list-style-type: none"> ● 连接线越短越好。 ● 连接线不应与其他信号线交叉。 ● 流经的电流变化较大的信号线不要在振荡器周围布线。 ● 要保持振荡器电容器的接地点电压与 Vss 相同。 ● 要将电容的地信号接入大电流地。 ● 不要从振荡器获取信号。 	p. 789 <input type="checkbox"/>
	软件			当主时钟停止副时钟工作时，在转换回主时钟前等待直到振荡稳定时间，由程序保证。	p. 789 <input type="checkbox"/>
	硬件		晶体谐振器 陶瓷谐振器	该振荡器常数是由振荡器生产商提供的在指定环境下评估得出的参考值。如果在实际应用中需要优化振荡器特性，请振荡器生产商在实验电路上评价。振荡器电压和频率仅为振荡器特性。使用 V850ES/JG2，内部工作条件不能超出 DC 特性和 AC 特性中规定的额定范围。	p. 790 <input type="checkbox"/>
			副时钟振荡器特性	上面所示的振荡频率仅为振荡器特性。因此使用 V850ES/JG2 内部工作条件不要超过 AC 特性和 DC 特性中所示的额定值。	p. 791 <input type="checkbox"/>
				当使用副时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。 <ul style="list-style-type: none"> ● 连接线越短越好。 ● 连接线不应与其他信号线交叉。 ● 流经的电流变化较大的信号线不要在振荡器周围布线。 ● 要保持振荡器电容器的接地点电压与 Vss 相同。 ● 不要将电容的地信号接入大电流地。 ● 要从振荡器获取信号。 	p. 791 <input type="checkbox"/>

章节	分类	功能	具体功能	注意事项	页码
第 28 章	硬件	电气特性	副时钟振荡器特性	为了缩减能耗,副时钟振荡器被设计为低幅电路,并且它比主时钟振荡器更容易受到噪声的影响。 因此在使用副时钟时更要注意布线方法。	p. 791
				关于谐振器的选择和振荡器常数,请客户自己评价或使用谐振器生产商的评价。	p. 791
			数据保持特性	转换到 STOP 模式和从 STOP 模式还原必须在额定的工作范围完成。	p. 796
			AC 特性	如果由于电路结构加载电容超过 50 pF,通过插入缓冲器或其他方法将设备的加载电容降到 50 pF 以下。	p. 797
	软件	I ² C 总线模式	在起始条件中,第一个时钟脉冲产生在保持时间后。	p. 815	
			系统内 SDA0n(在 SCL0n 信号的 VIHmin.)信号需要一个最少 300ns 的保持时间,以使在 SCL0n 的下降沿时占用未定义区域。	p. 815	
			如果系统不延长 SCL0n 信号低的保持时间(tLOW),仅需要满足最大数据保持时间(tHD: DAT)	p. 815	
			高速模式 I2C 总线可以用于正常模式 I2C 总线系统。在这种情况下,设定高速模式 I2C 总线,因此会遇到如下条件 ● 如果系统不延长 SCL0n 信号低的保持时间: tSU:DAT ≥ 250 ns ● 如果系统延长 SCL0n 信号低的保持时间: 发送如下数据位到 SDA0n 优先于 SCL0n 释放(tRmax. + tSU:DAT = 1,000 + 250 = 1,250 ns: 正常模式 I2C 总线规范).	p. 815	
			A/D 转换器	在 A/D 转换期间不要设置(读/写)复用端口;否则转换分辨率可能降低	p. 816
			编程特性	在出货后第一次对产品进行写操作时,“擦除 → 写”和“只写”作为一次重写。 示例 (P: 写入, E: 擦除) 出库产品 → P → E → P → E → P:3 次重写 出库产品 → E → P → E → P → E → P: 3 次重写	p. 820
第 30	硬件	推荐焊接条件	推荐焊接条件	不要一起使用不同的焊接方式(除局部加热外)。	p. 823
附录 A	软件	开发工具	RX850, RX850 Pro	购买 RX850 或 RX850 Pro 之前请先填写购买合同,签署许可协议。	p. 833
附录 C	软件	指令集列表	指令集	不要为通用寄存器 reg1 和 reg3.指定相同的寄存器。	p. 854

附录 E 修订历史

E.1 本版中主要修订之处

页码	描述
p. 217	修改 7.4 (3) TMPn I/O 控制寄存器 0 (TPnIOC0)
p. 239	增加 图 7-11 外部事件计数模式操作的寄存器设置 注意
p. 260	增加 图 7-22 单脉冲输出模式下的寄存器设置 注意
p. 305	增加 注意 8.4 (3) TMQ0 I/O 控制寄存器 0 (TQ0IOC0)
p. 330	增加 图 8-11 外部事件计数模式操作的寄存器设置 注意
p. 354	增加 图 8-22 单脉冲输出模式下的寄存器设置 注意
p. 418	修改 11.3 (1) 看门狗计时器模式寄存器 2 (WDTM2) 注意 4
p. 420	修改 11.3 (2) 看门狗计时器使能寄存器 (WDTE) 注意 3
p. 435	增加 表 13-2 正常转换模式下的转换时间选择 (ADA0HS1 Bit = 0) 注意 2
p. 436	增加 表 13-3 高速转换模式下的转换时间选择 (ADA0HS1 Bit = 1) 注意 2
p. 458	增加描述 13.6 (8) 读取 ADA0CRn 寄存器
p. 459	修改描述 13.6 (10) 高速转换模式
p. 459	增加 13.6 (11) A/D 转换时间
p. 476	修改 15.4 (1) UARTAn 控制寄存器 0 (UAnCTL0) 中对 UAnDIR 和 UAnCL 位的描述
p. 477	修改 15.4 (4) UARTAn 选项控制寄存器 0 (UAnOPT0) 中对 UAnSRF 位的描述
p. 481	修改描述 15.5 (1) 接收完成中断请求信号 (INTUAnR)
p. 487	增加 注意 1 和 2 到 15.6.4 SBF 接收
p. 510	修改 16.4 (1) CSIBn 控制寄存器 0 (CBnCTL0) 中的 注意
p. 517	增加 16.5 中断请求信号
pp. 518 到 550	修改 16.6 操作
p. 614	修改 表 17-6 等待周期
p. 617	修改 表 17-7 等待周期
p. 641	修改 18.3 (6) DMA 触发因子寄存器 0 到 3 (DTFR0 到 DTFR3) 中的 注意
p. 655	修改 18.13 (11) DMA 启动因子
p. 697	增加 注意 3 到 21.2 (1) 电源节省控制寄存器 (PSC)
p. 743	修改 表 26-2 基本功能
p. 743	修改 表 26-3 安全功能
p. 744	增加 表 26-4 安全设置
p. 746	修改 26.4.2 (1) UARTA0 中的传输率
p. 756	修改 表 26-7 Flash 内存控制命令
p. 763	修改 表 26-18 标准自编程流程
p. 766	修改 第 27 章 片上调试功能
p. 790	增加 (i) KYOCERA KINSEKI CORPORATION: Crystal resonator (TA = -40 to +85°C)
p. 790	增加 (ii) Murata Mfg. Co. Ltd.: Ceramic resonator (TA = -40 to +85°C)
p. 823	增加 第 30 章 推荐焊接条件
p. 824	增加 附录 A 开发工具
p. 855	增加 附录 D 注意事项列表
p. 891	增加 附录 E 修订历史

区域信息

本文档中的某些信息可能因国家不同而有所差异。用户在使用任何一种 NEC 产品之前，请与当地的 NEC 办事处联系，以获取权威的代理商和发行商信息。请验证以下内容：

- 设备的可用性
- 定货信息
- 产品发布进度表
- 相关技术资料的可用性
- 开发环境要求（例如：要求第三方工具和组件，主计算机，电源插头，AC 供电电源等）
- 网络要求

此外，对于商标、注册商标、出口限制条款和其他法律规定，不同的国家也有不同的要求。

详细信息请联系：

（中国区）

网址：

<http://www.cn.necel.com/>

<http://www.necel.com/>

[北京]

日电电子（中国）有限公司
中国北京市海淀区知春路 27 号
量子芯座 7，8，9，15 层
电话：(+86)10-8235-1155
传真：(+86)10-8235-7679

[深圳]

日电电子（中国）有限公司深圳分公司
深圳市福田区益田路卓越时代广场大厦 39 楼
3901，3902，3909 室
电话：(+86)755-8282-9800
传真：(+86)755-8282-9899

[上海]

日电电子（中国）有限公司上海分公司
中国上海市浦东新区银城中路 200 号
中银大厦 2409-2412 和 2509-2510 室
电话：(+86)21-5888-5400
传真：(+86)21-5888-5230

[香港]

香港日电电子有限公司
香港九龙旺角太子道西 193 号新世纪广场
第 2 座 16 楼 1601-1613 室
电话：(+852)2886-9318
传真：(+852)2886-9022
2886-9044

上海恩益禧电子国际贸易有限公司
中国上海市浦东新区银城中路 200 号
中银大厦 2511-2512 室
电话：(+86)21-5888-5400
传真：(+86)21-5888-5230