

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/JJ3

32 位单片机

硬件

μPD70F3743

μPD70F3744

μPD70F3745

μPD70F3746

文档编号 U18376CA3V0UD00 (第 3 版)

发布日期 2009 年 7 月 N

© NEC Electronics Corporation 2007

日本印刷

[备忘录]

关于CMOS器件的注意事项

① 输入引脚处的电压波形

输入噪声或由反射引起的波形失真可能导致故障发生。如果由于噪声等影响，使CMOS器件的输入电压范围处于在 V_{IL} (MAX) 和 V_{IH} (MIN) 之间，器件可能发生故障。在输入电平固定时以及输入电平从 V_{IL} (MAX) 到 V_{IH} (MIN) 的过渡期间，要谨防尖峰噪声影响器件。

② 未使用的输入引脚的处理

CMOS器件的输入端保持开路可能导致故障。如果一个输入引脚未被连接，则由于噪声等原因可能会产生内部输入电平，从而导致故障。CMOS器件的操作特性与双极性或NMOS器件不同。CMOS器件的输入电平必须借助上拉或下拉电路固定于高电平或低电平。每一个未使用引脚都应该连接到 V_{DD} 或 GND ，如果有可能作为输出引脚时，需要通过附加电阻连接到 V_{DD} 或 GND 。对未使用引脚的处理因器件而异，必须遵循与器件相关的规定和说明。

③ ESD防护措施

如果MOS器件周围有强电场，将会击穿氧化栅极，从而影响器件的工作。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。对于环境必须进行适当的控制。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。半导体器件的存放和运输必须使用抗静电容器、静电屏蔽袋或导电材料包装。所有包括工作台和工作面的测试和测量工具必须良好接地。操作员应当佩戴静电消除腕带以保证良好接地。不能用手直接接触半导体器件。对于装配有半导体器件的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器件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器件并没有被初始化。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。器件在收到复位信号后才进行初始化。具有复位功能的器件在上电后必须立即进行复位操作。

⑤ 电源开关顺序

一个器件内部工作和外部接口使用不同电源的情况下，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，可能会导致器件的内部组件过电压，产生异常电流，从而引起内部组件的误操作和性能的劣化。对每个器件电源的正确开关顺序，必须依据器件的规格说明分别进行判断。

⑥ 电源关闭状态下的输入信号

不要向没有加电的器件输入信号或提供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，从而引起器件的误操作，并产生异常电流，从而使内部组件劣化。每个器件电源关闭时的信号输入必须依据器件的规格说明分别进行判断。

注意事项：该产品使用的 SuperFlash® 科技授权来自 Silicon Storage Technology, Inc。

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。

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

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

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

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

SuperFlash 是 **Silicon Storage Technology, Inc.**在其它国家，包括美国和日本，注册的商标。

TRON 是实时操作系统内核（**The Realtime Operating System Nucleus**）的缩写。

ITRON 是工业 **TRON** 的缩写。

- 本文档信息发布于2009年7月。未来可能未经预先通知而进行更改。在实际进行生产设计时，请参阅各产品最新的数据规格书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货及其他信息。
- 未经本公司事先书面许可，禁止采用任何方式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做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，视音频设备，家电，加工机械，个人电气设备以及产业用机器人。

“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和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（如上定义）开发或制造的产品。

M5D 02.11-1

引言

读者 本手册适用于那些希望了解 V850ES/JJ3 单片机的功能，并使用 V850ES/JJ3 设计应用系统的用户。

目的 本手册适用于帮助用户了解下述编排组织中所述的 V850ES/JJ3 的硬件功能。

编排组织 本手册分为两个部分：硬件（本手册）和 架构（V850ES 架构用户手册）。

硬件	架构
<ul style="list-style-type: none">• 引脚功能• CPU 功能• 片上外设功能• Flash 存储器编程• 电气规范（目标）	<ul style="list-style-type: none">• 数据类型• 寄存器组• 指令格式和指令集• 中断和异常处理• 流水线操作

如何阅读本手册 假定本手册的读者具备电气工程、逻辑电路以及微控制器领域的常识。

要了解 V850ES/JJ3 单片机的全部功能。

→ 按照 **目录** 阅读本使用手册。

要查找已知名称的寄存器的详细情况

→ 请阅读 **附录 C 寄存器索引**

寄存器格式说明。

→ 位（其编号在各寄存器格式图的尖括（<>）内）名称在设备文件中定义为保留字。

要了解指令功能的详细情况

→ 分别参见可利用的 V850ES 架构用户手册。

要了解 V850ES/JJ3 单片机的电气规范。

→ 请参阅 **第二十九章 电气特性（目标）**

在本手册中，将“xxx 寄存器的 yyy 位”描述为“xxx.yyy 位”。请小心注意，如果在程序中描述“xxx.yyy”，编译程序/汇编程序则不能正确识别它。

标记 <R> 表示主要的修订内容。拷贝 PDF 文件中的“<R>”，并在“查找”区域内指定“<R>”，就可以很方便的搜索修订内容。

惯例	数据有效位:	高位在左, 低位在右 低电平有效表示:
	有效低电平表示:	$\overline{\text{xxx}}$ (在引脚或信号名称上有上划线)
	存储器映射地址:	高位地址在顶部, 低位地址在底部
	注:	正文中使用'注'标记的脚注
	注意事项:	需特别注意的信息
	备注:	补充信息
	数字表示法:	二进制 ... xxxx 或 xxxxB
		十进制 ... xxxx
		十六进制 ... xxxxB
	前缀表示 2 的幂 (地址空间、存储容量):	
		K (千): $2^{10} = 1,024$
		M (兆): $2^{20} = 1,024^2$
		G (千兆): $2^{30} = 1,024^3$

相关文档

本手册中的相关文档可能包括初稿版本。然而，初稿版本并未标注如此。

V850ES/JJ3 单片机相关文档

文档名称	文档编号
V850ES 体系结构用户手册	U15943E
V850ES/JJ3 硬件用户手册	本手册

开发工具的相关文档

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 具有编程功能的片上调试仿真器		U18371E
CA850 Ver. 3.00 C 编译程序包	操作篇	U17293E
	C 语言篇	U17291E
	汇编语言	U17292E
	连接指南	U17294E
PM+ Ver. 6.20 项目管理器		U17990E
ID850QB Ver. 3.20 集成调试器	操作篇	U17964E
SM850 Ver. 2.50 系统仿真器	操作篇	U16218E
SM850 Ver. 2.00 或更新的系统仿真器	外部组件的用户开放接口规范	U14873E
SM+系统仿真器	操作篇	U17246E
	用户开放界面	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 导言	19
1.1 概述	19
1.2 特性	21
1.3 应用领域	22
1.4 订购信息	22
1.5 引脚配置（顶视图）	23
1.6 功能框图结构	25
1.6.1 内部框图	25
1.6.2 内部单元	26
第二章 引脚功能	29
2.1 引脚功能列表	29
2.2 引脚状态	40
2.3 引脚 I/O 电路类型、I/O 缓冲器电源、及未使用引脚的连接	41
2.4 注意事项	45
第三章 CPU 功能	46
3.1 特性	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	57
3.4.3 存储器映射	58
3.4.4 存储区域	60
3.4.5 地址空间的使用建议	65
3.4.6 外围 I/O 寄存器	68
3.4.7 专用寄存器	80
3.4.8 注意事项	84
第四章 端口功能	87
4.1 特性	87
4.2 端口基本配置	87
4.3 端口配置	88
4.3.1 端口 0	93
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 6	110
4.3.7 端口 7	114
4.3.8 端口 8	116
4.3.9 端口 9	118

4.3.10	端口 CD	126
4.3.11	端口 CM	127
4.3.12	端口 CS	129
4.3.13	端口 CT	131
4.3.14	端口 DH	133
4.3.15	端口 DL	135
4.4	框图	138
4.5	使用复用功能时的端口寄存器设置	170
4.6	注意事项	179
4.6.1	设置端口引脚的注意事项	179
4.6.2	端口 n 寄存器 (Pn) 位操作指令的注意事项	182
4.6.3	片上调试引脚的注意事项	183
4.6.4	P05/INTP2/DRST 引脚的注意事项	183
4.6.5	电源开启时 P53 引脚的注意事项	183
4.6.6	迟滞特性	183
第五章	总线控制功能	184
5.1	特性	184
5.2	总线控制引脚	185
5.2.1	访问片内 ROM 存储器、片内 RAM 存储器或片上外围输入/输出端口时的引脚状态	185
5.2.2	各种工作模式下的引脚状态	185
5.3	存储模块功能	186
5.4	外部总线接口模式控制功能	187
5.5	总线访问	188
5.5.1	总线访问时钟数目	188
5.5.2	总线宽度设置功能	188
5.5.3	根据不同总线宽度进行访问	189
5.6	等待功能	196
5.6.1	可编程等待功能	196
5.6.2	外部等待功能	197
5.6.3	可编程等待和外部等待之间的关系	198
5.6.4	可编程地址等待功能	199
5.7	空闲状态插入功能	200
5.8	总线保持功能	201
5.8.1	功能概述	201
5.8.2	总线保持步骤	202
5.8.3	节电模式时的操作	202
5.9	总线优先权	203
5.10	总线时序	204
第六章	时钟发生功能	210
6.1	概述	210
6.2	配置	211
6.3	寄存器	213
6.4	操作	218
6.4.1	各时钟的操作	218
6.4.2	时钟输出功能	218
6.5	PLL 功能	219
6.5.1	概述	219

6.5.2	寄存器.....	219
6.5.3	用法	222
第七章	16 位定时器/事件计数器 P (TMP)	223
7.1	概述	223
7.2	功能	223
7.3	配置	224
7.4	寄存器.....	226
7.5	操作	238
7.5.1	间隔定时器模式 (TPnMD2 至 TPnMD0 位=000)	239
7.5.2	外部事件计数模式 (TPnMD2 至 TPnMD0 位=001)	249
7.5.3	外部触发脉冲输出模式 (TPnMD2 至 TPnMD0 位= 010)	257
7.5.4	单次触发脉冲输出模式 (TPnMD2 至 TPnMD0 位=011)	269
7.5.5	PWM 输出模式 (TPnMD2 至 TPnMD0 位=100)	276
7.5.6	自由运行定时器模式 (TPnMD2 至 TPnMD0 位=101)	285
7.5.7	脉冲宽度测量模式 (TPnMD2 至 TPnMD0 位=110)	302
7.5.8	定时器输出操作	308
7.6	选择器功能	309
7.7	注意事项	311
第八章	16 位定时器/事件计数器 Q (TMQ)	312
8.1	概述	312
8.2	功能	312
8.3	配置	313
8.4	寄存器.....	315
8.5	操作	331
8.5.1	间隔定时器模式 (TQ0MD2 至 TQ0MD0 位=000)	332
8.5.2	外部事件计数模式 (TQ0MD2 至 TQ0MD0 位=001)	341
8.5.3	外部触发脉冲输出模式 (TQ0MD2 至 TQ0MD0 位=010)	350
8.5.4	单次触发脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 011)	363
8.5.5	PWM 输出模式 (TQ0MD2 至 TQ0MD0 位=100)	372
8.5.6	自由运行定时器模式 (TQ0MD2 至 TQ0MD0 位=101)	383
8.5.7	脉冲宽度测量模式 (TQ0MD2 至 TQ0MD0 位=110)	403
8.5.8	定时器输出操作	409
8.6	注意事项	410
第九章	16 位间隔定时器 M (TMM)	411
9.1	概述	411
9.2	配置	412
9.3	寄存器.....	413
9.4	操作	414
9.4.1	间隔计时器模式	414
9.4.2	注意事项	418
第十章	钟表定时器功能	419
10.1	功能	419
10.2	配置	420
10.3	控制寄存器	422
10.4	操作	426

10.4.1	作为钟表定时器.....	426
10.4.2	作为间隔定时器.....	427
10.4.3	注意事项	428
第十一章	看门狗定时器 2 的功能.....	429
11.1	功能.....	429
11.2	配置.....	430
11.3	寄存器	431
11.4	操作.....	433
第十二章	实时输出功能（RTO）	434
12.1	功能.....	434
12.2	配置.....	435
12.3	寄存器	437
12.4	操作.....	439
12.5	用法.....	440
12.6	注意事项	440
第十三章	A/D 转换器	441
13.1	概述.....	441
13.2	功能.....	441
13.3	配置.....	442
13.4	寄存器	445
13.5	操作.....	456
13.5.1	基本操作	456
13.5.2	转换操作时序	457
13.5.3	触发模式	458
13.5.4	操作模式	460
13.5.5	掉电比较模式	464
13.6	注意事项	469
13.7	A/D 转换器特征表的阅读方法.....	473
第十四章	D/A 转换器	477
14.1	功能.....	477
14.2	配置.....	477
14.3	寄存器	478
14.4	操作.....	480
14.4.1	正常模式下的操作	480
14.4.2	实时输出模式下的操作	480
14.4.3	注意事项	481
第十五章	异步串行接口 A（UARTA）	482
15.1	UARTA 和其它串行接口的模式转换.....	482
15.1.1	CSIB4 和 UARTA0 的模式转换	482
15.1.2	UARTA2 和 I ² C00 的模式转换	483
15.1.3	UARTA1 和 I ² C02 的模式转换.....	484
15.2	特性.....	485
15.3	配置.....	486
15.4	寄存器	488

15.5	中断请求信号	494
15.6	操作	495
15.6.1	数据格式	495
15.6.2	SBF 发送/接收格式	497
15.6.3	SBF 发送	499
15.6.4	SBF 接收	500
15.6.5	UART 发送	502
15.6.6	连续发送过程	503
15.6.7	UART 接收	505
15.6.8	接收错误	506
15.6.9	奇偶类型和操作	508
15.6.10	接收数据噪声过滤器	509
15.7	专用波特率发生器	510
15.8	注意事项	518
第十六章	3-线可变长度串行 I/O (CSIB)	519
16.1	CSIB 和其它串行接口的模式转换	519
16.1.1	CSIB4 和 UAR _{TA0} 的模式转换	519
16.1.2	CSIB0 和 I ² C01 的模式转换	520
16.2	特性	520
16.3	配置	521
16.4	寄存器	523
16.5	中断请求信号	530
16.6	操作	531
16.6.1	单次传输模式 (主机模式, 发送模式)	531
16.6.2	单次传输模式 (主机模式, 接收模式)	533
16.6.3	单次传输模式 (主机模式, 发送/接收模式)	535
16.6.4	单次传输模式 (从机模式, 发送模式)	537
16.6.5	单次传输模式 (从机模式, 接收模式)	539
16.6.6	单次传输模式 (从机模式, 发送/接收模式)	541
16.6.7	连续传输模式 (主机模式, 发送模式)	543
16.6.8	连续传输模式 (主机模式, 接收模式)	545
16.6.9	连续传输模式 (主机模式, 发送/接收模式)	548
16.6.10	连续传输模式 (从机模式, 发送模式)	552
16.6.11	连续传输模式 (从机模式, 接收模式)	554
16.6.12	连续传输模式 (从机模式, 发送/接收模式)	557
16.6.13	接收错误	561
16.6.14	时钟时序	562
16.7	输出引脚	564
16.8	波特率发生器	565
16.8.1	波特率的产生	566
16.9	注意事项	567
第十七章	I ² C 总线	568
17.1	I ² C 总线和其它串行接口的模式转换	568
17.1.1	UAR _{TA2} 和 I ² C00 模式转换	568
17.1.2	CSIB0 和 I ² C01 模式转换	569
17.1.3	UAR _{TA1} 和 I ² C02 模式转换	570
17.2	特性	571

17.3	配置.....	572
17.4	寄存器.....	576
17.5	I ² C 总线模式功能.....	592
17.5.1	引脚配置.....	592
17.6	I ² C总线的定义和控制方法.....	593
17.6.1	开始条件.....	593
17.6.2	地址.....	594
17.6.3	传输方向指示.....	595
17.6.4	ACK.....	596
17.6.5	停止条件.....	597
17.6.6	等待状态.....	598
17.6.7	等待状态取消方法.....	600
17.7	I ² C中断请求信号 (INTIICn).....	601
17.7.1	主设备操作.....	601
17.7.2	从设备操作 (当接收从设备地址数据 (地址匹配) 时).....	604
17.7.3	从设备操作 (当接收扩展码时).....	608
17.7.4	不通信时的操作.....	612
17.7.5	仲裁失败操作 (仲裁失败后作为从设备操作).....	612
17.7.6	发生仲裁失败时的操作 (在仲裁失败后不进行通信).....	614
17.8	中断请求信号 (INTIICn) 的产生时序和等待控制.....	621
17.9	地址匹配检测方法.....	623
17.10	错误检测.....	623
17.11	扩展代码.....	623
17.12	仲裁.....	624
17.13	唤醒功能.....	625
17.14	通信保留.....	626
17.14.1	使能通信保留功能 (IICFn.IICRSVn 位 = 0) 时.....	626
17.14.2	禁止通信保留功能时 (IICFn.IICRSVn 位 = 1).....	630
17.15	注意事项.....	631
17.16	通信操作.....	632
17.16.1	主设备工作于单主系统.....	633
17.16.2	主设备工作于多主系统.....	634
17.16.3	从设备操作.....	637
17.17	数据通信的时序.....	640
第十八章	DMA 功能 (DMA 控制器).....	647
18.1	特性.....	647
18.2	配置.....	648
18.3	寄存器.....	649
18.4	传输目标.....	656
18.5	传输模式.....	657
18.6	传输类型.....	657
18.7	DMA 通道优先级.....	658
18.8	与 DMA 传输有关的时间.....	658
18.9	DMA 传输启动因素.....	659
18.10	DMA 中止因素.....	660
18.11	DMA 传输结束.....	660
18.12	操作时序.....	660

18.13	注意事项	665
第十九章	中断/异常处理功能	670
19.1	特性 670	
19.2	不可屏蔽中断	674
19.2.1	操作	676
19.2.2	恢复	677
19.2.3	NP 标志	678
19.3	可屏蔽中断	679
19.3.1	操作	679
19.3.2	恢复	681
19.3.3	可屏蔽中断的优先级	682
19.3.4	中断控制寄存器 (xxICn)	686
19.3.5	中断屏蔽寄存器 0 至 4 (IMR0 至 IMR4)	688
19.3.6	正在服务中优先级寄存器 (ISPR)	690
19.3.7	ID 标志	691
19.3.8	看门狗定时器模式寄存器 2 (WDTM2)	691
19.4	软件异常	692
19.4.1	操作	692
19.4.2	恢复	693
19.4.3	EP 标志	694
19.5	异常陷阱	695
19.5.1	非法操作码定义	695
19.5.2	调试陷阱	697
19.6	外部中断请求输入引脚 (NMI 和 INTP0 至 INTP8)	699
19.6.1	噪声消除	699
19.6.2	边沿检测	699
19.7	CPU 中断响应时间	705
19.8	中断不被 CPU 响应期间	706
19.9	注意事项	706
第二十章	按键中断功能	707
20.1	功能	707
20.2	寄存器	708
20.3	注意事项	708
第二十一章	待机功能	709
21.1	概述	709
21.2	寄存器	711
21.3	HALT 模式	714
21.3.1	设置和操作状态	714
21.3.2	解除 HALT 模式	714
21.4	IDLE1 模式	716
21.4.1	设置和操作状态	716
21.4.2	解除 IDLE1 模式	716
21.5	IDLE2 模式	718
21.5.1	设置和操作状态	718
21.5.2	释放 IDLE2 模式	718
21.5.3	解除 IDLE2 模式时确保建立时间	719

21.6	STOP 模式	721
21.6.1	设置和操作状态	721
21.6.2	释放 STOP 模式	721
21.6.3	在释放 STOP 模式时确保振荡稳定时间	724
21.7	副时钟操作模式	725
21.7.1	设置和操作状态	725
21.7.2	解除副时钟操作模式	725
21.8	副 IDLE 模式	727
21.8.1	设置和操作状态	727
21.8.2	解除副 IDLE 模式	727
第二十二章	复位功能	729
22.1	概述	729
22.2	检查复位源的寄存器	730
22.3	操作	731
22.3.1	通过 RESET 引脚的复位操作	731
22.3.2	通过看门狗定时器 2 的复位操作	733
22.3.3	通过低电压检测器的复位操作	735
22.3.4	复位解除后的操作	736
22.3.5	复位功能操作流程	737
第二十三章	时钟监控器	738
23.1	功能	738
23.2	配置	738
23.3	寄存器	739
23.4	操作	740
第二十四章	低电压检测器 (LVI)	743
24.1	功能	743
24.2	配置	743
24.3	寄存器	744
24.4	操作	746
24.4.1	使用内部复位信号	746
24.4.2	用作中断	747
24.5	RAM 保持电压检测操作	748
24.6	仿真功能	749
第二十五章	CRC 功能	750
25.1	功能	750
25.2	配置	750
25.3	寄存器	751
25.4	操作	752
25.5	使用方法	753
第二十六章	稳压器	755
26.1	概述	755
26.2	操作	756
第二十七章	Flash 存储器	757

27.1	特性	757
27.2	存储器配置	758
27.3	功能概述	760
27.4	通过专用 flash 编程器进行重写	763
27.4.1	编程环境	763
27.4.2	通信模式	764
27.4.3	Flash 存储器控制	770
27.4.4	通信模式的选择	771
27.4.5	通信命令	772
27.4.6	引脚连接	773
27.5	通过自编程进行重写	777
27.5.1	概述	777
27.5.2	特性	778
27.5.3	标准自编程流程	779
27.5.4	Flash 功能	780
27.5.5	引脚处理	780
27.5.6	用到的内部资源	781
第二十八章	片上调试功能	782
28.1	用 DCU 调试	783
28.1.1	连接电路示例	783
28.1.2	接口信号	783
28.1.3	可屏蔽功能	785
28.1.4	寄存器	785
28.1.5	操作	787
28.1.6	注意事项	787
28.2	不使用 DCU 调试	788
28.2.1	电路连接示例	788
28.2.2	可屏蔽功能	789
28.2.3	用户资源的保留	790
28.2.4	注意事项	796
28.3	ROM 安全功能	798
28.3.1	安全 ID	798
28.3.2	设置	799
第二十九章	电气规范	801
第三十章	封装图	836
第三十一章	推荐焊接条件	837
附录 A	开发工具	838
A.1	软件包	840
A.2	语言处理软件	840
A.3	控制软件	840
A.4	调试工具（硬件）	841
A.4.1	使用 IECUBE® QB-V850ESSX2 时	841
A.4.2	使用 MINICUBE QB-V850MINI 时	843
A.4.3	使用 MINICUBE2 QB-MINI2 时	844

A.5	调试工具（软件）	845
A.6	嵌入式软件.....	846
A.7	Flash 存储器写入工具	847
附录 B	V850ES/JJ3 和 V850ES/JJ2 之间的主要区别	848
附录 C	寄存器索引	850
附录 D	指令集列表	862
D.1	习惯用法	862
D.2	指令集 (按字母先后次序)	865
附录 E	注意事项列表	872
附录 F	修订历史	909
F.1	本版的主要修订	909
F.2	上一版的修订历史	910

第一章 导言

V850ES/JJ3 是 NEC 电子 V850 单片微控制器系列产品中的一员，专门针对实时控制应用的低功耗而设计。

1.1 概述

V850ES/JJ3 是一款 32 位单片微控制器，它包括 V850ES CPU 内核以及相关外设功能，例如 ROM/RAM、定时器/计数器、串行接口、A/D 转换器以及 D/A 转换器。

除了高速实时响应特性和单时钟基本指令之外，V850ES/JJ3 还支持乘法指令、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，这些指令由硬件乘法器来实现。此外，作为一种实时控制系统，V850ES/JJ3 还能够在超低功耗应用中实现卓越的性价比，例如家庭音响、打印机以及数字家电。

表 1-1 列出了 V850ES/JJ3 系列产品。

表 1-1. V850ES/JJ3 产品列表

部件编号		μPD70F3743	μPD70F3744	μPD70F3745	μPD70F3746
内部存储器器	Flash 存储器	384 KB	512 KB	768 KB	1024 KB
	RAM	32 KB	40 KB	60 KB	60 KB
存储空间	逻辑空间	64 MB			
	外部存储区	16 MB			
外部总线接口		地址总线： 24 位 地址数据总线： 8/16 位 支持复用总线模式/分离总线模式 片选信号： 4			
通用寄存器		32 位×32 个寄存器			
主时钟（振荡频率）		陶瓷/晶体 （PLL 模式下： f _x = 2.5 至 5 MHz（4 倍频）或 f _x = 2.5 至 4 MHz（8 倍频）， 在时钟直通模式下： f _x = 2.5 至 10 MHz）			
副时钟（振荡频率）		晶体（f _{XT} = 32.768 kHz）			
内部振荡器		f _R = 220 kHz（TYP）			
最短指令执行时间		31.25 ns（主时钟（f _{XX} ） = 32 MHz）			
DSP 功能		32 × 32 = 64: 125 至 156.25 ns (32 MHz 时) 32 × 32 + 32 = 32: 187.5 ns (32 MHz 时) 16 × 16 = 32: 31.25 至 62.5 ns (32 MHz)时 16 × 16 + 32 = 32: 93.75 ns (32 MHz 时)			
I/O 端口		I/O: 128（5 V 耐压/N 沟道漏极开路输出可选： 60）			
定时器	16 位定时器/事件计数器 P： 9 通道				
	16 位定时器/事件计数器 Q： 1 通道				
	16 位间隔定时器 M： 1 通道				
	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 × 2 通道			
A/D 转换器		10 位分辨率 × 16 通道			
D/A 转换器		8 位分辨率 × 2 通道			
串行接口		UART： 1 通道 UART/CSI： 1 通道 UART/I ² C 总线： 2 通道 CSI： 4 通道 CSI/I ² C 总线： 1 通道			
DMA 控制器		4 通道（传输目标： 片上外设 I/O， 内部 RAM， 外部存储器）			
中断源		外部： 10（10） [※] ， 内部： 61			
省电功能		HALT/IDLE1/IDLE2/STOP/副时钟/sub-IDLE 模式			
复位源		RESET 引脚输入， 看门狗定时器 2（WDT2）， 时钟监视器（CLM）， 低压检测器（LVI）			
DCU		提供（运行/断点）			
工作电源电压		2.85 至 3.6 V			
运行环境温度		-40 至+85°C			
封装		144 针塑封 LQFP（密间距）（20 × 20 mm）			

注 括号中的数字表示能够解除 STOP 模式的外部中断数量。

1.2 特性

- 最短指令执行周期: 31.25 ns (运行于 32MHz 的主时钟 (f_{xx}))
- 通用寄存器: 32 位×32 个寄存器
- CPU 特性:
 - 带符号乘法 (16 × 16 → 32): 1 至 2 个时钟。
 - 带符号乘法 (32 × 32 → 64): 1 至 5 个时钟。
 - 溢出处理 (包括上、下溢出检测功能)。
 - 32 位移位指令: 1 个时钟
 - 位操作指令。
 - 长/短两种格式的数据装载/存储指令。
- 存储空间: 64 MB 线性地址空间 (用于程序和数据)
 - 外部扩展: 最多 16 MB (包括用作内部 ROM/RAM 的 1 MB)
 - 内部存储器:
 - RAM: 32 KB/40 KB/60 KB (参见表 1-1)
 - Flash 存储器: 384 KB/512 KB/768 KB/1024 KB (参见表 1-1)
 - 外部总线接口:
 - 分离总线/复用总线输出
 - 8 位/16 位数据总线功能
 - 等待功能
 - 可编程等待功能。
 - 外部等待功能
 - 空闲状态功能
 - 总线保持功能
- 中断和异常处理:
 - 不可屏蔽中断: 2 个中断源
 - 可屏蔽中断: 60 个中断源
 - 软件异常: 32 个中断源
 - 异常陷阱: 2 个中断源
- I/O 线: I/O 端口: 128
- 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): 9 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 × 2 通道
- 串行接口:
 - 异步串行接口 A (UARTA)
 - 3 线长度可变串行接口 B (CSIB)
 - I²C 总线接口 (I²C)
 - UARTA/CSIB: 1 通道
 - UARTA/I²C: 2 通道
 - CSIB/I²C: 1 通道
 - CSIB: 4 通道
 - UARTA: 1 通道
- A/D 转换器: 10 位分辨率: 16 通道
- D/A 转换器: 8 位分辨率: 2 通道
- DMA 控制器: 4 通道
- CRC 功能: 可以为 8 位数据单元产生 16 位错误检测码
- DCU (调试控制单元): JTAG 接口
- 时钟发生器:
 - 在主时钟或副时钟操作期间
 - 有 7 个级别 CPU 时钟可选 (f_{xx}, f_{xx}/2, f_{xx}/4, f_{xx}/8, f_{xx}/16, f_{xx}/32, f_{XT})
 - 时钟直通模式/PLL 模式可选

- 内部振荡时钟: 220 kHz (典型值)
- 省电功能: HALT/IDLE1/IDLE2/STOP/副时钟/sub-IDLE 模式
- 封装: 144 针塑封 LQFP (密间距) (20 × 20)

1.3 应用领域

家庭音响、打印机、数字家电以及其它消费设备。

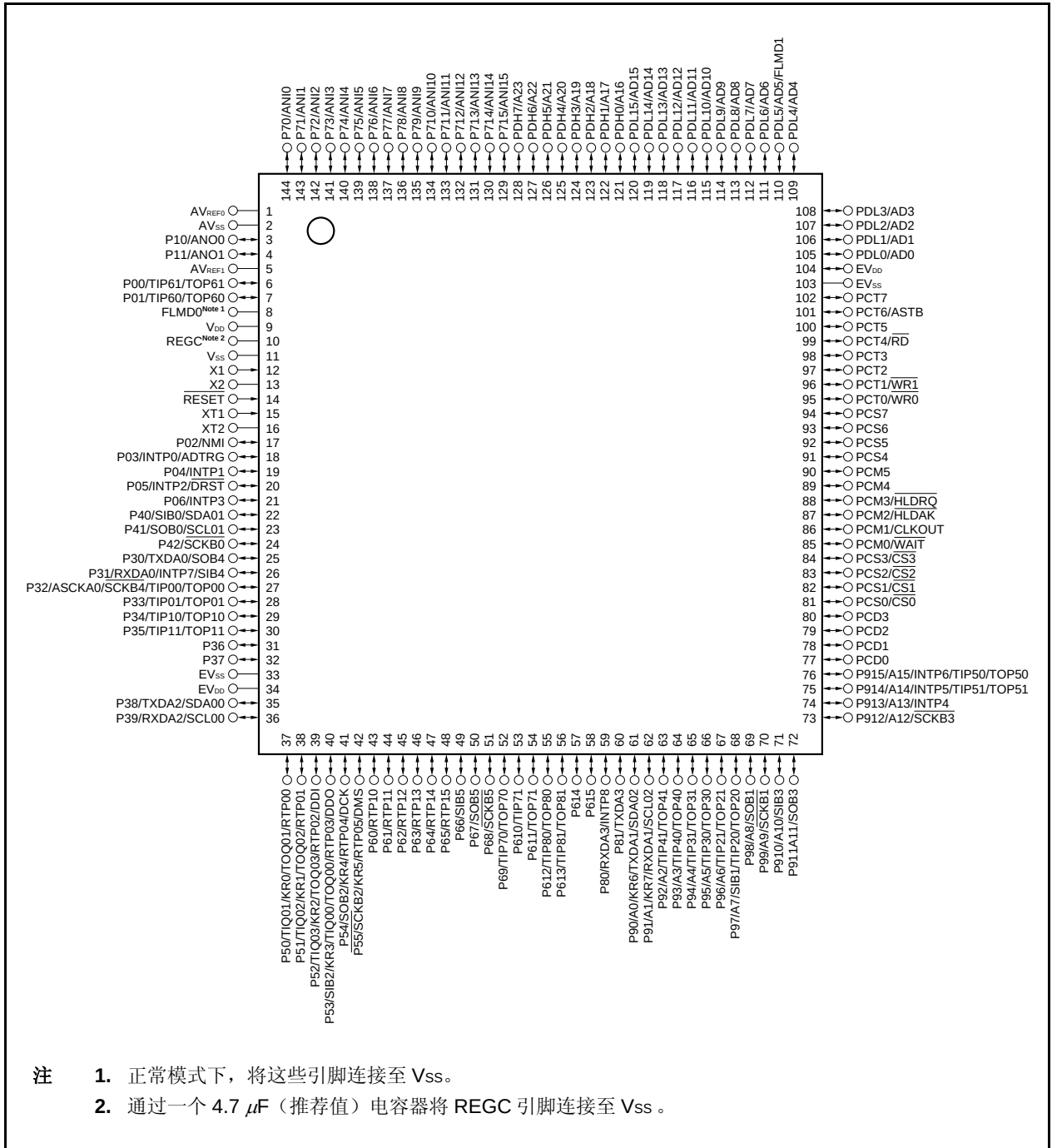
1.4 订购信息

部件编号	封装	内部 Flash 存储器
μPD70F3743GJ-GAE-AX	144 针塑封 LQFP (密间距) (20 × 20)	384 KB
μPD70F3744GJ-GAE-AX	144 针塑封 LQFP (密间距) (20 × 20)	512 KB
μPD70F3745GJ-GAE-AX	144 针塑封 LQFP (密间距) (20 × 20)	768 KB
μPD70F3746GJ-GAE-AX	144 针塑封 LQFP (密间距) (20 × 20)	1024 KB

备注 V850ES/JJ3 微控制器是无铅产品。

1.5 引脚配置 (顶视图)

144 针塑封 LQFP (密间距) (20 × 20)

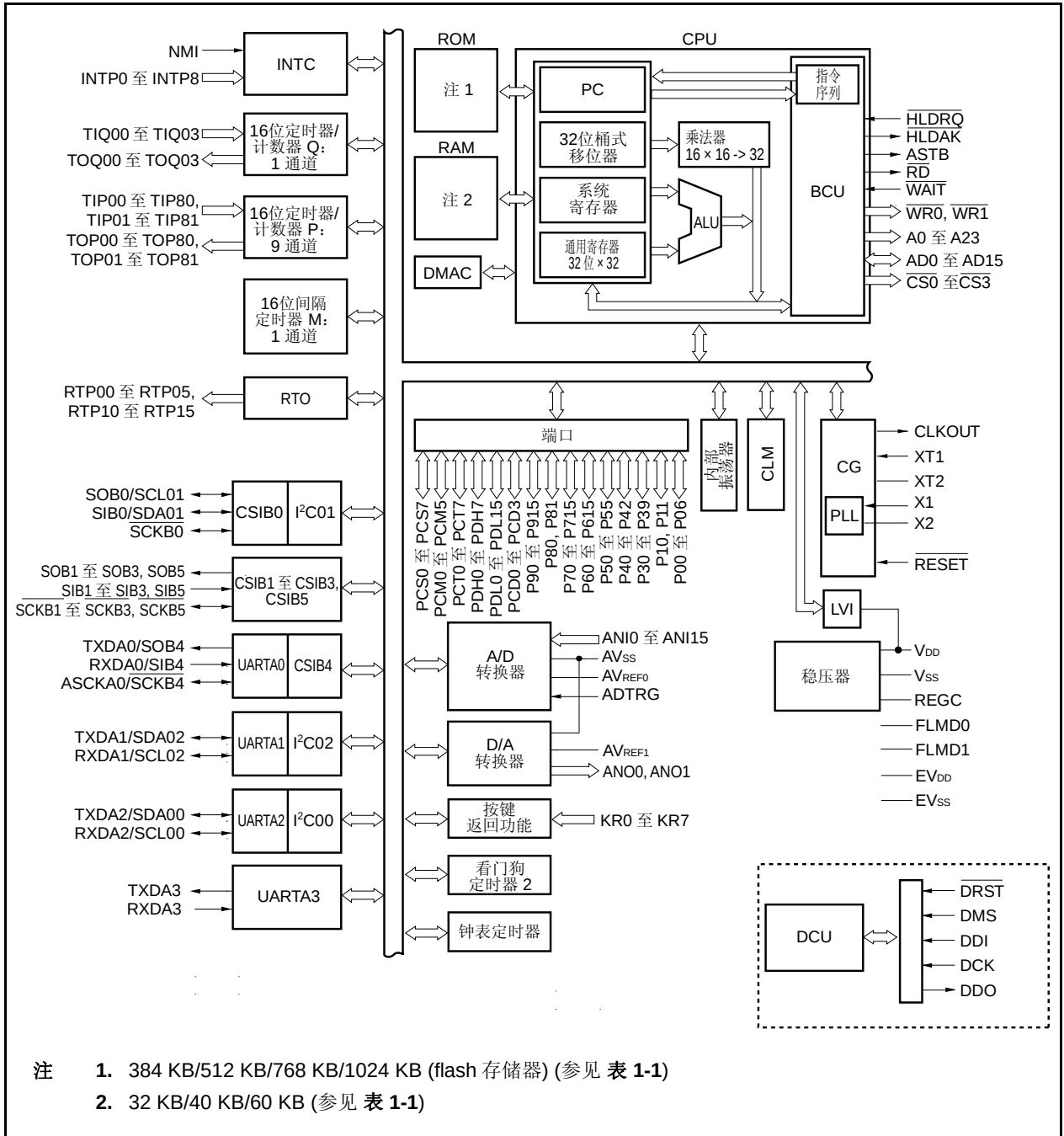
 μ PD70F3743GJ-GAE-AX μ PD70F3744GJ-GAE-AX μ PD70F3745GJ-GAE-AX μ PD70F3746GJ-GAE-AX

引脚名称

A0 至 A23:	地址总线	$\overline{\text{RD}}$:	读选通
AD0 至 AD15:	地址/数据总线	REGC:	稳压器控制
ADTRG:	A/D 触发信号输入	$\overline{\text{RESET}}$:	复位
ANI0 至 ANI15:	模拟输入	RTP00 至 RTP05:	
ANO0, ANO1:	模拟输出	RTP10 至 RTP15:	实时输出端口
ASCKA0:	异步串行时钟	RXDA0 至 RXDA3:	接收数据
ASTB:	地址选通	$\overline{\text{SCKB0}}$ 至 $\overline{\text{SCKB5}}$:	串行时钟
AVREF0, AVREF1:	模拟参考电压	SCL00 至 SCL02:	串行时钟
AVss:	模拟 Vss	SDA00 至 SDA02:	串行数据
CLKOUT:	时钟输出	SIB0 至 SIB5:	串行输入
$\overline{\text{CS0}}$ 至 $\overline{\text{CS3}}$:	片选	SOB0 至 SOB5:	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:	外部引脚的电源	TIP50, TIP51,	
EVss:	外部引脚接地	TIP60, TIP61,	
FLMD0, FLMD1:	Flash 编程模式	TIP70, TIP71,	
$\overline{\text{HLD\!AK}}$:	保持应答	TIP80, TIP81,	
$\overline{\text{HLDRQ}}$:	保持请求	TIQ00 至 TIQ03:	定时器输入
INTP0 至 INTP8:	外部中断输入	TOP00, TOP01,	
KR0 至 KR7:	按键返回	TOP10, TOP11,	
NMI:	不可屏蔽中断请求	TOP20, TOP21,	
P00 至 P06:	端口 0	TOP30, TOP31,	
P10, P11:	端口 1	TOP40, TOP41,	
P30 至 P39:	端口 3	TOP50, TOP51,	
P40 至 P42:	端口 4	TOP60, TOP61,	
P50 至 P55:	端口 5	TOP70, TOP71,	
P60 至 P615:	端口 6	TOP80, TOP81,	
P70 至 P715:	端口 7	TOQ00 至 TOQ03:	定时器输出
P80, P81:	端口 8	TXDA0 至 TXDA3:	发送数据
P90 至 P915:	端口 9	VDD:	电源
PCD0 至 PCD3:	端口 CD	Vss:	地
PCM0 至 PCM5:	端口 CM	$\overline{\text{WAIT}}$:	等待
PCS0 至 PCS7:	端口 CS	$\overline{\text{WR0}}$:	低字节写选通
PCT0 至 PCT7:	端口 CT	$\overline{\text{WR1}}$:	高字节写选通
PDH0 至 PDH7:	端口 DH	X1, X2:	主时钟晶振
PDL0 至 PDL15:	端口 DL	XT1, XT2:	副时钟晶振

1.6 功能框图结构

1.6.1 内部框图



1.6.2 内部单元

(1) CPU

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

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

(2) 总线控制单元（BCU）

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

(3) ROM

这是一个 1024/768/512/384 KB flash 存储器，其映射地址为 0000000H 至 00FFFFFFH/0000000H 至 00BFFFFH/0000000H 至 007FFFFH/0000000H 至 005FFFFH。取指令期间，CPU 可在一个时钟周期内访问。

(4) RAM

这是一个 60/40/32 KB RAM，其映射地址为 3FF0000H 至 3FFFFFFFFH/3FF5000H 至 3FFFFFFFFH/3FF7000H 至 3FFFFFFFFH。在数据访问期间，CPU 可在一个时钟周期内完成访问。

(5) 中断控制器（INTC）

该控制器用来处理片上外设和外部硬件产生的硬件中断请求（NMI、INTP0 至 INTP8）。可以为这些中断请求指定 8 个级别的中断优先级，同时也能执行多重服务控制。

(6) 时钟发生器（CG）

提供一个主时钟振荡器产生主时钟振荡频率（fx）和一个副时钟振荡器产生副时钟振荡频率（fxT）。在时钟直通模式中，fx 直接用作主时钟频率（fxx）。在 PLL 模式下，使用 fx 的 4 倍频或 8 倍频。

CPU 工作频率（fCPU）可以从七种类型中选择：fxx，fxx/2，fxx/4，fxx/8，fxx/16，fxx/32 和 fxT。

(7) 内部振荡器

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

(8) 定时器/计数器

芯片提供了 9 个通道的 16 位定时器/事件计数器 P（TMP），一个通道的 16 位定时器/事件计数器 Q（TMQ）和一个通道的 16 位间隔定时器 M（TMM）。

(9) 钟表定时器

该定时器通过对时钟（副时钟提供的 32.768 kHz 或自预分频器 3 提供的 32.768 kHz fBRG）计数来计算参考时间周期（0.5 s）。钟表定时器也能够用作主时钟的间隔定时器。

(10) 看门狗定时器 2

片内集成的看门狗定时器用于检测无意中形成的程序死循环、系统异常等。

内部振荡时钟、主时钟、或副时钟都可以选为源时钟。

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

(11) 串行接口

V850ES/JJ3 包含三种串行接口：异步串行接口A（UARTA）、3 线长度可变串行接口B（CSIB）、和I²C总线接口（I²C）。

使用 UARTA，数据经由 TXDA0 引脚至 TXDA3 引脚和 RXDA0 引脚至 RXDA3 引脚进行传输

使用 CSIB，数据经由 SOB0 引脚至 SOB5 引脚、SIB0 引脚至 SIB5 引脚、和 SCKB0 引脚至 SCKB5 引脚进行传输。

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

(12) A/D 转换器

10 位 A/D 转换器包含 16 个模拟输入引脚。使用逐次逼近的方式进行转换。

(13) D/A 转换器

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

(14) DMA 控制器

芯片提供了一个 4 通道的 DMA 转换器。根据片上外围 I/O 发出的中断请求，该控制器在内部 RAM 和片上外围 I/O 设备之间进行数据传输。

(15) 按键中断功能

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

(16) 实时输出功能

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

(17) CRC 功能

CRC 操作电路可以根据 8 位数据单元产生 16 位 CRC（循环冗余检测）码。

(18) DCU（调试控制单元）

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

(19) 端口

通用寄存器端口功能和控制引脚功能列举如下：

端口	I/O	复用功能
P0	7 位 I/O	定时器 I/O、NMI、外部中断、A/D 转换器触发、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
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
P6	16 位 I/O	实时输出、串行接口、定时器 I/O
P7	16 位 I/O	A/D 转换器模拟输入
P8	2 位 I/O	串行接口、外部中断
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
PCD	4 位 I/O	—
PCM	6 位 I/O	外部控制信号
PCS	8 位 I/O	片选输出
PCT	8 位 I/O	外部控制信号
PDH	8 位 I/O	外部地址总线
PDL	16 位 I/O	外部地址/数据总线

第二章 引脚功能

2.1 引脚功能列表

下表描述了 V850ES/JJ3 单片机中引脚的名称和功能。

有三种类型的引脚 I/O 缓冲器电源：AV_{REF0}、AV_{REF1}、和 EV_{DD}。这些电源和引脚的关系如下：

表 2-1. 引脚 I/O 缓冲器电源

电源	对应引脚
AV _{REF0}	端口 7
AV _{REF1}	端口 1
EV _{DD}	RESET, 端口 0、3 至 6、8、9, CD、CM、CS、CT、DH、DL

(1) 端口引脚

(1/4)

引脚名称	引脚编号	I/O	功能	复用功能
P00	6	I/O	端口 0 7 位 I/O 端口 可按位指定输入/输出。 可按位指定 N 沟道漏极开路输出。 5 V 耐压。	TIP61/TOP61
P01	7			TIP60/TOP60
P02	17			NMI
P03	18			INTP0/ADTRG
P04	19			INTP1
P05 ^注	20			INTP2/ $\overline{\text{DRST}}$
P06	21			INTP3
P10	3	I/O	端口 1 2 位 I/O 端口 可按位指定输入/输出。	ANO0
P11	4			ANO1
P30	25	I/O	端口 3 10 位 I/O 端口 可按位指定输入/输出。 可按位指定 N 沟道漏极开路输出。 5 V 耐压。	TXDA0/SOB4
P31	26			RXDA0/INTP7/SIB4
P32	27			ASCKA0/SCKB4/TIP00/TOP00
P33	28			TIP01/TOP01
P34	29			TIP10/TOP10
P35	30			TIP11/TOP11
P36	31			—
P37	32			—
P38	35			TXDA2/SDA00
P39	36			RXDA2/SCL00
P40	22	I/O	端口 4 3 位 I/O 端口 可按位指定输入/输出。 可按位指定 N 沟道漏极开路输出。 5 V 耐压。	SIB0/SDA01
P41	23			SOB0/SCL01
P42	24			$\overline{\text{SCKB0}}$
P50	37	I/O	端口 5 6 位 I/O 端口 可按位指定输入/输出。 可按位指定 N 沟道漏极开路输出。 5 V 耐压。	TIQ01/KR0/TOQ01/RTP00
P51	38			TIQ02/KR1/TOQ02/RTP01
P52	39			TIQ03/KR2/TOQ03/RTP02/DDI
P53	40			SIB2/KR3/TIQ00/TOQ00/RTP03/DDO
P54	41			SOB2/KR4/RTP04/DCK
P55	42			$\overline{\text{SCKB2}}$ /KR5/RTP05/DMS

注 内置一个下拉电阻器。将 OCDM.OCDM0 位清除为 0，可断开该电阻的连接。

(2/4)

引脚名称	引脚编号	I/O	功能	复用功能
P60	43	I/O	端口 6 16 位 I/O 端口 可按位指定输入/输出。 可按位指定 N 沟道漏极开路输出。 5 V 耐压。	RTP10
P61	44			RTP11
P62	45			RTP12
P63	46			RTP13
P64	47			RTP14
P65	48			RTP15
P66	49			SIB5
P67	50			SOB5
P68	51			SCKB5
P69	52			TIP70/TOP70
P610	53			TIP71
P611	54			TOP71
P612	55			TIP80/TOP80
P613	56			TIP81/TOP81
P614	57			—
P615	58			—
P70	144	I/O	端口 7 16 位 I/O 端口 可按位指定输入/输出。	ANI0
P71	143			ANI1
P72	142			ANI2
P73	141			ANI3
P74	140			ANI4
P75	139			ANI5
P76	138			ANI6
P77	137			ANI7
P78	136			ANI8
P79	135			ANI9
P710	134			ANI10
P711	133			ANI11
P712	132			ANI12
P713	131			ANI13
P714	130			ANI14
P715	129			ANI15
P80	59	I/O	端口 8 2 位 I/O 端口 可按位指定输入/输出。 可按位指定 N 沟道漏极开路输出。 5 V 耐压。	RXDA3/INTP8
P81	60			TXDA3

(3/4)

引脚名称	引脚编号	I/O	功能	复用功能
P90	61	I/O	端口 9 16 位 I/O 端口 可按位指定输入/输出。 可按位指定 N 沟道漏极开路输出。 5 V 耐压。	A0/KR6/TXDA1/SDA02
P91	62			A1/KR7/RXDA1/SCL02
P92	63			A2/TIP41/TOP41
P93	64			A3/TIP40/TOP40
P94	65			A4/TIP31/TOP31
P95	66			A5/TIP30/TOP30
P96	67			A6/TIP21/TOP21
P97	68			A7/SIB1/TIP20/TOP20
P98	69			A8/SOB1
P99	70			A9/SCKB1
P910	71			A10/SIB3
P911	72			A11/SOB3
P912	73			A12/SCKB3
P913	74			A13/INTP4
P914	75			A14/INTP5/TIP51/TOP51
P915	76			A15/INTP6/TIP50/TOP50
PCD0	77	I/O	端口 CD 4 位 I/O 端口 可按位指定输入/输出。	—
PCD1	78			—
PCD2	79			—
PCD3	80			—
PCM0	85	I/O	端口 CM 6 位 I/O 端口 可按位指定输入/输出。	WAIT
PCM1	86			CLKOUT
PCM2	87			HLDK
PCM3	88			HLDRQ
PCM4	89			—
PCM5	90			—
PCS0	81	I/O	端口 CS 8 位 I/O 端口 可按位指定输入/输出。	CS0
PCS1	82			CS1
PCS2	83			CS2
PCS3	84			CS3
PCS4	91			—
PCS5	92			—
PCS6	93			—
PCS7	94			—

引脚名称	引脚编号	I/O	功能	复用功能
PCT0	95	I/O	端口 CT 8 位 I/O 端口 可按位指定输入/输出。	$\overline{WR0}$
PCT1	96			$\overline{WR1}$
PCT2	97			—
PCT3	98			—
PCT4	99			$\overline{RD}$
PCT5	100			—
PCT6	101			ASTB
PCT7	102			—
PDH0	121	I/O	端口 DH 8 位 I/O 端口 可按位指定输入/输出。	A16
PDH1	122			A17
PDH2	123			A18
PDH3	124			A19
PDH4	125			A20
PDH5	126			A21
PDH6	127			A22
PDH7	128			A23
PDL0	105	I/O	端口 DL 16 位 I/O 端口 可按位指定输入/输出。	AD0
PDL1	106			AD1
PDL2	107			AD2
PDL3	108			AD3
PDL4	109			AD4
PDL5	110			AD5/FLMD1
PDL6	111			AD6
PDL7	112			AD7
PDL8	113			AD8
PDL9	114			AD9
PDL10	115			AD10
PDL11	116			AD11
PDL12	117			AD12
PDL13	118			AD13
PDL14	119			AD14
PDL15	120			AD15

(2) 非端口引脚

(1/6)

引脚名称	引脚编号	I/O	功能	复用功能
A0	61	输出	外部存储器的地址总线 (当使用分离总线时) 可以选择 N 沟道漏极开路输出。 5 V 耐压。	P90/KR6/TXDA1/SDA02
A1	62			P91/KR7/RXDA1/SCL02
A2	63			P92/TIP41/TOP41
A3	64			P93/TIP40/TOP40
A4	65			P94/TIP31/TOP31
A5	66			P95/TIP30/TOP30
A6	67			P96/TIP21/TOP21
A7	68			P97/SIB1/TIP20/TOP20
A8	69			P98/SOB1
A9	70			P99/SCKB1
A10	71			P910/SIB3
A11	72			P911/SOB3
A12	73			P912/SCKB3
A13	74			P913/INTP4
A14	75			P914/INTP5/TIP51/TOP51
A15	76			P915/INTP6/TIP50/TOP50
A16	121	输出	外部存储器的地址总线	PDH0
A17	122			PDH1
A18	123			PDH2
A19	124			PDH3
A20	125			PDH4
A21	126			PDH5
A22	127			PDH6
A23	128			PDH7
AD0	105	I/O	外部存储器的地址总线/数据总线	PDL0
AD1	106			PDL1
AD2	107			PDL2
AD3	108			PDL3
AD4	109			PDL4
AD5	110			PDL5/FLMD1
AD6	111			PDL6
AD7	112			PDL7
AD8	113			PDL8
AD9	114			PDL9
AD10	115			PDL10
AD11	116			PDL11
AD12	117			PDL12
AD13	118			PDL13
AD14	119			PDL14
AD15	120			PDL15
ADTRG	18	输入	A/D 转换器外部触发输入。5 V 耐压。	P03/INTP0

引脚名称	引脚编号	I/O	功能	复用功能
ANI0	144	输入	A/D 转换器的模拟电压输入	P70
ANI1	143			P71
ANI2	142			P72
ANI3	141			P73
ANI4	140			P74
ANI5	139			P75
ANI6	138			P76
ANI7	137			P77
ANI8	136			P78
ANI9	135			P79
ANI10	134			P710
ANI11	133			P711
ANI12	132			P712
ANI13	131			P713
ANI14	130			P714
ANI15	129			P715
ANO0	3	输出	D/A 转换器的模拟电压输出	P10
ANO1	4			P11
ASCKA0	27	输入	UARTA0 波特率时钟输入。5 V 耐压。	P32/SCKB4/TIP00/TOP00
ASTB	101	输出	用于外部存储器的地址选通信号输出	PCT6
AV _{REF0}	1	—	用于 A/D 转换器的参考电压输入，用于端口 7 的正极电源供应	—
AV _{REF1}	5		用于 D/A 转换器的参考电压输入，用于端口 1 的正极电源供应	—
AV _{SS}	2	—	用于 A/D 转换器 D/A 转换器的地电位（与 V _{SS} 电位相同）	—
CLKOUT	86	输出	内部系统时钟输出	PCM1
$\overline{\text{CS0}}$	81	输出	片选输出	PCS0
$\overline{\text{CS1}}$	82			PCS1
$\overline{\text{CS2}}$	83			PCS2
$\overline{\text{CS3}}$	84			PCS3
DCK	41	输入	调试时钟输入。5 V 耐压。	P54/SOB2/KR4/RTP04
DDI	39	输入	调试数据输入。5 V 耐压。	P52/TIQ03/KR2/TOQ03/RTP02
DDO [*]	40	输出	调试数据输出。 N 沟道漏极开路输出选择。5 V 耐压。	P53/SIB2/KR3/TIQ00/TOQ00/ RTP03
DMS	42	输入	调试模式选择输入。5 V 耐压。	P55/SCKB2/KR5/RTP05
$\overline{\text{DRST}}$	20	输入	调试复位输入。5 V 耐压。	P05/INTP2
EV _{DD}	34, 104	—	用于外部的正极电源供应（与 V _{DD} 电位相同）	—
EV _{SS}	33, 103	—	用于外部的地电位（与 V _{SS} 电位相同）	—
FLMD0	8	输入	Flash 存储器编程模式设置引脚	—
FLMD1	110			PDL5/AD5

注 在片上调试模式下，强行置为高电平输出。

引脚名称	引脚编号	I/O	功能	复用功能
H $\overline{\text{LDAK}}$	87	输出	总线保持应答输出	PCM2
H $\overline{\text{LDRQ}}$	88	输入	总线保持请求输入	PCM3
INTP0	18	输入	外部中断请求输入（可屏蔽，模拟噪声消除）。 可以为 INTP3 引脚选择模拟噪声消除或数字噪声消除。 5 V 耐压。	P03/ADTRG
INTP1	19			P04
INTP2	20			P05/ $\overline{\text{DRST}}$
INTP3	21			P06
INTP4	74			P913/A13
INTP5	75			P914/A14/TIP51/TOP51
INTP6	76			P915/A15/TIP50/TOP50
INTP7	26			P31/RXDA0/SIB4
INTP8	59			P80/RXDA3
KR0 ^{注1}	37	输入	按键中断输入（片上模拟噪声消除器） 5 V 耐压。	P50/TIQ01/TOQ01/RTP00
KR1 ^{注1}	38			P51/TIQ02/TOQ02/RTP01
KR2 ^{注1}	39			P52/TIQ03/TOQ03/RTP02/DDI
KR3 ^{注1}	40			P53/SIB2/TIQ00/TOQ00/RTP03/DDO
KR4 ^{注1}	41			P54/SOB2/RTP04/DCK
KR5 ^{注1}	42			P55/ $\overline{\text{SCKB2}}$ /RTP05/DMS
KR6 ^{注1}	61			P90/A0/TXDA1/SDA02
KR7 ^{注1}	62			P91/A1/RXDA1/SCL02
NMI ^{注2}	17	输入	外部中断输入（不可屏蔽，模拟噪声消除）。 5 V 耐压。	P02
$\overline{\text{RD}}$	99	输出	外部存储器的读取选通信号输出	PCT4
REGC	10	—	稳压器输出稳定电容的连接（4.7 μF （推荐值））	—
$\overline{\text{RESET}}$	14	输入	系统复位输入	—
RTP00	37	输出	实时输出端口 可以选择 N 沟道漏极开路输出 5 V 耐压。	P50/TIQ01/KR0/TOQ01
RTP01	38			P51/TIQ02/KR1/TOQ02
RTP02	39			P52/TIQ03/KR2/TOQ03/DDI
RTP03	40			P53/SIB2/KR3/TIQ00/TOQ00/DDO
RTP04	41			P54/SOB2/KR4/DCK
RTP05	42			P55/ $\overline{\text{SCKB2}}$ /KR5/DMS
RTP10	43			P60
RTP11	44			P61
RTP12	45			P62
RTP13	46			P63
RTP14	47			P64
RTP15	48			P65

- 注
1. 外部连接上拉引脚。
 2. NMI 的引脚复用功能是 P02 引脚。在复位之后作为 P02 引脚功能。要使能 NMI 引脚功能，设置 PMC0.PMC02 位为 1。NMI 引脚的初始设置是“无边沿检测”。通过 INTF0 和 INTRO 寄存器来选择 NMI 引脚的有效边沿。

引脚名称	引脚编号	I/O	功能	复用功能
RXDA0	26	输入	串行接收数据输入 (UARTA0 至 UARTA3) 5 V 耐压。	P31/INTP7/SIB4
RXDA1	62			P91/A1/KR7/SCL02
RXDA2	36			P39/SCL00
RXDA3	59			P80/INTP8
SCKB0	24	I/O	串行时钟 I/O (CSIB0 至 CSIB5) 可以选择 N 沟道漏极开路输出。 5 V 耐压。	P42
SCKB1	70			P99/A9
SCKB2	42			P55/KR5/RTP05/DMS
SCKB3	73			P912/A12
SCKB4	27			P32/ASCKA0/TIP00/TOP00
SCKB5	51			P68
SCL00	36	I/O	串行时钟 I/O (I ² C00 至 I ² C02) 可以选择 N 沟道漏极开路输出。 5 V 耐压。	P39/RXDA2
SCL01	23			P41/SOB0
SCL02	62			P91/A1/KR7/RXDA1
SDA00	35	I/O	串行发送/接收数据 I/O (I ² C00 至 I ² C02) 可以选择 N 沟道漏极开路输出。 5 V 耐压。	P38/TXDA2
SDA01	22			P40/SIB0
SDA02	61			P90/A0/KR6/TXDA1
SIB0	22	输入	串行接收数据输入 (CSIB0 至 CSIB5) 5 V 耐压。	P40/SDA01
SIB1	68			P97/A7/TIP20/TOP20
SIB2	40			P53/KR3/TIQ00/TOQ00/RTP03/DDO
SIB3	71			P910/A10
SIB4	26			P31/RXDA0/INTP7
SIB5	49			P66
SOB0	23	输出	串行发送数据输出 (CSIB0 至 CSIB5) 可以选择 N 沟道漏极开路输出。 5 V 耐压。	P41/SCL01
SOB1	69			P98/A8
SOB2	41			P54/KR4/RTP04/DCK
SOB3	72			P911/A11
SOB4	25			P30/TXDA0
SOB5	50			P67
TIP00	27	输入	外部事件计数输入/捕获触发输入/外部触发输入 (TMP0), 5 V 耐压。	P32/ASCKA0/SCKB4/TOP00
TIP01	28		捕获触发输入 (TMP0)。5 V 耐压。	P33/TOP01
TIP10	29		外部事件计数输入/捕获触发输入/外部触发输入 (TMP1), 5 V 耐压。	P34/TOP10
TIP11	30		捕获触发输入 (TMP1)。5 V 耐压。	P35/TOP11
TIP20	68		外部事件计数输入/捕获触发输入/外部触发输入 (TMP2), 5 V 耐压。	P97/A7/SIB1/TOP20
TIP21	67		捕获触发输入 (TMP2)。5 V 耐压。	P96/A6/TOP21
TIP30	66		外部事件计数输入/捕获触发输入/外部触发输入 (TMP3), 5 V 耐压。	P95/A5/TOP30
TIP31	65		捕获触发输入 (TMP3)。5 V 耐压。	P94/A4/TOP31
TIP40	64		外部事件计数输入/捕获触发输入/外部触发输入 (TMP4), 5 V 耐压。	P93/A3/TOP40

引脚名称	引脚编号	I/O	功能	复用功能
TIP41	63	输入	捕获触发输入 (TMP4)。5 V 耐压。	P92/A2/TOP41
TIP50	76		外部事件计数输入/捕获触发输入/外部触发输入 (TMP5)，5 V 耐压。	P915/A15/INTP6/TOP50
TIP51	75		捕获触发输入 (TMP5)。5 V 耐压。	P914/A14/INTP5/TOP51
TIP60	7		外部事件计数输入/捕获触发输入/外部触发输入 (TMP6)，5 V 耐压。	P01/TOP60
TIP61	6		捕获触发输入 (TMP6)。5 V 耐压。	P00/TOP61
TIP70	52		外部事件计数输入/捕获触发输入/外部触发输入 (TMP7)，5 V 耐压。	P69/TOP70
TIP71	53		捕获触发输入 (TMP7)。5 V 耐压。	P610
TIP80	55		外部事件计数输入/捕获触发输入/外部触发输入 (TMP8)，5 V 耐压。	P612/TOP80
TIP81	56		捕获触发输入 (TMP8)。5 V 耐压。	P613/TOP81
TIQ00	40		外部事件计数输入/捕获触发输入/外部触发输入 (TMQ0)，5 V 耐压。	P53/SIB2/KR3/TOQ00/RTP03/DDO
TIQ01	37		捕获触发输入 (TMQ0)，5 V 耐压。	P50/KR0/TOQ01/RTP00
TIQ02	38			P51/KR1/TOQ02/RTP01
TIQ03	39			P52/KR2/TOQ03/RTP02/DDI
TOP00	27	输出	定时器输出 (TMP0) 可以选择 N 沟道漏极开路输出选择，5 V 耐压。	P32/ASCKA0/SCKB4/TIP00
TOP01	28			P33/TIP01
TOP10	29		定时器输出 (TMP1) 可以选择 N 沟道漏极开路输出选择，5 V 耐压。	P34/TIP10
TOP11	30			P35/TIP11
TOP20	68		定时器输出 (TMP2) 可以选择 N 沟道漏极开路输出选择，5 V 耐压。	P97/A7/SIB1/TIP20
TOP21	67			P96/A6/TIP21
TOP30	66		定时器输出 (TMP3) 可以选择 N 沟道漏极开路输出选择，5 V 耐压。	P95/A5/TIP30
TOP31	65			P94/A4/TIP31
TOP40	64		定时器输出 (TMP4) 可以选择 N 沟道漏极开路输出选择，5 V 耐压。	P93/A3/TIP40
TOP41	63			P92/A2/TIP41
TOP50	76		定时器输出 (TMP5) 可以选择 N 沟道漏极开路输出选择，5 V 耐压。	P915/A15/INTP6/TIP50
TOP51	75			P914/A14/INTP5/TIP51
TOP60	7		定时器输出 (TMP6) 可以选择 N 沟道漏极开路输出选择，5 V 耐压。	P01/TIP60
TOP61	6			P00/TIP61
TOP70	52		定时器输出 (TMP7) 可以选择 N 沟道漏极开路输出选择，5 V 耐压。	P69/TIP70
TOP71	54			P611
TOP80	55		定时器输出 (TMP8) 可以选择 N 沟道漏极开路输出选择，5 V 耐压。	P612/TIP80
TOP81	56			P613/TIP81
TOQ00	40	输出	定时器输出 (TMQ0) 可以选择 N 沟道漏极开路输出选择，5 V 耐压。	P53/SIB2/KR3/TOQ00/RTP03/DDO
TOQ01	37			P50/TOQ01/KR0/RTP00
TOQ02	38			P51/RTP01/KR1/TOQ02
TOQ03	39			P52/TOQ03/KR2/RTP02/DDI
TXDA0	25	输出	串行发送数据输出 (UARTA0 至 UARTA3) 可以选择 N 沟道漏极开路输出选择。 5 V 耐压。	P30/SOB4
TXDA1	61			P90/A0/KR6/SDA02
TXDA2	35			P38/SDA00
TXDA3	60			P81

(6/6)

引脚名称	引脚编号	I/O	功能	复用功能
V _{DD}	9	—	内部电源正极引脚	—
V _{SS}	11	—	内部的地电平	—
WAIT	85	输入	外部等待输入	PCM0
WR0	95	输出	用于外部存储器的写入选通（低 8 位）	PCT0
WR1	96		用于外部存储器的写入选通（高 8 位）	PCT1
X1	12	输入	用于主时钟振荡电路的连接	—
X2	13	—		—
XT1	15	输入	用于副时钟振荡电路的连接	—
XT2	16	—		—

2.2 引脚状态

各种模式下的引脚工作状态描述如下：

表 2-2. 各种模式下的引脚工作状态

引脚名称	接通电源时 ^{註 1}	复位期间（接通电源除外）	HALT模式 ^{註 2}	IDLE1、IDLE2、副IDLE 模式 ^{註 2}	STOP 模式 ^{註 2}	空闲状态 ^{註 3}	总线保持	
P05/DRST	下拉	下拉 ^{註 4}	保持	保持	保持	保持	保持	
P10/ANO0, P11/ANO1	高阻	高阻	保持	保持	高阻	保持	保持	
P53/DDO	不确定	高阻 ^{註 5}	保持	保持	保持	保持	保持	
AD0 至 AD15	高阻 ^{註 6}	高阻 ^{註 6}	註 7, 8	高阻	高阻	保持	高阻	
A0 至 A15			不确定 ^{註 7, 9}					
A16 至 A23			不确定 ^{註 7}					
WAIT			—	—	—	—	—	
CLKOUT			运行	L	L	运行	运行	
WR0, WR1			H ^{註 7}	H	H	H	高阻	
RD								
ASTB								
HLDAK			运行 ^{註 7}					
HLDRQ				—	—	—	运行	
CS0 至 CS3				H ^{註 7}	H	H	保持	Hi-Z
其它端口引脚	高阻	高阻	保持	保持	保持	保持	保持	

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

备注

Hi-Z: 高阻

保持: 保持上一个外部总线周期的状态。

L: 低电平输出

H: 高电平输出

—: 没有采样输入（不响应）

2.3 引脚 I/O 电路类型、I/O 缓冲器电源、及未使用引脚的连接

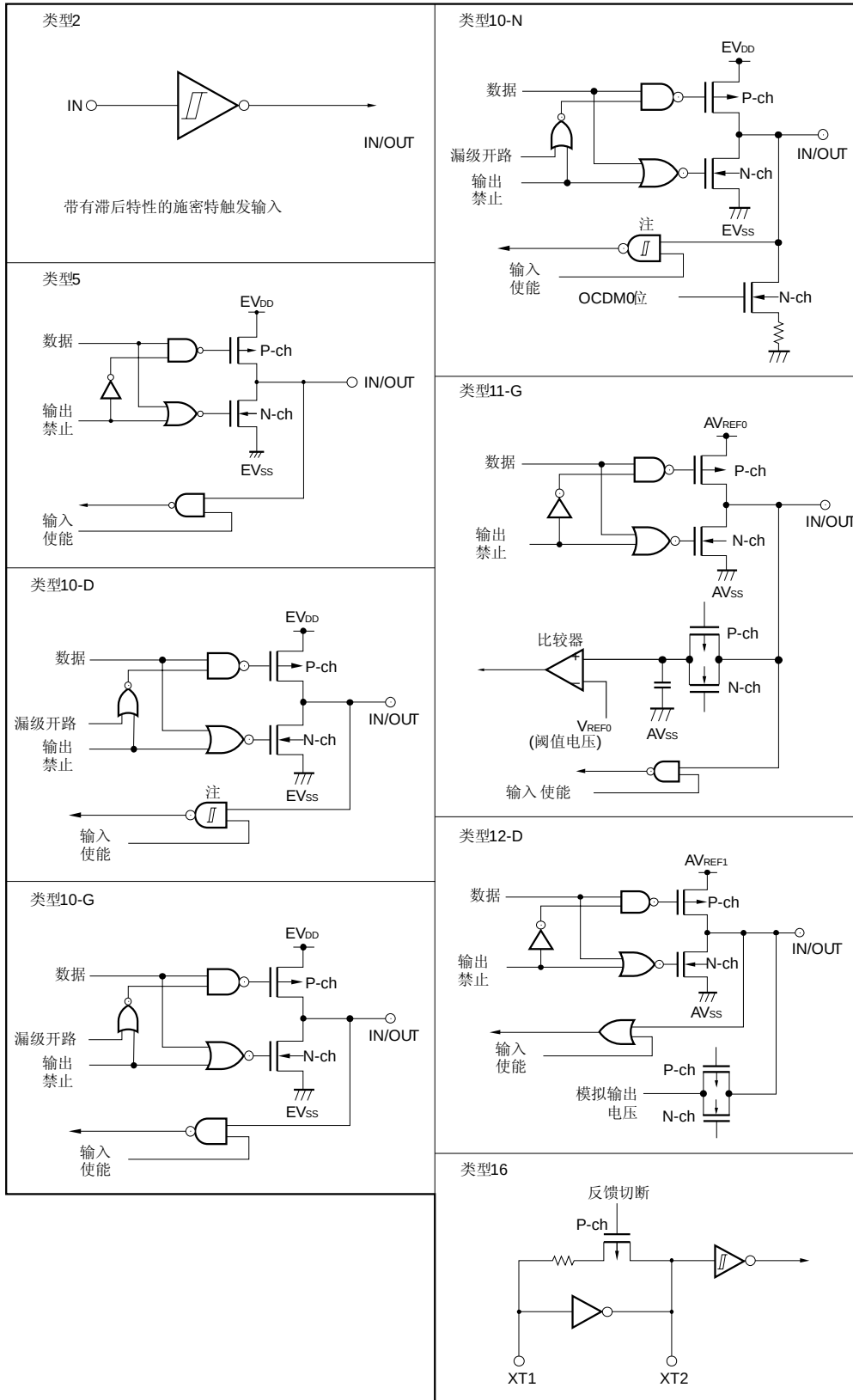
(1/3)

引脚	复用功能	引脚编号	I/O 电路类型	推荐的连接方式
P00	TIP61/TOP61	6	10-D	输入： 通过电阻单独连接至 EV _{DD} 或 EV _{SS} 输出： 保持开路。
P01	TIP60/TOP60	7		
P02	NMI	17		
P03	INTP0/ADTRG	18		
P04	INTP1	19		
P05	INTP2/ $\overline{\text{DRST}}$	20	10-N	输入： 通过电阻单独连接至 EV _{SS} 。 禁止固定连接至 V _{DD} 电平。 输出： 保持开路。 通过 $\overline{\text{RESET}}$ 引脚复位后内部下拉。
P06	INTP3	21	10-D	输入： 通过电阻单独连接至 EV _{DD} 或 EV _{SS} 输出： 保持开路。
P10, P11	ANO0, ANO1	3, 4	12-D	输入： 通过电阻单独连接至 AV _{REF1} 或 AV _{SS} 输出： 保持开路。
P30	TXDA0/SOB4	25	10-G	输入： 通过电阻单独连接至 EV _{DD} 或 EV _{SS} 输出： 保持开路。
P31	RXDA0/INTP7/SIB4	26	10-D	
P32	ASCKA0/ $\overline{\text{SCKB4}}$ /TIP00	27		
P33	TIP01/TOP01	28		
P34	TIP10/TOP10	29		
P35	TIP11/TOP11	30		
P36	—	31	10-G	
P37	—	32		
P38	TXDA2/SDA00	35	10-D	
P39	RXDA2/SCL00	36		
P40	SIB0/SDA01	22		
P41	SOB0/SCL01	23		
P42	$\overline{\text{SCKB0}}$	24		
P50	TIQ01/KR0/TOQ01/RTP00	37		
P51	TIQ02/KR1/TOQ02/RTP01	38		
P52	TIQ03/KR2/TOQ03/RTP02/DDI	39		
P53	SIB2/KR3/TIQ00/TOQ00/RTP03/ DDO	40		
P54	SOB2/KR4/RTP04/DCK	41		
P55	$\overline{\text{SCKB2}}$ /KR5/RTP05/DMS	42		

引脚	复用功能	引脚编号	I/O 电路类型	推荐的连接方式
P60 至 P65	RTP10 至 RTP15	43 至 48	10-G	输入： 通过电阻单独连接至 EV _{DD} 或 EV _{SS} 输出： 保持开路。
P66	SIB5	49	10-D	
P67	SOB5	50	10-G	
P68	SCKB5	51	10-D	
P69	TIP70/TOP70	52		
P610	TIP71	53		
P611	TOP71	54	10-G	
P612	TIP80/TOP80	55	10-D	
P613	TIP81/TOP81	56		
P614, P615	—	57, 58	10-G	
P70 至 P715	ANI0 至 ANI15	144 至 129	11-G	输入： 通过电阻单独连接至 AV _{REF0} 或 AV _{SS} 输出： 保持开路。
P80	RXDA3, INTP8	59	10-D	输入： 通过电阻单独连接至 EV _{DD} 或 EV _{SS} 输出： 保持开路。
P81	TXDA3	60	10-G	
P90	A0/KR6/TDXA1/SDA02	61	10-D	
P91	A1/KR7/RXDA1/SCL02	62		
P92	A2/TIP41/TOP41	63		
P93	A3/TIP40/TOP40	64		
P94	A4/TIP31/TOP31	65		
P95	A5/TIP30/TOP30	66		
P96	A6/TIP21/TOP21	67		
P97	A7/SIB1/TIP20/TOP20	68		
P98	A8/SOB1	69	10-G	
P99	A9/SCKB1	70	10-D	
P910	A10/SIB3	71		
P911	A11/SOB3	72	10-G	
P912	A12/SCKB3	73	10-D	
P913	A13/INTP4	74		
P914	A14/INTP5/TIP51/TOP51	75		
P915	A15/INTP6/TIP50/TOP50	76		
PCD0 至 PCD3	—	77 至 80	5	
PCM0	WAIT	85		
PCM1	CLKOUT	86		
PCM2	HLDKAK	87		
PCM3	HLDRQ	88		
PCM4, PCM5	—	89, 90		
PCS0 至 PCS3	CS0 至 CS3	81 至 84		
PCS4 至 PCS7	—	91 至 94		

引脚	复用功能	引脚编号	I/O 电路类型	推荐的连接方式
PCT0, PCT1	WR0, WR1	95, 96	5	输入：通过电阻单独连接至 EV _{DD} 或 EV _{SS} 输出：保持开路。
PCT2, PCT3	—	97, 98		
PCT4	RD	99		
PCT5	—	100		
PCT6	ASTB	101		
PCT7	—	102		
PDH0 至 PDH7	A16 至 A23	121 至 128		
PDL0 至 PDL4	AD0 至 AD4	105 至 109		
PDL5	AD5/FLMD1	110		
PDL6 至 PDL15	AD6 至 AD15	111 至 120		
AV _{REF0}	—	1	—	总是将该引脚连接到供电电源（即使在待机模式下）。
AV _{REF1}	—	5	—	
AV _{SS}	—	2	—	总是将该引脚直接连接到地（即使在待机模式下）。
EV _{DD}	—	34, 104	—	总是将该引脚连接到供电电源（即使在待机模式下）。
EV _{SS}	—	33, 103	—	总是将该引脚连接到地（即使在待机模式下）。
FLMD0	—	8	—	在 flash 存储器编程模式之外的其它模式下，直接连接至 V _{SS} 。
REGC	—	10	—	连接稳压器输出稳压电容（4.7 μ F（推荐值））
RESET	—	14	2	—
V _{DD}	—	9	—	总是将该引脚连接到供电电源（即使在待机模式下）。
V _{SS}	—	11	—	总是将该引脚直接连接到地（即使在待机模式下）。
X1	—	12	—	—
X2	—	13	—	—
XT1	—	15	16	连接至 V _{SS} 。
XT2	—	16	16	保持开路。

图 2-1 引脚 I/O 电路



注 端口模式下，没有迟滞特性。

2.4 注意事项

电源开启时，即便在复位期间，以下引脚也可能会暂时输出一个不确定电平。

- P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 引脚

第三章 CPU 功能

V850ES/JJ3 单片机的 CPU 是基于 RISC 架构的，且使用 5 级流水线，几乎所有指令的执行都可在一个时钟周期内完成。

3.1 特性

- 最短指令执行周期：31.25 ns（运行于 32MHz）。
30.5 μ s（运行于 32.768 kHz 的副时钟（f_{XT}））
- 存储空间：程序（物理地址）空间：64MB 线性
数据（逻辑地址）空间：4GB 线性
- 通用寄存器：32 位×32 个寄存器
- 内部 32 位架构
- 5 级流水线控制
- 乘法/除法指令
- 饱和运算指令
- 32 位移位指令：1 个时钟
- 长/短两种格式的数据装载/存储指令
- 4 种类型位操作指令：
 - SET1
 - CLR1
 - NOT1
 - TST1

3.2 CPU 寄存器组

V850ES/JJ3 的寄存器可分为两种类型：通用程序寄存器和专用系统寄存器。所有寄存器都是 32 位宽度。
详细信息，请参见 **V850ES 结构用户手册**。

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

3.2.1 程序寄存器组

程序寄存器组包括若干通用寄存器和一个程序计数器。

(1) 通用寄存器（r0 至 r31）

共有 32 个通用寄存器，r0 至 r31。这些寄存器中的任何一个都可用于存储数据变量或地址变量。然而，执行某些指令会隐式使用 r0 和 r30 寄存器，当使用这些寄存器时，一定要加以注意。寄存器 r0 的内容总是保持为 0，用于使用数据 0 的操作或 0 偏移量的寻址。当使用 SLD 和 SST 指令时，使用 r30 寄存器作为基址指针。汇编编译器和 C 编译器也隐式使用 r1、r3 至 r5、和 r31。使用这些寄存器之前，必须保存寄存器中的内容以防止丢失，并且在使用这些寄存器之后，将保存的内容恢复到寄存器中。r2 有时会被实时操作系统使用。如果实时操作系统没有使用 r2，那么它将可以用作变量寄存器。

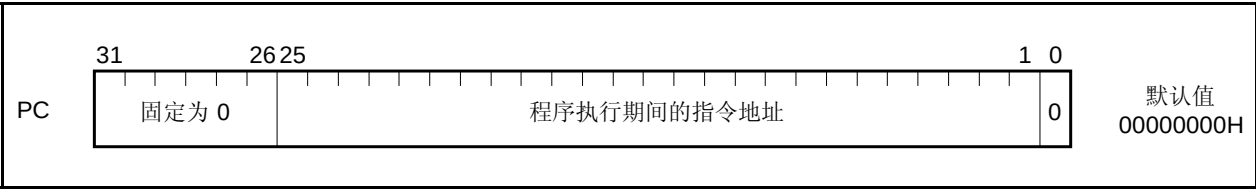
表 3-1. 程序寄存器

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

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

(2) 程序计数器（PC）

程序计数器用来保存程序执行期间的指令地址。该寄存器的低 26 位有效。其中第 31 位至第 26 位固定为 0，即使第 25 位向第 26 位有进位，也将忽略不计。第 0 位固定为 0，这意味着不能执行跳转到奇地址处的操作。



3.2.2 系统寄存器组

系统寄存器控制 CPU 的状态，并保存中断信息。

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

表 3-2. 系统寄存器号

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

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

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

备注 √：可以访问
×：禁止访问

(1) 中断状态保存寄存器 (EIPC 和 EIPSW)

EIPC 和 EIPSW 用于保存产生中断时的状态。

如果产生软件异常或者可屏蔽中断，那么程序计数器 (PC) 中的内容就保存到 EIPC 寄存器中，程序状态字 (PSW) 中的内容保存到 EIPSW 寄存器中 (如果产生的是非屏蔽中断，那么就将这些相应寄存器内容保存到 NMI 状态保存寄存器 (FEPC 及 FEPSW) 中)。

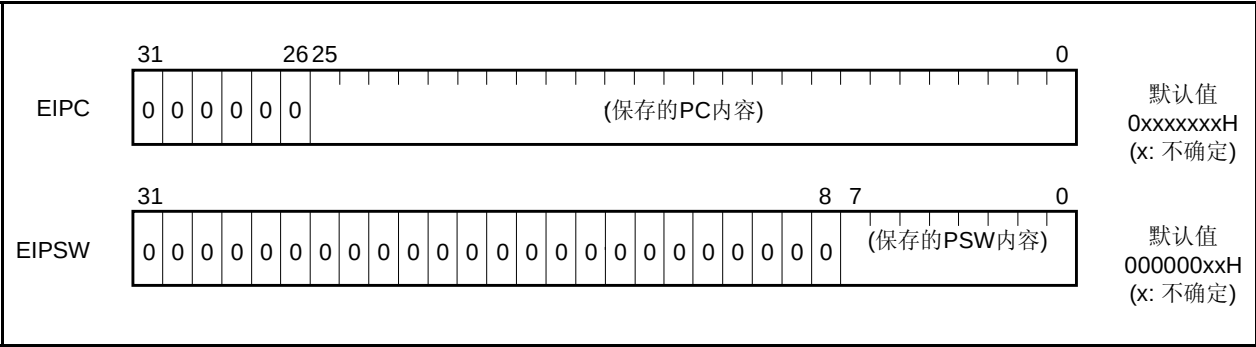
当有软件异常或者是可屏蔽中断产生时，就将当前执行指令的下一条指令地址保存到寄存器 EIPC 中，某些指令除外 (可参见 19.8 CPU 不响应中断期间)。

将 PSW 当前内容保存到 EIPSW 寄存器中。

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

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

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



(2) NMI 状态保存寄存器 (FEPC 和 FEPSW)

FEPC 和 FEPSW 用于保存产生不可屏蔽中断 (NMI) 时的状态。

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

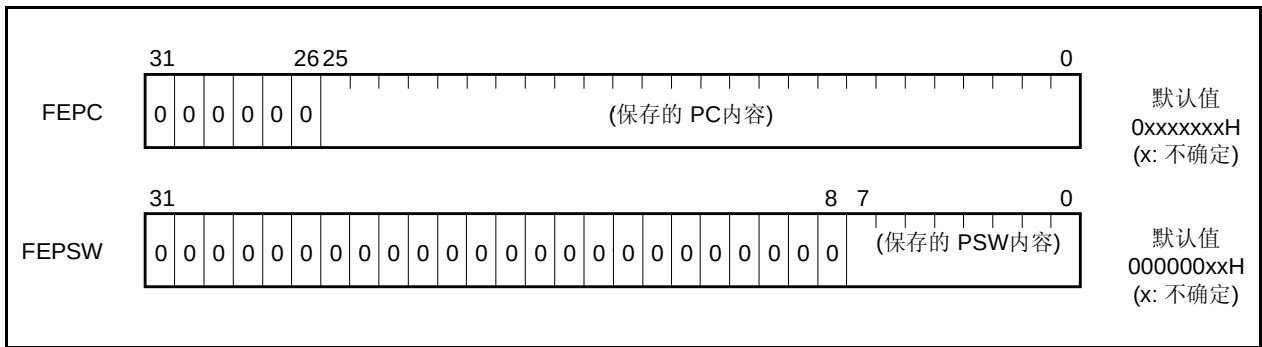
当有 NMI 产生时，就将当前执行指令的下一条指令地址保存到寄存器 FEPC 中，某些指令除外。

将 PSW 当前内容保存到 FEPSW 寄存器中。

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

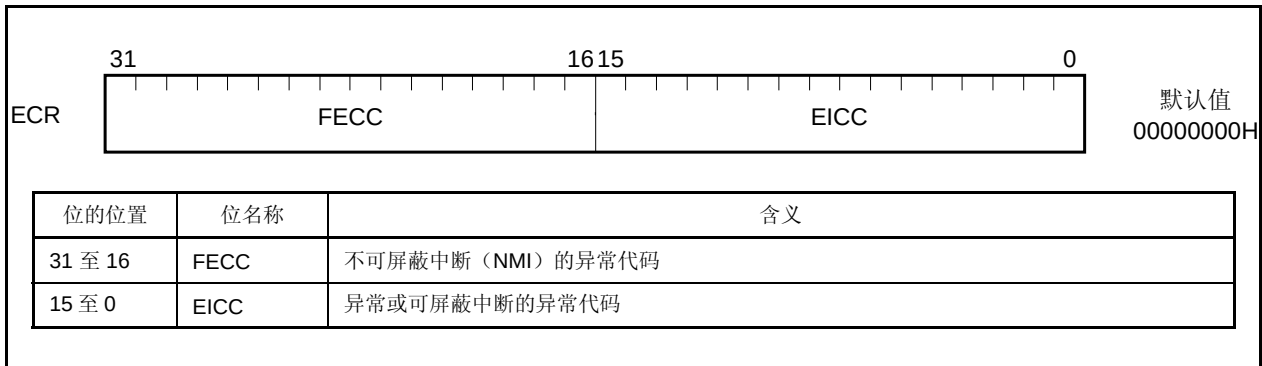
FEPC 的第 31 位至第 26 位以及 FEPSW 的第 31 位至第 8 位作为保留位用于将来功能扩展（这些位始终固定为 0）。

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



(3) 中断源寄存器 (ECR)

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



(4) 程序状态字 (PSW)

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

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

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

(1/2)

31

8 7 6 5 4 3 2 1 0

PSW

RFU

NP

EP

ID

SAT

CY

OV

S

Z

默认值
00000020H

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

备注

请阅读下页的“注”。

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

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



(6) 异常/调试陷阱状态保存寄存器 (DBPC 和 DBPSW)

DBPC 和 DBPSW 是异常/调试陷阱状态寄存器。

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

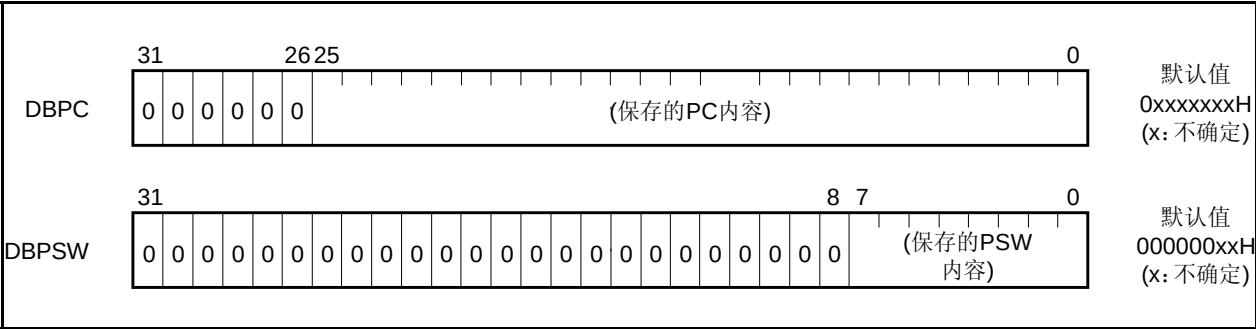
在异常陷阱或者调试陷阱产生时，保存到 DBPC 寄存器中的内容是当前执行指令的下一条指令的地址。

将 PSW 的当前内容保存到 DBPSW 寄存器中。

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

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

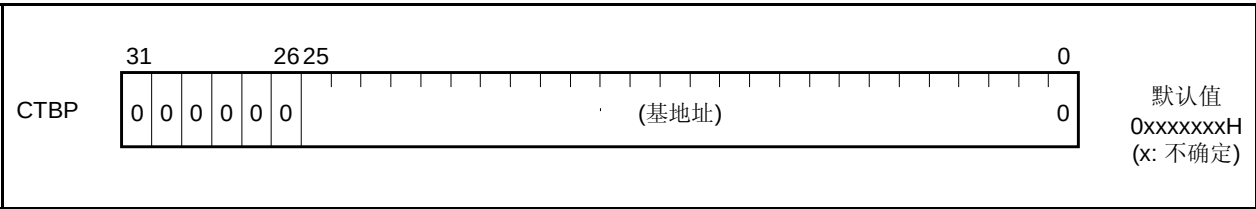
通过执行 DBRET 指令，将寄存器 DBPC 的值恢复到 PC 中，并将寄存器 DBPSW 的值恢复到 PSW 中。



(7) CALLT 指令基址指针 (CTBP)

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

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



3.3 操作模式

V850ES/JJ3 具有以下几种操作模式。

(1) 正常工作模式

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

(2) Flash 存储器编程模式

该模式下，使用 flash 编程器可以对内部 flash 存储器进行编程。

(3) 片上调试模式

V850ES/JJ3 提供使用 JTAG（Joint Test Action Group）通讯规范的片上调试功能。

详情参见第二十八章 片上调试功能。

3.3.1 指定操作模式

利用 FLMD0 引脚和 FLMD1 引脚来指定操作模式。

在正常操作模式下，复位解除后，向 FLMD0 引脚输入低电平。

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

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

备注

L: 低电平输入

H: 高电平输入

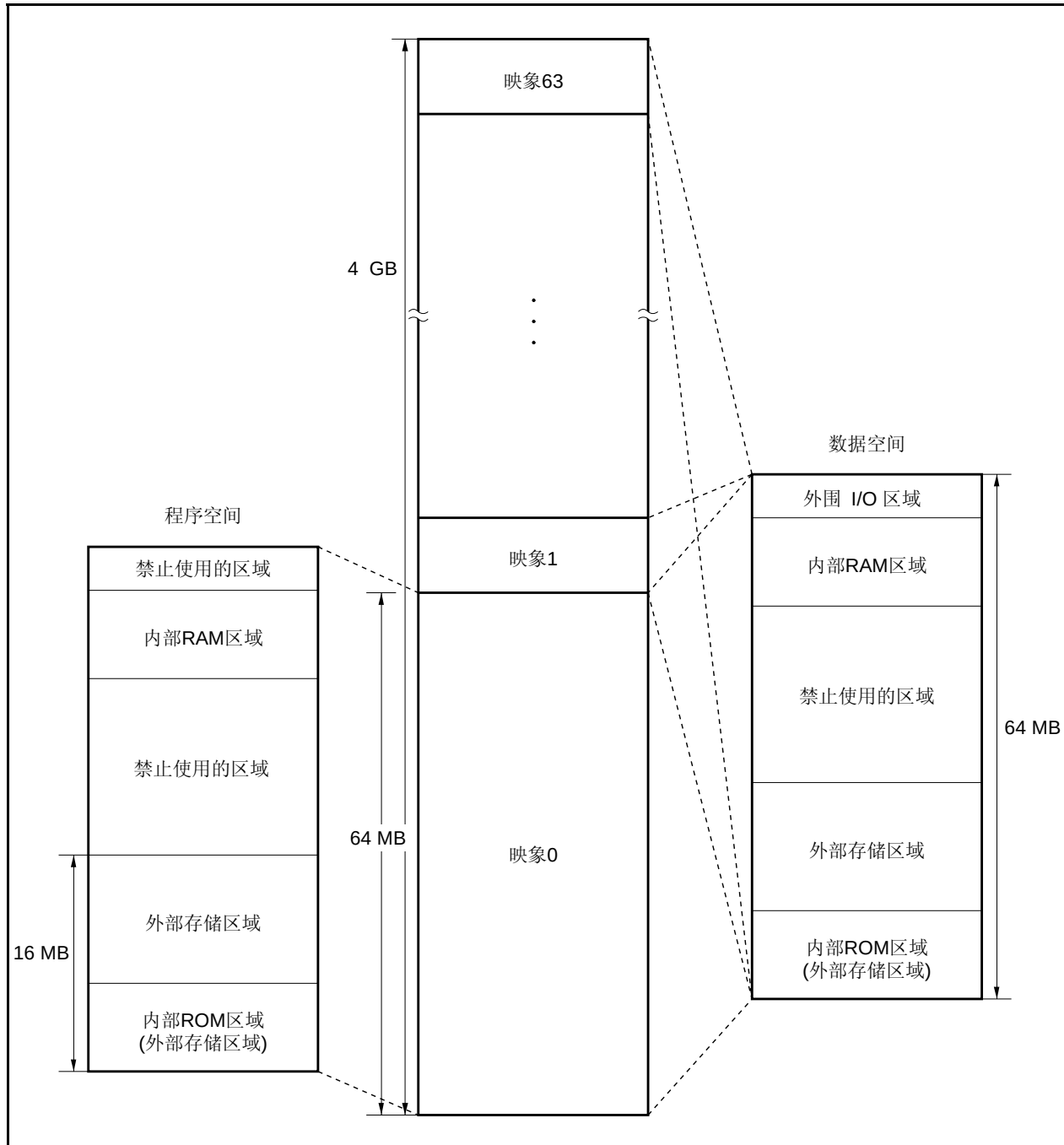
×: 无需理会

3.4 地址空间

3.4.1 CPU 地址空间

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

图 3-1. 地址空间映射



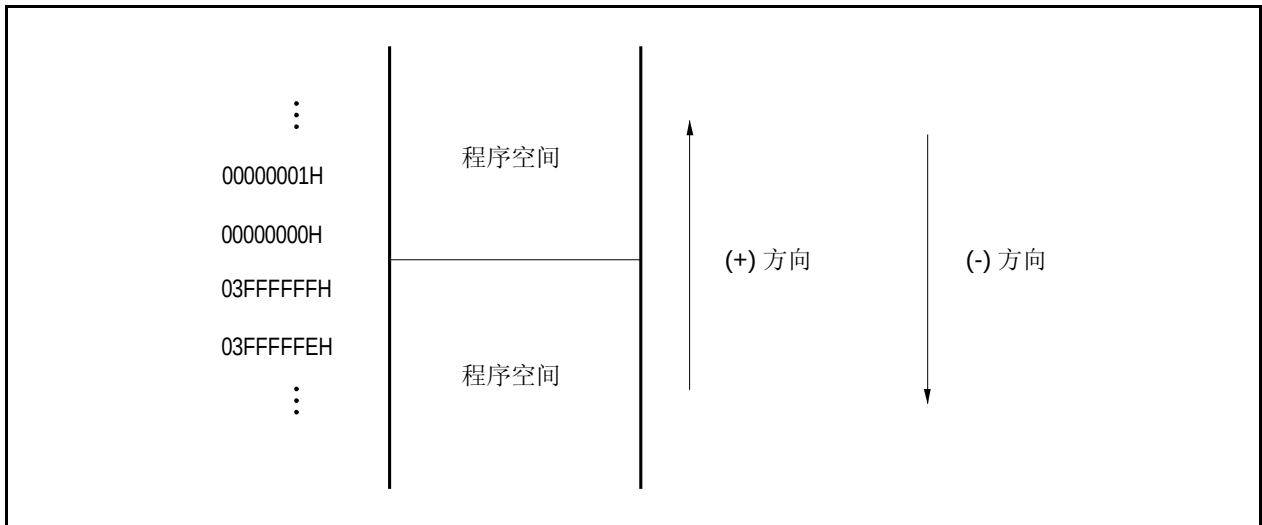
3.4.2 CPU 地址空间的绕回

(1) 程序空间

对于 32 位的 PC（程序计数器），其高 6 位固定为 0，且只有低 26 位是有效的。在计算跳转地址时，高 6 位会忽略第 25 位至第 26 位的进位或借位。

因此，程序空间的地址上限 03FFFFFFH 和地址下限 00000000H 是相邻的连续地址。像这种程序空间的最高地址和最低地址为相邻的连续地址的情况，称为地址的绕回。

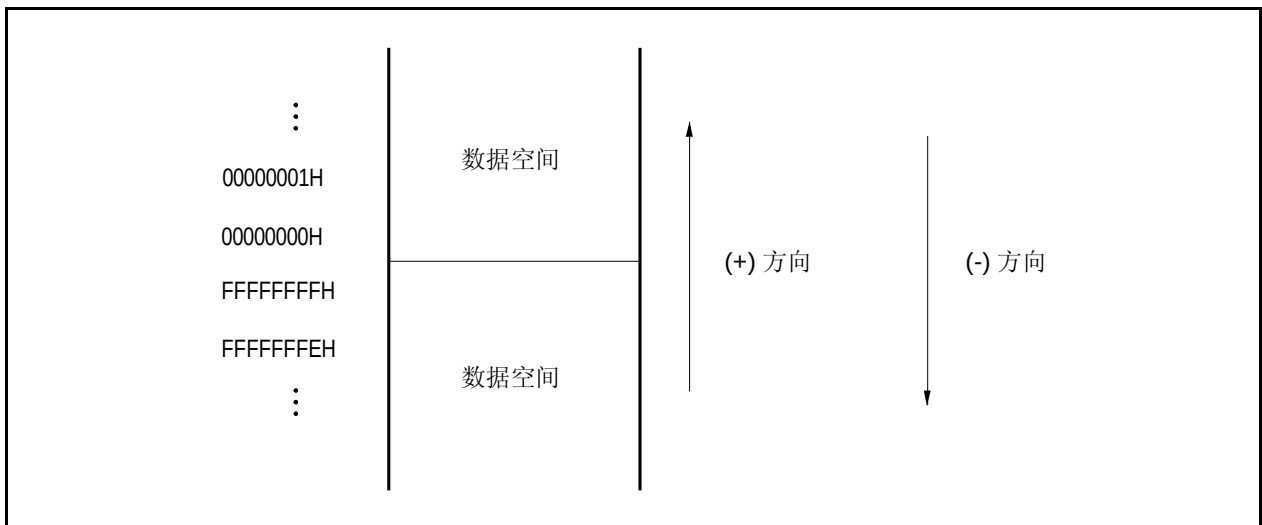
注意事项 由于地址 03FFF000H 至 03FFFFFFH 的 4 KB 区域为片上外围 I/O 区域，所以不能从该地址区域取指令。因此，如果跳转地址的计算结果会影响该区域，则不要执行此操作。



(2) 数据空间

操作数地址计算的结果超出 32 位时，将忽略该结果地址。

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



3.4.3 存储器映射

V850ES/JJ3 保留了下列区域。

图 3-2. 数据存储空间映射图（物理地址）

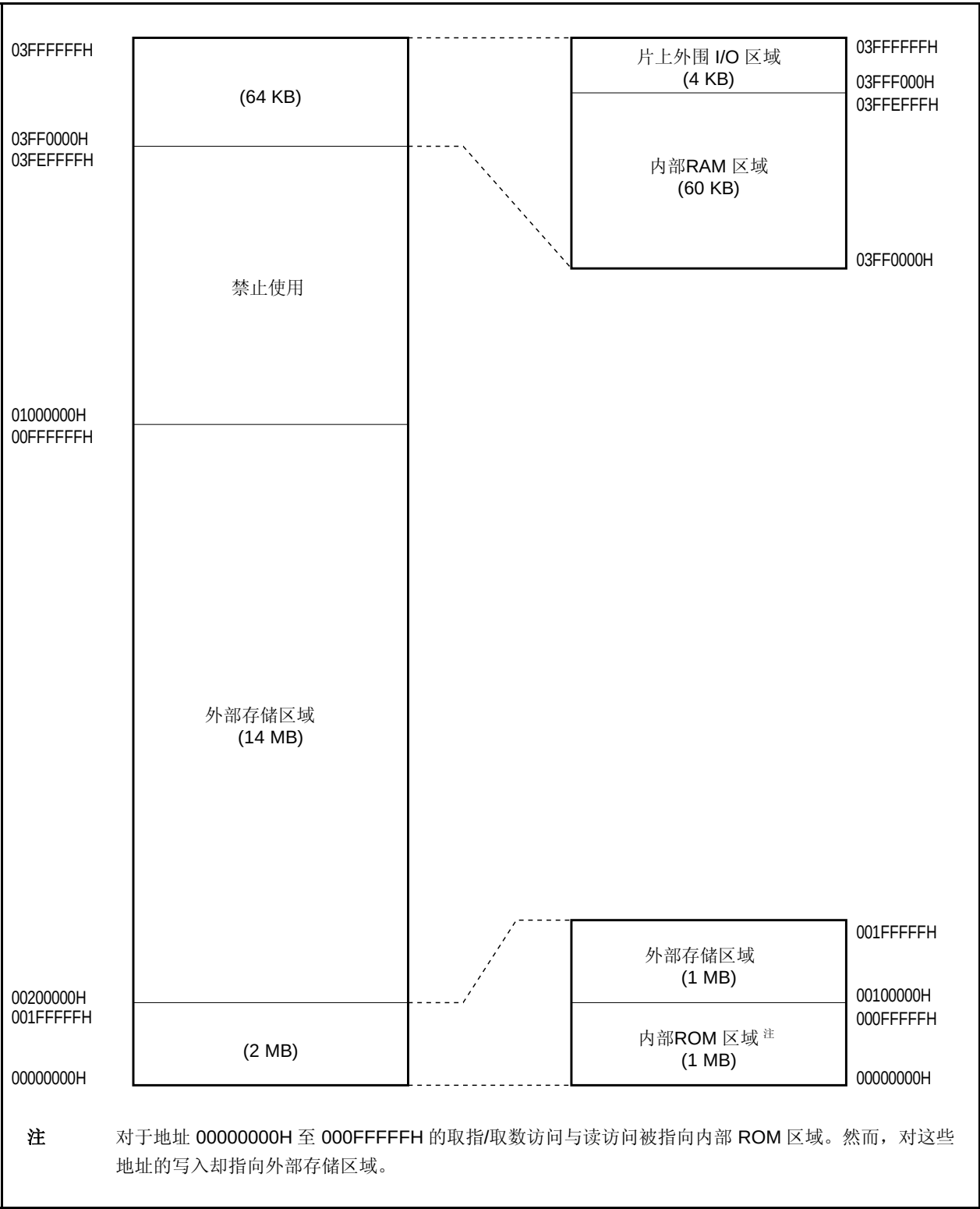
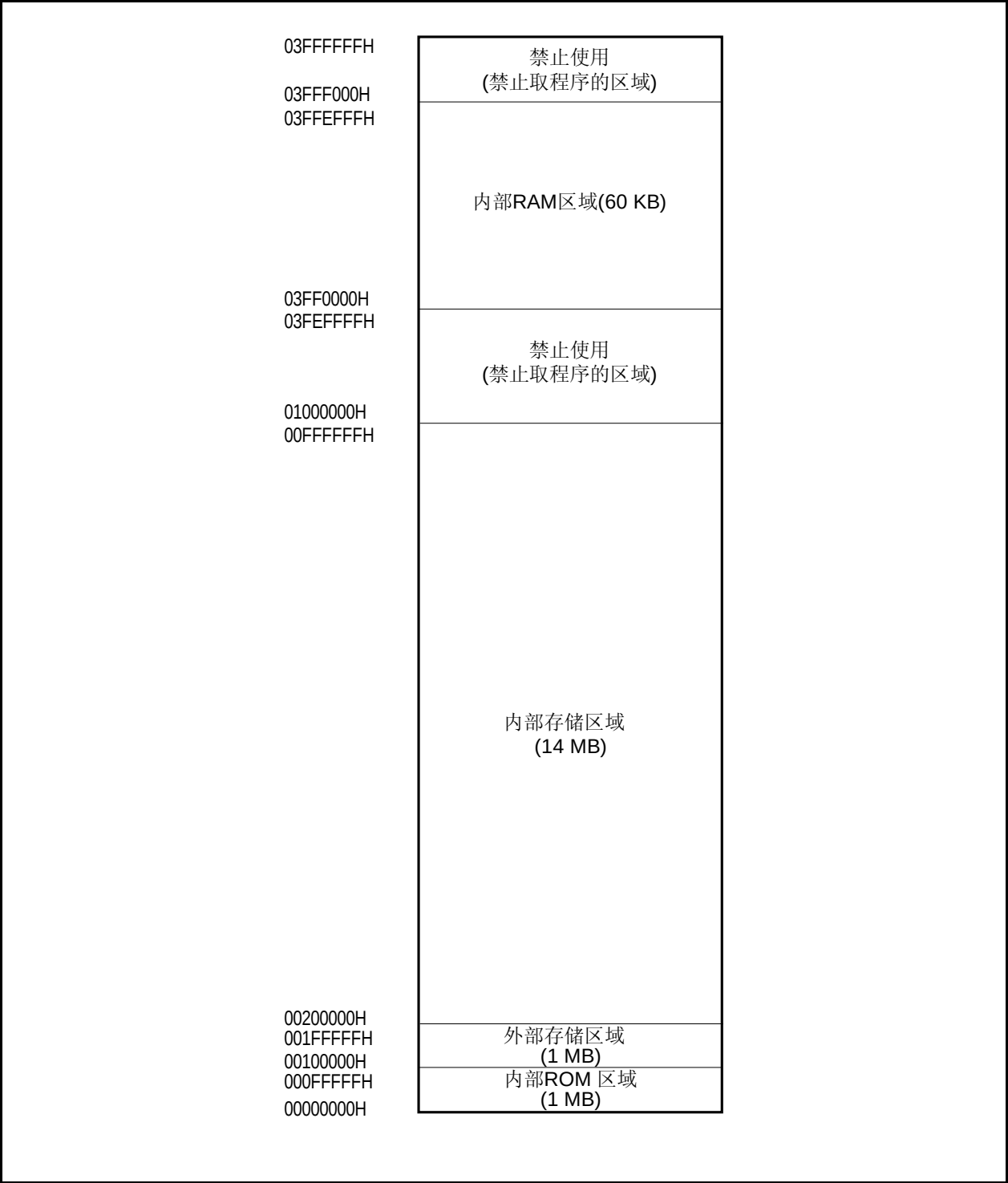


图 3-3. 程序存储器映射图



3.4.4 存储区域

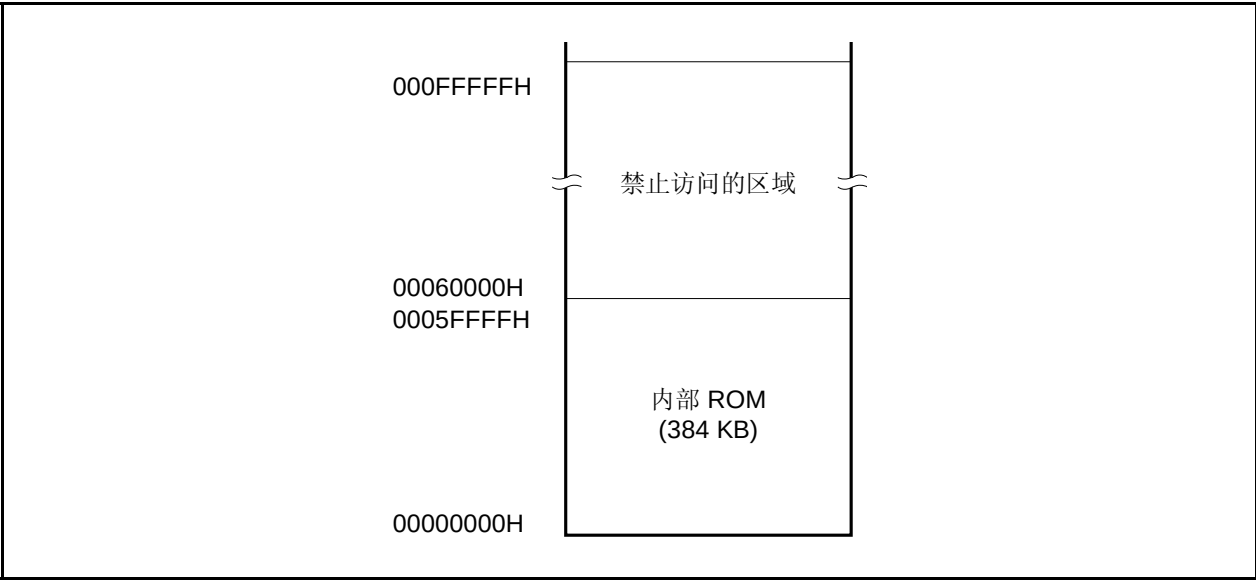
(1) 内部 ROM 区域

保留 1 MB 的区域作为内部 ROM 区域。

(a) 内部 ROM (384 KB)

μPD70F3743 中，该 384 KB 分配在地址 00000000H 至 0005FFFFH。
禁止访问地址 00060000H 至 000FFFFFH。

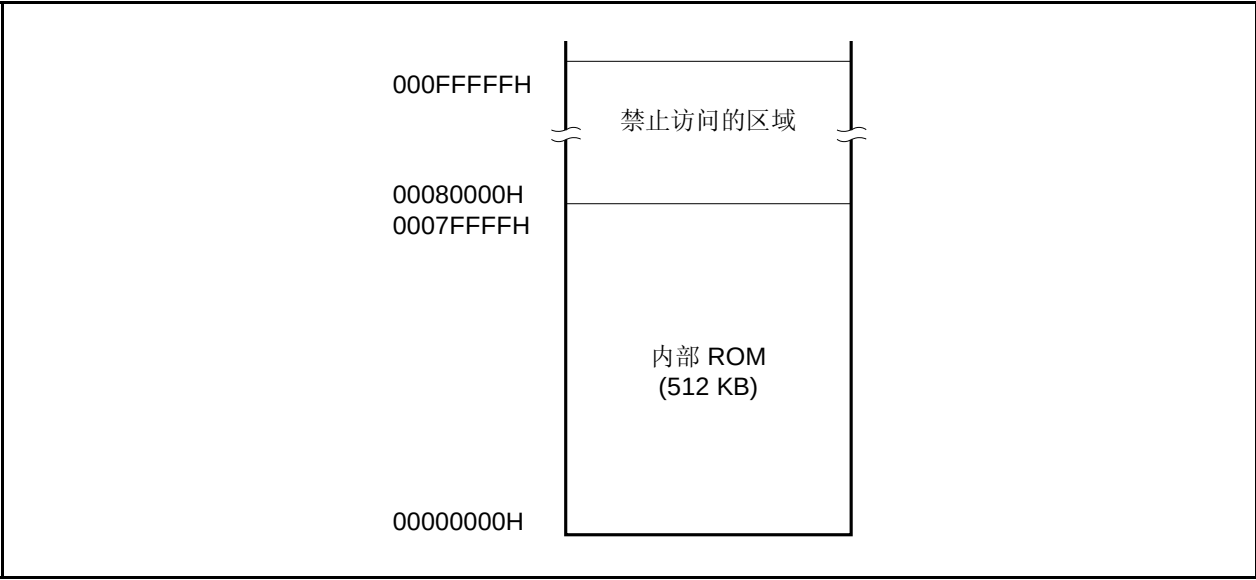
图 3-4. 内部 ROM 区域 (384 KB)



(b) 内部 ROM (512 KB)

μPD70F3744 中，该 512 KB 分配在地址 00000000H 至 0007FFFFH。
禁止访问地址 00080000H 至 000FFFFFH。

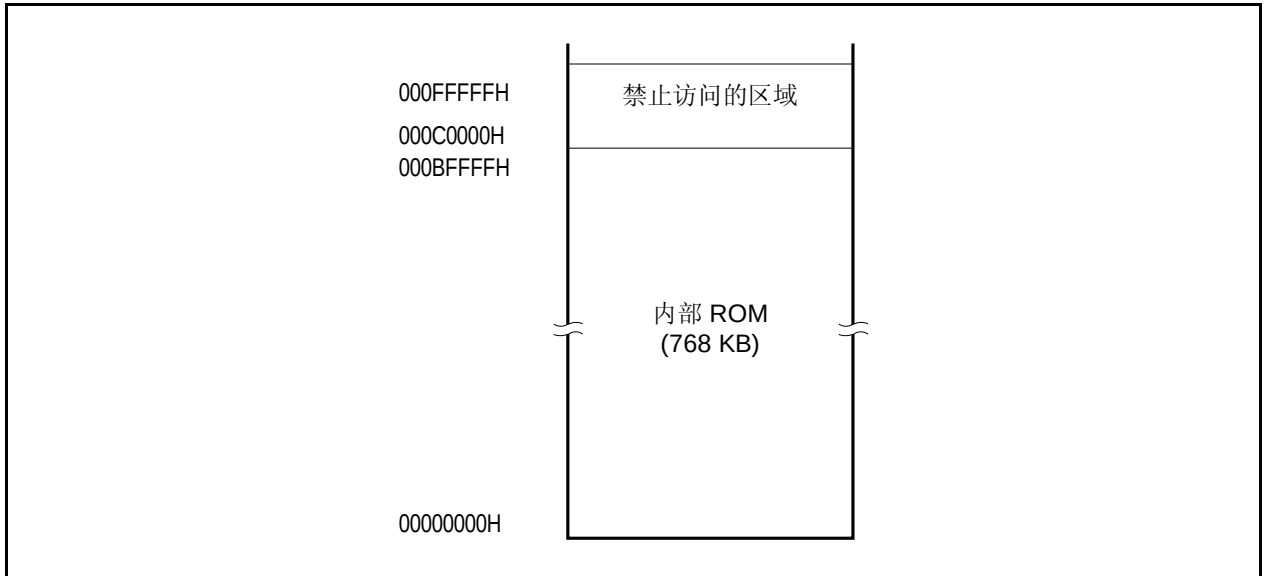
图 3-5. 内部 ROM 区域 (512 KB)



(c) 内部 ROM (768 KB)

μ PD70F3745 中，该 768 KB 分配在地址 00000000H 至 000BFFFFH。
禁止访问地址 000C0000H 至 000FFFFFFH。

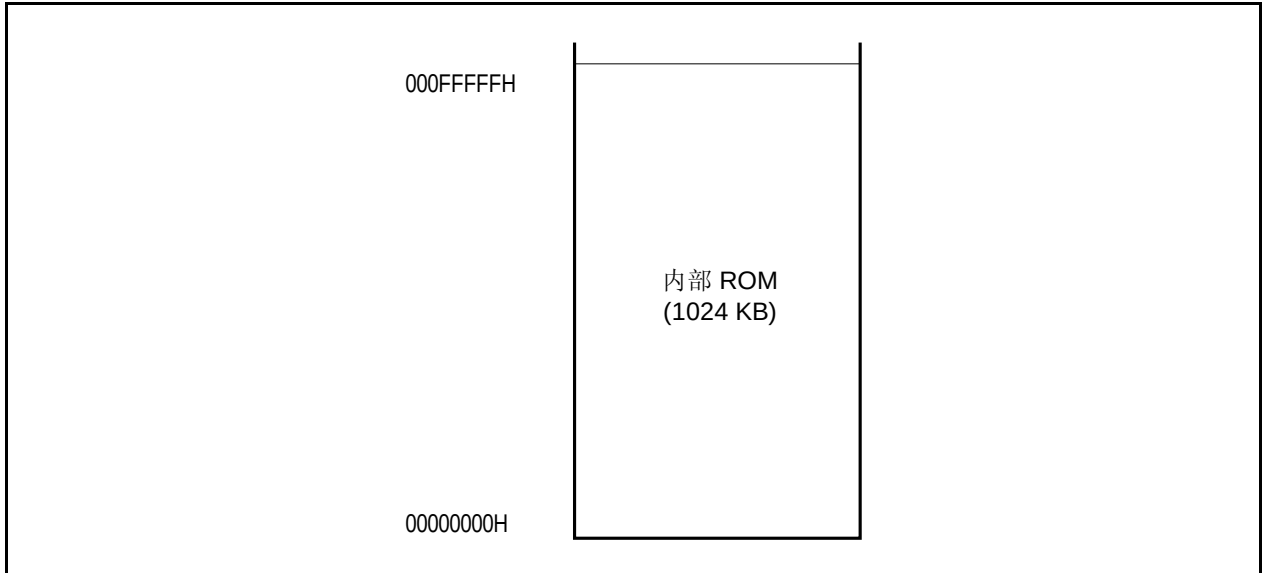
图 3-6. 内部 ROM 区域 (768 KB)



(d) 内部 ROM (1024 KB)

μ PD70F3746 中，该 1024 KB 分配在地址 00000000H 至 000FFFFFFH。

图 3-7. 内部 ROM 区域 (1024 KB)



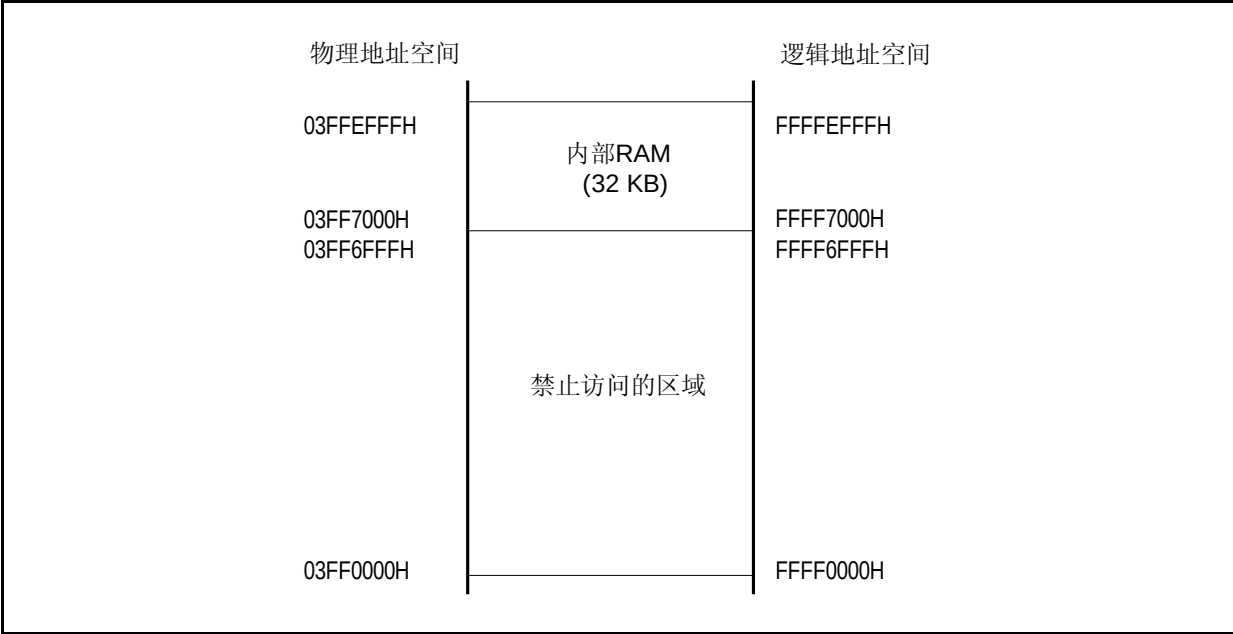
(2) 内部 RAM 区域

保留 60 KB 的区域作为内部 RAM 区域。

(a) 内部 RAM (32 KB)

μPD70F3743 中，该 32 KB 分配在地址 03FF7000H 至 03FF6FFFH。
禁止访问地址 03FF0000H 至 03FF6FFFH。

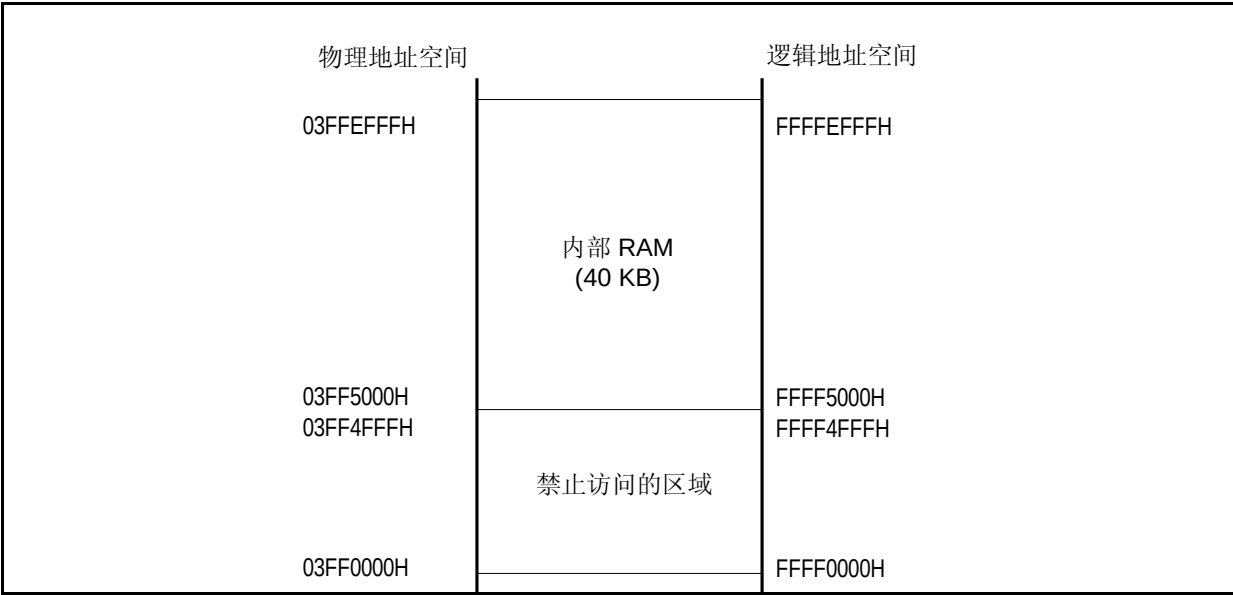
图 3-8. 内部 RAM 区域 (32 KB)



(b) 内部 RAM (40 KB)

μPD70F3744 中，该 40 KB 分配在地址 03FF5000H 至 03FF6FFFH。
禁止访问地址 03FF0000H 至 03FF4FFFH。

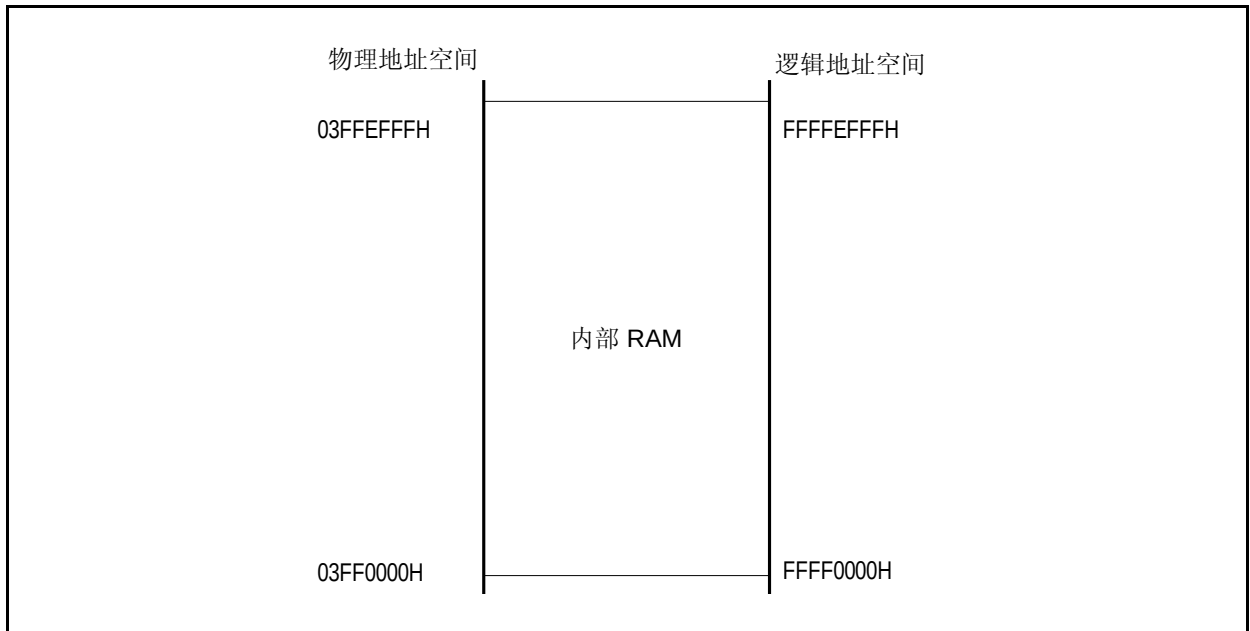
图 3-9. 内部 RAM 区域 (40 KB)



(c) 内部 RAM (60 KB)

μ PD70F3745 和 μ PD70F3746 中，该 60 KB 分配在地址 03FF0000H 至 03FFFEFFH。

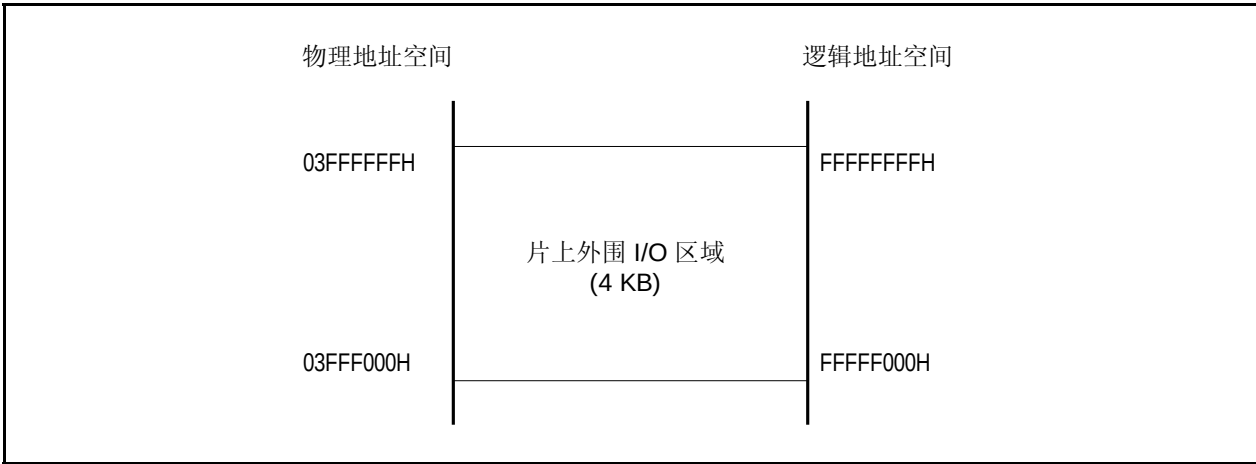
图 3-10. 内部 RAM 区域 (60 KB)



(3) 片上外围 I/O 区

保留地址 03FFF000H 至 03FFFFFFH 的 4 KB 存储区域作为片上外围 I/O 区域。

图 3-11. 片上外围 I/O 区域



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

- 注意事项**
1. 对寄存器进行字访问，会按照先低位后高位的顺序，对该字区域执行两次半字访问，其中忽略地址的低 2 位。
 2. 支持字节访问的寄存器，如果以半字为单位进行访问，当访问操作是读取操作时，那么该寄存器的高 8 位将不确定，数据被写入低 8 位。
 3. 没有定义为寄存器的地址区域被保留为将来扩展。如果对这些地址进行访问，其操作不确定而且也不保证操作的安全性。

(4) 外部存储区

保留 15 MB (00100000H 至 00FFFFFFH) 存储区域作为外部存储区域。详情参见第五章 总线控制功能。

3.4.5 地址空间的使用建议

V850ES/JJ3 架构要求访问数据空间的操作数时，必须有一个寄存器用作指针以产生地址。该指针±32 KB 范围的地址可以通过操作数指令直接存取。由于能够用作指针的通用寄存器数量是有限的，当改变指针值计算地址时，为了不影响系统的性能，应该尽可能多的指定通用寄存器来保存变量，这样也可以减少程序所占的空间。

(1) 程序空间

对于 32 位的 PC（程序计数器），其高 6 位固定为 0，而只有低 26 位是有效的。因此，对于程序空间，从地址 00000000H 开始的 64 MB 连续地址与存储器映射是无条件对应的。
要使用内部 RAM 作为程序空间，请访问下列地址。

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

RAM 空间大小	访问地址
60 KB	03FF0000H 至 03FFFFFFFFH
40 KB	03FF5000H 至 03FFFFFFFFH
32 KB	03FF7000H 至 03FFFFFFFFH

(2) 数据空间

对于 V850ES/JJ3 来说，4GB CPU 地址空间内可看作是 64 个 64MB 的地址空间。因此，26 位地址的最低有效位（第 25 位）作为符号扩展到 32 位，并作为地址分配。

(a) 地址绕回应用示例

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

示例：μPD70F3746

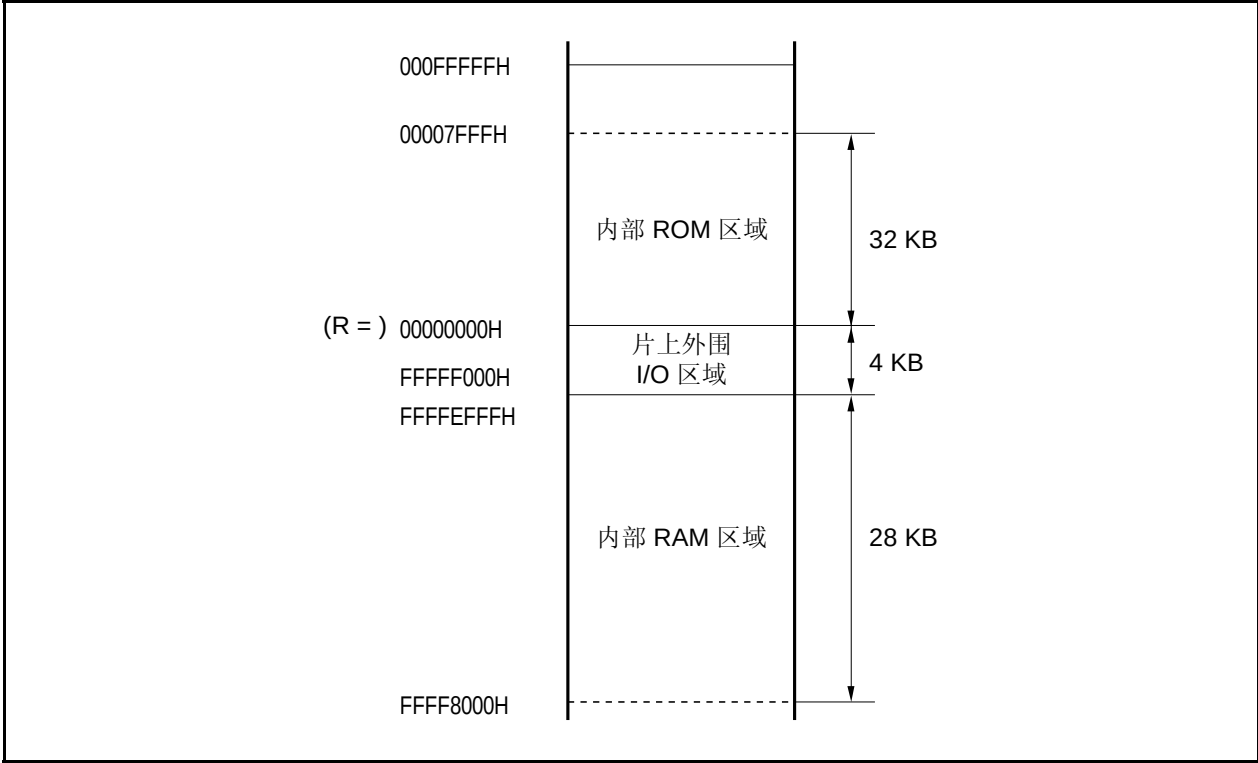
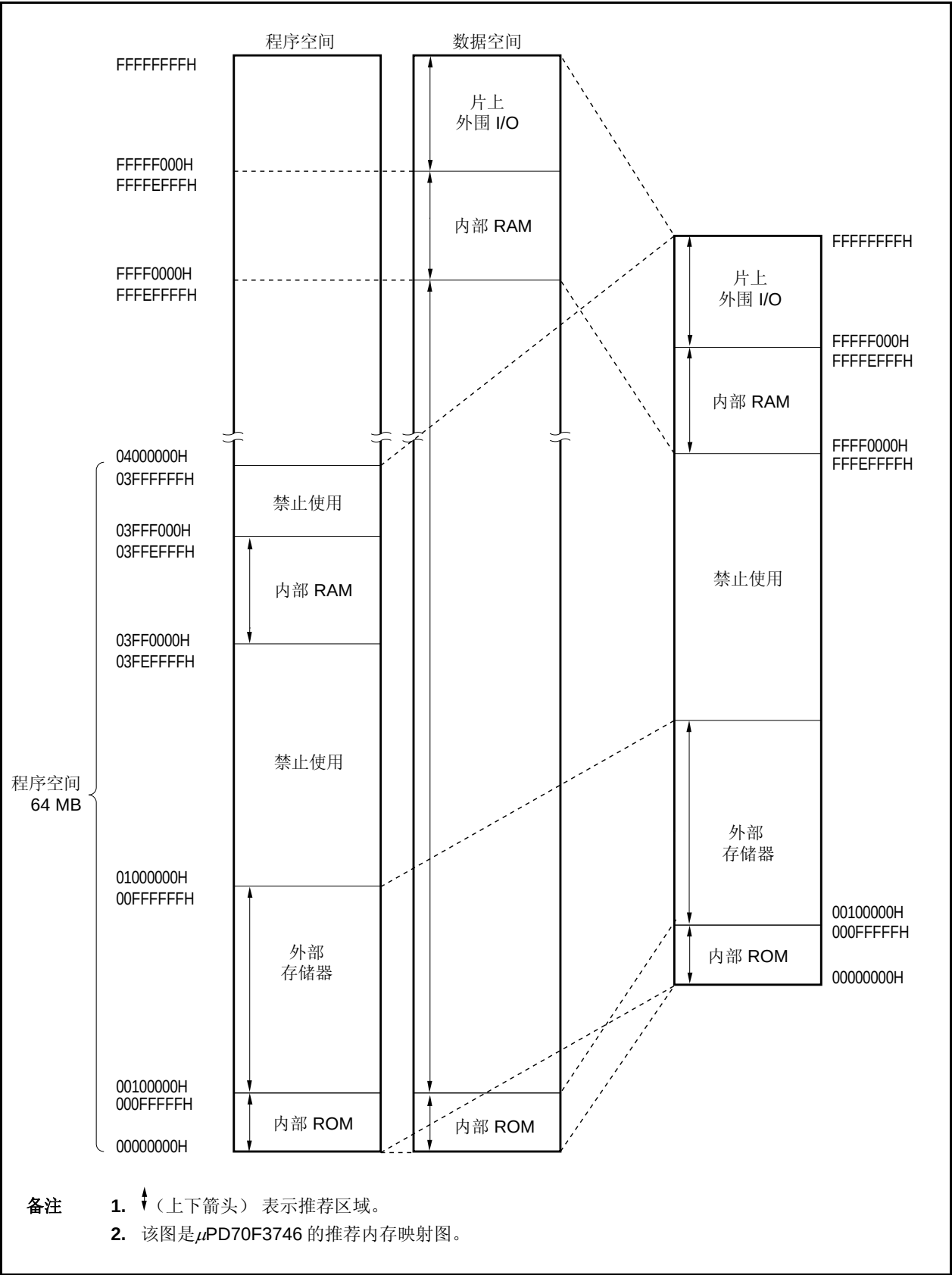


图 3-12. 推荐存储器映射图



3.4.6 外围 I/O 寄存器

(1/12)

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF004H	DL 端口寄存器	PDL	R/W			√	0000H ^注
FFFFF004H	DLL 端口寄存器	PDLL		√	√		00H ^注
FFFFF005H	DLH 端口寄存器	PDLH		√	√		00H ^注
FFFFF006H	DH 端口寄存器	PDH		√	√		00H ^注
FFFFF008H	CS 端口寄存器	PCS		√	√		00H ^注
FFFFF00AH	CT 端口寄存器	PCT		√	√		00H ^注
FFFFF00CH	CM 端口寄存器	PCM		√	√		00H ^注
FFFFF00EH	CD 端口寄存器	PCD		√	√		00H ^注
FFFFF024H	DL 端口模式寄存器	PMDL				√	FFFFH
FFFFF024H	DL 端口模式寄存器 L	PMDLL		√	√		FFH
FFFFF025H	DL 端口模式寄存器 H	PMDLH		√	√		FFH
FFFFF026H	DH 端口模式寄存器	PMDH		√	√		FFH
FFFFF028H	CS 端口模式寄存器	PMCS		√	√		FFH
FFFFF02AH	CT 端口模式寄存器	PMCT		√	√		FFH
FFFFF02CH	CM 端口模式寄存器	PMCM		√	√		FFH
FFFFF02EH	CD 端口模式寄存器	PMCD		√	√		FFH
FFFFF044H	DL 端口模式控制寄存器	PMCDL				√	0000H
FFFFF044H	DL 端口模式控制寄存器 L	PMCDLL		√	√		00H
FFFFF045H	DL 端口模式控制寄存器 H	PMCDLH		√	√		00H
FFFFF046H	DH 端口模式控制寄存器	PMCDH		√	√		00H
FFFFF048H	CS 端口模式控制寄存器	PMCCS		√	√		00H
FFFFF04AH	CT 端口模式控制寄存器	PMCCT		√	√		00H
FFFFF04CH	CM 端口模式控制寄存器	PMCCM		√	√		00H
FFFFF066H	总线宽度配置寄存器	BSC				√	5555H
FFFFF06EH	系统等待控制寄存器	VSWC			√		77H
FFFFF080H	DMA 源地址寄存器 0L	DSA0L				√	不确定
FFFFF082H	DMA 源地址寄存器 0H	DSA0H				√	不确定
FFFFF084H	DMA 目的地址寄存器 0L	DDA0L				√	不确定
FFFFF086H	DMA 目的地址寄存器 0H	DDA0H				√	不确定
FFFFF088H	DMA 源地址寄存器 1L	DSA1L				√	不确定
FFFFF08AH	DMA 源地址寄存器 1H	DSA1H				√	不确定
FFFFF08CH	DMA 目的地址寄存器 1L	DDA1L				√	不确定
FFFFF08EH	DMA 目的地址寄存器 1H	DDA1H				√	不确定
FFFFF090H	DMA 源地址寄存器 2L	DSA2L				√	不确定
FFFFF092H	DMA 源地址寄存器 2H	DSA2H				√	不确定
FFFFF094H	DMA 目的地址寄存器 2L	DDA2L				√	不确定
FFFFF096H	DMA 目的地址寄存器 2H	DDA2H				√	不确定
FFFFF098H	DMA 源地址寄存器 3L	DSA3L				√	不确定
FFFFF09AH	DMA 源地址寄存器 3H	DSA3H				√	不确定
FFFFF09CH	DMA 目的地址寄存器 3L	DDA3L				√	不确定
FFFFF09EH	DMA 目的地址寄存器 3H	DDA3H				√	不确定
FFFFF0C0H	DMA 传输计数寄存器 0	DBC0				√	不确定

注 输出锁存器的值为 00H 或 0000H。在输入期间，读取到的是引脚的状态。

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF0C2H	DMA 传输计数寄存器 1	DBC1	R/W			√	不确定
FFFFF0C4H	DMA 传输计数寄存器 2	DBC2				√	不确定
FFFFF0C6H	DMA 传输计数寄存器 3	DBC3				√	不确定
FFFFF0D0H	DMA 寻址控制寄存器 0	DADC0				√	0000H
FFFFF0D2H	DMA 寻址控制寄存器 1	DADC1				√	0000H
FFFFF0D4H	DMA 寻址控制寄存器 2	DADC2				√	0000H
FFFFF0D6H	DMA 寻址控制寄存器 3	DADC3				√	0000H
FFFFF0E0H	DMA 通道控制寄存器 0	DCHC0		√	√		00H
FFFFF0E2H	DMA 通道控制寄存器 1	DCHC1		√	√		00H
FFFFF0E4H	DMA 通道控制寄存器 2	DCHC2		√	√		00H
FFFFF0E6H	DMA 通道控制寄存器 3	DCHC3		√	√		00H
FFFFF100H	中断屏蔽寄存器 0	IMR0				√	FFFFH
FFFFF100H	中断屏蔽寄存器 0L	IMR0L		√	√		FFH
FFFFF101H	中断屏蔽寄存器 0H	IMR0H		√	√		FFH
FFFFF102H	中断屏蔽寄存器 1	IMR1				√	FFFFH
FFFFF102H	中断屏蔽寄存器 1L	IMR1L		√	√		FFH
FFFFF103H	中断屏蔽寄存器 1H	IMR1H		√	√		FFH
FFFFF104H	中断屏蔽寄存器 2	IMR2				√	FFFFH
FFFFF104H	中断屏蔽寄存器 2L	IMR2L		√	√		FFH
FFFFF105H	中断屏蔽寄存器 2H	IMR2H		√	√		FFH
FFFFF106H	中断屏蔽寄存器 3	IMR3				√	FFFFH
FFFFF106H	中断屏蔽寄存器 3L	IMR3L		√	√		FFH
FFFFF107H	中断屏蔽寄存器 3H	IMR3H		√	√		FFH
FFFFF108H	中断屏蔽寄存器 4	IMR4				√	FFFFH
FFFFF108H	中断屏蔽寄存器 4L	IMR4L		√	√		FFH
FFFFF109H	中断屏蔽寄存器 4H	IMR4H		√	√		FFH
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

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF132H	中断控制寄存器	TP10VIC	R/W	√	√		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	TP20VIC		√	√		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	TP30VIC		√	√		47H
FFFFF140H	中断控制寄存器	TP3CCIC0		√	√		47H
FFFFF142H	中断控制寄存器	TP3CCIC1		√	√		47H
FFFFF144H	中断控制寄存器	TP40VIC		√	√		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	TP50VIC		√	√		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
zcsFFFFF152H	中断控制寄存器	CB0RIC/IICIC1		√	√		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
FFFFF18EH	中断控制寄存器	PIC8		√	√		47H
FFFFF190H	中断控制寄存器	TP60VIC		√	√		47H
FFFFF192H	中断控制寄存器	TP6CCIC0		√	√		47H
FFFFF194H	中断控制寄存器	TP6CCIC1		√	√		47H
FFFFF196H	中断控制寄存器	TP70VIC		√	√		47H

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF198H	中断控制寄存器	TP7CCIC0	R/W	√	√		47H
FFFFF19AH	中断控制寄存器	TP7CCIC1		√	√		47H
FFFFF19CH	中断控制寄存器	TP8OVIC		√	√		47H
FFFFF19EH	中断控制寄存器	TP8CCIC0		√	√		47H
FFFFF1A0H	中断控制寄存器	TP8CCIC1		√	√		47H
FFFFF1A2H	中断控制寄存器	CB5RIC		√	√		47H
FFFFF1A4H	中断控制寄存器	CB5TIC		√	√		47H
FFFFF1A6H	中断控制寄存器	UA3RIC		√	√		47H
FFFFF1A8H	中断控制寄存器	UA3TIC		√	√		47H
FFFFF1FAH	正在服务的中断优先级寄存器	ISPR	R	√	√		00H
FFFFF1FCH	命令寄存器	PRCMD	W	√	√		不确定
FFFFF1FEH	省电控制寄存器	PSC	R/W	√	√		00H
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 转换结果寄存器 0	ADA0CR0	R			√	不确定
FFFFF211H	A/D 转换结果寄存器 0H	ADA0CR0H			√		不确定
FFFFF212H	A/D 转换结果寄存器 1	ADA0CR1				√	不确定
FFFFF213H	A/D 转换结果寄存器 1H	ADA0CR1H			√		不确定
FFFFF214H	A/D 转换结果寄存器 2	ADA0CR2				√	不确定
FFFFF215H	A/D 转换结果寄存器 2H	ADA0CR2H			√		不确定
FFFFF216H	A/D 转换结果寄存器 3	ADA0CR3				√	不确定
FFFFF217H	A/D 转换结果寄存器 3H	ADA0CR3H			√		不确定
FFFFF218H	A/D 转换结果寄存器 4	ADA0CR4				√	不确定
FFFFF219H	A/D 转换结果寄存器 4H	ADA0CR4H			√		不确定
FFFFF21AH	A/D 转换结果寄存器 5	ADA0CR5				√	不确定
FFFFF21BH	A/D 转换结果寄存器 5H	ADA0CR5H			√		不确定
FFFFF21CH	A/D 转换结果寄存器 6	ADA0CR6				√	不确定
FFFFF21DH	A/D 转换结果寄存器 6H	ADA0CR6H			√		不确定
FFFFF21EH	A/D 转换结果寄存器 7	ADA0CR7				√	不确定
FFFFF21FH	A/D 转换结果寄存器 7H	ADA0CR7H			√		不确定
FFFFF220H	A/D 转换结果寄存器 8	ADA0CR8				√	不确定
FFFFF221H	A/D 转换结果寄存器 8H	ADA0CR8H			√		不确定
FFFFF222H	A/D 转换结果寄存器 9	ADA0CR9				√	不确定
FFFFF223H	A/D 转换结果寄存器 9H	ADA0CR9H			√		不确定
FFFFF224H	A/D 转换结果寄存器 10	ADA0CR10				√	不确定
FFFFF225H	A/D 转换结果寄存器 10H	ADA0CR10H			√		不确定
FFFFF226H	A/D 转换结果寄存器 11	ADA0CR11				√	不确定
FFFFF227H	A/D 转换结果寄存器 11H	ADA0CR11H			√		不确定

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF228H	A/D 转换结果寄存器 12	ADA0CR12	R			√	不确定
FFFFF229H	A/D 转换结果寄存器 12H	ADA0CR12H			√		不确定
FFFFF22AH	A/D 转换结果寄存器 13	ADA0CR13				√	不确定
FFFFF22BH	A/D 转换结果寄存器 13H	ADA0CR13H			√		不确定
FFFFF22CH	A/D 转换结果寄存器 14	ADA0CR14				√	不确定
FFFFF22DH	A/D 转换结果寄存器 14H	ADA0CR14H			√		不确定
FFFFF22EH	A/D 转换结果寄存器 15	ADA0CR15				√	不确定
FFFFF22FH	A/D 转换结果寄存器 15H	ADA0CR15H			√		不确定
FFFFF280H	D/A 转换值设置寄存器 0	DA0CS0	R/W		√		00H
FFFFF281H	D/A 转换值设置寄存器 1	DA0CS1			√		00H
FFFFF282H	D/A 转换模式寄存器	DA0M		√	√		00H
FFFFF300H	按键返回模式寄存器	KRM		√	√		00H
FFFFF308H	选择器操作控制寄存器 0	SELCNT0		√	√		00H
FFFFF310H	CRC 输入寄存器	CRCIN			√		00H
FFFFF312H	CRC 数据寄存器	CRCD				√	0000H
FFFFF318H	噪声消除控制寄存器	NFC			√		00H
FFFFF320H	预分频模式寄存器 1	PRSM1		√	√		00H
FFFFF321H	预分频器比较寄存器 1	PRSCM1			√		00H
FFFFF324H	预分频模式寄存器 2	PRSM2		√	√		00H
FFFFF325H	预分频器比较寄存器 2	PRSCM2			√		00H
FFFFF328H	预分频模式寄存器 3	PRSM3		√	√		00H
FFFFF329H	预分频器比较寄存器 3	PRSCM3			√		00H
FFFFF340H	IIC 分频时钟选择寄存器 0	OCKS0			√		00H
FFFFF344H	IIC 分频时钟选择寄存器 1	OCKS1			√		00H
FFFFF400H	端口 0 寄存器	P0		√	√		00H ^注
FFFFF402H	端口 1 寄存器	P1		√	√		00H ^注
FFFFF406H	端口 3 寄存器	P3				√	0000H ^注
FFFFF406H	端口 3L 寄存器	P3L		√	√		00H ^注
FFFFF407H	端口 3H 寄存器	P3H		√	√		00H ^注
FFFFF408H	端口 4 寄存器	P4		√	√		00H ^注
FFFFF40AH	端口 5 寄存器	P5		√	√		00H ^注
FFFFF40CH	端口 6 寄存器	P6				√	0000H ^注
FFFFF40CH	端口 6L 寄存器	P6L		√	√		00H ^注
FFFFF40DH	端口 6H 寄存器	P6H		√	√		00H ^注
FFFFF40EH	端口 7L 寄存器	P7L		√	√		00H ^注
FFFFF40FH	端口 7H 寄存器	P7H		√	√		00H ^注
FFFFF410H	端口 8 寄存器	P8		√	√		00H ^注
FFFFF412H	端口 9 寄存器	P9				√	0000H ^注
FFFFF412H	端口 9L 寄存器	P9L		√	√		00H ^注
FFFFF413H	端口 9H 寄存器	P9H		√	√		00H ^注
FFFFF420H	端口 0 模式寄存器	PM0		√	√		FFH

注 输出锁存器的值为 00H 或 0000H。在输入期间，读取到的是引脚的状态。

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF422H	端口 1 模式寄存器	PM1	R/W	√	√		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		√	√		FFH
FFFFF42AH	端口 5 模式寄存器	PM5		√	√		FFH
FFFFF42CH	端口 6 模式寄存器	PM6				√	FFFFH
FFFFF42CH	端口 6 模式寄存器 L	PM6L		√	√		FFH
FFFFF42DH	端口 6 模式寄存器 H	PM6H		√	√		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
FFFFF430H	端口 8 模式寄存器	PM8		√	√		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
FFFFF44CH	端口 6 模式控制寄存器	PMC6				√	0000H
FFFFF44CH	端口 6 模式控制寄存器 L	PMC6L		√	√		00H
FFFFF44DH	端口 6 模式控制寄存器 H	PMC6H		√	√		00H
FFFFF450H	端口 8 模式控制寄存器	PMC8		√	√		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
FFFFF46DH	端口 6 功能控制寄存器 H	PFC6H		√	√		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

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF540H	TMQ0 控制寄存器 0	TQ0CTL0	R/W	√	√		00H
FFFFF541H	TMQ0 控制寄存器 1	TQ0CTL1		√	√		00H
FFFFF542H	TMQ0 I/O 控制寄存器 0	TQ0IOC0		√	√		00H
FFFFF543H	TMQ0 I/O 控制寄存器 1	TQ0IOC1		√	√		00H
FFFFF544H	TMQ0 I/O 控制寄存器 2	TQ0IOC2		√	√		00H
FFFFF545H	TMQ0 选项寄存器 0	TQ0OPT0		√	√		00H
FFFFF546H	TMQ0 捕获/比较寄存器 0	TQ0CCR0				√	0000H
FFFFF548H	TMQ0 捕获/比较寄存器 1	TQ0CCR1				√	0000H
FFFFF54AH	TMQ0 捕获/比较寄存器 2	TQ0CCR2				√	0000H
FFFFF54CH	TMQ0 捕获/比较寄存器 3	TQ0CCR3				√	0000H
FFFFF54EH	TMQ0 计数器读取缓冲寄存器	TQ0CNT	R			√	0000H
FFFFF590H	TMP0 控制寄存器 0	TP0CTL0	R/W	√	√		00H
FFFFF591H	TMP0 控制寄存器 1	TP0CTL1		√	√		00H
FFFFF592H	TMP0 I/O 控制寄存器 0	TP0IOC0		√	√		00H
FFFFF593H	TMP0 I/O 控制寄存器 1	TP0IOC1		√	√		00H
FFFFF594H	TMP0 I/O 控制寄存器 2	TP0IOC2		√	√		00H
FFFFF595H	TMP0 选项寄存器 0	TP0OPT0		√	√		00H
FFFFF596H	TMP0 捕获/比较寄存器 0	TP0CCR0				√	0000H
FFFFF598H	TMP0 捕获/比较寄存器 1	TP0CCR1				√	0000H
FFFFF59AH	TMP0 计数器读取缓冲寄存器	TP0CNT				√	0000H
FFFFF5A0H	TMP1 控制寄存器 0	TP1CTL0	R/W	√	√		00H
FFFFF5A1H	TMP1 控制寄存器 1	TP1CTL1		√	√		00H
FFFFF5A2H	TMP1 I/O 控制寄存器 0	TP1IOC0		√	√		00H
FFFFF5A3H	TMP1 I/O 控制寄存器 1	TP1IOC1		√	√		00H
FFFFF5A4H	TMP1 I/O 控制寄存器 2	TP1IOC2		√	√		00H
FFFFF5A5H	TMP1 选项寄存器 0	TP1OPT0		√	√		00H
FFFFF5A6H	TMP1 捕获/比较寄存器 0	TP1CCR0				√	0000H
FFFFF5A8H	TMP1 捕获/比较寄存器 1	TP1CCR1				√	0000H
FFFFF5AAH	TMP1 计数器读取缓冲寄存器	TP1CNT				√	0000H
FFFFF5B0H	TMP2 控制寄存器 0	TP2CTL0	R/W	√	√		00H
FFFFF5B1H	TMP2 控制寄存器 1	TP2CTL1		√	√		00H
FFFFF5B2H	TMP2 I/O 控制寄存器 0	TP2IOC0		√	√		00H
FFFFF5B3H	TMP2 I/O 控制寄存器 1	TP2IOC1		√	√		00H
FFFFF5B4H	TMP2 I/O 控制寄存器 2	TP2IOC2		√	√		00H
FFFFF5B5H	TMP2 选项寄存器 0	TP2OPT0		√	√		00H
FFFFF5B6H	TMP2 捕获/比较寄存器 0	TP2CCR0				√	0000H
FFFFF5B8H	TMP2 捕获/比较寄存器 1	TP2CCR1				√	0000H
FFFFF5BAH	TMP2 计数器读取缓冲寄存器	TP2CNT				√	0000H
FFFFF5C0H	TMP3 控制寄存器 0	TP3CTL0	R/W	√	√		00H
FFFFF5C1H	TMP3 控制寄存器 1	TP3CTL1		√	√		00H
FFFFF5C2H	TMP3 I/O 控制寄存器 0	TP3IOC0		√	√		00H
FFFFF5C3H	TMP3 I/O 控制寄存器 1	TP3IOC1		√	√		00H
FFFFF5C4H	TMP3 I/O 控制寄存器 2	TP3IOC2		√	√		00H

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF5C5H	TMP3 选项寄存器 0	TP3OPT0	R/W	√	√		00H
FFFFF5C6H	TMP3 捕获/比较寄存器 0	TP3CCR0				√	0000H
FFFFF5C8H	TMP3 捕获/比较寄存器 1	TP3CCR1				√	0000H
FFFFF5CAH	TMP3 计数器读取缓冲寄存器	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
FFFFF5D2H	TMP4 I/O 控制寄存器 0	TP4IOC0		√	√		00H
FFFFF5D3H	TMP4 I/O 控制寄存器 1	TP4IOC1		√	√		00H
FFFFF5D4H	TMP4 I/O 控制寄存器 2	TP4IOC2		√	√		00H
FFFFF5D5H	TMP4 选项寄存器 0	TP4OPT0		√	√		00H
FFFFF5D6H	TMP4 捕获/比较寄存器 0	TP4CCR0				√	0000H
FFFFF5D8H	TMP4 捕获/比较寄存器 1	TP4CCR1				√	0000H
FFFFF5DAH	TMP4 计数器读取缓冲寄存器	TP4CNT				√	0000H
FFFFF5E0H	TMP5 控制寄存器 0	TP5CTL0	R/W	√	√		00H
FFFFF5E1H	TMP5 控制寄存器 1	TP5CTL1		√	√		00H
FFFFF5E2H	TMP5 I/O 控制寄存器 0	TP5IOC0		√	√		00H
FFFFF5E3H	TMP5 I/O 控制寄存器 1	TP5IOC1		√	√		00H
FFFFF5E4H	TMP5 I/O 控制寄存器 2	TP5IOC2		√	√		00H
FFFFF5E5H	TMP5 选项寄存器 0	TP5OPT0		√	√		00H
FFFFF5E6H	TMP5 捕获/比较寄存器 0	TP5CCR0				√	0000H
FFFFF5E8H	TMP5 捕获/比较寄存器 1	TP5CCR1				√	0000H
FFFFF5EAH	TMP5 计数器读取缓冲寄存器	TP5CNT				√	0000H
FFFFF5F0H	TMP6 控制寄存器 0	TP6CTL0	R/W	√	√		00H
FFFFF5F1H	TMP6 控制寄存器 1	TP6CTL1		√	√		00H
FFFFF5F2H	TMP6 I/O 控制寄存器 0	TP6IOC0		√	√		00H
FFFFF5F3H	TMP6 I/O 控制寄存器 1	TP6IOC1		√	√		00H
FFFFF5F4H	TMP6 I/O 控制寄存器 2	TP6IOC2		√	√		00H
FFFFF5F5H	TMP6 选项寄存器 0	TP6OPT0		√	√		00H
FFFFF5F6H	TMP6 捕获/比较寄存器 0	TP6CCR0				√	0000H
FFFFF5F8H	TMP6 捕获/比较寄存器 1	TP6CCR1				√	0000H
FFFFF5FAH	TMP6 计数器读取缓冲寄存器	TP6CNT				√	0000H
FFFFF600H	TMP7 控制寄存器 0	TP7CTL0	R/W	√	√		00H
FFFFF601H	TMP7 控制寄存器 1	TP7CTL1		√	√		00H
FFFFF602H	TMP7 I/O 控制寄存器 0	TP7IOC0		√	√		00H
FFFFF603H	TMP7 I/O 控制寄存器 1	TP7IOC1		√	√		00H
FFFFF604H	TMP7 I/O 控制寄存器 2	TP7IOC2		√	√		00H
FFFFF605H	TMP7 选项寄存器 0	TP7OPT0		√	√		00H
FFFFF606H	TMP7 捕获/比较寄存器 0	TP7CCR0				√	0000H
FFFFF608H	TMP7 捕获/比较寄存器 1	TP7CCR1				√	0000H
FFFFF60AH	TMP7 计数器读取缓冲寄存器	TP7CNT				√	0000H
FFFFF610H	TMP8 控制寄存器 0	TP8CTL0	R/W	√	√		00H
FFFFF611H	TMP8 控制寄存器 1	TP8CTL1		√	√		00H

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF612H	TMP8 I/O 控制寄存器 0	TP8IOC0	R/W	√	√		00H
FFFFF613H	TMP8 I/O 控制寄存器 1	TP8IOC1		√	√		00H
FFFFF614H	TMP8 I/O 控制寄存器 2	TP8IOC2		√	√		00H
FFFFF615H	TMP8 选项寄存器 0	TP8OPT0		√	√		00H
FFFFF616H	TMP8 捕获/比较寄存器 0	TP8CCR0				√	0000H
FFFFF618H	TMP8 捕获/比较寄存器 1	TP8CCR1				√	0000H
FFFFF61AH	TMP8 计数器读取缓冲寄存器	TP8CNT	R			√	0000H
FFFFF680H	钟表定时器操作模式寄存器	WTM	R/W	√	√		00H
FFFFF690H	TMM0 控制寄存器 0	TM0CTL0		√	√		00H
FFFFF694H	TMM0 比较寄存器 0	TM0CMP0				√	0000H
FFFFF6C0H	振荡稳定时间选择寄存器	OSTS			√		06H
FFFFF6C1H	PLL 入锁时间指定寄存器	PLLS			√		03H
FFFFF6D0H	看门狗定时器模式寄存器 2	WDTM2			√		67H
FFFFF6D1H	看门狗定时器使能寄存器	WDTE			√		9AH
FFFFF6E0H	实时输出缓冲寄存器 0L	RTBL0		√	√		00H
FFFFF6E2H	实时输出缓冲寄存器 0H	RTBH0		√	√		00H
FFFFF6E4H	实时输出端口模式寄存器 0	RTPM0		√	√		00H
FFFFF6E5H	实时输出端口控制寄存器 0	RTPC0		√	√		00H
FFFFF6F0H	实时输出缓冲寄存器 1L	RTBL1		√	√		00H
FFFFF6F2H	实时输出缓冲寄存器 1H	RTBH1		√	√		00H
FFFFF6F4H	实时输出端口模式寄存器 1	RTPM1		√	√		00H
FFFFF6F5H	实时输出端口控制寄存器 1	RTPC1		√	√		00H
FFFFF706H	端口 3 功能控制扩展寄存器 L	PFCE3L		√	√		00H
FFFFF70AH	端口 5 功能控制扩展寄存器	PFCE5		√	√		00H
FFFFF712H	端口 9 功能控制扩展寄存器	PFCE9				√	0000H
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		√	√		00H
FFFFF80CH	内部振荡模式寄存器	RCM		√	√		00H
FFFFF810H	DMA 触发因素寄存器 0	DTFR0		√	√		00H
FFFFF812H	DMA 触发因素寄存器 1	DTFR1		√	√		00H
FFFFF814H	DMA 触发因素寄存器 2	DTFR2		√	√		00H
FFFFF816H	DMA 触发因素寄存器 3	DTFR3		√	√		00H
FFFFF820H	省电模式寄存器	PSMR		√	√		00H
FFFFF822H	时钟控制寄存器	CKC		√	√		0AH
FFFFF824H	锁定寄存器	LOCKR	R	√	√		00H
FFFFF828H	处理器时钟控制寄存器	PCC	R/W	√	√		03H
FFFFF82CH	PLL 控制寄存器	PLLCTL		√	√		01H
FFFFF82EH	CPU 工作时钟状态寄存器	CCLS	R	√	√		00H
FFFFF870H	时钟监视器模式寄存器	CLM	R/W	√	√		00H
FFFFF888H	复位源标志寄存器	RESF		√	√		00H
FFFFF890H	低电压检测寄存器	LVIM		√	√		00H

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF891H	低电压检测电平选择寄存器	LVIS	R/W		√		00H
FFFFF892H	内部 RAM 数据状态寄存器	RAMS		√	√		01H
FFFFF8B0H	预分频模式寄存器 0	PRSM0		√	√		00H
FFFFF8B1H	预分频器比较寄存器 0	PRSCM0			√		00H
FFFFF9FCH	片上调试模式寄存器	OCDM		√	√		01H
FFFFF9FEH	外围仿真寄存器 1	PEMU1 ^注		√	√		00H
FFFFFA00H	UARTA0 控制寄存器 0	UA0CTL0		√	√		10H
FFFFFA01H	UARTA0 控制寄存器 1	UA0CTL1			√		00H
FFFFFA02H	UARTA0 控制寄存器 2	UA0CTL2			√		FFH
FFFFFA03H	UARTA0 选项控制寄存器 0	UA0OPT0		√	√		14H
FFFFFA04H	UARTA0 状态寄存器	UA0STR		√	√		00H
FFFFFA06H	UARTA0 接收数据寄存器	UA0RX	R		√		FFH
FFFFFA07H	UARTA0 发送数据寄存器	UA0TX	R/W		√		FFH
FFFFFA10H	UARTA1 控制寄存器 0	UA1CTL0		√	√		10H
FFFFFA11H	UARTA1 控制寄存器 1	UA1CTL1			√		00H
FFFFFA12H	UARTA1 控制寄存器 2	UA1CTL2			√		FFH
FFFFFA13H	UARTA1 选项控制寄存器 0	UA1OPT0		√	√		14H
FFFFFA14H	UARTA1 状态寄存器	UA1STR		√	√		00H
FFFFFA16H	UARTA1 接收数据寄存器	UA1RX	R		√		FFH
FFFFFA17H	UARTA1 发送数据寄存器	UA1TX	R/W		√		FFH
FFFFFA20H	UARTA2 控制寄存器 0	UA2CTL0		√	√		10H
FFFFFA21H	UARTA2 控制寄存器 1	UA2CTL1			√		00H
FFFFFA22H	UARTA2 控制寄存器 2	UA2CTL2			√		FFH
FFFFFA23H	UARTA2 选项控制寄存器 0	UA2OPT0		√	√		14H
FFFFFA24H	UARTA2 状态寄存器	UA2STR		√	√		00H
FFFFFA26H	UARTA2 接收数据寄存器	UA2RX	R		√		FFH
FFFFFA27H	UARTA2 发送数据寄存器	UA2TX	R/W		√		FFH
FFFFFA30H	UARTA3 控制寄存器 0	UA3CTL0		√	√		10H
FFFFFA31H	UARTA3 控制寄存器 1	UA3CTL1			√		00H
FFFFFA32H	UARTA3 控制寄存器 2	UA3CTL2			√		FFH
FFFFFA33H	UARTA3 选项控制寄存器 0	UA3OPT0		√	√		14H
FFFFFA34H	UARTA3 状态寄存器	UA3STR		√	√		00H
FFFFFA36H	UARTA3 接收数据寄存器	UA3RX	R		√		FFH
FFFFFA37H	UARTA3 发送数据寄存器	UA3TX	R/W		√		FFH
FFFFFC00H	外部中断下降沿指定寄存器 0	INTF0		√	√		00H
FFFFFC06H	外部中断下降沿指定寄存器 3	INTF3		√	√		00H
FFFFFC10H	外部中断下降沿指定寄存器 8	INTF8		√	√		00H
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
FFFFFC30H	外部中断上升沿指定寄存器 8	INTR8		√	√		00H
FFFFFC33H	外部中断上升沿指定寄存器 9H	INTR9H		√	√		00H

注 仅在仿真期间

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFFC60H	端口 0 功能寄存器	PF0	R/W	√	√		00H
FFFFFC66H	端口 3 功能寄存器	PF3				√	0000H
FFFFFC66H	端口 3 功能寄存器 L	PF3L		√	√		00H
FFFFFC67H	端口 3 功能寄存器 H	PF3H		√	√		00H
FFFFFC68H	端口 4 功能寄存器	PF4		√	√		00H
FFFFFC6AH	端口 5 功能寄存器	PF5		√	√		00H
FFFFFC6CH	端口 6 功能寄存器	PF6				√	0000H
FFFFFC6CH	端口 6 功能寄存器 L	PF6L		√	√		00H
FFFFFC6DH	端口 6 功能寄存器 H	PF6H		√	√		00H
FFFFFC70H	端口 8 功能寄存器	PF8		√	√		00H
FFFFFC72H	端口 9 功能寄存器	PF9				√	0000H
FFFFFC72H	端口 9 功能寄存器 L	PF9L		√	√		00H
FFFFFC73H	端口 9 功能寄存器 H	PF9H		√	√		00H
FFFFFD00H	CSIB0 控制寄存器 0	CB0CTL0	R/W	√	√		01H
FFFFFD01H	CSIB0 控制寄存器 1	CB0CTL1		√	√		00H
FFFFFD02H	CSIB0 控制寄存器 2	CB0CTL2			√		00H
FFFFFD03H	CSIB0 状态寄存器	CB0STR		√	√		00H
FFFFFD04H	CSIB0 接收数据寄存器	CB0RX				√	0000H
FFFFFD04H	CSIB0 接收数据寄存器 L	CB0RXL			√		00H
FFFFFD06H	CSIB0 发送数据寄存器	CB0TX				√	0000H
FFFFFD06H	CSIB0 发送数据寄存器 L	CB0TXL			√		00H
FFFFFD10H	CSIB1 控制寄存器 0	CB1CTL0		√	√		01H
FFFFFD11H	CSIB1 控制寄存器 1	CB1CTL1		√	√		00H
FFFFFD12H	CSIB1 控制寄存器 2	CB1CTL2			√		00H
FFFFFD13H	CSIB1 状态寄存器	CB1STR		√	√		00H
FFFFFD14H	CSIB1 接收数据寄存器	CB1RX	R			√	0000H
FFFFFD14H	CSIB1 接收数据寄存器 L	CB1RXL			√		00H
FFFFFD16H	CSIB1 发送数据寄存器	CB1TX	R/W			√	0000H
FFFFFD16H	CSIB1 发送数据寄存器 L	CB1TXL			√		00H
FFFFFD20H	CSIB2 控制寄存器 0	CB2CTL0	R/W	√	√		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	R/W		√		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	R			√	0000H
FFFFFD34H	CSIB3 接收数据寄存器 L	CB3RXL			√		00H

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFFD36H	CSIB3 发送数据寄存器	CB3TX	R/W			√	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	R			√	0000H
FFFFFD44H	CSIB4 接收数据寄存器 L	CB4RXL			√		00H
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
FFFFFD50H	CSIB5 控制寄存器 0	CB5CTL0		√	√		01H
FFFFFD51H	CSIB5 控制寄存器 1	CB5CTL1		√	√		00H
FFFFFD52H	CSIB5 控制寄存器 2	CB5CTL2			√		00H
FFFFFD53H	CSIB5 状态寄存器	CB5STR		√	√		00H
FFFFFD54H	CSIB5 接收数据寄存器	CB5RX	R			√	0000H
FFFFFD54H	CSIB5 接收数据寄存器 L	CB5RXL			√		00H
FFFFFD56H	CSIB5 发送数据寄存器	CB5TX	R/W			√	0000H
FFFFFD56H	CSIB5 发送数据寄存器 L	CB5TXL			√		00H
FFFFFD80H	IIC 移位寄存器 0	IIC0			√		00H
FFFFFD82H	IIC 控制寄存器 0	IICC0		√	√		00H
FFFFFD83H	从地址寄存器 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	从地址寄存器 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
FFFFDDAAH	IIC 标志寄存器 2	IICF2	R/W	√	√		00H
FFFFFBEH	外部总线接口模式控制寄存器	EXIMC		√	√		00H

3.4.7 专用寄存器

专用寄存器是受到保护的寄存器，可以防止程序挂起引起的数据非法写入。V850ES/JJ3 具有以下八个专用寄存器。

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

此外，还提供 PRCDM 寄存器来防止对专用寄存器的非法写操作，这样应用系统就不会因为程序挂起而无意停止工作。对专用寄存器的写操作需要特定流程实现，非法保存操作将报告给 SYS 寄存器中。

(1) 设置专用寄存器数据

按照以下流程设置专用寄存器数据。

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

[实例] 使用 PSC 寄存器（设置待机模式）

```

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

```

读取专用寄存器时没有特定流程。

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

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

(2) 命令寄存器 (PRCMD)

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

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

复位后: 不确定 W 地址: FFFFFFF1FCH

	7	6	5	4	3	2	1	0
PRCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

(3) 系统状态寄存器 (SYS)

该寄存器包含了整个系统的操作状态的状态标志。
该寄存器可进行字节读写或者按位读写。
系统复位后，该寄存器的值置为 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		没有发生保护错误						
1		发生保护错误						

PRERR 标志的操作是按下列条件进行的。

(a) 设置条件 (PRERR 标志=1)

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

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

(b) 清零条件 (PRERR 标志=0)

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

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

3.4.8 注意事项

(1) 需要先行设置的寄存器

当使用 V850ES/JJ3 单片机时，必需确保首先对下列寄存器进行设置。

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

设置 VSWC、OCDM 和 WDTM2 寄存器后，再根据需要对其它寄存器进行设置。

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

(a) 系统等待控制寄存器 (VSWC)

VSWC 寄存器控制访问片上外围 I/O 寄存器时的总线等待时间。

访问片上外围 I/O 寄存器（没有等待周期）需要三个时钟。根据工作频率，V850ES/JJ3 需要不同的等待周期。按照下表设置不同工作频率下的 VSWC 寄存器值。

VSWC 寄存器按字节进行读写（地址：FFFFF06EH，默认值：77H）。

工作频率 (f _{CLK})	VSWC 的设置值	等待周期
32 kHz ≤ f _{CLK} < 16.6 MHz	00H	0（无等待）
16.6 MHz ≤ f _{CLK} ≤ 25 MHz	01H	1
25 MHz ≤ f _{CLK} ≤ 32 MHz	11H	2

(b) 片上调试模式寄存器 (OCDM)

详情参见第二十八章 片上调试功能。

(c) 看门狗定时器模式寄存器 2 (WDTM2)

WDTM2 寄存器用以设置看门狗定时器 2 的溢出时间和工作时钟。

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

详情参见第十一章 看门狗定时器 2 的功能。

(2) 访问指定的片上外围 I/O 寄存器

本产品有两种类型的内部系统总线。

一种是 CPU 总线，另外一种是用来与低速外围硬件接口的外围总线。

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

当执行实时处理任务时，需要特别注意这种情况。

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

访问条件以及如何计算需要插入的等待状态数量（CPU 时钟数）如下表所示：

外围功能	寄存器名称	访问类型	k
16 定时器/事件计数器 P (TMP) (n = 0 至 8)	TPnCNT	读取	1 或 2
	TPnCCR0, TPnCCR1	写入	• 第一次访问：不等待 连续写入：3 或 4
		读取	1 或 2
16 位定时器/事件计数器 Q (TMQ)	TQ0CNT	读取	1 或 2
	TQ0CCR0 至 TQ0CCR3	写入	• 第一次访问：不等待 • 连续写入：3 或 4
		读取	1 或 2
看门狗定时器 2 (WDT2)	WDTM2	写入 (WDT2 工作时)	3
实时输出功能 (RTO) (n = 0, 1)	RTBLn, RTBHn	写入 (RTPCn.RTPOEn 位 = 0 时)	1
A/D 转换器	ADA0M0	读取	1 或 2
	ADA0CR0 至 ADA0CR15	读取	1 或 2
	ADA0CR0H 至 ADA0CR15H	读取	1 或 2
I ² C00 至 I ² C02	IICS0 至 IICS2	读取	1
CRC	CRCD	写入	1

访问时必需的时钟数目 = 3 + i + j + (2 + j) × k

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

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

备注 i: VSWC 寄存器高 4 位的值 (0 或 1)

j: VSWC 寄存器低 4 位的值 (0 或 1)

(3) sld 指令和中断请求发生冲突的限制**(a) 描述**

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

指令 <1>

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

指令<2>

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

<举例>

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

 •

 •

<ii> mov r10, r28

<iii> sld.w 0x28, r10

(b) 对策

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

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

第四章 端口功能

4.1 特性

- I/O 端口：128
 - 5 V 耐压/N 沟道漏极开路输出可以切换：60（端口 0、3 至 6、8、9）
- 可按位指定输入/输出

4.2 端口基本配置

V850ES/JJ3 总共包括 128 个输入/输出端口，由端口 0，1，3 至 9，CD，CM，CS，CT，DH 和 DL 组成。端口配置如下所示：

图 4-1. 端口配置图

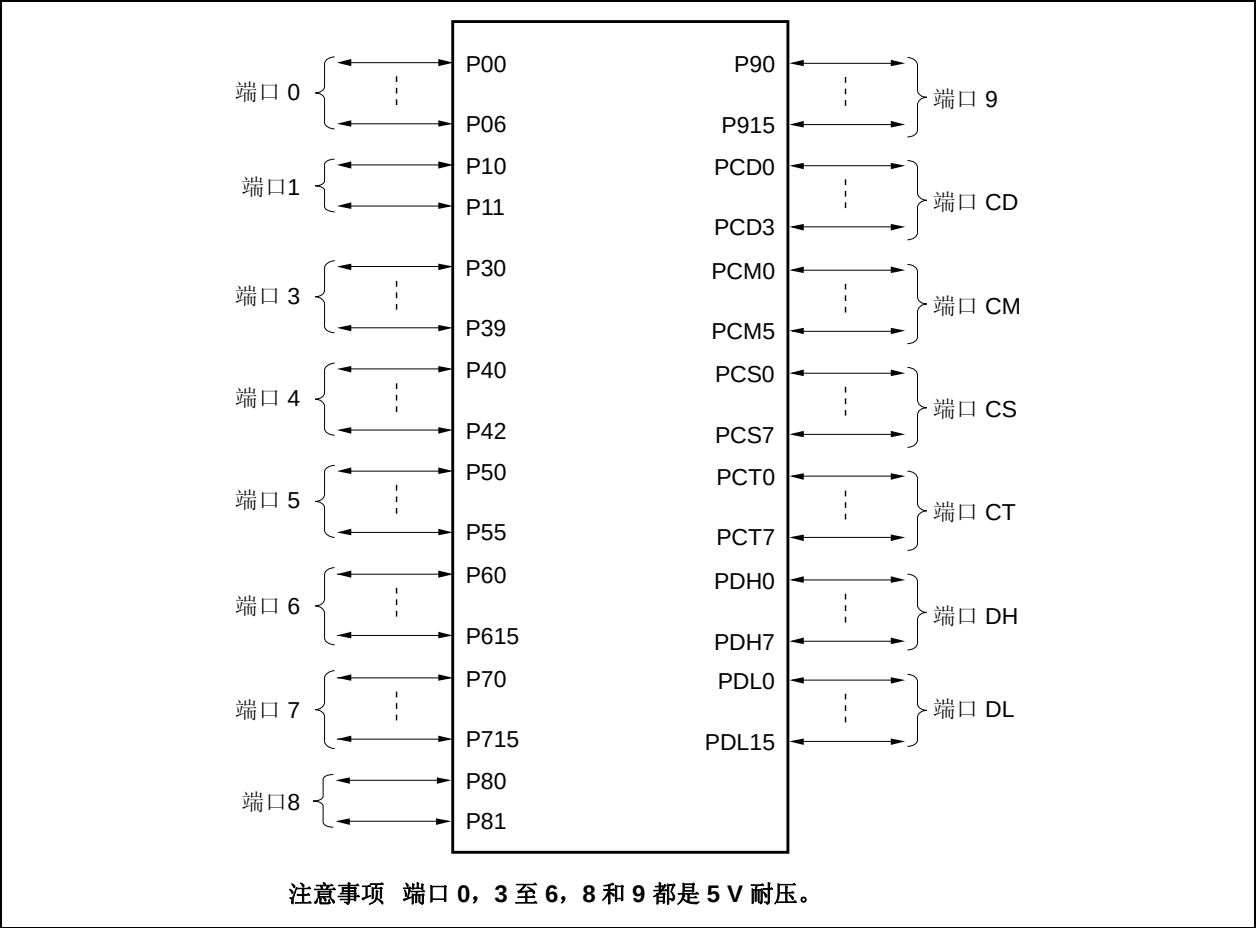


表 4-1. 引脚的输入/输出缓冲器电源供应

供电	对应引脚
AVREF0	端口 7
AVREF1	端口 1
EVDD	RESET， 端口 0，3 至 6，8，9，CD，CM，CS，CT，DH 和 DL

4.3 端口配置

表 4-2. 端口配置

项目	配置
控制寄存器	端口 n 模式寄存器 (PMn: n = 0, 1, 3 至 9, CD, CM, CS, CT, DH, DL) 端口 n 模式控制寄存器 (PMCn: n = 0, 3 至 6, 8, 9, CM, CS, CT, DH, DL) 端口 n 功能控制寄存器 (PFCn: n = 0, 3 至 6, 9) 端口 n 功能控制扩展寄存器 (PFCEn: n = 3, 5, 9) 端口 n 功能寄存器 (PFn: n = 0, 3 至 6, 8, 9)
端口	I/O: 128

(1) 端口 n 寄存器 (Pn)

通过写入或读取 Pn 寄存器，从外部器件输入数据，或输出数据到外部器件。

Pn 寄存器由一个保存输出数据的端口锁存器和一个读取引脚状态的电路组成。

Pn 寄存器的每一位对应端口的一个引脚，而且可以按位读出或写入。

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

Pnm	输出数据控制 (输出模式下)
0	输出0。
1	输出 1。

无论 PMCn 寄存器如何设置，对 Pn 寄存器的数据写入或读取如下所示：

表 4-3. Pn 寄存器的写入/读出

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

注 写入到输出锁存器的值会一直保留，直到有新的数值写入输出锁存器。

(2) 端口 n 模式寄存器 (PMn)

PMn 寄存器用于指定对应端口引脚的输入或输出模式。
该寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定为输入或输出模式。

复位后: FFH								R/W
	7	6	5	4	3	2	1	0
PMn	PMn7	PMn6	PMn5	PMn4	PMn3	PMn2	PMn1	PMn0

PMnm	输入/输出模式控制
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模式。
该寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定端口的工作模式。

复位后: 00H								R/W
PMCn	7	6	5	4	3	2	1	0
	PMCn7	PMCn6	PMCn5	PMCn4	PMCn3	PMCn2	PMCn1	PMCn0
PMCnm	操作模式说明							
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 寄存器用来指定需要使用的引脚复用功能。
该寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定端口引脚的复用功能。

复位后: 00H R/W

7	6	5	4	3	2	1	0
PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0

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

复位后: 00H R/W

	7	6	5	4	3	2	1	0
PFCEn	PFCEn7	PFCEn6	PFCEn5	PFCEn4	PFCEn3	PFCEn2	PFCEn1	PFCEn0

	7	6	5	4	3	2	1	0
PFCn	PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0

PFCEnm	PFCnm	复用功能说明
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 寄存器指定端口是正常输出还是 N 沟道漏极开路输出。
该寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定端口引脚的输出模式。

复位后: 00H R/W

PFn

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

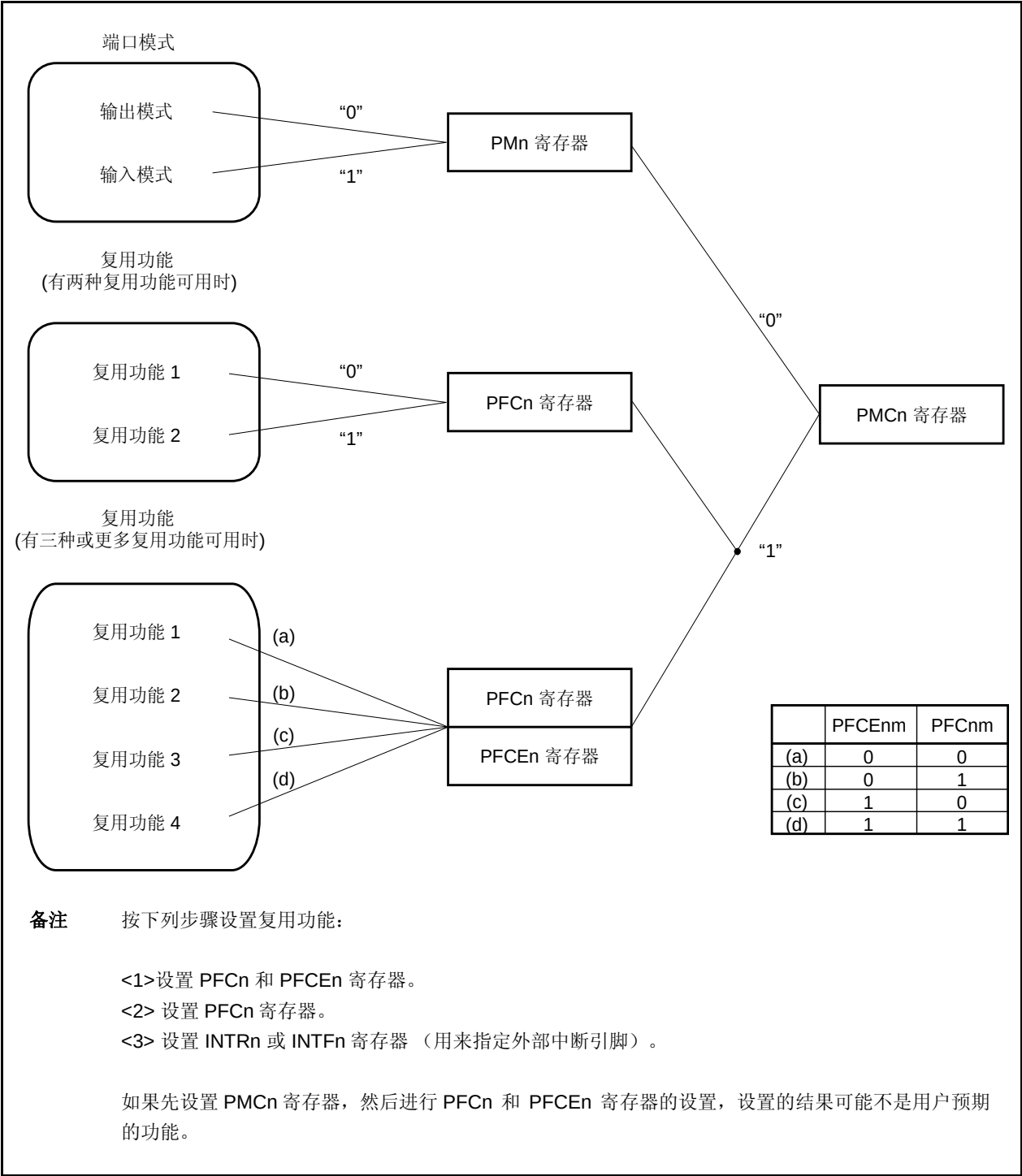
PFnm ^注	正常输出/N沟道漏极开路输出控制
0	正常输出 (CMOS 输出)
1	N沟道漏极开路输出

注 在端口模式下 (PMCnm 位=0)，只有 PMn 寄存器的 PMnm 位为 0 时 (指定为输出模式时)，PFn 寄存器的 PFnm 位才有效。当 PMnm 位为 1 (指定为输入模式时)，PFn 寄存器设置值无效。

(7) 端口设置

端口设置如下所示：

图 4-2. 各寄存器的设置和引脚功能



4.3.1 端口 0

端口 0 有 7 位输入输出端口，可以按位控制输入输出设置。
端口 0 包括以下复用功能引脚：

表 4-4. 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	框图类型
P00	6	TIP61/TOP61	I/O	可选作 N 沟道漏极开路输出	G-1
P01	7	TIP60/TOP60	I/O		G-1
P02	17	NMI	输入		L-1
P03	18	INTP0/ADTRG	输入		N-1
P04	19	INTP1	输入		L-1
P05	20	INTP2/ $\overline{\text{DRST}}^{\#}$	输入		AA-1
P06	21	INTP3	输入		L-1

注 $\overline{\text{DRST}}$ 引脚用于片上调试。
如果不使用片上调试功能，在 $\overline{\text{RESET}}$ 引脚的复位信号释放后，在 OCDM.OCDM0 位清 0 之前，将 P05/INTP2/ $\overline{\text{DRST}}$ 引脚固定为低电平。
细节请参见 4.6.3 片上调试引脚的注意事项。

注意事项 在复用功能引脚的输入模式下，P00 至 P06 具有迟滞特性，但端口模式没有迟滞特性。

(1) 端口 0 寄存器 (P0)

复位后: 00H (输出锁存) R/W 地址: FFFFF400H

	7	6	5	4	3	2	1	0
P0	0	P06	P05	P04	P03	P02	P01	P00

P0n	输出数据控制(输出模式下) (n = 0 至 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	PM01	PM00

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

(3) 端口 0 模式控制寄存器 (PMC0)

复位后: 00H R/W 地址: FFFFF440H

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

PMC06	P06 引脚操作模式说明	
0	I/O 端口	
1	INTP3 输入	

PMC05	P05 引脚操作模式说明	
0	I/O 端口	
1	INTP2 输入	

PMC04	P04 引脚操作模式说明	
0	I/O 端口	
1	INTP1 输入	

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

PMC02	P02 引脚操作模式说明	
0	I/O 端口	
1	NMI 输入	

PMC01	P01 引脚操作模式说明	
0	I/O 端口	
1	TIP60 输入/TOP60 输出	

PMC00	P00 引脚操作模式说明	
0	I/O port	
1	TIP61 输入/TOP61 输出	

注意事项 OCDM.OCDM0 位=1 时，无论 PMC05 位的值是多少，P05/INTP2/ $\overline{\text{DRST}}$ 引脚均作为 $\overline{\text{DRST}}$ 引脚使用。

(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	PFC01	PFC00

PFC03	P03 引脚复用功能说明
0	INTP0 输入
1	ADTRG 输入

PFC01	P01 引脚复用功能说明
0	TIP60 输入
1	TOP60 输出

PFC00	P00 引脚复用功能说明
0	TIP61 输入
1	TOP61 输出

(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	PF01	PF00

PF0n	正常输出或N沟道漏极开路输出的控制 (n=0至6)
0	正常输出 (CMOS输出)
1	N沟道漏极开路输出

注意事项 若某个输出引脚被上拉至 EV_{DD} 或更高电压时, 确保将 PF0n 位设置为 1。

4.3.2 端口 1

端口 1 有 2 位输入输出端口，可以按位控制输入输出设置。

端口 1 包括以下复用功能引脚：

表 4-5.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	输入/输出	备注	框图类型
P10	3	ANO0	输出	—	A-2
P11	4	ANO1	输出	—	A-2

(1) 端口 1 寄存器 (P1)

复位后: 00H (输出锁存) 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 D/A 转换期间，不要读取或写入 P1 寄存器（参见 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. P1n 用作复用功能 (ANOn 引脚输出) 时，将 PM1n 位设置为 1。
2. 当使用 P10/ANO0 和 P11/ANO1 引脚之一作为 I/O 端口，而另一个用作 D/A 输出引脚。在 D/A 输出期间，如果端口 I/O 电平不改变，则在应用系统中可以这样使用。

4.3.3 端口 3

端口 3 有 10 位输入输出端口，可以按位控制输入输出设置。

端口 3 包括以下复用功能引脚：

表 4-6. 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	输入/输出	备注	框图类型
P30	25	TXDA0/SOB4	输出	可设置为 N 沟道漏极开路输出	G-3
P31	26	RXDA0/INTP7/SIB4	输入		N-3
P32	27	ASCKA0/SCKB4/TIP00/TOP00	I/O		U-1
P33	28	TIP01/TOP01	I/O		G-1
P34	29	TIP10/TOP10	I/O		G-1
P35	30	TIP11/TOP11	I/O		G-1
P36	31	—	—		C-1
P37	32	—	—		C-1
P38	35	TXDA2/SDA00	I/O		G-12
P39	36	RXDA2/SCL00	I/O		G-6

注意事项 在复用功能引脚的输入模式下，P31 至 P35，P38 和 P39 具有迟滞特性，但端口模式没有迟滞特性。

(1) 端口 3 寄存器 (P3)

复位后: 0000H (输出锁存) 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 至 9)						
0	输出 0						
1	输出 1						

注意事项 请确保将第 15 位至第 10 位, 第 7 位和第 6 位 设置为 “0”。

- 备注**
1. P3 寄存器可以按 16 位宽度读出或写入。
但是当寄存器 P3 的高 8 位和低 8 位分别用作 P3H 寄存器和 P3L 寄存器时, P3 可以按字节或按位读出或写入。
 2. 若要按字节或按位读取或写入 P3 的第 8 位至第 15 位, 将这些位作为 P3H 寄存器的第 0 位至第 7 位。

(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 至 9)						
0	输出模式						
1	输入模式						

- 备注**
1. PM3 寄存器 可以按 16 位宽度读出或写入。
但是当寄存器 PM3 的高 8 位和低 8 位分别用作 PM3H 寄存器和 PM3L 寄存器时, PM3 可以按字节或按位读出或写入。
 2. 若要按字节或按位读取或写入 PM3 的第 8 位至第 15 位, 将这些位作为 PM3H 寄存器的第 0 位至第 7 位。

备注

1. PMC3 寄存器 可以按 16 位宽度读出或写入。
然而，寄存器 PMC3 的高 8 位和低 8 位分别用作 PMC3H 寄存器和 PMC3L 寄存器时， PMC3 可以按字节或按位读出或写入。
2. 若要按字节或按位读取或写入 PMC3 的第 8 位至第 15 位，将这些位作为 PMC3H 寄存器的第 0 位至第 7 位。

(4) 端口 3 功能控制寄存器 (PFC3)

复位后: 0000H	R/W	地址:	PFC3 FFFFF466H, PFC3L FFFFF466H, PFC3L FFFFF467H							
			15	14	13	12	11	10	9	8
PFC3 (PFC3H)			0	0	0	0	0	0	PFC39	PFC38
			7	6	5	4	3	2	1	0
(PFC3L)			0	0	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

备注

1.

关于复用功能规范的详情，参见 4.3.3（6） 端口 3 复用功能规范。

2.

PFC3 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PFC3 的高 8 位和低 8 位分别用作 PFC3H 寄存器和 PFC3L 寄存器时，PFC3 可以按字节或按位读出或写入。

3.

若要按字节或按位读取或写入 PFC3 的第 8 位至第 15 位，将这些位作为 PFC3H 寄存器的第 0 位至第 7 位。

(5) 端口 3 功能控制扩展寄存器 L (PFCE3L)

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

注意事项 请确保将第 7 位至第 3 位，第 1 位和第 0 位设置为 “0”。

备注 关于复用功能规范的详情，参见 4.3.3（6） 端口 3 复用功能规范。

(6) 端口 3 复用功能规范

PFC39	P39 引脚复用功能说明
0	RXDA2 输入
1	SCL00 输入

PFC38	P38 引脚复用功能说明
0	TXDA2 输出
1	SDA00 I/O

PFC35	P35 引脚复用功能说明
0	TIP11 输入
1	TOP11 输出

PFC34	P34 引脚复用功能说明
0	TIP10 输入
1	TOP10 输出

PFC33	P33 引脚复用功能说明
0	TIP01 输入
1	TOP01 输出

PFCE32	PFC32	P32 引脚复用功能说明
0	0	ASCKA0 输入
0	1	$\overline{\text{SCKB4}}$ I/O
1	0	TIP00 输入
1	1	TOP00 输出

PFC31	P31 引脚复用功能说明
0	RXDA0 输入/INTP7 ^注 输入
1	SIB4 输入

PFC30	P30 引脚复用功能说明
0	TXDA0 输出
1	SOB4 输出

注 INTP7 引脚和 RXDA0 引脚是复用功能引脚。当该引脚用作 RXDA0 引脚时，禁止对 INTP7 复用功能边缘检测（INTF3.INTF31 位和 INTR3.INTR31 位清 0）。当用作 INTP7 引脚，停止 UARTA0 接收（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	正常输出或 N沟道漏极开路输出控制 (n = 0 至 5, 8, 9)
0	正常输出 (CMOS 输出)
1	N沟道漏极开路输出

注意事项

若某个输出引脚被上拉至 EV_{DD} 或更高电压时，确保将 PF3n 位设置为 1。

备注

1. PF3 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PF3 的高 8 位和低 8 位分别用作 PF3H 寄存器和 PF3L 寄存器时，PF3 可以按字节或按位读出或写入。

2. 若要按字节或按位读取或写入 PF3 的第 8 位至第 15 位，将这些位作为 PF3H 寄存器的第 0 位至第 7 位。

4.3.4 端口 4

端口 4 有 3 位输入输出端口，可以按位控制输入输出设置。

端口 4 包括以下复用功能引脚：

表 4-7. 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	输入/输出	备注	框图类型
P40	22	SIB0/SDA01	I/O	可选作 N 沟道漏极开路输出	G-6
P41	23	SOB0/SCL01	I/O		G-12
P42	24	SCKB0	I/O		E-3

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

(1) 端口 4 寄存器 (P4)

复位后: 00H (输出锁存) R/W 地址: FFFFF408H

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

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

(2) 端口 4 模式寄存器 (PM4)

复位后: FFH R/W 地址: 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 至 2)
0	输出模式
1	输入模式

(3) 端口 4 模式控制寄存器 (PMC4)

复位后: 00H R/W 地址: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	0	0	0	0	0	PMC42	PMC41	PMC40

PMC42	P42 引脚操作模式说明
0	I/O 端口
1	$\overline{\text{SCKB0}}$ I/O

PMC41	P41 引脚操作模式说明
0	I/O 端口
1	SOB0 输出/SCL01 I/O

PMC40	P40 引脚操作模式说明
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	P41 引脚复用功能说明
0	SOB0 输出
1	SCL01 I/O

PFC40	P40 引脚复用功能说明
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: FFFFC68H

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

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

注意事项 若某个输出引脚被上拉至 EV_{DD} 或更高电压时, 确保将 PF4n 位设置为 1。

4.3.5 端口 5

端口 5 有 6 位输入输出端口，可以按位控制输入输出设置。

端口 5 包括以下复用功能引脚：

表 4-8. 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	输入/输出	备注	框图类型
P50	37	TIQ01/KR0/TOQ01/RTP00	I/O	可选作 N 沟道漏极开路输出	U-5
P51	38	TIQ02/KR1/TOQ02/RTP01	I/O		U-5
P52	39	TIQ03/KR2/TOQ03/RTP02/DDI ^注	I/O		U-6
P53	40	SIB2/KR3/TIQ00/TOQ00/RTP03/DDO ^注	I/O		U-7
P54	41	SOB2/KR4/RTP04/DCK ^注	I/O		U-8
P55	42	SCKB2/KR5/RTP05/DMS ^注	I/O		U-9

注 DDI, DDO, DCK 和 DMS 引脚用于片上调试。

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

详情参见 4.6.3 片上调试引脚的注意事项。

注意事项 1. 电源打开后的复位期间，P53 引脚可能暂时输出不确定电平。

2. 在复用功能引脚输入模式下，P50 至 P55 具有迟滞特性，但在端口模式下没有迟滞特性。

(1) 端口 5 寄存器 (P5)

复位后: 00H (输出锁存) 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 至 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 至 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	P55 引脚操作模式说明							
0	I/O 端口							
1	SCKB2 I/O/KR5 输入/RTP05 输出							
PMC54	P54 引脚操作模式说明							
0	I/O 端口							
1	SOB2 输出/KR4 输入/RTP04 输出							
PMC53	P53 引脚操作模式说明							
0	I/O 端口							
1	SIB2 输入/KR3 输入/TIQ00 输入/TOQ00 输出/RTP03 输出							
PMC52	P52 引脚操作模式说明							
0	I/O 端口							
1	TIQ03 输入/KR2 输入/TOQ03 输出/RTP02 输出							
PMC51	P51 引脚操作模式说明							
0	I/O 端口							
1	TIQ02 输入/KR1 输入/TOQ02 输出/RTP01 输出							
PMC50	P50 引脚操作模式说明							
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 关于复用功能规范的详情，参见 4.3.5 (6) 端口 5 复用功能规范。

(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 关于复用功能规范的详情, 参见 4.3.5 (6) 端口 5 复用功能规范。

(6) 端口 5 复用功能规范

PFCE55	PFC55	P55 引脚复用功能说明
0	0	$\overline{\text{SCKB2}}$ I/O
0	1	KR5 输入
1	0	禁止设置
1	1	RTP05 输出

PFCE54	PFC54	P54 引脚复用功能说明
0	0	SOB2 输出
0	1	KR4 输入
1	0	禁止设置
1	1	RTP04 输出

PFCE53	PFC53	P53 引脚复用功能说明
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 引脚复用功能说明
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 引脚复用功能说明
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 引脚复用功能说明
0	0	禁止设置
0	1	TIQ01 输入/KR0 ^注 输入
1	0	TOQ01 输出
1	1	RTP00 输出

注 KRn 引脚和 TIQ0m 引脚是复用功能引脚。当该引脚用作 TIQ0m 引脚时，禁止 KRn 引脚的按键返回检测功能（将 KRM.KRMn 位清除为 0）。同样，当该引脚用作 KRn 引脚时，禁止 TIQ0m 引脚边沿检测功能（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

PF5

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

PF5n	正常输出 或 N沟道漏极开路输出 控制(n = 0 至 5)
0	正常输出 (CMOS 输出)
1	N沟道漏极开路输出

注意事项 若某个输出引脚被上拉至 EV_{DD}或更高电压时，确保将 PF5n 位设置为 1。

4.3.6 端口 6

端口 6 有 16 位输入输出端口，可以按位控制输入输出设置。

端口 6 包括以下复用功能引脚：

表 4-9. 端口 6 复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	输入/输出	备注	框图类型
P60	43	RTP10	输出	可选作 N 沟道漏极开路输出	E-2
P61	44	RTP11	输出		E-2
P62	45	RTP12	输出		E-2
P63	46	RTP13	输出		E-2
P64	47	RTP14	输出		E-2
P65	48	RTP15	输出		E-2
P66	49	SIB5	输入		E-1
P67	50	SOB5	输出		E-2
P68	51	SCKB5	I/O		E-3
P69	52	TIP70/TOP70	I/O		G-1
P610	53	TIP71	输入		E-1
P611	54	TOP71	输出		E-2
P612	55	TIP80/TOP80	I/O		G-1
P613	56	TIP81/TOP81	I/O		G-1
P614	57	—	—		C-1
P615	58	—	—		C-1

注意事项 在复用功能引脚的输入模式下，P66，P68 至 P610，P612 和 P613 具有迟滞特性，但在端口模式下没有迟滞特性。

(1) 端口 6 寄存器 (P6)

复位后: 0000H (输出锁存) R/W 地址: P6 FFFFF40CH
P6L FFFFF40CH, P6LH FFFFF40DH

	15	14	13	12	11	10	9	8
P6 (P6H)	P615	P614	P613	P612	P611	P610	P69	P68

	7	6	5	4	3	2	1	0
(P6L)	P67	P66	P65	P64	P63	P62	P61	P60

P6n	输出数据控制 (输出模式下) (n = 0 至 15)
0	输出 0
1	输出 1

备注

1. P6 寄存器可以按 16 位宽度进行读写。
但是当寄存器 P6 的高 8 位和低 8 位分别用作 P6H 寄存器和 P6L 寄存器时, P6 可以按字节或按位读出或写入。
2. 若要按字节或按位读取或写入 P6 的第 8 位至第 15 位, 将这些位作为 P6H 寄存器的第 0 位至第 7 位。

(2) 端口 6 模式寄存器 (PM6)

复位后: FFFFH R/W 地址: PM6 FFFFF42CH
PM6L FFFFF42CH, PM6H FFFFF42DH

	15	14	13	12	11	10	9	8
PM6 (PM6H)	PM615	PM614	PM613	PM612	PM611	PM610	PM69	PM68

	7	6	5	4	3	2	1	0
(PM6L)	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60

PM6n	I/O 模式控制 (n = 0 至 15)
0	输出模式
1	输入模式

备注

1. PM6 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PM6 的高 8 位和低 8 位分别用作 PM6H 寄存器和 PM6L 寄存器时, PM6 可以按字节和按位读出或写入。
2. 若要按字节或按位读取或写入 PM6 的第 8 位至第 15 位, 将这些位作为 PM6H 寄存器的第 0 位至第 7 位。

(3) 端口 6 模式控制寄存器 (PMC6)

复位后: 0000H R/W 地址: PMC6 FFFFF44CH

PMC6L FFFFF44CH, PMC6H FFFFF44DH

	15	14	13	12	11	10	9	8
PMC6 (PMC6H)	0	0	PMC613	PMC612	PMC611	PMC610	PMC69	PMC68
	7	6	5	4	3	2	1	0
(PMC6L)	PMC67	PMC66	PMC65	PMC64	PMC63	PMC62	PMC61	PMC60
PMC613	P613 引脚操作模式说明							
0	I/O 端口							
1	TIP81 输入TOP81 输出							
PMC612	P612 引脚操作模式说明							
0	I/O 端口							
1	TIP80 输入/TOP80 输出							
PMC611	P611 引脚操作模式说明							
0	I/O 端口							
1	TOP71 输出							
PMC610	P610 引脚操作模式说明							
0	I/O 端口							
1	TIP71 输入							
PMC69	P69 引脚操作模式说明							
0	I/O 端口							
1	TIP70 输入/TOP70 输出							
PMC68	P68 引脚操作模式说明							
0	I/O 端口							
1	SCKB5 I/O							
PMC67	P67 引脚操作模式说明							
0	I/O 端口							
1	SOB5 输出							
PMC66	P66 引脚操作模式说明							
0	I/O 端口							
1	SIB5 输入							
PMC6m	P6m 引脚操作模式说明 (m = 0 至 5)							
0	I/O 端口							
1	RTP1m 输出							

备注

1. PMC6 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PMC6 的高 8 位和低 8 位分别用作 PMC6H 寄存器和 PMC6L 寄存器时, PMC6 可以按字节或按位读出或写入。
2. 若要按字节或按位读取或写入 PMC6 的第 8 位至第 15 位, 将这些位作为 PMC6H 寄存器的第 0 位至第 7 位。

(4) 端口 6 功能控制寄存器 H (PFC6H)

复位后: 00H R/W 地址: FFFFF46DH

	15	14	13	12	11	10	9	8
PFC6H	0	0	PFC613	PFC612	0	0	PFC69	0

PFC613	P613 引脚复用功能说明
0	TIP81 输入
1	TOP81 输出

PFC612	P612 引脚复用功能说明
0	TIP80 输入
1	TOP80 输出

PFC69	P69 引脚复用功能说明
0	TIP70 输入
1	TOP70 输出

(5) 端口 6 功能寄存器 (PF6)

复位后: 0000H R/W 地址: PF6 FFFFFC6CH
PF6L FFFFFC6CH, PF6H FFFFFC6DH

	15	14	13	12	11	10	9	8
PF6 (PF6H)	PF615	PF614	PF613	PF612	PF611	PF610	PF69	PF68

	7	6	5	4	3	2	1	0
(PF6L)	PF67	PF66	PF65	PF64	PF63	PF62	PF61	PF60

PF6n	正常输出 或 N沟道漏极开路输出控制 (n = 0 至 15)
0	正常输出 (CMOS 输出)
1	N沟道漏极开路输出

注意事项 若一个输出引脚被上拉至 EV_{DD} 或更高电压时, 确保将 PF6n 位设置为 1。**备注**

1. PF6 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PF6 的高 8 位和低 8 位分别用作 PF6H 寄存器和 PF6L 寄存器时, PF6 可以按字节和按位读出或写入。
2. 若要按字节或按位读取或写入 PF6 的第 8 位至第 15 位, 将这些位作为 PF6H 寄存器的第 0 位至第 7 位。

4.3.7 端口 7

端口 7 有 16 位输入输出端口，可以按位控制输入输出设置。

端口 7 包括以下复用功能引脚：

表 4-10. 端口 7 复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	输入/输出	备注	框图类型
P70	144	ANI0	输入	-	A-1
P71	143	ANI1	输入		A-1
P72	142	ANI2	输入		A-1
P73	141	ANI3	输入		A-1
P74	140	ANI4	输入		A-1
P77	139	ANI5	输入		A-1
P76	138	ANI6	输入		A-1
P77	137	ANI7	输入		A-1
P78	136	ANI8	输入		A-1
P79	135	ANI9	输入		A-1
P710	134	ANI10	输入		A-1
P711	133	ANI11	输入		A-1
P712	132	ANI12	输入		A-1
P713	131	ANI13	输入		A-1
P714	130	ANI14	输入		A-1
P715	129	ANI15	输入		A-1

(1) 端口 7 寄存器 H, 端口 7 寄存器 L (P7H, P7L)

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

	7	6	5	4	3	2	1	0
P7H	P715	P714	P713	P712	P711	P710	P79	P78

	7	6	5	4	3	2	1	0
P7L	P77	P76	P75	P74	P73	P72	P71	P70

P7n	输出数据控制 (输出模式下) (n = 0 至 15)
0	输出 0
1	输出 1

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

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

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

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

	7	6	5	4	3	2	1	0
PM7H	PM715	PM714	PM713	PM712	PM711	PM710	PM79	PM78

	7	6	5	4	3	2	1	0
PM7L	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	I/O 模式控制 (n = 0 至 15)
0	输出模式
1	输入模式

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

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

4.3.8 端口 8

端口 8 有 2 位输入输出端口，可以按位控制输入输出设置。

端口 8 包括以下复用功能引脚：

表 4-11. 端口 8 复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	输入/输出	备注	框图类型
P80	59	RXDA3/INTP8	输入	可选作 N 沟道漏极开路输出	L-2
P81	60	TXDA3	输出		E-2

注意事项 在复用功能引脚的输入模式下，P80 具有迟滞特性，但在端口模式下没有迟滞特性。

(1) 端口 8 寄存器(P8)

复位后: 00H (输出锁存) R/W 地址: FFFFF410H							
P8	7	6	5	4	3	2	1
	0	0	0	0	0	0	P81
							P80
P8n	输出数据控制 (n = 0, 1)						
0	输出 0						
1	输出 1						

(2) 端口 8 模式寄存器 (PM8)

复位后: FFH R/W 地址: FFFFF430H							
PM8	7	6	5	4	3	2	1
	1	1	1	1	1	1	PM81
							PM80
PM8n	I/O 模式控制 (n = 0, 1)						
0	输出模式						
1	输入模式						

(3) 端口 8 模式控制寄存器 (PMC8)

复位后: 00H R/W 地址: FFFFF450H

	7	6	5	4	3	2	1	0
PMC8	0	0	0	0	0	0	PMC81	PMC80

PMC81	P81 引脚操作模式说明
0	I/O 端口
1	TXDA3 输出

PMC80	P80 引脚操作模式说明
0	I/O 端口
1	RXDA3 输入/INTP8 ^注 输入

注 INTP8 引脚及 RXDA3 引脚为复用功能引脚。当该引脚用作 RXDA3 引脚时，请禁止 INTP8 复用功能引脚的边沿检测（清除 INTF8.INTF80 位 = 0 和 INTR8.INTR80 位 = 0）。当该引脚被用作 INTP8 引脚时，停止 UARTA3 接收（清除 UA3CTL0.UA3RXE 位为 0）。

(4) 端口 8 功能寄存器 (PF8)

复位后: 00H R/W 地址: FFFFFC70H

	7	6	5	4	3	2	1	0
PF8	0	0	0	0	0	0	PF81	PF80

PF8n	正常输出或 N沟道漏极开路输出控制 (n = 0, 1)
0	正常输出 (CMOS 输出)
1	N沟道漏极开路输出

注意事项 若一个输出引脚被上拉至 E_{VDD} 或更高电压时，确保将 PF8n 位设置为 1。

4.3.9 端口 9

端口 9 有 16 位输入输出端口，可以按位控制输入输出设置。

端口 9 包括以下复用功能引脚：

表 4-10. 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	输入/输出	备注	框图类型
P90	61	A0/KR6/TXDA1/SDA02	I/O	可选作 N 沟道漏极开路输出	U-10
P91	62	A1/KR7/RXDA1/SCL02	I/O		U-11
P92	63	A2/TIP41/TOP41	I/O		U-12
P93	64	A3/TIP40/TOP40	I/O		U-12
P94	65	A4/TIP31/TOP31	I/O		U-12
P95	66	A5/TIP30/TOP30	I/O		U-12
P96	67	A6/TIP21/TOP21	I/O		U-13
P97	68	A7/SIB1/TIP20/TOP20	I/O		U-14
P98	69	A8/SOB1	输出		G-3
P99	70	A9/ $\overline{\text{SCKB1}}$	I/O		G-5
P910	71	A10/SIB3	I/O		G-2
P911	72	A11/SOB3	输出		G-3
P912	73	A12/ $\overline{\text{SCKB3}}$	I/O		G-5
P913	74	A13/INTP4	I/O		N-2
P914	75	A14/INTP5/TIP51/TOP51	I/O		U-15
P915	76	A15/INTP6/TIP50/TOP50	I/O		U-15

注意事项 在复用功能引脚的输入模式下，P90 至 P97，P99，P910 以及 P912 至 P915 具有迟滞特性，但在端口模式下没有迟滞特性。

(1) 端口 9 寄存器 (P9)

复位后: 0000H (输出锁存) R/W 地址: P9 FFFFF412H,
P9L FFFFF412H, P9H FFFFF413H

	15	14	13	12	11	10	9	8
P9 (P9H)	P915	P914	P913	P912	P911	P910	P99	P98

	7	6	5	4	3	2	1	0
(P9L)	P97	P96	P95	P94	P93	P92	P91	P90

P9n	输出数据控制 (输出模式下) (n = 0 至 15)
0	输出 0
1	输出 1

备注

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

(2) 端口 9 模式寄存器 (PM9)

复位后: FFFFH R/W 地址: PM9 FFFFF432H,
PM9L FFFFF432H, PM9H FFFFF433H

	15	14	13	12	11	10	9	8
PM9 (PM9H)	PM915	PM914	PM913	PM912	PM911	PM910	PM99	PM98

	7	6	5	4	3	2	1	0
(PM9L)	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90

PM9n	I/O 模式控制(n = 0 至 15)
0	输出模式
1	输入模式

备注

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

(3) 端口 9 模式控制寄存器 (PMC9)

(1/2)

复位后：0000H R/W 地址： PMC9 FFFFF452H, PMC9L FFFFF452H, PMC9H FFFFF453H

	15	14	13	12	11	10	9	8
PMC9 (PMC9H)	PMC915	PMC914	PMC913	PMC912	PMC911	PMC910	PMC99	PMC98

	7	6	5	4	3	2	1	0
(PMC9L)	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

PMC915	P915引脚操作模式说明
0	I/O 端口
1	A15 输出/INTP6 输入/TIP50 输入/TOP50 输出

PMC914	P914引脚操作模式说明
0	I/O 端口
1	A14 输出/INTP5 输入/TIP51 输入/TOP51 输出

PMC913	P913引脚操作模式说明
0	I/O 端口
1	A13 输出/INTP4 输入

PMC912	P912引脚操作模式说明
0	I/O 端口
1	A12 输出/ $\overline{\text{SCKB3}}$ I/O

PMC911	P911引脚操作模式说明
0	I/O 端口
1	A11 输出/SOB3 输出

PMC910	P910引脚操作模式说明
0	I/O 端口
1	A10 输出/SIB3 输入

PMC99	P99引脚操作模式说明
0	I/O 端口
1	A9 输出/ $\overline{\text{SCKB1}}$ I/O

PMC98	P98引脚操作模式说明
0	I/O 端口
1	A8 输出/SOB1 输出

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

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

注意事项 当 A0 至 A15 引脚用作 P90 至 P915 引脚的复用功能时，请一次将 PMC9 寄存器的所有 16 位设置为 FFFFH。

(4) 端口 9 功能控制寄存器 (PFC9)

注意事项 执行分离地址总线输出(A0 至 A15)时, 将 PFC9 或 PFCE9 寄存器清除为 0000H 之后, 一次将 PMC9 寄存器的 16 位设置为 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. 关于复用功能规范的详情, 参见 4.3.9 (6) 端口 9 复用功能规范。
 2. PFC9 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PFC9 的高 8 位和低 8 位分别用作 PFC9H 寄存器和 PFC9L 寄存器时, PFC9 可以按字节和按位读出或写入。
 3. 若要按字节或按位读取或写入 PFC9 的第 8 位至第 15 位, 将这些位作为 PFC9H 寄存器的第 0 位至第 7 位。

(5) 端口 9 功能控制扩展寄存器 (PFCE9)

注意事项 当执行分离地址总线输出(A0 至 A15)时, 在 PFC9 或 PFCE9 清除为 0000H 之后, 一次将 PMC9 寄存器的所有 16 位设置为 FFFFH。

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

(6) 端口 9 复用功能规范

PFC915	PFC915	P915 引脚复用功能说明
0	0	A15 输出
0	1	INTP6 输入
1	0	TIP50 输入
1	1	TOP50 输出

PFC914	PFC914	P914 引脚复用功能说明
0	0	A14 输出
0	1	INTP5 输入
1	0	TIP51 输入
1	1	TOP51 输出

PFC913	P913 引脚复用功能说明
0	A13 输出
1	INTP4 输入

PFC912	P912 引脚复用功能说明
0	A12 输出
1	SCKB3 I/O

PFC911	P911 引脚复用功能说明
0	A11 输出
1	SOB3 输出

PFC910	P910 引脚复用功能说明
0	A10 输出
1	SIB3 输入

PFC99	P99 引脚复用功能说明
0	A9 输出
1	SCKB1 I/O

PFC98	P98 引脚复用功能说明
0	A8 输出
1	SOB1 输出

PFC97	PFC97	P97 引脚复用功能说明
0	0	A7 输出
0	1	SIB1 输入
1	0	TIP20 输入
1	1	TOP20 输出

PFCE96	PFC96	P96 引脚复用功能说明
0	0	A6 输出
0	1	禁止设置
1	0	TIP21 输入
1	1	TOP21 输出

PFCE95	PFC95	P95 引脚复用功能说明
0	0	A5 输出
0	1	TIP30 输入
1	0	TOP30 输出
1	1	禁止设置

PFCE94	PFC94	P94 引脚复用功能说明
0	0	A4 输出
0	1	TIP31 输入
1	0	TOP31 输出
1	1	禁止设置

PFCE93	PFC93	P93 引脚复用功能说明
0	0	A3 输出
0	1	TIP40 输入
1	0	TOP40 输出
1	1	禁止设置

PFCE92	PFC92	P92 引脚复用功能说明
0	0	A2 输出
0	1	TIP41 输入
1	0	TOP41 输出
1	1	禁止设置

PFCE91	PFC91	P91 引脚复用功能说明
0	0	A1 输出
0	1	KR7 输入
1	0	RXDA1 输入/KR7 输入 [※]
1	1	SCL02 I/O

PFCE90	PFC90	P90 引脚复用功能说明
0	0	A0 输出
0	1	KR6 输入
1	0	TXDA1 输出
1	1	SDA02 I/O

注 RXDA1 和 KR7 功能不能同时使用。使用 RXDA1 引脚时，不要使用 KR7 引脚。而使用 KR7 引脚时，不要使用 RXDA1 引脚（建议将 PFC91 位设置为 1，并且将 PFCE91 位设置为 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

	15	14	13	12	11	10	9	8
PF9 (PF9H)	PF915	PF914	PF913	PF912	PF911	PF910	PF99	PF98

	7	6	5	4	3	2	1	0
(PF9L)	PF97	PF96	PF95	PF94	PF93	PF92	PF91	PF90

PF9n	正常输出 或 N沟道漏极开路输出控制 (n = 0 至 15)
0	正常输出 (CMOS 输出)
1	N沟道漏极开路输出

注意事项 若一个输出引脚被上拉至 EV_{DD} 或更高电压时，确保将 PF9n 位设置为 1。

- 备注**
1. PF9 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PF9 的高 8 位和低 8 位分别用作 PF9H 寄存器和 PF9L 寄存器时，PF9 可以按字节和按位读出或写入。
 2. 若要按字节或按位读取或写入 PF9 的第 8 位至第 15 位，将这些位作为 PF9H 寄存器的第 0 位至第 7 位。

4.3.10 端口 CD

端口 CD 有 4 位输入输出端口，可以按位控制输入输出设置。

端口 CD 包括以下复用功能引脚：

表 4-13. 端口 CD 复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	输入/输出	备注	框图类型
PCD0	77	—	—	—	B-1
PCD1	78	—	—		B-1
PCD2	79	—	—		B-1
PCD3	80	—	—		B-1

(1) 端口 CD 寄存器 (PCD)

复位后: 00H (输出锁存) R/W 地址: FFFFF00EH

	7	6	5	4	3	2	1	0
PCD	0	0	0	0	PCD3	PCD2	PCD1	PCD0

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

(2) 端口 CD 模式寄存器 (PMCD)

复位后: FFH R/W 地址: FFFFF02EH

	7	6	5	4	3	2	1	0
PMCD	1	1	1	1	PMCD3	PMCD2	PMCD1	PMCD0

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

4.3.11 端口 CM

端口 CM 有 6 位输入输出端口，可以按位控制输入输出设置。

端口 CM 包括以下复用功能引脚：

表 4-14. 端口 CM 复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	输入/输出	备注	框图类型
PCM0	85	WAIT	输入	-	D-1
PCM1	86	CLKOUT	输出		D-2
PCM2	87	HLD $\overline{\text{AK}}$	输出		D-2
PCM3	88	HLD $\overline{\text{RQ}}$	输入		D-1
PCM4	89	-	-		B-1
PCM5	90	-	-		B-1

(1) 端口 CM 寄存器 (PCM)

复位后: 00H (输出锁存) R/W 地址: FFFFF00CH

	7	6	5	4	3	2	1	0
PCM	0	0	PCM5	PCM4	PCM3	PCM2	PCM1	PCM0

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

(2) 端口 CM 寄存器 (PMCM)

复位后: FFH R/W 地址: FFFFF02CH

	7	6	5	4	3	2	1	0
PMCM	1	1	PMCM5	PMCM4	PMCM3	PMCM2	PMCM1	PMCM0

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

(3) 端口 CM 模式控制寄存器 (PMCCM)

复位后: 00H R/W 地址: FFFF04CH

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

PMCCM3	PCM3 引脚操作模式说明
0	I/O 端口
1	$\overline{\text{HLDRQ}}$ 输入

PMCCM2	PCM2 引脚操作模式说明
0	I/O 端口
1	$\overline{\text{HLDAK}}$ 输出

PMCCM1	PCM1 引脚操作模式说明
0	I/O 端口
1	CLKOUT 输出

PMCCM0	PCM0 引脚操作模式说明
0	I/O 端口
1	$\overline{\text{WAIT}}$ 输入

4.3.12 端口 CS

端口 CS 有 8 位输入输出端口，可以按位控制输入输出设置。

端口 CS 包括以下复用功能引脚：

表 4-15. 端口 CS 复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	输入/输出	备注	框图类型
PCS0	81	$\overline{CS0}$	输出	—	D-2
PCS1	82	$\overline{CS1}$	输出		D-2
PCS2	83	$\overline{CS2}$	输出		D-2
PCS3	84	$\overline{CS3}$	输出		D-2
PCS4	91	—	—		B-1
PCS5	92	—	—		B-1
PCS6	93	—	—		B-1
PCS7	94	—	—		B-1

(1) 端口 CS 寄存器 (PCS)

复位后: 00H (输出锁存) R/W 地址: FFFFF008H

	7	6	5	4	3	2	1	0
PCS	PCS7	PCS6	PCS5	PCS4	PCS3	PCS2	PCS1	PCS0

PCS _n	输出数据控制 (输出模式下) (n = 0 至 7)
0	输出 0
1	输出 1

(2) 端口 CS 模式寄存器 (PMCS)

复位后: FFH R/W 地址: FFFFF028H

	7	6	5	4	3	2	1	0
PMCS	PMCS7	PMCS6	PMCS5	PMCS4	PMCS3	PMCS2	PMCS1	PMCS0

PMCS _n	I/O 模式控制 (n = 0 至 7)
0	输出模式
1	输入模式

(3) 端口 CS 模式控制寄存器 (PMCCS)

复位后: 00H R/W 地址: FFFFF048H

	7	6	5	4	3	2	1	0
PMCCS	0	0	0	0	PMCCS3	PMCCS2	PMCCS1	PMCCS0
PMCCS3	PCS3 引脚操作模式说明							
0	I/O 端口							
1	$\overline{\text{CS3}}$ 输出							
PMCCS2	PCS2 引脚操作模式说明							
0	I/O 端口							
1	$\overline{\text{CS2}}$ 输出							
PMCCS1	PCS1 引脚操作模式说明							
0	I/O 端口							
1	$\overline{\text{CS1}}$ 输出							
PMCCS0	PCS0 引脚操作模式说明							
0	I/O 端口							
1	$\overline{\text{CS0}}$ 输出							

4.3.13 端口 CT

端口 CT 有 8 位输入输出端口，可以按位控制输入输出设置。

端口 CT 包括以下复用功能引脚：

表 4-16. 端口 CT 复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	输入/输出	备注	框图类型
PCT0	95	$\overline{WR0}$	输出	—	D-2
PCT1	96	$\overline{WR1}$	输出		D-2
PCT2	97	—	—		B-1
PCT3	98	—	—		B-1
PCT4	99	$\overline{RD}$	输出		D-2
PCT5	100	—	—		B-1
PCT6	101	ASTB	输出		D-2
PCT7	102	—	—		B-1

(1) 端口 CT 寄存器 (PCT)

复位后: 00H (输出锁存) R/W 地址: FFFFF00AH

	7	6	5	4	3	2	1	0
PCT	PCT7	PCT6	PCT5	PCT4	PCT3	PCT2	PCT1	PCT0

PCTn	输出数据控制 (输出模式下) (n = 0 至 7)
0	输出 0
1	输出 1

(2) 端口 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	PMCT7	PMCT6	PMCT5	PMCT4	PMCT3	PMCT2	PMCT1	PMCT0

PMCTn	I/O 模式控制 (n = 0 至 7)
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

PMCCT

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

PMCCT6	PCT6 引脚操作模式说明
0	I/O 端口
1	ASTB 输出

PMCCT4	PCT4 引脚操作模式说明
0	I/O 端口
1	$\overline{\text{RD}}$ 输出

PMCCT1	PCT1 引脚操作模式说明
0	I/O 端口
1	$\overline{\text{WR1}}$ 输出

PMCCT0	PCT0 引脚操作模式说明
0	I/O 端口
1	$\overline{\text{WR0}}$ 输出

4.3.14 端口 DH

端口 DH 有 8 位输入输出端口，可以按位控制输入输出设置。

端口 DH 包括以下复用功能引脚：

表 4-17. 端口 DH 复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	输入/输出	备注	框图类型
PDH0	121	A16	输出	—	D-2
PDH1	122	A17	输出		D-2
PDH2	123	A18	输出		D-2
PDH3	124	A19	输出		D-2
PDH4	125	A20	输出		D-2
PDH5	126	A21	输出		D-2
PDH6	127	A22	输出		D-2
PDH7	128	A23	输出		D-2

(1) 端口 DH 寄存器 (PDH)

复位后: 00H (输出锁存) R/W 地址: FFFFF006H

	7	6	5	4	3	2	1	0
PDH	PDH7	PDH6	PDH5	PDH4	PDH3	PDH2	PDH1	PDH0

PDHn	输出数据控制 (输出模式下) (n = 0 至 7)
0	输出 0
1	输出 1

(2) 端口 DH 模式寄存器 (PMDH)

复位后: FFH R/W 地址: FFFFF026H

	7	6	5	4	3	2	1	0
PMDH	PMDH7	PMDH6	PMDH5	PMDH4	PMDH3	PMDH2	PMDH1	PMDH0

PMDHn	I/O 模式控制 (n = 0 至 7)
0	输出模式
1	输入模式

(3) 端口 DH 模式控制寄存器 (PMCDH)

复位后: 00H R/W 地址: FFFF046H

	7	6	5	4	3	2	1	0
PMCDH	PMCDH7	PMCDH6	PMCDH5	PMCDH4	PMCDH3	PMCDH2	PMCDH1	PMCDH0

PMCDHn	PDHn 引脚操作模式说明 (n = 0 至 7)
0	I/O 端口
1	Am 输出 (地址总线输出) (m = 16 至 23)

4.3.15 端口 DL

端口 DL 有 16 位输入输出端口，可以按位控制输入输出设置。

端口 DL 包括以下复用功能引脚：

表 4-18. 端口 DL 复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	输入/输出	备注	框图类型
PDL0	105	AD0	I/O	—	D-3
PDL1	106	AD1	I/O		D-3
PDL2	107	AD2	I/O		D-3
PDL3	108	AD3	I/O		D-3
PDL4	109	AD4	I/O		D-3
PDL5	110	AD5/FLMD1 ^注	I/O		D-3
PDL6	111	AD6	I/O		D-3
PDL7	112	AD7	I/O		D-3
PDL8	113	AD8	I/O		D-3
PDL9	114	AD9	I/O		D-3
PDL10	115	AD10	I/O		D-3
PDL11	116	AD11	I/O		D-3
PDL12	117	AD12	I/O		D-3
PDL13	118	AD13	I/O		D-3
PDL14	119	AD14	I/O		D-3
PDL15	120	AD15	I/O		D-3

注 该引脚设置为 flash 存储器编程模式后，不需要通过端口控制寄存器进行操作。详情参见第二十七章 FLASH 存储器。

(1) 端口 DL 寄存器 (PDL)

复位后: 0000H (输出锁存) R/W 地址: PDL FFFFF004H,
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
(PDLL)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

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

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

(2) 端口 DL 模式寄存器 (PMDL)

复位后: FFFFH R/W 地址: PMDL FFFFF024H,
PMDL 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 至 15)						
0	输出模式						
1	输入模式						

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

(3) 端口 DL 模式控制寄存器 (PMCDL)

复位后: 0000H R/W 地址: PMCDL 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	PDLn 引脚操作模式说明 (n = 0 至 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 (分离模式)，并且 BSC 寄存器的 BS30 至 BS00 位为 = 0 (8-位总线宽度)时，不要指定 AD8 至 AD15 引脚。

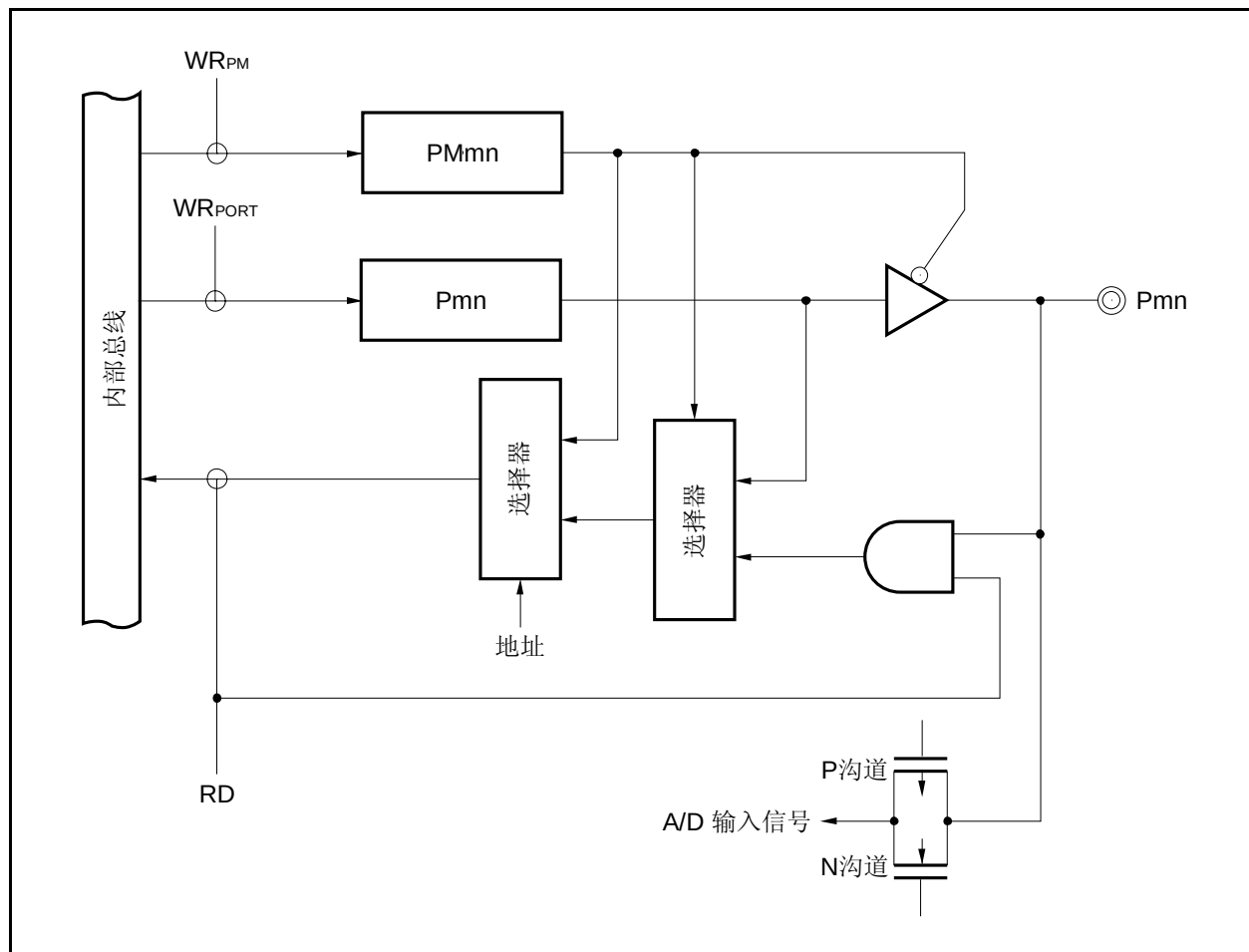
备注

1. PMCDL 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PMCDL 的高 8 位和低 8 位分别用作 PMCDLH 寄存器和 PMCDLL 寄存器时，PMCDL 可以按字节或按位读出或写入。

2. 若要按字节或按位读取或写入 PMCDL 的第 8 位至第 15 位，将这些位作为 PMCDLH 寄存器的第 0 位至第 7 位。

4.4 框图

图 4-3. A-1 型的功能框图



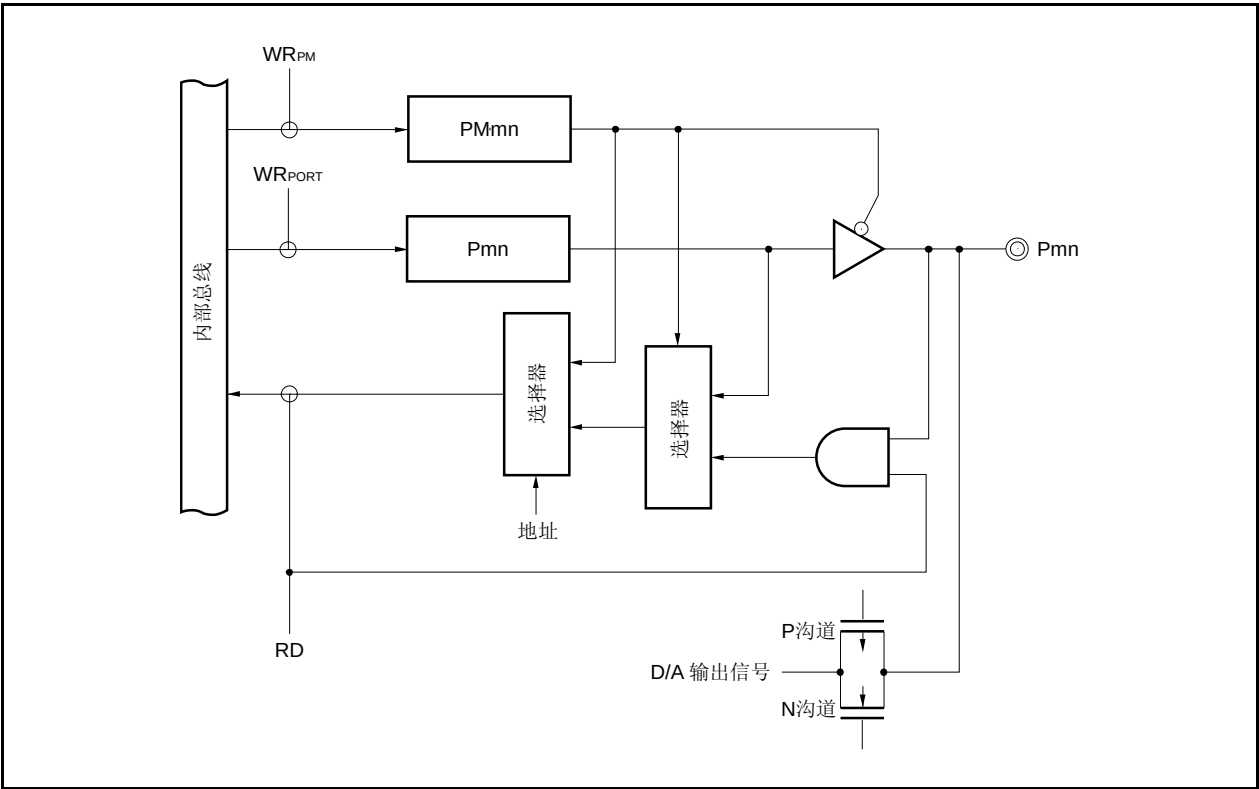
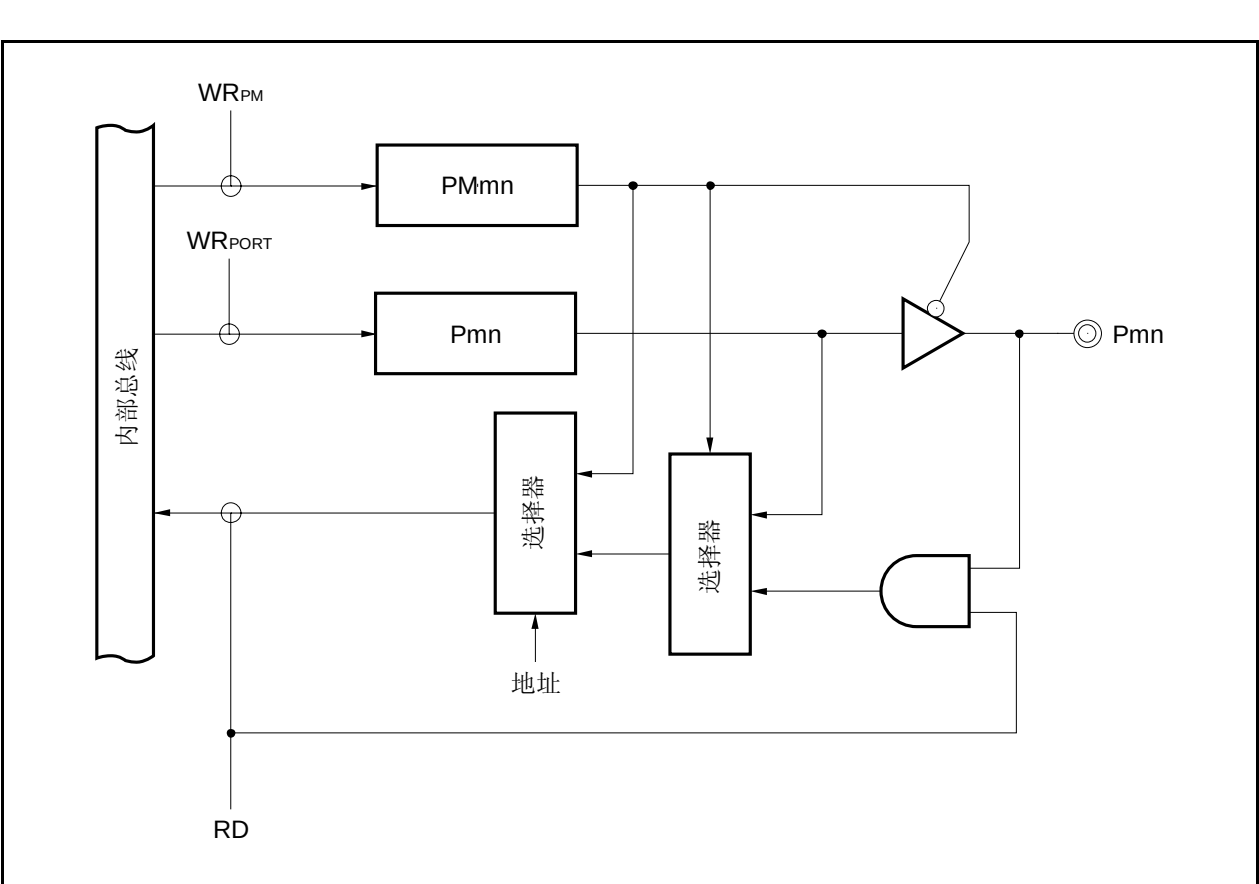


图 4-6. C-1 型的功能框图

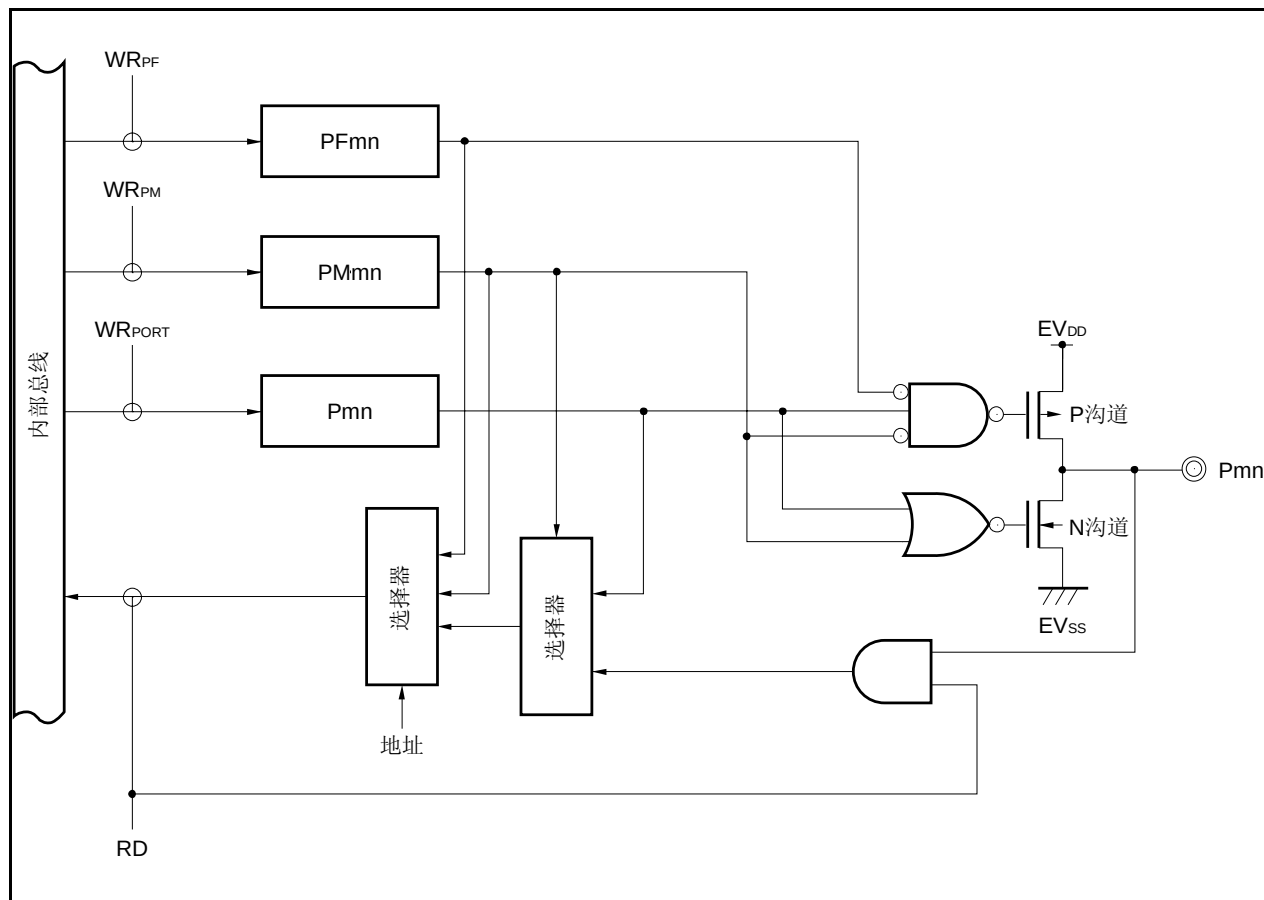


图 4-7. D-1 型的功能框图

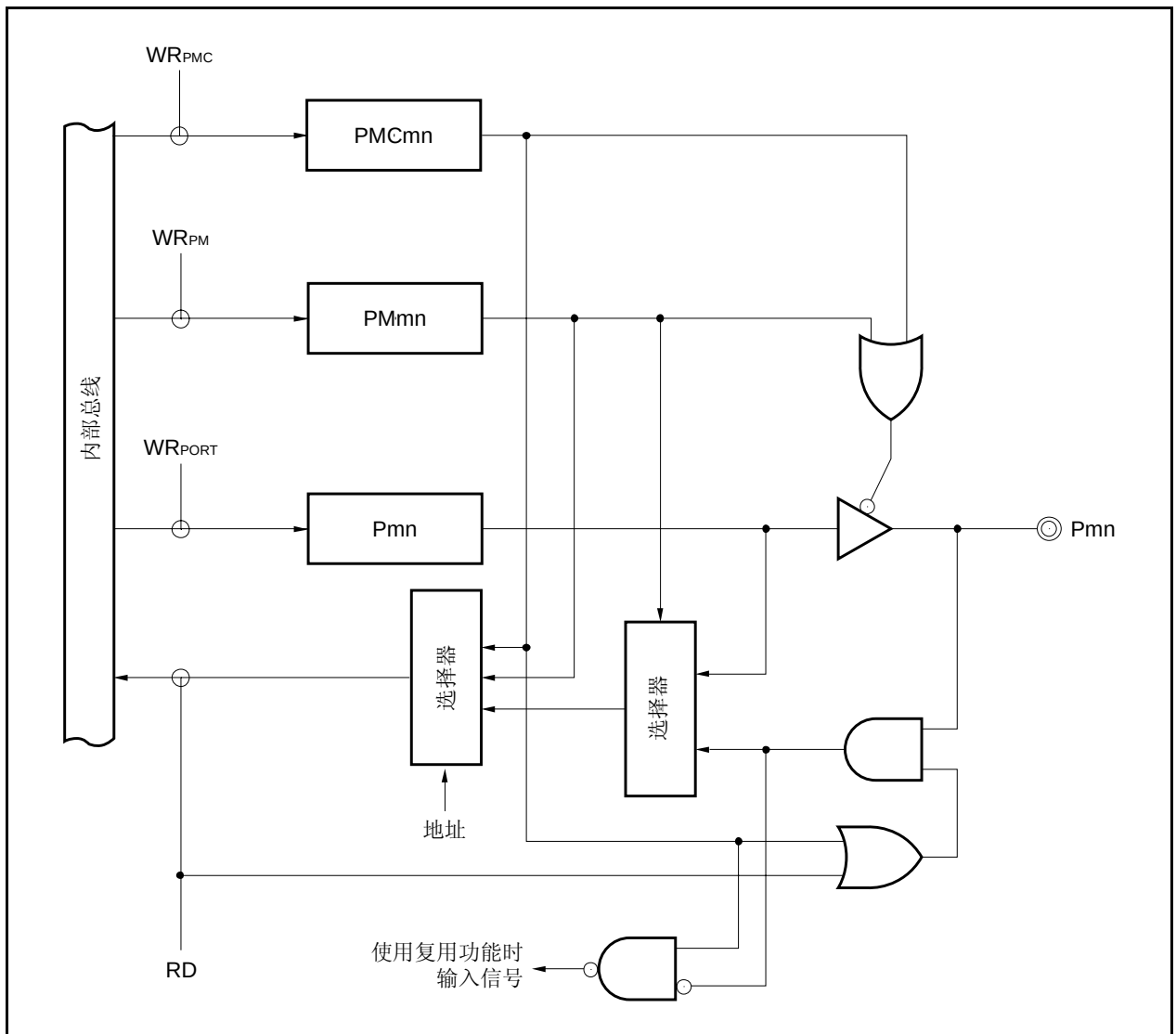


图 4-8. D-2 型的功能框图

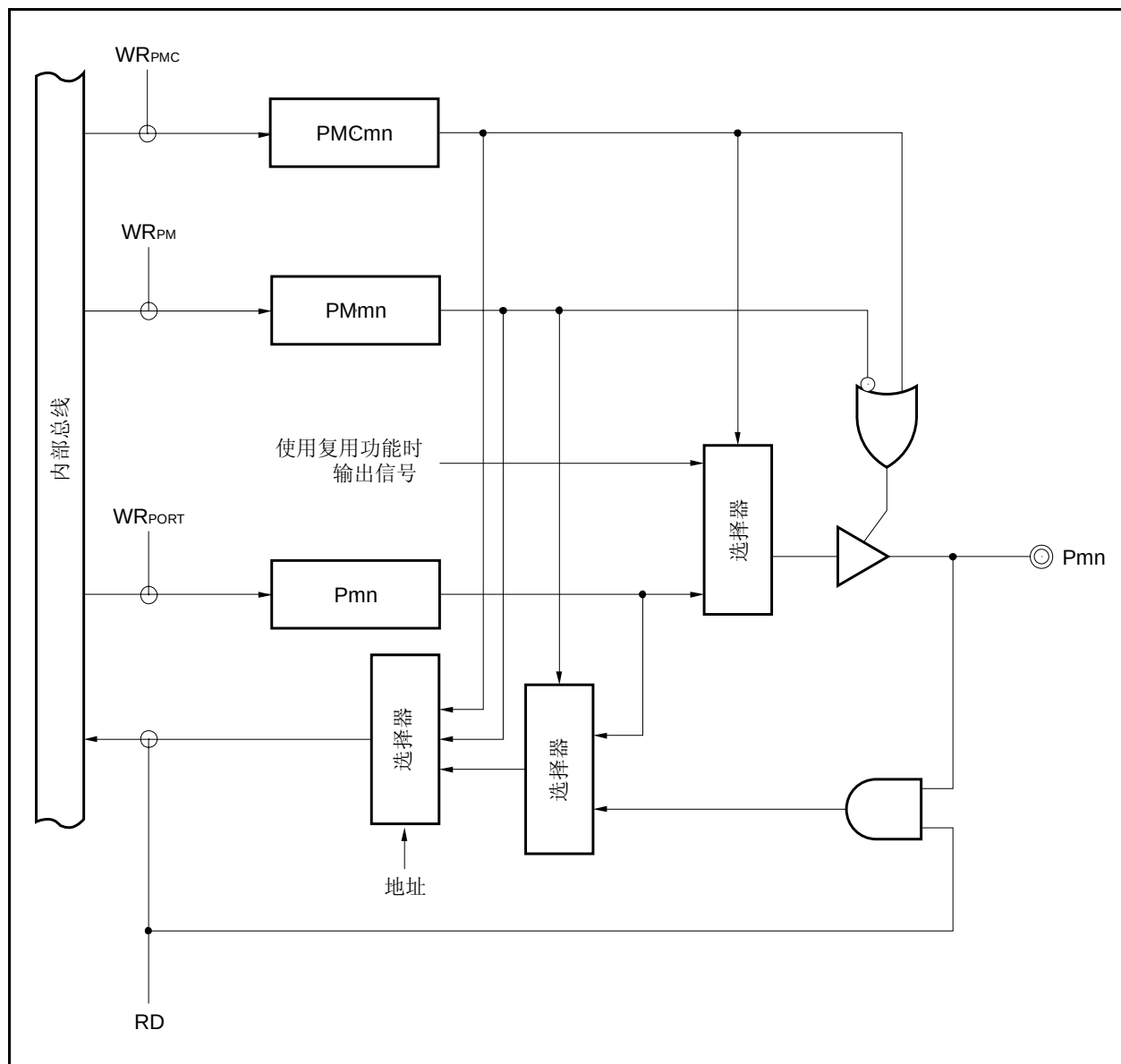


图 4-9. D-3 型的功能框图

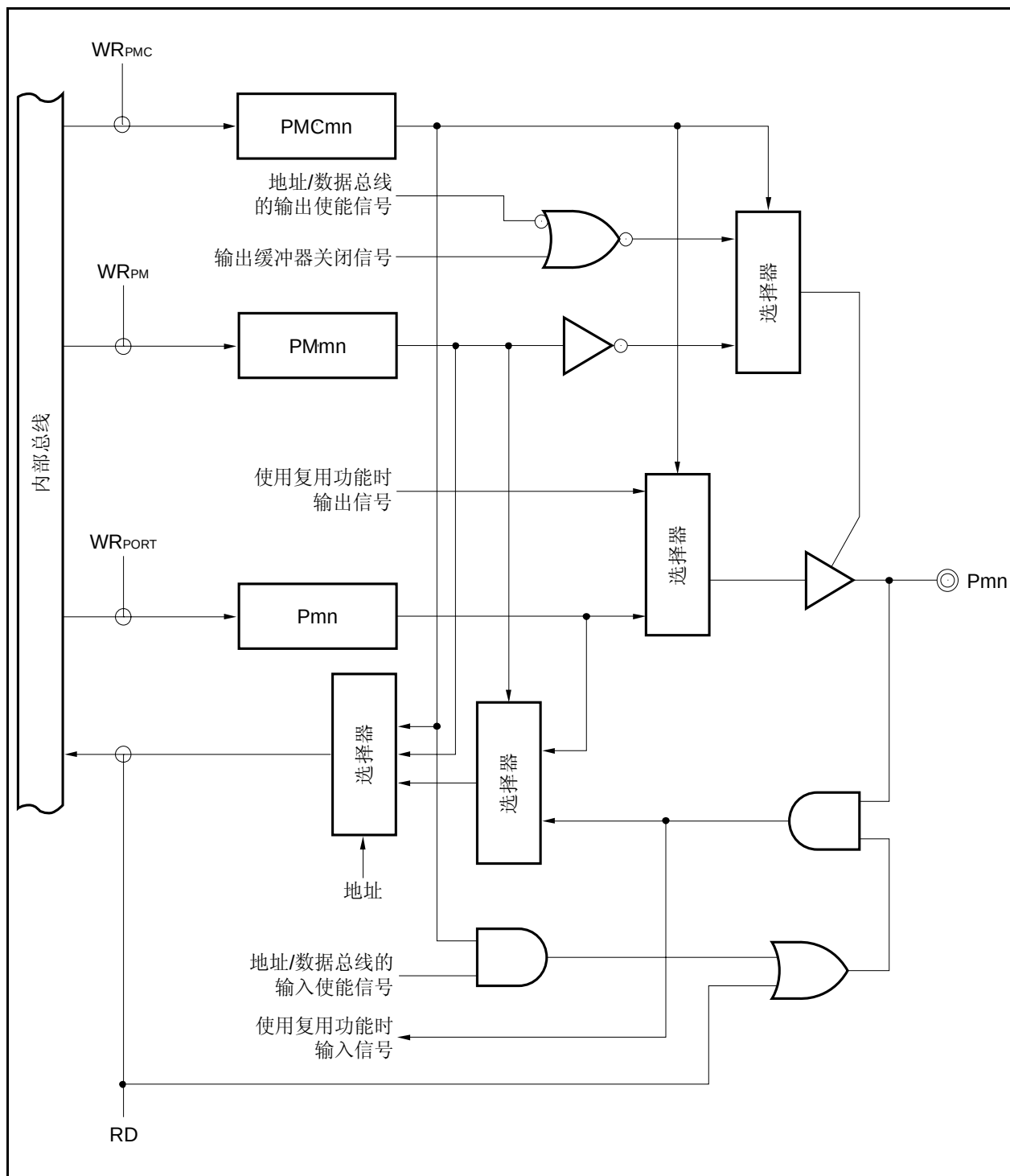


图 4-10. E-1 型的功能框图

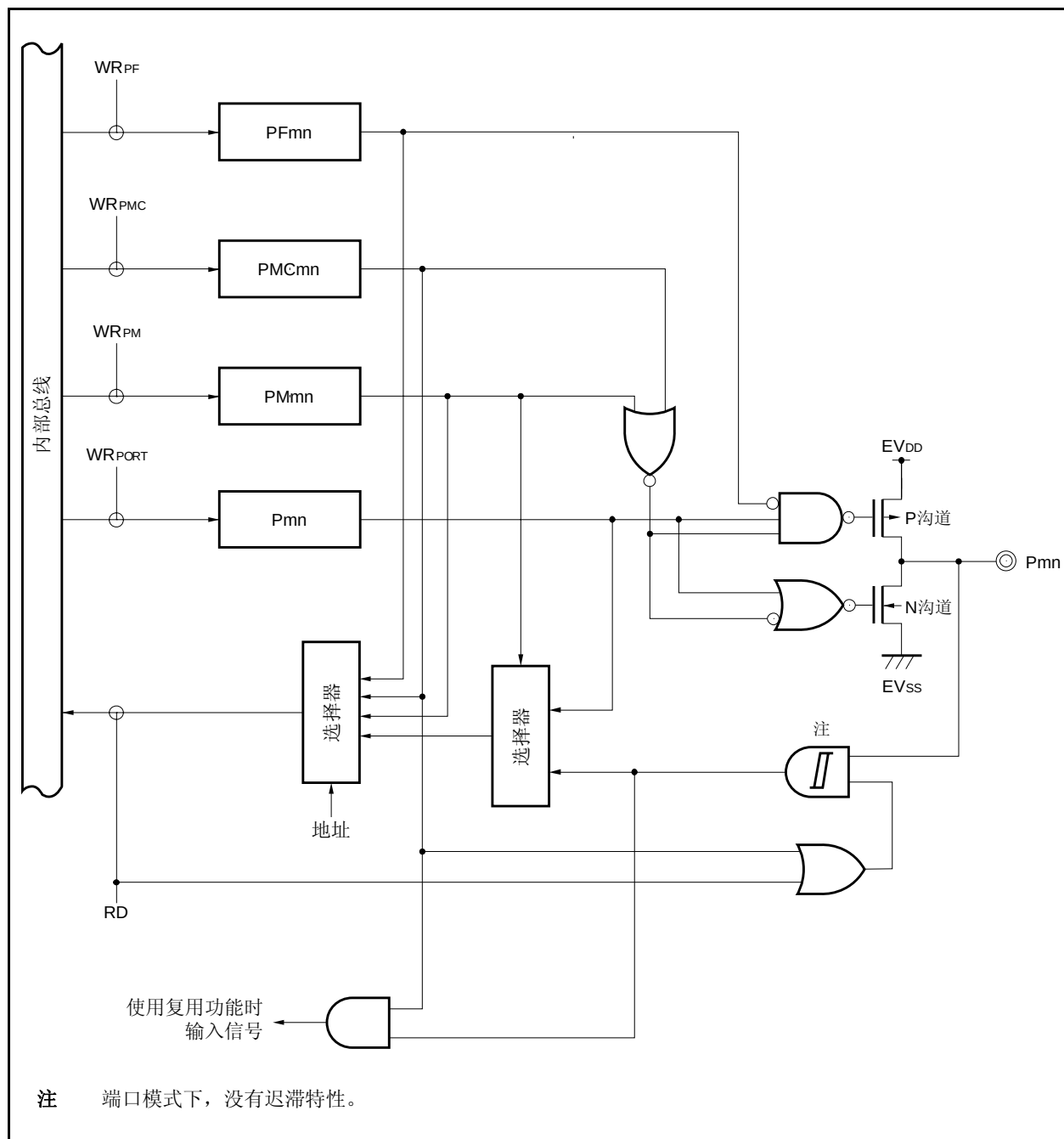


图 4-11. E-2 型的功能框图

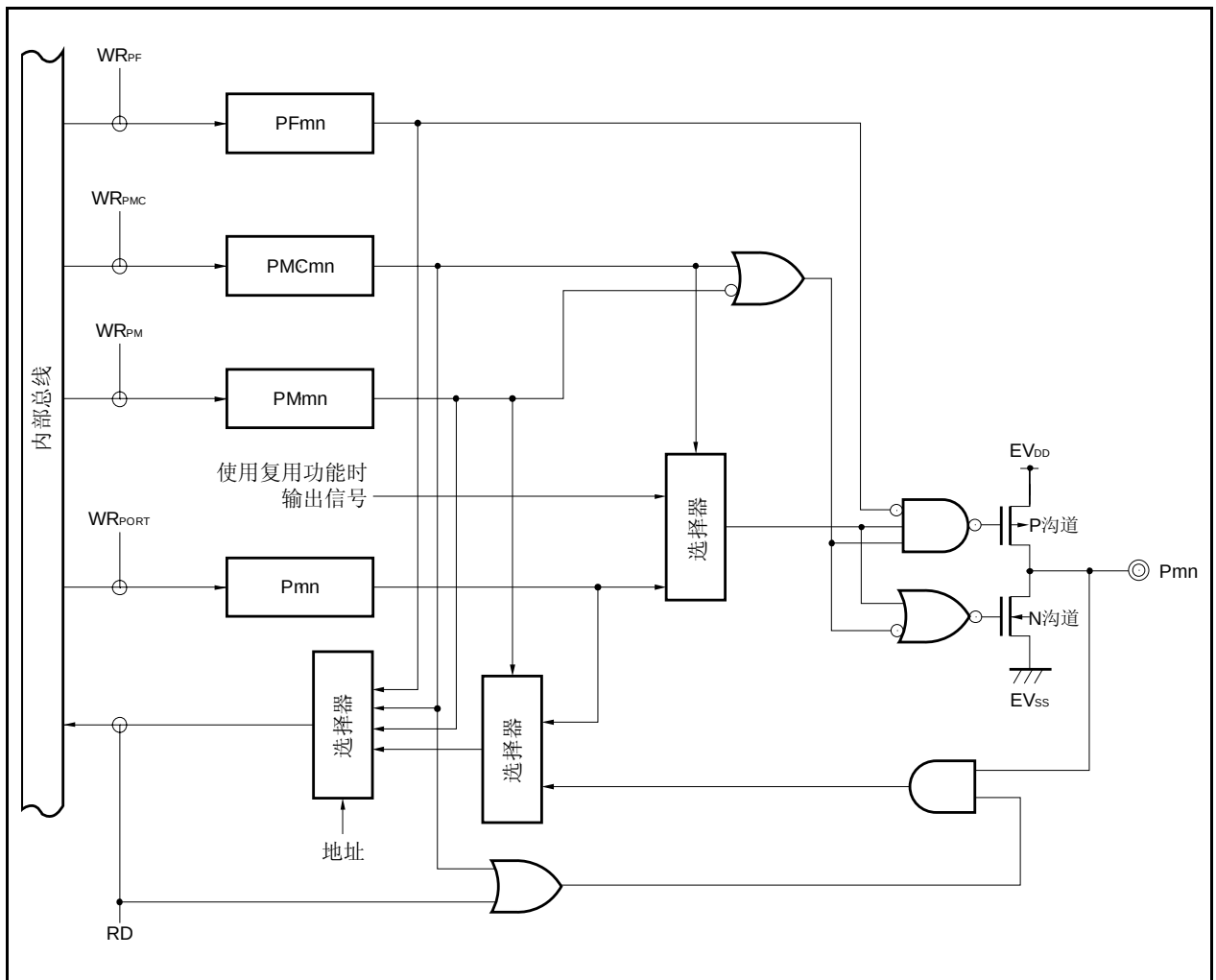


图 4-12. E-3 型的功能框图

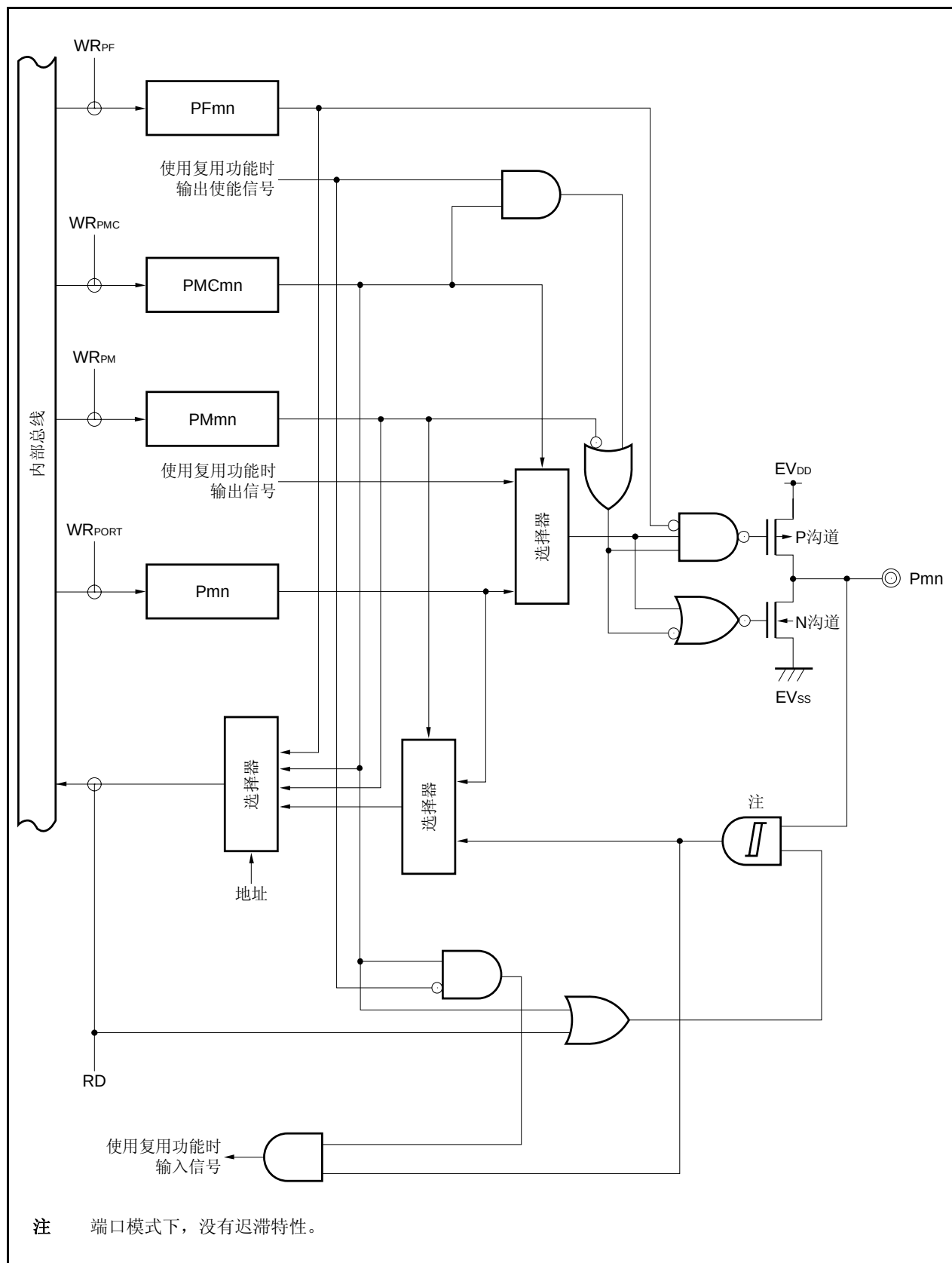


图 4-13. G-1 型的功能框图

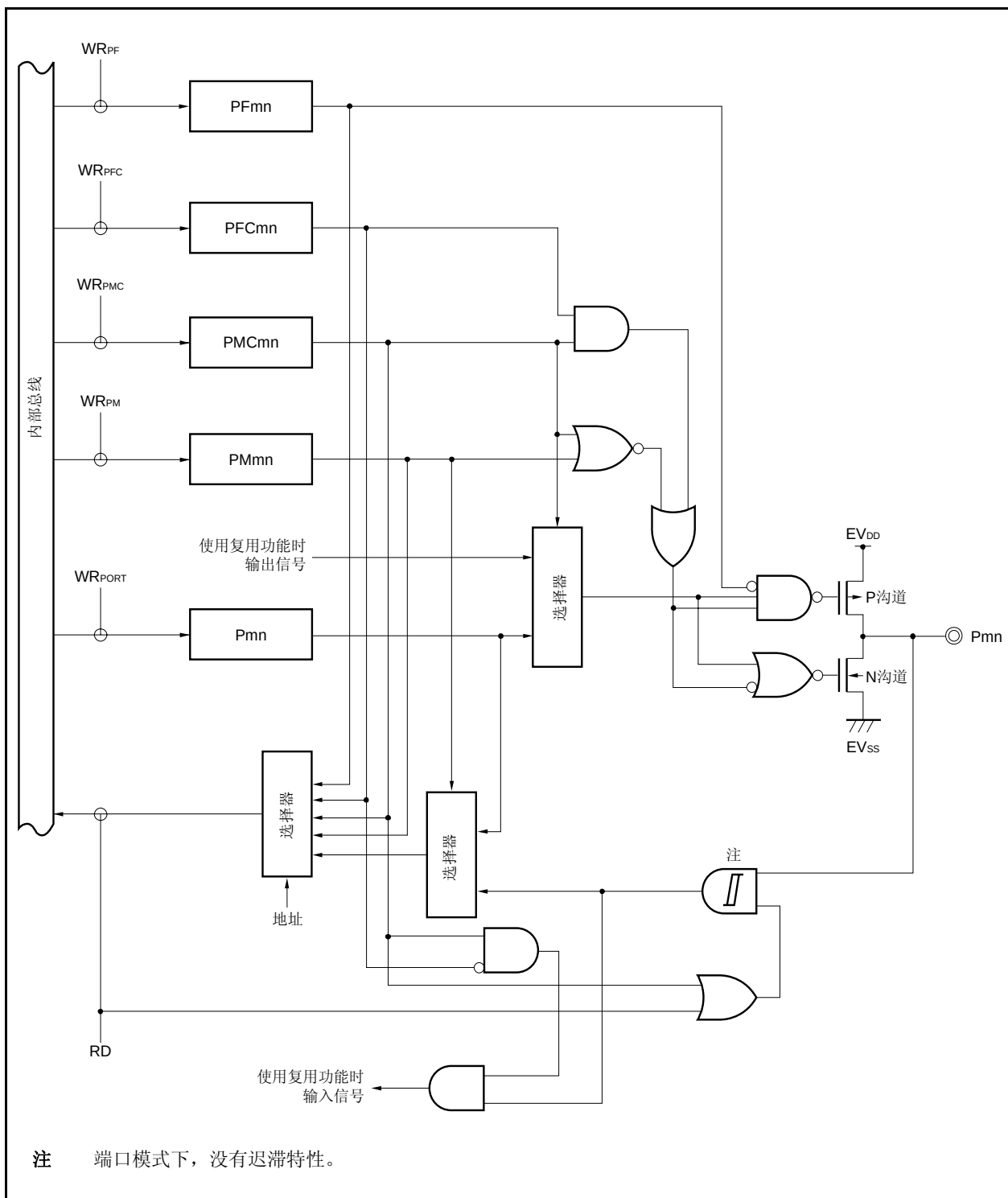


图 4-14. G-2 型的功能框图

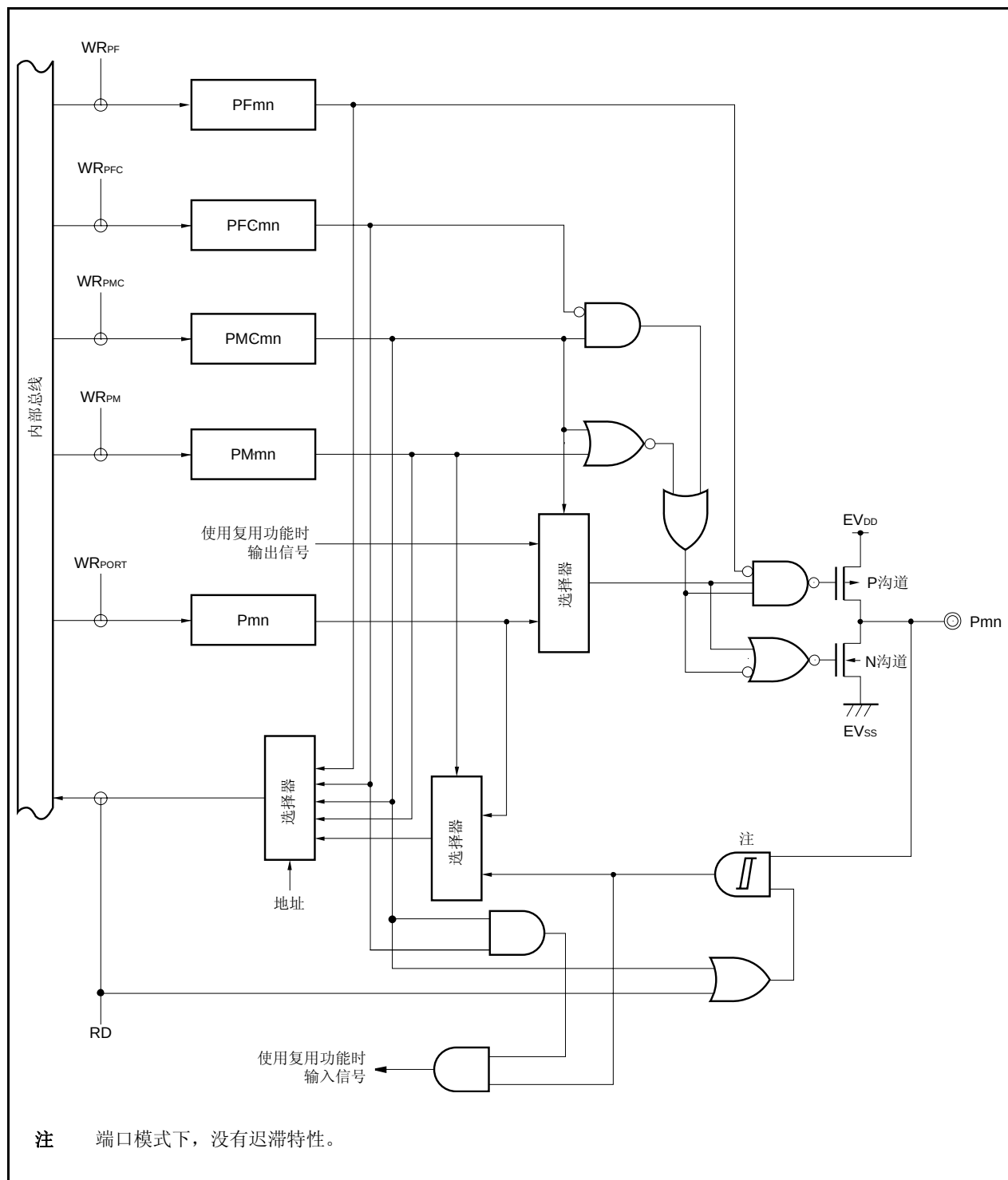


图 4-15. G-3 型的功能框图

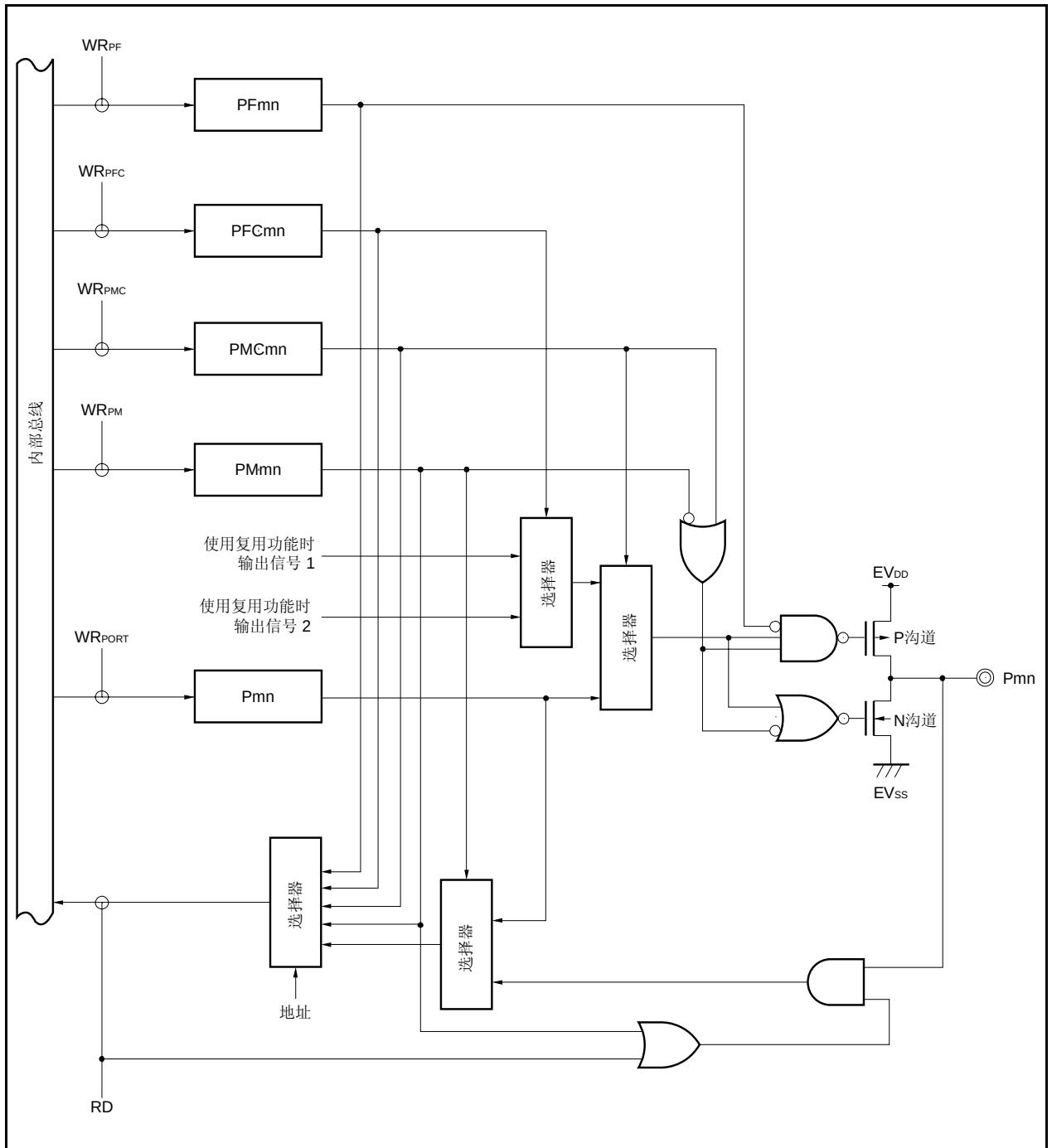


图 4-16. G-5 型的功能框图

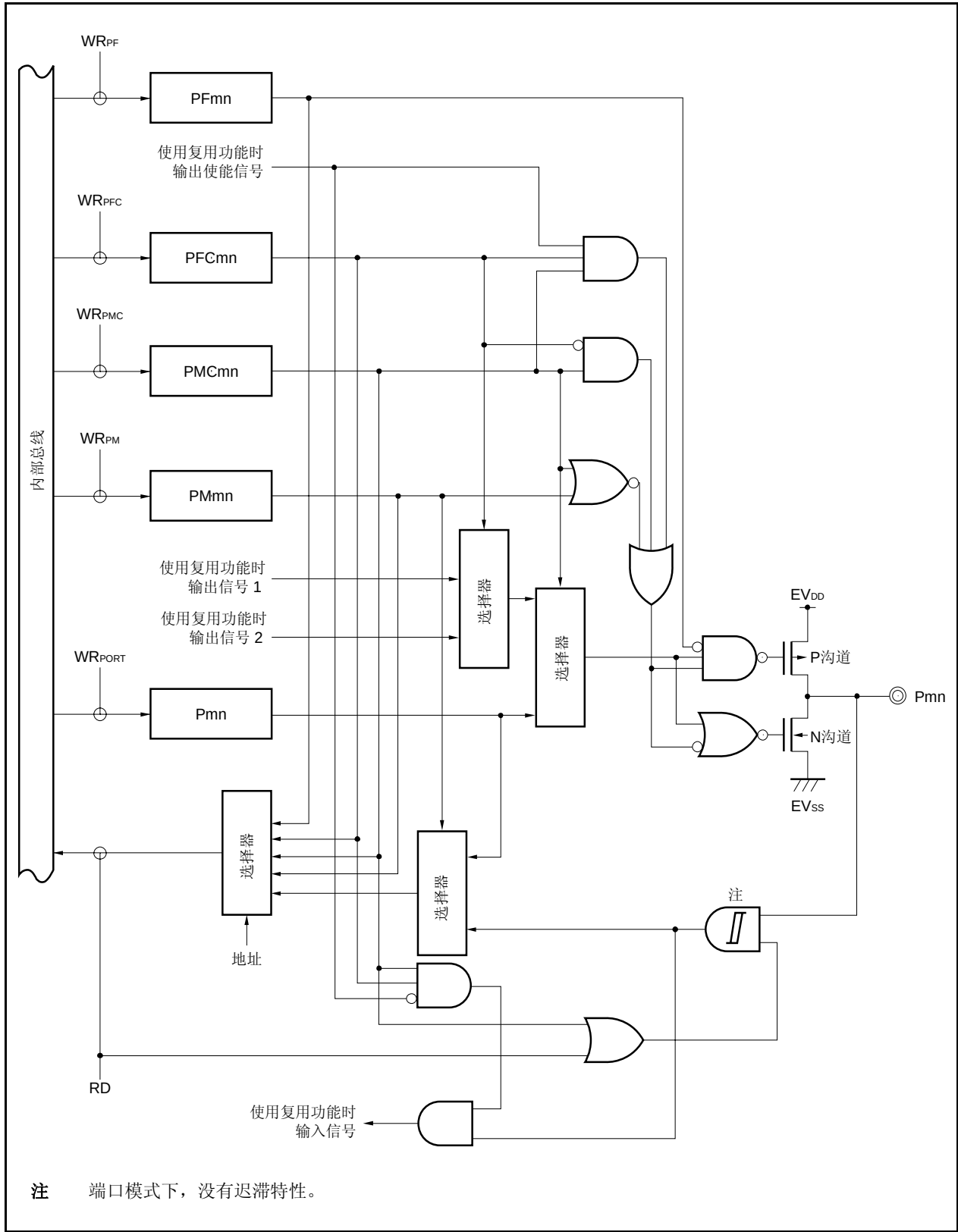


图 4-17. G-6 型的功能框图

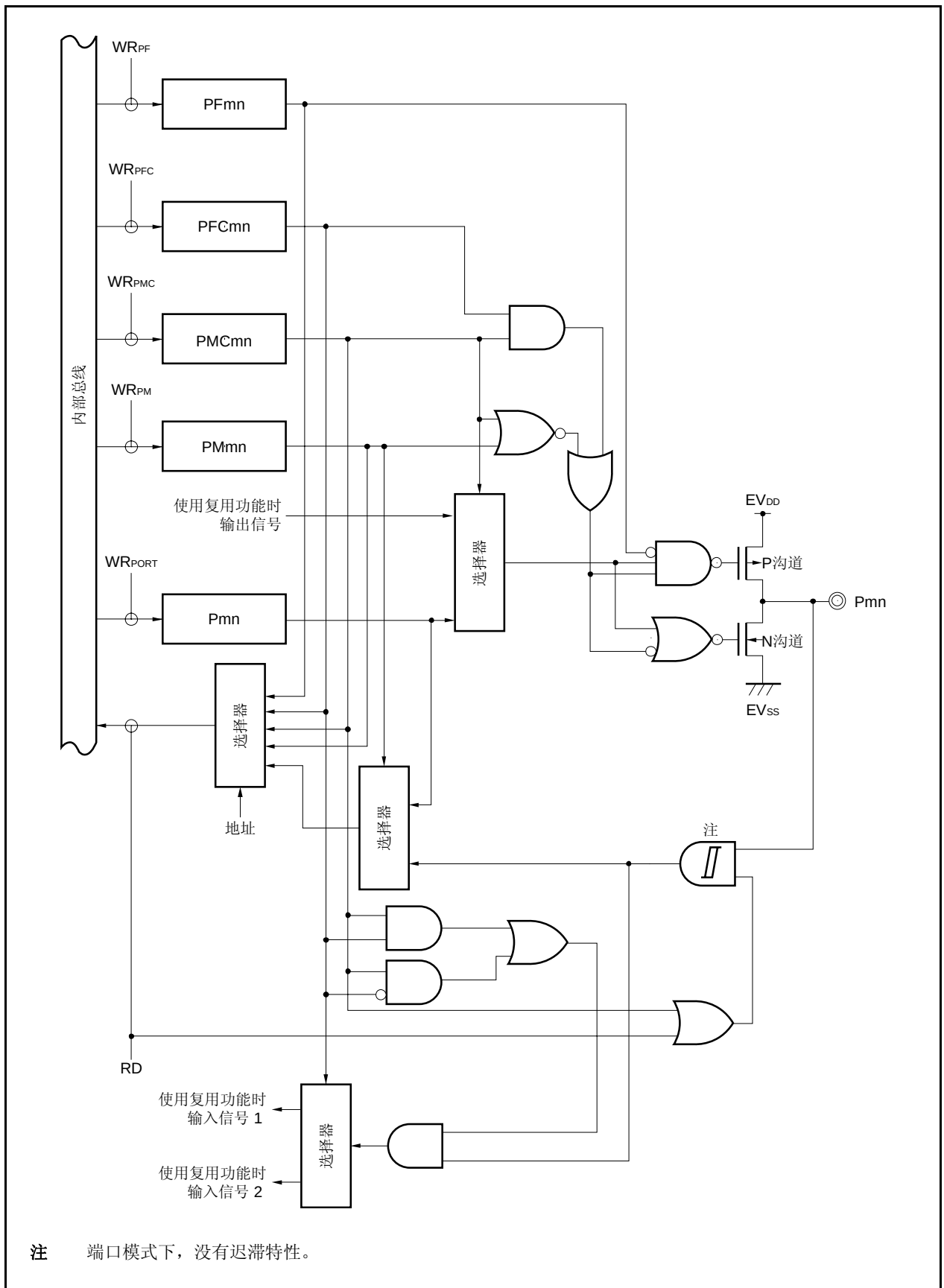


图 4-18. G-12 型的功能框图

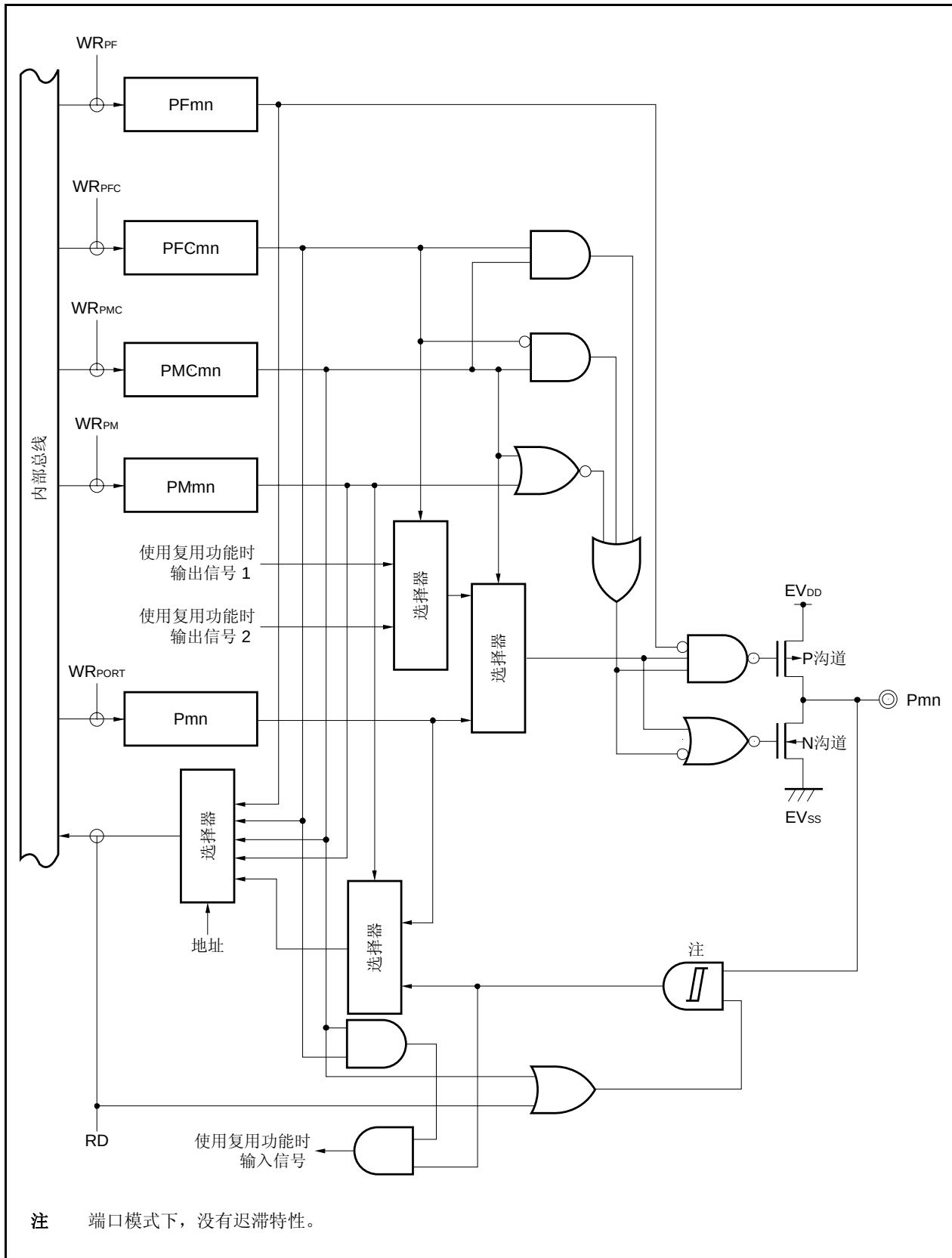


图 4-19. L-1 型的功能框图

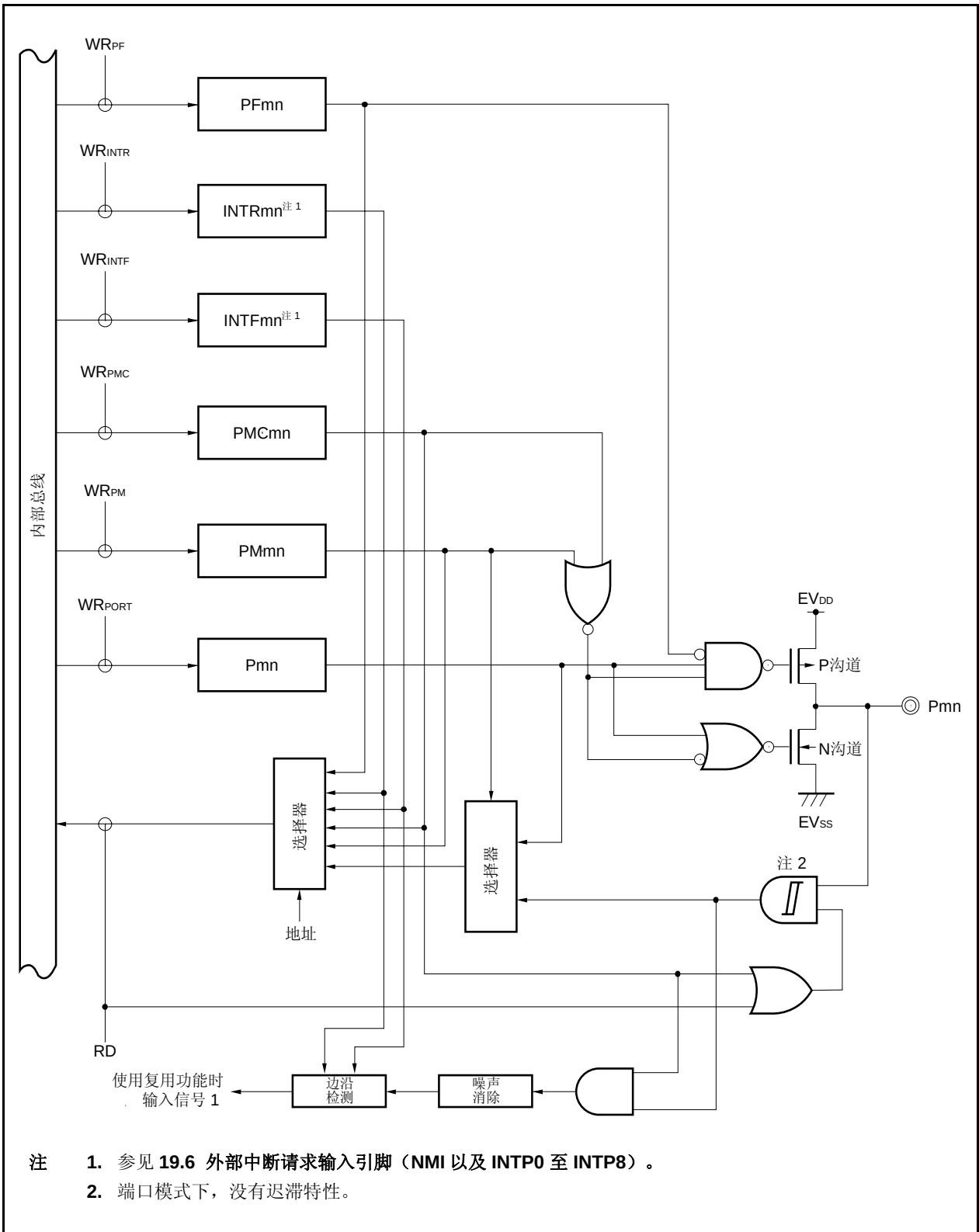


图 4-21. N-1 型的功能框图

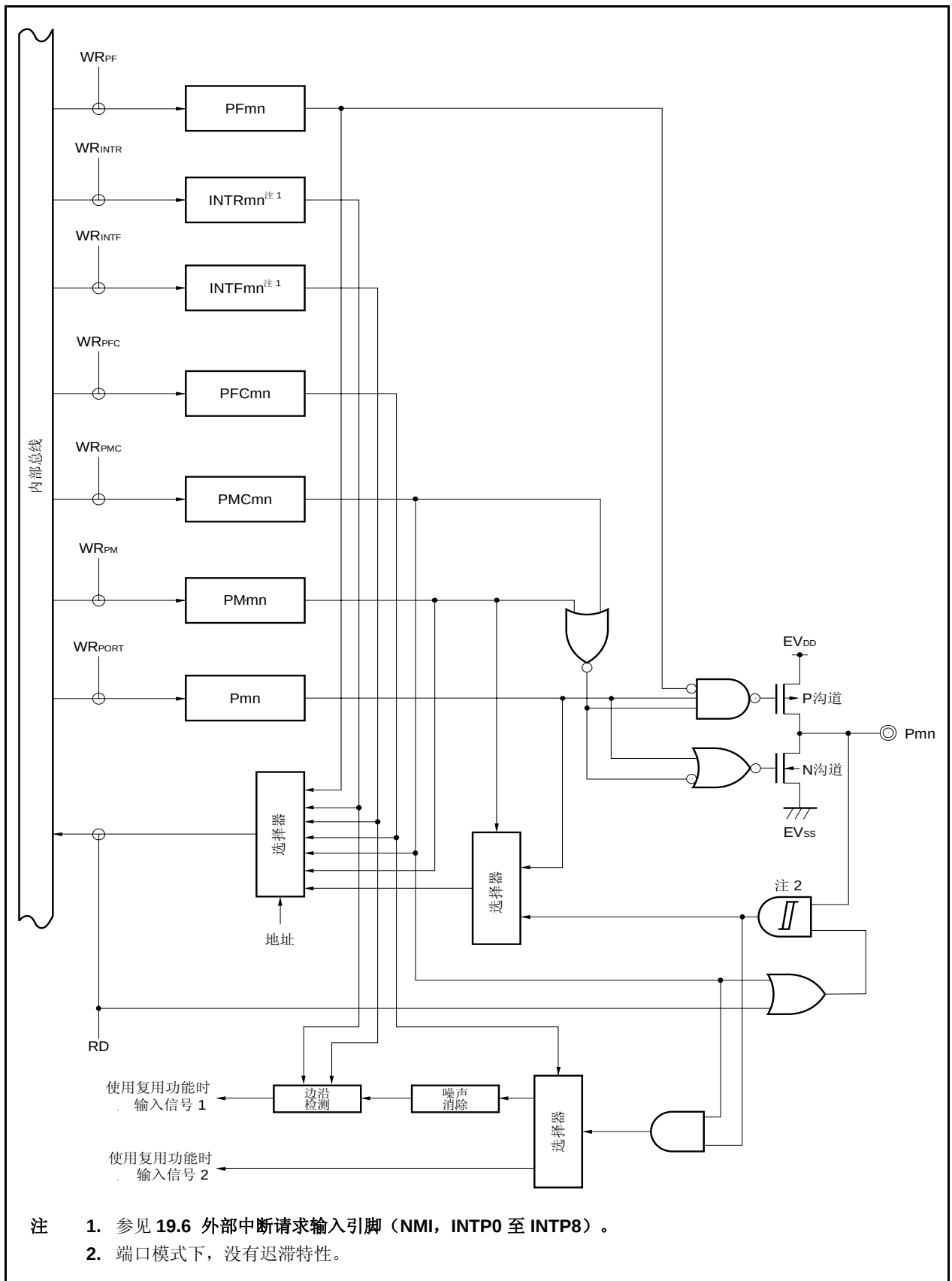


图 4-22. N-2 型的功能框图

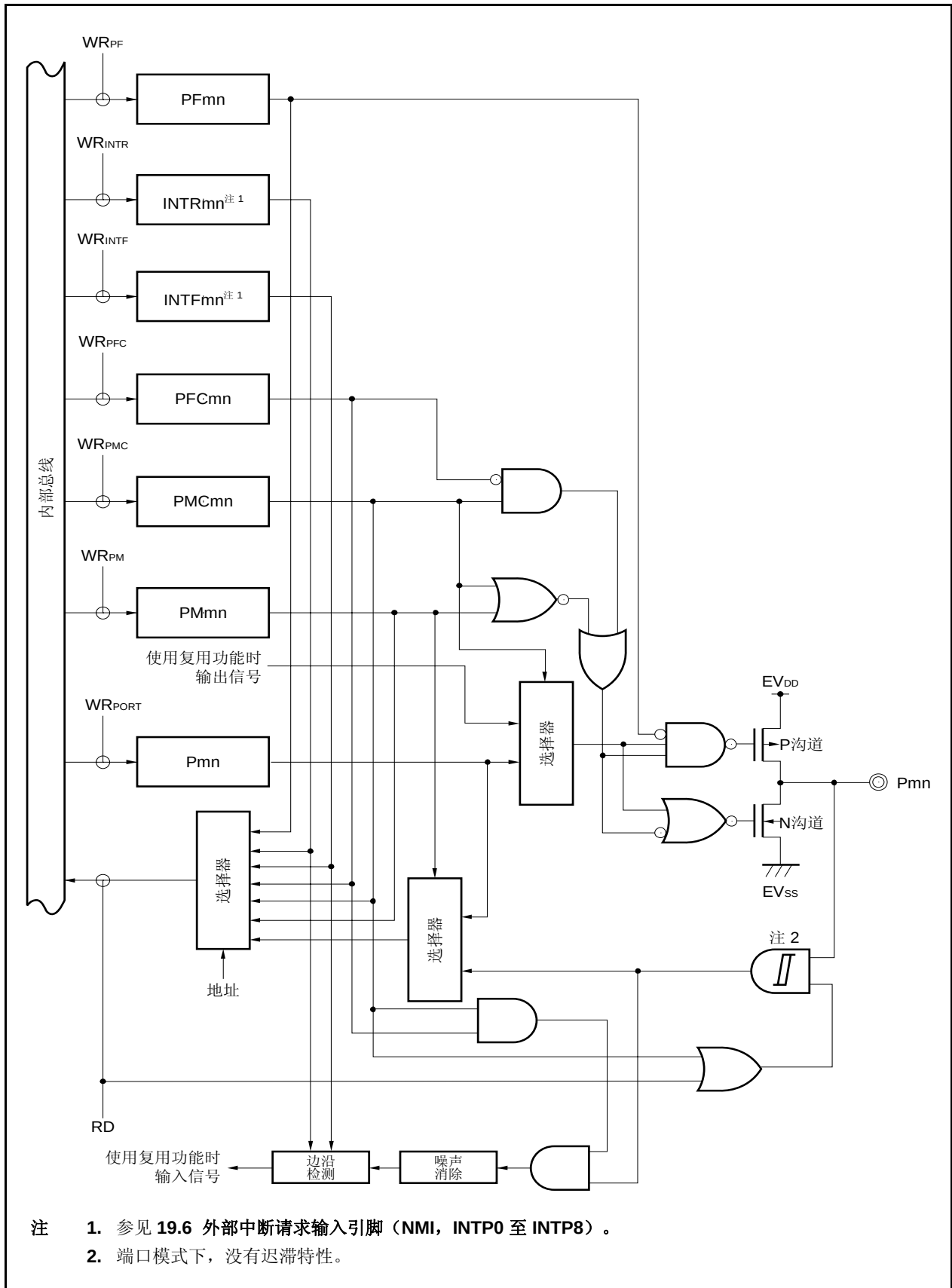


图 4-23. N-3 型的功能框图

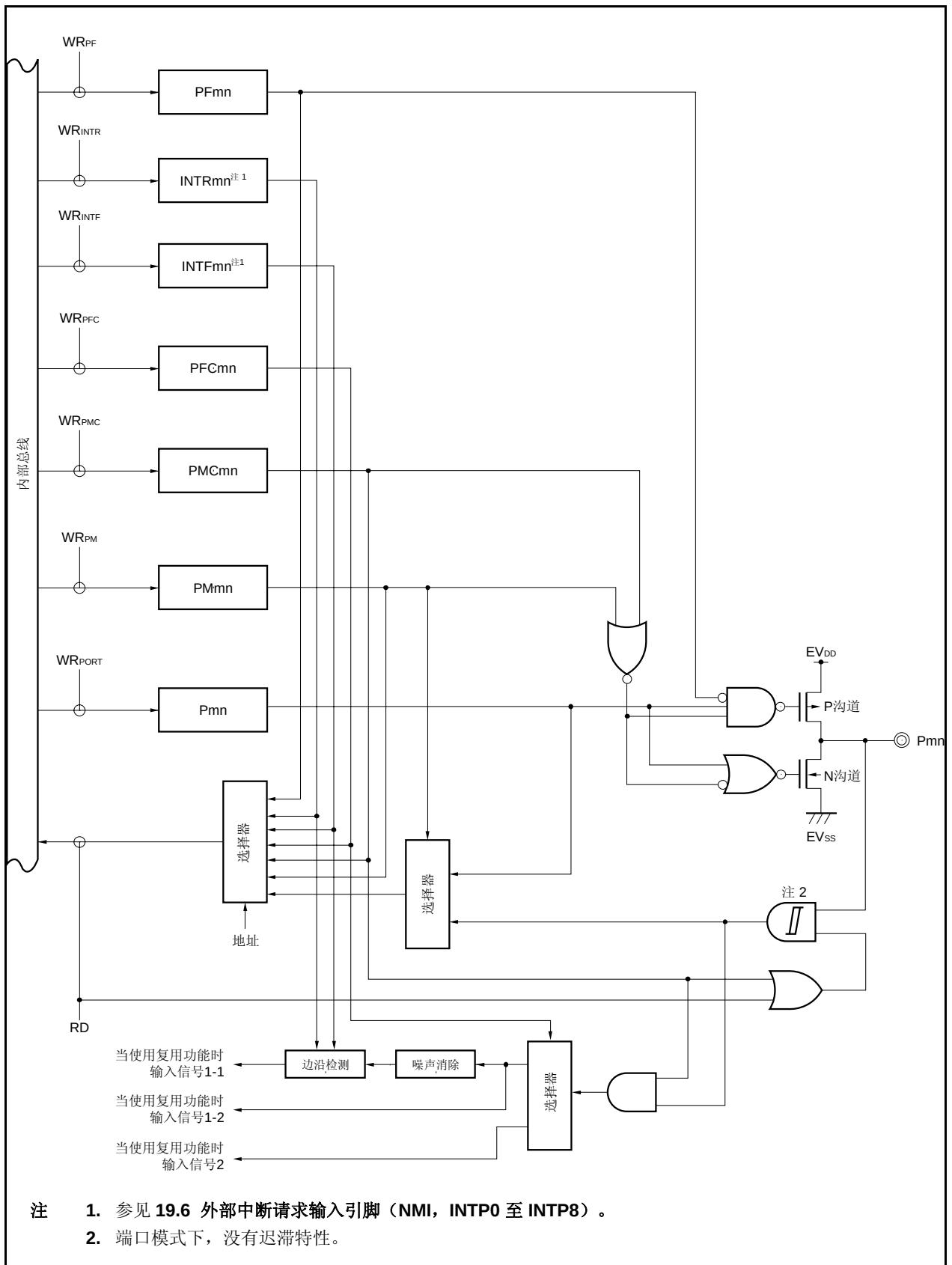


图 4-24. U-1 型的功能框图

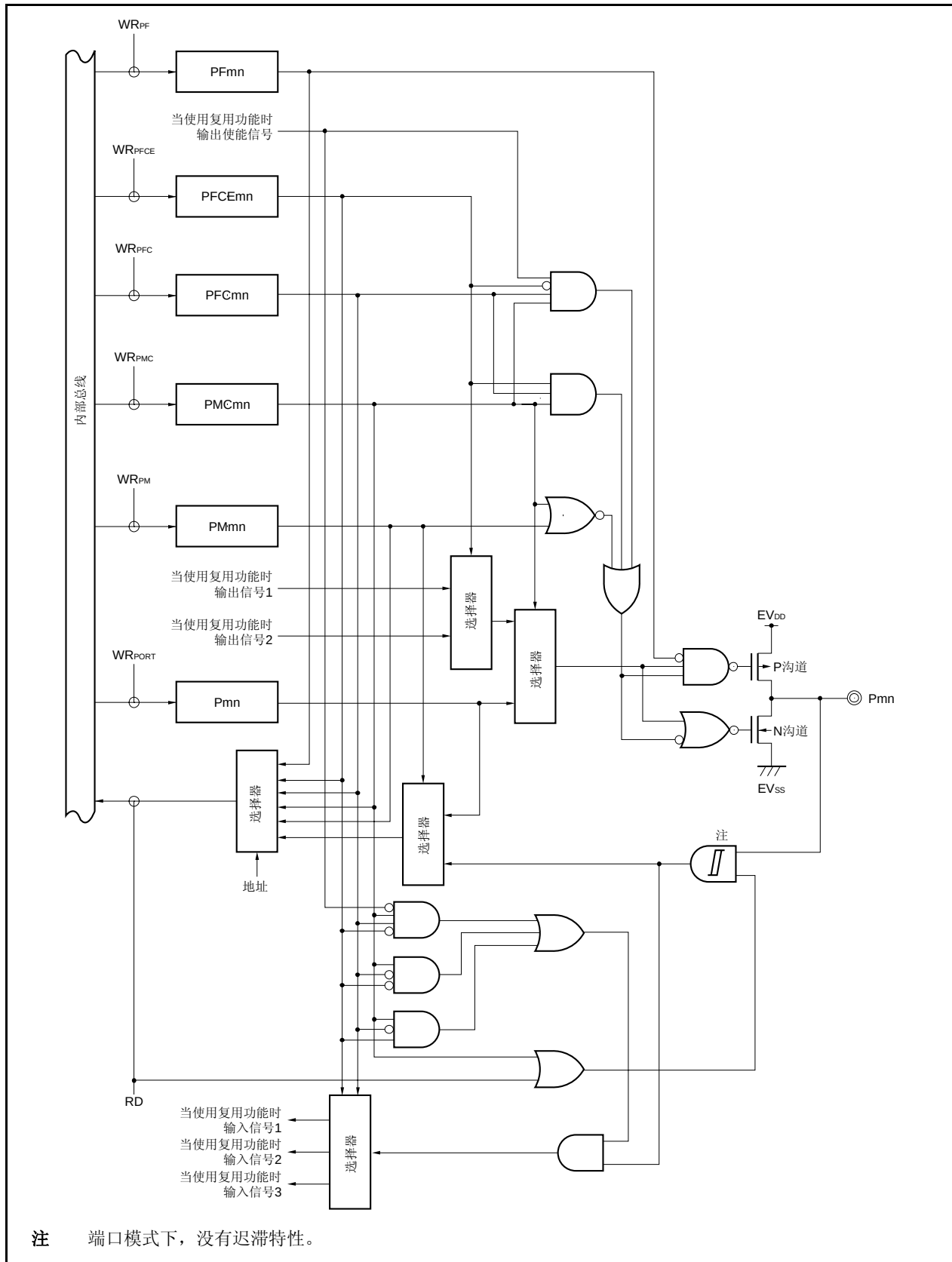


图 4-25. U-5 型的功能框图

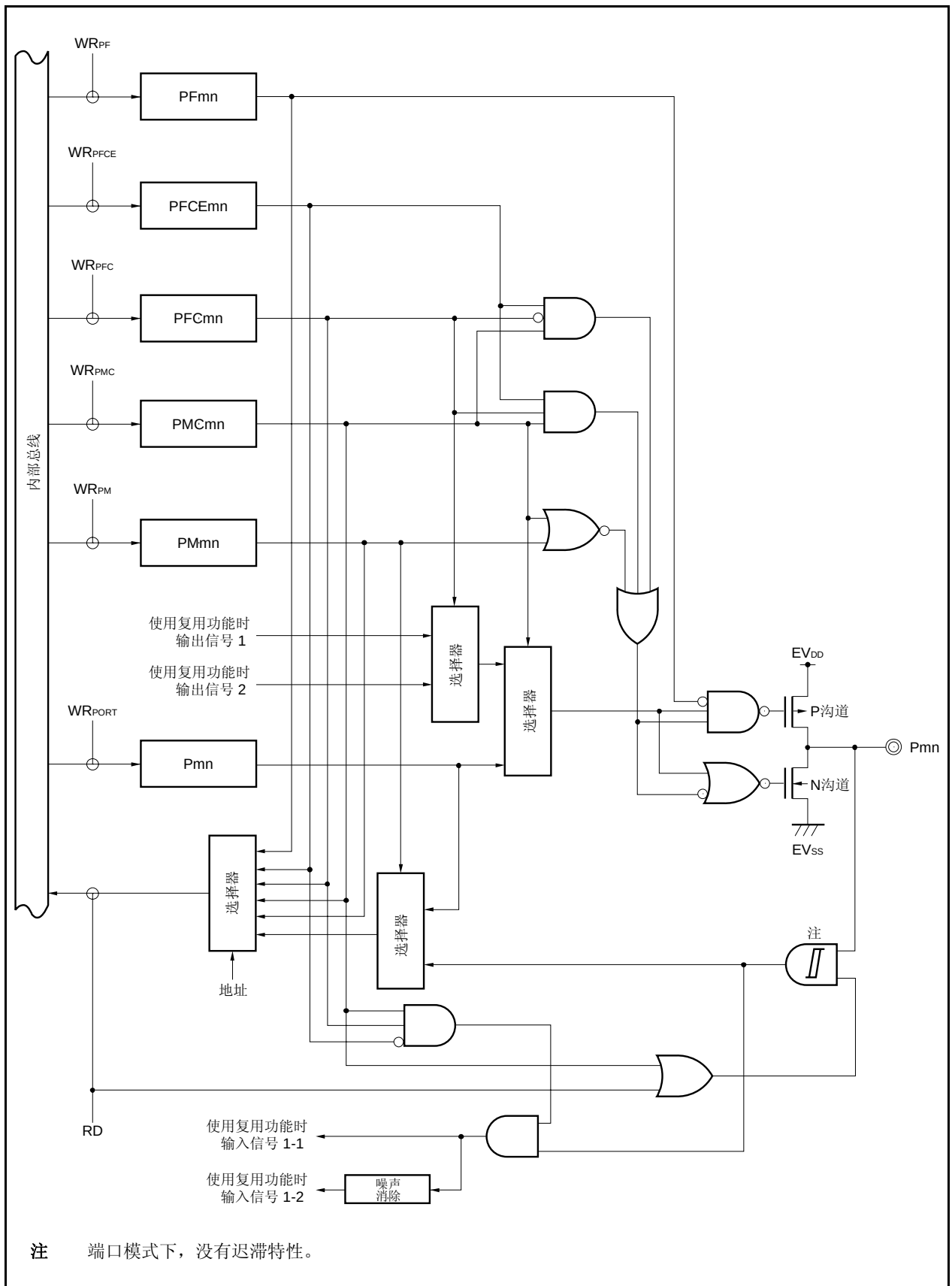


图 4-26. U-6 型的功能框图

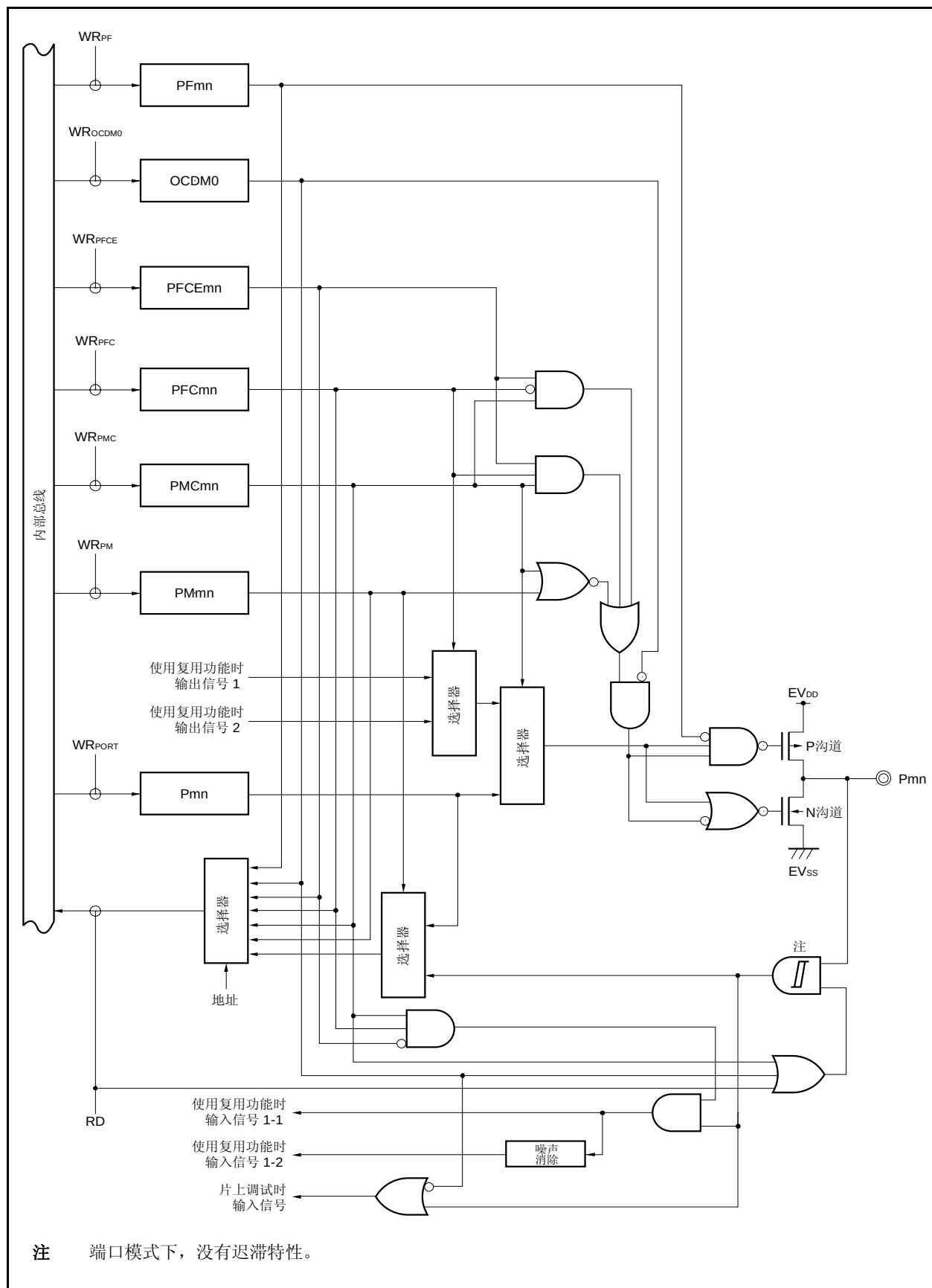


图 4-27. U-7 型的功能框图

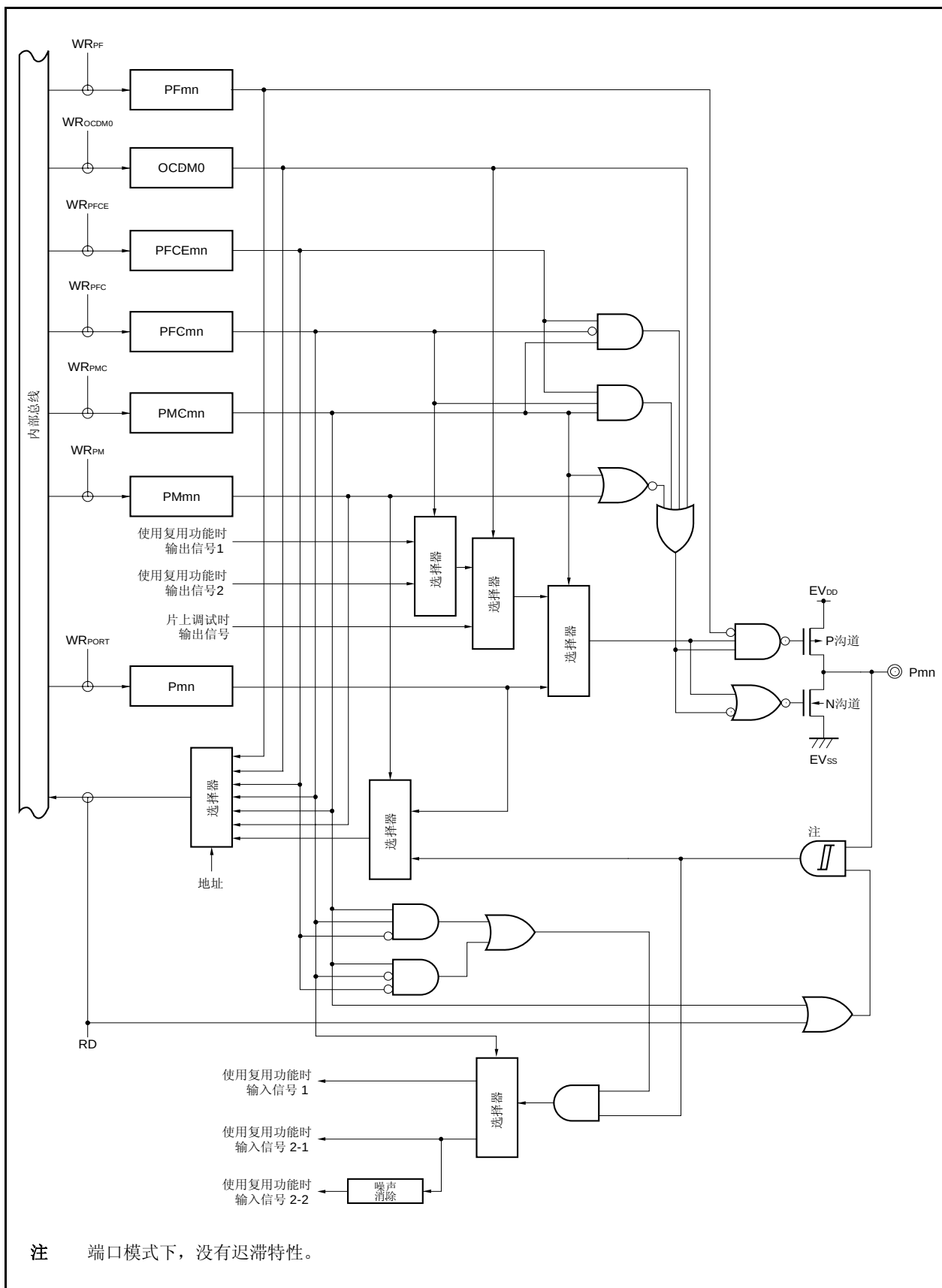


图 4-28. U-8 型的功能框图

图 4-29. U-9 型的功能框图

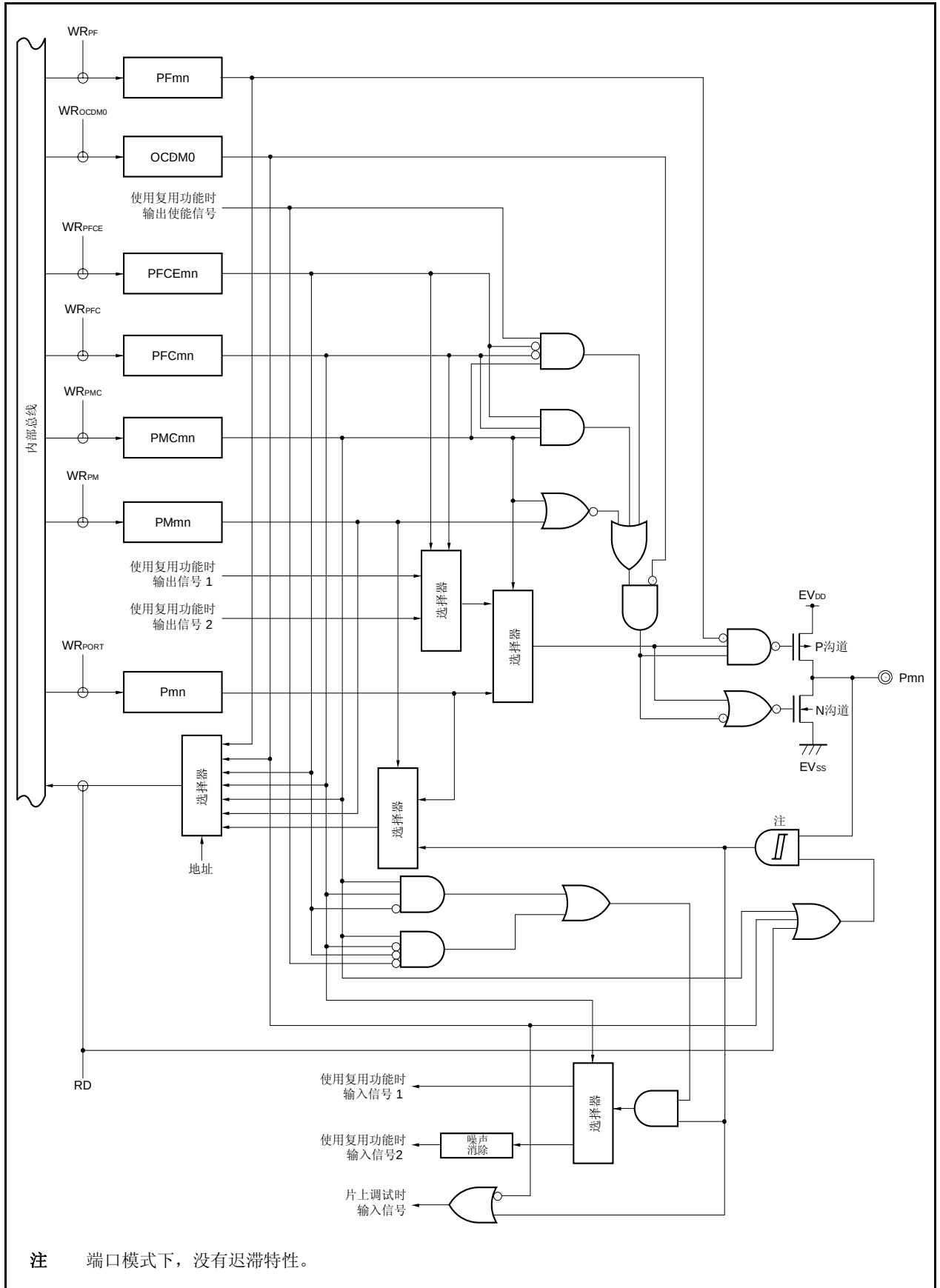


图 4-30. U-10 型的功能框图

图 4-31. U-11 型的功能框图

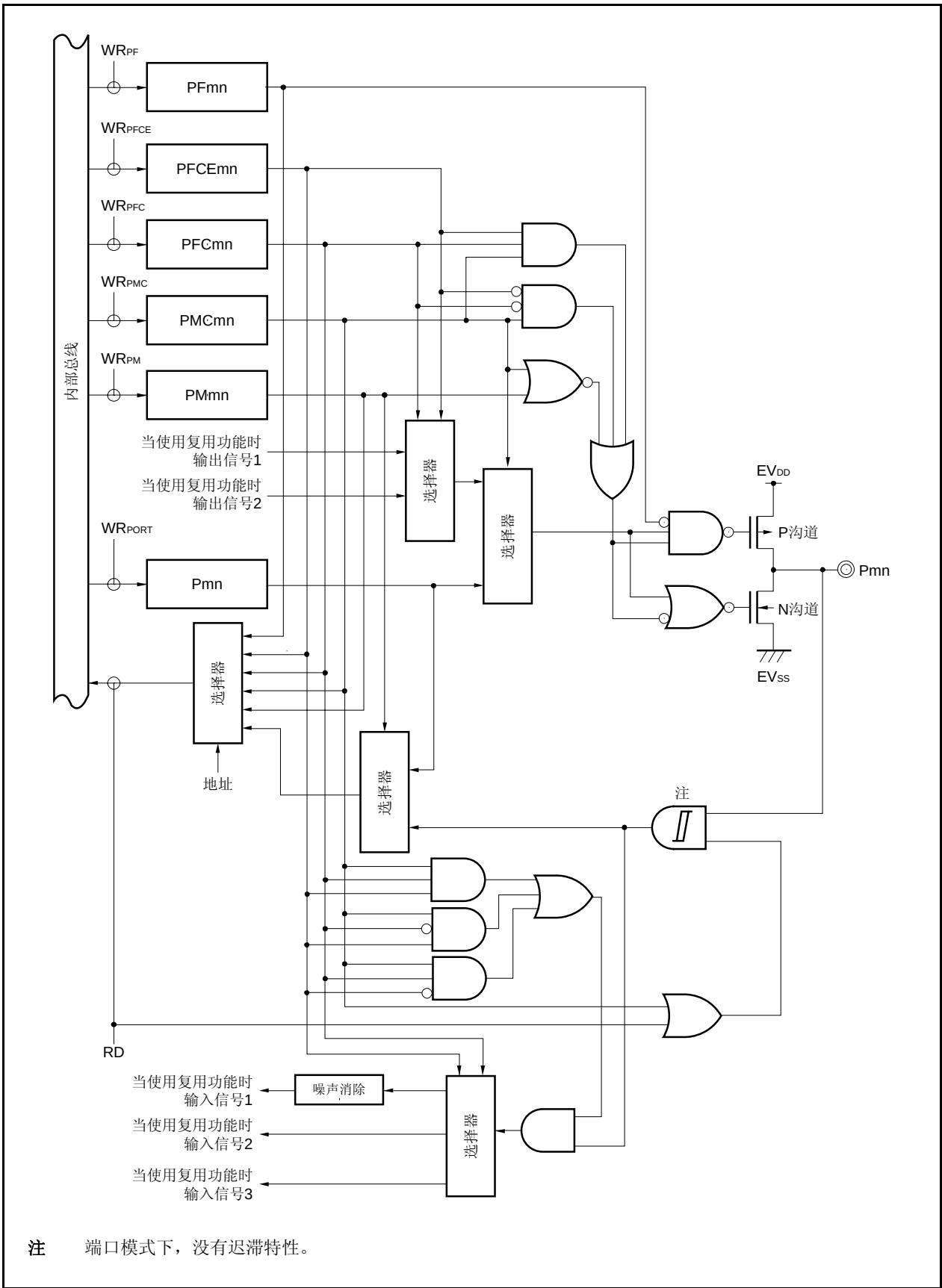


图 4-32. U-12 型的功能框图

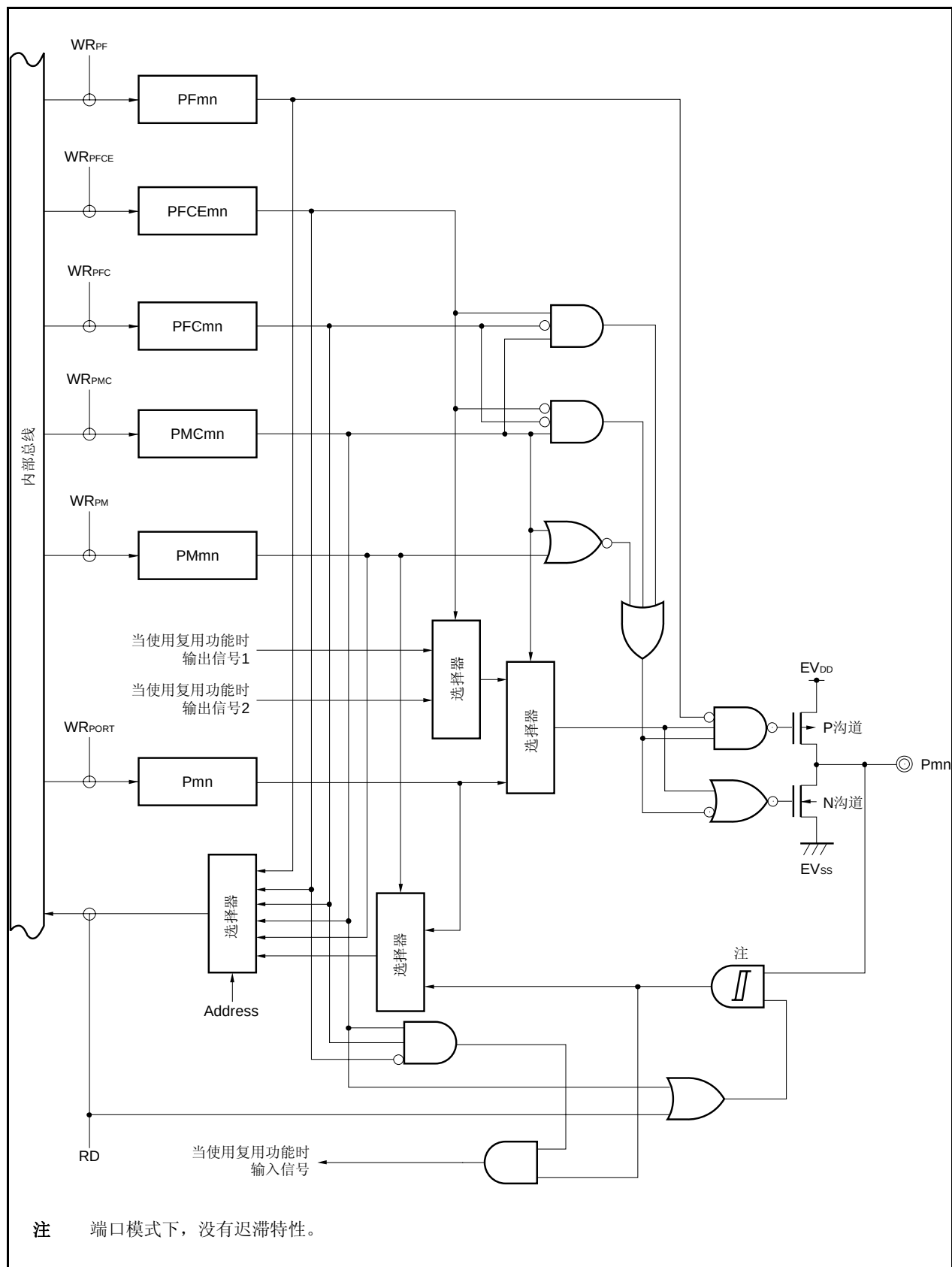


图 4-33. U-13 型的功能框图

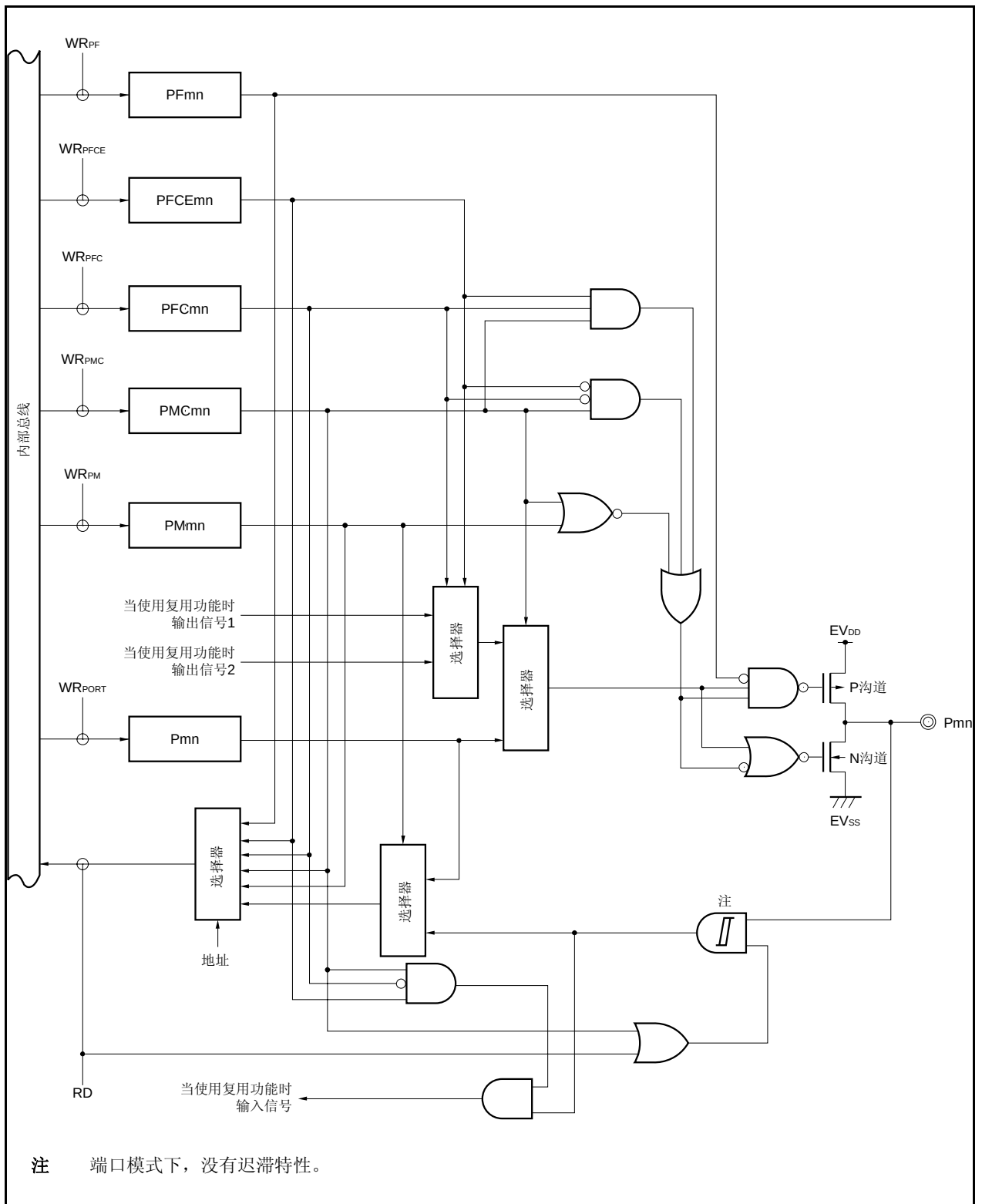


图 4-34. U-14 型的功能框图

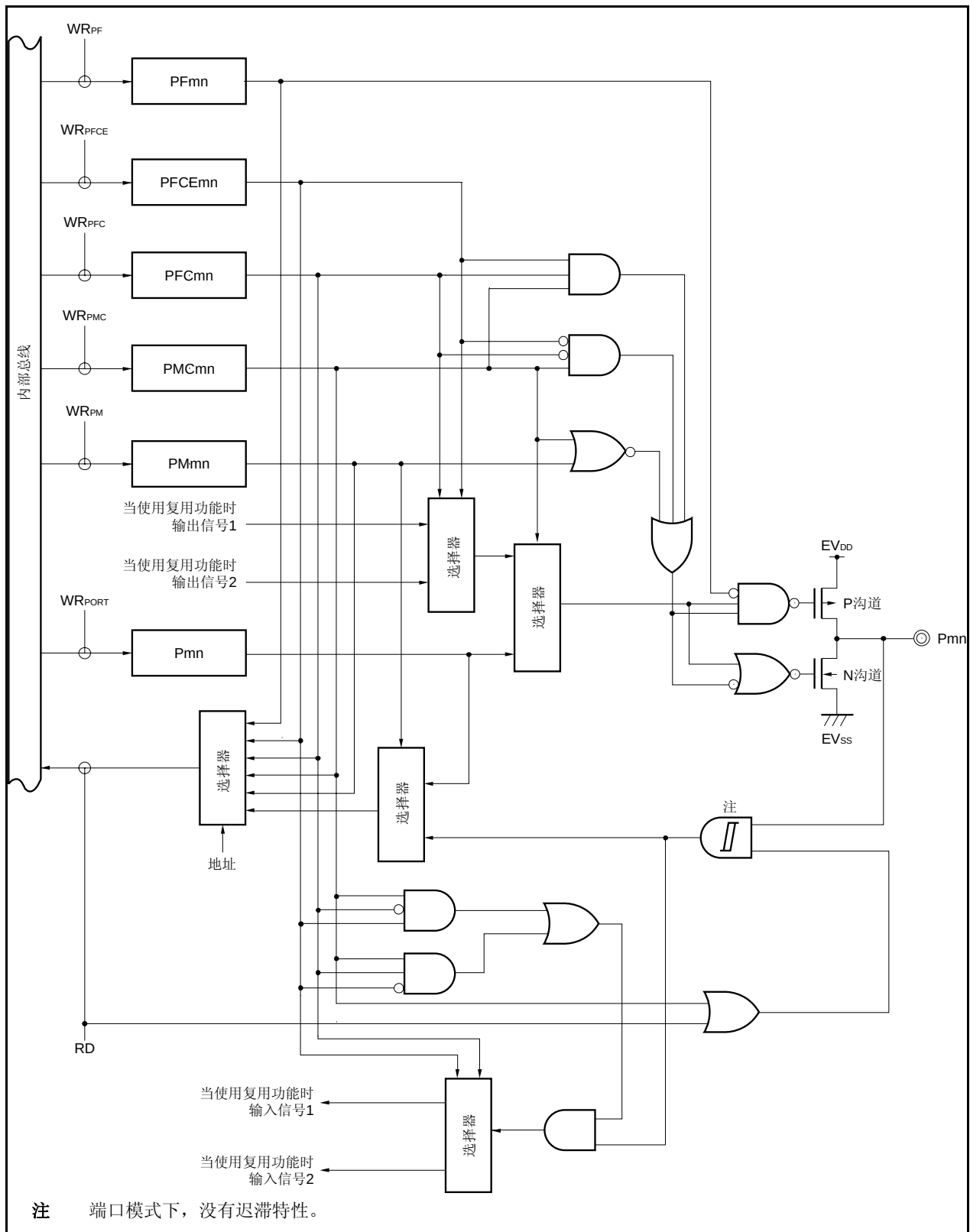


图 4-35. U-15 型的功能框图

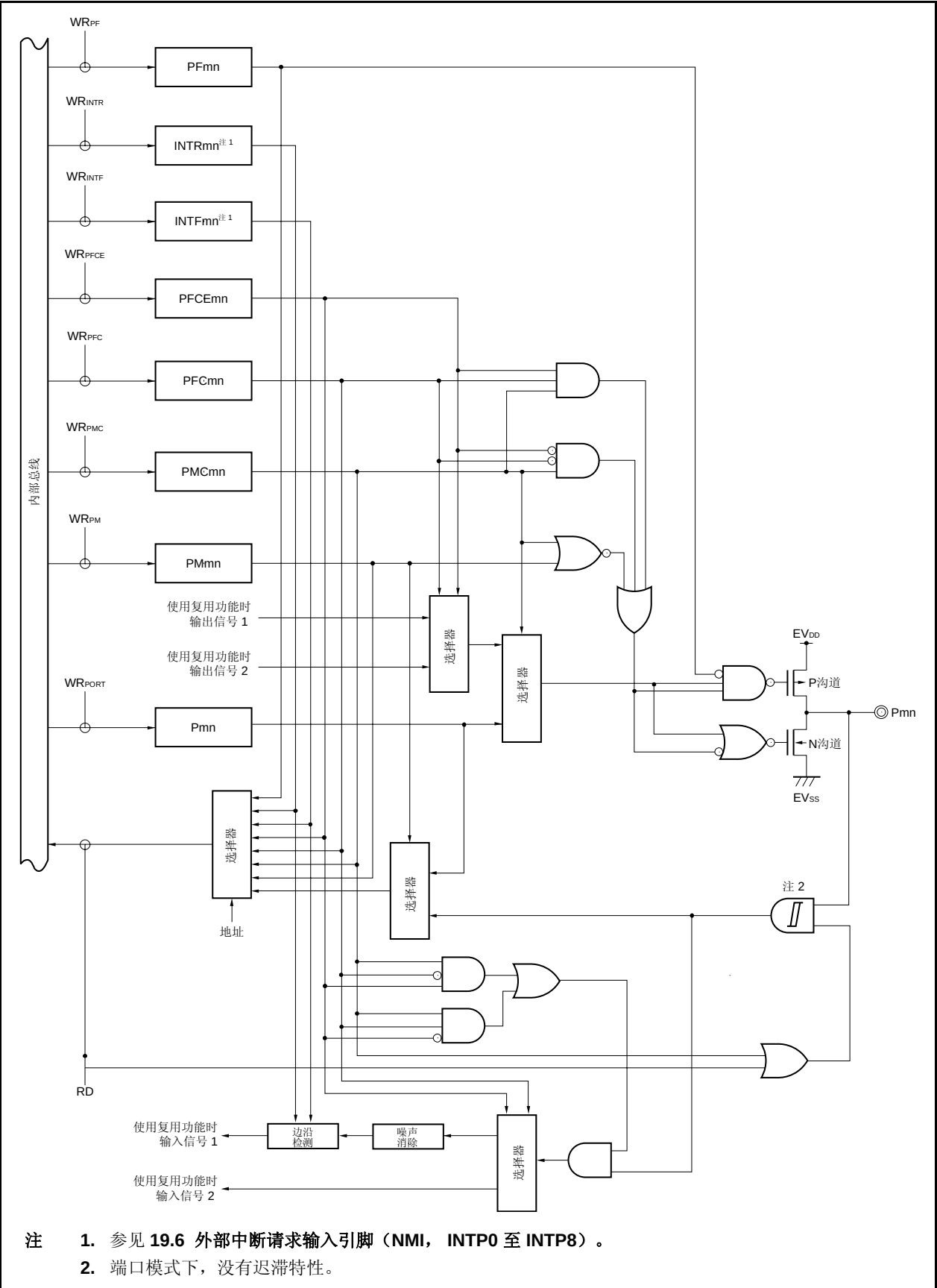
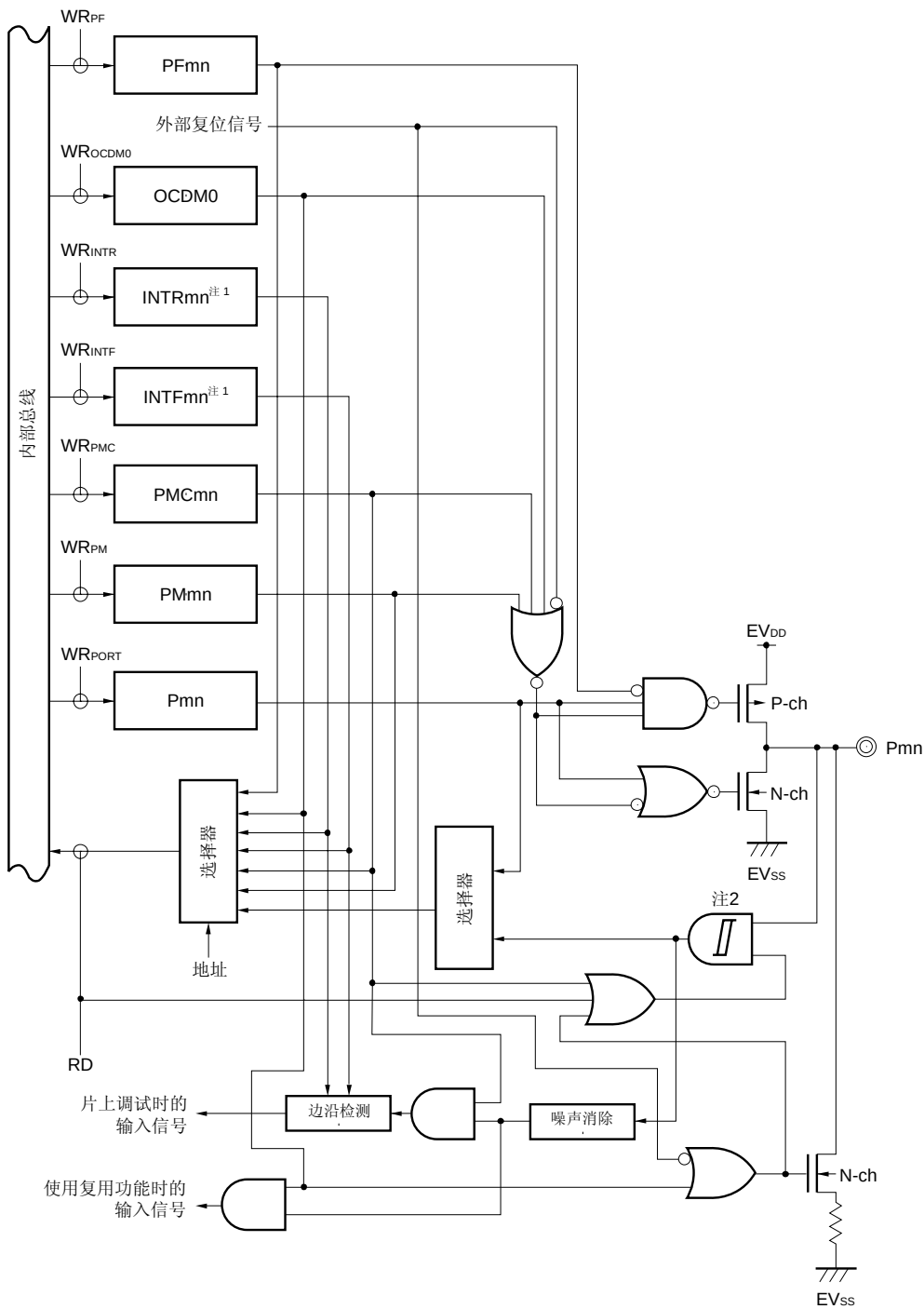


图 4-36. AA-1 型的功能框图



- 注 1. 参见 19.6 外部中断请求输入引脚（NMI 以及 INTP0 至 INTP8）。
2. 端口模式下，没有迟滞特性。

4.5 使用复用功能时的端口寄存器设置

表 4-19 显示各个端口用作复用功能时的端口寄存器设置。当一个端口引脚用作复用功能引脚时，请参考各个引脚的详细描述。

表 4-19. 端口引脚用作复用功能引脚（1/8）

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P00	TIP61	输入	P00 = 无需设置	PM00 = 无需设置	PMC00 = 1	–	PFC00 = 0	
	TOP61	输出	P00 = 无需设置	PM00 = 无需设置	PMC00 = 1	–	PFC00 = 1	
P01	TIP60	输入	P01 = 无需设置	PM01 = 无需设置	PMC01 = 1	–	PFC01 = 0	
	TOP60	输出	P01 = 无需设置	PM01 = 无需设置	PMC01 = 1	–	PFC01 = 1	
P02	NMI	输入	P02 = 无需设置	PM02 = 无需设置	PMC02 = 1	–	–	
P03	INTP0	输入	P03 = 无需设置	PM03 = 无需设置	PMC03 = 1	–	PFC03 = 0	
	ADTRG	输入	P03 = 无需设置	PM03 = 无需设置	PMC03 = 1	–	PFC03 = 1	
P04	INTP1	输入	P04 = 无需设置	PM04 = 无需设置	PMC04 = 1	–	–	
P05	INTP2	输入	P05 = 无需设置	PM05 = 无需设置	PMC05 = 1	–	–	
	DRST	输入	P05 = 无需设置	PM05 = 无需设置	PMC05 = 无需设置	–	–	OCDM0 (OCDM) = 1
P06	INTP3	输入	P06 = 无需设置	PM06 = 无需设置	PMC06 = 1	–	–	
P10	ANO0	输出	P10 = 无需设置	PM10 = 1	–	–	–	
P11	ANO1	输出	P11 = 无需设置	PM11 = 1	–	–	–	
P30	TXDA0	输出	P30 = 无需设置	PM30 = 无需设置	PMC30 = 1	–	PFC30 = 0	
	SOB4	输出	P30 = 无需设置	PM30 = 无需设置	PMC30 = 1	–	PFC30 = 1	
P31	RXDA0	输入	P31 = 无需设置	PM31 = 无需设置	PMC31 = 1	–	注, PFC31 = 0	
	INTP7	输入	P31 = 无需设置	PM31 = 无需设置	PMC31 = 1	–	注, PFC31 = 0	
	SIB4	输入	P31 = 无需设置	PM31 = 无需设置	PMC31 = 1	–	PFC31 = 1	
P32	ASCKA0	输入	P32 = 无需设置	PM32 = 无需设置	PMC32 = 1	PFCE32 = 0	PFC32 = 0	
	SCKB4	I/O	P32 = 无需设置	PM32 = 无需设置	PMC32 = 1	PFCE32 = 0	PFC32 = 1	
	TIP00	输入	P32 = 无需设置	PM32 = 无需设置	PMC32 = 1	PFCE32 = 1	PFC32 = 0	
	TOP00	输出	P32 = 无需设置	PM32 = 无需设置	PMC32 = 1	PFCE32 = 1	PFC32 = 1	

注 INTP7 和 RXDA0 是复用功能。当该引脚用作 RXDA0 时，禁止对复用功能 INTP7 引脚的边沿检测（将 INTF3.INTF31 和 INTR3.INTR31 位清 0）。当该引脚用作 INTP7 时，停止 UARTA0 接收操作（将 UA0CTL0.UA0RXE 位清除为 0）。

注意事项 当使用 P10 和 P11 其中一个引脚作为 I/O 端口，而另一个用作 D/A 输出引脚 (ANO0, ANO1)。在 D/A 输出期间，如果端口 I/O 电平不改变，则在应用系统中可以这样使用。

表 4-19. 端口引脚用作复用功能引脚 (2/8)

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P33	TIP01	输入	P33 = 无需设置	PM33 = 无需设置	PMC33 = 1	–	PFC33 = 0	
	TOP01	输出	P33 = 无需设置	PM33 = 无需设置	PMC33 = 1	–	PFC33 = 1	
P34	TIP10	输入	P34 = 无需设置	PM34 = 无需设置	PMC34 = 1	–	PFC34 = 0	
	TOP10	输入	P34 = 无需设置	PM34 = 无需设置	PMC34 = 1	–	PFC34 = 1	
P35	TIP11	输入	P35 = 无需设置	PM35 = 无需设置	PMC35 = 1	–	PFC35 = 0	
	TOP11	输出	P35 = 无需设置	PM35 = 无需设置	PMC35 = 1	–	PFC35 = 1	
P38	TXDA2	输出	P38 = 无需设置	PM38 = 无需设置	PMC38 = 1	–	PFC38 = 0	
	SDA00	I/O	P38 = 无需设置	PM38 = 无需设置	PMC38 = 1	–	PFC38 = 1	PF38 (PF3) = 1
P39	RXDA2	输入	P39 = 无需设置	PM39 = 无需设置	PMC39 = 1	–	PFC39 = 0	
	SCL00	I/O	P39 = 无需设置	PM39 = 无需设置	PMC39 = 1	–	PFC39 = 1	PF39 (PF3) = 1
P40	SIB0	输入	P40 = 无需设置	PM40 = 无需设置	PMC40 = 1	–	PFC40 = 0	
	SDA01	I/O	P40 = 无需设置	PM40 = 无需设置	PMC40 = 1	–	PFC40 = 1	PF40 (PF4) = 1
P41	SOB0	输出	P41 = 无需设置	PM41 = 无需设置	PMC41 = 1	–	PFC41 = 0	
	SCL01	I/O	P41 = 无需设置	PM41 = 无需设置	PMC41 = 1	–	PFC41 = 1	PF41 (PF4) = 1
P42	SCKB0	I/O	P42 = 无需设置	PM42 = 无需设置	PMC42 = 1	–	–	
P50	TIQ01	输入	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 0	PFC50 = 1	KRM0 (KRM) = 0
	KR0	输入	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 0	PFC50 = 1	TQ0TIG2, TQ0TIG3 (TQ0IOC1) = 0
	TOQ01	输出	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 1	PFC50 = 0	
	RTP00	输出	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 1	PFC50 = 1	
P51	TIQ02	输入	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 0	PFC51 = 1	KRM1 (KRM) = 0
	KR1	输入	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 0	PFC51 = 1	TQ0TIG4, TQ0TIG5 (TQ0IOC1) = 0
	TOQ02	输出	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 1	PFC51 = 0	
	RTP01	输出	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 1	PFC51 = 1	
P52	TIQ03	输入	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 0	PFC52 = 1	KRM2 (KRM) = 0
	KR2	输入	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 0	PFC52 = 1	TQ0TIG6, TQ0TIG7 (TQ0IOC1) = 0
	TOQ03	输出	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 1	PFC52 = 0	
	RTP02	输出	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 1	PFC52 = 1	
	DDI	输入	P52 = 无需设置	PM52 = 无需设置	PMC52 = 无需设置	PFCE52 = 无需设置	PFC52 = 无需设置	OCDM0 (OCDM) = 1

表 4-19. 端口引脚用作复用功能引脚（3/8）

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P53	SIB2	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 0	PFC53 = 0	
	TIQ00	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 0	PFC53 = 1	KRM3 (KRM) = 0
	KR3	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 0	PFC53 = 1	TQ0TIG0, TQ0TIG1 (TQ0IOC1) = 0, TQ0EES0, TQ0EES1 (TQ0IOC2) = 0, TQ0ETS0, TQ0ETS1 (TQ0IOC2) = 0
	TOQ00	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 1	PFC53 = 0	
	RTP03	输出	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 1	PFC53 = 1	
	DDO	输出	P53 = 无需设置	PM53 = 无需设置	PMC53 = 无需设置	PFCE53 = 无需设置	PFC53 = 无需设置	OCDM0 (OCDM) = 1
P54	SOB2	输出	P54 = 无需设置	PM54 = 无需设置	PMC54 = 1	PFCE54 = 0	PFC54 = 0	
	KR4	输入	P54 = 无需设置	PM54 = 无需设置	PMC54 = 1	PFCE54 = 0	PFC54 = 1	
	RTP04	输出	P54 = 无需设置	PM54 = 无需设置	PMC54 = 1	PFCE54 = 1	PFC54 = 1	
	DCK	输入	P54 = 无需设置	PM54 = 无需设置	PMC54 = 无需设置	PFCE54 = 无需设置	PFC54 = 无需设置	OCDM0 (OCDM) = 1
P55	SCKB2	I/O	P55 = 无需设置	PM55 = 无需设置	PMC55 = 1	PFCE55 = 0	PFC55 = 0	
	KR5	输入	P55 = 无需设置	PM55 = 无需设置	PMC55 = 1	PFCE55 = 0	PFC55 = 1	
	RTP05	输出	P55 = 无需设置	PM55 = 无需设置	PMC55 = 1	PFCE55 = 1	PFC55 = 1	
	DMS	输入	P55 = 无需设置	PM55 = 无需设置	PMC55 = 无需设置	PFCE55 = 无需设置	PFC55 = 无需设置	OCDM0 (OCDM) = 1
P60	RTP10	输出	P60 = 无需设置	PM60 = 无需设置	PMC60 = 1	—	—	
P61	RTP11	输出	P61 = 无需设置	PM61 = 无需设置	PMC61 = 1	—	—	
P62	RTP12	输出	P62 = 无需设置	PM62 = 无需设置	PMC62 = 1	—	—	
P63	RTP13	输出	P63 = 无需设置	PM63 = 无需设置	PMC63 = 1	—	—	
P64	RTP14	输出	P64 = 无需设置	PM64 = 无需设置	PMC64 = 1	—	—	
P65	RTP15	输出	P65 = 无需设置	PM65 = 无需设置	PMC65 = 1	—	—	
P66	SIB5	输入	P66 = 无需设置	PM66 = 无需设置	PMC66 = 1	—	—	
P67	SOB5	输出	P67 = 无需设置	PM67 = 无需设置	PMC67 = 1	—	—	
P68	SCKB5	I/O	P68 = 无需设置	PM68 = 无需设置	PMC68 = 1	—	—	
P69	TIP70	输入	P69 = 无需设置	PM69 = 无需设置	PMC69 = 1	—	PFC69 = 0	
	TOP70	输出	P69 = 无需设置	PM69 = 无需设置	PMC69 = 1	—	PFC69 = 1	
P610	TIP71	输入	P610 = 无需设置	PM610 = 无需设置	PMC610 = 1	—	—	
P611	TOP71	输出	P611 = 无需设置	PM611 = 无需设置	PMC611 = 1	—	—	

表 4-19. 端口引脚用作复用功能引脚（4/8）

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P612	TIP80	输入	P612 = 无需设置	PM612 = 无需设置	PMC612 = 1	–	PFC612 = 0	
	TOP80	输出	P612 = 无需设置	PM612 = 无需设置	PMC612 = 1	–	PFC612 = 1	
P613	TIP81	输入	P613 = 无需设置	PM613 = 无需设置	PMC613 = 1	–	PFC613 = 0	
	TOP81	输出	P613 = 无需设置	PM613 = 无需设置	PMC613 = 1	–	PFC613 = 1	
P70	ANI0	输入	P70 = 无需设置	PM70 = 1	–	–	–	
P71	ANI1	输入	P71 = 无需设置	PM71 = 1	–	–	–	
P72	ANI2	输入	P72 = 无需设置	PM72 = 1	–	–	–	
P73	ANI3	输入	P73 = 无需设置	PM73 = 1	–	–	–	
P74	ANI4	输入	P74 = 无需设置	PM74 = 1	–	–	–	
P75	ANI5	输入	P75 = 无需设置	PM75 = 1	–	–	–	
P76	ANI6	输入	P76 = 无需设置	PM76 = 1	–	–	–	
P77	ANI7	输入	P77 = 无需设置	PM77 = 1	–	–	–	
P78	ANI8	输入	P78 = 无需设置	PM78 = 1	–	–	–	
P79	ANI9	输入	P79 = 无需设置	PM79 = 1	–	–	–	
P710	ANI10	输入	P710 = 无需设置	PM710 = 1	–	–	–	
P711	ANI11	输入	P711 = 无需设置	PM711 = 1	–	–	–	
P712	ANI12	输入	P712 = 无需设置	PM712 = 1	–	–	–	
P713	ANI13	输入	P713 = 无需设置	PM713 = 1	–	–	–	
P714	ANI14	输入	P714 = 无需设置	PM714 = 1	–	–	–	
P715	ANI15	输入	P715 = 无需设置	PM715 = 1	–	–	–	
P80	RXDA3	输入	P80 = 无需设置	PM80 = 无需设置	PMC80 = 1	–	–	注
	INTP8	输入	P80 = 无需设置	PM80 = 无需设置	PMC80 = 1	–	–	注
P81	TXDA3	输出	P81 = 无需设置	PM81 = 无需设置	PMC81 = 1	–	–	

注 INTP8 引脚及 RXDA3 引脚为复用功引脚。当该引脚用作 RXDA3 引脚时，请禁止 INTP8 复用功能引脚的边沿检测（清除 INTF8.INTF80 位 = 0 和 INTR8.INTR80 位 = 0）。当该引脚被用作 INTP8 引脚时，停止 UARTA3 接收（清除 UA3CTL0.UA3RXE 位为 0）。

表 4-19. 端口引脚用作复用功能引脚（5/8）

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P90	A0	输出	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 0	PFC90 = 0	注 1
	KR6	输入	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 0	PFC90 = 1	
	TXDA1	输出	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 1	PFC90 = 0	
	SDA02	I/O	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 1	PFC90 = 1	PF90 (PF9) = 1
P91	A1	输出	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 0	PFC91 = 0	注 1
	KR7	输入	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 0	PFC91 = 1	
	RXDA1/KR7 ^{※2}	输入	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 1	PFC91 = 0	
	SCL02	I/O	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 1	PFC91 = 1	PF91 (PF9) = 1
P92	A2	输出	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 0	PFC92 = 0	注 1
	TIP41	输入	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 0	PFC92 = 1	
	TOP41	输出	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 1	PFC92 = 0	
P93	A3	输出	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 0	PFC93 = 0	注 1
	TIP40	输入	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 0	PFC93 = 1	
	TOP40	输出	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 1	PFC93 = 0	
P94	A4	输出	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 0	PFC94 = 0	注 1
	TIP31	输入	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 0	PFC94 = 1	
	TOP31	输出	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 1	PFC94 = 0	
P95	A5	输出	P95 = 无需设置	PM95 = 无需设置	PMC95 = 1	PFCE95 = 0	PFC95 = 0	注 1
	TIP30	输入	P95 = 无需设置	PM95 = 无需设置	PMC95 = 1	PFCE95 = 0	PFC95 = 1	
	TOP30	输出	P95 = 无需设置	PM95 = 无需设置	PMC95 = 1	PFCE95 = 1	PFC95 = 0	
P96	A6	输出	P96 = 无需设置	PM96 = 无需设置	PMC96 = 1	PFCE96 = 0	PFC96 = 0	注 1
	TIP21	输入	P96 = 无需设置	PM96 = 无需设置	PMC96 = 1	PFCE96 = 1	PFC96 = 0	
	TOP21	输出	P96 = 无需设置	PM96 = 无需设置	PMC96 = 1	PFCE96 = 1	PFC96 = 1	

注 1. 当 A0 至 A15 引脚用作复用功能时，请将 PMC9 寄存器的所有 16 位一次设置为 FFFFH。

2. RXDA1 和 KR7 功能不能同时使用。用作 RXDA1 引脚时，不要使用 KR7 引脚功能。而用作 KR7 引脚时，不要使用 RXDA1 引脚功能（建议将 PFC91 位设置为 1，并且将 PFCE91 位设置为 0）。

表 4-19. 端口引脚用作复用功能引脚（6/8）

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P97	A7	输出	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 0	PFC97 = 0	注
	SIB1	输入	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 0	PFC97 = 1	
	TIP20	输入	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 1	PFC97 = 0	
	TOP20	输出	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 1	PFC97 = 1	
P98	A8	输出	P98 = 无需设置	PM98 = 无需设置	PMC98 = 1	–	PFC98 = 0	注
	SOB1	输出	P98 = 无需设置	PM98 = 无需设置	PMC98 = 1	–	PFC98 = 1	
P99	A9	输出	P99 = 无需设置	PM99 = 无需设置	PMC99 = 1	–	PFC99 = 0	注
	SCKB1	I/O	P99 = 无需设置	PM99 = 无需设置	PMC99 = 1	–	PFC99 = 1	
P910	A10	输出	P910 = 无需设置	PM910 = 无需设置	PMC910 = 1	–	PFC910 = 0	注
	SIB3	输入	P910 = 无需设置	PM910 = 无需设置	PMC910 = 1	–	PFC910 = 1	
P911	A11	输出	P911 = 无需设置	PM911 = 无需设置	PMC911 = 1	–	PFC911 = 0	注
	SOB3	输出	P911 = 无需设置	PM911 = 无需设置	PMC911 = 1	–	PFC911 = 1	
P912	A12	输出	P912 = 无需设置	PM912 = 无需设置	PMC912 = 1	–	PFC912 = 0	注
	SCKB3	I/O	P912 = 无需设置	PM912 = 无需设置	PMC912 = 1	–	PFC912 = 1	
P913	A13	输出	P913 = 无需设置	PM913 = 无需设置	PMC913 = 1	–	PFC913 = 0	注
	INTP4	输入	P913 = 无需设置	PM913 = 无需设置	PMC913 = 1	–	PFC913 = 1	
P914	A14	输出	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 0	PFC914 = 0	注
	INTP5	输入	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 0	PFC914 = 1	
	TIP51	输入	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 1	PFC914 = 0	
	TOP51	输出	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 1	PFC914 = 1	
P915	A15	输出	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 0	PFC915 = 0	注
	INTP6	输入	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 0	PFC915 = 1	
	TIP50	输入	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 1	PFC915 = 0	
	TOP50	输出	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 1	PFC915 = 1	
PCS0	$\overline{\text{CS0}}$	输出	PCS0 = 无需设置	PMCS0 = 无需设置	PMCCS0 = 1	–	–	
PCS1	$\overline{\text{CS1}}$	输出	PCS1 = 无需设置	PMCS1 = 无需设置	PMCCS1 = 1	–	–	
PCS2	$\overline{\text{CS2}}$	输出	PCS2 = 无需设置	PMCS2 = 无需设置	PMCCS2 = 1	–	–	
PCS3	$\overline{\text{CS3}}$	输出	PCS3 = 无需设置	PMCS3 = 无需设置	PMCCS3 = 1	–	–	

注 当 A0 至 A15 引脚用作复用功能时，请将 PMC9 寄存器的所有 16 位一次设置为 FFFFH。

表 4-19. 端口引脚用作复用功能引脚（7/8）

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
PCM0	WAIT	输入	PCM0 = 无需设置	PMCM0 = 无需设置	PMCCM0 = 1	—	—	
PCM1	CLKOUT	输出	PCM1 = 无需设置	PMCM1 = 无需设置	PMCCM1 = 1	—	—	
PCM2	HLD $\overline{\text{AK}}$	输出	PCM2 = 无需设置	PMCM2 = 无需设置	PMCCM2 = 1	—	—	
PCM3	HLD $\overline{\text{RQ}}$	输入	PCM3 = 无需设置	PMCM3 = 无需设置	PMCCM3 = 1	—	—	
PCT0	WR $\overline{0}$	输出	PCT0 = 无需设置	PMCT0 = 无需设置	PMCC $\overline{\text{T0}}$ = 1	—	—	
PCT1	WR $\overline{1}$	输出	PCT1 = 无需设置	PMCT1 = 无需设置	PMCC $\overline{\text{T1}}$ = 1	—	—	
PCT4	RD $\overline{}$	输出	PCT4 = 无需设置	PMCT4 = 无需设置	PMCC $\overline{\text{T4}}$ = 1	—	—	
PCT6	ASTB	输出	PCT6 = 无需设置	PMCT6 = 无需设置	PMCC $\overline{\text{T6}}$ = 1	—	—	
PDH0	A16	输出	PDH0 = 无需设置	PMDH0 = 无需设置	PMCDH0 = 1	—	—	
PDH1	A17	输出	PDH1 = 无需设置	PMDH1 = 无需设置	PMCDH1 = 1	—	—	
PDH2	A18	输出	PDH2 = 无需设置	PMDH2 = 无需设置	PMCDH2 = 1	—	—	
PDH3	A19	输出	PDH3 = 无需设置	PMDH3 = 无需设置	PMCDH3 = 1	—	—	
PDH4	A20	输出	PDH4 = 无需设置	PMDH4 = 无需设置	PMCDH4 = 1	—	—	
PDH5	A21	输出	PDH5 = 无需设置	PMDH5 = 无需设置	PMCDH5 = 1	—	—	
PDH6	A22	输出	PDH6 = 无需设置	PMDH6 = 无需设置	PMCDH6 = 1	—	—	
PDH7	A23	输出	PDH7 = 无需设置	PMDH7 = 无需设置	PMCDH7 = 1	—	—	
PDL0	AD0	I/O	PDL0 = 无需设置	PMDL0 = 无需设置	PMCDL0 = 1	—	—	
PDL1	AD1	I/O	PDL1 = 无需设置	PMDL1 = 无需设置	PMCDL1 = 1	—	—	
PDL2	AD2	I/O	PDL2 = 无需设置	PMDL2 = 无需设置	PMCDL2 = 1	—	—	
PDL3	AD3	I/O	PDL3 = 无需设置	PMDL3 = 无需设置	PMCDL3 = 1	—	—	
PDL4	AD4	I/O	PDL4 = 无需设置	PMDL4 = 无需设置	PMCDL4 = 1	—	—	
PDL5	AD5	I/O	PDL5 = 无需设置	PMDL5 = 无需设置	PMCDL5 = 1	—	—	
	FLMD1 ^注	输入	PDL5 = 无需设置	PMDL5 = 无需设置	PMCDL5 = 无需设置	—	—	
PDL6	AD6	I/O	PDL6 = 无需设置	PMDL6 = 无需设置	PMCDL6 = 1	—	—	
PDL7	AD7	I/O	PDL7 = 无需设置	PMDL7 = 无需设置	PMCDL7 = 1	—	—	

注 该引脚设置为 flash 存储器编程模式，不需要通过端口控制寄存器进行操作。详情参见第二十七章 FLASH 存储器。

表 4-19. 端口引脚用作复用功能引脚（8/8）

引脚名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
PDL8	AD8	I/O	PDL8 = 无需设置	PMDL8 = 无需设置	PMCDL8 = 1	–	–	
PDL9	AD9	I/O	PDL9 = 无需设置	PMDL9 = 无需设置	PMCDL9 = 1	–	–	
PDL10	AD10	I/O	PDL10 = 无需设置	PMDL10 = 无需设置	PMCDL10 = 1	–	–	
PDL11	AD11	I/O	PDL11 = 无需设置	PMDL11 = 无需设置	PMCDL11 = 1	–	–	
PDL12	AD12	I/O	PDL12 = 无需设置	PMDL12 = 无需设置	PMCDL12 = 1	–	–	
PDL13	AD13	I/O	PDL13 = 无需设置	PMDL13 = 无需设置	PMCDL13 = 1	–	–	
PDL14	AD14	I/O	PDL14 = 无需设置	PMDL14 = 无需设置	PMCDL14 = 1	–	–	
PDL15	AD15	I/O	PDL15 = 无需设置	PMDL15 = 无需设置	PMCDL15 = 1	–	–	

4.6 注意事项

4.6.1 设置端口引脚的注意事项

- (1) V850ES/JJ3 中，通用端口功能和一些外围功能 I/O 引脚共用一个引脚。通过 PMCn 寄存器的设置，可以在通用端口（端口模式）与外围功能 I/O 引脚（复用功能模式）之间相互转换。有关该寄存器的设置步骤，注意以下事项：

(a) 从端口模式转换至复用功能模式的注意事项

从端口模式至复用功能模式的转换操作按以下顺序进行：

- | | |
|-------------------------------|------------|
| <1> 设置 PFn 寄存器 [※] ： | N 沟道漏极开路设置 |
| <2> 设置 PFCn 和 PFCEn 寄存器： | 复用功能选择 |
| <3> 设置 PMCn 寄存器的相应位为 1： | 转换至复用功能模式 |

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

下面会有一个具体举例。

注 只有 N 沟道漏极开路输出引脚

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

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

[实例] SCL01 引脚设置举例。

SCL01 引脚复用作 P41/SOB0 引脚。通过 PMC4，PFC4 和 PF4 寄存器选择有效引脚功能。

PMC41 位	PFC41 位	PF41 位	有效引脚功能
0	无需理会	1	P41 （输出端口模式，N 沟道漏极开路输出）
1	0	1	SOB0 输出 （N 沟道漏极开路输出）
	1	1	SCL01 输入/输出 （N 沟道漏极开路输出）

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

设置顺序	设置内容	引脚状态	引脚电平
<1>	初始值 (PMC41 位 = 0, PFC41 位 = 0, PF41 位 = 0)	端口模式 (输入)	高阻
<2>	PMC41 位 ← 1	SOB0 输出	低电平 (根据 CSIB0 的设置为高电平)
<3>	PFC41 位 ← 1	SCL01 I/O	高电平 (CMOS 输出)
<4>	PF41 位 ← 1	SCL01 I/O	高阻 (N 沟道漏极开路输出)

在步骤<2>中，因为复用功能SOB0 的信号是输出至该引脚，所以I²C通信可能会受影响。在步骤<2> 或 <3>的CMOS输出期间，有可能产生不必要的电流。

(b) 复用功能模式（输入）的注意事项

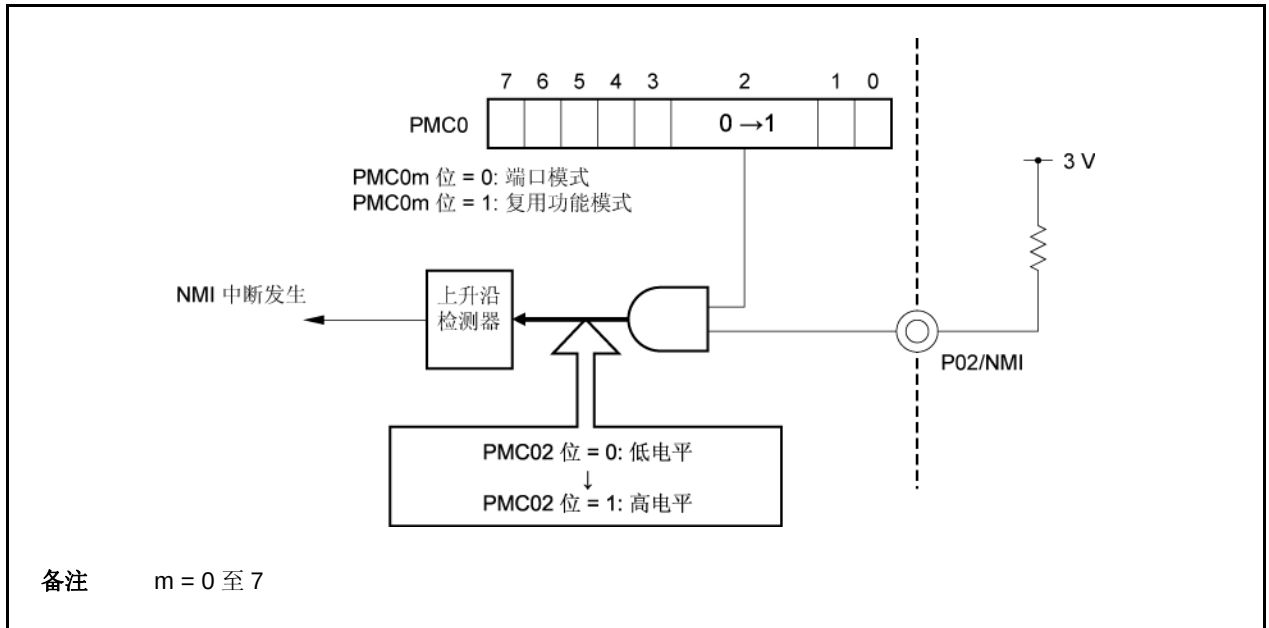
当 PMCn 寄存器的设定值和该引脚的电平进行与（AND）输出，使得 PMCn.PMCnm 位为 0 时，输入到复用功能模块的信号为低电平。因此，根据端口设置和复用功能的操作使能时序，可能会发生不可预料的操作。所以从端口模式转换至复用功能模式的操作按照以下顺序进行。

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

下面的例 1 和例 2 是具体举例。

[例 1] 从通用端口（P02）转换至外部中断引脚（NMI）
P02/NMI 引脚上拉如图 4-37，且上升沿指定为 NMI 引脚边沿检测设置时，在 P02 引脚转换至 NMI 引脚（PMC02 位 = 0 → 1）期间，即使高电平连续输入至 NMI 引脚，也会被检测认为是上升沿，就像低电平跳变到高电平那样，产生 NMI 中断。
为避免该情况，从 P02 引脚转换至 NMI 功能完成后，再设置 NMI 引脚的有效检测沿。

图 4-37. 从 P02 至 NMI 转换举例（不正确）



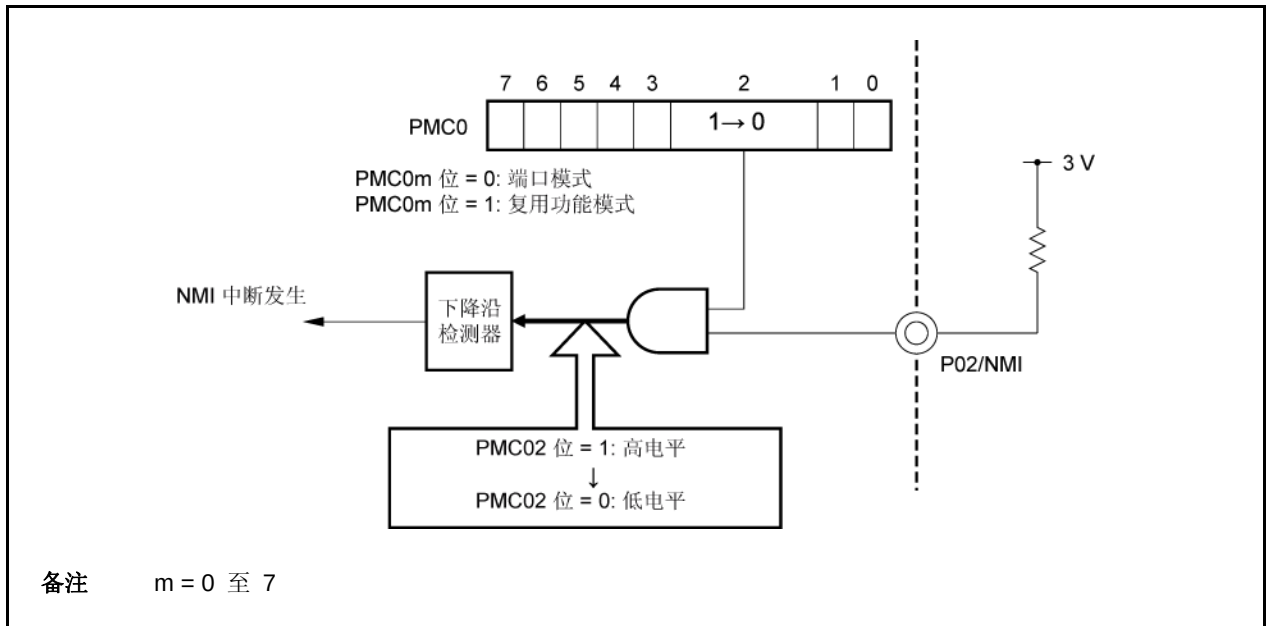
[例 2]

从外部引脚（NMI）转换至通用端口（P02）

P02/NMI 引脚上拉如图 4-38，且上升沿被指定为 NMI 引脚边沿检测设置时，在 NMI 引脚转换至 P02 引脚（PMCO2 位 = 1 → 0）期间，即使高电平连续输入至 NMI 引脚，也会被检测认为是下降沿，就像高电平跳变到低电平那样，产生 NMI 中断。

为避免此情况，在转换至 P02 引脚完成之前，设置 NMI 引脚边沿为“无边沿检测”。

图 4-38. 从 NMI 至 P02 转换举例（不正确）



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

4.6.2 端口 n 寄存器（Pn）位操作指令的注意事项

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

所以，当一个端口由输入模式转变为输出模式时，建议重写输出锁存器。

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

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

V850ES/JJ3 单片机按以下顺序执行位处理指令：

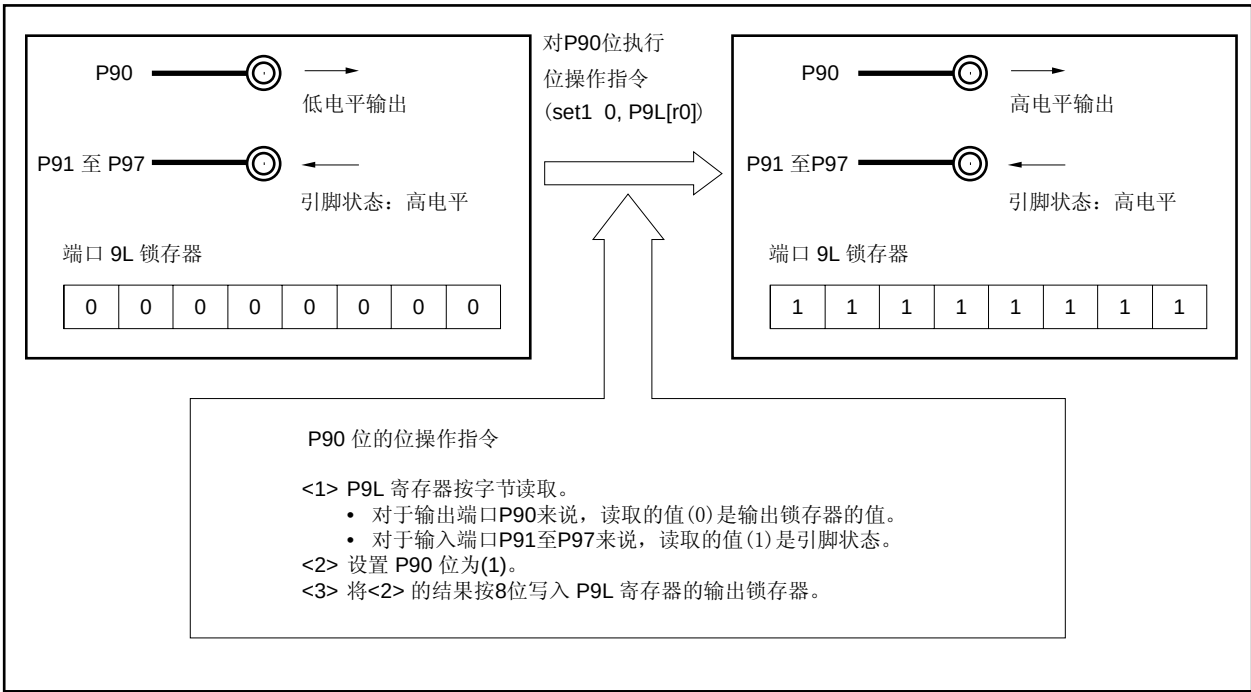
- <1> 寄存器 Pn 按 8 位读取。
- <2> 操作目标位。
- <3> 按 8 位写入寄存器 Pn。

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

在步骤 <2>时，位操作过程使该值变为 FFH。

在步骤 <3>时，位操作将 FFH 写入输出锁存器。

图 4-39. 位处理指令（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 引脚是片上调试引脚。

由 $\overline{\text{RESET}}$ 引脚引发的复位之后, P05/INTP2/ $\overline{\text{DRST}}$ 引脚被初始化作为片上调试引脚($\overline{\text{DRST}}$)。如果此时有高电平输入 $\overline{\text{DRST}}$ 引脚, 则进入片上调试功能, 就可以使用 DCK, DMS, DDI 和 DDO 引脚。

如果不使用片上调试功能, 必须进行以下操作。

- 将 OCDM 寄存器 (特殊寄存器) 的 OCDM0 位清 0。

此时, 从 $\overline{\text{RESET}}$ 引脚的复位解除直到上述操作完成, 应使 P05/INTP2/ $\overline{\text{DRST}}$ 引脚固定为低电平。

如果上述动作之前有高电平输入 $\overline{\text{DRST}}$ 引脚, 将会引起一个错误操作 (CPU 死锁)。操作 P05 引脚时必须格外小心。

注意事项 其它复位情况, 比如 WDT2RES 信号, 时钟监视器 (CLM), 或低压检测器 (LVI) 引发复位之后, P05/INTP2/ $\overline{\text{DRST}}$ 引脚并不会初始化作为片上调试功能引脚 ($\overline{\text{DRST}}$)。OCDM 寄存器保持当前值。

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

P05/INTP2/ $\overline{\text{DRST}}$ 引脚具有内部下拉电阻 (典型值 30 k Ω)。在 $\overline{\text{RESET}}$ 引脚引发复位之后, 下拉电阻被连接。当 OCDM0 位清除为 0 时, 下拉电阻被断开。

4.6.5 电源开启时 P53 引脚的注意事项

电源开启后的复位期间, 下列引脚可能暂时输出不确定电平。

- P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 引脚

4.6.6 迟滞特性

在端口模式下, 以下端口引脚没有迟滞特性。

P00 至 P06
 P31 至 P35, P38, P39
 P40 至 P42
 P50 至 P55
 P66, P68 至 P610, P612, P613
 P80
 P90 至 P97, P99, P910, P912 至 P915

第五章 总线控制功能

V850ES/JJ3 提供外部总线接口，通过该接口可以连接外部存储器设备，诸如 ROM、RAM 和 I/O 设备等。

5.1 特性

- 可以选择最小 3 个总线周期的复用总线输出，或最小 2 个总线周期的分离总线输出
- 8 位/16 位数据总线可选
- 等待功能：
 - 最高可达 7 个周期的可编程等待功能
 - 使用 $\overline{\text{WAIT}}$ 引脚的外部等待功能
- 空闲状态功能
- 总线保持功能
- 可连接最大 16 MB 的物理存储器（内部 ROM 空间为 1 MB）

5.2 总线控制引脚

用于连接外部设备的引脚列举如下表：

表 5-1. 总线控制引脚（复用总线）

总线控制引脚	复用功能引脚	输入/输出	功能
AD0 至 AD15	PDL0 至 PDL15	I/O	地址/数据总线
A16 至 A23	PDH0 至 PDH7	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	写选通信号
RD	PCT4	输出	读选通信号
ASTB	PCT6	输出	地址选通信号
HLD $\overline{RQ}$	PCM3	输入	总线保持控制
HLD $\overline{AK}$	PCM2	输出	
CS0 至 CS3	PCS0 至 PCS3	输出	片选

表 5-2. 外部控制引脚（分离总线）

总线控制引脚	复用功能引脚	输入/输出	功能
AD0 至 AD15	PDL0 至 PDL15	I/O	数据总线
A0 至 A15	P90 至 P915	输出	地址总线
A16 至 A23	PDH0 至 PDH7	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	写选通信号
RD	PCT4	输出	读选通信号
HLD $\overline{RQ}$	PCM3	输入	总线保持控制
HLD $\overline{AK}$	PCM2	输出	
CS0 至 CS3	PCS0 至 PCS3	输出	片选

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

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

表 5-3. 当访问片内 ROM 存储器、片内 RAM 存储器或片上外围输入/输出端口时的引脚状态

分离总线模式		复用总线模式	
地址总线 (A23 至 A0)	不确定	地址总线 (A23 至 A16)	不确定
数据总线 (AD15 至 AD0)	高阻	地址/数据总线 (AD15 至 AD0)	不确定
控制信号	无效	控制信号	无效

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

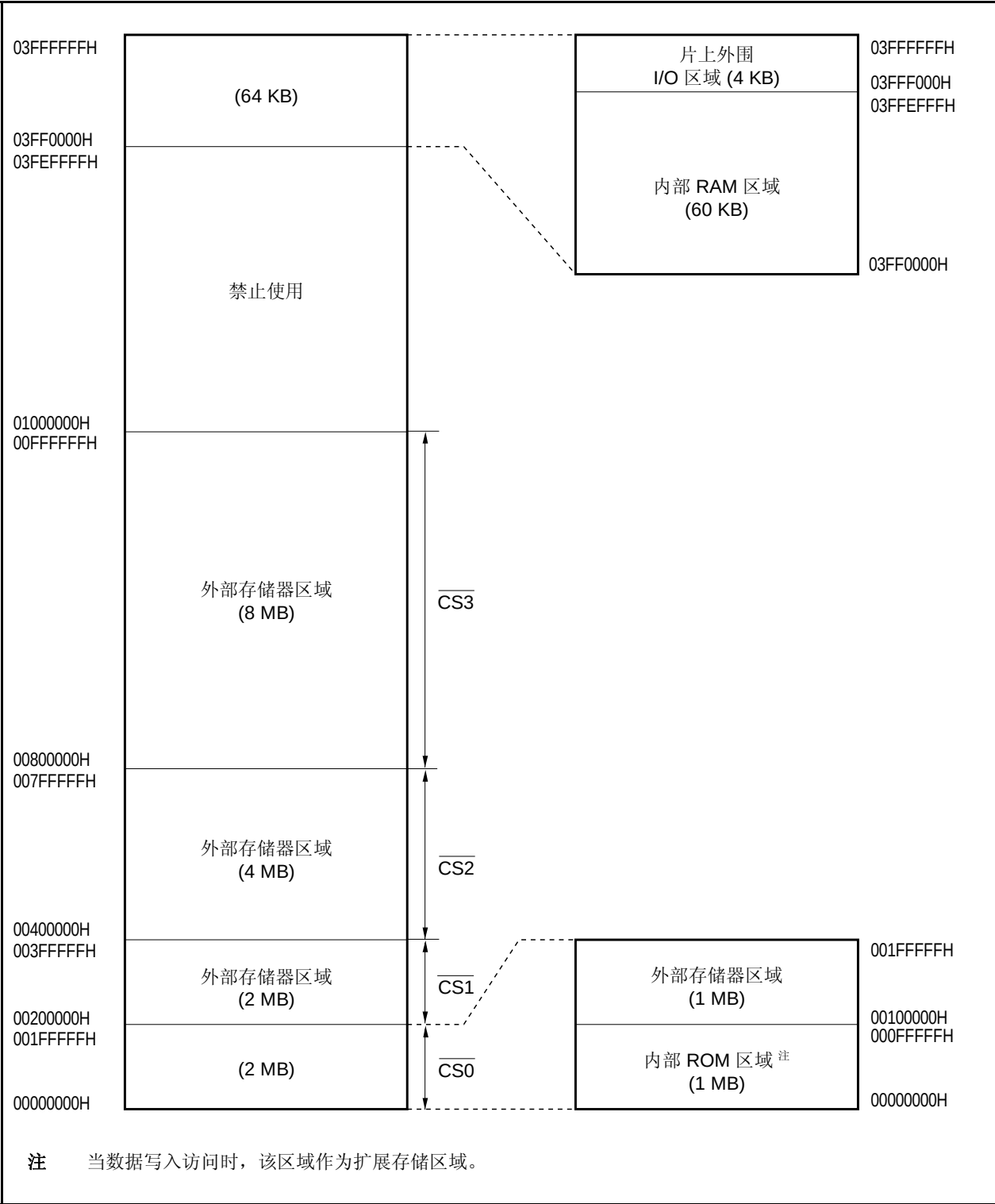
5.2.2 各种工作模式下的引脚状态

关于 V850ES/JJ3 单片机在各种工作模式下的引脚状态，参见 2.2 引脚状态。

5.3 存储模块功能

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

图 5-1. 数据存储映射：物理存储空间



5.4 外部总线接口模式控制功能

V850ES/JJ3 单片机有如下两种外部总线接口模式：

- 复用总线模式
- 分离总线模式

可以通过 EXIMC 寄存器来选择这两种模式。

(1) 外部总线接口模式控制寄存器 (EXIMC)

EXIMC 寄存器可以按字节或按位进行读取或写入。
系统复位后，该寄存器置为 00H。

复位后：00H R/W 地址：FFFFFFBEH								
EXIMC	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	SMSEL

SMSEL	模式选择
0	复用总线模式
1	分离总线模式

注意事项 在外部访问之前，从内部 ROM 或内部 RAM 设置 EXIMC 寄存器。
在设置 EXIMC 寄存器之后，请确保插入一条 NOP 指令。

5.5 总线访问

5.5.1 总线访问时钟数目

访问各个存储资源所需的基本时钟数目如下表：

区域（总线宽度） 总线周期类型	内部 ROM（32 位）	内部 RAM（32 位）	外部存储器（16 位）
取指周期（常规访问）	1	1^{n1}	$3 + n^{n2}$
取指周期（分支跳转）	2	2^{n1}	$3 + n^{n2}$
操作数访问	3	1	$3 + n^{n2}$

- 注
1. 如果与数据访问发生冲突则加 1。
 2. 当选择分离总线模式时， $2 + n$ 个时钟周期（n：等待状态的数目）

备注 单位：时钟数/访问

5.5.2 总线宽度设置功能

由 $\overline{CSn}$ 选择的每一个外部存储区都可以通过 BSC 寄存器设置。然而，总线宽度只可设置为 8 位和 16 位。
V850ES/JJ3 单片机的外部存储区由 $\overline{CS0}$ 至 $\overline{CS3}$ 选择。

（1）总线宽度设置寄存器（BSC）

BSC 寄存器可以按 16 位宽度进行读写。

系统复位后，该寄存器的值为 5555H。

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

复位后：5555H R/W 地址：FFFFF066H

	15	14	13	12	11	10	9	8
BSC	0	1	0	1	0	1	0	1
	7	6	5	4	3	2	1	0
	0	BS30	0	BS20	0	1	0	BS00
		☐		☐				☐
		CS3		CS2				CS0
BSn0	存储模块 CSn 空间的数据总线宽度 (n = 0, 2, 3)							
0	8 位							
1	16 位							

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

5.5.3 根据不同总线宽度进行访问

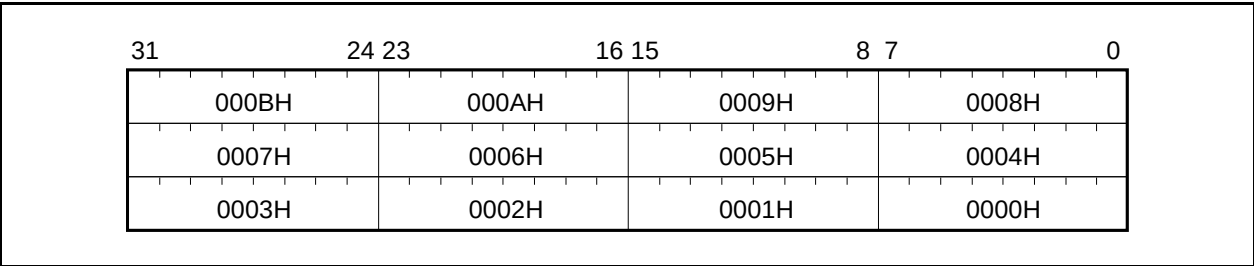
V850ES/JJ3 单片机按照 8 位、16 位或 32 位总线宽度访问片上外围输入/输出端口和外部存储器。总线宽度说明如下：

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

以上各种访问时的操作描述如下。所有数据访问都从低位开始。

V850ES/JJ3 单片机仅支持“小端”数据格式。

图 5-2. 字中的“小端”地址



(1) 数据空间

V850ES/JJ3 单片机具有地址不对齐功能。

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

(a) 访问半字长度数据（16 位）

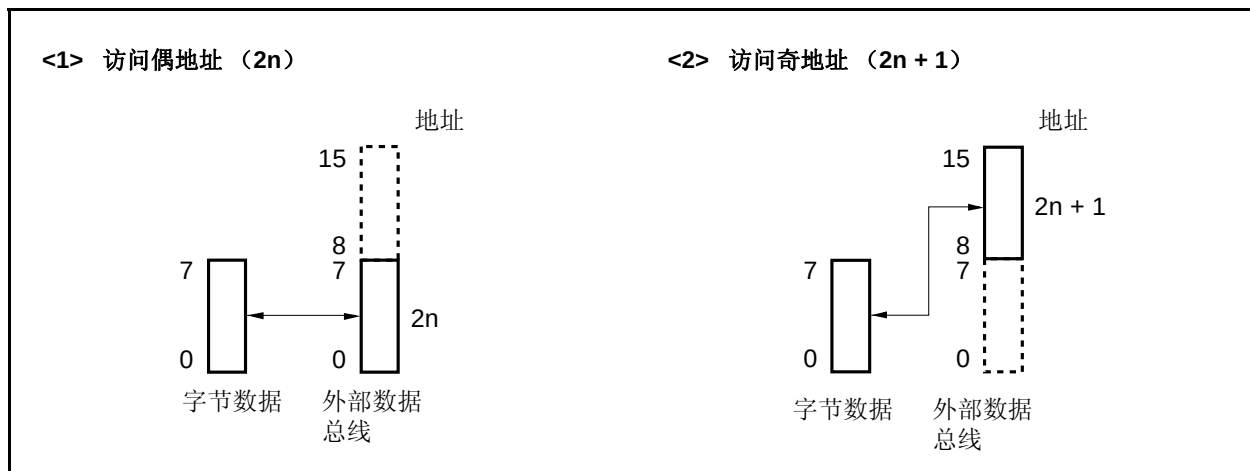
如果地址的最低有效位为 1，一个字节长度的总线访问周期需要产生两次。

(b) 访问字长度数据（32 位）

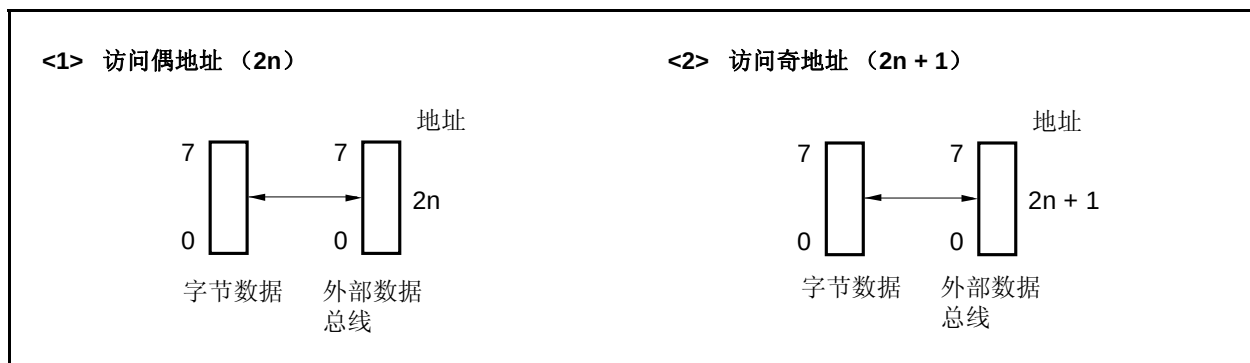
- (i) 如果地址的最低有效位为 1，所需总线访问周期次序：一个字节访问周期、一个半字访问周期、一个字节访问周期。
- (ii) 如果地址的最低两个有效位为 10，需要产生两次半字长度的总线访问周期。

(2) 字节访问 (8 位)

(a) 16 位数据总线宽度

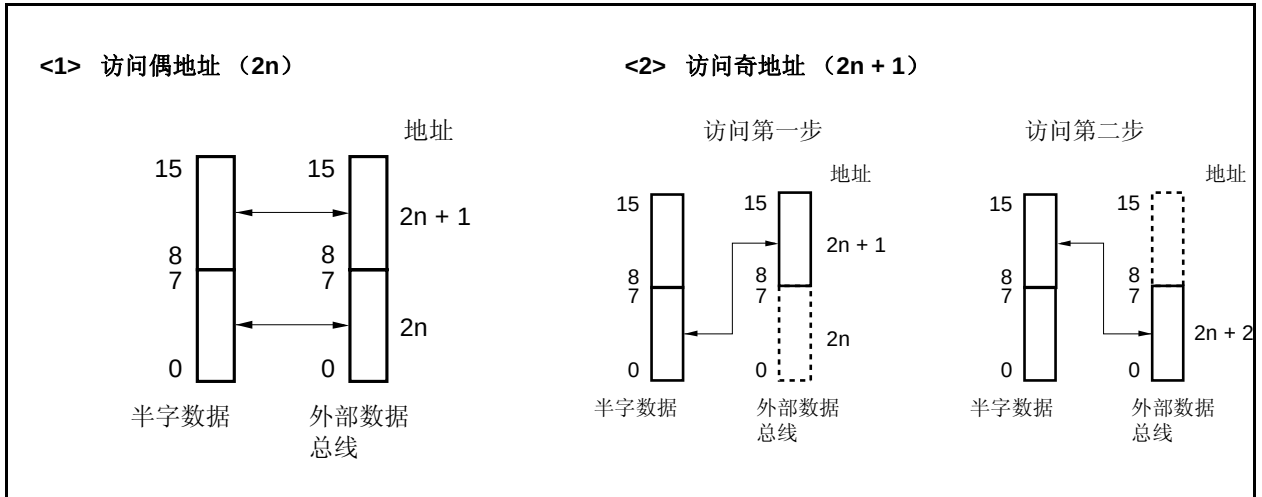


(b) 8 位数据总线宽度

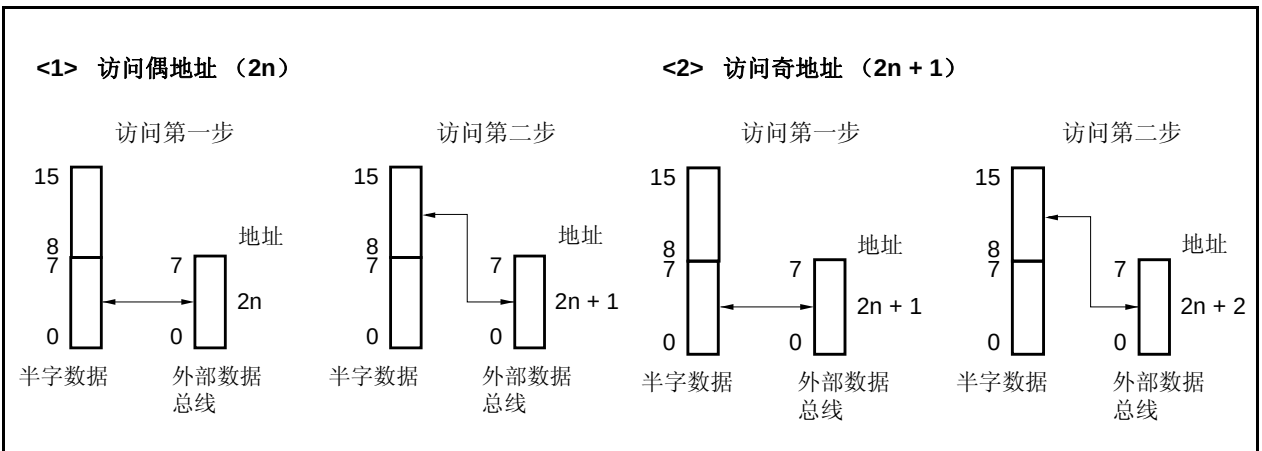


(3) 半字访问 (16 位)

(a) 16 位数据总线宽度

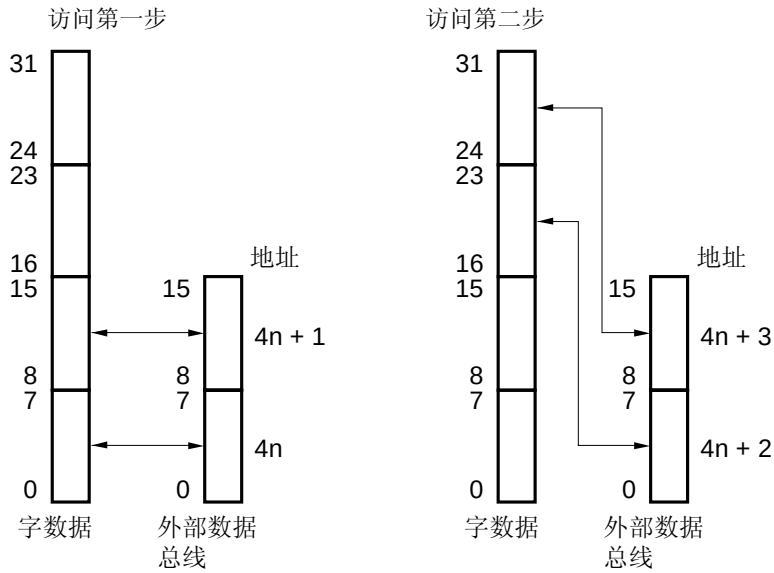
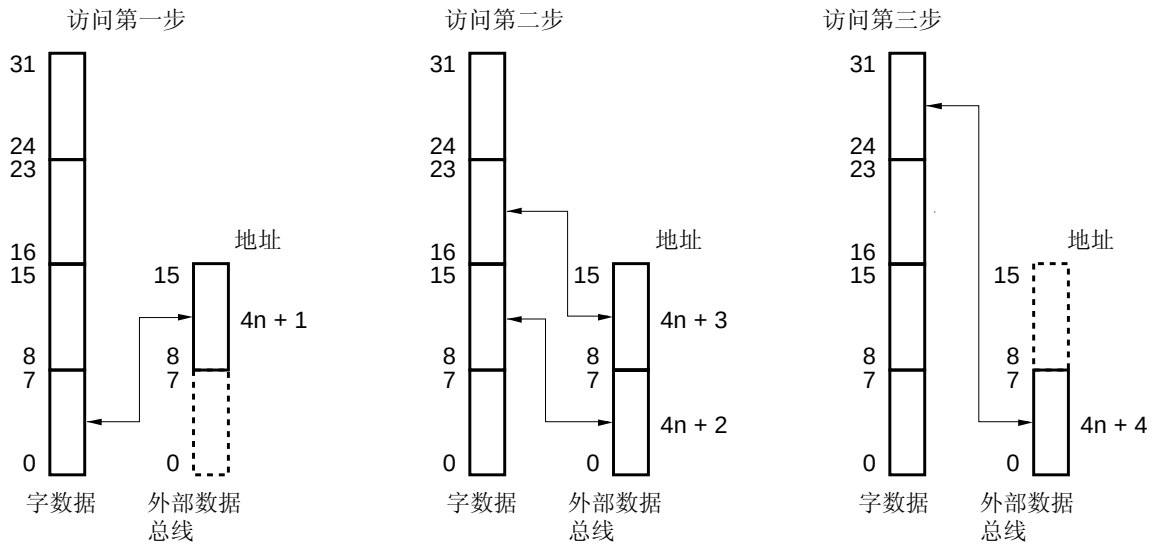


(b) 8 位数据总线宽度



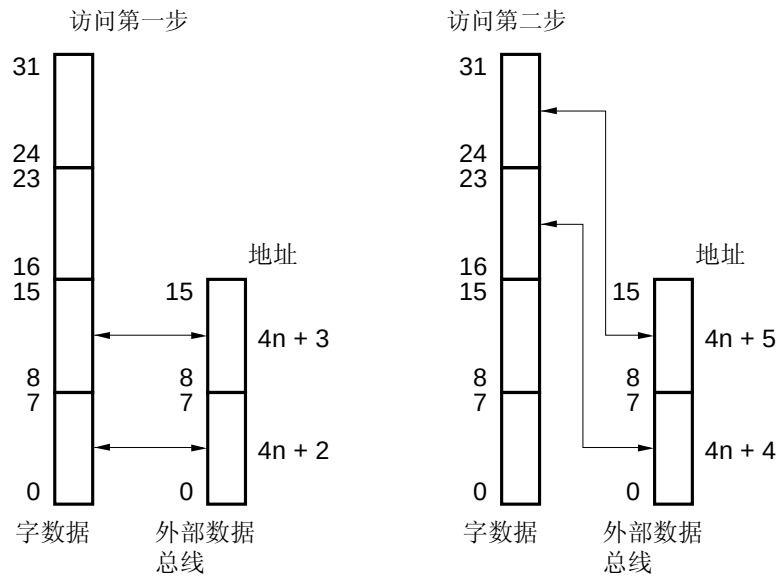
(4) 字访问 (32 位)

(a) 16 位数据总线宽度 (1/2)

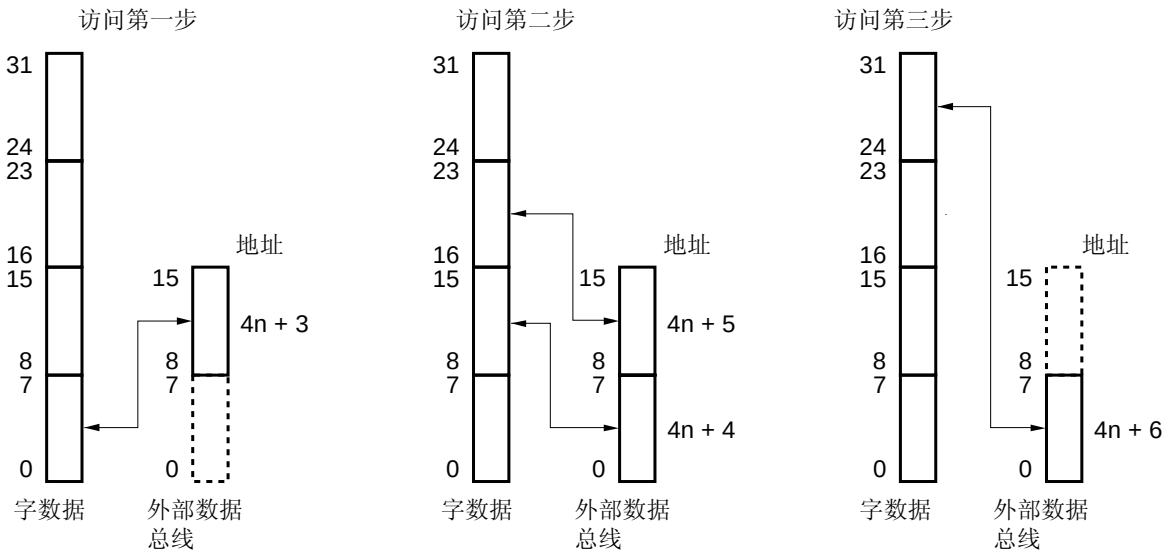
<1> 访问地址 ($4n$)<2> 访问地址 ($4n + 1$)

(a) 16 位数据总线宽度 (2/2)

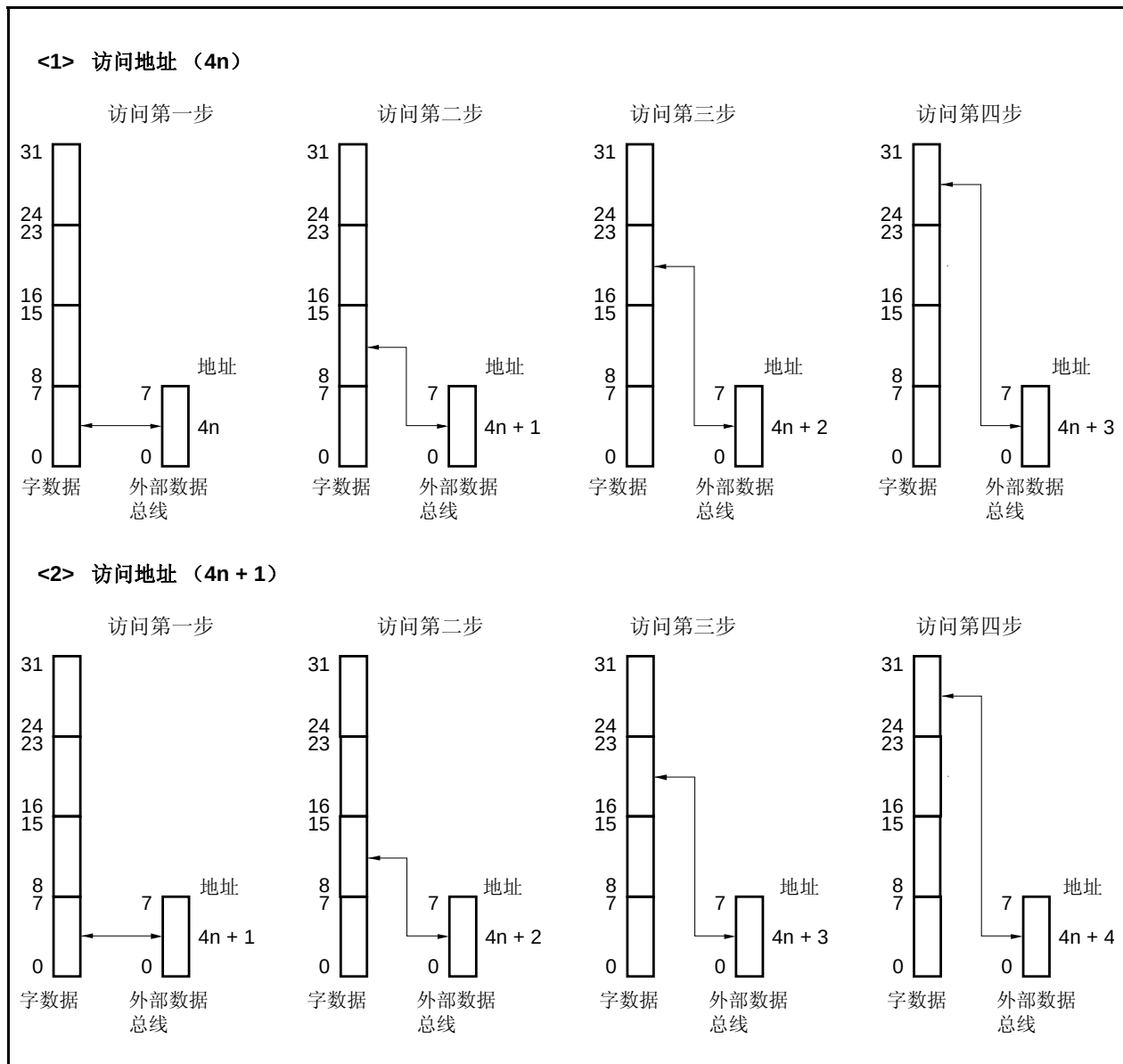
<3> 访问地址 ($4n + 2$)



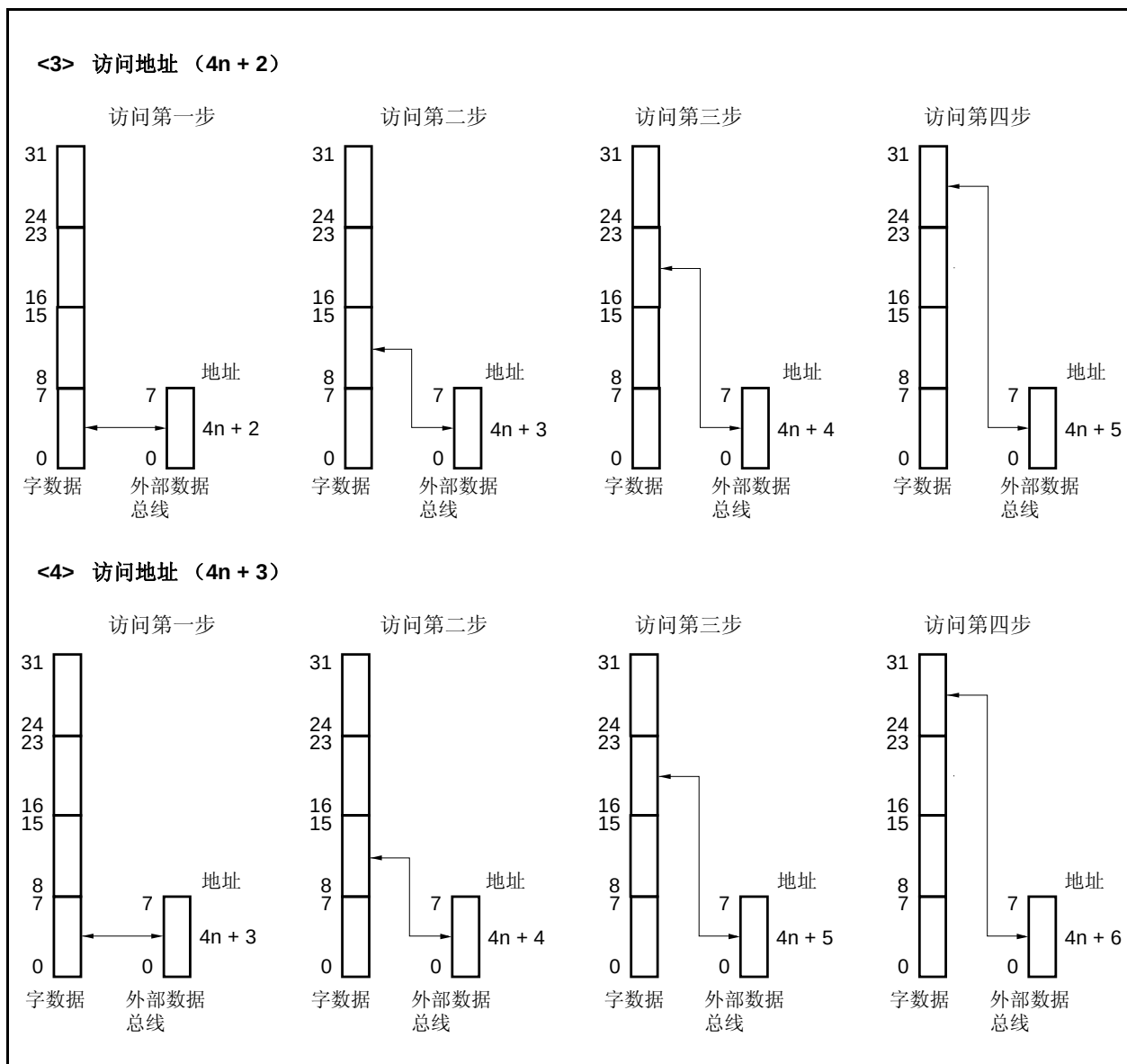
<4> 访问地址 ($4n + 3$)



(b) 8 位数据总线宽度 (1/2)



(b) 8 位数据总线宽度 (2/2)



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)

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

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

DWC0 寄存器可以按 16 位宽度进行读写。

系统复位后, DWC0 寄存器置为 7777H。

- 注意事项**
1. 对内部 ROM 和内部 RAM 的访问不需要可编程等待, 总是可以无等待访问。片上外围输入/输出端口也不需要可编程等待, 而是根据各自外围输入/输出端口的控制功能执行等待。
 2. 系统复位完成后写入 DWC0 寄存器, 此后不要再改变 DWC0 寄存器的值。而且, DWC0 寄存器设置完成之前, 不要访问外部存储空间。
 3. 当 V850ES/JJ3 用于分离总线模式, 且运行于 $f_{xx} > 20 \text{ MHz}$, 请确保插入一个或更多的等待。

复位后: 7777H		R/W		地址: FFFFF484H					
		15	14	13	12	11	10	9	8
DWC0	0	DW32	DW31	DW30	0	DW22	DW21	DW20	
	CS3				CS2				
		7	6	5	4	3	2	1	0
	0	DW12	DW11	DW10	0	DW02	DW01	DW00	
	CS1				CS0				
	DWn2	DWn1	DWn0	在CSn 空间 插入等待状态的数目 (n = 0 至 3)					
				复用总线	分离总线				
					f _{xx} ≤20 MHz			f _{xx} > 20 MHz	
	0	0	0	无	无			禁止设置	
	0	0	1	1					
	0	1	0	2					
	0	1	1	3					
	1	0	0	4					
	1	0	1	5					
	1	1	0	6					
	1	1	1	7					

注意事项 请确保将第 15, 11, 7 位和第 3 位设置为“0”。

5.6.2 外部等待功能

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

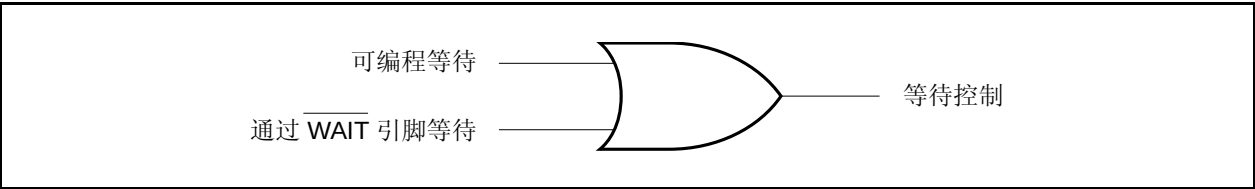
PCMO 引脚设置为复用功能时，使能外部等待功能。

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

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

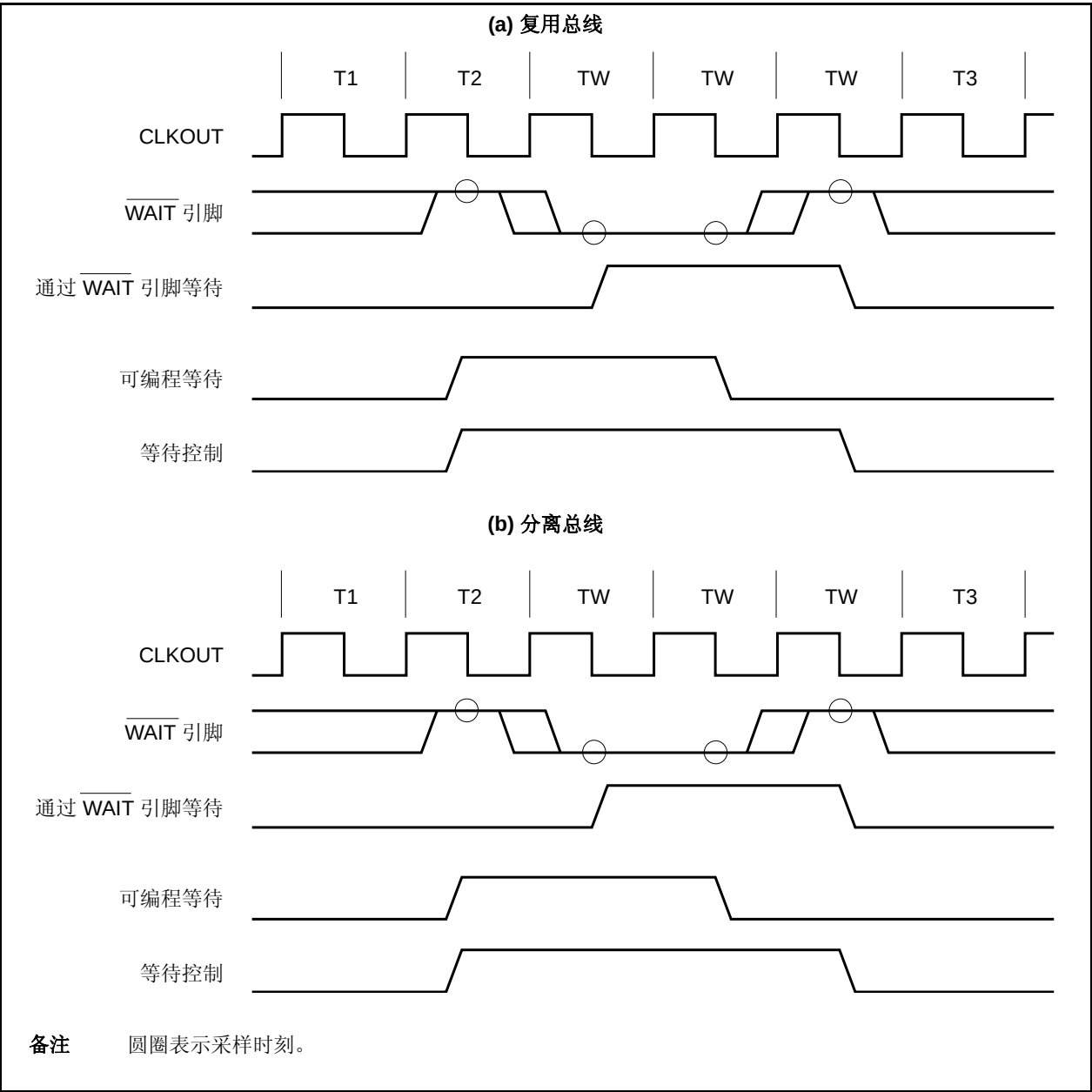
5.6.3 可编程等待和外部等待之间的关系

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



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

图 5-3. 插入等待举例



5.6.4 可编程地址等待功能

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

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

（1）地址等待控制寄存器（AWC）

AWC 寄存器可以按 16 位宽度进行读写。

系统复位后，AWC 寄存器被设置为 FFFFH。

- 注意事项**
1. 访问内部 ROM 区、内部 RAM 区和片上外围输入/输出区时，不需要插入地址建立等待周期和地址保持等待周期。
 2. 系统复位完成后写入 AWC 寄存器，此后不要再改变该设定值。而且，AWC 寄存器初始设置完成之前，不要访问外部存储空间。
 3. 当 V850ES/JJ3 运行于 $f_{xx} > 20 \text{ MHz}$ ，请确保插入地址保持等待和地址建立等待。

复位后: FFFFH

R/W

地址: FFFFF488H

15

14

13

12

11

10

9

8

AWC

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

76543210

AHW3	ASW3	AHW2	ASW2	AHW1	ASW1	AHW0	ASW0
------	------	------	------	------	------	------	------

CS3

CS2

CS1

CS0

AHWn	地址保持等待的插入说明 (n = 0 至 3)	
	$f_{xx} \leq 20\text{ MHz}$	$f_{xx} > 20\text{ MHz}$
0	没有插入	禁止设置
1	插入	插入

ASWn	地址建立等待的插入说明 (n = 0 至 3)	
	$f_{xx} \leq 20\text{ MHz}$	$f_{xx} > 20\text{ MHz}$
0	没有插入	禁止设置
1	插入	插入

注意事项

请确保将第 15 至 8 位设置为“1”。

5.7 空闲状态插入功能

为了方便和低速存储器的连接，对通过片选功能选择的各个空间执行访问时，在复用地址/数据总线模式下，在总线周期的 T3 状态后插入一个空闲状态（TI）。在分离总线模式下，在 T2 状态后插入一个空闲状态（TI）。在读取访问时插入一个空闲状态，可以保证存储器数据输出的浮动延迟时间（写访问期间，不能插入空闲状态）。

通过 BCC 寄存器，可以设置是否插入空闲状态。

系统复位之后，对所有区域的访问时序都会立刻插入一个空闲状态。

(1) 总线周期控制寄存器（BCC）

BCC 寄存器可以按 16 位宽度进行读写。

系统复位后，该寄存器 被置为 AAAAH。

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

复位后：AAAAH R/W 地址：FFFFFF48AH

	15	14	13	12	11	10	9	8
BCC	1	0	1	0	1	0	1	0
	7	6	5	4	3	2	1	0
	BC31	0	BC21	0	BC11	0	BC01	0
	<div></div>		<div></div>		<div></div>		<div></div>	
	CS3		CS2		CS1		CS0	
B _{Cn1}	空闲状态的插入说明 (n = 0 至 3)							
0	没有插入							
1	插入							

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

5.8 总线保持功能

5.8.1 功能概述

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

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

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

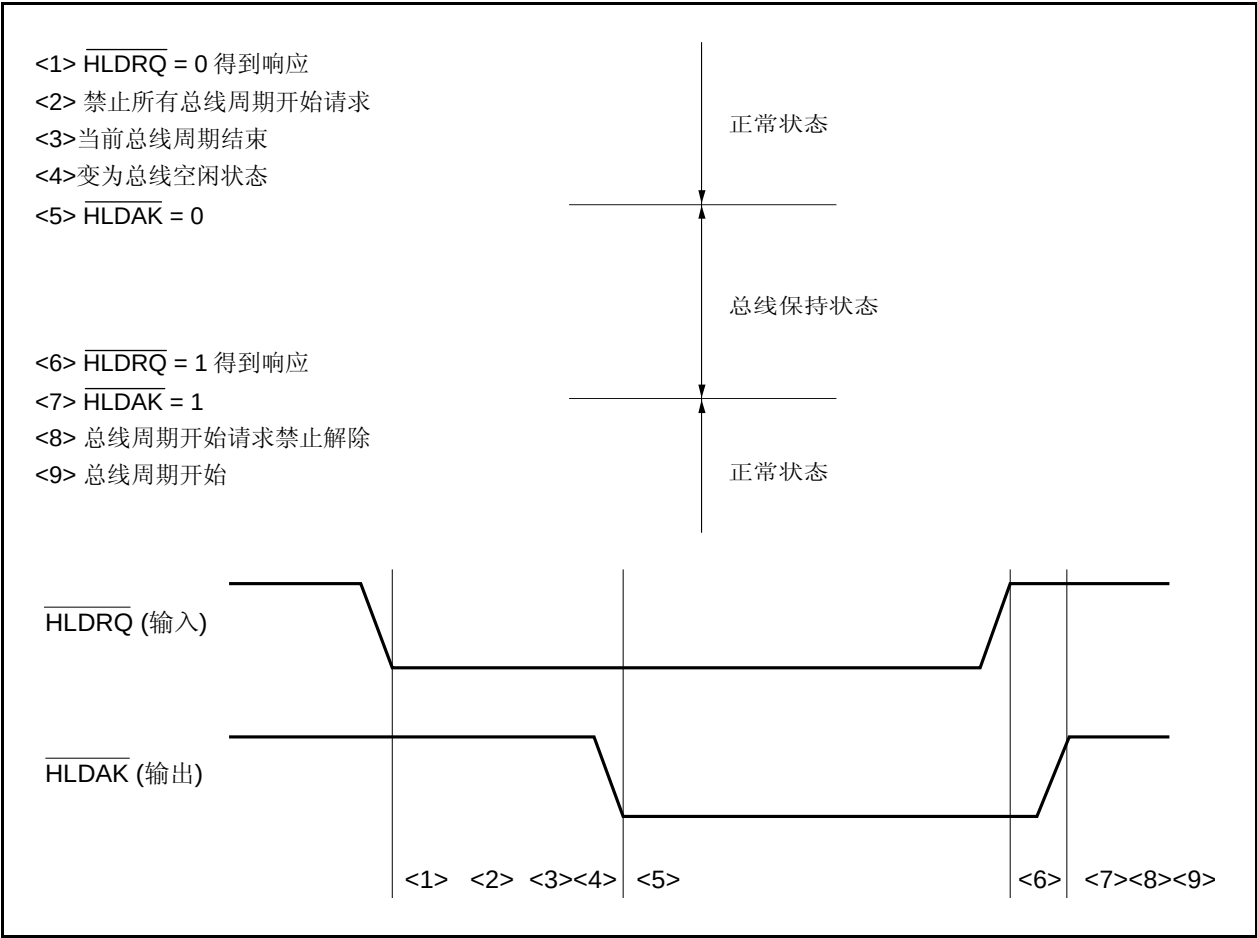
总线保持状态通过 $\overline{\text{HLDK}}$ 引脚的有效（低电平）来表示。总线保持功能允许多处理器系统的配置，其中存在两个或更多个总线主控器。

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

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

5.8.2 总线保持步骤

总线保持状态转变步骤如下所示：



5.8.3 节电模式时的操作

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

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

5.9 总线优先权

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

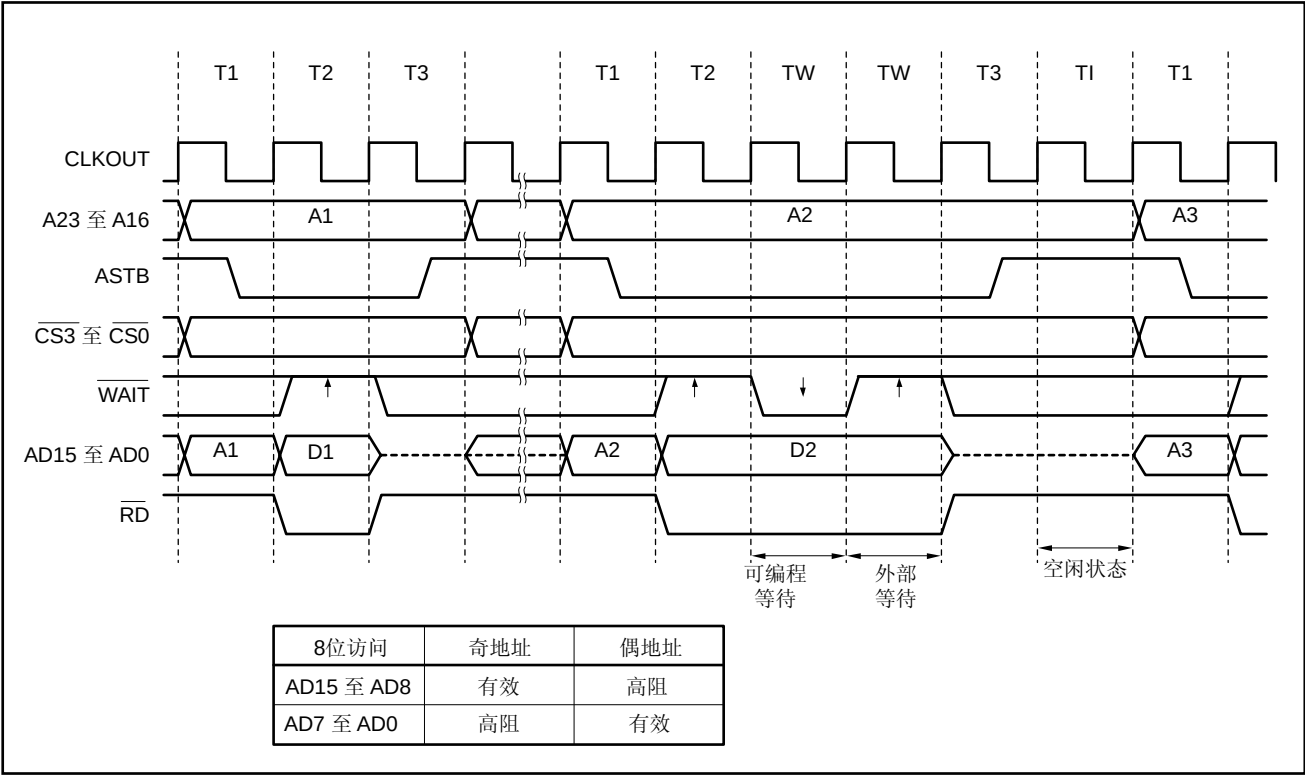
如果一条指令执行需要进行两次或两次以上的数据访问时，由于总线长度的限制，在数据访问期间不插入取指令操作和总线保持状态。

表 5-4. 总线优先级

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

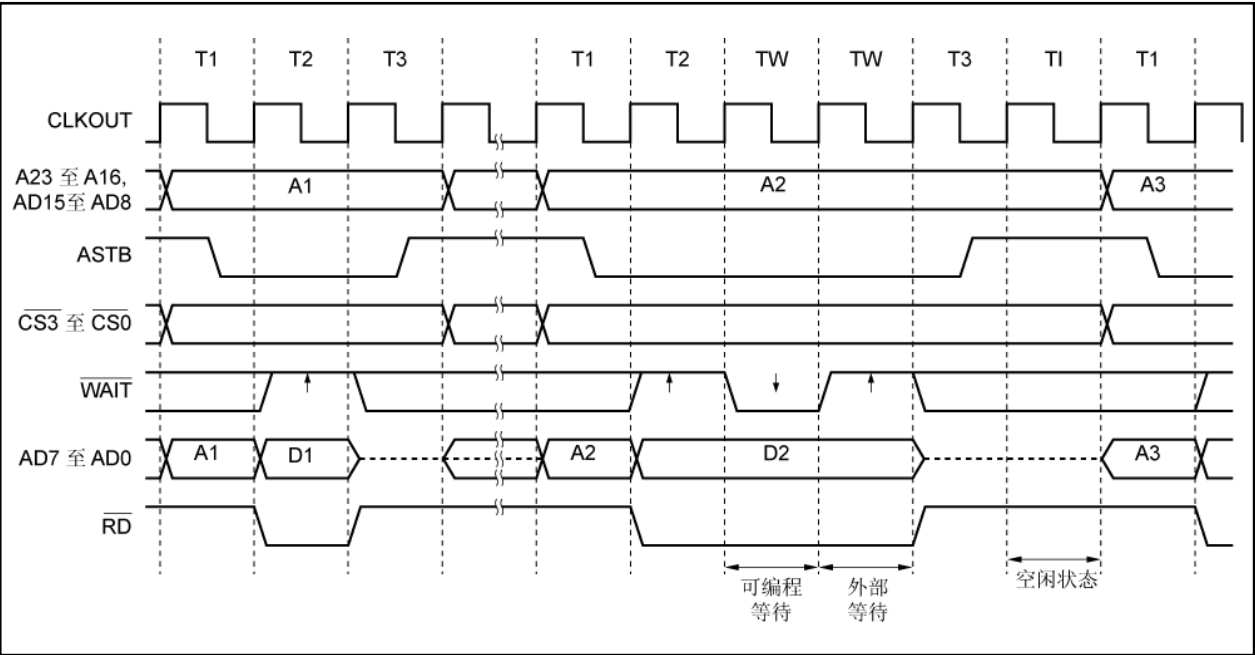
5.10 总线时序

图 5-4. 复用总线读时序（总线宽度：16 位，16 位存取）



备注 虚线表示高阻状态。

图 5-5. 复用总线读时序（总线宽度：8 位）



备注 虚线表示高阻状态。

图 5-6. 复用总线写入时序（总线宽度：16 位，16 位存取）

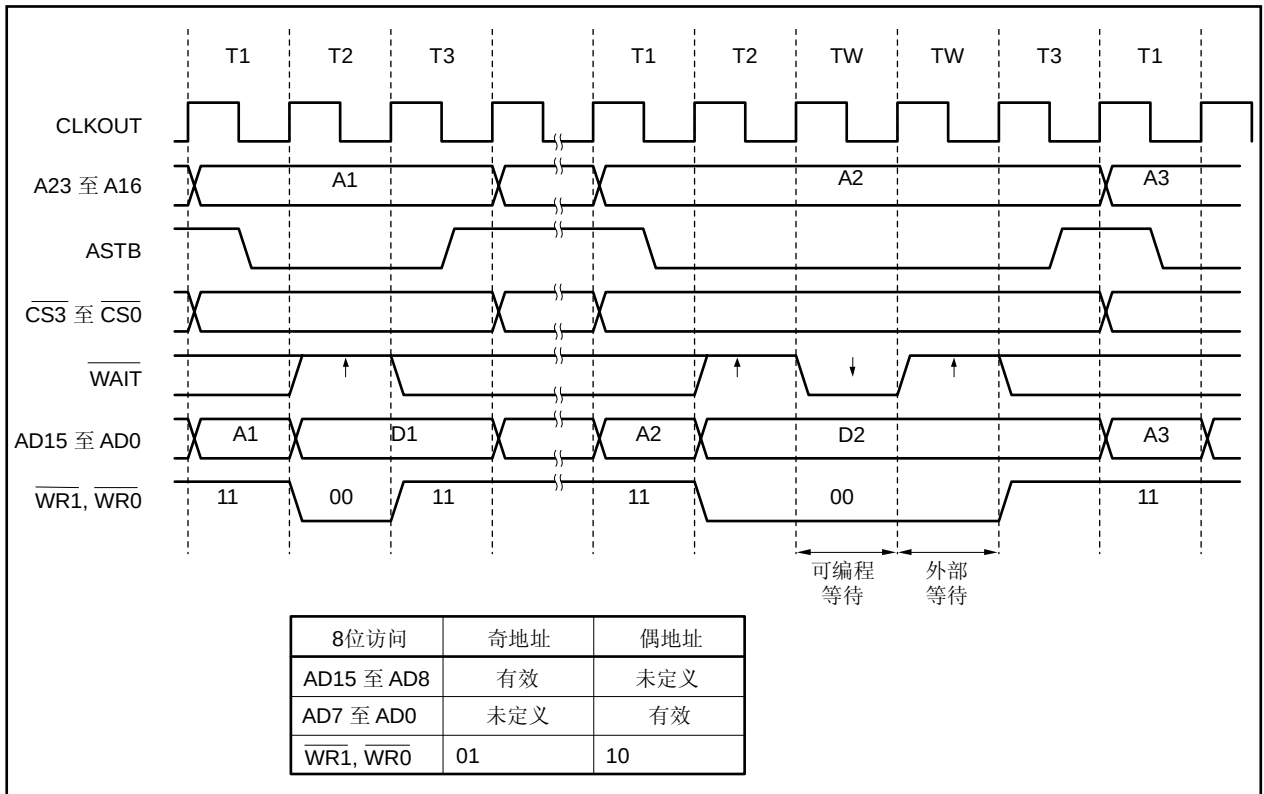


图 5-7. 复用总线写入时序（总线宽度：8 位）

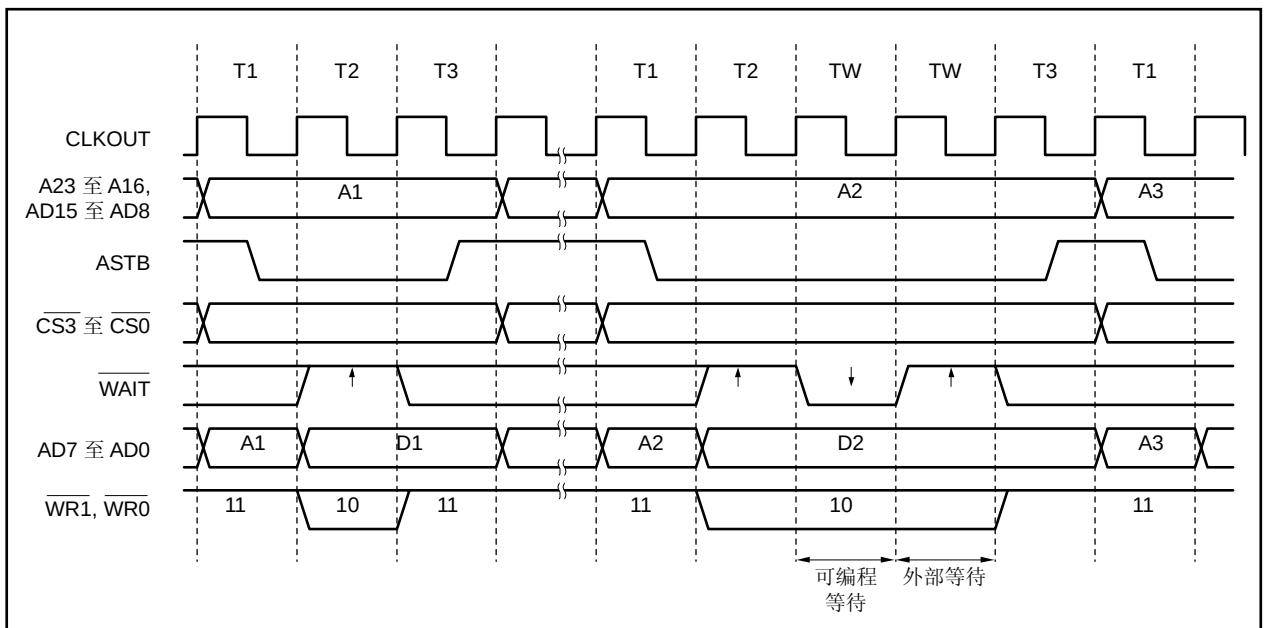


图 5-8. 复用总线保持时序（总线宽度：16 位，16 位存取）

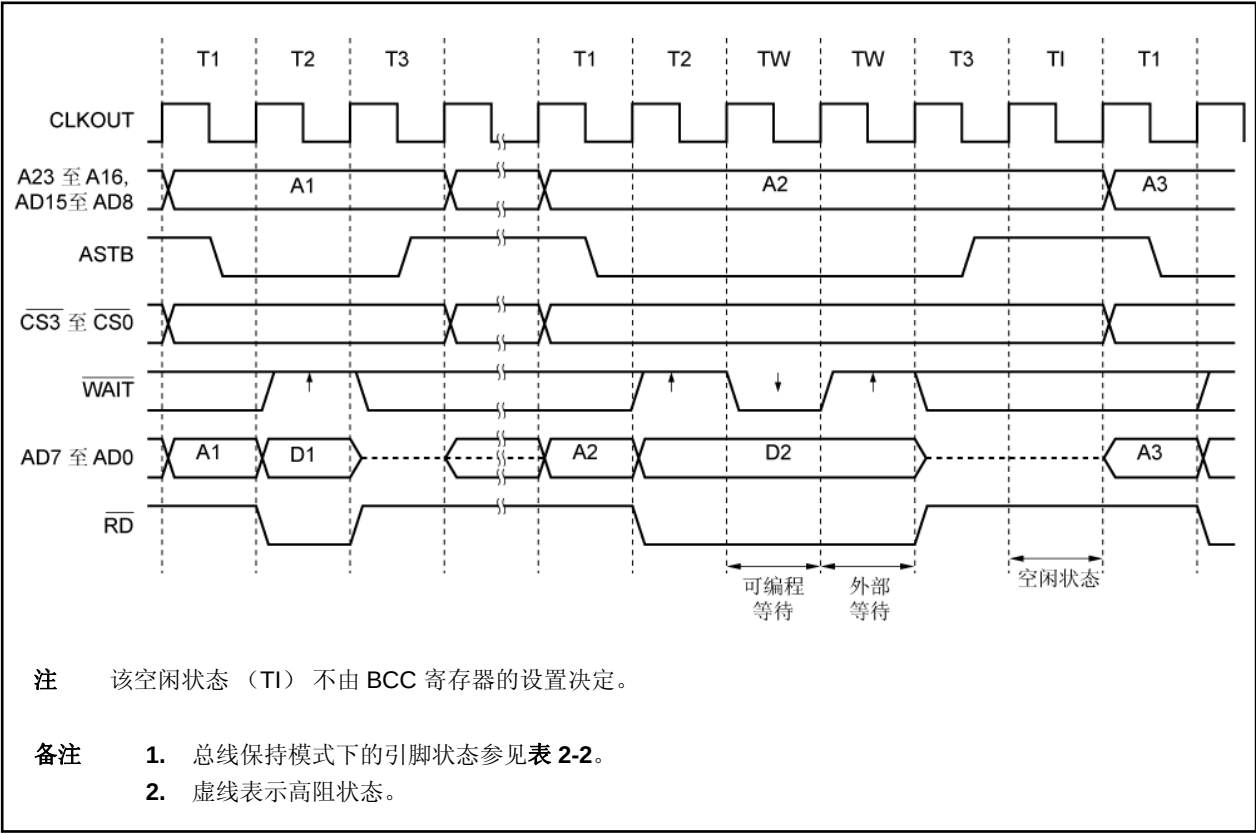
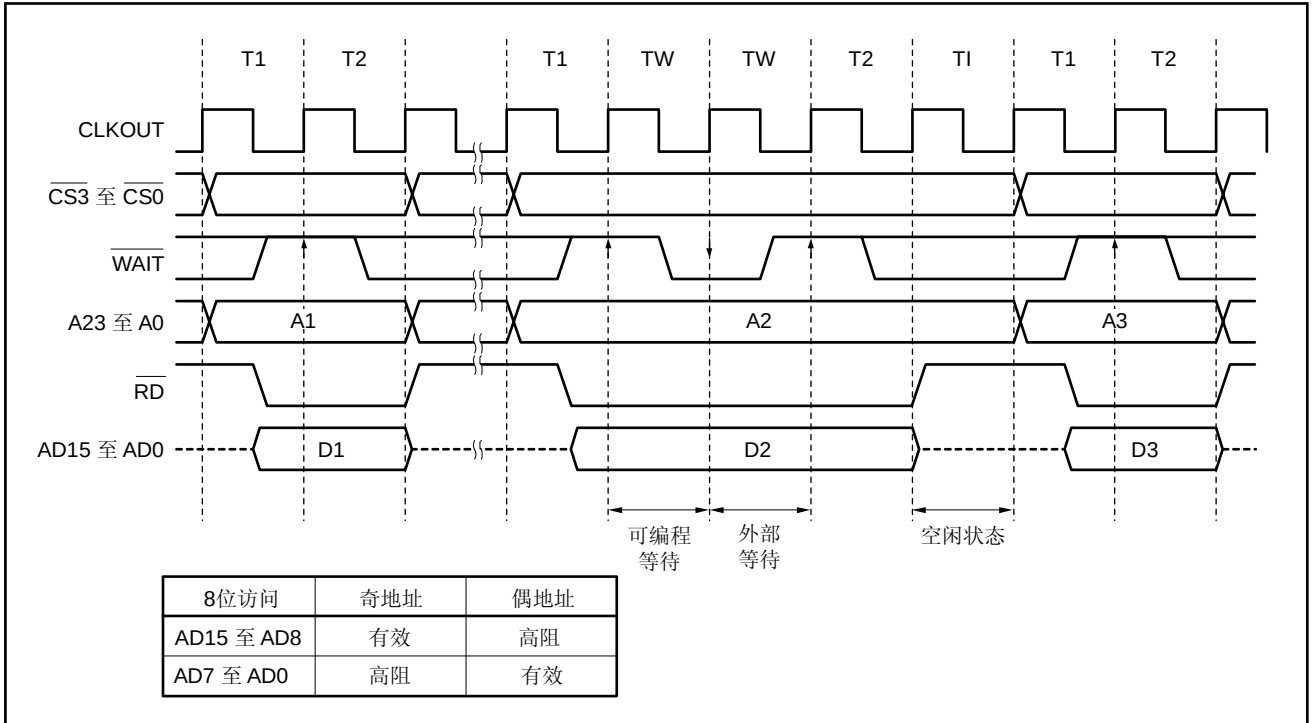
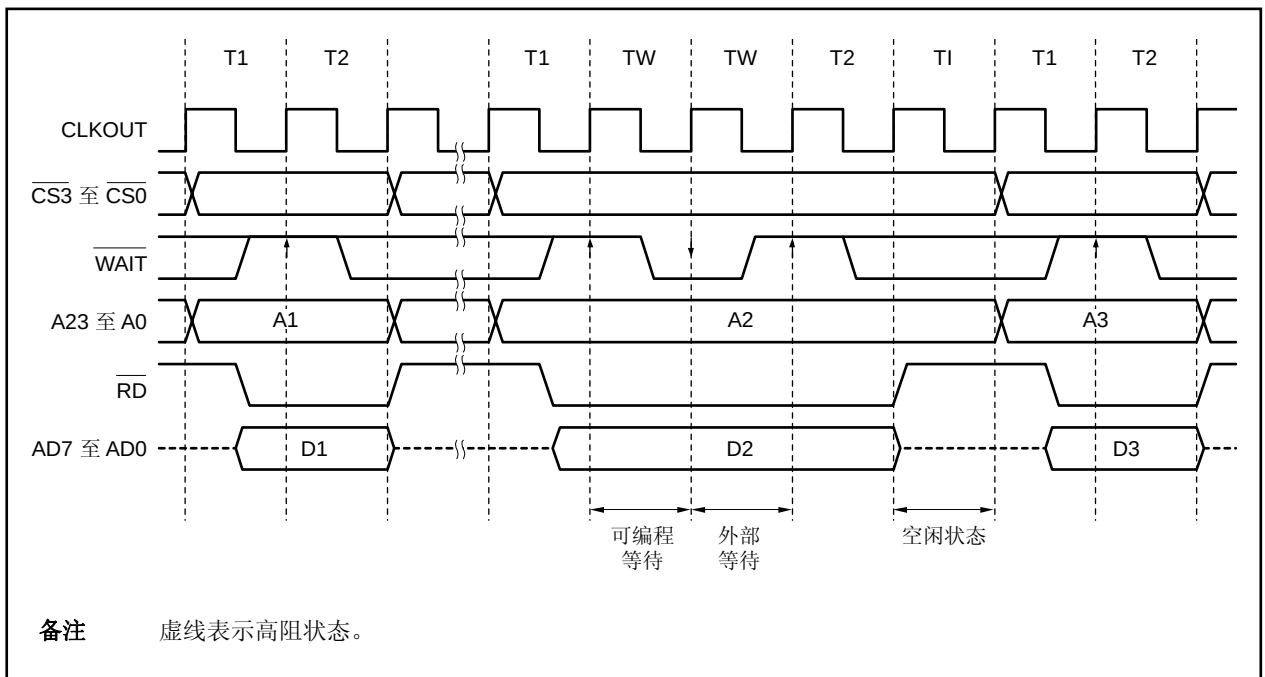


图 5-9. 分离总线读时序（总线宽度：16 位，16 位存取）



备注 虚线表示高阻状态。

图 5-10. 分离总线读时序（总线宽度：8 位）



备注 虚线表示高阻状态。

图 5-11. 分离总线写入时序（总线宽度：16 位，16 位存取）

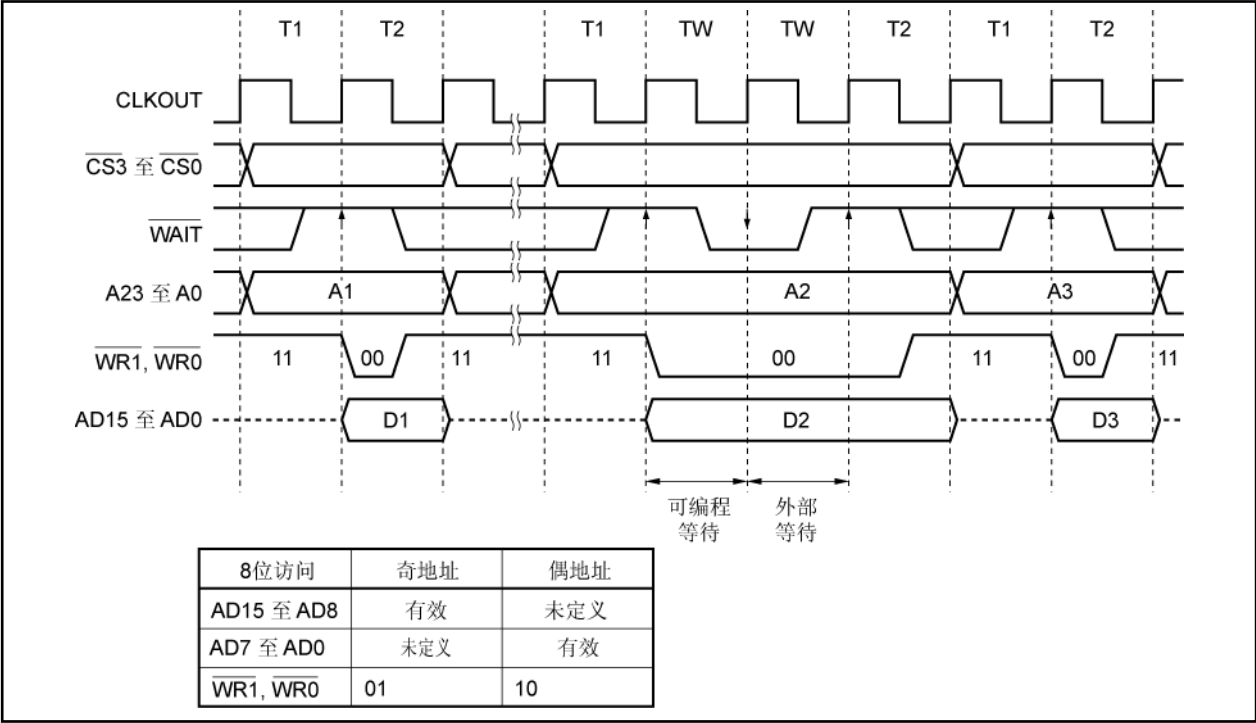


图 5-12. 分离总线写入时序（总线宽度：8 位）

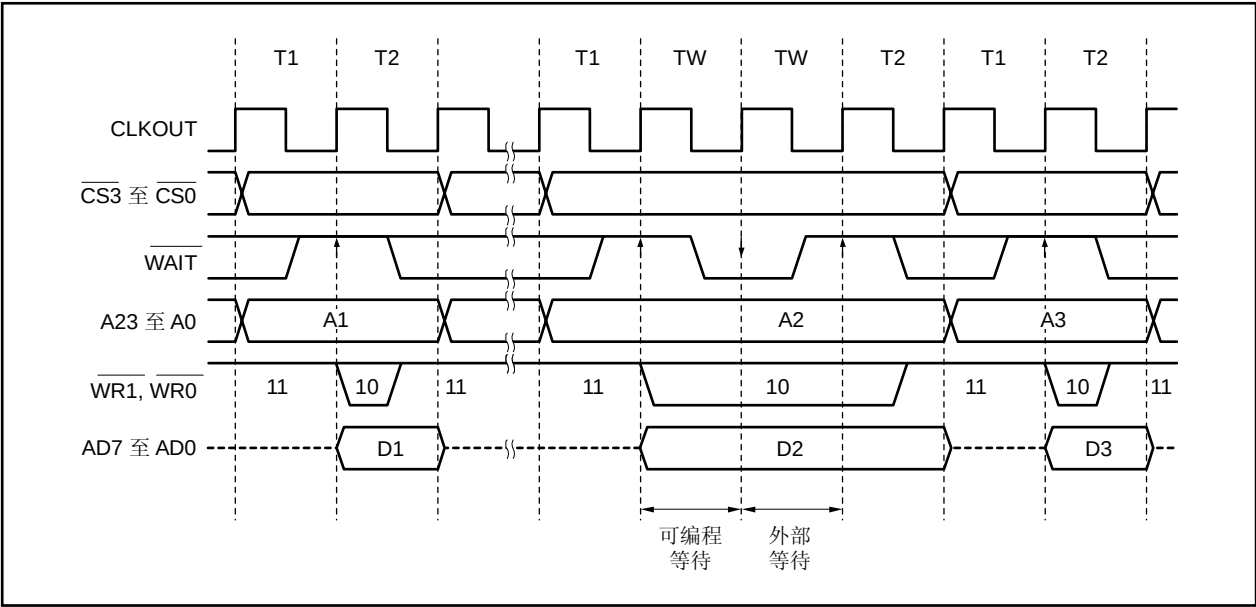


图 5-13. 分离总线保持时序（总线宽度：8 位，写入）

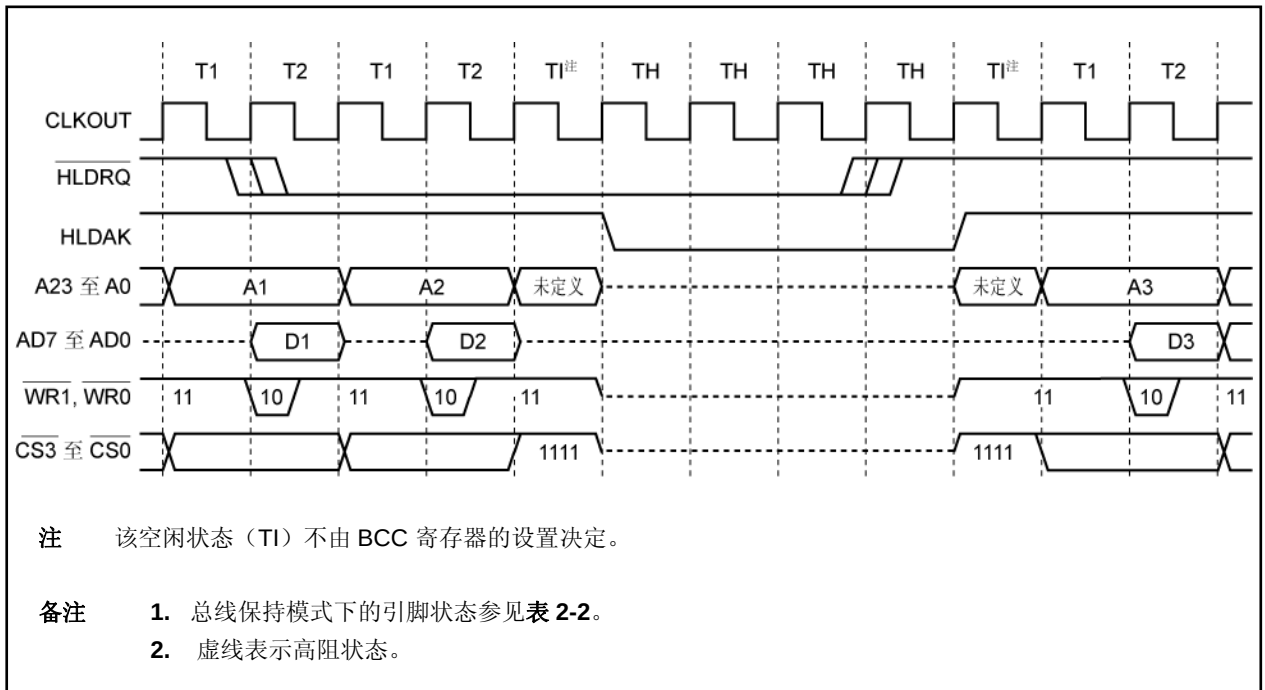
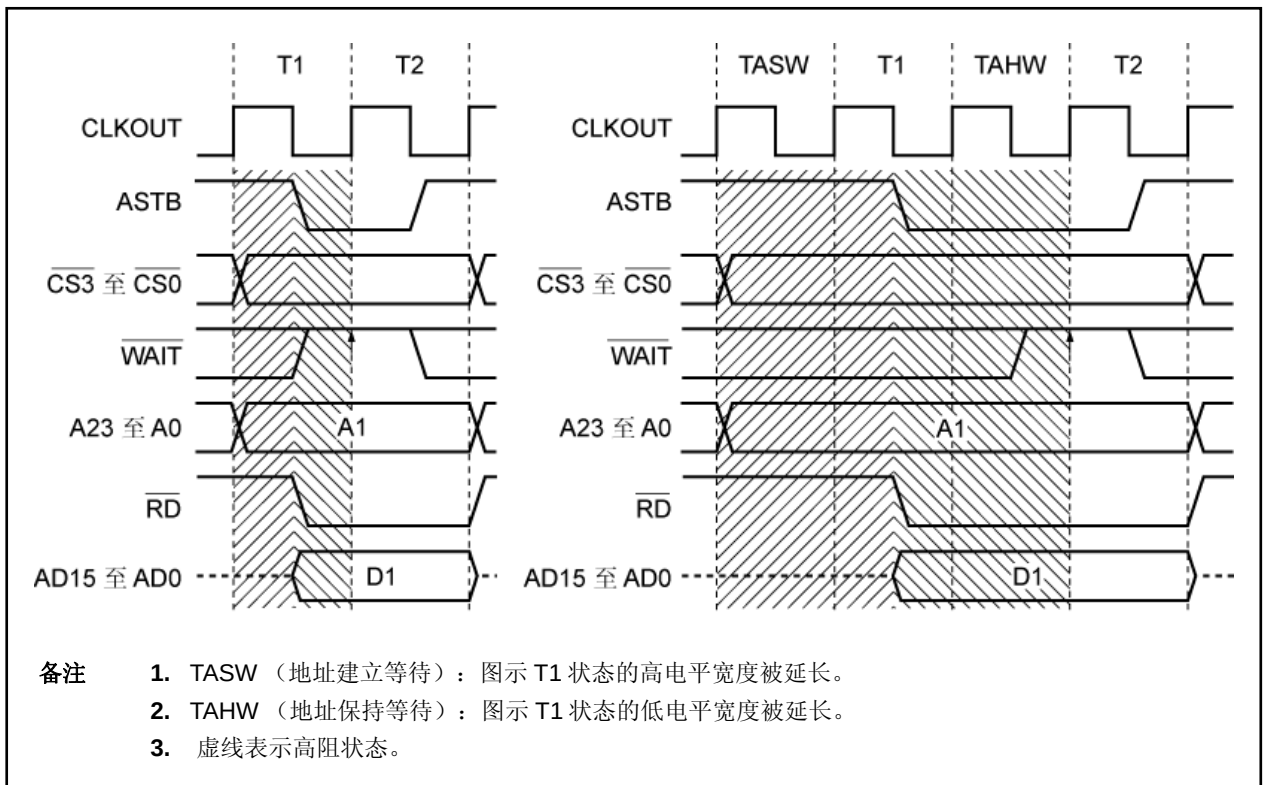


图 5-14. 地址等待时序（分离总线读，总线宽度：16 位，16 位存取）



6.1 概述

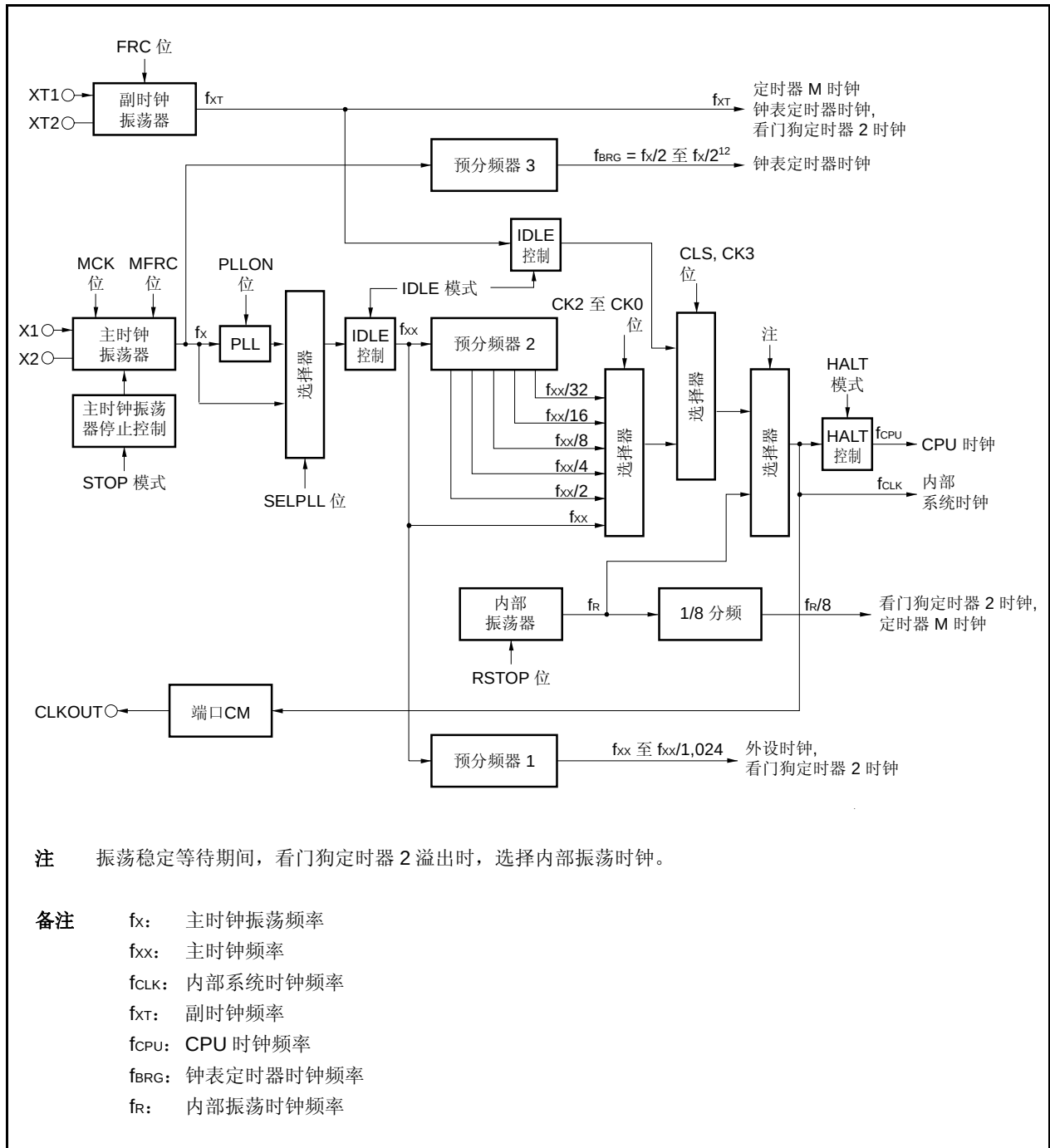
具备以下时钟发生功能：

- 主时钟振荡器
 - 时钟直通模式
 $f_x = 2.5$ 至 10 MHz ($f_{xx} = 2.5$ 至 10 MHz)
 - PLL 模式
 $f_x = 2.5$ 至 5 MHz ($\times 4$: $f_{xx} = 10$ 至 20 MHz)
 $f_x = 2.5$ 至 4 MHz ($\times 8$: $f_{xx} = 20$ 至 32 MHz)
- 副时钟振荡器
 - $f_{XT} = 32.768\text{ kHz}$
- PLL（锁相环）倍频（ $\times 4/\times 8$ ）功能
 - 时钟直通模式/PLL 模式可选
- 内部振荡器
 - $f_R = 220\text{ kHz}$ （典型值）
- 内部系统时钟的产生
 - 7 个级别（ f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/8$, $f_{xx}/16$, $f_{xx}/32$, f_{XT} ）
- 外设时钟产生功能
- 时钟输出功能

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

6.2 配置

图 6-1. 时钟发生器



(1) 主时钟振荡器

主时钟振荡器振荡产生以下频率 (f_x)。

- 时钟直通模式
 $f_x = 2.5$ 至 10 MHz
- PLL 模式
 $f_x = 2.5$ 至 5 MHz ($\times 4$)
 $f_x = 2.5$ 至 4 MHz ($\times 8$)

(2) 副时钟振荡器

副时钟振荡器振荡于 32.768 kHz (f_{xt}) 频率。

(3) 主时钟振荡器停止控制

该电路产生一个控制信号，可以使主时钟振荡器停止工作。

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

(4) 内部振荡器

以 220 kHz（典型值）频率振荡。

(5) 预分频器 1

该预分频器产生 (f_{xx} 至 $f_{xx}/1,024$) 频率的时钟，用来供给以下片上外设工作：TMP0 至 TMP8，TMQ0，TMM0，CSIB0 至 CSIB5，UARTA0 至 UARTA3，I²C00 至 I²C02，ADC 以及 WDT2。

(6) 预分频器 2

该电路对主时钟 (f_{xx}) 进行分频。

预分频器 2 产生的时钟 (f_{xx} 至 $f_{xx}/32$) 提供给一个选择器，产生 CPU 时钟 (f_{cpu}) 和内部系统时钟 (f_{clk})。

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

(7) 预分频器 3

该电路通过对主时钟振荡器产生的时钟 (f_x) 进行分频，得到特定频率 (32.768 kHz) 的时钟，并将该时钟供给钟表定时器模块。

详情请参见第十章 钟表定时器功能

(8) PLL

本电路对主时钟振荡器产生的时钟 (f_x) 进行 4 倍频或 8 倍频。

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

无论时钟是否通过 CKC.CKDIV0 位进行 4 倍频或 8 倍频，PLL 由 PLLCTL.PLLON 位进行启动或停止。

6.3 寄存器

(1) 处理器时钟控制寄存器 (PCC)

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

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被置为 03H。

复位后: 03H R/W 地址: FFFFF828H

PCC

	7	<6>	5	<4>	<3>	2	1	0
	FRC	MCK	MFRC	CLS ^注	CK3	CK2	CK1	CK0
	FRC	副时钟片上反馈电阻的使用						
	0	使用						
	1	未使用						
	MCK	主时钟振荡器控制						
	0	允许振荡						
	1	停止振荡						
	<ul style="list-style-type: none">当主时钟作为CPU时钟工作时，即使MCK位设置为(1)，主时钟的操作也不会停止。CPU时钟源切换为副时钟后，主时钟停止操作。将 MCK 位从0变为1之前，停止所有使用主时钟操作的片上外设功能。主时钟停止，器件工作于副时钟时，清除 MCK位为(0)并且在CPU时钟转换为主时钟之前，或操作片上外设功能之前，通过软件确保振荡稳定时间。							
	MFRC	主时钟片上反馈电阻的使用						
	0	使用 (使用陶瓷/晶体谐振器时)						
	1	不使用 (使用外部时钟时)						
	CLS ^注	CPU 时钟的状态 (f _{CPU})						
	0	主时钟操作						
	1	副时钟操作						
	CK3	CK2	CK1	CK0	时钟选择 (f _{CLK} /f _{CPU})			
	0	0	0	0	f _{xx}			
	0	0	0	1	f _{xx} /2			
	0	0	1	0	f _{xx} /4			
	0	0	1	1	f _{xx} /8			
	0	1	0	0	f _{xx} /16			
	0	1	0	1	f _{xx} /32			
	0	1	1	x	禁止设置			
	1	x	x	x	f _{XT}			

注 CLS 为只读位。

- 注意事项 1. 当 CLKOUT 正在输出时，不要改变 CPU 时钟（使用 CK3 至 CK0 位）。
2. 使用一个位处理指令来操作 CK3 位。当使用 8 位处理指令时，不要改变 CK2 至 CK0 位的设定值。

备注 x: 无需理会

(a) 设置主时钟运行→副时钟运行的举例：

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

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

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

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

[说明示例]

```

_DMA_DISABLE:
    ctrl      0, DCHCn[r0]          -- DMA 操作禁止。n = 0 至 3
<1> _SET_SUB_RUN :
    st.b      r0, PRCMD[r0]
    set1      3, PCC[r0]            -- CK3 位 ← 1
<2> _CHECK_CLS :
    tst1      4, PCC[r0]            -- 等待, 直到副时钟开始工作。
    bz        _CHECK_CLS
<3> _STOP_MAIN_CLOCK :
    st.b      r0, PRCMD[r0]
    set1      6, PCC[r0]            -- MCK 位 ← 1, 主时钟停止。
_DMA_ENABLE:
    setl      0, DCHCn[r0]          -- DMA 操作使能。 n = 0 至 3

```

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

(b) 设置副时钟工作→主时钟工作的举例：

- <1> MCK 位 ← 0: 主时钟振荡器开始工作。
- <2> 在程序中插入等待指令一直到主时钟振荡器稳定时间已过。
- <3> CK3 位 ← 0: 建议使用一个位处理指令。不要改变 CK2 至 CK0 位。
- <4> 主时钟工作: 从 CK3 位设置完成后，直到主时钟开始工作，需要以下时间：
 最大值： $1/f_{XT}$ ($1/\text{副时钟频率}$)。
 所以，CK3 位清为 0 后，应紧接着插入一条 NOP 指令，或通过读取 CLS 位来确定主时钟是否开始工作。

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

[说明示例]

```

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

```

备注 以上描述仅为举例。注意上面的<4>部分，在一个闭合循环中读取 CLS 位。

(2) 内部振荡模式寄存器 (RCM)

RCM 寄存器是 8 位寄存器，用于设置内部振荡器的工作模式。
该寄存器可进行字节读写或按位读写。
系统复位后，该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF80CH

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

RSTOP	内部振荡器的振荡/停止
0	内部振荡器振荡
1	内部振荡器停止

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

(3) CPU 工作时钟状态寄存器 (CCLS)

CCLS 寄存器表明 CPU 工作时钟的状态。
本寄存器为只读寄存器，可以按字节读取或按位读取。
系统复位后，该寄存器设为 00H。

复位后: 00H^注 R 地址: FFFFF82EH

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

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

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

6.4 操作

6.4.1 各时钟的操作

下表显示各个时钟的操作状态：

表 6-1. 各时钟的操作状态

寄存器设置和操作状态 目标时钟	PCC 寄存器								
	CLK 位 = 0, MCK 位 = 0					CLS 位 = 1, MCK 位 = 0		CLS 位 = 1, MCK 位 = 1	
	复位期间	振荡稳定定时器计数期间	HALT 模式	IDLE1, IDLE2 模式	STOP 模式	副时钟模式	次级空闲模式	副时钟模式	次级空闲模式
主时钟振荡器 (f _x)	×	○	○	○	×	○	○	×	×
副时钟振荡器 (f _{XT})	○	○	○	○	○	○	○	○	○
CPU 时钟 (f _{CPU})	×	×	×	×	×	○	×	○	×
内部系统时钟 (f _{CLK})	×	×	○	×	×	○	×	○	×
主时钟 (PLL 模式, f _{xx})	×	○ ^注	○	×	×	○	○	×	×
外设时钟 (f _{xx} 至 f _{xx} /1,024)	×	×	○	×	×	○	×	×	×
钟表定时器时钟 (主时钟源)	×	○	○	○	×	○	○	×	×
钟表定时器时钟 (副时钟源)	○	○	○	○	○	○	○	○	○
WDT2 时钟 (内部振荡)	×	○	○	○	○	○	○	○	○
WDT2 时钟 (主时钟源)	×	×	√	×	×	○	×	×	×
WDT2 时钟 (副时钟源)	○	○	○	○	○	○	○	○	○

注 入锁时间

备注 ○: 可运行
×: 已停止

6.4.2 时钟输出功能

时钟输出功能用于从 CLKOUT 引脚输出内部系统时钟 (f_{CLK})。

通过 PCC.CK3 至 PCC.CK0 位选择系统内部时钟 (f_{CLK})。

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

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

6.5 PLL 功能

6.5.1 概述

V850ES/JJ3 单片机中，CPU 和片上外设功能的工作时钟可以选择 PLL 功能输出的 4 倍或 8 倍振荡频率，或选择时钟直通模式。

- 当使用锁相环功能（×4）时： 输入时钟 = 2.5 至 5 MHz （输出：10 至 20 MHz）。
- 当使用锁相环功能（×8）时： 输入时钟 = 2.5 至 4 MHz （输出：20 至 32 MHz）。
- 时钟直通模式： 输入时钟 = 2.5 至 10 MHz （输出：2.5 至 10 MHz）。

6.5.2 寄存器

（1）锁相环控制寄存器（PLLCTL）

PLLCTL 寄存器是 8 位寄存器，用于控制 PLL 功能。
寄存器可进行字节读写或按位读写。
系统复位后，本寄存器设置为 01H。

复位后: 01H R/W 地址: FFFFF82CH

PLLCTL

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

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

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

注意事项

1. 当 PLLON 位被清 0 时，SELPLL 位自动清除为 0（时钟直通模式）。

2. 只有在 PLL 时钟频率稳定后，SELPLL 位才能够被设置为 1。如果不是这样（未锁定），向其写入数据都会将“0”写入 SELPLL 位。

(2) 时钟控制寄存器 (CKC)

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

CKC 寄存器控制 PLL 模式下的内部系统时钟。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器设为 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模式下内部系统时钟 (f _{xx})
0	f _{xx} = 4 × f _x (f _x = 2.5 至 5.0 MHz)
1	f _{xx} = 8 × f _x (f _x = 2.5 至 4.0 MHz)

- 注意事项
- 1. f_x = 5.0 至 10.0 MHz，不能使用 PLL 模式。
 - 2. 通过 CKC 寄存器在 4 倍和 8 倍之间改变倍频因素之前，设置时钟直通模式并停止 PLL。
 - 3. 请确保将第 3 位和第 1 位设置为“1”，并将第 7 位至第 4 位和第 2 位清除为“0”。

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

(3) 锁定寄存器 (LOCKR)

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

LOCKR 寄存器包括 LOCK 位，该位用于反映 PLL 频率稳定状态。

本寄存器为只读寄存器，按字节读取或按位读取。

系统复位后，该寄存器设为 00H。

复位后: 00H R 地址: FFFFF824H

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

LOCK	PLL 锁存状态检查
0	锁定状态
1	未锁定状态

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

[设置条件]

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

注 复位将该寄存器设置为 01H，复位解除且振荡稳定时间之后清为 00H。

[清除条件]

- 在复位解除后振荡稳定时间（OSTS 寄存器缺省时间（参见 21.2 （3） 振荡稳定时间选择寄存器（OSTS）））溢出时
- 在 STOP 模式释放后振荡稳定定时器（时间由 OSTS 寄存器设置）溢出时，该 STOP 模式是在 PLL 操作状态下设置的
- 在 PLL 锁存时间定时器（时间由 PLLS 寄存器设置）溢出时，当 PLLCTL.PLLON 位从 0 变到 1 后
- 在 IDLE2 模式释放时插入的建立时间（时间由 OSTS 寄存器设置）完成后，该 IDLE2 模式是在 PLL 操作状态下设置的

(4) PLL 锁定时间规范寄存器 (PLLS)

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

本寄存器可以按字节读取或写入。

系统复位后，该寄存器置为 03H。

复位后: 03H R/W 地址: FFFFF6C1H

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

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

- 注意事项**
1. 设置入锁时间为 800 μ s 或更长。
 2. 入锁期间不要改变 PLLS 寄存器的设置。

6.5.3 用法**(1) 当使用锁相环时**

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

(a) 从时钟直通模式转变至 IDLE2 或 STOP 模式

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

(b) 在 PLL 操作模式中转变至 IDLE 2 或 STOP 模式

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

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

(2) 当未使用锁相环时

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

第七章 16 位定时器/事件计数器 P (TMP)

定时器 P (TMP) 是 16 位定时器/事件计数器。

V850ES/JJ3 单片机有九个定时器/事件计数器通道：TMP0 至 TMP8。

7.1 概述

TMPn 概况如下：

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

备注 n = 0 至 8

7.2 功能

TMPn 具有如下功能：

- 间隔定时器
- 外部事件计数器
- 外部触发脉冲输出
- 单次触发脉冲输出
- PWM 输出
- 自由运行定时器
- 脉冲宽度测量

备注 n = 0 至 8

7.3 配置

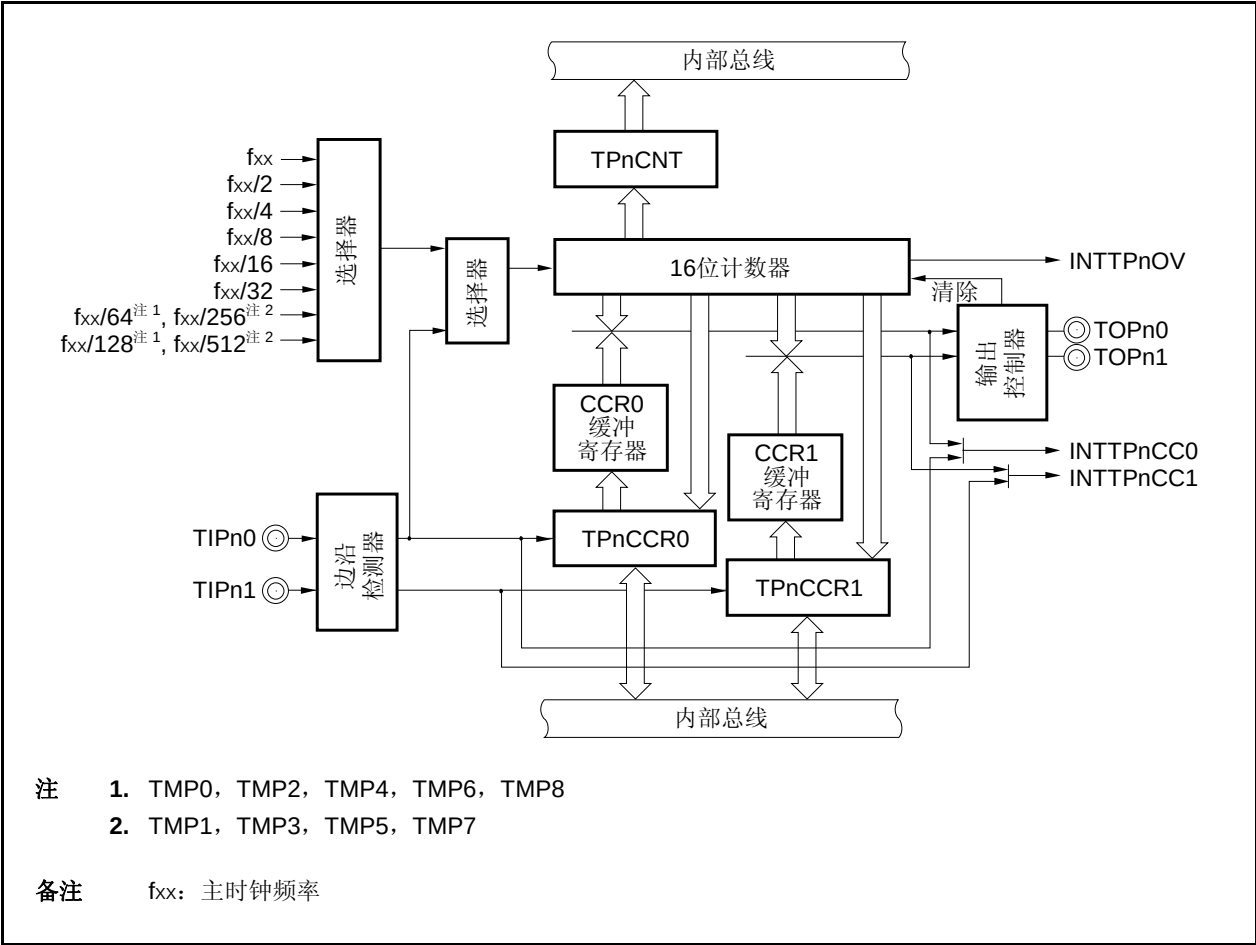
TMPn 包括下列硬件：

表 7-1. 定时器 TMPn 的配置

项目	配置
定时器寄存器	16 位计数器
寄存器	TMPn 捕获/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 输入/输出控制寄存器 0 至 2（TPnIOC0 至 TPnIOC2） TMPn 选项寄存器 0（TPnOPT0）

- 注 1. TIPn0 引脚功能可以复用为捕获触发、外部事件计数以及外部触发的信号输入引脚。
2. 当使用 TIPn0, TIPn1, TOPn0 和 TOPn1 引脚功能时，参见表 4-19 端口引脚用作复用功能引脚。
- 备注 n = 0 至 8

图 7-1. 定时器 TMPn 框图



- 注 1. TMP0, TMP2, TMP4, TMP6, TMP8
2. TMP1, TMP3, TMP5, TMP7

备注 fxx: 主时钟频率

(1) 16 位计数器

此 16 位计数器可以计数内部时钟或外部事件。

该计数器的计数值可以通过读取 TPnCNT 寄存器得到。

当 TPnCTL0.TPnCE 位= 0 时, 16 位计数器的值为 FFFFH。如果此时读取 TPnCNT 寄存器, 则读回的值为 0000H。

系统复位将 TPnCE 位清 0, 因此, 16 位计数器设置为 FFFFH。

(2) CCR0 缓冲寄存器

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

当 TPnCCR0 寄存器用作比较寄存器时, 写入 TPnCCR0 寄存器的值会传送到 CCR0 缓冲寄存器。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值相匹配时, 就会产生一个比较匹配中断请求信号 (INTTPnCC0)。

CCR0 缓冲寄存器不能直接读写。

CCR0 缓冲寄存器在复位后清零为 0000H, TPnCCR0 寄存器清零为 0000H。

(3) CCR1 缓冲寄存器

CCR1 缓冲寄存器是一个 16 位比较寄存器, 用于比较 16 位计数器的计数值。

当 TPnCCR1 寄存器用作比较寄存器时, 写入 TPnCCR1 寄存器的值会传送到 CCR1 缓冲寄存器。当 16 位计数器的计数值和 CCR1 缓冲寄存器的值匹配时, 也会产生一个比较匹配中断请求信号 (INTTPnCC1)。

CCR1 缓冲寄存器不能直接读写。

CCR1 缓冲寄存器在复位后清零为 0000H, TPnCCR1 寄存器清零为 0000H。

(4) 边沿检测器

该电路检测输入到 TIPn0 和 TIPn1 引脚信号的有效边沿。通过设置 TPnIOC1 和 TPnIOC2 寄存器, 可以选择无边沿、上升沿、下降沿、上升/下降双边沿作为有效沿。

(5) 输出控制器

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

(6) 选择器

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

7.4 寄存器

以下寄存器控制 TMPn:

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

备注

1. 当使用 TIPn0, TIPn1, TOPn0 和 TOPn1 引脚的功能时, 参见表 4-19 端口引脚用作复用功能引脚。
2. n = 0 至 8

(1) TMPn 控制寄存器 0 (TPnCTL0)

TPnCTL0 寄存器是 8 位寄存器，用于控制定时器 TMPn 的操作。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

随时可以用软件将相同的值写入 TPnCTL0 寄存器。

复位后: 00H R/W 地址: TP0CTL0 FFFFF590H, TP1CTL0 FFFFF5A0H,
TP2CTL0 FFFFF5B0H, TP3CTL0 FFFFF5C0H,
TP4CTL0 FFFFF5D0H, TP5CTL0 FFFFF5E0H,
TP6CTL0 FFFFF5F0H, TP7CTL0 FFFFF600H,
TP8CTL0 FFFFF610H

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

(n = 0 至 8)

TPnCE	TMPn 操作控制
0	TMPn 操作禁止 (TMPn 异步复位 ^注).
1	TMPn 操作允许 TMPn 操作开始

TPnCKS2	TPnCKS1	TPnCKS0	内部计数时钟选择	
			n = 0, 2, 4, 6, 8	n = 1, 3, 5, 7
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. 当 TPnCE 位=0 时，设置 TPnCKS2 至 TPnCKS0 位为“1”。
当 TPnCE 位的值从 0 变为 1 时，可以同时设定 TPnCKS2 至 TPnCKS0 位。
 2. 请确保将第 3 位至第 6 位清为“0”。

备注 f_{xx}: 主时钟频率

(2) TMPn 控制寄存器 1 (TPnCTL1)

TPnCTL1 寄存器是 8 位寄存器，用于控制定时器 TMPn 的操作。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: TP0CTL1 FFFFF591H, TP1CTL1 FFFFF5A1H,
 TP2CTL1 FFFFF5B1H, TP3CTL1 FFFFF5C1H,
 TP4CTL1 FFFFF5D1H, TP5CTL1 FFFFF5E1H,
 TP6CTL1 FFFFF5F1H, TP7CTL1 FFFFF601H,
 TP8CTL1 FFFFF611H

	7	<6>	<5>	4	3	2	1	0
TPnCTL1 (n = 0 至 8)	0	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0

TPnEST	软件触发控制
0	-
1	为外部触发输入产生一个有效的信号 • 单次触发脉冲输出模式: 将1 写入 TPnEST 位, 作为触发, 输出一个单次触发脉冲 • 外部触发脉冲输出模式: 将 1 写入 TPnEST 位, 作为触发, 输出一个PWM波形

TPnEEE	计数时钟选择
0	禁止使用外部事件计数输入操作 (对TPnCTL0.TPnCK0至 TPnCK2 位所选的计数时钟计数.)
1	允许使用外部事件计数输入操作 (对外部事件计数输入信号的有效沿计数.)
由TPnEEE位来选择是使用内部时钟计数, 还是外部事件计数输入的有效沿计数.	

TPnMD2	TPnMD1	TPnMD0	定时器模式选择
0	0	0	内部定时器模式
0	0	1	外部事件计数模式
0	1	0	外部触发脉冲输出模式
0	1	1	单次触发脉冲输出模式
1	0	0	PWM 输出模式
1	0	1	自由运行定时器模式
1	1	0	脉冲宽度测量模式
1	1	1	禁止设置

- 注意事项**
1. 只有在外触发脉冲输出模式或单次触发脉冲输出模式下, TPnEST 位有效。在其它模式下, 忽略向该位写入 1 的操作。
 2. 不论 TPnEEE 位的值如何, 在外部事件计数模式下, 都会选择外部事件输入。
 3. TPnCTL0.TPnCE = 0 时, 设置 TPnEEE 和 TPnMD2 至 TPnMD0 位 (TPnCE 位=1 时, 可以写入相同的值)。在 TPnCE 位= 1 时, 重写操作是无法保证的, 若重写发生错误, 应把 TPnCE 位清为 0, 再重新对这些位进行设定。
 4. 请确保将第 3 位, 第 4 位和第 7 位清为“0”。

(3) TMPn 输入/输出控制寄存器 0 (TPnIOC0)

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

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后：00H R/W 地址： TP0IOC0 FFFFF592H, TP1IOC0 FFFFF5A2H,
 TP2IOC0 FFFFF5B2H, TP3IOC0 FFFFF5C2H,
 TP4IOC0 FFFFF5D2H, TP5IOC0 FFFFF5E2H,
 TP6IOC0 FFFFF5F2H, TP7IOC0 FFFFF602H,
 TP8IOC0 FFFFF612H

	7	6	5	4	3	<2>	1	<0>
TPnIOC0 (n = 0 至 8)	0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0

TPnOL1	TOPn1 引脚输出电平设置 ^注
0	TOPn1 引脚开始输出高电平
1	TOPn1 引脚开始输出低电平

TPnOE1	TOPn1 引脚输出设置
0	定时器输出禁止 • 当 TPnOL1 位 = 0 时：TOPn1 引脚输出低电平 • 当 TPnOL1 位 = 1 时：TOPn1 引脚输出高电平
1	定时器输出允许 (从 TOPn1 引脚输出方波)。

TPnOL0	TOPn0 引脚输出电平设置 ^注
0	TOPn0 引脚开始输出高电平
1	TOPn0 引脚开始输出低电平

TPnOE0	TOPn0 引脚输出设置
0	定时器输出禁止 • 当 TPnOL0 位 = 0 时：TOPn0 引脚输出低电平 • 当 TPnOL0 位 = 1 时：TOPn0 引脚输出高电平
1	定时器输出允许 (从 TOPn0 引脚输出方波)。

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

· 当 TPnOLm 位 = 0 时



· 当 TPnOLm 位 = 1 时



- 注意事项**
1. 当 TPnCTL0.TPnCE 位 = 0 时，重新写入 TPnOL1，TPnOE1，TPnOL0 和 TPnOE0 位（当 TPnCE 位 = 1 时，可以写入相同的值）。如果重新写入不能正确执行，将 TPnCE 位清为 0，然后重新设定该位。
 2. 即使在 TPnCE 和 TPnOEm 位均为 0 时对 TPnOLm 位进行操作，TOPnm 引脚的输出电平也会改变 (m=0, 1)。

(4) TMPn 输入/输出控制寄存器 1 (TPnIOC1)

TPnIOC1 寄存器是 8 位寄存器，用于控制捕获触发输入信号 (TIPn0, TIPn1 引脚) 的有效边沿。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: TP0IOC1 FFFFF593H, TP1IOC1 FFFFF5A3H,
TP2IOC1 FFFFF5B3H, TP3IOC1 FFFFF5C3H,
TP4IOC1 FFFFF5D3H, TP5IOC1 FFFFF5E3H,
TP6IOC1 FFFFF5F3H, TP7IOC1 FFFFF603H,
TP8IOC1 FFFFF613H

	7	6	5	4	3	2	1	0
TPnIOC1 (n = 0 至 8)	0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0

TPnIS3	TPnIS2	捕获触发输入信号 (TIPn1 引脚) 有效沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

TPnIS1	TPnIS0	捕获触发输入信号 (TIPn0 引脚) 有效沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

- 注意事项**
1. 当 TPnCTL0.TPnCE 位=0 时，重新写入 TPnIS3 至 TPnIS0 位（当 TPnCE 位=1 时，可写入相同值）。如果重新写入不能正确执行，将 TPnCE 位清为 0，然后重新进行位的设定。
 2. 只有在自由运行定时器模式和脉冲宽度测量模式下，TPnIS3 至 TPnIS0 位才有效。在所有其它模式下，无法进行捕获操作。

(5) TMPn 输入/输出控制寄存器 2 (TPnIOC2)

TPnIOC2 寄存器是 8 位寄存器，用于控制外部事件计数器输入信号（TIPn0 引脚）和外部触发输入信号（TIPn0 引脚）的有效边沿。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: TP0IOC2 FFFFF594H, TP1IOC2 FFFFF5A4H,
 TP2IOC2 FFFFF5B4H, TP3IOC2 FFFFF5C4H,
 TP4IOC2 FFFFF5D4H, TP5IOC2 FFFFF5E4H,
 TP6IOC2 FFFFF5F4H, TP7IOC2 FFFFF604H,
 TP8IOC2 FFFFF614H

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

(n = 0 至 8)

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

TPnETS1	TPnETS0	外部触发输入信号 (TIPn0 引脚) 有效沿设置
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	双边沿检测

- 注意事项**
1. 当 **TPnCTL0.TPnCE 位 = 0** 时，重新写入 **TPnEES1**，**TPnEES0**，**TPnETS1** 和 **TPnETS0** 位（当 **TPnCE 位 = 1** 时，可以写入相同的值）。如果重新写入不能正确执行，将 **TPnCE 位** 清为 **0**，然后重新进行位设定。
 2. 只有当 **TPnCTL1.TPnEEE 位 = 1** 或设置为外部事件计数模式（**TPnCTL1.TPnMD2 至 TPnCTL1.TPnMD0 位 = 001**）时，**TPnEES1** 和 **TPnEES0** 位有效。
 3. 只有在外部触发脉冲输出模式（**TPnCTL1.TPnMD2 至 TPnCTL1.TPnMD0 位 = 010**）或在单次触发脉冲输出模式（**TPnCTL1.TPnMD2 至 TPnCTL1.TPnMD0 位 = 011**）时，**TPnETS1** 和 **TPnETS0** 位有效。

(6) TMPn 选项寄存器 0 (TPnOPT0)

TPnOPT0 寄存器是 8 位寄存器，用来设置捕获/比较操作和检测计数溢出。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: TP0OPT0 FFFFF595H, TP1OPT0 FFFFF5A5H,
TP2OPT0 FFFFF5B5H, TP3OPT0 FFFFF5C5H,
TP4OPT0 FFFFF5D5H, TP5OPT0 FFFFF5E5H,
TP6OPT0 FFFFF5F5H, TP7OPT0 FFFFF605H,
TP8OPT0 FFFFF615H

	7	6	5	4	3	2	1	<0>
TPnOPT0 (n = 0 至 8)	0	0	TPnCCS1	TPnCCS0	0	0	0	TPnOVF

TPnCCS1	TPnCCR1 寄存器捕获/比较选择
0	比较寄存器选择
1	捕获寄存器选择
TPnCCS1 位的设置仅在自由运行定时器模式下有效。	

TPnCCS0	TPnCCR0 寄存器捕获/比较选择
0	比较寄存器选择
1	捕获寄存器选择
TPnCCS0 位的设置仅在自由运行定时器模式下有效。	

TPnOVF	TMPn溢出检测标志
置位 (1)	发生溢出
复位(0)	TPnOVF 位写入0,或 TPnCTL0.TPnCE 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，当16位计数器的计数值由FFFFH变为0000H溢出时，TPnOVF位被置位。 在 TPnOVF 位被置为1的同时，产生中断请求信号(INTTPnOV)。 只有在自由运行定时器模式和脉冲宽度测量模式下，才会产生INTTPnOV 信号。 即使在TPnOVF 位 = 1时读取TPnOVF 位或 TPnOPT0 寄存器，TPnOVF 位不清零。 TPnOVF 位可读取也可写入，但是TPnOVF 位不能使用软件置1，写入1对TMPn的操作没有影响。 	

- 注意事项**
1. 当 TPnCE 位= 0 时，重新写入 TPnCCS1 和 TPnCCS0 位（当 TP0CE 位=1 时，可以写入相同的值）。如果重新写入不能正确执行，将 TPnCE 位清为 0，然后重新进行位的设定。
 2. 请确保将第 1 位至第 3 位，第 6 位和第 7 位清为“0”。

（7）TMPn 捕获/比较寄存器 0（TPnCCR0）

根据不同工作模式，TPnCCR0 寄存器可以用作捕获寄存器或比较寄存器。

在自由计时器模式下，根据 TPnOPT0.TPnCCS0 位的设定，TPnCCR0 寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TPnCCR0 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

TPnCCR0 寄存器可以在操作中进行读取和写入。

此寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

注意事项 在以下状态禁止访问 TPnCCR0 寄存器。详细说明，参见 3.4.8（2）访问片上外设输入/输出专用寄存器。

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

复位后： 0000H R/W 地址： TP0CCR0 FFFFF596H, TP1CCR0 FFFFF5A6H,
TP2CCR0 FFFFF5B6H, TP3CCR0 FFFFF5C6H,
TP4CCR0 FFFFF5D6H, TP5CCR0 FFFFF5E6H,
TP6CCR0 FFFFF5F6H, TP7CCR0 FFFFF606H,
TP8CCR0 FFFFF616H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR0 (n = 0 至 8)																

(a) 用作比较寄存器时的功能

即便在 TPnCTL0.TPnCE 位=1 时，TPnCCR0 寄存器仍可以重新写入。

TPnCCR0 寄存器的设定值传送给 CCR0 缓冲寄存器。当 16 位计数器的值与 CCR0 缓冲寄存器的值相匹配时，就会生成一个比较匹配中断请求信号（INTTPnCC0）。如果此时 TOPn0 引脚输出允许，则 TOPn0 引脚的输出会反相。

在间隔定时器模式、外部事件计数器模式、外部触发脉冲输出模式、单次触发脉冲输出模式或 PWM 输出模式时，当 TPnCCR0 寄存器作为周期寄存器时，如果 16 位计数器的计数值与 CCR0 缓冲寄存器数值匹配，则计数器清零为（0000H）。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下，当 TPnCCR0 寄存器用作捕获寄存器时，若检测到捕获触发输入引脚（TIPn0 引脚）的有效边沿，则 16 位计数器的当前计数值存储于 TPnCCR0 寄存器中。在脉宽测量模式下，若检测到捕获触发输入引脚（TIPn0 引脚）的有效边沿，16 位计数器的当前计数值就储存在 TPnCCR0 寄存器中，且 16 位计数器清零（0000H）。

即使捕获操作和读取 TPnCCR0 寄存器的操作发生冲突，还是可以正确读取 TPnCCR0 寄存器的值。

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

表 7-2. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

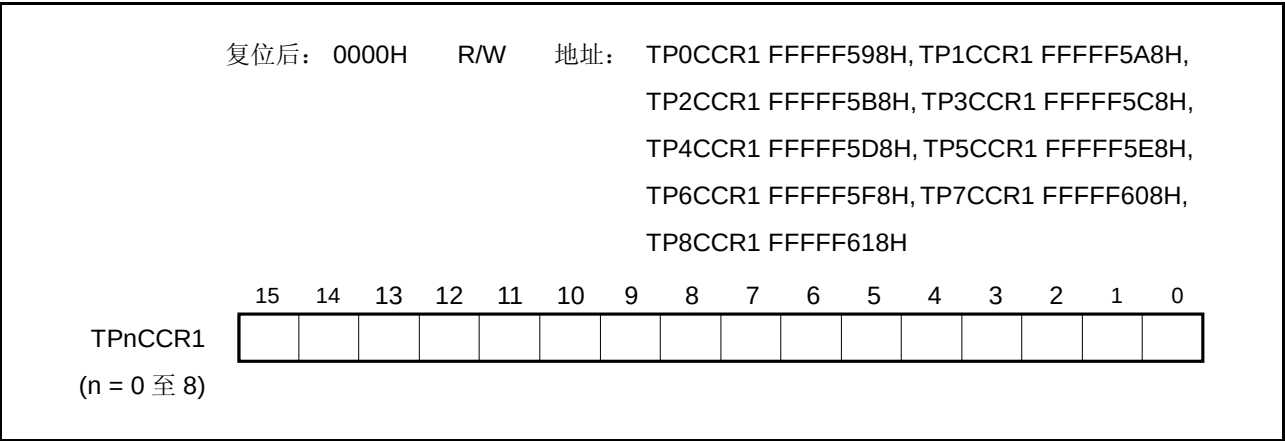
操作模式	捕获/比较寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	任何时候均可写入
外部事件计数器	比较寄存器	任何时候均可写入
外部触发脉冲输出	比较寄存器	批量写入
单次触发脉冲输出	比较寄存器	任何时候均可写入
PWM（脉宽调制信号）输出	比较寄存器	批量写入
自由运行定时器	捕获/比较寄存器	任何时候均可写入
脉冲宽度测量	捕获寄存器	—

(8) TMPn 捕获/比较寄存器 1（TPnCCR1）

根据不同工作模式，TPnCCR1 寄存器可以用作捕获寄存器或比较寄存器。
在自由计时器模式下，根据 TPnOPT0.TPnCCS1 位的设定，TPnCCR1 寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TPnCCR1 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。
TPnCCR1 寄存器可以在操作中进行读取和写入。
此寄存器可以按 16 位宽度进行读取或写入。
复位输入将该寄存器清为 0000H。

注意事项 在以下状态禁止访问 TPnCCR1 寄存器。详细说明，参见 3.4.8（2）访问片上外设输入/输出专用寄存器。

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



(a) 用作比较寄存器时的功能

即便在 TPnCTL0.TPnCE 位=1 时，TPnCCR1 寄存器仍可以重新写入。

TPnCCR1 寄存器的设定值传送给 CCR1 缓冲寄存器。当 16 位计数器的值与 CCR1 缓冲寄存器的值相匹配时，就会生成一个比较匹配中断请求信号（INTTPnCC1）。如果此时 TOPn1 引脚输出允许，则 TOPn1 引脚的输出会反相。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下，当 TPnCCR1 寄存器用作捕获寄存器时，若检测到捕获触发输入引脚（TIPn1 引脚）的有效边沿，则 16 位计数器的当前计数值存储于 TPnCCR1 寄存器中。在脉宽测量模式下，若检测到捕获触发输入引脚（TIPn1 引脚）的有效边沿，16 位计数器的当前计数值就储存在 TPnCCR1 寄存器中，且 16 位计数器清零（0000H）。

即使捕获操作和读取 TPnCCR1 寄存器的操作发生冲突，还是可以正确读取 TPnCCR1 寄存器的值。

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

表 7-3. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

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

7.5 操作

定时器 TMPn 可以执行如下操作。

操作	TPnCTL1.TPnEST 位 (软件触发位)	TIPn0 引脚 (外部触发输入)	捕获/比较寄存器 设置	比较寄存器写入
间隔计时器模式	无效	无效	仅作比较寄存器	任何时候均可写入
外部事件计数器模式 ^{注 1}	无效	无效	仅作比较寄存器	任何时候均可写入
外部触发脉冲输出模式 ^{注 2}	有效	有效	仅作比较寄存器	批量写入
单次触发脉冲输出模式 ^{注 2}	有效	有效	仅作比较寄存器	任何时候均可写入
PWM 输出模式	无效	无效	仅作比较寄存器	批量写入
自由定时器模式	无效	无效	可以切换	任何时候均可写入
脉冲宽度测量模式 ^{注 2}	无效	无效	仅作捕获寄存器	无效

- 注 1. 在外部事件计数器模式下，应该指定对 TIPn0 引脚捕获触发输入的有效边沿不作检测（将 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位清为“00”）。
2. 在外部触发脉冲输出模式、单次触发脉冲输出模式和脉冲宽度测量模式下，选择内部时钟作为计数时钟。（将 TPnCTL1.TPnEEE 位清为 0）。

备注 n = 0 至 8

7.5.1 间隔定时器模式（TPnMD2 至 TPnMD0 位=000）

在间隔定时器模式下，如果 TPnCTL0.TPnCE 位置为 1，则会以指定时间间隔生成中断请求信号（INTTPnCC0），从 TOPn0 引脚输出方波，指定的时间间隔等于方波的半周期。

通常在间隔定时器模式下不使用 TPnCCR1 寄存器。

图 7-2. 间隔定时器的配置

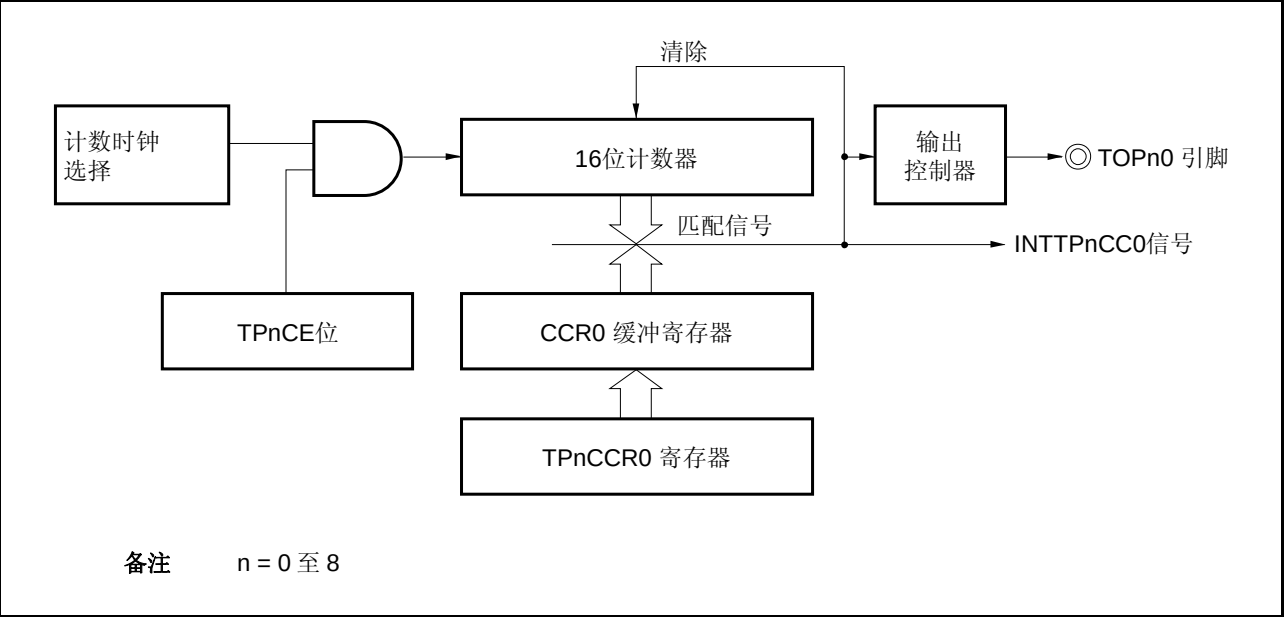
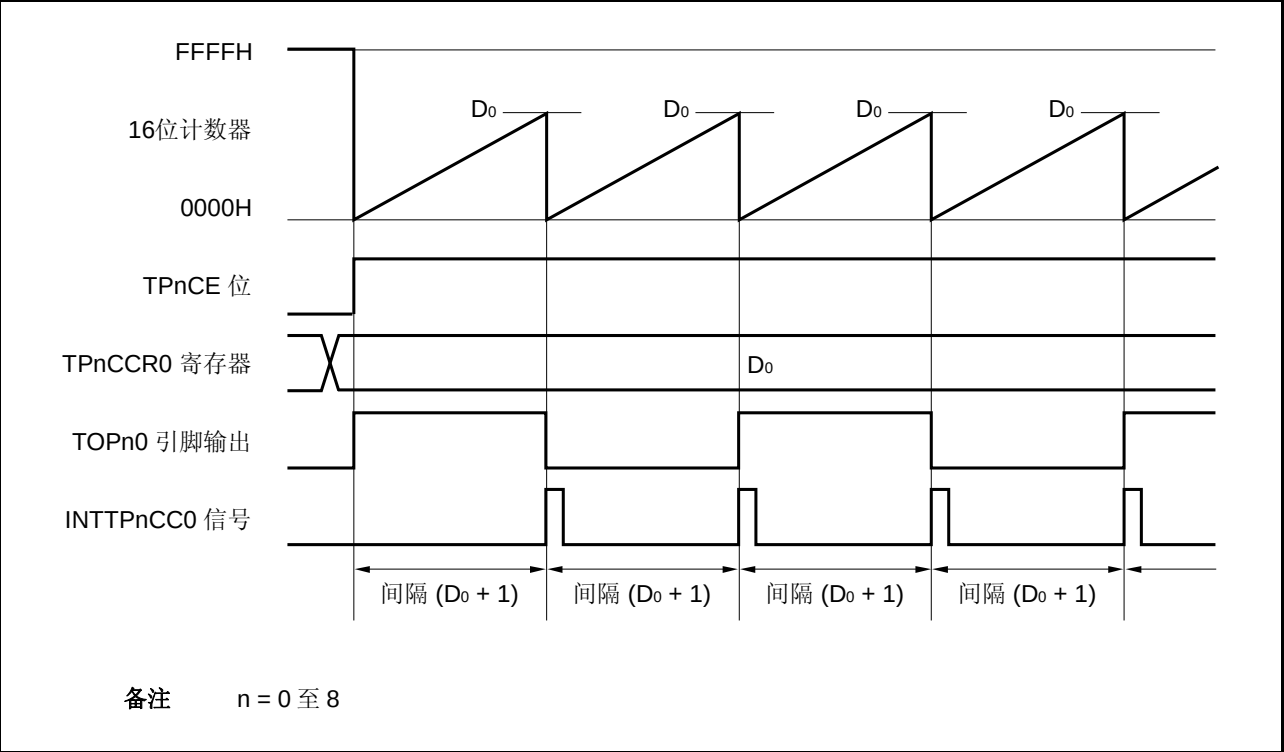


图 7-3. 间隔定时器模式的基本时序



当 TPnCE 位置为 1 时，16 位计数器的值由计数时钟同步从 FFFFH 清零为 0000H，计数器开始计数。此时，TOPn0 引脚的输出反相。此外，TPnCCR0 寄存器的设定值传送给 CCR0 缓冲寄存器。

当 16 位计数器的计数值和 CCR0 缓冲寄存器的值相匹配时，则 16 位计数器清为 0000H，TOPn0 引脚输出反相，并生成一个比较匹配中断请求信号（INTTPnCC0）。

时间间隔可用下述表达式计算。

间隔时间 = (TPnCCR0 寄存器的设定值+1) × 计数时钟周期

备注 n = 0 至 8

图 7-4. 间隔定时器模式操作时的寄存器设置（1/2）

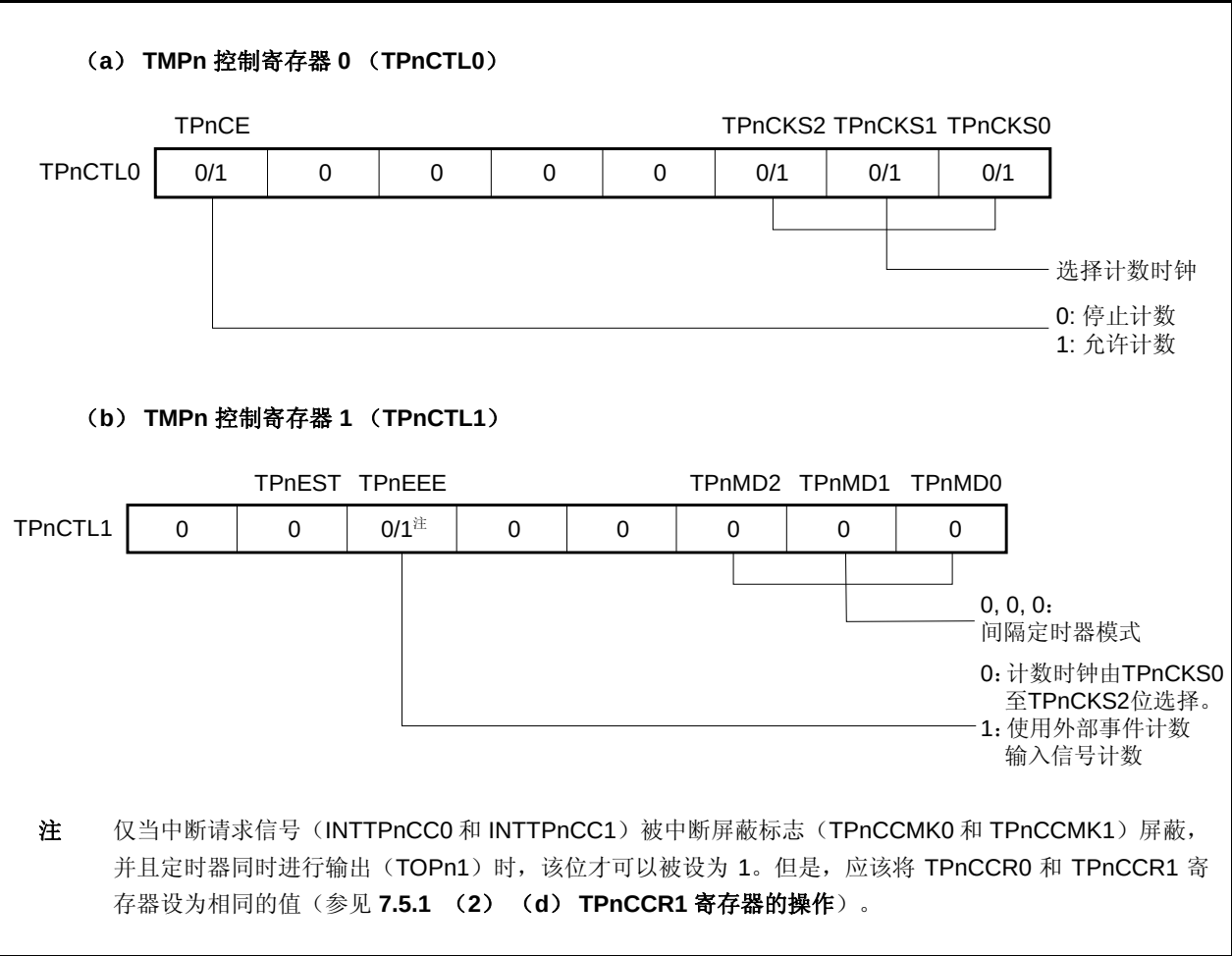
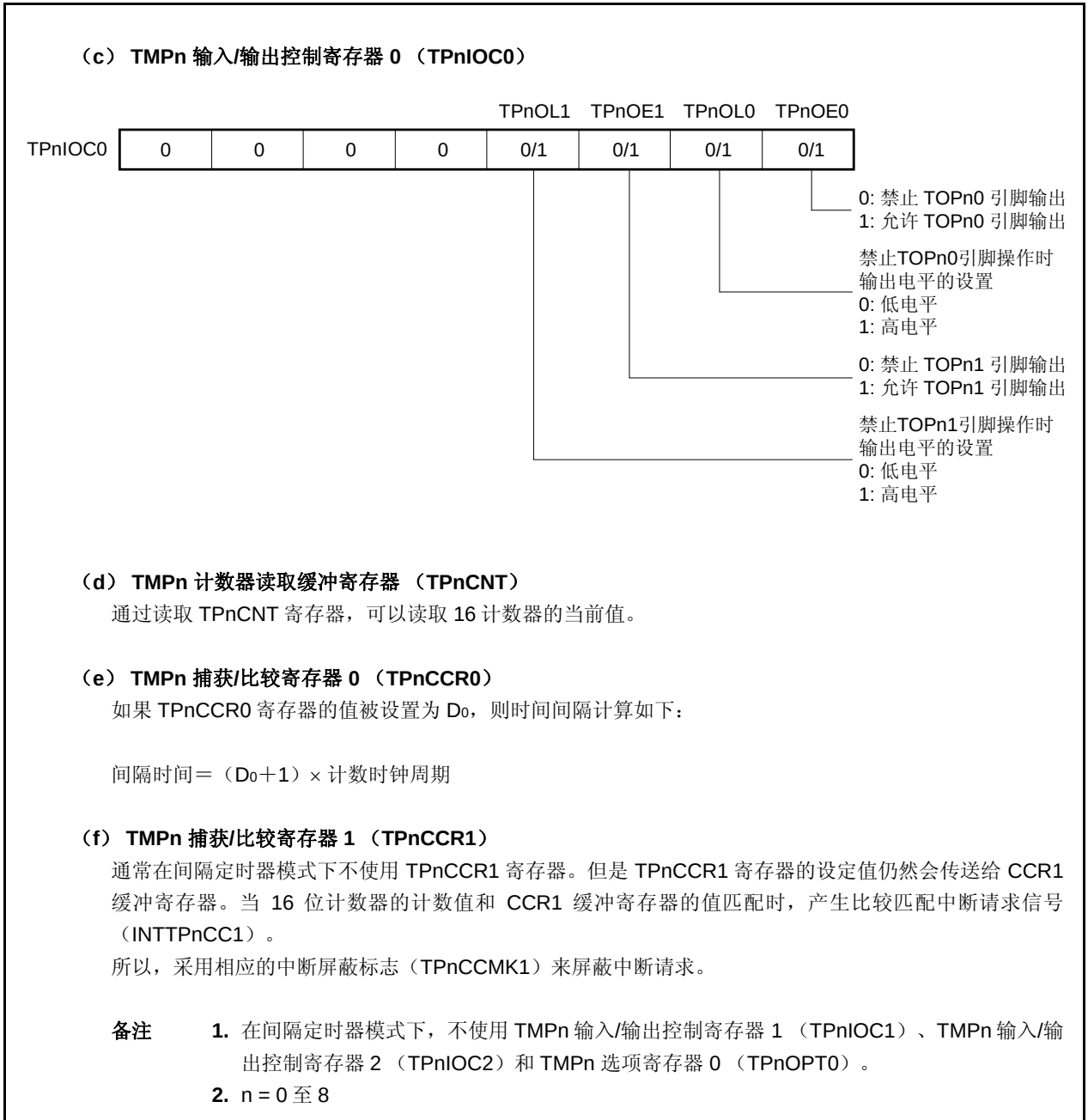
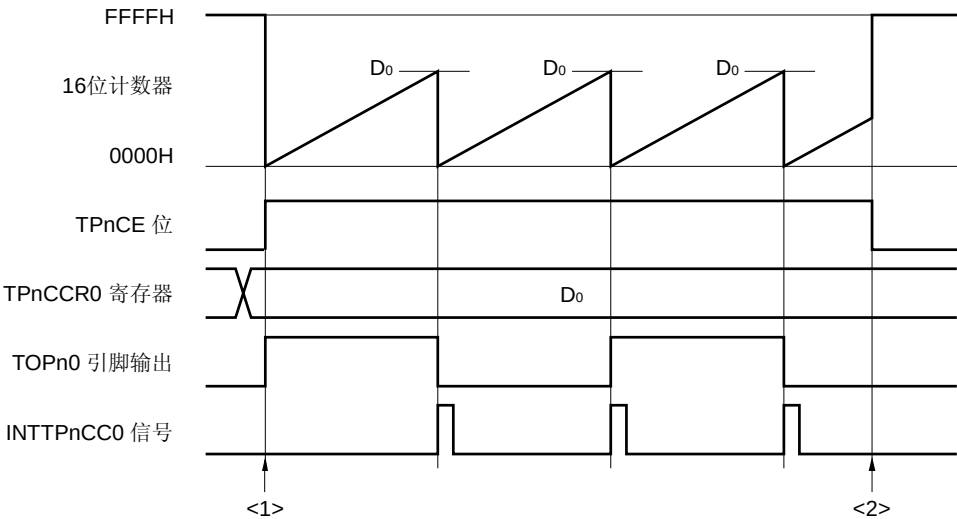


图 7-4. 间隔定时器模式操作时的寄存器设置（2/2）

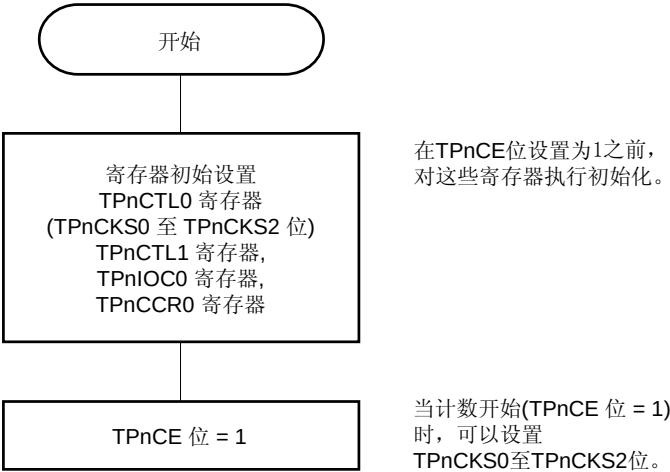


(1) 间隔定时器模式操作流程

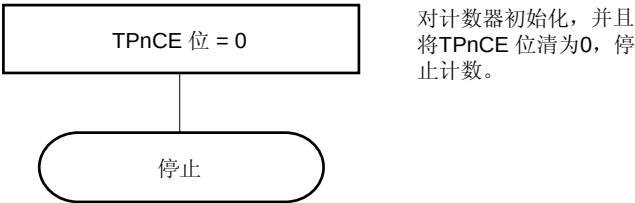
图 7-5. 间隔定时器模式下的软件处理流程



<1> 计数操作开始流程



<2> 计数操作停止流程

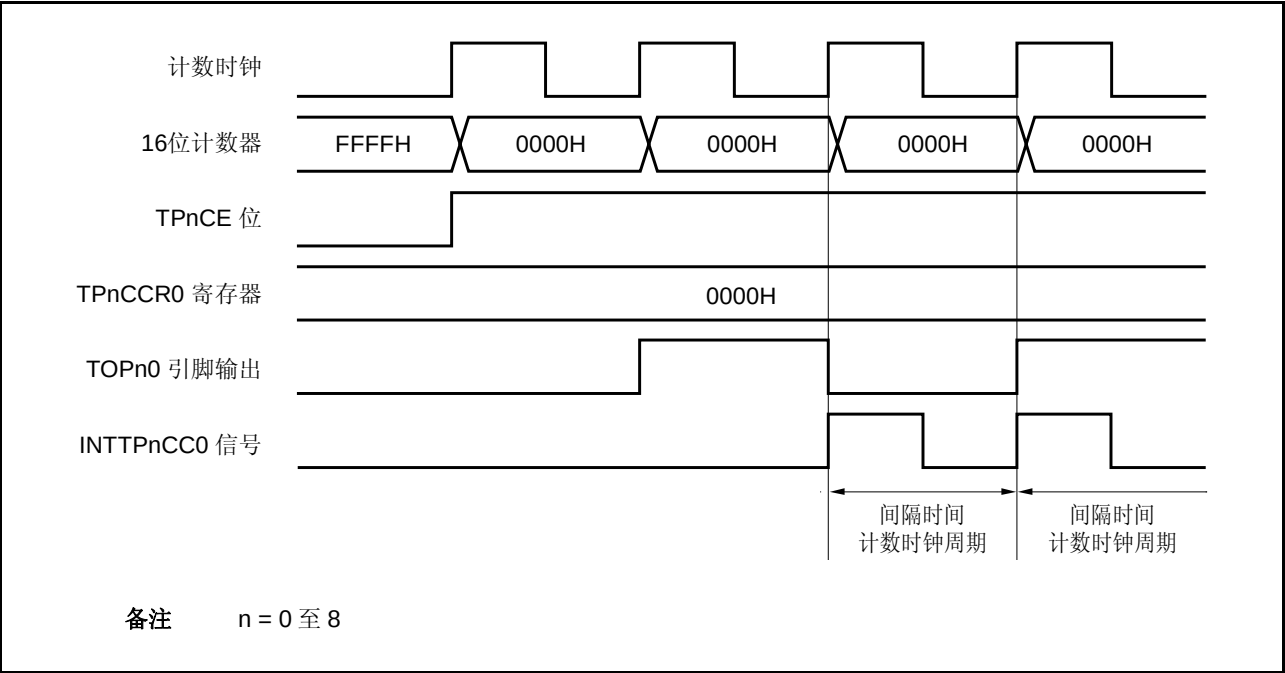


备注 n = 0 至 8

(2) 间隔定时器模式操作时序

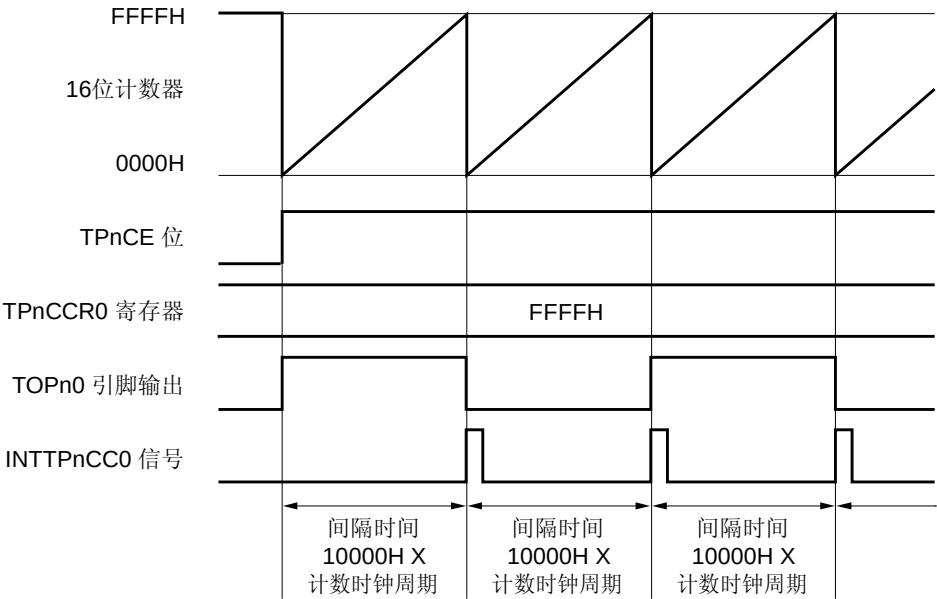
(a) TPnCCR0 寄存器清为 0000H 时的操作

如果 TPnCCR0 寄存器清为 0000H，第一个时钟之后的每个时钟计数时都会产生 INTTPnCC0 信号，而且，TOPn0 引脚的输出反相。
16 位计数器的值总是 0000H。



(b) TPnCCR0 寄存器被置为 FFFFH 时的操作

如果 TPnCCR0 寄存器置为 FFFFH，那么 16 位计数器计数到 FFFFH。该计数器在下一个计数时刻同步清零。产生 INTTPnCC0 信号，并且 TOPn0 引脚输出反相。此时，不产生溢出中断请求信号（INTTPnOV），而且，溢出标志（TPnOPT0.TPnOVF 位）也不会置 1。

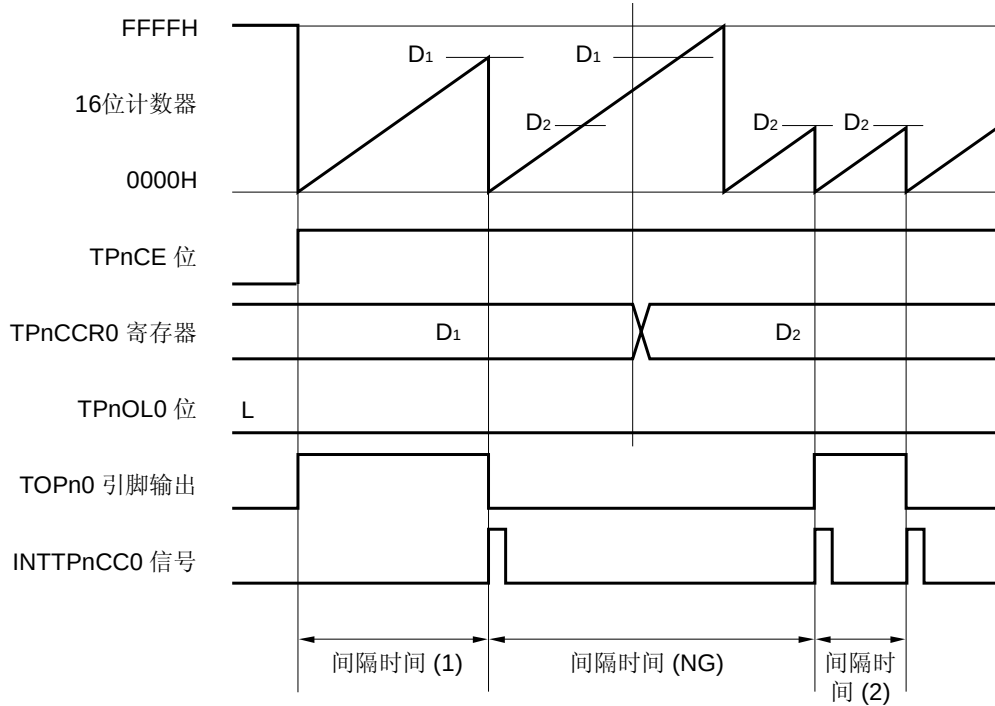


备注 n = 0 至 8

(c) 重新写入 TPnCCR0 寄存器的注意事项

要将 TPnCCR0 寄存器的值改为比当前值较小的值，应停止计数，然后改变其设定值。

如果在计数器工作期间，将 TPnCCR0 寄存器的值重写为比当前值较小的值，16 位计数器可能会溢出。



- 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. $n = 0$ 至 8

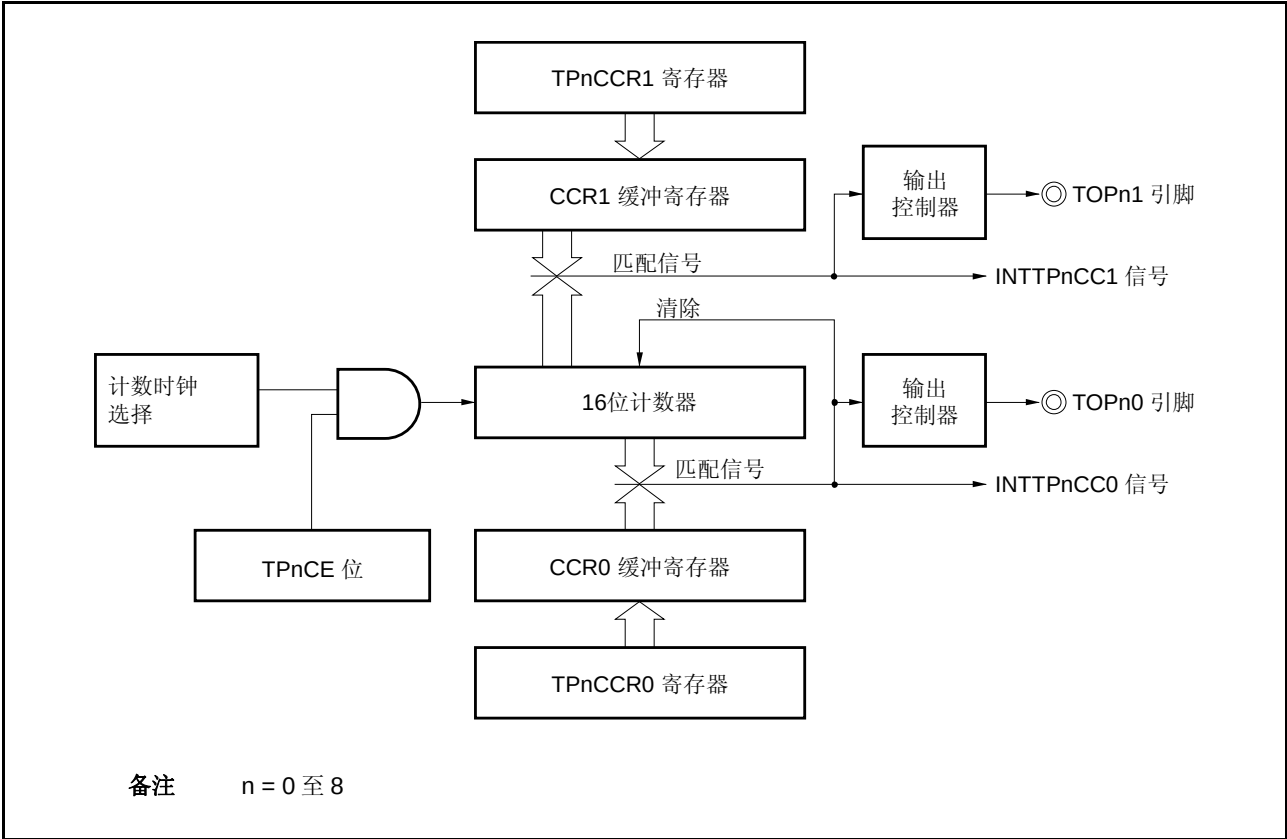
当 16 位计数器计数值大于 D_2 而小于 D_1 时，如果将 TPnCCR0 寄存器的值从 D_1 变为 D_2 ，在重写 TPnCCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。结果，用来比较的 16 位计数器的值是 D_2 。

但是，由于计数值已经超过了 D_2 ，所以 16 位计数器就会向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值和 D_2 相符时才产生 INTTPnCC0 信号，且 TOPn0 引脚的输出反相。

因此，可能不会在原先预计的间隔时间“ $(D_1 + 1) \times \text{计数时钟周期}$ ”或“ $(D_2 + 1) \times \text{计数时钟周期}$ ”时产生 INTTPnCC0 信号，而会以“ $(10000H + D_2 + 1) \times \text{计数时钟周期}$ ”的间隔产生中断请求信号。

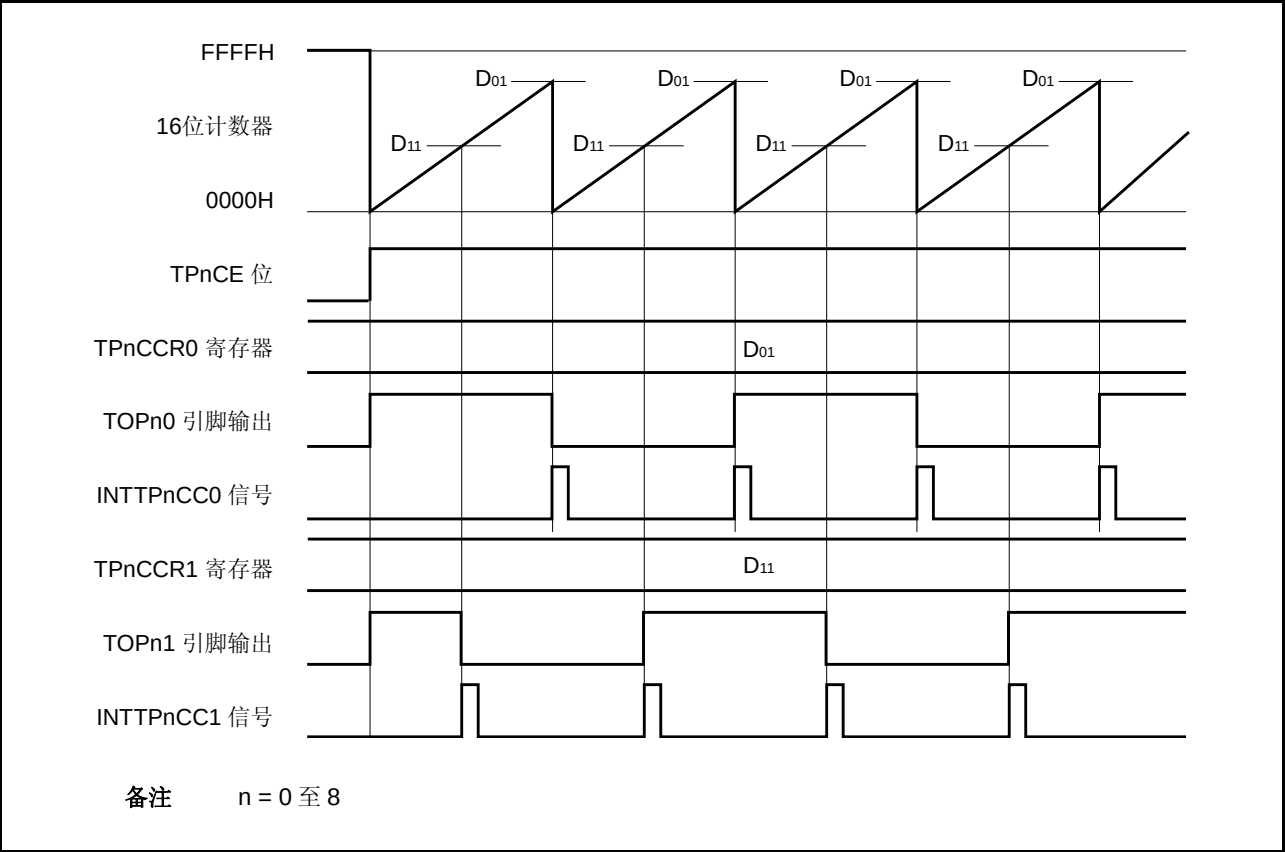
(d) TPnCCR1 寄存器的操作

图 7-6. TPnCCR1 寄存器的配置



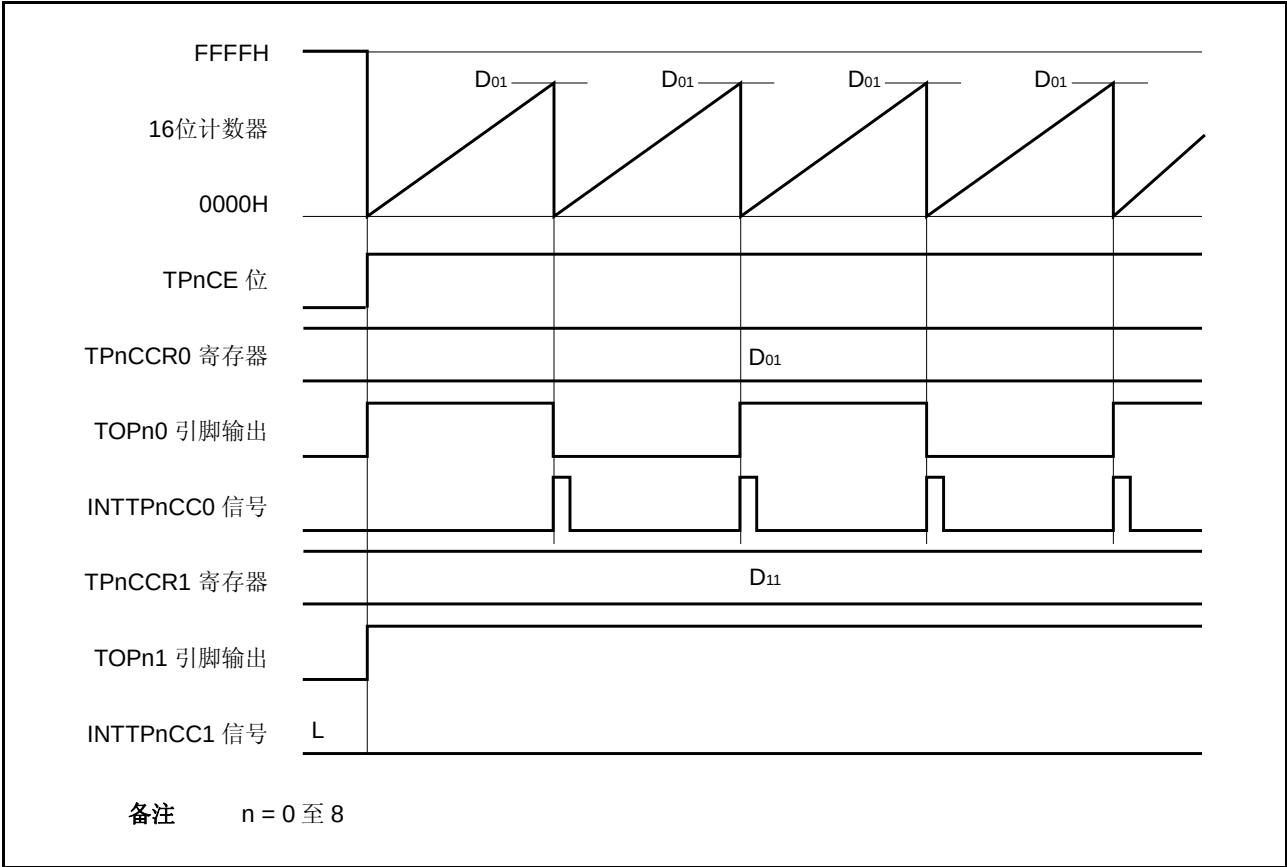
如果 TPnCCR1 寄存器的设定值小于 TPnCCR0 寄存器的设定值，则每个计数周期产生一次 INTTPnCC1 中断请求信号。同时，TOPn1 引脚的输出反相。
TOPn1 引脚输出一个方波，其周期与 TOPn0 引脚的输出周期相同。

图 7-7. 当 $D_{01} \geq D_{11}$ 时的时序图



如果 TPnCCR1 寄存器的设定值大于 TPnCCR0 寄存器的设定值，16 位计数器的计数值不会和 TPnCCR1 寄存器的值发生匹配。因此也就不会产生 INTTPnCC1 信号，TOPn1 引脚的输出也不会改变。

图 7-8. 当 $D_{01} < D_{11}$ 时的时序图



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

在外部事件计数器模式下，当 TPnCTL0.TPnCE 位置为 1 时，对外部事件计数输入信号的有效边沿进行计数，每次有效边沿的计数数量达到指定数目时，产生一个中断请求信号 (INTTPnCC0)。不能使用 TOPn0 引脚。

通常在外部事件计数器模式下，不使用 TPnCCR1 寄存器。

图 7-9. 外部事件计数模式的配置

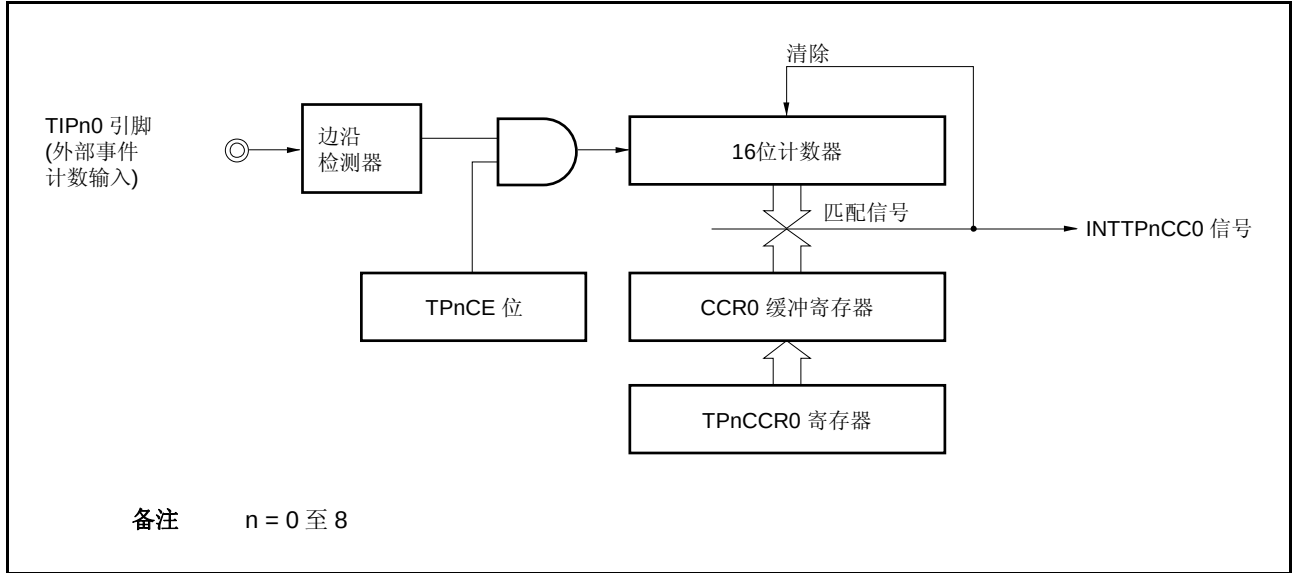
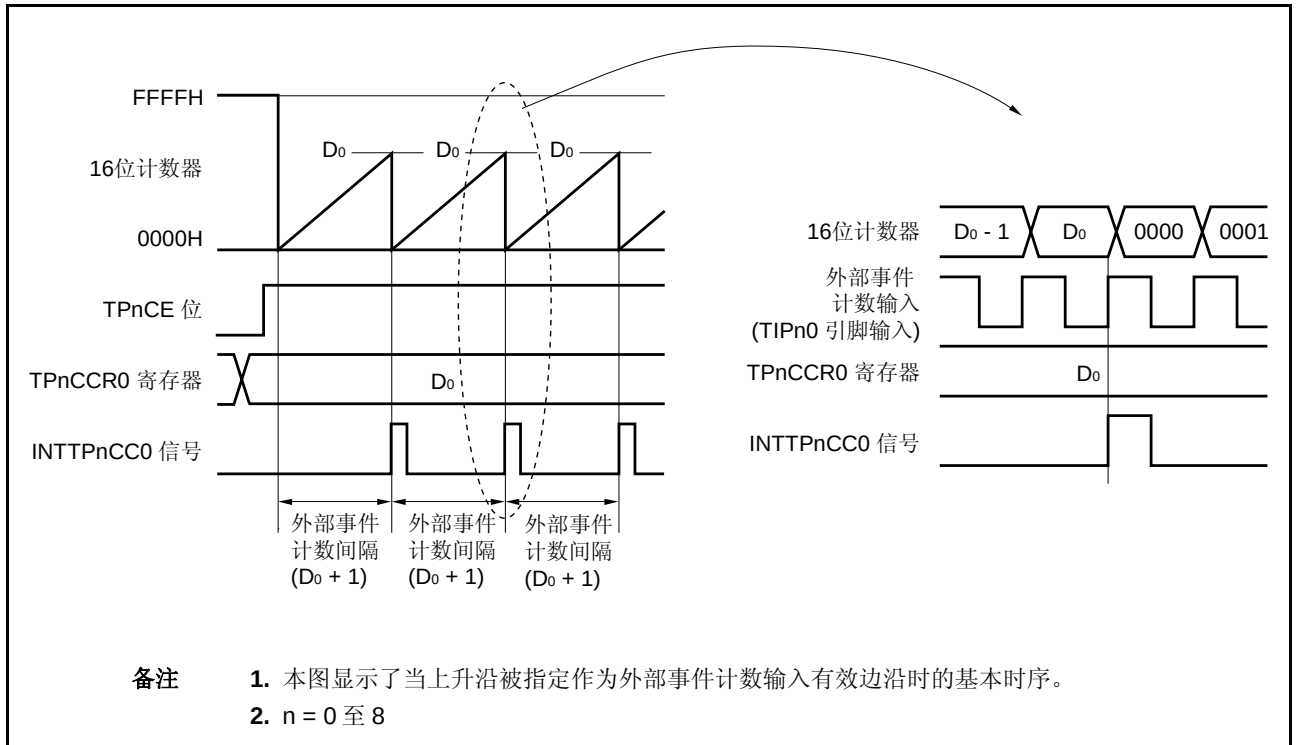


图 7-10. 外部事件计数模式的基本时序



当 TPnCE 位置为 1 时，16 位计数器的值从 FFFFH 清为 0000H。计数器对检测到的每一次外部事件计数输入的有效边沿进行计数。此外，TPnCCR0 寄存器的设定值被传送给 CCR0 缓冲寄存器。

当 16 位计数器的计数值和 CCR0 缓冲寄存器的数值相匹配时，16 位计数器清零为 0000H，同时生成一个比较匹配中断请求信号 (INTTPnCC0)。

每当外部事件计数输入的有效边沿数量达到 (TPnCCR0 寄存器设定值+1) 时，就会生成 INTTPnCC0 (比较匹配中断请求) 信号。

图 7-11. 外部事件计数模式下的寄存器操作设置 (1/2)

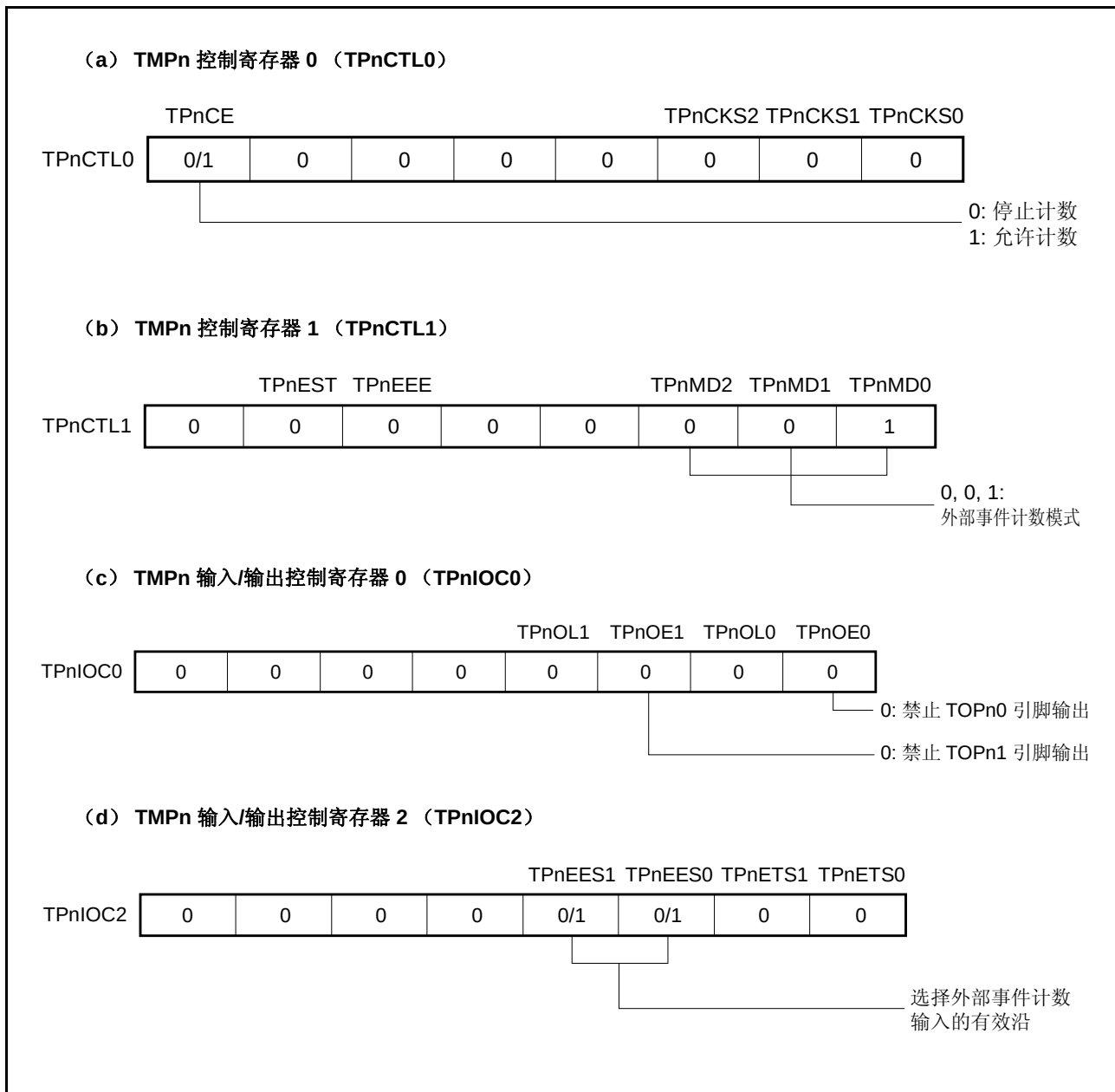


图 7-11. 外部事件计数模式下的寄存器操作设置 (2/2)

(e) TMPn 计数器读取缓冲寄存器 (TPnCNT)

可以通过读取 TPnCNT 寄存器，读取 16 位计数器的计数值。

(f) TMPn 捕获/比较寄存器 0 (TPnCCR0)

如果 TPnCCR0 寄存器的值设定为 D_0 ，当外部事件计数值达到 (D_0+1) 时，计数器清 0，同时产生一个比较匹配中断请求信号 (INTTPnCC0)。

(g) TMPn 捕获/比较寄存器 1 (TPnCCR1)

通常在外部事件计数器模式中不使用 TPnCCR1 寄存器。但是，TPnCCR1 寄存器的设定值仍然传送给 CCR1 缓冲寄存器。当 16 位计数器的计数值和 CCR1 缓冲寄存器的值匹配时，会产生一个比较匹配中断请求信号 (INTTPnCC1)。

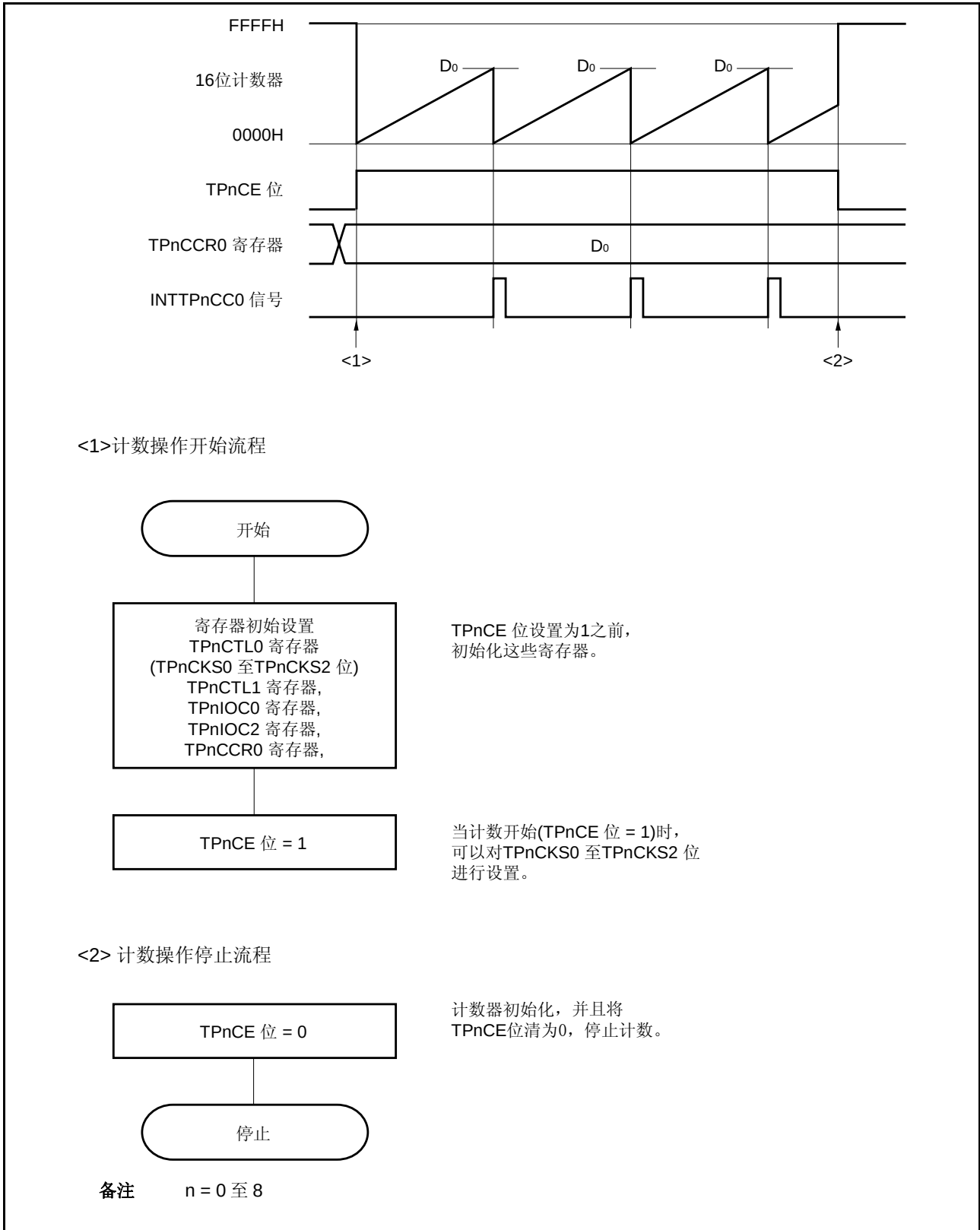
所以，需要使用中断屏蔽标志 (TPnCCMK1) 来屏蔽该中断请求信号。

注意事项 当外部时钟用作计数时钟时，外部时钟只能从 TIPn0 引脚输入。此时，将 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位清为 00 (捕获触发输入 (TIPn0 引脚)：无边沿检测)。

- 备注**
1. 在外部事件计数模式下，不使用 TMPn 输入/输出控制寄存器 1 (TPnIOC1) 和 TMPn 选项寄存器 0 (TPnOPT0)。
 2. $n = 0$ 至 8

(1) 外部事件计数模式的操作流程

图 7-12. 外部事件计数模式下的软件处理流程

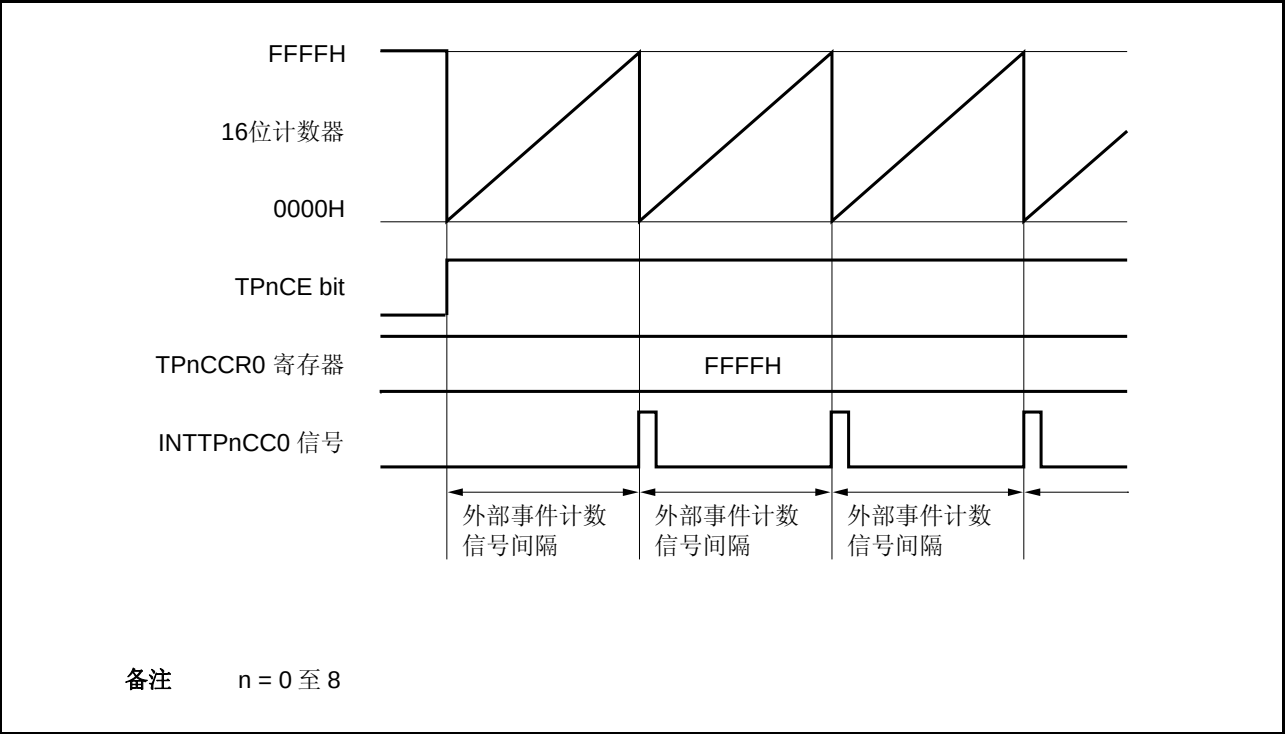


(2) 外部事件计数模式的操作时序

- 注意事项 1. 外部事件计数模式下，不要将 TPnCCR0 寄存器设置为 0000H。
2. 外部事件计数模式下，禁止使用定时器输出。如果使用外部事件计数输入来进行定时器输出，应设置为间隔定时器模式，并且选择外部事件计数输入允许的操作作为计数时钟（TPnCTL1.TPnMD2 至 TPnCTL1.TPnMD0 位= 000，TPnCTL1.TPnEEE 位= 1）。

(a) 如果 TPnCCR0 寄存器设置为 FFFFH 时的操作

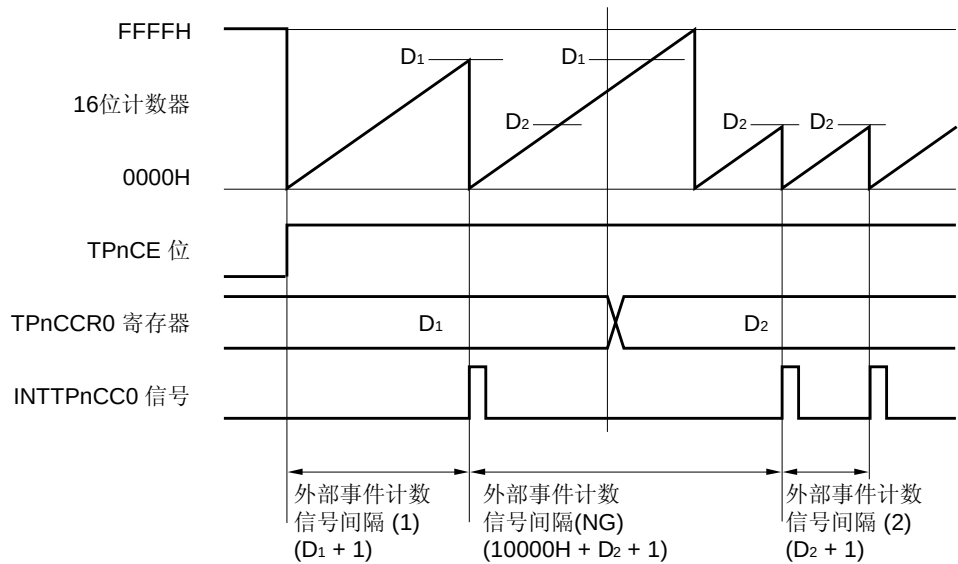
如果 TPnCCR0 寄存器设置为 FFFFH，16 位计数器每次检测到的外部事件输入信号有效边沿时，都计数到 FFFFH。当下一个计数到来时，16 位计数器同步清为 0000H，并且产生 INTTPnCC0 信号。此时，不设置 TPnOPT0.TPnOVF 位。



(b) 重写 TPnCCR0 寄存器的注意事项

要将 TPnCCR0 寄存器的值改为比当前值较小的值，应停止计数，然后改变其设定值。

如果在计数器工作期间，将 TPnCCR0 寄存器的值重写为比当前值较小的值，16 位计数器可能会溢出。



备注 n = 0 至 8

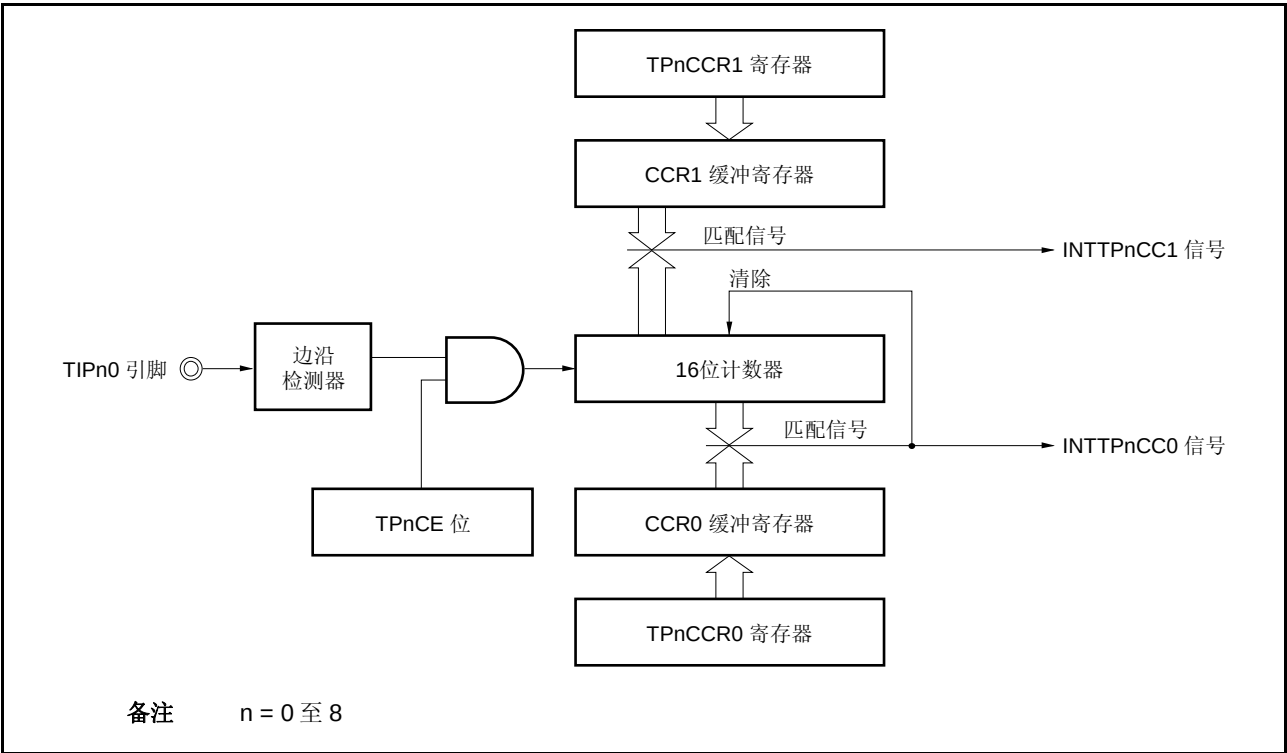
当 16 位计数器计数值大于 D2 而小于 D1 时，如果将 TPnCCR0 寄存器的值从 D1 变为 D2，则在重写 TPnCCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。因而，与 16 位计数器相比较的值会是 D2。

但是，由于计数值已经超过了 D2，所以 16 位计数器就会向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值与 D2 匹配时，会生成 INTTPnCC0 信号。

因此，可能不会在原先预计的“(D1+1)次”或“(D2+1)次”有效边沿生成 INTTPnCC0 中断请求信号，而可能在“(10000H+D2+1)次”有效边沿生成中断请求信号。

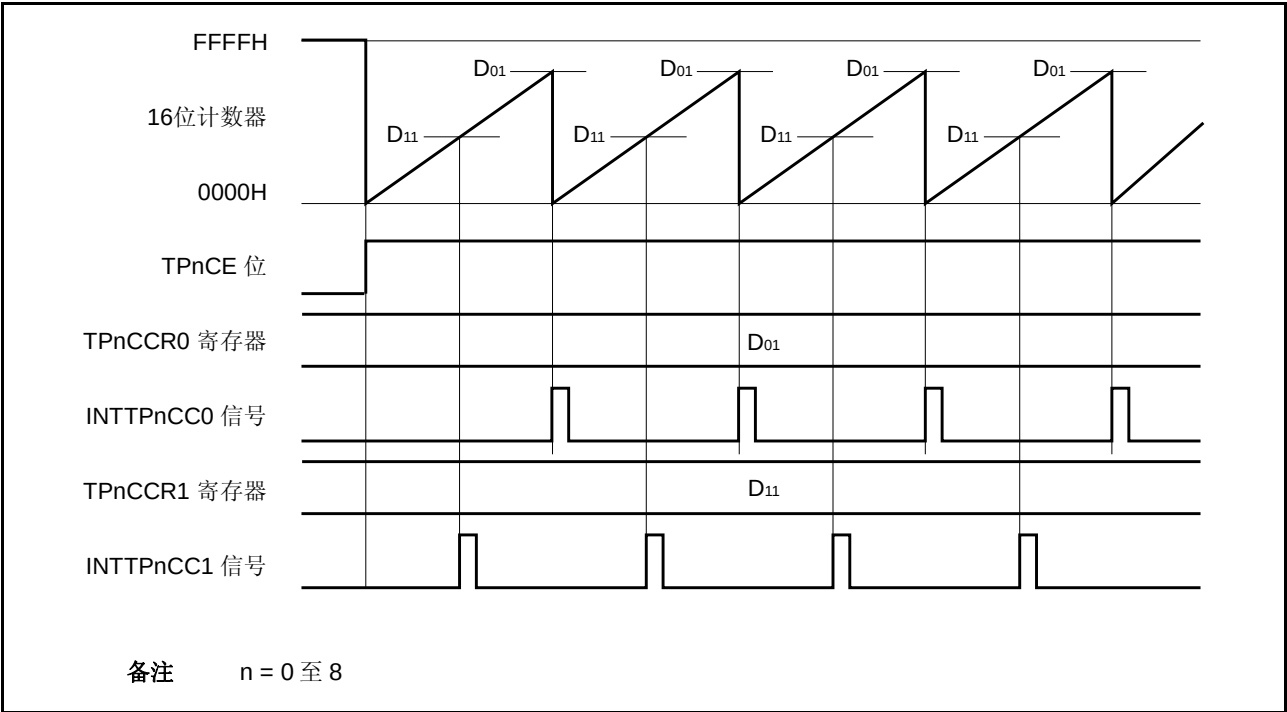
(c) TPnCCR1 寄存器的操作

图 7-13. TPnCCR1 寄存器的配置



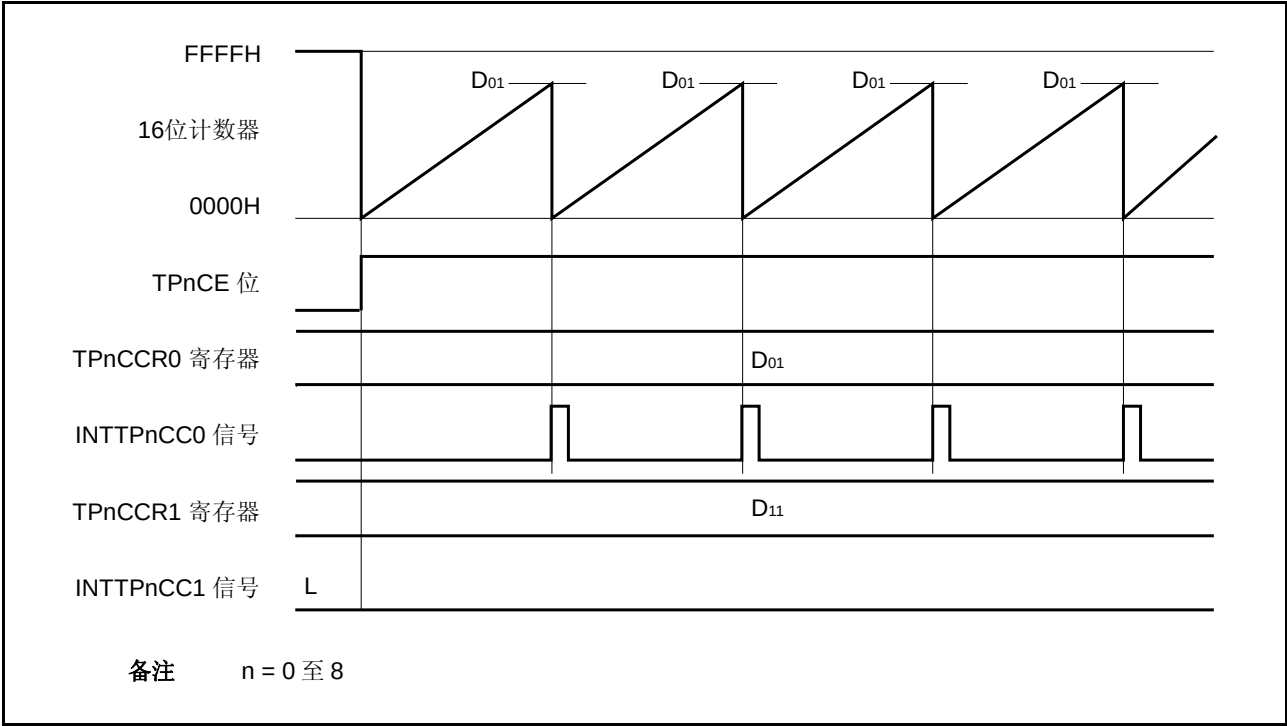
如果 TPnCCR1 寄存器的设定值小于 TPnCCR0 寄存器的设定值，则每个计数周期都产生一次 INTTPnCC1 中断请求信号。

图 7-14. 当 $D_{01} \geq D_{11}$ 时的时序图



如果 TPnCCR1 寄存器的设定值大于 TPnCCR0 寄存器的设定值，16 位计数器的计数值不会和 TPnCCR1 寄存器的值发生匹配。因此也就不会产生 INTTPnCC1 中断请求信号。

图 7-15. 当 D01 < D11 时的时序图



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

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

不使用外部触发，而是使用软件触发，同样可以得到输出脉冲。当使用软件触发时，TOPn0 引脚输出一个方波，其半周期等于输出 PWM 的一个周期。

图 7-16. 外部触发脉冲输出模式的配置

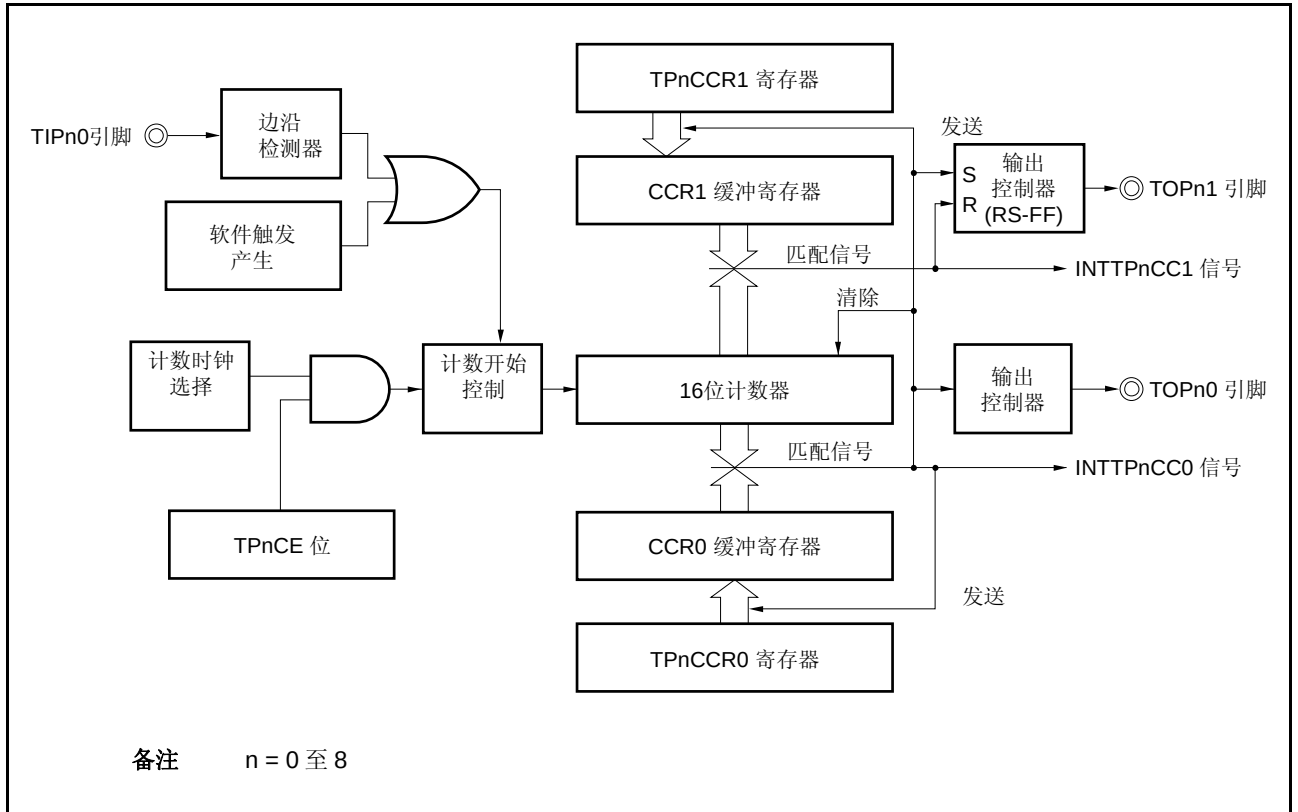
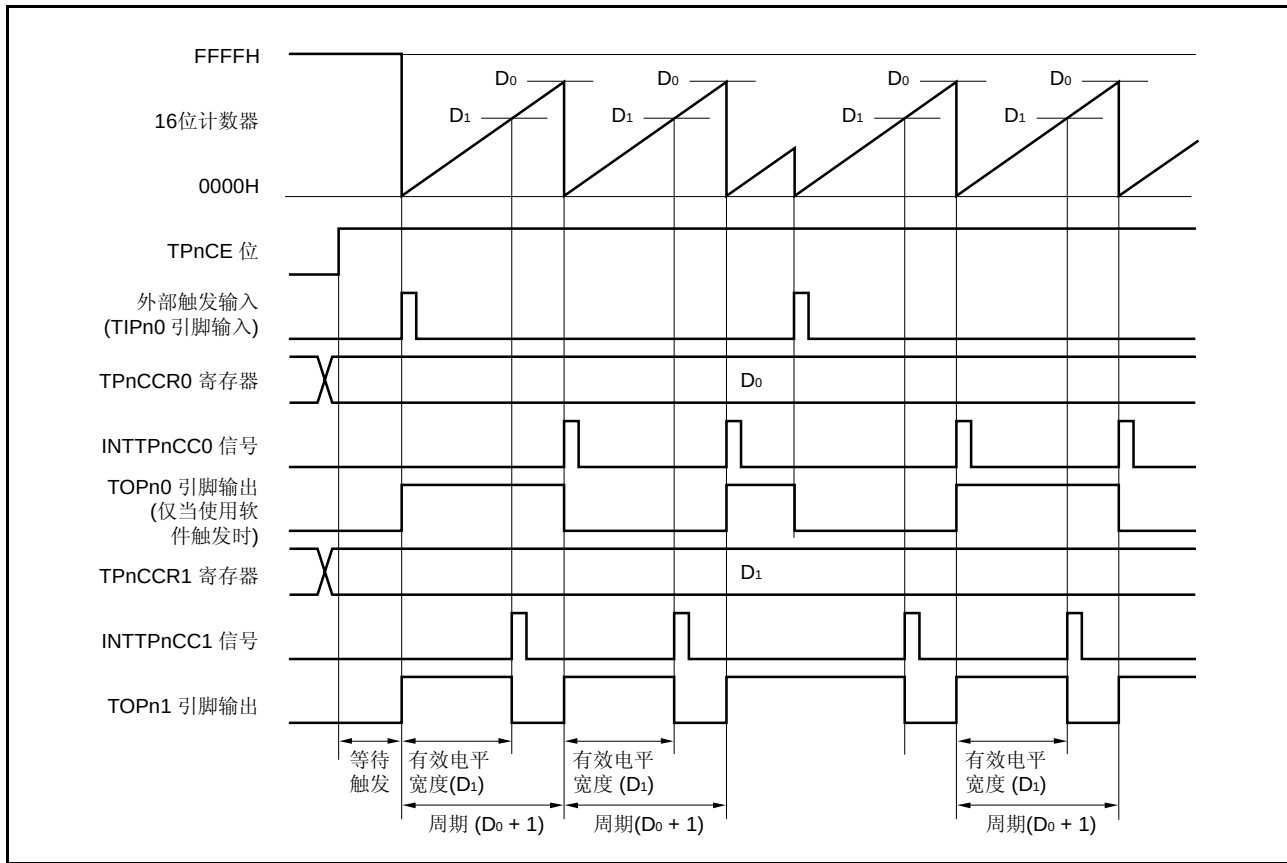


图 7-17. 外部触发脉冲输出模式的基本时序



当 TPNCE 位设置为 1 时，16 位定时器/事件计数器 P 等待触发。当触发产生时，16 位计数器从 FFFFH 清零为 0000H，同时开始计数，并从 TOPn1 引脚输出一个 PWM 波形。如果在计数运行中再次产生触发，则计数器清为 0000H 并重新开始计数。（TOPn0 引脚输出反相。在出现触发时，不管当前状态如何（高/低），TOPn1 引脚都会输出一个高电平。

PWM 波形的有效电平宽度、周期、占空比的计算如下。

有效电平宽度 = (TPnCCR1 寄存器的设置值) × 计数时钟周期

周期 = (TPnCCR0 寄存器的设置值 + 1) × 计数时钟周期

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

当 16 位计数器在其计数值与 CCR0 缓冲寄存器的值匹配之后，进行下一次计数时就会同步产生比较匹配中断请求信号 INTTPnCC0，且 16 位计数器清零为 0000H。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTPnCC1。

当 16 位计数器的计数值与 CCRm 缓冲寄存器的值匹配，而且 16 位计数器清零为 0000H 时，TPnCCRm 寄存器的设定值就被传送到 CCRm 缓冲寄存器。

外部触发输入信号的有效边沿，或将软件触发 (TPnCTL1.TPnEST 位) 设为 1，都可以用作触发信号。

备注 n = 0 至 8, m = 0, 1

图 7-18. 外部触发脉冲输出模式的寄存器设置 (1/2)

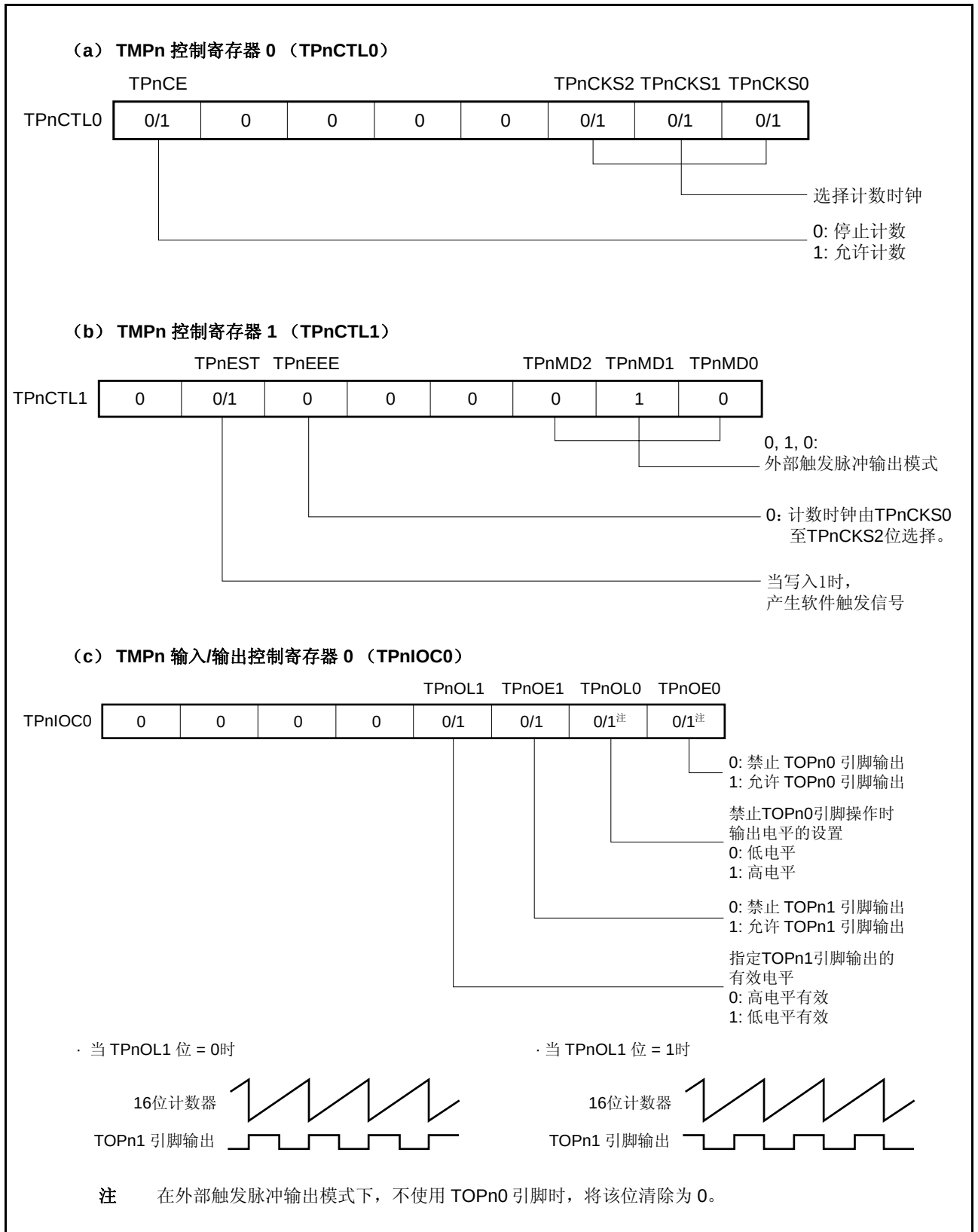
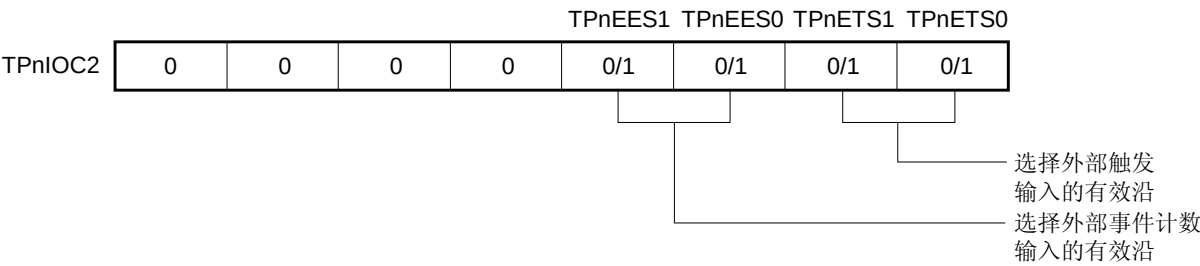


图 7-18. 外部触发脉冲输出模式的寄存器设置（2/2）

<R>

(d) TMPn 输入/输出控制寄存器 2（TPnIOC2）



(e) TMPn 计数器读取缓冲寄存器（TPnCNT）

通过读取 TPnCNT 寄存器，可以读出 16 位计数器的值。

(f) TMPn 捕获/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 ，则 PWM 波形的周期和有效电平的计算如下：

周期 = $(D_0 + 1) \times$ 计数时钟周期

有效电平宽度 = $D_1 \times$ 计数时钟周期

- 备注**
1. 在外部触发脉冲输出模式下，不使用 TMPn 输入/输出控制寄存器 1（TPnIOC1）和 TMPn 选项寄存器 0（TPnOPT0）。
 2. $n = 0$ 至 8

(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）

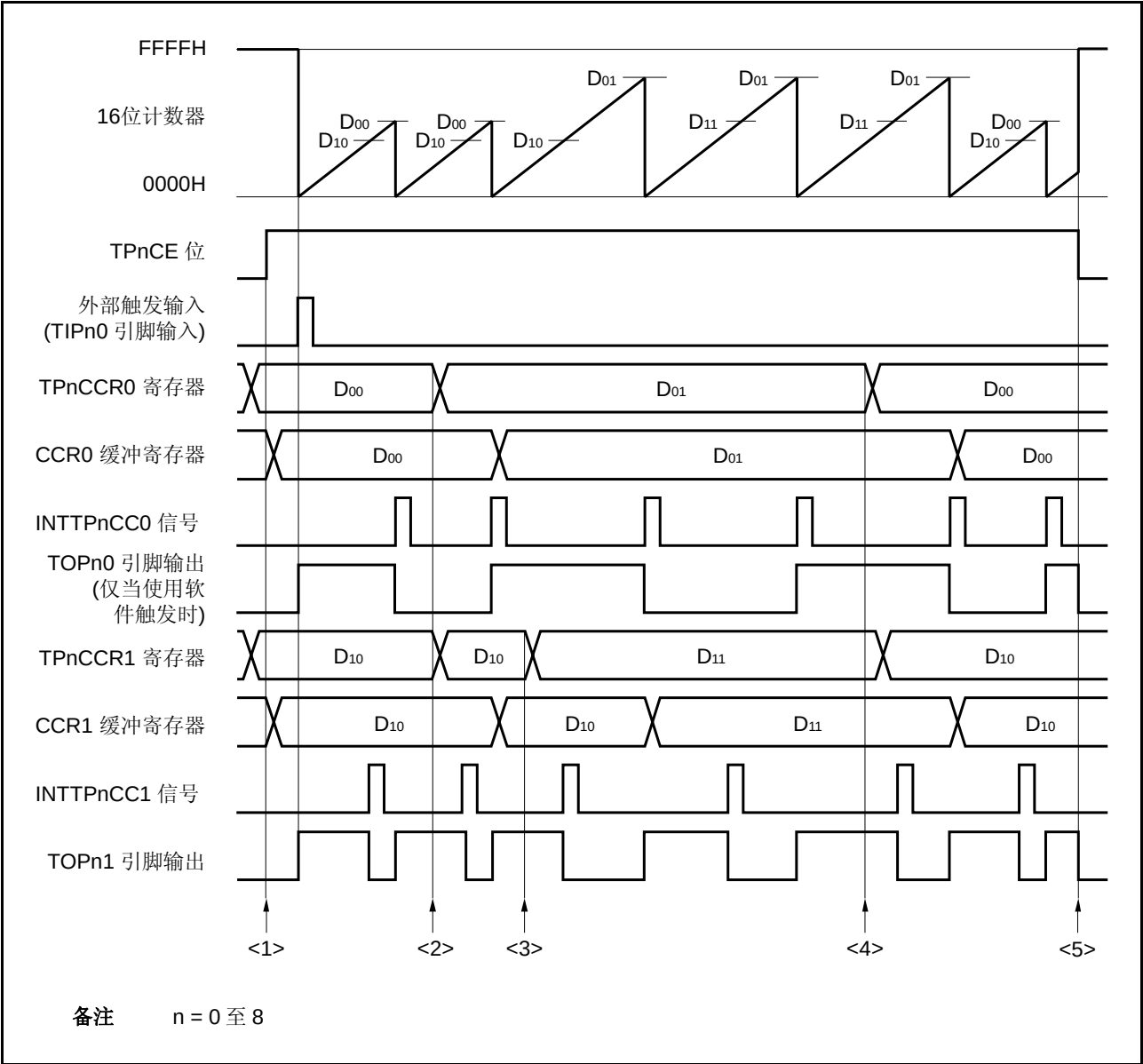
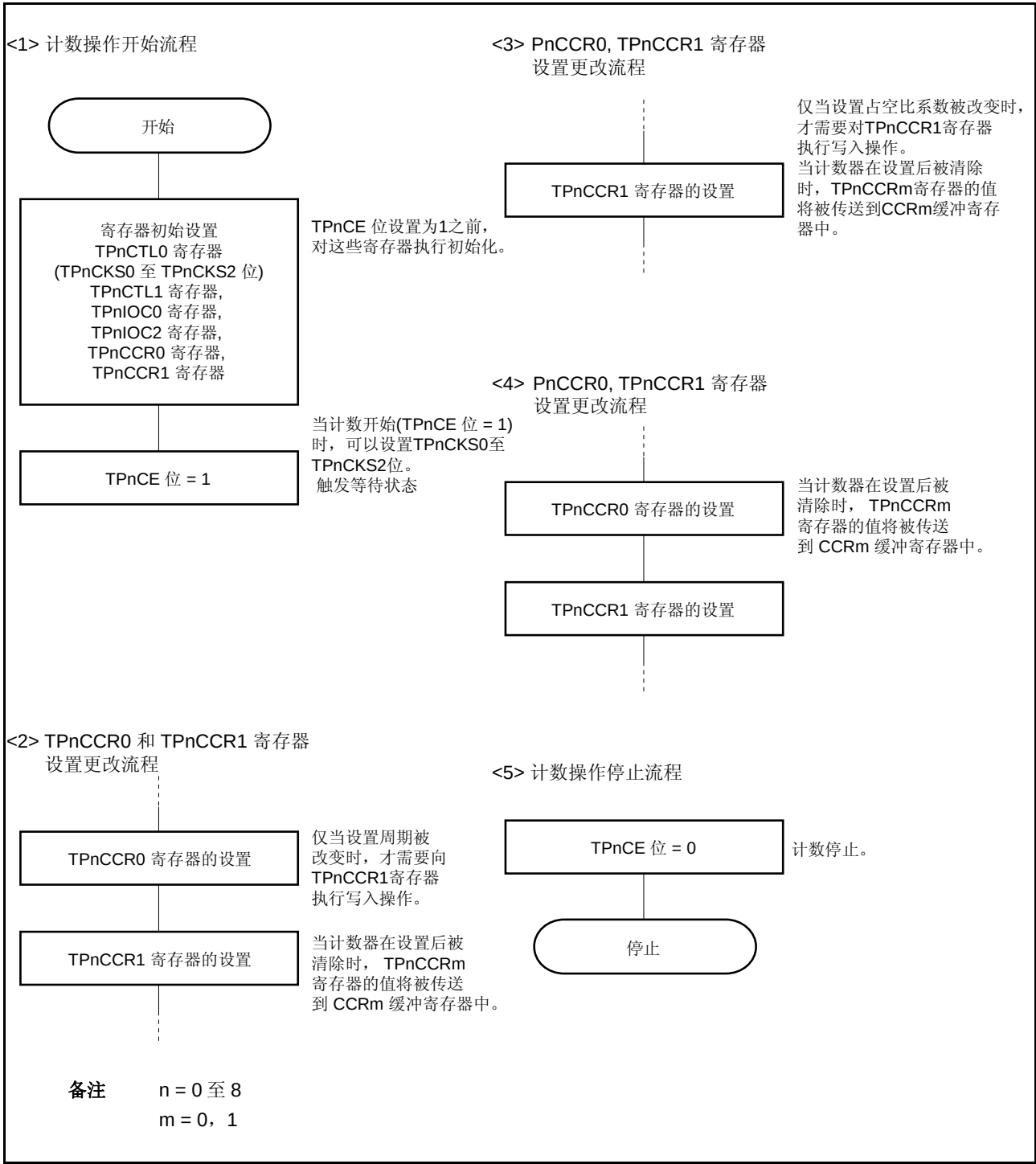


图 7-19. 外部触发脉冲输出模式的软件处理流程 (2/2)

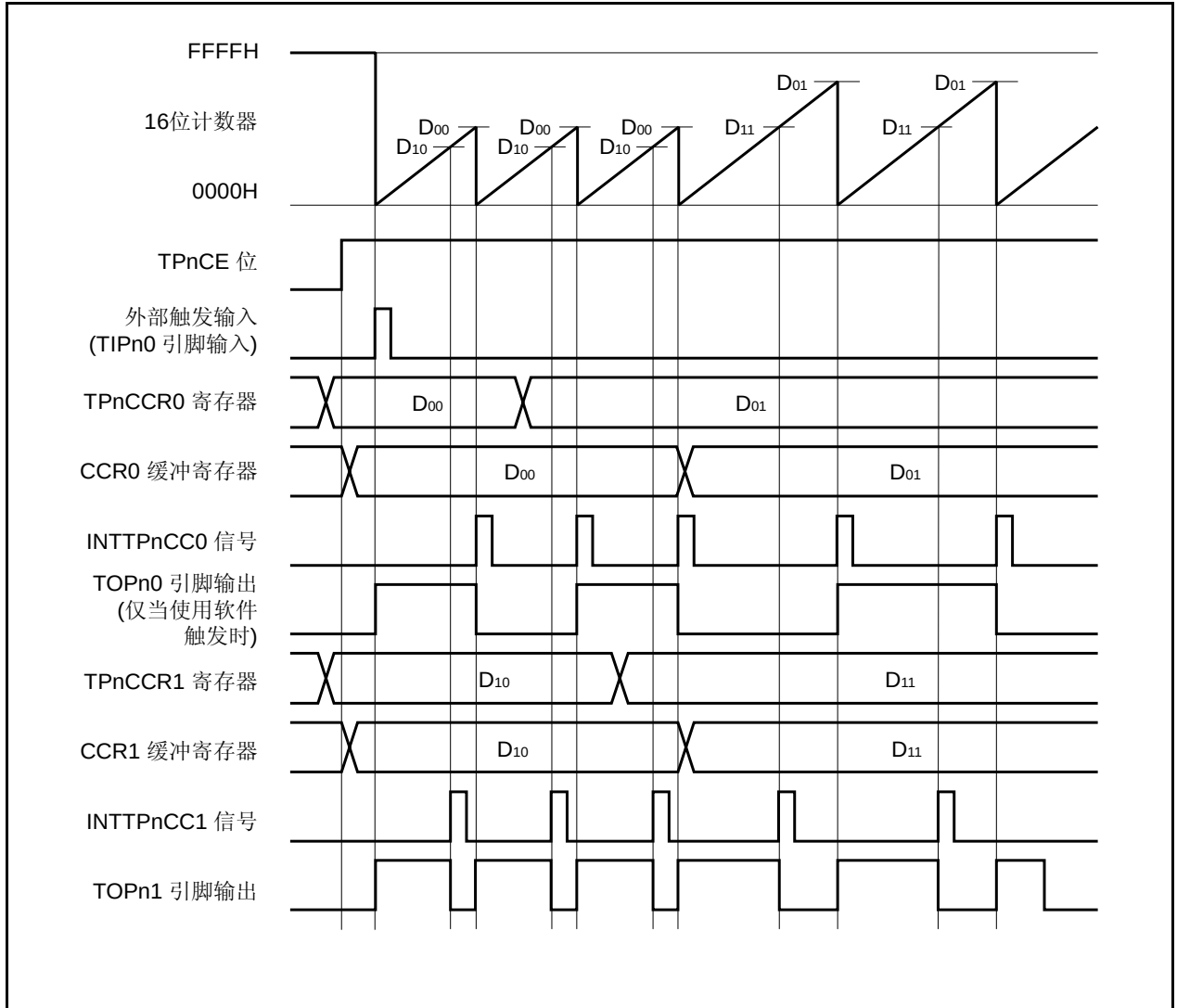


（2）外部触发脉冲输出模式的操作时序

（a）操作中改变脉冲宽度的注意事项

计数器计数过程中，若要改变 PWM 波形，应重新写入 TPnCCR1 寄存器。

当检测到 INTTPnCC0 信号以后，在 TPnCCR1 寄存器写入操作之后重写 TPnCCRm 寄存器。



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

如果 PWM 波形的周期和有效电平宽度都要改变，应先将周期设定到 TPnCCR0 寄存器，然后将有效电平宽度设定到 TPnCCR1 寄存器。

如果只改变 PWM 波形的周期，应先把周期设定给 TPnCCR0 寄存器，然后将相同值（原值）写入 TPnCCR1 寄存器。

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

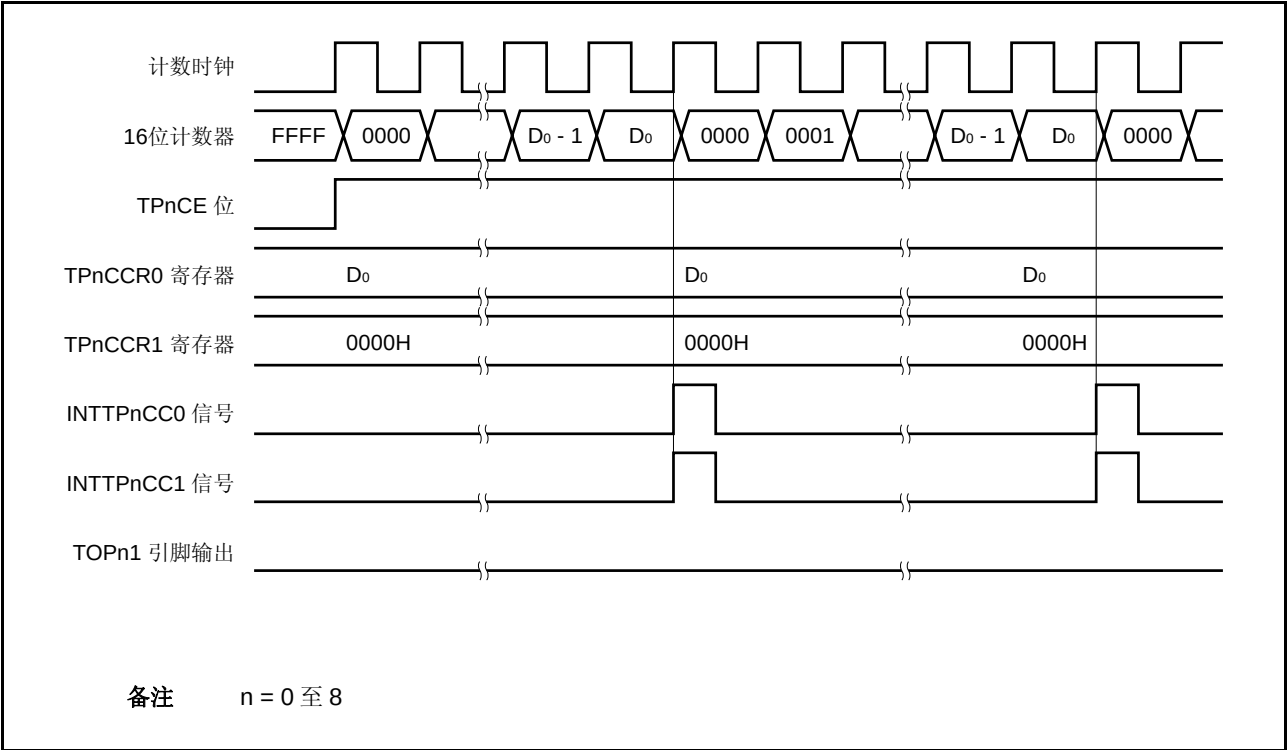
当数据写入 TPnCCR1 寄存器之后，在对 16 位计数器清零的同时，写入到 TPnCCRM 寄存器的值被传送到 CCRm 缓冲寄存器，并作为与 16 位计数器进行比较的值。

在写入 TPnCCR1 寄存器一次之后，要再次写 TPnCCR0 或 TPnCCR1 寄存器，应该在 INTTPnCC0 信号产生之后进行。否则，CCRM 缓冲寄存器的值可能不确定，因为把数据从 TPnCCRM 寄存器传送到 CCRm 缓冲寄存器的时序会和写入 TPnCCRM 寄存器的时序发生冲突。

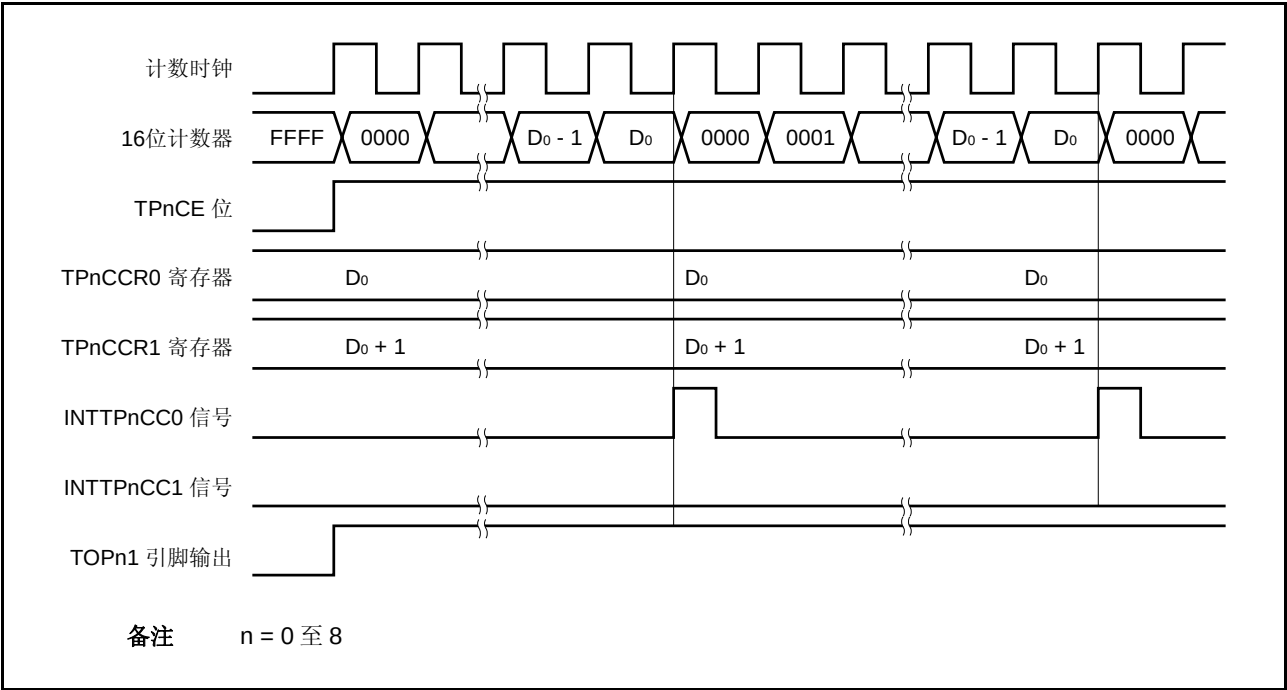
备注 $n = 0$ 至 8
 $m = 0, 1$

(b) PWM 波形的 0%和 100% 输出

为了输出 0%的波形，需要将 TPnCCR1 寄存器设置为 0000H。如果 TPnCCR0 寄存器的设定值为 FFFFH，则会在每个周期产生 INTTPnCC1 中断请求信号。

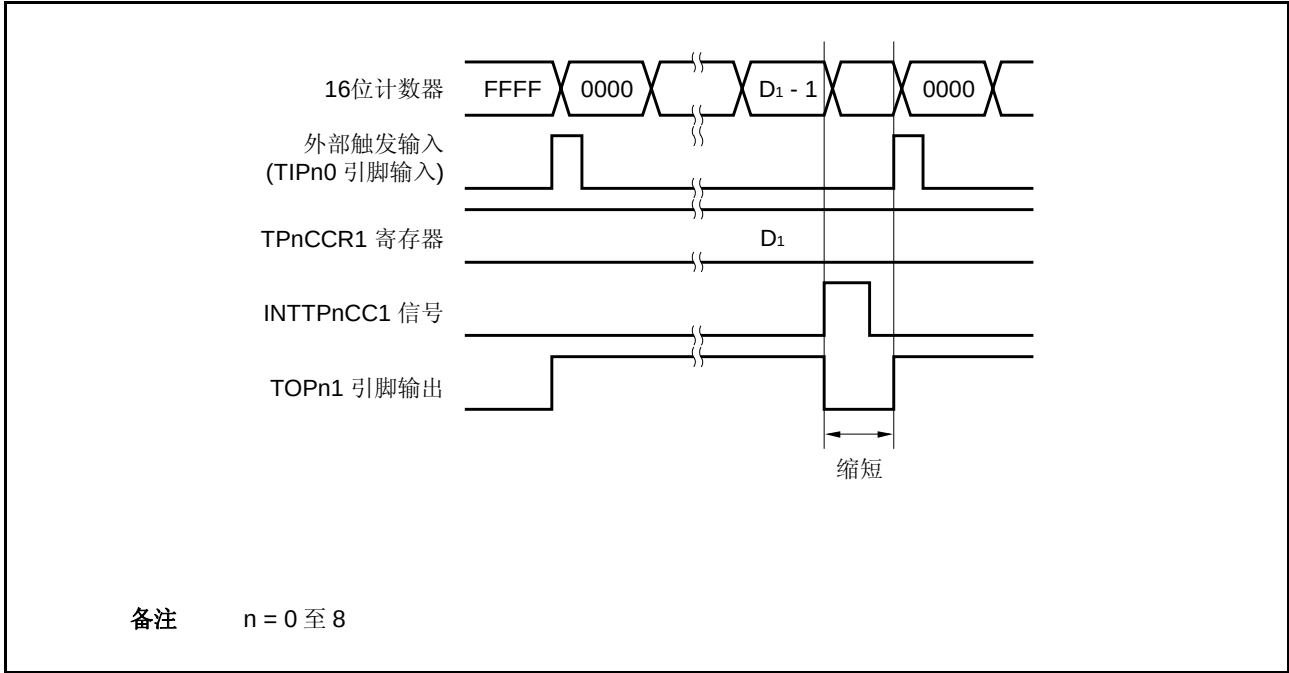


为了输出 100%的波形，需要将 TPnCCR1 寄存器的值设为：（TPnCCR0 寄存器的设定值+1）。如果 TPnCCR0 寄存器的设定值为 FFFFH，则不会产生 100%波形输出。

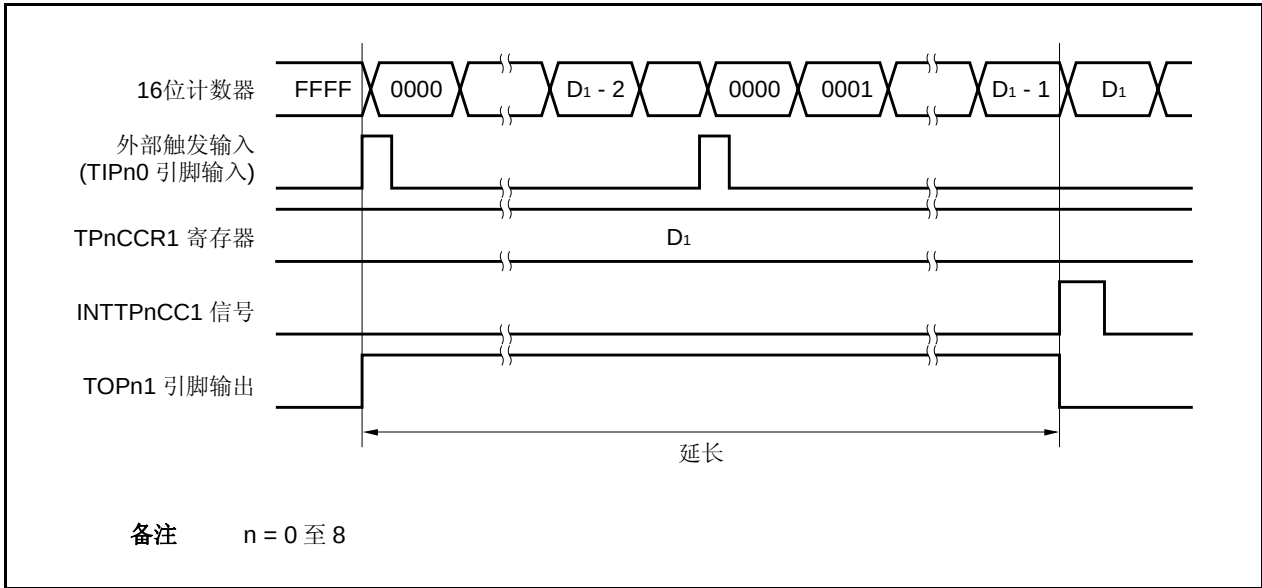


(c) 触发检测和 TPnCCR1 寄存器匹配之间的冲突

如果在 INTTPnCC1 信号产生后，立刻检测到触发信号，则 16 位计数器立刻清零为 0000H，TOPn1 引脚的输出信号生效，计数器继续计数。结果，PWM 波形的无效时段被缩短。

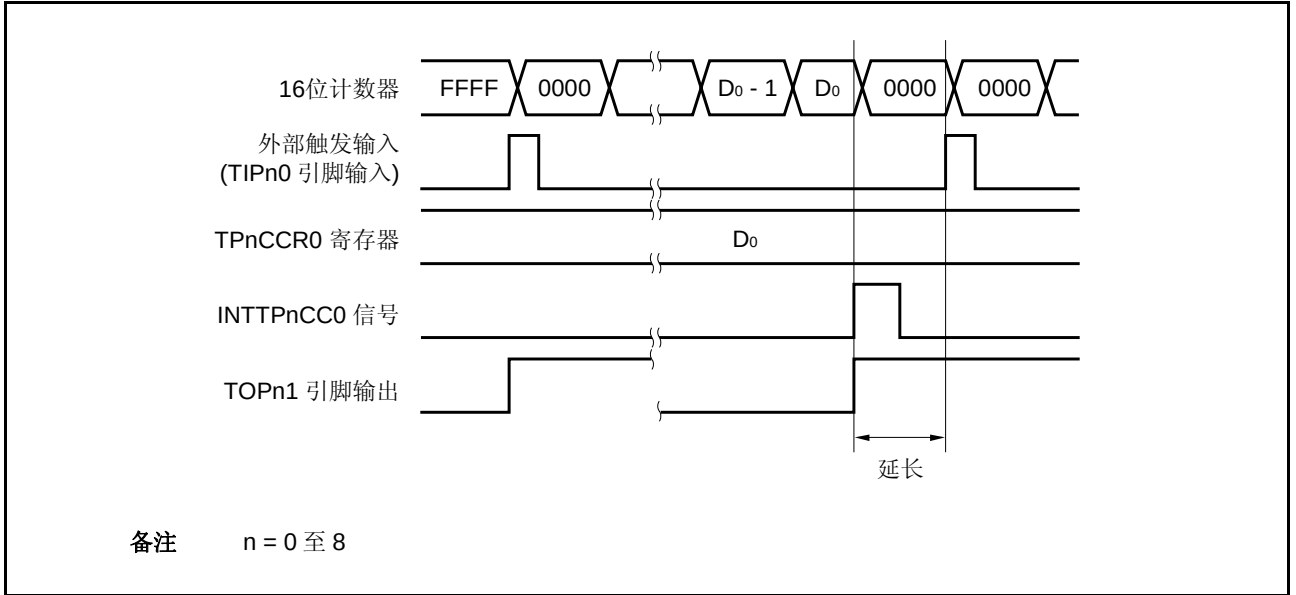


如果紧邻在 INTTPnCC1 信号产生之前检测到触发信号，则不会产生 INTTPnCC1 信号，而 16 位计数器清零为 0000H，计数继续进行。TOPn1 引脚的输出信号仍然保持有效。结果，PWM 波形的有效时段被延长。

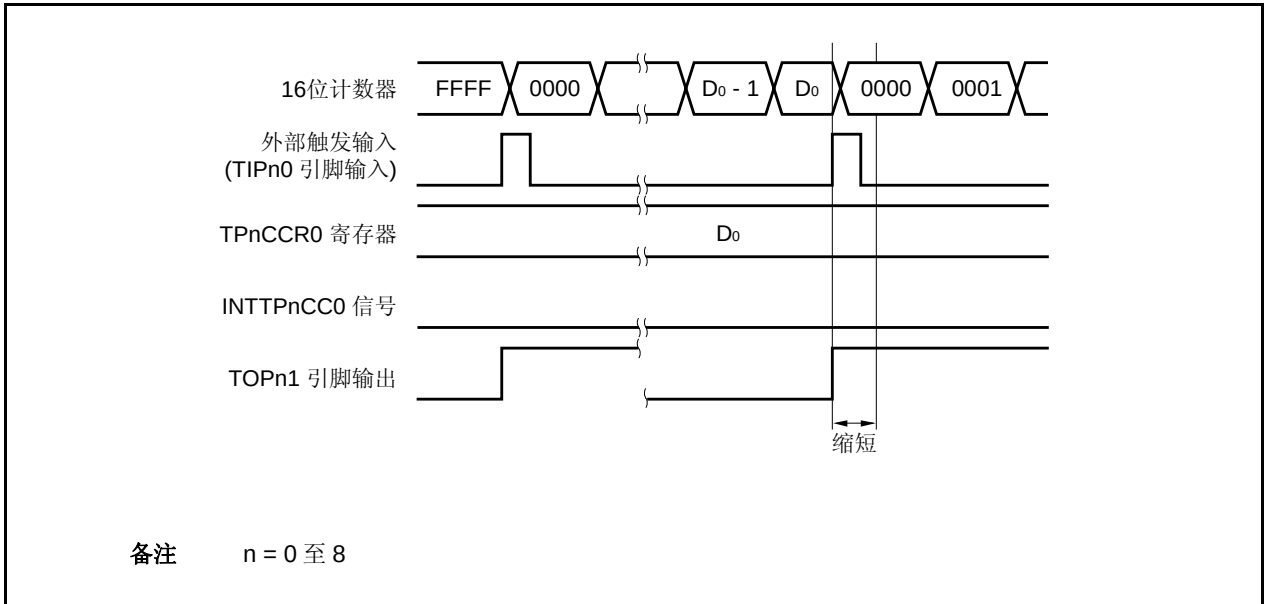


(d) 触发检测和 TPnCCR0 寄存器匹配之间的冲突

如果在 INTTPnCC0 信号产生后，立刻检测到触发信号，则 16 位计数器立刻清零为 0000H，并继续计数。因此，TOPn1 引脚的有效时段被延长，延长长度为 INTTPnCC0 信号产生时刻至检测到触发信号时刻的时间间隔。

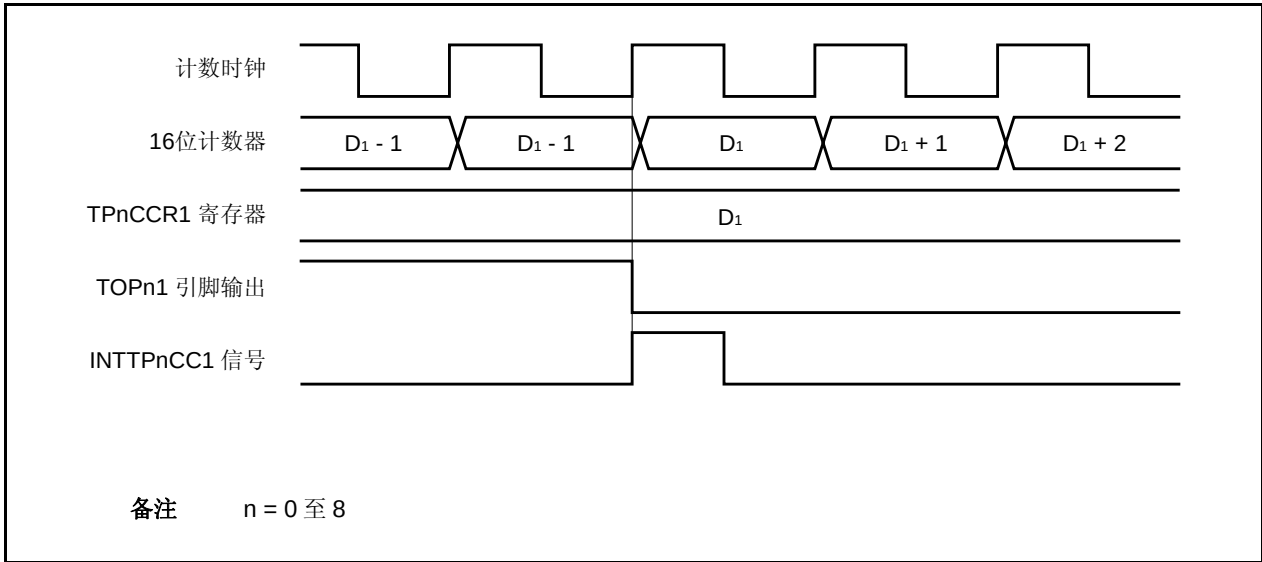


如果紧邻在 INTTPnCC0 信号产生之前检测到触发信号，则不会产生 INTTPnCC0 信号。16 位计数器清零为 0000H，TOPn1 引脚输出有效，计数器继续计数。结果，PWM 波形的无效时段被缩短。



(e) 比较匹配中断请求信号 (INTTPnCC1) 的发生时序

在外部触发脉冲输出模式下，INTTPnCC1 信号的发生时序与其它情况下 INTTPnCC1 信号的发生时序不同；当 16 位计数器的计数值与 TPnCCR1 寄存器的值匹配时，产生 INTTPnCC1 中断请求信号。



通常在 16 位计数器的计数值与 TPnCCR1 寄存器的值匹配之后，在下一次计数时钟同步发生 INTTPnCC1 信号。

但是，在外部触发脉冲输出模式下，INTTPnCC1 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOPn1 引脚输出信号变化的时序。

7.5.4 单次触发脉冲输出模式（TPnMD2 至 TPnMD0 位=011）

在单次触发脉冲输出模式下，当 TPnCTL0.TPnCE 位置为 1 时，16 位定时器/事件计数器 P 等待触发。在检测到外部触发输入的有效边沿时，16 位定时器/事件计数器 P 开始计数，同时从 TOPn1 引脚输出一个单次触发脉冲。

除外部触发外，也可以产生软件触发信号来输出该脉冲。在使用软件触发时，TOPn0 引脚在 16 位计数器计数时输出有效电平，在计数器停止（等待触发）时输出无效电平。

图 7-20. 单次触发脉冲输出模式的配置

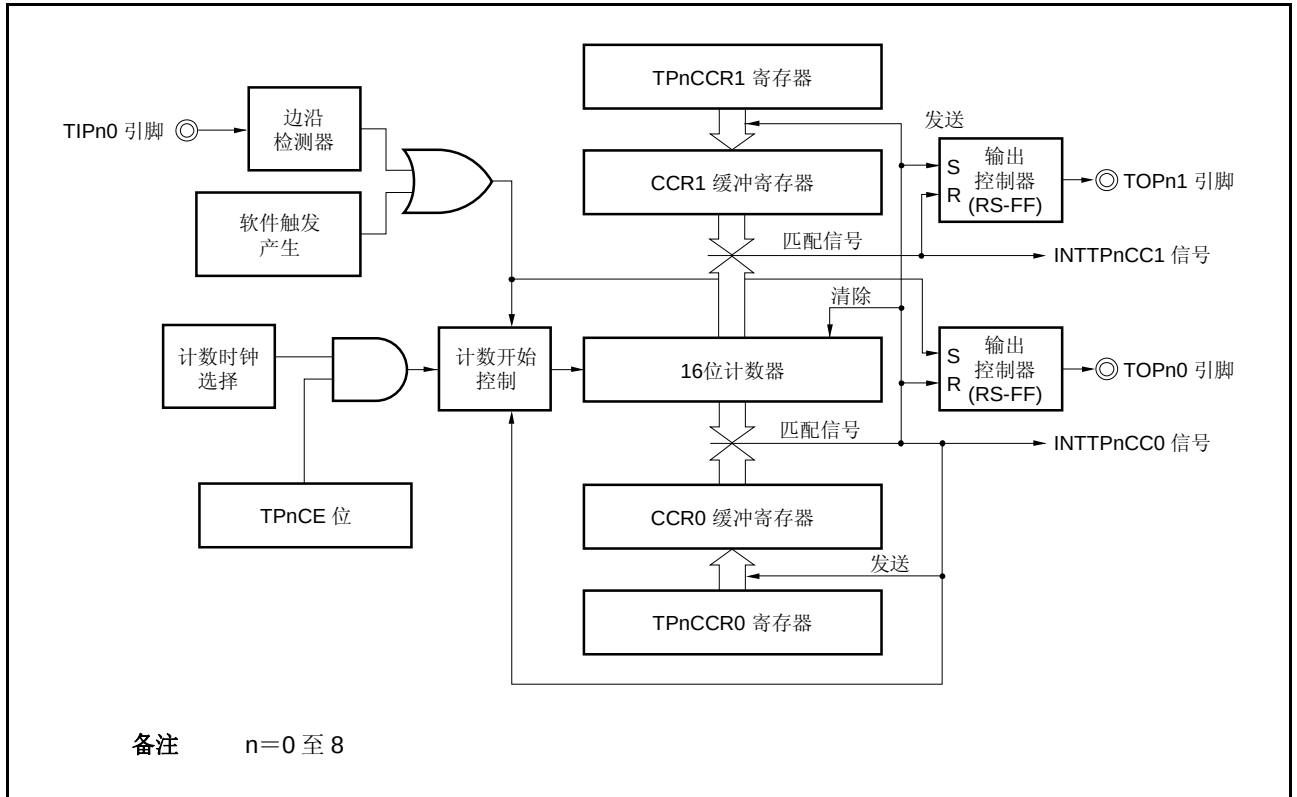
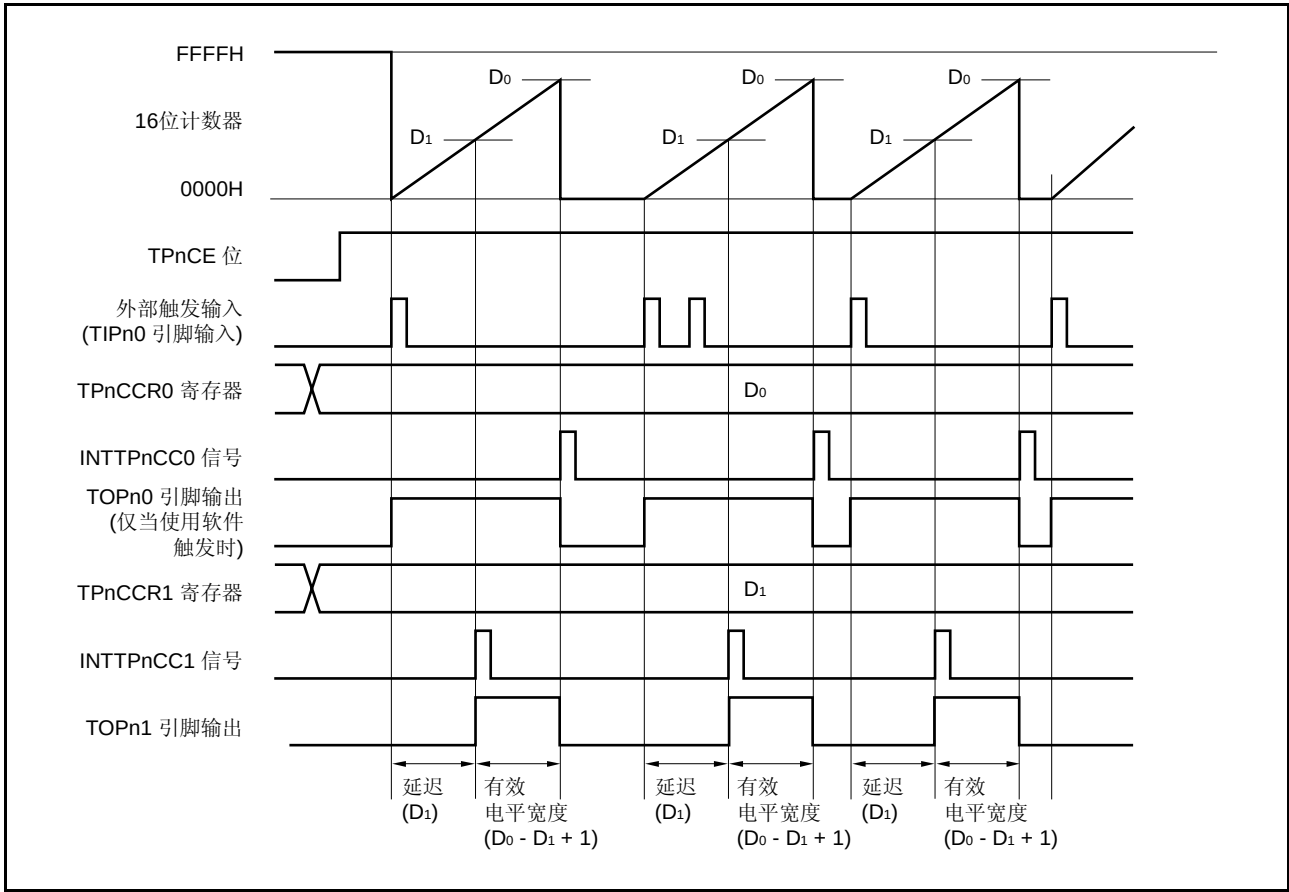


图 7-21. 单次触发脉冲输出模式的基本时序



当 TPnCE 位置为 1 时，16 位定时器/事件计数器 P 等待触发。在触发信号产生时，16 位计数器从 FFFFH 清零为 0000H，开始进行计数，并从 TOPn1 引脚输出一个单次触发脉冲。在输出单次触发脉冲后，16 位计数器设置为 FFFFH，计数停止，并重新等待触发。如果在单次触发脉冲输出期间再次产生触发信号，则忽略该触发信号。

单次触发脉冲的输出延迟时间和有效电平宽度可计算如下：

输出延迟时间 = (TPnCCR1 寄存器的设置值) × 计数时钟周期

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

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配之后，产生比较匹配中断请求信号 INTTPnCC0。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTPnCC1。

外部触发输入信号的有效边沿或将软件触发（TPnCTL1.TPnEST 位）信号设置为 1，都可用作触发信号。

备注 n=0 至 8
 m = 0, 1

图 7-22. 单次触发脉冲输出模式时的寄存器设置 (1/2)

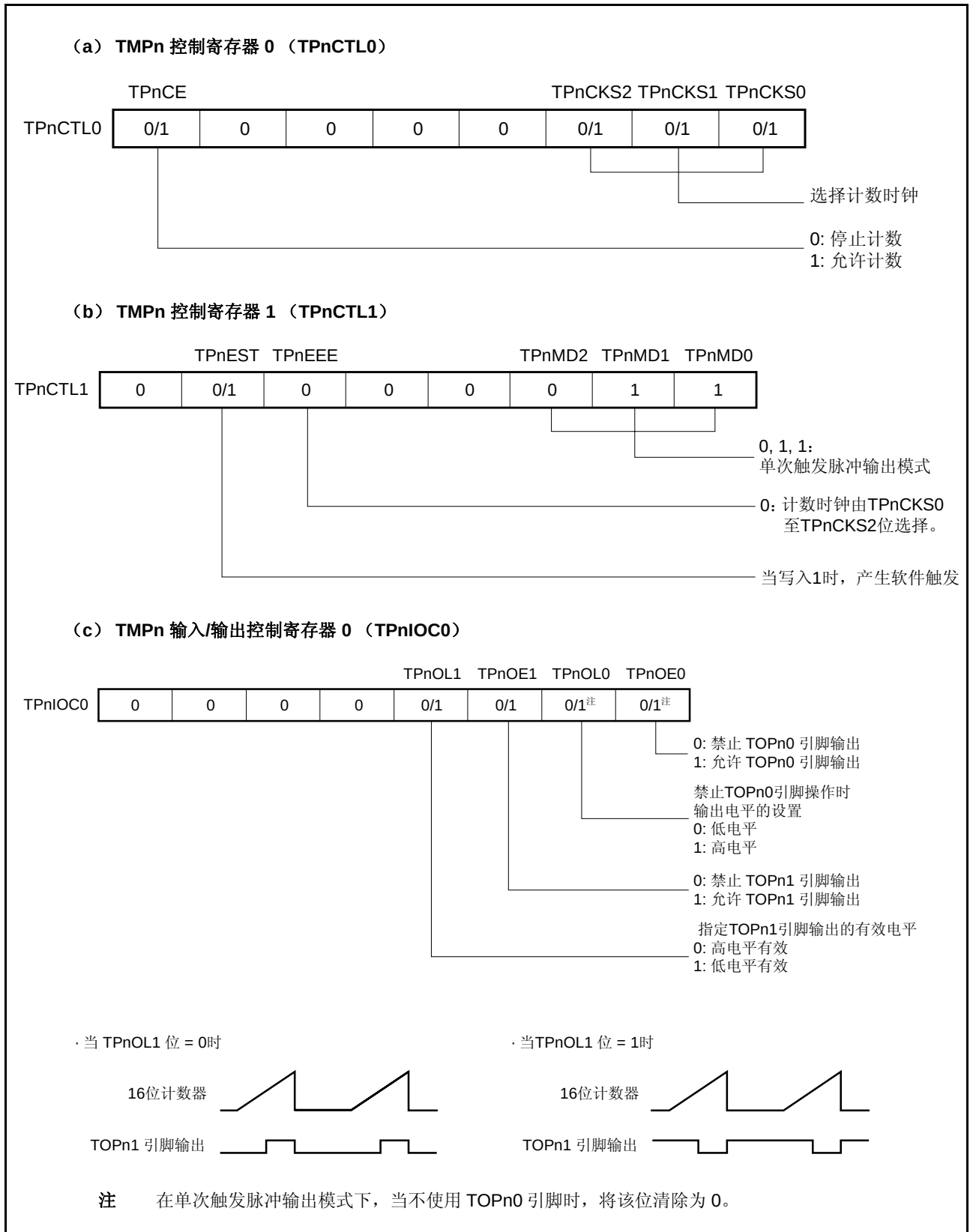
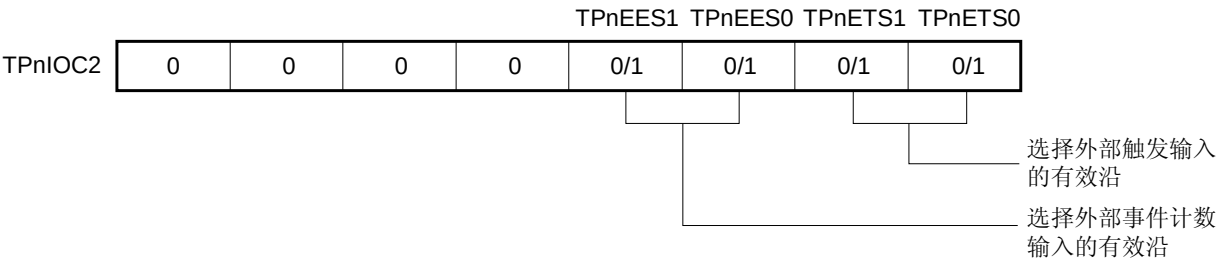


图 7-22. 单次触发脉冲输出模式时的寄存器设置（2/2）

<R>

(d) TMPn 输入/输出控制寄存器 2（TPnIOC2）



(e) TMPn 计数器读取缓冲寄存器（TPnCNT）

通过读取 TPnCNT 寄存器，可以读出 16 位计数器的值。

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

若把 TPnCCR0 寄存器的值设置为 D_0 ，TPnCCR1 寄存器的值设置为 D_1 ，则单次触发脉冲的有效电平宽度和输出延迟时间的计算如下。

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

输出延迟时间 = $D_1 \times$ 计数时钟周期

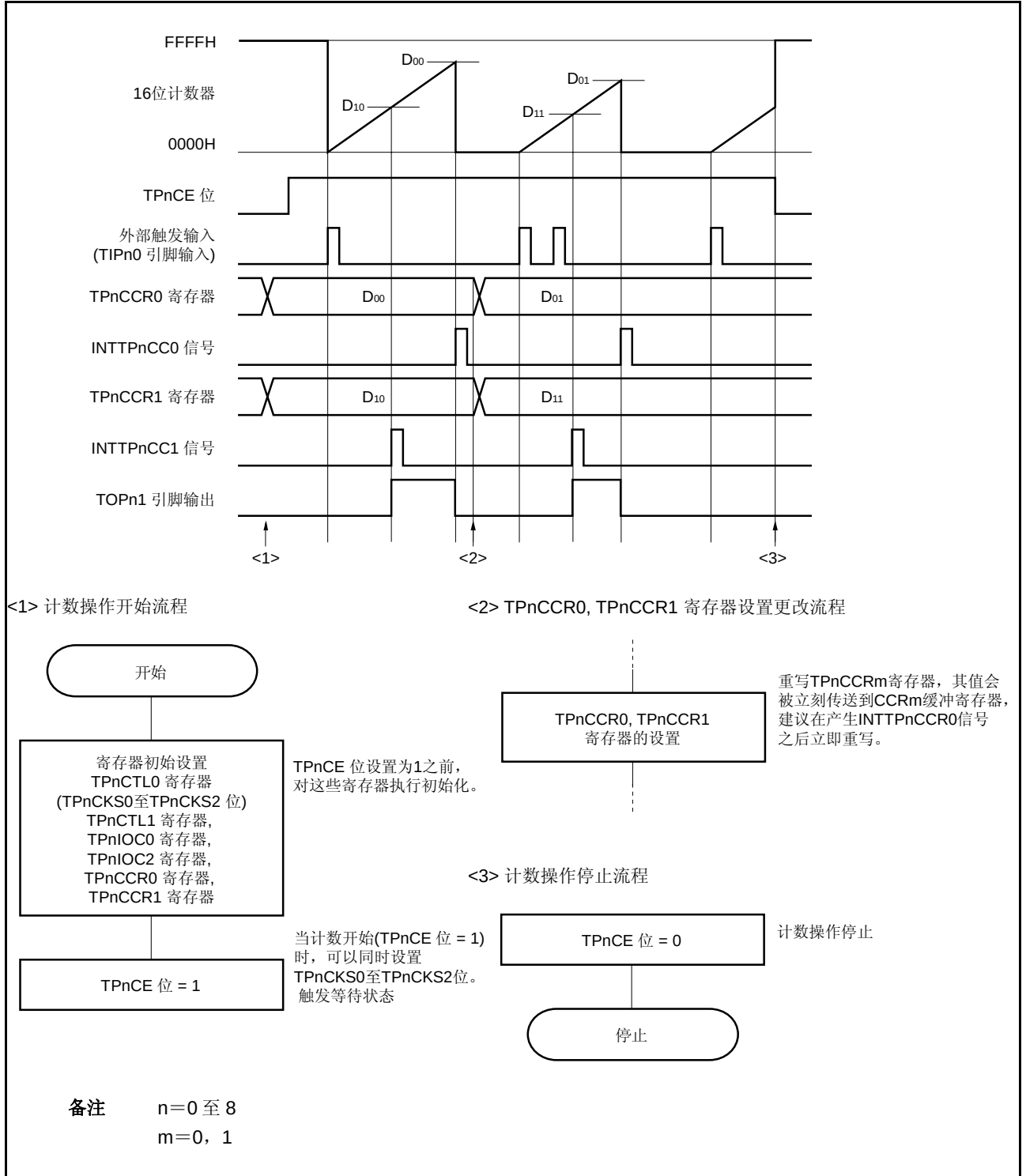
注意事项 如果 TPnCCR1 寄存器的设置值大于 TPnCCR0 寄存器的设置值，那么，即使在单次触发脉冲输出模式下也不会输出单次触发脉冲。

备注

1. 在单次触发脉冲输出模式下，不使用 TMPn 输入/输出控制寄存器 1（TPnIOC1）和 TMPn 选项寄存器 0（TPnOPT0）。
2. $n=0$ 至 8

(1) 单次触发脉冲输出模式下的操作流程

图 7-23. 单次触发脉冲输出模式的软件处理流程

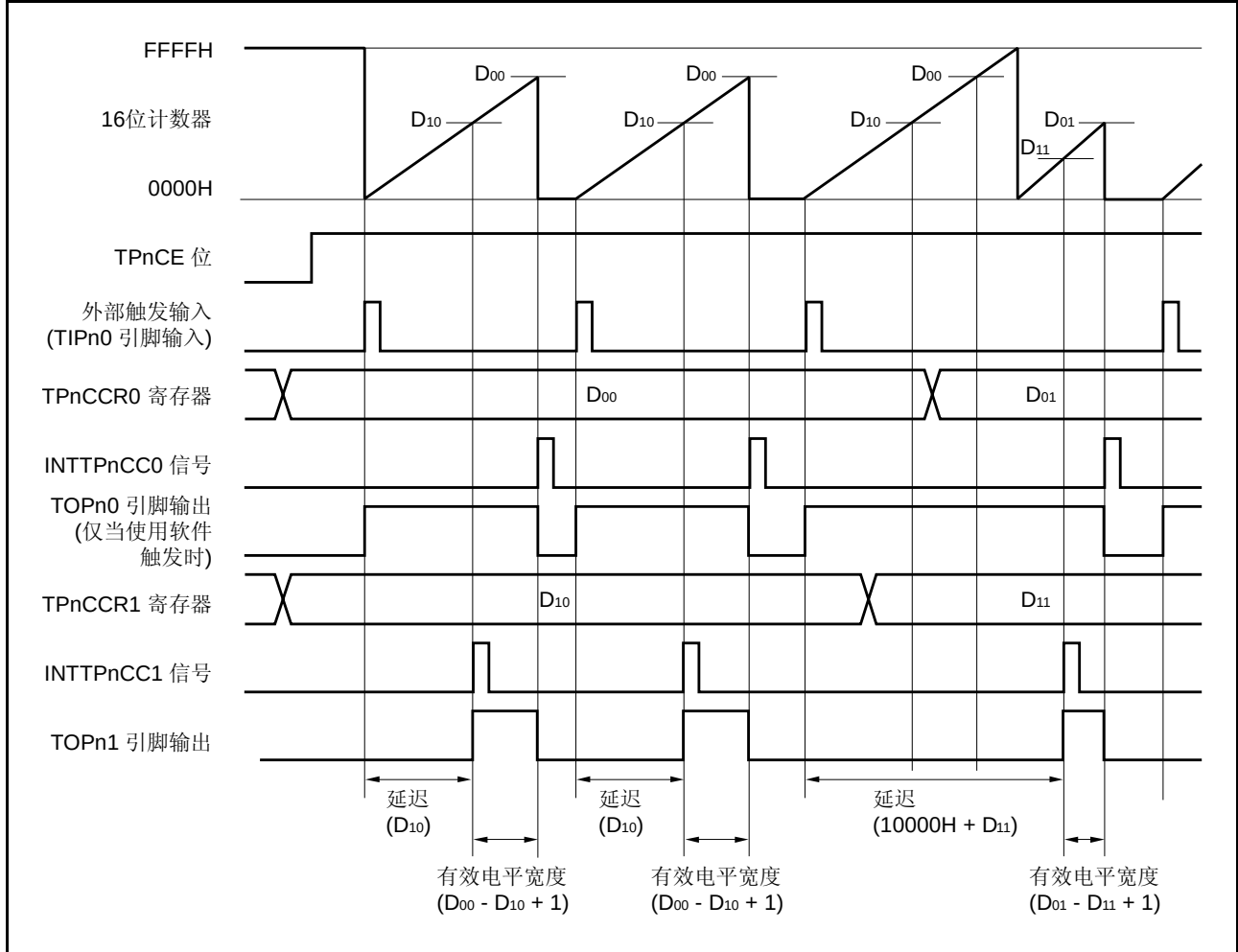


(2) 单次触发脉冲输出模式的操作时序

(a) 重写 TPnCCRm 寄存器的注意事项

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

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



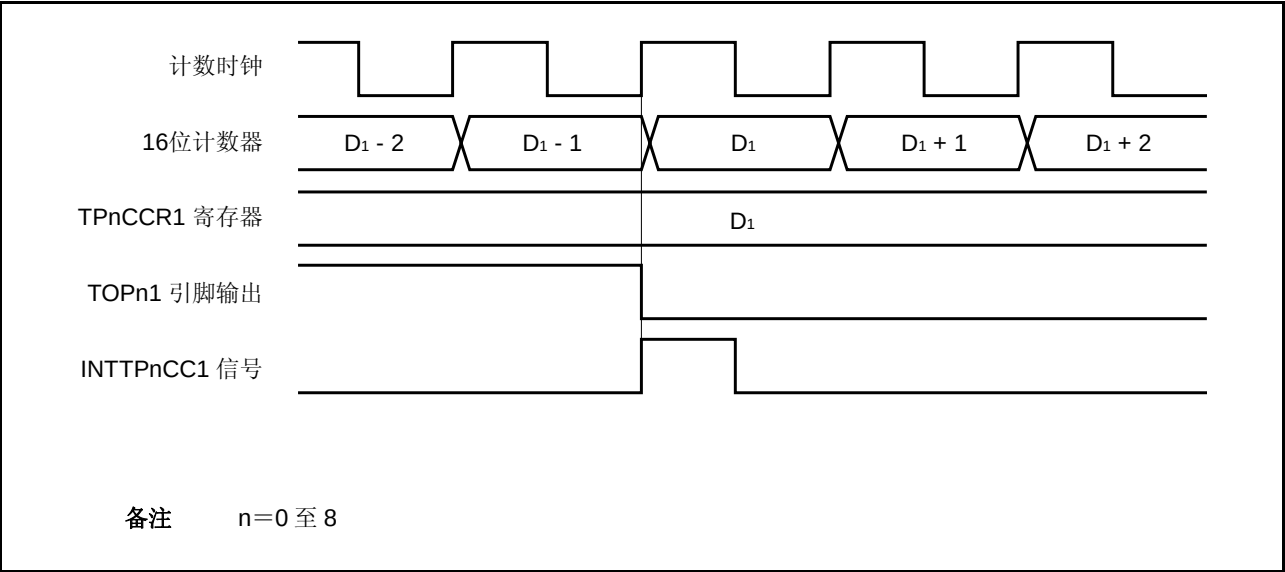
假设 $D_{00} > D_{01}$ 而且 $D_{10} > D_{11}$ ，TPnCCR0 寄存器由 D_{00} 重写为 D_{01} ，TPnCCR1 寄存器由 D_{10} 重写为 D_{11} ，如果在 16 位计数器的计数值大于 D_{11} 且小于 D_{10} 时重写 TPnCCR1 寄存器，并且在计数值大于 D_{01} 且小于 D_{00} 时重写 TPnCCR0 寄存器，则一旦重写寄存器，就会在各设置值上反映出来，并立即与计数值进行比较。计数器向上计数到 FFFFH 后，从 0000H 开始重新向上计数。当计数值与 D_{11} 匹配时，计数器会产生 INTTPnCC1 信号并使 TOPn1 引脚有效。当计数值与 D_{01} 匹配时，计数器产生 INTTPnCC0 信号，使 TOPn1 引脚失效，并停止计数。

因此，计数器输出脉冲的延迟时段或有效时段可能会与原先预计的单次触发脉冲有所不同。

备注 $n=0$ 至 8
 $m=0, 1$

(b) 比较匹配中断请求信号（INTTPnCC1）的发生时序

在单次触发脉冲输出模式下，比较匹配中断请求信号（INTTPnCC1）的发生时序和其它情况下的 INTTPnCC1 信号发生时序不同，当 16 位计数器的计数值与 TPnCCR1 寄存器的值匹配时，产生 INTTPnCC1 中断请求信号。



通常在 16 位计数器的当前计数值与 TPnCCR1 寄存器的值匹配之后，在下一次数值同步发生 INTTPnCC1 信号。

但是，在单次触发脉冲输出模式下，INTTPnCC1 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOPn1 引脚输出信号变化的时序。

备注 n=0 至 8

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

在 PWM 输出模式下，当 TPnCTL0.TPnCE 位被置为 1 时，PWM 波形从 TOPn1 引脚输出。
而且，由 TOPn0 引脚输出的脉冲，其半周期等于 PWM 的全周期。

图 7-24. PWM 输出模式的配置

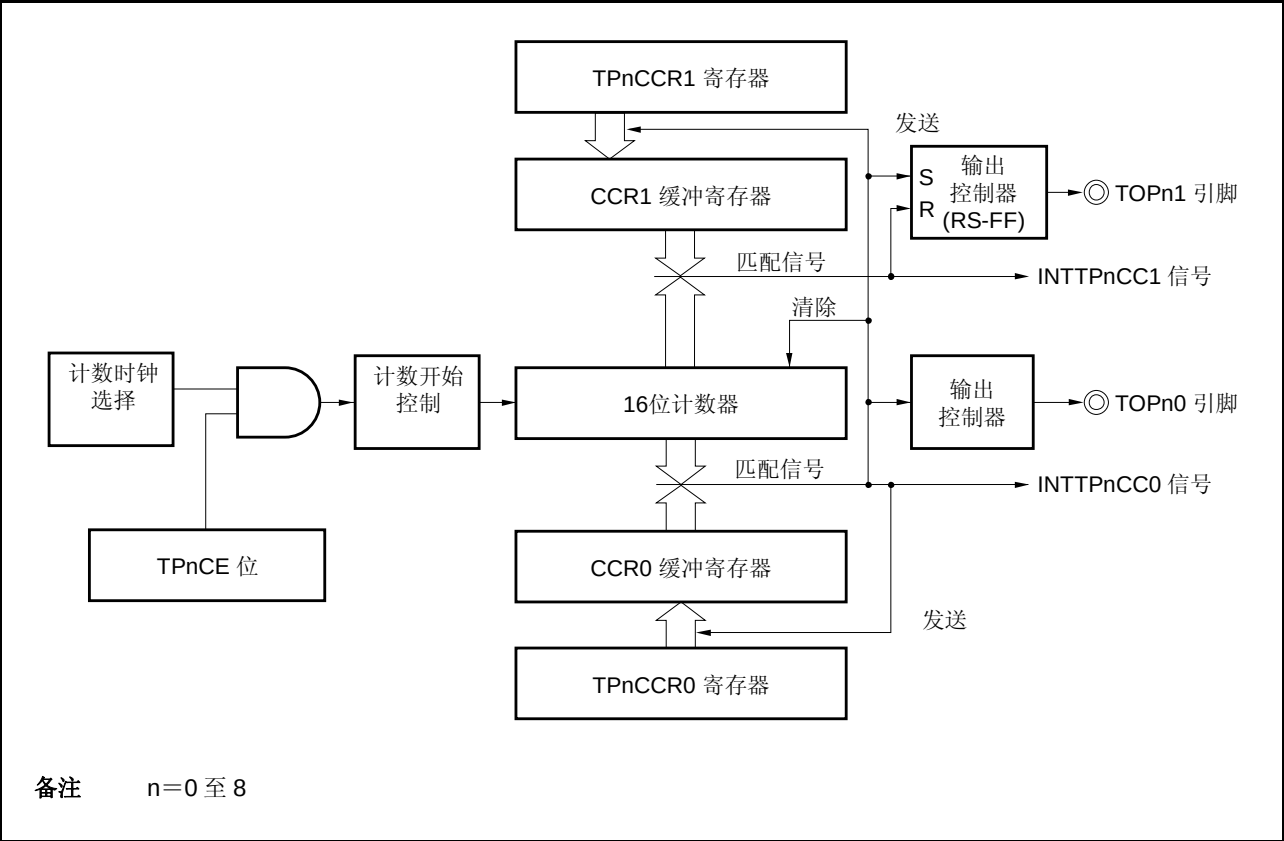
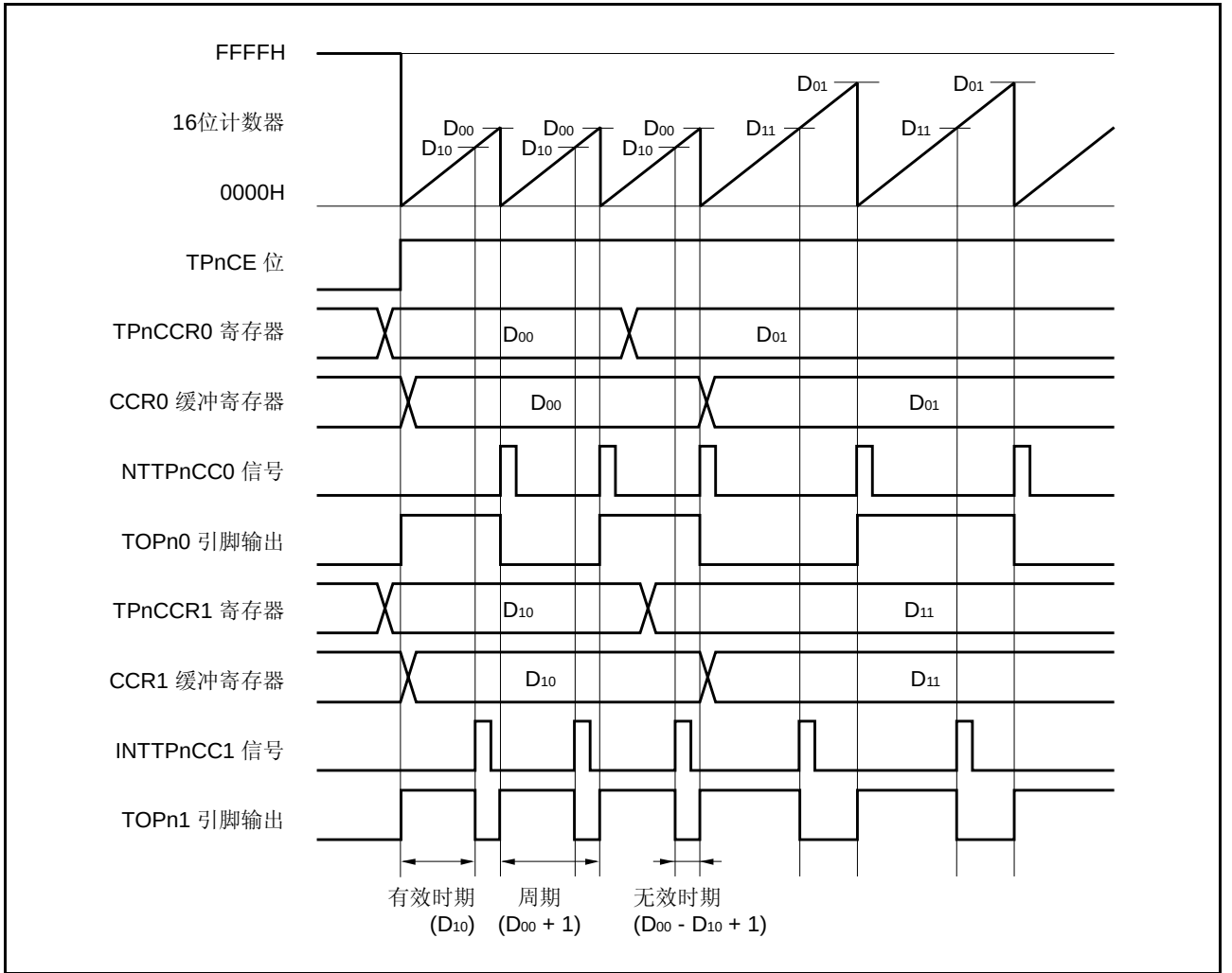


图 7-25. PWM 输出模式的基本时序



当 TPnCE 位置为 1 时，16 位计数器从 FFFFH 清零为 0000H，计数开始，同时从 TOPn1 引脚输出 PWM 波形。PWM 波形的有效电平宽度、周期、占空比系数的计算如下。

有效电平宽度 = (TPnCCR1 寄存器的设置值) × 计数时钟周期

周期 = (TPnCCR0 寄存器的设置值 + 1) × 计数时钟周期

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

在计数操作时，可通过重写 TPnCCRm 寄存器来改变 PWM 波形。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配时，16 位计数器清零为 0000H 时，新写入的值会发生作用。

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配之后，进行下一次计数时，会同步产生比较匹配中断请求信号 INTTPnCC0，且 16 位计数器清零为 0000H。当 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTPnCC1。

当 16 位计数器的计数值与 CCRm 缓冲寄存器的值匹配，而且 16 位计数器清零为 0000H 时，TPnCCRm 寄存器的设定值就被传送到 CCRm 缓冲寄存器。

备注 n=0 至 8, m=0, 1

图 7-26. PWM 输出模式时的寄存器设置 (1/2)

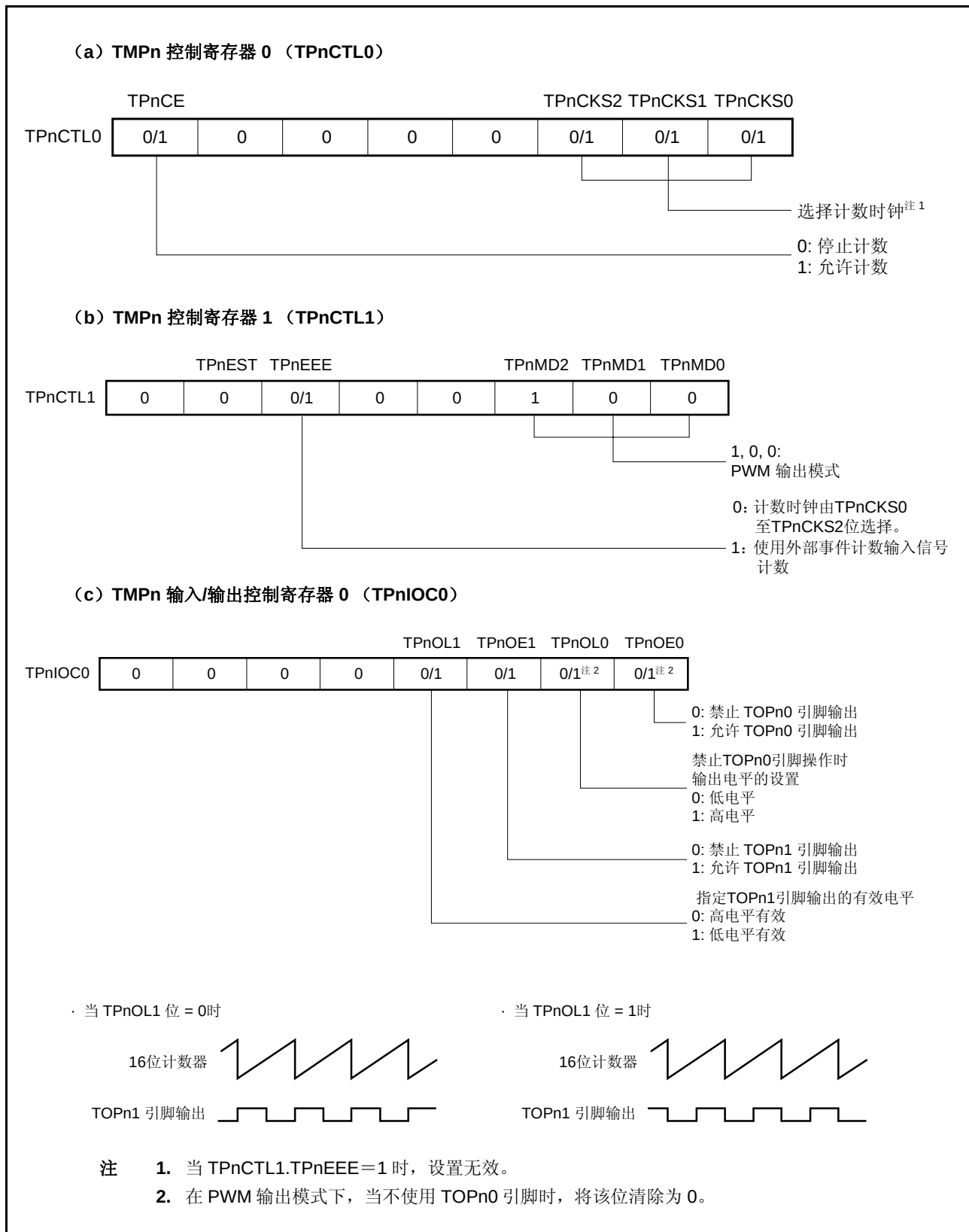
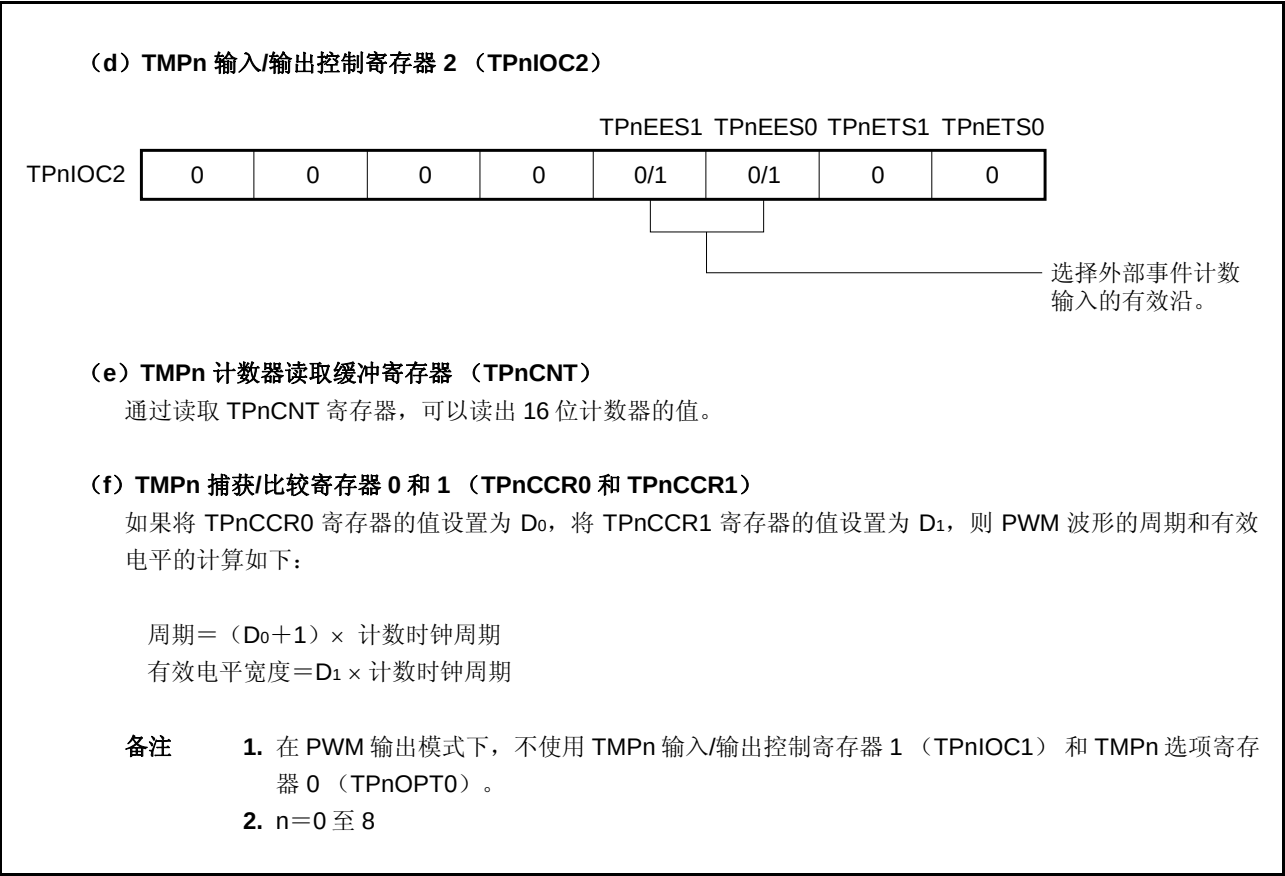


图 7-26. PWM 输出模式时的寄存器设置（2/2）



(1) PWM 输出模式下的操作流程

图 7-27. PWM 输出模式的软件处理流程（1/2）

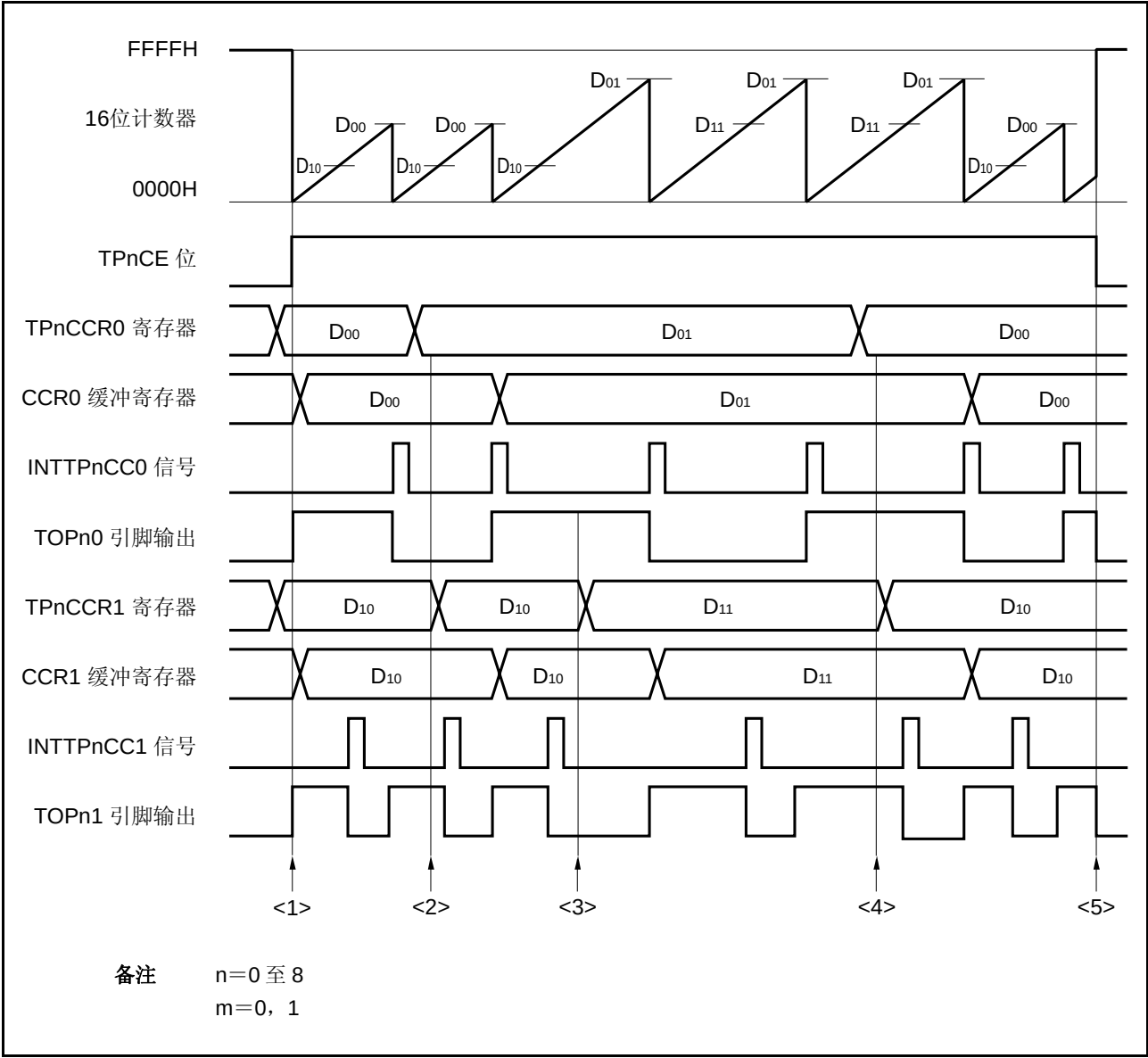
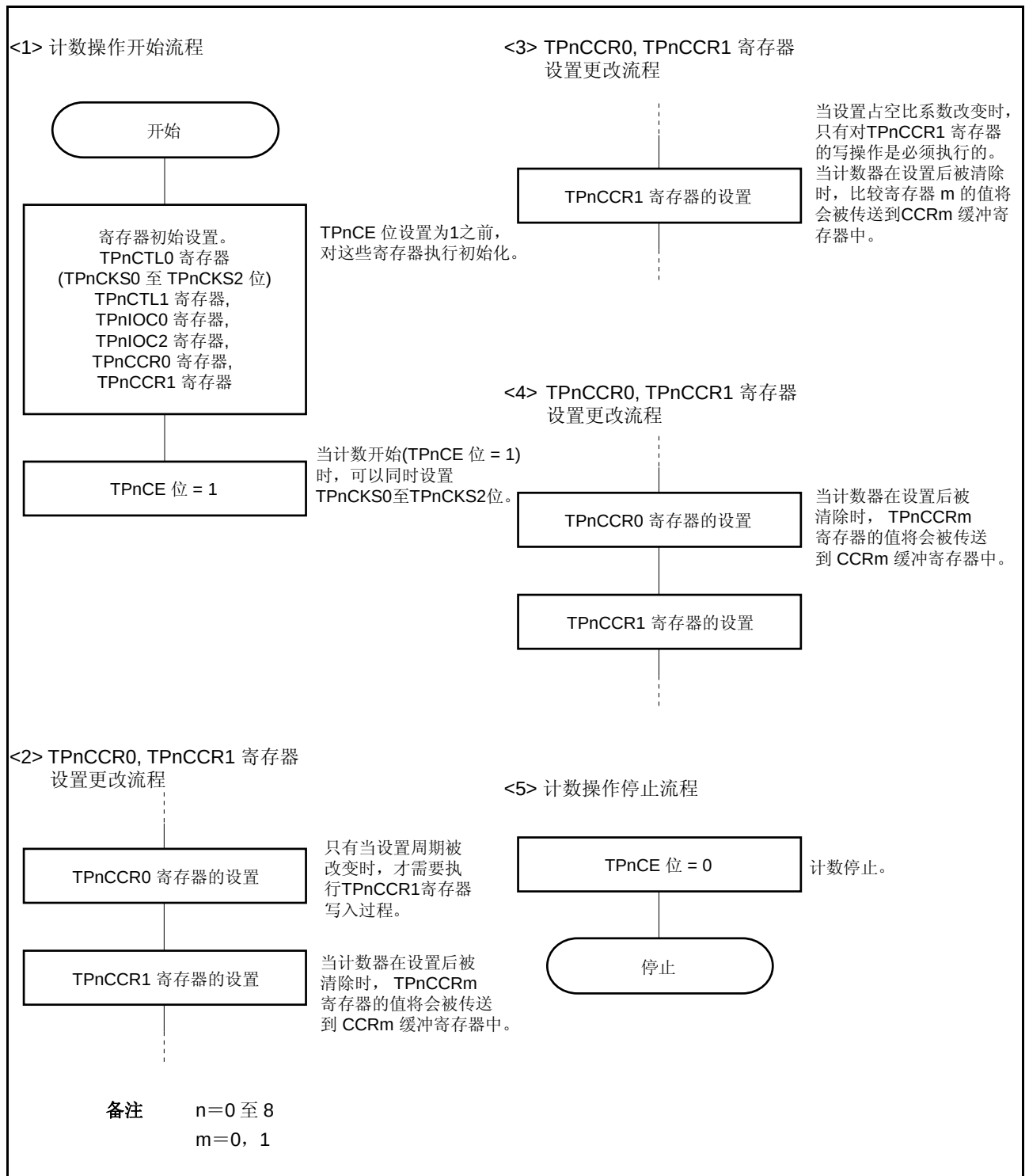


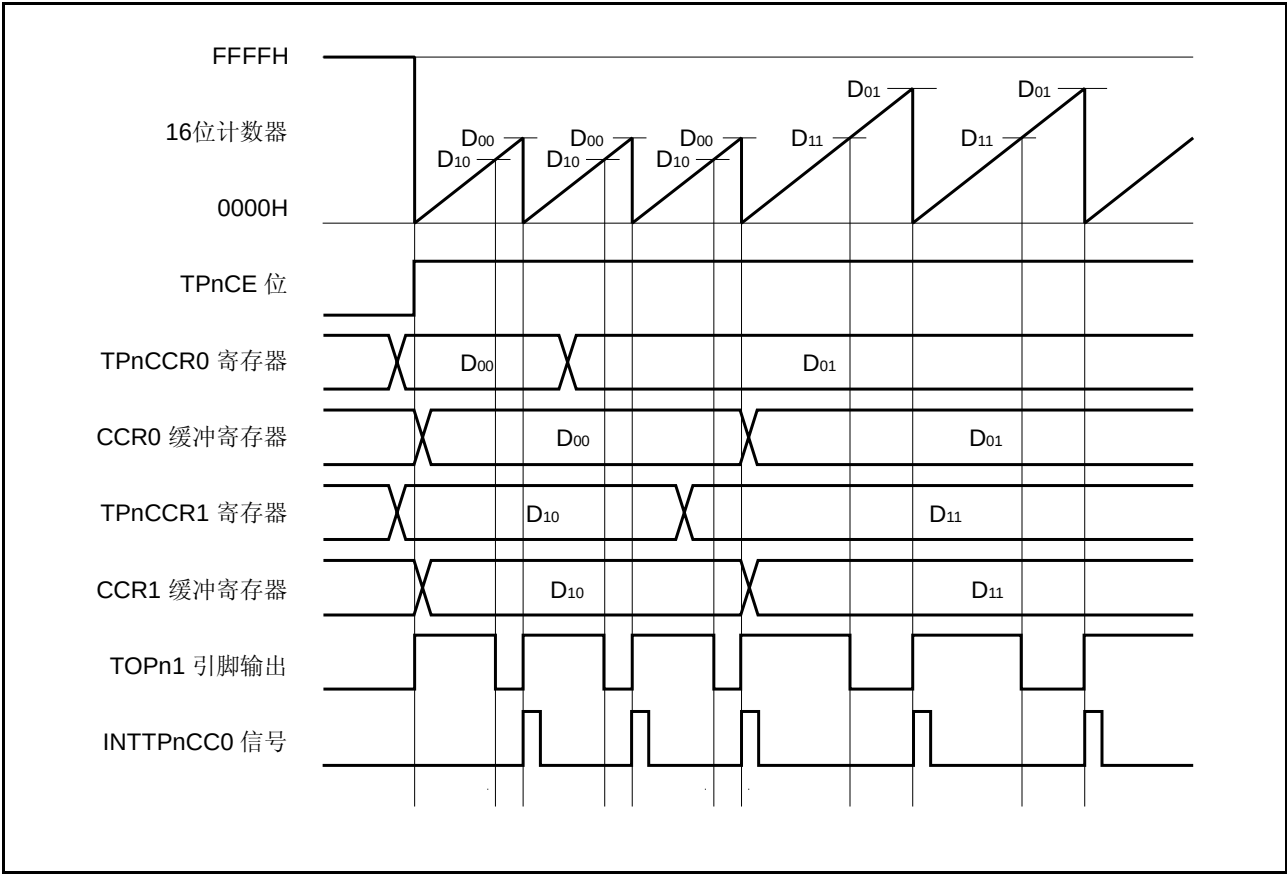
图 7-27. PWM 输出模式时的软件处理流程（2/2）



（2）PWM 输出模式的操作时序

（a）操作过程中改变脉冲宽度

计数器操作过程中，若要改变 PWM 波形，应重新写入 TPnCCR1 寄存器。
当检测到 INTTPnCC1 信号以后，在写入 TPnCCR1 寄存器之后重写 TPnCCRm 寄存器。

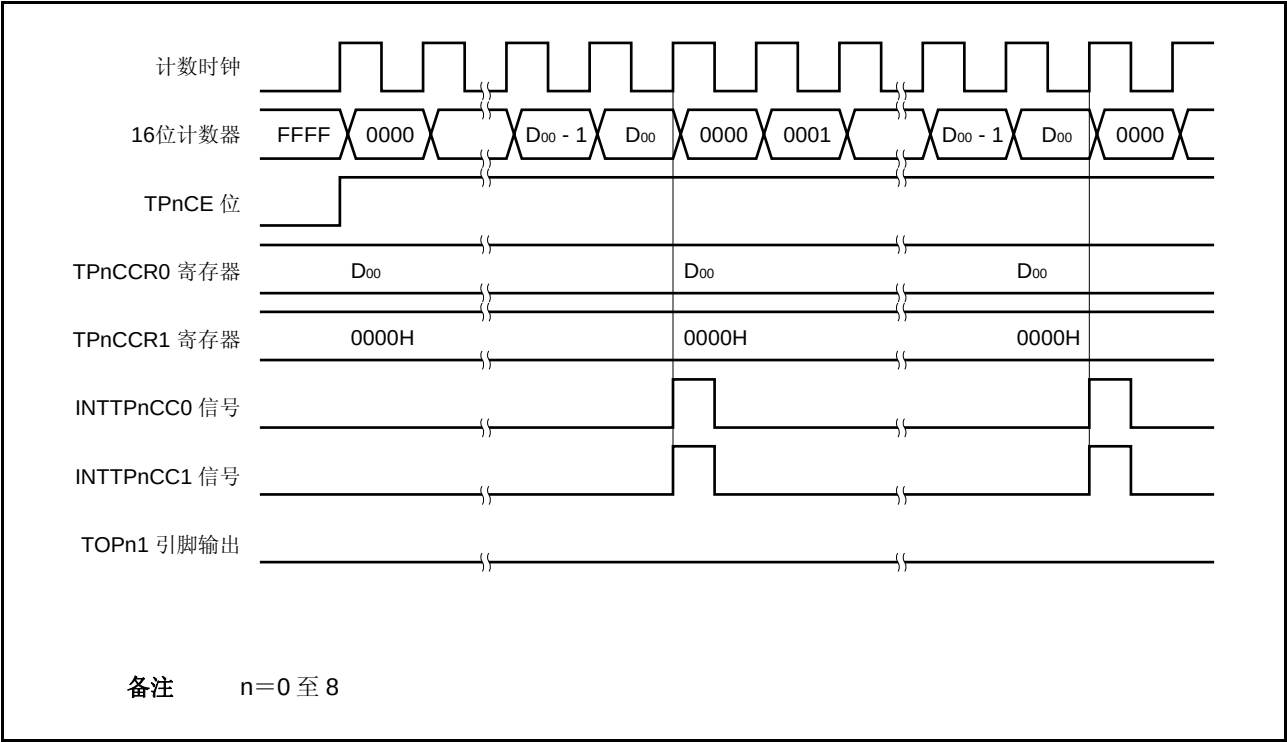


要把数据从 TPnCCRm 寄存器传送到 CCRm 缓冲寄存器，必须对 TPnCCR1 寄存器进行写入操作。
此时如果 PWM 波形的周期和有效电平都要改变，应先将周期设定到 TPnCCR0 寄存器，然后再将有效电平设定到 TPnCCR1 寄存器。
如果只改变 PWM 波形的周期，应首先把周期设定给 TPnCCR0 寄存器，然后将相同值（原值）写入 TPnCCR1 寄存器。
如果只改变 PWM 波形的有效电平宽度（占空比），仅需要设置 TPnCCR1 寄存器。
当数据写入 TPnCCR1 寄存器之后，在对 16 位计数器清零的同时，写入 TPnCCRm 寄存器的值被传送到 CCRm 缓冲寄存器，并作为与 16 位计数器进行比较的值。
在写入 TPnCCR1 寄存器一次之后，要再次写入 TPnCCR0 或 TPnCCR1 寄存器，应该在 INTTPnCC0 信号产生之后进行。否则，CCRM 缓冲寄存器的值可能不确定，因为把数据从 TPnCCRm 寄存器传送到 CCRm 缓冲寄存器的时序会和写入 TPnCCRm 寄存器时序发生冲突。

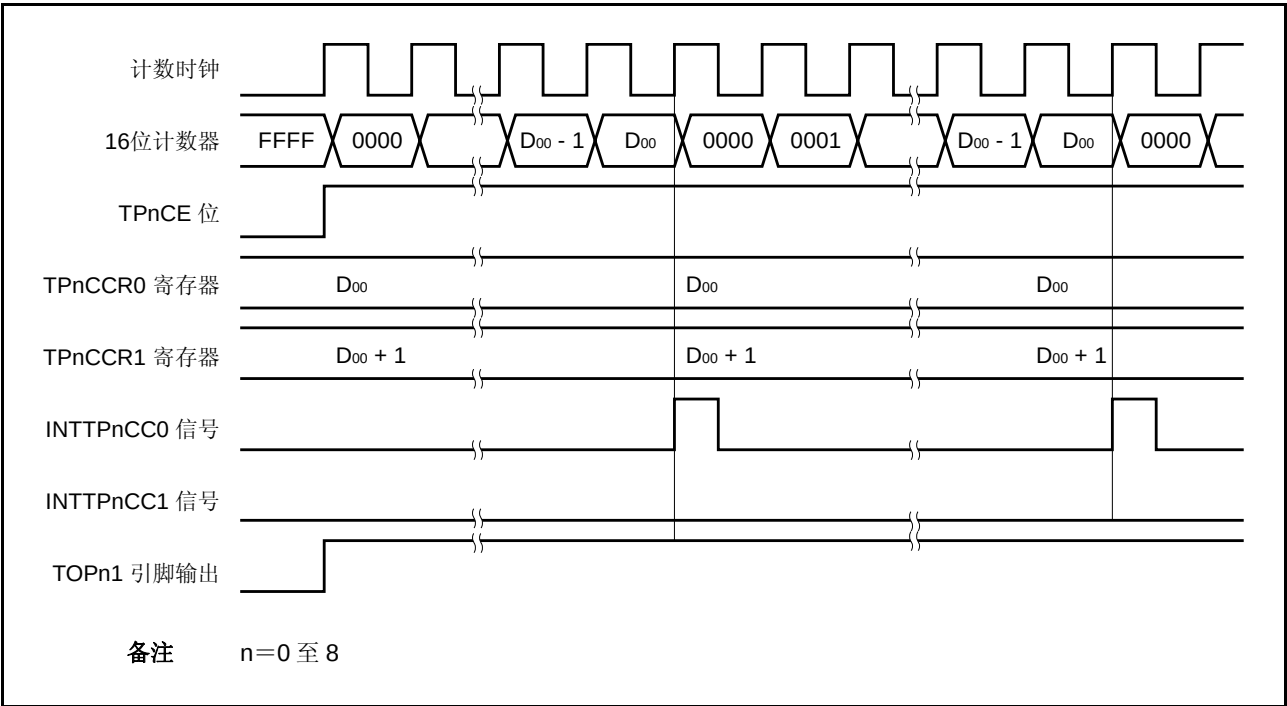
备注 n=0 至 8，m=0，1

(b) PWM 波形的 0%和 100% 输出

为了输出 0%的波形，需要将 TPnCCR1 寄存器设置为 0000H。如果 TPnCCR0 寄存器的设定值为 FFFFH，则会在每个周期产生 INTTPnCC1 中断请求信号。

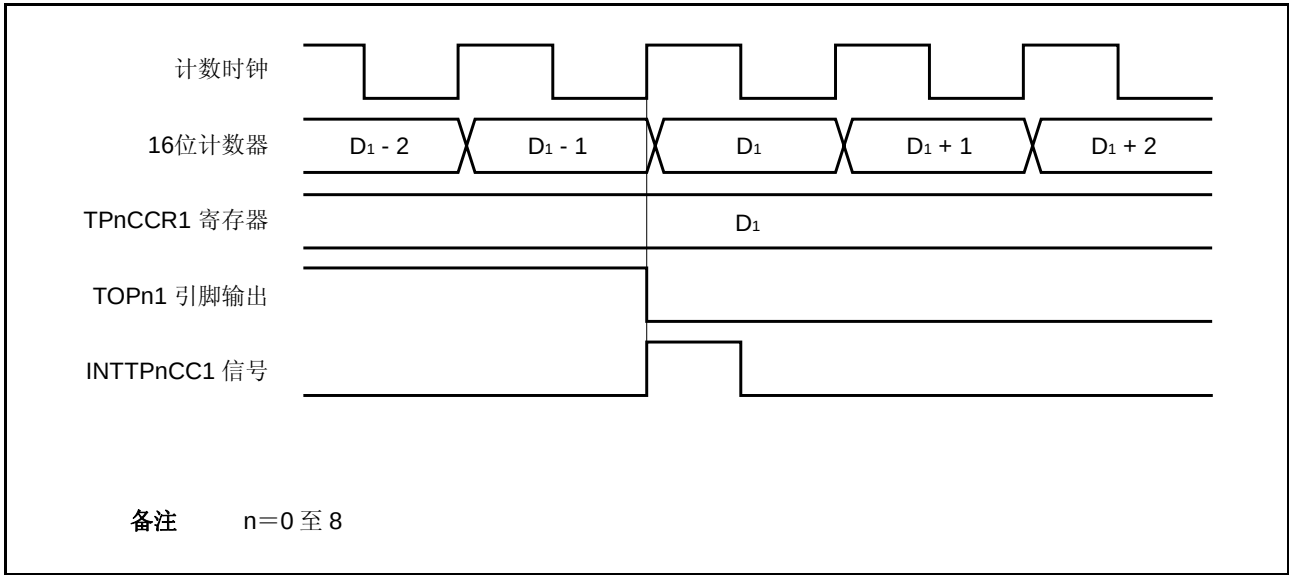


为了输出 100%的波形，需要将 TPnCCR1 寄存器的值设为：（TPnCCR0 寄存器的设定值+1）。如果 TPnCCR0 寄存器的设定值为 FFFFH，则不会产生 100%波形输出。



(c) 比较匹配中断请求信号 (INTTPnCC1) 的发生时序

在 PWM 输出模式下，比较匹配中断请求信号 (INTTPnCC1) 的发生时序与其它情况下 INTTPnCC1 信号的发生时序有所不同；当 16 位计数器的计数值与 TPnCCR1 寄存器的值匹配时，产生 INTTPnCC1 信号。



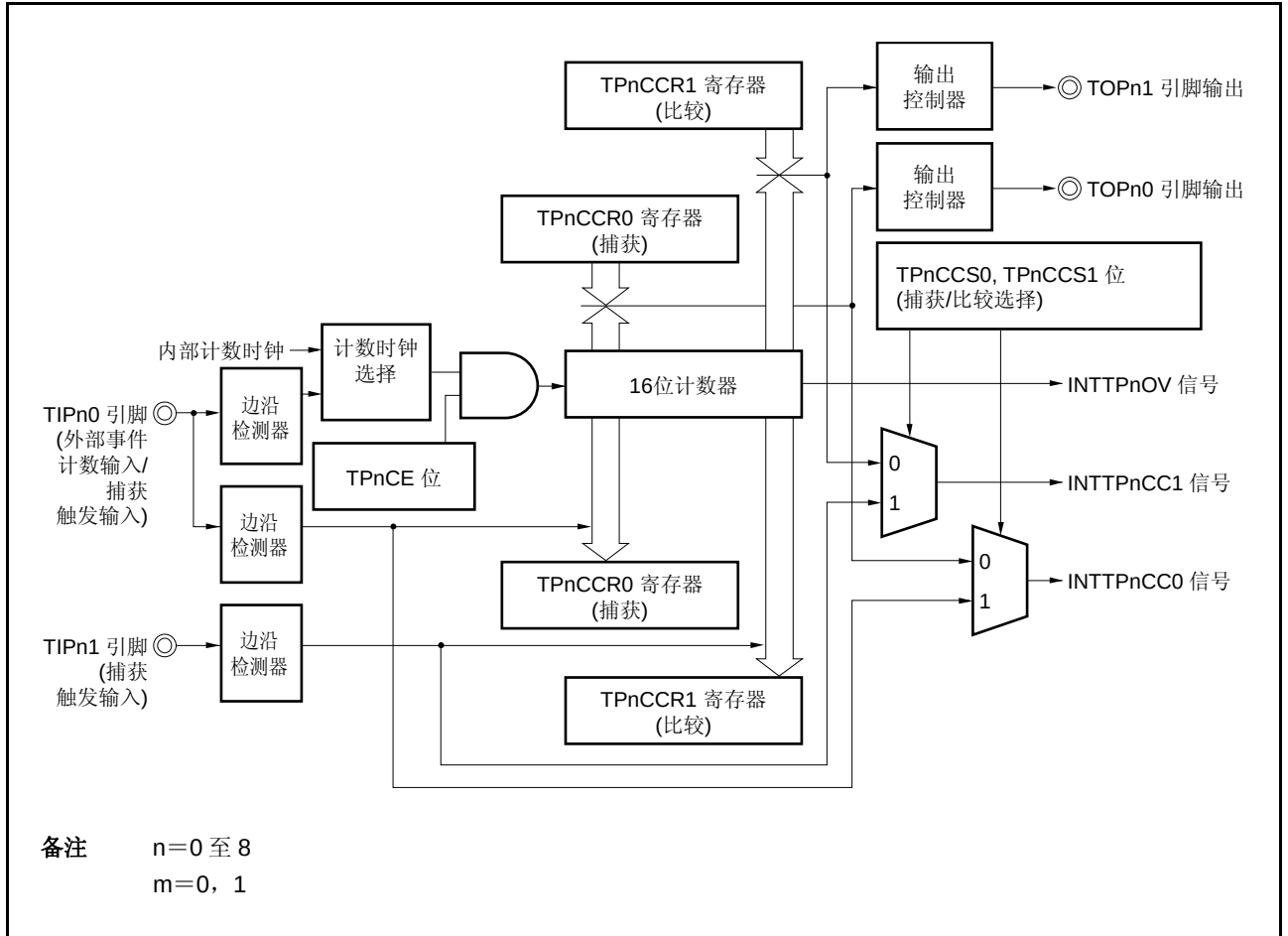
通常在 16 位计数器的计数值与 TPnCCR1 寄存器的值匹配之后，在下一次计数时同步发生 INTTPnCC1 信号。

但是，在 PWM 输出模式下，INTTPnCC1 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOPn1 引脚输出信号变化的时序。

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

在自由运行定时器模式下，当 TPnCTL0.TPnCE 位置为 1 时，16 位定时器/事件计数器 P 开始计数。此时，根据 TPnOPT0.TPnCCS0 和 TPnOPT0.TPnCCS1 位的设置，TPnCCRm 寄存器可以用作比较寄存器或捕获寄存器。

图 7-28. 自由运行定时器模式的配置

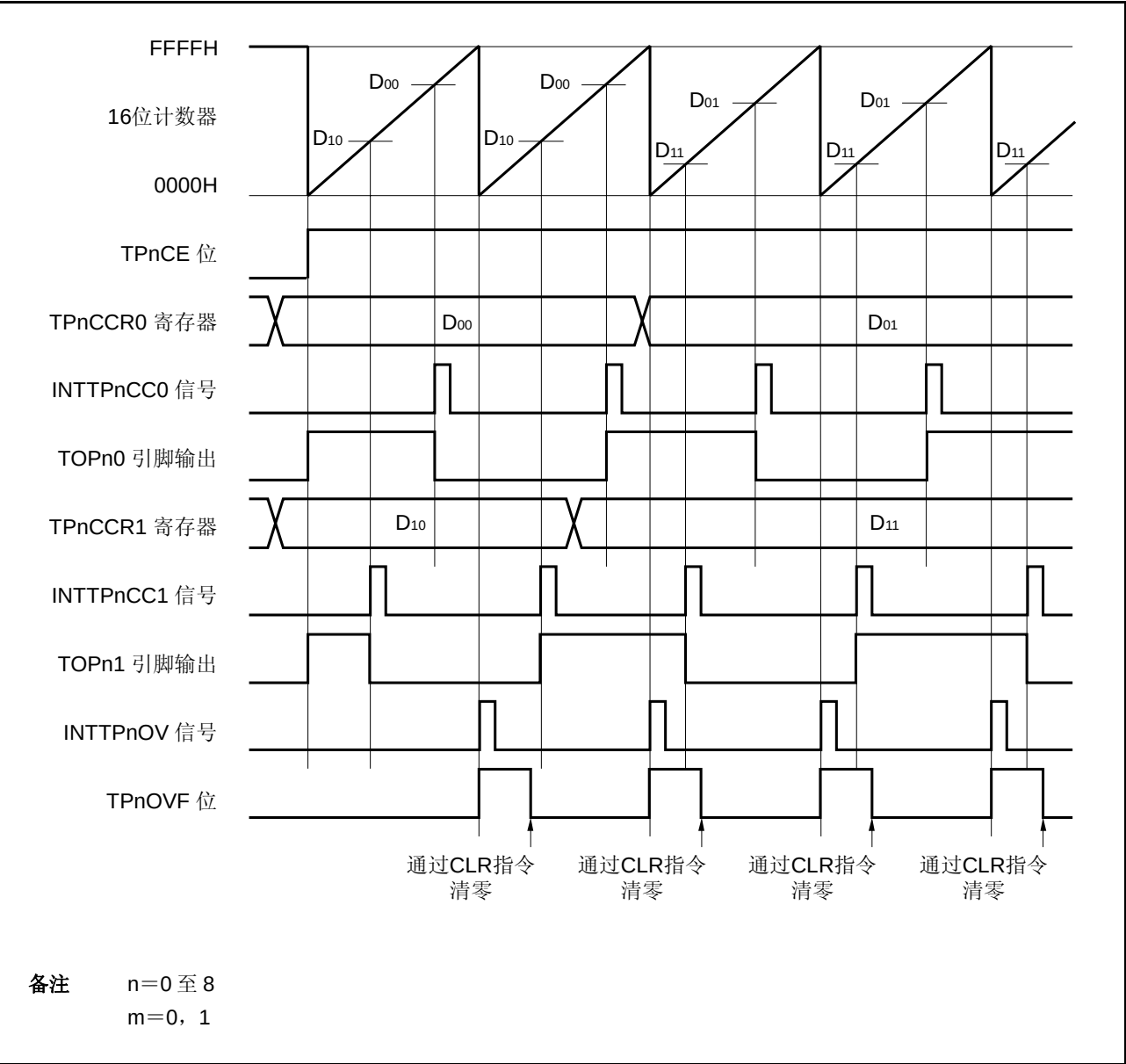


当 TPnCE 位置为 1 时，16 位定时器/事件计数器 P 开始计数，TOPn0 和 TOPn1 引脚的输出信号反相。此后当 16 位计数器的计数值与 TPnCCRm 寄存器的设定值匹配时，产生一个比较匹配中断请求信号（INTTPnCCm），而且 TOPnm 引脚的输出信号反相。

16 位计数器继续对计数时钟同步计数。当它计数到 FFFFH 时，会在下一个计数时钟时产生一个溢出中断请求信号（INTTPnOV），同时计数器清零为 0000H，然后继续进行计数。此时，溢出标志（TPnOPT0.TPnOVF 位）置为 1，通过软件运行 CLR 指令可以把溢出标志位清 0。

当计数器工作时，可以重写 TPnCCRm 寄存器。如果进行重写，则新写入的值会立刻反映出来，并与计数值进行比较。

图 7-29. 自由运行定时器模式下的基本时序（比较功能）



当 TPnCE 位置为 1 时，16 位计数器开始计数。当检测到 TIPnm 引脚输入信号的有效边沿时，16 位计数器的计数值存储在 TPnCCRm 寄存器中，并产生一个捕获中断请求信号 (INTTPnCCm)。

16 位计数器继续对计数时钟同步计数。当它计数到 FFFFH 时，会在下一个计数时钟时产生一个溢出中断请求信号 (INTTPnOV)，同时计数器清零为 0000H，然后继续进行计数。此时，溢出标志 (TPnOPT0.TPnOVF 位) 置为 1，通过软件运行 CLR 指令可以把溢出标志位清 0。

图 7-30. 自由运行定时器模式下的基本时序 (捕获功能)

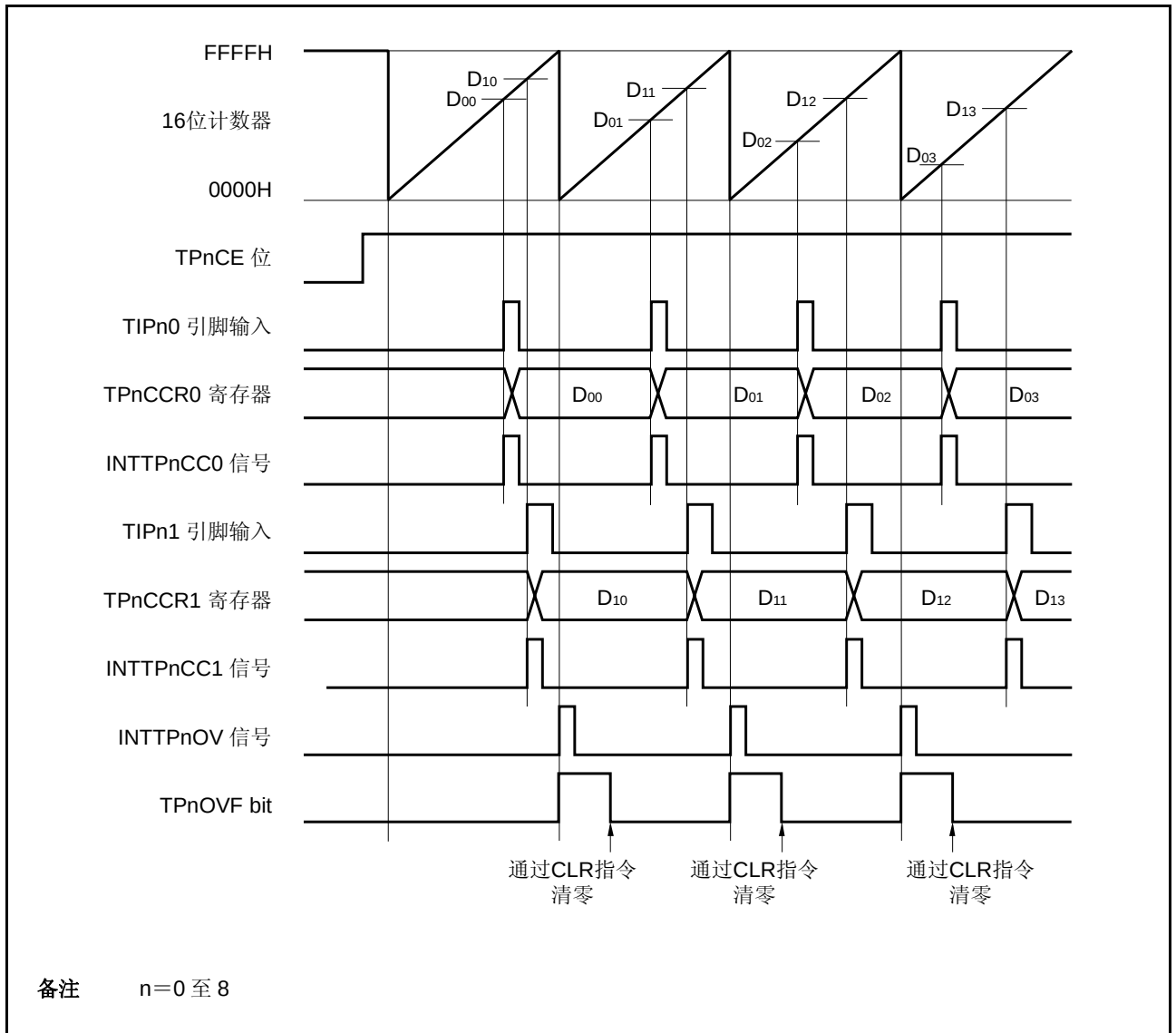


图 7-31. 自由运行定时器模式下的寄存器设置 (1/2)

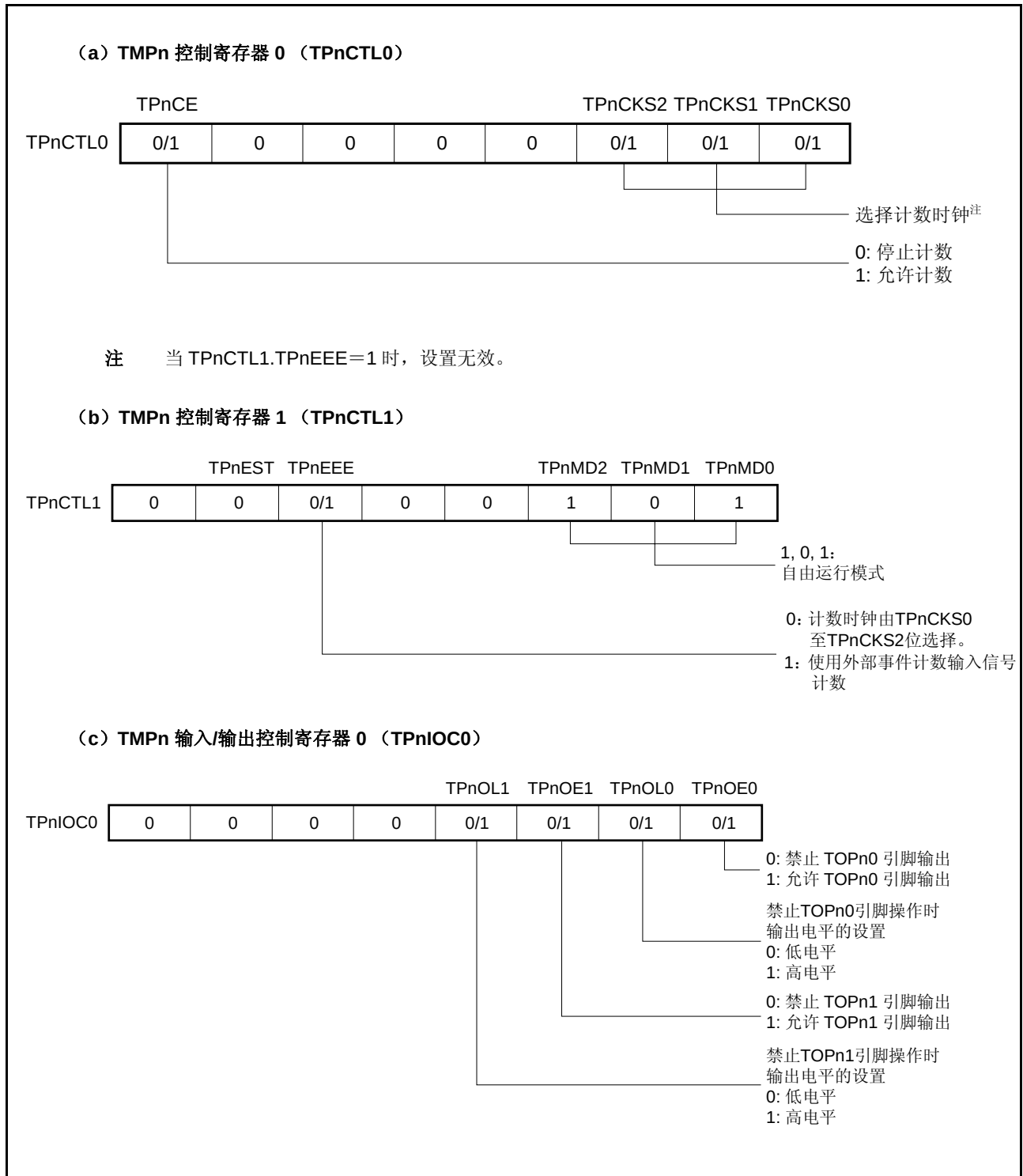
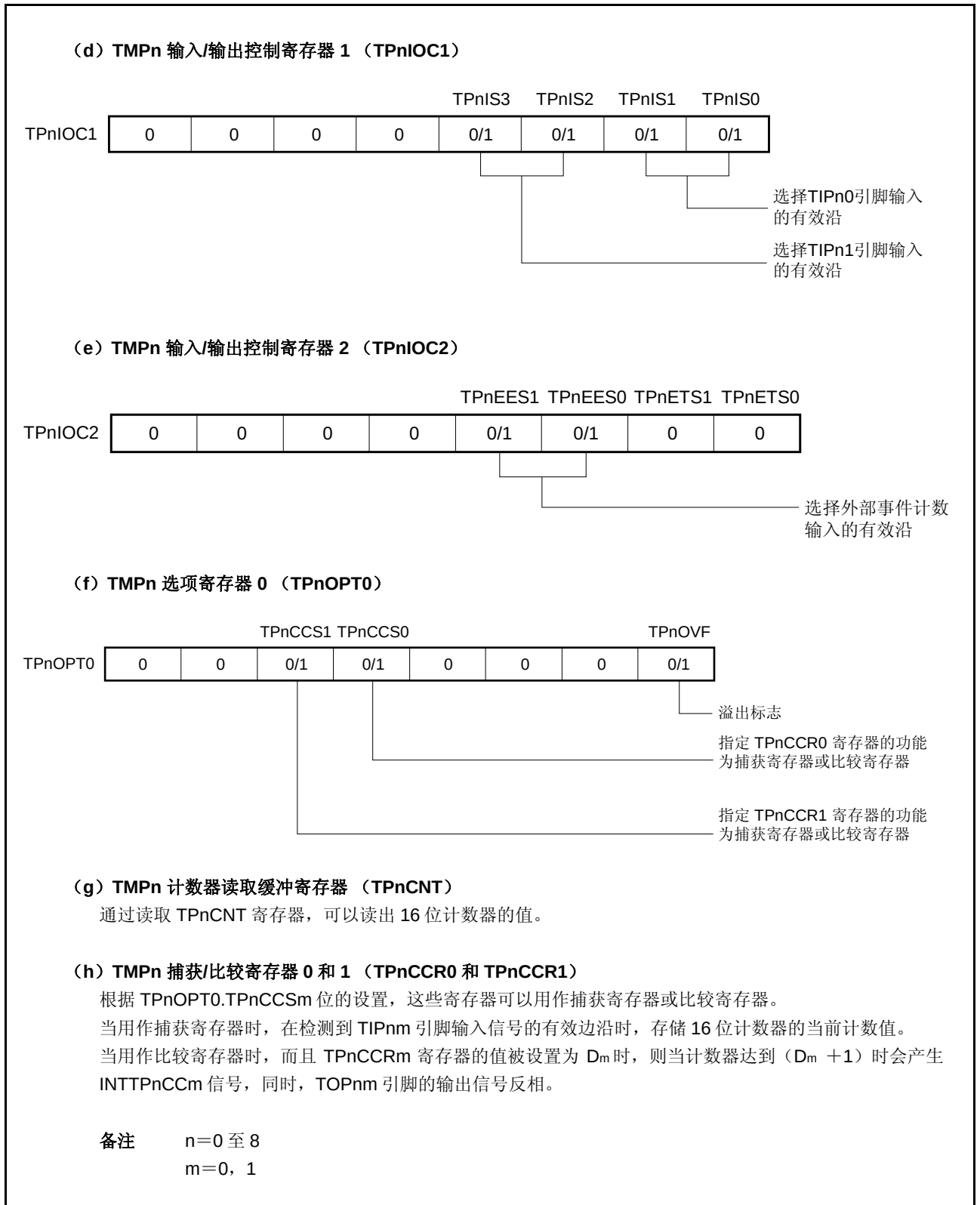


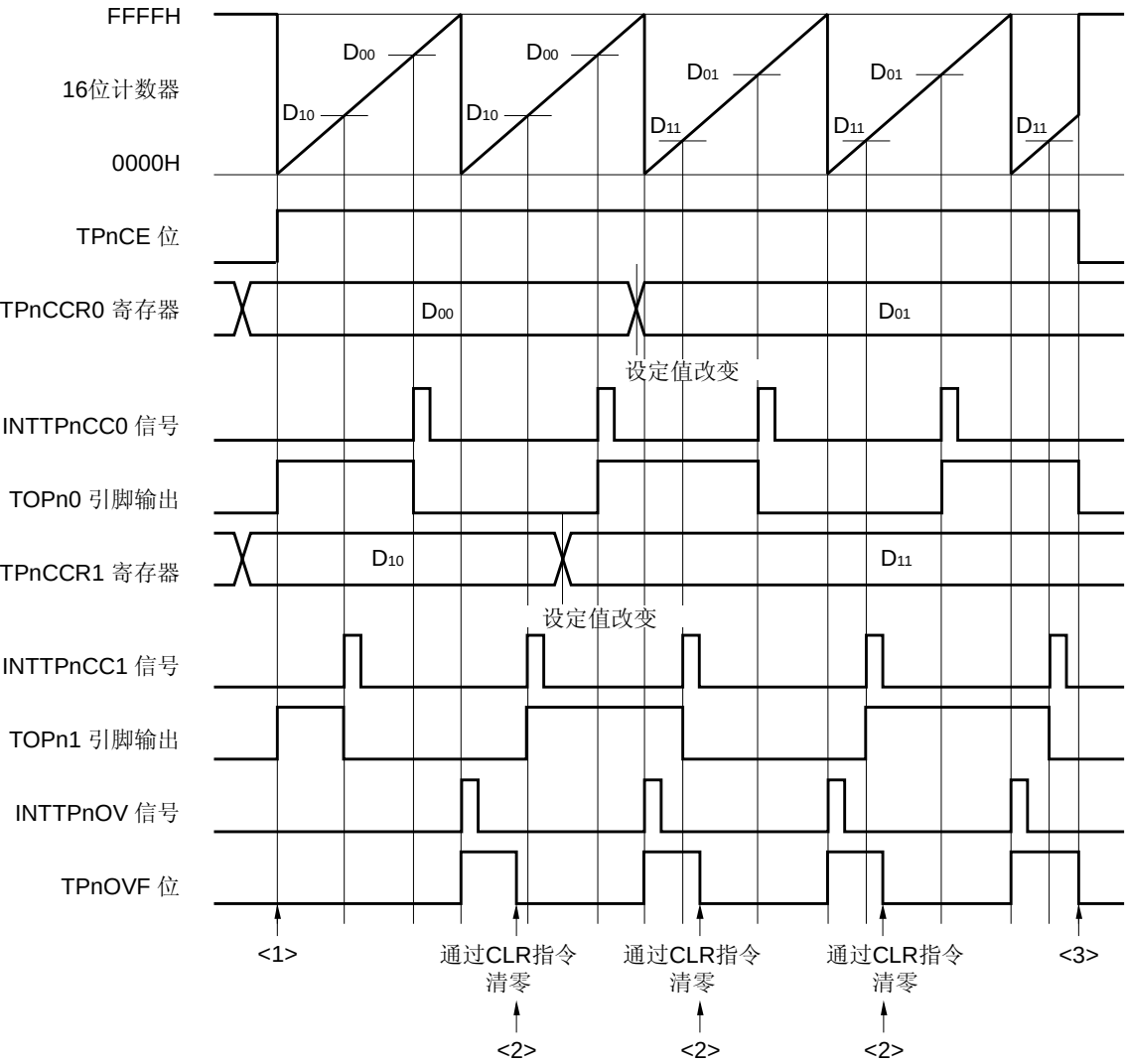
图 7-31. 自由运行定时器模式下的寄存器设置 (2/2)



（1）自由运行定时器模式的流程图

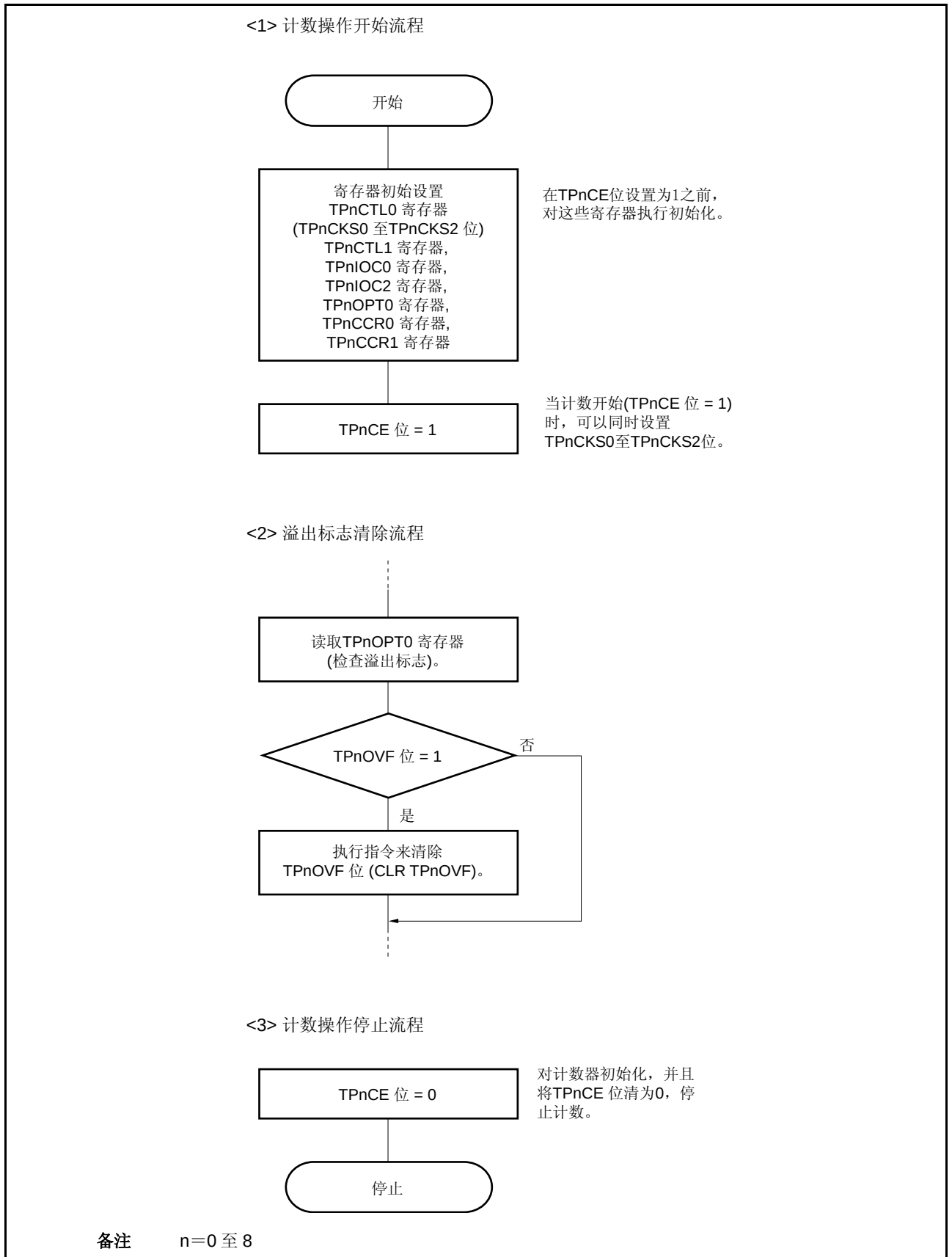
（a）当捕获/比较寄存器用作比较寄存器时

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）



备注 n=0 至 8

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) 当捕获/比较寄存器用作捕获寄存器时

图 7-33. 自由运行定时器模式的软件处理流程（捕获功能）（1/2）

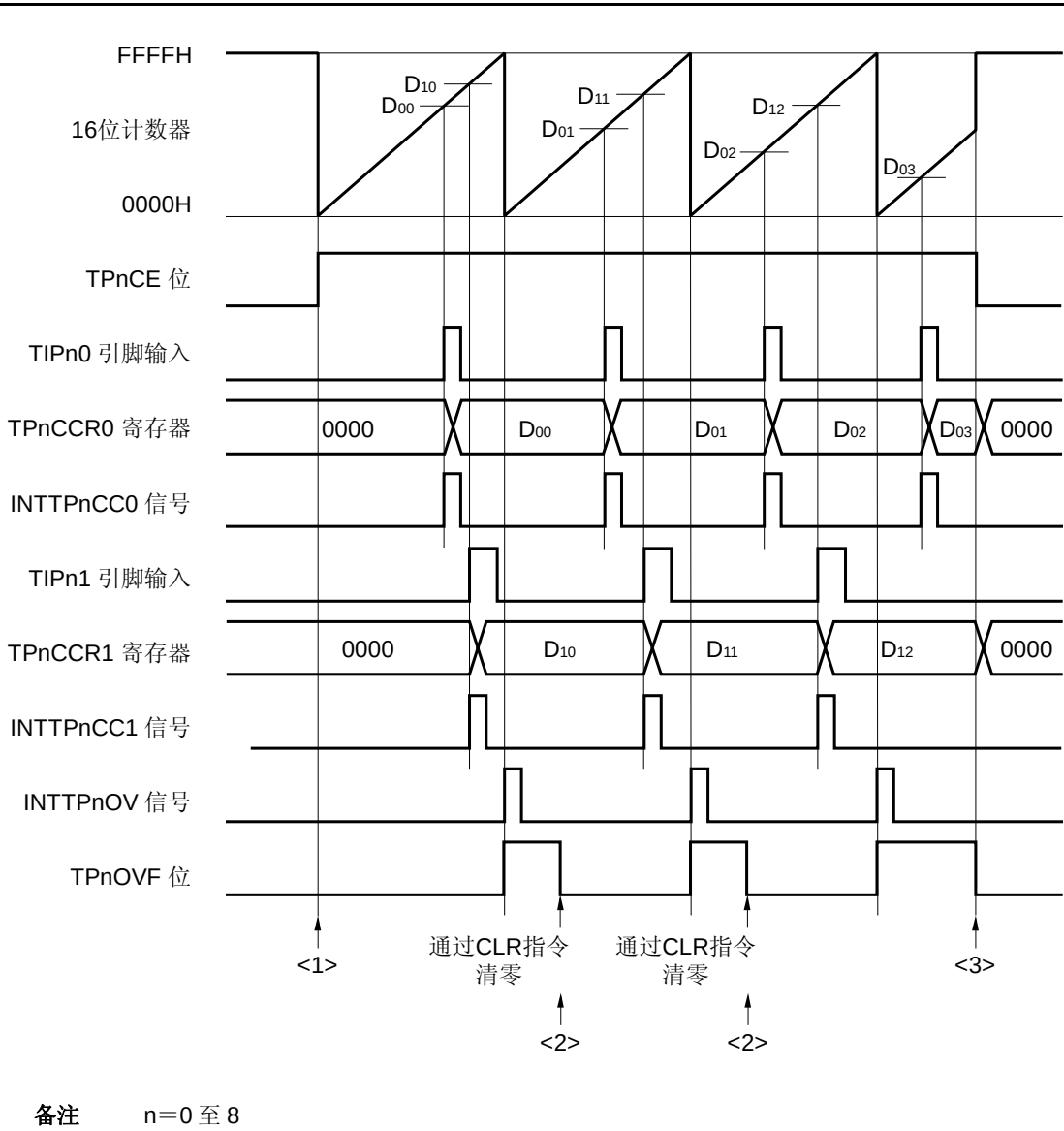
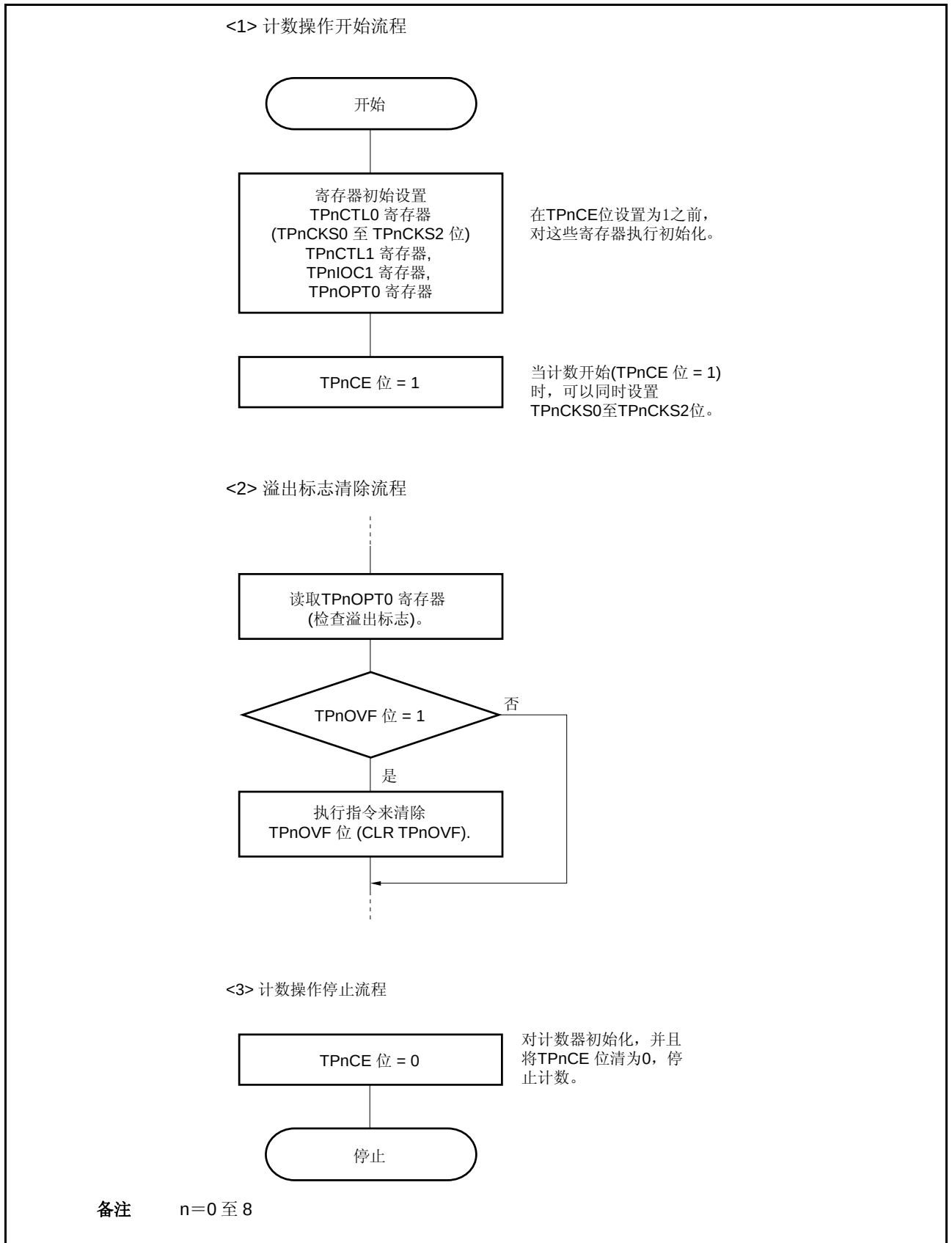


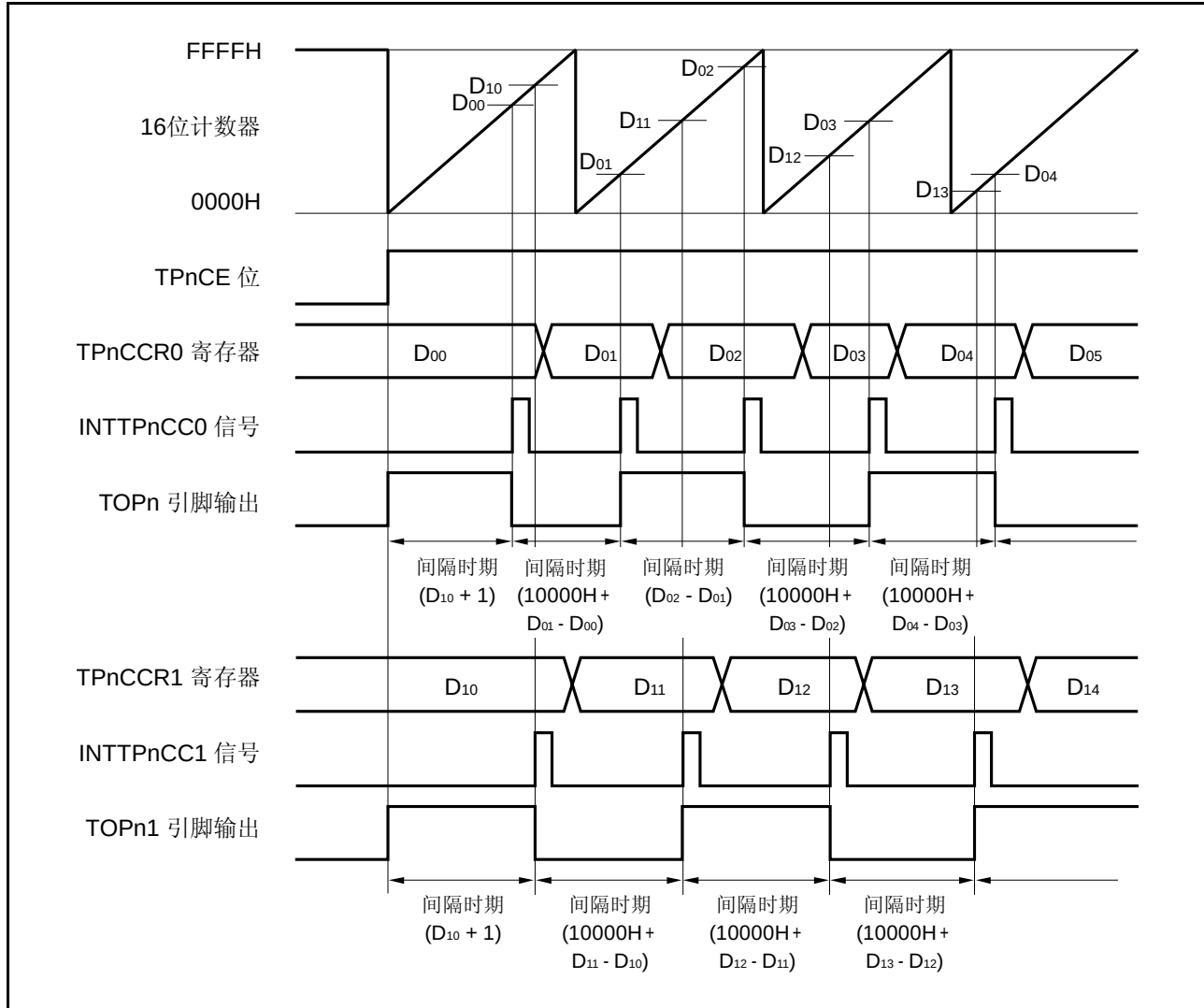
图 7-33. 自由运行定时器模式的软件处理流程（捕获功能）（2/2）



(2) 自由运行定时器模式的操作时序

(a) 比较寄存器的间隔定时操作

当 16 位定时器/事件计数器 P 用作间隔定时器，TPnCCRm 寄存器用作比较寄存器时，在每次检测到 INTTPnCCm 信号后，都需要通过软件设置一个比较值以产生下一个中断请求信号。



在自由运行定时器模式下进行时间间隔操作时，一个通道可以设置两个时间间隔。

要进行时间间隔操作，当检测到 INTTPnCCm 信号后，必须在中断服务程序中重置相应的 TPnCCRm 寄存器的值。

重置 TPnCCRm 寄存器的设置值可由下式计算，其中“Dm”是间隔时间：

比较寄存器默认值： $D_m - 1$

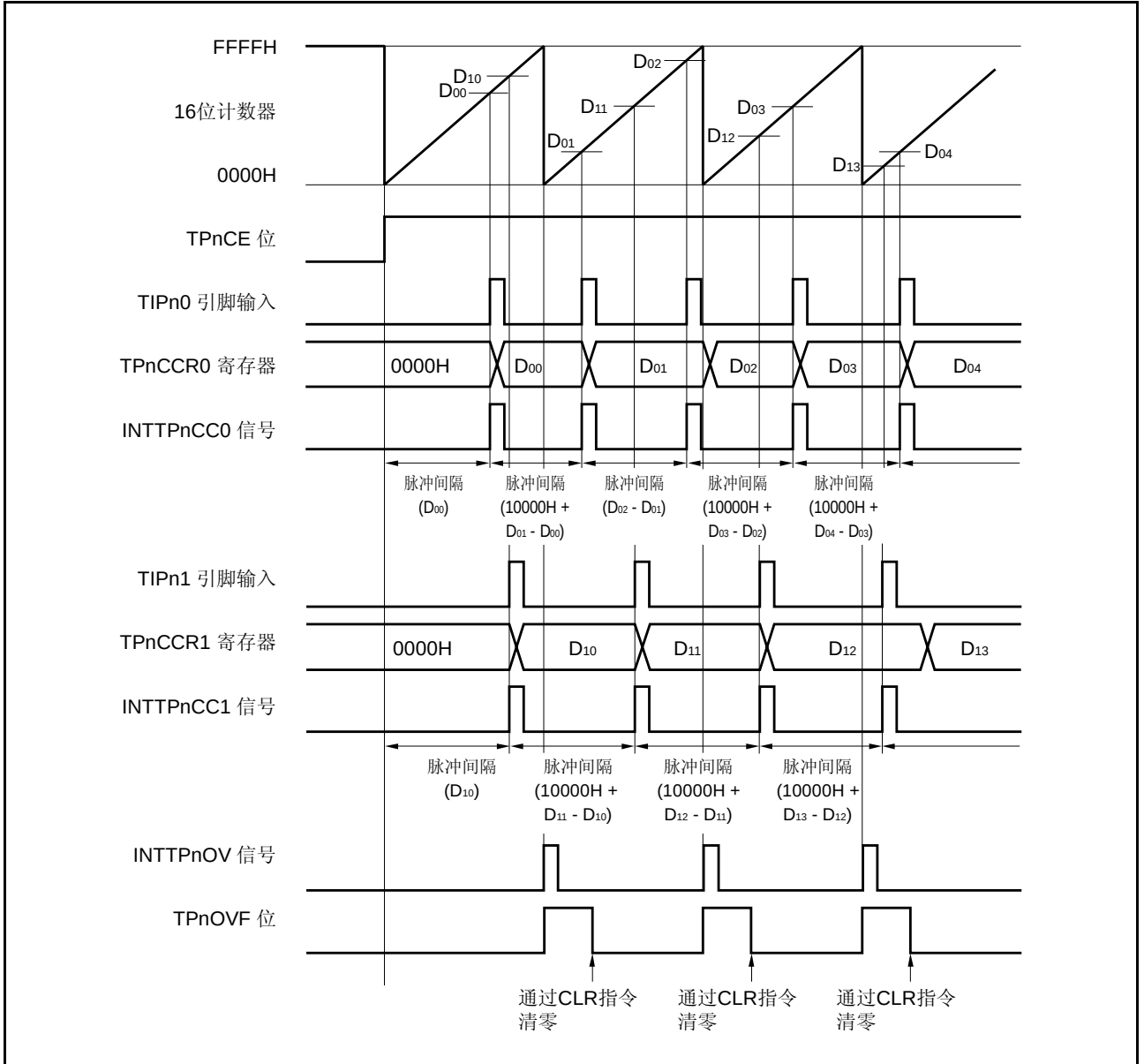
比较寄存器的第二次及其以后的设置值：上次设置值 + Dm

(若计算结果大于 FFFFH，则从结果中减去 10000H，并将此值设置给寄存器。)

备注 n=0 至 8
 m=0, 1

(b) 使用捕获寄存器进行脉冲宽度测量

在用 TPnCCRm 寄存器作为捕获寄存器进行脉宽测量时，每次检测到 INTTPnCCm 信号时，需要通过软件读取捕获寄存器并计算间隔时间。



在自由运行定时器模式下执行脉宽测量时，可以用一个通道测量两个脉冲宽度。

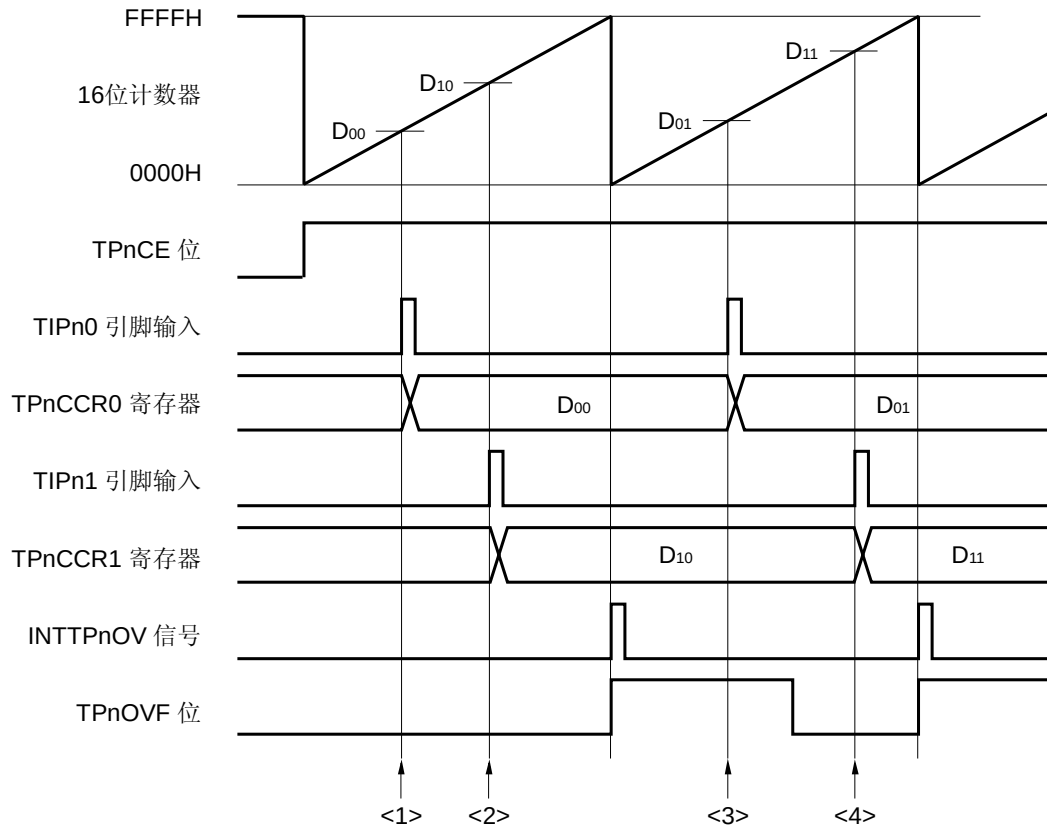
要测量脉冲宽度，可以与 INTTPnCCm 信号同步读取 TPnCCRm 寄存器的值，并计算读取值与先前读取值之间的差，得到脉冲宽度。

备注 n=0 至 8
 m=0, 1

(c) 当使用两个捕获寄存器时对溢出的处理

当使用两个捕获寄存器时必须格外小心。首先是一个不正确处理的示例，如下所示。

使用两个捕获寄存器时处理错误的示例



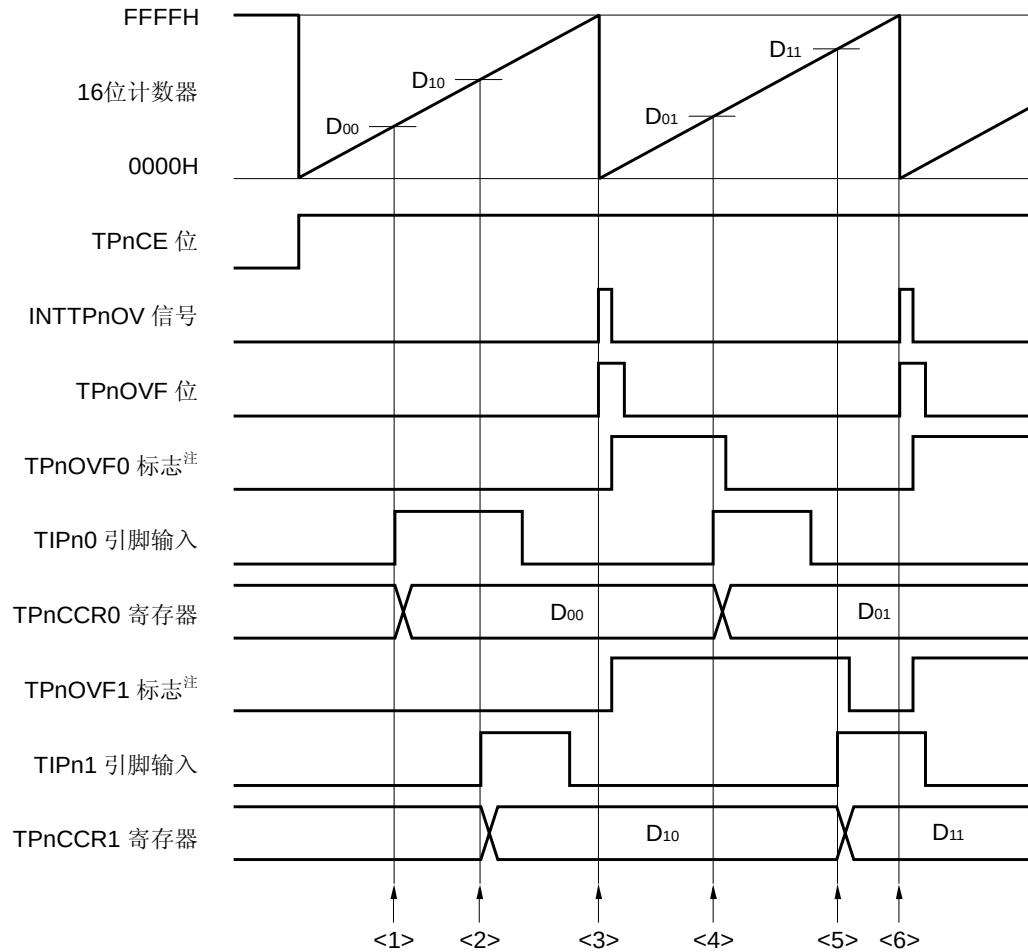
在自由运行定时器模式下测量两个脉冲宽度时，可能出现下列问题。

- <1> 读取 TPnCCR0 寄存器（TIPn0 引脚输入的默认值设置）。
- <2> 读取 TPnCCR1 寄存器（TIPn1 引脚输入的默认值设置）。
- <3> 读取 TPnCCR0 寄存器。
读溢出标志。若溢出标志为 1，将其清为 0。
因为溢出标志为 1，所以脉冲宽度可以用式 $(10000H + D_{01} - D_{00})$ 计算。
- <4> 读取 TPnCCR1 寄存器。
读溢出标志。因为此标志在<3>中被清零，所以，读取到 0。
因为溢出标志为 0，所以脉冲宽度可以用 $(D_{11} - D_{10})$ 计算。（不正确）

在使用两个捕获寄存器时，如果溢出标志由一个捕获寄存器清为 0，则另一个捕获寄存器可能得不到正确的脉冲宽度。

在使用两个捕获寄存器时，使用软件解决。关于如何使用软件的示例如下。

使用两个捕获寄存器时的示例（使用溢出中断）



注 由软件对内部 RAM 中的 TPnOVF0 和 TPnOVF1 标志进行设置。

<1> 读取 TPnCCR0 寄存器（TIPn0 引脚输入的默认值设置）。

<2> 读取 TPnCCR1 寄存器（TIPn1 引脚输入的默认值设置）。

<3> 发生溢出，在溢出中断服务程序中将 TPnOVF0 和 TPnOVF1 标志设置为 1，并将溢出标志清为 0。

<4> 读取 TPnCCR0 寄存器。

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

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

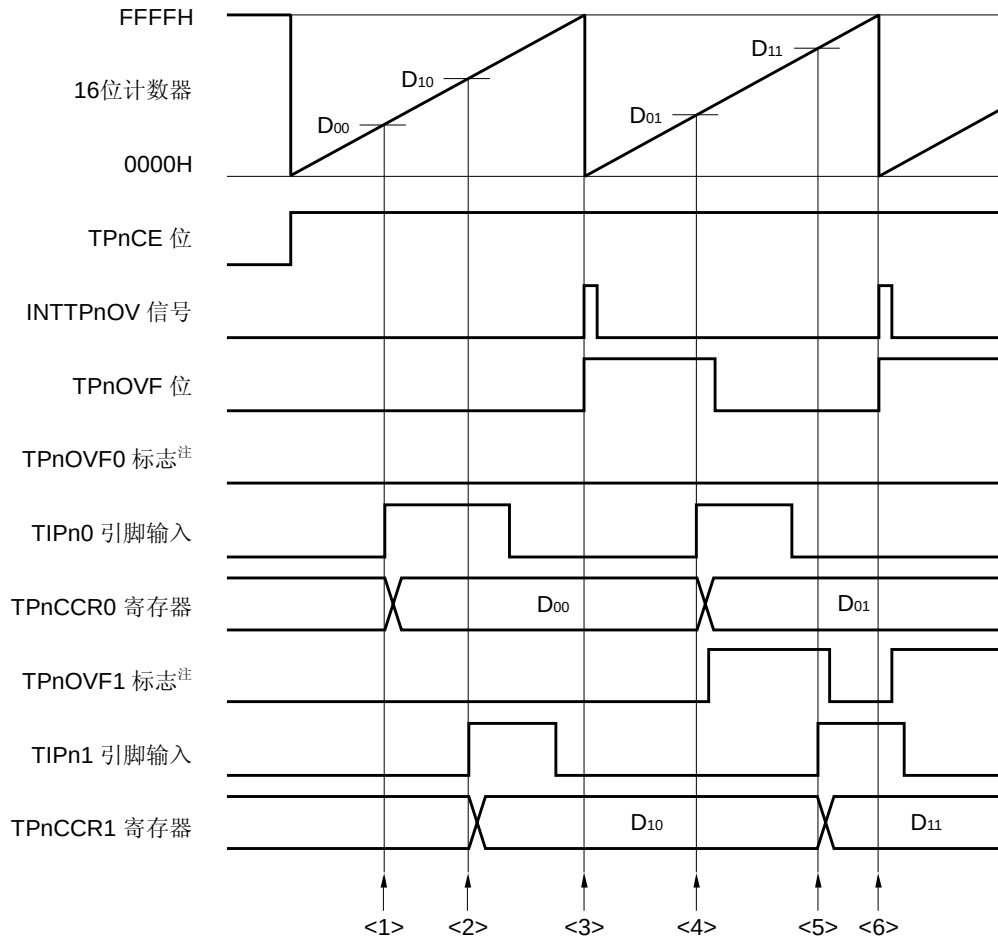
<5> 读取 TPnCCR1 寄存器。

读取 TPnOVF1 标志。如果 TPnOVF1 标志为 1，则将其清为 0（TPnOVF0 标志在<4>中已清零，而 TPnOVF1 标志仍然保持为 1）。

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

<6> 重复 <3>

使用两个捕获寄存器时的示例（不使用溢出中断）



注 由软件对内部 RAM 中的 TPnOVF0 和 TPnOVF1 标志进行设置。

<1> 读取 TPnCCR0 寄存器（TIPn0 引脚输入的默认值设置）。

<2> 读取 TPnCCR1 寄存器（TIPn1 引脚输入的默认值设置）。

<3> 发生溢出，软件不作任何处理。

<4> 读取 TPnCCR0 寄存器。

读溢出标志。若溢出标志为 1，则只把 TPnOVF1 标志置为 1，并把溢出标志清为 0。

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

<5> 读取 TPnCCR1 寄存器。

读溢出标志。因为溢出标志在<4>中被清零，所以读取到 0。

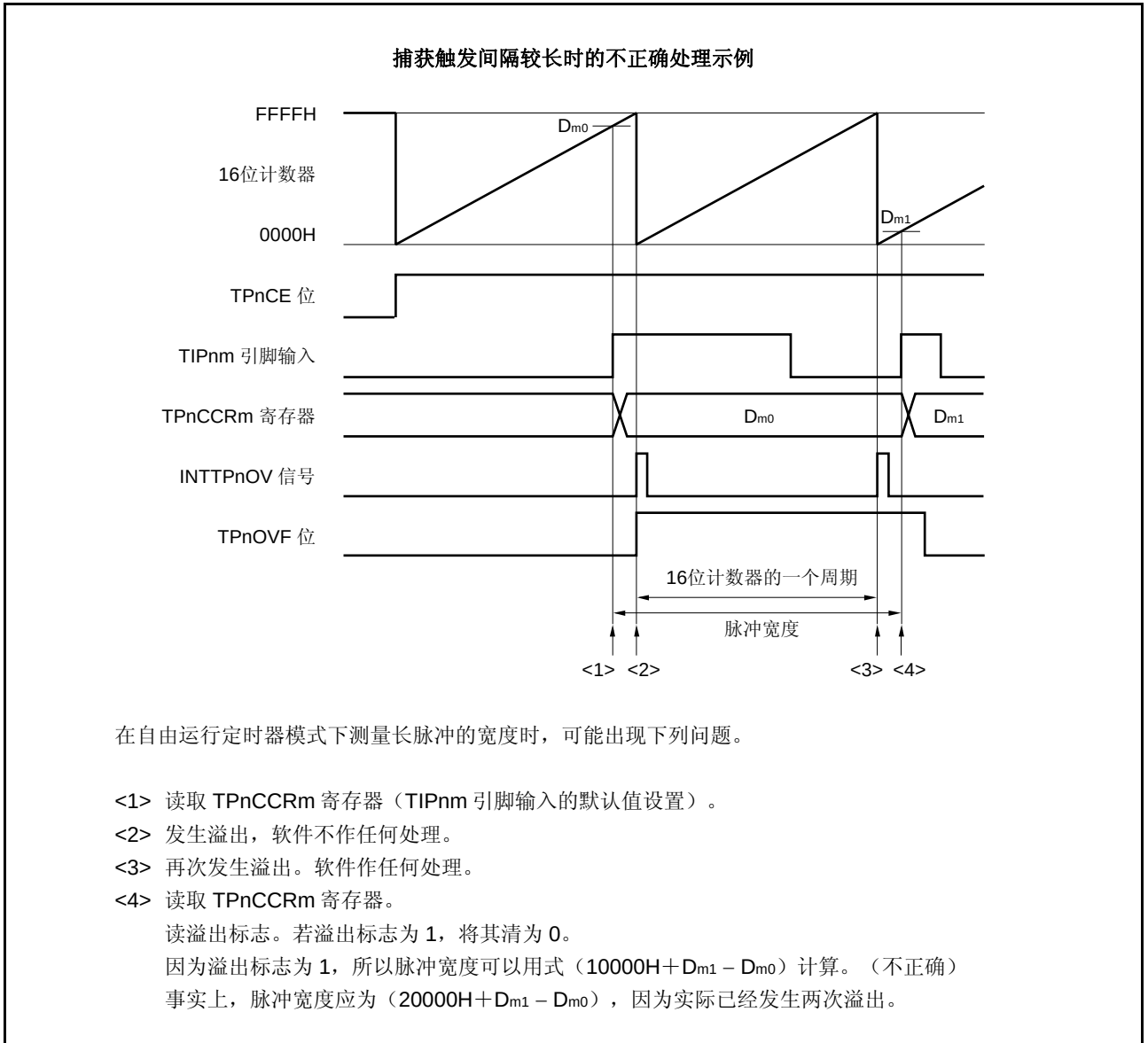
读取 TPnOVF1 标志。如果 TPnOVF1 标志为 1，则将其清为 0。

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

<6> 重复 <3>

(d) 当捕获触发间隔较长时的溢出处理

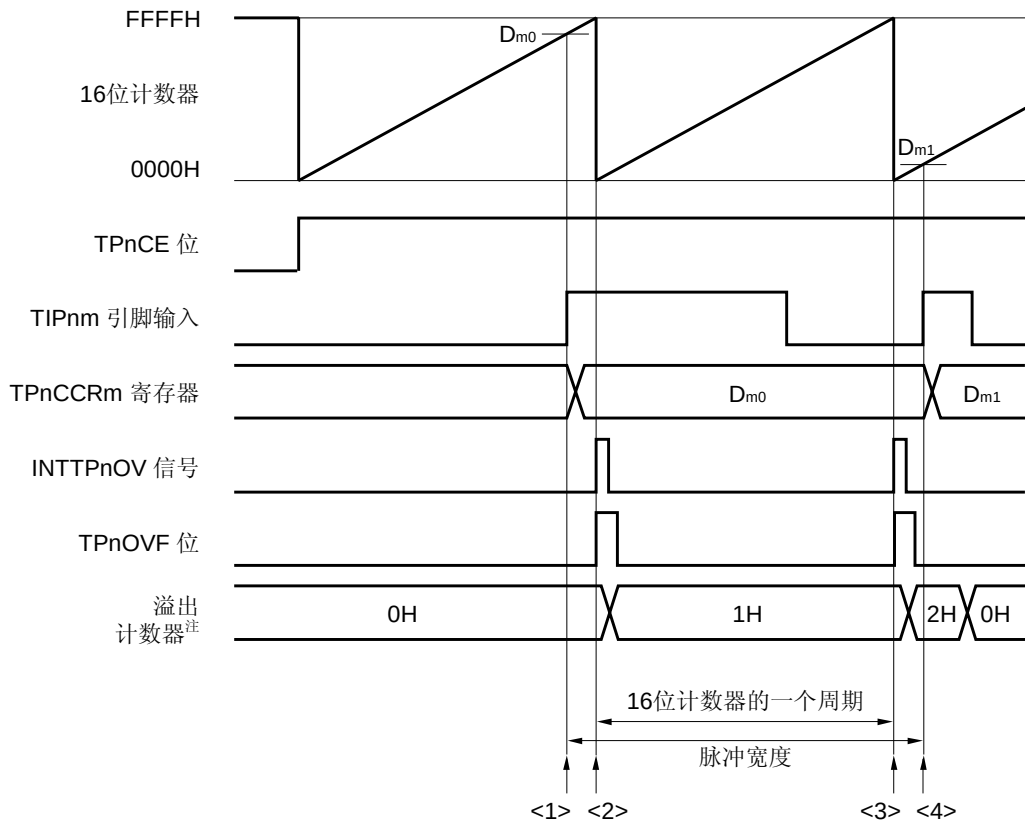
如果脉冲宽度大于 16 位计数器的一个计数循环周期，则必须加以注意，因为从第一个捕获触发信号到下一个触发信号之间可能发生多次溢出。首先是一个不正确处理的示例，如下所示。



如果在捕获触发间隔期间，发生两次或更多次的溢出，则无法获得正确的脉冲宽度。

如果捕获触发间隔时间较长时，应该减慢计数时钟以延长 16 位计数器的循环周期，或者使用软件来解决。关于如何使用软件，示例如下。

捕获触发间隔较长时的示例

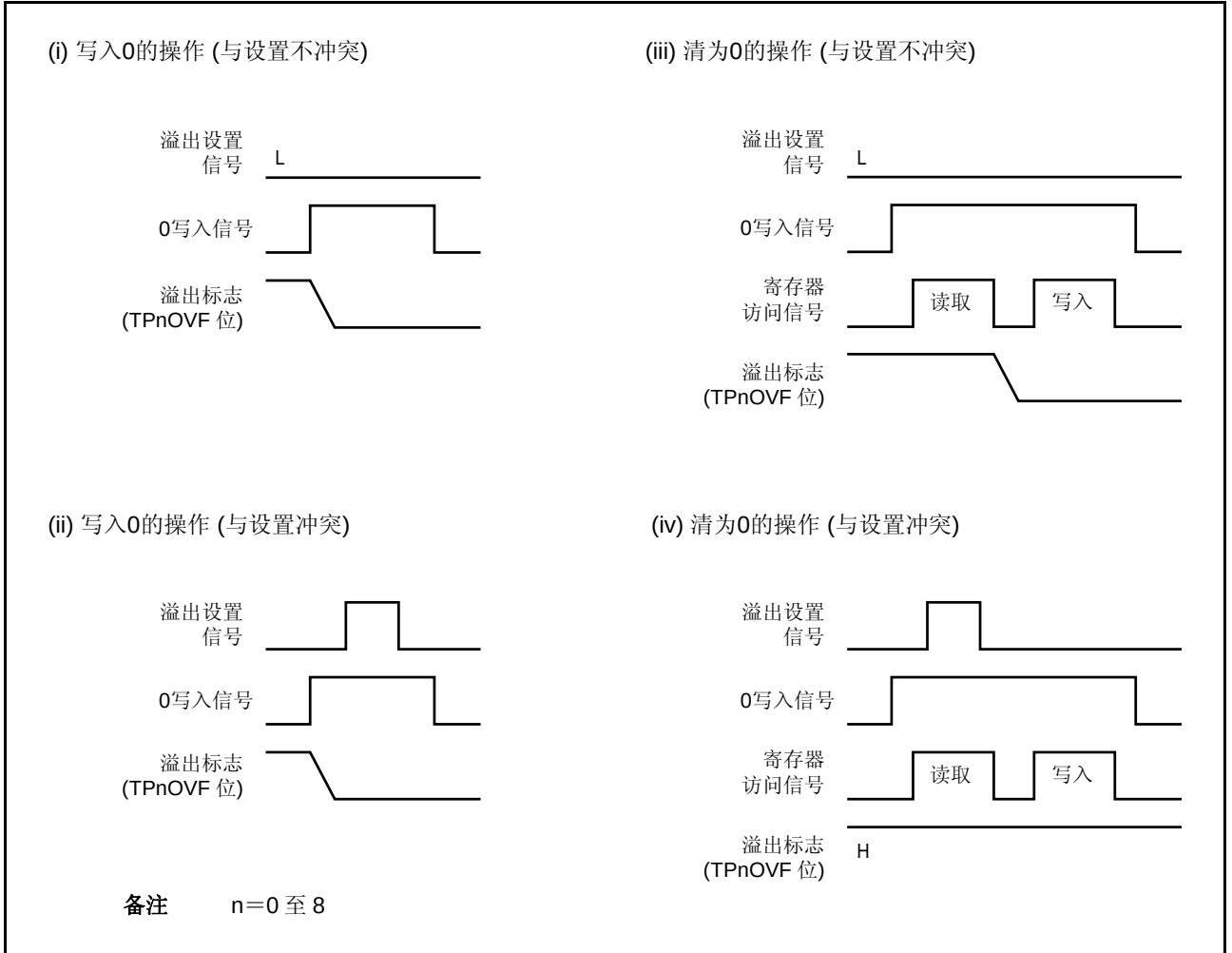


注 内部 RAM 中的溢出计数器由软件任意设置。

- <1> 读取 TPnCCRM 寄存器 (TIPnm 引脚输入的默认值设置)。
 - <2> 发生溢出, 在溢出中断服务程序中增加溢出计数器的值, 并将溢出标志清为 0。
 - <3> 再次发生溢出。在溢出中断服务程序中增加 (+1) 溢出计数器的值, 并把溢出标志清为 0。
 - <4> 读取 TPnCCRM 寄存器。
- 读取溢出计数器。
- 当溢出计数器为“N”时, 脉冲宽度可用式 $(N \times 10000H + D_{m1} - D_{m0})$ 计算。
- 在本例中, 脉冲宽度为 $(20000H + D_{m1} - D_{m0})$, 因为溢出实际发生了两次。
- 溢出计数器清零 (0H)。

（e）清除溢出标志

通过 CLR 指令将 TPnOVF 位清为 0，以及将 8 位数据（第 0 位是 0）写入 TPnOPT0 寄存器，可以将溢出标志清 0。要准确检测溢出，当 TPnOVF 位为 1 时读取 TPnOVF 位，然后用位操作指令将溢出标志清零。



要将溢出标志清为 0，应读取溢出标志，检查该标志如果已经被设置为 1，用 CLR 指令把它清零。若在未检查溢出标志是否为 1 就把 0 写入溢出标志中，则溢出的设置信息可能因为写入 0 而被清除（上图中的（ii））。因此，即使实际发生了溢出，软件也可能会判断为无溢出发生。

如果用 CLR 指令将溢出标志清为 0 时，CLR 指令的执行与当前发生的溢出相冲突，则在清零指令执行后溢出标志仍然会保持置位状态。

7.5.7 脉冲宽度测量模式（TPnMD2 至 TPnMD0 位=110）

在脉冲宽度测量模式下，当 TPnCTL0.TPnCE 位被置为 1 时，16 位定时器/事件计数器 P 开始计数。每次检测到 TIPnm 引脚输入信号的有效边沿时，16 位计数器的计数值就存储在 TPnCCRm 寄存器中，且 16 位计数器清零为 0000H。

在出现捕获中断请求信号（INTTPnCCm）之后，可以通过读取 TPnCCRm 寄存器来测量有效边沿的间隔。

选择 TIPn0 或 TIPn1 引脚作为捕获触发信号输入引脚。使用 TPnIOC1 寄存器将未使用的引脚指定为“无边沿检测”。

当使用外部时钟作为计数时钟时，测量 TIPn1 引脚的脉冲宽度，因为外部时钟固定在 TIPn0 引脚。此时，清除 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位为 00（捕获触发输入（TIPn0 引脚）：无边沿检测）。

图 7-34. 脉冲宽度测量模式的配置

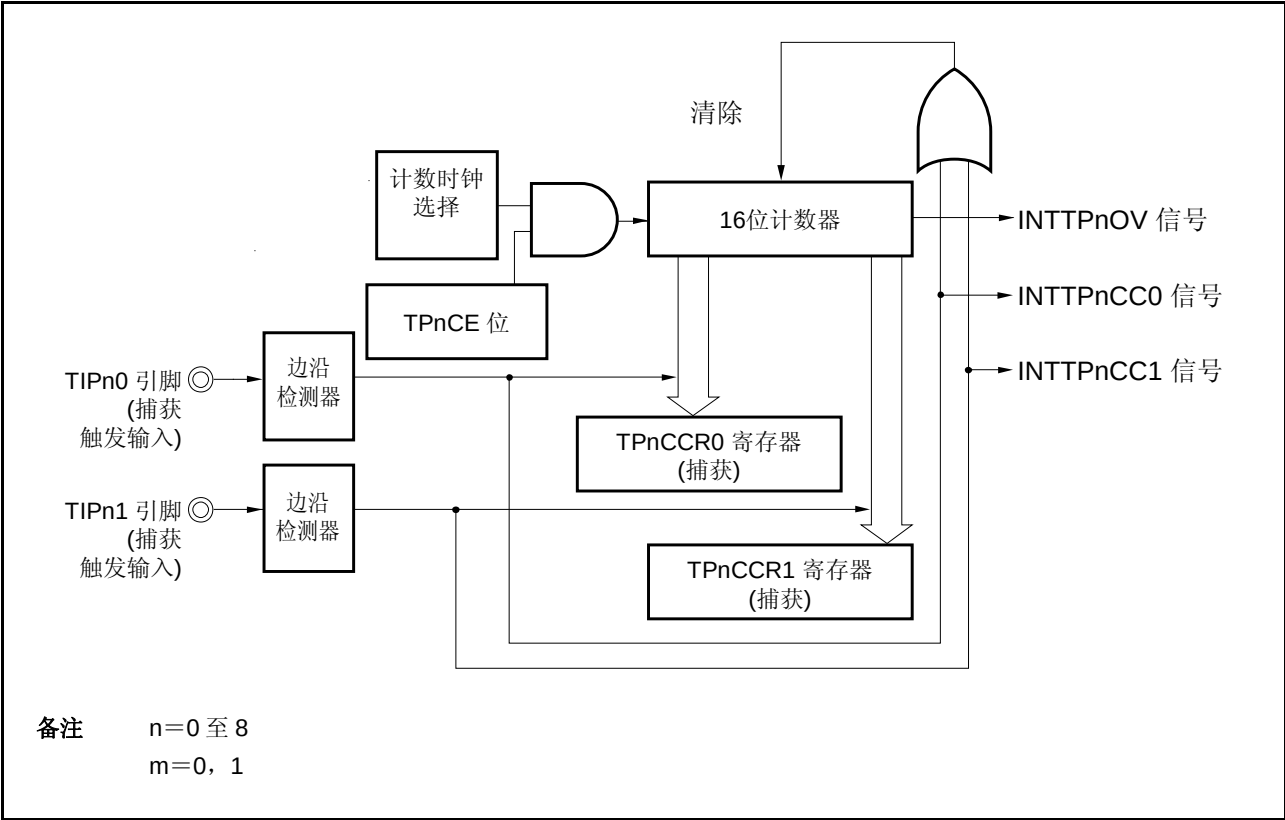
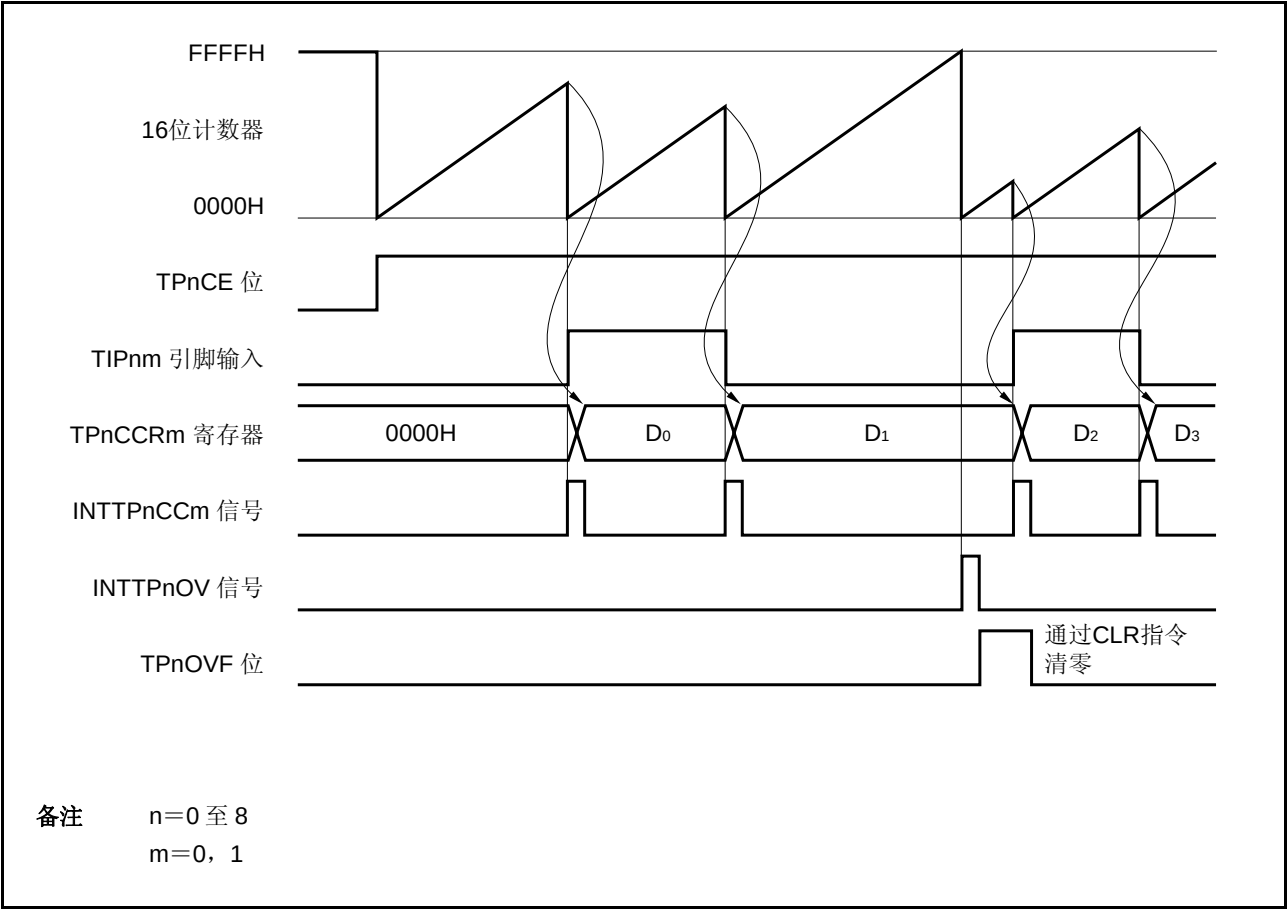


图 7-35. 脉冲宽度测量模式的基本时序



当 TPnCE 位被置为 1 时，16 位计数器开始计数。当检测到 TIPnm 引脚输入信号的有效边沿时，16 位计数器的计数值被存储到 TPnCCRm 寄存器中，16 位计数器清零为 0000H，并产生一个捕获中断请求信号（INTTPnCCm）。脉冲宽度的计算如下。

脉冲宽度 = 捕获值 × 计数时钟周期

当 16 位计数器计数到 FFFFH 时，即使没有有效边沿输入 TIPnm 引脚，在下一个计数时钟时，仍然会产生溢出中断请求信号（INTTPnOV），且计数器清零为 0000H，然后计数器继续计数。此时，溢出标志（TPnOPT0.TPnOVF 位）也被置为 1。使用软件执行 CLR 指令将溢出标志清为 0。

若溢出标志被设置为 1，则脉冲宽度的计算如下。

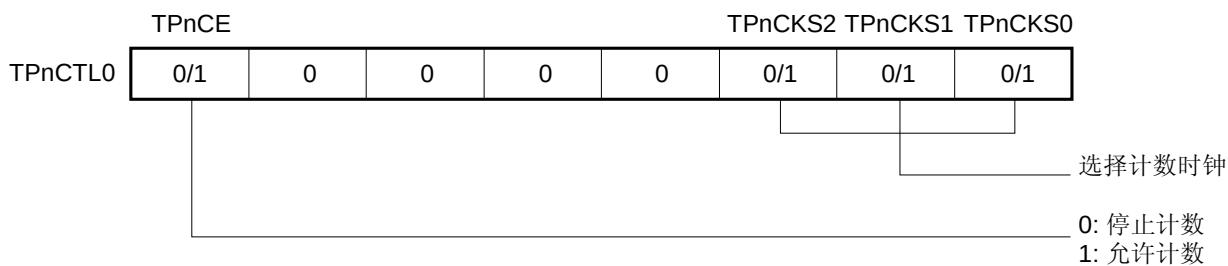
脉冲宽度 = (10000H × TPnOVF 位置位（1）的次数 + 捕获值) × 计数时钟周期

备注 n=0 至 8
m=0, 1

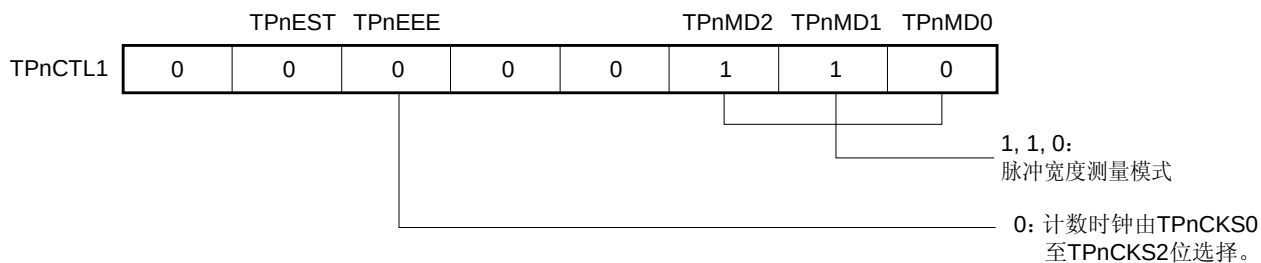
图 7-36. 脉冲宽度测量模式时的寄存器设置 (1/2)

<R>

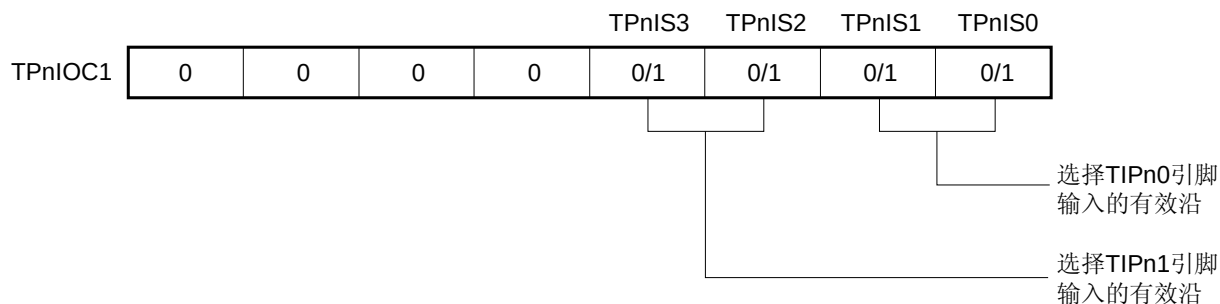
(a) TMPn 控制寄存器 0 (TPnCTL0)



(b) TMPn 控制寄存器 1 (TPnCTL1)



(c) TMPn 输入/输出控制寄存器 1 (TPnIOC1)



(d) TMPn 选项寄存器 0 (TPnOPT0)

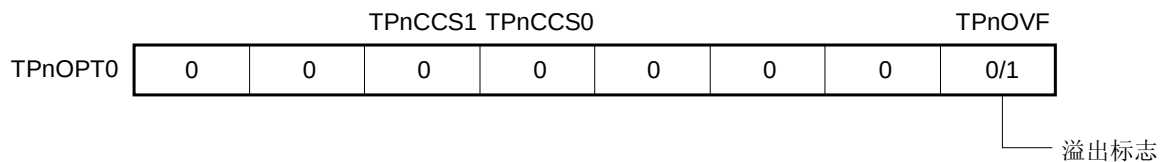


图 7-36. 脉冲宽度测量模式时的寄存器设置（2/2）

(e) TMPn 计数器读取缓冲寄存器（TPnCNT）

通过读取 TPnCNT 寄存器，可以读出 16 位计数器的值。

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

当检测到 TIPnm 引脚输入的有效边沿时，这些寄存器保存 16 位计数器的当前计数值。

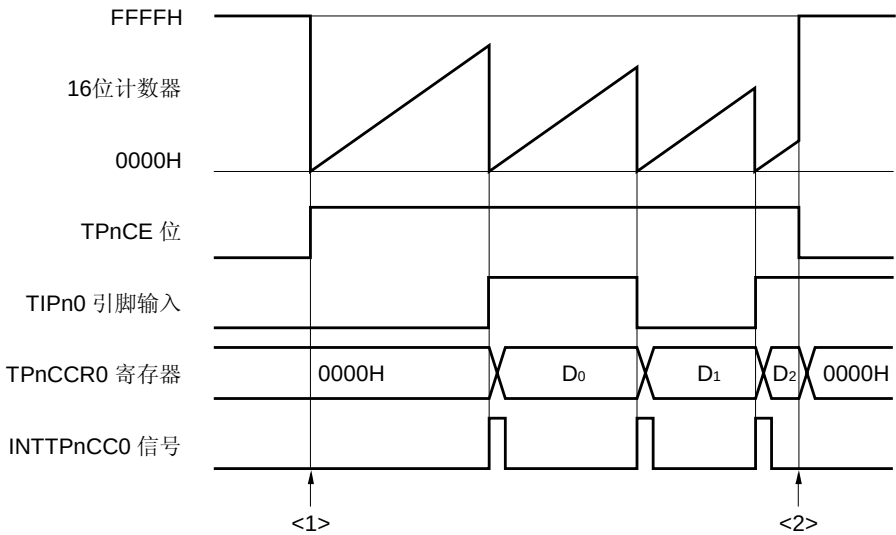
备注

1. 在脉冲宽度测量模式下，不使用 TMPn 输入/输出控制寄存器 0（TPnIOC0）和 TMPn 输入/输出控制寄存器 2（TPnIOC2）。
2. n=0 至 8
m=0, 1

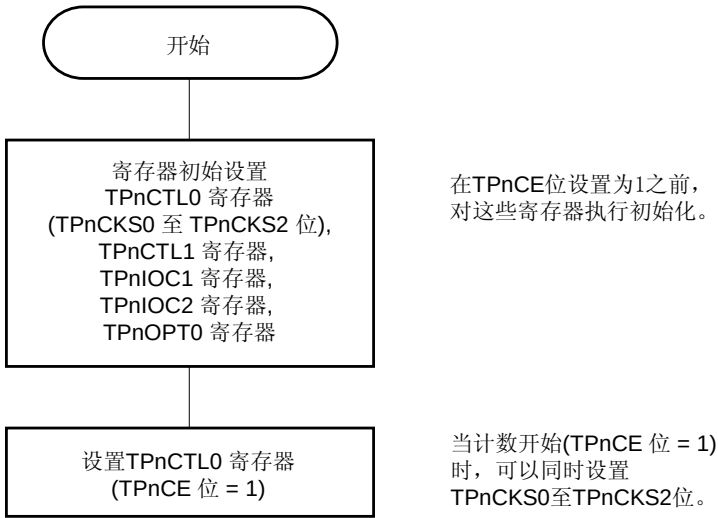
<R>

(1) 脉冲宽度测量模式的流程图

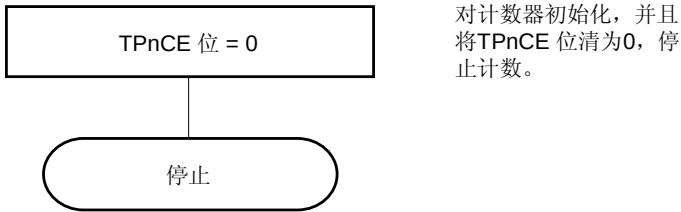
图 7-37. 脉冲宽度测量模式时的软件处理流程



<1> 计数操作开始流程



<2> 计数操作停止流程

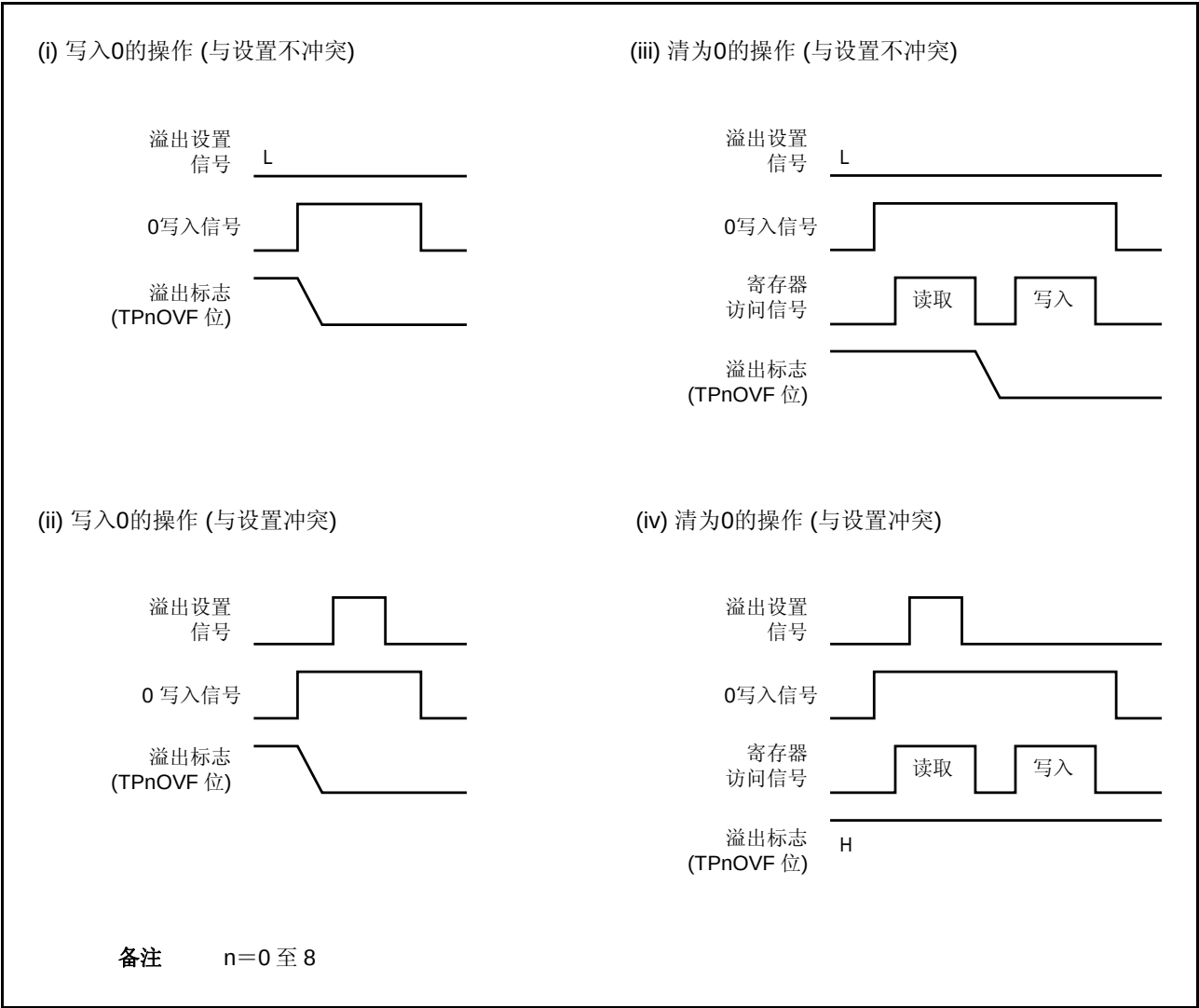


备注 n=0 至 8

(2) 脉冲宽度测量模式的操作时序

(a) 清除溢出标志

通过 CLR 指令将 TPnOVF 位清为 0，以及将 8 位数据（第 0 位是 0）写入 TPnOPT0 寄存器，可以将溢出标志清 0。要准确检测溢出，当 TPnOVF 位为 1 时读取 TPnOVF 位，然后用位操作指令将溢出标志清零。



要将溢出标志清为 0，应读取溢出标志，检查该标志如果已经被设置为 1，用 CLR 指令把它清零。若在未检查溢出标志是否为 1 就把 0 写入溢出标志中，则溢出的设置信息可能因为写入 0 而被清除（上图中的（ii））。因此，即使实际发生了溢出，软件也可能会判断为无溢出发生。

如果用 CLR 指令将溢出标志清为 0 时，CLR 指令的执行与当前发生的溢出相冲突，则在清零指令执行后溢出标志仍然会保持置位状态。

7.5.8 定时器输出操作

下表显示了 TOPn0 和 TOPn1 引脚的操作和输出电平。

表 7-4. 各种模式下的定时器输出控制

操作模式	TOPn1 引脚	TOPn0 引脚
间隔计时器模式	方波输出	
外部事件计数模式	方波输出	—
外部触发脉冲输出模式	外部触发脉冲输出	方波输出
单次触发脉冲输出模式	单次触发脉冲输出	
PWM 输出模式	PWM（脉宽调制信号）输出	
自由定时器模式	方波输出（只有使用比较功能时可用）	
脉冲宽度测量模式	—	

备注 n=0 至 8

表 7-5. 定时器输出控制位对 TOPn0 和 TOPn1 引脚影响的真值表

TPnIOC0.TPnOLm 位	TPnIOC0.TPnOEm 位	TPnCTL0.TPnCE 位	TOPnm 引脚电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前为低电平，开始计数后为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平，开始计数后为低电平

备注 n=0 至 8
 m=0, 1

7.6 选择器功能

在 V850ES/JJ3 单片机中，TMP 的捕获触发输入可以选择，是经过端口/定时器复用功能引脚的输入信号，还是经过外设输入/输出 (TMP/UARTA) 输入信号。

该功能可以实现下面的内容。

- TMP1 的 TIP10 和 TIP11 输入信号，可从端口/定时器复用功能引脚 (TIP10 和 TIP11 引脚) 和 UARTA 接收复用功能引脚 (RXDA0 和 RXDA1 引脚) 中选择。同样，TMP3 的 TIP31 输入信号，可从端口/定时器复用功能引脚 (TIP31 引脚) 和 UARTA 接收复用功能引脚 (RXDA3 引脚) 中选择。
→ 当选择用于 UARTA0、UARTA1 或 UARTA3 的 RXDA0、RXDA1 信号或 RXDA3 信号时，可以计算 UARTA LIN 接收传输率的波特率误差。

- 注意事项**
1. 当使用选择器功能时，请确保设置将要连接到捕获触发输入的 TMP 的端口/定时器复用功能引脚。
 2. 在设置选择器功能之前，禁止将要连接的外设输入/输出 (TMP/UARTA)。

捕获触发输入可以使用下列寄存器来选择。

(1) 选择器操作控制寄存器 0（SELCNT0）

SELCNT0 寄存器是 8 位寄存器，用于选择定时器 TMP1 和 TMP3 的捕获触发。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后：00H R/W 地址： FFFFF308H

	7	<6>	5	<4>	<3>	2	1	0
SELCNT0	0	ISEL6	0	ISEL4	ISEL3	0	0	0

ISEL6	TIP31 输入信号的选择(TMP3)
0	TIP31 引脚输入
1	RXDA3 引脚输入

ISEL4	TIP11输入信号的选择 (TMP1)
0	TIP11 引脚输入
1	RXDA1 引脚输入

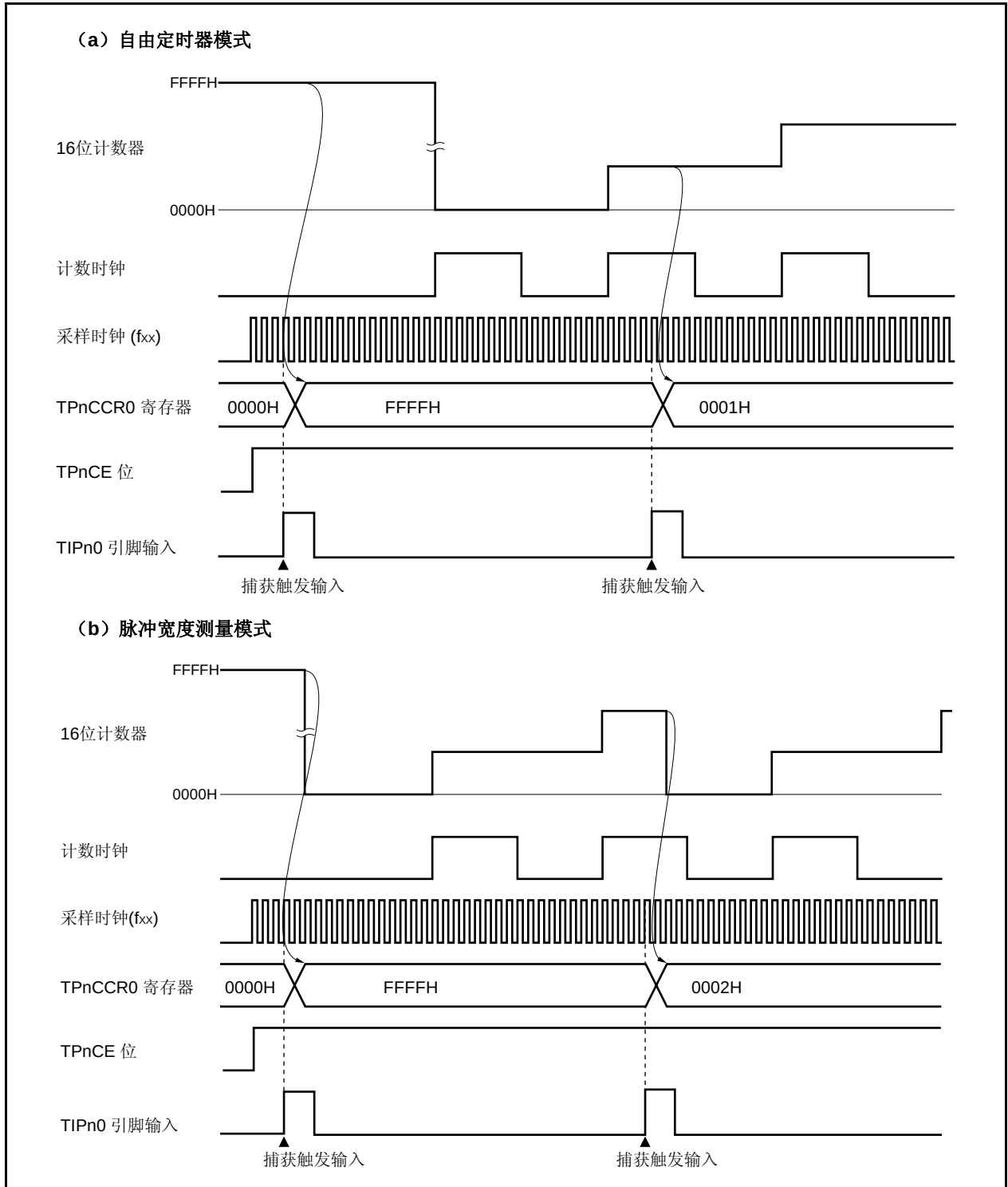
ISEL3	TIP10 输入信号的选择 (TMP1)
0	TIP10 引脚输入
1	RXDA0 引脚输入

- 注意事项
1. 当设置 ISEL3、ISEL4 或 ISEL6 位为 1 时，请确保为捕获触发输入设置相应的复用功能引脚。
 2. 请确保将第 7 位、第 5 位、第 2 位至第 0 位清除为“0”。

7.7 注意事项

(1) 捕获操作

当使用捕获操作，并且选择低速时钟作为计数时钟时，如果在设置 TPnCE 位为 1 之后，立即输入捕获触发，那么捕获到 TPnCCR0 和 TPnCCR1 寄存器中的可能是 FFFFH 而不是 0000H。



第八章 16 位定时器/事件计数器 Q (TMQ)

定时器 Q (TMQ) 是 16 位定时器/事件计数器。

V850ES/JJ3 单片机内置 TMQ0。

8.1 概述

定时器 TMQ0 的概况如下所示：

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

8.2 功能

定时器 TMQ0 具有下列功能：

- 间隔定时器
- 外部事件计数器
- 外部触发脉冲输出
- 单次触发脉冲输出
- PWM 输出
- 自由运行定时器
- 脉冲宽度测量

8.3 配置

TMQ0 包括下列硬件:

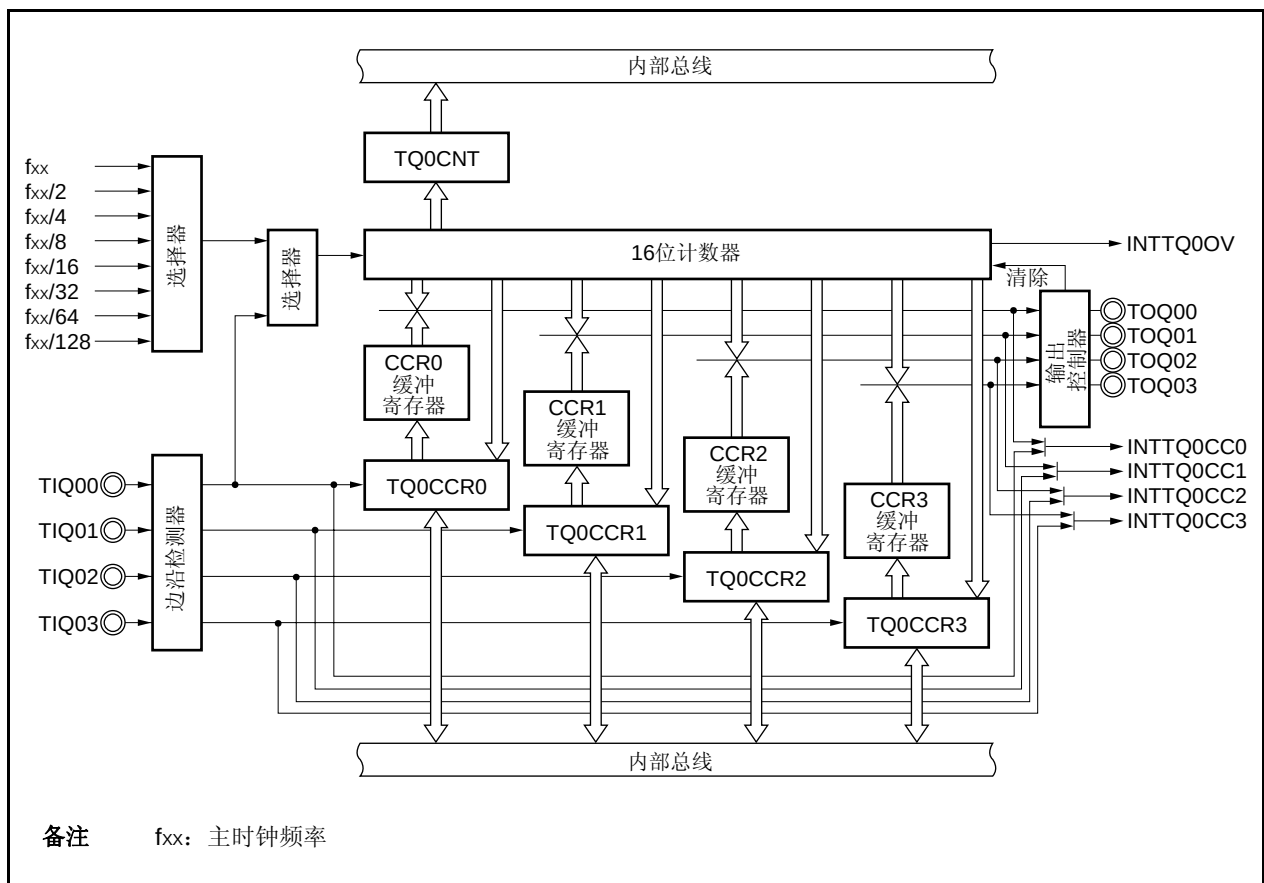
表 8-1. TMQ0 的配置

项目	配置
定时器寄存器	16 位计数器
寄存器	TMQ0 捕获/比较寄存器 0 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 ^{註1} 至 TIQ03 引脚)
定时器输出	4 (TOQ00 至 TOQ03 引脚)
控制寄存器 ^{註2}	TMQ0 控制寄存器 0, 1 (TQ0CTL0, TQ0CTL1) TMQ0 输入/输出 控制寄存器 0 至 2 (TQ0IOC0 至 TQ0IOC2) TMQ0 选项寄存器 0 (TQ0OPT0)

注

1. TIQ00 引脚功能可以复用为捕获触发、外部事件计数以及外部触发的信号输入引脚。
2. 当使用 TIQ00 至 TIQ03 和 TOQ00 至 TOQ03 引脚功能时，参见表 4-19 端口引脚用作复用功能引脚。

图 8-1. TMQ0 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计数。

该计数器的计数值可以通过读取 TQ0CNT 寄存器得到。

当 TQ0CTL0.TQ0CE 位 = 0 时, 16 位计数器的值为 FFFFH。如果此时读取 TQ0CNT 寄存器, 则读回的值为 0000H。

系统复位将 TQ0CE 位清 0, 因此, 16 位计数器设置为 FFFFH。

(2) CCR0 缓冲寄存器

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

当 TQ0CCR0 寄存器用作比较寄存器时, 写入 TQ0CCR0 寄存器的值会传送到 CCR0 缓冲寄存器。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值相匹配时, 就会产生一个比较匹配中断请求信号 (INTTQ0CC0)。

CCR0 缓冲寄存器不能直接读写。

CCR0 缓冲寄存器在复位后清零为 0000H, TQ0CCR0 寄存器清零为 0000H。

(3) CCR1 缓冲寄存器

CCR1 缓冲寄存器是一个 16 位比较寄存器, 用于比较 16 位计数器的计数值。

当 TQ0CCR1 寄存器用作比较寄存器时, 写入 TQ0CCR1 寄存器的值会传送到 CCR1 缓冲寄存器。当 16 位计数器的计数值和 CCR1 缓冲寄存器的值匹配时, 就会产生一个比较匹配中断请求信号 (INTTQ0CC1)。

CCR1 缓冲寄存器不能直接读写。

CCR1 缓冲寄存器在复位后清零为 0000H, TQ0CCR1 寄存器清零为 0000H。

(4) CCR2 缓冲寄存器

CCR2 缓冲寄存器是一个 16 位比较寄存器, 用于比较 16 位计数器的计数值。

当 TQ0CCR2 寄存器用作比较寄存器时, 写入 TQ0CCR2 寄存器的值会传送到 CCR2 缓冲寄存器。当 16 位计数器的计数值和 CCR2 缓冲寄存器的值匹配时, 就会产生一个比较匹配中断请求信号 (INTTQ0CC2)。

CCR2 缓冲寄存器不能直接读写。

CCR2 缓冲寄存器在复位后清零为 0000H, TQ0CCR2 寄存器清零为 0000H。

(5) CCR3 缓冲寄存器

CCR3 缓冲寄存器是一个 16 位比较寄存器, 用于比较 16 位计数器的计数值。

当 TQ0CCR3 寄存器用作比较寄存器时, 写入 TQ0CCR3 寄存器的值会传送到 CCR3 缓冲寄存器。当 16 位计数器的计数值与 CCR3 缓冲寄存器的值相匹配时, 就会产生一个比较匹配中断请求信号 (INTTQ0CC3)。

CCR3 缓冲寄存器不能直接读写。

CCR3 缓冲寄存器在复位后清零为 0000H, TQ0CCR3 寄存器清零为 0000H。

(6) 边沿检测器

该电路检测输入到 TIQ00 和 TIQ03 引脚信号的有效边沿。通过设置 TQ0IOC1 和 TQ0IOC2 寄存器, 可以选择各种有效沿: 无边沿、上升沿、下降沿、上升/下降双边沿作为有效沿。

(7) 输出控制器

该电路控制 TOQ00 至 TOQ03 引脚的输出。输出控制器由 TQ0IOC0 寄存器控制。

(8) 选择器

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

8.4 寄存器

以下寄存器控制 TMQ0：

- TMQ0 控制寄存器 0 (TQ0CTL0)
- TMQ0 控制寄存器 1 (TQ0CTL1)
- TMQ0 输入/输出控制寄存器 0 (TQ0IOC0)
- TMQ0 输入/输出控制寄存器 1 (TQ0IOC1)
- TMQ0 输入/输出控制寄存器 2 (TQ0IOC2)
- TMQ0 选项寄存器 0 (TQ0OPT0)
- TMQ0 捕获/比较寄存器 0 (TQ0CCR0)
- TMQ0 捕获/比较寄存器 1 (TQ0CCR1)
- TMQ0 捕获/比较寄存器 2 (TQ0CCR2)
- TMQ0 捕获/比较寄存器 3 (TQ0CCR3)
- TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

备注 当使用 TIQ00 至 TIQ03 和 TOQ00 至 TOQ03 引脚的功能时，参见表 4-19 端口引脚用作复用功能引脚。

(1) TMQ0 控制寄存器 0（TQ0CTL0）

TQ0CTL0 寄存器是 8 位寄存器，用于控制定时器 TMQ0 的操作。

该寄存器可以按字节读写或按位读写。

复位输入将该寄存器设为 00H。

随时可以用软件将相同的值写入 TQ0CTL0 寄存器。

复位后: 00H		R/W	地址: FFFF540H				
<7>		6	5	4	3	2	1 0
TQ0CTL0	TQ0CE	0	0	0	0	TQ0CKS2	TQ0CKS1 TQ0CKS0
TQ0CE		TMQ0 操作控制					
0		TMQ0 操作禁止 (TMQ0异步复位 ^注)。					
1		TMQ0 操作允许。TMQ0操作开始。					
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
1. 当 TQ0CE 位=0 时，设置 TQ0CKS2 至 TQ0CKS0 位为“1”。
当 TQ0CE 位的值从 0 变为 1 时，可以同时设定 TQ0CKS2 至 TQ0CKS0 位。
 2. 请确保将第 3 位至第 6 位清为“0”。

备注 f_{xx}: 主时钟频率

(2) TMQ0 控制寄存器 1 (TQ0CTL1)

TQ0CTL1 寄存器是 8 位寄存器，用于控制定时器 TMQ0 的操作。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: 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一个有效的信号。 · 在单次脉冲输出模式: 将1写入TQ0EST位作为触发, 输出一个单次脉冲。 · 在外部触发脉冲输出模式: 将1写入TQ0EST位作为触发, 输出一个PWM脉冲。

TQ0EEE	计数时钟选择
0	禁止使用外部事件计数输入操作。 (对由TQ0CTL0.TQ0CK0至 TQ0CK2位所选的计数时钟计数。)
1	允许使用外部事件计数输入操作。 (对外部事件计数输入信号的有效沿计数。)
TQ0EEE位选择使用内部时钟计数还是外部事件计数输入的有效沿计数。	

TQ0MD2	TQ0MD1	TQ0MD0	定时器模式选择
0	0	0	内部定时器模式
0	0	1	外部事件计数模式
0	1	0	外部触发脉冲输出模式
0	1	1	单次触发脉冲输出模式
1	0	0	PWM 输出模式
1	0	1	自由运行定时器模式
1	1	0	脉冲宽度测量模式
1	1	1	设置禁止

- 注意事项**
1. 只有在外触发脉冲输出模式或单次触发脉冲输出模式下，TQ0EST 位有效。在其它模式下，向该位写入 1 的操作都被忽略。
 2. 不论 TQ0EEE 位的取值如何，在外部事件计数模式下，都会选择外部事件输入。
 3. TQ0CTL0.TQ0CE 位 = 0 时，设置 TQ0EEE 和 TQ0MD2 至 TQ0MD0 位（TQ0CE 位 = 1 时，可以写入相同的值）。在 TQ0CE 位 = 1 时，重写操作是无法保证的，若重写发生错误，应把 TQ0CE 位清为 0，再重新对这些位进行设定。
 4. 请确保将第 3 位，第 4 位和第 7 位清为“0”。

(3) TMQ0 输入/输出控制寄存器 0 (TQ0IOC0)

TQ0IOC0 寄存器是 8 位寄存器，用于控制定时器的输出（TOQ00 至 TOQ03 引脚）。
该寄存器可按字节读写或按位读写。
复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF542H

7	<6>	5	<4>	3	<2>	1	<0>
TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0

TQ0OLm	TOQ0m引脚输出电平设置 (m = 0至3) ^注
0	高电平时，TOQ0m引脚开始输出
1	低电平时，TOQ0m引脚开始输出

TQ0OEm	TOQ0m引脚输出设置(m = 0至3)
0	定时器输出禁止 · 当TQ0OLm位= 0时: 低电平从TOQ0m引脚输出。 · 当TQ0OLm位= 1时: 高电平从TOQ0m引脚输出。
1	定时器输出允许 (从TOQ0m引脚输出方波)。

注 由 TQ0OLm 位指定的定时器输出引脚（TOQ0m）的输出电平如下所示。

· 当TQ0OLm 位 = 0时



· 当TQ0OLm 位 = 1时



- 注意事项
1. 当 TQ0CTL0.TQ0CE 位=0 时，重新写入 TQ0OLm 和 TQ0OEm 位（当 TQ0CE 位=1 时，可以写入相同值）。如果重新写入不能正确执行，将 TQ0CE 位清为 0，然后重新设定该位。
 2. 即使在 TQ0CE 和 TQ0OEm 位均为 0 时对 TQ0OLm 位进行操作，TOQ0m 引脚的输出电平也会改变。

备注 m=0 至 3

(4) TMQ0 输入/输出控制寄存器 1 (TQ0IOC1)

TQ0IOC1 寄存器是 8 位寄存器，用于控制捕获触发输入信号 (TIQ00 至 TIQ03 引脚) 的有效边沿。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF543H

	7	6	5	4	3	2	1	0
TQ0IOC1	TQ0IS7	TQ0IS6	TQ0IS5	TQ0IS4	TQ0IS3	TQ0IS2	TQ0IS1	TQ0IS0

TQ0IS7	TQ0IS6	捕获触发输入信号(TIQ03 引脚) 有效沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

TQ0IS5	TQ0IS4	捕获触发输入信号(TIQ02 引脚) 有效沿检测
0	0	无边沿检测 (捕获操作无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

TQ0IS3	TQ0IS2	捕获触发输入信号(TIQ01 引脚) 有效沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

TQ0IS1	TQ0IS0	捕获触发输入信号(TIQ00 引脚) 有效沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

- 注意事项**
1. 当 TQ0CTL0.TQ0CE 位 = 0 时，重新写入 TQ0IS7 至 TQ0IS0 位（当 TQ0CE 位 = 1 时，可以写入相同值）。如果重新写入不能正确执行，将 TQ0CE 位清为 0，然后重新进行位的设定。
 2. 只有在自由运行定时器模式和脉冲宽度测量模式下，TQ0IS7 至 TQ0IS0 位才有效。在所有其它模式下，无法进行捕获操作。

(5) TMQ0 输入/输出控制寄存器 2 (TQ0IOC2)

TQ0IOC2 寄存器是 8 位寄存器，用于控制外部事件计数器输入信号 (TIQ00 引脚) 和外部触发输入信号 (TIQ00 引脚) 的有效边沿。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H	R/W	地址:	FFFFFF544H							
			7	6	5	4	3	2	1	0
TQ0IOC2			0	0	0	0	TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0

TQ0EES1	TQ0EES0	外部事件计数输入信号(TIQ00 引脚) 有效沿设置
0	0	无边沿检测 (外部事件计数无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

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	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

- 注意事项
1. 当 TQ0CTL0.TQ0CE 位 = 0 时，重新写入 TQ0EES1，TQ0EES0，TQ0ETS1 和 TQ0ETS0 位（当 TQ0CE 位 = 1，可以写入相同的值）。如果重新写入不能正确执行，将 TQ0CE 位清为 0，然后重新进行位的设定。
 2. 只有当 TQ0CTL1.TQ0EEE 位 = 1 或设置为外部事件计数模式（TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 = 001）时，TQ0EES1 和 TQ0EES0 位有效。
 3. 只有在外触发脉冲输出模式（TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 010）或在单次触发脉冲输出模式（TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 011）时，TQ0ETS1 和 TQ0ETS0 位有效。

(6) TMQ0 选项寄存器 0（TQ0OPT0）

TQ0OPT0 寄存器为 8 位寄存器，用来设置捕获/比较操作和检测计数器溢出。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF545H

	7	6	5	4	3	2	1	<0>
TQ0OPT0	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0	0	0	0	TQ0OVF

TQ0CCSm	TQ0CCRm寄存器捕获/比较选择
0	比较寄存器选择
1	捕获寄存器选择
TQ0CCSm位的设置仅在自由运行定时器模式下有效。	

TQ0OVF	TMQ0溢出检测
置 (1)	发生溢出
复位 (0)	TQ0OVF位写入0或TQ0CTL0.TQ0CE位 = 0
<ul style="list-style-type: none"> 自由运行定时器模式下或脉冲宽度测量模式下，当16位计数器的计数值由FFFFH变为0000H发生溢出时，TQ0OVF位置位。 在TQ0OVF位被置为1的同时，产生中断请求信号(INTTQ0OV)。 只有在自由运行定时器模式和脉冲宽度测量模式下，才会产生INTTQ0OV信号。 即使当TQ0OVF位 = 1时读取TQ0OVF位或TQ0OPT0寄存器，TQ0OVF位也不清零。 TQ0OVF位可读取也可写入，但是TQ0OVF位不能使用软件置1，写入1对TMQ0的操作没有影响。 	

注意事项 1. 当 TQ0CTL0.TQ0CE 位 = 0 时，重新写入 TQ0CCS3 至 TQ0CCS0 位（当 TQ0CE 位 = 1 时，可以写入相同的值）。如果重新写入不能正确执行，将 TQ0CE 位清为 0，然后重新进行位的设定。

2. 请确保将第 1 位至第 3 位清为“0”。

备注 m=0 至 3

(7) TMQ0 捕获/比较寄存器 0 (TQ0CCR0)

根据不同工作模式，TQ0CCR0 寄存器可以用作捕获寄存器或比较寄存器。

在自由计时器模式下，根据 TQ0OPT0.TQ0CCS0 位的设定，该寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TQ0CCR0 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

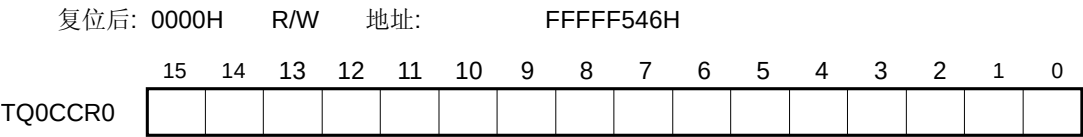
TQ0CCR0 寄存器可以在运行中进行读取和写入。

该寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

注意事项 在以下状态禁止访问 TQ0CCR0 寄存器。详细说明，参见 3.4.8 (2) 访问片上外设输入/输出专用寄存器。

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



(a) 用作比较寄存器时的功能

即便在 TQ0CTL0.TQ0CE 位=1 时，TQ0CCR0 寄存器仍可以重新写入。

TQ0CCR0 寄存器的设定值传送给 CCR0 缓冲寄存器。当 16 位计数器的值与 CCR0 缓冲寄存器的值相匹配时，就会生成一个比较匹配中断请求信号（INTTQ0CC0）。如果此时 TOQ00 引脚输出允许，则 TOQ00 引脚的输出会反相。

在间隔定时器模式、外部事件计数器模式、外部触发脉冲输出模式、单次触发脉冲输出模式或 PWM 输出模式时，当 TQ0CCR0 寄存器作为周期寄存器时，如果 16 位计数器的计数值与 CCR0 缓冲寄存器数值匹配，则计数器清零为（0000H）。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下，当 TQ0CCR0 寄存器用作捕获寄存器时，若检测到捕获触发输入引脚（TIQ00 引脚）的有效边沿，则 16 位计数器的当前计数值存储于 TQ0CCR0 寄存器中。在脉宽测量模式下，若检测到捕获触发输入引脚（TIQ00 引脚）的有效边沿，16 位计数器的当前计数值就储存在 TQ0CCR0 寄存器中，且 16 位计数器清零（0000H）。

即使捕获操作和读取 TQ0CCR0 寄存器的操作发生冲突，还是可以正确读取 TQ0CCR0 寄存器的值。

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

表 8-2. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

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

(8) TMQ0 捕获/比较寄存器 1 (TQ0CCR1)

根据不同工作模式，TQ0CCR1 寄存器可以用作捕获寄存器或比较寄存器。

在自由计时器模式下，根据 TQ0OPT0.TQ0CCS1 位的设定，该寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TQ0CCR1 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

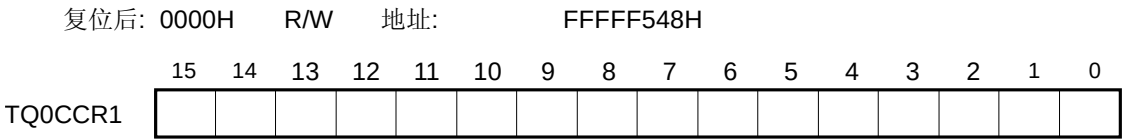
TQ0CCR1 寄存器可以在运行中进行读取和写入。

该寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

注意事项 在以下状态禁止访问 TQ0CCR1 寄存器。详细说明，参见 3.4.8 (2) 访问片上外设输入/输出专用寄存器。

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



(a) 用作比较寄存器时的功能

即便在 TQ0CTL0.TQ0CE 位= 1 时，TQ0CCR1 寄存器仍可以重新写入。

TQ0CCR1 寄存器的设定值传送给 CCR1 缓冲寄存器。当 16 位计数器的值与 CCR1 缓冲寄存器的值相匹配时，就会生成一个比较匹配中断请求信号（INTTQ0CC1）。如果此时 TOQ01 引脚输出允许，则 TOQ01 引脚的输出会反相。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下，当 TQ0CCR1 寄存器用作捕获寄存器时，若检测到捕获触发输入引脚（TIQ01 引脚）的有效边沿，则 16 位计数器的当前计数值存储于 TQ0CCR1 寄存器中。在脉宽测量模式下，若检测到捕获触发输入引脚（TIQ01 引脚）的有效边沿，16 位计数器的当前计数值就储存在 TQ0CCR1 寄存器中，且 16 位计数器清零（0000H）。

即使捕获操作和读取 TQ0CCR1 寄存器的操作发生冲突，还是可以正确读取 TQ0CCR1 寄存器的值。

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

表 8-3. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

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

(9) TMQ0 捕获/比较寄存器 2 (TQ0CCR2)

根据不同工作模式，TQ0CCR2 寄存器可以用作捕获寄存器或比较寄存器。

在自由计时器模式下，根据 TQ0OPT0.TQ0CCS2 位的设定，该寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TQ0CCR2 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

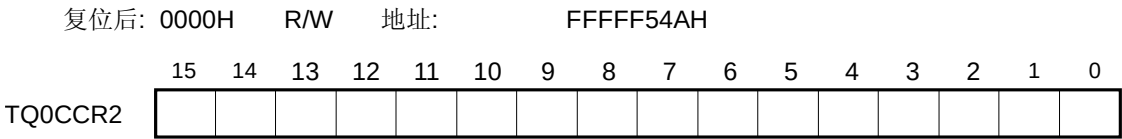
TQ0CCR2 寄存器可以在运行中进行读取和写入。

该寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

注意事项 在以下状态禁止访问 TQ0CCR2 寄存器。详细说明，参见 3.4.8 (2) 访问片上外设输入/输出专用寄存器。

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



(a) 用作比较寄存器时的功能

即便在 TQ0CTL0.TQ0CE 位= 1 时，TQ0CCR2 寄存器仍可以重新写入。

TQ0CCR2 寄存器的设定值传送给 CCR2 缓冲寄存器。当 16 位计数器的计数值和 CCR2 缓冲寄存器的值匹配时，也会产生一个比较匹配中断请求信号 (INTTQ0CC2)。如果此时 TOQ02 引脚输出允许，则 TOQ02 引脚的输出会反相。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下，当 TQ0CCR2 寄存器用作捕获寄存器时，若检测到捕获触发输入引脚 (TIQ02 引脚) 的有效边沿，则 16 位计数器的当前计数值存储于 TQ0CCR2 寄存器中。在脉宽测量模式下，若检测到捕获触发输入引脚 (TIQ02 引脚) 的有效边沿，16 位计数器的当前计数值就储存在 TQ0CCR2 寄存器中，且 16 位计数器清零 (0000H)。

即使捕获操作和读取 TQ0CCR2 寄存器的操作发生冲突，还是可以正确读取 TQ0CCR2 寄存器的值。

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

表 8-4. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

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

(10) TMQ0 捕获/比较寄存器 3 (TQ0CCR3)

根据不同工作模式，TQ0CCR3 寄存器可以用作捕获寄存器或比较寄存器。

在自由计时器模式下，根据 TQ0OPT0.TQ0CCS3 位的设定，该寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TQ0CCR3 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

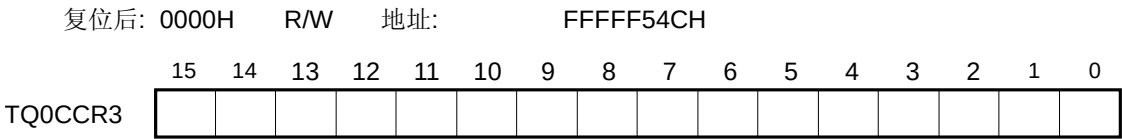
TQ0CCR3 寄存器可以在运行中进行读取和写入。

该寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

注意事项 在以下状态禁止访问 TQ0CCR3 寄存器。详细说明，参见 3.4.8 (2) 访问片上外设输入/输出专用寄存器。

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



(a) 用作比较寄存器时的功能

即便在 TQ0CTL0.TQ0CE 位= 1 时，TQ0CCR3 寄存器仍可以重新写入。

TQ0CCR3 寄存器的设定值传送给 CCR3 缓冲寄存器。当 16 位计数器的计数值与 CCR3 缓冲寄存器的值相匹配时，就会生成一个比较匹配中断请求信号（INTTQ0CC3）。如果此时 TOQ03 引脚输出允许，则 TOQ03 引脚的输出会反相。

(b) 用作捕获寄存器时的功能

在自由运行定时器模式下，当 TQ0CCR3 寄存器用作捕获寄存器时，若检测到捕获触发输入引脚（TIQ03 引脚）的有效边沿，则 16 位计数器的当前计数值存储于 TQ0CCR3 寄存器中。在脉宽测量模式下，若检测到捕获触发输入引脚（TIQ03 引脚）的有效边沿，16 位计数器的当前计数值就储存在 TQ0CCR3 寄存器中，且 16 位计数器清零（0000H）。

即使捕获操作和读取 TQ0CCR3 寄存器的操作发生冲突，还是可以正确读取 TQ0CCR3 寄存器的值。

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

表 8-5. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

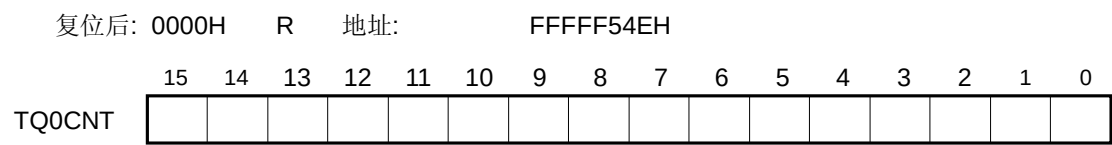
操作模式	捕获/比较寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	任何时候均可写入
外部事件计数器	比较寄存器	任何时候均可写入
外部触发脉冲输出	比较寄存器	批量写入
单次触发脉冲输出	比较寄存器	任何时候均可写入
PWM 输出	比较寄存器	批量写入
自由运行定时器	捕获/比较寄存器	任何时候均可写入
脉冲宽度测量	捕获寄存器	—

(11) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

TQ0CNT 寄存器是读出缓冲寄存器，可以读取 16 位计数器的计数值。
如果在 TQ0CTL0.TQ0CE 位=1 时读取该寄存器，可以读出 16 位定时器的计数值。
该寄存器为只读寄存器，按 16 位宽度读取。
当 TQ0CE 位=0 时，TQ0CNT 寄存器的值被清为 0000H。如果此时读取 TQ0CNT 寄存器，则不会读出 16 位计数器的值 (FFFFH)，读到的值为 0000H。
在复位后 TQ0CNT 寄存器的值被清为 0000H，因为 TPnCE 位被清为 0。

注意事项 在以下状态禁止访问 TQ0CNT 寄存器。详细说明，参见 3.4.8 (2) 访问片上外设输入/输出专用寄存器。

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



8.5 操作

定时器 TMQ0 可以执行如下操作。

操作	TQ0CTL1.TQ0EST 位 (软件触发位)	TIQ00 引脚 (外部触发输入)	捕获/比较寄存器设置	比较寄存器写入
间隔计时器模式	无效	无效	仅作比较寄存器	任何时候均可写入
外部事件计数器模式 ^{注 1}	无效	无效	仅作比较寄存器	任何时候均可写入
外部触发脉冲输出模式 ^{注 2}	有效	有效	仅作比较寄存器	批量写入
单次触发脉冲输出模式 ^{注 2}	有效	有效	仅作比较寄存器	任何时候均可写入
PWM 输出模式	无效	无效	仅作比较寄存器	批量写入
自由定时器模式	无效	无效	可以切换	任何时候均可写入
脉冲宽度测量模式 ^{注 2}	无效	无效	仅作捕获寄存器	无效

- 注**
1. 在用作外部事件计数器模式下，应该指定对 TIQ00 引脚捕获触发输入的有效边沿不作检测（将 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位清为“00”）。
 2. 在外部触发脉冲输出模式、单次触发脉冲输出模式和脉冲宽度测量模式下，选择内部时钟作为计数时钟。（将 TQ0CTL1.TQ0EEE 位清为 0）。

8.5.1 间隔定时器模式 (TQ0MD2 至 TQ0MD0 位=000)

在间隔定时器模式下，如果 TQ0CTL0.TQ0CE 位被置为 1，则会以指定时间间隔生成中断请求信号 (INTTQ0CC0)，从 TOQ00 引脚输出方波，指定的时间间隔等于方波的半周期。
通常在间隔定时器模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。

图 8-2. 间隔定时器的配置

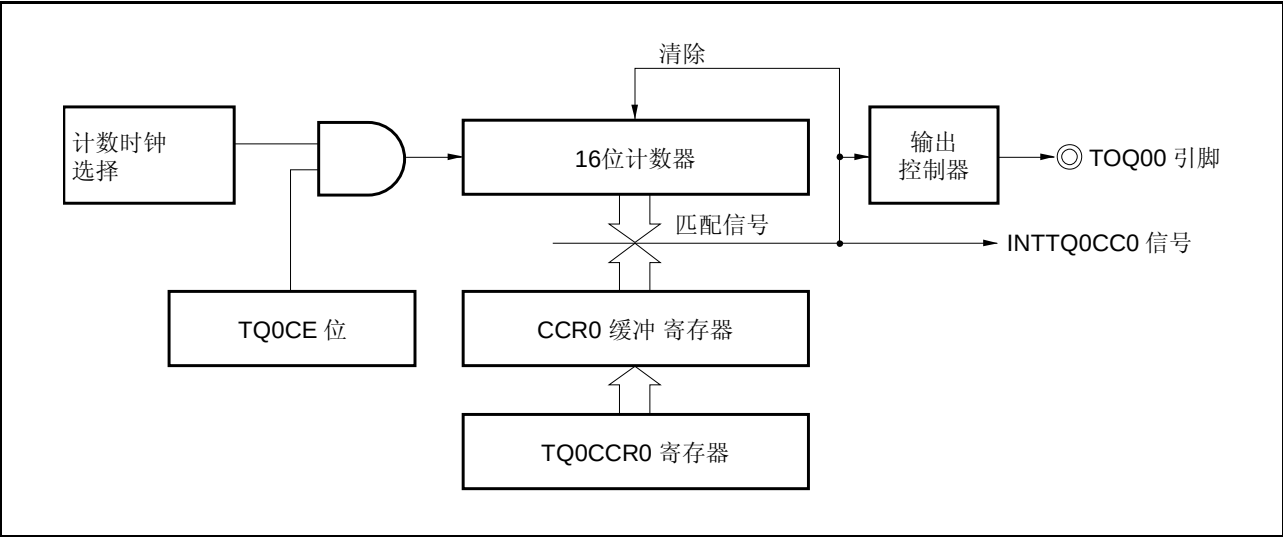
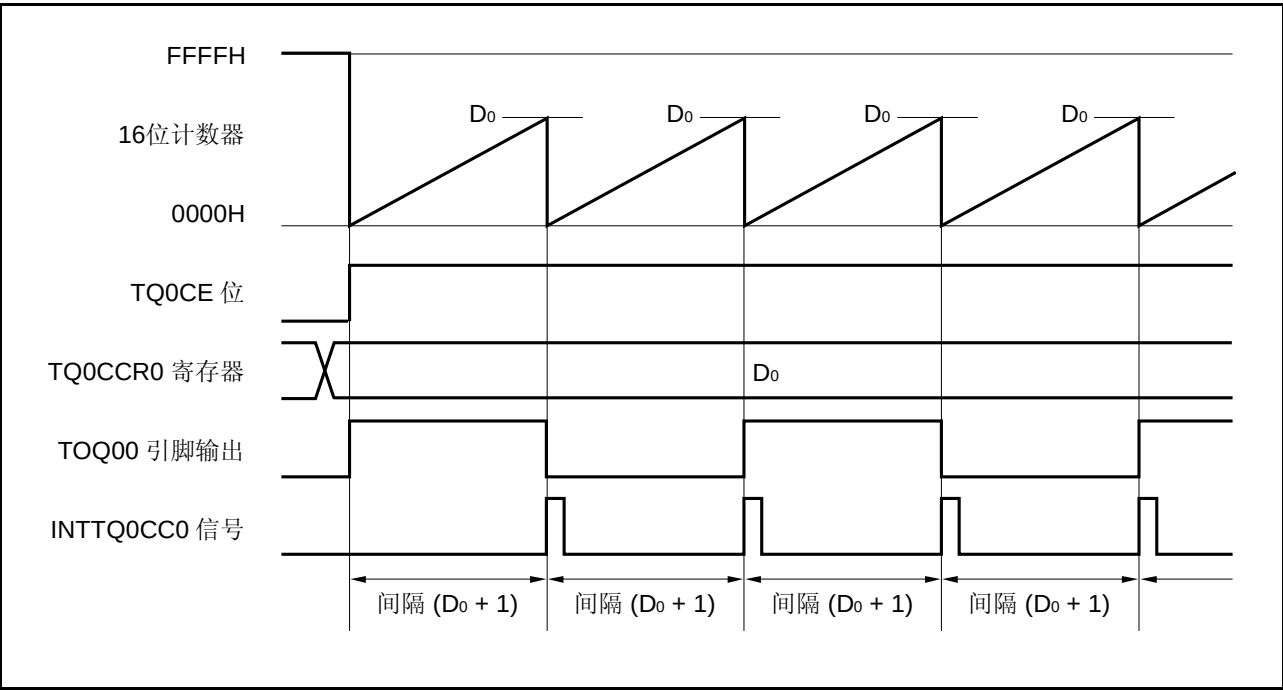


图 8-3. 间隔定时器模式的基本时序



当 TQ0CE 位被置为 1 时, 16 位计数器的值由计数时钟同步从 FFFFH 清零为 0000H, 计数器开始计数。此时, TOQ00 引脚的输出反相。此外, TQ0CCR0 寄存器的设定值传送给 CCR0 缓冲寄存器。

当 16 位计数器的计数值和 CCR0 缓冲寄存器的值相匹配时, 则 16 位计数器清为 0000H, TOQ00 引脚输出反相, 并生成一个比较匹配中断请求信号 (INTTQ0CC0)。

时间间隔可用下述表达式计算。

$$\text{时间间隔} = (\text{TQ0CCR0 寄存器的设定值} + 1) \times \text{计数时钟周期}$$

图 8-4. 间隔定时器模式操作时的寄存器设置 (1/2)

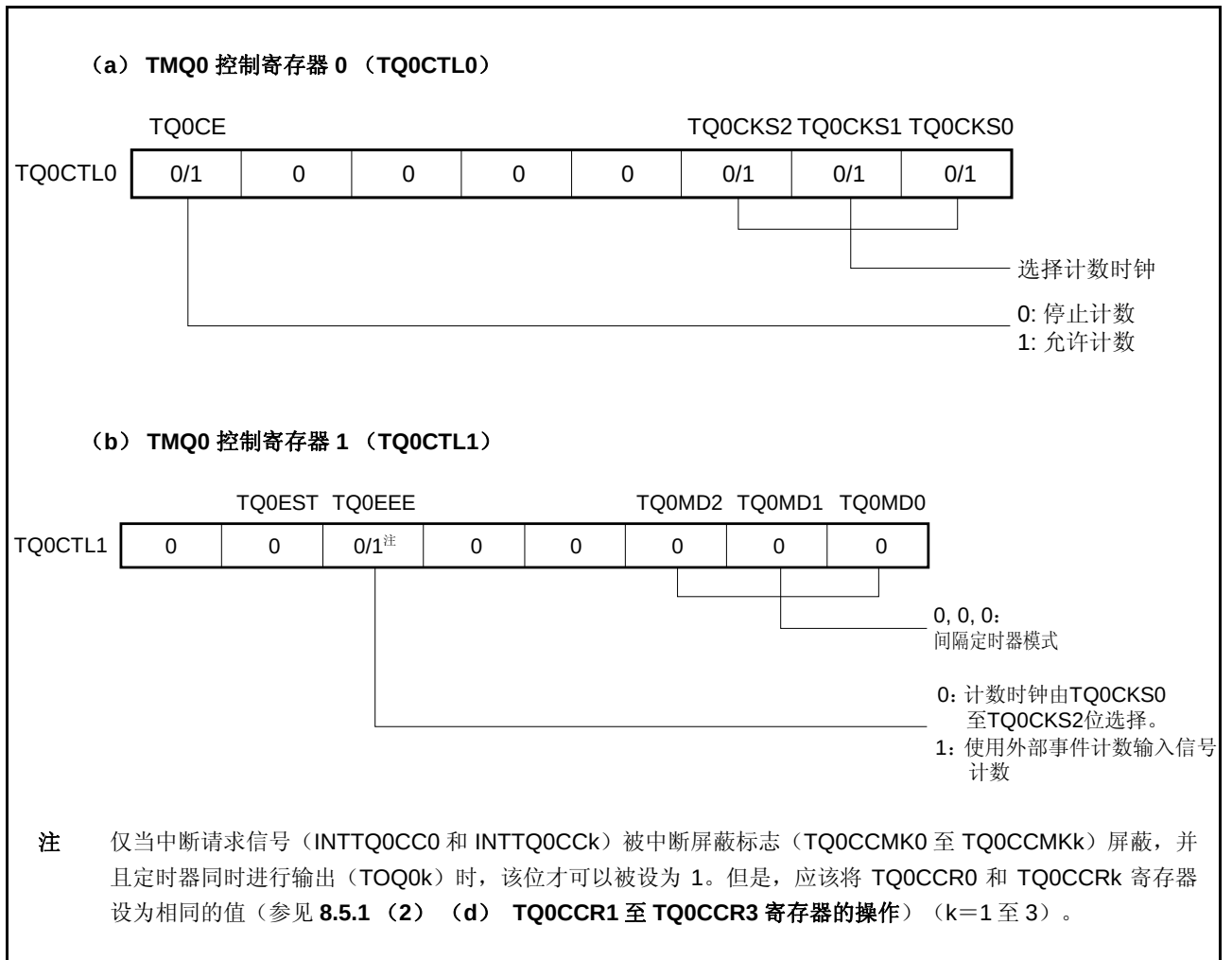
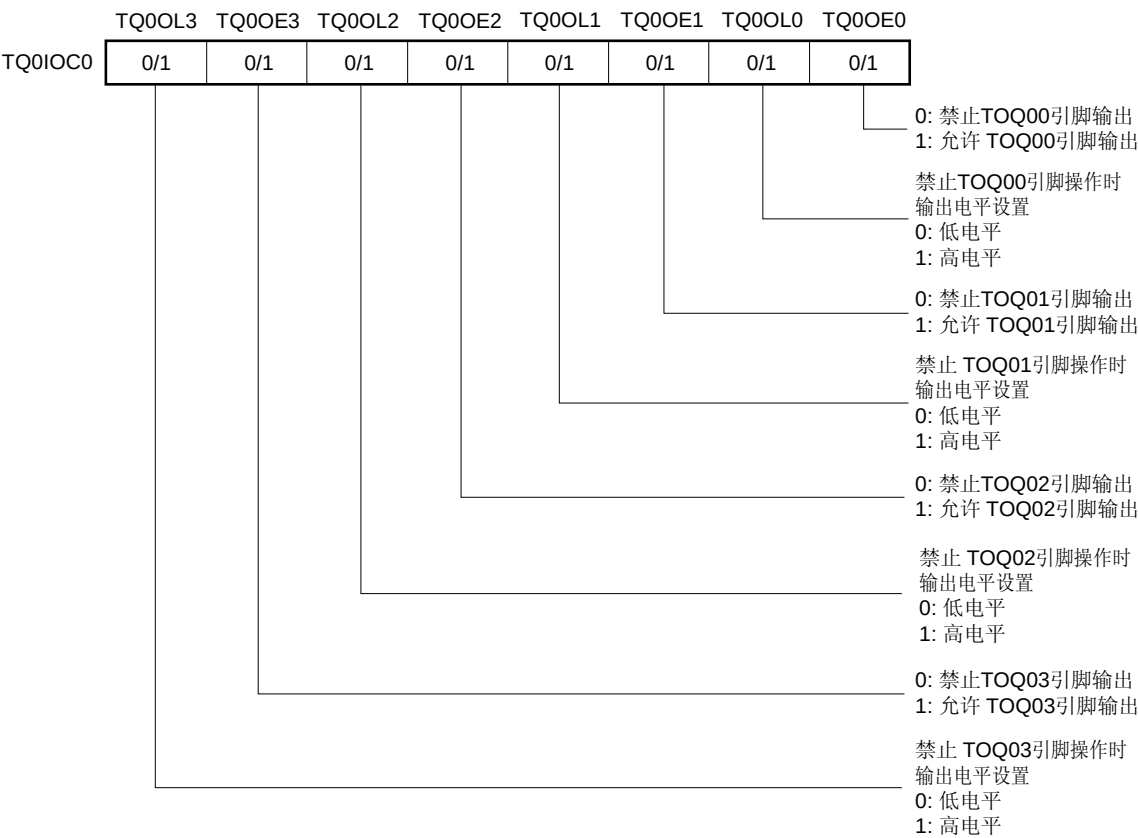


图 8-4. 间隔定时器模式操作时的寄存器设置 (2/2)

(c) TMQ0 输入/输出控制寄存器 0 (TQ0IOC0)



(d) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

通过读取 TQ0CNT 寄存器，可以读取 16 计数器的当前值。

(e) TMQ0 捕获/比较寄存器 0 (TQ0CCR0)

如果 TQ0CCR0 寄存器的值被设置为 D₀，则时间间隔的计算如下：

时间间隔 = (D₀ + 1) × 计数时钟周期

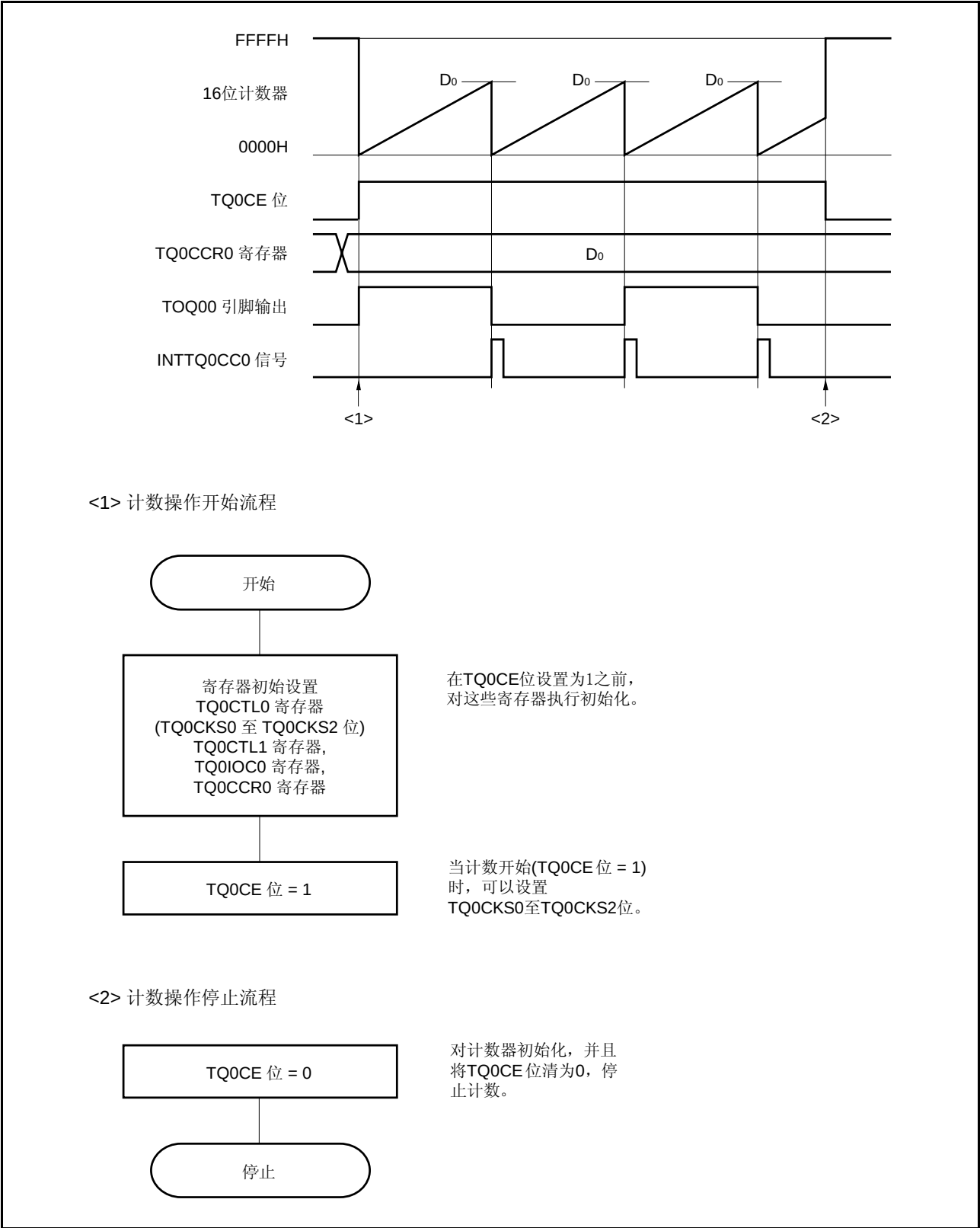
(f) TMQ0 捕获/比较寄存器 1 至 3 (TQ0CCR1 至 TQ0CCR3)

通常在间隔定时器模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。但是，TQ0CCR1 至 TQ0CCR3 寄存器的设定值仍然会被传送给 CCR1 至 CCR3 缓冲寄存器。当 16 位计数器的计数值与 CCR1 至 CCR3 缓冲寄存器的值匹配时，会产生比较匹配中断请求信号 (INTTQ0CC1 至 INTTQ0CCR3)。
所以，采用相应的中断屏蔽标志 (TQ0CCMK1 至 TQ0CCMK3) 屏蔽中断请求。

备注 在间隔定时器模式下，不使用 TMQ0 输入/输出控制寄存器 1 (TQ0IOC1)、TMQ0 输入/输出控制寄存器 2 (TQ0IOC2) 和 TMQ0 选项寄存器 0 (TQ0OPT0)。

(1) 间隔定时器模式操作流程

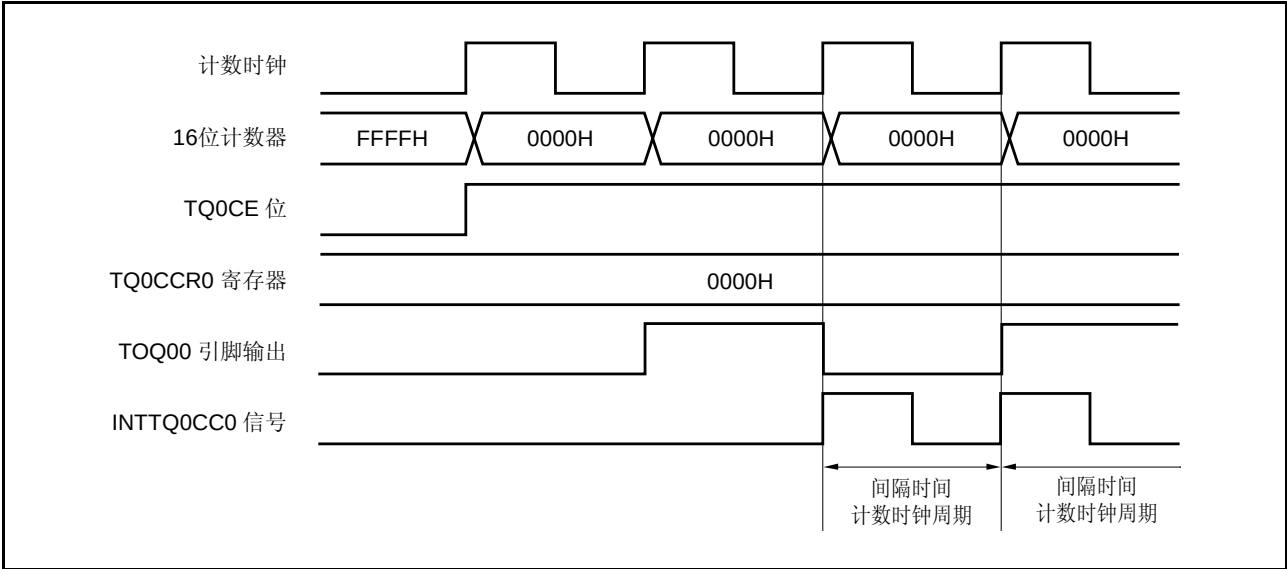
图 8-5. 间隔定时器模式下的软件处理流程



(2) 间隔定时器模式操作时序

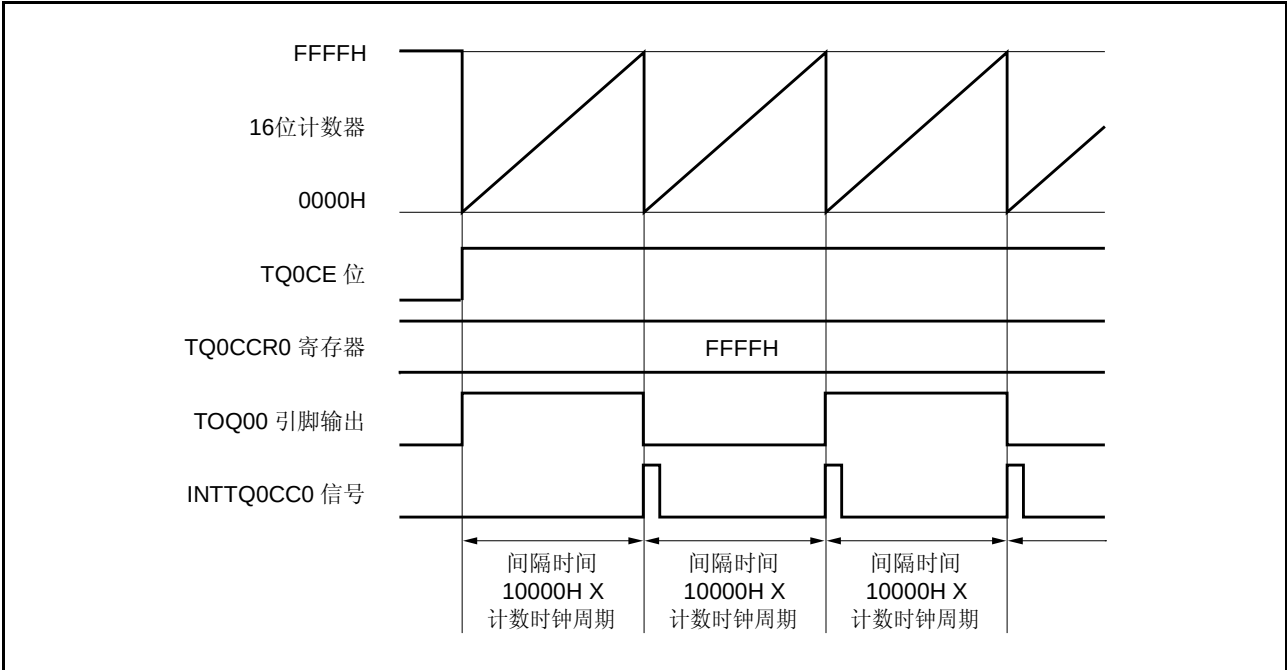
(a) TQ0CCR0 寄存器清为 0000H 时的操作

如果 TQ0CCR0 寄存器被清为 0000H，第一个时钟之后的每个时钟计数时都会产生 INTTQ0CC0 信号，而且，TOQ00 引脚的输出反相。
16 位计数器的值总是 0000H。



(b) TQ0CCR0 寄存器被设置为 FFFFH 时的操作

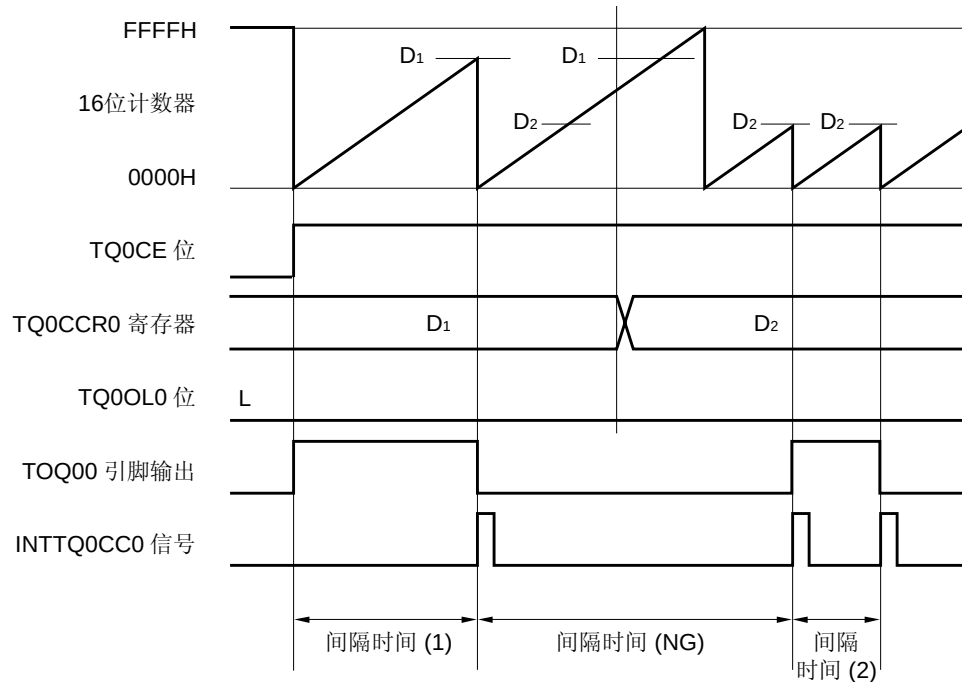
如果 TQ0CCR0 寄存器被置为 FFFFH，那么 16 位计数器可以计数到 FFFFH。该计数器在下一个计数时刻被同步清零。产生 INTTQ0CC0 信号，并且 TOQ00 引脚输出反相。此时，不产生溢出中断请求信号 (INTTQ0OV)，而且，溢出标志 (TQ0OPT0.TQ0OVF 位) 也不会置 1。



(c) 重新写入 TQ0CCR0 寄存器的注意事项

要将 TQ0CCR0 寄存器的值变为比当前值小的值，应停止计数，然后改变其设定值。

如果在计数器工作期间，将寄存器 TQ0CCR0 的值重写为比当前值小的值，16 位计数器可能会溢出。



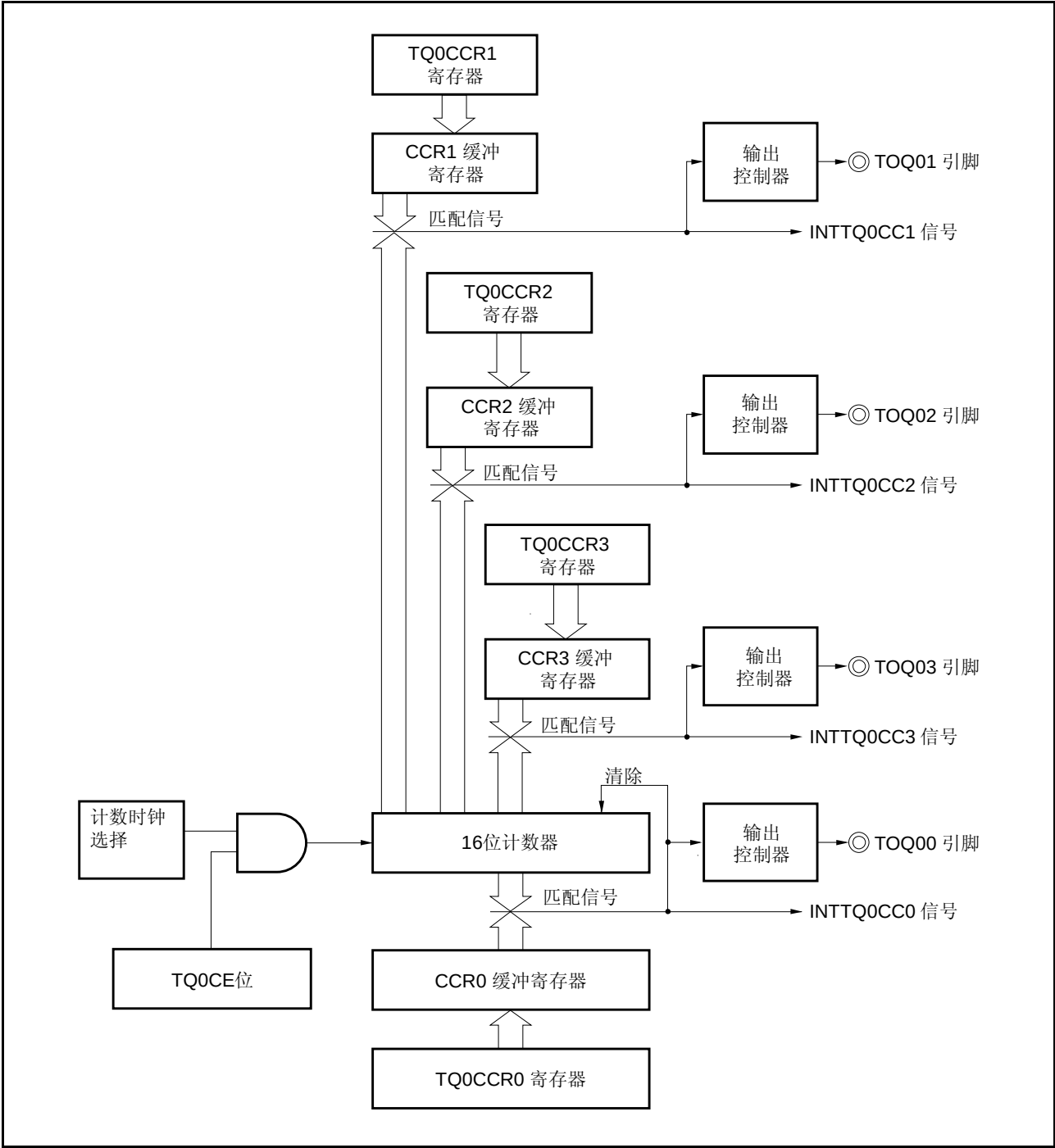
备注

- 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): $(D_2 + 1) \times \text{计数时钟周期}$

当 16 位计数器计数值大于 D_2 而小于 D_1 时，如果将 TQ0CCR0 寄存器的值从 D_1 变为 D_2 ，则在重写 TQ0CCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。结果，用来比较的 16 位计数器的值是 D_2 。但是，由于计数值已经超过了 D_2 ，所以 16 位计数器就会一直向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值和 D_2 相符时才产生 INTTQ0CC0 信号，且 TOQ00 引脚的输出反相。因此，可能不会在原先预计的间隔时间“ $(D_1 + 1) \times \text{计数时钟周期}$ ”或“ $(D_2 + 1) \times \text{计数时钟周期}$ ”时产生 INTTQ0CC0 信号，而会以“ $(10000H + D_2 + 1) \times \text{计数时钟周期}$ ”的间隔产生中断请求信号。

(d) TQ0CCR1 至 TQ0CCR3 寄存器的操作

图 8-6. TQ0CCR1 至 TQ0CCR3 寄存器的配置

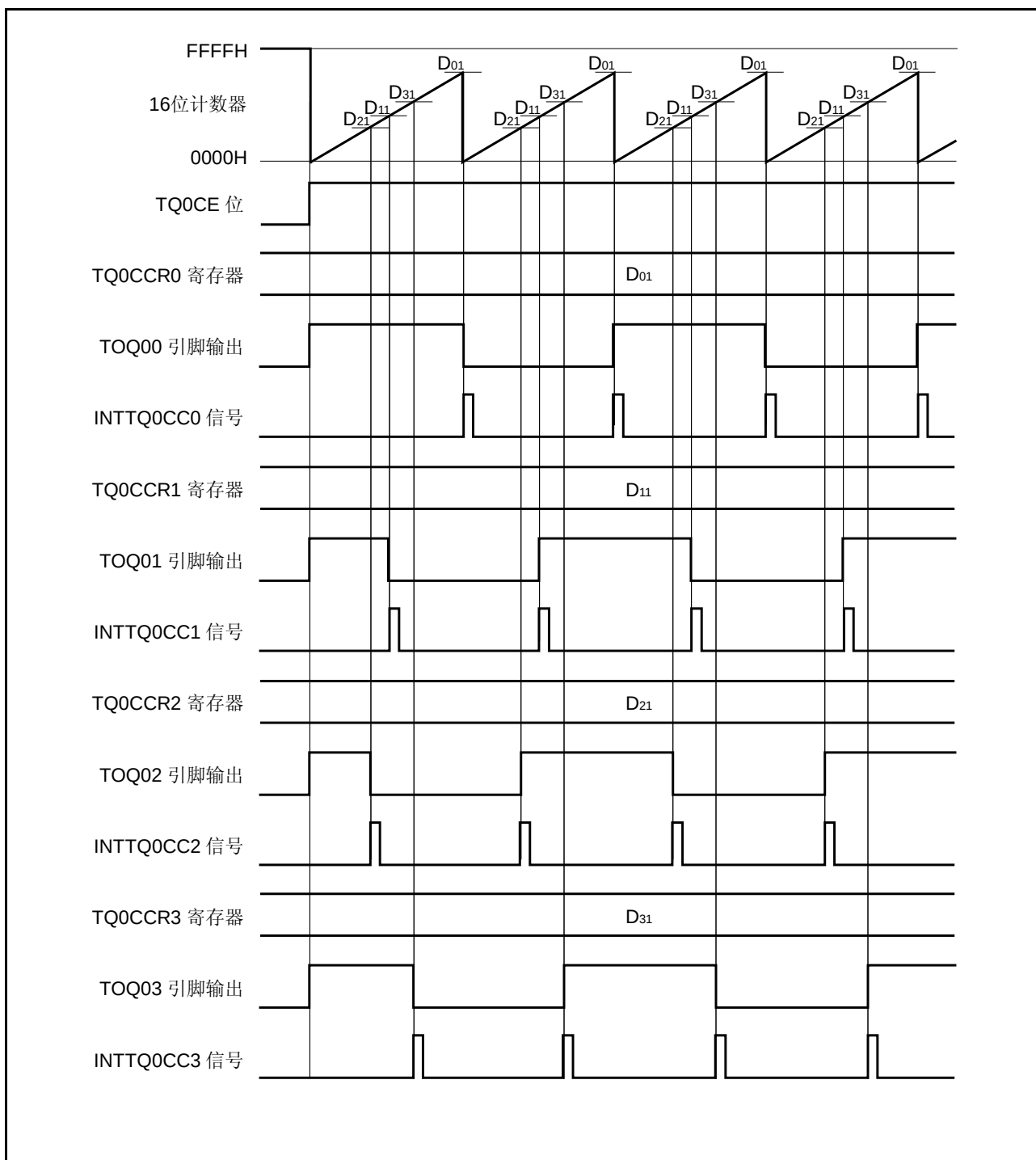


如果 $TQ0CCRk$ 寄存器的设定值小于 $TQ0CCR0$ 寄存器的设定值，则每个计数周期产生一次 $INTTQ0CCk$ 中断请求信号。同时， $TOPQ0k$ 引脚的输出反相。

$TOQ0k$ 引脚输出一个方波，其周期与 $TOQ00$ 引脚的输出周期相同。

备注 $k=1$ 至 3

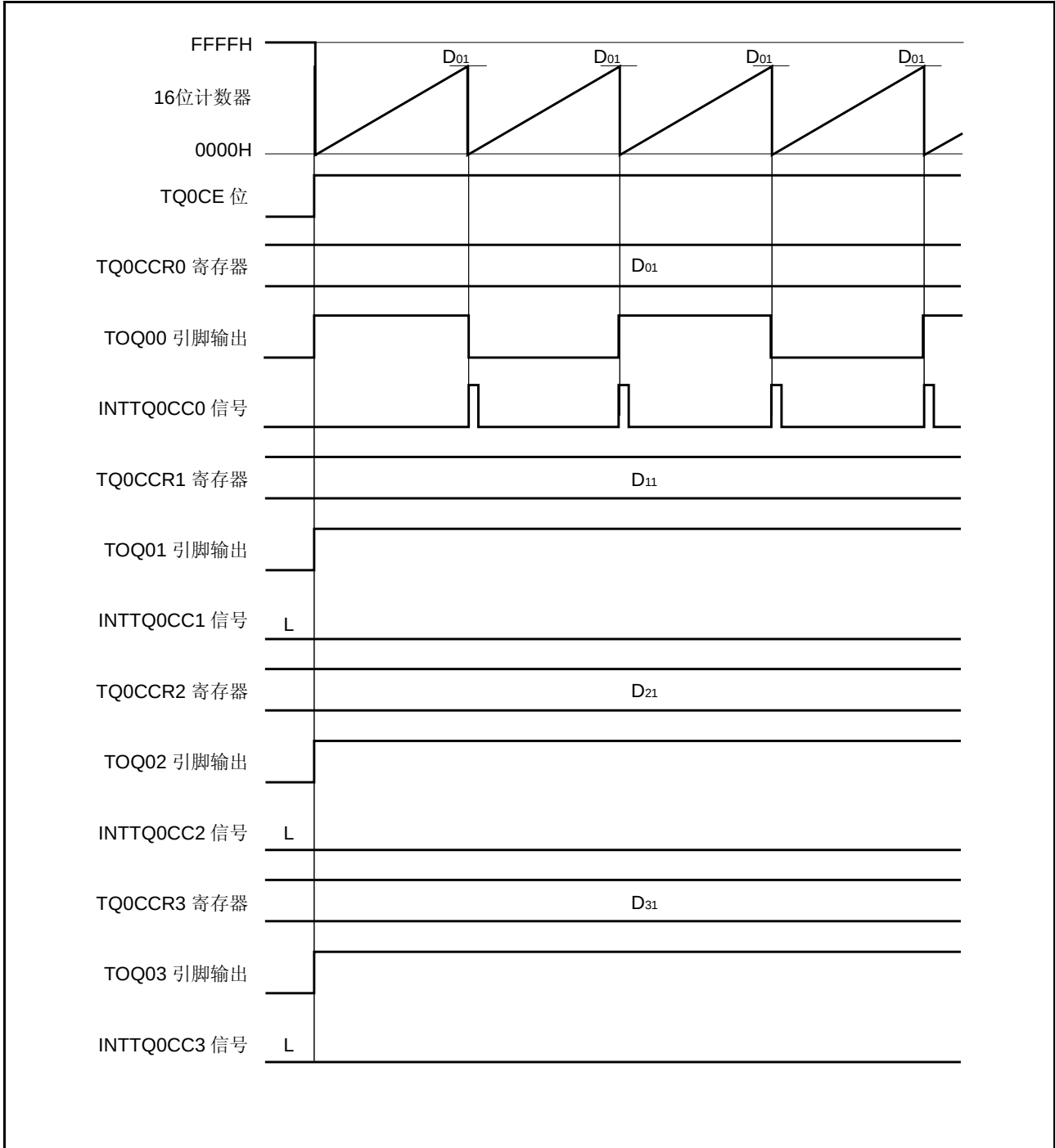
图 8-7. 当 $D_{01} \geq D_{k1}$ 时的时序图



如果 TQ0CCRk 寄存器的设定值大于 TQ0CCR0 寄存器的设定值，则 16 位计数器的计数值不会和 TQ0CCRk 寄存器的值发生匹配。因此也就不会产生 INTTQ0CCk 信号，TOQ0k 引脚的输出也不会改变。

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)

在外部事件计数器模式下，当 TQ0CTL0.TQ0CE 位被置为 1 时，对外部事件计数输入信号的有效边沿进行计数，每次有效边沿的计数数量达到指定数目时，产生一个中断请求信号 (INTTQ0CC0)。不能使用 TOQ00 引脚。

通常在外部事件计数模式下，不使用 TQ0CCR1 至 TQ0CCR3 寄存器。

图 8-9. 外部事件计数模式的配置

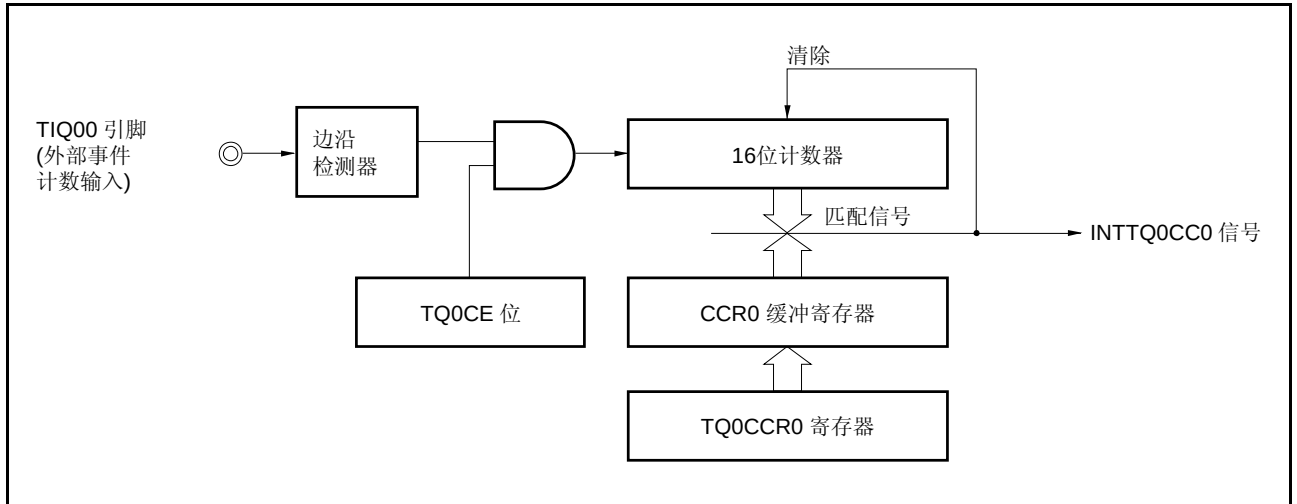
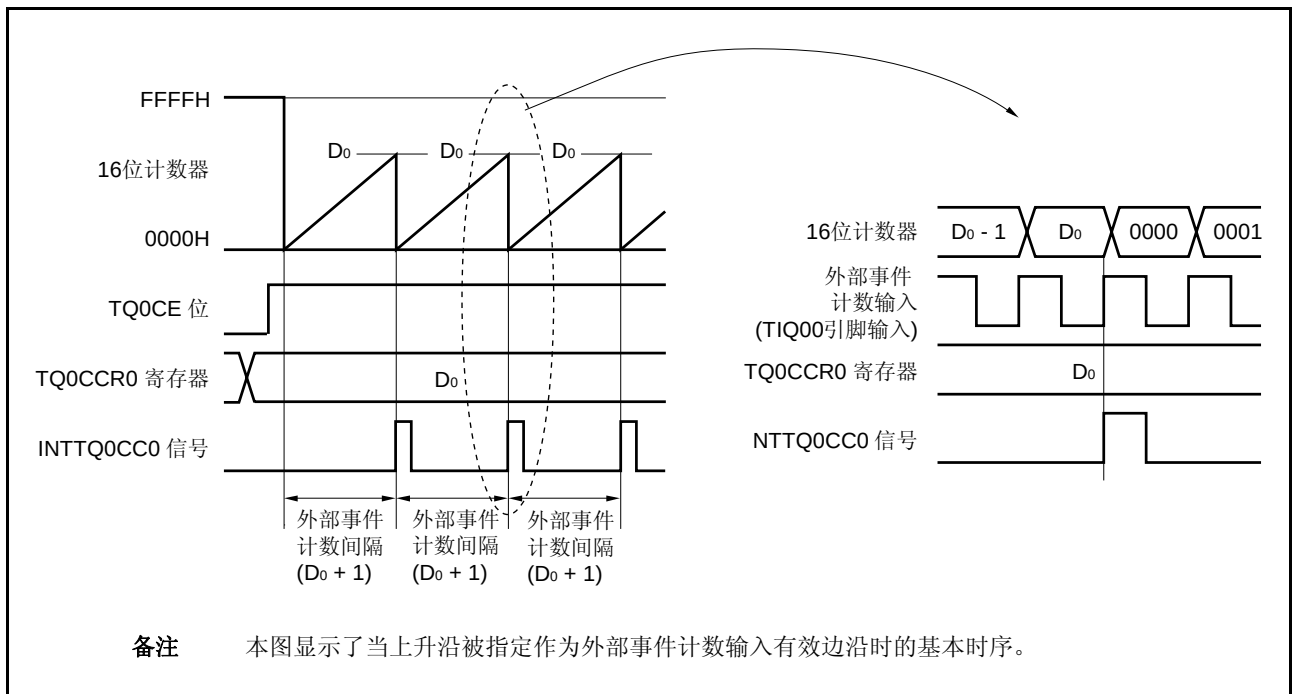


图 8-10. 外部事件计数模式的基本时序



当 TQ0CE 位被置为 1 时, 16 位计数器的值从 FFFFH 被清为 0000H。计数器对检测到的每一次外部事件计数输入的有效边沿进行计数。此外, TQ0CCR0 寄存器的设定值被传送给 CCR0 缓冲寄存器。

当 16 位计数器的计数值和 CCR0 缓冲寄存器的数值相匹配时, 16 位计数器清零为 0000H, 同时生成一个比较匹配中断请求信号 (INTTQ0CC0)。

每当外部事件计数输入的有效边沿数量检测达到 (TQ0CCR0 寄存器设定值+1) 时, 就会生成 INTTQ0CC0 (比较匹配中断请求) 信号。

图 8-11. 外部事件计数模式下的寄存器操作设置 (1/2)

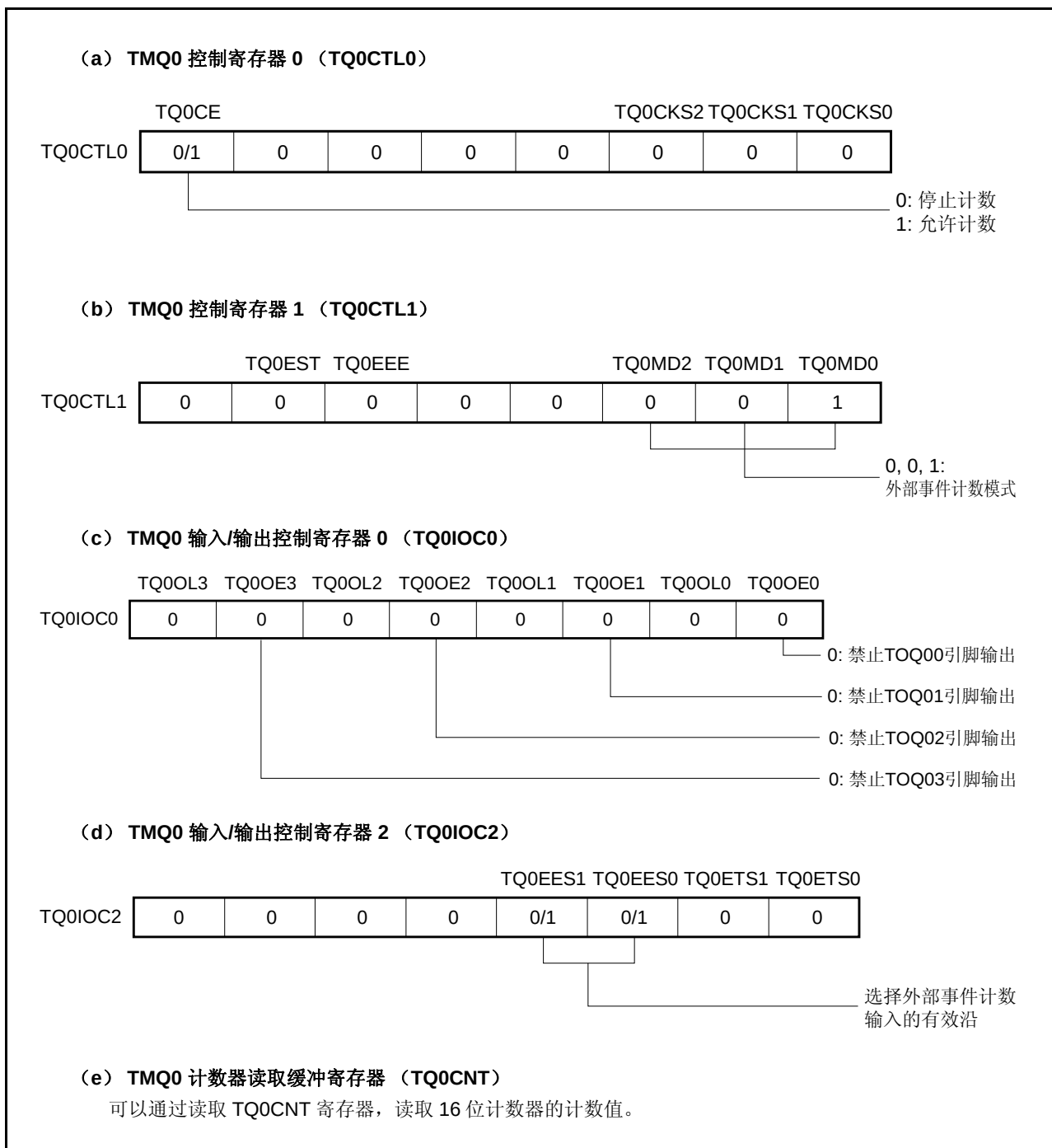


图 8-11. 外部事件计数模式下的寄存器操作设置 (2/2)

(f) TMQ0 捕获/比较寄存器 0 (TQ0CCR0)

如果 TQ0CCR0 寄存器的值被设定为 D_0 ，当外部事件计数值达到 (D_0+1) 时，计数器被清 0，同时产生一个比较匹配中断请求信号 (INTTQ0CC0)。

(g) TMQ0 捕获/比较寄存器 1 至 3 (TQ0CCR1 至 TQ0CCR3)

通常在外部事件计数模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。但是，TQ0CCR1 至 TQ0CCR3 寄存器的设定值仍然被传送给 CCR1 至 CCR3 缓冲寄存器。当 16 位计数器的计数值和 CCR1 至 CCR3 缓冲寄存器的值匹配时，会产生一个比较匹配中断请求信号 (INTTQ0CC1 至 INTTQ0CC3)。

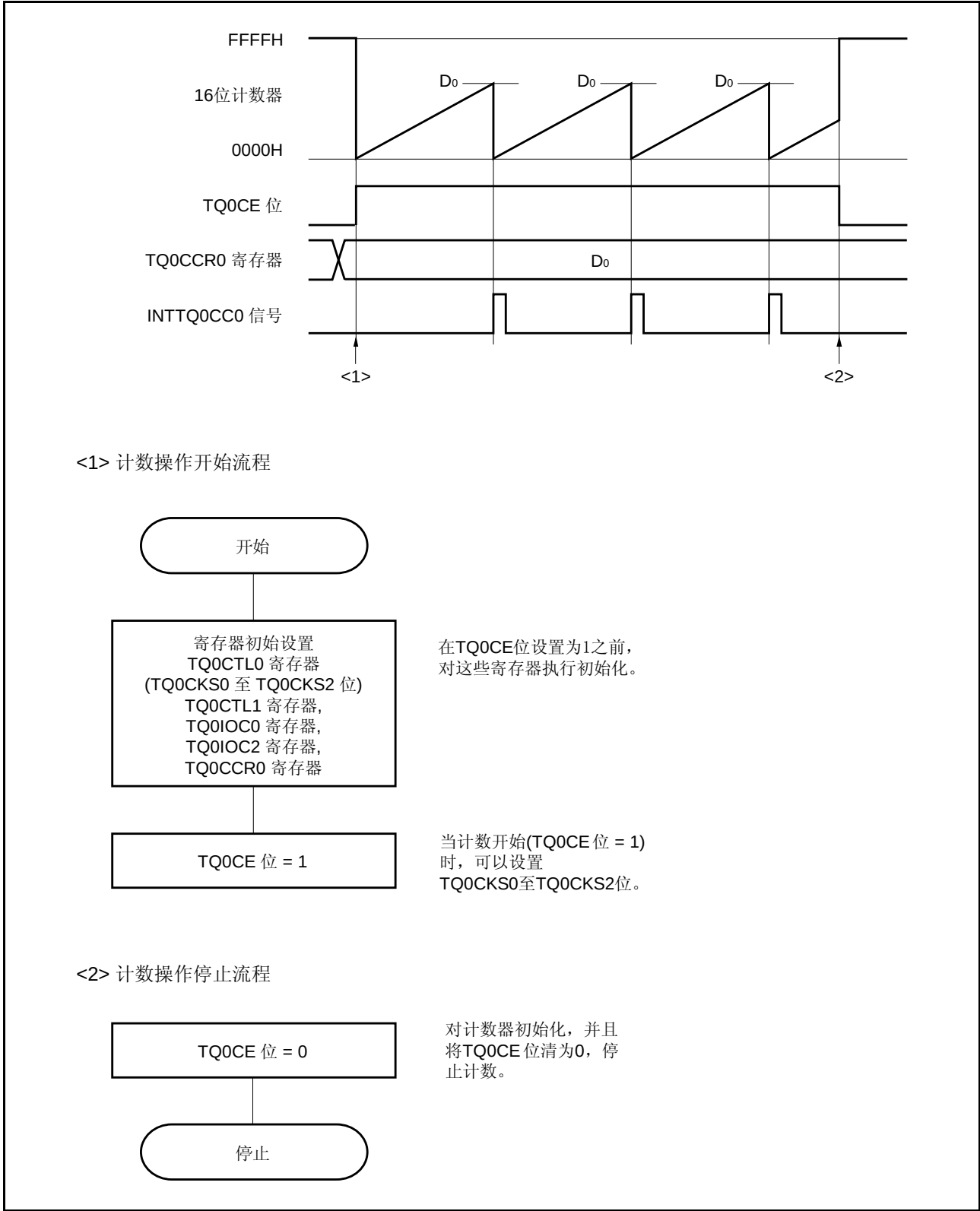
所以，采用中断屏蔽标志 (TQ0CCMK1 至 TQ0CCMK3) 来屏蔽中断信号。

注意事项 当外部时钟用作计数时钟时，外部时钟只能从 TIQ00 引脚输入。此时，将 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位清为 00 (捕获触发输入 (TIQ00 引脚)：无边沿检测)。

备注 在外部事件计数模式下，不使用 TMQ0 输入/输出控制寄存器 1 (TQ0IOC1) 和 TMQ0 选项寄存器 0 (TQ0OPT0)。

(1) 外部事件计数模式的操作流程

图 8-12. 外部事件计数模式下的软件处理流程

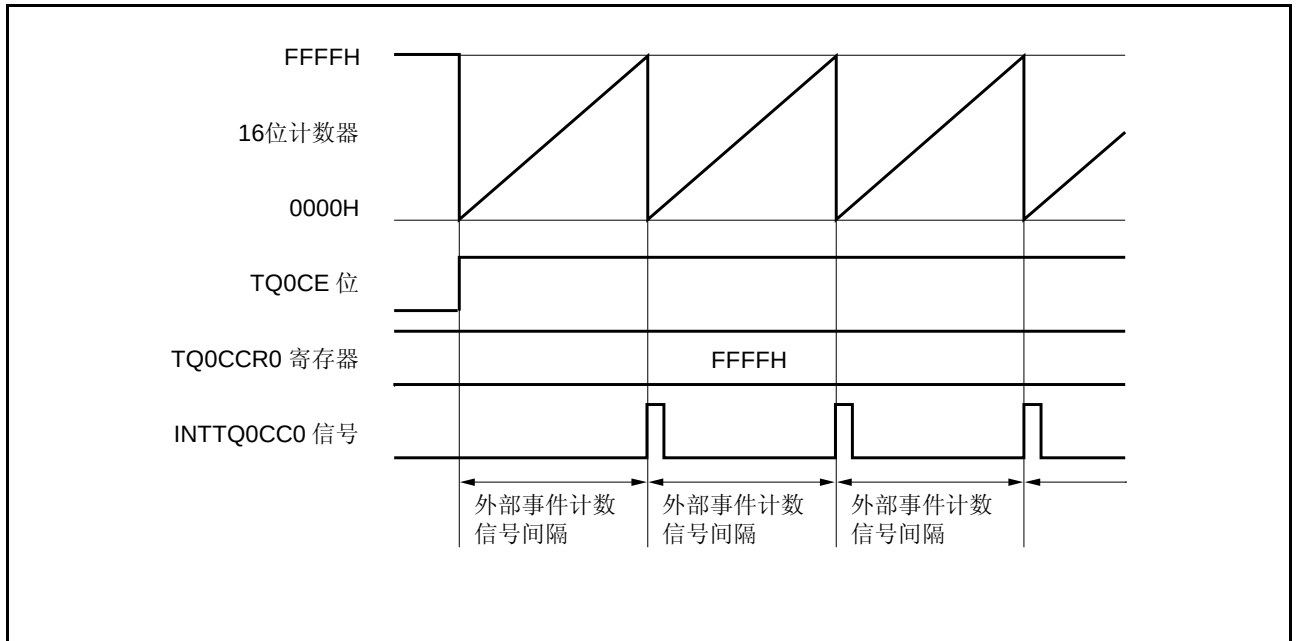


(2) 外部事件计数模式的操作时序

- 注意事项 1. 外部事件计数模式下，不要将 TQ0CCR0 寄存器设置为 0000H。
2. 外部事件计数模式下，禁止使用定时器输出。如果使用外部事件计数输入来进行定时器输出，应设置为间隔定时器模式，并且选择外部事件计数输入允许的操作作为计数时钟（TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位= 000，TQ0CTL1.TQ0EEE 位= 1）。

(a) 如果 TQ0CCR0 寄存器被设置为 FFFFH 时的操作

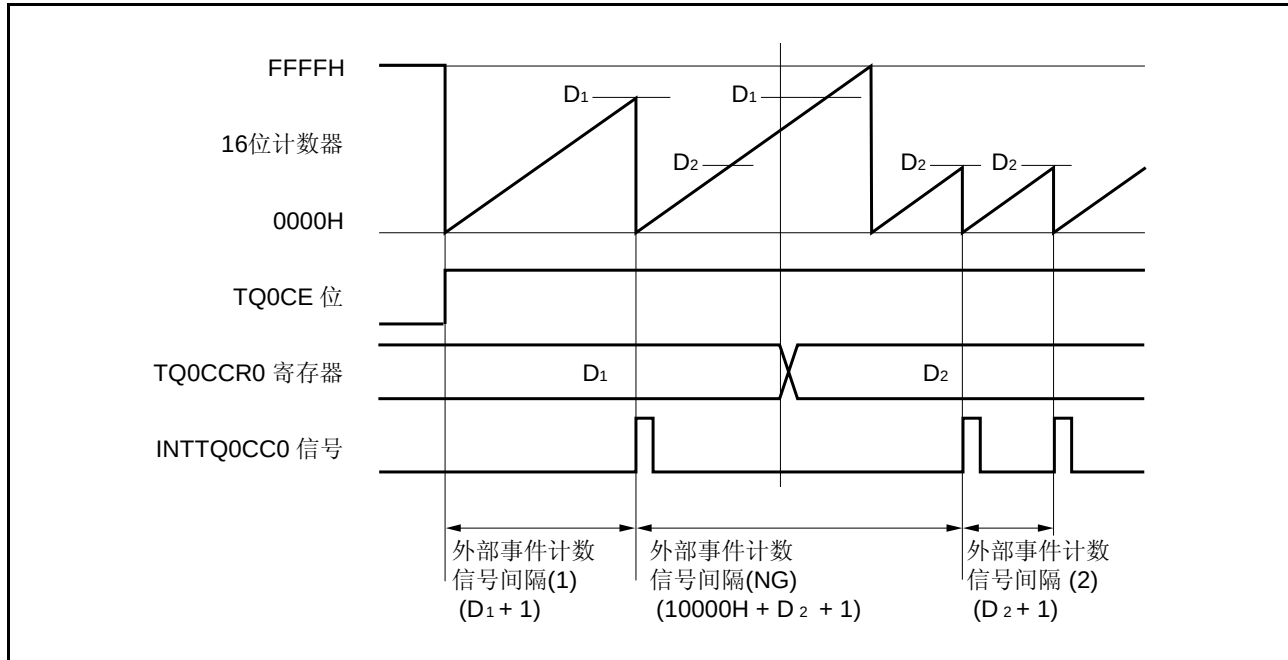
如果 TQ0CCR0 寄存器被设置为 FFFFH，16 位计数器每次检测到的外部事件输入信号有效边沿时，都计数到 FFFFH。当下一个计数到来时，16 位计数器被同步清为 0000H，并且产生 INTTQ0CC0 信号。此时，不设置 TQ0OPT0.TQ0OVF 位。



(b) 重写 TQ0CCR0 寄存器的注意事项

要将 TQ0CCR0 寄存器的值改为比当前值较小的值，应停止计数，然后改变其设定值。

如果在计数器进行计数期间，对寄存器 TQ0CCR0 的值进行重写，则 16 位计数器可能会溢出。



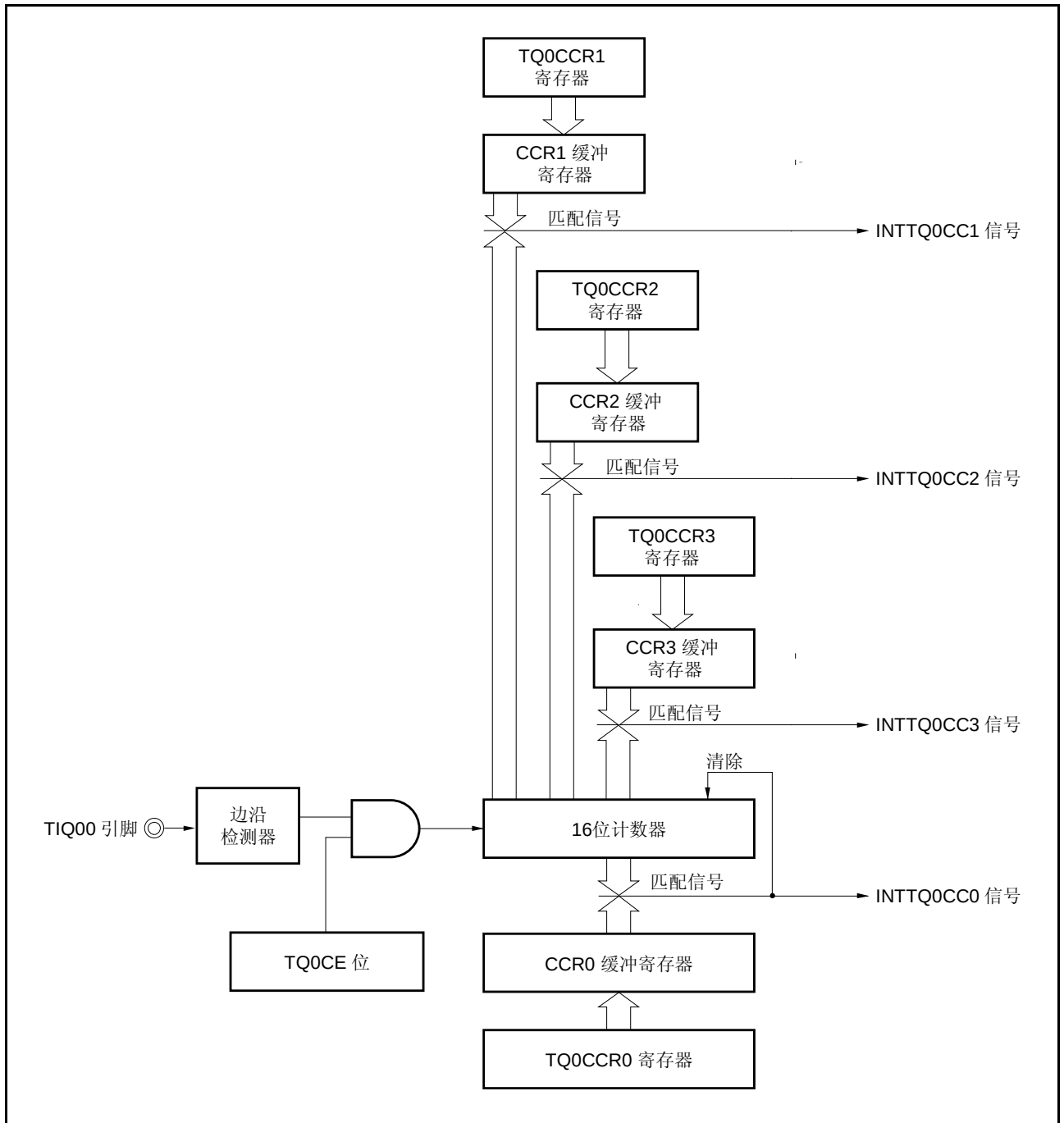
当 16 位计数器计数值大于而小于 D 时，如果将 TQ0CCR0 寄存器的值从 D_1 变为 D_2 ，则在重写 TQ0CCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。因而，与 16 位计数器相比较的值会是 D_2 。

但是，由于计数值已经超过了 D_2 ，所以 16 位计数器就会向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值与 D_2 匹配时，会生成 INTTQ0CC0 信号。

因此，可能不会在原先预计的“ $(D_1 + 1)$ 次”或“ $(D_2 + 1)$ 次”有效边沿生成 INTTQ0CC0 中断请求信号，而可能在“ $(10000H + D_2 + 1)$ 次”有效边沿生成中断请求信号。

(c) TQ0CCR1 至 TQ0CCR3 寄存器的操作

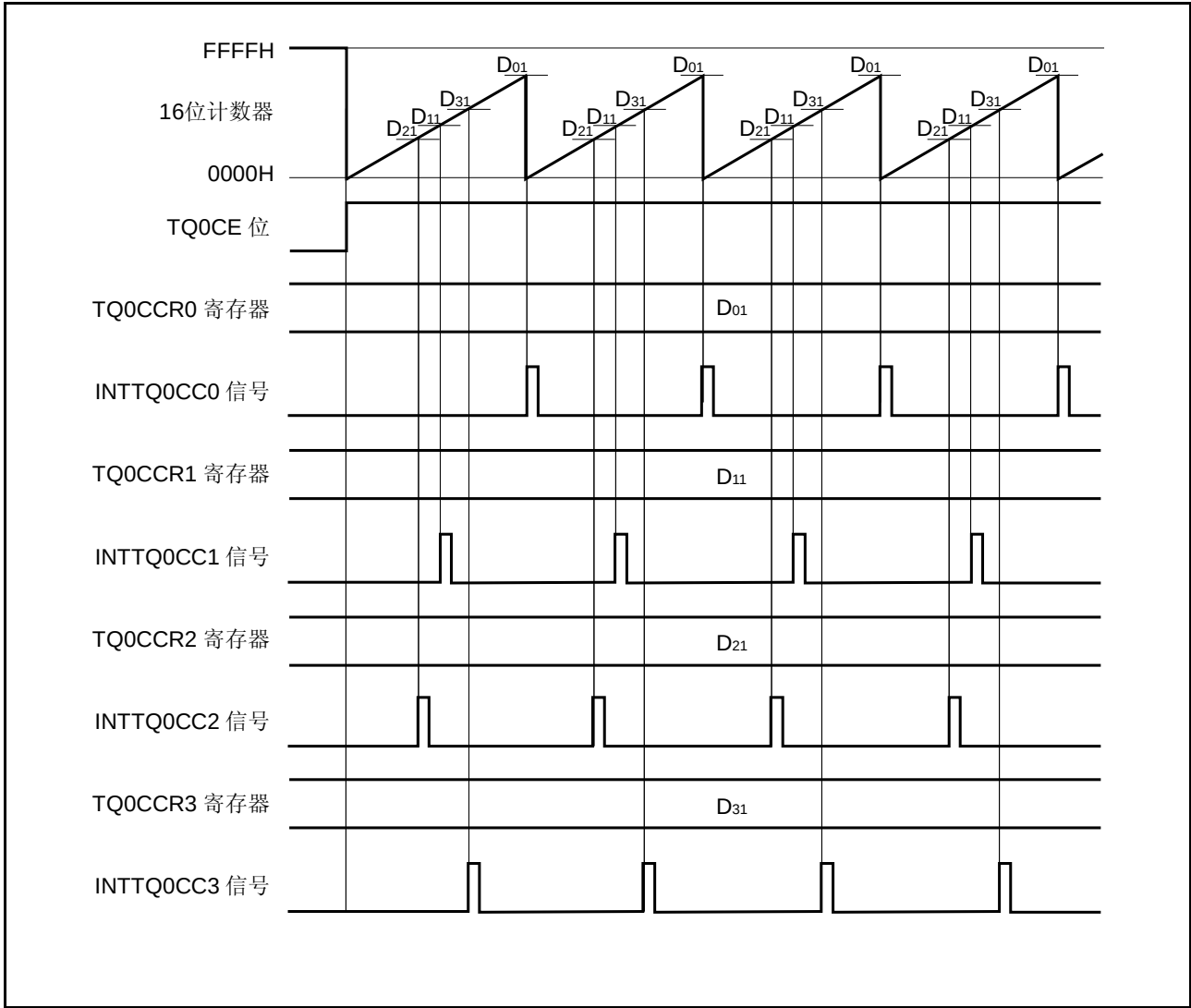
图 8-13. TQ0CCR1 至 TQ0CCR3 寄存器的配置



如果 TQ0CCRk 寄存器的设定值小于 TQ0CCR0 寄存器的设定值，则每个计数周期都产生一次 INTTQ0CCk 中断请求信号。

备注 k=1 至 3

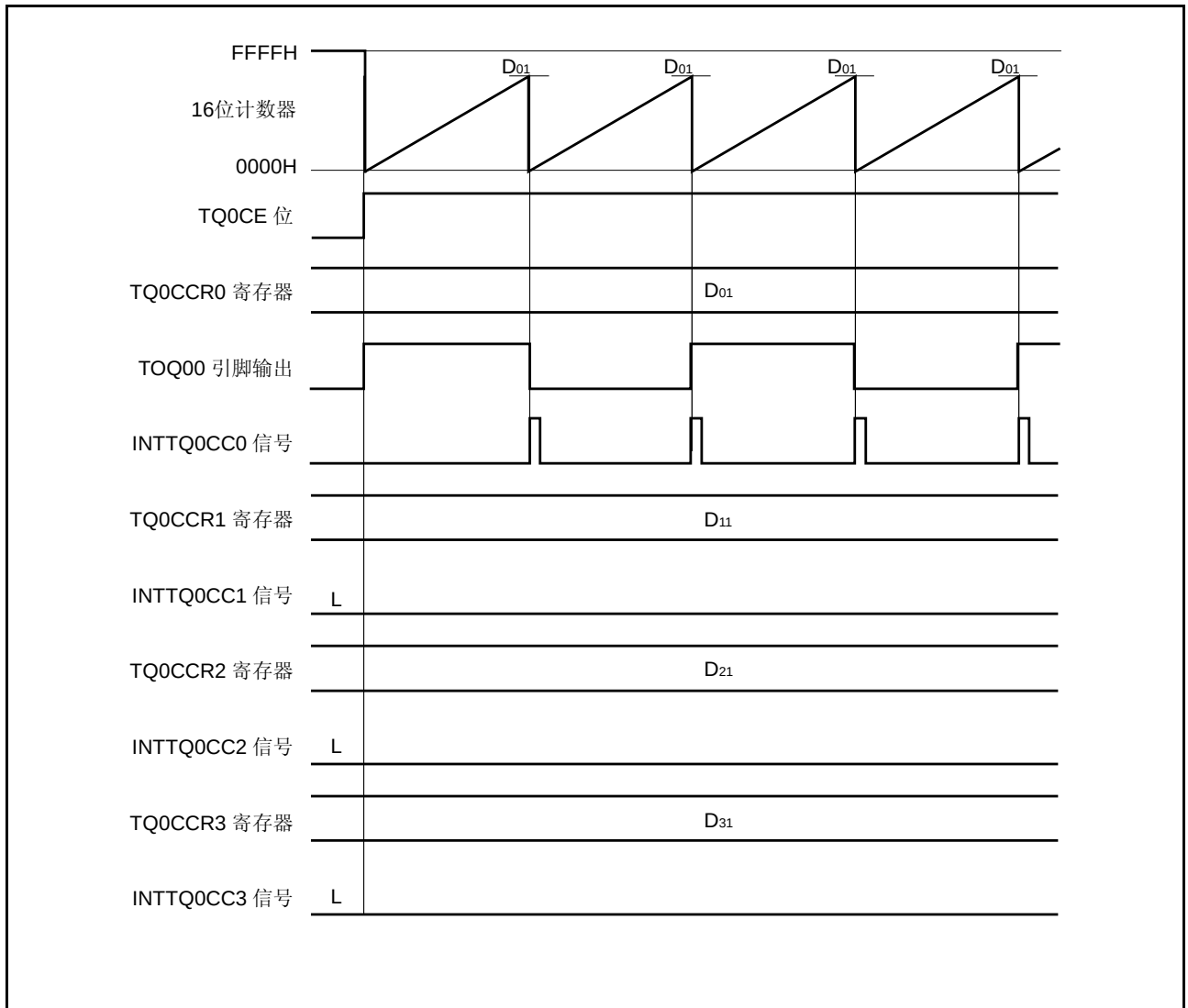
图 8-14. 当 $D_{01} \geq D_{k1}$ 时的时序图



如果 TQ0CCRk 寄存器的设定值大于 TQ0CCR0 寄存器的设定值, 16 位计数器的计数值不会和 TQ0CCRk 寄存器的值发生匹配。因此也就不会产生 INTTQ0CCk 中断请求信号。

备注 k=1 至 3

图 8-15. 当 $D_{01} < D_{k1}$ 时的时序图



8.5.3 外部触发脉冲输出模式 (TQ0MD2 至 TQ0MD0 位=010)

在外部触发脉冲输出模式下，当 TQ0CTL0.TQ0CE 位被置为 1 时，16 位定时器/事件计数器 Q 等待触发信号，当检测到外部触发输入信号的有效边沿后，16 位定时器/事件计数器 Q 开始计数，并且从 TOQ01 至 TOQ03 引脚输出一个 PWM 波形。

不使用外部触发，而是使用软件触发，同样可以得到输出脉冲。当使用软件触发时，TOQ00 引脚输出一个方波，其半周期等于输出 PWM 的一个周期。

图 8-16. 外部触发脉冲输出模式的配置

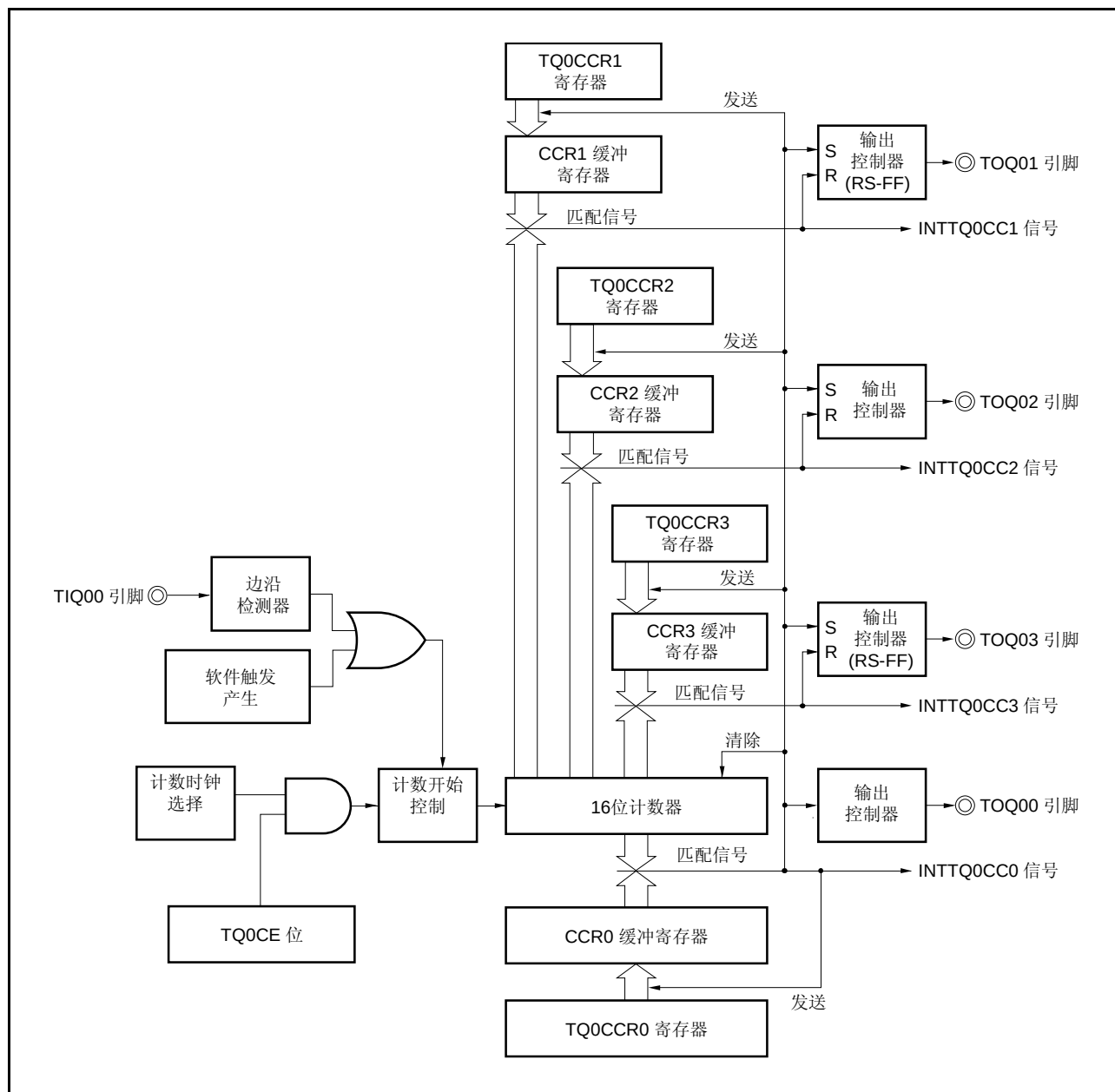
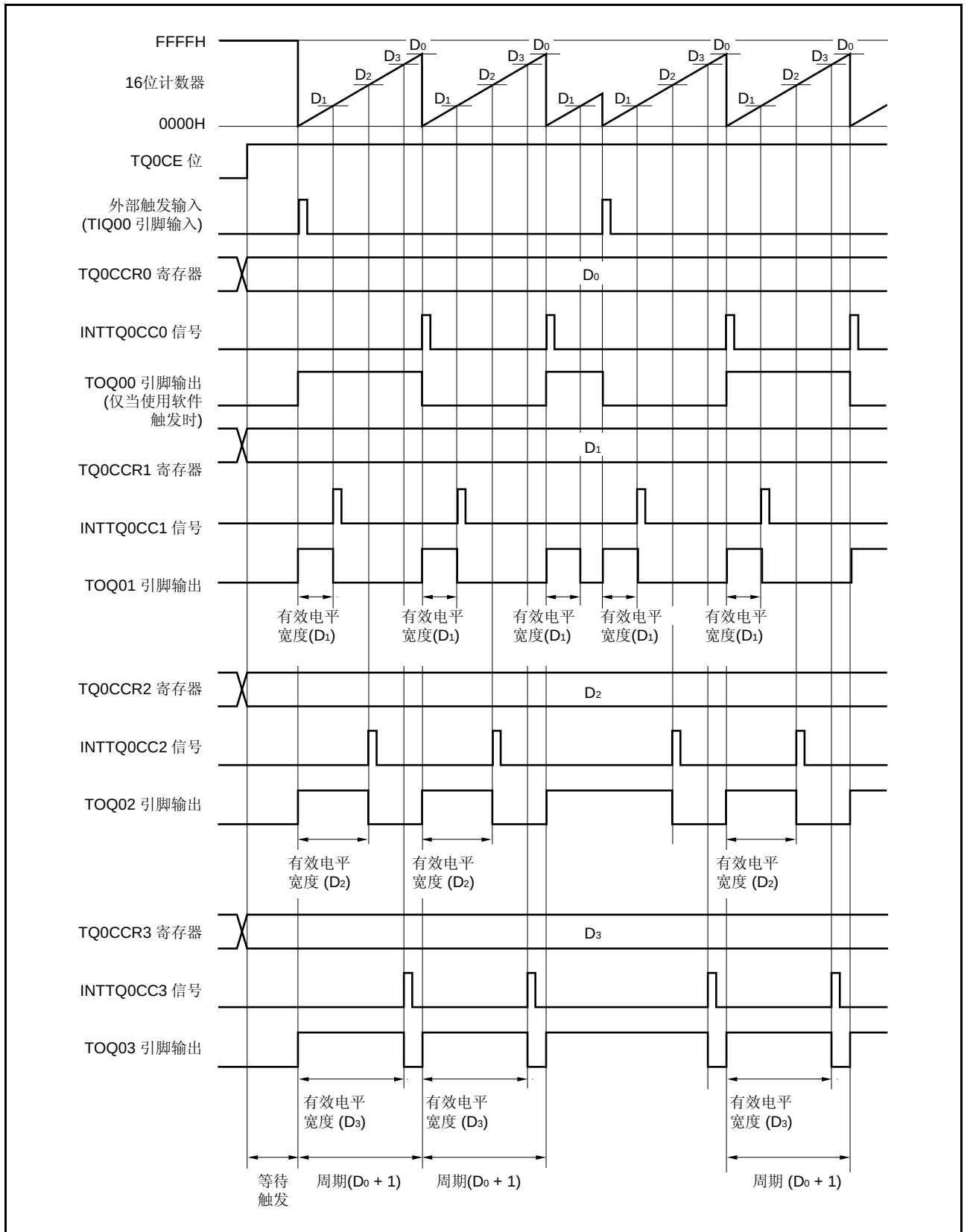


图 8-17. 外部触发脉冲输出模式的基本时序



当 TQ0CE 位设置为 1 时，16 位定时器/事件计数器 Q 等待触发。当触发产生时，16 位计数器从 FFFFH 清零为 0000H，同时开始计数，并从 TOQ0k 引脚输出一个 PWM 波形。如果在计数运行中再次产生触发，则计数器清为 0000H 并重新开始计数。（TOQ00 引脚输出反相。在产生触发时，不管当前状态如何（高/低），TOQ0k 引脚都会输出一个高电平。

PWM 波形的有效电平宽度、周期、占空比系数的计算如下。

有效电平宽度 = (TQ0CCRk 寄存器的设置值) × 计数时钟周期
周期 = (TQ0CCR0 寄存器的设置值 + 1) × 计数时钟周期
占空比系数 = (TQ0CCRk 寄存器的设置值) / (TQ0CCR0 寄存器的设置值 + 1)

当 16 位计数器在其计数值与 CCR0 缓冲寄存器的值匹配之后进行下一次计数时就会同步产生比较匹配（中断）请求信号 INTTQ0CC0，且 16 位计数器清零为 0000H。当 16 位计数器的计数值与 CCRk 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTQ0CCK。

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配，而且 16 位计数器清零为 0000H 时，TQ0CCRm 寄存器的设置值就被传送到 CCRm 缓冲寄存器。

外部触发输入信号的有效边沿，或将软件触发（TQ0CTL1.TQ0EST 位）设为 1，都可以用作触发信号。

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）

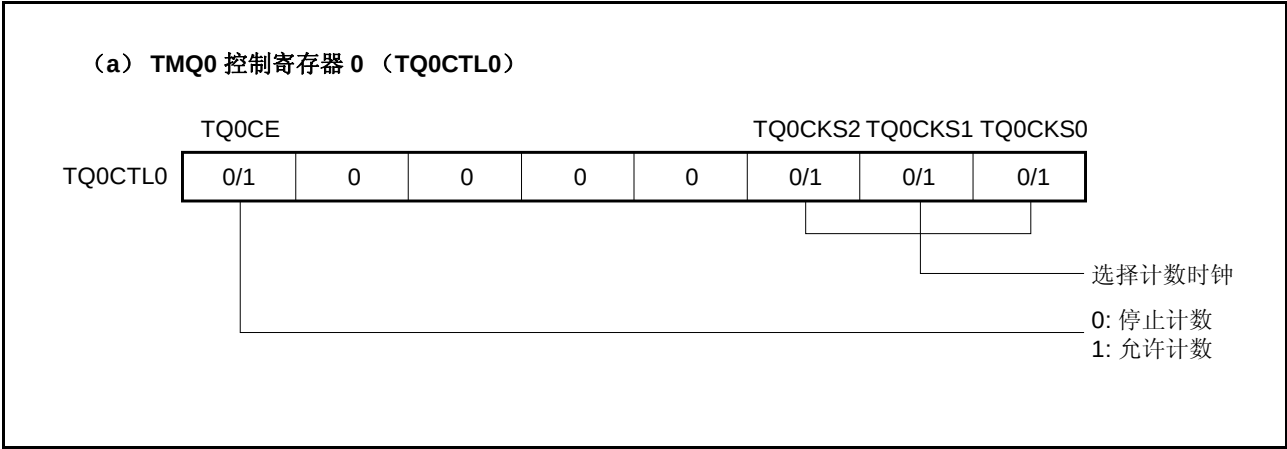
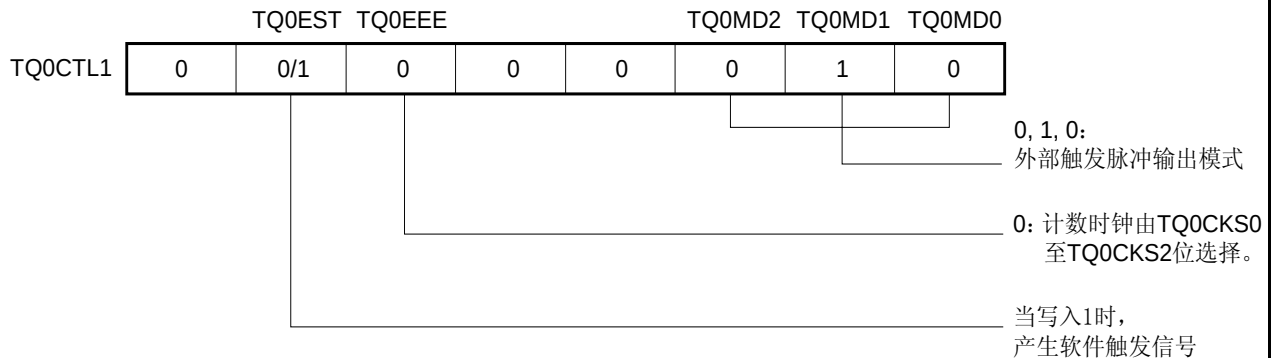


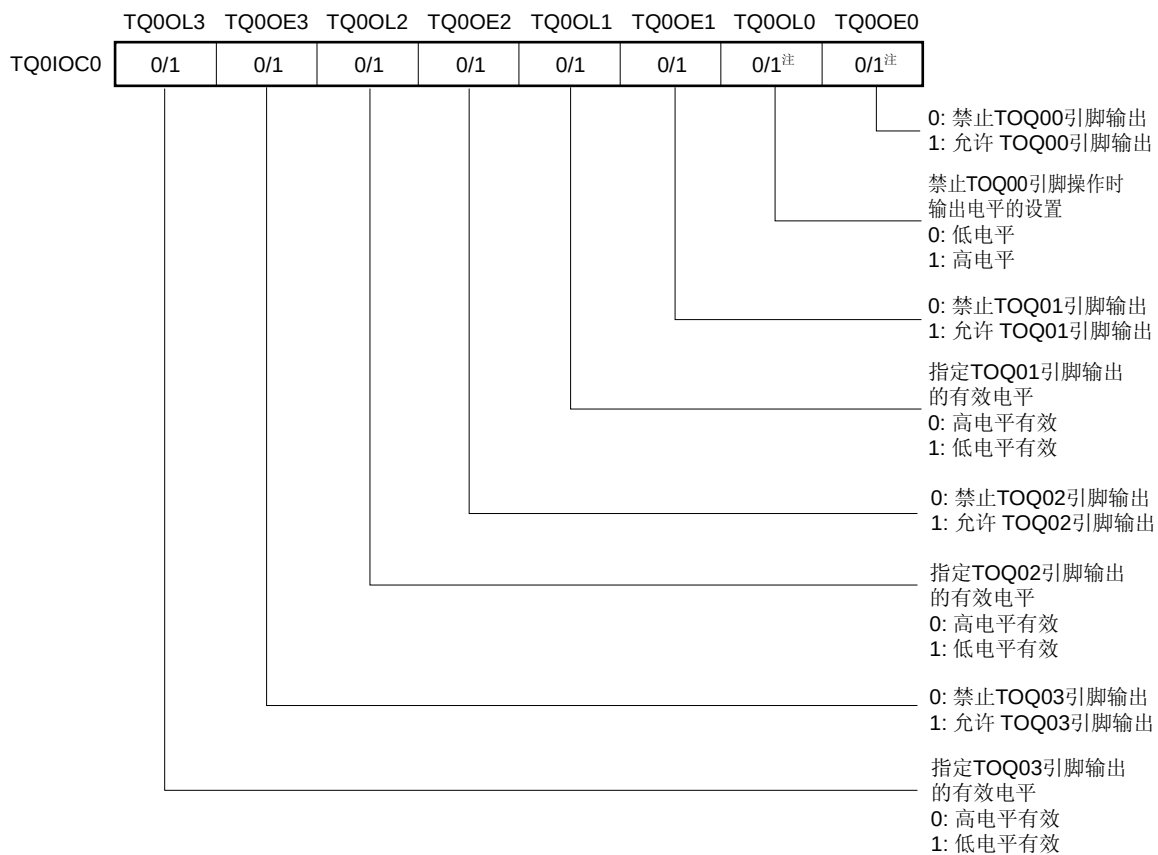
图 8-18. 外部触发脉冲输出模式的寄存器设置 (2/3)

<R>

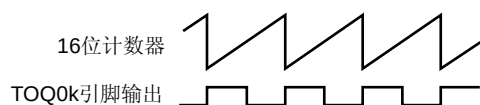
(b) TMQ0 控制寄存器 1 (TQ0CTL1)



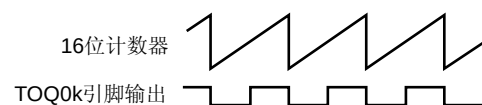
(c) TMQ0 输入/输出控制寄存器 0 (TQ0IOC0)



· 当TQ0OLk 位 = 0时



· 当TQ0OLk 位 = 1时

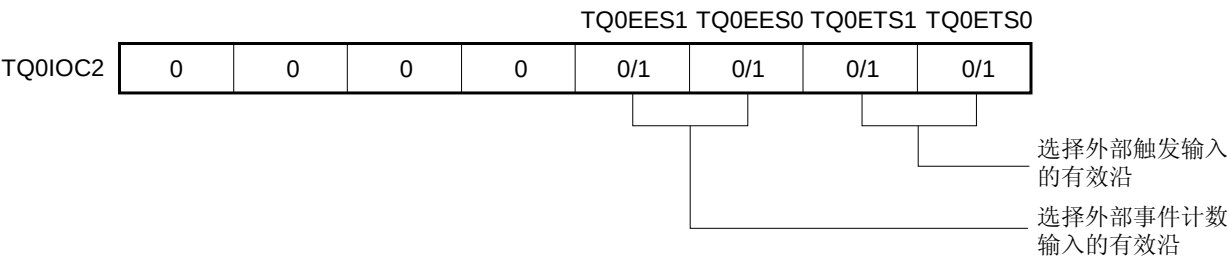


注 在外部触发脉冲输出模式下，不使用 TOQ00 引脚时，将该位清除为 0。

图 8-18. 外部触发脉冲输出模式的寄存器设置 (3/3)

<R>

(d) TMQ0 输入/输出控制寄存器 2 (TQ0IOC2)



(e) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

通过读取 TQ0CNT 寄存器，可以读出 16 位计数器的值。

(f) TMQ0 捕获/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 TQ0CCR1 寄存器的值设置为 D_1 ，将 TQ0CCR2 寄存器的值设置为 D_2 ，将 TQ0CCR3 寄存器的值设置为 D_3 ，则 PWM 波形的周期和有效电平的计算如下：

周期 = $(D_0 + 1) \times$ 计数时钟周期

TOQ01 引脚 PWM 波形有效电平宽度 = $D_1 \times$ 计数时钟周期

TOQ02 引脚 PWM 波形有效电平宽度 = $D_2 \times$ 计数时钟周期

TOQ03 引脚 PWM 波形有效电平宽度 = $D_3 \times$ 计数时钟周期

- 备注**
1. 在外部触发脉冲输出模式下，不使用 TMQ0 输入/输出控制寄存器 1 (TQ0IOC1) 和 TMQ0 选项寄存器 0 (TQ0OPT0)。
 2. 通过写入 TMQ0 捕获/比较寄存器 1 (TQ0CCR1)，可以更新 TMQ0 捕获/比较寄存器 (TQ0CCR2) 和 TMQ0 捕获/比较寄存器 3 (TQ0CCR3)。

(1) 外部触发脉冲输出模式的操作系统

图 8-19. 外部触发脉冲输出模式的软件处理流程 (1/2)

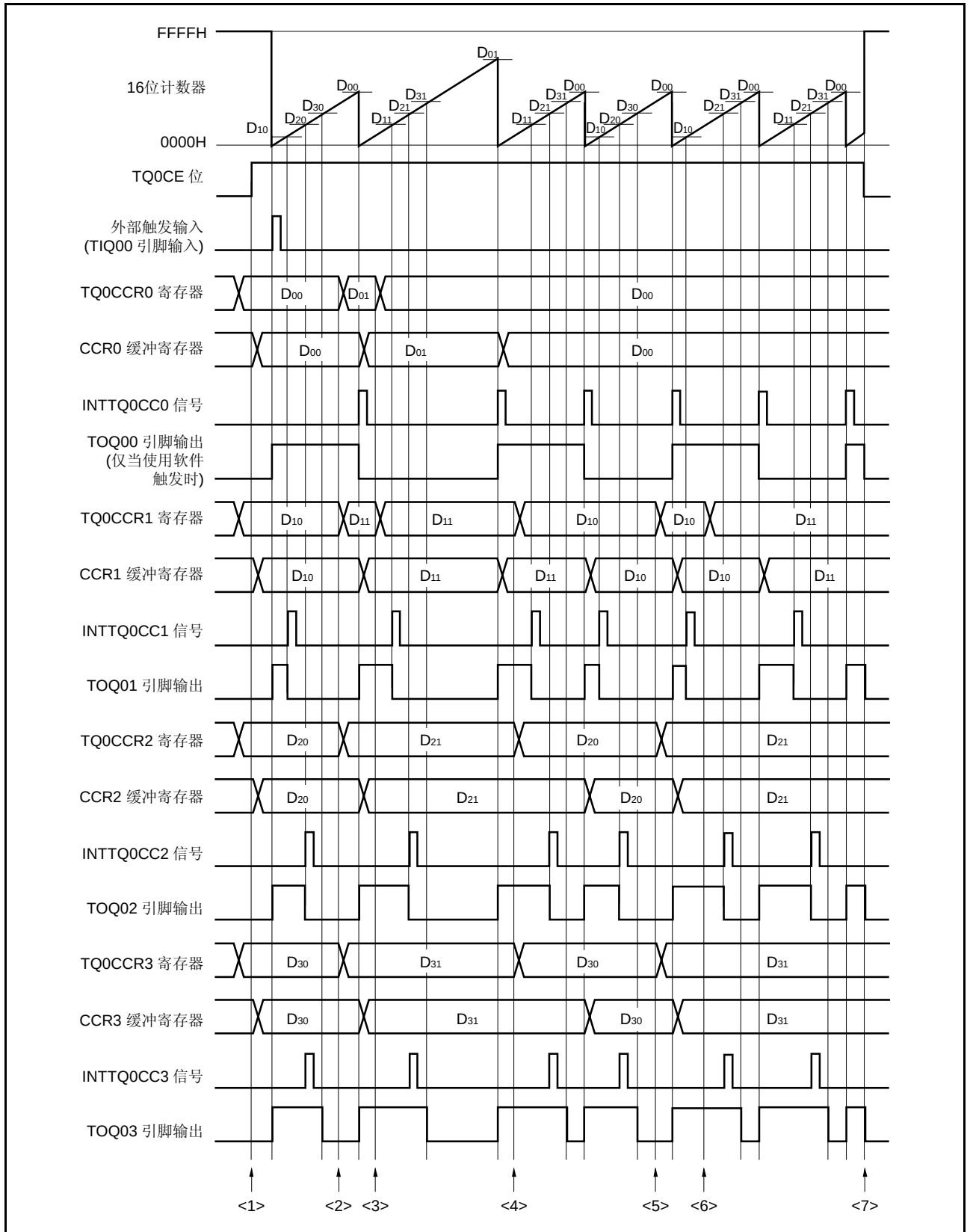
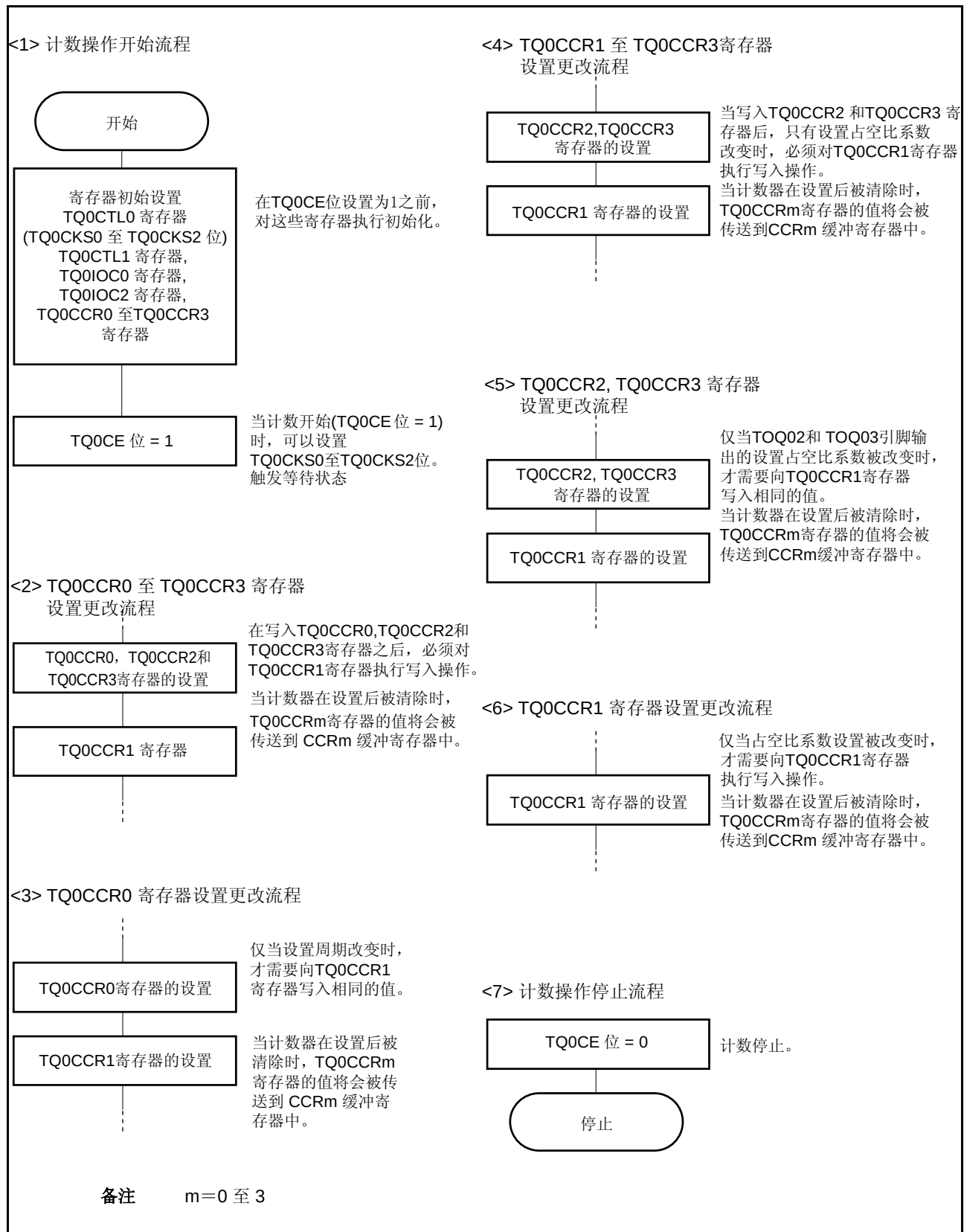


图 8-19. 外部触发脉冲输出模式的软件处理流程 (2/2)

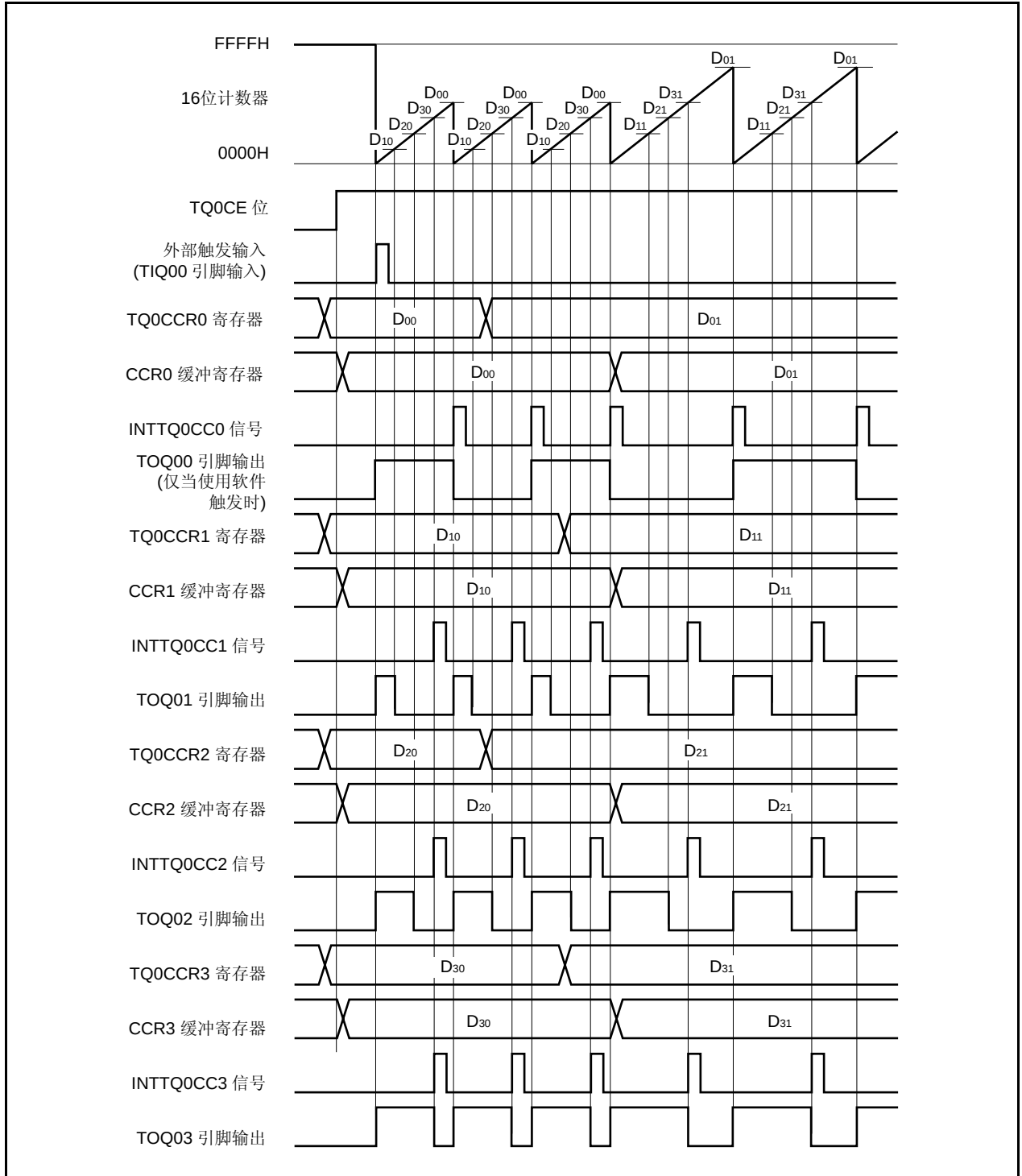


(2) 外部触发脉冲输出模式的操作时序

(a) 有关操作中改变脉冲宽度的注意事项

计数器计数过程中，若要改变 PWM 波形，应重新写入 TQ0CCR1 寄存器。

当检测到 INTTQ0CC0 信号以后，在 TQ0CCR1 寄存器写入操作之后重写 TQ0CCRk 寄存器。



为了将数据从 TQ0CCRM 寄存器传送到 CCRm 缓冲寄存器，必须写入 TQ0CCR1 寄存器。

如果 PWM 波形的周期和有效电平宽度都要改变，先将周期设定到 TQ0CCR0 寄存器，将有效电平宽度设定到 TQ0CCR2 和 TQ0CCR3 寄存器，然后将有效电平设定到 TQ0CCR1 寄存器。

如果只改变 PWM 波形的周期，应先把周期设置到 TQ0CCR0 寄存器，然后再将相同的值写入 TQ0CCR1 寄存器。

如果只改变 PWM 波形的有效电平宽度（占空比系数），需要先向 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。

如果只改变 TOQ01 引脚输出的 PWM 波形的有效电平宽度（占空比系数），仅需要设置 TQ0CCR1 寄存器。

如果只改变 TOQ02 和 TOQ03 引脚输出的 PWM 波形的有效电平宽度（占空比系数），要先向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平宽度，然后再将相同的值写入 TQ0CCR1 寄存器。

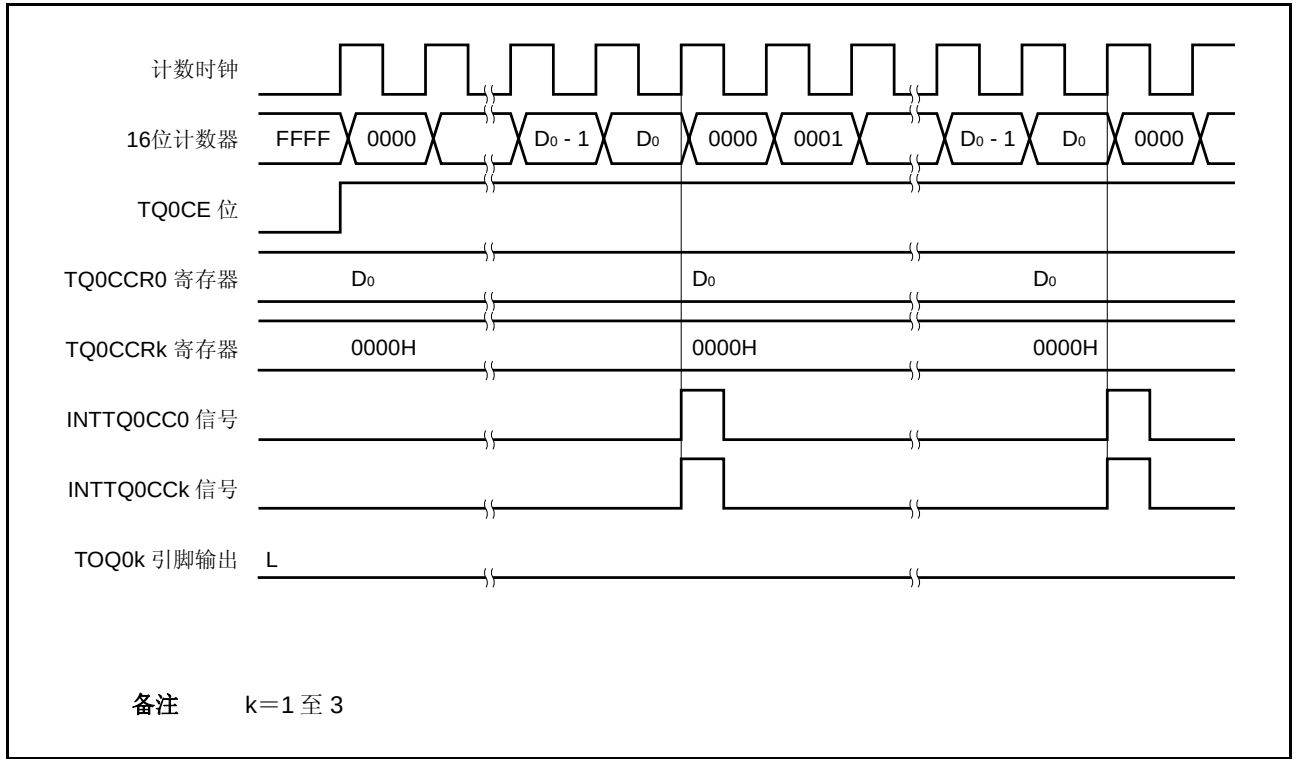
在数据写入 TQ0CCR1 寄存器后，在对 16 位计数器清零的同时，写入到 TQ0CCRM 寄存器的值会被传送到 CCRm 缓冲寄存器，并作为与 16 位计数器进行比较的值。

在写入 TQ0CCR1 寄存器一次之后，要再次写入 TQ0CCR0 至 TQ0CCR3 寄存器，应该在 INTTQ0CC0 信号产生之后进行。否则，CCRM 缓冲寄存器的值可能不确定，因为把数据从 TQ0CCRM 寄存器传送到 CCRm 缓冲寄存器的时序会和写入 TQ0CCRM 寄存器的时序发生冲突。

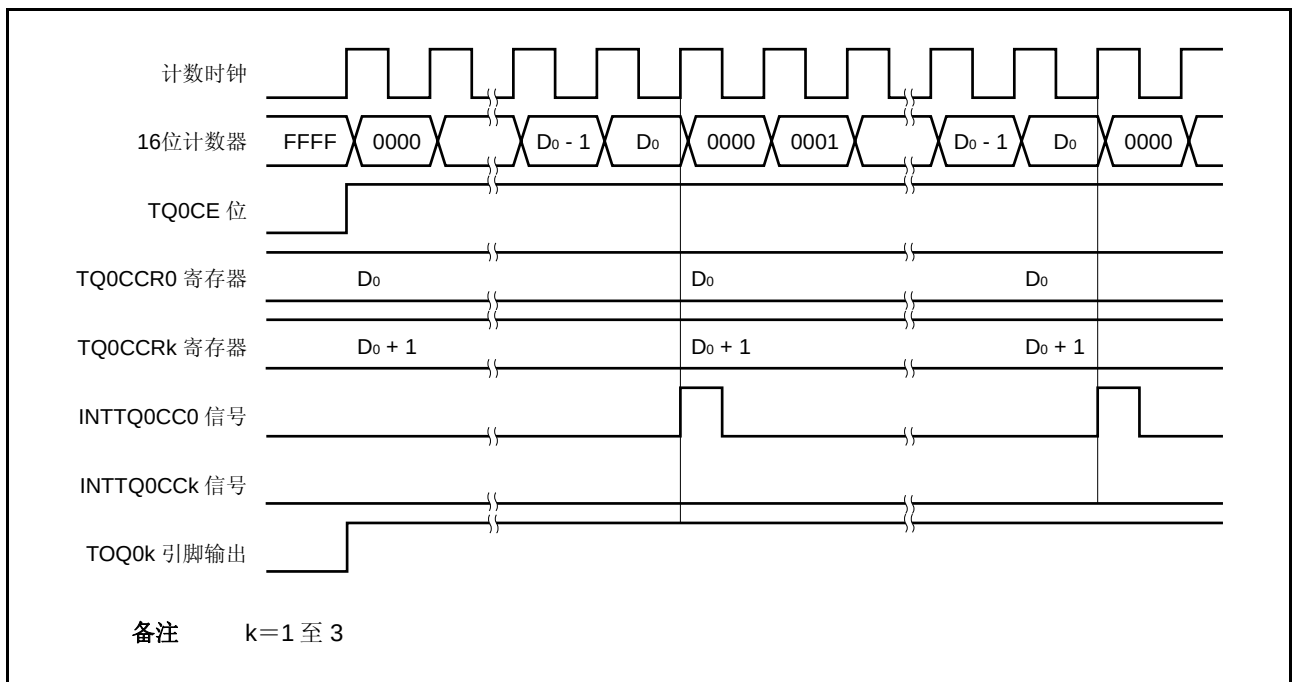
备注 m=0 至 3

(b) PWM 波形的 0%和 100% 输出

为了输出 0% 的波形，需要将 TQ0CCRk 寄存器设置为 0000H。如果 TQ0CCR0 寄存器的设定值为 FFFFH，则会在每个周期产生 INTTQ0CCK 中断请求信号。

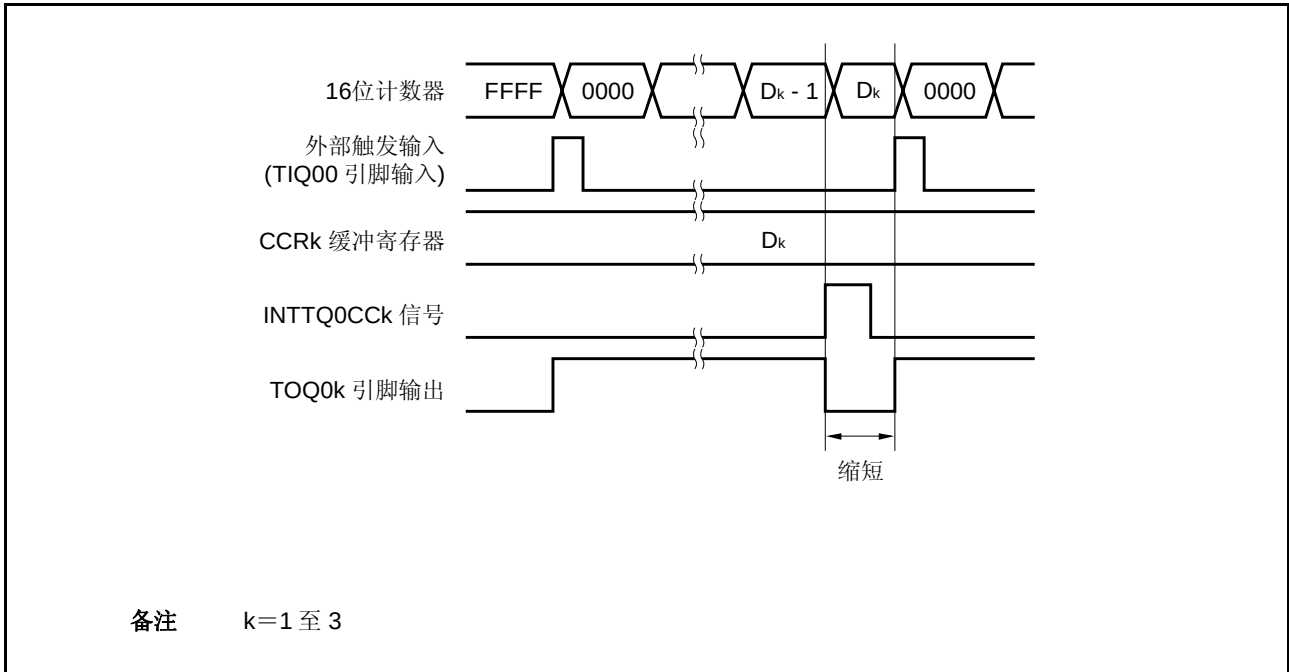


为了输出 100% 的波形，需要将 TQ0CCRk 寄存器的值设为：(TQ0CCR0 寄存器的设定值+1)。如果 TQ0CCR0 寄存器的设定值为 FFFFH，则不会产生 100% 波形输出。

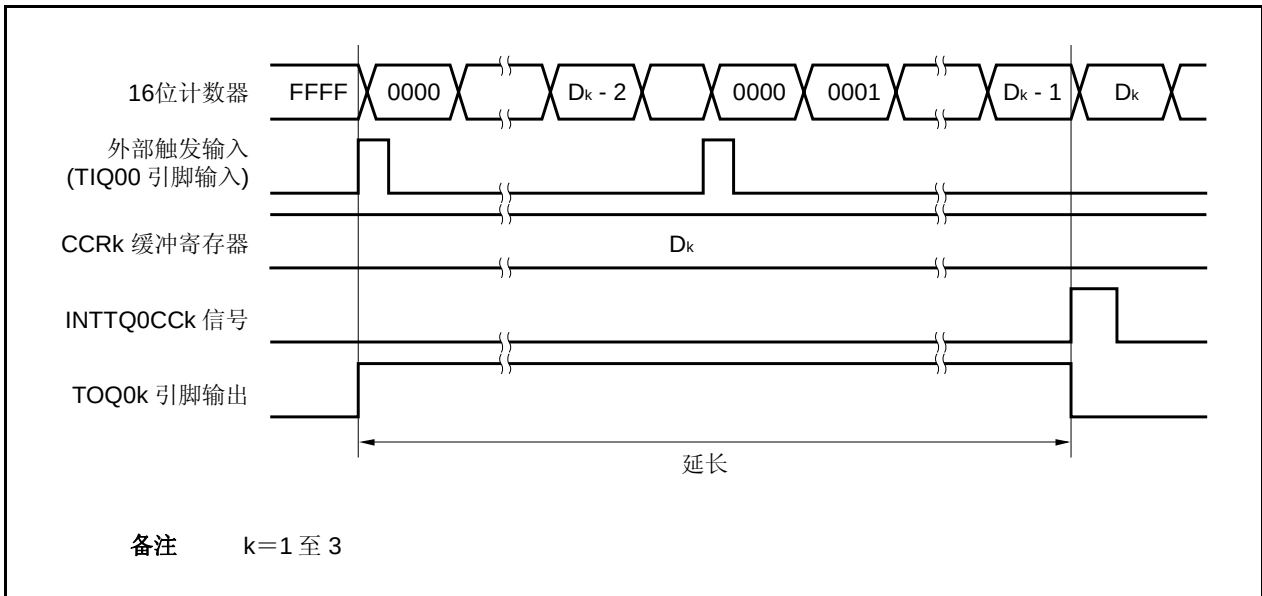


(c) 触发检测和 CCRk 缓冲寄存器匹配之间的冲突

如果在 INTTQ0CCK 信号产生后，立刻检测到触发信号，则 16 位计数器立刻清零为 0000H，TOQ0k 引脚的输出信号生效，计数器继续计数。结果，PWM 波形的无效时段被缩短。

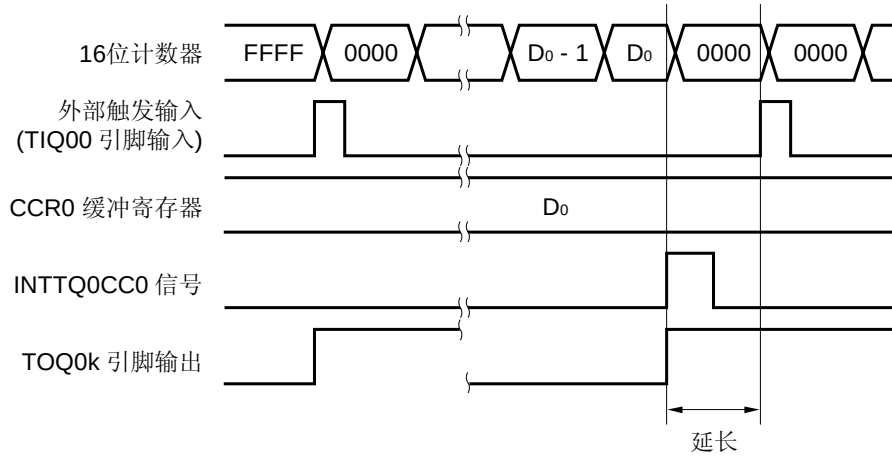


如果紧邻在 INTTQ0CCK 信号产生之前检测到触发信号，则不会产生 INTTQ0CCK 信号，而 16 位计数器清零为 0000H，计数继续进行。TOQ0k 引脚的输出信号仍然保持有效。结果，PWM 波形的有效时段被延长。



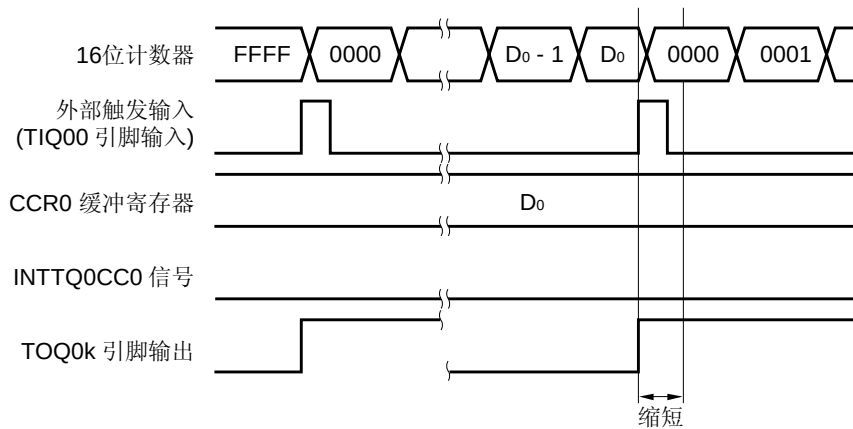
(d) 触发检测和 CCR0 缓冲寄存器匹配之间的冲突

如果在 INTTQ0CC0 信号产生后，立刻检测到触发信号，则 16 位计数器立刻清零为 0000H，并继续计数。因此，TOQ0k 引脚的有效时段被延长，延长长度为 INTTQ0CC0 信号产生时刻至检测到触发信号时刻的时间间隔。



备注 k=1 至 3

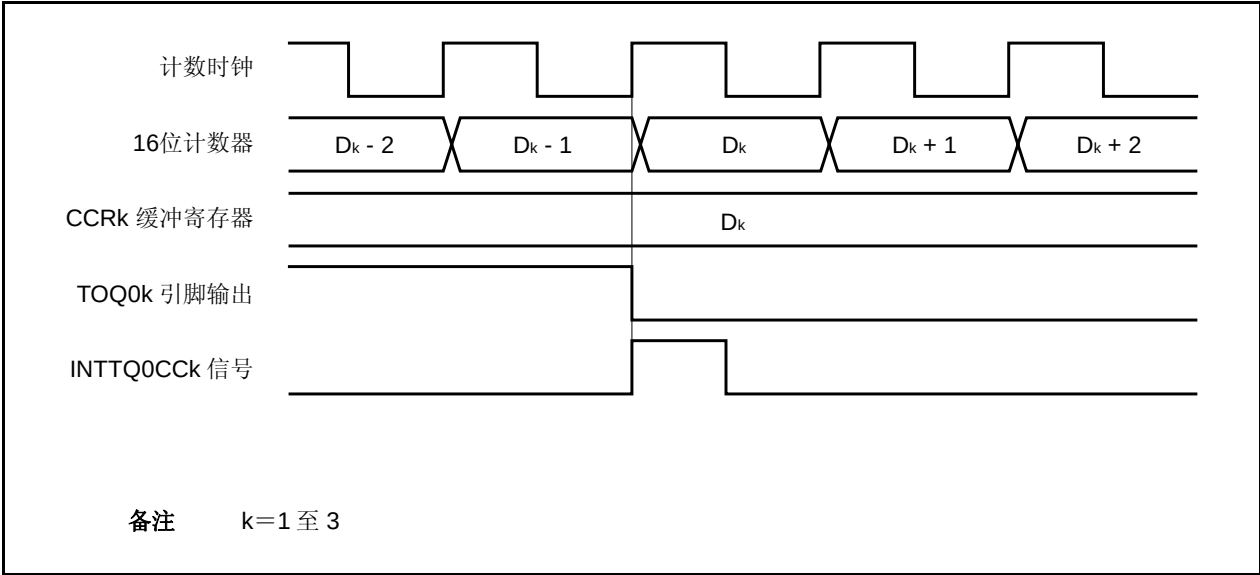
如果紧邻在 INTTQ0CC0 信号产生之前检测到触发信号，则不会产生 INTTQ0CC0 信号。16 位计数器清零为 0000H，TOQ0k 引脚输出有效，计数器继续计数。结果，PWM 波形的无效时段被缩短。



备注 k=1 至 3

(e) 比较匹配中断请求信号 (INTTQ0CCK) 的发生时序

在外部触发脉冲输出模式下，INTTQ0CCK 信号的发生时序与其它情况下 INTTQ0CCK 信号的发生时序不同；当 16 位计数器的计数值与 CCRk 缓冲寄存器的值匹配时，产生 INTTQ0CCK 中断请求信号。



通常在 16 位计数器的当前计数值与 CCRk 缓冲寄存器的值匹配之后，在下一次计数时同步发生 INTTQ0CCK 信号。

但是，在外部触发脉冲输出模式下，INTTQ0CCK 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOQ0k 引脚输出信号变化的时序。

8.5.4 单次触发脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 011)

在单次触发脉冲输出模式下，当 TQ0CTL0.TQ0CE 位被置为 1 时，16 位定时器/事件计数器 Q 等待触发。在检测到外部触发输入的有效边沿时，16 位定时器/事件计数器 Q 开始计数，同时从 TOQ01 至 TOQ03 引脚输出一个单次触发脉冲。

除外部触发外，也可以生成软件触发信号来输出该脉冲。在使用软件触发时，TOQ00 引脚在 16 位计数器计数时输出有效电平，在计数器停止（等待触发）时输出无效电平。

图 8-20. 单次触发脉冲输出模式的配置

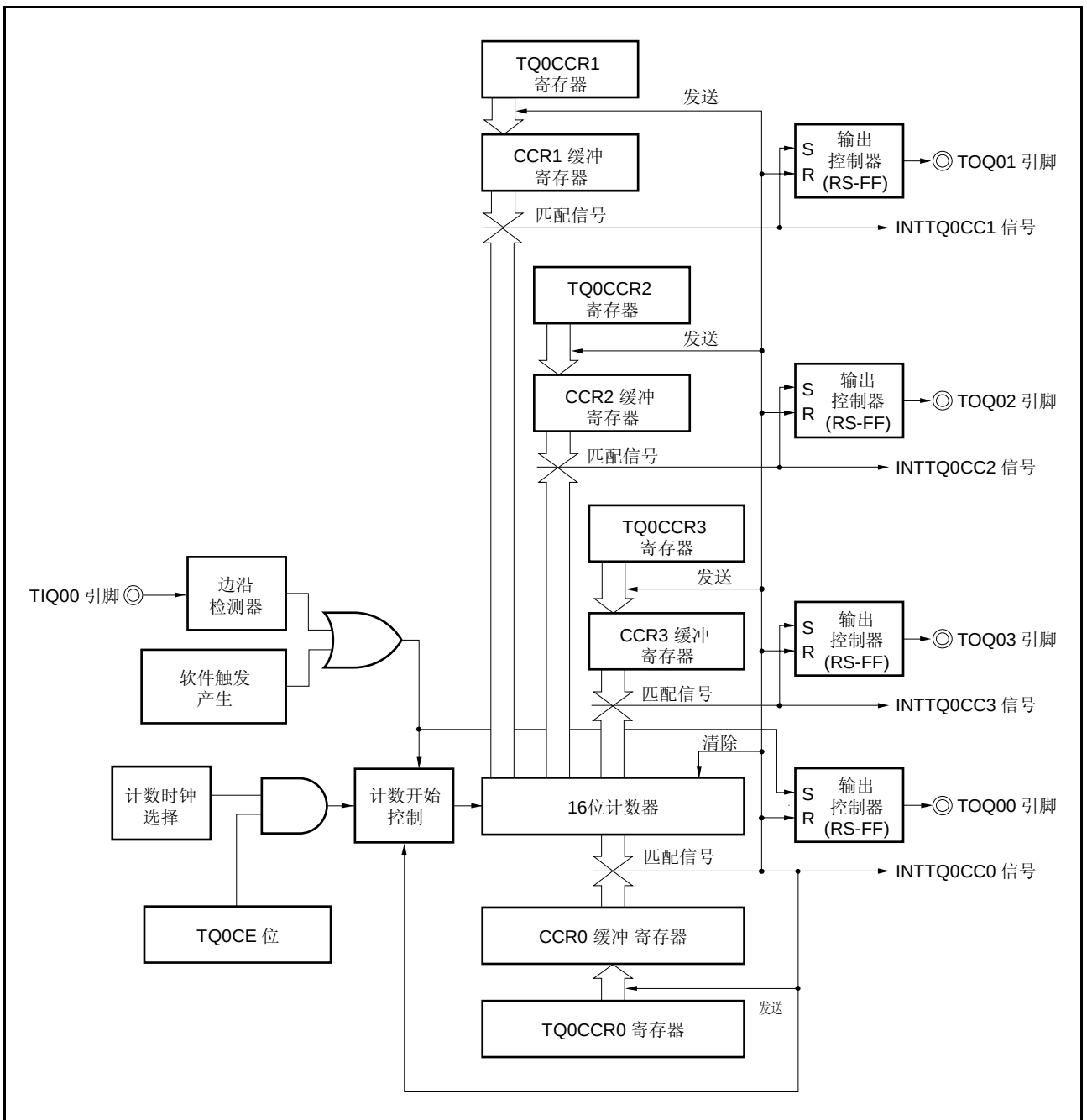
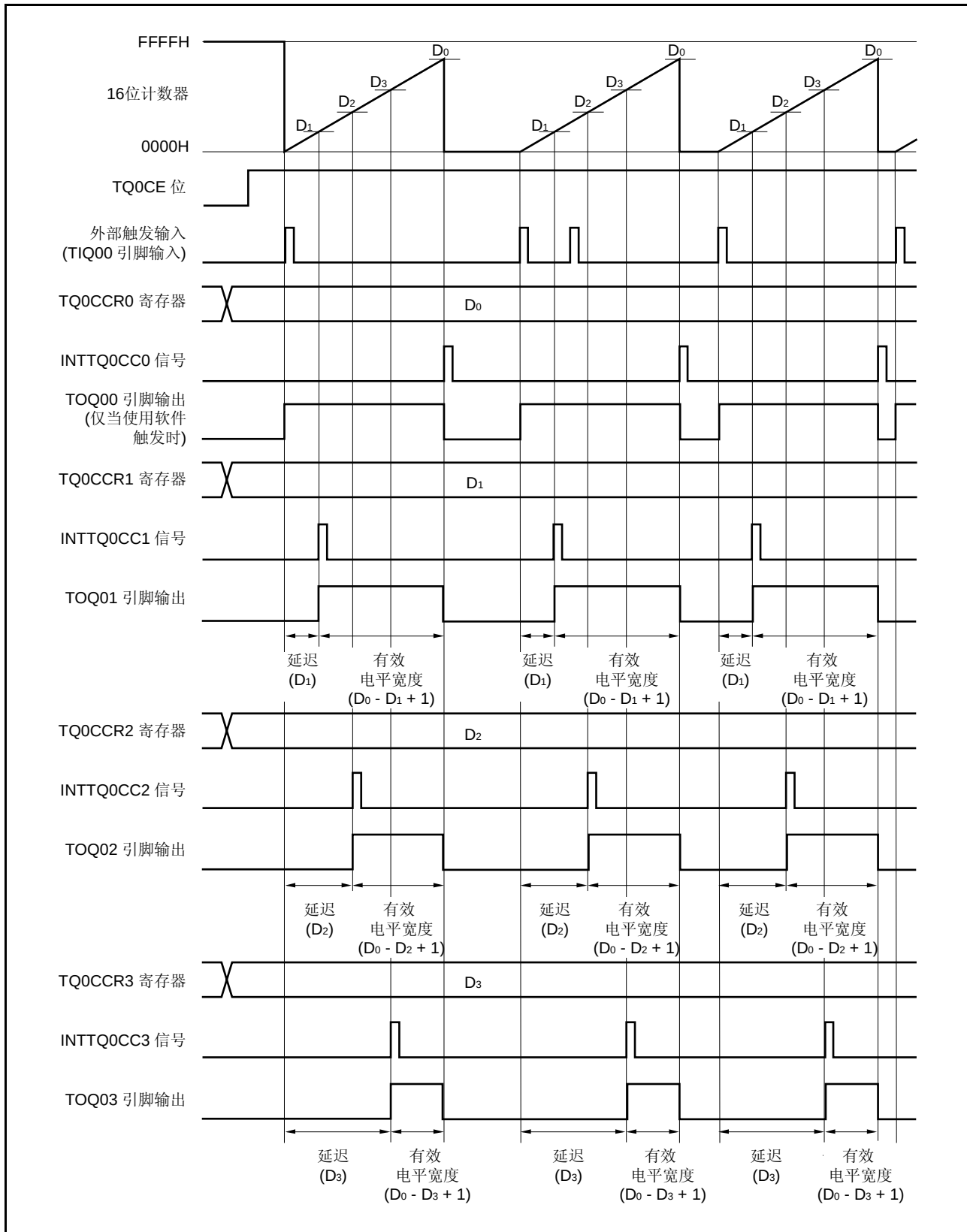


图 8-21. 单次触发脉冲输出模式的基本时序



当 TQ0CE 位置为 1 时，16 位定时器/事件计数器 Q 等待触发。在触发信号发生时，16 位计数器从 FFFFH 清零为 0000H，开始进行计数，并从 TOQ0k 引脚输出一个单次触发脉冲。在输出单次触发脉冲后，16 位计数器设置为 FFFFH，计数停止，并重新等待触发。如果在单次触发脉冲输出期间再次产生触发信号，则忽略该触发信号。

单次触发脉冲的输出延迟时间和有效电平宽度可计算如下：

输出延迟时间 = (TQ0CCRk 寄存器的设定值) × 计数时钟周期

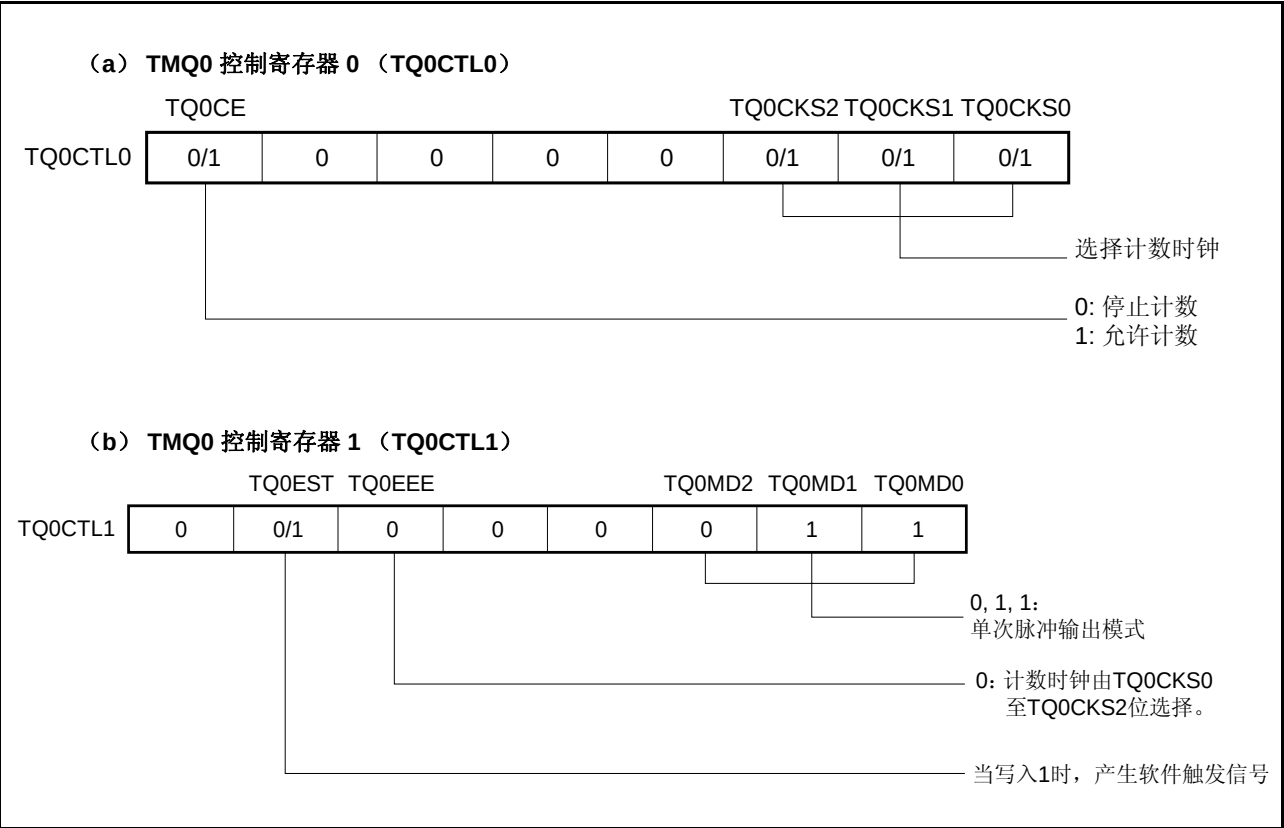
有效电平宽度 = (TQ0CCR0 寄存器的设定值 - TQ0CCRk 寄存器的设定值 + 1) × 计数时钟周期

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配之后，产生比较匹配中断请求信号 INTTQ0CC0。当 16 位计数器的计数值与 CCRk 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTQ0CCk。

外部触发输入信号的有效边沿或将软件触发 (TQ0CTL1.TQ0EST 位) 信号设置为 1，都可用作触发信号。

备注 k=1 至 3

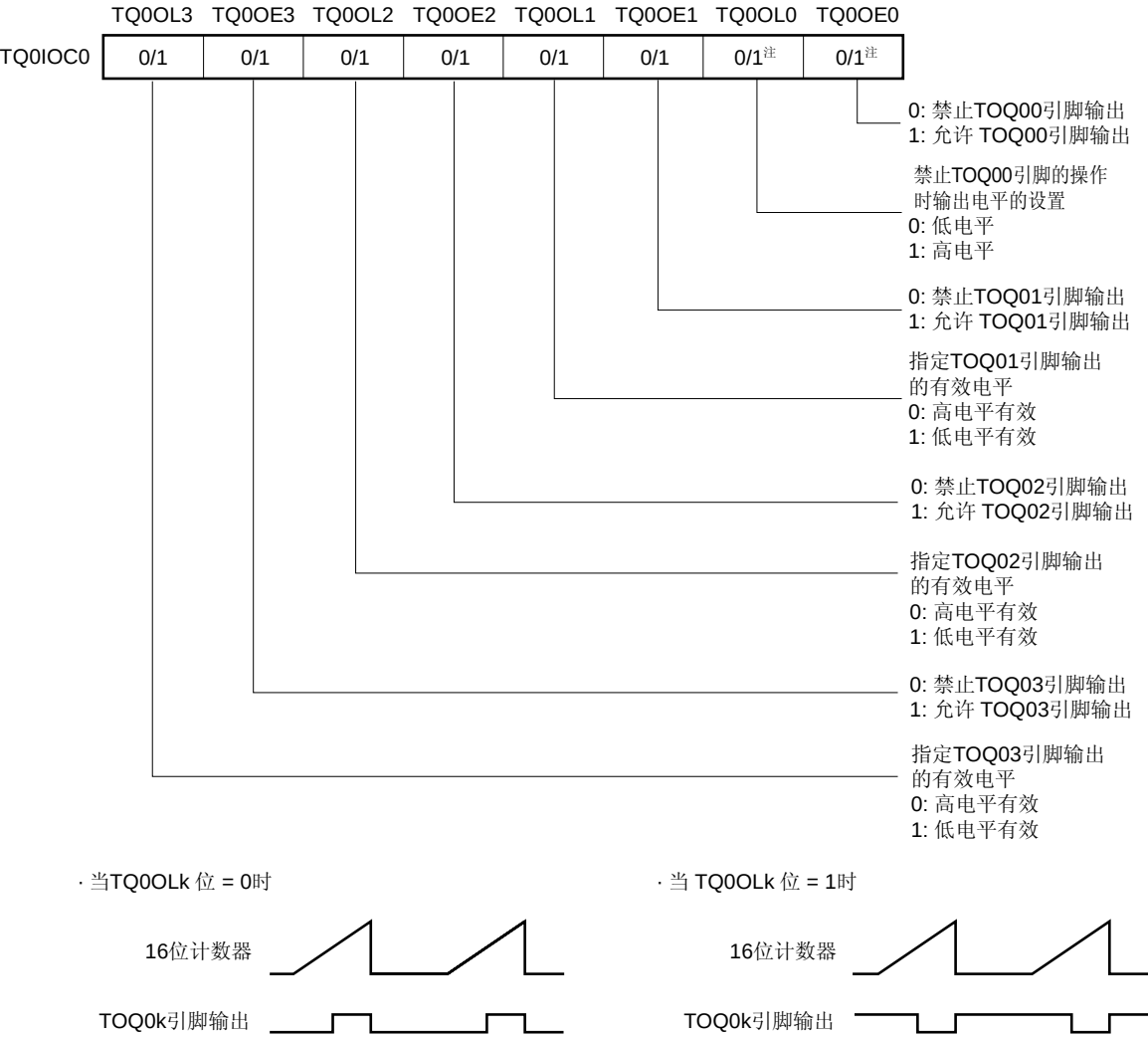
图 8-22. 单次触发脉冲输出模式时的寄存器设置 (1/3)



<R>

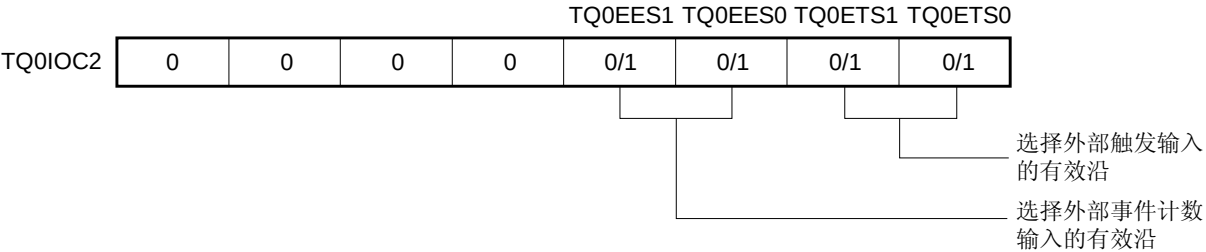
图 8-22. 单次触发脉冲输出模式时的寄存器设置 (2/3)

(c) TMQ0 输入/输出控制寄存器 0 (TQ0IOC0)



<R>

(d) TMQ0 输入/输出控制寄存器 2 (TQ0IOC2)



(e) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

通过读取 TQ0CNT 寄存器，可以读出 16 位计数器的值。

注 在单次触发脉冲输出模式下，不使用 TOQ00 引脚时，将该位清除为 0。

图 8-22. 单次触发脉冲输出模式时的寄存器设置 (3/3)

(f) TMQ0 捕获/比较寄存器 0 至 3 (TQ0CCR0 至 TQ0CCR3)

若把 TQ0CCR0 寄存器的值设置为 D_0 , TQ0CCR k 寄存器的值设置为 D_k , 则单次触发脉冲的有效电平宽度和输出延迟时间的计算如下。

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

输出延迟时间 = $D_k \times$ 计数时钟周期

注意事项 如果 TQ0CCR k 寄存器的设置值大于 TQ0CCR0 寄存器的设置值, 那么, 即使在单次触发脉冲输出模式下也不会输出单次触发脉冲。

备注

1. 在单次触发脉冲输出模式下, 不使用 TMQ0 输入/输出控制寄存器 1 (TQ0IOC1) 和 TMQ0 选项寄存器 0 (TQ0OPT0)。
2. $k=1$ 至 3

(1) 单次触发脉冲输出模式下的操作流程

图 8-23. 单次触发脉冲输出模式的软件处理流程 (1/2)

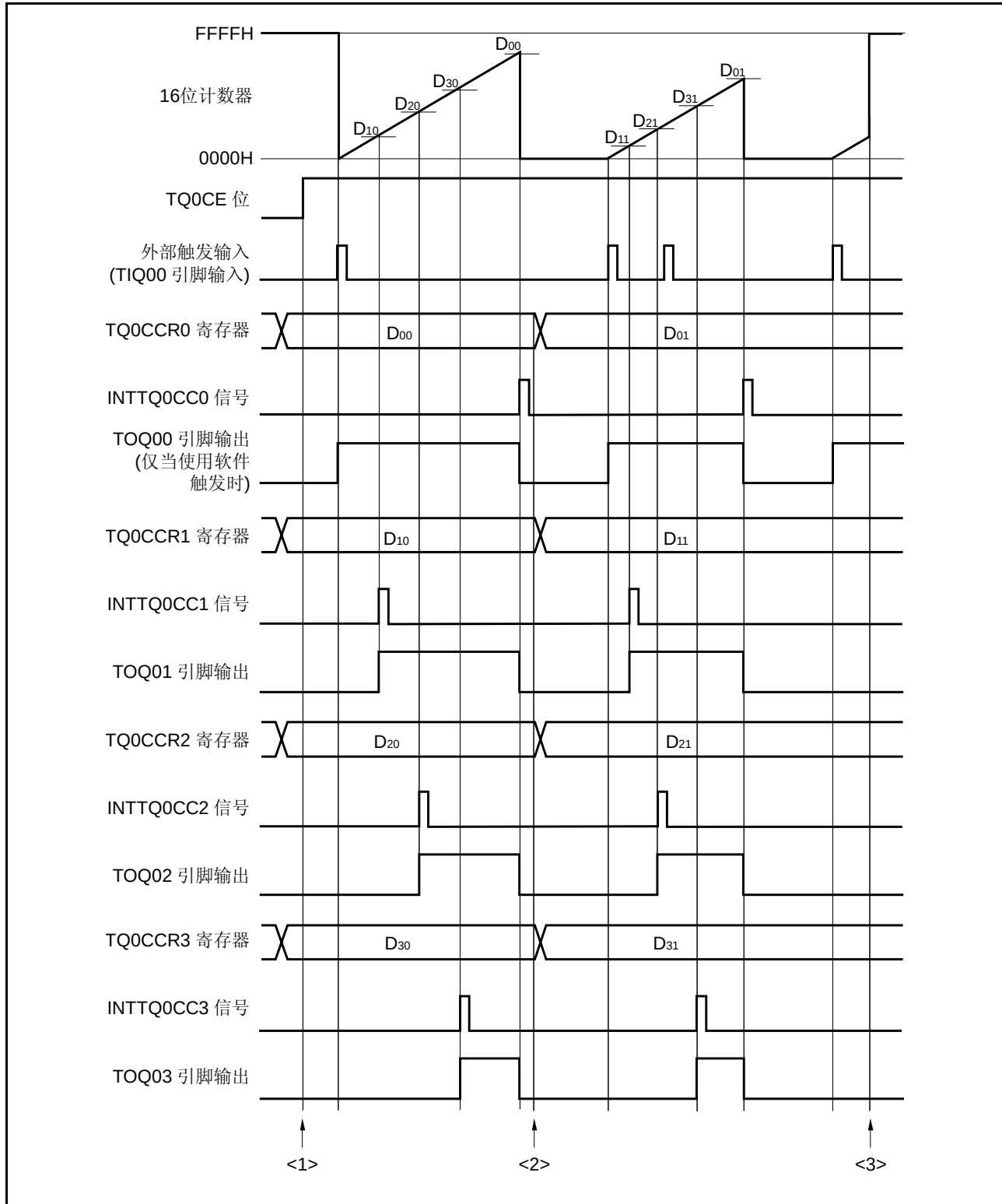
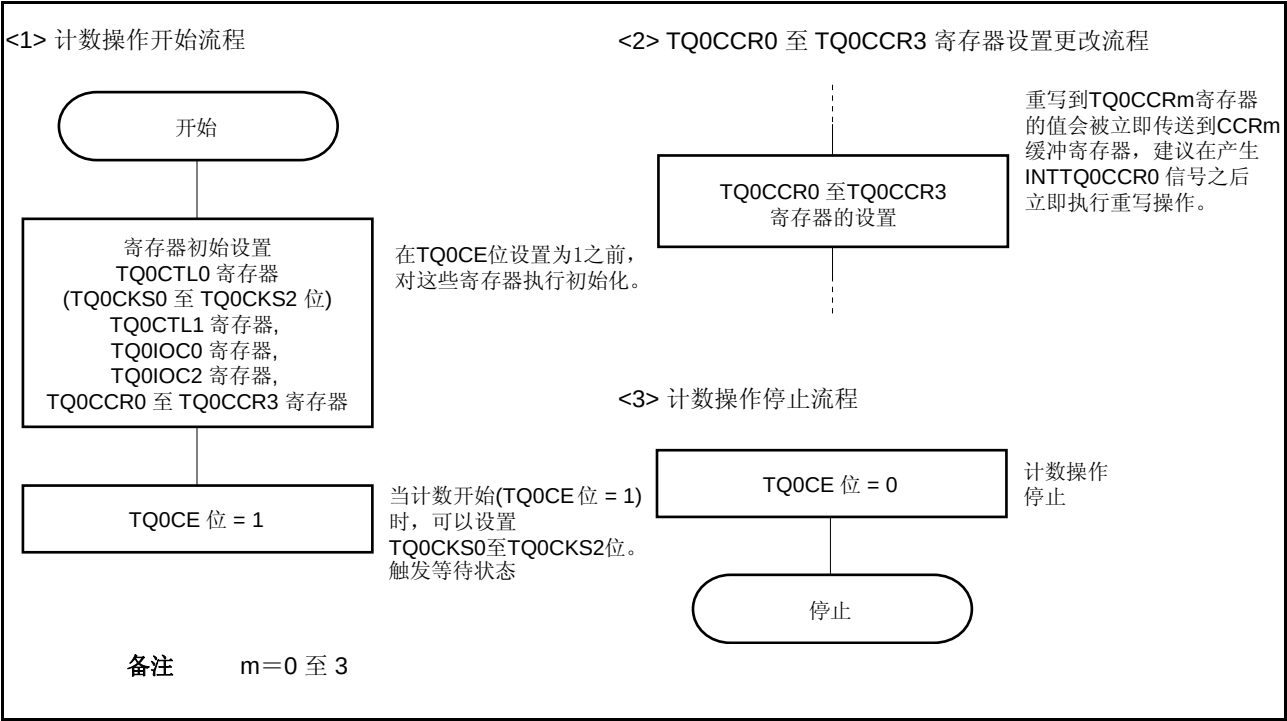


图 8-23. 单次触发脉冲输出模式的软件处理流程（2/2）

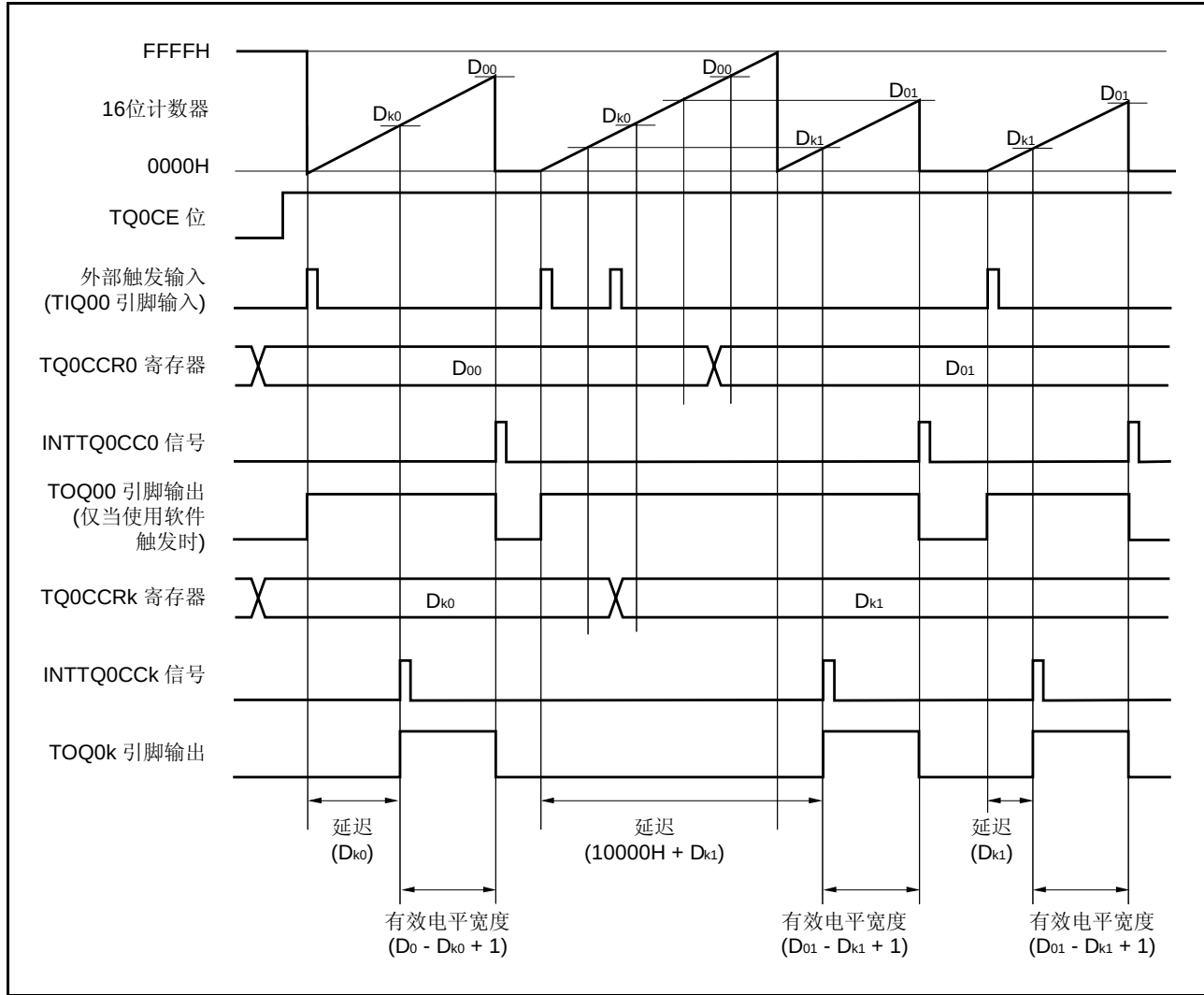


(2) 单次触发脉冲输出模式时的操作时序

(a) 重写 TQ0CCRm 寄存器的注意事项

要将 TQ0CCRm 寄存器的值改为比当前值较小的值，应停止计数，然后改变其设定值。

如果在计数器工作期间，将 TQ0CCR0 寄存器的值重写为比当前值较小的值，则 16 位计数器可能会溢出。



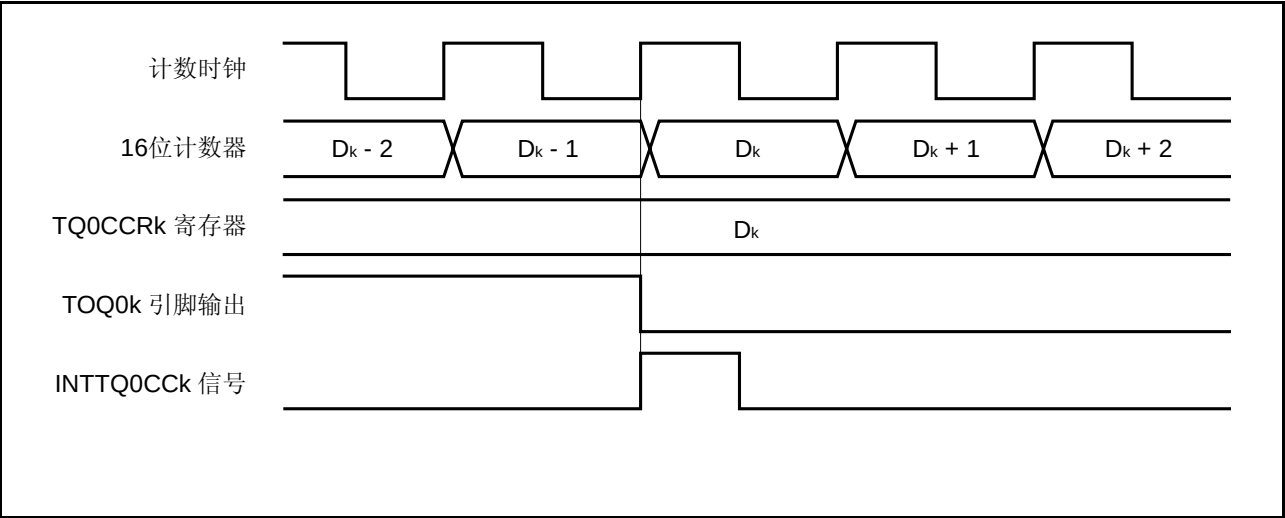
假设 $D_{00} > D_{01}$ 而且 $D_{k0} > D_{k1}$ ，当 TQ0CCR0 寄存器由 D_{00} 重写为 D_{01} ，TQ0CCRk 寄存器由 D_{k0} 重写为 D_{k1} 时，如果在 16 位计数器的计数值大于 D_{k1} 且小于 D_{k0} 时重写 TQ0CCRk 寄存器，并且在计数值大于 D_{01} 且小于 D_{00} 时重写 TQ0CCR0 寄存器，则一旦重写寄存器，就会在各设置值上反映出来，并立即与计数值进行比较。计数器向上计数到 FFFFH 后，从 0000H 开始重新向上计数。当计数值与 D_{k1} 匹配时，计数器会产生 INTTQ0CCk 信号并使 TOQ0k 引脚有效。当计数值与 D_{01} 匹配时，计数器产生 INTTQ0CC0 信号，使 TOQ0k 引脚失效，并停止计数。

因此，计数器输出脉冲的延迟时段或有效时段可能会与原先预计的单次触发脉冲有所不同。

备注 $k=1$ 至 3

(b) 比较匹配中断请求信号 (INTTQ0CCk) 的发生时序

单次触发脉冲输出模式下，比较匹配中断请求信号 (INTTQ0CCk) 的发生时序和其它情况下的 INTTQ0CCk 信号发生时序不同，当 16 位计数器的计数值与 TQ0CCRk 寄存器的值匹配时，产生 INTTQ0CCk 中断请求信号。



通常在 16 位计数器的当前计数值与 TQ0CCRk 寄存器的值匹配之后，在下一次计数时同步发生 INTTQ0CCk 信号。

但是，在单次触发脉冲输出模式下，INTTQ0CC1 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOQ0k 引脚输出信号变化的时序。

备注 k=1 至 3

8.5.5 PWM 输出模式 (TQ0MD2 至 TQ0MD0 位=100)

在 PWM 输出模式下，当 TQ0CTL0.TQ0CE 位被置为 1 时，PWM 波形从 TOQ01 至 TOQ03 引脚输出。

而且，由 TOQ00 引脚输出的脉冲，其半周期等于 PWM 的全周期。

图 8-24. PWM 输出模式的配置

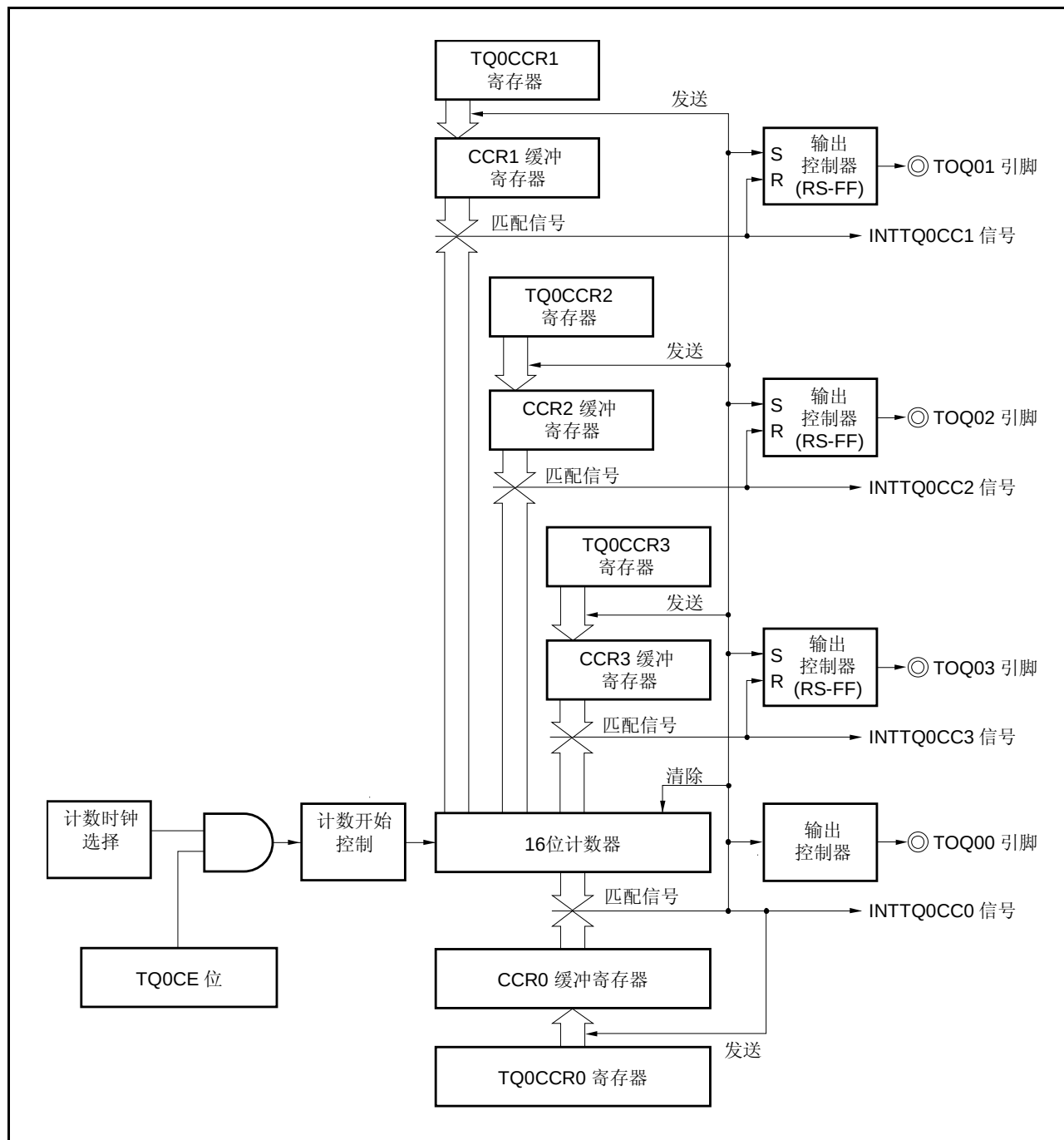
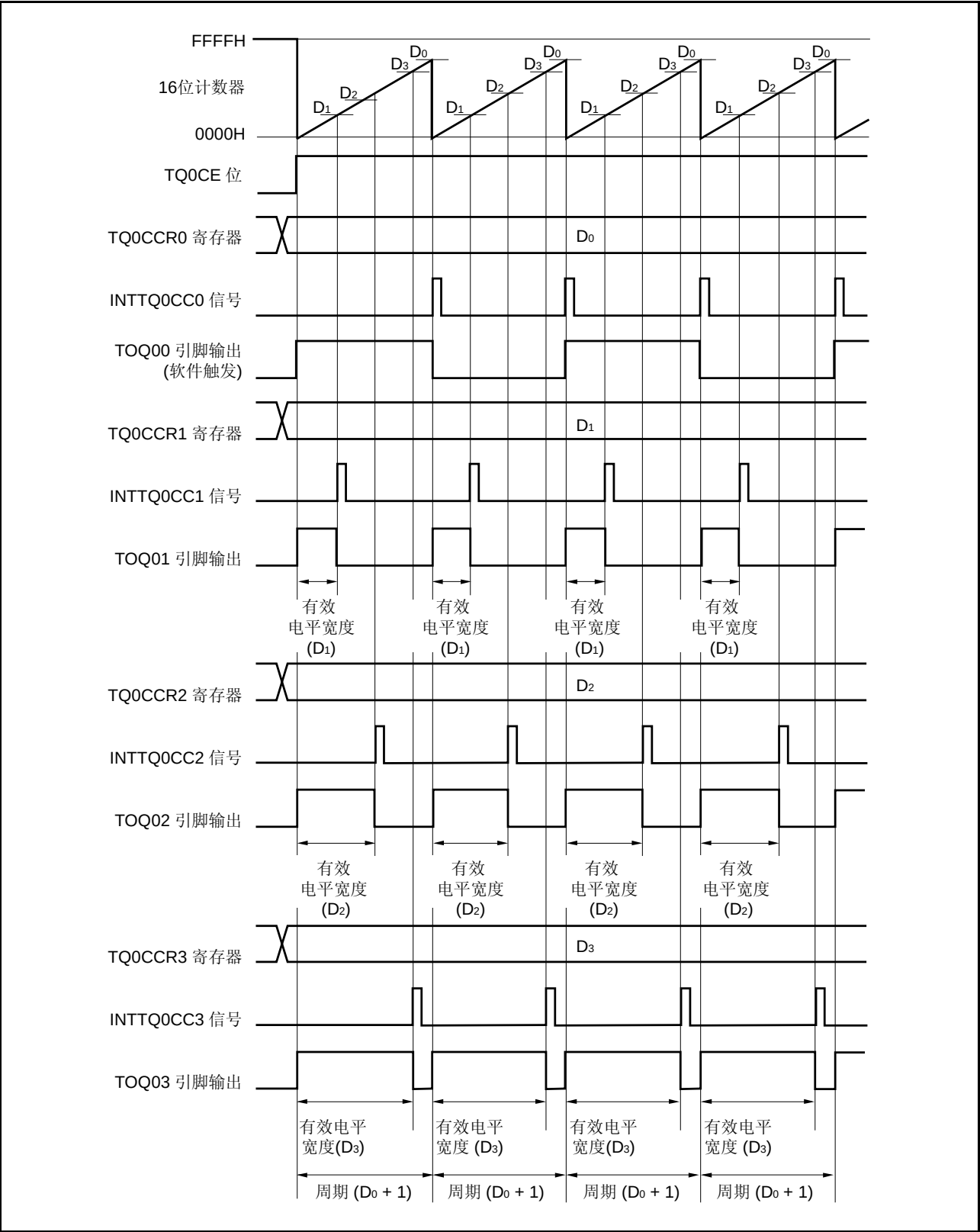


图 8-25. PWM 输出模式的基本时序



当 TQ0CE 位置为 1 时，16 位计数器从 FFFFH 清零为 0000H，计数开始，同时从 TOQ0k 引脚输出 PWM 波形。PWM 波形的有效电平宽度、周期、占空比系数的计算如下。

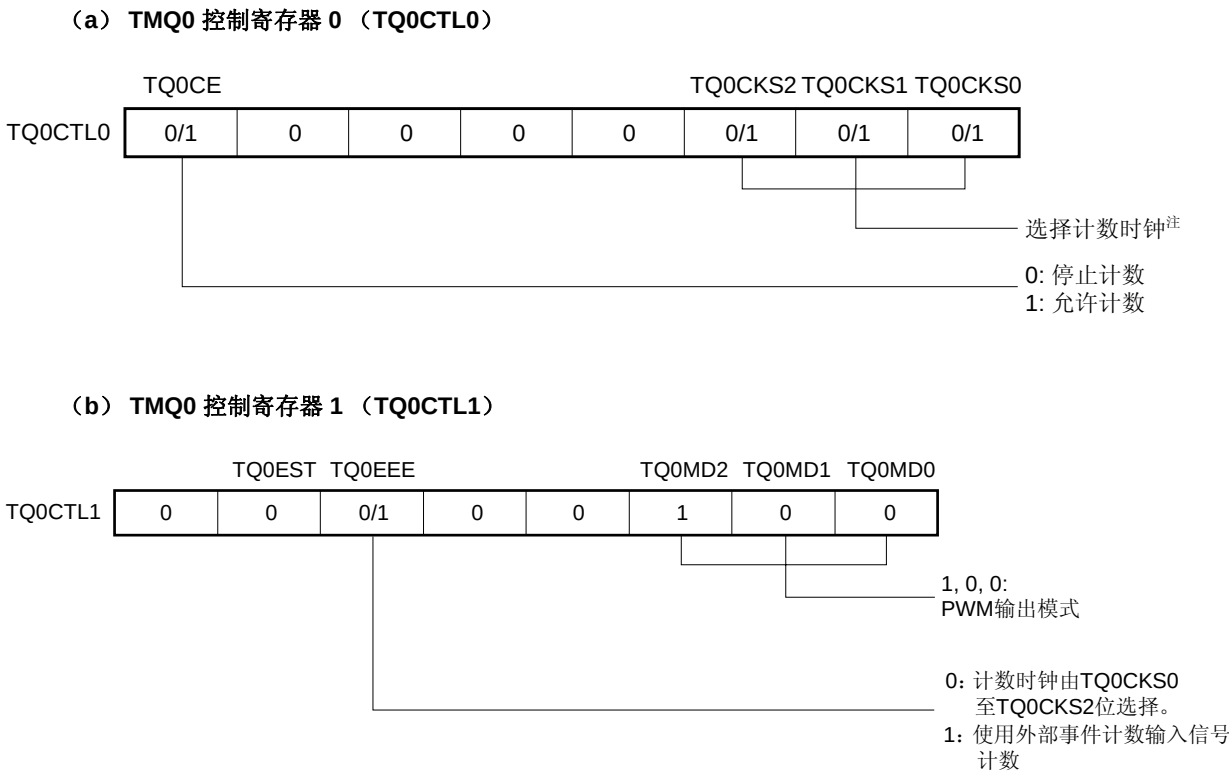
有效电平宽度 = (TQ0CCRk 寄存器的设定值) × 计数时钟周期
周期 = (TQ0CCR0 寄存器的设定值 + 1) × 计数时钟周期
占空比系数 = (TQ0CCRk 寄存器的设定值) / (TQ0CCR0 寄存器的设定值 + 1)

在计数操作时，可通过重写 TQ0CCRm 寄存器来改变 PWM 波形。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配时，16 位计数器清零为 0000H 时，新写入的值会发生作用。

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配之后，进行下一次计数时，会同步产生比较匹配中断请求信号 INTTQ0CC0，且 16 位计数器清零为 0000H。当 16 位计数器的计数值与 CCRk 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTQ0Cck。

备注 k=1 至 3
 m=0 至 3

图 8-26. PWM 输出模式时的寄存器设置 (1/3)



注 当 TQ0CTL1.TQ0EEE 位=1 时，设置无效。

图 8-26. PWM 输出模式时的寄存器设置 (2/3)

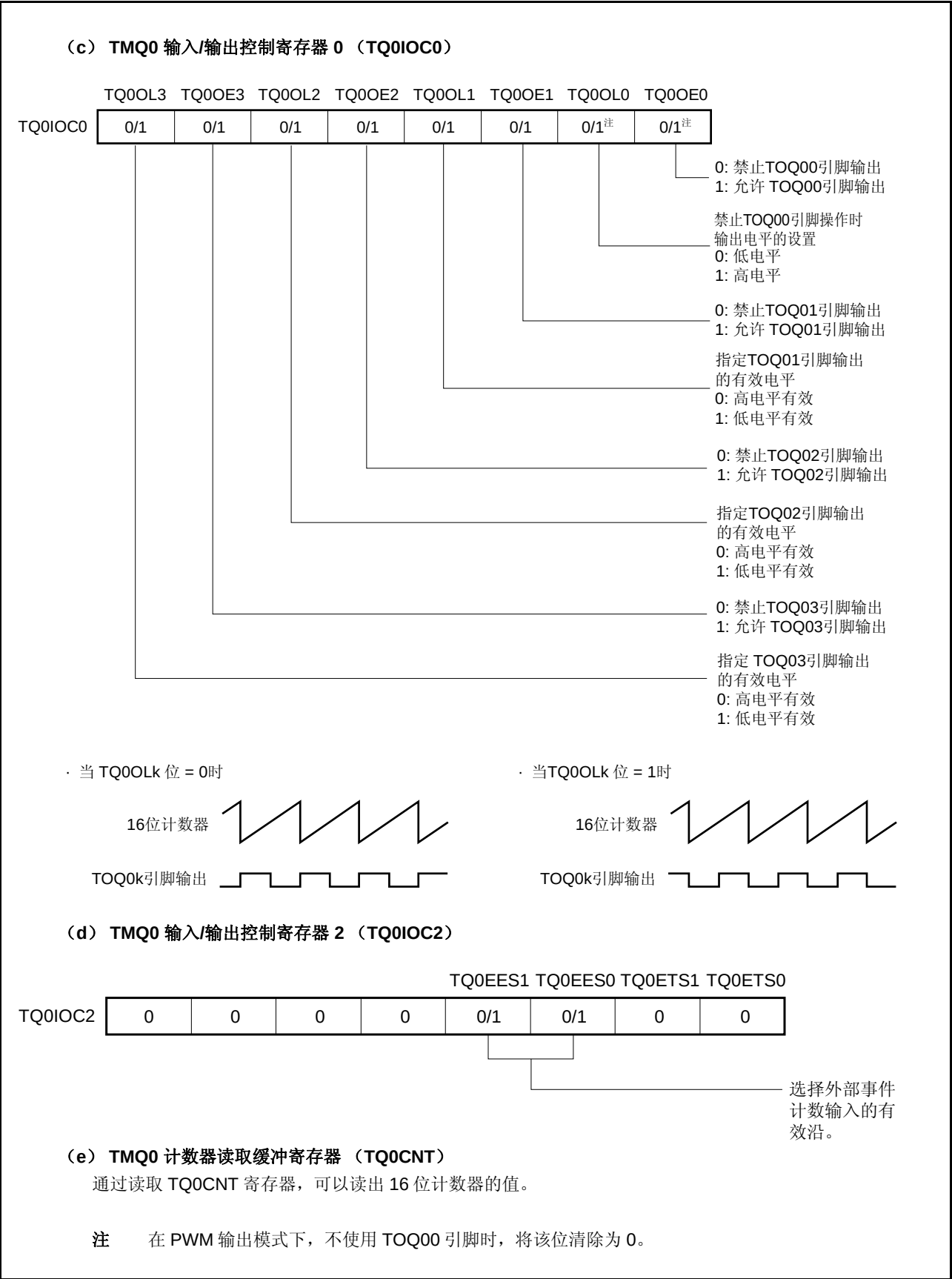


图 8-26. PWM 输出模式时的寄存器设置 (3/3)

(f) TMQ0 捕获/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 TQ0CCR1 寄存器的值设置为 D_k ，则 PWM 波形的周期和有效电平的计算如下：

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. 在 PWM 输出模式下，不使用 TMQ0 输入/输出控制寄存器 1 (TQ0IOC1) 和 TMQ0 选项寄存器 0 (TQ0OPT0)。
 2. 通过写入 TMQ0 捕获/比较寄存器 1 (TQ0CCR1)，可以更新 TMQ0 捕获/比较寄存器 (TQ0CCR2) 和 TMQ0 捕获/比较寄存器 3 (TQ0CCR3)。

(1) PWM 输出模式下的操作流程

图 8-27. PWM 输出模式的软件处理流程 (1/2)

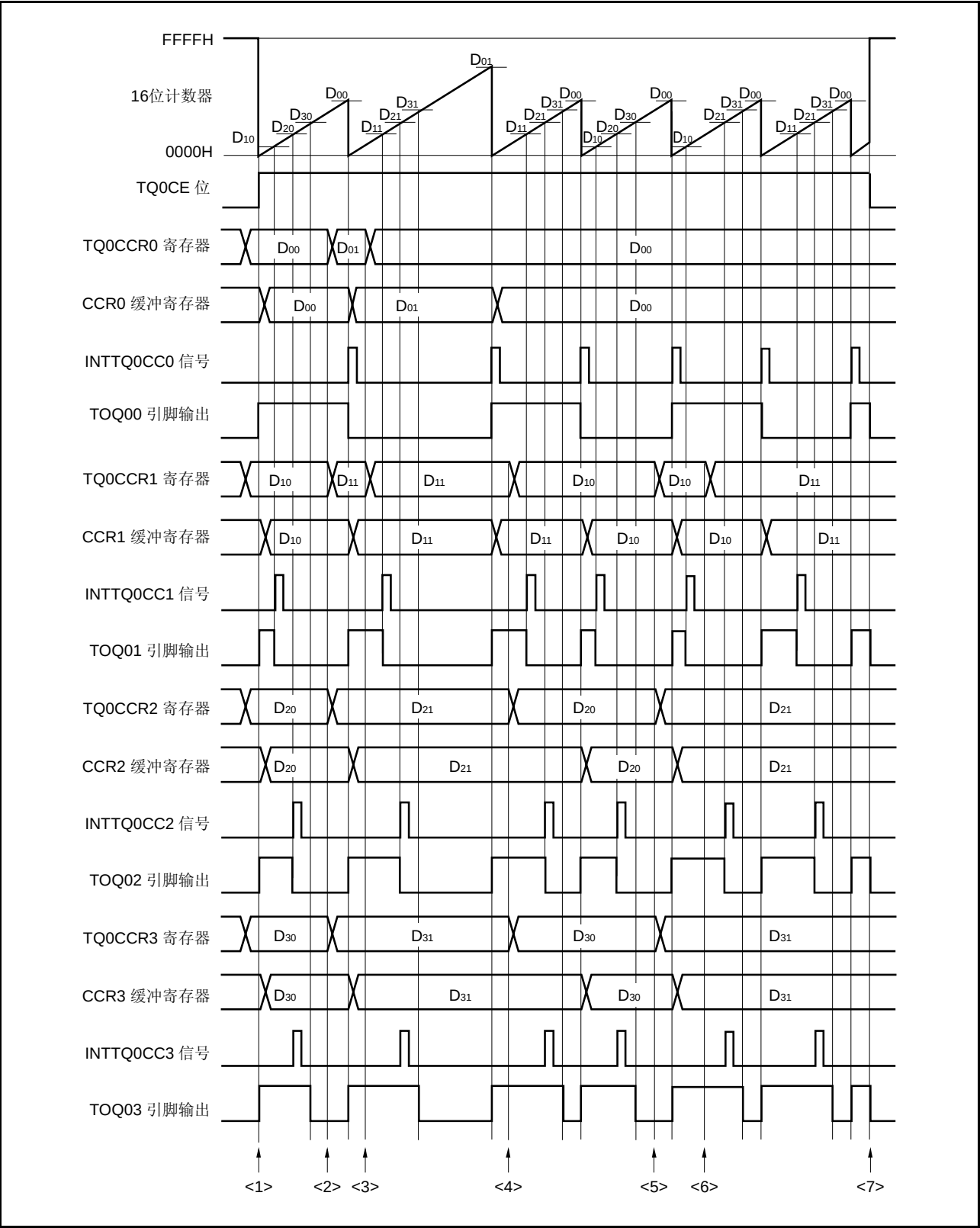
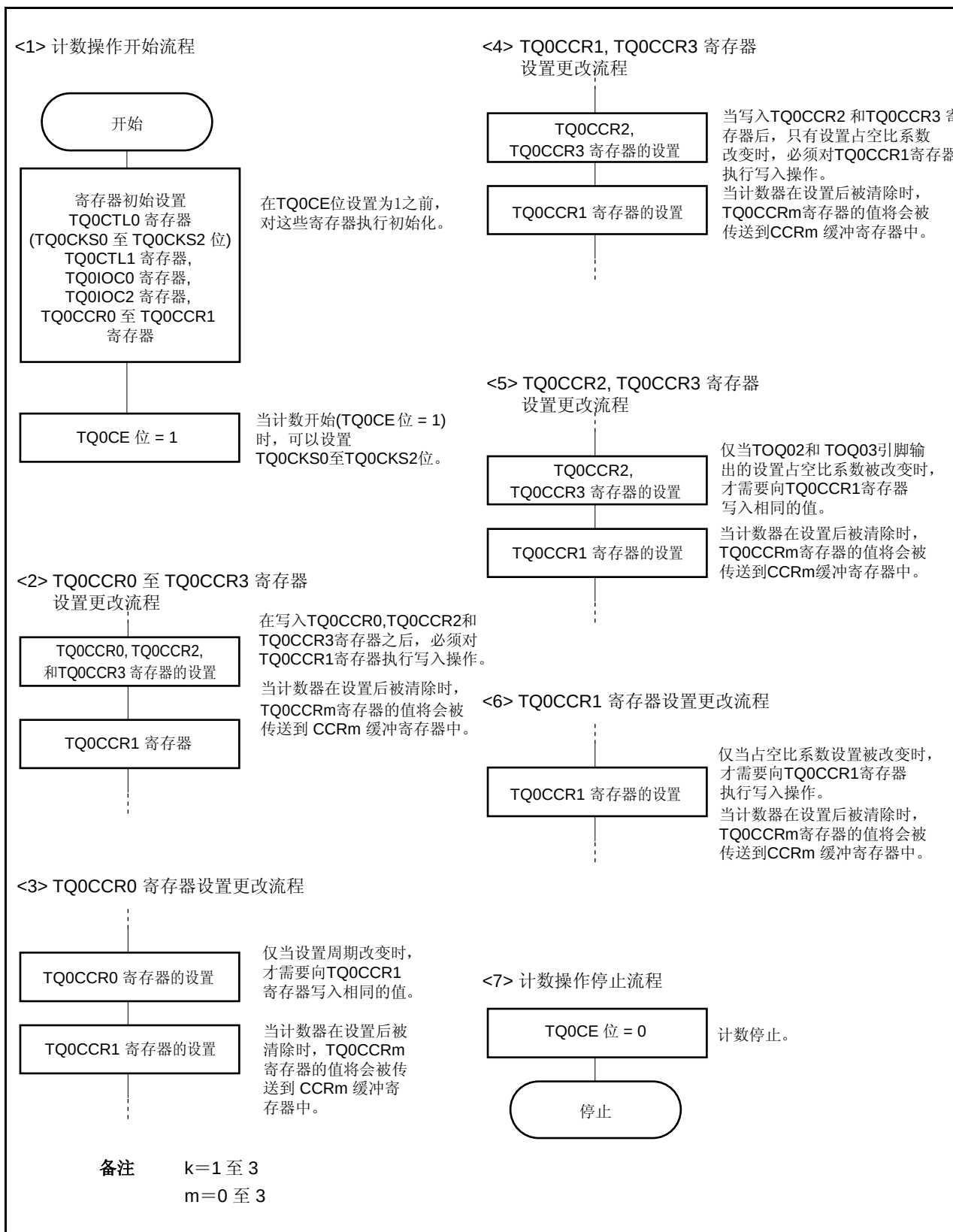


图 8-27. PWM 输出模式时的软件处理流程 (2/2)

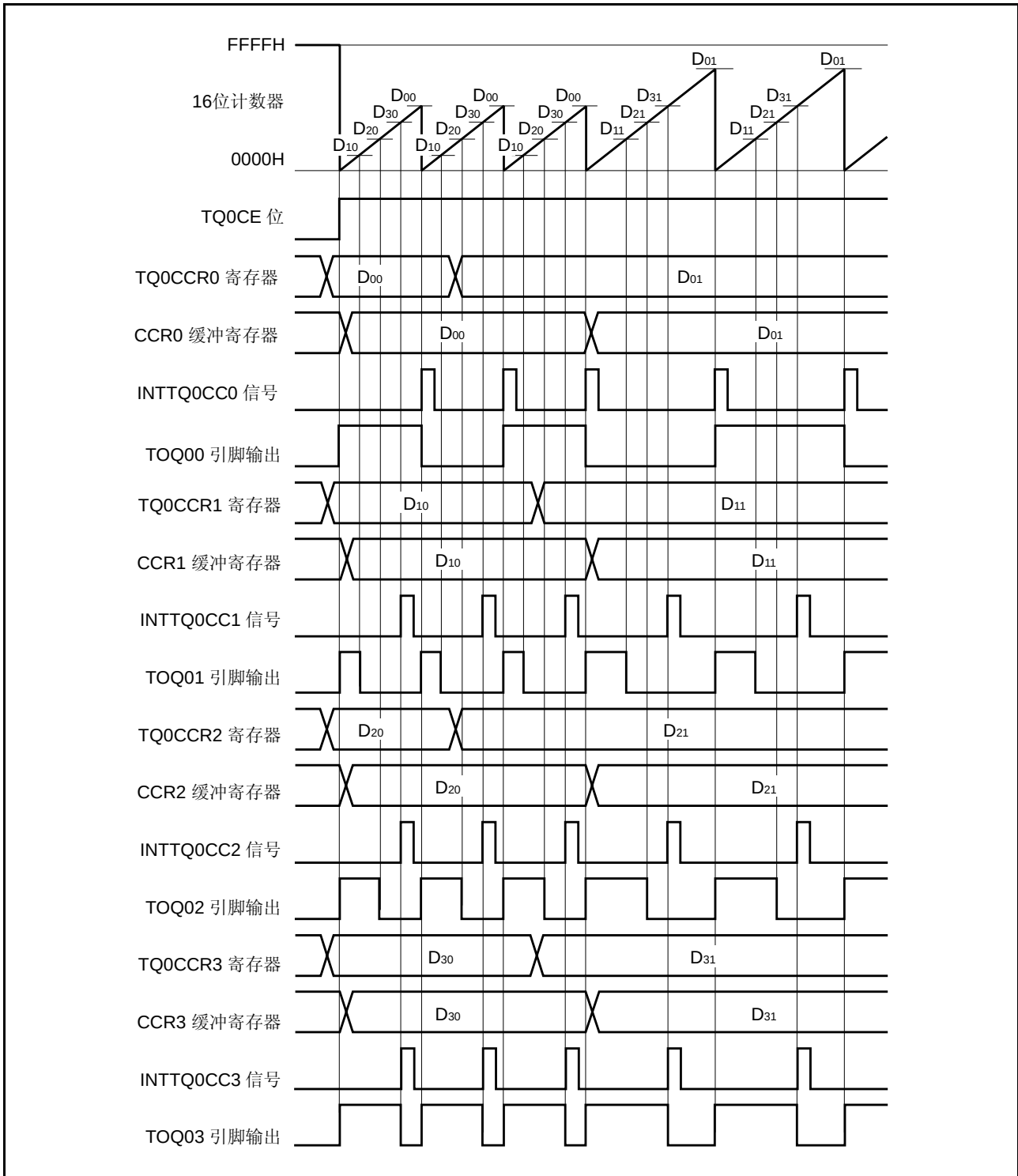


(2) PWM 输出模式的操作时序

(a) 操作过程中改变脉冲宽度

计数器操作过程中，若要改变 PWM 波形，应重新写入 TQ0CCR1 寄存器。

当检测到 INTTQ0CC1 信号以后，在写入 TQ0CCR1 寄存器之后重写 TQ0CCRk 寄存器。



为了将数据从 TQ0CCRm 寄存器传送到 CCRm 缓冲寄存器，必须对 TQ0CCR1 寄存器进行写入操作。

此时如果 PWM 波形的周期和有效电平都要改变，应先将周期设定到 TQ0CCR0 寄存器，再将有效电平宽度设定到 TQ0CCR2 和 TQ0CCR3 寄存器，然后将有效电平设定到 TQ0CCR1 寄存器。

如果只改变 PWM 波形的有效电平宽度（占空比系数），需要先向 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。

如果只改变 TOQ01 引脚输出的 PWM 波形的有效电平宽度（占空比系数），仅需要设置 TQ0CCR1 寄存器。

如果只改变 TOQ02 和 TOQ03 引脚输出的 PWM 波形的有效电平宽度（占空比系数），应先向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平宽度，然后将相同的值写入 TQ0CCR1 寄存器。

当数据写入 TQ0CCR1 寄存器之后，在对 16 位计数器清零的同时，写入到 TQ0CCRm 寄存器的值会被传送到 CCRm 缓冲寄存器，并作为与 16 位计数器进行比较的值。

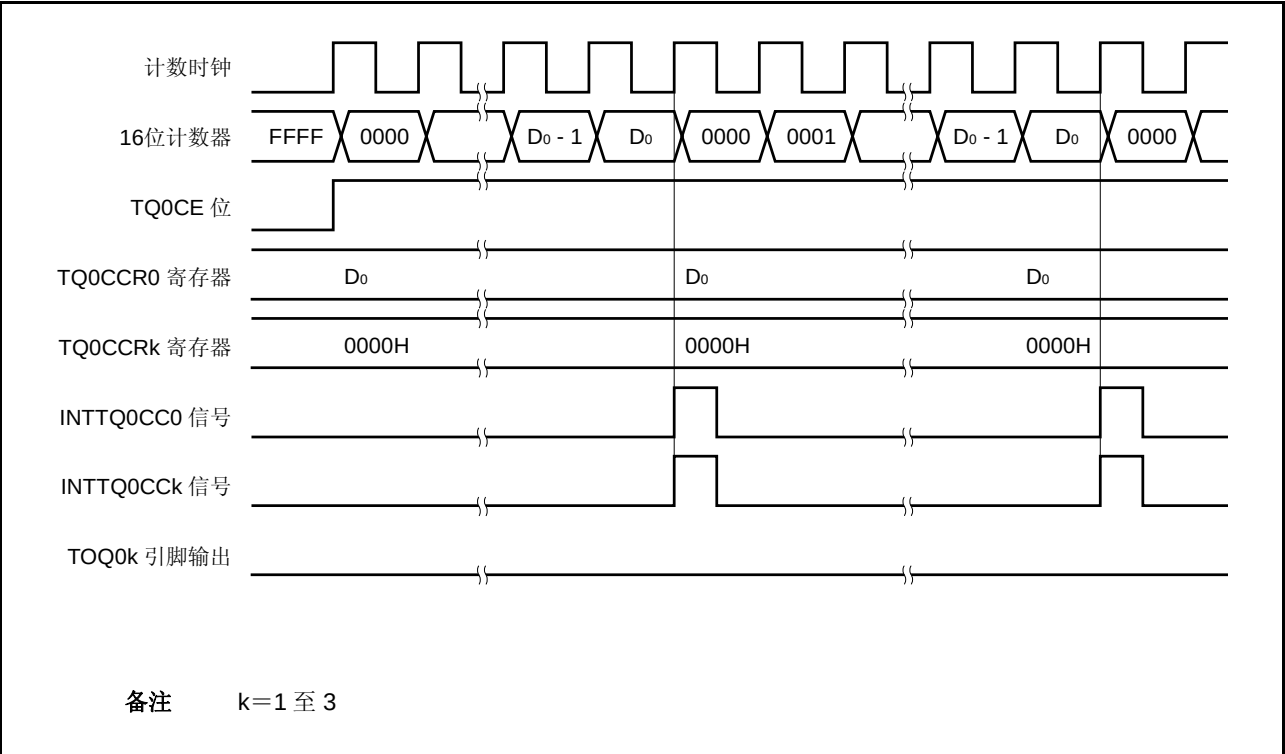
如果只改变 PWM 波形的周期，应先把周期设置到 TQ0CCR0 寄存器，然后将相同的值写入 TQ0CCR1 寄存器。

在写入 TQ0CCR1 寄存器一次之后，要再次写入 TQ0CCR0 至 TQ0CCR3 寄存器，应该在 INTTQ0CC0 信号产生之后进行。否则，CCRM 缓冲寄存器的值可能不确定，因为把数据从 TQ0CCRm 寄存器传送到 CCRm 缓冲寄存器的时序会和写入 TQ0CCRm 寄存器的时序发生冲突。

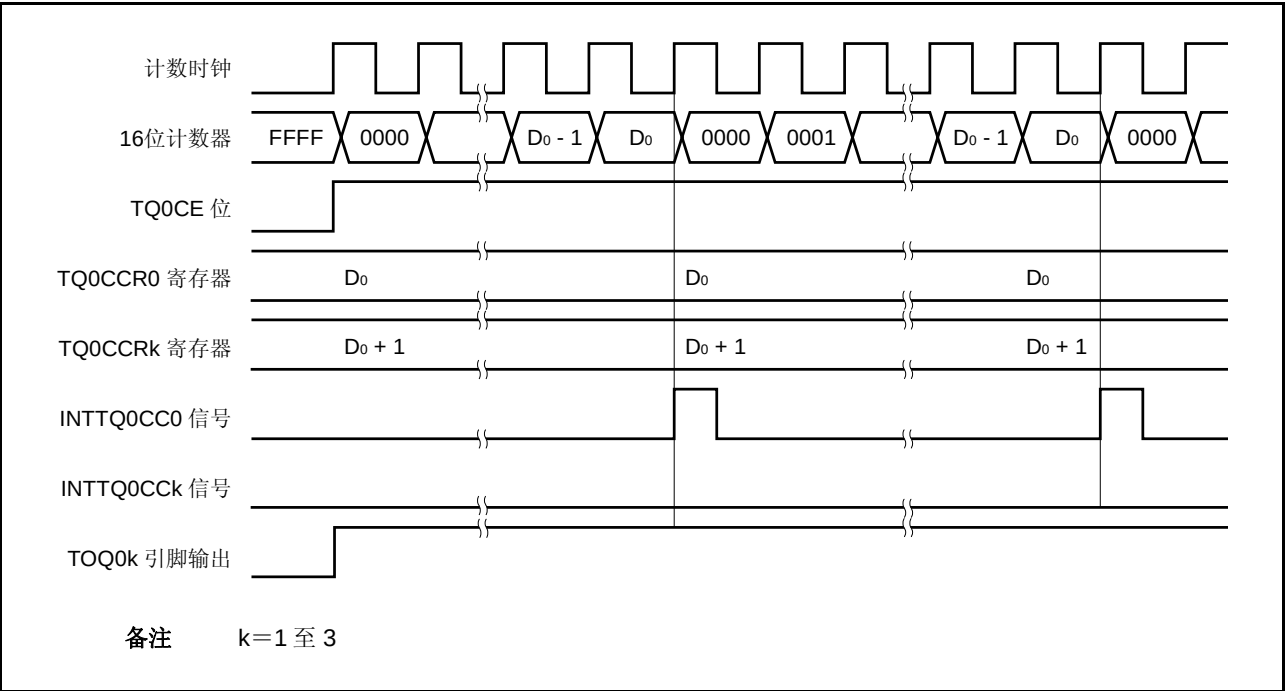
备注 m=0 至 3

(b) PWM 波形的 0%和 100% 输出

为了输出 0%的波形，需要将 TQ0CCRk 寄存器设置为 0000H。如果 TQ0CCR0 寄存器的设定值为 FFFFH，则会在每个周期产生 INTTQ0CCK 中断请求信号。

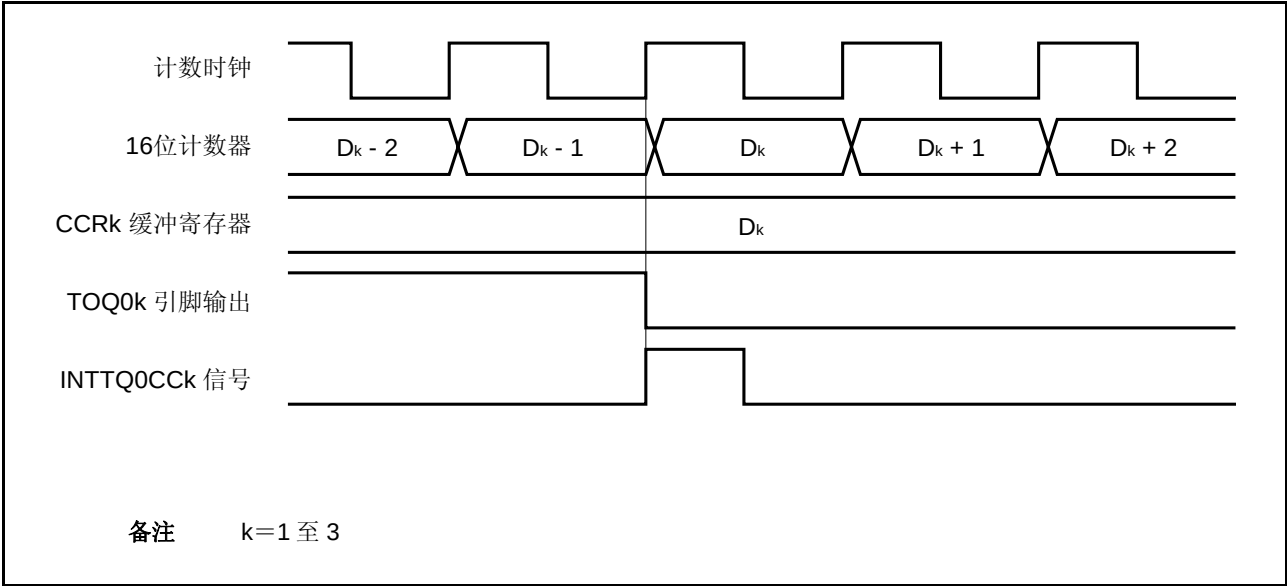


为了输出 100%的波形，需要将 TQ0CCRk 寄存器的值设为：（TQ0CCR0 寄存器的设定值 +1）。如果 TQ0CCR0 寄存器的设定值为 FFFFH，则不会产生 100%波形输出。



(c) 比较匹配中断请求信号 (INTTQ0CCk) 的发生时序

在 PWM 输出模式下，比较匹配中断请求信号 (INTTQ0CCk) 的发生时序与其它情况下 INTTQ0CCk 信号的发生时序有所不同；当 16 位计数器的计数值与 TQ0CCRk 寄存器的值匹配时，产生 INTTQ0CCk 信号。



通常在 16 位计数器的计数值与 TQ0CCRk 寄存器的值匹配之后，在下一次计数时同步发生 INTTQ0CCk 信号。

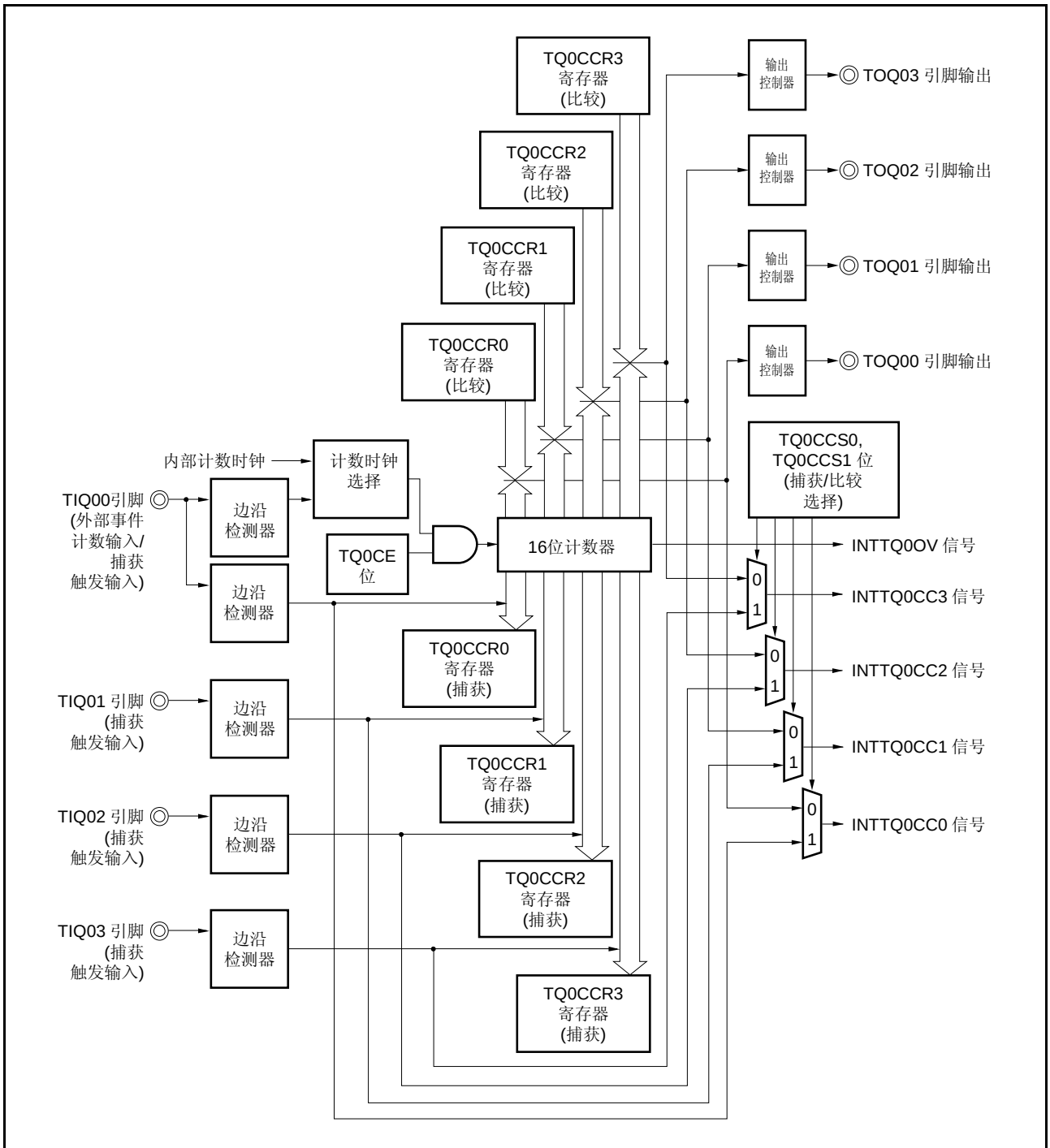
但是，在 PWM 输出模式下，INTTP0CC1 信号会提早一个计数时钟周期产生。该时序的变化是为了匹配 TOQ0k 引脚输出信号变化的时序。

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下, 当 TQ0CTL0.TQ0CE 位置为 1 时, 16 位定时器/事件计数器 Q 开始计数。此时, 根据 TQ0OPT0.TQ0CCS0 和 TQ0OPT0.TQ0CCS1 位的设置, TQ0CCRm 寄存器可以用作比较寄存器或捕获寄存器。

备注 m=0 至 3

图 8-28. 自由运行定时器模式的配置

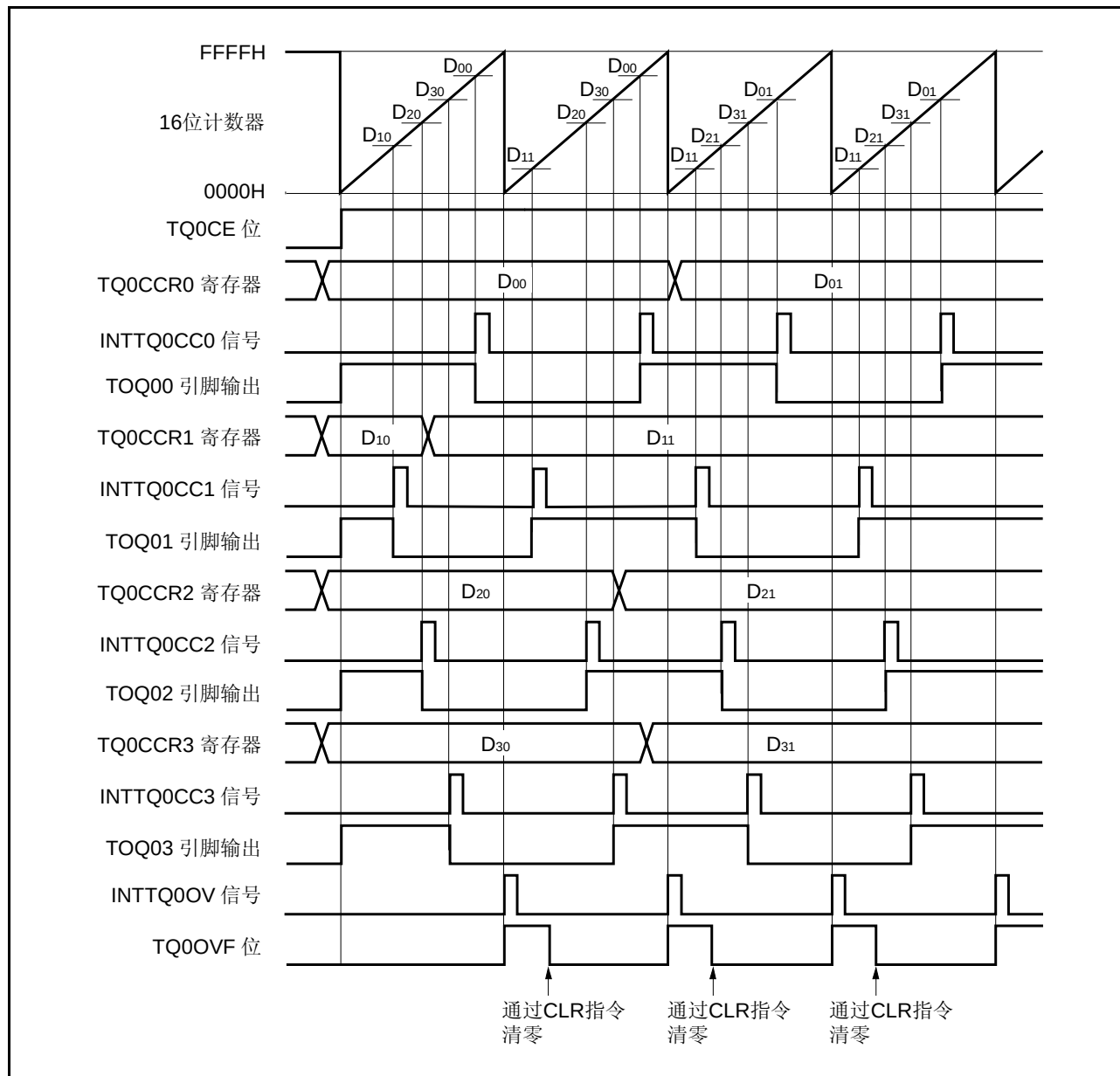


当 TQ0CE 位被置为 1 时, 16 位定时器/事件计数器 Q 开始计数, TOQ00 至 TOQ03 引脚的输出信号反相。此后当 16 位计数器的计数值与 TQ0CCRm 寄存器的设定值匹配时, 产生一个比较匹配中断请求信号 (INTTQ0CCm), 而且 TOQ0m 引脚的输出信号反相。

16 位计数器继续对计数时钟同步计数。当它计数到 FFFFH 时, 会在下一个计数时钟时产生一个溢出中断请求信号 (INTTQ0OV), 同时计数器清零为 0000H, 然后继续进行计数。此时, 溢出标志 (TQ0OPT0.TQ0OVF 位) 被置为 1, 通过软件运行 CLR 指令可以把溢出标志位清 0。

当计数器工作时, 可以重写 TQ0CCRm 寄存器。如果进行重写, 则新写入的值会立刻反映出来, 并与计数值进行比较。

图 8-29. 自由运行定时器模式下的基本时序 (比较功能)



当 TQ0CE 位被置为 1 时，16 位计数器开始计数。当检测到 TIQ0m 引脚输入信号的有效边沿时，16 位计数器的计数值存储在 TQ0CCRm 寄存器中，并产生一个捕获中断请求信号 (INTTQ0CCm)。

16 位计数器继续对计数时钟同步计数。当它计数到 FFFFH 时，会在下一个计数时钟时产生一个溢出中断请求信号 (INTTQ0OV)，同时计数器清零为 0000H，然后继续进行计数。此时，溢出标志 (TQ0OVF 位) 被置为 1，通过软件运行 CLR 指令可以把溢出标志位清 0。

图 8-30. 自由运行定时器模式下的基本时序（捕获功能）

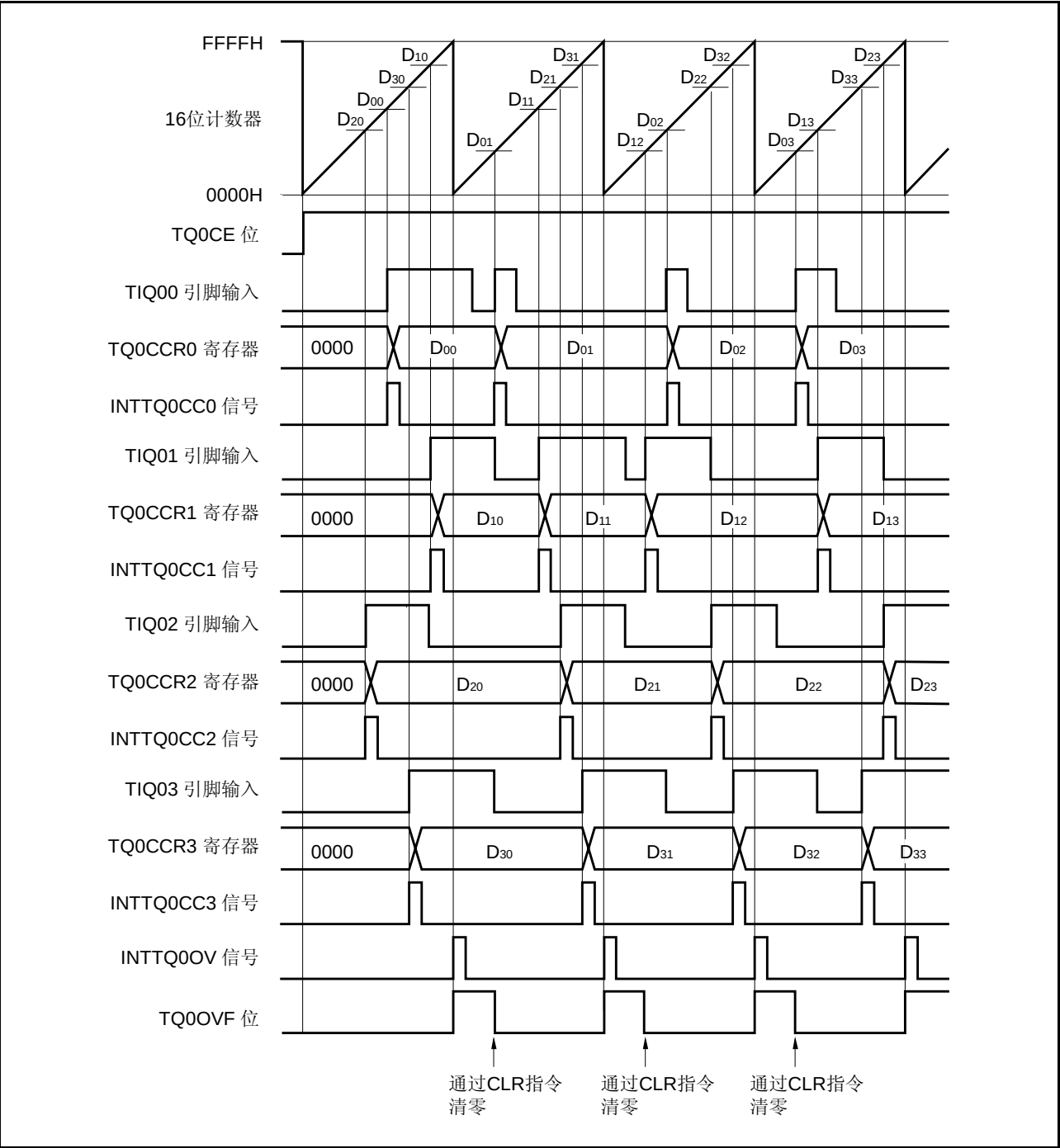


图 8-31. 自由运行定时器模式下的寄存器设置 (1/3)

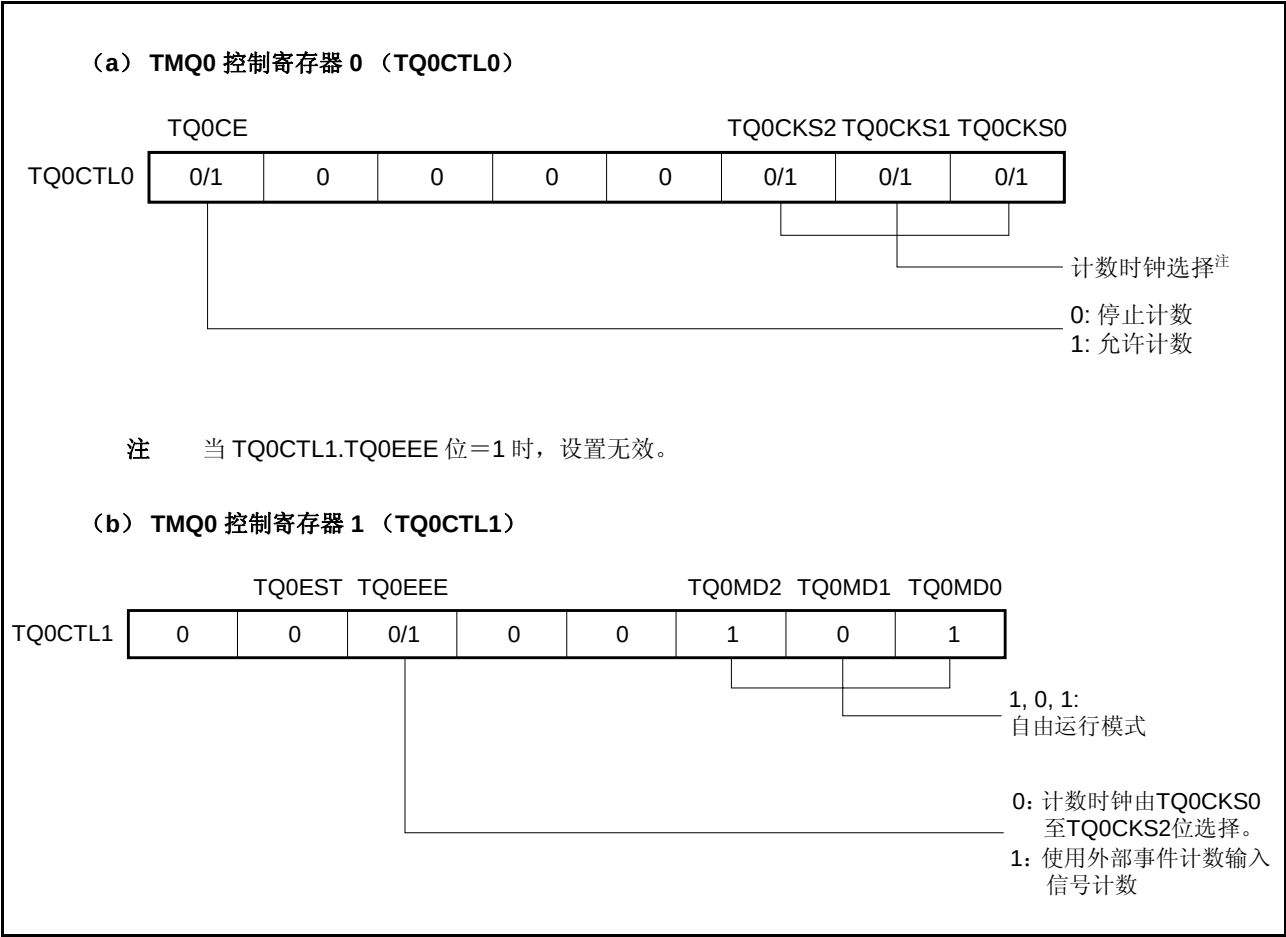


图 8-31. 自由运行定时器模式下的寄存器设置 (2/3)

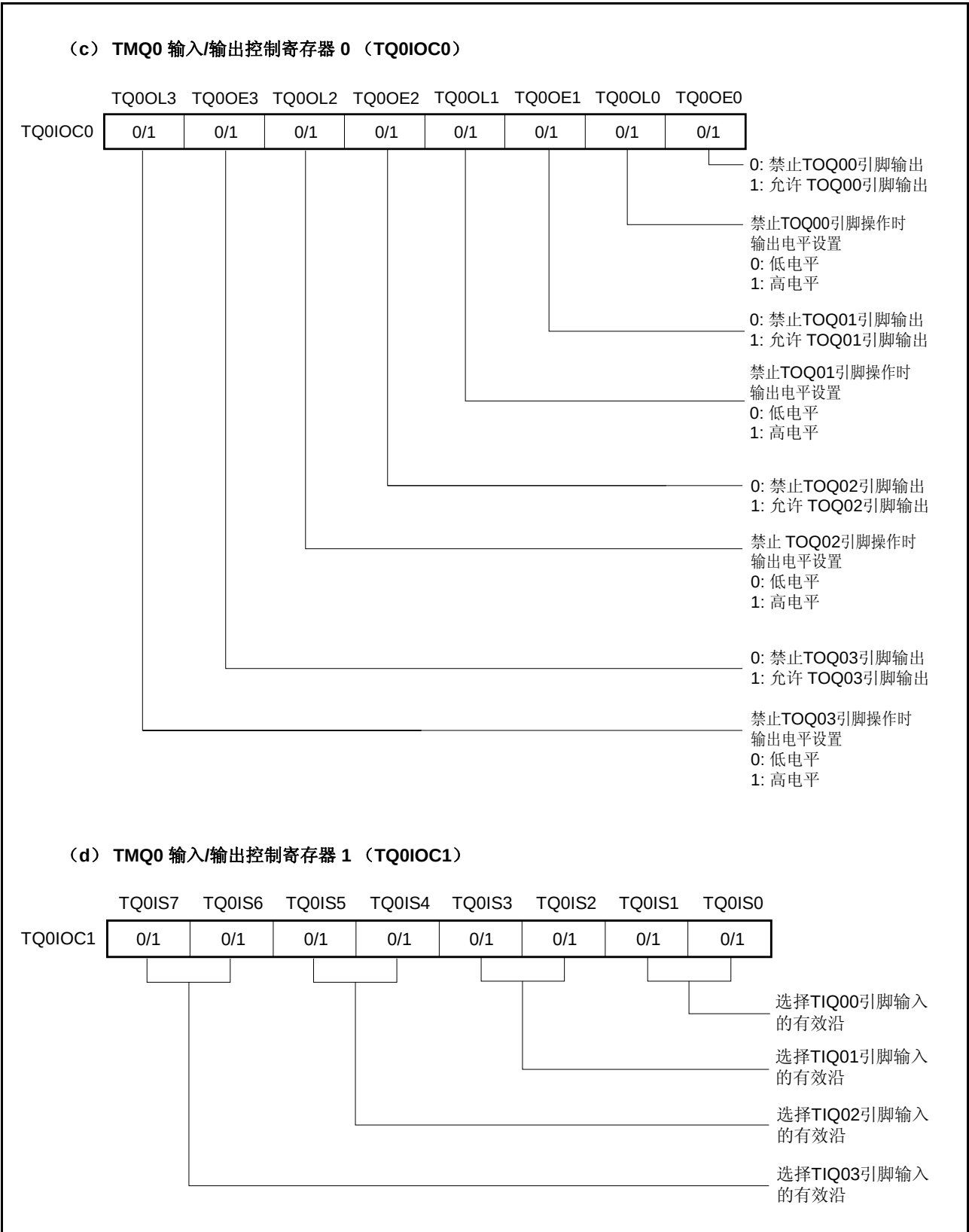
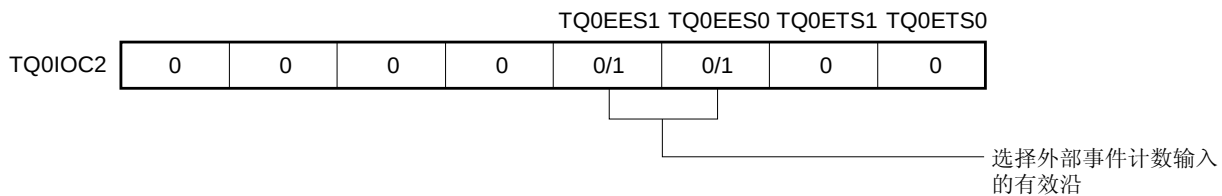
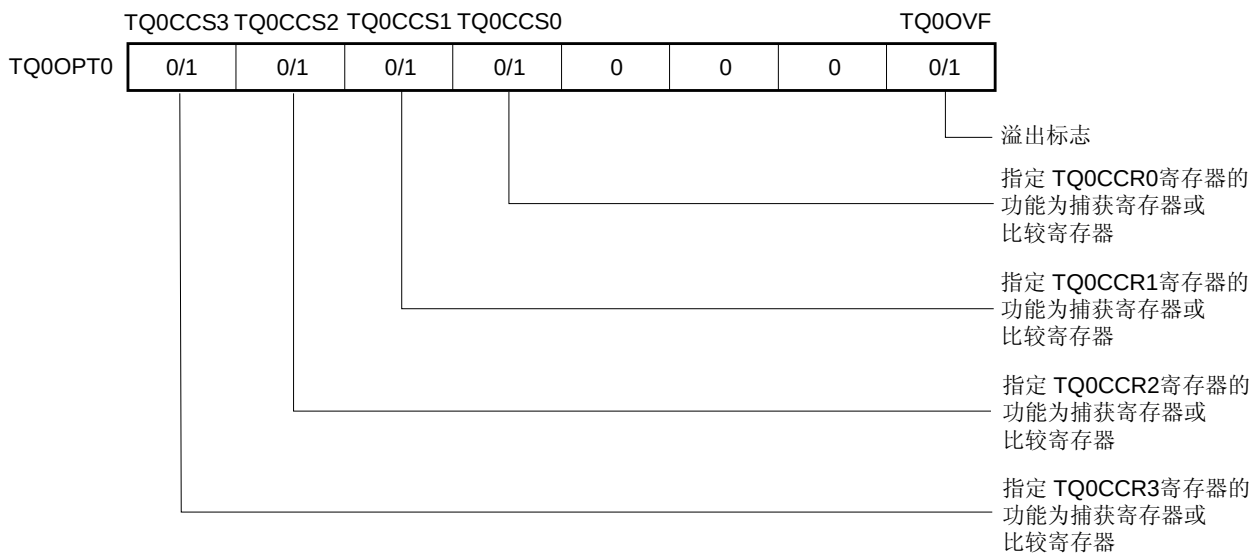


图 8-31. 自由运行定时器模式下的寄存器设置 (3/3)

(e) TMQ0 输入/输出控制寄存器 2 (TQ0IOC2)



(f) TMQ0 选项寄存器 0 (TQ0OPT0)



(g) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

通过读取 TQ0CNT 寄存器，可以读出 16 位计数器的值。

(h) TMQ0 捕获/比较寄存器 0 至 3 (TQ0CCR0 至 TQ0CCR3)

根据 TQ0OPT0.TQ0CCSm 位的设置，这些寄存器可以用作捕获寄存器或比较寄存器。

当用作捕获寄存器时，在检测到 TIQ0m 引脚输入信号的有效边沿时，存储 16 位计数器的当前计数值。

当用作比较寄存器时，而且 TQ0CCRm 寄存器的值被设置为 D_m 时，则当计数器达到 (D_m + 1) 时会产生 INTTQ0CCm 中断信号，同时，TOQ0m 引脚的输出信号反相。

备注 m=0 至 3

(1) 自由运行定时器模式时的操作流程

(a) 当捕获/比较寄存器用作比较寄存器时

图 8-32. 自由运行定时器模式的软件处理流程（比较功能）（1/2）

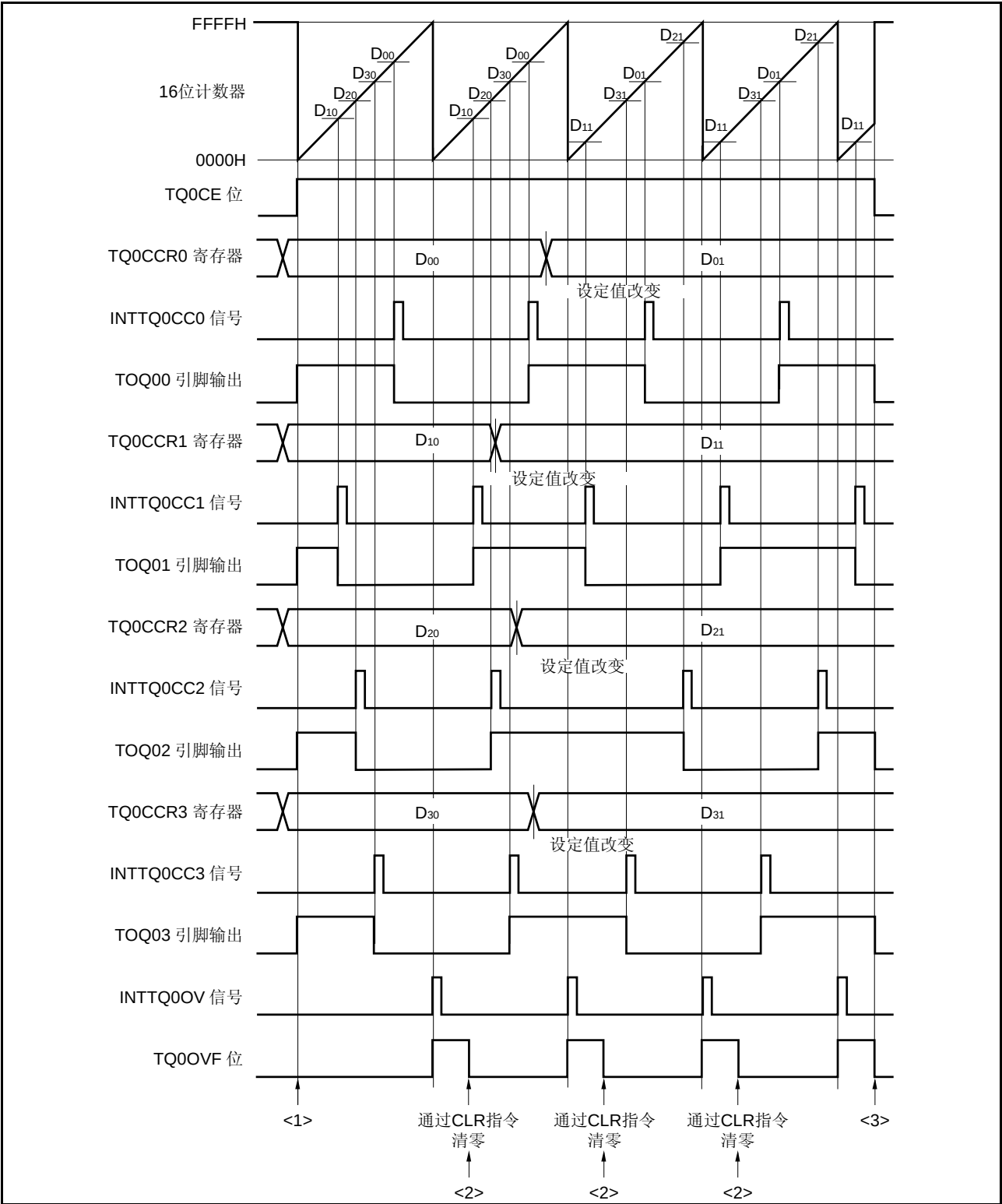
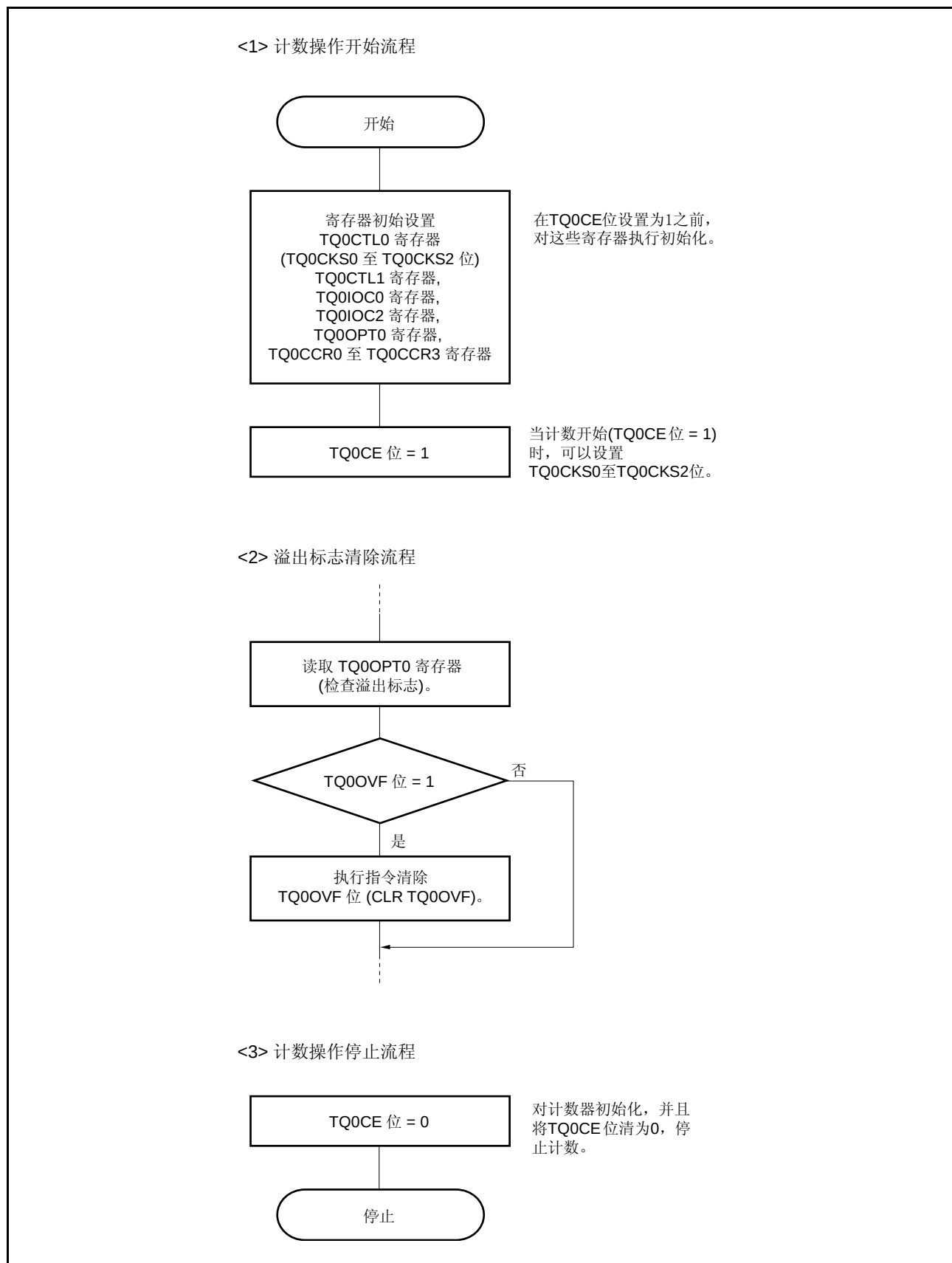


图 8-32. 自由运行定时器模式的软件处理流程（比较功能）（2/2）



(b) 当捕获/比较寄存器用作捕获寄存器时

图 8-33. 自由运行定时器模式的软件处理流程（捕获功能）（1/2）

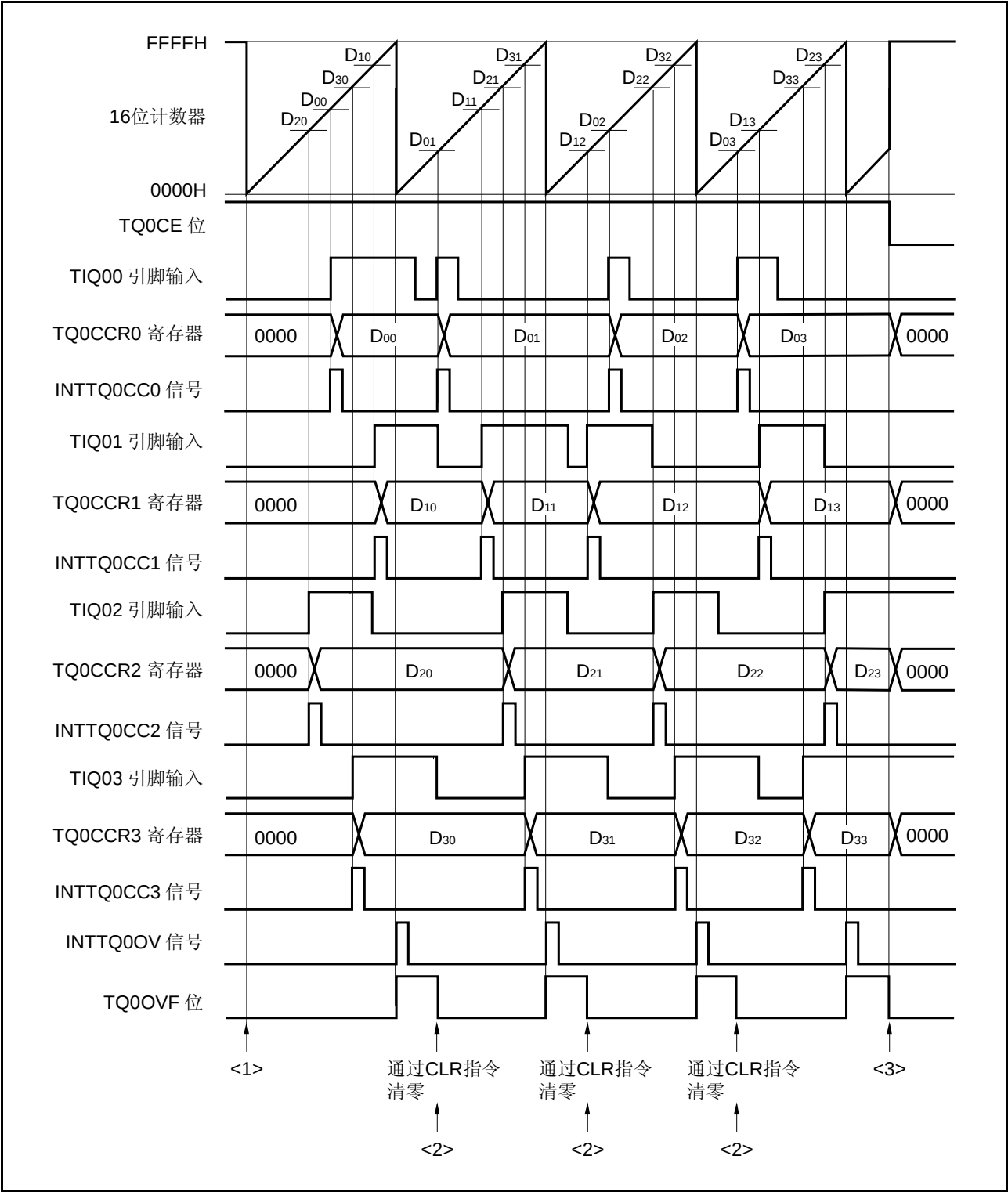
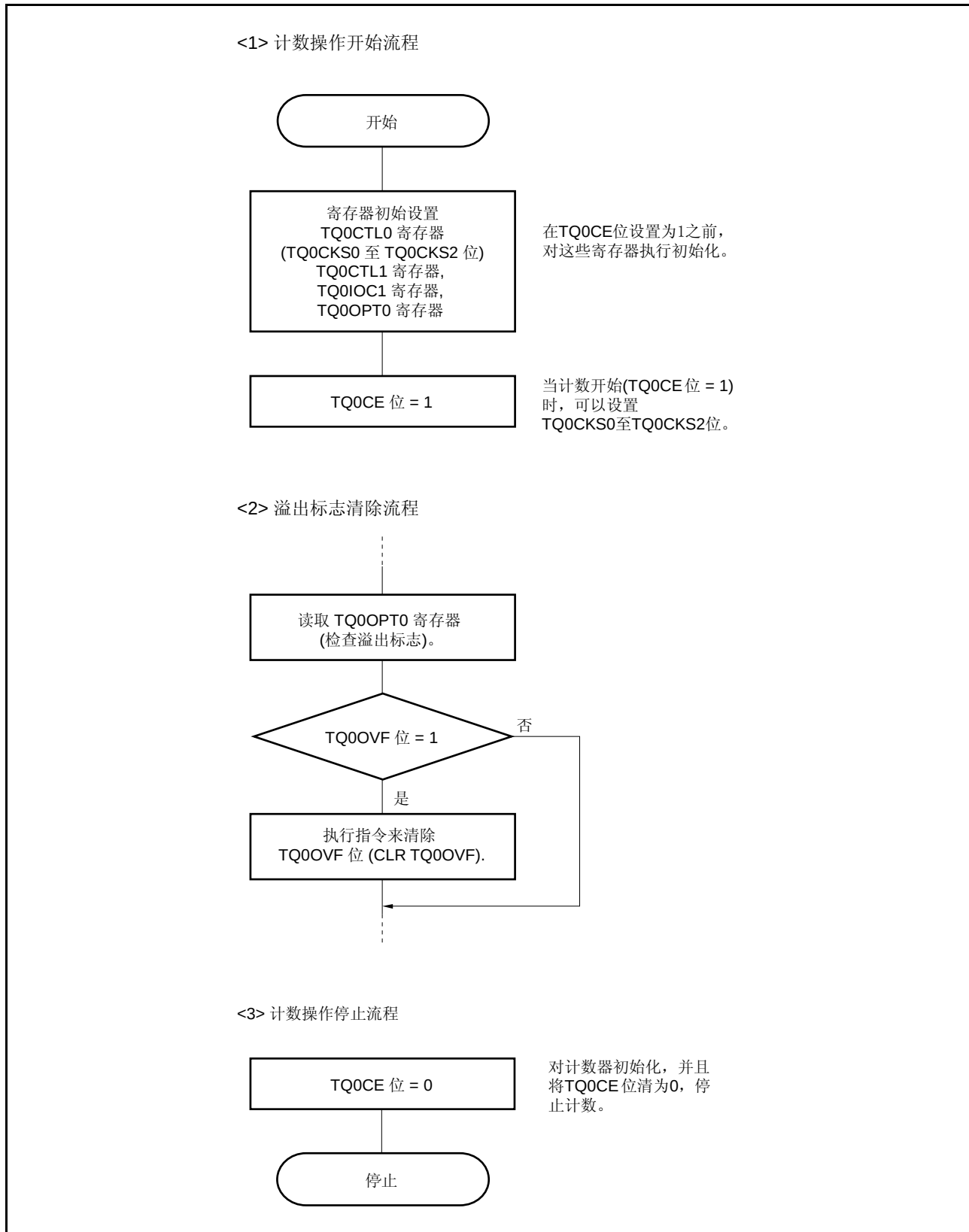


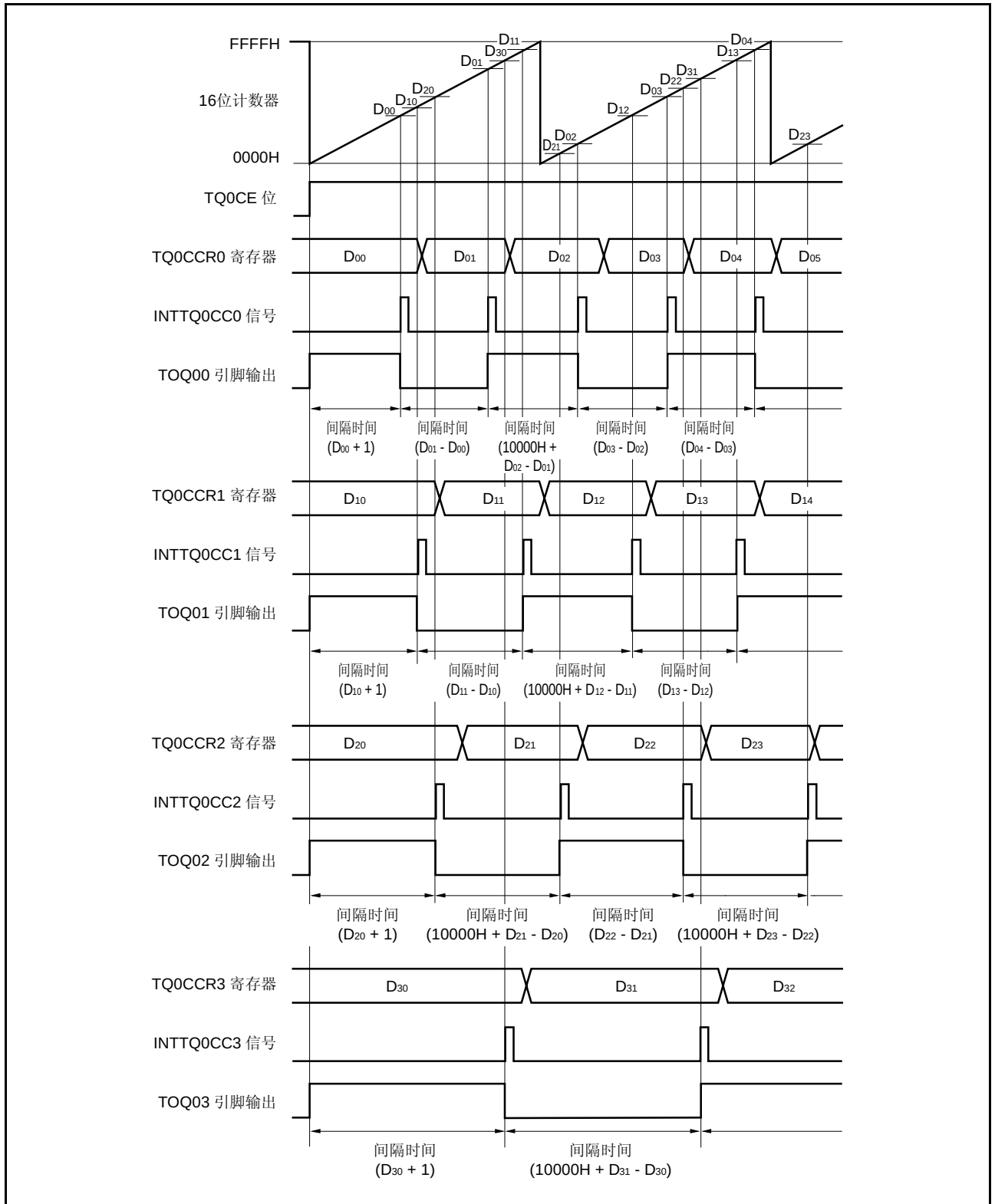
图 8-33. 自由运行定时器模式时的软件处理流程（捕获功能）（2/2）



(2) 自由运行定时器模式时的操作时序

(a) 比较寄存器的间隔定时操作

当 16 位定时器/事件计数器 Q 用作间隔定时器，TQ0CCRm 寄存器用作比较寄存器时，在每次检测到 INTTQ0CCm 信号后，都需要通过软件设置一个比较值以产生下一个中断请求信号。



在自由运行定时器模式下进行时间间隔操作时，一个通道可以设置两个时间间隔。

要进行时间间隔操作，当检测到 INTTQ0CCm 信号后，必须在中断服务程序中重置相应的 TQ0CCRm 寄存器的值。

重置 TQ0CCRm 寄存器的设置值可由下式计算，其中“Dm”是间隔时间：

比较寄存器默认值： $D_m - 1$

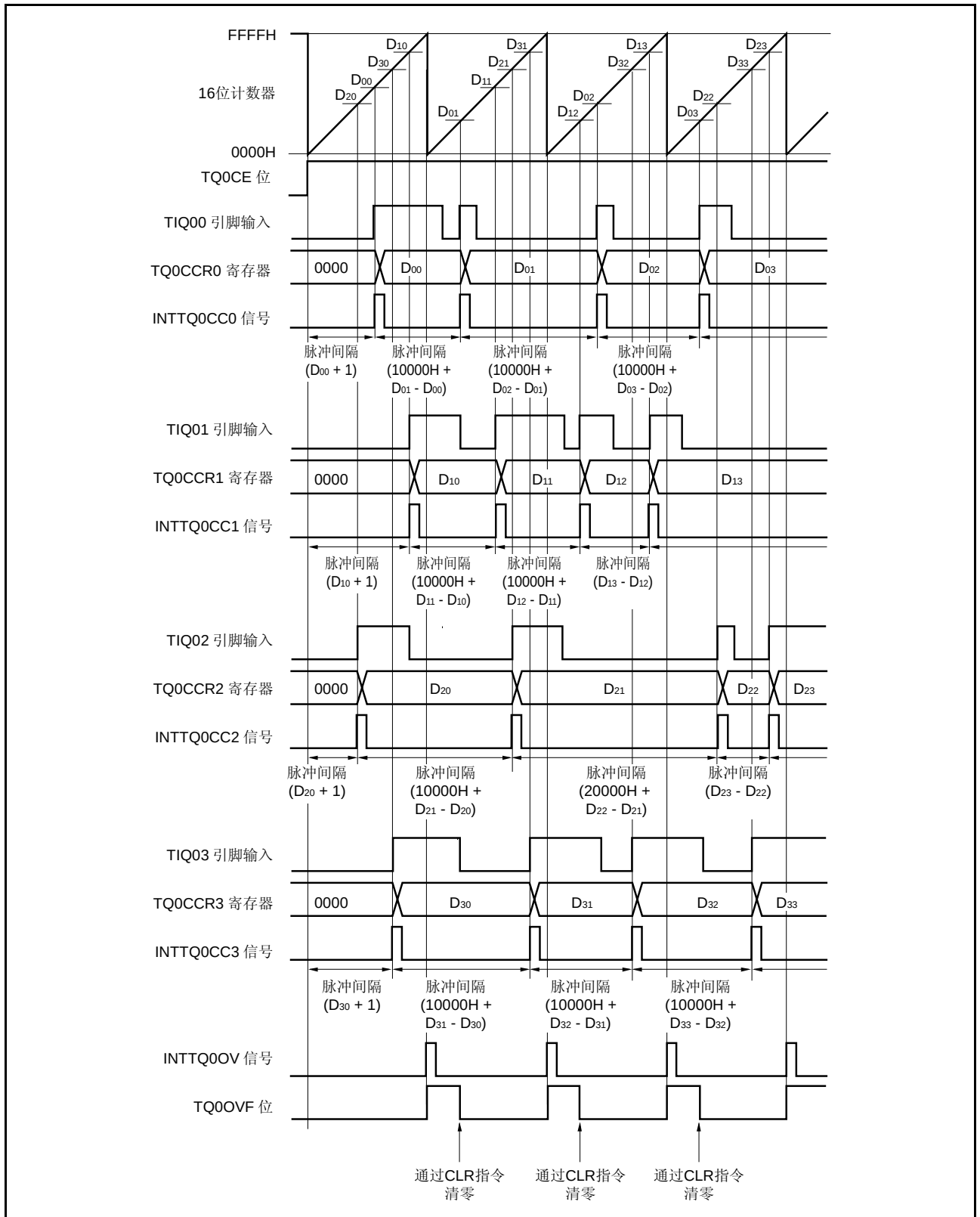
比较寄存器的第二次及其以后的设置值：上次设置值 + D_m

（若计算结果大于 FFFFH，则从结果中减去 10000H，并将此值设置给寄存器。）

备注 m=0 至 3

(b) 使用捕获寄存器进行脉冲宽度测量

在用 TQ0CCRm 寄存器作为捕获寄存器进行脉宽测量时，每次检测到 INTTQ0CCm 信号时，都需要通过软件读取捕获寄存器并计算间隔时间。



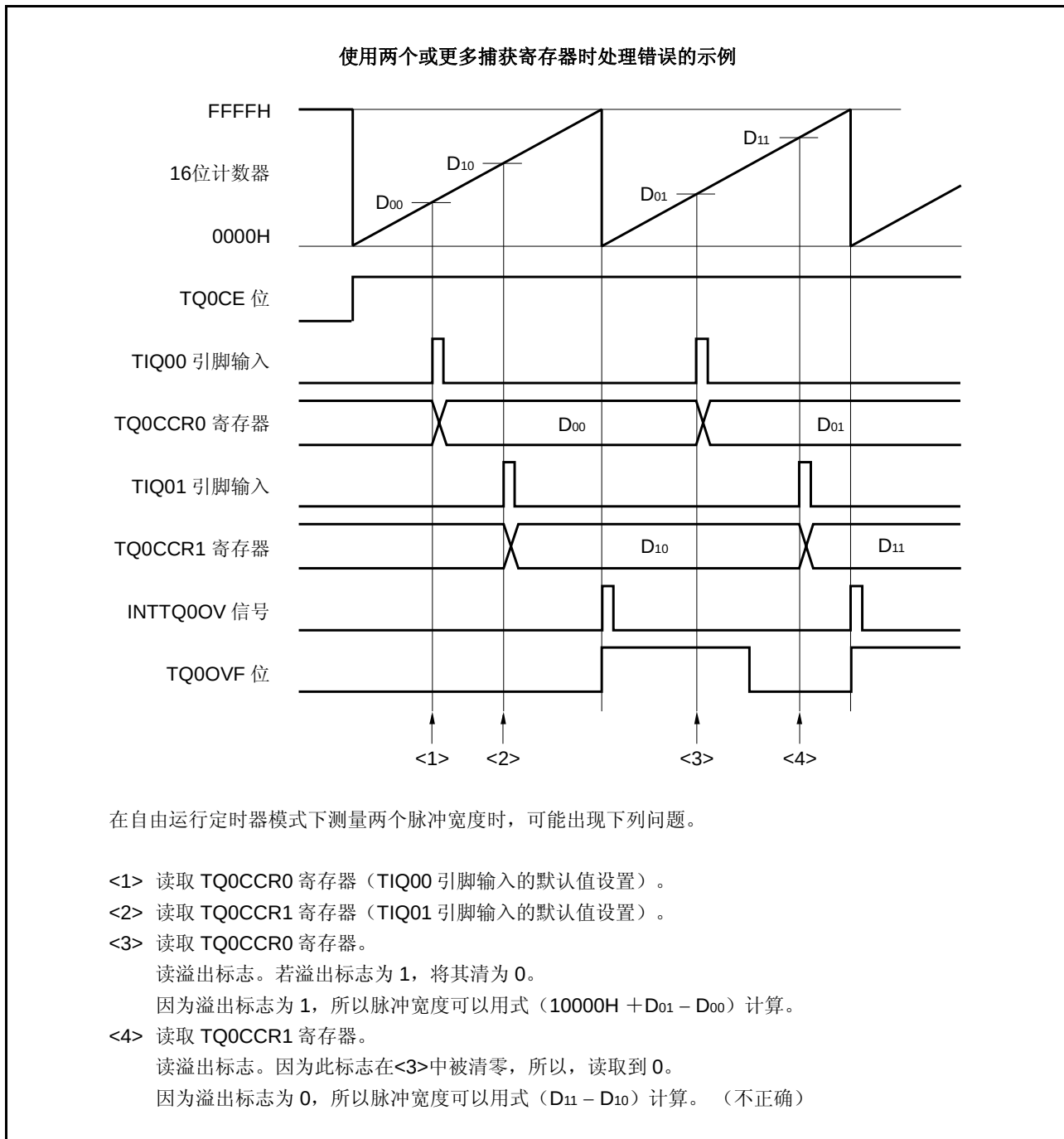
在自由运行定时器模式下执行脉宽测量时，可以用一个通道测量四个脉冲宽度。

要测量脉冲宽度，可以与 INTTQ0CCm 信号同步读取 TQ0CCRm 寄存器的值，并计算读取值与先前读取值之间的差，得到脉冲宽度。

备注 m=0 至 3

(c) 当使用两个或更多捕获寄存器时对溢出的处理

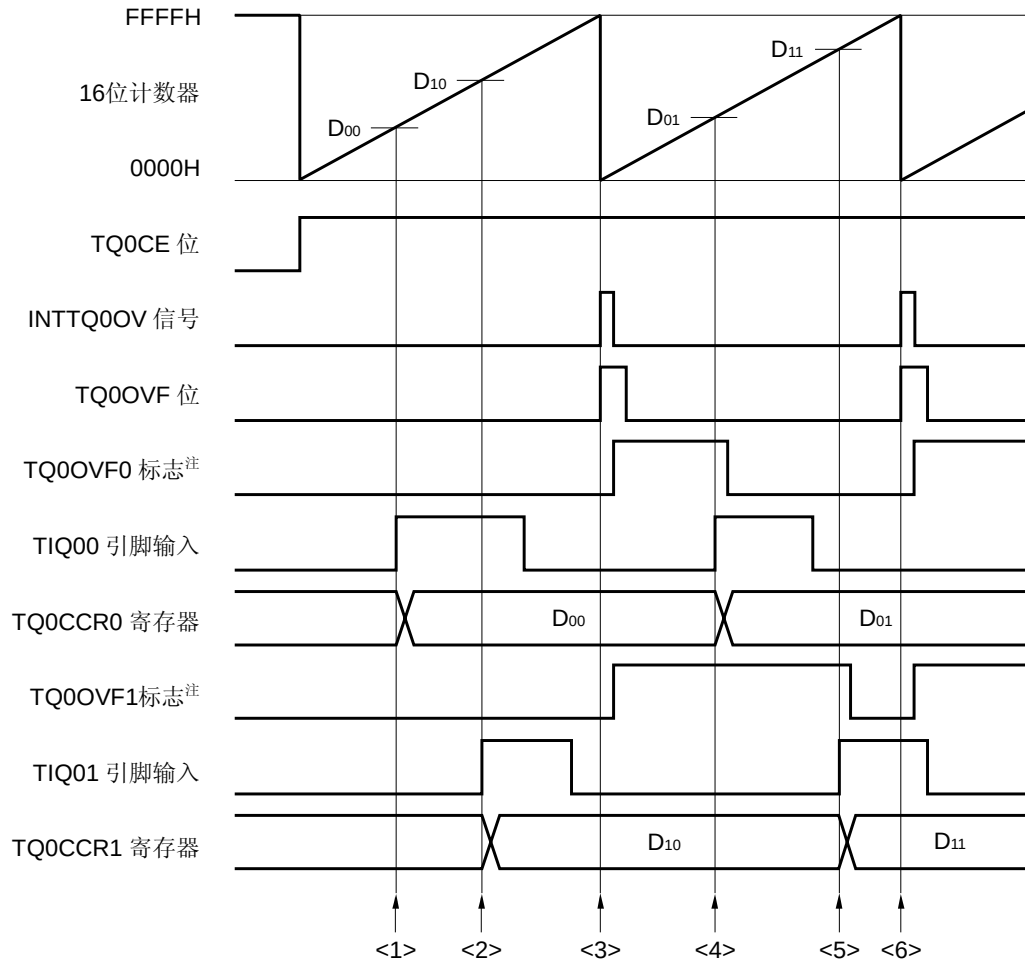
当使用两个捕获寄存器时必须格外小心。首先是一个不正确处理的示例，如下所示。



在使用两个捕获寄存器时，如果溢出标志由一个捕获寄存器清为 0，则另一个捕获寄存器可能得不到正确的脉冲宽度。

在使用两个捕获寄存器时，使用软件解决。关于如何使用软件的示例如下。

使用两个捕获寄存器时的示例（使用溢出中断）



注 由软件对内部 RAM 中的 TQ0OVF0 和 TQ0OVF1 标志进行设置。

<1> 读取 TQ0CCR0 寄存器（TIQ00 引脚输入的默认值设置）。

<2> 读取 TQ0CCR1 寄存器（TIQ01 引脚输入的默认值设置）。

<3> 发生溢出，在溢出中断服务程序中将 TQ0OVF0 和 TQ0OVF1 标志设置为 1，并将溢出标志清为 0。

<4> 读取 TQ0CCR0 寄存器。

读取 TQ0OVF0 标志。如果 TQ0OVF0 标志为 1，则将其清为 0。

因为 TQ0OVF0 标志为 1，所以脉冲宽度可以用式 $(10000H + D_{01} - D_{00})$ 计算。

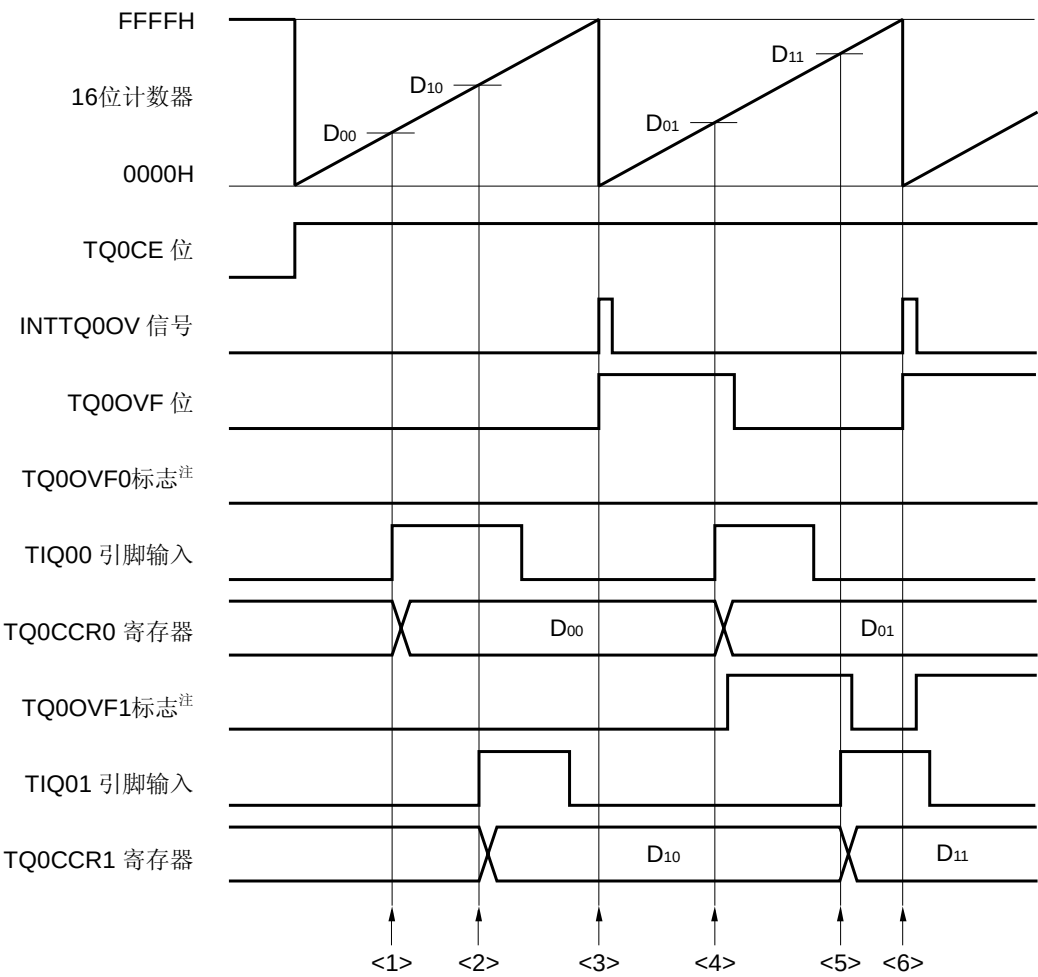
<5> 读取 TQ0CCR1 寄存器。

读取 TQ0OVF1 标志。如果 TQ0OVF1 标志为 1，则将其清为 0（TQ0OVF0 标志在<4>中已清零，而 TQ0OVF1 标志仍然保持为 1）。

因为 TQ0OVF1 标志为 1，所以脉冲宽度可以用式 $(10000H + D_{11} - D_{10})$ 计算。（正确）

<6> 重复 <3>

使用两个捕获寄存器时的示例（不使用溢出中断）



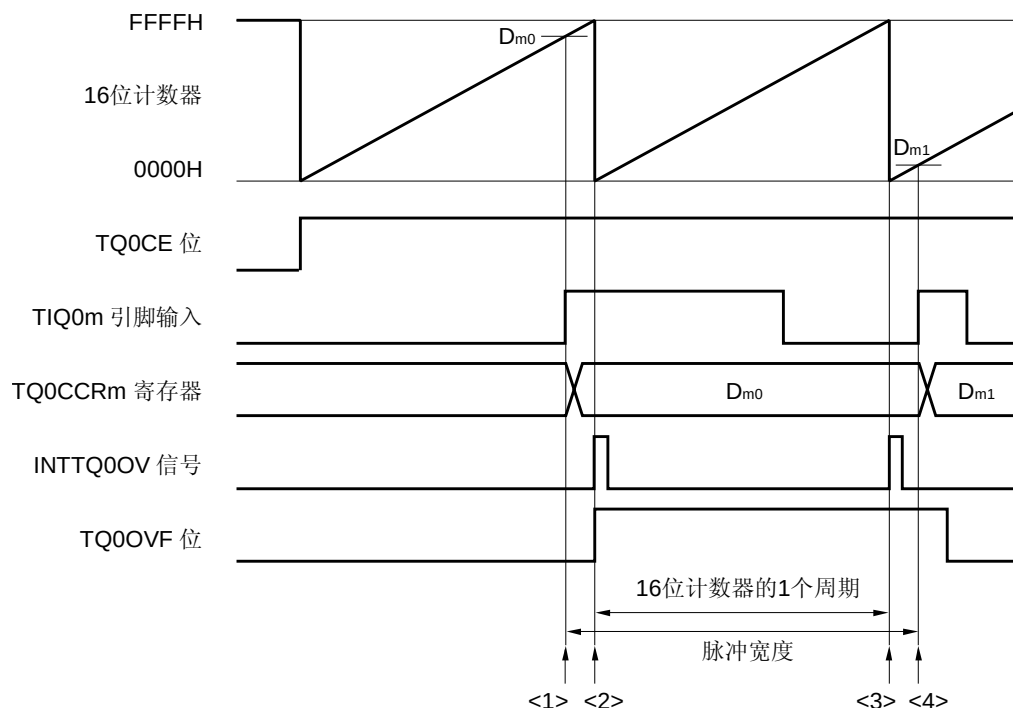
注 由软件对内部 RAM 中的 TQ0OVF0 和 TQ0OVF1 标志进行设置。

- <1> 读取 TQ0CCR0 寄存器（TIQ00 引脚输入的默认值设置）。
- <2> 读取 TQ0CCR1 寄存器（TIQ01 引脚输入的默认值设置）。
- <3> 发生溢出，软件不作任何处理。
- <4> 读取 TQ0CCR0 寄存器。
读溢出标志。若溢出标志为 1，则只把 TQ0OVF1 标志置为 1，并把溢出标志清为 0。
因为溢出标志为 1，所以脉冲宽度可以用式 $(10000H + D_{01} - D_{00})$ 计算。
- <5> 读取 TQ0CCR1 寄存器。
读溢出标志。因为溢出标志在<4>中被清零，所以读取 0。
读取 TQ0OVF1 标志。如果 TQ0OVF1 标志为 1，则将其清为 0。
因为 TQ0OVF1 标志为 1，所以脉冲宽度可以用式 $(10000H + D_{11} - D_{10})$ 计算。（正确）
- <6> 重复 <3>

(d) 当捕获触发间隔较长时的溢出处理

如果脉冲宽度大于 16 位计数器的一个计数循环周期，则必须加以注意，因为从第一个捕获触发信号到下一个触发信号之间可能发生多次溢出。首先是一个不正确处理的示例，如下所示，。

捕获触发间隔较长时的不正确处理示例



在自由运行定时器模式下测量长脉冲的宽度时，如果脉冲宽度较大，可能出现下列问题。

<1> 读取 TQ0CCRm 寄存器（TIQ0m 引脚输入的默认值设置）。

<2> 发生溢出，软件不作任何处理。

<3> 再次发生溢出。软件不作任何处理。

<4> 读取 TQ0CCRm 寄存器。

读溢出标志。若溢出标志为 1，将其清为 0。

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

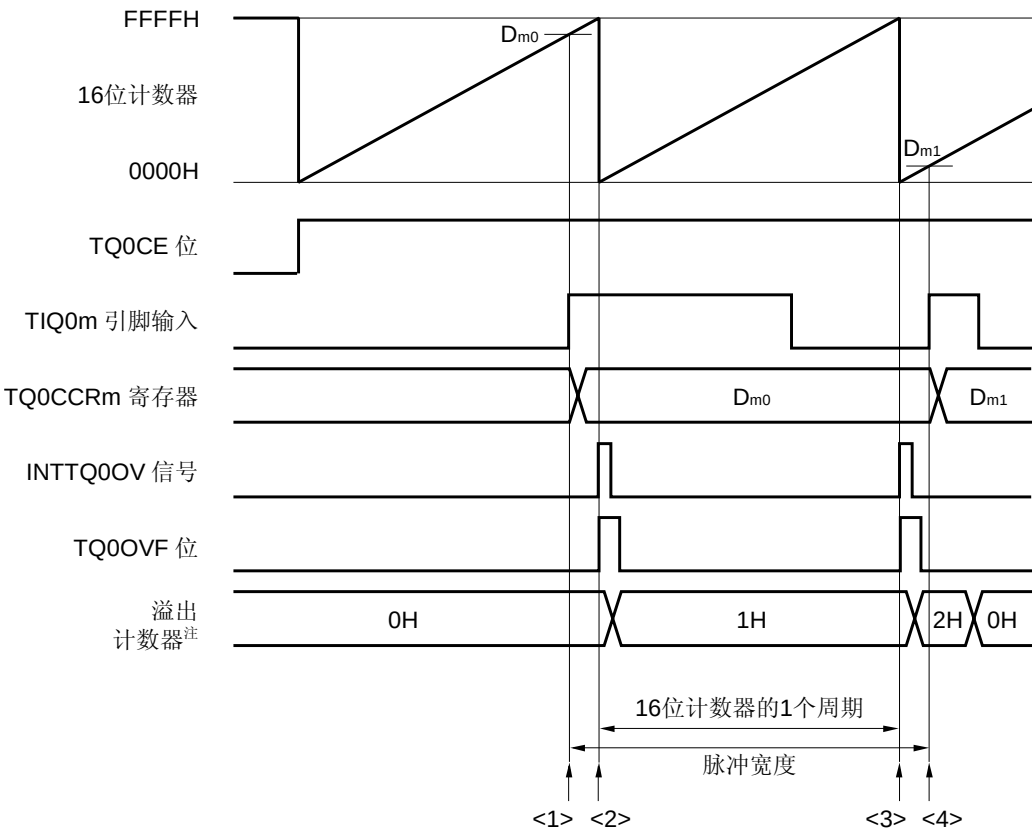
事实上，脉冲宽度应为 $(20000H + D_{m1} - D_{m0})$ ，因为实际已经发生两次溢出。

如果在捕获触发间隔期间，发生两次或更多次的溢出，则无法获得正确的脉冲宽度。

如果捕获触发间隔时间较长时，应该减慢计数时钟以延长 16 位计数器的循环周期，或者使用软件来解决。

关于如何使用软件，示例如下。

捕获触发间隔较长时的示例



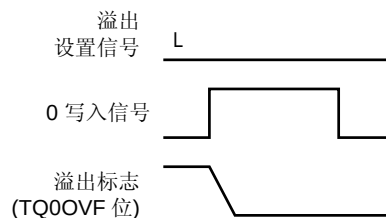
注 内部 RAM 中的溢出计数器由软件任意设置。

- <1> 读取 TQ0CCRm 寄存器 (TIQ0m 引脚输入的默认值设置)。
- <2> 发生溢出, 在溢出中断服务程序中增加溢出计数器的值, 并将溢出标志清为 0。
- <3> 再次发生溢出。在溢出中断服务程序中增加 (+1) 溢出计数器的值, 并把溢出标志清为 0。
- <4> 读取 TQ0CCRm 寄存器。
读取溢出计数器。
→ 当溢出计数器为“N”时, 脉冲宽度可用式 $(N \times 10000H + D_{m1} - D_{m0})$ 计算。
在本例中, 脉冲宽度为 $(20000H + D_{m1} - D_{m0})$, 因为溢出实际发生了两次。
溢出计数器清零 (0H)。

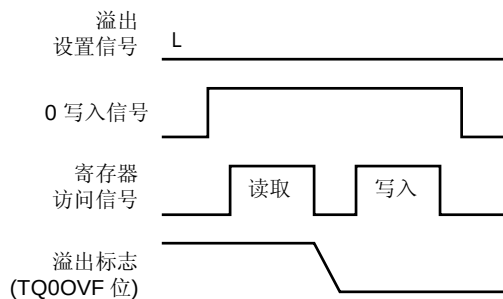
(e) 清除溢出标志

通过 CLR 指令将 TQ0OVF 位清为 0 以及将 8 位数据（第 0 位是 0）写入 TQ0OPT0 寄存器，可以将溢出标志清 0。要准确检测溢出，当 TQ0OVF 位为 1 时读取 TQ0OVF 位，然后用位操作指令将溢出标志清零。

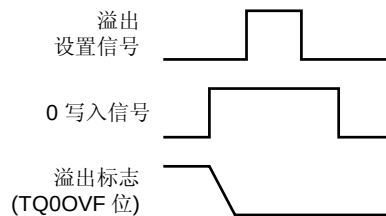
(i) 写入 0 的操作(与设置不冲突)



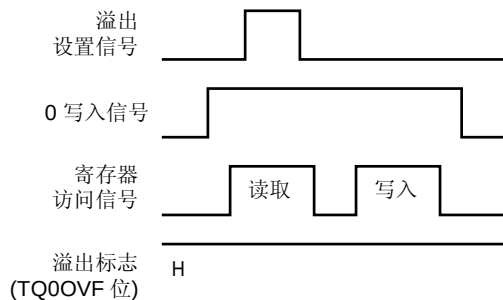
(iii) 清为0的操作 (与设置不冲突)



(ii) 写入 0 的操作 (与设置冲突)



(iv) 清为0的操作 (与设置冲突)



要将溢出标志清为 0，应读取溢出标志，检查该标志如果已经被设置为 1，并用 CLR 指令把它清零。若在未检查溢出标志是否为 1 就把 0 写入溢出标志中，则溢出的设置信息可能因为写入 0 而被清除（上图中的 (ii)）。因此，即使实际发生了溢出，软件也可能会判断为无溢出发生。

如果用 CLR 指令将溢出标志清为 0 时，CLR 指令的执行与当前发生的溢出相冲突，则在清零指令执行后溢出标志仍然会保持置位状态。

8.5.7 脉冲宽度测量模式 (TQ0MD2 至 TQ0MD0 位=110)

在脉冲宽度测量模式下，当 TQ0CTL0.TQ0CE 位被置为 1 时，16 位定时器/事件计数器 Q 开始计数。每次检测到 TIQ0m 引脚输入信号的有效边沿时，16 位计数器的计数值就存储在 TQ0CCRm 寄存器中，且 16 位计数器清零为 0000H。

在出现捕获中断请求信号 (INTTQ0CCm) 之后，可以通过读取 TQ0CCRm 寄存器来测量有效边沿的间隔。

选择 TIQ00 至 TIQ03 引脚作为捕获触发信号输入引脚。使用 TQ0IOC1 寄存器将未使用的引脚指定为“无边沿检测”。

在用外部时钟作为计数时钟时，测量 TIQ0k 引脚输入的脉冲宽度，因为外部时钟固定接入 TIQ00 引脚。此时，把 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位清零为 00（捕获触发输入 (TIQ00 引脚)：无边沿检测）。

备注 m=0 至 3
 k=1 至 3

图 8-34. 脉冲宽度测量模式的配置

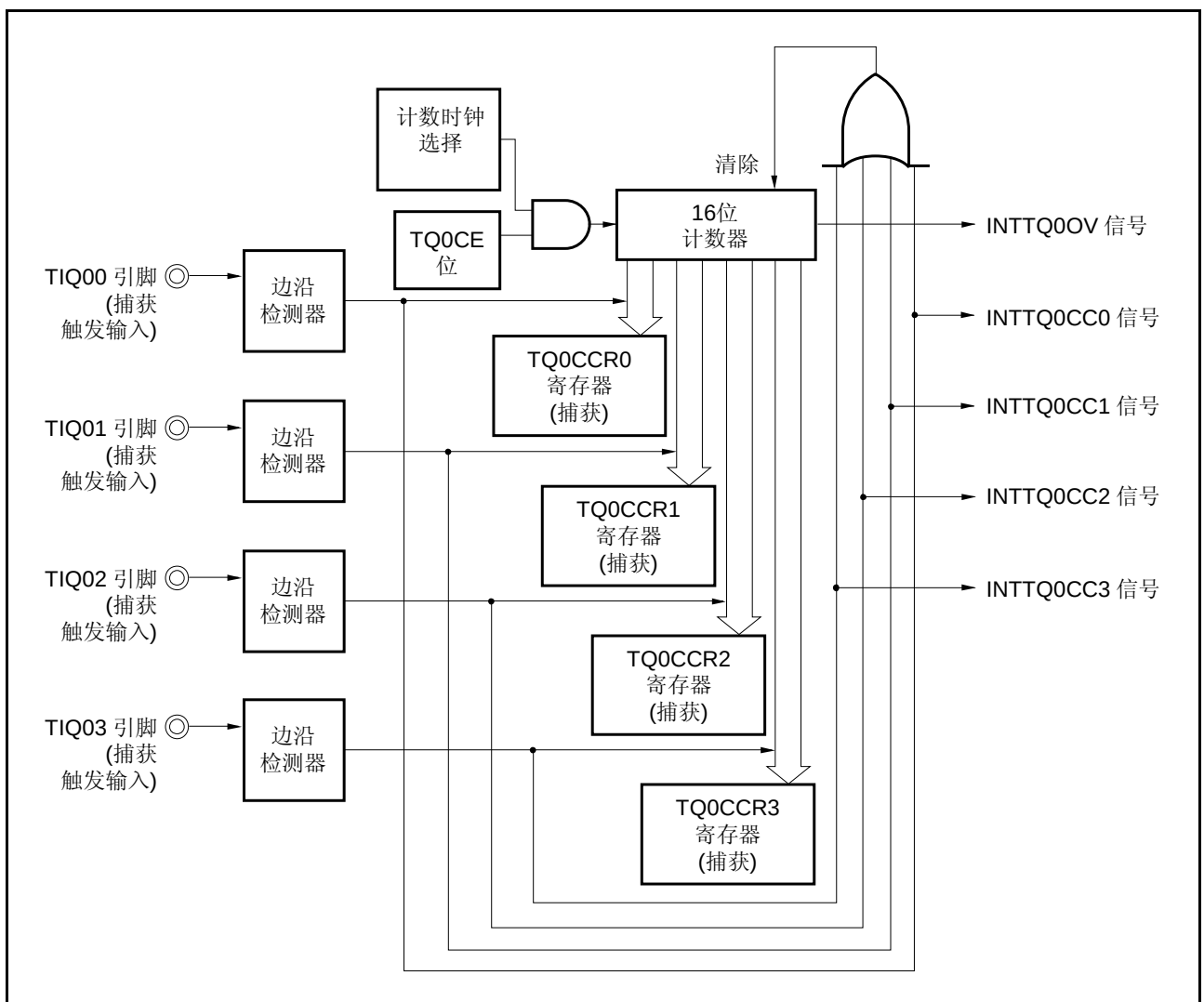
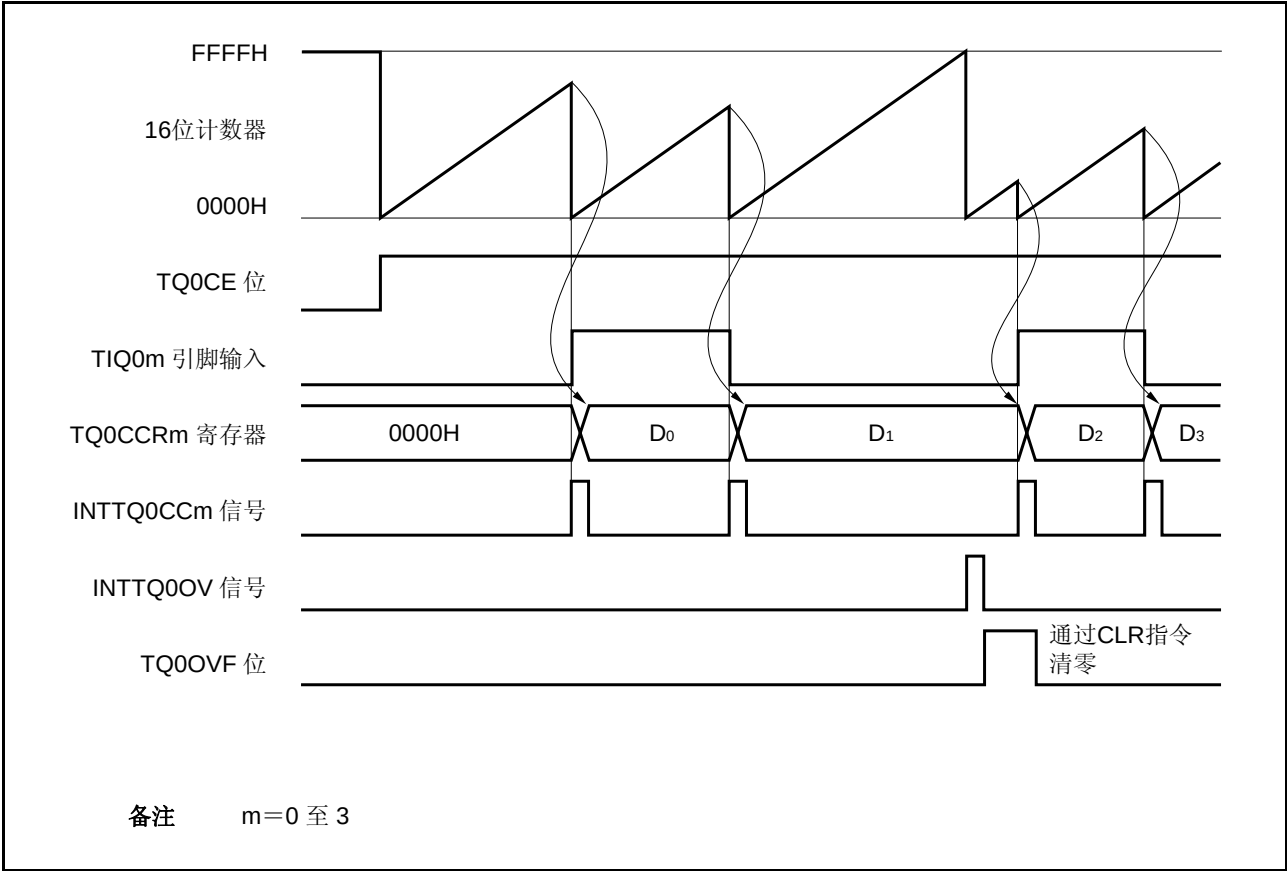


图 8-35. 脉冲宽度测量模式的基本时序



当 TQ0CE 位被置为 1 时，16 位计数器开始计数。当检测到 TIQ0m 引脚输入信号的有效边沿时，16 位计数器的计数值被存储到 TQ0CCRm 寄存器中，16 位计数器清零为 0000H，并产生一个捕获中断请求信号（INTTQ0CCm）。脉冲宽度的计算如下。

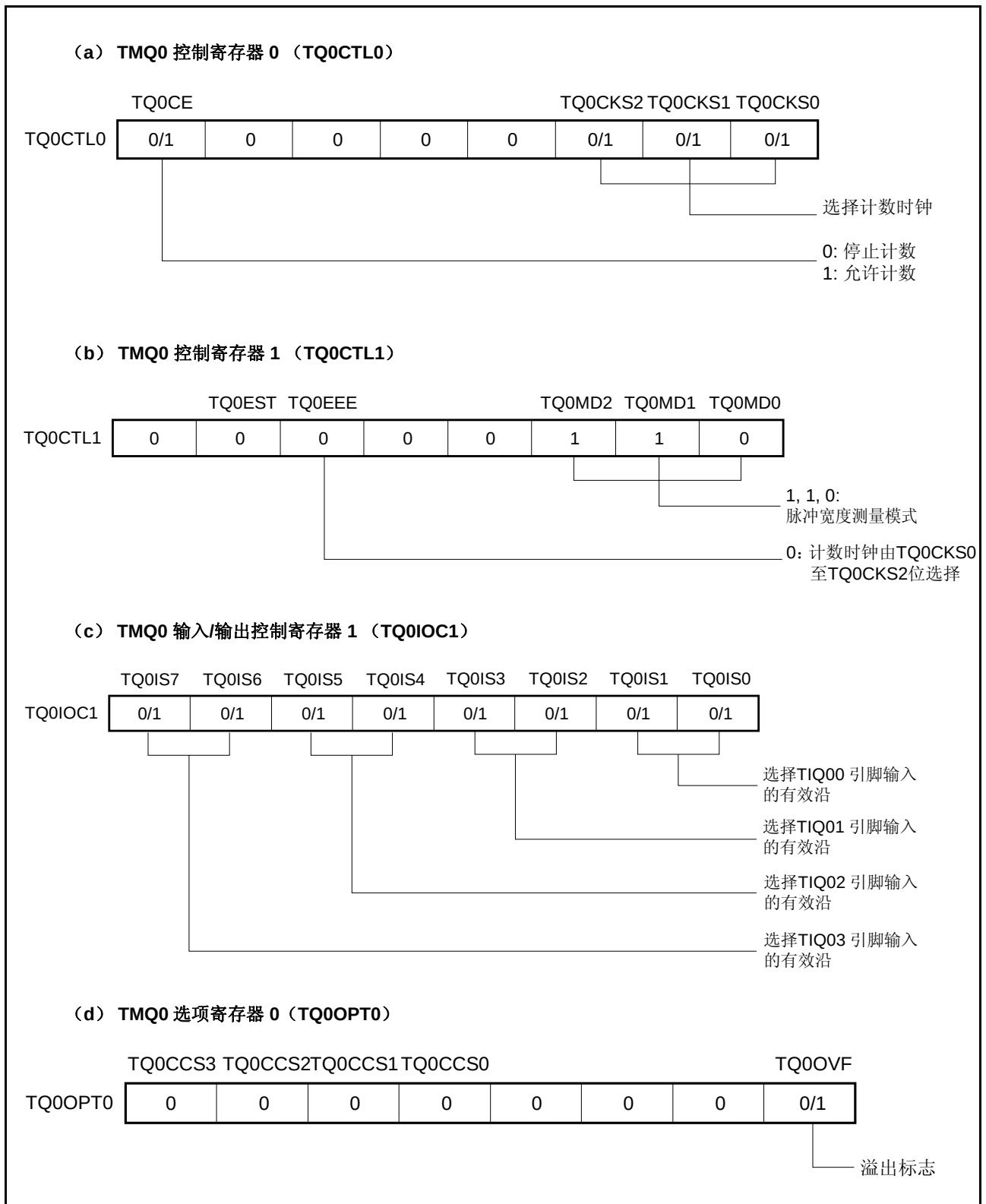
脉冲宽度 = 捕获值 × 计数时钟周期

当 16 位计数器计数到 FFFFH 时，即使没有有效边沿输入 TIQ0m 引脚，在下一个计数时钟时，仍然会产生溢出中断请求信号（INTTQ0OV），且计数器清零为 0000H，然后计数器继续计数。此时，溢出标志（TQ0OPT0.TQ0OVF 位）也被置为 1。使用软件执行 CLR 指令将溢出标志清为 0。若溢出标志被设置为 1，则脉冲宽度的计算如下。

脉冲宽度 = (10000H × TQ0OVF 位置位 (1) 的次数 + 捕获值) × 计数时钟周期

备注 m=0 至 3

图 8-36. 脉冲宽度测量模式时的寄存器设置 (1/2)



<R>

图 8-36. 脉冲宽度测量模式时的寄存器设置 (2/2)

(e) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

通过读取 TQ0CNT 寄存器，可以读出 16 位计数器的值。

(f) TMQ0 捕获/比较寄存器 0 至 3 (TQ0CCR0 至 TQ0CCR3)

当检测到 TIQ0m 引脚输入的有效边沿时，这些寄存器保存 16 位计数器的当前计数值。

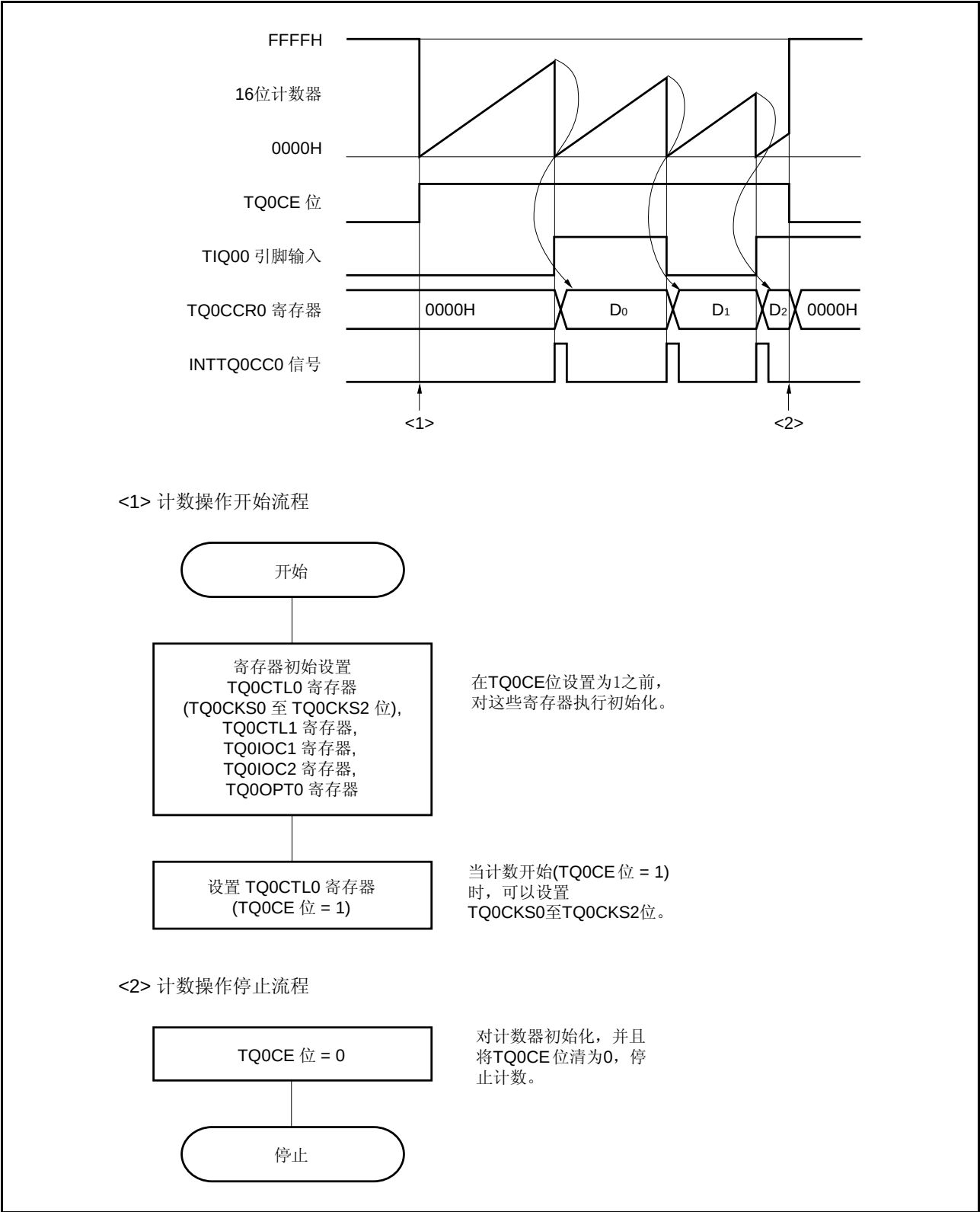
备注

1. 在脉冲宽度测量模式下，不使用 TMQ0 输入/输出控制寄存器 0 (TQ0IOC0) 和 TMQ0 输入/输出控制寄存器 2 (TQ0IOC2)。
2. m=0 至 3

<R>

(1) 脉冲宽度测量模式时的操作流程

图 8-37. 脉冲宽度测量模式的软件处理流程

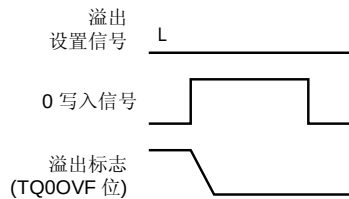


(2) 脉冲宽度测量模式时的操作时序

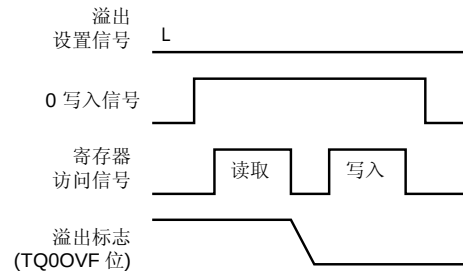
(a) 清除溢出标志

通过 CLR 指令将 TQ0OVF 位清为 0，以及将 8 位数据（第 0 位是“0”）写入 TQ0OPT0 寄存器，可以将溢出标志清 0。要准确检测溢出，当 TQ0OVF 位为 1 时读取 TQ0OVF 位，然后用位操作指令将溢出标志清零。

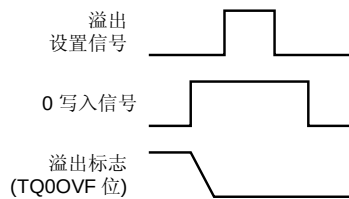
(i) 写入 0 的操作(与设置不冲突)



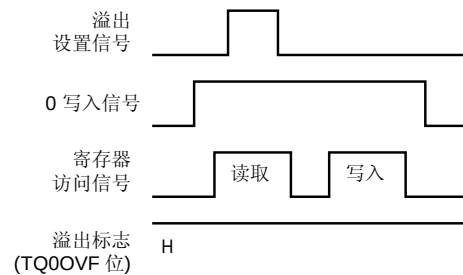
(iii) 清为 0 的操作 (与设置不冲突)



(ii) 写入 0 的操作 (与设置冲突)



(iv) 清为 0 的操作 (与设置冲突)



要将溢出标志清为 0，应读取溢出标志，检查该标志如果已经被设置为 1，用 CLR 指令把它清零。若在未检查溢出标志是否为 1 就把 0 写入溢出标志中，则溢出的设置信息可能因为写入 0 而被清除（上图中的 (ii)）。因此，即使实际发生了溢出，软件也可能会判断为无溢出发生。

如果用 CLR 指令将溢出标志清为 0 时，CLR 指令的执行与当前发生的溢出相冲突，则在清零指令执行后溢出标志仍然会保持置位状态。

8.5.8 定时器输出操作

下表显示了 TOQ00 至 TOQ03 引脚的操作和输出电平。

表 8-6. 各种模式下的定时器输出控制

操作模式	TOQ00 引脚	TOQ01 引脚	TOQ02 引脚	TOQ03 引脚
间隔计时器模式	方波输出			
外部事件计数模式	方波输出	—		
外部触发脉冲输出模式	方波输出	外部触发脉冲输出	外部触发脉冲输出	外部触发脉冲输出
单次触发脉冲输出模式		单次触发脉冲输出	单次触发脉冲输出	单次触发脉冲输出
PWM 输出模式		PWM (脉宽调制信号) 输出	PWM (脉宽调制信号) 输出	PWM (脉宽调制信号) 输出
自由定时器模式	方波输出 (只有使用比较功能时可用)			
脉冲宽度测量模式	—			

表 8-7. 定时器输出控制位对 TOQ00 至 TOQ03 引脚影响的真值表

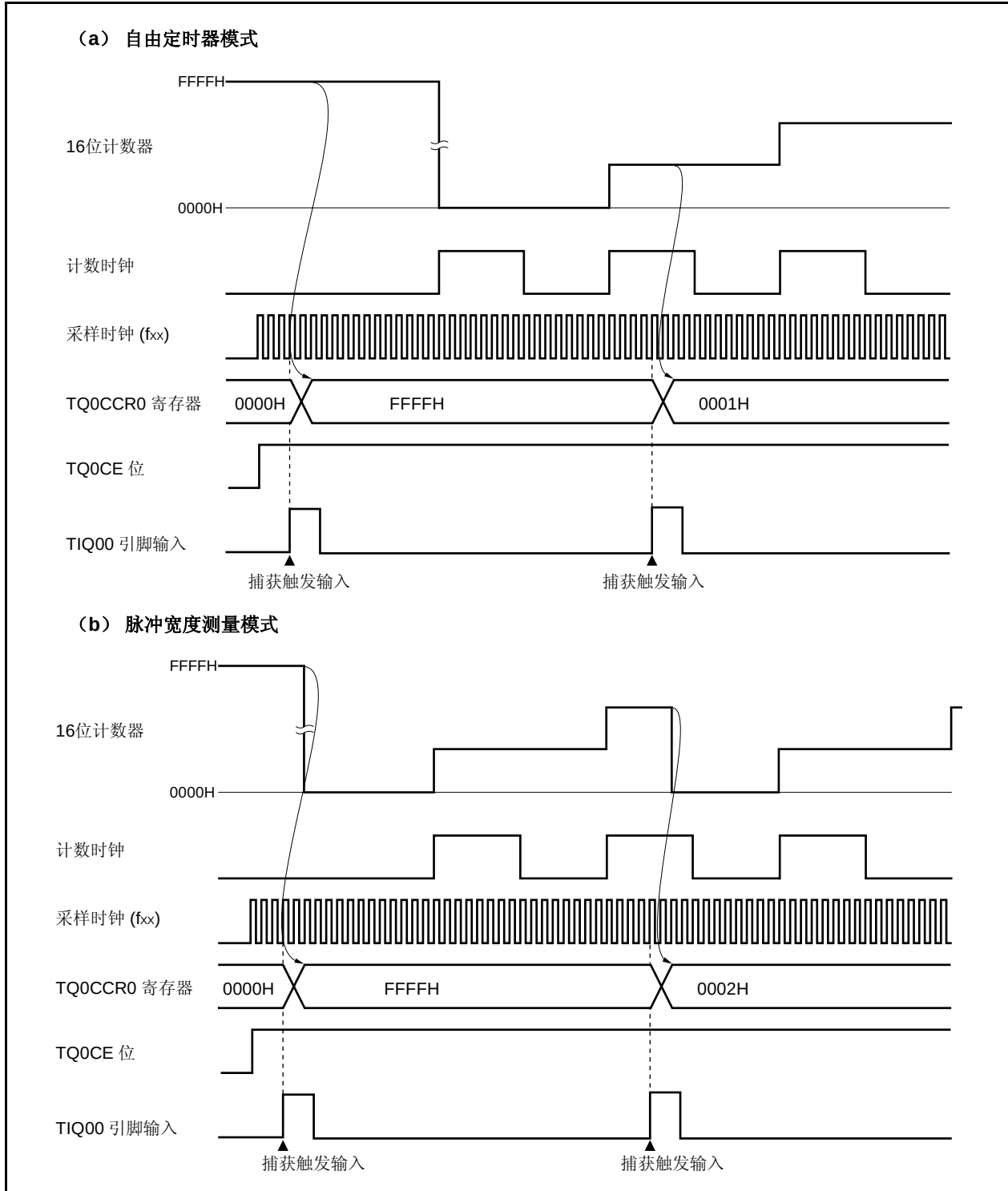
TQ0IOC0.TQ0OLm 位	TQ0IOC0.TQ0OEm 位	TQ0CTL0.TQ0CE 位	TOQ0m 引脚电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前为低电平, 开始计数后为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平, 开始计数后为低电平

备注 m=0 至 3

8.6 注意事项

(1) 捕获操作

当使用捕获操作，并且选择低速时钟作为计数时钟时，如果在 TQ0CE 位被设置为 1 之后立即输入捕获触发，那么捕获到 TQ0CCR0、TQ0CCR1、TQ0CCR2 和 TQ0CCR3 寄存器中的可能是 FFFFH 而不是 0000H。



9.1 概述

- 间隔功能
- 8 路时钟可选
- 16 位计数器 × 1
(定时器计数操作过程中, 不能读取 16 位计数器。)
- 比较寄存器 × 1
(定时器计数操作过程中, 不能写入比较寄存器。)
- 比较匹配中断 × 1

定时器 M 只支持清零并启动模式。不支持自由运行定时器模式。

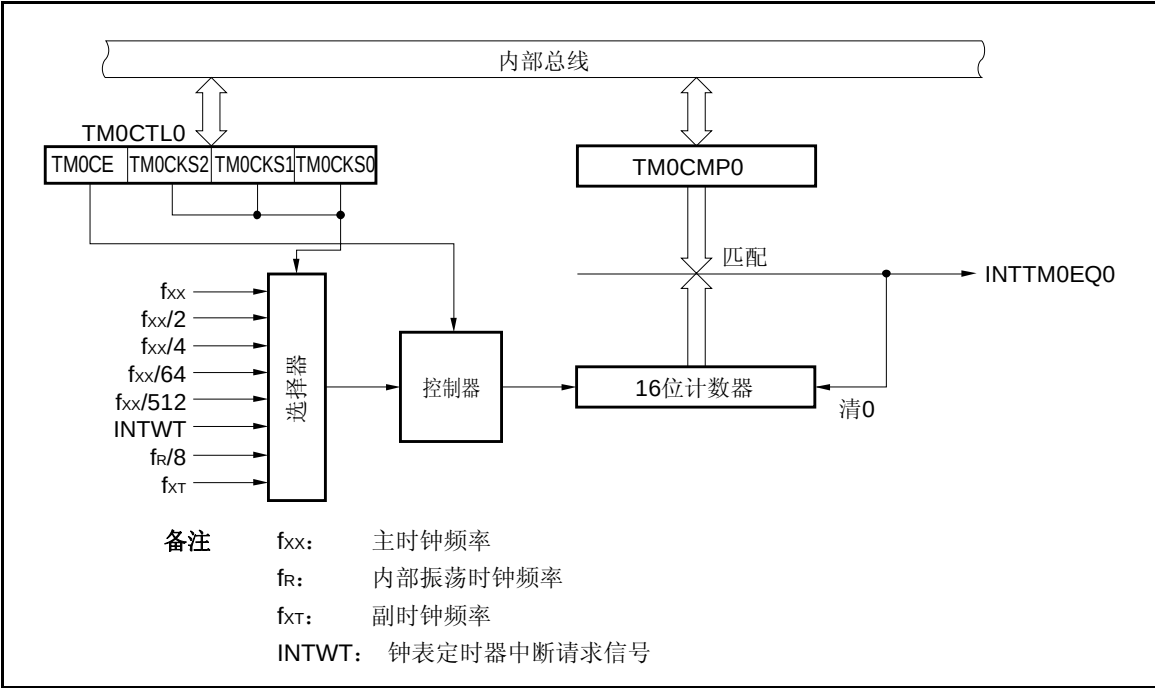
9.2 配置

TMM0 包括下列硬件：

表 9-1. TMM0 的配置

项目	配置
定时器寄存器	16 位计数器
寄存器	TMM0 比较寄存器 0（TM0CMP0）
控制寄存器	TMM0 控制寄存器 0（TM0CTL0）

图 9-1. TMM0 框图

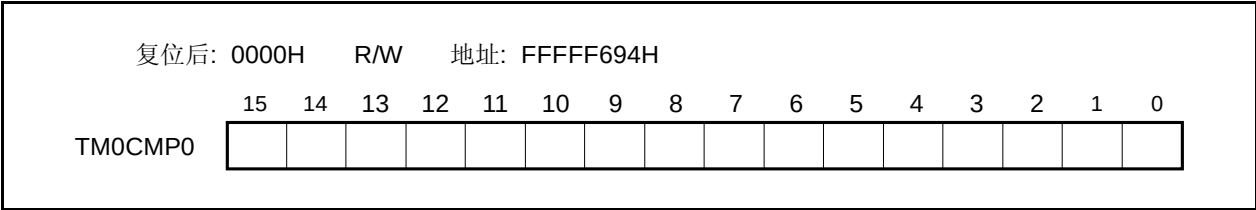


(1) 16 位计数器

该计数器是 16 位计数器，对内部时钟进行计数。
该计数器不可以读写。

(2) TMM0 比较寄存器 0（TM0CMP0）

TM0CMP0 寄存器是一个 16 位的比较寄存器。
该寄存器可以按 16 位宽度进行读取或写入。
复位输入将该寄存器清为 0000H。
随时可以用软件将相同的值写入 TM0CMP0 寄存器。
当 TM0CTL0.TM0CE 位 = 1 时，禁止重写 TM0CMP0 寄存器。



9.3 寄存器

(1) TMM0 控制寄存器（TM0CTL0）

TM0CTL0 寄存器是 8 位寄存器，用于控制定时器 TMM0 的操作。

该寄存器可以按字节读写或按位读写。

复位输入将该寄存器设为 00H。

随时可以用软件将相同的值写入 TM0CTL0 寄存器。

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	内部时钟操作使能/禁止说明
0	TMM0 操作禁止(16位计数器异步复位)。 操作时钟应用停止。
1	TMM0 操作使能。 操作时钟应用启动。TMM0操作开始。
TMM0内部时钟的控制和内部电路的复位与TM0CE位不同步。 当TM0CE 位清 0时，TMM0 的内部时钟被禁止 (固定为低电平) 且 16位计数器异步复位。	

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}

- 注意事项**
1. 当 TM0CE 位=0 时，对 TM0CKS2 至 TM0CKS0 置位。
将 TM0CE 的值从 0 改变为 1 时，不可以同时设置 TM0CKS2 到 TM0CKS0 位的值。
 2. 请确保将第 3 位至第 6 位清为“0”。

备注

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

9.4 操作

注意事项 请勿将 TM0CMP0 寄存器设置为 FFFFH。

9.4.1 间隔计时器模式

在间隔定时器模式中，如果 TM0CTL0.TM0CE 位置为 1，则会以指定时间间隔生成中断请求信号 (INTTM0EQ0)。

图 9-2. 间隔定时器的配置

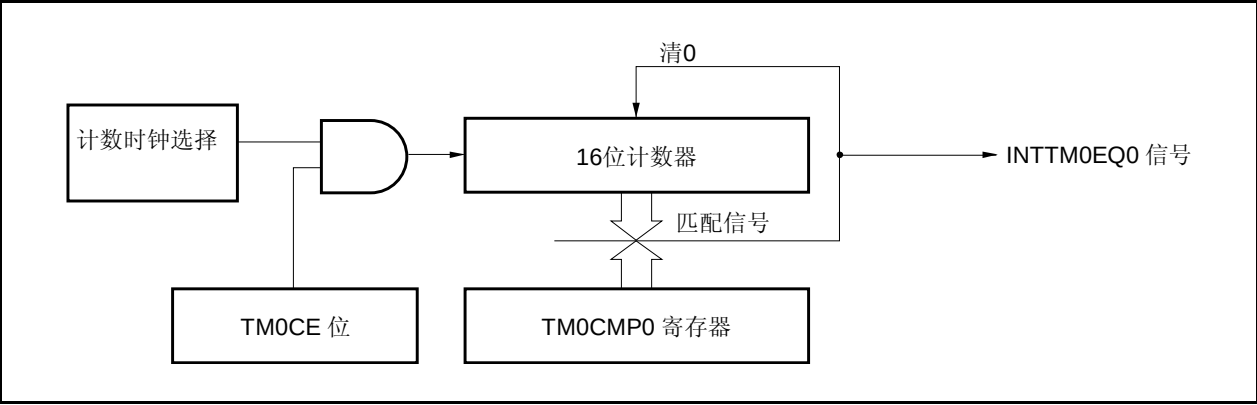
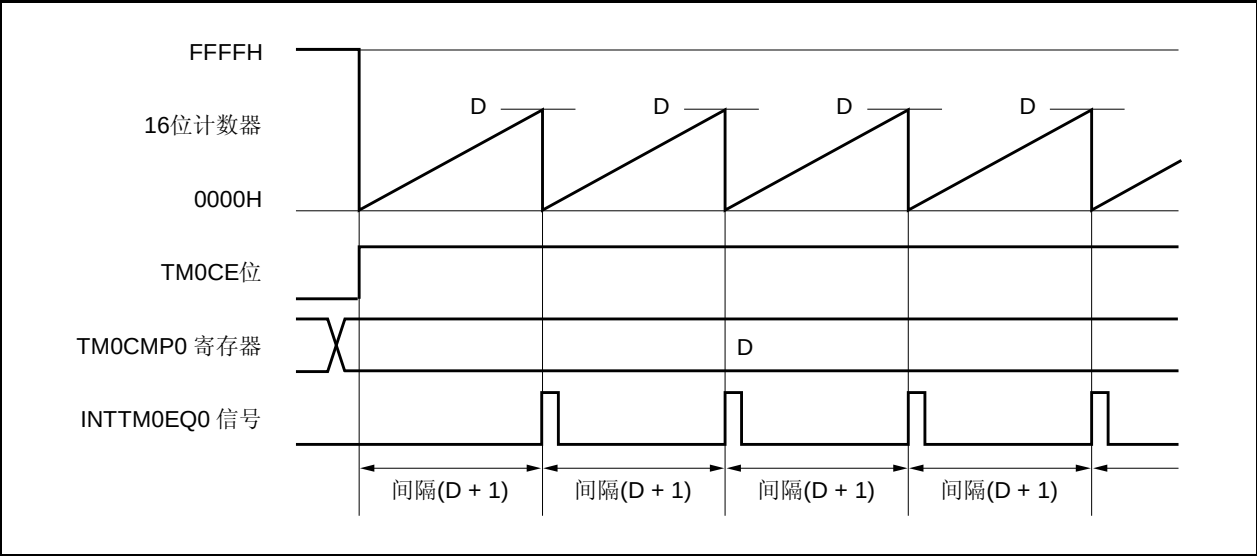


图 9-3. 间隔定时器工作的基本时序



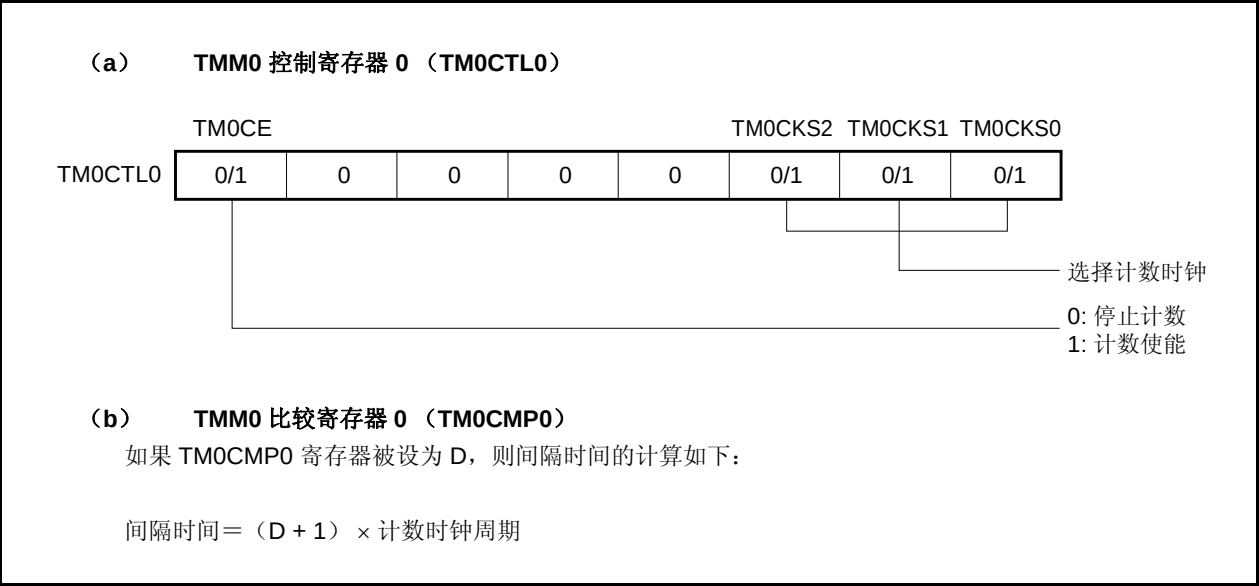
当 TM0CE 位置为 1 时，16 位计数器的值由计数时钟同步从 FFFFH 清零为 0000H，计数器开始计数。

当 16 位计数器的计数值和 TM0CMP0 寄存器的数值相匹配时，16 位计数器清零为 0000H，同时生成一个比较匹配中断请求信号 (INTTM0EQ0)。

时间间隔可用下述表达式计算。

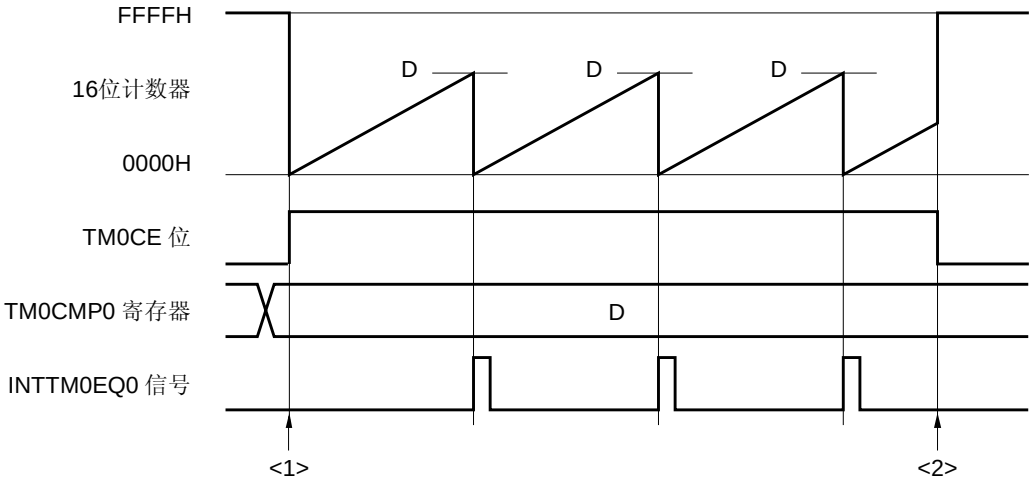
时间间隔 = (TM0CMP0 寄存器的设定值 + 1) × 计数时钟周期

图 9-4. 间隔定时器模式操作时的寄存器设置

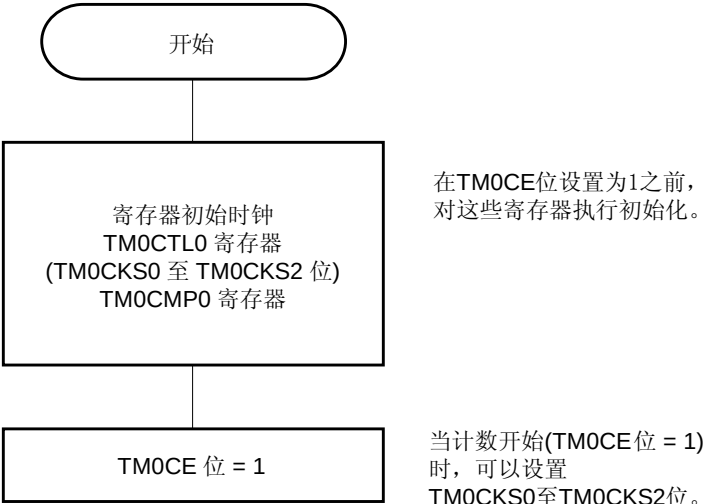


(1) 间隔定时器模式操作流程

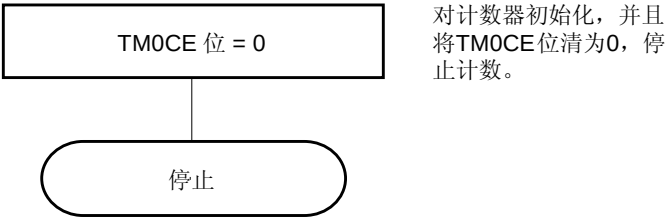
图 9-5. 间隔定时器模式下的软件处理流程



<1> 计数操作开始流程



<2> 计数操作停止流程

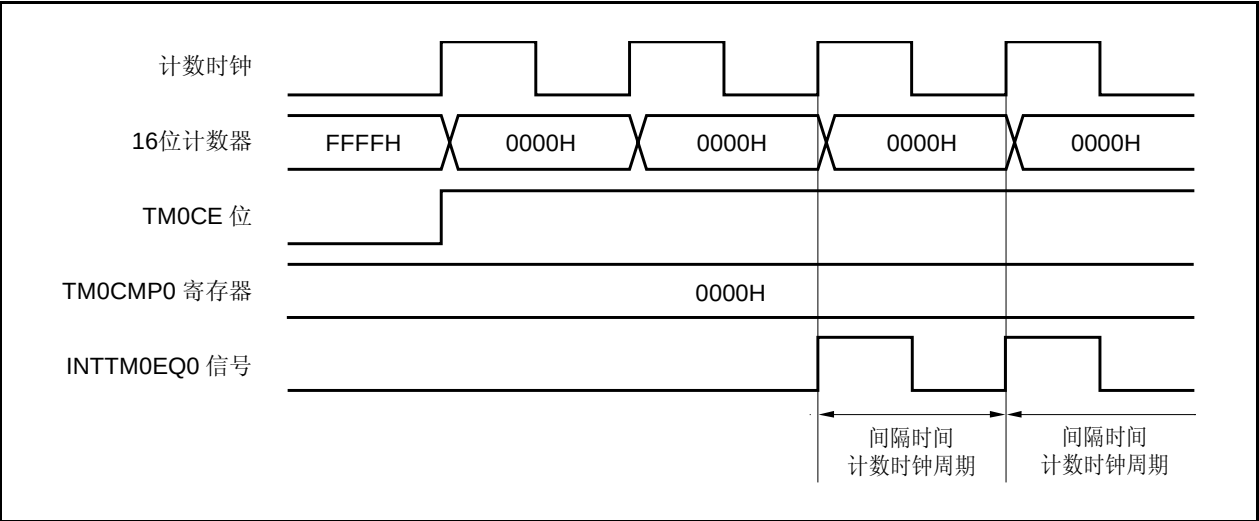


(2) 间隔定时器模式操作时序

注意事项 请勿将 TM0CMP0 寄存器设置为 FFFFH。

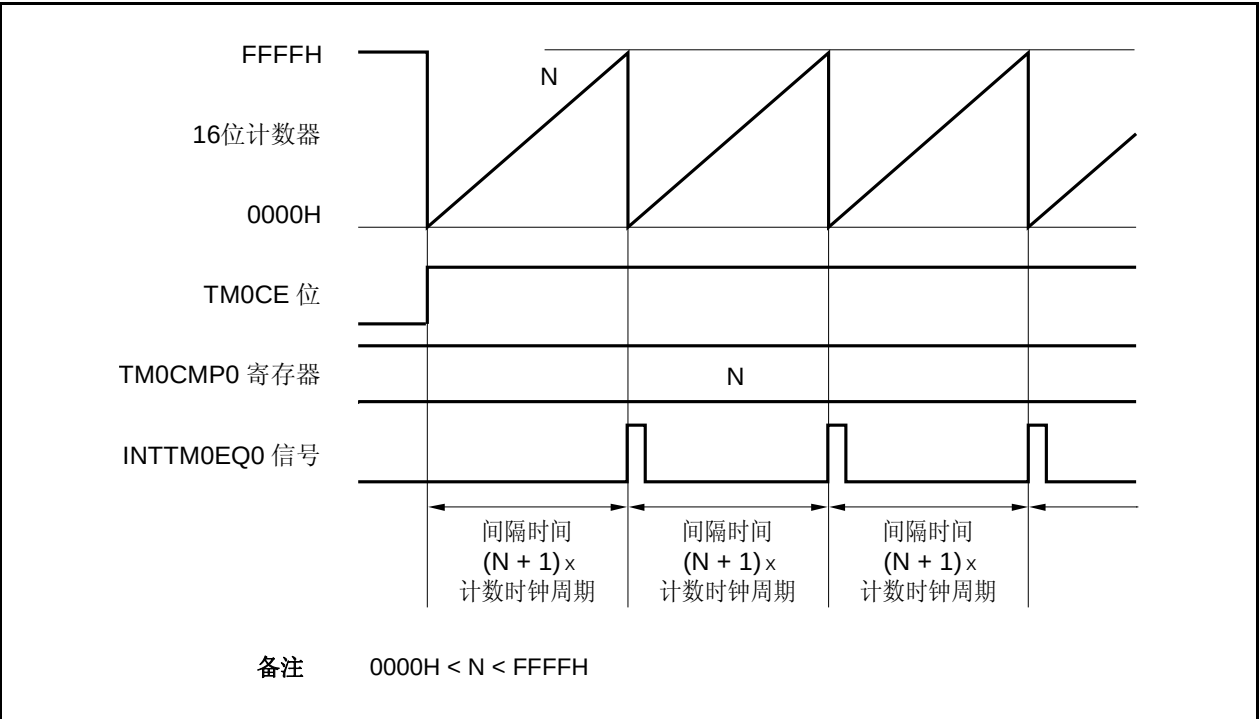
(a) TM0CMP0 寄存器设置为 0000H 时的操作

如果 TM0CMP0 寄存器被清为 0000H，那么每个计数时钟都会产生 INTTM0EQ0 信号。
16 位计数器的值总是 0000H。



(b) TM0CMP0 寄存器被设置为 N 时的操作

如果 TM0CMP0 寄存器设置为 N，则 16 位计数器最大计数至 N。计数器在下一个计数时刻被同步清为 0000H，并产生 INTTM0EQ0 信号。



9.4.2 注意事项

(1) 根据选定的计数时钟，在 TM0CTL0.TM0CE 位置为 1 后，需要 16 位计数器计数以下时间才能启动计数：

选择的计数时钟	计数启动前的最大时间
f_{xx} :	$2/f_{xx}$
$f_{xx}/2$	$6/f_{xx}$
$f_{xx}/4$	$24/f_{xx}$
$f_{xx}/64$	$128/f_{xx}$
$f_{xx}/512$	$1024/f_{xx}$
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，禁止重写 TM0CMP0 和 TM0CTL0 寄存器的值。

如果在 TM0CE 位 = 1 时对这两个寄存器进行重写，那么操作结果将无法保证。

如果这些寄存器被错误地改写，先将 TM0CTL0.TM0CE 位清除为 0，然后重置这些寄存器。

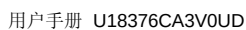
10.1 功能

钟表定时器具有以下功能

- 钟表定时器： 使用主时钟或副时钟，以 0.5 或 0.25 秒时间间隔产生中断请求信号（INTWT）。
- 间隔定时器： 按设定的时间间隔产生中断请求信号（INTWTI）。

钟表定时器和间隔定时器功能可以同时使用。

图 10-1. 钟表定时器框图



(1) 时钟控制

钟表定时器运行于主时钟时，该模块控制工作时钟（ f_x ）的提供和停止。

(2) 3 位预分频器

该预分频器对 f_x 分频，产生 $f_x/2$ ， $f_x/4$ 和 $f_x/8$ 频率时钟。

(3) 8 位计数器

8 位计数器对源时钟（ f_{BGCS} ）进行计数。

(4) 11 位预分频器

该预分频器对 f_w 进行分频，以产生 $f_w/2^4$ 至 $f_w/2^{11}$ 时钟。

(5) 5 位计数器

该计数器以 f_w 或 $f_w/2^9$ 进行计数，并在 $2^4/f_w$ ， $2^5/f_w$ ， $2^{12}/f_w$ 或 $2^{14}/f_w$ 时间间隔处产生钟表定时器中断请求信号。

(6) 选择器

钟表定时器有以下 5 个选择器：

- 用于在 f_x ， $f_x/2$ ， $f_x/4$ 或 $f_x/8$ 中选择作为钟表定时器源时钟的选择器。
- 用于选择主时钟（ f_x ）或副时钟（ f_{XT} ）作为钟表定时器时钟的选择器。
- 用于选择 f_w 或 $f_w/2^9$ 作为 5 位计数器计数时钟频率的选择器。
- 用于选择 $2^4/f_w$ ， $2^{13}/f_w$ ， $2^5/f_w$ 或 $2^{14}/f_w$ 作为 INTWT 信号产生时间间隔的选择器。
- 用于选择 $2^4/f_w$ 至 $2^{11}/f_w$ 作为间隔定时器中断请求信号（INTWTI）产生时间间隔的选择器。

(7) PRSCM 寄存器

该寄存器是 8 位比较寄存器，用于设置间隔时间。

(8) PRSM 寄存器

该寄存器控制提供给钟表定时器的时钟。

(9) WTM 寄存器

该寄存器是 8 位寄存器，用于控制钟表定时器/间隔定时器的工作，并设置中断请求信号产生的时间间隔。

10.3 控制寄存器

钟表定时器提供下列寄存器：

- 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）

PRSM0 寄存器控制钟表定时器计数时钟的发生。
该寄存器可按字节读写或按位读写。
系统复位后，该寄存器设为 00H。

复位后： 00H R/W 地址： FFFFF8B0H

7	6	5	<4>	3	2	1	0
0	0	0	BGCE0	0	0	BGCS01	BGCS00

BGCE0	主时钟操作使能
0	禁止
1	使能

BGCS01	BGCS00	钟表定时器源时钟 (f _{BGCS}) 的选择		
			5 MHz	4 MHz
0	0	f _x	200 ns	250 ns
0	1	f _x /2	400 ns	500 ns
1	0	f _x /4	800 ns	1 μs
1	1	f _x /8	1.6 μs	2 μs

- 注意事项
1. 钟表定时器工作期间，禁止改变 BGCS00 和 BGCS01 位的值。
 2. 在设置 BGCE0 位为 1 之前，先设置 PRSM0 寄存器。
 3. 根据主时钟频率设置 PRSM0 和 PRSCM0 寄存器，以得到 32.768 kHz 的 f_{BRG} 频率。

(2) 预分频比较寄存器 0 (PRSCM0)

PRSCM0 寄存器是 8 位比较寄存器。
该寄存器可按字节读取或写入。
系统复位后，该寄存器被设为 00H。

复位后: 00H R/W 地址: FFFFF8B1H

PRSCM0

7	6	5	4	3	2	1	0
PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00

注意事项

1. 钟表定时器工作期间，禁止重写 PRSCM0 寄存器。

2. 在设置 PRSM0.BGCE0 位为 1 之前，先设置 PRSCM0 寄存器。

3. 根据主时钟频率设置 PRSM0 和 PRSCM0 寄存器，以得到 32.768 kHz 的 f_{BRG} 频率。

关于 f_{BRG} 的计算如下：

$f_{BRG} = f_{BGCS}/2N$

备注 f_{BGCS}: 通过 PRSM0 寄存器设置的钟表定时器源时钟
N: PRSCM0 寄存器的设定值= 1 至 256
但是，仅当 PRSCM0 寄存器设置为 00H 时，N = 256。

(3) 钟表定时器工作模式寄存器 (WTM)

WTM 寄存器可以使能或禁止钟表定时器的计数时钟和工作，设置预分频器的间隔时间，控制 5 位计数器的操作并且设置钟表标志的设定时间。

设置 WTM 寄存器之前，设置 PRSM0 寄存器。

该寄存器可按字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

(1/2)

复位后：00H R/W 地址：FFFF680H

	7	6	5	4	3	2	<1>	<0>
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	WTM6	WTM5	WTM4	预分频器间隔时间的选择
0	0	0	0	$2^4/f_W$ (488 μ s: $f_W = f_{XT}$)
0	0	0	1	$2^5/f_W$ (977 μ s: $f_W = f_{XT}$)
0	0	1	0	$2^6/f_W$ (1.95 ms: $f_W = f_{XT}$)
0	0	1	1	$2^7/f_W$ (3.91 ms: $f_W = f_{XT}$)
0	1	0	0	$2^8/f_W$ (7.81 ms: $f_W = f_{XT}$)
0	1	0	1	$2^9/f_W$ (15.6 ms: $f_W = f_{XT}$)
0	1	1	0	$2^{10}/f_W$ (31.3 ms: $f_W = f_{XT}$)
0	1	1	1	$2^{11}/f_W$ (62.5 ms: $f_W = f_{XT}$)
1	0	0	0	$2^4/f_W$ (488 ms: $f_W = f_{BRG}$)
1	0	0	1	$2^5/f_W$ (977 ms: $f_W = f_{BRG}$)
1	0	1	0	$2^6/f_W$ (1.95 ms: $f_W = f_{BRG}$)
1	0	1	1	$2^7/f_W$ (3.90 ms: $f_W = f_{BRG}$)
1	1	0	0	$2^8/f_W$ (7.81 ms: $f_W = f_{BRG}$)
1	1	0	1	$2^9/f_W$ (15.6 ms: $f_W = f_{BRG}$)
1	1	1	0	$2^{10}/f_W$ (31.2 ms: $f_W = f_{BRG}$)
1	1	1	1	$2^{11}/f_W$ (62.5 ms: $f_W = f_{BRG}$)

WTM7	WTM3	WTM2	钟表标志设置时间的选择
0	0	0	$2^{14}/f_w$ (0.5 s: $f_w = f_{XT}$)
0	0	1	$2^{13}/f_w$ (0.25 s: $f_w = f_{XT}$)
0	1	0	$2^5/f_w$ (977 μ s: $f_w = f_{XT}$)
0	1	1	$2^4/f_w$ (488 μ s: $f_w = f_{XT}$)
1	0	0	$2^{14}/f_w$ (0.5 s: $f_w = f_{BRG}$)
1	0	1	$2^{13}/f_w$ (0.25 s: $f_w = f_{BRG}$)
1	1	0	$2^5/f_w$ (977 μ s: $f_w = f_{BRG}$)
1	1	1	$2^4/f_w$ (488 μ s: $f_w = f_{BRG}$)

WTM1	5位计数器操作的控制
0	操作停止后清零
1	启动

WTM0	钟表定时器操作允许
0	停止操作 (预分频器和5位计数器都清零)
1	允许操作

注意事项 当 **WTM0** 和 **WTM1** 位都为 0 时，重写 **WTM2** 至 **WTM7** 位。

备注

1. f_w : 钟表定时器时钟频率
2. 当 $f_w = 32.768\text{ kHz}$ 时，使用括号内数值。

10.4 操作

10.4.1 作为钟表定时器

钟表定时器以固定的时间间隔产生中断请求信号（INTWT）。使用副时钟（32.768 kHz）或主时钟，钟表定时器以 0.25 或 0.5 秒时间间隔工作。

WTM.WTM1 和 WTM.WTM0 位设置为 1 时，计数操作启动。WTM0 位清除为 0 时，11 位预分频器和 5 位计数器清零且计数操作停止。

当钟表定时器与间隔定时器同时工作时，通过清除 WTM1 位为 0 然后再清除 5 位计数器，可以对钟表定时器的时间进行调整。此时，钟表定时器可能会产生最大 15.6 ms 的误差，但间隔定时器不受影响。

如果主时钟用作钟表定时器的计数时钟，则用 PRSM0.BGCS01 和 BGCS00 位和 PRSCM0 寄存器的 8 位比较值，钟表定时器的计数时钟频率（f_{BRG}）可以被设置为 32.768 kHz。

当 PRSM0.BGCE0 位置位（1）时，f_{BRG} 提供给钟表定时器。

f_{BRG} 可通过下式进行计算。

$$f_{BRG} = f_x / (2^{m+1} \times N)$$

为了将 f_{BRG} 设置为 32.768 kHz，按以下公式计算设置到 BGCS01 和 BGCS00 位以及 PRSCM0 寄存器的值。

<1> 设置 $N = f_x / 65,536$ 。设置 $m = 0$ 。

<2> 在 N 的第一个十进制数取整，如果为偶数，在取整前将 N 设置为 N/2，m 为 m + 1。

<3> 重复步骤<2>，直到 N 为奇数或 m = 3。

<4> 将 N 的第一个十进制数位置取整结果设置到 PRSCM0 寄存器，m 的值设置到 BGCS01 和 BGCS00 位。

例：当 $f_x = 4.00 \text{ MHz}$ 时

<1> $N = 4,000,000 / 65,536 = 61.03\dots$, $m = 0$

<2>, <3> 因为 N（取整到第一个十进制位）是奇数， $N = 61$, $m = 0$ 。

<4> 设置 PRSCM0 寄存器的值：3DH（61），设置 BGCS01 和 BGCS00 位的值：00

此时，实际的 f_{BRG} 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（BGCS01 位和 BGCS00 位设定值）= 0 至 3

N：PRSCM0 寄存器的设定值= 1 至 256

但是，仅当 PRSCM0 寄存器设置为 00H 时，N = 256。

f_x：主时钟振荡频率

10.4.2 作为间隔定时器

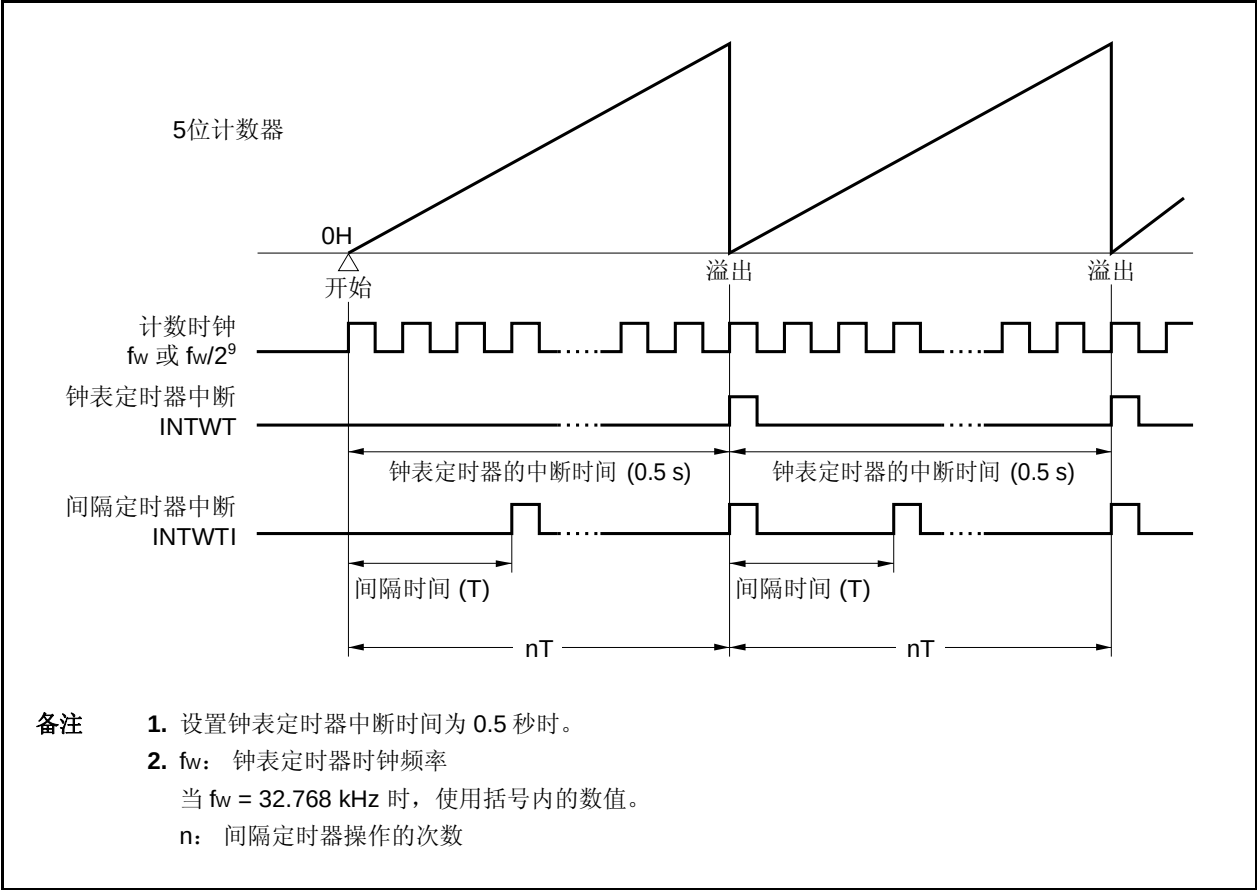
钟表定时器也可用作间隔定时器，按照事先设定的间隔重复产生中断请求信号（INTWTI）。
间隔时间由 WTM 寄存器的 WTM4 至 WTM7 位选择。

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

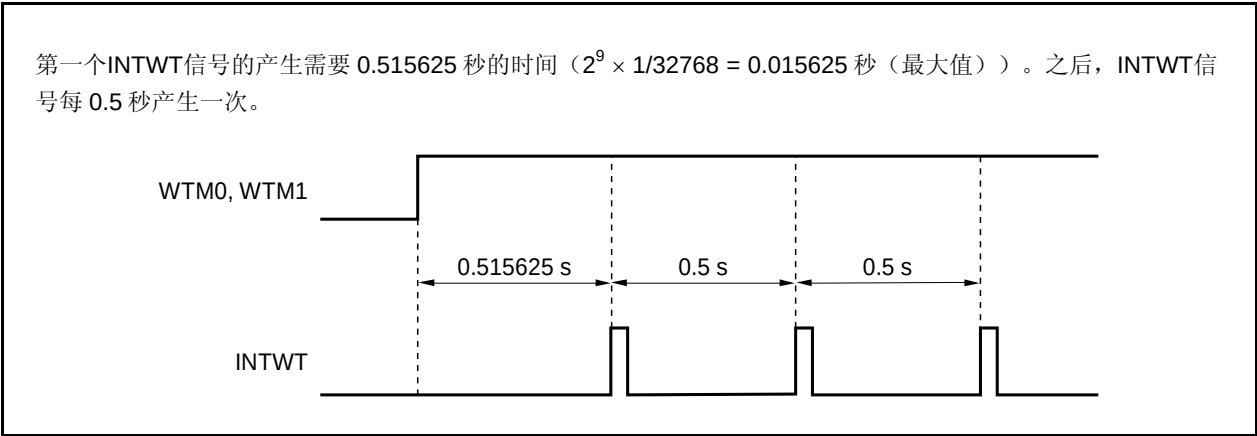
图 10-2. 钟表定时器/间隔定时器操作时序



10.4.3 注意事项

操作使能 (WTM.WTM1 和 WTM.WTM0 位 = 1) 后到首次产生钟表定时器中断请求 (INTWT) 信号, 需要一些时间。

图 10-3. 产生钟表定时器中断请求信号 (INTWT) 的示例 (当中断周期 = 0.5 秒时)



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 具有以下功能：

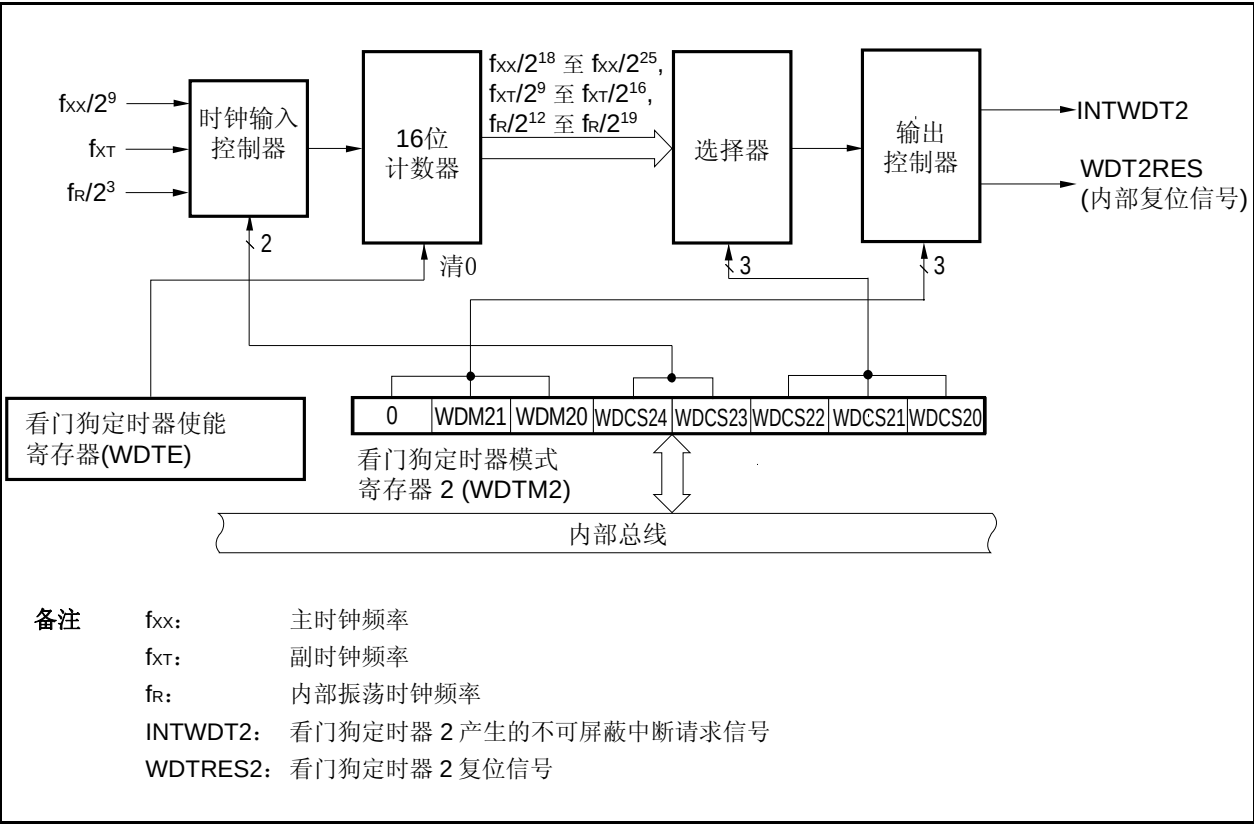
- 默认启动的看门狗定时器^{注1}
 - 复位模式：看门狗定时器 2 溢出后执行复位操作（产生 WDT2RES 信号）
 - 不可屏蔽中断请求模式：看门狗定时器 2 溢出后执行 NMI 操作（产生 INTWDT2 信号）^{注2}
- 时钟源可以选择主时钟、内部振荡时钟或副时钟

- 注**
1. 在复位模式下，复位释放后看门狗定时器 2 就会自动启动。
当不使用看门狗定时器 2 时，或者在系统复位执行前使用该功能将其停止，或者在下一个间隔时间内对看门狗定时器 2 执行一次清除然后再停止其工作。
即使不需要改变默认设置（复位模式，间隔时间： $f_R/2^{19}$ ），为了确保操作的正确性，也要对 WDTM2 寄存器执行一次写入操作。
 2. 关于不可屏蔽中断请求信号（INTWDT2）的不可屏蔽中断服务程序，参见 19.2.2（2）自 INTWDT2 的信号。

11.2 配置

看门狗定时器 2 的功能框图如下所示：

图 11-1. 看门狗定时器 2 的功能框图



看门狗定时器 2 包括下列硬件：

表 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)

WDTM2 寄存器用于设置看门狗定时器 2 的溢出时间和工作时钟。

该寄存器可按字节读取或写入。该寄存器读取次数不限，但是仅能在复位解除后写入一次。

复位输入将该寄存器设置为 67H。

注意事项 下列状态下禁止访问 WDTM2 寄存器。详细说明，参见 3.4.8 (2) 访问片上外设 I/O 专用寄存器。

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

复位后: 67H R/W 地址: FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	WDCS24	WDCS23	WDCS22	WDCS21	WDCS20

WDM21	WDM20	看门狗定时器2操作模式的选择
0	0	停止工作
0	1	不可屏蔽中断请求模式 ^L (产生INTWDT2 信号)
1	—	复位模式 (产生WDT2RES 信号)

- 注意事项**
1. 关于 WDCS20 至 WDCS24 位的细节，参见表 11-2 看门狗定时器 2 的时钟选择。
 2. 虽然停止内部振荡器的运行就可以停止看门狗定时器 2 的操作，还是要对 WDTM2 寄存器清零，以确保停止该定时器（为了避免由于误写入操作而选择主时钟或副时钟）。
 3. 复位后，如果对 WDTM2 寄存器执行了两次写入操作，那么将强制产生溢出信号且计数器复位。
 4. 为了有意产生溢出信号，仅需向 WDTM2 寄存器写入两次数据，或者向 WDTE 寄存器写入一次“ACH”之外的数值。
但是，当看门狗定时器 2 被设置为停止运行时，无论向 WDTM2 寄存器写入两次数据，还是向 WDTE 寄存器写入一次“ACH”之外的数值，都不会产生溢出信号。
 5. 为了停止看门狗定时器 2，将RCM.RSTOP位置为 1（停止内部振荡器），并且将 00H写入 WDTM2 寄存器。如果RCM.RSTOP位不能置为 1，就将WDCS23 位置为 1（2ⁿ/f_{xx}选择为时钟，且该时钟可以在IDLE1、IDLW2、sub-IDLE和副时钟操作模式下停止）。

表 11-2. 看门狗定时器 2 时钟选择

WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	选择时钟	100 kHz (MIN.)	220 kHz (TYP.)	400 kHz (MAX.)
0	0	0	0	0	$2^{12}/f_R$	41.0 ms	18.6 ms	10.2 ms
0	0	0	0	1	$2^{13}/f_R$	81.9 ms	37.2 ms	20.5 ms
0	0	0	1	0	$2^{14}/f_R$	163.8 ms	74.5 ms	41.0 ms
0	0	0	1	1	$2^{15}/f_R$	327.7 ms	148.9 ms	81.9 ms
0	0	1	0	0	$2^{16}/f_R$	655.4 ms	297.9 ms	163.8 ms
0	0	1	0	1	$2^{17}/f_R$	1,310.7 ms	595.8 ms	327.7 ms
0	0	1	1	0	$2^{18}/f_R$	2,621.4 ms	1,191.6 ms	655.4 ms
0	0	1	1	1	$2^{19}/f_R$	5,242.9 ms	2,383.1 ms	1,310.7 ms
						$f_{XX} = 32 \text{ MHz}$	$f_{XX} = 20 \text{ MHz}$	$f_{XX} = 10 \text{ MHz}$
0	1	0	0	0	$2^{18}/f_{XX}$	8.2 ms	13.1 ms	26.2 ms
0	1	0	0	1	$2^{19}/f_{XX}$	16.4 ms	26.2 ms	52.4 ms
0	1	0	1	0	$2^{20}/f_{XX}$	32.8 ms	52.4 ms	104.9 ms
0	1	0	1	1	$2^{21}/f_{XX}$	65.5 ms	104.9 ms	209.7 ms
0	1	1	0	0	$2^{22}/f_{XX}$	131.1 ms	209.7 ms	419.4 ms
0	1	1	0	1	$2^{23}/f_{XX}$	262.1 ms	419.4 ms	838.9 ms
0	1	1	1	0	$2^{24}/f_{XX}$	524.3 ms	838.9 ms	1,677.7 ms
0	1	1	1	1	$2^{25}/f_{XX}$	1,048.6 ms	1,677.7 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 WDTE 寄存器，可以清除看门狗定时器 2 的计数值，并且重新开始计数。

WDTE 寄存器可以按字节读取或写入。

复位输入将该寄存器设为 9AH。

复位后: 9AH R/W 地址: FFFFF6D1H

	7	6	5	4	3	2	1	0
WDTE								

注意事项 1. 当“ACH”之外的数值写入 WDTE 寄存器时，强制输出一个溢出信号。

2. 对 WDTE 寄存器执行 1 位存储处理指令，强制输出一个溢出信号。

3. 为了有意产生溢出信号，向 WDTE 寄存器写入一次“ACH”之外的数据，或者向 WDTM2 寄存器写入两次数据即可。

但是，当看门狗定时器 2 设置为停止运行时，无论向 WDTM2 寄存器写入两次数据，还是向 WDTE 寄存器写入一次“ACH”之外的数值，都不会产生溢出信号。

4. 寄存器的读出值总是“9AH”（与写入值“ACH”不同）。

11.4 操作

在复位模式下，当复位解除后，看门狗定时器 2 就会自动启动。

复位后，仅能使用字节访问对 WDTM2 寄存器写入一次。要使用看门狗定时器 2，应使用字节存储处理指令将工作模式以及间隔时间写入 WDTM2 寄存器。之后，将无法停止看门狗定时器的操作。

WDTM2 的 WDCS24 位至 WDCS20 位用来选择看门狗定时器 2 循环检测间隔时间。

向 WDTE 寄存器写入 ACH 可以清除看门狗定时器 2 的计数值，并且再次开始计数。计数操作开始后，在循环检测时间间隔内将 ACH 写入 WDTE。

如果直到时间间隔时间期满，都没有向 WDTE 寄存器写入 ACH，那么根据 WDM21 和 WDTM2.WDM20 位的设定值，将产生复位信号（WDT2RES）或不可屏蔽中断请求信号（INTWDT2）。

当 WDTM2.WDM21 位置 1（复位模式）时，如果在复位或待机状态解除后的振荡稳定期间发生 WDT 溢出，那么将不发生内部复位，CPU 时钟将切换到内部振荡时钟。

若不使用看门狗定时器 2，则向 WDTM2 寄存器写入 00H。

关于设置为不可屏蔽中断请求模式时的不可屏蔽中断服务，参见 19.2.2（2）自 INTWDT2 的信号。

第十二章 实时输出功能（RTO）

12.1 功能

实时输出功能将预先设置的数据传送至 RTBLn 和 RTBHn 寄存器中，然后在产生定时器中断时，通过硬件输出锁存把数据传送到外部设备中。数据传输到外部设备需要使用的引脚构成一个端口，称为实时输出功能（RTO）。

因为 RTO 能够输出无抖动信号，所以适用于控制步进电机。

在 V850ES/JJ3 单片机中提供两个 6 位的实时输出端口通道。

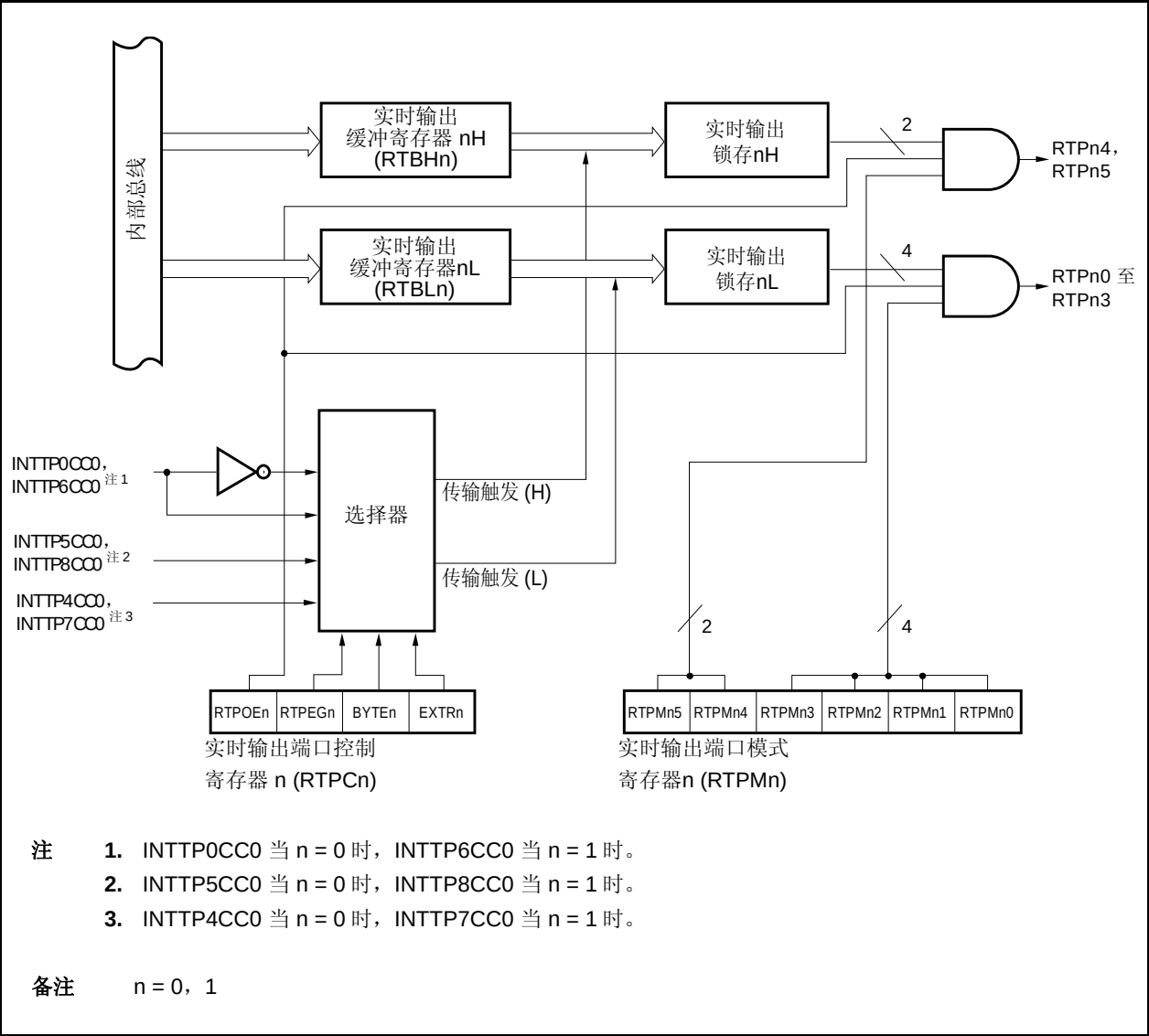
实时输出端口可以设置为端口模式或按位设置为实时输出端口模式。

备注 n = 0, 1

12.2 配置

RTO 的功能框图如下所示：

图 12-1. RTO 的功能框图



RTO 包括下列硬件：

表 12-1. RTO 的配置

项目	配置
寄存器	实时输出缓冲寄存器 nL, nH (RTBLn, RTBHn)
控制寄存器	实时输出端口模式寄存器 n (RTPMn) 实时输出端口控制寄存器 n (RTPCn)

（1）实时输出缓冲寄存器 nL, nH（RTBLn, RTBHn）

RTBLn 和 RTBHn 寄存器是 4 位寄存器，用于保存预设置的输出数据。

这些寄存器被映射到外设 I/O 寄存器区域中的独立地址。

这些寄存器可以按字节或按位进行读写。

复位输入会将这些寄存器设为 00H。

如果指定 4 位×1 通道 或 2 位×1 通道的操作模式（RTPCn.BYTEn 位= 0），数据可以分别设置到 RTBLn 和 RTBHn 寄存器。通过指定这些寄存器各自的地址，可以立即读出寄存器的数据。

如果指定 6 位×1 通道的操作模式（BYTEn 位= 1），向其中任一个寄存器写入数据，都可以把 8 位数据设置到 RTBLn 和 RTBHn 寄存器。此外，通过指定这些寄存器各自的地址，可以立即读出寄存器的数据。

RTBLn 和 RTBHn 寄存器的操作处理过程如表 12-2 所示：

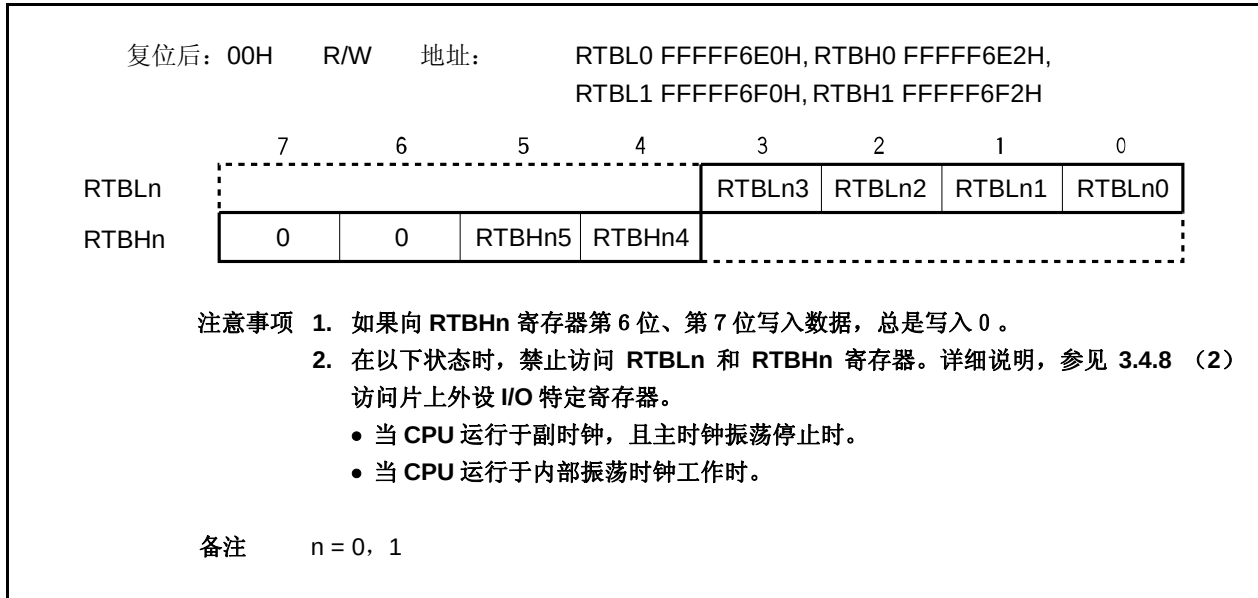


表 12-2. 使用 RTBLn 和 RTBHn 寄存器的处理操作

工作模式	处理的寄存器	读取		写 [*]	
		高 4 位	低 4 位	高 4 位	低 4 位
4 位×1 通道, 2 位×1 通道	RTBLn	RTBHn	RTBLn	无效	RTBLn
	RTBHn	RTBHn	RTBLn	RTBHn	无效
6 位×1 通道	RTBLn	RTBHn	RTBLn	RTBHn	RTBLn
	RTBHn	RTBHn	RTBLn	RTBHn	RTBLn

注 设置为实时输出端口后，在实时输出触发信号产生时，把输出数据设置到 RTBLn 和 RTBHn 寄存器中。

12.3 寄存器

RTO 由以下两个寄存器控制:

- 实时输出端口模式寄存器 n (RTPMn)
- 实时输出端口控制寄存器 n (RTPCn)

(1) 实时输出端口模式寄存器 n (RTPMn)

RTPMn 寄存器用于按位选择实时输出端口模式或端口模式。

该寄存器可以进行字节读写或按位读写。

系统复位后, 该寄存器设为 00H。

复位后: 00H R/W 地址: RTPM0 FFFFF6E4H, RTPM1 FFFFF6F4H

RTPMn (n = 0, 1)	7	6	5	4	3	2	1	0
	0	0	RTPMn5	RTPMn4	RTPMn3	RTPMn2	RTPMn1	RTPMn0
RTPMnm		实时输出端口的控制 (m = 0 至 5)						
0		实时输出禁止						
1		实时输出使能						

- 注意事项
1. 通过使能实时输出操作 (RTPCn.RTPOEn 位 = 1), RTPn0 至 RTPn5 引脚中设置为实时输出的位进行实时输出, 而设置为端口模式的位输出 0。
 2. 如果禁止实时输出操作 (RTPOEn 位= 0), 不管 RTPMn 寄存器如何设置, 实时输出引脚 (RTPn0 至 RTPn5) 都输出 0。
 3. 为了将该寄存器用作实时输出引脚 (RTPn0 至 RTPn5), 需用通过 PMC 和 PFC 寄存器将这些引脚设置为实时输出端口引脚。

（2）实时输出端口控制寄存器 n（RTPCn）

RTPCn 寄存器用于设置操作模式和实时输出端口的输出触发信号。

操作模式和实时输出端口输出触发信号的关系如表 12-3 和 12-4 所示。

该寄存器可以进行字节读写或按位读写。

系统复位后，该寄存器设为 00H。

复位后：00H R/W 地址：RTPC0 FFFF6E5H, RTPC1 FFFF6F5H

RTPCn (n = 0, 1)	<7>	6	5	4	3	2	1	0
	RTPOEn	RTPEGn	BYTEn	EXTRn	0	0	0	0
RTPOEn								
实时输出操作的控制								
0 禁止操作 ^{注 1}								
1 允许操作								
RTPEGn								
INTTPaCC0 (n = 0, a = 0, 4, 5)的有效边沿和 INTTPbCC0 (n = 1, b = 6 至 8) 信号								
0 下降沿 ^{注 2}								
1 上升沿								
BYTEn								
实时输出通道配置的说明								
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（RTPOEn 位 = 0）时，实时输出信号（RTPn0 至 RTPn5）的所有位都输出“0”。
2. INTTP0CC0 和 INTTP6CC0 信号输出一个时钟时间，该时钟是由 TMP0 或 TMP6 选择的计数时钟。

注意事项 仅当 RTPOEn 位 = 0 时，才能设置 RTPEGn、BYTEn 和 EXTRn 位。

表 12-3. 实时输出端口的操作模式和输出触发（n = 0）

BYTE0	EXTR0	工作模式	RTBH0（RTP04, RTP05）	RTBL0（RTP00 至 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	

表 12-4. 实时输出端口的操作模式和输出触发（n = 1）

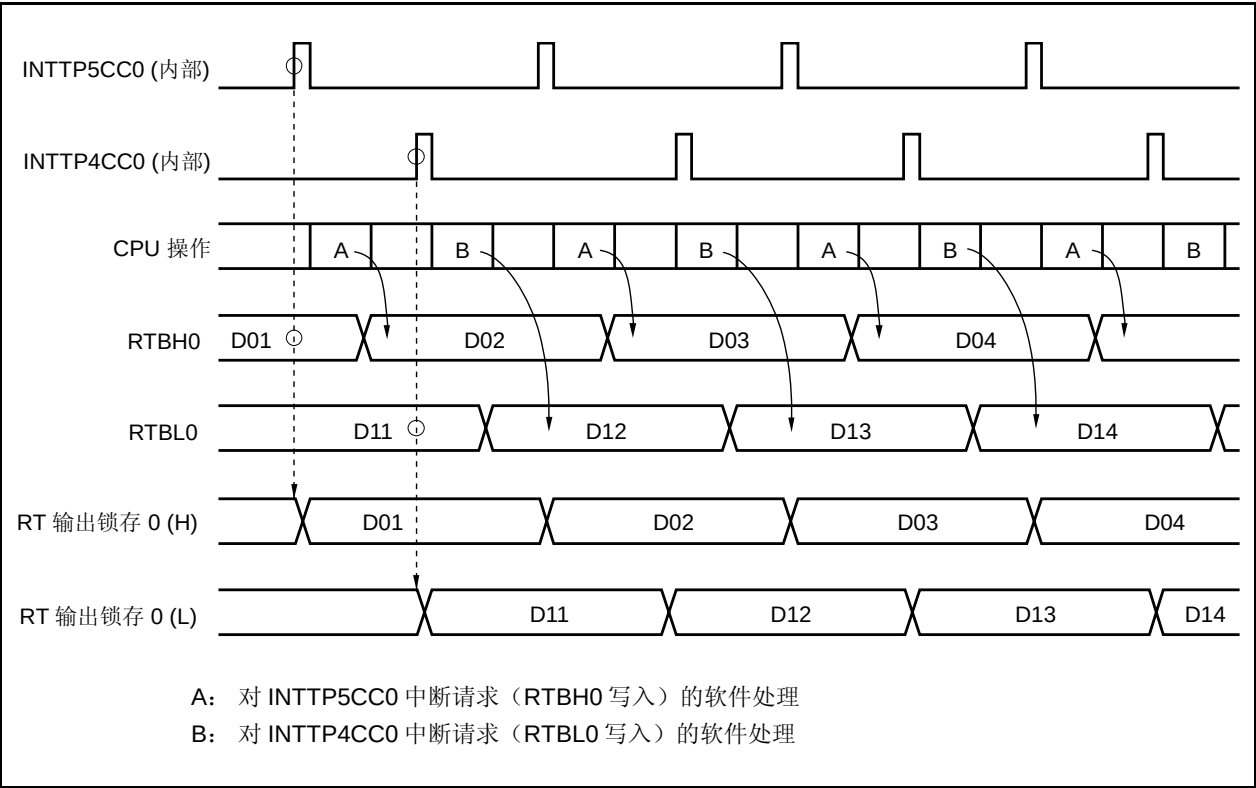
BYTE1	EXTR1	工作模式	RTBH1（RTP14, RTP15）	RTBL1（RTP10 至 RTP13）
0	0	4 位×1 通道,	INTTP8CC0	INTTP7CC0
	1	2 位×1 通道	INTTP7CC0	INTTP6CC0
1	0	6 位×1 通道	INTTP7CC0	
	1		INTTP6CC0	

12.4 操作

通过设置 RTPCn.RTPOEn 位为 1，使能实时输出操作，在选定的传输触发信号产生（由 RTPCn.EXTRn 和 RTPCn.BYTEn 位设置）时，RTBHn 和 RTBLn 寄存器的数据被同步传输到实时输出锁存器中。在已传输的数据中，只有那些在 RTPMn 寄存器中设置为允许实时输出的位，其数据才会从 RTPn0 至 RTPn5 位输出。RTPMn 寄存器中设置为禁止实时输出的位，输出 0。

通过将 RTPOEn 位清为 0，禁止实时输出操作，则不管 RTPMn 寄存器如何设置，RTPn0 至 RTPn5 信号都输出 0。

图 12-2. RTO0 的操作时序示例（当 EXTR0= 0，BYTE0 位= 0）



备注 关于待机期间的操作，参见第二十一章 待机功能。

12.5 用法

（1）禁止实时输出

将 RTPCn.RTPOEn 位清为 0。

（2）执行下列初始化操作。

- 设置端口 5 的复用功能引脚
将 PFC5.PFC5m 和 PFCE5.PFCE5m 位置为 1，然后将 PMC5.PMC5m 位置为 1（m = 0 至 5）。
- 按位指定实时输出端口模式或端口模式。
设置 RTPMn 寄存器。
- 通道配置：选择触发信号和有效沿。
设置 RTPCn.EXTRn、RTPCn.BYTEn 和 RTPCn.RTPEGn 位。
- 设置 RTBHn 和 RTBLn 寄存器^{注 1}的初始值。

（3）实时输出使能

设置 RTPOEn 位 = 1。

（4）在选择的传送触发信号产生^{注 2}时，把下一个输出值设置到 RTBHn 和 RTBLn 寄存器。

（5）通过所选触发信号的中断服务程序，把下一个实时输出值设置到 RTBHn 和 RTBLn 寄存器。

- 注**
1. 当 RTPOEn 位 = 0 时，如果写入 RTBHn 和 RTBLn 寄存器，则该值分别传送到实时输出锁存 nH 和 nL。
 2. 当 RTPOEn 位 = 1 时，即使写入 RTBHn 和 RTBLn 寄存器，则数据不传送到实时输出锁存 nH 和 nL。

12.6 注意事项

（1）通过软件避免以下的冲突：

- 实时输出禁止/使能切换（RTPOEn 位）和选择的实时输出输出触发信号之间的冲突。
- 在实时输出状态下对 RTBHn 和 RTBLn 寄存器的写入操作和选择的实时输出触发信号之间的冲突。

（2）在执行初始化之前，禁止实时输出（RTPOEn 位 = 0）。

（3）一旦实时输出被禁止（RTPOEn 位 = 0），确保重新使能实时输出（RTPOEn 位 = 0 → 1）之前，对 RTBHn 和 RTBLn 寄存器初始化。

第十三章 A/D 转换器

13.1 概述

A/D 转换器可以将模拟输入信号转换为数字值，具有 10 位分辨率，具有 16 路模拟输入信号通道（AN10 至 AN15）。

A/D 转换器具有下列特征：

- 10 位分辨率。
- 16 通道
- 逐次逼近法
- 工作电压： $AV_{REF0} = 3.0$ 至 3.6 V 。
- 模拟输入电压： 0 V 至 AV_{REF0}
- 操作模式具有下列功能：
 - 连续选择模式
 - 连续扫描模式
 - 单次选择模式
 - 单次扫描模式
- 触发模式具有下列功能：
 - 软件触发模式
 - 外部触发模式（外部，1）
 - 定时器触发模式
- 掉电监控功能（转换结果比较功能）

13.2 功能

（1）10 位分辨率的 A/D 转换

从 AN10 至 AN15 通道中选择模拟输入通道，且 A/D 转换器以 10 位分辨率进行反复操作。每次 A/D 转换完成后，都会产生一个中断请求信号（INTAD）。

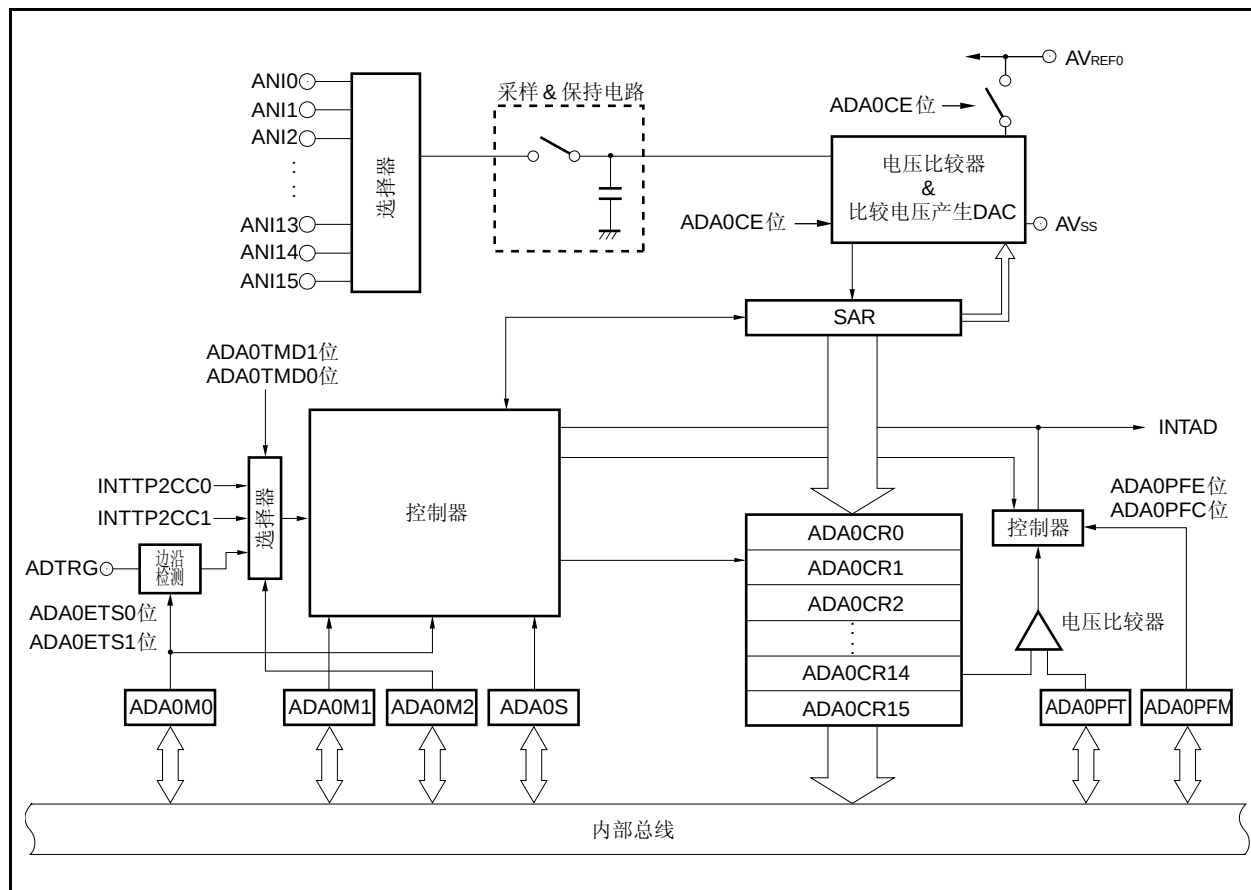
（2）掉电检测功能

此功能用来检测电池电压的下降。将 A/D 转换结果（ADA0CRnH 寄存器的值）和 ADA0PFT 寄存器值进行比较，仅当满足指定的比较条件（ $n = 0$ 至 15 ）时，才会产生 INTAD 信号。

13.3 配置

A/D 转换器的框图如下所示:

图 13-1. A/D 转换器框图



A/D 转换器包括下列硬件:

表 13-1. A/D 转换器的配置

项目	配置
模拟输入	16 个通道（AN10 引脚至 AN15 引脚）
寄存器	逐次逼近寄存器（SAR） A/D 转换结果寄存器 0 至 15（ADA0CR0 至 ADA0CR15） A/D 转换结果寄存器 0H 至 15H（ADCR0H 至 ADCR15H）： 仅能读取高 8 位
控制寄存器	A/D 转换器模式寄存器 0 至 2（ADA0M0 至 ADA0M2） A/D 转换器通道指定寄存器 0（ADA0S） 掉电比较模式寄存器（ADA0PFM） 掉电比较阈值寄存器（ADA0PFT）

(1) 逐次逼近寄存器 (SAR)

SAR 寄存器将模拟输入信号的电压值与比较电压产生 DAC (比较电压) 的输出值进行比较, 并且从最高有效位 (MSB) 开始保存比较结果。

当比较结果已经保存至最低有效位 (LSB) (也就是 A/D 转换完成) 时, SAR 寄存器的内容将被传送到 ADA0CRn 寄存器。

备注 n = 0 至 15

(2) A/D 转换结果寄存器 n (ADA0CRn), A/D 转换结果寄存器 nH (ADA0CRnH)

ADA0CRn 寄存器是一个存储 A/D 转换结果的 16 位寄存器。ADA0ARn 由 16 个寄存器组成, 并且 A/D 转换结果存储在对应于模拟输入的 AD0CRn 寄存器高 10 位。(低 6 位固定为 0。)

(3) A/D 转换器模式寄存器 0 (ADA0M0)

该寄存器用于指定操作模式并且控制 A/D 转换器的转换操作。

(4) A/D 转换器模式寄存器 1 (ADA0M1)

该寄存器用于设置待转换的模拟输入信号的转换时间。

(5) A/D 转换器模式寄存器 2 (ADA0M2)

该寄存器用于设置硬件触发模式。

(6) A/D 转换器通道指定寄存器 (ADA0S)

该寄存器用于设置输入端口, 该端口用于输入待转换的模拟电压。

(7) 掉电比较模式寄存器 (ADA0PFM)

该寄存器用于设置掉电监控模式。

(8) 掉电比较阈值寄存器 (ADA0PFT)

ADA0PFT 寄存器用于设置阈值, 和 A/D 转换结果寄存器 nH (ADA0CRnH) 的值进行比较。ADA0PFT 寄存器的 8 位数据与 A/D 转换结果寄存器 (ADA0CRnH) 的高 8 位进行比较。

(9) 控制器

当 A/D 转换完成时或使用掉电检测功能时, A/D 控制器将转换结果 (ADA0CRnH 寄存器的值) 与 ADA0PFT 寄存器的值进行比较, 当满足指定的比较条件时, 产生 INTAD 信号。

(10) 采样&保持电路

采样保持电路对输入电路选择的各个模拟输入信号进行采集, 并将采样数据发送到电压比较器。在 A/D 转换期间, 该电路保持采样得到的模拟输入信号电压。

(11) 电压比较器

电压比较器将采样保持电路得到的电压值与比较电压产生 DAC 的输出电压值进行比较。

(12) 比较电压产生 DAC

比较电压产生 DAC 连接在 AV_{REF0} 和 AV_{SS} 之间，并产生用于和模拟输入信号进行比较的电压。

(13) ANI0 至 ANI15 引脚

这些引脚是 16 路 A/D 转换通道的模拟输入引脚，用于输入将要转换为数字信号的模拟信号。除了 ADA0S 寄存器选定的模拟输入引脚外，其他引脚都可以用作输入端口引脚。

注意事项 确保输入 ANI0 引脚至 ANI15 引脚的电压没有超过额定值。如果输入至某通道的电压值为 AV_{REF0} 电压或者更高，那么此通道的转换值不确定，并且其它通道的转换值也会受到影响。

(14) AV_{REF0} 引脚

该引脚用于输入 A/D 转换器的参考电压。即使不使用 A/D 转换器，也要让该引脚的电压与 V_{DD} 引脚的电压相同。根据施加在 AV_{REF0} 引脚和 AV_{SS} 引脚之间的电压，将输入到 ANI0 至 ANI15 引脚的信号转化为数字信号。

(15) AV_{SS} 引脚

该引脚为 A/D 转换器的地引脚。即使不使用 A/D 转换器，也要保证该引脚的电压与 V_{SS} 引脚的电压相同。

13.4 寄存器

A/D 转换器由下列寄存器控制：

- A/D 转换器模式寄存器 0、1、2（ADA0M0、ADA0M1、ADA0M2）
- A/D 转换器通道指定寄存器 0（ADA0S）
- 掉电比较模式寄存器（ADA0PFM）

以下寄存器也可以使用：

- A/D 转换结果寄存器 n（ADA0CRn）
- A/D 转换结果寄存器 nH（ADA0CRnH）
- 掉电比较阈值寄存器（ADA0PFT）

(1) A/D 转换器模式寄存器 0（ADA0M0）

ADA0M0 寄存器是 8 位寄存器，用于指定操作模式并控制转换操作。
该寄存器可进行字节读写或按位读写。但是，ADA0EF 0 是只读位。
系统复位后，该寄存器设为 00H。

(1/2)

复位后: 00H R/W 地址: FFFFF200H

	<7>	6	5	4	3	2	1	<0>
ADA0M0	ADA0CE	0	ADA0MD1	ADA0MD0	ADA0ETS1	ADA0ETS0	ADA0TMD	ADA0EF

ADA0CE	A/D 转换控制	
0	停止 A/D 转换	
1	使能 A/D 转换	

ADA0MD1	ADA0MD0	A/D转换器操作模式说明
0	0	连续选择模式
0	1	连续扫描模式
1	0	单次选择模式
1	1	单次扫描模式

ADA0ETS1	ADA0ETS0	外部触发(ADTRG 引脚) 输入有效边沿的说明
0	0	无边沿检测
0	1	下降沿检测
1	0	上升沿检测
1	1	双边沿检测

ADA0TMD	触发模式说明
0	软件触发模式
1	外部触发模式/定时器触发模式

ADA0EF	A/D 转换器状态说明
0	A/D 转换停止
1	A/D 转换进行中

注意事项 1. 下列状态下禁止访问 ADA0M0 寄存器。详细说明，参见 3.4.8 (2) 访问片上外设 I/O 专用寄存器。

- 当 CPU 运行于副时钟，且主时钟振荡停止时。
 - 当 CPU 运行于内部振荡时钟工作时。
2. 对第 0 位的写入操作将忽略。
3. A/D 转换使能 (ADA0CE 位 = 1) 时，禁止改变 ADA0M1.ADA0FR2 位至 ADA0M1.ADA0FR0 位。
4. 下列模式中向 ADA0M0、ADA0M2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器写入数据，通过将 ADA0CE 位清除为 0 以停止 A/D 转换。在数据已经写入到寄存器之后，将 ADA0CE 位设置为 1，再次使能 A/D 转换操作。
- 正常转换模式下
 - 高速转换模式下的单次选择模式/单次扫描模式
- A/D 转换期间 (ADA0EF 位 = 1)，如果在其它模式下写入 ADA0M0、ADA0M2、ADA0S、ADA0PFM 和 ADA0PFT 寄存器，那么根据模式将会执行如下操作。
- 软件触发模式下：
 - A/D 转换停止，并再次从头开始执行。
 - 硬件触发模式下：
 - A/D 转换停止，且设置为触发等待状态。
5. 为了选择外部触发模式/定时器触发模式 (ADA0TMD 位 = 1)，设置为高速转换模式 (ADA0M1.ADA0HS1 位 = 1)。A/D 转换操作使能 (ADA0CE 位 = 1) 后，在随后插入的振荡稳定时间期间，不要输入触发信号。
6. 当不使用 A/D 转换器时，将 ADA0CE 位设置为 0，停止转换操作，降低功耗。

(2) A/D 转换器模式寄存器 1 (ADA0M1)

ADA0M1 是 8 位寄存器，用于指定转换时间。
该寄存器可以按字节读写或按位读写。
系统复位后，该寄存器设为 00H。

复位后： 00H R/W 地址： FFFFF201H

	7	6	5	4	3	2	1	0
ADA0M1	ADA0HS1	0	0	0	ADA0FR3	ADA0FR2	ADA0FR1	ADA0FR0
	ADA0HS1	正常转换模式/高速模式说明(A/D转换时间)						
	0	正常转换模式						
	1	高速转换模式						

- 注意事项
1. A/D 转换使能 (ADA0M0.ADA0CE 位= 1) 时，禁止改变 ADA0M1 寄存器。
 2. 为了选择外部触发模式/定时器触发模式 (ADA0M0.ADA0TMD 位= 1)，设置高速转换模式 (ADA0HS1 位= 1)。A/D 转换操作使能 (ADA0CE 位= 1) 后，在随后插入的振荡稳定时间期间，不要输入触发信号。
 3. 请确保将第 6 位至第 4 位清为“0”。

备注 关于设置 A/D 转换时间的举例，参见表 13-2 和表 13-3。

<R>

表 13-2. 正常转换模式下转换时间的选择 (ADA0HS1 位= 0)

ADA0FR3 至 ADA0FR0 位	A/D 转换时间					
	稳定时间+转换时间+等待时间	$f_{xx} = 32 \text{ MHz}$	$f_{xx} = 20 \text{ MHz}$	$f_{xx} = 16 \text{ MHz}$	$f_{xx} = 4 \text{ MHz}$	触发响应时间
0000	$66/f_{xx} (13/f_{xx} + 26/f_{xx} + 27/f_{xx})$	禁止设置	禁止设置	禁止设置	$16.50 \mu\text{s}$	$3/f_{xx}$
0001	$131/f_{xx} (26/f_{xx} + 52/f_{xx} + 53/f_{xx})$	禁止设置	$6.55 \mu\text{s}$	$8.19 \mu\text{s}$	禁止设置	$3/f_{xx}$
0010	$196/f_{xx} (39/f_{xx} + 78/f_{xx} + 79/f_{xx})$	禁止设置	$9.80 \mu\text{s}$	$12.25 \mu\text{s}$	禁止设置	$3/f_{xx}$
0011	$259/f_{xx} (50/f_{xx} + 104/f_{xx} + 105/f_{xx})$	$8.09 \mu\text{s}$	$12.95 \mu\text{s}$	$16.19 \mu\text{s}$	禁止设置	$3/f_{xx}$
0100	$311/f_{xx} (50/f_{xx} + 130/f_{xx} + 131/f_{xx})$	$9.72 \mu\text{s}$	$15.55 \mu\text{s}$	$19.44 \mu\text{s}$	禁止设置	$3/f_{xx}$
0101	$363/f_{xx} (50/f_{xx} + 156/f_{xx} + 157/f_{xx})$	$11.34 \mu\text{s}$	$18.15 \mu\text{s}$	$22.69 \mu\text{s}$	禁止设置	$3/f_{xx}$
0110	$415/f_{xx} (50/f_{xx} + 182/f_{xx} + 183/f_{xx})$	$12.97 \mu\text{s}$	$20.75 \mu\text{s}$	禁止设置	禁止设置	$3/f_{xx}$
0111	$467/f_{xx} (50/f_{xx} + 208/f_{xx} + 209/f_{xx})$	$14.59 \mu\text{s}$	$23.35 \mu\text{s}$	禁止设置	禁止设置	$3/f_{xx}$
1000	$519/f_{xx} (50/f_{xx} + 234/f_{xx} + 235/f_{xx})$	$16.22 \mu\text{s}$	禁止设置	禁止设置	禁止设置	$3/f_{xx}$
1001	$571/f_{xx} (50/f_{xx} + 260/f_{xx} + 261/f_{xx})$	$17.84 \mu\text{s}$	禁止设置	禁止设置	禁止设置	$3/f_{xx}$
1010	$623/f_{xx} (50/f_{xx} + 286/f_{xx} + 287/f_{xx})$	$19.47 \mu\text{s}$	禁止设置	禁止设置	禁止设置	$3/f_{xx}$
1011	$675/f_{xx} (50/f_{xx} + 312/f_{xx} + 313/f_{xx})$	$21.09 \mu\text{s}$	禁止设置	禁止设置	禁止设置	$3/f_{xx}$
其它情况	禁止设置					

备注

稳定时间: A/D 转换建立时间 ($1 \mu\text{s}$ 或更长)

转换时间: 实际的 A/D 转换时间 (2.6 至 $10.4 \mu\text{s}$)

等待时间: 下一次转换之前, 插入的等待时间

触发响应时间: 如果在稳定时间之后产生软件触发、外部触发或者定时器触发, 那么在转换时间之前插入触发响应时间。

在正常转换模式下, 自 ADA0M0.ADA0CE 位置为 1, 稳定时间过后, 开始转换, 并且 A/D 转换仅在转换时间 (2.6 至 $10.4 \mu\text{s}$) 执行。转换结束后操作停止, 等待时间过后, 产生 A/D 转换结束中断请求信号 (INTAD)。

因为转换操作在等待时间内停止, 所以降低了工作电流。

注意事项 1. 设置为 $2.6 \mu\text{s} \leq \text{转换时间} \leq 10.4 \mu\text{s}$

2. A/D 转换期间, 如果有数据写入 ADA0M0、ADA0M2、ADA0S、ADA0PFM 以及 ADA0PFT 寄存器或者有触发信号输入, 则重新执行转换。但是, 如果稳定时间结束时序与写入这些寄存器的操作发生冲突, 或者稳定时间结束时序与触发信号输入发生冲突, 那么将会重新插入 64 个时钟的稳定时间。

如果在插入的稳定时间结束时序时刻再次发生冲突, 则仍然重新插入稳定时间。因此, 不要将触发输入间隔和控制寄存器写入间隔设置为 64 个时钟或更少。

<R>

表 13-3. 高速转换模式下转换时间的选择 (ADA0HS1 位= 1)

ADA0FR3 至 ADA0FR0 位	A/D 转换时间					
	转换时间 (+ 稳定时间)	f _{xx} = 32 MHz	f _{xx} = 20 MHz	f _{xx} = 16 MHz	f _{xx} = 4 MHz	触发响 应时间
0000	26/f _{xx} (+13/f _{xx})	禁止设置	禁止设置	禁止设置	6.5 μ s (+3.25 μ s)	3/f _{xx}
0001	52/f _{xx} (+26/f _{xx})	禁止设置	2.6 μ s (+1.3 μ s)	3.25 μ s (+1.625 μ s)	禁止设置	3/f _{xx}
0010	78/f _{xx} (+39/f _{xx})	禁止设置	3.9 μ s (+1.95 μ s)	4.875 μ s (+2.4375 μ s)	禁止设置	3/f _{xx}
0011	104/f _{xx} (+50/f _{xx})	3.25 μ s (+1.5625 μ s)	5.2 μ s (+2.5 μ s)	6.5 μ s (+3.125 μ s)	禁止设置	3/f _{xx}
0100	130/f _{xx} (+50/f _{xx})	4.0625 μ s (+1.5625 μ s)	6.5 μ s (+2.55 μ s)	8.125 μ s (+3.125 μ s)	禁止设置	3/f _{xx}
0101	156/f _{xx} (+50/f _{xx})	4.875 μ s (+1.5625 μ s)	7.8 μ s (+2.5 μ s)	9.75 μ s (+3.125 μ s)	禁止设置	3/f _{xx}
0110	182/f _{xx} (+50/f _{xx})	5.6875 μ s (+1.5625 μ s)	9.1 μ s (+2.5 μ s)	禁止设置	禁止设置	3/f _{xx}
0111	208/f _{xx} (+50/f _{xx})	6.5 μ s (+1.5625 μ s)	10.4 μ s (+2.5 μ s)	禁止设置	禁止设置	3/f _{xx}
1000	234/f _{xx} (+50/f _{xx})	7.3125 μ s (+1.5625 μ s)	禁止设置	禁止设置	禁止设置	3/f _{xx}
1001	260/f _{xx} (+50/f _{xx})	8.125 μ s (+1.5625 μ s)	禁止设置	禁止设置	禁止设置	3/f _{xx}
1010	286/f _{xx} (+50/f _{xx})	8.9375 μ s (+1.5625 μ s)	禁止设置	禁止设置	禁止设置	3/f _{xx}
1011	312/f _{xx} (+50/f _{xx})	9.75 μ s (+1.5625 μ s)	禁止设置	禁止设置	禁止设置	3/f _{xx}
其它情况	禁止设置					

备注 转换时间： 实际的 A/D 转换时间 (2.6 至 10.4 μ s)

 稳定时间： A/D 转换建立时间 (1 μ s 或更长)

 触发响应时间： 如果在稳定时间之后产生软件触发、外部触发或者定时器触发，那么在转换时间之前插入触发响应时间。

在高速转换模式下，自 ADA0M0.ADA0CE 位置为 1，稳定时间过后，开始转换，并且 A/D 转换仅在转换时间 (2.6 至 10.4 μ s) 执行。转换结束后立即产生 A/D 转换结束中断请求信号 (INTAD)。

在连续转换模式下，仅在第一次转换前插入稳定时间，而在第二次转换之后 (A/D 转换器仍然运行) 不会插入。

- 注意事项** 1. 设置为 2.6 μ s ≤ 转换时间 ≤ 10.4 μ s
2. 在高速转换模式下，振荡稳定期间，禁止触发信号输入和对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 和 ADA0PFT 寄存器的重写。

(3) A/D 转换器模式寄存器 2 (ADA0M2)

ADA0M2 寄存器用于指定硬件触发模式。
该寄存器可以按字节读写或按位读写。
系统复位后，该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF203H

	7	6	5	4	3	2	1	0
ADA0M2	0	0	0	0	0	0	ADA0TMD1	ADA0TMD0

ADA0TMD1	ADA0TMD0	硬件触发模式说明
0	0	外部触发模式（检测到ADTRG引脚有效边沿时）
0	1	定时器触发模式 0 (产生INTTP2CC0 中断请求时)
1	0	定时器触发模式 1 (产生INTTP2CC1中断请求时)
1	1	禁止设置

- 注意事项
- 下列模式中向 ADA0M2 寄存器写入数据，通过将 AD0M0.ADA0CE 位清除为 0 以停止 A/D 转换。在数据已经写入到该寄存器之后，将 ADA0CE 位设置为 1，再次使能 A/D 转换操作。
 - 正常转换模式
 - 高速转换模式下的单次选择模式/单次扫描模式
 - 请确保将第 7 位至第 2 位清除为“0”。

(4) A/D 转换输入通道指定寄存器 0 (ADA0S)

ADA0S 寄存器用于指定将模拟电压转换为数字信号的输入引脚。

该寄存器可以按字节或按位进行读写。

系统复位后，该寄存器设为 00H。

复位后：00H R/W 地址：FFFF202H

	7	6	5	4	3	2	1	0
ADA0S	0	0	0	0	ADA0S3	ADA0S2	ADA0S1	ADA0S0

ADA0S3	ADA0S2	ADA0S1	ADA0S0	选择模式	扫描模式
0	0	0	0	ANI0	ANI0
0	0	0	1	ANI1	ANI0, ANI1
0	0	1	0	ANI2	ANI0 至 ANI2
0	0	1	1	ANI3	ANI0 至 ANI3
0	1	0	0	ANI4	ANI0 至 ANI4
0	1	0	1	ANI5	ANI0 至 ANI5
0	1	1	0	ANI6	ANI0 至 ANI6
0	1	1	1	ANI7	ANI0 至 ANI7
1	0	0	0	ANI8	ANI0 至 ANI8
1	0	0	1	ANI9	ANI0 至 ANI9
1	0	1	0	ANI10	ANI0 至 ANI10
1	0	1	1	ANI11	ANI0 至 ANI11
1	1	0	0	ANI12	ANI0 至 ANI12
1	1	0	1	ANI13	ANI0 至 ANI13
1	1	1	0	ANI14	ANI0 至 ANI14
1	1	1	1	ANI15	ANI0 至 ANI15

注意事项 1. 下列模式中向 ADA0S 寄存器写入数据，通过将 AD0M0.ADA0CE 位清除为 0 以停止 A/D 转换。在数据已经写入到该寄存器之后，将 ADA0CE 位设置为 1，再次使能 A/D 转换操作。

- 正常转换模式
- 高速转换模式下的单次选择模式/单次扫描模式

2. 请确保将第 7 位至第 4 位清为“0”。

(5) A/D 转换结果寄存器 n、nH (ADA0CRn, ADA0CRnH)

ADA0CRn 寄存器和 ADA0CRnH 寄存器用于存储 A/D 转换结果。

这些寄存器是以 8 位或 16 位宽度访问的只读寄存器。但是，指定 ADA0CRn 寄存器为 16 位访问，指定 ADA0CRnH 寄存器为 8 位访问。从 ADA0CRn 寄存器的高 10 位读取 10 位转换结果，同时低 6 位的读取值为 0。从 ADA0CRnH 寄存器读取高 8 位转换结果。

注意事项 下列状态下，禁止访问 ADA0CRn 寄存器和 ADA0CRnH 寄存器。

详细说明，参见 3.4.8 (2) 访问片上外设 I/O 专用寄存器。

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

复位后：不确定 R 地址： ADA0CR0 FFFFF210H, ADA0CR1 FFFFF212H,
ADA0CR2 FFFFF214H, ADA0CR3 FFFFF216H,
ADA0CR4 FFFFF218H, ADA0CR5 FFFFF21AH,
ADA0CR6 FFFFF21CH, ADA0CR7 FFFFF21EH,
ADA0CR8 FFFFF220H, ADA0CR9 FFFFF222H,
ADA0CR10 FFFFF224H, ADA0CR11 FFFFF226H,
ADA0CR12 FFFFF228H, ADA0CR13 FFFFF22AH,
ADA0CR14 FFFFF22CH, ADA0CR15 FFFFF22EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADA0CRn (n = 0 至 15)	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

复位后：不确定 R 地址： ADA0CR0H FFFFF211H, ADA0CR1H FFFFF213H,
ADA0CR2H FFFFF215H, ADA0CR3H FFFFF217H,
ADA0CR4H FFFFF219H, ADA0CR5H FFFFF21BH,
ADA0CR6H FFFFF21DH, ADA0CR7H FFFFF21FH,
ADA0CR8H FFFFF221H, ADA0CR9H FFFFF223H,
ADA0CR10H FFFFF225H, ADA0CR11H FFFFF227H,
ADA0CR12H FFFFF229H, ADA0CR13H FFFFF22BH,
ADA0CR14H FFFFF22DH, ADA0CR15H FFFFF22FH

	7	6	5	4	3	2	1	0
ADA0CRnH (n = 0 至 15)	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

注意事项 对 ADA0M0 寄存器和 ADA0S 寄存器的写入操作可能会导致 ADA0CRn 寄存器的内容不确定。转换完成后，在写入 ADA0M0 寄存器和 ADA0S 寄存器之前读取转换结果。如果没有按上述顺序执行，则读取结果可能会有误。

输入到模拟输入引脚（ANI0 至 ANI15）的模拟电压与转换结果（ADA0CRn 寄存器）之间的关系如下：

$$SAR = INT \left(\frac{V_{IN}}{AV_{REF0}} \times 1,024 + 0.5 \right)$$

$$ADA0CR^{\#} = SAR \times 64$$

或，

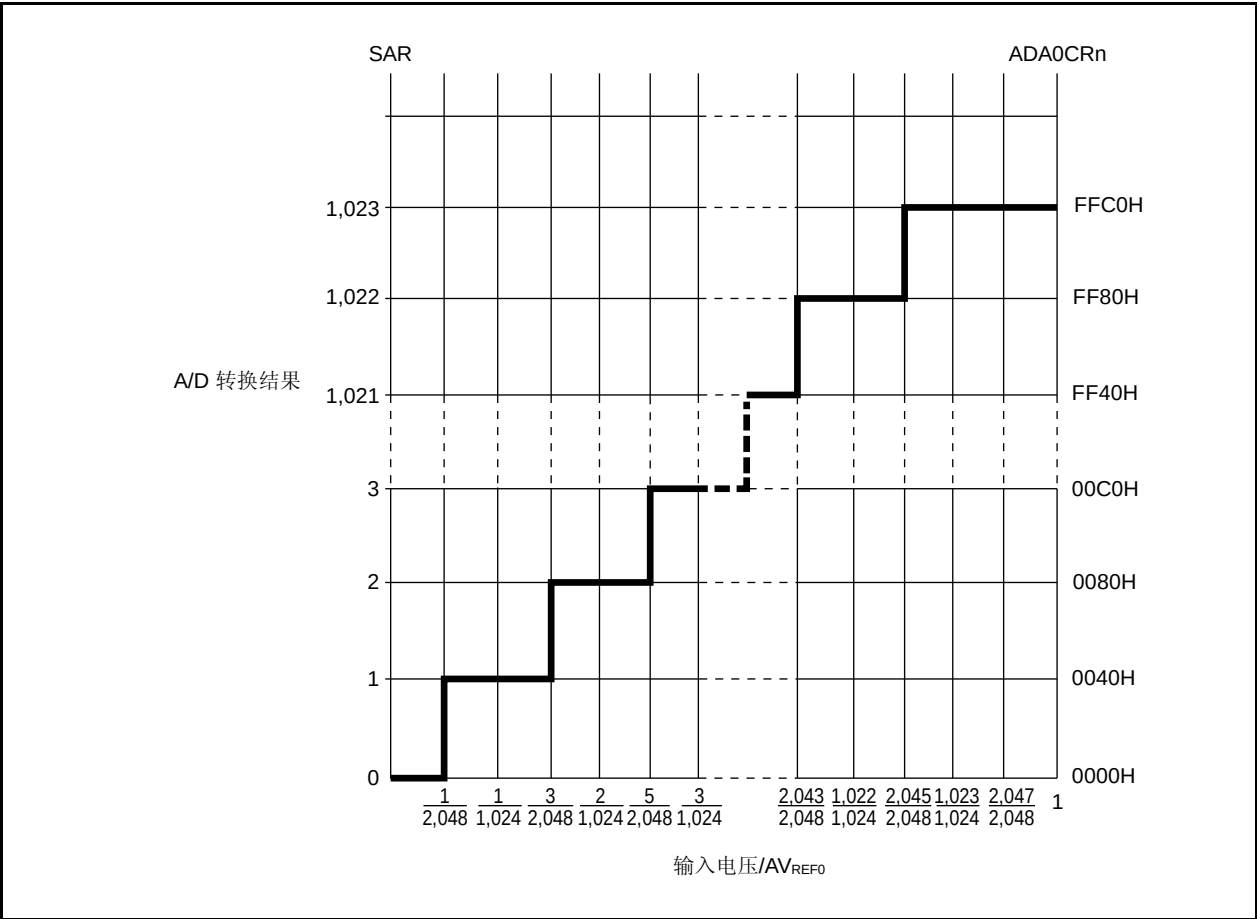
$$(SAR - 0.5) \times \frac{AV_{REF0}}{1,024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{REF0}}{1,024}$$

INT ()： 返回括号内数值整数部分的函数
VIN： 模拟输入电压
AVREF0： AVREF0 引脚电压
ADA0CR： ADA0CRn 寄存器的值

注 ADA0CRn 寄存器的低 6 位固定为 0。

下图显示了模拟输入电压和 A/D 转换结果的之间关系。

图 13-2. 模拟输入电压和 A/D 转换结果的关系



(6) 掉电比较模式寄存器 (ADA0PFM)

ADA0PFM 寄存器是 8 位寄存器，用于设置掉电比较模式。
该寄存器可按字节或按位进行读写。
系统复位后，该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF204H

	<7>	6	5	4	3	2	1	0
ADA0PFM	ADA0PFE	ADA0PFC	0	0	0	0	0	0

ADA0PFE	掉电比较使能/禁止的选择
0	掉电比较禁止
1	掉电比较使能

ADA0PFC	掉电比较模式的选择
0	当 $ADA0CRnH \geq ADA0PFT$ 时，产生中断请求信号 (INTAD)
1	当 $ADA0CRnH < ADA0PFT$ 时，产生中断请求信号 (INTAD)

- 注意事项
1. 在选择模式下，将设置到 ADA0PFT 寄存器的 8 位数据与 ADA0S 寄存器指定的 ADA0CRnH 寄存器的值进行比较。如果该结果符合 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CRn 寄存器，并且产生 INTAD 信号。但是，如果不符合条件，将不产生中断信号。
 2. 在扫描模式下，将设置到 ADA0PFT 寄存器的 8 位数据与 ADA0CR0H 寄存器的内容进行比较。如果该结果符合 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CR0 寄存器中，并且产生 INTAD 信号。但是，如果不符合条件，将不产生 INTAD 信号。无论比较结果如何，扫描操作完成之前，都会继续扫描，并且将转换结果存储在 ADA0CRn 寄存器中。但是，扫描操作完成后不产生 INTAD 信号。
 3. 下列模式中向 ADA0PFM 寄存器写入数据，通过将 AD0M0.ADA0CE 位清除为 0 以停止 A/D 转换。在数据已经被写入到该寄存器之后，将 ADA0CE 位设置为 1，再次使能 A/D 转换操作。
 - 正常转换模式
 - 高速转换模式下的单次选择模式/单次扫描模式

(7) 掉电比较阈值寄存器 (ADA0PFT)

ADA0PFT 寄存器用于设置掉电比较模式下的比较值。
该寄存器可以按字节或按位进行读写。
系统复位后，该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF205H

ADA0PFT

7	6	5	4	3	2	1	0

注意事项 下列模式中向 ADA0PFT 寄存器写入数据，通过将 AD0M0.ADA0CE 位清除为 0 以停止 A/D 转换。在数据已经写入到该寄存器之后，将 ADA0CE 位设置为 1，再次使能 A/D 转换操作。

- 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

13.5 操作

13.5.1 基本操作

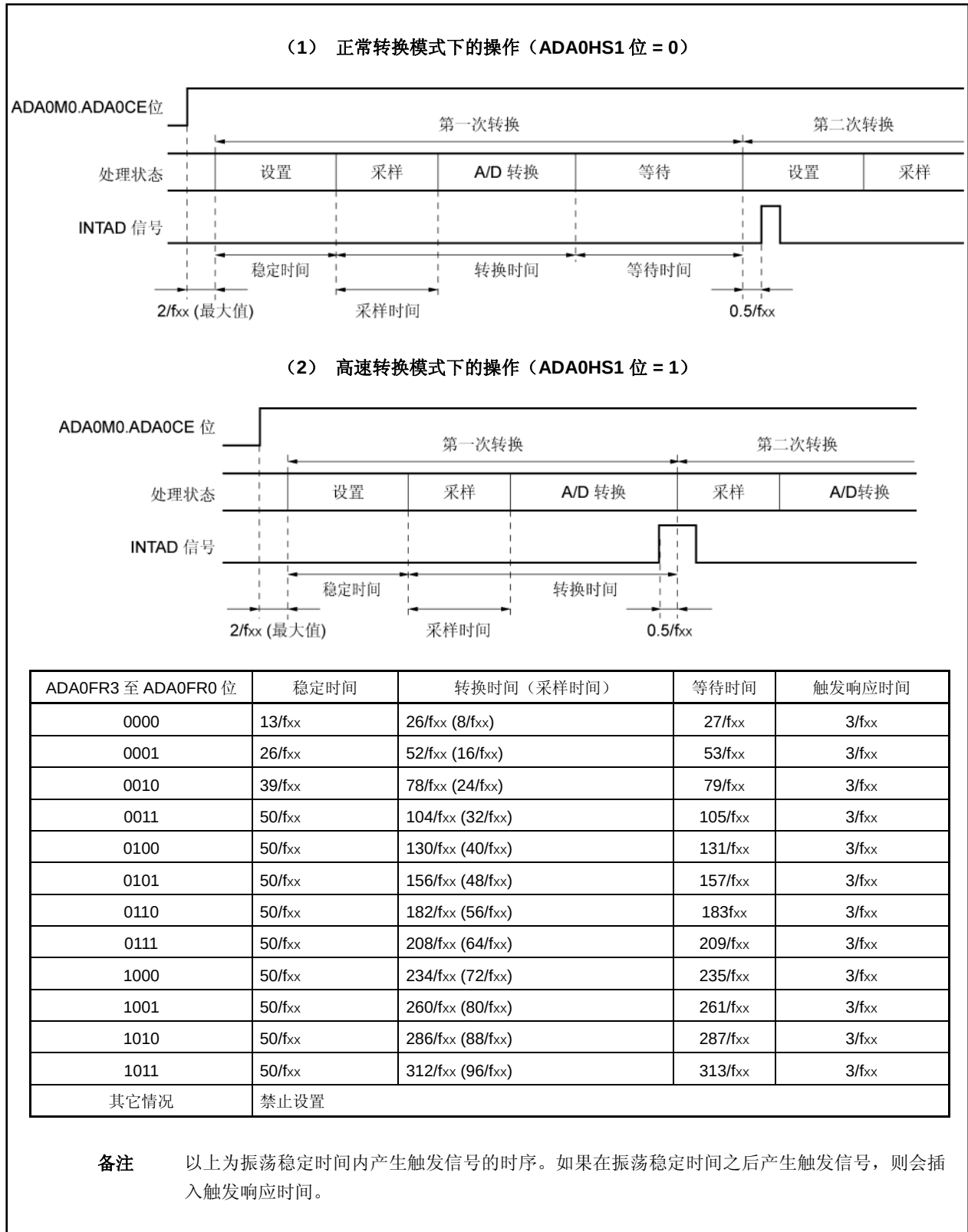
- <1> 使用 ADA0M0、ADA0M1、ADA0M2 和 ADA0S 寄存器来设置执行 A/D 转换的操作模式、触发模式以及转换时间。当置位 ADA0M0 寄存器的 ADA0CE 位时，在软件触发模式下转换开始，在外部或定时器触发模式下 A/D 转换器等待触发信号。
- <2> A/D 转换开始后，采样&保持电路对所选模拟输入通道的输入电压进行采样。
- <3> 采样&保持电路在指定时间内对输入通道进行采样，然后采样&保持电路进入保持状态，并且保持采样得到的输入模拟电压，直到 A/D 转换完成。
- <4> 将逐次逼近寄存器（SAR）的第 9 位置位，设置比较电压产生 DAC 为 $(1/2) AV_{REF0}$ 。
- <5> 电压比较器对比较电压产生 DAC 的电压和模拟输入电压之间的电压差进行比较。如果模拟输入电压高于 $(1/2) AV_{REF0}$ ，SAR 寄存器的 MSB 保持置位。如果模拟输入电压低于 $(1/2) AV_{REF0}$ ，MSB 位清零。
- <6> 接着，自动将逐次逼近寄存器（SAR）的第 8 位置位，并启动下一次比较。根据先前所设定的第 9 位的值，比较电压产生 DAC 的选择如下。
 - 第 9 位 = 1: $(3/4) AV_{REF0}$
 - 第 9 位 = 0: $(1/4) AV_{REF0}$
 该电压和模拟输入电压比较，并且根据比较结果，对第 8 位的操作如下：
 模拟输入电压 $\geq$ 比较电压：第 8 位 = 1
 模拟输入电压 $\leq$ 比较电压：第 8 位 = 0
- <7> 此项比较一直持续，直到 SAR 寄存器的第 0 位。
- <8> 10 位比较完成时，有效数字结果存储在 SAR 寄存器，然后再传送并存储在 ADA0CRn 寄存器。之后，产生 A/D 转换结束中断请求信号（INTAD）。
- <9> 单次选择模式下，转换停止^注。单次扫描模式下，扫描完成之后，立刻停止转换^注。连续选择模式下，重复步骤<2>至步骤<8>直到 ADA0M0.ADA0CE 位清为 0。连续扫描模式下，每个通道都重复步骤<2>至步骤<8>。

注 在外部触发模式时、定时器触发模式 0 时或定时器触发模式 1 时，进入触发等待状态。

备注 触发等待状态指稳定时间过后的状态。

13.5.2 转换操作时序

图 13-3. 转换操作时序（连续转换）



13.5.3 触发模式

通过设置触发模式指定转换操作的启动时序。触发模式包括软件触发模式和硬件触发模式。硬件触发模式包含定时器触发模式 0 和 1，以及外部触发模式。ADA0M0.ADA0TMD 位用于设置触发模式。通过 ADA0M2.ADA0TMD1 位和 ADA0M2.ADA0TMD0 位设置硬件触发模式。

(1) 软件触发模式

当 ADA0M0.ADA0CE 位置 1 时，对 ADA0S 寄存器指定的模拟输入引脚（ANI0 引脚至 ANI15 引脚）的信号进行转换。当转换完成时，转换结果存储在 ADA0CRn 寄存器。同时，产生 A/D 转换结束中断信号（INTAD）。

如果通过 ADA0M0.ADA0MD1 位和 ADA0M0.ADA0MD0 位指定操作模式为连续选择/扫描模式，那么第一次转换完成后，将启动下一次转换，直到 ADA0CE 位清 0。如果操作模式是单次选择/扫描模式，那么转换执行一次后结束。

转换开始时，ADA0M0.ADA0EF 位设为 1（表示转换正在进行）。

如果在转换期间对 ADA0M0、ADA、ADA0S、ADA0PFM 或 ADA0PFT 寄存器执行写入操作，那么此次转换中止，并再次从头开始转换。但是，在正常转换模式和高速转换模式下的单次选择模式/单次扫描模式中，禁止写入这些寄存器。

(2) 外部触发模式

该模式下，当外部触发信号输入（至 ADTRG 引脚）时，对 ADA0S 寄存器指定的模拟输入引脚（ANI0 至 ANI15）的信号进行转换。通过 ADA0M0.ADA0ETS1 位和 ADA0M0.ADA0ETS0 位来指定外部触发器的哪个边沿（也就是上升沿、下降沿或双边沿）被检测。ADA0CE 位设为 1 时，A/D 转换器等待触发，且当外部触发信号输入后，开始转换。

无论通过 ADA0MD1 位和 ADA0MD0 位设置的操作模式是连续选择、连续扫描、单次选择还是单次扫描模式，转换完成后，转换结果都存储在 ADA0CRn 寄存器中。同时，产生 INTAD 信号，并且 A/D 转换器等待下一次触发信号。

转换开始时，ADA0EF 位设为 1（表示转换正在进行）。然而，在 A/D 转换器等待触发期间，ADA0EF 位清为 0。如果在转换操作期间输入有效触发信号，转换操作中止，并再次从头开始。

如果在转换操作期间对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器执行写入操作，那么此次转换停止，并且 A/D 转换器等待再次触发。但是，在单次选择模式/单次扫描模式下，禁止写入这些寄存器。

注意事项 为了选择外部触发模式，需要设置为高速转换模式。A/D 转换操作使能（ADA0M0.ADA0CE 位= 1）后，在随后插入的振荡稳定时间期间，不要输入触发信号。

备注 触发等待状态指稳定时间过后的状态。

(3) 定时器触发模式

该模式下，通过该定时器的捕获/比较寄存器（连接至定时器）的比较匹配中断请求信号（INTTP2CC0 或 INTTP2CC1），开始对 ADA0S 寄存器指定的模拟输入引脚（ANI0 至 ANI15）的信号进行转换。通过 ADA0TMD1 位和 ADA0TMD0 位选择 INTTP2CC0 信号或 INTTP2CC1 信号，并且在指定的比较匹配中断请求信号的上升沿处开始转换。当 ADA0CE 位设为 1 时，A/D 转换器等待触发，当定时器的比较匹配中断请求信号输入后，开始转换。

转换完成时，无论通过 ADA0MD1 位和 ADA0MD0 位设置的操作模式是连续选择、连续扫描、单次选择还是单次扫描模式，转换结果都存储在 ADA0CRn 寄存器中。同时，产生 INTAD 信号，并且 A/D 转换器等待下一次触发。

转换开始时，ADA0EF 位设为 1（表示转换正在进行）。但是，在 A/D 转换器等待触发期间，ADA0EF 位清为 0。如果在转换操作期间输入有效触发，转换操作中止，并再次从头开始。

如果在转换操作期间对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器执行写入操作，那么此次转换停止，并且 A/D 转换器等待再次触发。但是，在单次选择模式/单次扫描模式下，禁止写入这些寄存器。

注意事项 为了选择定时器触发模式，需要设置为高速转换模式。A/D 转换操作使能（ADA0M0.ADA0CE 位=1）后，在随后插入的振荡稳定时间期间，不要输入触发信号。

备注 触发等待状态指稳定时间过后的状态。

13.5.4 操作模式

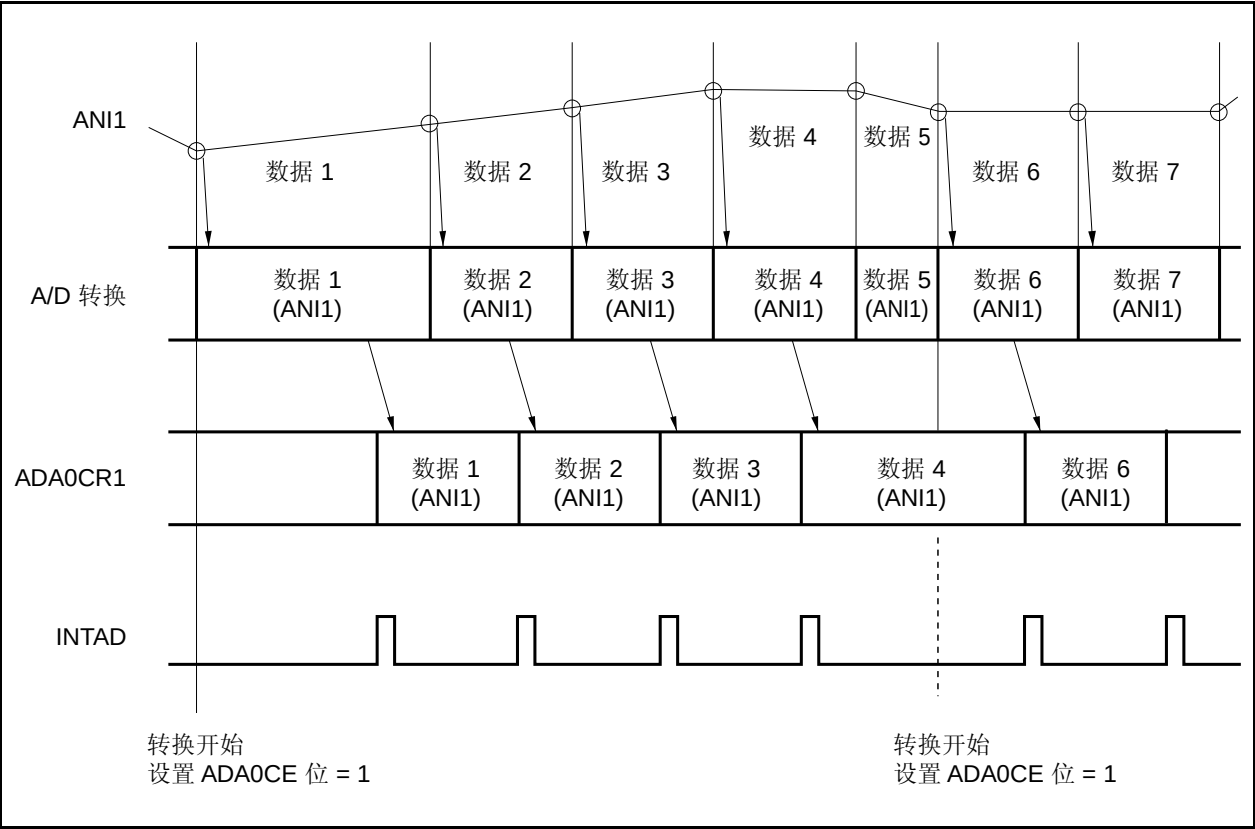
有四种操作模式可供选择：连续选择模式，连续扫描模式，单次选择模式以及单次扫描模式。在这四种操作模式下，需设置 ANI0 引脚至 ANI15 引脚。

通过 ADA0M0.ADA0MD1 位和 ADA0M0.ADA0MD0 位来选择操作模式。

(1) 连续选择模式

该模式下，ADA0S 寄存器选择一个模拟输入引脚的电压被连续转换为数字值。
转换结果保存在该模拟输入引脚相对应的 ADA0CRn 寄存器中。该模式下，按照一一对应的方式，一个模拟输入引脚对应一个 ADA0CRn 寄存器。每次 A/D 转换完成后，都会产生 A/D 转换结束中断请求信号（INTAD）。
转换完成后，则开始下一次转换，除非 ADA0M0.ADA0CE 位被清为 0（n = 0 至 15）。

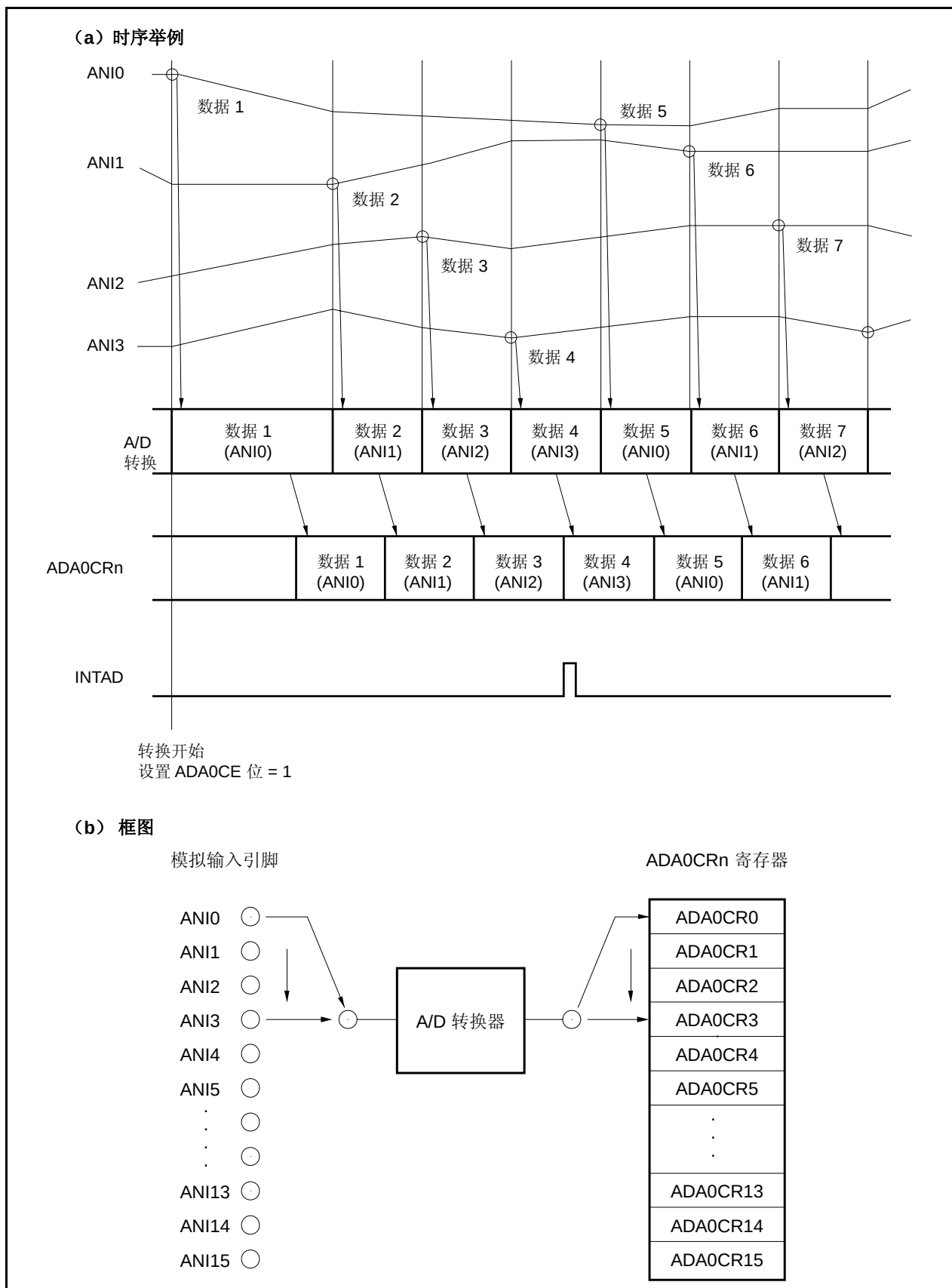
图 13-4. 连续选择模式操作的时序举例（ADA0S 寄存器= 01H）



(2) 连续扫描模式

该模式下，模拟输入引脚是顺序选择的，从 ANI0 引脚至 ADA0S 寄存器指定的引脚，并且将它们的输入值转换为数字值。
各个通道的转换结果保存在与各个模拟输入引脚相对应的 ADA0CRn 寄存器中。当寄存器指定的模拟输入引脚转换完成时，产生 INTAD 信号，并且又从 ANI0 引脚开始 A/D 扫描转换，除非 ADA0CE 位被清为 0（n = 0 至 15）。

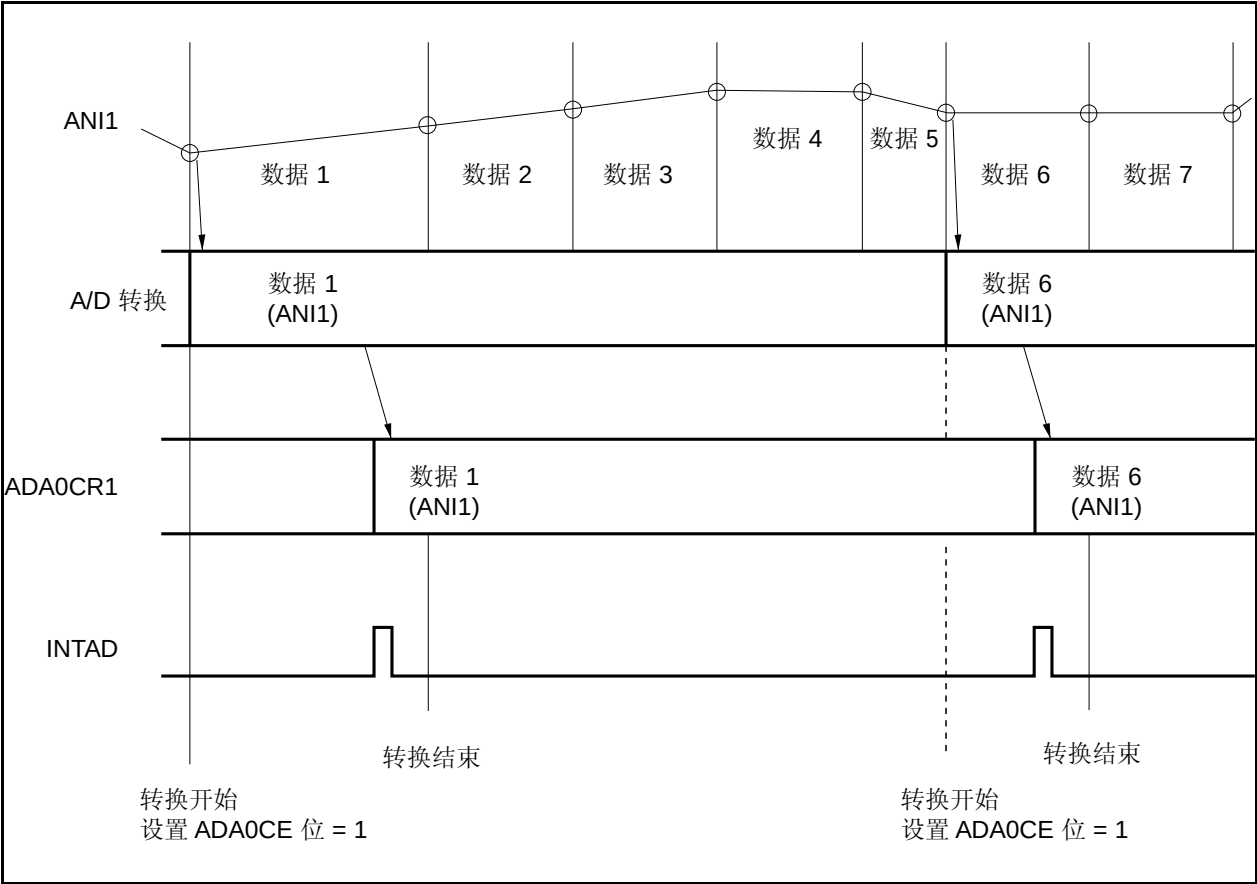
图 13-5. 连续扫描模式操作的时序举例 (ADA0S 寄存器= 03H)



(3) 单次选择模式

该模式下，对 ADA0S 寄存器选择的模拟输入引脚电压仅进行一次转换为数字值的操作。
转换结果保存在该模拟输入引脚相对应的 ADA0CRn 寄存器中。该模式下，模拟输入引脚与 ADA0CRn 寄存器一一对应。 A/D 转换完成一次后，产生 INTAD 信号。上述操作完成后，A/D 转换操作停止（n = 0 至 15）。

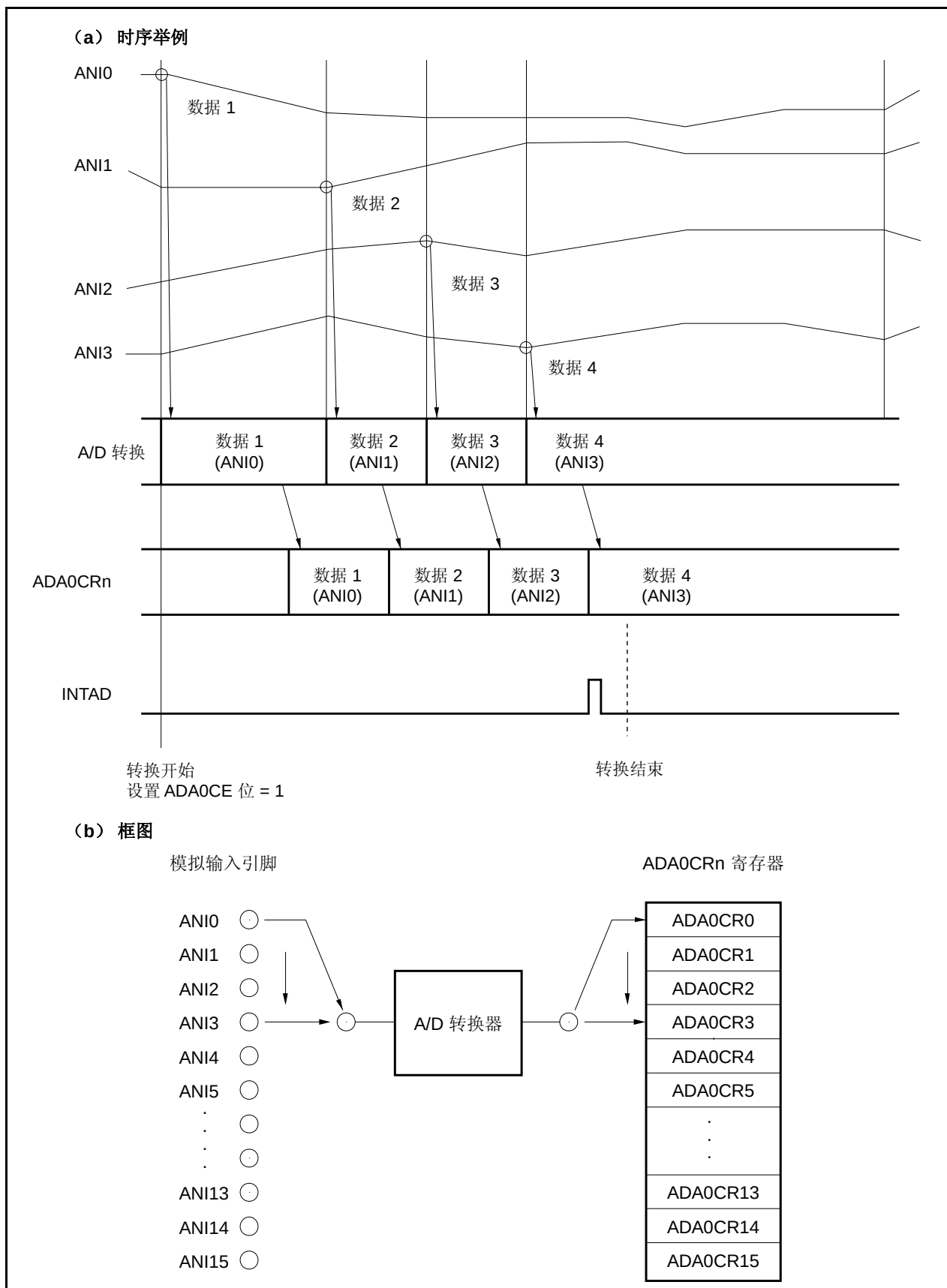
图 13-6. 单次选择模式操作的时序举例（ADA0S 寄存器= 01H）



(4) 单次扫描模式

该模式下，模拟输入引脚是顺序选择的，从 ANI0 引脚至 ADA0S 寄存器指定的引脚，并且将它们的输入值转换为数字值。
各个转换结果保存在各个模拟输入引脚相对应的 ADA0CRn 寄存器中。当 ADA0S 寄存器指定的模拟输入引脚转换完成时，产生 INTAD 信号。上述操作完成后，A/D 转换停止（n=0 到 15）。

图 13-7. 单次扫描模式操作的时序举例 (ADA0S 寄存器= 03H)



13.5.5 掉电比较模式

A/D 转换结束中断请求信号 (INTAD) 可以通过 ADA0PFM 寄存器和 ADA0PFT 寄存器控制。

- 当 ADA0PFM.ADA0PFE 位= 0 时，每次转换完成（正常使用 A/D 转换器）都产生 INTAD 信号。
- 当 ADA0PFE 位= 1 且 ADA0PFM.ADA0PFC 位= 0 时，ADA0CRnH 寄存器的值与转换完成时 ADA0PFT 寄存器的值进行比较，并且仅当 $\text{ADA0CRnH} \geq \text{ADA0PFT}$ 时，产生 INTAD 信号。
- 当 ADA0PFE 位= 1 且 ADA0PFC 位= 1 时，ADA0CRnH 寄存器的值与转换完成时 ADA0PFT 寄存器的值进行比较，并且仅当 $\text{ADA0CRnH} < \text{ADA0PFT}$ 时，产生 INTAD 信号。

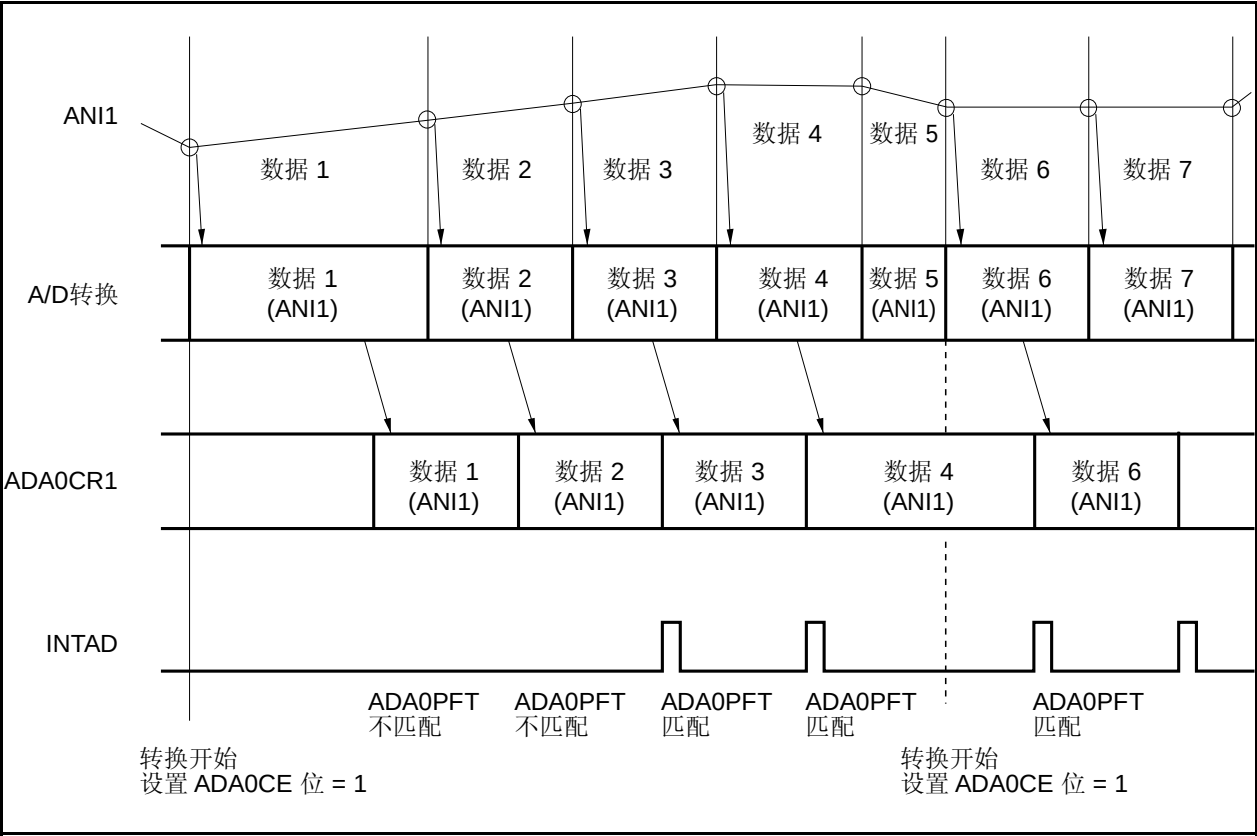
备注 n = 0 至 15

掉电比较模式中，有四种模式可供选择：连续选择模式，连续扫描模式，单次选择模式以及单次扫描模式。在这四种操作模式中，需设置 ANI0 引脚至 ANI15 引脚。

(1) 连续选择模式

该模式下，ADA0S 寄存器指定的模拟输入引脚的电压转换结果与 ADA0PFT 寄存器的设置值相比较。如果掉电比较的结果符合 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CRn 寄存器，并产生 INTAD 信号。如果不符合条件，转换结果仍存储在 ADA0CRn 寄存器中，但不产生 INTAD 信号。第一次转换完成后，则开始下一次转换，除非 ADA0M0.ADA0CE 位被清为 0（n = 0 至 15）。

图 13-8. 连续选择模式操作的时序举例（掉电比较模式时：ADA0S 寄存器 = 01H）

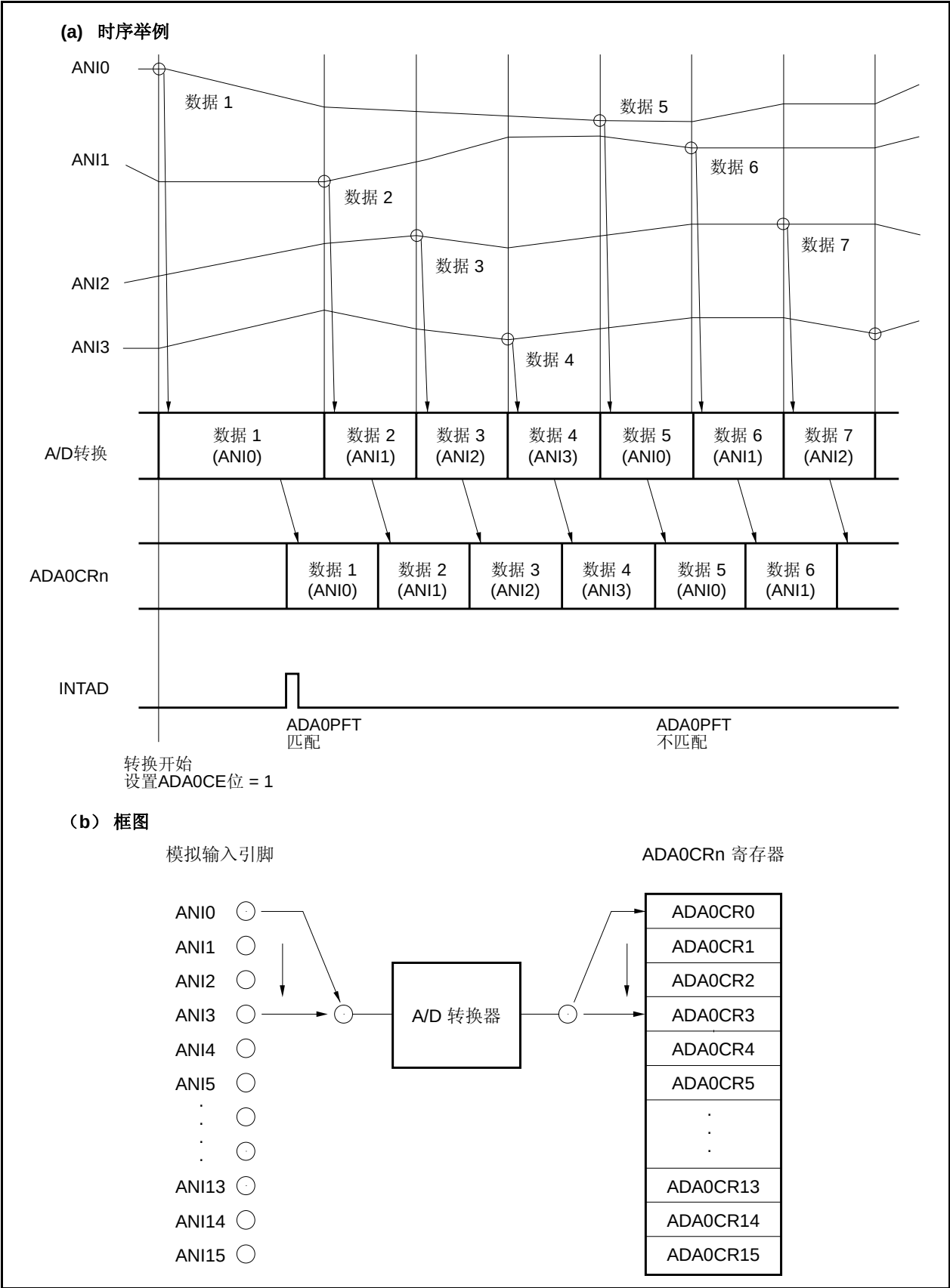


(2) 连续扫描模式

该模式下，模拟输入引脚是顺序选择的，从 ANI0 引脚至 ADA0S 寄存器指定的引脚，对它们的模拟输入值进行 A/D 转换，并且将 ADA0CR0H 寄存器的设置值与 ADA0PFT 寄存器的值作比较。如果掉电比较的结果符合 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CR0 寄存器，并产生 INTAD 信号。如果不符合条件，转换结果仍存储在 ADA0CR0 寄存器中，但不产生 INTAD 信号。

第一次转换结果存储在 ADA0CR0 寄存器之后，顺序对模拟输入引脚上的电压进行转换并存储，至 ADA0S 寄存器指定的模拟输入引脚。转换完成后，又从 ANI0 引脚启动下一轮转换，直到 ADA0CE 位被清为 0。

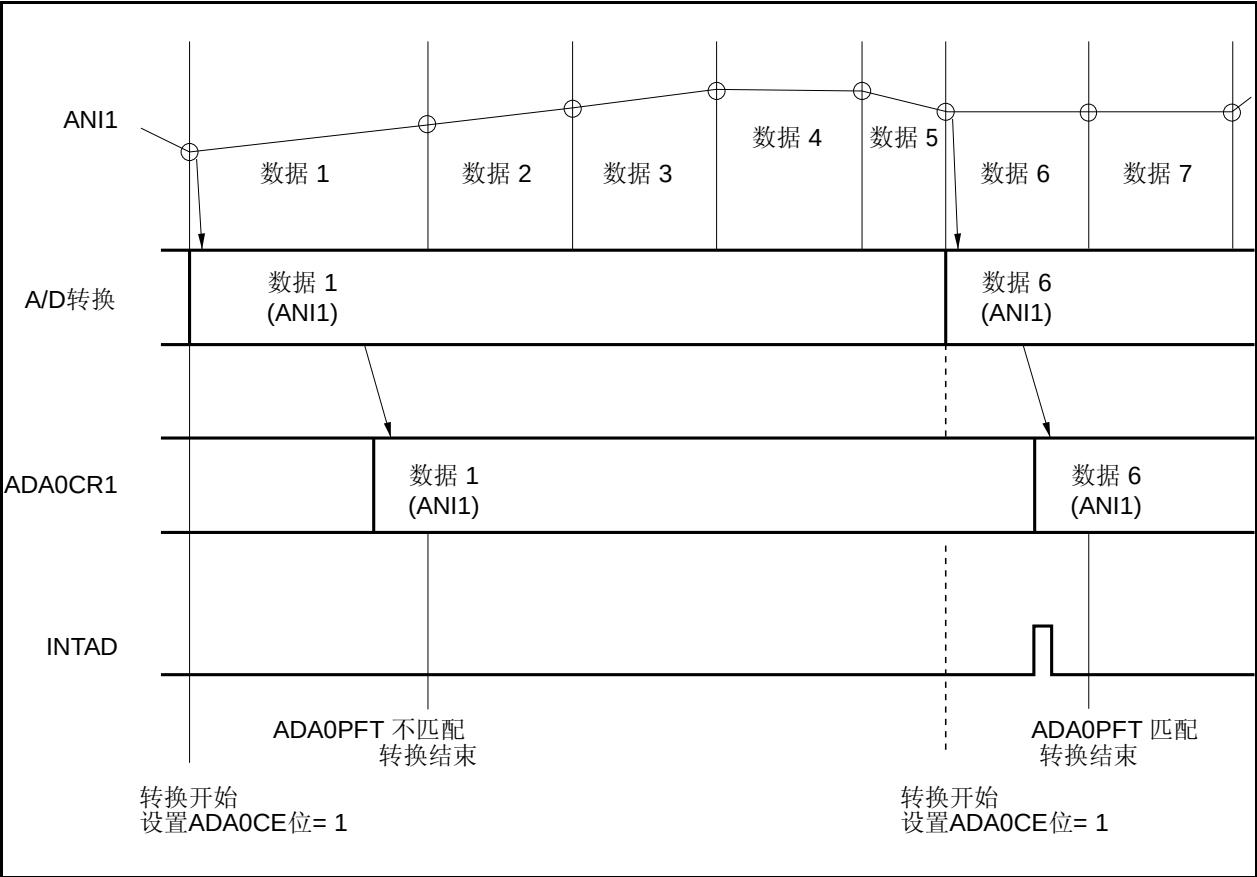
图 13-9. 连续扫描模式操作的时序举例（掉电比较模式时：ADA0S 寄存器 = 03H）



(3) 单次选择模式

该模式下，ADA0S 寄存器指定的模拟输入引脚的电压转换结果与 ADA0PFT 寄存器的设置值相比较。如果掉电比较的结果符合由 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CRn 寄存器，并产生 INTAD 信号。如果不符合条件，转换结果仍存储在 ADA0CRn 寄存器中，但不产生 INTAD 信号。上述操作完成后，停止转换。

图 13-10. 单次选择模式操作的时序举例（掉电比较模式时：ADA0S 寄存器 = 01H）

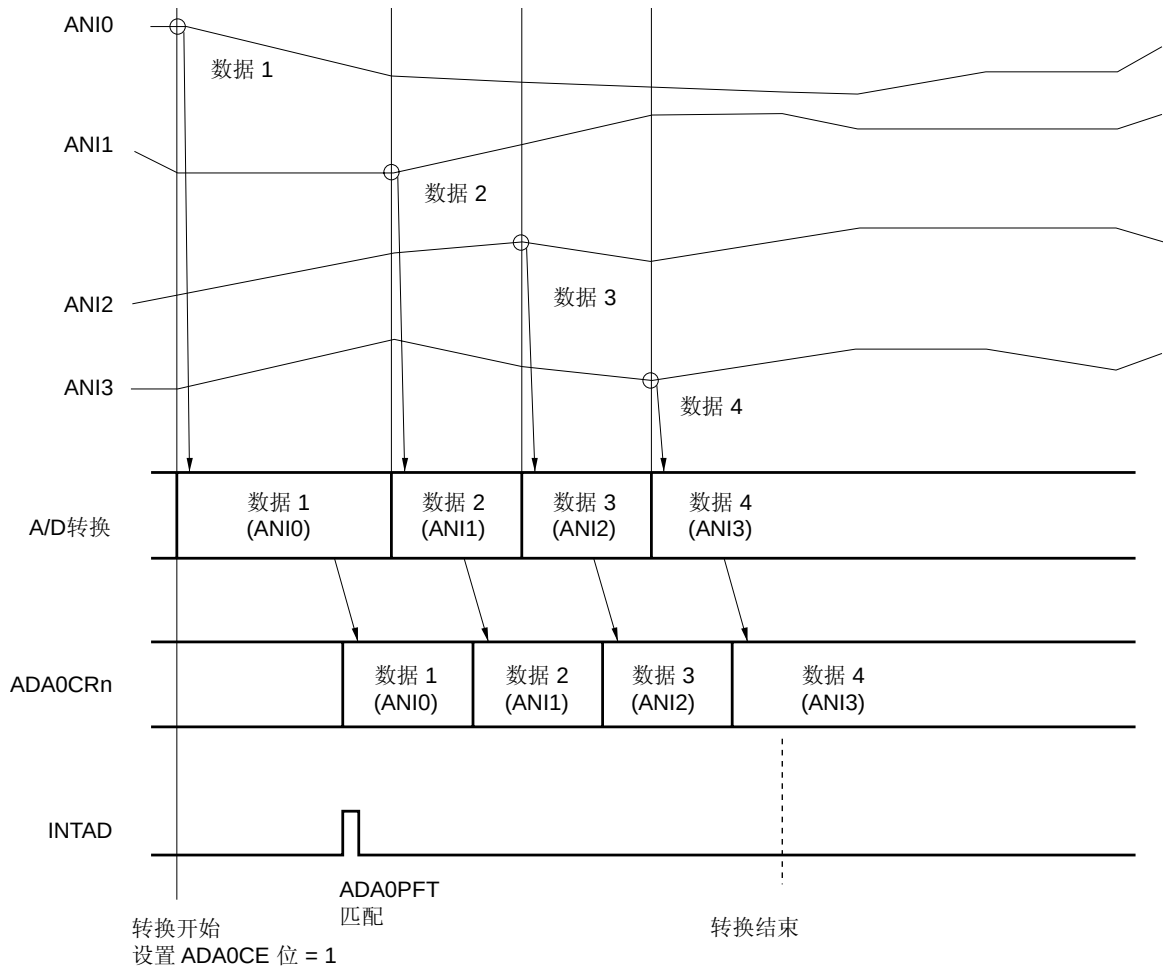


(4) 单次扫描模式

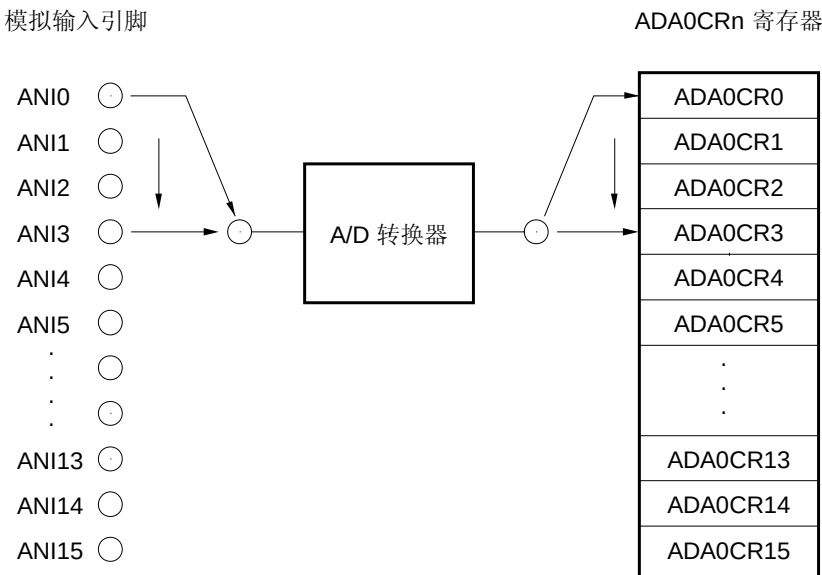
该模式下，模拟输入引脚是顺序选择的，从 ANI0 引脚至 ADA0S 寄存器指定的引脚，对它们的模拟输入值进行 A/D 转换，存储电压转换结果，并且将通道 0 的 ADA0CR0H 寄存器设置值与 ADA0PFT 寄存器的设置值作比较。如果掉电比较的结果符合由 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CR0 寄存器，并产生 INTAD 信号。如果不符合条件，转换结果仍存储在 ADA0CR0 寄存器中，但不产生 INTAD0 信号。第一次转换结果存储在 ADA0CR0 寄存器后，依次存储 ADA0S 寄存器指定的模拟输入引脚的转换信号结果。上述操作完成后，停止转换。

图 13-11. 单次扫描模式操作的时序举例（掉电比较模式时：ADA0S 寄存器 = 03H）

(a) 时序举例



(b) 框图



13.6 注意事项

(1) 不使用 A/D 转换器时

不使用 A/D 转换器时，将 ADA0M0.ADA0CE 位清 0，可以降低功耗。

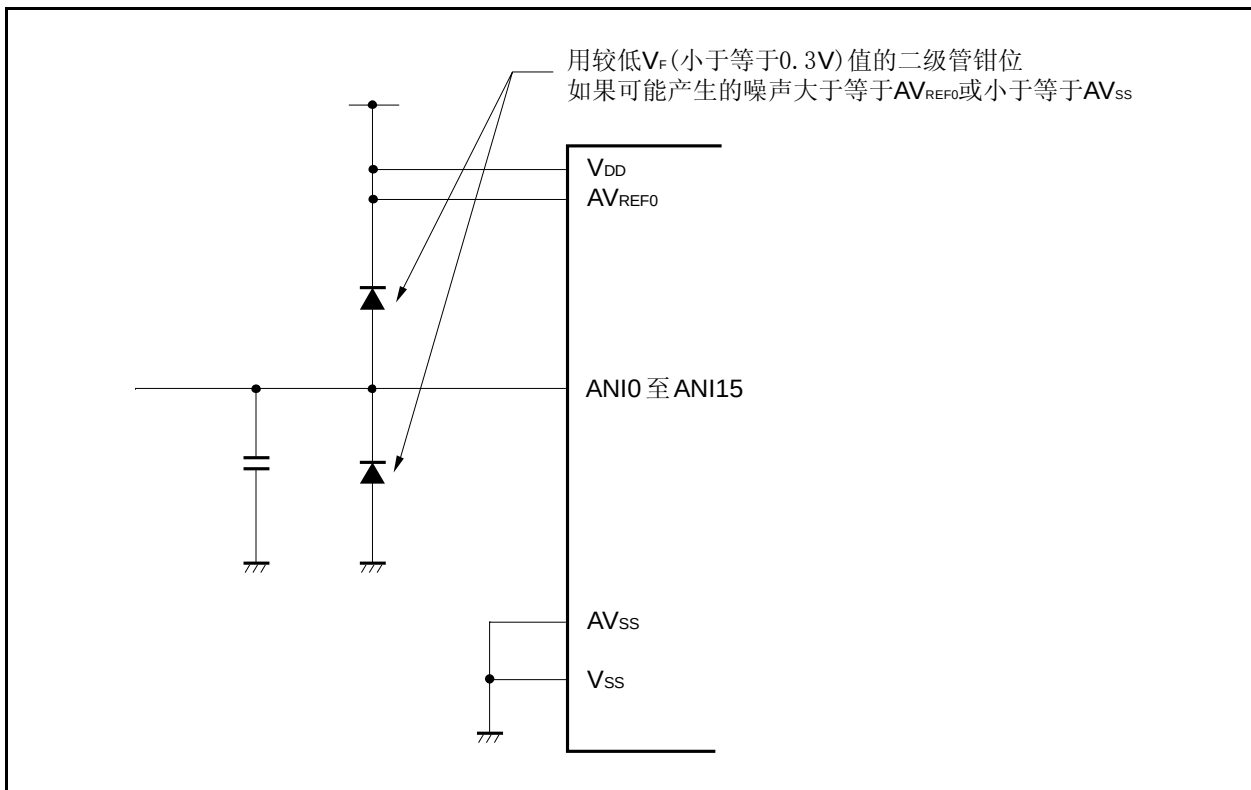
(2) ANI0 至 ANI15 引脚输入范围

向 ANI0 引脚至 ANI15 引脚输入的电压应该在指定范围内。如果大于等于 AV_{REF0} 或小于等于 AV_{SS} （即使在最大额定值范围内）的电压输入到这些引脚中的任意一个引脚，则此通道的转换值将不确定，并且其它通道的转换值也会受到影响。

(3) 应对噪声的措施

为了保证 10 位分辨率，必须有效防止噪声对 ANI0 引脚至 ANI15 引脚的干扰。噪声的影响随着模拟输入源输出阻抗的增大而加强。为了减小噪声，建议连接一个外部电容器，方式如图 13-12 所示。

图 13-12. 模拟输入引脚的处理



(4) 复用 I/O

模拟输入引脚（ANI0 至 ANI15）可以复用为端口引脚。当选择 ANI0 引脚至 ANI15 引脚的其中一个来执行 A/D 转换时，在转换期间不要执行读取输入端口或写入输出端口的指令，因为这样可能会降低转换的分辨率。

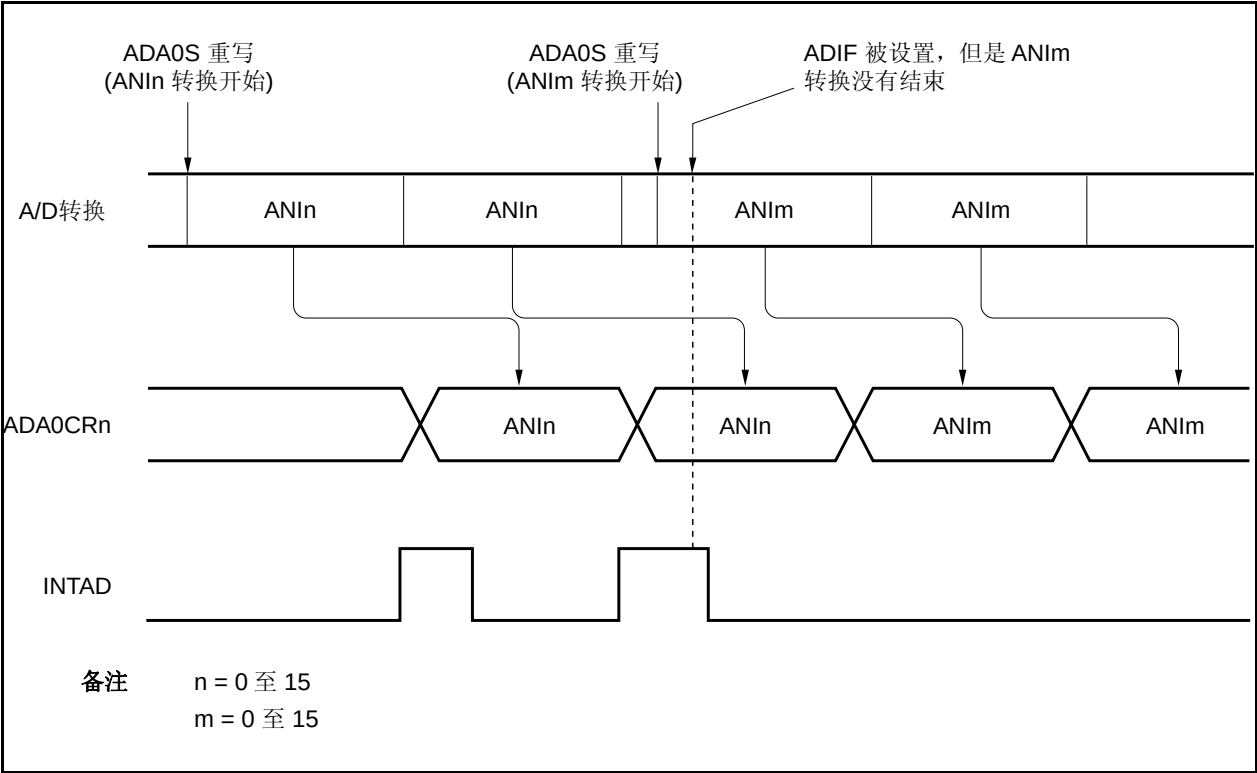
在 A/D 转换期间端口的某些引脚被设置作为输出，由于连接至端口引脚的外部电路干扰使得输出电流波动，输入引脚的转换分辨率也会下降。

如果有数字脉冲施加在与正被转换的输入信号引脚相邻的引脚上，由于耦合噪声的影响，A/D 转换值可能会无法预料。因此，不要在进行 A/D 转换引脚的邻近引脚上施加脉冲。

(5) 中断请求标志 (ADIF)

即使 ADA0S 寄存器的内容改变，也不会清除中断请求标志 (ADIF)。因此，A/D 转换期间如果改变模拟输入通道，那么先前所选模拟输入引脚信号的转换结果可能被存储，且有可能在紧邻 ADA0S 寄存器重写之前，转换结束中断请求标志被置位。如果写入 ADA0S 寄存器之后立即读取 ADIF 标志，那么即使新选择的模拟输入引脚的 A/D 转换还没有完成，也会置位 ADIF 标志。A/D 转换停止时，重新开始转换之前应该清除 ADIF 标志。

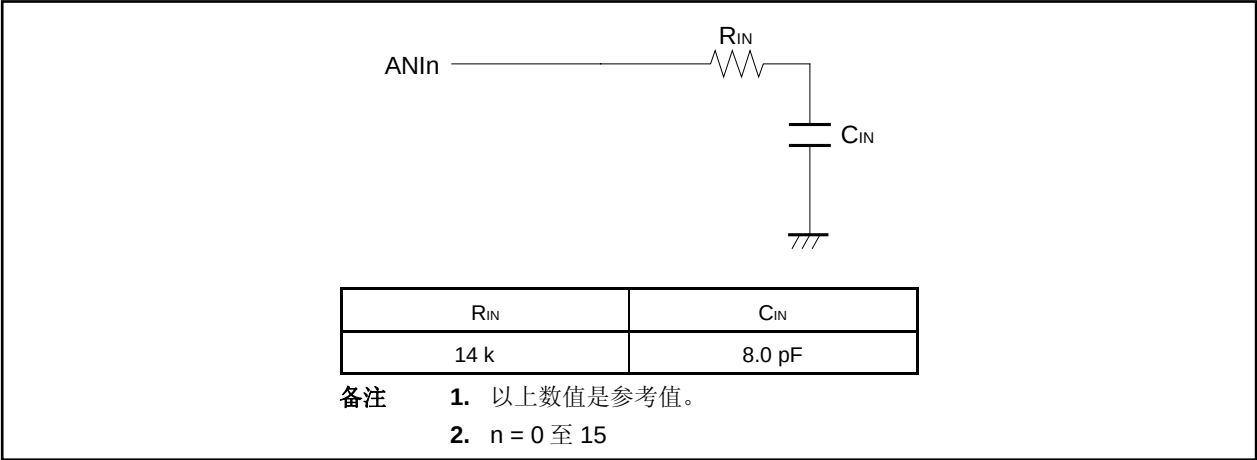
图 13-13. A/D 转换结束中断请求的发生时序



(6) 内部等效电路

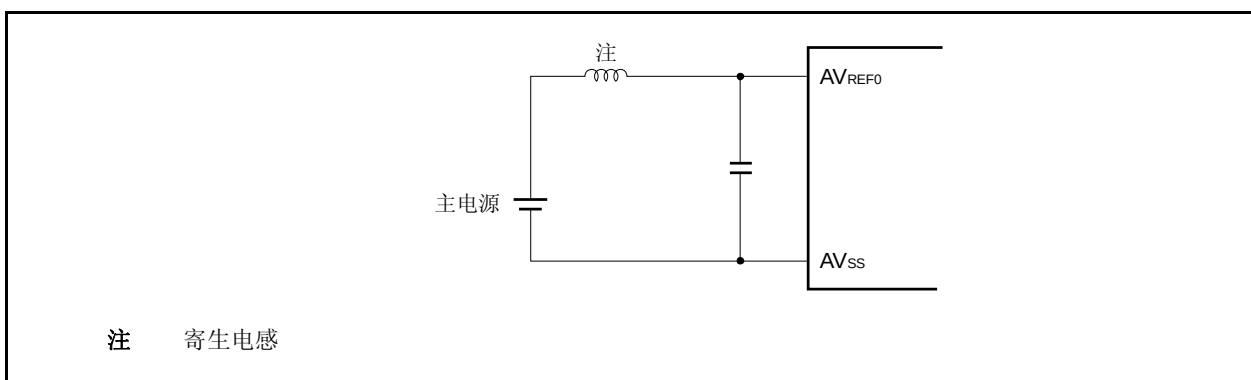
以下显示了模拟输入模块的等效电路。

图 13-14. ANIn 引脚的内部等效电路



(7) AV_{REF0} 引脚

- (a) AV_{REF0} 引脚用作 A/D 转换器的电源，也为复用功能端口提供电源。在使用备用电源的应用系统中，确保提供至 AV_{REF0} 引脚的电压与 V_{DD} 相同，如图 13-15 所示。
- (b) AV_{REF0} 引脚用作 A/D 转换器的参考电压引脚。如果 AV_{REF0} 引脚的输入电源具有高阻抗或者电流提供能力较弱，那么转换期间由于电流波动，参考电压可能会波动（尤其紧接在转换操作使能位 $ADA0CE$ 被置为 1 之后）。结果，转换精度可能下降。为了避免这种情况，建议在 AV_{REF0} 引脚和 AV_{SS} 引脚之间连接一个电容器来抑制参考电压的波动，如图 13-15 所示。
- (c) 如果连接至 AV_{REF0} 引脚的电源有具有高直流电阻（例如，加入二极管），由于 A/D 转换电流引起电压下降，所以开始转换时的电压可能低于转换停止时的电压。

图 13-15. AV_{REF0} 引脚处理实例**(8) 读取 $ADA0CRn$ 寄存器**

当写入 $ADA0M0$ 至 $ADAM2$ 、 $ADA0S$ 、 $ADA0PFM$ 或 $ADA0PFT$ 寄存器时， $ADA0CRn$ 寄存器的内容可能不确定。在转换完成之后且写入 $ADA0M0$ 至 $ADAM2$ 、 $ADA0S$ 、 $ADA0PFM$ 或 $ADA0PFT$ 寄存器之前，读取转换结果。当响应外部/定时器触发时， $ADA0CRn$ 寄存器的内容也可能不确定。在转换完成之后且响应下一次外部/定时器触发之前，读取转换结果。不同于上述时序时，可能读取不到正确的转换结果。

(9) 待机模式

在 STOP 模式下，由于 A/D 转换器停止工作，转换结果无效，因此能够降低功耗。STOP 模式解除后，操作重新开始，但是 STOP 模式解除后的 A/D 转换结果无效。STOP 模式解除后，需要使用 A/D 转换器时，在设置 STOP 模式之前或解除 STOP 模式之前，将 ADA0M0.ADA0CE 位清 0，然后在解除 STOP 模式之后，将 ADA0CE 位设为 1。

在 IDLE1、IDLE2 或副时钟操作模式下，操作继续。因此，为了降低功耗，将 ADA0M0.ADA0CE 位清 0。在 IDLE1 和 IDLE2 模式下，由于不能够保持模拟输入引脚的电压值，所以 IDLE1 和 IDLE2 模式解除后，A/D 转换结果无效。设置进入 IDLE1 和 IDLE2 模式之前的转换结果是有效的。

(10) 高速转换模式下

在高速转换模式下，振荡稳定期间，禁止触发信号的输入，也禁止对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 和 ADA0PFT 寄存器执行重写操作。

(11) A/D 转换时间

A/D 转换时间为稳定时间，转换时间，等待时间和触发响应时间的总和（关于这些时间的细节，参见表 13-2 正常转换模式（ADA0HS1 位= 0）下转换时间的选择和表 13-3 高速转换模式（ADA0HS1 位 = 1）下转换时间的选择。

在正常转换模式下，A/D 转换期间，如果对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 以及 ADA0PFT 寄存器执行重写操作或者输入触发信号，则重新执行转换。但是，如果稳定时间结束的时序与写入这些寄存器的时序发生冲突，或者稳定时间结束的时序与触发信号输入的时序发生冲突，那么将会插入 64 个时钟的稳定时间。

如果在插入的稳定时间结束时再次发生时序冲突，则仍然插入稳定时间。因此，不要将触发信号输入间隔和控制寄存器写入间隔设置为 64 个时钟或更少。

(12) A/D 转换结果的变化

因为供应电压的波动或噪声的影响，A/D 转换的结果可能会有变化。为了减小变化，在程序中应用抑制措施，例如取 A/D 转换结果平均值。

(13) A/D 转换结果迟滞特性

逐次比较类型 A/D 转换器在内部采样&保持电容器上保持输入电压，然后进行 A/D 转换。完成后，模拟输入电压仍然保持在内部采样保持电容器上。因此，可能发生以下现象：

- 当相同通道被用于 A/D 转换时，如果当前电压比先前的电压高或低，在转换结果受先前值影响的地方，可能出现滞后特性。因此，即使对相同的电压进行转换，结果也可能不同。
- 当切换模拟输入通道时，在转换结果受先前通道值影响的地方，可能出现滞后特性。这是因为 A/D 转换功能只有一个 A/D 转换器。因此，即使对相同的电压进行转换，结果也可能不同。

13.7 A/D 转换器特征表的阅读方法

本部分描述了 A/D 转换器的相关术语。

(1) 分辨率

分辨率为能够识别的最低模拟输入电压，也就是说，每位输出数字所占模拟输入电压的百分比，叫作 1 LSB（最低有效位）。1 LSB 对满量程的比率被表示为 %FSR（满标度量程）。%FSR 是百分比表示的可转换模拟输入电压比率，不管分辨率如何，总是可以用下式表示：

$$\begin{aligned} 1\%FSR &= (\text{可转换的模拟输入电压的最大值} - \text{可转换的模拟输入电压的最小值}) / 100 \\ &= (AV_{REF0} - 0) / 100 \\ &= AV_{REF0} / 100 \end{aligned}$$

当分辨率为 10 位时，1 LSB 表示如下：

$$\begin{aligned} 1 \text{ LSB} &= 1/2^{10} = 1/1,024 \\ &= 0.098\%FSR \end{aligned}$$

精确度是由总体误差决定，与分辨率无关。

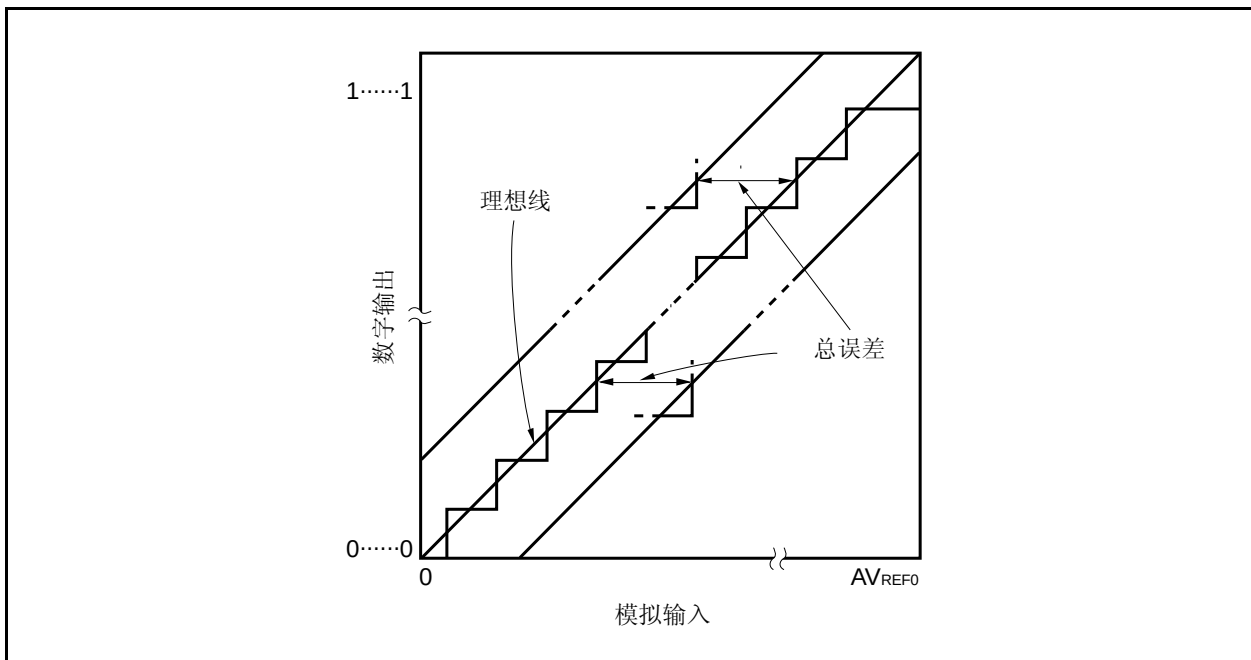
(2) 总体误差

总体误差是实际测量值和理论值之间的最大误差值。

零标度误差、满量程误差、线性误差，这些误差的综合构成总体误差。

在特性表中的总体误差里没有包含量化误差。

图 13-16. 总体误差

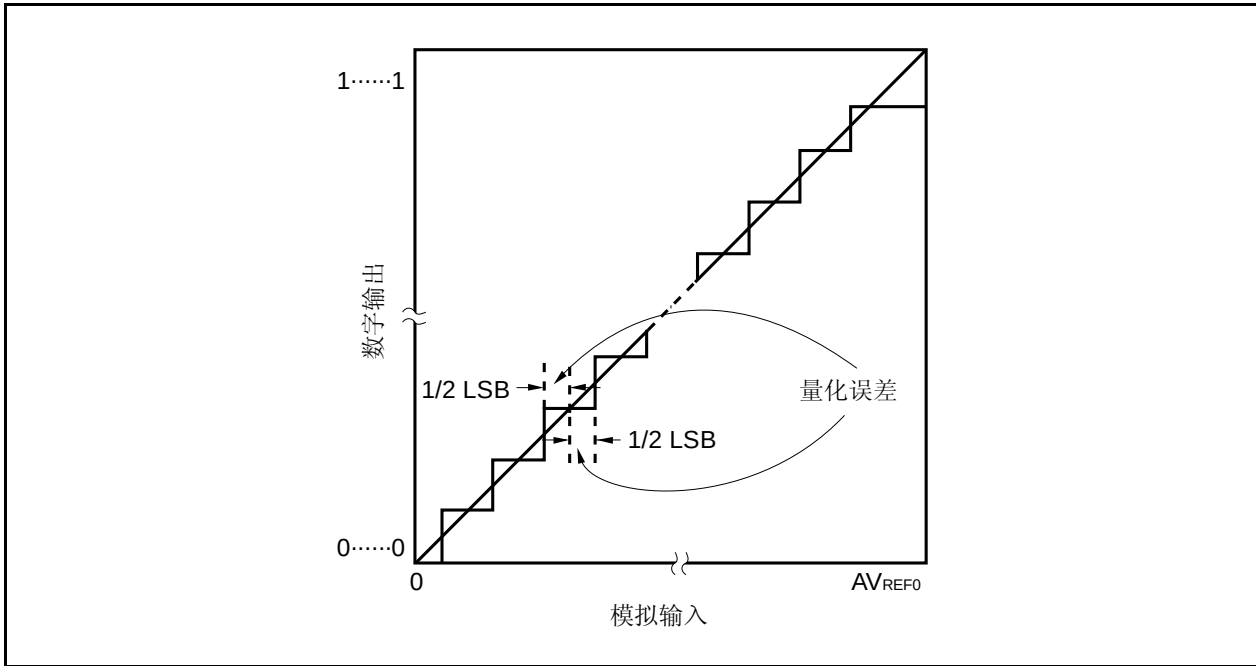


(3) 量化误差

当模拟值转换为数字值时，会不可避免的产生 $\pm 1/2$ LSB 的误差。因为 A/D 转换器将 $\pm 1/2$ LSB 范围内的模拟输入电压转换成相同的数字值，所以量化误差不能避免。

量化误差不包括在参数表中的总体误差、零标度误差、满量程误差、积分线性误差或微分线性误差之中。

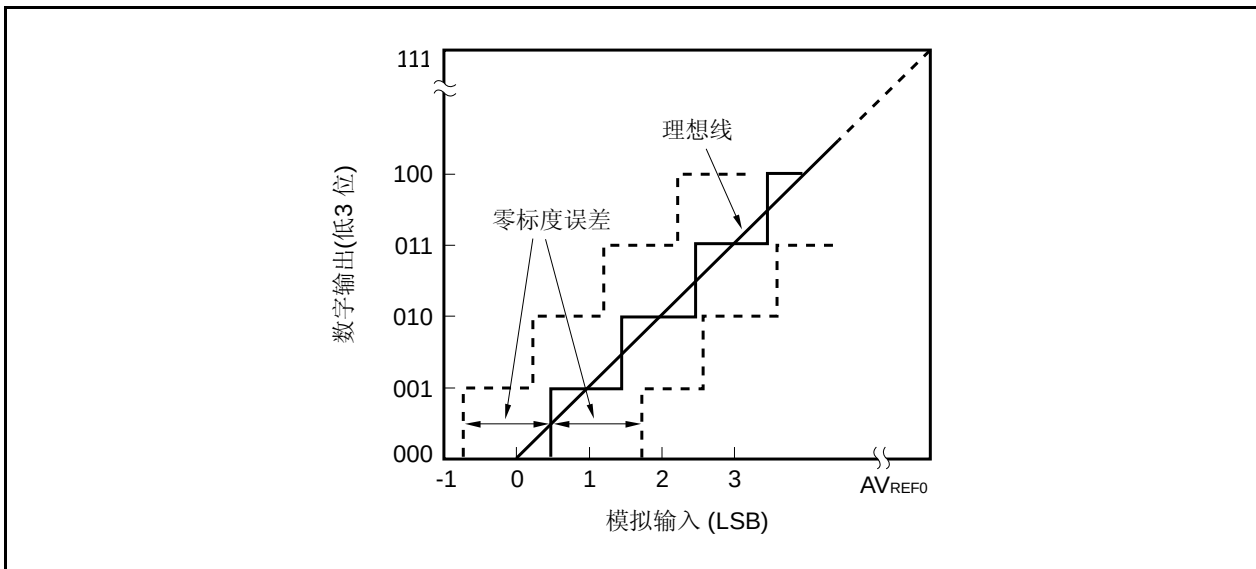
图 13-17. 量化误差



(4) 零标度误差

当数字输出从 0.....000 改变到 0.....001 ($1/2$ LSB) 时，零标度误差就是实际测量的模拟输入电压和理论值之间的差别。

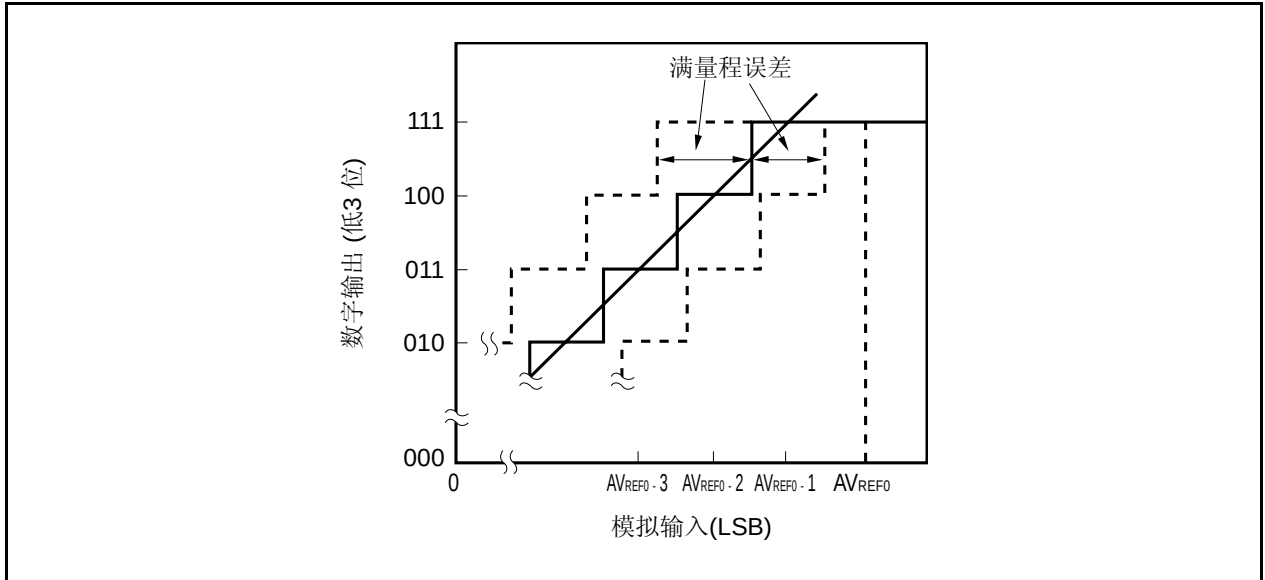
图 13-18. 零标度误差



(5) 满量程误差

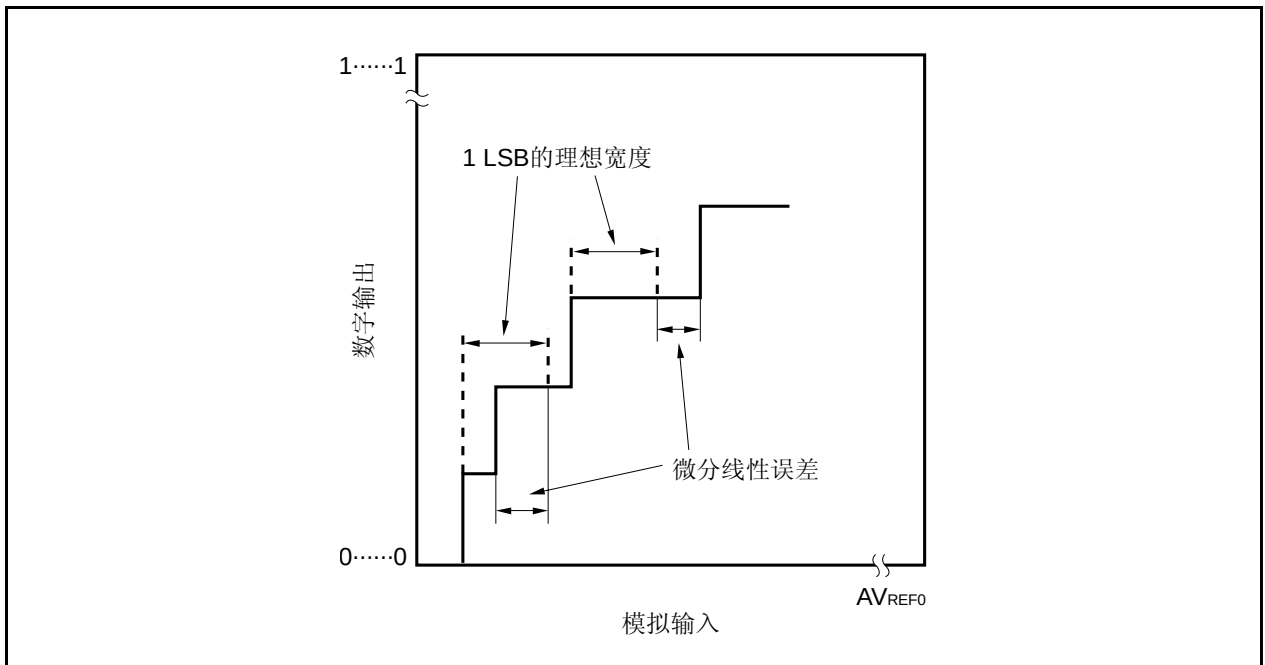
当数字输出从 1...110 改变到 1...111（满量程 - 3/2 LSB）时，满量程误差就是实际测量的模拟输入电压与理论值之间的差别。

图 13-19. 满量程误差

**(6) 微分线性误差**

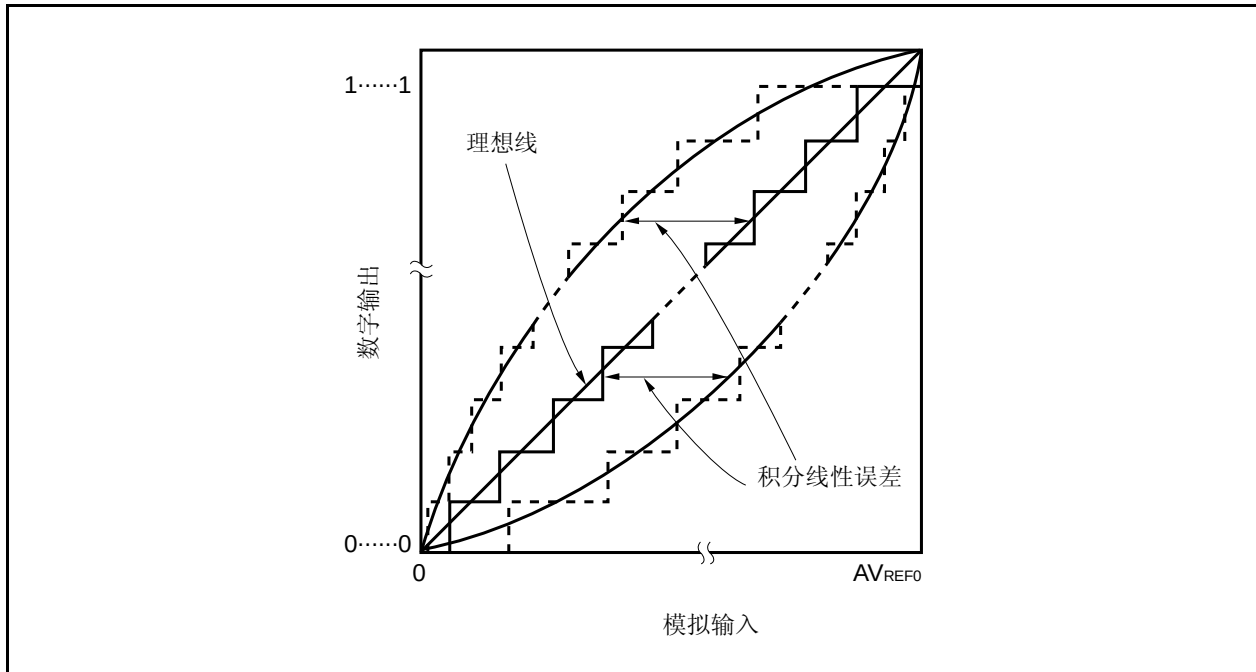
理想状态下，输出指定编码的宽度是 1 LSB。输出指定编码时，该误差显示了实际测量值和其理论值之间的差别。输入到相同通道的模拟输入引脚的电压从 AV_{SS} 逐位增加到 AV_{REF0} ，这代表了 A/D 转换的基本特性。当输入电压增加或减小时，或使用两个或更多的通道时，参见 13.7 (2) 总体误差。

图 13-20. 微分线性误差



(7) 积分线性误差

积分线性误差显示转换特性偏离理想线性关系的程度。它表示当零标度误差和满量程误差均为 0 时，实际测量值和其理论值之间的最大误差。

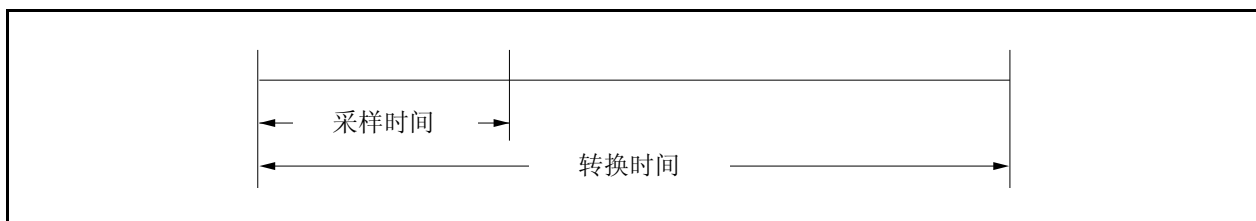
图 13-21. 积分线性误差**(8) 转换时间**

转换时间为触发信号产生后获得数字输出所需的时间。

特性表中，转换时间包含采样时间。

(9) 采样时间

采样时间，即模拟开关开启，将模拟电压载入采样&保持电路的时间。

图 13-22. 采样时间

14.1 功能

D/A 转换器具有下列功能：

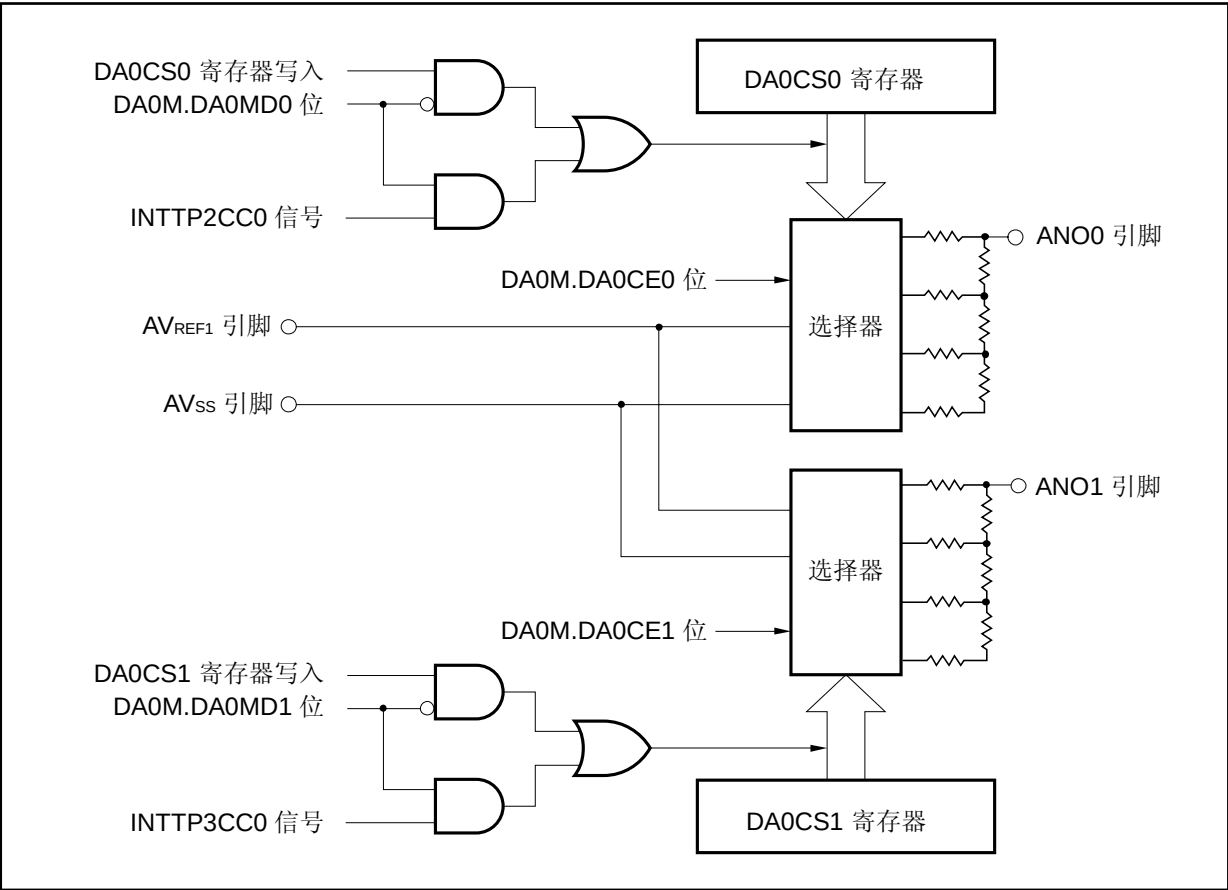
- 8 位精度 × 2 通道（DA0CS0，DA0CS1）
- R-2R 阶梯变换方法
- 稳定时间：最大 3 μ s（AVREF1 为 3.0 至 3.6 V 且外部阻抗 20 pF）
- 模拟输出电压： $AV_{REF1} \times m/256$ （m = 0 至 255；DA0CSn 寄存器的设定值）
- 操作模式：正常模式，实时输出模式

备注 n = 0, 1

14.2 配置

D/A 转换器配置如下所示：

图 14-1. D/A 转换器功能框图



注意事项 1. DAC0 和 DAC1 共用 AVREF1 引脚。
2. DAC0 和 DAC1 共用 AVss 引脚。D/A 转换器和 A/D 转换器共用 AVss 引脚。

D/A 转换器包括下列硬件：

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示：

- D/A 转换器模式寄存器（DA0M）
- D/A 转换值设置寄存器 0，1（DA0CS0，DA0CS1）

(1) D/A 转换器模式寄存器（DA0M）

DA0M 寄存器用于控制 D/A 转换器的操作。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFFFF282H

	7	6	<5>	<4>	3	2	1	0
DA0M	0	0	DA0CE1	DA0CE0	0	0	DA0MD1	DA0MD0

DA0CEn	D/A 转换器操作使能/禁止控制（n = 0，1）
0	禁止操作
1	使能操作

DA0MDn	D/A 转换器操作模式的选择（n = 0，1）
0	正常模式
1	实时输出模式 ^注

注 实时输出模式（DA0MD0 位 = 1）的输出触发信号如下：

- 当 n = 0 时：INTTP2CC0 信号（参见第七章 16 位定时器/事件计数器 P（TMP））。
- 当 n = 1 时：INTTP3CC0 信号（参见第七章 16 位定时器/事件计数器 P（TMP））。

(2) D/A 转换值设置寄存器 0, 1 (DA0CS0, DA0CS1)

DA0CS0 和 DA0CS1 寄存器用于设置输出到 ANO0 和 ANO1 引脚的模拟电压。
这些寄存器可以进行字节读取或写入。
系统复位后，这些寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF280H

7 6 5 4 3 2 1 0

DA0CS0

DA0CS07	DA0CS06	DA0CS05	DA0CS04	DA0CS03	DA0CS02	DA0CS01	DA0CS00
---------	---------	---------	---------	---------	---------	---------	---------

注意事项

在实时输出模式 (DA0M.DA0MDn 位 = 1) 下，在 INTTP2CC0/ INTTP3CC0 信号产生以前，设置 DA0CSn 寄存器。当 INTTP2CC0/ INTTP3CC0 信号产生时，D/A 转换器开始工作。

备注

n = 0, 1

14.4 操作

14.4.1 正常模式下的操作

将写入 DA0CSn 寄存器的操作作为触发信号，执行 D/A 转换操作。

设置方法如下：

- <1> 清除 DA0M.DA0MDn 位为 0（正常模式）。
- <2> 将要输出到 ANOn 引脚的模拟电压值写入 DA0CSn 寄存器。
以上步骤<1>和步骤<2>为初始化设置。
- <3> 设置 DA0M.DA0CEn 位为 1（D/A 转换使能）。
执行该设置时，D/A 转换启动。
- <4> 写入 DA0CSn 寄存器，以进行后续的 D/A 转换。
前一次的 D/A 转换结果会一直保持，直到执行下一次 D/A 转换。

- 备注**
- 1. 关于复用功能引脚的设置，参见表 4-19 端口引脚用作复用功能引脚。
 - 2. n = 0, 1

14.4.2 实时输出模式下的操作

使用 TMP2 和 TMP3 的中断请求信号（INTTP2CC0 和 INTTP3CC0）作为触发信号，进行 D/A 转换。

设置方法描述如下：

- <1> 设置 DA0M.DA0MDn 位为 1（实时输出模式）。
- <2> 将要输出到 ANOn 引脚的模拟电压值写入 DA0CSn 寄存器。
- <3> 设置 DA0M.DA0CEn 位为 1（D/A 转换使能）。
以上步骤<1>至步骤<3>为初始化设置。
- <4> 操作 TMP2 和 TMP3。
- <5> 当产生 INTTP2CC0 和 INTTP3CC0 信号时，D/A 转换器开始工作。
- <6> 此后，在每次产生 INTTP2CC0 和 INTTP3CC0 信号时，设置到 DA0CSn 寄存器的值会被输出。

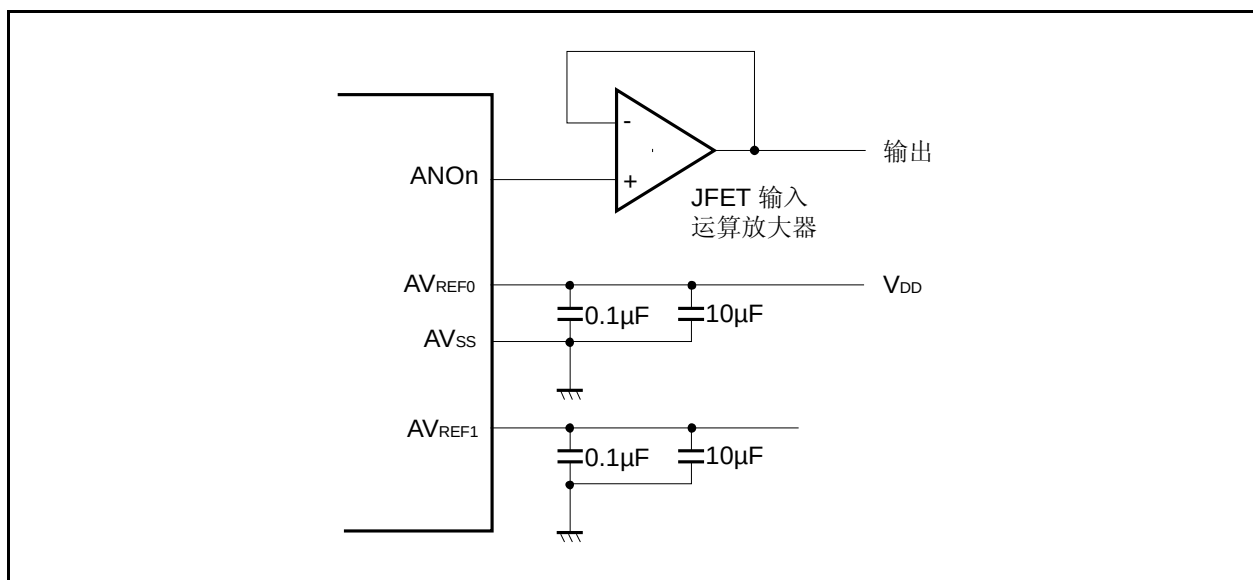
- 备注**
- 1. 步骤<5>之前，ANO0 和 ANO1 引脚的输出值不确定。
 - 2. 关于 ANO0 和 ANO1 引脚在 HALT，IDLE1，IDLE2 和 STOP 模式下的输出值，参见第二十一章 待机功能。
 - 3. 关于复用功能引脚的设置，参见表 4-19 端口引脚用作复用功能引脚。

14.4.3 注意事项

使用 V850ES/JJ3 的 D/A 转换器时，请遵循以下注意事项：

- (1) 实时输出模式下，触发信号发生时，不要改变 DA0CSn 寄存器的设定值。
- (2) 改变操作模式前，确保将 DA0M.DA0CEn 位清为 0。
- (3) 当使用 P10/AN00 和 P11/AN01 引脚之一作为 I/O 端口，而另一个用作 D/A 输出引脚。在 D/A 输出期间，如果端口 I/O 电平不改变，则在应用系统中可以这样使用。
- (4) 确保 $AV_{REF0} = V_{DD} = AV_{REF1} = 3.0$ 至 3.6 V。如果超出该范围，则操作无法保证。
- (5) 给 AV_{REF1} 上电的时序和给 AV_{REF0} 供电的时序相同。
- (6) 因为 D/A 转换器的输出阻抗高，所以 ANOn 引脚 ($n = 0, 1$) 没有电流输出。如果连接 $2\text{ M}\Omega$ 或更小的电阻，需在电阻和 ANOn 引脚之间插入一个 JFET(结型场效应晶体管)输入运算放大器。

图 14-2. 外部引脚连接举例



- (7) 因为在 STOP 模式下 D/A 转换器停止操作，ANO0 和 ANO1 引脚进入高阻抗状态，因而可以降低功耗。但是，在 IDLE1，IDLE2 或副时钟工作模式下，操作继续。为了降低功耗，将 DA0M.DA0CEn 位清为 0。

第十五章 异步串行接口 A（UARTA）

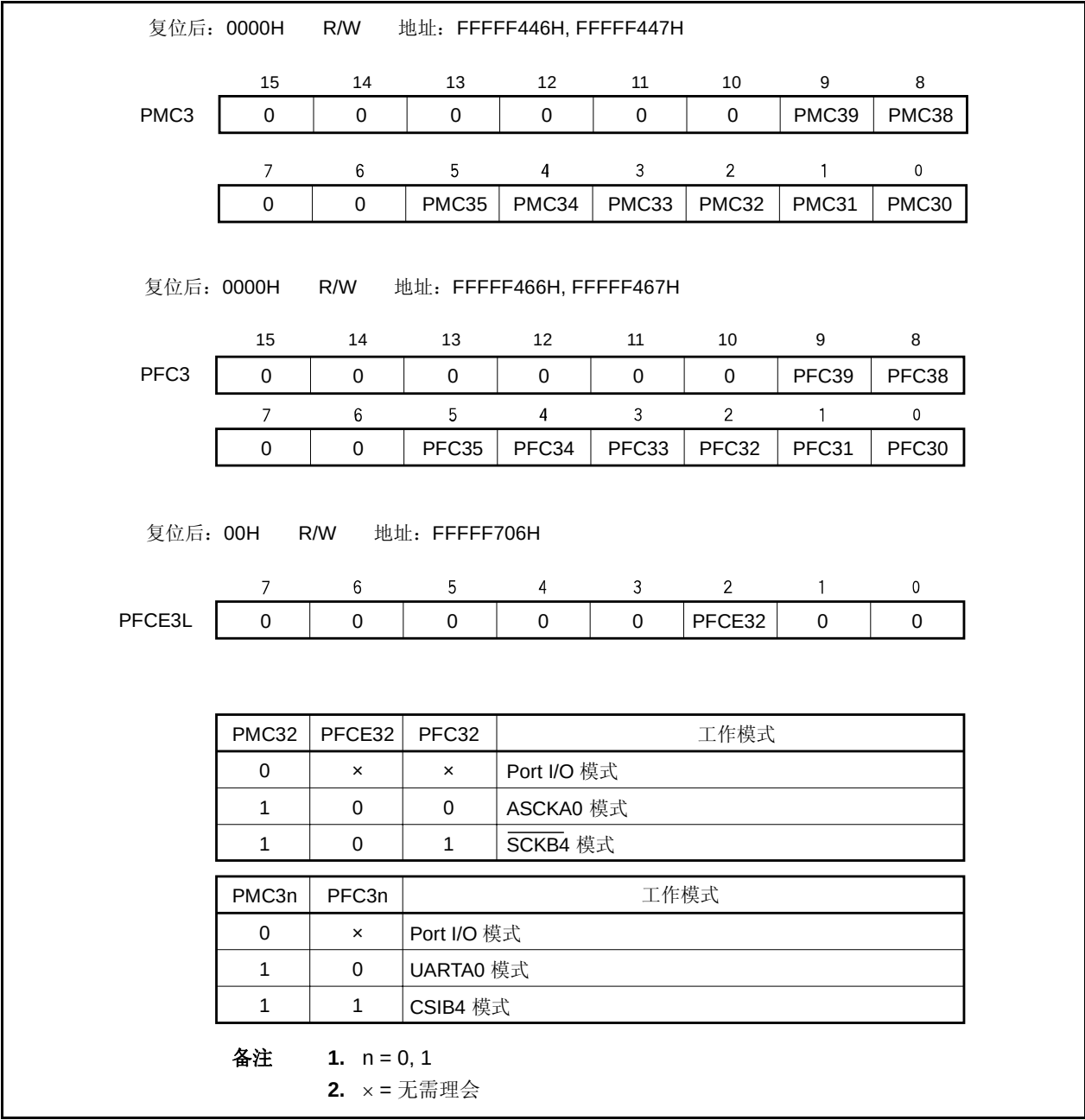
15.1 UARTA 和其它串行接口的模式转换

15.1.1 CSIB4 和 UARTA0 的模式转换

在 V850ES/JJ3 中，CSIB4 和 UARTA0 是相同引脚的复用功能，因而不能同时使用。使用之前，用 PMC3 和 PFC3 寄存器预先设置 UARTA0。

注意事项 如果在发送或接收期间切换这些功能，则 CSIB4 和 UARTA0 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 15-1. CSIB4 和 UARTA0 模式转换设置

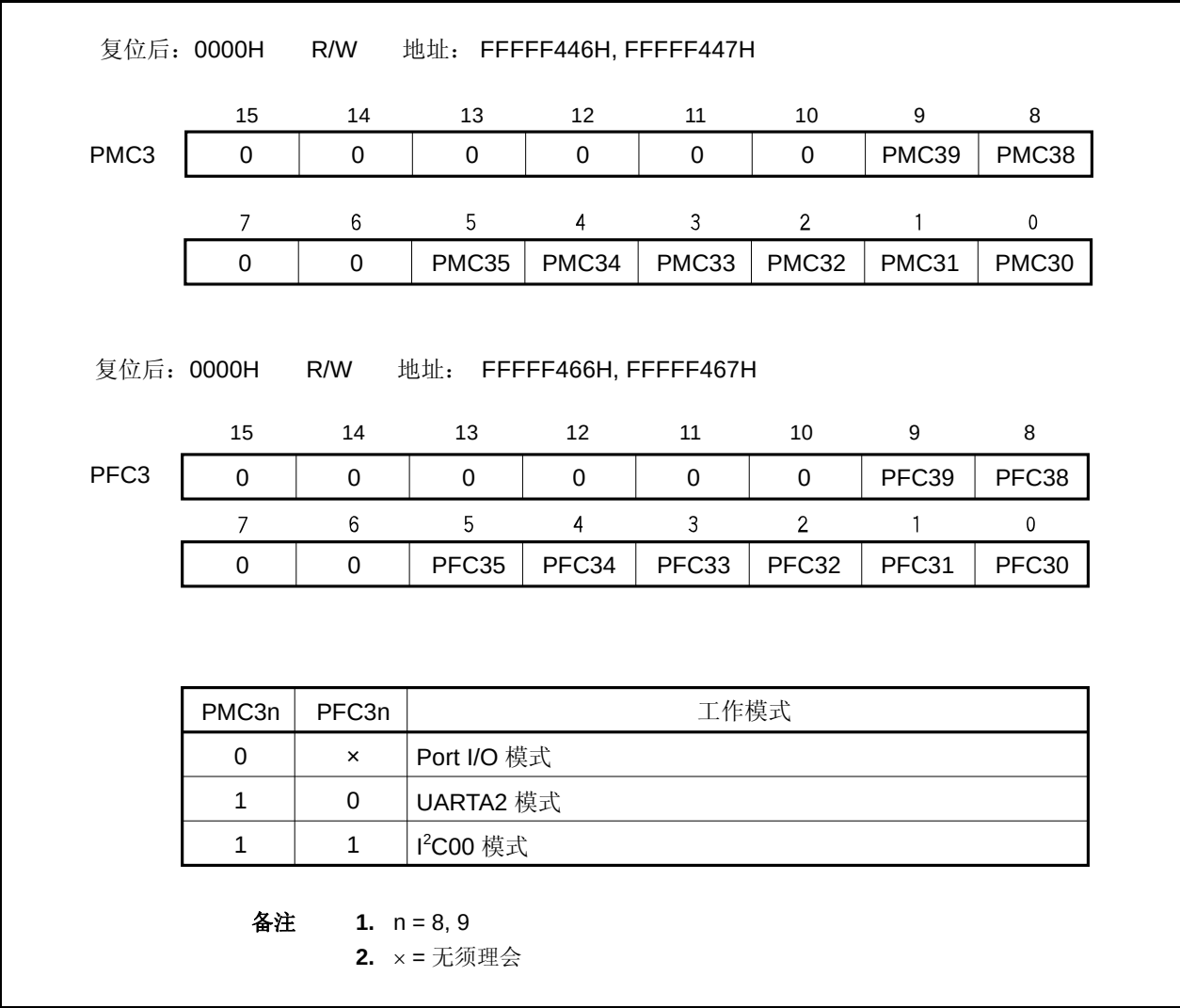


15.1.2 UARTA2 和I²C00 的模式转换

在V850ES/JJ3 中，UARTA2 和I²C00 是相同引脚的复用功能，因而不能同时使用。使用之前，用PMC3 和 PFC3 寄存器预先设置UARTA2。

注意事项 如果在发送或接收期间切换这些功能，则UARTA2 和I²C00 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 15-2. UARTA2 和I²C00 模式转换设置

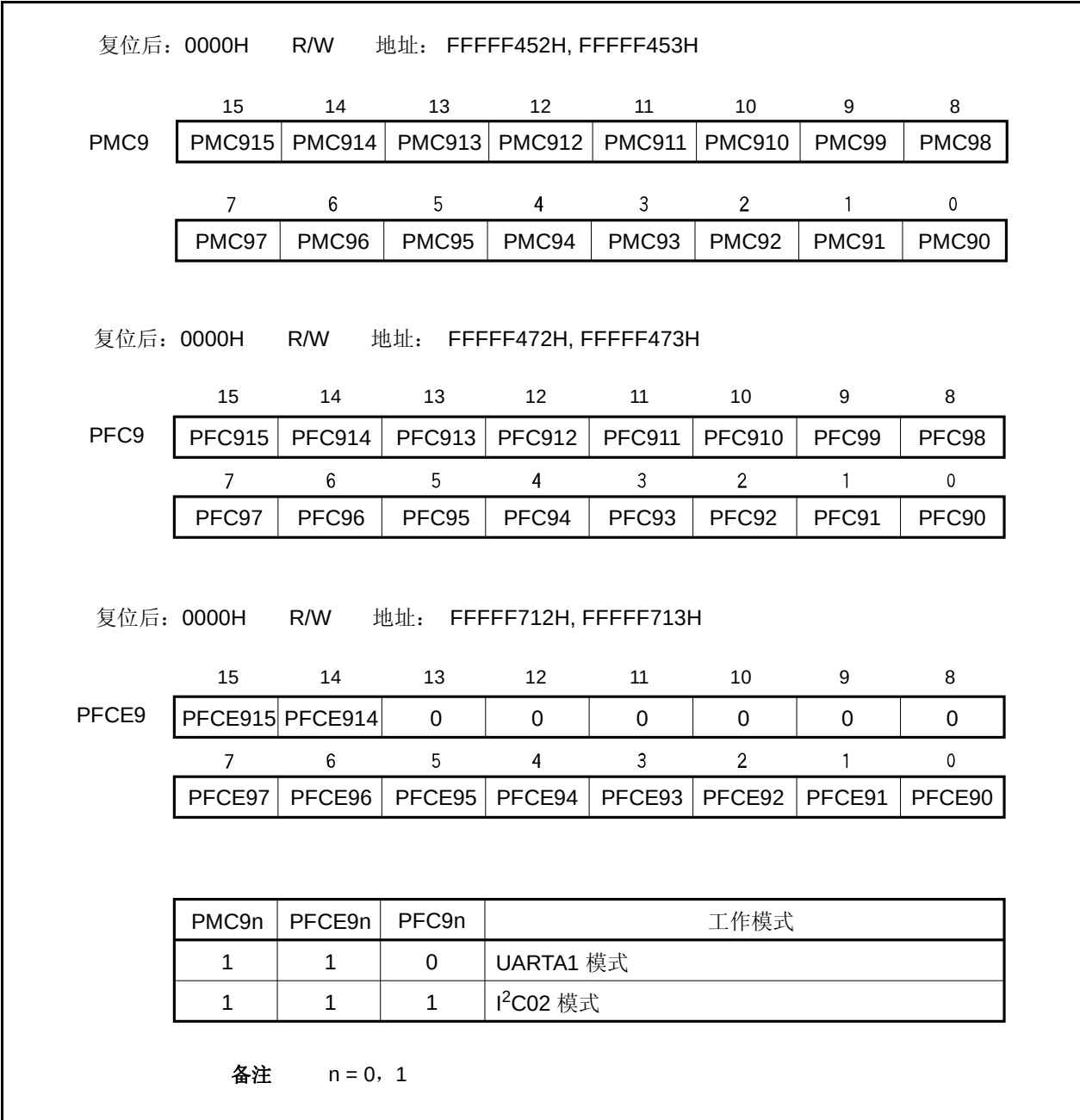


15.1.3 UARTA1 和 I²C02 的模式转换

在V850ES/JJ3 中，UARTA1 和 I²C02 是相同引脚的复用功能，因而不能同时使用。使用之前，用PMC9, PFC9 和 PMCE9 寄存器预先设置UARTA1。

注意事项 如果在发送或接收期间切换这些功能，则UARTA1 和 I²C02 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 15-3. UARTA1 和 I²C02 模式转换设置



15.2 特性

- 传输速率：300 bps 至 625 kbps（使用 32 MHz 的内部系统时钟和专用波特率发生器）
- 全双工通信：
 - 内部 UARTAn 接收数据寄存器 (UAnRX)
 - 内部 UARTAn 发送数据寄存器 (UAnTX)
- 2 引脚配置：
 - TXDAn：发送数据输出引脚
 - RXDAn：接收数据输入引脚
- 接收错误输出功能
 - 奇偶错误
 - 帧错误
 - 溢出错误
- 中断源：2
 - 接收完成中断 (INTUAnR)：

在接收使能状态下，串行传输完成后，当接收数据从接收移位寄存器传送到接收数据寄存器时，产生该中断。
 - 发送使能中断 (INTUAnT)：

在发送使能状态下，当数据从发送数据寄存器传送到发送移位寄存器时，产生该中断。
- 字符长度：7 或 8 位
- 奇偶校验功能：奇校验、偶校验、0 校验、无校验
- 发送停止位：1 或 2 位
- 片上专用波特率发生器
- 可选择 MSB/LSB 先行传输
- 发送/接收数据可以使输入/输出反相
- 在 LIN（局域内联网）通信格式下可以进行 SBF（同步间隔场）发送/接收
 - SBF 发送时，可选 13 至 20 位
 - SBF 接收时，可以识别 11 位或更多位
 - 提供 SBF 接收标志

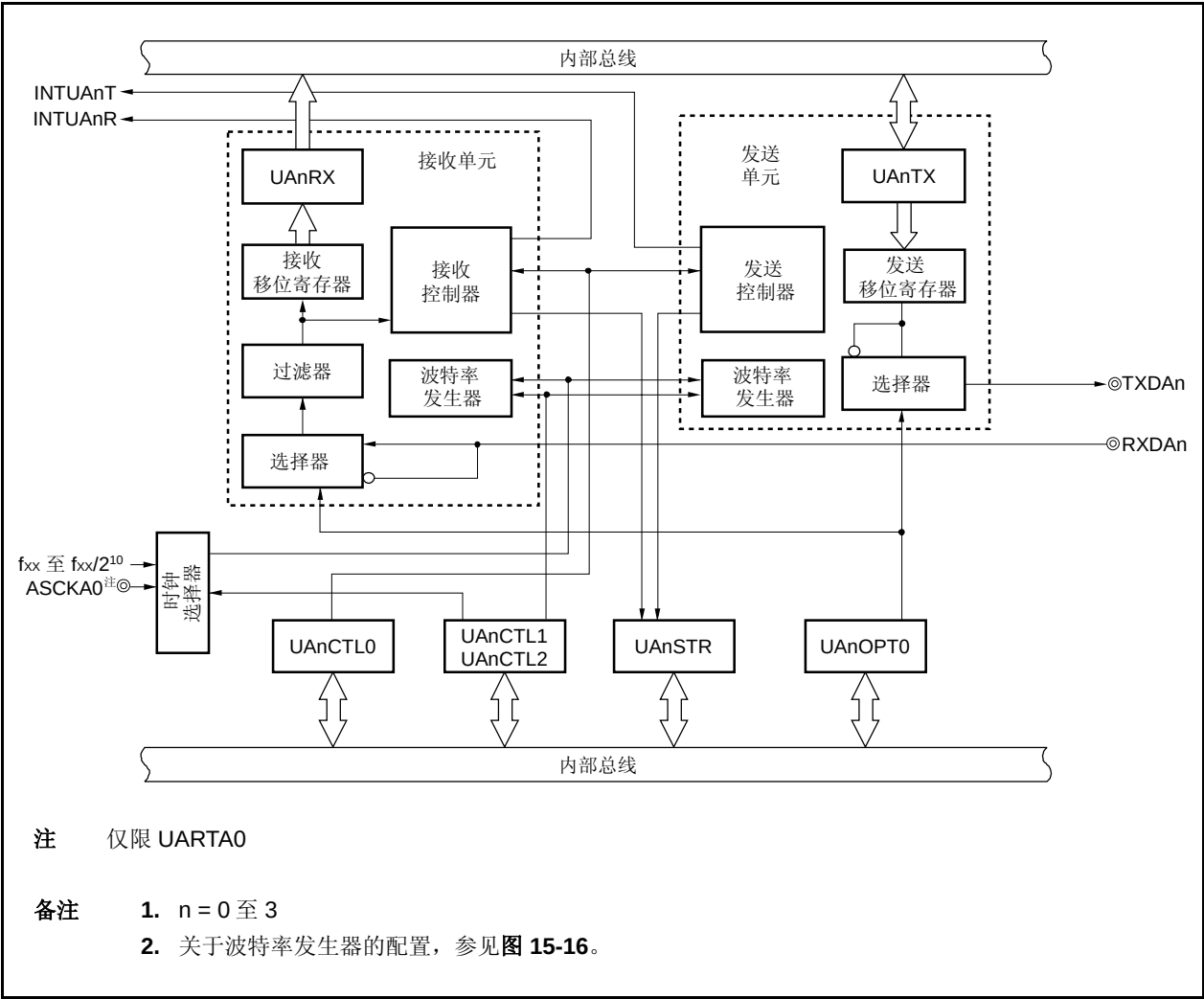
备注 n = 0 至 3

15.3 配置

UARTAn 的功能框图如下所示：

<R>

图 15-4. 异步串行接口 An 的功能框图



UARTAn 包括下列硬件：

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 选项控制寄存器 0 (UAnOPT0) UARTAn 状态寄存器 (UAnSTR) UARTAn 接收移位寄存器 UARTAn 接收数据寄存器 (UAnRX) UARTAn 发送移位寄存器 UARTAn 发送数据寄存器 (UAnTX)

(1) UARTAn 控制寄存器 0 (UAnCTL0)

UAnCTL0 寄存器是 8 位寄存器，用于指定 UARTAn 的操作。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

UAnCTL1 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)

UAnCTL2 寄存器是 8 位寄存器，用于控制 UARTAn 的波特率。

(4) UARTAn 选项控制寄存器 0 (UAnOPT0)

UAnOPT0 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)

UAnSTRn 寄存器由指示接收错误的标志位组成。当发生接收错误时，相应接收错误标志被置位 (1)。

(6) UARTAn 接收移位寄存器

本寄存器是移位寄存器，用于将输入至 RXDAn 引脚的串行数据转换为并行数据。接收到 1 字节数据并检测到停止位时，就将接收数据传输至 UAnRX 寄存器。

不能直接操作该寄存器。

(7) UARTAn 接收数据寄存器 (UAnRX)

UAnRX 是 8 位寄存器，用于保存接收数据。当接收到 7 个字符时，最高位清 0（当数据以 LSB 先行方式接收）。

在接收使能状态下，当 1 帧移入操作完毕时，接收数据同步从 UARTAn 接收移位寄存器传送到 UAnRX 寄存器。

传送到 UAnRX 寄存器也会引发接收完成中断请求信号 (INTUAnR)。

(8) UARTAn 发送移位寄存器

该移位寄存器用于将来自 UAnTX 寄存器的并行数据转换为串行数据。

当一个字节的数据从 UAnTX 寄存器传送过来时，移位寄存器的数据从 TXDAn 引脚输出。

不能直接操作该寄存器。

(9) UARTAn 发送数据寄存器 (UAnTX)

UAnTX 寄存器是 8 位发送数据缓存器。当发送的数据被写入 UAnTX 寄存器后，开始发送。当数据能够写入 UAnTX 寄存器时（当一帧数据从 UAnTX 寄存器传送到 UARTAn 发送移位寄存器时），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)

UAnCTL0 寄存器是 8 位寄存器，用于控制 UARTAn 的串行传输操作。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 10H。

(1/2)

复位后: 10H R/W 地址: UA0CTL0 FFFFFFFA00H, UA1CTL0 FFFFFFFA10H,
UA2CTL0 FFFFFFFA20H, UA3CTL0 FFFFFFFA30H

	<7>	<6>	<5>	<4>	3	2	1	0
UAnCTL0	UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL

(n = 0 至 3)

UAnPWR	UARTAn 操作 控制
0	禁止 UARTAn 操作 (UARTAn 异步复位)
1	允许 UARTAn 操作
UARTAn 操作由UAnPWR 位控制。通过清除UAnPWR位为0将 TXDAn 引脚输出固定为高电平 (如果UAnOPT0.UAnTDL 位 = 1, 则固定为低电平)。	

UAnTXE	发送操作使能
0	发送操作禁止
1	发送操作使能
<ul style="list-style-type: none"> 若要开始发送，设置 UAnPWR 位为 1 然后设置 UAnTXE 位为 1。 若要停止发送，清除 UAnTXE 位为0 然后清除 UAnPWR 位为 0。 若要初始化发送单元，清除UAnTXE位为0，等待两个周期的基础时钟，然后再次设置 UAnTXE 位为 1。否则无法执行初始化 (关于基础时钟，参见 15.7 (1) (a) 基础时钟)。 	

UAnRXE	接收操作使能
0	接收操作禁止
1	接收操作使能
<ul style="list-style-type: none"> 若要开始接收，设置 UAnPWR 位为 1 然后 设置 UAnRXE 位为 1。 若要停止接收，清除 UAnRXE 位为 0 然后清除 UAnPWR 位为 0。 若要初始化接收单元，清除 UAnRXE 位为 0, 等待两个周期的基础时钟，然后再次设置 UAnRXE 位为 1。否则无法执行初始化 (关于基础时钟，参见 15.7 (1) (a) 基础时钟)。 	

UAnDIR	传输方向选择
0	MSB先行
1	LSB先行
<ul style="list-style-type: none"> 只有 UAnPWR 位 = 0 或 UAnTXE 位 = UAnRXE 位 = 0 时，该寄存器才可以重写。 在 LIN 格式下进行发送和接收时，设置 UAnDIR 位为 1。 	

UAnPS1	UAnPS0	发送期间奇偶校验的选择	接收期间奇偶校验的选择
0	0	无奇偶校验输出	无奇偶校验接收
0	1	0 奇偶校验输出	0 奇偶校验接收
1	0	奇校验输出	奇校验检测
1	1	偶校验输出	偶校验检测
<ul style="list-style-type: none"> 只有 UAnPWR 位 = 0 或 UAnTXE 位 = UAnRXE 位 = 0 时，该寄存器才可以重写。 如果接收期间选择“0 奇偶校验接收”，则不进行奇偶校验检查。因此不设置 UAnSTR.UAnPE 位。 在 LIN 格式下进行发送和接收时，清除 UAnPS1 和 UAnPS0 位为 00。 			

UAnCL	1 帧的发送/接收数据字符长度的说明
0	7 位
1	8 位
<ul style="list-style-type: none"> 只有 UAnPWR 位 = 0 或 UAnTXE 位 = UAnRXE 位 = 0 时，该寄存器才可以重写。 在 LIN 格式下进行发送和接收时，设置 UAnCL 位为 1。 	

UAnSL	发送数据的停止位长度的说明
0	1 位
1	2 位
只有 UAnPWR 位 = 0 或 UAnTXE 位 = UAnRXE 位 = 0 时，该寄存器才可以重写。	

备注 关于奇偶校验的详情，参见 15.6.9 奇偶类型和操作。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

更多详情, 参见 15.7 (2) UARTAn 控制寄存器 1 (UAnCTL1)。

(3) UARTAn 控制寄存器 2 (UAnCTL2)

更多详情, 参见 15.7 (3) UARTAn 控制寄存器 2 (UAnCTL2)。

(4) UARTAn 选项控制寄存器 0 (UAnOPT0)

UAnOPT0 寄存器是 8 位寄存器, 用于控制 UARTAn 寄存器的串行传输操作。

该寄存器可进行字节读写或按位读写。

系统复位后, 该寄存器被设为 14H。

(1/2)

复位后: 14H R/W 地址: UA0OPT0 FFFFA03H, UA1OPT0 FFFFA13H,
UA2OPT0 FFFFA23H, UA3OPT0 FFFFA33H

	<7>	6	5	4	3	2	1	0
UAnOPT0	UAnSRF	UAnSRT	UAnSTT	UAnSLS2	UAnSLS1	UAnSLS0	UAnTDL	UAnRDL

(n = 0 至 3)

UAnSRF	SBF接收标志
0	UAnCTL0.UAnPWR 位 = UAnCTL0.UAnRXE 位 = 0 时。 在SBF 接收正常结束时。
1	SBF 接收中
<ul style="list-style-type: none">• LIN 通信期间, 对 SBF (同步间隔场) 的接收进行判断。• 发生SBF接收错误时, UAnSRF 位保持为1, 然后SBF接收重新开始。• UAnSRF 位是只读位。	

UAnSRT	SBF 接收 触发
0	-
1	SBF 接收 触发
<ul style="list-style-type: none">• LIN 通信期间该位是SBF接收触发位, 读出值总为“0”。• 为进行SBF 接收, 需设置 UAnSRT 位 (为 1) 以允许SBF接收。• 设置 UAnPWR 位 = UAnRXE 位 = 1后设置UAnSRT 位。	

UAnSTT	SBF 发送 触发
0	-
1	SBF 发送 触发
<ul style="list-style-type: none">• LIN 通信期间该位是SBF 发送触发位, 读出值总为“0”。• 置 UAnPWR 位 = UAnTXE 位 = 1后设置UAnSTT 位。	

注意事项 SBF 接收 (UAnSRF 位 = 1) 期间, 不要设置 UAnSRT 和 UAnSTT 位 (1)。

UAnSLS2	UAnSLS1	UAnSLS0	SBF 发送长度选择
1	0	1	13-位 输出(复位默认值)
1	1	0	14-位 输出
1	1	1	15-位 输出
0	0	0	16-位 输出
0	0	1	17-位 输出
0	1	0	18-位 输出
0	1	1	19-位 输出
1	0	0	20-位 输出
UAnPWR 位 = 0 或 UAnTXE 位 = 0时, 可设置该寄存器。			

UAnTDL	发送数据的电平
0	传输数据正常输出
1	传输数据反相输出
<ul style="list-style-type: none"> TXDAn 引脚的输出电平可使用UAnTDL 位进行反相。 UAnPWR 位 = 0 或UAnTXE 位 = 0时, 可设置该寄存器。 	

UAnRDL	接收数据电平
0	传输数据正常输入
1	传输数据反相输入
<ul style="list-style-type: none"> RXDAn 引脚的输入电平可使用UAnRDL 位进行反相。 UAnPWR 位 = 0 或 UAnRXE 位 = 0时, 可设置该寄存器。 	

(5) UARTAn 状态寄存器 (UAnSTR)

UAnSTR 寄存器是 8 位寄存器, 显示 UARTAn 的传输状态和接收错误的内容。

该寄存器可进行字节读写或按位读写, 但是, UAnTSF 位为只读位, 而 UAnPE、UAnFE 和 UAnOVE 位既可读也可写。但是, 这些位只能写入 0 来清除, 而不能通过写 1 来置位 (即便写入 1, 其值仍保持不变)。

初始化条件如下:

寄存器/位	初始条件
UAnSTR 寄存器	<ul style="list-style-type: none"> 清零 UAnCTL0.UAnPWR = 0
UAnTSF 位	<ul style="list-style-type: none"> UAnCTL0.UAnTXE = 0
UAnPE 位、UAnFE 位、UAnOVE 位	<ul style="list-style-type: none"> 写入 0 UAnCTL0.UAnRXE = 0

复位后: 00H R/W 地址: UA0STR FFFFFFFA04H, UA1STR FFFFFFFA14H,
UA2STR FFFFFFFA24H, UA3STR FFFFFFFA34H

	<7>	6	5	4	3	<2>	<1>	<0>
UAnSTR (n = 0 至 3)	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

UAnTSF	传输状态标志
0	<ul style="list-style-type: none"> 已经设置UAnPWR位=0 或 UAnRXE=0。 传输完成之后，UAnTX寄存器没有发来下一个数据。
1	写入 UAnTX 寄存器
<p>进行连续发送时，UAnTSF位总是为1。</p> <p>初始化发送单元时，在进行初始化前确保UAnTSF位= 0。</p> <p>如果在UAnTSF位= 1时执行初始化，则发送数据无法保证。</p>	

UAnPE	奇偶校验 错误标志
0	<ul style="list-style-type: none"> 已经设置UAnPWR位=0 或 UAnRXE=0。 已经写入 0 时。
1	接收期间数据的奇偶位与奇偶校验位不匹配时。
<ul style="list-style-type: none"> 通过设置UAnCTL0.UAnPS1 和 UAnCTL0.UAnPS0 位来控制UAnPE位的操作。 UAnPE位可读可写，但是只能写入0进行清除，而不能写入1进行置位，如果向该位写入1，其值保持不变。 	

UAnFE	帧错误标志
0	<ul style="list-style-type: none"> 已经设置UAnPWR位=0 或 UAnRXE=0。 已经写入 0 时。
1	接收期间没有检测到停止位。
<ul style="list-style-type: none"> 只检测接收数据停止位的第一位，而无需考虑UAnCTL0.UAnSL位的值。 UAnFE位可读可写，但是只能写入0进行清除，而不能写入1进行置位，如果向该位写入1，其值保持不变。 	

UAnOVE	溢出错误标志
0	<ul style="list-style-type: none"> 已经设置UAnPWR位=0 或 UAnRXE=0。 已经写入 0 时。
1	接收数据已经设置到 UAnRX 寄存器 并且在接收数据被读取之前下一个接收操作已经完成。
<ul style="list-style-type: none"> 发生溢出错误时，数据被丢弃，下一个接收数据不会被写入接收缓冲器。 UAnOVE位可读可写，但是只能写入0进行清除，而不能写入1进行置位，如果向该位写入1，其值保持不变。 	

UAnRX 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。

1 个字节的数据接收完成时，将存储于接收移位寄存器的数据传送到 UAnRX 寄存器。

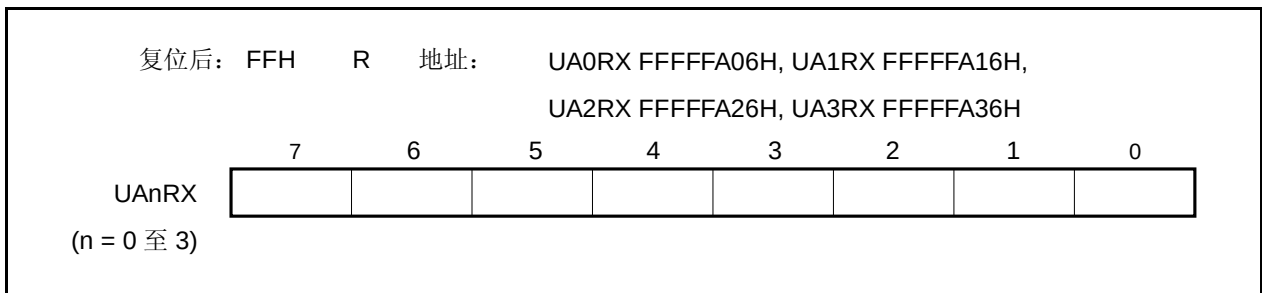
LSB 接收期间，数据长度指定为 7 位时，接收数据传输至 UAnRX 寄存器的第 6 位至第 0 位，MSB 位总为 0。

MSB 接收期间，将接收数据传输到 UAnRX 的第 7 位至第 1 位，且 LSB 位总为 0。

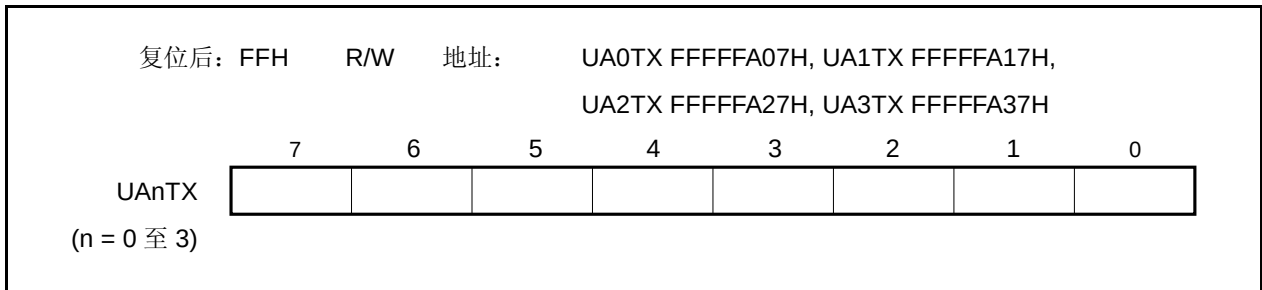
发生溢出（UAnOVE）错误时，此时的接收数据不会被传送到 UAnRX 寄存器，而是被丢弃。

该寄存器是只读寄存器，按字节读取。

除了复位输入之外，将 UAnCTL0.UAnPWR 位清除为 0，UAnRX 寄存器被置为 FFH。



UAnTX 寄存器是 8 位寄存器，用于设置发送数据。
该寄存器可进行字节读取或写入。
系统复位后，该寄存器被设为 FFH。



15.5 中断请求信号

UARTAn 会产生以下两种中断请求信号：

- 接收完成中断请求信号 (INTUAnR)
- 发送使能中断请求信号 (INTUAnT)

这两种中断请求信号的默认优先级为接收完成中断请求信号优先，然后为发送使能中断请求信号。

表 15-2. 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 (INTUAnR)

接收使能状态下，数据转移至接收移位寄存器并发送至 UAnRX 寄存器时，接收完成中断请求信号输出。

发生接收错误时，接收完成中断请求信号也会继续输出。因此，响应接收完成中断请求信号并读取数据时，读取 UAnSTR 寄存器并检查接收结果是否错误。

在接收禁止状态下，不产生接收完成中断请求信号。

(2) 发送使能中断请求信号 (INTUAnT)

在发送允许状态下，当发送数据从 UAnTX 寄存器传输至 UARTAn 发送移位寄存器，产生发送使能中断请求信号。

15.6 操作

15.6.1 数据格式

执行全双工串行数据的接收和发送。

如图 15-6 所示，一个发送/接收数据帧的数据由起始位、字符位、校验位以及停止位组成。

使用 UAnCTL0 寄存器指定一个数据帧内字符位长度、校验段，指定停止位长度，以及 MSB/LSB 传输方式。

另外，UART 中关于 TXDAn 位的输出/反相控制由 UAnOPT0.UAnTDL 位来执行。

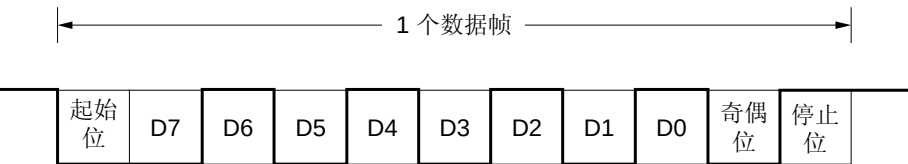
- 起始位 1 位
- 字符位 7 位/8 位
- 奇偶校验位 奇检验/偶校验/0 校验/无校验
- 停止位 1 位/2 位

图 15-5. UARTA 发送/接收数据格式

(a) 8 位数据长度, LSB, 偶校验, 一个停止位, 传输数据: 55H



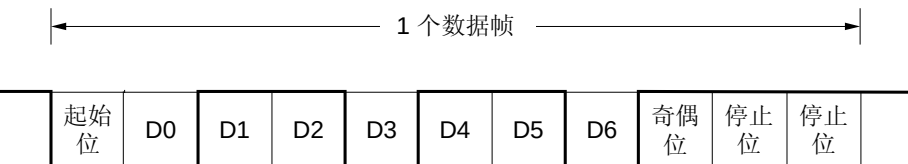
(b) 8 位数据长度, MSB, 偶校验, 一个停止位, 传输数据: 55H



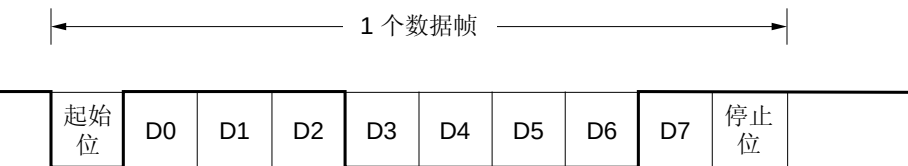
(c) 8 位数据长度, MSB, 偶校验, 一个停止位, 传输数据: 55H, TXDAn 反相



(d) 7 位数据长度, LSB, 奇校验, 2 个停止位, 传输数据: 36H



(e) 8 位数据长度, LSB, 无校验, 一个停止位, 传输数据: 87H



15.6.2 SBF 发送/接收格式

V850ES/JJ3 具有 SBF（同步间隔场）发送/接收控制功能，可以支持 LIN 功能的使用。

备注 LIN 表示局域内联网，是一种低速（1 至 20 kbps）串行通信协议，有助于降低汽车网络的成本。LIN 通信为单主机通信，一台主机最多可以连接 15 台从机。LIN 从机一般是控制开关、传动装置和传感器，且这些装置可通过 LIN 网络连接到 LIN 主机上。通常，LIN 主机接入诸如 CAN（控制器区域网络）之类的网络。另外，LIN 总线使用单线方式且通过遵循 ISO9141 规范的收发器连接至各个节点。LIN 协议中，主机以某一波特率信息发送一个数据帧，从机将其接收并校正波特率错误。因此，从机波特率错误在±15%范围内，通信可以正常进行。

图 15-6 和 15-7 表明 LIN 的发送和接收的处理。

图 15-6. LIN 发送操作概要

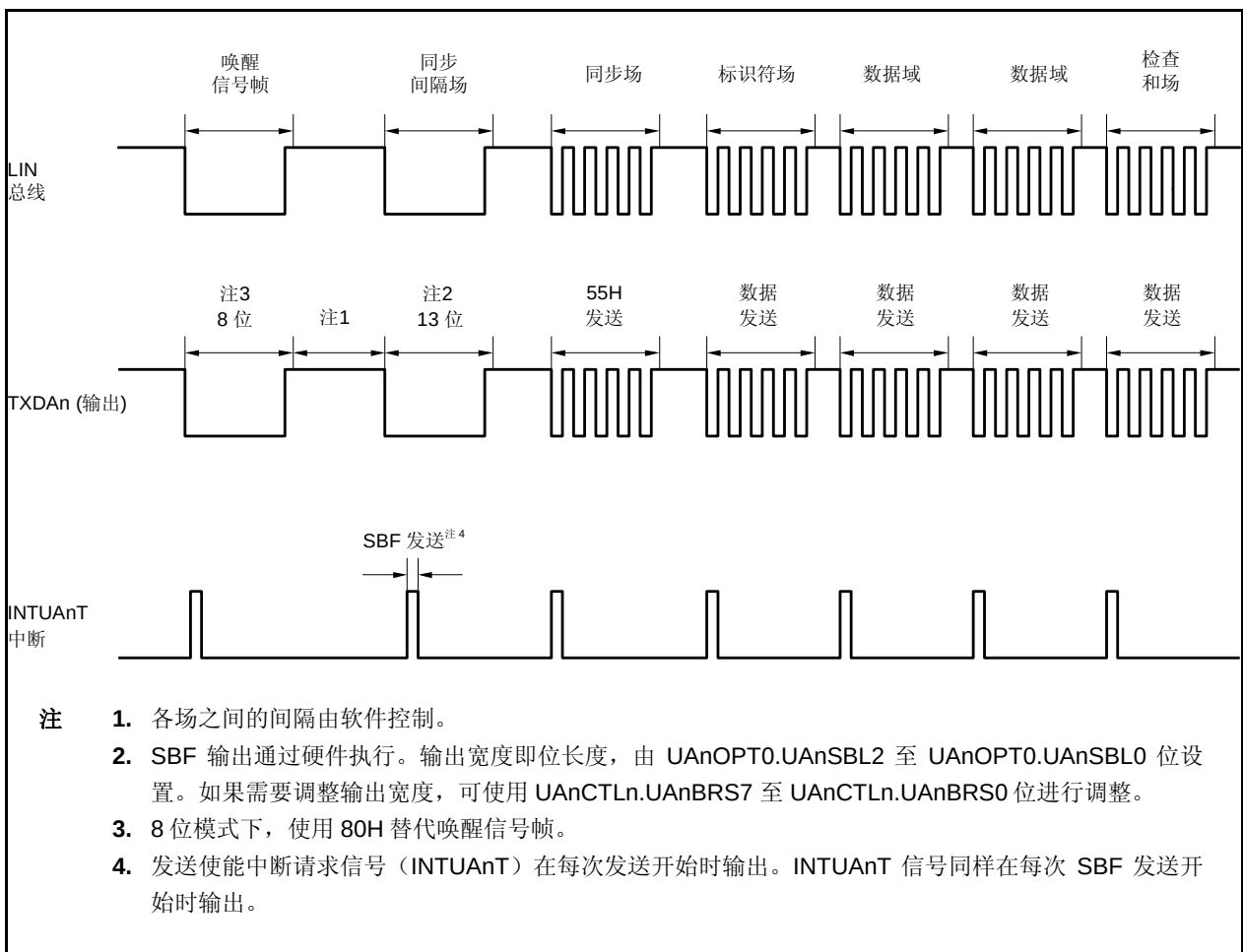
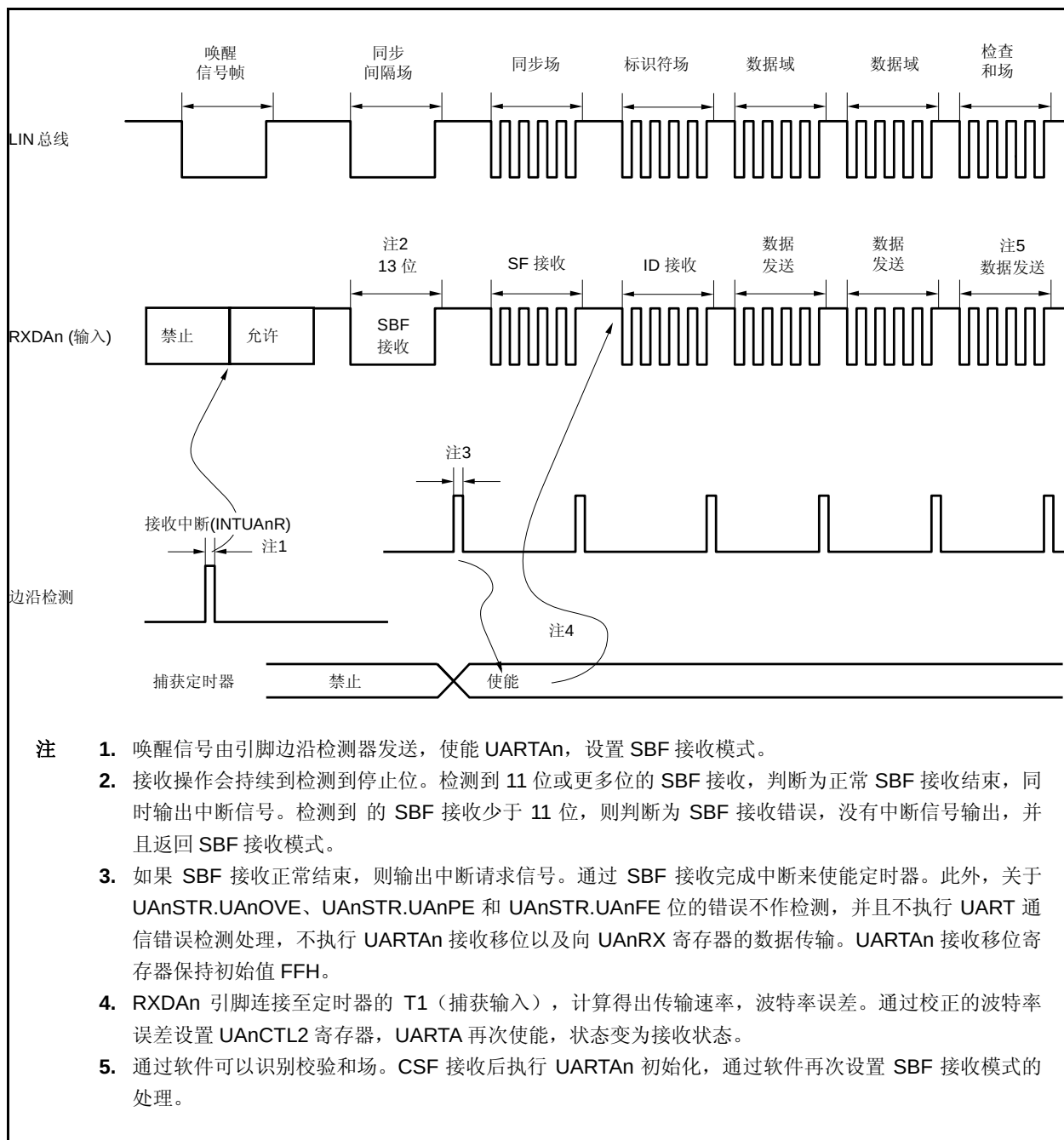


图 15-7. LIN 接收处理概要

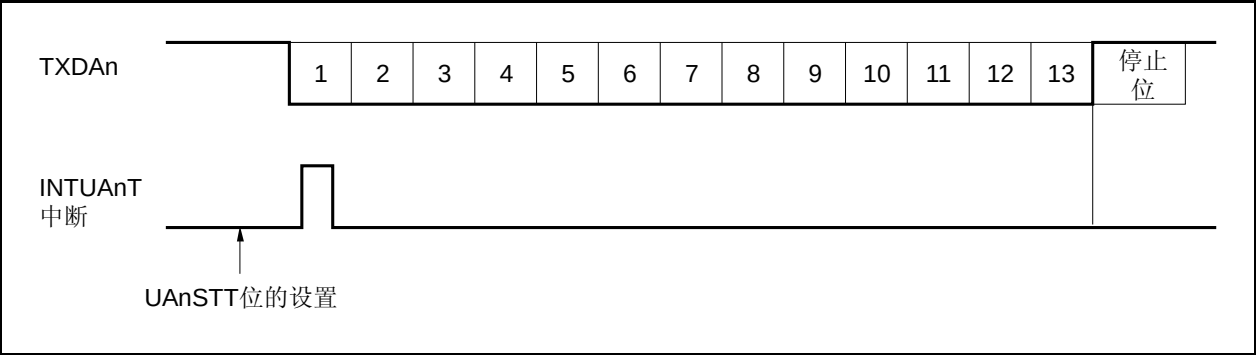


15.6.3 SBF 发送

当 UAnCTL0.UAnPWR 位 = UAnCTL0.UAnTXE 位 = 1 时，进入发送使能状态，通过设置 SBF 发送触发信号 (UAnOPT0.UAnSTT 位) 为 1，开始 SBF 发送。

之后，输出由 UAnOPT0.UAnSLS2 至 UAnOPT0.UAnSLS0 位指定的 13 位至 20 位宽度的低电平。SBF 发送开始时，产生发送使能中断请求信号 (INTUAnT)。在 SBF 发送结束后，UAnSTT 位自动清零。之后，恢复 UART 发送模式。暂停发送，直到下一个待发送数据写入 UAnTX 寄存器，或者直到 SBF 发送触发信号 (UAnSTT 位) 被置位。

图 15-8. SBF 发送



15.6.4 SBF 接收

设置 UAnCTL0.UAnPWR 位为 1，然后设置 UAnCTL0.UAnRXE 位为 1，进入接收使能状态。

将 SBF 接收触发 (UAnOPT0.UAnSTR 位) 置 1，来进入 SBF 接收等待状态。

SBF 接收等待状态下，RXDAn 引脚被监控并且开始检测起始位。和 UART 接收等待状态类似。

在检测到起始位之后，根据设置的波特率开始接收并且内部计数器开始向上计数。

接收到一个停止位时，如果 SBF 宽度为 11 位或更多位，则判断为正常操作，输出接收完成中断请求信号 (INTUAnR)。UAnOPT0.UAnSRF 位自动清除，SBF 接收结束。对 UAnSTR.UAnOVE、UAnSTR.UAnPE 和 UAnSTR.UAnFE 位的错误不作检测，不执行 UART 通信错误检测处理，不执行 UARTAn 接收移位以及向 UAnRX 寄存器的数据传输，UARTAn 接收移位寄存器保持初始值 FFH。如果 SBF 宽度为 10 位或更少位，则接收终止，作为错误处理且不输出中断，并返回 SBF 接收模式。此时，UAnSRF 位不被清除。

注意事项 1. 如果数据接收期间 SBF 被发送，会发生一个帧错误。

2. SBF 接收 (UAnSRF = 1) 期间，不要将 SBF 接收触发位 (UAnSRT) 和 SBF 发送触发位 (UAnSTT) 设置为 1。

图 15-9. SBF 接收 (1/2)

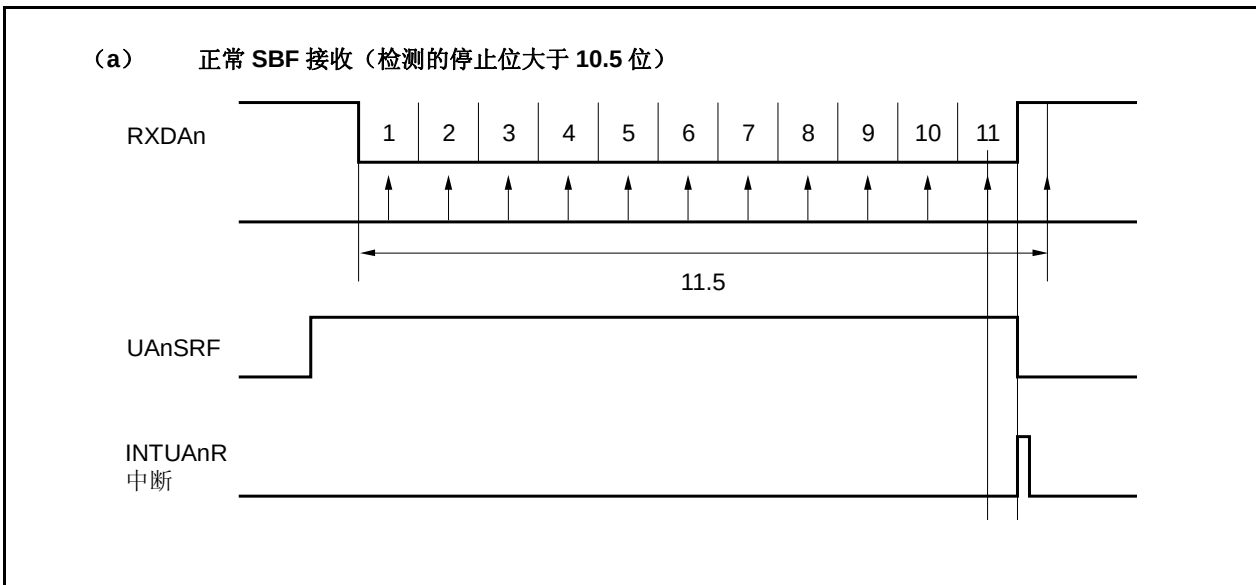
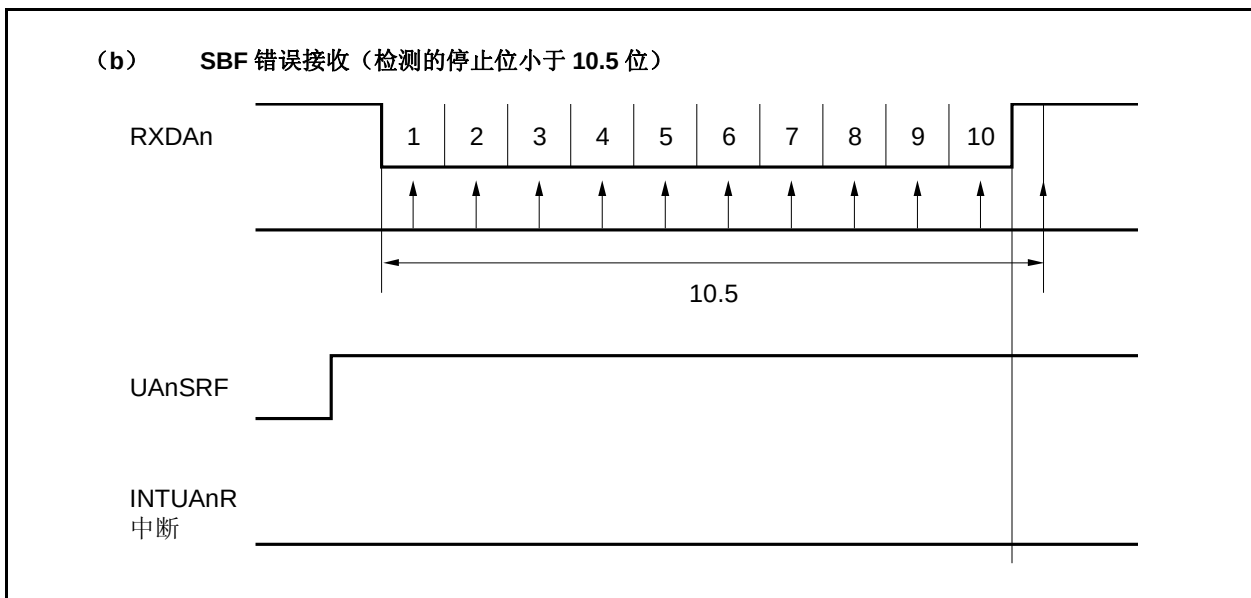


图 15-9. SBF 接收 (2/2)



15.6.5 UART 发送

设置 UAnCTL0.UAnPWR 位为 1，TXDAn 引脚输出高电平。

之后，设置 UAnCTL0.UAnTXE 位为 1，进入发送允许状态，将发送数据写入 UAnTX 寄存器启动发送。起始位、校验位以及停止位自动添加。

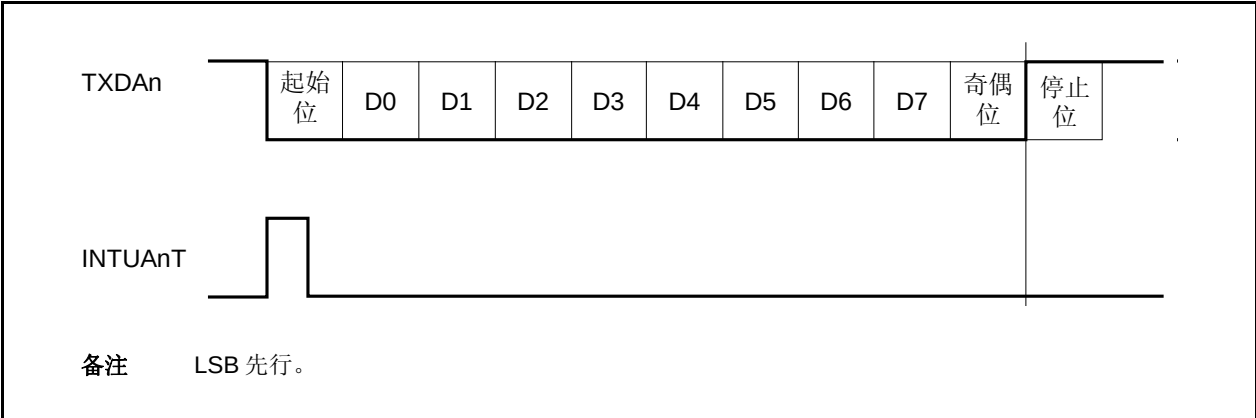
由于 UARTAn 中不提供 CTS（发送允许信号）输入引脚，所以需要使用一个端口来检查发送目的方已经允许接收。

当发送操作启动之时，UAnTX 寄存器内的数据被传送到 UARTAn 发送移位寄存器。

UAnTX 寄存器的数据发送至 UARTAn 发送移位寄存器的操作完成时，产生发送使能中断请求信号（INTUAnT），之后 UARTAn 发送移位寄存器的内容输出至 TXDAn 引脚。

INTUAnT 信号产生后，允许向 UAnTX 寄存器写入下一个发送数据。

图 15-10. UART 发送



15.6.6 连续发送过程

当 UARTAn 发送移位寄存器启动移位操作时，UARTAn 可以写入下一个数据到 UAnTX 寄存器，UARTAn 发送移位寄存器的发送时序可以由发送使能中断请求信号（INTUAnT）来判断。

在传输过程中将下一个待发送数据写入 UAnTX 寄存器使得传输效率更高。

注意事项 连续发送执行期间进行发送初始化时，请确保 UAnSTR.UAnTSF 位为 0，然后再执行初始化。UAnTSF 位为 1 时进行初始化，则发送数据无法保证。

图 15-11. 连续发送处理框图

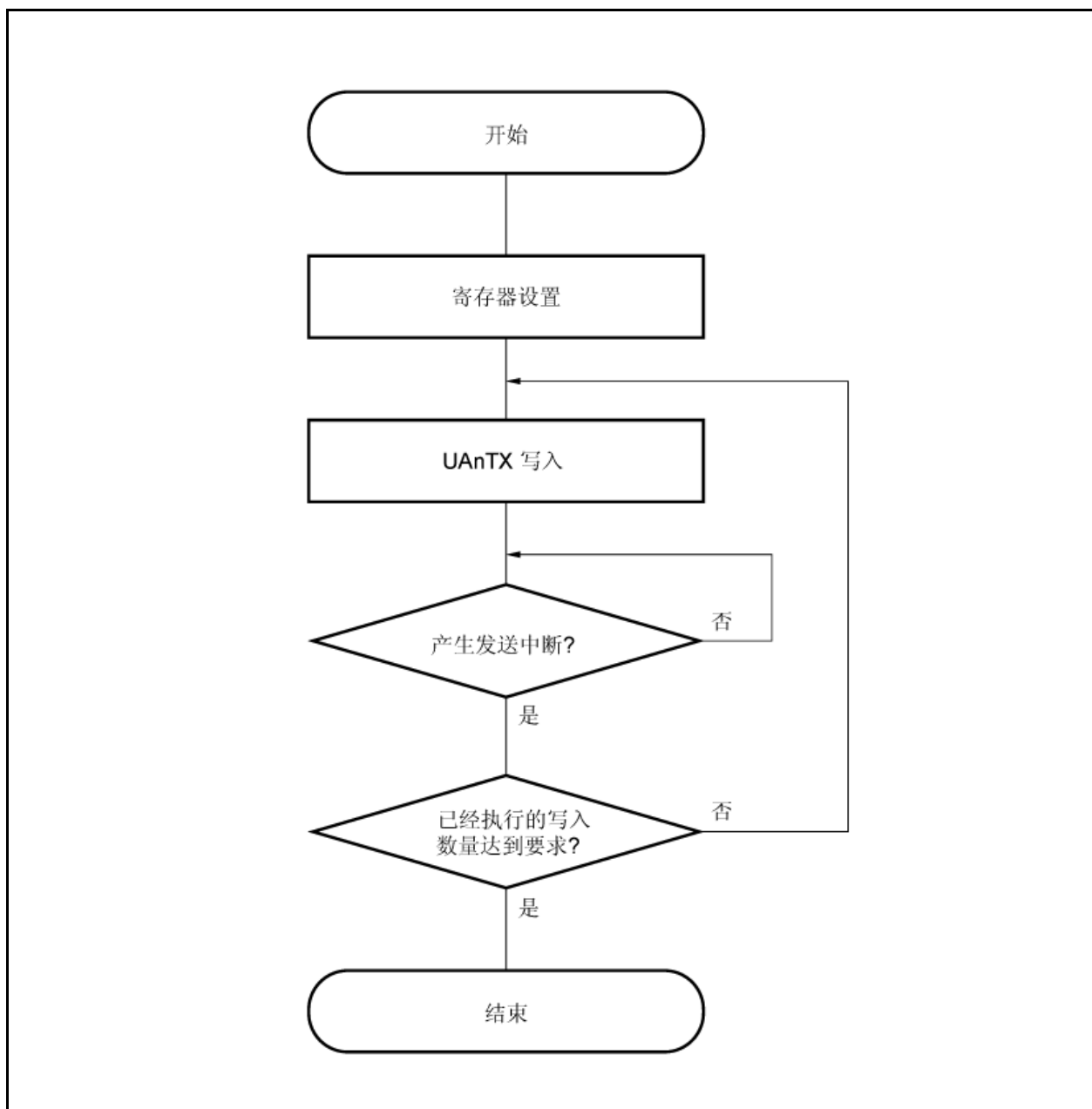
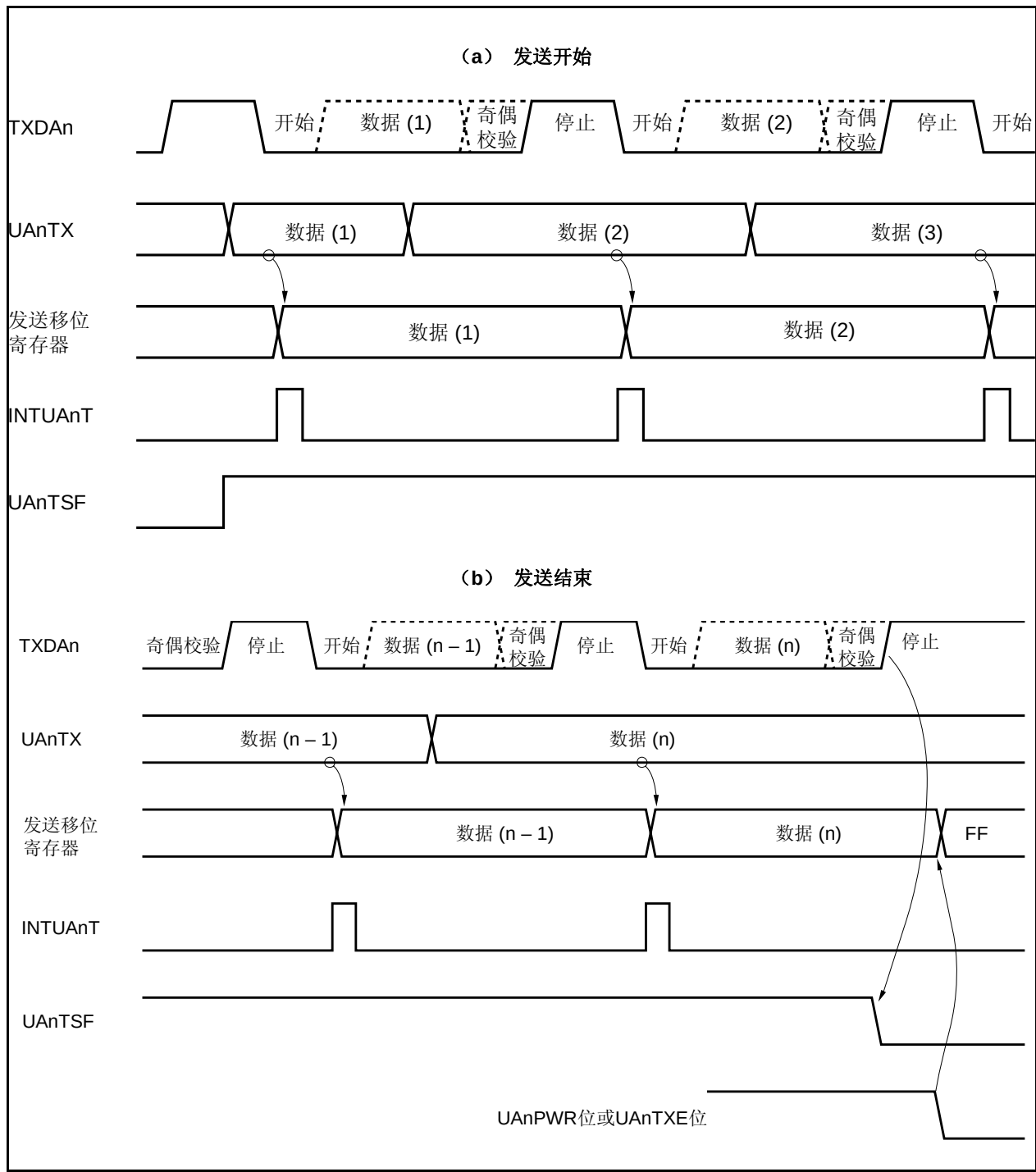


图 15-12. 连续发送操作时序



15.6.7 UART 接收

设置 UAnCTL0.UAnPWR 位为 1，然后设置 UAnCTL0.UAnRXE 位为 1，进入接收等待状态。接收等待状态下，RXDAn 引脚被监测并且进行起始位的检测。

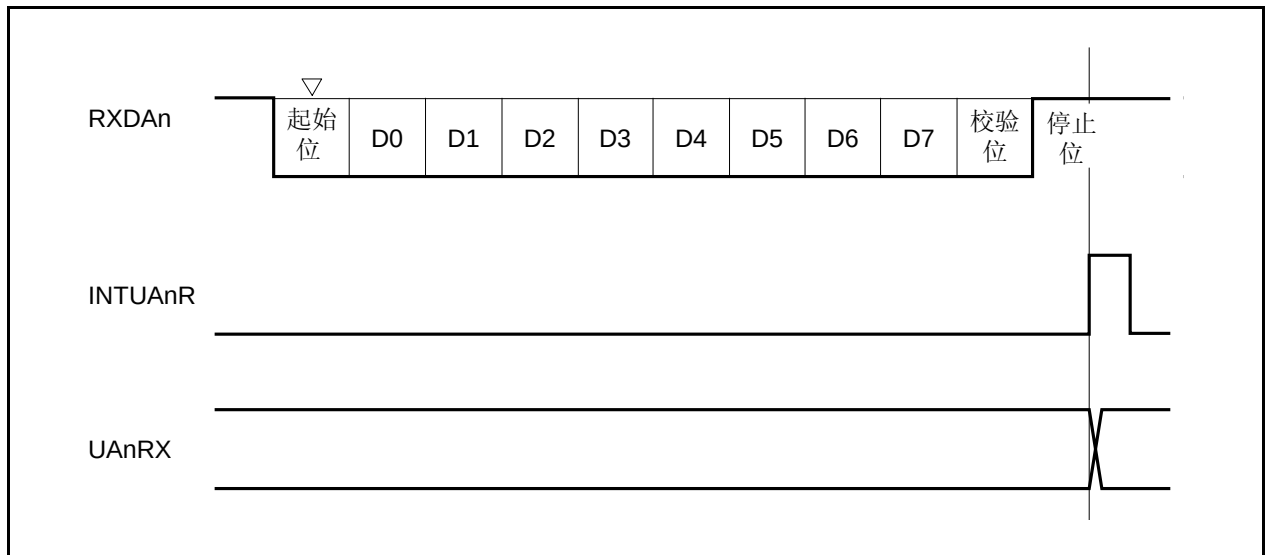
应用两步检测程序来进行起始位的检测。

首先检测到 RXDAn 引脚的上升沿，在下降沿处开始采样。如果 RXDAn 引脚在起始位采样点为低电平，确认是起始位。确认起始位之后，开始接收操作，按照设定的波特率，串行数据被存入 UARTAn 接收移位寄存器。

接收到停止位时，会输出接收完成中断请求信号 (INTUAnR) 时，UARTAn 接收移位寄存器的数据写入 UAnRX 寄存器。但是，如果发生溢出 (UAnSTR.UAnOVE 位) 错误，此时接收到的数据不写入 UAnRX 寄存器，而是被丢弃。

即使接收期间发生奇偶错误 (UAnSTR.UAnPE 位) 或帧错误 (UAnSTR.UAnFE 位)，接收还会继续进行，直到接收到第一个停止位，并在接收完成后输出 INTUAnR 信号。

图 15-13. UART 接收



- 注意事项**
1. 即使发生接收错误，也必须读取 UAnRX 寄存器。如果不读取 UAnRX，则下一个数据接收期间发生溢出错误，并且接收错误会不断地连续发生。
 2. 认为接收操作期间只有一个停止位。忽略第二个停止位。
 3. 当接收完成时，在产生接收结束中断请求信号 (INTUAnR) 之后读取 UAnRX 寄存器，并对 UAnPWR 位或 UAnRXE 位清 0。如果在 INTUAnR 信号产生之前对 UAnPWR 位或 UAnRXE 位清 0，则无法保证对 UAnRX 寄存器值的正确读取。
 4. 如果 UARTAn 的接收完成处理 (INTUAnR 信号产生) 和 UAnPWR 位=0 或 UAnRXE 位=0 发生冲突，则即 UAnRX 寄存器内没有存储数据，也可能产生 INTUAnR 信号。
为了完成接收而无需等待 INTUAnR 信号的产生，请确保将 UAnRIC 寄存器的中断请求标志 (UAnRIF) 清除 (0)，在中断控制寄存器 (UAnRIC) 的中断屏蔽标志 (UAnRMK) 置位 (1) 之后，然后设置 UAnPWR 位=0 或 UAnRXE 位=0。

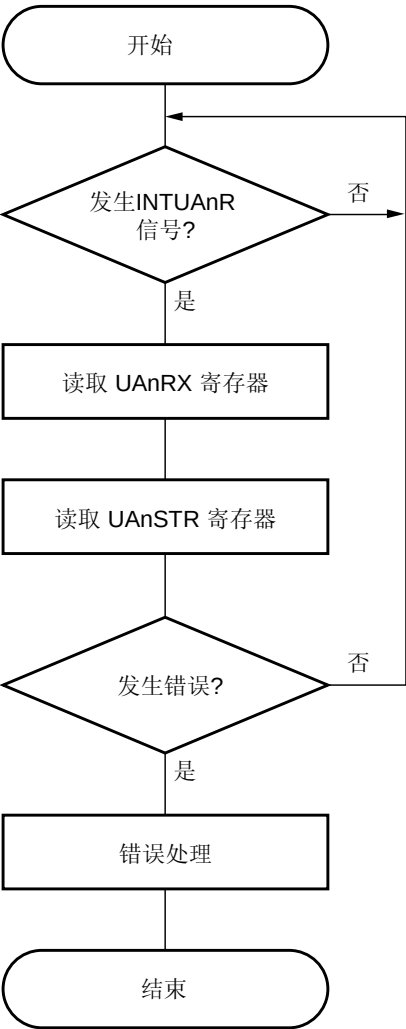
15.6.8 接收错误

接收过程中的错误有 3 种类型：奇偶检验错误、帧错误以及溢出错误。数据接收结果错误标志被设置在 UAnSTR 寄存器中，当错误发生时，输出接收完成中断请求信号（INTUAnR）。

通过读取 UAnSTR 寄存器的内容，可以确定在接收过程中发生了哪种错误。

读取完成后，写入 0 来清除接收错误标志。

- 接收数据读取流程：



注意事项 发生 INTUAnR 信号时，必须读取 UAnSTR 寄存器来检查错误。

- 接收错误原因

错误标志	接收错误	原因
UAnPE	奇偶错误	接收到的校验位和设定校验位不匹配
UAnFE	帧错误	未检测到停止位
UAnOVE	溢出错误	在数据从接收缓冲读出之前，下一个数据接收完成

发生接收错误时，根据错误类型执行以下步骤：

- 奇偶错误

如果是因为例如接收线上的噪声问题导致接收到错误数据，则丢弃接收到的数据并重新发送。

- 帧错误

接收端和发送端之间可能发生波特率错误，或者错误的检测到起始位。因为对通信格式来说这是致命错误，确认发送端操作停止，各自进行初始化处理，然后重新开始通信。

- 溢出错误

如果在读出接收数据之前，下一个数据接收完成，1 帧数据会丢弃。如果需要该数据，则进行重新发送。

注意事项 如果在连续接收期间发生接收错误中断，在下一个数据接收完成之前，必须读出 **UAnSTR** 寄存器的内容，然后进行错误处理。

15.6.9 奇偶类型和操作

注意事项 使用 LIN 功能时，将 UAnCTL0 寄存器的 UAnPS1 位和 UAnPS0 位固定为 00。

奇偶位用于检测通信数据中的位错误。通常情况下，发送端和接收端应该使用相同的奇偶校验。
在使用奇校验和偶校验情况下，可以检测到奇数计数的位错误。而在 0 奇偶校验和无校验情况下，检测不到错误。

(a) 偶校验

(i) 在发送期间

发送数据之中位值为“1”的数量，包括奇偶校验位，控制其数量为偶数。奇偶位的值如下：

- 发送数据中位值为“1”的数量为奇数： 1
- 发送数据中位值为“1”的数量为偶数： 0

(ii) 在接收期间

计算接收数据中位值为“1”的数量，包括校验位，如果为奇数，则输出奇偶校验错误。

(b) 奇校验

(i) 在发送期间

与偶校验相对，发送数据中位值为“1”的数量，包括奇偶校验位，控制其数量为奇数。奇偶位的值如下：

- 发送数据中位值为“1”的数量为奇数： 0
- 发送数据中位值为“1”的数量为偶数： 1

(ii) 在接收期间

计算接收数据中位值为“1”的数量，包括校验位，如果为偶数，则输出奇偶校验错误标志。

(c) 0 奇偶校验

在发送期间，奇偶校验位总是“0”，而与发送数据无关。

在接收期间，不执行奇偶校验。因而，无论奇偶位为“0”或“1”，均不会产生奇偶错误。

(d) 无奇偶校验

不向发送数据附加奇偶位。

设定无奇偶位进行接收。因为无奇偶位所以没有奇偶错误发生。

15.6.10 接收数据噪声过滤器

该滤波器使用预分频器输出的基础时钟对 RXDAn 引脚进行采样。

如果读到两次相同的采样值，则匹配检测器输出改变，并且被采样的 RXDAn 信号作为输入数据。所以不超过两个时钟宽度的数据被判定为噪声，不会被送交给内部电路（参见 图 15-15）。有关基础时钟，参见 15.7 (1) (a) 基础时钟。

另外，由于电路如图 15-14 所示，所以，接收操作的处理过程与外部信号状态相比，延迟了 3 个时钟周期。

图 15-14. 噪声过滤电路

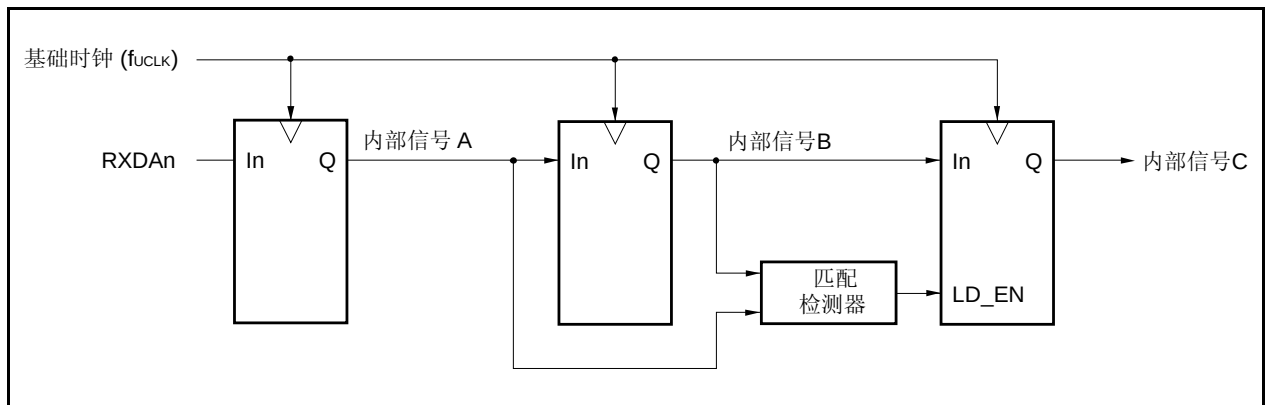
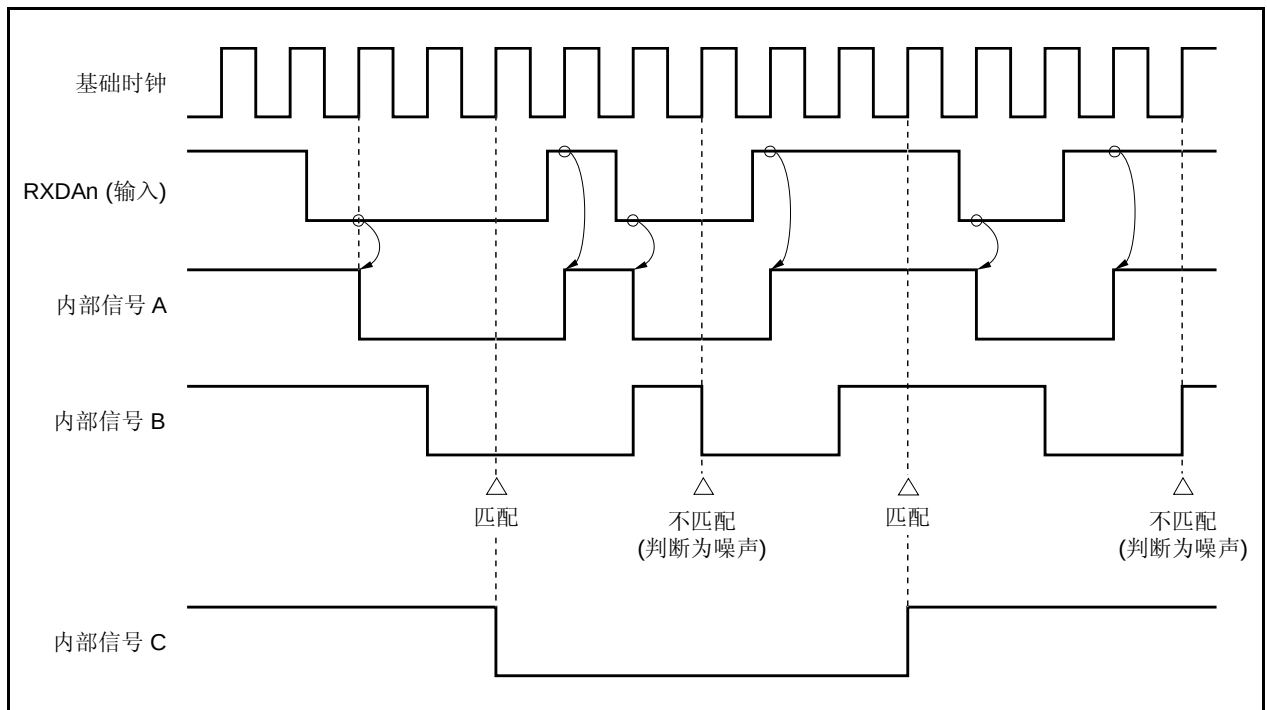


图 15-15. RXDAn 信号判定为噪声的时序



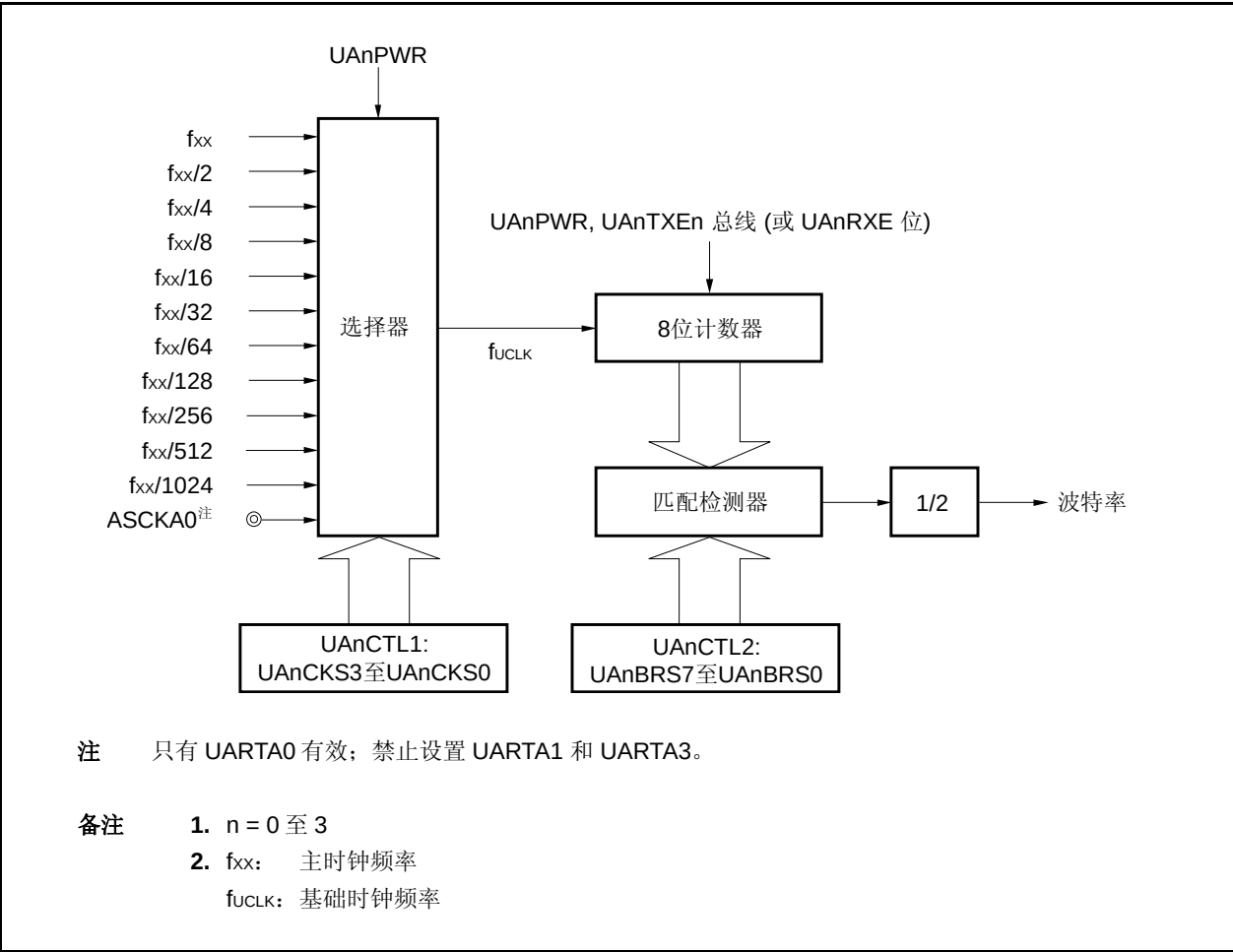
15.7 专用波特率发生器

专用波特率发生器包括一个时钟源选择器和一个 8 位可编程计数器，在 UARTAn 发送/接收期间产生串行时钟。关于串行时钟，对各个通道可以选择专用波特率发生器输出。

一个 8 位计数器用于发送，另一个用于接收。

(1) 波特率发生器的配置

图 15-16. 波特率发生器的配置



(a) 基础时钟

当 UAnCTL0.UAnPWR 位为 1 时，由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择的时钟被提供给 8 位计数器。该时钟称为基础时钟 (f_{UCLK})。

(b) 串行时钟的产生

通过设置 UAnCTL1 寄存器和 UAnCTL2 ($n = 0$ 至 3) 寄存器，来产生一个串行时钟。

基础时钟由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择。

8 位计数器的分频值可以使用 UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位进行设置。

(2) UARTn 控制寄存器 1 (UAnCTL1)

UAnCTL1 寄存器是 8 位寄存器，用于选择 UARTAn 基础时钟。

该寄存器可进行字节读取或写入。

系统复位后，该寄存器被设为 00H。

注意事项 在改写 UAnCTL1 寄存器之前，将 UAnCTL0.UAnPWR 位清 0。

复位后: 00H R/W 地址: UA0CTL1 FFFFFFFA01H, UA1CTL1 FFFFFFFA11H,
UA2CTL1 FFFFFFFA21H, UA3CTL1 FFFFFFFA31H

	7	6	5	4	3	2	1	0
UAnCTL1	0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0

(n = 0至3)

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	基础时钟(f _{uclk}) 的选择
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 (ASCKA0 引脚)
其它情况				禁止设置

注 只有 UARTA0 有效；禁止设置 UARTA1 和 UARTA3。

备注 f_{xx}: 主时钟频率

(3) UARTAn 控制寄存器 2 (UAnCTL2)

UAnCTL2 寄存器是 8 位寄存器，用于选择 UARTAn 的波特率（串行传输速度）。

该寄存器可进行字节读取或写入。

系统复位后，该寄存器被设为 FFH。

注意事项 在改写 UAnCTL2 寄存器之前，将 UAnCTL0.UAnPWR 位清除为 0，或将 UAnTXE 和 UAnRXE 位清除为 0。

复位后： FFH R/W 地址： UA0CTL2 FFFFFFFA02H, UA1CTL2 FFFFFFFA12H,
UA2CTL2 FFFFFFFA22H, UA3CTL2 FFFFFFFA32H

	7	6	5	4	3	2	1	0
UAnCTL2	UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0
(n = 0 至 3)								

UAn BRS7	UAn BRS6	UAn BRS5	UAn BRS4	UAn BRS3	UAn BRS2	UAn BRS1	UAn BRS0	缺省值 (k)	串行 时钟
0	0	0	0	0	0	x	x	x	禁止 设置
0	0	0	0	0	1	0	0	4	f _{UCLK} /4
0	0	0	0	0	1	0	1	5	f _{UCLK} /5
0	0	0	0	0	1	1	0	6	f _{UCLK} /6
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	f _{UCLK} /252
1	1	1	1	1	1	0	1	253	f _{UCLK} /253
1	1	1	1	1	1	1	0	254	f _{UCLK} /254
1	1	1	1	1	1	1	1	255	f _{UCLK} /255

备注 f_{UCLK}: 由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择的
时钟频率

(4) 波特率

波特率误差由下列公式计算。

$$\text{波特率} = \frac{f_{\text{CLK}}}{2 \times k} [\text{bps}]$$

使用内部时钟时，按以下公式（用 ASCKA0 引脚作为 UARTA0 的时钟时，按上式计算）

$$\text{波特率} = \frac{f_{\text{xx}}}{2^{m+1} \times k} [\text{bps}]$$

备注 f_{CLK} = 由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择的基础时钟频率
 f_{xx} : 主时钟频率
 m = 由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位设置的值 ($m = 0$ 至 10)
 k = 由 UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位设置的值 ($k = 4$ 至 255)

波特率误差由下列公式计算：

$$\begin{aligned} \text{误差} (\%) &= \left(\frac{\text{实际波特率 (有错误的波特率)}}{\text{目标波特率 (正确的波特率)}} - 1 \right) \times 100 [\%] \\ &= \left(\frac{f_{\text{CLK}}}{2 \times k \times \text{目标波特率}} - 1 \right) \times 100 [\%] \end{aligned}$$

使用内部时钟时，按以下公式（用 ASCKA0 引脚作为 UARTA0 的时钟时，按上式计算波特率误差）

$$\text{误差} (\%) = \left(\frac{f_{\text{xx}}}{2^{m+1} \times k \times \text{目标波特率}} - 1 \right) \times 100 [\%]$$

注意事项 1. 发送过程中，波特率误差必须在接收端的误差容限之内。
 2. 接收时的波特率误差满足一定范围，该范围参见“（5）接收中允许的波特率误差范围”。

要设置波特率，按以下算式计算，设置 UAnCTL1 和 UAnCTL2 寄存器（使用内部时钟时）。

- <1> 设置 k 等于 $f_{xx} / (2 \times \text{目标波特率})$ ，m 等于 0。
- <2> 如果 k 等于 256 或更大 ($k \geq 256$)，将 k 减小一半 ($k/2$)，m 加 1 ($m + 1$)。
- <3> 重复第<2>步，直到 k 小于 256 ($k < 256$)。
- <4> 对 k 的第一位小数四舍五入，得到最接近的整数。
如果四舍五入后 k 为 256，再次执行第<2>步将 k 设置为 128。
- <5> 将 m 的值设置到 UAnCTL1 寄存器，k 的值设置到 UAnCTL2 寄存器。

示例： 如果 $f_{xx} = 32 \text{ MHz}$ 而目标波特率 = 153,600 bps
 <1> $k = 32,000,000 / (2 \times 153,600) = 104.16\dots$ ，m = 0
 <2>，<3> $k = 104.16\dots < 256$ ，m = 0
 <4> 设置 UAnCTL2 寄存器的值： $k = 104 = 68\text{H}$ ，设置 UAnCTL1 寄存器的值： m = 0

$$\begin{aligned} \text{实际波特率} &= 32,000,000 / (2 \times 104) \\ &= 153,846 \text{ [bps]} \end{aligned}$$

$$\begin{aligned} \text{波特率误差} &= \{32,000,000 / (2 \times 104 \times 153,600) - 1\} \times 100 \\ &= 0.160 \text{ [\%]} \end{aligned}$$

以下为波特率设置的典型举例：

表 15-3. 波特率发生器设置数据

波特率 (bps)	$f_{xx} = 32 \text{ MHz}$			$f_{xx} = 20 \text{ MHz}$			$f_{xx} = 10 \text{ MHz}$		
	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)
300	08H	D0H	0.16	08H	82H	0.16	07H	82H	0.16
600	07H	D0H	0.16	07H	82H	0.16	06H	82H	0.16
1,200	06H	D0H	0.16	06H	82H	0.16	05H	82H	0.16
2,400	05H	D0H	0.16	05H	82H	0.16	04H	82H	0.16
4,800	04H	D0H	0.16	04H	82H	0.16	03H	82H	0.16
9,600	03H	D0H	0.16	03H	82H	0.16	02H	82H	0.16
19,200	02H	D0H	0.16	02H	82H	0.16	01H	82H	0.16
31,250	02H	80H	0.00	01H	A0H	0.00	00H	A0H	0.00
38,400	01H	D0H	0.16	01H	82H	0.16	00H	82H	0.16
76,800	00H	D0H	0.16	00H	82H	0.16	00H	41H	0.16
153,600	00H	68H	0.16	00H	41H	0.16	00H	21H	-1.36
312,500	00H	33H	0.39	00H	20H	0.00	00H	10H	0.00
625,000	00H	1AH	-1.54	00H	10H	0.00	00H	08H	0.00

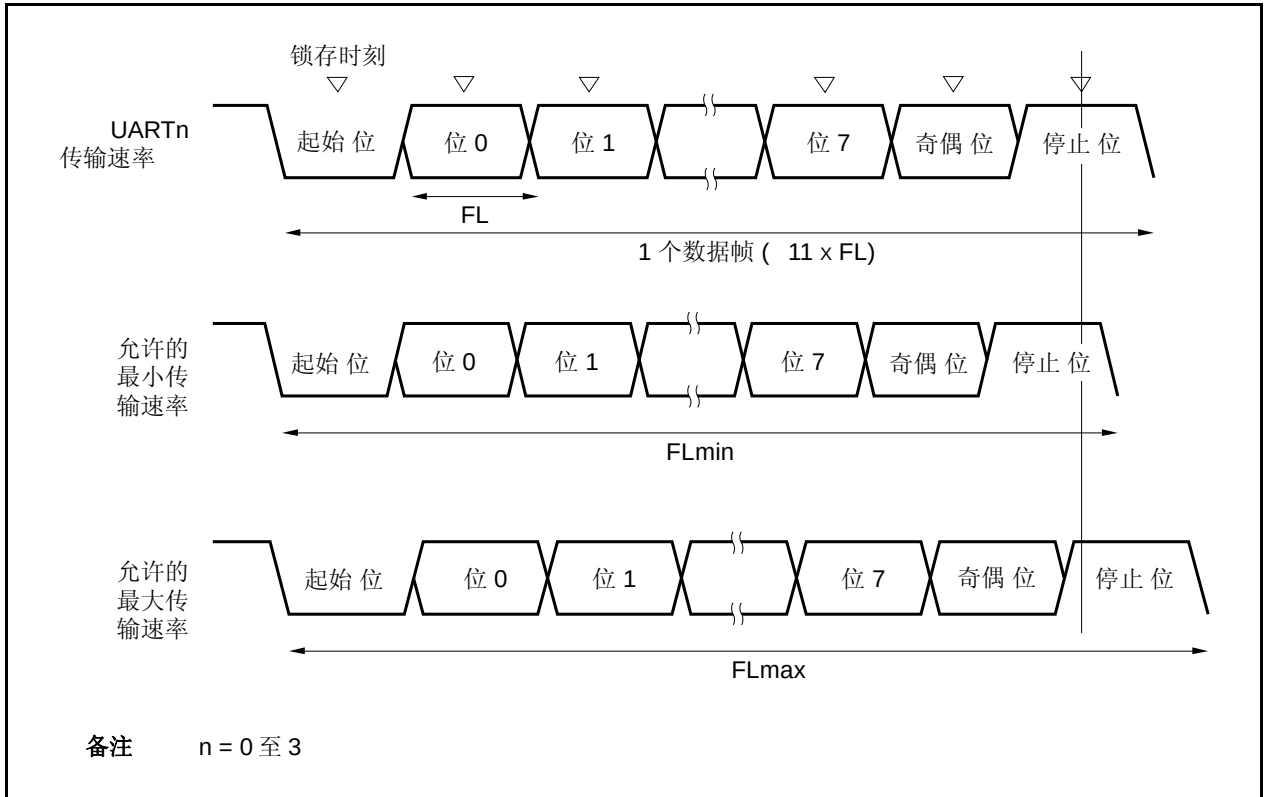
备注 f_{xx} : 主时钟频率
 ERR: 波特率误差 (%)

(5) 在接收期间允许的波特率范围

接收过程中，目的端允许的波特率误差范围如下所示：

注意事项 接收过程中使用以下公式，波特率误差必须设置在允许范围内。

图 15-17. 接收期间允许的波特率范围



如图 15-17 所示，接收数据的锁存时序取决于检测到起始位后 UAnCTL2 寄存器对计数器的设置值。如果在此锁存时序可以及时接收直到最末位数据（停止位），发送数据就可以被正常接收。当应用于 11 位接收时，以下为理论结果。

$$FL = (\text{波特率})^{-1}$$

波特率：UARTAn 波特率 (n = 0 至 3)

k: UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位的设定值 (n = 0 至 3)。

FL: 1 位数据长度

锁存时间余量: 2 个时钟

$$\text{最小允许传输速率: } FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

所以，可以从目的端接收的最大波特率如下所示：

$$BR_{max} = (FL_{min}/11)^{-1} \frac{22k}{21k + 2} \quad \text{Brate}$$

同理，可以得到下列公式来计算最大允许传输速率：

$$\frac{10}{11} \times FL_{max} = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FL_{max} = \frac{21k-2}{20k} FL \times 11$$

所以，可以从目的端接收的最小波特率如下所示：

$$BR_{min} = (FL_{max}/11)^{-1} \frac{20k}{21k - 2} \quad \text{Brate}$$

由上述公式计算 UARTAn 以及目的端的允许波特率误差，得到的最大和最小波特率值必须遵从以下表格。

表 15-4. 允许的最大/最小波特率误差

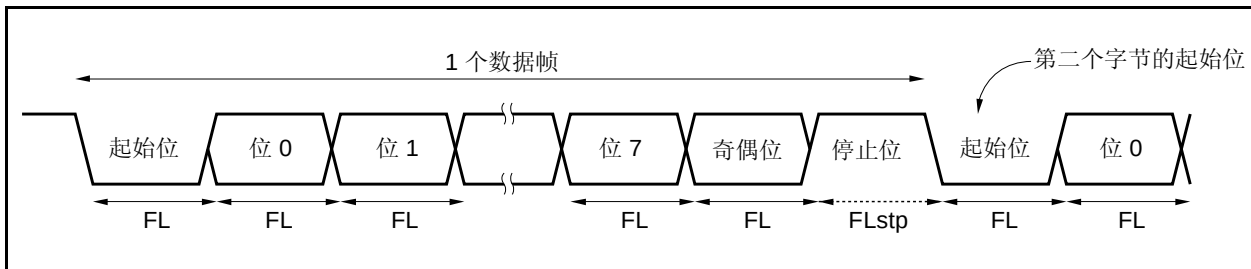
分频比 (k)	允许的最大波特率误差	允许的最小波特率误差
4	+2.32%	- 2.43%
8	+3.52%	- 3.61%
20	+4.26%	- 4.30%
50	+4.56%	- 4.58%
100	+4.66%	- 4.67%
255	+4.72%	- 4.72%

- 备注
1. 接收精度依赖于帧内的位数量、输入时钟频率和分频比 (k)。输入时钟频率越高且分频比 (k) 越大，则精度越高。
 2. k: UAnCTL2.UAnBRS7 位至 UAnCTL2.UAnBRS0 位的设定值 (n = 0 至 3)

(6) 连续发送期间的波特率

连续发送期间，从停止位到下一个起始位的传输间隔通常大于 2 个基础时钟。而时序的初始化由接收端通过检测起始位执行，因此，这对传输结果没有影响。

图 15-18. 连续发送期间的传送速率



假定 1 位数据长度为 FL ；停止位长度为 FL_{stp} ；基础时钟频率为 f_{uCLK} ，可以得到如下等式：

$$FL_{stp} = FL + 2/f_{uCLK}$$

所以，连续发送期间的传输速率如下。

$$\text{传输速率} = 11 \times FL + (2/f_{uCLK})$$

15.8 注意事项

- (1) 当提供给 UARTAn 的时钟停止（比如进入 IDLE1、IDLE2 或 STOP 模式）时，则操作停止，每一个寄存器都保留时钟供应停止前的瞬间值。TXDAn 引脚输出也保持时钟供给停止前的瞬时值。然而，恢复时钟供给后，不能保证正常操作。因此，在时钟恢复之后，应该将 UAnCTL0.UAnPWR 位、UAnCTL0.UAnRXEn 位和 UAnCTL0.UAnTXEn 位设置为 000，对电路进行初始化。
- (2) RXDA1 和 KR7 引脚不能同时使用。使用 RXDA1 引脚时，不要使用 KR7 引脚。若使用 KR7 引脚，就不要使用 RXDA1 引脚（建议将 PFC91 位置为 1，并清除 PFCE91 位为 0）。
- (3) UARTAn 中，不会产生由通信错误引起的中断。使用 DMA 传输功能进行发送数据和接收数据的传输时，即使传输期间发生错误（奇偶校验错误，溢出错误，帧错误），也不执行错误处理。或者在 DMA 传输完成后读取 UAnSTR 寄存器检查是否有错误，或者在通信期间读取 UAnSTR 寄存器来检查错误。
- (4) 按以下步骤启动 UARTAn。
 - <1> 将 UAnCTL0.UAnPWR 位置为 1。
 - <2> 设置该端口。
 - <3> 将 UAnCTL0.UAnTXE 位和 UAnCTL0.UAnRXE 位置为 1。
- (5) 按以下步骤停止 UARTAn。
 - <1> 将 UAnCTL0.UAnTXE 位和 UAnCTL0.UAnRXE 位清为 0。
 - <2> 设置端口，并设置 UAnCTL0.UAnPWR 位为 0（如果端口设置不改变，则不存在该问题）。
- (6) 发送模式下（UAnCTL0.UAnPWR 位 = 1 且 UAnCTL0.UAnTXE 位 = 1），因为写入 UAnTX 寄存器会开始发送，所以不要用软件向该寄存器写入相同的值。若连续发送相同值，则重写相同的值。
- (7) 在连续发送中，从停止位到下一个起始位的通信速率要比平常延长 2 个以上基础时钟。但是，接收端通过检测起始位进行时序的初始化，因此接收结果不受影响。

第十六章 3-线可变长度串行 I/O (CSIB)

16.1 CSIB 和其它串行接口的模式转换

16.1.1 CSIB4 和 UARTA0 的模式转换

在 V850ES/JJ3 中, CSIB4 和 UARTA0 是相同引脚的复用功能, 因而不能同时使用。使用之前, 用 PMC3 和 PFC3 寄存器预先设置 CSIB4。

注意事项 如果在发送或接收期间切换这些功能, 则 CSIB4 和 UARTA0 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 16-1. CSIB4 和 UARTA0 模式转换设置

复位后：0000H

R/W

地址： FFFFF446H, FFFFF447H

PMC3

15	14	13	12	11	10	9	8
0	0	0	0	0	0	PMC39	PMC38
7	6	5	4	3	2	1	0
0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

复位后：0000H

R/W

地址： FFFFF466H, FFFFF467H

PFC3

15	14	13	12	11	10	9	8
0	0	0	0	0	0	PFC39	PFC38
7	6	5	4	3	2	1	0
0	0	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

复位后：00H

R/W

地址： FFFFF706H

PFCE3L

7	6	5	4	3	2	1	0
0	0	0	0	0	PFCE32	0	0

PMC32	PFCE32	PFC32	工作模式
0	×	×	Port I/O 模式
1	0	0	ASCKA0 模式
1	0	1	SCKB4 模式

PMC3n	PFC3n	工作模式
0	×	Port I/O 模式
1	0	UARTA0 模式
1	1	CSIB4 模式

备注

1. n = 0, 1

2. × = 无需理会

16.1.2 CSIB0 和 I²C01 的模式转换

在V850ES/JJ3 中，CSIB0 和 I²C01 是相同引脚的复用功能，因而不能同时使用。使用之前，用PMC4 和 PFC4 寄存器预先设置CSIB0。

注意事项 如果在发送或接收期间切换这些功能，则CSIB0 和 I²C01 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 16-2. 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	x	端口 I/O 模式
1	0	CSIB0 模式
1	1	I ² C01 模式

16.2 特性

- 传输速率：最大 8 Mbps（f_{xx} = 32 MHz，使用内部时钟）
- 主机、从机模式可选
- 8-位至 16-位传输，3-线串行接口
- 中断请求信号（INTCBnT，INTCBnR）× 2
- 串行时钟和数据相位可翻转
- 传输数据长度可以在 8 位至 16 位之间选择任意位长度
- 可切换 MSB/LSB 先行传输数据
- 3-线传输 SOBn: 串行数据输出
 SIBn: 串行数据输入
 SCKBn: 串行时钟 I/O

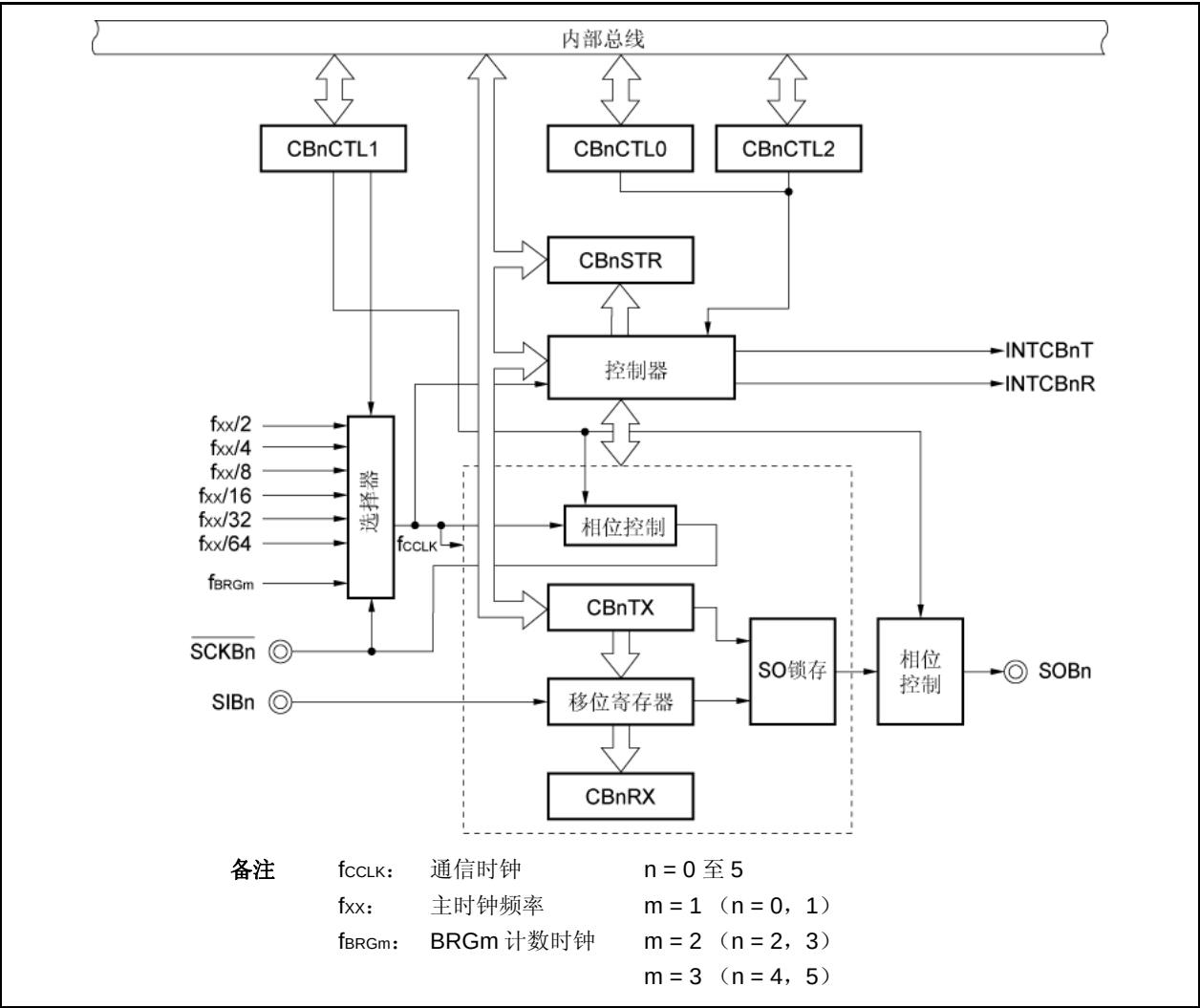
可指定为发送模式、接收模式以及发送/接收模式。

备注 n = 0 至 5

16.3 配置

以下显示 CSIBn 的功能框图。

图 16-3. CSIBn 框图



CSIBn 包括下列硬件：

表 16-1. CSIBn 的配置

项目	配置
寄存器	CSIBn 接收数据寄存器 (CBnRX) CSIBn 发送数据寄存器 (CBnTX)
控制寄存器	CSIBn 控制寄存器 0 (CBnCTL0) CSIBn 控制寄存器 1 (CBnCTL1) CSIBn 控制寄存器 2 (CBnCTL2) CSIBn 状态寄存器 (CBnSTR)

(1) CSIBn 接收数据寄存器 (CBnRX)

CBnRX 是 16-位缓存寄存器，用于存放接收数据。

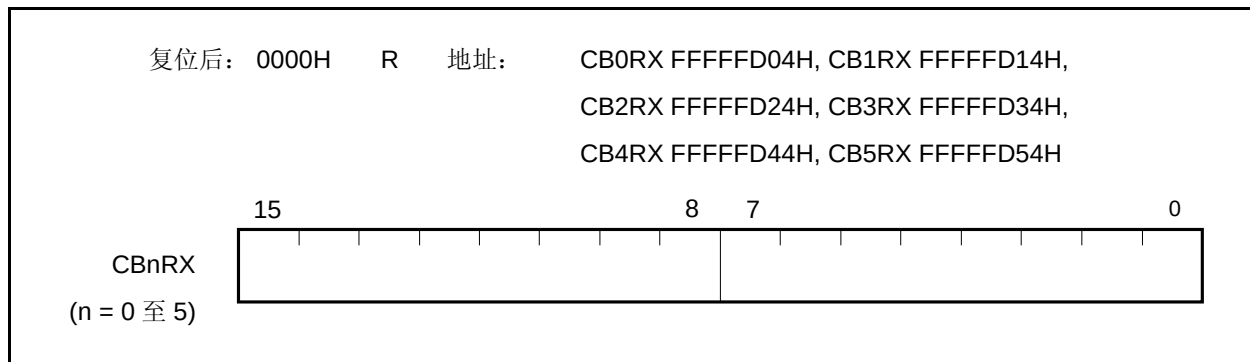
该寄存器是只读寄存器，以 16 位宽度读取。

在接收允许状态下，通过读取 CBnRX 寄存器启动接收操作。

如果传输数据长度为 8 位，该寄存器的低 8 位作为寄存器 CBnRXL，且只能以 8 位宽度进行读取。

系统复位后，该寄存器被设为 0000H。

除复位输入外，清除 CBnCTL0 寄存器的 CBnPWR 位（为 0），也可以对 CBnRX 寄存器进行初始化。



(2) CSIBn 发送数据寄存器 (CBnTX)

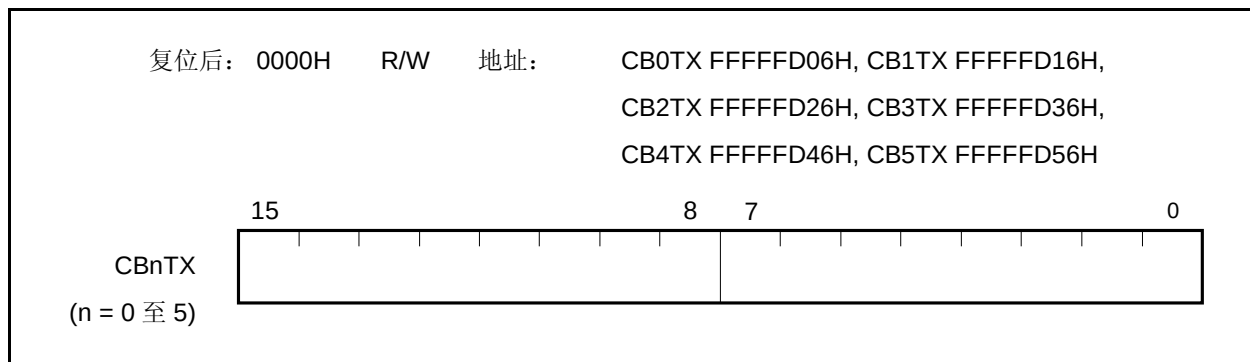
CBnTX 寄存器是 16 位缓冲寄存器，用于写入 CSIBn 的发送数据。

该寄存器可以 16 位宽度进行读取或写入。

在发送允许状态下，通过向 CBnTX 寄存器写入数据启动发送操作。

如果传输数据长度为 8 位，该寄存器的低 8 位作为寄存器 CBnTXL，且只能以 8 位宽度进行读取。

系统复位后，该寄存器被设为 0000H。



备注 通信开始条件如下所示。

发送模式 (CBnTXE 位 = 1, CBnRXE 位 = 0) :

写入 CBnTX 寄存器

发送/接收模式 (CBnTXE 位 = 1, CBnRXE 位 = 1) :

写入 CBnTX 寄存器

接收模式 (CBnTXE 位 = 0, CBnRXE 位 = 1):

从 CBnRX 寄存器读取数据

16.4 寄存器

以下寄存器用来控制 CSIBn。

- CSIBn 控制寄存器 0 (CBnCTL0)
- CSIBn 控制寄存器 1 (CBnCTL1)
- CSIBn 控制寄存器 2 (CBnCTL2)
- CSIBn 状态寄存器 (CBnSTR)

(1) CSIBn 控制寄存器 0 (CBnCTL0)

CBnCTL0 寄存器用于控制 CSIBn 串行传输操作。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 01H。

(1/3)

复位后：01H R/W 地址： CB0CTL0 FFFFFFFD00H, CB1CTL0 FFFFFFFD10H,
CB2CTL0 FFFFFFFD20H, CB3CTL0 FFFFFFFD30H,
CB4CTL0 FFFFFFFD40H, CB5CTL0 FFFFFFFD50H

	<7>	<6>	<5>	<4>	3	2	1	<0>
CBnCTL0 (n = 0 至 5)	CBnPWR	CBnTXE ^注	CBnRXE ^注	CBnDIR ^注	0	0	CBnTMS ^注	CBnSCE

CBnPWR	CSIBn 操作禁止/使能的说明
0	禁止 CSIBn 操作并且清除 CBnSTR 寄存器
1	使能 CSIBn 操作
• CBnPWR 位控制 CSIBn 的操作，并复位内部电路。	

CBnTXE ^注	发送操作禁止/使能的说明
0	禁止发送操作
1	使能发送操作
• CBnTXE 位为 0 时，SOBn 输出低电平。	

CBnRXE ^注	接收操作禁止/使能的说明
0	禁止接收操作
1	使能接收操作
• CBnRXE 位清为 0 时，不会输出接收完成中断，即使发送来的数据是预定用于禁止接收操作，也不更新接收数据 (CBnRX 寄存器)。	

注 只有当 CBnPWR 位 = 0 时，这些位才可重写。可以在重写这些位的同时设置 CBnPWR 位 = 1。

注意事项 若要强制挂起发送/接收，将 CBnPWR 位清为 0，而不是清除 CBnRXE 和 CBnTXE 位。
此时，时钟输出停止。

CBnDIR ^注	传输方向模式 (MSB/LSB)的说明
0	MSB先行传输
1	LSB先行传输

CBnTMS ^注	传输模式说明
0	单次传输模式
1	连续发送模式

<p>[单次传输模式] 通信完成后，发生接收完成中断 (INTCBnR)。 即使允许发送 (CBnTXE 位 = 1)，也不产生发送使能中断(INTCBnT)。 如果通信期间(CBnSTR.CBnTSF 位 = 1)写入下一个发送数据，此数据被忽略， 且不开始下一次通信。同样如果设置为只接收通信(CBnTXE位=0，CBnRXE位=1)， 即使通信期间读取接收数据 (CBnSTR.CBnTSF位=1)，也不开始进行下一次通信。</p> <p>[连续发送模式] 在通信期间 (CBnSTR.CBnTSF位=1) 写入下一个发送数据，使能连续发送。 产生发送使能中断(INTCBnT)后，允许写入下一个发送数据。如果连续传输模式被 设置为只接收通信(CBnTXE 位 = 0,CBnRXE 位 = 1)，产生接收完成中断 (INTCBnR) 后，下一次接收连续开始，而不考虑CBnRX寄存器的读取操作。 因此，从CBnRX寄存器立即读取接收数据。如果此读取操作被延迟，则会发生 溢出错误 (CBnOVE位=1)。</p>	
--	--

注 只有当 CBnPWR 位= 0 时，这些位才可重写。
但是，重写这些位的同时可以设置 CBnPWR 位= 1 。

CBnSCE	开始传输禁止/使能的说明
0	通信启动触发无效
1	通信启动触发有效

• 主机模式下
 该位允许或禁止通信启动触发。
 (a) 单次发送或发送/接收模式或连续发送或连续发送/接收模式下
 CBnSCE 位的设置对通信操作没有影响。
 (b) 单次接收模式下
 因为要通过读取接收数据 (CBnRX 寄存器) 以开始接收, 所以在
 读取最后一个接收数据之前将 CBnSCE 位清除为 0, 来禁止接收开始^{注1}。
 (c) 连续接收模式下
 最后一个数据接收完成前一个通信时钟将 CBnSCE 位清除为 0, 使得
 最后一个数据接收^{注2}完成后禁止启动接收。

• 从机模式下
 该位允许或禁止通信开始触发。
 设置 CBnSCE 位为 1。

[CBnSCE 位的用法]

• 单次接收模式下
 <1> 最后一个数据接收完成时通过 INTCBnR 中断服务, 在读取 CBnRX
 寄存器前清除 CBnSCE 位为 0。
 <2> 确认 CBnSTR.CBnTSF 位 = 0 后, 清除 CBnRXE 位为 0 以禁止接收。
 若要继续接收, 通过伪读取 CBnRX 寄存器将 CBnSCE 位设置为 1
 来启动下一次接收。

• 连续接收模式下
 <1> 最后一个数据接收期间通过 INTCBnR 中断服务来清除 CBnSCE 位为 0。
 <2> 读取 CBnRX 寄存器。
 <3> 响应 CBnTIR 中断后读取 CBnRX 寄存器来读取最后一个接收数据。
 <4> 确认 CBnSTR.CBnTSF 位 = 0 后, 清除 CBnRXE 位为 0 以禁止接收。
 若要继续接收, 通过伪读取 CBnRX 将 CBnSCE 位设置为 1 以等待
 下一次接收。

- 注**
1. 如果其值为 1 时, 读取 CBnSCE 位, 则开始下一次通信操作。
 2. 最末数据接收结束之前的一个通信时钟, CBnSCE 位不清 0, 则下一次通信操作自动开始。

注意事项 请确保将第 3 位和第 2 位清除为“0”。

(2) CSIBn 控制寄存器 1 (CBnCTL1)

CBnCTL 是 8 位寄存器，用于控制 CSIBn 串行传输操作。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

注意事项 仅当 CBnCTL0.CBnPWR 位= 0 时，CBnCTL1 寄存器才可以重写。

复位后: 00H R/W 地址: CB0CTL1 FFFFFFFD01H, CB1CTL1 FFFFFFFD11H,
CB2CTL1 FFFFFFFD21H, CB3CTL1 FFFFFFFD31H,
CB4CTL1 FFFFFFFD41H, CB5CTL1 FFFFFFFD51H

	7	6	5	4	3	2	1	0
CBnCTL1 (n = 0 至 5)	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

	CBnCKP	CBnDAP	关于 SCKBn 的数据发送/接收 时序说明
通信 类型 1	0	0	
通信 类型 2	0	1	
通信 类型 3	1	0	
通信 类型 4	1	1	

CBnCKS2	CBnCKS1	CBnCKS0	通信时钟 (f _{CCLK}) ^注	模式
0	0	0	f _{xx} /2	主模式
0	0	1	f _{xx} /4	主模式
0	1	0	f _{xx} /8	主模式
0	1	1	f _{xx} /16	主模式
1	0	0	f _{xx} /32	主模式
1	0	1	f _{xx} /64	主模式
1	1	0	f _{BRGm}	主模式
1	1	1	外部时钟 (SCKBn)	从模式

注 设置通信时钟 (f_{CCLK}) 为 8 MHz 或更低。

备注 当 n = 0 或 1, m = 1

当 n = 2 或 3, m = 2

当 n = 4 或 5, m = 3

关于 f_{BRGm} 详情, 参见 16.8 波特率发生器。

(3) CSIBn 控制寄存器 2 (CBnCTL2)

CBnCTL2 是 8 位寄存器，用于控制 CSIBn 串行传输的位数量。

该寄存器可进行字节读取或写入。

系统复位后，该寄存器被设为 00H。

注意事项 仅当 CBnCTL0.CBnPWR 位= 0 或 CBnTXE 和 CBnRXE 位均为 0 时，CBnCTL2 寄存器才可以重写。

复位后：00H R/W 地址： CB0CTL2 FFFFFFFD02H, CB1CTL2 FFFFFFFD12H,
CB2CTL2 FFFFFFFD22H, CB3CTL2 FFFFFFFD32H,
CB4CTL2 FFFFFFFD42H, CB5CTL2 FFFFFFFD52H

	7	6	5	4	3	2	1	0
CBnCTL2 (n = 0 至 5)	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

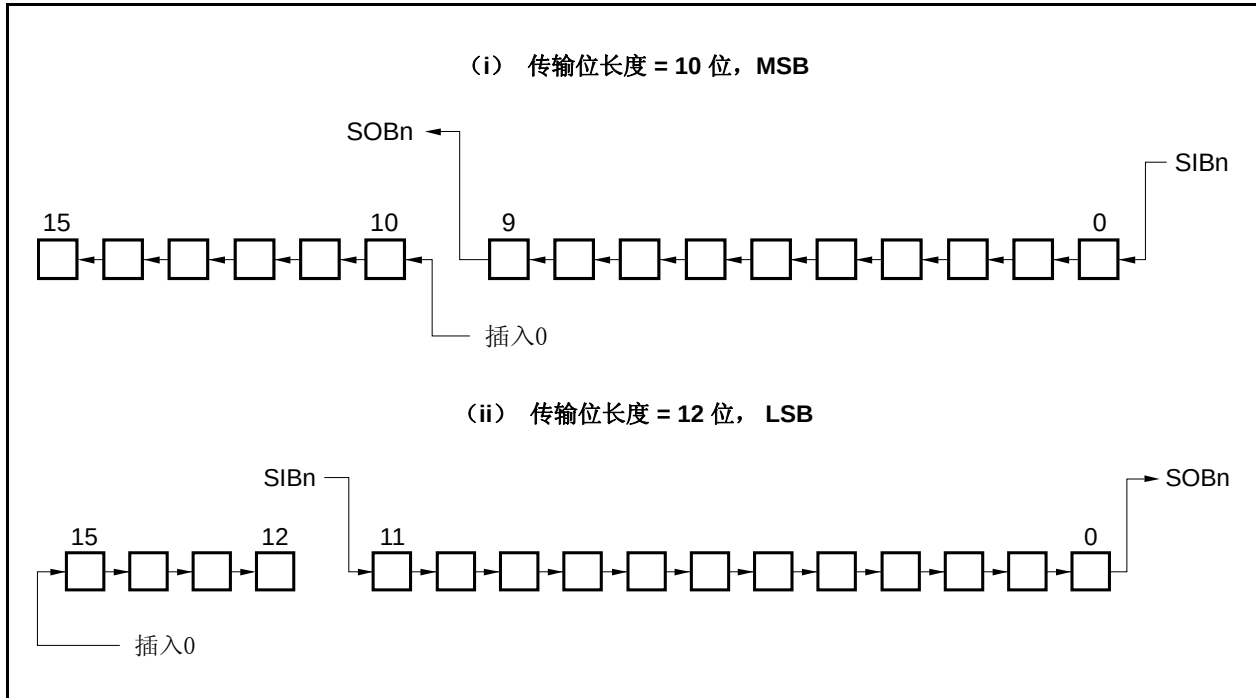
CBnCL3	CBnCL2	CBnCL1	CBnCL0	串行寄存器位的长度
0	0	0	0	8 位
0	0	0	1	9 位
0	0	1	0	10 位
0	0	1	1	11 位
0	1	0	0	12 位
0	1	0	1	13 位
0	1	1	0	14 位
0	1	1	1	15 位
1	x	x	x	16 位

- 备注**
1. 如果传输位数不是 8 位或 16 位，从 CBnTX 和 CBnRX 寄存器的 LSB 起进行填充。
 2. x：无需理会

(a) 传输数据长度改变功能

可以使用 CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位，在 8 位和 16 位之间以位为单位设置 CSIBn 传输的数据长度。

当设置的传输比特长度不是 16 位时，从 LSB 开始向 CBnTX 或 CBnRX 寄存器设置数据，而忽略传输起始位是 MSB 还是 LSB。没有使用的高位可以设置为任意数据，但是，在随后的串行传输中接收的对应数据变为 0。



(4) CSIBn 状态寄存器 (CBnSTR)

CBnSTR 是 8 位寄存器，用于显示 CSIBn 的状态。

该寄存器可以按 8 位或 16 位宽度进行读取或写入，但是，CBnTSF 标志只读。

系统复位后，该寄存器被设为 00H。

除复位输入之外，可以通过清除 CBnCTL0.CBnPWR 位（为 0）对 CBnSTR 寄存器进行初始化。

复位后：00H R/W 地址： CB0STR FFFFFFFD03H, CB1STR FFFFFFFD13H,
CB2STR FFFFFFFD23H, CB3STR FFFFFFFD33H,
CB4STR FFFFFFFD43H, CB5STR FFFFFFFD53H

	<7>	6	5	4	3	2	1	<0>
CBnSTR (n = 0 至 5)	CBnTSF	0	0	0	0	0	0	CBnOVE

CBnTSF	通信状态标志
0	通信停止
1	通信进行中
<ul style="list-style-type: none"> 发送期间，CBnTX寄存器中的数据准备好时，设置该寄存器。接收期间，进行CBnRX 寄存器伪读取时，设置该寄存器。传输结束时，在最后时钟的有效沿该标志清除为 0。 	

CBnOVE	溢出错误标志
0	无溢出
1	溢出
<ul style="list-style-type: none"> 当接收操作完成时，下一个接收已经完成而CPU没有读取接收缓冲器的值，发生溢出错误。此情况下，CBnOVE标志表示溢出错误发生状态。 即使单次传输模式下 CBnOVE 位同样有效。因此，只用作发送时，需注意以下事项： <ul style="list-style-type: none"> 不要检查 CBnOVE 标志。 即使不需要接收数据，仍要读取该位。 写入0以清除 CBnOVE 标志。不能通过写入1进行置位。 	

16.5 中断请求信号

CSIBn 可以产生以下两种中断请求信号：

- 接收完成中断请求信号 (INTCBnR)
- 发送使能中断请求信号 (INTCBnT)

这两种中断请求信号中，默认接收完成中断请求信号有较高优先级，而发送允许中断请求信号优先级较低。

表 16-2. 中断与其默认优先级

中断	优先级
接收完成	高
发送允许	低

(1) 接收完成中断请求信号 (INTCBnR)

接收使能状态下，接收数据传送到 CBnRX 寄存器，产生接收完成中断请求信号。

如果发生溢出错误，同样也会产生中断请求信号。

响应接收完成中断请求信号并读取数据时，读取 CBnSTR 寄存器并检查接收结果是否错误。

单次传输模式中，即使只进行发送，传输完成后也会产生 INTCBnR 中断请求信号。

(2) 发送使能中断请求信号 (INTCBnT)

连续发送或连续发送/接收模式中，发送数据从 CBnTX 寄存器移出，并且一旦允许写入 CBnTX 寄存器，就会产生发送允许中断请求信号。

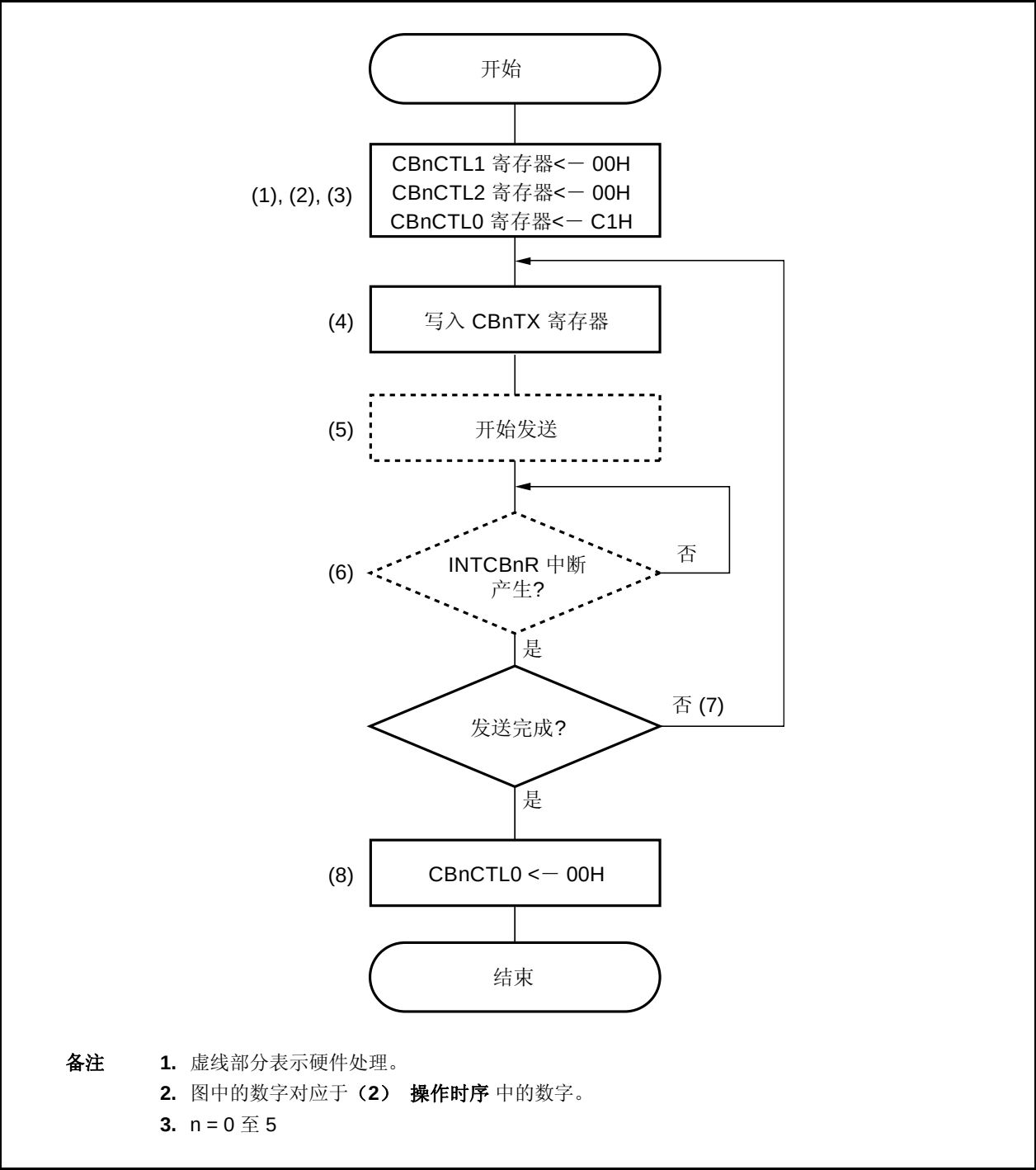
在单次发送和单次发送/接收模式中，不产生 INTCBnT 中断。

16.6 操作

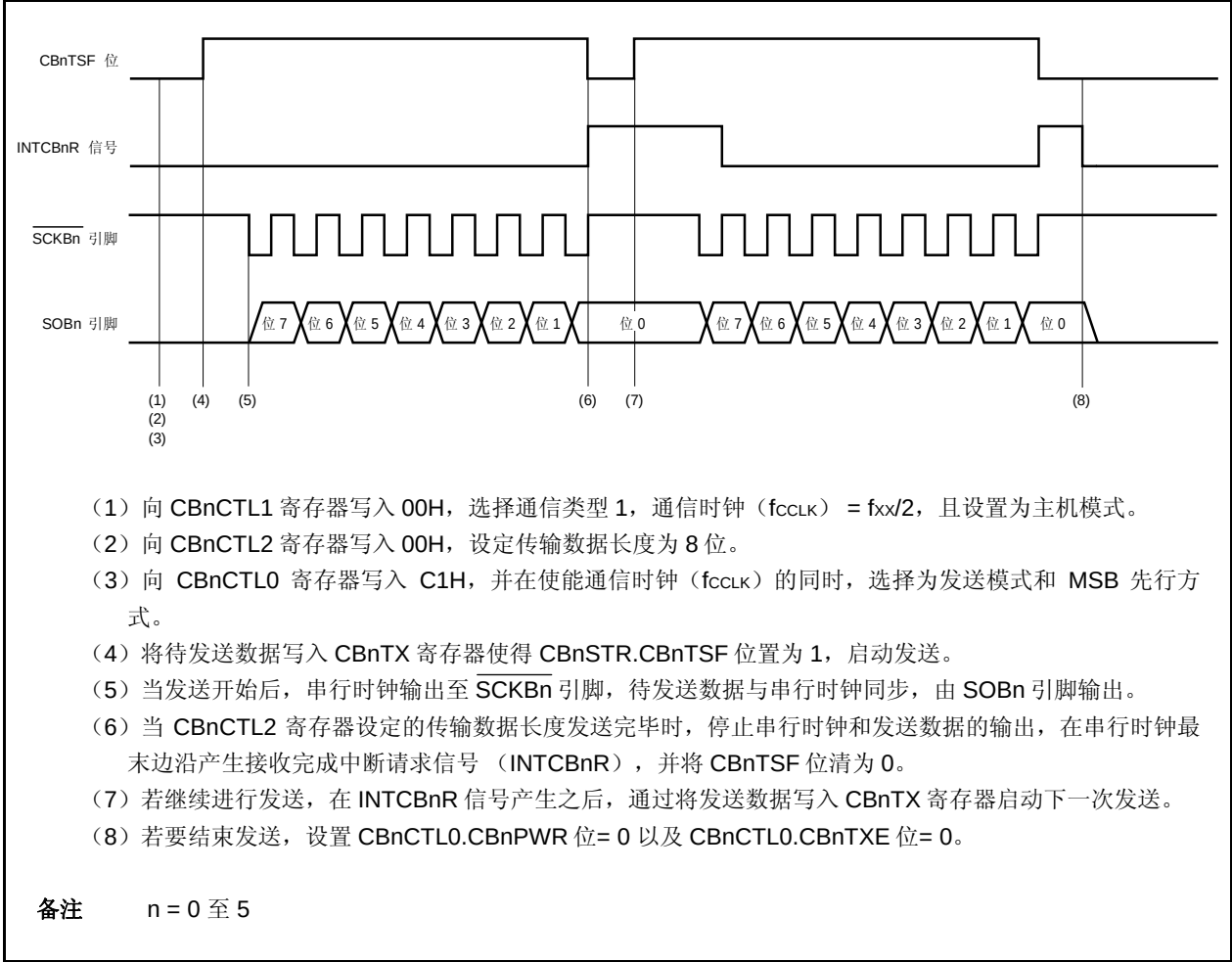
16.6.1 单次传输模式（主机模式，发送模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCCLK}）= f_{xx} /2（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000），传输数据长度 = 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

（1）操作流程



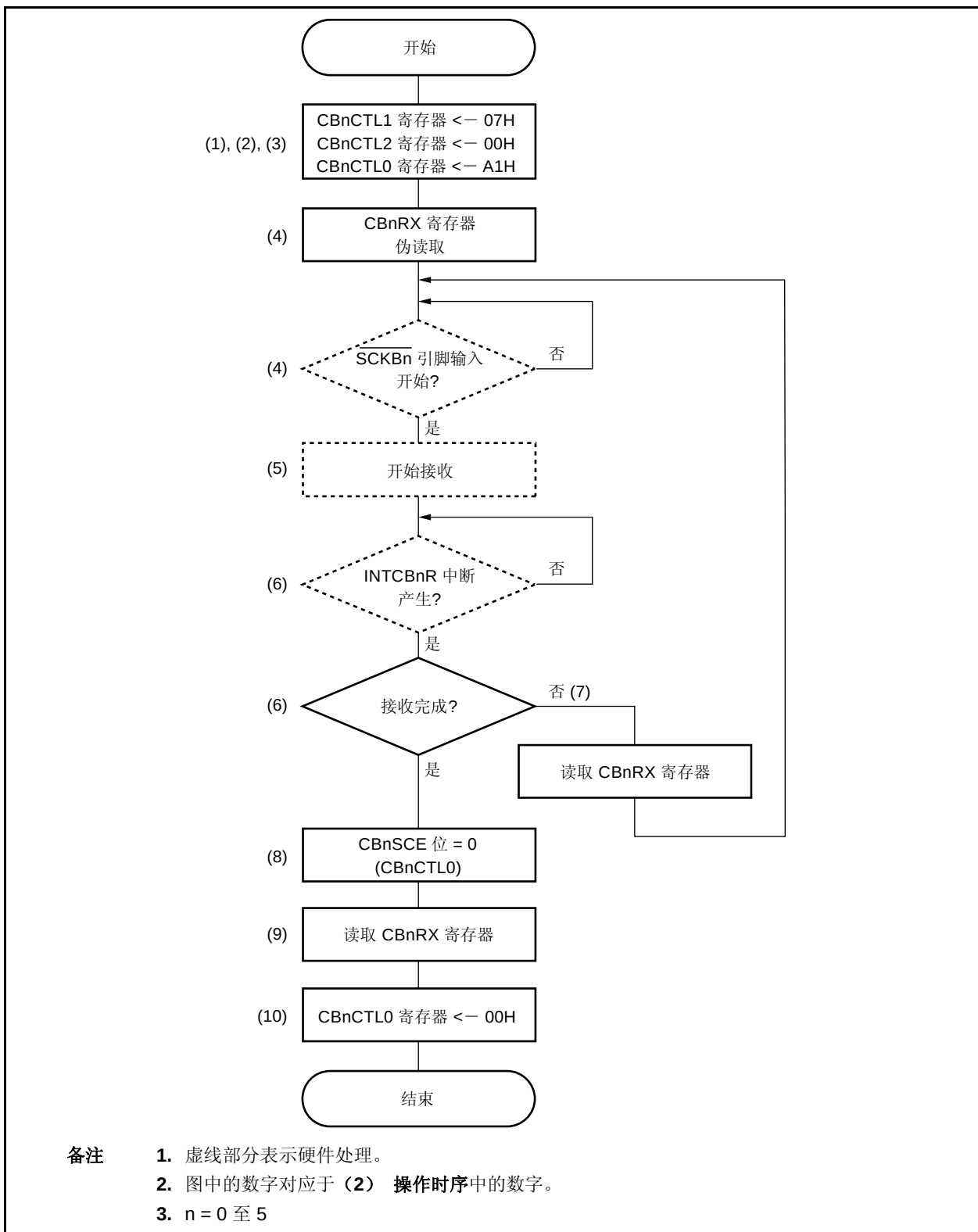
(2) 操作时序



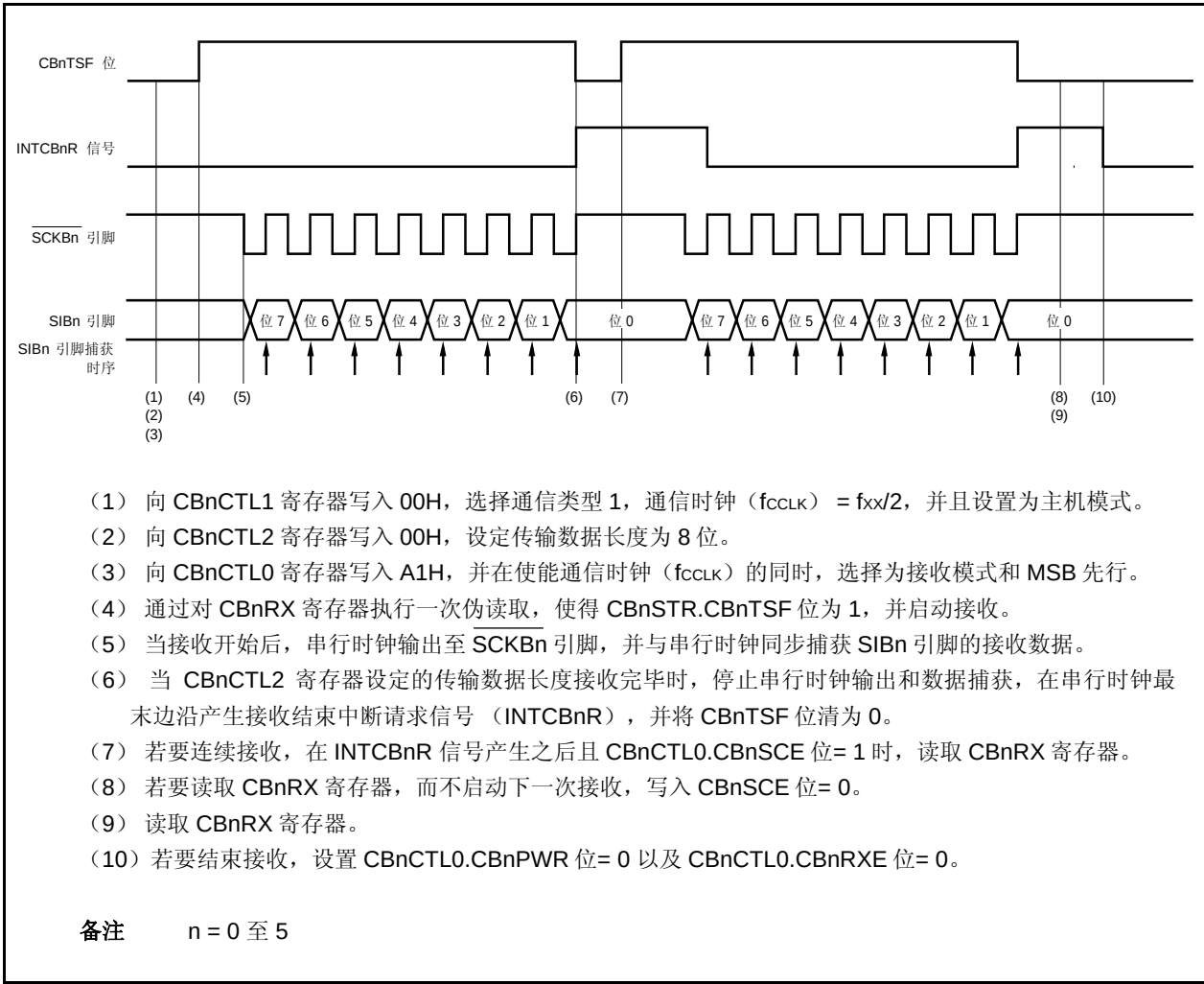
16.6.2 单次传输模式（主机模式，接收模式）

MSB 先行（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 至 CBnCTL1.CBnCKS0 位 = 000），传输数据长度 = 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

(1) 操作流程



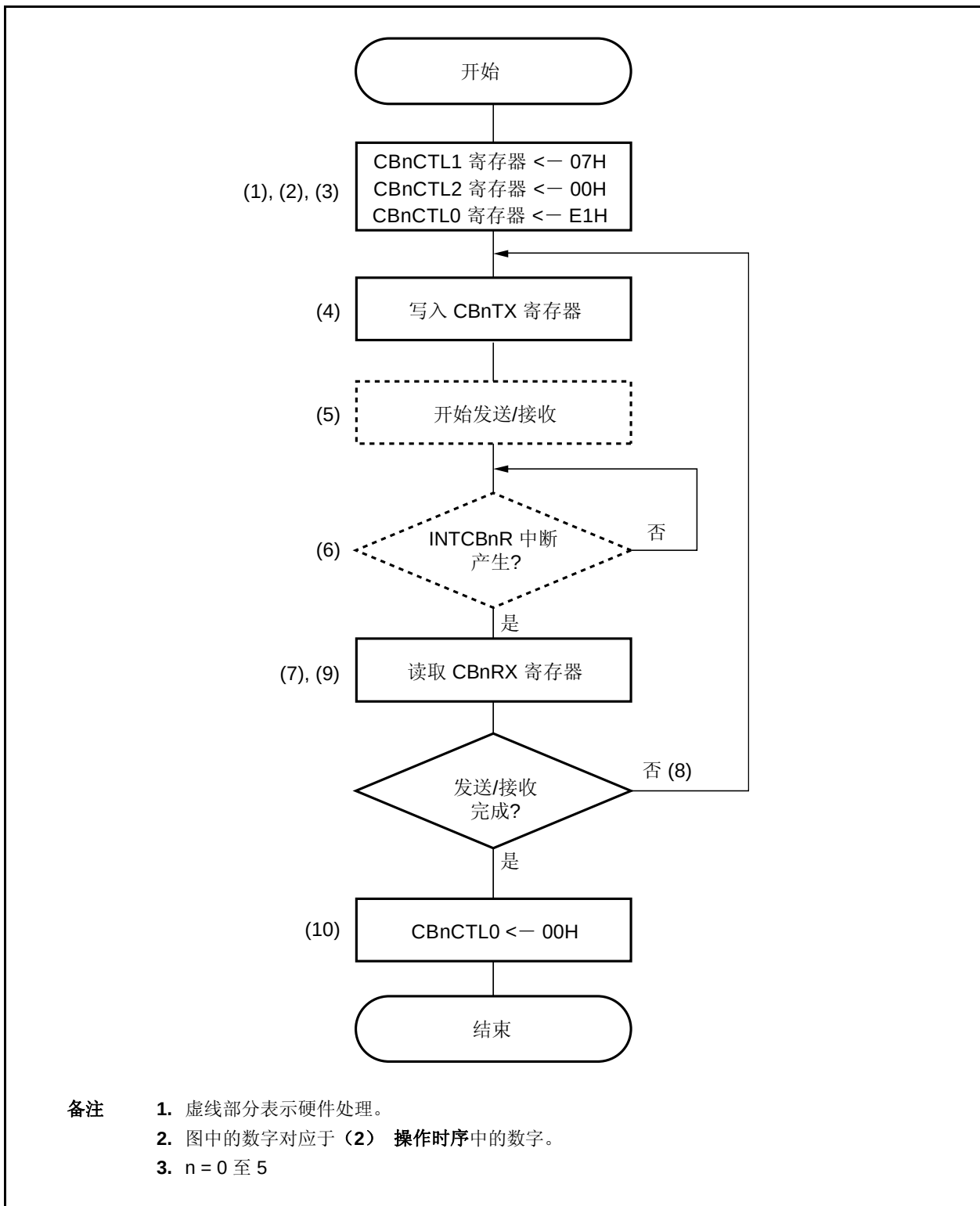
(2) 操作时序



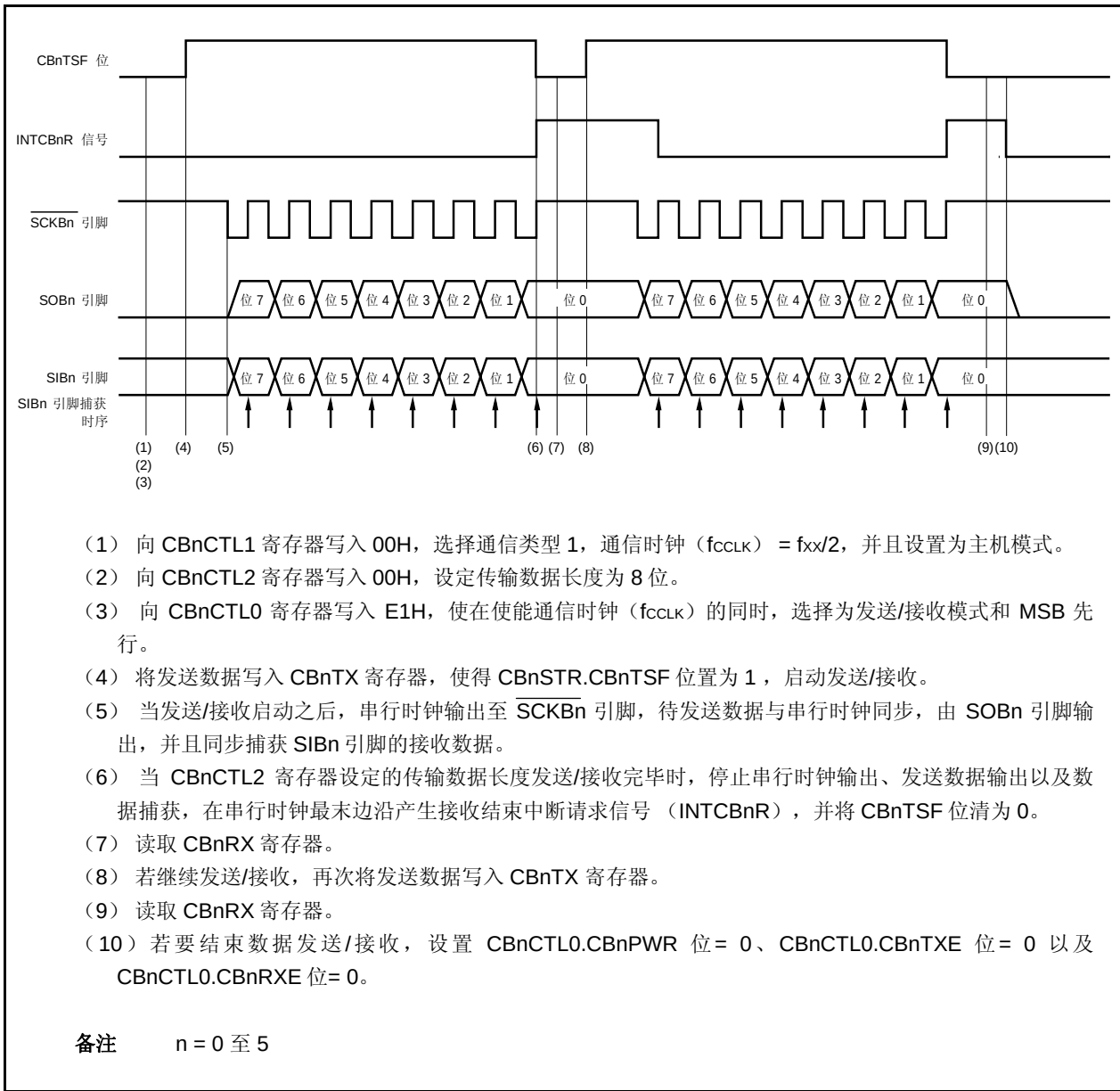
16.6.3 单次传输模式（主机模式，发送/接收模式）

MSB 先行（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 至 CBnCTL1.CBnCKS0 位 = 000），传输数据长度 = 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

(1) 操作流程



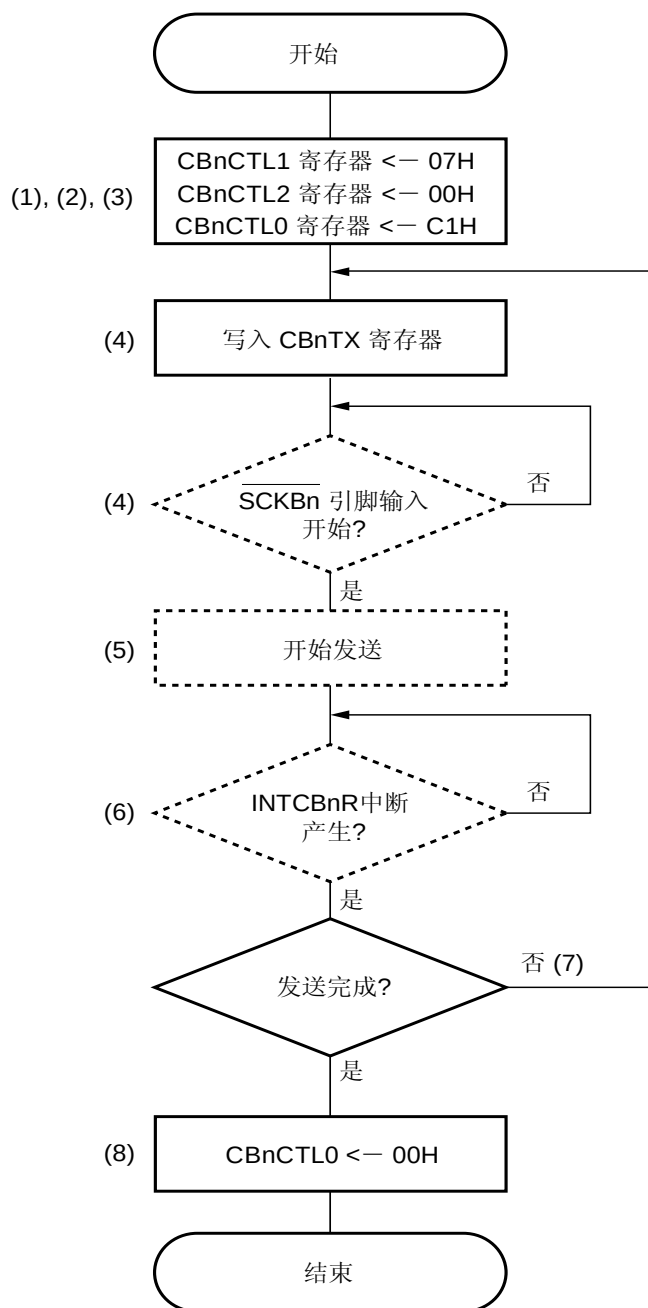
(2) 操作时序



16.6.4 单次传输模式（从机模式，发送模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= 外部时钟（ $\overline{\text{SCKBn}}$ ）（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

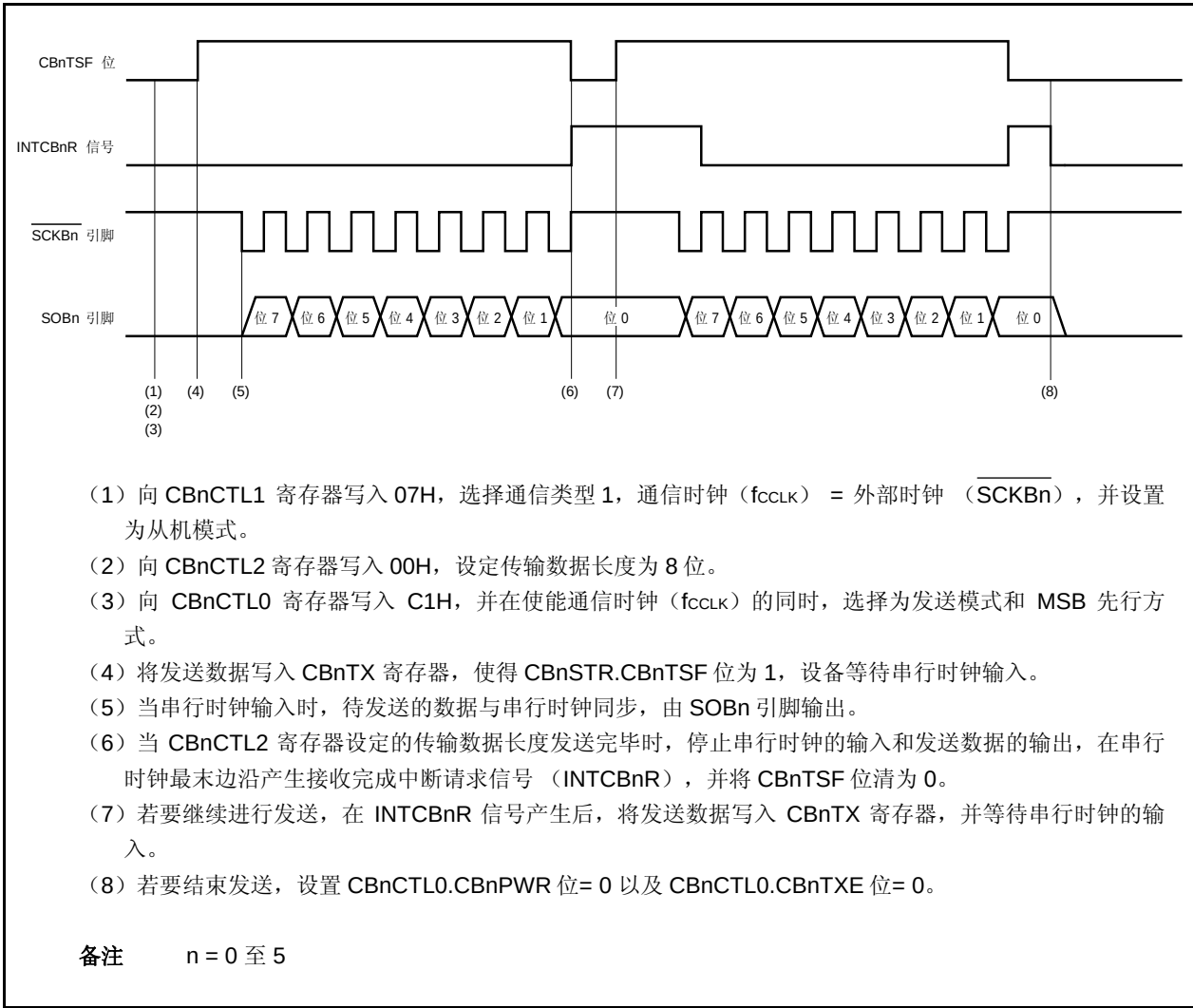
(1) 操作流程



备注

1. 虚线部分表示硬件处理。
2. 图中的数字对应于 (2) 操作时序中的数字。
3. n = 0 至 5

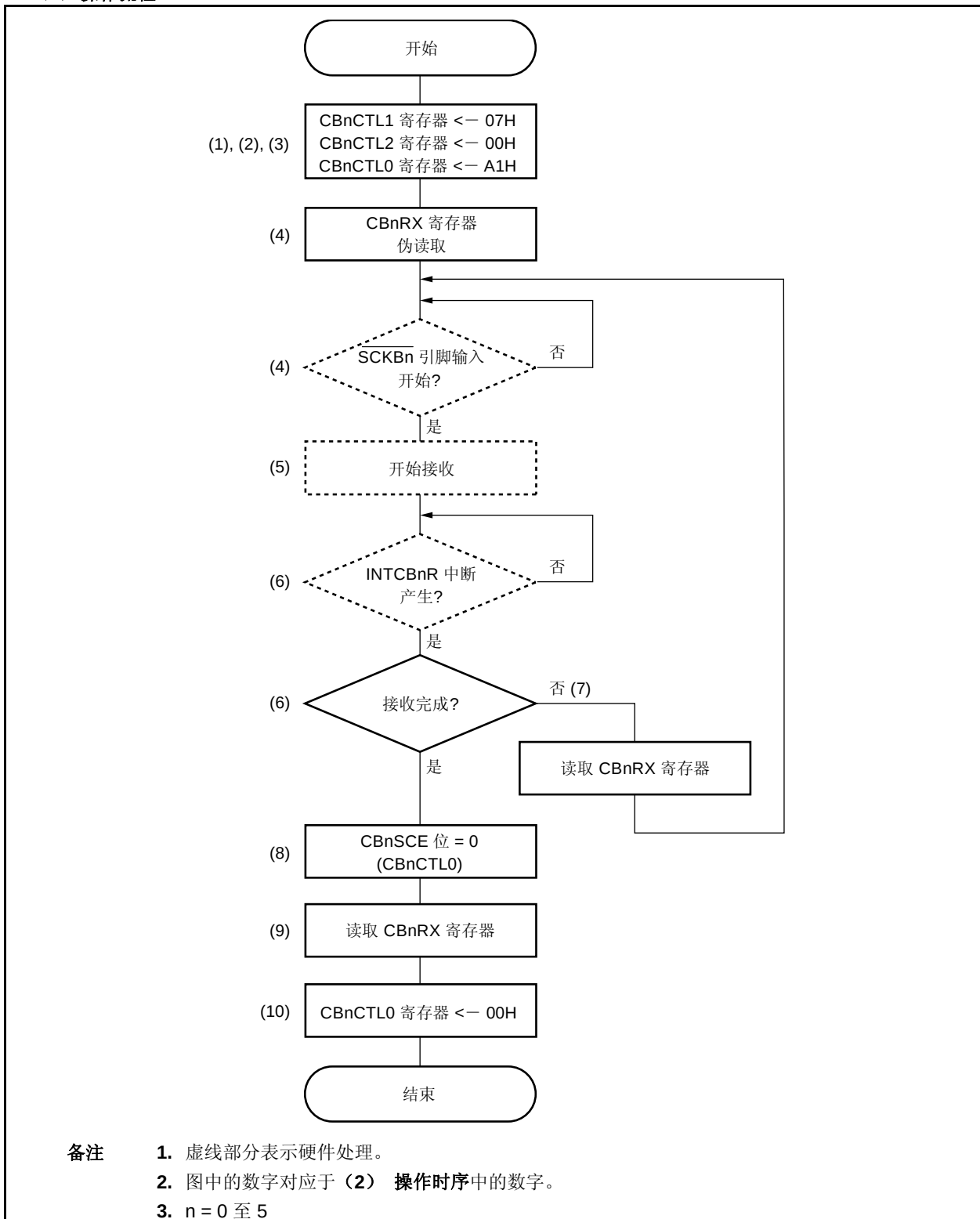
(2) 操作时序



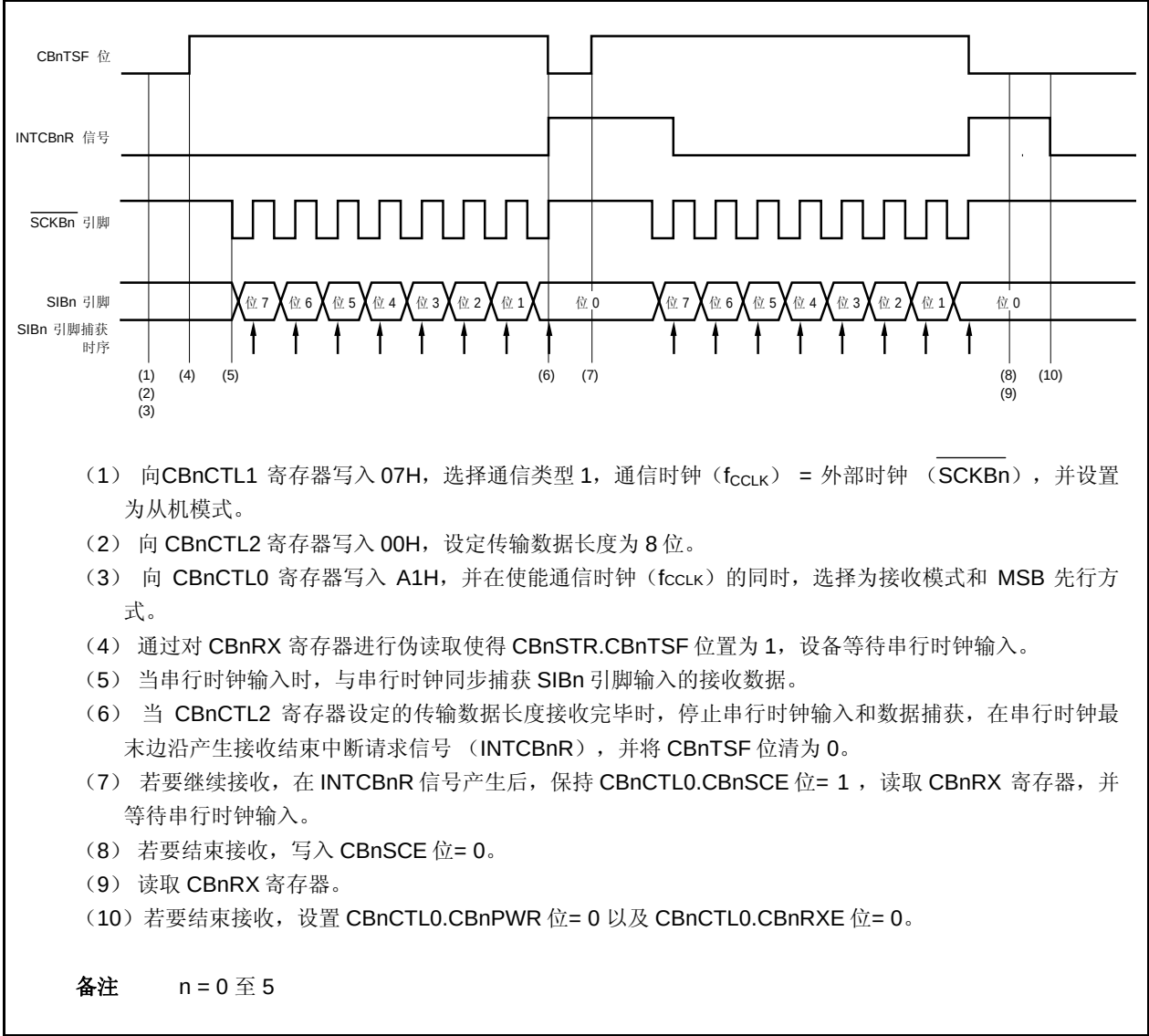
16.6.5 单次传输模式 (从机模式, 接收模式)

MSB 先行 (CBnCTL0.CBnDIR 位= 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00), 通信时钟 (f_{CCLK}) = 外部时钟 ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111), 传输数据长度= 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000)。

(1) 操作流程



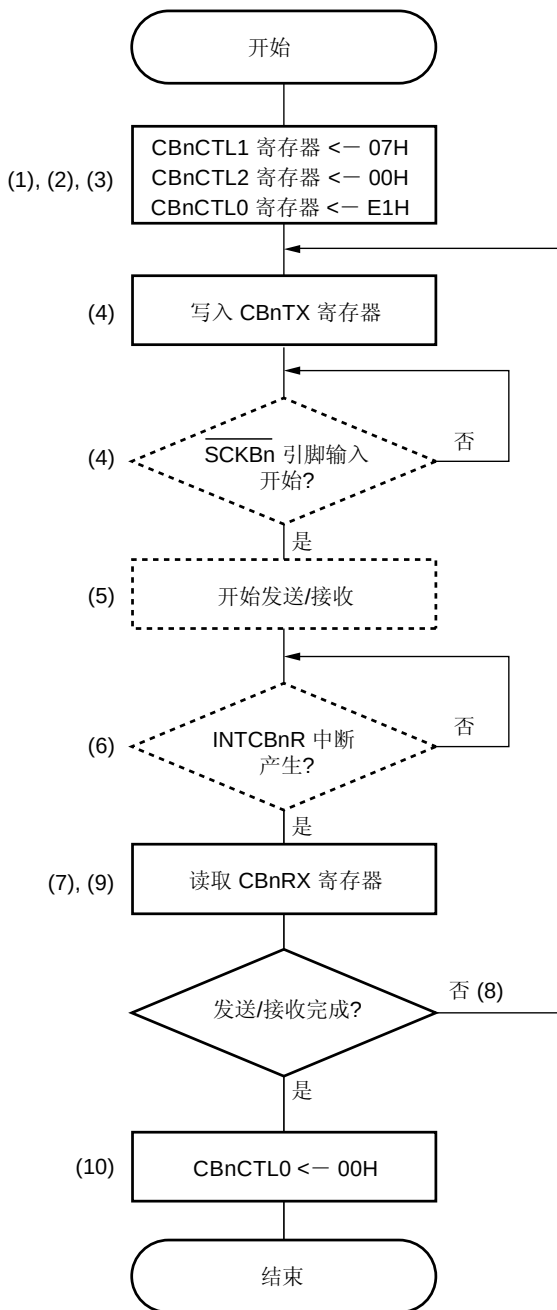
(2) 操作时序



16.6.6 单次传输模式（从机模式，发送/接收模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= 外部时钟（SCKBn）（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000）。

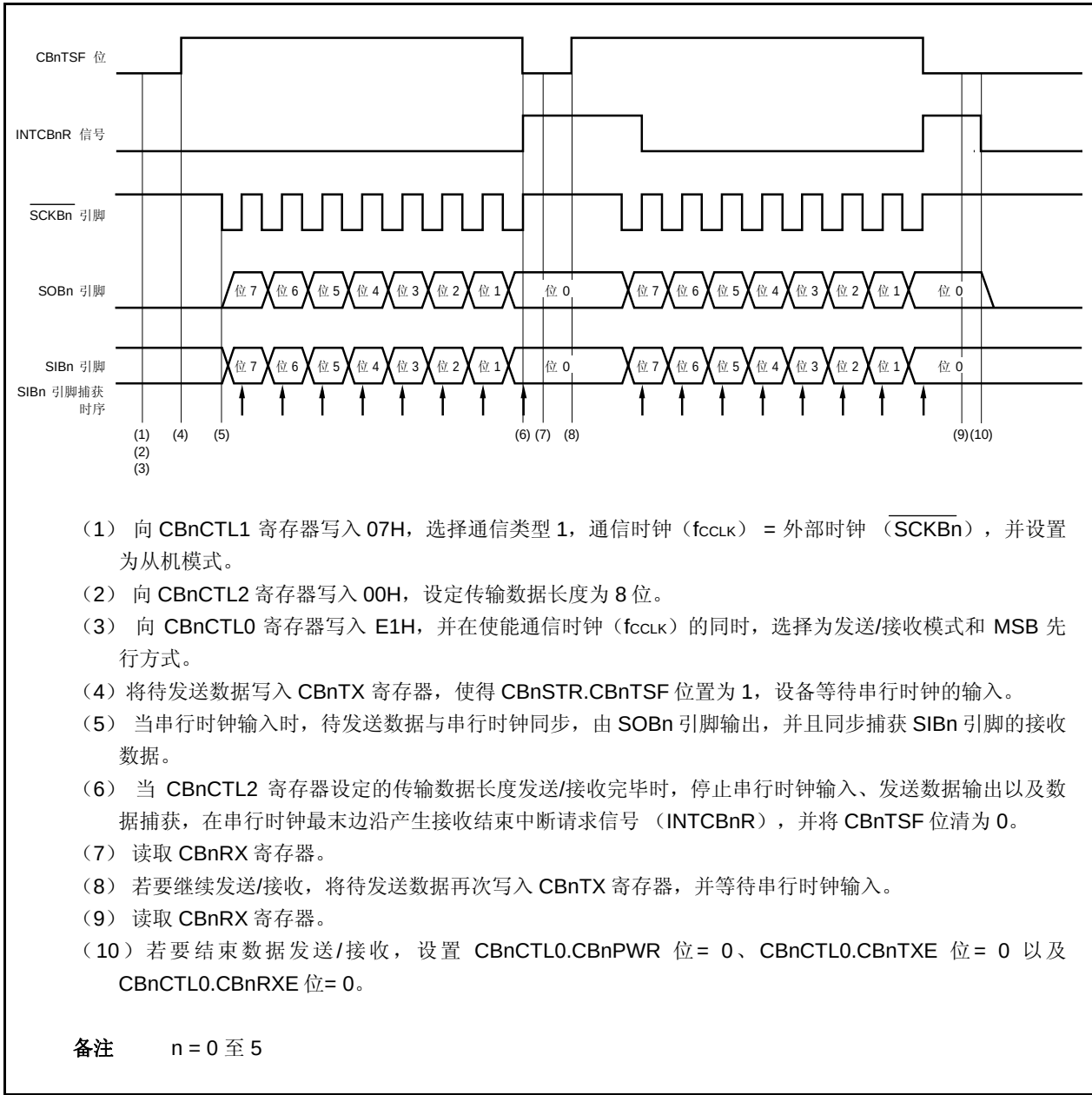
(1) 操作流程



备注

1. 虚线部分表示硬件处理。
2. 图中的数字对应于 (2) 操作时序中的数字。
3. n = 0 至 5

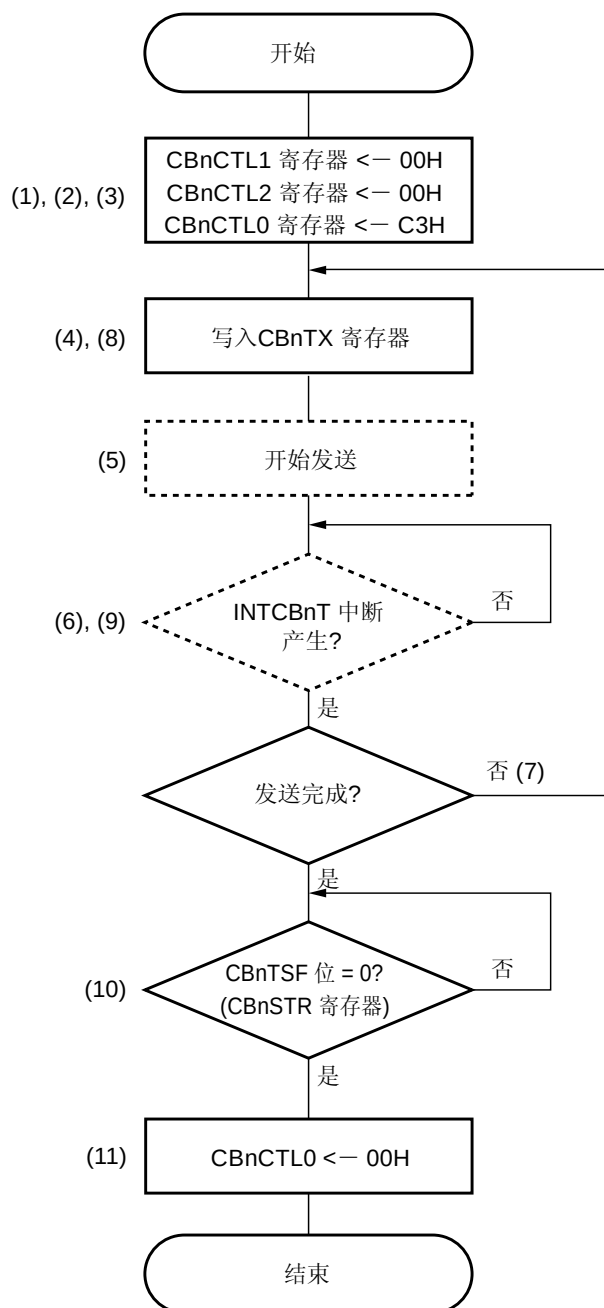
(2) 操作时序



16.6.7 连续传输模式（主机模式，发送模式）

MSB 先行（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 至 CBnCTL1.CBnCKS0 位 = 000），传输数据长度 = 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

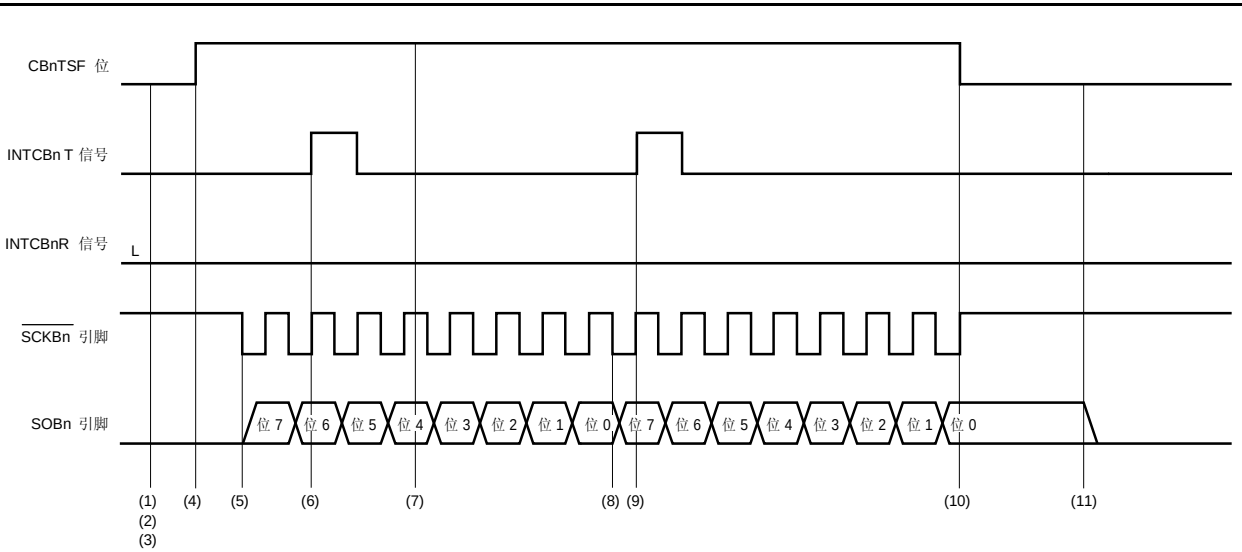
(1) 操作流程



备注

1. 虚线部分表示硬件处理。
2. 图中的数字对应于 (2) 操作时序中的数字。
3. n = 0 至 5

(2) 操作时序



- (1) 向 CBnCTL1 寄存器写入 00H, 选择通信类型 1, 通信时钟 (f_{CCLK}) = $f_{xx}/2$, 并设置为主机模式。
- (2) 向 CBnCTL2 寄存器写入 00H, 设定传输数据长度为 8 位。
- (3) 向 CBnCTL0 寄存器写入 C3H, 在使能通信时钟 (f_{CCLK}) 操作的同时, 选择为发送模式、MSB 先行以及连续传输模式。
- (4) 将待发送数据写入 CBnTX 寄存器使得 CBnSTR.CBnTSF 位置为 1, 启动发送。
- (5) 当发送开始后, 输出串行时钟至 SCKBn 引脚, 待发送数据与串行时钟同步, 由 SOBn 引脚输出。
- (6) 当发送数据从 CBnTX 寄存器传输至移位寄存器完毕, 且允许写入 CBnTX 寄存器时, 产生发送允许中断请求信号 (INTCBnT)。
- (7) 若要连续发送, 在产生 INTCBnT 信号之后, 再次将发送数据写入 CBnTX 寄存器。
- (8) 当通信完成之前, 新的发送数据已经写入 CBnTX 寄存器, 则在本次通信完成后立即开始下一次通信。
- (9) 从 CBnTX 寄存器传输发送数据至移位寄存器完毕, 且产生 INTCBnT 信号。若要结束当前的连续发送传输, 不要写入 CBnTX 寄存器。
- (10) 当传输完毕之前下一个发送数据未被写入 CBnTX 寄存器, 传输完毕后, 则停止输入到 SCKBn 引脚的串行时钟, 并将 CBnTSF 位清除为 0。
- (11) 若要解除发送使能状态, 在检查 CBnTSF 位 = 0 之后, 设置 CBnCTL0.CBnPWR 位 = 0 和 CBnCTL0.CBnTXE 位 = 0。

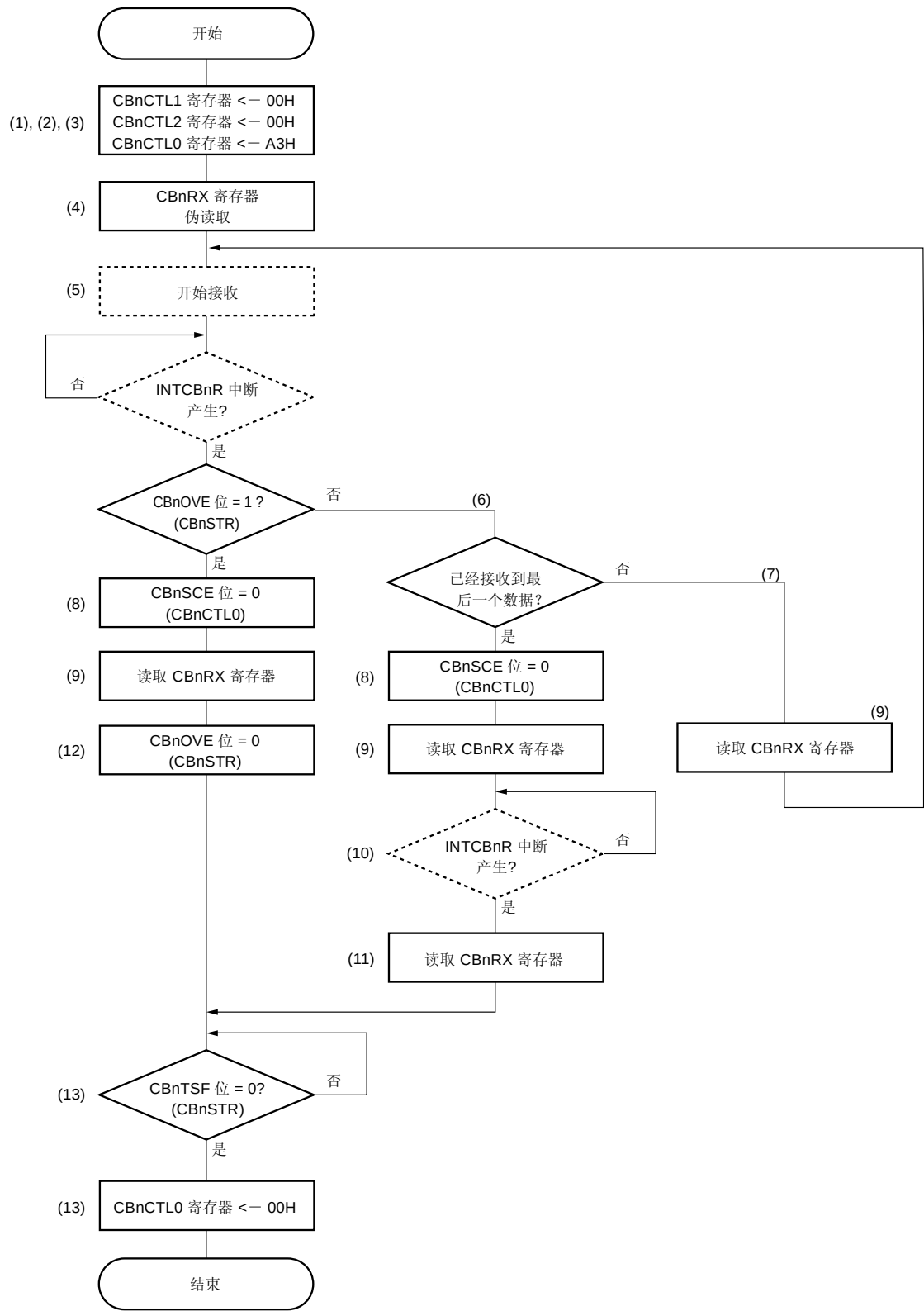
注意事项 连续发送模式下, 不产生接收完成中断请求信号 (INTCBnR)。

备注 n = 0 至 5

16.6.8 连续传输模式（主机模式，接收模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（ f_{CLK} ）= $f_{\text{xx}} / 2$ （CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

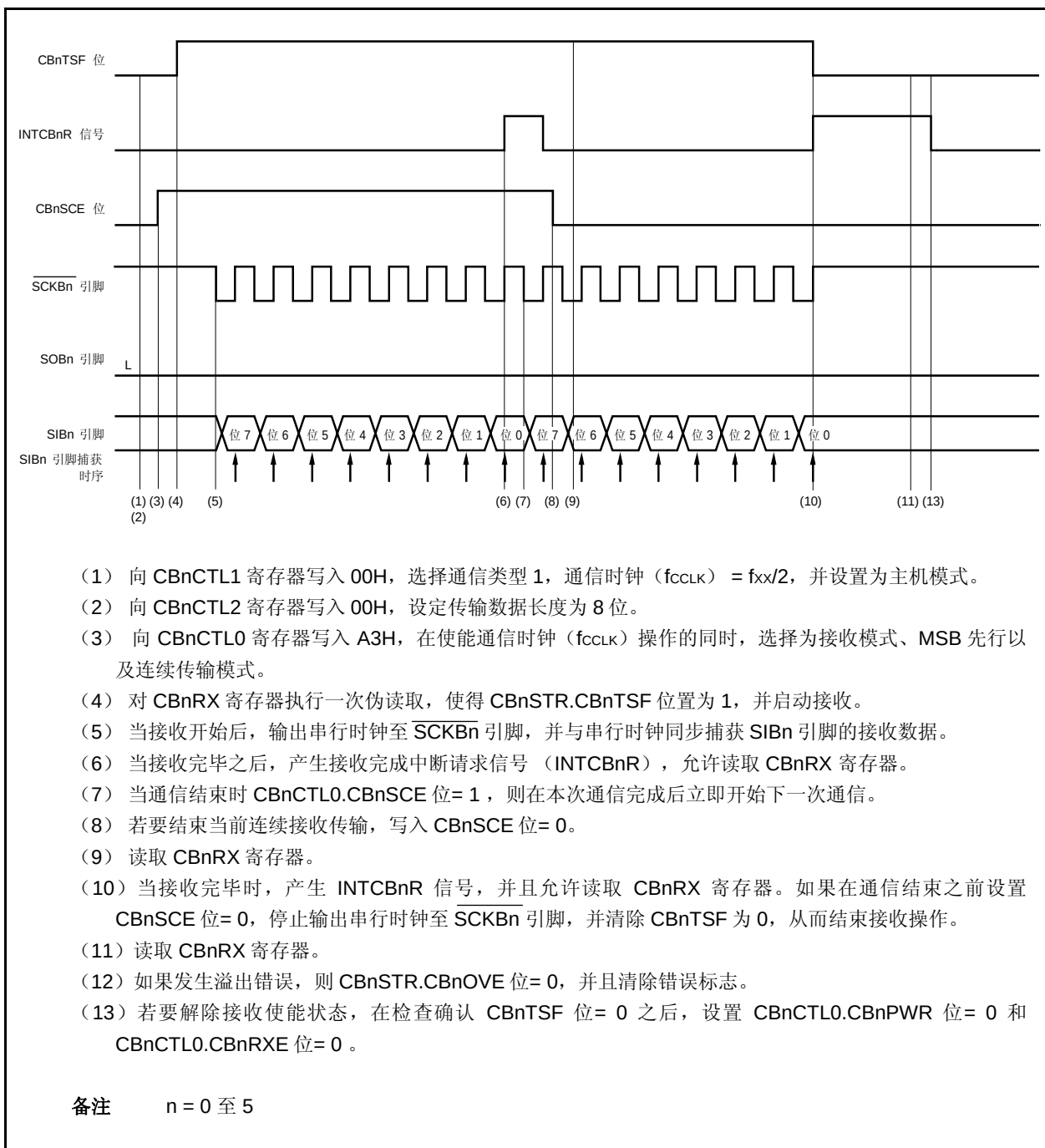
(1) 操作流程



备注

- 1. 虚线部分表示硬件处理。
- 2. 图中的数字对应于 (2) 操作时序中的数字。
- 3. n = 0 至 5

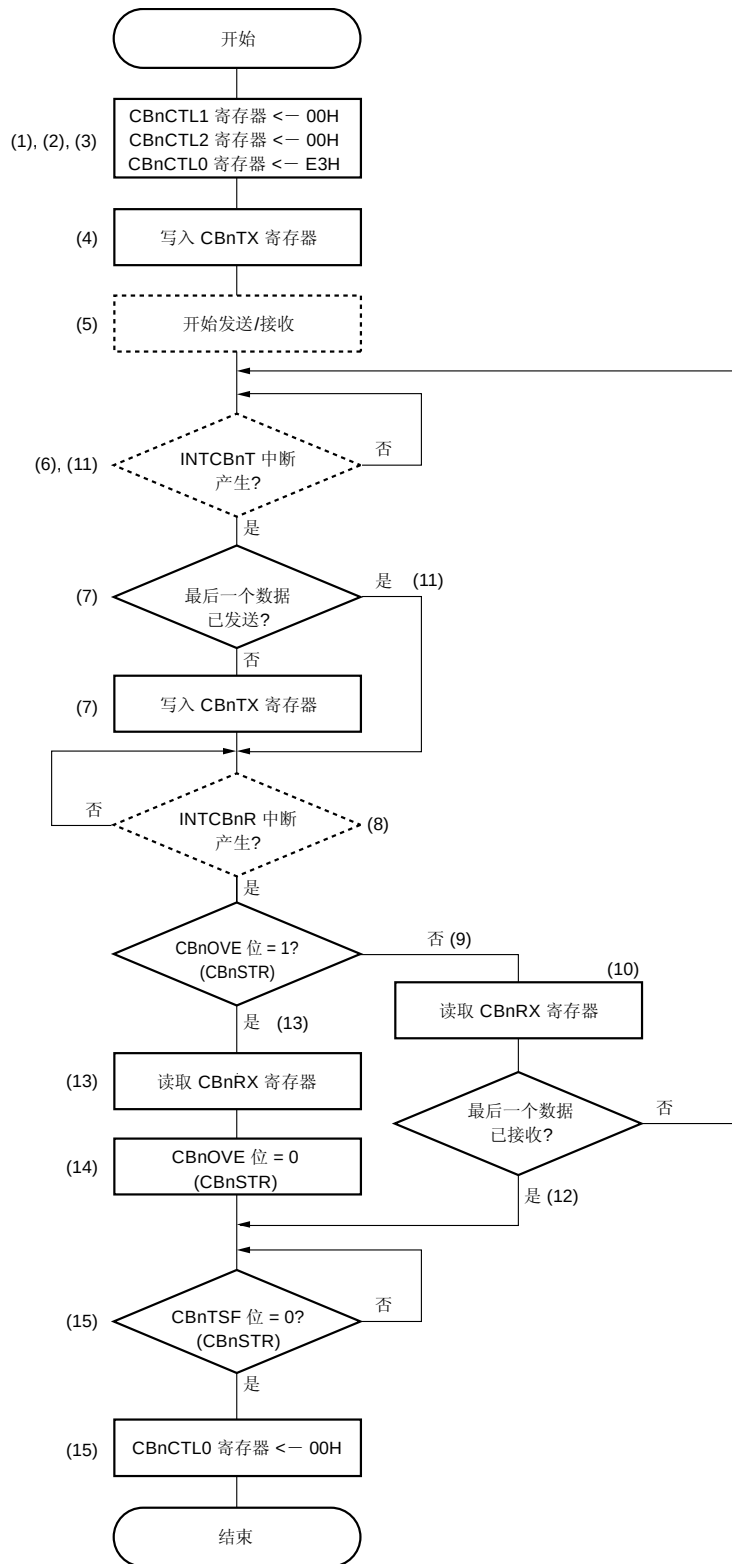
(2) 操作时序



16.6.9 连续传输模式（主机模式，发送/接收模式）

MSB 先行（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 至 CBnCTL1.CBnCKS0 位 = 000），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

(1) 操作流程

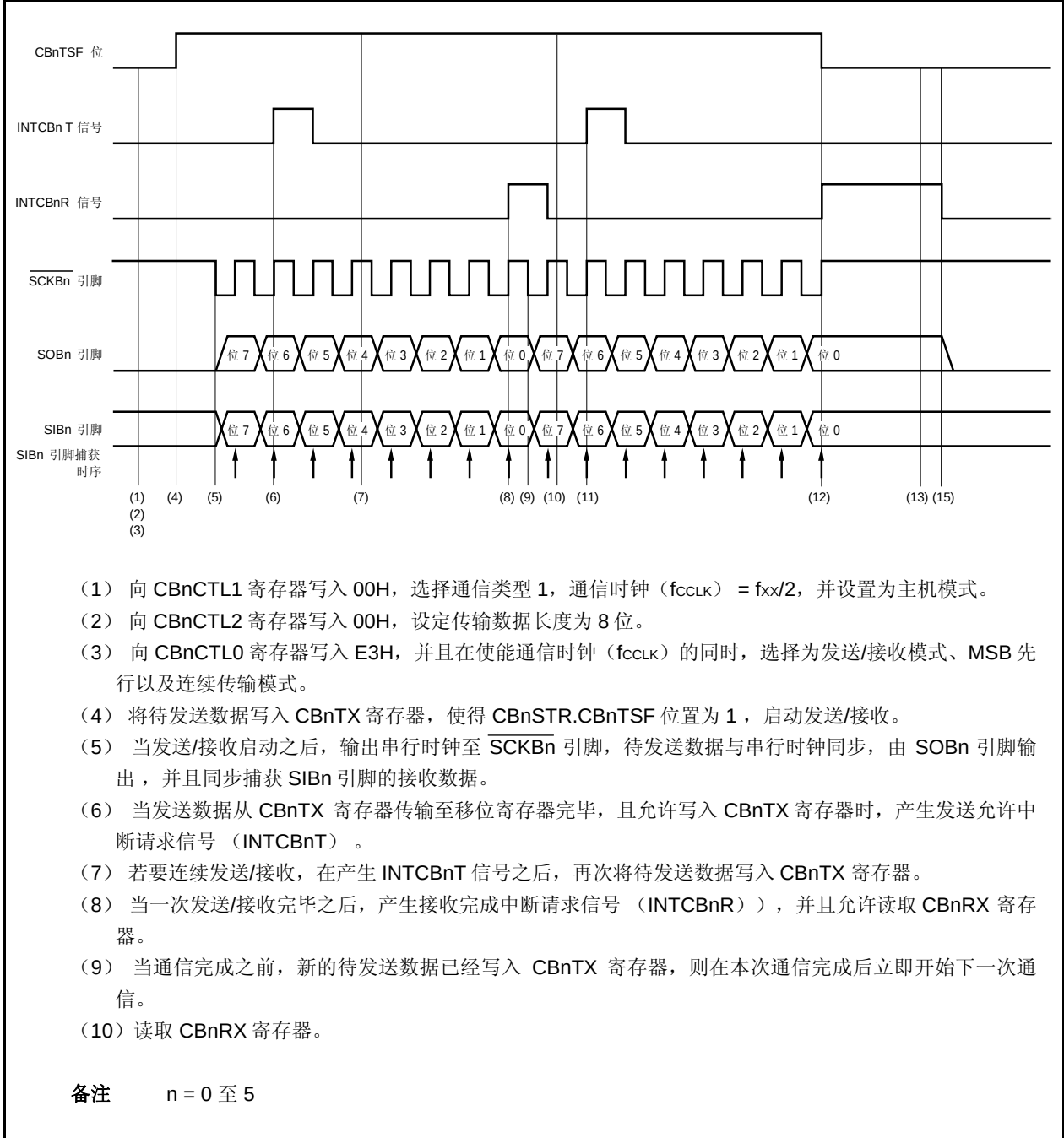


备注

1. 虚线部分表示硬件处理。
2. 图中的数字对应于 (2) 操作时序中的数字。
3. n = 0 至 5

(2) 操作时序

(1/2)



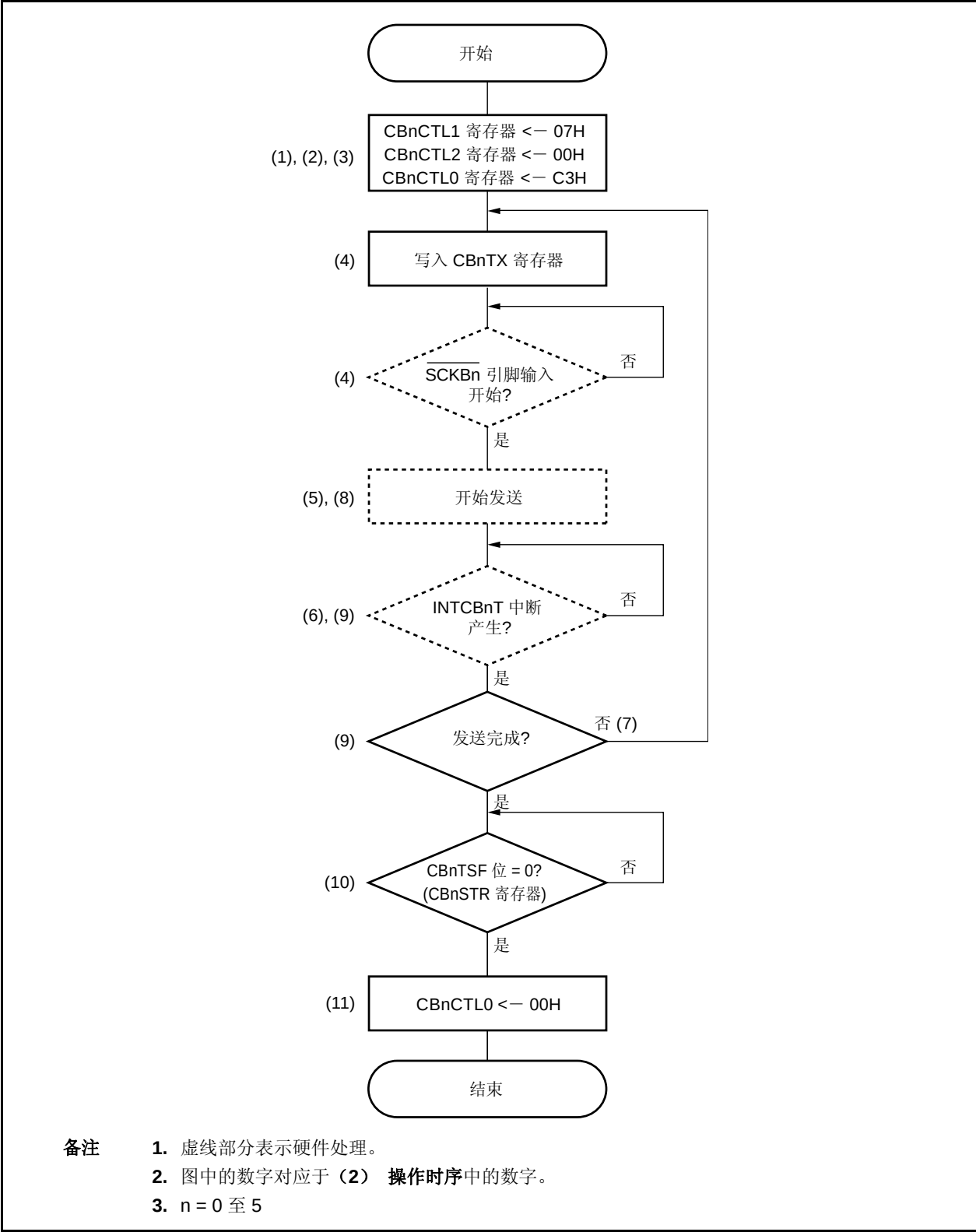
- (11) 从 CBnTX 寄存器传输发送数据至移位寄存器完毕，且产生 INTCBnT 信号。若要结束当前的发送/接收传输，则不写入 CBnTX 寄存器。
- (12) 当传输完毕之前下一个待发送数据未被写入 CBnTX 寄存器，传输完毕后，则停止输出串行时钟至 SCKBn 引脚，并将 CBnTSF 位清除为 0。
- (13) 发生接收错误中断请求信号 (INTCBnR) 时，读取 CBnRX 寄存器。
- (14) 如果发生溢出错误，则 CBnSTR.CBnOVE 位= 0，并且清除错误标志。
- (15) 若要解除发送/接收使能状态，在检查确认 CBnTSF 位= 0 之后，设置 CBnCTL0.CBnPWR 位= 0，CBnCTL0.CBnTXE 位= 0 以及 CBnCTL0.CBnRXE 位= 0。

备注 n = 0 至 5

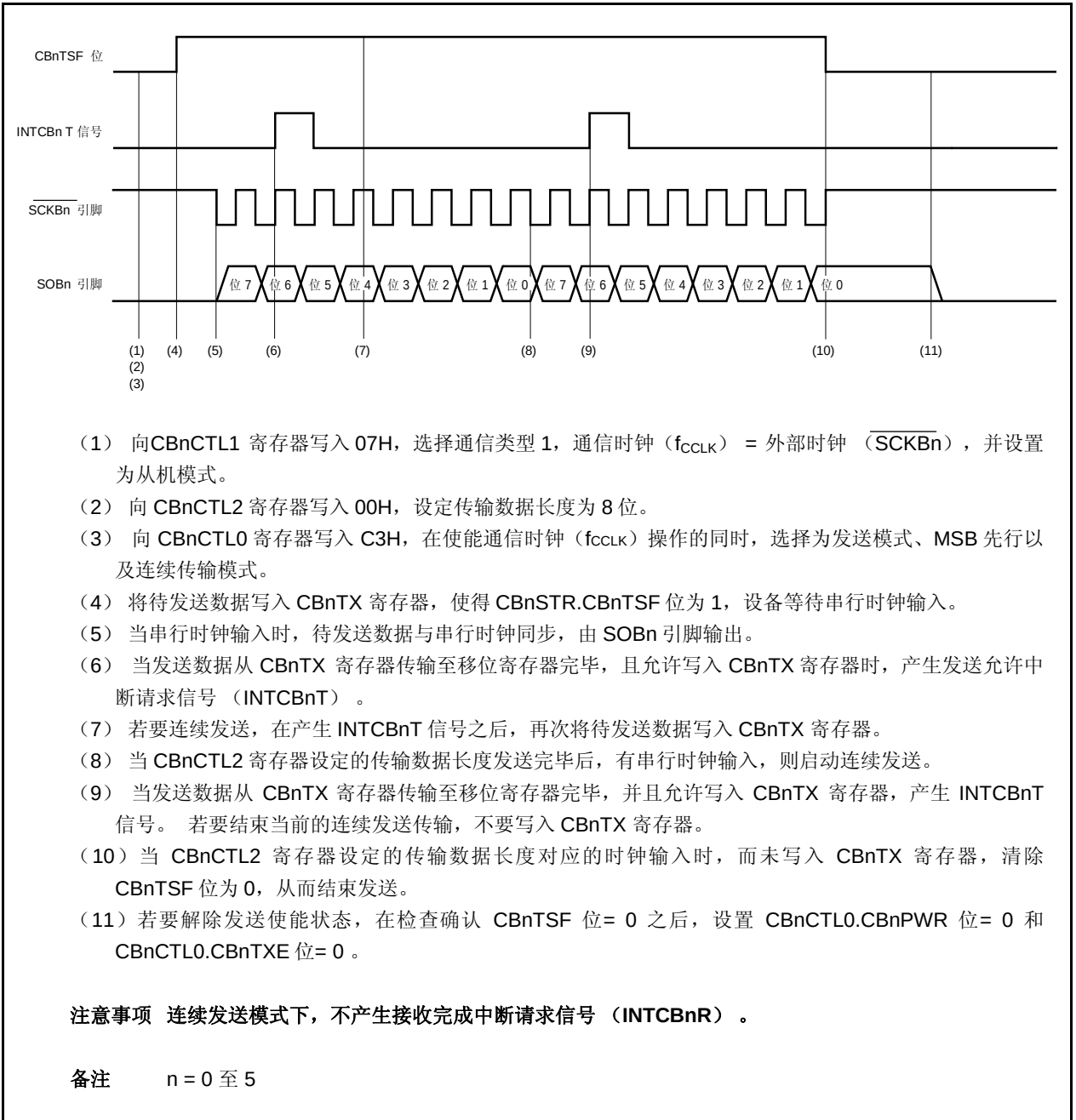
16.6.10 连续传输模式（从机模式，发送模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CLK}）= 外部时钟（SCKBn）（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

（1）操作流程



(2) 操作时序



16.6.11 连续传输模式（从机模式，接收模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= 外部时钟（SCKBn）（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

The flowchart illustrates the SPI Master Mode operation. It begins with an oval labeled '开始' (Start). A rectangular box contains the initialization of control registers: 'CBnCTL1 寄存器 ← 07H', 'CBnCTL2 寄存器 ← 00H', and 'CBnCTL0 寄存器 ← A3H'. This is followed by a box for 'CBnRX 寄存器 dummy 读取' (CBnRX register dummy read). A decision diamond asks 'SCKBn 引脚 输入开始?' (SCKBn pin input start?). If '否' (No), it loops back to the dummy read. If '是' (Yes), it proceeds to a dashed box labeled '接收开始' (Receive start). Another decision diamond asks 'INTCBnR 中断 产生?' (INTCBnR interrupt occur?). If '否' (No), it loops back to the '接收开始' box. If '是' (Yes), it checks 'CBnOVE 位 = 1? (CBnSTR)'. If '否' (No), it proceeds to a decision diamond '已经接收到最后一个数据?' (Already received the last data?). If '否' (No), it goes to a box '读取 CBnRX 寄存器' (Read CBnRX register) and then to a decision diamond 'INTCBnR 中断 产生?'. If '否' (No), it loops back to the '已经接收到最后一个数据?' diamond. If '是' (Yes), it goes to a box '读取 CBnRX 寄存器'. If '是' (Yes) for 'CBnOVE 位 = 1?', it goes to a box 'CBnSCE 位 = 0 (CBnCTL0)', then '读取 CBnRX 寄存器', then 'CBnOVE 位 = 0 (CBnSTR)', and then to a decision diamond 'CBnTSF 位 = 0? (CBnSTR)'. If '否' (No), it loops back to the '已经接收到最后一个数据?' diamond. If '是' (Yes), it goes to a box 'CBnCTL0 寄存器 ← 00H' and finally to an oval labeled '结束' (End).

```

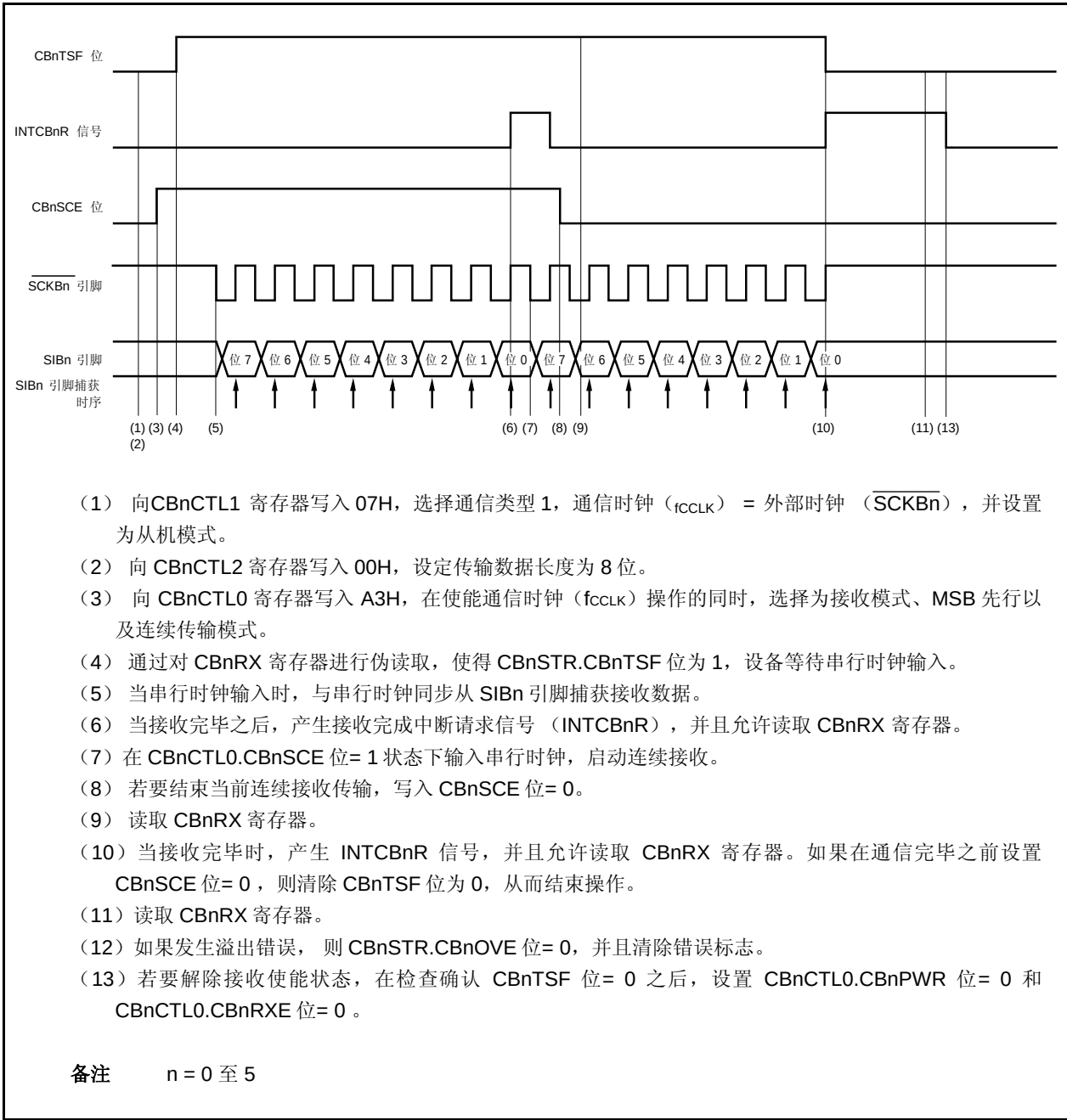
graph TD
    Start([开始]) --> Init[CBnCTL1 寄存器 ← 07H  
CBnCTL2 寄存器 ← 00H  
CBnCTL0 寄存器 ← A3H]
    Init --> Dummy[CBnRX 寄存器  
dummy 读取]
    Dummy --> SCKBn{SCKBn 引脚  
输入开始?}
    SCKBn -- 否 --> Dummy
    SCKBn -- 是 --> ReceiveStart[接收开始]
    ReceiveStart --> INTCBnR{INTCBnR 中断  
产生?}
    INTCBnR -- 否 --> ReceiveStart
    INTCBnR -- 是 --> CBnOVE1{CBnOVE 位 = 1?  
(CBnSTR)}
    CBnOVE1 -- 否 --> LastData{已经接收到最后  
一个数据?}
    CBnOVE1 -- 是 --> CBnSCE0[CBnSCE 位 = 0  
(CBnCTL0)]
    CBnSCE0 --> ReadCBnRX1[读取 CBnRX 寄存器]
    ReadCBnRX1 --> INTCBnR2{INTCBnR 中断  
产生?}
    INTCBnR2 -- 否 --> LastData
    INTCBnR2 -- 是 --> ReadCBnRX2[读取 CBnRX 寄存器]
    ReadCBnRX2 --> CBnTSF0{CBnTSF 位 = 0?  
(CBnSTR)}
    CBnTSF0 -- 否 --> LastData
    CBnTSF0 -- 是 --> CBnCTL0_00H[CBnCTL0 寄存器 ← 00H]
    CBnCTL0_00H --> End([结束])
    LastData -- 否 --> ReadCBnRX3[读取 CBnRX 寄存器]
    ReadCBnRX3 --> INTCBnR2
    LastData -- 是 --> ReadCBnRX3
  
```

备注

1. 虚线部分表示硬件处理。
2. 图中的数字和（2）操作时序中的处理数字相对应。
3. n = 0 至 5

1. 虚线部分表示硬件处理。
2. 图中的数字和 (2) 操作时序中的处理数字相对应。
3. $n = 0$ 至 5

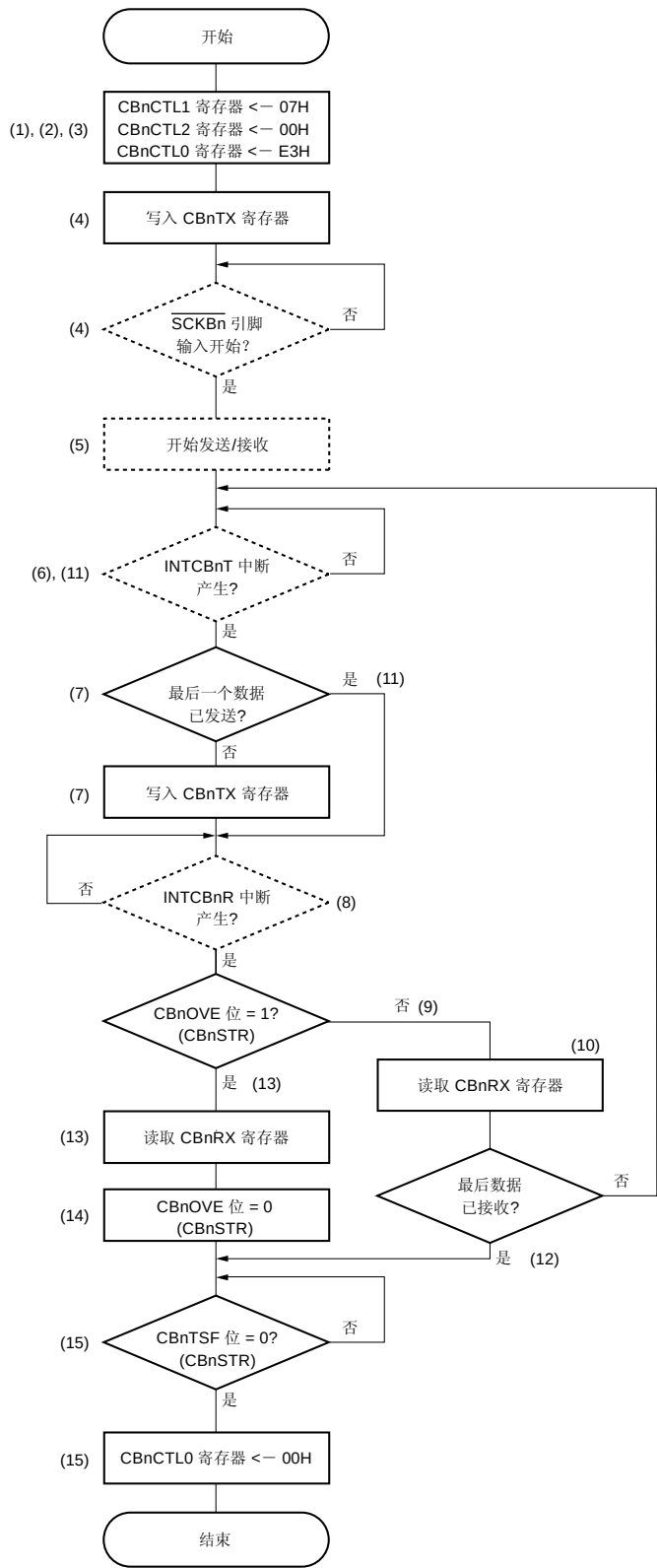
(2) 操作时序



16.6.12 连续传输模式（从机模式，发送/接收模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= 外部时钟（ $\overline{\text{SCKBn}}$ ）（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

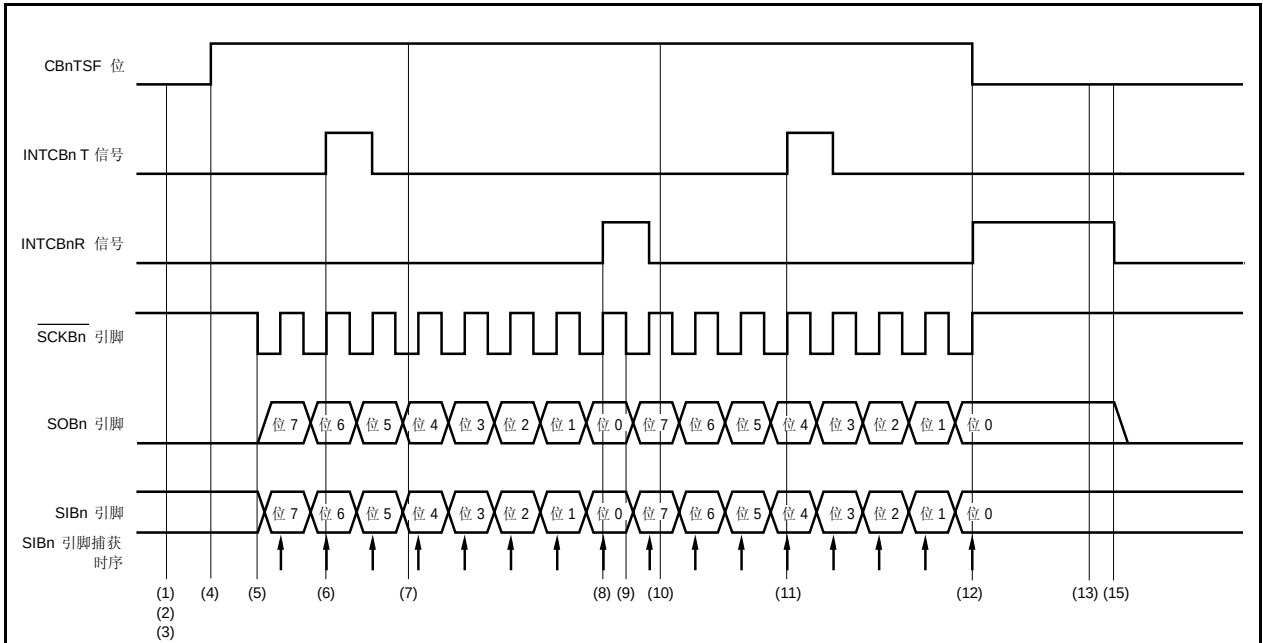
(1) 操作流程



- 备注
- 1. 虚线部分表示硬件处理。
 - 2. 图中的数字对应于 (2) 操作时序中的数字。
 - 3. n = 0 至 5

(2) 操作时序

(1/2)



- (1) 向 CBnCTL1 寄存器写入 07H, 选择通信类型 1, 通信时钟 (f_{CCLK}) = 外部时钟 (SCKBn), 并设置为从机模式。
- (2) 向 CBnCTL2 寄存器写入 00H, 设定传输数据长度为 8 位。
- (3) 向 CBnCTL0 寄存器写入 E3H, 在使能通信时钟 (f_{CCLK}) 的同时, 选择为发送/接收模式、MSB 先行以及连续传输模式。
- (4) 将待发送数据写入 CBnTX 寄存器, 使得 CBnSTR.CBnTSF 位置为 1, 设备等待串行时钟输入。
- (5) 当串行时钟输入时, 与串行时钟同步, 输出发送数据至 SOBn 引脚, 并且捕获 SIBn 引脚的接收数据。
- (6) 当发送数据从 CBnTX 寄存器传输至移位寄存器完毕, 且允许写入 CBnTX 寄存器时, 产生发送允许中断请求信号 (INTCBnT)。
- (7) 若要连续发送, 在产生 INTCBnT 信号之后, 再次将发送数据写入 CBnTX 寄存器。
- (8) 当 CBnCTL2 寄存器设定的传输数据长度接收完毕后, 产生接收完成中断请求信号 (INTCBnR), 并且允许读取 CBnRX 寄存器。
- (9) 当串行时钟连续输入, 启动连续发送/接收。
- (10) 读取 CBnRX 寄存器。
- (11) 当发送数据从 CBnTX 寄存器至移位寄存器传输完毕, 并且允许写入 CBnTX 寄存器, 产生 INTCBnT 信号。若要结束当前的发送/接收传输, 则不写入 CBnTX 寄存器。

备注 n = 0 至 5

- (12) 当 CBnCTL2 设定的传输数据长度相应的时钟输入时，而未写入 CBnTX 寄存器，则产生 INTCBnR 信号。将 CBnTSF 位清除为 0，从而结束发送/接收。
- (13) 产生 INTCBnR 信号时，读取 CBnRX 寄存器。
- (14) 如果发生溢出错误，则 CBnSTR.CBnOVE 位= 0，并且清除错误标志。
- (15) 若要解除发送/接收使能状态，在检查确认 CBnTSF 位= 0 之后，设置 CBnCTL0.CBnPWR 位= 0，CBnCTL0.CBnTXE 位= 0 以及 CBnCTL0.CBnRXE 位= 0。

备注 n = 0 至 5

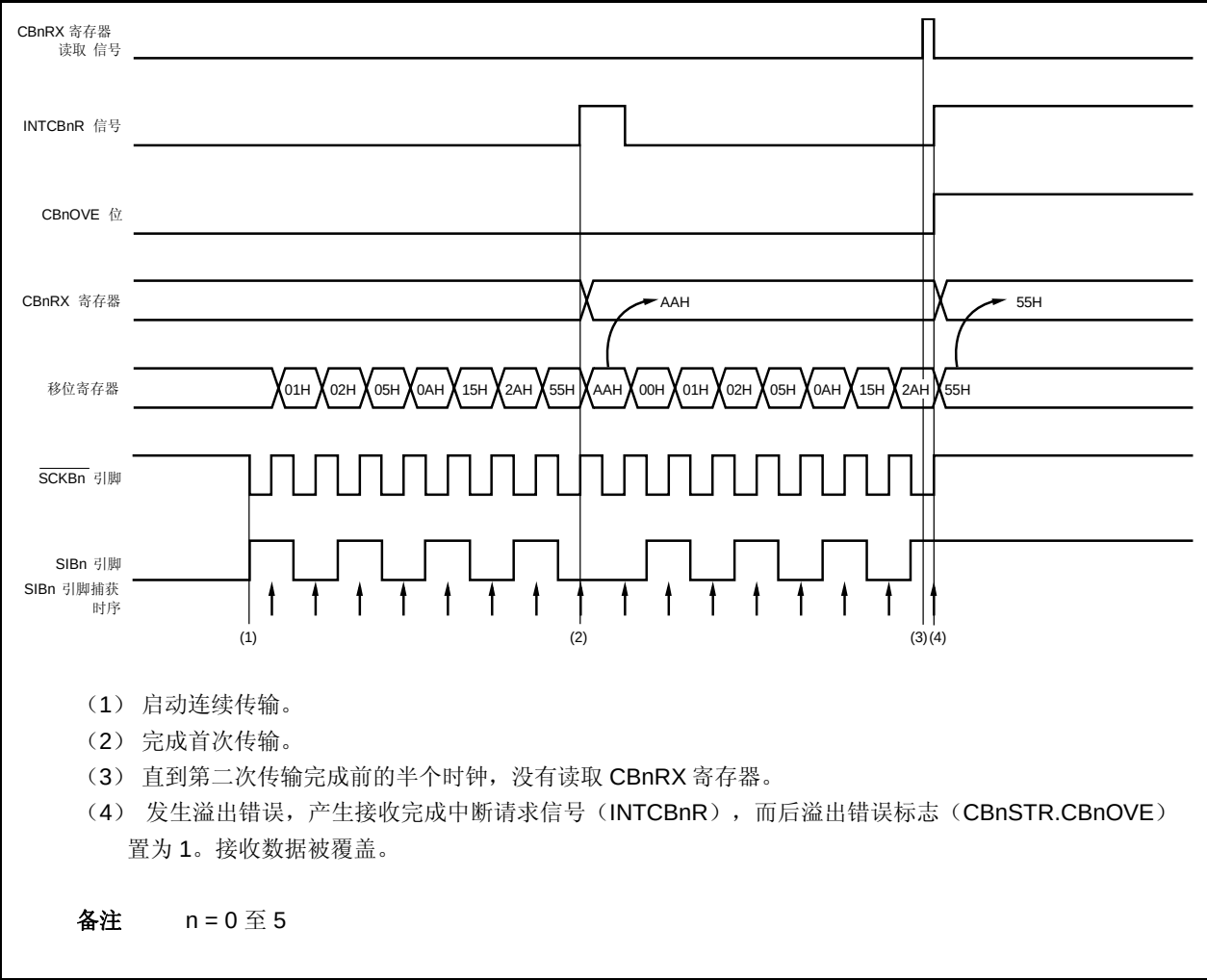
16.6.13 接收错误

连续传输模式下，执行传输时允许接收（CBnCTL0.CBnRXE 位= 1），INTCBnR 信号产生后，在读取 CBnRX 寄存器之前完成下一次接收操作时，则再次产生接收完成中断请求信号（INTCBnR），并且溢出错误标志（CBnSTR.CBnOVE）被置为 1。

即使发生了溢出错误，由于 CBnRX 已经被更新，先前接收到的数据丢失。即使发生了接收错误，如果在读取 CBnRX 寄存器之前下次接收完毕，则再次产生 INTCBnR 信号。

为了避免发生溢出错误，从产生 INTCBnR 信号到采样下次接收最末位数据之前的半个时钟，完成对 CBnRX 寄存器的读取。

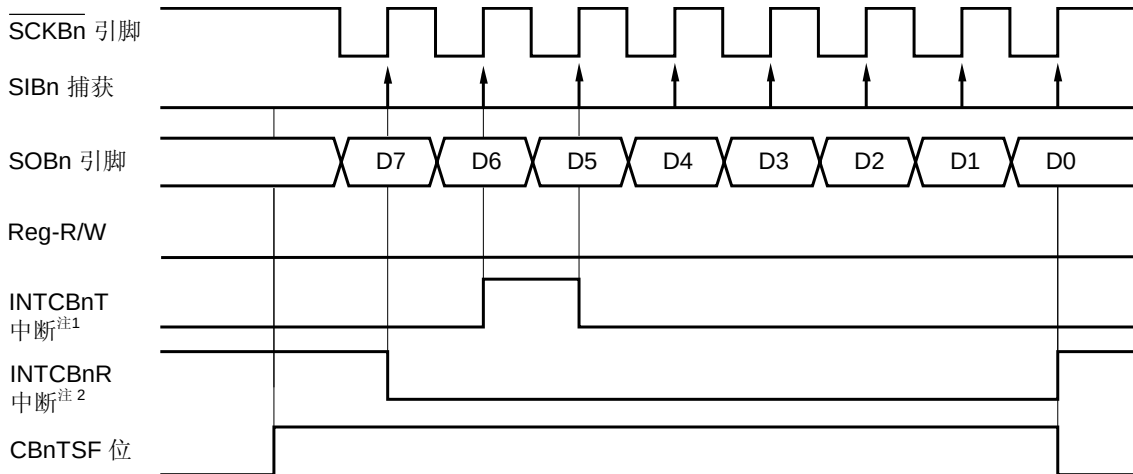
(1) 操作时序



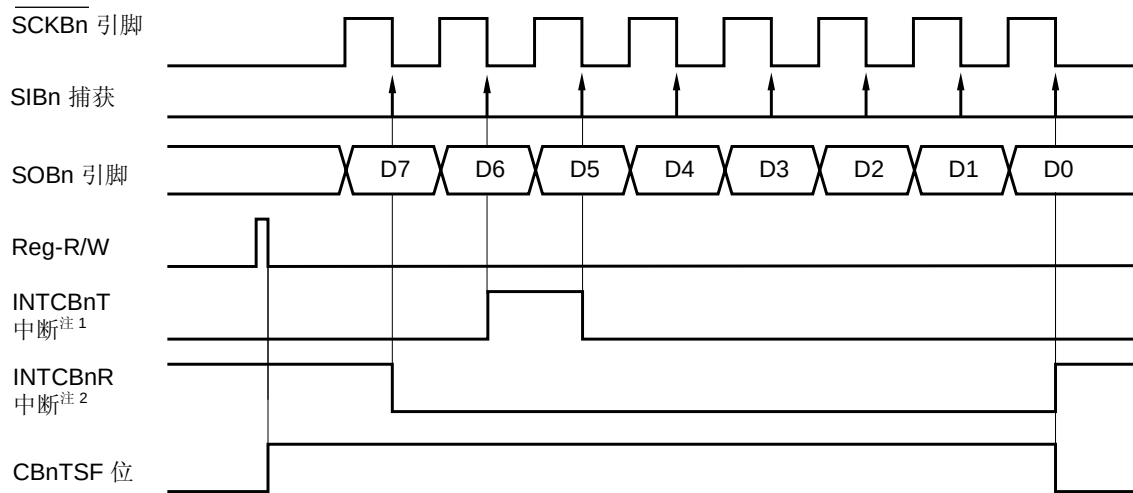
16.6.14 时钟时序

(1/2)

(i) 通信类型 1 (CBnCKP 和 CBnDAP 位 = 00)



(ii) 通信类型 3 (CBnCKP 和 CBnDAP 位 = 10)



注 1. 在连续发送模式或连续发送/接收模式下，当写入 CBnTX 寄存器的数据传输到数据移位寄存器时，INTCBnT 中断置位。在单次发送模式或单次发送/接收模式下，不产生 INTCBnT 中断请求信号，但是在通信结束时产生 INTCBnR 中断请求信号。

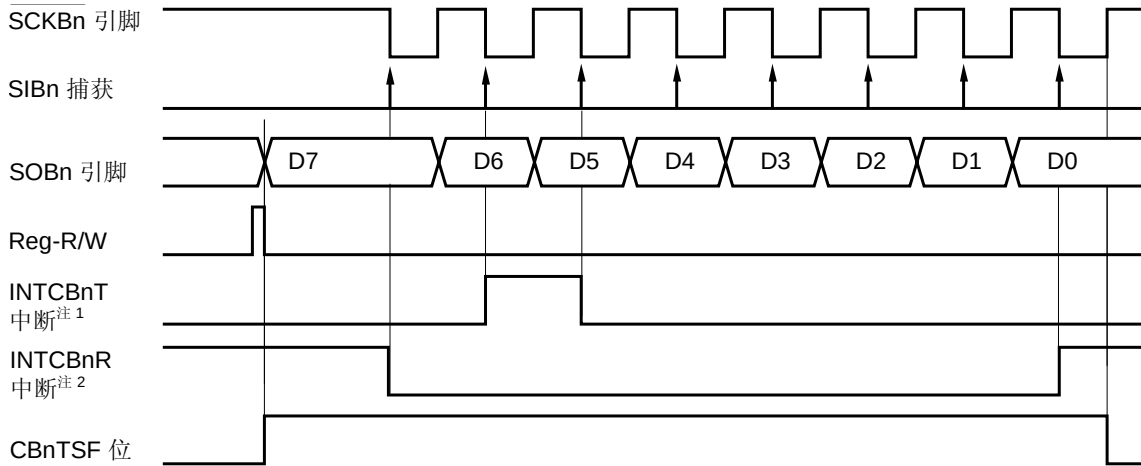
2. 如果接收正确结束，将产生 INTCBnR 中断，而且在接收使能的情况下，接收数据在 CBnRX 寄存器内就绪。在单次模式下，即使在发送模式下，仍然在通信结束时产生 INTCBnR 中断请求信号。

注意事项 在单次传输模式下，CBnTSF 位置 1 时对 CBnTX 寄存器的写入被忽略。这对传输过程的操作没有影响。

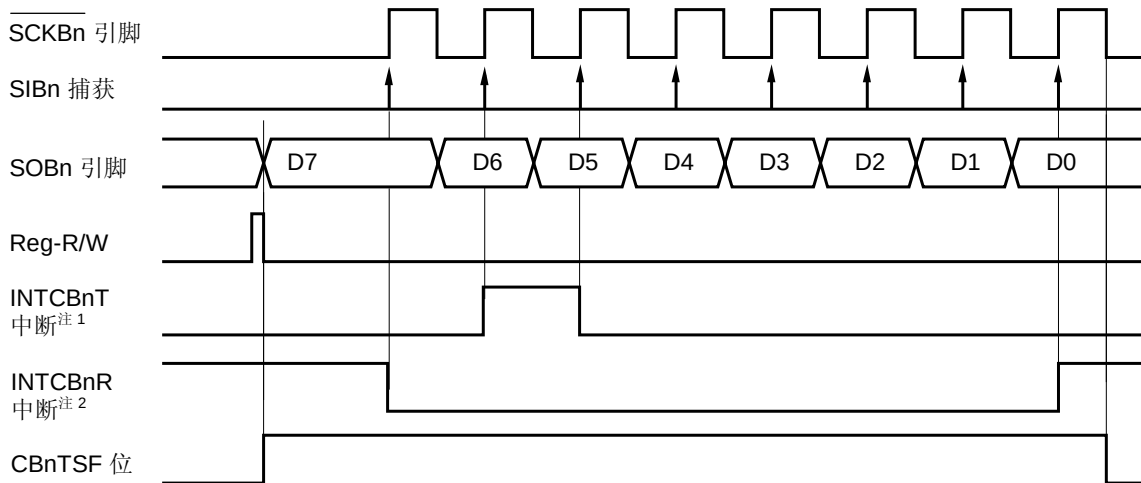
例如，通过产生的 INTCBnR 信号启动 DMA 时，若下一个数据被写入 CBnTX 寄存器，因为 CBnTSF 位为 1，被写入数据不能传输。

对于这样的应用，应该使用连续传输模式，而不要使用单次传输模式。

(iii) 通信类型 2 (CBnCKP 和 CBnDAP 位 = 01)



(iv) 通信类型 4 (CBnCKP 和 CBnDAP 位 = 11)



- 注**
1. 在连续发送模式或连续发送/接收模式下，当写入 CBnTX 寄存器的数据被传输到数据移位寄存器时，INTCBnT 中断置位。在单次发送模式或单次发送/接收模式下，不产生 INTCBnT 中断请求信号，但是在通信结束时产生 INTCBnR 中断请求信号。
 2. 如果接收正确结束，将产生 INTCBnR 中断，而且在接收使能的情况下，接收数据在 CBnRX 寄存器内就绪。在单次模式下，即使在发送模式下，仍然在通信结束时产生 INTCBnR 中断请求信号。

注意事项 在单次传输模式下，当 CBnTSF 位置为 1 时对 CBnTX 寄存器的写入被忽略。这对传输过程的操作没有影响。

例如，通过产生的 INTCBnR 信号启动 DMA 时，若下一个数据被写入 CBnTX 寄存器，因为 CBnTSF 位为 1，被写入数据不能传输。

对于这样的应用，应该使用连续传输模式，而不要使用单次传输模式。

16.7 输出引脚

(1) $\overline{\text{SCKBn}}$ 引脚

当 CSIBn 操作禁止 (CBnCTL0.CBnPWR 位= 0) 时, $\overline{\text{SCKBn}}$ 引脚输出状态如下所示。

CBnCKP	CBnCKS2	CBnCKS1	CBnCKS0	$\overline{\text{SCKBn}}$ 引脚输出
0	1	1	1	高阻抗状态
	其它			固定为高电平
1	1	1	1	高阻抗状态
	其它			固定为低电平

- 备注**
1. 如果 CBnCTL1.CBnCKP 和 CBnCKS2 至 CBnCKS0 位中有任何位被重写, $\overline{\text{SCKBn}}$ 引脚的输出电平改变。
 2. n = 0 至 5

(2) SOBn 引脚

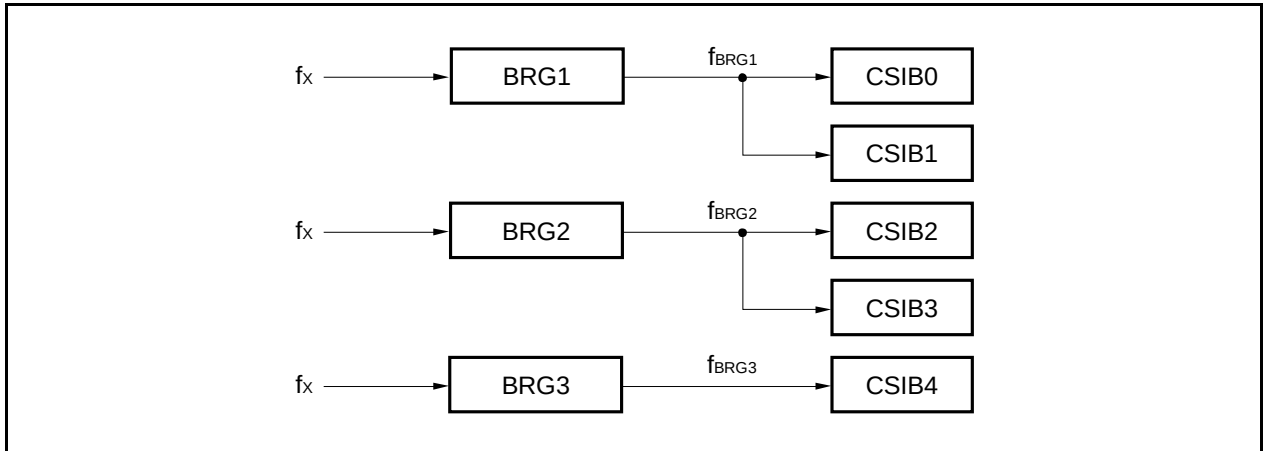
当 CSIBn 操作禁止 (CBnPWR 位= 0) 时, SOBn 引脚输出状态如下所示。

CBnTXE	CBnDAP	CBnDIR	SOBn 引脚输出
0	x	x	固定为低电平
1	0	x	SOBn 锁存值 (低电平)。
	1	0	CBnTX0 值 (MSB)
		1	CBnTX0 值 (LSB)

- 备注**
1. 如果 CBnCTL0.CBnTXE 位, CBnCTL0.CBnDIR 位以及 CBnCTL1.CBnDAP 位中有任何位被重写, SOBn 引脚输出会改变。
 2. x: 无需理会
 3. n = 0 至 5

16.8 波特率发生器

BRG1 至 BRG3 和 CSIB0 至 CSIB5 波特率发生器的连接如下所示：



(1) 预分频模式寄存器 1 至 3 (PRSM1 至 PRSM3)

PRSM1 至 PRSM3 寄存器用于控制 CSIB 波特率信号的发生。

这些寄存器可以按字节或按位进行读写。

系统复位后，这些寄存器被设置为 00H。

复位后：00H R/W 地址： PRSM1 FFFFF320H, PRSM2 FFFFF324H,
PRSM3 FFFFF328H

	7	6	5	<4>	3	2	1	0
PRSMm (m = 1 至 3)	0	0	0	BGCEm	0	0	BGCSm1	BGCSm0

BGCEm	波特率输出
0	禁止
1	使能

BGCSm1	BGCSm0	输入时钟选择 (f _{BGCSm})	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
1. 在操作期间，禁止重写 PRSMm 寄存器。
 2. 在设置 BGCEm 位为 1 之前，设置 PRSMm 寄存器。
 3. 请确保将第 7 位至第 5 位，第 3 位和第 2 位清“0”。

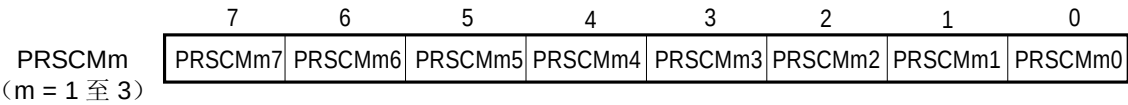
(2) 预分频比较寄存器 1 至 3 (PRSCM1 至 PRSCM3)

PRSCM1 至 PRSCM3 寄存器是 8 位比较寄存器。

这些寄存器可以按字节进行读取或写入。

系统复位后，这些寄存器被设置为 00H。

复位后：00H R/W 地址：PRSCM1 FFFF321H, PRSCM2 FFFF325H,
PRSCM3 FFFF329H



- 注意事项
1. 操作期间，禁止重写 PRSCMm 寄存器。
 2. 在设置 PRSMm.BGCEm 位为 1 之前，设置 PRSCMm 寄存器。

16.8.1 波特率的产生

通过对主时钟的分频产生发送/接收时钟。从主时钟产生的波特率由以下公式计算获得：

$$f_{BRGm} = \frac{f_{xx}}{2^{k+1} \times N}$$

注意事项 设置 f_{BRGm} 为 8 MHz 或更低。

- 备注
- f_{BRGm} : BRGm 计数时钟
 - f_{xx} : 主时钟振荡频率
 - k: PRSMm 寄存器设定值 = 0 至 3
 - N: PRSCMm 寄存器设定值 = 1 至 256
但是仅当 PRSCMm 寄存器设置为 00H 时，N = 256。
 - m = 1 至 3

16.9 注意事项

(1) 使用 DMA 传输发送数据和接收数据时，即使串行传输期间发生溢出错误，也不执行错误处理。DMA 传输完成后通过读取 CBnSTR.CBnOVE 位来检查是否发生溢出错误。

(2) 关于操作期间 (CBnCTL0.CBnPWR 位 = 1) 禁止重写的寄存器，如果操作期间进行了错误的重写，须将 CBnCTL0.CBnPWR 位清 0，然后初始化 CSIBn。

操作期间禁止重写的寄存器列举如下：

- 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

(3) 通信类型 2 和 4 (CBnCTL1.CBnDAP 位 = 1) 中，在发生接收完成中断 (INTCBnR) 后半 $\overline{\text{SCKBn}}$ 时钟时，CBnSTR.CBnTSF 位被清除。

单次传输模式下，通信期间 (CBnTSF 位 = 1) 写入的下一个待发送数据被忽略，并且不启动下次通信。同样如果设置为只接收通信 (CBnCTL0.CBnTXE 位 = 0，CBnCTL0.CBnRXE 位 = 1)，即使在通信期间读取接收数据，也不启动下一次通信。

因此，在通信类型 2 或 4 时使用单次传输模式，需特别注意以下情况：

- 若要启动下一次发送，在确认 CBnTSF 位 = 0 之后，将待发送数据写入 CBnTX 寄存器。
- 设置为只接收通信 (CBnTXE 位 = 0，CBnRXE 位 = 1) 时，若要连续执行下一次接收，在确认 CBnTSF 位 = 0 之后，读取 CBnRX 寄存器。

或者使用连续传输模式来代替单次传输模式。尤其在用 DMA 传输时，建议使用连续传输模式。

备注 n = 0 至 5

第十七章 I²C 总线

要使用 I²C 总线功能，分别使用 P38/SDA00，P39/SCL00，P40/SDA01，P41/SCL01，P90/SDA02 和 P91/SCL02 引脚作为串行发送/接收数据输入/输出引脚（SDA00 至 SDA02）和串行时钟输入/输出引脚（SCL00 至 SCL02），并且将它们设置为N沟道漏极开路输出。

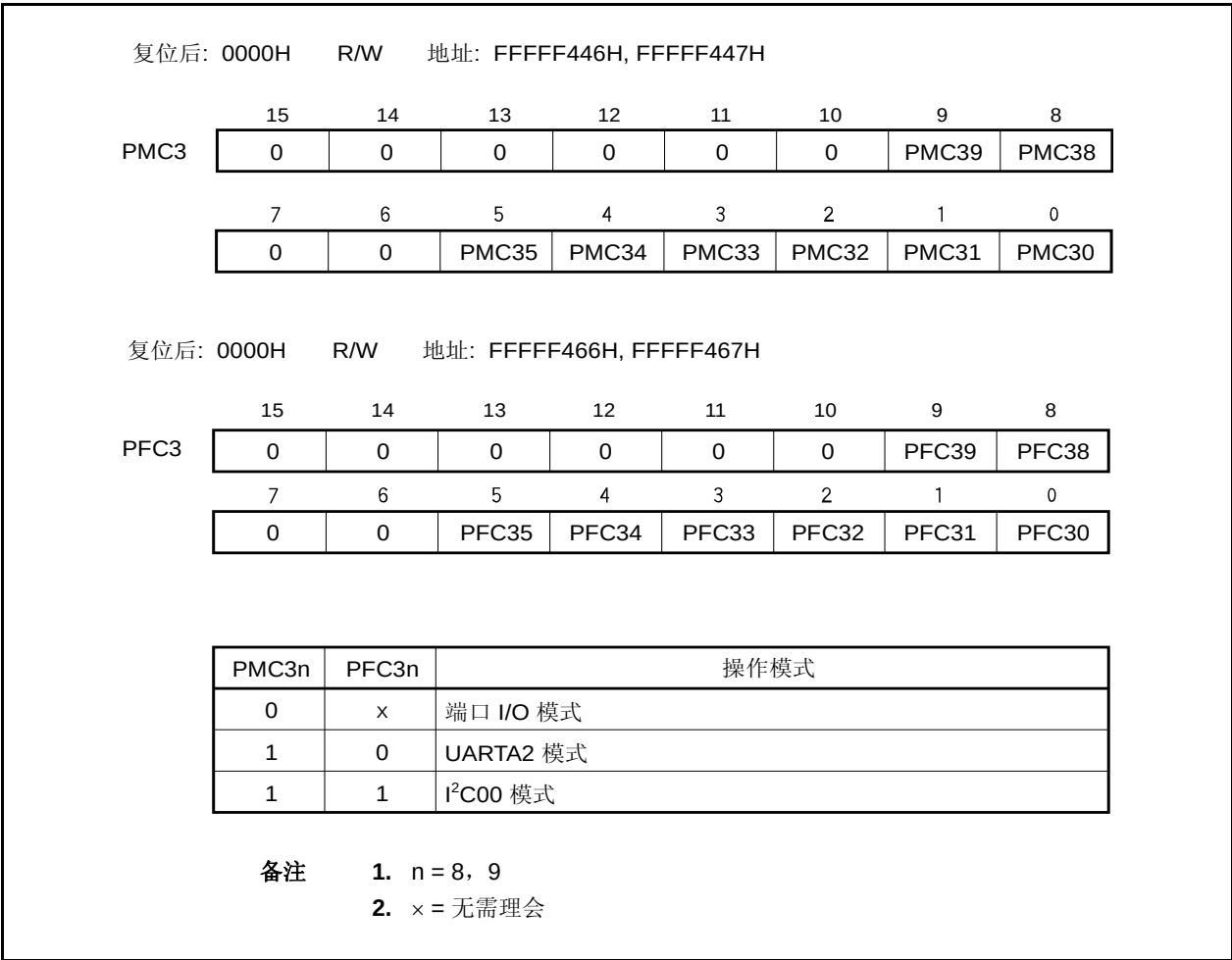
17.1 I²C总线和其它串行接口的模式转换

17.1.1 UARTA2 和I²C00 模式转换

V850ES/JJ3 单片机中，UARTA2 和I²C00 是相同引脚的复用功能，因此不能同时使用。使用前，通过PMC3 和 PFC3 寄存器预先设置I²C00。

注意事项 如果发送或接收期间切换这些功能，则UARTA2 和I²C00 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 17-1. UARTA2 和I²C00 模式转换设置

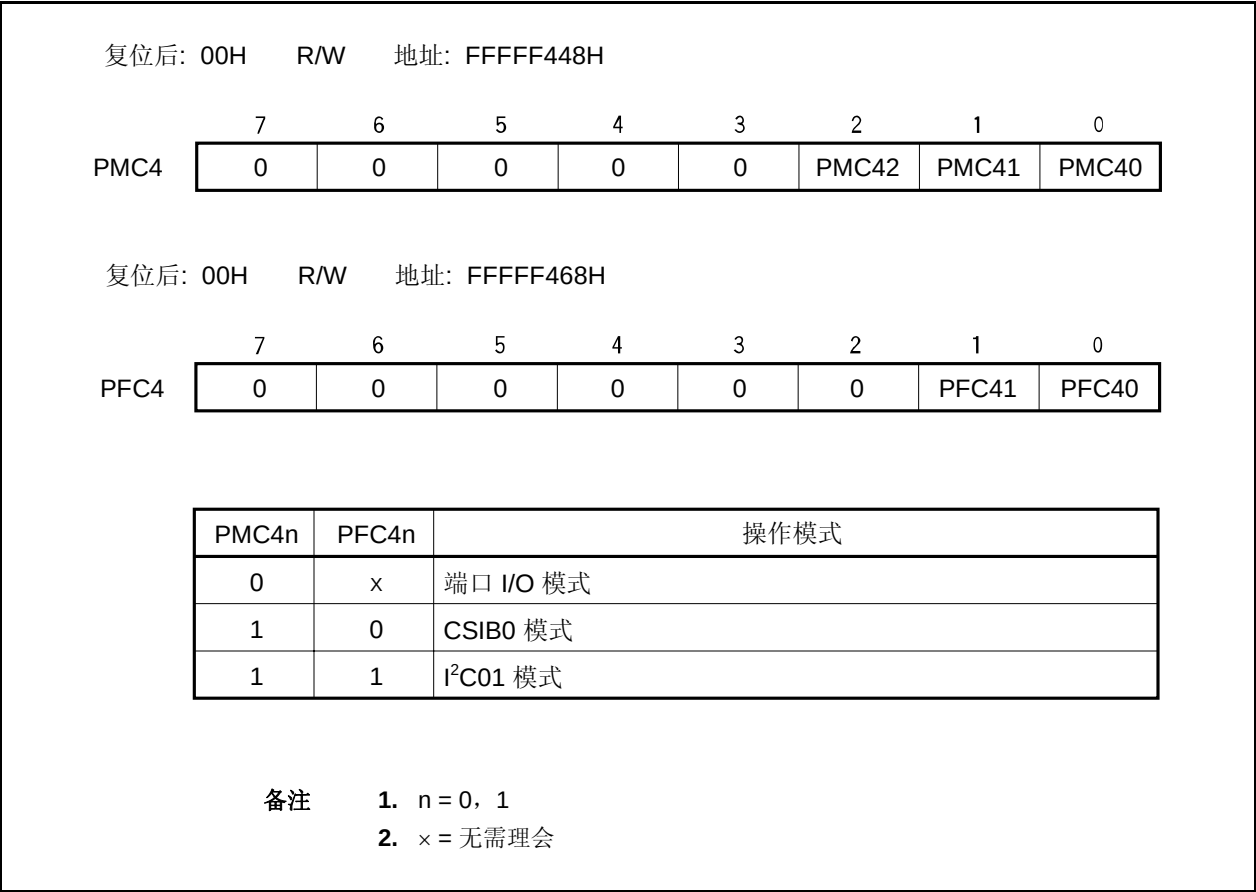


17.1.2 CSIB0 和 I²C01 模式转换

V850ES/JJ3 单片机中，CSIB0 和 I²C01 是相同引脚的复用功能，因此不能同时使用。使用前，通过PMC4 和 PFC4 寄存器预先设置I²C01。

注意事项 如果发送或接收期间切换这些功能，则CSIB0 和 I²C01 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 17-2. CSIB0 和 I²C01 模式转换设置



17.1.3 UARTA1 和 I²C02 模式转换

V850ES/JJ3 单片机中，UARTA1 和 I²C02 是相同引脚的复用功能，因此不能同时使用。使用前，通过 PMC9、PFC9 和 PMCE9 寄存器预先设置 I²C02。

注意事项 如果发送或接收期间切换这些功能，则 UARTA1 和 I²C02 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 17-3. UARTA1 和 I²C02 模式转换设置

复位后: 0000H

R/W

地址: FFFFF452H, FFFFF453H

PMC9

15	14	13	12	11	10	9	8
PMC915	PMC914	PMC913	PMC912	PMC911	PMC910	PMC99	PMC98
7	6	5	4	3	2	1	0
PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

复位后: 0000H

R/W

地址: FFFFF472H, FFFFF473H

PFC9

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

复位后: 0000H

R/W

地址: FFFFF712H, FFFFF713H

PFCE9

15	14	13	12	11	10	9	8
PFCE915	PFCE914	0	0	0	0	0	0
7	6	5	4	3	2	1	0
PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

PMC9n	PFCE9n	PFC9n	操作模式
1	1	0	UARTA1 模式
1	1	1	I ² C02 模式

备注

n = 0, 1

17.2 特性

I²C00 至 I²C02 具有下列两种模式：

- 操作停止模式
- I²C（IC内部）总线模式（支持多主设备）

（1） 操作停止模式

在该模式下，不执行串行传输，因此可以降低功耗。

（2） I²C 总线模式（支持多主设备）

该模式通过两条信号线在多个设备之间进行 8 位数据传输：一个串行时钟引脚（SCL0n）和一个串行数据总线引脚（SDA0n）。

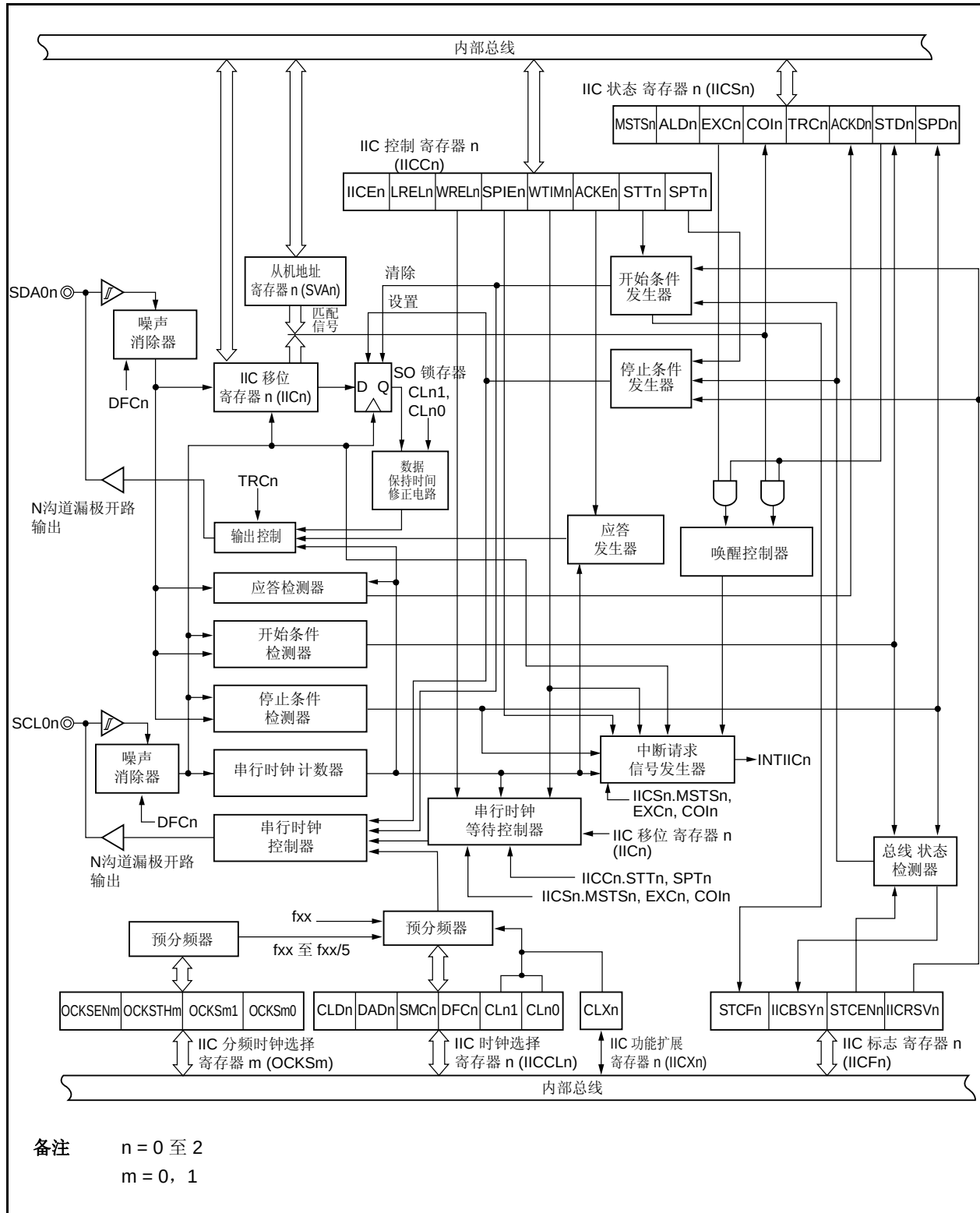
该模式符合I²C 总线规范格式，主设备可以通过其串行数据总线，为从设备产生“开始条件”、“地址”、“传输方向规范”、“传据”和“停止条件”等数据。从设备通过硬件自动检测接收到的状态和数据。这种功能可以使控制I²C总线的应用程序得到简化。

由于SCL0n 和 SDA0n 引脚被用于 N沟道漏极开路输出，因此I²C0n 的串行时钟线和串行数据线都需要上拉电阻。

备注 n = 0 至 2

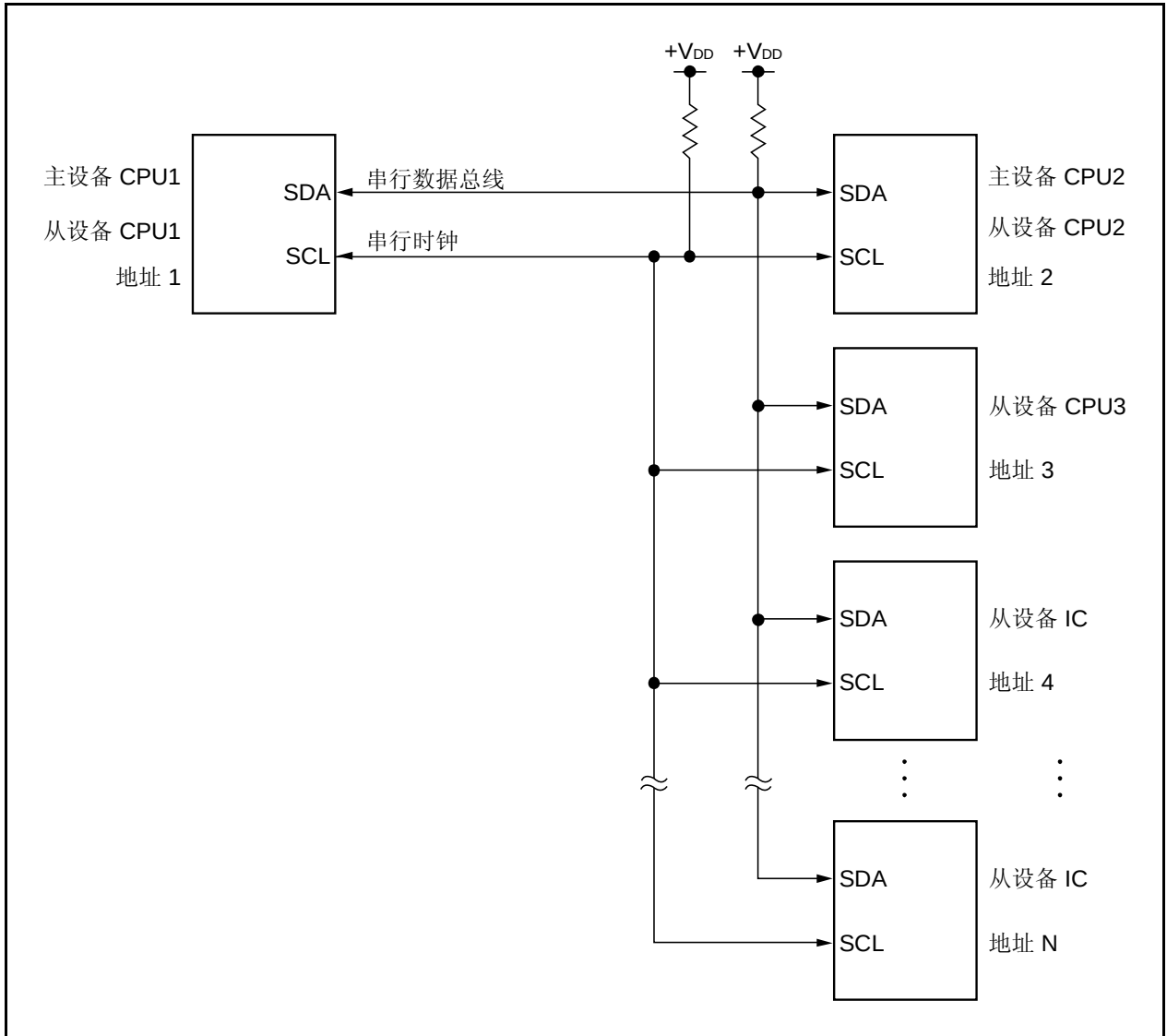
17.3 配置

I²C0n的功能框图如下所示：

图 17-4. I²C0n的框图

以下展示了一个串行总线结构配置示例。

图 17-5. 使用I²C 总线的串行总线配置示例



I²C0n包括下列硬件（n = 0 至 2）。

表 17-1. I²C0n的配置

项目	配置
寄存器	IIC 移位寄存器 n（IICn） 从设备地址寄存器 n（SVAn）
控制寄存器	IIC 控制寄存器 n（IICCN） IIC 状态寄存器 n（IICSn） IIC 标志寄存器 n（IICFn） IIC 时钟选择寄存器 n（IICCLn） IIC 功能扩展寄存器 n（IICXn） IIC 时钟分频选择寄存器 0, 1（OCKS0, OCKS1）

(1) IIC 移位寄存器 n（IICn）

IICn 寄存器将 8 位串行数据转换为 8 位并行数据，也可将 8 位并行数据转换为 8 为串行数据，可用于发送和接收（n = 0 至 2）。

对 IICn 寄存器的写入和读取操作被用来控制实际的发送和接收操作。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被置为 00H。

(2) 从设备地址寄存器 n（SVAn）

在从模式下，SVAn 寄存器用于设定本地地址（n = 0 至 2）。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被置为 00H。

(3) SO 锁存器

SO 锁存器用于保持 SDA0n 引脚的输出电平（n = 0 至 2）。

(4) 唤醒控制器

该寄存器收到的地址和 SVAn 寄存器设定的地址相匹配时，或接收到扩展码时，该电路就会产生一个中断请求信号（INTIICn）（n = 0 至 2）。

(5) 预分频器

用于选择使用的采样时钟。

(6) 串行时钟计数器

在发送/接收操作过程中，串行时钟计数器对输出的串行时钟和输入的串行时钟进行计数，且用于验证发送或接收到的 8 位数据。

(7) 中断请求信号发生器

该电路控制中断请求信号 (INTIICn) 的产生。

以下两种触发都可以产生一个 I²C 中断。

- 串行时钟的第 8 个或第 9 个 (由 IICn.WTIMn 位设定) 时钟的下降沿。
- 因为检测到停止条件 (由 IICn.SPIEn 位设定) 而发生中断。

备注 n = 0 至 2

(8) 串行时钟控制器

在主模式下, 该电路从采样时钟生成时钟并经由 SCL0n 引脚输出 (n = 0 至 2)。

(9) 串行时钟等待控制器

该电路控制等待时序。

(10) $\overline{\text{ACK}}$ 发生器、停止条件检测器、开始条件检测器以及 $\overline{\text{ACK}}$ 检测器

这些电路用来产生和检测各种状态。

(11) 数据保持时间修正电路

该电路根据相应的 SCL0n 引脚下降沿, 为数据产生保持时间。

(12) 开始条件发生器

当 IICn.STTn 位被置位时, 产生一个开始条件。

但是, 在通信保留禁止状态 (IICFn.IICRSVn 位 = 1) 下, 如果总线未释放 (IICFn.IICBSYn 位 = 1), 则请求被忽略, 且 IICFn.STCFn 位被置为 1。

(13) 停止条件发生器

当 IICn.SPTn 位被置位时, 产生一个停止条件。

(14) 总线状态检测器

通过检测开始条件和停止条件, 确认总线是否被释放。

但是, 总线状态不能在操作后立即被检测到, 因此使用 IICFn.STCENn 位将总线状态检测器设置为初始状态。

17.4 寄存器

I²C00 至 I²C02 由下列寄存器进行控制：

- IIC 控制寄存器 0 至 2 (IICC0 至 IICC2)
- IIC 状态寄存器 0 至 2 (IICS0 至 IICS2)
- IIC 标志寄存器 0 至 2 (IICF0 至 IICF2)
- IIC 时钟选择寄存器 0 至 2 (IICCL0 至 IICCL2)
- IIC 功能扩展寄存器 0 至 2 (IICX0 至 IICX2)
- IIC 时钟分频选择寄存器 0, 1 (OCKS0, OCKS1)

以下寄存器也可以使用：

- IIC 移位寄存器 0 至 2 (IIC0 至 IIC2)
- 从设备地址寄存器 0 至 2 (SVA0 至 SVA2)

备注 关于复用功能引脚的设置，参见表 4-19 端口引脚用作复用功能引脚。

(1) IIC 控制寄存器 0 至 2 (IICC0 至 IICC2)

IIC_{Cn} 寄存器允许/停止 I²C_{0n} 操作，设置等待时间并设置其它 I²C 操作 (n = 0 至 2)。

这些寄存器可以按字节或按位进行读取和写入。但是，当 IIC_{En} 位为 0 时或在等待期间，对 SPI_{En}、WTIM_n 和 ACK_{En} 位置位。当 IIC_{En} 位由“0”设置为“1”时，这些位也可以被同时置位。

系统复位后，这些寄存器被置为 00H。

复位后: 00H R/W 地址: IICC0 FFFFFFFD82H, IICC1 FFFFFFFD92H, IICC2 FFFFFFFDA2H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICn	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

(n = 0 至 2)

IICEn	I ² Cn操作允许/禁止的说明
0	操作停止。 IICSn寄存器复位 ^{#1} 。内部操作停止。
1	允许操作。
SCL0n 和 SDA0n 信号线是高电平时, 请确保将该位设置为 1。	
清零条件 (IICEn 位 = 0)	
<ul style="list-style-type: none"> 由指令清除。 复位后。 	
置位条件 (IICEn 位 = 1)	
<ul style="list-style-type: none"> 由指令设置。 	

LRELn ^{#2}	退出通信
0	正常操作
1	从当前通信操作退出, 并设置为待机模式。在执行之后该设置被自动清除。当接收到本地无关扩展码的情况下也可以使用。 SCL0n 和 SDA0n 信号线被设为高阻状态。 STTn 和 SPTn 位以及 IICSn 寄存器的 MSTSn, EXCn, COIn, TRCn, ACKDn 和 STDn 位被清零。
从通信中退出之后的待机状态一直保持有效, 直到遇到下面的通信入口条件。	
<ul style="list-style-type: none"> 检测到停止条件后, 在主模式下重新开始。 在开始条件以后, 发生地址匹配或接收到扩展码。 	
清零条件 (LRELn 位 = 0)	
<ul style="list-style-type: none"> 执行后自动清除。 复位后。 	
置位条件 (LRELn 位 = 1)	
<ul style="list-style-type: none"> 由指令设置。 	

WRELn ^{#2}	等待状态取消控制
0	等待状态不能取消。
1	等待状态被取消。等待取消以后, 该设置被自动清除。
清零条件 (WRELn 位 = 0)	
<ul style="list-style-type: none"> 执行后自动清除。 复位后。 	
置位条件 (WRELn 位 = 1)	
<ul style="list-style-type: none"> 由指令设置。 	

注 1. IICSn 寄存器、IICFn.STCFn 和 IICFn.IICBSYn 位、IICCLn.CLDn 位和 IICCLn.DADn 位被重置。
2. 当 IICEn 位 = 0 时, 该标志信号无效。

注意事项当 SCL0n 信号线为高电平且 SDA0n 信号线为低电平时, 如果 I²Cn 操作使能 (IICEn 位 = 1), 则立即检测到开始条件。为了避免这种情况, I²Cn 操作使能以后, 立刻使用位处理指令将 LRELn 位设置为 1。

备注 数据被设置后进行读取时, LRELn 和 WRELn 位为 0。

SPIEn ^注	当检测到停止条件时，使能/禁止产生中断请求	
0	禁止	
1	使能	
清零条件（SPIEn 位 = 0）		置位条件（SPIEn 位 = 1）
<ul style="list-style-type: none"> 由指令清除。 复位后。 		<ul style="list-style-type: none"> 由指令设置。

WTIMn ^注	等待状态和产生中断请求的控制	
0	在第 8 个时钟的下降沿产生中断请求信号。 主模式：在输出 8 个时钟以后，时钟输出被置为低电平，并且设置等待状态。 从模式：在输入 8 个时钟以后，时钟被设置为低电平，并为主设备设置等待状态。	
1	在第 9 个时钟信号的下降沿产生中断请求信号。 主模式：在输出 9 个时钟以后，时钟输出被设置为低电平，并且设置等待状态。 从模式：在输入 9 个时钟以后，时钟被设置为低电平，并为主设备设置等待状态。	
地址传输期间，不论该位如何设置，在第 9 个时钟下降沿处发生中断。地址传输完成后该位的设置生效。主模式下，在地址传输期间的第 9 个时钟下降沿处，插入一个等待状态。对于接收到本地地址的从设备，产生 ACK 信号之后的第 9 个时钟的下降沿处插入一个等待状态。但是，当从设备收到扩展码后，在第 8 个时钟的下降沿插入等待状态。		
清零条件（WTIMn 位 = 0）		置位条件（WTIMn 位 = 1）
• 由指令清除。 • 复位后。		• 由指令设置。

ACKEn ^注	应答控制
0	禁止应答。
1	允许应答。在第 9 个时钟期间，SDA0n 信号线设置为低电平。_____
ACKEn 位的设置对从设备的地址接收是无效的，这种情况下，地址匹配时产生 ACK 信号。 但是，在接收扩展码地址时，ACKEn 位的设置有效。在接收到扩展码的系统中设置 ACKEn 位。	
清零条件（ACKEn 位 = 0）	置位条件（ACKEn 位 = 1）
<ul style="list-style-type: none">• 由指令清除。• 复位后。	<ul style="list-style-type: none">• 由指令设置。

注 当 IICEn 位 = 0 时，该标志信号无效。

备注 n = 0 至 2

STTn	开始条件触发
0	不产生开始条件。
1	<p>当总线被释放（在 STOP 模式下）： 产生一个开始条件（作为主设备启动）。当 SCL0n 信号线为高电平时，SDA0n 信号线由高电平变为低电平，这样就可以产生开始条件。经过一段合理的时间之后，SCL0n 信号线变为低电平。</p> <p>与第三方通信期间： 如果使能通信保留功能（IICFn.IICRSVn 位 = 0）： <ul style="list-style-type: none"> 该触发信号作为开始条件保留标志。当设置为 1 时，释放总线然后自动产生一个开始条件。 如果禁止通信保留功能（IICRSVn = 1）： <ul style="list-style-type: none"> IICFn.STCFn 位设置为 1 以清除 STTn 位的设置信息（1），该触发信号不能产生开始条件。 处于等待状态（作为主设备）： 等待状态释放后，产生重新开始条件。</p>
<p>设置时序的有关注意事项：</p> <p>主设备接收： 传输期间不能设置为 1，只能在 ACKEn 位已经设置 0，且从设备已经得知最末接收时，才可以置为 1。</p> <p>主设备发送： 在 ACK 期间，通常不会产生开始条件。在第 9 个时钟输出后的等待期间置为 1。</p> <p>从设备： 即使禁止通信保留功能（IICRSVn 位 = 1），也可以进入通信保留状态。</p> <ul style="list-style-type: none"> 禁止 SPTn 位的同时设置为 1。 当 STTn 位设置为 1 时，禁止再次将 STTn 位置 1，直到设置被清 0 为止。 	
清零条件（STTn 位 = 0）	置位条件（STTn 位 = 1）
<ul style="list-style-type: none"> 在通信保留禁止状态，当 STTn 位设置为 1 时。 仲裁失败后清除。 主设备产生开始条件后清除。 当 LRELn 位 = 1 时（通信保存）。 当 IICEn 位 = 0（操作停止）时。 复位后 	<ul style="list-style-type: none"> 由指令设置。

备注 1. 如果在数据设置后立即读取 STTn 位，其值为 0。
 2. n = 0 至 2

SPTn	停止条件触发
0	不产生停止条件。
1	产生停止条件（主设备传输的终结）。 SDA0 信号线变为低电平之后，将 SCL0n 信号线设置为高电平，或者一直等待直到 SCL0n 引脚变为高电平。然后经过一段合理的时间后，SDA0n 信号线由低电平变为高电平，产生停止条件。
设置时序的有关注意事项 主设备接收：传输期间不能被设置为 1，只能在 ACKEn 位已经设置为 0，且从设备已经得知最末接收后的等待期间，才可以被置为 1。 主设备发送：在 ACK 接收期间，通常不会产生停止条件。在第 9 个时钟输出后的等待期间置为 1。 <ul style="list-style-type: none"> 不能和 STTn 位同时置 1。 只有在主模式下，SPTn 位才可以设置为 1。 在 WTIMn 位已经设置 0 时，如果在 8 个时钟输出后的等待期间将 SPTn 位设置为 1，注意，将在第 9 个时钟的高电平期间产生停止条件。 在 8 个时钟输出后的等待期间，WTIMn 位应该从 0 变为 1，且在第 9 个时钟输出后的等待期间应该将 SPTn 位设置为 1。 当 SPTn 位设置为 1 时，禁止再次将 SPTn 位置为 1，直到设置被清除为 0。 	
清零条件（SPTn 位 = 0）	置位条件（SPTn 位 = 1）
<ul style="list-style-type: none"> 仲裁失败后清除。 检测到停止条件后，自动清除。 当 LRELn 位 = 1 时（通信保存）。 IICEn 位 = 0（操作停止）时。 复位后。 	<ul style="list-style-type: none"> 由指令设置。

注 只有在主设备模式下，才可以将 SPTn 位设置为 1。但是，当 IICRSVn 位为 0 时，改变操作使能状态后检测到第一个停止条件前，SPTn 位必须设置为 1 并产生一个停止条件。详情参见 17.15 注意事项。

注意事项 当 TRCn 位 = 1 时，在第 9 个时钟期间 WRELn 位被置 1 且等待状态被取消，之后 TRCn 位被清除为 0 且 SDA0n 信号线被置为高阻状态。

备注

- 如果在数据设置以后立即读取 SPTn 位，其值为 0。
- n = 0 至 2

(2) IIC 状态寄存器 0 至 2 (IICS0 至 IICS2)

IICS_n寄存器表明I²C0_n的状态 (n = 0 至 2)。

这些寄存器只能按字节或按位进行读取。但是, 只有当 IICCN.STTN 位为 1 时或在等待期间才能读取 IICS_n 寄存器。

系统复位后, 这些寄存器被置为 00H。

注意事项 在以下状态时, 禁止访问 IICS_n 寄存器。详情参见 3.4.8 (2) 访问专用片上外设 I/O 寄存器。

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

(1/3)

复位后: 00H

R

地址: IICS0 FFFFFFFD86H, IICS1 FFFFFFFD96H, IICS2 FFFFFFFDA6H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS _n	MSTS _n	ALD _n	EXC _n	COL _n	TRC _n	ACKD _n	STD _n	SPD _n

(n = 0 至 2)

MSTS _n	主设备状态
0	从设备状态或通信待机状态
1	主设备通信状态
清零条件 (MSTS _n 位 = 0)	
<ul style="list-style-type: none"> • 当检测到停止条件时。 • 当 ALD_n 位 = 1 时 (仲裁失败)。 • 由 LREL_n 位 = 1 清除 (通信保存)。 • 当 IICEn 位由 1 变为 0 (工作停止) 时。 • 复位后。 	
置位条件 (MSTS _n 位 = 1)	
<ul style="list-style-type: none"> • 当产生开始条件时。 	

ALD _n	仲裁失败检测
0	该状态意味着不存在总线仲裁, 或仲裁结果为“成功”。
1	该状态表示仲裁结果为“失败”。MSTS _n 位被清除为 0。
清零条件 (ALD _n 位 = 0)	
<ul style="list-style-type: none"> • IICS_n 寄存器被读取^注后, 自动清除。 • 当 IICEn 位由 1 变为 0 (工作停止) 时。 • 复位后。 	
置位条件 (ALD _n 位 = 1)	
<ul style="list-style-type: none"> • 当总线仲裁结果为“失败”。 	

EXC _n	扩展码接收的检测
0	未接收到扩展码。
1	接收到扩展码。
清零条件 (EXC _n 位 = 0)	
<ul style="list-style-type: none"> • 当检测到开始条件时。 • 当检测到停止条件时。 • 通过 LREL_n 位 = 1 清除 (通信保存)。 • 当 IICEn 位由 1 变为 0 时 (操作停止)。 • 复位后。 	
置位条件 (EXC _n 位 = 1)	
<ul style="list-style-type: none"> • 当接收地址数据的高 4 位是“0000”或“1111”时 (在第 8 个时钟的上升沿置位)。 	

注 对 IICS_n 寄存器的其它位进行位操作时, ALD_n 位也被清除。

COIn	匹配地址检测	
0	地址不匹配	
1	地址匹配	
清零条件 (COIn 位 = 0)		置位条件 (COIn 位 = 1)
<ul style="list-style-type: none"> 当检测到开始条件时。 当检测到停止条件时。 通过 LRELn 位 = 1 清除 (通信保存)。 当 IICEn 位由 1 变为 0 (工作停止) 时。 复位后。 		<ul style="list-style-type: none"> 当接收到的地址和本地地址 (SVAn 寄存器) 匹配时 (在第 8 个时钟的上升沿置位)。

TRCn	发送/接收状态检测	
0	接收状态 (除发送状态外)。SDA0n 信号线被置为高阻态。	
1	发送状态。允许 SO 锁存器内的数值从 SDA0n 信号线输出 (在第一个字节的第 9 个时钟下降沿开始生效)。	
清零条件 (TRCn 位 = 0)		置位条件 (TRCn 位 = 1)
<ul style="list-style-type: none"> 当检测到停止条件时。 通过 LRELn 位 = 1 清除 (通信保存)。 当 IICEn 位由 1 变为 0 (工作停止) 时。 通过 IICn.WRELn 位 = 1[※]清除。 当 ALDn 位从 0 变为 1 (仲裁失败) 时。 复位后。 主设备： <ul style="list-style-type: none"> 当第一个字节的 LSB (传输方向指示位) 输出为“1”时。 从设备： <ul style="list-style-type: none"> 当检测到开始条件时。 当不用于通信时。		主设备： <ul style="list-style-type: none"> 当产生开始条件时。 当第一个字节的 LSB (传输方向指示位) 输出为“0”时。 从设备： <ul style="list-style-type: none"> 当第一个字节的 LSB (传输方向指示位) 输入为“1”时。

ACKDn	ACK 检测	
0	未检测到 ACK。	
1	检测到 ACK。	
清零条件 (ACKDn 位 = 0)		置位条件 (ACKDn 位 = 1)
<ul style="list-style-type: none"> 当检测到停止条件时。 在下一字节第一个时钟的上升沿。 通过 LRELn 位 = 1 清除 (通信从模式)。 当 IICEn 位由 1 变为 0 (工作停止) 时。 复位后。 		<ul style="list-style-type: none"> 在 SCL0n 引脚的第 9 个时钟的上升沿 SDA0n 引脚被设置为低电平后。

注 当 WRELn 位设置为 1，且在第九个时钟通过 TRCn 位 = 1 取消等待状态时，TRCn 被清除为 0，而且 SDA0n 信号线变为高阻态。

备注 n = 0 至 2

STDn	开始条件检测	
0	未检测到开始条件。	
1	检测到开始条件。这表示在地址传输时期有效时段内。	
清零条件（STDn 位 = 0）		置位条件（STDn 位 = 1）
<ul style="list-style-type: none">• 当检测到停止条件时。• 在地址传输后下一个字节第一个时钟的上升沿。• 通过 LRELn 位 = 1 清除（通信保存）。• 当 IICEn 位 由 1 变为 0（工作停止）时。• 复位后。		<ul style="list-style-type: none">• 当检测到开始条件时。

SPDn	停止条件检测	
0	未检测到停止条件。	
1	检测到停止条件。主设备的通信被终结，并且释放总线。	
清零条件（SPDn 位 = 0）		置位条件（SPDn 位 = 1）
<ul style="list-style-type: none">• 在位置该位之后的地址传输第一个时钟上升沿处，以及检测到开始条件时。• 当 IICEn 位 由 1 变为 0（工作停止）时。• 复位后。		<ul style="list-style-type: none">• 当检测到停止条件时。

备注 n = 0 至 2

(3) IIC 标志寄存器 0 至 2 (IICF0 至 IICF2)

IICFn 寄存器设置 I²C0n 操作模式并指示 I²C 总线状态。

这些寄存器可以按字节或按位进行读取和写入。但是，STCFn 和 IICBSYn 位只读。

IICRSVn 使能/禁止通信保留功能（参见 **17.14 通信保留**）。

IICBSYn 位的初始值通过 STCENn 位设置（参见 **17.15 注意事项**）。

IICRSVn 和 STCENn 位只能在 I²C0n 操作禁止（IICCN.IICEn 位 = 0）时写入。操作使能后，IICFn 可被读取（n = 0 至 2）。

系统复位后，这些寄存器被置为 00H。

复位后: 00H R/W^注 地址: IICF0 FFFFFFFD8AH, IICF1 FFFFFFFD9AH, IICF2 FFFFFFFDAAH

	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

(n = 0 至 2)

STCFn	STTn 位清除
0	发出开始条件。
1	不发出开始条件, STTn 位被清除。
清零条件 (STCFn 位 = 0)	
<ul style="list-style-type: none"> 通过 IICFn.STTn 位 = 1 清除。 当 IICFn.IICEn 位 = 0 时。 复位后。 	
置位条件 (STCFn 位 = 1)	
<ul style="list-style-type: none"> 禁止通信保留 (IICRSVn 位 = 1) 期间, 不发出开始条件且 STTn 标志被清除为 0 时。 	

IICBSYn	I ² C0n 总线状态
0	总线释放状态 (STCENn 位 = 1 时的默认通信状态)。
1	总线通信状态 (STCENn 位 = 0 时的默认通信状态)。
清零条件 (IICBSYn 位 = 0)	
<ul style="list-style-type: none"> 当检测到停止条件时。 IICEn 位 = 0 时。 复位后。 	
置位条件 (IICBSYn 位 = 1)	
<ul style="list-style-type: none"> 当检测到开始条件时。 当 STCENn 位 = 0 时, 通过设置 IICEn 位来置位。 	

STCENn	初始化开始使能触发
0	不能产生开始条件, 直到操作使能后 (IICEn 位 = 1) 检测到停止条件为止。
1	产生开始条件, 即使操作使能后 (IICEn 位 = 1) 没有检测到停止条件。
清零条件 (STCENn 位 = 0)	
<ul style="list-style-type: none"> 当检测到开始条件时。 复位后。 	
置位条件 (STCENn 位 = 1)	
<ul style="list-style-type: none"> 由指令设置。 	

IICRSVn	通信保留功能禁止位
0	使能通信保留
1	禁止通信保留
清零条件 (IICRSVn 位 = 0)	
<ul style="list-style-type: none"> 由指令清除。 复位后。 	
置位条件 (IICRSVn 位 = 1)	
<ul style="list-style-type: none"> 由指令设置。 	

注 第 6 位和第 7 位为只读位。

- 注意事项**
1. 仅当操作停止时 (IICEn 位 = 0), 才可以写入 STCENn 位。
 2. 当 STCENn 位 = 1 时, 紧随在 I²Cn 总线操作使能之后, 不管总线的实际状态如何, 总是认为处于总线释放状态 (IICFn.IICBSYn 位 = 0)。因此, 要产生首个开始条件 (STTn 位 = 1), 必需确认总线已经被释放, 这样不会干扰其它通信。
 3. 仅当操作停止时 (IICEn 位 = 0), 才可以写入 IICRSVn 位。

(4) IIC 时钟选择寄存器 0 至 2 (IICCL0 至 IICCL2)

IICCLn 寄存器为 I²C0n 设置传输时钟。

这些寄存器可以按字节或按位进行读取或写入。但是，CLDn 和 DADn 位是只读位。

当 IICn.IICEn 位 = 0 时，设置 IICCLn 寄存器。

组合使用 SMCn, CLn1 和 CLn0 位以及 IICXn.CLXn 位与 OCKSm 寄存器的 OCKSTHm, OCKSm1 和 OCKSm0 位进行设置（参见 17.4 (6) I²C0n 传输时钟设置方式）（n = 0 至 2, m = 0, 1）。

系统复位后，这些寄存器被置为 00H。

复位后：00H

R/W^{*}

地址：IICCL0 FFFFFFFD84H, IICCL1 FFFFFFFD94H, IICCL2 FFFFFFFDA4H

	7	6	<5>	<4>	3	2	1	0
IICCLn	0	0	CLDn	DADn	SMCn	DFCn	CLn1	CLn0
(n = 0 至 2)								
CLDn	SCL0n 引脚电平的检测（仅当 IICEn.IICEn 位 = 1 时有效）							
0	SCL0n 引脚检测为低电平。							
1	SCL0n 引脚检测为高电平。							
清零条件（CLDn 位 = 0）					置位条件（CLDn 位 = 1）			
<ul style="list-style-type: none">当 SCL0n 引脚为低电平时。IICEn 位 = 0（操作停止）时。复位后。					<ul style="list-style-type: none">当 SCL0n 引脚为高电平时。			

DADn	SDA0n 引脚电平的检测（仅当 IICEn 位 = 1 时有效）							
0	SDA0n 引脚检测为低电平。							
1	SDA0n 引脚检测为高电平。							
清零条件（DADn 位 = 0）					置位条件（DADn 位 = 1）			
<ul style="list-style-type: none">当 SDA0n 引脚为低电平时。IICEn 位 = 0（操作停止）时。复位后。					<ul style="list-style-type: none">当 SDA0n 引脚为高电平时。			

SMCn	工作模式切换							
0	工作于标准模式。							
1	工作于高速模式。							

DFCn	数字滤波器操作控制							
0	数字滤波器关闭。							
1	数字滤波器开启。							
数字滤波器只能在高速模式下使用。								
在高速模式下，不论 DFCn 位如何设置（开启/关闭），传输时钟都不会改变。								
数字滤波器用于消除高速模式下的噪声。								

注 第 4 位和第 5 位是只读位。

注意事项 请确保将第 7 位和第 6 位清除为“0”。

备注 当 IICn.IICEn 位 = 0 时，读取 CLDn 和 DADn 位，其值为 0。

(5) IIC 功能扩展寄存器 0 至 2 (IICX0 至 IICX2)

IICXn 寄存器设置 I²C0n 功能扩展（仅在高速模式下有效）。

这些寄存器可以按字节或按位进行读取或写入。

组合使用 IICCLn 寄存器的 SMCn, CLn1 和 CLn0 位以及 OCKSm 寄存器的 OCKSTHm, OCKSm1 和 OCKSm0 位, 对 CLXn 位进行设置（参见 17.4 (6) I²C0n 传输时钟设置方式）（m = 0, 1）。

当 IICCN.IICEn 位 = 0 时, 设置 IICXn 寄存器。

系统复位后, 这些寄存器被置为 00H。

复位后: 00H R/W 地址: IICX0 FFFFFFFD85H, IICX1 FFFFFFFD95H, IICX2 FFFFFFFDA5H

	7	6	5	4	3	2	1	<0>
IICXn	0	0	0	0	0	0	0	CLXn

(n = 0 至 2)

(6) I²C0n 传输时钟设置方式

使用下列表达式计算 I²C0n 传输时钟频率 (f_{SCL}) (n = 0 至 2)。

$$f_{SCL} = 1 / (m \times T + t_R + t_F)$$

m = 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)。

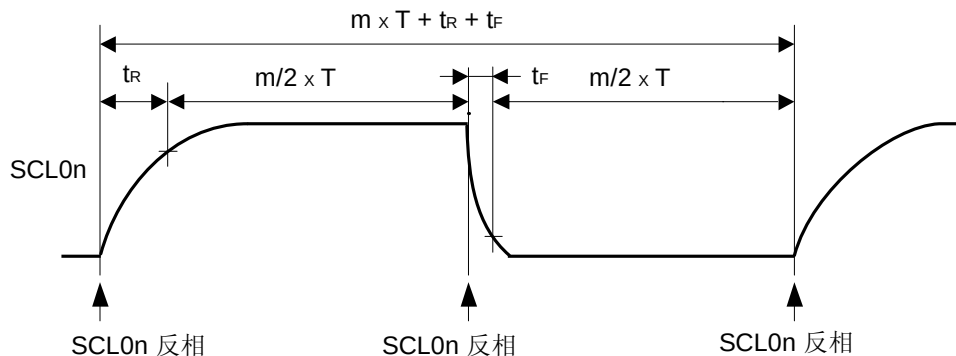
T: 1/f_{xx}

t_R: SCL0n 引脚上升时间

t_F: SCL0n 引脚下降时间

例如, 当 f_{xx} = 19.2 MHz, m = 198, t_R = 200 ns 且 t_F = 50 ns 时, I²C0n 传输时钟频率 (f_{SCL}) 使用下面表达式计算。

$$f_{SCL} = 1 / (198 \times 52 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 94.7 \text{ kHz}$$



组合使用 IICCLn 寄存器的 SMCn, CLn1 和 CLn0 位以及 IICXn 寄存器的 CLXn 位、OCKSm 寄存器的 OCKSTHm, OCKSm1 和 OCKSm0 位来设置选择时钟 (n = 0 至 2, m = 0, 1)。

表 17-2. 时钟设置 (1/2)

IICX0	IICCL0			选择时钟	传输时钟	可设置的主时钟 频率 (f _{xx}) 范围	操作模式
第 0 位	第 3 位	第 1 位	第 0 位				
CLX0	SMC0	CL01	CL00				
0	0	0	0	f _{xx} (当 OCKS0 = 18H 时)	f _{xx} /44	2.00 MHz ≤ f _{xx} ≤ 4.19 MHz	标准模式 (SMC0 位 = 0)
				f _{xx} /2 (当 OCKS0 = 10H 时)	f _{xx} /88	4.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
				f _{xx} /3 (当 OCKS0 = 11H 时)	f _{xx} /132	6.00 MHz ≤ f _{xx} ≤ 12.57 MHz	
				f _{xx} /4 (当 OCKS0 = 12H 时)	f _{xx} /176	8.00 MHz ≤ f _{xx} ≤ 16.76 MHz	
				f _{xx} /5 (当 OCKS0 = 13H 时)	f _{xx} /220	10.00 MHz ≤ f _{xx} ≤ 20.95 MHz	
0	0	0	1	f _{xx} (当 OCKS0 = 18H 时)	f _{xx} /86	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	
				f _{xx} /2 (当 OCKS0 = 10H 时)	f _{xx} /172	8.38 MHz ≤ f _{xx} ≤ 16.76 MHz	
				f _{xx} /3 (当 OCKS0 = 11H 时)	f _{xx} /258	12.57 MHz ≤ f _{xx} ≤ 25.14 MHz	
				f _{xx} /4 (当 OCKS0 = 12H 时)	f _{xx} /344	16.76 MHz ≤ f _{xx} ≤ 32.00 MHz	
				f _{xx} /5 (当 OCKS0 = 13H 时)	f _{xx} /430	20.95 MHz ≤ f _{xx} ≤ 32.00 MHz	
0	0	1	0	f _{xx} ^注	f _{xx} /86	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	高速模式 (SMC0 位 = 1)
0	0	1	1	f _{xx} (当 OCKS0 = 18H 时)	f _{xx} /66	6.40 MHz	
				f _{xx} /2 (当 OCKS0 = 10H 时)	f _{xx} /132	12.80 MHz	
				f _{xx} /3 (当 OCKS0 = 11H 时)	f _{xx} /198	19.20 MHz	
				f _{xx} /4 (当 OCKS0 = 12H 时)	f _{xx} /264	25.60 MHz	
				f _{xx} /5 (当 OCKS0 = 13H 时)	f _{xx} /330	32.00 MHz	
0	1	0	×	f _{xx} (当 OCKS0 = 18H 时)	f _{xx} /24	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	
				f _{xx} /2 (当 OCKS0 = 10H 时)	f _{xx} /48	8.00 MHz ≤ f _{xx} ≤ 16.76 MHz	
				f _{xx} /3 (当 OCKS0 = 11H 时)	f _{xx} /72	12.00 MHz ≤ f _{xx} ≤ 25.14 MHz	
				f _{xx} /4 (当 OCKS0 = 12H 时)	f _{xx} /96	16.00 MHz ≤ f _{xx} ≤ 32.00 MHz	
0	1	1	0	f _{xx} ^注	f _{xx} /24	4.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
0	1	1	1	f _{xx} (当 OCKS0 = 18H 时)	f _{xx} /18	6.40 MHz	
				f _{xx} /2 (当 OCKS0 = 10H 时)	f _{xx} /36	12.80 MHz	
				f _{xx} /3 (当 OCKS0 = 11H 时)	f _{xx} /54	19.20 MHz	
				f _{xx} /4 (当 OCKS0 = 12H 时)	f _{xx} /72	25.60 MHz	
				f _{xx} /5 (当 OCKS0 = 13H 时)	f _{xx} /90	32.00 MHz	
1	1	0	×	f _{xx} (当 OCKS0 = 18H 时)	f _{xx} /12	4.00 MHz ≤ f _{xx} ≤ 4.19 MHz	
				f _{xx} /2 (当 OCKS0 = 10H 时)	f _{xx} /24	8.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
				f _{xx} /3 (当 OCKS0 = 11H 时)	f _{xx} /36	12.00 MHz ≤ f _{xx} ≤ 12.57 MHz	
				f _{xx} /4 (当 OCKS0 = 12H 时)	f _{xx} /48	16.00 MHz ≤ f _{xx} ≤ 16.67 MHz	
				f _{xx} /5 (当 OCKS0 = 13H 时)	f _{xx} /60	20.00 MHz ≤ f _{xx} ≤ 20.95 MHz	
1	1	1	0	f _{xx} ^注	f _{xx} /12	4.00 MHz ≤ f _{xx} ≤ 4.19 MHz	
其它情况				禁止设置	—	—	—

注 由于选择时钟是f_{xx}，与OCKS0寄存器的设定值无关，所以将OCKS0寄存器清除为00H（I²C分频时钟停止状态）。

备注 ×：无需理会

表 17-2. 时钟设置 (2/2)

IICXm	IICCLm			选择时钟	传输时钟	可设置的主时钟 频率 (f _{xx}) 范围	操作模式
第 0 位	第 3 位	第 1 位	第 0 位				
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 位 = 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} ≤ 20.95 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} ≤ 25.14 MHz	
				f _{xx} /4 (当 OCKS1 = 12H 时)	f _{xx} /344	16.76 MHz ≤ f _{xx} ≤ 32.00 MHz	
				f _{xx} /5 (当 OCKS1 = 13H 时)	f _{xx} /430	20.95 MHz ≤ f _{xx} ≤ 32.00 MHz	
0	0	1	0	f _{xx} [※]	f _{xx} /86	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	高速模式 (SMCm 位 = 1)
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	
				f _{xx} /4 (当 OCKS1 = 12H 时)	f _{xx} /264	25.60 MHz	
				f _{xx} /5 (当 OCKS1 = 13H 时)	f _{xx} /330	32.00 MHz	
0	1	0	×	f _{xx} (当 OCKS1 = 18H 时)	f _{xx} /24	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	
				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} ≤ 25.14 MHz	
				f _{xx} /4 (当 OCKS1 = 12H 时)	f _{xx} /96	16.00 MHz ≤ f _{xx} ≤ 32.00 MHz	
0	1	1	0	f _{xx} [※]	f _{xx} /24	4.00 MHz ≤ f _{xx} ≤ 8.38 MHz	高速模式 (SMCm 位 = 1)
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	
				f _{xx} /4 (当 OCKS1 = 12H 时)	f _{xx} /72	25.60 MHz	
				f _{xx} /5 (当 OCKS1 = 13H 时)	f _{xx} /90	32.00 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 ≤ f _{xx} ≤ 20.95 MHz	
1	1	1	0	f _{xx} [※]	f _{xx} /12	4.00 MHz ≤ f _{xx} ≤ 4.19 MHz	
其它情况				禁止设置	—	—	—

注 由于选择时钟是f_{xx}，与OCKS1 寄存器设定值无关，所以将OCKS1 寄存器清除为 00H (I²C分频时钟停止状态)。

备注 1. m = 1, 2
2. ×: 无需理会

(7) IIC 分频时钟选择寄存器 0, 1 (OCS0, OCS1)

OCSm寄存器控制I²C0n分频时钟 (n = 0 至 2, m = 0, 1)

该寄存器通过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。

该寄存器可以按字节进行读取或写入。

系统复位后, 该寄存器被置为 00H。

复位后: 00H R/W 地址: OCS0 FFFFF340H, OCS1 FFFFF344H

	7	6	5	4	3	2	1	0
OCSm	0	0	0	OCSm	OCSm	0	OCSm1	OCSm0

(m = 0, 1)

OCSm	I ² C 分频时钟的操作设置
0	禁止 I ² C 分频时钟工作
1	使能 I ² C 分频时钟工作

OCSm	OCSm1	OCSm0	I ² C 分频时钟的选择
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}
其它			禁止设置

(8) IIC 移位寄存器 0 至 2 (IIC0 至 IIC2)

IICn 移位寄存器和串行时钟同步, 用于串行发送/接收 (移位操作)。该移位寄存器可以按字节读取或写入, 但是, 在数据传输期间, 不应该向 IICn 移位寄存器写入数据。

只有在等待期间, 可以访问 (读/写) IICn 移位寄存器。在等待期间以外的其它通信状态时, 禁止访问该寄存器。

但是, 对于主设备, 只有在传输触发位 (IICn.STTn 位) 被置为 1 后, IICn 移位寄存器才可以写入一次。

在等待期间写入 IICn 寄存器, 释放一个等待状态, 且启动数据传输 (n = 0 至 2)。

系统复位后, 该寄存器被置为 00H。

复位后: 00H R/W 地址: IIC0 FFFFFD80H, IIC1 FFFFFD90H, IIC2 FFFFFDA0H

	7	6	5	4	3	2	1	0
IICn								

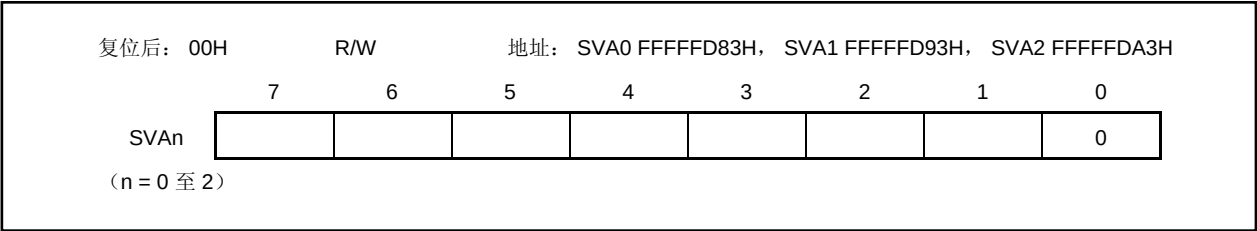
(n = 0 至 2)

(9) 从设备地址寄存器 0 至 2 (SVA0 至 SVA2)

SVAn 寄存器保持I²C 总线的从地址 (n = 0 至 2)。

该移位寄存器可以按字节读取或写入，第 0 位固定为 0。但是当 IICSn.STDn 位 = 1 (开始条件检测) 时，禁止重写该寄存器。

系统复位后，该寄存器被置为 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 至 2）。

SCL0n该引脚用于串行时钟输入和输出。

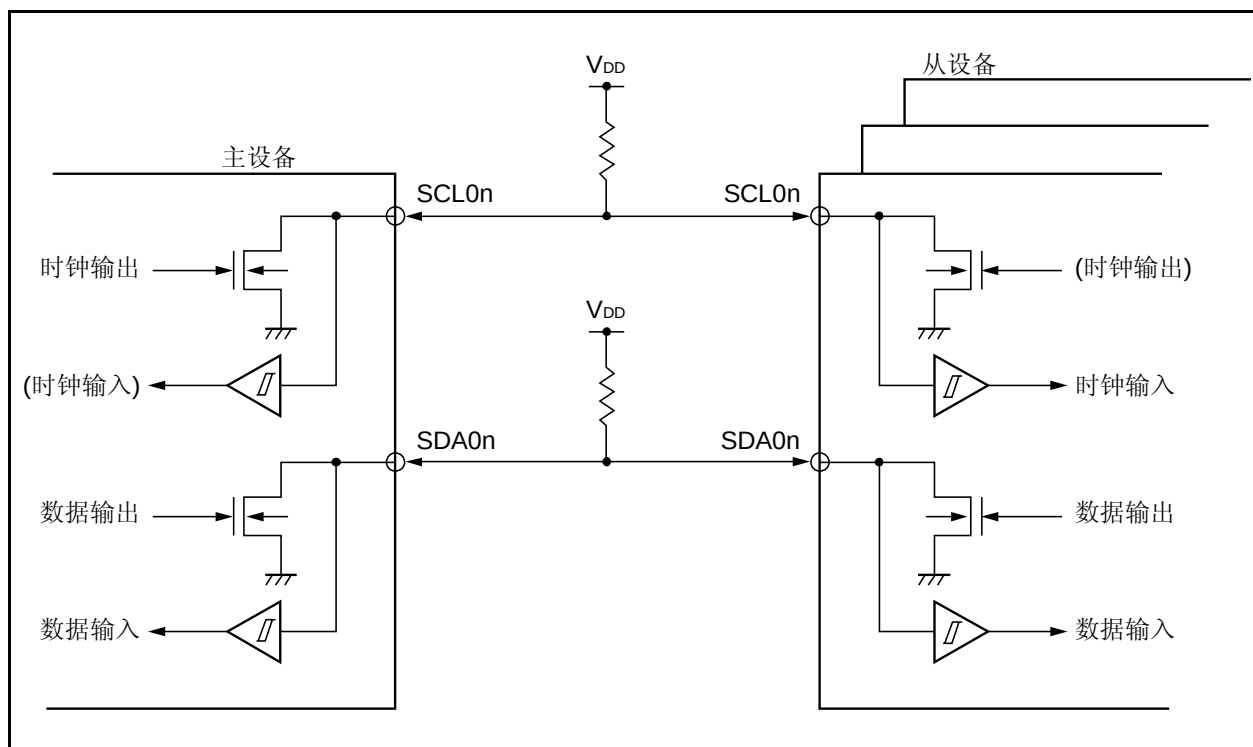
对于主设备和从设备，该引脚都是 N 沟道漏极开路输出，输入为施密特输入。

SDA0n该引脚用于串行数据输入和输出。

对于主设备和从设备，该引脚都是 N 沟道漏极开路输出，输入为施密特输入。

由于串行时钟信号线和串行数据总线输出都是 N 沟道漏极开路输出，所以都需要外接上拉电阻。

图 17-6. 引脚配置图

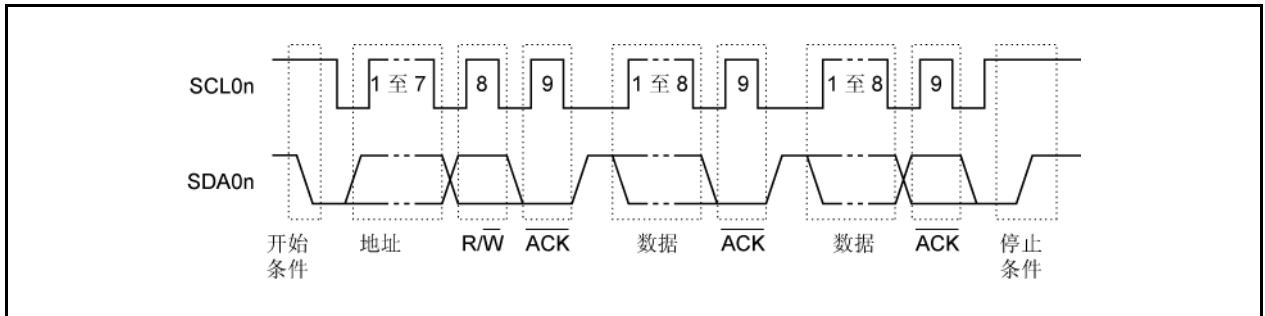


17.6 I²C总线的定义和控制方法

以下章节描述I²C总线的串行数据通信格式和I²C总线使用的信号。

关于I²C总线的串行数据总线产生的“开始条件”、“地址”、“传输方向”、“数据”以及“停止条件”的传输时序，如下所示。

图 17-7. I²C 总线串行数据传输时序



主设备产生开始条件、从设备地址和停止条件。

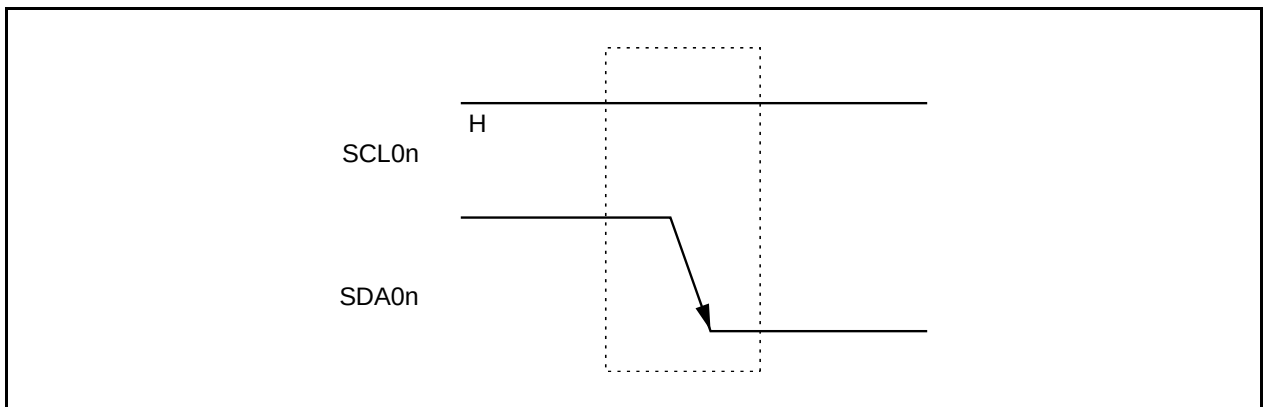
ACK 信号既可由主设备产生，也可由从设备产生。（通常情况下，它由接收 8 位数据的设备产生）。

串行时钟（SCL0n）由主设备连续输出。但是，在从设备中 SCL0n 引脚的低电平时段可以被延长，可以插入一个等待状态（n = 0 至 2）。

17.6.1 开始条件

当 SCL0n 引脚为高电平而且 SDA0n 引脚由高电平变为低电平时，形成一个开始条件。SCL0n 和 SDA0n 引脚的开始条件是启动串行传输时主设备输出至从设备的信号。从设备可以检测开始条件（n = 0 至 2）。

图 17-8. 开始条件



当检测到停止条件（IICSn.SPDn 位 = 1）后，当 IICC0.STT0 位被置位（1）时，输出一个开始条件。当检测到开始条件时，IICSn.STDn 位被置位（1）。（n = 0 至 2）

注意事项 在和其它设备进行通信时，当 V850ES/JJ3 的 IICn.IICEn 位被设置为 1 时，根据通信线的状态，可能会检测到开始条件。SCL0n 和 SDA0n 信号线是高电平时，请确保将 IICn.IICEn 位设置为 1。

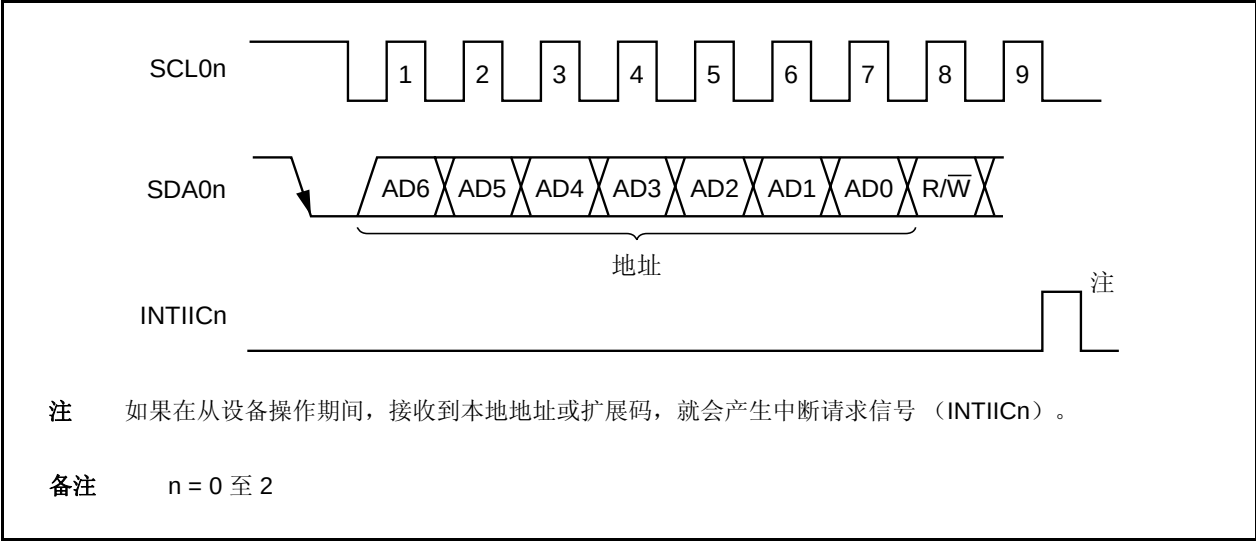
17.6.2 地址

紧随开始条件之后的 7 位数据被定义为地址。

地址是 7 位数据段，输出地址用来选择通过总线连接到主设备的其中一个从设备。所以，每一个接入总线的从设备必须具有一个独一无二的地址。

从设备包含检测开始条件的硬件，并检查 7 位地址数据是否和 SVAn 寄存器内储存的数值匹配。如果地址数据和 SVAn 寄存器内储存的数据值相匹配，这个从设备就被选择，并且和主设备进行通信，直到主设备产生一个开始条件或结束条件。（n = 0 至 2）

图 17-9. 地址



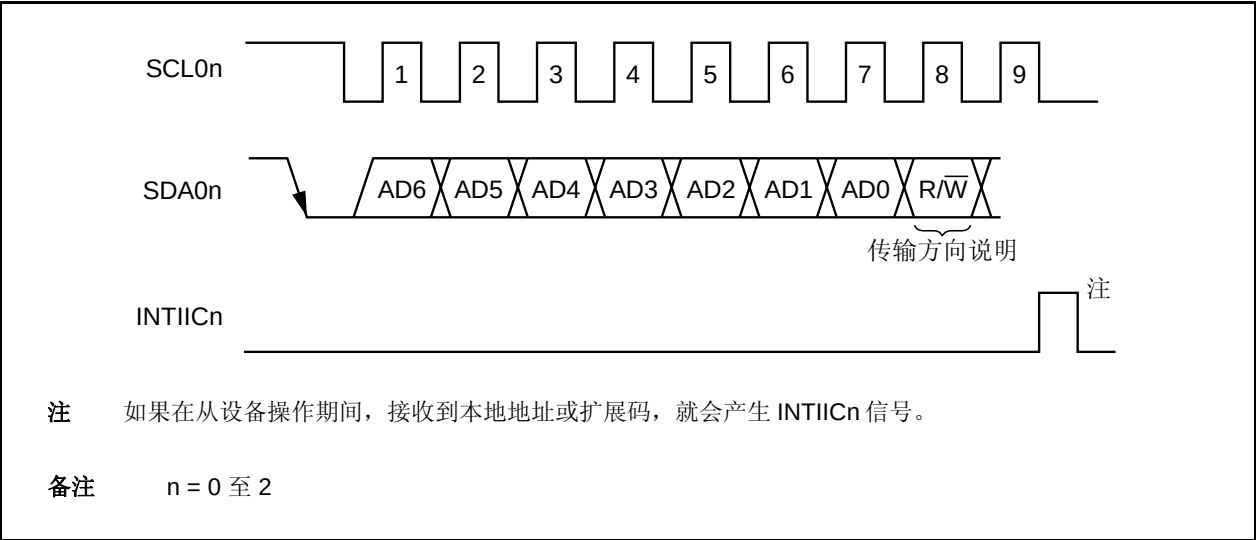
从设备地址和第 8 位一起被写入 IIC 移位寄存器 n（IICn），然后被输出。接收到的地址数据被写入 IICn 寄存器（n = 0 至 2）。第 8 位用来指示传输方向，在 17.6.3 指定传输方向中描述。

从设备地址被分配至 IICn 寄存器的高 7 位。

17.6.3 传输方向指示

除了 7 位地址数据之外，主设备发送 1 位数据指示传输方向。当该传输方向指示位为 0 时，表示主设备向从设备传输数据。当该传输方向指示位为 1 时，表示主设备由从设备接收数据。

图 17-10. 传输方向指示



17.6.4 ACK

$\overline{\text{ACK}}$ 用来确认发送设备和接收设备的串行数据传输状态。

每接收到的 8 位数据，接收设备返回一次 $\overline{\text{ACK}}$ 信号。

通常情况下，发送设备发送 8 位数据后，接收到一个 $\overline{\text{ACK}}$ 信号。当 $\overline{\text{ACK}}$ 信号由接收端返回后，接收被判定为正常并且继续进行操作。使用 IICSn.ACKDn 位对 $\overline{\text{ACK}}$ 信号的检测进行确认。

当主设备是接收设备时，收到最后一个数据之后，不返回 $\overline{\text{ACK}}$ 信号且产生停止条件。当从设备为接收设备且不返回 $\overline{\text{ACK}}$ 信号时，主设备产生一个停止条件或一个重新开始条件，然后停止当前的通信传输。返回 $\overline{\text{ACK}}$ 失败的情况可由以下因素导致：

- (a) 接收没有正常执行。
- (b) 接收到最末数据。
- (c) 指定地址上的接收设备（从设备）不存在。

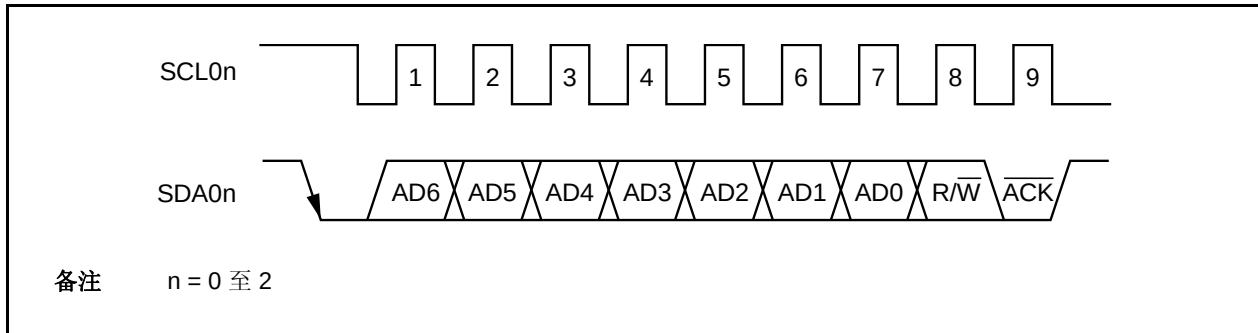
在第 9 个时钟期间，当接收设备将 SDA0n 信号线设置为低电平时，产生 $\overline{\text{ACK}}$ 信号（接收正常）。

当 IICCN.ACKEn 位设置为 1 时，使能自动产生 $\overline{\text{ACK}}$ 。发送 7 位地址之后的第 8 位会导致 IICSn.TRCn 位被置位。通常情况下，设置 ACKE0 位为 1 作为接收的应答（TRCn 位 = 0）。

当从设备进行接收（TRCn 位 = 0）时，如果从设备不能接收数据或不再需要接收更多数据，将 ACKEn 位清除为 0，提示主设备，不能再接收更多数据。

与上述相似，当主设备进行接收（TRCn 位 = 0）时，并且不再需要后续数据，将 ACKEn 位清除为 0，来防止 $\overline{\text{ACK}}$ 信号的产生。这样可以通知从设备（发送设备）数据发送结束（发送停止）。

图 17-11. $\overline{\text{ACK}}$



接收到本地地址后，不管 ACKEn 位的设置值如何，都会自动产生 $\overline{\text{ACK}}$ 信号。如果收到的地址不是本地地址（NACK），不产生 $\overline{\text{ACK}}$ 信号。

接收到扩展码时，在产生 $\overline{\text{ACK}}$ 信号之前，将 ACKEn 位设置为 1。

数据接收期间 $\overline{\text{ACK}}$ 的产生方式取决于等待时序的设置，描述如下：

- 当选择 8 时钟等待时（IICCN.WTIMn 位 = 0）：
如果等待状态取消前将 ACKEn 位设置为 1， $\overline{\text{ACK}}$ 信号在 SCL0n 引脚第 8 个时钟下降沿产生。
- 当选择 9 时钟等待时（IICCN.WTIMn 位 = 1）：
如果 ACKEn 位被预先设置为 1，产生 $\overline{\text{ACK}}$ 信号。

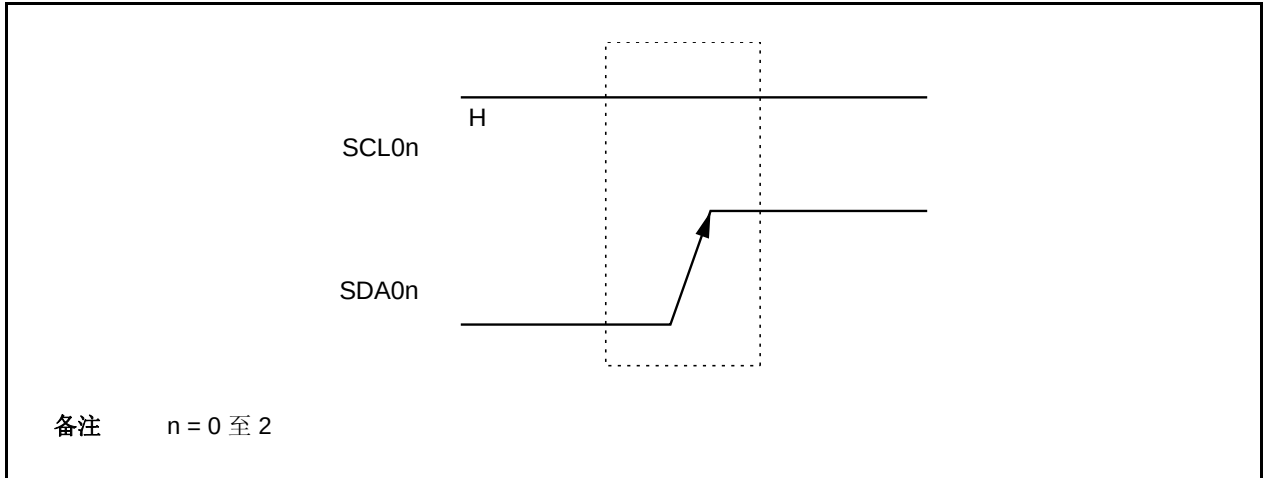
备注 n = 0 至 2

17.6.5 停止条件

当 SCL0n 引脚为高电平时，SDA0n 引脚由低电平变为高电平，即产生一个停止条件（n = 0 至 2）。

当主设备发向从设备的串行传输完成之后，产生一个停止条件。被用作从设备时，可以检测开始条件。

图 17-12. 停止条件



当 IICn.SPTn 位被设置为 1 时，发生一个停止条件。当检测到停止条件时，IICSn.SPDn 位被置为 1，且当 IICn.SPIEn 位置为 1 时，产生一个中断请求信号（INTIICn）。（n = 0 至 2）

17.6.6 等待状态

等待状态用来通知通信伙伴，设备（主设备或从设备）已经准备就绪可以进行收发数据（即：处于等待状态）。

将 SCL0n 引脚设为低电平，通知通信伙伴本机已经处于等待状态。当主设备和从设备的等待状态都被取消时，下一次传输可以开始。（n = 0 至 2）

图 17-13. 等待状态（1/2）

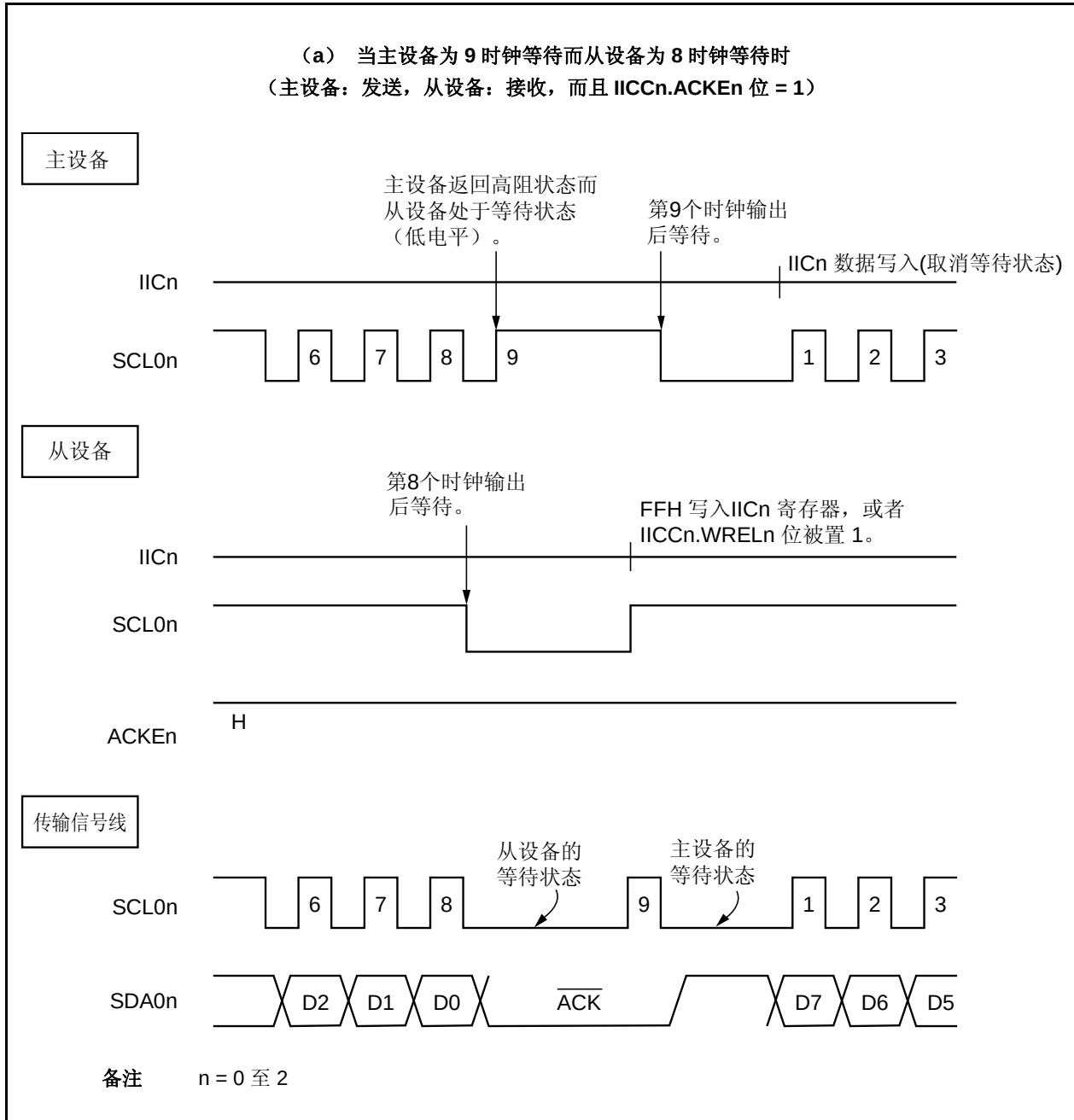
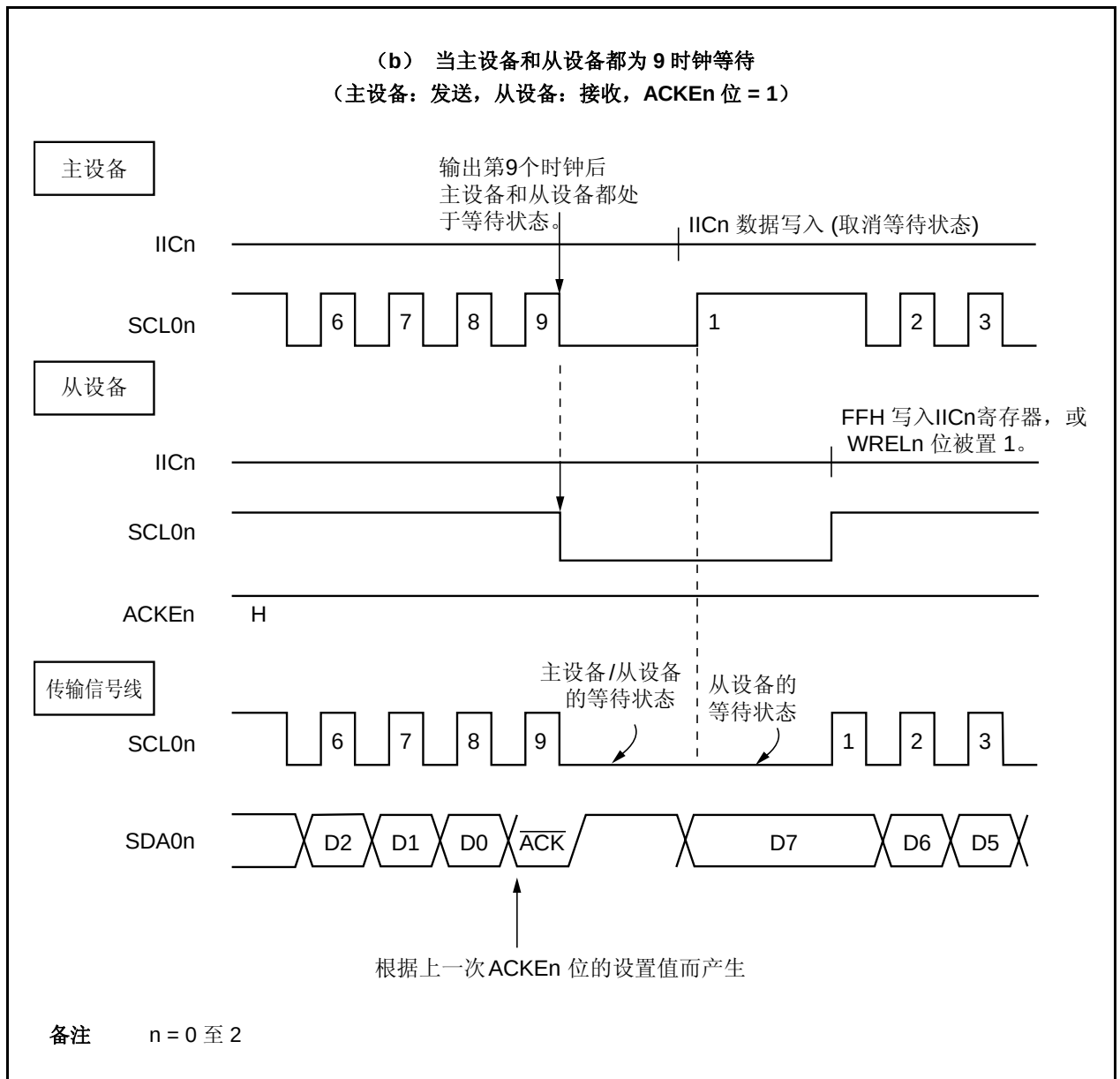


图 17-13. 等待状态 (2/2)



根据 IICn.WTIMn 位的设置, 可能会自动产生等待状态。(n = 0 至 2)

通常情况下, 当 IICn.WRELn 位置为 1, 或向接收端的 IICn 寄存器写入 FFH 时, 取消等待状态, 发送端向 IICn 寄存器写入数据, 清除等待状态。

主设备还可以通过以下方式取消等待状态。

- 将 IICn.STTn 位置为 1。
- 将 IICn.SPTn 位置为 1。

17.6.7 等待状态取消方法

在使用I²C0n的情况下，一般使用以下方式取消等待状态（n = 0 至 2）。

- 向 IICn 寄存器写入数据。
- 将 IICn.WRELn 位置为 1（取消等待状态）。
- 将 IICn.STTn 位置为 1（产生开始条件）。
- 将 IICn.SPTn 位置为 1（产生停止条件）。

如果执行任何一种等待取消动作，I²C0n将取消等待状态，并重新开始通信。

当取消等待状态并发送数据（包括地址）时，向 IICn 寄存器写入数据。

为了在取消等待状态之后接收数据，或完成数据发送，将 WRELn 位置为 1。

为了在取消等待状态之后产生一个重新开始条件，将 STTn 位设置为 1。

为了在取消等待状态之后产生一个停止条件，将 SPTn 位设置为 1。

对每个等待状态只可执行一次取消操作。

例如，如果将 WRELn 位设置为 1，取消等待状态，然后将数据写入 IICn 寄存器，SDA0n 信号线的时序变化和 IICn 寄存器的写入时序可能发生冲突，可能导致输出到 SDA0n 信号线上的数据不正确。

即使在其它操作情况下，如果通信中途停止，将 IICn.IICEn 位清除为 0 也会停止通信，使能等待状态的取消。

如果I²C 总线因噪声等原因被死锁，将IICn.LRELn 位设置为 1，会退出通信操作，使能等待状态的取消。

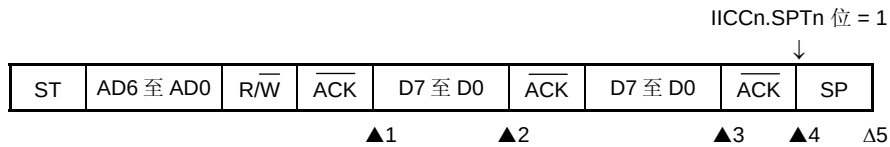
17.7 I²C中断请求信号 (INTIICn)

以下展示了 INTIICn 中断请求信号产生时序和 INTIICn 信号时序处的 IICSn 寄存器的值。(n = 0 至 2)

17.7.1 主设备操作

(1) 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止 (正常发送/接收)

<1> 当 IICn.WTIMn 位 = 0 时



▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 1000X000B

▲3: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)

▲4: IICSn 寄存器 = 1000XX00B

Δ5: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 1000X100B

▲3: IICSn 寄存器 = 1000XX00B

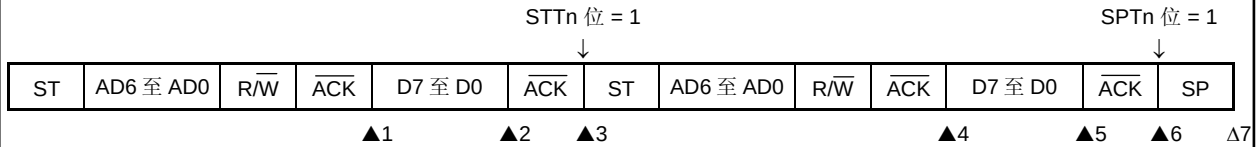
Δ4: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
2. n = 0 至 2

(2) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止 (重新开始)

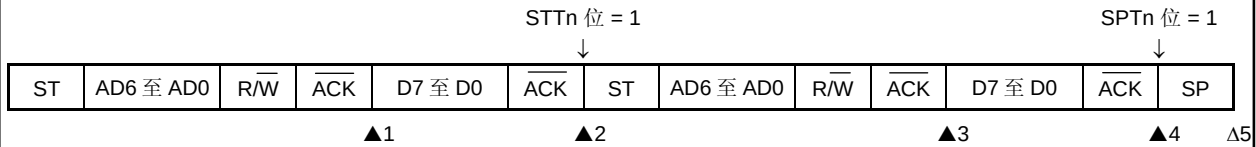
<1> 当 WTIMn 位 = 0 时



- ▲1: IICSn 寄存器 = 1000X110B
 ▲2: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)
 ▲3: IICSn 寄存器 = 1000XX00B (WTIMn 位 = 0)
 ▲4: IICSn 寄存器 = 1000X110B (WTIMn 位 = 0)
 ▲5: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)
 ▲6: IICSn 寄存器 = 1000XX00B
 Δ7: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

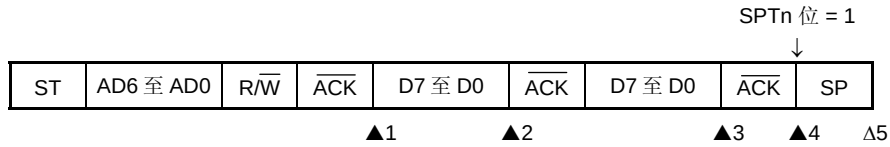


- ▲1: IICSn 寄存器 = 1000X110B
 ▲2: IICSn 寄存器 = 1000XX00B
 ▲3: IICSn 寄存器 = 1000X110B
 ▲4: IICSn 寄存器 = 1000XX00B
 Δ5: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

(3) 开始 ~ 扩展码 ~ 数据 ~ 数据 ~ 停止 (发送扩展码)

<1> 当 WTIMn 位 = 0 时

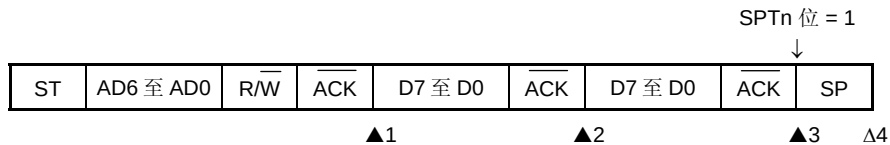


- ▲1: IICSn 寄存器 = 1010X110B
- ▲2: IICSn 寄存器 = 1010X000B
- ▲3: IICSn 寄存器 = 1010X000B (WTIMn 位 = 1)
- ▲4: IICSn 寄存器 = 1010XX00B
- Δ5: IICSn 寄存器 = 00000001B

备注

- 1. ▲: 总是产生
Δ: 仅当 SPIEn 位 = 1 时产生
X: 无需理会
- 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



- ▲1: IICSn 寄存器 = 1010X110B
- ▲2: IICSn 寄存器 = 1010X100B
- ▲3: IICSn 寄存器 = 1010XX00B
- Δ4: IICSn 寄存器 = 00000001B

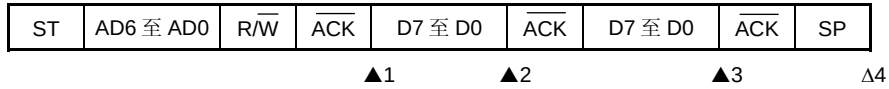
备注

- 1. ▲: 总是产生
Δ: 仅当 SPIEn 位 = 1 时产生
X: 无需理会
- 2. n = 0 至 2

17.7.2 从设备操作（当接收从设备地址数据（地址匹配）时）

(1) 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止

<1> 当 IICn.WTIMn 位 = 0 时



▲1: IICSn 寄存器 = 0001X110B

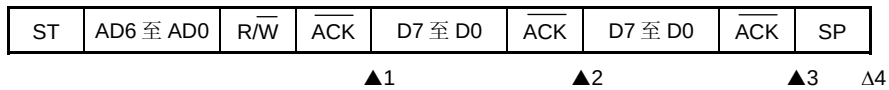
▲2: IICSn 寄存器 = 0001X000B

▲3: IICSn 寄存器 = 0001X000B

Δ4: IICSn 寄存器 = 00000001B

- 备注
- ▲: 总是产生
Δ: 仅当 IICn.SPIEn 位 = 1 时产生
X: 无需理会
 - n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1: IICSn 寄存器 = 0001X110B

▲2: IICSn 寄存器 = 0001X100B

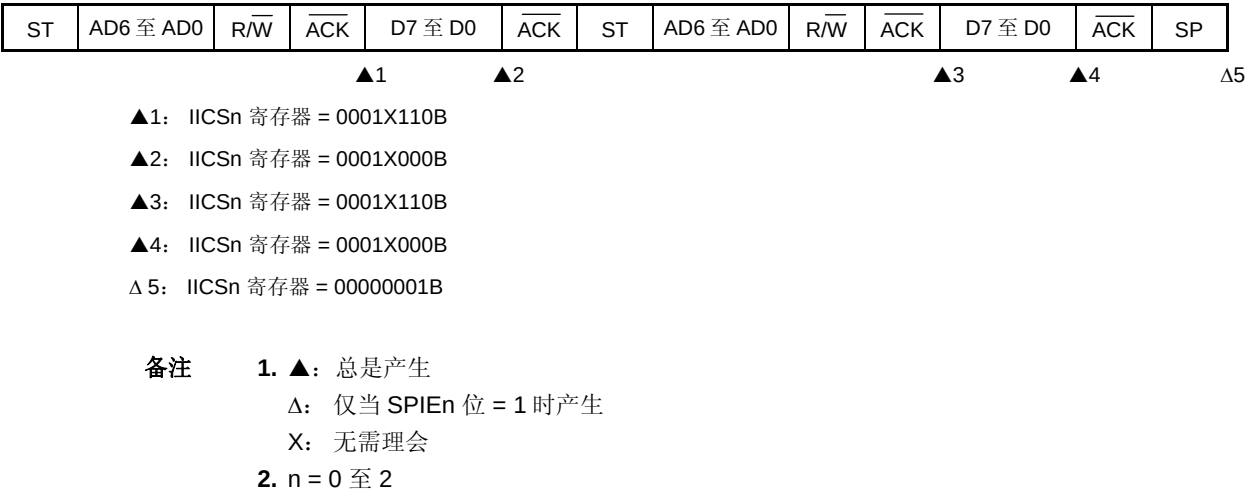
▲3: IICSn 寄存器 = 0001XX00B

Δ4: IICSn 寄存器 = 00000001B

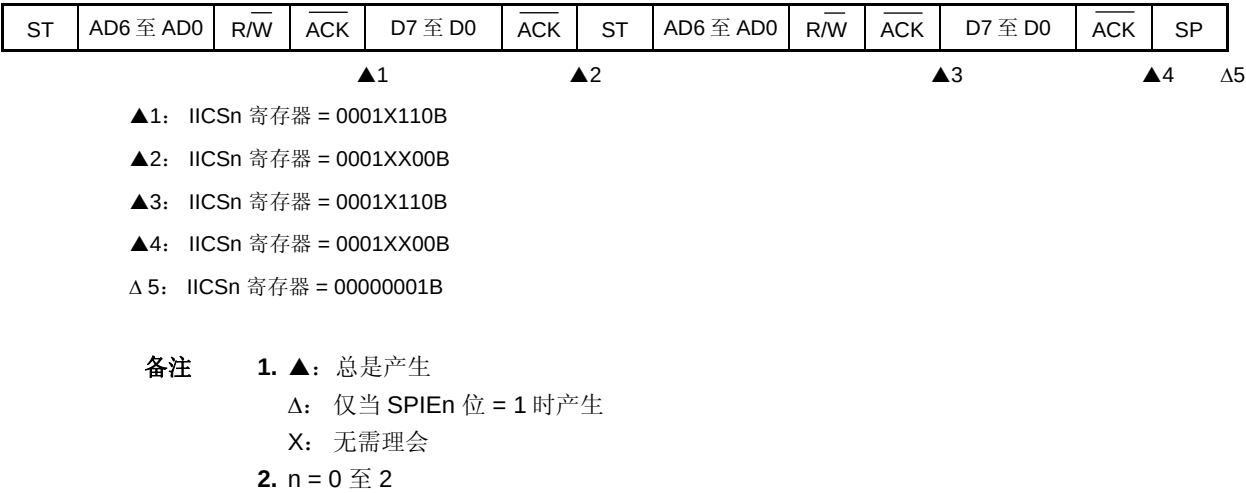
- 备注
- ▲: 总是产生
Δ: 仅当 SPIEn 位 = 1 时产生
X: 无需理会
 - n = 0 至 2

(2) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0 (重新开始后, 地址匹配) 时

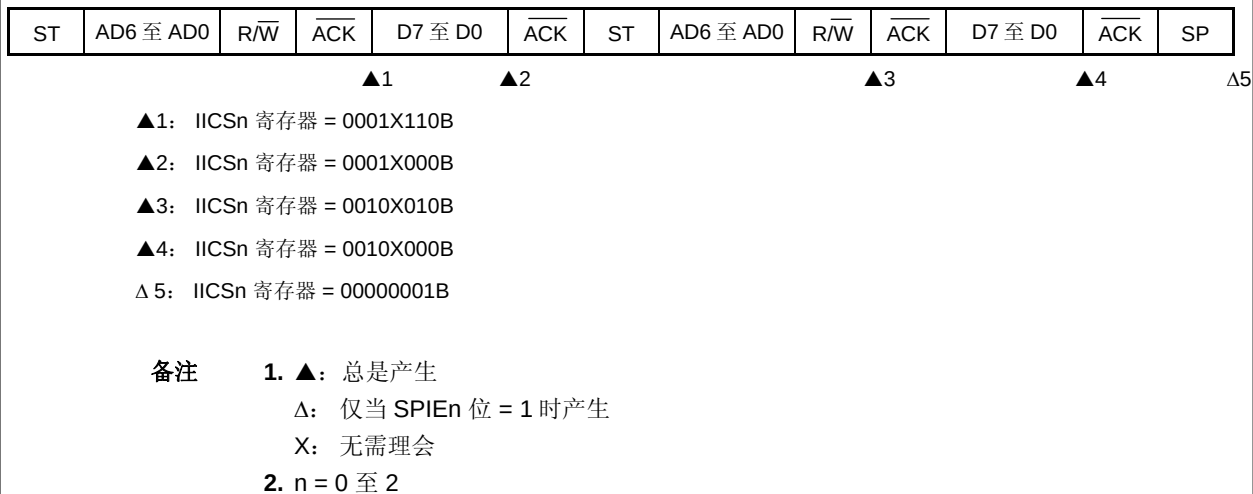


<2> 当 WTIMn 位 = 1 (重新开始后, 地址匹配) 时

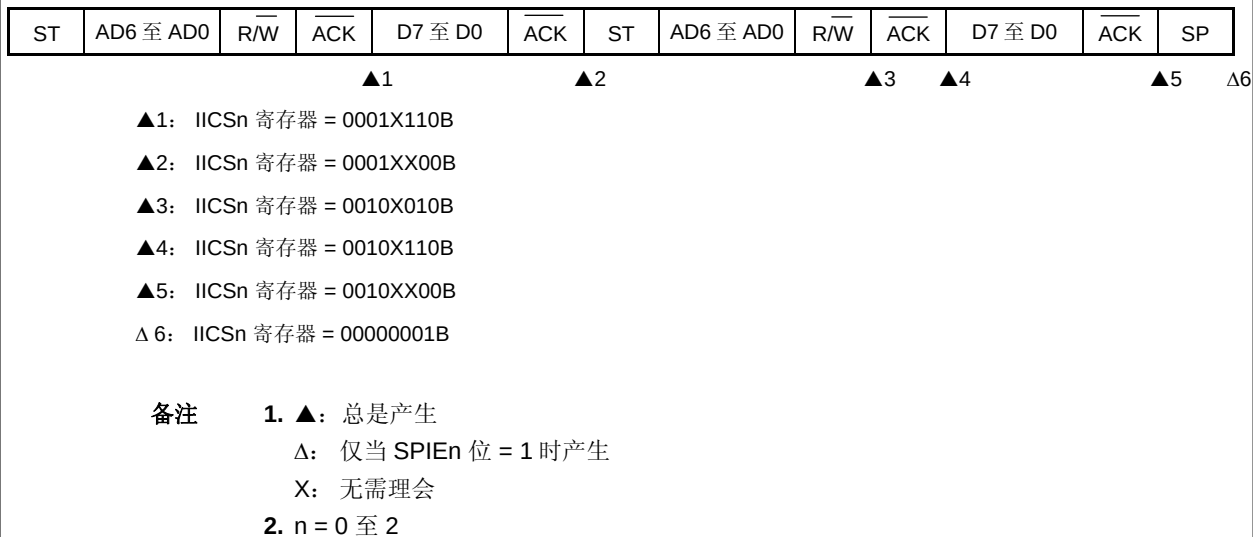


(3) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 扩展码 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0 (重新开始后, 扩展码接收) 时



<2> 当 WTIMn 位 = 1 (重新开始后, 扩展码接收) 时



<1> 当 WTIMn 位 = 0（重新开始后，地址不匹配（= 无扩展码））时

ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	ST	AD6 至 AD0	R/W	ACK	D7 至 D0	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

- ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
- n = 0 至 2

<2> 当 WTIMn 位 = 1（重新开始后，地址不匹配（= 无扩展码））时

ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	ST	AD6 至 AD0	R/W	ACK	D7 至 D0	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

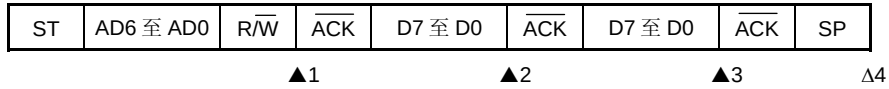
备注

- ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
- n = 0 至 2

17.7.3 从设备操作（当接收扩展码时）

(1) 开始 ~ 扩展码 ~ 数据 ~ 数据 ~ 停止

<1> 当 IICn.WTIMn 位 = 0 时



▲1: IICSn 寄存器 = 0010X010B

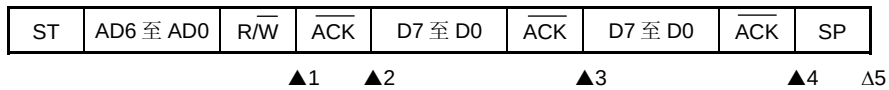
▲2: IICSn 寄存器 = 0010X000B

▲3: IICSn 寄存器 = 0010X000B

Δ4: IICSn 寄存器 = 00000001B

- 备注
- ▲: 总是产生
Δ: 仅当 IICn.SPIEn 位 = 1 时产生
X: 无需理会
 - n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1: IICSn 寄存器 = 0010X010B

▲2: IICSn 寄存器 = 0010X110B

▲3: IICSn 寄存器 = 0010X100B

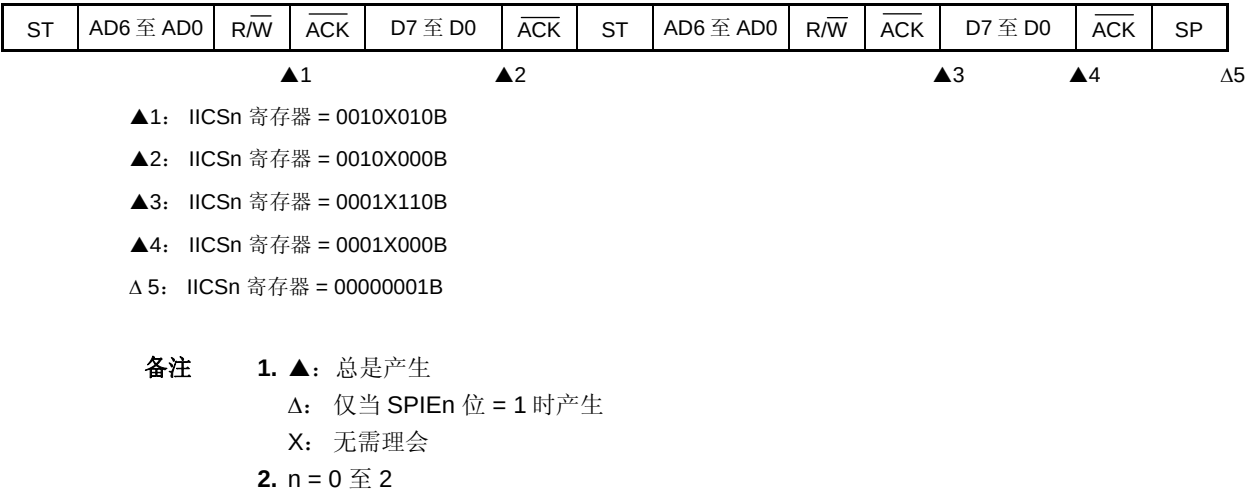
▲4: IICSn 寄存器 = 0010XX00B

Δ5: IICSn 寄存器 = 00000001B

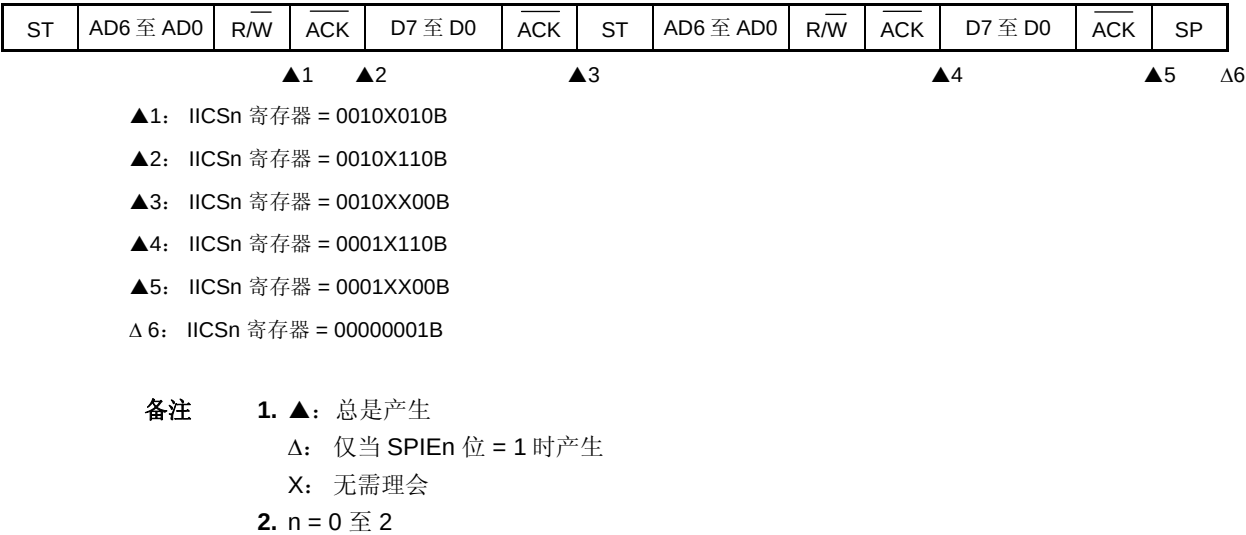
- 备注
- ▲: 总是产生
Δ: 仅当 SPIEn 位 = 1 时产生
X: 无需理会
 - n = 0 至 2

(2) 开始 ~ 扩展码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0（重新开始后，地址匹配）时

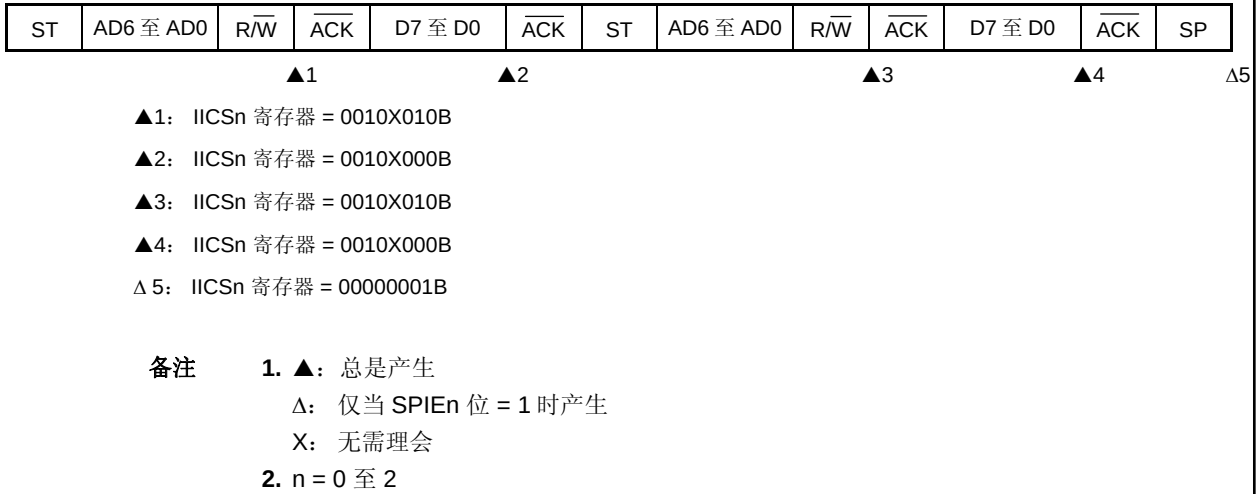


<2> 当 WTIMn 位 = 1（重新开始后，地址匹配）时

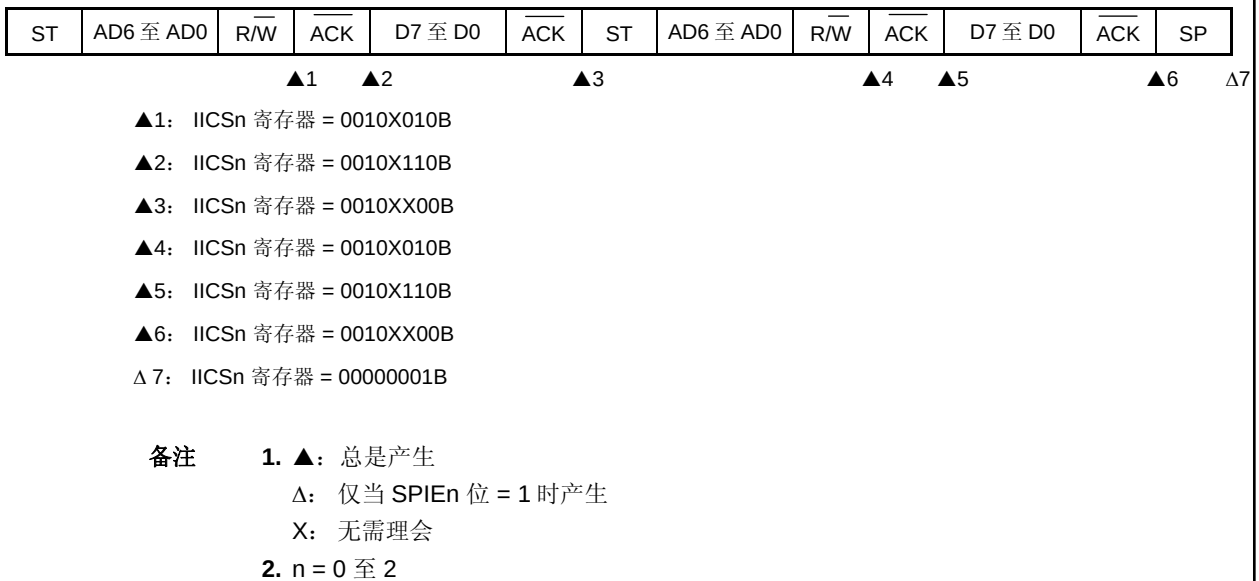


(3) 开始 ~ 扩展码 ~ 数据 ~ 开始 ~ 扩展码 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0 (重新开始后, 扩展码接收) 时



<2> 当 WTIMn 位 = 1 (重新开始后, 扩展码接收) 时



(4) 开始 ~ 扩展码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0 (重新开始后, 地址不匹配 (= 无扩展码)) 时

ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	ST	AD6 至 AD0	R/W	ACK	D7 至 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

Δ 4: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生

Δ: 仅当 SPIEn 位 = 1 时产生

X: 无需理会

2. n = 0 至 2

<2> 当 WTIMn 位 = 1 (重新开始后, 地址不匹配 (= 无扩展码)) 时

ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	ST	AD6 至 AD0	R/W	ACK	D7 至 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

Δ 5: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生

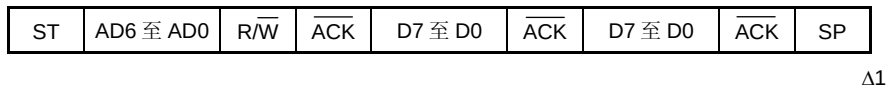
Δ: 仅当 SPIEn 位 = 1 时产生

X: 无需理会

2. n = 0 至 2

17.7.4 不通信时的操作

(1) 开始 ~ 扩展码 ~ 数据 ~ 数据 ~ 停止



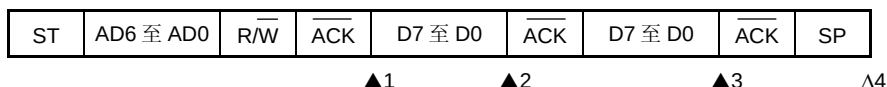
Δ1: IICSn 寄存器 = 00000001B

备注 1. Δ: 仅当 SPIEn 位 = 1 时产生
2. n = 0 至 2

17.7.5 仲裁失败操作（仲裁失败后作为从设备操作）

(1) 当在从设备地址数据传输期间发生仲裁失败

<1> 当 IICn.WTIMn 位 = 0 时



▲1: IICSn 寄存器 = 0101X110B（例如：中断服务期间读取 IICSn.ALDn 位时）

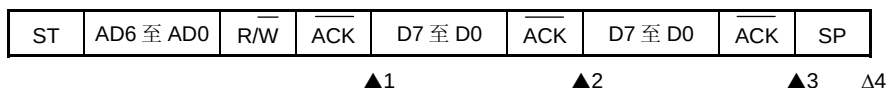
▲2: IICSn 寄存器 = 0001X000B

▲3: IICSn 寄存器 = 0001X000B

Δ4: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是产生
Δ: 仅当 IICn.SPIEn 位 = 1 时产生
X: 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1: IICSn 寄存器 = 0101X110B（例如：中断服务期间读取 ALDn 位时）

▲2: IICSn 寄存器 = 0001X100B

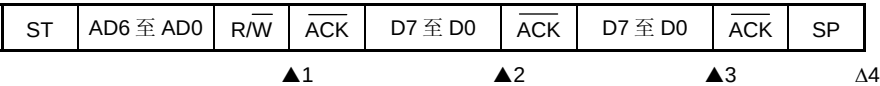
▲3: IICSn 寄存器 = 0001XX00B

Δ4: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是产生
Δ: 仅当 SPIEn 位 = 1 时产生
X: 无需理会
2. n = 0 至 2

(2) 在扩展码的传输期间发生仲裁失败

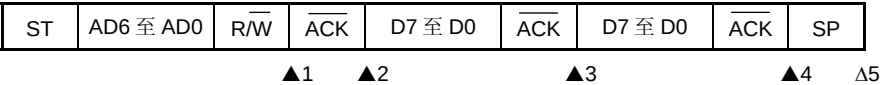
<1> 当 WTIMn 位 = 0 时



- ▲1: IICSn 寄存器 = 0110X010B (例如: 中断服务期间读取 ALDn 位时)
▲2: IICSn 寄存器 = 0010X000B
▲3: IICSn 寄存器 = 0010X000B
Δ4: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

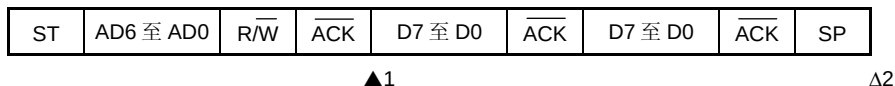


- ▲1: IICSn 寄存器 = 0110X010B (例如: 中断服务期间读取 ALDn 位时)
▲2: IICSn 寄存器 = 0010X110B
▲3: IICSn 寄存器 = 0010X100B
▲4: IICSn 寄存器 = 0010XX00B
Δ5: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

17.7.6 发生仲裁失败时的操作（在仲裁失败后不进行通信）

(1) 在从设备地址数据传输期间发生仲裁失败

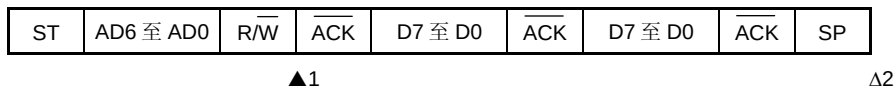


▲1: IICSn 寄存器 = 01000110B（例如：中断服务期间读取 IICSn.ALDn 位时）

Δ 2: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
Δ: 仅当 IICn.SPIEn 位 = 1 时产生
 2. n = 0 至 2

(2) 在扩展码的传输期间发生仲裁失败



▲1: IICSn 寄存器 = 0110X010B（例如：中断服务期间读取 ALDn 位时）

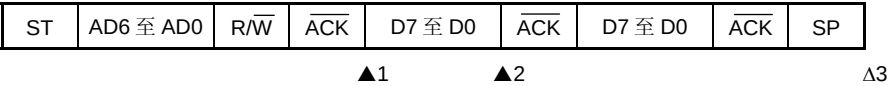
由软件设置 IICn.LRELn 位。

Δ 2: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
Δ: 仅当 SPIEn 位 = 1 时产生
X: 无需理会
 2. n = 0 至 2

(3) 在数据传输期间发生仲裁失败

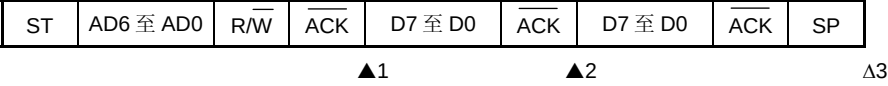
<1> 当 IICSn.WTIMn 位 = 0 时



- ▲1: IICSn 寄存器 = 10001110B
▲2: IICSn 寄存器 = 01000000B (例如: 中断服务期间读取 ALDn 位时)
Δ 3: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

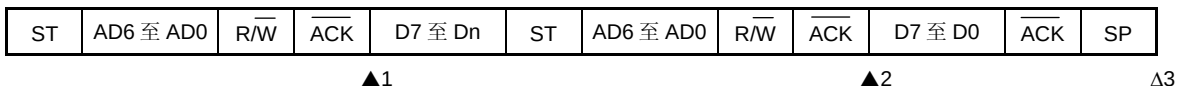


- ▲1: IICSn 寄存器 = 10001110B
▲2: IICSn 寄存器 = 01000100B (例如: 中断服务期间读取 ALDn 位时)
Δ 3: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 2. n = 0 至 2

(4) 在数据传输期间，因为重新开始条件导致发生仲裁失败

<1> 无扩展码（例如：地址不匹配）



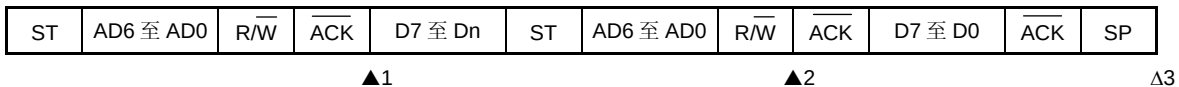
▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 01000110B（例如：中断服务期间读取 ALDn 位时）

Δ3: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. Dn = D6 至 D0
 n = 0 至 2

<2> 扩展代码



▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 0110X010B（例如：中断服务期间读取 ALDn 位时）

由软件设置 IICn.LRELn 位。

Δ3: IICSn 寄存器 = 00000001B

- 备注**
1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. Dn = D6 至 D0
 n = 0 至 2

ST	AD6 至 AD0	$\overline{R/W}$	$\overline{ACK}$	D7 至 Dn	SP
----	-----------	------------------	------------------	---------	----

▲1 Δ2

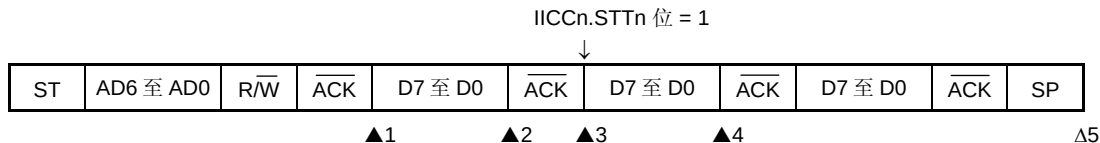
▲1: IICSn 寄存器 = 1000X110B
Δ 2: IICSn 寄存器 = 01000001B

备注

- ▲: 总是产生
Δ: 仅当 SPIEn 位 = 1 时产生
X: 无需理会
- Dn = D6 至 D0
n = 0 至 2

(6) 当试图产生重新开始条件时，因为 SDA0n 引脚的低电平导致发生仲裁失败

<1> 当 WTIMn 位 = 0 时



▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)

▲3: IICSn 寄存器 = 1000XX00B (WTIMn 位 = 0)

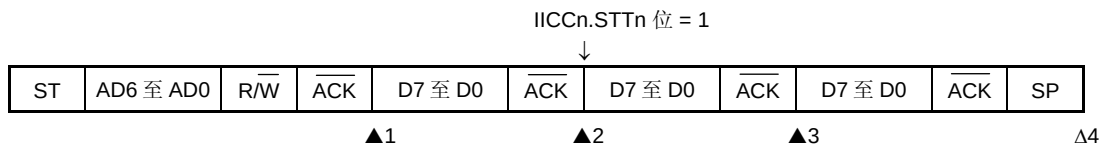
▲4: IICSn 寄存器 = 01000000B (例如：中断服务期间读取 ALDn 位时)

Δ5: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
Δ: 仅当 SPIEn 位 = 1 时产生
X: 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 1000XX00B

▲3: IICSn 寄存器 = 01000100B (例如：中断服务期间读取 ALDn 位时)

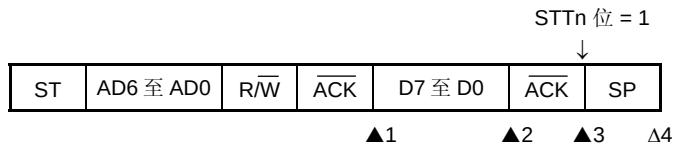
Δ4: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
Δ: 仅当 SPIEn 位 = 1 时产生
X: 无需理会
2. n = 0 至 2

(7) 当试图产生重新开始条件时，由于停止条件导致发生仲裁失败

<1> 当 WTIMn 位 = 0 时



▲1: IICSn 寄存器 = 1000X110B

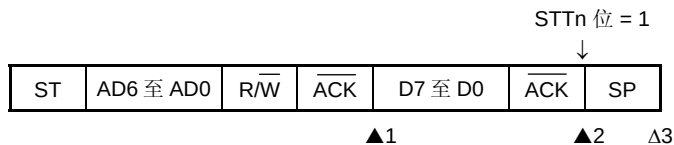
▲2: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)

▲3: IICSn 寄存器 = 1000XX00B

Δ 4: IICSn 寄存器 = 01000001B

备注 1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1: IICSn 寄存器 = 1000X110B

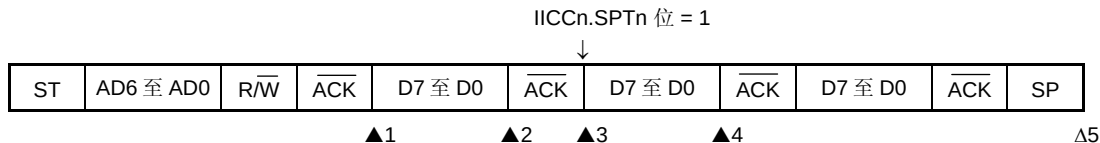
▲2: IICSn 寄存器 = 1000XX00B

Δ 3: IICSn 寄存器 = 01000001B

备注 1. ▲: 总是产生
 Δ: 仅当 SPIEn 位 = 1 时产生
 X: 无需理会
 2. n = 0 至 2

(8) 当试图产生停止条件时，由于 SDA0n 引脚为低电平导致发生仲裁失败

<1> 当 WTIMn 位 = 0 时



▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)

▲3: IICSn 寄存器 = 1000XX00B (WTIMn 位 = 0)

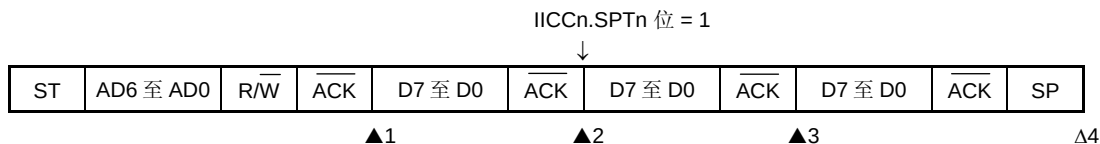
▲4: IICSn 寄存器 = 01000000B (例如: 中断服务期间读取 ALDn 位时)

Δ5: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
Δ: 仅当 SPIEn 位 = 1 时产生
X: 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 1000XX00B

▲3: IICSn 寄存器 = 01000000B (例如: 中断服务期间读取 ALDn 位时)

Δ4: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是产生
Δ: 仅当 SPIEn 位 = 1 时产生
X: 无需理会
2. n = 0 至 2

17.8 中断请求信号（INTIICn）的产生时序和等待控制

IICn.WTIMn 位的设置可以决定 INTIICn 寄存器的产生时序和对应的等待控制，如下所示。（n = 0 至 2）

表 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	数据发送	地址	数据接收	数据发送
0	9 ^{#1,2}	8 ^{#2}	8 ^{#2}	9	8	8
1	9 ^{#1,2}	9 ^{#2}	9 ^{#2}	9	9	9

- 注
1. 仅当接收的地址和 SVAn 寄存器设定的值相匹配时，在第 9 个时钟的下降沿产生从设备的 INTIICn 信号和等待时期。
此刻，不论 IICn.ACKEn 位如何设置，都产生 ACK 信号，对于已经接收到扩展码的从设备，INTIICn 信号在第 8 个时钟的下降沿发生。
当重新开始后地址不匹配时，INTIICn 信号在第 9 个时钟的下降沿发生，但是，不发生等待。
 2. 如果接收地址和 SVAn 寄存器的内容不匹配，而且不能收到扩展码，则既不产生 INTIICn 信号，也不产生等待状态。

- 备注
- 1.表中的数量表示串行时钟信号的时钟数目，中断请求信号和等待控制都和这些时钟信号的下降沿同步。
 2. n = 0 至 2

(1) 地址发送/接收期间

- 从设备操作： 决定中断请求和等待时序时不考虑 WTIMn 位的设置。
- 主设备操作： 不论 WTIMn 位如何设置，中断请求和等待时序都在第 9 个时钟的下降沿发生。

(2) 在数据接收期间

- 主设备/从设备操作： 中断请求和等待时序由 WTIMn 位的设置决定。

(3) 数据发送期间

- 主设备/从设备操作： 中断请求和等待时序由 WTIMn 位的设置决定。

(4) 等待状态取消方法

等待状态的四种取消方法如下所示：

- 将 IICn.WRELn 位置为 1。
- 向 IICn 寄存器写入数据。
- 开始条件的设置 (IICn.STTn 位 = 1) ^注。
- 停止条件的设置 (IICn.SPTn 位 = 1) ^注。

注 仅限主设备。

当已经选择 8 时钟等待时 (WTIMn 位 = 0)，在取消等待之前必须确定是否已经产生 $\overline{ACK}$ 信号。

备注 n = 0 至 2

(5) 停止条件检测

当检测到停止条件时，会产生 INTIICn 信号。

备注 n = 0 至 2

17.9 地址匹配检测方法

在I²C总线模式下，通过发送从设备的地址，主设备可以选择特定的从设备。

地址匹配的检测由硬件自动执行。当本地地址被置入 SVAn 寄存器以后，且置入 SVAn 寄存器的本地地址和主设备发送的从设备地址匹配时，或接收到扩展码时，产生一个 INTIICn 信号。（n = 0 至 2）

17.10 错误检测

在I²C总线模式下，串行数据总线引脚（SDA0n）在数据传送期间的状态被发送设备的IICn 寄存器捕获，于是，发送之前IICn 寄存器内的数据就可以和已发送的IICn数据进行比较，从而能够检测传输错误。当比较值不匹配时，判断为发生传输错误。（n = 0 至 2）

17.11 扩展代码

- （1）当接收地址的高 4 位是 0000 或 1111 时，扩展码标志（IICSn.EXCn 位）被置位，表示扩展码接收，且在第 8 个时钟的下降沿处产生一个中断请求信号（INTIICn）。（n = 0 至 2）
存储在 SVAn 寄存器中的本地地址不受影响。

- （2）如果通过 10 位地址传输将 11110xx0 设置到 SVAn 寄存器，且 11110xx0 数据由主设备发出，其结果如下所示。注意，INTIICn 信号在第 8 个时钟的下降沿处产生（n = 0 至 2）。

- 数据匹配的高 4 位：EXCn 位 = 1
- 数据匹配的第 7 位：IICSn.COIn 位 = 1

- （3）根据扩展码之后的数据，中断请求信号发生后的处理也各不相同，这些处理由软件执行。

例如，接收到扩展码以后，当不希望作为从设备操作时，将 IICn.LRELn 位设置为 1，CPU 将进入下一次通信等待状态。

表 17-4. 扩展码位的定义

从设备地址	R/W 位	描述
0000 000	0	通用呼叫地址
0000 000	1	开始字节
0000 001	X	CBUS 地址
0000 010	X	为不同总线格式保留的地址
1111 0xx	X	10 位从设备地址规范

17.12 仲裁

当多个主设备同时产生开始条件时（在 IICSn.STDn 位被置 1 之前，IICn.STTn 位已被置为 1），调整时钟数目，直到数据不同，多个主设备之间的通信才能执行，这种操作被称为仲裁（n = 0 至 2）。

当一个主设备在仲裁中失败时，在仲裁失败发生时序处将仲裁失败标志（IICSn.ALDn 位）设置为 1，且 SCL0n 信号线和 SDA0n 信号线都被设置为高阻状态，总线被释放。（n = 0 至 2）

仲裁失败的检测基于下次中断请求信号（INTIICn）的时序（当检测到停止条件时，第 8 或第 9 个时钟）和通过软件对 ALDn 位置 1 的设置（n = 0 至 2）。

关于中断请求信号时序的详情，参见 17.7 I²C 中断请求信号（INTIICn）。

图 17-14. 仲裁时序示例

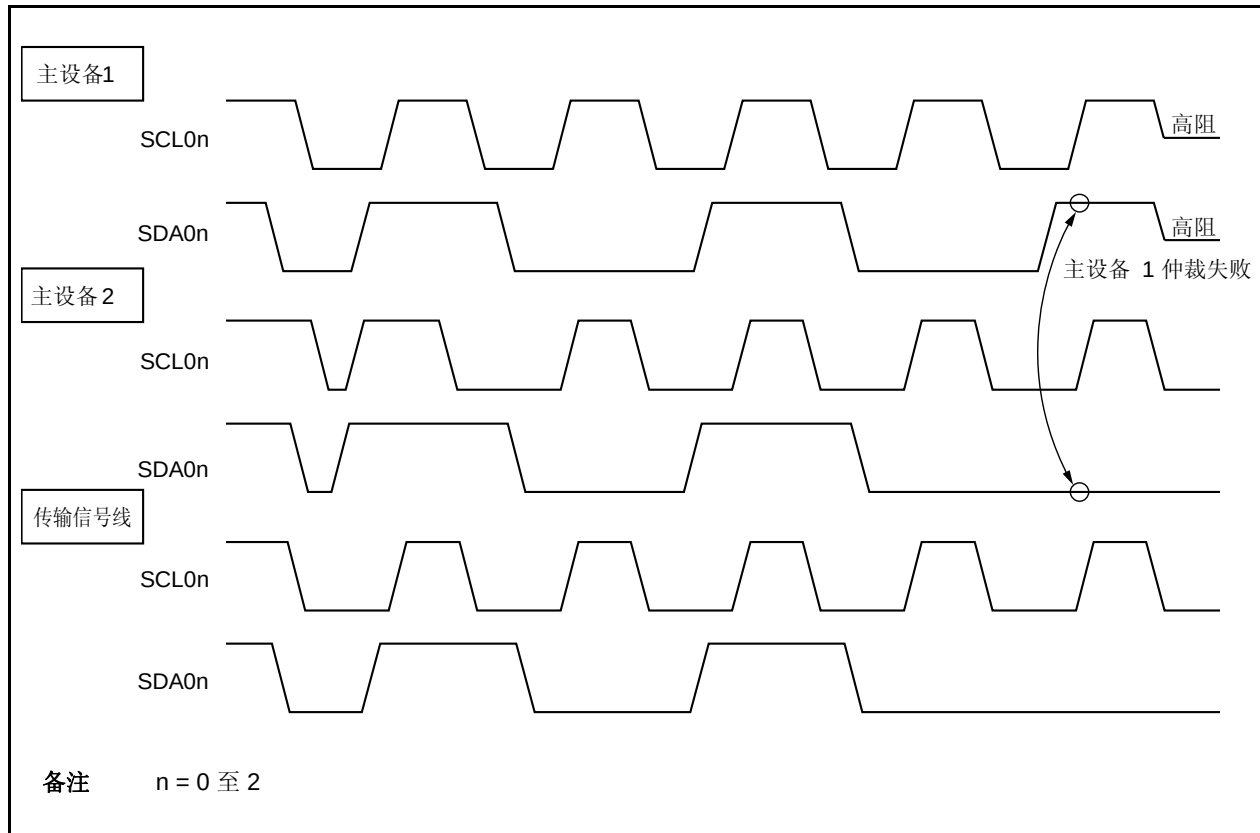


表 17-5. 仲裁期间的状态和中断请求信号发生时序

仲裁期间的状态	中断请求发生时序
发送地址传输	字节传输之后的第 8 个或第 9 个时钟的下降沿 ^{注 1}
地址传输后读出/写入数据	
发送扩展代码	
扩展码传输后读出/写入数据	
发送数据	
数据接收后 $\overline{\text{ACK}}$ 传送期间	
在数据传输期间检测到重新开始条件	
在数据传输期间检测到停止条件	当产生停止条件时（当 IICn.SPIEn 位 = 1 时） ^{注 2}
试图产生重新开始条件时 SDA0n 引脚是低电平	字节传输之后的第 8 个或第 9 个时钟的下降沿 ^{注 1}
试图产生重新开始条件时检测到停止条件	当产生停止条件时（当 IICn.SPIEn 位 = 1 时） ^{注 2}
试图产生停止条件时 SDA0n 引脚是低电平	字节传输之后的第 8 个或第 9 个时钟的下降沿 ^{注 1}
试图产生重新开始条件时 SCL0n 引脚是低电平	

- 注
1. 当 IICn.WTIMn 位 = 1 时，INTIICn 信号在第 9 个时钟的下降沿发生。当 WTIMn 位 = 0 且接收到扩展码的从地址时，INTIICn 信号在第 8 个时钟的下降沿发生（n = 0 至 2）。
 2. 当有可能发生仲裁时，在主设备操作要设置 SPIEn 位为 1（n = 0 至 2）。

17.13 唤醒功能

已经接收到本地地址和扩展码时，I²C 总线从设备的功能是产生一个中断请求信号（INTIICn）。

当地址数据不匹配时，该功能就会阻止不必要的 INTIICn 信号，从而使处理更有效率。

当检测到开始条件后，待机模式被唤醒。仲裁失败可能将主设备（已经产生开始条件）变为从设备，在这种可能情况下发送地址时，待机模式唤醒有效。

但是，当检测到停止条件后，不论唤醒功能如何设置，IICn.SPIEn 位总是置位，这可以决定 INTIICn 信号的使能或禁止（n = 0 至 2）。

17.14 通信保留

17.14.1 使能通信保留功能（IICFn.IICRSVn 位 = 0）时

在目前不能使用总线的情况下，开始主设备通信，通信保留功能可以在总线释放后使能开始条件的传输。在两种模式下，总线不能使用。

- 当仲裁结果既不是主设备操作，又不是从设备操作。
- 当接收到扩展码后且从设备操作被禁止时（当 IICFn.LRELn 位被置为 1 时，不返回 $\overline{\text{ACK}}$ 信号且总线释放）（n = 0 至 2）。

如果在不能使用总线时，将 IICFn.STTn 位置 1，将自动产生一个开始条件，并且在总线释放后（检测到停止条件后）设置等待状态。

检测到总线释放（检测到停止条件时）时，写入 IICFn 寄存器导致开始传送主设备地址。此时，IICFn.SPIEn 位应该被设置为 1（n = 0 至 2）。

当 STTn 位被设置为 1 后，根据总线状态决定工作模式（开始条件或通信保留）（n = 0 至 2）。

如果总线已经被释放产生开始条件

如果总线未被释放（待机模式）通信保留

要检测为 STTn 位决定何种工作模式，将 STTn 位设置为 1，待等待时期完毕后，检查 IICSn.MSTS_n 位（n = 0 至 2）。

应该由软件设置的等待期间在表 17-6 中列出。

这些等待时期可以通过 IICCLn 寄存器的 SMCn，CLn1 和 CLn0 位以及 IICXn.CLXn 位进行设置（n = 0 至 2）。

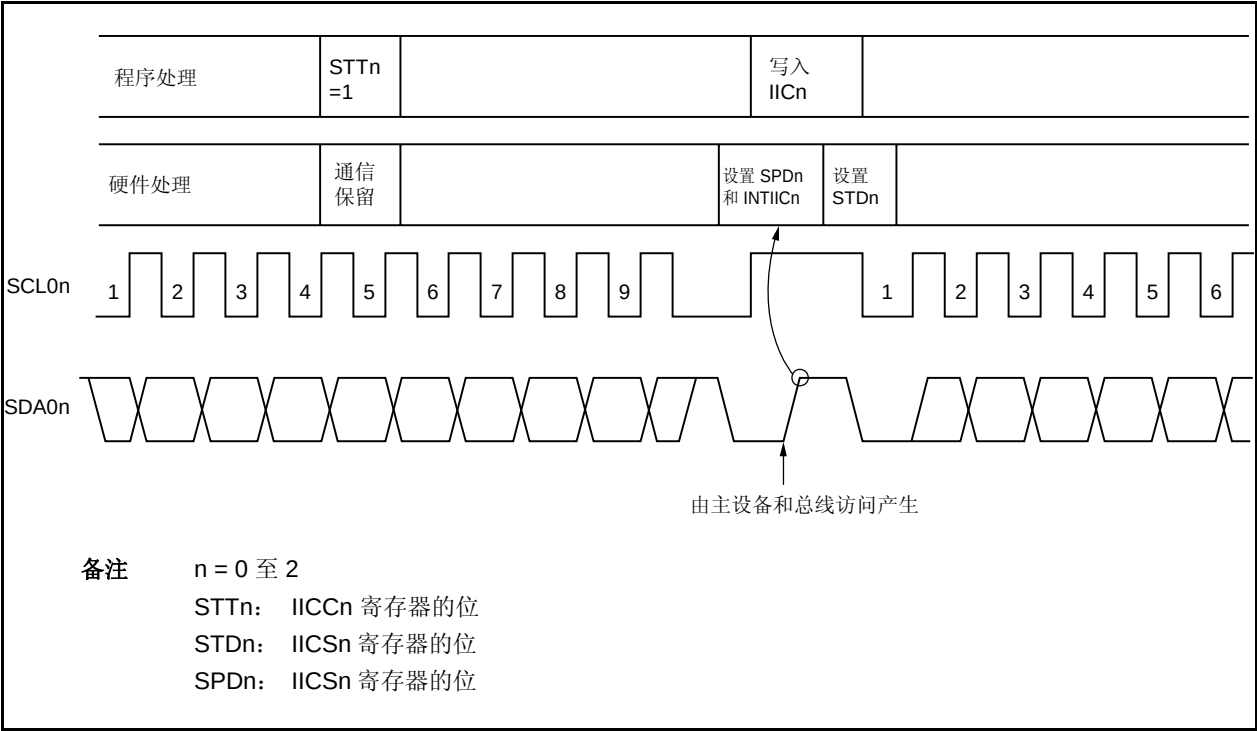
表 17-6. 等待时期

时钟选择	CLXn	SMCn	CLn1	CLn0	等待时间
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} /5 (当 OCKSm = 13H 时)	0	0	0	1	235 个时钟
f _{xx}	0	0	1	0	47 个时钟
f _{xx} (当 OCKSm = 18H 时)	0	0	1	1	37 个时钟
f _{xx} /2 (当 OCKSm = 10H 时)	0	0	1	1	74 个时钟
f _{xx} /3 (当 OCKSm = 11H 时)	0	0	1	1	111 个时钟
f _{xx} /4 (当 OCKSm = 12H 时)	0	0	1	1	148 个时钟
f _{xx} /5 (当 OCKSm = 13H 时)	0	0	1	1	185 个时钟
f _{xx} (当 OCKSm = 18H 时)	0	1	0	×	16 个时钟
f _{xx} /2 (当 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} /5 (当 OCKSm = 13H 时)	0	1	0	×	80 个时钟
f _{xx}	0	1	1	0	16 个时钟
f _{xx} (当 OCKSm = 18H 时)	0	1	1	1	13 个时钟
f _{xx} /2 (当 OCKSm = 10H 时)	0	1	1	1	26 个时钟
f _{xx} /3 (当 OCKSm = 11H 时)	0	1	1	1	39 个时钟
f _{xx} /4 (当 OCKSm = 12H 时)	0	1	1	1	52 个时钟
f _{xx} /5 (当 OCKSm = 13H 时)	0	1	1	1	65 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
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 至 2
m = 0, 1
 2. × = 无需理会

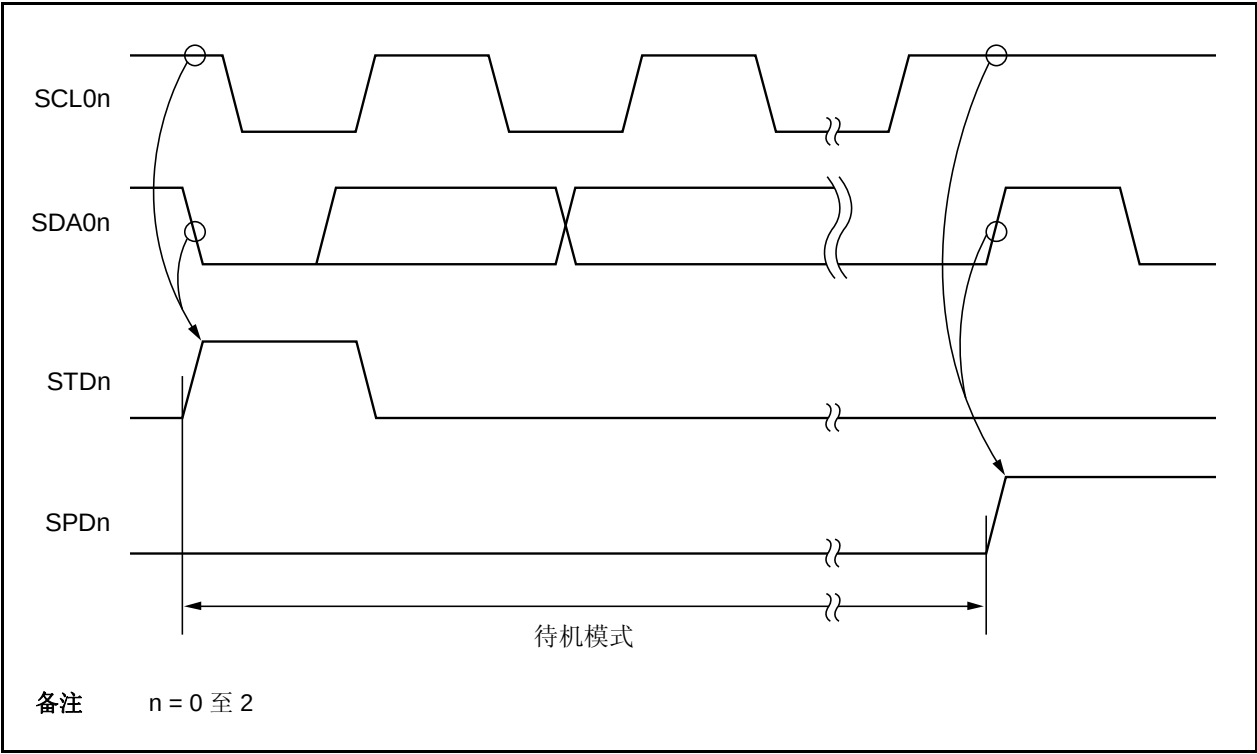
通信保留时序如下所示。

图 17-15. 通信保留时序



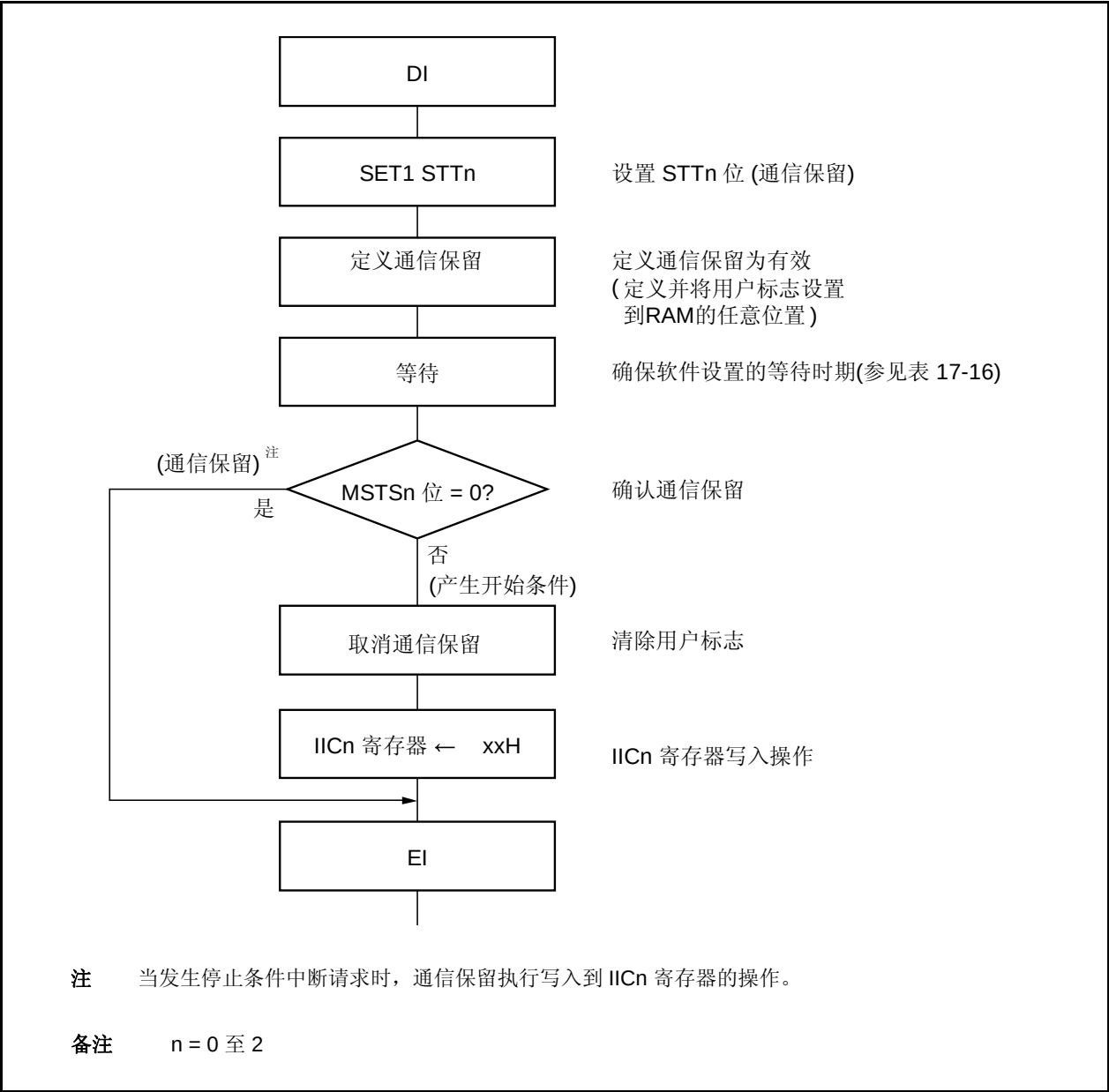
通过以下时序受理通信保留。IICSn.STDn 位设置为 1 后，在检测到停止条件前，设置 IICn.STTn 位为 1 就可以产生一次通信保留（n = 0 至 2）。

图 17-16. 受理通信保留的时序



通信保留的流程图展示如下。

图 17-17. 通信保留的流程图



17.14.2 禁止通信保留功能时 (IICFn.IICRSVn 位 = 1)

在总线通信中总线没有被使用时，当 IICFn.STTn 位被置位时，通信保留请求被拒绝，且不会产生开始条件。有两种模式下不能使用总线。

- 当仲裁结果既不是主设备操作，又不是从设备操作。
- 当接收到扩展码后且从设备操作被禁止时（当 IICFn.LRELn 位被置为 1 时，不返回 $\overline{\text{ACK}}$ 信号且总线释放）（n = 0 至 2）。

为了确认是产生了开始条件，还是请求遭到拒绝，检查 IICFn.STCFn 标志。在 STTn 位被设置为 1 后，直到 STCFn 标志被置位，需要的时间在表 17-7 中列出。因而，通过软件来保证这段需要的时间。

表 17-7. 等待期

OCKSENm	OCKSm1	OCKSm0	CLn1	CLn0	等待期
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. ×: 无需理会
 2. n = 0 至 2
 m = 0, 1

17.15 注意事项

- (1) 当 IICFn.STCENn 位 = 0 时

I²C0n 操作使能之后，不管总线的实际状态如何，总是立即认为处于总线通信状态（IICFn.IICBSYn 位 = 1）。要在尚未检测到停止条件的情况下执行主设备通信，先产生一个停止条件，在开始主设备通信之前释放总线。使用以下次序来产生停止条件。

- <1> 设置 IICCLn 寄存器。
- <2> 设置 IICFn.IICEn 位。
- <3> 设置 IICFn.SPTn 位。

- (2) 当 IICFn.STCENn 位 = 1 时

I²C0n 操作使能之后，不管总线的实际状态如何，总是立即认为处于总线释放状态（IICFn.IICBSYn 位 = 0）。若要产生首个开始条件（IICFn.STTn 位 = 1），必需确认总线已经被释放，以免干扰其它通信。

- (3) 在和其它设备进行通信时，当 V850ES/JJ3 的 IICFn.IICEn 位被设置为 1，根据通信线的状态，可能会检测到开始条件。当 SCL0n 和 SDA0n 信号线是高电平时，请确保将 IICFn.IICEn 位设置为 1。

- (4) 使能操作（IICFn.IICEn 位 = 1）之前，通过 IICCLn，IICXn 和 OCKSm 寄存器决定工作时钟频率。要改变操作时钟频率，要将 IICFn.IICEn 位清 0 一次。

- (5) 当 IICFn.STTn 和 IICFn.SPTn 位已经被设置为 1，若不首先将其清 0，则一定不要对其再次置位。

- (6) 如果传输已经被保留，设置 IICFn.SPIEn 位为 1，这样，检测到停止条件就会产生一个中断请求信号。当产生一个中断请求信号后，向 I²Cn 写入通信数据将会释放等待状态，然后传输开始。如果检测到停止条件后不产生中断请求信号，由于未产生中断请求，传输将暂停在等待状态。但是，没有必要为了软件对 IICFn.MSTSnn 位的检测而将 SPIEn 位置为 1。

备注 n = 0 至 2
 m = 0, 1

17.16 通信操作

以下展示 3 种操作过程的流程图。

(1) 主设备工作于单主系统

当使用 V850ES/JJ3 单片机作为单主系统中的主设备时，流程图如下所示。

流程图大体可以分为初始设置和通信处理两部分。在启动时执行初始设置。如果需要和从设备进行通信，进行通信准备后，执行通信处理。

(2) 主设备工作于多主系统

在 I²C0n 总线多主系统中，当总线参与通信之后，不能由 I²C 总线规范判定总线是释放还是被使用。这里，当数据线和时钟线都在一定的时期（1 帧）内保持为高电平，V850ES/JJ3 就认为总线已释放并可以进行通信。

操作流程可以大体分为初始设置、通信等待和通信处理。V850ES/JJ3 在仲裁中失败并被指定为从设备的处理情况这里予以省略，而只是展示作为主设备的处理操作。启动时执行初始设置以便参与通信。然后，作为主设备等待通信请求或作为从设备等待指定。在通信处理中执行实际的通信，且支持与从设备的发送/接收操作和与其它主设备间的仲裁处理。

(3) 从设备操作

当 V850ES/JJ3 用作 I²C0n 总线的从设备时的示例如下。

当用作从设备时，由一个中断启动操作。在启动时执行初始设置，然后，等待 INTIICn 中断信号的发生（通信等待）。当 INTIICn 中断发生时，通信状态被判定，并且判定结果作为一个标志传送给主处理程序。

通过检查该标志，可以执行必需的通信处理。

备注 n = 0 至 2

17.16.2 主设备工作于多主系统

图 17-19. 多主系统中的主设备操作 (1/3)

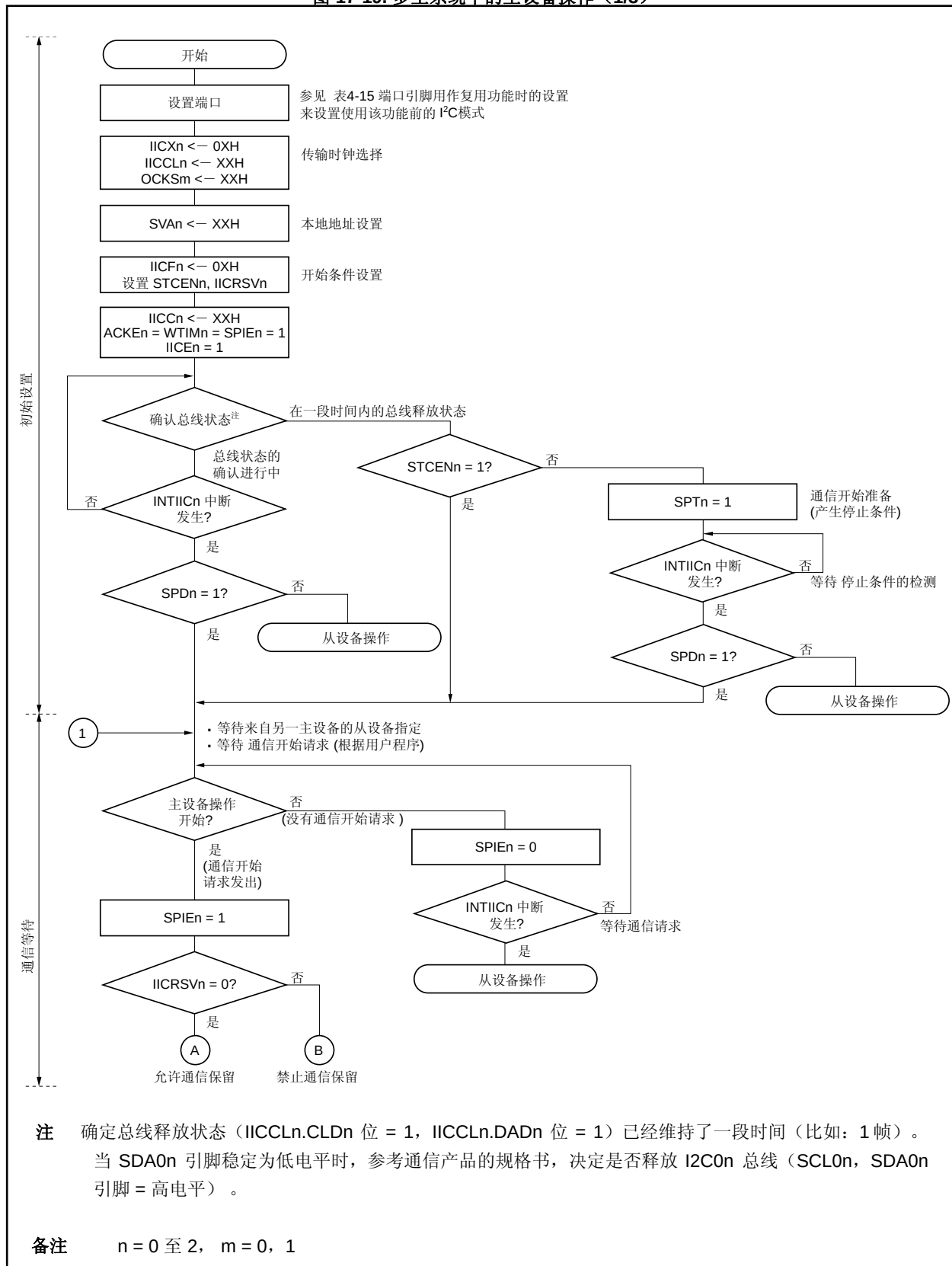


图 17-19. 多主系统中的主设备操作 (2/3)

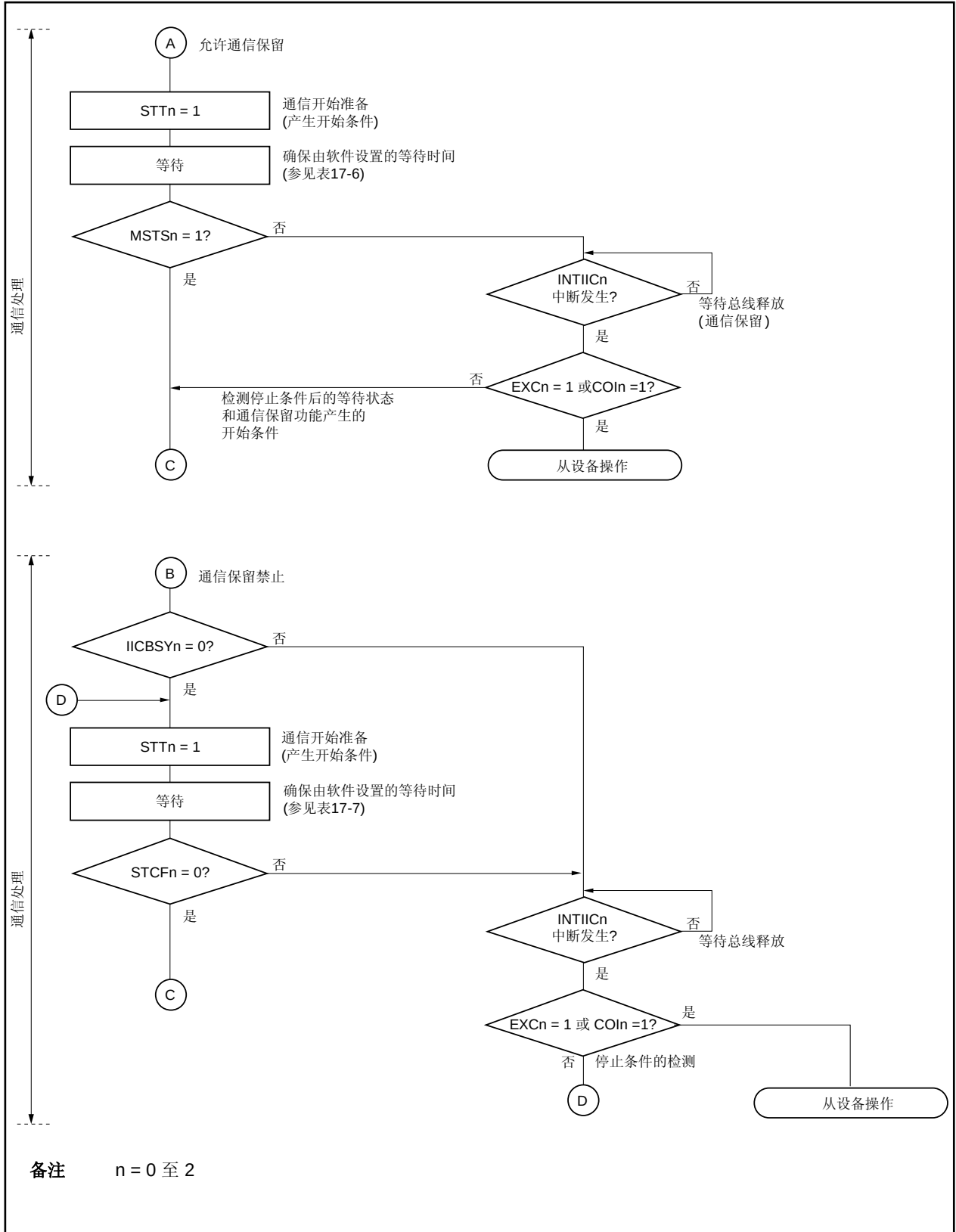
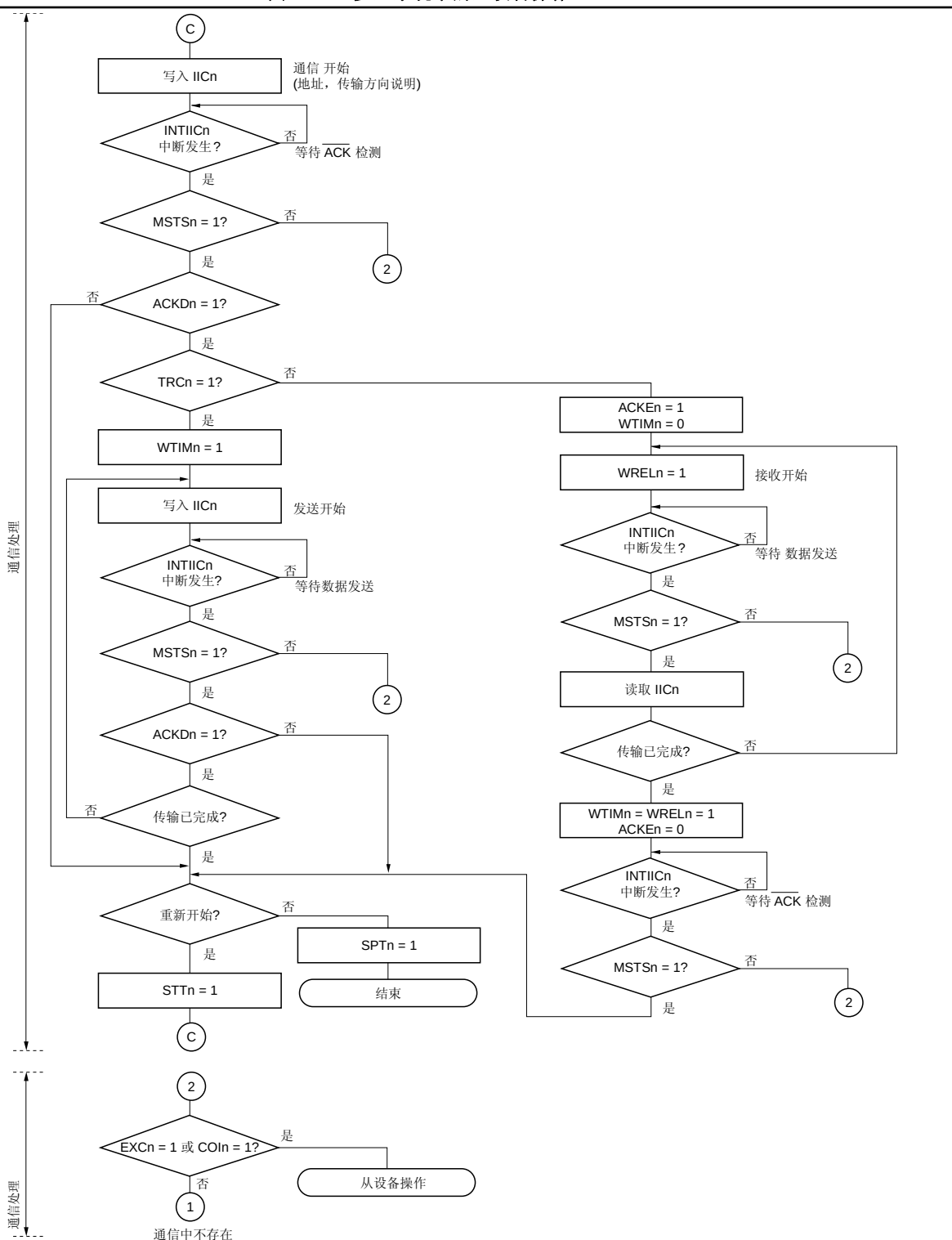


图 17-19. 多主系统中的主设备操作 (3/3)



备注

1. 确认发送和接收格式与参与通信的产品规格一致。
2. 当 V850ES/JJ3 单片机在多主系统中用作主设备时，每次发生 INTIICn 中断时都要读取 IICSn.MSTSn 位以确定仲裁结果。
3. 当 V850ES/JJ3 单片机在多主系统中用作从设备时，每次发生 INTIICn 中断时都要使用 IICSn 和 IICFn 寄存器以决定下次处理的状态。
4. n = 0 至 2

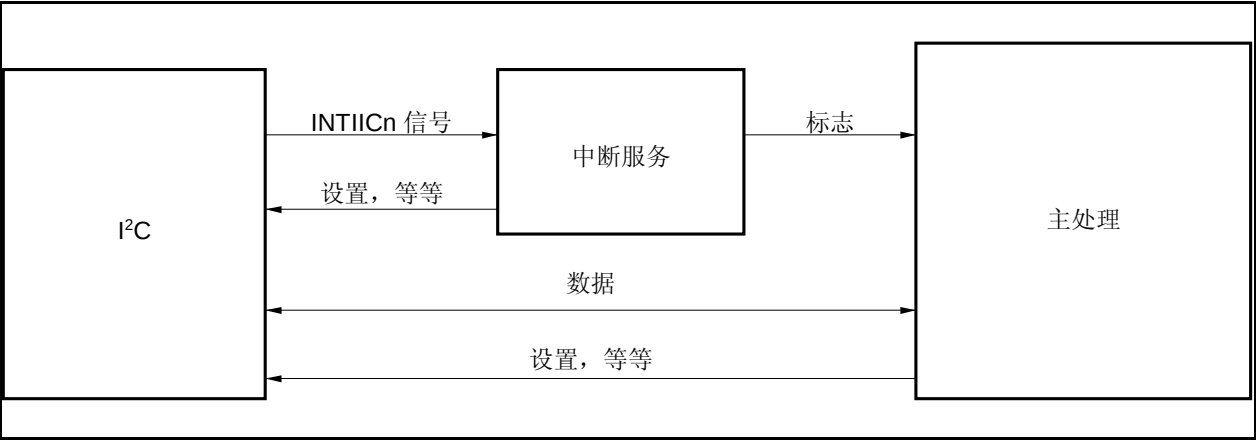
17.16.3 从设备操作

以下展示从设备操作的处理过程。

基本而言，从设备的操作都是由事件驱动的，所以，必需通过 INTIICn 中断进行处理（处理需要操作状态的重大改变，比如，通信过程中检测到停止条件）。

以下描述假定数据通信不支持扩展码，还假设只有在状态改变处理时才执行 INTIICn 中断服务，并且在主程序处理期间执行实际的数据通信。

图 17-20. 从设备操作期间的软件概况



所以，系统准备了以下 3 个标志，以便将其传送给主程序，从而执行数据传输处理，而不使用 INTIICn 信号。

(1) 通信模式标志

该标志表示以下通信状态：

清除模式：	未进行数据通信
通信模式：	正在进行数据通信（检测到有效地址，检测到停止条件，未检测到来自主设备的 $\overline{\text{ACK}}$ 以及地址不匹配）

(2) 就绪标志

该标志表示使能数据通信。这和正常数据传输期间发生 INTIICn 中断的状态相同。该标志在中断处理模块内被置位，在主程序模块中被清除。第一个数据发送的就绪标志并不在中断处理模块内置位，所以，第一个数据发送后不需要清除处理（地址匹配被视为下一个数据的请求）。

(3) 通信方向标志

该标志表示数据通信的方向，和 IICSn.TRCn 位的值相同。

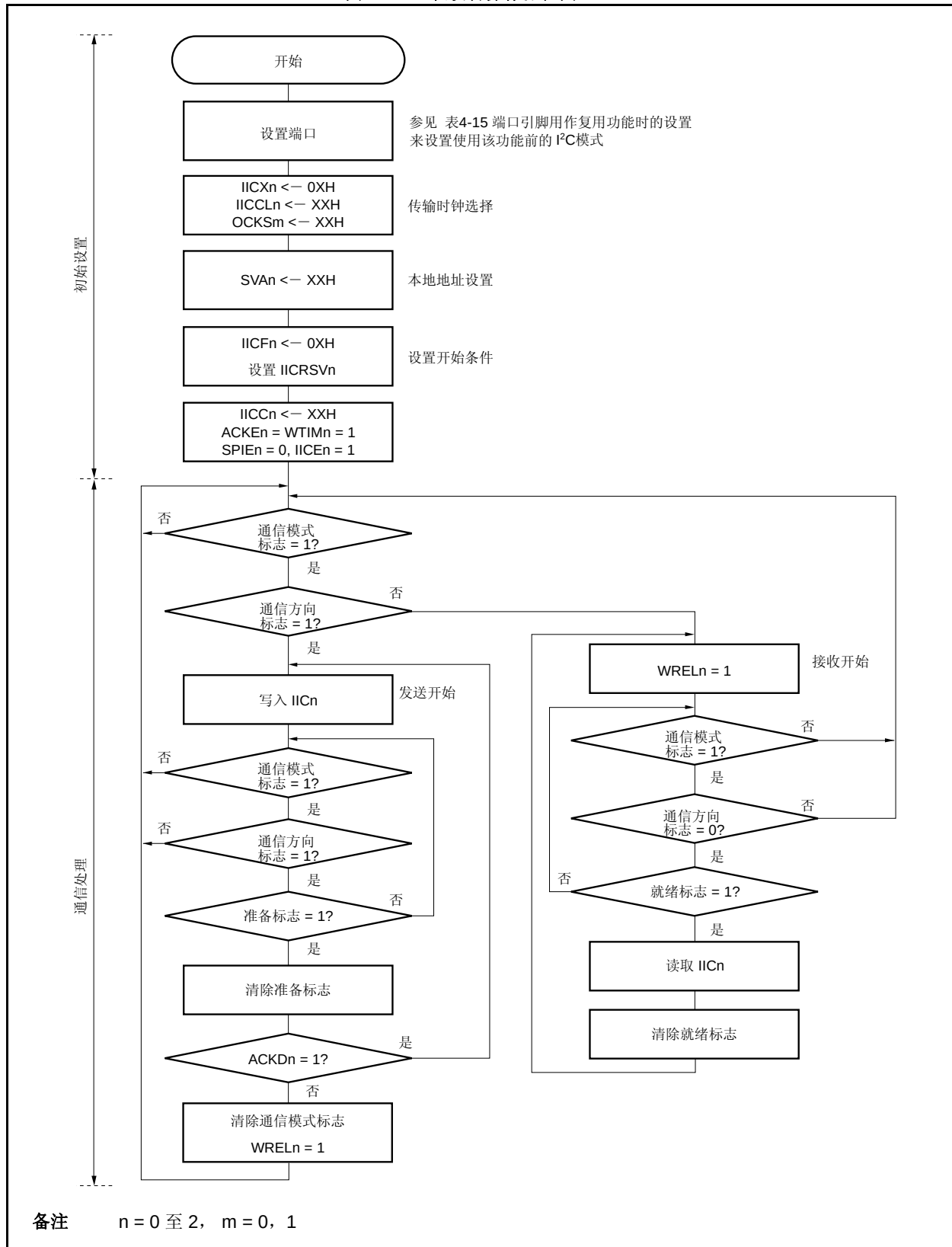
下面展示了从设备操作期间主程序模块的操作。

启动 I²C0n 并且等待通信使能状态。当通信被使能后，使用通信模式标志和就绪标志执行传输（停止条件和开始条件的处理由中断处理执行，条件通过标志位确认）。

对于发送操作，重复发送直到主设备停止返回 $\overline{\text{ACK}}$ 信号，当主设备停止返回 $\overline{\text{ACK}}$ 信号时，传输完毕。

对于接收操作，接收指定数量的数据，且在传输完成后，不为下一个数据立即返回 $\overline{\text{ACK}}$ 信号。此后，主设备产生停止条件或重新开始条件。这将导致退出通信。

图 17-21. 从设备操作流程图中 (1)

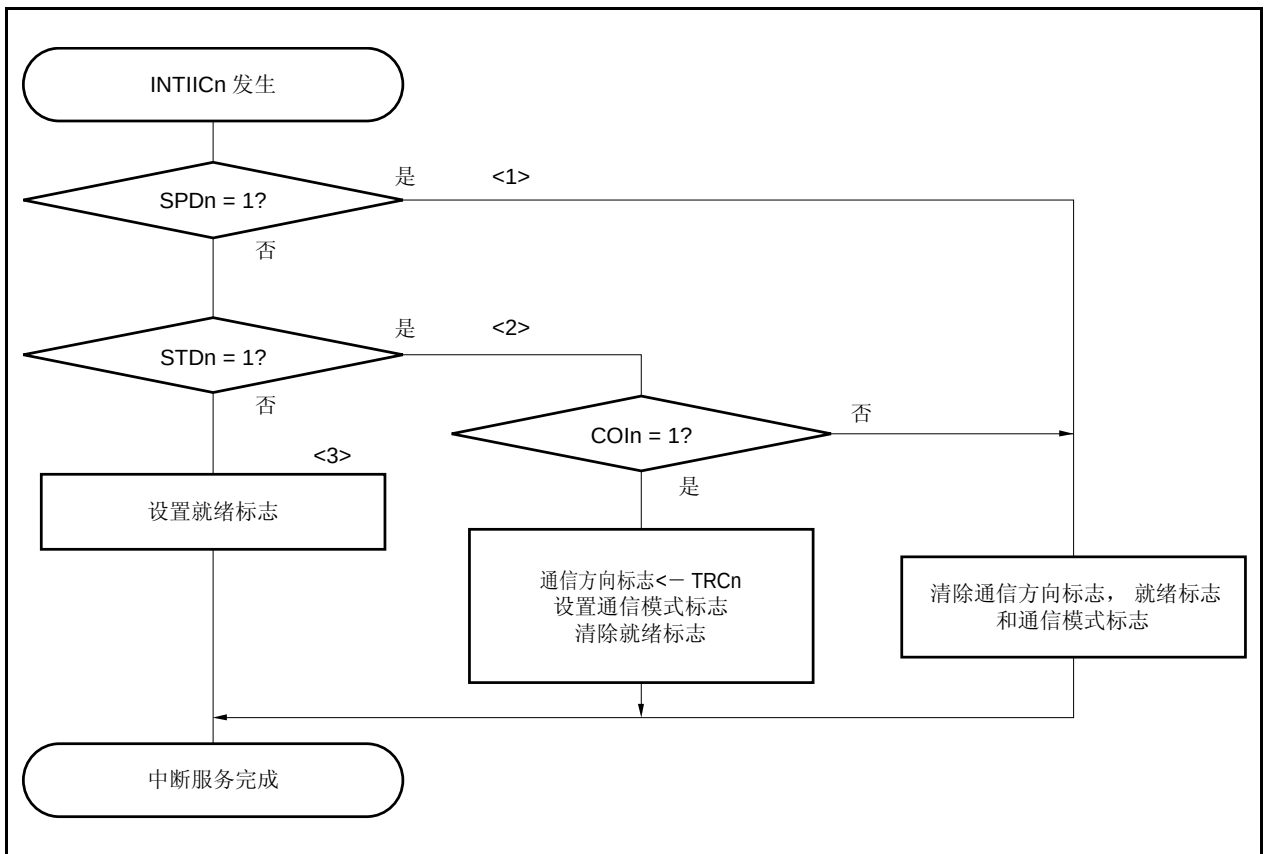


下面展示了从设备使用 INTIICn 中断进行处理的一个示例（这里假定不使用扩展码）。在 INTIICn 中断期间，状态得以确认并且执行下列步骤。

- <1> 当检测到停止条件时，通信终止。
- <2> 当检测到开始条件后，进行地址确认。如果地址不匹配，则通信终止。如果地址匹配，设置通信模式并且释放等待状态，从中断中返回操作处理（就绪标志被清除）。
- <3> 对于数据的发送/接收过程，当就绪标志置位时，I²C0n总线保持为等待状态，从中断中返回操作处理。

备注 上述<1>至<3>步骤对应图 17-22 从设备操作流程（2）的<1>至<3>步骤。

图 17-22. 从设备操作流程（2）



17.17 数据通信的时序

当使用 I²C 总线模式时，主设备通过串行总线输出一个地址，在多个从设备中选择其中一个作为自己的通信伙伴。
输出从设备地址以后，主设备发送 IICSn.TRCn 位指定数据传输方向，然后，开始与从设备进行串行通信。

IICn 寄存器的移位操作和串行时钟引脚（SCL0n）信号的下降沿同步。发送数据被传送到 SO 锁存器并经由 SDA0n 引脚输出（MSB 先行）。

由 SDA0n 引脚输入的数据在 SCL0n 引脚信号的上升沿被 IICn 寄存器捕获。
数据通信时序如下所示。

备注 n = 0 至 2

图 17-23. 主设备向从设备传输示例
(当主设备和从设备都选择 9 时钟等待时) (1/3)

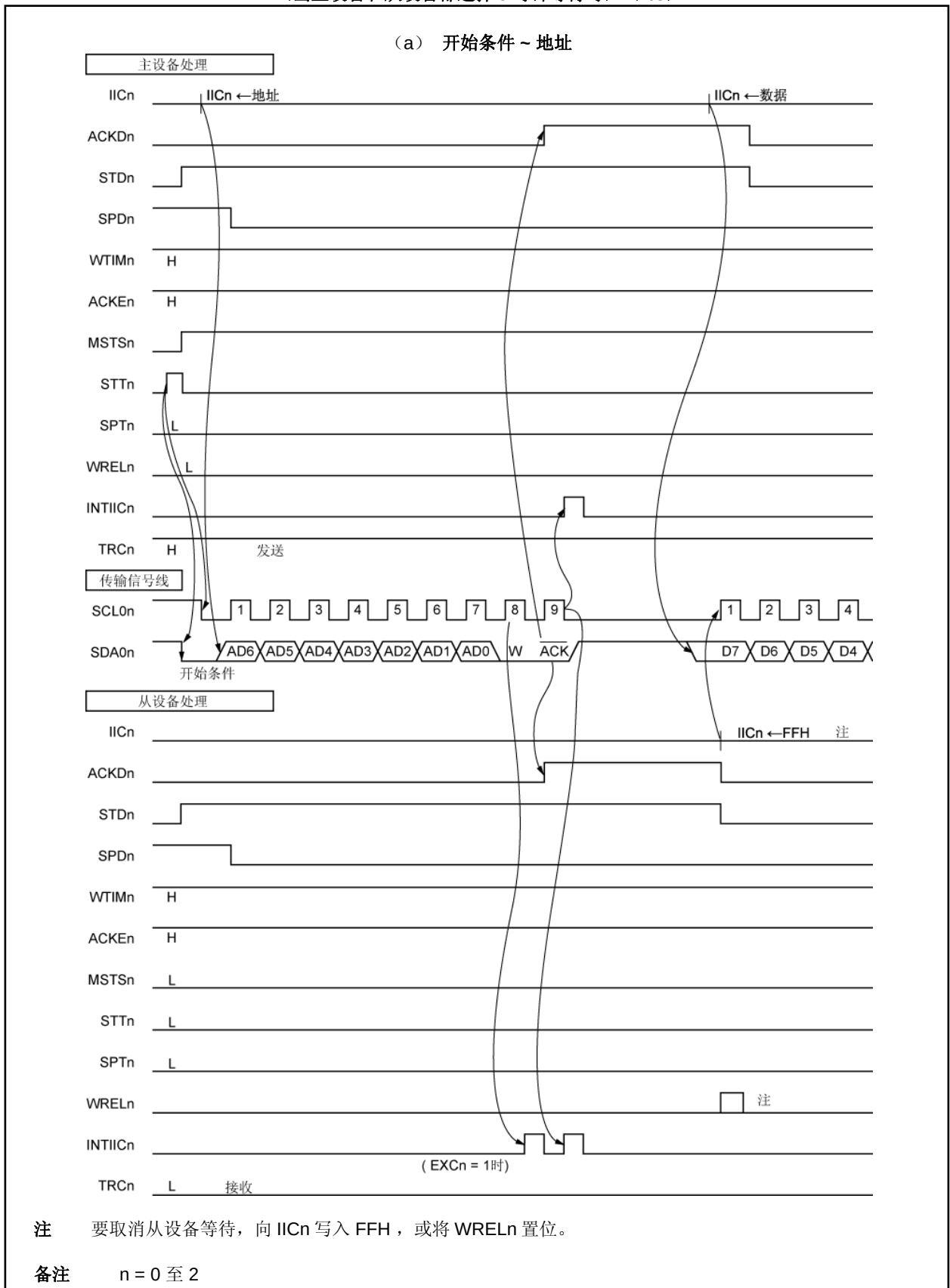
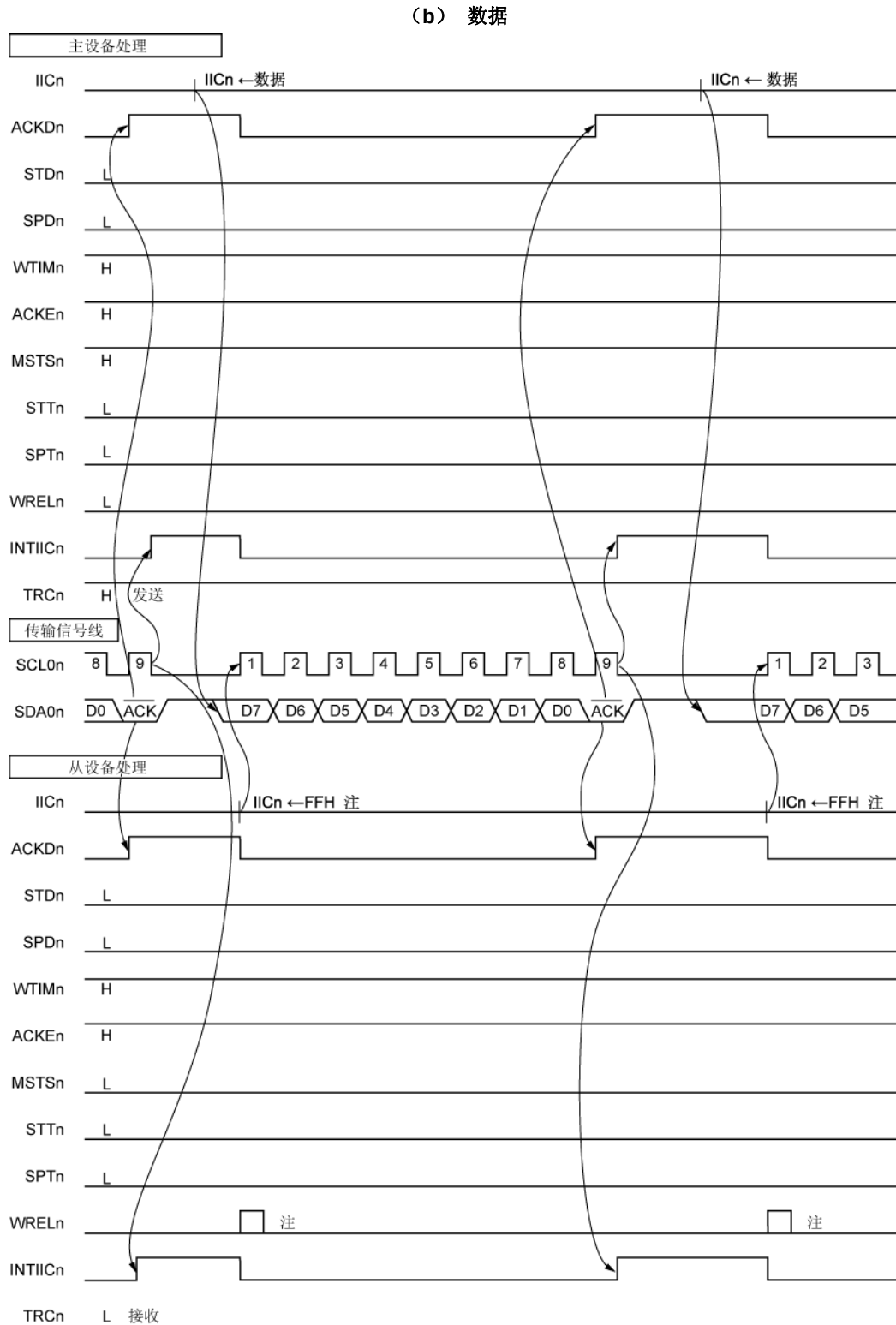


图 17-23. 主设备向从设备传输示例
(当主设备和从设备都选择 9 时钟等待时) (2/3)



注 要取消从设备等待，向 IICn 写入 FFH，或将 WRELn 置位。

备注 n = 0 至 2

图 17-23. 主设备向从设备传输示例
(当主设备和从设备都选择 9 时钟等待时) (3/3)

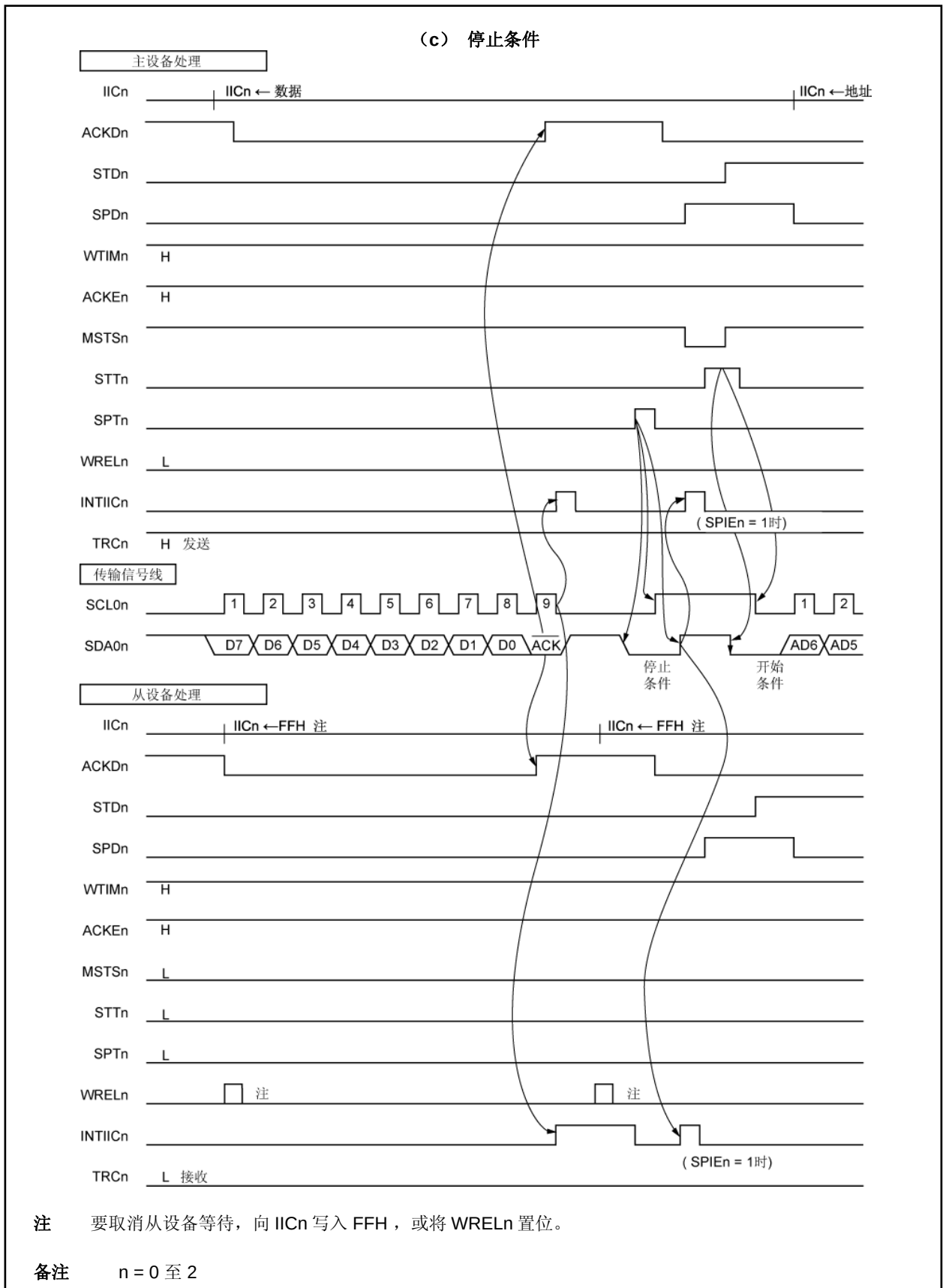
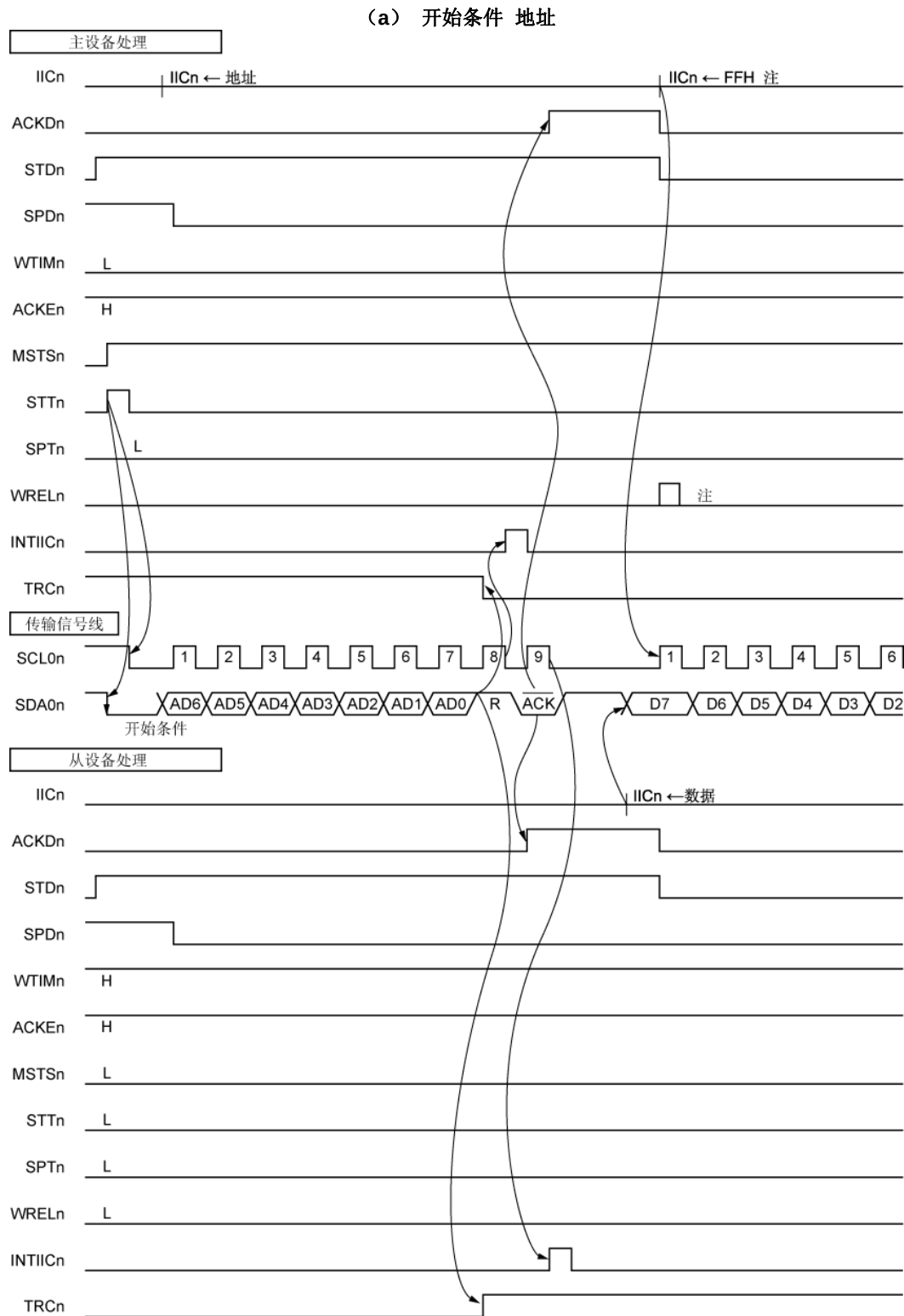


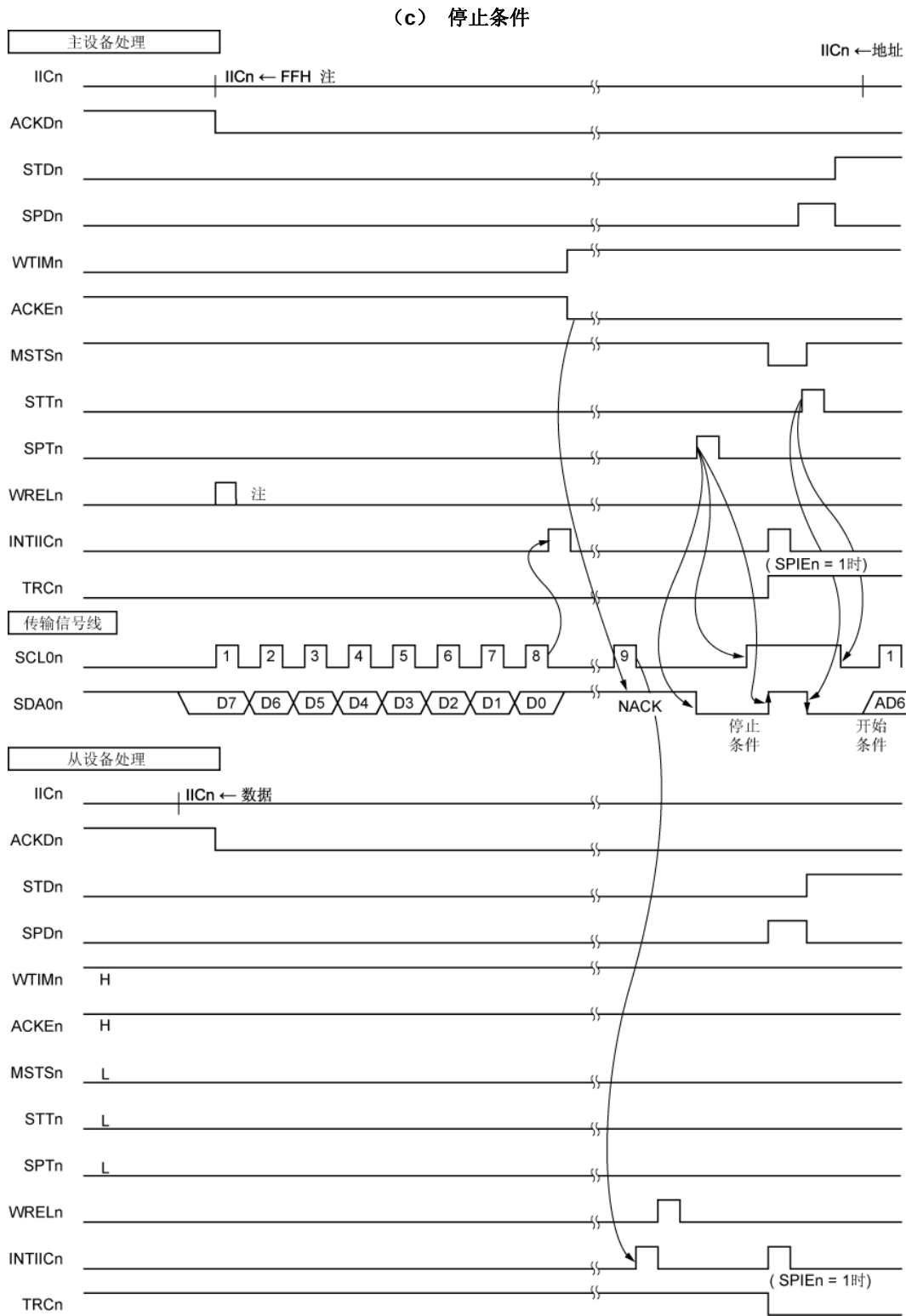
图 17-24. 从设备向主设备传输示例
(选择主设备 8 时钟等待, 从设备 9 时钟等待) (1/3)



注 要取消主设备等待, 向 IICn 写入 FFH, 或将 WRELn 置位。

备注 n = 0 至 2

图 17-24. 从设备向主设备传输示例
(选择主设备 8 时钟等待, 从设备 9 时钟等待) (3/3)



注 要取消主设备等待, 向 IICn 写入 FFH, 或将 WRELn 置位。

备注 n = 0 至 2

第十八章 DMA 功能（DMA 控制器）

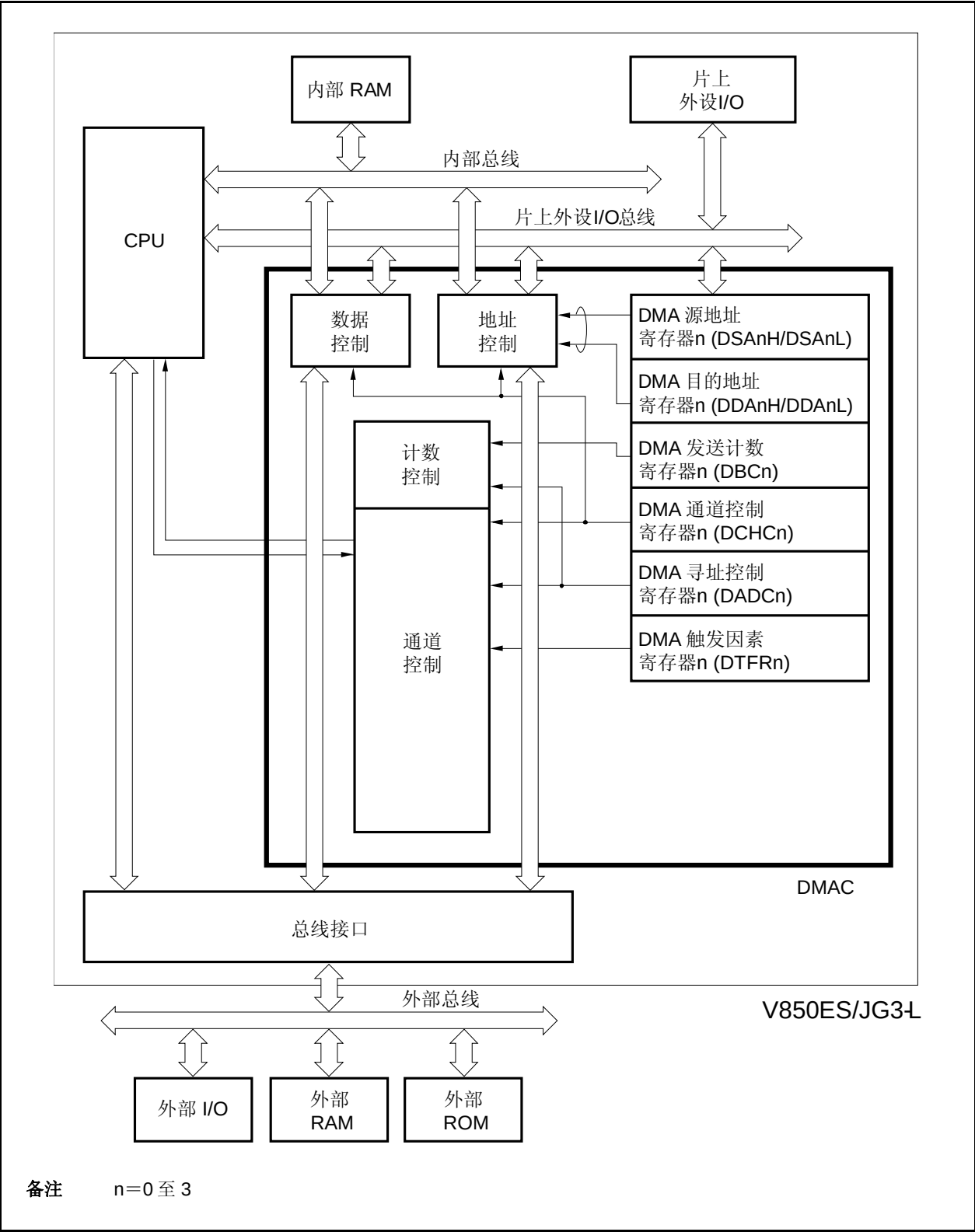
V850ES/JJ3 单片机包括一个直接存储器访问（DMA）控制器（DMAC），用于执行和控制 DMA 传输。

根据片上外设 I/O（串行接口、定时器/计数器和 A/D 转换器），外部输入引脚的中断或软件触发发出的 DMA 请求，DMAC 控制存储器与 I/O 之间、存储器之间或 I/O 之间的数据传输。（存储器是指内部 RAM 或外部存储器）

18.1 特性

- 4 路独立 DMA 通道
- 传输单位：8/16 位
- 最大传输计数：65,536 (2^{16})
- 传输类型：双周期传输
- 传输模式：单次传输模式
- 传输请求：
 - 来自片上外设 I/O（串行接口、定时器/计数器和 A/D 转换器）或来自外部输入引脚中断的请求。
 - 来自软件触发的请求。
- 传输对象：
 - 内部 RAM ↔ 外围 I/O
 - 外围 I/O ↔ 外围 I/O
 - 内部 RAM ↔ 外部存储器
 - 外部存储器 ↔ 外围 I/O
 - 外部存储器 ↔ 外部存储器

18.2 配置



18.3 寄存器

(1) DMA 源地址寄存器 0 至 3（DSA0 至 DSA3）

DSA0 至 DSA3 寄存器用于为 DMA 通道 n 设置 DMA 源地址（每个 26 位）（n=0 至 3）。

这些寄存器被分为两个 16 位寄存器，即 DSA_nH 和 DSA_nL。

这些寄存器可以按 16 位宽度读取或写入。

复位后: 不确定 R/W 地址: DSA0H 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
DSA _n H (n = 0 至 3)	IR	0	0	0	0	0	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSA _n L (n = 0 至 3)	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0

IR	DMA 传送源分类
0	外部存储或片上外设 I/O
1	内部 RAM

SA25 至 SA16	设定 DMA 传送源地址 (A25 至 A16) (缺省值未定义)。 DMA 传送时, 保持下一个 DMA 传送源地址。 当 DMA 传送完成后, 保持最初设定的 DMA 地址。
-------------	--

SA15 至 SA0	设定 DMA 传送源地址 (A15 至 A0) (缺省值未定义)。 DMA 传送时, 保持下一个 DMA 传送源地址。 当 DMA 传送完成后, 保持最初设定的 DMA 地址。
------------	---

- 注意事项**
1. 请确保将 DSA_nH 寄存器的第 14 位至第 10 位清为“0”。
 2. 当 DMA 传输禁止（DCHCn.Enn 位=0）时，在以下时序设置 DSA_nH 和 DSA_nL 寄存器。
 - 从复位后到第一次 DMA 传输启动期间
 - 从通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间
 - 从 DMA 传输完成后（DCHCn.TCn 位=1）到下一次 DMA 传输启动期间
 3. 当 DSA_n 寄存器的值被读取时，两个 16 位寄存器，即 DSA_nH 和 DSA_nL 都被读出。如果读取与更新产生冲突，则可能是对正在被更新的值进行读取（请参见 18.13 “注意事项”）。
 4. 系统复位后，在启动 DMA 发送之前，设置 DSA_nH，DSA_nL，DDA_nH，DDA_nL 和 DBCn 寄存器。如果没有设置这些寄存器，DMA 传输启动后的操作无法保证。

(2) DMA 目的地址寄存器 0 至 3（DDA0 至 DDA3）

DDA0 至 DDA3 寄存器用于为 DMA 通道 n 设置 DMA 目的地址（每个 26 位）（n=0 至 3）。

这些寄存器被分为两个 16 位寄存器，即 DDAnH 和 DDAnL。

这些寄存器可以按 16 位宽度读取或写入。

复位后: 不确定 R/W 地址: DDA0H FFFFF086H, DDA1H FFFFF08EH,
DDA2H FFFFF096H, DDA3H FFFFF09EH,
DDA0L FFFFF084H, DDA1L FFFFF08CH,
DDA2L FFFFF094H, DDA3L FFFFF09CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDAnH (n = 0 至 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 至 3)	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

IR	DMA 传送目的分类
0	外部存储器或片上外设 I/O
1	内部 RAM

DA25 至 DA16	设定一个 DMA 传送目的地 (A25 至 A16) (缺省值不确定)。 DMA 传送期间, 保持下一个 DMA 传送目的地址。 当 DMA 传送完成时, 保持最初设定的 DMA 地址。
-------------	--

DA15 至 DA0	设定一个 DMA 传送目的地 (A15 至 A0) (缺省值不确定)。 DMA 传送期间, 保持下一个 DMA 传送目的地址。 当 DMA 传送完成时, 保持最初设定的 DMA 地址。
------------	---

- 注意事项**
1. 请确保将 DSAnH 寄存器的第 14 位至第 10 位清为“0”。
 2. 当 DMA 传输禁止（DCHCn.Enn 位=0）时，在以下时序设置 DDAnH 和 DDAnL 寄存器。
 - 从复位后到第一次 DMA 传输启动期间。
 - 从通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间。
 - 从 DMA 传输完成后（DCHCn.TCn 位=1）到下一次 DMA 传输启动期间。
 3. 当 DDAn 寄存器的值被读取时，两个 16 位寄存器，即 DDAnH 和 DDAnL 被读出。如果读取与更新产生冲突，则可能是对正在被更新的值进行读取（请参见 18.13 “注意事项”）。
 4. 系统复位后，在开始 DMA 发送之前，设置 DSAnH, DSAnL, DDAnH, DDAnL 和 DBCn 寄存器。如果没有设置这些寄存器，DMA 传输启动后的操作无法保证。

(3) DMA 传输计数寄存器 0 至 3 (DBC0 至 DBC3)

DBC0 至 DBC3 寄存器是 16 位寄存器，用于设置 DMA 通道 n 的字节传输计数 (n=0 至 3)。

在 DMA 传输过程中，这些寄存器保存剩余的传输计数。

无论传输数据的单位 (8/16 位)，每传输一次，这些寄存器就递减 1，当发生借位时，传输终止。

这些寄存器可以按 16 位宽度读取或写入。

复位后: 不确定 R/W 地址: DBC0 FFFFF0C0H, DBC1 FFFFF0C2H,
DBC2 FFFFF0C4H, DBC3 FFFFF0C6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBCn (n = 0 至 3)	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0

BC15至 BC0	传送计数设定 或 DMA传送时保持传送计数
0000H	传送计数1或保持传送计数
0001H	传送计数2或保持传送计数
:	:
FFFFH	传送计数 65,536 (2^{16})，或保持传送计数
DMA传送完成时，保存最初设定的传送数据数目。	

注意事项 1. 当 DMA 传输禁止 (DCHCn.Enn 位=0) 时，在下列时序设置 DBCn 寄存器。

- 从复位后到第一次 DMA 传输启动期间
 - 从通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间
 - 从 DMA 传输完成后 (DCHCn.TCn 位=1) 到下一次 DMA 传输启动期间
- 2.** 系统复位后，在开始 DMA 发送之前，设置 DSA nH, DSA nL, DDA nH, DDA nL 和 DBCn 寄存器。如果没有设置这些寄存器，DMA 传输启动之后的操作无法保证。

（4）DMA 寻址控制寄存器 0 至 3（DADC0 至 DADC3）

DADC0 至 DADC3 寄存器是 16 位寄存器，用于控制 DMA 通道 n 的 DMA 传输模式（n=0 至 3）。

这些寄存器可以按 16 位宽度读取或写入。

系统复位后，这些寄存器被设置为 0000H。

复位后: 0000H R/W 地址: DADC0 FFFFF0D0H, DADC1 FFFFF0D2H,
DADC2 FFFFF0D4H, DADC3 FFFFF0D6H

	15	14	13	12	11	10	9	8
DADCn (n = 0 至 3)	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	发送数据宽度的设置
0	8 位
1	16 位

SAD1	SAD0	发送源地址计数方向的设置
0	0	增加
0	1	减少
1	0	固定
1	1	禁止设置

DAD1	DAD0	目的地址计数方向的设置
0	0	增加
0	1	减少
1	0	固定
1	1	禁止设置

- 注意事项**
1. 请确保将 DADCn 寄存器的第 15 位、第 13 位至第 8 位，以及第 3 位至第 0 位清为“0”。
 2. 当 DMA 传输禁止（DCHCn.Enn 位=0）时，在下列时序设置 DADCn 寄存器。
 - 从复位后到第一次 DMA 传输启动期间。
 - 从通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间。
 - 从 DMA 传输完成后（DCHCn.TCn 位=1）到下一次 DMA 传输启动期间。
 3. DS0 位指定传输数据的宽度，但是无法控制总线的宽度。如果设置了 8 位数据长度（DS0 位=0），则不使用低位数据总线。
 4. 如果传输数据宽度被设置为 16 位（DS0 位=1），则不能从奇地址开始传输。总是从低位地址的第一位为 0 的地址启动传输。
 5. 如果 DMA 传输是在一个片上外设 I/O 寄存器（作为传输源或目的地）上执行的，请确保指定的传输宽度与寄存器宽度相同。例如，要在一个 8 位寄存器上执行 DMA 传输，务必指定 8 位宽度传输方式。

(5) DMA 通道控制寄存器 0 至 3 (DCHC0 至 DCHC3)

DCHC0 至 DCHC3 寄存器是 8 位寄存器，用于控制 DMA 通道 n 的 DMA 传输操作模式。

该寄存器可进行字节读写或按位读写。（但是，第 7 位是只读的，而第 1 位和第 2 位是只写的。如果读取第 1 位或第 2 位，则读出的值将始终是 0。）

系统复位后，该寄存器被设为 00H。

复位后: 00H R/W 地址: DCHC0 FFFF0E0H, DCHC1 FFFF0E2H,
DCHC2 FFFF0E4H, DCHC3 FFFF0E6H

	<7>	6	5	4	3	<2>	<1>	<0>
DCHCn (n = 0至3)	TCn ^{注1}	0	0	0	0	INITn ^{注2}	STGn ^{注2}	Enn

TCn ^{注1}	状态标志指示DMA通道n的DMA传输是否完成
0	DMA传送未完成。
1	DMA传送已完成。
DMA传送的最后设定为1，当读取时清为0。	

INITn ^{注2}	DMA传送禁止(Enn位= 0)时，如果INITn位设定为1， 可以初始化DMA传送状态。 在DMA传送完成(TCn位设定为1)之前，重新设定DMA传送 状态(重新设定DDAnH, DDAnL, DSAnH, DSAnL, DBCn, 以及 DADCn寄存器)，确保初始化DMA通道。 当对DMA控制器进行初始化时，务必阅读18.13 注意事项 中的步骤描述。
---------------------	--

STGn ^{注2}	这是DMA传送的软件启动触发信号。 如果在DMA传送允许状态(TCn 位 = 0, Enn 位 = 1)时，该位被置为1， DMA传送启动。
--------------------	--

Enn	设置是否允许DMA通道n的DMA传输
0	DMA传送禁止
1	DMA传送允许
当Enn位被设置为1时，DMA传送允许。 当DMA传送完成时(产生一个终端计数)，该位自动清为0。 通过将Enn位清为0，中断DMA传送。如要恢复，则重新将Enn位设置为1。 当终止或恢复DMA传送时，严格遵守18.13 注意事项 中的步骤描述。	

注 1. TCn 位只读。

2. INITn 和 STGn 位只写。

注意事项 1. 请确保将 DCHCn 寄存器的第 6 位至第 3 位清为“0”。

2. DMA 传输完成后（当产生末尾计数时），Enn 位被清零，然后 TCn 位被置 1。如果正在更新 DCHCn 寄存器同时发生读取操作，可能会读出一个值表示“传输未完成且传输被禁止”（TCn 位 = 0 且 Enn 位 = 0）。

（6）DMA 触发因素寄存器 0 至 3（DTFR0 至 DTFR3）

DTFR0 至 DTFR3 寄存器是 8 位寄存器，控制片上外设 I/O 的中断请求信号产生的 DMA 传输启动触发源。

这些寄存器设置的中断请求信号用作 DMA 传输启动因素。

这些寄存器可以按字节读取或写入。但是，DFn 位可以按位进行读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H R/W 地址: DTFR0 FFFFF810H, DTFR1 FFFFF812H,
DTFR2 FFFFF814H, DTFR3 FFFFF816H

DTFRn	<7>							0
	DFn	0	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0

(n = 0 至 3)

DFn ^注	DMA 传送请求状态标志
0	无DMA 传送请求
1	DMA 传送请求

注 不要通过软件设置 DFn 位为 1。当 DMA 传输禁止时，如果发生了一个被指定作为启动 DMA 传输的触发原因，则将 0 写入该位，以清除 DMA 传输请求。

- 注意事项** 1. 当 DMA 传输禁止（DCHCn.Enn 位=0）时，在下列时序设置 IFCn5 至 IFCn0 位。
- 从复位后到第一次 DMA 传输启动期间。
 - 从通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间。
 - 从 DMA 传输完成后（DCHCn.TCn 位=1）到下一次 DMA 传输启动期间。
2. 在待机模式（IDLE1、IDLE2、STOP 或 sub-IDLE 模式）中产生的中断请求信号不能启动 DMA 传输周期（DFn 位也不会置 1）。
3. 通过 IFCn5 至 IFCn0 位选定 DMA 启动因素，当选定的片上外设 I/O 产生中断时，不论 DMA 发送操作是允许还是禁止，DFn 位都会置 1。如果这种状态下 DMA 是使能的，则 DMA 传输立即启动。

备注 关于 IFCn5 至 IFCn0 位，请参见表 18-1 DMA 启动因素。

表 18-1. DMA 启动因素 (1/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	中断源
0	0	0	0	0	0	由中断产生的 DMA 请求被禁止
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

备注 n=0 至 3

表 18-1. DMA 启动因素（2/2）

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	中断源
1	0	1	0	1	1	INTUA2T
1	0	1	1	0	0	INTAD
1	0	1	1	0	1	INTKR
1	1	0	0	0	1	INTP8
1	1	0	0	1	0	INTTP6CC0
1	1	0	0	1	1	INTTP6CC1
1	1	0	1	0	0	INTTP7CC0
1	1	0	1	0	1	INTTP7CC1
1	1	0	1	1	0	INTTP8CC0
1	1	0	1	1	1	INTTP8CC1
1	1	1	0	0	0	INTCB5R
1	1	1	0	0	1	INTCB5T
1	1	1	0	1	0	INTUA3R
1	1	1	0	1	1	INTUA3T
其它情况						禁止设置

备注 n=0 至 3

18.4 传输目标

表 18-2 说明了传输目标之间的关系（√：允许传输，×：禁止传输）。

表 18-2 传输目标之间的关系

		传输目的地			
		内部 ROM	片上外设 I/O	内部 RAM	外部存储器
传输源	片上外设 I/O	×	√	√	√
	内部 RAM	×	√	×	√
	外部存储器	×	√	√	√
	内部 ROM	×	×	×	×

注意事项 在表 18-2 中，用“×”标记的传输目的地和传输源组合，其操作将无法保证。

18.5 传输模式

支持单次传输模式。

在单次传输模式下，在每次字节/半字传输时总线都被释放。如果有后续的 DMA 传输请求，则再执行一次传输。该操作一直进行下去，直到发生一个终止计数。

当 DMAC 释放总线后，如果发生另一个更高优先级的 DMA 传输请求，则较高优先级的 DMA 请求将被优先处理。

在一个传输周期中，如果产生了同一通道的新的传输请求和另一个低优先级通道的传输请求，则当总线释放给 CPU 后，执行具有低优先级的通道的 DMA 传输。（同一通道的新的传输请求在传输周期中被忽略）

18.6 传输类型

作为传输模式，可以支持双周期传输。

在双周期传输中，数据传输在两个周期中完成，即读周期和写周期。

在读周期中，输出传输源地址，并执行从源地址向 DMAC 的读操作。在写周期中，输出传输目的地址，并执行从 DMAC 向目的地的写操作。

在读周期和写周期之间总是插入一个时钟的空闲周期。在进行双周期的 DMA 传输时，如果传输源与目的地的数据总线宽度不同，则执行以下操作。

<16 位数据传输>

<1> 从 32 位总线 → 16 位总线传输

先产生一个读周期（高 16 位处于高阻状态），接着产生一个写周期（16 位）。

<2> 从 16/32 位总线向 8 位总线传输

产生一次 16 位读周期，然后产生两次 8 位写周期。

<3> 从 8 位总线向 16/32 位总线传输

先产生两次 8 位读周期，然后产生一次 16 位写周期。

<4> 16 位总线与 32 位总线之间传输

产生一次 16 位读周期，然后产生一次 16 位写周期。

对于在片上外设 I/O 寄存器（传输源或目的地）上执行的 DMA 传输，请确保指定的传输宽度与寄存器宽度相同。例如，要在一个 8 位寄存器上执行 DMA 传输，务必指定 8 位宽度传输方式。

备注 每个传输目标（传输源/目的地）的总线宽度如下：

- 片上外设 I/O: 16 位总线宽度
- 内部 RAM: 32 位总线宽度
- 外部存储器: 8 位或 16 位总线宽度

18.7 DMA 通道优先级

DMA 通道优先级固定如下所示：

DMA 通道 0 > DMA 通道 1 > DMA 通道 2 > DMA 通道 3

每个传输周期都要检查这些优先级。

18.8 与 DMA 传输有关的时间

响应 DMA 请求所需的时间、DMA 传输需要的最少时钟数如下所示：

单次传输： DMA响应时间（<1>）+ 传输源存储器访问（<2>）+ 1^{#1} + 传输目的地存储器访问（<2>）

DMA 周期		最少执行时钟数
<1> DMA 请求响应时间		4 时钟（最小值）+ 噪声消除时间 ^{#2}
<2> 存储器访问	外部存储器访问	取决于连接的存储器
	内部 RAM 访问	2 时钟 ^{#3}
	外设 I/O 寄存器访问	3 时钟 + VSWC 寄存器设定的等待周期数 ^{#4}

- 注
- 1. DMA 传输时，在读周期和写周期之间总是插入一个时钟。
 - 2. 如果指定一个外部中断（INTPn）作为启动 DMA 传输的触发信号，则增加噪声消除时间（n=0 至 8）。
 - 3. 一个 DMA 周期需要两个时钟。
 - 4. 访问某个特定的外设 I/O 寄存器，需要更多等待周期（详细情况，请参见 3.4.8（2））。

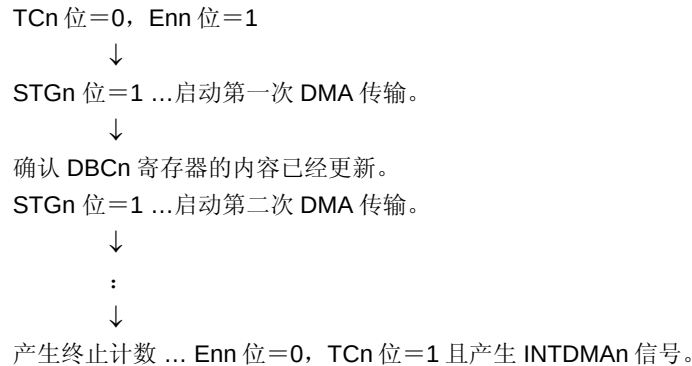
18.9 DMA 传输启动因素

DMA 传输启动因素有两种，介绍如下：

(1) 软件请求

当 DCHCn.TCn 位=1 且 Enn 位= 1（允许 DMA 传输）时，如果将 STGn 位设置为 1，则启动 DMA 传输。

若在此之后立刻请求下一个 DMA 传输周期，则应使用 DBCn 寄存器，确认前一个 DMA 传输周期已经完成，并再次将 STGn 位设置为 1（n=0 至 3）。



(2) 片上外设 I/O 请求

当 DCHCn.TCn 位=0 且 Enn 位=1（允许 DMA 传输）时，如果从 DTFRn 寄存器设置的片上外设 I/O 产生一个中断请求，则 DMA 传输启动。

- 注意事项**
1. 两个启动因素（软件触发和硬件触发）不能用于一个 DMA 通道。如果对一个 DMA 通道同时产生两个启动因素，则只有其中一个有效。且不能确认有效启动因素。
 2. 在前一个 DMA 传输请求产生后或在前一个 DMA 传输周期中，产生的新传输请求将被忽略（清除）。
 3. 同一个 DMA 通道的传输请求时间间隔取决于 DMA 传输周期总线等待的设置、其他通道的开始状态或外部总线保持请求也会影响该间隔。特别如注意事项 2 所述，在 DMA 传输周期前或在 DMA 传输周期中，同一个通道产生的新的传输请求将被忽略。因此，同一个 DMA 通道的传输请求时间间隔必须足够长。当使用软件触发时，可以通过更新 DBCn 寄存器来检查之前的 DMA 传输周期完成情况。

18.10 DMA 中止因素

如果发生总线保持, 则 DMA 传输将中止。

如果在内部存储器/片上外设 I/O 和内部存储器//片上外设 I/O 之间传输, 同样会发生 DMA 传输中止。

当总线保持被清除, 则 DMA 传输将恢复。

18.11 DMA 传输结束

当 DMA 传输已经完成 DBCn 寄存器设置的次数, 当 DCHCn.Enn 位被清零、TCn 位被置 1 后, 中断控制器 (INTC) 将会发现一个 DMA 传输结束中断请求信号 (INTDMA_n) (n=0 至 3)。

V850ES/JJ3 单片机不向外部器件输出终止计数信号。因此, 通过 DMA 传输结束中断或检测 TCn 位来确认 DMA 传输的完成情况。

18.12 操作时序

图 18-1 至图 18-4 显示了 DMA 的操作时序。

图 18-1. DMA 的优先级 (1)

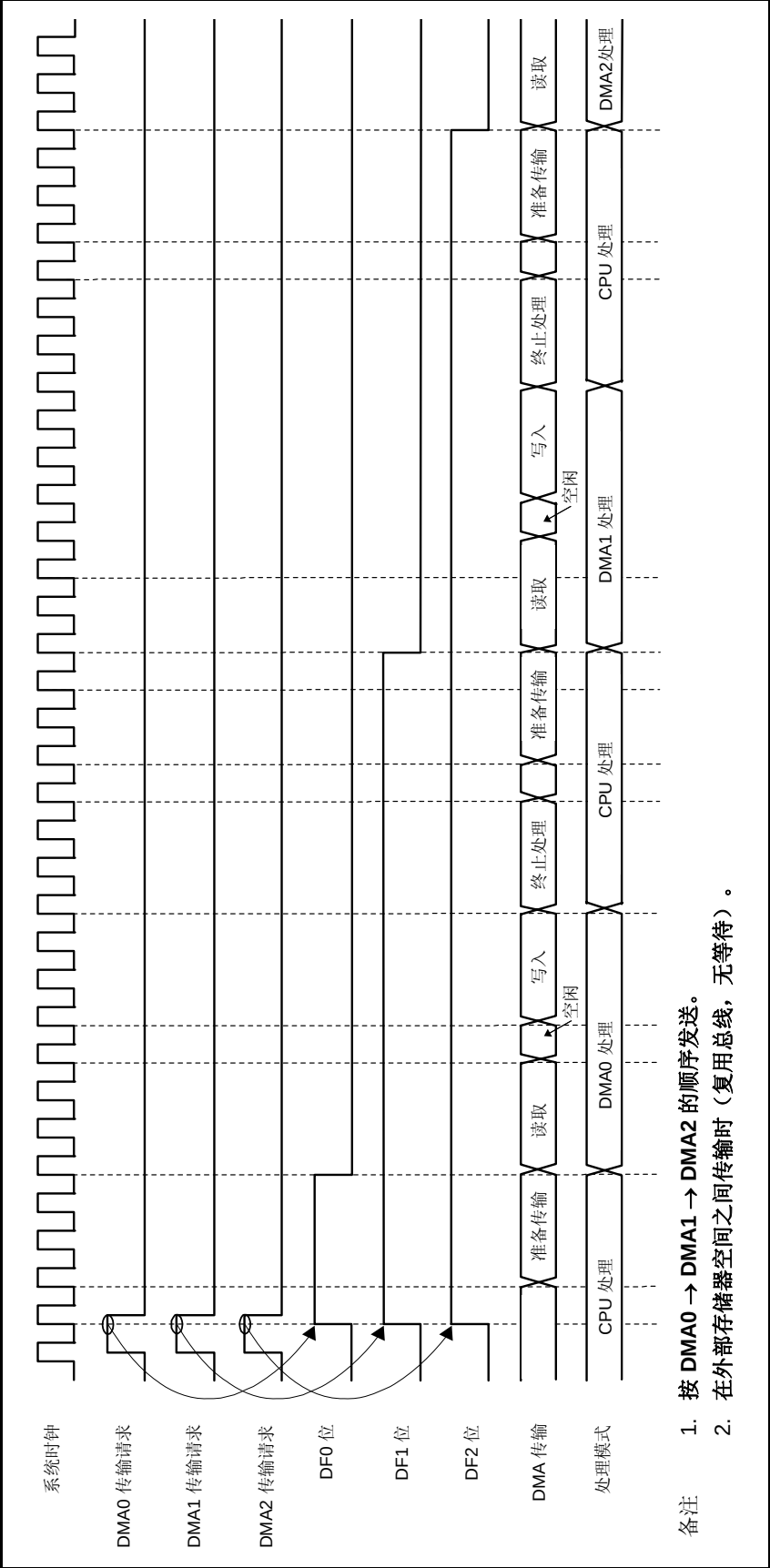


图 18-2. DMA 的优先级 (2)

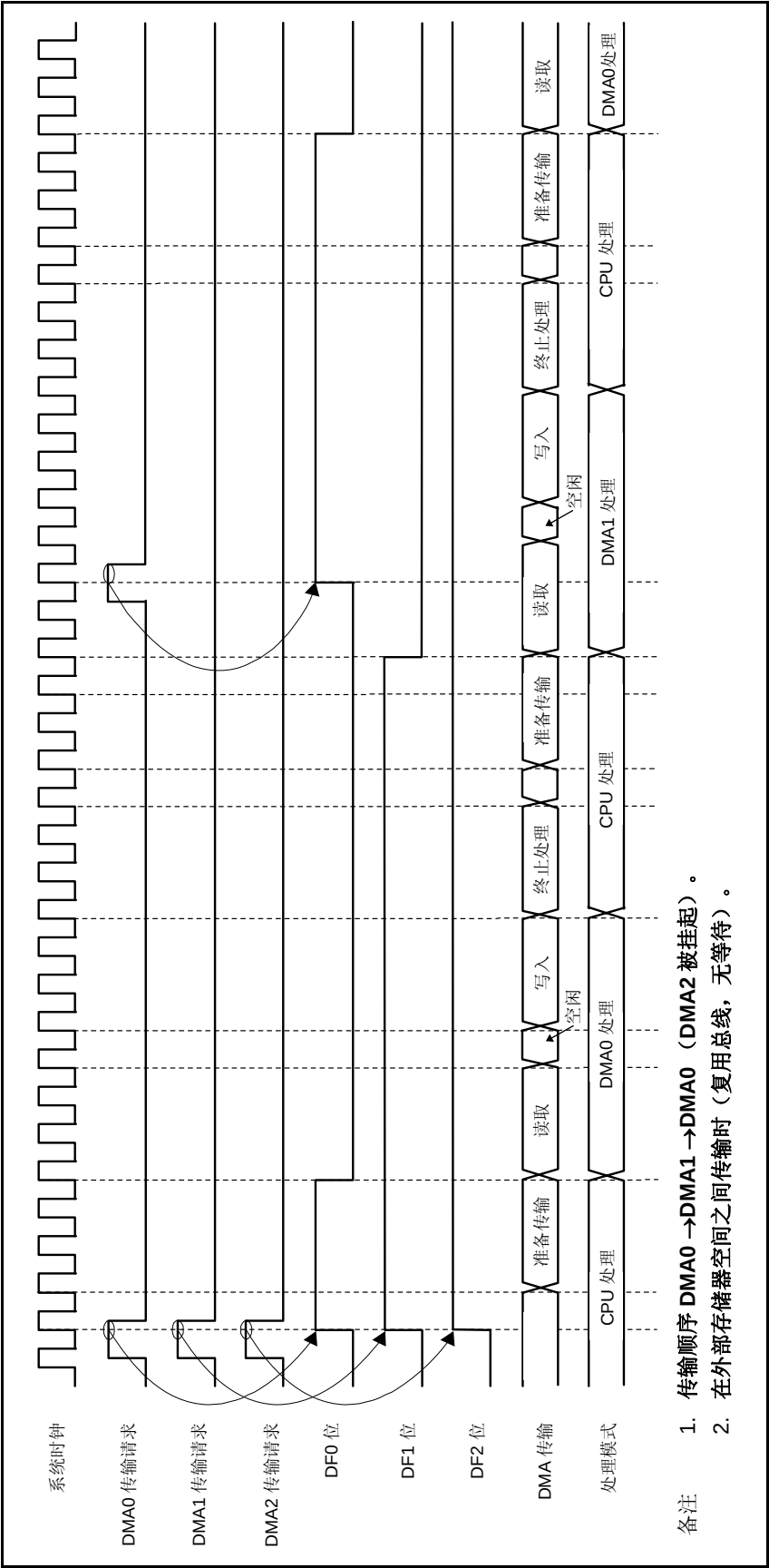


图 18-3 DMA 传输请求被忽略的周期（1）

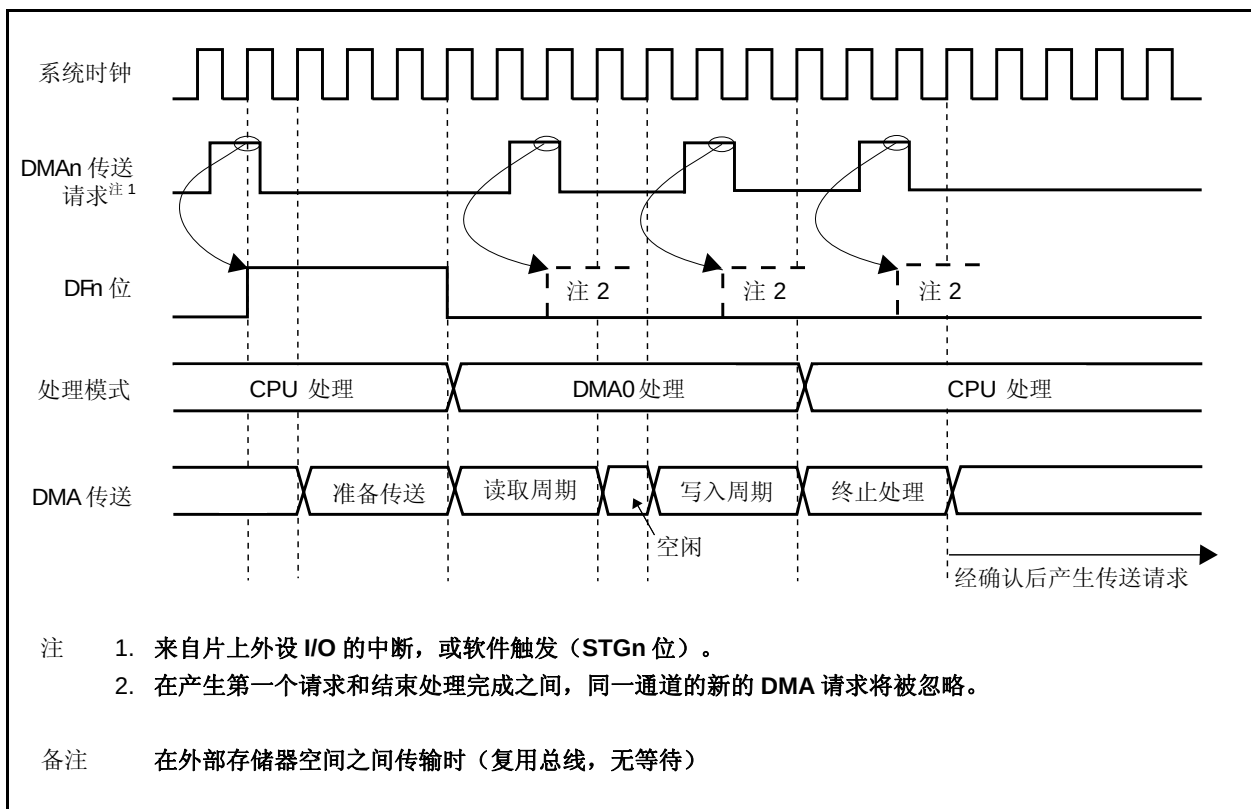
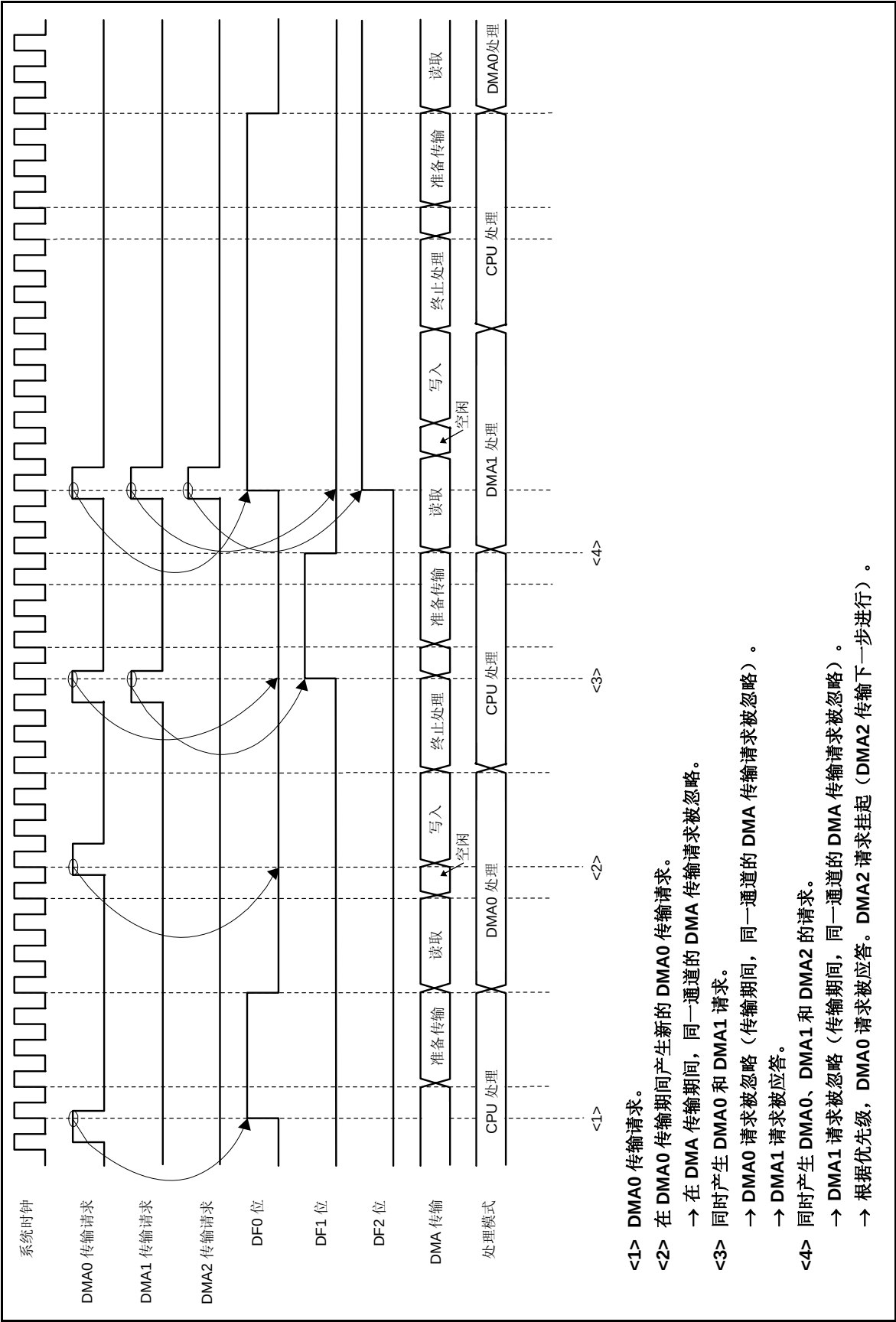


图 18-4 DMA 传输请求被忽略的周期（2）



18.13 注意事项

(1) VSWC 寄存器注意事项

在使用 DMAC 时，确保根据工作频率为 VSWC 寄存器设置一个合适的值。

如果使用了 VSWC 寄存器的默认值 (77H)，或者给 VSWC 寄存器设置了一个不适当的值，则操作将不能正确执行 (关于寄存器的详细情况，请参见 3.4.8 (1) (a) 系统等待控制寄存器 (VSWC))。

(2) 在内部 RAM 中执行 DMA 传输的注意事项

在内部 RAM 中执行以下指令时，不要执行与内部 RAM (传输源/目的地) 之间进行数据交互的 DMA 传输，因为 CPU 随后可能会发生错误。

- 位于内部 RAM 中的位操作指令 (SET1, CLR1, 或 NOT1)。
- 访问内部 RAM 中非对齐地址的数据访问指令。

反之，当 DMA 传输的执行与内部 RAM (传输源/目的地) 之间数据交互时，不要执行以上两条指令。

(3) 读 DCHCn.TCn 位 (n=0 至 3) 的注意事项

当读取 TCn 位时，TCn 位被清零。但是它不会自动清零，即使在一个特定时刻被读取也不会清零。要准确清零 TCn 位，请增加以下处理操作：

(a) 通过轮询 TCn 位等待 DMA 传输完成时

确认 TCn 位已经被设置为 1 (在读取 TCn 位=1 后)，然后再读 TCn 位三次以上。

(b) 在断服务例程中的读取 TCn 位时

执行 TCn 位的读取操作三次。

(4) DMA 传输初始化方法 (将 DCHCn.INITn 位设置为 1)

当执行 DMA 传输的通道要进行初始化时, 即使 INITn 位被置为 1, 通道可能也无法初始化。要准确地初始化通道, 执行以下两种方法之一。

<R>

(a) 暂时停止所有 DMA 通道的传输

使用以下<1>至<7>步, 对执行 DMA 传输的通道进行初始化。

但是, 请注意, 当执行第<5>步时, TCn 位将被清零。确保其他处理程序不需要 TCn 位为 1。

<1> 禁止中断 (DI)。

<2> 除了被强制终止的 DMA 通道之外, 读取所有通道的 DCHCn.Enn 位, 然后将值传输到通用寄存器。

<3> 清除使用的 DMA 通道 (包括被强制终止的通道) 的 Enn 位为 0。执行两次清除指令来清除最后一个 DMA 通道的 Enn 位。如果 DMA 传输目标 (传输源/目的地) 涉及到内部 RAM, 则执行三次清除指令。

举例: 如果通道 0, 1, 2 被使用, 则按以下顺序执行指令 (如果传输目标不是内部 RAM)。

- 写入 DCHC0 = 00H (将 E00 位清除为 0)
- 写入 DCHC1 = 00H (将 E11 位清除为 0)
- 写入 DCHC2 = 00H (将 E22 位清除为 0)
- 再次写入 DCHC2 = 00H (将 E22 位清除为 0)

<4> 对强制终止的通道, 向其写入 DCHCn = 04 (将 INITn 位设置为 1)。

<5> 读取所有未被强制终止的通道的 TCn 位。如果 TCn 位和第<2>步读取的 Enn 位均为 1 (逻辑与 (AND) 结果是 1), 将已保存的 Enn 位清零。

<6> 第<5>步的操作完成后, 将 Enn 位的值写入 DCHCn 寄存器。

<7> 使能中断 (EI)。

注意事项 1. 确保执行上述步骤中的第<5>步, 以防止对某通道 Enn 位的非法设置, 该通道的 DMA 传输在第<2>步和第<3>步之间正常完成。

2. 当使用位操作指令时, 步骤<3>和<4> (Enn 位清除 (0) 和 INITn 位置位 (1)) 被禁止, 因为 TCn 位被清除为 0。

(b) 重复执行 INITn 位设置，直到传输被正确地强制终止

- <1> 如果该通道将要被强制终止（停止片上外设 I/O 的操作），则对来自该通道 DMA 请求源的请求不作响应。
- <2> 使用 DTFRn.DFn 位，确认待强制终止的通道的 DMA 传输请求未处于挂起状态。如果 DMA 传输请求被挂起，等待，直到挂起请求执行完成。
- <3> 已经确认待强制终止的通道的 DMA 传输请求未处于挂起状态，则将 Enn 位清零。
- <4> 再次将待强制终止的通道的 Enn 位清零。
如果待强制终止的通道的传输目标（传输源/目的地）是内部 RAM，则再次执行该操作。
- <5> 将最初设定给待强制终止的通道的初始传输数目拷贝到通用寄存器。
- <6> 将待强制终止的通道的 INITn 位置为 1。
- <7> 读取待强制终止的通道的 DBCn 寄存器的值，然后将其与在第<5>步中的拷贝值进行比较。如果两个值不匹配，重复操作第<6>和<7>步。

- 备注**
- 1. 在第<7>步中读取 DBCn 寄存器的值时，如果正确完成强制终止，则读取获得初始传输数目。否则，读取到的值是传输的剩余数目。
 - 2. 注意，如果频繁地使用待强制终止的 DMA 通道之外的通道进行 DMA 传输，则方法 (b) 会消耗较长时间。

(5) 暂时停止 DMA 传输的步骤（清除 Enn 位）

使用以下步骤对操作状态下的 DMA 传输进行停止和恢复：

- <1> 对来自 DMA 请求源的传输请求（停止片上外设 I/O 的操作）不作响应。
- <2> 使用 DFn 位（检查 DFn 位是否等于 0），确认 DMA 传输请求未处于挂起状态。
如果请求处于挂起状态，等待，直到挂起 DMA 传输的请求执行完成。
- <3> 如果确认没有 DMA 传输请求处于挂起状态，将 Enn 位清零（该操作停止 DMA 传输）。
- <4> 将 Enn 位设置为 1，以恢复 DMA 传输。
- <5> 对已经停止的 DMA 请求源的操作进行恢复（开始片上外设 I/O 的操作）。

(6) 存储边界

在 DMA 传输过程中，如果传输源或目的地的地址超出了 DMA 对象（外部存储器、内部 RAM 或片上外设 I/O）的区域，则操作将无法保证。

(7) 传输非对齐的数据

不支持采用 16 位总线宽度进行非对齐数据的 DMA 传输。

如果一个奇地址被指定为传输源或目的地，则地址的最低有效位被强制假定为 0。

<R> (8) CPU 的总线仲裁

因为 DMA 控制器拥有比 CPU 更高优先级的总线权限，在 DMA 传输期间，发生的 CPU 访问将保持挂起状态直到 DMA 传输周期完成，总线被释放给 CPU 为止。

但是，CPU 可以访问未参与 DMA 传输的内部 ROM 和内部 RAM。

- 在外部存储器与片上外设 I/O 之间执行 DMA 传输时，CPU 可以访问内部 ROM 和内部 RAM。
- 在片上外设 I/O 与内部 RAM 之间执行 DMA 传输时，CPU 可以访问内部 ROM。

(9) DMA 操作时不可重写的寄存器/位

当未进行 DMA 操作时，按以下时序设置下列寄存器。

[寄存器]

- DSAnH, DSAnL, DDA nH, DDA nL, DBCn 和 DADCn 寄存器。
- DTFRn.IFCn5 至 DTFRn.IFCn0 位。

[设置时序]

- 从复位后到第一次 DMA 传输启动期间。
- 从通道初始化后到 DMA 传输启动期间。
- 从 DMA 传输完成后 (TCn 位=1) 到下一次 DMA 传输启动期间。

(10) 请确保将以下寄存器的指定位设置为 0

- DSAnH 寄存器的第 14 位至第 10 位。
- DDA nH 寄存器的第 14 位至第 10 位。
- DADCn 寄存器的第 15 位，第 13 位至第 8 位和第 3 位至第 0 位。
- DCHCn 寄存器的第 6 位至第 3 位。

(11) DMA 启动因素

不要使用相同启动因素开启两个或更多 DMA 通道。如果使用相同的启动因素开启了两个或更多的 DMA 通道，那么已经设定的 DMA 通道可能会再启动，或具有较低优先级的 DMA 通道将在具有较高优先级的 DMA 通道之前被应答，操作无法保证。

(12) 读取 DSAn 和 DDAn 寄存器的值

在 DMA 传输过程中，可能对 DSAn 和 DDAn 寄存器中正在更新的值 (n=0 至 3) 进行读取。

例如，当 DMA 传输源地址 (DSAn 寄存器) 为 0000FFFFH，且计数方向是递增 (DADCn.SAD1 和 DADCn.SAD0 位=00) 时，如果先读取 DSAnH 寄存器，然后再读取 DSAnL 寄存器，则根据是否在读取 DSAnH 寄存器之后立即执行 DMA 传输，DSAnL 寄存器的值会有所不同，描述如下。

(a) 如果在读取 DSAn 寄存器时未执行 DMA 传输

<1> 读取 DSAnH 寄存器的值: DSAnH=0000H

<2> 读取 DSAnL 寄存器的值: DSAnL=FFFFH

(b) 如果在读取 DSAn 寄存器时发生 DMA 传输

<1> 读取 DSAnH 寄存器的值: DSAnH=0000H

<2> 发生 DMA 传输

<3> DSAn 寄存器递增: DSAn=00100000H

<4> 读取 DSAnL 寄存器的值: DSAnL=0000H

第十九章 中断/异常处理功能

V850ES/JJ3 为中断服务提供专用的中断控制器（INTC），可以处理总共 71 个中断请求。

所谓中断，是独立于程序执行而发生的事件，而异常则是与程序执行相关的事件。

V850ES/JJ3 能够处理来自于片上外设硬件设备和外部源的中断请求信号。不仅如此，还可以通过 TRAP 指令（软件异常）或产生一个异常事件（比如，取非法操作码）（异常陷阱）来启动异常处理。

19.1 特性

- 中断
 - 不可屏蔽中断：2 个中断源
 - 可屏蔽中断：外部：9 个中断源，内部：60 个中断源
 - 8 级可编程优先级（可屏蔽中断）
 - 依照优先级控制多重中断
 - 可以对每一个可屏蔽中断请求进行屏蔽
 - 噪声消除、边沿检测以及为外部中断请求信号指定有效边沿
- 异常
 - 软件异常：32 个中断源
 - 异常陷阱：2 中断源（非法操作码异常）

表 19-1 列出了中断/异常发生源。

表 19-1. 中断源列表 (1/3)

类型	类别	默认优先级	名称	触发	产生单元	异常码	句柄地址	被恢复的 PC	中断控制寄存器
复位	中断	–	RESET	RESET 引脚输入 由内部源引发的复位	RESET	0000H	00000000H	不确定	–
不可屏蔽	中断	–	NMI	NMI 引脚有效沿输入	引脚	0010H	00000010H	下一 PC 值	–
		–	INTWDT2	WDT2 溢出	WDT2	0020H	00000020H	注 1	–
软件异常	异常	–	TRAP0n ^{#2}	TRAP 指令	–	004nH ^{#2}	00000040H	下一 PC 值	–
		–	TRAP1n ^{#2}	TRAP 指令	–	005nH ^{#2}	00000050H	下一 PC 值	–
异常陷阱	异常	–	ILGOP/ DBG0	非法操作码 DBTRAP 指令	–	0060H	00000060H	下一 PC 值	–
可屏蔽	中断	0	INTLVI	低电压检测	POCLVI	0080H	00000080H	下一 PC 值	LVIIC
		1	INTP0	外部中断引脚输入边沿检测 (INTP0)	引脚	0090H	00000090H	下一 PC 值	PIC0
		2	INTP1	外部中断引脚输入边沿检测 (INTP1)	引脚	00A0H	000000A0H	下一 PC 值	PIC1
		3	INTP2	外部中断引脚输入边沿检测 (INTP2)	引脚	00B0H	000000B0H	下一 PC 值	PIC2
		4	INTP3	外部中断引脚输入边沿检测 (INTP3)	引脚	00C0H	000000C0H	下一 PC 值	PIC3
		5	INTP4	外部中断引脚输入边沿检测 (INTP4)	引脚	00D0H	000000D0H	下一 PC 值	PIC4
		6	INTP5	外部中断引脚输入边沿检测 (INTP5)	引脚	00E0H	000000E0H	下一 PC 值	PIC5
		7	INTP6	外部中断引脚输入边沿检测 (INTP6)	引脚	00F0H	000000F0H	下一 PC 值	PIC6
		8	INTP7	外部中断引脚输入边沿检测 (INTP7)	引脚	0100H	00000100H	下一 PC 值	PIC7
		9	INTTQ0OV	TMQ0 溢出	TMQ0	0110H	00000110H	下一 PC 值	TQ0OVIC
		10	INTTQ0CC0	TMQ0 捕获 0/比较 0 匹配	TMQ0	0120H	00000120H	下一 PC 值	TQ0CCIC0
		11	INTTQ0CC1	TMQ0 捕获 1/比较 1 匹配	TMQ0	0130H	00000130H	下一 PC 值	TQ0CCIC1
		12	INTTQ0CC2	TMQ0 捕获 2/比较 2 匹配	TMQ0	0140H	00000140H	下一 PC 值	TQ0CCIC2
		13	INTTQ0CC3	TMQ0 捕获 3/比较 3 匹配	TMQ0	0150H	00000150H	下一 PC 值	TQ0CCIC3
		14	INTTP0OV	TMP0 溢出	TMP0	0160H	00000160H	下一 PC 值	TP0OVIC
		15	INTTP0CC0	TMP0 捕获 0/比较 0 匹配	TMP0	0170H	00000170H	下一 PC 值	TP0CCIC0
		16	INTTP0CC1	TMP0 捕获 1/比较 1 匹配	TMP0	0180H	00000180H	下一 PC 值	TP0CCIC1
		17	INTTP1OV	TMP1 溢出	TMP1	0190H	00000190H	下一 PC 值	TP1OVIC
		18	INTTP1CC0	TMP1 捕获 0/比较 0 匹配	TMP1	01A0H	000001A0H	下一 PC 值	TP1CCIC0
		19	INTTP1CC1	TMP1 捕获 1/比较 1 匹配	TMP1	01B0H	000001B0H	下一 PC 值	TP1CCIC1
		20	INTTP2OV	TMP2 溢出	TMP2	01C0H	000001C0H	下一 PC 值	TP2OVIC
		21	INTTP2CC0	TMP2 捕获 0/比较 0 匹配	TMP2	01D0H	000001D0H	下一 PC 值	TP2CCIC0
		22	INTTP2CC1	TMP2 捕获 1/比较 1 匹配	TMP2	01E0H	000001E0H	下一 PC 值	TP2CCIC1
		23	INTTP3OV	TMP3 溢出	TMP3	01F0H	000001F0H	下一 PC 值	TP3OVIC
		24	INTTP3CC0	TMP3 捕获 0/比较 0 匹配	TMP3	0200H	00000200H	下一 PC 值	TP3CCIC0
		25	INTTP3CC1	TMP3 捕获 1/比较 1 匹配	TMP3	0210H	00000210H	下一 PC 值	TP3CCIC1

注 1. 关于 INTWDT2 的恢复情况, 参见 19.2.2 (2) 来自 INTWDT2 的信号。
2. n=0 至 FH

表 19-1. 中断源列表 (2/3)

类型	类别	默认优先级	名称	触发	产生单元	异常码	句柄地址	被恢复的 PC	中断控制寄存器
可屏蔽	中断	26	INTTP4OV	TMP4 溢出	TMP4	0220H	00000220H	下一 PC 值	TP4OVIC
		27	INTTP4CC0	TMP4 捕获 0/比较 0 匹配	TMP4	0230H	00000230H	下一 PC 值	TP4CCIC0
		28	INTTP4CC1	TMP4 捕获 1/比较 1 匹配	TMP4	0240H	00000240H	下一 PC 值	TP4CCIC1
		29	INTTP5OV	TMP5 溢出	TMP5	0250H	00000250H	下一 PC 值	TP5OVIC
		30	INTTP5CC0	TMP5 捕获 0/比较 0 匹配	TMP5	0260H	00000260H	下一 PC 值	TP5CCIC0
		31	INTTP5CC1	TMP5 捕获 1/比较 1 匹配	TMP5	0270H	00000270H	下一 PC 值	TP5CCIC1
		32	INTTMOEQ0	TMM0 比较匹配	TMM0	0280H	00000280H	下一 PC 值	TM0EQIC0
		33	INTCB0R/ INTIIC1	CSIB0 接收完成/ CSIB0 接收错误/ IIC1 发送完成	CSIB0/ IIC1	0290H	00000290H	下一 PC 值	CB0RIC/ IICIC1
		34	INTCB0T	CSIB0 连续发送写入使能	CSIB0	02A0H	000002A0H	下一 PC 值	CB0TIC
		35	INTCB1R	CSIB1 接收完成/ CSIB1 接收错误	CSIB1	02B0H	000002B0H	下一 PC 值	CB1RIC
		36	INTCB1T	CSIB1 连续发送写入使能	CSIB1	02C0H	000002C0H	下一 PC 值	CB1TIC
		37	INTCB2R	CSIB2 接收完成/ CSIB2 接收错误	CSIB2	02D0H	000002D0H	下一 PC 值	CB2RIC
		38	INTCB2T	CSIB2 连续发送写入使能	CSIB2	02E0H	000002E0H	下一 PC 值	CB2TIC
		39	INTCB3R	CSIB3 接收完成/ CSIB3 接收错误	CSIB3	02F0H	000002F0H	下一 PC 值	CB3RIC
		40	INTCB3T	CSIB3 连续发送写入使能	CSIB3	0300H	00000300H	下一 PC 值	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 PC 值	UA0RIC/ CB4RIC
		42	INTUA0T/ INTCB4T	UARTA0 连续发送使能/ CSIB4 连续发送写入使能	UARTA0/ CSIB4	0320H	00000320H	下一 PC 值	UA0TIC/ CB4TIC
		43	INTUA1R/ INTIIC2	UARTA1 接收完成/ UARTA1 接收错误/ IIC2 发送完成	UARTA1/ IIC2	0330H	00000330H	下一 PC 值	UA1RIC/ IICIC2
		44	INTUA1T	UARTA1 连续发送使能	UARTA1	0340H	00000340H	下一 PC 值	UA1TIC
		45	INTUA2R/ INTIIC0	UARTA2 接收完成/ IIC0 发送完成	UARTA/ IIC0	0350H	00000350H	下一 PC 值	UA2RIC/ IICIC0
		46	INTUA2T	UARTA2 连续发送使能	UARTA2	0360H	00000360H	下一 PC 值	UA2TIC
		47	INTAD	A/D 转换完成	A/D	0370H	00000370H	下一 PC 值	ADIC
		48	INTDMA0	DMA0 传输完成	DMA	0380H	00000380H	下一 PC 值	DMAIC0
		49	INTDMA1	DMA1 传输完成	DMA	0390H	00000390H	下一 PC 值	DMAIC1
		50	INTDMA2	DMA2 传输完成	DMA	03A0H	000003A0H	下一 PC 值	DMAIC2
		51	INTDMA3	DMA3 传输完成	DMA	03B0H	000003B0H	下一 PC 值	DMAIC3
		52	INTKR	按键返回中断	KR	03C0H	000003C0H	下一 PC 值	KRIC
		53	INTWTI	钟表定时器中断	WT	03D0H	000003D0H	下一 PC 值	WTIIC
		54	INTWT	钟表定时器参考时间	WT	03E0H	000003E0H	下一 PC 值	WTIC

表 19-1. 中断源列表 (3/3)

类型	类别	默认优先级	名称	触发	产生单元	异常码	句柄地址	被恢复的 PC	中断控制寄存器
可屏蔽	中断	55	INTP8	外部中断引脚输入边沿检测 (INTP8)	引脚	0470H	00000470H	nextPC	PIC8
		56	INTTP6OV	TMP6 溢出	TMP6	0480H	00000480H	nextPC	TP6OVIC
		57	INTTP6CC0	TMP6 捕获 0/比较 0 匹配	TMP6	0490H	00000490H	nextPC	TP6CCIC0
		58	INTTP6CC1	TMP6 捕获 1/比较 1 匹配	TMP6	04A0H	000004A0H	nextPC	TP6CCIC1
		59	INTTP7OV	TMP7 溢出	TMP7	04B0H	000004B0H	nextPC	TP7OVIC
		60	INTTP7CC0	TMP7 捕获 0/比较 0 匹配	TMP7	04C0H	000004C0H	nextPC	TP7CCIC0
		61	INTTP7CC1	TMP7 捕获 1/比较 1 匹配	TMP7	04D0H	000004D0H	nextPC	TP7CCIC1
		62	INTTP8OV	TMP8 溢出	TMP8	04E0H	000004E0H	nextPC	TP8OVIC
		63	INTTP8CC0	TMP8 捕获 0/比较 0 匹配	TMP8	04F0H	000004F0H	nextPC	TP8CCIC0
		64	INTTP8CC1	TMP8 捕获 1/比较 1 匹配	TMP8	0500H	00000500H	nextPC	TP8CCIC1
		65	INTCB5R	CSIB5 接收完成	CSIB5	0510H	00000510H	nextPC	CB5RIC
		66	INTCB5T	CSIB5 连续发送写入使能	CSIB5	0520H	00000520H	nextPC	CB5TIC
		67	INTUA3R	UART3 连续接收完成	UARTA3	0530H	00000530H	nextPC	UA3RIC
		68	INTUA3T	UARTA3 连续发送使能	UARTA3	0540H	00000540H	nextPC	UA3TIC

- 备注**
- 默认优先级：当两个或更多可屏蔽中断请求同时发生时的优先级次序。最高优先级是 0。
不可屏蔽中断的优先级次序是：INTWDT2>NMI。
被恢复的 PC：当中断服务开始时，程序计数器 (PC) 的值被存入 EIPC、FEPC 或 DBPC。然而，需要注意的是：在执行以下任意一条指令时，如果程序响应一个不可屏蔽中断或可屏蔽中断，则被恢复的 PC 不是下一 PC 值 (nextPC)。(如果在中断执行期间响应一个中断，执行停止，之后在中断服务完成后恢复执行。)
 - 载入指令 (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)。
 - 除法指令 (DIV, DIVH, DIVU, DIVHU)。
 - PREPARE, DISPOSE 指令 (仅当在堆栈指针被更新前产生中断)。
 下一 PC 值：在中断/异常处理之后，由此开始进行处理的 PC 值。
 - 当发生非法操作码异常时，非法指令的执行地址计算方法为：(被恢复的 PC-4)。

19.2 不可屏蔽中断

即便处于中断禁止（DI）状态，不可屏蔽中断请求信号也可以无条件地得到响应。不可屏蔽中断（NMI）并不遵从优先级控制，而且，比所有其他中断请求信号更优先。

该产品有以下两种不可屏蔽中断请求信号。

- NMI 引脚输入（NMI）。
- 看门狗定时器溢出产生的不可屏蔽中断请求信号（INTWDT2）。

NMI 引脚的有效沿可以选择为以下 4 种类型：“上升沿”、“下降沿”、“双边沿”和“无边沿检测”。

当 WDTM2.WDM21 和 WDTM2.WDM20 位被设置为“01”时，看门狗定时器 2（INTWDT2）的溢出会产生不可屏蔽中断请求信号。

当两个或两个以上不可屏蔽中断请求信号同时发生时，较高优先级的中断得到相应（低优先级的中断请求信号被忽略），如下所示：

INTWDT2 > NMI

在 NMI 正在被处理时，发生新的 NMI 或 INTWDT2 请求信号，则按以下情况处理。

（1）如果在 NMI 处理时，发生新的 NMI 请求信号

不管 PSW.NP 位的值如何，新的 NMI 请求信号都会被挂起。挂起的 NMI 请求信号在当前正在处理的 NMI 请求完毕后得到响应（在 RETI 指令执行完成之后）。

（2）如果在 NMI 处理时，发生 INTWDT2 请求信号

在 NMI 处理期间，如果 NP 位保持置位状态（1），INTWDT2 请求信号被挂起。挂起的 INTWDT2 请求信号在当前正在处理的 NMI 请求完毕后得到响应（在 RETI 指令执行完成之后）。

在 NMI 请求处理期间，如果 NP 位被清除为 0，执行新产生的 INTWDT2 请求信号（NMI 处理服务停止）。

注意事项 关于对不可屏蔽中断请求信号（INTWDT2）执行的不可屏蔽中断服务，参见 19.2.2（2）来自 INTWDT2 信号。

图 19-1. 不可屏蔽中断请求信号的应答操作（1/2）

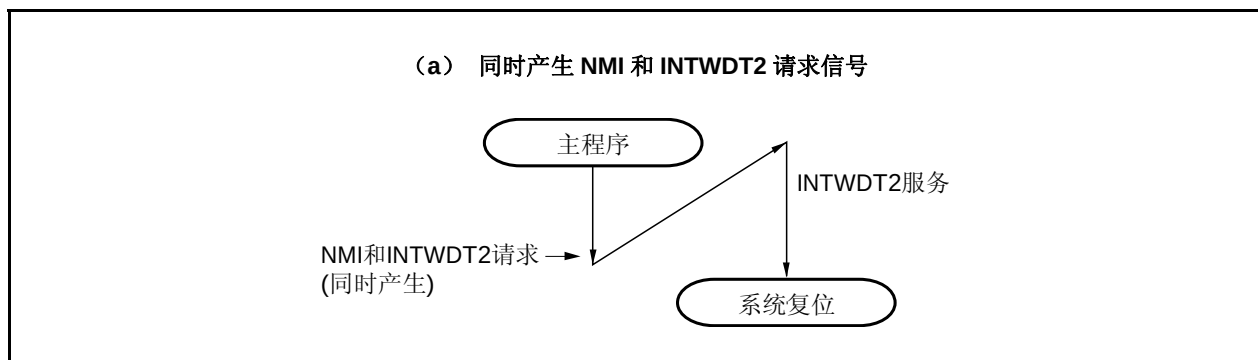
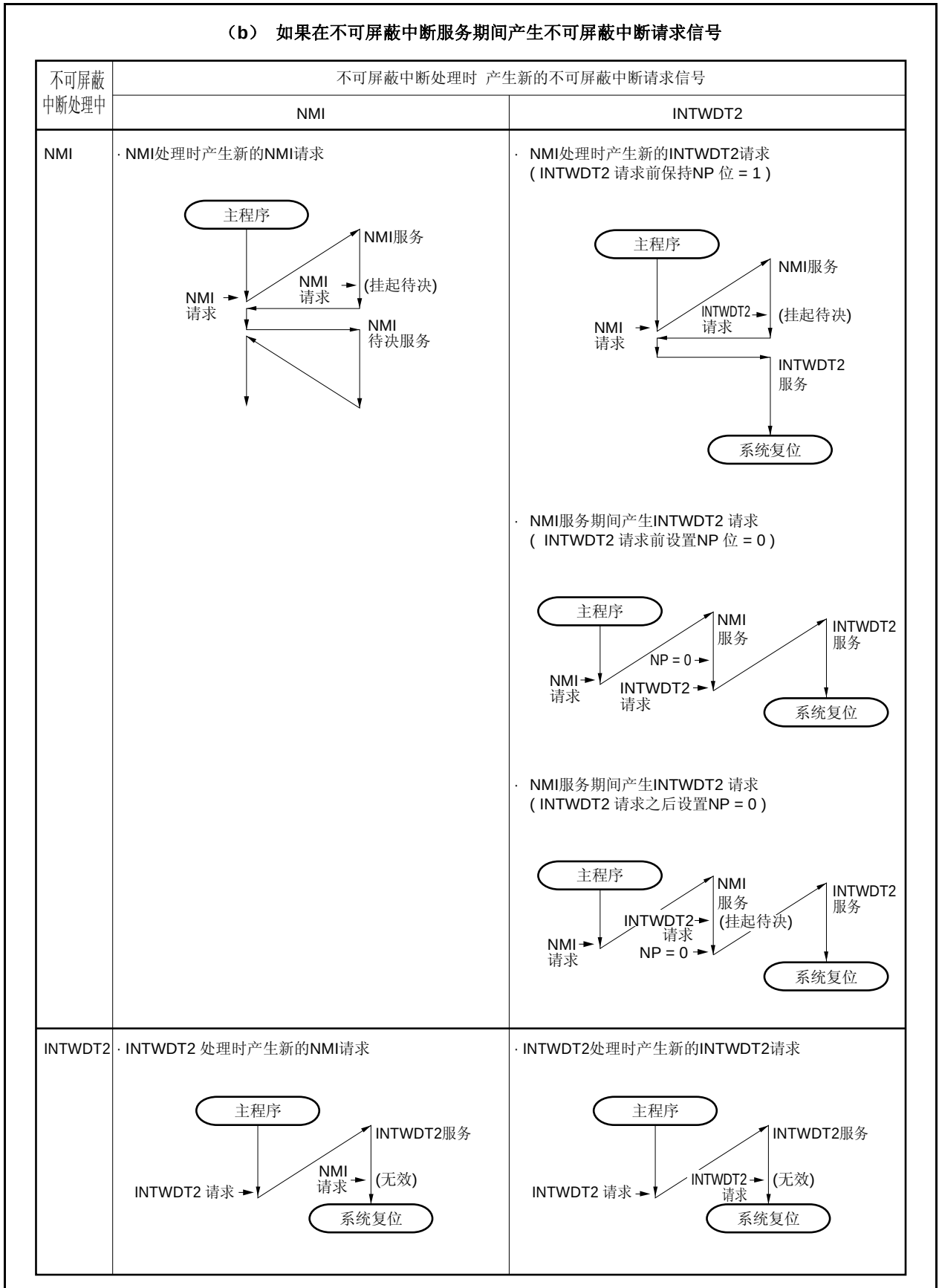


图 19-1. 不可屏蔽中断请求信号的应答操作 (2/2)



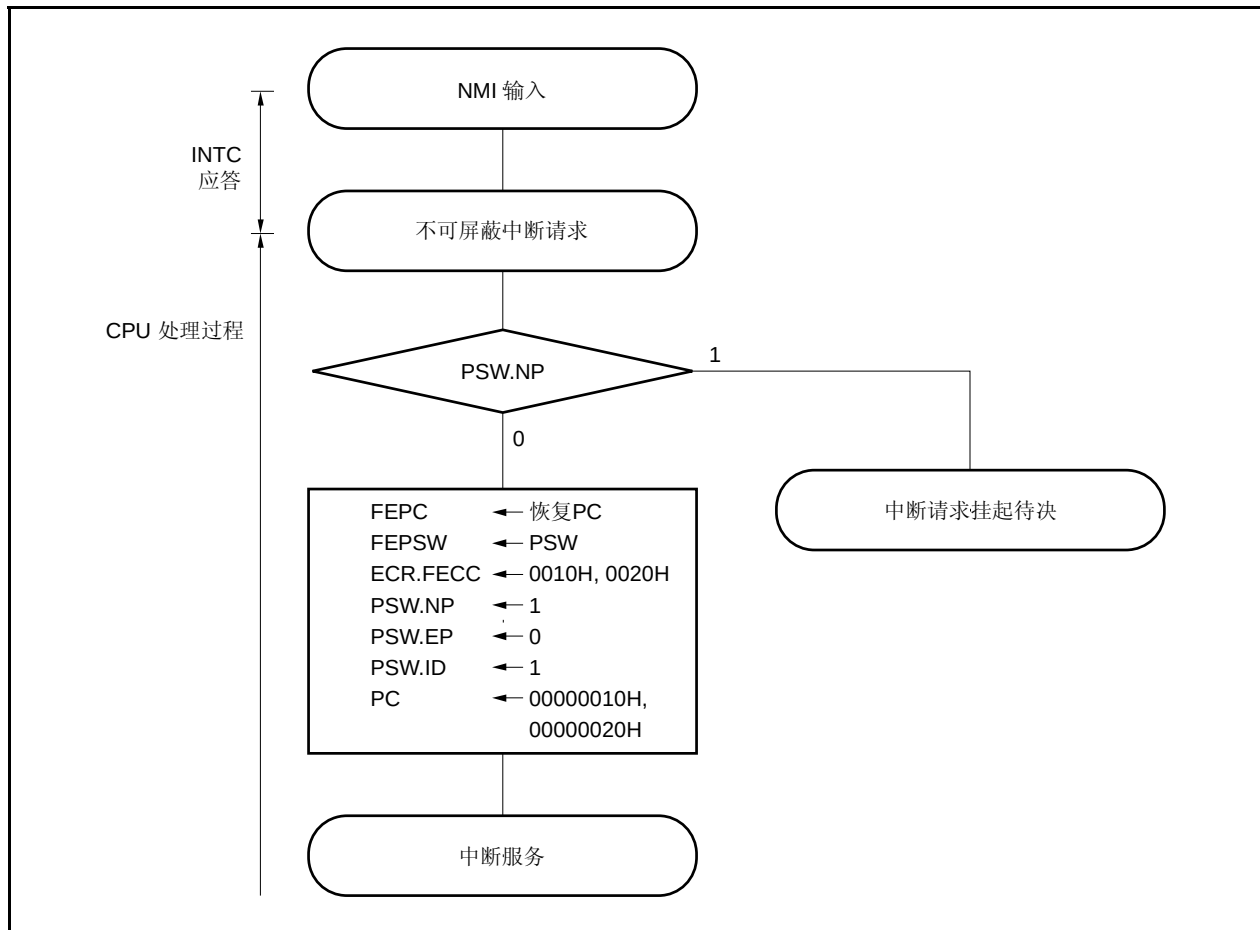
19.2.1 操作

如果产生一个不可屏蔽中断请求信号，则 CPU 执行以下处理过程，并将控制权转移至一个句柄程序。

- <1> 将待恢复的 PC 保存至 FEPC。
- <2> 将当前 PSW 保存至 FEPSW。
- <3> 将异常码 (0010H, 0020H) 写入 ECR 的高半字 (FECC)。
- <4> 将 PSW.NP 和 PSW.ID 位置为 1，并将 PSW.EP 位清为 0。
- <5> 将不可屏蔽中断相应的句柄地址 (00000010H, 00000020H) 载入 PC，并转移控制。

不可屏蔽中断的服务流程如下所示：

图 19-2. 不可屏蔽中断服务流程



19.2.2 恢复

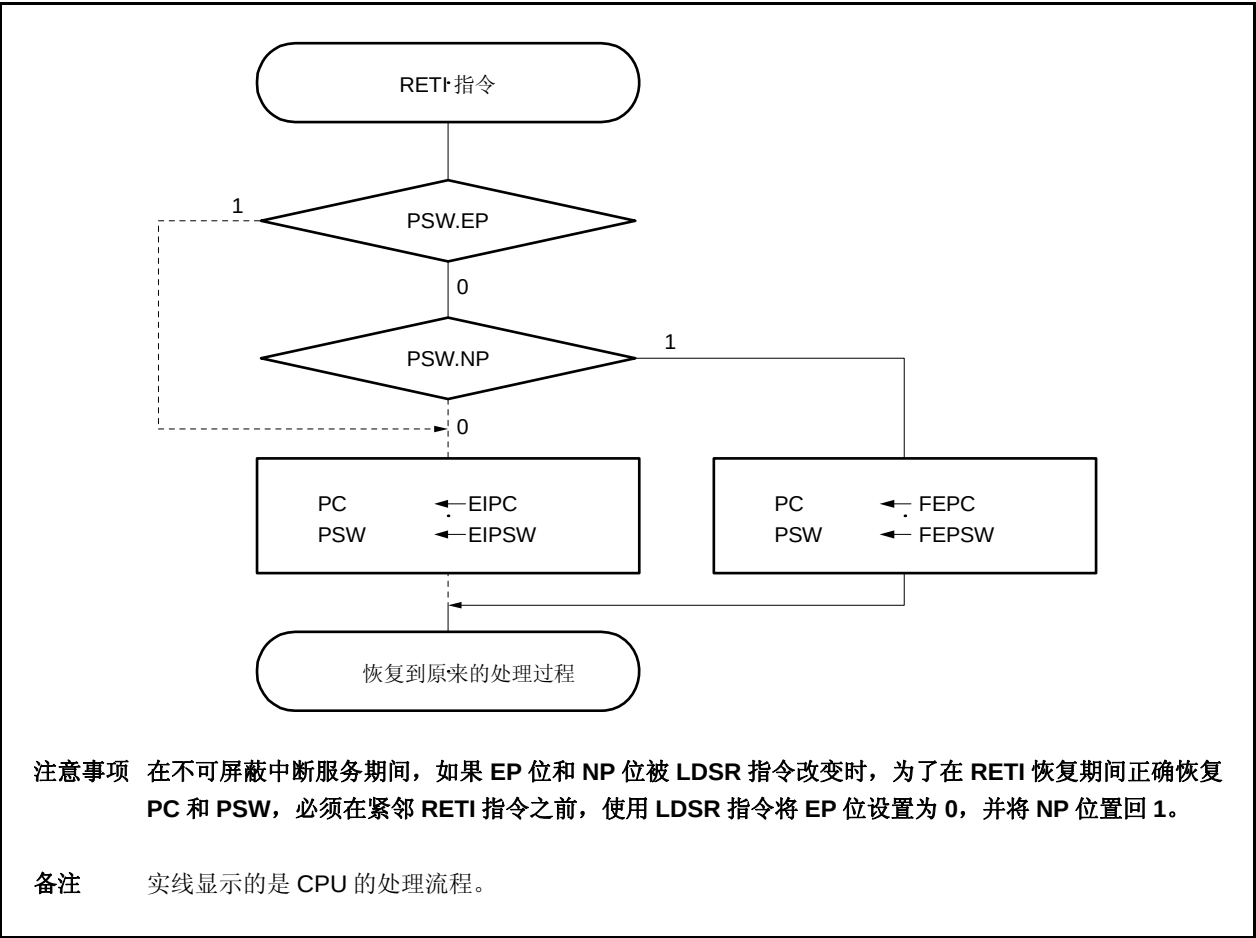
(1) 来自 NMI 引脚的输入

通过 RETI 指令，从 NMI 服务的恢复处理。
当执行 RETI 指令时，CPU 执行以下处理过程，并且控制转移到被恢复的 PC 地址。

- <1> 因为 PSW.EP 位是 0 且 PSW.NP 位是 1，所以，由 FEPC 和 FEPSW 分别载入相应的 PC 和 PSW 的恢复值。
- <2> 控制转移返回到被恢复的 PC 和 PSW 的地址。

RETI 指令的处理过程如下所示：

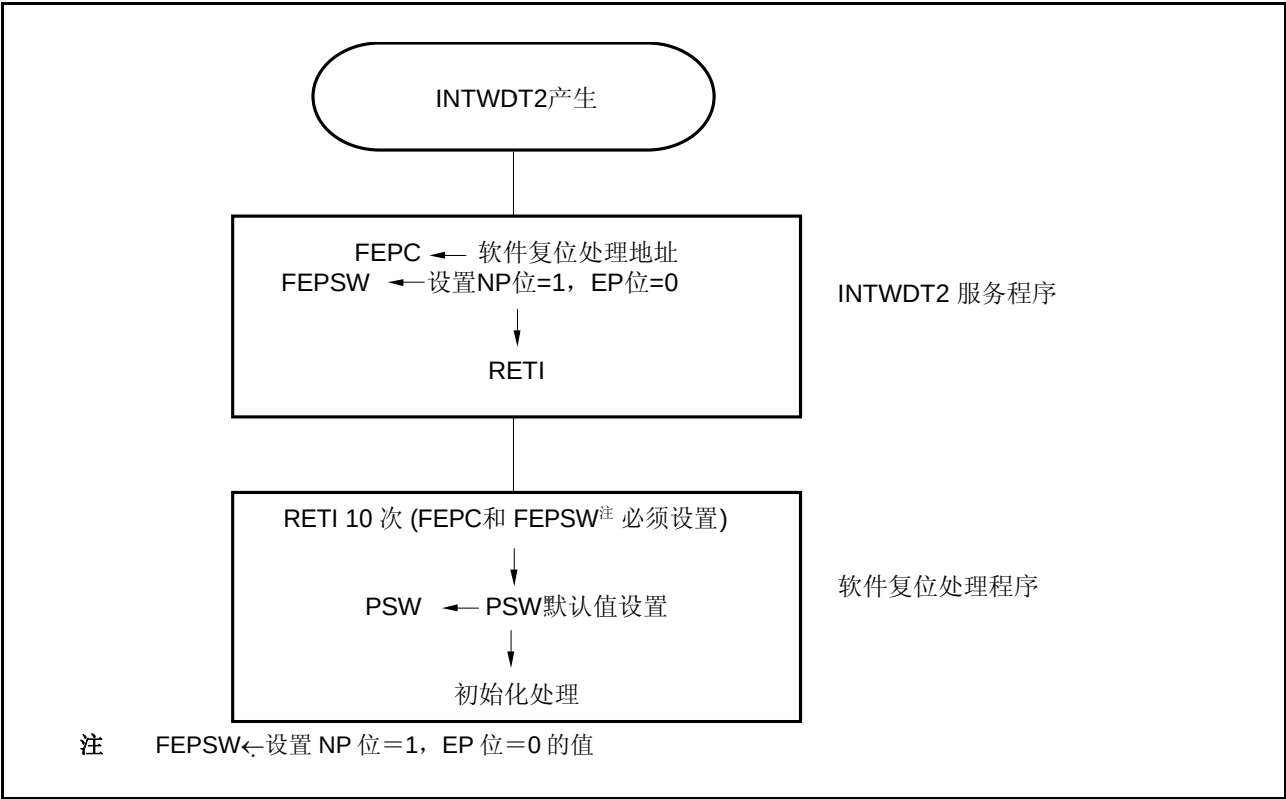
图 19-3. RETI 指令处理过程



(2) 来自 INTWDT2 的信号

对于不可屏蔽中断请求（INTWDT2）执行的不可屏蔽中断服务，其恢复时禁止使用 RETI 指令。执行下面的软件复位处理过程。

图 19-4. 软件复位处理过程



19.2.3 NP 标志

NP 标志是一个状态标志，用于指示不可屏蔽中断服务正在进行。
当不可屏蔽中断请求信号被响应之后，该标志置位，并且，屏蔽不可屏蔽中断请求，禁止响应多重中断。

复位后: 00000020H

	31																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
--	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

19.3 可屏蔽中断

可屏蔽中断请求信号可以由中断控制寄存器进行屏蔽。V850ES/JJ3 单片机有 69 个可屏蔽中断源。

如果同时发生两个或更多的可屏蔽中断请求信号，根据默认的优先级进行响应。除了默认优先级之外，使用中断控制寄存器（可编程优先级控制）可以指定 8 个优先等级。

当一个中断请求信号已经得到响应，则禁止响应其他可屏蔽中断请求信号，并且中断禁止（DI）状态置位。

当在中断服务程序中执行 EI 指令，中断使能（EI）状态置位，可以使能那些比正在处理的中断请求信号（由中断控制寄存器指定）具有更高的优先级的中断服务。注意，只有具有较高优先级的中断才有这种能力；具有相同优先等级的中断不能于嵌套。

但是，为了允许多重中断（中断嵌套），在执行 EI 指令之前有必要将 EIPC 和 EIPSW 保存到存储器或通用寄存器中，并在使用 RETI 指令之前执行 DI 指令，以恢复 EIPC 和 EIPSW 原始值。

19.3.1 操作

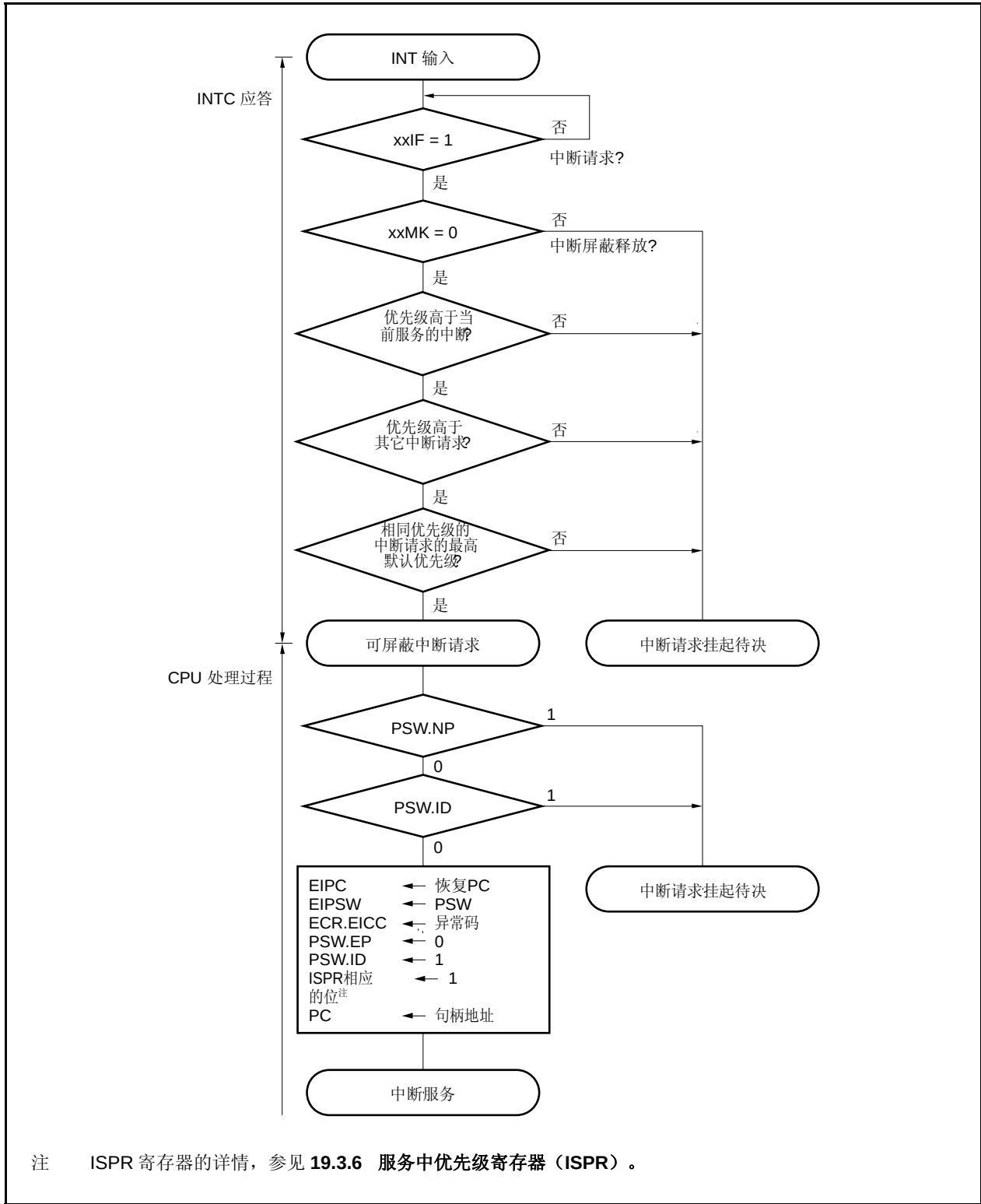
如果发生可屏蔽中断，CPU 执行以下处理过程，并控制转移至句柄程序。

- <1> 将 PC 保存至 EIPC。
- <2> 将 PSW 保存至 EIPSW。
- <3> 将异常码写入 ECR（EICC）的低半字。
- <4> 对 PSW.ID 位置 1，并对 PSW.EP 位清 0。
- <5> 将各个中断对应的处理程序地址设置到 PC 指针，并进行控制转移。

由中断控制寄存器（INTC）屏蔽的可屏蔽中断请求信号，和在另外的中断服务处理时（当 PSW.NP 位=1 或 ID 位=1）产生的可屏蔽中断请求信号，都被保持挂起状态存放在 INTC 内。这种情况下，如果取消对可屏蔽中断的屏蔽或使用 RETI 或 LDSR 指令将 NP 或 ID 清除为 0，则根据被挂起的可屏蔽中断请求的优先级判断，来启动一个新的可屏蔽中断服务。

下图显示了可屏蔽中断的服务过程：

图 19-5. 可屏蔽中断服务过程



注 ISPR 寄存器的详情，参见 **19.3.6 服务中优先级寄存器 (ISPR)**。

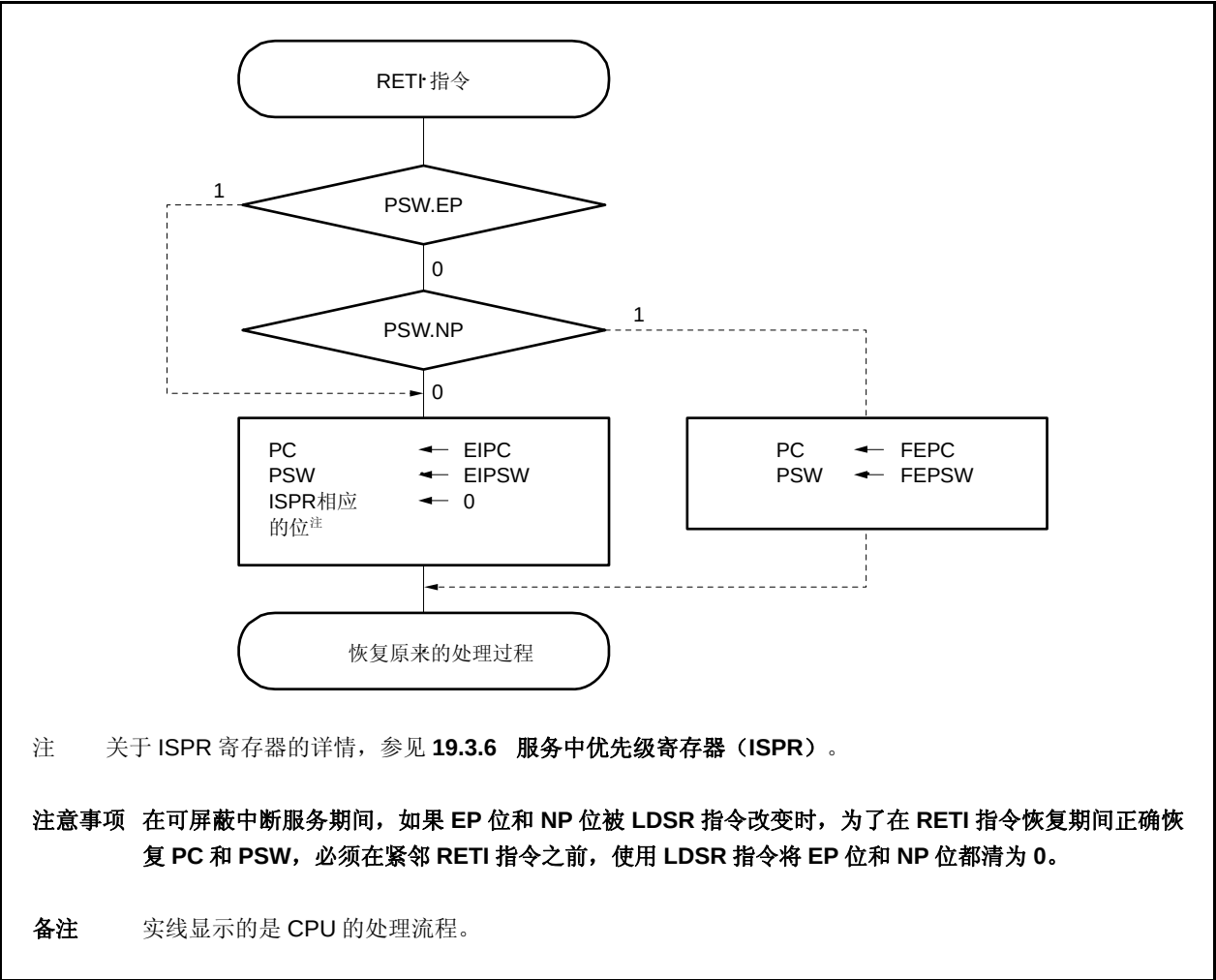
19.3.2 恢复

执行 RETI 指令，可以从可屏蔽中断服务中恢复。
当执行 RETI 指令时，CPU 执行以下处理过程，并且控制转移到被恢复的 PC 地址。

- <1> 因为 PSW.EP 位为 0 且 PSW.NP 位为 0，所以由 EIPC 和 EIPSW 分别载入 PC 和 PSW 的恢复值。
- <2> 控制转移返回到被恢复的 PC 和 PSW 的地址。

RETI 指令的处理过程如下所示：

图 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，也就是在一个中断正在服务处理时响应另一个中断。多重中断可以通过优先等级进行控制。

有两种类型的优先级控制方式：基于默认优先级的控制，以及基于中断控制寄存器（xxICn）的中断优先级指定位（xxPRn）确定的可编程优先级控制。当两个或两个以上中断同时发生，且 xxPRn 位指定的优先级都相同时，按照每个中断请求类型预先设定（默认优先级）的优先级次序进行处理。更多信息参见 **表 19-1 中断源列表**。通过设置优先级标志，可编程优先级控制将中断请求信号划分为 8 个等级。

注意，当中断请求信号被响应时，PSW.ID 标志自动被置 1。所以，当使用多重中断时，应该预先将 ID 标志清为 0（比如，在中断服务程序中插入一条 EI 指令）来设置多重中断允许模式。

备注 **xx:** 每一个外设单元的标志名称（参见 **表 19-2 中断控制寄存器（xxICn）**）。

n: 外设单元编号（参见**表 19-2 中断控制寄存器（xxICn）**）。

图 19-7. 当一个中断服务时发生另一个中断请求的处理示例 (1/2)

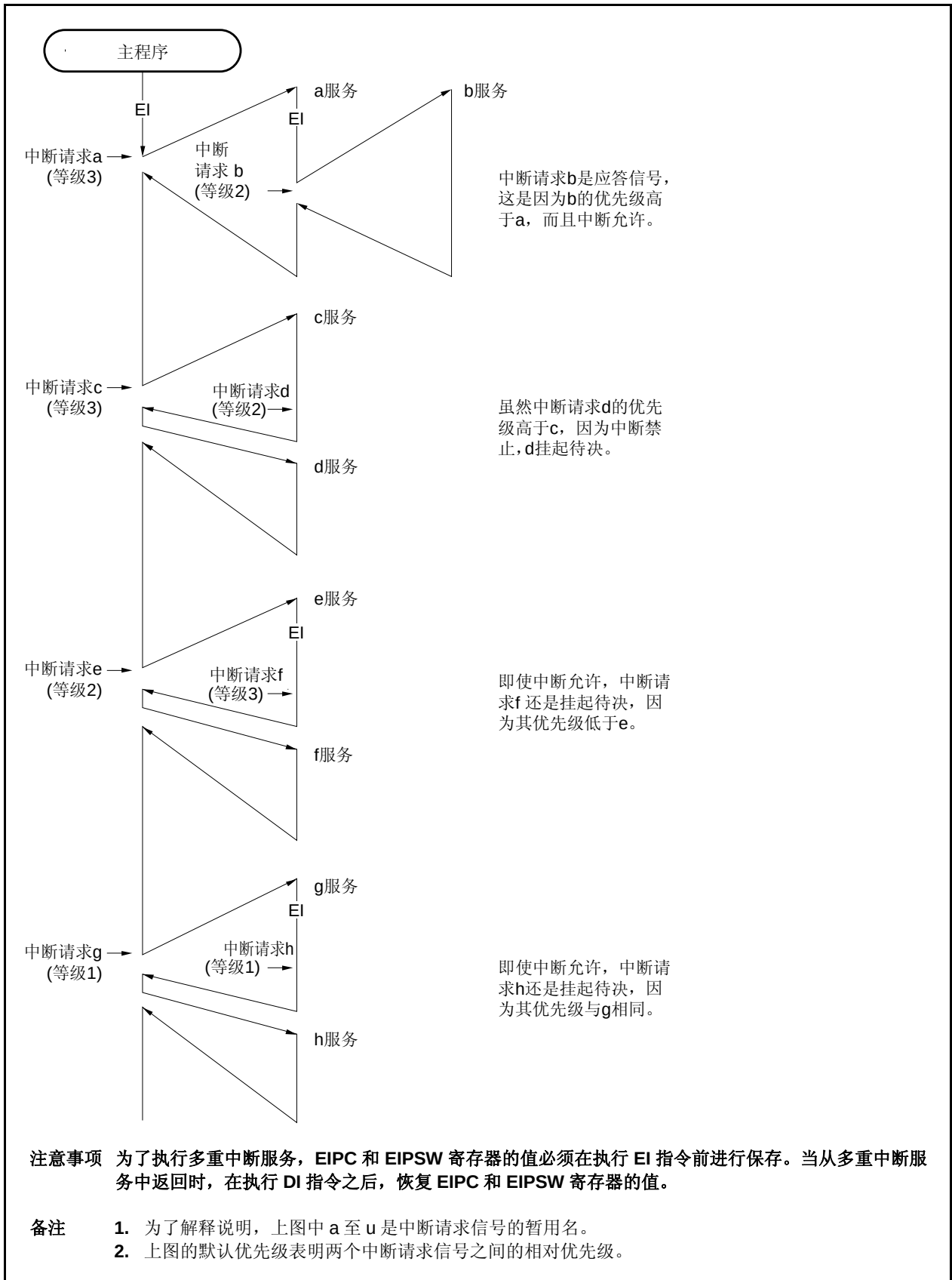


图 19-7. 一个中断服务当中而另一个中断请求发生的处理举例 (2/2)

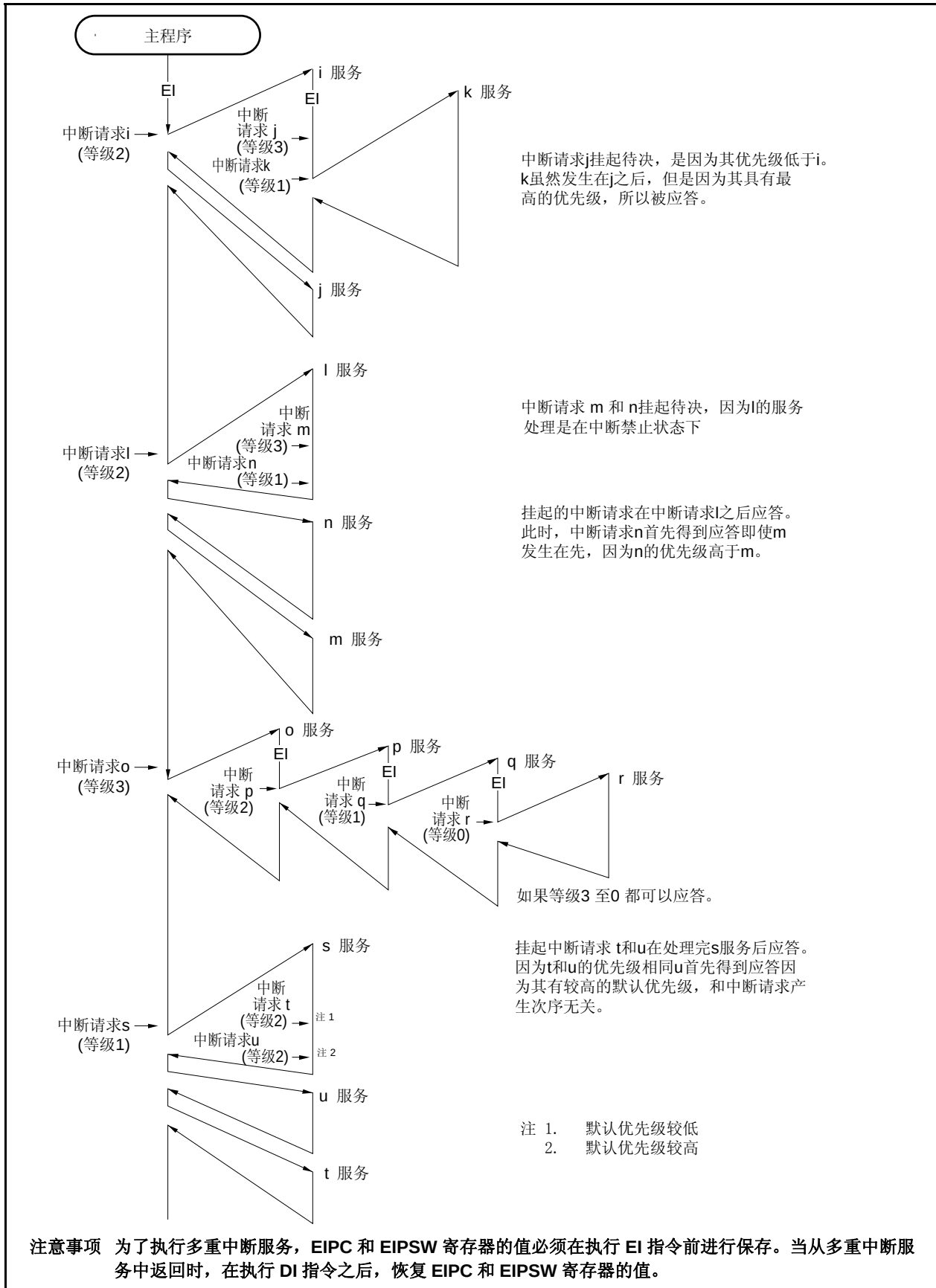
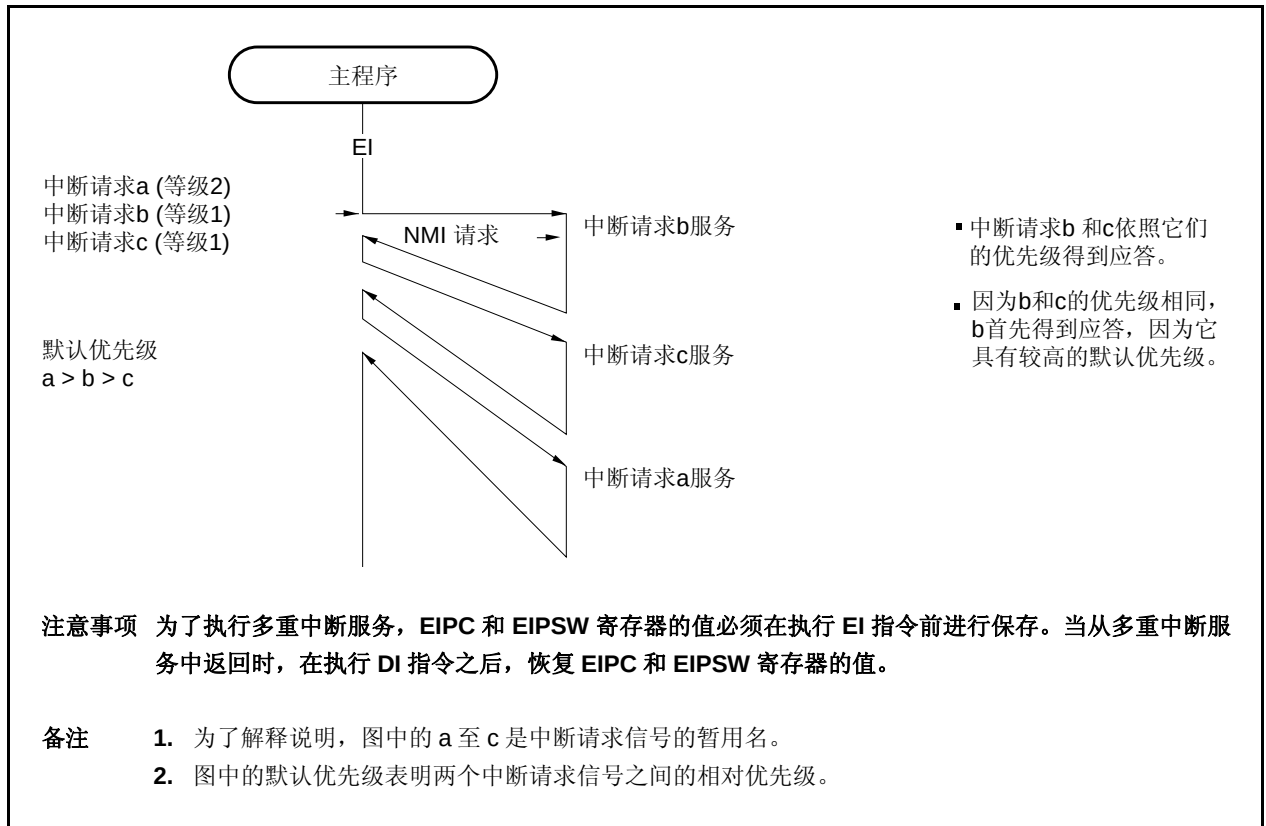


图 19-8. 同时发生的中断请求信号服务举例



19.3.4 中断控制寄存器（xxICn）

每一个可屏蔽中断请求信号（可屏蔽中断）都被指定一个 xxICn 寄存器，用来设置每一个可屏蔽中断请求的控制条件。

该寄存器可以按字节或按位进行读写。

系统复位后，寄存器被置为 47H。

注意事项 读取 xxICn.xxIFn 位时，禁止中断（DI）或屏蔽中断。如果在中断允许（EI）时或在中断未屏蔽的情况下读取 xxIFn 位，当响应一个中断与读取该位的时序发生冲突时，可能读不出正确值。

复位后: 47H R/W 地址: FFFFFFF112H 至 FFFFFFF184H

	<7>	<6>	5	4	3	2	1	0
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	中断请求标志 ^注
0	不产生中断请求
1	产生中断请求

xxMKn	中断屏蔽标志
0	中断服务允许
1	中断服务禁止 (挂起待决)

xxPRn2	xxPRn1	xxPRn0	中断优先级指定位
0	0	0	指定等级0 (最高)
0	0	1	指定等级1
0	1	0	指定等级2
0	1	1	指定等级3
1	0	0	指定等级4
1	0	1	指定等级5
1	1	0	指定等级6
1	1	1	指定等级7 (最低)

注 如果一个中断请求信号得到响应，xxIFn 标志由硬件自动复位。

备注 xx: 每一个外设单元的标志名称（参见 表 19-2 中断控制寄存器（xxICn））。
 n : 外设单元编号（参见表 19-2 中断控制寄存器（xxICn））。

中断控制寄存器的地址及位如下所示：

表 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	WTIIC	WTIIF	WTIMK	0	0	0	WTIPR2	WTIPR1	WTIPR0
FFFFF17CH	WTIC	WTIF	WTMK	0	0	0	WTPR2	WTPR1	WTPR0
FFFFF18EH	PIC8	PIF8	PMK8	0	0	0	PPR82	PPR81	PPR80
FFFFF190H	TP6OVIC	TP6OVIF	TP6OVMK	0	0	0	TP6OVPR2	TP6OVPR1	TP6OVPR0
FFFFF192H	TP6CCIC0	TP6CCIF0	TP6CCMK0	0	0	0	TP6CCPR02	TP6CCPR01	TP6CCPR00
FFFFF194H	TP6CCIC1	TP6CCIF1	TP6CCMK1	0	0	0	TP6CCPR12	TP6CCPR11	TP6CCPR10
FFFFF196H	TP7OVIC	TP7OVIF	TP7OVMK	0	0	0	TP7OVPR2	TP7OVPR1	TP7OVPR0
FFFFF198H	TP7CCIC0	TP7CCIF0	TP7CCMK0	0	0	0	TP7CCPR02	TP7CCPR01	TP7CCPR00
FFFFF19AH	TP7CCIC1	TP7CCIF1	TP7CCMK1	0	0	0	TP7CCPR12	TP7CCPR11	TP7CCPR10
FFFFF19CH	TP8OVIC	TP8OVIF	TP8OVMK	0	0	0	TP8OVPR2	TP8OVPR1	TP8OVPR0
FFFFF19EH	TP8CCIC0	TP8CCIF0	TP8CCMK0	0	0	0	TP8CCPR02	TP8CCPR01	TP8CCPR00
FFFFF1A0H	TP8CCIC1	TP8CCIF1	TP8CCMK1	0	0	0	TP8CCPR12	TP8CCPR11	TP8CCPR10
FFFFF1A2H	CB5RIC	CB5RIF	CB5RMK	0	0	0	CB5RPR2	CB5RPR1	CB5RPR0
FFFFF1A4H	CB5TIC	CB5TIF	CB5TMK	0	0	0	CB5TPR2	CB5TPR1	CB5TPR0
FFFFF1A6H	UA3RIC	UA3RIF	UA3RMK	0	0	0	UA3RPR2	UA3RPR1	UA3RPR0
FFFFF1A8H	UA3TIC	UA3TIF	UA3TMK	0	0	0	UA3TPR2	UA3TPR1	UA3TPR0

19.3.5 中断屏蔽寄存器 0 至 4 (IMR0 至 IMR4)

IMR0 至 IMR4 寄存器用于设置可屏蔽中断的中断屏蔽状态。IMR0 至 IMR4 寄存器的 xxMKn 位等同于 xxICn.xxMKn 位。

IMRm 寄存器可以按 16 位宽度进行读取或写入 (m=0 至 4)。

如果 IMRm 寄存器的高 8 位用作 IMRmH 寄存器, 低 8 位用作 IMRmL 寄存器, 则这些寄存器可以按字节或按位进行读写 (m=0 至 4)。

系统复位后, 这些寄存器被设置为 FFFFH。

注意事项 设备文件将 xxICn.xxMKn 位定义为保留字。如果使用 xxMKn 的名称进行位操作, xxICn 寄存器的内容被改写, 而不是 IMRm 寄存器被改写 (结果, IMRm 寄存器的内容也被改写)。

复位后: FFFFH		R/W	地址:		IMR4 FFFFF108H, IMR4L FFFFF108H, IMR4H FFFFF109H					
			15	14	13	12	11	10	9	8
IMR4 (IMR4H ^註)			1	1	1	UA3TMK	UA3RMK	CB5TMK	CB5RMK	TP8CCMK1
			7	6	5	4	3	2	1	0
IMR4L			TP8CCMK0	TP8OVMK	TP7CCMK1	TP7CCMK0	TP7OVMK	TP6CCMK1	TP6CCMK0	TP6OVMK
复位后: FFFFH		R/W	地址:		IMR3 FFFFF106H, IMR3L FFFFF106H, IMR3H FFFFF107H					
			15	14	13	12	11	10	9	8
IMR3 (IMR3H ^註)			PMK8	1	1	1	1	1	1	1
			7	6	5	4	3	2	1	0
IMR3L			1	WTMK	WTIMK	KRMK	DMAMK3	DMAMK2	DMAMK1	DMAMK0
复位后: FFFFH		R/W	地址:		IMR2 FFFFF104H, IMR2L FFFFF104H, IMR2H FFFFF105H					
			15	14	13	12	11	10	9	8
IMR2 (IMR2H ^註)			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: FFFFH		R/W	地址:		IMR1 FFFFF102H, IMR1L FFFFF102H, IMR1H FFFFF103H					
			15	14	13	12	11	10	9	8
IMR1 (IMR1H ^註)			TP5CCMK1	TP5CCMK0	TP5OVMK	TP4CCMK1	TP4CCMK0	TP4OVMK	TP3CCMK1	TP3CCMK0
			7	6	5	4	3	2	1	0
IMR1L			TP3OVMK	TP2CCMK1	TP2CCMK0	TP2OVMK	TP1CCMK1	TP1CCMK0	TP1OVMK	TP0CCMK1
复位后: FFFFH		R/W	地址:		IMR0 FFFFF100H, IMR0L FFFFF100H, IMR0H FFFFF101H					
			15	14	13	12	11	10	9	8
IMR0 (IMR0H ^註)			TP0CCMK0	TP0OVMK	TQ0CCMK3	TQ0CCMK2	TQ0CCMK1	TQ0CCMK0	TQ0OVMK	PMK7
			7	6	5	4	3	2	1	0
IMR0L			PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK
			xxMKn	中断屏蔽标志的设置						
			0	中断服务使能						
			1	中断服务禁止						

注 若要按字节或按位读取 IMR0 至 IMR4 寄存器的第 8 位至第 15 位，需指定这些位作为 IMR0H 至 IMR4H 寄存器的第 0 位至第 7 位。

注意事项 将 IMR4 寄存器的第 13 位至第 15 位和 IMR3 寄存器的第 7 位至第 14 位设置为 1。如果这些位的设置被改变，操作将无法保证。

备注 xx: 每一个外设单元的标志名称（参见 表 19-2 中断控制寄存器（xxICn））。
n : 外设单元编号（参见 表 19-2 中断控制寄存器（xxICn））。

19.3.6 正在服务中优先级寄存器 (ISPR)

ISPR 寄存器保存当前响应的可屏蔽中断的优先级。当响应一个中断请求信号时，该寄存器中与中断请求信号优先级对应的位被置为 1，并且在中断服务期间一直保持置位。

当执行 RETI 指令时，对应最高优先级中断请求信号的位被硬件自动清 0。但是，从不可屏蔽中断服务或异常处理中返回时，并不被复位为 0。

该寄存器为只读寄存器，按字节或按位读取。

系统复位后，该寄存器被设为 00H。

注意事项 中断允许 (EI) 状态下，如果在读取 ISPR 寄存器的同时响应一个中断，读取的 ISPR 寄存器值可能是响应中断后其对应位被置位后的值。为了在响应中断前准确读出 ISPR 寄存器的值，应在中断禁止 (DI) 的状态下读取。

复位后: 00H R 地址: FFFFF1FAH

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	当前正在应答中断的优先级
0	没有响应优先级为n的中断请求信号
1	正在响应优先级为n的中断请求信号

备注 n=0 至 7 (优先级等级)

19.3.7 ID 标志

该标志用于控制可屏蔽中断的操作状态，并且存储有关使能或禁止中断请求信号的控制信息。中断禁止标志（ID）位于 PSW 内。

系统复位后，该标志被置为 00000020H。

复位后: 00000020H

	31																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
--	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

19.3.8 看门狗定时器模式寄存器 2（WDTM2）

该寄存器可以按字节或按位进行读写（详情参见 第十一章 看门狗定时器 2 功能）。

系统复位后，该寄存器被设置为 67H。

复位后: 67H R/W 地址: FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	0	0	0	0	0

WDM21	WDM20	看门狗定时器操作模式选择
0	0	停止操作
0	1	不可屏蔽中断请求模式
1	×	复位模式 (初始值)

19.4 软件异常

当 CPU 执行 TRAP 指令时，产生软件异常，并且总是能够得到响应。

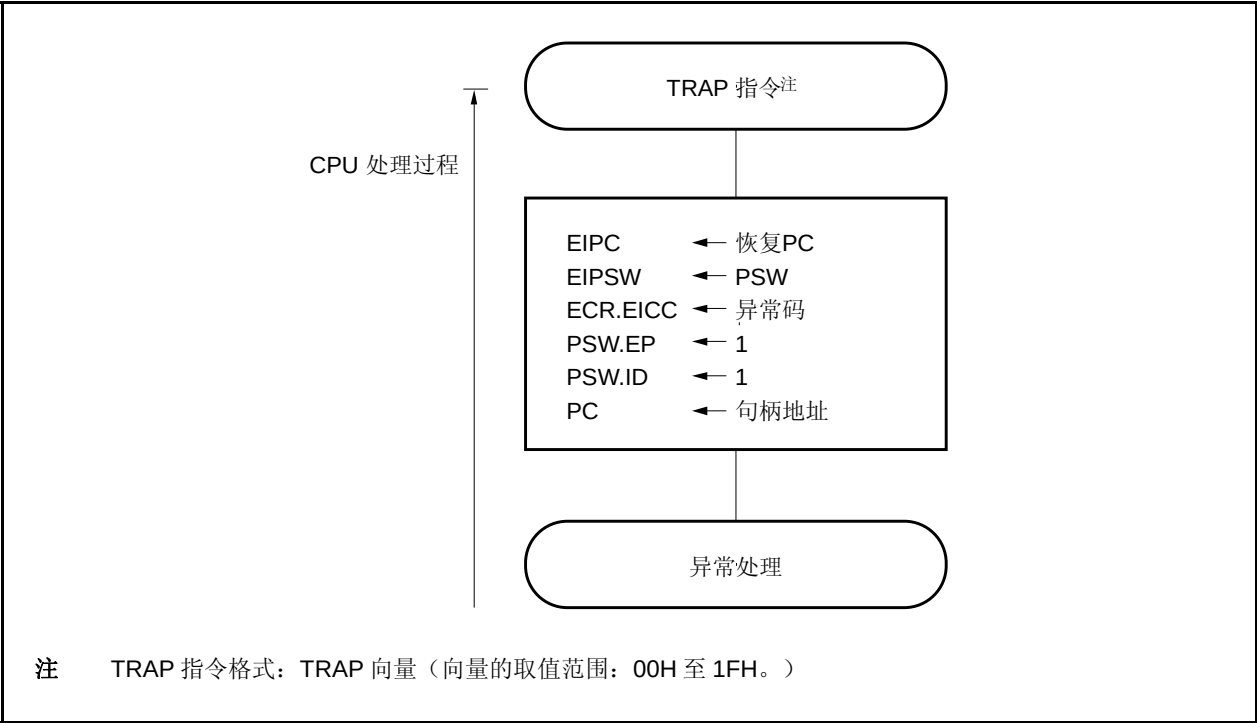
19.4.1 操作

如果发生软件异常，CPU 执行以下处理，并控制转移到句柄程序。

- <1> 将 PC 保存至 EIPC。
- <2> 将 PSW 保存至 EIPSW。
- <3> 写入异常码到 ECR（中断源）的低 16 位（EICC）。
- <4> 将 PSW.EP 和 PSW.ID 位置为 1。
- <5> 将软件异常对应的句柄地址（00000040H 或 00000050H）设置到 PC 指针，并进行控制转移。

软件异常的处理过程如下所示：

图 19-9. 软件异常的处理过程



句柄地址由 TRAP 指令的操作数（向量）决定。如果向量为 00H 至 0FH，句柄地址是 00000040H，而如果向量为 10H 至 1FH，句柄地址是 00000050H。

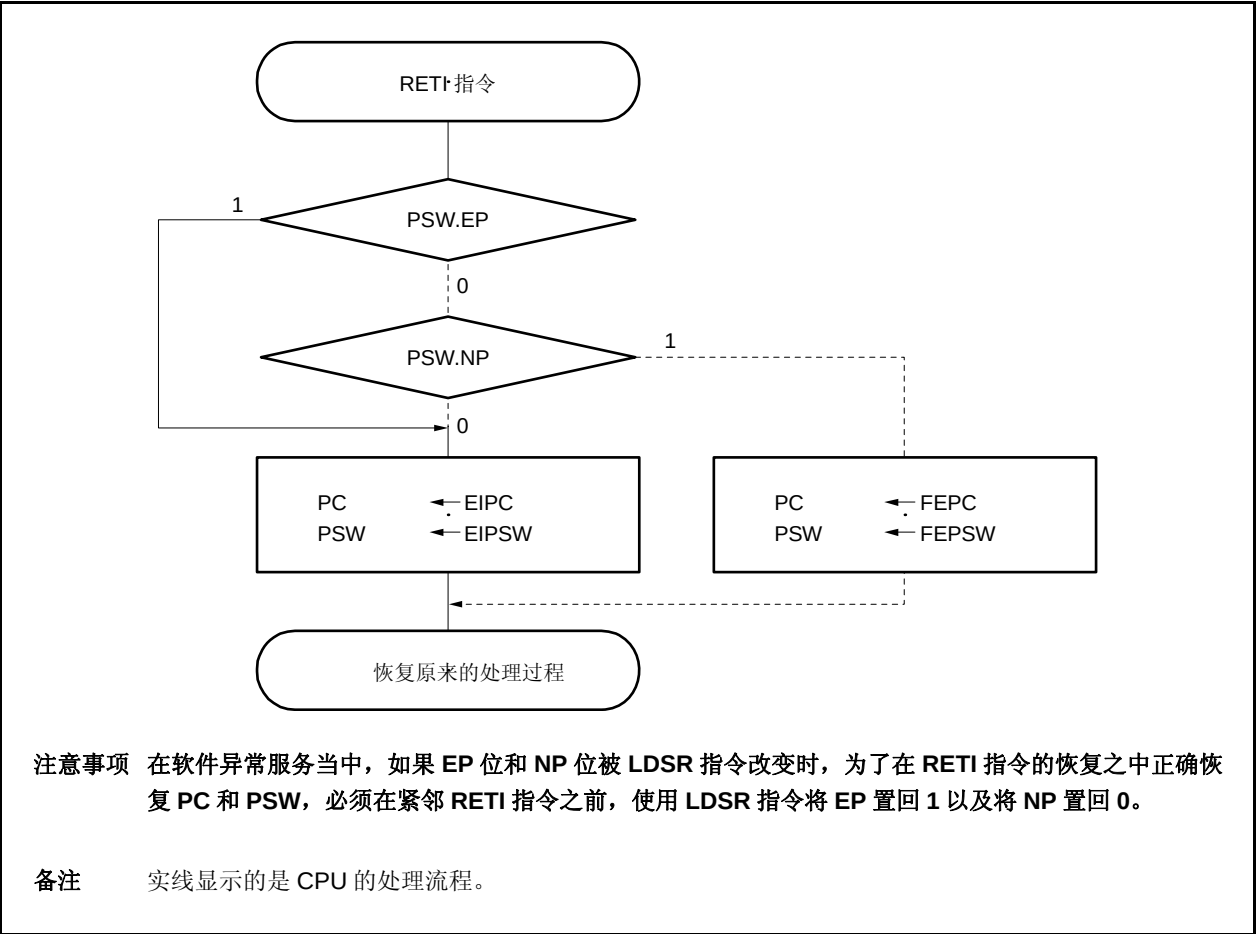
19.4.2 恢复

执行 RETI 指令，可以从软件异常处理中恢复。
通过执行 RETI 指令，CPU 执行以下处理并将控制转移至被恢复的 PC 地址。

- <1> 由于 PSW.EP 位为 1，由 EIPC 和 EIPSW 载入被恢复的 PC 和 PSW。
- <2> 控制转移至被恢复的 PC 和 PSW 的地址。

RETI 指令的处理过程如下所示：

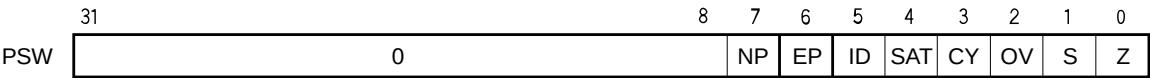
图 19-10. RETI 指令处理过程



19.4.3 EP 标志

EP 是一个状态标志，用于指示异常处理正在处理中。当异常发生时该位被置位。

复位后: 00000020H



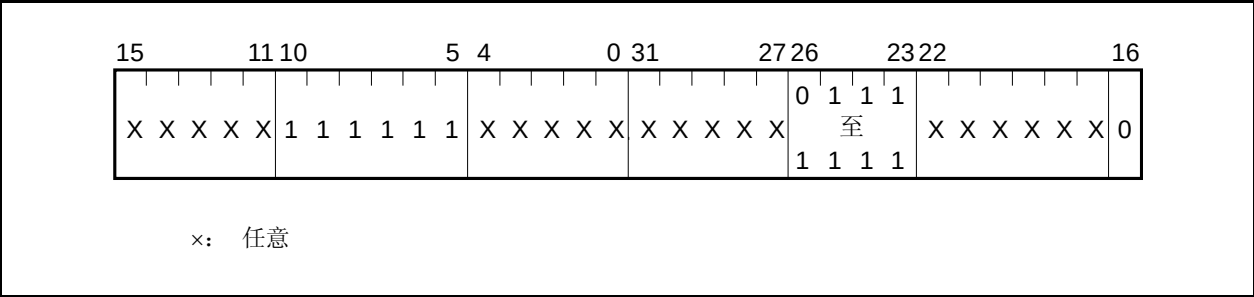
EP	异常处理状态
0	没有进行异常处理。
1	正在进行异常处理。

19.5 异常陷阱

异常陷阱是一个中断，当非法执行指令时产生异常陷阱中断请求。在 V850ES/JJ3 单片机中，非法操作码陷阱（ILGOP：非法操作码陷阱）被认为是一个异常陷阱。

19.5.1 非法操作码定义

非法指令有 111111B 作为操作码（第 10 位至第 5 位），0111B 至 1111B 作为次操作码（第 26 位至第 23 位），0B 作为另一个次操作码（第 16 位）。当执行的指令适用于这样的非法指令时，就会产生一个异常陷阱。



注意事项 建议不要使用非法操作码，因为非法操作码将来可能被指定为指令。

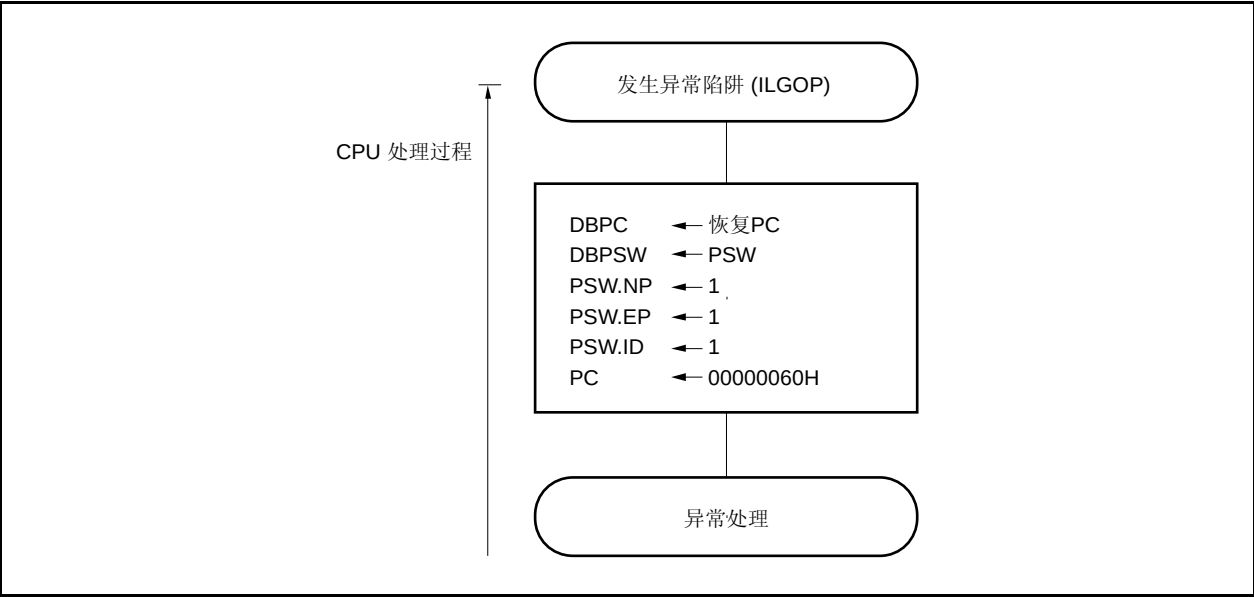
(1) 操作

当发生异常陷阱时，CPU 执行以下处理，并将控制转移到句柄程序：

- <1> 将待恢复的 PC 保存至 DBPC。
- <2> 将当前 PSW 保存至 DBPSW。
- <3> 将 PSW.NP，PSW.EP 和 PSW.ID 位置为 1。
- <4> 将异常陷阱对应的处理程序地址（00000060H）设置到 PC 指针，并进行控制转移。

异常陷阱的处理过程如下所示：

图 19-11. 异常陷阱处理过程



(2) 恢复

执行 DBRET 指令，可以从异常陷阱中恢复。

通过执行 DBRET 指令，CPU 进行以下处理并将控制转移至被恢复的 PC 地址。

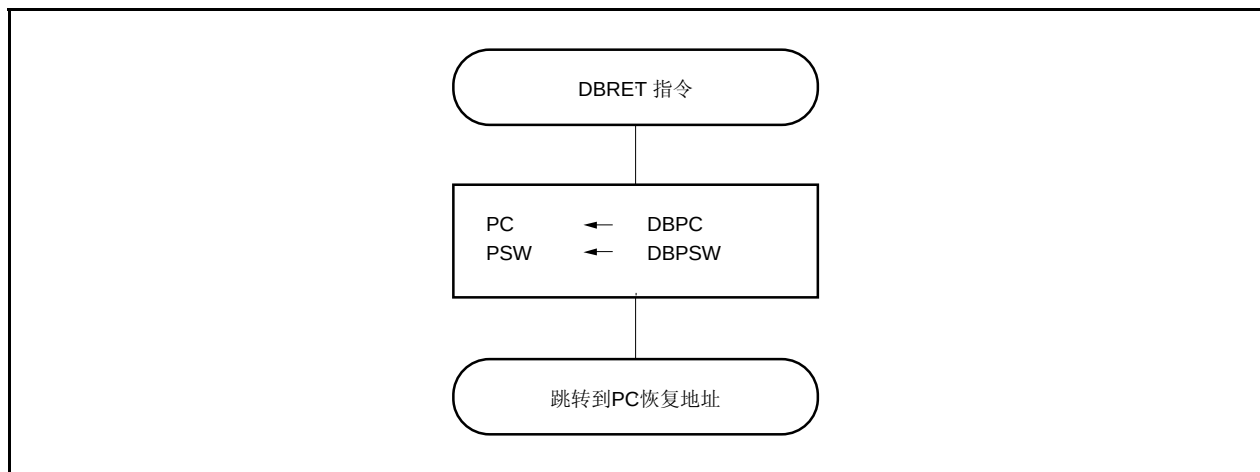
<1> 从 DBPC 和 DBPSW 中载入被恢复的 PC 和 PSW。

<2> 控制转移至被恢复的 PC 和 PSW 所指定的地址。

注意事项 只有在执行非法操作码和 DBRET 指令之间的间隔内，才可以访问 DBPC 和 DBPSW。

从异常陷阱中恢复的处理过程如下所示：

图 19-12. 异常陷阱恢复处理过程



19.5.2 调试陷阱

调试陷阱是一个异常，在执行 DBTRAP 指令时产生，总是可以得到响应。

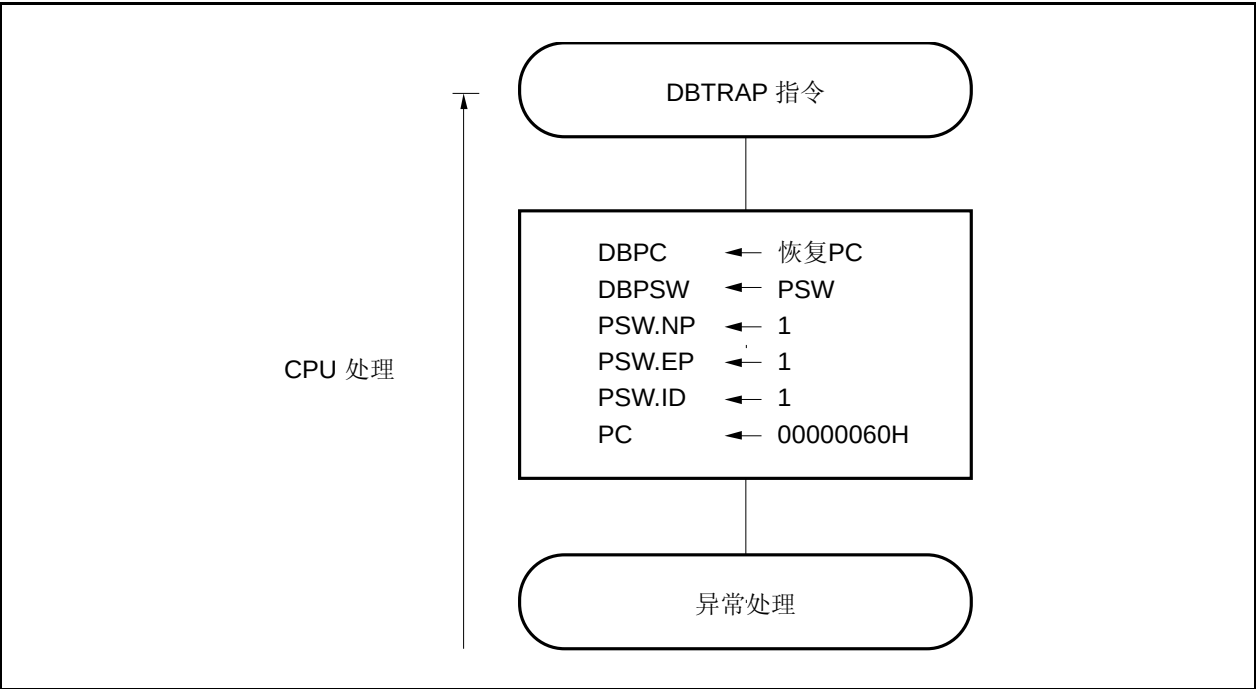
(1) 操作

当发生调试陷阱时，CPU 执行以下处理过程。

- <1> 将待恢复的 PC 保存至 DBPC。
- <2> 将当前 PSW 保存至 DBPSW。
- <3> 将 PSW.NP，PSW.EP 和 PSW.ID 位置为 1。
- <4> 将调试陷阱的句柄地址（00000060H）设置到 PC 指针，并进行控制转移。

调试陷阱的处理过程如下所示：

图 19-13. 调试陷阱处理过程



(2) 恢复

执行 DBRET 指令，可以从异常陷阱中进行恢复。

当执行 DBRET 指令时，CPU 执行以下处理过程并将控制转移到恢复 PC 地址。

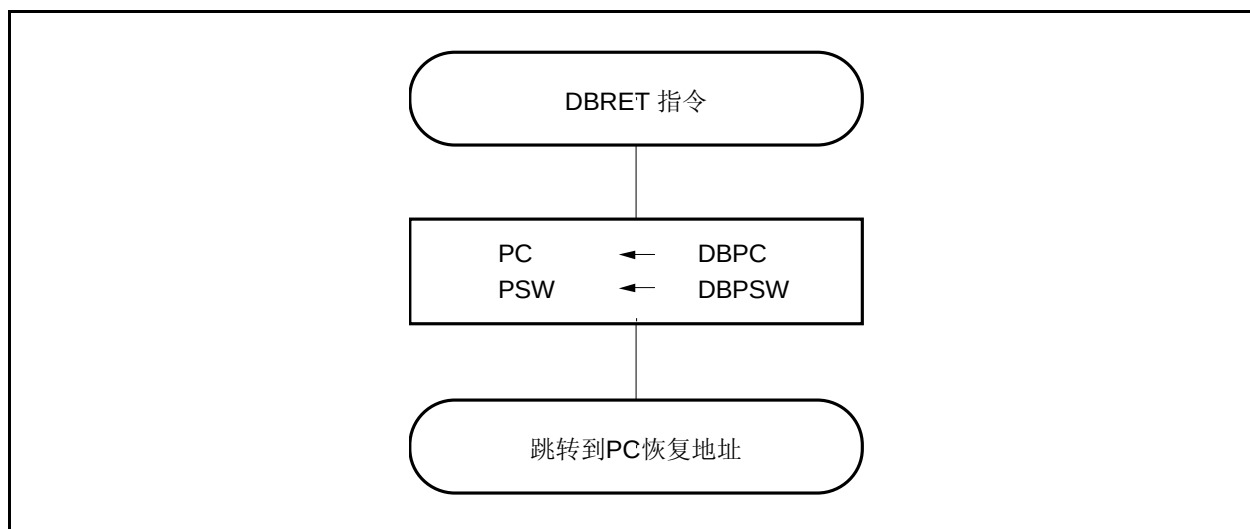
<1> 从 DBPC 和 DBPSW 中读取被恢复的 PC 和 PSW。

<2> 控制转移返回到被恢复的 PC 和 PSW 的地址。

注意事项 只有在执行 DBTRAP 和 DBRET 指令之间的间隔内，才可以访问 DBPC 和 DBPSW。

从调试陷阱中恢复的处理过程如下所示：

图 19-14. 从调试陷阱中恢复的处理流程



19.6 外部中断请求输入引脚（NMI 和 INTP0 至 INTP8）

19.6.1 噪声消除

（1）NMI 引脚的噪声消除

NMI 引脚具有模拟延迟的内部噪声消除电路。因此，输入 NMI 引脚的电平需要保持特定的或更长的时间，否则将不会被检测为边沿。因此，经过一定的时间之后，才能检测到边沿。

NMI 引脚能用来解除 STOP 模式。在 STOP 模式下，使用系统时钟，所以噪声无法消除，因为此时内部系统时钟是停止的。

（2）INTP0 至 INTP8 引脚的噪声消除

INTP0 至 INTP8 引脚具有模拟延迟的内部噪声消除电路。因此，输入 NMI 引脚的电平需要保持特定的或更长的时间，否则将不会被检测为边沿。因此，经过一定的时间之后，才能检测到边沿。

19.6.2 边沿检测

NMI 和 INTP0 至 INTP8 各个引脚输入信号的有效边沿可以从以下 4 种类型选择：

- 上升沿
- 下降沿
- 上升和下降双边沿
- 无边沿检测

在复位后，不检测 NMI 引脚的边沿。所以，除非使用 INTF0 和 INTR0 寄存器指定一种有效边沿，否则不响应中断请求信号（NMI 引脚作为正常端口引脚功能）。

(1) 外部中断下降沿和上升沿指定寄存器 0 (INTF0, INTR0)

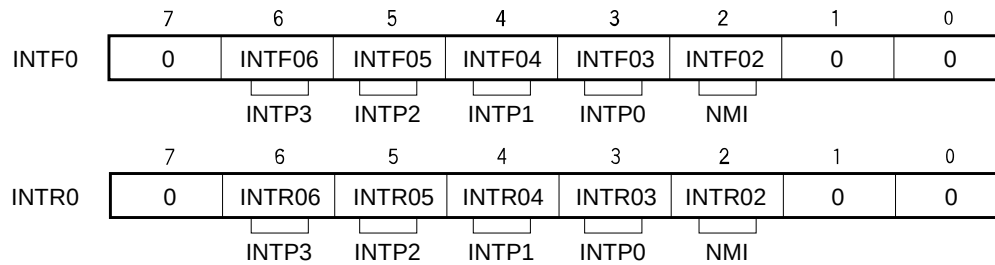
INTF0 和 INTR0 寄存器是 8 位寄存器，第 2 位用于指定 NMI 引脚的边沿检测，第 3 位至第 6 位用于指定外部中断引脚 (INTP0 至 INTP3) 的下降沿和上升沿检测。

这些寄存器可以按字节或按位进行读写。

系统复位后，这些寄存器被设置为 00H。

注意事项 当引脚功能由外部中断功能（复用功能）改变为端口功能时，可能会检测到一个边沿。所以，将 INTF0n 和 INTR0n 位清为 00，然后再设置端口模式。

复位后: 00H R/W 地址: INTF0 FFFFFFFC00H, INTR0 FFFFFFFC20H



备注 关于如何指定有效边沿，参见 表 19-3。

表 19-3. 有效边沿设定

INTF0n	INTR0n	有效边沿设定 (n=2 至 6)
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升和下降双边沿

注意事项 当这些寄存器不用于设置 NMI 或 INTP0 至 INTP3 引脚时，请确保将 INTF0n 和 INTR0n 位清为 00。

备注 n=2: NMI 引脚的控制
 n=3 至 6: INTP0 至 INTP3 引脚的控制

(2) 外部中断下降沿和上升沿指定寄存器 3 (INTF3, INTR3)

INTF3 和 INTR3 寄存器是 8 位寄存器，用于指定外部中断引脚 (INTP7) 的下降沿和上升沿检测。
这些寄存器可以按字节或按位进行读写。
系统复位后，这些寄存器被设置为 00H。

- 注意事项
- 1. 当引脚功能由外部中断功能（复用功能）改变为端口功能时，可能会检测到一个边沿。所以，将 INTF31 和 INTR31 位清为 00，然后再设置端口模式。
 - 2. INTP7 引脚和 RXDA0 引脚为复用功能引脚。当此引脚用作 RXDA0 时，禁止对复用功能引脚 INTP7 的边沿检测（将 INTF3.INTF31 和 INTR3.INTR31 位清为 0）。当此引脚用作 INTP7，停止 UARTA0 接收（将 UA0CTL0.UA0RXE 位清为 0）。

复位后: 00H R/W 地址: INTF3 FFFFC06H, INTR3 FFFFC26H

INTF3

7	6	5	4	3	2	1	0
0	0	0	0	0	0	INTF31	0

INTP7

INTR3

7	6	5	4	3	2	1	0
0	0	0	0	0	0	INTR31	0

INTP7

备注 关于如何指定有效边沿，参见 表 19-4。

表 19-4. 有效边沿设定

INTF31	INTR31	有效边沿指定
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升和下降双边沿

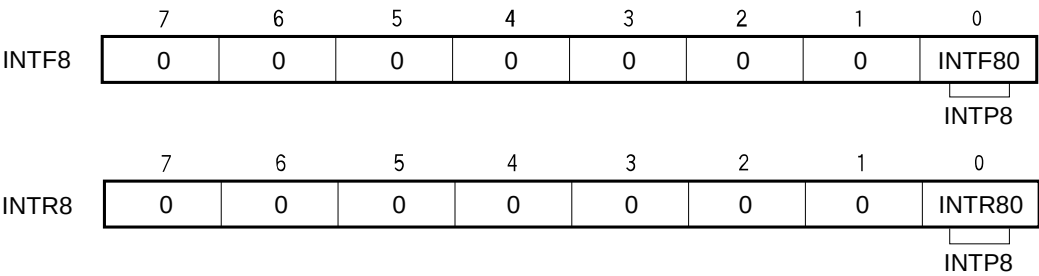
注意事项 当这些寄存器不用于设置 INTP7 引脚时，请确保将 INTF31 和 INTR31 位清为 00。

(3) 外部中断下降沿和上升沿指定寄存器 8 (INTF8, INTR8)

INTF8 和 INTR8 寄存器是 8 位寄存器，用于指定外部中断引脚 (INTP8) 的下降沿和上升沿检测。
这些寄存器可以按字节或按位进行读写。
系统复位后，这些寄存器被设置为 00H。

- 注意事项**
- 1. 当引脚功能由外部中断功能（复用功能）改变为端口功能时，可能会检测到一个边沿。所以，将 INTF80 和 INTR80 位清为 00，然后再设置端口模式。
 - 2. INTP8 引脚和 RXDA3 引脚为复用功能引脚。当此引脚用作 RXDA3 时，禁止对复用功能引脚 INTP8 的边沿检测（将 INTF8.INTF80 和 INTR8.INTR80 位清除为 0）。当此引脚用作 INTP8，停止 UARTA3 接收（将 UA3CTL0.UA3RXE 位清除为 0）。

复位后: 00H R/W 地址: INTF8 FFFFFC10H, INTR8 FFFFFC30H



备注 关于如何指定有效边沿，参见 表 19-5。

表 19-5. 有效边沿设定

INTF80	INTR80	有效边沿指定
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升和下降双边沿

注意事项 当这些寄存器不用于设置 INTP8 引脚时，请确保将 INTF80 和 INTR80 位清除为 00。

(4) 外部中断下降沿和上升沿指定寄存器 9H (INTF9H, INTR9H)

INTF9H 和 INTR9H 寄存器是 8 位寄存器，用于指定外部中断引脚 (INTP4 至 INTP6) 的下降沿和上升沿检测。这些寄存器可以按字节或按位进行读写。
系统复位后，这些寄存器被设置为 00H。

注意事项 当引脚功能由外部中断功能（复用功能）改变为端口功能，可能会检测到一个边沿。所以，将 INTF9n 和 INTR9n 位清为 00，然后再设置端口模式。

复位后: 00H R/W 地址: INTF9H FFFFC13H, INTR9H FFFFC33H

INTF9H

15	14	13	12	11	10	9	8
INTF915	INTF914	INTF913	0	0	0	0	0
<div>INTP6</div>	<div>INTP5</div>	<div>INTP4</div>					

INTR9H

15	14	13	12	11	10	9	8
INTR915	INTR914	INTR913	0	0	0	0	0
<div>INTP6</div>	<div>INTP5</div>	<div>INTP4</div>					

备注 关于如何指定有效边沿，参见 表 19-6。

表 19-6. 有效边沿设定

INTF9n	INTR9n	有效边沿设定 (n=13 至 15)
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升和下降双边沿

注意事项 当这些寄存器不用于设置 INTP4 至 INTP6 引脚时，请确保将 INTF9n 和 INTR9n 位清为 00。

备注 n=13 至 15: INTP4 至 INTP6 引脚的控制

(5) 噪声消除控制寄存器 (NFC)

INTP3 引脚具有一个数字噪声消除器。使用 NFC 寄存器来执行噪声消除的设置。

当选择数字噪声消除时，数字采样的采样时钟可从以下范围选择：f_{xx}/64，f_{xx}/128，f_{xx}/256，f_{xx}/512，f_{xx}/1,024 和 f_{XT}。执行 3 次采样。

当选择数字噪声消除时，如果指令采样的时钟在待机模式下被停止，则不能使用 INTP3 中断请求信号来释放待机模式。当使用 f_{XT} 作为采样时钟时，可以使用 INTP3 中断请求信号来释放副时钟操作模式活儿 IDLE1/IDLE2/STOP/sub-IDLE 模式。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被设为 00H。

注意事项 在采样时钟改变后，需要 3 个采样时钟来初始化数字噪声消除器。因此，如果在采样时钟改变后，有 INTP3 有效沿在这 3 个采样时钟内输入，可能会产生一个中断请求信号。因此，当使用中断和 DMA 功能时，注意以下几点。

- 当使用中断功能时，在 3 个采样时钟过后，对中断请求标志 (PIC3.PIF3 位) 清零后再使能中断。
- 当使用 DMA 功能 (由 INTP3 启动) 时，在 3 个采样时钟过后使能 DMA。

复位后: 00H R/W 地址: FFFFF318H

	7	6	5	4	3	2	1	0
NFC	NFEN	0	0	0	0	NFC2	NFC1	NFC0

NFEN	INTP3 引脚噪声消除设置
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			设置禁止

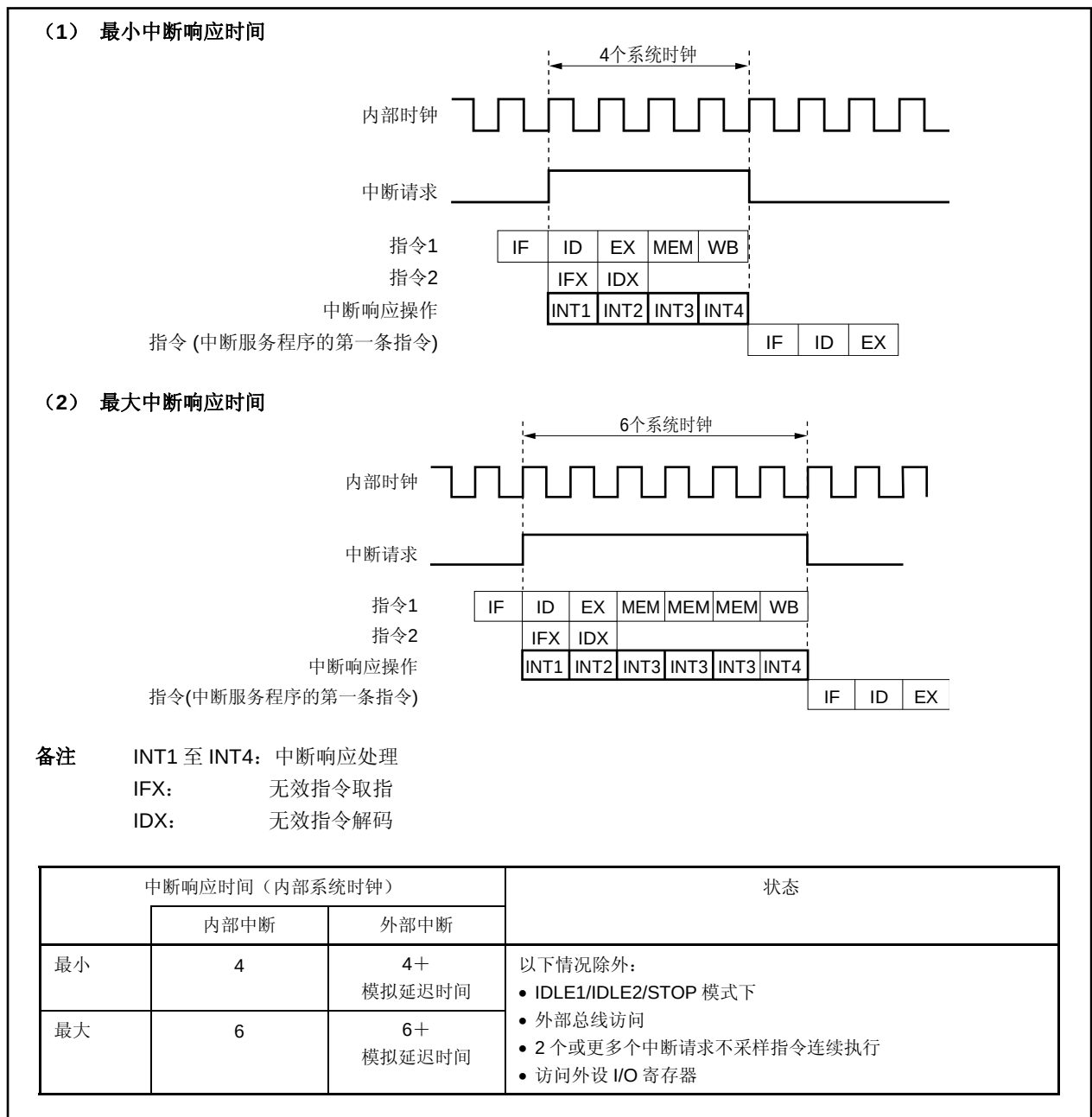
- 备注**
1. 因为采样执行了 3 次，则可靠消除噪音需要 2 个采样时钟宽度。
 2. 当噪声宽度小于 2 个采样时钟时，如果噪声与采样时钟的输入同步，就会产生一个中断请求信号。

19.7 CPU 中断响应时间

除了下列情况外，CPU 的中断响应时间最小为 4 个时钟周期。若要连续输入中断请求信号，至少需要在前一中断之后 5 个时钟周期才能输入下一中断请求信号。

- IDLE1/IDLE2/STOP 模式下
- 当访问外部总线时
- 当中断请求不采样指令被连续执行时（参见 19.8 中断不被 CPU 响应期间）。
- 当访问中断控制寄存器时

图 19-15. 中断请求信号响应的流水作业（概述）



19.8 中断不被 CPU 响应期间

当指令正在执行时，CPU 可以响应中断。但是，在中断请求不采样指令和下一条指令之间不会响应中断（中断挂起）。

中断请求不采样指令如下所示：

- EI 指令
- DI 指令
- LDSR reg2, 0x5 指令（对于 PSW）
- PRCMD 寄存器的存储指令
- 对下列寄存器的存储指令和 SET1, NOT1, CLR1 操作指令
 - 中断相关的寄存器：
中断控制寄存器（xxICn），中断屏蔽寄存器 0 至 4（IMR0 至 IMR4）
 - 节电控制寄存器（PSC）
 - 片上调试模式寄存器（OCDM）

备注 xx: 每一个外设单元的标志名称（参见 表 19-2 中断控制寄存器（xxICn））。

 n : 外设单元编号（参见 表 19-2 中断控制寄存器（xxICn））。

19.9 注意事项

NMI 引脚和 P02 引脚是复用功能引脚，而且在复位后作为正常端口引脚功能。要使能 NMI 引脚，通过 PMC0 寄存器使 NMI 引脚有效。NMI 引脚的初始设置是“无边沿检测”。使用 INTF0 和 INTR0 寄存器选择 NMI 引脚有效边沿。

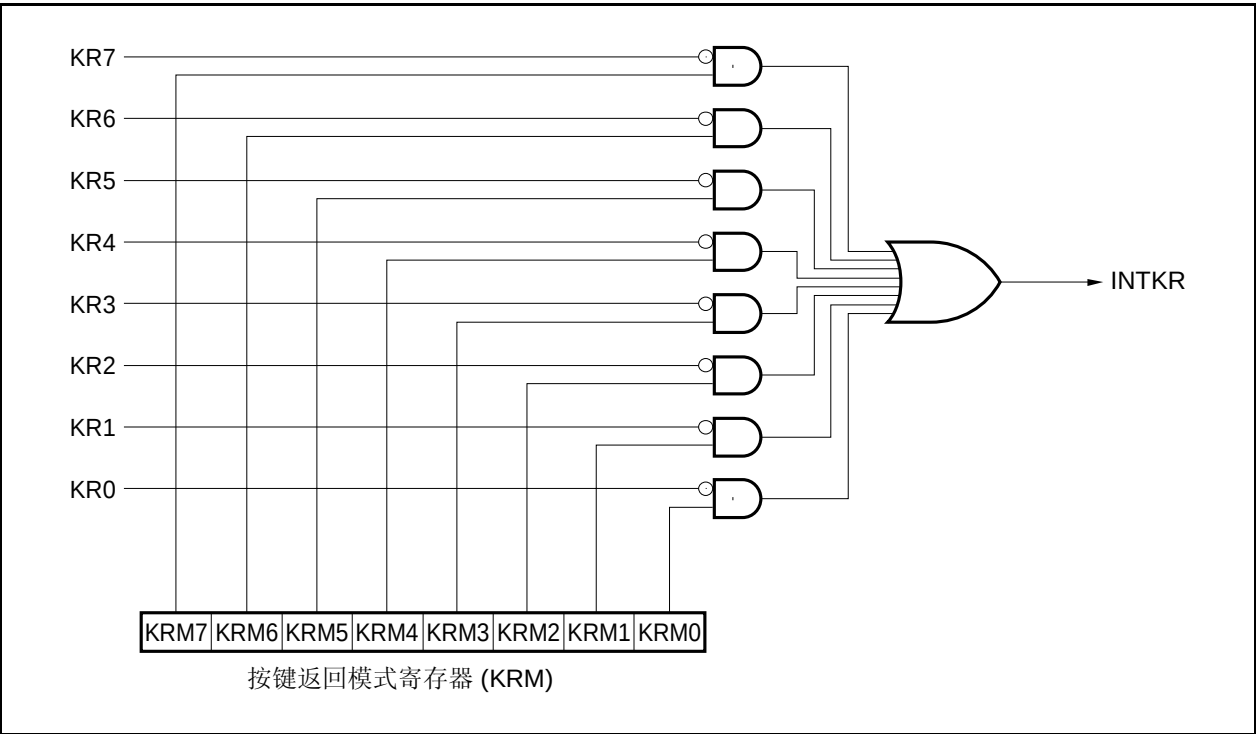
20.1 功能

通过设置 KRM 寄存器，向八个按键输入引脚（KR0 至 KR7）输入一个下降沿，可以产生一个按键中断请求信号（INTKR）。

表 20-1. 按键返回检测引脚的分配

标志	引脚说明
KRM0	用 1 位单元控制 KR0 信号
KRM1	用 1 位单元控制 KR1 信号
KRM2	用 1 位单元控制 KR2 信号
KRM3	用 1 位单元控制 KR3 信号
KRM4	用 1 位单元控制 KR4 信号
KRM5	用 1 位单元控制 KR5 信号
KRM6	用 1 位单元控制 KR6 信号
KRM7	用 1 位单元控制 KR7 信号

图 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 寄存器控制 KRM0 至 KRM7 位来使用 KR0 至 KR7 信号。

该寄存器可以按字节或按位读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H R/W 地址: FFFFF300H

	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	按键返回模式控制
0	禁止检测按键返回信号
1	检测按键返回信号

注意事项 将 KRM 寄存器进行一次清零（00H）后，再进行重写操作。

备注 关于复用功能引脚的设置，参见表 4-19 端口引脚用作复用功能引脚。

20.3 注意事项

- (1) 如果 KR0 至 KR7 中有任意某个引脚输入低电平，即使有下降沿输入其它引脚，也不会产生 INTKR 信号。
- (2) RXDA1 和 KR7 引脚不能同时使用。使用 RXDA1 引脚时，不要使用 KR7 引脚。若使用 KR7 引脚，就不要使用 RXDA1 引脚（建议将 PFC91 位设置为 1，并清除 PFCE91 位为 0）。
- (3) 如果改变 KRM 寄存器，就可能产生一个中断请求信号（INTKR）。为防止此种情况，在禁止中断（DI）或屏蔽之后再改变 KRM 寄存器，然后，将中断请求标志（KRIC.KRIF 位）清零后，允许中断（EI）或取消屏蔽。
- (4) 若要使用按键中断功能，需确保将端口引脚设置为按键返回引脚，然后允许 KRM 寄存器操作。若要从按键返回引脚切换至端口引脚，需要禁止 KRM 寄存器操作，然后再设置端口引脚。

第二十一章 待机功能

21.1 概述

通过结合运用待机模式并为应用程序选择恰当的模式，可以有效减少系统的功耗。表 21-1 列出了可用待机模式：

表 21-1. 待机模式

模式	功能概述
HALT 模式	此模式下，仅停止 CPU 的工作时钟。
IDLE1 模式	此模式下，除振荡器，PLL ^注 和flash存储器之外，停止其它所有内部电路的操作。
IDLE2 模式	此模式下，除振荡器外，停止其它所有内部电路的操作。
STOP 模式	此模式下，除副时钟振荡器外，停止其它所有内部电路的操作。
副时钟工作模式	此模式下，副时钟用作内部系统时钟。
副 IDLE 模式	副时钟工作模式下，除振荡器外，停止其它所有内部电路的操作。

注 PLL 保持模式转换之前的瞬时操作状态。



21.2 寄存器

(1) 节电控制寄存器 (PSC)

PSC 寄存器是 8 位寄存器，用于控制待机功能。该寄存器的 STP 位用于指定 STOP 模式。该寄存器是专用寄存器，只能通过特定的序列组合来写入（参见 3.4.7 专用寄存器）。

该寄存器可以按字节或位进行读写。

系统复位后，该寄存器被设置为 00H。

复位后：00H R/W 地址：FFFF1FEH

	7	<6>	<5>	<4>	3	2	<1>	0
PSC	0	NMI1M	NMI0M	INTM	0	0	STP	0

NMI1M	INTWDT2信号发生时解除待机模式的控制
0	允许通过INTWDT2信号解除待机模式
1	禁止通过INTWDT2信号解除待机模式

NMI0M	通过 NMI 引脚输入控制待机模式解除
0	NMI引脚输入有效时解除待机模式
1	NMI引脚输入无效时解除待机模式

INTM	通过可屏蔽中断请求信号控制待机模式的解除
0	可屏蔽中断请求信号有效时解除待机模式
1	可屏蔽中断请求信号无效时解除待机模式

STP	待机模式 ^注 设置
0	正常模式
1	待机模式

注 可以由 STP 位设置的待机模式有 IDLE1，IDLE2，STOP 或副 IDLE 模式。

- 注意事项**
1. 设置 IDLE1，IDLE2，STOP，或副 IDLE 模式之前，设置 PSMR.PSM1 和 PSMR.PSM0 位，然后再设置 STP 位。
 2. 当 HALT 模式解除后，NMI1M，NMI0M，和 INTM 位的设置无效。
 3. 如果在 NMI1M，NMI0M，或 INTM 位设为 1 的同时把 STP 位也设为 1，则对 NMI1M，NMI0M，或 INTM 位的设置无效。如果在设置 IDLE1/IDLE2/STOP 模式时有未屏蔽中断请求信号处于挂起状态，则应设置中断请求信号对应的位（NMI1M，NMI0M，或 INTM）为 1，然后再设置 STP 位为 1。

(2) 节电模式寄存器 (PSMR)

PSMR 寄存器是 8 位寄存器，用于控制节电模式下的操作状态及时钟操作。

该寄存器可按字节或按位进行读写。

复位输入将该寄存器设置为 00H。

复位后：00H R/W 地址：FFFFF820H

	7	6	5	4	3	2	<1>	<0>
PSMR	0	0	0	0	0	0	PSM1	PSM0

PSM1	PSM0	软件待机模式下的操作说明
0	0	IDLE1，副IDLE 模式
0	1	STOP 模式
1	0	IDLE2，副IDLE 模式
1	1	STOP 模式

- 注意事项
- 1. 请确保将第 2 位至第 7 位清为“0”。
 - 2. 仅当 PSC.STP 位为 1 时，PSM0 和 PSM1 位才有效。

- 备注
- IDLE1: 此模式下，除振荡器和其它一些电路（flash 存储器和 PLL）之外，停止所有的操作。在解除 IDLE1 模式后，无需经过振荡稳定保证时间，就可以恢复正常操作模式。这和 HALT 模式一样。
 - IDLE2: 此模式下，除振荡器之外，所有的操作停止。解除 IDLE2 模式后，在经过 OSTS 寄存器指定的振荡稳定时间后，恢复正常操作模式。
 - STOP: 此模式下，除副时钟振荡器之外，所有的操作停止。释放 STOP 模式后，在经过 OSTS 寄存器指定的振荡稳定时间后，恢复正常操作模式。
 - 副 IDLE: 此模式下，除振荡器之外，暂停所有的操作。在中断请求信号解除 IDLE 模式后，经过 12 个副时钟周期的保证时间后，恢复副时钟操作模式。

(3) 振荡稳定时间选择寄存器 (OSTS)

在释放 STOP 模式之后至振荡稳定之前需要等待的时间，或在解除 IDLE2 模式之后至片上 flash 存储器稳定之前需要等待的时间，由 OSTS 寄存器控制。

OSTS 寄存器可以按字节进行读写。

复位输入将寄存器设置为 06H。

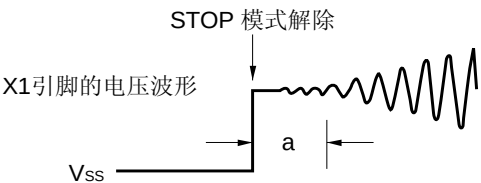
复位后: 06H R/W 地址: FFFFF6C0H

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	振荡稳定时间/建立时间的选择 ^注	f _x	
				4 MHz	5 MHz
0	0	0	2 ¹⁰ /f _x	0.256 ms	0.205 ms
0	0	1	2 ¹¹ /f _x	0.512 ms	0.410 ms
0	1	0	2 ¹² /f _x	1.024 ms	0.819 ms
0	1	1	2 ¹³ /f _x	2.048 ms	1.638 ms
1	0	0	2 ¹⁴ /f _x	4.096 ms	3.277 ms
1	0	1	2 ¹⁵ /f _x	8.192 ms	6.554 ms
1	1	0	2 ¹⁶ /f _x	16.38 ms	13.107 ms
1	1	1	禁止设置		

注 当释放 STOP 模式和 IDLE2 模式时，需要不同的振荡稳定时间和建立时间。

注意事项 1. 不管是因为复位输入还是因为中断请求信号解除了 STOP 模式，在释放 STOP 模式之后的等待时间不包括时钟开始振荡之前（下图的“a”）的那段时间。



- 2. 请确保将第 3 位至第 7 位清为“0”。
- 3. 复位解除后的振荡稳定时间是 2¹⁶/f_x（因为OSTS寄存器的初始值= 06H）。

备注 f_x=主时钟振荡器频率

21.3 HALT 模式

21.3.1 设置和操作状态

在正常操作模式下，执行一条专用指令（HALT），可以设置为 HALT 模式。

在 HALT 模式下，时钟振荡器继续工作。只是停止 CPU 的时钟供给；其它片上外设功能的时钟不受影响。

结果，程序停止执行，内部 RAM 保持设置为 HALT 模式之前的内容。独立于 CPU 指令处理的片上外设功能继续工作。

表 21-3 显示了 HALT 模式下的操作状态。

通过 HALT 模式和正常操作模式的间歇结合操作，可以减少系统的平均电流消耗。

注意事项 1. 在 HALT 指令后，插入五个或更多的 NOP 指令。

2. 如果在执行 HALT 指令时有一个未屏蔽的中断请求信号被挂起，则状态切换至 HALT 模式，但是 HALT 模式立即被该中断请求解除。

21.3.2 解除 HALT 模式

在 HALT 模式下，有多种方式可以解除 HALT 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽的外部中断请求信号（INTP0 至 INTP8 引脚输入）、来自 HALT 模式下可操作外设的未屏蔽的内部中断请求信号，或复位信号（由 RESET 引脚输入的复位、WDT2RES 信号，低电压检测器（LVI），或时钟监视器（CLM））。

HALT 模式释放后，恢复为正常操作模式。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号解除 HALT 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号，可以解除 HALT 模式，且与中断请求信号的优先级无关。但是，如果在一个中断服务程序中设置 HALT 模式，则之后发生的中断请求信号的服务如下：

（a）如果发出的中断请求信号优先级低于当前正在服务的中断请求，则仅解除 HALT 模式，并且对该中断请求信号不作响应。该中断请求信号被保持。

（b）如果发出的中断请求信号优先级高于当前正在服务的中断请求（包括不可屏蔽中断请求信号），则 HALT 模式解除，且响应该中断请求信号。

表 21-2. 用中断请求信号解除 HALT 模式后的操作

解除源	中断允许（EI）状态	中断禁止（DI）状态
不可屏蔽中断请求信号	执行对句柄地址的跳转。	
可屏蔽中断请求信号	执行对句柄地址的跳转或执行下一条指令。	执行下一条指令。

(2) 通过复位解除 HALT 模式

执行的操作与正常的复位操作相同。

表 21-3. HALT 模式下的操作状态

项目		运行状态	
		不使用副时钟时	使用副时钟时
主时钟振荡器		振荡使能	
副时钟振荡器		—	振荡使能
内部振荡器		振荡使能	
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 至 TMP8)		可操作	
定时器 Q (TMQ0)		可操作	
定时器 M (TMM0)		当选择 f_{XT} 之外的时钟为计数时钟时，可操作	可操作
钟表定时器		当 f_x (BRG 分频) 选择为计数时钟时，可操作	可操作
看门狗定时器 2		当选择 f_{XT} 之外的时钟为计数时钟时，可操作	可操作
串行接口	CSIB0 至 CSIB5	可操作	
	I ² C00 至 I ² C02	可操作	
	UARTA0 至 UARTA3	可操作	
A/D 转换器		可操作	
D/A 转换器		可操作	
实时输出功能 (RTO)		可操作	
按键中断功能 (KR)		可操作	
CRC 操作电路		可操作 (没有数据输入到 CRCIN 寄存器，因为 CPU 已经停止)	
外部总线接口		参见 2.2 引脚状态。	
端口功能		保持设置为 HALT 模式前的状态。	
内部数据		CPU 寄存器、状态、数据及其它内部数据，例如内部 RAM 的内容，被保持为设置 HALT 模式之前的状态。	

21.4 IDLE1 模式

21.4.1 设置和操作状态

在正常操作模式下，通过将 PSMR.PSM1 和 PSMR.PSM0 位清 00，并且将 PSC.STP 设置为 1，可以设置 IDLE1 模式。

在 IDLE1 模式下，时钟振荡器、PLL、以及 flash 存储器继续操作，但停止 CPU 和其它片上外设功能的时钟供给。

结果，程序停止执行，且内部 RAM 保持与设置为 IDLE1 模式前的内容。CPU 及其它片上外设功能停止工作。但是，可以在副时钟或外部时钟下工作的片上外设功能继续工作。

表 21-5 显示了 IDLE1 模式下的操作状态。

与 HALT 模式相比，IDLE1 模式可以减少更多功耗，因为停止了片上外设功能的运行。主时钟振荡器不停止，所以在解除 IDLE1 模式后，无需等待振荡稳定时间就可以恢复到正常操作模式，HALT 模式解除时也是如此。

注意事项 1，为了设置 IDLE1 模式，需要将数据存储到 PSC 寄存器，在存储指令之后，应该插入五个或更多 NOP 指令。

2。如果在设置 IDLE1 模式时，有一个未屏蔽的中断请求信号被挂起，则 IDLE1 模式立即被这个等待响应的中断请求解除。

21.4.2 解除 IDLE1 模式

在 IDLE1 模式下，有多种方式可以解除 IDLE1 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽外部中断请求信号（INTP0 至 INTP8 引脚输入）、来自 IDLE1 模式下可操作的外设功能的未屏蔽的内部中断请求信号，或复位信号（由 RESET 引脚输入复位、WDT2RES 信号，低电压检测器（LVI），或时钟监视器（CLM））。

IDLE1 模式释放后，恢复为正常操作模式。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号来解除 IDLE1 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号，可以解除 IDLE1 模式，且与中断请求信号的优先级无关。如果在一个中断服务程序中设置 IDLE1 模式，则之后产生的中断请求信号的服务如下：

（a）如果发出的中断请求信号优先级低于或等于当前正在服务的中断请求，则仅解除 IDLE1 模式，并且对该中断请求信号不作响应。该中断请求信号被保持。

（b）如果发出的中断请求信号优先级高于当前正在服务的中断请求（包括不可屏蔽中断请求信号），则解除 IDLE1 模式，且响应该中断请求信号。

注意事项 通过 PSC.NMI1M、PSC.NMI0M 和 PSC.INTM 位设置为 1，一个中断请求信号将被禁止，该中断请求变为无效，并且 IDLE1 模式不会被解除。

表 21-4.通过中断请求信号解除 IDLE1 模式后的操作

解除源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	执行对句柄地址的跳转。	
可屏蔽中断请求信号	执行对句柄地址的跳转或执行下一条指令。	执行下一条指令。

(2) 通过复位解除 IDLE1 模式

执行的操作与正常的复位操作相同。

表 21-5. IDLE1 模式下的操作状态

项目		运行状态	
		不使用副时钟时	使用副时钟时
主时钟振荡器		振荡使能	
副时钟振荡器		—	振荡使能
内部振荡器		振荡使能	
PLL		可操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是待机模式可以释放）	
定时器 P（TMP0 至 TMP8）		停止操作	
定时器 Q（TMQ0）		停止操作	
定时器 M（TMM0）		当 $f_{R/8}$ 选择为计数时钟时，可操作	当 $f_{R/8}$ 或 f_{XT} 选择为计数时钟时，可操作
钟表定时器		当 f_x （BRG 分频）选择为计数时钟时，可操作	可操作
看门狗定时器 2		当 f_R 选择为计数时钟时，可操作	当 f_R 或 f_{XT} 选择为计数时钟时，可操作
串行接口	CSIB0 至 CSIB5	$\overline{SCKBn}$ 输入时钟被选为计数时钟时，可操作（ $n=0$ 至 5）	
	I ² C00 至 I ² C02	停止操作	
	UARTA0 至 UARTA3	停止操作（但是当选择 ASCKA0 引脚时钟时，UARTA0 可操作）	
A/D 转换器		保持操作（转换结果保持） [※]	
D/A 转换器		保持操作（输出保持 [※] ）	
实时输出功能（RTO）		停止操作（输出保持）	
按键中断功能（KR）		可操作	
CRC 操作电路		停止操作	
外部总线接口		参见 2.2 引脚状态。	
端口功能		保持设置为 IDLE1 模式前的状态	
内部数据		CPU 寄存器、状态位、数据及其它内部数据，例如内部 RAM 的内容，都保持为设置 IDLE1 模式之前的状态。	

注 为了降低功率损耗，在切换到 IDLE1 模式之前，停止 A/D 转换器和 D/A 转换器。

21.5 IDLE2 模式

21.5.1 设置和操作状态

在正常操作模式下，通过将 PSMR.PSM1 和 PSMR.PSM0 位设置为 10，并且将 PSC.STP 设置为 1，可以设置为 IDLE2 模式。

在 IDLE2 模式下，时钟振荡器继续操作，但停止对 CPU、PLL、flash 存储器以及其它片上外设功能的时钟供给。

结果，程序停止执行，且内部 RAM 保持与设置为 IDLE2 模式前的内容。CPU、PLL 以及其它片上外设功能停止工作。但是，可以在副时钟或外部时钟下工作的片上外设功能继续工作。

表 21-7 显示了 IDLE2 模式下的操作状态。

与 IDLE1 模式相比，IDLE2 模式可以减少更多功耗，因为停止了片上外设功能、PLL 和 flash 存储器的运行。但是，因为 PLL 和 flash 存储器停止，当解除 IDLE2 模式时，PLL 和 flash 存储器需要一段建立时间。

注意事项 1. 为了设置 IDLE2 模式，需要将数据存储到 PSC 寄存器，在存储指令之后，应该插入五个或更多 NOP 指令。

2. 如果在设置 IDLE2 模式时，有一个未屏蔽的中断请求信号被挂起，则 IDLE2 模式立即被这个等待响应的中断请求解除。

21.5.2 释放 IDLE2 模式

在 IDLE2 模式下，有多种方式可以解除 IDLE2 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽外部中断请求信号（INTP0 至 INTP8 引脚输入）、来自 IDLE2 模式下可操作的外设功能的未屏蔽的内部中断请求信号，或复位信号（由 RESET 引脚输入复位、WDT2RES 信号，低电压检测器（LVI），或时钟监视器（CLM））。PLL 返回到设置为 IDLE2 模式前的操作状态。

IDLE2 模式释放后，恢复为正常操作模式。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号来解除 IDLE2 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号，可以解除 IDLE2 模式，且与中断请求信号的优先级无关。如果在一个中断服务程序中设置 IDLE2 模式，则之后产生的中断请求信号的服务如下：

（a）如果发出的中断请求信号优先级低于当前正在服务的中断请求，则仅解除 IDLE2 模式，并且对该中断请求信号不作响应。该中断请求信号被保持。

（b）如果发出的中断请求信号优先级高于当前正在服务的中断请求（包括不可屏蔽中断请求信号），则解除 IDLE2 模式，且响应该中断请求信号。

注意事项 通过 PSC.NMI1M、PSC.NMI0M 和 PSC.INTM 位设置为 1，一个中断请求信号将被禁止，该中断请求变为无效，并且 IDLE2 模式不会被解除。

表 21-6. 通过中断请求信号解除 IDLE2 模式后的操作

解除源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	在确保的建立时间过后, 执行对句柄地址的跳转。	
可屏蔽中断请求信号	在确保的建立时间之后, 执行对句柄地址的跳转或执行下一条指令。	在确保的建立时间之后执行下一条指令。

(2) 通过复位解除 IDLE2 模式

执行的操作与正常的复位操作相同。

表 21-7. IDLE2 模式下的操作状态

IDLE2 模式的设置 项目		运行状态	
		不使用副时钟时	使用副时钟时
主时钟振荡器		振荡使能	
副时钟振荡器		—	振荡使能
内部振荡器		振荡使能	
PLL		停止操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作 (但是待机模式解除允许)	
定时器 P (TMP0 至 TMP8)		停止操作	
定时器 Q (TMQ0)		停止操作	
定时器 M (TMM0)		当 $f_{R/8}$ 选择为计数时钟时可操作	当 $f_{R/8}$ 或 f_{XT} 选择为计数时钟时可操作
钟表定时器		当 f_x (BRG 分频) 选择为计数时钟时可操作	可操作
看门狗定时器 2		当 f_R 选择为计数时钟时可操作	当 f_R 或 f_{XT} 选择为计数时钟时可操作
串行接口	CSIB0 至 CSIB5	$\overline{SCKBn}$ 输入时钟被选为计数时钟时可操作 ($n=0$ 至 5)	
	I ² C00 至 I ² C02	停止操作	
	UARTA0 至 UARTA3	停止操作 (但是当选择 ASCKA0 引脚时钟时, UARTA0 可操作)	
A/D 转换器		保持操作 (转换结果保持) [※]	
D/A 转换器		保持操作 (输出保持) [※]	
实时输出功能 (RTO)		停止操作 (输出保持)	
按键中断功能 (KR)		可操作	
CRC 操作电路		停止操作	
外部总线接口		参见 2.2 引脚状态。	
端口功能		保持设置为 IDLE2 模式前的状态	
内部数据		CPU 寄存器、状态位、数据及其它内部数据, 例如内部 RAM 的内容, 都保持为设置 IDLE2 模式之前的状态。	

注 为了降低功率损耗, 在切换到 IDLE2 模式之前, 停止 A/D 转换器和 D/A 转换器。

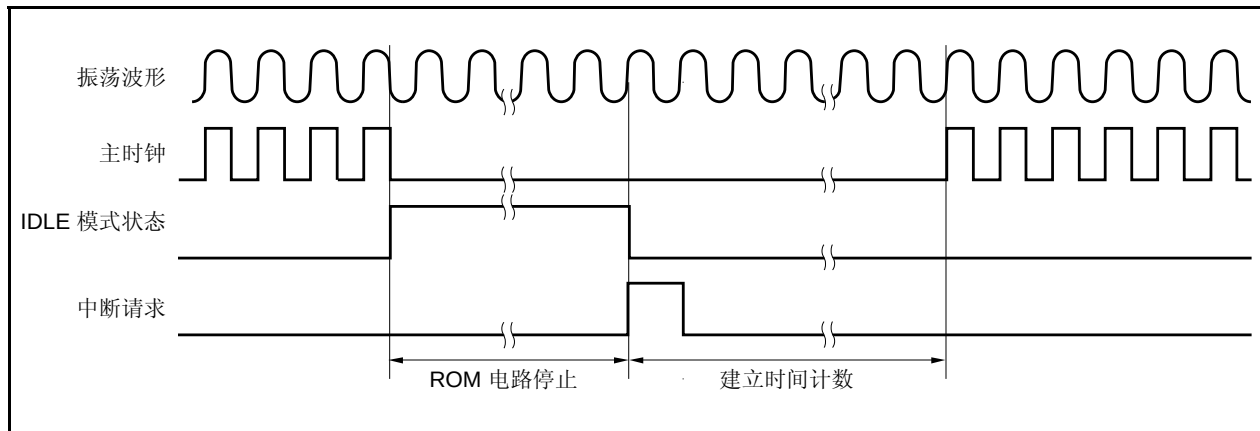
21.5.3 解除 IDLE2 模式时确保建立时间

在解除 IDLE2 模式之后, 要确保 flash 存储器的建立时间, 这是因为在设置 IDLE2 模式后, 除了主时钟振荡器之外的模块都停止操作。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号来解除 IDLE2 模式

设置 OSTS 寄存器，确保指定的建立时间。

当产生释放源时，依据 OSTS 寄存器的设置，专用内部定时器开始计数。当其溢出时，恢复正常工作模式。



(2) 通过复位解除 ($\overline{\text{RESET}}$ 引脚输入, WDT2RES 产生)

此操作与正常复位的操作相同。

振荡稳定时间是 OSTS 寄存器的初始值, $2^{16}/f_x$ 。

21.6 STOP 模式

21.6.1 设置和操作状态

在正常操作模式下，将 PSMR.PSM1 和 PSMR.PSM0 位设为 01 或 11，并将 PSC.STP 设为 1，可以设置为 STOP 模式。

在 STOP 模式下，副时钟振荡器继续工作，但主时钟振荡器停止。停止对 CPU 和片上外设功能的时钟供给。

结果，程序停止执行，且内部 RAM 保持与设置为设置 STOP 模式之前的内容。运行于副时钟振荡器或外部时钟的片上外围功能继续工作。

表 21-9 显示了在 STOP 模式下的操作状态。

因为 STOP 模式使主时钟振荡器的运行停止，所以可将功耗降低到低于 IDLE2 模式的水平。若不使用副时钟振荡器、内部振荡器和外部时钟，功耗达到最小化，仅有漏电流通过。

- 注意事项**
1. 为了设置 STOP 模式，需要将数据存储在 PSC 寄存器，在存储指令之后，应该插入五个或更多 NOP 指令。
 2. 如果在设置 STOP 模式时，有一个未屏蔽的中断请求信号被挂起，则 STOP 模式立即被这个等待响应的中断请求解除。

21.6.2 释放 STOP 模式

有多种方式可以释放 STOP 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽外部中断请求信号（INTP0 至 INTP8 引脚输入）、来自 STOP 模式下可操作的外设功能的未屏蔽的内部中断请求信号，或复位信号（由 RESET 引脚输入、WDT2RES 信号，低电压检测器（LVI）引发的复位）。

在 STOP 模式解除后，在经过保证的振荡稳定时间后，恢复为正常操作模式。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号来释放 STOP 模式

通过不可屏蔽中断请求信号或未屏蔽中断请求信号，可释放 STOP 模式，且与中断请求信号的优先级无关。但是，如果在一个中断服务程序中设置 STOP 模式，则对之后发出的中断请求信号的服务如下。

- （a）如果发出的中断请求信号优先级低于当前正在服务的中断请求，则释放 STOP 模式，并且对该中断请求信号不作响应。该中断请求信号被保持。
- （b）如果发出的中断请求信号优先级高于当前正在服务的中断请求（包括不可屏蔽中断请求信号），则释放 STOP 模式，且响应该中断请求信号。

注意事项 通过 PSC.NMI1M、PSC.NMI0M 和 PSC.INTM 位设置为 1，一个中断请求信号将被禁止，该中断请求变为无效，并且 STOP 模式不会被解除。

表 21-8. 通过中断请求信号释放 STOP 模式之后的操作

解除源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	在经过确保的振荡稳定时间之后，执行对句柄地址的跳转。	
可屏蔽中断请求信号	在经过确保的振荡稳定时间之后，执行对句柄地址的跳转或执行下一条指令。	在经过确保的振荡稳定时间之后，执行下一条指令。

(2) 通过复位释放 STOP 模式

执行的操作与正常的复位操作相同。

表 21-9. STOP 模式的操作状态

STOP 模式的设置 项目		运行状态	
		不使用副时钟时	使用副时钟时
主时钟振荡器		振荡停止	
副时钟振荡器		—	振荡使能
内部振荡器		振荡使能	
PLL		停止操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是待机模式解除允许）	
定时器 P（TMP0 至 TMP8）		停止操作	
定时器 Q（TMQ0）		停止操作	
定时器 M（TMM0）		当 $f_{R/8}$ 选择为计数时钟时，可操作	当 $f_{R/8}$ 或 f_{XT} 选择为计数时钟时，可操作
钟表定时器		停止操作	当 f_{XT} 选择为计数时钟时，可操作
看门狗定时器 2		当 f_R 选择为计数时钟时，可操作	当 f_R 或 f_{XT} 选择为计数时钟时，可操作
串行接口	CSIB0 至 CSIB5	SCKBn 输入时钟被选为计数时钟时可操作（n=0 至 5）	
	I ² C00 至 I ² C02	停止操作	
	UARTA0 至 UARTA3	停止操作（但是当选择 ASCKA0 输入时钟时，UARTA0 可以操作）	
A/D 转换器		停止操作（转换结果不确定） ^{注 1, 2}	
D/A 转换器		停止操作 ^{注 3, 4} （高阻抗输出）	
实时输出功能（RTO）		停止操作（输出保持）	
按键中断功能（KR）		可操作	
CRC 操作电路		停止操作	
外部总线接口		参见 2.2 引脚状态。	
端口功能		保持设置为 STOP 模式前的状态。	
内部数据		CPU 寄存器、状态位、数据及其它内部数据，例如内部 RAM 的内容，都保持为设置 STOP 模式之前的状态。	

- 注
1. 如果在 A/D 转换器操作时设置 STOP 模式，则 A/D 转换器自动停止，并且在 STOP 模式释放后再次开始操作。但是这种情况下，STOP 模式释放后的 A/D 转换结果无效。STOP 模式设置前的所有 A/D 转换值都无效。
 2. 即使在 A/D 转换器操作时设置 STOP 模式，功耗也同样减少，和设置 STOP 模式前停止 A/D 转换器的功耗相等。
 3. 如果在 D/A 转换器操作时设置 STOP 模式，那么 D/A 转换器自动停止并且引脚状态变为高阻。STOP 模式解除后，恢复 D/A 转换，设定的时间过后，返回到设置 STOP 模式之前的输出电平状态。
 4. 即使在 D/A 转换器操作时设置 STOP 模式，功耗也同样减少，和设置 STOP 模式前停止 D/A 转换器的功耗相等。

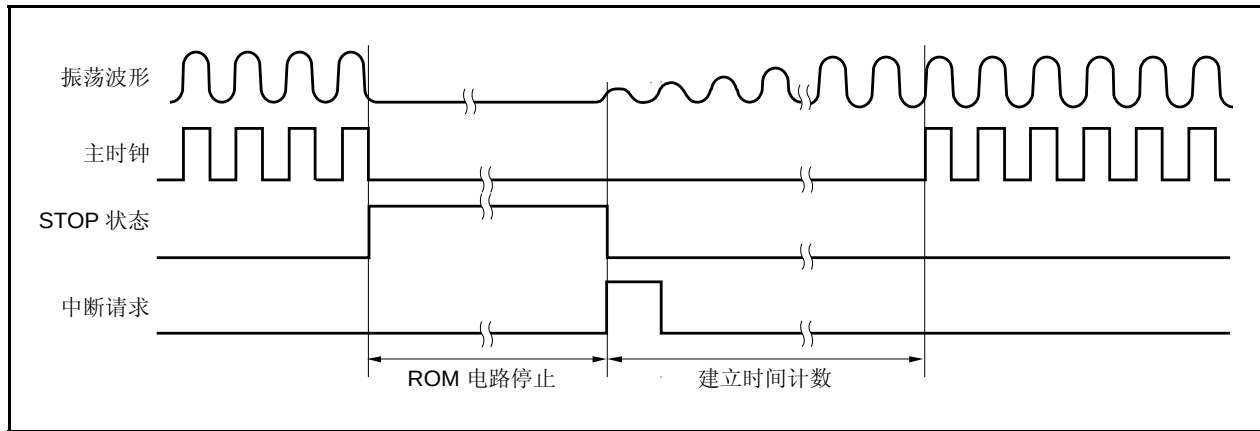
21.6.3 在释放 STOP 模式时确保振荡稳定时间

为了主时钟振荡器，在释放 STOP 模式后需要一段时间来确保主时钟振荡器的振荡稳定，因为在设置 STOP 模式之后，主时钟振荡器的操作停止。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号来释放 STOP 模式

设置 OSTS 寄存器，确保振荡稳定时间。

当产生释放源时，依据 OSTS 寄存器的设置，专用内部定时器开始计数。当其溢出时，恢复正常工作模式。



(2) 通过复位解除

执行的操作与正常的复位操作相同。

振荡稳定时间是 OSTS 寄存器的初始值， $2^{16}/f_x$ 。

21.7 副时钟操作模式

21.7.1 设置和操作状态

在正常操作模式下，把 PCC.CK3 位设置为 1，可以设置为副时钟工作模式。

当设置为副时钟工作模式时，内部系统时钟从主时钟变为副时钟。通过 PCC.CLS 位来检查时钟是否已经完成转换。

当 PCC.MCK 位设置为 1 时，主时钟振荡器停止运行。结果，系统仅运行于副时钟。

在副时钟工作模式下，功耗可以降至低于正常操作模式的级别，因为副时钟用作内部系统时钟。此外，停止主时钟振荡器的运行，功耗可进一步降低至 STOP 模式的级别。

图 21-10 显示了副时钟操作模式下的操作状态。

- 注意事项**
1. 在对 CK3 位进行操作时，不要改变 PCC.CK2 至 PCC.CK0 位的设置值（建议用位操作指令对此位进行操作）。PCC 寄存器的详情，参见“6.3（1）处理器时钟控制寄存器（PCC）”。
 2. 若下列条件不满足，则改变 CK2 至 CK0 位使条件能够满足，并设置为副时钟工作模式。
内部系统时钟（f_{CLK}）> 副时钟（f_{XT}=32.768kHz）× 4

备注 内部系统时钟（f_{CLK}）： 根据 CK2 至 CK0 位的设置，由主时钟（f_{XX}）产生的时钟。

21.7.2 解除副时钟操作模式

当 CK3 位被清除为 0 时，通过复位信号（RESET 引脚的输入、WDT2RES 信号、低电压检测器（LVI）或时钟监视器（CLM）引发的复位）来解除副时钟操作模式。

如果主时钟停止（MCK 位 = 1），将 MCK 位置为 1，用软件保证主时钟的振荡稳定时间，并将 CK3 位清除为 0。

当副时钟工作模式解除时，恢复为正常操作模式。

- 注意事项** 在操作 CK3 位时，不要改变 CK2 至 CK0 位的设置值（建议用位操作指令对此位进行操作）。
PCC 寄存器的详情，参见“6.3（1）处理器时钟控制寄存器（PCC）”。

表 21-10. 副时钟工作模式的操作状态

副时钟工作模式的设置 项目		运行状态	
		主时钟振荡时	主时钟停止时
副时钟振荡器		振荡使能	
内部振荡器		振荡使能	
PLL		可操作	停止操作 ^注
CPU		可操作	
DMA		可操作	
中断控制器		可操作	
定时器 P（TMP0 至 TMP8）		可操作	停止操作
定时器 Q（TMQ0）		可操作	停止操作
定时器 M（TMM0）		可操作	当 $f_R/8$ 或 f_{XT} 选择为计数时钟时，可操作
钟表定时器		可操作	当 f_{XT} 选择为计数时钟时，可操作
看门狗定时器 2		可操作	当 f_R 或 f_{XT} 选择为计数时钟时，可操作
串行接口	CSIB0 至 CSIB5	可操作	$\overline{SCKBn}$ 输入时钟被选为计数时钟时，可操作（ $n=0$ 至 5）
	I ² C00 至 I ² C02	可操作	停止操作
	UARTA0 至 UARTA3	可操作	停止操作（但是当选择 ASCKA0 引脚时钟时，UARTA0，可操作）
A/D 转换器		可操作	停止操作
D/A 转换器		可操作	
实时输出功能（RTO）		可操作	停止操作（输出保持）
按键中断功能（KR）		可操作	
CRC 操作电路		可操作	
外部总线接口		参见 2.2 引脚状态。	
端口功能		可设置	
内部数据		可设置	

注 在停止主时钟前，请确保停止锁相环（PLLCTL.PLLON 位=0）。

注意事项 当 CPU 运行于副时钟而且主时钟振荡停止时，在发生等待时禁止访问寄存器。如果已经产生等待，只能通过复位解除（参见 3.4.8（2））。

21.8 副 IDLE 模式

21.8.1 设置和操作状态

在副时钟工作模式下，将 PSMR.PSM1 和 PSMR.PSM0 位设置为 00 或 10 并将 PSC.STP 位设置为 1，可以设置为副 IDLE 模式。

该模式下，时钟振荡器继续工作，但停止对 CPU、Flash 存储器及其它片上外设功能的时钟供给。

结果，程序停止执行，且内部 RAM 保持为设置副 IDLE 模式之前的内容。CPU 及其它片上外设功能停止。但是，运行于副时钟或外部时钟的片上外设功能继续工作。

因为副 IDLE 模式停止了 CPU、flash 存储器及其它片上外设功能的操作，所以可以比副时钟工作模式降低更多功耗。如果在主时钟停止后设置为副 IDLE 模式，则该模式下的电流消耗可减少至与 STOP 模式同样低的水平。

表 21-12 显示了在副 IDLE 模式下的操作状态。

- 注意事项**
1. 为了设置副 IDLE 模式，需要将数据存储在 PSC 寄存器，在存储指令之后，应该插入五个或更多 NOP 指令。
 2. 如果在设置副 IDLE 模式时，有一个未屏蔽的中断请求信号被挂起，则副 IDLE 模式立即被这个等待响应的中断请求解除。

21.8.2 解除副 IDLE 模式

有多种方式可以解除副 IDLE 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽外部中断请求信号（INTP0 至 INTP8 引脚输入）、在副 IDLE 模式下可操作的外设功能的未屏蔽的内部中断请求信号，或复位信号（由 RESET 引脚输入、WDT2RES 信号，低电压检测器（LVI）或时钟监视器（CLM）引发的复位）。PLL 返回到设置副 IDLE 模式之前的操作状态。

通过中断请求信号解除副 IDLE 模式时，设置为副时钟工作模式。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号解除副 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，可解除副 IDLE 模式，且与中断请求信号的优先级无关。

但是，如果在一个中断服务程序中设置副 IDLE 模式，则对之后发出的中断请求信号的服务如下：

- （a）如果发出的中断请求信号优先级低于当前正在服务的中断请求，则副 IDLE 模式解除，并且对该中断请求信号不作响应。该中断请求信号被保持。
- （b）如果发出的中断请求信号优先级高于当前正在服务的中断请求（包括不可屏蔽中断请求信号），则副 IDLE 模式解除，且响应该中断请求信号。

- 注意事项**
1. 通过 PSC.NMI1M、PSC.NMI0M 和 PSC.INTM 位设置为 1，一个中断请求信号将被禁止，该中断请求变为无效，并且副 IDLE 模式不会被释放。
 2. 当副 IDLE 模式解除时，从产生解除副 IDLE 模式的中断请求信号至模式被解除需要经过副时钟的 12 个周期（大约 366 μ s）。

表 21-11. 通过中断请求信号解除副 IDLE 模式后的操作

解除源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	执行对句柄地址的跳转。	
可屏蔽中断请求信号	执行对句柄地址的跳转或执行下一条指令。	执行下一条指令。

(2) 通过复位解除副 IDLE 模式

执行的操作与正常的复位操作相同。

表 21-12. 副 IDLE 模式的操作状态

副 IDLE 模式的设置 项目		运行状态	
		主时钟振荡时	主时钟停止时
副时钟振荡器		振荡使能	
内部振荡器		振荡使能	
PLL		可操作	停止操作 ^{注1}
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是待机模式解除允许）	
定时器 P（TMP0 至 TMP8）		停止操作	
定时器 Q（TMQ0）		停止操作	
定时器 M（TMM0）		当 f_R 或 f_{XT} 选择为计数时钟时，可操作	
钟表定时器		可操作	当 f_{XT} 选择为计数时钟时，可操作
看门狗定时器 2		当 f_R 或 f_{XT} 选择为计数时钟时，可操作	
串行接口	CSIB0 至 CSIB5	SCKBn 输入时钟被选为计数时钟时，可操作（n=0 至 5）	
	I ² C00 至 I ² C02	停止操作	
	UARTA0 至 UARTA3	停止操作（但是当选择 ASCKA0 引脚时钟时，UARTA0 可操作）	
A/D 转换器		保持操作（转换结果保持） ^{注2}	
D/A 转换器		保持操作（输出保持） ^{注2}	
实时输出功能（RTO）		停止操作（输出保持）	
按键中断功能（KR）		可操作	
CRC 操作电路		停止操作	
外部总线接口		参见 2.2 引脚状态（与 IDLE1 和 IDLE2 模式相同的操作）。	
端口功能		保持设置为副 IDLE 模式前的状态。	
内部数据		CPU 寄存器、状态位、数据及其它内部数据，例如内部 RAM 的内容，都保持为设置副 IDLE 模式之前的状态。	

- 注
1. 在停止主时钟前，请确保停止锁相环（PLLCTL.PLLON 位=0）。
 2. 为了降低功率损耗，在进入副 IDLE 模式前，停止 A/D 转换器和 D/A 转换器。

第二十二章 复位功能

22.1 概述

可以使用以下复位功能：

- (1) 4 种类型的复位源
 - 通过 $\overline{\text{RESET}}$ 引脚的外部复位输入
 - 通过看门狗定时器 2 (WDT2) 的溢出复位 (WDT2RES)
 - 通过比较低电压检测器 (LVI) 供电电压和被检测电压而系统复位
 - 通过时钟监控器 (CLM) 检测振荡停止而系统复位

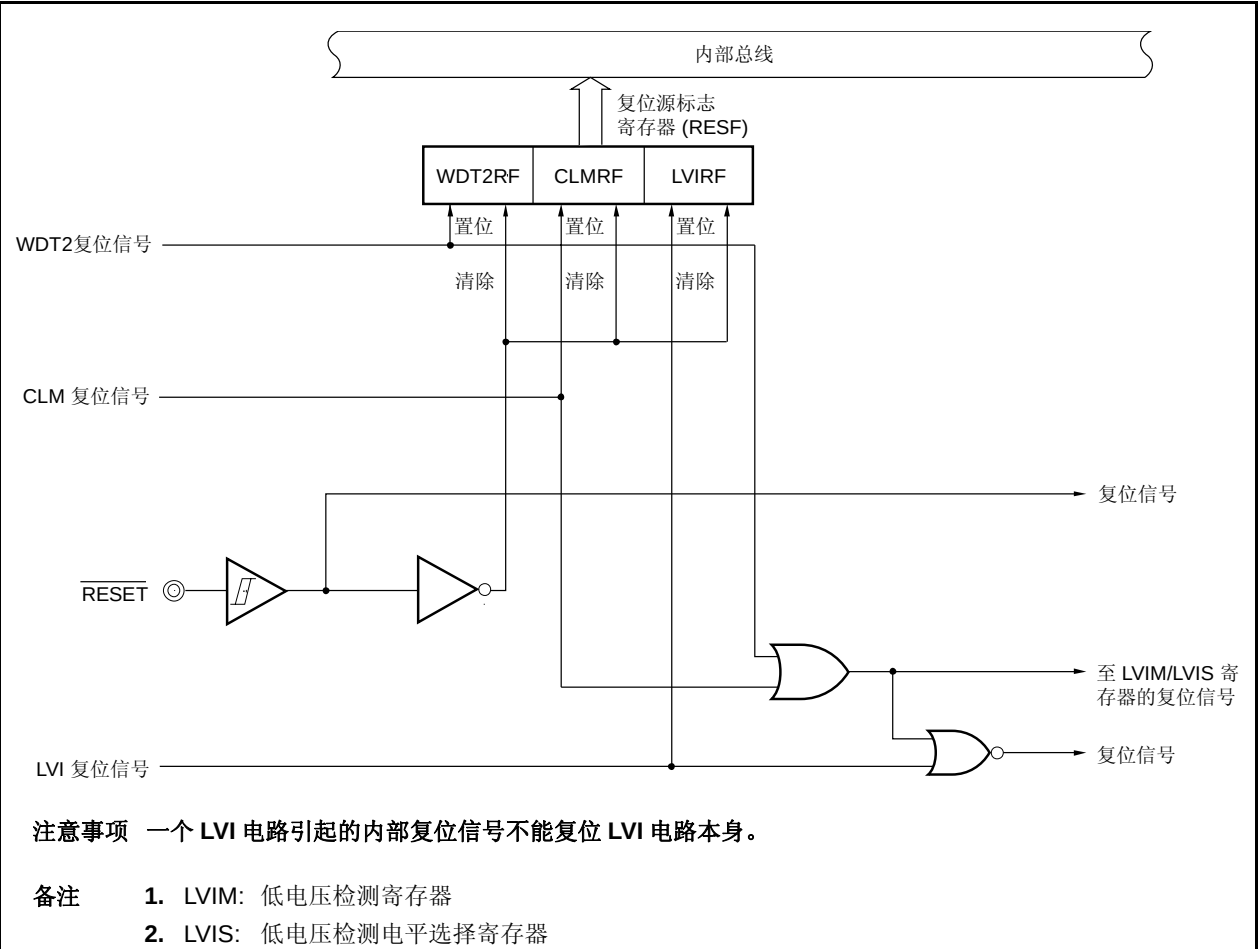
复位解除后，可以通过复位源标志寄存器 (RESF) 确定复位源。

(2) 紧急操作模式

如果 WDT2 在复位后插入的主时钟振荡稳定期间内溢出，则主时钟振荡被判断为异常，且 CPU 开始运行于内部振荡时钟。

注意事项 紧急操作模式下，除了那些能够运行于内部振荡时钟的模块的寄存器，如用于中断、端口功能、WDT2 或定时器 M 等的寄存器，不要访问其它片上外设 I/O 寄存器。此外，在该模式下禁止使用外部时钟 CSIB0 至 CSIB5 和 UARTA0 进行操作。

图 22-1. 复位功能框图



22.2 检查复位源的寄存器

V850ES/JJ3 单片机有 4 种类型的复位源。复位解除后，可以由复位源标志寄存器（RESF）确认已经发生的复位源。

(1) 复位源标志寄存器（RESF）

RESF 寄存器是一个专用寄存器，它只能通过一个特定序列的组合来写入（参见 3.4.7 专用寄存器）

RESF 寄存器表明复位信号由哪个源产生。

该寄存器可按字节或按位进行读取或写入。

RESET 引脚输入将该寄存器清为 00H。如果复位源不是 RESET 引脚的复位信号，其默认值将不同。

复位后：00H^注 R/W 地址：FFFFF888H

	7	6	5	4	3	2	1	0
RESF	0	0	0	WDT2RF	0	0	CLMRF	LVIRF

WDT2RF	来自WDT2的复位信号
0	未产生
1	已产生

CLMRF	来自CLM的复位信号
0	未产生
1	已产生

LVIRF	来自LVI的复位信号
0	未产生
1	已产生

注 当通过 RESET 引脚执行一个复位时，RESF 寄存器的值被清为 00H。当通过看门狗定时器 2（WDT2）、低电压检测器（LVI）或时钟监视器（CLM）执行一个复位时，该寄存器的复位标志（WDT2RF 位、CLMRF 位和 LVIRF 位）被置位。而其它复位源时保持不变。

注意事项 寄存器的每一位只能被写入“0”。如果写入“0”和置位标志（发生复位时）发生冲突，则置位标志优先。

22.3 操作

22.3.1 通过 $\overline{\text{RESET}}$ 引脚的复位操作

当 $\overline{\text{RESET}}$ 引脚输入低电平时，系统复位，各个硬件单元被初始化。

当 $\overline{\text{RESET}}$ 引脚的电平由低变为高时，复位状态被解除。

表 22-1. $\overline{\text{RESET}}$ 引脚输入的硬件状态

项目	复位期间	复位后
主时钟振荡器 (fx)	振荡停止	振荡开始
副时钟振荡器 (fxT)	振荡继续	
内部 振荡器	振荡停止	振荡开始
外部时钟 (fx 至 fx/1,024)	操作停止	在确保振荡稳定时间之后开始操作
内部系统时钟 (fCLK) , CPU 时钟 (fCPU)	操作停止	在确保振荡稳定时间之后开始操作 (初始化至 fxx/8)
CPU	初始化	在确保振荡稳定时间之后程序开始执行
看门狗定时器 2	操作停止 (初始化为 0)	用内部振荡时钟作源时钟, 从 0 开始往上计数
内部 RAM	在加电复位时或 CPU 存取和复位输入冲突 (数据被损坏) 时, 状态不确定; 否则, 保留复位后的瞬时值。	
I/O 线 (端口/复用功能引脚)	高阻 ^②	
片上外设 I/O 寄存器	初始化到指定状态, OCDM 寄存器设为 (01H)。	
其它片上外设功能	操作停止	在确保振荡稳定时间之后开始操作

注 当电源开启时, 即使在复位期间, 下列引脚也可能会暂时输出一个不确定的电平。

- P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 引脚

注意事项 OCDM 寄存器通过 $\overline{\text{RESET}}$ 引脚的输入来初始化。因此, 必须注意的是: 在复位解除之后 OCDM.OCDM0 位被清除之前, 如果有高电平输入 P05/ $\overline{\text{DRST}}$ 引脚, 会进入片上调试模式。详情参见第四章 端口功能。

图 22-2. RESET 引脚输入引起的复位操作时序

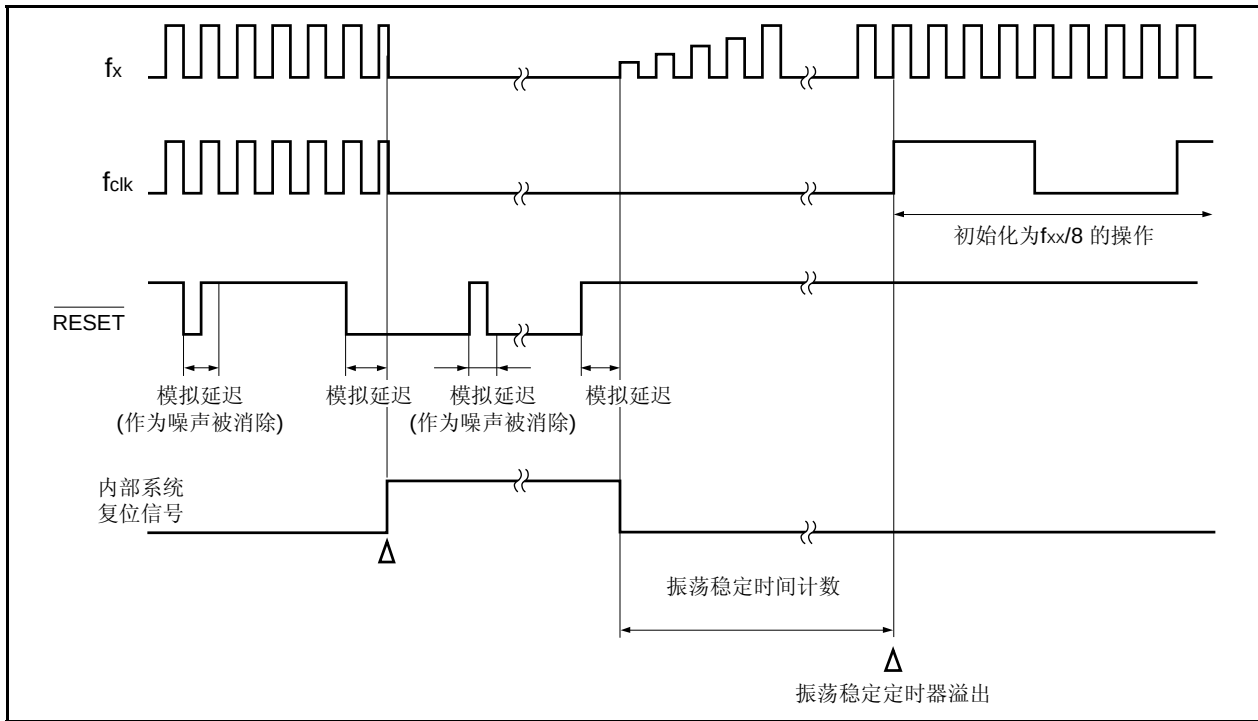
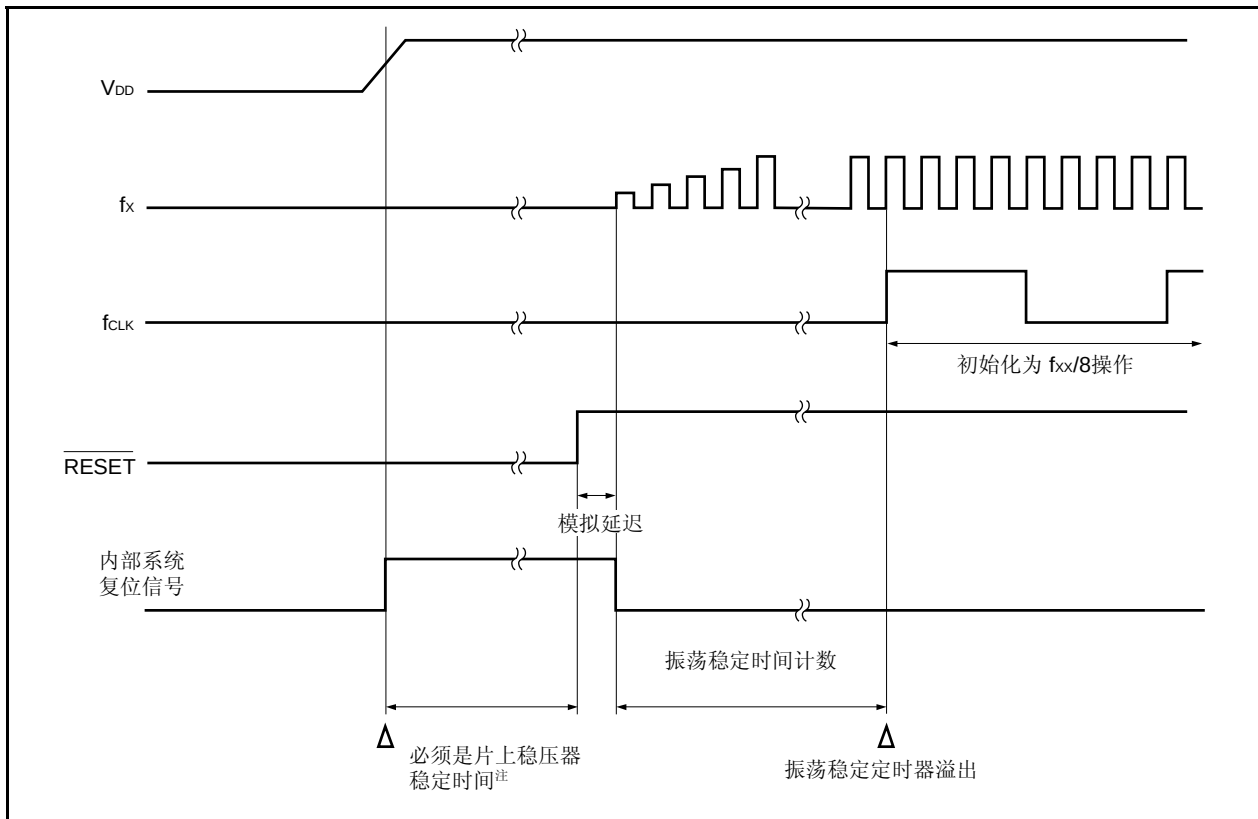


图 22-3. 上电复位操作的时序



22.3.2 通过看门狗定时器 2 的复位操作

若看门狗定时器 2 被设置为溢出复位工作模式，当看门狗定时器 2 发生溢出时（产生 WDT2RES 信号），则执行系统复位，并且硬件被初始化为初始状态。

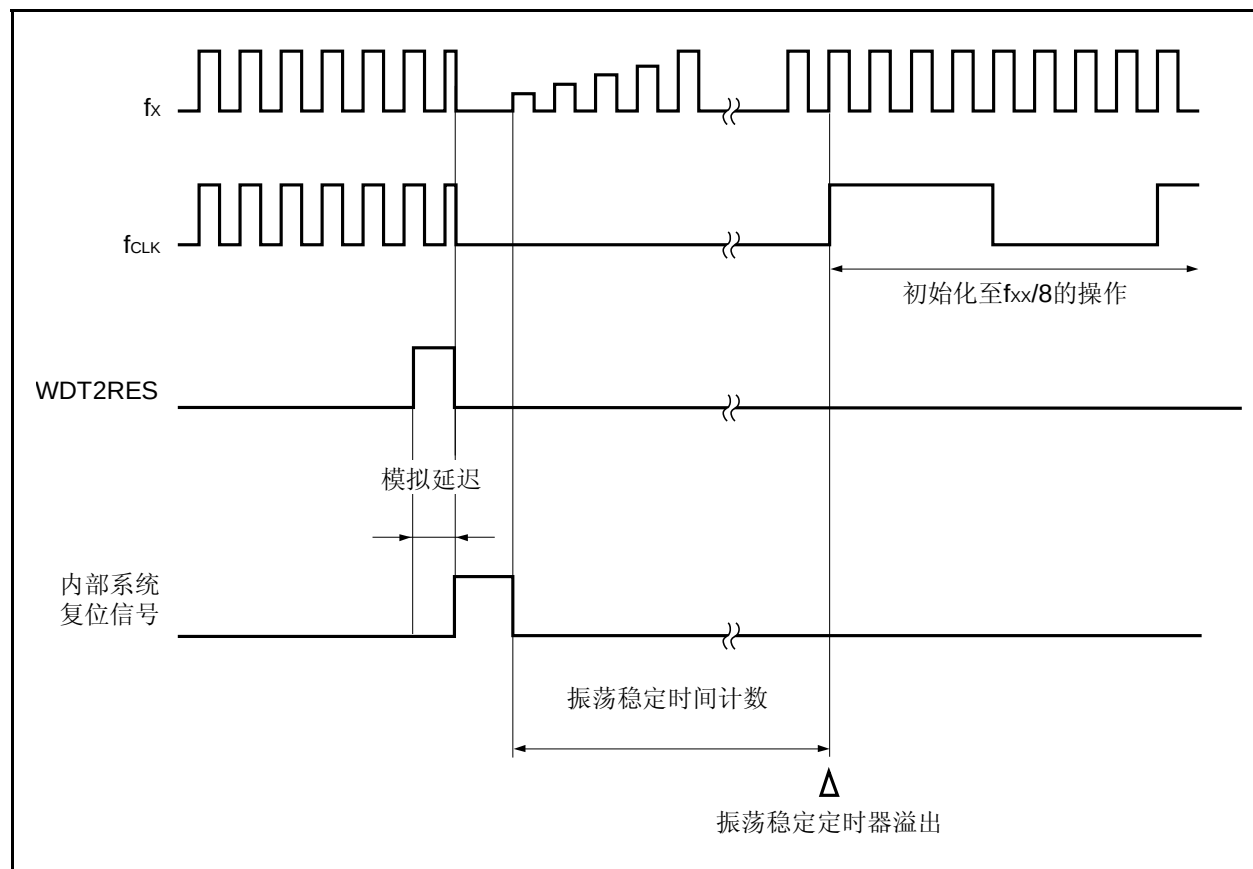
在看门狗定时器 2 溢出后，进入复位状态并持续预定时间（模拟延迟），并且，复位状态随后被自动解除。

在复位期间主时钟振荡器停止。

表 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 (f_x)	振荡停止	振荡开始
副时钟振荡器 (f_{xt})	振荡继续	
内部振荡器	振荡停止	振荡开始
外部时钟 (f_{xx} 至 $f_{xx}/1,024$)	操作停止	在确保振荡稳定时间之后开始操作
内部系统时钟 (f_{xx})， CPU 时钟 (f_{CPU})	操作停止	在确保振荡稳定时间之后开始操作（初始化为 $f_{xx}/8$ ）
CPU	初始化	在确保振荡稳定时间之后程序开始执行
看门狗定时器 2	操作停止（初始化为 0）	内部振荡时钟作为源时钟，从 0 开始往上计数。
内部 RAM	在加电复位时或当 CPU 存取和复位输入冲突（数据被损坏）时，状态不确定，否则，保留复位后的瞬时值。	
I/O 线（端口/复用功能引脚）	高阻	
片上外设 I/O 寄存器	初始化为特定状态，保留 OCDM 寄存器的值。	
其它片上外设功能	操作停止	在确保振荡稳定时间之后开始操作。

图 22-4. 通过 WDT2RES 信号产生的复位操作时序



22.3.3 通过低电压检测器的复位操作

当 LVI 操作使能时，如果供电电压下降到低电压检测器的检测电压以下，执行一次系统复位（当 LVIM.LVIMD 位被设为 1 时），且硬件被初始化到初始状态。

从检测到供电电压下降一直持续到供电电压上升超过 LVI 检测电压，一直都保持在复位状态。

在复位期间主时钟振荡器停止。

当 LVIMD 位= 0 时，如果检测到一个低电压则产生一个中断请求信号（INTLVI）。

表 22-3. 由低电压检测器复位操作时的硬件状态

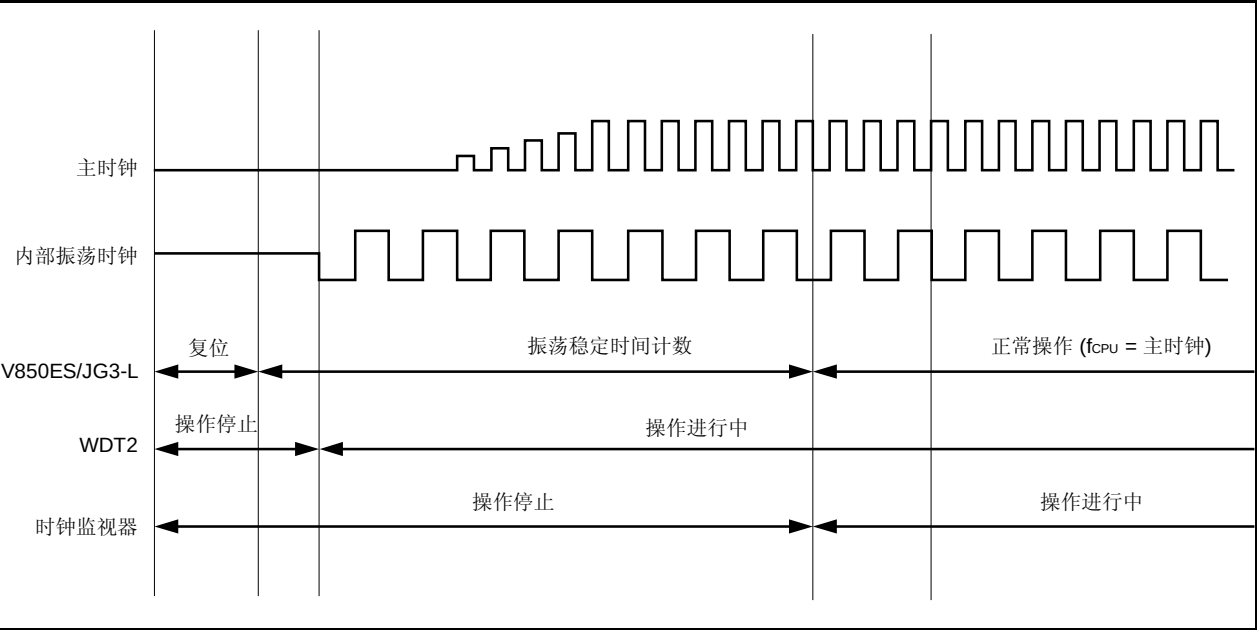
项目	复位期间	复位后
主时钟振荡器 (fx)	振荡停止	振荡开始
副时钟振荡器 (fxT)	振荡继续	
内部振荡器 r	振荡停止	振荡开始
外部时钟 (fx 至 fx/1,024)	操作停止	在确保振荡稳定时间之后开始操作
内部系统 时钟 (fxx) , CPU 时钟 (fCPU)	操作停止	在确保振荡稳定时间之后开始操作（初始化为 fxx/8）
CPU	初始化	在确保振荡稳定时间之后程序开始执行
看门狗定时器 2	操作停止（初始化为 0）	内部振荡时钟作为源时钟，从 0 开始往上计数。
内部 RAM	在加电复位时或当 CPU 访问和复位输入冲突（数据被损坏）时，则会状态不确定。否则，保留复位后的瞬时值	
I/O 线（端口/复用功能引脚）	高阻	
片上外设 I/O 寄存器	初始化到指定状态，保留 OCDM 寄存器的值。	
LVI	操作继续	
其它片上外围功能	操作停止	在确保振荡稳定时间之后开始操作。

备注 关于低电压检测器的复位时序，参见 第二十四章 低电压检测器（LVI）。

22.3.4 复位解除后的操作

复位解除后，主时钟开始振荡且确保经过振荡稳定时间（OSTS 寄存器初始值： $2^{16}/f_x$ ）之后，CPU 开始执行程序。使用内部振荡时钟作为源时钟时，在复位解除之后 WDT2 立刻开始运行。

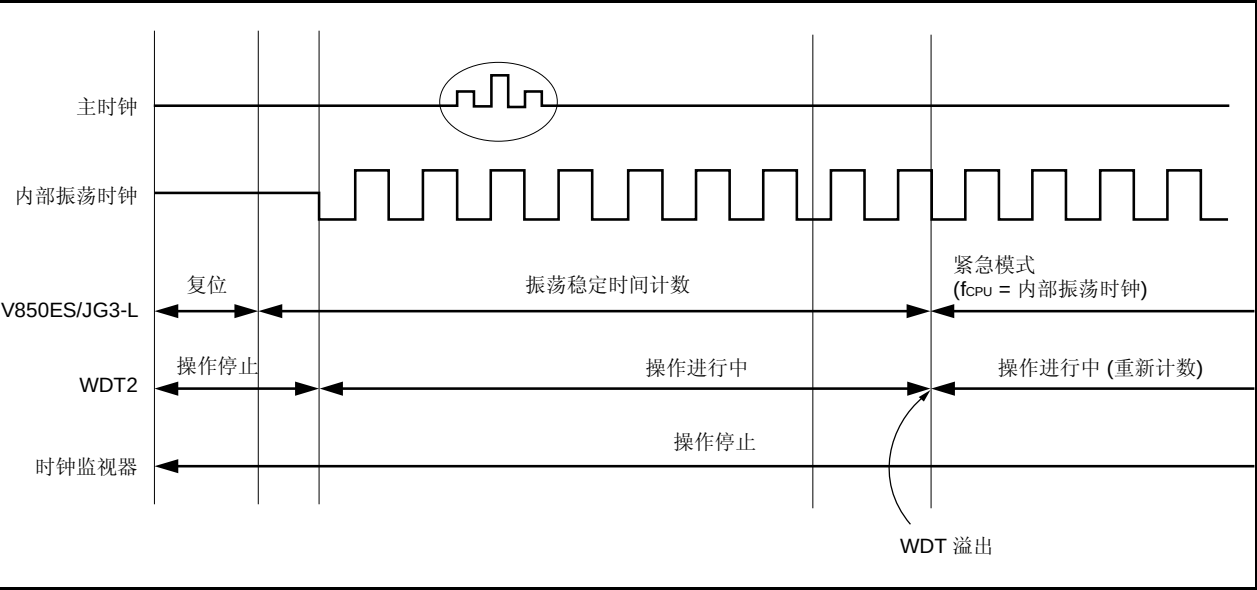
图 22-5. 复位解除后的操作



(1) 紧急操作模式

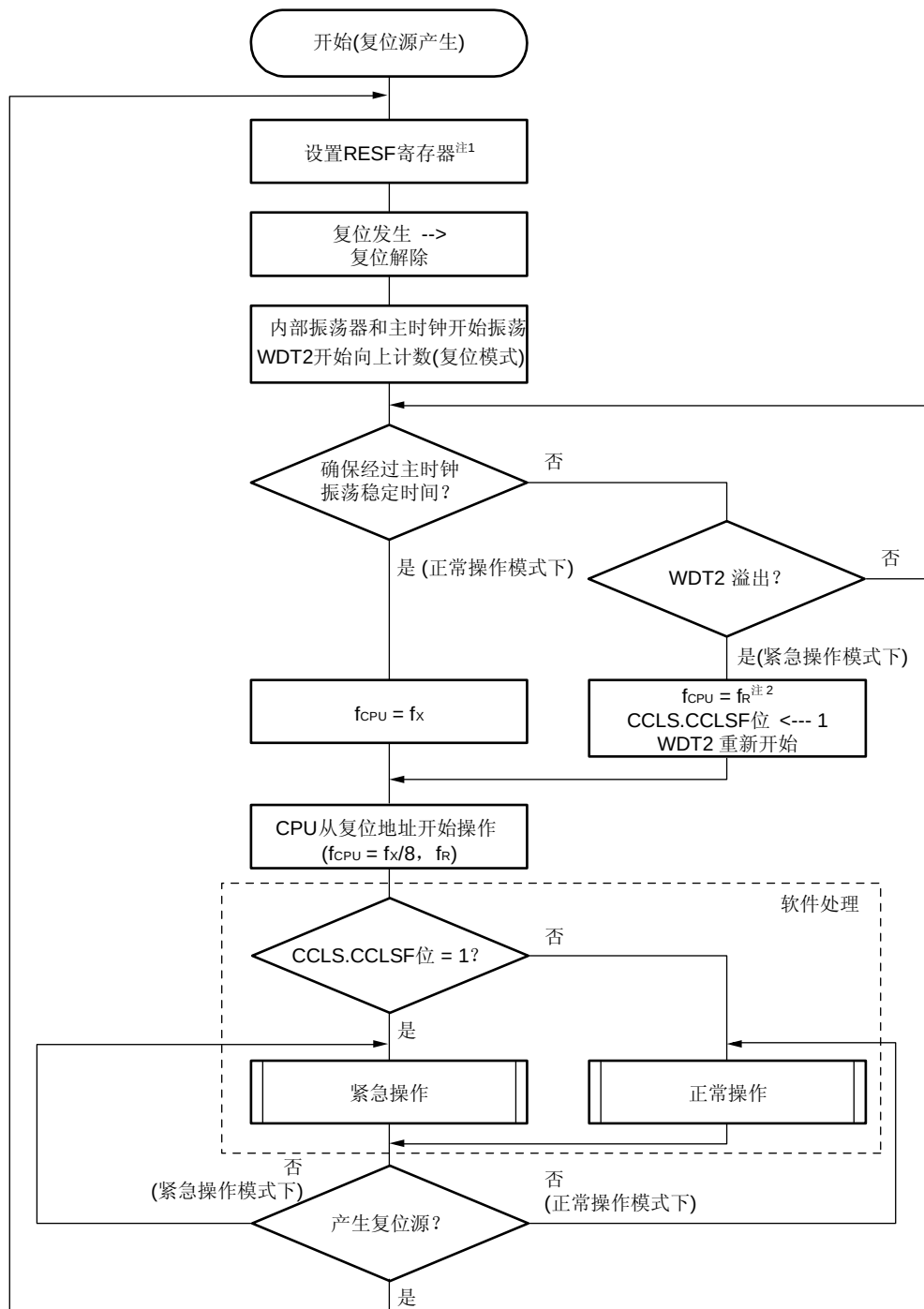
如果在振荡稳定时间完成之前主时钟发生异常，则在执行 CPU 程序之前，WDT2 就已经溢出。此时，将内部振荡时钟作为源时钟，CPU 开始执行程序。

图 22-6. 复位解除后的操作



可以使用 CPU 操作时钟状态寄存器（CCLS）来检测 CPU 操作时钟的状态。

22.3.5 复位功能操作流程



注 1. 根据复位源对不同的位置 1。

复位源	WDT2RF 位	CRMRF 位	LVIRF 位
RESET 引脚	0	0	0
WDT2	1	保留复位前的值。	保留复位前的值。
CLM	保留复位前的值。	1	保留复位前的值。
LVI	保留复位前的值。	保留复位前的值。	1

2. 内部振荡器不能停止。

23.1 功能

时钟监控器使用内部振荡时钟，对主时钟进行采样，当主时钟振荡停止时，产生复位请求信号。

一旦通过操作使能标志允许时钟监控器的操作，则除了复位之外，没有办法可以使之清 0。

当时钟监控器引发复位时，RESF.CLMRF 位置位。关于 RESF 寄存器的细节，参见 22.2 检查复位源寄存器。

在下列条件时，时钟监控器自动停止：

- 解除 STOP 模式后的振荡稳定期间
- 主时钟停止时（从副时钟工作期内 PCC.MCK 位 = 1 时开始，直至主时钟工作期内 PCC.CLS 位 = 0）
- 采样时钟（内部振荡时钟）停止时
- 当 CPU 运行于内部振荡时钟工作时

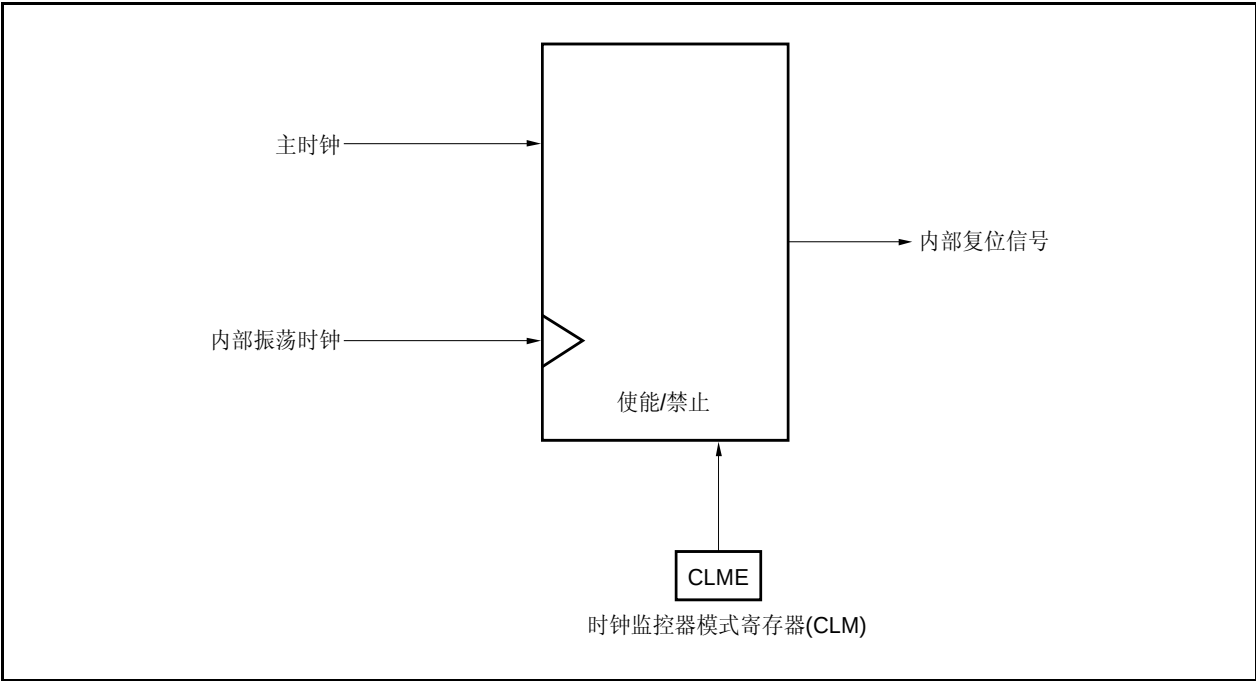
23.2 配置

时钟监控器包含以下硬件。

表 23-1. 时钟监控器的配置

项目	配置
控制寄存器	时钟监控模式寄存器（CLM）

图 23-1. 通过 RESET 引脚输入进行复位操作的时序



23.3 寄存器

时钟监控器由时钟监控器模式寄存器（CLM）控制。

(1) 时钟监控器模式寄存器（CLM）

CLM 寄存器是专用寄存器。只有通过特定序列的组合才能写入该寄存器（见 3.4.7 专用寄存器）。
该寄存器用于设置时钟监控器的操作模式。
该寄存器可以按字节或按位进行读写。
系统复位后，设置该寄存器为 00H。

复位后：00H R/W 地址：FFFFF870H							
	7	6	5	4	3	2	1 <0>
CLM	0	0	0	0	0	0	CLME
CLME		时钟监控器操作使能或禁止					
0		禁止时钟监控器操作。					
1		允许时钟监控器操作。					

- 注意事项
- 一旦 CLME 位设置为 1，除非复位，不能以任何其它方式将其清 0。
 - 当时钟监控器引发复位时，CLME 位被清为 0，而 RESF.CLMRF 位被置为 1。

23.4 操作

本节介绍时钟监控器的功能。启动条件和停止条件如下：

<启动条件>

将 CLM.CLME 位设为 1，允许操作。

<停止条件>

- 解除 STOP 模式后对振荡稳定时间进行计数时
- 当主时钟停止时（从副时钟工作期内 PCC.MCK 位 = 1 时开始，直到主时钟工作期内 PCC.CLS 位 = 0）
- 当 采样时钟（内部振荡时钟）停止时
- 当 CPU 运行于内部振荡器时钟时

表 23-2. 时钟监控器操作状态
(内部振荡器时钟工作期间，CLM.CLME 位 = 1 时)

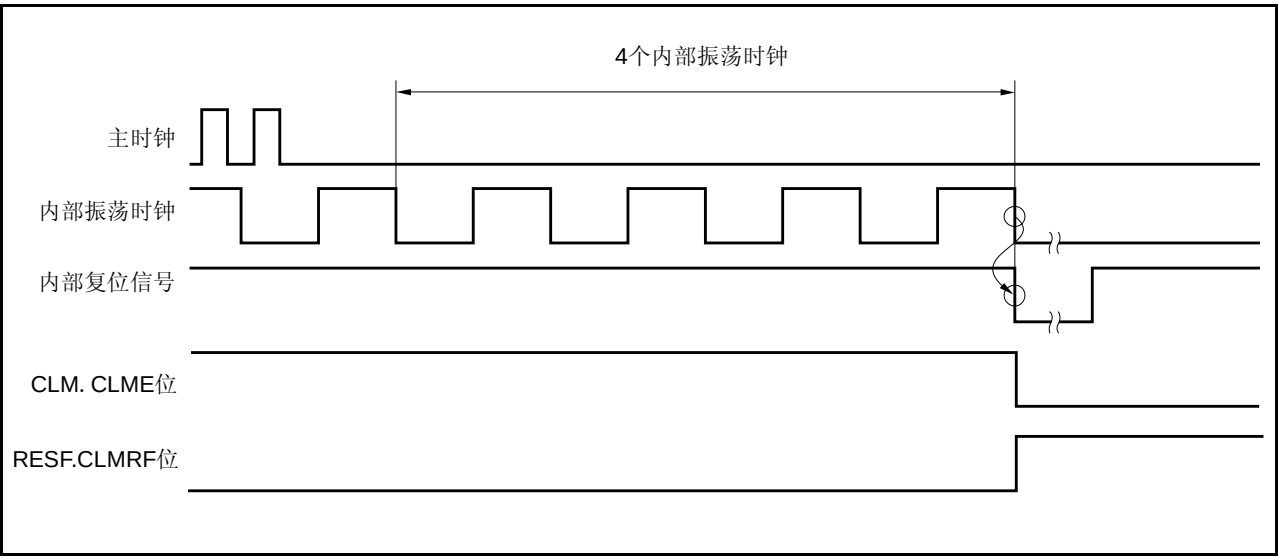
CPU 工作时钟	操作模式	主时钟状态	内部振荡时钟状态	时钟监控器状态
主时钟	HALT 模式	振荡	振荡 ^{注 1}	工作 ^{注 2}
	IDLE1, IDLE2 模式	振荡	振荡 ^{注 1}	工作 ^{注 2}
	STOP 模式	停止	振荡 ^{注 1}	停止
副时钟（PCC 寄存器的 MCK 位 = 0）	Sub-IDLE 模式	振荡	振荡 ^{注 1}	工作 ^{注 2}
副时钟（PCC 寄存器 MCK 位=1）	Sub-IDLE 模式	停止	振荡 ^{注 1}	停止
内部振荡时钟	—	停止	振荡 ^{注 3}	停止
复位期间	—	停止	停止	停止

- 注
1. 通过设置 RCM.RSTOP 位为 1，可以停止内部振荡器。
 2. 当内部振荡器停止时，时钟监控器也同样停止。
 3. 内部振荡器不能通过软件停止。

(1) 主时钟振荡停止时的操作 (CLME 位=1)

如果在 CLME 位 = 1 时主时钟振荡停止，则产生一个内部复位信号，如图 23-2 所示。

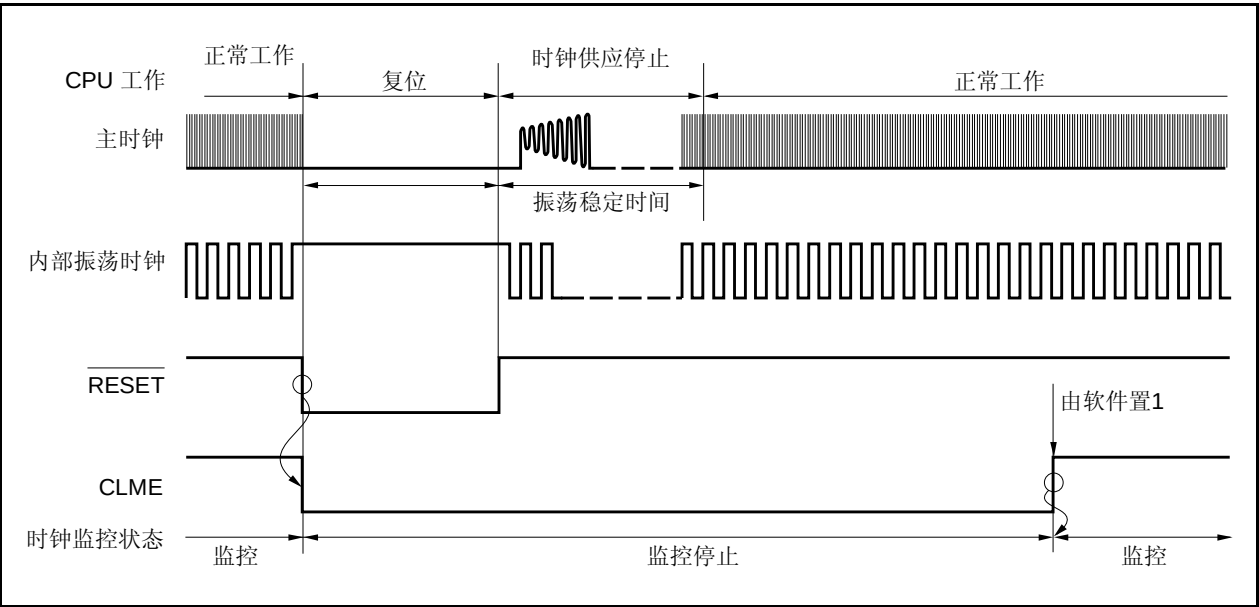
图 23-2. 因主时钟振荡器停止造成的复位时间



(2) RESET 输入后时钟监控器的状态

RESET 输入将 CLM.CLME 位清为 0，并停止时钟监控器的操作。在主时钟振荡稳定时间结束后，由软件将 CLME 位设为 1，则开始监控。

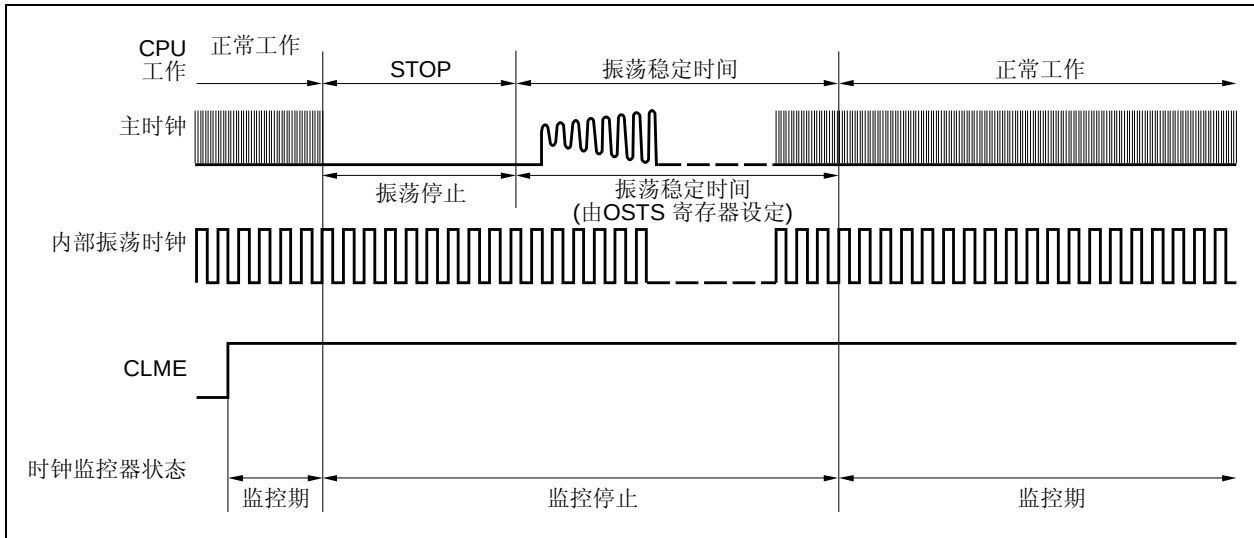
图 23-3. RESET 输入后的时钟监控器状态
(RESET 输入且在主时钟振荡稳定时间结束后设置 CLM.CLME 位 = 1)



(3) 在 STOP 模式下或 STOP 模式解除后的操作

如果在 CLM.CLME 位 = 1 时设置 STOP 模式，那么监控器会在 STOP 模式下停止工作，在振荡稳定时间计数期间也停止工作。振荡稳定时间过后，监控器将会自动开始工作。

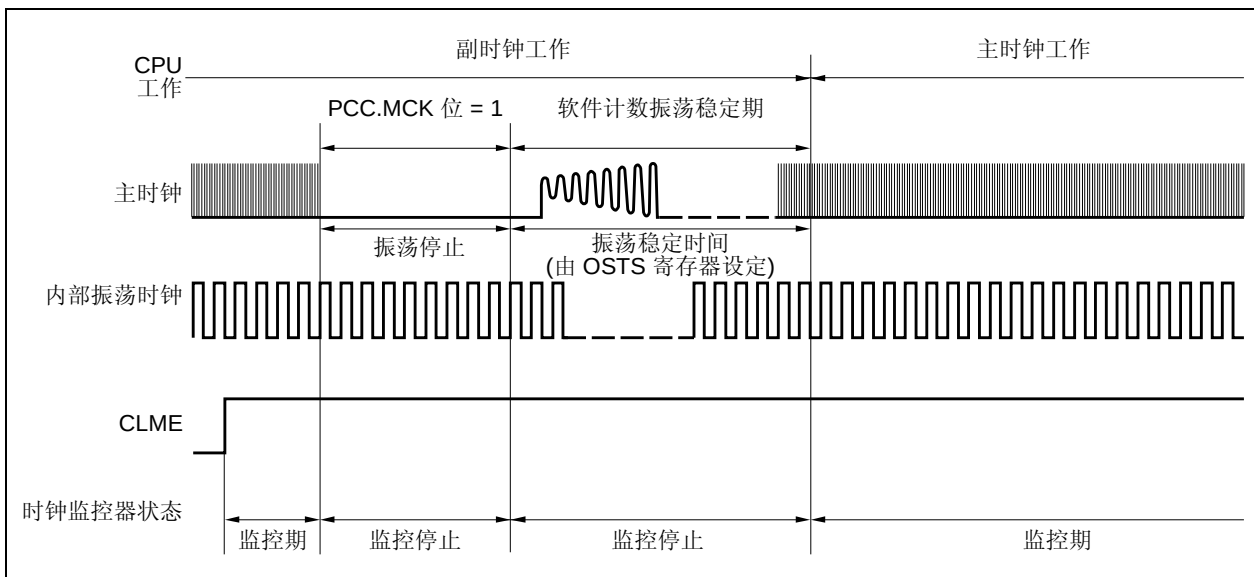
图 23-4. 在 STOP 模式下或 STOP 模式解除后的操作



(4) 主时钟停止（任意）时的操作

在副时钟操作（PCC.CLS 位 = 1）期间，或通过设置 PCC.MCK 为 1 使主时钟停止工作时，监控器操作停止，直到主时钟开始工作（PCC.CLS 位 = 0）。当主时钟开始工作时，监控器自动开始工作。

图 23-5. 主时钟停止（任意）时的操作



(5) 当 CPU 运行于内部振荡时钟（CCLS.CCLSIF 位= 1）时的操作

当 CCLSIF 位为 1 时，即使 CLME 位被置为 1，监控器也不会停止工作。

第二十四章 低电压检测器 (LVI)

24.1 功能

低电压检测器 (LVI) 具有以下功能。

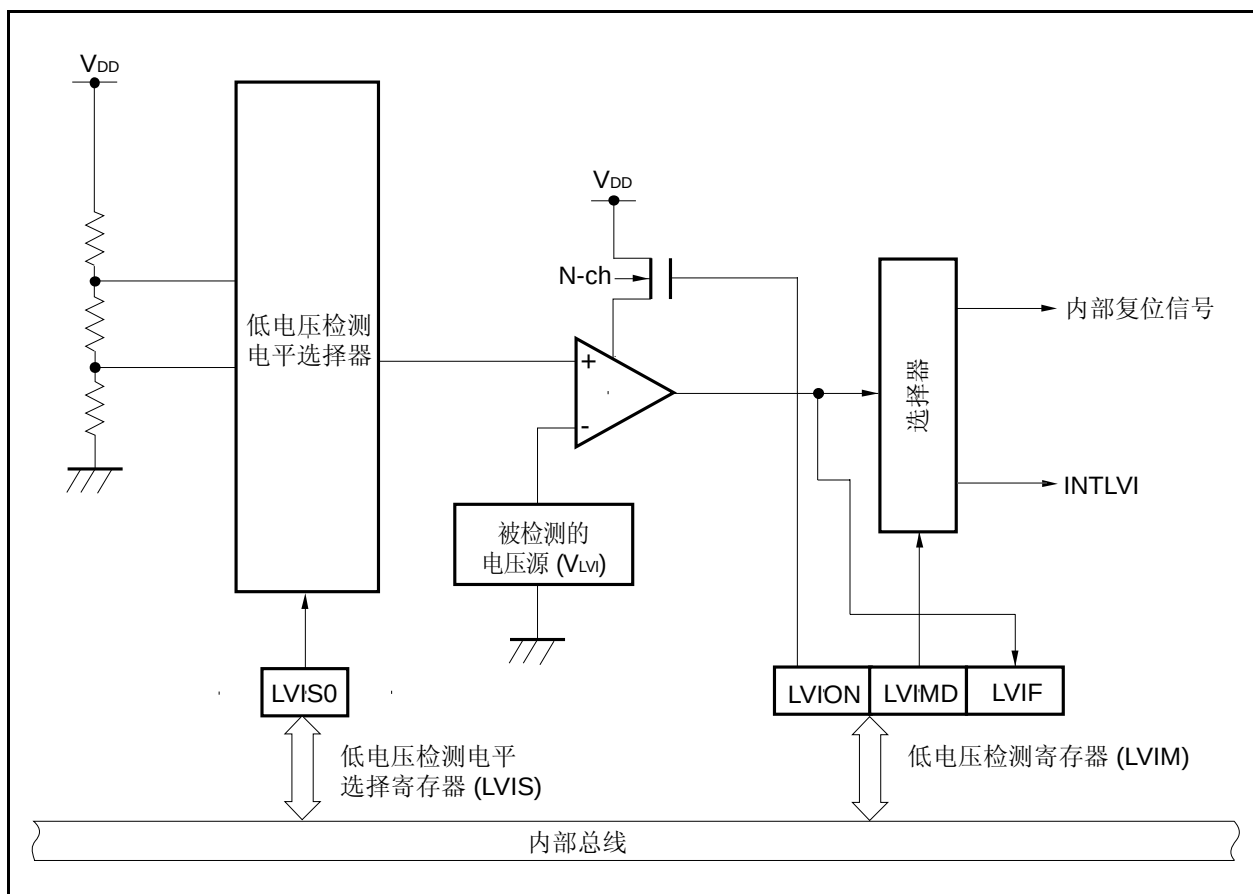
- 如果在使用低电压检测后发生中断，低电压检测器比较供电电压 (V_{DD}) 和检测电压 (V_{LVI})，并且，在供电电压下降或上升经过检测电压时，产生一个内部中断信号。
- 如果在使用低电压检测后发生复位，当供电电压 (V_{DD}) 下降经过检测电压 (V_{LVI}) 时，低电压检测器产生一个内部复位信号。
- 可由软件选择为中断或复位信号。
- 可以在 STOP 模式下工作。

低电压检测器用来产生复位信号，当复位信号产生时 RESF.LVIRF 位被设置为 1。关于 RESF 寄存器的详情，参见 22.2 复位源检查寄存器。

24.2 配置

低电压检测器的框图如下所示。

图 24-1. 低电压寄存器框图



24.3 寄存器

低电压检测器由下列寄存器控制。

- 低电压检测寄存器 (LVIM)
- 低电压检测电平选择寄存器 (LVIS)
- 内部 RAM 数据状态寄存器 (RAMS)

(1) 低电压检测寄存器 (LVIM)

LVIM 是一个专用寄存器。只有通过特定序列的组合才能写入该寄存器 (参见 3.4.7 专用寄存器)。
LVIM 寄存器用于使能或禁止低电压检测, 并且设置低电压检测器的工作模式。
该寄存器可以按字节或按位进行读取或写入, 但是, LVIF 位为只读位。

复位后: 注 1		R/W		地址: FFFFF890H				
<7>		6	5	4	3	2	<1>	<0>
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF
LVION		使能或禁止低电压检测操作						
0		禁止工作。						
1		使能工作。						
LVIMD		选择低电压检测的工作模式						
0		当供电电压下降或上升经过被检测电压时, 产生中断信号 INTLVI。						
1		当 供电电压 < 被检测电压时, 产生内部复位信号 LVIRES。						
LVIF ^{注 2}		低电压检测标志						
0		当供电电压 > 检测电压, 或操作禁止时。						
1		当前接入电源的供电电压 < 检测电压时。						

- 注 1. 由低电压检测复位: 82H
由其它复位源复位: 00H
2. 在 LVI 开始工作 (LVION 位 = 1) 或产生 INTLVI 之后, 使用 LVIF 位来确认供电电压的状态。

- 注意事项 1. 当 LVION 和 LVIMD 位为 1 时, 低电压检测器不能停止, 除非低电压检测之外的其他复位源产生复位信号。
2. 当 LVION 位设置为 1 时, LVI 电路内的比较器开始工作。在 LVION 位置位后, 使用 LVIF 位检测电压前, 应该由软件控制等待 0.2 ms 或更长的时间。
3. 请确保将第 6 位至第 2 位清除为 “0”。

(2) 低电压检测电平选择寄存器 (LVIS)

LVIS 寄存器用于选择低电压检测的检测电平值。

该寄存器能够按字节或按位进行读取或写入。

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	检测电平
0	2.95 V（典型值）±0.10 V
1	保留（禁止设置）

注 由低电压检测复位: 保持
由其它复位源复位: 00H

注意事项 1. 在 LVIM.LVION 和 LVIM.LVIMD 位被设置为 1 之后, 低电压检测之外的其他复位源产生复位请求, 否则该寄存器不能写入。

2. 请确保将第 7 位至第 1 位清除为“0”。

(3) 内部 RAM 数据状态寄存器 (RAMS)

RAMS 寄存器是专用寄存器, 只能通过特定的序列组合来写入 (参见 3.4.7 专用寄存器)。

该寄存器是标志寄存器, 表示内部 RAM 是否有效。

该寄存器能够按字节或按位进行读取或写入。

对 RAMF 位的置位/清除条件如下。

- 置位条件: 对低于指令指定电平的电压的检测
- 清除条件: 按照专用序列写入 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 电压检测						
0		未检测到电压低于 RAM 保持电压						
1		检测到电压低于 RAM 保持电压						

注 仅当检测到电压跌落低于 RAM 保持电压时, 该寄存器被复位。

24.4 操作

由 LVIM.VIMD 位的设置来决定，产生中断信号 (INTLVI) 或内部复位信号。
如何指定每个操作的描述和时序图如下。

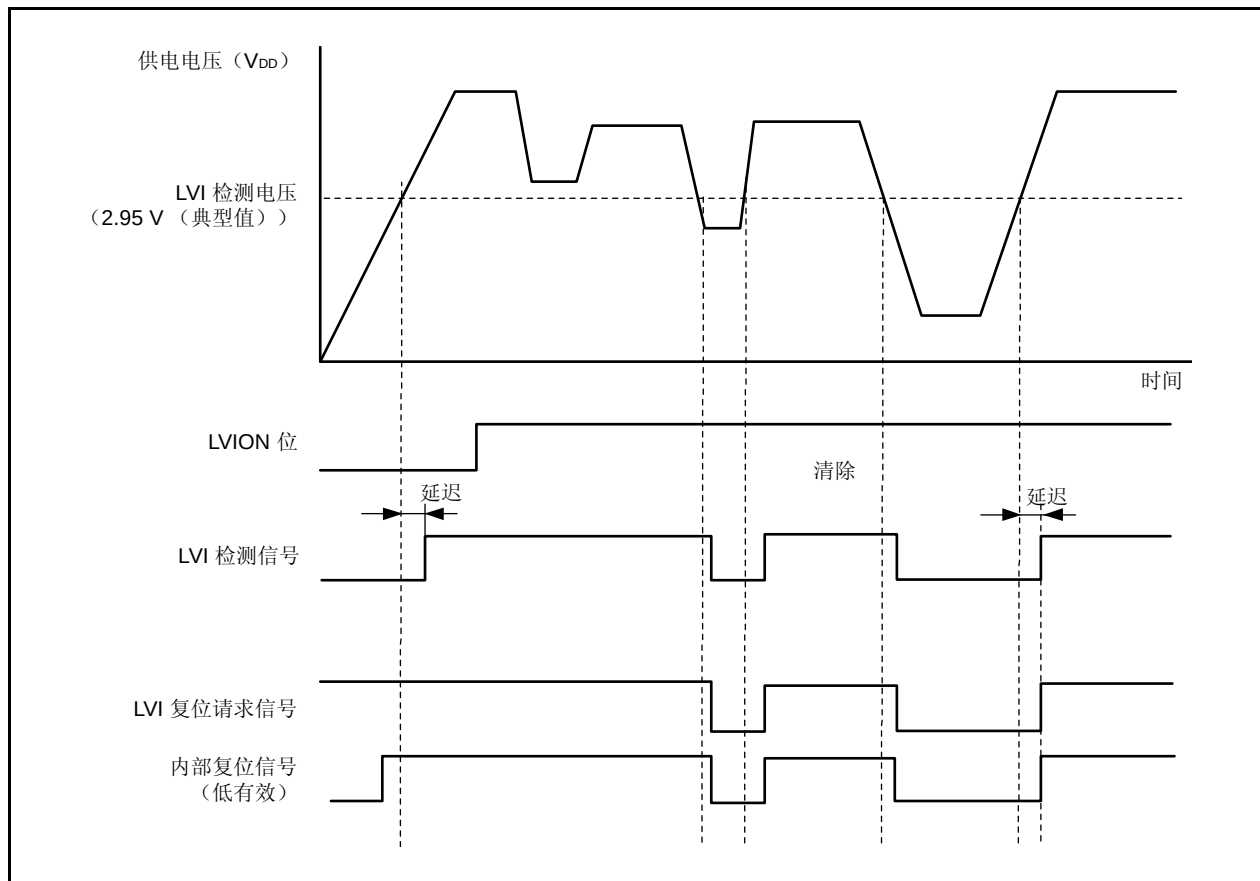
24.4.1 使用内部复位信号

<开始操作>

- <1> 屏蔽 LVI 中断。
- <2> 通过 LVIS.LVIS0 位选择检测电压。
- <3> 将 LVIM.LVION 位设为 1（使能操作）。
- <4> 通过软件插入一个 0.2 ms（最大值）或更长的等待周期。
- <5> 通过 LVIM.LVIF 位检测是否 供电电压 > 检测电压。
- <6> 将 LVIMD 位设为 1（产生一个内部复位信号）。

注意事项 如果 LVIMD 位被设为 1，LVIM 和 LVIS 寄存器的内容不能改变，除非产生一个 LVI 之外的复位请求。

图 24-2. 低电压检测器操作时序 (LVIMD 位= 1)



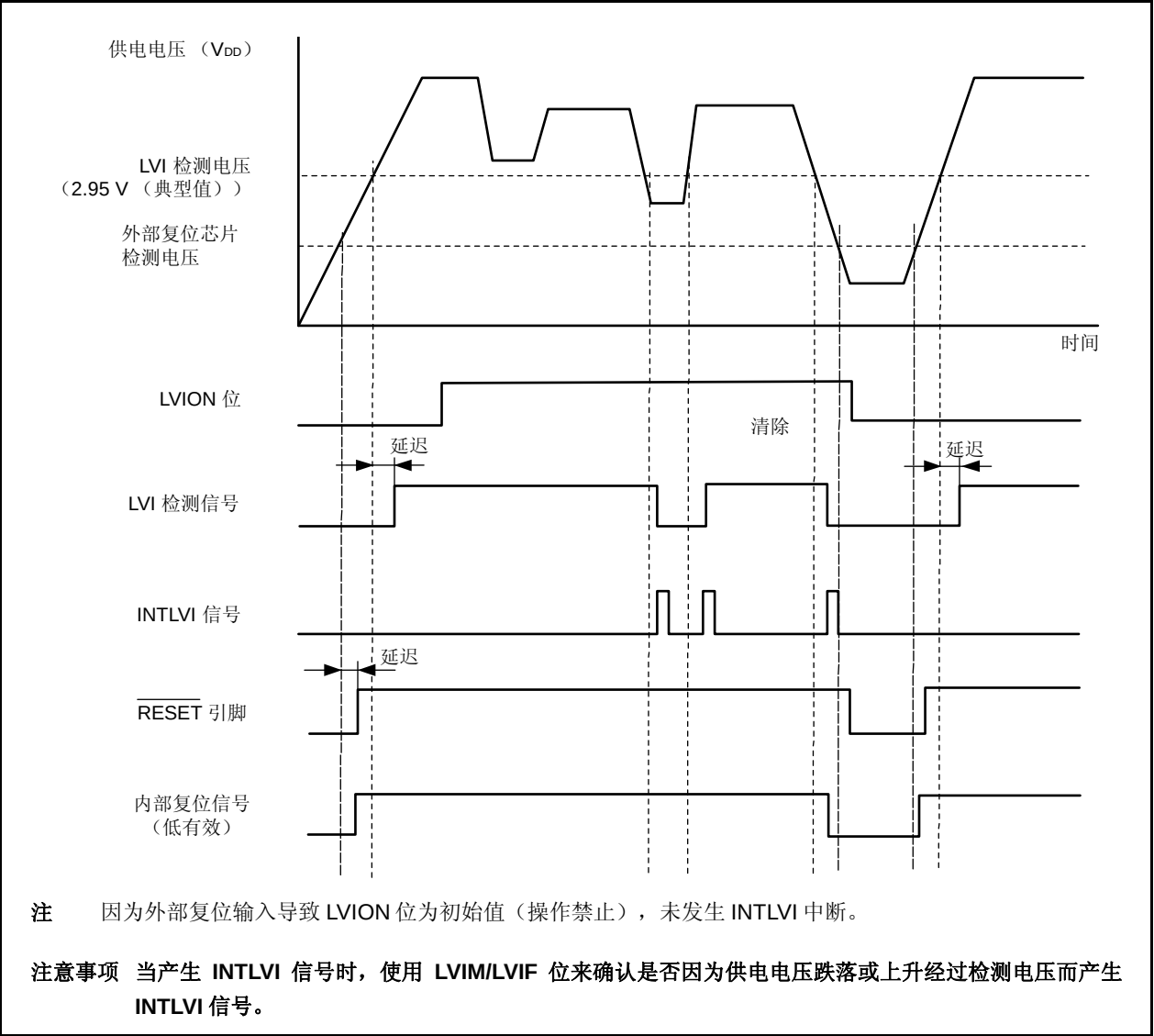
24.4.2 用作中断

- <开始操作>
- <1> 屏蔽 LVI 的中断。
 - <2> 通过 LVIS.LVIS0 位选择检测电压。
 - <3> 将 LVIM.LVION 位置为 1（使能操作）。
 - <4> 通过软件插入一个 0.2 ms（最大值）或更长的等待周期。
 - <5> 通过 LVIM.LVIF 位检测是否 供电电压 > 检测电压。
 - <6> 清除 LVI 的中断请求标志。
 - <7> 解除 LVI 的中断屏蔽。

<停止操作>

将 LVION 位清除为 0。

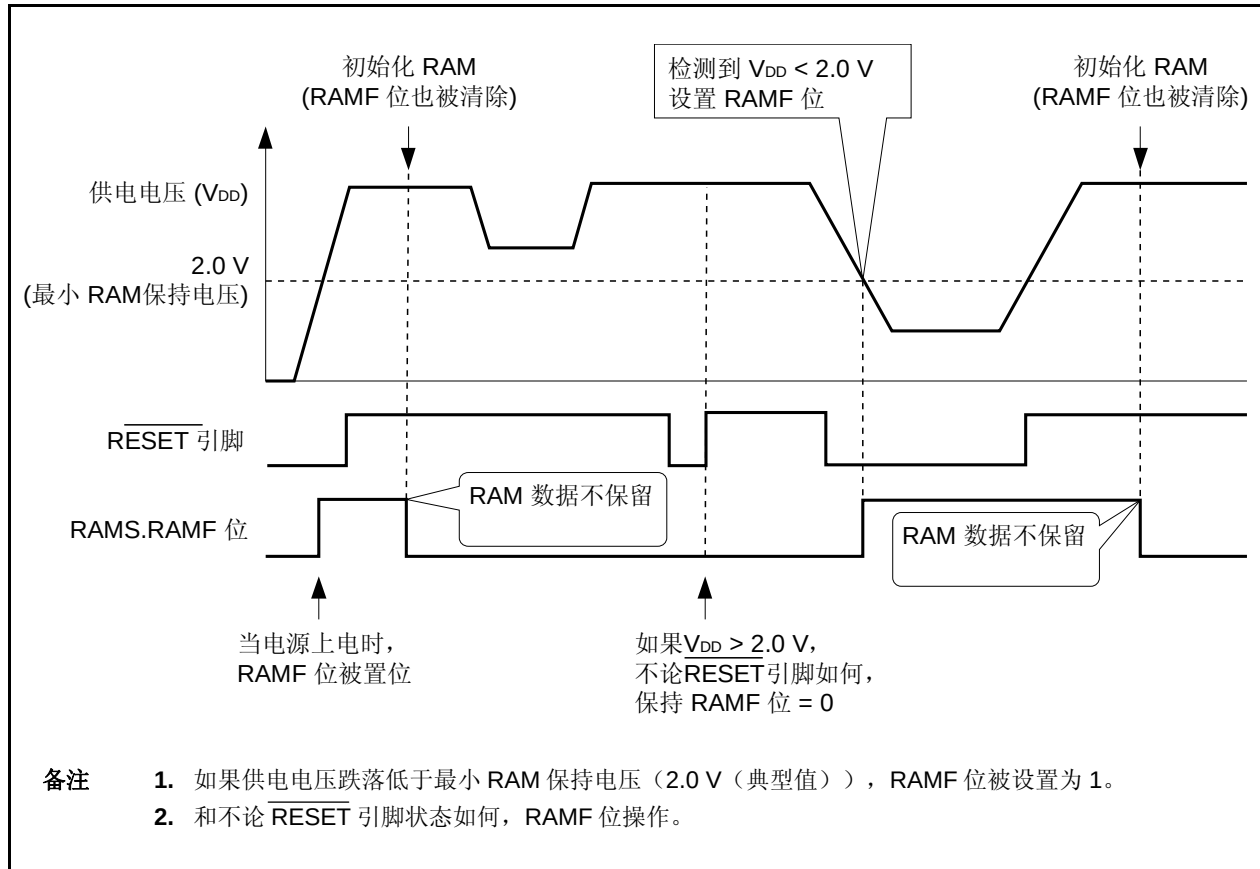
图 24-3. 低电压检测器的操作时序 (LVIMD 位= 0)



24.5 RAM 保持电压检测操作

供电电压和检测电压进行比较。当供电电压跌落低于检测电压时（包括供电电压），RAMS.RAMF 位被设置为 1。

图 24-4. RAM 保持电压检测功能的操作时序



24.6 仿真功能

当使用在线仿真器时，RAM 保持标志（RAMS.RAMF 位）的操作可以虚拟控制，且可以在调试器上通过操作 PEMU1 寄存器来仿真。

在仿真模式下，该寄存器有效。在正常模式下，该寄存器无效。

(1) 外围仿真寄存器 1 (PEMU1)

复位后: 00H R/W 地址: FFFFF9FEH

	7	6	5	4	3	2	1	0
PEMU1	0	0	0	0	0	EVARAMIN	0	0

EVARAMIN	RAM 保持电压检测信号的虚拟说明
0	对低于 RAM 保持电压的电压不检测
1	对低于 RAM 保持电压的电压进行检测（设置 RAMF 标志）

注意事项 该位不会被自动清除。

[用法]

当使用在线仿真器时，通过在调试器上写入该寄存器来实现 RAMF 的虚拟仿真。

- <1> CPU 断点（CPU 操作停止）
- <2> 通过使用一个寄存器写入命令将 EVARAMIN 位设置为 1。
通过将 EVARAMIN 位设置为 1，RAMF 位在硬件上被设置为 1（内部 RAM 数据无效）。
- <3> 通过使用一个寄存器写入命令将 EVARAMIN 位清除为 0。
除非执行该操作（将 EVARAMIN 位清除为 0），RAMF 位不能被 CPU 操作指令清除为 0。
- <4> 运行 CPU 且恢复仿真。

25.1 功能

- 使用 CRC 功能电路进行数据块的错误检测。
- 使用CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$)多项式以 8 位为单元对任何长度的数据块产生 16 位CRC校验码。
- 初始值设置到 CRCD 寄存器之后，每当 1 个字节数据被传送到 CRCIN 寄存器，CRC 码就被置入 CRC 数据寄存器。

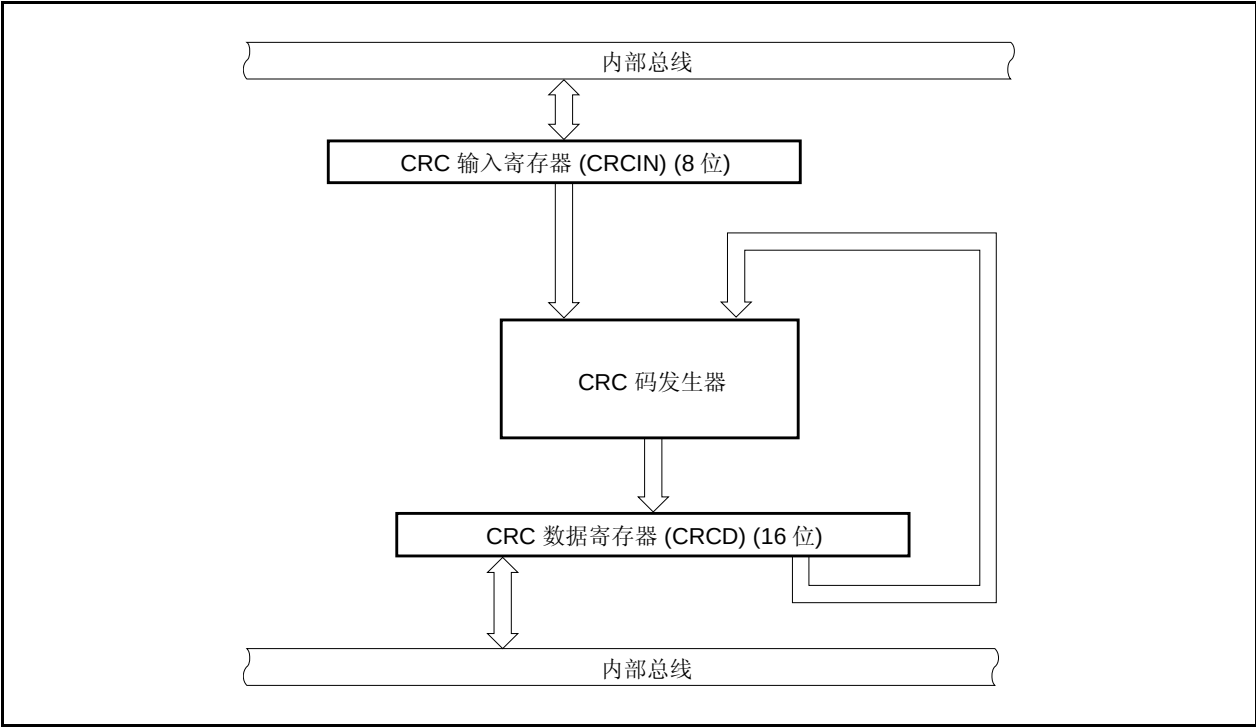
25.2 配置

CRC 控制功能包含下列硬件：

表 25-1. CRC 配置

项目	配置
控制寄存器	CRC 输入寄存器 (CRCIN)
	CRC 数据寄存器 (CRCD)

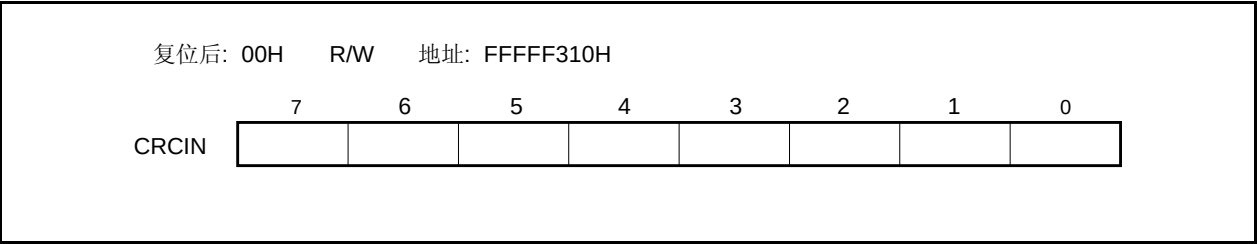
图 25-1. CRC 寄存器的功能框图



25.3 寄存器

(1) CRC 输入寄存器 (CRCIN)

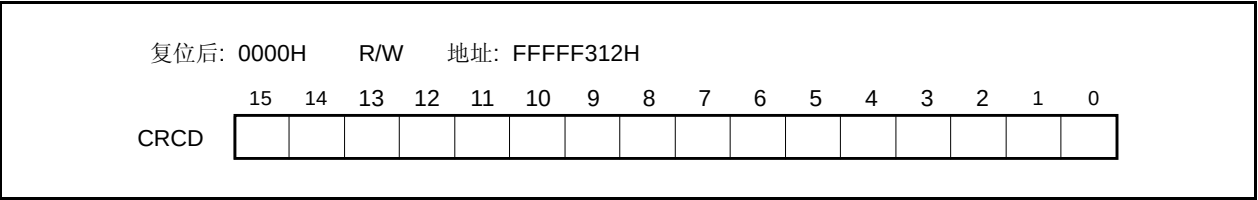
CRC 输入寄存器是 8 位寄存器，用于设置数据。
该寄存器可按字节进行读写。
系统复位后，该寄存器被设为 00H。



(2) CRC 数据寄存器 (CRCIN)

CRCD 寄存器为 16 位寄存器，用来存储 CRC-CCITT 运行结果。
该寄存器可按 16 位宽度进行读写。
系统复位后，该寄存器被清为 0000H。

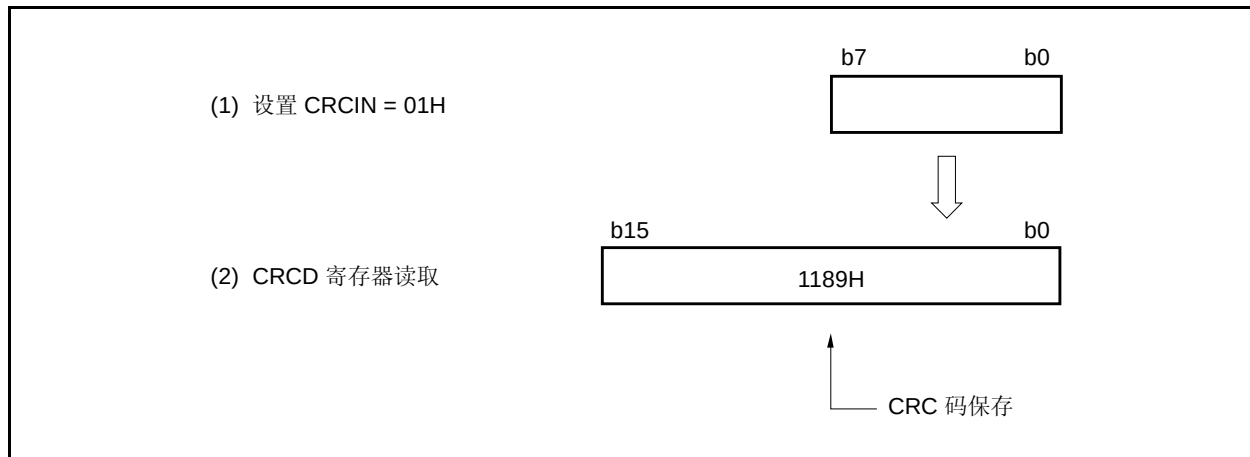
- 注意事项 在以下状态禁止访问 CRCD 寄存器。详情参见 3.4.9 (2) 访问片上专用外设 I/O 寄存器。
- 当 CPU 运行于副时钟时，而主时钟振荡停止。
 - 当 CPU 运行于内部振荡器时钟工作时。



25.4 操作

CRC 工作电路的举例如下。

图 25-2. CRC 工作电路操作示例 (LSB 先行)



当LSB先行时，01H码为(1000 0000)。所以，当使用modulo-2 操作公式，令 $(1000\ 0000) \times X^{16}$ 除以 $(1\ 0001\ 0000\ 0010\ 0001)$ 时，由生成多项式 $X^{16} + X^{12} + X^5 + 1$ 得到的CRC码为其余数。

modulo-2 操作基于以下公式进行。

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= 0 \\ -1 &= 1 \end{aligned}$$

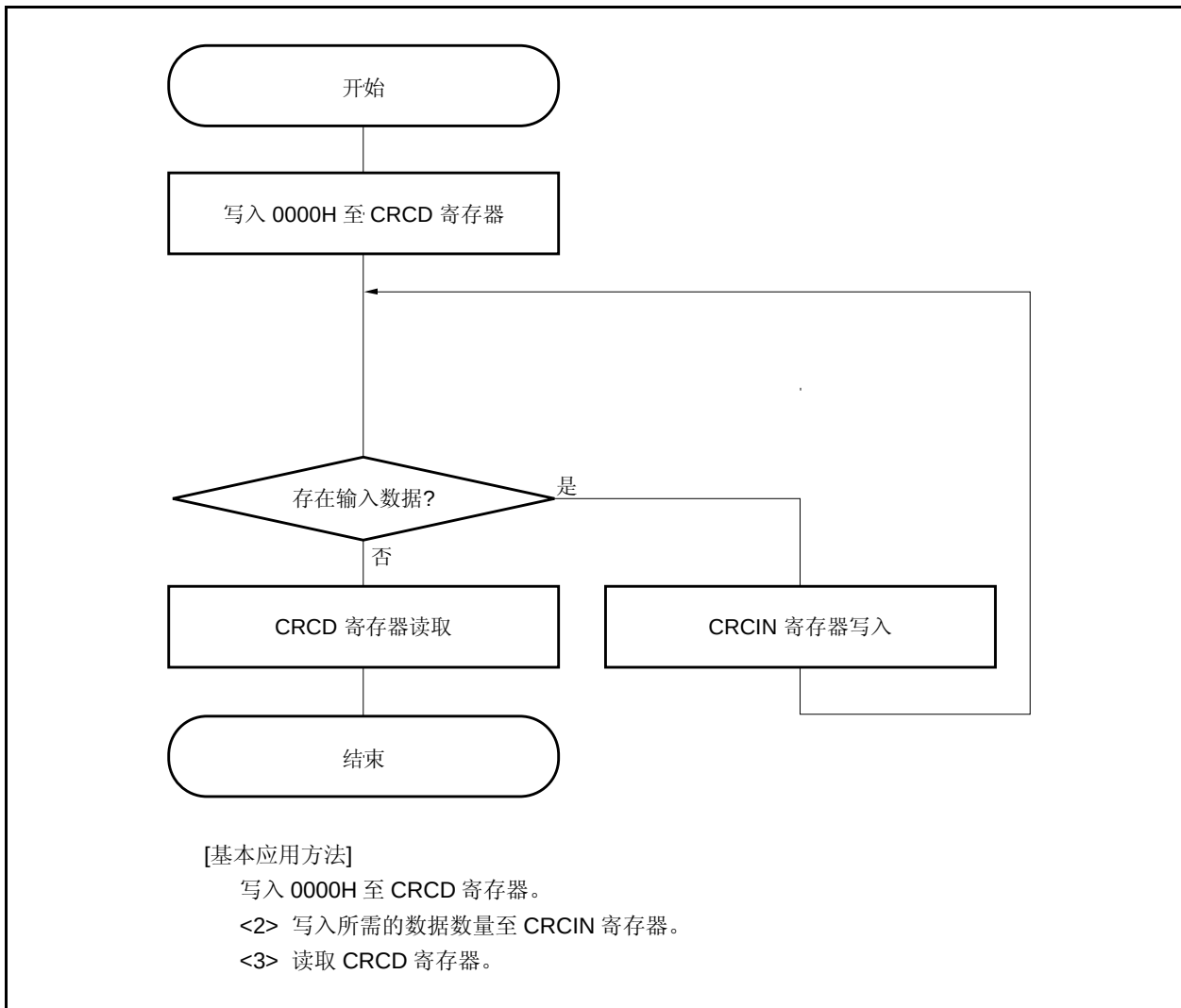
$$\begin{array}{r} \text{LSB} \swarrow \quad \quad \quad \searrow \text{MSB} \\ 1\ 0001\ 0000\ 0010\ 0001 \overline{) 1000\ 0000\ 0000\ 0000\ 0000\ 0000} \\ \underline{1000\ 1000\ 0001\ 0000\ 1} \\ 1000\ 0001\ 0000\ 1000\ 0 \\ \underline{1000\ 1000\ 0001\ 0000\ 1} \\ 1001\ 0001\ 1000\ 1000 \\ \swarrow \text{LSB} \quad \quad \quad \searrow \text{MSB} \end{array}$$

所以，CRC 码成为 $\overbrace{1001}^9 \overbrace{0001}^8 \overbrace{1000}^1 \overbrace{1000}^1$ 。由于使用 LSB 先行方式，则对应 16 进制表示的 1189H。

25.5 使用方法

以下描述如何使用 CRC 逻辑电路。

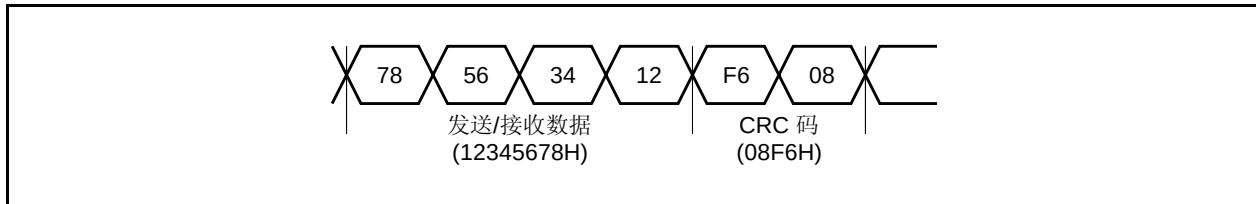
图 25-3. CRC 操作流程



当发送/接收数据由多个字节组成，如果同发送/接收数据一起发送/接收其 CRC 码，通信错误可以很容易得被检测出来。

下面以发送 12345678H (0001 0010 0011 0100 0101 0110 0111 1000B)且 LSB 先行为例，进行解释说明。

图 25-4. CRC 发送举例



发送端的设置过程。

- <1> 写入初始值 0000H 至 CRCD 寄存器。
- <2> 写入待发送的 1 字节数据至发送缓存寄存器。(与此同时，相同数据被写入 CRCIN 寄存器)
- <3> 当发送多字节数据时，每次发送数据被写入发送缓存寄存器时，写入相同数据至 CRCIN 寄存器。
- <4> 在所有数据已发送后，将 CRCD 寄存器(CRC 码)的内容写入发送缓存寄存器并进行发送。(由于是 LSB 先行发送，由低字节开始发送数据，然后才是高字节。)

接收端的设置过程。

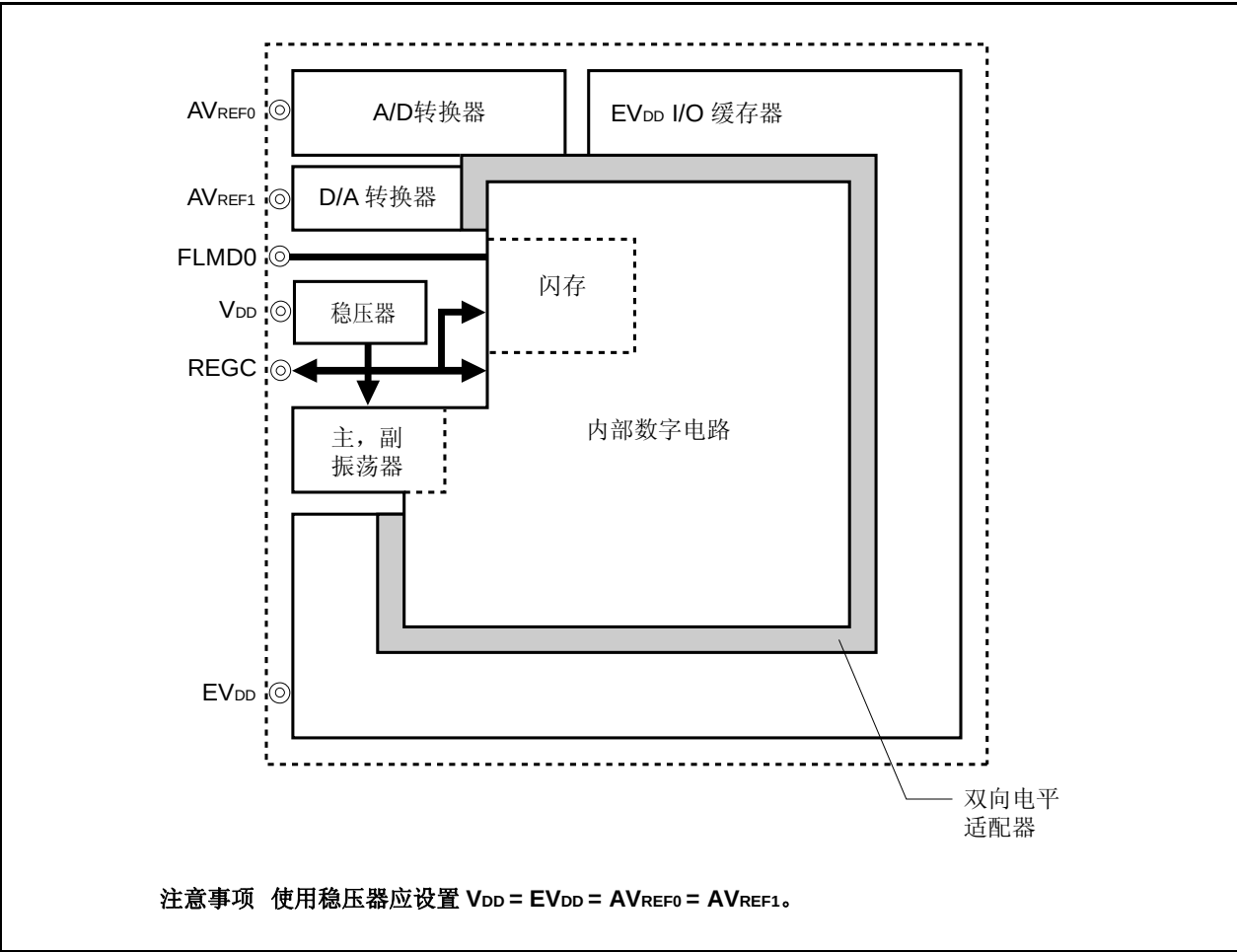
- <1> 写入初始值 0000H 至 CRCD 寄存器。
- <2> 当第一个字节接收完毕时，将接收数据写入 CRCIN 寄存器。
- <3> 如果接收多个字节数据，每次接收完毕时都将接收数据写入 CRCIN 寄存器。(正常接收情况下，当所有接收数据都已写入 CRCIN 寄存器，接收端的 CRCD 寄存器的内容应该和发送端 CRCD 寄存器的内容相同。)
- <4> 然后，CRC 码从发送端发出，与接收数据相似，将该数据写入 CRCIN 寄存器。
- <5> 当接收完包括 CRC 码在内的所有数据后，如果 CRCD 寄存器的内容为 0000H，则接收正常。如果 CRCD 寄存器的内容不是 0000H，这表示发生通信错误，所以，向发送端发送重发请求。

26.1 概述

V850ES/JJ3 包括一个稳压器，用于减少功率消耗和降低噪声。

该稳压器为振荡器模块和内部逻辑电路（除了 A/D 变换器，D/A 变换器和输出缓存器）提供一个阶梯下降的 V_{DD} 电源供电电压。稳压器输出电压设置为 2.5 V（典型值）。

图 26-1. 稳压器



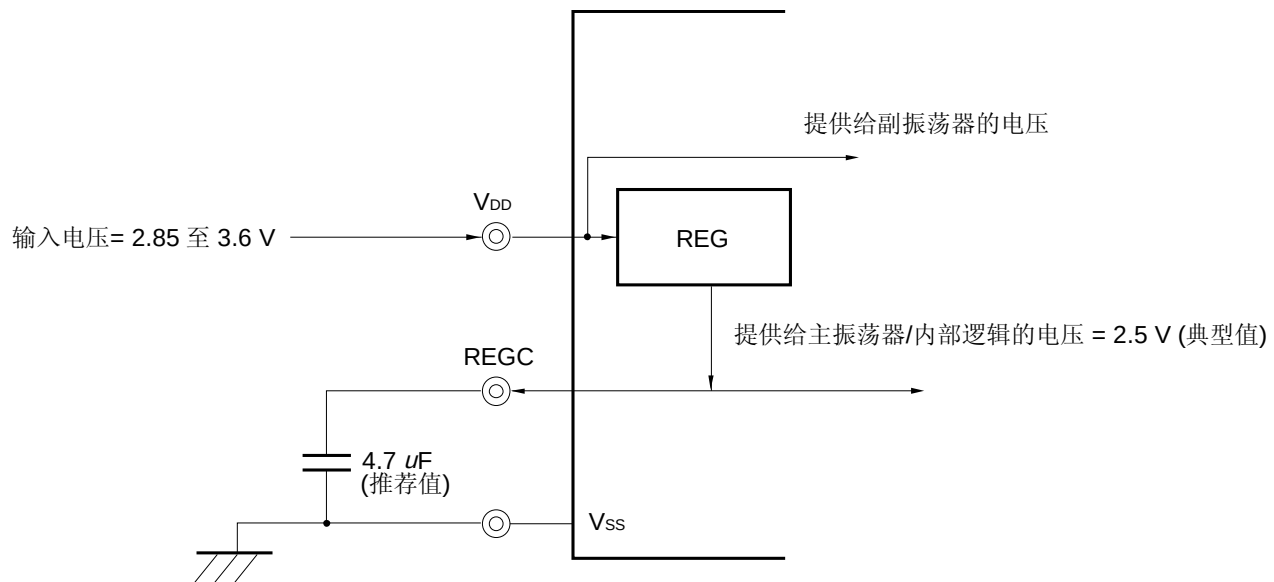
26.2 操作

V850ES/JJ3 的稳压器能在任何模式（比如正常工作模式、HALT 模式、IDLE1 模式、IDLE2 模式、STOP 模式或者在复位期间）下保持运行。

请确保在 REGC 引脚接一个电容（ $4.7\mu\text{F}$ （推荐值））来稳定稳压器的输出。

稳压器引脚的连接方式如下图所示。

图 26-2. REGC 引脚连接



第二十七章 Flash 存储器

V850ES/JJ3 单片机内置有一个 flash 存储器。

- μ PD70F3743: 384 KB flash 存储器
- μ PD70F3744: 512 KB flash 存储器
- μ PD70F3745: 768 KB flash 存储器
- μ PD70F3746: 1024 KB flash 存储器

Flash 存储器版本为开发环境和批量生产应用提供以下的优势:

- V850ES/JJ3 被焊接到目标系统之后改换软件。
- 开始批量生产时的数据调整。
- 根据不同模式的小规模生产式样提供不同的软件。
- 便于库存管理。
- 出厂后更新软件。

27.1 特性

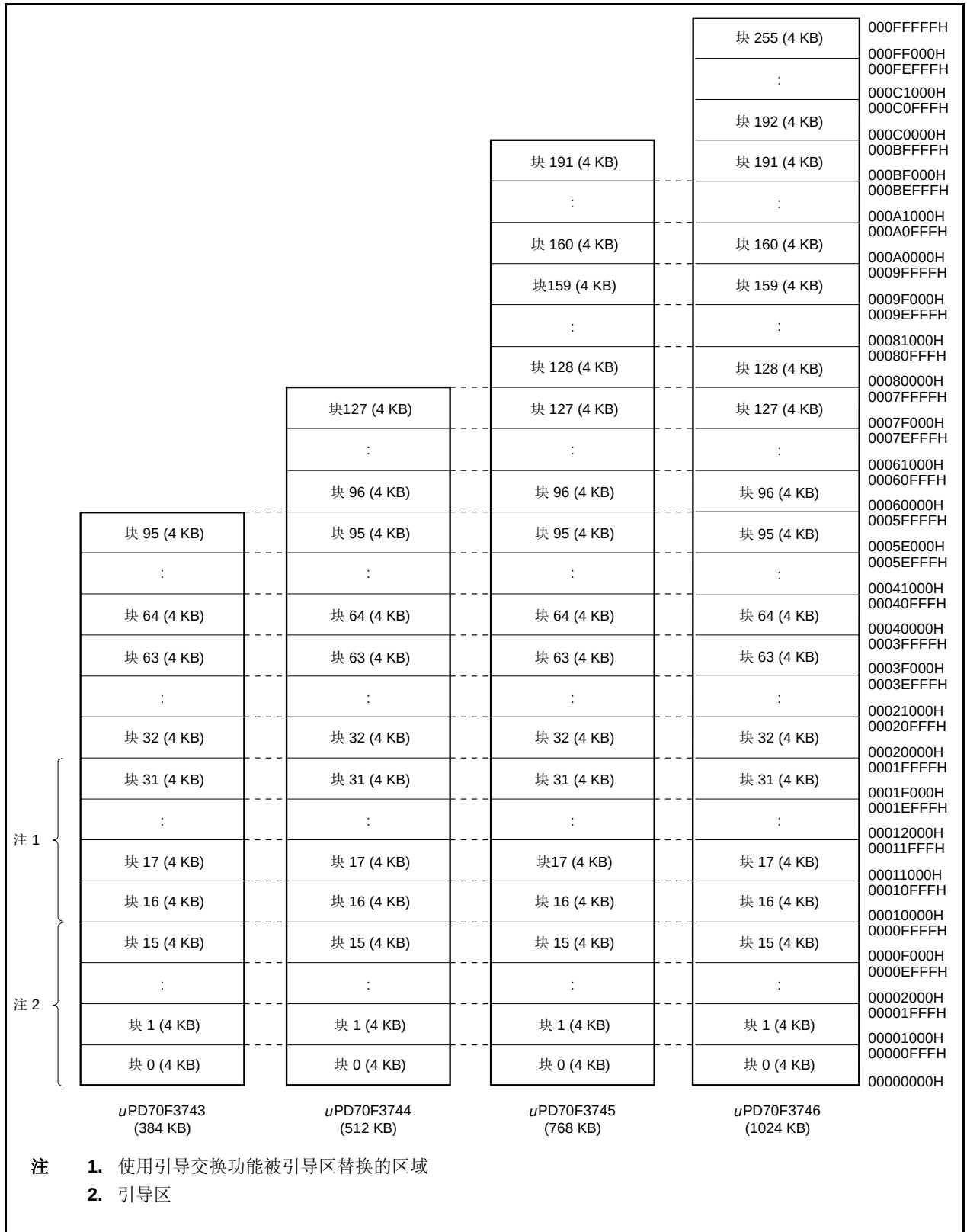
- 4-字节/1-时钟访问（当读取指令时）
- 容量: 1024 KB/768 KB/512 KB/384 KB
- 写入电压: 用一个单供电电压进行擦除/写入
- 重写方式:
 - 通过串行接口使用专用的 flash 编程器进行通信重写（在线/离线编程）。
 - 通过用户程序重写 flash 存储器（自编程）。
- 支持 flash 存储器禁止写入功能（安全功能）。
- 使用引导交换功能通过自编程方式对整个 flash 存储区安全重写。
- 自编程期间可以响应中断。

27.2 存储器配置

V850ES/JJ3 单片机内部 flash 存储区划分为 4 KB 的块，且可以按块为单位进行编程/擦除。也可以一次性擦除所有的或部分的模块。

当使用引导交换功能时，块地址为 0 至 15 的物理存储区被块地址为 16 至 31 的物理存储区替换。关于引导交换功能的详情，参见 **27.5 通过自编程重写**。

图 27-1. Flash 存储器映射



27.3 功能概述

不管 V850ES/JJ3 是否被焊接到目标系统上（离线/在线编程），V850ES/JJ3 的内部 flash 存储器都可通过专用 flash 编程器的重写功能进行重写。

此外，还支持安全功能，可以禁止对已经写入内部 flash 存储器的用户程序进行重写，因此，程序不能被未经授权者改变。

假设目标系统生产/出厂后改变程序，则使用用户程序（自编程）的重写功能是理想的选择。同样还支持安全重写整个 flash 存储器的引导交换功能。此外，自编程期间还支持中断服务，所以 flash 存储器可以在多种条件下进行重写，比如和外部设备通信时。

表 27-1. 重写方式

重写方式	功能概述	工作模式
在线编程	设备安装到目标系统之后，flash 存储器可使用专用 flash 编程器重写。	Flash 存储器编程模式
离线编程	在设备安装到目标系统之前，flash 存储器可通过专用 flash 存储器编程器和专用程序适配板（FA 系列）重写。	
自编程	Flash 存储器可通过执行用户程序来重写，用户程序通过离线/在线编程预先写入 flash 存储器。（在自编程时，不能从内部 flash 存储器区取指令，也不能对内部 flash 存储器区进行数据存取。因此，重写程序必须预先传送到内部 RAM 或外部存储器。）	正常操作模式

备注 FA 系列为 Naito Densei Machida Mfg. Co., Ltd 的产品。

<R>

表 27-2. 基本功能

功能	功能概述	支持 (√: 可支持, ×: 不支持)	
		在线/离线编程	自编程
空白检测	检查整个存储器的擦除状态。	√	√
片擦除	一次全部擦除整个存储器的内容。	√	× ^注
块擦除	指定存储块的内容被擦除。	√	√
编程	执行到指定地址的写入，并校验检查写入的等级是否安全	√	√
校验/检查和	将从 flash 存储器读出的数据和由 flash 编程器发送的数据进行比较。	√	× (可由用户程序读取)
读取	写入 flash 存储区的数据被读取。	√	×
安全设置	禁止使用片擦除命令、块擦除命令、编程命令以及读取指令，且禁止重写引导区。	(	((仅当设置由使能变为禁止时，可以支持)

注 为块擦除功能选择了整片存储器，这时是可能的。

下表列出了安全功能。出厂之后，默认禁止片擦除命令、禁止块擦除命令、禁止编程命令、禁止读取命令以及禁止重写引导块簇功能，且可通过在线/离线编程进行重写来完成安全设置。各个安全功能可以同时与其它功能组合使用。

表 27-3. 安全功能

功能	功能概述
禁止片擦除命令	在所有块上禁止执行片擦除命令和块擦除命令。一旦设定为禁止之后，所有的关于禁止的设置都无法重新初始化设定， 因为片擦除命令不能执行。
禁止块擦除命令	在所有块上禁止执行块擦除指令。关于禁止的设置可以通过执行片擦除指令来初始化。
禁止编程命令	所有模块上禁止执行编程和块擦除命令。关于禁止的设置可以通过执行片擦除命令来初始化。
禁止读取命令	所有块上禁止执行读取命令。关于禁止的设置可以通过执行片擦除命令来初始化。
禁止重写引导块簇	可以保护从块 0 到指定块的引导块簇的内容。禁止对已经受到保护的引导块簇执行重写（擦除和写入）。即使执行片擦除命令，所有的关于禁止的设置也无法重新初始化设定。 可以指定的块最大编号如下： 384 KB 版本： 95 块 512/768/1024 KB 版本： 127 块

<R>

<R>

表 27-4. 安全设置

功能	设置各个安全性能时的擦除、写入、读取操作 (√: 可执行, ×: 不可执行, -: 不支持)		安全设置的注意事项	
	在线/离线编程	自编程	在线/离线编程	自编程
禁止片擦除命令	片擦除命令: × 块擦除命令: × 编程命令: √ ^{注1} 读取命令: √	片擦除: () 块擦除 (Flash 块擦除): () 写入 (Flash 字写入 FlashWordWrite): √ 读取 (FlashWordRead): √	关于禁止的设置 无法被初始化	仅当设置由使能 变为禁止时, 可 以支持。
禁止块擦除命令	片擦除命令: () 块擦除命令: () 编程命令: () 读取命令: ()	片擦除: () 块擦除 (Flash 块擦除): () 写入 (FlashWordWrite): √ 读取 (FlashWordRead): √	关于禁止的设置 可以通过片擦除 命令进行初始化	
禁止编程命令	片擦除命令: √ 块擦除命令: × 编程命令: × 读取命令: √	片擦除: - 块擦除 (Flash 块擦除): √ 写入 (FlashWordWrite): √ 读取 (FlashWordRead): √	关于禁止的设置 可以通过片擦除 命令进行初始化	
禁止读取命令	片擦除命令: √ 块擦除命令: √ 编程命令: √ 读取命令: ×	片擦除: - 块擦除 (Flash 块擦除): √ 写入 (FlashWordWrite): () 读取 (Flash 字读取 FlashWordRead): √	关于禁止的设置 可以通过片擦除 命令进行初始化	
禁止重写引导 区簇	片擦除命令: × 块擦除命令: × ^{注2} 编程命令: × ^{注2} 读取命令: : √	片擦除: - 块擦除 (Flash块擦除): × ^{注2} 写入 (FlashWordWrite): (注 2 读取 (Flash字读取 FlashWordRead): √	关于禁止的设置 无法被初始化	仅当设置由使能 变为禁止时, 可 以支持。 ^{注3}

- 注
1. 在这种情况下, 因为擦除命令无效, 与已写入数据不同的数据将无法被写入 flash 存储器。
 2. 在引导块簇之外可以执行。
 3. 在复位输入之后, 引导块簇重写禁止功能变为有效。

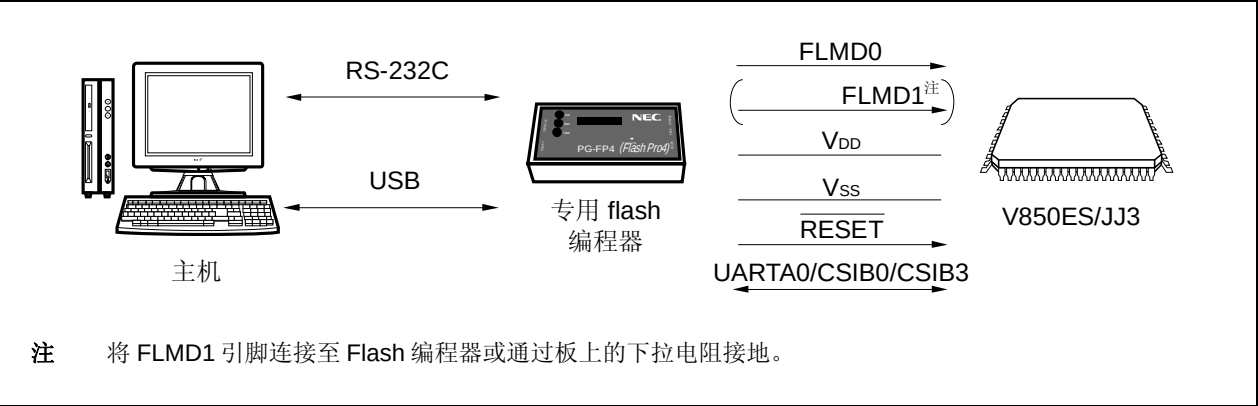
27.4 通过专用 flash 编程器进行重写

V850ES/JJ3 被安装到目标系统（在线编程）之后，flash 存储器可通过专用 flash 编程器进行重写。Flash 存储器也可以使用专用的编程适配器（FA 系列）在芯片安装到目标系统前进行重写（离线编程）。

27.4.1 编程环境

下面显示了将程序写入 V850ES/JJ3 的 flash 存储器所需的环境。

图 27-2. 程序写入 flash 存储器所需的环境



需要一台主机用来控制专用 flash 编程器。

UARTA0，CSIB0 或 CSIB3 被用作专用 flash 编程器和 V850ES/JJ3 之间的接口，执行写入、擦除等操作。离线编程需要一个专用编程适配器（FA 系列）。

- FA-70F3368GJ-UEN-RX（已连线）
- FA-144GJ-UEN-A（未连线：需要连线）

备注 FA 系列为 Naito Densai Machida Mfg. Co., Ltd 的产品。

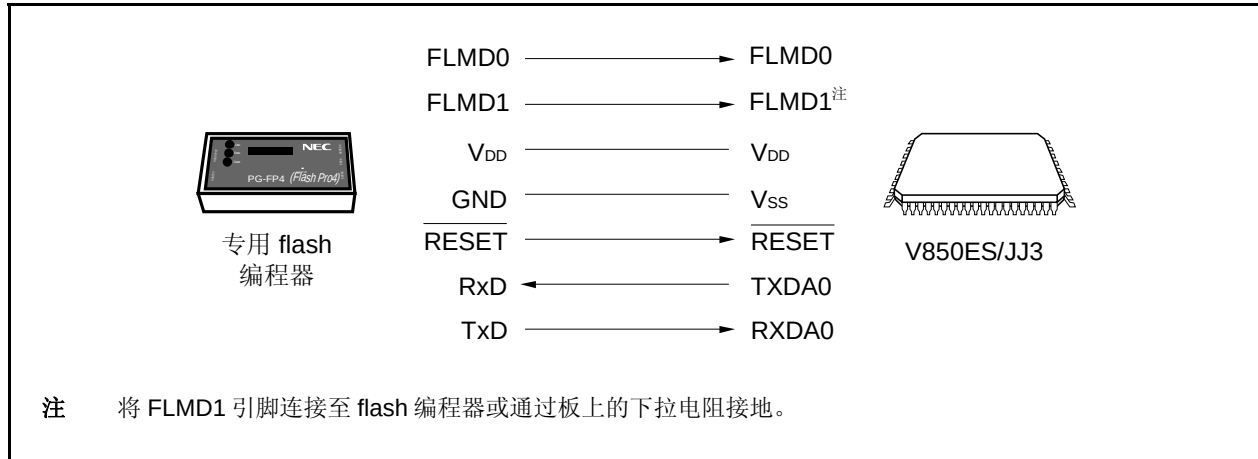
27.4.2 通信模式

专用 flash 编程器和 V850ES/JJ3 的通信方式是串行通信，使用 V850ES/JJ3 的 UARTA0，CSIB0 或 CSIB3 接口。

(1) UARTA0

传输速率： 9,600 至 153,600 bps

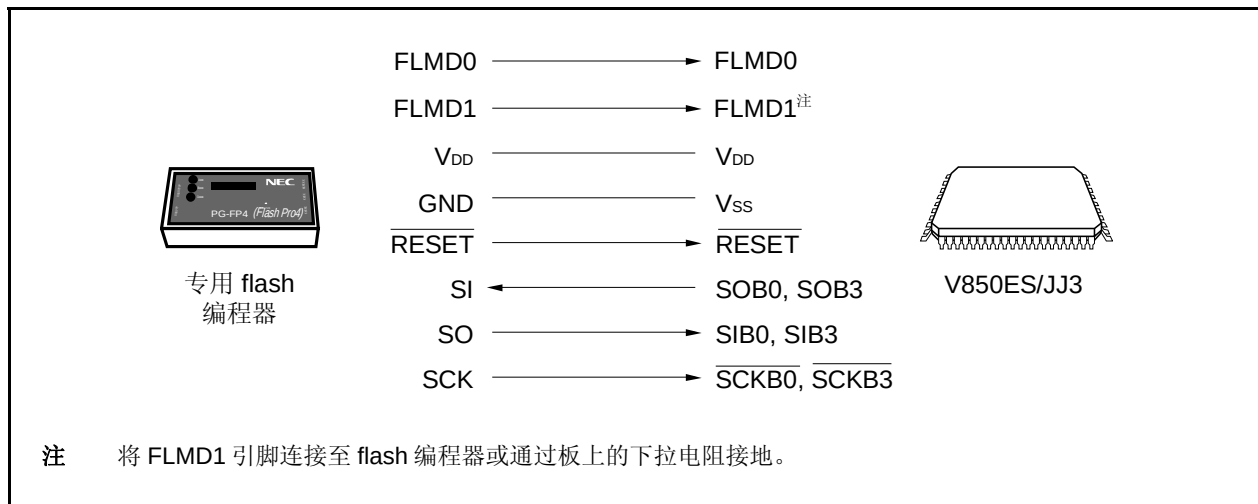
图 27-3. 和专用 flash 编程器通信 (UARTA0)



(2) CSIB0, CSIB3

串行时钟：2.4 kHz 至 2.5 MHz (MSB 先行)

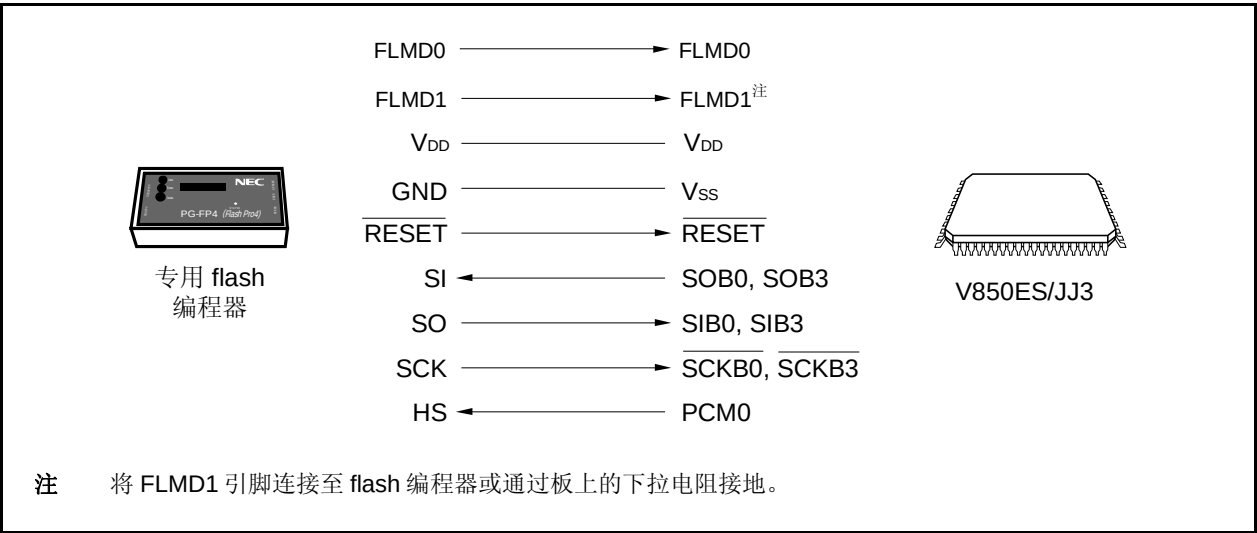
图 27-4. 和专用 flash 编程器通信 (CSIB0, CSIB3)



(3) CSIB0 + HS, CSIB3 + HS

串行时钟：2.4 kHz 至 2.5 MHz（MSB 先行）

图 27-5. 和专用 flash 编程器通信（CSIB0 + HS）



专用 flash 编程器输出串行时钟，V850ES/JJ3 作为从机工作。

当专用 flash 编程器使用 PG-FP4 时，会产生下面的信号到 V850ES/JJ3。详情参见 PG-FP4 用户手册（U15260E）。

表 27-5. 专用 Flash 编程器的信号连接（PG-FP4）

PG-FP4			V850ES/JJ3	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	写入使能/禁止	FLMD0	○	○	○
FLMD1	输出	写入使能/禁止	FLMD1	○ ^{注1}	○ ^{注1}	○ ^{注1}
VDD	(	VDD 电压产生/电压监控	VDD	┐	┐	○
GND	(	地	VSS	┐	┐	┐
CLK	输出	输出至 V850ES/JJ3 的时钟	X1, X2	× ^{注2}	× ^{注2}	× ^{注2}
RESET	输出	复位信号	RESET	○	○	○
SI/RxD	输入	接收信号	SOB0, SOB3/TXDA0	○	○	○
SO/TxD	输出	发送信号	SIB0, SIB3/RXDA0	○	○	○
SCK	输出	传输时钟	SCKB0, SCKB3	×	○	○
HS	输入	CSIB0 + HS, CSIB3 + HS 通信的握手信号	PCM0	×	×	○

- 注 1. 将这些引脚按图 27-6 和图 27-7 所示进行连接，或通过板上的下拉电阻将其接地。
2. 不能通过 flash 编程器的 CLK 引脚提供时钟。在版上创建一个振荡器来提供时钟。

备注 ○：必须进行连接。

×：不必进行连接。

表 27-6. V850ES/JJ3 的 flash 写入适配器 (FA-144GJ-UEN-A) 的连线 (1/2)

Flash 编程器 (FG-FP4) 连接引脚			FA 板的引脚名称	当使用 CSIB0 + HS 时		当使用 CSIB0 时		当使用 UARTA0 时	
信号名称	I/O	引脚功能		引脚名称	引脚编号	引脚名称	引脚编号	引脚名称	引脚编号
SI/RxD	输入	接收信号	SI	P41/SOB0/SCL01	23	P41/SOB0/SCL01	23	P30/TXDA0/SOB4	25
SO/TxD	输出	发送信号	SO	P40/SIB0/SDA01	22	P40/SIB0/SDA01	22	P31/RXDA0/INTP7/SIB4	26
SCK	输出	传输时钟	SCK	P42/ $\overline{\text{SCKB0}}$	24	P42/ $\overline{\text{SCKB0}}$	24	不需要	—
CLK	输出	至 V850ES/JJ3 的时钟	X1	不需要	—	不需要	—	不需要	—
			X2	不需要	—	不需要	—	不需要	—
/RESET	输出	复位信号	/RESET	RESET	14	RESET	14	RESET	14
FLMD0	输出	写入电压	FLMD0	FLMD0	8	FLMD0	8	FLMD0	8
FLMD1	输出	写入电压	FLMD1	PDL5/AD5/FLMD1	110	PDL5/AD5/FLMD1	110	PDL5/AD5/FLMD1	110
HS	输入	CSI0 + HS 通信的握手信号	RESERVE/HS	PCM0/WAIT	85	不需要	—	不需要	—
VDD	—	VDD 电压产生/电压监控	VDD	V _{DD}	9	V _{DD}	9	V _{DD}	9
				EV _{DD}	34, 104	EV _{DD}	34, 104	EV _{DD}	34, 104
				AV _{REF0}	1	AV _{REF0}	1	AV _{REF0}	1
				AV _{REF1}	5	AV _{REF1}	5	AV _{REF1}	5
GND	—	地	GND	V _{SS}	11	V _{SS}	11	V _{SS}	11
				AV _{SS}	2	AV _{SS}	2	AV _{SS}	2
				EV _{SS}	33, 103	EV _{SS}	33, 103	EV _{SS}	33, 103

注意事项 1. 请确保将 REGC 引脚通过 4.7 μ F (推荐值) 电容接地。

2. 不能通过 flash 编程器的 CLK 引脚提供时钟。在板上创建一个振荡器来提供时钟。

表 27-6. V850ES/JJ3 的 flash 写入适配器 (FA-144GJ-UEN-A) 的连线 (2/2)

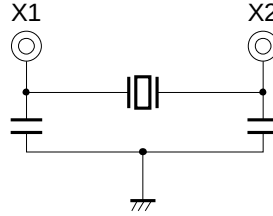
Flash 编程器 (FG-FP4) 连接引脚			FA 板的引脚 名称	当使用 CSIB3 + HS 时		使用 CSIB3 时	
信号 名称	I/O	引脚功能		引脚名称	引脚 编号	引脚名称	引脚 编号
SI/RxD	输入	接收信号	SI	P911/A11/SOB3	72	P911/A11/SOB3	72
SO/TxD	输出	发送信号	SO	P910/A10/SIB3	71	P910/A10/SIB3	71
SCK	输出	传输时钟	SCK	P912/A12/ $\overline{\text{SCKB3}}$	73	P912/A12/ $\overline{\text{SCKB3}}$	73
CLK	输出	至 V850ES/ JJ3 的 时钟	X1	不需要	–	不需要	–
			X2	不需要	–	不需要	–
/RESET	输出	复位信号	/RESET	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14
FLMD0	输出	写入电压	FLMD0	FLMD0	8	FLMD0	8
FLMD1	输出	写入电压	FLMD1	PDL5/AD5/FLMD1	110	PDL5/AD5/FLMD1	110
HS	输入	CSI0 + HS 通信的 握手信号	RESERVE/HS	$\overline{\text{PCM0/WAIT}}$	85	不需要	–
VDD	–	VDD 电压产生/ 电 压监控	VDD	V _{DD}	9	V _{DD}	9
				EV _{DD}	34, 104	EV _{DD}	34, 104
				AV _{REF0}	1	AV _{REF0}	1
				AV _{REF1}	5	AV _{REF1}	5
GND	–	地	GND	V _{SS}	11	V _{SS}	11
				AV _{SS}	2	AV _{SS}	2
				EV _{SS}	33, 103	EV _{SS}	33, 103

- 注意事项
1. 请确保将 REGC 引脚通过 4.7 μF (推荐值) 电容接地。
 2. 不能通过 flash 编程器的 CLK 引脚提供时钟。在板上创建一个振荡器来提供时钟。

图 27-6. V850ES/JJ3 Flash 写入适配器 (FA-144GJ-UEN-A) 的连线示例
(CSIB0 + HS 模式) (2/2)

- 注
1. FLMD1 引脚的接线如下所示，或通过板上的下拉电阻将其接地。
 2. 当使用 CSIB3 时使用的引脚。
 3. 在 flash 写入适配器上创建一个振荡器（如虚线所示）来提供时钟。
这里给出一个振荡器的示例。

示例：



4. 当使用 UARTA0 时使用的引脚。

注意事项 不要将高电平输入到 $\overline{\text{DRST}}$ 引脚。

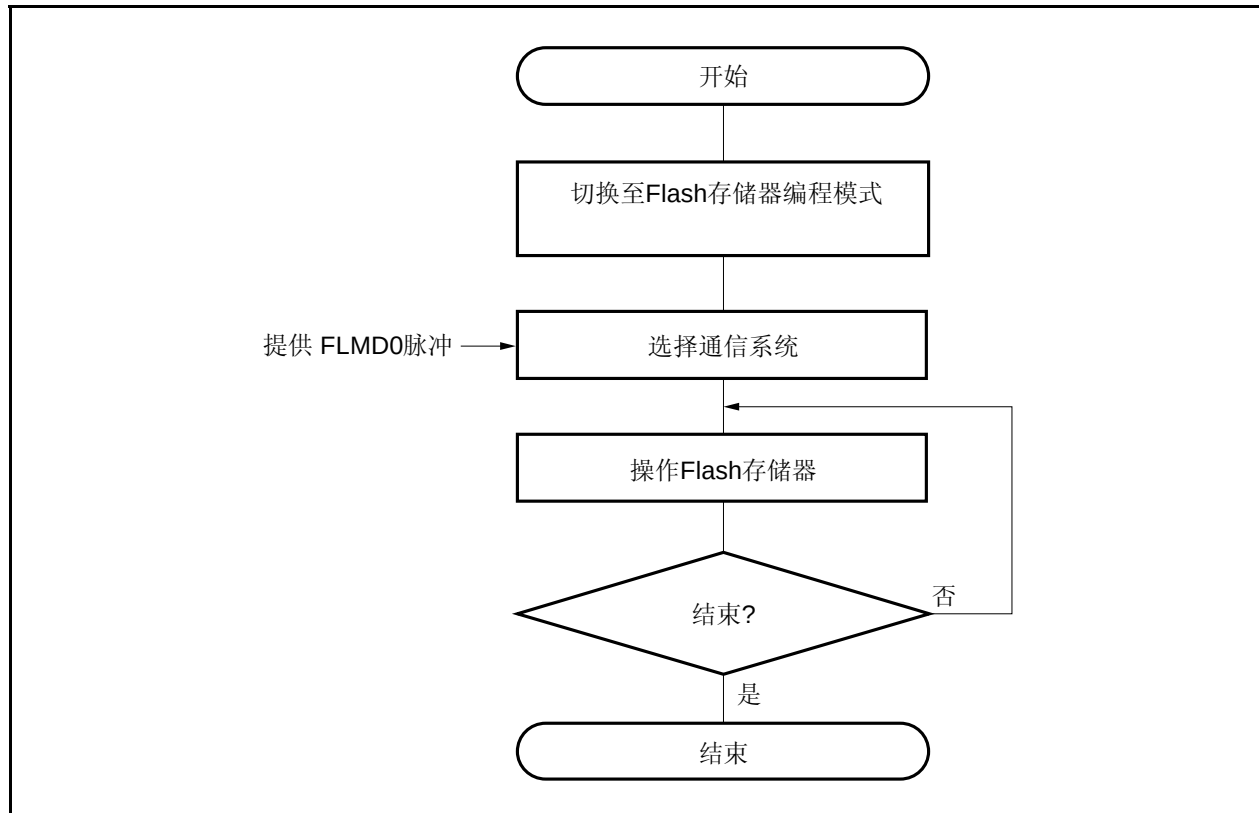
- 备注
1. 对于未列出的引脚的处理，按照未使用引脚的处理方法。（参见 2.3 引脚 I/O 电路类型、I/O 缓冲器电源以及未使用引脚的连接）。
 2. 该适配器适用于 144 针塑封 LQFP 封装。

27.4.3 Flash 存储器控制

以下显示了操作 flash 存储器的步骤。

<R>

图 27-7. 操作 flash 存储器的步骤

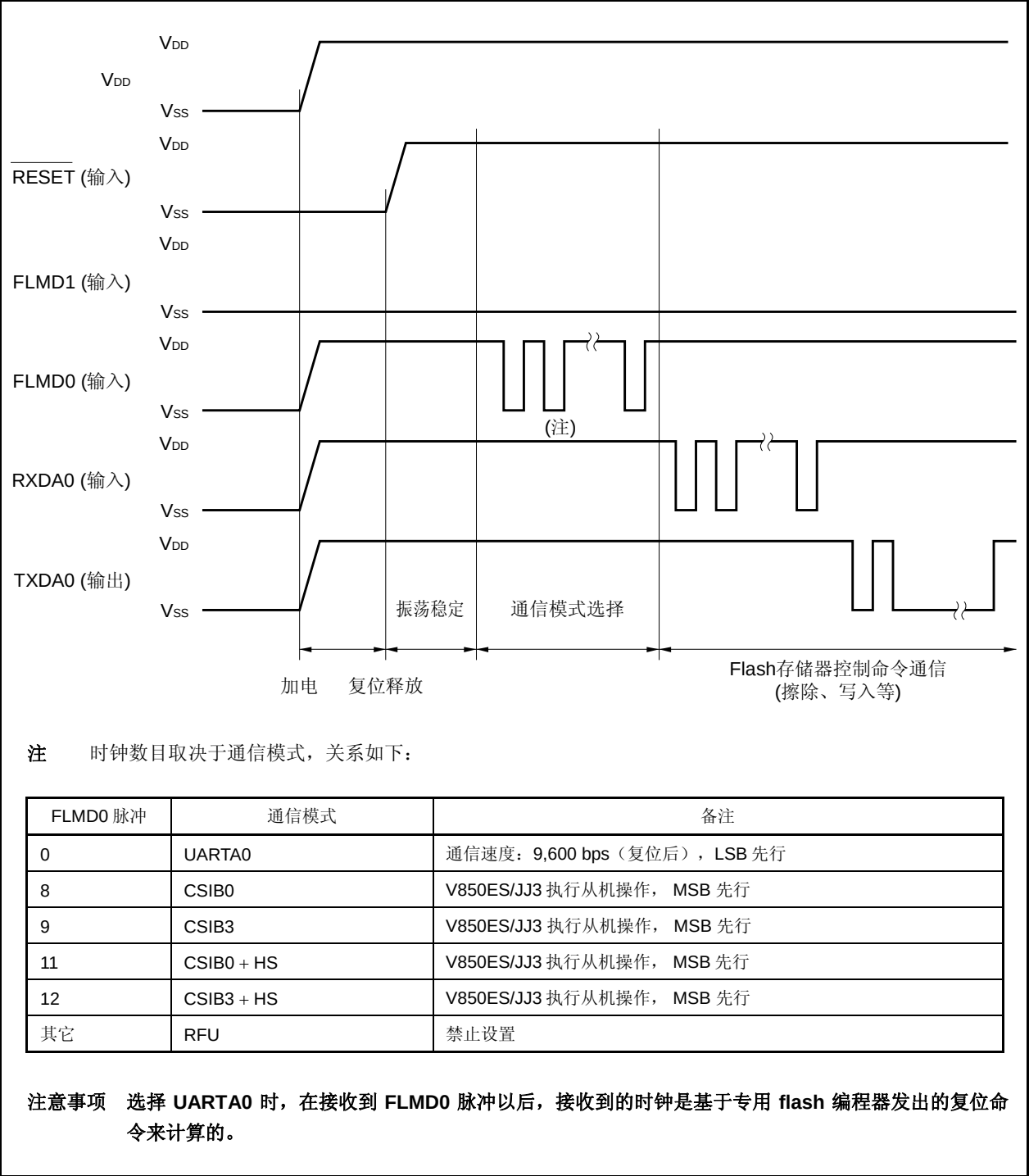


27.4.4 通信模式的选择

在 V850ES/JJ3 单片机中，切换至 flash 存储器编程模式后，通过输入脉冲（最多 12 个脉冲）至 FLMD0 引脚来选择通信模式。FLMD0 脉冲由专用 flash 编程器产生。

下面显示了脉冲数目和通信模式之间的关系。

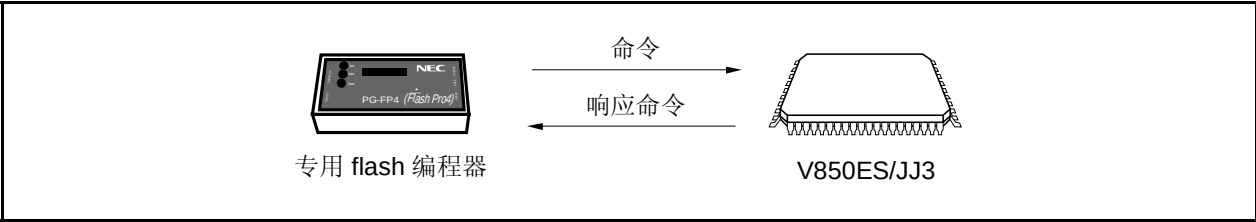
图 27-8. 通信模式的选择



27.4.5 通信命令

V850ES/JJ3 通过指令的方式与专用 flash 编程器进行通信。从专用 flash 编程器发至 V850ES/JJ3 的信号称为“命令”。从 V850ES/JJ3 发出至 flash 专用编程器的应答信号称为“应答命令”。

图 27-9. 通信命令



下面显示了 V850ES/JJ3 用来控制 flash 存储器的命令。所有的这些命令都由专用 flash 编程器产生，而 V850ES/JJ3 执行与命令相应的处理。

表 27-7. 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	
空白检查	块空白检查命令	√	√	√	检查指定块的存储内容是否已被正确擦除。
擦除	片擦除命令	√	√	√	擦除整个存储器的内容。
	块擦除命令	√	√	√	擦除指定块的存储内容。
写入	编程命令	√	√	√	写入指定的地址范围，且执行内容检验检查。
校验	校验命令	√	√	√	比较存储器中指定地址范围的内容和 flash 编程器传送的数据
	校验和命令	√	√	√	读取指定地址范围内的检查和。
<R> 读取	读取命令	√	√	√	读取写入到 flash 存储器中的数据
系统设置控制	硅签字命令	√	√	√	读取硅签字信息。
	安全设置命令	√	√	√	禁止片擦除命令、块擦除命令、编程命令、读取命令和引导区重写。

27.4.6 引脚连接

当执行在线写入时，将目标系统上安装一个连接器用以连接专用 flash 编程器。另外，电路板上还要具有将正常操作模式切换至 flash 存储器编程模式的功能。

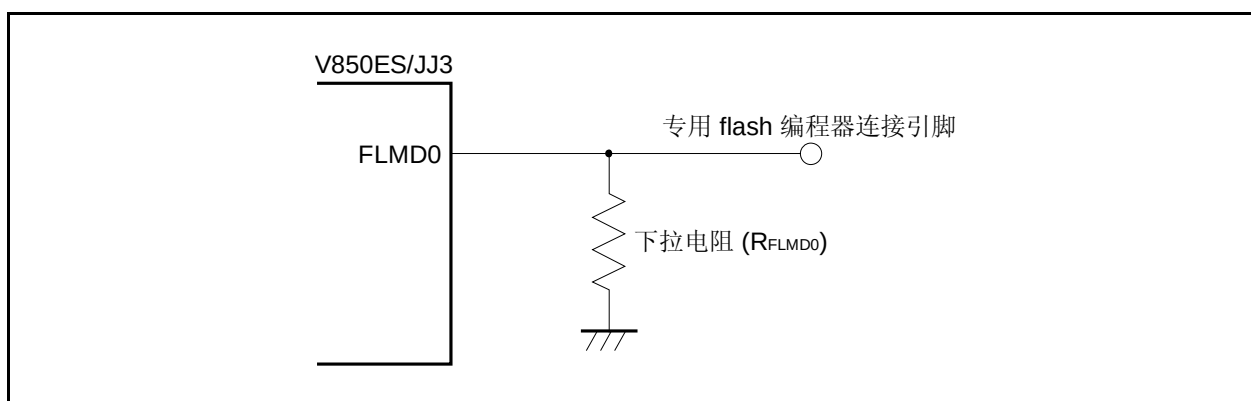
在 flash 存储器编程模式下，所有在 flash 存储器编程过程中未使用的引脚，其状态与复位后的瞬时状态相同。因此，当外部设备不能识别复位后的瞬时状态时，需要对引脚进行处理。

(1) FLMD0 引脚

在正常操作模式下，将电压值为 V_{SS} 的电平输入至 FLMD0 引脚。在 flash 存储器编程模式，为 FLMD0 引脚提供电平为 V_{DD} 的写入电压。

因为在自编程模式时，FLMD0 引脚被用作写入保护引脚，因此在写入 flash 存储器之前，必须通过端口控制等手段给 FLMD0 引脚提供一个电平为 V_{DD} 的电压。详情参见 27.5.5 (1) FLMD0 引脚。

图 27-10. FLMD0 引脚连接示例



(2) FLMD1 引脚

当 FLMD0 引脚输入 0V 时，FLMD1 引脚不起作用。当给 FLMD0 引脚提供 V_{DD} 电平时，进入 flash 存储器编程模式，因此必须给 FLMD1 引脚输入 0V。下面是 FLMD1 引脚的一个连接示例。

图 27-11. FLMD1 引脚连接示例

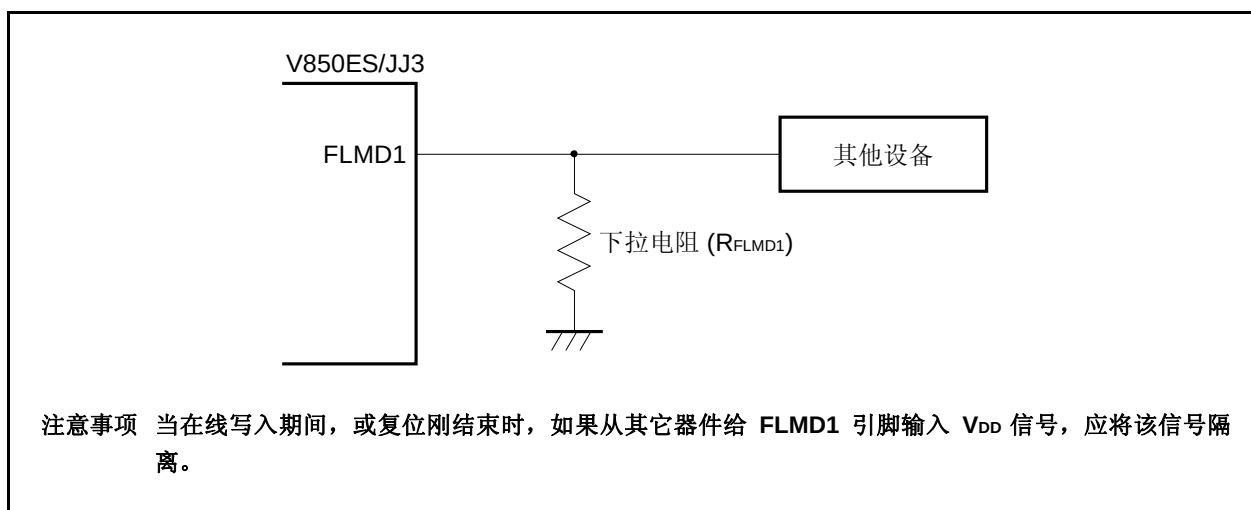


表 27-8. 复位解除后 FLMD0 和 FLMD1 引脚与操作模式之间的关系

FLMD0	FLMD1	操作模式
0	忽略	正常操作模式
V _{DD}	0	Flash 存储器编程模式
V _{DD}	V _{DD}	禁止设置

(3) 串行接口引脚

下面给出各个串行接口使用的引脚。

表 27-9. 串行接口所用引脚

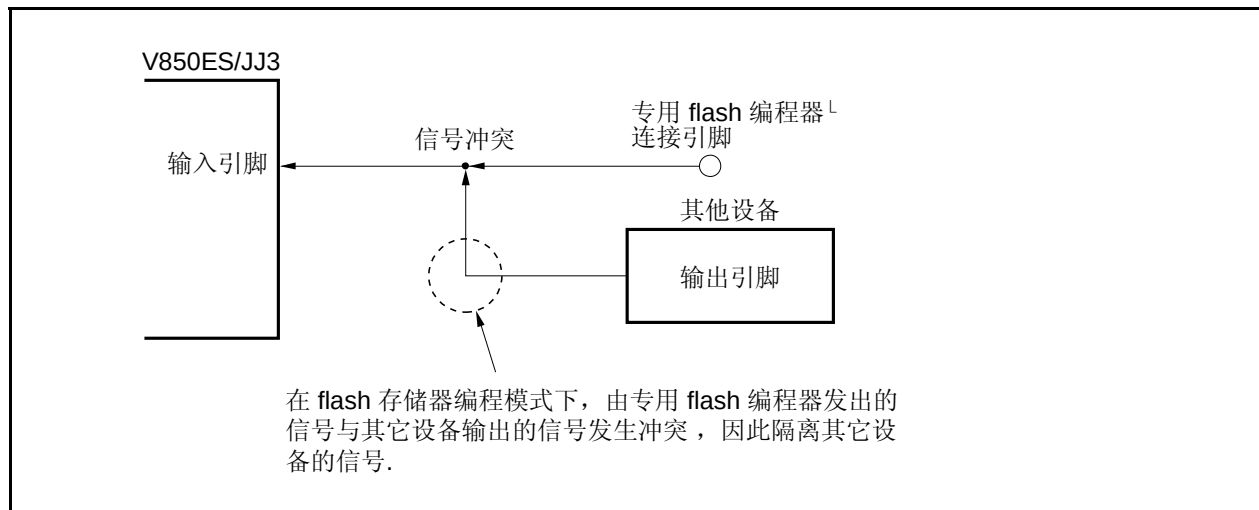
串行接口	所用引脚
UARTA0	TXDA0, RXDA0
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB3	SOB3, SIB3, $\overline{\text{SCKB3}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, PCM0
CSIB3 + HS	SOB3, SIB3, $\overline{\text{SCKB3}}$, PCM0

当将专用 flash 编程器接到某个串行接口引脚，该接口已经与板上其它设备连接时，应注意避免信号冲突或其它设备的误操作。

(a) 信号冲突

当专用 flash 编程器（输出）连接到某个串行接口引脚（输入），该接口已经与板上其它设备（输出）连接时，会发生信号冲突。为避免信号冲突，隔离其与其它设备的连接或将其它设备设为输出高阻状态。

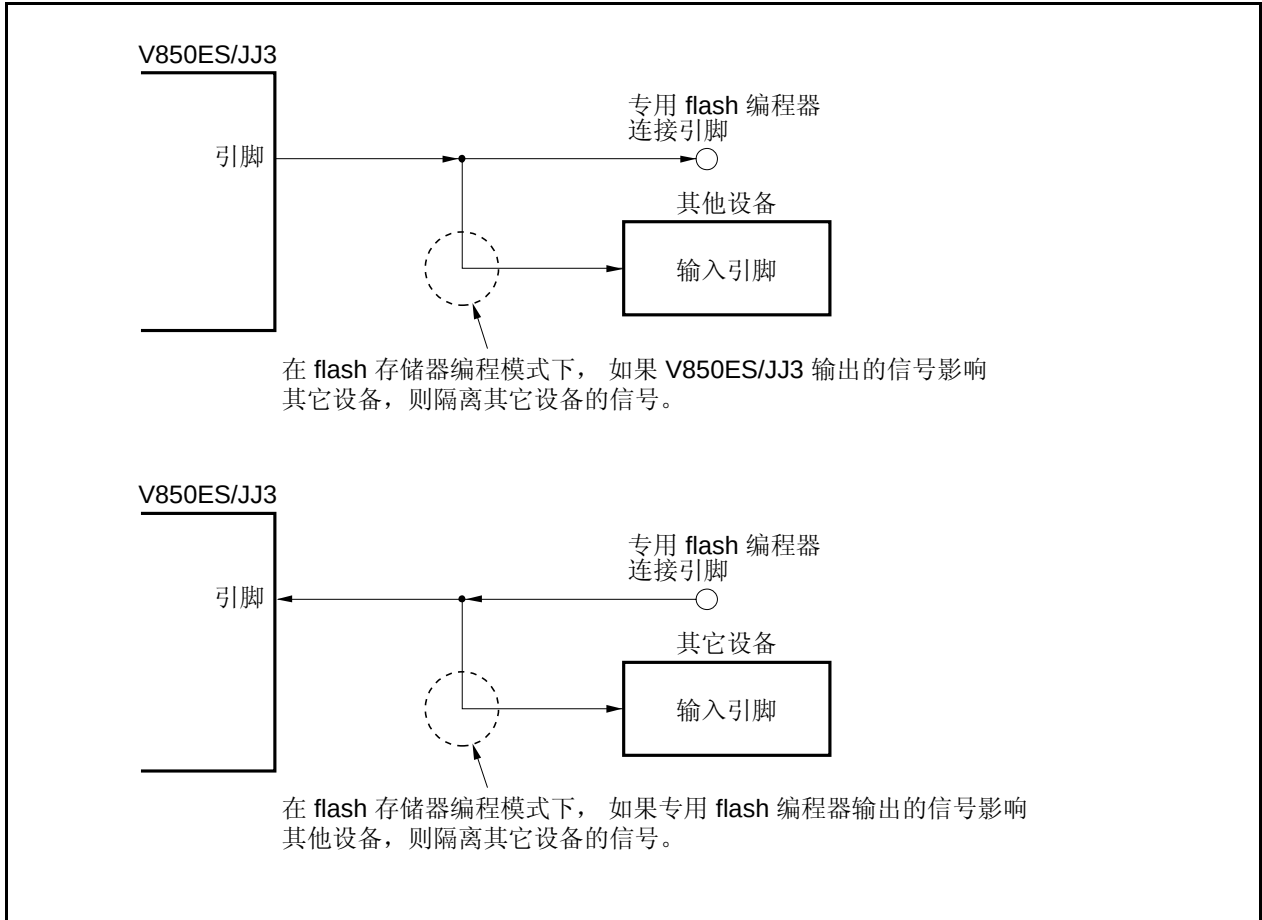
图 27-12. 信号冲突（串行接口输入引脚）



(b) 其它设备的误操作

当专用 flash 编程器（输出或输入）接到到某个串行接口引脚（输入或输出），该接口已经与板上其它设备（输入）连接时，信号输出到其它设备，导致设备的误操作。为避免这种情况。隔离与其它设备的连接。

图 27-13. 其它设备的误操作

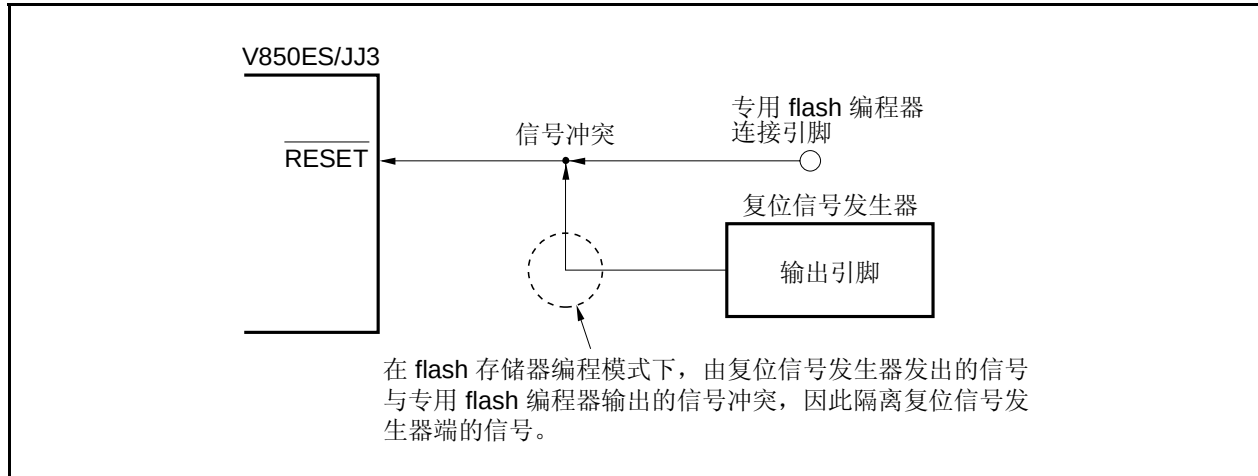


(4) RESET 引脚

当专用 flash 编程器的复位信号接到 $\overline{\text{RESET}}$ 引脚，该引脚已经和板上的复位信号发生器相连时，会发生信号冲突。为避免信号间的冲突，隔离其它复位信号发生器的连接。

在 Flash 存储器编程模式下，当复位信号由用户系统输入时，编程操作将不能正确执行。所以，不要输入任何专用 flash 编程器复位信号之外的信号。

图 27-14. 信号冲突 ($\overline{\text{RESET}}$ 引脚)



(5) 端口引脚 (包括 NMI)

当系统切换至 flash 存储器编程模式时，所有在 flash 存储器编程模式中不使用的引脚处于与复位后瞬时状态相同的状态。如果连接至各个端口的外部器件无法识别端口复位后的瞬时状态，引脚就需要适当的处理，如通过电阻接至 V_{DD} 或 V_{SS} 。

(6) 其它信号引脚

X1、X2、XT1、XT2 和 REGC 的连接状态，与正常操作模式相同。

flash 存储器编程期间，向 $\overline{\text{DRST}}$ 引脚输入低电平或开路断开。不要输入高电平。

(7) 供电

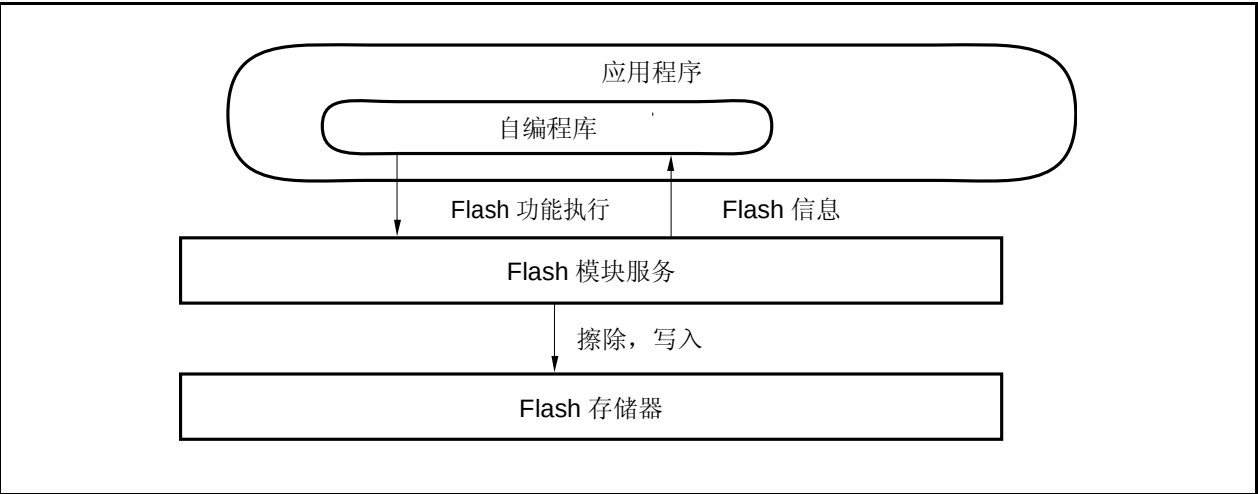
提供正常操作模式时的电源 (V_{DD} 、 V_{SS} 、 EV_{DD} 、 EV_{SS} 、 AV_{REF0} 、 AV_{REF1} 、 AV_{SS})。

27.5 通过自编程进行重写

27.5.1 概述

V850ES/JJ3 支持 flash 宏服务，该服务允许用户程序自行重写内部 flash 存储器。通过用户应用程序对该接口和一个自编程库的使用来重写 flash 存储器，flash 存储器可被预先传送到内部 RAM 或外部存储器的用户程序重写。从而可以更新用户程序并现场改写常量数据。

图 27-15. 自编程的概念

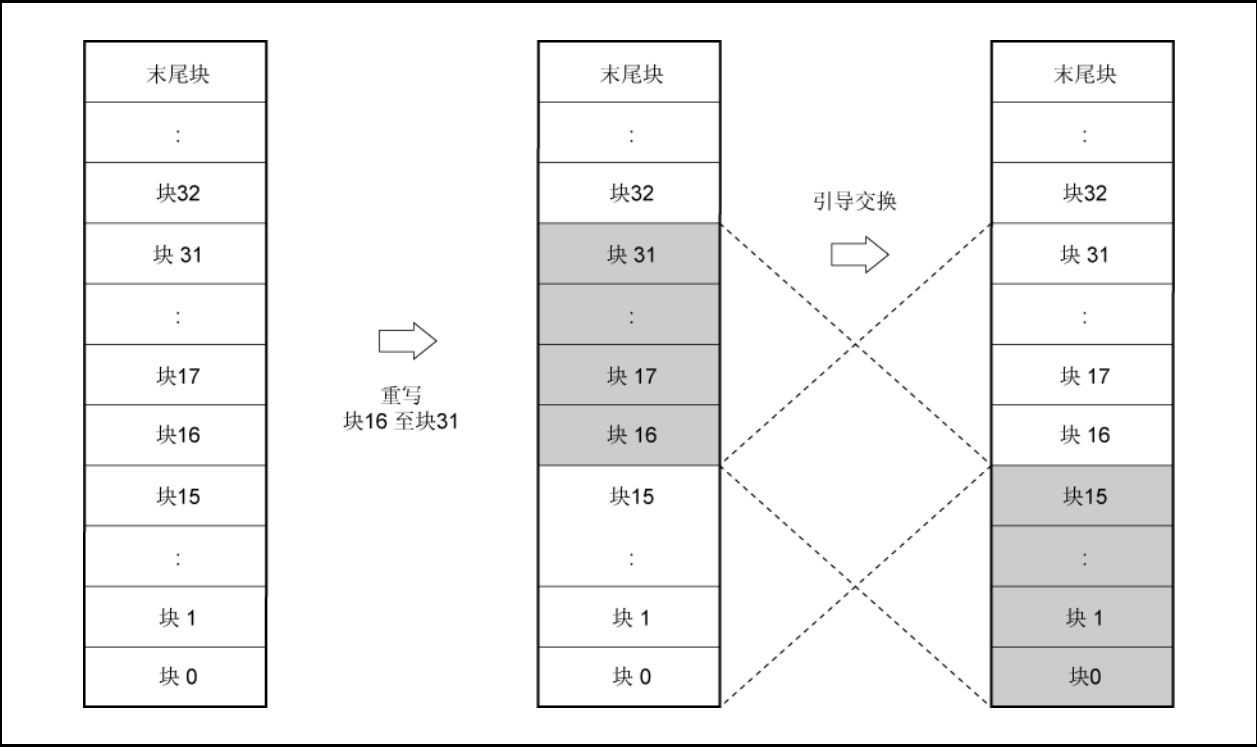


27.5.2 特性

(1) 安全自编程（引导交换功能）

V850ES/JJ3 支持引导交换功能，可将物理存储器的块 0 至 15 与块 16 至 31 互换。预先写入起始程序，该程序将要被写入块 16 至块 31，然后交换物理存储器，即使在重写期间发生断电，整个区域也可以安全地重新写入，因为正确用户程序始终存在于块 0 至块 15 中。

图 27-16. 重写整个存储区（引导交换）



(2) 中断支持

自编程期间，不能从 flash 存储器提取指令。因此，即使发生一个中断，写入到 flash 存储器的用户处理程序也不能使用。

因此，为了在V850ES/JJ3 自编程期间使用中断，处理转移到内部RAM的指定地址^注。安排一条跳转指令，将程序处理转移到内部RAM中指定地址^注上的用户中断服务程序。

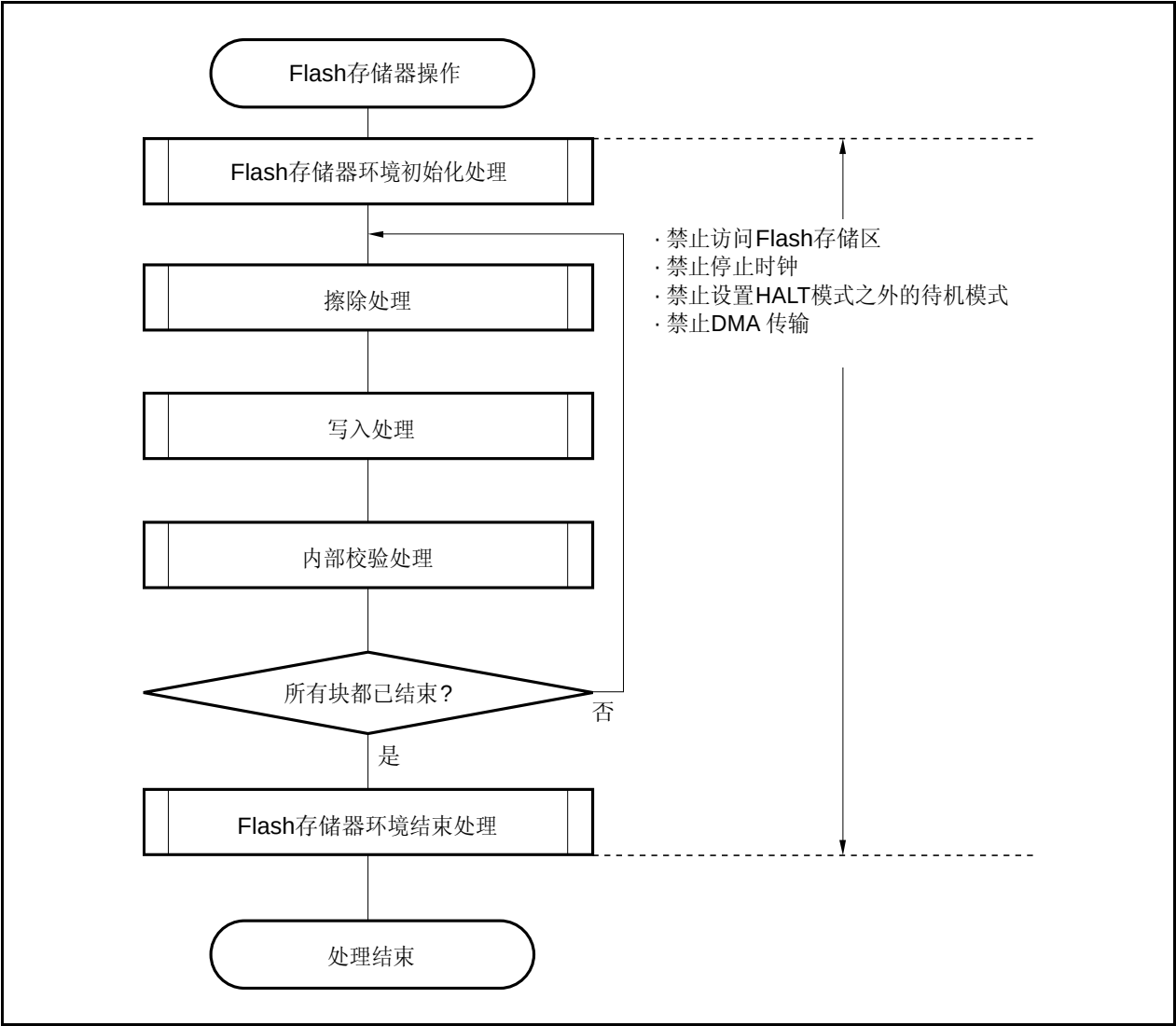
注 NMI 中断： 内部 RAM 的起始地址。
 可屏蔽中断： 内部 RAM 起始地址 + 4 的地址。

27.5.3 标准自编程流程

通过 flash 自编程对 flash 存储器重写的整个过程图示如下。

<R>

图 27-17. 标准自编程流程



27.5.4 Flash 功能

<R>

表 27-10. Flash 功能列表

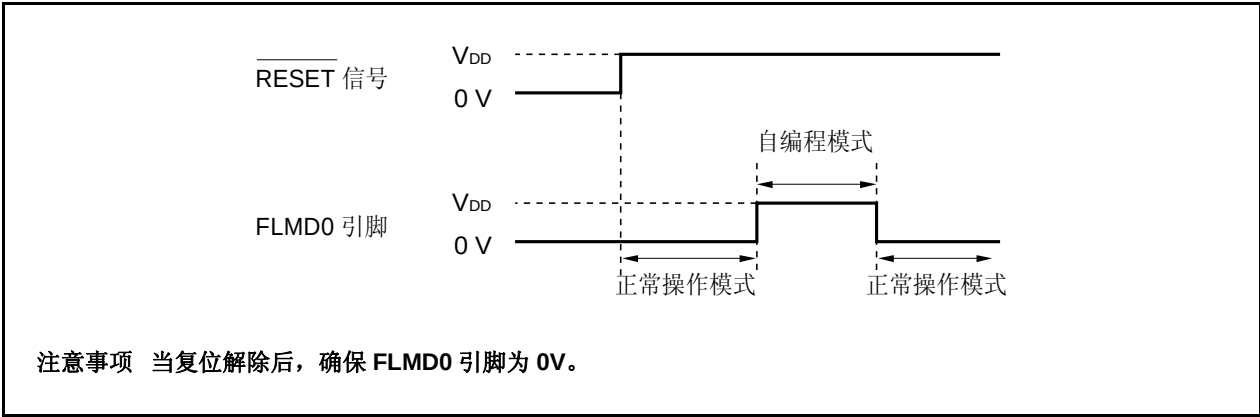
功能名称	概述	支持
FlashInit	自编程库初始化	√
FlashEnv	Flash 环境启动/停止	(
FlashFLMDCheck	FLMD 引脚检查	(
FlashStatusCheck	硬件处理执行状态检查	(
FlashBlockErase	块擦除	(
FlashWordWrite	数据写入	(
FlashBlockIVerify	块的内部校验	(
FlashBlockBlankCheck	块空白检查	√
FlashSetInfo	Flash 存储器信息设置	√
FlashGetInfo	Flash 存储器信息获取	√
FlashBootSwap	执行引导交换	√

27.5.5 引脚处理

(1) FLMD0 引脚

FLMD0 引脚用于在复位解除后设置操作模式，并在自重写期间保护 flash 存储器不被写入。因此有必要在复位解除后和执行正常操作时保持 FLMD0 引脚的电压为 0V。在重新写入存储器之前，通过端口控制自编程模式期间有电平为 V_{DD} 的电压输入 FLMD0 引脚。
当完成自编程时，FLMD0 引脚电压值必须返回 0V。

图 27-18. 模式变换时序



27.5.6 用到的内部资源

下表列出了自编程需要使用的内部资源。这些内部资源也可以自由地用于除了自编程之外的用途。

<R>

表 27-11. 用到的内部资源

资源名称	说明
堆栈区 ^注	用户使用堆栈的扩展，被库使用（内部 RAM 和外部 RAM 中都可以使用）。
库代码 ^注	库的程序实体（可以用于被操作的 flash 存储器模块之外的任何空间）。
应用程序	作为用户应用程序执行。 调用 flash 存储器功能。
可屏蔽中断	可在用户应用程序的执行状态或自编程状态下使用。因为处理过程转移到内部 RAM 起始地址+4 的地址，为了在自编程状态中使用该中断，预先安排一条跳转指令，将程序处理转移到内部 RAM 起始地址+4 的地址上的用户中断服务程序。
NMI 中断	可在用户应用程序执行状态或自编程状态下使用。因为处理过程被转移到内部 RAM 起始地址，为了在自编程状态时使用该中断，预先安排一条跳转指令，将程序处理转移到内部 RAM 起始地址上的用户中断服务程序。

注 关于用到的资源，参见 Flash 存储器自编程库用户手册。

第二十八章 片上调试功能

V850ES/JJ3 单片机的片上调试功能可由下面两种方法实现。

- 使用 DCU （调试控制单元）
V850ES/JJ3 中，使用 $\overline{\text{DRST}}$ 、DCK、DMS、DDI 和 DDO 引脚作为调试接口引脚来实现片上调试功能。
- 不使用 DCU
通过 MINICUBE2 或类似的设备来实现片上调试功能，使用用户资源来替代 DCU。

下表显示了两种片上调试功能的特性。

表 28-1. 片上调试功能特性

		使用 DCU 调试	不使用 DCU 调试
调试接口引脚		$\overline{\text{DRST}}$ 、DCK、DMS、DDI、DDO	<ul style="list-style-type: none"> • 当使用 UARTA0 时 RXD0, TXD0 • 使用 CSIB0 时 SIB0、SOB0、$\overline{\text{SCKB0}}$、HS (PCM0) • 使用 CSIB3 时 SIB3、SOB3、$\overline{\text{SCKB3}}$、HS (PCM0)
用户资源保障		不需要	必需
硬件中断功能		2 点	2 点
软件中断功能	4 点	4 点	4 点
	2000 点	2000 点	2000 点
实时RAM 监控功能 ^{注1}		可用	可用
动态内存修改（DMM）功能 ^{注2}		可用	可用
屏蔽功能		$\overline{\text{Reset}}$ 、 $\overline{\text{NMI}}$ 、 $\overline{\text{INTWDT2}}$ 、 $\overline{\text{HLDRQ}}$ 、 $\overline{\text{WAIT}}$	$\overline{\text{RESET}}$ 引脚
ROM 安全功能		10-字节 ID 码鉴定	10-字节 ID 码鉴定
使用硬件		MINICUBE 等.	MINICUBE2 等
跟踪功能		不支持	不支持
调试中断接口功能（DBINT）		不支持	不支持

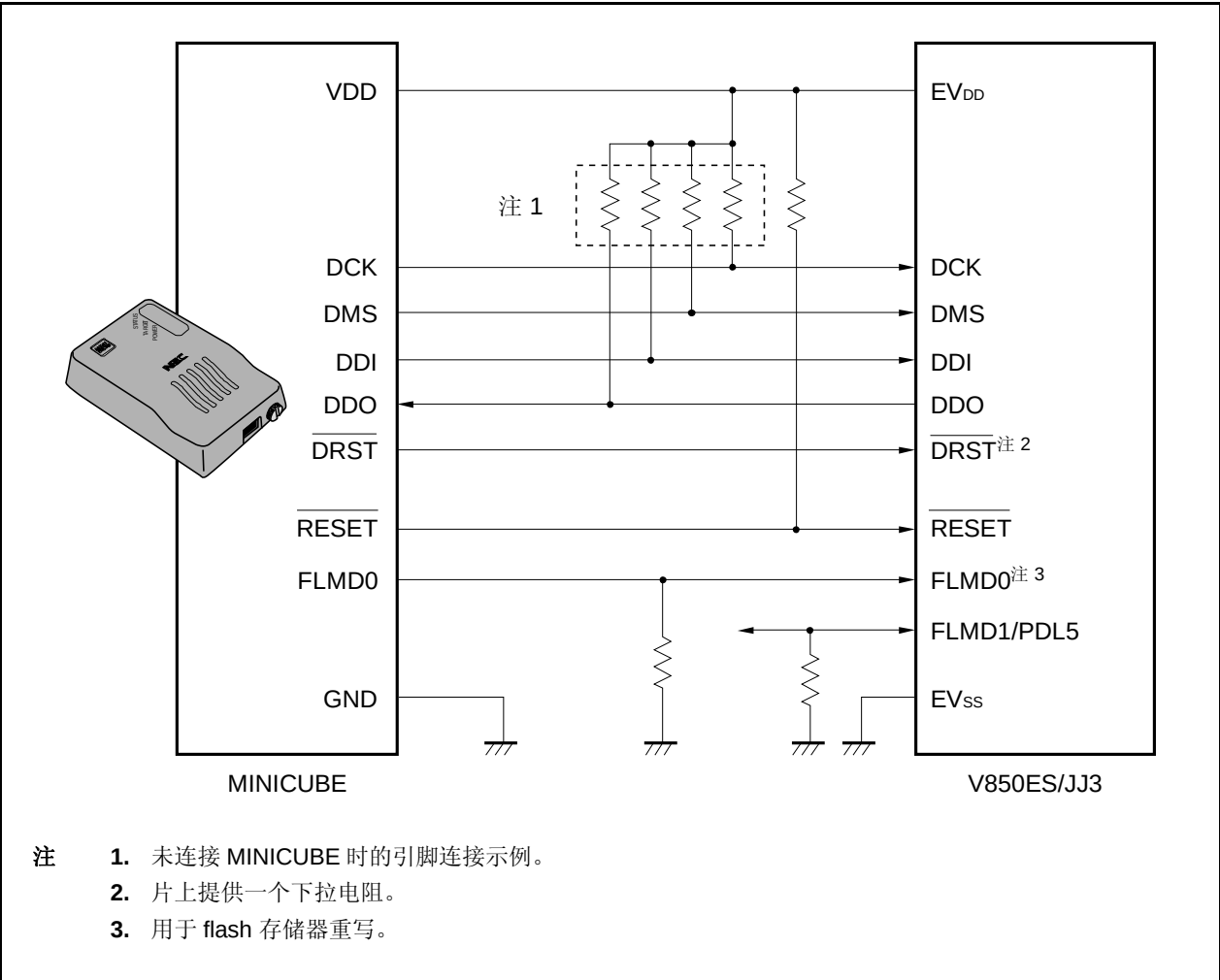
- 注
1. 这是在程序执行时读出存储器内容的功能。
 2. 这是在程序执行时重写 RAM 内容的功能。

28.1 用 DCU 调试

通过调试接口引脚（ $\overline{\text{DRST}}$, DCK, DMS, DDI 和 DDO）连接片上调试仿真器（MINICUBE）来调试程序。

28.1.1 连接电路示例

图 28-1. 调试接口用作通信接口时的电路连接示例



28.1.2 接口信号

接口信号的描述如下：

(1) $\overline{\text{DRST}}$

它是片上调试单元的复位输入信号。它是负逻辑信号，异步初始化调试单元。

集成调试器启动后，当 MINICUBE 检测到目标系统的 V_{DD} 时使得 $\overline{\text{DRST}}$ 上升，并且启动设备上的片上调试单元。

当 $\overline{\text{DRST}}$ 信号上升为高后，CPU 产生一个复位信号。

在启动集成调试器开始调试后，总是会产生一个 CPU 复位。

(2) DCK

它是时钟输入信号。由 MINICUBE 为其提供 20 MHz 或 10 MHz 的时钟。在片上调试单元，在 DCK 信号的上升沿对 DMS 和 DDI 信号进行采样，而在其下降沿输出数据 DDO。

(3) DMS

它是传输模式选择信号。调试单元的传输状态随 DMS 信号的电平改变。

(4) DDI

这是数据输入信号。在片上调试单元内在 DCK 的上升沿对其进行采样。

(5) DDO

这是数据输出信号。在 DCK 信号的下降沿由片上调试单元输出该信号。

(6) EV_{DD}

该信号用于检测目标系统的 VDD。如果没有检测到目标系统的 VDD，则由 MINICUBE 输出该信号 ($\overline{\text{DRST}}$, DCK, DMS, DDI, FLMD0 和 $\overline{\text{RESET}}$)，并进入高阻状态。

(7) FLMD0

flash 存储器的自编程功能用于通过集成调试器把数据下载到 flash 存储器。在 flash 存储器自编程期间，FLMD0 引脚必须保持为高。此外，在 FLMD0 引脚接一个下拉电阻。

FLMD0 引脚可通过下面两种方式进行控制：

<1> 由 MINICUBE 控制：

将 MINICUBE 的 FLMD0 信号连接到 FLMD0 引脚。

在正常模式下，MINICUBE 不作任何驱动（高阻）。

在中断期间，当集成调试器执行下载功能时 MINICUBE 将 FLMD0 引脚上升为高电平。

<2> 由端口控制：

将设备的任一端口连接到 FLMD0 引脚。

可能会用到由用户程序实现 flash 存储器自编程功能的同一个端口。

在集成调试器的控制端，在执行下载功能之前，对端口引脚进行设置使端口引脚为高电平，或在执行下载功能之后，降低端口引脚的电平。

详情参见 ID850QB Ver.3.10 集成调试器操作用户手册（U17435E）。

(8) $\overline{\text{RESET}}$

它是系统复位输入引脚。如果通过用户程序设置 OCDM 寄存器 OCDM0 位的值，使得 $\overline{\text{DRST}}$ 引脚无效，那么将无法执行片上调试。因此，通过 $\overline{\text{RESET}}$ 引脚，由 MINICUBE 引发复位，可以使得 $\overline{\text{DRST}}$ 引脚生效（初始化）。

28.1.3 可屏蔽功能

Reset、NMI、INTWDT2、 $\overline{\text{WAIT}}$ 和 $\overline{\text{HLDRQ}}$ 信号都可以被屏蔽。
调试器（ID850QB）的可屏蔽功能以及相应的 V850ES/JJ3 功能列举如下：

表 28-2. 可屏蔽功能

ID850QB 的可屏蔽功能	相应的 V850ES/JJ3 功能
NMI0	NMI 引脚输入
NMI2	产生不可屏蔽中断请求信号（INTWDT2）
STOP	—
HOLD	$\overline{\text{HLDRQ}}$ 引脚输入
RESET	由 $\overline{\text{RESET}}$ 引脚输入、低电压检测器、时钟监控或看门狗定时器（WDT2）溢出，产生复位信号
WAIT	$\overline{\text{WAIT}}$ 引脚输入

28.1.4 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。该寄存器是一个专用寄存器，只能通过一个特定序列的组合来写入（参见 3.4.7 专用寄存器）。该寄存器也用于指定支持片上调试功能的引脚是用作片上调试引脚，还是作为普通的端口/外围功能引脚。它也可以用来断开 P05/INTP2/ $\overline{\text{DRST}}$ 引脚的内部下拉电阻。
只有在 $\overline{\text{DRST}}$ 引脚输入低电平时，才能够写入 OCDM 寄存器。
该寄存器可按字节或按位进行读写。

复位后：01H^注 R/W 地址：FFFFF9FCH

	7	6	5	4	3	2	1	<0>
OCDM	0	0	0	0	0	0	0	OCDM0

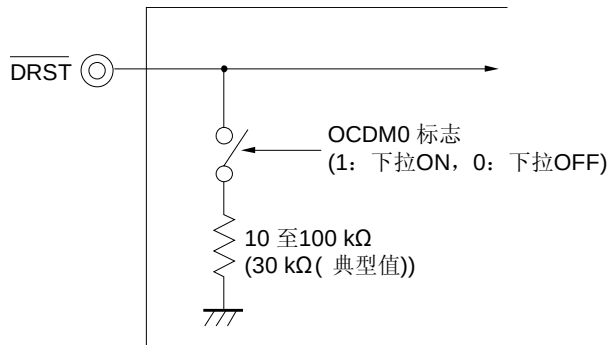
OCDM0	操作模式
0	选择正常操作模式（可以被复用为片上调试功能引脚的引脚用作端口/外设功能引脚）并且断开P05/INTP2/DRST引脚的片上下拉电阻。
1	当 $\overline{\text{DRST}}$ 引脚为低电平时： 正常操作模式(可以被复用功能为片上调试功能引脚的引脚用作端口/外设功能引脚) 当 $\overline{\text{DRST}}$ 引脚为高电平时： 片上调试模式 (可以被复用功能为片上调试功能引脚的引脚用作片上调试模式引脚)

注 $\overline{\text{RESET}}$ 输入将该寄存器设为 01H。由 WDT2RES 信号、时钟监控 (CLM) 或低电压检测器 (LVI) 引发复位后，OCDM 寄存器的值仍然保持。

注意事项 1. 外部复位后，当 DDI、DDO、DCK 以及 DMS 引脚不作为片上调试引脚而作为端口引脚时，必须执行下列所有操作。

- 给 P05/INTP2/DRST 引脚输入一个低电平。
- 将 OCDM0 位置位。这种情况下，进行下列操作。
 - <1> 将 OCDM0 位清 0。
 - <2> 将 P05/INTP2/DRST 引脚固定为低电平，直到<1>完成。

2. DRST 引脚有一个下拉电阻。当 OCDM0 标志被清 0 时，断开该电阻。



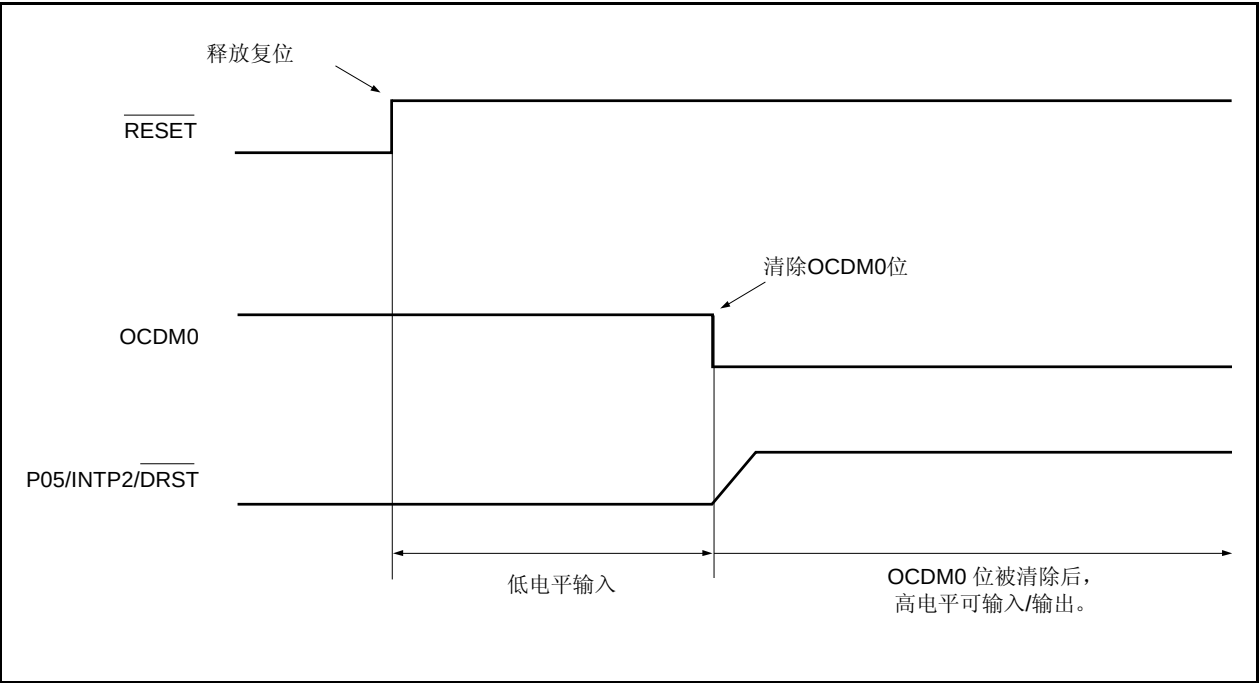
28.1.5 操作

在下表条件时，片上调试功能无效。
当不使用该功能时，在 OCDM.OCDM0 标志清 0 之前， $\overline{\text{DRST}}$ 引脚一直保持为低电平。

<div>OCDM0 标志</div> <div>$\overline{\text{DRST}}$ 引脚</div>	0	1
L	无效	无效
H	无效	有效

备注 L: 低电平输入
 H: 高电平输入

图 28-2. 不使用片上调试功能时的时序



28.1.6 注意事项

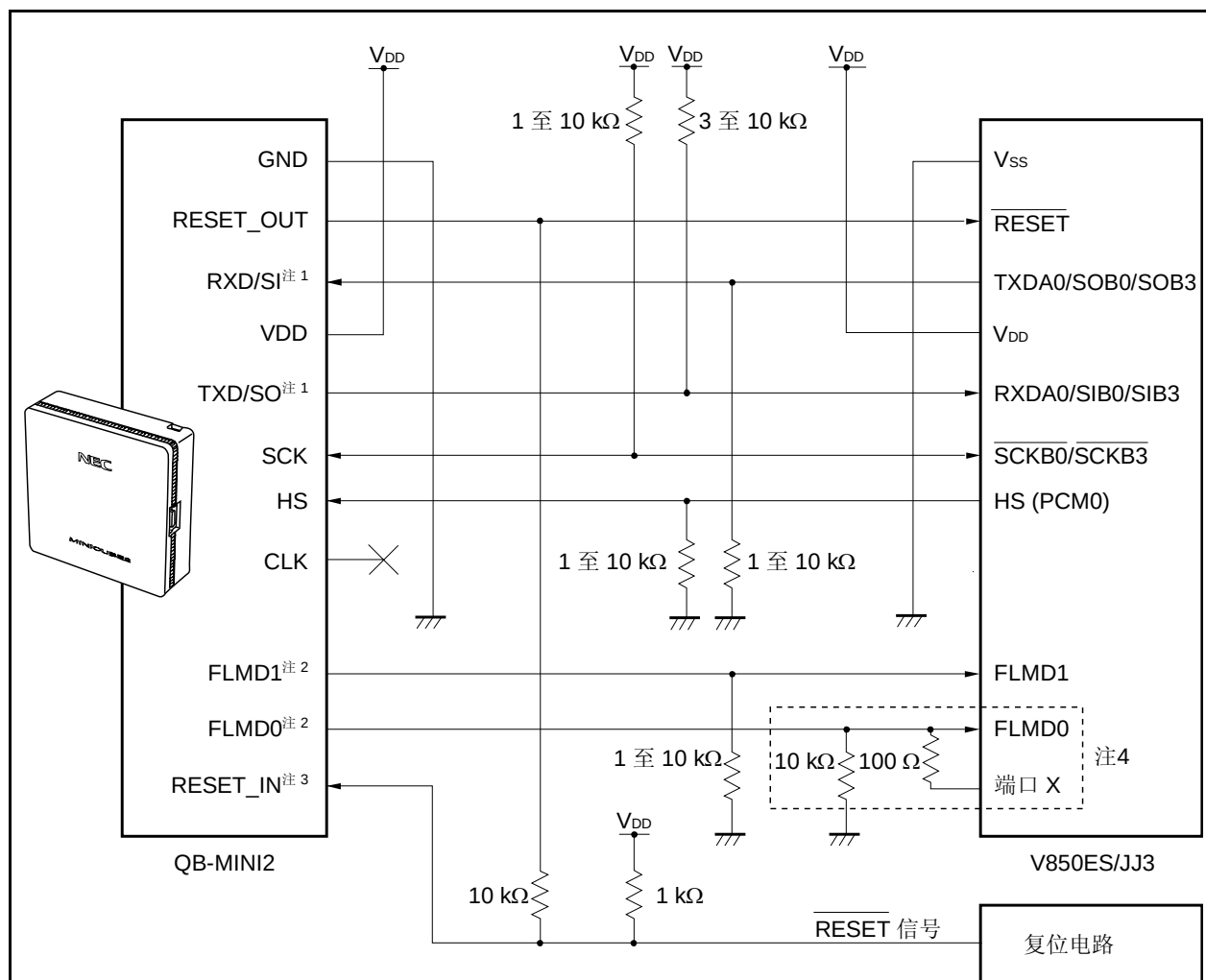
- (1) 如果在 RUN（程序执行）期间输入一个复位信号（来自目标系统或来自内部复位源的复位信号），则断点功能可能会误操作。
- (2) 即使复位信号被屏蔽功能所屏蔽，如果在引脚输入一个复位信号也可能使得 I/O 缓存器（端口引脚）复位。
- (3) 中断时 reset 引脚被屏蔽，且 CPU 和外围 I/O 不复位。如果在执行用户程序时，flash 存储器由 DMM 重写或由 RAM 监控功能读取之后立刻产生引脚复位或内部复位，那么 CPU 和外围 I/O 可能无法正确复位。
- (4) 在片上调试模式下，DDO 引脚被强制设为高电平输出。

28.2 不使用 DCU 调试

下面描述如何不用 DCU 而使用 MINICUBE2 来实现片上调试功能。使用 MINICUBE2 的 UARTA0 (RXDA0 和 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

28.2.1 电路连接示例

图 28-3. UARTA0/CSIB0/CSIB3 用作通信接口时的电路连接示例



- 注
1. 连接 V850ES/JJ3 的 TXDA0/SOB0/SOB3（发送端）和目标连接器的 RXD/SI（接收端），还要将目标连接器的 TXD/SO（发送端）连接到 V850ES/JJ3 的 RXDA0/SIB0/SIB3（接收端）。
 2. 除了存储器在调试的中断时被重写，连接到这个引脚（FLMD0，FLMD1）的 V850ES/JJ3 引脚可用作复用功能引脚，因为该引脚处于高阻状态。
 3. 该连接设计是假设 RESET 信号由 N 通道漏极开路缓存器输出（输出电阻：100Ω或更小）。
 4. 虚线内的电路是为 flash 存储器自编程设计的，通过端口控制 FLMD0 引脚。使用该端口输入或输出高电平。当不执行 flash 存储器自编程时，接到 FLMD0 引脚的下拉电阻可以在 1 kΩ至 10 kΩ之间

备注 当 UARTA0、CSIB0 或 CSIB3 用作通信接口时，涉及的引脚参见表 28-3。

表 28-3. V850ES/JJ3 和 MINICUBE2 之间的接线

MINICUBE2 (QB-MINI2) 的引脚配置			使用 CSIB0-HS		使用 CSIB3-HS		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
SI/RxD	输入	该引脚用于接收自 V850ES/JJ3 的指令和数据	P41/SOB0	23	P911/SOB3	54	P30/TXDA0	25
SO/TxD	输出	该引脚用于发送指令和数据至 V850ES/JJ3	P40/SIB0	22	P910/SIB3	53	P31/RXDA0	26
SCK	输出	该引脚用于 3-线串行通信的时钟输出	P42/SCKB0	24	P912/SCKB3	55	不需要	—
CLK	输出	到 V850ES/JJ3 的时钟输出引脚	不需要	—	不需要	—	不需要	—
RESET_OUT	输出	到 V850ES/JJ3 的复位输出引脚	RESET	14	RESET	14	RESET	14
FLMD0	输出	该输出引脚用于将 V850ES/JJ3 设置为调试模式或编程模式	FLMD0	8	FLMD0	8	FLMD0	8
FLMD1	输出	该输出引脚用于设置编程模式	PDL5/FLMD1	110	PDL5/FLMD1	76	PDL5/FLMD1	110
HS	输入	CSIO + HS 通信的握手信号	PCM0/WAIT	85	PCM0/WAIT	61	不需要	—
GND	—	接地	V _{SS}	11	V _{SS}	11	V _{SS}	11
			AV _{SS}	2	AV _{SS}	2	AV _{SS}	2
			EV _{SS}	33, 103	EV _{SS}	33, 69	EV _{SS}	33, 103
RESET_IN	输入	目标系统的复位输入						

28.2.2 可屏蔽功能

只有复位信号可以被屏蔽。

调试器 (ID850QB) 的可屏蔽功能以及相应的 V850ES/JJ3 的功能列举如下：

表 28-4. 可屏蔽功能

ID850QB 的可屏蔽功能	相应的 V850ES/JJ3 功能
NMI0	—
NMI1	—
NMI2	—
STOP	—
HOLD	—
RESET	由 RESET 引脚输入产生复位信号
WAIT	—

28.2.3 用户资源的保留

用户必须准备下列资源来实现 MINICUBE2 和目标器件间的通信功能，并完成各调试功能。这些内容需要在用户程序中设置或使用编译器选项设置。

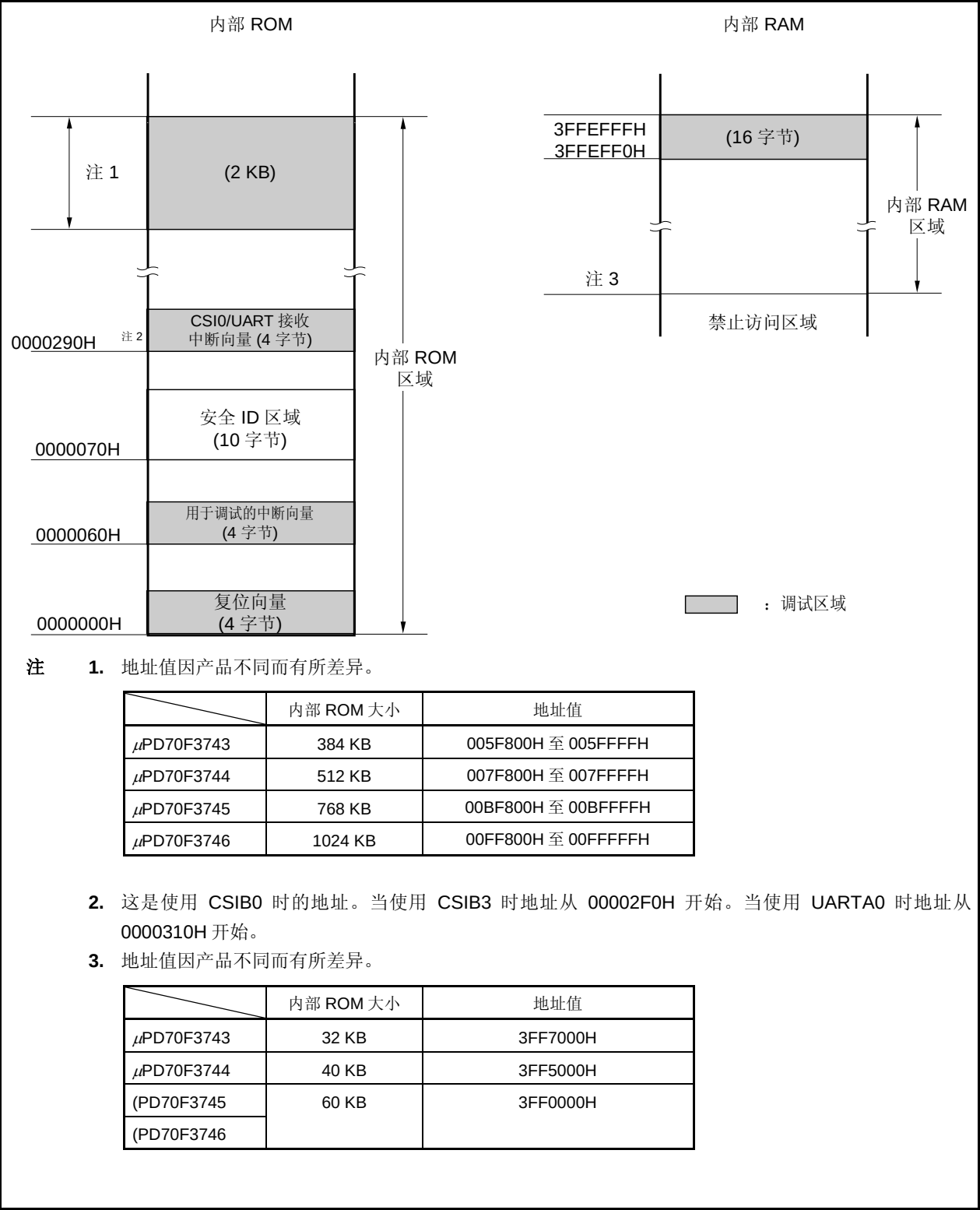
（1）存储器空间的保留

图 28-4 中的阴影部分是预留区，用于放置调试监控程序，因此用户程序和数据不能分配在这些空间。必须保证这些空间不被用户程序使用。

（2）安全 ID 设置

ID 码必须嵌入到图 28-4 中 0000070H 至 0000079H 的区域，以防止存储内容被未经授权读取。详情参见 **28.3 ROM 安全功能**。

图 28-4. 调试监控程序的内存空间分配



(3) 复位向量

复位向量包括调试监控程序所需的转移指令。

[如何保留区域]

没有必要刻意保护该区域。但是，当下载一个程序时，调试器根据下列情况重写复位向量。若重写的模式与下列情况不符，则调试器产生一个错误（使用 ID850QB 时为 F0C34）。

(a) 当从 0 地址开始连续放置两个空指令时

重写前		重写后
0x0 nop	→	跳转到 0x0 处的调试监控程序
0x2 nop		0x4 xxxx
0x4 xxxx		

(b) 当从 0 地址开始连续放置两个 0xFFFF 时（已擦除器件）

重写前		重写后
0x0 0xFFFF	→	跳转到 0x0 处的调试监控程序
0x2 0xFFFF		0x4 xxxx
0x4 xxxx		

(c) jr 指令放在 0 地址（使用 CA850 时）

重写前		重写后
0x0 jr disp22	→	跳转到 0x0 处的调试监控程序
		0x4 jr disp22 - 4

(d) 从 0 地址开始连续放置 mov32 和 jmp（使用 IAR 编译 ICCV850 时）

重写前		重写后
0x0 mov imm32,reg1	→	跳转到 0x0 处的调试监控程序
0x6 jmp [reg1]		0x4 mov imm32,reg1
		0xa jmp [reg1]

(e) 在 0 地址处放置用于跳转到调试监控程序的指令

重写前		重写后
跳转到 0x0 处的调试监控程序	→	没有改变

(4) 调试监控程序的保留区域

图 28-4 中的阴影部分是分配给调试监控程序的区域。监控程序对调试通信接口和 CPU 的 RUN 或中断处理过程执行初始化。内部 ROM 区必须填充为 0xFF。该区域禁止被用户程序重写。

[如何保留区域]

若用户程序不使用该区域，则没有必要进行保护。

但是，为了避免在调试器启动时可能出现的问题，建议预先使用编译器保留本区域。

下面给出保留区域的示例，使用 NEC 电子的编译器 CA850。加上汇编源文件和连接伪指令代码，如下所示。

- 汇编源文件（添加下列代码作为汇编源文件。）

```
-- Secures 2 KB space for monitor ROM section
.section "MonitorROM", const
.space    0x800, 0xff

-- Secures interrupt vector for debugging
.section "DBG0"
.space   4, 0xff

-- Secures interrupt vector for serial communication
-- Change the section name according to the serial communication mode used
.section "INTCB0R"
.space   4, 0xff

-- Secures 16-byte space for monitor RAM section
.section "MonitorRAM", bss
.lcomm  monitorramsym, 16, 4    -- defines symbol monitorramsym
```

- 连接伪指令（将下列代码添加到连接伪指令文件）

下面给出一个例子，它的内部 ROM 为 1024 KB（结束地址为 00FFFFFFH）且内部 RAM 为 60 KB（结束地址为 3FFEFFFH）。

```
MROMSEG      : !LOAD ?R V0x0ff800{
               MonitorROM    = $PROGBITS    ?A MonitorROM;
};

MRAMSEG      : !LOAD ?RW V0x03ffeff0{
               MonitorRAM    = $NOBITS      ?AW MonitorRAM;
};
```

(5) 通信串行接口的保留

UARTA0, CSIB0 或 CSIB3 用于 MINICUBE2 和目标系统之间的通信。与串行接口模式相关的设置通过调试监控程序来完成，但如果设置被用户程序改变，则可能发生通信错误。
为了防止发生这样的问题，必须在用户程序中对通信串行接口进行保留。

[如何保留通信串行接口]

- 片上调试模式寄存器（OCDM）
使用 UARTA0, CSIB0 或 CSIB3 的片上调试功能时，设置 OCDM 寄存器功能为正常模式。务必按照下面步骤进行设置。
 - P05/INTP2/DRST 引脚输入低电平。
 - 将 OCDM0 位设置如下：
 - <1> 将 OCDM0 位清 0。
 - <2> 在<1>完成之前将 P05/INTP2/DRST 引脚输入固定为低电平。
- 串行接口寄存器
在用户程序中不要设置 CSIB0, CSIB3 或 UARTA0 相关的寄存器。
- 中断屏蔽寄存器
使用 CSIB0 时，不要屏蔽发送结束中断（INTCB0R）。使用 CSIB3 时，不要屏蔽发送结束中断（INTCB3R）。使用 UARTA0 时，不要屏蔽接收结束中断（INTUA0R）。

(a) 使用 CSIB0 时								
	7	6	5	4	3	2	1	0
CB0RIC	×	0	×	×	×	×	×	×

(b) 使用 CSIB3 时								
	7	6	5	4	3	2	1	0
CB3RIC	×	0	×	×	×	×	×	×

(c) 使用 UARTA0 时								
	7	6	5	4	3	2	1	0
UA0RIC	×	0	×	×	×	×	×	×

备注

×

无须理会

• 使用 UARTA0 时的端口寄存器

当使用 UARTA0 时，通过调试监控程序对端口寄存器进行设置，使 TXDA0 和 RXDA0 引脚有效。在调试期间用户程序不要改变下列寄存器的设置。（同样的值可被覆盖写入）

	7	6	5	4	3	2	1	0
PFC3	×	×	×	×	×	×	0	0
	7	6	5	4	3	2	1	0
PMC3L	×	×	×	×	×	×	1	1
备注	×：无需理会							

• 使用 CSIB0 的端口寄存器

使用 CSIB0 时，通过调试监控程序对端口寄存器进行设置，使 SIB0、SOB0、 $\overline{\text{SCKB0}}$ 和 HS（PMC0）引脚生效。在调试期间用户程序不要改变下列寄存器的设置。（同样的值可被覆盖写入）

(a) SIB0、SOB0 和 $\overline{\text{SCKB0}}$ 设置								
	7	6	5	4	3	2	1	0
PMC4	×	×	×	×	×	1	1	1
	7	6	5	4	3	2	1	0
PFC4	×	×	×	×	×	×	0	0
(b) HS（PMC0 引脚）设置								
	7	6	5	4	3	2	1	0
PMCM	×	×	×	×	×	×	×	0
	7	6	5	4	3	2	1	0
PCM	×	×	×	×	×	×	×	注
注	禁止写入该位。 对应 HS 引脚的端口值根据调试器的状态由监控程序改变。为了按照字节进行端口寄存器的设置，用户程序通常会使用读取—修改—写入方法。但是，如果在写入之前发生一个调试中断，就可能执行一个意外的操作。							
备注	×：无需理会							

- 使用 CSIB3 的端口寄存器

使用 CSIB3 时，通过调试监控程序对端口寄存器进行设置，使 SIB3、SOB3、 $\overline{\text{SCKB3}}$ 和 HS（PMC0）引脚生效。在调试期间用户程序不要改变下列寄存器的设置。（同样的值可被覆盖写入）

(a) SIB3、SOB3 和 $\overline{\text{SCKB3}}$ 设置

	7	6	5	4	3	2	1	0
PMC9H	x	x	x	1	1	1	x	x

	7	6	5	4	3	2	1	0
PFC9H	x	x	x	1	1	1	x	x

(b) HS（PMC0 引脚）设置

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

	7	6	5	4	3	2	1	0
PCM	x	x	x	x	x	x	x	注

注 禁止写入该位。

对应 HS 引脚的端口值根据调试器的状态由监控程序改变。为了按照字节进行端口寄存器的设置，用户程序通常会使用读取—修改—写入方法。但是，如果在写入之前发生一个调试中断，就可能会执行一个意外的操作。

备注 x：无需理会

28.2.4 注意事项

(1) 被调试设备的处理

不要将用于调试的设备安装在批量生产的产品上，因为在调试过程中 flash 存储器被重写过，所以不能保证 flash 存储器的重写次数。此外，不要将调试监控程序嵌入到批量生产的产品中。

(2) 当中断不能执行时

如果满足下列条件之一，则强制断点无法执行。

- 中断被禁止（DI）。
- 发给串行接口发出的中断被屏蔽，该中断用于 MINICUBE2 和目标器件之间的通信。
- 在通过可屏蔽中断释放待机时，禁止进入待机模式。
- MINICUBE2 和目标器件之间的通信模式为 UARTA0，且主时钟已被停止。

(3) 当不执行伪实时 RAM 监控 (RRM) 功能和 DMM 功能时

如果满足下列条件之一，则伪 RRM 功能和 DMM 功能无法操作。

- 中断禁止 (DI)。
- 串行接口发出的中断被屏蔽，该中断用于 MINICUBE2 和目标器件之间的通信。
- 禁止在通过可屏蔽中断解除待机时进入待机模式。
- MINICUBE2 和目标器件之间的通信模式为 UARTA0，且主时钟已被停止。
- MINICUBE2 和目标器件之间的通信模式为 UARTA0，并且用于通信的时钟和调试器中指定的时钟不同。

(4) 允许由伪 RRM 和 DMM 功能解除待机模式。

如果满足下列条件之一，则可以由伪 RRM 功能和 DMM 功能来解除待机模式。

- MINICUBE2 和目标器件之间的通信模式为 CSIB0 或 CSIB3。
- MINICUBE2 和目标器件之间的通信模式为 UARTA0，且提供主时钟。

(5) 使用 DMM 功能写入需要特定序列写入的外围 I/O 寄存器。

需要特定序列的外围 I/O 寄存器，不能用 DMM 功能写入。

(6) Flash 自编程

如果分配给调试监控程序的空间被 flash 存储器自编程重写，那么调试器将不能再正常运行。

28.3 ROM 安全功能

28.3.1 安全 ID

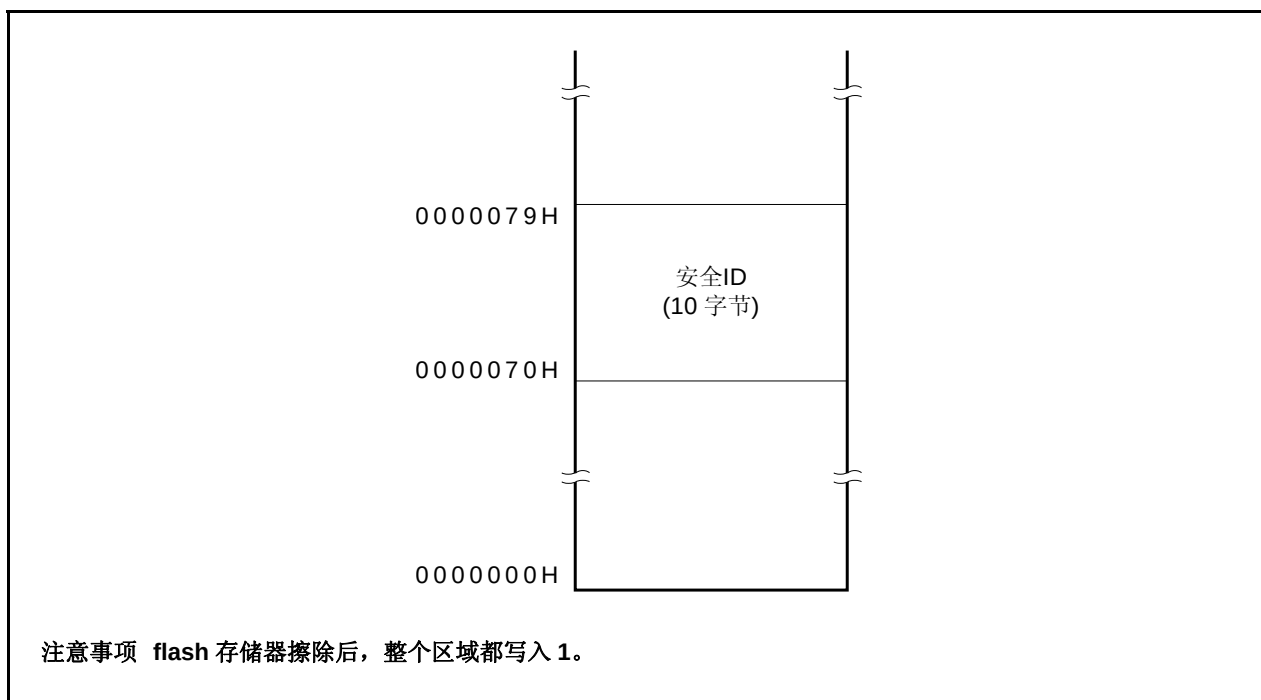
V850ES/JJ3 的 flash 存储器版本执行鉴定，通过使用一个 10 字节的 ID 码，防止 flash 存储器中的内容在由片上仿真器进行片上调试时被未经授权读取。

在 10 字节的片上 flash 存储区地址从 0000070H 至 0000079H 设置 ID 码，以便允许调试器执行 ID 鉴定。

如果 ID 匹配，则安全保护解除，且允许对 flash 存储器进行读取，也允许使用片上调试仿真器。

- 在 0000070H 至 0000079H 之间设置 10 字节的 ID 码。
- 0000079H 的第 7 位是片上调试仿真器的使能标志。
(0: 禁用, 1: 使能)
- 当启动片上调试仿真器后，调试器要求输入 ID。当输入的 ID 和 0000070H 至 0000079H 间设置的 ID 码匹配时，调试器启动。
- 即使 ID 码匹配，当片上调试仿真器的使能标志为 0 时，也不能进行调试。

图 28-5. 安全 ID 区



28.3.2 设置

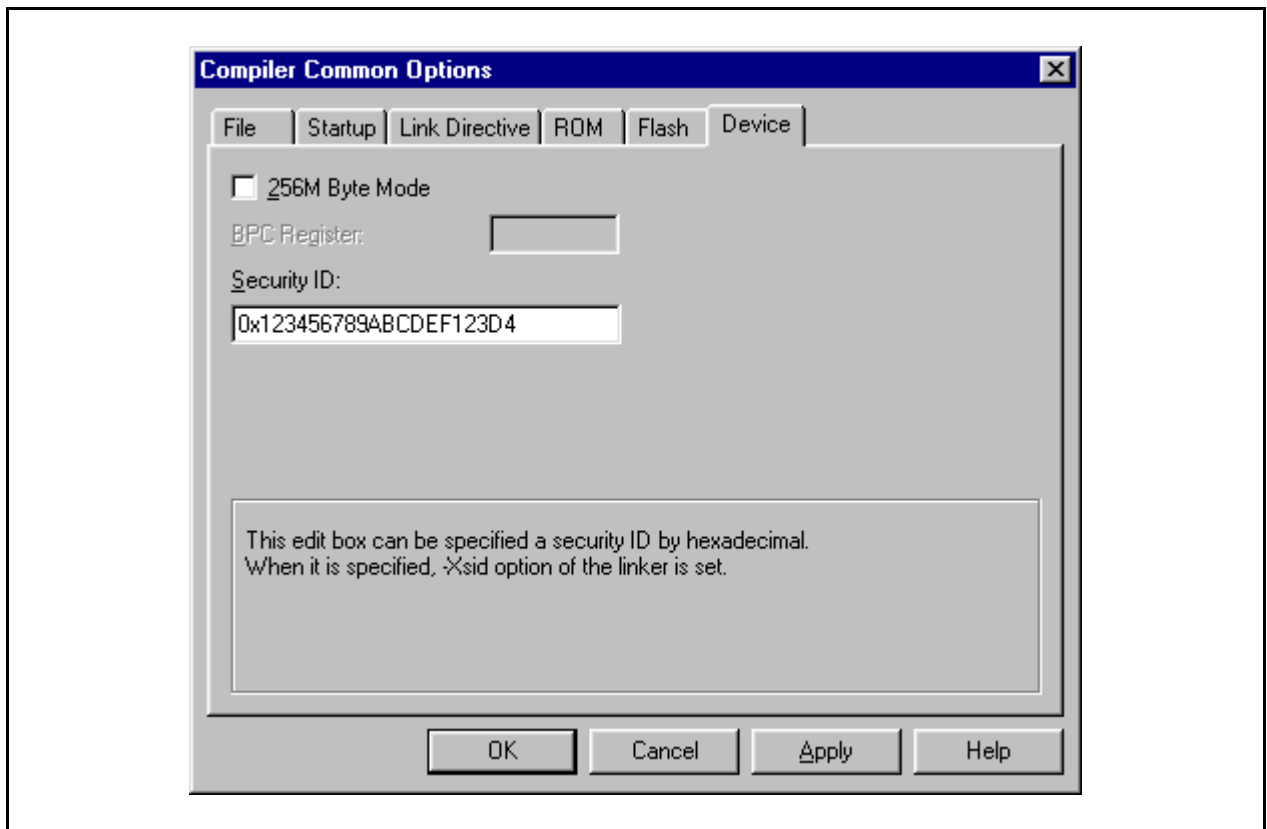
下面在表 28-5 中显示如何设置 ID 码。

当 ID 码按照表 28-5 进行设置时，在 ID850QB 的配置对话框中的 ID 码输入为 “123456789ABCDEF123D4”（ID 码不区分大小写）。

表 28-5. 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

可以为支持 CA850 Ver. 3.10 或更新版本的设备文件指定 ID 码，并且使用 PM+编译器的通用选项设置来指定安全 ID。



[程序示例（当使用 CA850 Ver. 3.10 或更高版本）]

```
#-----  
#      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

将上述程序示例添加到启动文件中。

第二十九章 电气规范

最大额定值（= 25° C）（1/2）

参数	符号	条件	额定值范围	单位
供应电压	V _{DD}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	−0.5至+4.6	V
	EV _{DD}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	−0.5至+4.6	V
	AV _{REF0}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	−0.5至+4.6	V
	AV _{REF1}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	−0.5至+4.6	V
	V _{SS}	V _{SS} = EV _{SS} = AV _{SS}	−0.5至+0.5	V
	AV _{SS}	V _{SS} = EV _{SS} = AV _{SS}	−0.5至+0.5	V
	EV _{SS}	V _{SS} = EV _{SS} = AV _{SS}	−0.5至+0.5	V
输入电压	V _{I1}	RESET, FLMD0, PCD0至PCD3, PCM0至PCM5, PCS0至PCS7, PCT0至PCT7, PDH0至PDH7, PDL0至PDL15	−0.5至EV _{DD} + 0.5 ^{#1}	V
	V _{I2}	P10, P11	−0.5至AV _{REF1} + 0.5 ^{#1}	V
	V _{I3}	X1, X2	−0.5至V _{RO} ^{#2} + 0.5 ^{#1}	V
	V _{I4}	P00至P06, P30至P39, P40至P42, P50至P55, P60至P615, P80, P81, P90至P915	−0.5至+6.0	V
	V _{I5}	XT1, XT2	−0.5至V _{DD} + 0.5 ^{#1}	V
模拟输入电压	V _{IAN}	P70至P715	−0.5至AV _{REF0} + 0.5 ^{#1}	V

- 注
1. 请确保所有供电电压都不超过最大额定值（最大值）。
 2. 片上稳压器输出电压（2.5 V（典型值））。

备注 除非另外说明，复用功能引脚的特性与端口引脚的特性相同。

最大额定值 ($T_A = 25^\circ\text{C}$) (2/2)

参数	符号	条件		额定值范围	单位
输出电流，低	I _{OL}	P00至P06, P30至P39, P40至P42, P50至P55, P60至P615, P80, P81, P90至P915	每个引脚	4	mA
			全部引脚总和	50	mA
		PCD0至PCD3, PCM0至PCM5, PCS0至PCS7, PCT0至PCT7, PDH0至PDH7, PDL0至PDL15	每个引脚	4	mA
			全部引脚总和	50	mA
		P10, P11	每个引脚	4	mA
			全部引脚总和	8	mA
		P70至P715	每个引脚	4	mA
			全部引脚总和	20	mA
输出电流，高	I _{OH}	P00至P06, P30至P39, P40至P42, P50至P55, P60至P615, P80, P81, P90至P915	每个引脚	−4	mA
			全部引脚总和	−50	mA
		PCD0至PCD3, PCM0至PCM5, PCS0至PCS7, PCT0至PCT7, PDH0至PDH7, PDL0至PDL15	每个引脚	−4	mA
			全部引脚总和	−50	mA
		P10, P11	每个引脚	−4	mA
			全部引脚总和	−8	mA
		P70至P715	每个引脚	−4	mA
			全部引脚总和	−20	mA
工作环境温度	T _A			−40至+85	°C
储存温度	T _{stg}			−40至+125	°C

注意事项 1. 不要将芯片的输出（或输入/输出）引脚直接互连，或直接连接至 V_{DD} 、 V_{CC} 和 GND 。但是，漏极开路或集电极开路引脚可以直接互连。

如果输出引脚能够设置为高阻状态，并且外部电路的输出时序设计时已经考虑避免输出冲突，芯片和外部电路之间的输出引脚才可以直接相连。

- 2.** 任何参数，哪怕瞬间超过最大额定值，都可能使产品性能受损。就是说，产品工作在最大额定值时处于物理损坏的边缘，所以，产品必须在不超出最大额定值的条件下使用。DC 和 AC 特性的额定值和工作条件代表在正常工作期间的性能保障范围。

备注 除非另外说明，复用功能引脚的特性与端口引脚的特性相同。

电容 (TA = -40 至 +85°C, VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = AVSS = 0 V)

参数	符号	条件	最小值	典型值	最大值	单位
I/O 电容	Cio	fx = 1 MHz 未测量的引脚返回至0 V			10	pF

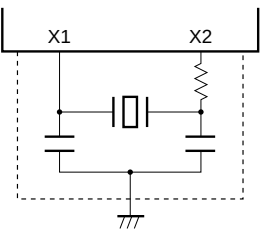
工作条件

(TA = -40 至+85°C, VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = AVSS = 0 V)

内部系统时钟频率	条件	供应电压			单位
		VDD	EVDD	AVREF0, AVREF1	
fxx = 2.5 至 32 MHz	C = 4.7 μF (推荐值), A/D 转换器停止, D/A 转换器停止	2.85至3.6	2.85至3.6	2.85至3.6	V
	C = 4.7 μF (推荐值), A/D 转换器工作, D/A 转换器工作	3.0至3.6	3.0至3.6	3.0至3.6	V
fxt = 32.768 kHz	C = 4.7 μF (推荐值), A/D 转换器停止, D/A 转换器停止	2.85至3.6	2.85至3.6	2.85至3.6	V

<R> 主时钟振荡器特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2$ 至 3.6 V , $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

谐振器	电路举例	参数	条件	最小值	典型值	最大值	单位
陶瓷谐振器 /晶体谐振器		振荡频率 (f_x) ^{注1}		2.5		10	MHz
		振荡稳定时间 ^{注2}	复位解除后		$2^{16}/f_x$		s
			STOP模式解除后	$1^{\text{注4}}$	注 3		ms
			IDLE2 模式解除后	$350^{\text{注4}}$	注 3		μs

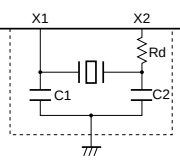
- 注
- 以上所示的振荡频率仅表示振荡器特性。使用 V850ES/JJ3, 内部工作条件不要超出 **AC 特性**和 **DC 特性**的额定值。
 - 从振荡开始到振荡电路稳定所需的时间。
 - 该值由 **OSTS** 寄存器的设置决定。
 - flash 存储器需要的建立时间, 使用 **OSTS** 寄存器来确保建立时间。

注意事项 在使用主时钟振荡器时, 上图中虚线框部分的连线应遵从如下的连接方式, 避免线间电容的负面影响。

- 连线要保证尽可能短。
 - 振荡器连线不要和其它的信号线互相交叉。
 - 布线时不要将振荡器连线靠近通过大波动电流的信号线
 - 保持振荡器电容的地电位和 V_{SS} 相同。
 - 不要将电容的接地点连接在大电流通过的地模式。
 - 不要从振荡器取信号。
- 当主时钟停止, 设备运行于副时钟时, 在由副时钟切换回主时钟之前, 应该由程序来实现振荡稳定时间的等待。
 - 谐振器的选择和振荡器常数, 客户需要自行评估或向谐振器生产商申请进行评估。

<R>

(i) KYOCERA KINSEKI CORPORATION: 晶体振荡器 ($T_A = -10$ 至 $+70^{\circ}\text{C}$)

类型	电路举例	零件登记号	振荡频率 f_x (MHz)	推荐电路常数			振荡电压范围	
				C1 (pF)	C2 (pF)	Rd (Ω)	MIN. (V)	MAX. (V)
表面贴装		CX49GFWB04000D0PESZZ	4.000	8	8	0	2.2	3.6
		CX49GFWB05000D0PESZZ	5.000	8	8	0	2.2	3.6
		CX49GFWB06000D0PESZZ	6.000	8	8	0	2.2	3.6
		CX49GFWB08000D0PESZZ	8.000	8	8	0	2.2	3.6
		CX49GFWB10000D0PESZZ	10.000	8	8	0	2.2	3.6

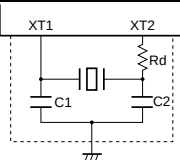
注意事项 该振荡常数是振荡器生产商在特定环境下进行评估得出的参考值。

如果在实际应用中必须对振荡器特性进行优化, 请向振荡器生产商申请对实现电路的评估。

振荡电压和振荡频率仅表示振荡器特性。使用 V850ES/JJ3, 内部工作条件不要超出 AC 特性和 DC 特性的额定值。

<R>

(ii) Murata Mfg. Co. Ltd.: 陶瓷振荡器 ($T_A = -20$ 至 $+80^{\circ}\text{C}$)

类型	电路举例	零件登记号	振荡频率 f_x (MHz)	推荐电路常数			振荡电压范围	
				C1 (pF)	C2 (pF)	Rd (Ω)	MIN. (V)	MAX. (V)
表面贴装		CSTCC2M50G56-R0	2.500	(47)	(47)	3300	2.85	3.6
		CSTCR4M00G55-R0	4.000	(39)	(39)	1500	2.85	3.6
		CSTCR5M00G53-R0	5.000	(15)	(15)	1500	2.85	3.6
		CSTCR6M00G53-R0	6.000	(15)	(15)	1500	2.85	3.6
		CSTCE8M00G55-R0	8.000	(33)	(33)	330	2.85	3.6
		CSTCE10M0G52-R0	10.000	(10)	(10)	470	2.85	3.6
引线式		CSTLS4M00G56-B0	4.000	(47)	(47)	1500	2.85	3.6
		CSTLS5M00G53-B0	5.000	(15)	(15)	1500	2.85	3.6
		CSTLS6M00G53-B0	6.000	(15)	(15)	1500	2.85	3.6
		CSTLS8M00G53-B0	8.000	(15)	(15)	680	2.85	3.6
		CSTLS10M0G53-B0	10.000	(15)	(15)	680	2.85	3.6

注意事项 该振荡常数是振荡器生产商在特定环境下进行评估得出的参考值。

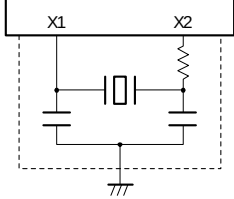
如果在实际应用中必须对振荡器特性进行优化, 请向振荡器生产商申请对实现电路的评估。

振荡电压和振荡频率仅表示振荡器特性。使用 V850ES/JJ3, 内部工作条件不要超出 AC 特性和 DC 特性的额定值。

备注 图中 C1 和 C2 列的括号中的数值表示包含在振荡器中的电容。

副时钟振荡器特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

谐振器	电路举例	参数	条件	最小值	典型值	最大值	单位
晶体谐振器		振荡频率 (f_{XT}) ^{注 1}		32	32.768	35	kHz
		振荡稳定时间 ^{注 2}				10	s

注 1. 以上所示的振荡频率仅表示振荡器特性。使用 V850ES/JJ3，内部工作条件不要超出 **AC 特性**和 **DC 特性**的额定值。

2. 从 V_{DD} 达到振荡电压范围（2.85 V（最小值））起，到晶体振荡器稳定所需的时间。

注意事项 1. 在使用副时钟振荡器时，上图中虚线框部分的连线应遵从如下的连接方式，避免线间电容的负面影响。

- 连线要保证尽可能短。
- 振荡器连线不要和其它的信号线互相交叉。
- 布线时不要将振荡器连线靠近通过大波动电流的信号线
- 保持振荡器电容的地电位和 V_{SS} 相同。
- 不要将电容的接地点连接在大电流通过的地模式。
- 不要从振荡器取信号。

2. 为了降低功耗，副时钟振荡器被设计为低振幅电路，而比起主时钟振荡器更容易受由于噪声引起的干扰。所以，在使用副时钟时需要特别注意布线的方式。

3. 有关谐振器的选择和振荡器常量，需要用户自行评估振荡特性或者向谐振器生产商申请对谐振器的评估。

PLL 特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
输入频率	f_x	×4 模式	2.5		5	MHz
		×8 模式	2.5		4	MHz
输出频率	f_{xx}	×4 模式	10		20	MHz
		×8 模式	20		32	MHz
锁定时间	t_{PLL}	在 V_{DD} 达到2.85 V (最小值)之后			800	μs

内部振荡器特性

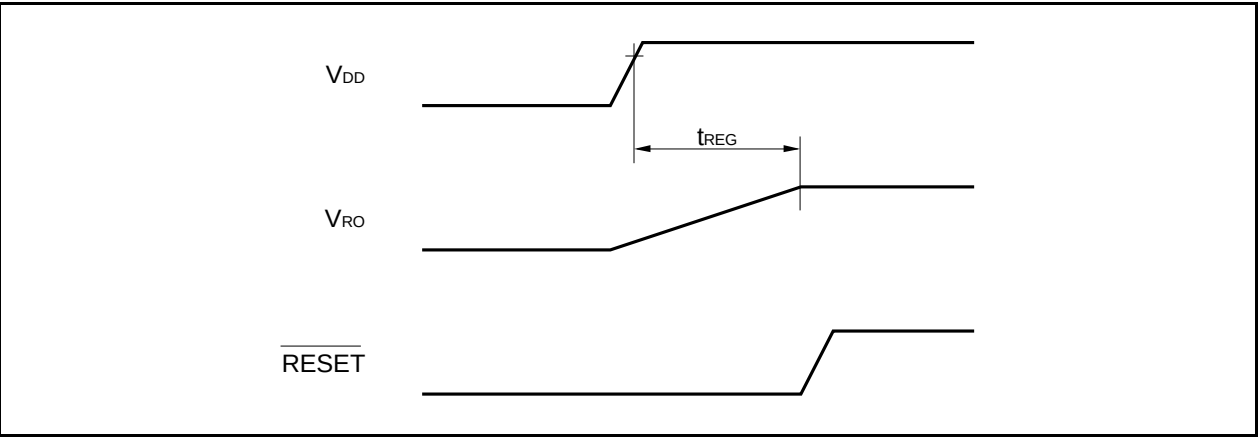
($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
输出频率	f_R		100	220	400	kHz

稳压器特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
输入电压	V_{DD}	$f_{xx} = 32\text{ MHz (MAX.)}$	2.85		3.6	V
输出电压	V_{RO}			2.5		V
稳压器输出稳定时间	t_{REG}	在 V_{DD} 达到2.85 V (最小值)之后, 稳定电容 $C = 4.7\text{ }\mu\text{F}$ (推荐值) 连接到 REGC 引脚			1	ms



DC 特性

(T_A = -40 至 +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V) (1/3)

参数	符号	条件	最小值	典型值	最大值	单位
输入电压, 高	V _{IH1}	RESET, FLMD0	0.8EV _{DD}		EV _{DD}	V
	V _{IH2}	P00至P06, P30至P37, P42, P50至P55, P60至P615, P80, P81, P92至P915	0.8EV _{DD}		5.5	V
	V _{IH3}	P38, P39, P40, P41, P90, P91	0.7EV _{DD}		5.5	V
	V _{IH4}	PCD0至PCD3, PCM0至PCM5, PCS0至PCS7, PCT0至PCT7, PDH0至PDH7, PDL0至PDL15	0.7EV _{DD}		EV _{DD}	V
	V _{IH5}	P70至P715	0.7AV _{REF0}		AV _{REF0}	V
	V _{IH6}	P10, P11	0.7AV _{REF1}		AV _{REF1}	V
输入电压, 低	V _{IL1}	RESET, FLMD0	EV _{SS}		0.2EV _{DD}	V
	V _{IL2}	P00至P06, P30至P37, P42, P50至P55, P60至P615, P80, P81, P92至P915	EV _{SS}		0.2EV _{DD}	V
	V _{IL3}	P38, P39, P40, P41, P90, P91	EV _{SS}		0.3EV _{DD}	V
	V _{IL4}	PCD0至PCD3, PCM0至PCM5, PCS0至PCS7, PCT0至PCT7, PDH0至PDH7, PDL0至PDL15	EV _{SS}		0.3EV _{DD}	V
	V _{IL5}	P70至P715	AV _{SS}		0.3AV _{REF0}	V
	V _{IL6}	P10, P11	AV _{SS}		0.3AV _{REF1}	V
输入漏电电流, 高	I _{LIH}	V _I = V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}			5	μA
输入漏电电流, 低	I _{LIL}	V _I = 0 V			-5	μA
输出漏电电流, 高	I _{LOH}	V _O = V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}			5	μA
输出漏电电流, 低	I _{LOL}	V _O = 0 V			-5	μA

备注 除非另外说明, 复用功能引脚的特性与端口引脚的特性相同。

DC 特性

(T_A = -40 至 +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V) (2/3)

参数	符号	条件			最小值	典型值	最大值	单位
输出电压, 高	V _{OH1}	P00至P06, P30至P39, P40至P42, P50至P55, P60至P615, P80, P81, P90至P915	每个引脚 I _{OH} = -1.0 mA	全部引脚总和 -20 mA	EV _{DD} - 1.0		EV _{DD}	V
			每个引脚 I _{OH} = -100 μA	全部引脚总和 -6.0 mA	EV _{DD} - 0.5		EV _{DD}	V
	V _{OH2}	PCD0至PCD3, PCM0至PCM5, PCS0至PCS7, PCT0至PCT7, PDH0至PDH7, PDL0至PDL15	每个引脚 I _{OH} = -1.0 mA	全部引脚总和 -20 mA	EV _{DD} - 1.0		EV _{DD}	V
			每个引脚 I _{OH} = -100 μA	全部引脚总和 -5.0 mA	EV _{DD} - 0.5		EV _{DD}	V
	V _{OH3}	P70至P715	每个引脚 I _{OH} = -0.4 mA	全部引脚总和 -6.4 mA	AV _{REF0} - 1.0		AV _{REF0}	V
			每个引脚 I _{OH} = -100 μA	全部引脚总和 -1.6 mA	AV _{REF0} - 0.5		AV _{REF0}	V
	V _{OH4}	P10, P11	每个引脚 I _{OH} = -0.4 mA	全部引脚总和 -0.8 mA	AV _{REF1} - 1.0		AV _{REF1}	V
			每个引脚 I _{OH} = -100 μA	全部引脚总和 -0.2 mA	AV _{REF1} - 0.5		AV _{REF1}	V
输出电压, 低	V _{OL1}	P00至P06, P30至P37, P42, P50至P55, P60至P615, P80, P81, P92至P915	每个引脚 I _{OL} = 1.0 mA	全部引脚总和 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}	PCD0至PCD3, PCM0至PCM5, PCS0至PCS7, PCT0至PCT7, PDH0至PDH7, PDL0至PDL15	每个引脚 I _{OL} = 1.0 mA	全部引脚总和 20 mA	0		0.4	V
	V _{OL4}	P10, P11 P70至P715	每个引脚 I _{OL} = 0.4 mA	全部引脚总和 7.2 mA	0		0.4	V
软件下拉电阻 [※]	R ₁	P05	V _I = V _{DD}		10	20	100	kΩ

备注

1. 除非另外说明, 复用功能引脚的特性与端口引脚的特性相同。
2. 当单个引脚没有满足 I_{OH} 和 I_{OL} 条件但满足所有引脚的总值符合该条件时, 则仅该引脚不满足 DC 特性。

DC 特性

(T_A = -40 至 +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V) (3/3)

参数	符号	条件	最小值	典型值	最大值	单位
供应电流 ^注	I _{DD1}	正常工作	f _{xx} = 32 MHz (f _x = 4 MHz)	40	64	mA
			f _{xx} = 20 MHz (f _x = 5 MHz)	30	50	mA
	I _{DD2}	HALT 模式	f _{xx} = 32 MHz (f _x = 4 MHz)	27	45	mA
			f _{xx} = 20 MHz (f _x = 5 MHz)	19	30	mA
	I _{DD3}	IDLE1 模式	f _{xx} = 5 MHz (f _x = 5 MHz), PLL 关闭	0.9	2.4	mA
	I _{DD4}	IDLE2 模式	f _{xx} = 5 MHz (f _x = 5 MHz), PLL 关闭	0.3	0.8	mA
	I _{DD5}	副时钟工作模式	f _{XT} = 32.768 kHz, 主时钟停止, 内部振荡器停止	80	600	μA
	I _{DD6}	Sub-IDLE 模式	f _{XT} = 32.768 kHz, 主时钟停止, 内部振荡器停止	11	100	μA
	I _{DD7}	STOP 模式	副时钟停止, 内部振荡器停止	8	80	μA
			副时钟工作, 内部振荡器停止	11	90	μA
			副时钟停止, 内部振荡器工作	13	90	μA
	I _{DD8}	Flash存储器编程 模式	f _{xx} = 32 MHz (f _x = 4 MHz)	45	74	mA
			f _{xx} = 20 MHz (f _x = 5 MHz)	34	60	mA

注 V_{DD} 和 EV_{DD} 电流的总值。不包括通过输出缓冲器、A/D 转换器、D/A 转换器和片上下拉电阻的电流。

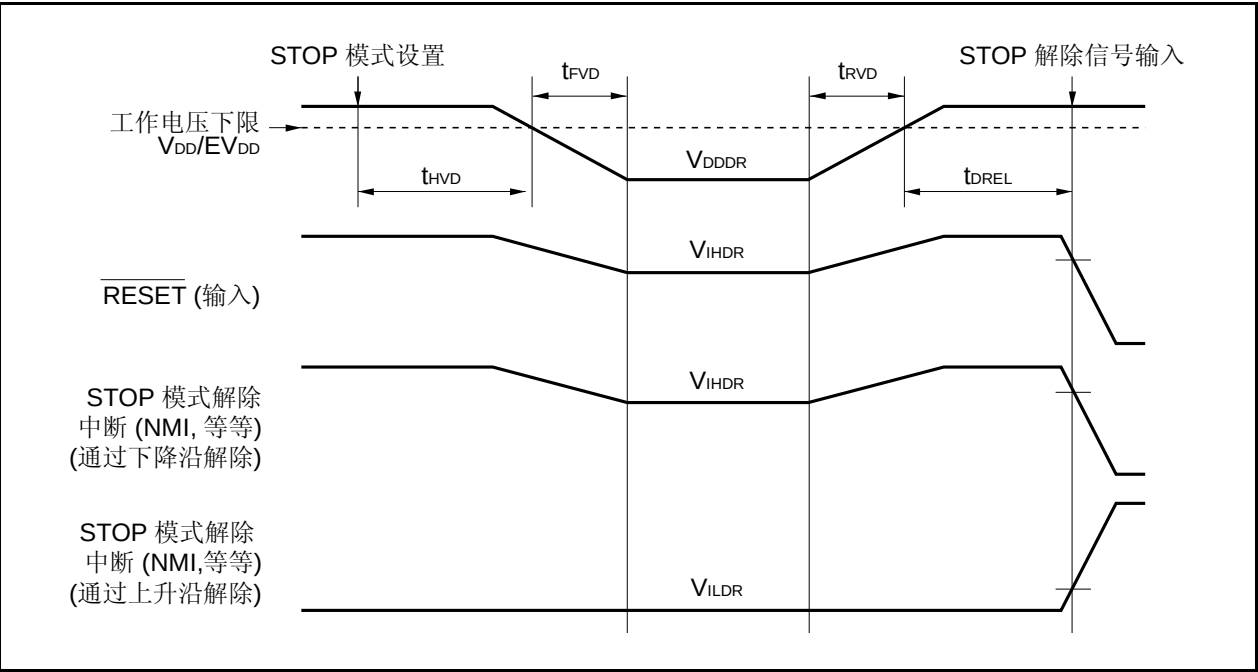
数据保持特性

STOP 模式下

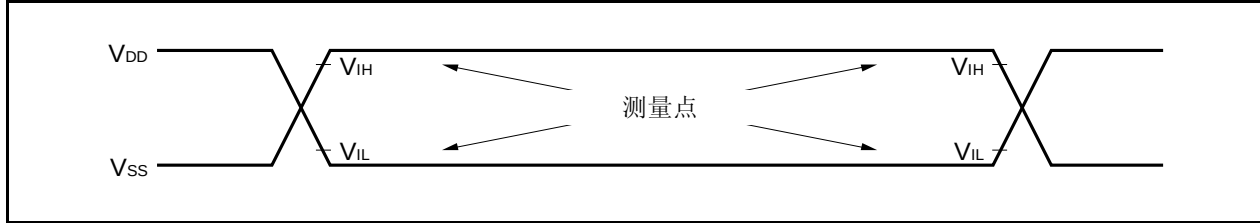
($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	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), $V_{DDDR} = 2.0\text{ V}$		8	80	μA
供应电压上升时间	t_{RVD}		200			μs
供应电压下降时间	t_{FVD}		200			μs
供应电压保持时间	t_{HVD}	设置STOP 模式后	0			ms
STOP 解除信号输入时间	t_{DREL}	V_{DD} 达到2.85 V (最小值) 之后	0			ms
数据保持输入电压, 高	V_{IHDR}	$V_{DD} = EV_{DD} = V_{DDDR}$	$0.9V_{DDDR}$		V_{DDDR}	V
数据保持输入电压, 低	V_{ILDR}	$V_{DD} = EV_{DD} = V_{DDDR}$	0		$0.1V_{DDDR}$	V

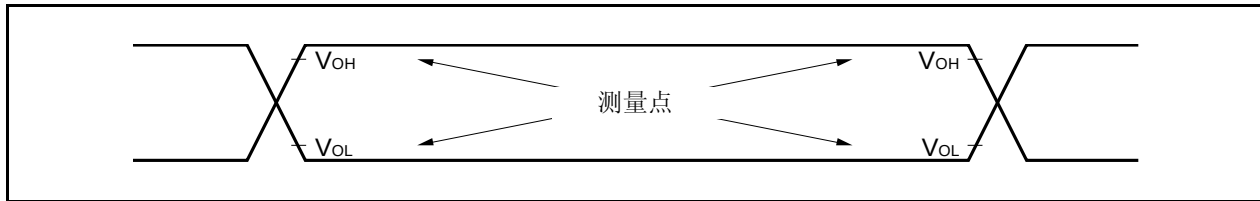
注意事项 转换到 STOP 模式以及从 STOP 模式中恢复都必须在额定工作范围内进行。



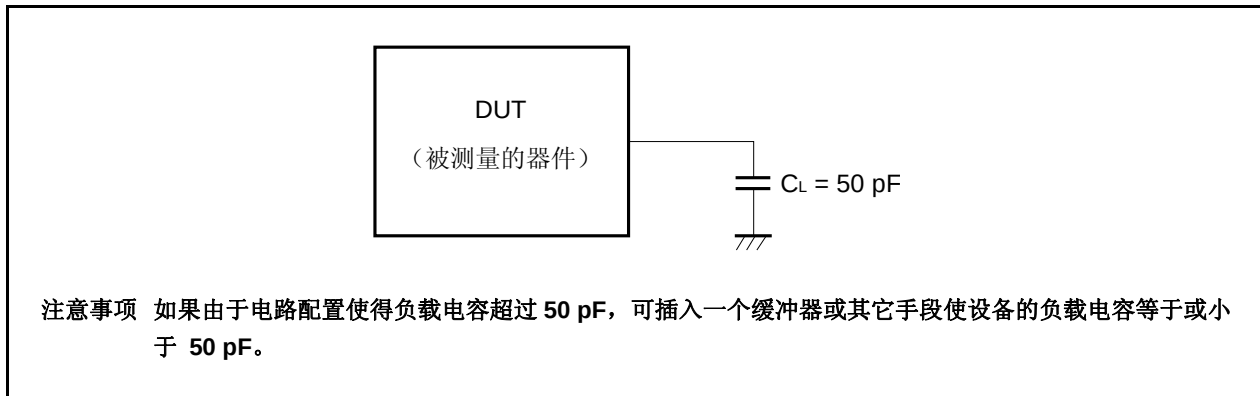
AC 特性

AC 测试输入测量点 (V_{DD} , AV_{REF0} , AV_{REF1} , EV_{DD})

AC 测试输出测量点



负载条件

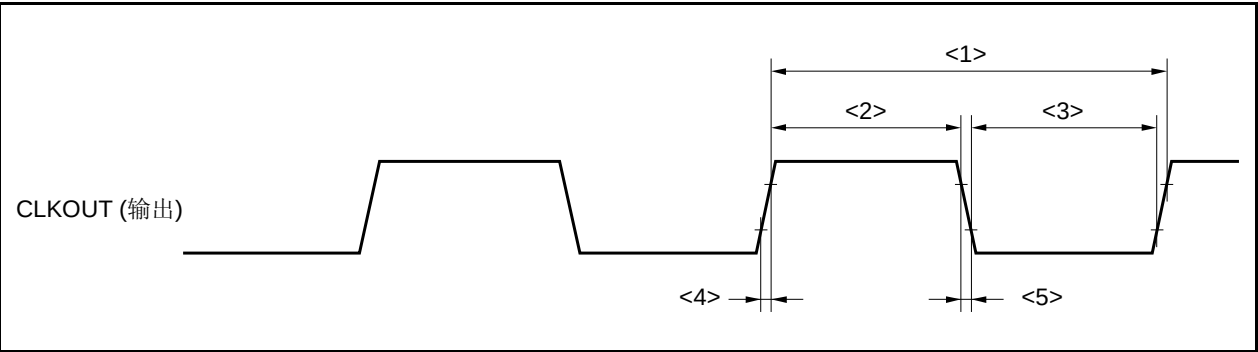


CLKOUT 输出时序

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号		条件	最小值	最大值	单位
输出周期	t_{CYK}	<1>		31.25 ns	31.25 μs	
高电平宽度	t_{WKH}	<2>		$t_{CYK}/2 - 6$		ns
低电平宽度	t_{WKL}	<3>		$t_{CYK}/2 - 6$		ns
上升时间	t_{KR}	<4>			6	ns
下降时间	t_{KF}	<5>			6	ns

时钟时序



总线时序

(1) 复用总线模式

注意事项 运行于 $f_{xx} > 20 \text{ MHz}$ 时，请确保插入地址保持等待和地址建立等待。

(a) 读/写周期 (CLKOUT 异步)

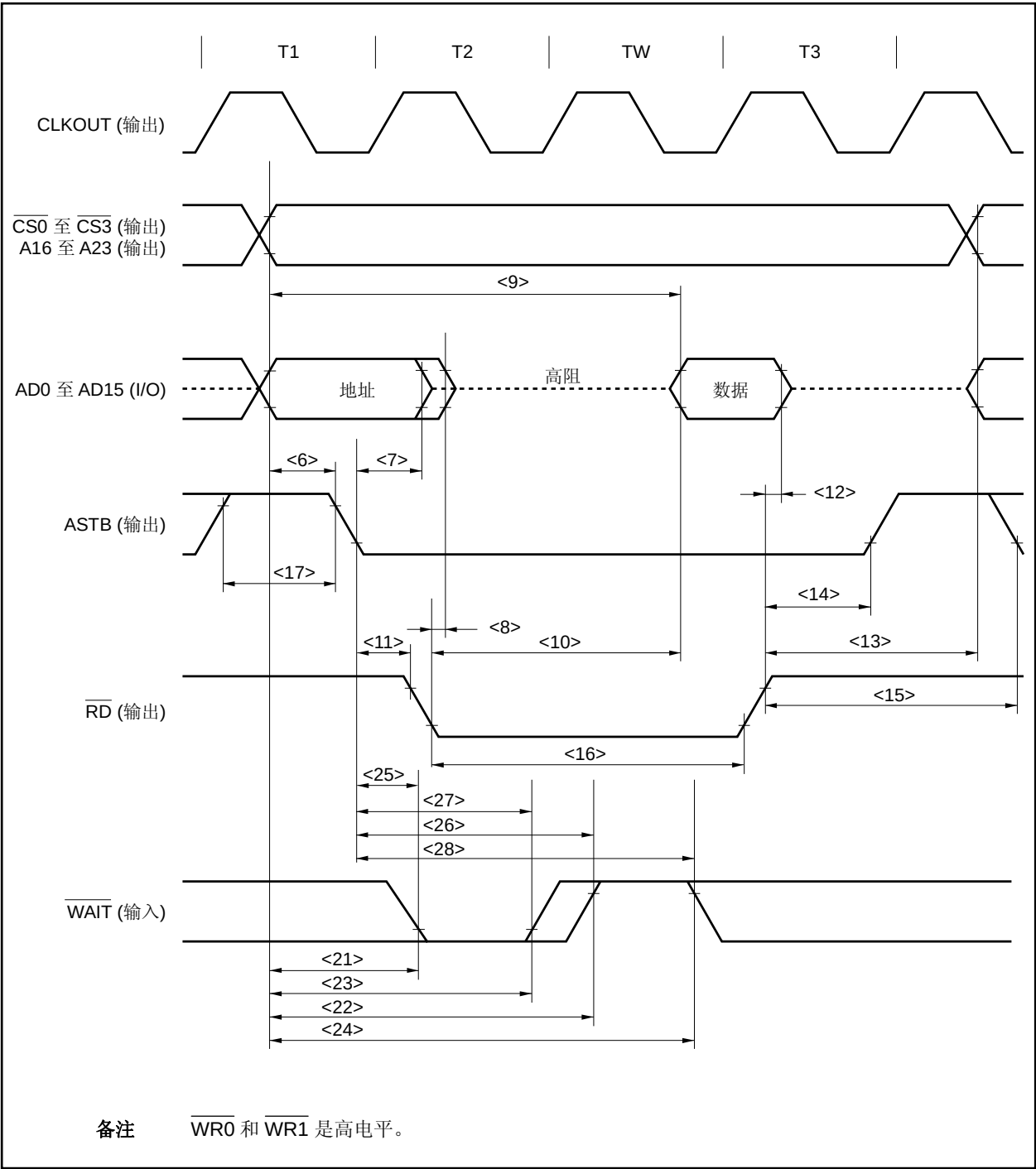
($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0 \text{ V}$, $C_L = 50 \text{ pF}$)

参数	符号	条件	最小值	最大值	单位
地址建立时间 (至 $\overline{ASTB}\downarrow$)	t_{SAST}	<6>	$(0.5 + t_{ASW})T - 20$		ns
地址保持时间 (从 $\overline{ASTB}\downarrow$ 起)	t_{HSTA}	<7>	$(0.5 + t_{AHW})T - 15$		ns
自 $\overline{RD}\downarrow$ 至地址悬空的延迟时间	t_{FRDA}	<8>		16	ns
自地址有效到数据输入建立时间	t_{SAID}	<9>		$(2 + n + t_{ASW} + t_{AHW})T - 35$	ns
自 $\overline{RD}\downarrow$ 到数据输入建立时间	t_{SRID}	<10>		$(1 + n)T - 25$	ns
自 $\overline{ASTB}\downarrow$ 至 $\overline{RD}$ 、 $\overline{WRm}\downarrow$ 的延迟时间	$t_{DSTRDWR}$	<11>	$(0.5 + t_{AHW})T - 15$		ns
数据输入保持时间 (自 $\overline{RD}\uparrow$ 起)	t_{HRDID}	<12>	0		ns
自 $\overline{RD}\uparrow$ 到地址输出时间	t_{DRDA}	<13>	$(1 + i)T - 15$		ns
自 $\overline{RD}$ 、 $\overline{WRm}\uparrow$ 到 $\overline{ASTB}\uparrow$ 的延迟时间	$t_{DRDWRST}$	<14>	$0.5T - 15$		ns
自 $\overline{RD}\uparrow$ 到 $\overline{ASTB}\downarrow$ 的延迟时间	t_{DRDST}	<15>	$(1.5 + i + t_{ASW})T - 15$		ns
$\overline{RD}$ 、 $\overline{WRm}$ 低电平宽度	t_{WRDWRL}	<16>	$(1 + n)T - 15$		ns
$\overline{ASTB}$ 高电平宽度	t_{WSTH}	<17>	$(1 + i + t_{ASW})T - 15$		ns
自 $\overline{WRm}\downarrow$ 起的数据输出时间	t_{DWROD}	<18>		15	ns
数据输出建立时间 (至 $\overline{WRm}\uparrow$)	t_{SODWR}	<19>	$(1 + n)T - 20$		ns
数据输出保持时间 (自 $\overline{WRm}\uparrow$ 起)	t_{HWROD}	<20>	$T - 15$		ns
$\overline{WAIT}$ 建立时间 (至地址有效)	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
$\overline{WAIT}$ 保持时间 (自地址有效起)	t_{HAWT1}	<23> $n \geq 1$	$(0.5 + n + t_{ASW} + t_{AHW})T$		ns
	t_{HAWT2}	<24>	$(1.5 + n + t_{ASW} + t_{AHW})T$		ns
$\overline{WAIT}$ 建立时间 (至 $\overline{ASTB}\downarrow$)	t_{SSTWT1}	<25> $n \geq 1$		$(1 + t_{AHW})T - 25$	ns
	t_{SSTWT2}	<26>		$(1 + n + t_{AHW})T - 25$	ns
$\overline{WAIT}$ 保持时间 (自 $\overline{ASTB}\downarrow$ 起)	t_{HSTWT1}	<27> $n \geq 1$	$(n + t_{AHW})T$		ns
	t_{HSTWT2}	<28>	$(1 + n + t_{AHW})T$		ns

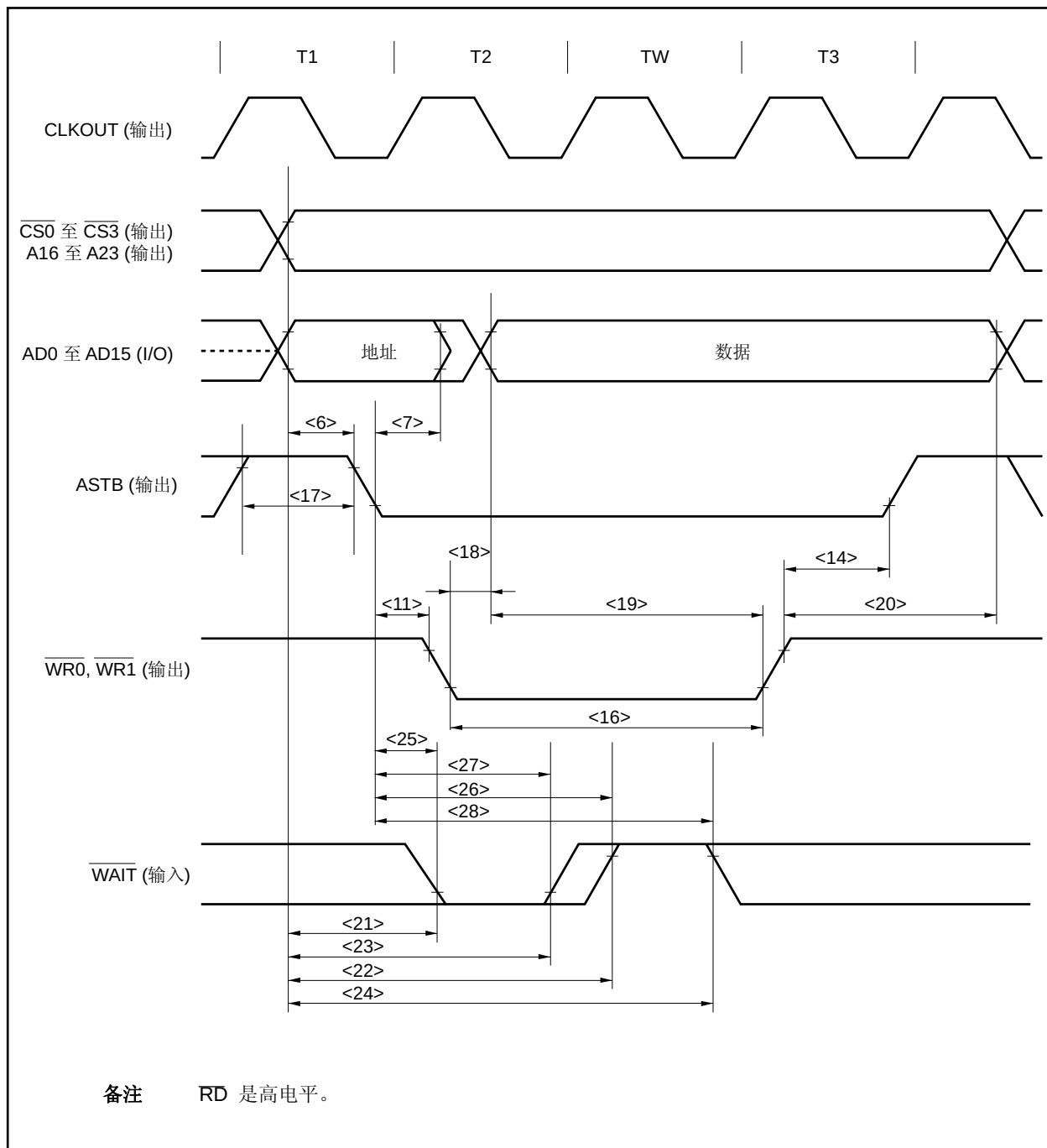
备注

1. t_{ASW} : 地址建立等待的时钟数目。
 t_{AHW} : 地址保持等待的时钟数目。
2. $T = 1/f_{CPU}$ (f_{CPU} : CPU 工作时钟频率)
3. n : 在总线周期中插入的等待时钟数目
当插入可编程等待时, 采样时序改变。
4. $m = 0, 1$
5. i : 读周期后插入的空闲状态数目 (0 或 1)
6. 在上述说明中的值是当占空比为 1: 1 的时钟从 X1 输入时的值。

读周期（CLKOUT 异步）：复用总线模式



写周期 (CLKOUT 异步): 复用总线模式



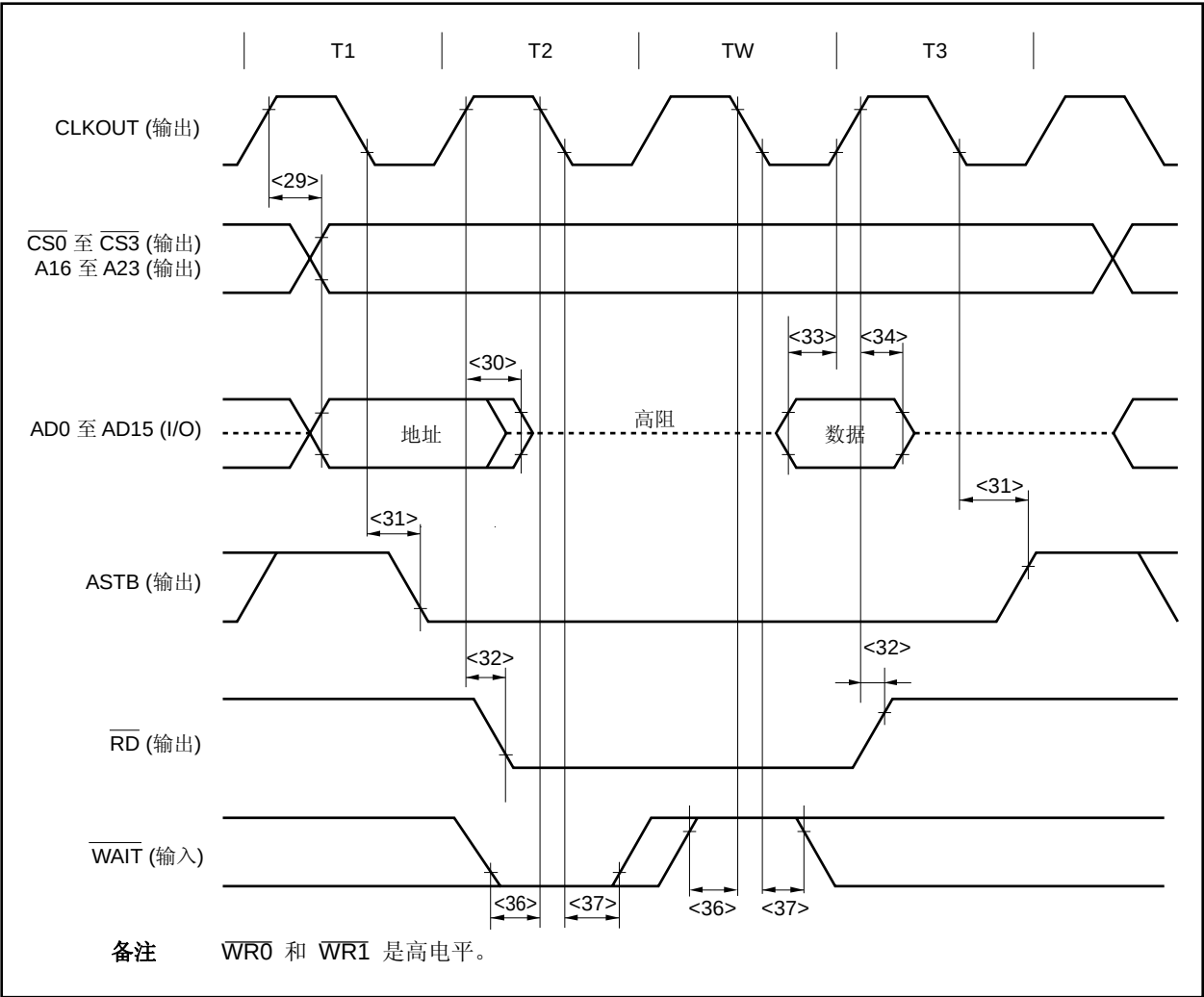
(b) 读/写周期 (CLKOUT 同步): 复用总线模式

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

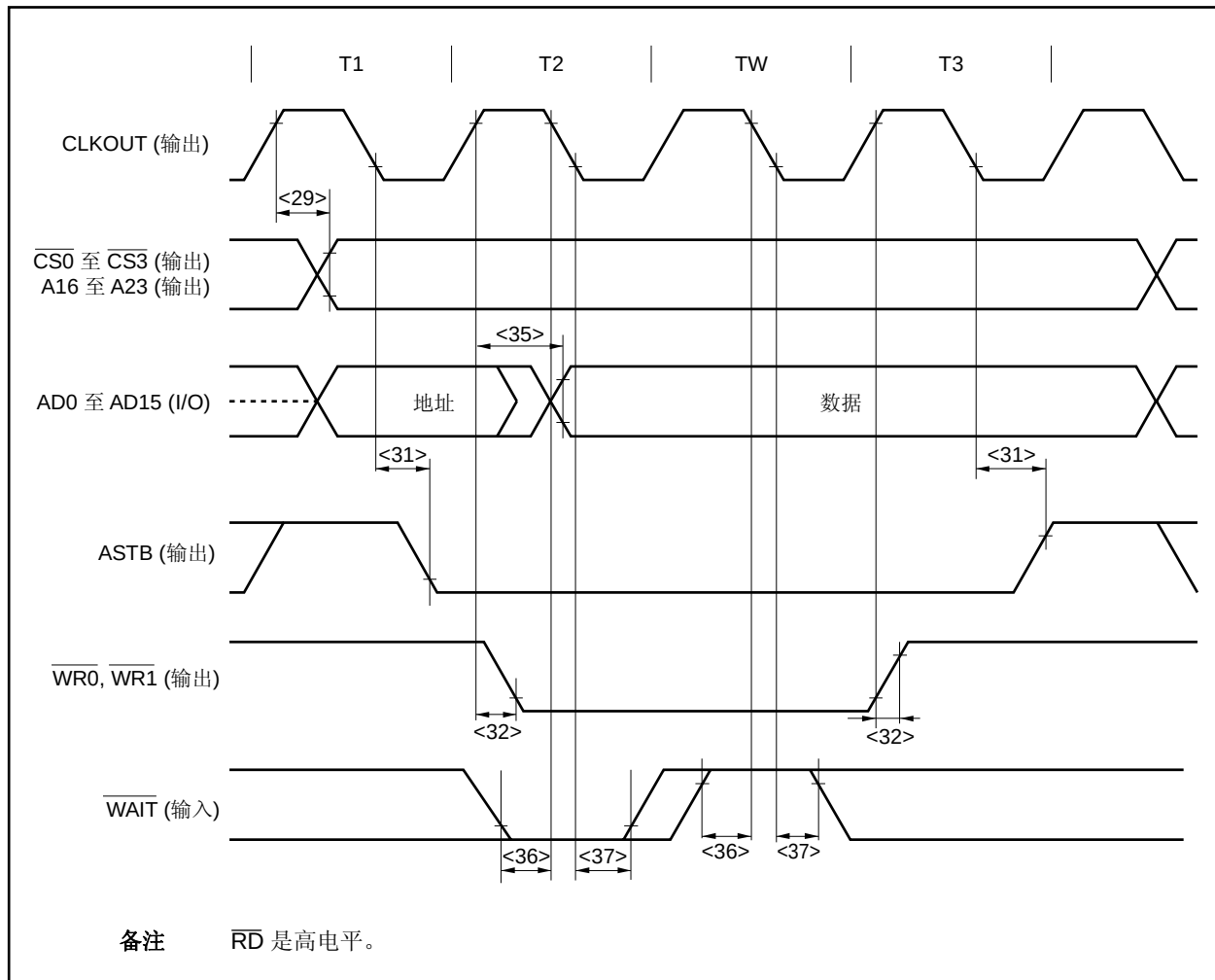
参数	符号	条件	最小值	最大值	单位
自 CLKOUT↑到地址有效的延迟时间	t_{DKA}	<29>	0	25	ns
自 CLKOUT↑到地址悬空延迟时间	t_{FKA}	<30>	0	19	ns
自 CLKOUT↓到 ASTB 的延迟时间	t_{DKST}	<31>	-12	7	ns
自 CLKOUT↑到 $\overline{RD}$ 、 $\overline{WRm}$ 的延迟时间	t_{DKRDWR}	<32>	-5	14	ns
数据输入建立时间 (至 CLKOUT↑)	t_{SIDK}	<33>	15		ns
数据输入保持时间 (自 CLKOUT↑)	t_{HKID}	<34>	5		ns
自 CLKOUT↑起数据输出延迟时间	t_{DKOD}	<35>		19	ns
$\overline{WAIT}$ 建立时间 (至 CLKOUT↓)	t_{SWTK}	<36>	20		ns
$\overline{WAIT}$ 保持时间 (自 CLKOUT↓)	t_{HKWT}	<37>	5		ns

备注 1. $m = 0, 1$
2. 在上述说明中的值是当占空比为 1: 1 的时钟从 X1 输入时的值。

读周期 (CLKOUT 同步): 复用总线模式



写周期 (CLKOUT 同步): 复用总线模式



(2) 分离总线模式

注意事项 运行于 $f_{xx} > 20 \text{ MHz}$ 时, 请确保插入地址保持等待和地址建立等待。

(a) 读/写周期 (CLKOUT 异步): 分离总线模式

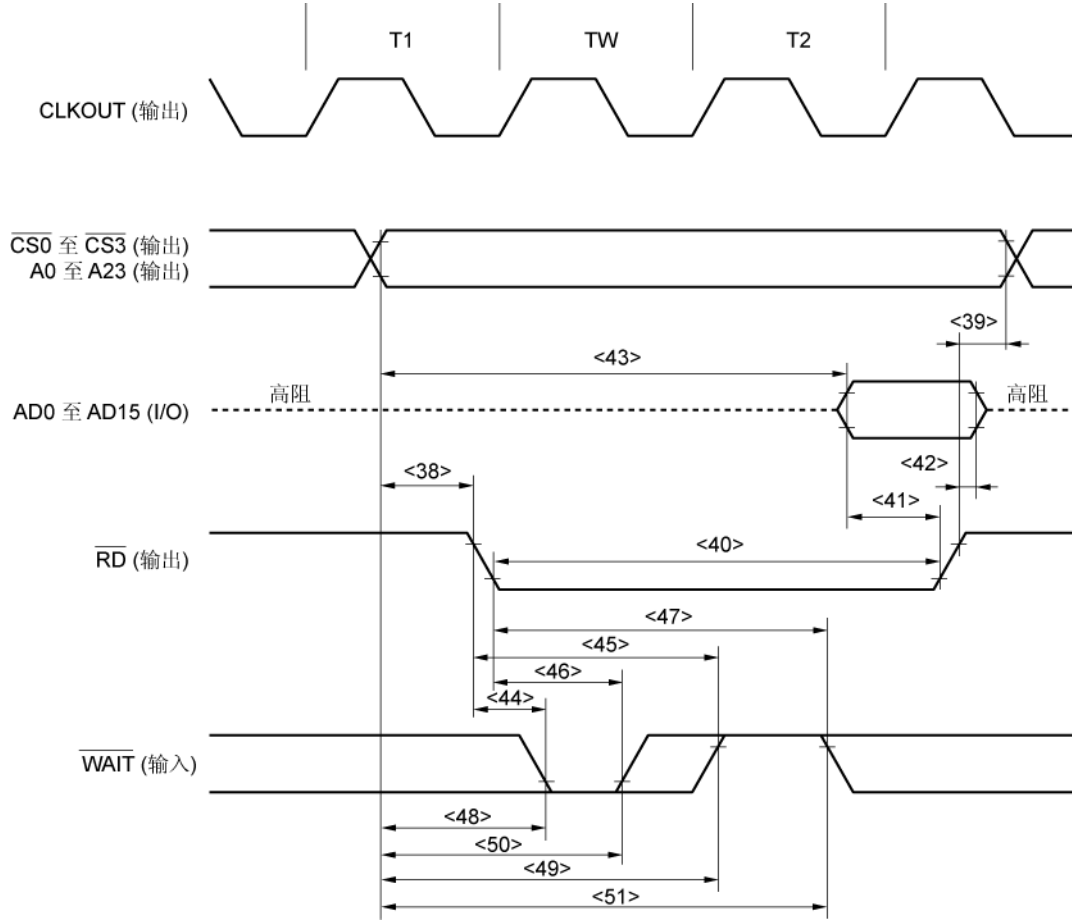
($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0 \text{ V}$, $C_L = 50 \text{ pF}$)

参数	符号	条件	最小值	最大值	单位
地址建立时间 (至 $\overline{RD}\downarrow$)	t_{SARD}	<38>	$(0.5 + t_{ASW})T - 27$		ns
地址保持时间 (从 $\overline{RD}\uparrow$ 起)	t_{HARD}	<39>	$1T - 3.5$		ns
$\overline{RD}$ 低电平宽度	t_{WRDL}	<40>	$(1.5 + n + t_{AHW})T - 10$		ns
数据建立时间 (至 $\overline{RD}\uparrow$)	t_{SISD}	<41>	23		ns
数据保持时间 (从 $\overline{RD}\uparrow$ 起)	t_{HISD}	<42>	-3.5		ns
数据建立时间 (至地址)	t_{SAID}	<43>		$(2 + n + t_{ASW} + t_{AHW})T - 40$	ns
$\overline{WAIT}$ 建立时间 (至 $\overline{RD}\downarrow$)	t_{SRDWT1}	<44>		$(0.5 + t_{AHW})T - 25$	ns
	t_{SRDWT2}	<45>		$(0.5 + n + t_{AHW})T - 25$	ns
$\overline{WAIT}$ 保持时间 (从 $\overline{RD}\downarrow$ 起)	t_{HRDWT1}	<46>	$(n - 0.5 + t_{AHW})T$		ns
	t_{HRDWT2}	<47>	$(n + 0.5 + t_{AHW})T$		ns
$\overline{WAIT}$ 建立时间 (至地址)	t_{SAWT1}	<48>		$(1 + t_{ASW} + t_{AHW})T - 45$	ns
	t_{SAWT2}	<49>		$(1 + n + t_{ASW} + t_{AHW})T - 45$	ns
$\overline{WAIT}$ 保持时间 (从地址起)	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} : 地址建立等待的时钟数目。
 t_{AHW} : 地址保持等待的时钟数目。
2. $T = 1/f_{CPU}$ (f_{CPU} : CPU 工作时钟频率)
3. n : 在总线周期中插入的等待时钟数目
当插入可编程等待时, 采样时序改变。
4. i : 读周期后插入的空闲状态数目 (0 或 1)
5. 在上述说明中的值是当占空比为 1: 1 的时钟从 X1 输入时的值。

读周期（CLKOUT 异步）：分离总线模式



备注 $\overline{WR0}$ 和 $\overline{WR1}$ 是高电平。

(b) 写周期 (CLKOUT 异步): 分离总线模式

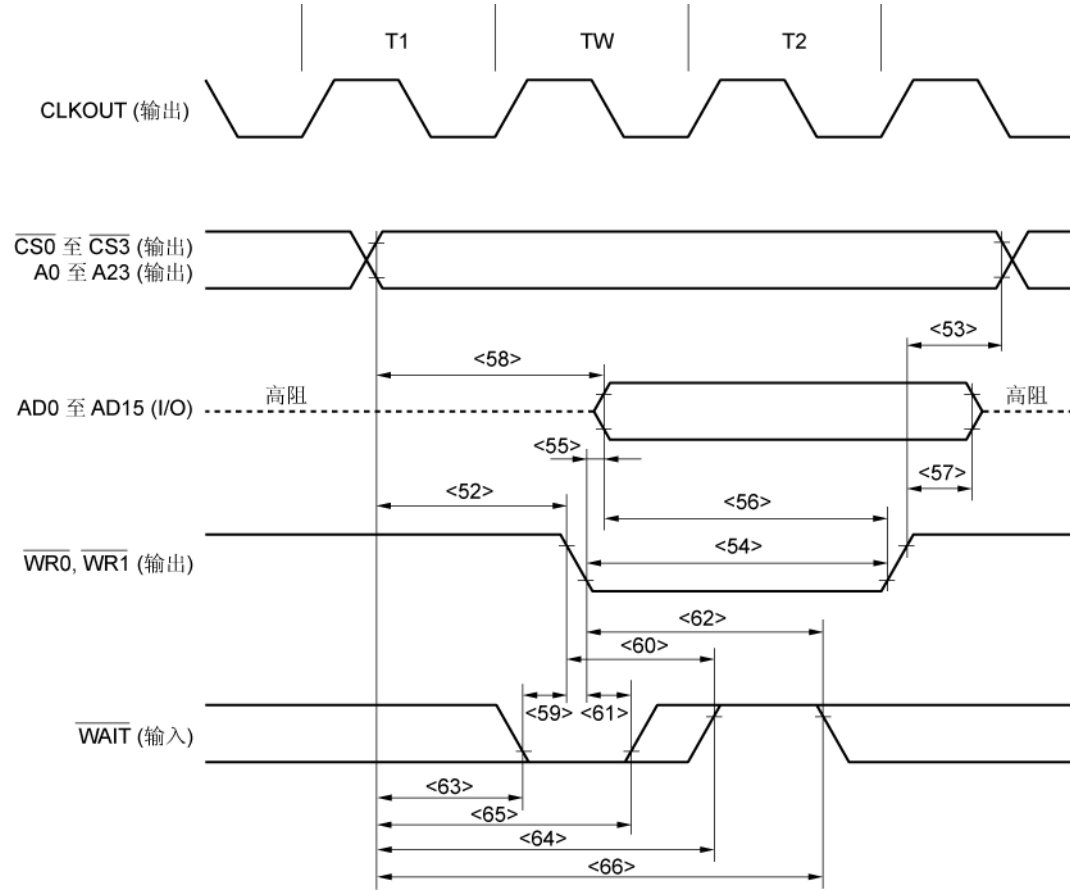
(T_A = -40 至 +85°C, V_{DD} = E_{VDD} = A_{VREF0} = A_{VREF1}, V_{SS} = E_{VSS} = A_{VSS} = 0 V, C_L = 50 pF)

参数	符号	条件	最小值	最大值	单位
地址建立时间 (至 $\overline{WRm}\downarrow$)	t _{SAWR}	<52>	(1 + t _{ASW} + t _{AHW})T - 27		ns
地址保持时间 (从 $\overline{WRm}\uparrow$ 起)	t _{HAWR}	<53>	0.5T - 6		ns
$\overline{WRm}$ 低电平宽度	t _{WWRL}	<54>	(0.5 + n)T - 10		ns
数据输出时间 从 $\overline{WRm}\downarrow$ 起	t _{DOSDW}	<55>	-5		ns
数据建立时间 (至 $\overline{WRm}\downarrow$)	t _{SOSDW}	<56>	(0.5 + n)T - 20		ns
数据保持时间(从 $\overline{WRm}\uparrow$ 起)	t _{HOSDW}	<57>	0.5T - 7		ns
数据建立时间 (至地址)	t _{SAOD}	<58>	(1 + t _{ASW} + t _{AHW})T - 25		ns
$\overline{WAIT}$ 建立时间 (至 $\overline{WRm}\downarrow$)	t _{SWRWT1}	<59>	22		ns
	t _{SWRWT2}	<60>		nT - 22	ns
$\overline{WAIT}$ 保持时间 (从 $\overline{WRm}\downarrow$)	t _{HWRWT1}	<61>	0		ns
	t _{HWRWT2}	<62>	nT		ns
$\overline{WAIT}$ 建立时间 (至地址)	t _{SAWT1}	<63>		(1 + t _{ASW} + t _{AHW})T - 45	ns
	t _{SAWT2}	<64>		(1 + n + t _{ASW} + t _{AHW})T - 45	ns
$\overline{WAIT}$ 保持时间 (从地址起)	t _{HAWT1}	<65>	(n + t _{ASW} + t _{AHW})T		ns
	t _{HAWT2}	<66>	(1 + n + t _{ASW} + t _{AHW})T		ns

备注

1. m = 0, 1
2. t_{ASW}: 地址建立等待的时钟数目。
t_{AHW}: 地址保持等待的时钟数目。
3. T = 1/f_{CPU} (f_{CPU}: CPU 工作时钟频率)
4. n: 在总线周期中插入的等待时钟数目
当插入可编程等待时, 采样时序改变。
5. 在上述说明中的值是当占空比为 1: 1 的时钟从 X1 输入时的值。

写周期（CLKOUT 异步）：分离总线模式



备注 $\overline{RD}$ 是高电平。

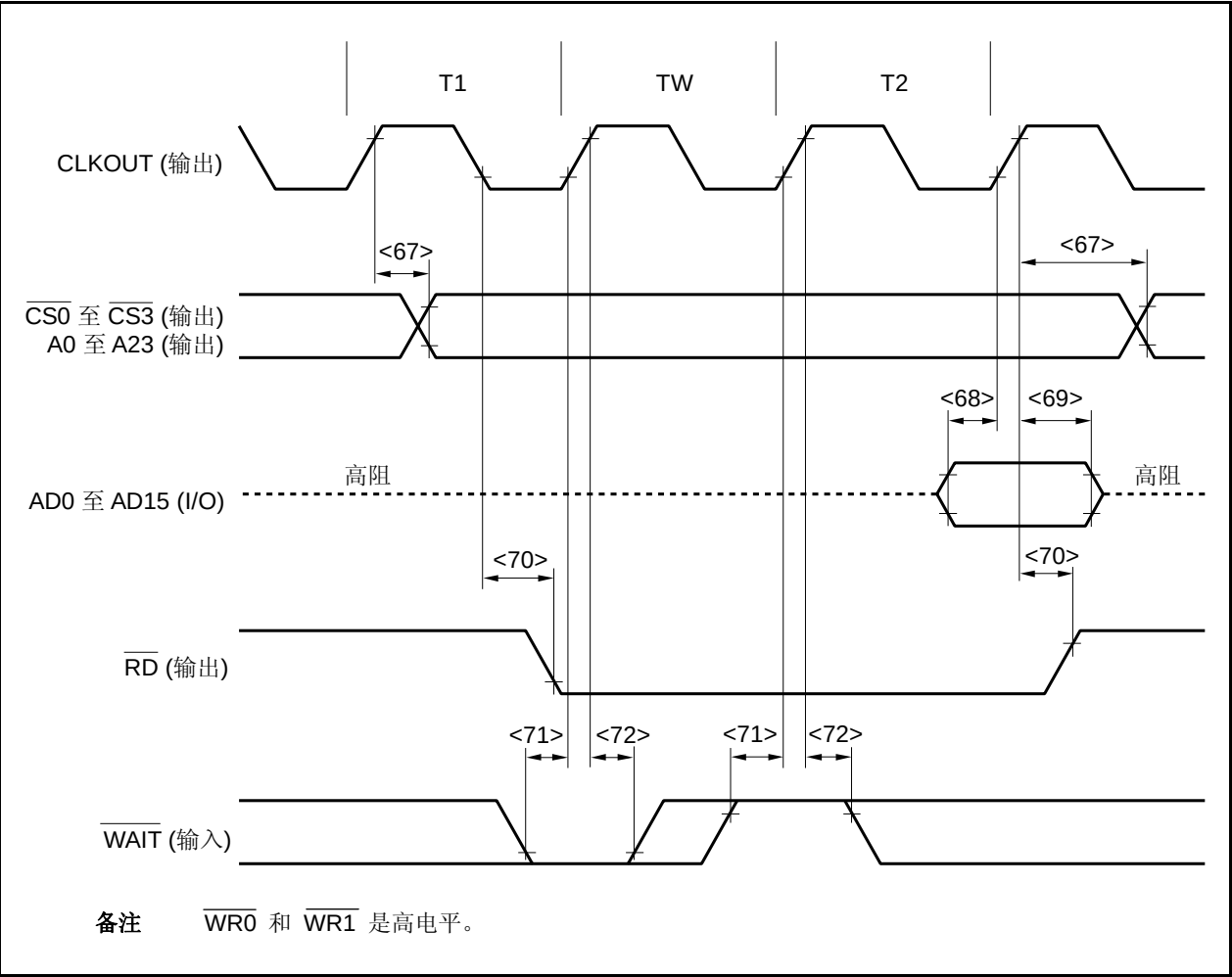
(c) 读周期 (CLKOUT 同步): 分离总线模式

(TA = -40 至+85°C, VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = AVSS = 0 V, CL = 50 pF)

参数	符号		条件	最小值	最大值	单位
自 CLKOUT↑到地址, CS 有效的延迟时间	tDKSA	<67>		0	27	ns
数据输入建立时间 (至 CLKOUT↑)	tSISDK	<68>		20		ns
数据输入保持时间 (自 CLKOUT↑起)	tHKISD	<69>		0		ns
从 CLKOUT↓到 RD 的延迟时间	tDKSR	<70>		-2	12	ns
WAIT 建立时间 (至 CLKOUT↓)	tSWTK	<71>		20		ns
WAIT 保持时间 (自 CLKOUT↓起)	tHKWT	<72>		0		ns

备注 在上述说明中的值是当占空比为 1: 1 的时钟从 X1 输入时的值。

读周期 (CLKOUT 同步, 1 等待): 分离总线模式



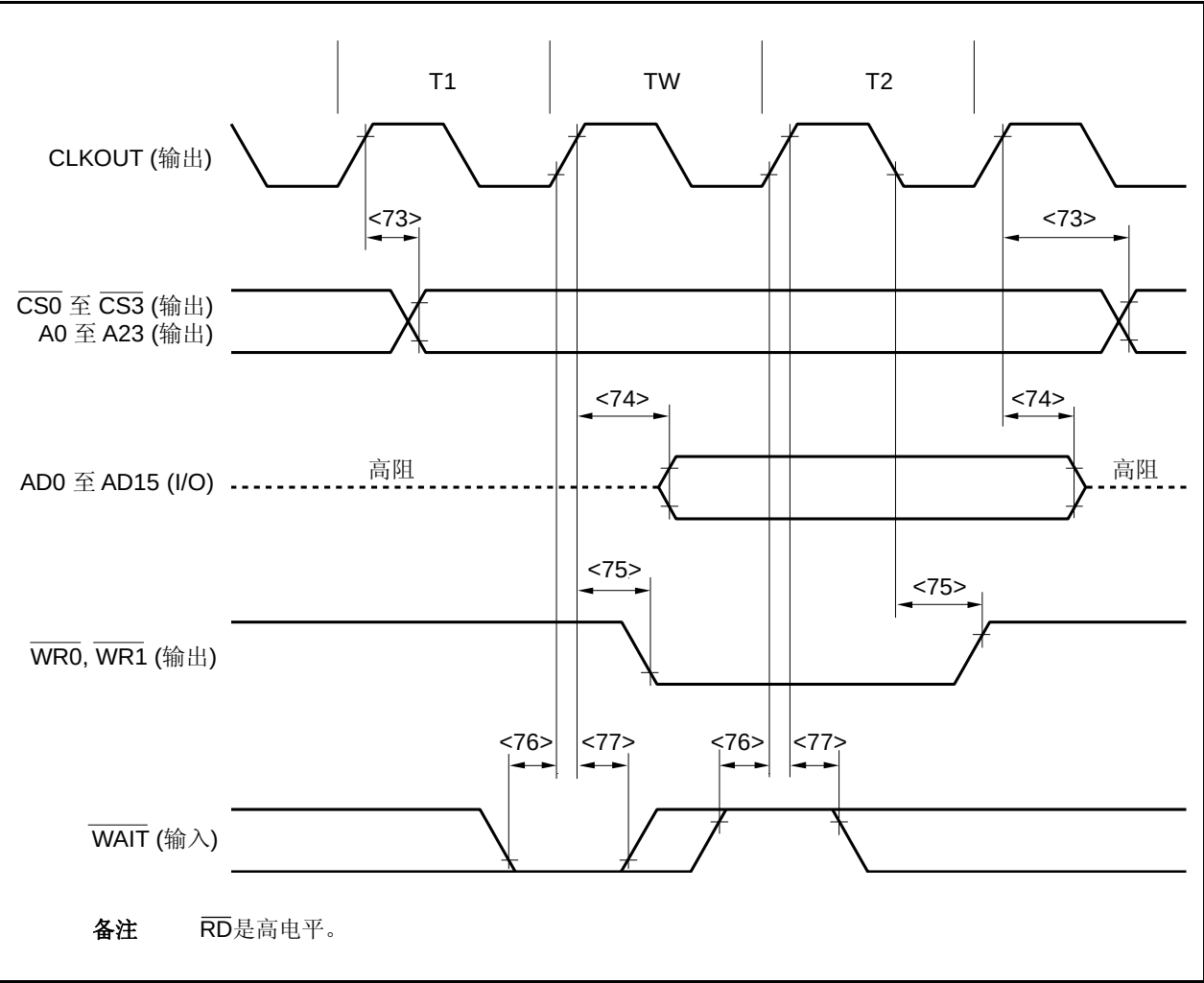
(d) 写周期 (CLKOUT 同步): 分离总线模式

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	最大值	单位
从 CLKOUT↑ 到地址, CS 的延迟时间	t_{DKSA}	<73>	0	27	ns
从 CLKOUT↑ 到数据输出的延迟时间	t_{DKSD}	<74>	0	18	ns
从 CLKOUT↑↓ 到 WRm 的延迟时间	t_{DKSW}	<75>	-2	12	ns
$\overline{\text{WAIT}}$ 建立时间 (至 CLKOUT↑)	t_{SWTK}	<76>	20		ns
$\overline{\text{WAIT}}$ 保持时间 (从 CLKOUT↑起)	t_{HKWT}	<77>	0		ns

- 备注
1. $m = 0, 1$
 2. 在上述说明中的值是当占空比为 1: 1 的时钟从 X1 输入时的值。

写周期 (CLKOUT 同步): 分离总线模式



(3) 总线保持

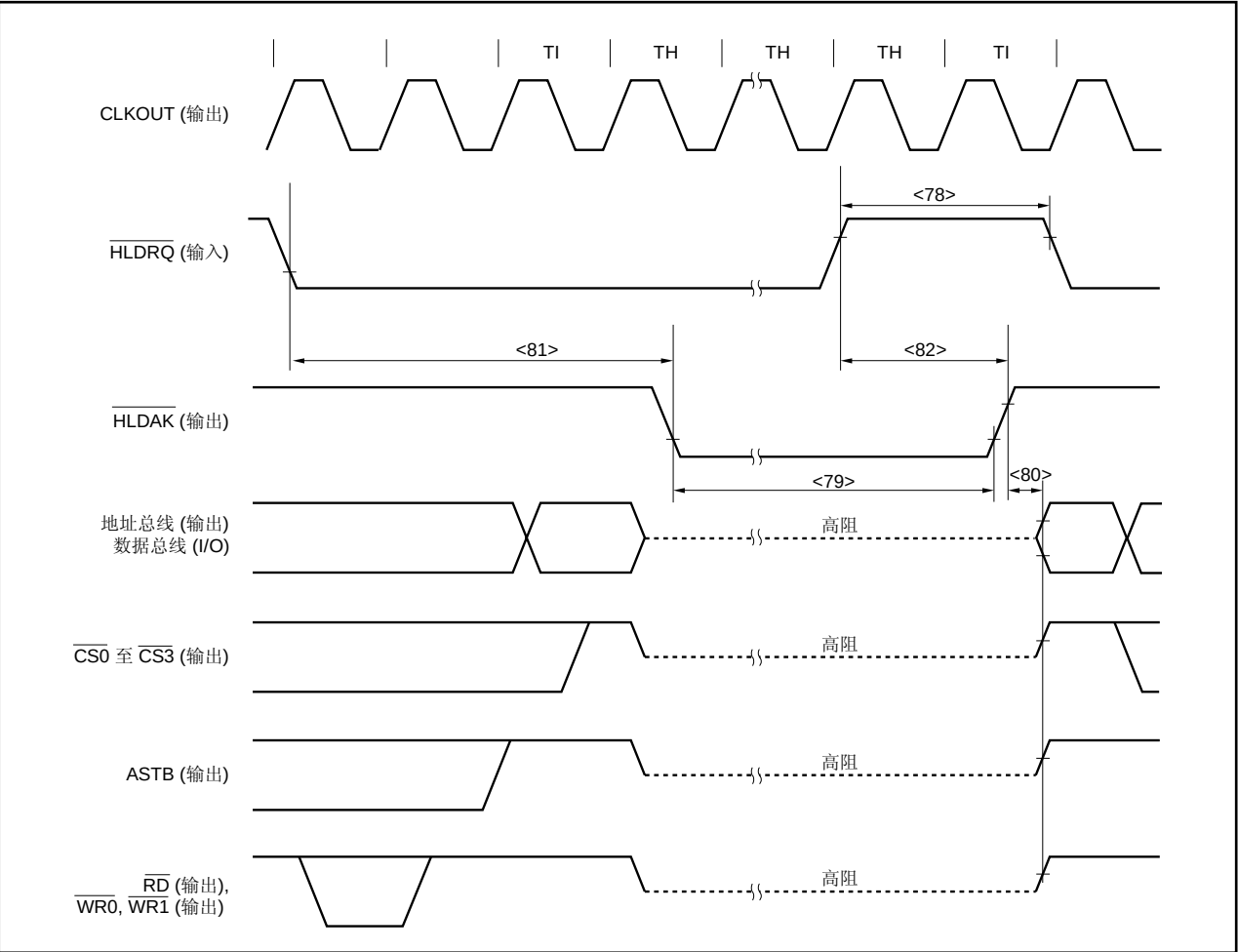
(a) CLKOUT 异步

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	最大值	单位
$\overline{\text{HLDRQ}}$ 高电平宽度	t_{WHQH}	<78>	$T + 10$		ns
$\overline{\text{HLDK}}$ 低电平宽度	t_{WHAL}	<79>	$T - 15$		ns
自 $\overline{\text{HLDK}}\uparrow$ 到总线输出的延时	t_{DHAC}	<80>	-3		ns
自 $\overline{\text{HLDRQ}}\downarrow$ 到 $\overline{\text{HLDK}}\downarrow$ 的延时	t_{DHQA1}	<81>		$(2n + 7.5)T + 26$	ns
自 $\overline{\text{HLDRQ}}\uparrow$ 到 $\overline{\text{HLDK}}\uparrow$ 的延时	t_{DHQA2}	<82>	$0.5T$	$1.5T + 26$	ns

- 备注
- 1. $T = 1/f_{\text{CPU}}$ (f_{CPU} : CPU 工作时钟频率)
 - 2. n : 在总线周期中插入的等待时钟数目
当插入可编程等待时, 采样时序改变。
 - 3. 在上述说明中的值是当占空比为 1: 1 的时钟从 X1 输入时的值。

总线保持 (CLKOUT 异步)



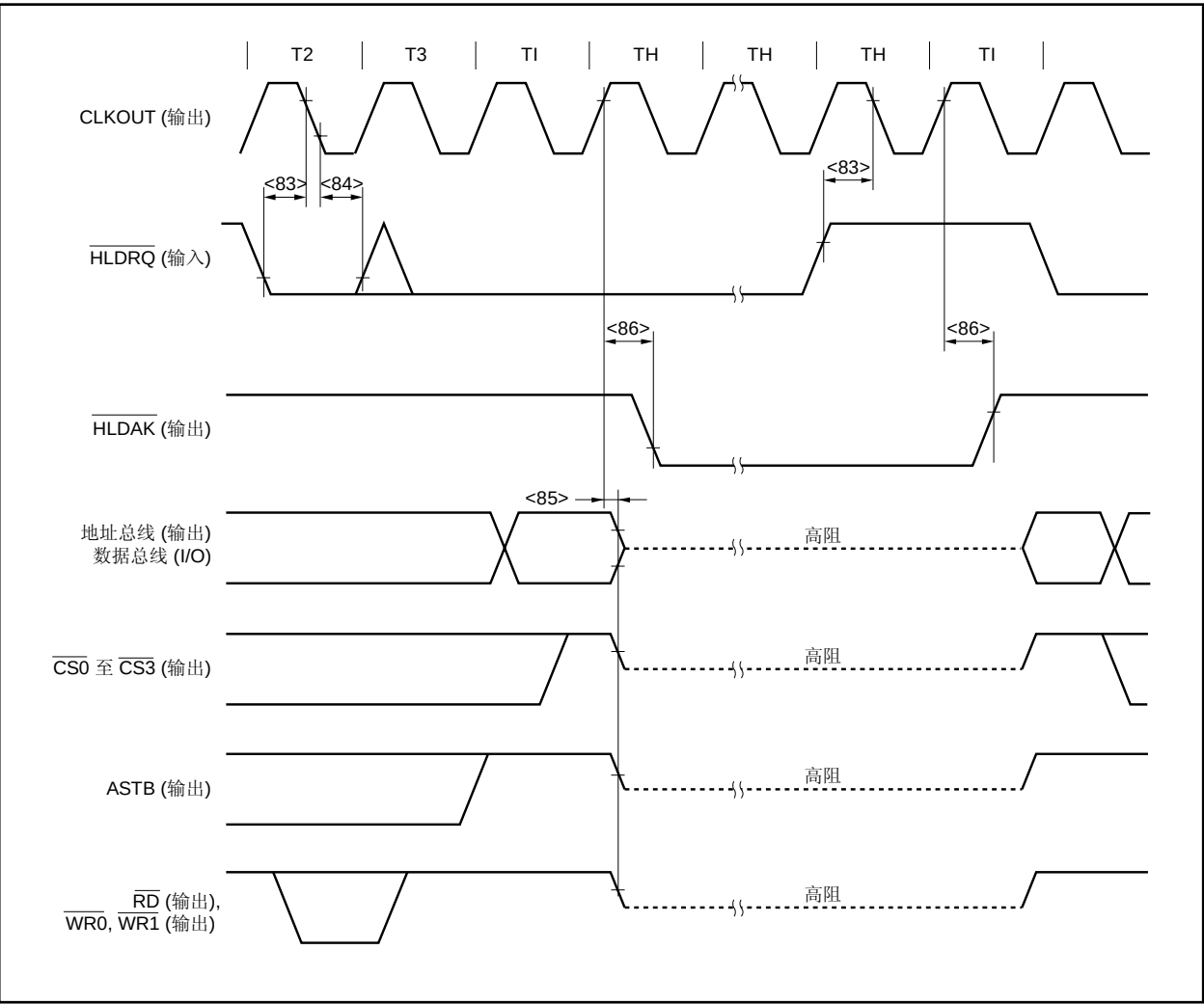
(b) CLKOUT 同步

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号		条件	最小值	最大值	单位
H $\overline{\text{LDRQ}}$ 建立时间 (至 CLKOUT $\downarrow$)	t_{SHQK}	<83>		20		ns
H $\overline{\text{LDRQ}}$ 保持时间 (自 CLKOUT $\downarrow$)	t_{HKHQ}	<84>		5		ns
自 CLKOUT $\uparrow$ 至总线悬空的延时	t_{DKF}	<85>			19	ns
自 CLKOUT $\uparrow$ 至 H $\overline{\text{LDAK}}$ 有效时的延时	t_{DKHA}	<86>			19	ns

备注 在上述说明中的值是当占空比为 1: 1 的时钟从 X1 输入时的值。

总线保持 (CLKOUT 同步)

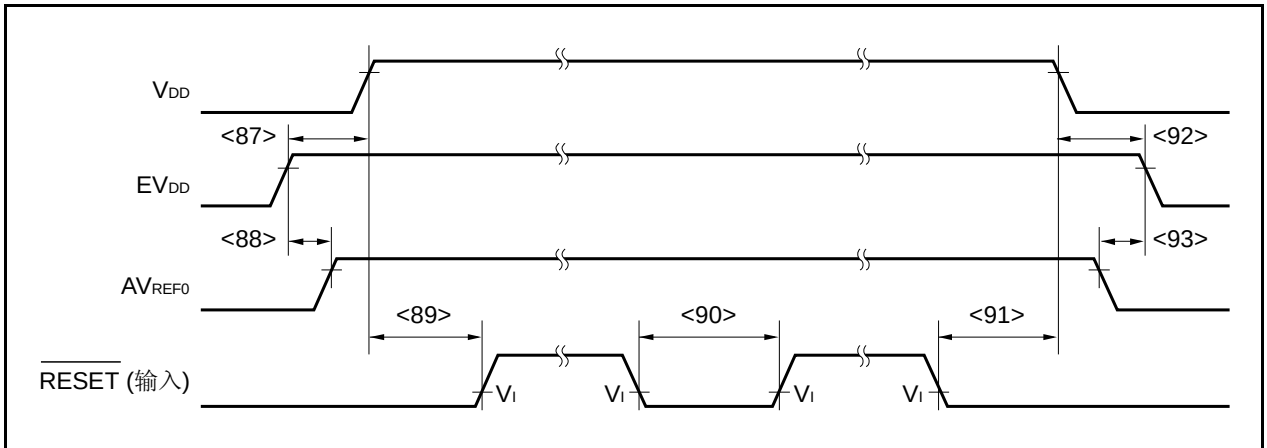


电源开启/关闭/复位时序

(TA = -40 至 +85°C, VSS = AVSS = EVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
EVDD↑ → VDD↑	tREL	<87>	0		ns
EVDD↑ → AVREF0, AVREF1↑	tREA	<88>	0	tREL	ns
VDD↑ → RESET↑	tRER	<89>	500 + tREG [‡]		ns
RESET低电平宽度	tWRSL	模拟噪声消除 (flash擦除/写入期间)	500		ns
		模拟噪声消除	500		ns
RESET↓ → VDD↓	tFRE	<91>	500		ns
VDD↓ → EVDD↓	tFEL	<92>	0		ns
AVREF0↓ → EVDD↓	tFEA	<93>	0	tFEL	ns

注 取决于片上稳压器的特性。



中断时序

(TA = -40 至+85°C, VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	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至8 (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至8 (模拟噪声消除)	500		ns
		n = 3 (数字噪声消除)	3T _{SMP} + 20		ns

备注 T_{SMP}: 噪声消除采样时钟周期。

按键返回时序

(TA = -40 至 +85°C, VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
KRn 高电平宽度	tWKRH	模拟噪声消除	500		ns
KRn 低电平宽度	tWKRL	模拟噪声消除	500		ns

备注 n = 0 至 7

定时器时序

(TA = -40 至 +85°C, VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
TI 高电平宽度	tTIH	TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, TIP31, TIP40, TIP41, TIP50, TIP51, TIP60, TIP61, TIP70, TIP71, TIP80, TIP81, TIQ00至TIQ03	2T + 20		ns
TI 低电平宽度	tTIL		2T + 20		ns

备注 T = 1/fxx

UART 时序

(TA = -40 至 +85°C, VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
发送速率				625	kbps
ASCK0 周期时间				10	MHz

CSIB 时序

(1) 主模式

(TA = -40 至+85°C, VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = AVSS = 0 V, CL = 50 pF)

参数	符号		条件	最小值	最大值	单位
SCKBn周期时间	t _{KCY1}	<94>		125		ns
SCKBn高/低电平宽度	t _{KH1} , t _{KL1}	<95>		t _{KCY1} /2 – 8		ns
SIBn建立时间 (到SCKBn↑)	t _{SIK1}	<96>		27		ns
SIBn保持时间(从SCKBn↑)	t _{KS1}	<97>		27		ns
从SCKBn↓ 到SOBn输出的延迟时间	t _{KSO1}	<98>			27	ns

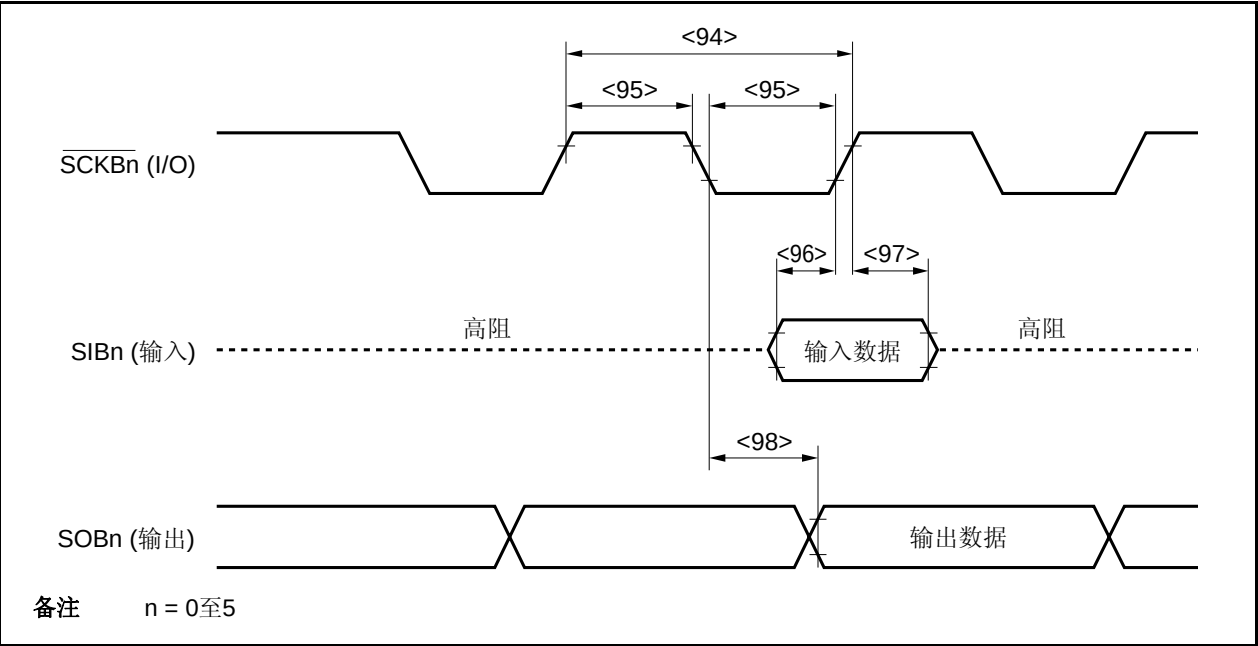
备注 n = 0 至 5

(2) 从模式

(TA = -40 至+85°C, VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = AVSS = 0 V, CL = 50 pF)

参数	符号		条件	最小值	最大值	单位
SCKBn 周期时间	t _{KCY2}	<94>		125		ns
SCKBn高/低电平宽度	t _{KH2} , t _{KL2}	<95>		54.5		ns
SIBn建立时间 (到 SCKBn↑)	t _{SIK2}	<96>		27		ns
SIBn保持时间(从SCKBn↑)	t _{KS2}	<97>		27		ns
从SCKBn↓ 到SOBn输出的延迟时间	t _{KSO2}	<98>			27	ns

备注 n = 0 至 5



I²C 总线模式(T_A = -40 至 +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

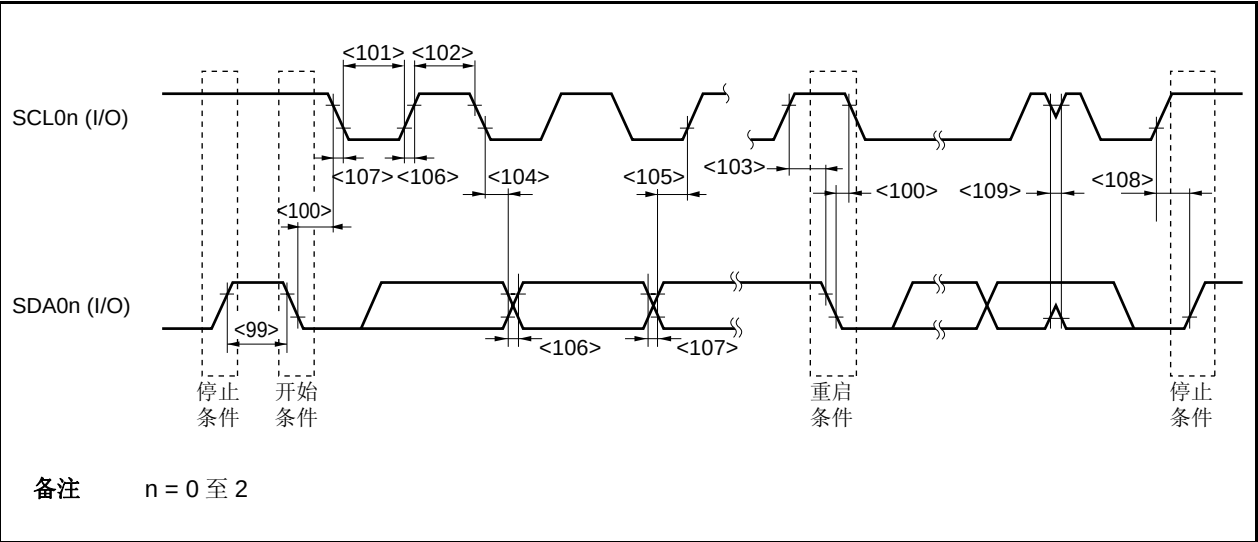
参数	符号		普通模式		高速模式		单位
			最小值	最大值	最小值	最大值	
SCL0n 时钟频率	f _{CLK}		0	100	0	400	kHz
总线空闲时间 (开始和停止状态之间)	t _{BUF}	<99>	4.7	—	1.3	—	μs
保持时间 ¹	t _{HD: STA}	<100>	4.0	—	0.6	—	μs
SCL0n 时钟低电平宽度	t _{LOW}	<101>	4.7	—	1.3	—	μs
SCL0n 时钟高电平宽度	t _{HIGH}	<102>	4.0	—	0.6	—	μs
开始/重新开始状态的建立时间	t _{SU: STA}	<103>	4.7	—	0.6	—	μs
数据保持时间	CBUS 兼容主机	t _{HD: DAT}	<104>	5.0	—	—	μs
	I ² C 模式		0 ²	—	0 ²	0.9 ³	μs
数据建立时间	t _{SU: DAT}	<105>	250	—	100 ⁴	—	ns
SDA0n 和 SCL0n 信号上升时间	t _R	<106>	—	1000	20 + 0.1Cb ⁵	300	ns
SDA0n 和 SCL0n 信号下降时间	t _F	<107>	—	300	20 + 0.1Cb ⁵	300	ns
停止状态建立时间	t _{SU: STO}	<108>	4.0	—	0.6	—	μs
由输入滤波器抑制的毛刺脉冲宽度	t _{SP}	<109>	—	—	0	50	ns
每个信号总线的电容负载	C _b		—	400	—	400	pF

- 注**
1. 在开始条件时，第一个时钟脉冲在保持时间后产生。
 2. 系统内部 SDA0n (在 SCL0n 信号的 V_{IHmin} 时) 信号需要最短 300 ns 的保持时间，以占用 SCL0n 信号下降沿时的不确定区域。
 3. 如果系统不延长 SCL0n 信号低电平保持时间 (t_{LOW})，则只需要满足最大数据保持时间 (t_{HD: DAT})。
 4. 在正常模式 I²C 总线系统中可以适用高速模式 I²C 总线。这种情况下，设置高速模式 I²C 总线使其满足以下条件。
 - 如果系统不延长 SCL0n 信号低电平保持时间：

$$t_{SU: DAT} \geq 250 \text{ ns}$$
 - 如果系统延长 SCL0n 信号低电平保持时间：
 发送后继的数据位到 SDA0n 信号线优先于 SCL0n 信号线释放 (t_{Rmax} + t_{SU: DAT} = 1,000 + 250 = 1,250 ns：正常 I²C 总线模式规范)。
 5. C_b：每条总线的总电容 (单位：pF)。

备注 n = 0 至 2

i²C 总线模式



A/D 转换器

(T_A = -40 至+85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, 3.0 V ≤ AV_{REF0} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
分辨率					10	位
总体误差 [‡]		3.0 V ≤ AV _{REF0} ≤ 3.6 V			±0.6	%FSR
转换时间	t _{CONV}		2.6		24	μs
零标称误差					±0.5	%FSR
满量程误差					±0.5	%FSR
非线性误差					±4.0	LSB
微分线性误差					±4.0	LSB
模拟输入电压	V _{IAN}		AV _{SS}		AV _{REF0}	V
参考电压	AV _{REF0}		3.0		3.6	V
AV _{REF0} 电流	AI _{REF0}	正常转换模式下		3	6.5	mA
		高速转换模式下		4	10	mA
		A/D转换器未使用时			5	μA

注 量化误差除外 (±0.05% FSR)。

注意事项 在 A/D 转换期间, 不要设置 (读/写) 复用功能端口, 否则转换分辨率可能会下降。

备注 LSB: 最低有效位
FSR: 满量程范围

D/A 转换器

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $3.0\text{ V} \leq AV_{REF1} \leq 3.6\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	典型值	最大值	单位
分辨率					8	位
总误差		$R = 2\text{ M}\Omega$			± 1.2	%FSR
设置时间		$C = 20\text{ pF}$			3	μs
寄存器输出	R_o	输出数据 55H		6.42		$\text{k}\Omega$
参考电压	AV_{REF1}		3.0		3.6	V
AVREF1 电流 ^注	AI_{REF1}	D/A 转换工作		1	2.5	mA
		D/A 转换停止			5	μA

注 1. 量化误差除外 ($\pm 0.5\text{ LSB}$)。
2. D/A 转换器的 1 通道的值。

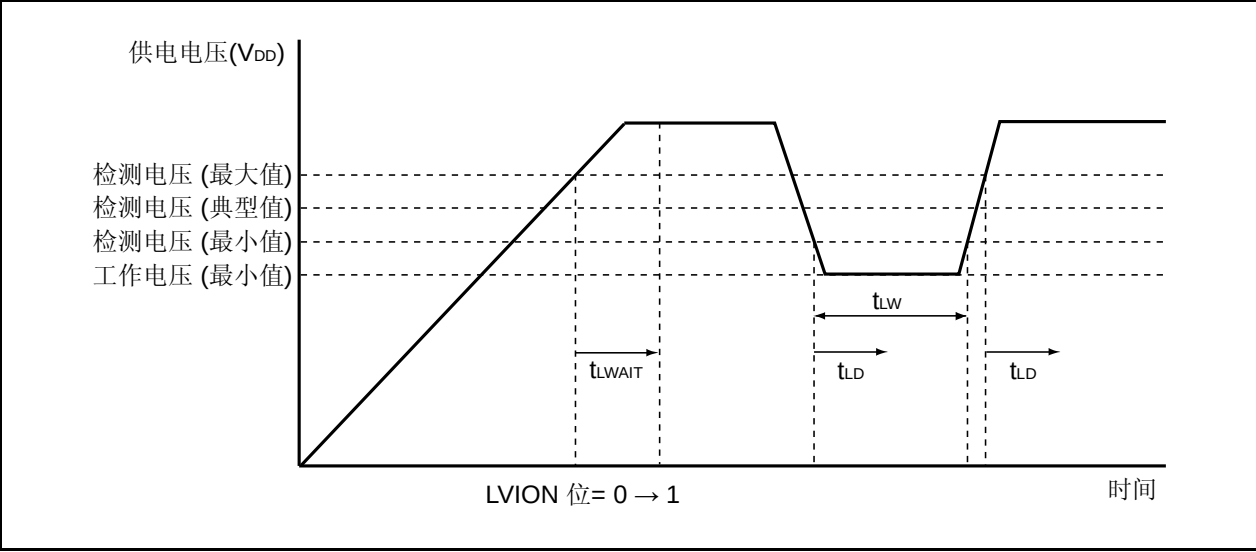
备注 R 是输出引脚负载电阻, C 是输出引脚负载电容。

LVI 电路特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V_{LV10}		2.85	2.95	3.05	V
响应时间 ^注	t_{LD}	在 V_{DD} 达到 V_{LV10} (最大值), 或在 V_{DD} 已跌落到 V_{LV10} (最大值) 之后		0.2	2.0	ms
最小脉冲宽度	t_{LW}		0.2			ms
参考电压稳定等待时间	t_{LWAIT}	在 V_{DD} 达到 2.85 V (最小值) 之后		0.1	0.2	ms

注 检测到检测电压和输出一个中断或复位信号所需的时间。

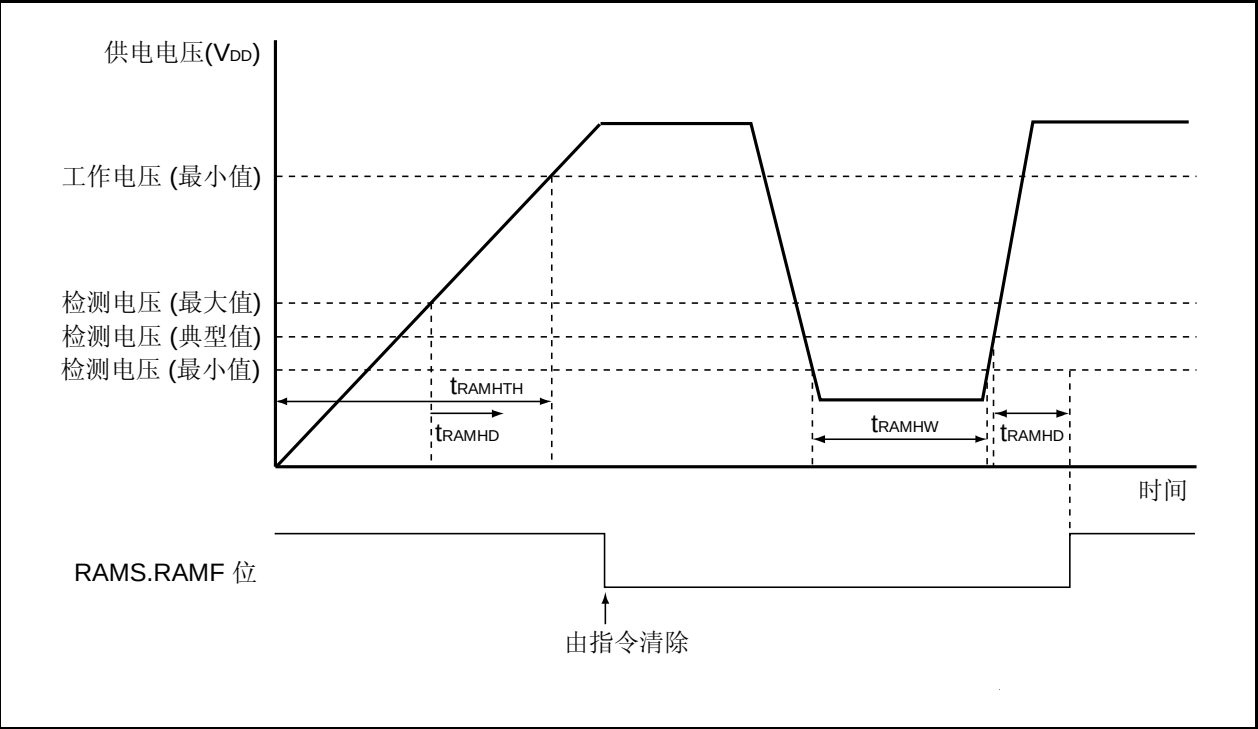


RAM 保持检测

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V_{RAMH}		1.9	2.0	2.1	V
供电电压上升时间	t_{RAMHTH}	$V_{DD} = 0$ 至 2.85 V	0.002			ms
响应时间 ^注	t_{RAMHD}	在 V_{DD} 达到 2.1 V 之后		0.2	3.0	ms
最小脉冲宽度	t_{RAMHW}		0.2			ms

注 检测电压所需的时间，设置 RAMS.RAMF 位所需的时间。



Flash 存储器编程特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

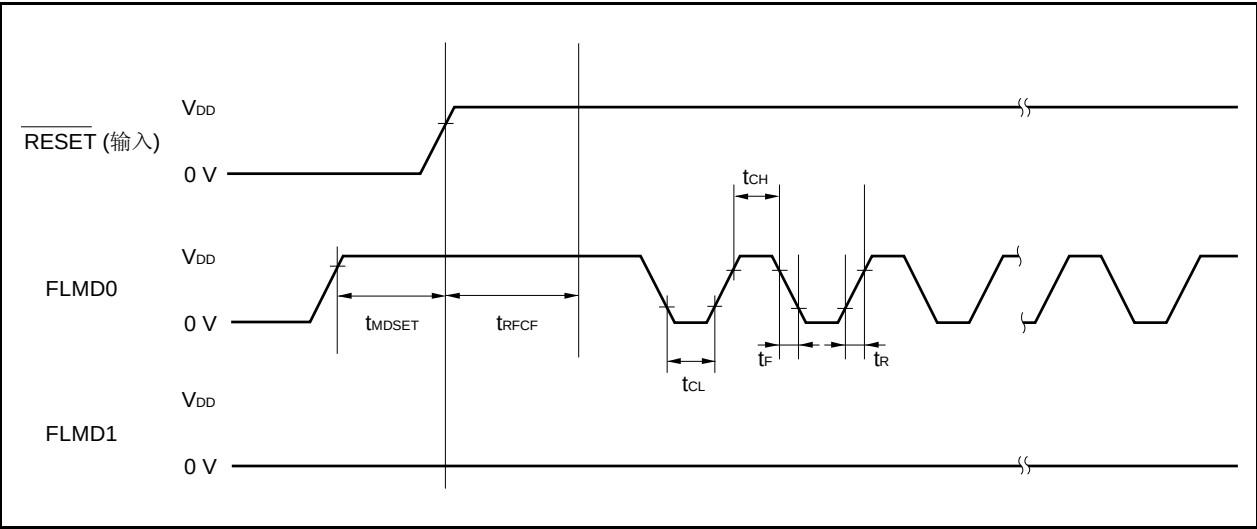
(1) 基本特性

参数	符号	条件	最小值	典型值	最大值	单位
工作频率	f_{CPU}		2.5		32	MHz
供电电压	V_{DD}		2.85		3.6	V
重写次数	C_{WRT}				100	次
编程温度	t_{PRG}		-40		+85	$^{\circ}\text{C}$

(2) 串行写入操作特性

参数	符号	条件	最小值	典型值	最大值	单位
FLMD0, FLMD1建立时间	t_{MDSET}		2		3000	ms
自 $\overline{\text{RESET}}\uparrow$ 到FLMD0 计数启动时间	t_{RFCF}	$f_x = 2.5$ 至 10 MHz	800			μs
FLMD0 计数器 高电平宽度/ 低电平宽度	$t_{\text{CH}}/t_{\text{CL}}$		10		100	μs
FLMD0 计数器上升时间/下降时间	$t_{\text{r}}/t_{\text{f}}$				1	μs

Flash 写入模式建立时序



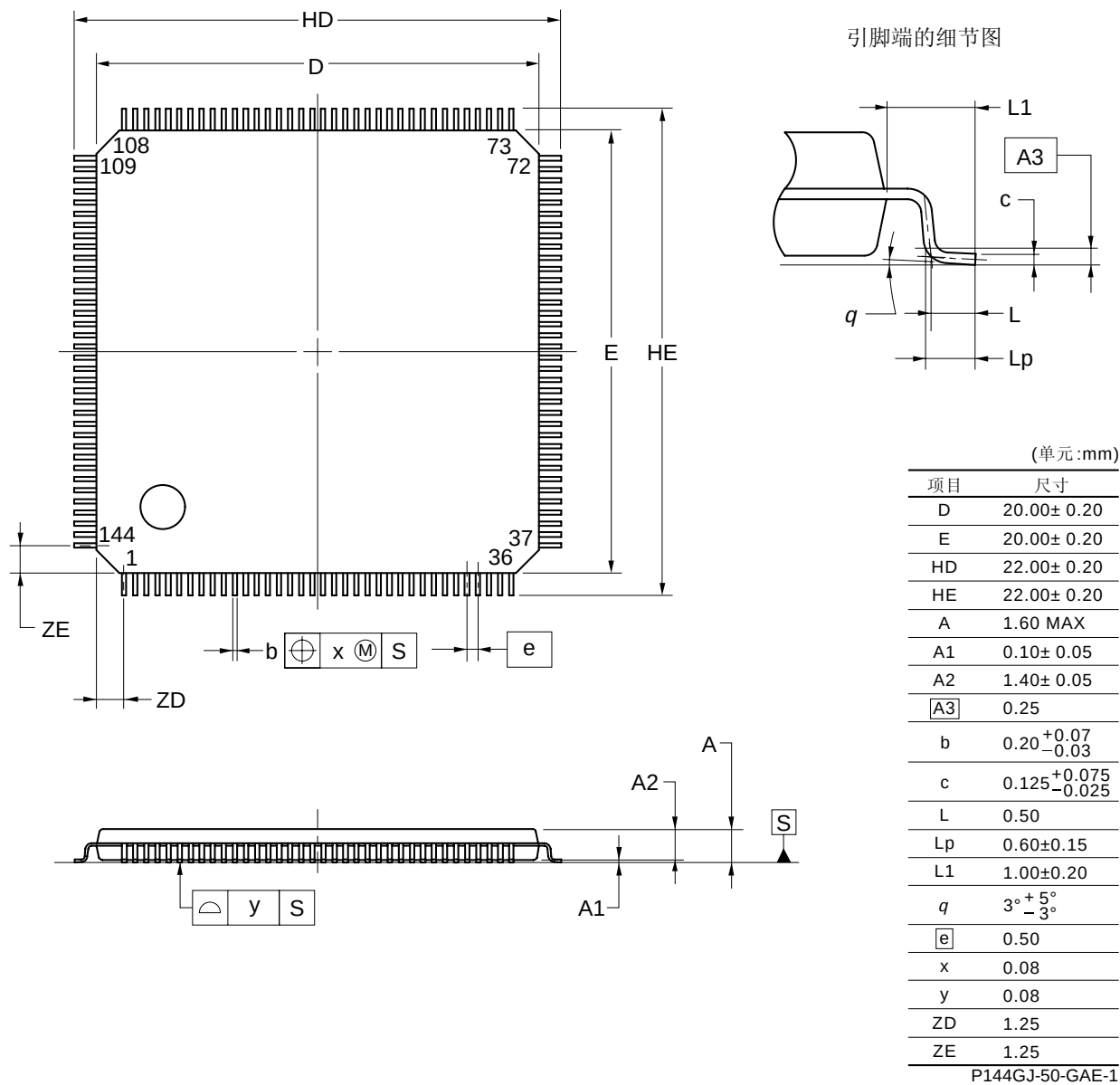
(3) 编程特性

参数	符号	条件	最小值	典型值	最大值	单位
芯片擦除时间		f _{xx} = 32 MHz 批量擦除		105		ms
每 256 字节的写入时间		f _{xx} = 32 MHz		2.0		ms
块内部验证时间		f _{xx} = 32 MHz		10		ms
块空白检查时间		f _{xx} = 32 MHz		0.5		ms
Flash 存储器信息设置时间		f _{xx} = 32 MHz		30		ms

注意事项 当首次对出厂产品进行写入时，不论是进行“擦除-写入”操作，还是“仅写入”操作，都被当作是一次重写。

举例（P：写入， E：擦除）：
出厂产品 → P → E → P → E → P： 3 次重写
出厂产品 → E → P → E → P → E → P： 3 次重写

144-针塑封 LQFP (密间距) (20x20)



© NEC Electronics Corporation 2007

第三十一章 推荐焊接条件

V850ES/JJ3 产品应该在下列推荐条件下进行焊接和安装。
相关技术信息，请浏览下面的网页。

半导体设备装配手册 (<http://www.necel.com/pkg/en/mount/index.html>)

表 31-1. 表面贴装类型焊接条件

μPD70F3743GJ-GAE-AX:	144-针塑封 LQFP (密间距) (20 × 20)
μPD70F3744GJ-GAE-AX:	144-针塑封 LQFP (密间距) (20 × 20)
μPD70F3745GJ-GAE-AX:	144-针塑封 LQFP (密间距) (20 × 20)
μPD70F3746GJ-GAE-AX:	144-针塑封 LQFP (密间距) (20 × 20)

焊接方法	焊接条件	推荐条件符号
红外回流	封装峰值温度: 260°C, 时间: 最长 60 秒 (大于等于 220°C), 次数: 小于等于 3 次, 曝光限制: 7 天 [※] (此后, 在 125°C 预烘 20 至 72 小时)	IR60-207-3
局部加热	引脚温度: 350°C 最大值., 时间: 最大 3 秒 (每行引脚)	—

注 打开干燥包装后, 在有效存放期内将其存储在 25°C 或更低温度, 65% RH 或更低湿度的环境中。

注意事项 不要一起使用不同的焊接方法 (局部加热法除外)。

- 备注
- 1. 部件编号末尾带有-AX 的产品是无铅产品。
 - 2. 如果不使用以上推荐的焊接方法和条件, 请与 NEC 电子公司销售代理联系。

附录 A 开发工具

以下开发工具可用来开发使用 V850ES/JJ3 单片机的系统。

图 A-1 显示了开发工具的配置。

- **支持 PC98-NX 系列**

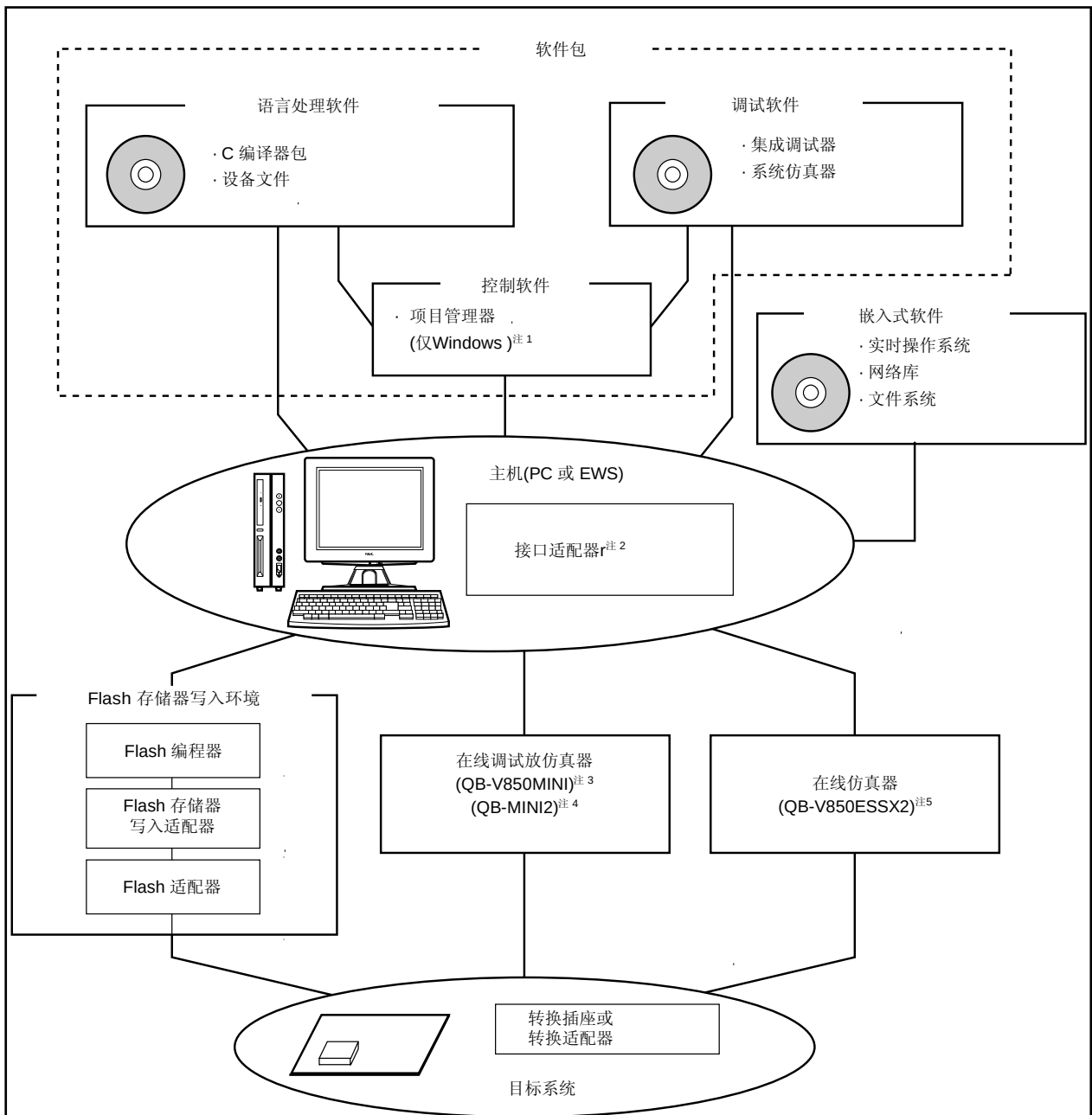
除非另行说明，IBM PC/AT™ 支持兼容的产品兼容于PC98-NX 系列计算机。当使用PC98-NX系列计算机时，可参见IBM PC/AT兼容机说明。

- **Windows™**

除非另行说明，“Windows”指以下操作系统。

- Windows 98, 2000
- Windows Me
- Windows XP
- Windows NT™ Ver. 4.0

图 A-1. 开发工具配置



注

1. 项目管理器 PM+ 包含在 C 编译包内。
PM+ 只能在 Windows 下使用。
2. QB-V850MINI、QB-MINI2 和 QB-V850ESSX2 仅支持 USB 接口。
3. QB-V850MINI 提供 ID850QB、USB 接口电缆、OCD 电缆、自检板、KEL 适配器和 KEL 连接器。所有其它部件均为可选。
4. QB-MINI2 提供 USB 接口电缆、16-引脚目标电缆、10-引脚目标电缆和 78K0-OCD 板（不提供集成调试器。）所有其它部件均为可选。
5. QB-V850ESSX2 提供 ID850QB、简易 flash 编程器、供电单元和 USB 接口适配器。所有其它部件均为可选。

A.1 软件包

V850 微控制器的软件包 SP850	本软件包内包括用于 V850 微控制器的通用开发工具（软件）。
	型号： μ SxxxxSP850

备注 xxxx型号随使用的主机和操作系统的不同而异。

μ SxxxxSP850

xxxx	主机	操作系统	提供形式
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 编译包	本编译器将用 C 语言写的程序转换为微控制器执行的目标代码。本编译器由项目管理器 PM+ 内启动。
	型号： μ SxxxxCA703000
DF703746 设备文件	本文件包含器件的特有信息。 该设备文件应和开发工具（CA850、SM850 或 ID850QB）配合使用。 相应的主机和操作系统随使用的工具不同而不同。T

备注 xxxx型号随使用的主机和操作系统的不同而异。

μ SxxxxCA703000

xxxx	主机	操作系统	提供形式
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，设计使得用户可以在 Windows 环境下有效地开发用户程序。所有用户开发过程中的操作，诸如启动编辑器、构建以及进行调试等，都可以在 PM+ 中进行。 <注意事项> PM+ 包含在 C 编译包 CA850 之中。 它仅能在 Windows 下使用。
--------------	---

A.4 调试工具（硬件）

A.4.1 使用IECUBE® QB-V850ESSX2 时

当将 QB-V850ESSX2 连接至主机（PC-9821 系列，PC/AT 兼容机）时的系统配置如下所示。即使不备有可选部件，照样可以完成连接。

图 A-2. 系统配置（使用 QB-V850ESSX2 时）（1/2）

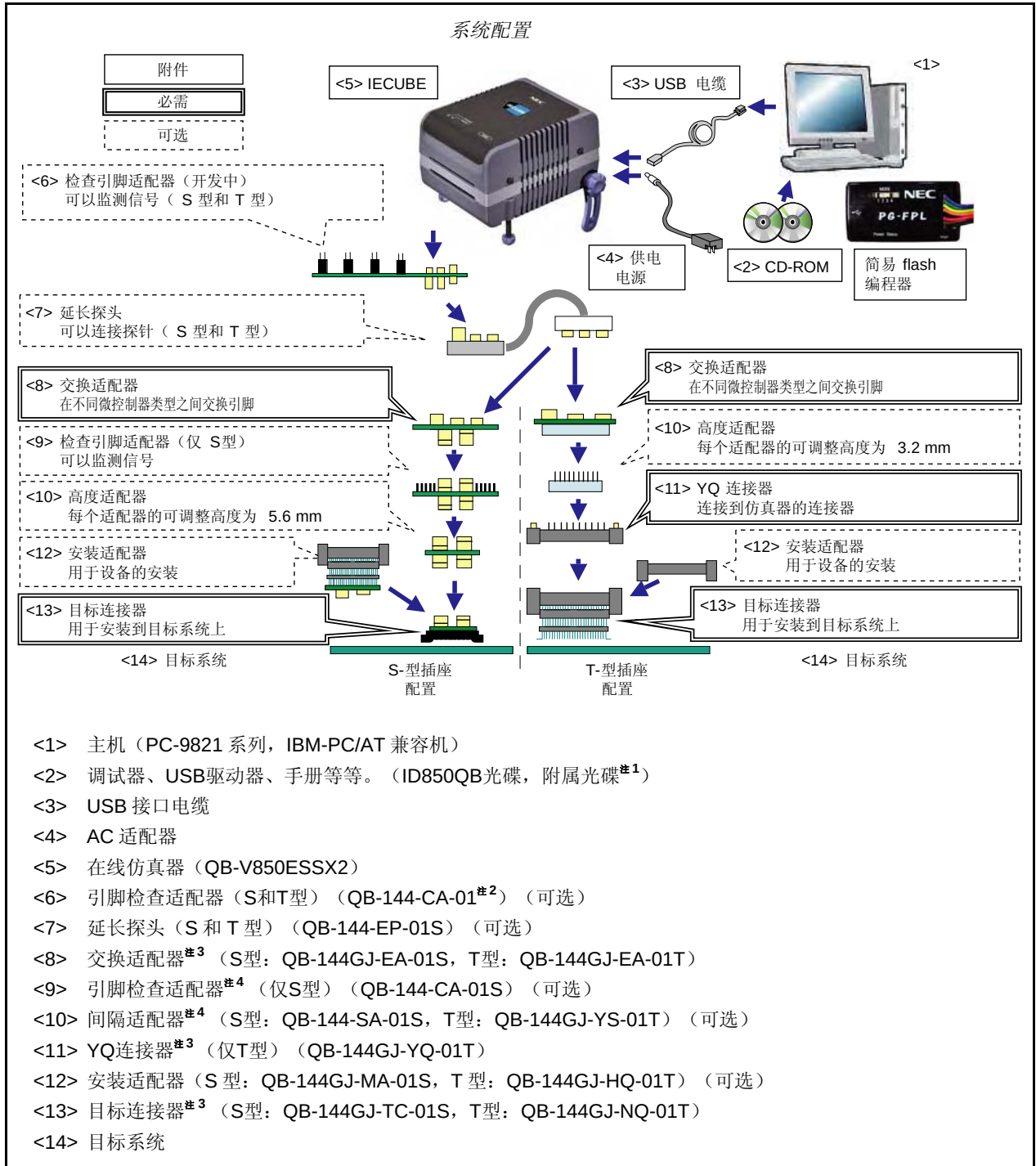


图 A-2. 系统配置（使用 QB-V850ESSX2 时）（2/2）

- 注
1. 从 NEC 电子的网站上下载设备文件。
<http://www.necel.com/micro/ods/eng/>
 2. 正在开发中。
 3. 根据订货码决定设备供应。
 - 当订购 QB-V850ESSX2-ZZZ 时：
不提供交换适配器和目标连接器
 - 当订购 QB-V850ESSX2-S144GJ 时
提供 QB-144GJ-EA-01S 和 QB-144GJ-TC-01S
 - 当订购 QB-V850ESSX2-T144GJ 时
提供 QB-144GJ-EA-01T、QB-144GJ-YQ-01T 和 QB-144GJ-NQ-01T
 4. 当使用 <9> 和 <10>二者时，不必考虑 <9> 和 <10> 之间次序。

<5> QB-V850ESSX2 ^注 在线仿真器	当开发使用 V850ES/JJ3 的应用系统时，使用该在线仿真器调试其硬件和软件。它支持集成调试器 ID850QB。该仿真器应与一个供电单元和仿真探头配合使用。使用 USB 接口电缆将该仿真器与主机连接。
<3> USB 接口电缆	连接主机和 QB-V850ESSX2 的电缆。
<4> AC 适配器	通过替换 AC 插头，可以支持 144V 至 240 V 供电。
<8> QB-144GJ-EA-01S QB-144GJ-EA-01T 交换适配器	执行引脚转换的适配器。
<9> QB-144-CA-01S 检查引脚适配器	使用示波器等波形检测的适配器
<10> QB-144-SA-01S QB-144GJ-YS-01T 间隔适配器	调整高度的适配器
<11> QB-144GJ-YQ-01T YQ 连接器	连接目标连接器和交换适配器的转换适配器
<12> QB-144GJ-MA-01S QB-144GJ-HQ-01T 安装适配器	使用插座安装 V850ES/JJ3 的适配器
<13> QB-144GJ-TC-01S QB-144GJ-NQ-01T 目标连接器	焊接到目标系统上的连接器。

注 QB-V850ESSX2 提供一个供电单元、USB 接口电缆和简易 flash 存储器编程器。它还提供集成调试器 ID850QB 作为控制软件。

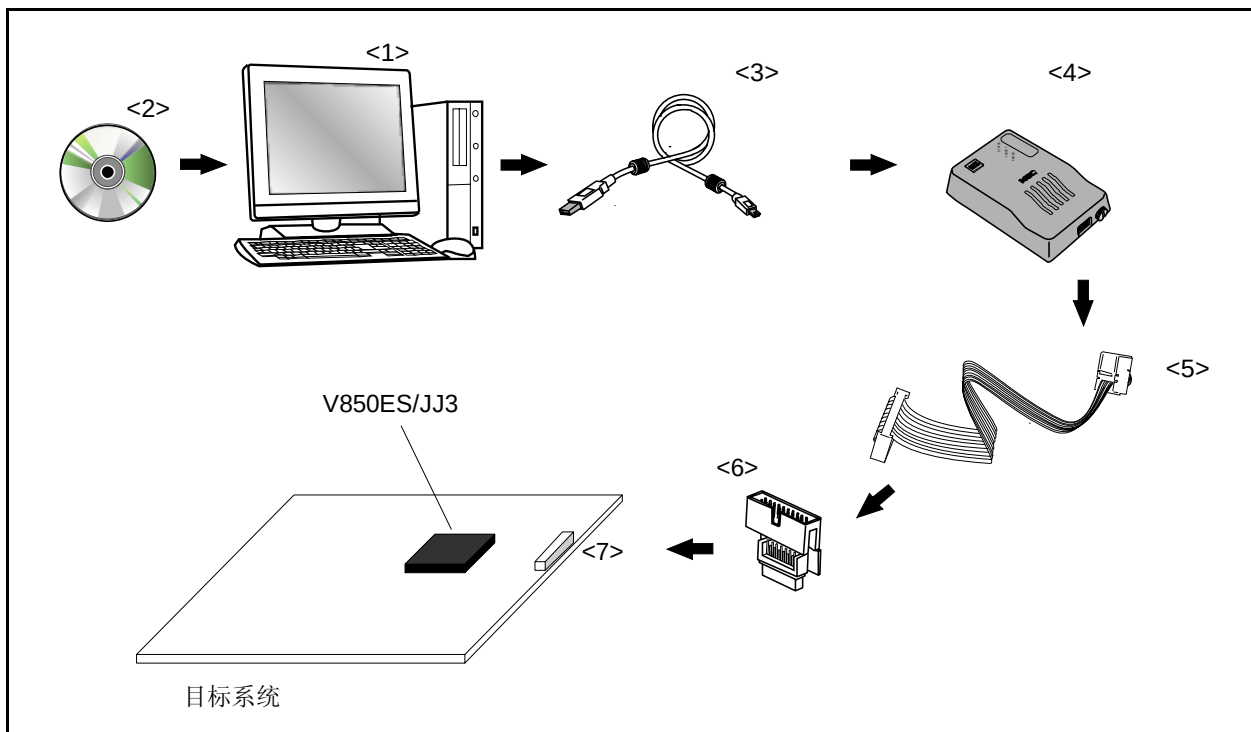
备注 尖括号中的数字对应于图 A-2 中的数字。

A.4.2 使用 MINICUBE QB-V850MINI 时

(1) 使用 MINICUBE 在线仿真

将 MINICUBE 连接到主机（PC-9821 系列，PC/AT 兼容机）时的系统配置如下所示。

图 A-3. 在线仿真系统配置



<1> 主机	配有 USB 接口的 PC。
<2> CD-ROM ^{※1}	CD-ROM 内有诸如集成调试器 ID850QB, N-Wire 检查器, 设备驱动和文档。它和 MINICUBE 一并提供。
<3> USB 接口电缆	连接主机和 MINICUBE 的 USB 电缆。它和 MINICUBE 一并提供。电缆长度大约为 2m。
<4> MINICUBE 在线仿真器	当开发使用 V850ES/JJ3 的应用系统时, 使用该在线调试仿真器调试其硬件和软件, 它支持集成调试器 ID850QB。
<5> OCD 电缆	连接 MINICUBE 和目标系统的电缆。它和 MINICUBE 一并提供。电缆长度大约为 20 cm。
<6> 连接器转换板 KEL 适配器	该转换板与 MINICUBE 一并提供。
<7> MINICUBE 连接器 KEL 连接器 ^{※2}	8830E-026-170S（与 MINICUBE 一并提供） 8830E-026-170L（单独售卖）

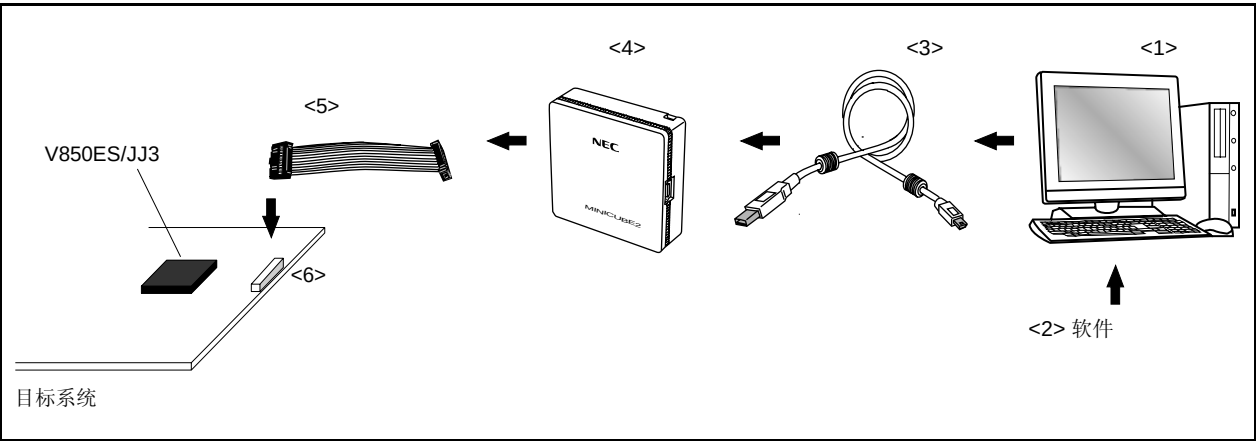
- 注
1. 从 NEC 电子网站上下载设备文件。
<http://www.necel.com/micro/ods/eng/index.html>
 2. KEL 公司的产品

备注 尖括号中的数字对应于图 A-3 中的数字。

A.4.3 使用 MINICUBE2 QB-MINI2 时

将 MINICUBE2 连接至主机（PC-9821 系列，PC/AT 兼容机）时的系统配置如下所示。

图 A-4. 在线仿真系统配置



<1> 主机	配有 USB 端口的 PC。
<2> 软件	集成调试器 ID850QB，设备文件等等。从 NEC 电子网站上下载设备文件。 http://www.necel.com/micro/ods/eng/
<3> USB 接口电缆	连接主机和 MINICUBE 的 USB 电缆。它与 I MINICUBE 一并提供。电缆长度大约为 2m。
<4> MINICUBE2 在线仿真器	当开发使用 V850ES/JJ3 的应用系统时，使用该在线调试仿真器调试其硬件和软件，它支持集成调试器 ID850QB。
<5> 16-引脚目标电缆	连接 MINICUBE2 和目标系统的电缆。它与 MINICUBE 一并提供。电缆长度大约为 15 cm。
<6> 目标连接器（单独售卖）	使用一个 2.54 mm 间距的 16-引脚通用连接器。

备注 尖括号中的数字对应于图 A-4 中的数字。

A.5 调试工具（软件）

SM850（开发中） 系统仿真器	<p>该仿真器配合 V850 微控制器使用。SM850 是基于 Windows 的软件。在主机上对目标系统仿真时，可以对 C 源文件和汇编源文件进行调试。通过使用 SM850，可以独立于硬件进行应用系统的逻辑验证和性能验证。因此，可以提供开发效率和改善软件质量。应该和设备文件配合使用。</p> <p>产品编号： μSxxxxSM703000</p>
ID850QB 集成调试器	<p>本调试器支持 V850 微控制器的在线仿真。ID850QB 为基于 Windows 操作系统的软件。它具有改进的 C-兼容调试功能，并且，使用一个集成窗口显示源程序追踪执行结果，以及相关追踪结果的源文件、反汇编显示、存储显示。它应与设备文件配合使用。</p> <p>产品编号： μSxxxx ID703000-QB（ID850QB）</p>

备注 xxxx型号随所用的主机和操作系统不同而异。

μ SxxxxSM703000

μ SxxxxID703000-QB

xxxx	主机	操作系统	提供形式
AB17	PC-9800 系列，	Windows（日文版）	CD-ROM
BB17	IBM PC/AT 兼容机	Windows（英文版）	

A.6 嵌入式软件

RX850, RX850 Pro 实时操作系统	RX850 和 RX850 Pro 为实时操作系统，遵从 μ ITRON 3.0 规范。 提供一个工具（配置器），用来产生多重信息表。 RX850 Pro 比 RX850 具有更多功能。
	型号： μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$ （RX850） μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$ （RX850 Pro）
RX-FS850 （文件系统）	这是一个 FAT 文件系统功能。 该文件系统支持 CD-ROM 文件系统功能。 该文件系统被实时操作系统 RX850 Pro 使用。

注意事项 欲购买 RX850 或 RX850 Pro 实时操作系统，首先填写购买申请表并签署授权许可文件。

备注 型号中的 xxxx 和 随使用的主机和操作系统的不同而异。

μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$

μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$

$\Delta\Delta\Delta\Delta$	产品概要	用于批量生产的最大数目
001	评估产品	不要用于量产
100K	批量生产产品	0.1 百万单位
001M		1 百万单位
010M		10 百万单位
S01	源程序	为量产的源程序

xxxx	主机	操作系统	提供形式
AB17	PC-9800 系列， IBM PC/AT 兼容机	Windows（日文版）	CD-ROM
BB17		Windows（英文版）	
3K17	SPARCstation	Solaris（Rel. 2.5.1）	

A.7 Flash 存储器写入工具

Flashpro IV (产品编号: PG-FP4) Flash 编程器	Flash 编程器专用于具备片上 Flash 存储器的微控制器编程。
QB-MINI2 (MINICUBE2)	具有编程功能的片上调试仿真器。
FA-144GJ-UEN-A Flash 存储器写入适配器	Flash 存储器写入适配器, 用来连接 Flashpro IV 等。(未连接)。
FA-70F3368GJ-UEN-RX Flash 存储器写入适配器	Flash 存储器写入适配器, 用来连接 Flashpro IV 等。(已连接)。

备注 FA-144GJ-UEN-A 和 FA-70F3368GJ-UEN-RX 是 Naito Densei Machida Mfg. Co., Ltd. 的产品。
TEL: +81-42-750-4172

附录 B V850ES/JJ3 和 V850ES/JJ2 之间的主要区别

V850ES/JJ3 和 V850ES/JJ2 之间的主要区别列举如下。详情参见各对应章节。

表 B-1. V850ES/JJ3 和 V850ES/JJ2 之间的主要区别 (1/2)

主要区别		V850ES/JJ3	V850ES/JJ2	参见:
BV _{DD} , BV _{SS} 引脚		修改为 EV _{DD} , EV _{SS} 引脚	提供	全文
引言: 最短指令执行时间		31.25 ns	50 ns	1.2
引脚功能: P10/ANO0, P11/ANO1 的引脚状态 (上电时)		Hi-Z	不确定	2.2
CPU 功能	内部 flash 存储器	384/512/768/1024 KB	128/256/384/512/640 KB	3.4.4 (1)
	内部 RAM	32/40/60 KB	12/24/32/40/48 KB	3.4.4 (2)
A/D 转换器: 转换期间采样时间的比例		8/26 个时钟	4/26 个时钟	13.5.2
复位功能: 内部系统复位释放之后的固件操作		无	提供 (参见手册 (U17714E) 的 22.3.4 (2))	—
低电压检测器 (LVI)	低电压检测中断(INTLVI) 的发生源	当供电电压下降或上升经过检测电压时	当供电电压下降到低于检测电压时	24.3 (1)
	低电压检测电平	2.85 至 3.05 V (2.95 V (TYP.))	2.85 至 3.15 V (3.0 V (TYP.))	24.3 (2)
	RAMS.RAMF 位设置条件	<ul style="list-style-type: none"> 检测到电压低于检测电压 由指令置位 	<ul style="list-style-type: none"> 检测到电压低于检测电压 由指令置位 发生 WDT2 和 CLM 引发的复位 在内部 RAM 访问期间, 发生 RESET 引脚引发的复位 	24.3 (3)
CRC 功能		提供	无	第 25 章
稳压器: 向副振荡器提供时钟		供电电压 (V _{DD})	稳压器输出电压	26.1
Flash 存储器	块配置	块 0 至最末块: 每个 4 KB	块 0 至块 3: 每个 28 KB 块 4 至块 7: 每个 4 KB 块 8 至最末块: 每个 64 KB	27.2
	启动区域	64 KB	56 KB	
片上调试功能	和软件断点相关的复位注意事项	无	提供 (参见手册 (U17714E) 的 27.1.6 (3))	—

表 B-1. V850ES/JJ3 和 V850ES/JJ2 之间的主要区别 (2/2)

主要区别		V850ES/JJ3	V850ES/JJ2	参见:
电气参数	工作条件 (内部系统时钟频率)	$f_{xx} = 2.5$ 至 32 MHz	$f_{xx} = 2.5$ 至 20 MHz	第 29 章
	内部振荡器特性(输出频率)	220 kHz (TYP.) (最小值和最大值和 V850ES/JJ2 相等)	200 kHz (TYP.)	
	DC 特性 (供电电流)	存在附加的参数	—	
	Bus 时序	存在修改的参数	—	
	CSIB 时序	存在修改的参数	—	
	D/A 转换器 (输出阻抗)	6.42 k Ω	3.5 k Ω	
	LVI 电路特性 (检测电压)	2.85 至 3.05 V (2.95 V (TYP.))	2.85 至 3.15 V (3.0 V (TYP.))	
	RAM 保持检测 (响应时间)	3.0 ms (MAX.)	2.0 ms (MAX.)	
封装图		P144GJ-50-GAE	S144GJ-50-UEN	第 30 章
推荐焊接条件		待定	提供	—

附录 C 寄存器索引

(1/12)

符号	名称	单元	页码
ADA0CR0	A/D 转换结果寄存器 0	ADC	452
ADA0CR0H	A/D 转换结果寄存器 0H	ADC	452
ADA0CR1	A/D 转换结果寄存器 1	ADC	452
ADA0CR1H	A/D 转换结果寄存器 1H	ADC	452
ADA0CR2	A/D 转换结果寄存器 2	ADC	452
ADA0CR2H	A/D 转换结果寄存器 2H	ADC	452
ADA0CR3	A/D 转换结果寄存器 3	ADC	452
ADA0CR3H	A/D 转换结果寄存器 3H	ADC	452
ADA0CR4	A/D 转换结果寄存器 4	ADC	452
ADA0CR4H	A/D 转换结果寄存器 4H	ADC	452
ADA0CR5	A/D 转换结果寄存器 5	ADC	452
ADA0CR5H	A/D 转换结果寄存器 5H	ADC	452
ADA0CR6	A/D 转换结果寄存器 6	ADC	452
ADA0CR6H	A/D 转换结果寄存器 6H	ADC	452
ADA0CR7	A/D 转换结果寄存器 7	ADC	452
ADA0CR7H	A/D 转换结果寄存器 7H	ADC	452
ADA0CR8	A/D 转换结果寄存器 8	ADC	452
ADA0CR8H	A/D 转换结果寄存器 8H	ADC	452
ADA0CR9	A/D 转换结果寄存器 9	ADC	452
ADA0CR9H	A/D 转换结果寄存器 9H	ADC	452
ADA0CR10	A/D 转换结果寄存器 10	ADC	452
ADA0CR10H	A/D 转换结果寄存器 10H	ADC	452
ADA0CR11	A/D 转换结果寄存器 11	ADC	452
ADA0CR11H	A/D 转换结果寄存器 11H	ADC	452
ADA0CR12	A/D 转换结果寄存器 12	ADC	452
ADA0CR12H	A/D 转换结果寄存器 12H	ADC	452
ADA0CR13	A/D 转换结果寄存器 13	ADC	452
ADA0CR13H	A/D 转换结果寄存器 13H	ADC	452
ADA0CR14	A/D 转换结果寄存器 14	ADC	452
ADA0CR14H	A/D 转换结果寄存器 14H	ADC	452
ADA0CR15	A/D 转换结果寄存器 15	ADC	452
ADA0CR15H	A/D 转换结果寄存器 15H	ADC	452
ADA0M0	A/D 转换器模式寄存器 0	ADC	445
ADA0M1	A/D 转换器模式寄存器 1	ADC	447
ADA0M2	A/D 转换器模式寄存器 2	ADC	450
ADA0PFM	掉电比较模式寄存器	ADC	454
ADA0PFT	掉电比较阈值寄存器	ADC	455
ADA0S	A/D 转换器通道指定寄存器	ADC	451
ADIC	中断控制寄存器	INTC	686
AWC	地址等待控制寄存器	BCU	199
BCC	总线周期控制寄存器	BCU	200
BSC	总线宽度配置寄存器	BCU	188

(2/12)

符号	名称	单元	页码
CB0CTL0	CSIB0 控制寄存器 0	CSI	523
CB0CTL1	CSIB0 控制寄存器 1	CSI	526
CB0CTL2	CSIB0 控制寄存器 2	CSI	527
CB0RIC	中断控制寄存器	INTC	686
CB0RX	CSIB0 接收数据寄存器	CSI	522
CB0RXL	CSIB0 接收数据寄存器 L	CSI	522
CB0STR	CSIB0 状态寄存器	CSI	529
CB0TIC	中断控制寄存器	INTC	686
CB0TX	CSIB0 发送数据寄存器	CSI	522
CB0TXL	CSIB0 发送数据寄存器 L	CSI	522
CB1CTL0	CSIB1 控制寄存器 0	CSI	523
CB1CTL1	CSIB1 控制寄存器 1	CSI	526
CB1CTL2	CSIB1 控制寄存器 2	CSI	527
CB1RIC	中断控制寄存器	INTC	686
CB1RX	CSIB1 接收数据寄存器	CSI	522
CB1RXL	CSIB1 接收数据寄存器 L	CSI	522
CB1STR	CSIB1 状态寄存器	CSI	529
CB1TIC	中断控制寄存器	INTC	686
CB1TX	CSIB1 发送数据寄存器	CSI	522
CB1TXL	CSIB1 发送数据寄存器 L	CSI	527
CB2CTL0	CSIB2 控制寄存器 0	CSI	523
CB2CTL1	CSIB2 控制寄存器 1	CSI	526
CB2CTL2	CSIB2 控制寄存器 2	CSI	527
CB2RIC	中断控制寄存器	INTC	686
CB2RX	CSIB2 接收数据寄存器	CSI	522
CB2RXL	CSIB2 接收数据寄存器 L	CSI	522
CB2STR	CSIB2 状态寄存器	CSI	529
CB2TIC	中断控制寄存器	INTC	686
CB2TX	CSIB2 发送数据寄存器	CSI	522
CB2TXL	CSIB2 发送数据寄存器 L	CSI	522
CB3CTL0	CSIB3 控制寄存器 0	CSI	523
CB3CTL1	CSIB3 控制寄存器 1	CSI	526
CB3CTL2	CSIB3 控制寄存器 2	CSI	527
CB3RIC	中断控制寄存器	INTC	686
CB3RX	CSIB3 接收数据寄存器	CSI	522
CB3RXL	CSIB3 接收数据寄存器 L	CSI	527
CB3STR	CSIB3 状态寄存器	CSI	529
CB3TIC	中断控制寄存器	INTC	686
CB3TX	CSIB3 发送数据寄存器	CSI	522
CB3TXL	CSIB3 发送数据寄存器 L	CSI	522
CB4CTL0	CSIB4 控制寄存器 0	CSI	523
CB4CTL1	CSIB4 控制寄存器 1	CSI	526
CB4CTL2	CSIB4 控制寄存器 2	CSI	527
CB4RIC	中断控制寄存器	INTC	686
CB4RX	CSIB4 接收数据寄存器	CSI	522

(3/12)

符号	名称	单元	页码
CB4RXL	CSIB4 接收数据寄存器 L	CSI	522
CB4STR	CSIB4 状态寄存器	CSI	529
CB4TIC	中断控制寄存器	INTC	686
CB4TX	CSIB4 发送数据寄存器	CSI	522
CB4TXL	CSIB4 发送数据寄存器 L	CSI	522
CB5CTL0	CSIB5 控制寄存器 0	CSI	523
CB5CTL1	CSIB5 控制寄存器 1	CSI	526
CB5CTL2	CSIB5 控制寄存器 2	CSI	527
CB5RIC	中断控制寄存器	INTC	686
CB5RX	CSIB5 接收数据寄存器	CSI	522
CB5RXL	CSIB5 接收数据寄存器 L	CSI	522
CB5STR	CSIB5 状态寄存器	CSI	529
CB5TIC	中断控制寄存器	INTC	686
CB5TX	CSIB5 发送数据寄存器	CSI	522
CB5TXL	CSIB5 发送数据寄存器 L	CSI	522
CCLS	CPU 工作时钟状态寄存器	CG	217
CKC	时钟控制寄存器	CLM	220
CLM	时钟监视器模式寄存器	CG	739
CRCD	CRC 数据寄存器	CRC	751
CRCIN	CRC 输入寄存器	CRC	751
CTBP	CALLT 基准指针	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	479
DA0CS1	D/A 转换值设置寄存器 1	DAC	479
DA0M	D/A 转换模式寄存器	DAC	478
DADC0	DMA 寻址控制寄存器 0	DMA	652
DADC1	DMA 寻址控制寄存器 1	DMA	652
DADC2	DMA 寻址控制寄存器 2	DMA	652
DADC3	DMA 寻址控制寄存器 3	DMA	652
DBC0	DMA 传输计数寄存器 0	DMA	651
DBC1	DMA 传输计数寄存器 1	DMA	651
DBC2	DMA 传输计数寄存器 2	DMA	651
DBC3	DMA 传输计数寄存器 3	DMA	651
DBPC	异常/调试陷阱状态保存寄存器	CPU	54
DBPSW	异常/调试陷阱状态保存寄存器	CPU	54
DCHC0	DMA 通道控制寄存器 0	DMA	653
DCHC1	DMA 通道控制寄存器 1	DMA	653
DCHC2	DMA 通道控制寄存器 2	DMA	653
DCHC3	DMA 通道控制寄存器 3	DMA	653
DDA0H	DMA 目的地址寄存器 0H	DMA	650
DDA0L	DMA 目的地址寄存器 0L	DMA	650
DDA1H	DMA 目的地址寄存器 1H	DMA	650
DDA1L	DMA 目的地址寄存器 1L	DMA	650
DDA2H	DMA 目的地址寄存器 2H	DMA	650

符号	名称	单元	页码
DDA2L	DMA 目的地址寄存器 2L	DMA	650
DDA3H	DMA 目的地址寄存器 3H	DMA	650
DDA3L	DMA 目的地址寄存器 3L	DMA	650
DMAIC0	中断控制寄存器	INTC	686
DMAIC1	中断控制寄存器	INTC	686
DMAIC2	中断控制寄存器	INTC	686
DMAIC3	中断控制寄存器	INTC	686
DSA0H	DMA 源地址寄存器 0H	DMA	649
DSA0L	DMA 源地址寄存器 0L	DMA	649
DSA1H	DMA 源地址寄存器 1H	DMA	649
DSA1L	DMA 源地址寄存器 1L	DMA	649
DSA2H	DMA 源地址寄存器 2H	DMA	649
DSA2L	DMA 源地址寄存器 2L	DMA	649
DSA3H	DMA 源地址寄存器 3H	DMA	649
DSA3L	DMA 源地址寄存器 3L	DMA	649
DTFR0	DMA 触发因素寄存器 0	DMA	654
DTFR1	DMA 触发因素寄存器 1	DMA	654
DTFR2	DMA 触发因素寄存器 2	DMA	654
DTFR3	DMA 触发因素寄存器 3	DMA	654
DWC0	数据等待控制寄存器 0	BCU	196
ECR	中断资源寄存器	CPU	51
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	187
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	590
IIC1	IIC 移位寄存器 1	I ² C	590
IIC2	IIC 移位寄存器 2	I ² C	590
IICC0	IIC 控制寄存器 0	I ² C	576
IICC1	IIC 控制寄存器 1	I ² C	576
IICC2	IIC 控制寄存器 2	I ² C	576
IICCL0	IIC 时钟选择寄存器 0	I ² C	586
IICCL1	IIC 时钟选择寄存器 1	I ² C	586
IICCL2	IIC 时钟选择寄存器 2	I ² C	586
IICF0	IIC 标志寄存器 0	I ² C	584
IICF1	IIC 标志寄存器 1	I ² C	584
IICF2	IIC 标志寄存器 2	I ² C	584
IICIC0	中断控制寄存器	INTC	686
IICIC1	中断控制寄存器	INTC	686
IICIC2	中断控制寄存器	INTC	686
IICS0	IIC 状态寄存器 0	I ² C	581
IICS1	IIC 状态寄存器 1	I ² C	581
IICS2	IIC 状态寄存器 2	I ² C	581
IICX0	IIC 功能扩展寄存器 0	I ² C	587

(5/12)

符号	名称	单元	页码
IICX1	IIC 功能扩展寄存器 1	I ² C	587
IICX2	IIC 功能扩展寄存器 2	I ² C	587
IMR0	中断屏蔽寄存器 0	INTC	686
IMR0H	中断屏蔽寄存器 0H	INTC	686
IMR0L	中断屏蔽寄存器 0L	INTC	686
IMR1	中断屏蔽寄存器 1	INTC	686
IMR1H	中断屏蔽寄存器 1H	INTC	686
IMR1L	中断屏蔽寄存器 1L	INTC	686
IMR2	中断屏蔽寄存器 2	INTC	686
IMR2H	中断屏蔽寄存器 2H	INTC	686
IMR2L	中断屏蔽寄存器 2L	INTC	686
IMR3	中断屏蔽寄存器 3	INTC	686
IMR3H	中断屏蔽寄存器 3H	INTC	686
IMR3L	中断屏蔽寄存器 3L	INTC	686
IMR4	中断屏蔽寄存器 4	INTC	686
IMR4H	中断屏蔽寄存器 4H	INTC	686
IMR4L	中断屏蔽寄存器 4L	INTC	686
INTF0	外部下降沿指定寄存器 0	INTC	700
INTF3	外部下降沿指定寄存器 3	INTC	701
INTF8	外部下降沿指定寄存器 8	INTC	702
INTF9H	外部下降沿指定寄存器 9H	INTC	703
INTR0	外部上升沿指定寄存器 0	INTC	700
INTR3	外部上升沿指定寄存器 3	INTC	701
INTR8	外部上升沿指定寄存器 8	INTC	702
INTR9H	外部上升沿指定寄存器 9H	INTC	703
ISPR	正在服务的中断优先级寄存器	INTC	690
KRIC	中断控制寄存器	INTC	686
KRM	按键返回模式寄存器	KR	708
LOCKR	锁定寄存器	CG	221
LVIIC	中断控制寄存器	INTC	686
LVIM	低电压检测寄存器	LVI	744
LVIS	低电压检测电平选择寄存器	LVI	745
NFC	噪声消除控制寄存器	INTC	704
OCDM	片上调试模式寄存器	DCU	785
OCKS0	IIC 分频时钟选择寄存器 0	I ² C	590
OCKS1	IIC 分频时钟选择寄存器 1	I ² C	590
OSTS	振荡稳定时间选择寄存器	WDT	713
P0	端口 0 寄存器	端口	93
P1	端口 1 寄存器	端口	97
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	99
P4	端口 4 寄存器	端口	104
P5	端口 5 寄存器	端口	106
P6	端口 6 寄存器	端口	111

符号	名称	单元	页码
P6H	端口 6 寄存器 H	端口	111
P6L	端口 6 寄存器 L	端口	111
P7H	端口 7 寄存器 H	端口	115
P7L	端口 7 寄存器 L	端口	115
P8	端口 8 寄存器	端口	116
P9	端口 9 寄存器	端口	119
P9H	端口 9 寄存器 H	端口	119
P9L	端口 9 寄存器 L	端口	119
PC	程序计数器	CPU	48
PCC	处理器时钟控制寄存器	CG	213
PCD	CD 端口寄存器	端口	126
PCM	CM 端口寄存器	端口	127
PCS	CS 端口寄存器	端口	129
PCT	CT 端口寄存器	端口	131
PDH	DH 端口寄存器	端口	133
PDL	DL 端口寄存器	端口	136
PDLH	DL 端口寄存器 H	端口	136
PDLL	DL 端口寄存器 L	端口	136
PEMU1	外围仿真寄存器 1	CPU	749
PF0	端口 0 功能寄存器	端口	96
PF3	端口 3 功能寄存器	端口	103
PF3H	端口 3 功能寄存器 H	端口	103
PF3L	端口 3 功能寄存器 L	端口	103
PF4	端口 4 功能寄存器	端口	105
PF5	端口 5 功能寄存器	端口	109
PF6	端口 6 功能寄存器	端口	113
PF6H	端口 6 功能寄存器 H	端口	113
PF6L	端口 6 功能寄存器 L	端口	113
PF8	端口 8 功能寄存器	端口	117
PF9	端口 9 功能寄存器	端口	125
PF9H	端口 9 功能寄存器 H	端口	125
PF9L	端口 9 功能寄存器 L	端口	125
PFC0	端口 0 功能控制寄存器	端口	96
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	101
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	107
PFC6H	端口 6 功能控制寄存器 H	端口	113
PFC9	端口 9 功能控制寄存器	端口	122
PFC9H	端口 9 功能控制寄存器 H	端口	122
PFC9L	端口 9 功能控制寄存器 L	端口	122
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	122

(7/12)

符号	名称	单元	页码
PFCE9H	端口 9 功能控制扩展寄存器 H	端口	122
PFCE9L	端口 9 功能控制扩展寄存器 L	端口	122
PIC0	中断控制寄存器	INTC	686
PIC1	中断控制寄存器	INTC	686
PIC2	中断控制寄存器	INTC	686
PIC3	中断控制寄存器	INTC	686
PIC4	中断控制寄存器	INTC	686
PIC5	中断控制寄存器	INTC	686
PIC6	中断控制寄存器	INTC	686
PIC7	中断控制寄存器	INTC	686
PIC8	中断控制寄存器	INTC	686
PLLCTL	PLL 控制寄存器	CG	219
PLLS	PLL 入锁时间指定寄存器	CG	222
PM0	端口 0 模式寄存器	端口	94
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	106
PM6	端口 6 模式寄存器	端口	111
PM6H	端口 6 模式寄存器 H	端口	111
PM6L	端口 6 模式寄存器 L	端口	111
PM7H	端口 7 模式寄存器 H	端口	115
PM7L	端口 7 模式寄存器 L	端口	115
PM8	端口 8 模式寄存器	端口	116
PM9	端口 9 模式寄存器	端口	119
PM9H	端口 9 模式寄存器 H	端口	119
PM9L	端口 9 模式寄存器 L	端口	119
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
PMC6	端口 6 模式控制寄存器	端口	112
PMC6H	端口 6 模式控制寄存器 H	端口	112
PMC6L	端口 6 模式控制寄存器 L	端口	112
PMC8	端口 8 模式控制寄存器	端口	117
PMC9	端口 9 模式控制寄存器	端口	120
PMC9H	端口 9 模式控制寄存器 H	端口	120
PMC9L	端口 9 模式控制寄存器 L	端口	120
PMCCM	CM 端口模式控制寄存器	端口	128
PMCCS	CS 端口模式控制寄存器	端口	130
PMCCCT	CT 端口模式控制寄存器	端口	132

符号	名称	单元	页码
PMCD	CD 端口模式寄存器	端口	126
PMCDH	DH 端口模式控制寄存器	端口	134
PMCDL	DL 端口模式控制寄存器	端口	137
PMCDLH	DL 端口模式控制寄存器 H	端口	137
PMCDLL	DL 端口模式控制寄存器 L	端口	137
PMCM	CM 端口模式寄存器	端口	127
PMCS	CS 端口模式寄存器	端口	129
PMCT	CT 端口模式寄存器	端口	131
PMDH	DH 端口模式寄存器	端口	133
PMDL	DL 端口模式寄存器	端口	136
PMDLH	DL 端口模式寄存器 H	端口	136
PMDLL	DL 端口模式寄存器 L	端口	136
PRCMD	命令寄存器	CPU	82
PRSCM0	预分频器比较寄存器 0	WT	423
PRSCM1	预分频器比较寄存器 1	CSI	566
PRSCM2	预分频器比较寄存器 2	CSI	566
PRSCM3	预分频器比较寄存器 3	CSI	566
PRSM0	预分频模式寄存器 0	WT	422
PRSM1	预分频模式寄存器 1	CSI	565
PRSM2	预分频模式寄存器 2	CSI	565
PRSM3	预分频模式寄存器 3	CSI	565
PSC	省电控制寄存器	CG	711
PSMR	省电模式寄存器	CG	712
PSW	程序状态字	CPU	52
r0 至 r31	通用寄存器	CPU	48
RAMS	内部 RAM 数据状态寄存器	CG	745
RCM	内部振荡模式寄存器	CG	217
RESF	复位源标志寄存器	Reset	730
RTBH0	实时输出缓冲寄存器 0H	RTP	436
RTBH1	实时输出缓冲寄存器 1H	RTP	436
RTBL0	实时输出缓冲寄存器 0L	RTP	436
RTBL1	实时输出缓冲寄存器 1L	RTP	436
RTPC0	实时输出端口控制寄存器 0	RTP	438
RTPC1	实时输出端口控制寄存器 1	RTP	438
RTPM0	实时输出端口模式寄存器 0	RTP	437
RTPM1	实时输出端口模式寄存器 1	RTP	437
SELCNT0	选择器操作控制寄存器 0	定时器	310
SVA0	从地址寄存器 0	I ² C	591
SVA1	从地址寄存器 1	I ² C	591
SVA2	从地址寄存器 2	I ² C	591
SYS	系统状态寄存器	CPU	83
TM0CMP0	TMM0 比较寄存器 0	定时器	412
TM0CTL0	TMM0 控制寄存器 0	定时器	413
TM0EQIC0	中断控制寄存器	INTC	686
TP0CCIC0	中断控制寄存器	INTC	686

(9/12)

符号	名称	单元	页码
TP0CCIC1	中断控制寄存器	INTC	686
TP0CCR0	TMP0 捕获/比较寄存器 0	定时器	233
TP0CCR1	TMP0 捕获/比较寄存器 1	定时器	235
TP0CNT	TMP0 计数器读取缓冲寄存器	定时器	237
TP0CTL0	TMP0 控制寄存器 0	定时器	227
TP0CTL1	TMP0 控制寄存器 1	定时器	227
TP0IOC0	TMP0 I/O 控制寄存器 0	定时器	229
TP0IOC1	TMP0 I/O 控制寄存器 1	定时器	230
TP0IOC2	TMP0 I/O 控制寄存器 2	定时器	231
TP0OPT0	TMP0 选项寄存器 0	定时器	232
TP0OVIC	中断控制寄存器	INTC	686
TP1CCIC0	中断控制寄存器	INTC	686
TP1CCIC1	中断控制寄存器	INTC	686
TP1CCR0	TMP1 捕获/比较寄存器 0	定时器	233
TP1CCR1	TMP1 捕获/比较寄存器 1	定时器	235
TP1CNT	TMP1 计数器读取缓冲寄存器	定时器	237
TP1CTL0	TMP1 控制寄存器 0	定时器	227
TP1CTL1	TMP1 控制寄存器 1	定时器	227
TP1IOC0	TMP1 I/O 控制寄存器 0	定时器	229
TP1IOC1	TMP1 I/O 控制寄存器 1	定时器	230
TP1IOC2	TMP1 I/O 控制寄存器 2	定时器	231
TP1OPT0	TMP1 选项寄存器 0	定时器	232
TP1OVIC	中断控制寄存器	INTC	686
TP2CCIC0	中断控制寄存器	INTC	686
TP2CCIC1	中断控制寄存器	INTC	686
TP2CCR0	TMP2 捕获/比较寄存器 0	定时器	233
TP2CCR1	TMP2 捕获/比较寄存器 1	定时器	235
TP2CNT	TMP2 计数器读取缓冲寄存器	定时器	237
TP2CTL0	TMP2 控制寄存器 0	定时器	227
TP2CTL1	TMP2 控制寄存器 1	定时器	227
TP2IOC0	TMP2 I/O 控制寄存器 0	定时器	229
TP2IOC1	TMP2 I/O 控制寄存器 1	定时器	230
TP2IOC2	TMP2 I/O 控制寄存器 2	定时器	231
TP2OPT0	TMP2 选项寄存器 0	定时器	232
TP2OVIC	中断控制寄存器	INTC	686
TP3CCIC0	中断控制寄存器	INTC	686
TP3CCIC1	中断控制寄存器	INTC	686
TP3CCR0	TMP3 捕获/比较寄存器 0	定时器	233
TP3CCR1	TMP3 捕获/比较寄存器 1	定时器	235
TP3CNT	TMP3 计数器读取缓冲寄存器	定时器	237
TP3CTL0	TMP3 控制寄存器 0	定时器	227
TP3CTL1	TMP3 控制寄存器 1	定时器	227
TP3IOC0	TMP3 I/O 控制寄存器 0	定时器	229
TP3IOC1	TMP3 I/O 控制寄存器 1	定时器	230
TP3IOC2	TMP3 I/O 控制寄存器 2	定时器	231

(10/12)

符号	名称	单元	页码
TP3OPT0	TMP3 选项寄存器 0	定时器	232
TP3OVIC	中断控制寄存器	INTC	686
TP4CCIC0	中断控制寄存器	INTC	686
TP4CCIC1	中断控制寄存器	INTC	686
TP4CCR0	TMP4 捕获/比较寄存器 0	定时器	233
TP4CCR1	TMP4 捕获/比较寄存器 1	定时器	235
TP4CNT	TMP4 计数器读取缓冲寄存器	定时器	237
TP4CTL0	TMP4 控制寄存器 0	定时器	227
TP4CTL1	TMP4 控制寄存器 1	定时器	227
TP4IOC0	TMP4 I/O 控制寄存器 0	定时器	229
TP4IOC1	TMP4 I/O 控制寄存器 1	定时器	230
TP4IOC2	TMP4 I/O 控制寄存器 2	定时器	231
TP4OPT0	TMP4 选项寄存器 0	定时器	232
TP4OVIC	中断控制寄存器	INTC	686
TP5CCIC0	中断控制寄存器	INTC	686
TP5CCIC1	中断控制寄存器	INTC	686
TP5CCR0	TMP5 捕获/比较寄存器 0	定时器	233
TP5CCR1	TMP5 捕获/比较寄存器 1	定时器	235
TP5CNT	TMP5 计数器读取缓冲寄存器	定时器	237
TP5CTL0	TMP5 控制寄存器 0	定时器	227
TP5CTL1	TMP5 控制寄存器 1	定时器	227
TP5IOC0	TMP5 I/O 控制寄存器 0	定时器	229
TP5IOC1	TMP5 I/O 控制寄存器 1	定时器	230
TP5IOC2	TMP5 I/O 控制寄存器 2	定时器	231
TP5OPT0	TMP5 选项寄存器 0	定时器	232
TP5OVIC	中断控制寄存器	INTC	686
TP6CCIC0	中断控制寄存器	INTC	686
TP6CCIC1	中断控制寄存器	INTC	686
TP6CCR0	TMP6 捕获/比较寄存器 0	定时器	233
TP6CCR1	TMP6 捕获/比较寄存器 1	定时器	235
TP6CNT	TMP6 计数器读取缓冲寄存器	定时器	237
TP6CTL0	TMP6 控制寄存器 0	定时器	227
TP6CTL1	TMP6 控制寄存器 1	定时器	227
TP6IOC0	TMP6 I/O 控制寄存器 0	定时器	229
TP6IOC1	TMP6 I/O 控制寄存器 1	定时器	230
TP6IOC2	TMP6 I/O 控制寄存器 2	定时器	231
TP6OPT0	TMP6 选项寄存器 0	定时器	232
TP6OVIC	中断控制寄存器	INTC	686
TP7CCIC0	中断控制寄存器	INTC	686
TP7CCIC1	中断控制寄存器	INTC	686
TP7CCR0	TMP7 捕获/比较寄存器 0	定时器	233
TP7CCR1	TMP7 捕获/比较寄存器 1	定时器	235
TP7CNT	TMP7 计数器读取缓冲寄存器	定时器	237
TP7CTL0	TMP7 控制寄存器 0	定时器	227
TP7CTL1	TMP7 控制寄存器 1	定时器	227

(11/12)

符号	名称	单元	页码
TP7IOC0	TMP7 I/O 控制寄存器 0	定时器	229
TP7IOC1	TMP7 I/O 控制寄存器 1	定时器	230
TP7IOC2	TMP7 I/O 控制寄存器 2	定时器	231
TP7OPT0	TMP7 选项寄存器 0	定时器	232
TP7OVIC	中断控制寄存器	INTC	686
TP8CCIC0	中断控制寄存器	INTC	686
TP8CCIC1	中断控制寄存器	INTC	686
TP8CCR0	TMP8 捕获/比较寄存器 0	定时器	233
TP8CCR1	TMP8 捕获/比较寄存器 1	定时器	235
TP8CNT	TMP8 计数器读取缓冲寄存器	定时器	237
TP8CTL0	TMP8 控制寄存器 0	定时器	227
TP8CTL1	TMP8 控制寄存器 1	定时器	227
TP8IOC0	TMP8 I/O 控制寄存器 0	定时器	229
TP8IOC1	TMP8 I/O 控制寄存器 1	定时器	230
TP8IOC2	TMP8 I/O 控制寄存器 2	定时器	231
TP8OPT0	TMP8 选项寄存器 0	定时器	232
TP8OVIC	中断控制寄存器	INTC	686
TQ0CCIC0	中断控制寄存器	INTC	686
TQ0CCIC1	中断控制寄存器	INTC	686
TQ0CCIC2	中断控制寄存器	INTC	686
TQ0CCIC3	中断控制寄存器	INTC	686
TQ0CCR0	TMQ0 捕获/比较寄存器 0	定时器	322
TQ0CCR1	TMQ0 捕获/比较寄存器 1	定时器	324
TQ0CCR2	TMQ0 捕获/比较寄存器 2	定时器	326
TQ0CCR3	TMQ0 捕获/比较寄存器 3	定时器	328
TQ0CNT	TMQ0 计数器读取缓冲寄存器	定时器	330
TQ0CTL0	TMQ0 控制寄存器 0	定时器	316
TQ0CTL1	TMQ0 控制寄存器 1	定时器	317
TQ0IOC0	TMQ0 I/O 控制寄存器 0	定时器	318
TQ0IOC1	TMQ0 I/O 控制寄存器 1	定时器	319
TQ0IOC2	TMQ0 I/O 控制寄存器 2	定时器	320
TQ0OPT0	TMQ0 选项寄存器 0	定时器	321
TQ0OVIC	中断控制寄存器	INTC	686
UA0CTL0	UARTA0 控制寄存器 0	UART	488
UA0CTL1	UARTA0 控制寄存器 1	UART	511
UA0CTL2	UARTA0 控制寄存器 2	UART	512
UA0OPT0	UARTA0 选项控制寄存器 0	UART	490
UA0RIC	中断控制寄存器	INTC	686
UA0RX	UARTA0 接收数据寄存器	UART	493
UA0STR	UARTA0 状态寄存器	UART	491
UA0TIC	中断控制寄存器	INTC	686
UA0TX	UARTA0 发送数据寄存器	UART	493
UA1CTL0	UARTA1 控制寄存器 0	UART	488
UA1CTL1	UARTA1 控制寄存器 1	UART	511
UA1CTL2	UARTA1 控制寄存器 2	UART	512

(12/12)

符号	名称	单元	页码
UA1OPT0	UARTA1 选项控制寄存器 0	UART	490
UA1RIC	中断控制寄存器	INTC	686
UA1RX	UARTA1 接收数据寄存器	UART	493
UA1STR	UARTA1 状态寄存器	UART	491
UA1TIC	中断控制寄存器	INTC	686
UA1TX	UARTA1 发送数据寄存器	UART	493
UA2CTL0	UARTA2 控制寄存器 0	UART	488
UA2CTL1	UARTA2 控制寄存器 1	UART	511
UA2CTL2	UARTA2 控制寄存器 2	UART	512
UA2OPT0	UARTA2 选项控制寄存器 0	UART	490
UA2RIC	中断控制寄存器	INTC	686
UA2RX	UARTA2 接收数据寄存器	UART	493
UA2STR	UARTA2 状态寄存器	UART	491
UA2TIC	中断控制寄存器	INTC	686
UA2TX	UARTA2 发送数据寄存器	UART	493
UA3CTL0	UARTA3 控制寄存器 0	UART	488
UA3CTL1	UARTA3 控制寄存器 1	UART	511
UA3CTL2	UARTA3 控制寄存器 2	UART	512
UA3OPT0	UARTA3 选项控制寄存器 0	UART	490
UA3RIC	中断控制寄存器	INTC	686
UA3RX	UARTA3 接收数据寄存器	UART	493
UA3STR	UARTA3 状态寄存器	UART	491
UA3TIC	中断控制寄存器	INTC	686
UA3TX	UARTA3 发送数据寄存器	UART	493
VSWC	系统等待控制寄存器	CPU	84
WDTE	看门狗定时器使能寄存器	WDT	432
WDTM2	看门狗定时器模式寄存器 2	WDT	431, 691
WTIC	中断控制寄存器	INTC	686
WTIIC	中断控制寄存器	INTC	686
WTM	钟表定时器操作模式寄存器	WT	424

附录 D 指令集列表

D.1 习惯用法

(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，在某指令里也用作源寄存器。
reg3	通用寄存器：主要用作存储除法结果的余数，以及乘法结果的高 32 位。
bit#3	指明位号的 3 位数据。
immX	X 位立即数。
dispX	X 位置换数据。
regID	系统寄存器号。
vector	用来指明陷阱向量的 5 位数据 (00H 至 1FH)。
cccc	用来显示状态码的 4 位数据。
sp	堆栈指针 (r3)。
ep	元指针 (r30)。
listX	X 项寄存器列表。

(2) 用于描述操作码的寄存器符号

寄存器符号	解释
R	指定 reg1 或 regID 的 1 位代码数据。
r	指定 reg2 的 1 位代码数据。
w	指定 reg3 的 1 位代码数据。
d	1 位置换数据。
l	1 位立即数 (指示立即数的高位)。
i	1 位立即数。
cccc	显示条件码的 4 位数据。
CCCC	显示 Bcond 指令条件码的 4 位数据。
bbb	指定位号的 3 位数据。
L	在寄存器列表中指示编程寄存器的 1 位数据。

(3) 用于操作的寄存器符号

寄存器符号	解释
←	为之输入
GR []	通用寄存器
SR []	系统寄存器
zero-extend (n)	用零扩展 n 直至满字长。
sign-extend (n)	用符号扩展 n 直至满字长。
load-memory (a, b)	从地址 a 起读取存储长度为 b 的数据。
store-memory (a, b, c)	向地址 c 起存储长度为 a 的字节写入数据 b。
load-memory-bit (a, b)	从地址 a 载入位 b。
store-memory-bit (a, b, c)	写入 c 至地址 a 的 b 位。
saturated (n)	执行 n 的饱和运算 (n 为 2 的补码)。 如此的话, 结果如下处理: n ≥ 7FFFFFFFH, 令其为 7FFFFFFFH。 n ≤ 80000000H, 令其为 80000000H。
result	反映标志的结果。
Byte	字节 (8 位)
Halfword	半字 (16 位)
Word	字 (32 位)
+	加
−	减
	位并拼
×	乘
÷	除
%	除法结果的余数
AND	逻辑乘
OR	逻辑加
XOR	逻辑异或
NOT	逻辑非
logically shift left by	逻辑左移
logically shift right by	逻辑右移
arithmetically shift right by	算术右移

(4) 用于执行时钟的寄存器符号

寄存器符号	解释
i	如果执行第一条指令后紧接着执行另一条指令 (流式执行)。
r	如果执行第一条指令后紧接着重复执行相同指令 (重复执行)。
l	如果紧接之后的指令执行使用当前指令执行结果 (等待执行)。

(5) 用于标志操作的寄存器符号

标识符	说明
(Blank)	无改变。
0	清为 0。
X	根据结果置位或清 0。
R	恢复先前的保存值。

(6) 条件码

条件码 (cccc)	条件公式	说明
0 0 0 0	$OV = 1$	溢出
1 0 0 0	$OV = 0$	无溢出
0 0 0 1	$CY = 1$	进位 小(相比小)
1 0 0 1	$CY = 0$	无进位 不小(相比大于或等于)
0 0 1 0	$Z = 1$	0
1 0 1 0	$Z = 0$	非 0
0 0 1 1	$(CY \text{ or } Z) = 1$	不高(相比小于或等于 l)
1 0 1 1	$(CY \text{ or } Z) = 0$	高(相比大于)
0 1 0 0	$S = 1$	负
1 1 0 0	$S = 0$	正
0 1 0 1	—	恒为(无条件的)
1 1 0 1	$SAT = 1$	饱和
0 1 1 0	$(S \text{ xor } OV) = 1$	小于有符号数。
1 1 1 0	$(S \text{ xor } OV) = 0$	大于或等于有符号数。
0 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 1$	小于或等于有符号数。
1 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 0$	大于有符号数。

D.2 指令集 (按字母先后次序)

(1/6)

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]	1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiii	GR[reg2]←GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]	1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)	1	1	1		0	×	×	
Bcond	disp9	ddddd1011dddcccc 注 1	如果条件满足 则: PC←PC+sign-extend(disp9)	2 注 2	2 注 2	2 注 2					
			如果条件不满足	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) GR[reg2] (31 : 24) GR[reg2] (7 : 0) GR[reg2] (15 : 8)	1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) GR[reg2] (15 : 8) GR [reg2] (23 : 16) GR[reg2] (31 : 24)	1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adrr,Halfword))	4	4	4					
CLR1	bit#3,disp16[reg1]	10bbb111110RRRRR dddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adrr,bit#3)) Store-memory-bit(adrr,bit#3,0)	3 注 3	3 注 3	3 注 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adrr,reg2)) Store-memory-bit(adrr,reg2,0)	3 注 3	3 注 3	3 注 3				×	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiii wwwwww011000cccc0	如果条件满足 则 GR[reg3]←sign-extended(imm5) 否则 GR[reg3]←GR[reg2]	1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRRR wwwwww011001cccc0	如果条件满足 则 GR[reg3]←GR[reg1] 否则 GR[reg3]←GR[reg2]	1	1	1					
CMP	reg1,reg2	rrrrr001111RRRRR	result←GR[reg2]−GR[reg1]	1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010011iiii	result←GR[reg2]−sign-extend(imm5)	1	1	1	×	×	×	×	
CTRET		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW	3	3	3	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW	3	3	3	R	R	R	R	R

(2/6)

助记符	操作数	操作码	操作	执行时钟			标志				
				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	0000011001iiiiL LLLLLLLLLLLL00000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 重复以上两个步骤直到表 12 中所有寄存器被载入。	n+1 注 4	n+1 注 4	n+1 注 4					
	imm5,list12,[reg1]	0000011001iiiiL LLLLLLLLLLLLRRRRR 注 5	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 重复以上两个步骤直到表 12 中所有寄存器载入 PC←GR[reg1]	n+3 注 4	n+3 注 4	n+3 注 4					
DIV	reg1,reg2,reg3	rrrrr11111RRRRR wwwww01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVH	reg1,reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1]注 6	35	35	35		×	×	×	
	reg1,reg2,reg3	rrrrr11111RRRRR wwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1]注 6 GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	rrrrr11111RRRRR wwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1]注 6 GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	rrrrr11111RRRRR wwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	停止	1	1	1					
HSW	reg2,reg3	rrrrr11111100000 wwwww01101000100	GR[reg3]←GR[reg2](15 : 0) GR[reg2] (31 : 16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110ddddd dddddddddddddd0 注 7	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110ddddd dddddddddddddd0 注 7	PC←PC+sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	注 11					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR dddddddddddddd1 注 8, 10	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,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	rrrrr111001RRRRR dddddddddddddd0 注 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	注 11					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 注 12	SR[regID]←GR[reg2]	1	1	1					
			除非 regID = PSW regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR dddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	注 11					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR dddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)	1	1	注 11					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]	1	1	1					
	imm5,reg2	rrrrr010000iiii	GR[reg2]←sign-extend(imm5)	1	1	1					
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii iiiiiiiiiiiiiiii	GR[reg1]←imm32	2	2	2					
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 ∥ 0 ¹⁶)	1	1	1					
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000 注 14	GR[reg3] ∥ GR[reg2]←GR[reg2]xGR[reg1]	1	4	5					
	imm9,reg2,reg3	rrrrr111111iiii wwwww01001IIII00 注 13	GR[reg3] ∥ GR[reg2]←GR[reg2]xsign-extend(imm9)	1	4	5					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2] ⁶ xGR[reg1] ⁶	1	1	2					
	imm5,reg2	rrrrr010111iiii	GR[reg2]←GR[reg2] ⁶ xsign-extend(imm5)	1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] ⁶ ximm16	1	1	2					
MULU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010 注 14	GR[reg3] ∥ GR[reg2]←GR[reg2]xGR[reg1]	1	4	5					
	imm9,reg2,reg3	rrrrr111111iiii wwwww01001IIII10 注 13	GR[reg3] ∥ GR[reg2]←GR[reg2]xzero-extend(imm9)	1	4	5					
NOP		0000000000000000	至少一个时钟周期内不作任何操作。	1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])	1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb11110RRRRR dddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)	3 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	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiL LLLLLLLLLLLL00001	Store-memory(sp−4,GR[reg in list12],Word) sp←sp−4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 注4	n+1 注4	n+1 注4					
	list12,imm5, sp/imm ^{#15}	0000011110iiiiL LLLLLLLLLLLLff011 imm16/imm32 注 16	Store-memory(sp−4,GR[reg in list12],Word) sp←sp+4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend (imm5) ep←sp/imm	n+2 注4 注17	n+2 注4 注17	n+2 注4 注17					
RETI		000001111100000 0000000101000000	如果 PSW.EP=1 那么 PC ←EIPC PSW ←EIPSW 否则, 如果 PSW.NP=1 则 PC ←FEPC PSW ←FEPSW 否则 PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R
SAR	reg1,reg2	rrrrr11111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr111110cccc 0000001000000000	如果条件满足 则 GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H 否则 GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1					
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5,reg2	rrrrr010001iiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5))	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]−GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR 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 0000000000000000	如果条件满足 则 GR[reg2]←00000001H 否则 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]	00bbb11110RRRRR dddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 注3	3 注3	3 注3				×	
	reg2,[reg1]	rrrrr11111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 注3	3 注3	3 注3				×	
SHL	reg1,reg2	rrrrr11111RRRRR 0000000011100000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr11111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110dddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	注9					
SLD.BU	disp4[ep],reg2	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 18, 20	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	注9					
SLD.W	disp8[ep],reg2	rrrrr1010dddddd0 注 21	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	注9					
SST.B	reg2,disp7[ep]	rrrrr0111dddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001dddddd 注 19	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Halfword)	1	1	1					
SST.W	reg2,disp8[ep]	rrrrr1010dddddd1 注 21	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1					
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR dddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR dddddddddddddd0 注 8	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Halfword)	1	1	1					
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR dddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Word)	1	1	1					
STSR	regID,reg2	rrrrr11111RRRRR 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	00000000010RRRRR	adr←(PC+2) + (GR [reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr,Halfword)) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15 : 0))	1	1	1					
TRAP	vector	0 0 0 0 0 1 1 1 1 1 i i i i i 0000000100000000	EIPC ←PC+4 (Restored PC) EIPSW ←PSW ECR.EICC ←Interrupt code PSW.EP ←1 PSW.ID ←1 PC ←00000040H (when vector is 00H to 0FH) 00000050H (when vector is 10H to 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR ddddddddddddddddd	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 i i i i i i i i i i i i i i i i	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. 如果之前紧邻一条指令改写 PSW 的内容, 则为 3。
 3. 如果没有等待状态 (3 + 读取访问等待状态数)。
 4. n 为列表 12 中加载寄存器的总数。(根据等待状态数目, n 为列表 12 中加载寄存器的总数。如果 n = 0, 和 n = 1 时的操作相同)
 5. RRRRR: 除了 00000 之外。
 6. 只有低半字有效。
 7. dddddddddddddddddddd: disp22 的高 21 位。
 8. dddddddddddddddd: disp16 的高 15 位。
 9. 依照等待状态的数目(如果没有等待状态, 则为 1)。
 10. b: disp16 的位 0。
 11. 依照等待状态的数目(如果没有等待状态, 则为 2)。

- 注 12. 本指令中，为了助记符描述方便，使源寄存器为 `reg2`，而操作码位置上使用 `reg1` 寄存器。所以，助记符描述中寄存器的含义和其它指令有所不同。
- rrrrr = 指定 `reg1D`
RRRRR = 指定 `reg2`
13. `iiii`: `imm9` 的低 5 位。
IIII: `imm9` 的高 4 位。
14. 不要将为通用寄存器 `reg1` 和 `reg3` 指定同一寄存器。
15. `sp/imm`: 由次操作码的位 19 和位 20 指定。
16. `ff = 00`: 将 `sp` 载入 `ep`。
01: 将符号扩展的 16 位立即数(位 47 至 32)载入 `ep`。
10: 将经过 16 位逻辑左移的 16 位立即数 (位 47 至 32) 载入 `ep`。
11: 将 32 位立即数 (位 63 至 32) 载入 `ep`。
17. If `imm = imm32`, `n + 3` 时钟。
18. `rrrrr`: 非 00000。
19. `ddddddd`: `disp8` 的高 8 位。
20. `dddd`: `disp5` 的高 4 位。
21. `dddddd`: `disp8` 的高 6 位。

附录 E 注意事项列表

该附录列举了本文件中所述的注意事项。

表内的“分类（硬件/软件）”如下。

硬件： 微控制器内部/外部硬件的注意事项

软件： 软件的注意事项，例如寄存器设置或程序

(1/37)

章节	分类	功能	功能细节	注意事项	页码
第一章	硬件	引言	FLMD0	正常模式下，将该引脚与V _{SS} 相连接。	p. 23 □
			REGC	通过一个4.7 μF（推荐值）的电容器将REGC引脚连接到V _{SS} 。	p. 23 □
第二章	软件	引脚功能	P05	内置一个下拉电阻。通过清除OCDM.OCDM0位，可以断开该电阻的连接。	p. 30 □
	硬件		DDO	在片上调试模式下，强制输出高电平。	p. 35 □
	软件		KR0 至 KR7:	外部连接一个下拉电阻。	p. 36 □
			NMI	NMI引脚可复用为P02引脚功能。复位后，作为P02引脚功能。要使能NMI引脚，设置PMC0.PMC02 位为 1。NMI引脚的初始设置为“无边沿检测”。使用INTF0 和 INTR0寄存器选择NMI引脚的有效沿。	p. 36 □
	硬件		当开启电源时	当电源开启时，下列引脚可能会在复位期间暂时输出一个不确定的电平。 ● P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 引脚	p. 45 □
	第三章		软件	CPU 功能	EIPC 寄存器，EIPSW 寄存器，FEPC 寄存器，FEPSW 寄存器
EIPC, FEPC, CTPC 寄存器		即使EIPC或FEPC、或CTPC的第0位被LDSR指令设置为1，当中断服务程序执行之后通过RETI指令返回到主程序时，第0位被忽略（这是因为PC的第0位固定为0）。设置EIPC，FEPC和CTPC的值是偶数（第0位 = 0）。			p. 49 □
程序空间		由于地址03FFF000H 至 03FFFFFFH的4 KB范围是片上外围I/O区域，因此不能从这个区域取指。因此，要避免执行分支地址计算结果可能涉及该范围的操作。			p. 57 □
片上外围 I/O 区域		当按照字单位访问一个寄存器时，一个字区域按照低位和高位的顺序分两次半字访问，其中最低2位地址被忽略。			p. 64 □
		如果一个寄存器支持字节访问，使用半字单位访问时，读取该寄存器，则高8位数据不确定，数据被写入低8位。		p. 64 □	
		未被指定作为寄存器的地址为将来的扩展进行保留。当访问这些地址时，操作不确定且操作无法保证。		p. 64 □	
内部 RAM 区域		如果分支指令发生在内部RAM区域的上限处，那么将不会产生跨越片上外围I/O区域的预取指（无效取指）。		p. 65 □	

章节	分类	功能	功能细节	注意事项	页码
第三章	软件	CPU 功能	向特殊寄存器设置数据	设置IDLE 1模式, IDLE 2模式, 或STOP 模式(通过将PSC.STP位设置为1)后, 必须要立即插入5条或更多条NOP指令。	p. 81 □
				当执行存储指令, 对命令寄存器进行数据写入时, 不响应中断。这是因为假定上述的第<3>和第<4>步骤是通过连续存储指令执行的。若在第<3>和第<4>步骤之间放置有其他指令, 而这个指令又进行了中断的响应, 则上述的操作序列无法建立, 从而导致误操作。	p. 81 □
				尽管对PRCMD寄存器写入的数据为虚数据, 但还是使用通用寄存器来向PRCMD寄存器(示例中的<3>)写入虚数据, 这个通用寄存器就是设置特殊寄存器(示例中的<4>)时使用的通用寄存器。当使用通用寄存器进行寻址时, 上述原则同样适用。	p. 81 □
			SYS 寄存器	如果在对PRCMD寄存器进行写入访问之后, 立即向SYS寄存器(并非特殊寄存器)的PRERR位写入0, 那么PRERR位被清除为0(写入访问优先)。	p. 83 □
				如果在对PRCMD寄存器进行写入访问之后, 立即将数据写入PRCMD寄存器(并非特殊寄存器), 那么PRERR位被设置为1。	p. 83 □
			首先设置的寄存器	当使用V850ES/JJ3器件时, 请确保首先设置下列寄存器。 ●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	访问一个片上外围I/O寄存器需要3个时钟周期(不包括一个等待周期)。根据操作频率, V850ES/JJ3需要不同的等待周期数。因此, 请根据所使用的频率, 将以下对应的数值设置到VSWC寄存器中。	p. 84 □
			访问专用的片上外围I/O寄存器	在以下状态时, 禁止访问上述的寄存器。如果产生了等待状态, 那么只有复位才可以清除等待状态。 ●当CPU运行于副时钟且主时钟振荡停止时 ●当CPU运行于内部振荡时钟时	p. 85 □
第四章	硬件	端口功能	基本端口配置	端口 0、3至6、8和9 都是 5 V 耐压。	p. 87 □
	软件		PFn 寄存器	只有在端口模式下(PMCnm 位 = 0)当PMn寄存器的PMnm位为0时, PFn寄存器的PFnm位有效(当指定为输出模式时)。当PMnm 位为1(当指定为输入模式)时, PFn寄存器的设置值无效。	p. 91 □
	软件, 硬件		端口0	DRST引脚用于片上调试。 如果未使用片上调试, 当RESET引脚的复位信号释放和OCDM.OCDM0位被清除(0)之间时, P05/INTP2/DRST 引脚固定为低电平。 详情参见 4.6.3 片上调试引脚的注意事项。	p. 93 □
	硬件			P00至P06引脚在复用功能的输入模式下具有磁滞特性, 但在端口模式下不具有磁滞特性。	p. 93 □
	软件		PMC0 寄存器	当OCDM.OCDM0 位为1时, 无论PMC05位的值如何, P05/INTP2/DRST引脚都会成为DRST引脚。	p. 95 □
			PF0 寄存器	当一个输出引脚被上拉到 EV _{DD} 或更高时, 请确保设置PF0n位为 1。	p. 96 □
			P1 寄存器	在 D/A 转换期间, 不要读取或写入P1寄存器(参见 14.4.3 注意事项)。	p. 97 □
			PM1 寄存器	当使用P1n作为复用功能(ANOn 引脚输出)时, 设置 PM1n 位为 1。	p. 97 □

□

(3/37)

章节	分类	功能	功能细节	注意事项	页码
第四章	软件	端口功能	PM1 寄存器	当使用P10 和P11引脚的其中一个作为I/O端口，而另外一个引脚作为一路 D/A 输出引脚时，在D/A输出期间端口 I/O 电平不会改变的应用中，才能这样使用。	p. 97 □
	硬件		端口3	在复用功能的输入模式下，P31至P35、P38和P39引脚具有磁滞特性，但在端口模式下不具有磁滞特性。	p. 98 □
	软件		P3 寄存器	要按照字节或按位读/写P3 寄存器的第8位至第15位，将他们指定为P3H 寄存器的第0位至第7位。	p. 99 □
			PM3 寄存器	要按照字节或按位读/写PM3 寄存器的第8位至第15位，将他们指定为PM3H 寄存器的第0位至第7位。	p. 99 □
			PMC3 寄存器	请确保将第15位至第10位、第7位和第6位清除为“0”。	p. 100 □
				要按照字节或按位读/写PMC3 寄存器的第8位至第15位，将他们指定为PMC3H 寄存器的第0位至第7位。	p. 100 □
			PFC3 寄存器	要按照字节或按位读/写PFC3 寄存器的第8位至第15位，将他们指定为PFC3H 寄存器的第0位至第7位	p. 101 □
			PFCE3L 寄存器	请确保将第7位至第3位、第1位和第0位清除为“0”。	p. 101 □
			PFC31/RXDA0 输入/INTP7 输入	INTP7 引脚及 RXDA0 引脚是复用功能引脚。当该引脚用作RXDA0引脚时，请禁止INTP7复用功能引脚的边沿检测。（清除 INTF3.INTF31 位和INTR3.INTR31 位为 0。）当该引脚被用作 INTP7引脚时，停止 UARTA0 接收。（清除 UA0CTL0.UA0RXE 位为 0。）	p. 102 □
			PF3 寄存器	当一个输出引脚被上拉到 EV _{DD} 或更高时，请确保设置PF3n位为 1。	p. 103 □
				要按照字节或按位读/写PF3 寄存器的第8位至第15位，将他们指定为PF3H 寄存器的第0位至第7位	p. 103 □
	硬件		端口4	在复用功能的输入模式下，P40至P42引脚具有磁滞特性，但在端口模式下不具有磁滞特性。	p. 104 □
	软件		PF4 寄存器	当一个输出引脚被上拉到 EV _{DD} 或更高时，请确保设置PF4n位为 1。	p. 105 □
	软件，硬件		端口5	DDI, DDO, DCK和DMS引脚用于片上调试功能。 如果未使用片上调试，当RESET引脚的复位信号释放和OCDM.OCDM0位被清除（0）之间时，P05/INTP2/DRST 引脚固定为低电平。 详情参见 4.6.3 片上调试引脚的注意事项。	p. 106 □
				当电源开启时，P53引脚可能会在复位期间暂时输出一个不确定的电平。	p. 106 □
				在复用功能的输入模式下，P50至P55引脚具有磁滞特性，但在端口模式下不具有磁滞特性。	p. 106 □
	软件		端口 5 复用功能规范	KRn引脚和 TIQ0m 引脚是复用功能引脚。当用作TIQ0m引脚时，需要禁止复用功能KRn引脚的按键返回检测。（将KRM.KRMn位清除为0。） 同样，当该引脚被用作KRn引脚时，需要禁止复用功能TIQ0m引脚的边沿检测（n = 0 至 3，m = 0 至 3）。	p. 109 □
			PF5 寄存器	当一个输出引脚被上拉到 EV _{DD} 或更高时，请确保设置PF5n位为 1。	p. 109 □
	硬件		端口6	在复用功能的输入模式下，P66、P68至P610、P612和P613引脚具有磁滞特性，但在端口模式下不具有磁滞特性。	p. 110 □

章节	分类	功能	功能细节	注意事项	页码
第四章	软件	功能	P6 寄存器	要按照字节或按位读/写P6寄存器的第8位至第15位，将他们指定为P6H 寄存器的第0位至第7位。	p. 111 □
			PM6 寄存器	要按照字节或按位读/写PM6寄存器的第8位至第15位，将他们指定为PM6H 寄存器的第0位至第7位。	p. 111 □
			PMC6 寄存器	要按照字节或按位读/写PMC6 寄存器的第8位至第15位，将他们指定为PMC6H 寄存器的第0位至第7位。	p. 112 □
			PF6 寄存器	当一个输出引脚被上拉到 EV _{DD} 或更高时，请确保设置PF6n位为 1。	p. 113 □
				要按照字节或按位读/写PF6寄存器的第8位至第15位，将他们指定为PF6H 寄存器的第0位至第7位	p. 113 □
			P7H 寄存器， P7L 寄存器	在A/D 转换期间，不要读取/写入P7H 和 P7L 寄存器（参见 13.6（4）复用 I/O ）。	p. 115 □
			PM7H 寄存器， PM7L 寄存器	当使用P7n作为复用功能（ANIn 引脚）时，设置 PM7n 位为 1。	p. 115 □
	硬件	端口8	P80引脚在复用功能的输入模式下具有磁滞特性，但在端口模式下不具有磁滞特性。	p. 116 □	
		软件	PMC8 寄存器	INTP8 引脚及 RXDA3 引脚为复用功引脚。当该引脚用作RXDA3引脚时，请禁止 INTP8复用功能引脚的边沿检测。（清除 INTF8.INTF80 位 = 0 和INTR8.INTR80 位 = 0。） 当该引脚被用作 INTP8引脚时，停止 UARTA3 接收。（清除 UA3CTL0.UA3RXE位为 0。）	p. 117 □
	硬件	PF8 寄存器	当一个输出引脚被上拉到 EV _{DD} 或更高时，请确保设置PF8n位为 1。	p. 117 □	
		端口9	在复用功能的输入模式下，P90至P97、P99、P910和P912至P915引脚具有磁滞特性，但在端口模式下不具有磁滞特性。	p. 118 □	
	软件	P9 寄存器	要按照字节或按位读/写P9 寄存器的第8位至第15位，将他们指定为P9H 寄存器的第0位至第7位。	p. 119 □	
		PM9 寄存器	要按照字节或按位读/写PM9 寄存器的第8位至第15位，将他们指定为PM9H 寄存器的第0位至第7位。	p. 119 □	
		PMC9 寄存器	要按照字节或按位读/写PMC9 寄存器的第8位至第15位，将他们指定为PMC9H 寄存器的第0位至第7位。	p. 120 □	
			当使用A0至A15引脚作为P90至P915引脚的复用功能时，请确保一次设置PMC9寄存器的所有16位内容为FFFFH。	p. 121 □	
		PFC9 寄存器	当执行分离地址总线输出（A0至A15）时，将PFC9或PFCE9寄存器为0000H之后，请一次设置PMC9寄存器的所有16位内容为FFFFH。	p. 122 □	
			要按照字节或按位读/写PFC9 寄存器的第8位至第15位，将他们指定为PFC9H 寄存器的第0位至第7位	p. 122 □	
		PFCE9 寄存器	当执行分离地址总线输出（A0至A15）时，将PFC9或PFCE9寄存器为0000H之后，请一次设置PMC9寄存器的所有16位内容为FFFFH。	p. 122 □	
			要按照字节或按位读/写PFCE9 寄存器的第8位至第15位，将他们指定为PFCE9H 寄存器的第0位至第7位	p. 122 □	
		端口9复用功能的规格	请确保不要同时使用RXDA1和KR7引脚。当使用RXDA1引脚时，不要使用KR7引脚。当使用KR7引脚时，不要使用RXDA1引脚（推荐设置PFC91位设置为1，并且将PFCE91位清除为0）。	p. 124 □	
		PF9 寄存器	当一个输出引脚被上拉到 EV _{DD} 或更高时，请确保设置PF9n位为 1。	p. 125 □	
			要按照字节或按位读/写PF9 寄存器的第8位至第15位，将他们指定为PF9H 寄存器的第0位至第7位	p. 125 □	
		PDL 寄存器	要按照字节或按位读/写PDL 寄存器的第8位至第15位，将他们指定为PDLH 寄存器的第0位至第7位。	p. 136 □	

章节	分类	功能	功能细节	注意事项	页码
第四章	软件	端口功能	PMDL 寄存器	要按照字节或按位读/写PMDL 寄存器的第8位至第15位，将他们指定为PMDLH 寄存器的第0位至第7位。	p. 136 □
			PMCDL 寄存器	当EXIMC寄存器的 SMSEL位 = 1（分离模式）和BSC寄存器的BS30至BS00位 = 0 时（8位总线宽度），不要指定AD8至AD15引脚。	p. 137 □
				要按照字节或按位读/写PMCDL 寄存器的第8位至第15位，将他们指定为PMCDLH 寄存器的第0位至第7位。	p. 137 □
			使用端口引脚作为复用功能引脚时	INTP7 引脚及 RXDA0 引脚是复用功能引脚。当该引脚用作RXDA0引脚时，请禁止INTP7复用功能引脚的边沿检测。（清除 INTF3.INTF31 位和INTR3.INTR31 位为 0。）当该引脚被用作 INTP7引脚时，停止 UARTA0 接收。（清除 UA0CTL0.UA0RXE 位为 0。）	p. 171 □
				当使用P10和P11引脚之一作为I/O端口，而另一个用作D/A输出引脚（ANO0, ANO1）。请确保在D/A输出期间，端口I/O电平不改变。	p. 171 □
				INTP8 引脚及 RXDA3 引脚是复用功能引脚。当该引脚用作RXDA3引脚时，请禁止INTP8复用功能引脚的边沿检测。（清除 INTF8.INTF80 位和INTR8.INTR80 位为 0。）当该引脚被用作 INTP8引脚时，停止 UARTA3 接收。（清除 UA0CTL3.UA3RXE 位为 0。）	p. 174 □
				当设置A0至A15引脚作为复用功能时，请一次设置PMC9寄存器的所有16位内容为 FFFFH。	p. 175 □
				请确保不要同时使用RXDA1和KR7引脚。当使用RXDA1引脚时，不要使用KR7引脚。当使用KR7引脚时，不要使用RXDA1引脚（推荐设置PFC91位设置为1，并且将PFCE91位清除为0）。	p. 175 □
				当设置A0至A15引脚作为复用功能时，请一次设置PMC9寄存器的所有16位内容为 FFFFH。	p. 176 □
			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，请按照下列顺序。 <1> 设置PFn 寄存器 ^注 ： N沟道开漏设置 <2> 设置PFCn 和PFCEn 寄存器： 复用功能选择 <3> 设置PMCn寄存器的对应位为1： 切换到复用功能模式 如果首先设置了PMCn 寄存器，需要注意，此时或按照PFn、PFCn 和PFCEn 寄存器的设置决定引脚状态的改变时，有可能发生意外的操作。	p. 179 □
				无论是端口模式/复用功能模式，Pn 寄存器的读取和写入如下。 • 读取Pn 寄存器： 读取端口输出锁存器的值（当PMn.PMnm 位= 0），或读取引脚状态（PMn.PMnm 位= 1）。 • 写入Pn 寄存器： 写入端口输出锁存器	p. 179 □
			复用功能模式（输入）的注意事项	当 PMCn.PMCnm 位为0时，因为对PMCn 寄存器的设置值和引脚电平进行AND（与操作）输出，输入到复用功能模块的信号为低电平。于是，取决于端口设置和复用功能操作使能时序，可能会发生不可预料的操作。因此，从端口模式切换到复用功能要按照下列顺序进行。 • 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，然后使能复用功能操作。 • 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. 180 □

章节	分类	功能	功能细节	注意事项	页码
第四章	软件	端口功能	端口模式下的 PFn.PFnm 位	在端口模式下，仅在输出模式时（PMn.PMnm 位 = 0）PFn.PFnm 位有效。再输入模式（PMnm 位 = 1），PFnm 位的值不在缓冲器中反映。	p. 181 □
			对端口n寄存器（Pn）进行位操作指令的注意事项	当在一个支持输入和输出的端口上执行1位操作指令时，在写入目标位之外，不打算操作的输入端口的输出锁存值可能会被写入。因此，当端口从输入模式切换为输出模式时，推荐重写输出锁存。	p. 182 □
	硬件, 软件	片上调试引脚的注意事项	如果不使用片上调试，必须进行下列动作。 ●清除OCDM 寄存器（特殊寄存器）的OCDM0位为（0）。 此时，从RESET引脚的复位被释放，直到采取上述动作，P05/INTP2/DRST 引脚一直被固定为低电平。 如果在采取上述动作之前，有高电平输入DRST引脚，可能会导致误操作（CPU死锁）。 处理P05时要尤其注意。	p. 183 □	
			在WDT2RES信号、时钟监视器（CLM）、或低电压检测器（LVI）引发的复位之后，P05/INTP2/DRST引脚不会被初始化作为片上调试引脚（DRST）。OCDM寄存器保持当前值。	p. 183 □	
			P05/INTP2/DRST引脚的注意事项	P05/INTP2/DRST引脚有一个内部下拉电阻（30 kΩ 典型值）。在RESET引脚引发的复位之后，一个下拉电阻被连接。当OCDM0位被清除（0），下拉电阻被断开。	p. 183 □
			当电源开启时，P53引脚的注意事项	当电源开启时，下列引脚可能会在复位期间暂时输出一个不确定的电平。 ●P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 引脚	p. 183 □
			磁滞特性	在端口模式下，下列端口引脚不具有磁滞特性。 P00 至 P06 P31 至 P35, P38, P39 P40 至 P42 P50 至 P55 P66、P68至P610、P612、P613 P80 P90 至 P97, P99, P910, P912 至 P915	p. 183 □
硬件					
第五章	软件	总线控制功能	访问内置ROM时的引脚状态	对内置ROM区域进行写入访问时，地址、数据以及控制信号被激活，和对外部存储器区域进行访问时的激活方式相同。	p. 185 □
			EXIMC寄存器	在进行外部访问之前，从内部ROM或内部RAM区域设置EXIMC寄存器。 在设置EXIMC寄存器之后，请确保插入一条NOP指令。	p. 187 □
			BSC 寄存器	在复位之后写入BSC寄存器，然后不要更改设置值。此外，在完成对BSC寄存器的初始设置之前，请不要访问外部存储器区域。	p. 188 □
				请确保设置第14位、第12位、第10位和第8位设置为“1”，设置第15位、第13位、第11位、第9位、第7位、第5位、第3位和第1位为“0”。	p. 188 □
			DWC0 寄存器	内置ROM和内置RAM无需可编程等待，对它们的访问总是无需插入等待状态。片上外围I/O区域同样无需可编程等待，仅执行来自各个外围功能的等待控制。	p. 196 □

(7/37)

章节	分类	功能	功能细节	注意事项	页码
第五章	软件	总线控制功能	DWC0 寄存器	在复位之后写入DWC0寄存器，然后不要更改设置值。同样，在完成对DWC0寄存器的初始设置之前，请不要访问外部存储器区域。	p. 196 □
				当V850ES/JJ3使用分离总线模式且操作频率 $f_{xx} > 20 \text{ MHz}$ 时，请确保插入一个或更多等待。	p. 196 □
				请确保将第15位、第11位、第7位和第3位清除为“0”。	p. 196 □
			AWC 寄存器	当访问内置ROM区域、内置RAM区域和片上外围I/O区域时，不会插入地址建立等待和地址保持等待周期。	p. 199 □
				在复位之后写入AWC寄存器，然后不要更改设置值。同样，在完成对AWC寄存器的初始设置之前，请不要访问外部存储器区域。	p. 199 □
				当V850ES/JJ3操作频率 $f_{xx} > 20 \text{ MHz}$ 时，请确保插入地址保持等待和地址建立等待。	p. 199 □
				请确保将第15位至第8位设置为“1”。	p. 199 □
			BCC 寄存器	内置ROM、内置RAM和片上外围I/O区域无需插入空闲状态。	p. 200 □
				在复位之后写入BCC寄存器，然后不要更改设置值。同样，在完成对BCC寄存器的初始设置之前，请不要访问外部存储器区域。	p. 200 □
				请确保设置第15位、第13位、第11位和第9位设置为“1”，设置第14位、第12位、第10位、第8位、第6位、第4位、第2位和第0位为“0”。	p. 200 □
第六章	软件	时钟发生功能	PCC 寄存器	当正在输出CLKOUT时，不要改变CPU时钟（通过CK3至CK0位）。	p. 214 □
				操作CK3位时，请使用1位操作指令。当使用8位操作指令时，不要改变CK2至CK0位的设置值。	p. 214 □
				当停止主时钟时，也要停止PLL。同样，工作于主时钟的片上外围功能的操作也被停止。	p. 215 □
				如果不满足下列条件，则改变CK2至CK0位的设置，以满足该条件，然后再切换到副时钟工作模式。 内部系统时钟 (f_{CLK}) > 副时钟 (f_{XT} : 32.768 kHz) $\times 4$	p. 215 □
				只有在主时钟振荡稳定之后，才使能工作于主时钟频率的片上外围功能。如果在经过振荡稳定时间之前，它们的操作就被使能，则可能发生误操作。	p. 216 □
			RCM 寄存器	当CPU工作于内部振荡器时钟时（CCLS.CCLS F位 = 1），不可以停止内部振荡器。不要将RSTOP位设置为1。	p. 217 □
				即使RSTOP位被设置为1，如果CCLS.CCLS F位被置1，内部振荡器开始振荡。（在振荡稳定期间产生WDT溢出时）此时，RSTOP位仍然被设置为1。	p. 217 □
			PLLCTL 寄存器	当PLLON位被清除为0时，SELPLL位也会被自动清除为0（时钟直通模式）。	p. 219 □
				仅当PLL时钟频率稳定后，SELPLL位可以被设置为1。如果没有稳定（未入锁），如果向其写入数据，SELPLL位会被写入“0”。	p. 219 □
			CKC 寄存器	在 $f_x = 5.0$ 至 10.0 MHz 时，不能使用PLL模式。	p. 220 □
				在通过使用CKC寄存器将倍频系数改变为4和8之前，设置时钟直通模式并停止PLL。	p. 220 □
				请确保设置第3位和第1位设置为“1”，设置第7位至第4位和第2位为“0”。	p. 220 □

章节	分类	功能	功能细节	注意事项	页码
第六章	软件	时钟发生功能	LOCKR 寄存器	LOCK寄存器不能实时反映PLL的锁定状态。	p. 221 □
			PLLS 寄存器	设置入锁时间为800 μ s 或更长。	p. 222 □
				在入锁过程中，不要改变PLLS寄存器的设置。	p. 222 □
第七章	软件	16位定时器/事件计数器 P (TMP)	TPnCTL0 寄存器	当TPnCE位 = 0时，对TPnCKS2至TPnCKS0 位进行设置。当TPnCE 位的值从0改变为1时，可以同时设置TPnCKS2至TPnCKS0位。	p. 227 □
				请确保将第3位至第6位清除为“0”。	p. 227 □
			TPnCTL1 寄存器	只有在外部触发脉冲输出模式或单次触发脉冲输出模式下，TPnEST位有效。在其它模式下，对该位写入1的操作将被忽略。	p. 228 □
				在外部事件计数模式下，无论TPnEEE位如何设置，都会选择外部事件计数输入。	p. 228 □
				当TPnCTL0.TPnCE 位 = 0时，设置TPnEEE 和 TQnMD2 至TPnMD0位（当TPnCE 位 = 1时，可以对这些位写入相同的值）。如果在TPnCE位 = 1时对这些位重写，那么操作将无法保证。如果因为误操作而重写，则要将TPnCE位清除为0，然后再次设置这些位。	p. 228 □
				请确保将第3位、第4位和第7位清除为“0”。	p. 228 □
			TPnIOC0 寄存器	当TPnCTL0.TPnCE 位 = 0时，重写TPnOL1、TPnOE1、TPnOL0 和 TPnOE0 位（当TPnCE位 = 1时，可以向这些位写入相同的值）。如果因为误操作而重写，则要将TPnCE位清除为0，然后再次设置这些位。	p. 229 □
				即使当TPnCE位和TPnOEm位为0时操作TPnOLm位，TOPnm引脚输出电平会不同（m = 0, 1）。	p. 229 □
			TPnIOC1 寄存器	当TPnCTL0.TPnCE 位 = 0时，重写TPnIS3至TPnIS0位（当TPnCE位 = 1时，可以向这些位写入相同的值）。如果因为误操作而重写，则要将TPnCE位清除为0，然后再次设置这些位。	p. 230 □
				只有在自由运行定时器模式和脉宽测量模式下，TPnIS3至TPnIS0 位有效。其它模式下，无法进行捕捉操作。	p. 230 □
			TPnIOC2 寄存器	当TPnCTL0.TPnCE 位 = 0时，重写TPnEES1、TPnEES0、TPnETS1和TPnETS0 位（当TPnCE位 = 1时，可以向这些位写入相同的值）。如果因为误操作而重写，则要将TPnCE位清除为0，然后再次设置这些位。	p. 231 □
				只有当TPnCTL1.TPnEEE 位 = 1 或设置为外部事件计数模式（TPnCTL1.TPnMD2至TPnCTL1.TPnMD0 位 = 001）时，TPnEES1 和 TPnEES0 位才有效。	p. 231 □
				只有在外部触发脉冲输出模式（TPnCTL1.TPnMD2至TPnCTL1.TPnMD0 位 = 010）或单次触发脉冲输出模式（TPnCTL1.TPnMD2至TPnCTL1.TPnMD0 = 011）下，TPnETS1 和 TPnETS0 位才有效。	p. 231 □
			TPnOPT0 寄存器	当TPnCE 位 = 0时，重写TPnCCS1和TPnCCS0位（当TPnCE位 = 1时，可以向这些位写入相同的值）。如果因为误操作而重写，则要将TPnCE位清除为0，然后再次设置这些位。	p. 232 □
				请确保将第1位至第3位、第6位和第7位清除为“0”。	p. 232 □
			TPnCCR0 寄存器	在下列状态时，禁止访问TPnCCR0寄存器。详情参见 3.4.8 (2) 访问专用片上外围I/O寄存器 。 <ul style="list-style-type: none"> 当CPU运行于副时钟且主时钟振荡停止时 当CPU 运行于内部振荡时钟时 	p. 233 □

章节	分类	功能	功能细节	注意事项	页码
第七章	软件	16位定时器/事件计数器 P (TMP)	TPnCCR1 寄存器	在下列状态时，禁止访问TPnCCR1寄存器。详情参见 3.4.8 (2) 访问专用片上外围I/O寄存器 。 • 当CPU运行于副时钟且主时钟振荡停止时 • 当CPU 运行于内部振荡时钟时	p. 235 □
			TPnCNT 寄存器	在下列状态时，禁止访问TPnCNT寄存器。详情参见 3.4.8 (2) 访问专用片上外围I/O寄存器 。 • 当CPU运行于副时钟且主时钟振荡停止时 • 当CPU 运行于内部振荡时钟时	p. 237 □
			憋崮	要使用外部事件计数模式，请指定TIPn0引脚捕获触发输入的有效边沿不作检测（通过将TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位清除为“00”）。	p. 238 □
				当使用外部触发脉冲输出模式、单次触发脉冲输出模式和脉宽测量模式时，选择内部时钟作为计数时钟（通过将TPnCTL1.TPnEEE位清除为0）。	p. 238 □
			间隔定时器模式（TPnMD2至TPnMD0 位 = 000）	仅当中断请求信号（INTTPnCC0 和 INTTPnCC1）被中断屏蔽标志（TPnCCMK0 和TPnCCMK1）屏蔽且同时执行定时器输出（TOPn1）时，该位才可以被设置为1。但是，设置TPnCCR0 和 TPnCCR1 寄存器为相同的值。（参见 7.5.1 (2) (d) TPnCCR1 寄存器的操作 ）。	p. 240 □
			重写TPnCCR0 寄存器需要注意的事项	要将TPnCCR0寄存器改变为较小的值时，要停止一次计数，然后改变设置值。如果在计数期间将TPnCCR0寄存器改变为较小的值，那么16位计数器可能会溢出。	p. 245 □
			外部事件计数模式下的寄存器设置	当外部时钟用作计数时钟时，外部时钟仅可以从TIPn0引脚输入。此时，设置TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0位为00。（捕获触发器输入（TIPn0 引脚）：无边沿检测）。	p. 251 □
			外部事件计数模式（TPnMD2至TPnMD0 位 = 001）	在外部事件计数模式下，不要将TPnCCR0 寄存器设置为0000H。	p. 253 □
				在外部事件计数模式下，禁止使用定时器输出。如果使用外部事件计数输入执行定时器输出，请设置为间隔定时器模式，并通过外部事件计数输入为计数时钟选择操作使能（TPnCTL1.TPnMD2至TPnCTL1.TPnMD0 位 = 000，TPnCTL1.TPnEEE 位 = 1）。	p. 253 □
			重写TPnCCR0 寄存器需要注意的事项	要将TPnCCR0寄存器改变为较小的值时，要停止一次计数，然后改变设置值。如果在计数期间将TPnCCR0寄存器改变为较小的值，那么16位计数器可能会溢出。	p. 254 □
			TPnIOC0, TPnOE0, TPnOL0 位	在外部触发脉冲输出模式下不使用TOPn0引脚时，将该位清除为0。	p. 259 □
			操作期间改变脉冲宽度需要注意的事项	当计数器计数时，要改变PWM波形，则应该在最后写入TPnCCR1寄存器。在检测到INTTPnCC0信号之后，写入TPnCCR1寄存器，然后再重写TPnCCRm 寄存器。	p. 263 □
			TPnIOC0.TPnOE0, TPnOL0 位	当单次触发脉冲输出模式下不使用TOPn0引脚时，将该位清除为0。	p. 271 □
			在单脉冲输出模式下操作所需的寄存器设置	即使在单次触发脉冲输出模式下，如果TPnCCR1寄存器中的设置值大于TPnCCR0 寄存器中的设置值，也不会输出单次触发脉冲。	p. 272 □

章节	分类	功能	功能细节	注意事项	页码
第七章	软件	16位定时器/事件计数器 P (TMP)	重写TPnCCRM寄存器需要注意的事项	要将TPnCCRM寄存器改变为较小的值时, 要停止一次计数, 然后改变设置值。如果在计数期间将TPnCCRM寄存器改变为较小的值, 那么16位计数器可能会溢出。	p. 274 □
			TPnIOC0.TPnOE0, TPnOLO 位	在PWM输出模式下不使用TOPn0引脚时, 将该位清除为0。	p. 278 □
			选择器功能	当使用选择器功能时, 请确保为TMP设置端口/定时器复用功能引脚连接到捕获触发输入。	p. 309 □
				设置选择器功能之前, 禁止连接的周边I/O器件 (TMP/UARTA)。	p. 309 □
			SELCNT0 寄存器	当设置 ISEL3、ISEL4位或ISEL6位为“1”, 请确保设置对应的复用功能引脚为捕获触发输入。	p. 310 □
				请确保将第7位、第5位、第2位至第0位清除为“0”。	p. 310 □
			捕获操作	当使用捕获操作且选择一个较慢的时钟作为计数时钟, 如果在TPnCE位被设置为1之后, 立刻检测到捕获触发信号, 则捕获到TPnCCR0和TPnCCR1寄存器中计数值将不是0000H, 而是FFFFH。	p. 311 □
第八章	软件	16位定时器/事件计数器 Q (TMQ)	TQnCTL0 寄存器	当TQnCE 位 = 0时, 设置TQnCKS2至TQnCKS0 位。当TQnCE位的值从0改变为1时, 可以同时设置TQnCKS2至TQ0CKS0位。	p. 316 □
				请确保将第3位至第6位清除为“0”。	p. 316 □
			TQ0CTL1 寄存器	仅在外部触发脉冲输出模式或单次触发脉冲输出模式下, TQ0EST位有效。在其它模式下, 对该位写入1的操作将被忽略。	p. 317 □
				在外部事件计数模式下, 无论TQ0EEE位如何设置, 都会选择外部事件计数输入。	p. 317 □
				当TQ0CTL0.TQ0CE 位 = 0时, 设置TQ0EEE 和 TQ0MD2 至TQ0MD0位 (当TQ0CE 位 = 1时, 可以向这些位写入相同的值)。如果在TQ0CE位 = 1时执行重写, 那么操作将无法保证。如果因为误操作而重写, 则将TQ0CE位清除为0, 然后再次设置这些位。	p. 317 □
				请确保将第3位、第4位和第7位清除为“0”。	p. 317 □
			TQ0IOC0 寄存器	当TQ0CTL0.TQ0CE 位 = 0时, 重写TQ0OLm 和 TQ0OEm位 (当TQ0CE 位 = 1时, 可以向这些位写入相同的值)。如果因为误操作而重写, 则要将TQ0CE位清除为0, 然后再次设置这些位。	p. 318 □
				当 TQ0CE 和 TQ0OEm 位都为0时, 即使对TQ0OLm位进行操作, TQ0Om 引脚的输出电平不同。	p. 318 □
			TQ0IOC1 寄存器	当TQ0CTL0.TQ0CE 位 = 0时, 重写 TQ0IS7至TQ0IS0位 (当TQ0CE 位 = 1时, 可以向这些位写入相同的值)。如果因为误操作而重写, 则要将TQ0CE位清除为0, 然后再次设置这些位。	p. 319 □
				仅在自由运行定时器模式和脉宽测量模式下, TQnIS7 至TQnIS0位有效。其它模式下, 无法进行捕捉操作。	p. 319 □
			TQ0IOC2 寄存器	当TQ0CTL0.TQ0CE 位 = 0时, 重写TQ0EES1、TQ0EES0、TQ0ETS1和TQ0ETS0位 (当TQ0CE 位 = 1时, 可以向这些位写入相同的值)。如果因为误操作而重写, 则要将TQ0CE位清除为0, 然后再次设置这些位。	p. 320 □
				只有当TQ0CTL1.TQ0EEE 位 = 1 或被设置为外部事件计数模式 (TQ0CTL1.TQ0MD2至TQ0CTL1.TQ0MD0 位 = 001) 时, TQ0EES1 和 TQ0EES0 位有效。	p. 320 □

章节	分类	功能	功能细节	注意事项	页码
第八章	软件	16位定时器/事件计数器 Q (TMQ)	TQ0IOC2 寄存器	只有在外部触发脉冲输出模式 (TQ0CTL1.TQ0MD2至TQ0CTL1.TQ0MD0 位 = 010) 或单次触发脉冲输出模式 (TQ0CTL1.TQ0MD2至TQ0CTL1.TQ0MD0 = 011) 下, TQ0ETS1 和 TQ0ETS0 位有效。	p. 320 □
			TQ0OPT0 寄存器	当TQ0CTL0.TQ0CE 位 = 0时, 重写 TQ0CCS3至TQ0CCS0位 (当TQ0CE 位 = 1 时, 可以向这些位写入相同的值)。 如果因为误操作而重写, 则要将TQ0CE位清除为0, 然后再次设置这些位。 请确保将第1位至第3位清除为“0”。	p. 321 □ p. 321 □
			TQ0CCR0 寄存器	在下列状态时, 禁止访问TQ0CCR0寄存器。详情参见 3.4.8 (2) 访问专用片上外围I/O寄存器 。 • 当CPU运行于副时钟且主时钟振荡停止时 • 当CPU 运行于内部振荡时钟时	p. 322 □
			TQ0CCR1 寄存器	在下列状态时, 禁止访问TQ0CCR1寄存器。详情参见 3.4.8 (2) 访问专用片上外围I/O寄存器 。 • 当CPU运行于副时钟且主时钟振荡停止时 • 当CPU 运行于内部振荡时钟时	p. 324 □
			TQ0CCR2 寄存器	在下列状态时, 禁止访问TQ0CCR2寄存器。详情参见 3.4.8 (2) 访问专用片上外围I/O寄存器 。 • 当CPU运行于副时钟且主时钟振荡停止时 • 当CPU 运行于内部振荡时钟时	p. 326 □
			TQ0CCR3 寄存器	在下列状态时, 禁止访问TQ0CCR3寄存器。详情参见 3.4.8 (2) 访问专用片上外围I/O寄存器 。 • 当CPU运行于副时钟且主时钟振荡停止时 • 当CPU 运行于内部振荡时钟时	p. 328 □
			TQ0CNT寄存器	在下列状态时, 禁止访问TQ0CNT寄存器。详情参见 3.4.8 (2) 访问专用片上外围I/O寄存器 。 • 当CPU运行于副时钟且主时钟振荡停止时 • 当CPU 运行于内部振荡时钟时	p. 330 □
			外部事件计数模式	要使用外部事件计数模式, 请指定 TIQ00引脚捕获触发输入的有效边沿不作检测 (通过将TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位清除为“00”)。	p. 331 □
			外部触发脉冲输出模式, 单次触发脉冲输出模式, 脉宽测量模式	当使用外部触发脉冲输出模式、单次触发脉冲输出模式和脉宽测量模式时, 选择内部时钟作为计数时钟 (通过将TQ0CTL1.TQ0EEE位清除为0)。	p. 331 □
			TQ0CTL1.TQ0EEE 位	只有在中断请求信号 (INTTQ0CC0和INTTQ0CCK) 被中断屏蔽标志 (TQ0CCMK0至TQ0CCMKk) 屏蔽且同时执行定时器输出 (TOQ0k) 时, 该位才可以被设置为1。但是, 必须对TQ0CCR0 和TQ0CCRk 寄存器设置相同的值。(参见 8.5.1 (2) (d) TQ0CCR1至TQ0CCR3 寄存器的操作)。	p. 333 □

章节	分类	功能	功能细节	注意事项	页码
第八章	软件	16位定时器/事件计数器 Q (TMQ)	重写TQ0CCR0寄存器时需要注意的事项	要将TQ0CCR0寄存器改变为较小的值时，要停止一次计数，然后改变设置值。如果在计数期间将TQ0CCR0寄存器改变为较小的值，那么16位计数器可能会溢出。	pp. 337, 346 □
			外部事件计数模式下的寄存器设置	当外部时钟被用作计数时钟时，外部时钟仅可以从TIQ00引脚输入。此时，设置TQ0IOC1.TQ0IS1和TQ0IOC1.TQ0IS0位为00。（捕获触发输入（TIQ00引脚）：无边沿检测）。	p. 343 □
			外部事件计数模式（TQ0MD2至TQ0MD0 bits = 001）	在外部事件计数模式下，不要将TQ0CCR0寄存器设置为0000H。	p. 345 □
				在外部事件计数模式下，禁止使用定时器输出。如果使用外部事件计数输入执行定时器输出，请设置为间隔定时器模式，并通过外部事件计数输入为计数时钟选择操作使能（TQ0CTL1.TQ0MD2至TQ0CTL1.TQ0MD0位 = 000，TQ0CTL1.TQ0EEE位 = 1）。	p. 345 □
			TQ0IOC0.TQ0OE0，TQ0OL0位	在外部触发脉冲输出模式下不使用TOQ00引脚时，将该位清除为0。	p. 353 □
			操作期间改变脉冲宽度需要注意的事项	当计数器计数时，要改变PWM波形，则应该在最后写入TQ0CCR1寄存器。在检测到INTTQ0CC0信号之后，写入TQ0CCR1寄存器，然后再重写TQ0CCRk寄存器。	p. 357 □
			TQ0IOC0.TQ0OE0，TQ0OL0位	在单次触发脉冲输出模式下不使用TOQ00引脚时，将该位清除为0。	p. 366 □
			在单脉冲输出模式下操作所需的寄存器设置	即使在单次触发脉冲输出模式下，如果TQ0CCRk寄存器中的设置值大于TQ0CCR0寄存器中的设置值，也不会输出单次触发脉冲。	p. 367 □
			重写TQ0CCRM寄存器时需要注意的事项	要将TQ0CCRM寄存器改变为较小的值时，要停止一次计数，然后改变设置值。如果在计数期间将TQ0CCR0寄存器改变为较小的值，那么16位计数器可能会溢出。	p. 370 □
			TQ0IC0.TQ0OE0，TQ0OL0位	在PWM输出模式下不使用TOQ00引脚时，将该位清除为0。	p. 375 □
第九章	软件	16位间隔定时器 M (TMM)	TM0CTL0 寄存器	当TM0CE位= 0时，设置TM0CKS2至TM0CKS0位。当TM0CE的值从0改变位1时，不可以同时设置TM0CKS2至TM0CKS0位的值。	p. 413 □
				请确保将第3位至第6位清除为“0”。	p. 413 □
			间隔定时器模式下的操作	不要将TM0CMP0寄存器设置为FFFFH。	pp. 414, 417 □
			计数开始	根据选择的计数时钟，在TM0CTL0.TM0CE位被设置为1之后，当16位计数器达到以下时间间隔，就开始计数。	p. 418 □

章节	分类	功能	功能细节	注意事项	页码
第九章	软件	16位间隔定时器 M (TMM)	TM0CMP0, TM0CTL0 寄存器	当TMM0工作时，禁止重写TM0CMP0和TM0CTL0 寄存器。 如果在TM0CE位为1时重写这些寄存器，那么操作将无法保证。 如果这两个寄存器被错误地改写，将TM0CTL0.TM0CE位清除为0，然后重置这些寄存器。	p. 418 □
第十章	软件	钟表定时器功能	PRSM0 寄存器	在钟表定时器操作期间，不要改变的BGCS00 和 BGCS01位的值。	p. 422 □
				在设置BGCE0位为1之前，设置PRSM0寄存器。	p. 422 □
				根据所使用的主时钟频率，设置PRSM0 和PRSCM0 寄存器，以获得32.768 kHz的f _{BRG} 频率。	p. 422 □
			PRSCM0 寄存器	在钟表定时器操作期间，不要重写PRSCM0寄存器。	p. 423 □
				在设置 PRSM0.BGCE0 位为1之前，设置PRSCM0寄存器。	p. 423 □
				根据所使用的主时钟频率，设置PRSM0 和PRSCM0 寄存器，以获得32.768 kHz的f _{BRG} 频率。	p. 423 □
			WTM 寄存器	当WTM0 和WTM1位都为0时，重写WTM2至WTM7位。	p. 425 □
			注意事项	在操作使能（WTM.WTM1 和WTM.WTM0位 = 1）之后，在产生第一次钟表定时器中断请求信号（INTWT）之前，需要一段时间。	p. 428 □
	硬件		需要经过0.515625 秒（最大值），才会产生第1个INTWT信号（2 ⁹ × 1/32768 = 0.015625 秒（最大值））。 之后每隔0.5 秒产生1次INTWT信号。	p. 428 □	
第十一章	软件	看门狗2功能	默认-启动看门狗定时器	复位释放后，看门狗定时器 2在复位模式下自动启动。 当不使用看门狗定时器 2时，则在执行复位前通过该功能停止其操作，或清除一次看门狗定时器2，并在下一次间隔时间内停止其操作。 同样，即使无需改变默认设置（复位模式，间隔时间：f _R /2 ¹⁹ ），为了验证，仅写入WDTM2一次。	p. 429 □
				由不可屏蔽中断请求信号（INTWDT2）引发的不可屏蔽中断服务，参见19.2.2（2） 来自INTWDT2 的信号。	p. 429 □
		WDTM2 寄存器	下述状态下，禁止访问WDTM2 寄存器。详情参见3.4.8（2） 访问专用片上外围 I/O寄存器。 ● 当CPU运行于副时钟且主时钟振荡停止时 ● 当CPU 运行于内部振荡时钟时	p. 431 □	
			WDCS20至WDCS24 位的详情参见表 11-2 看门狗定时器 2 时钟选择。	p. 431 □	
			虽然通过停止内部振荡器的操作就可以停止看门狗定时器2，清除WDTM2寄存器为00H，以保证停止定时器（要防止因为错误写入操作选择了主时钟或副时钟）。	p. 431 □	
			如果在复位后WDTM2寄存器被改写两次，则强制产生溢出信号，且计数器被复位。	p. 431 □	
			要故意产生一个溢出信号，仅需要两次写入WDTM2寄存器，或者向WDTE寄存器写入一次非“ACH”的值。 然而，当看门狗定时器 2被设置为停止操作时，即使向WDTM2寄存器写入两次数据，或者写入一次非“ACH”的值，也不会产生溢出信号。	p. 431 □	

章节	分类	功能	功能细节	注意事项	页码
第十一章	软件	看门狗2功能	WDTM2 寄存器	要停止看门狗定时器 2 的操作，将RCM.RSTOP 位设置为1（停止内部振荡器）且向WDTM2寄存器写入00H。如果RCM.RSTOP位不能被设置为1，设置WDSCS23 位为1（选择2 ⁿ f _{xx} ，且在IDLE1、IDLW2、副IDLE和副时钟操作模式下可以停止时钟）。	p. 431 □
			WDTE 寄存器	当非“ACH”值被写入WDTE寄存器时，强制输出溢出信号。	p. 432 □
				当对WDTE寄存器执行1位存储器操作指令时，强制输出溢出信号。	p. 432 □
				要故意产生一个溢出信号，仅需要向WDTE寄存器写入一次非“ACH”的值，或者向WDTM2寄存器写入两次。 然而，当看门狗定时器 2 被设置为停止操作时，即使向WDTM2寄存器写入了两次数据，或者向WDTE寄存器写入一次非“ACH”值，也不会产生溢出信号。	p. 432 □
				WDTE 寄存器的读取值为“9AH”（与写入值“ACH”不同）。	p. 432 □
第十二章	软件	实时计数输出功能（RTO）	RTBLn, RTBHn 寄存器	当向RTBHn 寄存器的第6位和第7位写入时，总是写入0。	p. 436 □
				在以下状态时，禁止访问RTBLn和RTBHn寄存器。详情参见3.4.8（2）访问专用片上外围I/O寄存器。 • 当CPU运行于副时钟且主时钟振荡停止时 • 当CPU 运行于内部振荡时钟时	p. 436 □
				在设置实时输出端口之后，在产生一个实时输出触发时，将输出数据设置到RTBLn和RTBHn寄存器。	p. 436 □
			RTPMn寄存器	通过使能实时输出操作（RTPCn.RTPOEn 位 = 1），RTPn0至RTPn5引脚中被设置为实时输出的位进行实时输出，而设置为端口模式的位输出0。	p. 437 □
				如果禁止实时输出操作（RTPOEn位 = 0），不管RTPMn寄存器如何设置，实时输出引脚（RTPn0至RTPn5）都输出0。	p. 437 □
				为了将该寄存器用作实时输出引脚（RTPn0 to RTPn5），需用通过PMC 和 PFC 寄存器将这些引脚设置为实时输出端口引脚。	p. 437 □
			RTPCn寄存器	仅当RTPOEn 位 = 0时，设置RTPEGn、BYTEn和EXTRn 位。	p. 438 □
			实时输出操作	通过软件避免以下的冲突。 • 实时输出禁止/使能切换（（RTPOEn 位）和选择的实时输出输出触发信号之间的冲突。 • 在实时输出状态下对RTBHn和RTBLn寄存器的写入操作和选择的实时输出触发信号之间的冲突。	p. 440 □
			初始化	在执行初始化之前，禁止实时输出（RTPOEn位 = 0）。	p. 440 □
第十三章	硬件	A/D 转换器	ANIO至ANI15 引脚	确保输入ANIO至ANI15的电压不超过额定值。 特别是，如果输入某个通道的电压大于或等于AV _{REF0} ，这个通道的转换值将不确定，其它通道的转换值也将受到影响。	p. 444 □
			ADA0M0 寄存器	下述状态时，禁止访问ADA0M0 寄存器。详情参见3.4.8（2）访问专用片上外围I/O寄存器。 • 当CPU运行于副时钟且主时钟振荡停止时 • 当CPU 运行于内部振荡时钟时	p. 446 □
	软件				

(15/37)

章节	分类	功能	功能细节	注意事项	页码
第十三章	软件	A/D 转换器	ADA0M0 寄存器	对第0位的写入操作将被忽略。	p. 446 □
				当A/D转换被使能（ADA0CE 位= 1）时，禁止改变ADA0M1.ADA0FR2至ADA0M1.ADA0FR0位。	p. 446 □
				下列模式中，向ADA0M0、ADA0M2、ADA0S、ADA0PFM或 ADA0PFT寄存器写入数据，通过将ADA0CE位清除为0，停止A/D转换。在数据被写入到该寄存器之后，通过设置ADA0CE位为1，再次使能A/D转换。 • 正常转换模式下 • 高速转换模式的单次选择模式/单次扫描模式 在其它模式下，如果在A/D转换期间（ADA0EF位= 1）写入ADA0M0、ADA0M2、ADA0S、ADA0PFM和 ADA0PFT寄存器，根据当前模式，将会执行如下操作。 • 软件触发模式 A/D转换停止，并再次从头开始执行。 • 硬件触发模式 A/D转换停止，且设置为触发等待状态。	p. 446 □
				为了选择外部触发模式/定时器触发模式（ADA0TMD位= 1），设置为高速转换模式（ADA0M1.ADA0HS1位= 1）。A/D转换操作使能（ADA0CE位= 1）之后，在随后插入的振荡稳定时间期间，不要输入触发信号。	p. 446 □
				当不使用A/D转换器时，设置ADA0CE 位为0，停止操作以减少功耗。	p. 446 □
			ADA0M1 寄存器	A/D转换使能（ADA0M0.ADA0CE位= 1）时，禁止改变ADA0M1寄存器。	p. 447 □
				为了选择外部触发模式/定时器触发模式（ADA0M0.ADA0TMD位= 1），设置高速转换模式（ADA0HS1位= 1）。A/D转换操作使能（ADA0CE位= 1）之后，在随后插入的振荡稳定时间期间，不要输入触发信号。	p. 447 □
				请确保将第6位至第4位清除为“0”。	p. 447 □
			正常转换模式下（ADA0HS1 位= 0）转换时间的选择	设置 $2.6\ \mu\text{s} \leq \text{转换时间} \leq 10.4\ \mu\text{s}$ 。	p. 448 □
				A/D转换期间，如果有数据写入ADA0M0、ADA0M2、ADA0S、ADA0PFM以及ADA0PFT寄存器，或者有触发信号输入，则重新进行转换。但是，如果稳定时间结束时序与写入这些寄存器的操作发生冲突，或者稳定时间结束时序与触发信号输入发生冲突，那么将会重新插入64个时钟的稳定时间。 如果在插入的稳定时间结束时序时刻再次发生冲突，则重新插入稳定时间。因此，不要将触发输入间隔和控制寄存器写入间隔设置为64个时钟或更少。	p. 448 □
			高速转换模式下（ADA0HS1 位= 1）转换时间的选择	设置 $2.6\ \mu\text{s} \leq \text{转换时间} \leq 10.4\ \mu\text{s}$ 。	p. 449 □
				在高速转换模式下，振荡稳定期间，触发信号输入和对ADA0M0、ADA0M2、ADA0S、ADA0PFM和ADA0PFT寄存器的重写都被禁止。	p. 449 □
			ADA0M2 寄存器	下列模式中，向ADA0M2寄存器写入数据，通过将ADA0M0.ADA0CE位清除为0，停止A/D转换。在数据被写入到该寄存器之后，通过设置ADA0CE位为1，再次使能A/D转换。 • 正常转换模式下 • 高速转换模式的单次选择模式/单次扫描模式	p. 450 □
				请确保将第7位至第2位清除为“0”。	p. 450 □

章节	分类	功能	功能细节	注意事项	页码
第十三章	软件	A/D 转换器	ADA0S 寄存器	<p>下列模式中，向ADA0S寄存器写入数据，通过将AD0M0.ADA0CE位清除为0，停止A/D转换。在数据被写入到该寄存器之后，通过设置ADA0CE位为1，再次使能A/D转换。</p> <ul style="list-style-type: none"> • 正常转换模式下 • 高速转换模式的单次选择模式/单次扫描模式 <p>请确保将第7位至第4位清除为“0”。</p>	p. 451 □
			ADA0CRn, ADA0CRnH 寄存器	<p>在下述状态时，禁止访问ADA0CRn 和ADA0CRnH 寄存器。详情参见3.4.8 (2) 访问专用片上外围I/O寄存器。</p> <ul style="list-style-type: none"> • 当CPU运行于副时钟且主时钟振荡停止时 • 当CPU 运行于内部振荡时钟时 <p>向ADA0M0 和ADA0S 寄存器的写入操作可能会导致ADA0CRn 寄存器的内容变为不确定。转换完成之后，在写入 ADA0M0 和ADA0S 寄存器之前，读取转换结果。如果不按上述顺序进行，可能读不到正确的转换结果。</p>	p. 451 □
			ADA0PFM 寄存器	<p>在选择模式下，将设置到ADA0PFT寄存器的8位数据与ADA0S寄存器指定的ADA0CRnH寄存器的值进行比较。如果该结果符合ADA0PFC位指定的条件，那么将转换结果存储在ADA0CRn寄存器，并且产生INTAD信号。但是，如果不符合条件，将不产生中断信号。</p>	p. 452 □
				<p>在扫描模式下，将设置到ADA0PFT寄存器的8位数据与ADA0CR0H寄存器的内容进行比较。如果该结果符合ADA0PFC位指定的条件，那么将转换结果存储在ADA0CR0寄存器中，并且产生INTAD信号。但是，如果不符合条件，将不产生INTAD信号。无论比较结果如何，扫描操作完成之前，都会继续扫描，并且将转换结果存储在ADA0CRn寄存器中。但是，扫描操作完成后不产生INTAD信号。</p>	p. 452 □
				<p>下列模式中，向ADA0PFM寄存器写入数据，通过将AD0M0.ADA0CE位清除为0，停止A/D转换。在数据被写入到该寄存器之后，通过设置ADA0CE位为1，再次使能A/D转换。</p> <ul style="list-style-type: none"> • 正常转换模式下 • 高速转换模式的单次选择模式/单次扫描模式 	p. 454 □
			ADA0PFT 寄存器	<p>下列模式中，向ADA0PFT寄存器写入数据，通过将AD0M0.ADA0CE位清除为0，停止A/D转换。在数据被写入到该寄存器之后，通过设置ADA0CE位为1，再次使能A/D转换。</p> <ul style="list-style-type: none"> • 正常转换模式下 • 高速转换模式的单次选择模式/单次扫描模式 	p. 454 □
			外部触发模式	<p>为了选择外部触发模式，需要设置为高速转换模式。A/D转换操作使能(ADA0M0.ADA0CE位= 1)后，在随后插入的振荡稳定时间期间，不要输入触发信号。</p>	p. 455 □
			定时器触发模式	<p>为了选择定时器触发模式，需要设置为高速转换模式。A/D转换操作使能(ADA0M0.ADA0CE位= 1)后，在随后插入的振荡稳定时间期间，不要输入触发信号。</p>	p. 458 □
			不使用A/D 转换器时	<p>不使用A/D转换器时，将ADA0M0.ADA0CE位清0，可以降低功耗。</p>	p. 459 □
					p. 469 □

章节	分类	功能	功能细节	注意事项	页码
第十三章	软件	A/D 转换器	ANIO至ANI15 引脚的输入范围	向ANIO引脚至ANI15引脚输入的电压应该在指定范围内。 如果大于等于AVREF0或小于等于AVss（即使在最大额定值范围内）的电压输入到这些引脚中的任意一个引脚，则此通道的转换值将不确定，并且其它通道的转换值也会受到影响。	p. 469 □
			抑制噪声的方法	为了保证10位分辨率，必须有效防止噪声对ANIO引脚至ANI15引脚的干扰。 噪声的影响随着模拟输入源输出阻抗的增大而加强。 为了减小噪声，建议连接一个外部电容器，方式如图13-12所示。	p. 469 □
			复用I/O	模拟输入引脚（ANIO至ANI15）可以复用为端口引脚。 当选择ANIO引脚至ANI15引脚的其中一个来执行A/D转换时，在转换期间不要执行读取输入端口或写入输出端口的指令，因为这样可能会降低转换的分辨率。 在A/D转换期间端口的某些引脚被设置作为输出，如果由于连接至端口引脚的外部电路干扰使得输出电流波动，输入引脚的转换分辨率也会下降。 如果有数字脉冲施加在与正被转换的输入信号引脚相邻的引脚上，由于耦合噪声的影响，A/D转换值可能与预期不同。 因此，不要在进行A/D转换引脚的邻近引脚上施加脉冲。	p. 469 □
			中断请求标志（ADIF）	即使ADA0S寄存器的内容改变，也不会清除中断请求标志（ADIF）。 因此，如果在A/D转换期间改变模拟输入通道，那么上次所选模拟输入引脚信号的转换结果可能被存储，且有可能在紧邻ADA0S寄存器重写之前，转换结束中断请求标志被置位。 如果写入ADA0S寄存器之后立即读取ADIF标志，那么即使新选择的模拟输入引脚的A/D转换还没有完成，也会置位ADIF标志。 A/D转换停止时，重新开始转换之前应该清除ADIF标志。	p. 470 □
	硬件		AVREF0引脚	(a) AVREF0引脚用作A/D转换器的电源，也为复用功能端口提供电源。 在使用备用电源的应用系统中，请确保提供至AVREF0引脚的电压与VDD相同，如图13-15所示。 (b) AVREF0引脚用作A/D转换器的参考电压引脚。 如果AVREF0引脚的输入电源具有高阻抗或者电流提供能力较弱，那么转换期间由于电流波动，参考电压可能会波动（尤其紧接在转换操作使能位ADA0CE被置为1之后）。 结果，转换精度可能会下降。 为了避免这种情况，建议在AVREF0引脚和AVss引脚之间连接一个电容器来抑制参考电压的波动，如图13-15所示。 (c) 如果连接至AVREF0引脚的电源有具有高直流电阻（例如，加入二极管），由于A/D转换电流引起电压下降，所以开始转换时的电压可能低于转换停止时的电压。	p. 471 □
	软件		读取 ADA0CRn 寄存器	当写入ADA0M0 至ADAM2、ADA0S、ADA0PFM或 ADA0PFT寄存器时，ADA0CRn寄存器的内容可能不确定。 在转换完成之后且写入ADA0M0至ADAM2、ADA0S、ADA0PFM或ADA0PFT 寄存器之前，读取转换结果。 同时，当响应外部/定时器触发时，ADA0CRn寄存器的内容也可能不确定。 在转换完成之后且响应下一次外部/定时器触发之前，读取转换结果。 不同于上述时序时，可能读取不到正确的转换结果。	p. 471 □

(18/37)

章节	分类	功能	功能细节	注意事项	页码
第十三章	软件	A/D 转换器	待机模式	在STOP模式下，由于A/D转换器停止工作，转换结果无效，因此能够降低功耗。STOP模式解除后，操作重新开始，但是STOP模式解除后的A/D转换结果无效。STOP模式解除后，需要使用A/D转换器时，在设置STOP模式之前或解除STOP模式之前，将ADA0M0.ADA0CE位清0，然后在解除STOP模式之后，将ADA0CE位设为1。 在IDLE1、IDLE2或副时钟操作模式下，操作继续。因此，为了降低功耗，将ADA0M0.ADA0CE位清0。在IDLE1和IDLE2模式下，由于不能够保持模拟输入引脚的电压值，所以IDLE1和IDLE2模式解除后，A/D转换结果无效。 设置进入IDLE1和IDLE2模式之前的转换结果是有效的。	p. 471 □
			高速转换模式	在高速转换模式下，振荡稳定期间，禁止触发信号的输入，也禁止对ADA0M0、ADA0M2、ADA0S、ADA0PFM和ADA0PFT寄存器执行重写操作。	p. 472 □
			A/D转换时间	A/D转换时间为稳定时间，转换时间，等待时间和触发响应时间的总和（关于这些时间的细节，参见表13-2 正常转换模式（ADA0HS1位= 0）下转换时间的选择和表13-3 高速转换模式（ADA0HS1位= 1）下转换时间的选择。 在正常转换模式下，A/D转换期间，如果对ADA0M0、ADA0M2、ADA0S、ADA0PFM以及ADA0PFT寄存器执行重写操作或者输入触发信号，则重新执行转换。但是，如果稳定时间结束时序与写入这些寄存器的操作发生冲突，或者稳定时间结束时序与触发信号输入发生冲突，那么将会重新插入64个时钟的稳定时间。如果在插入的稳定时间结束时序时刻再次发生冲突，则重新插入稳定时间。因此，不要将触发输入间隔和控制寄存器写入间隔设置为64个时钟或更少。	p. 472 □
			A/D转换结果的漂移	因为供应电压的波动或噪声的影响，A/D转换的结果可能会有变化。为了减小漂移，在程序中应用抑制措施，例如取A/D转换结果平均值。	p. 472 □
	硬件		A/D 转换转换结果磁滞特性	逐次比较类型A/D转换器在内部采样&保持电容器上保持输入电压，然后进行A/D转换。A/D转换完成后，模拟输入电压仍然保持在内部采样保持电容器上。结果，可能会产生下列现象。 • 当相同通道被用于A/D转换时，如果当前电压比先前的电压高或低，在转换结果受先前值影响的地方，可能出现磁滞特性。因此，即使对相同的电压进行转换，结果也可能不同。 • 当切换模拟输入通道时，在转换结果受先前通道值影响的地方，可能出现磁滞特性。这是因为A/D转换功能只有一个A/D转换器。因此，即使对相同的电压进行转换，结果也可能不同。	p. 472 □
第十四章	硬件	D/A 转换器	D/A 转换器	DAC0和DAC1共用 AVREF1 引脚。	p. 477 □
				DAC0和DAC1共用 AVss 引脚。D/A 转换器和 A/D转换器共用 AVss 引脚。	p. 477 □

章节	分类	功能	功能细节	注意事项	页码
第十四章	硬件	D/A 转换器	DA0M寄存器	在实时输出模式（DA0MDn 位 = 1）下的输出触发如下所示。 • 当n = 0时： INTTP2CC0信号（参见第七章 16位定时器/事件计数器 P（TMP））。 • 当n = 1时： INTTP3CC0信号（参见第七章 16位定时器/事件计数器 P（TMP））。	p. 478 □
			DA0CS0， DA0CS1寄存器	在实时输出模式（DA0M.DA0MDn 位 = 1）下，在INTTP2CC0/ INTTP3CC0信号产生以前，设置DA0CSn 寄存器。当INTTP2CC0/ INTTP3CC0 信号产生时，D/A 转换器开始工作。	p. 479 □
	软件	注意事项	实时输出模式下，触发信号发生时，不要改变DA0CSn寄存器的设定值。	p. 481 □	
			改变操作模式之前，确保将DA0M.DA0CEn位清为0。	p. 481 □	
			当使用P10/AN00 和P11/AN01引脚之一作为I/O端口，而另一个用作D/A输出引脚。在D/A输出期间，如果端口I/O电平不改变，则在应用系统中可以这样使用。	p. 481 □	
			确保AVREF0 = VDD = AVREF1 = 3.0 至 3.6 V。如果超出该范围，则操作无法保证。	p. 481 □	
			给AVREF1上电的时序和给AVREF1供电的时序相同。	p. 481 □	
			因为D/A转换器的输出阻抗高，所以ANOn引脚（n = 0，1）没有电流输出。如果连接2 MΩ或更小的电阻，需在电阻和ANOn引脚之间插入一个JFET（结型场效应晶体管）输入运算放大器。	p. 481 □	
	硬件		因为在STOP模式下D/A转换器停止操作，ANO0和ANO1引脚进入高阻抗状态，因而可以降低功耗。 但是，在IDLE1，IDLE2或副时钟工作模式下，操作继续。为了降低功耗，将DA0M.DA0CEn位清为0。	p. 481 □	
第十五章	软件	异步串行接口 A（UARTA）	CSIB4和 UARTA0模式切换	如果在发送或接收期间切换这些功能，则CSIB4和 UARTA0的发送/接收操作得不到保证。请确保禁止未使用的功能。	p. 482 □
			UARTA2和 I ² C00模式切换	如果在发送或接收期间切换这些功能，则UARTA2和 I ² C00的发送/接收操作得不到保证。请确保禁止未使用的功能。	p. 483 □
			UARTA1和 I ² C02模式切换	如果在发送或接收期间切换这些功能，则UARTA1和 I ² C02的发送/接收操作得不到保证。请确保禁止未使用的功能。	p. 484 □
			UAnOPT0 寄存器	在SBF 接收（UAnSRF 位= 1）期间，不要将UAnSRT 和UAnSTT 置位（为 1）。	p. 490 □
			SBF 接收	如果数据接收期间SBF被发送，会发生一个帧错误。	p. 500 □
				SBF接收（UAnSRF = 1）期间，不要将SBF接收触发位（UAnSRT）和SBF发送触发位（UAnSTT）设置为1。	p. 500 □
			连续发送	连续发送执行期间进行发送初始化时，请确保UAnSTR.UAnTSF位为0，然后再执行初始化。当UAnTSF位为1时进行初始化，则发送数据无法保证。	p. 503 □
			UART 接收	即使发生接收错误，也必须读取UAnRX寄存器。如果不读取UAnRX，则下一个数据接收期间发生溢出错误，并且接收错误会不断地连续发生。	p. 505 □
				认为接收操作期间只有一个停止位。忽略第二个停止位。	p. 505 □

章节	分类	功能	功能细节	注意事项	页码
第十五章	软件	异步串行接口 A (UARTA)	UART 接收	当接收完成时，在产生接收结束中断请求信号 (INTUAnR) 之后读取UAnRX寄存器，并对UAnPWR位或UAnRXE位清0。如果在INTUAnR 信号产生之前对UAnPWR位或UAnRXE位清0，则无法保证对UAnRX寄存器值的正确读取。	p. 505 □
				如果UARTAn的接收完成处理 (INTUAnR 信号产生) 和 UAnPWR位= 0 或 UAnRXE位 = 0发生冲突，则即UAnRX 寄存器内没有存储数据，也可能产生INTUAnR信号。 为了完成接收而无需等待INTUAnR信号的产生，请确保将UAnRIC寄存器的中断请求标志 (UAnRIF) 清除 (0)，在中断控制寄存器 (UAnRIC) 的中断屏蔽标志 (UAnRMK) 置位 (1) 之后，然后设置UAnPWR 位= 0或UAnRXE 位 = 0。	p. 505 □
			接收错误	发生INTUAnR信号时，必须读取UAnSTR寄存器来检查错误。	p. 506 □
				如果在连续接收期间发生接收错误中断，在下一个数据接收完成之前，必须读出UAnSTR寄存器的内容，然后进行错误处理。	p. 507 □
			LIN 功能	使用LIN功能时，将UAnCTL0寄存器的UAnPS1位和 UAnPS0位固定为00。	p. 508 □
			UAnCTL1 寄存器	在改写UAnCTL1寄存器之前，将UAnCTL0.UAnPWR位清除为0。	p. 511 □
			UAnCTL2 寄存器	在改写UAnCTL2寄存器之前，将UAnCTL0.UAnPWR 位清除为 0，或将 UAnTXE和UAnRXE 位清除为0。	p. 512 □
			波特率误差	发送过程中，波特率误差必须在接收端的误差容限之内。	p. 513 □
				接收时的波特率误差满足一定范围，该范围参见 (5) 接收中允许的波特率误差范围。	p. 513 □
			接收期间允许的波特率范围	接收期间的波特率误差，必须确保在允许的误差范围内，可用以下公式计算。	p. 515 □
			当提供给UARTAn的时钟停止工作时	当提供给UARTAn的时钟停止 (比如进入IDLE1、IDLE2或STOP 模式) 时，则操作停止，每一个寄存器都保留时钟供应停止前的瞬间值。TXDAn引脚输出也保持时钟供给停止前的瞬间值。然而，恢复时钟供给后，不能保证正常操作。因此，在时钟恢复之后，应该将UAnCTL0.UAnPWR位、UAnCTL0.UAnRXEn位和UAnCTL0.UAnTXEn位设置为000，对电路进行初始化。	p. 518 □
			RXDA1 引脚 KR7 引脚	请确保不要同时使用RXDA1和KR7引脚。要使用RXDA1引脚，就不要使用KR7引脚。若使用KR7引脚，就不要使用RXDA1引脚 (建议将PFCE91位置为1，并清除PFCE91位为0)。	p. 518 □
			使用DMA 执行发送数据和接收数据的传输	UARTAn中，不会产生由通信错误引起的中断。使用DMA传输功能进行发送数据和接收数据的传输时，即使传输期间发生错误 (奇偶校验错误，溢出错误，帧错误)，也不执行错误处理。或者在DMA传输完成之后，读取UAnSTR寄存器检查是否有错误，或者在通信期间读取UAnSTR寄存器来检查错误。	p. 518 □

章节	分类	功能	功能细节	注意事项	页码
第十五章	软件	异步串行接口 A (UARTA)	启动UARTAn	按以下步骤启动UARTAn。 <1> 设置 UAnCTL0.UAnPWR 位为 1。 <2> 设置该端口。 <3> 设置 UAnCTL0.UAnTXE 位为 1, UAnCTL0.UAnRXE 位为 1。	p. 518 □
			停止UARTAn	按以下步骤停止UARTAn。 <1> 设置UAnCTL0.UAnTXE 位为0, UAnCTL0.UAnRXE位为0。 <2> 设置UAnCTL0.UAnPWR位为0 (如果端口设置不改变, 则不存在该问题)。	p. 518 □
			发送模式	发送模式下 (UAnCTL0.UAnPWR 位 = 1且UAnCTL0.UAnTXE 位 = 1), 因为写入UAnTX寄存器会开始发送, 所以不要用软件向该寄存器写入相同的值。要连续发送相同的值, 则重写相同的值。	p. 518 □
			连续发送	在连续发送中, 从停止位到下一个起始位的通信速率要比平常延长2个以上基础时钟。但是, 接收端通过检测起始位进行时序的初始化, 因此接收结果不受影响。	p. 518 □
第十六章	软件	3线可变量度串行I/O (CSIB)	CSIB4和UARTA0模式切换	如果在发送或接收期间切换这些功能, 则CSIB4和 UARTA0的发送/接收操作得不到保证。请确保禁止未使用的功能。	p. 519 □
			CSIB0和I ² C01模式切换	如果在发送或接收期间切换这些功能, 则CSIB0和I ² C01的发送/接收操作得不到保证。请确保禁止未使用的功能。	p. 520 □
			CBnCTL0 寄存器	要强制挂起发送/接收, 将CBnPWR位清为0, 而不是清除CBnRXE 和 CBnTXE 位。此时, 时钟输出停止。	p. 523 □
				请确保将第3位和第2位清除为“0”。	p. 525 □
			CBnCTL1 寄存器	仅当CBnCTL0.CBnPWR位= 0时, CBnCTL1 寄存器才可以重写。	p. 526 □
				设置通信时钟 (f _{CCLK}) 为 8 MHz或更慢。	p. 526 □
			CBnCTL2 寄存器	仅当CBnCTL0.CBnPWR位= 0 或CBnTXE和CBnRXE位均为0时, CBnCTL2 寄存器才可以重写。	p. 527 □
			连续传输模式 (主模式, 发送模式)	连续发送模式下, 不产生接收完成中断请求信号 (INTCBnR)。	p. 544 □
			连续传输模式 (从模式, 发送模式)	连续发送模式下, 不产生接收完成中断请求信号 (INTCBnR)。	p. 553 □
			时钟时序	在单次传输模式下, CBnTSF位置1时对CBnTX寄存器的写入被忽略。这对传输过程的操作没有影响。 例如, 通过产生的INTCBnR信号启动DMA时, 若下一个数据被写入CBnTX寄存器, 因为CBnTSF位为1, 被写入数据不能传输。对于这样的应用, 应该使用连续传输模式, 而不要使用单次传输模式。	pp. 562, 563 □
			PRSM1至PRSM3 寄存器	在操作期间, 不要重写PRSMm寄存器。	p. 565 □
				在设置 BGCEm 位为1之前, 设置PRSMm寄存器。	p. 565 □
			PRSCM1至PRSCM3寄存器	在操作期间, 不要重写PRSCMm 寄存器。	p. 566 □
				在设置 PRSMm.BGCEm 位为1之前, 设置PRSCMm 寄存器。	p. 566 □

章节	分类	功能	功能细节	注意事项	页码
第十六章	软件	3线可变长度串行I/O (CSIB)	波特率的产生	设置f _{BRGm} 为8 MHz或更慢。	p. 566 □
			当使用DMA传输接收和发送的数据	使用DMA传输发送数据和接收数据时，即使串行传输期间发生溢出错误，也不执行错误处理。DMA传输完成后通过读取CBnSTR.CBnOVE位来检查是否发生溢出错误。	p. 567 □
			CBnCTL0 寄存器 CBnCTL1 寄存器 CBnCTL2 寄存器	关于操作期间 (CBnCTL0.CBnPWR位为1) 禁止重写的寄存器，如果操作期间进行了错误的重写，须将CBnCTL0.CBnPWR位清除为0，然后初始化CSIBn。 操作期间禁止重写的寄存器列举如下。 • 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. 567 □
			通信类型 2, 4	通信类型2和4 (CBnCTL1.CBnDAP 位 = 1) 中，在发生接收完成中断 (INTCBnR) 之后半个SCKBn时钟时，CBnSTR.CBnTSF位被清除。 单次传输模式下，通信期间 (CBnTSF 位 = 1) 写入的下一个待发送数据被忽略，并且不启动下次通信。同样如果设置为只接收通信 (CBnCTL0.CBnTXE 位 = 0, CBnCTL0.CBnRXE 位 = 1)，即使在通信期间读取接收数据，也不启动下一次通信。 因此，在通信类型2或4时 (CBnDAP 位 = 1) 使用单次传输模式，需特别注意以下情况。 • 若要启动下一次发送，在确认CBnTSF 位 = 0之后，将待发送数据写入CBnTX寄存器。 • 设置为只接收通信 (CBnTXE 位 = 0, CBnRXE 位 = 1) 时，若要连续执行下一次接收，在确认CBnTSF 位 = 0之后，读取CBnRX寄存器。 或者，使用连续传输模式来代替单次传输模式。尤其在用DMA传输时，推荐使用连续传输模式。	p. 567 □
第十七章	软件	I ² C总线	I ² C总线	要使用I ² C总线功能，需设置P38/SDA00, P39/SCL00, P40/SDA01, P41/SCL01, P90/SDA02和 P91/SCL02引脚作为串行发送/接收数据输入/输出引脚 (SDA00至 SDA02) 和串行时钟输入/输出引脚 (SCL00 至 SCL02)，并且将它们设置为N沟道漏极开路输出。	p. 568 □
			UARTA2和I ² C00模式切换	如果在发送或接收期间切换这些功能，则UARTA2和I ² C00的发送/接收操作得不到保证。请确保禁止未使用的功能。	p. 568 □
			CSIB0和I ² C01模式切换	如果在发送或接收期间切换这些功能，则CSIB0和I ² C01的发送/接收操作得不到保证。请确保禁止未使用的功能。	p. 569 □
			UARTA1和I ² C02模式切换	如果在发送或接收期间切换这些功能，则UARTA1和I ² C02的发送/接收操作得不到保证。请确保禁止未使用的功能。	p. 570 □
			IICC0至IICC2寄存器	当SCL0n信号线为高电平且SDA0n信号线为低电平时，如果I ² Cn 操作使能 (IICEn 位 = 1)，则立即检测到开始条件。为了避免这种情况，I ² Cn 操作使能以后，立刻使用位处理指令将LRELn 位设置为 1。	p. 577 □

章节	分类	功能	功能细节	注意事项	页码
第十七章	软件	I ² C总线	IICC0至IICC2寄存器	只有在主设备模式下，才可以将SPTn 位设置为1。但是，当IICRSVn位为0时，改变操作使能状态后检测到第一个停止条件前， SPTn 位必须设置为1并产生一个停止条件。详情参见 17.15 注意事项 。	p. 580 □
				当TRCn 位 = 1时，在第9个时钟期间WRELn位被置1且等待状态被取消，之后TRCn位被清除为0且SDA0n信号线被置为高阻状态。	p. 580 □
			IICS0至IICS2 寄存器	在下列状态时，禁止访问IICSn寄存器。详情参见 3.4.8 (2) 访问专用片上外围 I/O寄存器 。 • 当CPU运行于副时钟且主时钟振荡停止时 • 当CPU 运行于内部振荡时钟时	p. 581 □
				当WRELn位设置为1，且在第九个时钟通过TRCn 位 = 1取消等待状态时，TRCn 被清除为 0，而且SDA0n信号线变为高阻态。	p. 582 □
			IICF0至IICF2 寄存器	仅当操作停止时（IICEn 位 = 0），才可以写入STCENn 位。	p. 585 □
				当STCENn位 = 1时，紧随在I ² Cn 总线操作使能之后，不管总线的实际状态如何，总是认为处于总线释放状态（IICBSYn 位 = 0）。因此，要产生首个开始条件（STTn位 = 1），必需确认总线已经被释放，这样不会干扰其它通信。	p. 585 □
				仅当操作停止时（IICEn 位 = 0），才可以写入IICRSVn 位。	p. 585 □
			IICCL0至IICCL2寄存器	请确保将第7位和第6位清除为“0”。	p. 586 □
				不论OCKS0 寄存器的设置值如何，选择时钟是f _{xx} ，将OCKS0寄存器清除为00H（I ² C分频时钟停止状态）。	p. 588 □
				不论OCKS1 寄存器的设置值如何，选择时钟是f _{xx} ，将OCKS1寄存器清除为00H（I ² C分频时钟停止状态）。	p. 589 □
			开始条件	在和其它设备进行通信时，当V850ES/JJ3的IICn.IICEn位被设置为1时，根据通信线的状态，可能会检测到开始条件。SCL0n 和 SDA0n信号线是高电平时，请确保将IICn.IICEn位设置为1。	p. 593 □
			在仲裁期间和中断请求信号产生时序的状态	当IICn.WTIMn 位 = 1，在第九个时钟的下降沿处产生INTIICn信号。当WTIMn 位 = 0 且接收到扩展码的从地址时，在第八个时钟的下降沿处产生INTIICn信号（n = 0至2）。	p. 625 □
				当有可能发生仲裁时，为主设备操作将SPIEn位设置为1（n = 0至2）。	p. 625 □
			当IICFn.STCENn 位 = 0时	I ² C0n 操作使能之后，不管总线的实际状态如何，总是立即认为处于总线通信状态（IICFn.IICBSYn 位 = 1）。要在尚未检测到停止条件的情况下执行主设备通信，先产生一个停止条件，然后在开始主设备通信之前释放总线。 使用以下列步骤来产生停止条件。 <1> 设置 IICCLn 寄存器。 <2> 设置 IICn.IICEn 位。 <3> 设置 IICn.SPTn 位。	p. 631 □
			当IICFn.STCENn 位 = 1时	I ² C0n 操作使能之后，不管总线的实际状态如何，总是立即认为处于总线释放状态（IICFn.IICBSYn 位 = 0）。要产生首个开始条件（IICn.STTn位 = 1），必需确认总线已经被释放，这样不会干扰其它通信。	p. 631 □

章节	分类	功能	功能细节	注意事项	页码
第十七章	软件	I ² C总线	当其他设备之间的通信正在进行时	在其它设备之间正在进行通信时，当V850ES/JJ3的IICn.IICEn位被设置为1，根据通信线的状态，可能会检测到开始条件。SCL0n 和 SDA0n信号线是高电平时，请确保将IICn.IICEn位设置为1。	p. 631 □
			操作使能	使能操作（IICn.IICEn 位 = 1）之前，通过IICCLn、IICXn 和OCKSm寄存器决定工作时钟频率。要改变操作时钟频率，要将 IICn.IICEn 位清0一次。	p. 631 □
			IICn.STTn, SPTn 位	当 IICn.STTn 和 IICn.SPTn 位已经被设置为1，若不首先将其清0，则一定不要对其再次置位。	p. 631 □
			发送保留	如果传输已经被保留，设置IICn.SPIEn 位为1，这样，检测到停止条件就会产生一个中断请求信号。当产生一个中断请求信号后，向I ² Cn写入通信数据将会释放等待状态，然后传输开始。如果检测到停止条件后不产生中断请求信号，由于未产生中断请求，传输将暂停在等待状态。但是，无需为软件对IICSn.MSTSn 位的检测而将SPIEn 位置为 1。	p. 631 □
			单主系统下的主设备	释放I ² C0n总线（SCL0n、SDA0n 引脚 = 高电平），与参与通信的产品规格一致。 例如，当EEPROM输出低电平到SDA0n 引脚时，设置SCL0n引脚为输出口并从输出口输出时钟脉冲，直到SDA0n引脚为稳定的高电平。	p. 633 □
			多主系统下的主设备	确定总线释放状态（IICCLn.CLDn 位 = 1，IICCLn.DADn 位 = 1）已经维持了一段时间（比如：1帧）。当SDA0n 引脚稳定为低电平时，参考通信产品的规格书，决定是否释放I ² C0n 总线（SCL0n、SDA0n引脚 = 高电平）。	p. 634 □
				确认发送和接收格式与参与通信的产品规格一致。	p. 636 □
				当V850ES/JJ3单片机在多主系统中用作主设备时，每次发生INTIICn中断时都要读取IICSn.MSTSn 位以确定仲裁结果。	p. 636 □
				当V850ES/JJ3单片机在多主系统中用作从设备时，每次发生INTIICn中断时都要使用IICSn 和 IICFn 寄存器以决定下次处理的状态。	p. 636 □
			从等待取消	要取消从设备等待，向IICn写入 FFH，或将WRELn置位。	pp. 641至 643 □
			主等待取消	要取消主设备等待，向IICn写入 FFH，或将WRELn置位。	pp. 644至 646 □
第十八章	软件	DMA 功能（DMA 控制器）	DSA0至DSA3 寄存器	请确保将DSAnH寄存器的第14位至第10位清除为0。	p. 649 □
				当DMA传送禁止（DCHCn.Enn位=0）时，在以下时序设置DSAnH和 DSAnL寄存器。 <ul style="list-style-type: none"> 从复位后到第一个DMA 传送启动之间的时期 从通道被DCHCn.INITn位初始化后到DMA传送启动期间 从DMA传送完成（DCHCn.TCn位=1）到下一次DMA传送启动期间 	p. 649 □

章节	分类	功能	功能细节	注意事项	页码
第十八章	软件	DMA 功能 (DMA 控制器)	DSA0至DSA3 寄存器	当DSAn寄存器的值被读取时，两个16位寄存器，即DSAnH和DSAnL都被读出。如果读取操作与数值更新发生冲突，可能会读取正在被更新的数值（参见 18.13 注意事项 ）。	p. 649 □
				系统复位后，在启动DMA发送之前，设置DSAnH、DSAnL、DDAnH、DDAnL和DBCn寄存器。如果没有设置这些寄存器，DMA传输启动后的操作无法保证。	p. 649 □
			DDA0至DDA3 寄存器	请确保将DSAnH寄存器的第14位至第10位清除为0。	p. 650 □
				当DMA传送禁止（DCHCn.Enn位=0）时，在以下时序设置DDAnH和DDAnL寄存器。	p. 650 □
				<ul style="list-style-type: none"> 从复位后到第一个DMA 传送启动之间的时期 从通道被DCHCn.INITn位初始化后到DMA传送启动期间 从DMA传送完成（DCHCn.TCn位=1）到下一次DMA传送启动期间 	
				当DDAn寄存器的值被读取时，两个16位寄存器，即DDAnH 和 DDAnL被读出。如果读取操作与数值更新发生冲突，可能会读取正在被更新的数值（参见 18.13 注意事项 ）。	p. 650 □
				系统复位后，在启动DMA发送之前，设置DSAnH、DSAnL、DDAnH、DDAnL和DBCn寄存器。如果没有设置这些寄存器，DMA传输启动后的操作无法保证。	p. 650 □
			DBC0至DBC3 寄存器	当DMA传送禁止（DCHCn.Enn位=0）时，在下列时序设置DBCn寄存器。	p. 651 □
				<ul style="list-style-type: none"> 从复位后到第一个DMA 传送启动之间的时期 从通道被DCHCn.INITn位初始化后到DMA传送启动期间 从DMA传送完成（DCHCn.TCn位=1）到下一次DMA传送启动期间 	
				系统复位后，在启动DMA发送之前，设置DSAnH、DSAnL、DDAnH、DDAnL和DBCn寄存器。如果没有设置这些寄存器，DMA传输启动后的操作无法保证。	p. 651 □
			DADC0至 DADC3 寄存器	请确保将DADCn寄存器的第15位、第13位至第8位，以及第3位至第0位清为“0”。	p. 652 □
				当DMA传送禁止（DCHCn.Enn位=0）时，在下列时序设置 DADCn寄存器。	p. 652 □
				<ul style="list-style-type: none"> 从复位后到第一个DMA 传送启动之间的时期 从通道被DCHCn.INITn位初始化后到DMA传送启动期间 从DMA传送完成（DCHCn.TCn位=1）到下一次DMA传送启动期间 	
				DS0位指定传输数据的宽度，但是无法控制总线的宽度。如果设置了8位数据长度（DS0位=0），则不使用低位数据总线。	p. 652 □
				如果传送数据宽度被设置为16位（DS0位=1），则不能从奇地址开始传送。总是从低位地址的第一位为0的地址启动传输。	p. 652 □
				如果DMA传送是在一个片上外设 I/O寄存器（作为传送源或目的地）上执行的，请确保指定的传送宽度与寄存器宽度相同。例如，要在一个8位寄存器上执行DMA传送，务必指定8位宽度传输方式。	p. 652 □

章节	分类	功能	功能细节	注意事项	页码
第十八章	软件	DMA 功能 (DMA 控制器)	DCHC0至 DCHC3 寄存器	TCn位为只读位。	p. 653 □
				INITn 和STGn位是只写位。	p. 653 □
				请确保将DCHCn寄存器的第6位至第3位清除为“0”。	p. 653 □
				DMA 传送完成后（当产生末尾计数时），Enn位被清零，然后TCn位被置1。如果正在更新DCHCn寄存器的同时发生读取操作，可能会读出一个值表示“传送未完成且传送被禁止”（TCn位=0且Enn位=0）。	p. 653 □
			DTFR0至 DTFR3 寄存器	不要通过软件将DFn位设置为“1”。当DMA传送禁止时，如果发生了一个被指定作为启动DMA传送的触发原因，则将0写入该位，以清除DMA传送请求。	p. 654 □
				当DMA传送禁止（DCHCn.Enn位=0）时，在下列时序设置IFCn5至IFCn0位。 • 从复位后到第一个DMA 传送启动之间的时期 • 从通道被DCHCn.INITn位初始化后到DMA传送启动期间 • 从DMA传送完成（DCHCn.TCn位=1）到下一次DMA传送启动期间	p. 654 □
				在待机模式（IDLE1、IDLE2、STOP或sub-IDLE模式）中产生的中断请求信号不能启动DMA传送周期（DFn位也不会置1）。	p. 654 □
				通过IFCn5 至 IFCn0位选择DMA启动因素，当从选定的片上外设I/O产生中断时，不论DMA发送操作是允许还是禁止，DFn位都会置1。如果这种状态下DMA是使能的，则DMA传送立即启动。	p. 654 □
			传输目标之间的关系	在表18-2中标记为“x”的传送目的和源之间执行操作，操作结果将无法保证。	p. 656 □
			片上外围I/O的 请求	两个启动因素（软件触发和硬件触发）不能用于一个DMA通道。如果对一个DMA通道同时产生两个启动因素，则只有其中一个有效。并且不能确认有效启动因素。	p. 659 □
				在前一个DMA传送请求产生后或在前一个DMA传送周期中，产生的新传送请求将被忽略（清除）。	p. 659 □
				同一个DMA通道的传送请求时间间隔取决于DMA传送周期总线等待的设置、其他通道的开始状态或外部总线保持请求也会影响该间隔。特别如注意事项2所述，在DMA传送周期前或在DMA传送周期中，同一个通道产生的新的传送请求将被忽略。因此，对系统来说，同一个DMA通道的传送请求时间间隔必须足够长。当使用软件触发时，可以通过更新DBCn寄存器来检查之前的DMA传送周期完成情况。	p. 659 □
			VSWC 寄存器的 注意事项	在使用DMAC时，确保根据工作频率为VSWC寄存器设置一个合适的值。如果使用了VSWC寄存器的默认值（77H），或者给VSWC寄存器设置了一个不适当的值，则操作将不能正确执行（关于寄存器的详细情况，请参见3.4.8（1）（a）系统等待控制寄存器（VSWC））。	p. 665 □

章节	分类	功能	功能细节	注意事项	页码
第十八章	软件	DMA 功能 (DMA 控制器)	在内部RAM中执行DMA传输的注意事项	在内部RAM中执行以下指令时，不要执行与内部RAM（传送源/目的地）之间进行数据交互的DMA传送，因为CPU随后可能无法正确操作。 <ul style="list-style-type: none"> 位于内部RAM中的位操作指令（SET1、CLR1或 NOT1）。 访问内部RAM中非对齐地址的数据访问指令。 反之，当DMA传送的执行与内部RAM（传送源/目的地）之间数据交互时，不要执行以上两条指令。	p. 665 □
			读取 DCHCn.TCn 位的注意事项	当读取TCn位时，TCn位被清零。但是它不会自动清零，即使在一个特定时刻被读取也不会清零。为了正确地将TCn位清零，要增加下列处理过程。 (a) 通过轮询TCn位等待DMA传输完成时 确认TCn位已经被设置为1（在读取TCn位=1后），然后再读TCn位三次。 (b) 在断服务例程中的读取TCn位时 执行TCn位的读取操作三次。	p. 665 □
		DMA 传输初始化过程（将 DCHCn.INITn 位设置为1）		当执行DMA传输的通道要进行初始化时，即使INITn位被置为1，通道可能也无法初始化。要准确地初始化通道，执行以下两种方法之一。 (a) 暂时停止所有DMA通道的传输 使用以下<1>至<7>步，对执行DMA传输的通道进行初始化。 请注意，当执行第<5>步时，TCn位将被清除为0。确保其他处理程序不需要TCn位为1。 <1> 禁止中断（DI）。 <2> 除了被强制终止的DMA通道之外，读取所有通道的DCHCn.Enn位，然后将值传输到通用寄存器。 <3> 清除使用的DMA通道（包括被强制终止的通道）的Enn位为0。为了将最后一个DMA通道的 Enn位清零，需要执行两次清除指令。如果DMA传输目标（传输源/目的地）涉及到内部RAM，则执行三次清除指令。 示例： 如果使用通道0、1、2，则按以下顺序执行指令（如果传输目标不是内部 RAM）。 <ul style="list-style-type: none"> 将 DCHC0.E00 位清除为0。 将DCHC1.E11 位清除为0。 将DCHC2.E22 位清除为0。 再次将DCHC2.E22 位清除为0。 <4> 将要被强制终止的通道，其INITn位被设置为1。 <5> 读取所有未被强制终止的通道的TCn位。如果TCn位和第<2>步读取的Enn位均为1（逻辑与（AND）结果是1），将已保存的Enn位清零。 <6> 第<5>步的操作完成后，将Enn位的值写入DCHCn寄存器。 <7> 使能中断（EI）。	p. 666 □
				确保执行上述步骤中的第<5>步，以防止对某通道Enn位的非法设置，该通道的DMA传输在第<2>步和第<3>步之间正常完成。	p. 666 □
				(b) 重复执行INITn位设置，直到传输被正确地强制终止 <1> 如果该通道将要被强制终止（停止片上外设I/O的操作），则对来自该通道DMA请求源的请求不作响应。 <2> 使用DTFRn.DFn 位，确认待强制终止的通道的DMA传输请求未处于挂起状态。如果DMA传输请求被挂起，等待，直到挂起请求执行完成。 <3> 已经确认待强制终止的通道的DMA传输请求未处于挂起状态，则将Enn位清零。	p. 667 □

章节	分类	功能	功能细节	注意事项	页码
第十八章	软件	DMA 功能 (DMA 控制器)	DMA 传输初始化过程 (将 DCHCn.INITn 位设置为1)	<p><4> 再次将待强制终止的通道的Enn位清零。 如果待强制终止的通道的传输目标 (传输源/目的地) 是内部RAM, 则再次执行该操作。</p> <p><5> 将最初设定给待强制终止的通道的初始传输数目拷贝到通用寄存器。</p> <p><6> 将要被强制终止的通道, 其INITn位被设置为1。</p> <p><7> 读取待强制终止的通道的DBCn 寄存器的值, 然后将其与在第<5>步中的拷贝值进行比较。如果两个值不匹配, 重复操作第<6>和<7>步。</p>	p. 667 □
			暂时停止DMA传输的过程 (清除 Enn位)	<p>使用按以下步骤执行DMA传送的停止和恢复。</p> <p><1> 对来自DMA请求源的传输请求 (停止片上外设I/O的操作) 不作响应。</p> <p><2> 使用DFn位 (检查DFn位是否等于 0), 确认DMA传输请求未处于挂起状态。如果请求处于挂起状态, 等待, 直到挂起DMA传输的请求执行完成。</p> <p><3> 如果确认没有DMA传输请求处于挂起状态, 将Enn位清零 (该操作停止DMA传输)。</p> <p><4> 将Enn位设置为1, 以恢复DMA传输。</p> <p><5> 对已经停止的DMA请求源的操作进行恢复 (启动片上外围I/O的操作)。</p>	p. 667 □
			存储器边界	在DMA传输过程中, 如果传输源或目的地的地址超出了DMA对象 (外部存储器、内部RAM或片上外设I/O) 的区域, 则操作将无法保证。	p. 667 □
			传输未对齐的数据	<p>不支持采用16位总线宽度进行非对齐数据的DMA传输。</p> <p>如果一个奇地址被指定为传输源或目的地, 则地址的最低有效位被强制假定为0。</p>	p. 667 □
			CPU的总线仲裁	<p>因为DMA控制器拥有比CPU更高优先级的总线权限, 在DMA传输期间, 发生的CPU访问将保持挂起状态直到DMA传输周期完成, 总线被释放给CPU为止。但是, CPU可以访问那些未执行DMA传送的外部存储器, 片上外围 I/O和内部RAM。</p> <ul style="list-style-type: none"> 当DMA传输在外部储存器和片上外围I/O之间执行时, CPU可以访问内部RAM。 当DMA传输在外部储存器和外部存储器之间执行时, CPU可以访问内部RAM和片上外围I/O。 	p. 668 □
			在DMA操作期间不可重写的寄存器/位	<p>当未进行DMA操作时, 按以下时序设置下列寄存器。</p> <p>[寄存器]</p> <ul style="list-style-type: none"> DSAnH、DSAnL、DDAnH、DDAnL、DBCn和DADCn 寄存器 DTFRn.IFCn5至DTFRn.IFCn0位 <p>[设置时序]</p> <ul style="list-style-type: none"> 从复位后到第一次DMA传输启动期间。 从通道初始化后到DMA传输启动期间。 从DMA传输完成后 (TCn 位=1) 到下一次DMA传输启动期间。 	p. 668 □
			DSAnH 寄存器 DDAnH 寄存器 DADCn 寄存器 DCHCn 寄存器	<p>请确保将以下寄存器的指定位设置为0。</p> <ul style="list-style-type: none"> DSAnH寄存器的第14位至第10位。 DDAnH寄存器的第14位至第10位。 DADCn寄存器的第15位, 第13位至第8位和第3位至第0位。 DCHCn寄存器的第6位至第3位。 	p. 668 □

章节	分类	功能	功能细节	注意事项	页码
第十八章	软件	DMA 功能 (DMA 控制器)	DMA 启动因素	不要使用相同启动因素开启两个或更多DMA通道。如果使用相同的启动因素开启了两个或更多的DMA通道,那么已经设定的DMA通道可能会再启动,或具有较低优先级的DMA通道将在具有较高优先级的DMA通道之前被应答。操作无法保证。	p. 668 □
			读取DSAn 和 DDAn 寄存器的值	在DMA传输过程中,可能对DSAn和DDAn寄存器中正在更新的值(n=0至3)进行读取。 例如,当DMA传输源地址(DSAn寄存器)为0000FFFFH,且计数方向是递增(DADCn.SAD1和DADCn.SAD0位=00)时,如果先读取DSAnH寄存器,然后再读取DSAnL寄存器,则根据是否在读取DSAnH寄存器之后立即执行DMA传输,DSAnL寄存器的值会有所不同,描述如下。 (a) 如果在读取DSAn寄存器时未执行DMA传输 <1> 读取DSAnH寄存器的值: DSAnH = 0000H <2> 读取 DSAnL寄存器的值: DSAnL = FFFFH (b) 如果在读取DSAn寄存器时发生DMA传输 <1> 读取DSAnH寄存器的值: DSAnH = 0000H <2> 发生DMA传输 <3>DSAn寄存器递增: DSAn = 00100000H <4> 读取 DSAnL寄存器的值: DSAnL = 0000H	p. 669 □
第十九章	软件	中断/异常处理功能	非屏蔽中断	关于对不可屏蔽中断请求信号(INTWDT2)执行的不可屏蔽中断服务,参见 19.2.2 (2) 来自INTWDT2信号 。	p. 674 □
				在不可屏蔽中断服务期间,如果EP位和NP位被LDSR指令改变时,为了在RETI恢复期间正确恢复PC和PSW,必须在紧邻RETI指令之前,使用LDSR指令将EP位设置为0,并将NP位置回1。	p. 677 □
			可屏蔽中断	在可屏蔽中断服务期间,如果EP位和NP位被LDSR指令改变时,为了在RETI指令恢复期间正确恢复PC和PSW,必须在紧邻RETI指令之前,使用LDSR指令将EP位和NP位都清除为0。	p. 681 □
			多重中断	为了执行多重中断服务,EIPC和EIPSW寄存器的值必须在执行EI指令前进行保存。当从多重中断服务中返回时,在执行DI指令之后,恢复EIPC和EIPSW寄存器的值。	pp. 683至 685 □
			中断控制寄存器	禁止中断(DI)或者屏蔽中断来读取xxICn.xxIFn位。如果在使能中断或是中断未被屏蔽的时候,读取xxIFn位,当响应中断和对该位的读取发生冲突时,可能读不到正确的值。	p. 686 □
				如果一个中断请求型号被响应,xxIFn标志被硬件自动复位。	p. 686 □
			IMR0至IMR4 寄存器	该设备文件将xxICn.xxMKn位定义为保留字。如果使用xxMKn的名称进行位操作,xxICn寄存器的内容被改写,而不是IMRm寄存器(结果,IMRm寄存器的内容也被改写)。	p. 688 □
				为了以8位或1位为单元读取IMR0至IMR4寄存器的第8位至第15位,需要将它们设定为IMR0H至IMR4H寄存器的第0位至第7位。	p. 689 □
				将IMR4寄存器的第13位至第15位和IMR3寄存器的第7位至第14位设置为1。如果改变这些位的设置,操作将无法保证。	p. 689 □

(30/37)

章节	分类	功能	功能细节	注意事项	页码
第十九章	软件	中断/异常处理功能	ISPR 寄存器	中断允许 (EI) 状态下, 如果在读取 ISPR 寄存器的同时响应一个中断, 读取的 ISPR 寄存器值可能是响应中断后其对应位被置位后的值。为了在响应中断前准确读出 ISPR 寄存器的值, 应在中断禁止 (DI) 的状态下读取。	p. 690 □
			从软件异常处理中恢复	在软件异常服务当中, 如果 EP 位和 NP 位被 LDSR 指令改变时, 为了在 RETI 指令的恢复之中正确恢复 PC 和 PSW, 必须在紧邻 RETI 指令之前, 使用 LDSR 指令将 EP 置回 1 以及将 NP 置回 0。	p. 693 □
			非法操作数定义	因为这些指令将来可能被指定为新用途, 建议不要使用非法操作码。	p. 695 □
			从异常陷阱中恢复	只有在执行非法操作码和 DBRET 指令之间的间隔内, 才可以访问 DBPC 和 DBPSW。	p. 696 □
			从调试陷阱中恢复	只有在执行 DBTRAP 和 DBRET 指令之间的间隔内, 才可以访问 DBPC 和 DBPSW。	p. 698 □
			INTF0, INTR0 寄存器	当引脚功能由外部中断功能 (复用功能) 改变为端口功能时, 可能会检测到一个边沿。因此, 首先要将 INTF0n 位和 INTR0n 位清除为 00, 然后再设置端口模式。	p. 700 □
				当这些寄存器不用于设置 NMI 或 INTP0 至 INTP3 引脚时, 请确保将 INTF0n 和 INTR0n 位清除为 00。	p. 700 □
			INTF3, INTR3 寄存器	当引脚功能由外部中断功能 (复用功能) 改变为端口功能时, 可能会检测到一个边沿。因此, 首先要将 INTF31 和 INTR31 位清除为 00, 然后再设置端口模式。	p. 701 □
				INTP7 引脚及 RXDA0 引脚是复用功能引脚。当使用该引脚作为 RXDA0 引脚时, 禁止复用功能 INTP7 引脚的边沿检测 (将 INTF3.INTF31 位及 INTR3.INTR31 位清除为 0)。当该引脚被用作 INTP7 引脚时, 停止 UARTA0 接收 (UA0CTL0.UA0RXE 位清除为 0)。	p. 701 □
				当这些寄存器不用于设置 INTP7 引脚时, 请确保将 INTF31 和 INTR31 位清除为 00。	p. 701 □
			INTF8, INTR8 寄存器	当引脚功能由外部中断功能 (复用功能) 改变为端口功能时, 可能会检测到一个边沿。因此, 首先要将 INTF80 位和 INTR80 位清除为 00, 然后再设定端口模式。	p. 702 □
				INTP8 引脚及 RXDA3 引脚是复用功能引脚。当将引脚作为 RXDA3 引脚使用时, 禁用 INTP8 复用功能引脚的边沿检测功能 (清零 INTF8.INTF80 位及 INTR8.INTR80 位)。当将引脚作 INTP8 引脚使用时, 停止 UARTA3 接收 (清零 UA3CTL0.UA3RXE 位)。	p. 702 □
				当这些寄存器不用于设置 INTP8 引脚时, 请确保将 INTF80 和 INTR80 位清除为 00。	p. 702 □
			INTF9H, INTR9H 寄存器	当引脚功能由外部中断功能 (复用功能) 改变为端口功能时, 可能会检测到一个边沿。因此, 首先要将 INTF9n 位和 INTR9n 位清除为 0, 然后再设定端口模式。	p. 703 □
				当这些寄存器不用于设置 INTP4 至 INTP6 引脚时, 请确保将 INTF9n 和 INTR9n 位清除为 00。	p. 703 □

章节	分类	功能	功能细节	注意事项	页码
第十九章	软件	中断/异常处理功能	NFC 寄存器	在采样时钟改变后，需要3个采样时钟来初始化数字噪声消除器。因此，如果在采样时钟改变后，有INTP3有效沿在这3个采样时钟内输入，可能会产生一个中断请求信号。 因此，当使用中断和DMA功能时，注意以下几点。 • 当使用中断功能时，在3个采样时钟过后，对中断请求标志（PIC3.PIF3位）清零后再使能中断。 • 当使用DMA功能（由INTP3启动）时，在3个采样时钟过后使能DMA。	p. 704 □
			NMI 引脚	NMI引脚复用功能作为P02引脚功能，而且在复位后作为正常端口引脚功能。要使能NMI引脚，通过PMC0寄存器使NMI引脚有效。NMI引脚的初始设置是“无边沿检测”。使用INTF0和INTR0寄存器来选择NMI引脚有效边沿。	p. 706 □
第二十章	软件	按键中断功能	KRM 寄存器	在KRM 寄存器被清为00H后，重写KRM 寄存器。	p. 708 □
				如果KRM寄存器被改变，则可能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）清除为0，再使能中断或清除屏蔽。	p. 708 □
			KR0至KR7 引脚	如果KR0 至 KR7中有任意某个引脚输入低电平，即使有下降沿输入其它引脚，也不会产生INTKR信号。	p. 708 □
			RXDA1 引脚 KR7 引脚	请确保不要同时使用RXDA1和KR7引脚。要使用RXDA1引脚，就不要使用KR7引脚。 若使用KR7引脚，就不要使用RXDA1引脚（建议将PFC91位置为1，并清除PFCE91位为0）。	p. 708 □
			使用按键中断功能	若要使用按键中断功能，需确保将端口引脚设置为按键返回引脚，然后允许KRM寄存器操作。要从按键返回引脚切换至端口引脚，需要禁止KRM寄存器操作，然后再设置端口引脚。	p. 708 □
第二十一章	软件	待机功能	PSC 寄存器	设置IDLE1、IDLE2、STOP或副IDLE模式之前，设置PSMR.PSM1和PSMR.PSM0位，然后再设置STP位。	p. 711 □
				当HALT模式解除后，NMI1M、NMI0M和INTM位的设置无效。	p. 711 □
				如果在NMI1M、NMI0M或INTM位设为1的同时把STP位也被设置为1，则对NMI1M、NMI0M或INTM位的设置无效。如果在设置IDLE1/IDLE2/STOP模式时有未屏蔽中断请求信号处于挂起状态，则应设置中断请求信号对应的位（NMI1M，NMI0M，或INTM）为1，然后再设置STP位为1。	p. 711 □
			PSMR 寄存器	请确保将第2位至第7位清除为“0”。	p. 712 □
				仅当PSC.STP位为1时，PSM0和PSM1位才有效。	p. 712 □
			OSTS 寄存器	不管是因为复位输入还是因为中断请求信号解除了STOP模式，在解除STOP模式之后的等待时间不包括时钟开始振荡之前（下图的“a”）的那段时间。	p. 713 □
				请确保将第3位至第7位清除为“0”。	p. 713 □
				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. 713 □
			HALT 模式	在HALT指令之后插入五个或更多的NOP指令。	p. 714 □

章节	分类	功能	功能细节	注意事项	页码
第二十一章	软件	待机功能	HALT 模式	如果在执行HALT指令时有一个未屏蔽的中断请求信号被挂起，则状态切换至HALT模式，但是HALT模式立即被该中断请求解除。	p. 714 □
			IDLE1 模式	为了设置IDLE1模式，需要将数据存储在PSC寄存器，在存储指令之后，应该插入五个或更多NOP指令。	p. 716 □
				如果在设置IDLE1模式时，有一个未屏蔽的中断请求信号被挂起，则IDLE1模式立即被这个等待响应的中断请求解除。	p. 716 □
			释放 IDLE1 模式	通过PSC.NMI1M、PSC.NMI0M和PSC.INTM位设置为1，一个中断请求信号将被禁止，该中断请求变为无效，并且IDLE1模式不会被解除。	p. 716 □
			IDLE2 模式	为了设置IDLE2模式，需要将数据存储在PSC寄存器，在存储指令之后，应该插入五个或更多NOP指令。	p. 718 □
				如果在设置IDLE2模式时，有一个未屏蔽的中断请求信号被挂起，则IDLE2模式立即被这个等待响应的中断请求解除。	p. 718 □
			释放IDLE2 模式	通过PSC.NMI1M、PSC.NMI0M、和PSC.INTM位设置为1（禁止）的中断请求信号将被禁止，并且IDLE2模式不会被释放。	p. 718 □
			STOP 模式	为了设置STOP模式，需要将数据存储在PSC寄存器，在存储指令之后，应该插入五个或更多NOP指令。	p. 721 □
				如果在设置STOP模式时，有一个未屏蔽的中断请求信号被挂起，则STOP模式立即被这个等待响应的中断请求解除。	p. 721 □
			释放STOP 模式	通过PSC.NMI1M、PSC.NMI0M和PSC.INTM位设置为1，一个中断请求信号将被禁止，该中断请求变为无效，并且STOP模式不会被解除。	p. 721 □
			副时钟操作模式	在对CK3位进行操作时，不要改变PCC.CK2至PCC.CK0位的设定值（建议用位操作指令对此位进行操作）。PCC寄存器的详情，参见 6.3（1）处理器时钟控制寄存器（PCC） 。	p. 725 □
				如果下列条件不满足，则改变CK2至CK0位以便条件能够满足，并设置为副时钟工作模式。 内部系统时钟（fCLK）> 副时钟（fXT=32.768kHz）× 4	p. 725 □
			释放副时钟操作模式	在对CK3位进行操作时，不要改变CK2至CK0位的设置值（建议用位操作指令对此位进行操作）。PCC寄存器的详情，参见 6.3（1）处理器时钟控制寄存器（PCC） 。	p. 725 □
				在停止主时钟之前，请确保停止PLL（PLLCTL.PLLON位 = 0）。	p. 726 □
				当CPU运行于副时钟而且主时钟振荡停止时，在发生等待时禁止访问寄存器。如果已经产生等待，只能通过复位解除（参见 3.4.8（2） ）。	p. 726 □
			副IDLE 模式	为了设置副IDLE模式，需要将数据存储在PSC寄存器，在存储指令之后，应该插入五个或更多NOP指令。	p. 727 □
				如果在设置副IDLE模式时，有一个未屏蔽的中断请求信号被挂起，则副IDLE模式立即被这个等待响应的中断请求释放。	p. 727 □
			释放副IDLE 模式	通过PSC.NMI1M、PSC.NMI0M和PSC.INTM位设置为1，一个中断请求信号将被禁止，该中断请求变为无效，并且副IDLE模式不会被释放。	p. 727 □

(33/37)

章节	分类	功能	功能细节	注意事项	页码
第二十一章	软件	待机功能	释放副IDLE 模式	当副IDLE 模式/低电压 副IDLE模式解除时，从产生解除副IDLE模式/低压副IDLE 模式的中断请求信号至模式被解除需要经过副时钟的12个周期（大约366 μ s）。	p. 727 ☐
			在副IDLE 模式中的操作状态	在停止主时钟之前，请确保停止PLL（PLLCTL.PLLON位 = 0）。	p. 728 ☐
				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。	p. 728 ☐
第二十二章	软件	复位功能	紧急操作模式	紧急操作模式下，除了中断功能、端口功能、WDT2和定时器M等能够在内部时钟振荡工作的模块的寄存器，不要访问其它片上外围I/O寄存器。此外，在该模式下禁止使用外部时钟对CSIB0至CSIB5 以及UARTA0进行操作。	p. 729 ☐
			复位功能	一个LVI电路引起的内部复位信号不能复位LVI电路本身。	p. 729 ☐
			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（发生复位时）发生冲突，则标志的设置优先。	p. 730 ☐
	硬件	RESET 引脚输入的硬件状态	当电源开启时，下列引脚可能会在复位期间暂时输出一个不确定的电平。 ● P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 引脚	p. 731 ☐	
			OCDM寄存器通过RESET引脚的输入来初始化。因此，必须注意，在复位解除之后OCDM.OCDM0位被清除之前，如果有高电平输入P05/DRST引脚，可能会进入片上调试模式。详情参见 第四章 端口功能。	p. 731 ☐	
第二十三章	软件	时钟监视器	CLM 寄存器	一旦CLME位设置为1，除非复位，不能以任何其它方式将其清0。	p. 739 ☐
				当时钟监控器引发复位时，CLME位被清为0，而RESF.CLMRF位被置为1。	p. 739 ☐
			内部振荡器	通过设置RCM.RSTOP位为1，可以停止内部振荡器。	p. 740 ☐
				当内部振荡器停止时，时钟监视器也同样停止。	p. 740 ☐
				内部振荡器不能通过软件停止。	p. 740 ☐
第二十四章	软件	低电压检测器（LVI）	LVIM 寄存器	当LVION 和 LVIMD 位为1时，低电压检测器不能停止，除非低电压检测之外的其他复位源产生复位信号。	p. 744 ☐
				当 LVION 位设置为1时，LVI 电路内的比较器开始工作。	p. 744 ☐
				在LVION位置位后，使用LVIF位检测电压前，应该由软件控制等待 0.2 ms或更长的时间。	p. 744 ☐
				请确保将第6位至第2位清除为“0”。	p. 744 ☐
		LVIS 寄存器	该寄存器不能写入，除非在LVIM.LVION 和LVIM.LVIMD 位被设置为1之后，低电压检测之外的其他复位源产生复位请求。	p. 745 ☐	
			请确保将第7位至第1位清除为“0”。	p. 745 ☐	
		用作内部复位信号	如果LVIMD 位被设为1，LVIM 和 LVIS 寄存器的内容不能改变，除非产生一个LVI 之外的复位请求。	p. 746 ☐	
		用作中断	当产生INTLVI信号后，通过LVIM.LVIF位确认是否因为供电电压下降或上升经过检测电压而产生INTLVI信号。	p. 747 ☐	
PEMU1 寄存器	此位不会被自动清除。	p. 749 ☐			

章节	分类	功能	功能细节	注意事项	页码
第二十五章	软件	CRC功能	CRCD寄存器	在下列状态时，禁止访问CRCD 寄存器。 详情参见 3.4.9 (2) 访问专用片上外围 I/O寄存器 。 <ul style="list-style-type: none"> 当CPU运行于副时钟且主时钟振荡停止时 当CPU 运行于内部振荡时钟时 	p. 751 □
第二十七章	硬件	稳压器	稳压器	使用稳压器应设置 $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$ 。	p. 755 □
第二十七章	Flash 存储器	Flash 存储器	FLMD1引脚	将FLMD1引脚连接至Flash编程器或通过板上的下拉电阻接地。	pp. 763 至 765 □
			PG-FP4	连接图27-6所示的引脚，或然后通过板上的下拉电阻连接至GND。 无法通过flash编程器的CLK引脚提供时钟。在板上创建一个振荡器来提供时钟。	p. 765 □
			FA-144GJ-UEN-A	请确保将REGC引脚通过4.7 μ F（推荐值）电容接地。	pp. 766, 767 □
				无法通过flash编程器的CLK引脚提供时钟。在板上创建一个振荡器来提供时钟。	pp. 766, 767 □
				将FLMD1引脚连接至Flash编程器或通过板上的下拉电阻接地。	p. 769 □
				通过在flash存储器写入适配器（虚线内）上创建一个振荡器提供时钟。	p. 769 □
				不要将高电平输入DRST引脚。	p. 769 □
			通信模式的选择	当选择了UART0时，在接收FLMD0脉冲后，基于专用Flash 存储器编程器发出的复位命令来计算接收时钟。	p. 771 □
			FLMD1引脚	如果在片上写入过程中和复位之后从别的设备立即输入 V_{DD} 信号到FLMD1引脚，则隔离此信号。	p. 773 □
			FLMD1引脚	请确保当复位释放时FLMD0引脚为0V。	p. 780 □
第二十八章	硬件，软件	片上调试功能	OCDM 寄存器	外部复位后，当DDI、DDO、DCK以及DMS引脚不作为片上调试引脚而作为端口引脚时，则必须执行下列所有操作。 <ul style="list-style-type: none"> 输入低电压到P05/INTP2/DRST 引脚。 设置ODCM0 位。 在这种情况下，进行下列操作。 <ol style="list-style-type: none"> OCDM0 位清除为0。 P05/INTP2/DRST引脚固定为低电平直到<1> 完成。 DRST引脚有片上下拉电阻。当OCDM0 标志被清除为0时，这个电阻被断开。	p. 786 □
					p. 786 □
	软件		注意事项 (DUC)	如果在RUN（程序执行）期间输入一个复位信号（来自目标系统或来自内部复位源的复位信号），则断点功能可能会误操作。	p. 787 □
				即使复位信号被屏蔽功能所屏蔽，如果在引脚输入一个复位信号也可能使得I/O寄存器（端口引脚）复位。	p. 787 □
				断点时引脚复位被屏蔽，且CPU和外围I/O不复位。如果在执行用户程序时，flash 存储器由DMM重写或由RAM监控功能读取之后立刻产生引脚复位或内部复位，那么CPU和外设I/O可能无法正确复位。	p. 787 □

章节	分类	功能	功能细节	注意事项	页码
第二十八章	硬件	片上调试功能	注意事项 (DUC)	在片上调试模式下，DDO 引脚强制设置为高电平输出。	p. 787 □
			注意事项 （非 DUC）	不要将用于调试的设备安装在批量生产的产品上，因为在调试过程中flash 存储器被重写过，所以不能保证flash 存储器的重写次数。 此外，不要将调试监控程序嵌入到批量生产的产品中。	p. 796 □
	软件			如果满足下列条件之一，则强制断点无法执行。 ● 中断被禁止（DI） ● 发给串行接口发出的中断被屏蔽，该中断用于MINICUBE2和目标器件之间的通信。 ● 在通过可屏蔽中断释放待机时，禁止进入待机模式。 ● MINICUBE2和目标器件之间的通信模式为UARTA0，且主时钟已被停止。	p. 796 □
				如果满足下列条件之一，则伪RRM功能和DMM功能无法操作。 ● 中断被禁止（DI） ● 发给串行接口发出的中断被屏蔽，该中断用于MINICUBE2和目标器件之间的通信。 ● 在通过可屏蔽中断释放待机时，禁止进入待机模式。 ● MINICUBE2和目标器件之间的通信模式为UARTA0，且主时钟已被停止。 ● MINICUBE2和目标器件之间的通信模式为UARTA0，并且用于通信的时钟和调试器中指定的时钟不同。	p. 797 □
				如果满足下列条件之一，则可以由伪RRM功能和DMM功能来解除待机模式。 ● MINICUBE2 和目标器件之间的通信模式为CSIB0或CSIB3。 ● MINICUBE2和目标器件之间的通信模式为UARTA0，且提供主时钟。	p. 797 □
				需要特定序列的外围I/O寄存器，不能用DMM功能写入。	p. 797 □
				如果分配给调试监控程序的空间被flash 存储器自编程所重写，那么调试器将不能再正常运行。	p. 797 □
				安全ID	在flash存储器被擦除后，全部区域被写入1。
第二十九章	硬件	电气特性	最大额定值	请确保所有供电电压都不超过最大额定值（最大值）。	p. 801 □
				不要将芯片的输出（或输入/输出）引脚直接互连，或直接连接至VDD、Vcc和GND。但是，漏极开路或集电极开路引脚可以直接互连。 如果输出引脚能够设置为高阻状态，并且外部电路的输出时序设计时已经考虑避免输出冲突，芯片和外部电路之间的输出引脚才可以直接相连。	p. 802 □

(36/37)

章节	分类	功能	功能细节	注意事项	页码	
第二十九章	硬件	电气特性	最大额定值	任何参数，哪怕瞬间超过最大额定值，都可能使产品性能受损。就是说，产品工作在最大额定值时处于物理损坏的边缘，所以，产品必须在不出最大额定值的条件下使用。DC和AC特性的额定值和工作条件代表在正常工作期间的性能保障范围。	p. 802 □	
			主时钟振荡器特性	以上所示的振荡频率仅表示振荡器特性。使用V850ES/JJ3，内部工作条件不要超出AC特性和DC特性的额定值。	p. 804 □	
				flash存储器所需的建立时间。使用OSTS寄存器来确保建立时间。	p. 804 □	
				在使用主时钟振荡器时，上图中虚线框部分的连线应遵从如下的连接方式，避免线间电容的负面影响。 ● 连接线越短越好。 ● 振荡器连线不要和其它的信号线互相交叉。 ● 布线时不要将振荡器连线靠近通过大波动电流的信号线。 ● 保持振荡器电容的地电位和Vss相同。 ● 不要将电容的接地点连接在大电流通过的地模式。 ● 不要从振荡器获取信号。	p. 804 □	
				当主时钟停止，设备运行于副时钟时，在由副时钟切换回主时钟之前，应该由程序来实现振荡稳定时间的等待。	p. 804 □	
	软件		关于谐振器的选择和振荡器常数，请客户自行评估，或向振荡器生产商申请对实现电路进行评估。	p. 804 □		
			副时钟振荡器特性	以上所示的振荡频率仅表示振荡器特性。使用V850ES/JJ3，内部工作条件不要超出AC特性和DC特性的额定值。	p. 806 □	
				在使用副时钟振荡器时，上图中虚线框部分的连线应遵从如下的连接方式，避免线间电容的负面影响。 ● 连接线越短越好。 ● 振荡器连线不要和其它的信号线互相交叉。 ● 布线时不要将振荡器连线靠近通过大波动电流的信号线。 ● 保持振荡器电容的地电位和Vss相同。 ● 不要将电容的接地点连接在大电流通过的地模式。 ● 不要从振荡器获取信号。	p. 806 □	

章节	分类	功能	功能细节	注意事项	页码
第二十九章	硬件	电气特性	副时钟振荡器特性	为了降低功耗，副时钟振荡器被设计为低振幅电路，而比起主时钟振荡器更容易受由于噪声引起的干扰。 在使用副时钟时需要特别注意布线的方式。	p. 806 □
				关于谐振器的选择和振荡器常数，请客户自行评估，或向振荡器生产商申请对实现电路进行评估。	p. 806 □
			数据保持特性	转换到STOP模式以及从STOP模式中恢复都必须在额定工作范围内进行。	p. 811 □
			AC 特性	如果由于电路配置使得负载电容超过50 pF，可插入一个缓冲器或其它手段使设备的负载电容等于或小于 50 pF。	p. 812 □
	软件	总线时序（复用总线模式）	当运行于 $f_{xx} > 20 \text{ MHz}$ 时，请确保插入地址保持等待和地址建立等待。	p. 814 □	
		总线时序（分离总线模式）	当运行于 $f_{xx} > 20 \text{ MHz}$ 时，请确保插入地址保持等待和地址建立等待和数据等待。	p. 819 □	
		I ² C总线模式	在开始条件时，第一个时钟脉冲在保持时间后产生。	p. 830 □	
			系统内部SDA0n（在SCL0n信号的V _{IHmin} .时）信号需要最短300 ns的保持时间，以占用SCL0n信号下降沿时的不确定区域。	p. 830 □	
			如果系统不延长 SCL0n信号低电平保持时间（t _{LOW} ），则只需要满足最大数据保持时间（t _{HD: DAT} ）。	p. 830 □	
			在正常模式I ² C总线系统中可以适用高速模式I ² C总线。这种情况下，设置高速模式I ² C总线使其满足以下条件。 ● 如果系统不延长SCL0n信号的低电平保持时间： t _{SU:DAT} ≥ 250 ns ● 如果系统延长SCL0n信号的低电平保持时间： 发送后续的数据位到SDA0n信号线优先于SCL0n信号线释放（t _{Rmax} + t _{SU: DAT} = 1,000 + 250 = 1,250 ns: 正常I ² C总线模式规范）。	p. 830 □	
		A/D 转换器	在A/D转换期间，不要设置（读/写）复用功能端口，否则转换分辨率可能会下降。	p. 831 □	
		编程特性	当首次对出厂产品进行写入时，不论是进行“擦除-写入”操作，还是“仅写入”操作，都被当作是一次重写。 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. 835 □	
附录A	软件	开发工具	RX850, RX850 Pro	要购买RX850 或 RX850 Pro，首先需要填写购买申请表并且签署许可协议。	p. 846 □
附录D	软件	指令集列表	指令集	对于通用寄存器reg1 和 reg3，不要指定相同的寄存器。	p. 871 □

附录 F 修订历史

F.1 本版的主要修订

页码	描述
全文	<ul style="list-style-type: none"> 改变全部产品的开发中状态 → 开发完成 μPD70F3743GJ-GAE-AX, 70F3744GJ-GAE-AX, 70F3745GJ-GAE-AX, 70F3746GJ-GAE-AX
pp. 259, 260	图 7-18. 外部触发脉冲输出模式的寄存器设置的修改
pp. 271, 272	图 7-22. 单次触发脉冲输出模式时的寄存器设置的修改
pp. 304, 305	图 7-36. 脉冲宽度测量模式时的寄存器设置的修改
pp. 353, 354	图 8-18. 外部触发脉冲输出模式时的寄存器设置的修改
pp. 365, 366	图 8-22. 单次触发脉冲输出模式时的寄存器设置的修改
pp. 405, 406	图 8-36. 脉冲宽度测量模式时的寄存器设置的修改
p. 448	表 13-2 正常转换模式下 (ADA0HS1 位 = 0) 的转换时间选择的修改
p. 449	表 13-3 高速转换模式下 (ADA0HS1 位 = 1) 的转换时间选择的修改
p. 457	图 13-3 转换操作时序 (连续转换) 的修改
p. 486	图 15-4. 异步串行接口 An 的功能框图的修改
p. 666	18.13 (4) (a) 暂时停止所有 DMA 通道的传输的修改
p. 668	18.13 (8) CPU 的总线仲裁的修改
p. 761	表 27-2 基本功能的修改
p. 761	表 27-3 安全功能的修改
p. 762	表 27-4 安全设置的修改
p. 770	图 27-7. 操作 flash 存储器的步骤的修改
p. 772	表 27-7 Flash 存储器控制命令的修改
p. 779	图 27-17. 标准自编程流程的修改
p. 780	表 27-10 Flash 功能列表的修改
p. 781	表 27-11 使用的内部资源的修改
p. 805	增加第 29 章 (i) KYOCERA KINSEKI CORPORATION: 晶体谐振器 (TA = -10 至 +70°C)
p. 805	增加第 29 章 (ii) Murata Mfg. Co. Ltd.: 陶瓷谐振器 (TA = -20 至 +80°C)
p. 837	增加第 31 章 推荐焊接条件

F.2 上一版的修订历史

下面所列是本版之前所有修订历史。“适用范围：”表示修订内容针对的章节。

版本	描述	适用范围：
第二版	增加 μ PD70F3743 和 μ PD70F3744	全文
	增加 3.4.4 (1) (a) 内部 ROM (384 KB)	第三章 CPU 功能
	增加 3.4.4 (1) (b) 内部 ROM (512 KB)	
	增加 3.4.4 (2) (a) 内部 RAM (32 KB)	
	增加 3.4.4 (2) (b) 内部 RAM (40 KB)	
	图 4-8 类型 D-2 的框图的修改	第四章 端口功能
	图 4-11 类型 E-2 的框图的修改	
	图 4-12 类型 E-3 的框图的修改	
	图 4-13 类型 G-1 的框图的修改	
	图 4-14 类型 G-2 的框图的修改	
	图 4-15 类型 G-3 的框图的修改	
	图 4-16 类型 G-5 的框图的修改	
	图 4-17 类型 G-6 的框图的修改	
	图 4-18 类型 G-12 的框图的修改	
	图 4-20 类型 L-2 的框图的修改	
	图 4-22 类型 N-2 的框图的修改	
	图 4-24 类型 U-1 的框图的修改	
	图 4-25 类型 U-5 的框图的修改	
	图 4-26 类型 U-6 的框图的修改	
	图 4-27 类型 U-7 的框图的修改	
	图 4-28 类型 U-8 的框图的修改	
	图 4-29 类型 U-9 的框图的修改	
	图 4-30 类型 U-10 的框图的修改	
	图 4-31 类型 U-11 的框图的修改	
	图 4-32 类型 U-12 的框图的修改	
	图 4-33 类型 U-13 的框图的修改	
	图 4-34 类型 U-14 的框图的修改	
	图 4-35 类型 U-15 的框图的修改	
	图 4-36 类型 AA-1 的框图的修改	
	15.3 (5) UARTAn 状态寄存器 (UAnSTR) 的修改	第十五章 异步串行接口 A (UARTA)
	27.1 特征 的修改	第二十七章 FLASH 存储器
	图 27-1 Flash 存储器映射的修改	
	图 28-4. 调试监控程序的内存空间分配的修改	第二十八章 片上调试功能
	表 B-1 V850ES/JJ3 和 V850ES/JJ2 之间的主要区别的修改	附录 B V850ES/JJ3 和 V850ES/JJ2 之间的主要区别
	增加附录 F 修订历史	附录 F 修订历史

详细信息请联系:

中国区

MCU 技术支持热线:

电话: +86-400-700-0606 (普通话)

服务时间: 9:00-12:00, 13:00-17:00 (不含法定节假日)

网址:

<http://www.cn.necel.com/> (中文)

<http://www.necel.com/> (英文)

[北京]

日电电子(中国)有限公司

中国北京市海淀区知春路 27 号

量子芯座 7, 8, 9, 15 层

电话: (+86) 10-8235-1155

传真: (+86) 10-8235-7679

[深圳]

日电电子(中国)有限公司深圳分公司

深圳市福田区益田路卓越时代广场大厦 39 楼

3901, 3902, 3909 室

电话: (+86) 755-8282-9800

传真: (+86) 755-8282-9899

[上海]

日电电子(中国)有限公司上海分公司

中国上海市浦东新区银城中路 200 号

中银大厦 2409-2412 和 2509-2510 室

电话: (+86) 21-5888-5400

传真: (+86) 21-5888-5230

[香港]

香港日电电子有限公司

香港九龙旺角太子道西 193 号新世纪广场

第 2 座 16 楼 1601-1613 室

电话: (+852) 2886-9318

传真: (+852) 2886-9022

2886-9044

上海恩益禧电子国际贸易有限公司

中国上海市浦东新区银城中路 200 号

中银大厦 2511-2512 室

电话: (+86) 21-5888-5400

传真: (+86) 21-5888-5230

[成都]

日电电子(中国)有限公司成都分公司

四川省成都市二环路南三段 15 号

天华大厦 608 室

电话: (+86)28-8512-5224

传真: (+86)28-8512-5334

[长春]

日电电子(中国)有限公司长春分公司

吉林省长春市朝阳区

西安大路 727 号中银大厦 A 座 1609 室

电话: (+86)431-8859-7533 / 8859-8533

传真: (+86)431-8680-2944

[大连]

日电电子(中国)有限公司长春分公司

大连市中山路 88 号天安国际大厦 2701 室

电话: (+86)411-8230-8815 / 8230-8825

传真: (+86)411-8230-8835