

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/IE2

32 位单片机

硬件

μ PD70F3713

μ PD70F3714

文档编号 U17716CA2V0UD00 (第二版)
发布日期 2008 年 06 月 N

© 日本电气电子株式会社 2005
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。

- 本文档信息发布于2008年06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项应用提供支持。

（注）

（1）本声明中的“本公司”是指日本电气电子株式会社（NEC Electronics Corporation）及其控股公司。

（2）本声明中的“本公司产品”是指所有由日本电气电子株式会社或为日本电气电子株式会社（如上定义）开发或制造的产品。

M8E 02.11-1

引言

读者对象	本手册适用于那些希望了解 V850ES/IE2 单片机的功能，并使用 V850ES/IE2 设计应用系统的用户。
目的	本手册用于帮助用户了解硬件功能。
编排组织	V850ES/IE2 用户手册分为两部分： 硬件（本手册）和 架构（V850ES 架构用户手册）。 各用户手册的组织如下：

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	<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/IE2 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 B “寄存器索引”**。
- 寄存器格式说明。
→关于位编号被装入尖括号“<>”内的位，其位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。
→请参阅 V850ES 架构用户手册。
- 要了解 V850ES/IE2 单片机的电气规范。
→请参阅**第 19 章 电气特性**。

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。

“<R>”标志表示主要修订之处。 将“<R>”复制到 PDF 文件中“查找”字段内，可以很方便搜索到修订的内容。

约定

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“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 ... xxxxH
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): $2^{30} = 1,024^3$
数据类型	字 32 位 半字 16 位 字节 8 位

相关文档

本手册中的相关文档可能包括早期版本。然而，早期版本并未标注如此。

V850ES/IE2 单片机相关文档

文档名称	文档编号
V850ES 架构用户手册	U15943E
V850ES/IE2 硬件用户手册	本手册

有关开发工具的文档 (用户手册)

文档名称	文档编号	
QB-V850ESIX2 (在线仿真器)	U17909E	
QB-MINI2 (具有编程功能的片上调试仿真器)	U18371E	
CA850 (3.00 版) (C 编译程序包)	操作	U17293E
	C 语言	U17291E
	汇编语言	U17292E
	连接指南	U17294E
PM+ (6.30 版) (项目管理器)	U18416E	
ID850QB (3.40 版) (集成调试器)	操作	U18604E
TW850 (2.00 版) (性能分析协调工具)	U17241E	
RX850 (3.20 版) (实时操作系统)	基础	U13430E
	安装	U17419E
	技术	U13431E
	任务调试器	U17420E
RX850 Pro (3.21 版) (实时操作系统)	基础	U18165E
	安装	U17421E
	技术	U13772E
	任务调试器	U17422E
AZ850 (3.30 版) (系统性能分析器)	U17423E	
PG-FP4 Flash 存储器编程器	U15260E	
PG-FP5 Flash 存储器编程器	U18865E	

目录

第一章 引言	14
1.1 概要	14
1.2 特性	15
1.3 应用程序	16
1.4 订货信息	16
1.5 引脚配置	17
1.6 功能块	19
第二章 引脚功能	22
2.1 引脚功能列表	22
2.2 引脚 I/O 电路以及不使用引脚的推荐连接	26
2.3 引脚 I/O 电路	28
第三章 CPU 功能	29
3.1 特性	29
3.2 CPU 寄存器组	30
3.2.1 程序寄存器组	31
3.2.2 系统寄存器组	32
3.3 操作模式	38
3.4 地址空间	39
3.4.1 CPU 地址空间	39
3.4.2 CPU 地址空间的绕回	40
3.4.3 内存映射	41
3.4.4 区域	43
3.4.5 地址空间使用建议	47
3.4.6 片上外围 I/O 寄存器	50
3.4.7 特殊寄存器	56
3.4.8 系统等待控制寄存器 (VSWC)	61
第四章 端口功能	62
4.1 特点	62
4.2 端口基本配置	62
4.3 端口配置	63
4.3.1 端口 0	67
4.3.2 端口 1	73
4.3.3 端口 2	85
4.3.4 端口 3	91
4.3.5 端口 4	98
4.3.6 端口 DL	105
4.4 各种端口设置的输出数据和读取值	108
4.5 使用复用功能时的端口寄存器设置	113
4.6 噪声消除器	117
4.7 注意事项	119
4.7.1 设置端口引脚的注意事项	119

4.7.2	端口 n 寄存器 (Pn) 位操作指令的注意事项	120
第五章	时钟发生器	121
5.1	概述	121
5.2	配置	122
5.3	控制寄存器	124
5.4	PLL 功能	130
5.4.1	概述	130
5.4.2	PLL 模式	130
5.4.3	时钟直通模式	130
5.5	操作	131
5.5.1	各时钟的操作	131
5.5.2	操作时序	132
5.6	时钟监视器	135
第六章	16 位定时器 / 事件计数器 P (TMP)	137
6.1	概要	137
6.2	功能	137
6.3	配置	138
6.4	寄存器	143
6.5	定时器输出操作	156
6.6	操作	157
6.6.1	间隔定时器模式 (TPnMD2 至 TPnMD0 位 = 000)	165
6.6.2	外部事件计数模式 (TPkMD2 至 TPkMD0 位 = 001)	177
6.6.3	外部触发脉冲输出模式 (TPmMD2 至 TPmMD0 位 = 010)	186
6.6.4	单次脉冲输出模式 (TPmMD2 至 TPmMD0 位 = 011)	199
6.6.5	PWM 输出模式 (TPmMD2 至 TPmMD0 位 = 100)	206
6.6.6	自由运行定时器模式 (TPnMD2 至 TPnMD0 位 = 101)	215
6.6.7	脉宽测量模式 (TPkMD2 至 TPkMD0 位 = 110)	232
第七章	16 位定时器 / 事件计数器 Q (TMQ)	238
7.1	概要	238
7.2	功能	239
7.3	配置	239
7.4	寄存器	244
7.5	定时器输出操作	260
7.6	操作	261
7.6.1	间隔定时器模式 (TQnMD2 至 TQnMD0 位 = 000)	269
7.6.2	外部事件计数模式 (TQ0MD2 至 TQ0MD0 位 = 001)	281
7.6.3	外部触发脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 010)	291
7.6.4	单次脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 011)	305
7.6.5	PWM 输出模式 (TQ0MD2 至 TQ0MD0 位 = 100)	314
7.6.6	自由运行定时器模式 (TQnMD2 至 TQnMD0 位 = 101)	325
7.6.7	脉宽测量模式 (TQ0MD2 至 TQ0MD0 位 = 110)	345
第八章	16 位间隔定时器 M (TMM)	351
8.1	概述	351

8.2	配置	352
8.3	控制寄存器	353
8.4	操作	354
8.4.1	间隔定时器模式	354
8.5	注意事项	358
第九章 马达控制功能		359
9.1	功能概述	359
9.2	配置	360
9.3	控制寄存器	364
9.4	操作	377
9.4.1	系统概述	377
9.4.2	死区时间控制（负相位波形信号的产生）	382
9.4.3	中断选择功能	389
9.4.4	重写具有传送功能的寄存器的操作	396
9.4.5	A/D 转换开始触发器信号输出的 TMP1 调谐操作	414
9.4.6	A/D 转换开始触发输出功能	417
第十章 看门狗定时器功能		422
10.1	功能	422
10.2	配置	423
10.3	控制寄存器	424
10.4	操作	425
10.5	注意事项	425
第十一章 A/D 转换器 0 和 1		426
11.1	特性	426
11.2	配置	427
11.3	控制寄存器	431
11.4	操作	438
11.4.1	基本操作	438
11.4.2	操作模式和触发模式	439
11.5	软件触发模式中的操作	450
11.5.1	连续选择模式的操作	450
11.5.2	连续扫描模式的操作	453
11.5.3	单次脉冲选择模式的操作	454
11.5.4	单次脉冲扫描模式的操作	456
11.6	定时器触发模式中的操作	457
11.6.1	连续选择模式 / 单次脉冲选择模式的操作	458
11.6.2	连续扫描模式 / 单次脉冲扫描模式的操作	460
11.7	外部触发模式中的操作	461
11.7.1	连续选择模式 / 单次脉冲选择模式的操作	462
11.7.2	连续扫描模式 / 单次脉冲扫描模式的操作	464
11.8	内部等效电路	465
11.9	操作时的注意事项	467
11.9.1	停止转换操作	467
11.9.2	定时器 / 外部触发间隔	467
11.9.3	待机模式中的操作	467

11.9.4	定时器触发模式中的定时器中断请求信号	468
11.9.5	稳定时间内重新转换时的开始触发输入	468
11.9.6	A/D 转换结果的误差	468
11.9.7	A/D 转换结果滞后特性	468
11.9.8	设置单脉冲模式和软件触发模式的限制	469
11.10	A/D 转换器特征表的阅读方法	470
第十二章	异步串行接口 A (UARTA)	474
12.1	特性	474
12.2	配置	475
12.3	控制寄存器	477
12.4	中断请求信号	482
12.5	操作	483
12.5.1	数据格式	483
12.5.2	UART 发送	485
12.5.3	连续发送的过程	486
12.5.4	UART 接收	488
12.5.5	接收错误	489
12.5.6	校验类型和操作	490
12.5.7	接收数据的噪声过滤器	491
12.6	专用波特率发生器	492
12.7	注意事项	499
第十三章	3 线长度可变串行 I/O (CSIB)	500
13.1	特性	500
13.2	配置	501
13.3	控制寄存器	503
13.4	操作	510
13.4.1	单独传输模式 (主机模式, 发送模式)	510
13.4.2	单独传输模式 (主机模式, 接收模式)	512
13.4.3	单独传输模式 (主机模式, 发送/接收模式)	514
13.4.4	单独传输模式 (从机模式, 发送模式)	516
13.4.5	单独传输模式 (从机模式, 接收模式)	518
13.4.6	单独传输模式 (从机模式, 发送/接收模式)	520
13.4.7	连续传输模式 (主机模式, 发送模式)	522
13.4.8	连续传输模式 (主机模式, 接收模式)	524
13.4.9	连续传输模式 (主机模式, 发送/接收模式)	527
13.4.10	连续传输模式 (从机模式, 发送模式)	531
13.4.11	连续传输模式 (从机模式, 接收模式)	533
13.4.12	连续传输模式 (从机模式, 发送/接收模式)	536
13.4.13	接收错误	540
13.4.14	时钟时序	541
13.5	输出引脚	543
第十四章	中断/异常处理功能	544
14.1	特性	544
14.2	不可屏蔽中断	548
14.2.1	操作	549

14.2.2	恢复	550
14.2.3	不可屏蔽中断状态标志 (NP)	551
14.3	可屏蔽中断	552
14.3.1	操作	552
14.3.2	恢复	554
14.3.3	可屏蔽中断的优先级	555
14.3.4	中断控制寄存器 (xxICn)	559
14.3.5	中断屏蔽寄存器 0 至 3 (IMR0 至 IMR3)	562
14.3.6	正在进行服务优先级寄存器	564
14.3.7	可屏蔽中断状态标志 (ID)	565
14.4	外部中断请求输入引脚 (INTP0 至 INTP6)	566
14.4.1	噪声消除	566
14.4.2	边沿检测	567
14.5	软件异常	568
14.5.1	操作	568
14.5.2	恢复	569
14.5.3	异常状态标志 (EP)	570
14.6	异常陷阱	571
14.6.1	非法操作码定义	571
14.6.2	调试陷阱	573
14.7	多重中断服务控制	575
14.8	CPU 的中断响应时间	577
14.9	CPU 不响应中断的时期	578
14.10	注意事项	578
第十五章	待机功能	579
15.1	概述	579
15.2	控制寄存器	581
15.3	HALT 模式	583
15.3.1	设置和操作状态	583
15.3.2	解除 HALT 模式	583
15.4	IDLE 模式	585
15.4.1	设置和操作状态	585
15.4.2	解除 IDLE 模式	585
15.5	STOP 模式	587
15.5.1	设置和操作状态	587
15.5.2	解除 STOP 模式	587
15.6	保证振荡稳定时间	589
第十六章	复位功能	590
16.1	概述	590
16.2	检测复位源的寄存器	590
16.3	操作	592
16.3.1	通过 RESET 引脚实现的复位操作	592
16.3.2	通过看门狗定时器 (WDT) 溢出实现的复位操作 (WDTRES)	594
16.3.3	低电压监测器 (LVI)	595
16.3.4	上电清除电路 (POC)	603

第十七章	稳压器	605
17.1	概述	605
17.2	操作	606
第十八章	FLASH 存储器	607
18.1	特点	607
18.2	存储器配置	608
18.3	功能概述	609
18.4	通过专用 Flash 存储器编程器重写	613
18.4.1	编程环境	613
18.4.2	通信模式	614
18.4.3	Flash 存储器控制	619
18.4.4	通信模式的选择	620
18.4.5	通信命令	621
18.4.6	引脚连接	622
18.5	通过自编程进行重写（仅限 μ PD70F3714）	627
18.5.1	概述	627
18.5.2	特点	628
18.5.3	标准自编程流程	629
18.5.4	Flash 存储器功能	630
18.5.5	引脚处理	630
18.5.6	使用的内部资源	631
第十九章	电气特性	632
第二十章	封装图	651
<R>	第二十一章 推荐的焊接条件	652
附录 A	注意事项	653
A.1	sld 指令与中断请求之间的冲突限制	653
A.1.1	描述说明	653
A.1.2	相应措施	653
附录 B	寄存器索引	654
附录 C	指令集列表	659
C.1	常规指令	659
C.2	指令集（按字母顺序）	662
<R>	附录 D 修订历史	669
D.1	此版本中的主要修订内容	669

第一章 引言

V850ES/IE2 是日电子公司（NEC Electronics）的 V850 系列单片微控制器产品之一，主要应用在实时控制应用中的低功耗操作。

1.1 概要

V850ES/IE2 是一款 32 位单片机，它包括 V850ES CPU 内核和诸如 ROM/RAM、定时器/计数器、串行接口、看门狗定时器以及 A/D 转换器之类的外设功能。

除了实时响应特性和单时钟基本指令之外，V850ES/IE2 单片机还包含了诸如由硬件乘法器来实现的乘法指令、饱和运算指令、位操作指令等为数字伺服控制应用而优化的指令。此外，作为一种实时控制系统，V850ES/IE2 还允许操作诸如马达变频控制这样性价比极高的应用程序。

1.2 特性

- 最短指令执行时间: 50 ns (工作于内部 20 MHz 的时钟频率状态下)
- 通用寄存器: 32 位× 32 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。
 - 饱和运算指令 (具有上溢/下溢检测功能) 。
 - 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。
 - 带符号的读取指令。
- 内部存储器

产品编号	内部 ROM	内部 RAM
PD70F3713	64 KB (flash 存储器)	6 KB
PD70F3714	128 KB (flash 存储器)	6 KB
- 中断/异常:
 - 不可屏蔽中断: 1 个中断源 (外部: 无, 内部: 1) 。
 - 可屏蔽中断: 42 个中断源 (外部: 7, 内部: 35) 。
 - 软件异常: 32 个异常源。
 - 异常陷阱: 2 个异常源。
- I/O 线: I/O 端口: 39
- 定时器/计数器功能:
 - 16 位间隔定时器 M(TMM): 1 个通道。
 - 16 位定时器/事件计数器 Q (TMQ) : 2 个通道。
 - 16 位定时器/事件计数器 P (TMP) : 4 个通道。
 - 马达控制功能 (使用定时器 TMQ: 1 个通道 (TMQ1), TMP:1 个通道 (TMP1)) 。
 - 带有死区时间的 16 位精度 6 相位 PWM 功能: 1 个通道。
 - 高阻抗输出控制功能。
 - 定时器调谐操作功能。
 - 任意周期设置功能。
 - 任意死区时间设置功能。
 - 看门狗定时器: 1 个通道。
- 串行接口:
 - 异步串行接口 A (UARTA) 。
 - 3 线长度可变串行 I/O (CSIB) 。
 - CSIB: 1 个通道。
 - UARTA: 2 个通道。
- A/D 转换器: 10 位分辨率 A/D 转换器 (A/D 转换器 0 和 1) : 4 个通道× 2 个单位

- 时钟发生器: 可连接 2.5 MHz 谐振器（禁止外部时钟输入）。
通过 PLL 锁相环实现的倍频功能（固定乘以 8, $f_{xx} = 20 \text{ MHz}$ ）。
CPU 时钟分频功能 (f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/8$)。
- 省电功能: HALT/IDLE/ STOP 模式。
- 上电清除功能
- 低电压检测功能
- 自编程功能: 仅在 $\mu\text{PD70F3714}$ 中支持（在 $\mu\text{PD70F3713}$ 中不支持）。
- 封装: 64 引脚塑封 LQFP (14 × 14)
- 工作电压: $V_{DD} = EV_{DD} = 3.5 \text{ to } 5.5 \text{ V}$
 $AV_{DD0}, AV_{DD1} = 4.5 \text{ to } 5.5 \text{ V}$
- 工作环境温度: $T_A = -40 \text{ to } +85^\circ\text{C}$

1.3 应用系统

- 家用电器（例如变频空调、电冰箱、洗衣机等）。
- 工业设备（例如马达控制和通用变频器等）

1.4 订货信息

产品编号	封装	内部 ROM
$\mu\text{PD70F3713GC-8BS-A}$	64 引脚塑封 LQFP (14 × 14)	Flash 存储器 (64 KB)
$\mu\text{PD70F3714GC-8BS-A}$	64 引脚塑封 LQFP (14 × 14)	Flash 存储器 (128 KB)

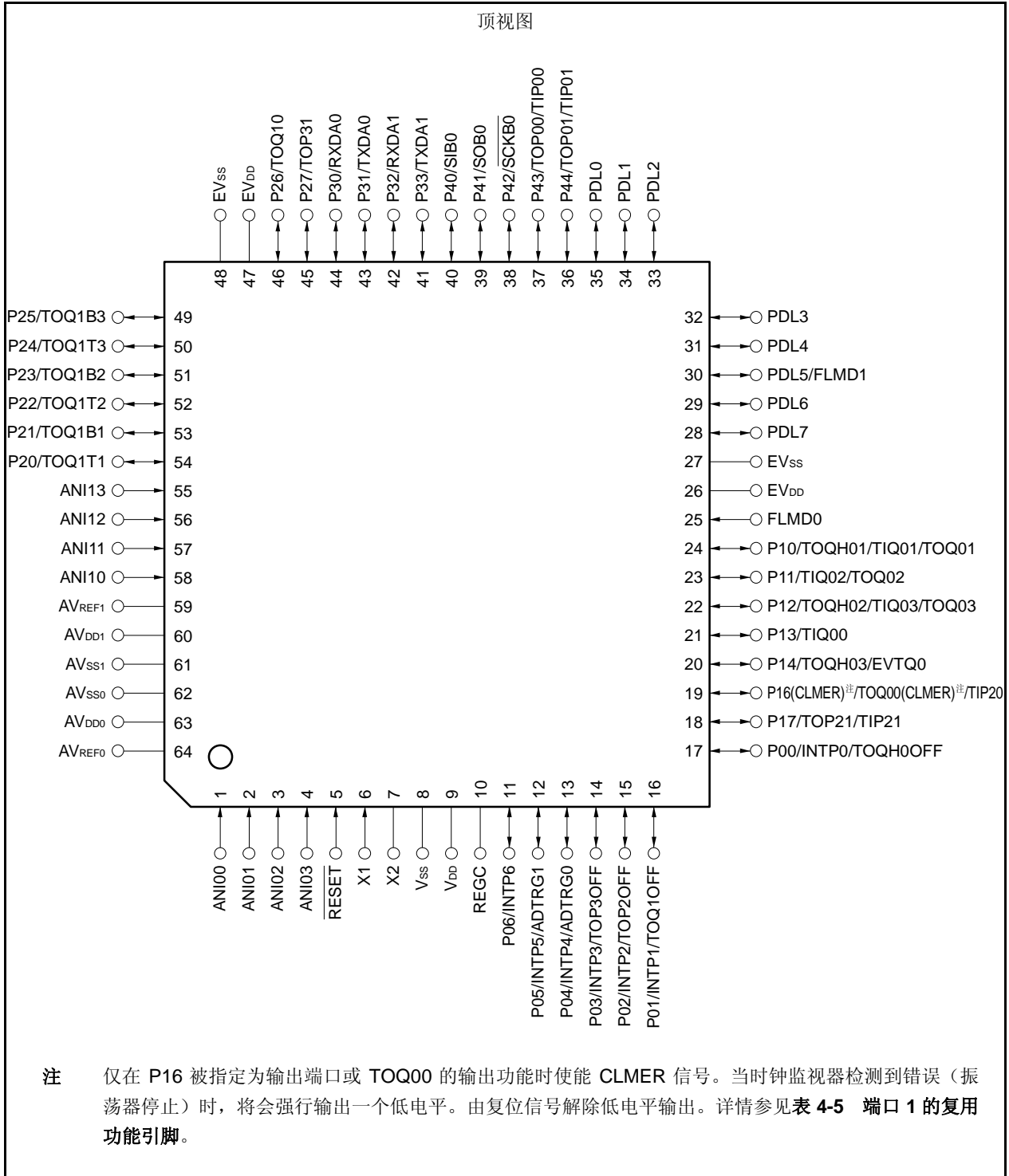
备注 产品编号末尾带有-A 的为无铅产品。

1.5 引脚配置

• 64 引脚塑封 LQFP (14 × 14)

μPD70F3713GC-8BS-A

μPD70F3714GC-8BS-A

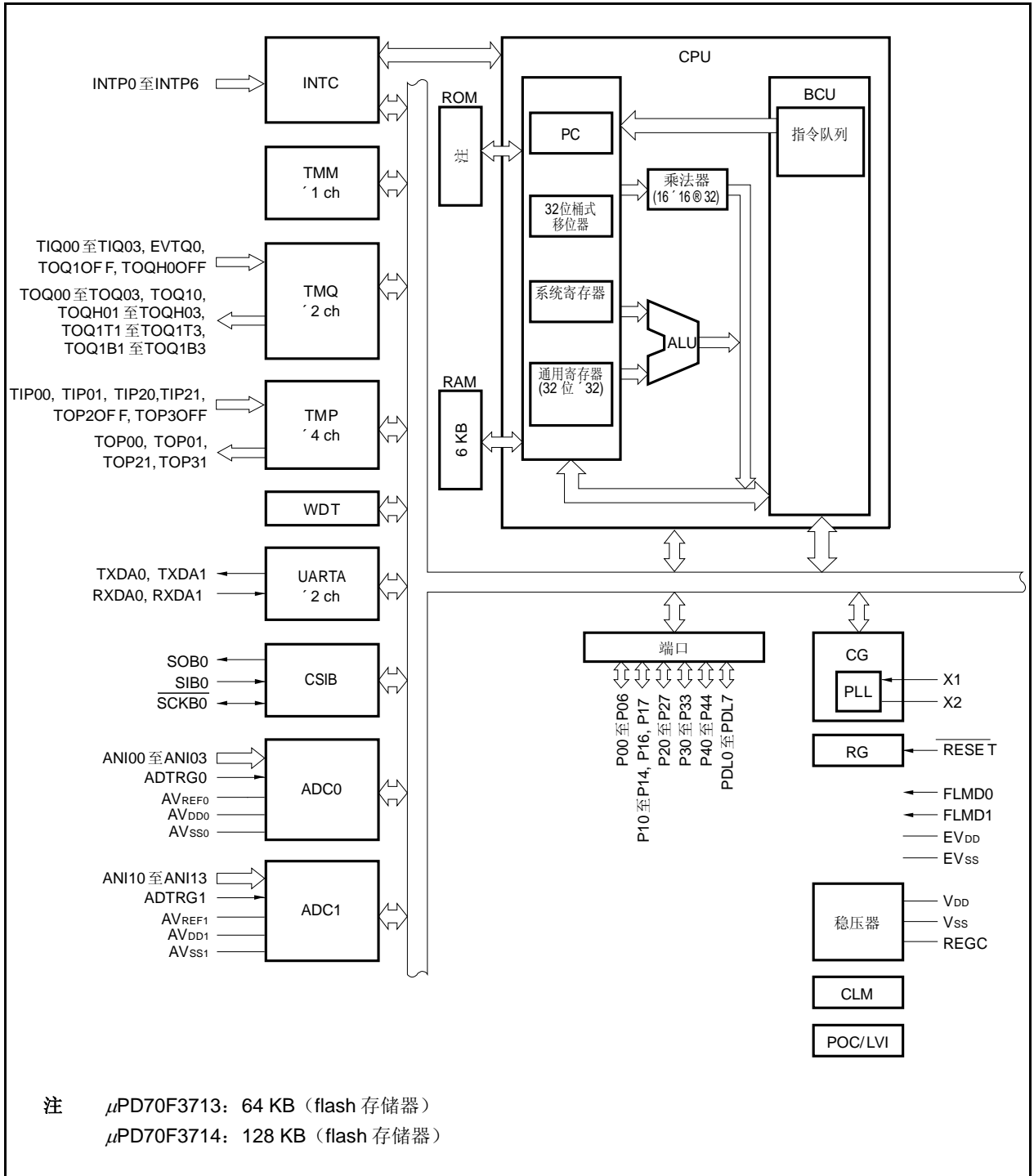


引脚定义

ADTRG0、ADTRG1:	A/D 触发信号输入	$\overline{\text{SCKB0}}$:	串行时钟
ANI00 至 ANI03,		SIB0:	串行输入
ANI10 至 ANI13:	模拟输入	SOB0:	串行输出
AVDD0、AVDD1:	模拟电源	TIP00、TIP01,	
AVREF0、AVREF1:	模拟参考电压	TIP20、TIP21,	
AVSS0、AVSS1:	模拟接地	TIQ00 至 TIQ03:	定时器触发输入
EVDD:	端口电压	TOP00、TOP01,	
EVSS:	端口接地	TOP21、TOP31,	
EVTQ0:	定时器事件计数输入	TOQ1B1 至 TOQ1B3,	
FLMD0、FLMD1:	Flash 编程模式	TOQ1T1 至 TOQ1T3,	
INTP0 至 INTP6:	外部中断输入	TOQ00 至 TOQ03,	
P00 至 P06:	端口 0	TOQ10,	
P10 至 P14、P16、	端口 1	TOQH01 至 TOQH03:	定时器输出
P17:			
P20 至 P27:	端口 2	TOP2OFF、TOP3OFF,	
P30 至 P33:	端口 3	TOQ1OFF、TOQH0OFF:	定时器输出关闭
P40 至 P44:	端口 4	TXDA0、TXDA1:	发送数据
PDLO 至 PDL7:	端口 DL	VDD:	电源
REGC:	稳压器控制	VSS:	接地
$\overline{\text{RESET}}$:	复位	X1、X2:	时钟振荡器引脚
RXDA0、RXDA1:	接收数据		

1.6 功能块

(1) 内部结构框图



(2) 内部器件**(a) CPU**

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

其它片内固有硬件诸如一个乘法器（16 位×16 位→32 位）和一个移位器（32 位），协助加速处理复杂操作。

(b) 总线控制单元(BCU)

BCU 控制内部总线。

(c) ROM

此为自地址 00000000H 所映射的 flash 存储器。

在指令获取期间，CPU 可在 1 个时钟周期内完成对 ROM/flash 存储器的访问。内部 ROM 容量和区域因下述产品的不同而异。

元件编号	内部 ROM 容量	内部 ROM 区域
μPD70F3713	64 KB (flash 存储器)	xn000000H 至 xn00FFFFH
μPD70F3714	128 KB (flash 存储器)	xn000000H 至 xn01FFFFH

备注 n = xx11B

(d) RAM

这是映射于地址 xnFFD800H 至 xnFFEFFFH 的 6KB 内部 RAM。

取指或数据存取期间，CPU 可在 1 个时钟周期内完成对数据的访问。

备注 n = xx11B

(e) 中断控制器(INTC)

该控制器用来处理自片上外围硬件和外部硬件的硬件中断请求（INTP0 到 INTP6）。可以对这些中断请求指定 8 个级别的中断优先级，同时也能执行多中断服务控制。

(f) 时钟发生器 (CG)

时钟发生器包括两个基本操作模式：PLL 模式（固定乘以 8）和时钟直通模式。它可以生成四种类型的时钟（f_{xx}，f_{xx}/2，f_{xx}/4，f_{xx}/8），并将其中一个应用为 CPU（f_{cpu}）的操作时钟。

(g) 定时器/计数器

该器件包括一个 16 位间隔定时器 M（TMM）通道，两个 16 位定时器/事件计数器 Q（TMQ）通道和四个 16 位定时器/事件计数器 P（TMP）通道，并能够测量脉冲间隔宽度和频率，允许用于马达控制的变频功能且可以输出可编程脉冲。

(h) 看门狗定时器 (WDT)

看门狗定时器用于检测程序循环，系统异常等。

它会在溢出发生后产生一个不可屏蔽的中断请求信号（INTWDT）或内部复位信号（WDTRES）。

(i) 串行接口

V850ES/IE2 包括两个异步串行接口 A (UARTA) 通道和一个作为串行接口的 3 线可变长度串行 I/O (CSIB) 通道。

使用 UARTA 时, 数据经由 TXDAn 引脚和 RXDAn 引脚 (n=0、1) 进行传输。

使用 CSIB 时, 数据经由 SOB0 引脚、SIB0 引脚, 和 $\overline{SCKB0}$ 引脚进行传输。

(j) A/D 转换器 (ADC)

V850ES/IE2 包含带有四个模拟输入引脚的双通道 10 位 A/D 转换器 (ADC0 和 ADC1)。

(k) 端口

如下所示, 以下端口兼有通用端口功能和控制引脚功能。

端口	I/O	复用功能
端口 0	7 位 I/O	定时器/计数器输入, 外部中断输入, A/D 转换器的外部触发器输入
端口 1	7 位 I/O	定时器/计数器 I/O
端口 2	8 位 I/O	定时器/计数器输出
端口 3	4 位 I/O	串行接口 I/O
端口 4	5 位 I/O	串行接口 I/O, 定时器/计数器 I/O
端口 DL	8 位 I/O	-

第二章 引脚功能

2.1 引脚功能列表

下表列出了 V850ES/IE2 单片机中引脚的名称和功能。这些引脚根据其功能可以分为端口引脚和非端口引脚。

为引脚的 I/O 缓冲器提供两个电源：A/D 转换器（AV_{DD0} 和 AV_{DD1}）的电源和外部引脚（EV_{DD}）的电源。电源和引脚关系如下所示：

表 2-1. 各引脚 I/O 缓冲器的电源

电源	对应引脚
AV _{DD0} 、AV _{DD1}	ANI00 至 ANI03、ANI10 至 ANI13
EV _{DD}	端口 0 至端口 4、端口 DL、RESET

(1) 端口引脚

(1/2)

引脚名称	引脚编号	I/O	功能	复用功能
P00	17	I/O	端口 0 7 位 I/O 端口 允许按位读取输入数据 / 写入输出数据。 能够按位指定片内上拉电阻的用途（片内上拉电阻只能在使用引脚的复用功能时在端口模式的输入模式中进行连接）。	INTP0/TOQH0OFF
P01	16			INTP1/TOQ1OFF
P02	15			INTP2/TOP2OFF
P03	14			INTP3/TOP3OFF
P04	13			INTP4/ADTRG0
P05	12			INTP5/ADTRG1
P06	11			INTP6
P10	24	I/O	端口 0 7 位 I/O 端口 允许按位读取输入数据/写入输出数据。 可以按位指定片内上拉电阻器的用途（在使用引脚复用功能的输入模式，且 TOP21 和 TOQH01 至 TOQH03 引脚在其复用功能被使用并作为输出引脚时，片内上拉电阻才仅能在端口模式的输入模式中进行连接，并进入高阻状态。）	TOQH01/TIQ01/TOQ01
P11	23			TIQ02/TOQ02
P12	22			TOQH02/TIQ03/TOQ03
P13	21			TIQ00
P14	20			TOQH03/EVTQ0
P16 (CLMER) [‡]	19			TOQ00 (CLMER) [‡] /TIP20
P17	18			TOP21/TIP21

注 仅在 P16 被指定为输出端口或 TOQ00 的输出功能时使能 CLMER 信号。当时钟监视器检测到错误（振荡器停止）时，将会强行输出一个低电平。由复位信号解除低电平输出。详情参见表 4-5 端口 1 的复用功能引脚。

引脚名称	引脚编号	I/O	功能	复用功能
P20	54	I/O	端口 2 8 位 I/O 端口 允许按位读取输入数据 / 写入输出数据。 可以按位指定片内上拉电阻器的用途（片内上拉电阻只能在端口模式的输入模式中进行连接，或者当 TOQ1T1 至 TOQ1T3 和 TOQ1B1 至 TOQ1B3 引脚以及 TOP31 引脚在其复用功能被使用且作为输出引脚时，才进入高阻状态。）	TOQ1T1
P21	53			TOQ1B1
P22	52			TOQ1T2
P23	51			TOQ1B2
P24	50			TOQ1T3
P25	49			TOQ1B3
P26	46			TOQ10
P27	45			TOP31
P30	44	I/O	端口 3 4 位 I/O 端口 允许按位读取输入数据/写入输出数据。 可以按位指定片内上拉电阻器的用途（片内上拉电阻器只能在使用引脚复用功能的输入模式时在端口模式的输入模式中进行连接。）	RXDA0
P31	43			TXDA0
P32	42			RXDA1
P33	41			TXDA1
P40	40	I/O	端口 4 5 位 I/O 端口 允许按位读取输入数据/写入输出数据。 可以按位指定一个片内上拉电阻器（片内上拉电阻器只能在输入模式时使用引脚复用功能，且在端口模式的输入模式中进行连接，包括从机模式中 SCKB0 引脚的情形。）	SIB0
P41	39			SOB0
P42	38			SCKB0
P43	37			TOP00/TIP00
P44	36			TOP01/TIP01
PDL0	35	I/O	端口 DL 8 位 I/O 端口 允许按位读取输入数据/写入输出数据。 可以按位指定一个片上上拉电阻器（片上上拉电阻器可以在引脚处于端口模式和输入模式时进行连接）	-
PDL1	34			-
PDL2	33			-
PDL3	32			-
PDL4	31			-
PDL5	30			FLMD1
PDL6	29			-
PDL7	28			-

<R>

(2) 非端口引脚

(1/2)

引脚名称	引脚编号	I/O	功能	复用功能
ADTRG0	13	输入	A/D 转换器 0, 1 的外部触发器输入	INTP4/P04
ADTRG1	12	输入		INTP5/P05
ANI00	1	输入	模拟输入至 A/D 转换器 0, 1	-
ANI01	2	输入		-
ANI02	3	输入		-
ANI03	4	输入		-
ANI10	58	输入		-
ANI11	57	输入		-
ANI12	56	输入		-
ANI13	55	输入		-
AV _{DD0}	63	-		A/D 转换器 0, 1 的正级电源 (与 V _{DD} 电位相同)
AV _{DD1}	60	-	-	
AV _{REF0}	64	-	A/D 转换器 0, 1 的参考电压输入 (与 AV _{DD0} 和 AV _{DD1} 电位相同)	-
AV _{REF1}	59	-		-
AV _{SS0}	62	-	A/D 转换器 0, 1 的地电位 (与 V _{SS} 电位相同)	-
AV _{SS1}	61	-		-
EV _{DD}	26, 47	-	外部引脚的正级电源	-
EV _{SS}	27, 48	-	外部引脚的地电位	-
EVTQ0	20	输入	TMQ0 的外部事件计数输入	TOQH03/P14
FLMD0	25	输入	设置 flash 存储器编程模式的引脚	-
FLMD1	30	输入		PDL5
INTP0	17	输入	外部可屏蔽中断请求输入	TOQH0OFF/P00
INTP1	16			TOQ1OFF/P01
INTP2	15			TOP2OFF/P02
INTP3	14			TOP3OFF/P03
INTP4	13			ADTRG0/P04
INTP5	12			ADTRG1/P05
INTP6	11			P06
REGC	10	-	稳压器输出稳定电容连接	-
RESET	5	输入	系统复位输入	-
RXDA0	44	输入	UARTA0, UARTA1 的串行接收数据输入	P30
RXDA1	42			P32
SCKB0	38	I/O	CSIB0 的串行时钟 I/O	P42
SIB0	40	输入	CSIB0 的串行接收数据输入	P40
SOB0	39	输出	CSIB0 的串行发送数据输出	P41

引脚名称	引脚编号	I/O	功能	复用功能
TIP00	37	输入	TMP0 的外部事件计数输入/外部触发器输入/捕获触发器输入	TOP00/P43
TIP01	36		TMP0 捕获触发器输入	TOP01/P44
TIP20	19		TMP2 的外部事件计数输入/外部触发器输入/捕获触发器输入	TOQ00 (CLMER) [‡] / P16 (CLMER) [‡]
TIP21	18		TMP2 捕获触发器输入	TOP21/P17
TIQ00	21	输入	TMQ0 捕获触发器输入	P13
TIQ01	24			TOQH01/TOQ01/P10
TIQ02	23			TOQ02/P11
TIQ03	22			TOQH02/TOQ03/P12
TOP00	37	输出	TMP0、TMP2 的脉冲信号输出	TIP00/P43
TOP01	36			TIP01/P44
TOP21	18			TIP21/P17
TOP2OFF	15	输入	高阻抗输出控制的信号输入。	INTP2/P02
TOP31	45	输出	TMP3 的脉冲信号输出	P27
TOP3OFF	14	输入	高阻抗输出控制信号输入。	INTP3/P03
TOQ00 (CLMER) [‡]	19	输出	TMQ0 的脉冲信号输出	TIP20/P16 (CLMER) [‡]
TOQ01	24			TOQH01/TIQ01/P10
TOQ02	23			TIQ02/P11
TOQ03	22			TOQH02/TIQ03/P12
TOQ10	46	输出	TMQ1 的脉冲信号输出	P26
TOQ1B1	53	输出	6 相位 PWM 的脉冲信号输出	P21
TOQ1B2	51			P23
TOQ1B3	49			P25
TOQ1OFF	16	输入	高阻抗输出控制信号输入。	INTP1/P01
TOQ1T1	54	输出	6 相位 PWM 的脉冲信号输出	P20
TOQ1T2	52			P22
TOQ1T3	50			P24
TOQH01	24	输出	由 TMQ0 脉冲信号输出的高阻输出和 TOQH0OFF 引脚输入的有效沿	TIQ01/TOQ01/P10
TOQH02	22			TIQ03/TOQ03/P12
TOQH03	20			EVTQ0/P14
TOQH0OFF	17	输入	高阻抗输出控制信号输入。	INTP0/P00
TXDA0	43	输出	UARTA0, UARTA1 的串行发送数据输出	P31
TXDA1	41			P33
V _{DD}	9	-	内部器件的正极电源	-
V _{SS}	8	-	内部器件的地电位	-
X1	6	输入	系统时钟的谐振器连接引脚	-
X2	7	-		-

注 仅在 P16 被指定为输出端口或 TOQ00 的输出功能时使能 CLMER 信号。当时钟监视器检测到错误（振荡器停止）时，将会强行输出一个低电平。由复位信号解除低电平输出。详情参见表 4-5 端口 1 的复用功能引脚。

2.2 引脚I/O电路以及空闲引脚连接建议

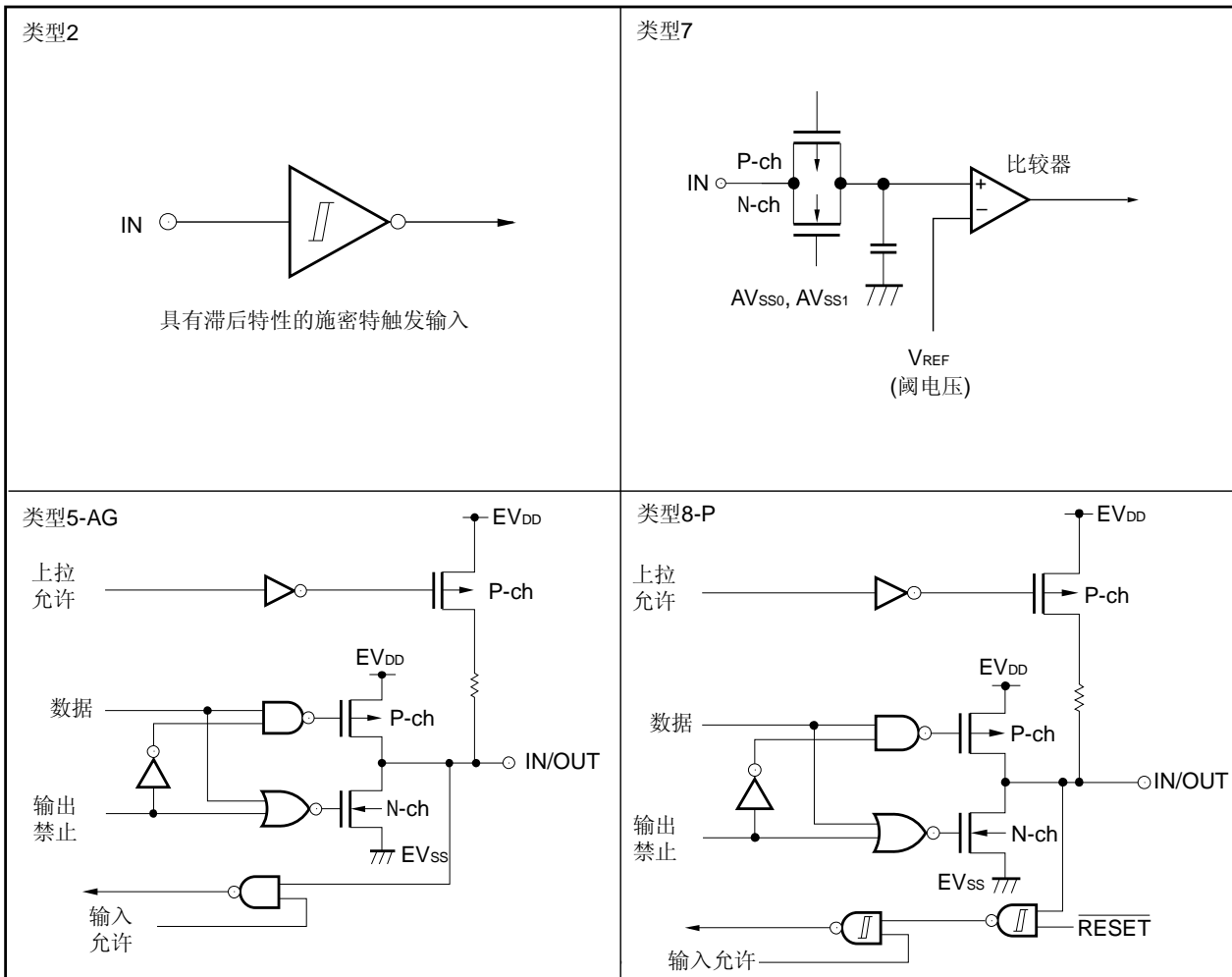
(1/2)

引脚名称	复用功能引脚名称	引脚编号	I/O 电路类型	推荐的连接方式	
P00	INTP0/TOQH0OFF	17	8-P	输入： 通过电阻单独连接至 EV _{DD} 或 EV _{SS} 输出： 保持开路。	
P01	INTP1/TOQ1OFF	16			
P02	INTP2/TOP2OFF	15			
P03	INTP3/TOP3OFF	14			
P04	INTP4/ADTRG0	13			
P05	INTP5/ADTRG1	12			
P06	INTP6	11			
P10	TOQH01/TIQ01/TOQ01	24			
P11	TIQ02/TOQ02	23			
P12	TOQH02/TIQ03/TOQ03	22			
P13	TIQ00	21			
P14	TOQH03/EVTQ0	20			
P16 (CLMER)*	TOQ00 (CLMER)*/TIP20	19			
P17	TOP21/TIP21	18			
P20	TOQ1T1	54			5-AG
P21	TOQ1B1	53			
P22	TOQ1T2	52			
P23	TOQ1B2	51			
P24	TOQ1T3	50			
P25	TOQ1B3	49			
P26	TOQ10	46			
P27	TOP31	45			
P30	RXDA0	44	8-P		
P31	TXDA0	43	5-AG		
P32	RXDA1	42	8-P		
P33	TXDA1	41	5-AG		
P40	SIB0	40	8-P		
P41	SOB0	39	5-AG		
P42	$\overline{\text{SCKB0}}$	38	8-P		
P43	TOP00/TIP00	37			
P44	TOP01/TIP01	36			

注 仅在 P16 被指定为输出端口或 TOQ00 的输出功能时使能 CLMER 信号。当时钟监视器检测到错误（振荡器停止）时，将会强行输出一个低电平。由复位信号解除低电平输出。详情参见表 4-5 端口 1 的复用功能引脚。

引脚名称	复用功能引脚名称	引脚编号	I/O 电路类型	推荐的连接方式
PDL0	-	35	5-AG	输入：通过电阻单独连接至 EV _{DD} 或 EV _{SS} 输出：保持开路。
PDL1	-	34		
PDL2	-	33		
PDL3	-	32		
PDL4	-	31		
PDL5	FLMD1	30		
PDL6	-	29		
PDL7	-	28		
ANI00	-	1	7	通过电阻单独连接至 AV _{DD0} 、AV _{DD1} 、AV _{SS0} 、或 AV _{SS1}
ANI01	-	2		
ANI02	-	3		
ANI03	-	4		
ANI10	-	58		
ANI11	-	57		
ANI12	-	56		
ANI13	-	55		
RESET	-	5	2	-
FLMD0	-	25		-

2.3 引脚I/O电路



第三章 CPU功能

V850ES/IE2 单片机的 CPU 是基于 RISC 架构设计的，利用 5 级流水线控制，几乎所有指令均可在一个时钟周期内完成。

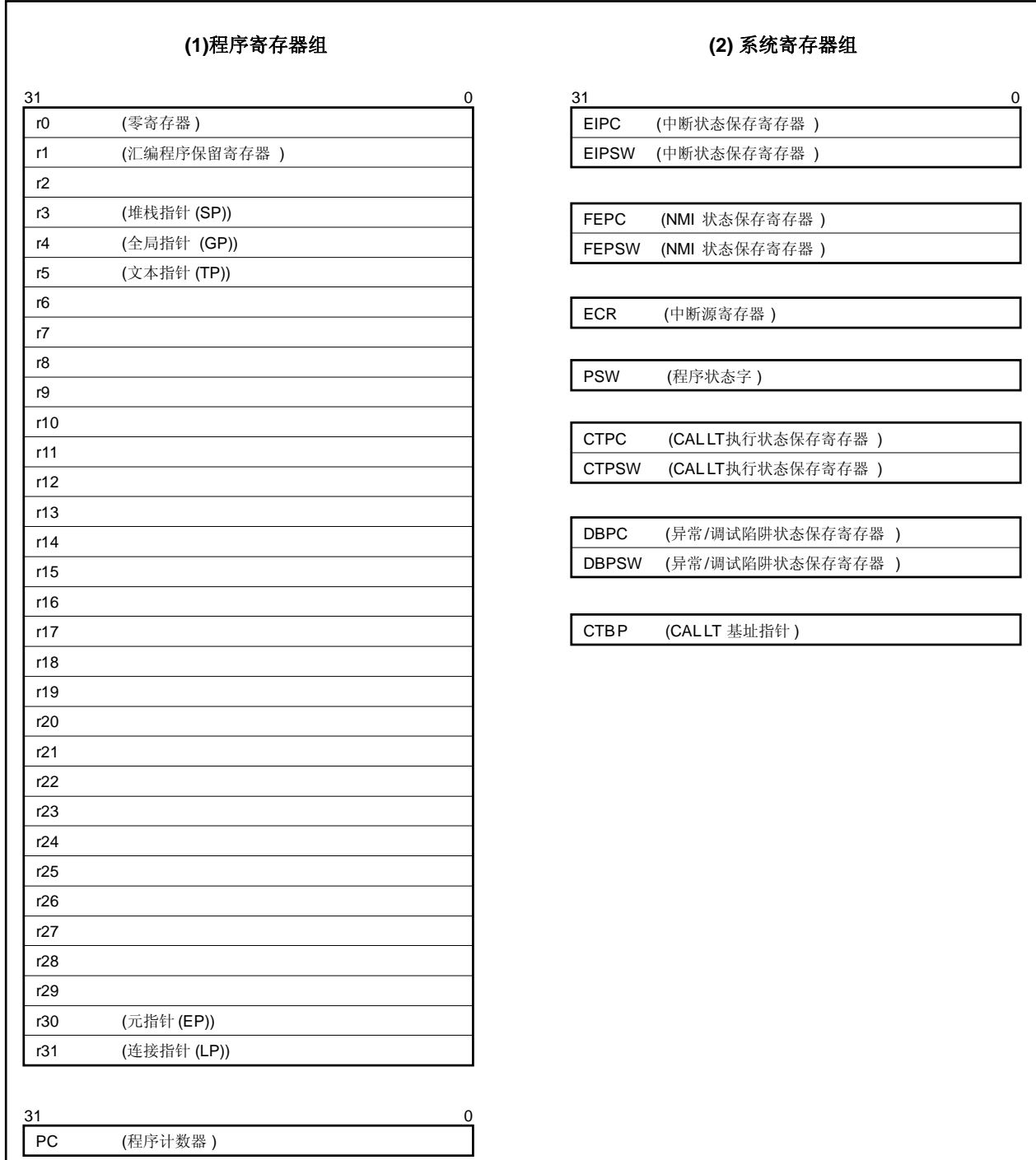
3.1 特性

- 最短指令执行时间： 50 ns
(@ 20 MHz 的时钟频率下： 4.5 至 5.5 V (使用 A/D 转换器时)，
3.5 至 5.5 V (不使用 A/D 转换器时))。
- 内存空间： 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 寄存器
- 内部 32 位结构。
- 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/IE2 的 CPU 寄存器可分为两种类型：通用程序寄存器组和专用系统寄存器组。所有寄存器都具有 32 位宽度。

详情参见 **V850ES 架构用户手册**。



3.2.1 程序寄存器组

程序寄存器组包括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) 可供选择。所有这些寄存器都可用作数据变量或地址变量。

执行某些指令会隐式使用 r0 和 r30 寄存器, 使用这些寄存器时一定要加以注意。寄存器 r0 总是保持为 0, 用于使用数据 0 的操作和 0 偏移量寻址。当使用 SLD 和 SST 指令访问存储器时, 寄存器 r30 用作基址指针。

汇编编译器和 C 语言编译器也隐式使用 r1、r3 至 r5、和 r31。因此, 在使用这些寄存器之前, 必须保存寄存器中的内容以防其内容丢失, 并且使用寄存器后, 这些内容必须重新恢复到寄存器中。通过实时操作系统使用 r2 时会出现一些情况。如果 r2 没有被实时操作系统使用, 那么它将可以用作变量寄存器。

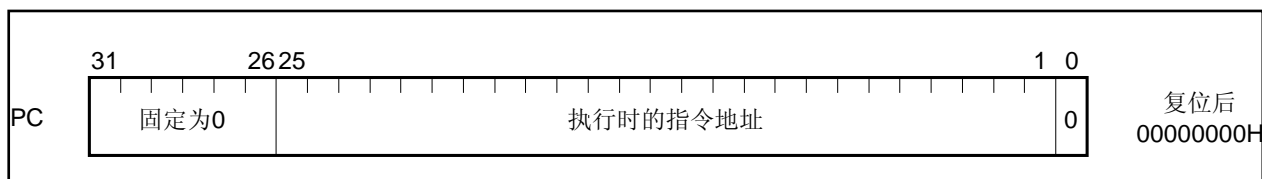
表 3-1. 通用寄存器

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	生成 32 位立即数的工作寄存器
r2	用作地址变量/数据变量寄存器(如果所用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	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	地址/数据变量寄存器	
r30	元指针	用作访问内存的基址指针
r31	链接指针	调用函数时由编译器使用

(2) 程序计数器 (PC)

程序计数器用来存放指令执行期间的指令地址。该寄存器低 26 位有效, 其中位 31 至位 26 固定为 0, 即使在位 25 向位 26 有进位也将忽略不计。

位 0 固定为 0, 且不能执行至奇地址处的转移。



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）设置以下所显示的系统寄存器编号，以实现系统寄存器的读取和写入。

表 3-2. 系统寄存器编号

系统寄存器编号	系统寄存器名称	是否允许操作数制定	
		LDSR 指令	STSR 指令
0	中断状态保存寄存器 (EIPC) ^{註1}	是	是
1	中断状态保存寄存器 (EIPSW) ^{註1}	是	是
2	NMI 状态保存寄存器 (FEPC)	是	是
3	NMI 状态保存寄存器 (FEPSW)	是	是
4	中断源寄存器(ECR)	否	是
5	程序状态字(PSW)	是	是
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)	否	否
16	CALLT 指令执行状态保存寄存器(CTPC)	是	是
17	CALLT 指令执行状态保存寄存器(CTPSW)	是	是
18	异常/调试陷阱状态保存寄存器(DBPC)	是 ^{註2}	是 ^{註2}
19	异常/调试陷阱状态保存寄存器 (DBPSW)	是 ^{註2}	是 ^{註2}
20	CALLT 指令基址指针(CTBP)	是	是
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)	否	否

- 注
1. 由于只有一组这样中断状态保存寄存器可用，所以，如果允许多重中断，这些寄存器的内容必须在程序中通过编程加以保存。
 2. 只能在 DBTRAP 指令或非法操作码执行后，DBRET 指令执行之前进行访问。

注意事项 即使 EIPC、FEPC、或 CTPC 的位 0 由 LDSR 指令设置为 1，中断服务之后 RETI 指令返回期间仍然会忽略位 0 (因为 PC 的位 0 固定为 0)。因此当设置 EIPC，FEPC 或 CTPC 时，设置数值应为偶数 (位 0 = 0)

(1) 中断状态保存寄存器(EIPC 和 EIPSW)

有两个中断状态保存寄存器： EIPC 和 EIPSW。

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

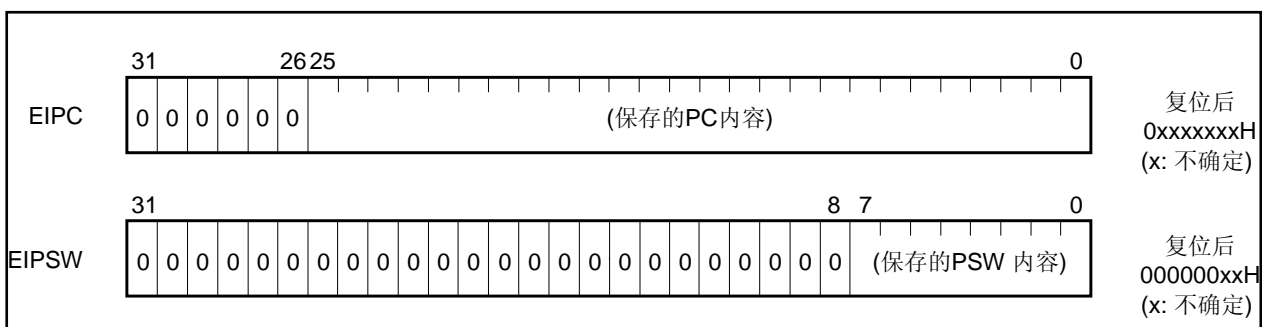
当有软件异常或者是可屏蔽中断产生时，就将正在执行指令的下一条指令地址保存到寄存器 EIPC 中。有些指令除外(可参见 **14.9 CPU 不响应中断期间**)。

将 PSW 当前内容保存到 EIPSW 寄存器中。

由于只有一组这样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在程序中通过编程加以保存。

EIPC 的位 31 至位 26 以及 EIPSW 的位 31 至位 8 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，EIPC 和 EIPSW 中的值将会分别恢复到 PC 和 PSW 中。



(2) NMI 状态保存寄存器(FEPC、FEPSW)

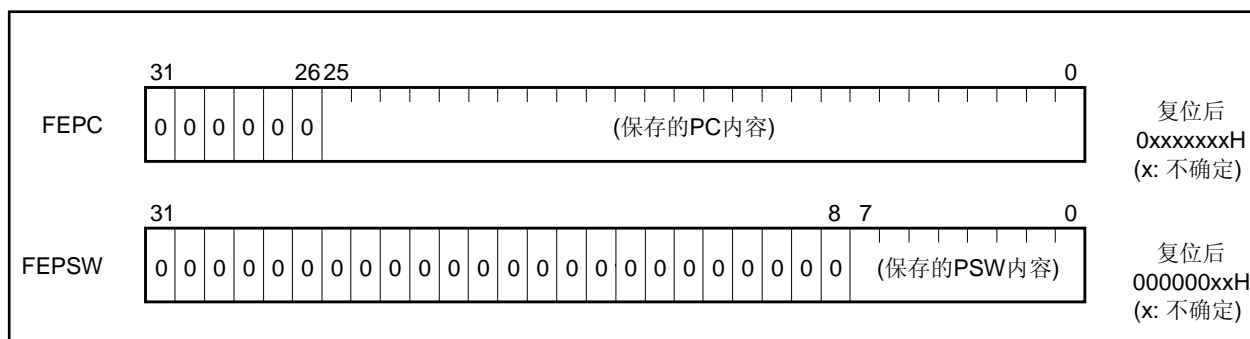
有两个 NMI 状态保存寄存器： FEPC 和 FEPSW。

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

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

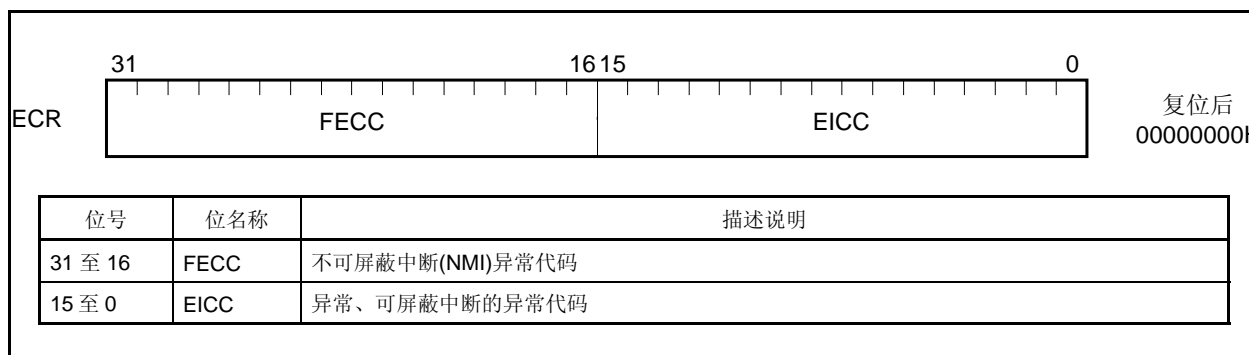
FEPC 的位 31 至位 26 以及 FEPSW 的位 31 至位 8 作为保留位用于将来功能扩展(这些位始终固定为 0)。

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，中断或异常源信息就保存在该中断源寄存器(ECR)中。该寄存器存放的是每一个中断源的异常代码。由于该寄存器是只读寄存器，所以，不能通过 LDSR 指令将数据写入其中。



(4) 程序状态字 (PSW)

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

如果利用 LDSR 指令更改了该寄存器的内容, 那么, 在 LDSR 指令完成后, 其新内容就会立即生效。 在通过 LDSR 指令对程序状态字(PSW)执行写操作期间, 中断请求响应将保持挂起状态。

位 31 至位 8 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

复位后
00000020H

PSW

RFU

NP

EP

ID

SAT

CY

OV

S

Z

位号	标志名称	描述说明
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。 异常产生后该标志置为 1。 即使该位置 1, 也可以响应中断请求。 0: 没有处理异常。 1: 正在处理异常。
5	ID	标识是否允许响应可屏蔽中断请求。 0: 允许中断 (EI)。 1: 禁止中断 (DI)。
4	SAT ^注	表示执行饱和和运算指令的结果溢出且饱和。 由于该位为一累积标志位, 所以, 当饱和和运算指令的结果饱和时, 该位便置 1, 即使下一条指令的运算结果没有饱和, 该位也不会清 0。 执行算术运算指令时, 该位既不置 1, 也不清 0。 0: 不饱和。 1: 饱和
3	CY	表示运算结果有无进位或借位产生 0: 无进位或借位产生 1: 发生进位或借位
2	OV ^注	表示在运算过程中有无溢出。 0: 没有发生溢出。 1: 发生溢出。
1	S ^注	表示运算结果是否为负数。 0: 运算结果为正数或零。 1: 运算结果为负值。
0	Z	显示运算结果是否为零。 0: 运算结果不为 0。 1: 运算结果为 0。

备注 “注”在下页进行说明。

注 在饱和运算过程中，饱和运算结果通过 OV 标志和 S 标志的值来确定。在饱和运算执行时，只有当 OV 标志位置 1 时，SAT 标志位才置为 1。

运算结果状态	标志位状态			饱和运算结果
	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）	运算前保持原值	0	0	实际运算结果
负值（未越下界）			1	

(5) CALLT 指令执行状态保存寄存器 (CTPC、CTPSW)

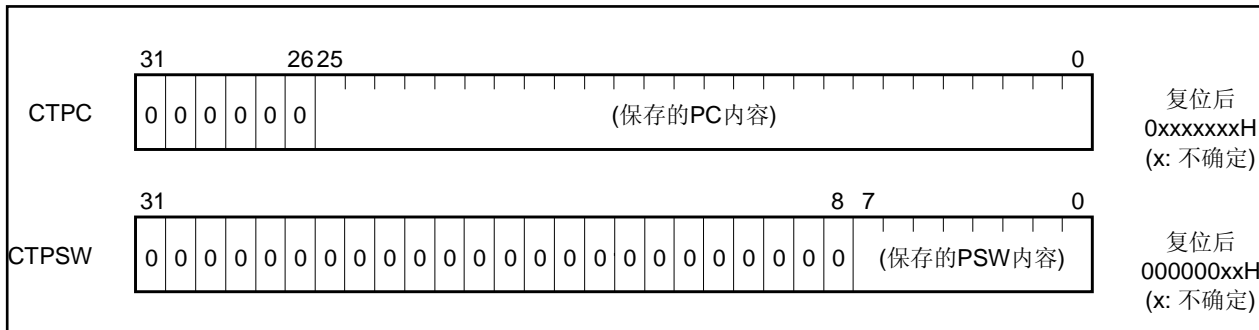
有两个 CALLT 指令执行状态保存寄存器：CTPC 和 CTPSW。

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

寄存器 CTPC 中保存的内容是 CALLT 指令下一条指令的地址。

PSW 寄存器的当前内容保存到 CTPSW 寄存器中。

CTPSW 的位 31 至位 26 和 CTPSW 的位 31 至位 8 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、DBPSW)

有两个异常/调试陷阱状态保存寄存器：DBPC 和 DBPSW。

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

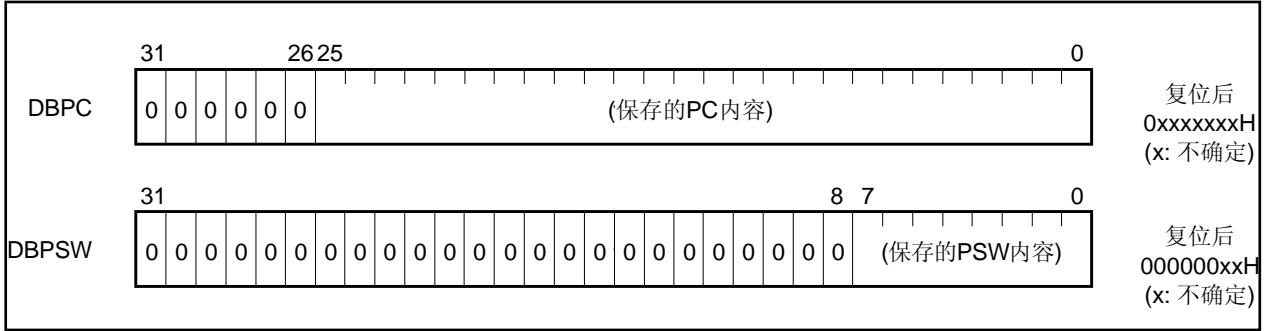
在异常陷阱或者调试陷阱产生时，保存到 DBPC 寄存器中的内容是正在执行指令的下一条指令的地址。

PSW 寄存器的当前内容保存到 DBPSW 寄存器中。

<R> 这些寄存器只能在执行 DBTRAP 指令/非法操作码和执行 DBRET 指令之间的间隔时间内进行读取。

DBPC 的位 31 至位 26 以及 DBPSW 的位 31 至位 8 作为保留位用于将来功能扩展(这些位始终固定为 0)。

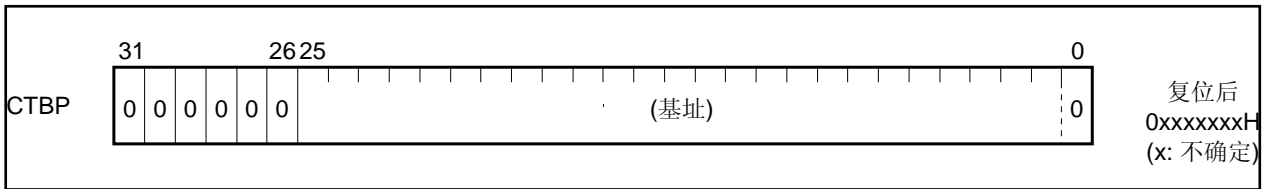
执行 DBRET 指令后，将寄存器 DBPC 的值恢复到 PC 中，寄存器 DBPSW 的值恢复到 PSW 中。



(7) CALLT 基址指针(CTBP)

CALLT 基址指针(CTBP)用于指定表地址并且生成一个目标地址(位 0 固定为 0)。

位 31 至位 26 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/IE2 具有以下几种操作模式。

(1) 正常操作模式

系统从复位状态中恢复后，与总线接口关联的各个引脚均设置为端口模式。程序执行转移到内部 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

设置为该模式后，程序可以通过 flash 编程器写入内部 flash 存储器。

(a) 操作模式说明

操作模式根据 FLMD0 和 FLMD1 引脚的状态（输入电平）来确定。

在正常操作模式下，复位后，在 FLMD0 引脚有低电平信号输入。

连接 flash 编程器后，在 flash 存储器编程模式中，通过 flash 编程器会有高电平输入 FLMD0 引脚；然而，在自编程模式中，则会由内部电路输入高电平。

在应用系统中，牢记这些引脚的技术要求，在运行期间，不要改变这些引脚的设置。

FLMD0	FLMD1	操作模式
L	×	正常操作模式
H	L	Flash 存储器编程模式
H	H	设置禁止

H: 高电平

L: 低电平

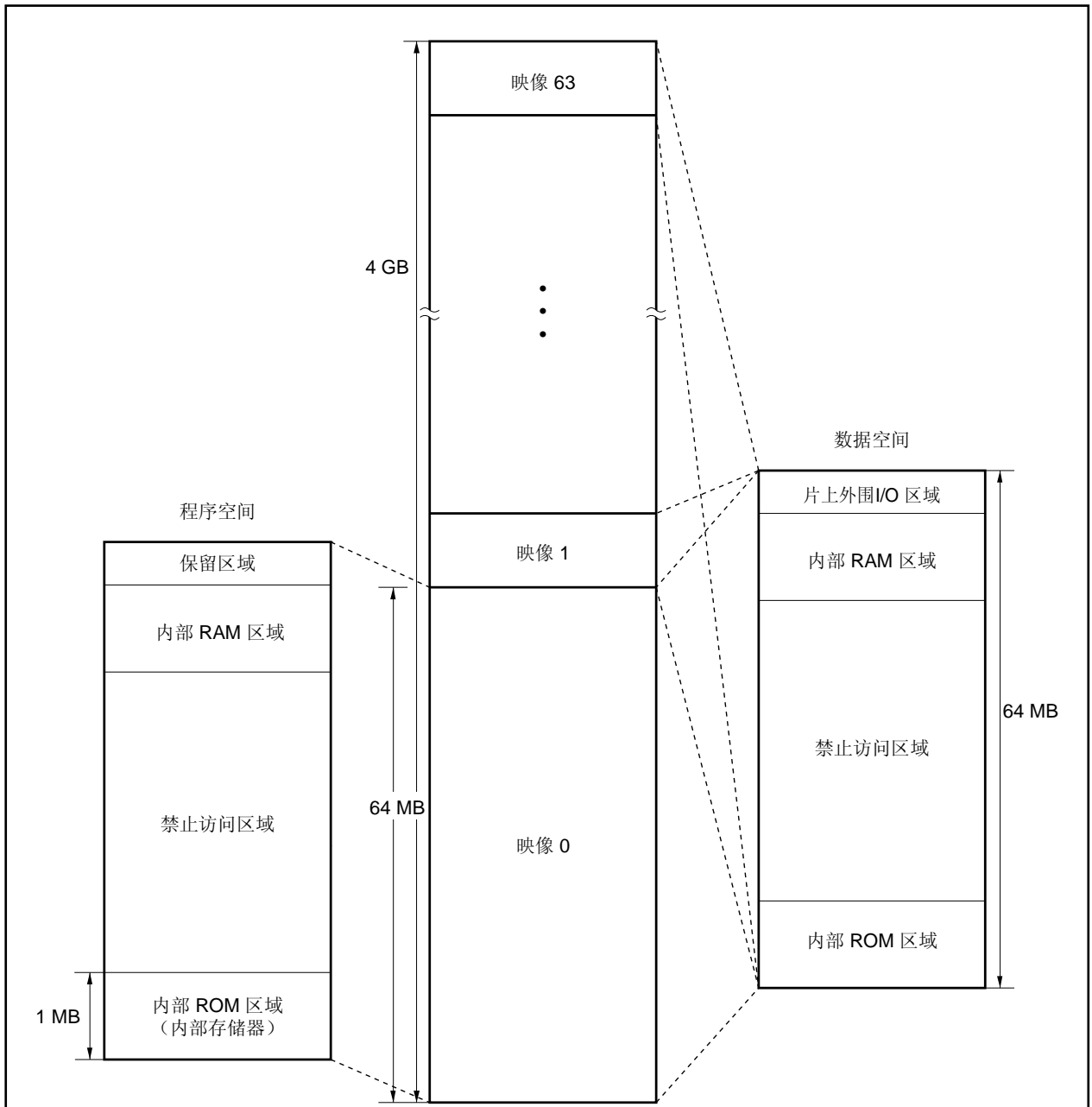
×: 不必理会

3.4 地址空间

3.4.1 CPU地址空间

对于指令地址来说，在最大为 64MB 的线性地址空间（程序空间）中支持一个最大为 1MB 的内部 ROM 区域和一个内部 RAM 区域。对于操作数寻址（数据访问）来说，则支持一个最大为 4GB 的线性地址空间（数据空间）。然而，这个 4GB 的地址空间被视为 64MB 物理地址空间的 64 个映射。这意味着无论位 31 至位 26 的取值如何，系统总是对同样 64MB 物理地址空间进行访问。

图 3-1. 地址空间映射



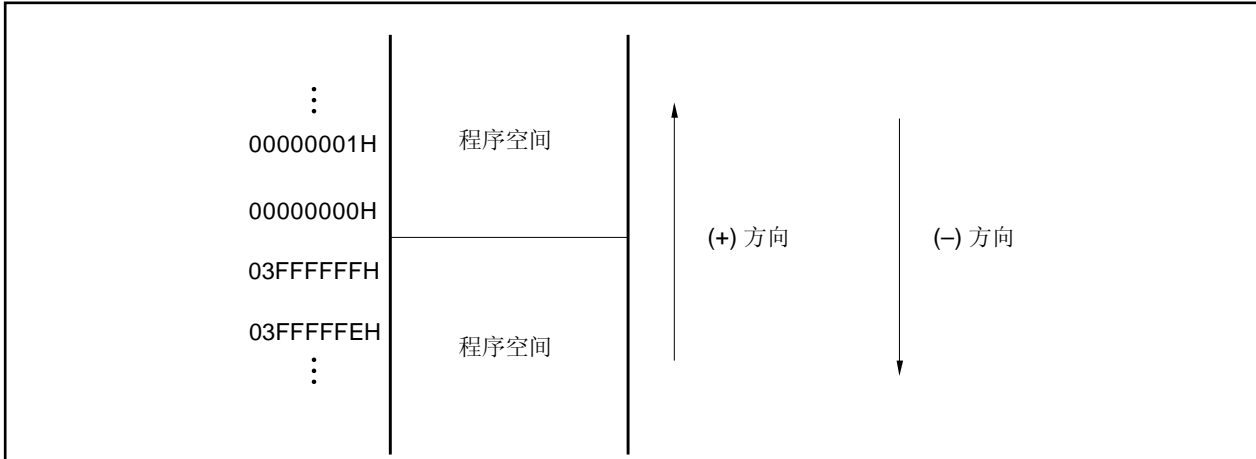
3.4.2 CPU 地址空间的绕回

(1) 程序空间

对于 32 位的 PC（程序计数器），其高 6 位固定为 0，且仅有低 26 位是有效位。即使转移地址计算的结果是位 25 至位 26 发生了进位或借位，高 6 位仍会忽略并保持为 0。

因此，程序空间的上限地址 03FFFFFFH 和下限地址 00000000H 是连续相邻地址，程序空间则环绕于这两个边界地址内。

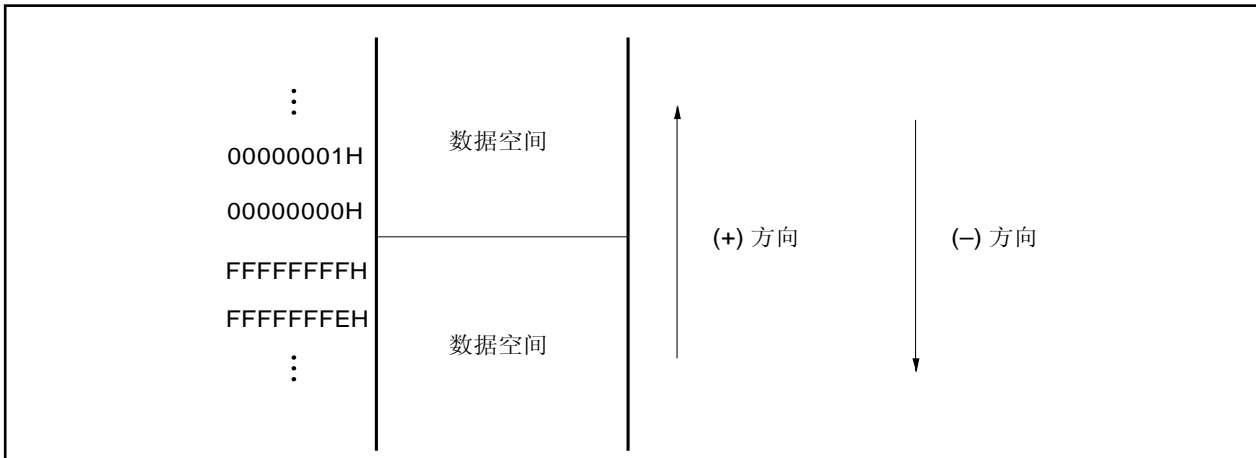
注意事项 由于地址 03FFF000H 至 03FFFFFFH 的 4 KB 地址区域为片上外围 I/O 区域，所以不能从该地址区域取指令。因此，不要执行任何分支操作指令，其目标地址将会留在该区域的任何位置。



(2) 数据空间

操作数地址计算操作结果超出 32 位时，该结果地址将被忽略。

因此，数据空间的上限地址 FFFFFFFFH 和下限地址 00000000H 是连续相邻地址，数据空间则环绕于这两个边界地址内。



3.4.3 内存映射

V850ES/IE2 保留了如下所示的存储区域。

图 3-2. 数据内存映射(物理地址)

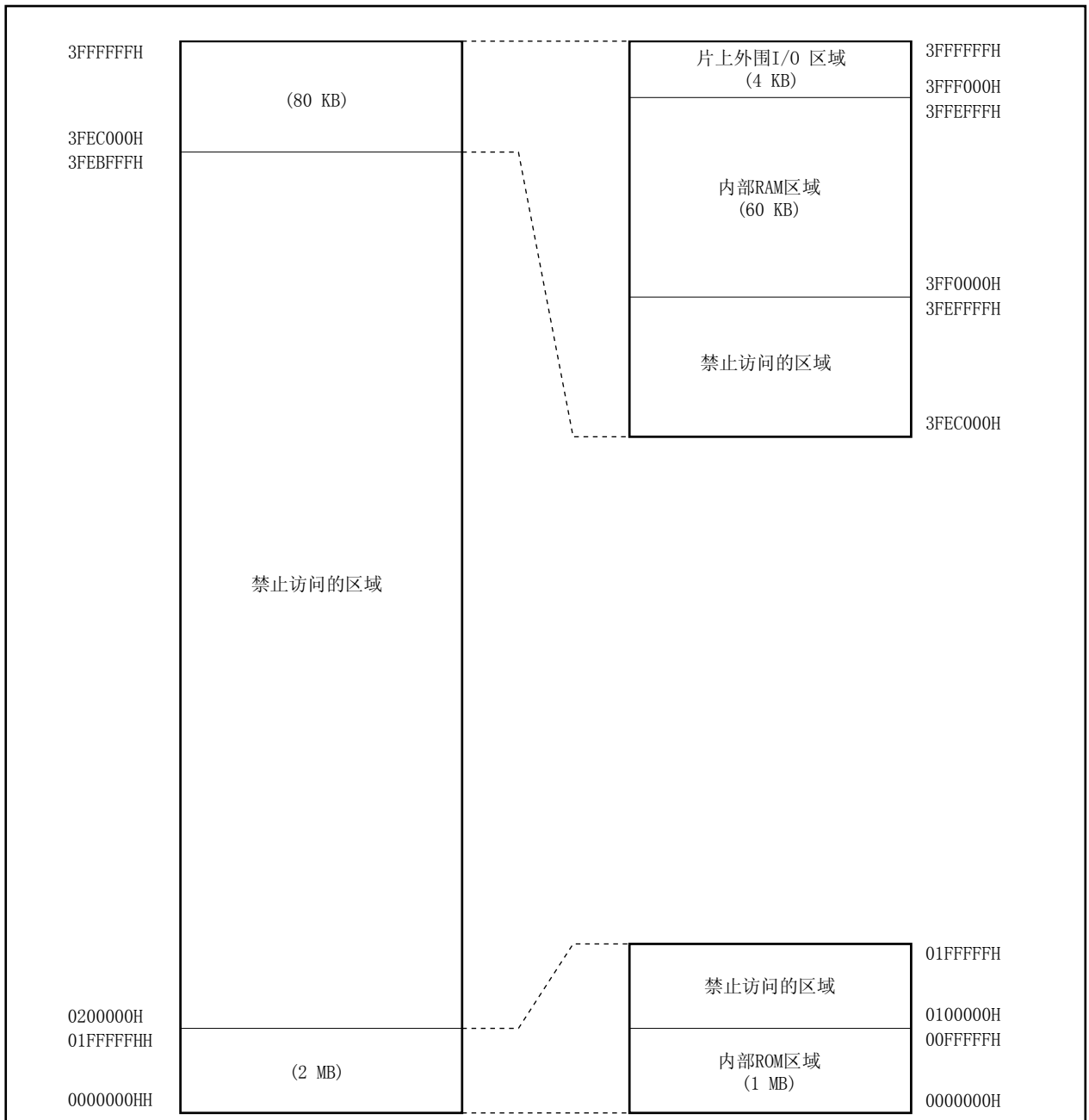
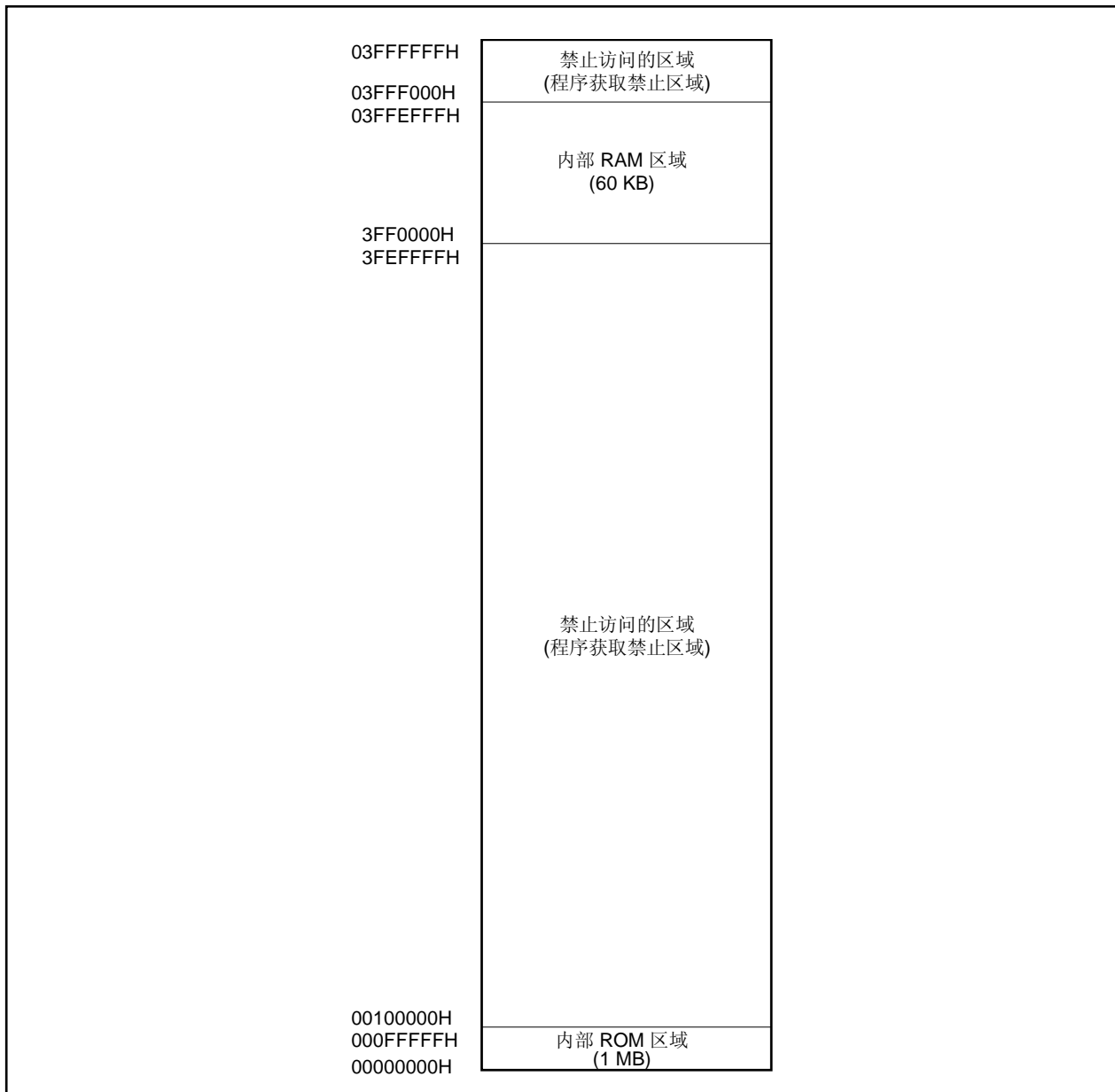


图 3-3. 程序内存映射



3.4.4 存储区域

(1) 内部 ROM 区域

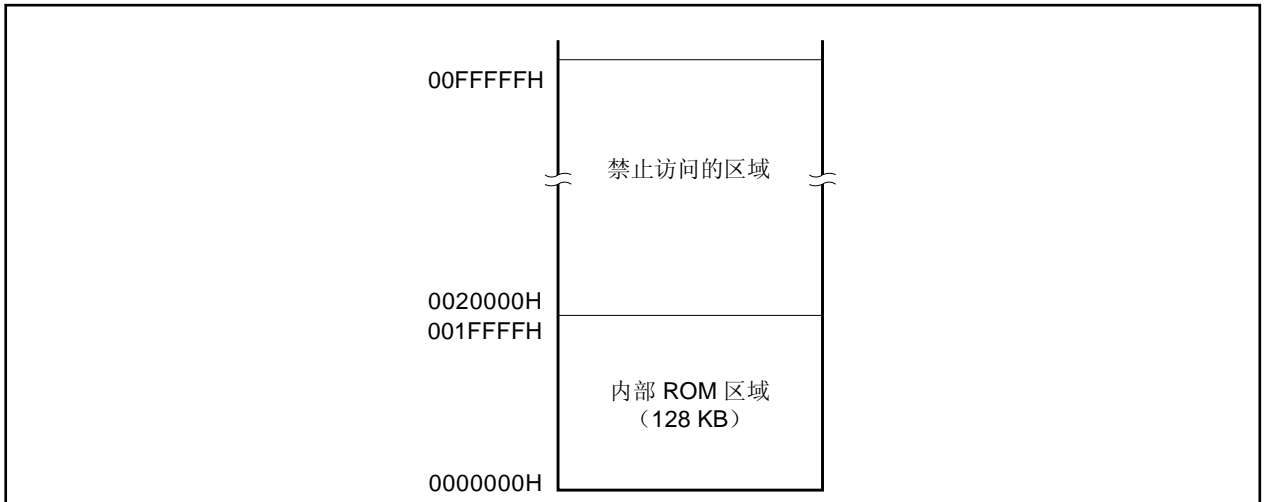
地址 0000000H 至 00FFFFFFH 的 1MB 存储区域保留作为内部 ROM 区域。

(a) 内部 ROM (128 KB)

在 μ PD70F3714 中提供了 0000000H 至 001FFFFH 的 128KB 存储区域。

地址 0020000H 至 00FFFFFFH 是一块禁止访问的区域。

图 3-4. 内部 ROM 区域 (128 KB)

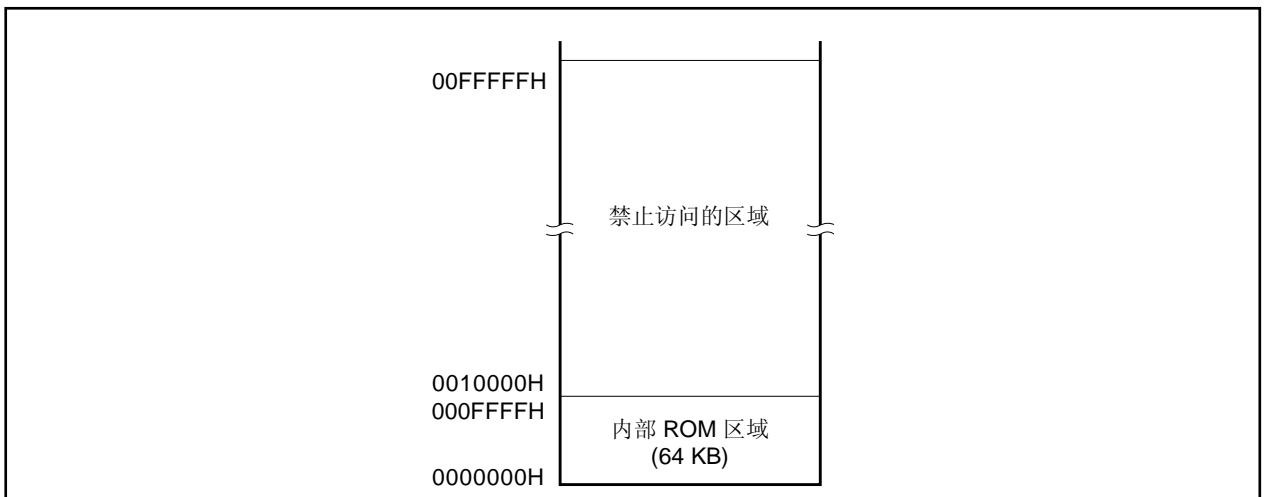


(b) 内部 ROM (64KB)

在 μ PD70F3713 中提供了 0000000H 至 000FFFFH 的 64KB 存储区域。

地址 0010000 至 00FFFFFFH 是一块禁止访问的区域。

图 3-5. 内部 ROM 区域 (64 KB)



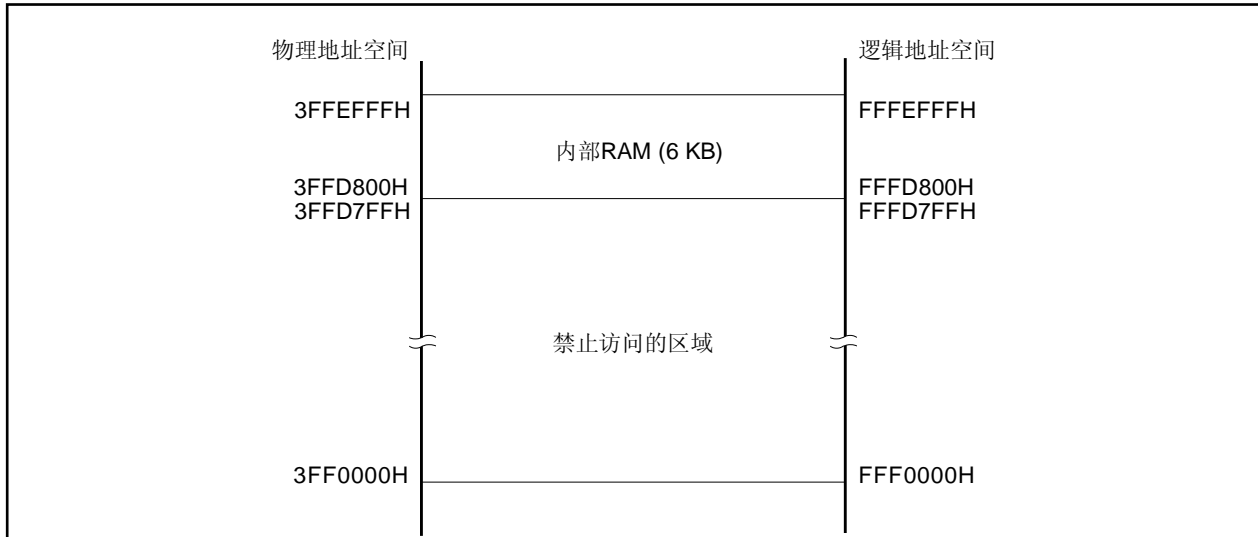
(2) 内部 RAM 区域

地址 3FF0000H 至 3FFFFFFFFH 的 60 KB 存储区域保留作为内部 RAM 区域。

地址 3FFD800H 至 3FFFFFFFFH 的 6KB 存储区域作为 V850ES/IE2 的物理内部 RAM 区域。

地址 3FF0000H 至 3FFD7FFFH 是一块禁止访问的区域。

图 3-6. 内部 RAM 区域 (6KB)



(c) 内部存储器容量设置寄存器 (IMS)

IMS 寄存器用于设置 V850ES/IE2 内部 RAM 的容量。

该寄存器只能按字节进行写入操作。

复位输入将该寄存器设为 00H。

- 注意事项**
1. 访问内部 RAM 前对 IMS 寄存器进行写操作。复位解除后, 仅能对该寄存器进行一次写入操作。
 2. 确保将 01H 写入 IMS 寄存器。
 3. CA850 所提供的示例启动程序包含了将内部 RAM 区域清零的一段代码。因此, 在执行清零程序前需要对 IMS 寄存器进行设置。
 在使用示例启动程序时, 应在启动程序中 __START 标签后立即添加以下[说明示例]中所显示的指令<2> 至<5>。
 指令<2>中的“0x11”是 VSWC 寄存器的设置值, 而指令<4>中的“0x01”则是 IMS 寄存器的设置值。

[说明示例]

```

<1>__START:
<2>mov  0x11,  r13
<3>st.b r13,  VSWC
<4>mov  0x01,  r12
<5>st.b r12,  IMS
<6>mov  #_tp_TEXT, tp
      :
      :
    
```

} 添加

备注 使用合作伙伴工具时, 根据注意事项 1 至 3 中的内容进行设置。
 此外, 应按如下进行描述以定义 IMS 寄存器。

```
#define IMS (*(volatile unsigned char *)0xffff9f0)
```

复位后: 00H W 地址: FFFFF9F0H

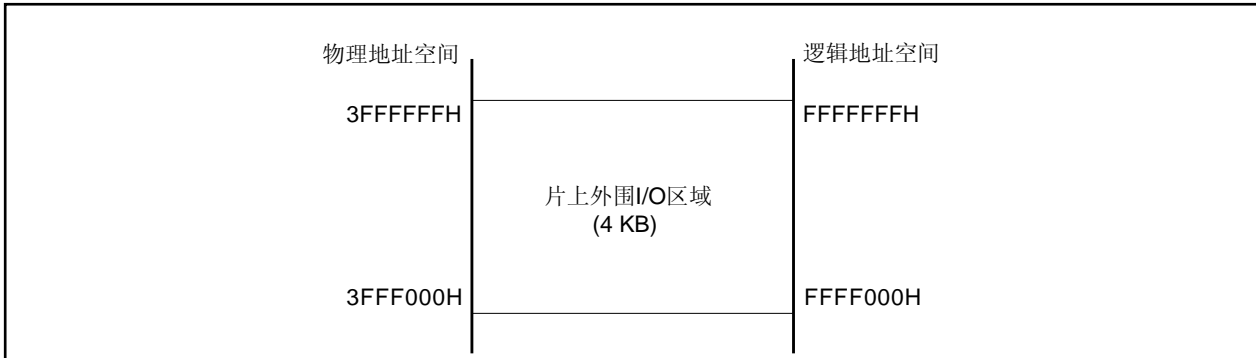
	7	6	5	4	3	2	1	0
IMS	0	0	0	0	0	0	0	RAM0

RAM0	内部RAM容量说明
0	不确定
1	6 KB (3FFD800H to 3FFEFFFH (RAM size of V850ES/IE2))

(3) 片上外围 I/O 区域

从 03FFF000H 至 03FFFFFFFH 的 4KB 地址空间保留用作片上外围 I/O 区域。

图 3-7. 片上外围 I/O 区域



已由诸如片上外围 I/O 操作模式说明以及状态监测之类功能赋值的片上外围 I/O 寄存器映射于片上外围 I/O 区域。注意，该区域不允许程序获取。

- 注意事项**
1. 如果尝试对寄存器进行字访问，则将会对该字区域执行两次半字的访问，先低位后高位，其中忽略低 2 位地址。
 2. 以字节为单位进行访问的寄存器，以半字为单位进行访问时，如果访问操作是读取操作，那么该寄存器的高 8 位将不明确。同样，写数据时，只将低 8 位数据写入寄存器。
 3. 没有定义的地址区域保留为将来扩展之用。如果对这些地址进行访问，其操作不确定且不保证操作的有效性。

3.4.5 地址空间使用建议

V850ES/IE2 的架构要求：当访问数据空间的操作数时，必须确保有一个寄存器用作指针以生成地址。该指针中 $\pm 32\text{KB}$ 地址区域内的操作数可以直接由指令存取。由于能够用作指针的通用寄存器数量有限，当指针值改变引起地址计算时，为了不影响系统的性能，要指定尽可能多的通用寄存器来保存变量值，这样也会减少程序所占的空间。

(1) 程序空间

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

要将内部 RAM 区域用作程序空间，则访问地址 3FFD800H 至 3FFEFFFH (6 KB)。

注意事项 如果分支指令在内部 RAM 区间的上沿地址处发生，那么将不会产生位于片上外围 I/O 区域中的预取址（无效取址）。

(2) 数据空间

对于 V850ES/IE2 来说，可看作是在 4GB CPU 地址空间内有 64 个 64MB 的地址空间。因此，26 位地址的最低有效位(第 25 位)是作为符号扩展位被扩展至 32 位而用作地址值的。

(a) 地址环绕应用举例

如果 R = r0 (零寄存器) 由 LD/ST disp16[R]指令使用, 那么就可以通过带符号扩展的 disp16 对从 00000000H 开始的±32KB 地址范围空间进行寻址。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, 而且几乎不需要专门用作指针的寄存器。

举例: μ PD70F3714

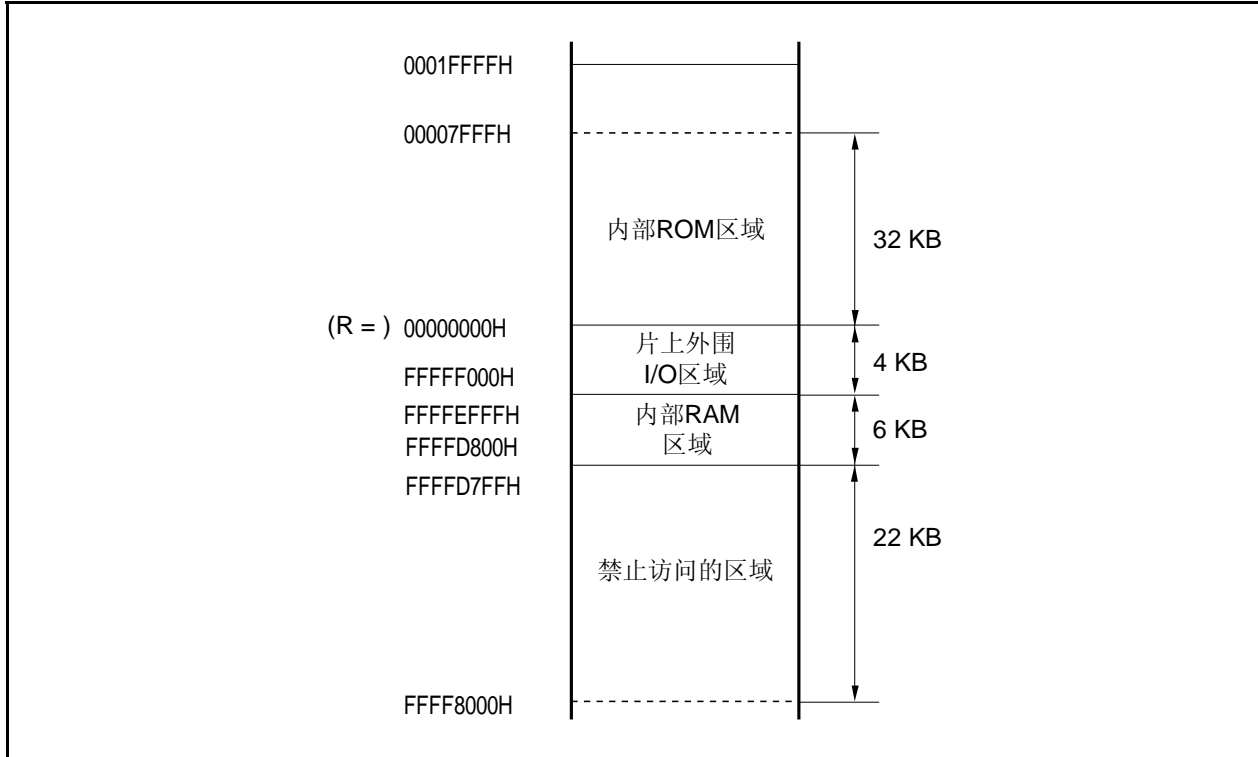
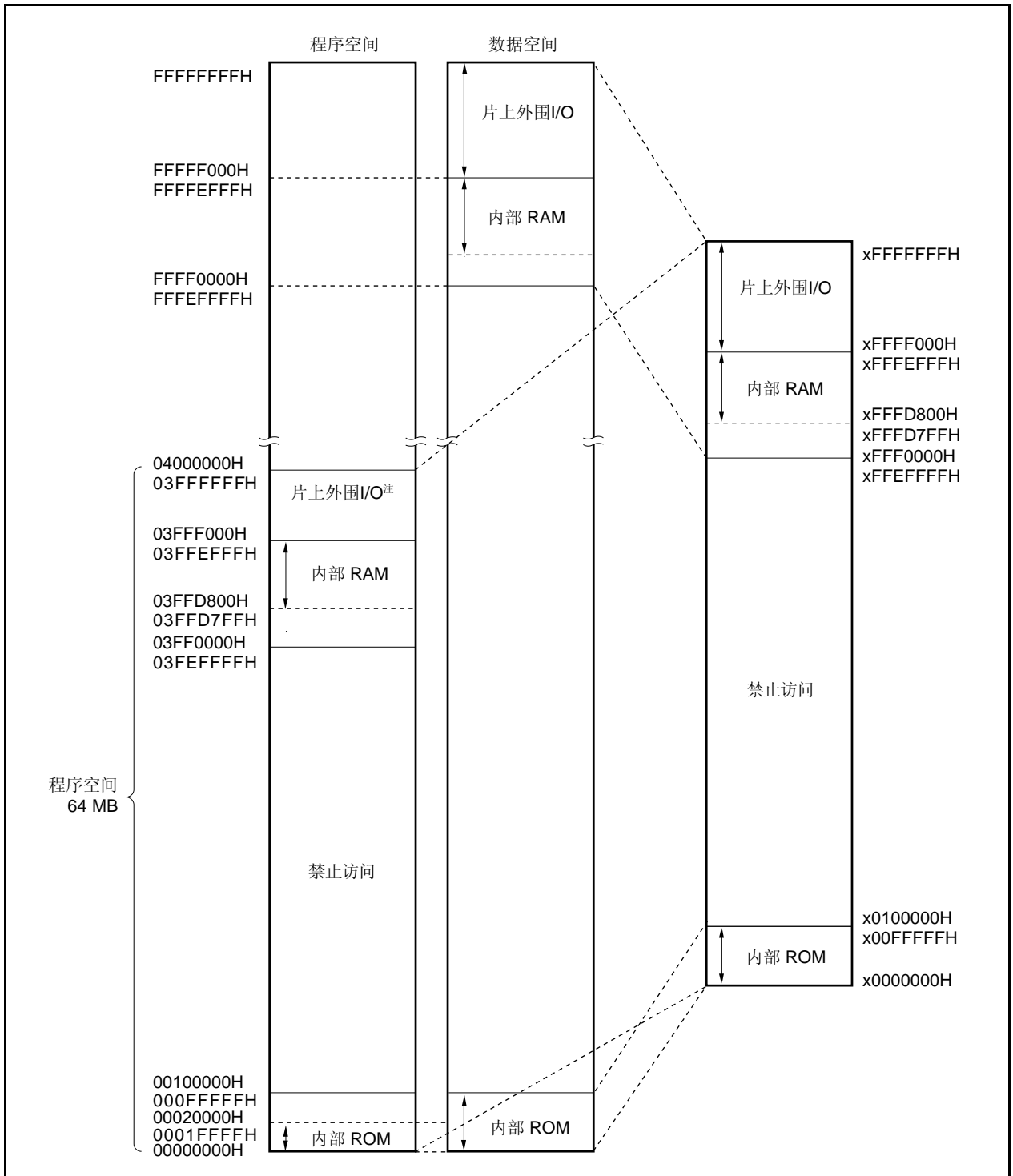


图 3-8. 推荐的内存映射



注 禁止访问该区域。要访问该区域中的片上外围 I/O，则对地址 FFFF000H 至 FFFFFFFFH 进行指定。

- 备注
1. ↓表示推荐的区域。
 2. 该图为 μ PD70F3714 的推荐内存映射图。

3.4.6 片上外围I/O寄存器

(1/6)

地址	功能寄存器名称	符号	R/W	操作位单元			复位后的值
				1	8	16	
FFFFFF004H	端口 DL 寄存器 L	PDLL	R/W	√	√		不确定
FFFFFF024H	端口 DL 模式寄存器 L	PMDL		√	√		FFH
FFFFFF06EH	系统等待控制寄存器	VSWC			√		77H
FFFFFF100H	内部屏蔽寄存器 0	IMR0				√	FFFFH
FFFFFF100H	中断屏蔽寄存器 0L	IMR0L		√	√		FFH
FFFFFF101H	中断屏蔽寄存器 0H	IMR0H		√	√		FFH
FFFFFF102H	中断屏蔽寄存器 1	IMR1				√	FFFFH
FFFFFF102H	中断屏蔽寄存器 1L	IMR1L		√	√		FFH
FFFFFF103H	中断屏蔽寄存器 1H	IMR1H		√	√		FFH
FFFFFF104H	中断屏蔽寄存器 2	IMR2				√	FFFFH
FFFFFF104H	中断屏蔽寄存器 2L	IMR2L		√	√		FFH
FFFFFF105H	中断屏蔽寄存器 2H	IMR2H		√	√		FFH
FFFFFF106H	中断屏蔽寄存器 3	IMR3				√	FFFFH
FFFFFF106H	中断屏蔽寄存器 3L	IMR3L		√	√		FFH
FFFFFF107H	中断屏蔽寄存器 3H	IMR3H		√	√		FFH
FFFFFF110H	中断控制寄存器	PIC0		√	√		47H
FFFFFF112H	中断控制寄存器	PIC1		√	√		47H
FFFFFF114H	中断控制寄存器	PIC2		√	√		47H
FFFFFF116H	中断控制寄存器	PIC3		√	√		47H
FFFFFF118H	中断控制寄存器	PIC4		√	√		47H
FFFFFF11AH	中断控制寄存器	PIC5		√	√		47H
FFFFFF11CH	中断控制寄存器	PIC6		√	√		47H
FFFFFF11EH	中断控制寄存器	LVIIIC		√	√		47H
FFFFFF124H	中断控制寄存器	TQ0OVIC		√	√		47H
FFFFFF126H	中断控制寄存器	TQ0CCIC0		√	√		47H
FFFFFF128H	中断控制寄存器	TQ0CCIC1		√	√		47H
FFFFFF12AH	中断控制寄存器	TQ0CCIC2		√	√		47H
FFFFFF12CH	中断控制寄存器	TQ0CCIC3		√	√		47H
FFFFFF12EH	中断控制寄存器	TQ1OVIC		√	√		47H
FFFFFF130H	中断控制寄存器	TQ1CCIC0		√	√		47H
FFFFFF132H	中断控制寄存器	TQ1CCIC1		√	√		47H
FFFFFF134H	中断控制寄存器	TQ1CCIC2		√	√		47H
FFFFFF136H	中断控制寄存器	TQ1CCIC3		√	√		47H
FFFFFF148H	中断控制寄存器	TP0OVIC		√	√		47H
FFFFFF14AH	中断控制寄存器	TP0CCIC0		√	√		47H
FFFFFF14CH	中断控制寄存器	TP0CCIC1		√	√		47H
FFFFFF14EH	中断控制寄存器	TP1OVIC		√	√		47H
FFFFFF150H	中断控制寄存器	TP1CCIC0		√	√		47H
FFFFFF152H	中断控制寄存器	TP1CCIC1		√	√		47H
FFFFFF154H	中断控制寄存器	TP2OVIC		√	√		47H
FFFFFF156H	中断控制寄存器	TP2CCIC0		√	√		47H

地址	功能寄存器名称	符号	R/W	操作位单元			复位后的值	
				1	8	16		
FFFFF158H	中断控制寄存器	TP2CCIC1	R/W	√	√		47H	
FFFFF15AH	中断控制寄存器	TP3OVIC		√	√		47H	
FFFFF15CH	中断控制寄存器	TP3CCIC0		√	√		47H	
FFFFF15EH	中断控制寄存器	TP3CCIC1		√	√		47H	
FFFFF168H	中断控制寄存器	UA0REIC		√	√		47H	
FFFFF16AH	中断控制寄存器	UA0RIC		√	√		47H	
FFFFF16CH	中断控制寄存器	UA0TIC		√	√		47H	
FFFFF16EH	中断控制寄存器	CB0REIC		√	√		47H	
FFFFF170H	中断控制寄存器	CB0RIC		√	√		47H	
FFFFF172H	中断控制寄存器	CB0TIC		√	√		47H	
FFFFF174H	中断控制寄存器	UA1REIC		√	√		47H	
FFFFF176H	中断控制寄存器	UA1RIC		√	√		47H	
FFFFF178H	中断控制寄存器	UA1TIC		√	√		47H	
FFFFF180H	中断控制寄存器	AD0IC		√	√		47H	
FFFFF182H	中断控制寄存器	AD1IC		√	√		47H	
FFFFF186H	中断控制寄存器	TM0EQIC0		√	√		47H	
FFFFF1FAH	当前服务优先级寄存器	ISPR	R	√	√		00H	
FFFFF1FCH	命令寄存器	PRCMD	W		√		不确定	
FFFFF1FEH	节能控制寄存器	PSC	R/W	√	√		00H	
FFFFF200H	A/D 转换 0 模式寄存器 0	ADA0M0		√	√		00H	
FFFFF201H	A/D 转换 0 模式寄存器 1	ADA0M1		√	√		00H	
FFFFF202H	A/D 转换 0 通道指定寄存器	ADA0S		√	√		00H	
FFFFF203H	A/D 转换 0 模式寄存器 2	ADA0M2		√	√		00H	
FFFFF210H	A/D0 转换结果寄存器 0	ADA0CR0	R			√	不确定	
FFFFF211H	A/D0 转换结果寄存器 0H	ADA0CR0H				√	不确定	
FFFFF212H	A/D0 转换结果寄存器 1	ADA0CR1					√	不确定
FFFFF213H	A/D0 转换结果寄存器 1H	ADA0CR1H				√	不确定	
FFFFF214H	A/D0 转换结果寄存器 2	ADA0CR2					√	不确定
FFFFF215H	A/D0 转换结果寄存器 2H	ADA0CR2H				√	不确定	
FFFFF216H	A/D0 转换结果寄存器 3	ADA0CR3					√	不确定
FFFFF217H	A/D0 转换结果寄存器 3H	ADA0CR3H				√	不确定	
FFFFF220H	A/D 转换 1 模式寄存器 0	ADA1M0	R/W	√	√		00H	
FFFFF221H	A/D 转换 1 模式寄存器 1	ADA1M1		√	√		00H	
FFFFF222H	A/D 转换通道指定寄存器	ADA1S		√	√		00H	
FFFFF223H	A/D 转换 1 模式寄存器 2	ADA1M2		√	√		00H	
FFFFF230H	A/D1 转换结果寄存器 0	ADA1CR0	R			√	不确定	
FFFFF231H	A/D1 转换结果寄存器 0H	ADA1CR0H				√	不确定	
FFFFF232H	A/D1 转换结果寄存器 1	ADA1CR1					√	不确定
FFFFF233H	A/D1 转换结果寄存器 1H	ADA1CR1H				√	不确定	
FFFFF234H	A/D1 转换结果寄存器 2	ADA1CR2					√	不确定
FFFFF235H	A/D1 转换结果寄存器 2H	ADA1CR2H				√	不确定	
FFFFF236H	A/D1 转换结果寄存器 3	ADA1CR3					√	不确定
FFFFF237H	A/D1 转换结果寄存器 3H	ADA1CR3H				√	不确定	

地址	功能寄存器名称	符号	R/W	操作位单元			复位后的值
				1	8	16	
FFFFFF310H	外部中断噪声消除控制寄存器	INTPNRC	R/W	√	√		00H
FFFFFF400H	端口 0 寄存器	P0		√	√		不确定
FFFFFF402H	端口 1 寄存器	P1		√	√		不确定
FFFFFF404H	端口 2 寄存器	P2		√	√		不确定
FFFFFF406H	端口 3 寄存器	P3		√	√		不确定
FFFFFF408H	端口 4 寄存器	P4		√	√		不确定
FFFFFF420H	端口 0 模式寄存器	PM0		√	√		FFH
FFFFFF422H	端口 1 模式寄存器	PM1		√	√		FFH
FFFFFF424H	端口 2 模式寄存器	PM2		√	√		FFH
FFFFFF426H	端口 3 模式寄存器	PM3		√	√		FFH
FFFFFF428H	端口 4 模式寄存器	PM4		√	√		FFH
FFFFFF440H	端口 0 模式控制寄存器	PMC0		√	√		00H
FFFFFF442H	端口 1 模式控制寄存器	PMC1		√	√		00H
FFFFFF444H	端口 2 模式控制寄存器	PMC2		√	√		00H
FFFFFF446H	端口 3 模式控制寄存器	PMC3		√	√		00H
FFFFFF448H	端口 4 模式控制寄存器	PMC4		√	√		00H
FFFFFF462H	端口 1 功能控制寄存器	PFC1		√	√		00H
FFFFFF466H	端口 3 功能控制寄存器	PFC3		√	√		00H
FFFFFF468H	端口 4 功能控制寄存器	PFC4		√	√		00H
FFFFFF540H	TMM0 控制寄存器 0	TM0CTL0		√	√		00H
FFFFFF544H	TMM0 比较寄存器 0	TM0CMP0				√	0000H
FFFFFF5C0H	TMQ0 控制寄存器 0	TQ0CTL0		√	√		00H
FFFFFF5C1H	TMQ0 控制寄存器 1	TQ0CTL1		√	√		00H
FFFFFF5C2H	TMQ0 I/O 控制寄存器 0	TQ0IOC0		√	√		00H
FFFFFF5C3H	TMQ0 I/O 控制寄存器 1	TQ0IOC1		√	√		00H
FFFFFF5C4H	TMQ0 I/O 控制寄存器 2	TQ0IOC2		√	√		00H
FFFFFF5C5H	TMQ0 选择寄存器 0	TQ0OPT0		√	√		00H
FFFFFF5C6H	TMQ0 捕获/比较寄存器 0	TQ0CCR0				√	0000H
FFFFFF5C8H	TMQ0 捕获/比较寄存器 1	TQ0CCR1				√	0000H
FFFFFF5CAH	TMQ0 捕获/比较寄存器 2	TQ0CCR2				√	0000H
FFFFFF5CCH	TMQ0 捕获/比较寄存器 3	TQ0CCR3				√	0000H
FFFFFF5CEH	TMQ0 计数器读取缓冲寄存器	TQ0CNT		R		√	0000H
FFFFFF5F0H	高阻输出控制寄存器 00	HZA0CTL0		R/W	√	√	
FFFFFF5F1H	高阻输出控制寄存器 01	HZA0CTL1	√		√		00H
FFFFFF600H	TMQ1 控制寄存器 0	TQ1CTL0	√		√		00H
FFFFFF601H	TMQ1 控制寄存器 1	TQ1CTL1	√		√		00H
FFFFFF602H	TMQ1 I/O 控制寄存器 0	TQ1IOC0	√		√		00H
FFFFFF605H	TMQ1 选项寄存器 0	TQ1OPT0	√		√		00H
FFFFFF606H	TMQ1 捕获/比较寄存器 0	TQ1CCR0				√	0000H
FFFFFF608H	TMQ1 捕获/比较寄存器 1	TQ1CCR1				√	0000H
FFFFFF60AH	TMQ1 捕获/比较寄存器 2	TQ1CCR2				√	0000H

地址	功能寄存器名称	符号	R/W	操作位单元			复位后的值	
				1	8	16		
FFFF60CH	TMQ1 捕获/比较寄存器 3	TQ1CCR3	R/W			√	0000H	
FFFF60EH	TMQ1 计数器读取缓冲寄存器	TQ1CNT	R			√	0000H	
FFFF620H	TMQ1 选项寄存器 1	TQ1OPT1	R/W	√	√		00H	
FFFF621H	TMQ1 选项寄存器 2	TQ1OPT2		√	√		00H	
FFFF622H	TMQ1 I/O 控制寄存器 3	TQ1IOC3		√	√		A8H	
FFFF623H	TMQ1 选项寄存器 3	TQ1OPT3		√	√		00H	
FFFF624H	TMQ1 停滞时间比较寄存器	TQ1DTC				√	0000H	
FFFF630H	高阻输出控制寄存器 10	HZA1CTL0		√	√		00H	
FFFF631H	高阻输出控制寄存器 11	HZA1CTL1		√	√		00H	
FFFF640H	TMP0 控制寄存器 0	TP0CTL0		√	√		00H	
FFFF641H	TMP0 控制寄存器 1	TP0CTL1		√	√		00H	
FFFF642H	TMP0 I/O 控制寄存器 0	TP0IOC0		√	√		00H	
FFFF643H	TMP0 I/O 控制寄存器 1	TP0IOC1		√	√		00H	
FFFF644H	TMP0 I/O 控制寄存器 2	TP0IOC2		√	√		00H	
FFFF645H	TMP0 选项寄存器 0	TP0OPT0		√	√		00H	
FFFF646H	TMP0 捕获/比较寄存器 0	TP0CCR0				√	0000H	
FFFF648H	TMP0 捕获/比较寄存器 1	TP0CCR1				√	0000H	
FFFF64AH	TMP0 计数器读取缓冲寄存器	TP0CNT		R			√	0000H
FFFF660H	TMP1 控制寄存器 0	TP1CTL0	R/W	√	√		00H	
FFFF661H	TMP1 控制寄存器 1	TP1CTL1		√	√		00H	
FFFF665H	TMP1 选项寄存器 0	TP1OPT0		√	√		00H	
FFFF666H	TMP1 捕获/比较寄存器 0	TP1CCR0				√	0000H	
FFFF668H	TMP1 捕获/比较寄存器 1	TP1CCR1				√	0000H	
FFFF66AH	TMP1 计数器读取缓冲寄存器	TP1CNT		R			√	0000H
FFFF680H	TMP2 控制寄存器 0	TP2CTL0	R/W	√	√		00H	
FFFF681H	TMP2 控制寄存器 1	TP2CTL1		√	√		00H	
FFFF682H	TMP2 I/O 控制寄存器 0	TP2IOC0		√	√		00H	
FFFF683H	TMP2 I/O 控制寄存器 1	TP2IOC1		√	√		00H	
FFFF684H	TMP2 I/O 控制寄存器 2	TP2IOC2		√	√		00H	
FFFF685H	TMP2 选项寄存器 0	TP2OPT0		√	√		00H	
FFFF686H	TMP2 捕获/比较寄存器 0	TP2CCR0				√	0000H	
FFFF688H	TMP2 捕获/比较寄存器 1	TP2CCR1				√	0000H	
FFFF68AH	TMP2 计数器读取缓冲寄存器	TP2CNT		R			√	0000H
FFFF6A0H	TMP3 控制寄存器 0	TP3CTL0		R/W	√	√		00H
FFFF6A1H	TMP3 控制寄存器 1	TP3CTL1	√		√		00H	
FFFF6A2H	TMP3 I/O 控制寄存器 0	TP3IOC0	√		√		00H	
FFFF6A5H	TMP3 选项寄存器 0	TP3OPT0	√		√		00H	
FFFF6A6H	TMP3 捕获/比较寄存器 0	TP3CCR0				√	0000H	
FFFF6A8H	TMP3 捕获/比较寄存器 1	TP3CCR1				√	0000H	
FFFF6AAH	TMP3 计数器读取缓冲寄存器	TP3CNT	R			√	0000H	
FFFF6C0H	振荡稳定时间选择寄存器	OSTS	R/W		√		06H	
FFFF6D0H	看门狗定时器模式寄存器	WDTM			√		67H	
FFFF6D1H	看门狗定时器允许寄存器	WDTE			√		1AH	
FFFF702H	端口 1 功能控制扩展寄存器	PFCE1		√	√		00H	

地址	功能寄存器名称	符号	R/W	操作位单元				复位后的值
				1	8	16	32	
FFFFF802H	系统状态寄存器	SYS	R/W	√	√			00H
FFFFF820H	节能模式寄存器	PSMR		√	√			00H
FFFFF822H	时钟控制寄存器	CKC			√			0AH
FFFFF828H	处理器时钟控制寄存器	PCC		√	√			03H
FFFFF82CH	PLL 控制寄存器	PLLCTL		√	√			01H
FFFFF870H	时钟监视器模式寄存器	CLM		√	√			00H
FFFFF888H	复位源标志寄存器	RESF			√			00H/10H/11H
FFFFF890H	低电压检测寄存器	LVIM		√	√			00H/82H
FFFFF891H	低电压检测电平选择寄存器	LVIS			√			00H
FFFFF892H	内部 RAM 数据状态寄存器	RAMS		√	√			01H
FFFFF8A8H	复位源标志寄存器 2	RESF2	R		√			00H/01H/ 10H/11H
FFFFF8AAH	系统状态寄存器	SYS2	R/W	√	√			00H
FFFFF9ECH	命令寄存器 2	PRCMD2	W		√			不确定
FFFFF9F0H	内部存储器容量设置寄存器	IMS			√			00H
FFFFFA00H	UARTA0 控制寄存器 0	UA0CTL0	R/W	√	√			10H
FFFFFA01H	UARTA0 控制寄存器 1	UA0CTL1			√			00H
FFFFFA02H	UARTA0 控制寄存器 2	UA0CTL2			√			FFH
FFFFFA03H	UARTA0 选项控制寄存器 0	UA0OPT0		√	√			14H
FFFFFA04H	UARTA0 状态寄存器	UA0STR		√	√			00H
FFFFFA06H	UARTA0 接收数据寄存器	UA0RX		R		√		
FFFFFA07H	UARTA0 发送数据寄存器	UA0TX	R/W		√			FFH
FFFFFA10H	UARTA1 控制寄存器 0	UA1CTL0		√	√			10H
FFFFFA11H	UARTA1 控制寄存器 1	UA1CTL1			√			00H
FFFFFA12H	UARTA1 控制寄存器 2	UA1CTL2			√			FFH
FFFFFA13H	UARTA1 选项控制寄存器 0	UA1OPT0		√	√			14H
FFFFFA14H	UARTA1 状态寄存器	UA1STR		√	√			00H
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
FFFFFC00H	外部中断下降沿指定寄存器 0	INTF0		√	√			00H
FFFFFC20H	外部中断上升沿指定寄存器 0	INTR0		√	√			00H
FFFFFC40H	上拉选项寄存器 0	PU0		√	√			00H
FFFFFC42H	上拉选项寄存器 1	PU1		√	√			00H
FFFFFC44H	上拉选项寄存器 2	PU2		√	√			00H
FFFFFC46H	上拉选项寄存器 3	PU3		√	√			00H
FFFFFC48H	上拉选项寄存器 4	PU4		√	√			00H
FFFFFD00H	CSIB0 控制寄存器 0	CB0CTL0		√	√			01H
FFFFFD01H	CSIB0 控制寄存器 1	CB0CTL1		√	√			00H
FFFFFD02H	CSIB0 控制寄存器 2	CB0CTL2		√			00H	
FFFFFD03H	CSIB0 状态寄存器	CB0STR	√	√			00H	

(6/6)

地址	功能寄存器名称	符号	R/W	操作位单元			复位后的值	
				1	8	16		
FFFFFFD04H	CSIB0 接收数据寄存器	CB0RX	R			√	0000H	
FFFFFFD04H	CSIB0 接收数据寄存器 L	CB0RXL			√			00H
FFFFFFD06H	CSIB0 发送数据寄存器	CB0TX	R/W			√	0000H	
FFFFFFD06H	CSIB0 发送数据寄存器 L	CB0TXL			√			00H
FFFFFF44H	上拉选项寄存器 DLL	PUDLL		√	√			00H

3.4.7 特殊寄存器

特殊寄存器是为了防止由程序挂起所引起的数据非法写入而受到保护的寄存器。
V850ES/IE2 具有以下七种特殊寄存器，这些寄存器被分为两种类型。

[支持由 SYS.PRERR 位发出错误报告的特殊寄存器]

- 节能控制寄存器(PSC)
- 时钟控制寄存器 (CKC)
- 处理器时钟控制寄存器 (PCC)
- 时钟监视器模式寄存器 (CLM)
- 复位源标志寄存器 (RESF)

<R> **注意事项** 对以上五种特殊寄存器中任何一种进行写入时，使用命令寄存器 **PRCMD**。

[支持由 SYS.PRERR2 位发出错误报告的特殊寄存器]

- 低电压检测寄存器 (LVIM)。
- 内部 RAM 数据状态寄存器 (RAMS)

<R> **注意事项** 对以上两种特殊寄存器中任何一种进行写入时，使用命令寄存器 **PRCMD2**。

此外，器件还提供了一个命令寄存器来防止对特殊寄存器的非法写操作，这样应用系统就不会因为程序挂起而停止工作了。对特殊寄存器的写操作是通过特定流程实现的，非法保存操作将记录在系统状态寄存器中。

(1) 设置特殊寄存器数据

按照以下步骤设置特殊寄存器数据：

- <1> 将向特殊寄存器设置的数据放入通用寄存器中。
- <2> 将步骤 <1> 中存储的数据写入命令寄存器。
- <3> 将设置数据写入特殊寄存器中（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）。
- (<4>至<8> 插入 NOP 指令（5 条）^註

[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 IDLE 和 STOP 模式）。
<1>MOV 0x02, r10
<2>ST.B r10, PRCMD[r0]     ; 写入 PRCMD 寄存器。
<3>ST.B r10, PSC[r0]      ; 设置 PSC 寄存器。
<4>NOP註                   ; 空操作指令
<5>NOP註                   ; 空操作指令
<6>NOP註                   ; 空操作指令
<7>NOP註                   ; 空操作指令
<8>NOP註                   ; 空操作指令
(下一条指令)

```

读取特殊寄存器时没有特定流程。

注 设置 IDLE1 模式或 STOP 模式（通过将 PSC.STB 位置 1）后，要立即插入 5 条 NOP 指令或更多。

- 注意事项**
1. 执行存储指令以对命令寄存器进行写操作时，将不响应中断。 应该是通过连续的存储指令执行上述步骤<2> 和 <3>的。 若在步骤 <2> 和 <3> 之间安置了其他指令，而该指令又引起了中断响应，则将打乱上述的顺序导致设备误操作。
 2. 尽管对命令寄存器写入的数据为伪数据，但还是请通过使用保存指令来使用与设置特殊寄存器（实例中的步骤<3>）的通用寄存器相同的寄存器，用来向命令寄存器（实例中的步骤<2>）写入伪数据。当通用寄存器用于寻址时，同样使用这种用法。
使用位操作指令设置特殊寄存器（实例中的步骤<3>）的示例如下所示：

```
CLR1 0, RESF[r0]
```

(2) 命令寄存器

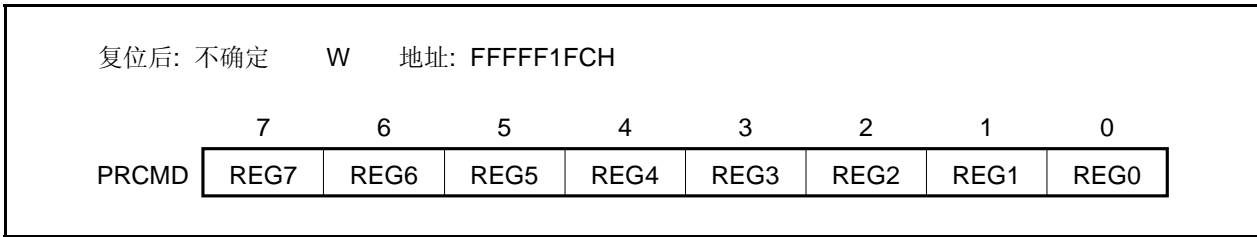
PRCMD 是用于保护那些会严重影响应用系统的寄存器不被轻易写入，从而使应用系统不会因为程序挂起而意外停止工作的 8 位寄存器。该寄存器可以通过一个特殊寄存器的设置用作 PRCMD 或 PRCMD2。数据预先写入命令寄存器后，至特殊寄存器的第一次写操作才是有效的。这样，特殊寄存器的值仅能以特定的流程进行写入，这样就能防止对特殊寄存器的非法写操作。

(a) 命令寄存器(PRCMD)

PRCMD 寄存器仅能按字节进行写操作（若对其进行读取，则所读数据不确定）。

复位会使该寄存器的内容不确定。

注意事项 写入 PRCMD 寄存器后，则至特殊寄存器 PSC、CKC、PCC、CLM 和 RESF 的写入操作使能。

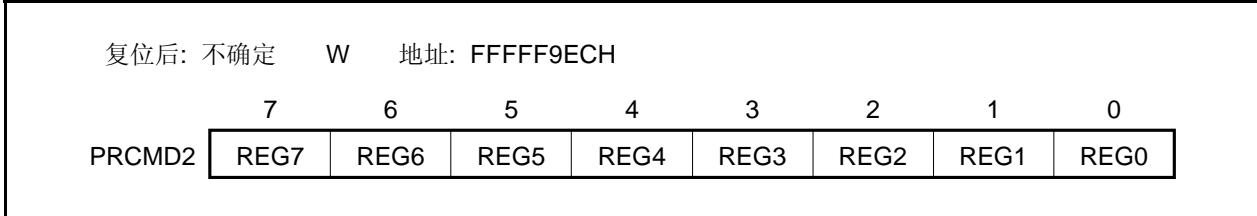


(b) 命令寄存器 2 (PRCMD2)

PRCMD2 寄存器仅能按字节进行写操作（若对其进行读取，则所读数据不确定）。

复位会使该寄存器的内容不确定。

注意事项 写入 PRCMD2 寄存器后，则至特殊寄存器 LVIM 和 RAMS 的写入操作使能。



(3) 系统状态寄存器

该寄存器中包含了用于显示整个系统的操作状态的状态标志。通过特殊寄存器的设置，该寄存器可用作 SYS 或 SYS2。

(a) 系统状态寄存器(SYS)

如果没有按正确的顺序对该寄存器进行写操作，其中包括对 PRCMD 寄存器的访问，那么数据将不会写入预期的寄存器，也将会发生保护错误，且会设置 PRERR 标志。由 CPU 中的指令对该寄存器进行清零操作。

该寄存器可按字节或位进行读写。

复位输入将该寄存器设为 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		保护错误检测					
	0	未产生保护错误						
	1	产生保护错误						

(b) 系统状态寄存器 2 (SYS2)

如果没有按正确的顺序对该寄存器进行写操作，其中包括对 PRCMD2 寄存器的访问，那么数据将不会写入预期的寄存器，也将会发生保护错误，且会设置 PRERR2 标志。由 CPU 中的指令对该寄存器进行清零操作。

该寄存器可按字节或位进行读写。

复位输入将该寄存器设为 00H。

复位后: 00H		R/W		地址: FFFFF8AAH				
SYS2	7	6	5	4	3	2	1	<0>
	0	0	0	0	0	0	0	PRERR2
	PRERR2		保护错误检测					
	0	未产生保护错误						
	1	产生保护错误						

PRERR 标志的操作条件如下所示： 有关 PRERR2 标志的操作条件，请在下述说明中将 PRCMD 和 SYS 当作 PRCMD2 和 SYS2 来阅读。

(i) 设置条件 (PRERR 标志=1)

- 当向特殊寄存器写入数据之前没有对 PRCMD 寄存器进行写入时（在执行 **3.4.7 (1) 至特殊寄存器写入数据** 中的步骤 <3> 之前没有执行步骤 <2>）
- 当对 PRCMD 寄存器进行写入操作后，没有进行对特殊寄存器的写入，而是写入了片上周边 I/O 寄存器时（如果 **3.4.7 (1) 至特殊寄存器写入数据** 中的步骤<3>不是对特殊寄存器的设置）

备注 即使在对 PRCMD 寄存器进行写访问和对特殊寄存器进行写访问（例如对内部 RAM 的访问）期间读取片上外围 I/O 寄存器（执行位操作指令除外），也不会设置 PRERR 标志，且数据可能会写入特殊寄存器。

(ii) 清零条件 (PRERR 标志=0)

- 将 0 写入 PRERR 标志位时。
- 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（并非特殊寄存器）的 PRERR 位，那么 PRERR 位将清为零（写访问优先）。
2. 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（该寄存器并不是特殊寄存器），那么，PRERR 位被设为 1。

3.4.8 系统等待控制寄存器 (VSWC)

VSWC 是为片上外围 I/O 寄存器控制总线访问等待的寄存器。

然而，在 V850ES/IE2 中，对于 V850ES CPU 核心的片上外围 I/O 寄存器的访问是在 3 个时钟内完成的，除了 3 个时钟外还需要一个等待。设置 11H（设置等待 2 个时钟）至 VSWC 寄存器。

VSWC 寄存器以字节为单位进行读写（地址：FFFFFF06EH，初始值：77H）。

CPU 时钟频率 (f_{CPU})	VSWC 设置值
$312.5 \text{ kHz} \leq f_{CPU} \leq 20 \text{ MHz}$	11H

注意事项 使用 V850ES/IE2 时，必须先对 VSWC 寄存器进行设置。

设置 VSWC 寄存器后再根据需要对其他寄存器进行设置。

备注 当对包括显示片上外围功能（诸如 STATUS1n 的寄存器）的状态标志的寄存器，或显示定时器计数值的寄存器进行访问时，如果标志和计数值改变时的时序和访问寄存器的时序发生重叠，那么将会执行寄存器访问重试操作。因此，访问片上外围 I/O 寄存器可能需要一段比较长的时间。

第四章 端口功能

4.1 特点

- 端口： 39
- 允许以 1 位为单位读取输入数据 / 写入输出数据。
- 片内上拉电阻可以以 1 位为单位进行连接(仅限于端口 0 至 4 以及 DL)。

然而,端口处于输入模式且端口功能的复用功能引脚用作输入引脚时,可以连接片内上拉电阻。当这些引脚由于 TOP2OFF, TOQ1OFF, TOP3OFF, 以及 TOQH0OFF 引脚或软件的处理而进入高阻抗状态时,片内上拉电阻可以连接到 TOP21, TOQ1T1 至 TOQ1T3, TOQ1B1 至 TOQ1B3, TOP31, 以及 TOQH01 至 TOQH03 引脚,而这些引脚在使用相应端口的复用功能时用作输出引脚。

4.2 端口基本配置

V850ES/IE2 总共包括 39 个标识为端口 0 至 4 以及 DL 的 I/O 端口。端口配置如下所示:

图 4-1. 端口配置

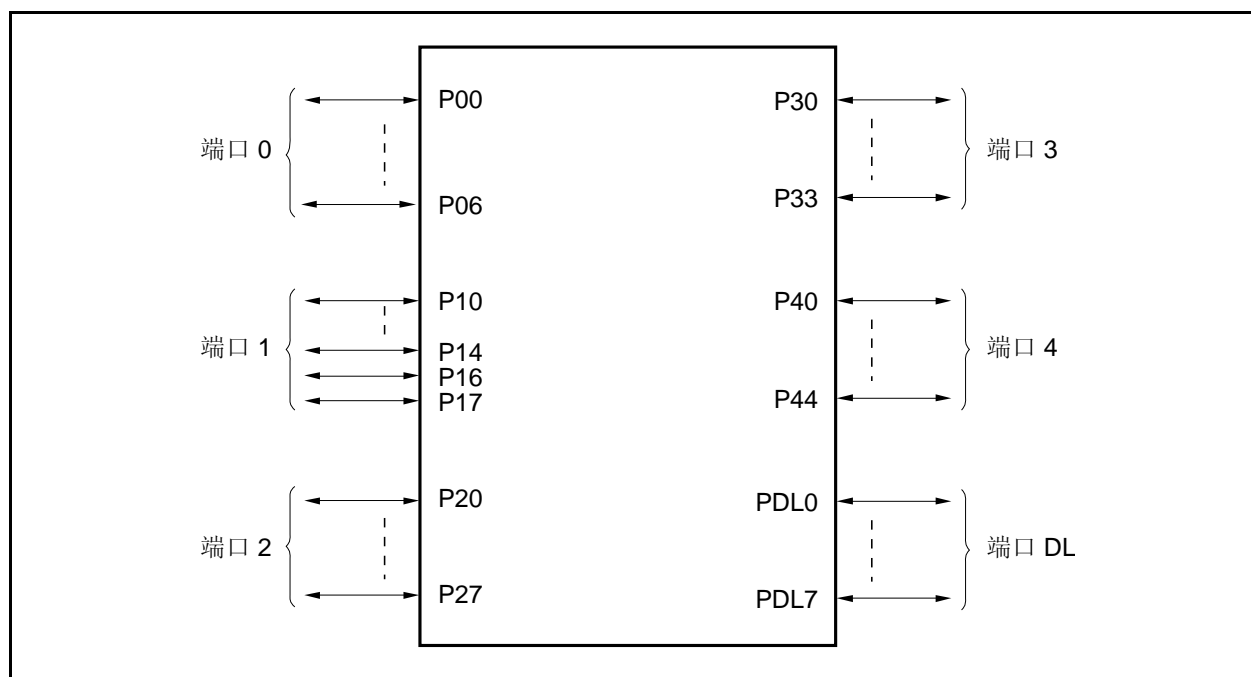


表 4-1. 各引脚 I/O 缓冲器的供电电源

供电电源	对应引脚
AV _{DD0} 、AV _{DD1}	ANI00 至 ANI03、ANI10 至 ANI13
EV _{DD}	端口 0 至 4、端口 DL、RESET

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 寄存器 (Pn: n = 0 至 4、DLL) 端口 n 模式寄存器 (PMn: n = 0 至 4、DLL) 端口 n 模式控制寄存器 (PMcn: n = 0 至 4) 端口 n 功能控制寄存器 (PFCn: n = 1, 3, 4) 端口 1 功能控制扩展寄存器 (PFCE1) 上拉电阻选项寄存器 (PUn: n = 0 至 4、DLL)
端口	I/O: 39
上拉电阻	软件控制: 39

(1) 端口 n 寄存器 (Pn)

通过写入或读取 Pn 寄存器，将数据从外部器件输入或输出到外部器件。

Pn 寄存器由一个保存输出数据的端口锁存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: 未定义 R/W								
	7	6	5	4	3	2	1	0
Pn	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0
	输出数据的控制(输出模式下)							
	Pnm							
	0	输出 0。						
	1	输出 1。						

无论 PMcn 寄存器如何设置，Pn 寄存器的数据写入或读取如下所示：

表 4-3. Pn 寄存器的写入和读出

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1} 。 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)	数据输出写入输出锁存器。 不影响引脚状态 ^{注1}	读取引脚状态 ^{注3} 。

- 注
1. 写入输出锁存器的值一直保留，直到新值写入输出锁存器为止。
 2. 同样，PMn 寄存器处于输出模式并且设置为复用功能时，Pn 寄存器的值将被读取。
 3. 如果 PMn 寄存器处于输出模式并且设置为复用功能，则不论复用功能是输入还是输出功能，都将会读取引脚状态。

(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，而且输入或输出模式可以按位设定。

复位后: FFH		R/W						
	7	6	5	4	3	2	1	0
PMn	PMn7	PMn6	PMn5	PMn4	PMn3	PMn2	PMn1	PMn0
PMnm	输入的控制/输出模式							
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，而且端口的模式可以按位设定。

复位后: 00H		R/W						
	7	6	5	4	3	2	1	0
PMcn	PMcn7	PMcn6	PMcn5	PMcn4	PMcn3	PMcn2	PMcn1	PMcn0
PMcnm	操作模式的规范							
0	端口模式							
1	复用功能							

(4) 端口 n 功能控制寄存器 (PFCn)

如果引脚有两个复用功能，PFCn 寄存器指定要使用端口引脚的复用功能。

该寄存器的每一位对应端口 n 的一个引脚，而且端口的复用功能可以按位设定。

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

(5) 端口 n 功能控制扩展寄存器(PFCEn)

如果引脚有三个或更多的复用功能，PFCEn 寄存器指定要使用引脚的复用功能。
该寄存器的每一位对应端口 n 的一个引脚，而且端口的复用功能可以按位设定。

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	复用功能的规范					
	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) 上拉电阻选项寄存器 (PUn)

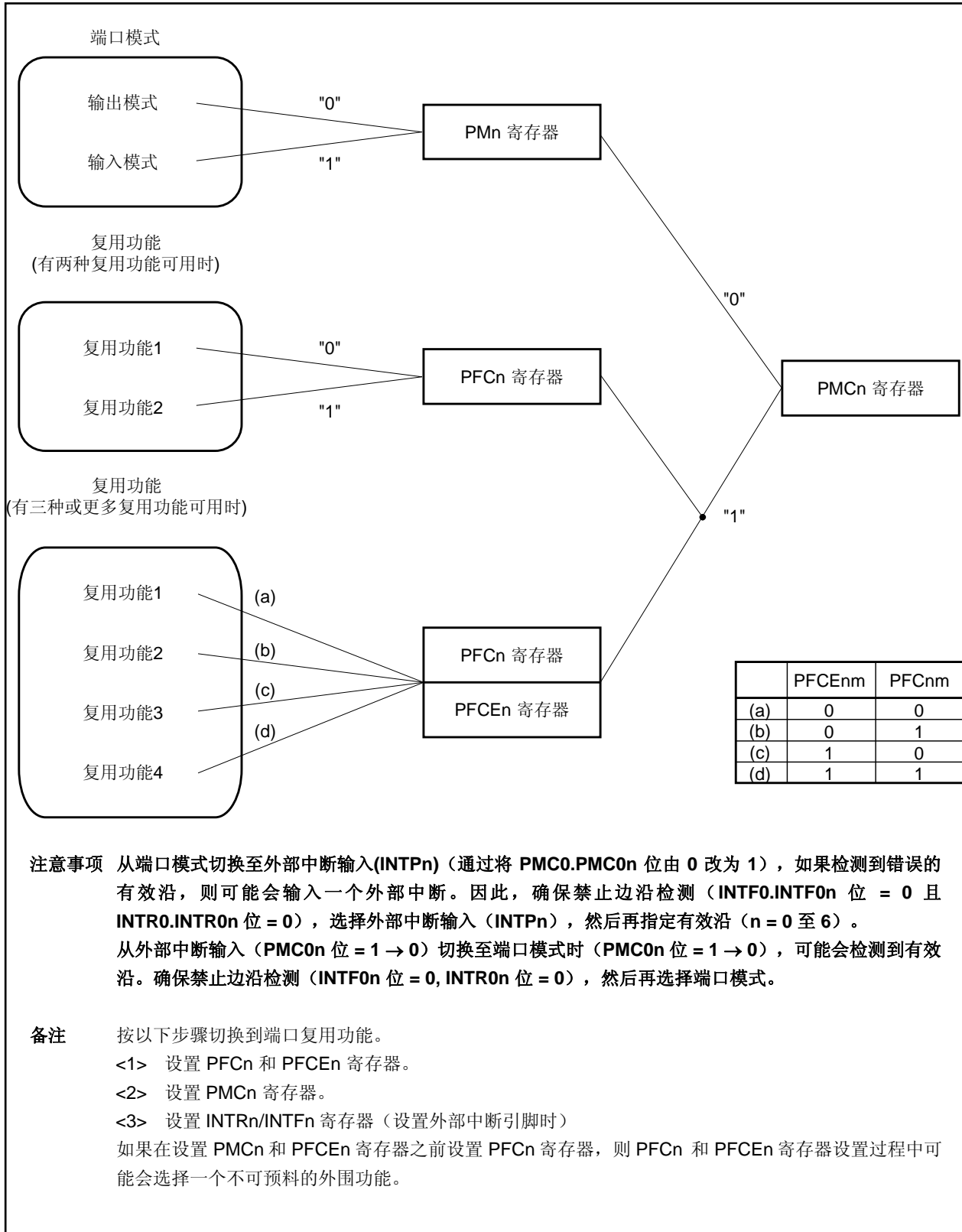
上拉电阻选项寄存器 PUn 指定片内上拉电阻是否连接。
上拉电阻选项寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定。

复位后: 00H		R/W						
	7	6	5	4	3	2	1	0
PUn	PUn7	PUn6	PUn5	PUn4	PUn3	PUn2	PUn1	PUn0
	PUnm	片内上拉电阻连接的控制						
	0	未连接						
	1	已连接						

(7) 端口设置

如下所示进行端口设置：

图 4-2. 寄存器设置和引脚功能



4.3.1 端口 0

端口 0 可用 1 位单元设置为输入或输出模式。

端口 0 具有以下的复用功能引脚。：

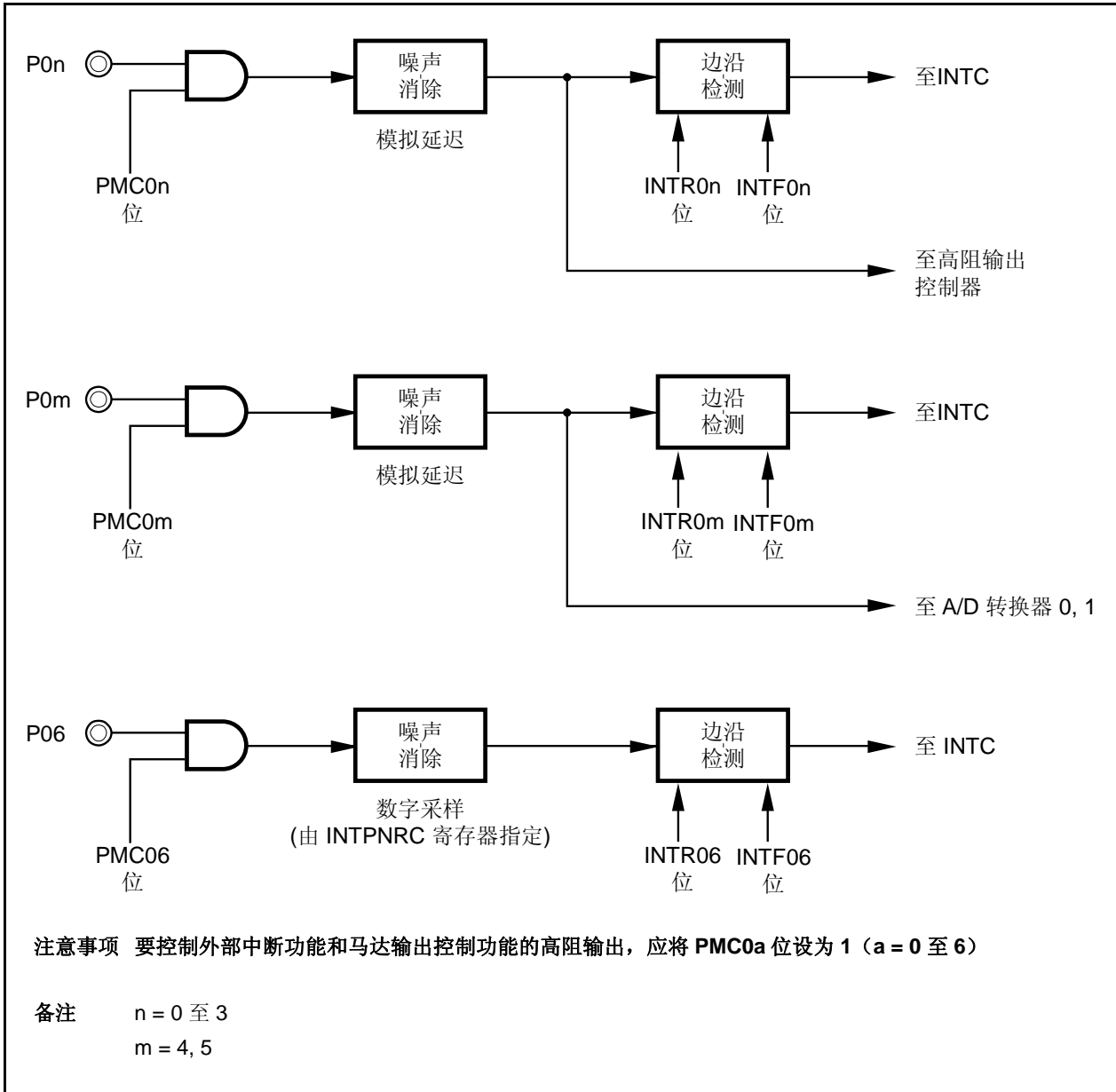
表 4-4. 端口 0 的复用功能引脚

引脚名称	引脚编号	复用功能引脚	输入/输出	上拉 ^{注 1}
P00 ^{注 2}	17	INTP0/TOQH0OFF ^{注 3}	输入	提供
P01 ^{注 2}	16	INTP1/TOQ1OFF ^{注 3}	输入	
P02 ^{注 2}	15	INTP2/TOP2OFF ^{注 3}	输入	
P03 ^{注 2}	14	INTP3/TOP3OFF ^{注 3}	输入	
P04 ^{注 2}	13	INTP4/ADTRG0 ^{注 3}	输入	
P05 ^{注 2}	12	INTP5/ADTRG1 ^{注 3}	输入	
P06 ^{注 2}	11	INTP6	输入	

- 注
1. 软件上拉功能
 2. 这些引脚在端口模式下读取时用作施密特触发输入。
 3. 经过端口消除噪声（模拟延迟）后，TOQH0OFF、TOQ1OFF、TOP2OFF、TOP3OFF、ADTRG0 以及 ADTRG1 信号输入到高阻抗输出控制器（参见第九章 马达控制功能）和 A/D 转换器 0 和 1（参见第十一章 A/D 转换器 0 和 1）。此外，被检测到边沿的信号作为 INTP_n（n = 0 至 5）输入到中断控制器（INTC）。边沿检测通过高阻抗输出控制器和 A/D 转换器 0 和 1 来完成。

- 注意事项
1. 要控制马达控制定时器的高阻抗输出，应先确保将 PMC0.PMC0n 位设为 1，然后再指定要检测的边沿并允许高阻抗输出控制器（n = 1 至 3）的操作，这是因为在检测到错误的有效沿时，马达控制定时器的输出可能会进入高阻抗状态。
 2. 要将 A/D 触发信号输入到 A/D 转换器 0 或 1，应先确保将 PMC0.PMC0n 位设为 1，然后再指定要检测的边沿并允许 A/D 转换器 0 或 1 的操作，因为在检测到错误的有效沿（n = 4, 5）时可能会输入触发信号。
 3. 从端口模式切换至外部中断输入（INTP_n）时（通过将 PMC0.PMC0n 位由 0 改为 1），如果检测到错误的有效沿，则可能会输入一个外部中断。因此，确保禁止边沿检测（INTF0.INTF0n 位 = 0 且 INTR0.INTR0n 位 = 0），选择外部中断输入（INTP_n），然后再指定有效沿（n = 0 至 6）。从外部中断输入切换至端口模式时（PMC0n 位 = 1 → 0），可能会检测到有效沿。确保禁止边沿检测（INTF0n 位 = 0, INTR0n 位 = 0），然后再选择端口模式。

噪声消除功能包含在端口 0 的复用功能中。



(1) 寄存器

(a) 端口 0 寄存器 (P0)

复位后: 未定义 R/W 地址: FFFFF400H

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

P0n	输出数据的控制 (输出模式下) (n = 0 至 6)
0	输出 0。
1	输出 1。

(b) 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	输入控制/输出模式 (端口模式下) (n = 0 至 6)
0	输出模式
1	输入模式

(c) 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 引脚的操作模式规范							
0	I/O 端口							
1	INTP6 输入							
PMC05	P05 引脚的操作模式规范							
0	I/O 端口							
1	INTP5 输入/ADTRG1 输入							
PMC04	P04 引脚的操作模式规范							
0	I/O 端口							
1	INTP4 输入/ADTRG0 输入							
PMC03	P03 引脚的操作模式规范							
0	I/O 端口							
1	INTP3 输入/TOP3OFF 输入							
PMC02	P02 引脚的操作模式规范							
0	I/O 端口							
1	INTP2 输入/TOP2OFF 输入							
PMC01	P01 引脚的操作模式规范							
0	I/O 端口							
1	INTP1 输入/TOQ1OFF 输入							
PMC00	P00 引脚的操作模式规范							
0	I/O 端口							
1	INTP0 输入/TOQH0OFF 输入							

(d) 端口 0 上拉电阻选项寄存器 (PU0)

复位后: 00H R/W 地址: FFFFFC40H

	7	6	5	4	3	2	1	0
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00
PU0n	片内上拉电阻连接的控制 (n = 0 至 6)							
0	不连接							
1	连接 ^注							

注 片内上拉电阻只有在引脚处于端口模式或引脚功能为复用功能时才可以进行连接。引脚处于输出状态时不能连接上拉电阻。

(2) 框图

图 4-3. P00 至 P05 引脚的框图

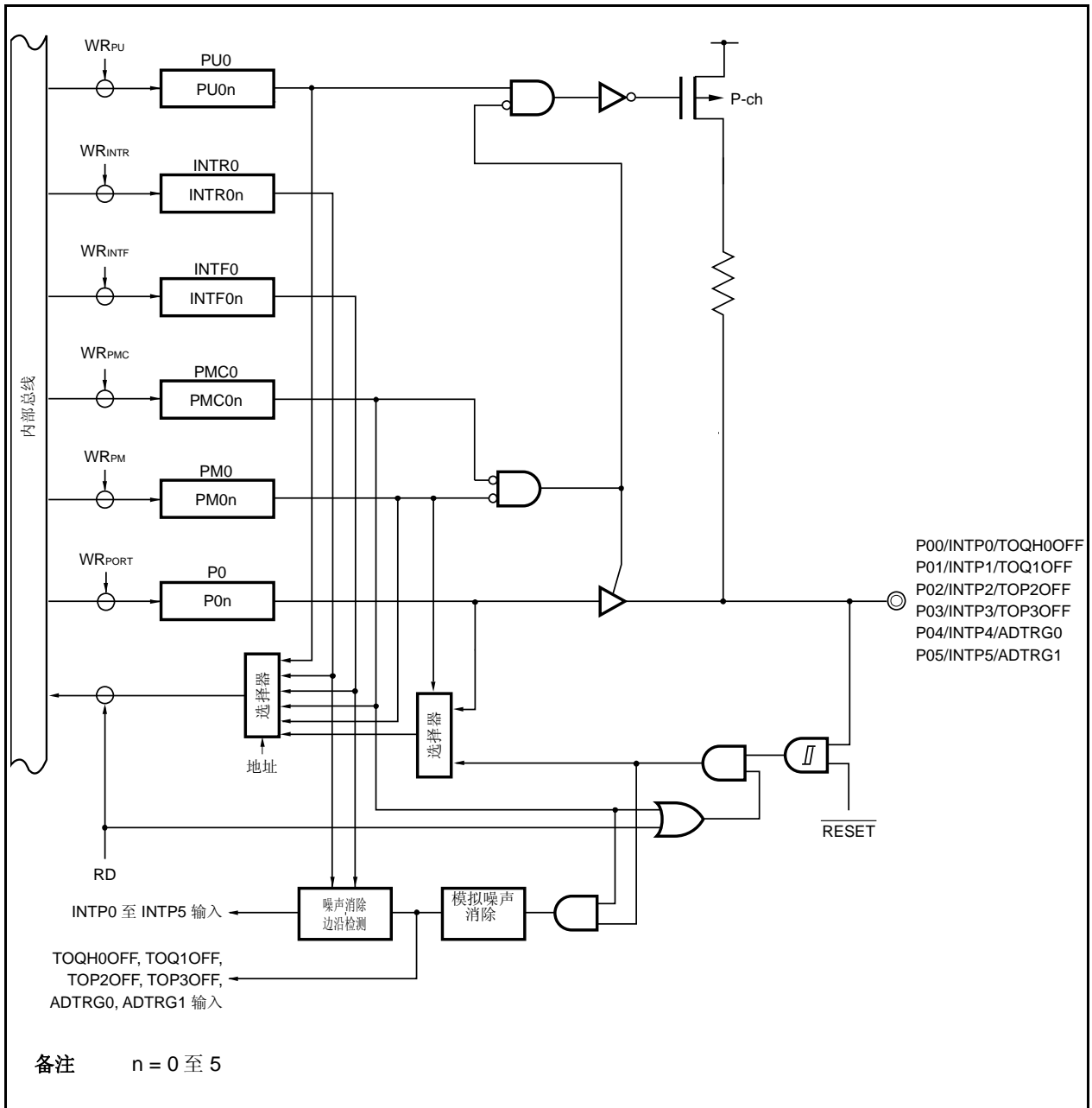
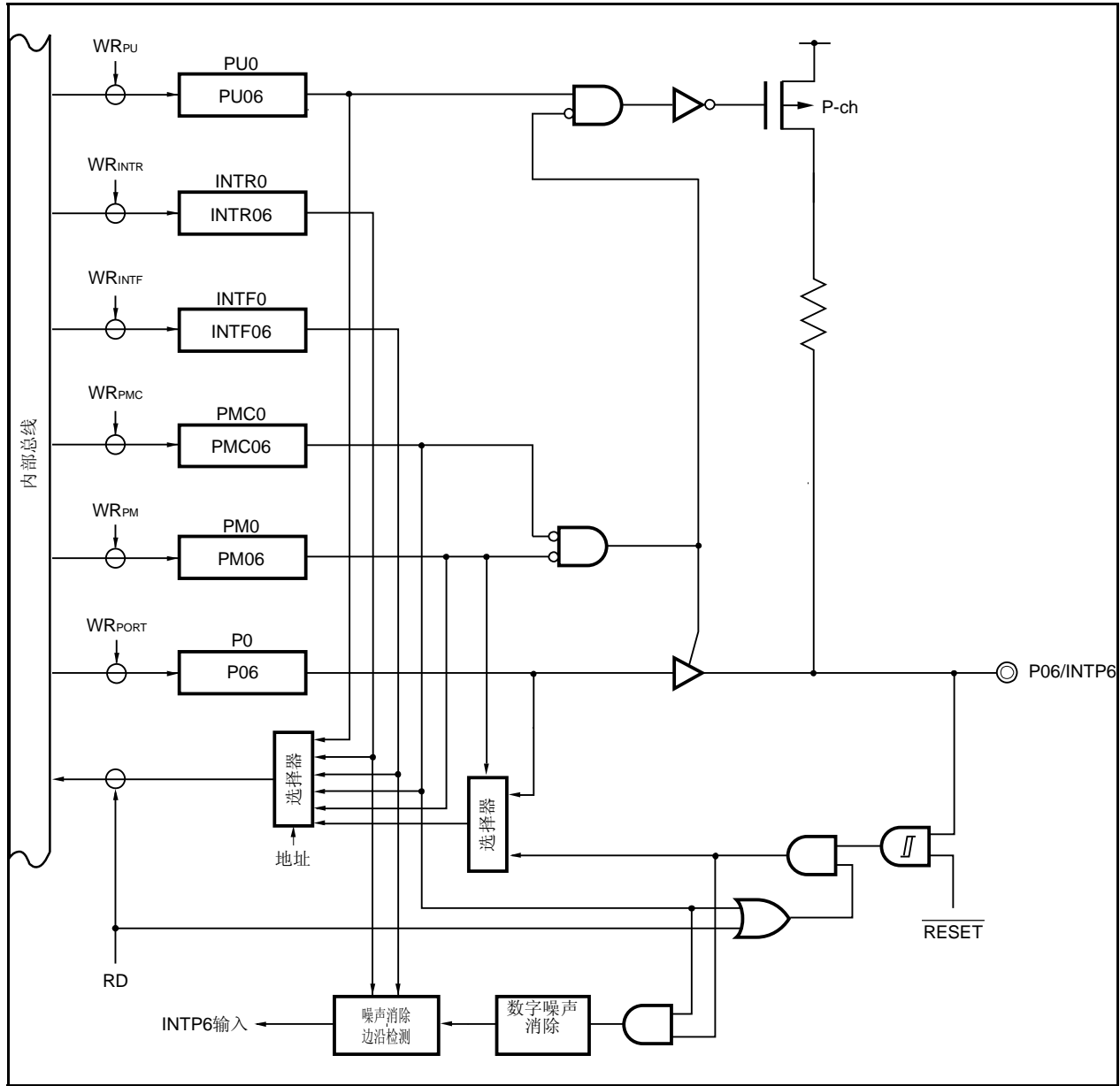


图 4-4. P06 引脚框图



4.3.2 端口 1

端口 1 可用 1 位单元设置为输入或输出模式。

端口 1 具有以下的复用功能引脚：

表 4-5. 端口 1 的复用功能引脚

引脚名称	引脚编号	复用功能引脚	输入/输出	上拉 ^{注1}
P10 ^{注2}	24	TOQH01/TIQ01/TOQ01	I/O	提供
P11 ^{注2}	23	TIQ02/TOQ02	I/O	
P12 ^{注2}	22	TOQH02/TIQ03/TOQ03	I/O	
P13 ^{注2}	21	TIQ00	输入	
P14 ^{注2}	20	TOQH03/EVTQ0	输入	
P16 ^{注2} (CLMER)	19	TOQ00(CLMER)/TIP20	I/O	
P17 ^{注2}	18	TOP21/TIP21	I/O	

- 注
1. 软件上拉控制功能
 2. 这些引脚在端口模式下读取时用作施密特触发输入。

注意事项 1. 当 P10、P12、P14 和 P17 用作 TOQH01 至 TOQH03 和 TOP21 时，输出将会在以下信号输出时停止。

- 高阻输出控制器的高阻设置信号的输出
 - 时钟监视器的时钟停止检测信号的输出
2. 如果将 P16 设置为输出端口或 TOQ00 输出功能，那么时钟监视器检测到错误（振荡器停止）时，CLMER 信号（低电平）将会从 P16 中输出。如果将 P16 设置为输入端口或 TIP20 输入功能，那么 CLMER 信号将不会被输出。输出 CLMER 信号时，将会保持输出 CLMER 信号直到复位信号生成为止。CLMER 信号有效时，P16、PM16、PMC16、PFC16 以及 PU16 位不能写入。

(1) 寄存器

(a) 端口 1 寄存器 (P1)

复位后: 未定义 R/W 地址: FFFFF402H

	7	6	5	4	3	2	1	0
P1	P17	P16	0	P14	P13	P12	P11	P10

P1n	输出数据控制 (输出模式下) (n = 0 至 4, 6, 7)
0	输出 0。
1	输出 1。

(b) 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	PM17	PM16	1	PM14	PM13	PM12	PM11	PM10

PM1n	输入控制/输出模式 (端口模式下) (n = 0 至 4, 6, 7)
0	输出模式 ^注
1	输入模式

注 如果 P16 用作输出端口，那么时钟监测器检测到错误（振荡器停止）时，CLMER 信号（低电平）将会从 P16 中输出。低电平输出通过复位信号来释放。详情参见表 4-5 端口 1 的复用功能引脚。

(c) 端口 1 模式控制寄存器(PMC1)

复位后: 00H R/W 地址: FFFFF442H

	7	6	5	4	3	2	1	0
PMC1	PMC17	PMC16	0	PMC14	PMC13	PMC12	PMC11	PMC10
	PMC17	P17 引脚的操作模式规范						
	0	I/O 端口						
	1	TOP21 输出/TIP21 输入						
	PMC16	P16 引脚的操作模式规范						
	0	I/O 端口						
	1	TOQ00 (CLMER) 输出/TIP20 输入						
	PMC14	P14 引脚的操作模式规范						
	0	I/O 端口						
	1	TOQH03 输出/EVTQ0 输入						
	PMC13	P13 引脚的操作模式规范						
	0	I/O 端口						
	1	TIQ00 输入						
	PMC12	P12 引脚的操作模式规范						
	0	I/O 端口						
	1	TOQH02 输出/TIQ03 输入/TOQ03 输出						
	PMC11	P11 引脚的操作模式规范						
	0	I/O 端口						
	1	TIQ02 输入/TOQ02 输出						
	PMC10	P10 引脚的操作模式规范						
	0	I/O 端口						
	1	TOQH01 输出/TIQ01 输入/TOQ01 输出						

(d) 端口 1 功能控制寄存器(PFC1)

复位后: 00H R/W 地址: FFFFF462H

	7	6	5	4	3	2	1	0
PFC1	PFC17	PFC16	0	PFC14	PFC13	PFC12	PFC11	PFC10

备注 关于复用功能指定的详细信息，参见 4.3.2 (1) (f) 端口 1 的复用功能的设置。

(e) 端口 1 功能控制扩展寄存器 (PFCE1)

复位后: 00H R/W 地址: FFFFF702H

	7	6	5	4	3	2	1	0
PFCE1	0	0	0	0	0	PFCE12	PFCE11	PFCE10

备注 关于复用功能指定的详细信息，参见 4.3.2 (1) (f) 端口 1 的复用功能的设置。

(f) 端口 1 的复用功能的设置

PFC17	P17 引脚的复选功能规范
0	TOP21 输出
1	TIP21 输入

PFC16	P16 引脚的复选功能规范
0	TOQ00 (CLMER) 输出 ^{注1}
1	TIP20 输入

PFC14	P14 引脚的复选功能规范
0	TOQH03 输出 ^{注2}
1	EVTQ0 输入

PFC13	P13 引脚的复选功能规范
0	设置禁止
1	TIQ00 输入

PFCE12	PFC12	P12 引脚的复选功能规范
0	0	TOQH02 输出 ^{注2}
0	1	TIQ03 输入
1	0	TOQ03 输出
1	1	设置禁止

PFCE11	PFC11	P11 引脚的复选功能规范
0	0	设置禁止
0	1	TIQ02 输入
1	0	TOQ02 输出
1	1	设置禁止

PFCE10	PFC10	P10 引脚的复选功能规范
0	0	TOQH01 输出 ^{注2}
0	1	TIQ01 输入
1	0	TOQ01 输出
1	1	设置禁止

- 注
1. 如果 P16 用作 TOQ00 输出引脚，那么时钟监测器检测到错误（振荡器停止）时，CLMER 信号（低电平）将会从 P16 中输出。低电平输出通过复位信号来释放。详情参见表 4-5 端口 1 的复用功能引脚。
 2. TMQ0 不处于 PWM 输出模式时这些设置将会被禁止。

(g) 端口 1 上拉电阻选项寄存器(PU1)

复位后: 00H R/W 地址: FFFFC42H

	7	6	5	4	3	2	1	0
PU1	PU17	PU16	0	PU14	PU13	PU12	PU11	PU10

PU1n	片内上拉电阻连接控制 (n = 0 至4, 6, 7)
0	不连接
1	连接 ^注

注 片内上拉电阻只有在引脚处于端口模式的输入模式或处于复用功能模式的输入引脚时才可以进行连接。此外，当引脚在复用功能模式下由于 TOQH0OFF 和 TOP2OFF 引脚或软件处理而进入高阻状态时，片内上拉电阻将只能连接到 TOQH01 至 TOQH03 和 TOP21 引脚上。
引脚处于输出状态时不能连接电阻。

(2) 框图

图 4-5. P10 和 P12 引脚的框图

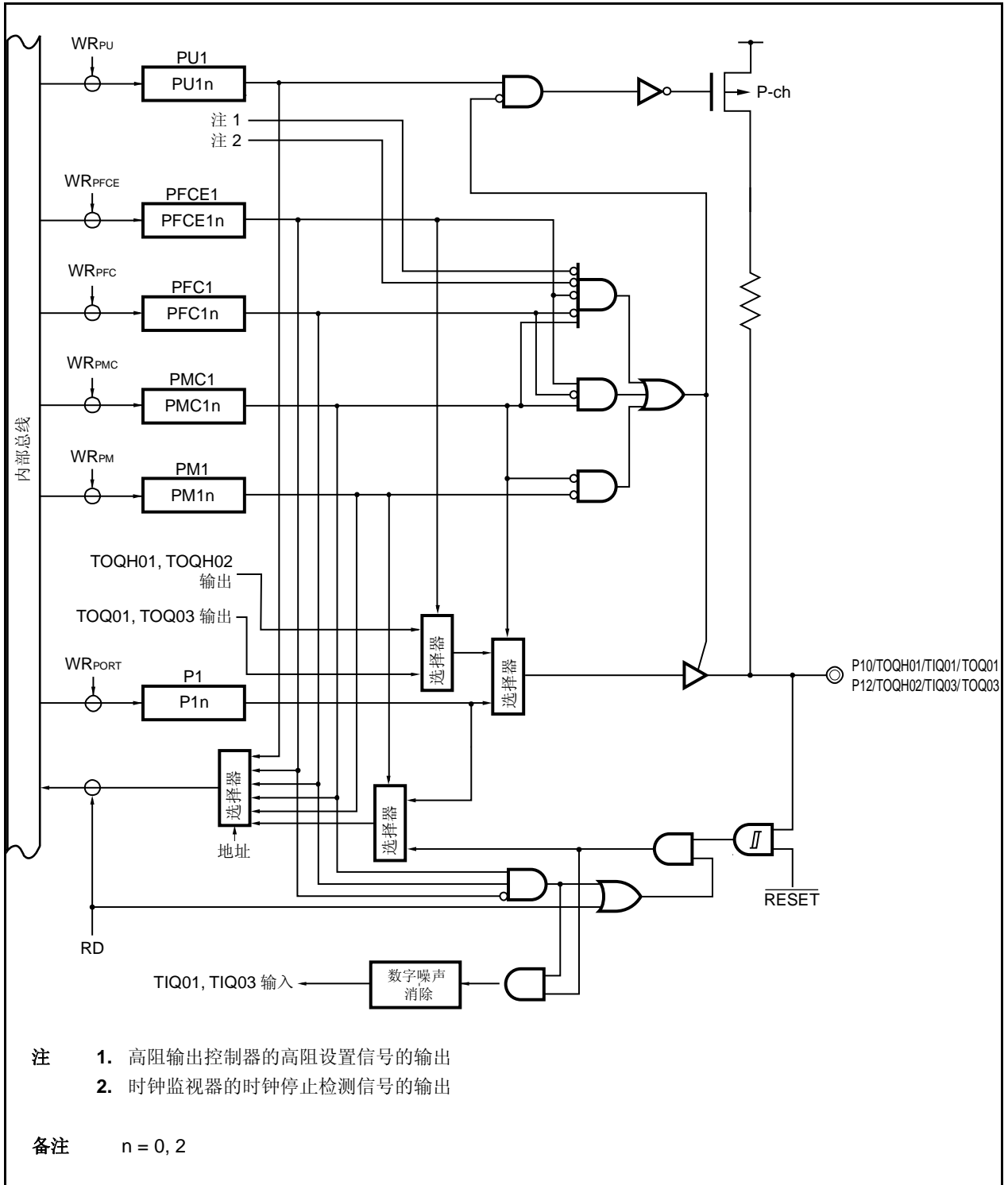


图 4-6. P11 引脚框图

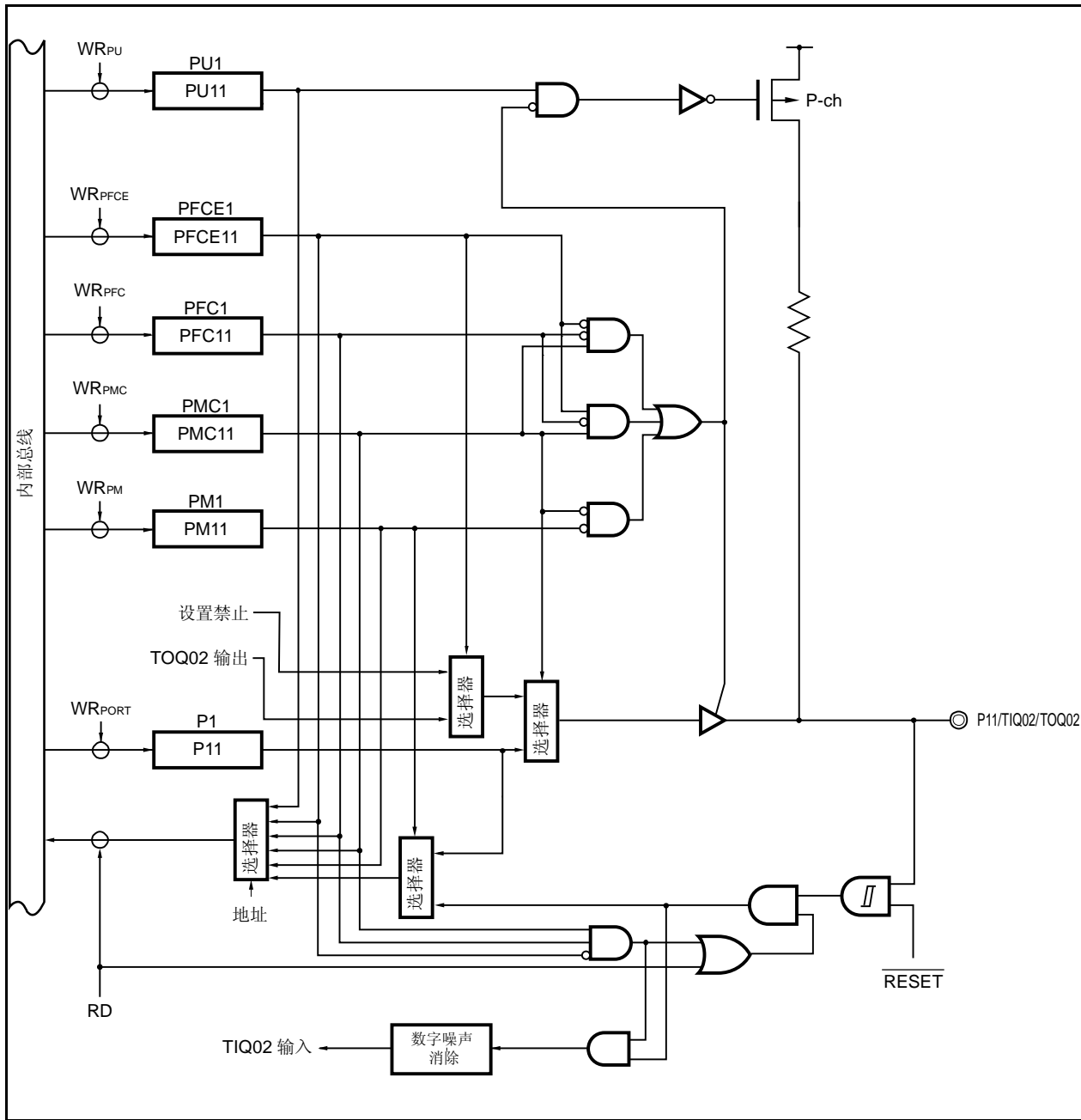


图 4-7. P13 引脚框图

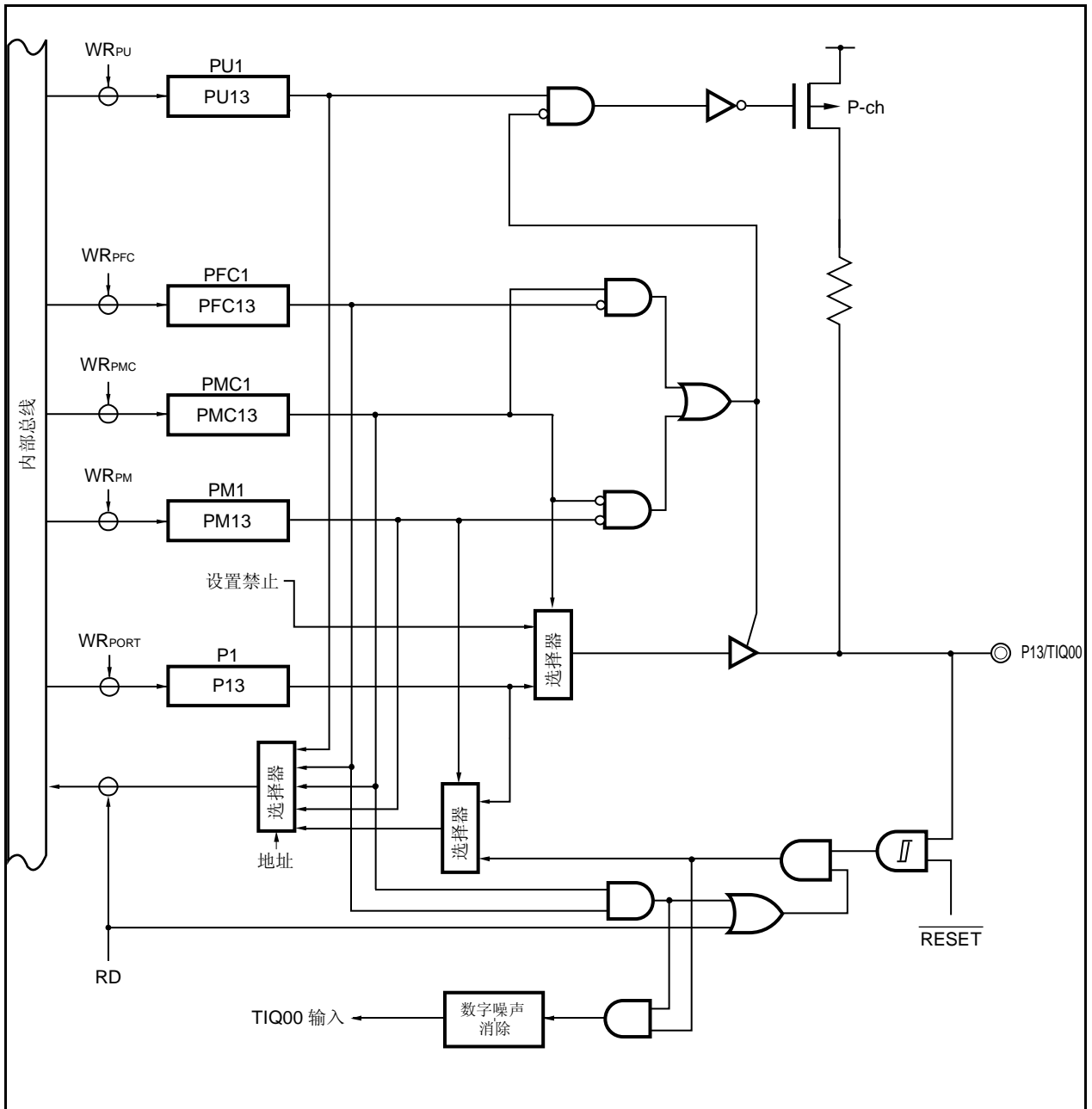


图 4-8. P14 引脚框图

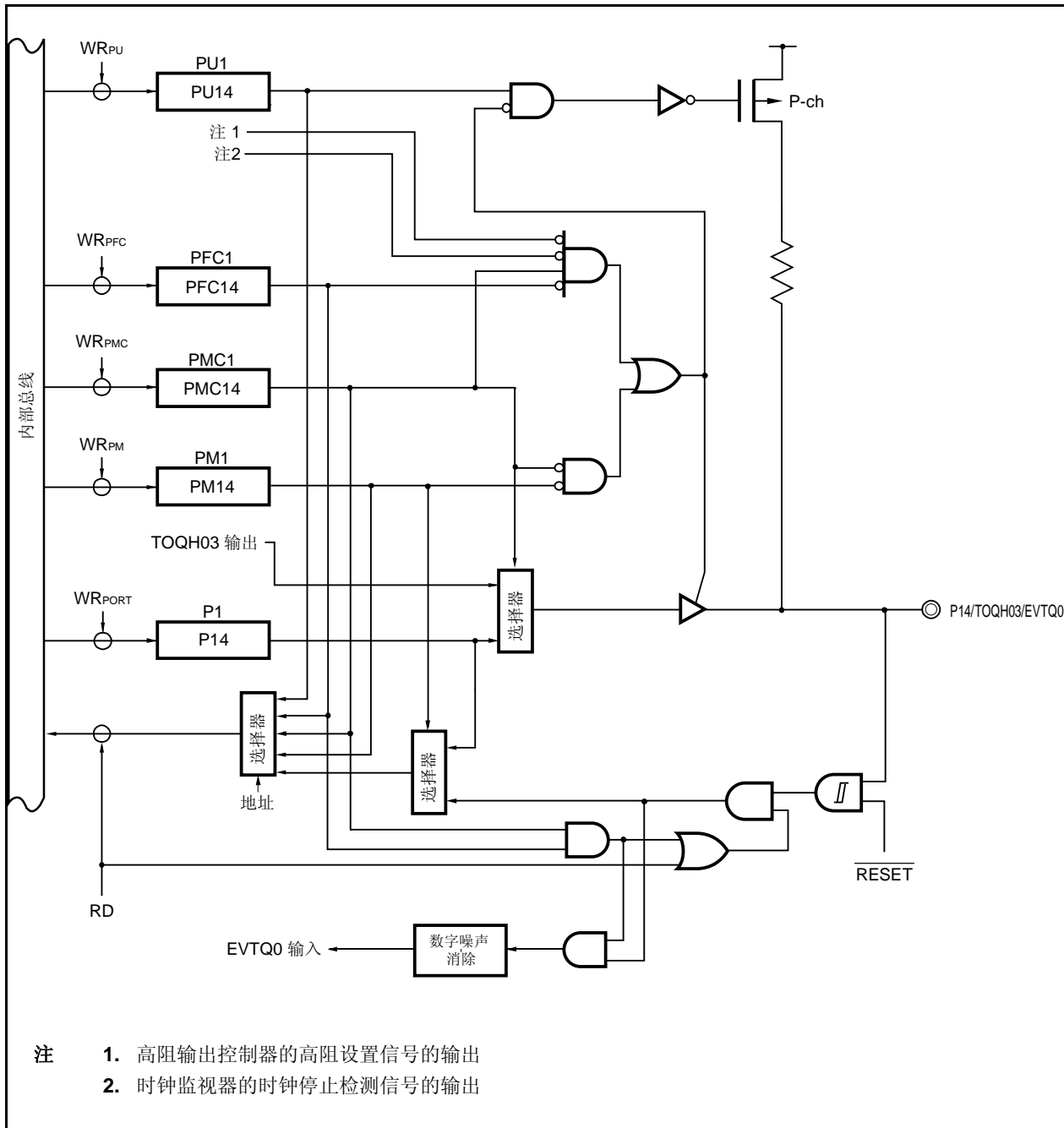


图 4-9. P16 引脚框图

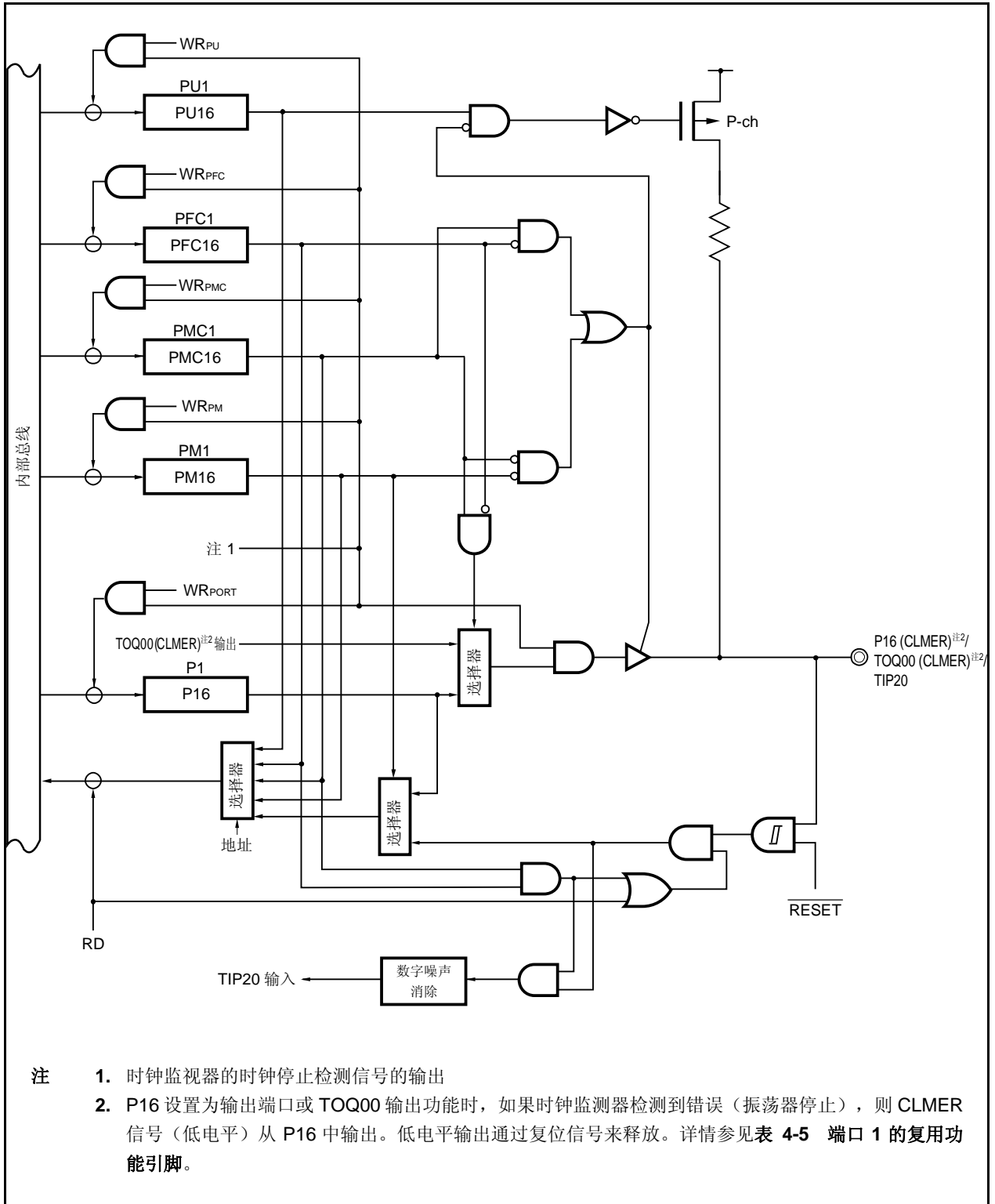
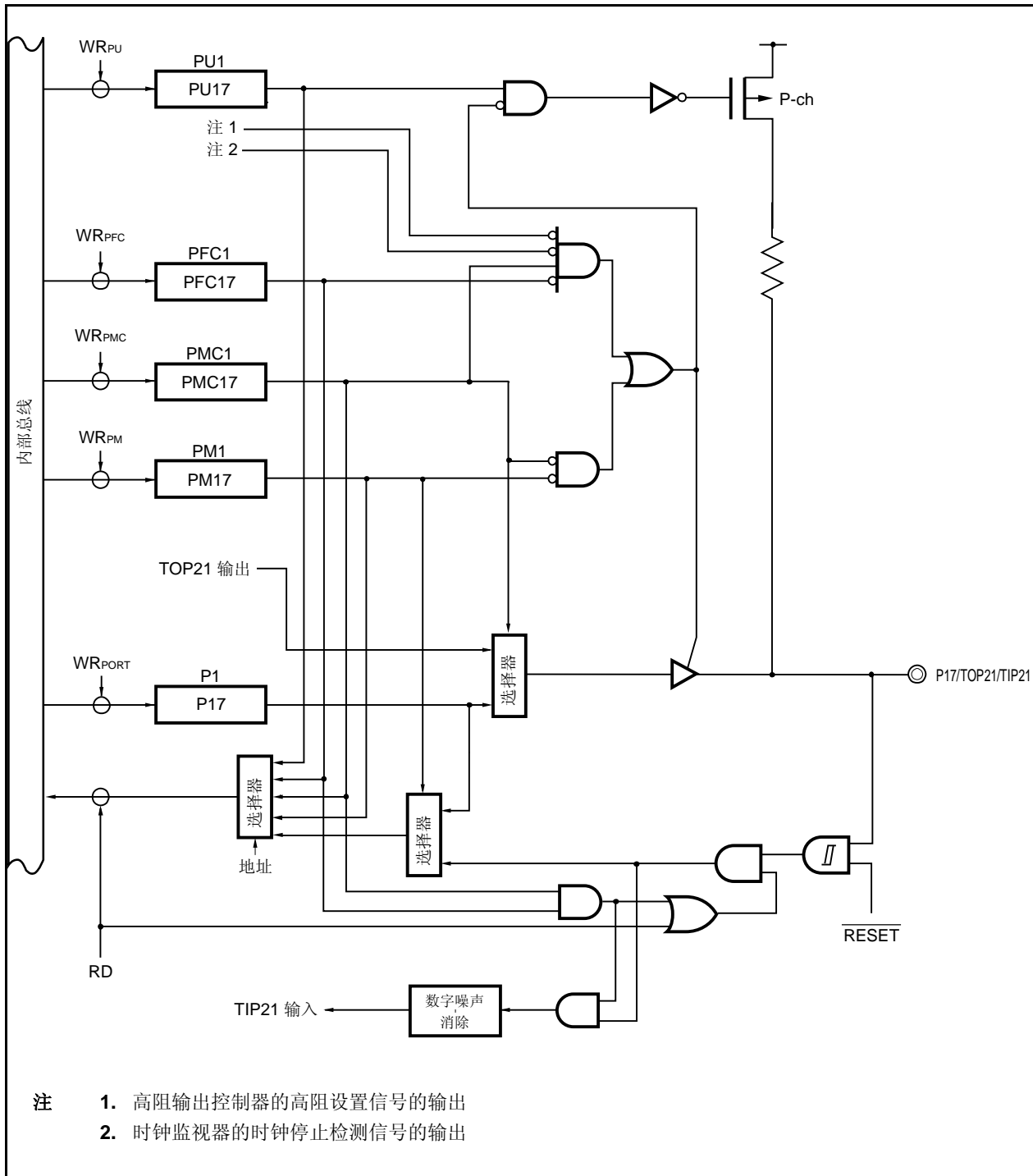


图 4-10. P17 引脚框图



4.3.3 端口 2

端口 2 可用 1 位单元设置为输入或输出模式。

端口 2 具有以下的复用功能引脚：

表 4-6. 端口 2 的复用功能引脚

引脚名称	引脚编号	复用功能引脚	输入/输出	上拉 ^注
P20	54	TOQ1T1	输出	提供
P21	53	TOQ1B1	输出	
P22	52	TOQ1T2	输出	
P23	51	TOQ1B2	输出	
P24	50	TOQ1T3	输出	
P25	49	TOQ1B3	输出	
P26	46	TOQ10	输出	
P27	45	TOP31	输出	

注 软件上拉控制功能

注意事项 当 P20 至 P25 和 P27 用作 TOQ1T1 至 TOQ1T3, TOQ1B1 至 TOQ1B3 以及 TOP31 时, 输出将会在以下信号输出时停止。

- 高阻输出控制器的高阻设置信号的输出
- 时钟监视器的时钟停止检测信号的输出

(1) 寄存器

(a) 端口 2 寄存器 (P2)

复位后: 未定义 R/W 地址: FFFFF404H

	7	6	5	4	3	2	1	0
P2	P27	P26	P25	P24	P23	P22	P21	P20

P2n	输出数据控制 (输出模式下) (n = 0 至 7)
0	输出 0。
1	输出 1。

(b) 端口 2 模式寄存器 (PM2)

复位后: FFH R/W 地址: FFFFF424H

	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

PM2n	输入控制/输出模式 (端口模式下) (n = 0 至 7)
0	输出模式
1	输入模式

(c) 端口 2 模式控制寄存器 (PMC2)

复位后: 00H R/W 地址: FFFFF444H

	7	6	5	4	3	2	1	0
PMC2	PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20

PMC27	P27 引脚的操作模式规范
0	I/O 端口
1	TOP31 输出

PMC26	P26 引脚的操作模式规范
0	I/O 端口
1	TOQ10 输出

PMC25	P25 引脚的操作模式规范
0	I/O 端口
1	TOQ1B3 输出

PMC24	P24 引脚的操作模式规范
0	I/O 端口
1	TOQ1T3 输出

PMC23	P23 引脚的操作模式规范
0	I/O 端口
1	TOQ1B2 输出

PMC22	P22 引脚的操作模式规范
0	I/O 端口
1	TOQ1T2 输出

PMC21	P21 引脚的操作模式规范
0	I/O 端口
1	TOQ1B1 输出

PMC20	P20 引脚的操作模式规范
0	I/O 端口
1	TOQ1T1 输出

(d) 端口 2 上拉电阻选择寄存器 (PU2)

复位后: 00H R/W 地址: FFFFFFFC44H

	7	6	5	4	3	2	1	0
PU2	PU27	PU26	PU25	PU24	PU23	PU22	PU21	PU20

PU2n	片内上拉电阻连接控制 (n = 0 至 7)
0	不连接
1	连接 ^注

注 片内上拉电阻只有在引脚处于端口模式的输入模式时才可以进行连接。此外，当引脚在复用功能模式下由于 TOQH1OFF 和 TOP3OFF 引脚或软件处理而进入高阻状态时，片内上拉电阻将只能连接到 TOQ1T1 至 TOQ1T3、TOQ1B1 至 TOQ1B3 以及 TOP31 引脚上。引脚处于输出状态时不能连接电阻。

(2) 框图

图 4-11. P20 至 P25 以及 P27 引脚的框图

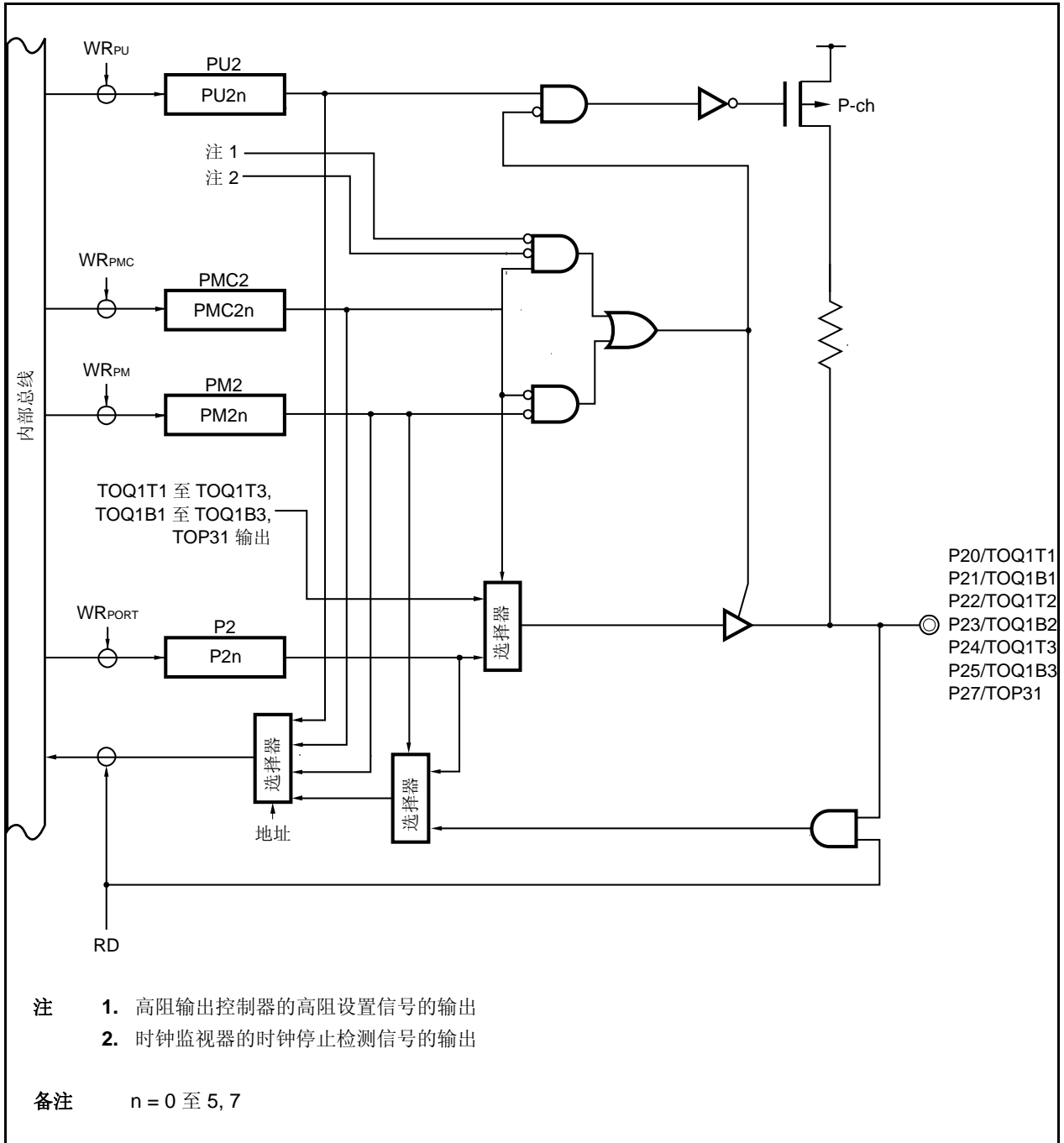
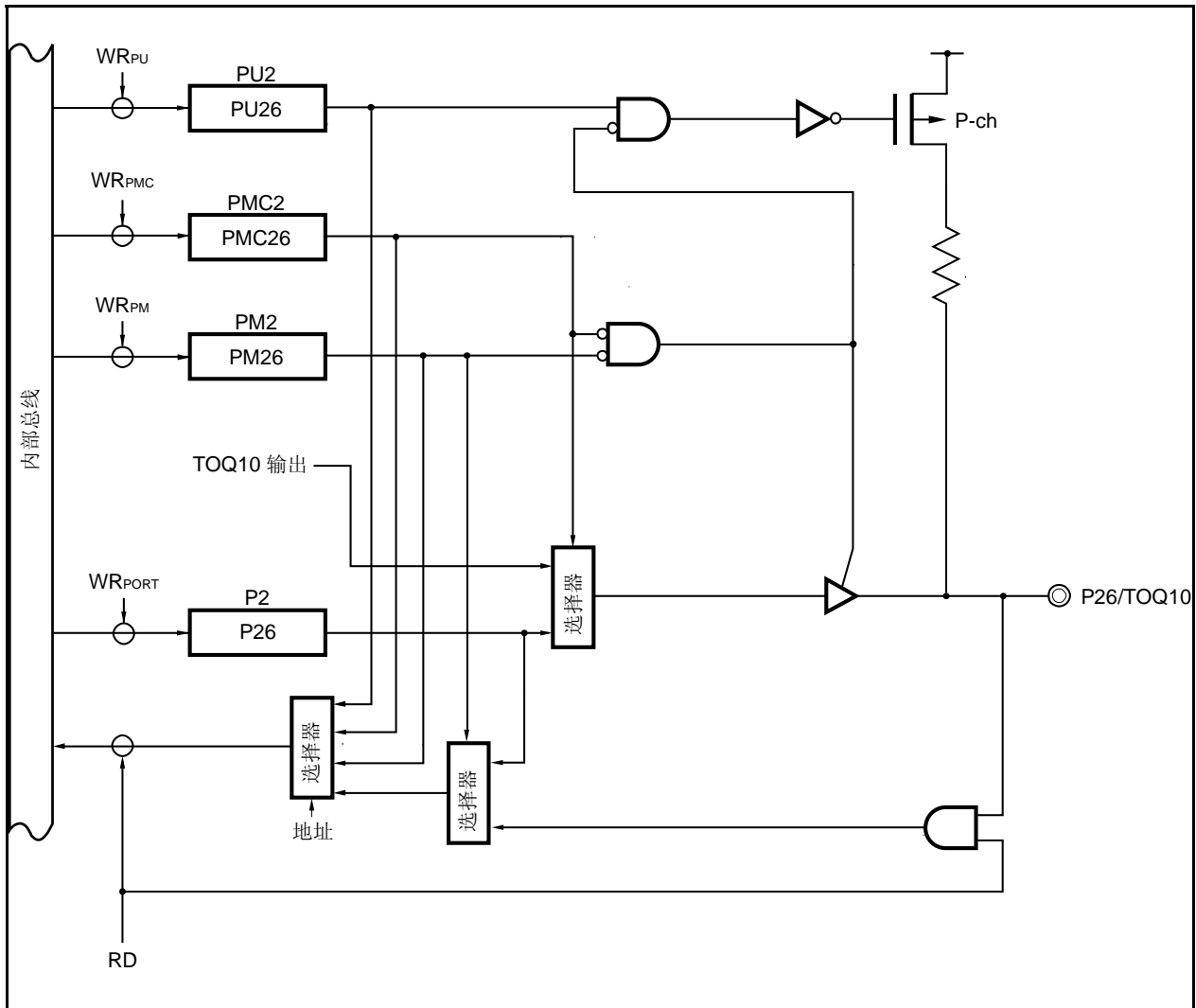


图 4-12. P26 引脚框图



4.3.4 端口 3

端口 3 可用 1 位单元设置为输入或输出模式。

端口 3 具有以下的复用功能引脚：

表 4-7. 端口 3 的复用功能引脚

引脚名称	引脚编号	复用功能引脚	输入/输出	上拉 ^{注1}
P30 ^{注2}	44	RXDA0	输入	提供
P31	43	TXDA0	输出	
P32 ^{注2}	42	RXDA1	输入	
P33	41	TXDA1	输出	

- 注
1. 软件上拉控制功能
 2. 这些引脚在端口模式下读取时用作施密特触发输入。

(1) 寄存器

(a) 端口 3 寄存器 (P3)

复位后: 未定义		R/W	地址: FFFFFFF406H					
	7	6	5	4	3	2	1	0
P3	0	0	0	0	P33	P32	P31	P30
P3n	输出数据控制 (输出模式下) (n = 0 至 3)							
0	输出 0。							
1	输出 1。							

(b) 端口 3 模式寄存器 (PM3)

复位后: FFH		R/W	地址: FFFFFFF426H					
	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30
PM3n	输入控制/输出模式 (端口模式下) (n = 0 至 3)							
0	输出模式							
1	输入模式							

(c) 端口 3 模式控制寄存器(PMC3)

复位后: 00H R/W 地址: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	0	0	0	0	PMC33	PMC32	PMC31	PMC30
PMC33	P33 引脚的操作模式规范							
0	I/O 端口							
1	TXDA1 输出							
PMC32	P32 引脚的操作模式规范							
0	I/O 端口							
1	RXDA1 输入							
PMC31	P31 引脚的操作模式规范							
0	I/O 端口							
1	TXDA0 输出							
PMC30	P30 引脚的操作模式规范							
0	I/O 端口							
1	RXDA0 输入							

(d) 端口 3 功能控制寄存器 (PFC3)

复位后: 00H R/W 地址: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	0	0	0	0	PFC33	PFC32	0	0
PFC33	P33 引脚的复用功能规范							
0	设置禁止							
1	TXDA1 输出							
PFC32	P32 引脚的复用功能规范							
0	设置禁止							
1	RXDA1 输入							

(e) 端口 3 上拉电阻选择寄存器 (PU3)

复位后: 00H R/W 地址: FFFFFFFC46H

	7	6	5	4	3	2	1	0
PU3	0	0	0	0	PU33	PU32	PU31	PU30

PU3n	片内上拉电阻连接控制 (n = 0 至 3)
0	不连接
1	连接 ^注

注 片内上拉电阻只有在引脚处于端口模式的输入模式或处于复用功能模式的输入引脚时才可以进行连接。引脚处于输出状态时不能连接电阻。

(2) 框图

图 4-13. P30 引脚框图

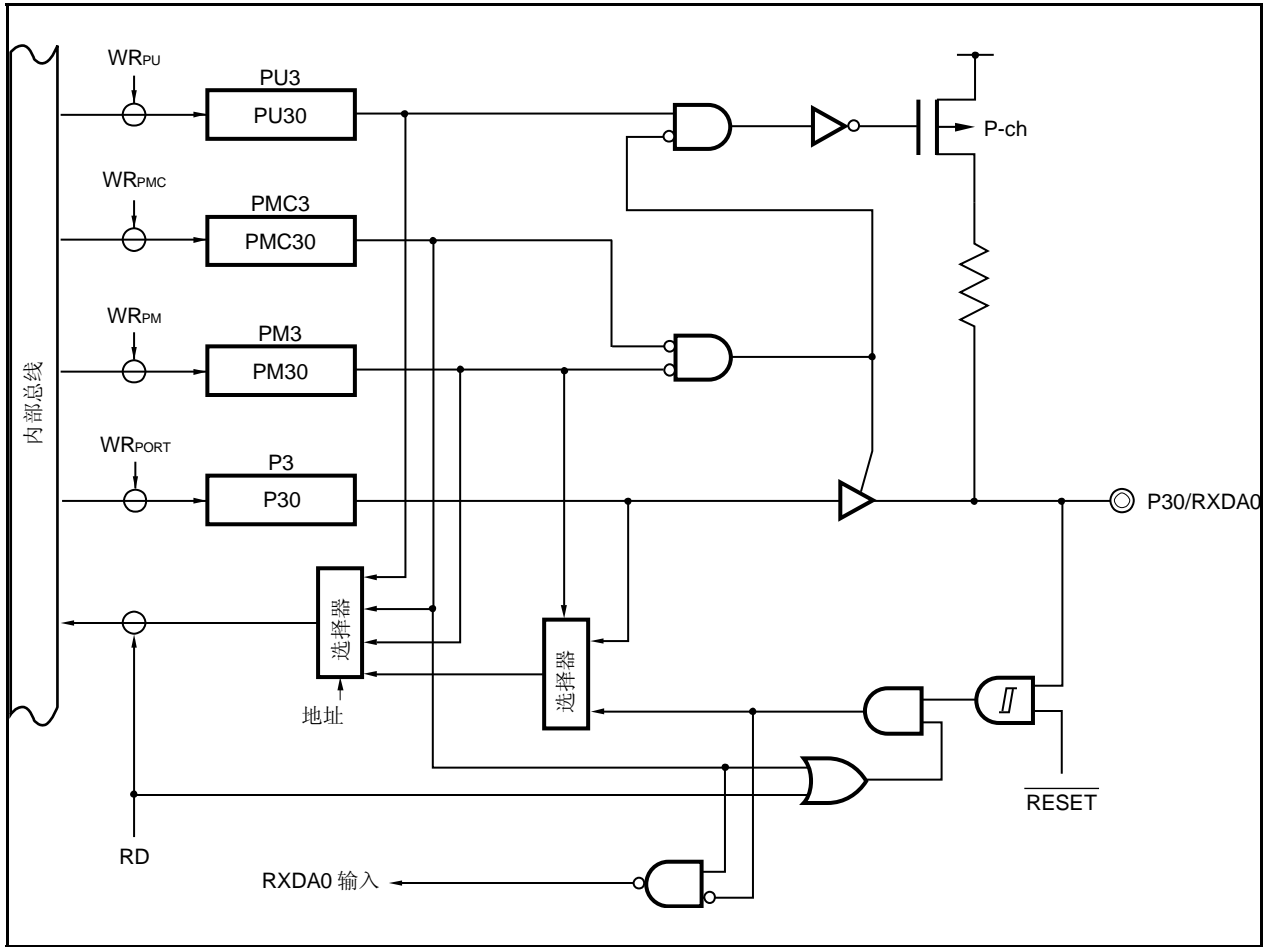


图 4-14. P31 引脚框图

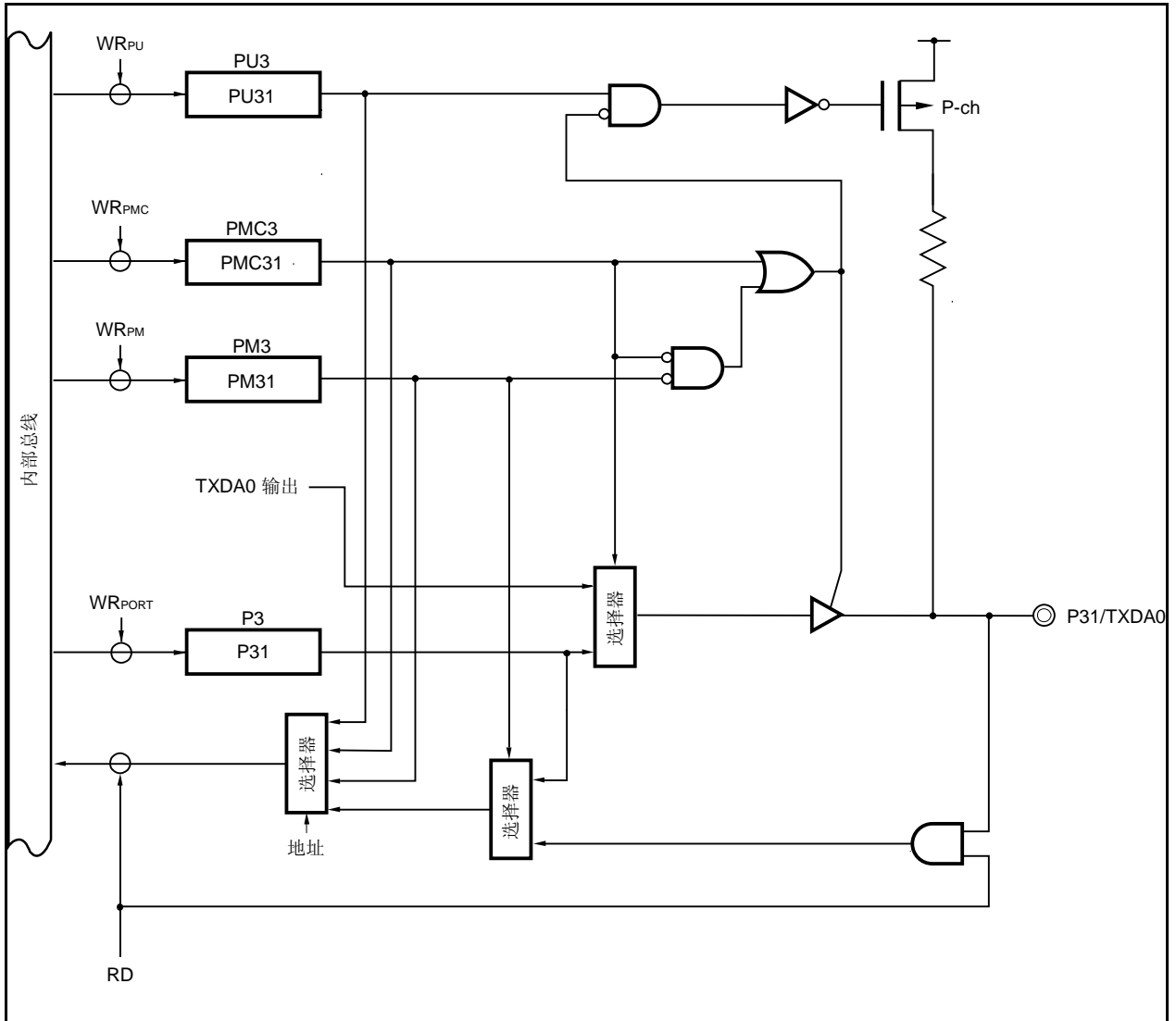


图 4-15. P32 引脚框图

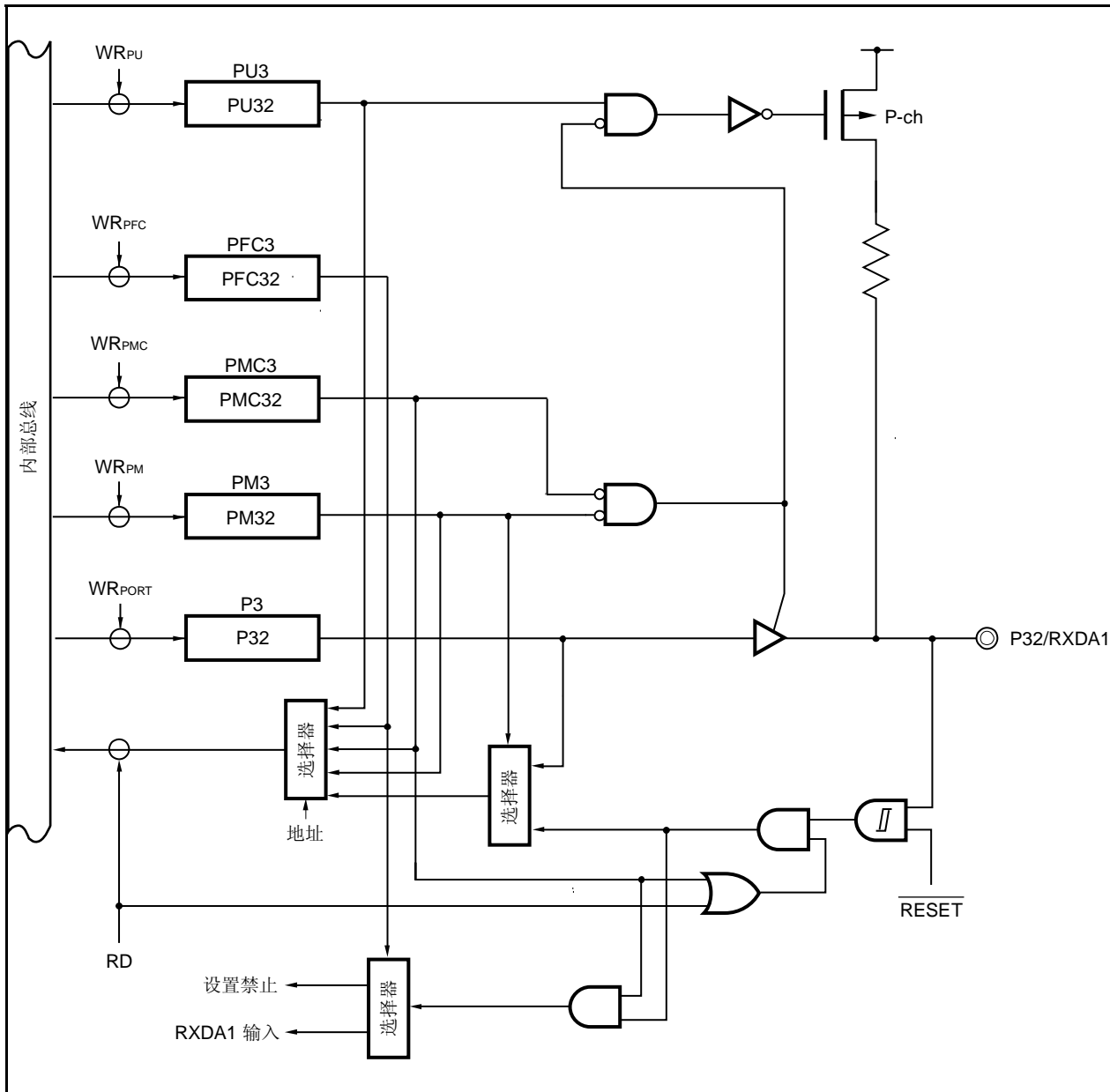
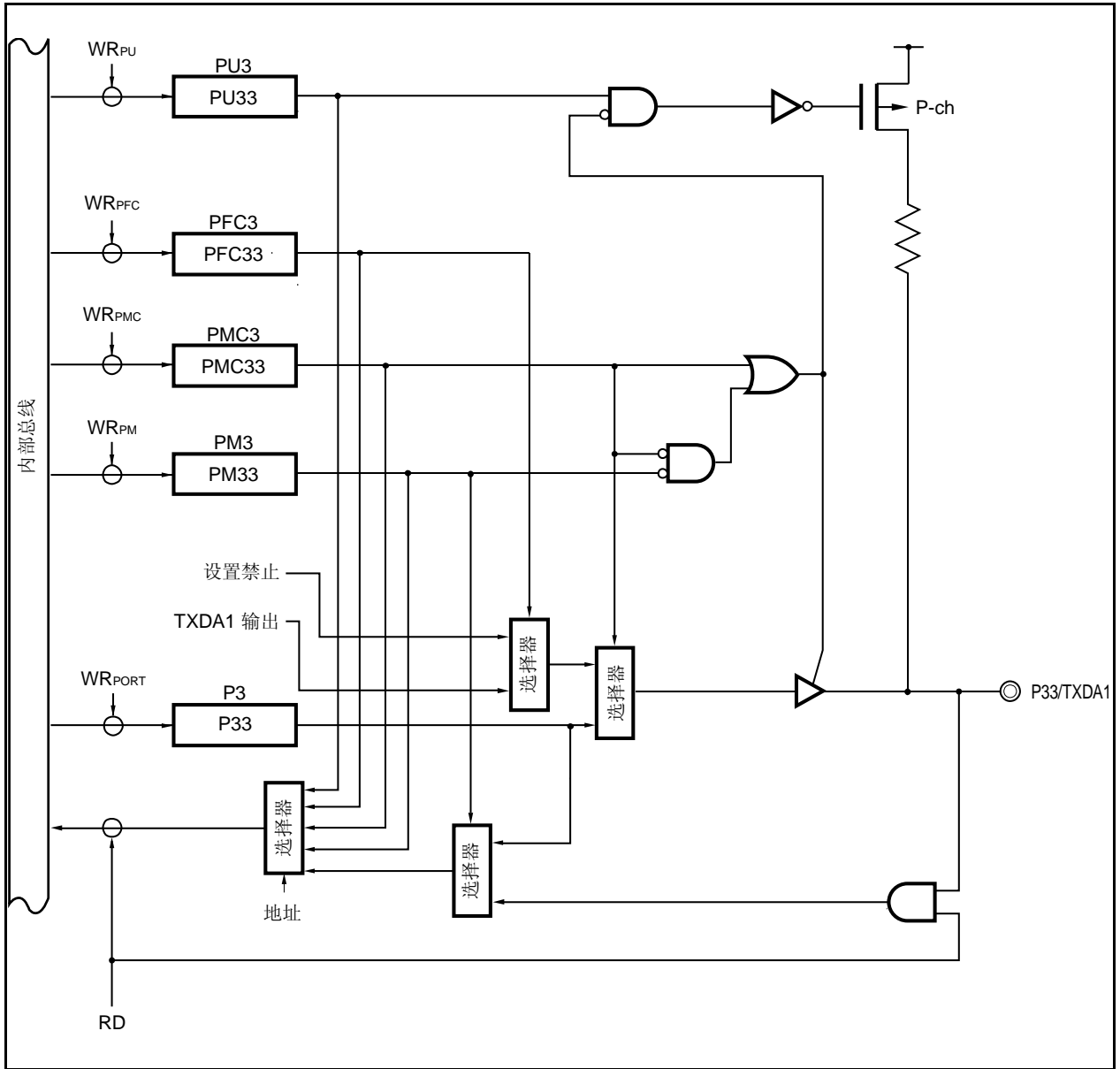


图 4-16. P33 引脚框图



4.3.5 端口 4

端口 4 可用 1 位单元设置为输入或输出模式。

端口 4 具有以下的复用功能引脚：

表 4-8. 端口 4 的复用功能引脚

引脚名称	引脚编号	复用功能引脚	输入/输出	上拉 ^{注1}
P40 ^{注2}	40	SIB0	输入	提供
P41	39	SOB0	输出	
P42 ^{注2}	38	SCKB0	输入/输出	
P43 ^{注2}	37	TOP00/TIP00	输入/输出	
P44 ^{注2}	36	TOP01/TIP01	输入/输出	

- 注
1. 软件上拉控制功能
 2. 这些引脚在端口模式下读取时用作施密特触发输入。

(1) 寄存器

(a) 端口 4 寄存器 (P4)

复位后: 未定义								R/W	地址: FFFFF408H							
	7	6	5	4	3	2	1	0								
P4	0	0	0	P44	P43	P42	P41	P40								
	P4n		输出数据控制 (输出模式下) (n = 0 至 4)													
	0		输出 0。													
	1		输出 1。													

(b) 端口 4 模式寄存器 (PM4)

复位后: FFH								R/W	地址: FFFFF428H							
	7	6	5	4	3	2	1	0								
PM4	1	1	1	PM44	PM43	PM42	PM41	PM40								
	PM4n		输入控制/输出模式 (端口模式下) (n = 0 至 4)													
	0		输出模式													
	1		输入模式													

(c) 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	PMC44	PMC43	PMC42	PMC41	PMC40

PMC44	P44 引脚的操作模式规范
0	I/O 端口
1	TOP01 输出/TIP01 输入

PMC43	P43 引脚的操作模式规范
0	I/O 端口
1	TOP00 输出/TIP00 输入

PMC42	P42 引脚的操作模式规范
0	I/O 端口
1	SCKB0 I/O

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

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

(d) 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	PFC44	PFC43	0	0	0

PFC44	P44 引脚的复用功能规范
0	TOP01 输出
1	TIP01 输入

PFC43	P43 引脚的复用功能规范
0	TOP00 输出
1	TIP00 输入

(e) 端口 4 上拉电阻选择寄存器(PU4)

复位后: 00H R/W 地址: FFFFC48H

	7	6	5	4	3	2	1	0
PU4	0	0	0	PU44	PU43	PU42	PU41	PU40

PU4n	片内上拉电阻连接控制 (n = 0 至 4)
0	不连接
1	连接 ^注

注 片内上拉电阻只有在引脚处于端口模式的输入模式或处于复用功能模式的输入引脚时才可以进行连接（包括处于 SCKB0 引脚从模式时）。引脚处于输出状态时不能连接电阻。

(2) 框图

图 4-17. P40 引脚框图

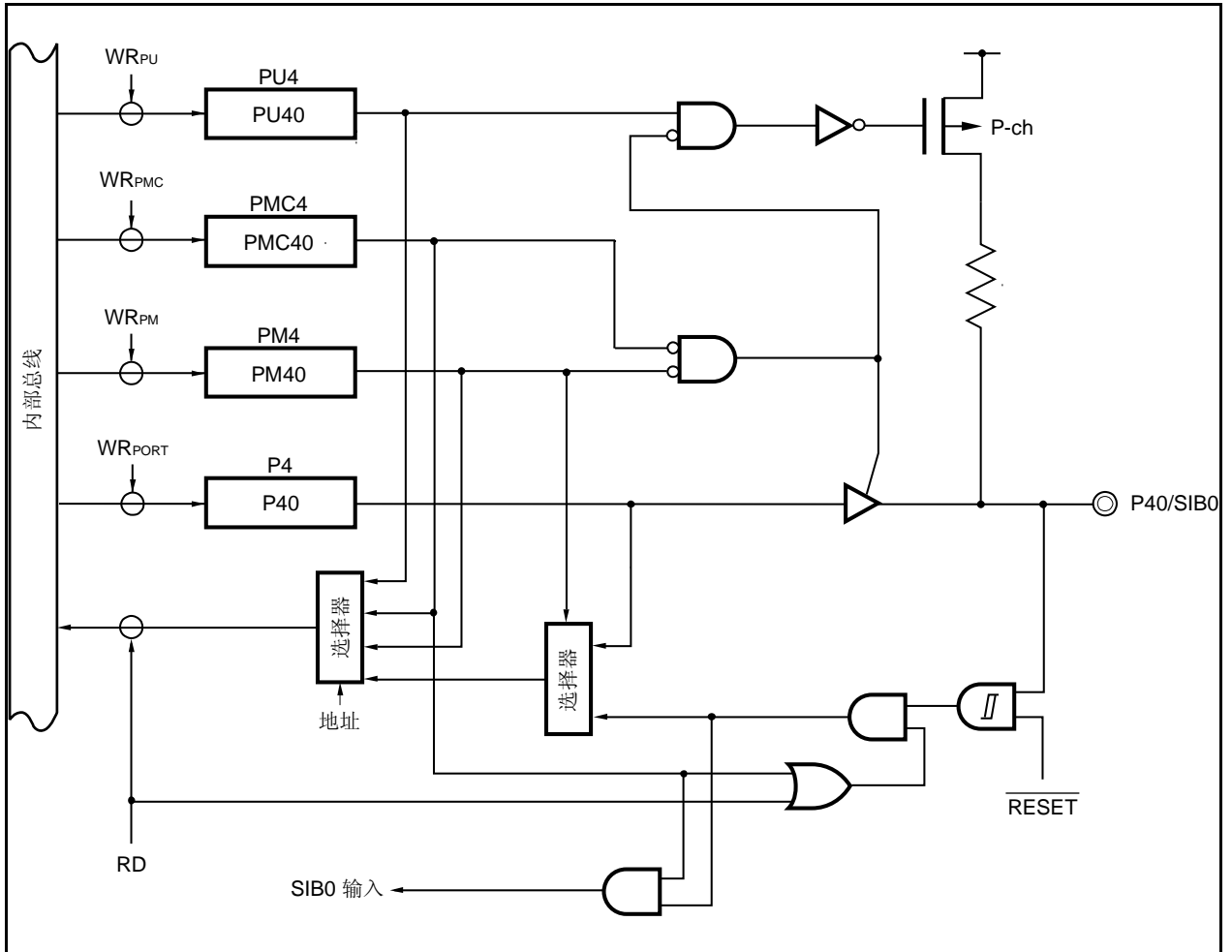


图 4-18. P41 引脚框图

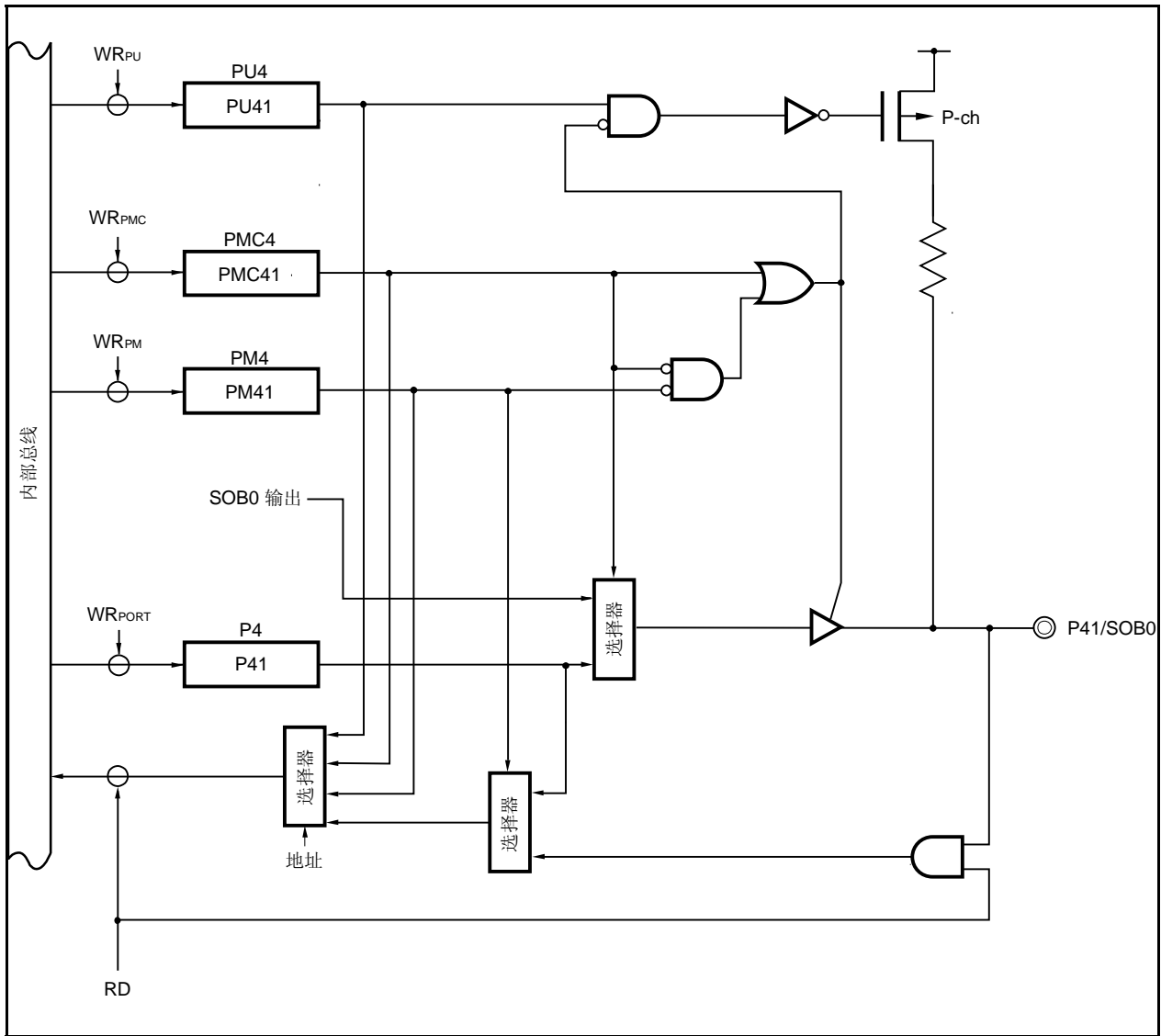


图 4-19. P42 引脚框图

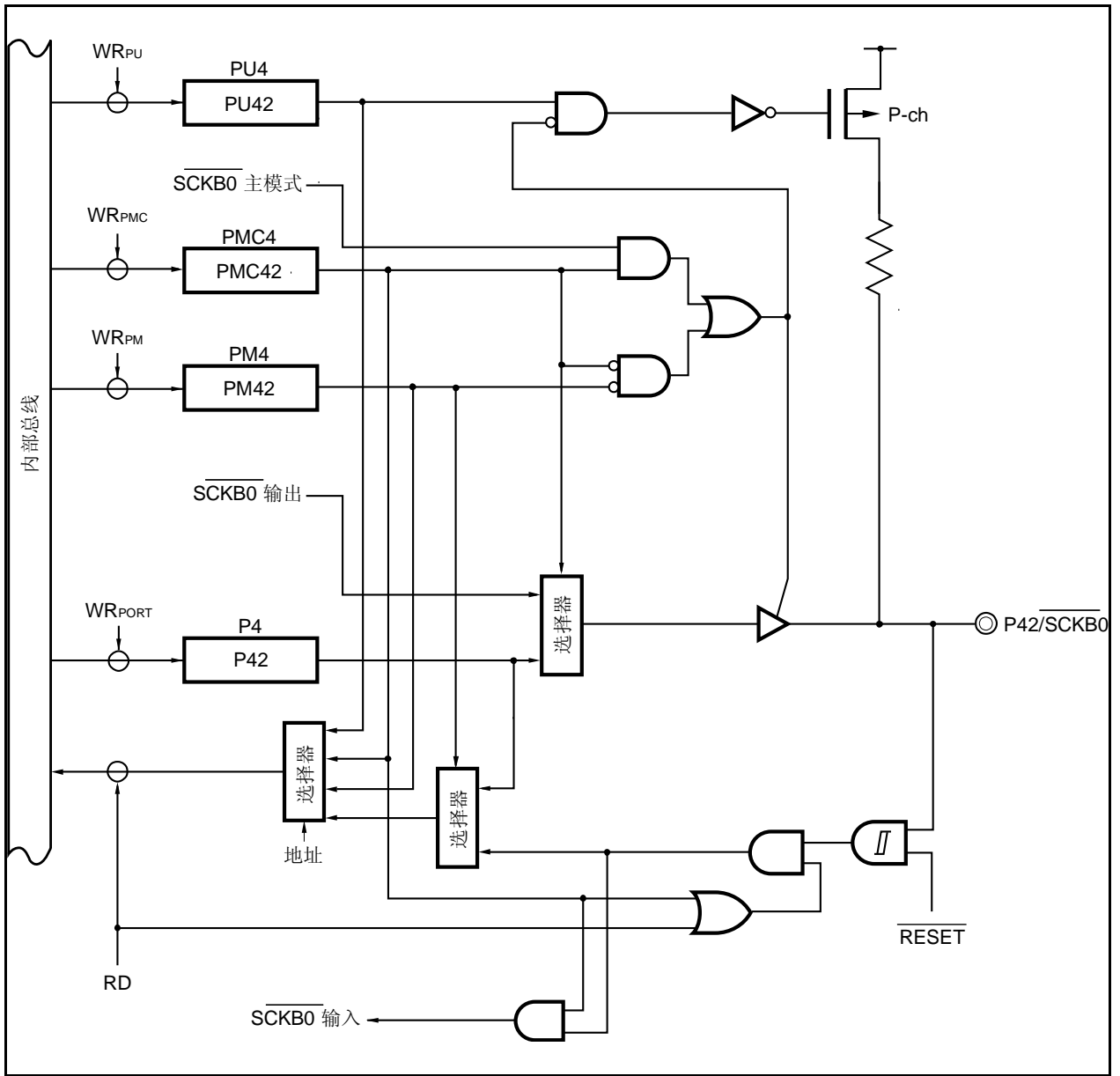
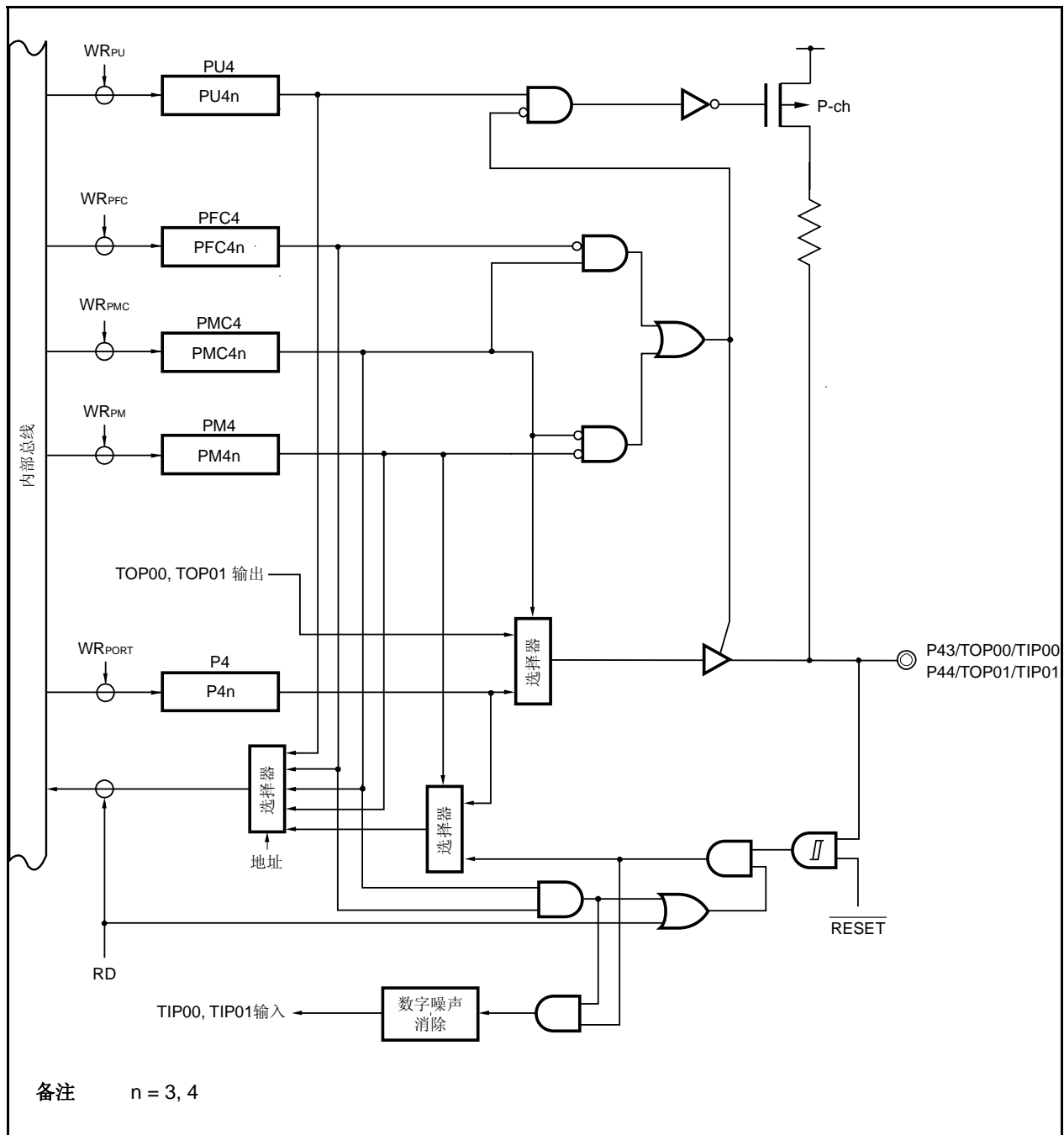


图 4-20. P43 和 P44 引脚的框图



4.3.6 端口DL

端口 DL 可用 1 位单元设置为输入或输出模式。

端口 DL 具有以下的复用功能引脚：

表 4-9. 端口 DL 的复用功能引脚

引脚名称	引脚编号	复用功能引脚	输入/输出	上拉 ^{注1}
PDL0	35	-	-	提供
PDL1	34	-	-	
PDL2	33	-	-	
PDL3	32	-	-	
PDL4	31	-	-	
PDL5	30	FLMD1 ^{注2}	-	
PDL6	29	-	-	
PDL7	28	-	-	

- 注
1. 软件上拉控制功能
 2. 该引脚在 flash 存储器编程模式下使用，因此不需要通过使用端口控制寄存器来操作。细节参见第十八章 flash 存储器。

(1) 寄存器

(a) 端口 DL 寄存器 L (PDLL)

复位后: 未定义 R/W 地址: FFFFF004H

	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 7)
0	输出 0。
1	输出 1。

(b) 端口 DL 模式寄存器 L (PMDLL)

复位后: FFH R/W 地址: FFFFF024H

	7	6	5	4	3	2	1	0
PMDLL	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0

PMDLn	输入控制/输出模式(端口模式下) (n = 0 至 7)
0	输出模式
1	输入模式

(c) 上拉电阻选项寄存器 DLL (PUDLL)

复位后: 00H R/W 地址: FFFFFFF44H

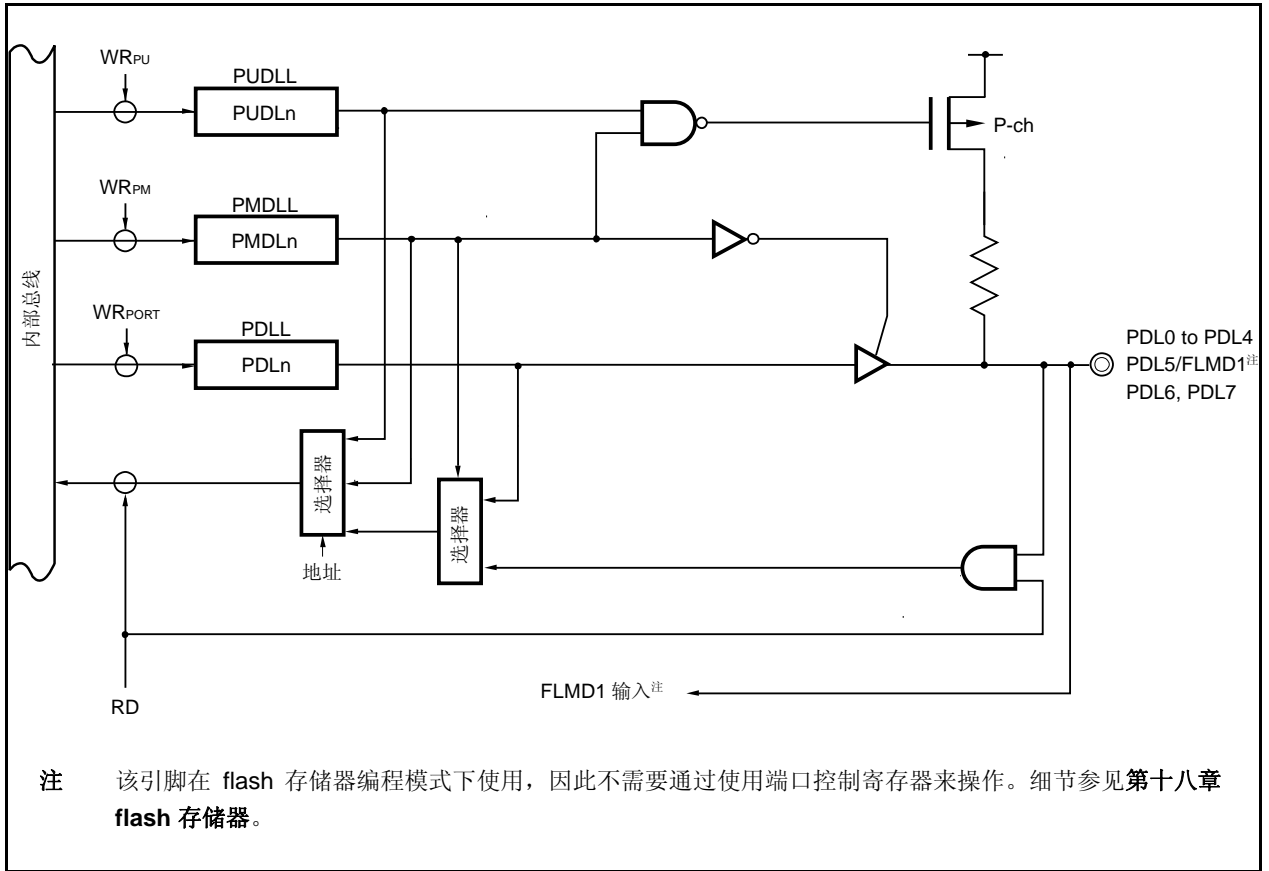
	7	6	5	4	3	2	1	0
PUDLL	PUDL7	PUDL6	PUDL5	PUDL4	PUDL3	PUDL2	PUDL1	PUDL0

PUDLn	片内上拉电阻连接控制 (n = 0 至 7)
0	不连接
1	连接 ^注

注 片内上拉电阻只有在引脚处于端口模式的输入模式时才可以进行连接。引脚处于输出状态时不能连接电阻。

(2) 框图

图 4-21. PDL0 至 PDL7 引脚的框图



4.4 各种设置端口的输出数据和读取值

以下所示为用于选择每个端口设置中各个引脚的复用功能，输出数据以及读取值的值。除了这里所显示设置外，每个外围功能控制寄存器的设置也是必需的。

表 4-10. 各个设置的输出数据和端口读取值 (1/4)

端口名称	功能	PMCmn	PFCEmn	PFCmn	PMmn	输出数据	Pmn 读取值	备注	
P00 至 P06	输出端口	0	无	无	0	端口锁存	端口锁存		
	输入端口				1	-	引脚电平		
	INTP0 至 INTP5 ^注 , INTP6	1	无	无	0	-	端口锁存		需要指定有效沿
					1		引脚电平		
P10, P12	输出端口	0	x	x	0	端口锁存	端口锁存		
	输入端口				1	-	引脚电平		
	TOQH01, TOQH02	1	0	0	0	复用输出（定时 器输出）	端口锁存		
					1		引脚电平		
	TIQ01, TOQ03	1	0	1	0	-	端口锁存	复用输入（定时器输入）	
					1		引脚电平		
	TOQ01, TOQ03	1	1	0	0	复用输出（定时 器输出）	端口锁存		
					1		引脚电平		
P11	输出端口	0	x	x	0	端口锁存	端口锁存		
	输入端口				1	-	引脚电平		
	TIQ02	1	0	1	0	-	端口锁存	复用输入（定时器输入）	
					1		引脚电平		
	TOQ02	1	1	0	0	复用输出（定时 器输出）	端口锁存		
					1		引脚电平		
P13	输出端口	0	无	x	0	端口锁存	端口锁存		
	输入端口				1	-	引脚电平		
	TIQ00	1	无	1	0	-	端口锁存	复用输入（定时器输入）	
					1		引脚电平		

注 包括 TOQH0OFF、TOQ1OFF、TOP2OFF、TOP3OFF、ADTRG0 和 ADTRG1。

备注 x: 无需理会

表 4-10. 各个设置的输出数据和端口读取值 (2/4)

端口名称	功能	PMCmn	PFCEmn	PFCmn	PMmn	输出数据	Pmn 读取值	备注
P14	输出端口	0	无	x	0	端口锁存	端口锁存	
	输入端口				1	-	引脚电平	
	TOQH03	1	无	0	0	复用输出（定时器输出）	端口锁存	
					1		引脚电平	
EVTQ0	1	无	1	0	-	端口锁存	复用输入（定时器输入）	
				1		引脚电平		
P16 (CLMER) [‡] , P17	输出端口 [‡]	0	无	x	0	端口锁存	端口锁存	
	输入端口				1	-	引脚电平	
	TOQ00 (CLMER) [‡] , TOP21	1	无	0	0	复用输出（定时器输出）	端口锁存	
					1		引脚电平	
TIP20, TIP21	1	无	1	0	-	端口锁存	复用输入（定时器输入）	
				1		引脚电平		
P20 至 P27	输出端口	0	无	无	0	端口锁存	端口锁存	
	输入端口				1	-	引脚电平	
	TOQ1T1 至 TOQ1T3, TOQ1B1 至 TOQ1B3, TOQ10, TOP31	1	无	无	0	复用输出（定时器输出）	端口锁存	
1	引脚电平							
P30	输出端口	0	无	无	0	端口锁存	端口锁存	
	输入端口				1	-	引脚电平	
	RXDA0	1	无	无	0	-	端口锁存	
1	引脚电平							
P31	输出端口	0	无	无	0	端口锁存	端口锁存	
	输入端口				1	-	引脚电平	
	TXDA0	1	无	无	0	复用输出（串行输出）	端口锁存	
1					引脚电平			

注 P16 设置为输出端口或 TOQ00 输出功能时，如果时钟监测器检测到错误（振荡器停止），则 CLMER 信号（低电平）从 P16 中输出。低电平输出通过复位信号来释放。详情参见表 4-5 端口 1 的复用功能引脚。

备注 x: 无需理会

表 4-10. 各个设置的输出数据和端口读取值 (3/4)

端口名称	功能	PMCmn	PFCEmn	PFCmn	PMmn	输出数据	Pmn 读取值	备注
P32	输出端口	0	无	x	0	端口锁存	端口锁存	
	输入端口				1	-	引脚电平	
	RXDA1	1	无	1	0	-	端口锁存	
				1		引脚电平		
P33	输出端口	0	无	x	0	端口锁存	端口锁存	
	输入端口				1	-	引脚电平	
	TXDA1	1	无	1	0	复用输出（串行输出）	端口锁存	
				1		引脚电平		
P40	输出端口	0	无	无	0	端口锁存	端口锁存	
	输入端口				1	-	引脚电平	
	SIB0	1	无	无	0	-	端口锁存	
				1		引脚电平		
P41	输出端口	0	无	无	0	端口锁存	端口锁存	
	输入端口				1	-	引脚电平	
	SOB0	1	无	无	0	复用输出（串行输出）	端口锁存	
				1		引脚电平		
P42	输出端口	0	无	无	0	端口锁存	端口锁存	
	输入端口				1	-	引脚电平	
	SCKB0	1	无	无	0	复用输出（串行输出）	端口锁存	
				1		引脚电平		

备注 x: 无需理会

表 4-10. 各个设置的输出数据和端口读取值 (4/4)

端口名称	功能	PMCmn	PFCEmn	PFCmn	PMmn	输出数据	Pmn 读取值	备注
P43, P44	输出端口	0	无	x	0	端口锁存	端口锁存	
	输入端口				1	-	引脚电平	
	TOP00, TOP01	1	无	0	0	复用输出 (定时器输出)	端口锁存	
					1		引脚电平	
TIP00, TIP01	1	无	1	0	-	端口锁存	复用输入 (定时器输入)	
				1		引脚电平		
PDL0 至 PDL4, PDL5 ^注 , PDL6, PDL7	输出端口	无	无	无	0	端口锁存	端口锁存	
	输入端口				1	-	引脚电平	

注 PDL5 引脚也可以用作在 flash 编程模式中进行设置的引脚。使用端口控制寄存器时该引脚不需要进行操作。细节参见第十八章 flash 存储器。

备注 x: 无需理会

4.5 使用复用功能时的端口寄存器设置

以下显示各个端口用作复用功能时的端口寄存器设置。当端口引脚用作复用功能引脚时，请参考各引脚的说明。

表 4-11. 端口引脚用作复用功能引脚 (1/3)

引脚名称	复用引脚		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	INTP0	输入	P00 = 不需要设置	PM00 = 不需要设置	PMC00 = 1	-	-	
	TOQH0OFF	输入	P00 = 不需要设置	PM00 = 不需要设置	PMC00 = 1	-	-	
P01	INTP1	输入	P01 = 不需要设置	PM01 = 不需要设置	PMC01 = 1	-	-	
	TOQ1OFF	输入	P01 = 不需要设置	PM01 = 不需要设置	PMC01 = 1	-	-	
P02	INTP2	输入	P02 = 不需要设置	PM02 = 不需要设置	PMC02 = 1	-	-	
	TOQ2OFF	输入	P02 = 不需要设置	PM02 = 不需要设置	PMC02 = 1	-	-	
P03	INTP3	输入	P03 = 不需要设置	PM03 = 不需要设置	PMC03 = 1	-	-	
	TOQ3OFF	输入	P03 = 不需要设置	PM03 = 不需要设置	PMC03 = 1	-	-	
P04	INTP4	输入	P04 = 不需要设置	PM04 = 不需要设置	PMC04 = 1	-	-	
	ADTRG0	输入	P04 = 不需要设置	PM04 = 不需要设置	PMC04 = 1	-	-	
P05	INTP5	输入	P05 = 不需要设置	PM05 = 不需要设置	PMC05 = 1	-	-	
	ADTRG1	输入	P05 = 不需要设置	PM05 = 不需要设置	PMC05 = 1	-	-	
P06	INTP6	输入	P06 = 不需要设置	PM06 = 不需要设置	PMC06 = 1	-	-	
P10	TOQH01	输出	P10 = 不需要设置	PM10 = 不需要设置	PMC10 = 1	PFCE10 = 0	PFC10 = 0	
	TIQ01	输入	P10 = 不需要设置	PM10 = 不需要设置	PMC10 = 1	PFCE10 = 0	PFC10 = 1	
	TOQ01	输出	P10 = 不需要设置	PM10 = 不需要设置	PMC10 = 1	PFCE10 = 1	PFC10 = 0	
P11	TIQ02	输入	P11 = 不需要设置	PM11 = 不需要设置	PMC11 = 1	PFCE11 = 0	PFC11 = 1	
	TOQ02	输出	P11 = 不需要设置	PM11 = 不需要设置	PMC11 = 1	PFCE11 = 1	PFC11 = 0	
P12	TOQH02	输出	P12 = 不需要设置	PM12 = 不需要设置	PMC12 = 1	PFCE12 = 0	PFC12 = 0	
	TIQ03	输入	P12 = 不需要设置	PM12 = 不需要设置	PMC12 = 1	PFCE12 = 0	PFC12 = 1	
	TOQ03	输出	P12 = 不需要设置	PM12 = 不需要设置	PMC12 = 1	PFCE12 = 1	PFC12 = 0	
P13	TIQ00	输入	P13 = 不需要设置	PM13 = 不需要设置	PMC13 = 1	-	PFC13 = 1	
P14	TOQH03	输出	P14 = 不需要设置	PM14 = 不需要设置	PMC14 = 1	-	PFC14 = 0	
	EVTQ0	输入	P14 = 不需要设置	PM14 = 不需要设置	PMC14 = 1	-	PFC14 = 1	

表 4-11. 端口引脚用作复用功能引脚 (2/3)

引脚名称	复用引脚		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						
P16 (CLMER) [‡]	TOQ00 (CLMER) [‡]	输出	P16 = 不需要设置	PM16 = 不需要设置	PMC16 = 1	-	PFC16 = 0	
	TIP20	输入	P16 = 不需要设置	PM16 = 不需要设置	PMC16 = 1	-	PFC16 = 1	
P17	TOP21	输出	P17 = 不需要设置	PM17 = 不需要设置	PMC17 = 1	-	PFC17 = 0	
	TIP21	输入	P17 = 不需要设置	PM17 = 不需要设置	PMC17 = 1	-	PFC17 = 1	
P20	TOQ1T1	输出	P20 = 不需要设置	PM20 = 不需要设置	PMC20 = 1	-	-	
P21	TOQ1B1	输出	P21 = 不需要设置	PM21 = 不需要设置	PMC21 = 1	-	-	
P22	TOQ1T2	输出	P22 = 不需要设置	PM22 = 不需要设置	PMC22 = 1	-	-	
P23	TOQ1B2	输出	P23 = 不需要设置	PM23 = 不需要设置	PMC23 = 1	-	-	
P24	TOQ1T3	输出	P24 = 不需要设置	PM24 = 不需要设置	PMC24 = 1	-	-	
P25	TOQ1B3	输出	P25 = 不需要设置	PM25 = 不需要设置	PMC25 = 1	-	-	
P26	TOQ10	输出	P26 = 不需要设置	PM26 = 不需要设置	PMC26 = 1	-	-	
P27	TOP31	输出	P27 = 不需要设置	PM27 = 不需要设置	PMC27 = 1	-	-	
P30	RXDA0	输入	P30 = 不需要设置	PM30 = 不需要设置	PMC30 = 1	-	-	
P31	TXDA0	输出	P31 = 不需要设置	PM31 = 不需要设置	PMC31 = 1	-	-	
P32	RXDA1	输入	P32 = 不需要设置	PM32 = 不需要设置	PMC32 = 1	-	PFC32 = 1	
P33	TXDA1	输出	P33 = 不需要设置	PM33 = 不需要设置	PMC33 = 1	-	PFC33 = 1	
P40	SIB0	输入	P40 = 不需要设置	PM40 = 不需要设置	PMC40 = 1	-	-	
P41	SOB0	输出	P41 = 不需要设置	PM41 = 不需要设置	PMC41 = 1	-	-	
P42	$\overline{\text{SCKB0}}$	I/O	P42 = 不需要设置	PM42 = 不需要设置	PMC42 = 1	-	-	

注 P16 设置为输出端口或 TOQ00 输出功能时，如果时钟监测器检测到错误（振荡器停止），则 CLMER 信号（低电平）从 P16 中输出。低电平输出通过复位信号来释放。详情参见表 4-5 端口 1 的复用功能引脚。

表 4-11. 端口引脚用作复用功能引脚 (3/3)

引脚名称	复用引脚		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						
P43	TOP00	输出	P43 = 不需要设置	PM43 = 不需要设置	PMC43 = 1	-	PFC43 = 0	
	TIP00	输入	P43 = 不需要设置	PM43 = 不需要设置	PMC43 = 1	-	PFC43 = 1	
P44	TOP01	输出	P44 = 不需要设置	PM44 = 不需要设置	PMC44 = 1	-	PFC44 = 0	
	TIP01	输入	P44 = 不需要设置	PM44 = 不需要设置	PMC44 = 1	-	PFC44 = 1	
PDL0	-	-	PDL0 = 不需要设置	PMDL0 = 不需要设置	-	-	-	
PDL1	-	-	PDL1 = 不需要设置	PMDL1 = 不需要设置	-	-	-	
PDL2	-	-	PDL2 = 不需要设置	PMDL2 = 不需要设置	-	-	-	
PDL3	-	-	PDL3 = 不需要设置	PMDL3 = 不需要设置	-	-	-	
PDL4	-	-	PDL4 = 不需要设置	PMDL4 = 不需要设置	-	-	-	
PDL5 ^注	FLMD1 ^注	输入	PDL5 = 不需要设置	PMDL5 = 不需要设置	-	-	-	
PDL6	-	-	PDL6 = 不需要设置	PMDL6 = 不需要设置	-	-	-	
PDL7	-	-	PDL7 = 不需要设置	PMDL7 = 不需要设置	-	-	-	

注 PDL5 引脚也可以用作在 flash 编程模式中进行设置的引脚 (FLMD1)。使用端口控制寄存器时该引脚不需要进行操作。细节参见第十八章 flash 存储器。

4.6 噪声消除器

为以下引脚提供了用于确保噪声消除时间的控制器。在噪声消除时间内改变的输入信号不会在内部响应。

<R>

表 4-12. 噪声消除器

设备	目标引脚	延迟类型	噪声消除宽度	采样时钟
复位	RESET	模拟延迟	几个 10 ns (TYP.)	-
模式引脚	FLMD0			
<ul style="list-style-type: none"> • 中断 (INTC)^注 • 马达控制定时器的高阻抗输出控制功能 • A/D 转换器 (ADC) 	INTP0/TOQH0OFF INTP1/TOQ1OFF INTP2/TOP2OFF INTP3/TOP3OFF INTP4/ADTRG0 INTP5/ADTRG1	数字延迟	500 ns (MIN.)	
	INTP6		4 至 5 个时钟 (400 ns (20 MHz 处)) (800 ns (20 MHz 处)) (1.6 μ s (20 MHz 处)) (3.2 μ s (20 MHz 处))	
定时器 Q (TMQ)	TIQ00 TIQ01 TIQ02 TIQ03 EVTQ0	数字延迟	4 至 5 个时钟 (200 ns (20 MHz 处))	f _{xx} (50 ns (20 MHz 处))
定时器 P (TMP)	TIP00 TIP01 TIP20 TIP21			

注 不同于 INTP6 的可屏蔽中断输入可用作 IDLE 或 STOP 模式的释放源。

注意事项 1. 可屏蔽中断引脚用于释放待机模式。

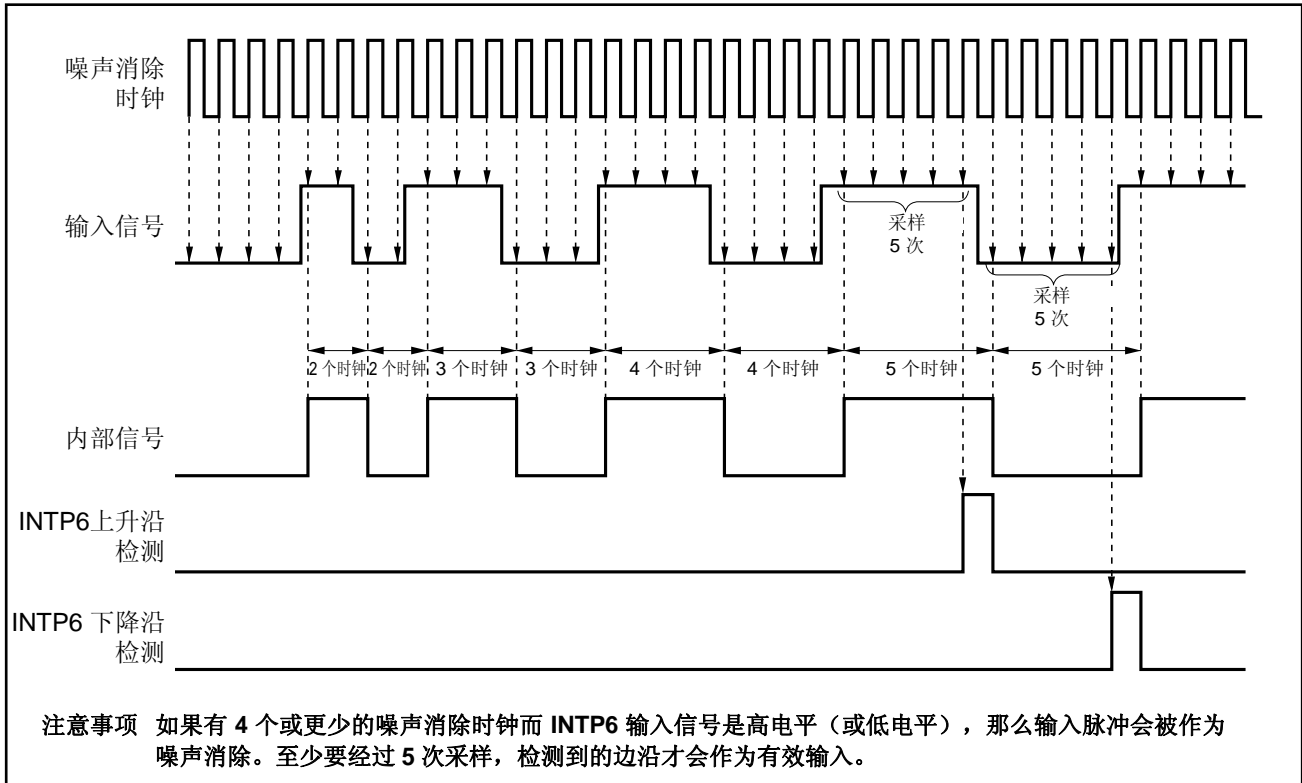
2. 数字延迟引脚的噪声过滤器使用时钟采样，因此当外围时钟 (f_{xx}) 停止时输入信号将不能响应。

3. 噪声消除器只能在使用复用功能时运行。

以下是 INTP6 引脚的数字噪声消除时序举例。

<R>

图 4-22 噪声消除时序举例



<R>

(1) 外部中断噪声消除控制寄存器(INTPNRC)

INTPNRC 寄存器用于选择 INTP6 引脚消除数字噪声的采样时钟。如果同一电平不能连续检测到 5 次，则信号会被作为噪声消除。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

- 注意事项**
1. 如果输入脉冲持续在 4 至 5 个时钟期间，则脉冲被检测为有效沿还是作为噪声消除变得不确定。因此要脉冲实际检测为有效沿，同一脉冲电平输入必须持续 5 个或更长的时钟。
 2. 如果噪声和采样时钟同步产生，则通过给输入引脚加入过滤器除此噪声。
 3. 如果引脚用作普通输入端口引脚则噪声不能被消除。

复位后: 00H		R/W	地址: FFFFF310H					
	7	6	5	4	3	2	1	0
INTPNRC	0	0	0	0	0	0	INTPNRC1	INTPNRC0
	INTPNRC1	INTPNRC0	选择采样时钟					
	0	0	f _{xx} /16					
	0	1	f _{xx} /8					
	1	0	f _{xx} /4					
	1	1	f _{xx} /2					

4.7 注意事项**4.7.1 设置端口引脚的注意事项**

- (1) 按以下步骤设置端口寄存器：

- <1> 设置 和 寄存器。
- <2> 设置 PMCN 寄存器。
- <3> 设置 INTFn 和 INTRn 寄存器。

如果在设置 PMCN 和 PFCEn 寄存器之前设置 PFCn 寄存器，则 PFCn 和 PFCEn 寄存器设置过程中可能会选择一个未预料的外围功能。

- (2) 片内上拉电阻只有在引脚处于端口模式的输入模式或处于复用功能模式的输入引脚时才可以进行连接。此外，当引脚由于 TOQ1OFF、TOP2OFF、TOP3OFF 或 TOQH0OFF 引脚或软件处理而进入高阻状态时，片内上拉电阻可连接到 TOP21、TOQ1T1 至 TOQ1T3、TOQ1B1 至 TOQ1B3、TOP31 以及 TOQH01 至 TOQH03 引脚上。

4.7.2 端口寄存器 (Pn) 位操作指令注意事项

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，还可能对不属于操作对象的输入端口的输出锁存器写入不需要的值。

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。

<举例说明> 当 P20 为一个输出端口，同时，P21 到 P27 作为输入端口（所有引脚状态为高电平），而端口锁存器的值为 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，则端口锁存的值为 FFH。

解释说明：PMnm 位为 1 的端口中 Pn 寄存器的写/读对象是各自的输出锁存和引脚状态。

V850ES/IE2 单片机按以下次序执行位操作指令：

<1> 寄存器 Pn 按 8 位读取。

<2> 操作目标位。

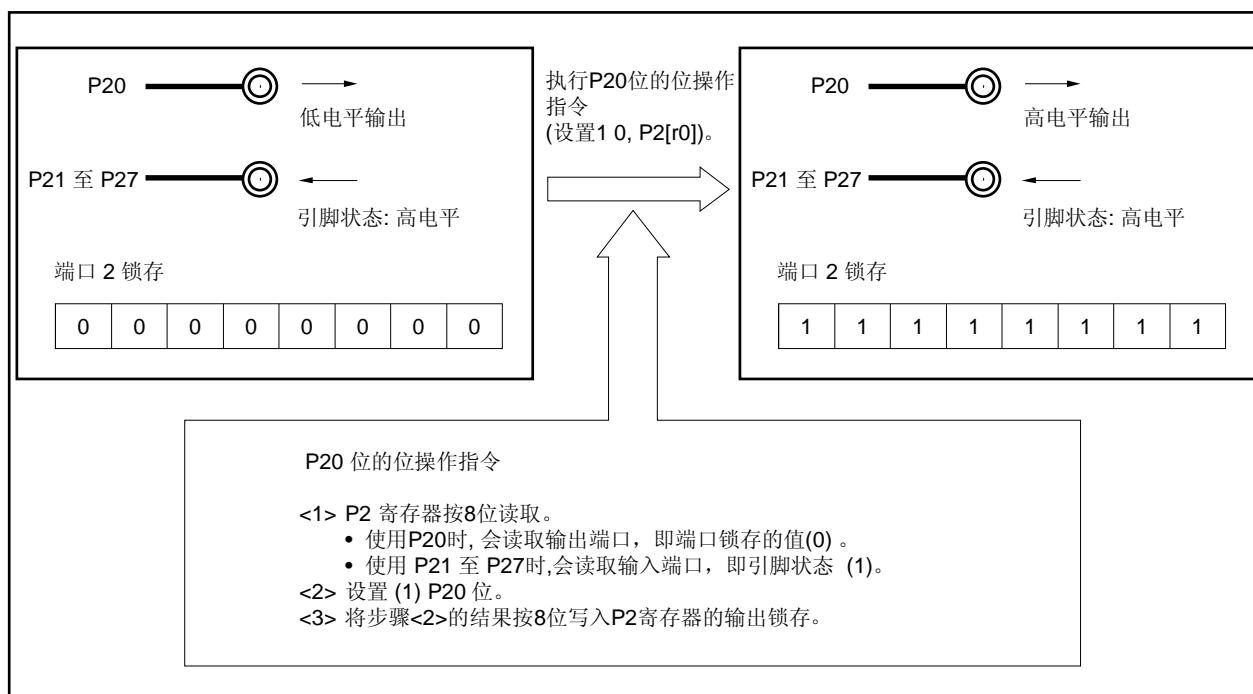
<3> 按 8 位写入寄存器 Pn。

在步骤 <1>中，当作为输入端口的 P21 至 P27 引脚状态被读取时，作为输出端口的 P20 引脚的输出锁存（0）的值被读取。如果此时 P21 至 P27 引脚的状态为高电平，则读取的值为 FEH。

在步骤 <2>时，位操作过程使之变为 FFH。

在步骤 <3>时，位操作将 FFH 写入输出锁存器。

图 4-23. 位操作指令 (P20 引脚)



第五章 时钟发生器

5.1 概述

时钟发生器的特点如下：

- 振荡器
 - PLL 模式中： $f_x = 2.5 \text{ MHz}$ ($f_{xx} = 20 \text{ MHz}$)
 - 时钟直通模式中： $f_x = 2.5 \text{ MHz}$ ($f_{xx} = 2.5 \text{ MHz}$)
- PLL（锁相环）倍频（固定×8）功能
 - 时钟直通模式/PLL 模式选择。
- 内部系统时钟产生
 - 4 种 (f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/8$)
- 外围时钟产生功能。
- 振荡稳定时间选择。

备注 f_x : 振荡频率
 f_{xx} : 系统时钟频率

5.2 配置

图 5-1. 时钟发生器

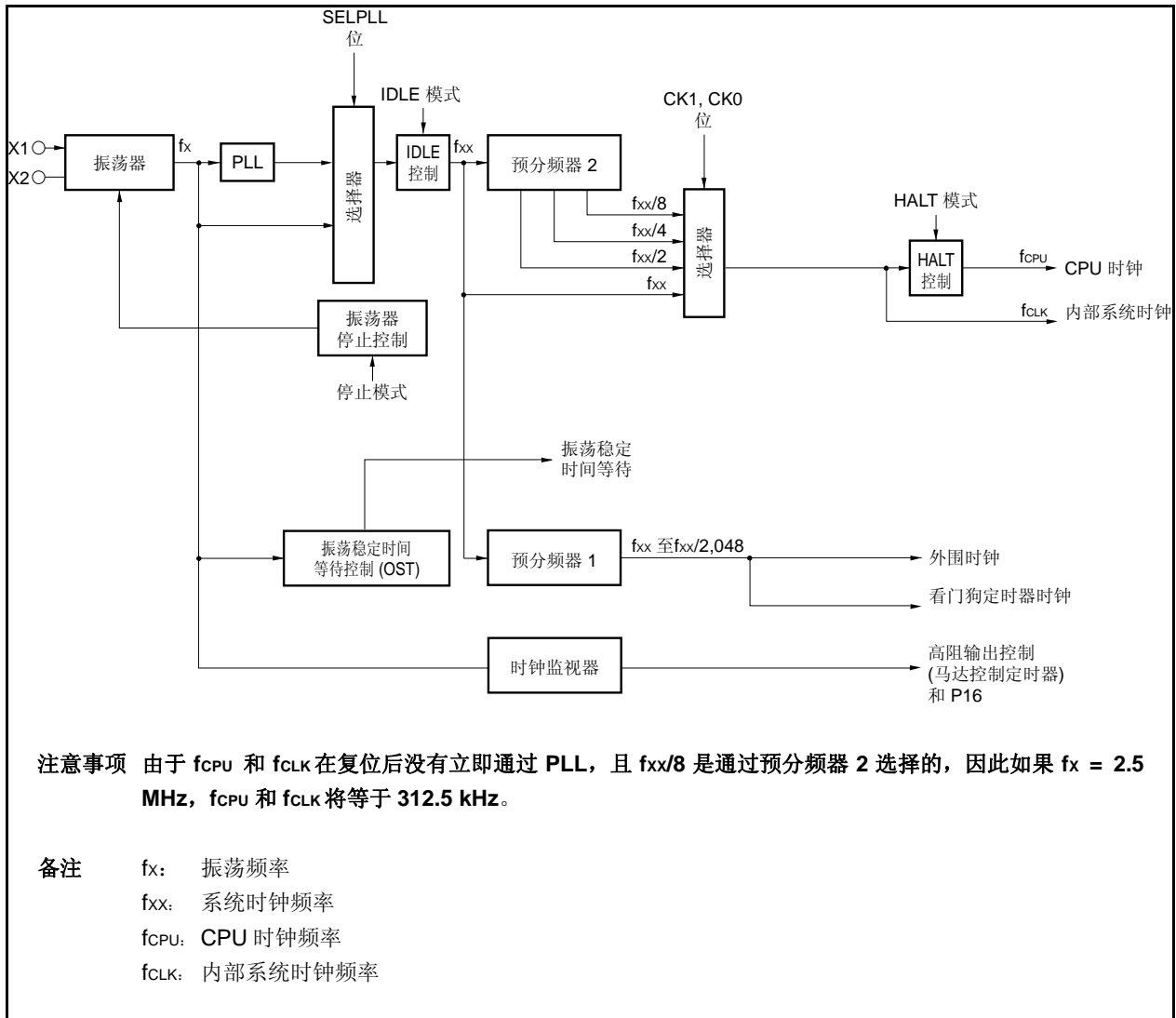


表 5-1. 各功能模块的操作时钟

功能模块	操作时钟
CPU	f_{CPU} (通过 PCC 寄存器从 f_{xx} 至 $f_{xx}/8$ 中选择)
中断控制器	f_{CLK} (通过 PCC 寄存器从 f_{xx} 至 $f_{xx}/8$ 中选择)
定时器 (不包括看门狗定时器)	f_{xx}
看门狗定时器	$f_{xx}/512$
UARTA	f_{UCLK} (通过 UAnCTL1 寄存器从 f_{xx} 至 $f_{xx}/2,048$ 中选择)
CSIB	f_{CCLK} (通过 CB0CTL1 寄存器从 $f_{xx}/2$ 至 $f_{xx}/128$ 中选择)
A/D 转换器 0, 1	f_{xx}

备注

1. 外围时钟
2. $n = 0, 1$

(1) 振荡器

主振荡器产生以下振荡频率 (f_x) :

- PLL 模式中 (固定 $\times 8$) : $f_x = 2.5 \text{ MHz}$ ($f_{xx} = 20 \text{ MHz}$)
- 时钟直通模式中: $f_x = 2.5 \text{ MHz}$ ($f_{xx} = 2.5 \text{ MHz}$)

(2) IDLE 控制

除从模式下振荡器、PLL、时钟监视器操作和 CSIB 外其它所有功能都被停止。

(3) HALT 控制

只有 CPU 时钟 (f_{CPU}) 被停止。

(4) PLL

本电路对振荡器产生的时钟 (f_x) 进行 8 倍频。

可在两种模式下进行工作: 时钟直通模式, 在这种模式中 f_x 按 PLL 控制寄存器 (PLLCTL) 中 SELPLL 位的设置来输出; 以及 PLL 模式, 在这种模式中会输出一个倍频时钟。

在 PLL 模式中, PLL 的输出频率为 20 MHz。

(5) 预分频器 1

该预分频器产生供给片上周边元件的时钟 (f_{xx} 至 $f_{xx}/2,048$)。

(6) 预分频器 2

该电路对系统时钟 (f_{xx}) 进行分频。

生成供给 CPU 时钟 (f_{CPU}) 和内部系统时钟 (f_{CLK}) 的时钟 (f_{xx} 至 $f_{xx}/8$)。

(7) 振荡稳定时间等待控制 (OST)

该器件测量从振荡器产生的时钟输入开始直到振荡稳定所需的时间。它也计算 PLL 锁定时间。计数时钟可以在 $2^{14}/f_x$ 至 $2^{16}/f_x$ 中选择。

(8) 时钟监视器

时钟监视器通过使用内部振荡时钟来采样由振荡器 (f_x) 产生的时钟。当它检测到错误 (振荡停止) 时, 马达控制定时器的输出将会进入高阻状态。CLMER 信号 (低电平) 会从 P16 中输出 (详细信息请参见第四章 端口功能和第九章 马达控制功能)。低电平输出通过复位信号来释放。

注 CLMER 信号 (低电平) 只有在 P16 被指定为输出端口或 TOQ00 输出功能时有效。

5.3 控制寄存器

时钟发生器由以下七种寄存器来控制：

- 锁相环控制寄存器（PLLCTL）
- 时钟控制寄存器（CKC）
- 处理器时钟控制寄存器（PCC）
- 节电控制寄存器（PSC）
- 节电模式寄存器（PSMR）
- 振荡稳定时间选择寄存器（OSTS）
- 时钟监视器模式寄存器（CLM）

(1) 锁相环控制寄存器（PLLCTL）

PLLCTL 寄存器选择 CPU 操作时钟。

该寄存器可进行 8 位或 1 位读写。

系统复位后本寄存器被设置为 01H。

复位后: 01H	R/W	地址: FFFFF82CH						
	7	6	5	4	3	2	<1>	0
PLLCTL	0	0	0	0	0	0	SELPLL	1
	CPU操作时钟选择							
	SELPLL							
	0	时钟直连模式						
	1	PLL 模式						

- 注意事项**
1. 确保将位 7 至 2 清除为“0”，并将位 0 设置为“1”。
 2. 只有在 PLL 时钟频率稳定后，SELPLL 位才能够被设置为 1。如果在 PLL 时钟频率不稳定（未锁期间）时重写 SELPLL 位，那么该位会写入 0。因此，一定要确定已经设置 PLL 模式。
使用以下程序作为参考：

```
_loop:   set1    1, PLLCTL
        tst1    1, PLLCTL
        bz     _loop
```

（下个指令）

3. 使用 PLL 模式（SELPLL 位= 1）前，务必将 CKC 寄存器设置为 0BH。除非 CKC 寄存器已经被设置，否则不能保证操作。

(2) 时钟控制寄存器 (CKC)

CKC 寄存器用于控制 PLL 模式。使用 PLL 模式 (PLLCTL.SELPLL 位= 1) 前, 务必将 CKC 寄存器设置为 0BH。除非 CKC 寄存器已经被设置, 否则不能保证操作。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)。

该寄存器可按 8 位进行读写。

系统复位后该寄存器被设为 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	0/1	0/1

- 注意事项**
1. CKC 寄存器的默认值为 0AH。然而, 务必在进行初始设置期间将该寄存器设置为 0BH。在线仿真器的默认值未定义, 但在使用在线仿真器时也应将寄存器设置为 0BH。
 2. 写入 CKC 寄存器时, 需使用命令寄存器 PRCMD。

<R>

(3) 处理器时钟控制寄存器 (PCC)

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

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被置为 03H。

复位后: 03H R/W 地址: FFFFF828H

	7	6	5	4	3	2	1	0
PCC	0	0	0	0	0	0	CK1	CK0

CK1	CK0	时钟选择 (fCLK/fCPU)
0	0	fxx
0	1	fxx/2
1	0	fxx/4
1	1	fxx/8

- 注意事项**
1. 一定要将位 2 至 7 设置为“0”。
 2. 选择 PLL 模式后 (PLLCTL.SELPLL 位 = 1) 再对 PCC 寄存器进行设置。
 3. 写入 PCC 寄存器时, 需使用命令寄存器 PRCMD。

<R>

(4) 节电控制寄存器 (PSC)

PSC 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）。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

复位后: 00H R/W 地址: FFFFF1FEH

	7	6	5	<4>	3	2	<1>	0
PSC	0	0	0	INTM	0	0	STB	0

INTM	通过可屏蔽中断请求(INTxx ^注)控制等待模式
0	通过允许的INTxx请求释放待命模式
1	通过禁止的INTxx请求释放待命模式

STB	操作模式设置
0	正常模式
1	待机模式

注 详细信息请参见表 14-1 中断源列表。

注意事项 1. 一定要将位 0、2、3 以及 5 至 7 清为“0”。

2. 在通过将 STB 位设为 1 来设置待机模式前，务必先将 PCC 寄存器设为 03H，然后将 STB 位设为 1。否则，待机模式可能不被设置或者释放。释放待机模式后，将 PCC 寄存器的值改为期望值。

3. 要设置为 IDLE 模式或 STOP 模式，则应按顺序先将 PCC 寄存器设为 03H，再设置 PSMR.PSM0 位，然后将 STB 位设为 1。

4. 写入 PSC 寄存器时，需使用命令寄存器 PRCMD。

<R>

(5) 节电模式寄存器 (PSMR)

PSMR 寄存器是控制软件待机模式中操作的 8 位寄存器。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 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	0	PSM0

PSM0	软件待机模式中的操作规范
0	IDLE 模式
1	STOP 模式

- 注意事项**
1. 一定要将位 1 至 7 设置为“0”。
 2. 仅当 PSC.STB 位为 1 时，PSM0 位才有效。

(6) 振荡稳定时间选择寄存器 (OSTS)

通过中断请求释放 STOP 模式后，OSTS 寄存器将会选择振荡稳定时间直到振荡稳定为止。

该寄存器可按 8 位进行读写。

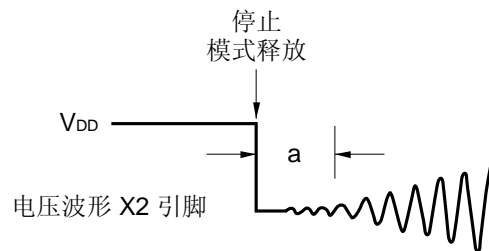
复位设置将寄存器清为 06H。

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	振荡稳定时间的选择 ($f_x = 2.5 \text{ MHz}$)
1	0	0	$2^{14}/f_x$ (6.55 ms)
1	0	1	$2^{15}/f_x$ (13.1 ms)
1	1	0	$2^{16}/f_x$ (26.2 ms)
其他情况			禁止

注意事项 1. 等待时间不包括紧接 STOP 模式释放后直到时钟振荡启动（下图中“a”）的时间。



2. 将时间设置为谐振器稳定 OSTS 寄存器所需时间的两倍（因为振荡稳定时间的一半是 PLL 的稳定时间）。
3. 复位之后的振荡稳定时间为 $2^{16}/f_x$ （因为 OSTS 寄存器的默认值= 06H）。
4. 一定要将位 3 至 7 设置为“0”。

备注 f_x : 振荡频率

(7) 时钟监视器模式寄存器 (CLM)

CLM 寄存器设置时钟监视器的操作模式。只有通过特定的流程才能写入该寄存器（见 3.4.7 特殊寄存器）。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

复位后: 00H R/W 地址: FFFFF870H

	7	6	5	4	3	2	1	0
CLM	0	0	0	0	0	0	0	CLME

CLME	时钟监视器操作控制
0	禁止时钟监视操作
1	允许时钟监视操作

- 注意事项**
1. CLME 位只会在复位后清除为 0。
 2. CLME 位 = 1 时，如果满足以下条件，那么时钟监视器的功能将会被强行终止。
 - 在 STOP 模式被释放后的振荡稳定时间计数期间。
 3. CLME 位 = 1 时，如果检测到振荡停止 (fx)，那么马达控制定时器的输出会进入高阻状态。
关于目标定时器输出请参见图 9-4。
 4. 写入 CLM 寄存器时，需使用命令寄存器 PRCMD。

<R>

5.4 PLL功能

5.4.1 概述

外围模块的 CPU 和操作时钟可以在 8 倍振荡频率输出和时钟直通模式间进行转换。

当使用 PLL 功能时: 输入时钟 (fx) = 2.5 MHz, 输出时钟 (fxx) = 20 MHz

时钟直通模式: 输入时钟 (fx) = 2.5 MHz, 输出时钟 (fxx) = 2.5 MHz

5.4.2 PLL 模式

在 PLL 模式中, 振荡频率 (fx) 通过 PLL 被扩大 8 倍生成系统时钟 (fxx)。

在 PLL 模式中, 时钟从振荡器输入到 PLL。 锁定时间 (频率稳定时间) 过去后, 在相位会被锁定在一个特殊的频率上且振荡稳定期间, 必须为内部电路提供一个具有稳定频率的时钟。在 V850ES/IE2 中, 复位释放后会自动获取锁定时间。

5.4.3 时钟直通模式

在时钟直通模式中, 会生成具有与振荡频率 (fx) 相同频率的系统时钟 (fxx)。

5.5 操作

5.5.1 各时钟的操作

下表显示各个时钟的操作状态：

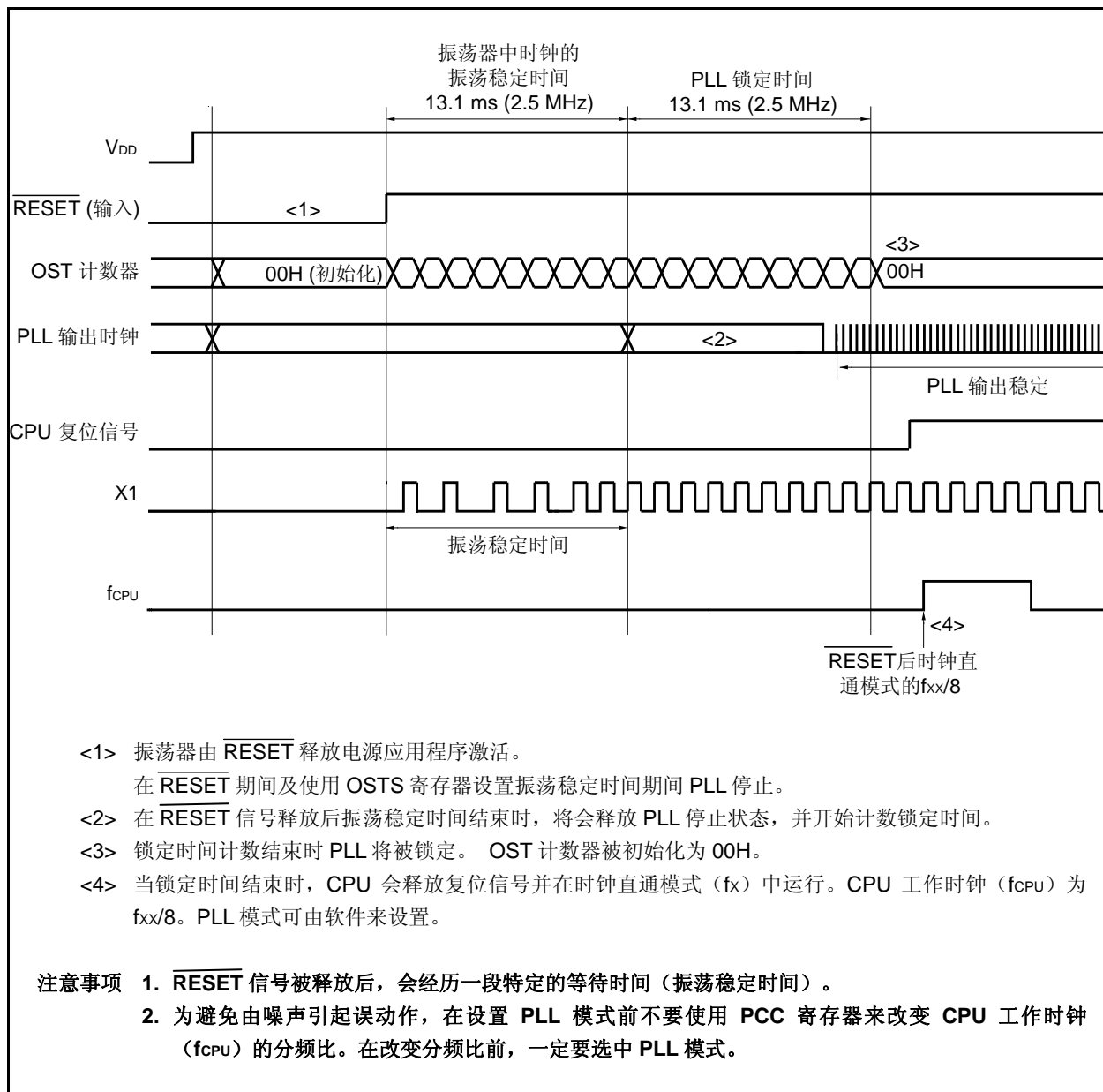
表 5-2. 各时钟的操作状态

节电模式	振荡器 (fx)	PLL (fxx)	内部系统时 钟 (fCLK)	CPU 时钟 (fCPU)	外围时钟 (fxx 至 fxx/2,048)	看门狗定时 器时钟 (fxx/512)
正常操作	√	√	√	√	√	√
RESET 引脚输入期间	×	×	×	×	×	×
振荡稳定时间计数期间	√	√	×	×	×	×
HALT 模式	√	√	√	×	√	√
IDLE 模式	√	√	×	×	×	×
STOP 模式	×	×	×	×	×	×

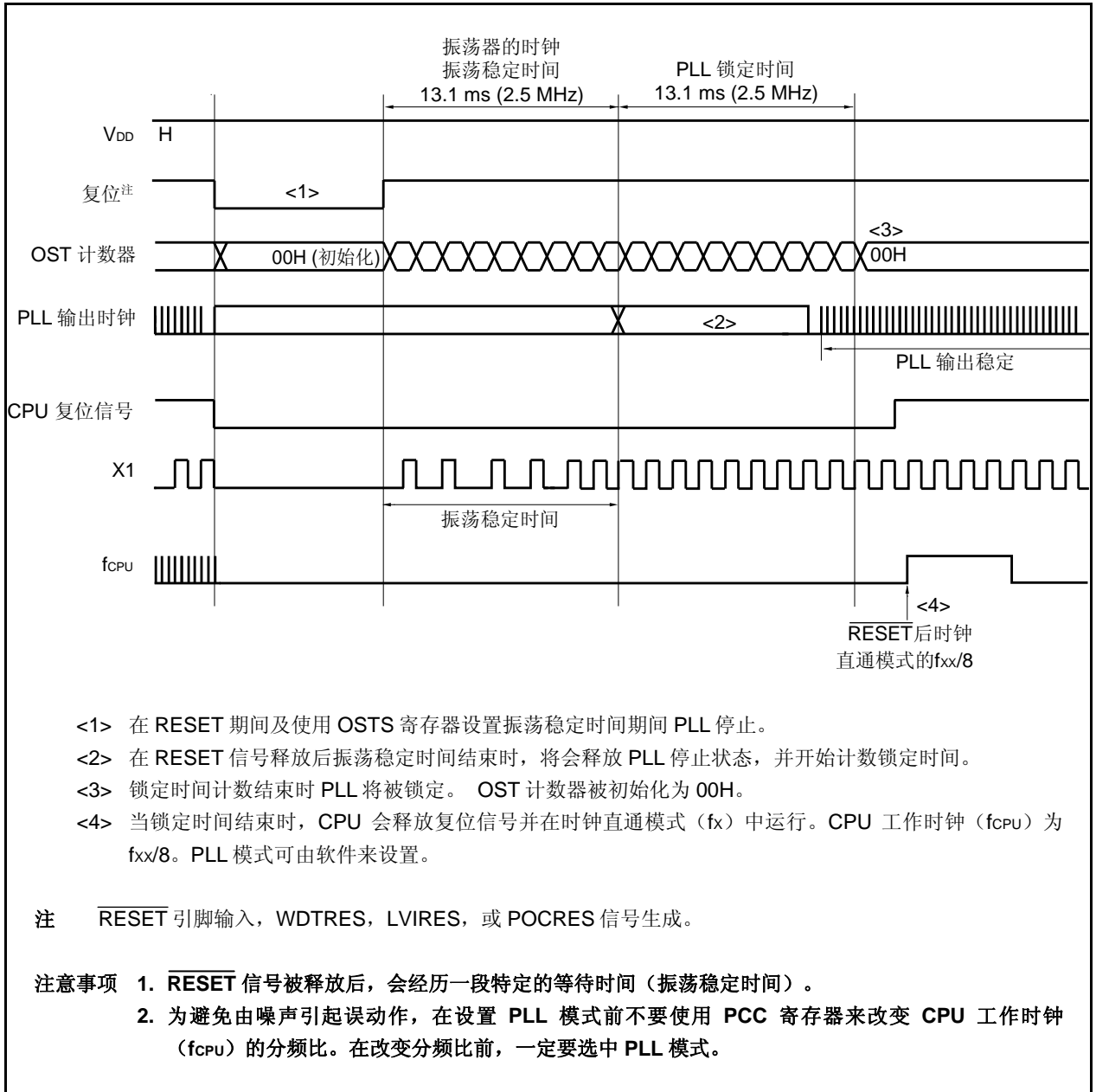
备注 √: 运行
 ×: 停止

5.5.2 操作时序

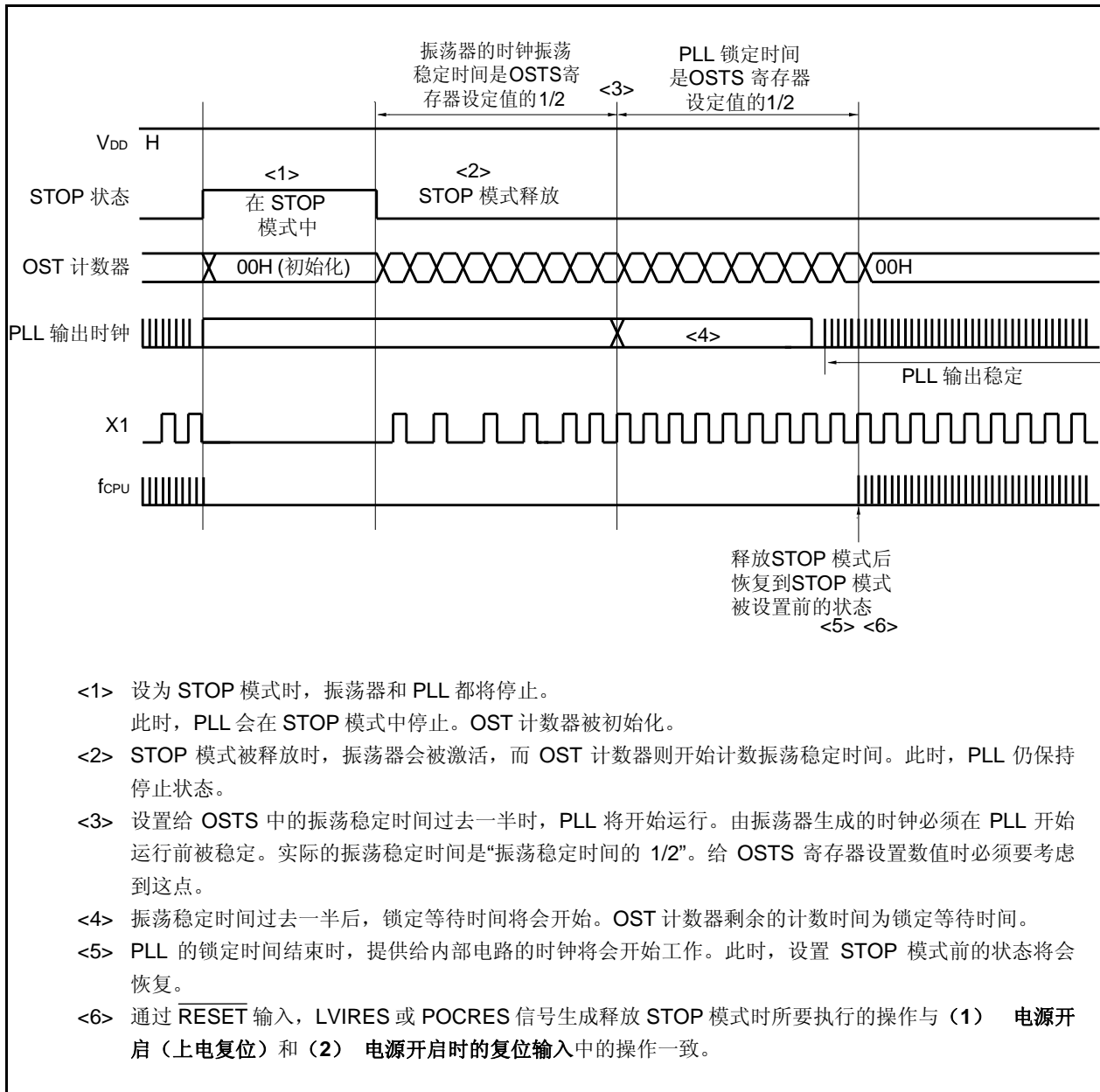
(1) 电源开启（上电复位）



(2) 电源开启时的复位输入



(3) 通过中断请求释放 STOP 模式



5.6 时钟监视器

(1) 功能

时钟监视器通过使用内部振荡时钟来采样由振荡器 (fx) 产生的时钟。当它检测到错误 (振荡停止) 时, 马达控制定时器的输出将会进入高阻状态。CLMER 信号 (低电平) 会从 P16 中输出 (详细信息请参见第四章 端口功能和第九章 马达控制功能)。低电平输出通过复位信号来释放。

注 CLMER 信号 (低电平) 只有在 P16 被设置为输出端口或 TOQ00 输出功能时有效。

(2) 操作

以下说明了时钟监视器的功能。操作开始条件和操作停止条件如下所示。

<操作开始条件>

设置 CLM.CLME 位为使能 (1)。

<操作停止条件>

STOP 模式被释放后对振荡稳定时间进行计数时。

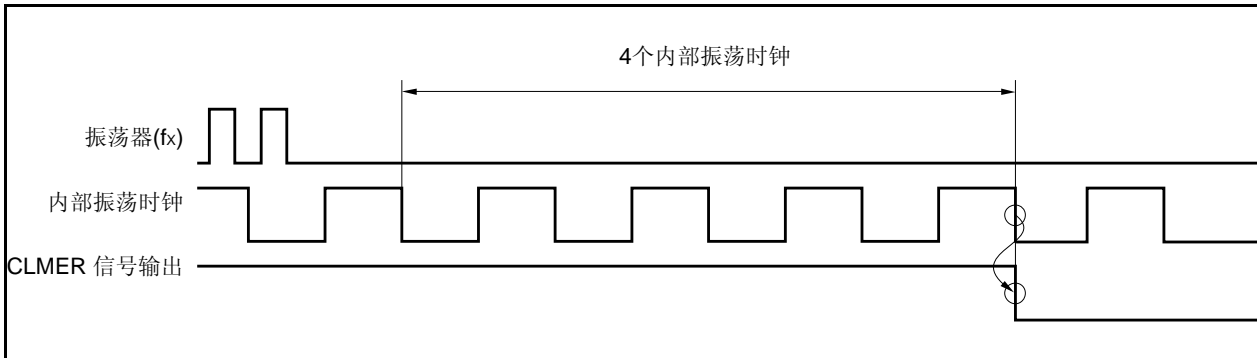
表 5-3. 时钟监视器的操作状态 (CLM.CLME 位 = 1 时)

操作模式	由振荡器生成的时钟 (fx) 的状态	内部振荡时钟的状态	时钟监视器的状态
正常操作模式	振荡	振荡	运行
HALT 模式	振荡	振荡	运行
IDLE 模式	振荡	振荡	运行
STOP 模式	停止	振荡	停止
复位期间	停止	停止	停止

(a) 振荡器停止时的操作 (CLM.CLME 位 = 1)

如果 CLME 位 = 1 时振荡器停止，CLMER 信号将会从 P16 中输出。

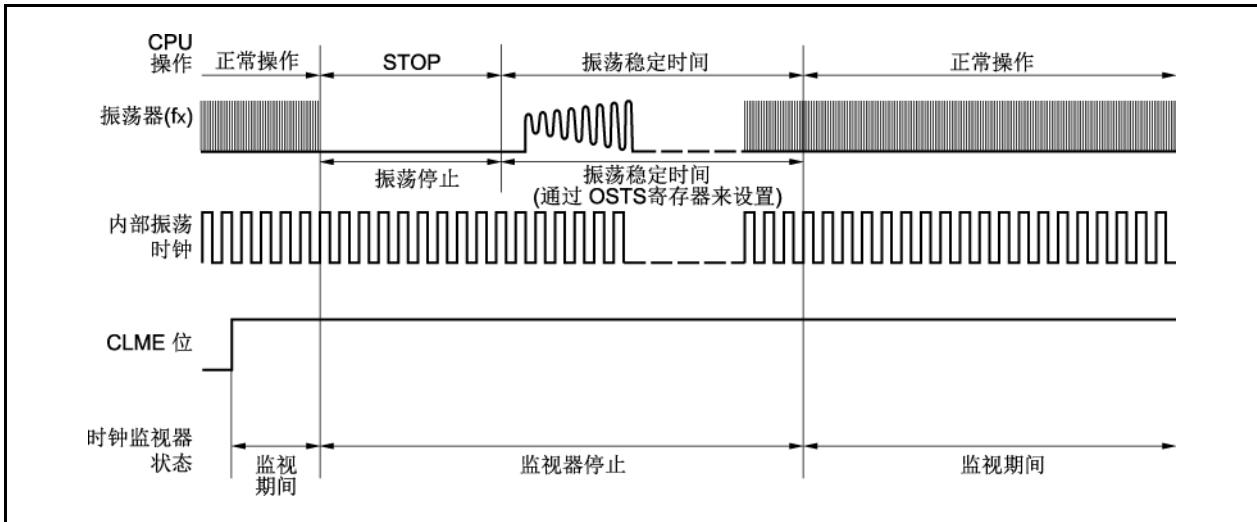
图 5-2. 主时钟停止时的振荡



(b) 在 STOP 模式时或 STOP 模式被释放后的操作

如果在 CLME 位 = 1 时设为 STOP 模式，那么监视器会在 STOP 模式中停止操作且此时会对振荡稳定时间进行计数。振荡稳定时间过后监视器将会自动开始工作。

图 5-3. 在 STOP 模式时或 STOP 模式释放后的操作



第六章 16 位定时器/事件计数器 P (TMP)

定时器 P (TMP) 是一个 16 位定时器/事件计数器。
V850ES/IE2 包括 TMP0 至 TMP3。

6.1 概要

以下介绍了 TMPn 的通道 (n = 0 至 3)。

表 6-1. TMPn 的概述

名称	TMP0	TMP1	TMP2	TMP3
时钟选择	8 通道	8 通道	8 通道	8 通道
捕获触发输入引脚	2	无	2	无
外部事件计数输入引脚	1	无	1	无
外部触发输入引脚	1	无	1	无
定时器计数器	1	1	1	1
捕获/比较寄存器	2	2 [#]	2	2 [#]
捕获/比较匹配中断请求信号	2	2 [#]	2	2 [#]
溢出中断请求信号	1	1	1	1
定时器输出引脚	2	无	1	1

<R>

注 仅限比较功能。

6.2 功能

如下表所示 (n = 0 至 3)，各个通道可以实现的 TMPn 的功能是不同的。

表 6-2. TMPn 的功能

功能	TMP0	TMP1	TMP2	TMP3
间隔定时器	√	√	√	√
外部事件计数器	√	×	√	×
外部触发脉冲输出	√	×	√	√ [#]
单次脉冲输出	√	×	√	√ [#]
PWM 输出	√	×	√	√
自由运行定时器	√	√	√	√
脉冲宽度测量	√	×	√	×
定时器调谐操作	×	√ (TMQ1)	×	×

注 只能通过软件触发实现。不能使用外部触发输入。

6.3 配置

定时器 TMPn 包括以下硬件：

表 6-3. TMPn 的配置

名称	配置
定时器寄存器	16 位计数器× 1
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 和 CCR1 缓冲寄存器
定时器输入	总共 4 个引脚 (TIP00 ^{註1} , TIP01, TIP20 ^{註1} , TIP21 引脚) ^{註2}
定时器输出	总共 4 个引脚 (TOP00, TOP01, TOP21, TOP31 引脚) ^{註3}
控制寄存器	TMPn 控制寄存器 0, 1 (TPnCTL0, TPnCTL1) TMPm 输入/输出控制寄存器 (TPmIOC0) TMPk 输入/输出控制寄存器 1, 2 (TPkIOC0, TPkIOC2) TMPn 选择寄存器 0 (TPnOPT0)

- 注**
1. TIP00 引脚和 TIP20 引脚复用功能包括：捕获输入引脚、外部事件计数输入引脚以及外部触发输入引脚共用。
 2. 不提供给 TMP1 和 TMP3。
 3. 不提供给 TMP1。

备注

n = 0 至 3
m = 0, 2, 3
k = 0, 2

图 6-1. TMP0 的框图

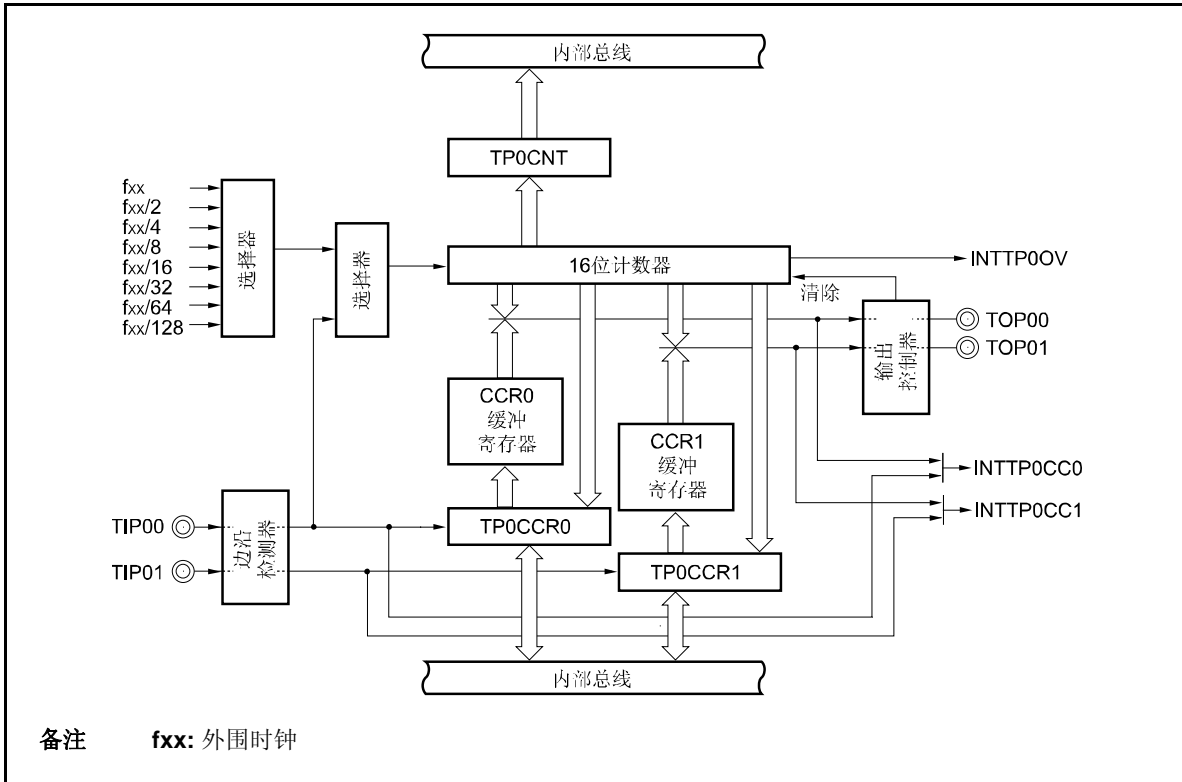


图 6-2. TMP1 的框图

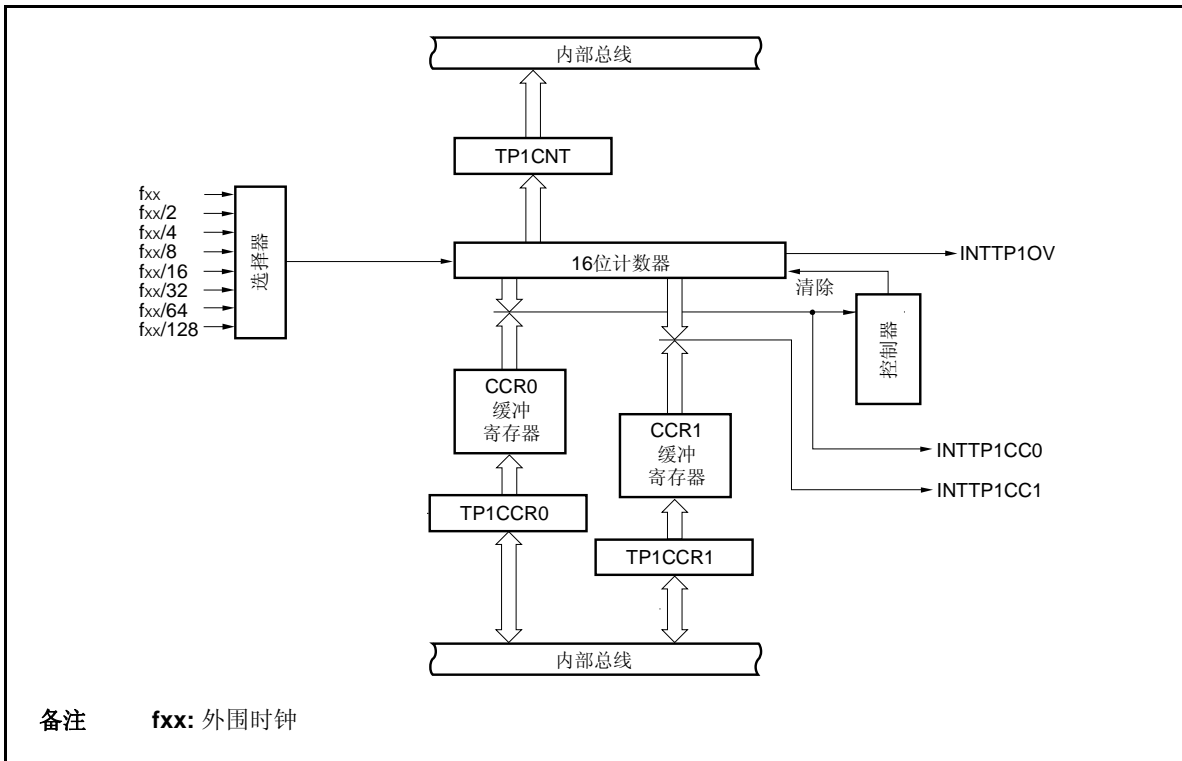


图 6-3. TMP2 的框图

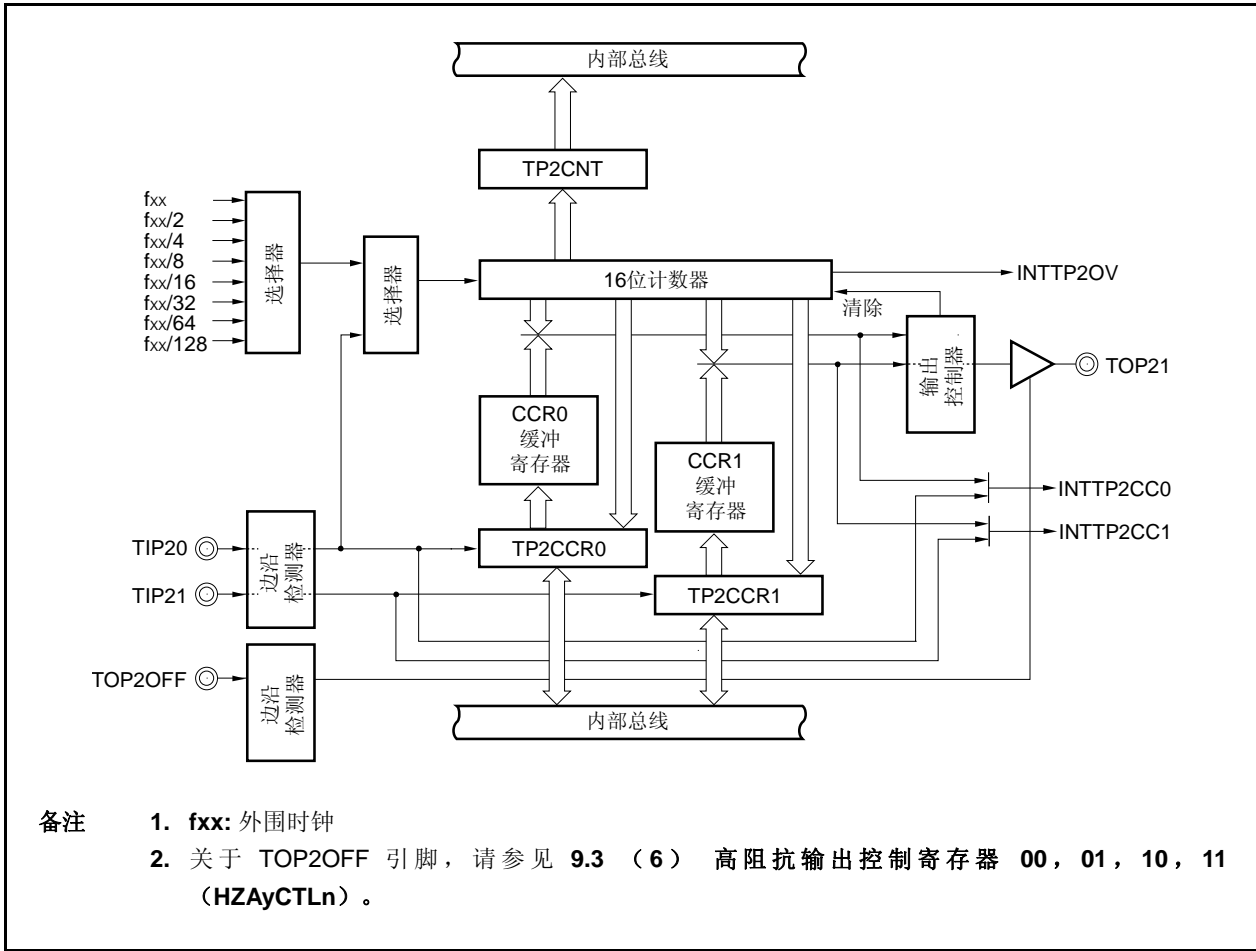
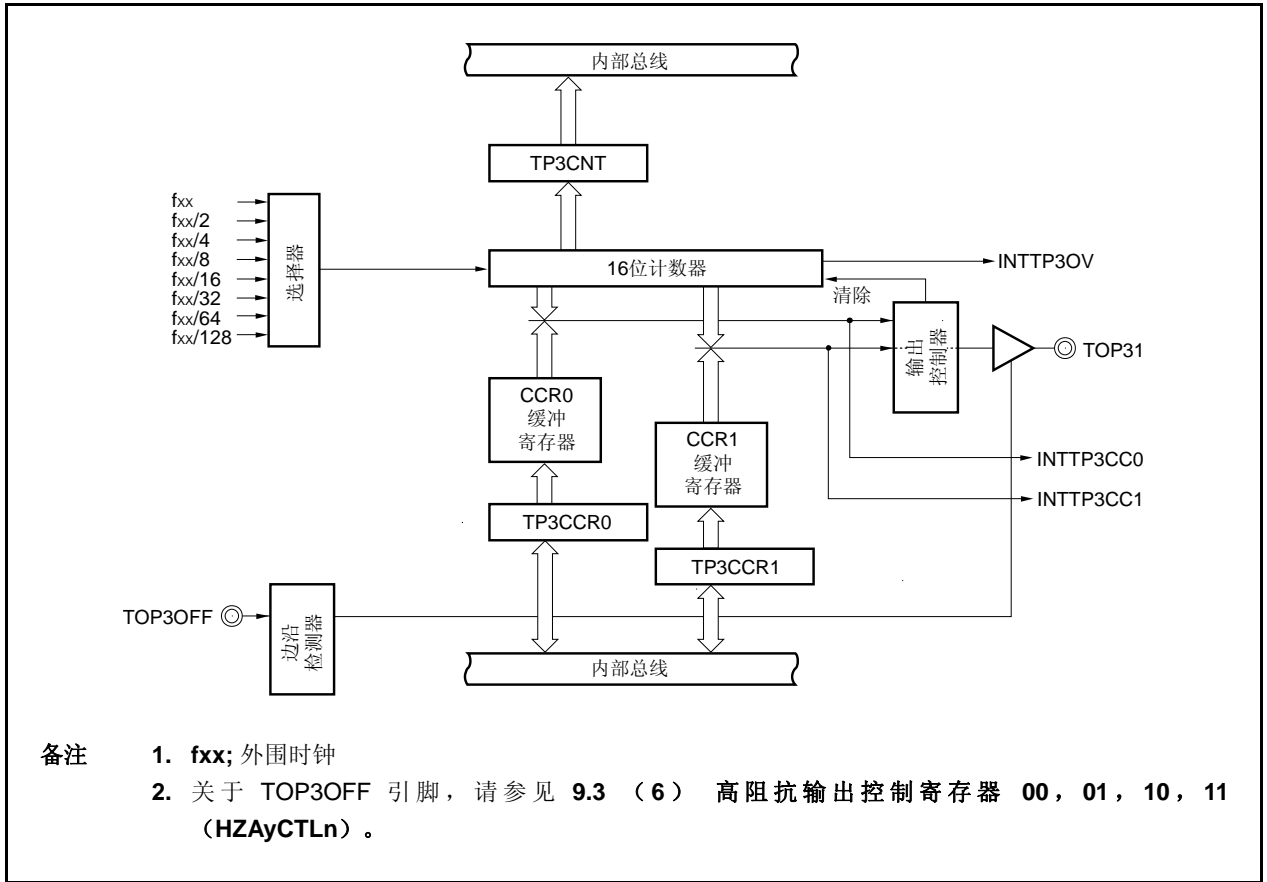


图 6-4. TMP3 的框图



(1) 16 位计数器

此 16 位计数器可以对内部时钟或外部事件计数。

可以使用 TPnCNT 寄存器读取该计数器的计数值。

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

TPnCE 位在复位后清除为 0。

(2) CCR0 缓冲寄存器

CCR1 是一个用于比较 16 位计数器的计数值的 16 位比较寄存器。

当 TPnCCR0 寄存器用作比较寄存器时, 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 缓冲寄存器的值相符时, 就会生成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。

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

(3) CCR1 缓冲寄存器

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

当 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 缓冲寄存器中。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。

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

(4) 边沿检测器

该电路检测输入到 TIP00, TIP01, TIP20 和 TIP21 引脚信号的有效沿，使用 TP0IOC1, TP0IOC2, TP0IOC2 和 TP2IOC2 寄存器设置，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。

(5) 输出控制器

该电路控制 TOP00, TOP01, TOP21 和 TOP31 引脚的输出，TOP00, TOP01, TOP21 和 TOP31 引脚的输出由 TP0IOC0, TP2IOC0 和 TP3IOC0 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为计数时钟。

6.4 寄存器

(1) TMPn 控制寄存器 0 (TPnCTL0)

TPnCTL0 寄存器为 8 位寄存器，用来控制定时器 TMPn 的操作。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

可使用软件向 TPnCTL0 寄存器写入相同的值。

复位后: 00H R/W 地址: TP0CTL0 FFFFF640H, TP1CTL0 FFFFF660H,
TP2CTL0 FFFFF680H, TP3CTL0 FFFFF6A0H

	<7>	6	5	4	3	2	1	0
TPnCTL0	TPnCE	0	0	0	0	TPnCKS2	TPnCKS1	TPnCKS0
(n = 0 至 3, m = 0, 2, 3)								
	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				
	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 TPnOPT0.TPnOVF 位和 16 位计数器同时复位。此外，定时器输出 (TOP00, TOP01, TOP21, 以及 TOP31 引脚) 与 16 位计数器同时复位到 TPmIOC0 寄存器设定状态。

注意事项 1. 当 TPnCE 位=0 时，设置 TPnCKS2 至 TPnCKS0 位。

当 TPnCE 位的值从 0 变为 1 时，可以同时设定 TPnCKS2 至 TPnCKS0 位。

2. 确保位 3 至 6 设置为 0。

备注 f_{xx}: 外围时钟

(2) TMPn 控制寄存器 1 (TPnCTL1)

TPnCTL1 寄存器为 8 位寄存器，用来控制定时器 TMPn 的操作。

该寄存器可进行 8 位或 1 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: TP0CTL1 FFFFF641H, TP1CTL1 FFFFF661H,
TP2CTL1 FFFFF681H, TP3CTL1 FFFFF6A1H

TPnCTL1	7	6	5	4	3	2	1	0
(n = 0 至 3 m = 0, 2, 3 k = 0, 2)	TP1SYE ^{注1}	TPmEST ^{注2}	TPKEEE ^{注3}	0	0	TPnMD2	TPnMD1	TPnMD0
TP1SYE ^{注1}		操作模式选择						
0		TMP1 单独模式						
1		调谐操作模式 (参见9.4.5)						
在调谐操作期间, TMP1只能用作A/D转换器, 0和1 A/D转换开始触发系数在调谐操作模式中, 该位始终与TMQ1同步操作。								
TPmEST ^{注2}		软件触发控制						
0		-						
1		为外部触发输入产生一个有效信号。 · 在单脉冲输出模式中: 通过写入1至 TPmEST位, 单脉冲输出为触发信号。 · 在外部触发脉冲输出模式中: 通过写入1至 TPmEST位, PWM波形输出为触发信号。						
读取 TPmEST位始终为0。								
TPKEEE ^{注3}		计数时钟选择						
0		禁止通过外部事件计数输入(TIPk0引脚)的操作。 (使用通过TPkCTL0.TPkCKS0 至 TPkCTL0.TPkCKS2位所选择的计数时钟执行计数。)						
1		允许通过外部事件计数输入(TIPk0引脚)的操作 ^{注4} (通过外部事件计数输入信号的有效沿执行计数。)						
TPKEEE 位用于选择计数是通过外部事件计数时钟还是外部事件计数输入的有效沿来执行。								

- 注
1. 该位只能在 TMP1 中设置。务必将 TMP0, TMP2, 以及 TMP3 的位 7 清除为 0。关于调谐操作模式的详细信息, 请参见第 9 章 马达控制功能。
 2. 该位只能在 TMP0, TMP2, 以及 TMP3 中设置。务必将 TMP1 的位 6 清除为 0。
 3. 该位只能在 TMP0 以及 TMP2 中设置。务必将 TMP1 和 TMP3 的位 5 清除为 0。
 4. 将捕获触发输入 (TIPk0 引脚) 的有效边沿选择设置为“无边沿检测”。

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

注 每个通道中可以实现的设置是不同的。详细信息，请参见表 6-8 至 6-11。

- 注意事项
1. TPmEST 位只有在外触发脉冲输出模式或单次脉冲输出模式时有效。在其它模式下，忽略向该位写入 1 的操作。
 2. 不论 TPkEEE 位的取值如何，在外部事件计数模式下，都选择外部事件输入端。
 3. 当 TPnCTL0.TPnCE 位 = 0 时，置位 TP1SYE 位，TPkEEE 位和 TPnMD2 至 TPnMD0 位。
(TPnCE 位 = 1 时可写入相同的值。) 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，再重新进行位设定。
 4. 一定要把位 3 至 4 清为 0。

<R>

(3) TMPm 输入/输出控制寄存器 0 (TPmIOC0)

TPmIOC0 寄存器为 8 位寄存器，用来控制定时器的输出 (TOP00, TOPm1 引脚)。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

备注 TMP1 不包含 TP1IOC0 寄存器。

(1/2)

复位后: 00H R/W 地址: TP0IOC0 FFFFF642H, TP2IOC0 FFFFF682H,
TP3IOC0 FFFFF6A2H

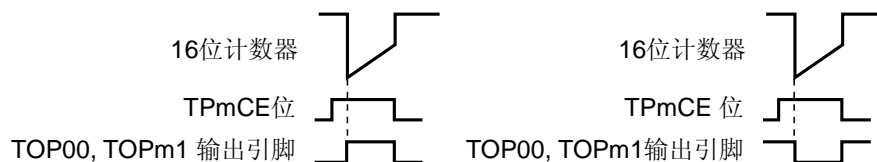
	7	6	5	4	3	<2>	1	<0>
TPmIOC0 (m = 0, 2, 3)	0	0	0	0	TPmOL1	TPmOE1	TP0OL0 ^{注1}	TP0OE0 ^{注1}
TPmOL1	TOPm1 引脚输出电平设置 ^{注2}							
0	TOPm1 引脚在高电平开始输出。							
1	TOPm1 引脚在低电平开始输出。							
TPmOE1	TOPm1 引脚输出设置							
0	定时器输出禁止 · TPmOL1 位 = 0 时，从 TOPm1 引脚输出低电平。 · TPmOL1 位 = 1 时，从 TOPm1 引脚输出高电平。							
1	定时器输出允许 (从 TOPm1 引脚输出脉冲。)							
TP0OL0 ^{注1}	TOP00 引脚输出电平设置 ^{注2}							
0	TOP00 引脚在高电平时开始输出。							
1	TOP00 引脚在低电平时开始输出。							
TP0OE0 ^{注1}	TOP00 引脚输出设置							
0	定时器输出禁止 · TP0OL0 位 = 0 时，从 TOP00 引脚输出低电平。 · TP0OL0 位 = 1 时，从 TOP00 引脚输出高电平。							
1	定时器输出允许 (从 TOP00 引脚输出脉冲。)							

- 注**
1. 仅对 TMP0 有效。一定要将 TMP2 和 TMP3 的位 1 和位 0 清为 0。
 2. 由 TPmOLa (a = 0, 1) 位指定的定时器输出引脚 (TOP00 和 TOPm1) 的输出电平如下所示 (a = 0, 1)。

· 当 TPmOLa 位 = 0 时

· 当 TPmOLa 位 = 1 时

<R>



<R>

- 注意事项
1. 在输出模式下当 TOP00 和 TOPm1 置位时，如果 TPmIOC0 寄存器的设置改变，则引脚的输出改变。在输入模式和使端口进入高阻状态时设置端口时，则引脚的状态不会改变。
 2. 当 TPmCTL0.TPnCE 位 = 0 时，重新写入 TPmOL1, TPmOE1, TP0OL0 和 TP0OE0 位。（当 TPmCE 位 = 1 时，可以写入相同的值。）如果重新写入不能正确执行，将 TPmCE 位清为 0，然后重新进行位设定。
 3. 在 TPmCE, TP0OE0 和 TPmOE1 位为 0 时，即使对 TP0OL0 和 TPmOL1 位进行了操作，TOP00 和 TOPm1 引脚的输出电平也会改变。

(4) TMPk 输入/输出控制寄存器 1 (TPkIOC1)

TPkIOC1 寄存器为 8 位寄存器，控制捕获触发输入信号 (TIPk0, TIPk1 引脚) 的有效边沿。
该寄存器可进行 8 位或 1 位读写。
系统复位后该寄存器被设为 00H。

备注 TMP1 和 TMP3 不包含 TP1IOC2 和 TP3IOC2 寄存器。

复位后: 00H R/W 地址: TP0IOC1 FFFFF643H, TP2IOC1 FFFFF683H

	7	6	5	4	3	2	1	0
TPkIOC1 (k = 0, 2)	0	0	0	0	TPkIS3	TPkIS2	TPkIS1	TPkIS0

TPkIS3	TPkIS2	捕获触发输入信号 (TIPk1 引脚) 有效沿设置
0	0	无边沿检测(捕获操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双边沿

TPkIS1	TPkIS0	捕获触发输入信号1 (TIPk0 引脚) 有效沿设置
0	0	无边沿检测(捕获操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双边沿

- 注意事项**
1. 在 TPkCTL0.TPkCE 位 = 0 时才可对 TPkIS3 至 TPkIS0 位进行改写。(TPkCE 位 = 1 时可以向这些位写入相同的值)。如果由于误操作而引起改写，则要将 TPKCE 位清零然后再次设置这些位。
 2. TPkIS3 至 TPkIS0 位只有在自由运行定时器模式 (仅当 TPnOPT0.TPkCCS1 位和 TPKCCS0 位 = 11 时) 和脉冲宽度测量模式下才有效。在其他模式下，不能进行捕获操作 (仅限 TMP0 和 TMP2)。

(5) TMPk 输入/输出控制寄存器 2 (TPkIOC2)

TPkIOC2 寄存器为 8 位寄存器，控制外部事件计数器的输入信号 (TIPk0 引脚) 和外部触发输入信号 (TIPk0 引脚) 的有效边沿。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

备注 TMP1 和 TMP3 不包含 TP1IOC2 和 TP3IOC2 寄存器。

复位后: 00H R/W 地址: TP0IOC2 FFFF644H, TP2IOC2 FFFF684H

	7	6	5	4	3	2	1	0
TPkIOC2	0	0	0	0	TPKEES1	TPKEES0	TPKETS1	TPKETS0

(k = 0, 2)

TPKEES1	TPKEES0	外部事件计数输入信号1 (TIPk0 引脚) 有效沿设置
0	0	无边沿检测 (外部事件计数无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双边沿

TPKETS1	TPKETS0	外部触发输入信号1 (TIPk0 引脚) 有效沿设置
0	0	无边沿检测 (外部触发无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测双边沿

- 注意事项**
1. 在 TPkCTL0.TPkCE 位 = 0 时才可对 TPKEES1, TPKEES0, TPKETS1, 和 TPKETS0 位进行改写 (TPkCE 位 = 1 时可以向这些位写入相同的值)。如果重新写入不能正确执行, 将 TPkCE 位清为 0, 然后重新进行位设定。
 2. 只有在 TPkCTL1.TPKEEE 位 = 1 或设置了外部事件计数模式 (TPkCTL1.TPkMD2 至 TPkCTL1.TPkMD0 位 = 001) 时, TPKEES1 和 TPKEES0 位才有效。
 3. TPKETS1 位和 TPKETS0 位只有在外部触发脉冲输出模式或单次脉冲输出模式时才有效。

(6) TMPn 选择寄存器 0 (TPnOPT0)

TPnOPT0 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。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

复位后: 00H R/W 地址: TP0OPT0 FFFFF645H, TP1OPT0 FFFFF665H,
TP2OPT0 FFFFF685H, TP3OPT0 FFFFF6A5H

	7	6	5	4	3	2	1	<0>
TPnOPT0 (n = 0 至 3, k = 0, 2)	0	0	TPkCCS1 ^注	TPkCCS0 ^注	0	0	0	TPnOVF

TPkCCS1 ^注	TPkCCR1 寄存器捕获/比较选择
0	选择比较寄存器
1	选择捕获寄存器 (通过 TPkCTL0.TPkCE 位= 0来清除)
TPkCCS1 位的设置仅在自由运行模式中有效。	

TPkCCS0 ^注	TPkCCR0 寄存器捕获/比较选择
0	选择比较寄存器
1	选择捕获寄存器 (通过 TPkCTL0.TPkCE 位= 0来清除)
TPkCCS0 位的设置仅在自由运行模式中有效。	

TPnOVF	TMPn 溢出检测标志
置(1)	发生溢出
复位(0)	0 写入 TPnOVF 位或 TPnCTL0.TPnCE 位 = 0
<ul style="list-style-type: none"> · 在自由运行定时器模式或脉冲宽度测量模式中，当16位计数器的值从FFFFH到0000H时，TPnOVF位置为1。 · TPnOVF位置为1的同时会生成一个溢出中断请求信号(INTTPnOV)。INTTPnOV信号不会在自由运行定时器模式和脉冲宽度测量模式外的其他模式中生成。 · 即使在TPnOVF位 = 1时TPnOVF位或TPnOPT0寄存器被读取，TPnOVF位也不会清零。 · 在生成INTTPnOV信号后会清除TPnOVF位为0，在此之前必须确保（通过读取）TPnOVF位被置1。 · TPnOVF位既可以读取也可以写入，但TPnOVF位不能通过软件置为1。写入1对TMPn的操作不会产生影响。 	

注 仅对 TMP0 和 TMP2 有效。一定要将 TMP1 和 TMP3 的位 5 和位 4 清为 0。

注意事项 1. 当 TPkCE 位=0 时，重新写入 TPkCCS1 和 TPkCCS0 位。（当 TPkCE 位=1 时可写入相同值。）如果重新写入不能正确执行，将 TPkCE 位清为 0，然后重新进行位设定。

2. 一定要将位 1 至 3，6 和 7 清为“0”。

(7) TMPn 捕获/比较寄存器 0 (TPnCCR0)

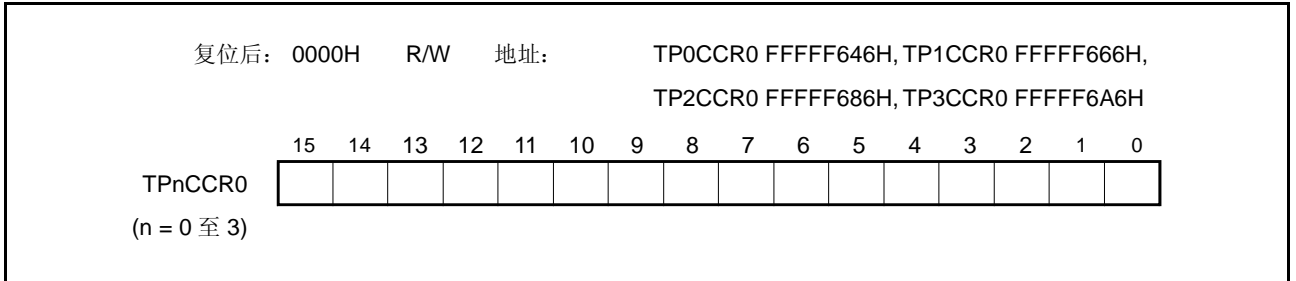
根据不同工作模式，TP0CCR0 和 TP2CCR0 寄存器是可以用作捕获寄存器或比较寄存器的 16 位寄存器。TP1CCR0 和 TP3CCR0 寄存器是只能用作比较寄存器的 16 位寄存器。

只有在自由运行模式下，依据 TP0OPT0.TP0CCS0 位或 TP2OPT0.TP2CCS0 位的设定，TP0CCR0 和 TP2CCR0 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，TP0CCR0 寄存器只可以用作捕获寄存器，而在其它模式时，该寄存器只可以用作比较寄存器。

TP0CCR0 寄存器可以在运行中进行读取和写入。

该寄存器可以按 16 位宽度进行读写。

复位输入将把该寄存器设置为 0000H。



(a) 用作比较寄存器

即便在 TPnCTL0.TPnCE 位=1 时, TPnCCR0 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 TOP00 引脚输出允许, 则 TOP00 引脚的输出会反相。(在 V850ES/IE2 中不提供 TOP10 引脚, TOP20 引脚和 TOP30 引脚)。

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)。

在 TPnCTL0.TPnCE=0 时, 比较寄存器不能清除。

(b) 用作捕获寄存器时的功能 (仅限 TP0CCR0 和 TP2CCR0 寄存器)

在自由运行定时器模式下, 当 TPkCCR0 寄存器用作捕获寄存器时, 若检测到捕获触发输入引脚 (TIPk0 引脚) 的有效边缘, 则 16 位计数器的当前计数值存储于 TPkCCR0 寄存器中。在脉宽测量模式下, 若检测到捕获触发输入引脚 (TIPk0) 的有效边缘, 16 位计数器的当前计数值就储存在 TPkCCR0 寄存器中, 且 16 位计数器清零 (0000H)。

即使捕获操作与对 TPkCCR0 寄存器的读取发生冲突, 还是可以正确读取 TPkCCR0 寄存器的值。

在 TPkCTL0.TPkCE 位=0 时, 捕获寄存器不能清除。

备注 k = 0, 2

下表显示捕获/比较寄存器在各工作模式下的功能, 以及怎样向比较寄存器写入数据。

表 6-4. 捕获/比较寄存器在各模式下的功能和如何写入比较寄存器

工作模式	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 ^{注1}	比较寄存器	任何时候均可写入
外部触发脉冲输出 ^{注2}	比较寄存器	批写入 ^{注4}
外部触发脉冲输出 ^{注2}	比较寄存器	任何时候均可写入
PWM 输出 ^{注3}	比较寄存器	批写入 ^{注4}
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 ^{注1}	捕获寄存器	无

- 注**
1. 仅限 TMP0 和 TMP2
 2. 仅限 TMP0 和 TMP2 (软件触发模式中还有 TMP3)
 3. 仅限 TMP0, TMP2 和 TMP3
 4. 写入 TPnCCR1 寄存器的是触发脉冲。

备注 关于随时写入和批写入的详细信息, 请参见 6.6 (2) 随时写入和批写入。

(8) TMPn 捕获/比较寄存器 1 (TPnCCR1)

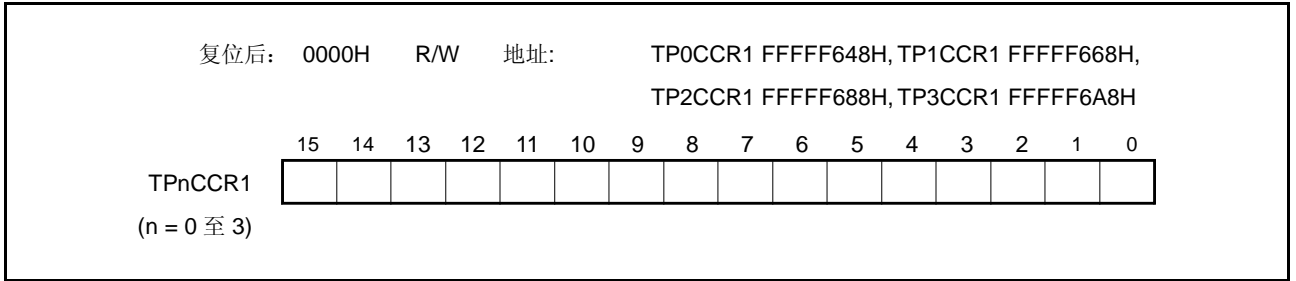
根据不同工作模式，TP0CCR1 和 TP2CCR1 寄存器是可以用作捕获寄存器或比较寄存器的 16 位寄存器。TP1CCR1 和 TP3CCR1 寄存器是只能用作比较寄存器的 16 位寄存器。

TP0CCR1 和 TP2CCR1 寄存器只有在自由运行模式下，可以根据 TP0OPT0.TP0CCS1 或 TP2OPT0.TP2CCS1 位的不同设置用作捕获寄存器或比较寄存器。在脉宽测量模式中，TPnCCR1 寄存器只能用作捕获寄存器，而在其它模式时，该寄存器只能用作比较寄存器。

TPnCCR1 寄存器可以在操作过程中进行读取和写入。

该寄存器可以 16 位为单位进行读取和写入。

复位输入将把该寄存器设置为 0000H。



(a) 用作比较寄存器时的功能

即便在 TPnCTL0.TPnCE 位=1 时, TPnCCR1 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 TOPm1 引脚输出允许, 则 TOPm1 引脚输出反相 (V850ES/IE2 中不提供 TOP11 引脚)。

TPnCTL0.TPnCE 位 = 0 时没有清除比较寄存器。

备注 m = 0, 2, 3

(b) 作为捕获寄存器时的功能 (仅限 TP0CCR1 和 TP2CCR1 寄存器)

当 TPkCCR1 寄存器在自由运行定时器模式下被用作捕获寄存器时, 如果检测到了捕获触发输入引脚 (TIPk1 引脚) 的有效边沿, 那么 16 位计数器的计数值将被保存到 TPkCCR1 寄存器中。在脉宽测量模式下, 如果检测到捕获输入引脚 (TIPk1 引脚) 的有效沿, 那么 16 位计数器的计数值将被保存到 TPkCCR1 寄存器, 同时 16 位计数器清为 (0000H)。

即使捕获操作和读取 TPkCCR1 寄存器操作冲突, 仍可正确读出 TPkCCR1 寄存器中的值。

在 TPkCTL0.TPkCE=0 时, 捕获寄存器不能清除。

备注 k = 0, 2

下表中列出了捕获/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 6-5. 捕获/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

工作模式	捕获/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	任何时候均可写入
外部事件计数器 ^{注1}	比较寄存器	任何时候均可写入
外部触发脉冲输出 ^{注2}	比较寄存器	批写入 ^{注4}
外部触发脉冲输出 ^{注2}	比较寄存器	任何时候均可写入
PWM 输出 ^{注3}	比较寄存器	批写入 ^{注4}
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 ^{注1}	捕获寄存器	无

- 注**
1. 仅限 TMP0 和 TMP2。
 2. 仅限 TMP0 和 TMP2 (软件触发模式中还有 TMP3)。
 3. 仅限 TMP0, TMP2, 和 TMP3。
 4. 写入 TPnCCR1 寄存器的是触发脉冲。

备注 关于随时写入和批写入的详细信息, 请参见 6.6 (2) 随时写入和批写入。

(9) TMPn 计数器读取缓冲寄存器 (TPnCNT)

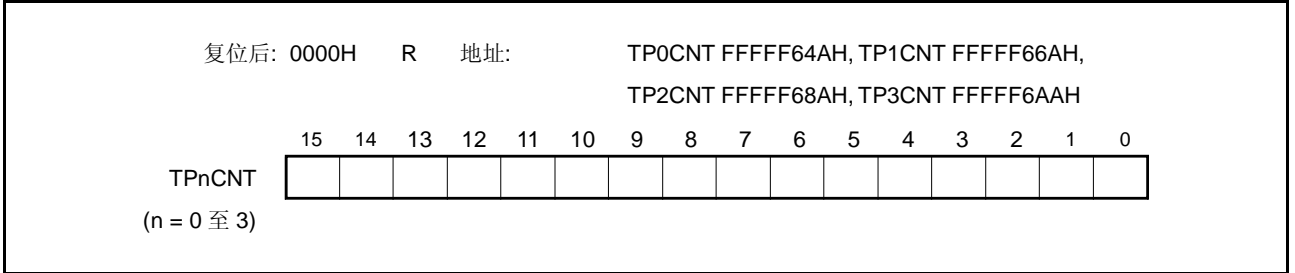
可以通过读取缓冲寄存器 TPnCNT 来读取 16 位计数器的计数值。

如果在 TPnCTL0.TPnCE 位=1 时读取该寄存器，16 位定时器的计数值可以被读出。

该寄存器为只读寄存器，可由 16 位存储器指令设置该寄存器。

当 TPnCE 位= 0 时，TPnCNT 寄存器的值被清为 0000H。如果此时读取 TPnCNT 寄存器，则不会读出 16 位计数器的值 (FFFFH)，读出的值为 0000H。+

复位后，TPnCE 位被清零的同时 TPnCNT 寄存器的值被清除为 0000H。



6.5 定时器输出操作

TOP00 引脚和 TOPm1 引脚的工作模式与输出电平的关系如下表所示。

表 6-6. 各种模式下的定时器输出控制

工作模式	TOPm1 引脚	TOP00 引脚
间隔计时器模式	PWM 输出	
外部事件计数模式	无	
外部触发脉冲输出模式	外部触发脉冲输出	PWM 输出
单次脉冲输出模式	单次脉冲输出	
PWM 输出模式	PWM 输出	
自由定时器模式	PWM 输出 (仅限使用比较功能时)	
脉冲宽度测量模式	无	

备注 m = 0, 2, 3

表 6-7. TOP00 和 TOPm1 引脚受控于定时器输出控制位的真值表

TPmIOC0.TPmOLa 位	TPmIOC0.TPmOEa 位	TPmCTL0.TPmCE 位	TOPma 引脚的电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前为低电平, 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 时 a = 0, 1
m = 2, 3 时 a = 1

6.6 操作

每个通道中可以实现的 TMPn 功能是不同的。各个通道的功能如下所示。

表 6-8. 各个模式中 TMP0 的规范

操作	TP0CTL1.TP0EST 位 (软件触发位)	TIP00 引脚 (外部触 发输入)	捕获/比较寄存器设置	比较寄存器的写入方 法
间隔计时器模式	无效	无效	仅比较寄存器有效	任何时候均可写入
外部事件计数模式 ^{注1}	无效	无效	仅比较寄存器有效	任何时候均可写入
外部触发脉冲输出模式 ^{注2}	有效	有效	仅比较寄存器有效	批写入
外部触发脉冲输出模式 ^{注2}	有效	有效	仅比较寄存器有效	任何时候均可写入
PWM 输出模式	无效	无效	仅比较寄存器有效	批写入
自由定时器模式	无效	无效	可变换的	任何时候均可写入
脉冲宽度测量模式 ^{注2}	无效	无效	仅捕获寄存器有效	无效

- 注**
1. 使用外部事件计数模式时，将 TIP00 引脚捕获触发输入有效边沿选择设置为“无边沿检测”。（将 TP0IOC1.TP0IS1 和 TP0IOC1.TP0IS0 位清除为 00。）
 2. 当用作外部触发脉冲输出模式、单次脉冲输出模式和脉冲宽度测量模式时，选择内部时钟作为计数时钟。（通过清除 TP0CTL1.TP0EEE 位为 0）。

备注 TIP00 引脚的复用功能包括捕获触发输入，外部事件计数输入和外部触发输入。

表 6-9. 各个模式中 TMP1 的规范

操作	软件触发位	外部触发输入	捕获/比较寄存器设置	比较寄存器的写入方 法
间隔计时器模式	无效	无效	仅比较寄存器有效	任何时候均可写入
外部事件计数模式	无			
外部触发脉冲输出模式	无			
单次脉冲输出模式	无			
PWM 输出模式	无			
自由定时器模式	无效	无效	仅比较寄存器有效	任何时候均可写入
脉冲宽度测量模式	无			

- 备注**
1. TMP1 不包含定时器输入引脚 (TIP10, TIP11) 和定时器输出引脚 (TOP10, TOP11)。它具有指示 16 位计数器的值与 TP1CCR0 和 TP1CCR1 寄存器的值间是否匹配的中断请求信号 (INTTP1CC0, INTTP1CC1)。
 2. TMP1 具有通过 TMQ1 执行调谐的功能。详细信息请参见第 9 章 马达控制功能。

表 6-10. 各个模式中 TMP2 的规范

操作	TP2CTL1.TP2EST 位 (软件触发位)	TIP20 引脚 (外部触 发输入)	捕获/比较寄存器设置	比较寄存器的写入方 法
间隔计时器模式	无效	无效	仅比较寄存器有效	任何时候均可写入
外部事件计数模式 ^{注 1}	无效	无效	仅比较寄存器有效	任何时候均可写入
外部触发脉冲输出模式 ^{注 2}	有效	有效	仅比较寄存器有效	批写入
外部触发脉冲输出模式 ^{注 2}	有效	有效	仅比较寄存器有效	任何时候均可写入
PWM 输出模式	无效	无效	仅比较寄存器有效	批写入
自由定时器模式	无效	无效	可变换的	任何时候均可写入
脉冲宽度测量模式 ^{注 2}	无效	无效	仅捕获寄存器有效	无效

- 注**
1. 使用外部事件计数模式时，将 TIP20 引脚捕获触发输入有效边沿选择设置为无边沿检测。（将 TP2IOC1.TP2IS1 和 TO2IOC1.TP2ISO 位清除为 00。）
 2. 使用外部触发脉冲输出模式和单脉冲输出模式时，请选择内部时钟作为计数时钟（通过将 TP2CTL1.TP2EEE 位清零）。

备注 TIP20 引脚的复用功能包括捕获触发输入，外部事件计数输入和外部触发输入。

表 6-11. 各个模式中 TMP3 的规范

操作	TP3CTL1.TP3EST 位 (软件触发位)	外部触发输入	捕获/比较寄存器设置	比较寄存器的写入方 法
间隔计时器模式	无效	无效	仅比较寄存器有效	任何时候均可写入
外部事件计数模式	无			
外部触发脉冲输出模式 ^注	有效	无效	仅比较寄存器有效	批写入
外部触发脉冲输出模式 ^注	有效	无效	仅比较寄存器有效	任何时候均可写入
PWM 输出模式	无效	无效	仅比较寄存器有效	批写入
自由运行模式。	无效	无效	仅比较寄存器有效	任何时候均可写入
脉冲宽度测量模式	无			

注 使用外部触发脉冲输出模式和单脉冲输出模式时，不能输入外部触发脉冲。只能使用软件触发（通过 TP3CTL1.TP3EST 位来设置）。

备注 TMP3 不包含定时器输入引脚（TIP30，TIP31）和定时器输出引脚（TOP30）。提供 16 位计数器与 TP3CCR0 和 TP3CCR1 寄存器的匹配中断请求信号（INTTP3CC0，INTTP3CC1）。

(1) 计数器基本操作

本节介绍了 16 位计数器的基本操作。 详细信息，请参照各个模式中对操作的具体描述。

备注 n = 0 至 3
 k = 0, 2

<R>

(a) 计数起始操作

- 外部事件计数模式

当 TPkCTL0.TPkCE 位置为从 0 至 1 时，16 位计数器被设置为 0000H。

然后，开始向上计数到 0001H, 0002H, 0003H, ...检测每一次外部事件计数输入 (TIPk0) 的有效边沿。

- 在非上述模式中

所有模式中，TMPn 的 16 位计数器都从默认值 FFFFH 开始计数。

它从 FFFFH 开始累计至 0000H, 0001H, 0002H, 0003H 等等。

(b) 清除操作

16 位计数器在其值与比较寄存器的值匹配并且被清除时清除为 0000H，也会在该值被捕获且被清除时清除为 0000H。在计数器开始计数后，或当计数器溢出时所发生的从 FFFFH 至 0000H 的计数操作不是清除操作。因此，不会生成 INTTPnCC0 和 INTTPnCC1 中断信号。

(c) 溢出操作

在自由运行定时器模式或脉宽测量模式中，计数器从 FFFFH 减至至 0000H 时，16 位计数器将会发生溢出。如果计数器溢出，TPnOPT0.TPnOVF 位将会被设为 1 并生成一个中断请求信号 (INTTPnOV)。需注意 INTTPnOV 信号在以下条件中不会生成。

- 计数操作开始后。
- 如果计数器的值与比较值 FFFFH 匹配并被清除。
- FFFFH 在脉宽测量模式中被捕获并清除，且计数器从 FFFFH 计数至 0000H 时。

注意事项 溢出中断请求信号 (INTTPnOV) 生成后，务必检测溢出标志 (TPnOVF 位) 是否被置为 1。

(d) 计数操作期间计数器读取操作

在计数操作期间，TMPn 的 16 位计数器的值可以通过使用 TPnCNT 寄存器来读取。TPnCTL0.TPnCE 位 = 1 时，可以通过读取 TPnCNT 寄存器来读取 16 位计数器的值。然而，在 TPnCE 位 = 0 时，16 位计数器的值为 FFFFH，而 TPnCNT 寄存器的值则为 0000H。

(e) 中断操作

TMPn 生成以下三种类型的中断请求信号。

- INTTPnCC0 中断: 该信号的功能相当于 CCR0 缓冲寄存器的匹配中断请求信号以及 TPnCCR0 寄存器的捕获中断请求信号。
- INTTPnCC1 中断: 该信号的功能相当于 CCR1 缓冲寄存器的匹配中断请求信号以及 TPnCCR1 寄存器的捕获中断请求信号。
- INTTPnOV 中断: 该信号的功能相当于一个溢出中断请求信号。

(2) 随时写入和批写入

在定时器操作 (TPnCTL0.TPnCE 位 = 1) 期间, TMPn 中的 TPnCCR0 和 TPnCCR1 寄存器可以被重写, 但 CCR0 和 CCR1 缓冲寄存器的写入方法 (随时写入, 批写入) 会根据模式的不同而变化。

(a) 随时写入

在该模式中, 在定时器操作过程中 (n = 0 至 3) 数据可以随时从 TPnCCR0 和 TPnCCR1 寄存器传送到 CCR0 和 CCR1 缓冲寄存器中。

图 6-5. 随时写入的基本操作流程

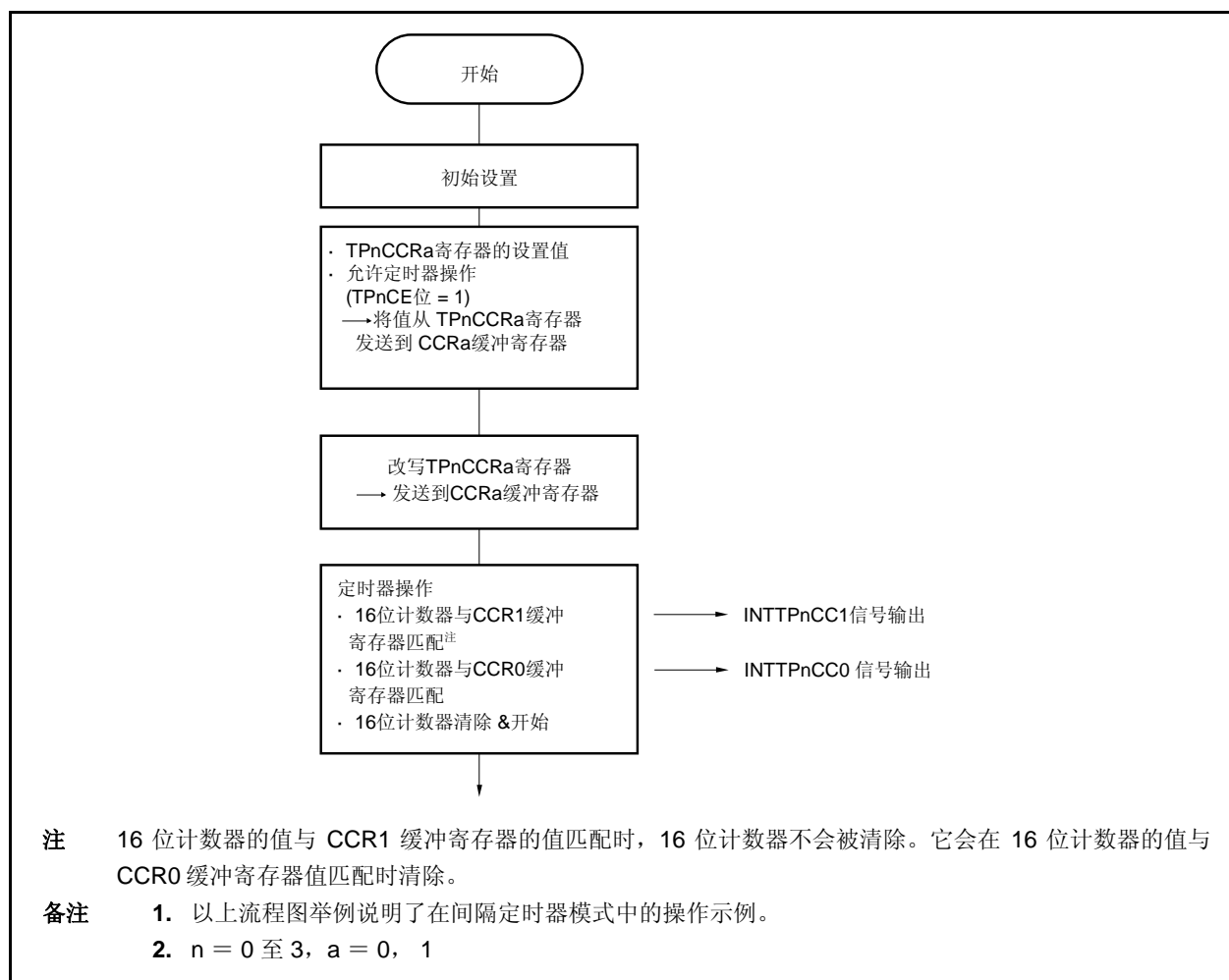
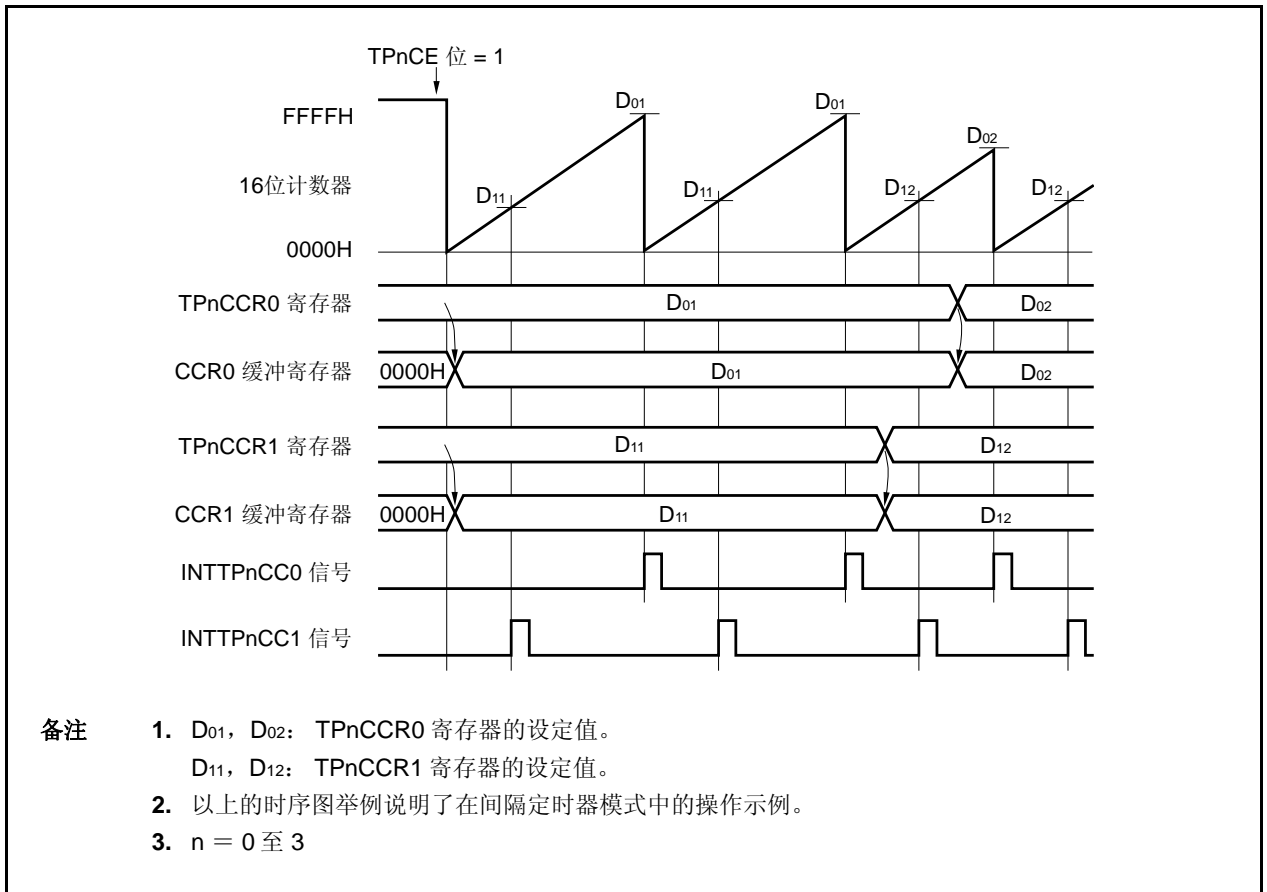


图 6-6. 随时写入的时序



(b) 批写入

在该模式中，在定时器操作过程中所有数据会立即从 TPmCCR0 和 TPmCCR1 寄存器传送到 CCR0 和 CCR1 缓冲寄存器中。该数据会在 16 位计数器的值与 CCR0 缓冲寄存器值匹配时转移。通过对 TPmCCR1 寄存器执行写入操作可以允许转移。

允许还是禁止下个转移时序是通过对 TPmCCR1 寄存器执行或不执行写入操作来控制的。

为了在改写 TPmCCR0 和 TPmCCR1 寄存器时将设定值转变为 16 位计数器的比较值（换句话说，为了让该值转移到 CCR0 和 CCR1 缓冲寄存器中），在 16 位计数器的值与 CCR0 寄存器的值匹配前，需要先对 TPmCCR0 寄存器进行改写，然后再对 TPmCCR1 寄存器进行写入操作。因此，一旦 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配，TPmCCR0 和 TPmCCR1 寄存器的值将会被转移到 CCR0 和 CCR1 缓冲寄存器中。所以即使只想改写 TPmCCR0 寄存器的值，也需向 TPmCCR1 寄存器写入相同的值（与 TPmCCR1 寄存器的预置值相同）

图 6-7. 批写入的基本操作流程

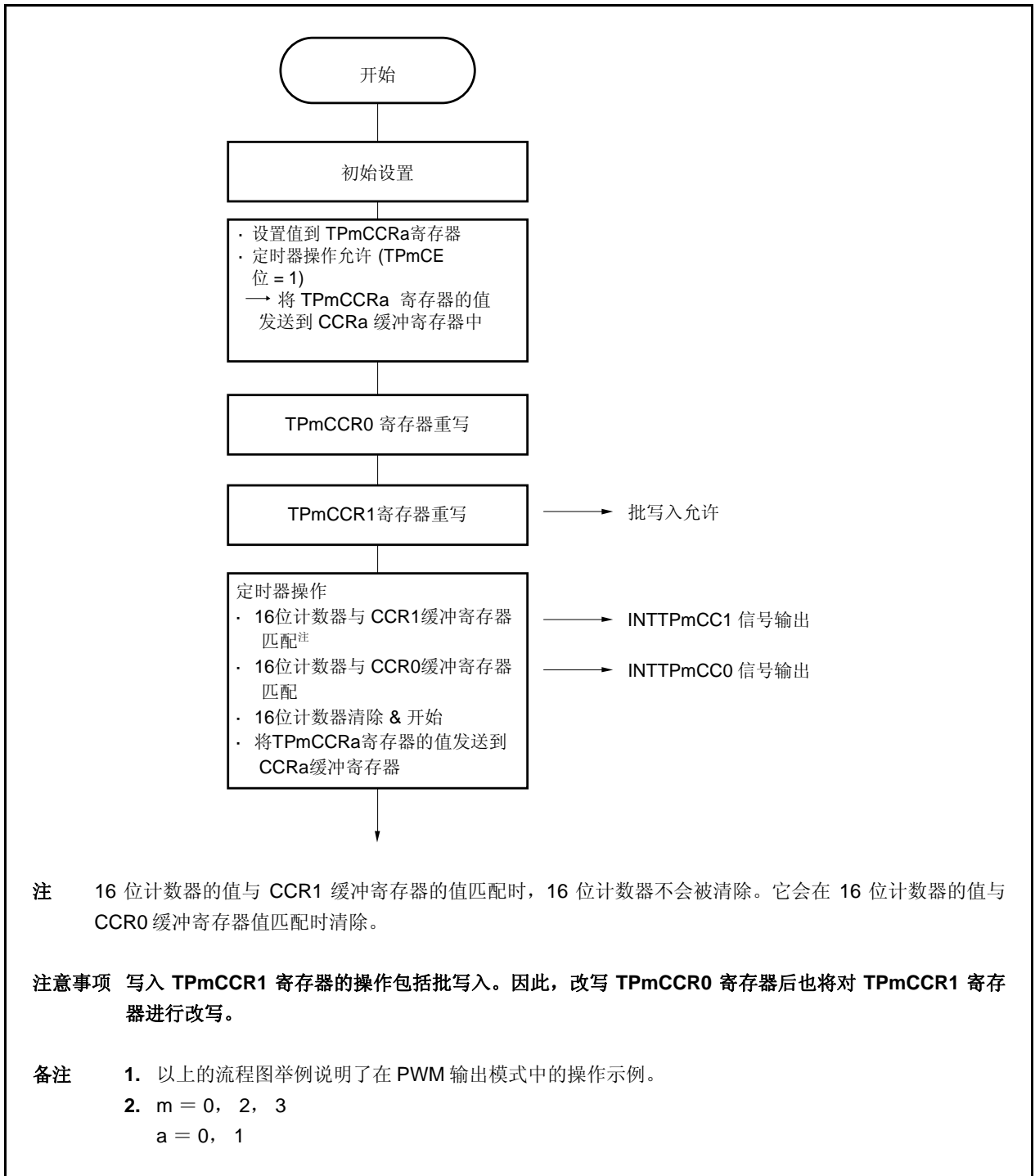
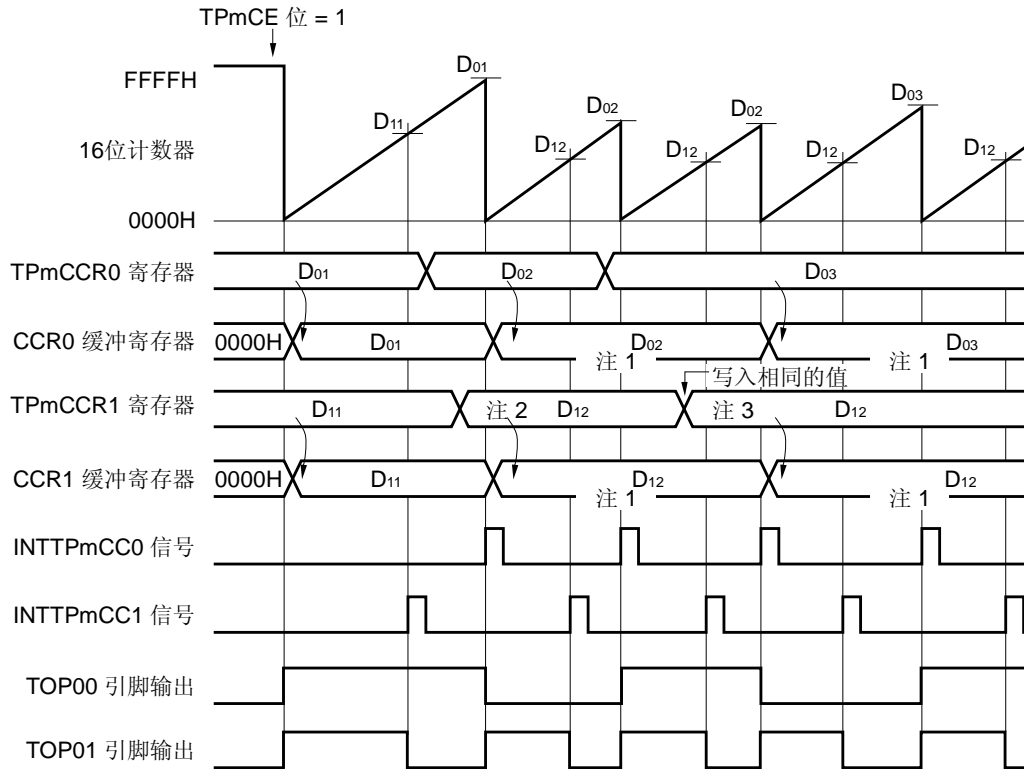


图 6-8. 批写入的时序



- 注**
1. 由于 TPmCCR1 寄存器没有被改写，因此 D03 不会被转移。
 2. 由于 TPmCCR1 寄存器已经被写入 (D12)，因此，一旦 16 位计数器的值与 TPmCCR0 寄存器 (D01) 的值匹配，数据将会被转移到 CCR1 缓冲寄存器中。
 3. 由于 TPmCCR1 寄存器已经被写入 (D12)，因此，一旦 16 位计数器的值与 TPmCCR0 寄存器 (D02) 的值匹配，数据将会被转移到 CCR1 缓冲寄存器中。

- 备注**
1. D01, D02, D03: TPmCCR0 寄存器的设定值。
D11, D12: TPmCCR1 寄存器的设定值。
 2. 以上的时序图举例说明了在 PWM 输出模式中的操作示例。
 3. m = 0, 2, 3

6.6.1 间隔定时器模式 (TPnMD2 至 TPnMD0 位=000)

在间隔定时器模式中, 如果 TPnCTL0.TPnCE 位被设为 1, 中断请求信号 (INTTPnCC0) 将会在 TPnCCR0 寄存器所设置的间隔中生成。可以从 TOP00 引脚中输出 (仅限 TMP0) 周期是间隔时间的 2 倍的 50% 占空比的 PWM 波形。

间隔定时器模式中不使用 TPnCCR1 寄存器。然而, TPnCCR1 寄存器的设定值将会转移到 CCR1 缓冲寄存器中, 而在 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配时将会输出一个比较匹配中断请求信号 (INTTPnCC1)。此外, 在 INTTPnCC1 信号生成时反转的 50% 占空比的 PWM 波形可以从 TOPm1 引脚中输出。

即便在定时器运行时, TPnCCR0 和 TPnCCR1 寄存器也可以进行改写。

备注 m = 0, 2, 3

图 6-9. 间隔定时器的配置

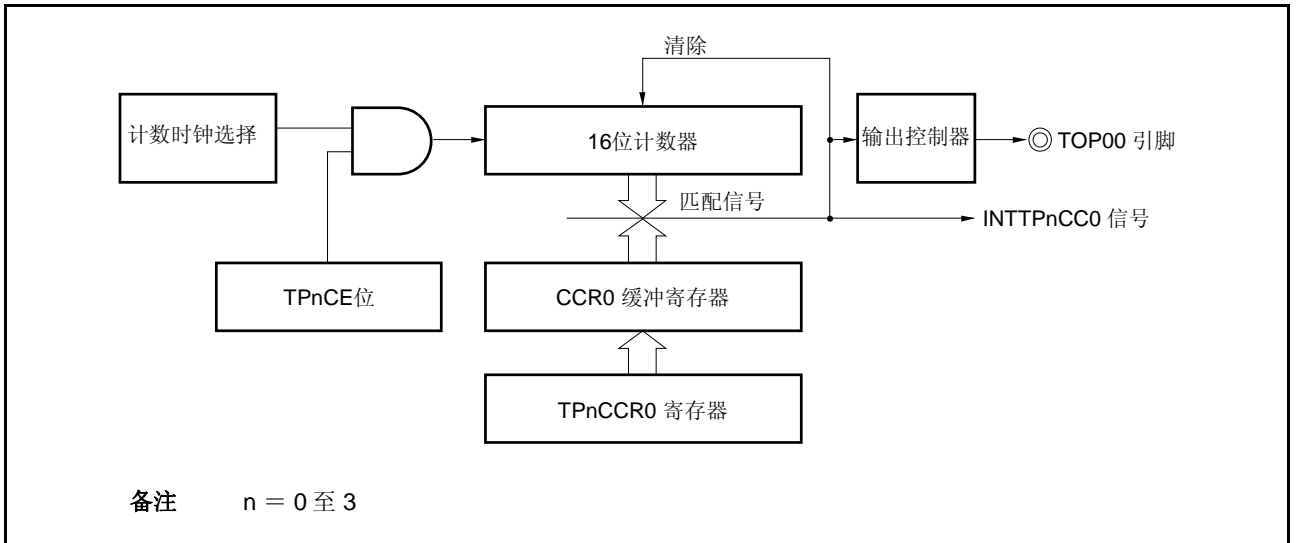
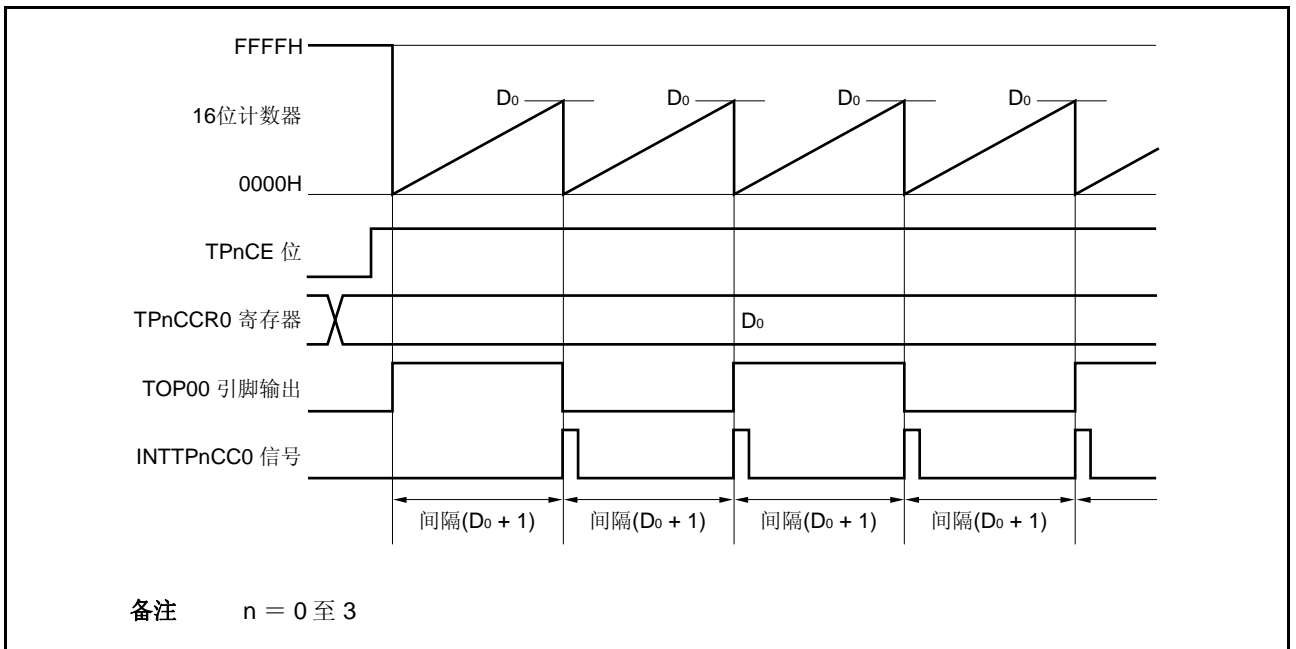


图 6-10. 间隔定时器工作的基本时序



在 TPnCE 位被置 1 时，16 位计数器将与计数时钟同步从 FFFFH 清零为 0000H 并开始计数。此时，TOP00 引脚的输出电平被反转。此外，TPnCCR0 寄存器的设定值传送到 CCR0 缓冲寄存器中。

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，TOP00 引脚输出反相，并生成一个比较匹配中断请求信号 (INTTPnCC0)。

时间间隔可用下式计算。

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

备注 n = 0 至 3

图 6-11. 间隔定时器模式操作时的寄存器设置 (1/3)

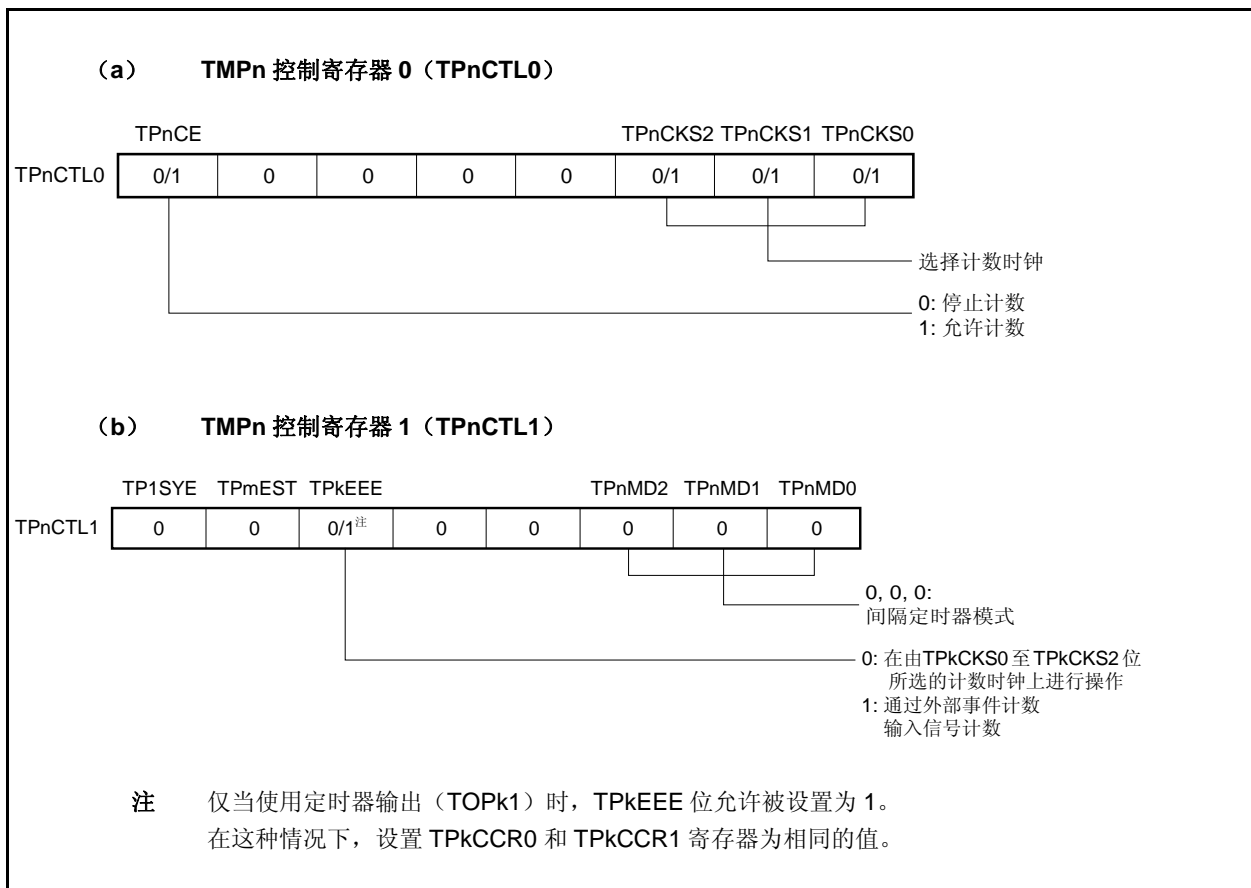
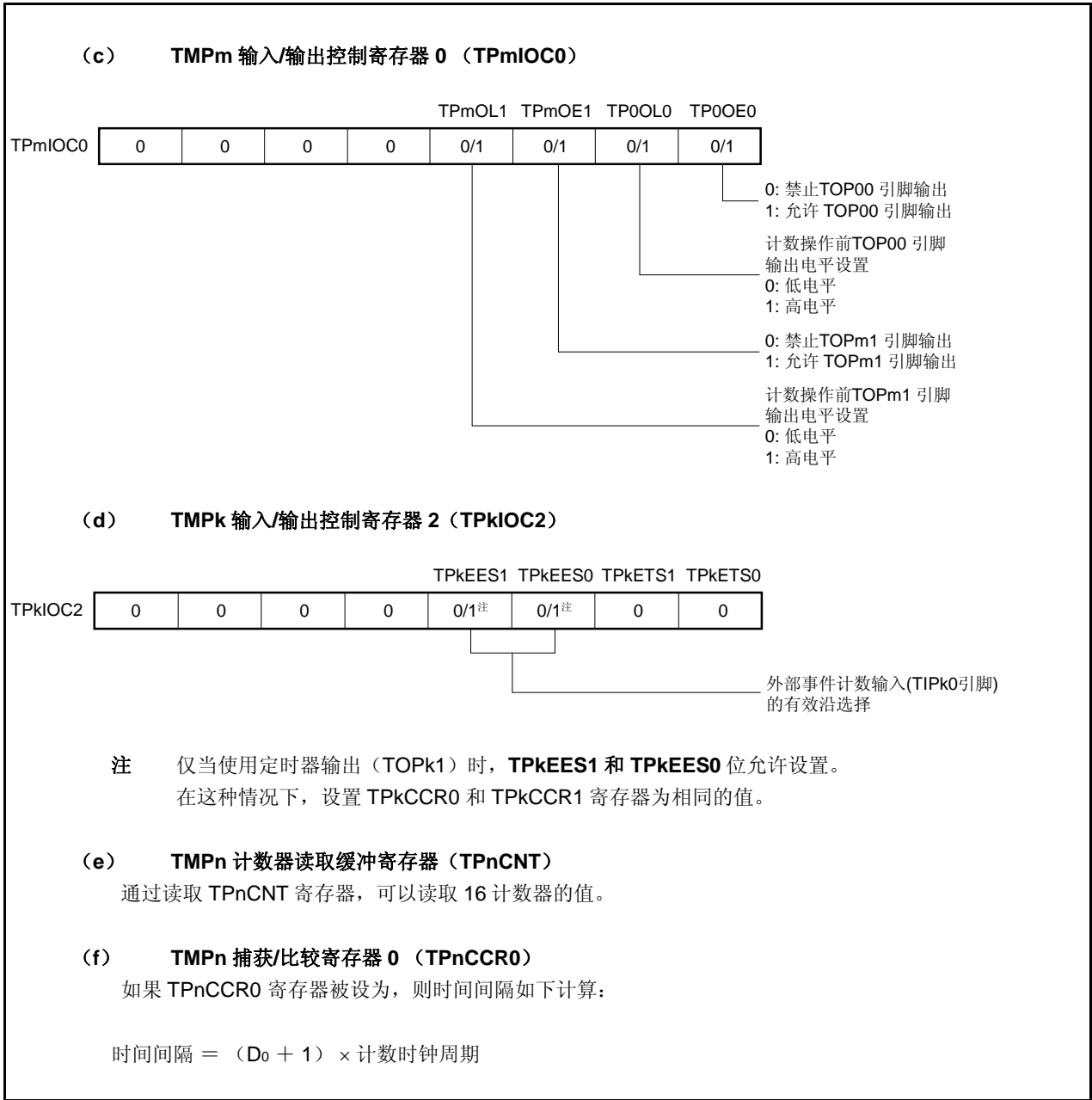


图 6-11. 间隔定时器模式操作时的寄存器设置 (2/3)



<R>

图 6-11. 间隔定时器模式操作时的寄存器设置 (3/3)

(g) TMPn 捕获/比较寄存器 1 (TPnCCR1)

间隔定时器模式中不使用 TPnCCR1 寄存器。但对 TPnCCR1 寄存器设置的数值也将被传送到 CCR1 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 缓冲寄存器中的数值匹配时，TOPm1 引脚输出将会被反转并且会产生一个比较匹配中断请求信号 (INTTPnCC1)。

通过把与 TPmCCR0 寄存器中所设定值相同的值设置到该寄存器中，带有 50% 占空比的 PWM 波形将从 TOPm1 引脚中输出。

TPnCCR1 寄存器没有被使用时，建议将其值设为 FFFFH。此外，还会通过中断屏蔽标志 (TPnCCIC1.TPnCCMK1) 来屏蔽寄存器。

- <R> **备注**
1. 在间隔定时器模式时，不使用 TMPk 输入/输出控制寄存器 1 (TPkIOC1) 和 TMPn 选择寄存器 0 (TPnOPT0)。
 2. n = 0 至 3,
m = 0, 2, 3
k = 0, 2

(1) 间隔定时器模式工作流程

图 6-12. 间隔定时器模式时的软件处理流程 (1/2)

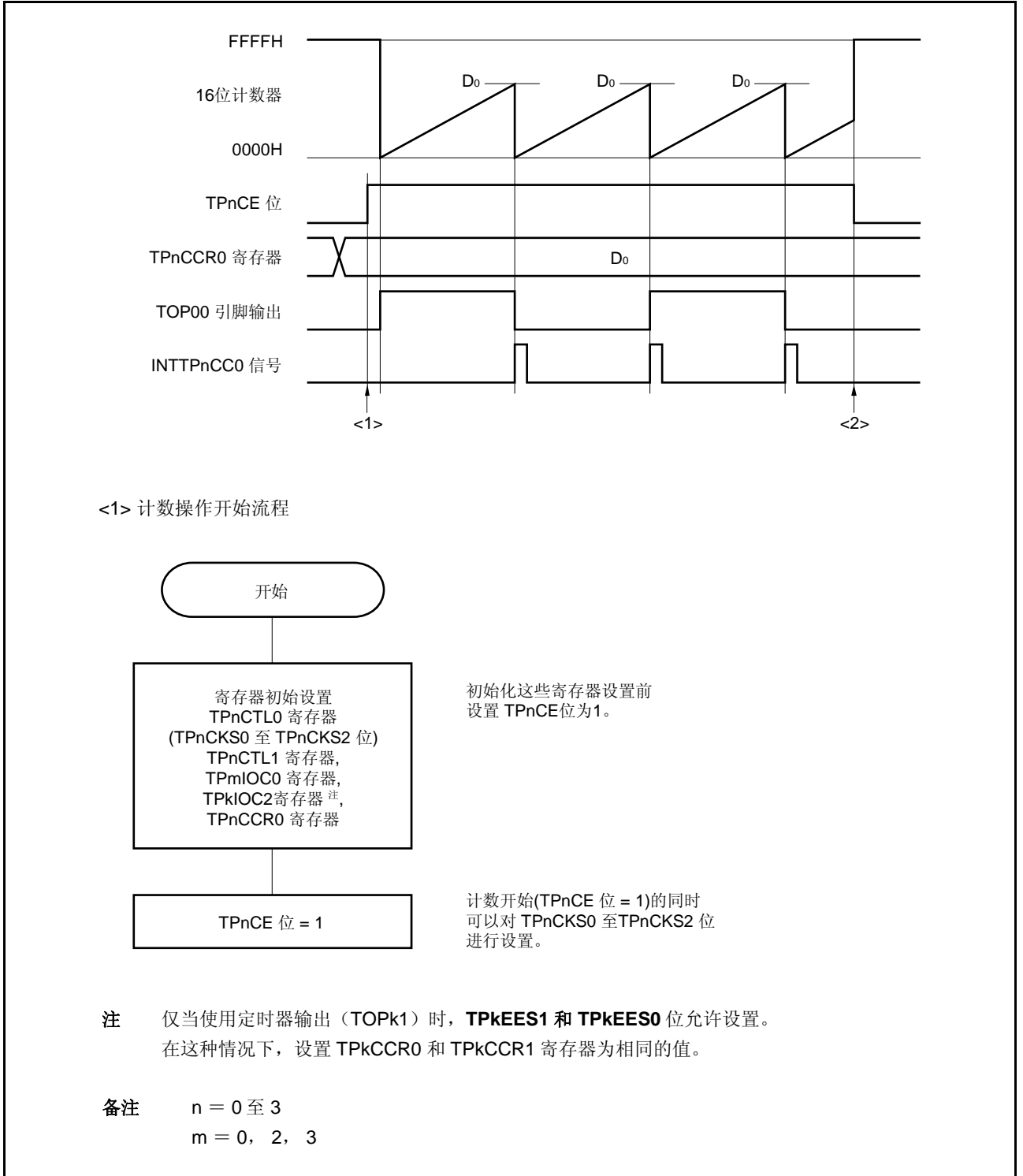
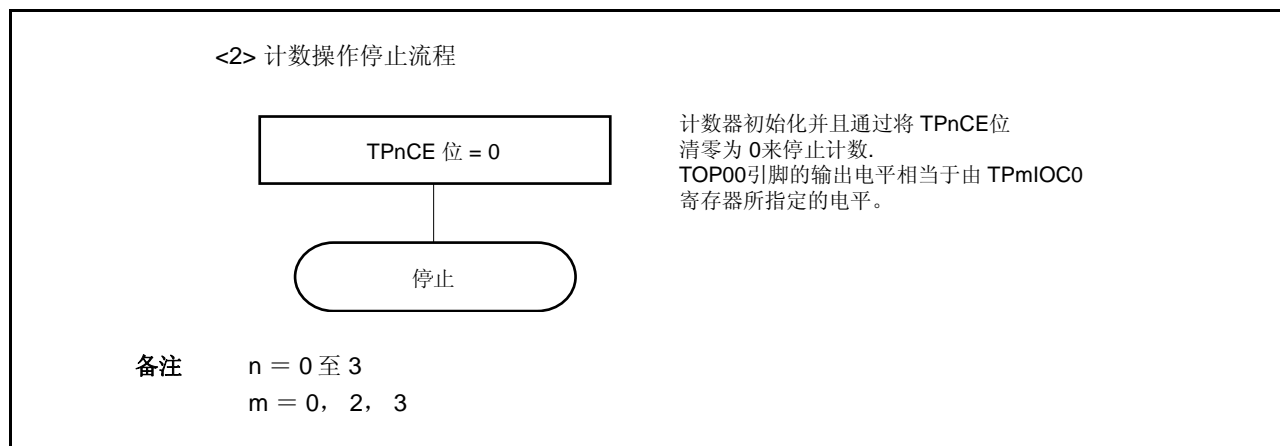


图 6-12. 间隔定时器模式时的软件处理流程 (2/2)

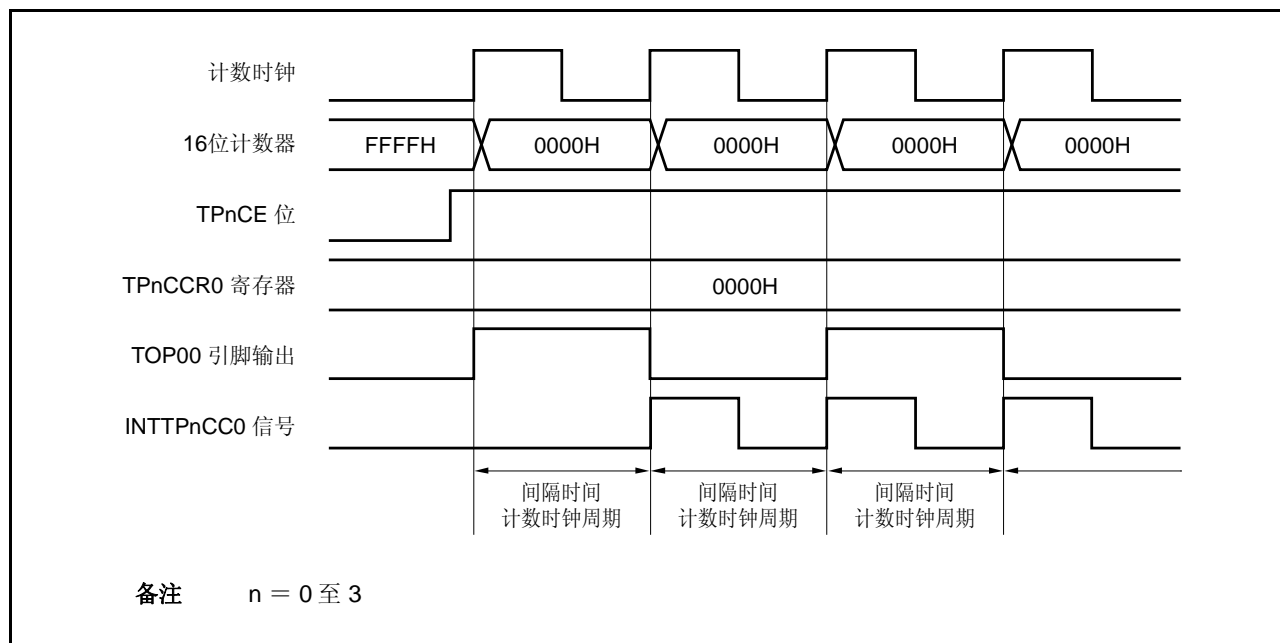


(2) 间隔定时器模式工作时序

(a) 如果 TPnCCR0 寄存器被设置为 0000H 时的操作

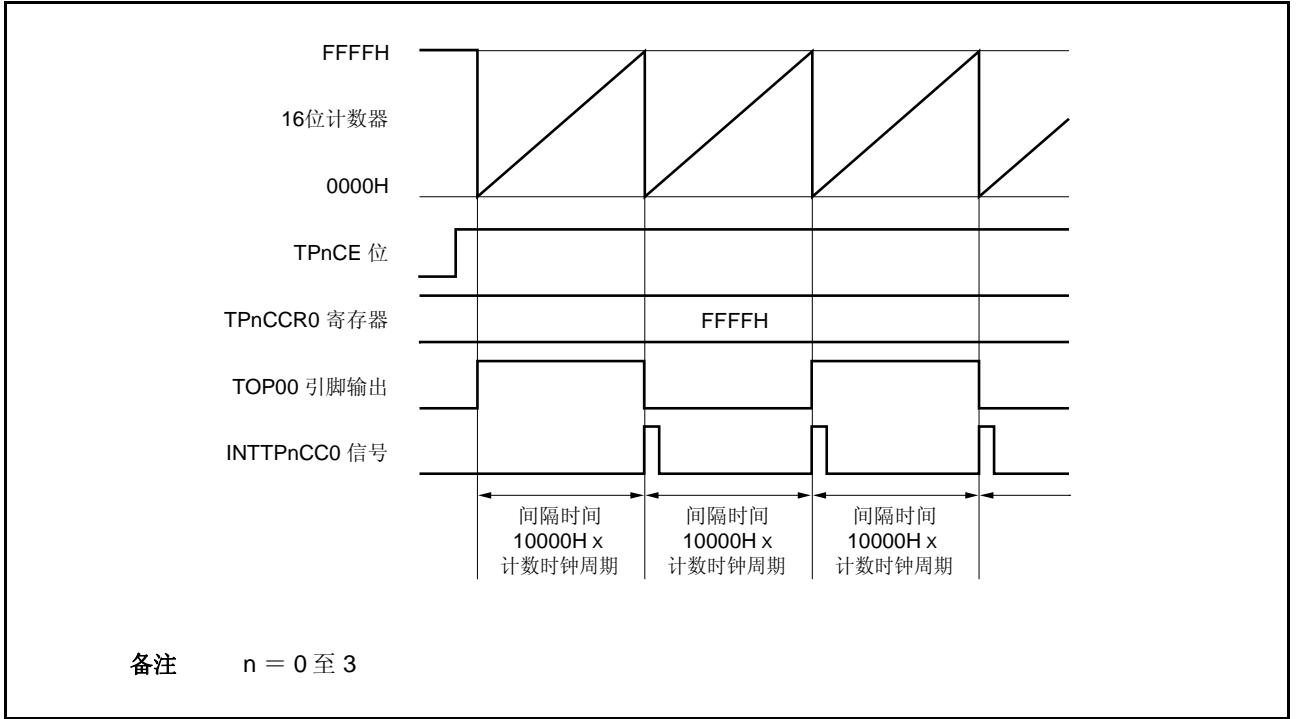
如果 TPnCCR0 寄存器置为 0000H，每个计数时钟都会产生 INTTPnCC0 信号，而且，TOP00 引脚的输出反相。

16 位计数器的值总是 0000H。



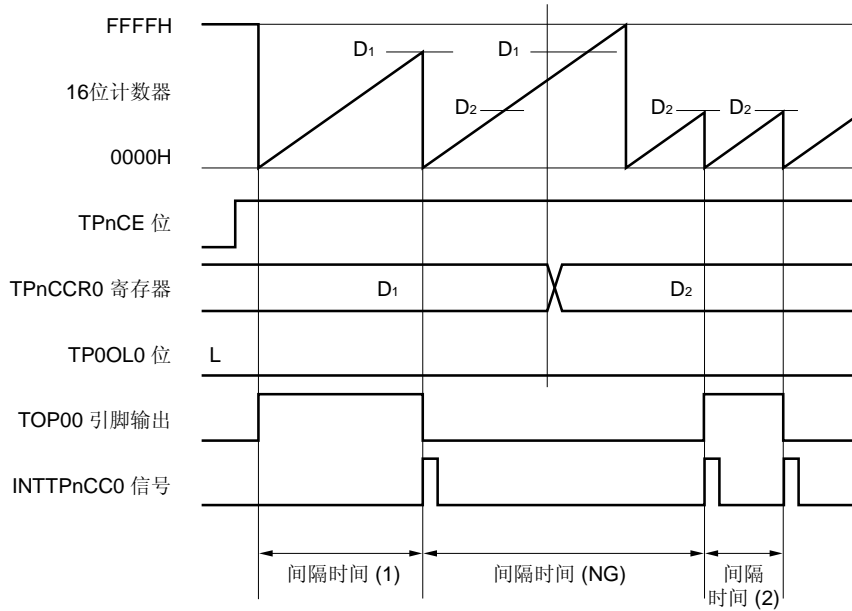
(b) TPnCCR0 寄存器设置为 FFFFH 时的操作

如果 TPnCCR0 寄存器被设置为 FFFFH，则 16 位计数器可以计数直至 FFFFH。并在下一个计数时钟时刻同步被清除为 0000H。同时产生 INTTPnCC0 信号，并且 TOP00 引脚输出反相。此时，并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。



(c) 重新写入 TPnCCR0 寄存器的注意事项

如果在计数过程中将 TPnCCR0 寄存器的值减小，那么 16 位计数器可能产生溢出。当可能发生溢出时，停止计数再更改设定值。

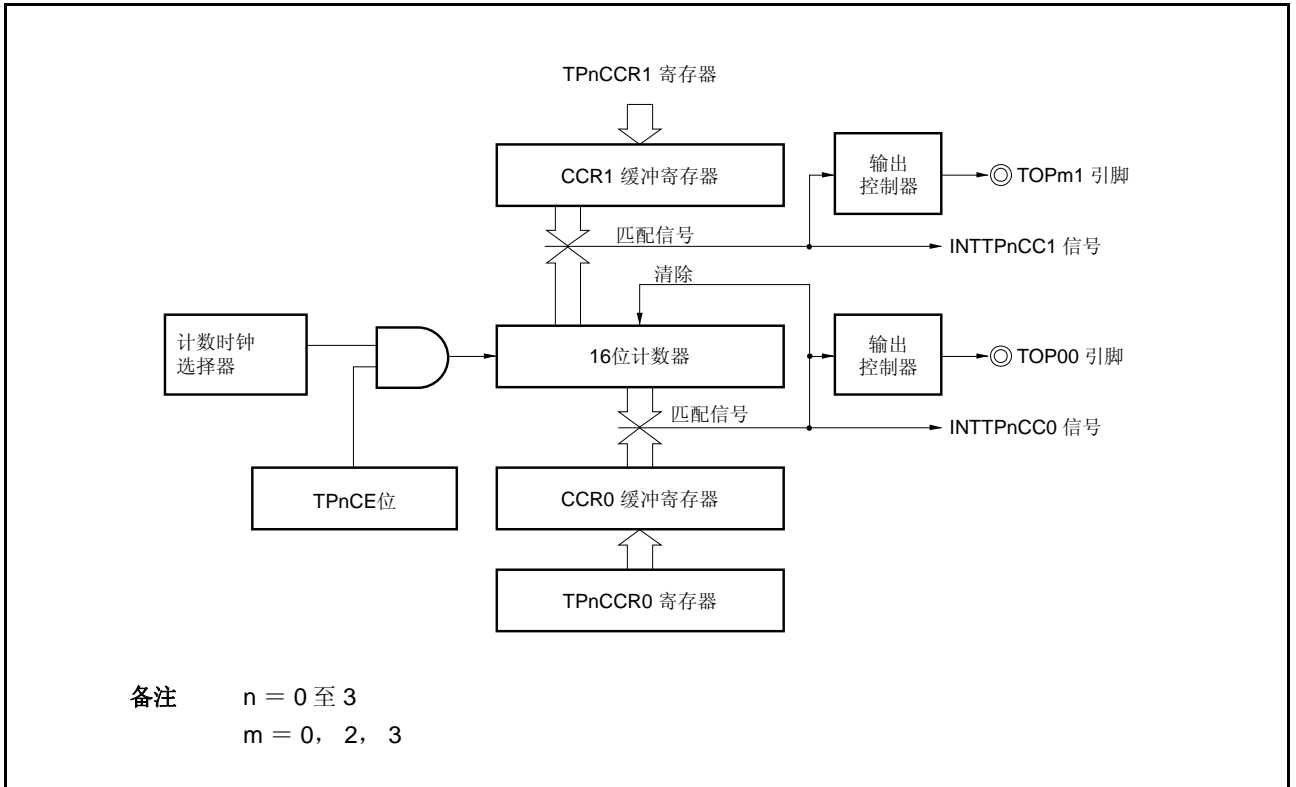


- 备注**
1. 间隔时间 (1) : $(D_1 + 1) \times$ 计数时钟周期
 间隔时间 (NG) : $(10000H + D_2 + 1) \times$ 计数时钟周期
 间隔时间 (2) : $(D_2 + 1) \times$ 计数时钟周期
 2. $n = 0$ 至 3

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 ，则在重写 TPnCCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。结果，用来比较的 16 位计数器的值是 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 直到溢出，然后从 0000H 开始重新计数。当计数值和 D_2 相符时，会生成 INTTPnCC0 信号，且 TOP00 引脚输出信号反相。因此，INTTPnCC0 信号可能不会在原先预计的间隔时间“ $(D_1 + 1) \times$ 计数时钟周期”或“ $(D_2 + 1) \times$ 计数时钟周期”下产生，而会以“ $(10000H + D_2 + 1) \times$ 计数时钟周期”的间隔产生。

(d) TPnCCR1 寄存器的操作

图 6-13. TPnCCR1 寄存器的配置



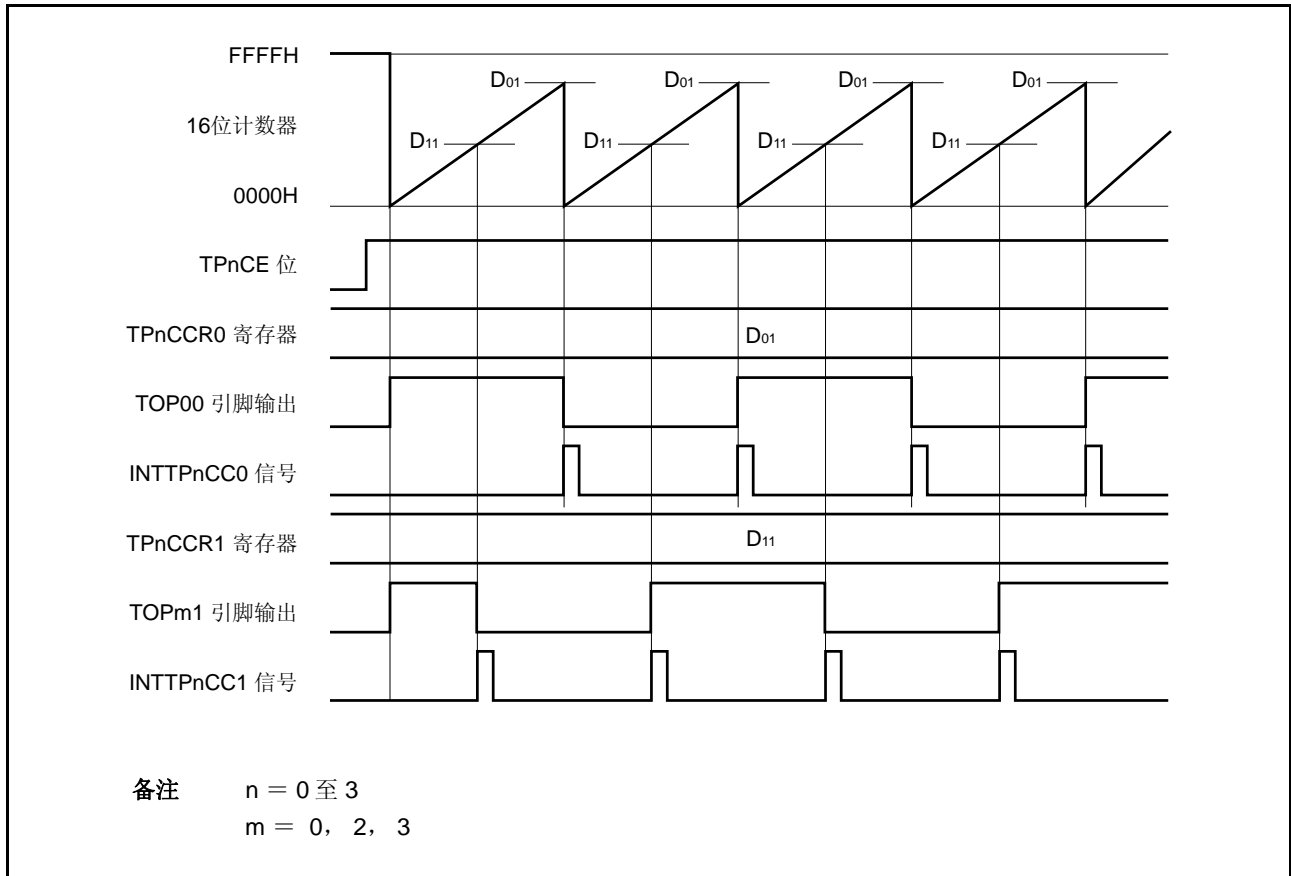
当 TPnCCR1 寄存器的值被设为与 TPnCCR0 寄存器相同的值时，INTTPnCC1 信号将会在 INTTPnCC0 信号与 TOPm1 引脚输出被反转的同时被生成。换句话说，占空比为 50% 的 PWM 波形可以从 TOPm1 引脚中输出。

以下显示了将不同于 TPnCCR0 寄存器中所设定的值设置至 TPnCCR1 寄存器时的操作。

如果 TPnCCR1 寄存器的设置值小于 TPnCCR0 寄存器的设置值，INTTPnCC1 信号每计数时钟周期产生一次，同时，TOPm1 引脚输出反相。

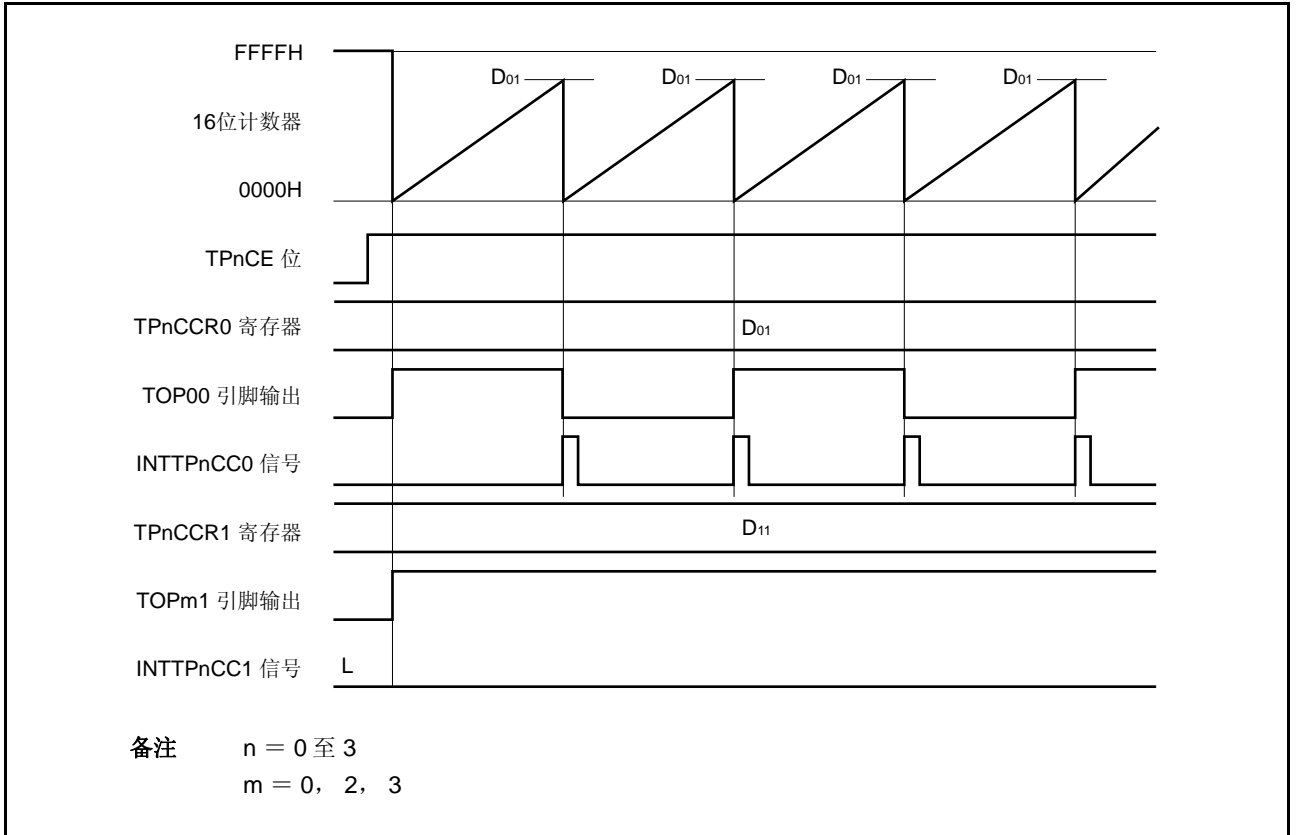
在输出一个短宽脉冲后，TOPm1 引脚将会输出一个 50% 占空比的 PWM 波形。

图 6-14. 当 $D_{01} \geq D_{11}$ 时的时序图



如果 TPnCCR1 寄存器的设置值大于 TPnCCR0 寄存器的设置值，16 位计数器的计数值不能与 TPnCCR1 寄存器的值相匹配，因而，不会产生 INTTPnCC1 信号，TOPm1 引脚输出也不会改变。
 TPnCCR1 寄存器没有被使用时，建议将其值设为 FFFFH。

图 6-15. 当 $D_{01} < D_{11}$ 时的时序图



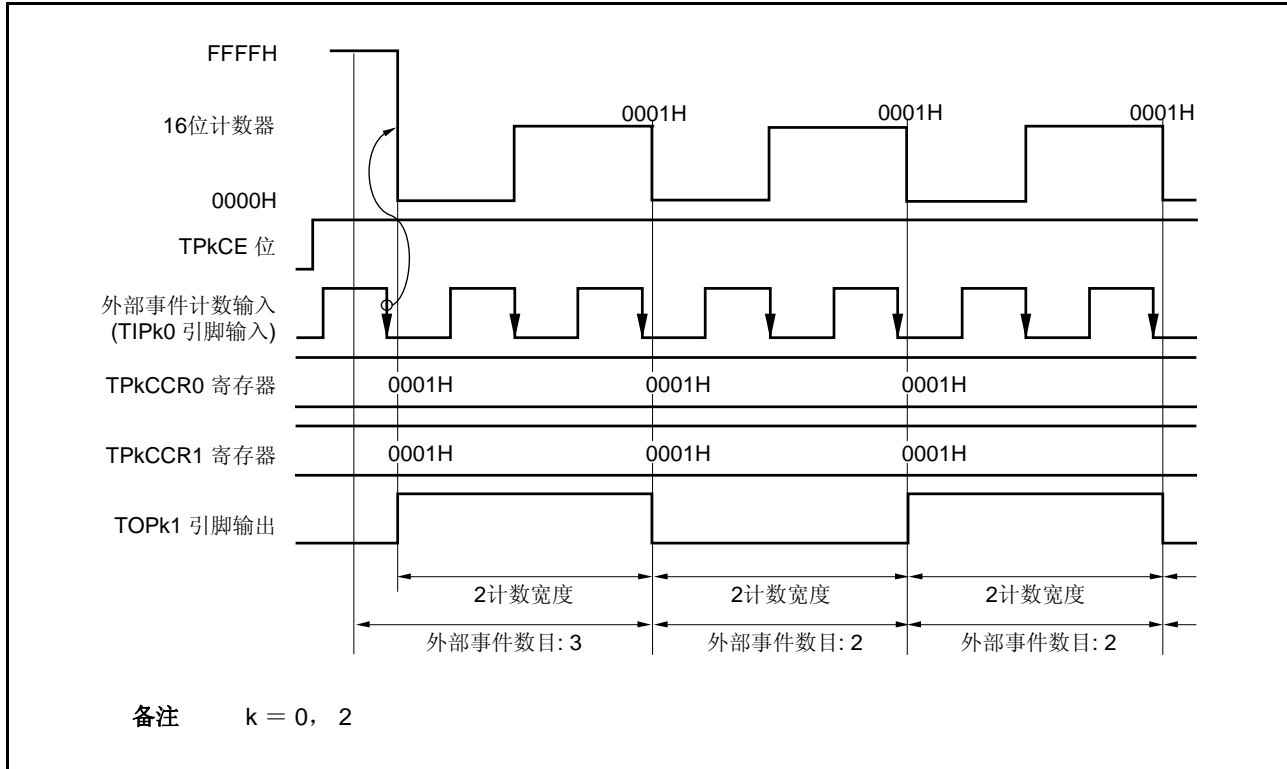
<R> (3) 外部事件计数输入引脚 (TIPk0) 的操作

(a) 操作

在间隔定时器模式下, 来计数 16 位计数器在外部事件计数输入 (TIPk0) 的有效沿, 在 TPkCE 位的设置从 0 到 1 后, 在第一个外部事件计数输入的有效沿清除 16 位计数器从 FFFFH 至 0000H。

当 TPkCCR0 和 TPkCCR1 寄存器都设置为 0001H 时, 16 位计数器每计数两次 TOPk1 引脚的输出反转一次。

在间隔定时器模式下, 仅当定时器的输出 (TOPk1) 用作外部事件计数输入时, TPkCTL1.TPKEEE 位能被设置为 1。



6.6.2 外部事件计数器模式 (TPkMD2 至 TPkMD0 位=001)

该模式仅在 TMP0 和 TMP2 中有效。

在外部事件计数模式，只有当 TPkCTL0.TPkCE 位被设为 1 时外部事件计数输入 (TIPk0) 的有效沿才会被计数，且每次计数到 TPkCCR0 寄存器所设置边沿数量后都将会产生中断请求信号 (INTTPkCC0)。TOP00 和 TOPk1 引脚不能被使用。在外部事件计数输入模式下使用 TOPk1 引脚时，在间隔定时器模式下设置 TPkCTL1.TPkEEE 位为 1 (参见 6.6.1 外部事件计数输入引脚(TIPk0)的操作)。

<R>

在外部事件计数器模式中不使用 TPkCCR1 寄存器。

注意事项 在外部事件计数模式下，TPkCCR0 和 TPkCCR1 寄存器不能设置为 0000H。

图 6-16. 外部事件计数器模式的配置

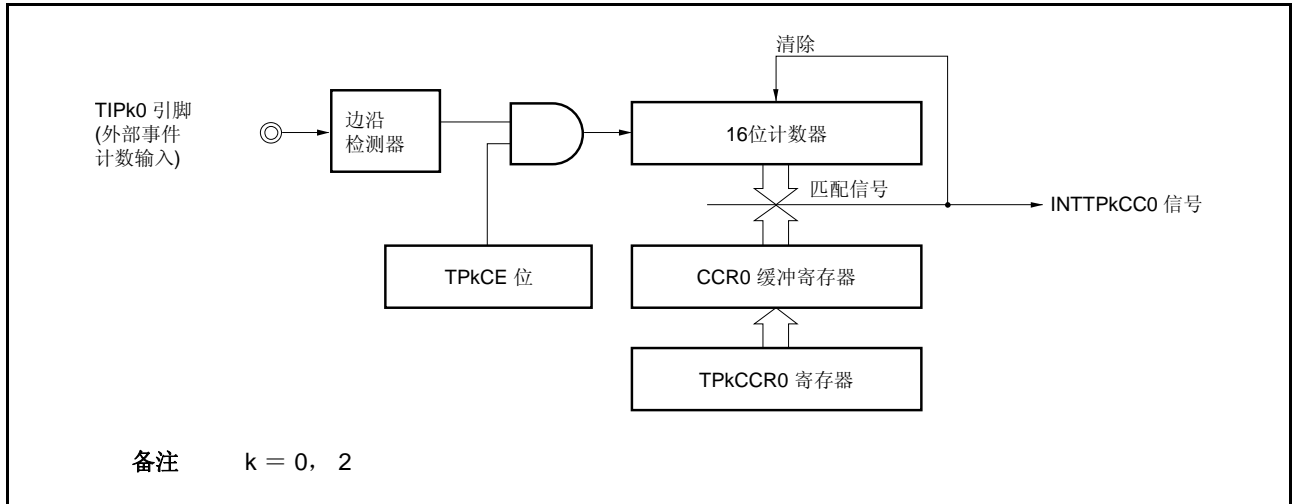
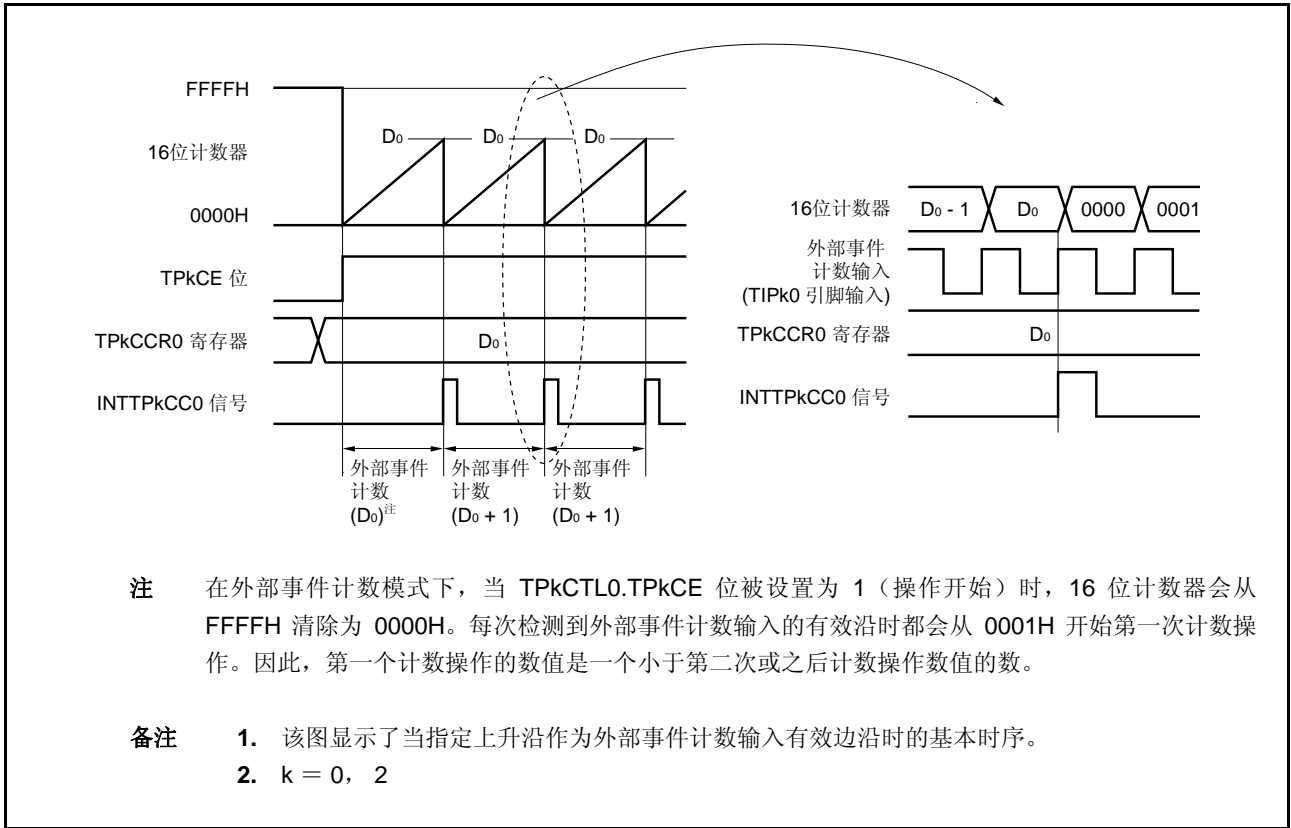


图 6-17. 外部事件计数器模式的基本时序



当 **TPkCE** 位设置为 1 时，16 位计数器的值由 **FFFFH** 被清为 **0000H**，计数器在每次检测到外部事件计数输入信号的有效边沿时进行一次计数。此外，**TPkCCR0** 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 (**INTTPkCC0**)。

当外部事件计数输入的有效沿被检测的次数等于“**TPkCCR0** 寄存器中所设置的值”时，将会第一次生成 **INTTPkCC0** 信号。在此之后，每当检测到“**TPkCCR0** 寄存器设置值 + 1”次外部事件计数的有效沿时，都会产生 **INTTPkCC0** 信号。

图 6-18. 外部事件计数模式下的寄存器操作设置 (1/2)

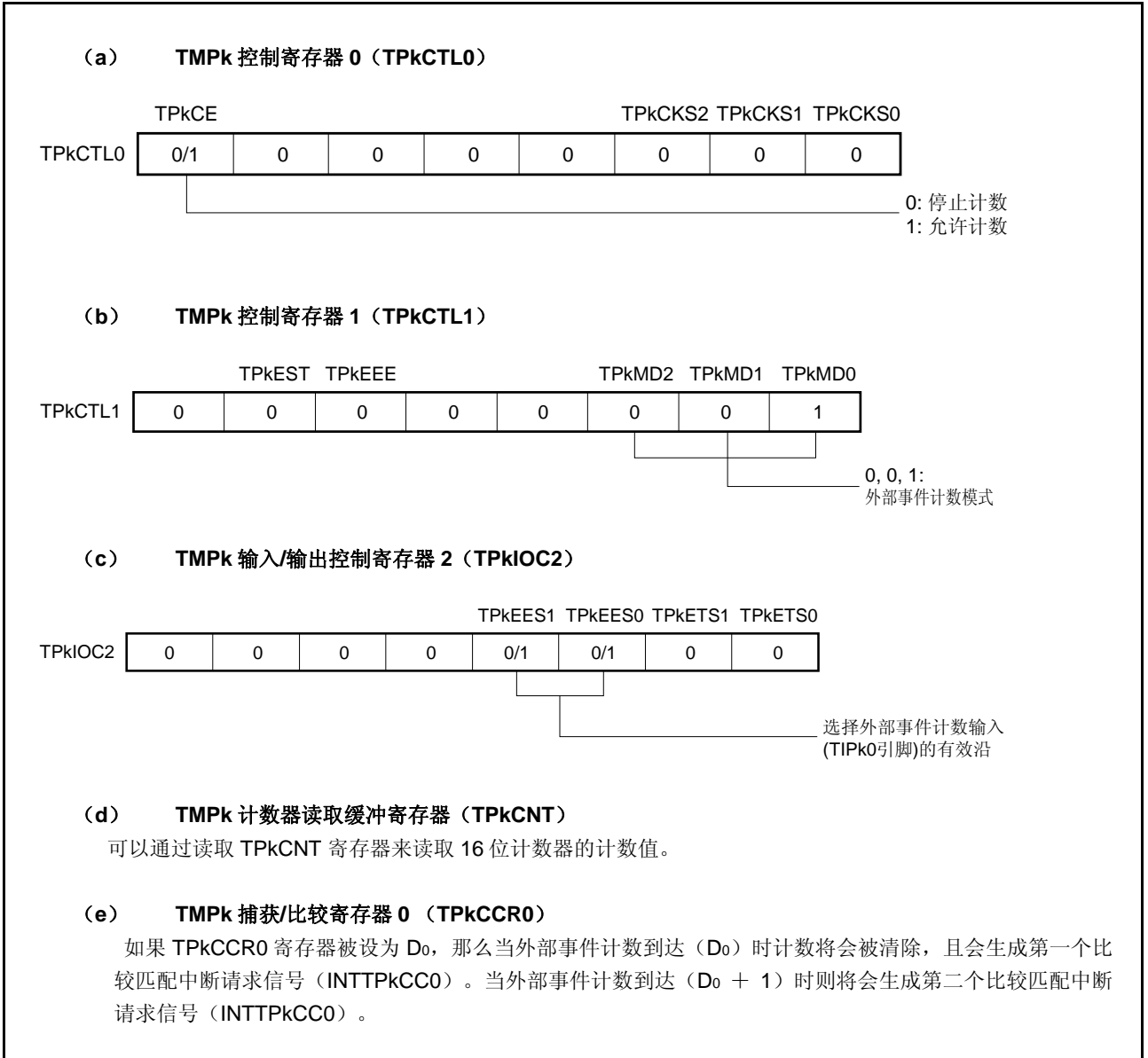


图 6-18. 外部事件计数模式下的寄存器操作设置 (2/2)

(f) TMPk 捕获/比较寄存器 1 (TPkCCR1)

外部事件计数模式时不使用 TPkCCR1 寄存器，然而，TPkCCR1 寄存器的设定值仍然会传送到 CCR1 缓冲寄存器中。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 (INTTPkCC1)。

TPkCCR1 寄存器没有被使用时，建议将其值设为 FFFFH。此外，通过中断屏蔽标志 (TPkCCIC1.TPkCCMK1) 屏蔽寄存器。

<R>

<R>

注意事项 1. 设置 TPkIOC0 寄存器为 00H。

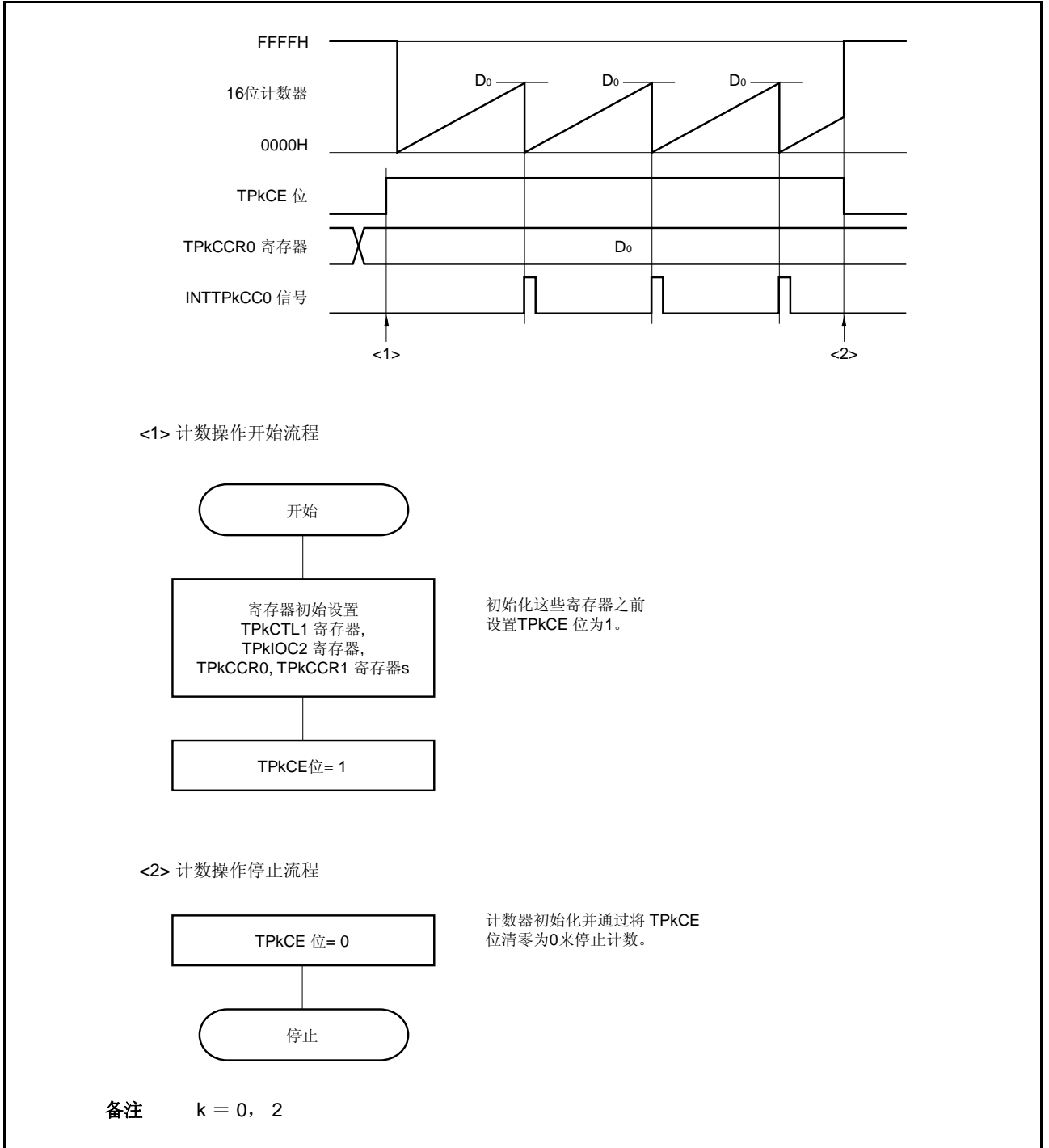
2. 当外部时钟用作计数时钟时，外部时钟仅能从 TIPk0 引脚输入。此时，把 TPkIOC1.TPkIS1 和 TPkIOC1.TPkIS0 位设置为 00 (捕获触发输入 (TIPk0 引脚)：无边沿检测)。

备注 1. 在外部事件计数模式下，并未使用 TMPk 输入/输出控制寄存器 0 (TPkIOC0)，TMPk 输入/输出控制寄存器 1 (TPkIOC1) 以及 TMPn 选项寄存器 0 (TPkOPT0)。

2. k = 0, 2

(1) 外部事件计数器模式的操作系统

图 6-19. 外部事件计数器模式下的软件处理流程



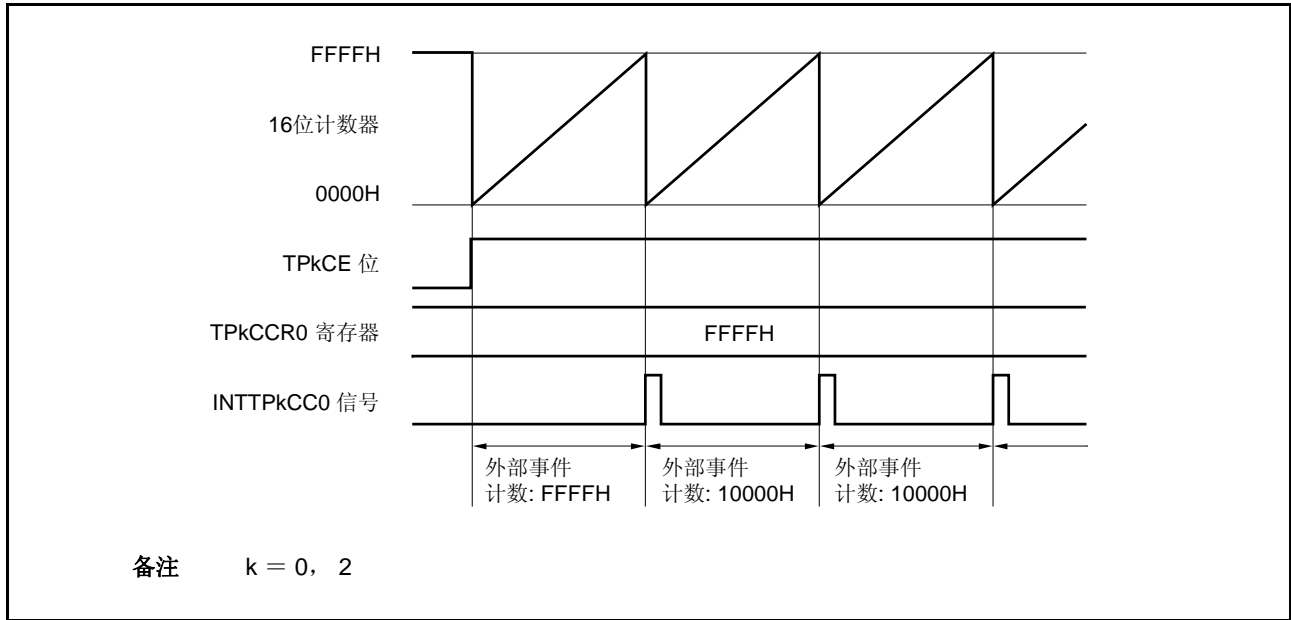
(2) 外部事件计数器模式的工作时序

注意事项 1. 在外部事件计数模式下，TPkCCR0 和 TPkCCR1 寄存器不能清零为 0000H。

<R> 2. 在外部事件计数模式下，定时器输出 (TOP00, TOPk1) 使用被禁止。如果使用定时器输出 (TOPk1) 与外部事件计数输入 (TIPk0)，设置间隔定时器模式，并且允许计数时钟操作与外部事件计数输入 (TPkCTL1.TPKEEE 位 = 1) (参见 6.6.1 (3) 外部事件计数输入引脚(TIPk0) 的操作)。

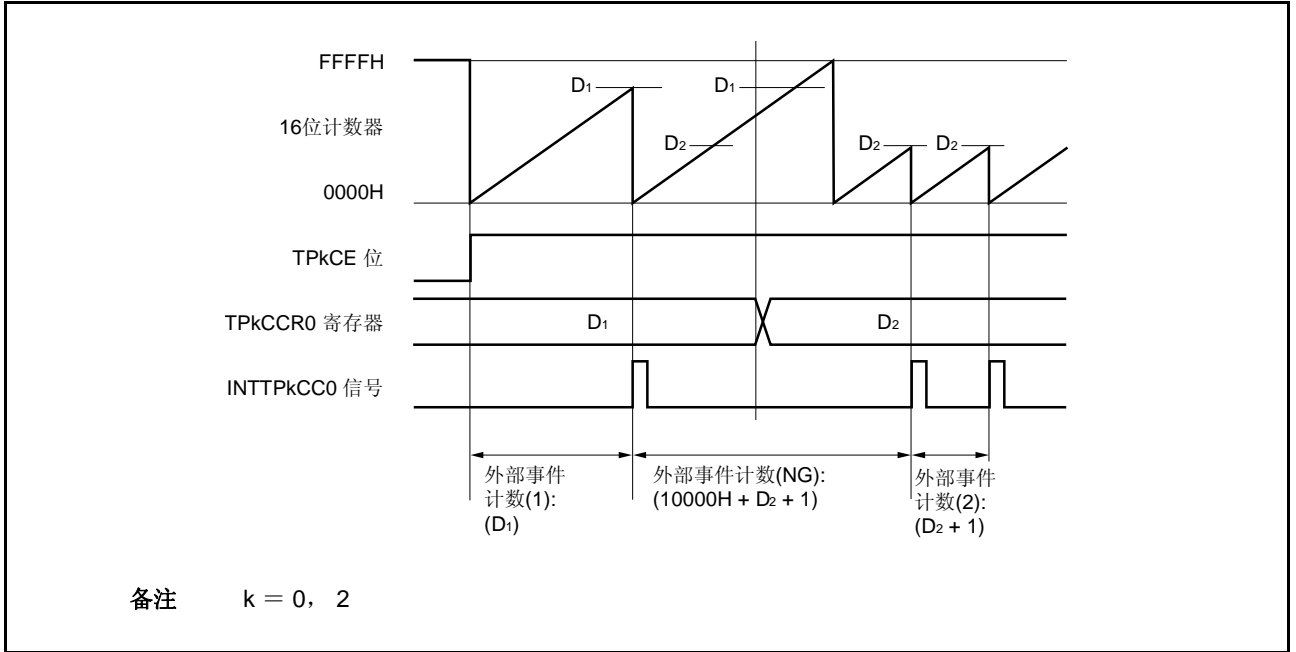
(a) TPkCCR0 寄存器设置为 FFFFH 时的操作

如果 TPkCCR0 寄存器被设置为 FFFFH，16 位计数器每次计数是在检测到的外部事件输入信号有效边沿时，直到计数达到 FFFFH，然后，16 位计数器于下一次计数时同步清为 0000H，同时，产生 INTTPkCC0 信号。此时，不设置 TPkOPT0.TPKOVF 位。



(b) 关于重写 TPkCCR0 寄存器时的注意事项

如果在计数过程中将 TPkCCR0 寄存器的值减小，那么 16 位计数器可能产生溢出。当发生溢出时，将停止一次计数并更改设定值。

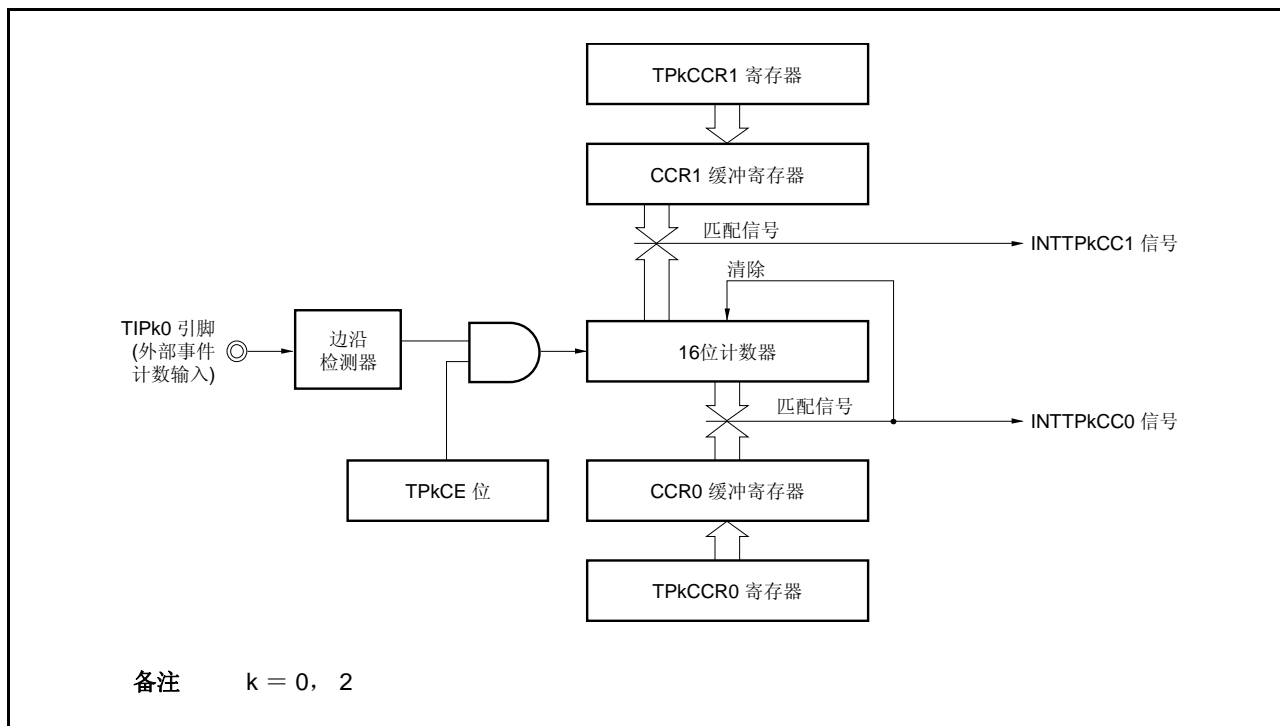


当 16 位计数器计数值大于 D₂ 而小于 D₁ 时，如果将 TPkCCR0 寄存器的值从 D₁ 变为 D₂，则在重写 TPkCCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。因而，与 16 位计数器相比较的值会是 D₂。但是，由于计数值已经超过了 D₂，所以 16 位计数器就会一直向上累积计数到 FFFFH 直到溢出，然后从 0000H 开始重新计数。当计数值与 D₂ 匹配时，会生成 INTTPkCC0 信号。

因此，INTTPkCC0 中断请求信号可能不会在原先设定的“(D₁ + 1) 次”或“(D₂ + 1) 次”有效边缘上生成，而可能在“(10000H + D₂ + 1) 次”有效边缘上生成。

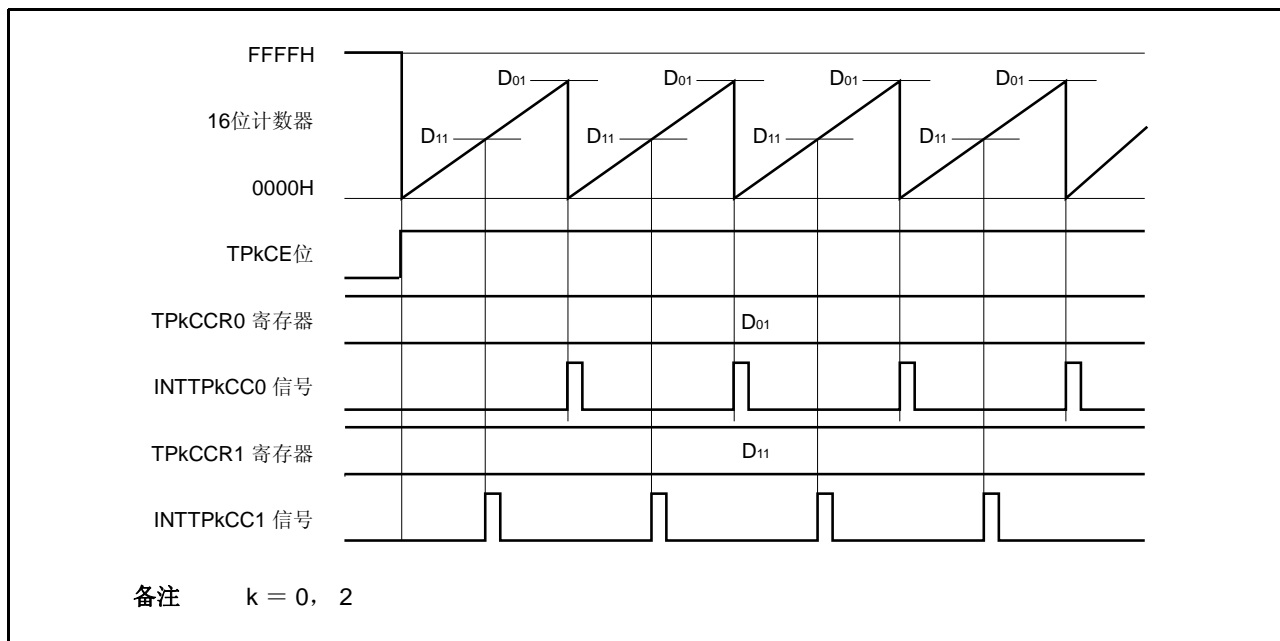
(c) TPkCCR1 寄存器的操作

图 6-20. TPkCCR1 寄存器的配置



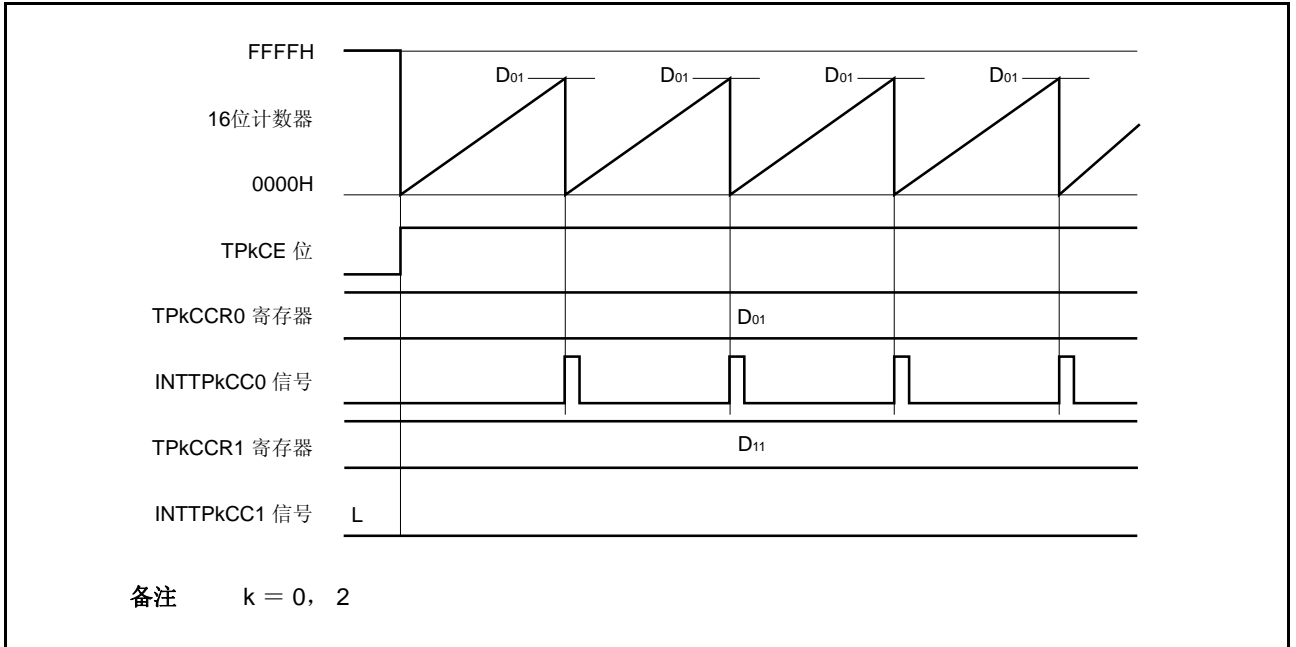
如果 TPkCCR1 寄存器的设定值小于 TPkCCR0 寄存器的设定值，则 INTTPkCC1 中断请求信号每个计数周期产生一次。

图 6-21. 当 $D_{01} \geq D_{11}$ 时的时序图



如果 TPkCCR1 寄存器的设定值大于 TPkCCR0 寄存器的设定值，则不会产生 INTTPkCC1 中断请求信号，因为 16 位计数器的计数值和 TPkCCR1 寄存器的值不匹配。
 TPkCCR1 寄存器没有被使用时，建议将其值设为 FFFFH。

图 6-22. 当 $D_{01} < D_{11}$ 时的时序图



6.6.3 外部触发脉冲输出模式 (TPmMD2 至 TPmMD0 位 = 010)

此模式仅在 TMP0, TMP2, 和 TMP3 中有效 (软件触发时仅限 TMP3)。

在外部触发脉冲输出模式下, 当 TPmCTL0.TPmCE 位被设置为 1 时, 16 位定时器/事件计数器 P 等待触发, 当检测到外部触发输入 (TIPk0) 的有效边沿后, 16 位定时器/事件计数器 P 开始计数, 同时从 TOPm1 引脚输出一个脉宽调制信号波形。

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。使用软件触发时, 带有 50% 占空比的 PWM 波形也会从 TOP00 引脚输出, 其中 PWM 波形的周期的一半等于 TPmCCR0 寄存器的值 + 1。

<R>

图 6-23. 外部触发脉冲输出模式的配置

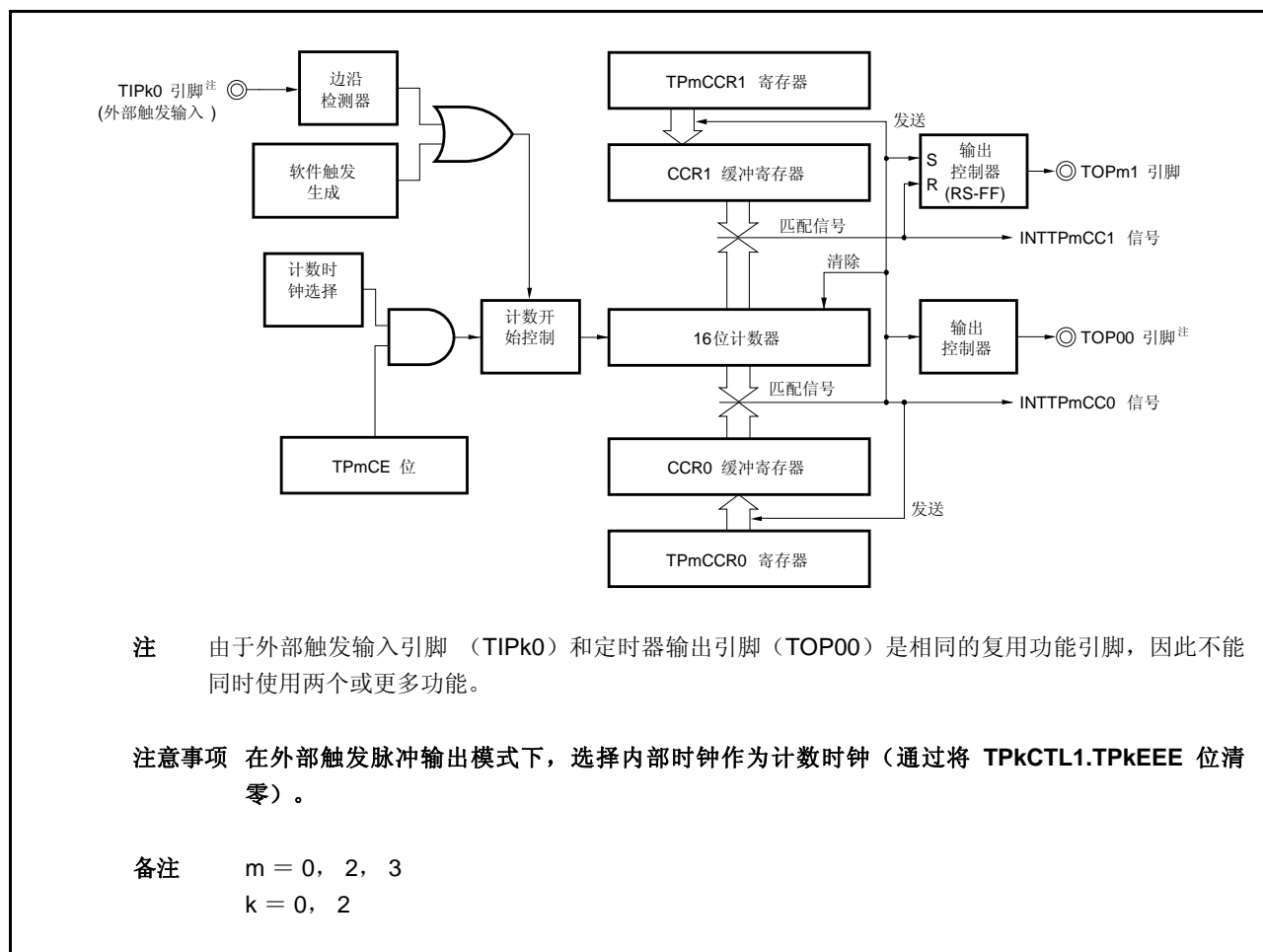
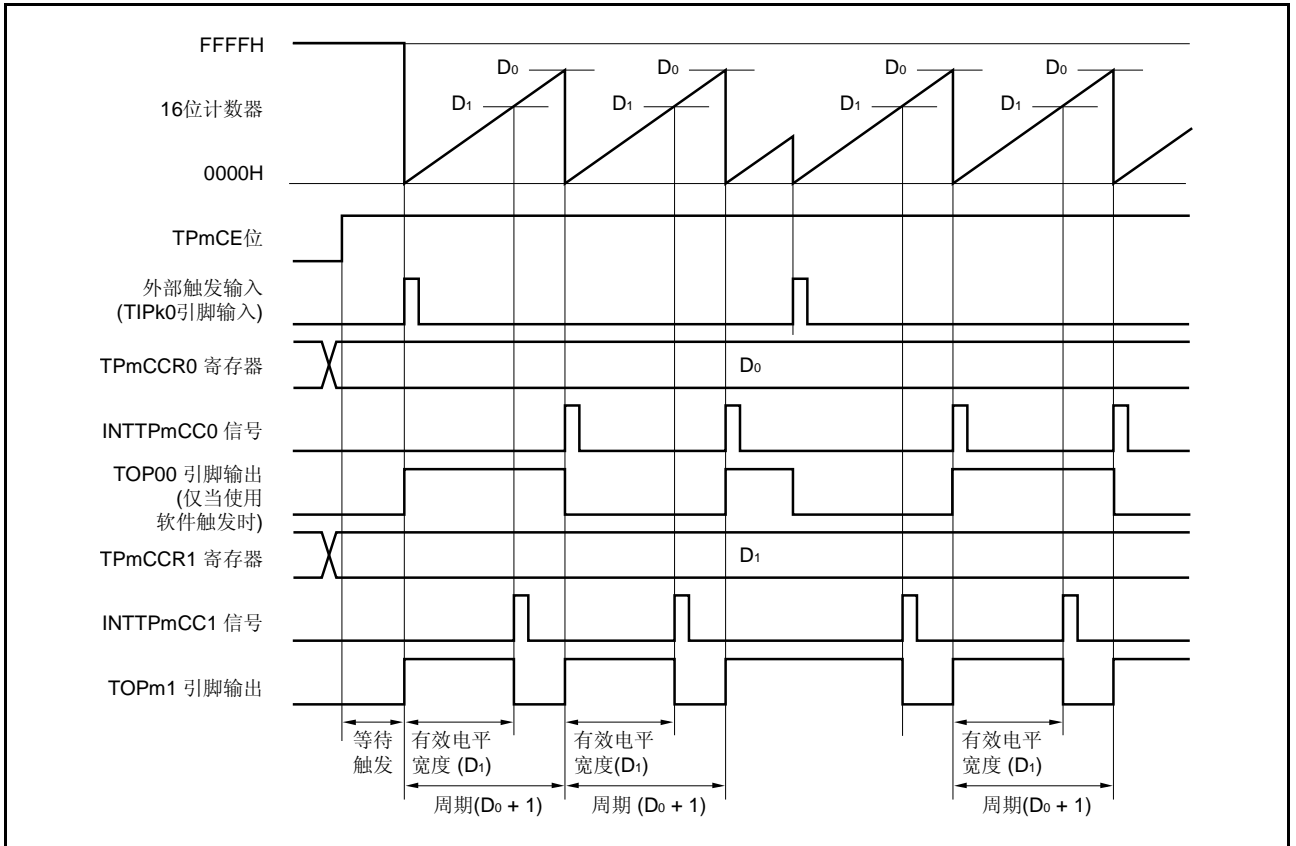


图 6-24. 外部触发脉冲输出模式的基本时序



当 TPmCE 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 TOPm1 引脚输出一个脉宽调制信号波形。若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。（TOP00 引脚输出反相。在出现触发时，不管当前状态如何（高/低），TOPm1 引脚都会输出高电平。

脉冲宽度调制信号波形的有效电平宽度、周期、占空比计算如下。

有效电平宽度 = (TPmCCR1 寄存器的设置值) × 计数时钟周期

周期 = (TPmCCR0 寄存器的设置值 + 1) × 计数时钟周期

占空比系数 = (TPmCCR1 寄存器的设置值) / (TPmCCR0 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向上计数时，会同步产生比较匹配中断请求信号 INTTPmCC0，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 INTTPmCC1。

当 16 位计数器的计数值与 CCRa 缓冲寄存器的值匹配，而且 16 位计数器清零为 0000H 时，TPmCCR a 寄存器设置的值就传送到 CCRa 缓冲寄存器。

外部触发输入 (TIPk0) 的有效边缘或将软件触发 (TPmCTL1.TPmEST 位) 设置为 1，都可用作触发信号。

备注 m = 0, 2, 3

k = 0, 2

a = 0, 1

图 6-25. 外部触发脉冲输出模式的寄存器设置 (1/2)

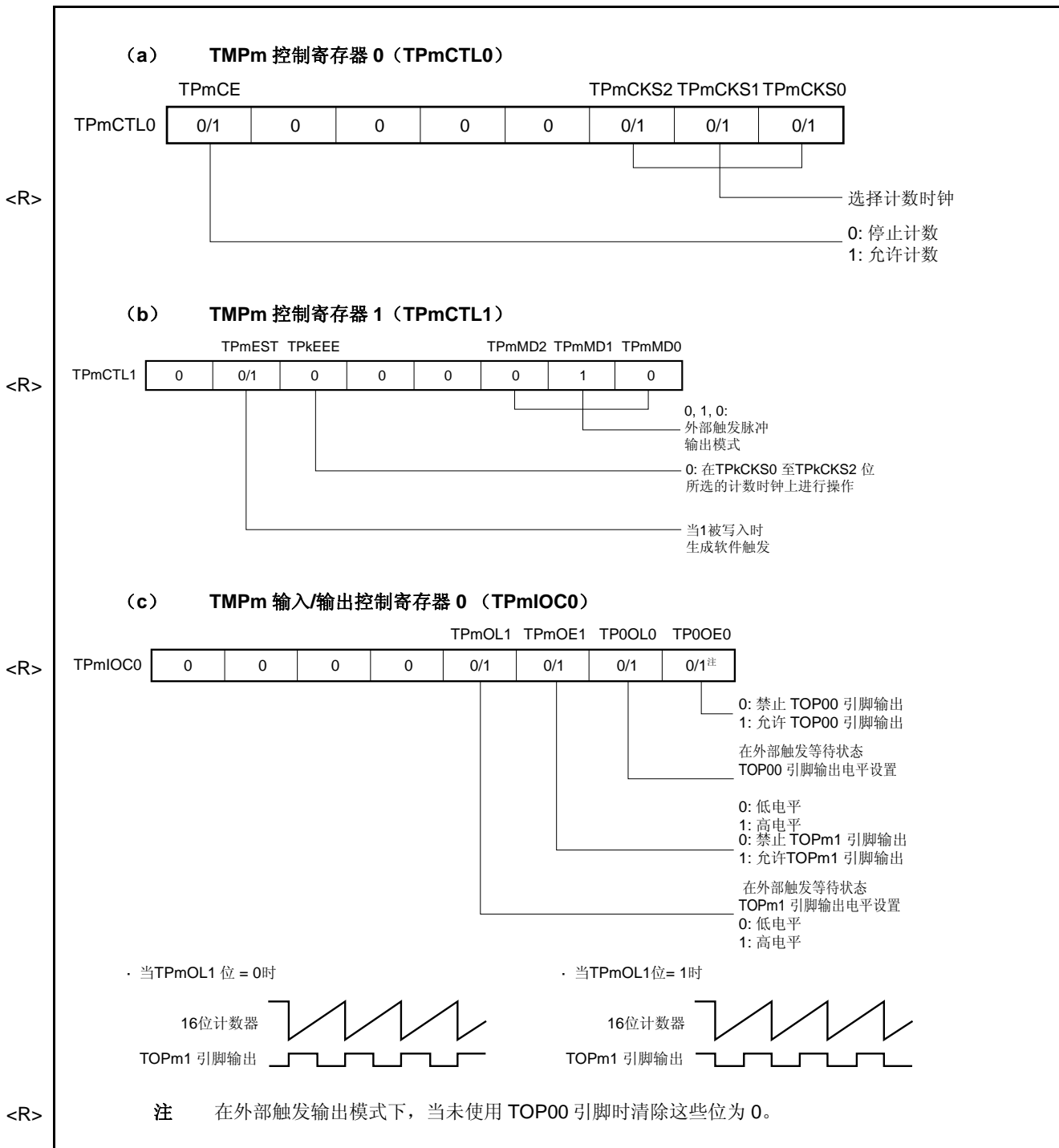
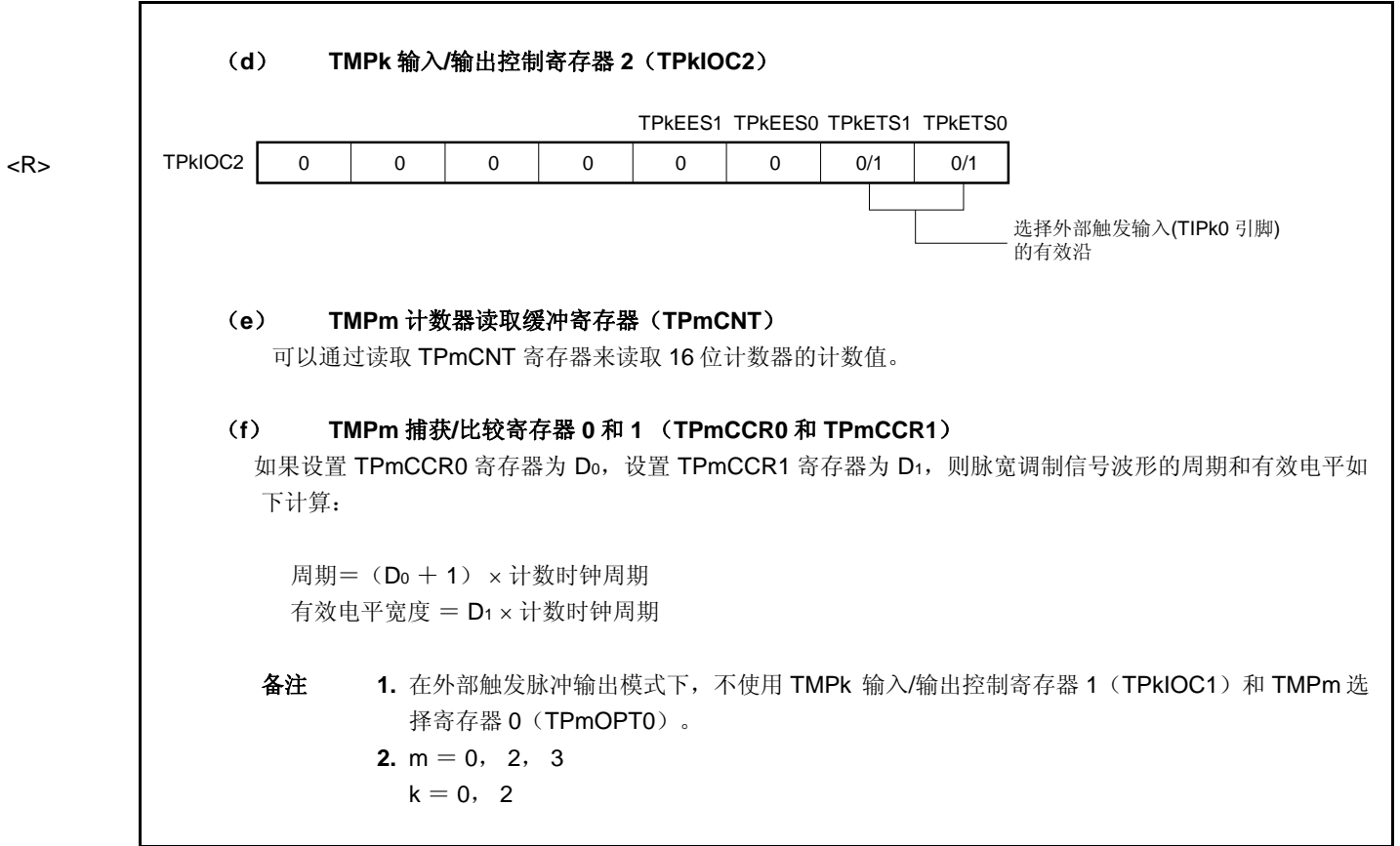


图 6-25. 外部触发脉冲输出模式的寄存器设置 (2/2)



(1) 外部触发脉冲输出模式的工作流程

图 6-26. 外部触发脉冲输出模式的软件处理流程 (1/2)

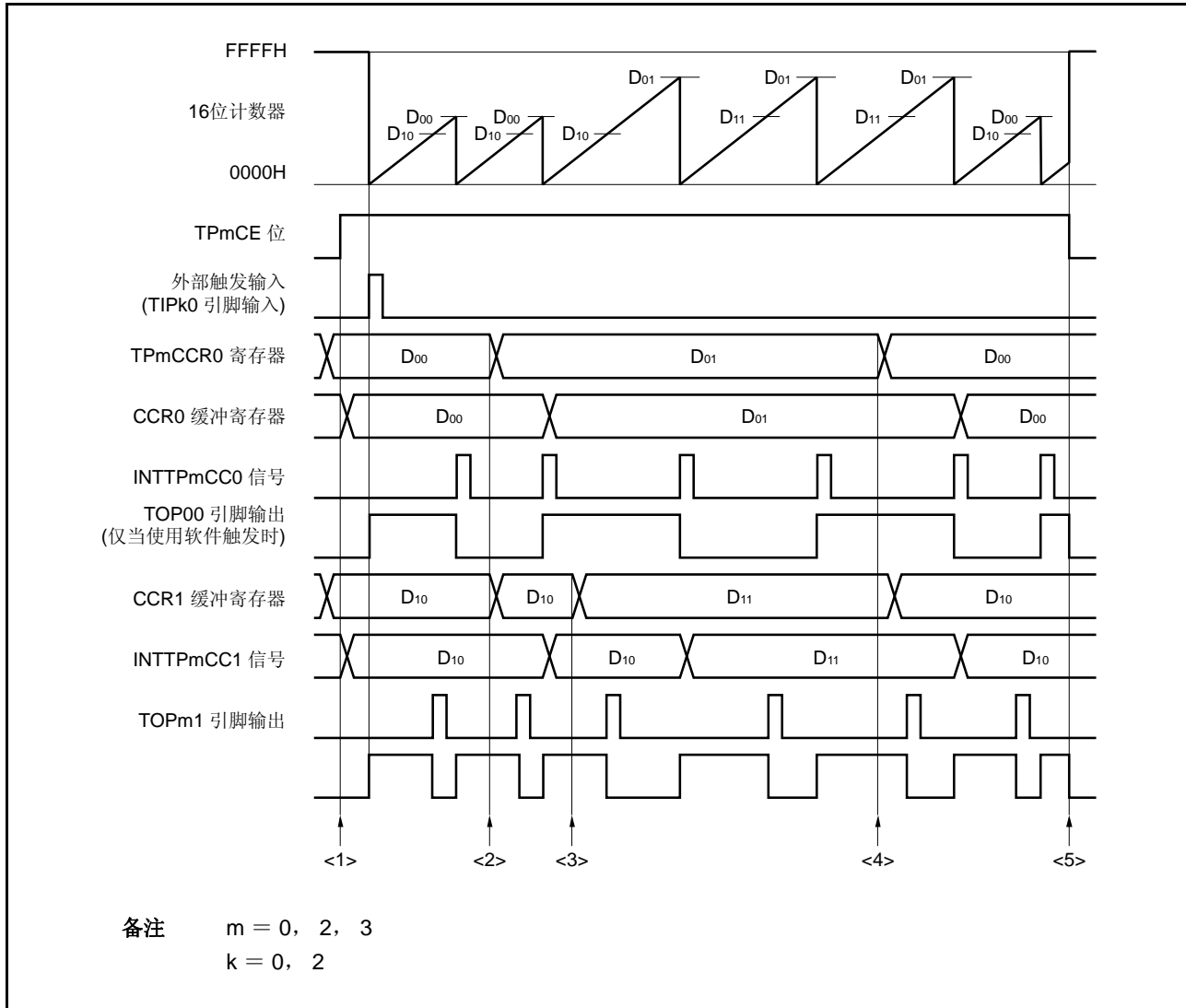
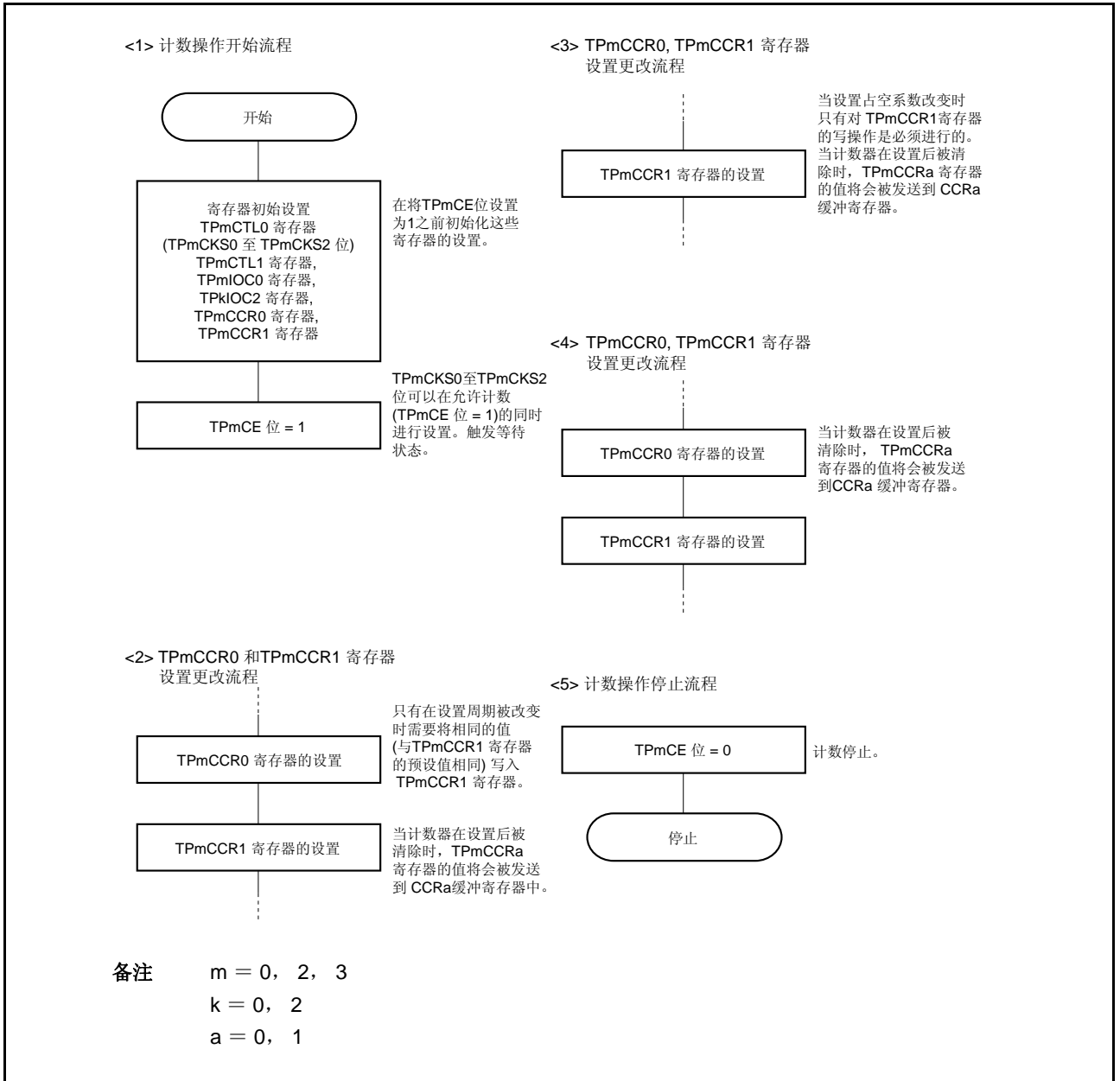


图 6-26. 外部触发脉冲输出模式的软件处理流程 (2/2)

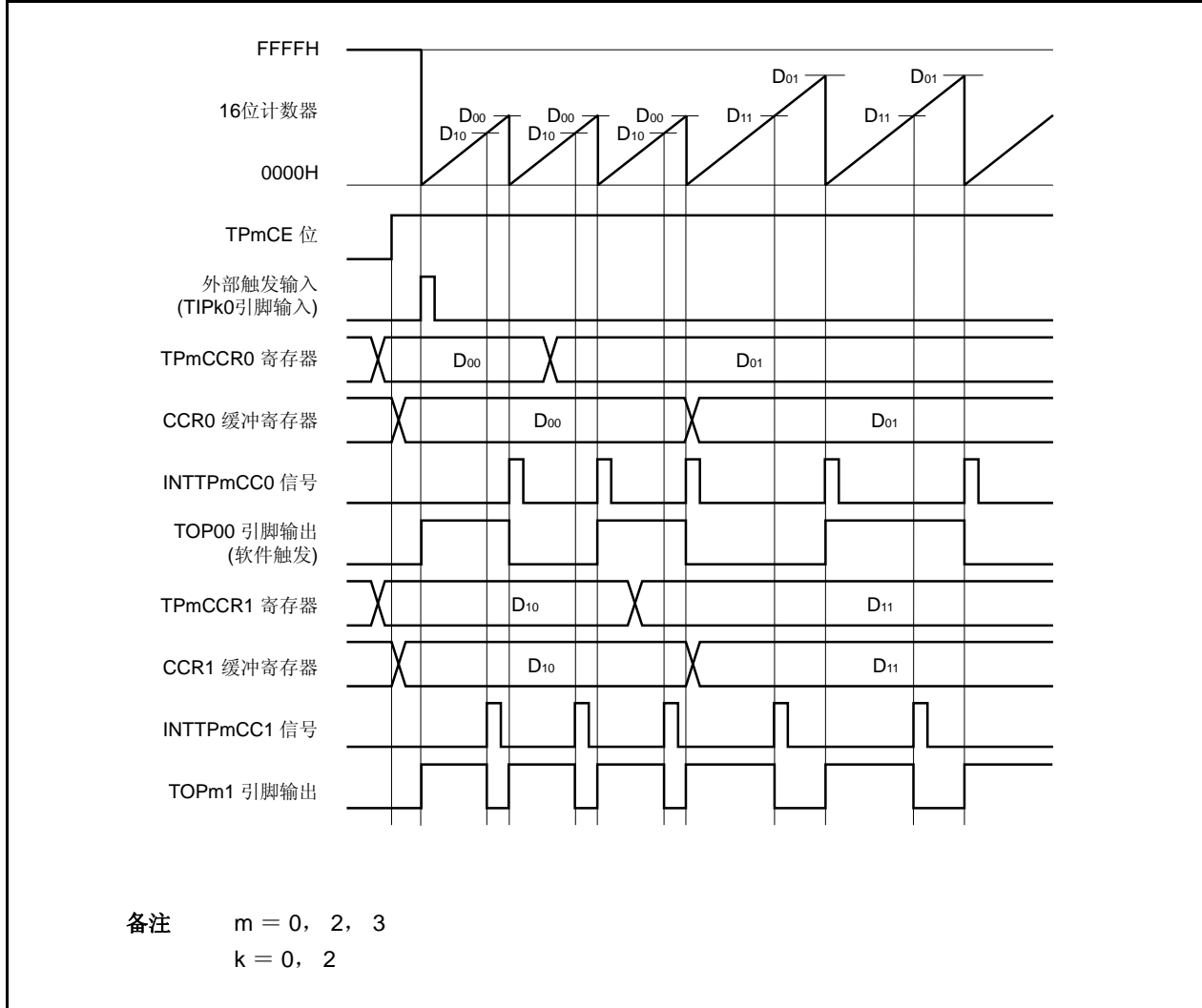


(2) 外部触发脉冲输出模式的操作时序

(a) 有关操作中改变脉冲宽度的注意事项

计数器正在工作时，改变脉宽调制信号波形应最后写入 TPmCCR1 寄存器。

在检测到 INTTPmCC1 信号后，将其写入 TPmCCR1 寄存器，然后再重写 TPmCCRa 寄存器。



为了将数据从 TPmCCRa 寄存器传送到 CCRa 缓冲寄存器，必须写入 TPmCCR1 寄存器。

要改变脉宽调制信号波形的周期和有效电平宽度，应首先将周期设定给 TPmCCR0 寄存器，然后将有效电平宽度设定给 TPmCCR1 寄存器。

只改变 PWM 波形的周期时，先将周期值写入 TPmCCR0 寄存器，再向 TPmCCR1 寄存器写入相同的值（与 TPmCCR1 寄存器的预置值相同）。

如果只改变脉宽调制信号波形的有效电平宽度（占空比），那就只需要设置 TPmCCR1 寄存器。

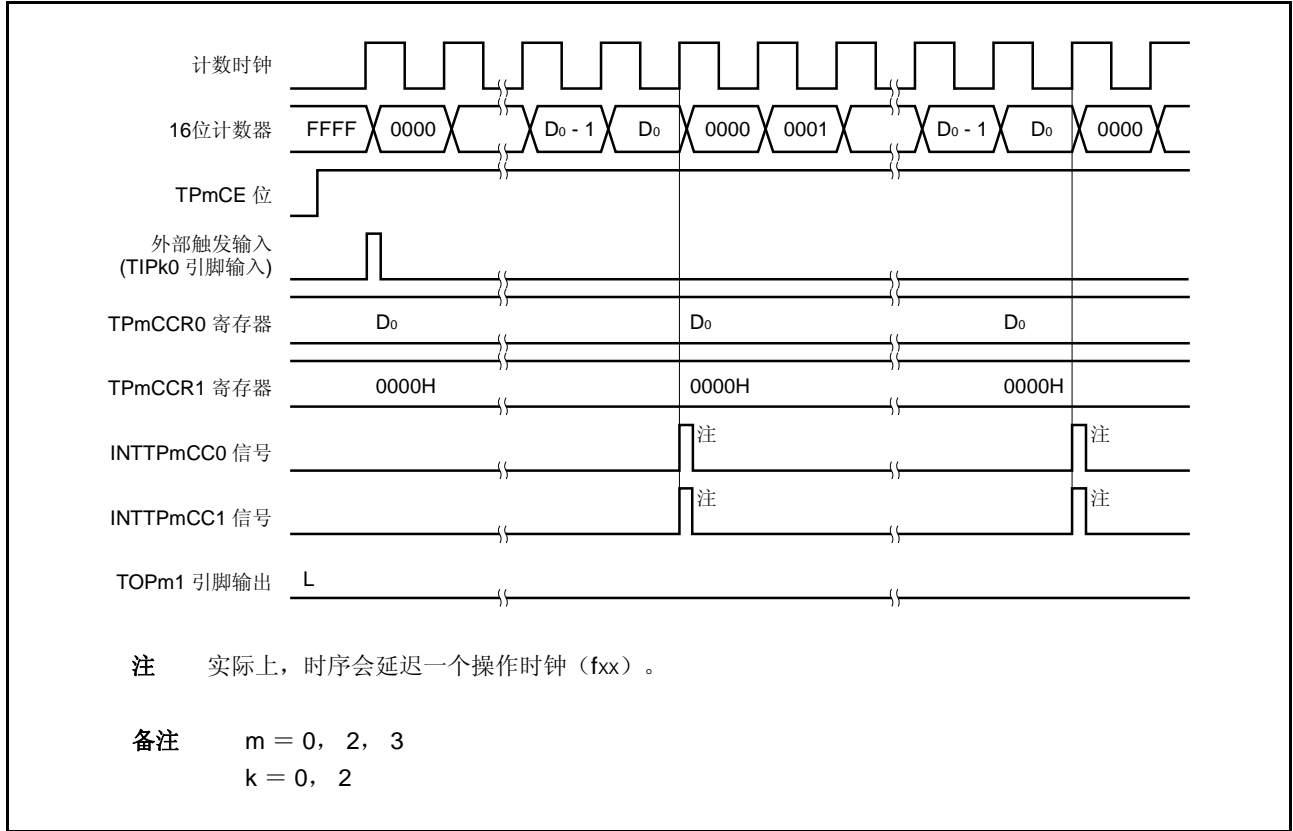
在数据写入 TPmCCR1 寄存器后，写入 TPmCCRa 寄存器的值会传送到 CCRa 缓冲寄存器，与 16 位计数器清零同步，并作为与 16 位计数器进行比较的值。

要在写入 TPmCCR1 寄存器一次之后，再次写 TPmCCR0 或 TPmCCR1 寄存器，应在 INTTPmCC0 信号产生之后写入，否则，CCRa 缓冲寄存器的值可能不确定，因为把数据从 TPmCCRa 寄存器传送到 CCRa 缓冲寄存器的时间与写入 TPmCCRa 寄存器相冲突。

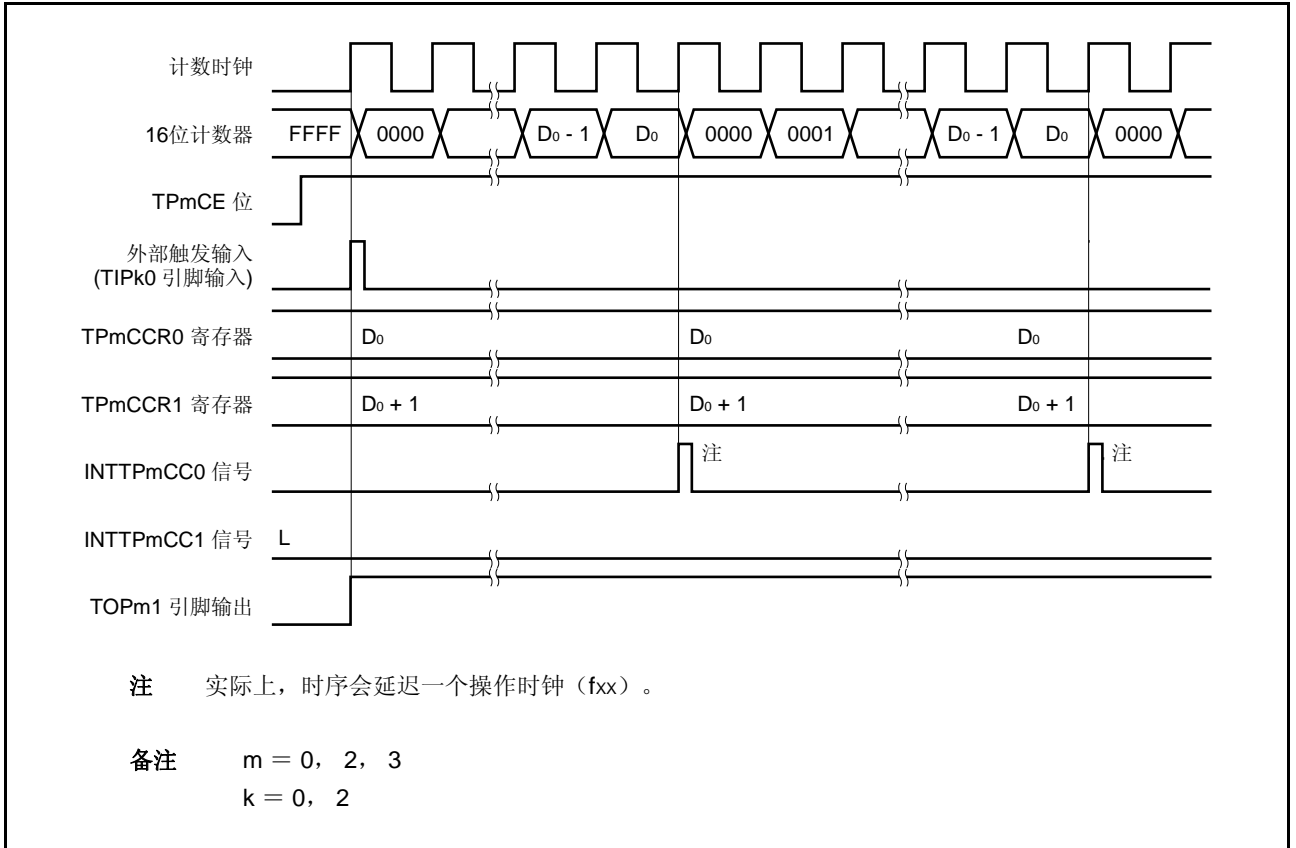
备注 m = 0, 2, 3
 a = 0, 1

(b) 脉宽调制信号波形的 0% 和 100% 输出

为了输出 0% 的波形，将 TPmCCR1 寄存器设置为 0000H。在下次 16 位计数器的值与 CCR0 缓冲寄存器的值匹配后，16 位计数器将会被清除为 0000H 且会生成 INTTPmCC0 和 INTTPmCC1 信号。



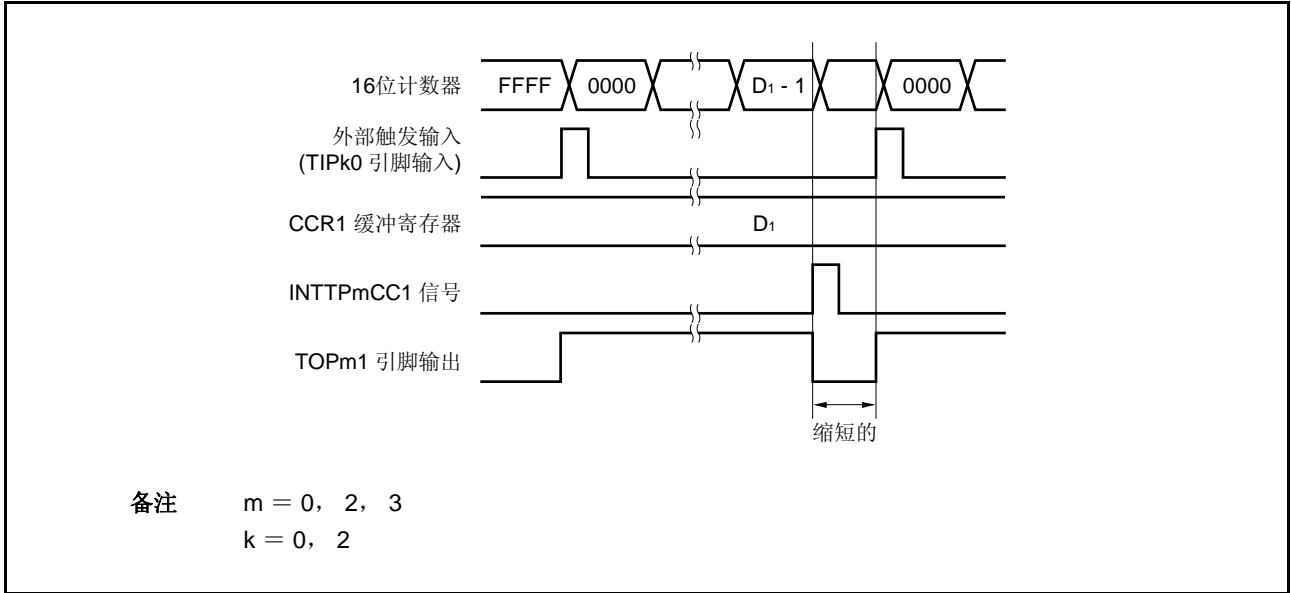
为了输出 100% 的波形，需要将 TPmCCR1 寄存器的值设为：(TPmCCR0 寄存器的设定值 + 1)。如果 TPmCCR0 寄存器的设定值为 FFFFH，则不会产生 100% 波形输出。



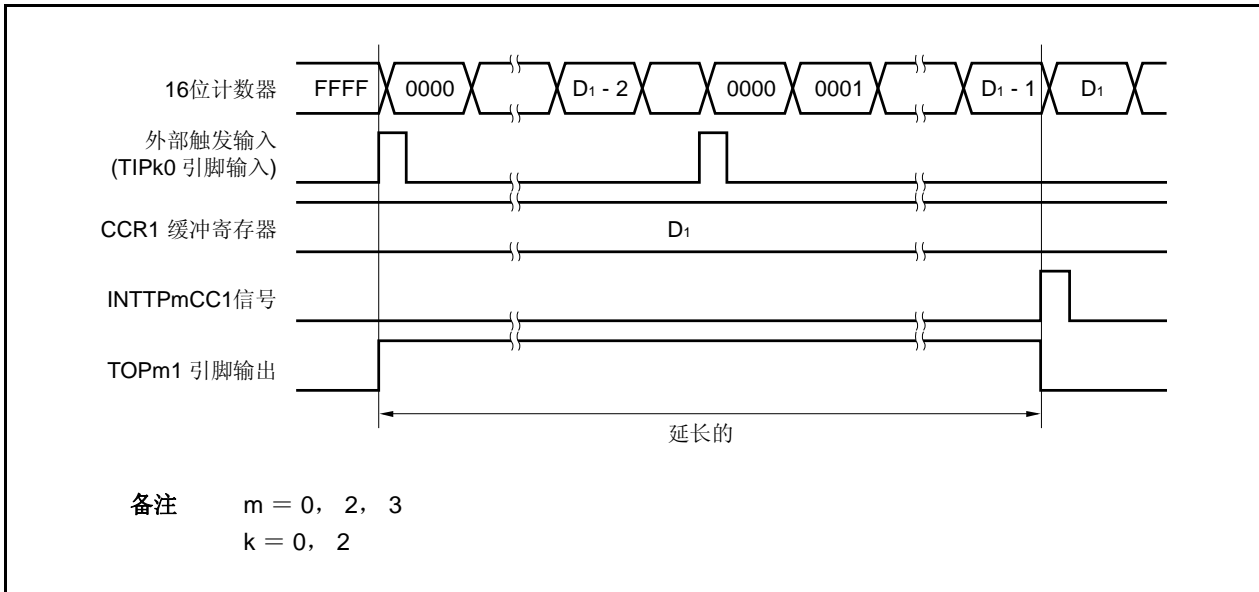
<R>

(c) 触发检测和 CCR1 缓冲寄存器匹配之间的冲突

如果在 INTTPmCC1 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，TOPm1 引脚的输出信号被置位生效，计数器继续计数，结果，缩短了脉宽调制信号波形的无效时域。

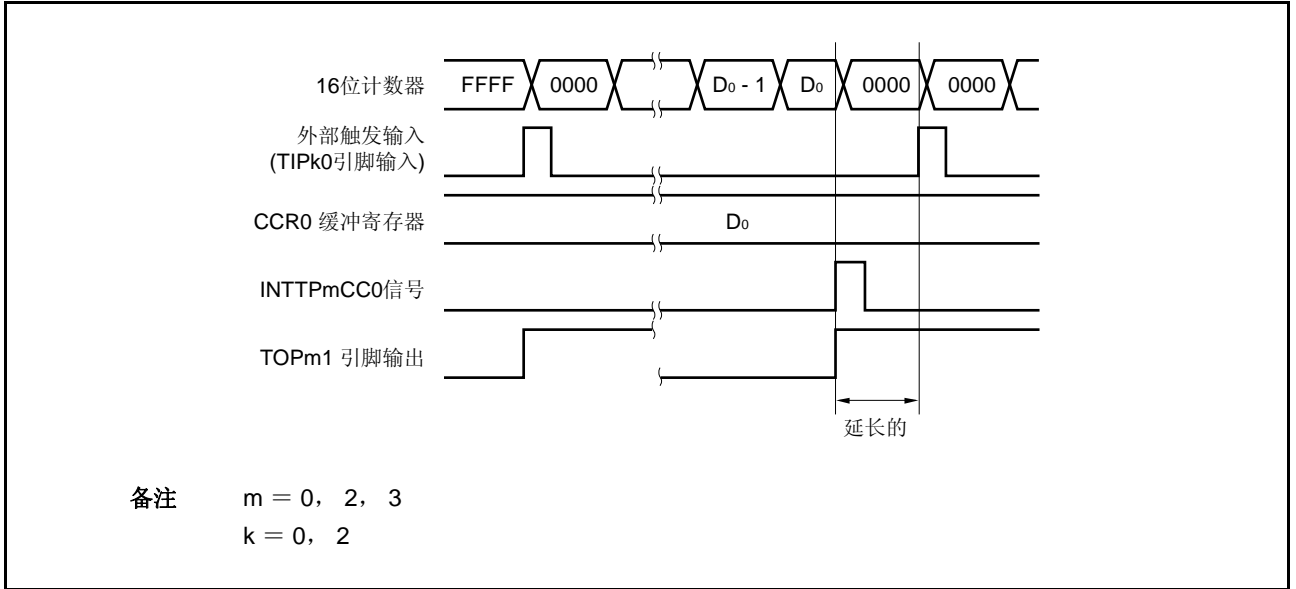


如果刚好在 INTTPmCC1 信号产生前检测到触发信号，则不会产生 INTTPmCC1 信号，而 16 位计数器清零为 0000H，计数继续进行。TOPm1 引脚的输出信号也会继续保持为有效电平。这种情况下，PWM 波形的有效电平时间就会被延长。

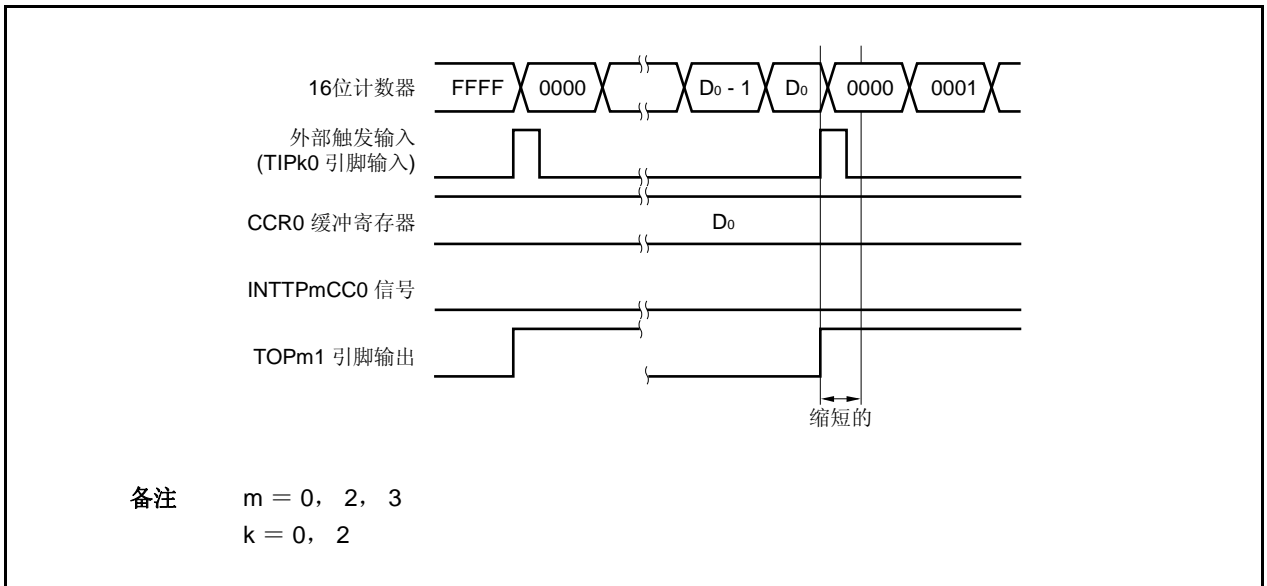


(d) 触发检测和 CCR0 缓冲寄存器匹配之间的冲突

如果刚好在 INTTPmCC0 信号产生后检测到触发，则 16 位计数器清零为 0000H，并继续计数。因此，TOPm1 引脚的有效时段被延长，延长时间为自 INTTPmCC0 信号产生时刻到触发被检测到时刻。

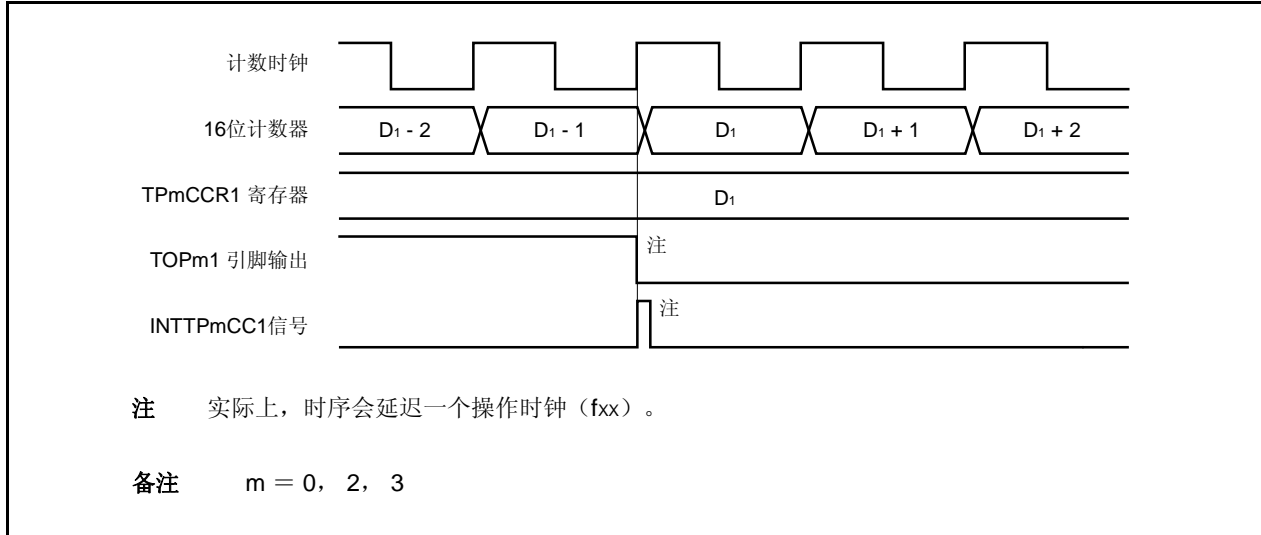


如果刚好在产生 INTTPmCC0 信号之前检测到触发信号，则不会产生 INTTPmCC0 信号。16 位计数器清零为 0000H，TOPm1 引脚输出被置位，计数器继续计数。结果，缩短了脉宽调制信号波形的无效时段。



(e) 比较匹配中断请求信号 (INTTPmCC1) 的发生时序

在外部触发脉冲输出模式下，INTTPmCC1 信号的发生时序与其它情况下 INTTPmCC1 信号的时序有所不同；当 16 位计数器的计数值和 TPmCCR1 寄存器的值匹配时，产生 INTTPmCC1 信号。



通常情况下，在 16 位计数器的当前计数值与 TPmCCR1 寄存器的值匹配之后，INTTPmCC1 信号与下一次计数同步发生。

但是，在外部触发脉冲输出模式下，其产生会提早一个计数时钟周期。这是由于为了匹配 TOPm1 引脚输出信号的变化，时序发生了变化。

6.6.4 单次脉冲输出模式(TPmMD2 至TPmMD0 位 = 011)

此模式仅在 TMP0, TMP2, 和 TMP3 中有效 (软件触发时仅限 TMP3)。

在单次脉冲输出模式下, 当 TPmCTL0.TPmCE 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(TIPk0)的有效边沿时, 16 位定时器/事件计数器 P 开始计数, 同时从 TOPm1 引脚输出一个单次脉冲。

除外部触发外, 也可以生成软件触发来输出该脉冲。在使用软件触发时, TOP00 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。

<R>

图 6-27. 单次脉冲输出模式的配置

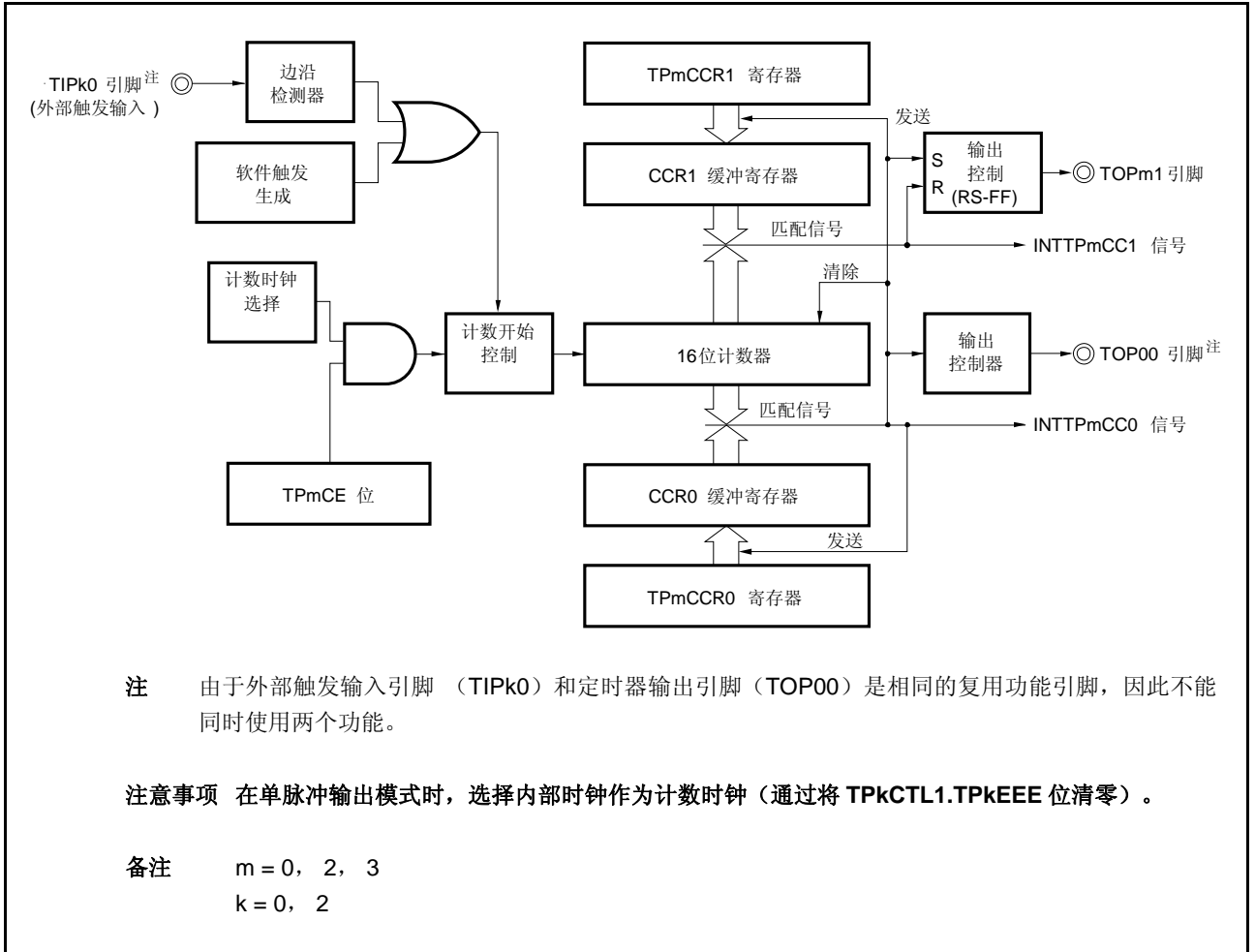
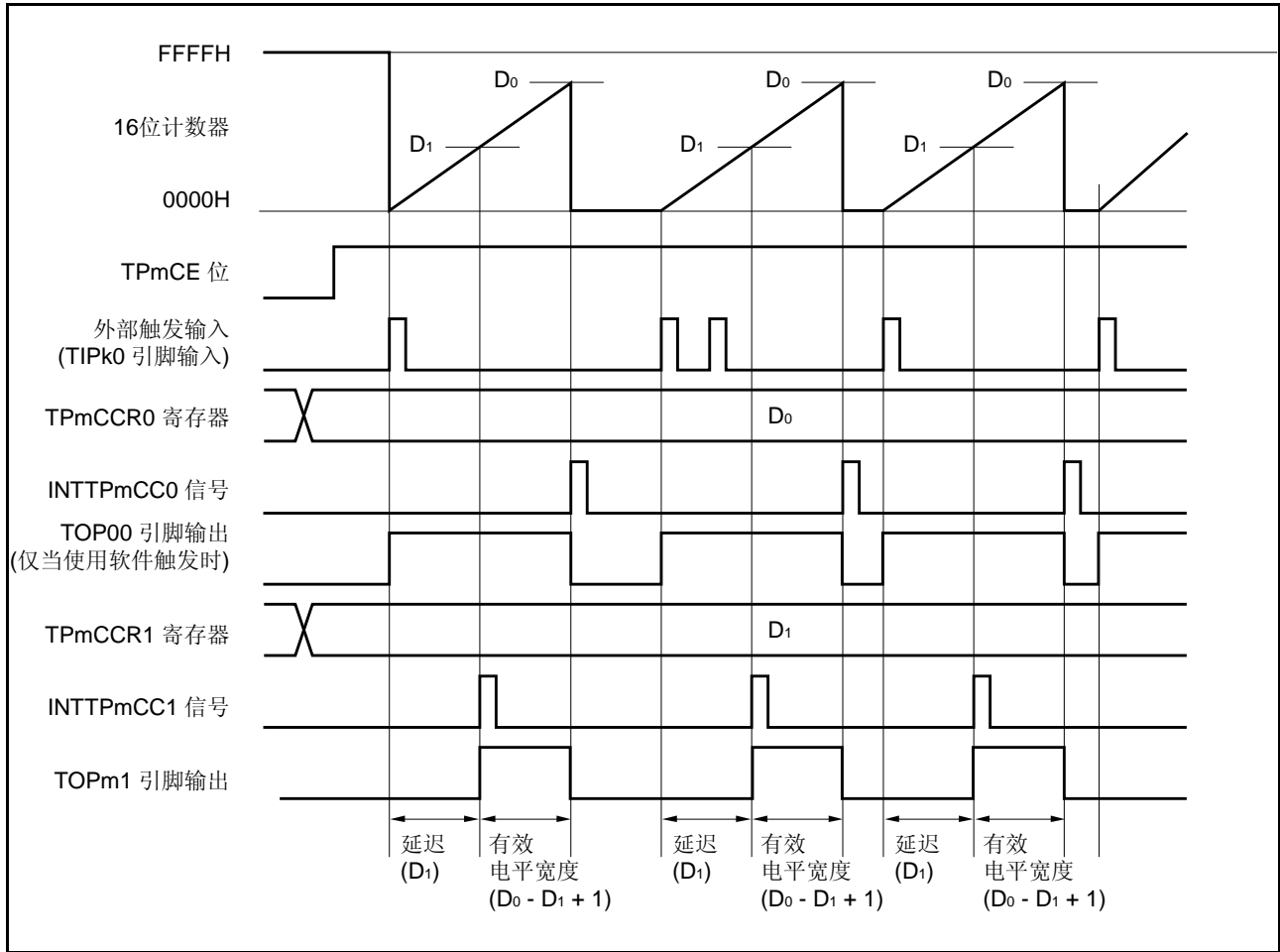


图 6-28. 单次脉冲输出模式的基本时序



当 TPmCE 位设置为 1 时，16 位定时器/事件计数器 P 等待触发。在触发发生时，16 位计数器从 FFFFH 清零为 0000H，开始进行计数，并从 TOPm1 引脚输出一个单次脉冲。在输出单次脉冲后，16 位计数器被清除为 0000H 并停止计数，进入等待触发状态。当再次产生触发时，16 位计数器从 0000H 开始计数。当 TOP01 引脚正在输出单次脉冲期间再次发生触发，则该触发被忽略。

单次脉冲的输出延迟期和有效电平宽度可计算如下：

输出延迟时间 = (TPmCCR1 寄存器的设置值) × 计数时钟周期

有效电平宽度 = (TPmCCR0 寄存器的设置值 - TPmCCR1 寄存器的设置值 + 1) × 计数时钟周期

当 16 位计数器在其计数值与 CCR0 缓冲寄存器的值匹配之后进行下一个计数时，产生 INTTPmCC0 比较匹配中断请求信号。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 INTTPmCC1。

外部触发输入(TIPk0 引脚)的有效边缘或将软件触发 (TPmCTL1.TPmEST 位) 设置为 1，都可用作触发信号。

备注 $m = 0, 2, 3$
 $k = 0, 2$

图 6-29. 单次脉冲输出模式时的寄存器设置 (1/2)

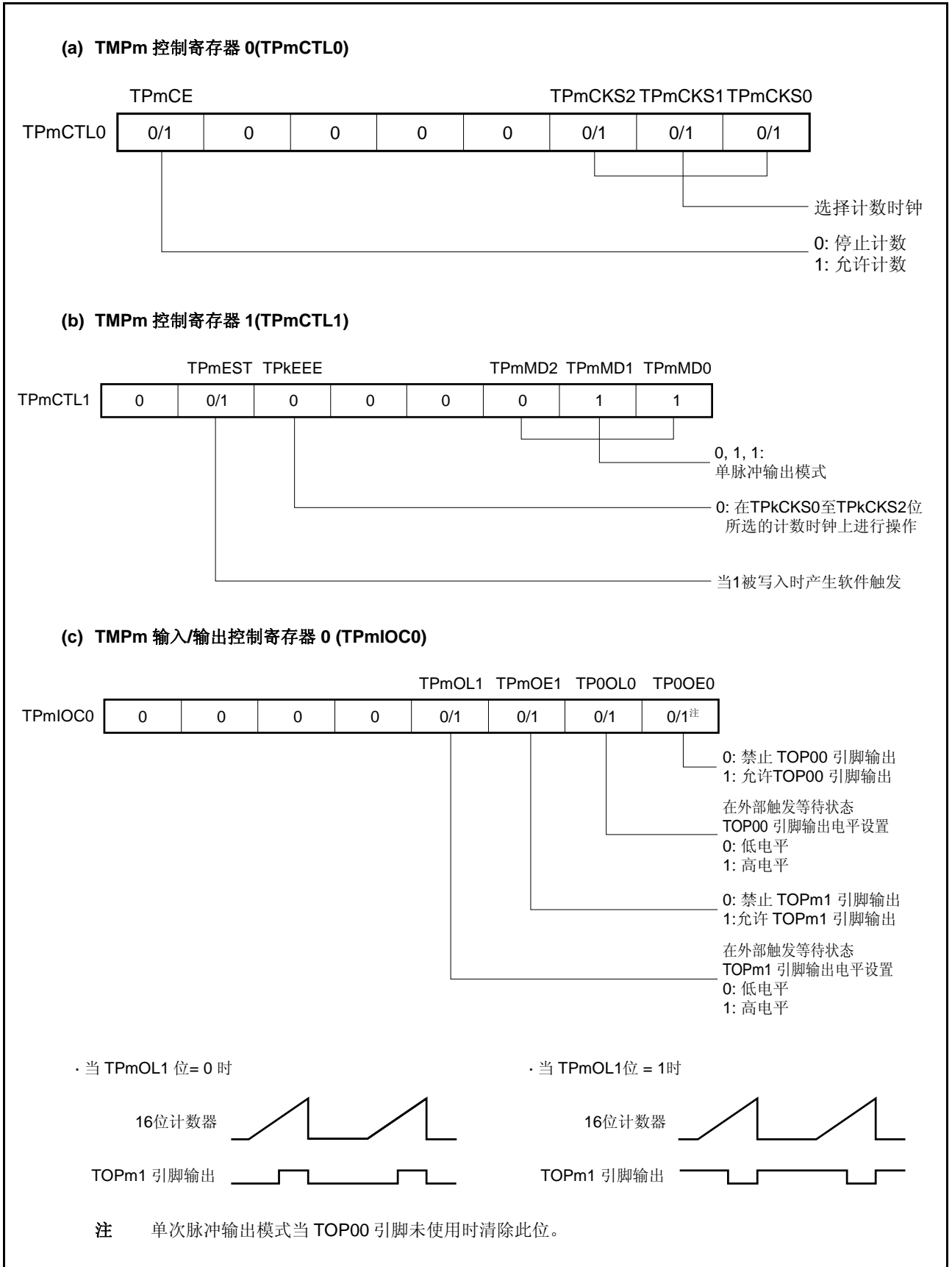
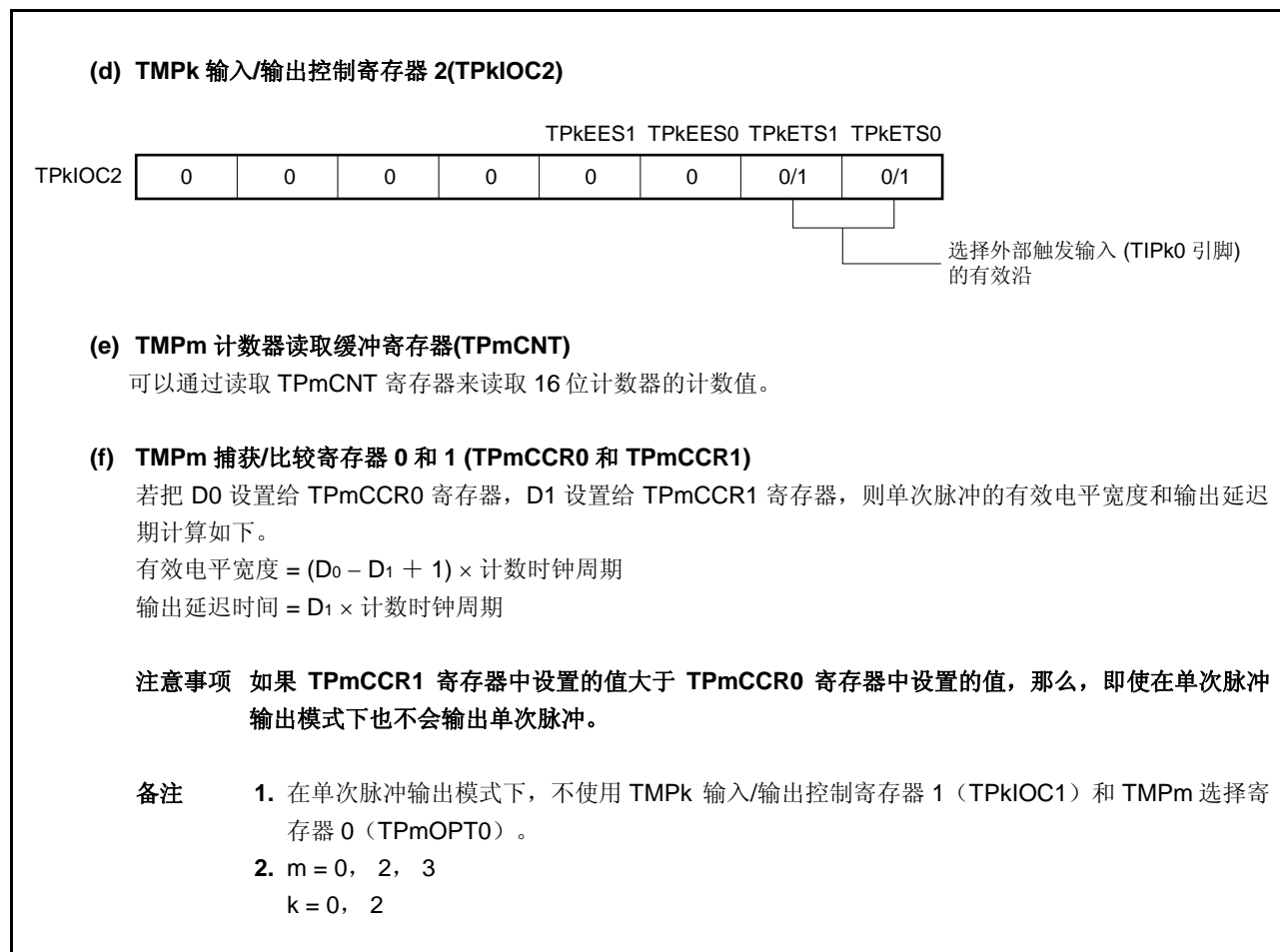


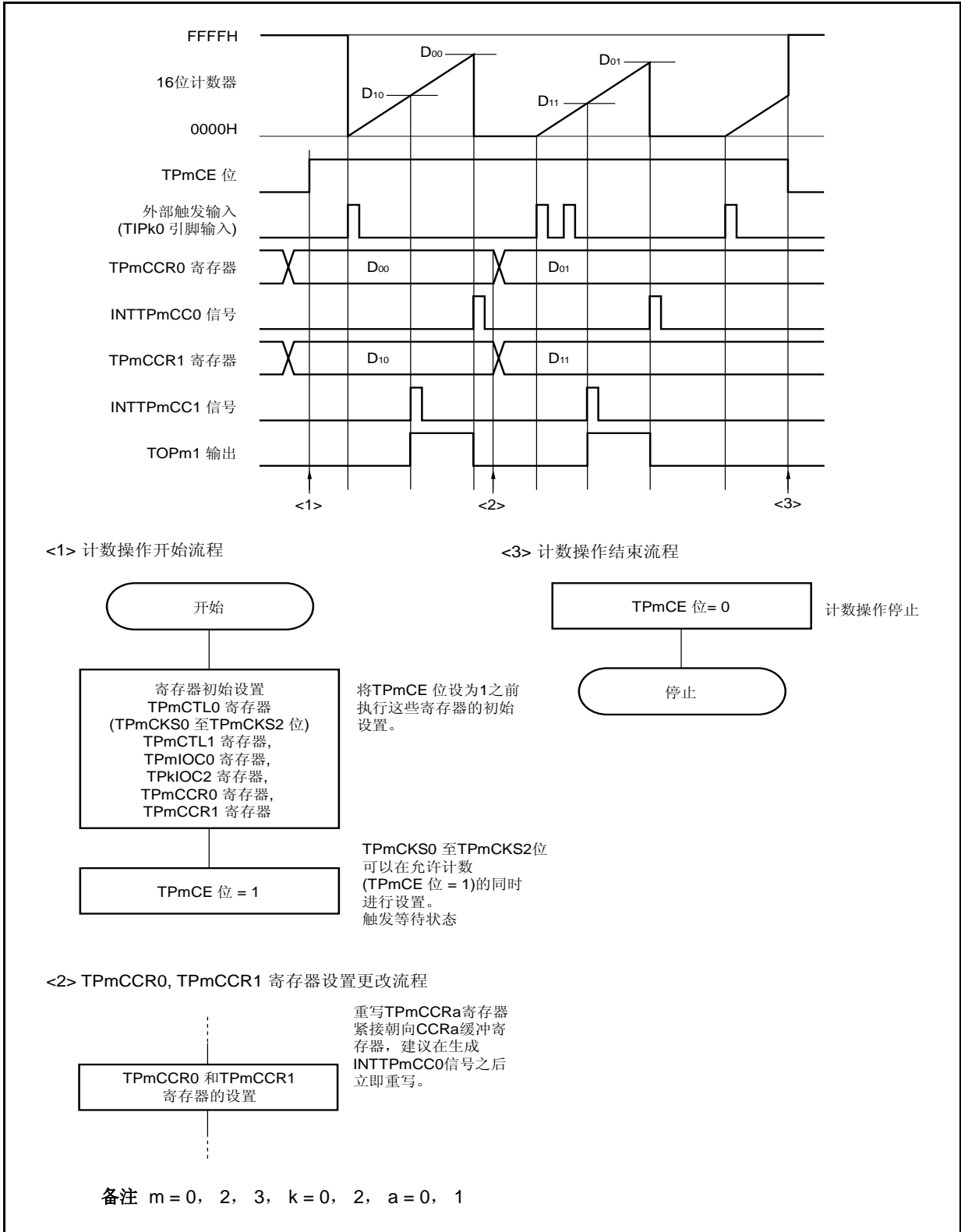
图 6-29. 单次脉冲输出模式时的寄存器设置 (2/2)



(1) 单次脉冲输出模式下的操作流程

<R>

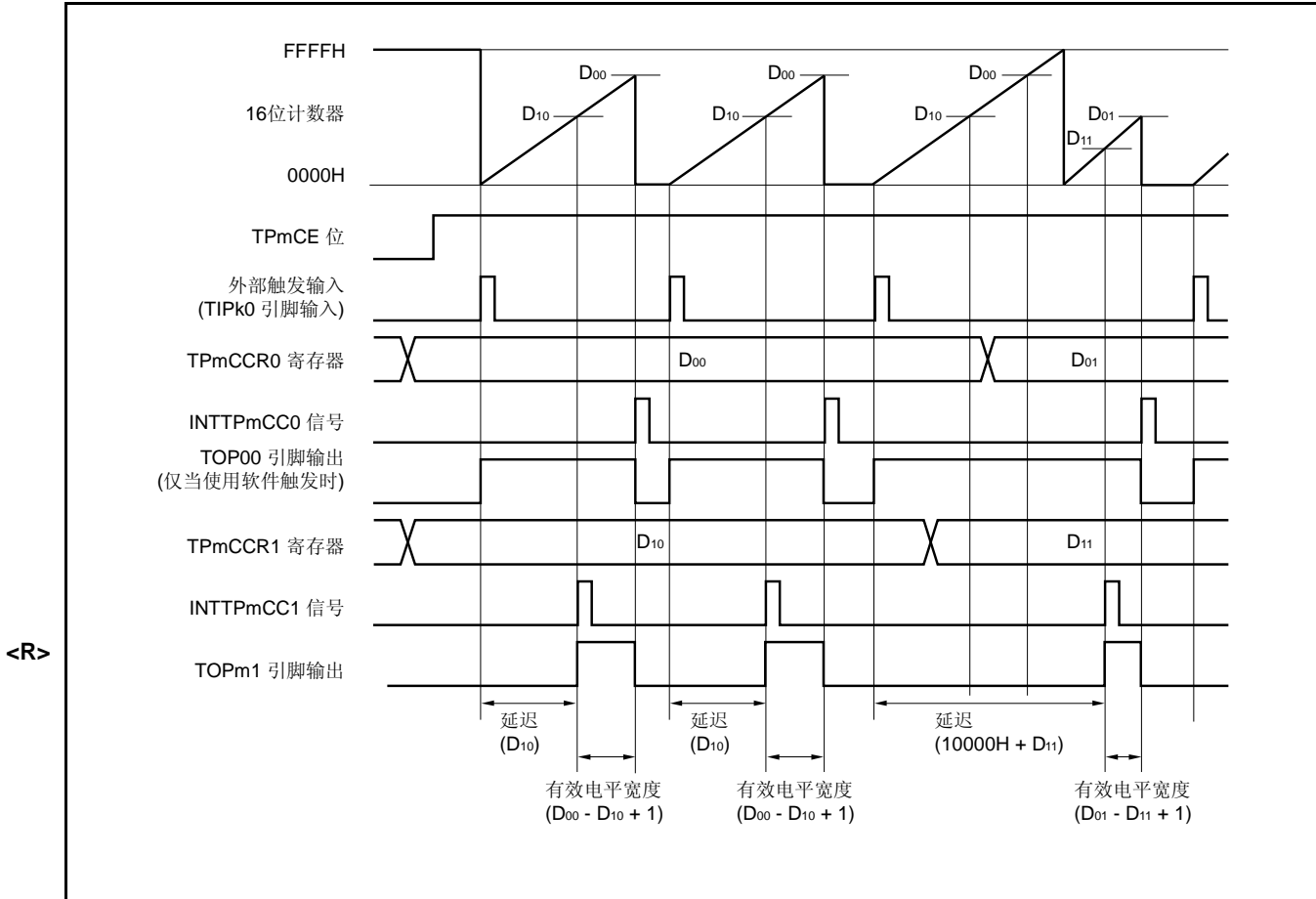
图 6-30. 单次脉冲输出模式下的软件处理流程



(2) 单次脉冲输出模式时的工作时序

(a) 改写 TPmCCRa 寄存器时需要注意的事项

如果在计数器工作期间，将 TPmCCRa 寄存器的值重写为比当前值较小的值，16 位计数器会发生溢出。如可能发生溢出，停止计数再更改设定值。



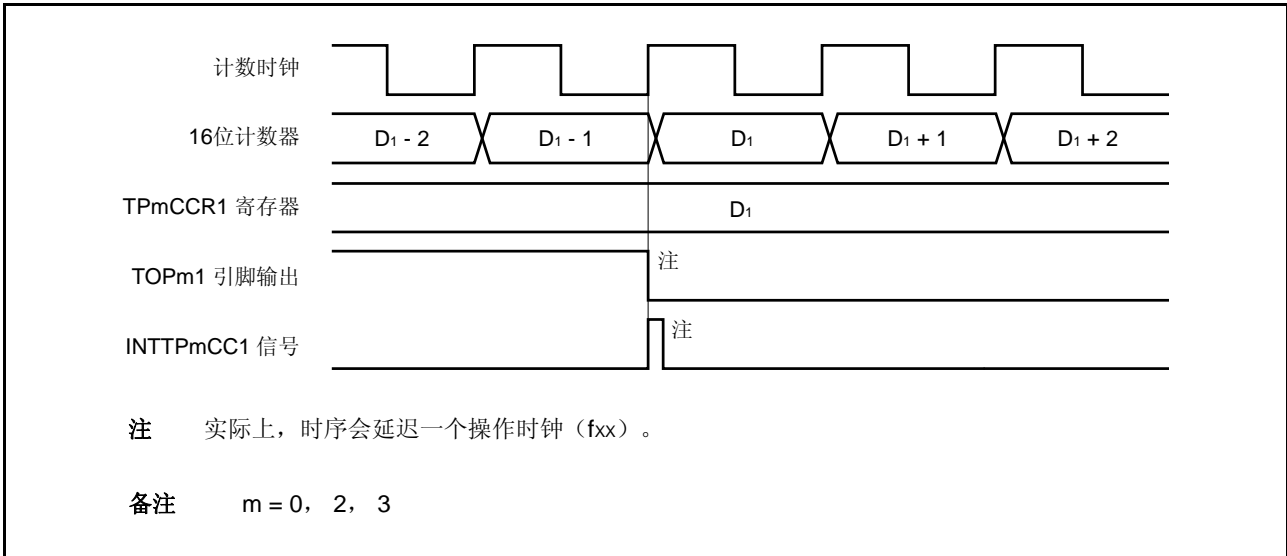
如果 $D_{00} > D_{01}$ 且 $D_{10} > D_{11}$ ，当 TPmCCR0 寄存器从 D_{00} 重写为 D_{01} ，TPmCCR1 寄存器从 D_{10} 重写为 D_{11} 时，若 TPmCCR1 寄存器在 16 位计数器的计数值大于 D_{11} 且小于 D_{10} 时重写，以及 TPmCCR0 寄存器在计数值大于 D_{01} 且小于 D_{00} 时重写，则一旦重写寄存器，设定值都将在改写后立刻生效并立即与计数值进行比较。计数器向上计数到 FFFFH 后，从 0000H 开始重新向上计数。当计数值与 D_{11} 匹配时，计数器会产生 INTTPmCC1 信号并使 TOPm1 引脚置位有效。当计数值与 D_{01} 匹配时，计数器产生 INTTPmCC0 信号，使 TOPm1 引脚清除置位，并停止计数。

因此，计数器输出一个带延迟的或有效高电平脉冲期脉冲，这可能会与原先预期的单次脉冲有所不同。

备注 $m = 0, 2, 3, k = 0, 2, a = 0, 1$

(b) 比较匹配中断请求信号(INTTPmCC1)的发生时序

单脉冲输出模式下 INTTPmCC1 信号的产生时序与其它模式下 INTTPmCC1 信号的产生时序有所不同；INTTPmCC1 信号在 16 位计数器的计数值与 TPmCCR1 寄存器的值相匹配时产生。



通常情况下, 在 16 位计数器的当前计数值与 TPmCCR1 寄存器的值匹配之后, INTTPmCC1 信号与下一次计数同步发生。

但是, 在单脉冲输出模式下, INTTP0CC1 信号会提早一个计数时钟周期产生。这是由于为了匹配 TOP01 引脚输出信号的变化, 时序发生了变化。

6.6.5 PWM 输出模式(TPmMD2 至 TPmMD0 位 = 100)

该模式仅在 TMP0, TMP2 和 TMP3 中有效。

在脉宽调制信号输出模式下, 当 TPmCTL0.TPmCE 位被设定为 1 时, 脉宽调制信号波形从 TOPm1 引脚输出。

此外, 从 TOP00 引脚中会输出 50%占空比的 PWM 波形, 其中 PWM 波形半个周期值为 TPmCCR0 寄存器的设定值 + 1。

图 6-31. 脉宽调制信号输出模式的配置

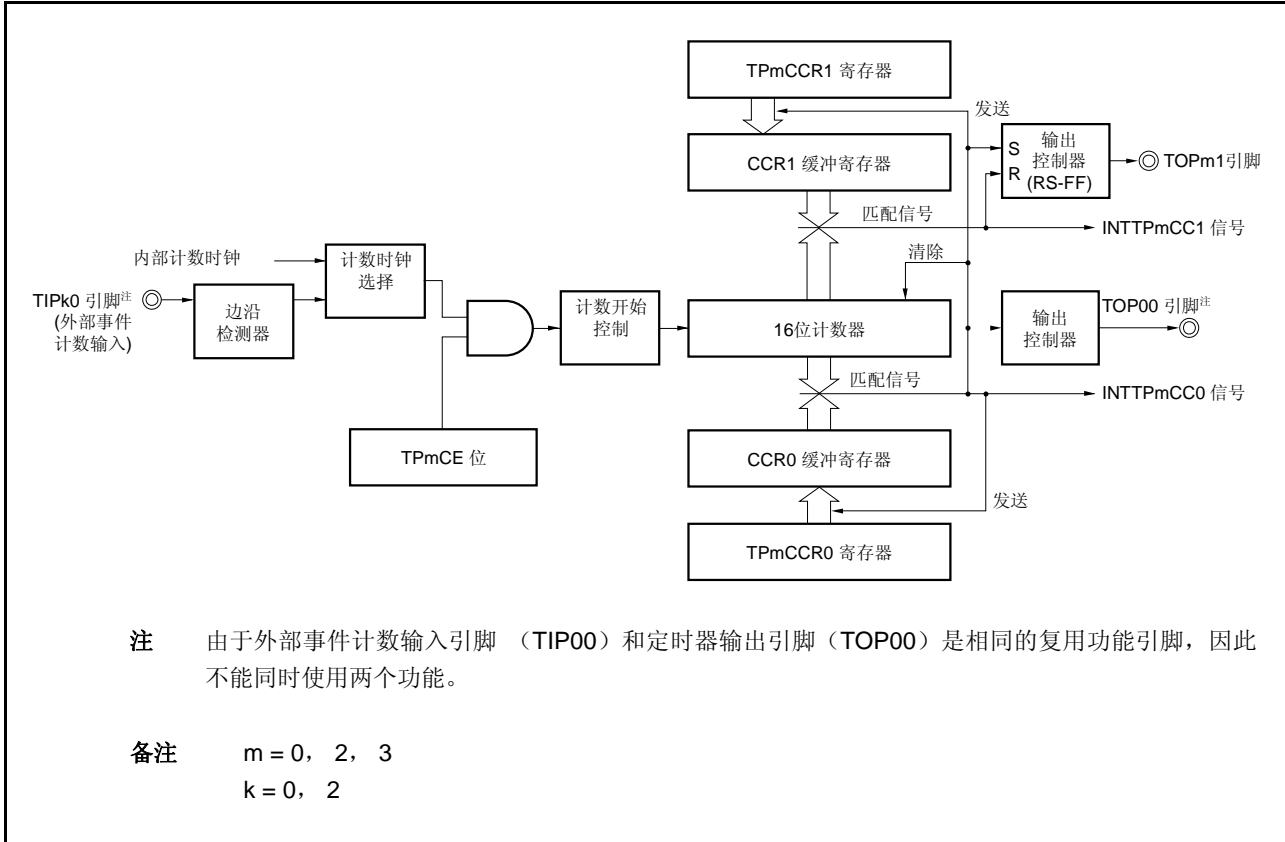
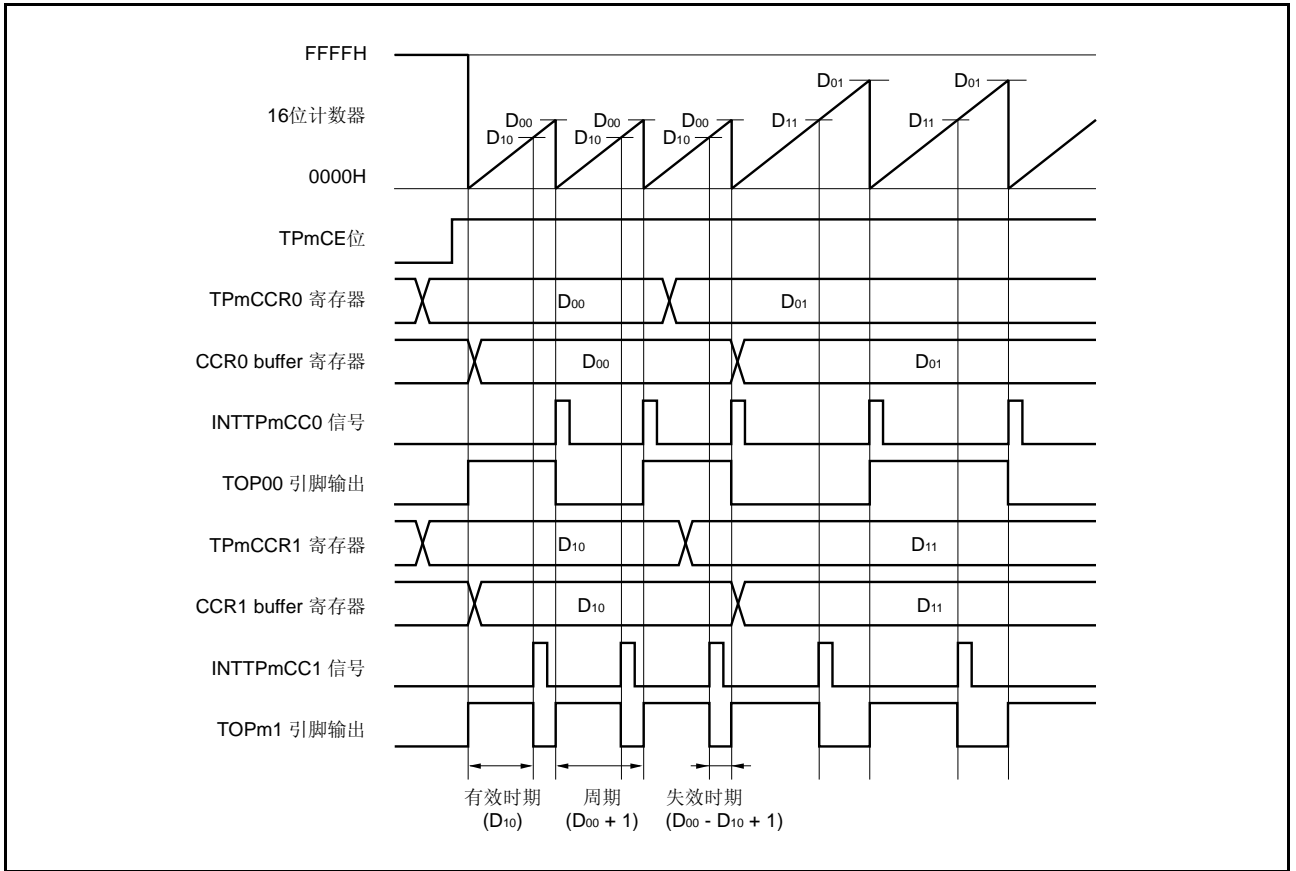


图 6-32. 脉宽调制信号输出模式的基本时序



当 TPmCE 位设置为 1 时，16 位计数器从 FFFFH 清零为 0000H，开始进行计数，同时从 TOPm1 引脚输出脉宽调制信号波形。

脉宽调制信号波形的有效电平宽度、周期、占空比计算如下。

有效电平宽度 = (TPmCCR1 寄存器的设置值) × 计数时钟周期

周期 = (TPmCCR0 寄存器的设置值 + 1) × 计数时钟周期

占空比 = (TPmCCR1 寄存器的设置值) / (TPmCCR0 寄存器的设置值 + 1)

在进行计数时，可通过重写 TPmCCR_a 寄存器来改变脉宽调制信号波形。改写后的数值在当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配后生效，16 位计数器被清零为 0000H。

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向上计数时，会同步产生比较匹配中断请求信号 INTTPmCC0，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 INTTPmCC1。

当 16 位计数器的计数值与 CCR_a 缓冲寄存器的值匹配，而且 16 位计数器清零为 0000H 时，TPmCCR_a 寄存器设置的值就传送到 CCR_a 缓冲寄存器。

备注 m = 0, 2, 3, a = 0, 1

图 6-33. 脉宽调制信号输出模式时的寄存器设置 (1/2)

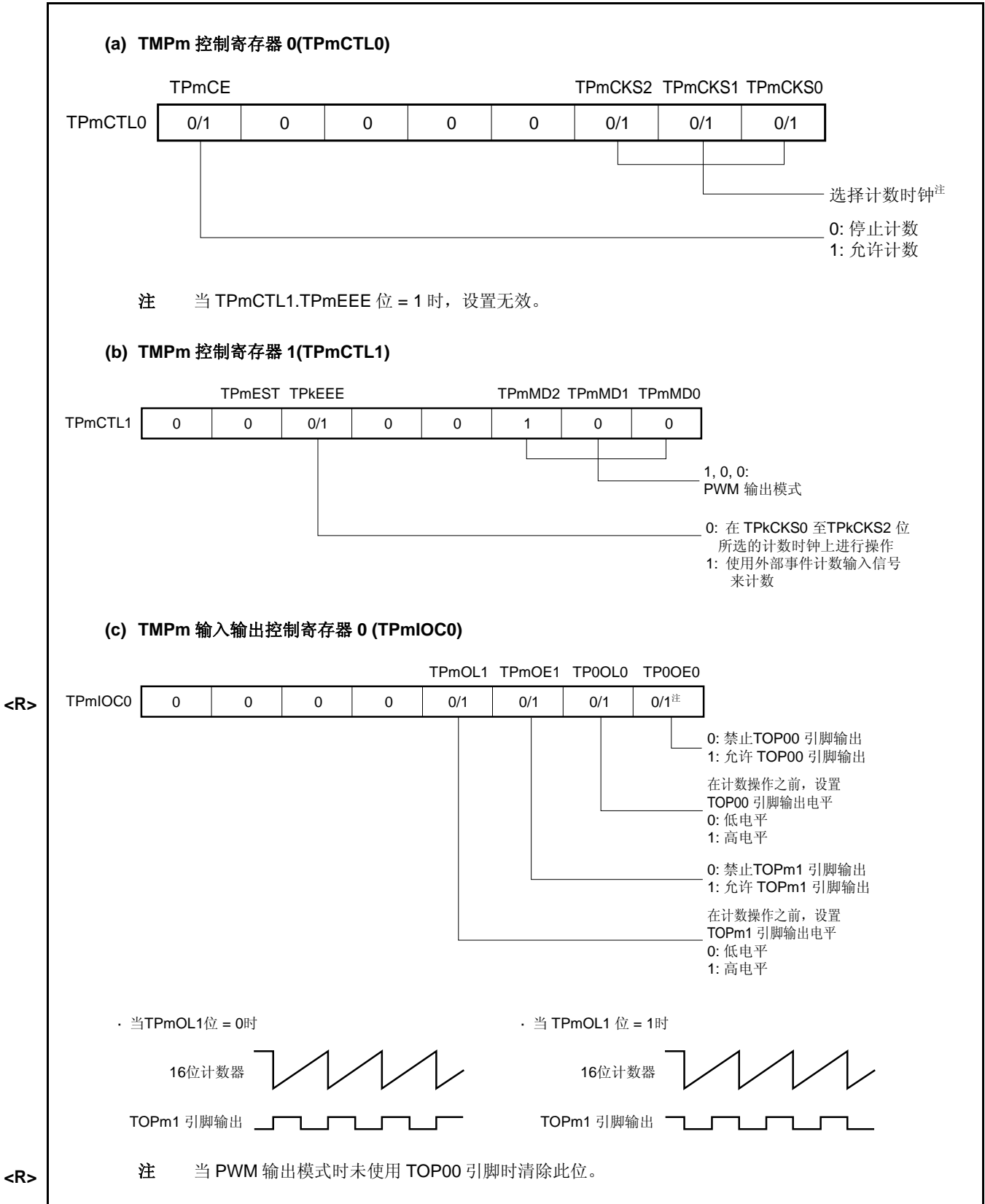
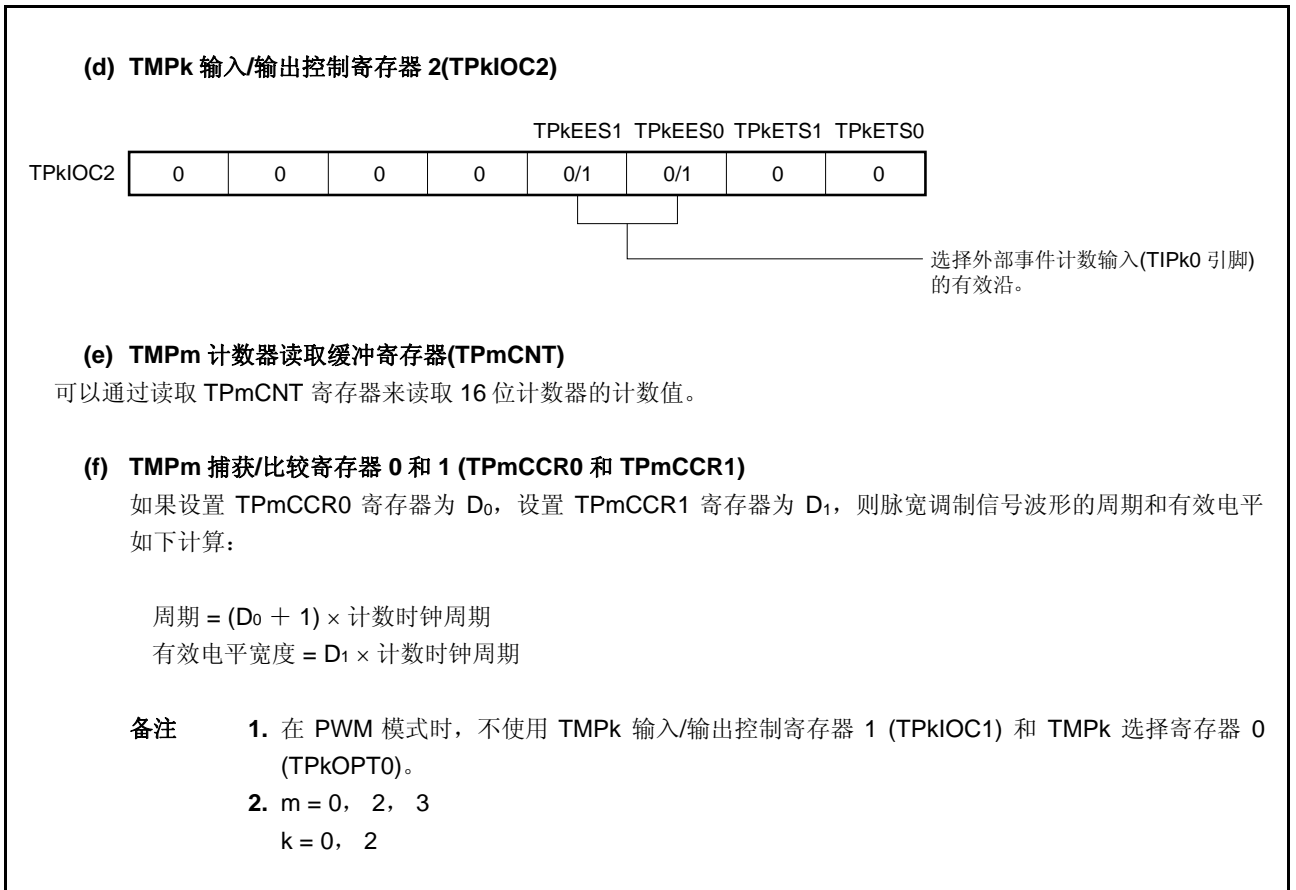


图 6-33. 脉宽调制信号输出模式时的寄存器设置(2/2)



(1) 脉宽调制信号输出模式下的工作流程

图 6-34. 脉宽调制信号输出模式时的软件处理流程(1/2)

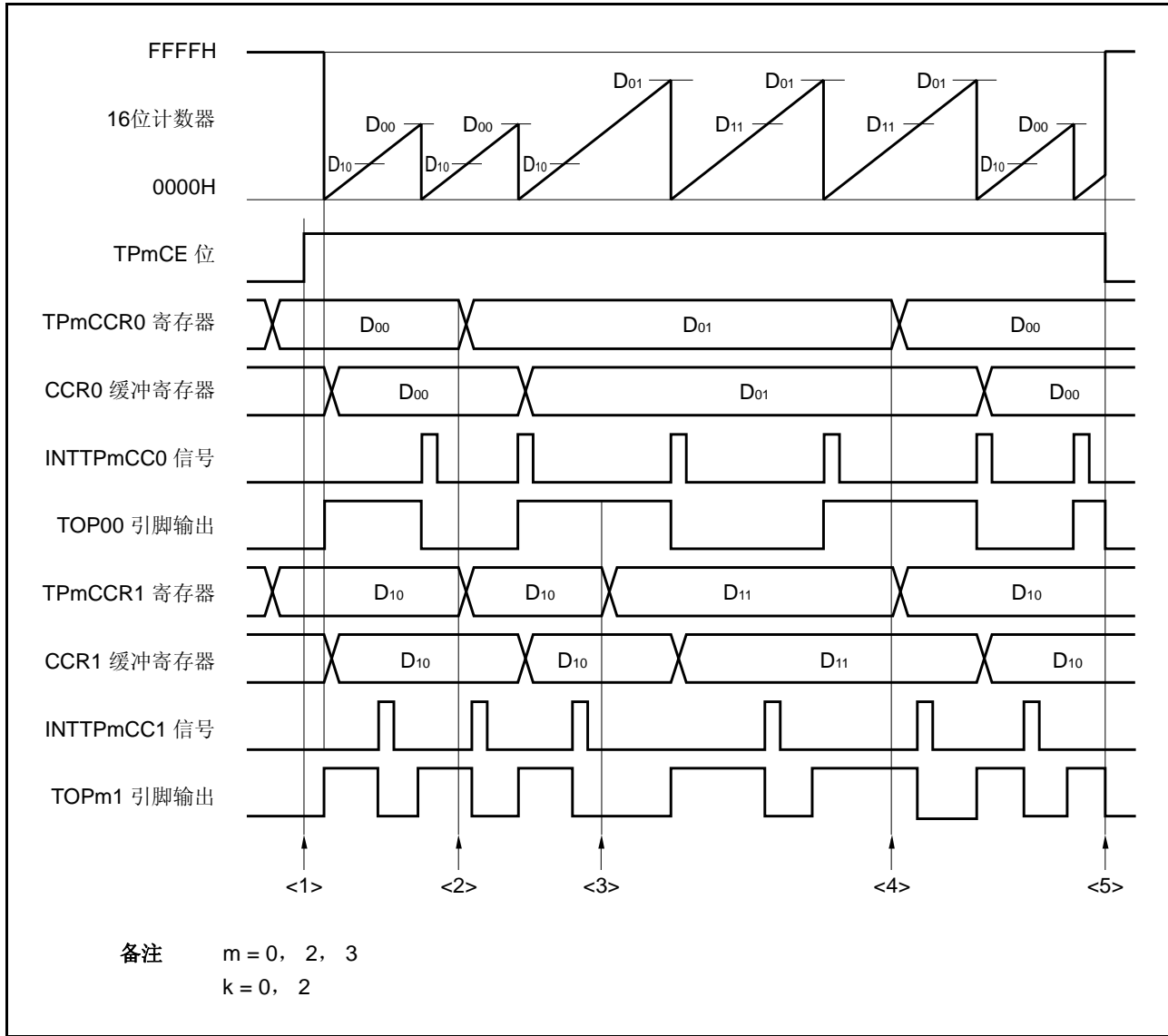
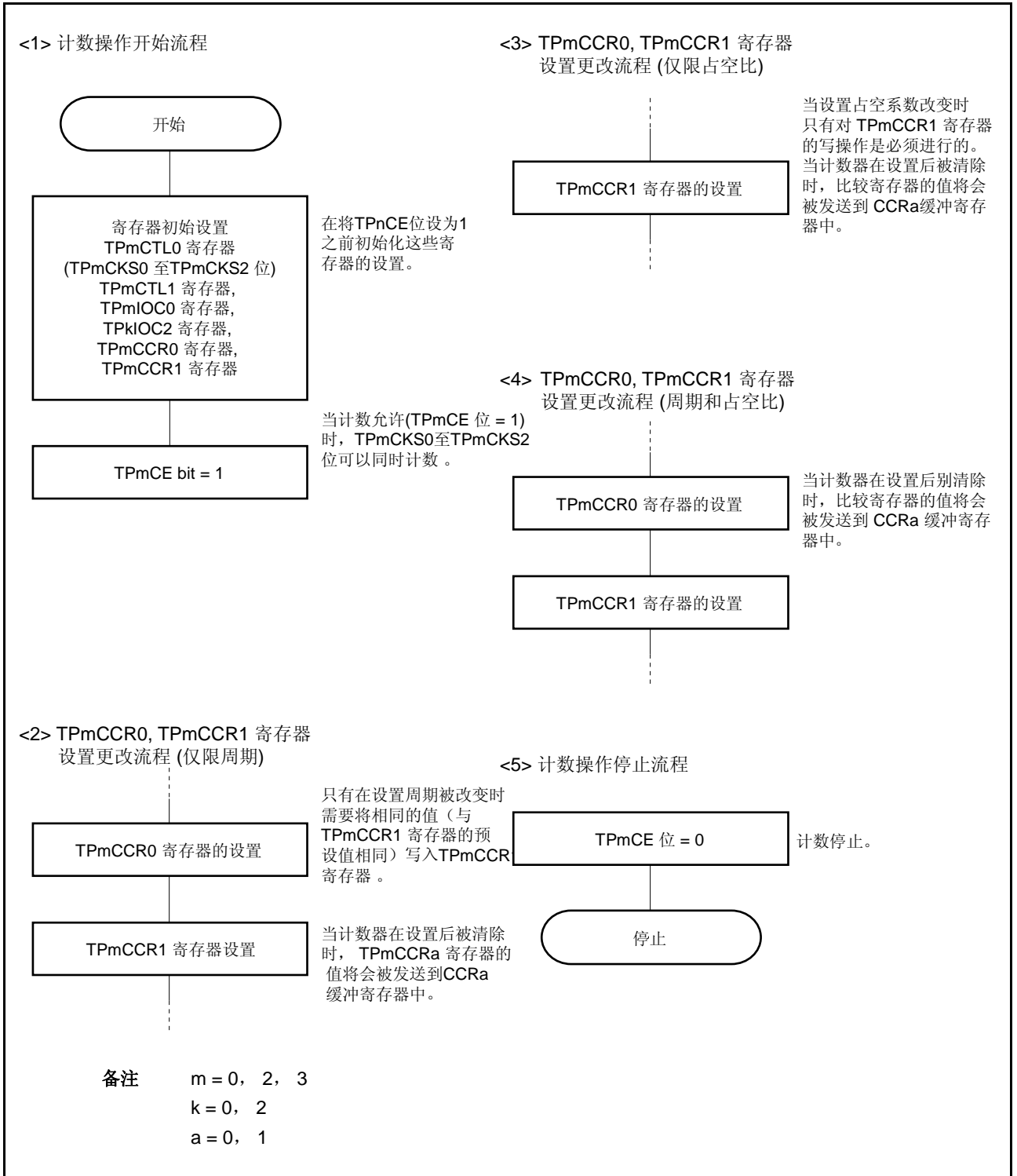


图 6-34. 脉宽调制信号输出模式时的软件处理流程(2/2)

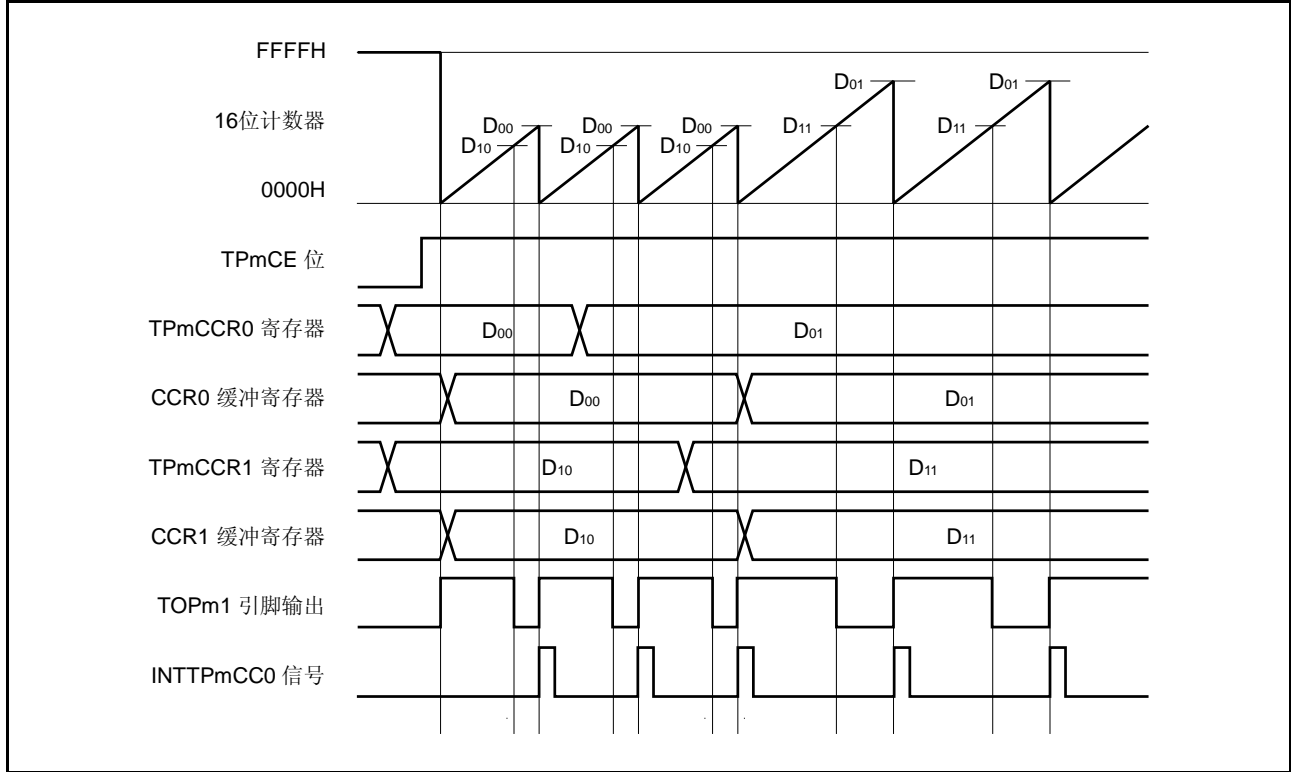


(2) 脉宽调制信号输出模式的操作时序

(a) 工作过程中改变脉冲宽度

计数器正在工作时，若改变 PWM 波形，应最后写入 TPmCCR1 寄存器。

在检测到 INTTPmCC0 信号后，写入 TPmCCR1 寄存器，然后再重写 TPmCCRa 寄存器。



要把数据从 TPmCCRa 寄存器传送到 CCRa 缓冲寄存器，必须写 TPmCCR1 寄存器。

此时要同时改变脉宽调制信号波形的周期和有效电平，应首先把周期设置给 TPmCCR0 寄存器，然后把有效电平设置给 TPmCCR1 寄存器。

只改变 PWM 波形的周期时，先将周期值写入 TPmCCR0 寄存器，再向 TPmCCR1 寄存器写入相同的值（与 TPmCCR1 寄存器的预置值相同）。

如果只改变脉宽调制信号波形的有效电平宽度（占空比），那就只需要设置 TPmCCR1 寄存器。

在数据写入 TPmCCR1 寄存器后，写入 TPmCCRa 寄存器的值会传送到 CCRa 缓冲寄存器，与 16 位计数器清零同步，并作为与 16 位计数器进行比较的值。

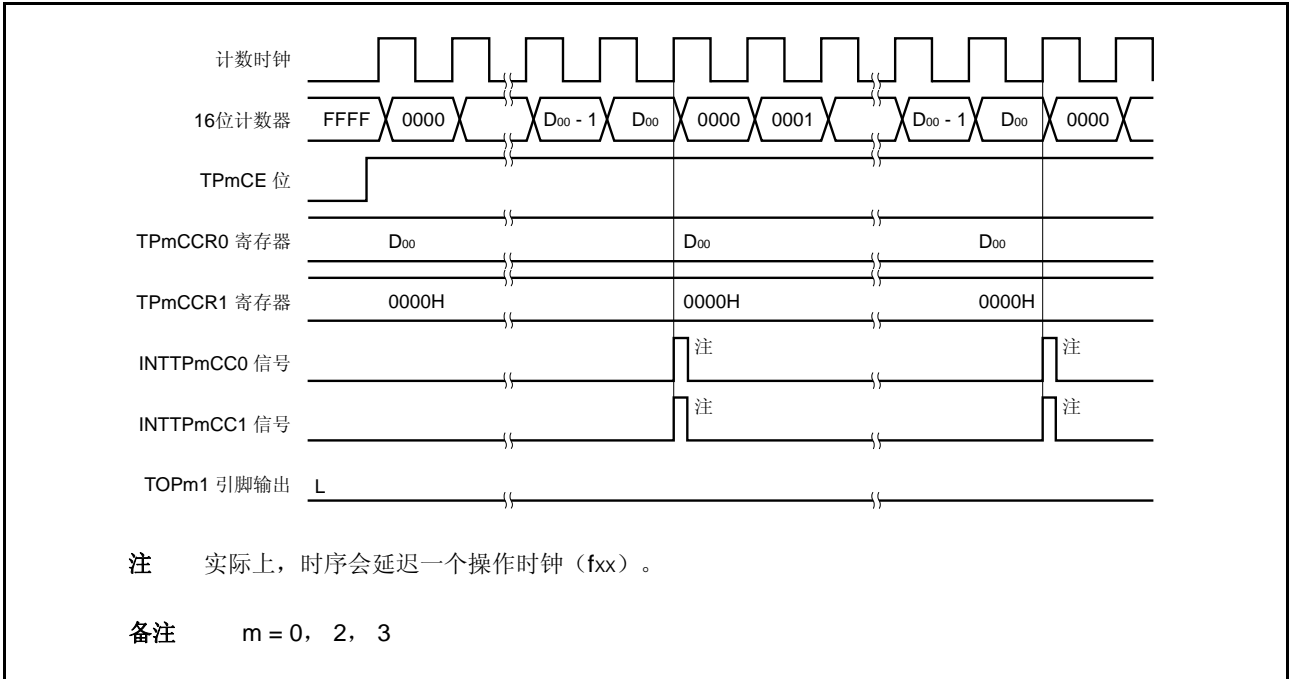
要在写入 TPmCCR1 寄存器一次之后，再次写 TPmCCR0 或 TPmCCR1 寄存器，应在 INTTPmCC0 信号产生之后进行。否则，CCRa 缓冲寄存器的值可能不确定，因为把数据从 TPmCCRa 寄存器传送到 CCRa 缓冲寄存器的时序与写 TPmCCRa 寄存器时序发生冲突。

备注 m = 0, 2, 3
 a = 0, 1

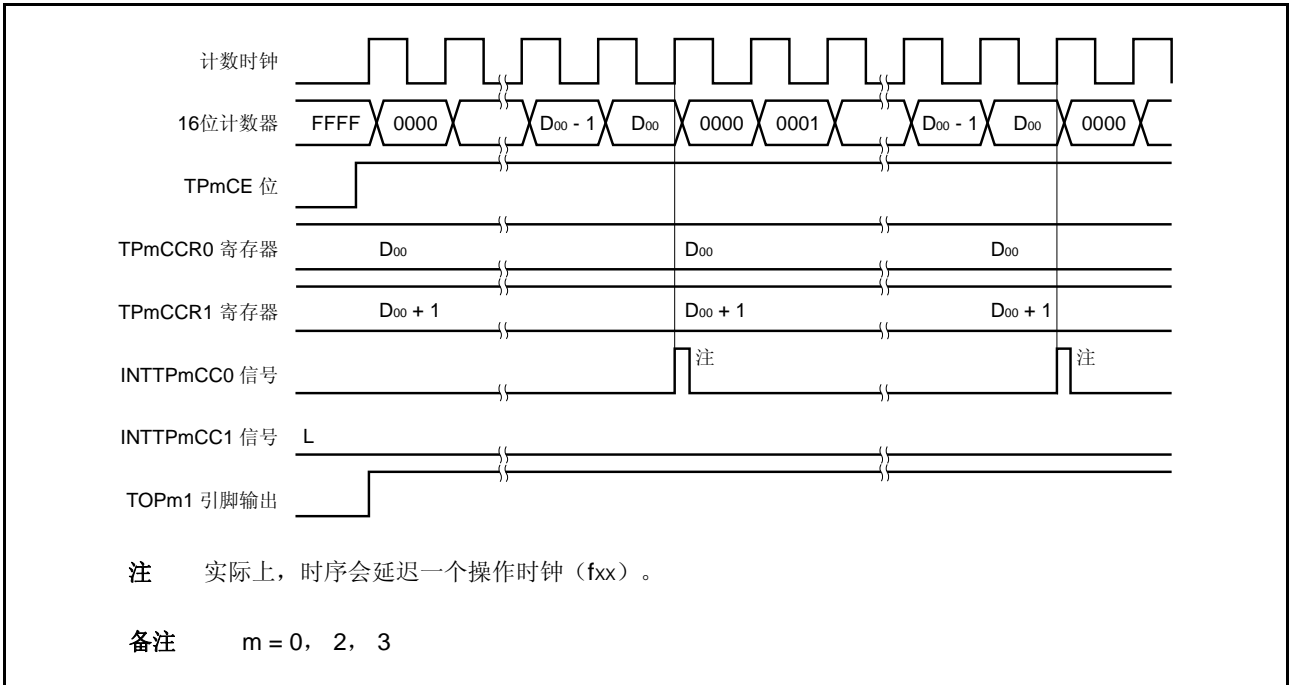
(b) 脉宽调制信号波形的 0%和 100% 输出

为了输出 0% 的波形，需要将 TPmCCR1 寄存器设置为 0000H。

在下次 16 位计数器的值与 CCR0 缓冲寄存器的值匹配后，16 位计数器将会被清除为 0000H 且会生成 INTTPmCC0 和 INTTPmCC1 信号。



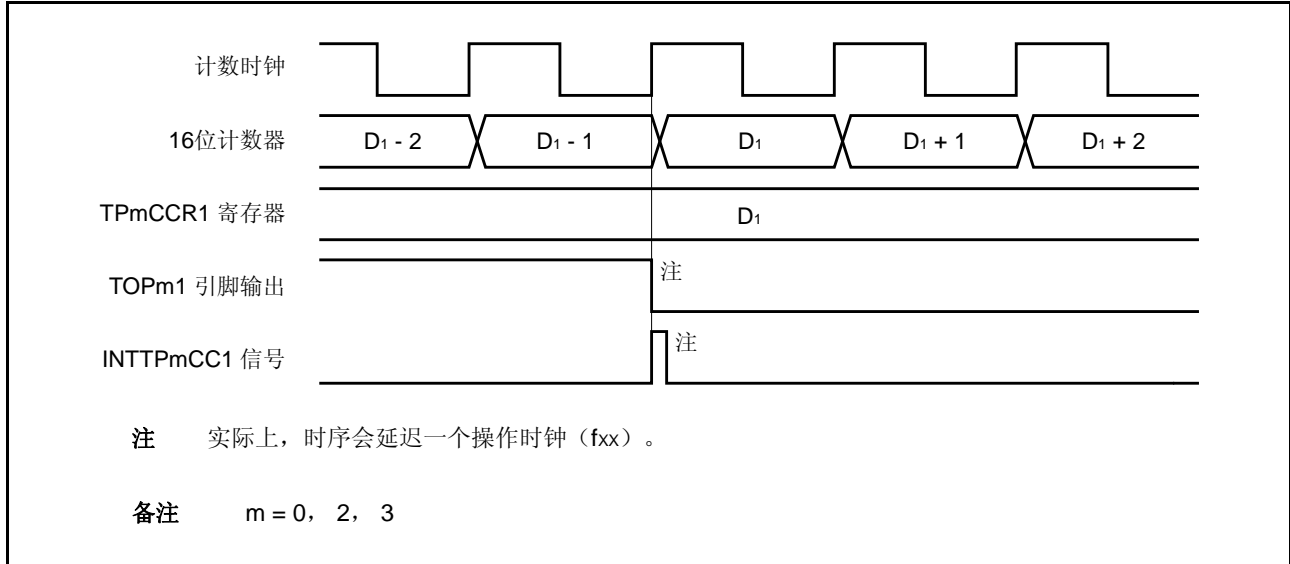
为了输出 100%的波形，需要将 TPmCCR1 寄存器的值设为：(TPmCCR0 寄存器的设定值 + 1)。如果 TPmCCR0 寄存器的设定值为 FFFFH，则不会产生 100%波形输出。



<R>

(c) 比较匹配中断请求信号(INTTPmCC1)的发生时序

在脉宽调制信号输出模式下，INTTPmCC1 信号的发生时序与其它模式下 INTTPmCC1 信号的时序有所不同；当 16 位计数器的计数值和 TPmCCR1 寄存器的值匹配时，产生 INTTPmCC1 信号。



通常情况下，在 16 位计数器的当前计数值与 TPmCCR1 寄存器的值匹配之后，INTTPmCC1 信号与和下一次计数同步发生。

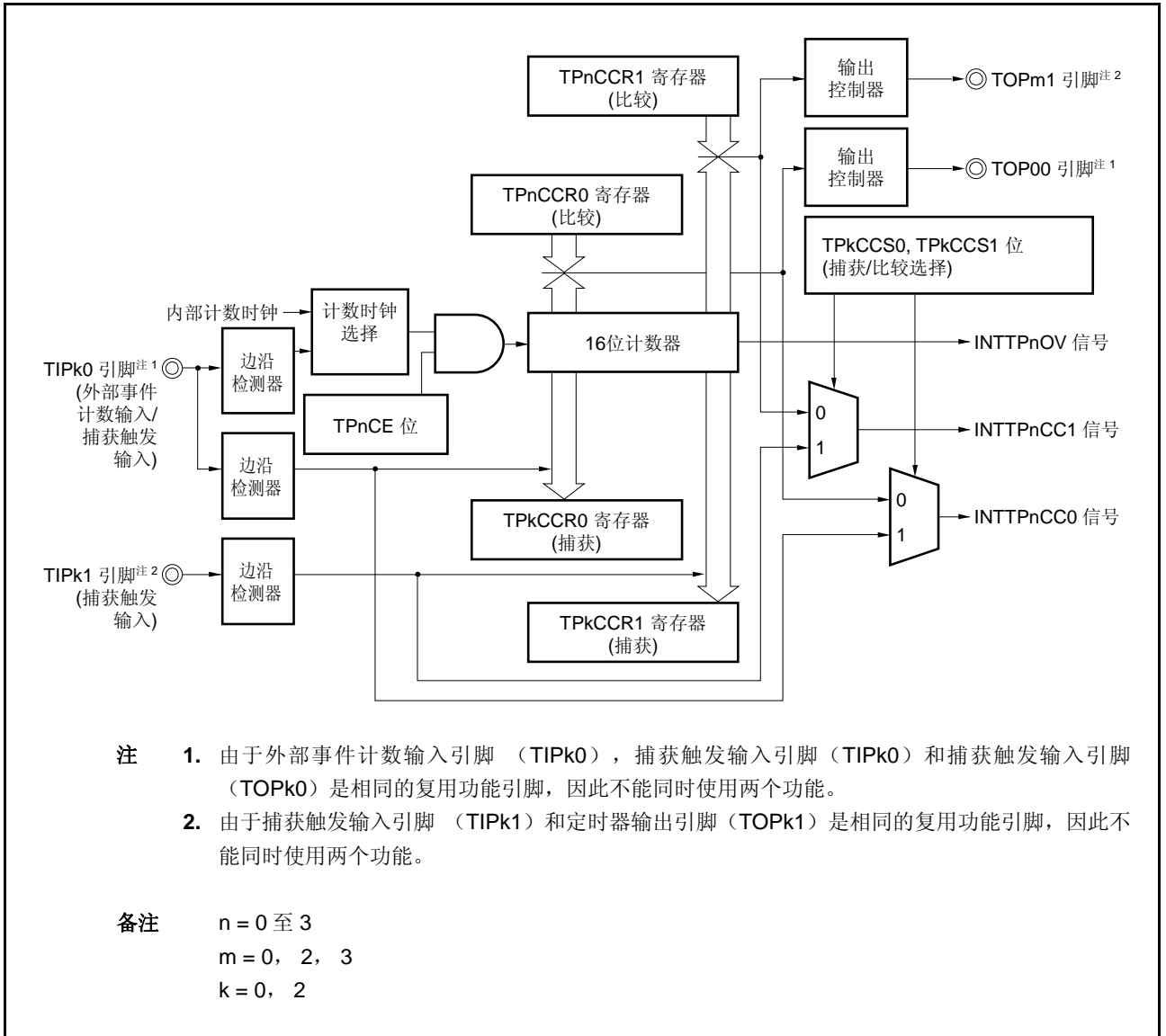
但是，在脉宽调制信号输出模式下，INTTPmCC1 信号会提早一个计数时钟周期产生。这是由于为了匹配 TOPm1 引脚输出信号的变化，时序发生了变化。

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

比较功能对 TMP0 至 TMP3 都有效。捕获功能仅对 TMP0 和 TMP2 有效。

在自由运行定时器模式下，当 TPnCTL0.TPnCE 位被置为 1 时，16 位定时器/事件计数器 P 开始计数。此时，TPkCCRa 寄存器和 TPkCCR1 寄存器根据 TPkOPT0.TPkCCS0 和 TPkOPT0.TPkCCS1 位的设置，可以用作比较寄存器或捕获寄存器。

图 6-35. 自由运行定时器模式的配置



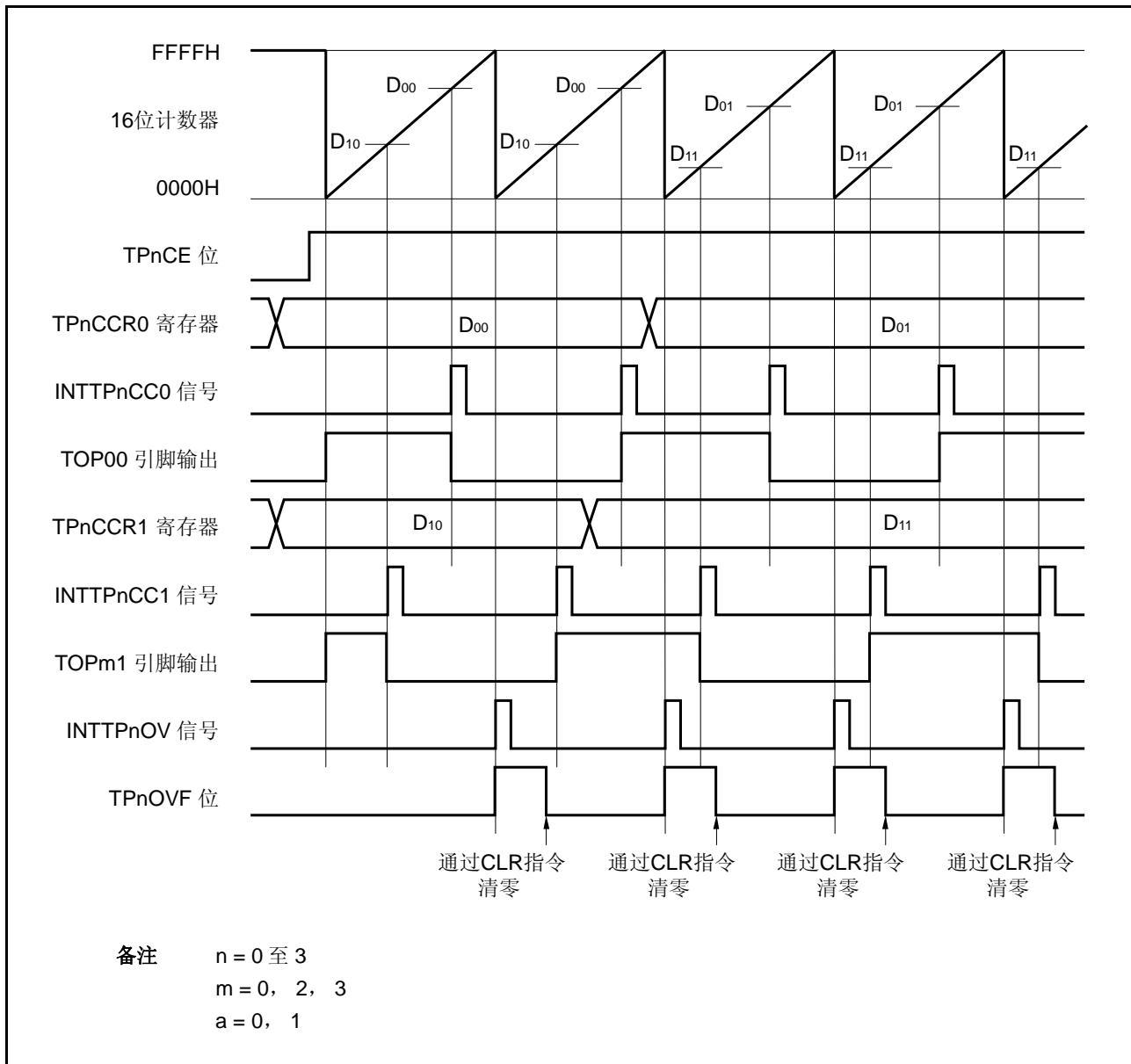
• 比较操作

当 TPnCE 位置为 1 时，16 位定时器/事件计数器 P 开始计数，TOP00 和 TOPm1 引脚的输出信号反相。此后当 16 位计数器的计数值与 TPnCCRa 寄存器的设置值匹配时，产生一个比较匹配中断请求信号 (INTTPnCCa)，而且 TOP00 引脚和 TOPm1 引脚的输出信号反相。

16 位计数器继续与计数时钟同步进行计数。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，继续进行计数。这种情况下，溢出标志 (TPnOPT0.TPnOVF 位) 也将被设置为 1。在确认溢出标志被设为 1 后通过软件使用 CLR 指令将溢出标志清零。

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

图 6-36. 自由运行定时器模式下的基本时序(比较功能)



- 捕获操作

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

16 位计数器继续与计数时钟同步进行计数。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（INTTPkOV），同时计数器清零为 0000H，继续进行计数。这种情况下，溢出标志（TPkOPT0.TPkOVF 位）也将被设置为 1。在确认溢出标志被设为 1 后通过软件使用 CLR 指令将溢出标志清零。

图 6-37. 自由运行定时器模式下的基本时序 (捕获功能)

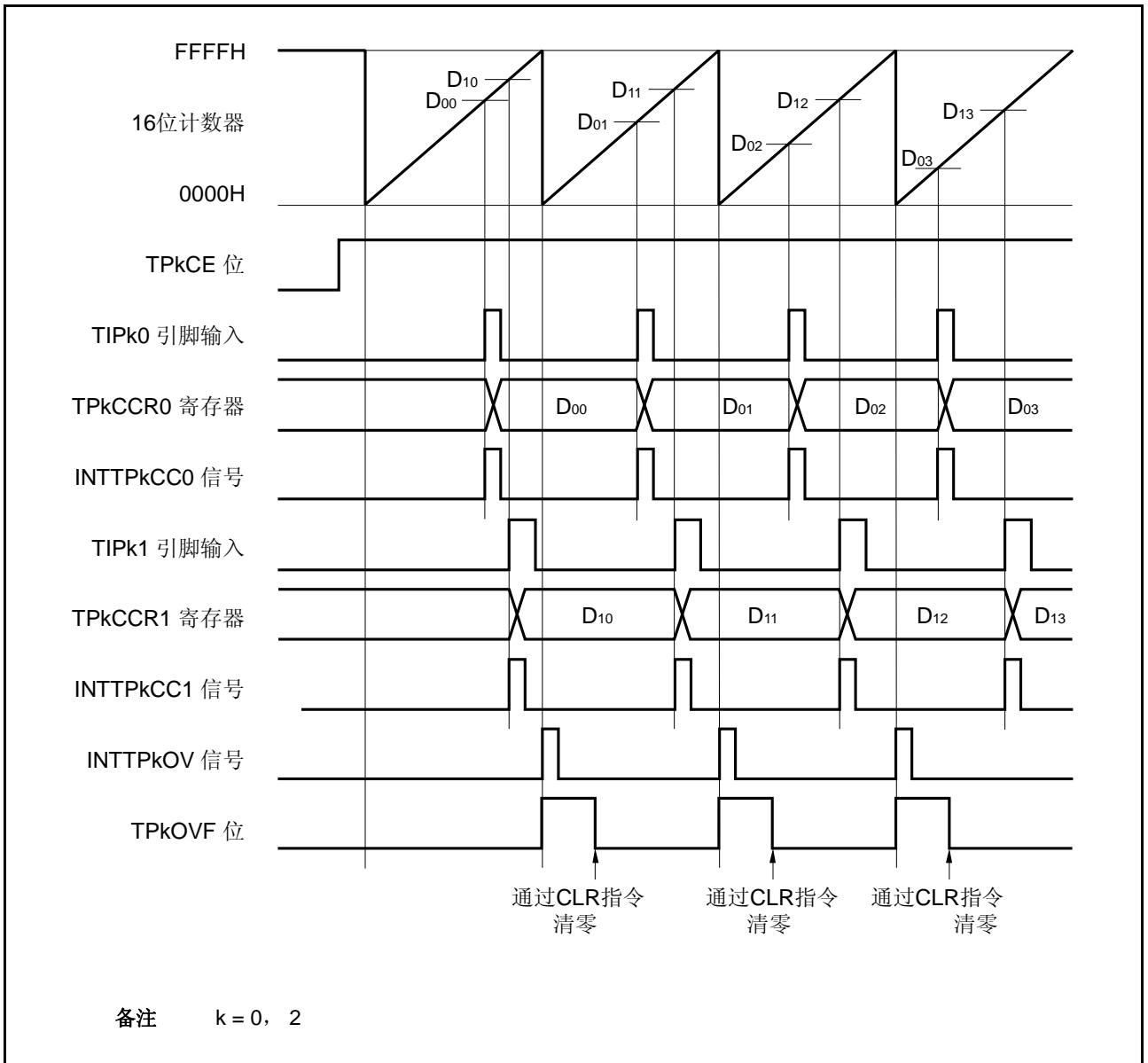


图 6-38. 自由运行定时器模式下的寄存器设置(1/2)

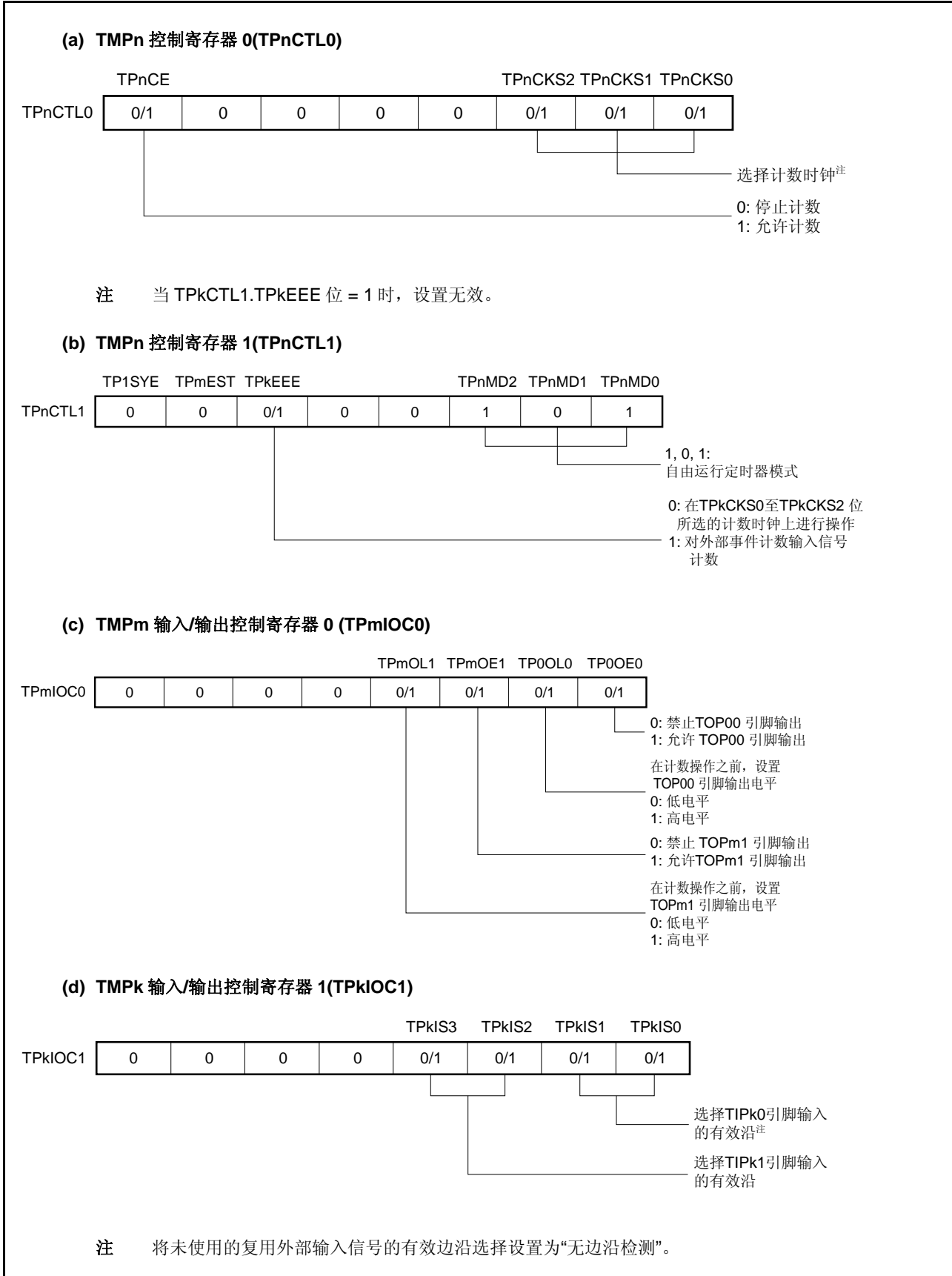
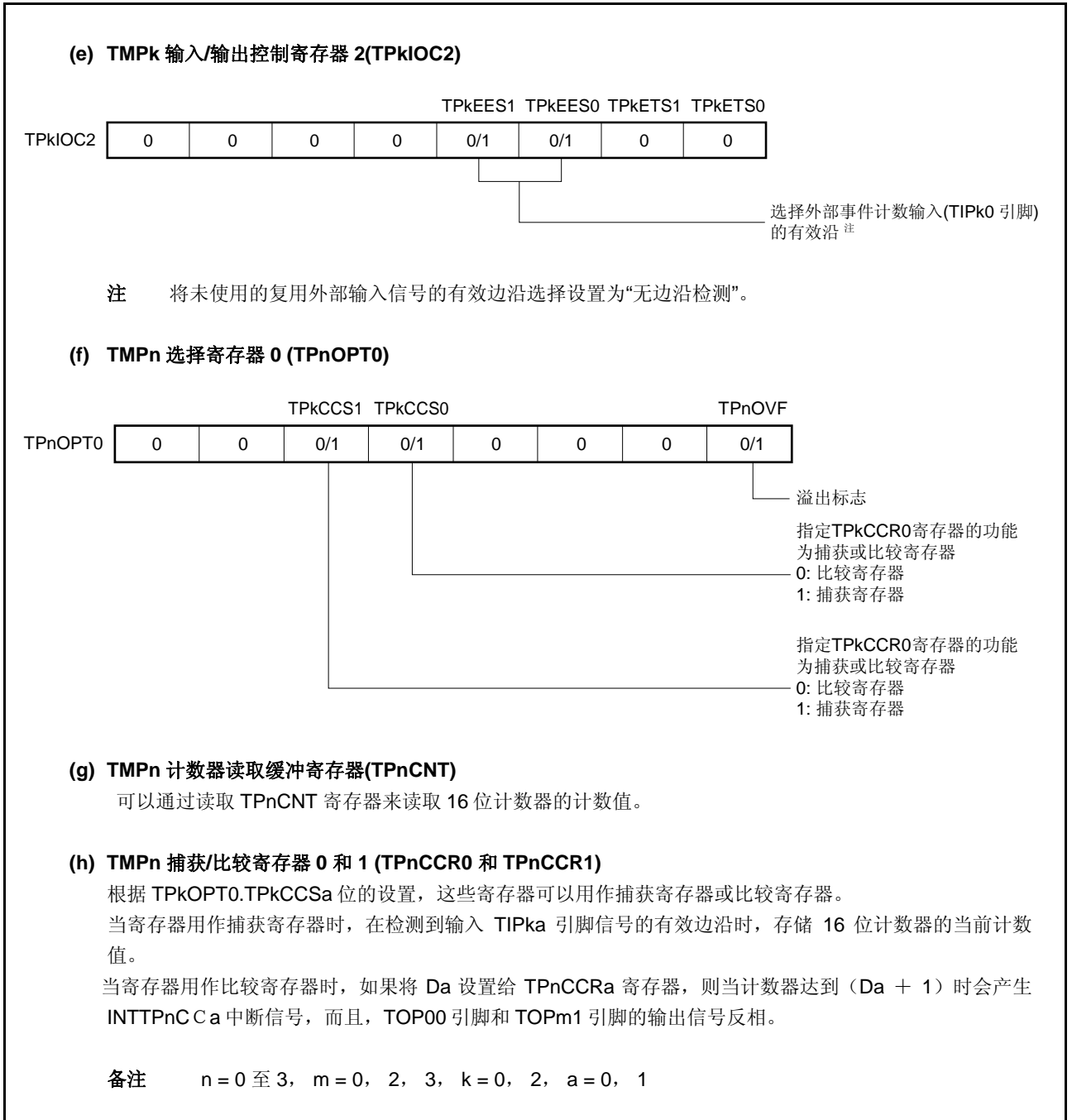


图 6-38. 自由运行定时器模式下的寄存器设置(2/2)



(1) 自由运行定时器模式时的操作流程

(a) 当捕获/比较寄存器用作比较寄存器时

图 6-39. 自由运行定时器模式时的软件处理流程(比较功能) (1/2)

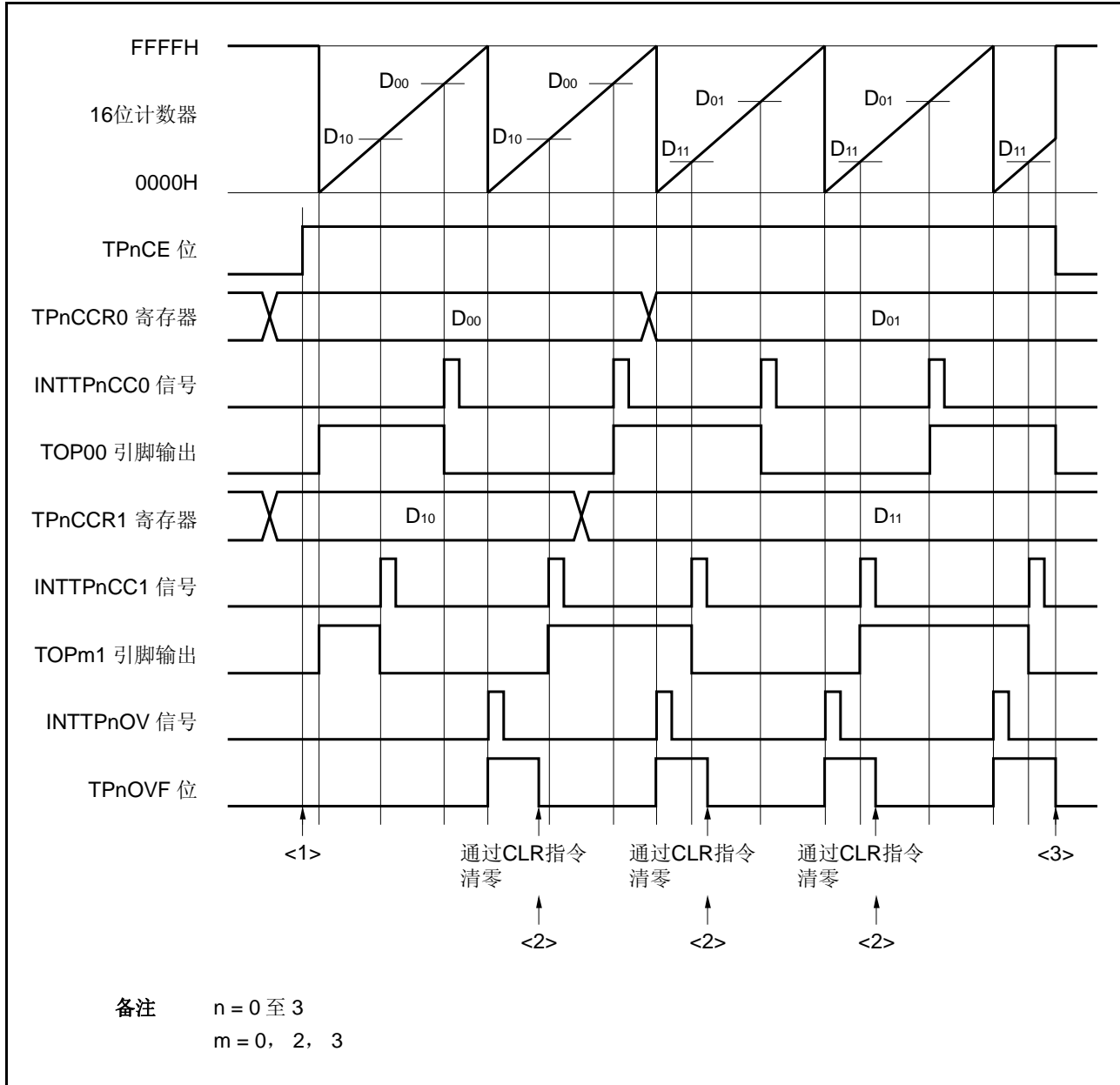
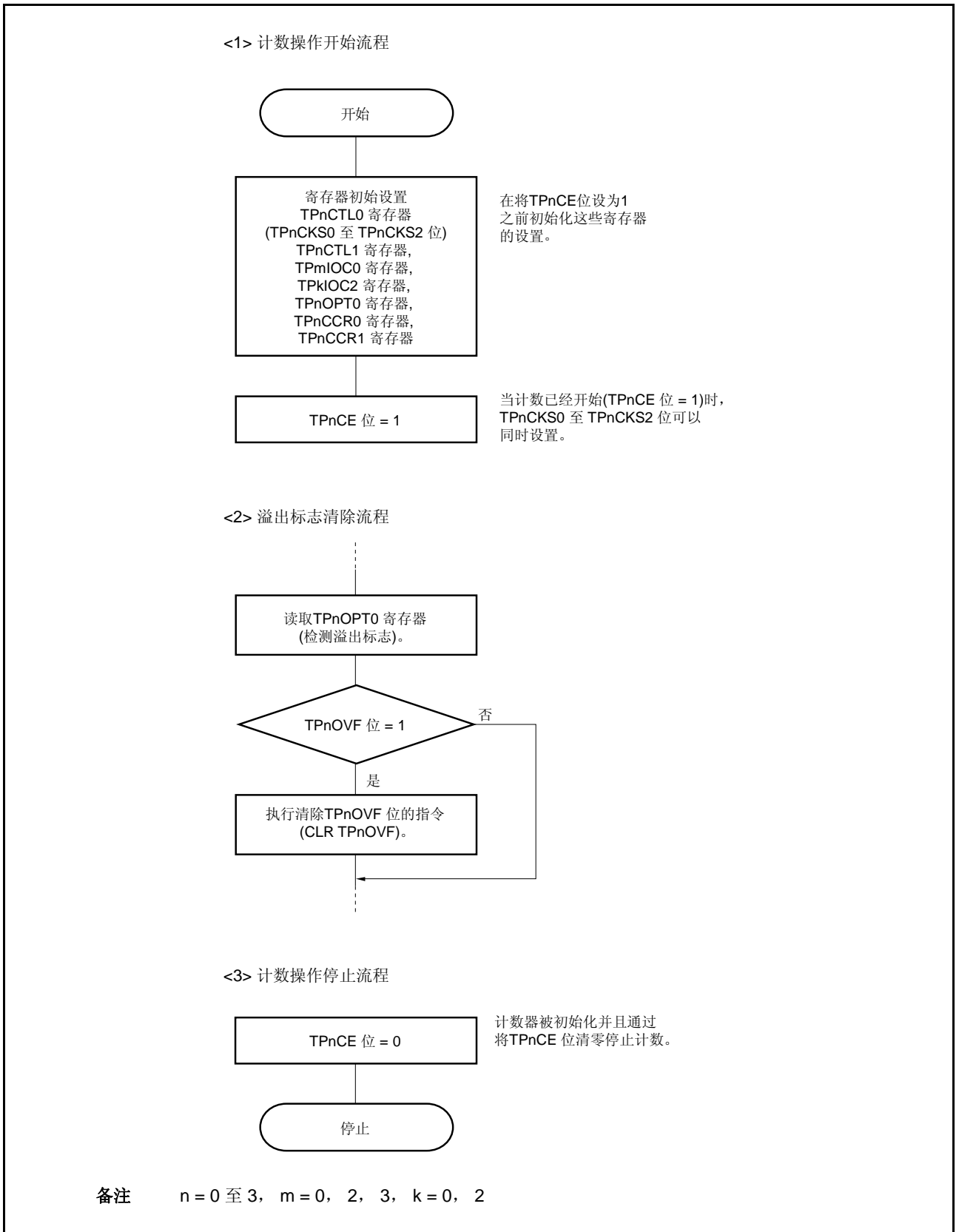


图 6-39. 自由运行定时器模式时的软件处理流程(比较功能) (2/2)



(b) 当捕获/比较寄存器用作捕获寄存器时

图 6-40. 自由运行定时器模式时的软件处理流程(捕获功能) (1/2)

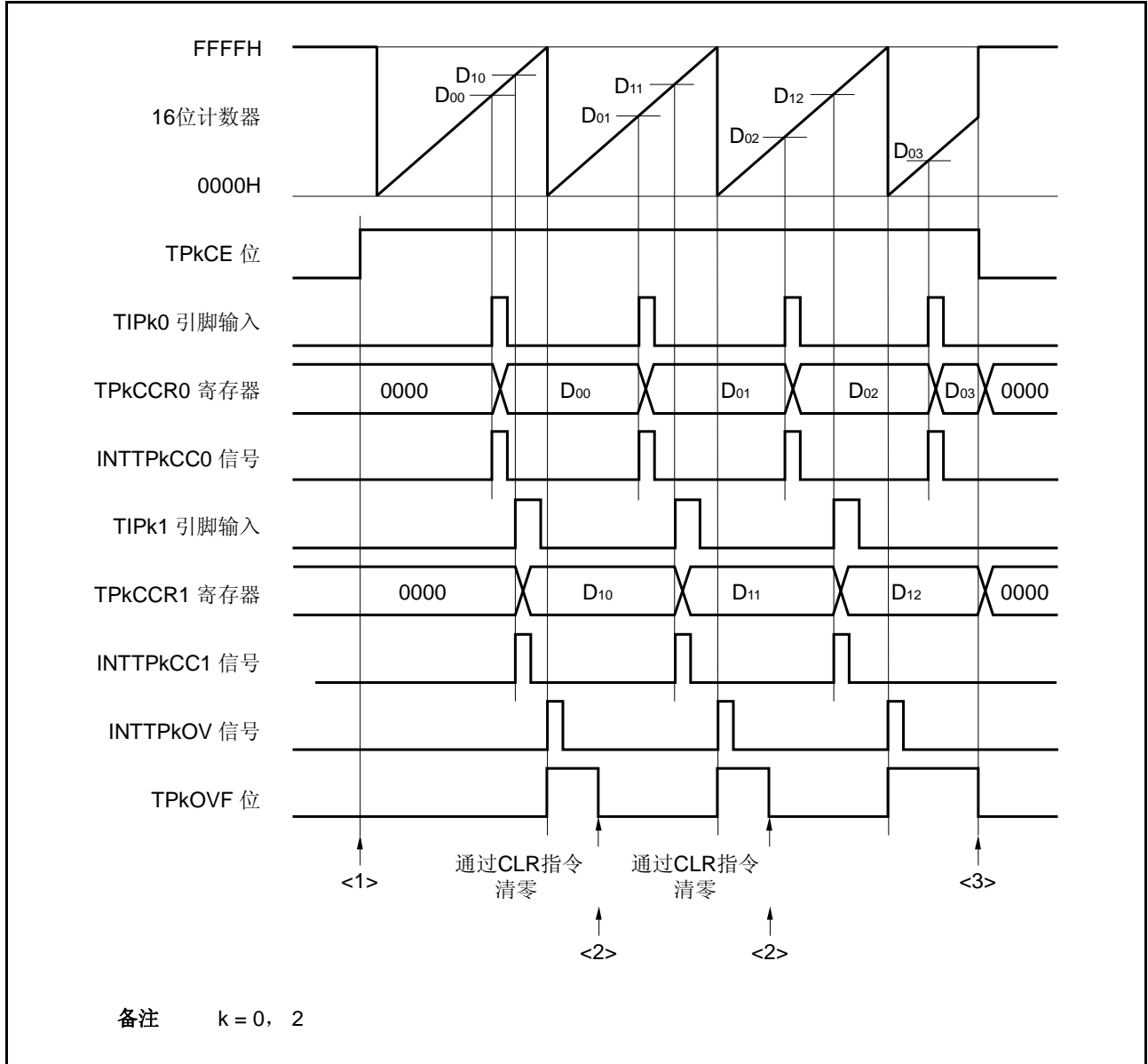
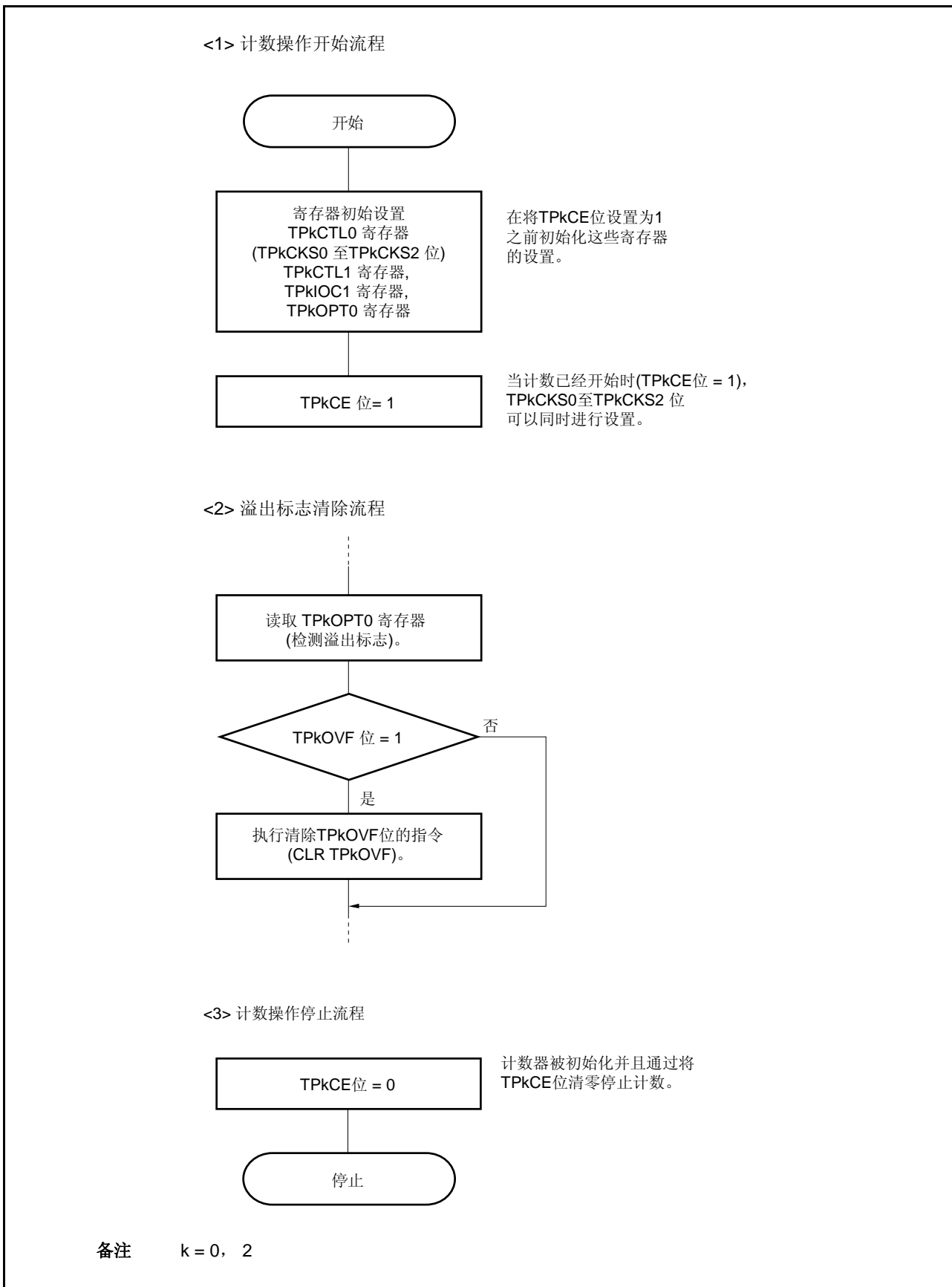


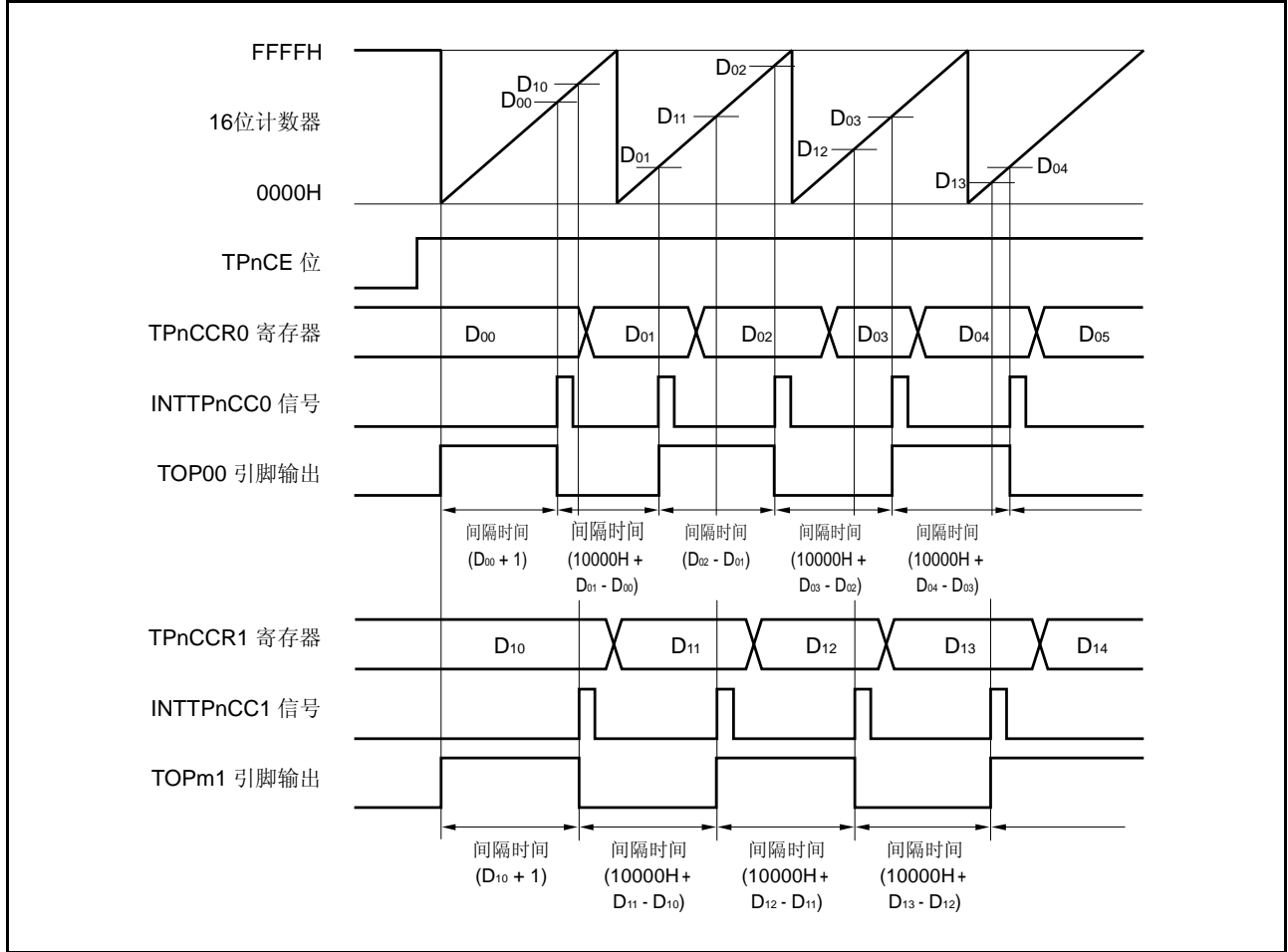
图 6-40. 自由运行定时器模式时的软件处理流程(捕获功能) (2/2)



(2) 自由运行定时器模式时的工作时序

(a) 使用比较寄存器的间隔定时操作

当 16 位定时器/事件计数器 P 用作间隔定时器而 TPnCCRa 寄存器用作比较寄存器时，在每次检测到 INTTPnCCa 信号后，有必要利用软件操作设置一个比较值，以便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 INTTPnCCa 信号后，必须在中断服务程序中重置相应的 TPnCCRa 寄存器的值。

重置 TPnCCRa 寄存器的设置值可由下式计算，其中“Da”是间隔时间：

比较寄存器最初设定值： $D_a - 1$

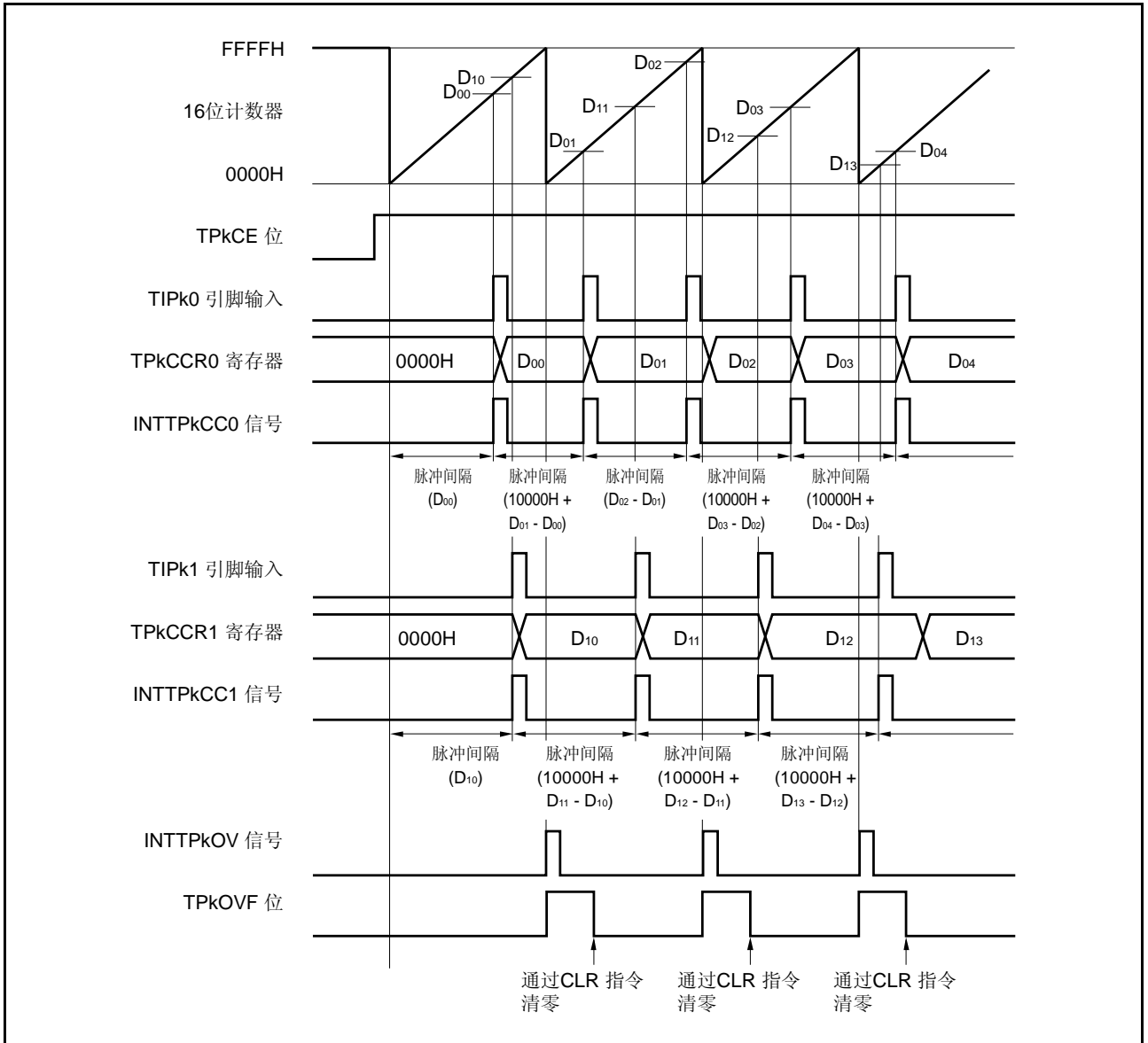
比较寄存器的第二次及其以后的设置值：上次设置值 + D_a

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

备注 $n = 0$ 至 3
 $m = 0, 2, 3$
 $a = 0, 1$

(b) 使用捕获寄存器进行脉冲宽度测量

在用 TPkCCRa 寄存器作为捕获寄存器进行脉宽测量时，每次检测到 INTTPkCCa 信号时都需要进行软件处理读取捕获寄存器及计算时间间隔。



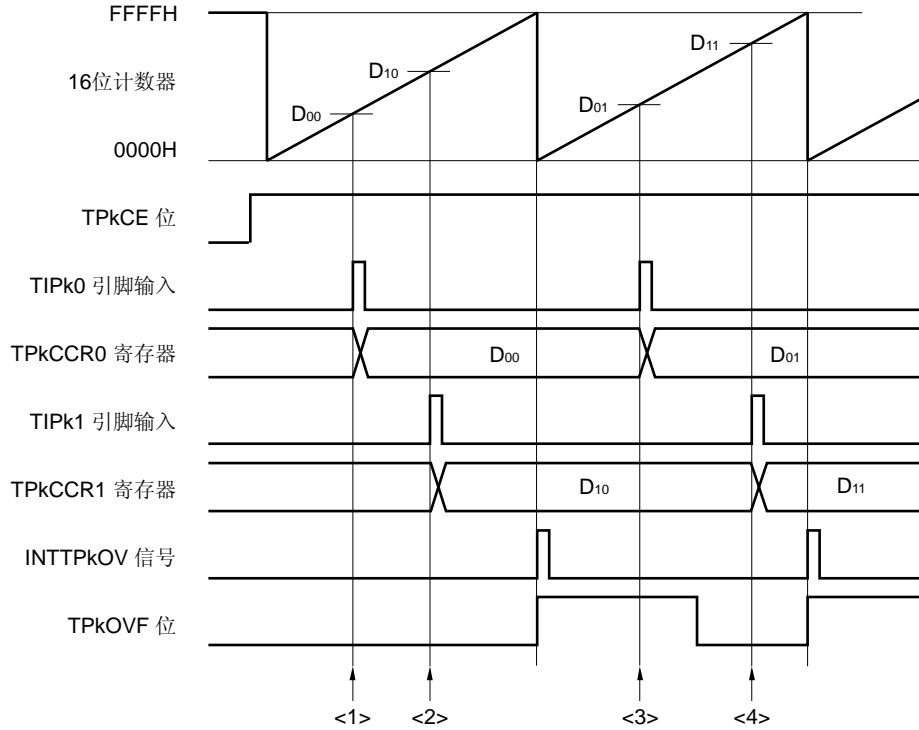
在自由运行定时器模式下执行脉宽测量时，可以用一个通道测量两个脉冲宽度。要测量脉冲宽度的话，可以通过与 INTTPkCCa 信号同步读取 TPkCCRa 寄存器的值，并计算本次读取值与上一次读取值之间的差，计算得到脉冲宽度。

备注 $k = 0, 2$
 $a = 0, 1$

(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，必须要格外小心。首先，给出一个错误处理的例子，如下所示。

使用两个捕获寄存器时处理错误的例子



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。

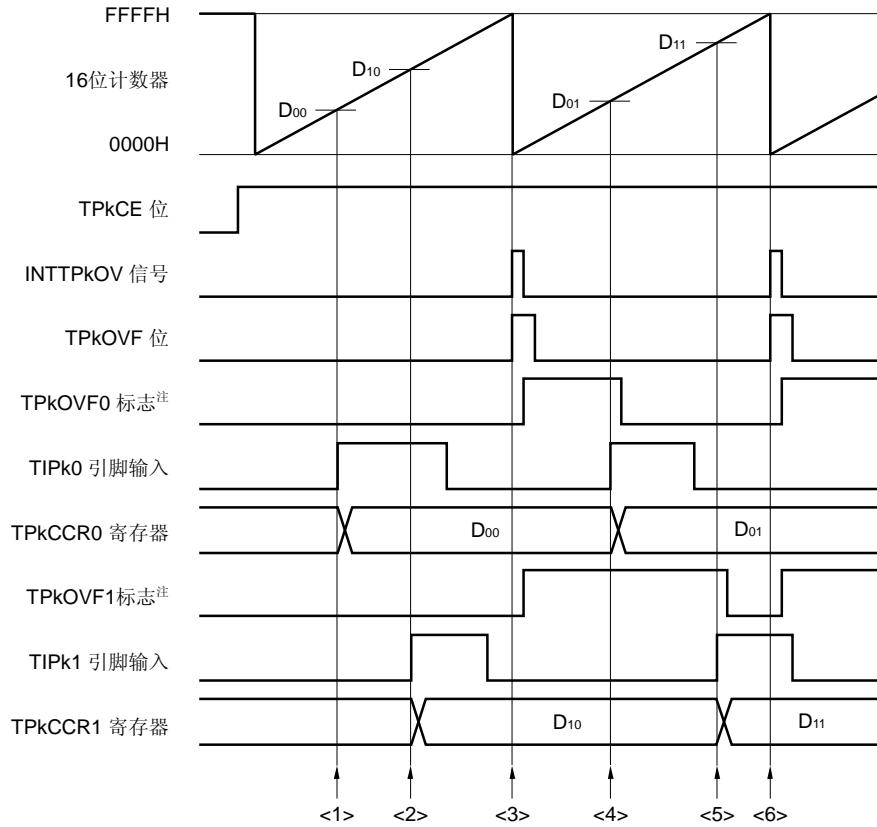
- <1> 读取 TPkCCR 0 寄存器（设置 TIPk0 引脚输入的默认值）。
- <2> 读取 TP k CCR1 寄存器（设置 TIPk1 引脚输入的默认值）。
- <3> 读取 TPkCCR0 寄存器。
读溢出标志。若溢出标志为 1，将其清为 0。
因为溢出标志为 1，所以脉冲宽度可以用 $(10000H + D_{01} - D_{00})$ 计算。
- <4> 读取 TPkCCR1 寄存器。
读溢出标志。因为此标志在<3>中清零了，所以，读取到 0。
因为溢出标志为 0，所以脉冲宽度可以用 $(D_{11} - D_{10})$ 计算。（不正确）

备注 k = 0, 2

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不会得到正确的脉冲宽度。

在使用两个捕获寄存器时，使用软件进行控制。关于如何使用软件，示例如下。

使用两个捕获寄存器时的例子 (使用溢出中断)



注 由软件对内部 RAM 中的 TPkOVF0 和 TPkOVF1 标志进行设置。

<1> 读取 TPkCCR0 寄存器 (设置 TIPk0 引脚输入的默认值)。

<2> 读取 TPkCCR1 寄存器 (设置 TIPk1 引脚输入的默认值)。

<3> 产生了溢出。在溢出中断服务程序中将 TPkOVF0 和 TPkOVF1 的标志设置为 1, 并将溢出标志清为 0。

<4> 读取 TPkCCR0 寄存器。

读取 TPkOVF0 标志。如果 TPkOVF0 标志为 1, 则将其清为 0。

因为 TPkOVF0 标志为 1, 所以脉冲宽度可以利用 $(10000H + D_{01} - D_{00})$ 计算得到。

<5> 读取 TPkCCR1 寄存器。

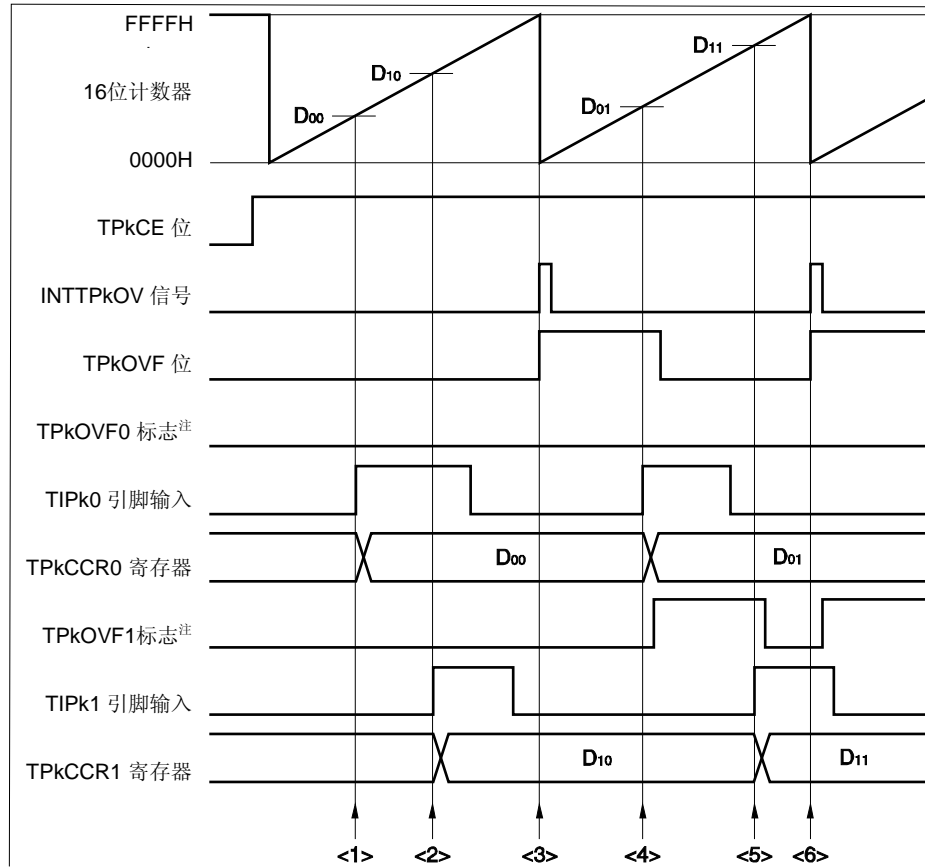
读取 TPkOVF1 标志, 如果 TPkOVF1 标志为 1, 则将其清为 0 (TPkOVF0 标志在<4>中已清零, 而 TPkOVF1 标志仍然保持为 1)。

因为 TPkOVF1 标志是 1, 所以脉冲宽度可以用 $(10000H + D_{11} - D_{10})$ 计算。(正确)

<6> 重复 <3>

备注 k = 0, 2

使用两个捕获寄存器时的例子 (不使用溢出中断)



注 由软件对内部 RAM 中的 TPkOVF0 和 TPkOVF1 标志进行设置。

<1> 读取 TPkCCR0 寄存器（设置 TIPk0 引脚输入的默认值）。

<2> 读取 TPkCCR1 寄存器（设置 TIPk1 引脚输入的默认值）。

<3> 出现溢出，软件不进行任何处理。

<4> 读取 TPkCCR0 寄存器。

读溢出标志。若溢出标志为 1，则只把 TPkOVF1 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 $(10000H + D_{01} - D_{00})$ 计算。

<5> 读取 TPkCCR1 寄存器。

读溢出标志。因为溢出标志在<4>中清零了，所以读到 0。

读取 TPkOVF1 标志。如果 TPkOVF1 标志为 1，则将其清为 0。

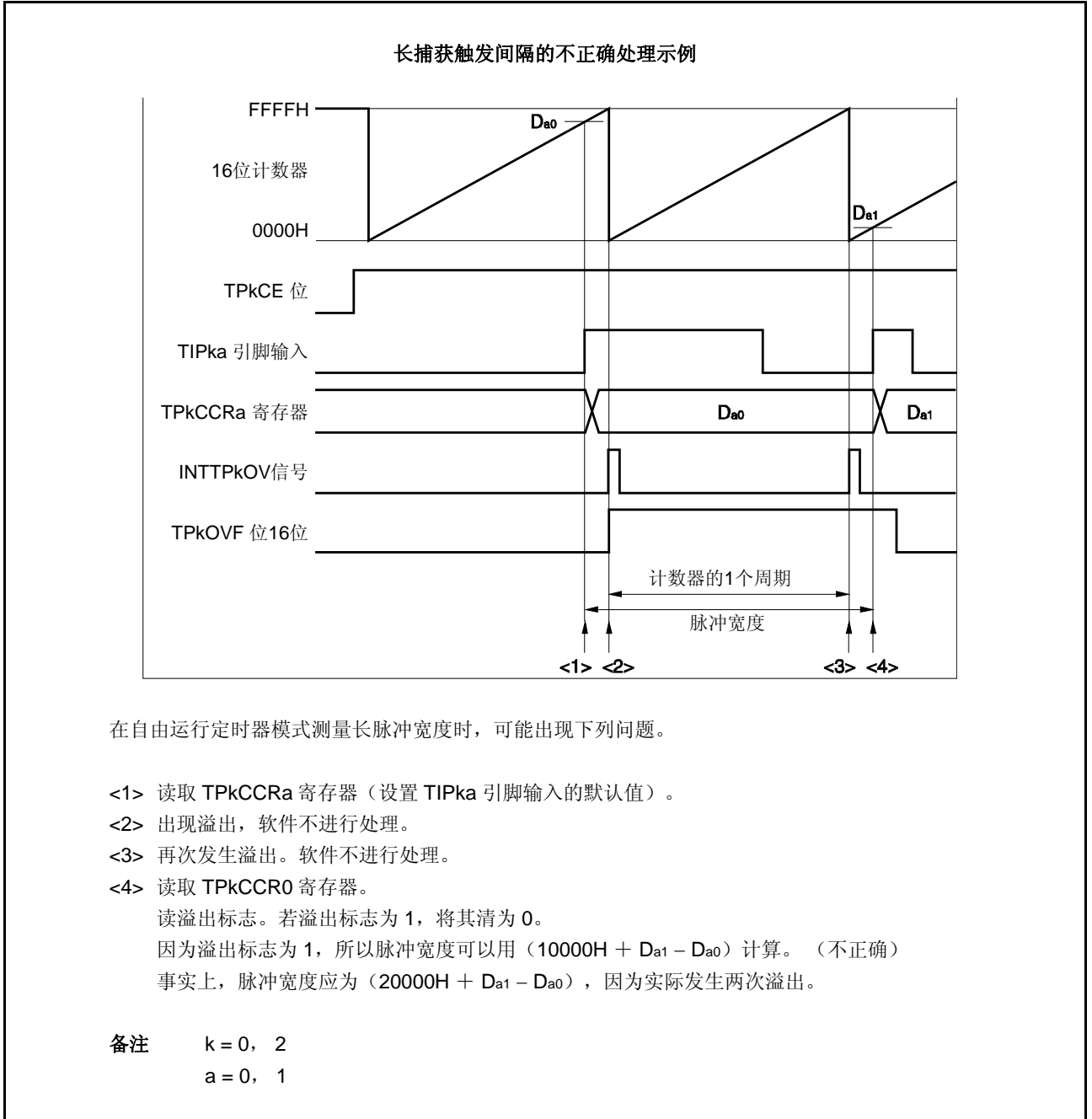
因为 TPkOVF1 标志是 1，所以脉冲宽度可以用 $(10000H + D_{11} - D_{10})$ 计算。（正确）

<6> 重复 <3>

备注 k = 0, 2

(d) 长捕获触发间隔时对溢出的处理

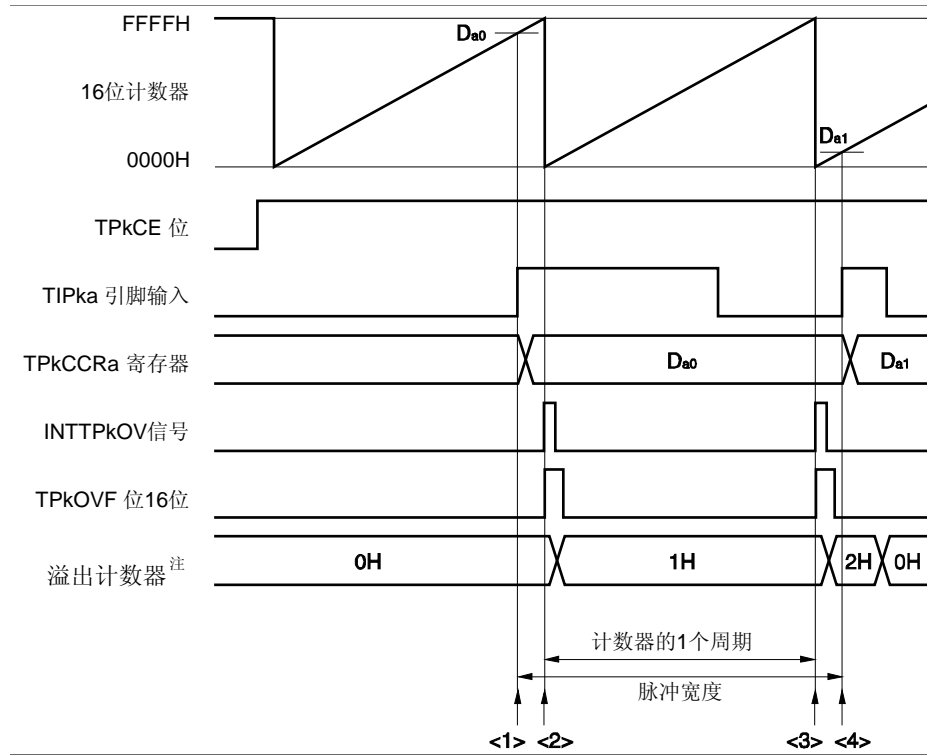
如果脉冲宽度大于 16 位计数器的一个计数循环，则必须加以注意，因为从第一个捕获触发到下一个之间可能发生不止一次溢出。首先，如下所示，是一个不正确处理的例子。



如果长捕获触发间隔期间，发生了两次或更多次的溢出，则无法获得正确的脉冲宽度。

如果捕获触发间隔为长间隔（相对于计数循环），应减慢计数时钟以延长 16 位计数器的循环周期，或者使用软件加以解决。关于如何使用软件，示例如下。

长捕获触发间隔时的举例



注 内部 RAM 中的溢出计数器由软件任意设置。

- <1> 读取 TPkCCRa 寄存器（设置 TIPka 引脚输入的默认值）。
- <2> 发生溢出，增加溢出计数器的值（+1），并在溢出中断服务程序中把溢出标志清为 0。
- <3> 再次发生溢出。增加溢出计数器的值（+1），并在溢出中断服务程序中把溢出标志清为 0。
- <4> 读取 TPkCCRa 寄存器。
读取溢出计数器。
→ 当溢出计数器为“N”时，脉冲宽度可用 $(N \times 10000H + D_{a1} - D_{a0})$ 计算。
在本例中，脉冲宽度为 $(20000H + D_{a1} - D_{a0})$ ，因为溢出发生了两次。
最后，溢出计数器清零（0H）。

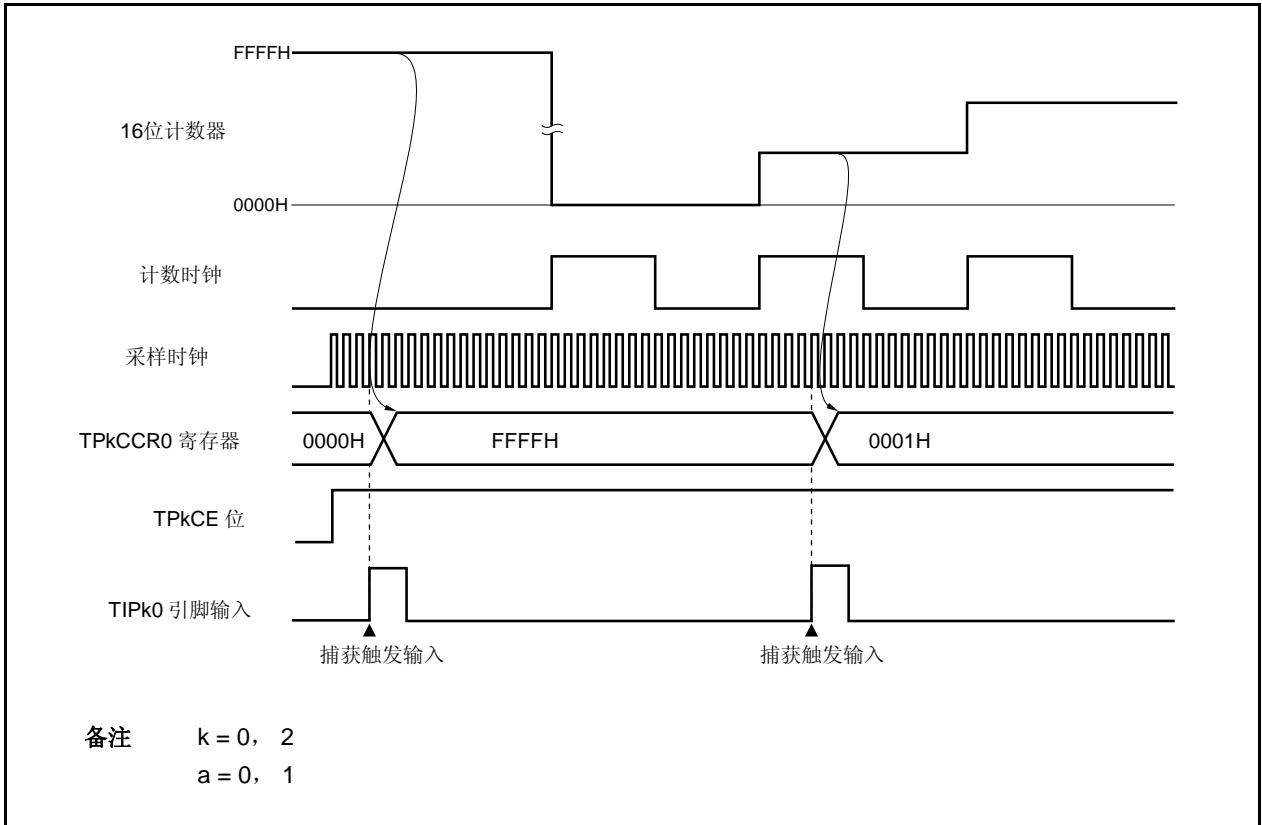
备注 k = 0, 2
a = 0, 1

(e) 清除溢出标志

在 TPnOVF 为 1 时读取该位后，使用 CLR 指令将 TPnOVF 位清零可以将溢出标志清零，而在 TPnOVF 位为 1 时读取该位后，通过将 8 位数据（位 0 为 0）写入 TPnOPT0 寄存器也可以将溢出标志清零。

(3) 捕获操作的注意事项

如果使用捕获操作并且将低速时钟选择为计数时钟时，如果在 TPkCTL0.TPkCE 位置 1 之后捕获触发器被立即输入，那么可以被捕获到 TPkCCRa 寄存器中的是 FFFFH，而不是 0000H。



6.6.7 脉冲宽度测量模式 (TPkMD2 至 TPkMD0 位 = 110)

该模式仅在 TMP0 和 TMP2 中有效。

在脉冲宽度测量模式下，当 TPkCTL0.TPkCE 位置为 1 时，16 位计时器/事件计数器 P 开始计数。每次检测到 TIPka 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 TPkCCRa 寄存器中，且 16 位计数器清零为 0000H。

有效沿之间时间间隔，可以在出现捕获中断请求信号 (INTTPkCCa) 之后，通过读取 TPkCCRa 寄存器的值来计算。

如图 6-42 所示，选择 TIPk0 或 TIPk1 引脚作为操作触发输入引脚并使用 TPkIOC1 寄存器将未使用的引脚设置为“无边沿检测”。

<R>

图 6-41. 脉冲宽度测量模式的配置

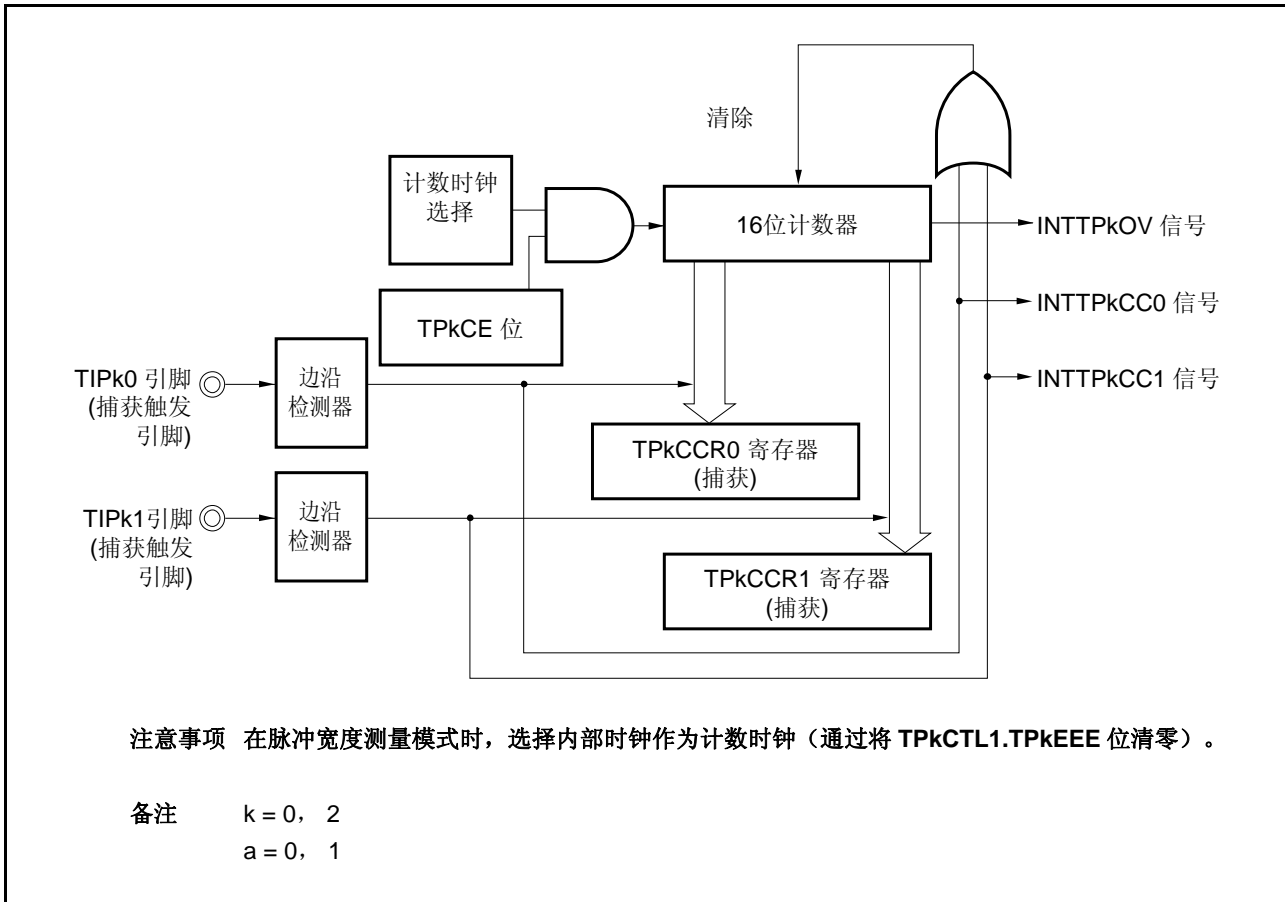
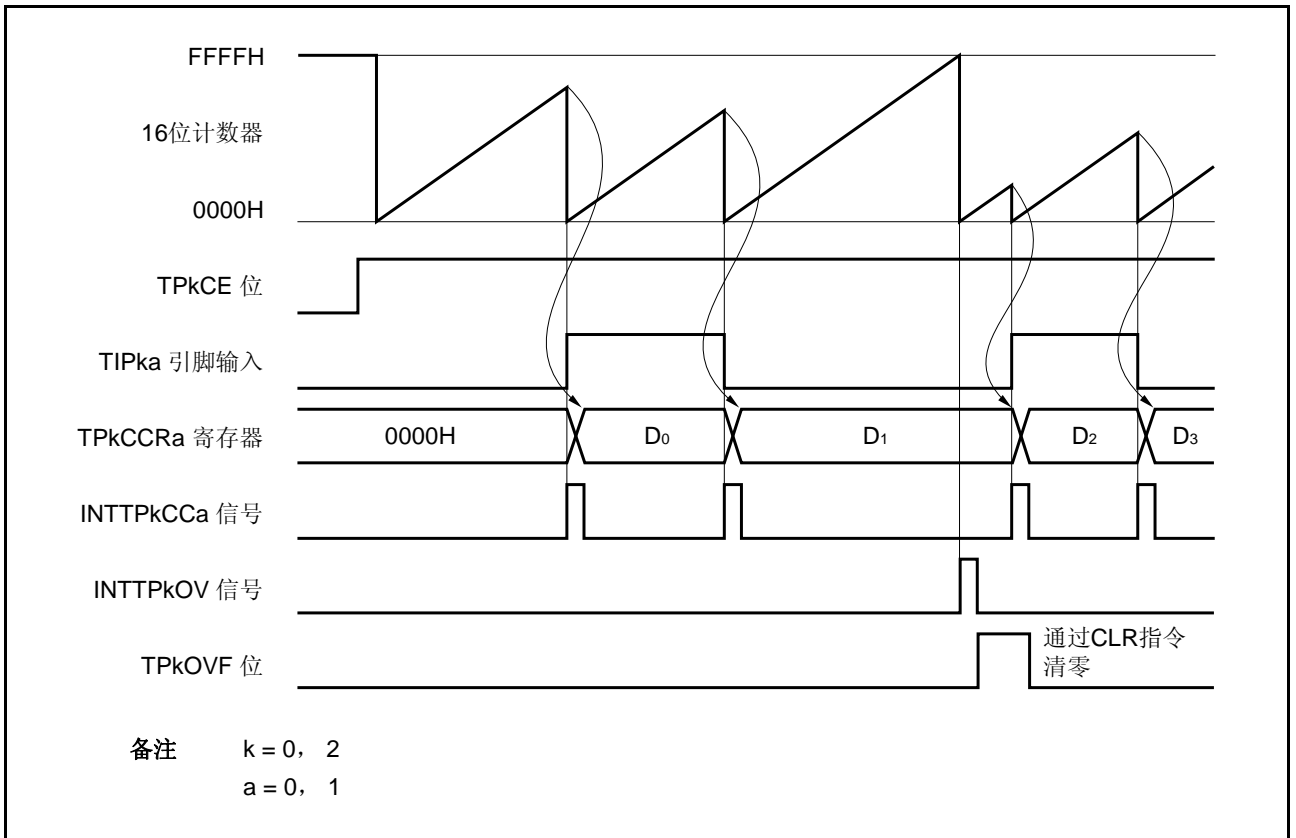


图 6-42. 脉冲宽度测量模式的基本时序



当 TPkCE 位置为 1 时，16 位计数器开始计数。此后检测到 TIPka 引脚输入信号的有效边沿时，16 位计数器的计数值存储在 TPkCCRa 寄存器中，16 位计数器清零为 0000H，并生成一个捕获中断请求信号（INTTPkCCa）。

脉冲宽度计算如下。

$$\langle R \rangle \quad \text{脉冲宽度} = \text{捕获值} \times \text{计数时钟周期}$$

即使 16 位计数器一直计数到 FFFFH，TIPnm 引脚输入信号一直没有有效边沿，仍然会在下一个计数时钟时产生溢出中断请求信号（INTTPkOV），且计数器清零为 0000H，计数器继续计数。此时，溢出标志（TPkOPT0.TPkOVF 位）也被置为 1。使用软件执行 CLR 指令将溢出标志清为 0。

若溢出标志设置为 1，则脉冲宽度可计算如下。

$$\text{脉冲宽度} = (10000\text{H} \times \text{TPkOVF 位被置为 1 的次数} + \text{捕获值}) \times \text{计数时钟周期}$$

$\langle R \rangle$

$$\text{备注} \quad k = 0, 2 \\ a = 0, 1$$

图 6-43. 脉冲宽度测量模式时的寄存器设置(1/2)

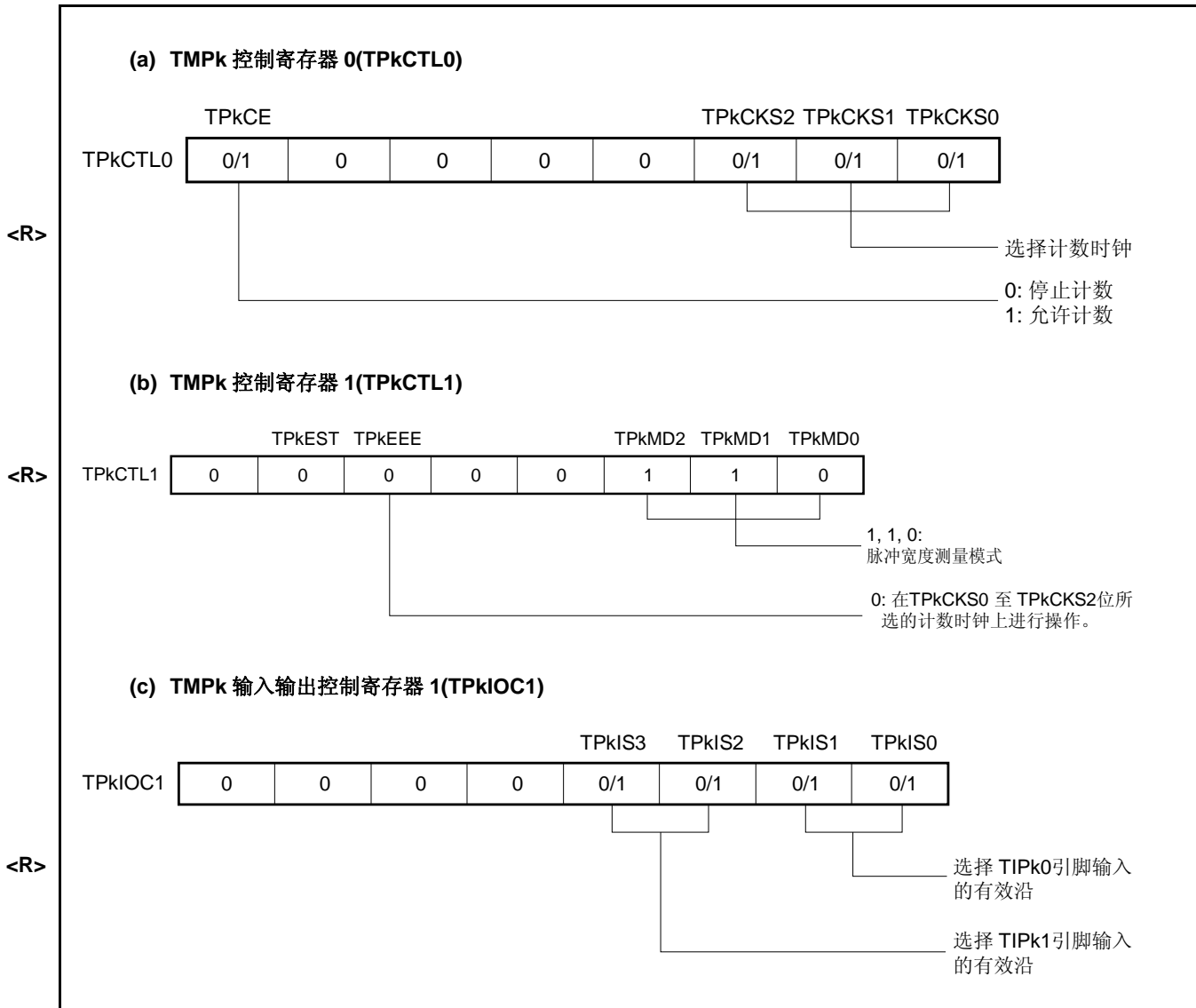
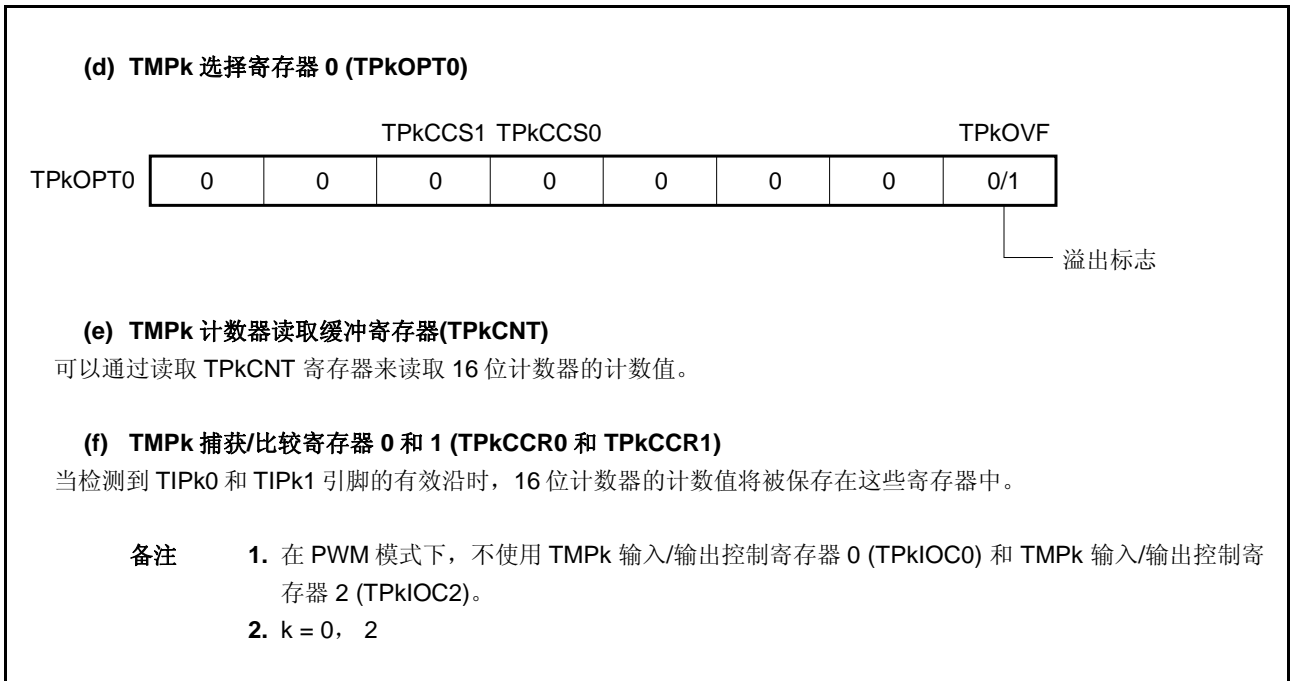


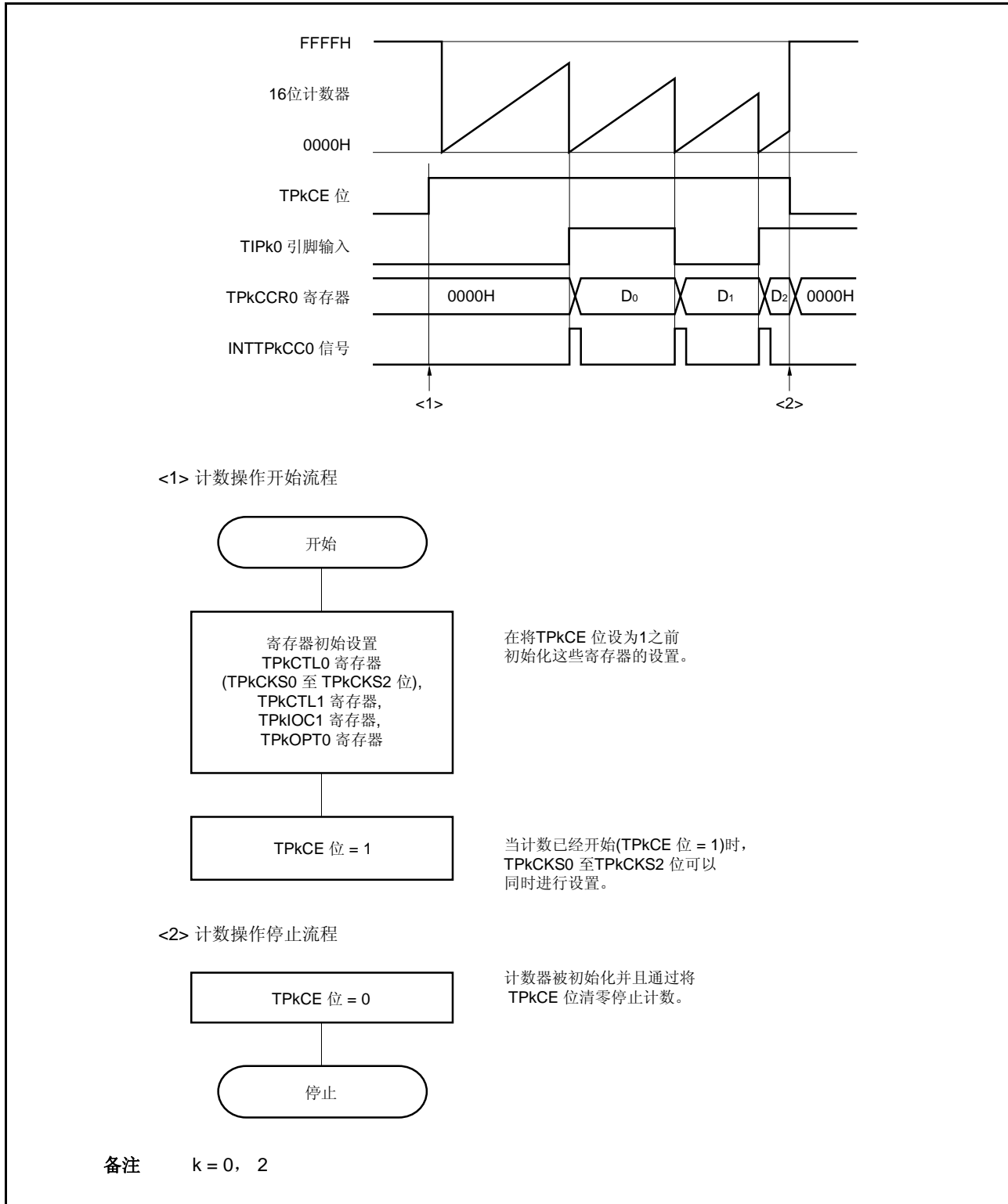
图 6-43. 脉冲宽度测量模式时的寄存器设置(2/2)



<R>

(1) 脉冲宽度测量模式时的工作流程

图 6-44. 脉冲宽度测量模式时的软件处理流程



<R>

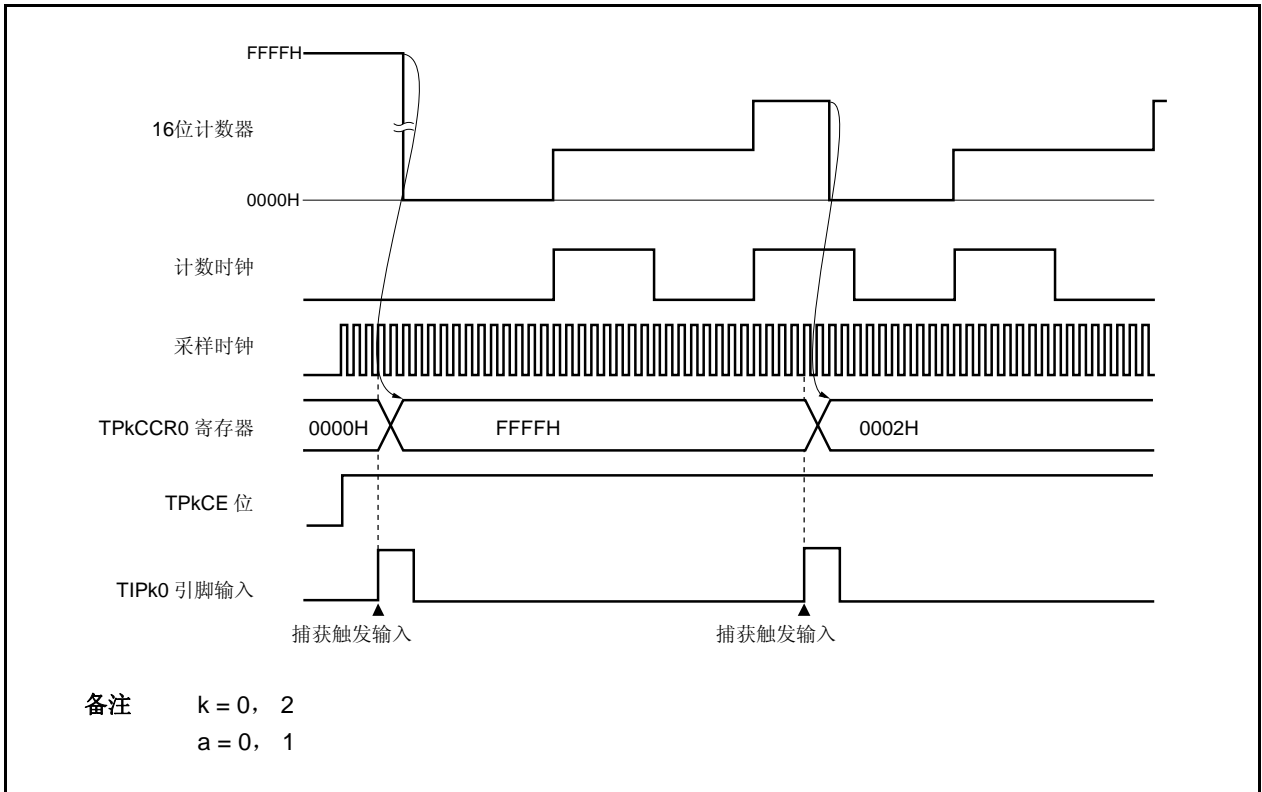
(2) 脉冲宽度测量模式时的工作时序

(a) 清除溢出标志

在 TPkOVF 为 1 时读取该位后，使用 CLR 指令将 TPkOVF 位清零可以将溢出标志清零，而在 TPkOVF 位为 1 时读取该位后，通过将 8 位数据（位 0 为 0）写入 TPkOPT0 寄存器也可以将溢出标志清零。

(3) 注

如果使用捕获操作并且将低速时钟选择为计数时钟时，如果 TPkCTL0.TPkCE 位置 1 之后捕获触发器被立即输入，TPkCCRa 寄存器可能捕获到 FFFFH，而不是 0000H。



第七章 16 位定时器/事件计数器 Q (TMQ)

定时器 Q (TMQ) 是一个 16 位定时器/事件计数器。V850ES/IE2 包括 TMQ1 和 TMQ0。

7.1 概要

以下介绍 TMQn 通道 (n = 0, 1)。

注意事项 如果 P16 被用作 TOQ00 输出引脚或输出端口, 那么当通过时钟监测器检测到错误 (振荡器停止) 时, CLMER 信号 (低电平) 将会从 P16 中输出。低电平输出通过复位信号来释放。详细信息, 参见表 4-5 端口 1 的复用功能引脚。

表 7-1. TMQn 的概述

名称	TMQ0	TMQ1
时钟选择	8 通道	8 通道
捕获触发输入引脚	4	无
外部事件计数输入引脚	1	无
外部触发输入引脚	无	无
定时器计数器	1	1
捕获/比较寄存器	4	4 ^{注 1}
捕获/比较匹配中断请求信号	4	4 ^{注 1}
溢出中断请求信号	1	1
定时器输出引脚 ^{注 2}	4	1

<R>

- 注**
1. 仅限比较功能
 2. 这是 TMQn 输出引脚的个数; 它不包括 TMQOP1 的输出引脚。关于 TMQOP1 的输出引脚的详细信息, 请参见第 9 章 马达控制功能。

7.2 功能

每个通道中可以实现的 TMQn 功能是不同的。各个通道的功能如下表所示 (n = 0, 1)。

表 7-2. TMQn 的功能

功能	TMQ0	TMQ1
6 相位 PWM 输出 ^{注 1}	×	√
间隔定时器	√	√
外部事件计数器	√	×
外部触发脉冲输出	√ ^{注 2}	×
单次脉冲输出	√ ^{注 2}	×
PWM 输出	√	×
自由运行定时器	√	√
脉冲宽度测量	√	×

- 注
1. 这个连接到 TMQOP1 上。详细信息，请参见第 9 章 马达控制功能。
 2. 只能通过软件触发来实现。不提供外部触发输入引脚。

7.3 配置

TMQn 包括以下硬件：

表 7-3. TMQn 的配置

名称	配置
定时器寄存器	16 位计数器 × 1
寄存器	TMQn 计数器读取缓冲寄存器 (TQnCNT)：总共 2 个 TMQn 捕获/比较寄存器 0 至 3 (TQnCCR0 至 TQnCCR3)：总共 8 个 CCR0 至 CCR3 缓冲寄存器：总共 8 个
定时器输入	总共 5 个 (TIQ00 至 TIQ03, EVTQ0 引脚)
定时器输出	总共 6 个 (TOQ00, TOQ01/TOQH01, TOQ02, TOQ03/TOQH02, TOQH03, TOQ10 引脚)
控制寄存器	TMQn 控制寄存器 0, 1 (TQnCTL0, TQnCTL1) TMQn 输入/输出控制寄存器 0 (TQnIOC0) TMQ0 输入/输出控制寄存器 1, 2 (TQ0IOC1, TQ0IOC2) TMQn 选项寄存器 0 (TQnOPT0)

备注 n = 0, 1

图 7-1. TMQ0 的框图

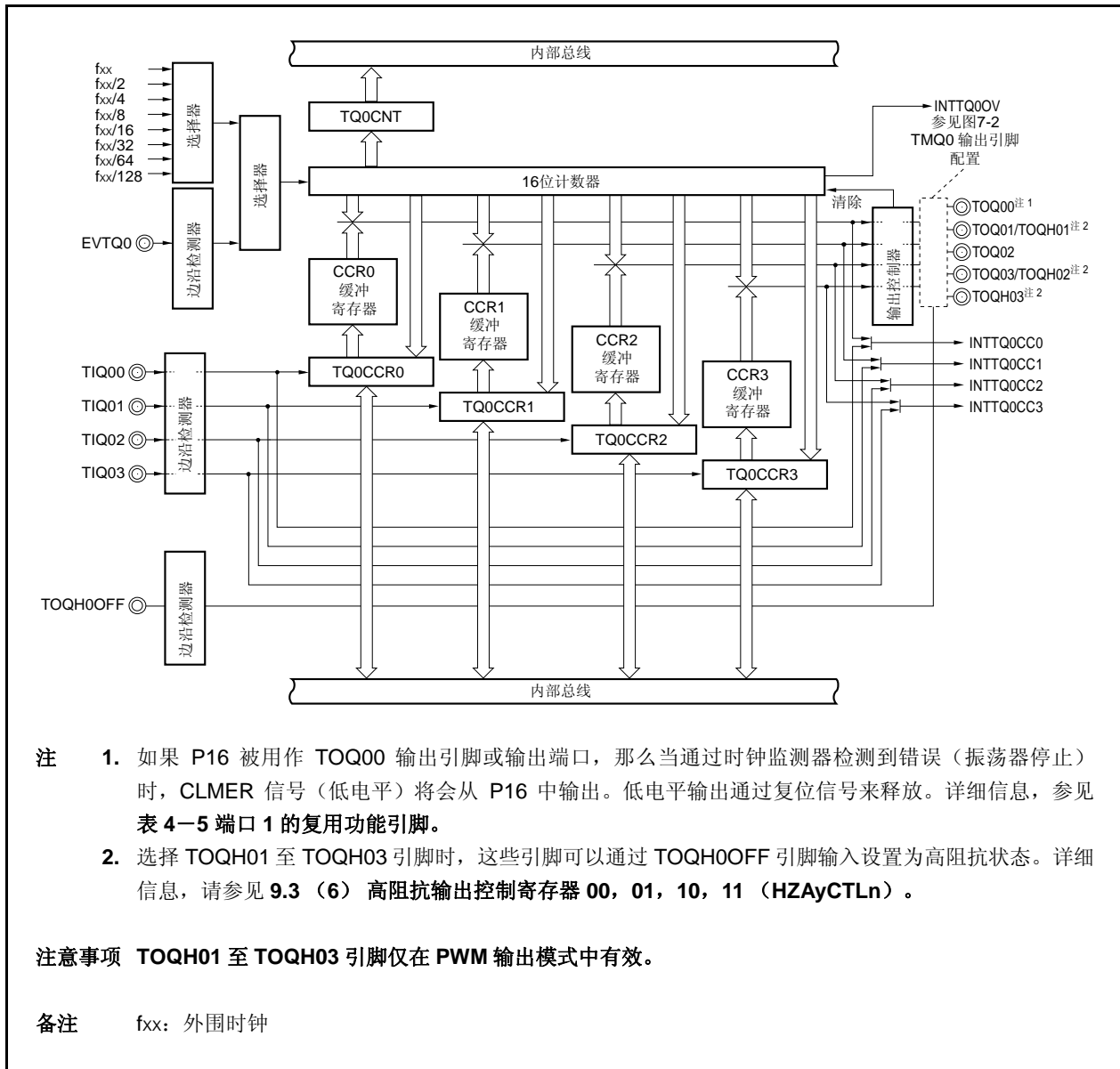


图 7-2. TMQ0 输出引脚的配置

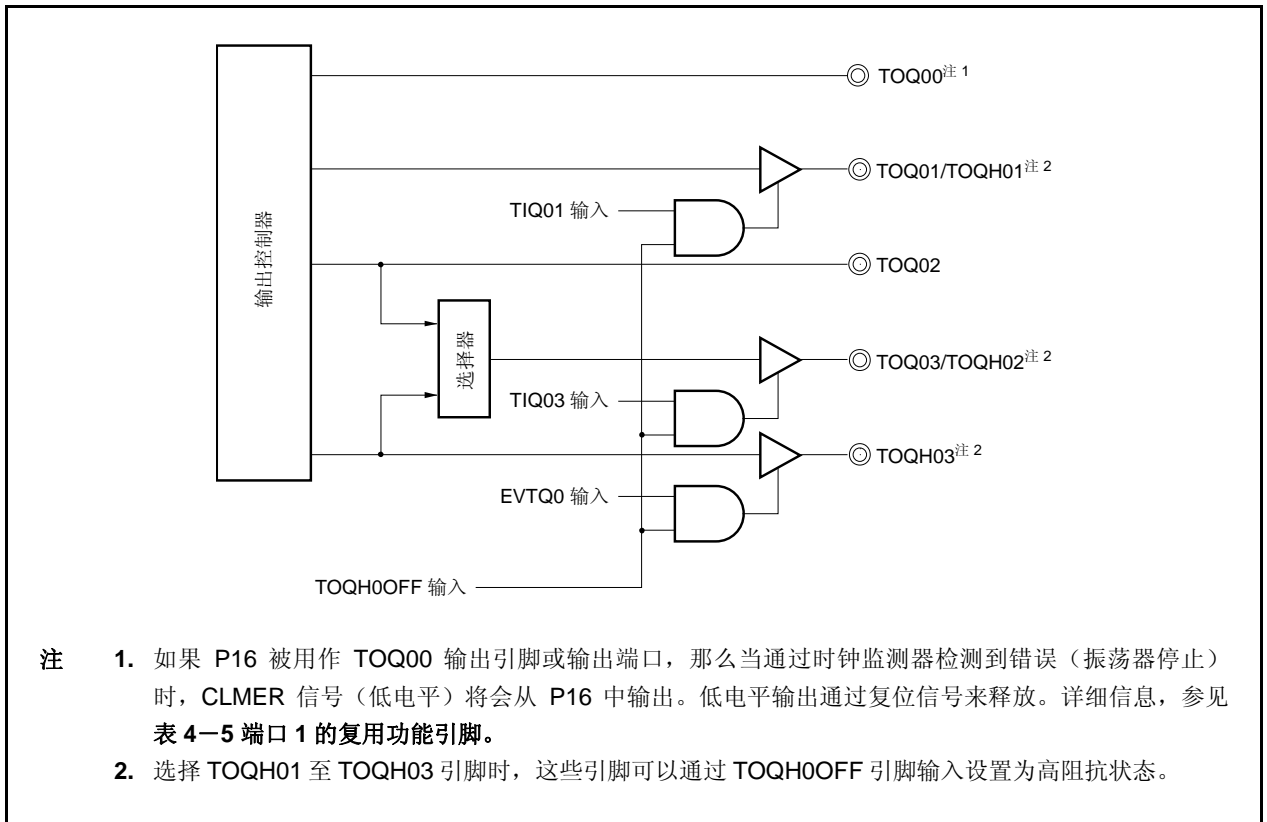
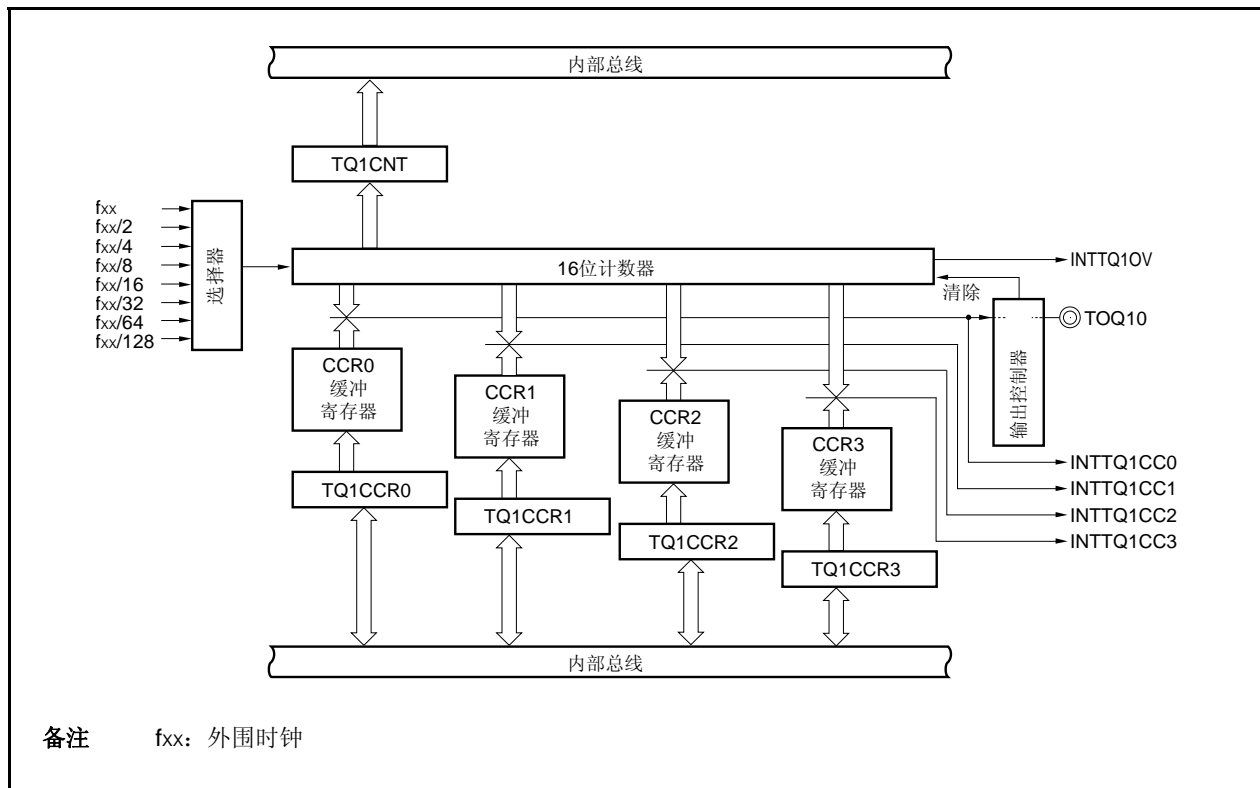


图7-3. TMQ1的框图



(1) 16位计数器

该16位计数器可以对内部时钟或外部事件计数。

可以使用TQnCNT寄存器来读取该计数器的计数值。

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

TQnCE位在复位后清除为0。

(2) CCR0缓冲寄存器

该寄存器为16位比较寄存器，用于比较16位计数器的计数值。

当TQnCCR0寄存器作为比较寄存器使用时，向TQnCCR0寄存器写入的数据将被传送至CCR0缓冲寄存器。

如果16位计数器的计数值与CCR0缓冲寄存器的值匹配，会产生一个比较匹配中断请求信号(INTTQnCC0)。

CCR0缓冲寄存器不能直接读写。

TQnCCR0寄存器在复位后清零为0000H，并且CCR0寄存器清零为0000H。

(3) CCR1 缓冲寄存器

此为 16 位比较寄存器，用于比较 16 位计数器的计数值。

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

CCR1 缓冲寄存器不能直接读写。

TQnCCR1 寄存器在复位后清零为 0000H，并且 CCR1 寄存器清零为 0000H。

(4) CCR2 缓冲寄存器

此为 16 位比较寄存器，用于比较 16 位计数器的计数值。

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

CCR2 缓冲寄存器不能直接读写。

TQnCCR2 寄存器在复位后清零为 0000H，并且 CCR2 寄存器清零为 0000H。

(5) CCR3 缓冲寄存器

此为 16 位比较寄存器，用于比较 16 位计数器的计数值。

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

CCR3 缓冲寄存器不能直接读写。

TQnCCR3 寄存器在复位后清零为 0000H，并且 CCR3 寄存器清零为 0000H。

(6) 边沿检测器

该电路检测输入到 TIQ00 至 TIQ03 和 EVTQ0 引脚信号的有效沿，使用 TQ0IOC1 和 TQ0IOC2 寄存器设置，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。

(7) 输出控制器

该电路用于控制 TOQ00, TOQ01/TOQH01, TOQ02, TOQ03/TOQH02, TOQH03, 以及 TOQ10 引脚的输出。TOQ00, TOQ01/TOQH01, TOQ02, TOQ03/TOQH02, 以及 TOQH03 引脚的输出由 TQ0IOC0 寄存器来控制。TOQ10 引脚的输出通过 TQ1IOC0 寄存器来控制。

(8) 选择器

此选择器用于选择 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为计数时钟。

7.4 寄存器

(1) TMQn 控制寄存器 0 (TQnCTL0)

TQnCTL0 寄存器为 8 位寄存器，用于控制定时器 TMQn 的操作。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

总是可以使用软件将相同的值写入 TQnCTL0 寄存器。

复位后： 00H R/W 地址： TQ0CTL0 FFFFF5C0H, TQ1CTL0 FFFFF600H

	<7>	6	5	4	3	2	1	0
TQnCTL0	TQnCE	0	0	0	0	TQnCKS2	TQnCKS1	TQnCKS0

(n = 0, 1)

TQnCE	TMQn 操作控制
0	TMQn 操作禁止 (TMQn 异步复位 ^注)
1	TMQn 操作允许。开始 TMQn 操作。

TQnCKS2	TQnCKS1	TQnCKS0	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 TQnOPT0.TQnOVF 位和 16 位计数器同步复位。此外，定时器输出 (TOQ00, TOQ01/TOQH01, TOQ02, TOQ03/TOQH02, TOQH03, 以及 TOQ10 引脚) 与 16 位计数器同时复位到 TQnIOC0 寄存器的设定状态。

注意事项 1. 当 TQnCE 位=0 时，设置 TQnCKS2 至 TQnCKS0 位。

当 TQnCE 位的值从 0 变为 1 时，可以同时设定 TQnCKS2 至 TQnCKS0 位。

2. 务必将位 3 至 6 清为“0”。

备注 f_{xx}: 外围时钟

(2) TMQn 控制寄存器 1 (TQnCTL1)

TQnCTL1 寄存器为 8 位寄存器，用于控制定时器 TMQn 的操作。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

复位后: 00H R/W 地址: TQ0CTL1 FFFF5C1H, TQ1CTL1 FFFF601H

	7	6	5	4	3	2	1	0
TQnCTL1 (n = 0, 1)	0	TQ0EST ^{注1}	TQ0EEE ^{注1}	0	0	TQnMD2	TQnMD1	TQnMD0

TQ0EST ^{注1}	软件触发控制
0	-
1	为外部触发输入产生一个有效信号。 · 在单脉冲输出模式中: 通过写1至 TQ0EST 位, 单脉冲输出为触发信号。 · 在外部触发脉冲输出模式中: 通过写1至TQ0EST位, PWM 波形输出为触发信号。
TQ0EST 位的读取值始终为0。	

TQ0EEE ^{注1}	计数时钟选择
0	禁止使用外部事件计数输入 (EVTQ0引脚)的操作。 (使用TQ0CTL0.TQ0CKS0 至 TQ0CTL0.TQ0CKS2位所选择的计数时钟执行计数。)
1	允许使用外部事件计数输入 (EVTQ0引脚)的操作。 (通过外部事件计数输入信号的有效沿执行计数。)
TQ0EEE 位用于选择计数是通过内部计数时钟还是外部事件计数输入的有效沿来执行。	

TQnMD2	TQnMD1	TQnMD0	定时器模式选择
0	0	0	间隔定时器模式
0	0	1	外部事件计数模式 ^{注2}
0	1	0	外部触发脉冲输出模式 ^{注2}
0	1	1	单脉冲输出模式 ^{注2}
1	0	0	PWM 输出模式 ^{注2}
1	0	1	自由运行定时器模式
1	1	0	脉冲宽度测量模式 ^{注2}
1	1	1	6相位 PWM 输出模式 ^{注3}

- 注
1. 这些位只能在 TMQ0 中进行设置。务必将 TMQ1 的位 5 和 6 清除为 0。
 2. 这些模式只能在 TMQ0 中进行设置。不能在 TMQ1 中对它们进行设置。
 3. 这些模式只能在 TMQ1 中进行设置。不能在 TMQ0 中对它们进行设置。当只使用 TMQ1 时, 不能使用该模式。详细信息, 请参见第 9 章 马达控制功能。

- 注意事项
1. TQ0EST 位只有在外部触发脉冲输出模式或单次脉冲输出模式时有效。在其它模式下, 忽略向该位写入 1 的操作。
 2. 不论 TQ0EEE 位的取值如何, 在外部事件计数模式下, 都选择外部事件输入端。
 3. 当 TQnCTL0.TQnCE = 0 时, 设置 TQnEEE 和 TQnMD2 至 TQnMD0 位。(当 TQnCE 位=1 时可写入相同的值。)当 TQnCE 位=1 时重写操作是无法保证的, 若重写发生错误, 应将 TQnCE 位清为 0, 再重新进行位设定。
 4. 一定要将位 3, 4 和 7 清为“0”。

(3) TMQn 输入/输出控制寄存器 0 (TQnIOC0)

TQnIOC0 寄存器为 8 位寄存器，用于控制定时器的输出 (TOQ00, TOQ01/TOQH01, TOQ02, TOQ03/TOQH02, TOQH03, 以及 TOQ1T1 至 TOQ1T3 引脚)。

该寄存器可进行 8 位或 1 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: TQ0IOC0 FFFFF5C2H, TQ1IOC0 FFFFF602H

	7	<6>	5	<4>	3	<2>	1	<0>
TQnIOC0	TQ0OL3 ^{注1}	TQ0OE3 ^{注1}	TQ0OL2 ^{注1}	TQ0OE2 ^{注1}	TQ0OL1 ^{注1}	TQ0OE1 ^{注1}	TQnOLO	TQnOEO
$\left. \begin{matrix} n = 0, 1 \\ b = 1 \text{ 至 } 3 \end{matrix} \right\}$	TQnOLm	TOQnm, TOQH0b, TOQ1Tb 引脚的输出电平设置 ^{注2} (TMQ0: m = 0 至 3, TMQ1: m = 0)						
	0	TOQnm, TOQH0b, 以及 TOQ1Tb 引脚从高电平开始输出。						
	1	TOQnm, TOQH0b, 以及 TOQ1Tb 引脚从低电平开始输出。						
TQnOEm	TOQnm, TOQH0b, TOQ1Tb 引脚的输出电平设置 (TMQ0: m = 0 至 3, TMQ1: m = 0)							
0	定时器输出禁止 · 当 TQnOLm 位 = 0 时, 低电平从 TOQnm, TOQH0b, 以及 TOQ1Tb 引脚输出。 · 当 TQnOLm 位 = 1 时, 高电平从 TOQnm, TOQH0b, 以及 TOQ1Tb 引脚输出。							
1	定时器输出允许 (从 TOQnm, TOQH0b, 以及 TOQ1Tb 引脚输出一个脉冲。)							

注 1. 将 TMQ1 用作间隔定时器或自由运行定时器时, 务必将 TQ1IOC0 寄存器的位 2 至 7 清除为 0。

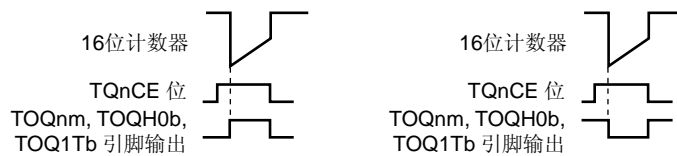
此外, 将 TMQ1 的 TOQ1T1 至 TOQ1T3 和 TOQ1B1 至 TOQ1B3 引脚用作 6 相位 PWM 输出时, 应将 TQ1IOC0 寄存器的位 2, 4 和 6 设置为 1, 并将位 3, 5 和 7 设置为 0 或 1。

2. 由 TQnOLm 位指定的定时器输出引脚 (TOQnm, TOQH0b 和 TOQ1Tb) 的输出电平如下所示。

· 当 TQnOLm 位 = 0 时

· 当 TQnOLm 位 = 1 时

<R>



<R>

注意事项 1. 当 TOQnm, TOQH0b 和 TOQ1Tb 设置为输出模式时, 如果 TQnIOC0 寄存器的设置改变, 则引脚的输出改变。在输入模式和使端口进入高阻状态时设置端口, 则引脚的状态不会改变。

- 注意事项
2. 当 **TQnCTL0.TQnCE** 位 = 0 时, 可对 **TQnOLm** 和 **TQnOEm** 位进行重写。(**TQnCE** 位 = 1 时可以向这些位写入相同的值)。如果由于误操作而引起改写, 则要将 **TQnCE** 位清零然后再次设置这些位。
 3. 在 **TQnCE** 和 **TQnOE** 位均为 0 时, 如果对 **TQnOLm** 位进行了操作, 那么 **TOQnm**, **TOQH0b** 和 **TOQ1Tb** 引脚的输出电平会改变。
 4. 要在 6 相位 PWM 输出模式中生成 **TOQ1Tb** 引脚输出以及 A/D 转换器 0 和 1 的 A/D 转换开始触发信号, 务必使用 **TQ1IOC0** 寄存器来设置 **TOQ1Tb** 引脚输出模式。此时, 务必将 **TQ1OL0** 位清零, 并将 **TQ1OE0** 位设为 1 (**b = 1 至 3**)。

(4) TMQ0 输入/输出控制寄存器 1 (TQ0IOC1)

TQ0IOC1 寄存器为 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。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

备注 TMQ1 不包含 TQ1IOC1 寄存器。

复位后: 00H R/W 地址: FFFFF5C3H

	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

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，可对 TQ0IS7 至 TQ0IS0 位进行重写。(TQ0CE 位 = 1 时可以向这些位写入相同的值)。如果由于误操作而引起改写，则要将 TQ0CE 位清零然后再次设置这些位。
 2. TQ0IS7 至 TQ0IS0 位只有在自由运行定时器模式 (仅当 TQ0OPT0.TQ0CCS3 至 TQ0OPT0.TQ0CCS0 位 = 1111 时) 和脉冲宽度测量模式下才有效。在其他模式下，不能进行捕获操作。

(5) TMQ0 输入/输出控制寄存器 2 (TQ0IOC2)

TQ0IOC2 寄存器为 8 位寄存器，用于控制外部事件计数输入信号 (EVTQ0 引脚) 的有效沿。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

备注 TMQ1 不包含 TQ1IOC2 寄存器。

复位后: 00H R/W 地址: FFFFF5C4H

	7	6	5	4	3	2	1	0
TQ0IOC2	0	0	0	0	TQ0EES1	TQ0EES0	0	0

TQ0EES1	TQ0EES0	外部事件计数输入信号 (EVTQ0 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

- 注意事项**
1. 当 TQ0CTL0.TQ0CE 位 = 0 时，可对 TQ0EES1 和 TQ0EES0 位进行重写。(TQ0CE 位 = 1 时可以向这些位写入相同的值)。如果由于误操作而引起重写，则要将 TQ0CE 位清零然后再次设置这些位。
 2. TQ0EES1 和 TQ0EES0 位只有在 TQ0CTL1.TQ0EEE 位 = 1 时或在外部事件计数模式下 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位=001) 才有效。
 3. 一定要将位 0, 1 和 4 至 7 清为“0”。

(6) TMQn 选择寄存器 0 (TQnOPT0)

TQnOPT0 寄存器是一个 8 位寄存器，用来设置捕获/比较操作和检测计数器溢出。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

复位后: 00H R/W 地址: TQ0OPT0 FFFFF5C5H, TQ1OPT0 FFFFF605H

	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
TQnOPT0 (n = 0, 1)	TQ0CCS3 ^{注1}	TQ0CCS2 ^{注1}	TQ0CCS1 ^{注1}	TQ0CCS0 ^{注1}	0	TQ1CMS ^{注2}	TQ1CUF ^{注2}	TQnOVF

TQ0CCSm	TQ0CCRm 寄存器捕获/比较选择 (m = 0 至 3)
0	比较寄存器选择
1	捕获寄存器选择(通过TQ0CTL0.TQ0CE 位 = 0来清除)
TQ0CCSm 位的设置仅在自由运行定时器模式中有效。	

TQnOVF	TMQn 溢出标志
置 (1)	发生溢出。
复位 (0)	0 写入 TQnOVF 位或 TQnCTL0.TQnCE 位 = 0
<ul style="list-style-type: none"> · 在自由运行定时器模式或脉冲宽度测量模式中，当16位计数器的值从FFFFH溢出到0000H时， TQnOVF 位被置为1。 · TQnOVF 位被置为1的同时会产生一个溢出中断请求信号 (INTTQnOV)。INTTQnOV 信号不会在自由运行定时器模式和脉冲宽度测量模式外的其他模式中生成。 · 即使在 TQnOVF 位 = 1时，读取TQnOVF 位 TQnOPT0 寄存器，TQnOVF 位也不会被清零。 · 在生成INTTQnOV信号后清除 TQnOVF为0，在此之前一定要确保（通过读取）TQnOVF位被置 1。 · TQnOVF 位可以读取或写入，单 TQnOVF 位不能通过软件置为 (1)。写入1 对 TMQn 的操作不会产生任何影响。 	

- 注**
1. 仅在 TMQ0 中有效。务必将 TMQ1 中的位 7 至 4 清为 0。
 2. 仅在 TMQ1 中有效。务必将 TMQ0 的位 2 和 1 清除为 0。关于 TQ1CMS 和 TQ1CUF 位的详细信息，请参见第 9 章 马达控制功能。

- 注意事项**
1. 当 TQ0CE 位 = 0 时，可对 TQ0CCS3 到 TQ0CCS0 位进行重写（当 TQ0CE 位 = 1 时可以向这些位写入相同的值）。如果由于误操作而引起重写，则要将 TQ0CE 位清零然后再次设置这些位。
 2. 务必将位 3 清为“0”。

(7) TMQn 捕获/比较寄存器 0 (TQnCCR0)

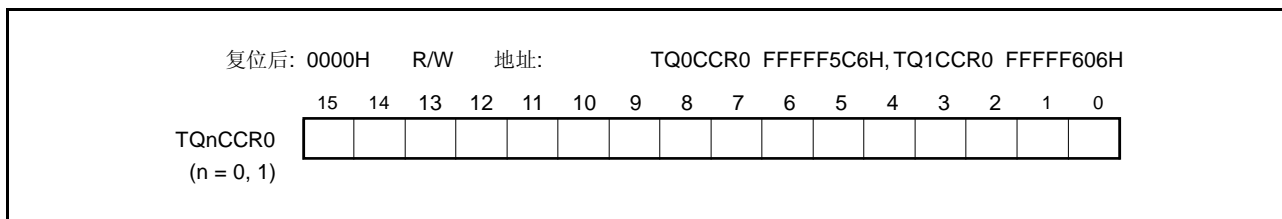
TQ0CCR0 寄存器是 16 位寄存器，根据不同的工作模式，可用作捕获寄存器或比较寄存器。TQ1CCR0 寄存器只能用作 16 位比较寄存器。

只有在自由运行模式下，根据 TQ0OPT0.TQ0CCS0 位设置的不同，TQ0CCR0 寄存器可以用作捕获寄存器或比较寄存器。在脉宽测量模式下，TQ0CCR0 寄存器只能用作捕获寄存器。在其它模式下，该寄存器只能用作比较寄存器。

TQnCCR0 寄存器可以在操作过程中进行读取和写入。

该寄存器可以 16 位为单位进行读取和写入。

复位设置将寄存器清为 0000H。



(a) 作为比较寄存器时的功能

即便在 TQnCTL0.TQnCE 位=1 时, TQnCCR0 仍可以重新写入。

TQnCCR0 寄存器的设置值将被传送到 CCR0 缓冲寄存器中。当 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相匹配时, 会产生一个比较匹配中断请求信号 (INTTQnCC0)。如果此时允许 TOQn0 引脚输出, 则 TOQn0 引脚的输出将反向。

在间隔定时器模式、外部事件计数器模式^{注1}、外部触发脉冲输出模式^{注2}、单次脉冲输出模式^{注2}或脉宽调制信号输出模式^{注2}时, 当 TQnCCR0 寄存器作为循环寄存器时, 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)。

TQnCTL0.TQnCE 位=0 时没有清除比较寄存器。

注 这些模式只能在 TMQ0 中进行设置。它们不能在 TMQ1 中进行设置。

(b) 作为捕获寄存器时的功能 (仅限 TQ0CCR0 寄存器)

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) 的有效边缘, 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) 的有效边缘, 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)。

即使捕获操作与对 TQ0CCR0 寄存器的读取发生冲突, 还是可以正确读取 TQ0CCR0 寄存器的值。

在 TQ0CTL0.TQ0CE 位=0 时, 捕获寄存器不能清除。

下表显示捕获/比较寄存器在各工作模式下的功能, 以及怎样向比较寄存器写入数据。

表 7-4. 捕获/比较寄存器在各模式下的功能和如何写入比较寄存器

工作模式	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 ^{注1}	比较寄存器	任何时候均可写入
外部触发脉冲输出 ^{注1, 2}	比较寄存器	批写入 ^{注3}
单脉冲输出 ^{注1, 2}	比较寄存器	任何时候均可写入
PWM 输出 ^{注1}	比较寄存器	批写入 ^{注3}
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 ^{注1}	捕获寄存器	无

- 注**
1. 仅限 TMQ0。
 2. 该模式只能通过软件触发来设置。不提供外部触发输入引脚。
 3. 写入 TQ0CCR1 寄存器的是触发脉冲。

备注 关于随时写入和批写入的详细信息, 请参见 7.6 (2) 随时写入和批写入。

(8) TMQn 捕获/比较寄存器 1 (TQnCCR1)

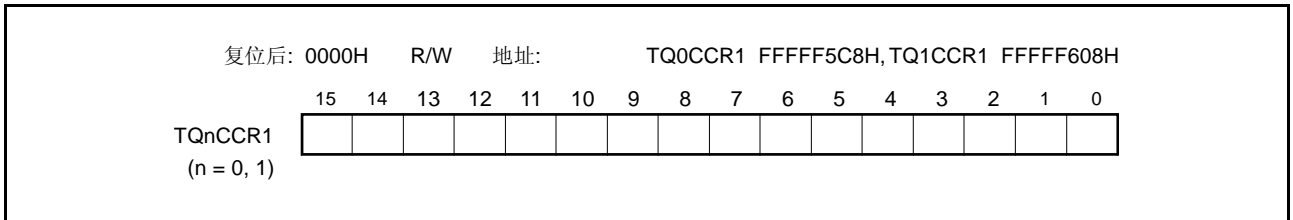
TQ0CCR1 寄存器是 16 位寄存器，根据不同的工作模式，可用作捕获寄存器或比较寄存器。TQ1CCR1 寄存器只能用作比较寄存器的 16 位寄存器。

只有在自由运行模式下，根据 TQ0OPT0.TQ0CCS1 位设置的不同，TQ0CCR1 寄存器可以用作捕获寄存器或比较寄存器。在脉宽测量模式下，TQ0CCR1 寄存器只能用作捕获寄存器。在其它模式下，该寄存器只能用作比较寄存器。

TQnCCR1 寄存器可以在操作过程中进行读取和写入。

该寄存器可以 16 位为单位进行读取和写入。

复位设置将寄存器清为 0000H。



(a) 作为比较寄存器时的功能

即便在 TQnCTL0.TQnCE 位=1 时，TQnCCR1 仍可以重新写入。

TQnCCR1 寄存器的设置值将被传送到 CCR1 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 缓冲寄存器中的数值相匹配时，会产生一个比较匹配中断请求信号 (INTTQnCC1)。如果此时允许 TOQ01/TOQH01 引脚输出，则 TOQ01/TOQH01 引脚的输出将反向（不提供 TOQ11 和 TOQH11 引脚）。

TQnCTL0.TQnCE 位=0 时没有清除比较寄存器。

(b) 作为捕获寄存器时的功能（仅限 TQ0CCR1 寄存器）

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）的有效边缘，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）的有效边缘，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）。

即使捕获操作与对 TQ0CCR1 寄存器的读取发生冲突，还是可以正确读取 TQ0CCR1 寄存器的值。

在 TQ0CTL0.TQ0CE 位=0 时，捕获寄存器不能清除。

下表显示捕获/比较寄存器在各工作模式下的功能，以及怎样向比较寄存器写入数据。

表 7-5. 捕获/比较寄存器在各模式下的功能和如何写入比较寄存器

工作模式	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 ^{注1}	比较寄存器	任何时候均可写入
外部触发脉冲输出 ^{注1, 2}	比较寄存器	批写入 ^{注3}
单脉冲输出 ^{注1, 2}	比较寄存器	任何时候均可写入
PWM 输出 ^{注1}	比较寄存器	批写入 ^{注3}
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 ^{注1}	捕获寄存器	无

注 1. 仅限 TMQ0。

2. 该模式只能通过软件触发来设置。不提供外部触发输入引脚。

3. 写入 TQ0CCR1 寄存器的是触发脉冲。

备注 关于随时写入和批写入的详细信息，请参见 **7.6 (2) 随时写入和批写入**。

(9) TMQn 捕获/比较寄存器 2 (TQnCCR2)

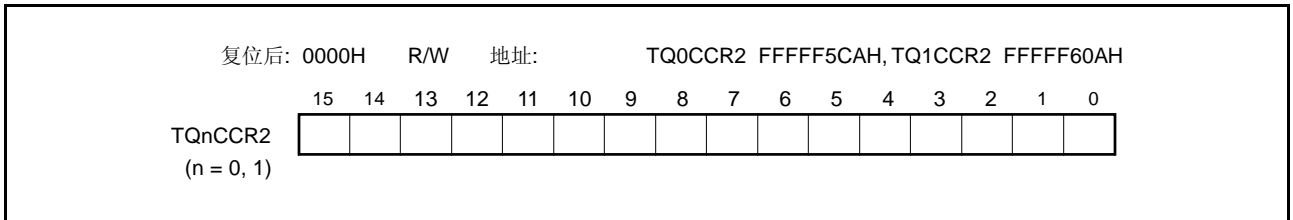
TQ0CCR2 寄存器是 16 位寄存器，根据不同的工作模式，可用作捕获寄存器或比较寄存器。TQ1CCR2 寄存器只能用作比较寄存器的 16 位寄存器。

只有在自由运行模式下，根据 TQ0OPT0.TQ0CCS2 位设置的不同，TQ0CCR2 寄存器可以用作捕获寄存器或比较寄存器。在脉宽测量模式下，TQ0CCR2 寄存器只能用作捕获寄存器。在其它模式下，该寄存器只能用作比较寄存器。

TQnCCR2 寄存器可以在操作过程中进行读取和写入。

该寄存器可以 16 位为单位进行读取和写入。

复位设置将寄存器清为 0000H。



(a) 作为比较寄存器时的功能

即便在 TQnCTL0.TQnCE 位=1 时，TQnCCR2 仍可以重新写入。

TQnCCR2 寄存器的设置值将被传送到 CCR2 缓冲寄存器中。当 16 位计数器的计数值与 CCR2 缓冲寄存器中的数值相匹配时，会产生一个比较匹配中断请求信号 (INTTQnCC2)。如果此时允许 TOQ02/TOQH02 引脚输出，则 TOQ02/TOQH02 引脚的输出将反向（不提供 TOQ12 和 TOQH12 引脚）。

TQnCTL0.TQnCE 位=0 时没有清除比较寄存器。

(b) 作为捕获寄存器时的功能（仅限 TQ0CCR2 寄存器）

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）的有效边缘，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）的有效边缘，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）。

即使捕获操作与对 TQ0CCR2 寄存器的读取发生冲突，还是可以正确读取 TQ0CCR2 寄存器的值。

在 TQ0CTL0.TQ0CE 位=0 时，捕获寄存器不能清除。

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

表 7-6. 捕获/比较寄存器在各模式下的功能和如何写入比较寄存器

工作模式	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 ^{注1}	比较寄存器	任何时候均可写入
外部触发脉冲输出 ^{注1, 2}	比较寄存器	批写入 ^{注3}
单脉冲输出 ^{注1, 2}	比较寄存器	任何时候均可写入
PWM 输出 ^{注1}	比较寄存器	批写入 ^{注3}
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 ^{注1}	捕获寄存器	无

注 1. 仅限 TMQ0。

2. 该模式只能通过软件触发来设置。不提供外部触发输入引脚。

3. 写入 TQ0CCR1 寄存器的是触发脉冲。

备注 关于随时写入和批写入的详细信息，请参见 **7.6 (2) 随时写入和批写入**。

(10) TMQn 捕获/比较寄存器 3 (TQnCCR3)

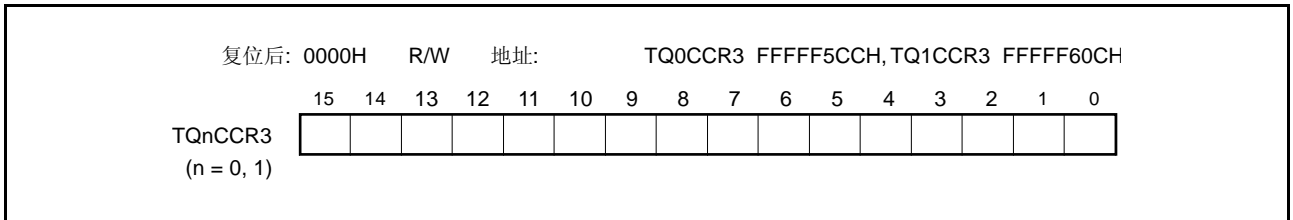
TQ0CCR3 寄存器是 16 位寄存器，根据不同的工作模式，可用作捕获寄存器或比较寄存器。TQ1CCR3 寄存器只能用作比较寄存器的 16 位寄存器。

只有在自由运行模式下，根据 TQ0OPT0.TQ0CCS3 位设置的不同，TQ0CCR3 寄存器可以用作捕获寄存器或比较寄存器。在脉宽测量模式下，TQ0CCR3 寄存器只能用作捕获寄存器。在其它模式下，该寄存器只能用作比较寄存器。

TQnCCR3 寄存器可以在操作过程中进行读取和写入。

该寄存器可以 16 位为单位进行读取和写入。

复位设置将寄存器清为 0000H。



(a) 作为比较寄存器时的功能

即便在 TQnCTL0.TQnCE 位=1 时，TQnCCR3 仍可以重新写入。

TQnCCR3 寄存器的设置值将被传送到 CCR3 缓冲寄存器中。当 16 位计数器的计数值与 CCR3 缓冲寄存器中的数值相匹配时，会产生一个比较匹配中断请求信号 (INTTQnCC3)。如果此时允许 TOQ03/TOQH03 引脚输出，则 TOQ03/TOQH03 引脚的输出将反向（不提供 TOQ13 和 TOQH13 引脚）。

TQnCTL0.TQnCE 位=0 时没有清除比较寄存器。

(b) 作为捕获寄存器时的功能（仅限 TQ0CCR3 寄存器）

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) 的有效边缘，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) 的有效边缘，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)。

即使捕获操作与对 TQ0CCR3 寄存器的读取发生冲突，还是可以正确读取 TQ0CCR3 寄存器的值。

在 TQ0CTL0.TQ0CE 位=0 时，捕获寄存器不能清除。

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

表 7-7. 捕获/比较寄存器在各模式下的功能和如何写入比较寄存器

工作模式	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 ^{注1}	比较寄存器	任何时候均可写入
外部触发脉冲输出 ^{注1, 2}	比较寄存器	批写入 ^{注3}
单脉冲输出 ^{注1, 2}	比较寄存器	任何时候均可写入
PWM 输出 ^{注1}	比较寄存器	批写入 ^{注3}
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 ^{注1}	捕获寄存器	无

注 1. 仅限 TMQ0。

2. 该模式只能通过软件触发来设置。不提供外部触发输入引脚。

3. 写入 TQ0CCR1 寄存器的是触发脉冲。

备注 关于随时写入和批写入的详细信息，请参见 7.6 (2) 随时写入和批写入。

(11) TMQn 计数器读取缓冲寄存器 (TQnCNT)

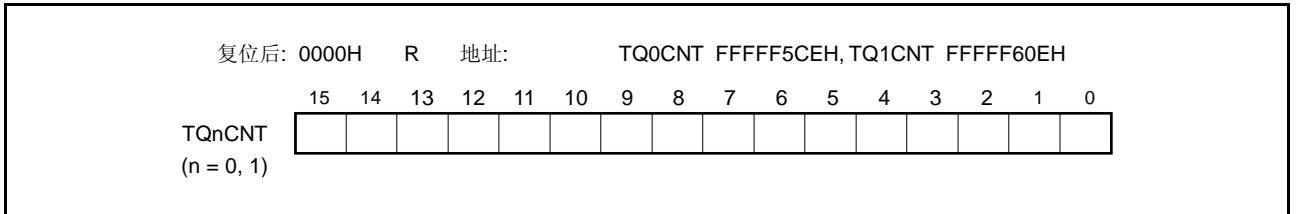
可以通过读取缓冲寄存器 TQnCNT 来读取 16 位计数器的计数值。

如果该寄存器在 TQnCTL0.TQnCE = 1 时被读取，那么读取的值就是 16 位计数器的计数值。

该寄存器为只读寄存器，按 16 位读取。

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

复位后，TQnCE 位被清零的同时 TQnCNT 寄存器的值被清除为 0000H。



7.5 定时器输出操作

TOQ00 至 TOQ03, TOQ10, 以及 TOQH01 至 TOQH03 引脚的工作模式与输出电平的关系如下表所示。

表 7-8. 各种模式下的定时器输出控制

工作模式	TOQn0 引脚	TOQ01 至 TOQ03 引脚	TOQH01 至 TOQH03 引脚
间隔计时器模式	PWM 输出		无
外部事件计数模式	无		
外部触发脉冲输出模式	PWM 输出 ^注	外部触发脉冲输出	无
单次脉冲输出模式		单次脉冲输出	
PWM 输出模式		PWM 输出	
自由定时器模式	PWM 输出 (仅限使用比较功能时)		无
脉冲宽度测量模式	无		

注 仅限 TOQ00 引脚。

备注 n = 0, 1

表 7-9. TOQ00 至 TOQ03, TOQ10, 以及 TOQH01 至 TOQH03 引脚受控于定时器输出控制位的真值表

TQnIOC0.TQnOLa 位	TQnIOC0.TQnOEa 位	TQnCTL0.TQnCE 位	TOQna 和 TOQH1b 位的引脚电平
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 当 n = 0 时, a = 0 至 3。

当 n = 1 时, a = 0。

b = 1 至 3

7.6 操作

在 TMQ0 和 TMQ1 间可以实现的功能是不同的。各个通道的功能如下所示。

表 7-10. 各个模式中 TMQ0 的规范

操作	TQ0CTL1.TQ0EST 位 (软件触发位)	捕获/比较寄存器设置	比较寄存器写入方法
间隔计时器模式	无效	仅比较寄存器有效	任何时候均可写入
外部事件计数模式	无效	仅比较寄存器有效	任何时候均可写入
外部触发脉冲输出模式	有效	仅比较寄存器有效	批写入
单次脉冲输出模式	有效	仅比较寄存器有效	任何时候均可写入
PWM 输出模式	无效	仅比较寄存器有效	批写入
自由定时器模式	无效	可变换	任何时候均可写入
脉冲宽度测量模式	无效	仅捕获寄存器有效	无效

表 7-11. 各个模式中 TMQ1 的规范

操作	软件触发位	捕获/比较寄存器设置	比较寄存器写入方法
间隔计时器模式	无效	仅比较寄存器有效	任何时候均可写入
外部事件计数模式	无		
外部触发脉冲输出模式	无		
单次脉冲输出模式	无		
PWM 输出模式	无		
自由定时器模式	无效	仅比较寄存器有效	任何时候均可写入
脉冲宽度测量模式	无		

备注 TMQ1 具有通过 TMP1 执行调谐的功能。详细信息，请参见第 9 章 马达控制功能。

(1) 计数器基本操作

本节介绍了 16 位计数器的基本操作。详细信息，请参照各个模式中对操作的具体描述。

备注 n = 0, 1
 a = 0 至 3

<R>

(a) 计数起始操作

- 外部事件计数模式

当 TQ0CTL0.TQ0CE 位从 0 置为 1 时，16 位计数器置为 0000H。

然后，开始向上计数到 0001H，0002H，0003H，...检测每一次外部事件计数输入 (EVTQ0) 的有效边沿。

- 在非上述模式中

所有模式中，TMQn 的 16 位计数器都从默认值 FFFFH 开始计数。

它从 FFFFH 开始累计至 0000H，0001H，0002H，0003H 等等。

(b) 清除操作

16 位计数器在其值与比较寄存器的值匹配并且被捕获时被清除为 0000H。在计数器开始计数后，或当计数器溢出时所发生的从 FFFFH 至 0000H 的计数操作不是清除操作。因此，不会生成 INTTQnCCa 中断信号。

(c) 溢出操作

在自由运行定时器模式或脉宽测量模式中，计数器从 FFFFH 减至 0000H 时，16 位计数器将会发生溢出。

如果计数器溢出，TQnOPT0.TQnOVF 位将会被设为 1 并生成一个中断请求信号 (INTTQnOV)。注意 INTTQnOV 信号不会在以下条件中产生。

- 计数操作开始后。
- 如果计数器的值与比较值 FFFFH 匹配并被清除。
- 当 FFFFH 在脉冲宽度测量模式中被捕获，且计数器从 FFFFH 累计至 0000H 时。

注意事项 溢出中断请求信号 (INTTQnOV) 生成后，务必检测溢出标志 (TQnOVF 位) 是否被置为 1。

(d) 计数操作期间计数器读取操作

在计数操作期间，TMQn 的 16 位计数器的值可以通过使用 TQnCNT 寄存器来读取。TQnCTL0.TQnCE 位 = 1 时，可以通过读取 TQ0CNT 寄存器来读取 16 位计数器的值。然而，在 TQnCE 位 = 0 时，16 位计数器的值为 FFFFH，而 TQnCNT 寄存器的值则为 0000H。

(e) 中断操作

TMQn 生成以下五种中断请求信号。

- INTTQnCC0 中断: 该信号的功能相当于 CCR0 缓冲寄存器的匹配中断请求信号以及 TQnCCR0 寄存器的捕获中断请求信号。
- INTTQnCC1 中断: 该信号的功能相当于 CCR1 缓冲寄存器的匹配中断请求信号以及 TQnCCR1 寄存器的捕获中断请求信号。
- INTTQnCC2 中断: 该信号的功能相当于 CCR2 缓冲寄存器的匹配中断请求信号以及 TQnCCR2 寄存器的捕获中断请求信号。
- INTTQnCC3 中断: 该信号的功能相当于 CCR3 缓冲寄存器的匹配中断请求信号以及 TQnCCR3 寄存器的捕获中断请求信号。
- INTTQnOV 中断: 该信号的功能相当于一个溢出中断请求信号。

(2) 随时写入和批写入

在定时器操作 (TQnCTL0.TQnCE 位 = 1) 期间, TMQn 中的 TQnCCR0 至 TQnCCR3 寄存器可以被重写, 但 CCR0 至 CCR3 缓冲寄存器的写入方法 (随时写入, 批写入) 会根据模式的不同而变化。

(a) 随时写入

在该模式中, 在定时器操作过程中 ($n = 0, 1$) 数据可以随时从 TQnCCR0 至 TQnCCR3 寄存器转移到 CCR0 至 CCR3 缓冲寄存器中。

图 7-4. 随时写入的基本操作流程图

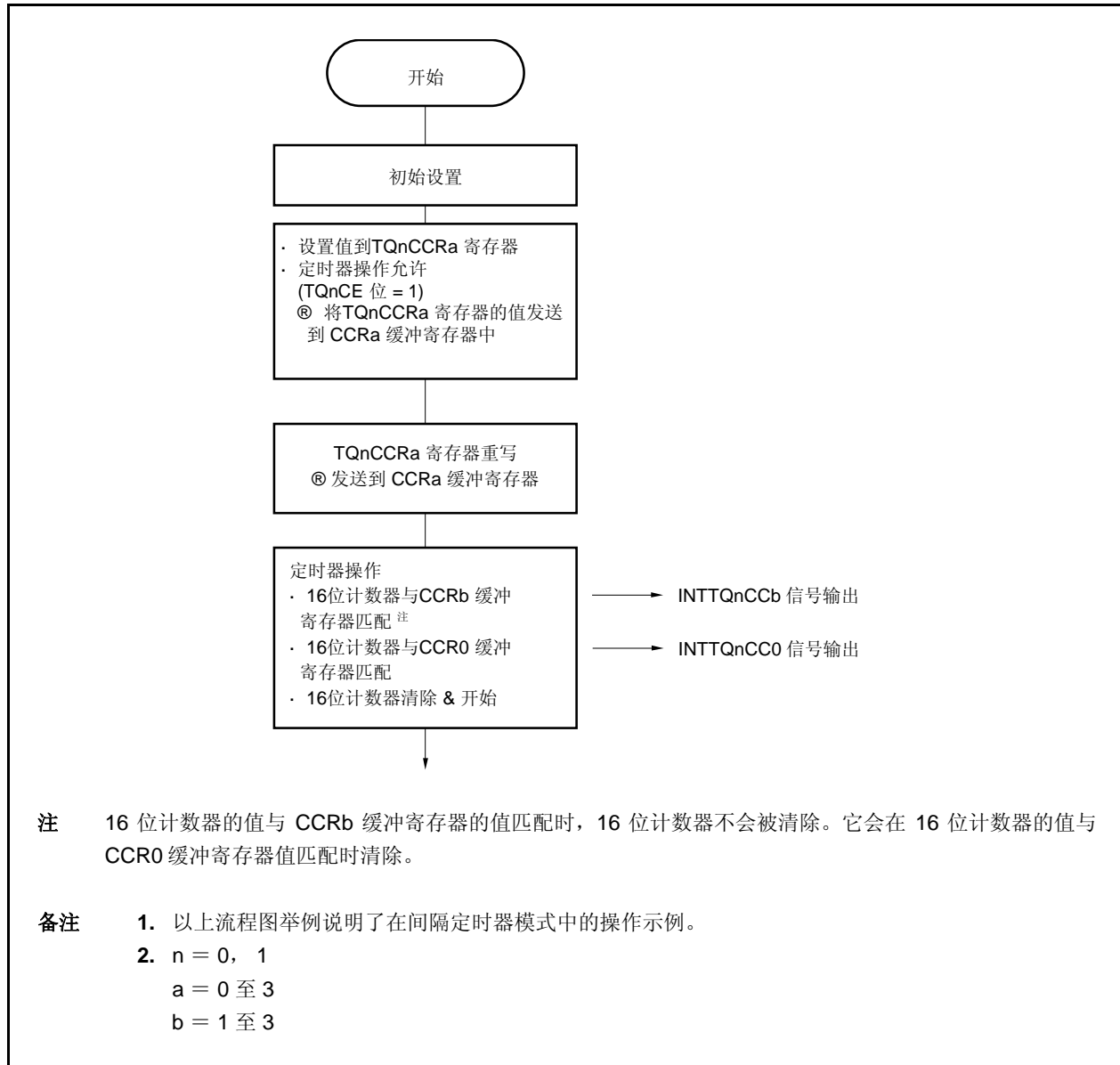
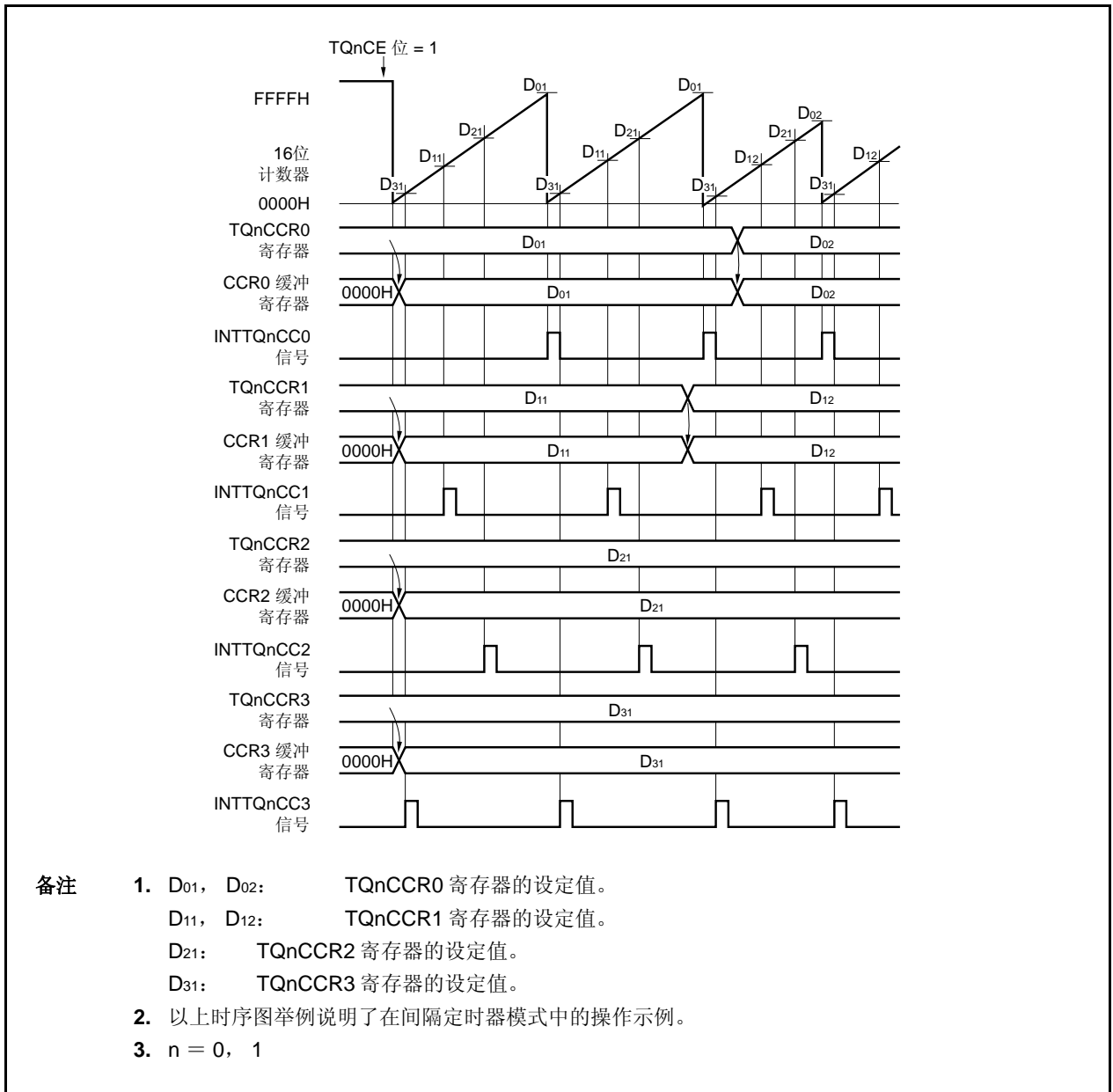


图 7-5. 随时写入的时序



(b) 批写入

在该模式中，在定时器操作过程中所有数据会立即从 TQ0CCR0 至 TQ0CCR3 寄存器转移到 CCR0 至 CCR3 缓冲寄存器中。该数据会在 16 位计数器的值与 CCR0 缓冲寄存器值匹配时转移。通过对 TQ0CCR1 寄存器执行写入操作可以允许转移。

允许还是禁止下个转移时序通过对 TQ0CCR1 寄存器执行或不执行写入操作来控制。

为了在重写 TQ0CCR0 至 TQ0CCR3 寄存器时将设定值转变为 16 位计数器的比较值（换句话说，为了让该值转移到 CCR0 至 CCR3 缓冲寄存器中），在 16 位计数器的值与 CCR0 寄存器的值匹配前，需要先对 TQ0CCR0 寄存器进行重写，最后再对 TQ0CCR1 寄存器进行写入操作。在 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配时，TQ0CCR0 至 TQ0CCR3 寄存器的值被传送到 CCR0 至 CCR3 缓冲寄存器中。因此，即使你只想重写 TQ0CCR0，TQ0CCR2，或 TQ0CCR3 寄存器的值，相同的值（与 TQ0CCR1 寄存器的预设值相同）也会被写入到 TQ0CCR1 寄存器中。

备注 TMQ1 不能在它可以通过批写入进行重写的模式中进行设置。

图 7-6. 批写入的基本操作流程

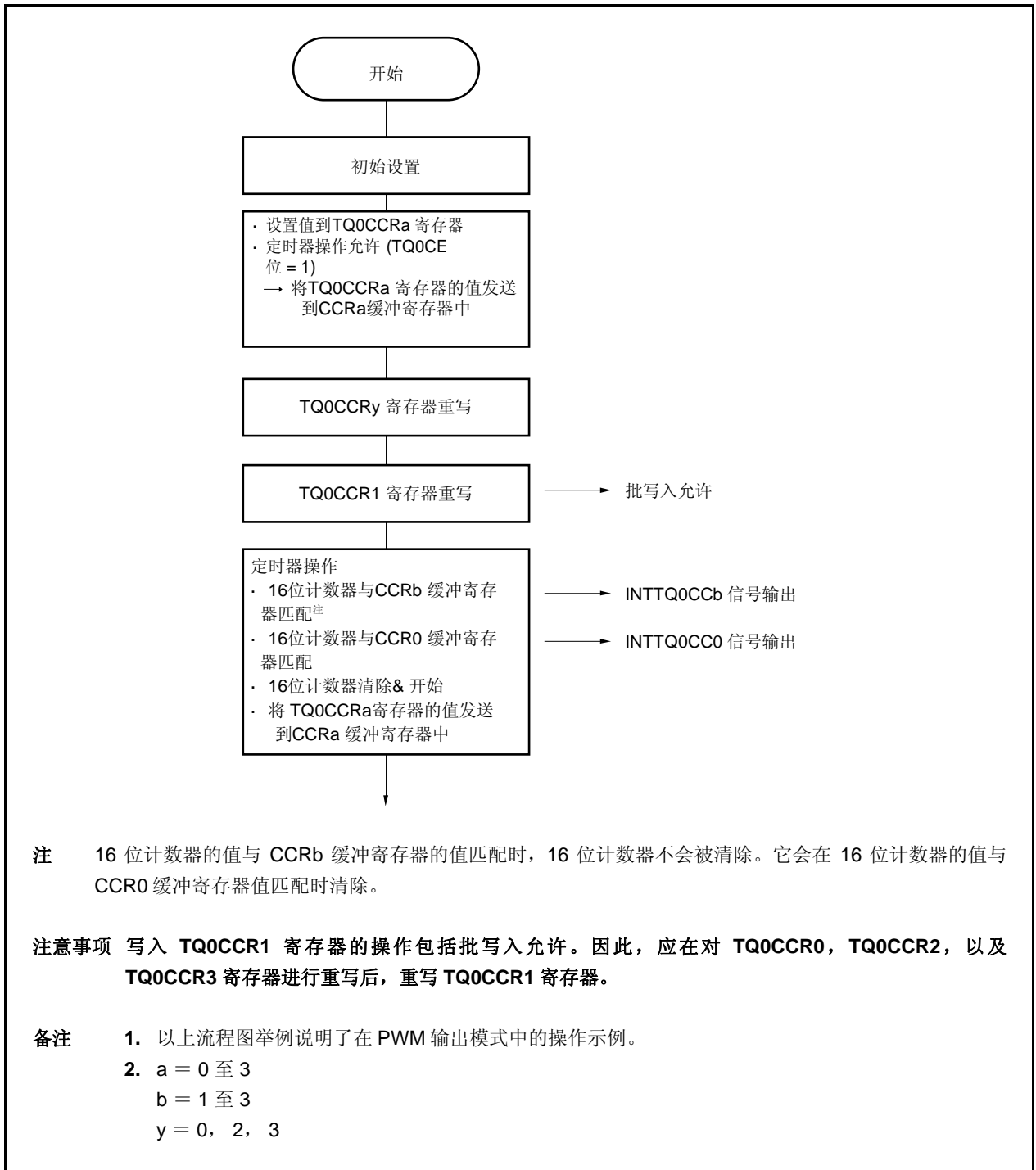
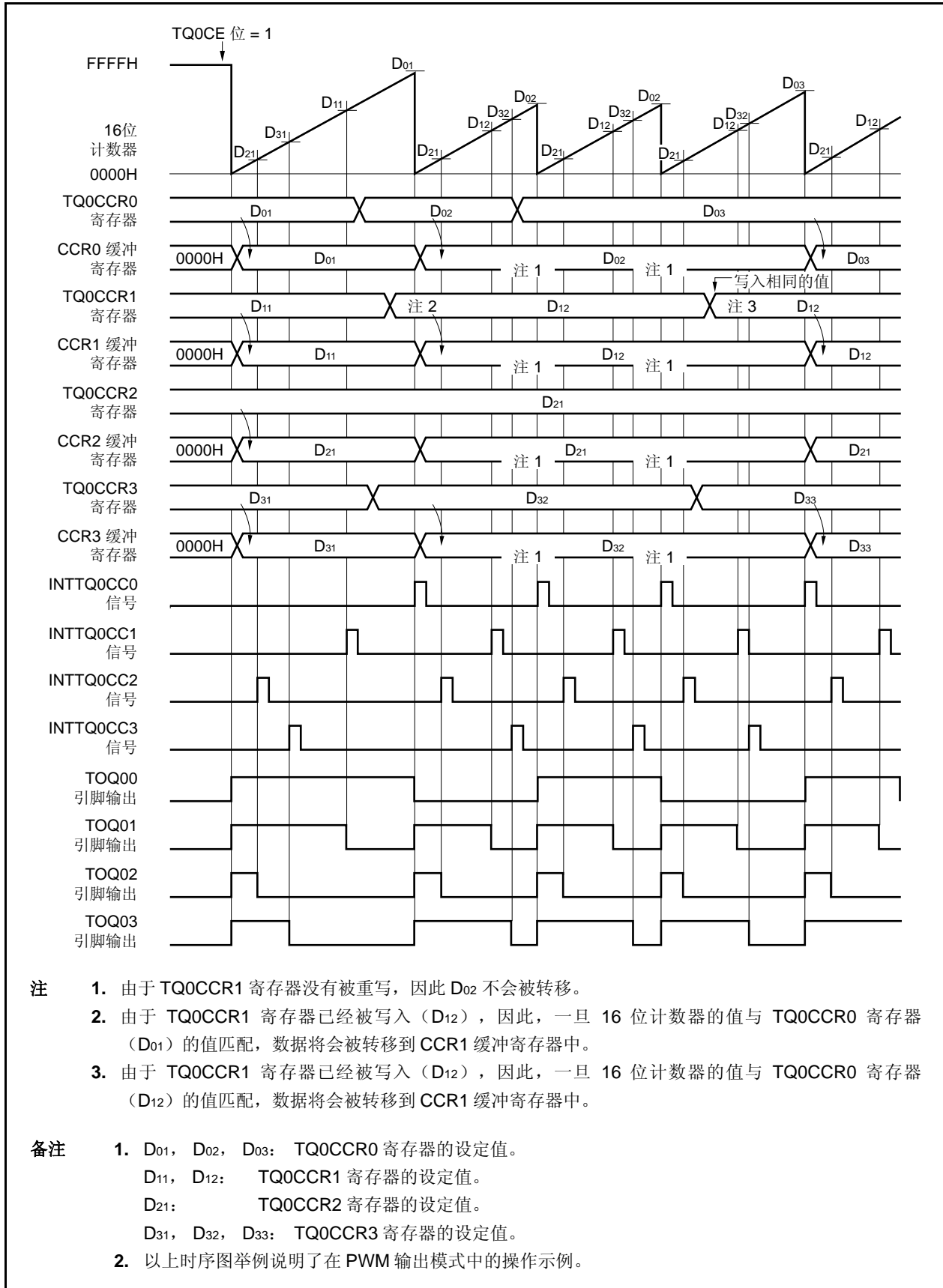


图 7-7. 批写入的时序



7.6.1 间隔定时器模式 (TQnMD2 至 TQnMD0 位=000)

在间隔定时器模式中, 如果 TQnCTL0.TQnCE 位被设置为 1, 中断请求信号 (INTTQnCC0) 将会在 TQnCCR0 寄存器所设置的时间间隔中生成。半周期等于间隔时间的 50% 占空比的 PWM 波形可以从 TOQn0 引脚中输出。

间隔定时器模式中不使用 TQnCCR1 至 TQnCCR3 寄存器。然而, TQnCCR1 至 TQnCCR3 寄存器的设定值将会转移到 CCR1 至 CCR3 缓冲寄存器中, 而在 16 位计数器的计数值与 CCR1 至 CCR3 缓冲寄存器的值匹配时将会输出比较匹配中断请求信号 (INTTQnCC1 至 INTTQnCC3)。此外, 在 INTTQnCC1 至 INTTQnCC3 信号生成时反向的带有 50% 占空比的 PWM 波形可以从 TOQ01 至 TOQ03 引脚中输出。

即便在定时器运行时, TQnCCR1 至 TQnCCR3 寄存器也可以进行重写。

图 7-8. 间隔定时器的配置

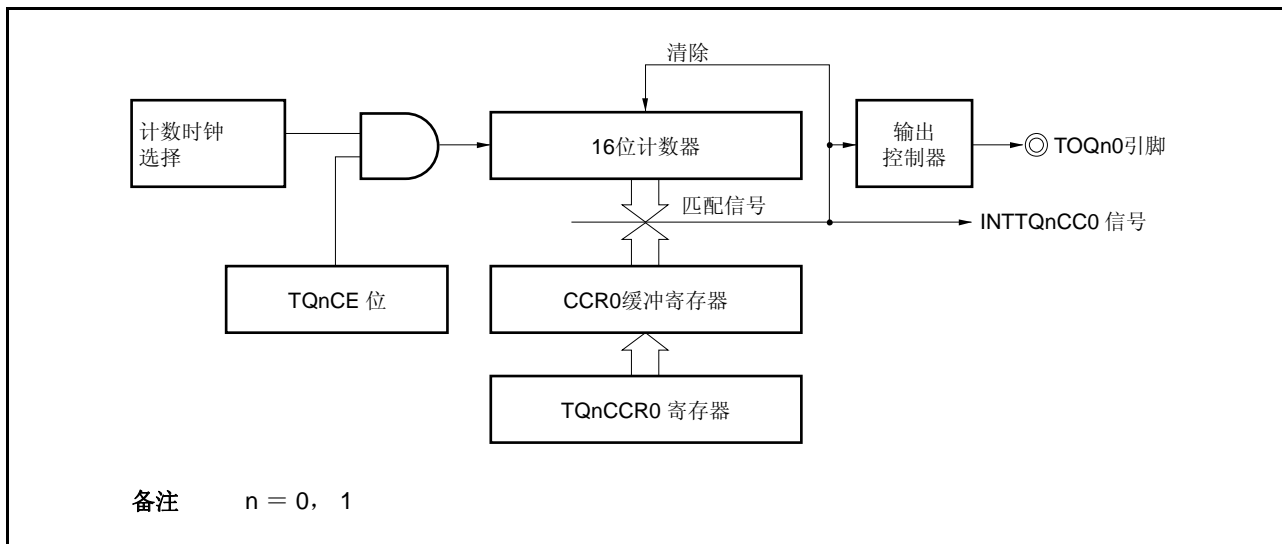
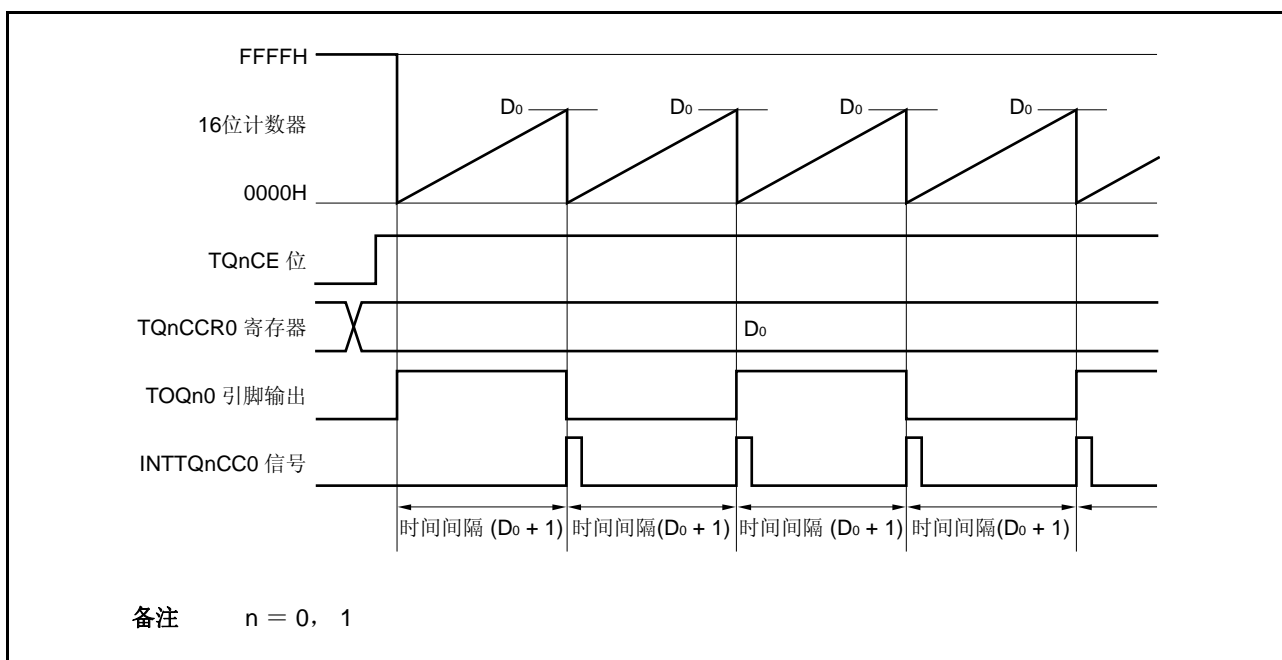


图 7-9. 间隔定时器工作的基本时序



当 TQnCE 位被置为 1 时，16 位计数器的值与计数时钟同步从 FFFFH 清零为 0000H，并且计数器开始计数，同时，TOQn0 引脚的输出反向。此外，TQnCCR0 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，16 位计数器清为 0000H，TOQn0 引脚输出反向，并生成一个比较匹配中断请求信号 (INTTQnCC0)。

时间间隔可用下式计算。

$$\text{时间间隔} = (\text{TQnCCR0 寄存器的设定值} + 1) \times \text{计数时钟周期}$$

备注 n = 0, 1

图 7-10. 间隔定时器模式操作时的寄存器设置 (1/3)

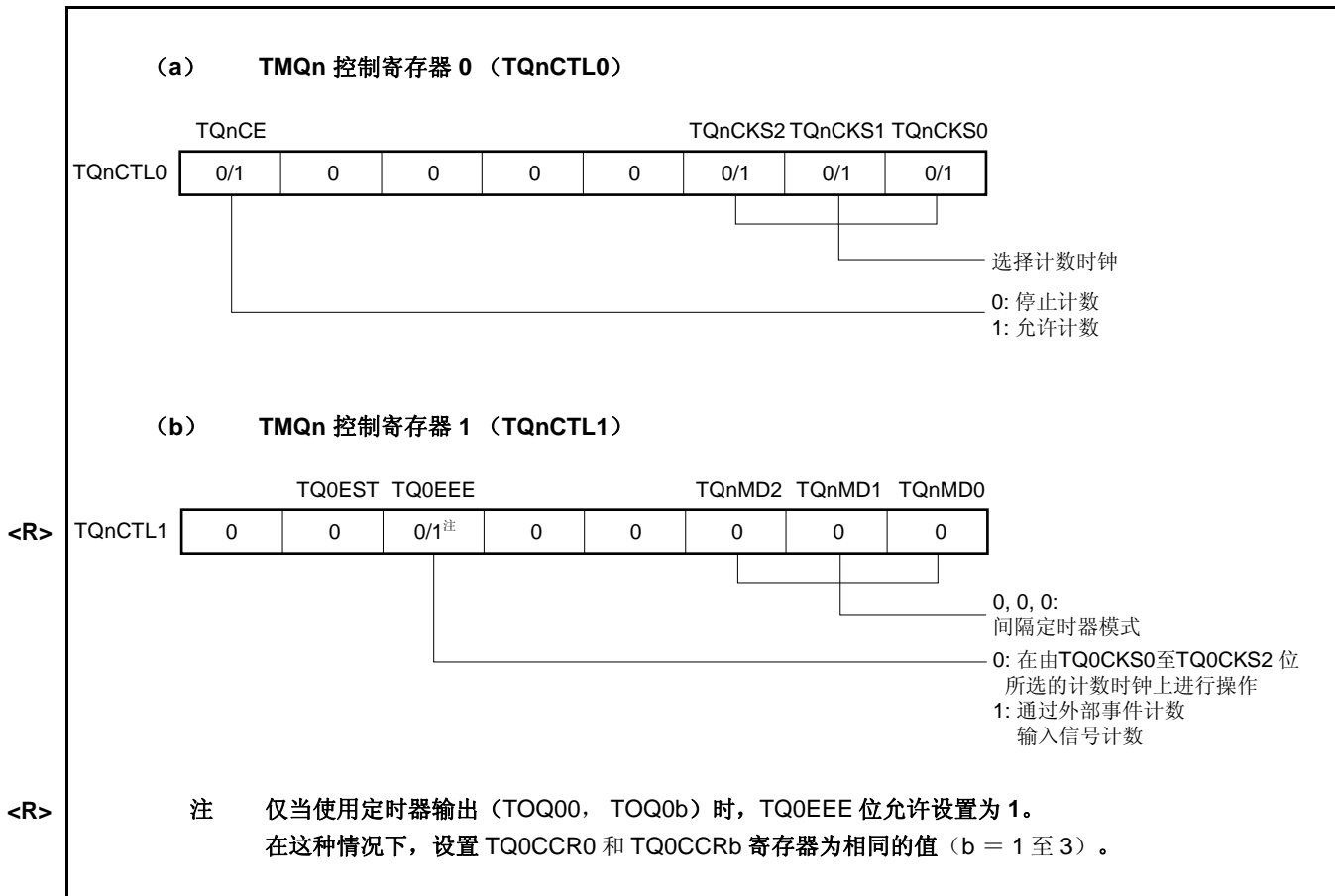
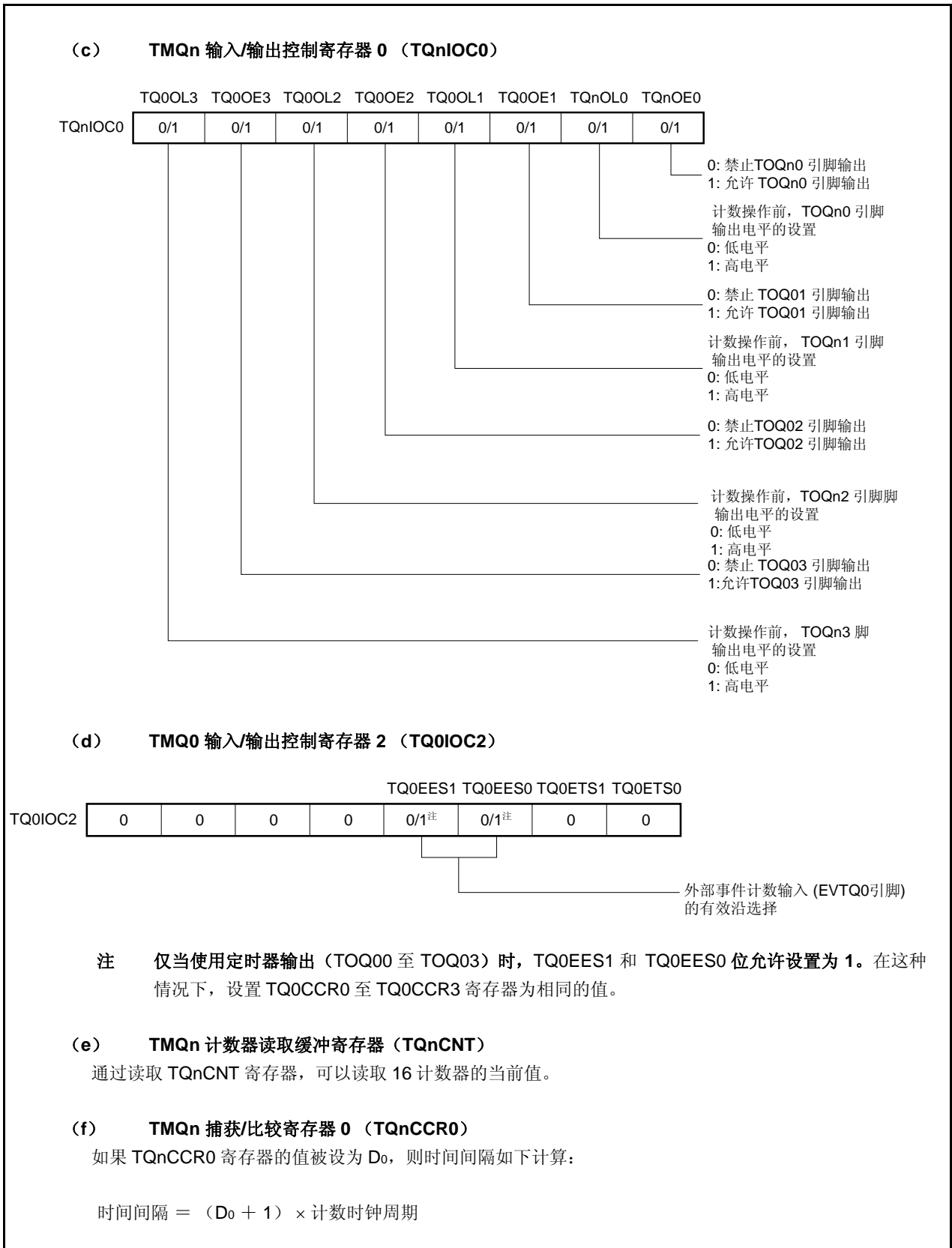


图 7-10. 间隔定时器模式操作时的寄存器设置 (2/3)



<R>

图 7-10. 间隔定时器模式操作时的寄存器设置 (3/3)

(g) TMQn 捕获/比较寄存器 1 至 3 (TQnCCR1 至 TQnCCR3)

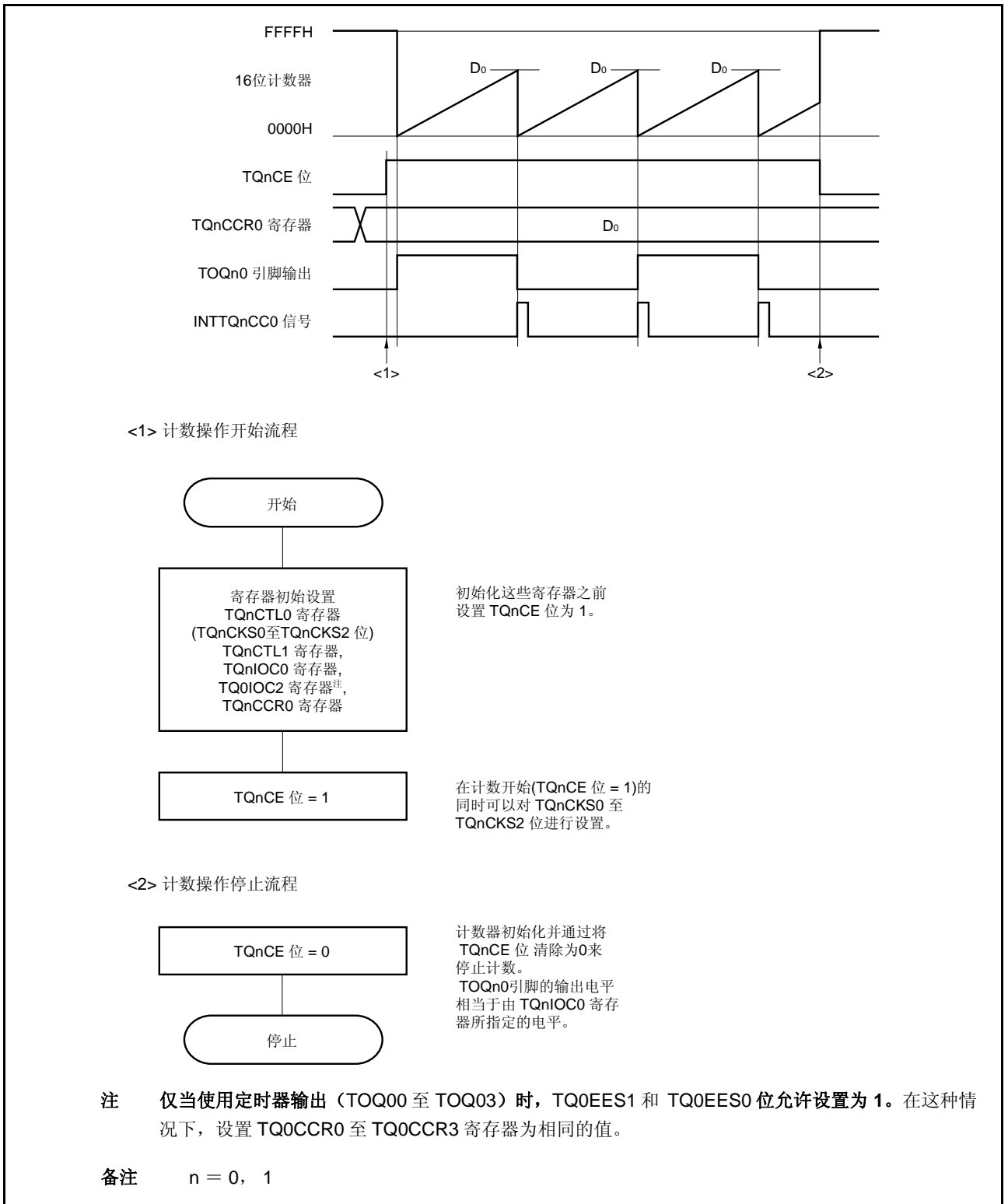
间隔定时器模式中不使用 TQnCCR1 至 TQnCCR3 寄存器。但对 TQnCCR1 至 TQnCCR3 寄存器设置的数值也将被传送到 CCR1 至 CCR3 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 至 CCR3 缓冲寄存器中的数值匹配时，TOQ01 至 TOQ03 引脚输出将会被反向并且会产生比较匹配中断请求信号 (INTTQnCC1 至 INTTQnCC3)。

TQnCCR1 至 TQnCCR3 寄存器没有被使用时，建议将其值设为 FFFFH。此外，还会通过中断屏蔽标志 (TQnCCIC1.TQnCCMK1 至 TQnCCIC3.TQnCCMK3) 来屏蔽寄存器。

- <R> 备注
1. 在脉宽调制信号输出模式时，不使用 TMQ0 输入/输出控制寄存器 1 (TQ0IOC1) 和 TMQn 选择寄存器 0 (TQnOPT0)。
 2. $n = 0, 1$

(1) 间隔定时器模式工作流程

图 7-11. 间隔定时器模式下的软件处理流程

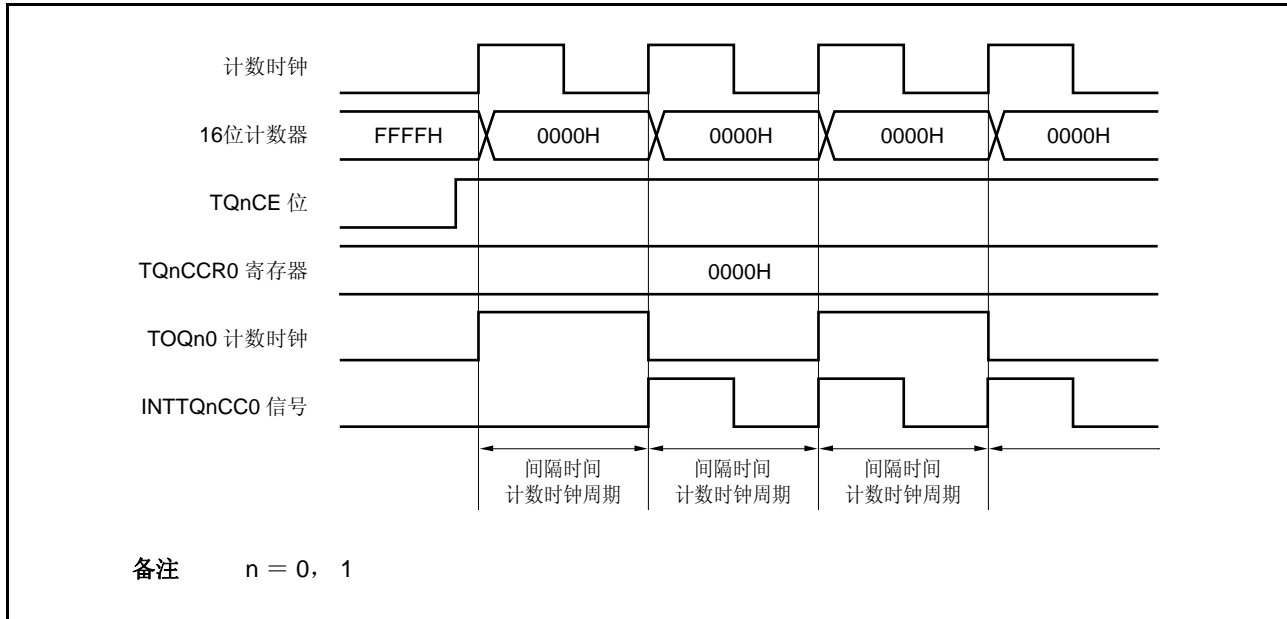


(2) 间隔定时器模式工作时序

(a) TQnCCR0 寄存器被设置为 0000H 时的操作

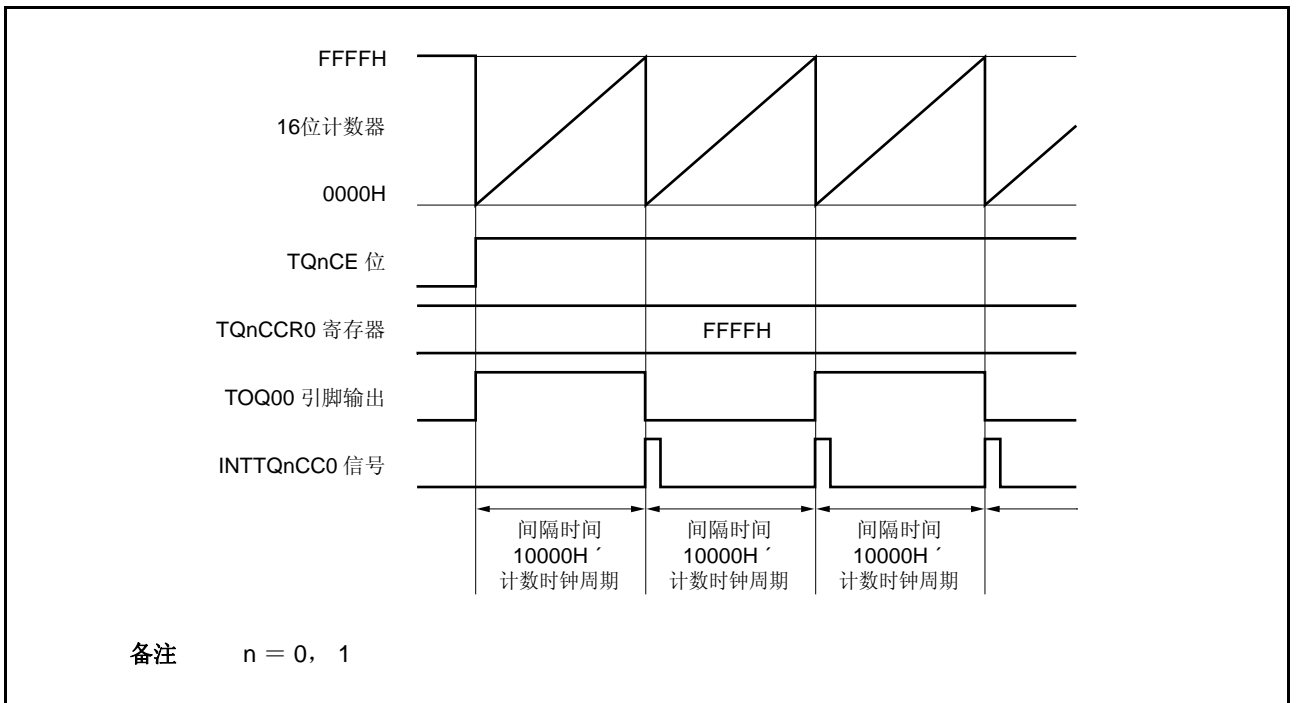
如果 TQnCCR0 寄存器被设置为 0000H，每个计数时钟都会产生 INTTQnCC0 信号，而且，TOQn0 引脚的输出反向。

16 位计数器的值总是 0000H。



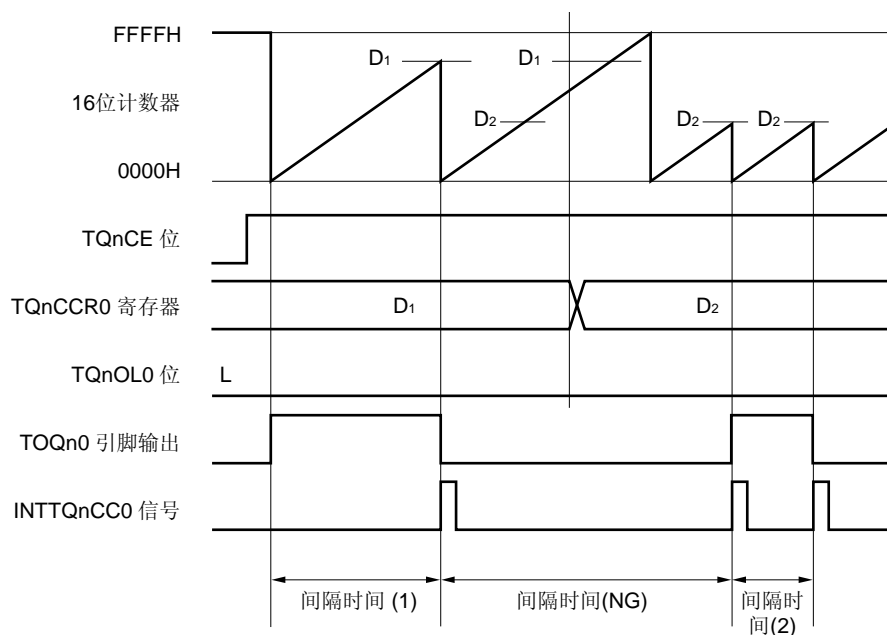
(b) TQnCCR0 寄存器被设置为 FFFFH 时的操作

如果 TQnCCR0 寄存器被设置为 FFFFH，那么 16 位计数器可以计数直至 FFFFH。并在下一个计数时钟时刻同步被清除为 0000H。产生 INTTQnCC0 信号，并且 TOQn0 引脚输出电平反向。此时，并不产生溢出中断请求信号 (INTTQnOV)，而且，溢出标志 (TQnOPT0.TQnOVF 位) 也不会置 1。



(c) 重写 TQnCCR0 寄存器的注意事项

如果在计数器工作期间，将 TQnCCR0 寄存器的值重写为比当前值较小的值，16 位计数器会发生溢出。当发生溢出时，将停止一次计数并更改设定值。

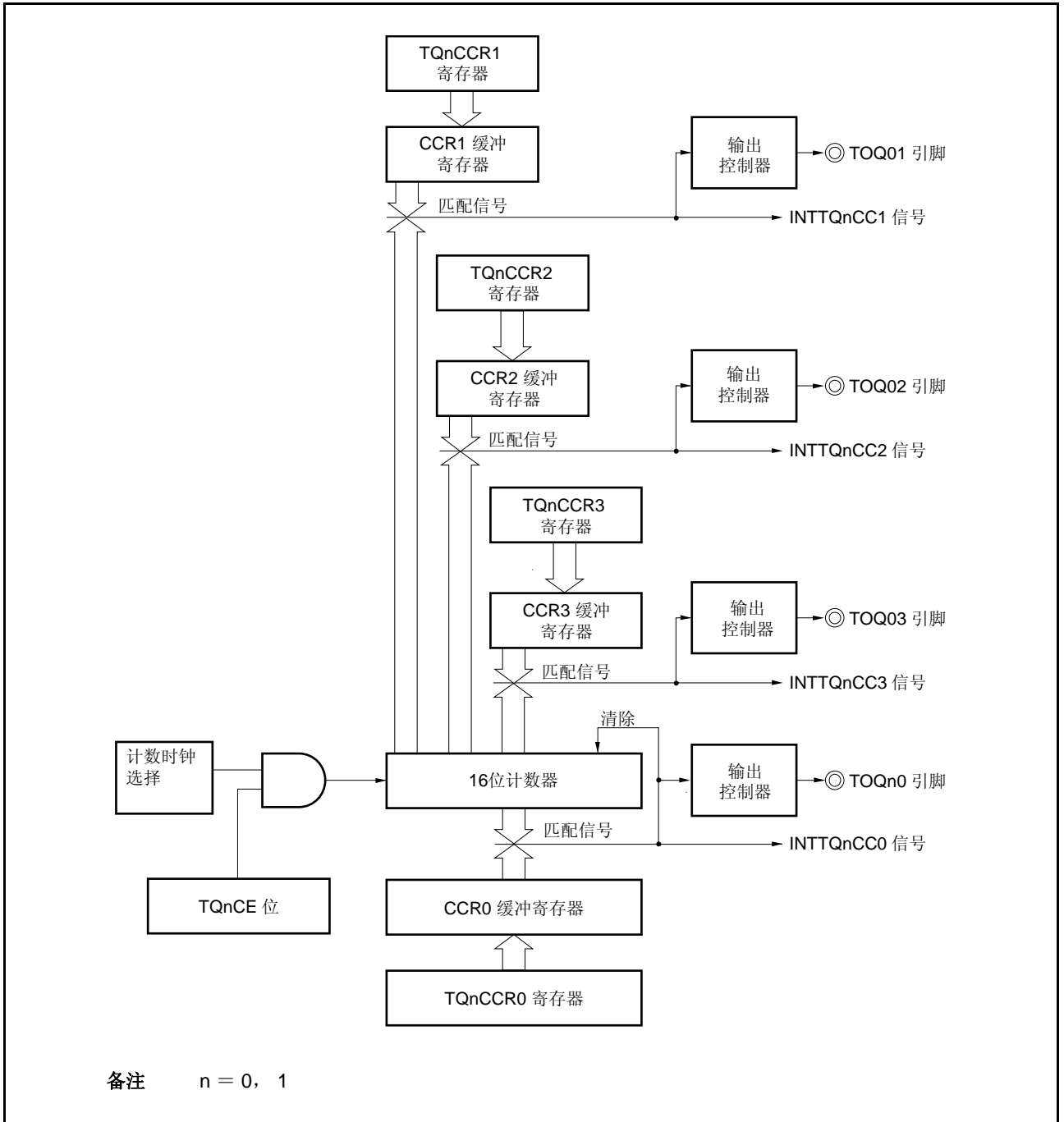


- 备注**
1. 间隔时间 (1) : $(D_1 + 1) \times$ 计数时钟周期
 间隔时间 (NG) : $(10000H + D_2 + 1) \times$ 计数时钟周期
 间隔时间 (2) : $(D_2 + 1) \times$ 计数时钟周期
 2. $n = 0, 1$

当 16 位计数器计数值大于 D_2 而小于 D_1 时，如果将 TQnCCR0 寄存器的值从 D_1 变为 D_2 ，则在重写 TQnCCR0 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_1 ，所以 16 位计数器就会一直向上累积计数到 FFFFH 直到溢出，然后从 0000H 开始重新计数。当计数值和 D_2 相等时，会生成 INTTQnCC0 信号，且 TOQn0 引脚输出信号反向。因此，INTTQnCC0 信号可能不会在原先预计的间隔时间“ $(D_1 + 1) \times$ 计数时钟周期”或“ $(D_2 + 1) \times$ 计数时钟周期”下产生，而会以“ $(10000H + D_2 + 1) \times$ 计数时钟周期”的间隔产生。

(d) TQnCCR1 至 TQnCCR3 寄存器的操作

图 7-12. TQnCCR1 至 TQnCCR3 寄存器的配置



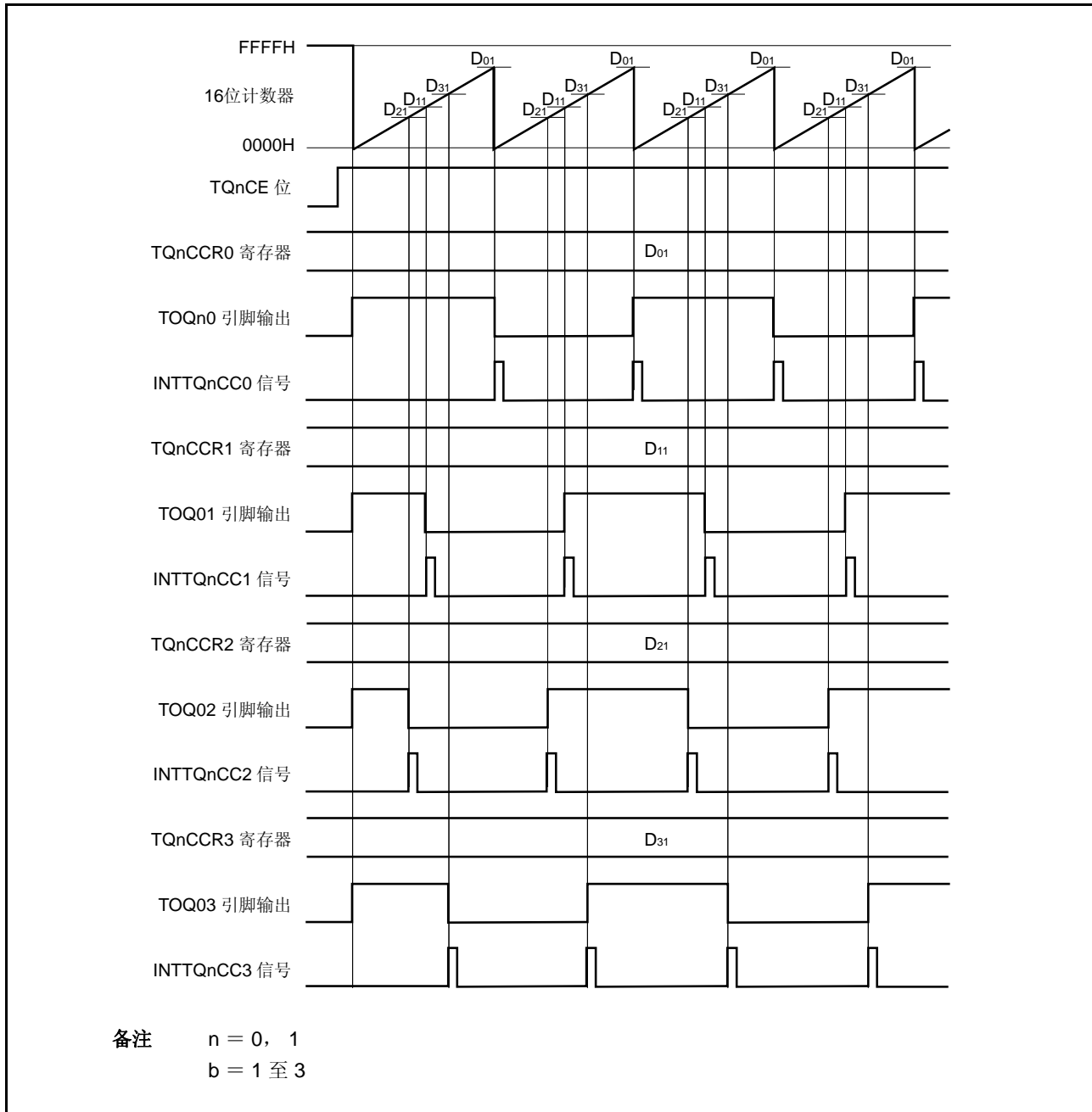
<R> 当 TQnCCRb 寄存器的值被设为与 TQnCCR0 寄存器相同的值时，INTTQnCCb 信号将会在 INTTQnCC0 信号与 TOQ0b 引脚输出被反向的同时生成。换句话说，带有 50% 占空比的 PWM 波形可以从 TOQ0b 引脚中输出。

以下所示为将不同于 TQnCCR0 寄存器中所设定的值设置至 TQnCCRb 寄存器时的操作。

如果 TQnCCRb 寄存器的设置值小于 TQnCCR0 寄存器的设置值，INTTQnCCb 信号每计数时钟周期产生一次，同时，TOQ0b 引脚输出反向。

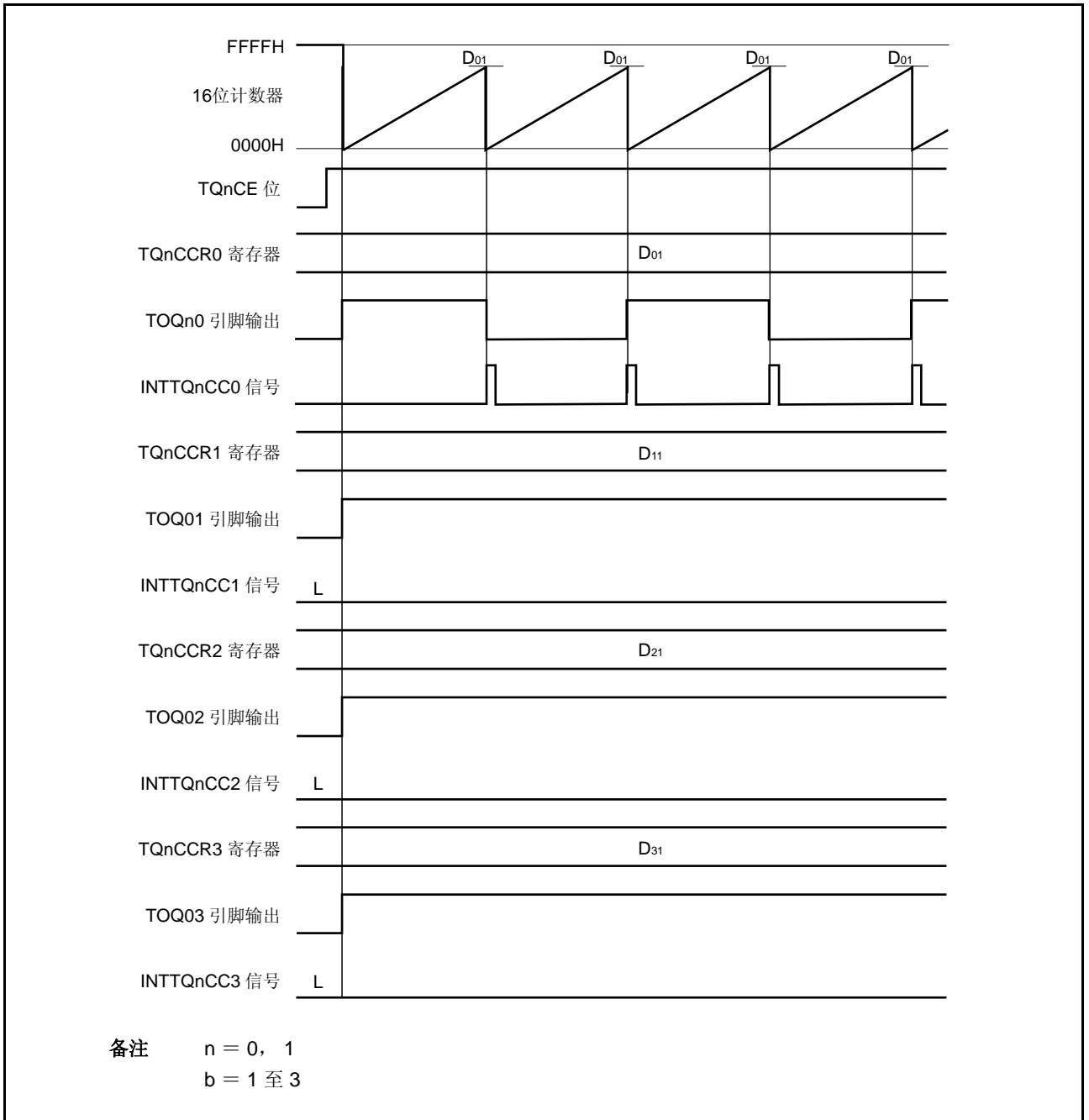
<R> 之后首先输出短宽度脉冲，TOQ0b 引脚输出一个带有 50% 占空比的 PWM 波形。

图 7-13. 当 $D_{01} \geq D_{b1}$ 时的时序图



如果 TQnCCRb 寄存器的设置值大于 TQnCCR0 寄存器的设置值，16 位计数器的计数值不能与 TQnCCRb 寄存器的值相匹配，因而，不会产生 INTTQnCCb 信号，TOQ0b 引脚输出也不会改变。
 当不使用 TQnCCRb 寄存器时，建议将其值设为 FFFFH。

图 7-14. 当 $D_{01} < D_{b1}$ 时的时序图



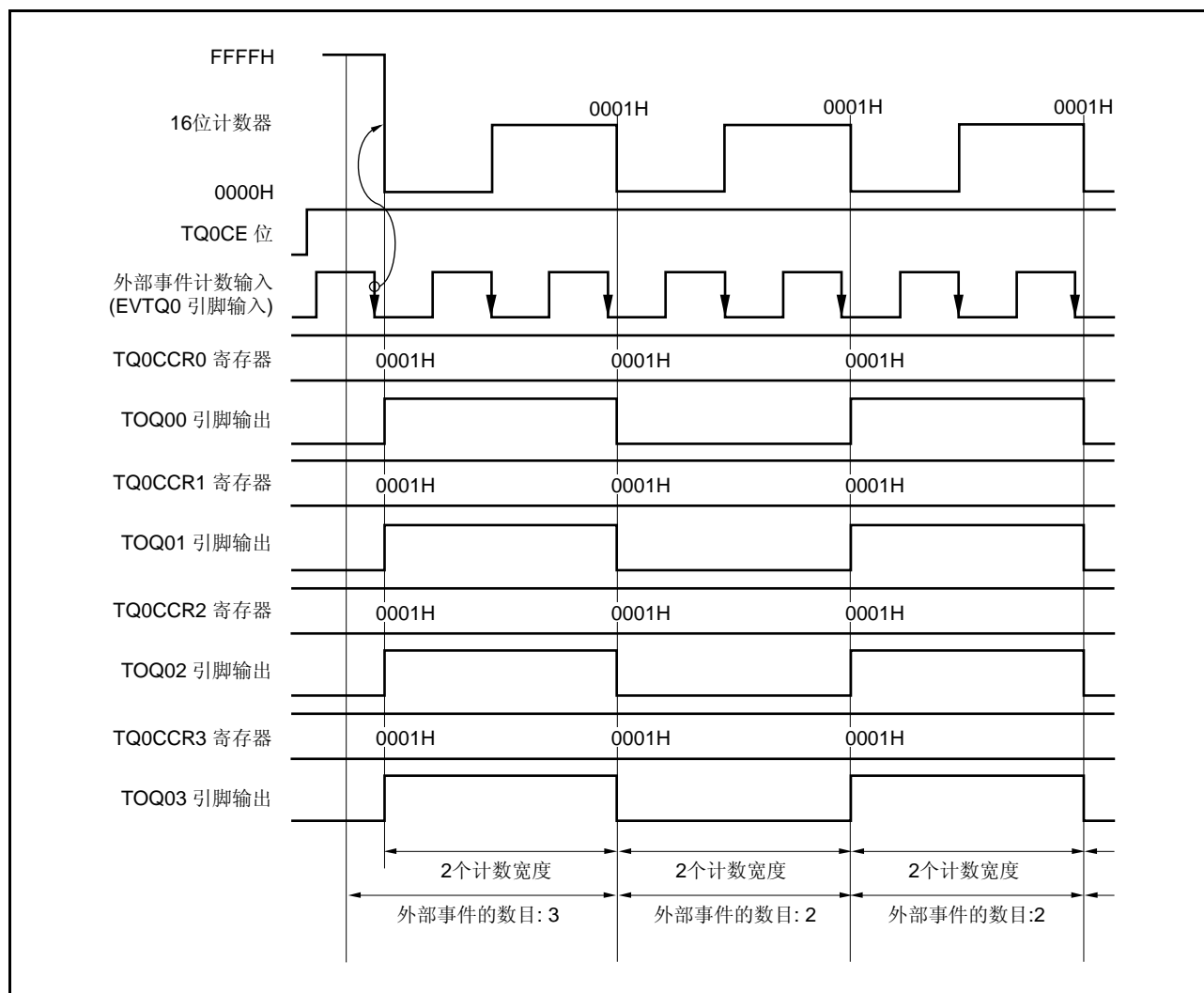
<R> (3) 外部事件计数输入操作 (EVTQ0)

(a) 操作

在间隔定时器模式下，来计数 16 位计数器在外部事件计数输入 (EVTQ0) 的有效沿，在 TQ0CE 位的设置从 0 到 1 后，在第一个外部事件计数输入的有效沿清除 16 位计数器从 FFFFH 至 0000H。

当 TQ0CCR0 和 TQ0CCRb 寄存器都设置为 0001H 时，16 位计数器每计数两次 TOQ00 和 TOQ0b 引脚的输出反向一次。

在间隔定时器模式下，仅当定时器的输出 (TOQ00, TOQ0b) 用作外部事件计数输入时，TQ0CTL0.TQ0EEE 位能被设置为 1。



7.6.2 外部事件计数器模式 (TQ0MD2 至 TQ0MD0 位=001)

该模式仅在 TMQ0 中有效。

在外部事件计数器模式下，当 TQ0CTL0.TQ0CE 位被设置为 1 时，外部事件计数输入信号的有效边沿被计数，而且每次计数到 TQ0CCR0 寄存器所设置边沿数量后都将会产生中断请求信号 (INTTQ0CC0)。不使用 TOQ00 至 TOQ03 引脚。在外部事件计数输入模式下使用 TOQ00 至 TOQ03 引脚时，在间隔定时器模式下设置 TQ0CTL1.TQ0EEE 位为 1 (参见 7.6.1 (3) 外部事件计数输入 (EVTQ0) 的操作)。

外部事件计数模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。

注意事项 外部事件计数器模式下，TQ0CCR0 和 TQ0CCR3 寄存器不能清除为 0000H。

图 7-15. 外部事件计数器模式的配置

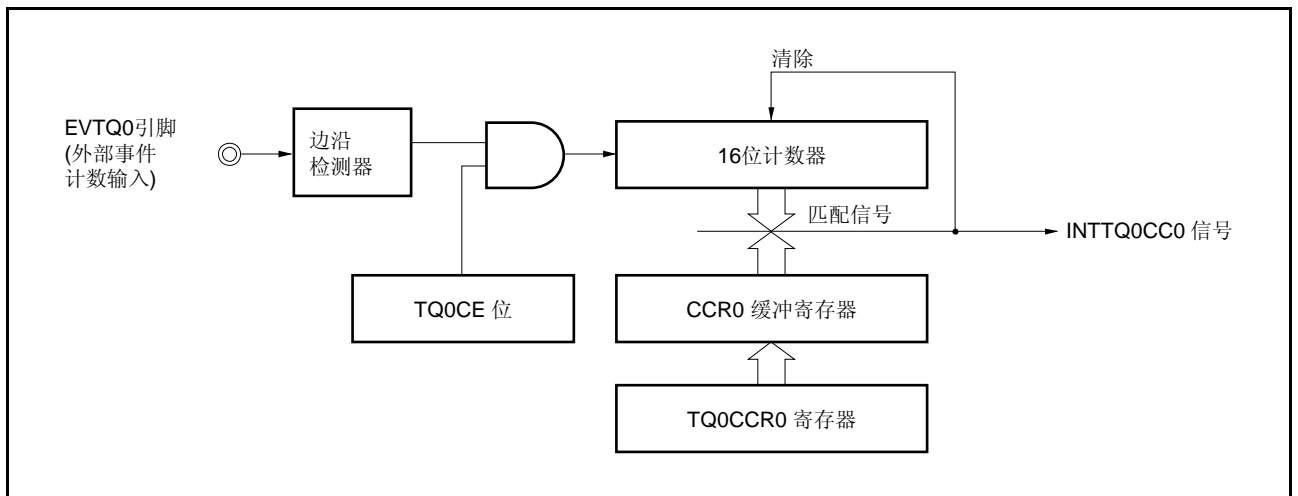
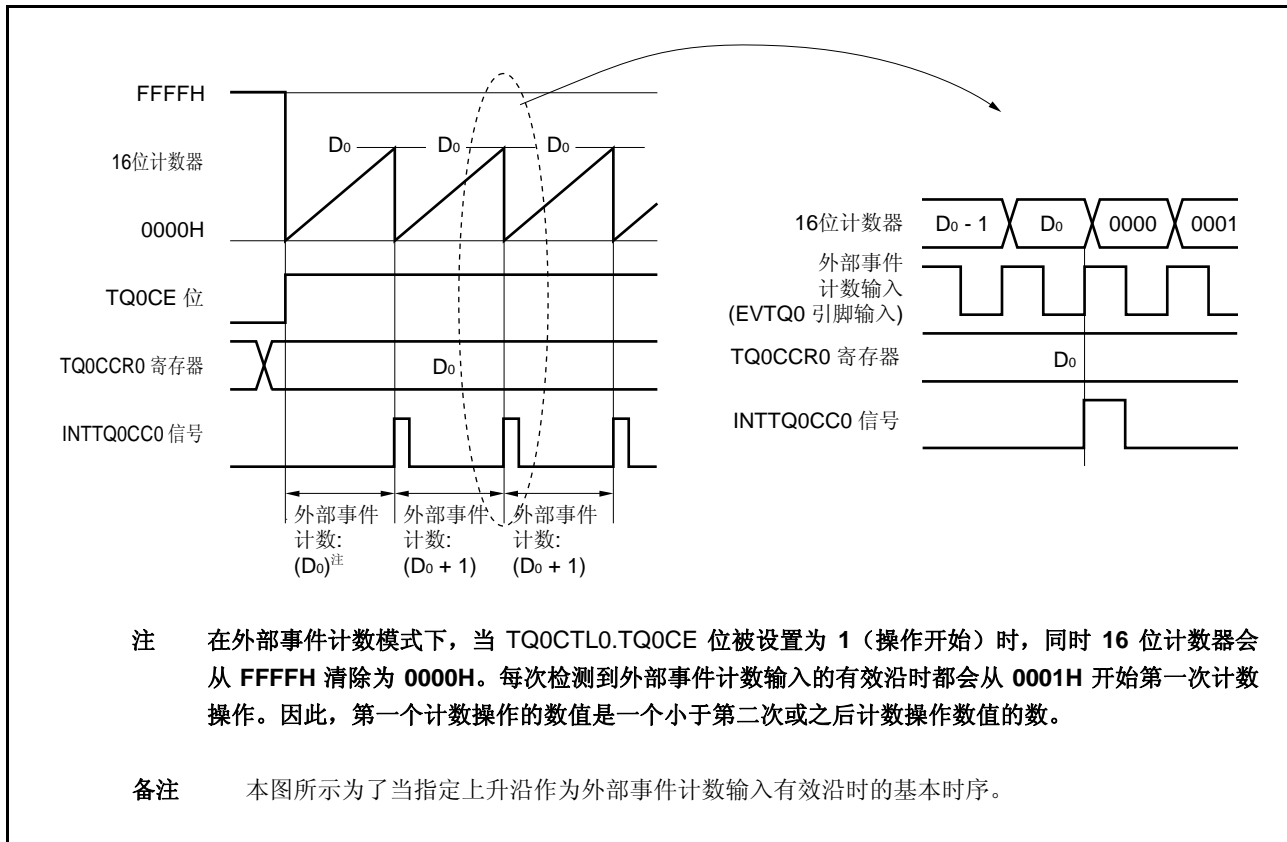


图 7-16. 外部事件计数器模式的基本时序



当 TQ0CE 位被设置为 1 时，16 位计数器的值由 FFFFH 被清为 0000H，计数器在每次检测到外部事件计数输入信号的有效边沿时进行一次计数。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 缓冲寄存器的数值相匹配时，16 位计数器被清零为 0000H，同时生成一个比较匹配中断请求信号 (INTTQ0CC0)。

当外部事件计数输入的有效沿被检测的次数等于“TQ0CCR0 寄存器中所设置的值”时，将会第一次生成 INTTQ0CC0 信号。在此之后，每当检测到“TQ0CCR0 寄存器设置值 + 1”次外部事件计数的有效沿时，都会产生 INTTQ0CC0 信号。

图 7-17. 外部事件计数模式下的寄存器操作设置 (1/2)

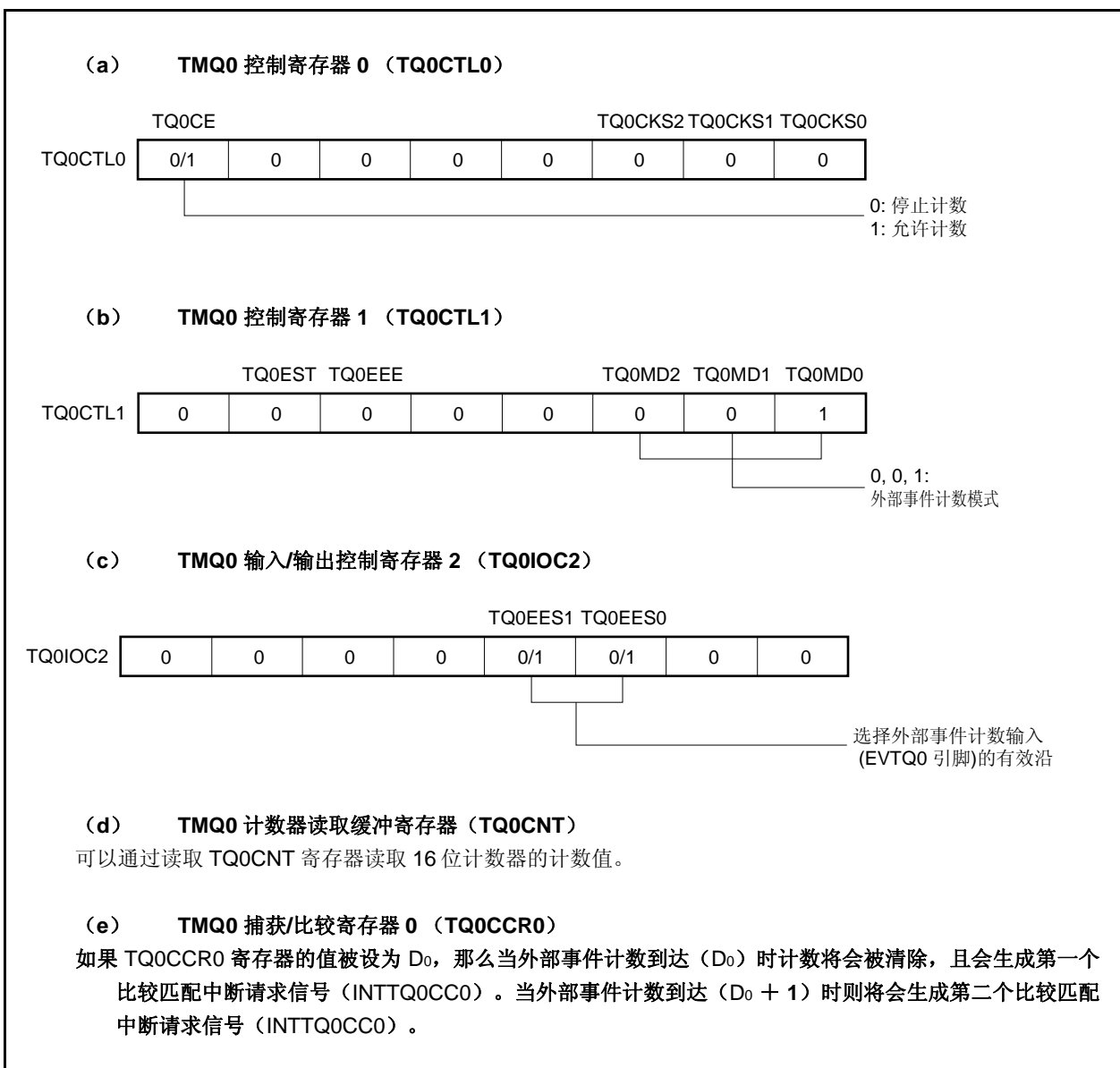


图 7-17. 外部事件计数模式下的寄存器操作设置 (2/2)

(f) TMQ0 捕获/比较寄存器 1 至 3 (TQ0CCR1 至 TQ0CCR3)

外部事件计数模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。但对 TQ0CCR1 至 TQ0CCR3 寄存器设置的数值也将被传送到 CCR1 至 CCR3 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 至 CCR3 缓冲寄存器中的数值匹配时, 产生比较匹配中断请求信号 (INTTQ0CC1 至 INTTQ0CC3)。

TQ0CCR1 至 TQ0CCR3 寄存器没有被使用时, 建议将其值设为 FFFFH。此外, 还会通过中断屏蔽标志 (TQ0CCIC1.TQ0CCMK1 至 TQ0CCIC3.TQ0CCMK3) 来屏蔽寄存器。

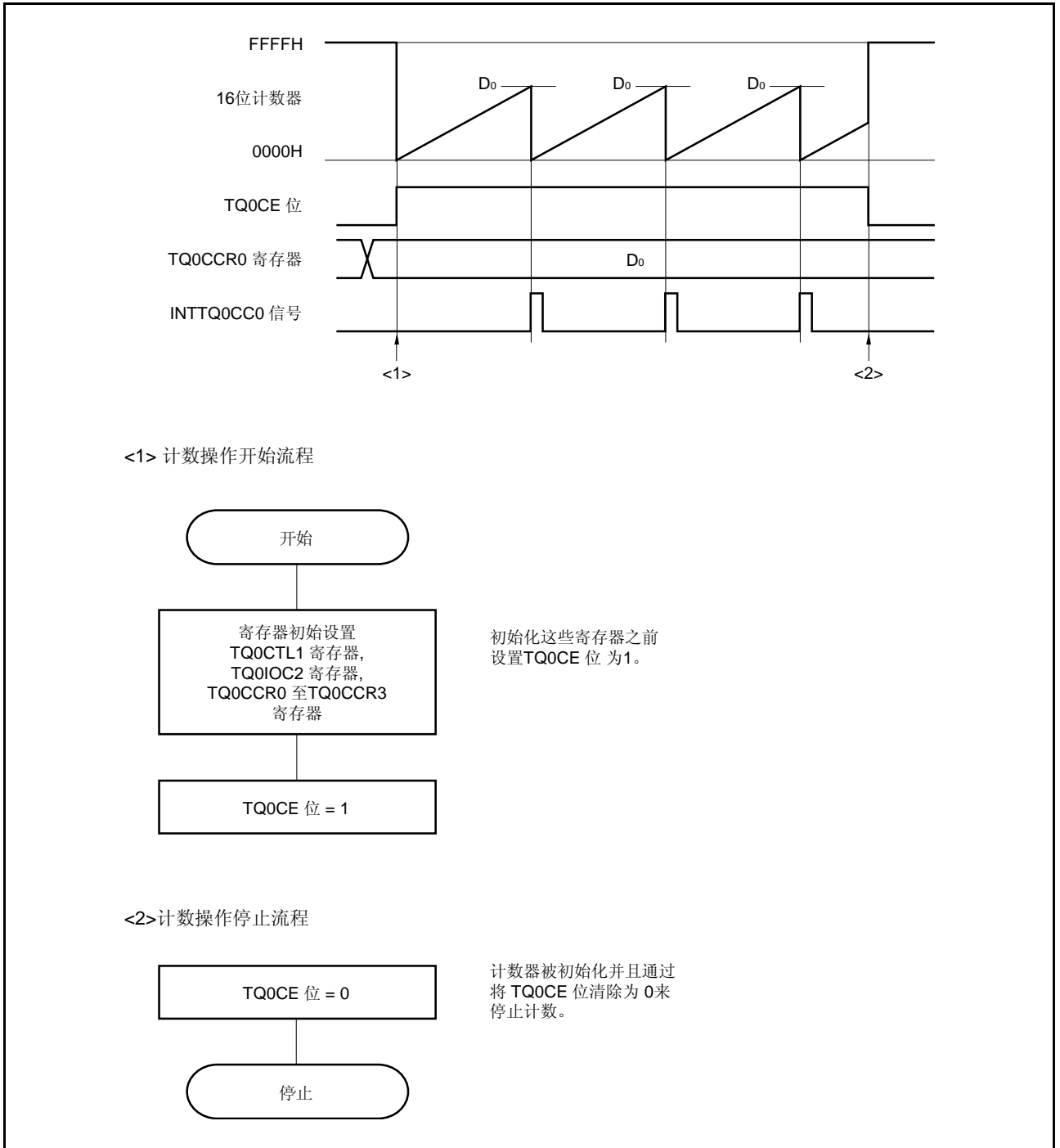
<R>

注意事项 将 TQ0IOC0 寄存器设为 00H。

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 选择寄存器 0 (TQ0OPT0)。

(1) 外部事件计数器模式的操作流程

图 7-18. 外部事件计数器模式下的软件处理流程



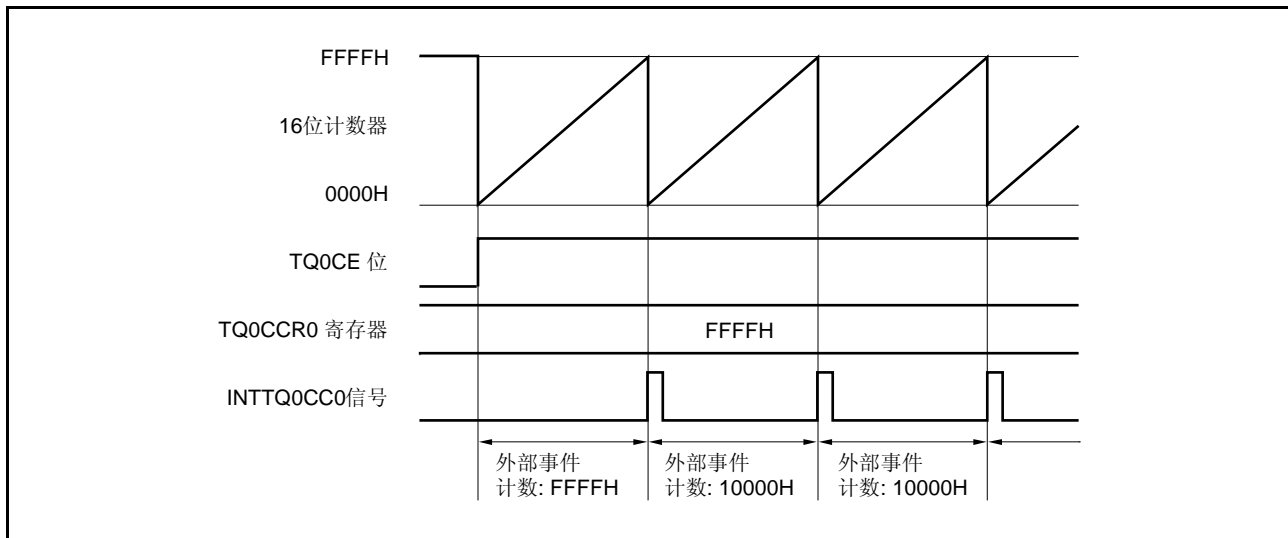
(2) 外部事件计数器模式的工作时序

注意事项 1. 外部事件计数器模式下，TQ0CCR0 和 TQ0CCR3 寄存器不能清除为 0000H。

<R> 2. 在外部事件计数模式下，定时器输出 (TOQ00 至 TOQ03) 的使用被禁止。如果使用定时器输出 (TOQ00 至 TOQ03) 与外部事件计数输入 (EVTQ0)，设置间隔定时器模式，并且允许计数时钟操作与外部事件计数输入 (TQ0CTL1.TQ0EEE 位 = 1) (参见 7.6.1 (3) 外部事件计数输入 (EVTQ0) 的操作)。

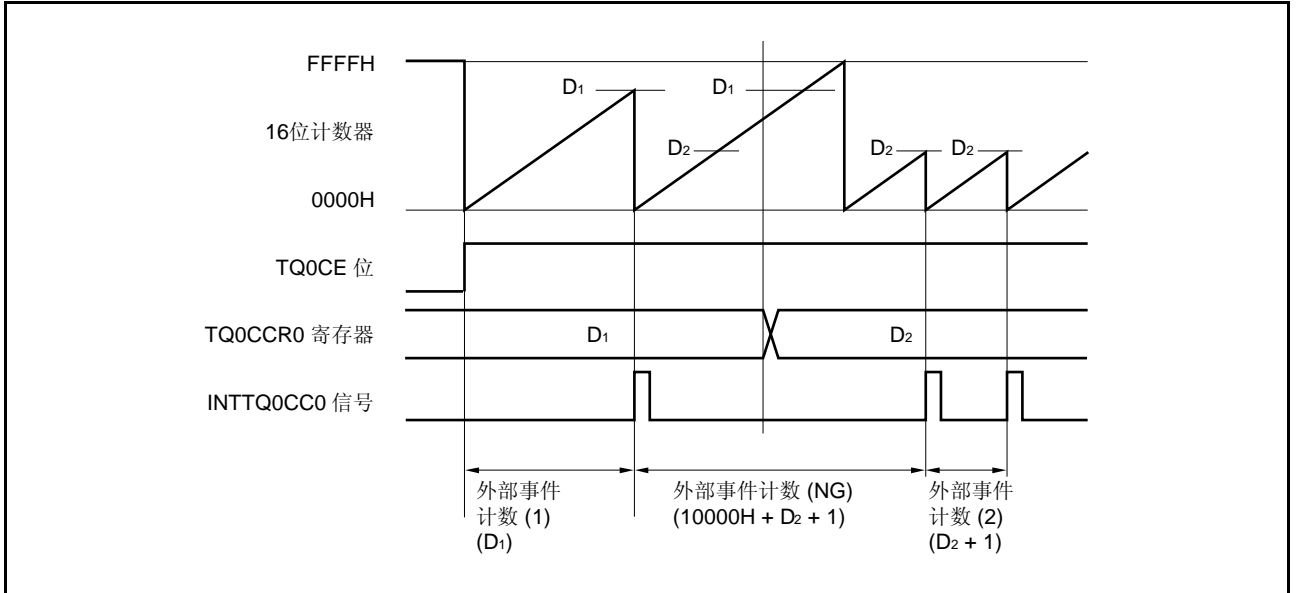
(a) TQ0CCR0 寄存器被设置为 FFFFH 时的操作

如果 TQ0CCR0 寄存器被设置为 FFFFH，16 位计数器每次计数是在检测到的外部事件输入信号有效边沿时，直到计数达到 FFFFH，然后，16 位计数器于下一次计数时同步清为 0000H，同时，产生 INTTQ0CC0 信号。此时，不设置 TQ0OPT0.TQ0OVF 位。



(b) 重写 TQ0CCR0 寄存器时的注意事项

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，16 位计数器会发生溢出。当发生溢出时，将停止一次计数并更改设定值。

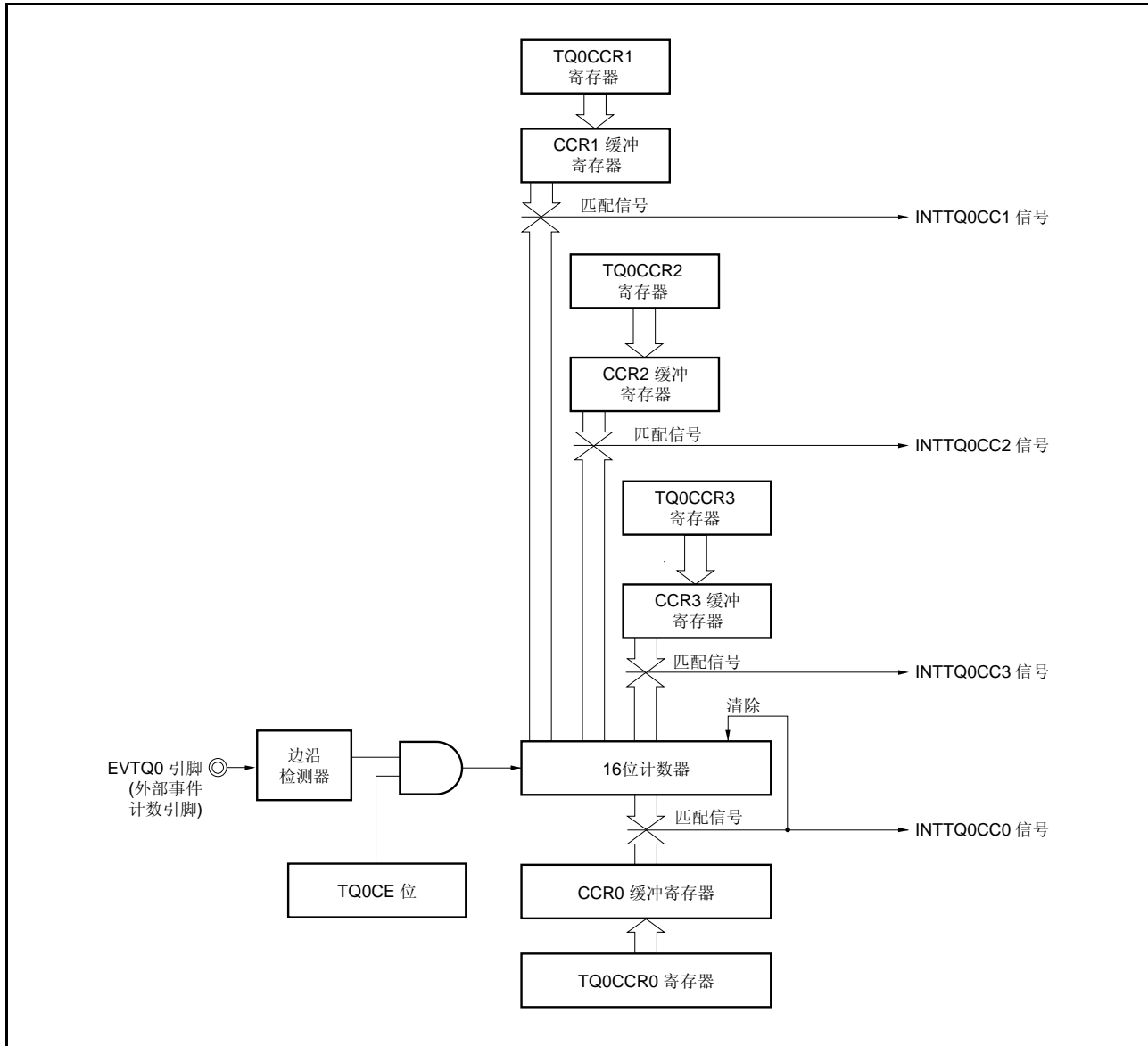


当 16 位计数器计数值大于 D_1 而小于 D_2 时，如果将 TQ0CCR0 寄存器的值从 D_1 变为 D_2 ，则在重写 TQ0CCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。因而，与 16 位计数器相比较的值会是 D_2 。但是，由于计数值已经超过了 D_2 ，所以 16 位计数器就会一直向上累积计数到 FFFFH 直到溢出，然后从 0000H 开始重新计数。当计数值与 D_2 匹配时，会生成 INTTQ0CC0 信号。

因此，INTTQ0CC0 中断请求信号可能不会在原先预计的“ $(D_1 + 1)$ 次”或“ $(D_2 + 1)$ 次”有效边缘上生成，而可能在“ $(10000H + D_2 + 1)$ 次”有效边缘上生成。

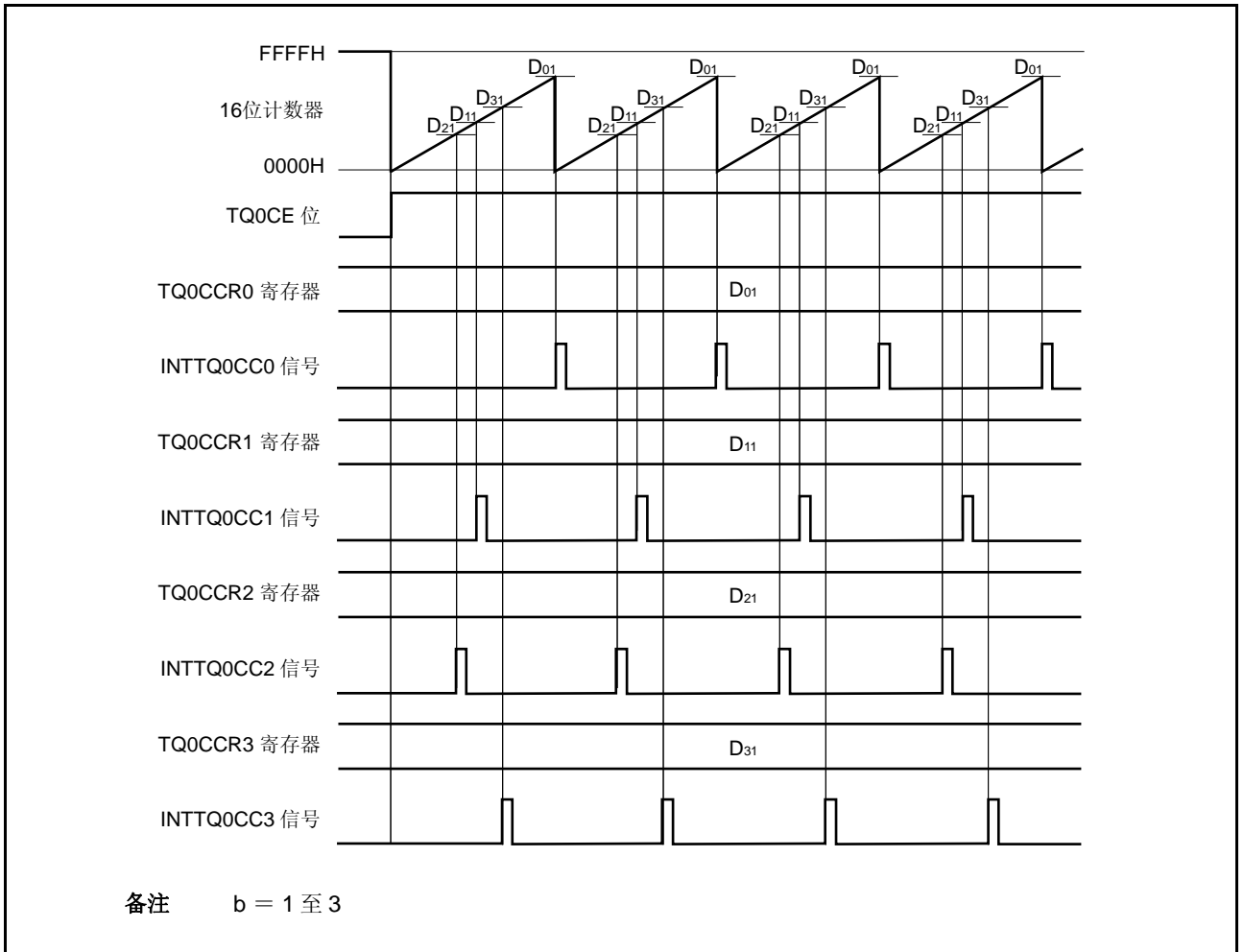
(c) TQ0CCR1 至 TQ0CCR3 寄存器的操作

图 7-19. TQ0CCR1 至 TQ0CCR3 寄存器的配置



如果 TQ0CCRb 寄存器的设定值小于 TQ0CCR0 寄存器的设定值，则 INTTQ0CCb 中断请求信号每个计数周期产生一次。

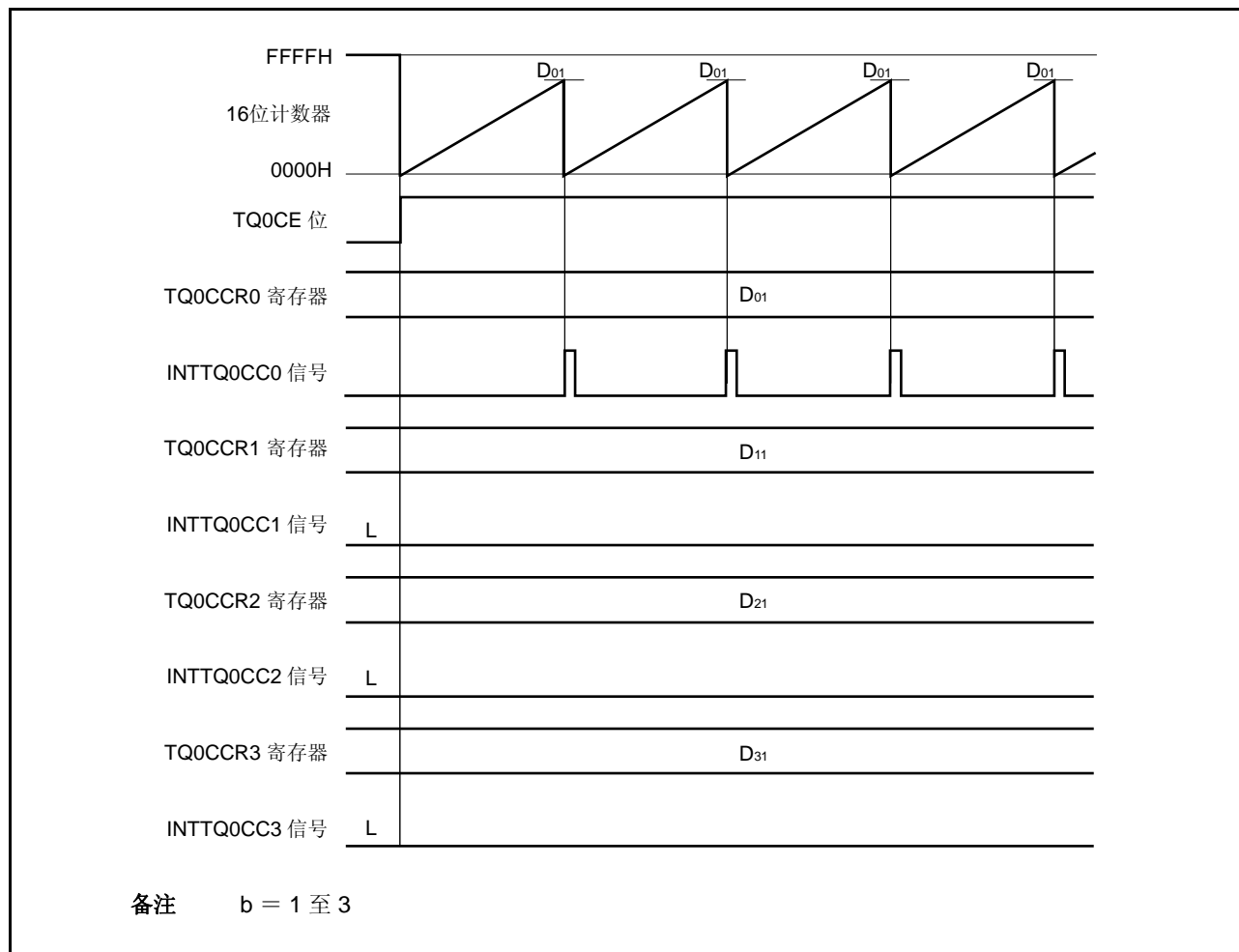
图 7-20. 当 $D_{01} \geq D_{b1}$ 时的时序图



如果 TQ0CCRb 寄存器的设定值大于 TQ0CCR0 寄存器的设定值，则不会产生 INTTQ0CCb 中断请求信号，因为 16 位计数器的计数值和 TQ0CCRb 寄存器的值不匹配。

当不使用 TQ0CCRb 寄存器时，建议将其值设为 FFFFH。

图 7-21. 当 $D_{01} < D_{b1}$ 时的时序图



7.6.3 外部触发脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 010)

该模式仅在 TMQ0 中有效。

在外部触发脉冲输出模式下，当 TQ0CTL0.TQ0CE 位被设置为 1 时，16 位定时器/事件计数器 Q 等待触发，当产生软件触发时，16 位定时器/事件计数器 Q 开始计数，并且从 TOQ01 至 TOQ03 引脚输出一个脉宽调制信号波形。带有 50% 占空比的 PWM 波形也可以从 TOQ00 引脚中输出，它的半周期等于 TQ0CCR0 寄存器的设定值 + 1。

不提供外部触发输入引脚。

图 7-22. 外部触发脉冲输出模式的配置

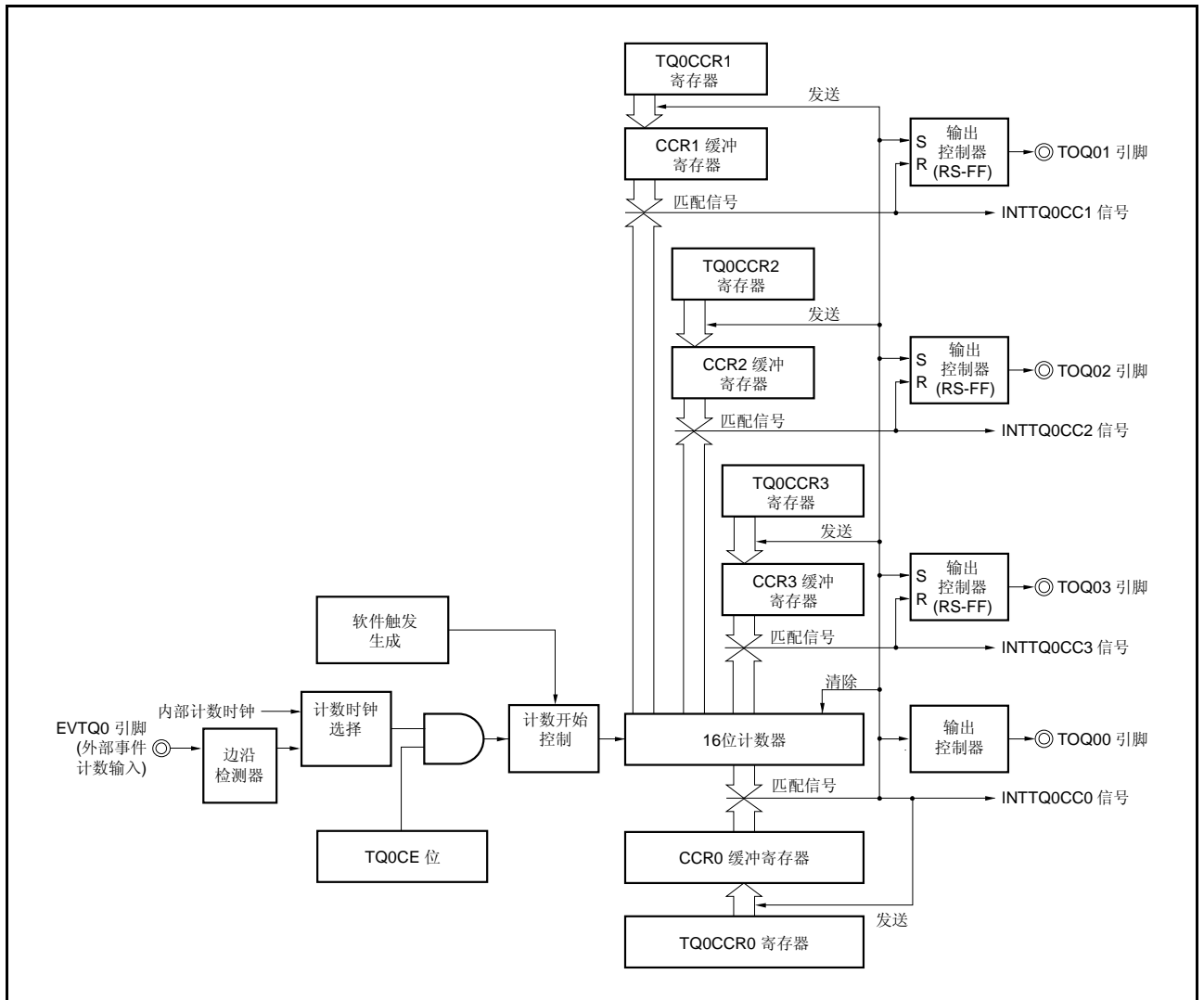
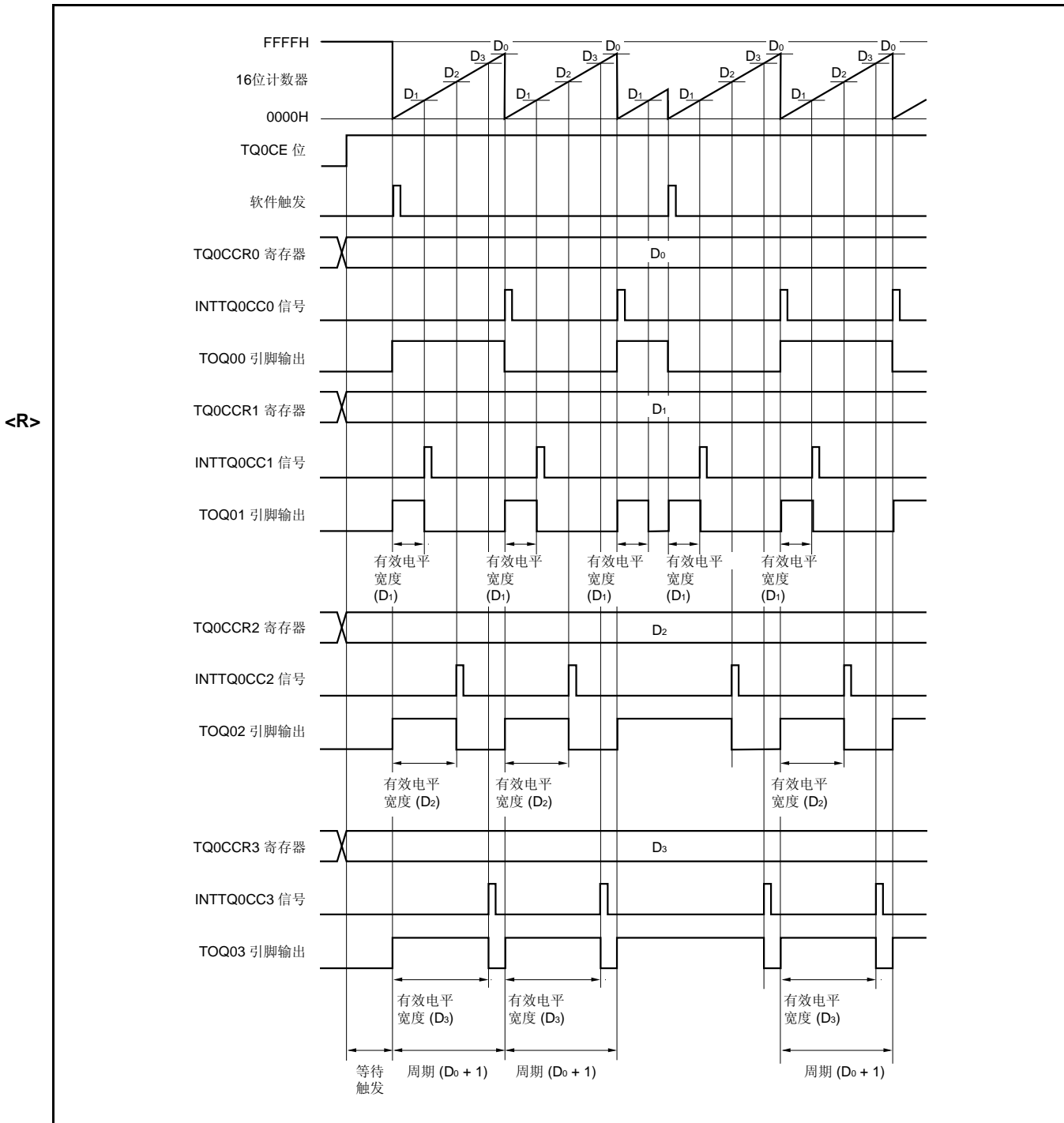


图 7-23. 外部触发脉冲输出模式的基本时序



当 TQ0CE 位被设置为 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 TOQ0b 引脚输出一个脉宽调制信号波形。若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 引脚输出反向。在出现触发时，不管当前状态如何（高/低），TOQ0b 引脚都会输出一个高电平。）

脉冲宽度调制信号波形的有效电平宽度、周期、占空比计算如下。

$$\text{有效电平宽度} = (\text{TQ0CCRb 寄存器的设定值}) \times \text{计数时钟周期}$$

周期 = (TQ0CCR0 寄存器的设定值 + 1) × 计数时钟周期

占空比系数 = (TQ0CCRB 寄存器的设定值) / (TQ0CCR0 寄存器的设定值 + 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向上计数时，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 CCRb 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTQ0CCb。

当 16 位计数器的计数值与 CCRa 缓冲寄存器的值匹配，而且 16 位计数器清零为 0000H 时，TQ0CCRa 寄存器设置的值就传送到 CCRa 缓冲寄存器。

只有软件触发 (TQ0CTL1.TQ0EST 位) 设为 1 被用作触发信号。

备注 a = 0 至 3
b = 1 至 3

图 7-24. 外部触发脉冲输出模式的寄存器设置 (1/3)

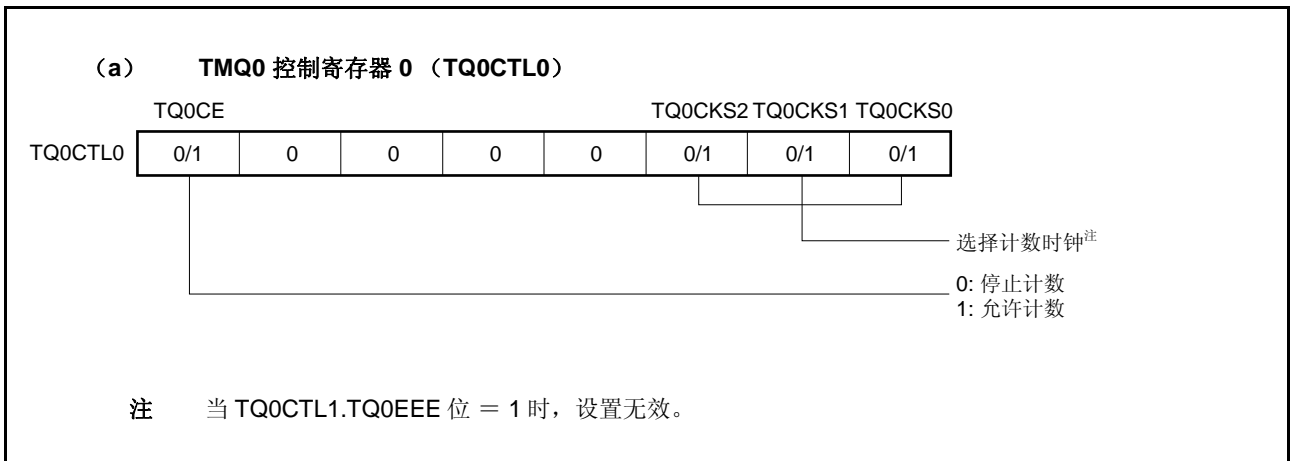


图 7-24. 外部触发脉冲输出模式的寄存器设置 (2/3)

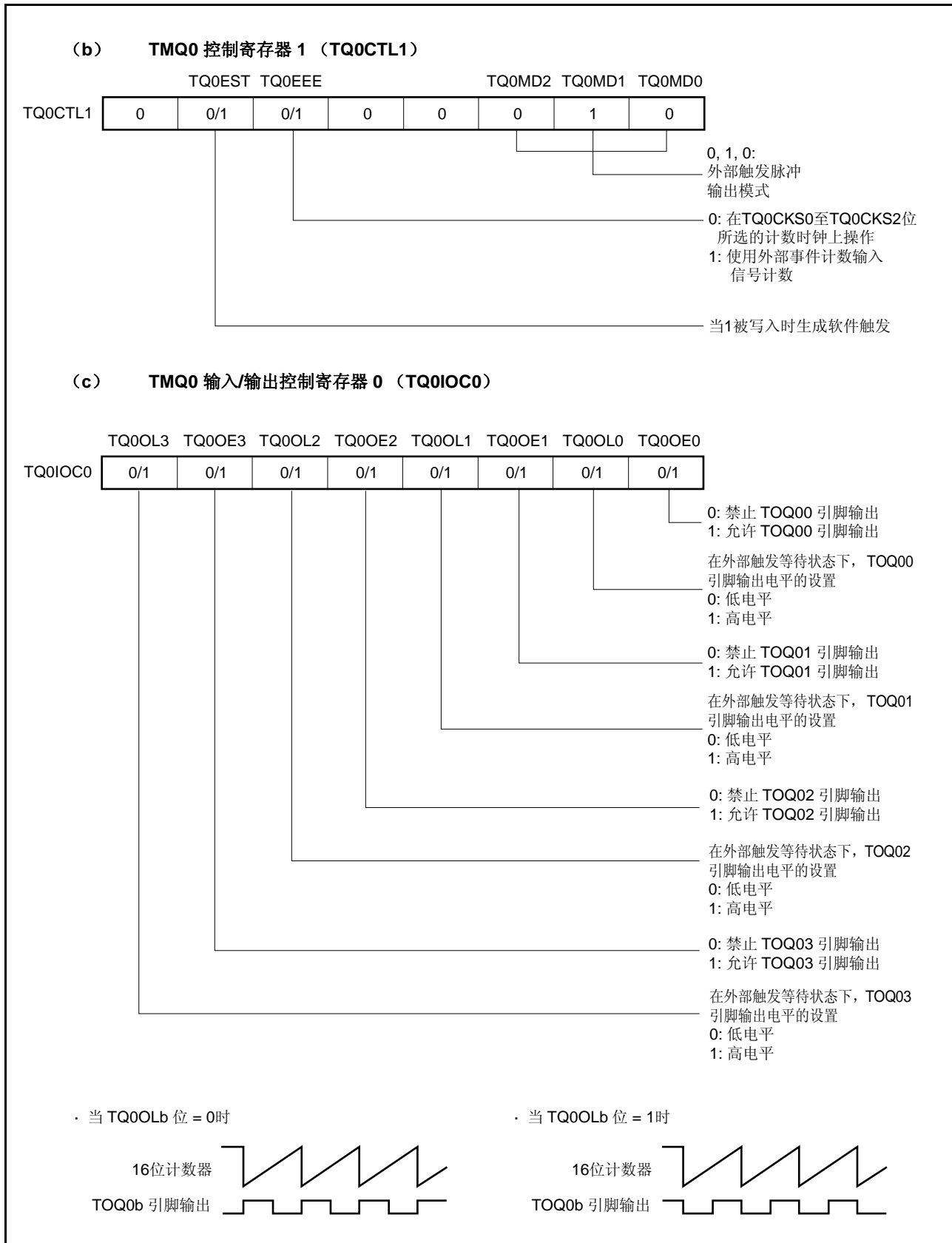
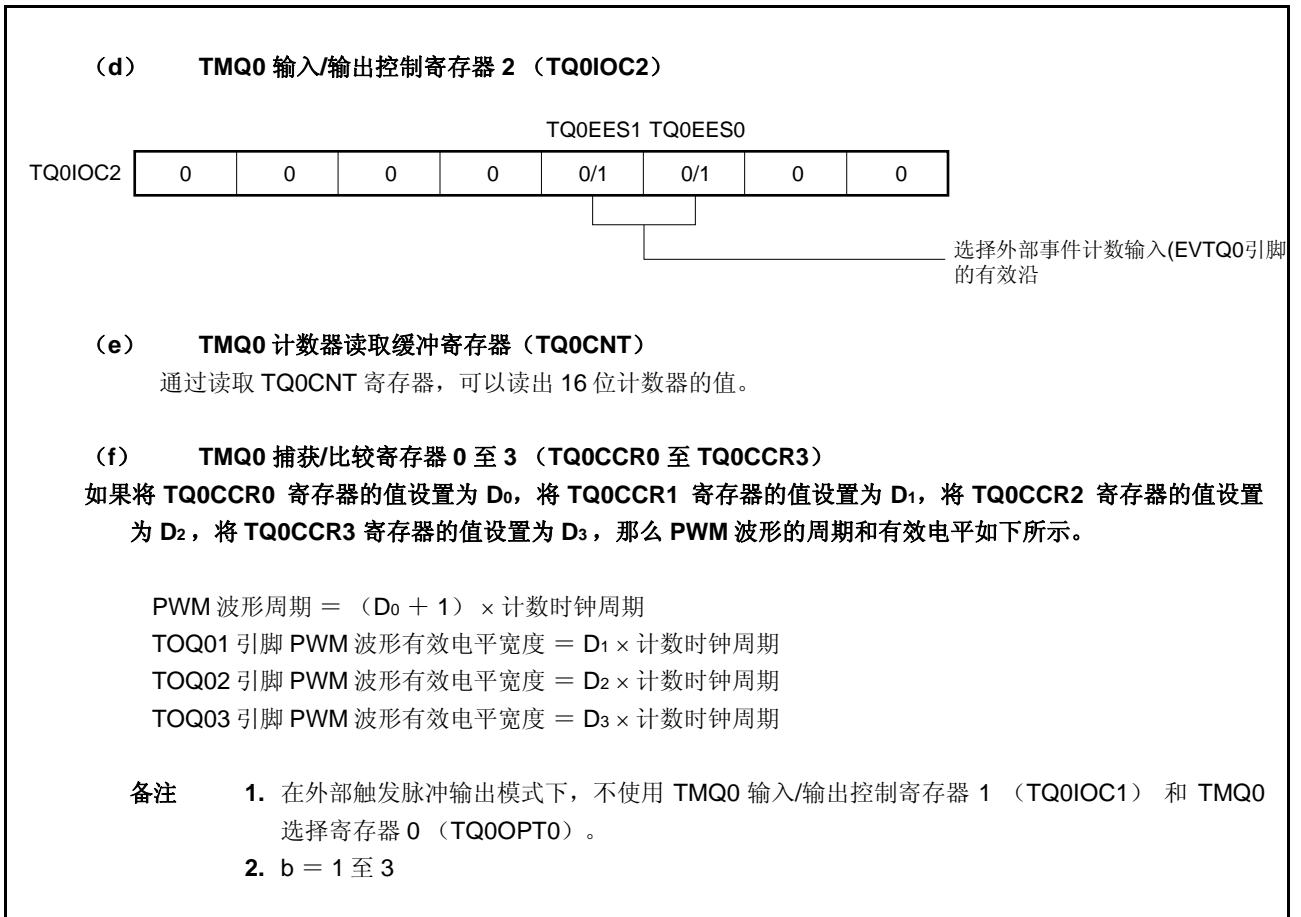


图 7—24. 外部触发脉冲输出模式的寄存器设置 (3/3)



(1) 外部触发脉冲输出模式的工作流程

图7-25. 外部触发脉冲输出模式的软件处理流程 (1/2)

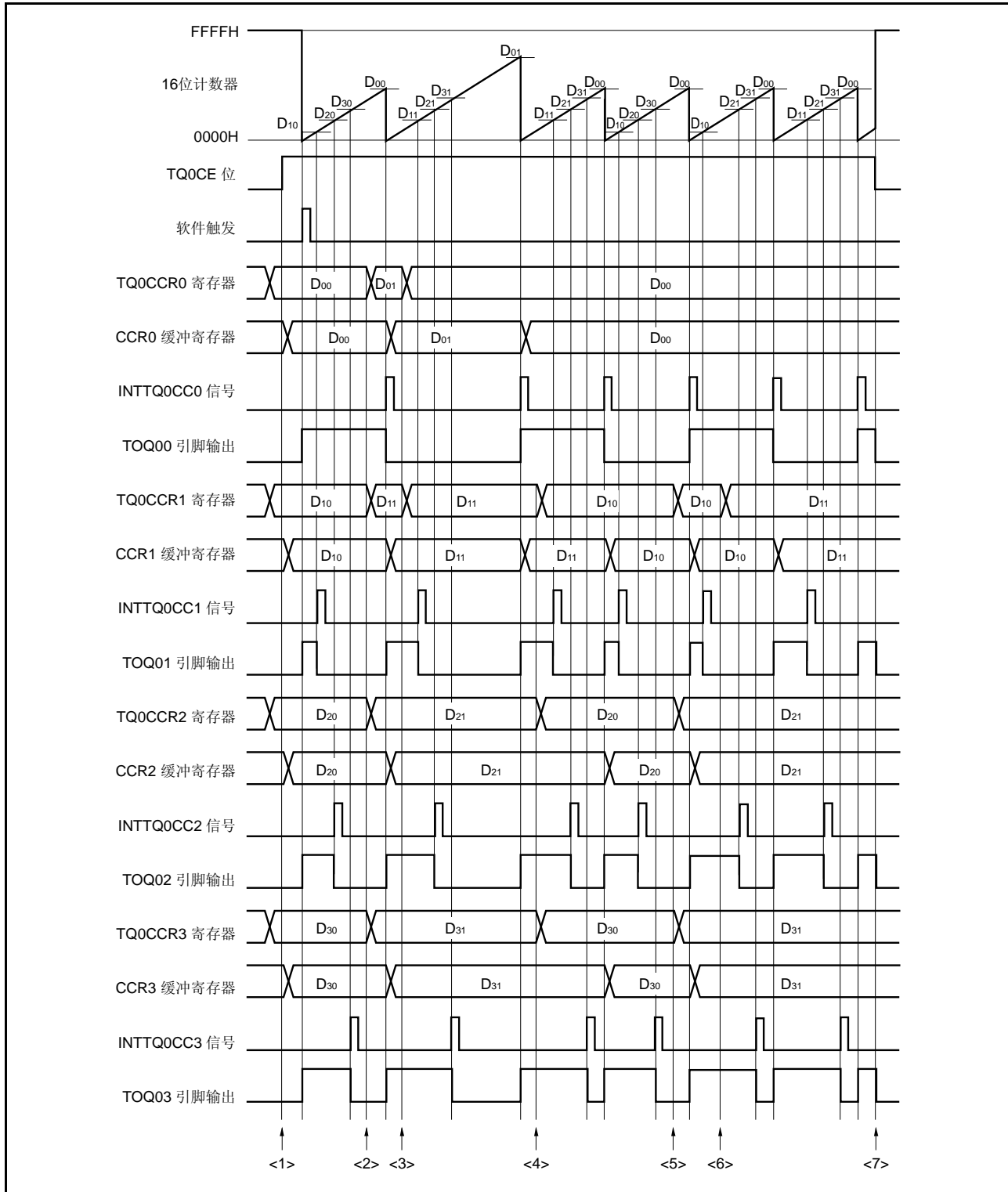
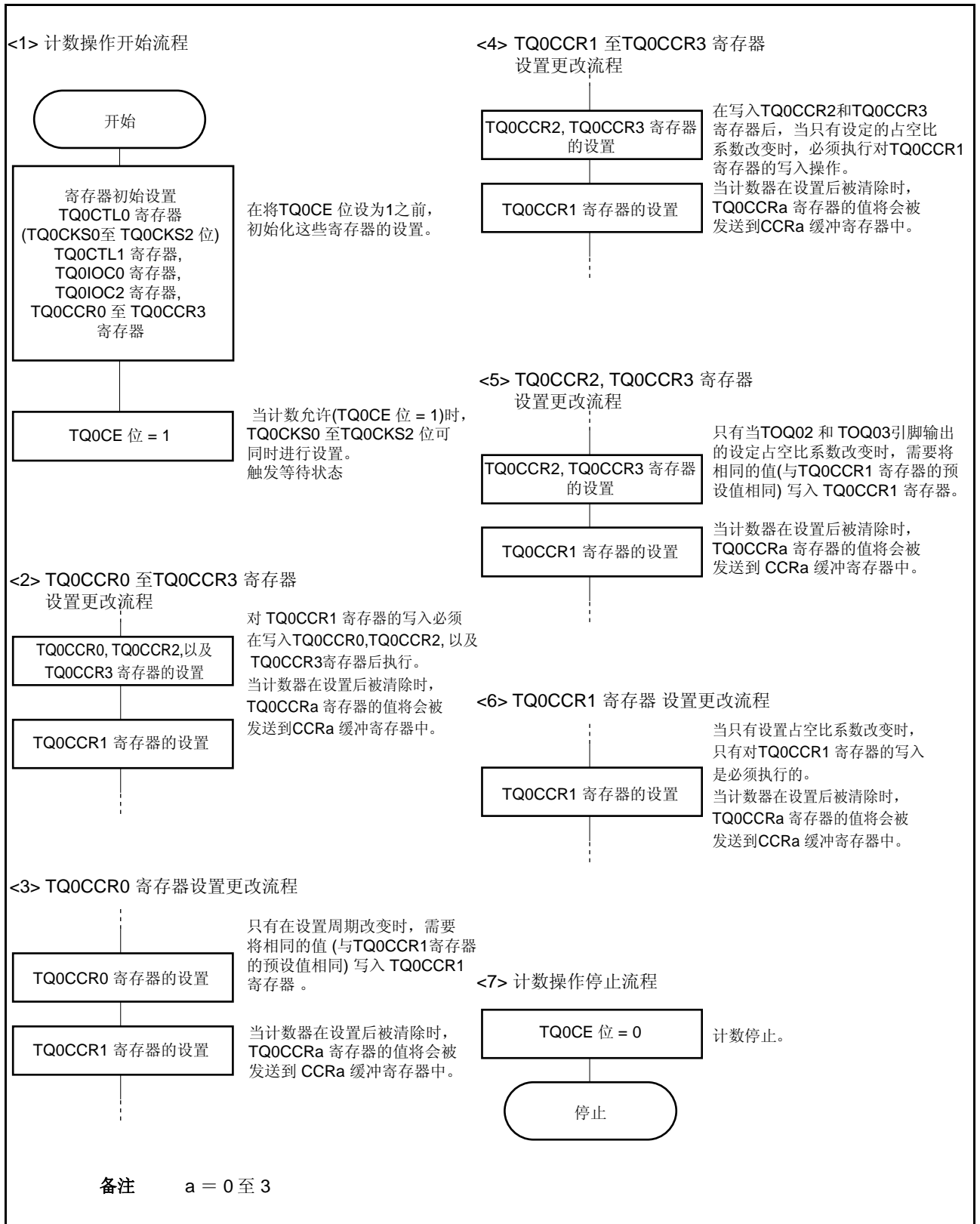


图 7-25. 外部触发脉冲输出模式的软件处理流程 (2/2)



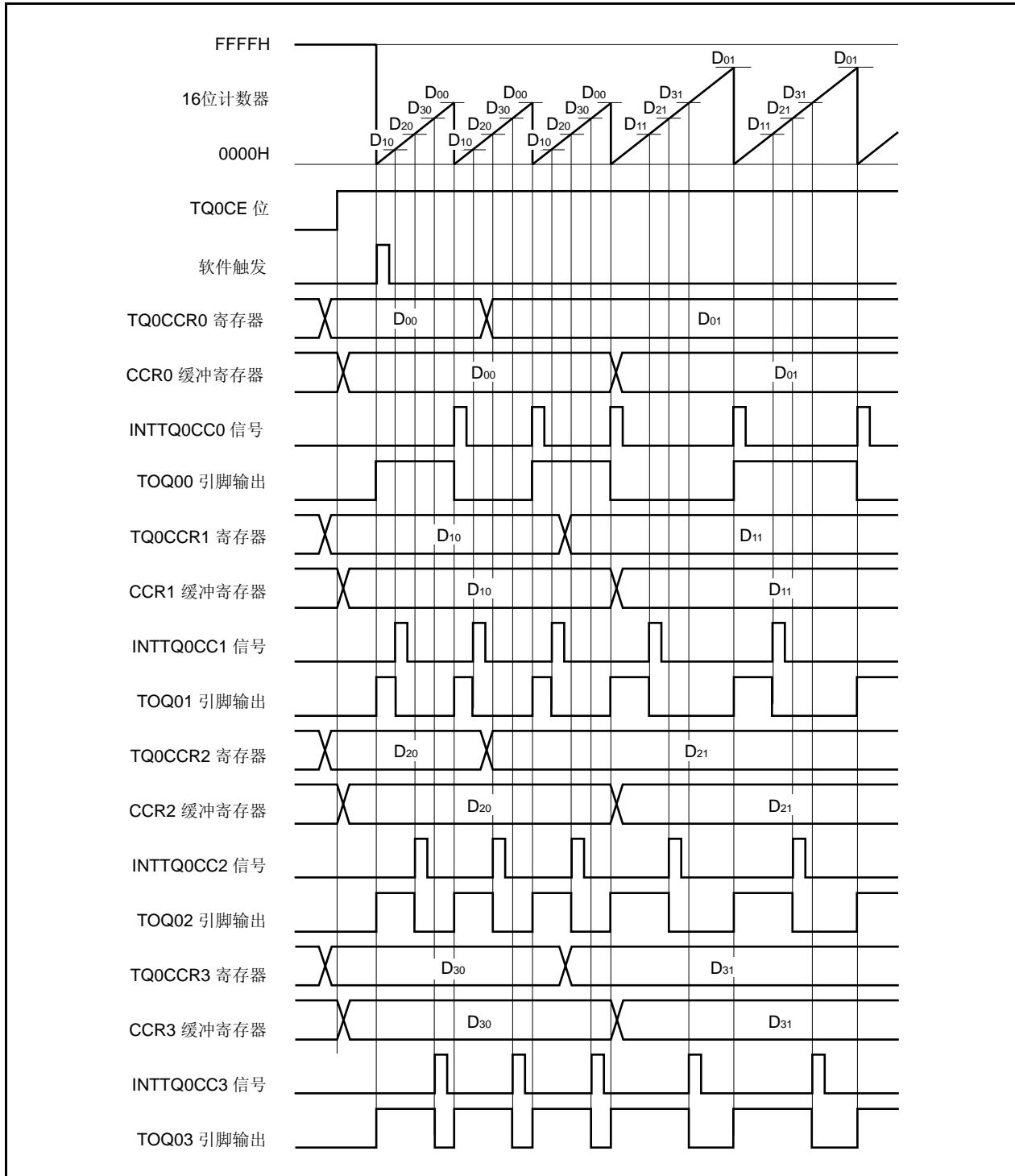
(2) 外部触发脉冲输出模式的操作时序

(a) 有关操作中改变脉冲宽度的注意事项

计数器正在工作时，要改变脉宽调制信号波形应最后写入 TQ0CCR1 寄存器。

检测到 INTTQ0CC0 信号以后，在写入 TQ0CCR1 寄存器之后重写 TQ0CCRb 寄存器。

备注 b = 1 至 3



为了将数据从 TQ0CCRa 寄存器传送到 CCRa 缓冲寄存器，必须写入 TQ0CCR1 寄存器。

要同时改变脉宽调制信号波形的周期和有效电平宽度，应首先将周期设定至 TQ0CCR0 寄存器，将有效电平宽度设定至 TQ0CCR2 和 TQ0CCR3 寄存器，然后将有效电平设定至 TQ0CCR1 寄存器。

如果只改变脉宽调制信号波形的周期，应首先把周期设置至 TQ0CCR0 寄存器，然后将相同值（与 TQ0CCR1 寄存器预设值相同的值）写入 TQ0CCR1 寄存器。

只改变脉宽调制信号波形的有效电平宽度（占空比系数）时，首先需要向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平值，然后再向 TQ0CCR1 寄存器设置有效电平。

若只改变通过 TOQ01 引脚输出的脉宽调制信号波形的有效电平宽度（占空比系数）时，只需要设置 TQ0CCR1 寄存器。

若只改变通过 TOQ02 和 TOQ03 引脚输出的脉宽调制信号波形的有效电平宽度（占空比系数）时，首先要向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平宽度，然后向 TQ0CCR1 寄存器写入相同的值（等于 TQ0CCR1 寄存器的预置值）。

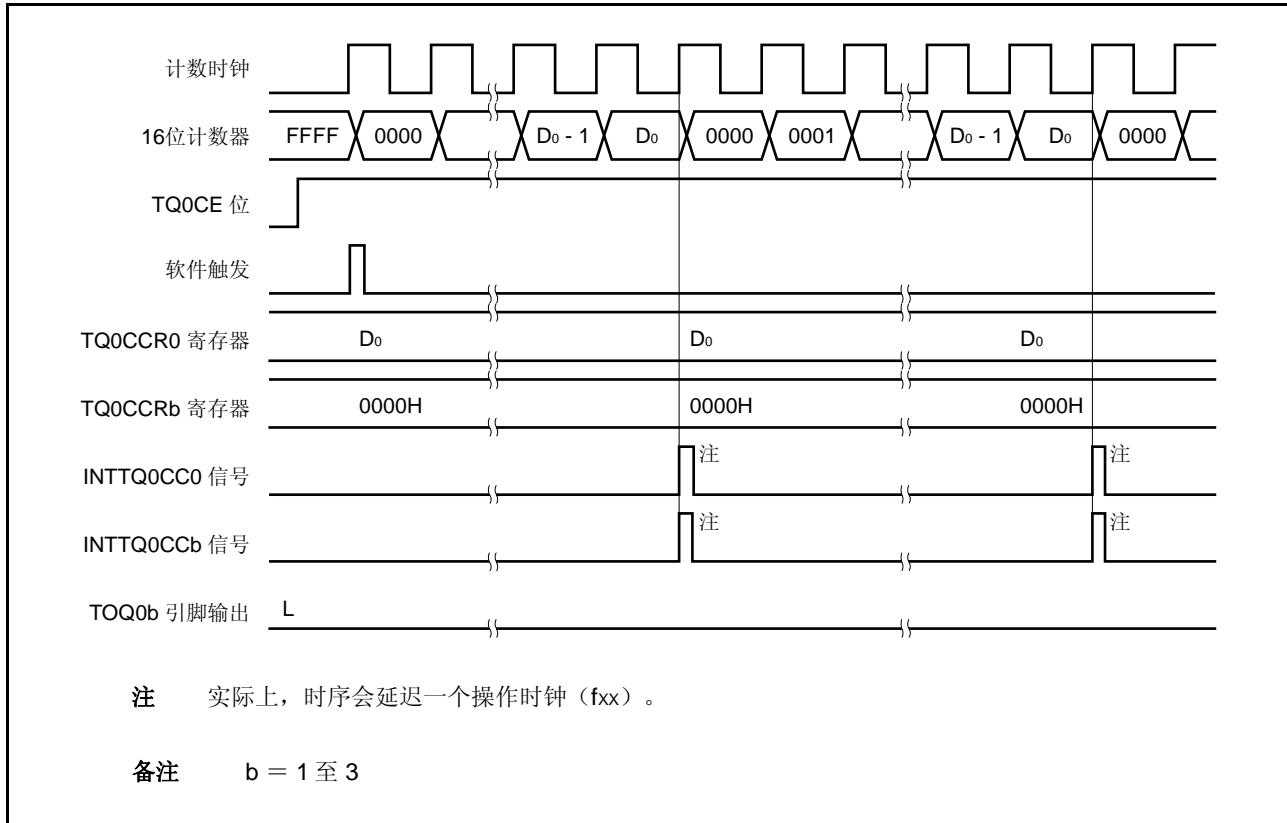
在数据写入 TQ0CCR1 寄存器后，写入 TQ0CCRa 寄存器的值会传送到 CCRa 缓冲寄存器，与 16 位计数器清零同步，并作为与 16 位计数器进行比较的值。

如果要在写入 TQ0CCR1 寄存器一次之后，再次写入 TQ0CCR0 至 TQ0CCR3 寄存器，应在 INTTQ0CC0 信号产生之后，否则，CCRa 缓冲寄存器的值可能不确定，因为把数据从 TQ0CCRa 寄存器传送到 CCRa 缓冲寄存器的时间与写入 TQ0CCRa 寄存器相冲突。

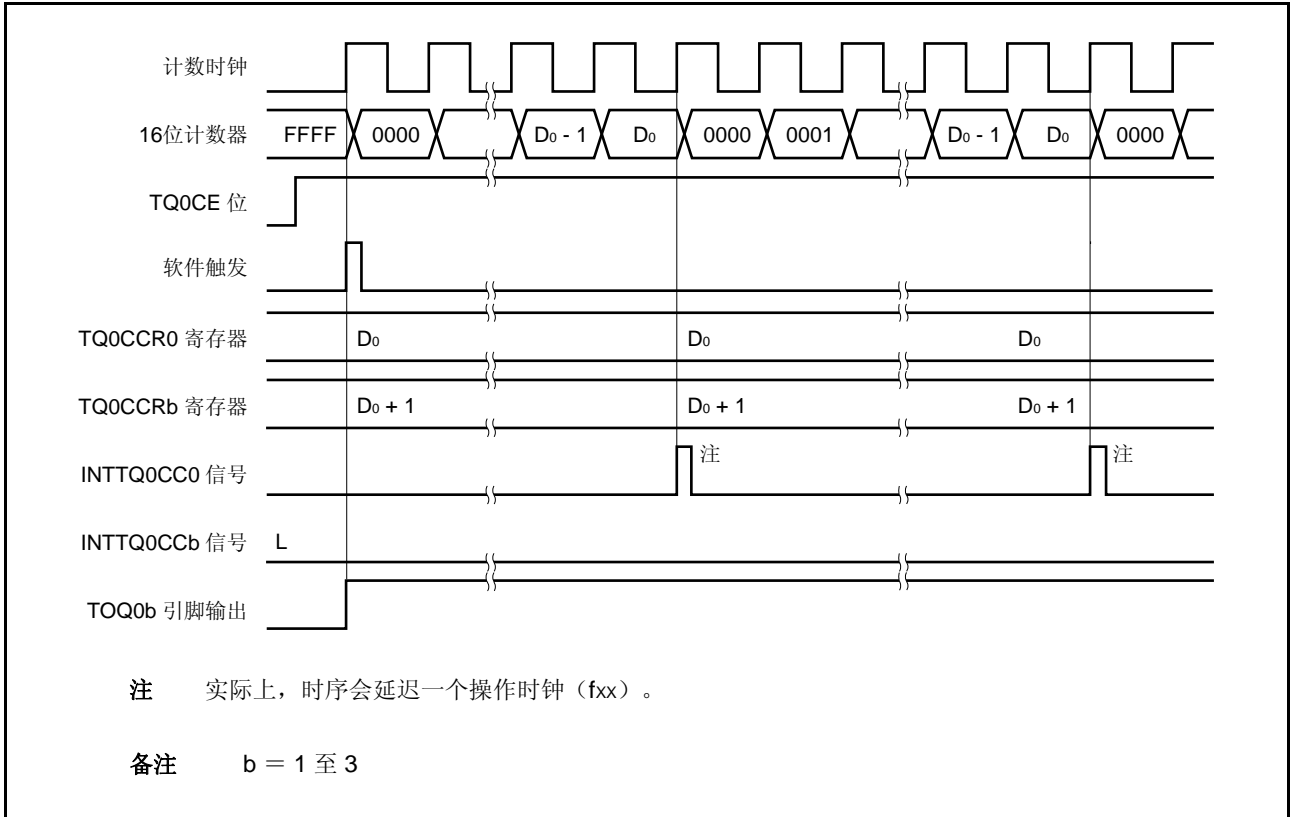
备注 a = 0 至 3

(b) 脉宽调制信号波形的 0% 和 100% 输出

若要输出 0% 的波形，则需将 TQ0CCRb 寄存器设置为 0000H。在下一次 16 位计数器的值与 CCR0 缓冲寄存器的值匹配时，16 位计数器将会被清除为 0000H 且会生成 INTTQ0CC0 和 INTTQ0CCb 信号。



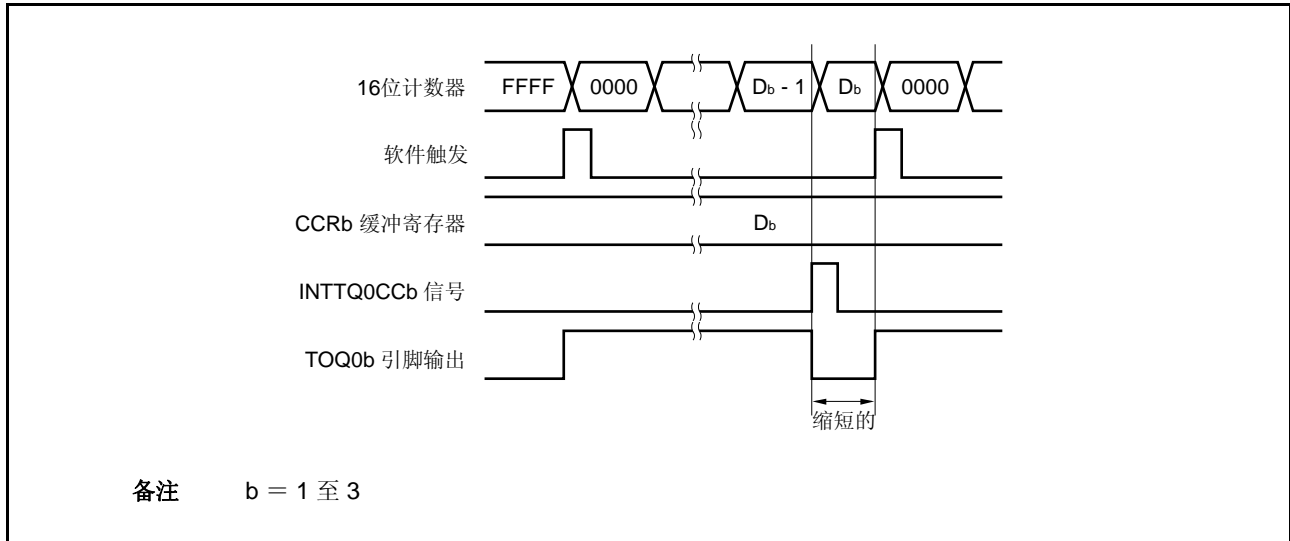
若要输出 100% 波形，则需将 TQ0CCRb 寄存器设置为 (TQ0CCR0 寄存器的设定值 + 1)。TQ0CCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。



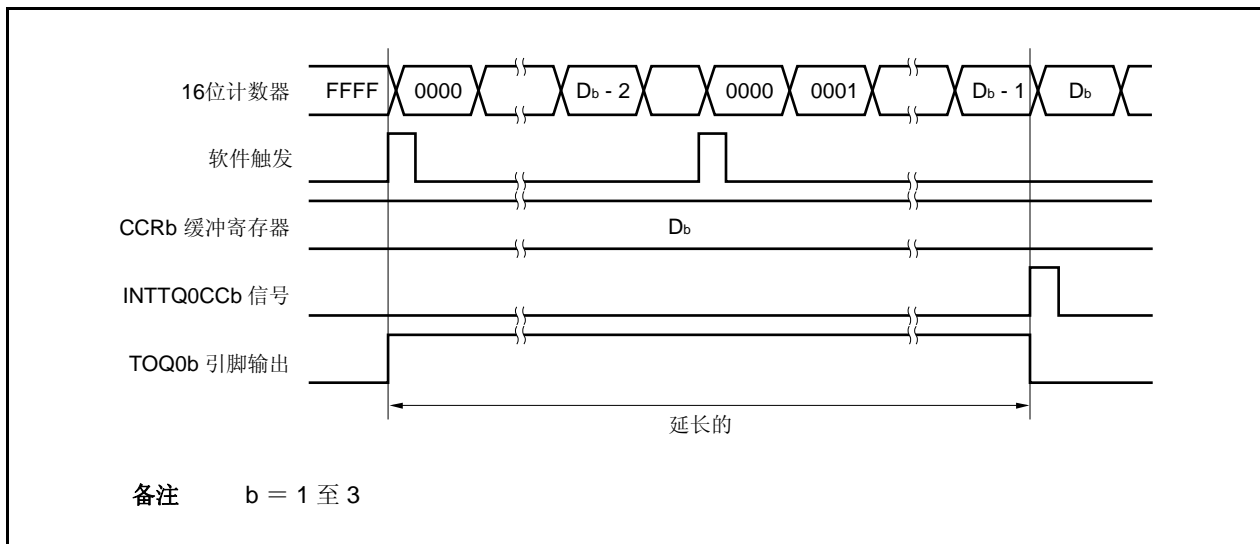
<R>

(c) 触发检测和 CCRb 缓冲寄存器匹配之间的冲突

如果在 INTTQ0CCb 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，TOQ0b 引脚的输出信号被置位生效，计数器继续计数，结果，缩短了脉宽调制信号波形的无效时间。

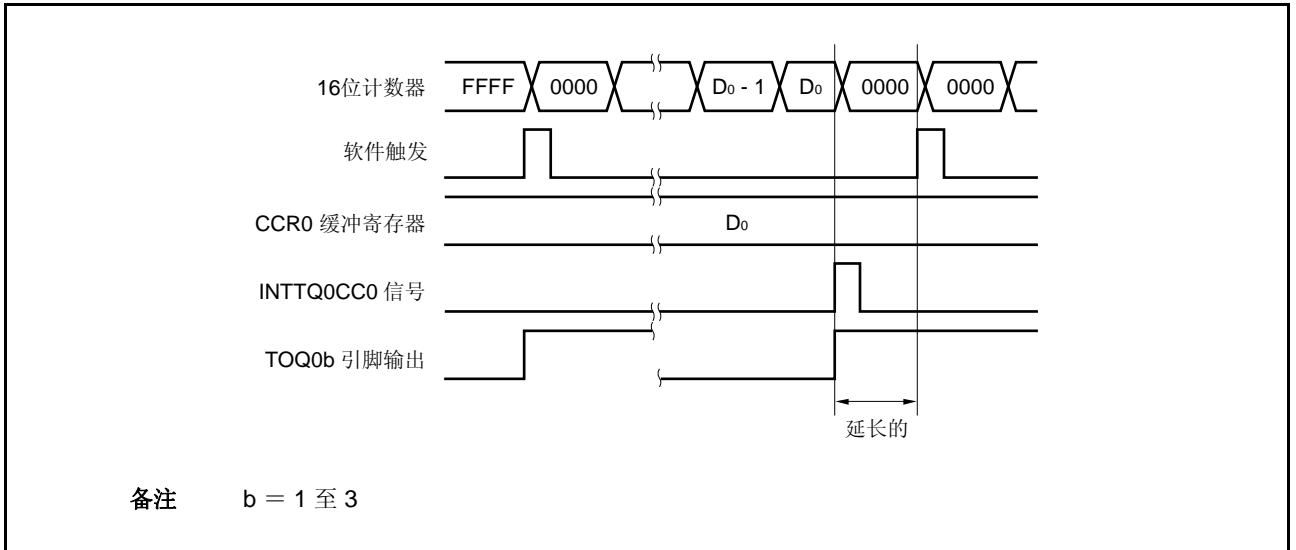


如果刚好在信号产生前检测到触发信号，则不会产生信号，而 16 位计数器清零为 0000H，计数继续进行。TOQ0b 引脚的输出信号仍然保持有效。结果，延长了脉宽调制信号波形的有效时间。

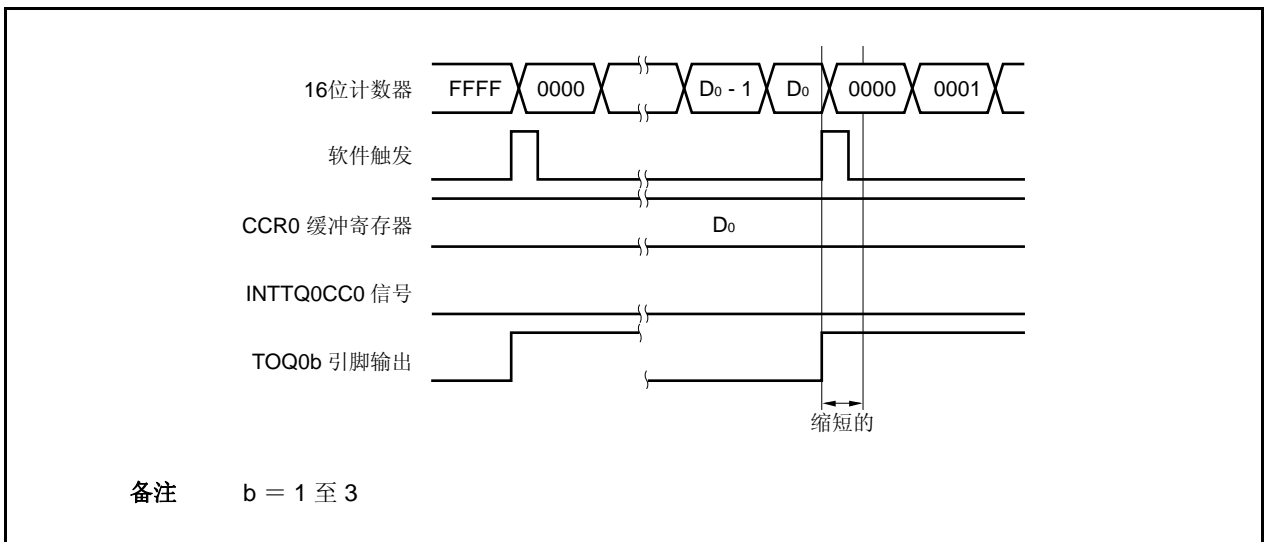


(d) 触发检测和 CCR0 缓冲寄存器匹配之间的冲突

如果刚好在 INTTQ0CC0 信号产生后检测到触发，则 16 位计数器清零为 0000H，并继续计数。因此，TOQ0b 引脚的有效时间被延长，延长时间为自 INTTQ0CC0 信号产生时刻到触发被检测到时刻。

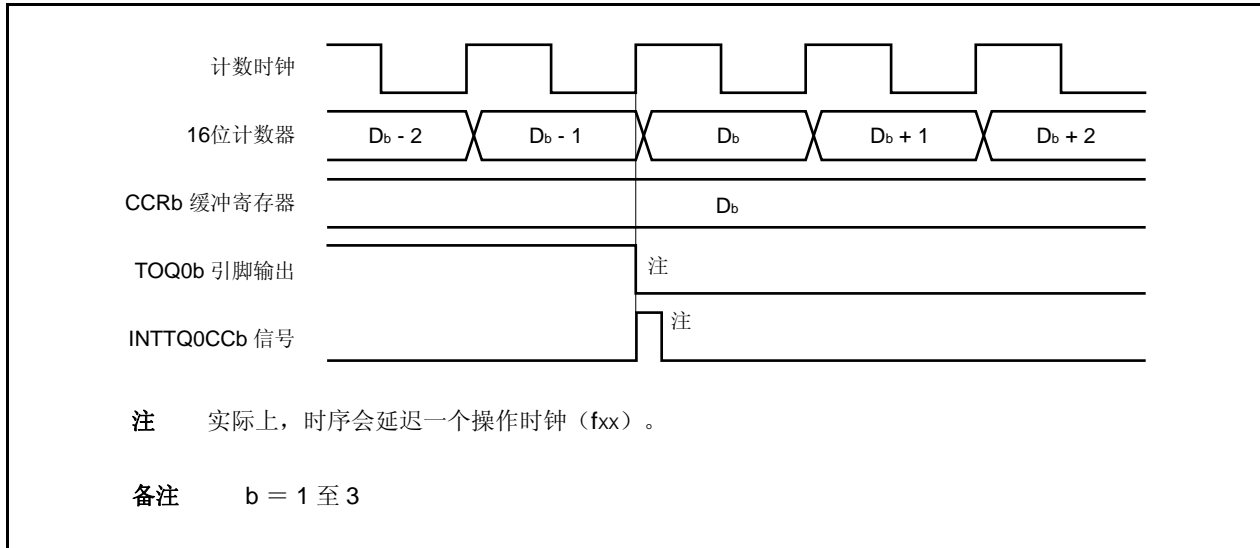


如果在紧接 INTTQ0CC0 信号产生之前检测到触发信号，那么将不会产生 INTTQ0CC0 信号，16 位计数器被清除为 0000H，TOQ0b 引脚继续保持为有效电平，计数器重新开始计数。结果，缩短了脉宽调制信号波形的无效时间。



(e) 比较匹配中断请求信号 (INTTQ0CCb) 的发生时序

INTTQ0CCb 信号在外部触发脉冲输出模式下的发生时序与其它 INTTQ0CCb 信号的时序不同；当 16 位计数器的计数值与 CCRb 缓冲寄存器的值匹配时，产生 INTTQ0CCb 信号。



通常情况下，在 16 位计数器的当前计数值与 CCRb 缓冲寄存器的值匹配之后，INTTQ0CCb 信号与下一次计数同步发生。

但是，在外部触发脉冲输出模式下，其产生会提早一个计数时钟周期。这是由于为了匹配 TOQ0b 引脚输出信号的变化，时序发生了变化。

7.6.4 单次脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 011)

该模式仅在 TMQ0 中有效。

在单脉冲输出模式下，当 TQ0CTL0.TQ0CE 位被设置为 1 时，16 位定时器/事件计数器 Q 等待触发。当产生软件触发时，16 位定时器/事件计数器 Q 开始计数，并且从 TOQ01 至 TOQ03 引脚输出一个脉宽调制信号波形。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 7-26. 单次脉冲输出模式的配置

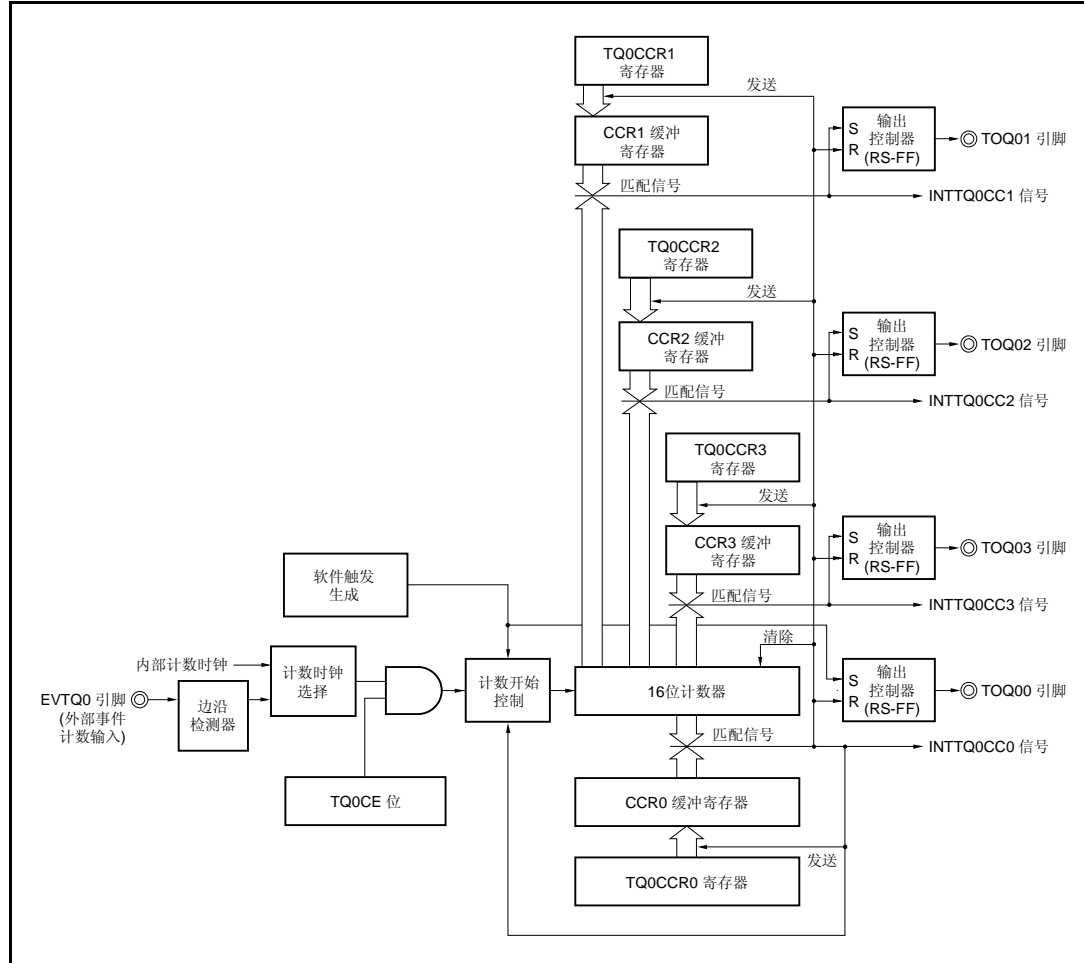
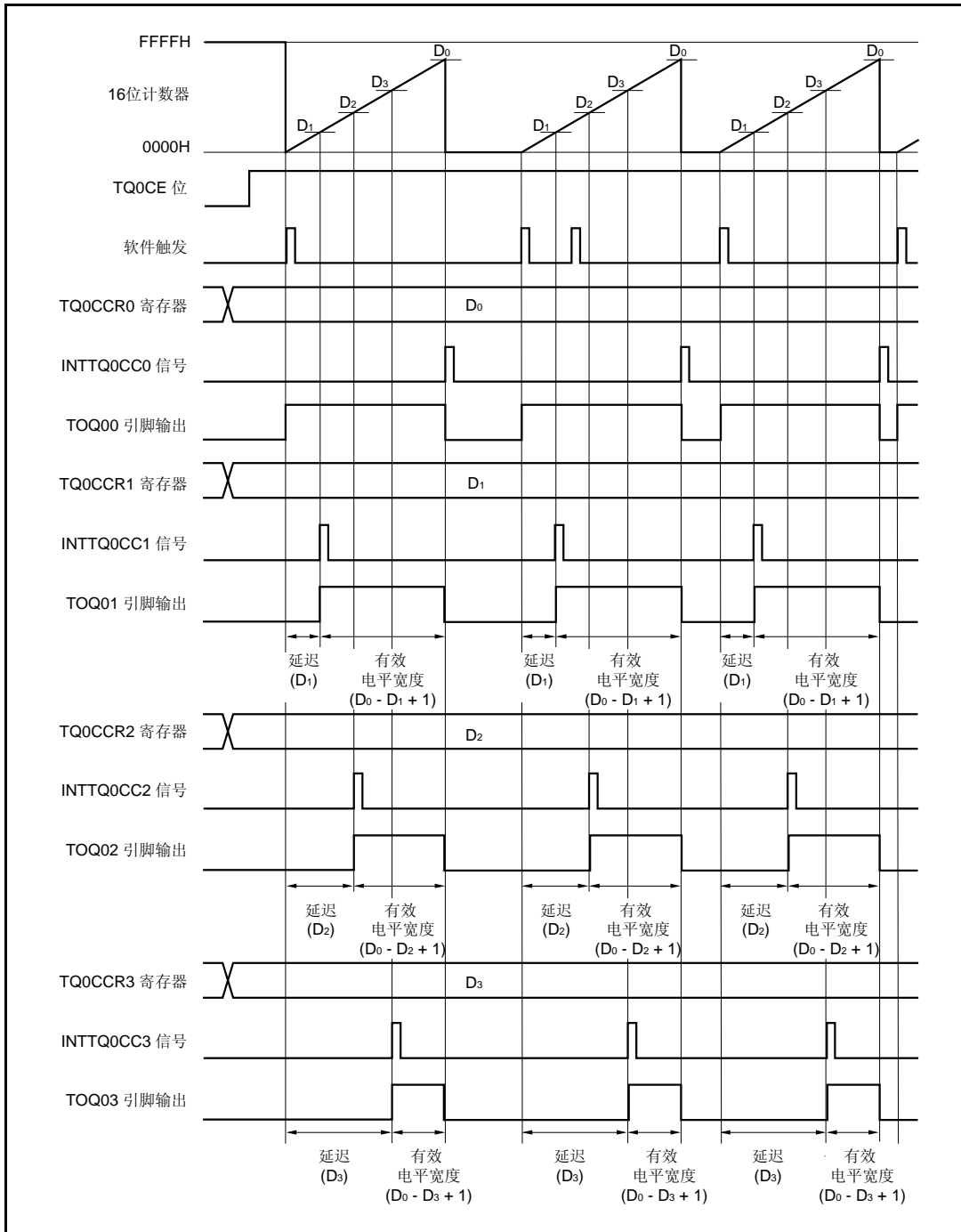


图 7-27. 单次脉冲输出模式的基本时序



当 TQ0CE 位被设置为 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 TQ0b 引脚输出一个单脉冲。在输出单脉冲后，16 位计数器被清除为 0000H 时，计数停止，并重新等待触发。当再次产生触发时，16 位计数器从 0000H 开始计数。如果 TOP01 引脚正在输出单脉冲期间再次发生触发，则该触发被忽略。

单脉冲的输出延迟时间和有效电平宽度可计算如下：

$$\text{输出延迟时间} = (\text{TQ0CCRb 寄存器的设定值}) \times \text{计数时钟周期}$$

$$\text{有效电平宽度} = (\text{TQ0CCR0 寄存器的设定值} - \text{TQ0CCRb 寄存器的设定值} + 1) \times \text{计数时钟周期}$$

当 16 位计数器在其计数值与 CCR0 缓冲寄存器的值匹配之后进行下一个计数时，产生 INTTQ0CC0 比较匹配中断请求信号。当 16 位计数器的计数值与 CCRb 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTQ0CCb。

只有软件触发 (TQ0CTL1.TQ0EST 位) 设为 1 被用作触发信号。

备注 b = 1 至 3

图 7-28. 单脉冲输出模式时的寄存器设置 (1/3)

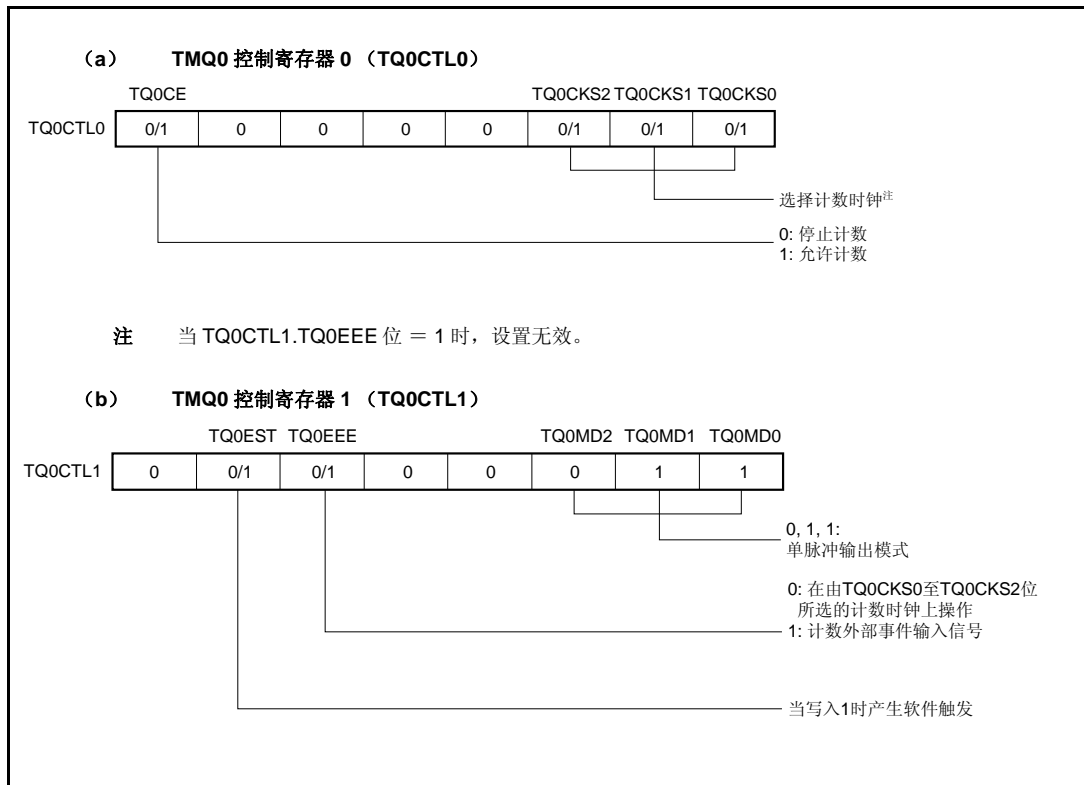


图 7-28. 单次脉冲输出模式时的寄存器设置 (2/3)

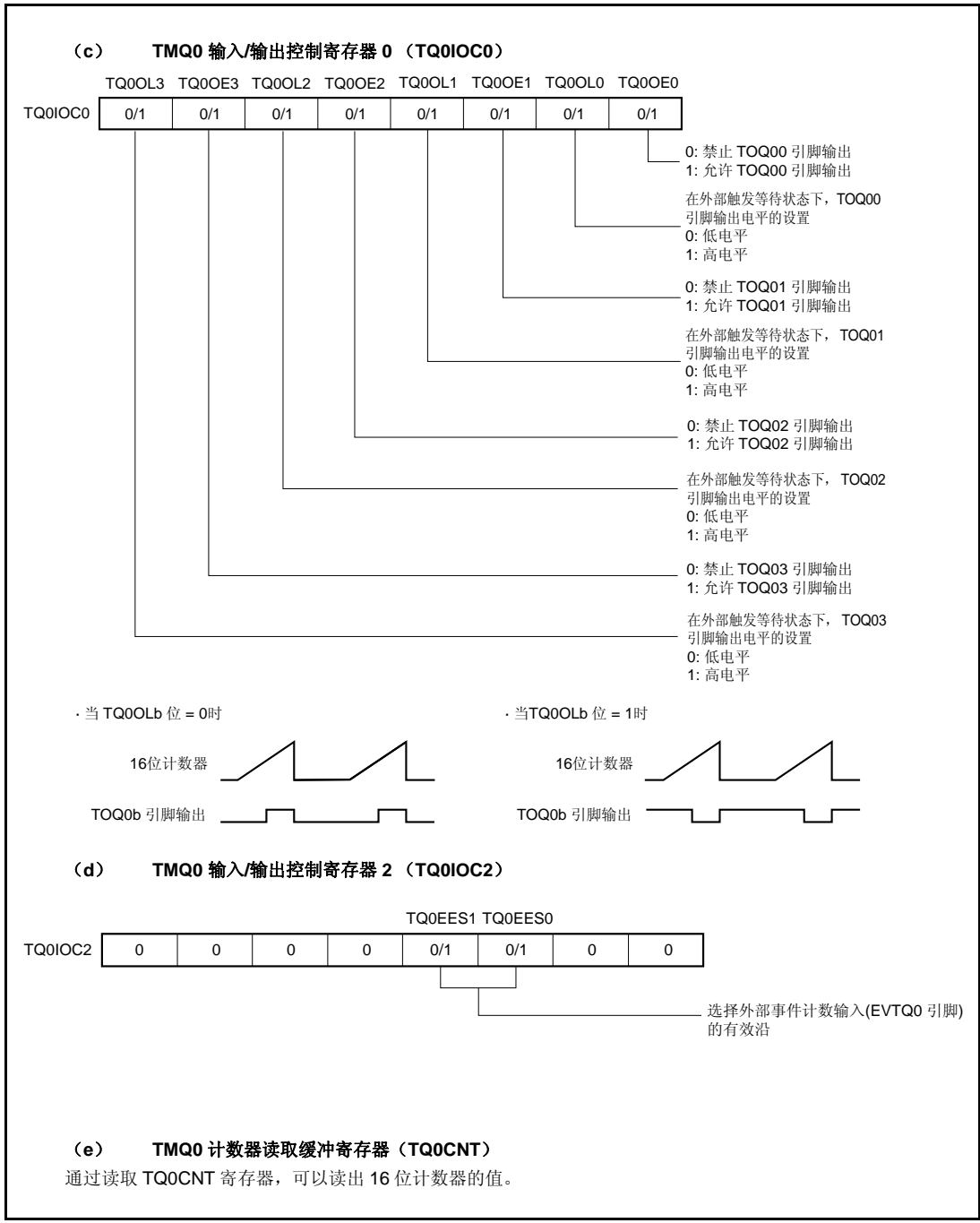


图 7-28. 单次脉冲输出模式时的寄存器设置 (3/3)

(f) TMQ0 捕获/比较寄存器 0 至 3 (TQ0CCR0 至 TQ0CCR3)

如果将 TQ0CCR0 寄存器的值设置为 D_0 ，将 TQ0CCRB 寄存器的值设置为 D_b ，则单次脉冲的有效电平宽度和输出延迟时间计算如下。

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

输出延迟时间 = $D_b \times$ 计数时钟周期

注意事项 如果 TQ0CCRB 寄存器中设置的值大于 TQ0CCR0 寄存器中设置的值，那么，即使在单次脉冲输出模式下也不会输出单次脉冲。

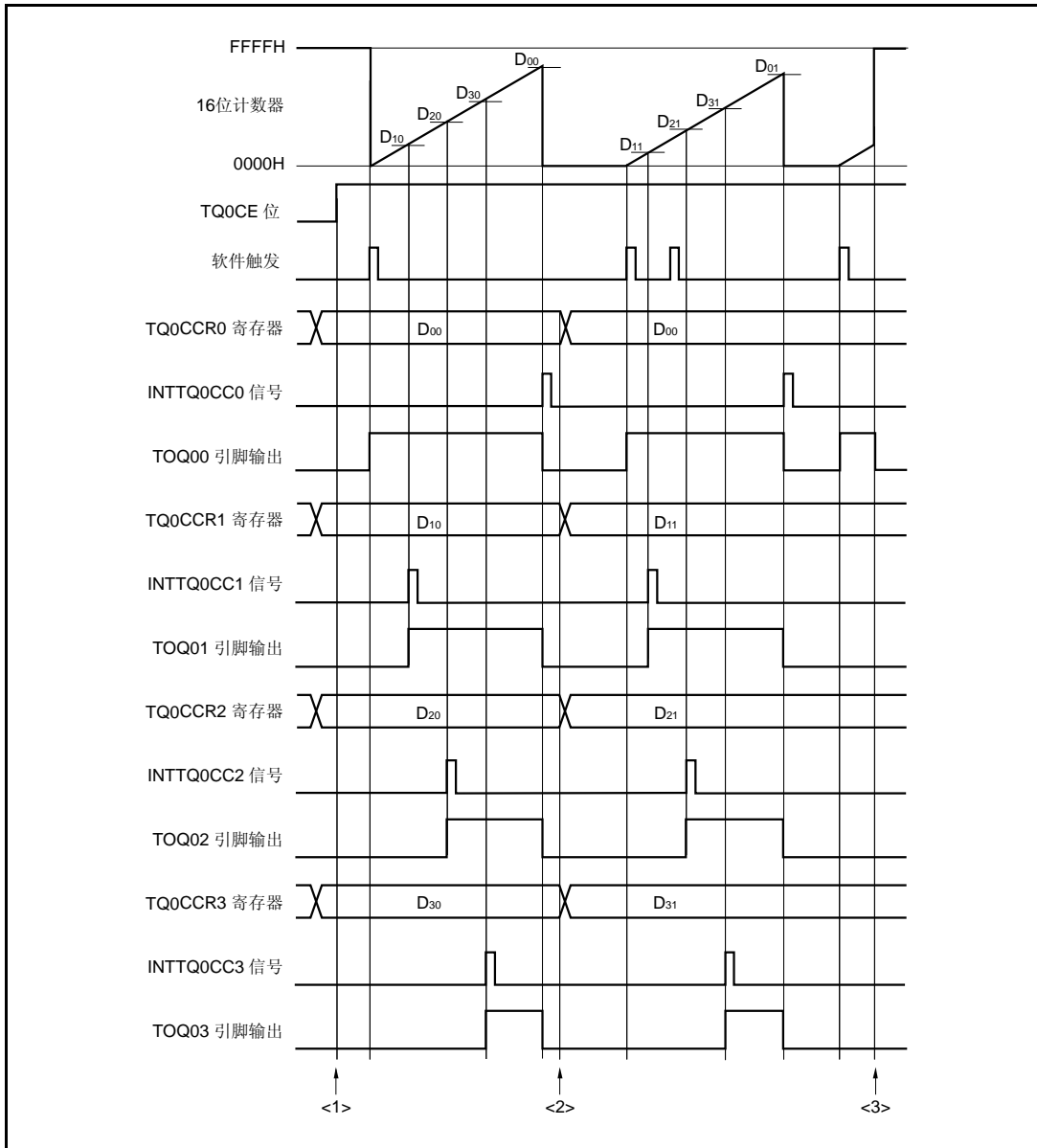
备注

1. 在单次脉冲输出模式下，不使用 TMQ0 输入/输出控制寄存器 1 (TQ0IOC1) 和 TMQ0 选择寄存器 0 (TQ0OPT0)。
2. $b = 1$ 至 3

(1) 单次脉冲输出模式下的操作流程

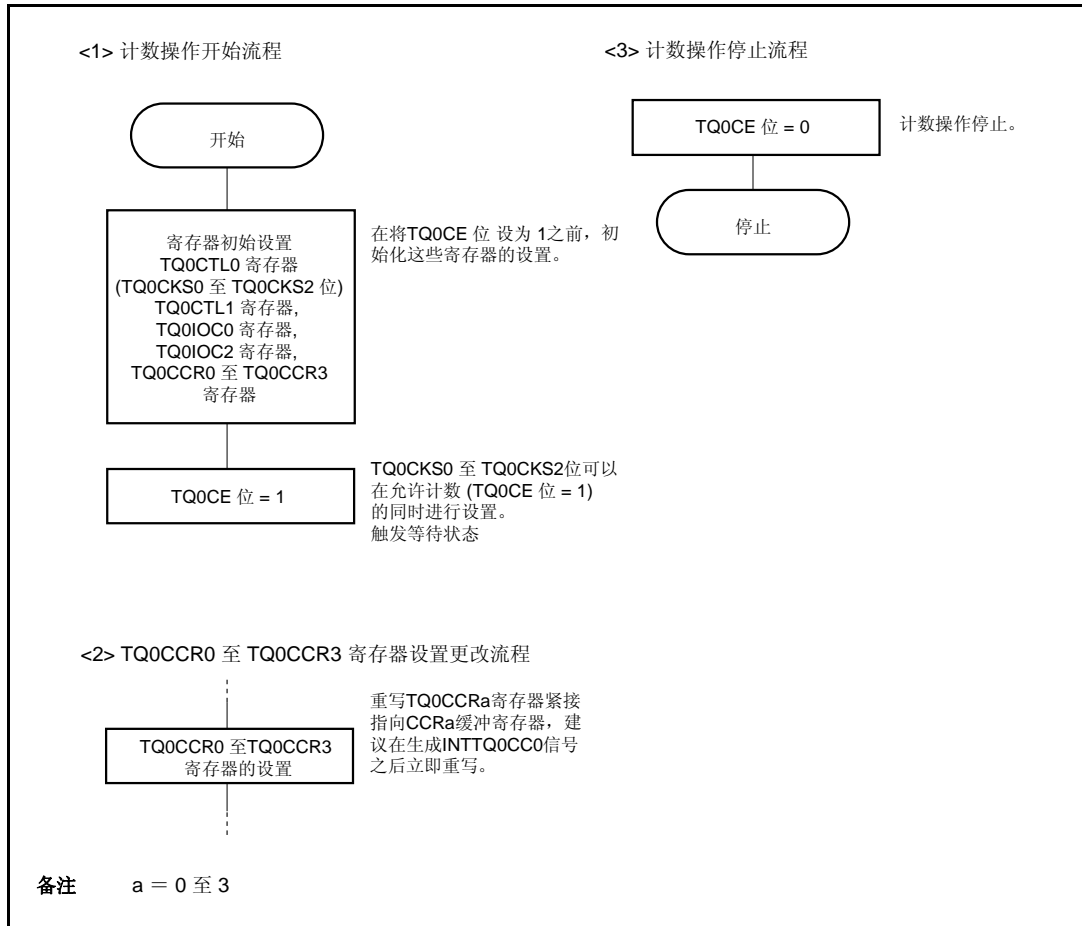
<R>

图 7-29. 单次脉冲输出模式的软件处理流程 (1/2)



<R>

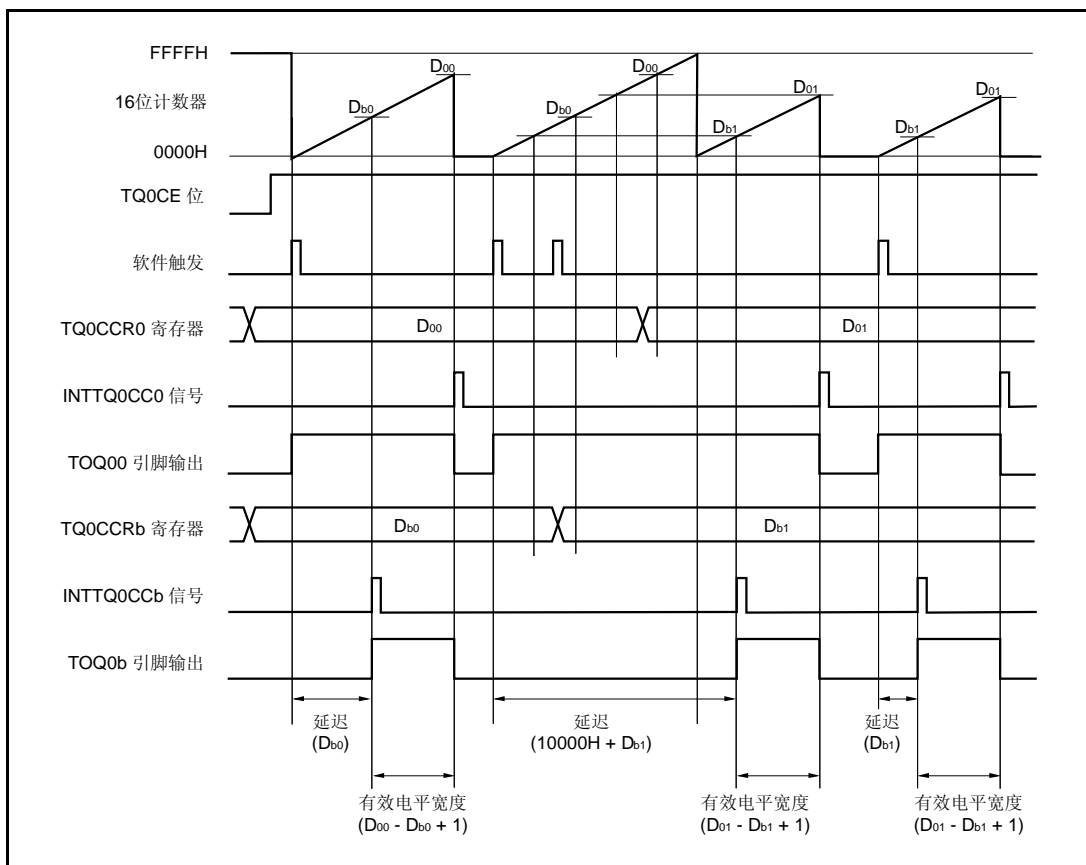
图 7-29. 单次脉冲输出模式的软件处理流程 (2/2)



(2) 单次脉冲输出模式时的工作时序

(a) 重写 TQ0CCRa 寄存器的注意事项

如果在计数器工作期间，将 TQ0CCRa 寄存器的值重写为比当前值较小的值，16 位计数器会发生溢出。发生溢出时，停止一次计数并更改设定值。



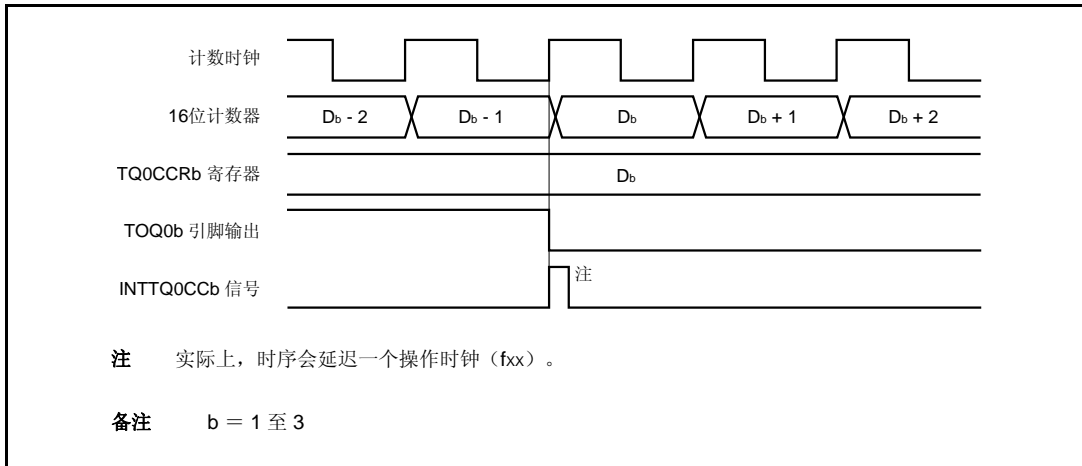
如果 $D_{00} > D_{01}$ 而且 $D_{b0} > D_{b1}$ ，当 TQ0CCR0 寄存器从 D_{00} 重写为 D_{01} ，而且 TQ0CCRb 寄存器从 D_{b0} 重写为 D_{b1} 时，若 TQ0CCRb 寄存器在 16 位计数器的计数值大于 D_{b1} 且小于 D_{b0} ，TQ0CCR0 寄存器在计数值大于 D_{01} 且小于 D_{00} 时重写，则一旦重写寄存器，各设置值立即与计数值进行比较。计数器向上计数到 FFFFH 后，从 0000H 开始重新向上计数。当计数值与 D_{b1} 匹配时，计数器会产生 INTTQ0CCb 信号并使 TOQ0b 引脚置位有效。当计数值与 D_{01} 匹配时，计数器产生 INTTQ0CC0 信号，使 TOQ0b 引脚清除置位，并停止计数。

因此，计数器输出脉冲的延迟期或有效期可能会与原先预计的单次脉冲有所不同。

备注 a = 0 至 3, b = 1 至 3

(b) 比较匹配中断请求信号 (INTTQ0CCb) 的发生时序

单次脉冲输出模式下，比较匹配中断请求信号 INTTQ0CCb 的发生时序和其它情况下的 INTTQ0CCb 信号不同，当 16 位计数器的计数值与 TQ0CCRb 寄存器的值匹配时，就会产生 INTTQ0CCb 中断请求信号。



通常情况下，在 16 位计数器的当前计数值与 TQ0CCRb 寄存器的值匹配之后，INTTQ0CCb 信号与下一次计数同步发生。

但是，在单次脉冲输出模式下，INTTQ0CCb 信号会提早一个计数时钟周期产生。这是由于为了匹配 TOQ0b 引脚输出信号的变化，时序发生了变化。

7.6.5 PWM输出模式 (TQ0MD2 至 TQ0MD0 位 = 100)

该模式仅在 TMQ0 中有效。

在脉宽调制信号输出模式下，当 TQ0CTL0.TQ0CE 位被设定为 1 时，脉宽调制信号波形从 TOQ01 至 TOQ03 (TOQH01 至 TOQH03) 引脚输出。

此外，带有 50%占空比的脉宽调制信号波形也可以从 TOQ00 引脚中输出，它的半周期等于 TQ0CCR0 寄存器的设定值 + 1。

图 7-30. 脉宽调制信号输出模式的配置

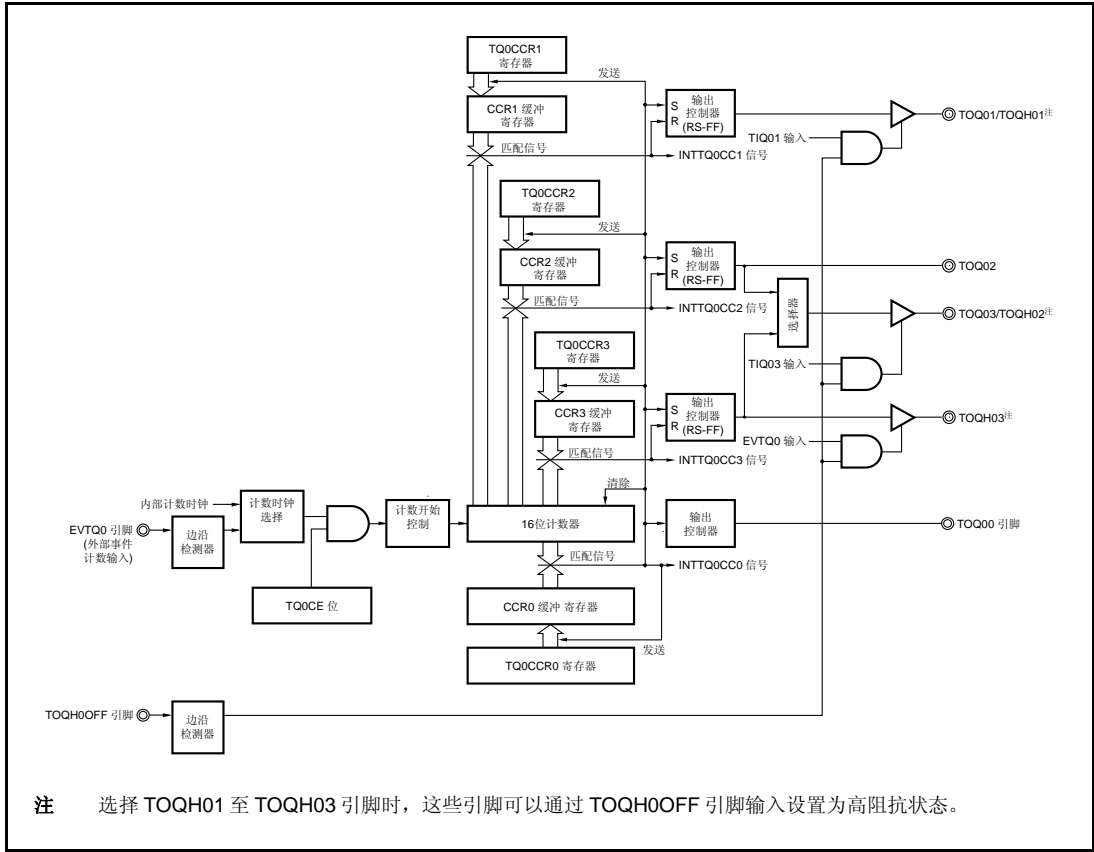
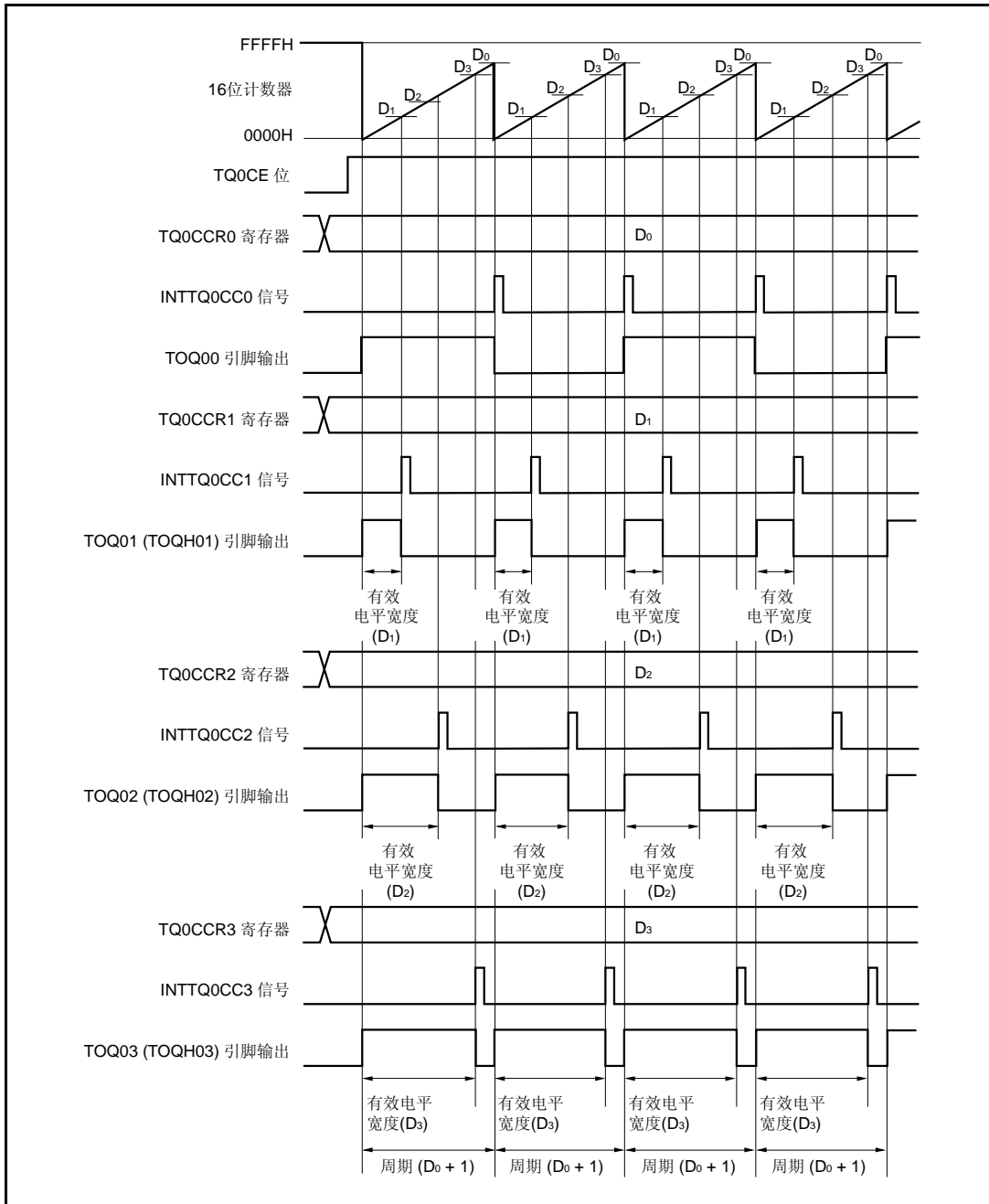


图 7-31. 脉宽调制信号输出模式的基本时序



当 TQ0CE 位被设置为 1 时，16 位计数器从 FFFFH 清零为 0000H，开始进行计数，同时从 TOQ0b 引脚 (TOQH0b) 输出脉冲宽度调制信号波形。

脉冲宽度调制信号波形的有效电平宽度、周期、占空比计算如下。

$$\begin{aligned} \text{有效电平宽度} &= (\text{TQ0CCRb 寄存器的设定值}) \times \text{计数时钟周期} \\ \text{周期} &= (\text{TQ0CCR0 寄存器的设定值} + 1) \times \text{计数时钟周期} \\ \text{占空比系数} &= (\text{TQ0CCRb 寄存器的设定值}) / (\text{TQ0CCR0 寄存器的设定值} + 1) \end{aligned}$$

在进行计数时，可通过重写 TQ0CCRa 寄存器来改变脉冲宽度调制信号波形。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，新写入的值会得到反映。

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向上计数时，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 CCRb 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTQ0CCb。

备注 a = 0 至 3
b = 1 至 3

图 7-32. 脉宽调制信号输出模式时的寄存器设置 (1/3)

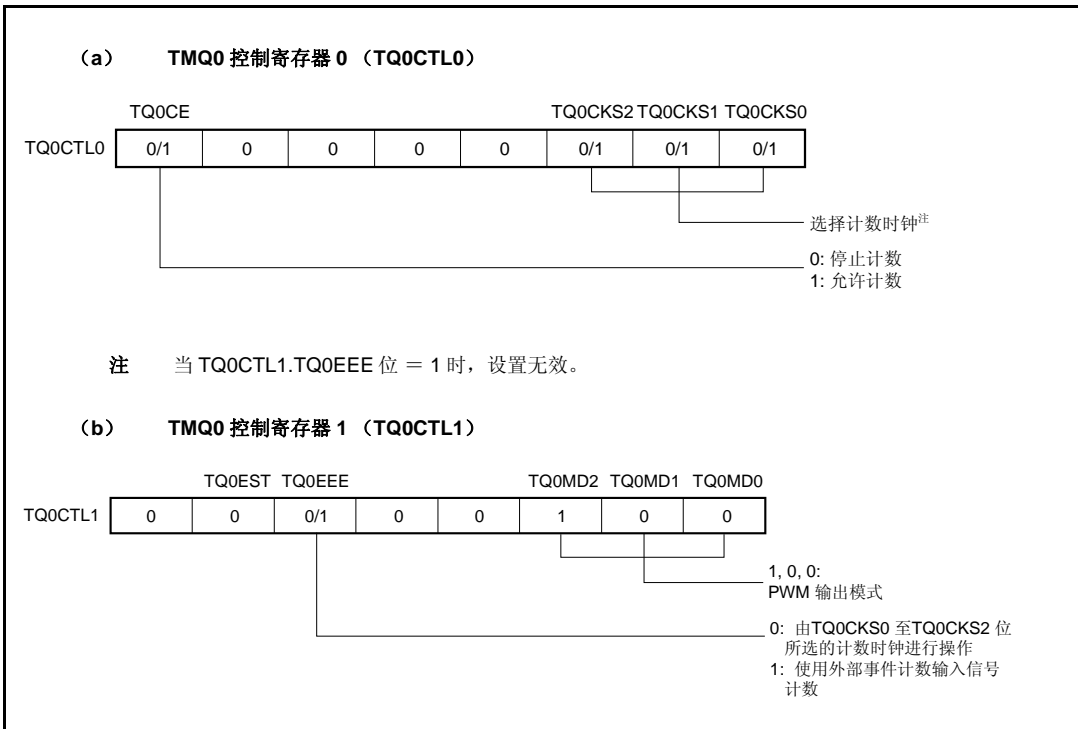


图 7-32. 脉宽调制信号输出模式时的寄存器设置 (2/3)

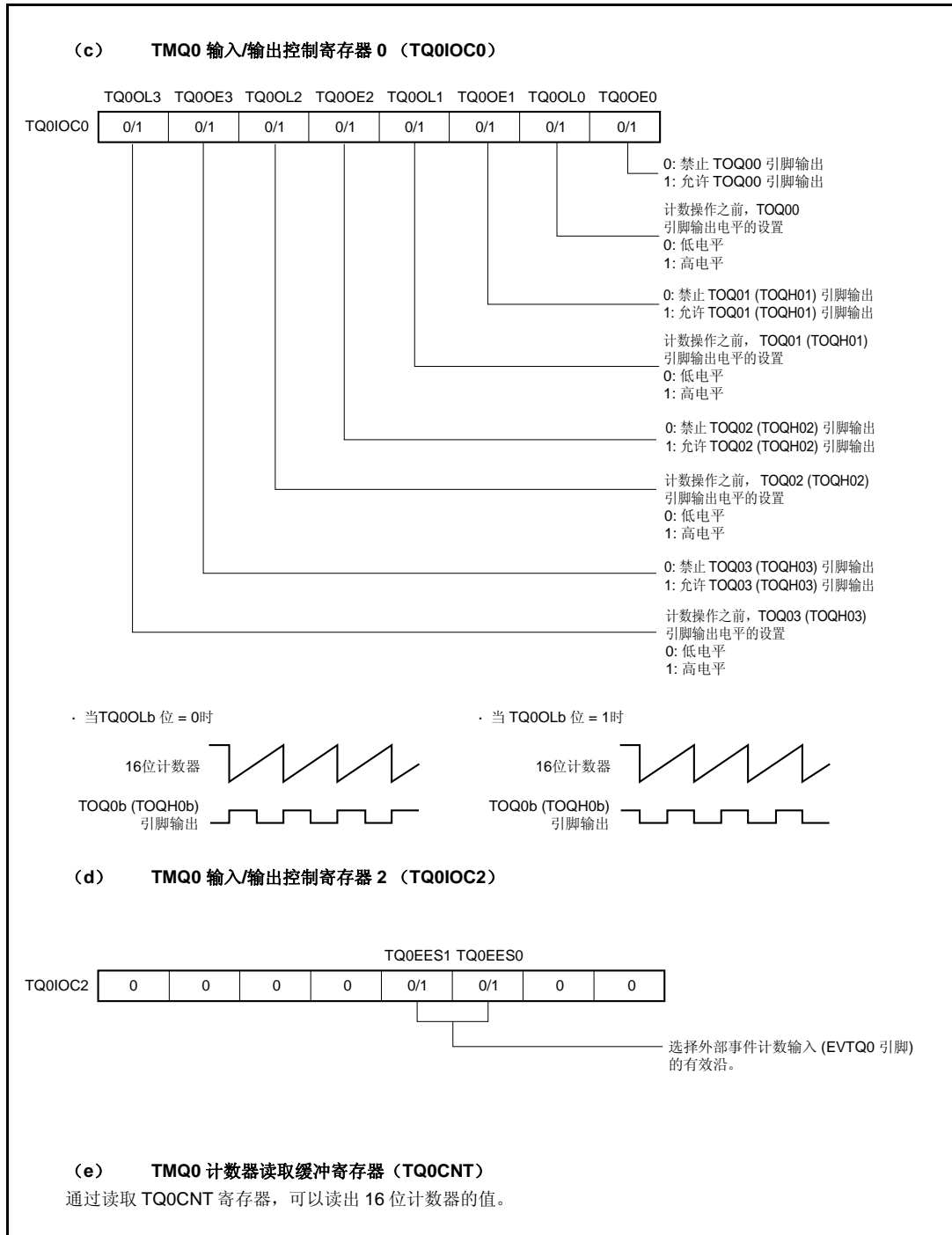


图 7-32. 脉宽调制信号输出模式时的寄存器设置 (3/3)

(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 TQ0CCRB 寄存器的值设置为 D_b ，则脉宽调制信号波形的周期和有效电平如下计算：

PWM 波形周期 = $(D_0 + 1) \times$ 计数时钟周期

PWM 波形有效电平宽度 = $D_b \times$ 计数时钟周期

- 备注**
1. 在脉宽调制信号输出模式时，不使用 TMQ0 输入/输出控制寄存器 1 (TQ0IOC1) 和 TMQ0 选择寄存器 0 (TQ0OPT0)。
 2. $b = 1$ 至 3

(1) 脉宽调制信号输出模式下的工作流程

图 7-33. 脉宽调制信号输出模式时的软件处理流程 (1/2)

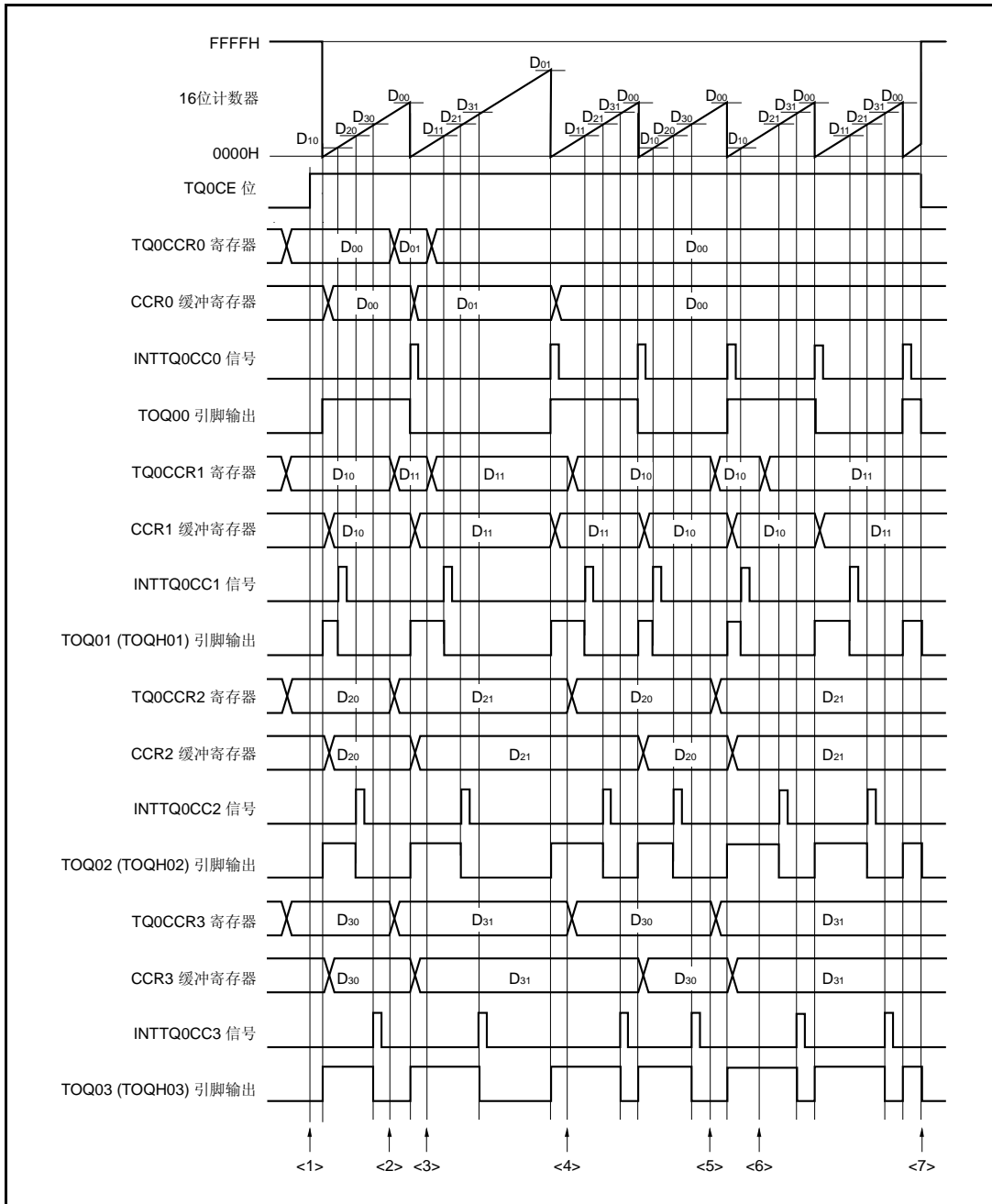
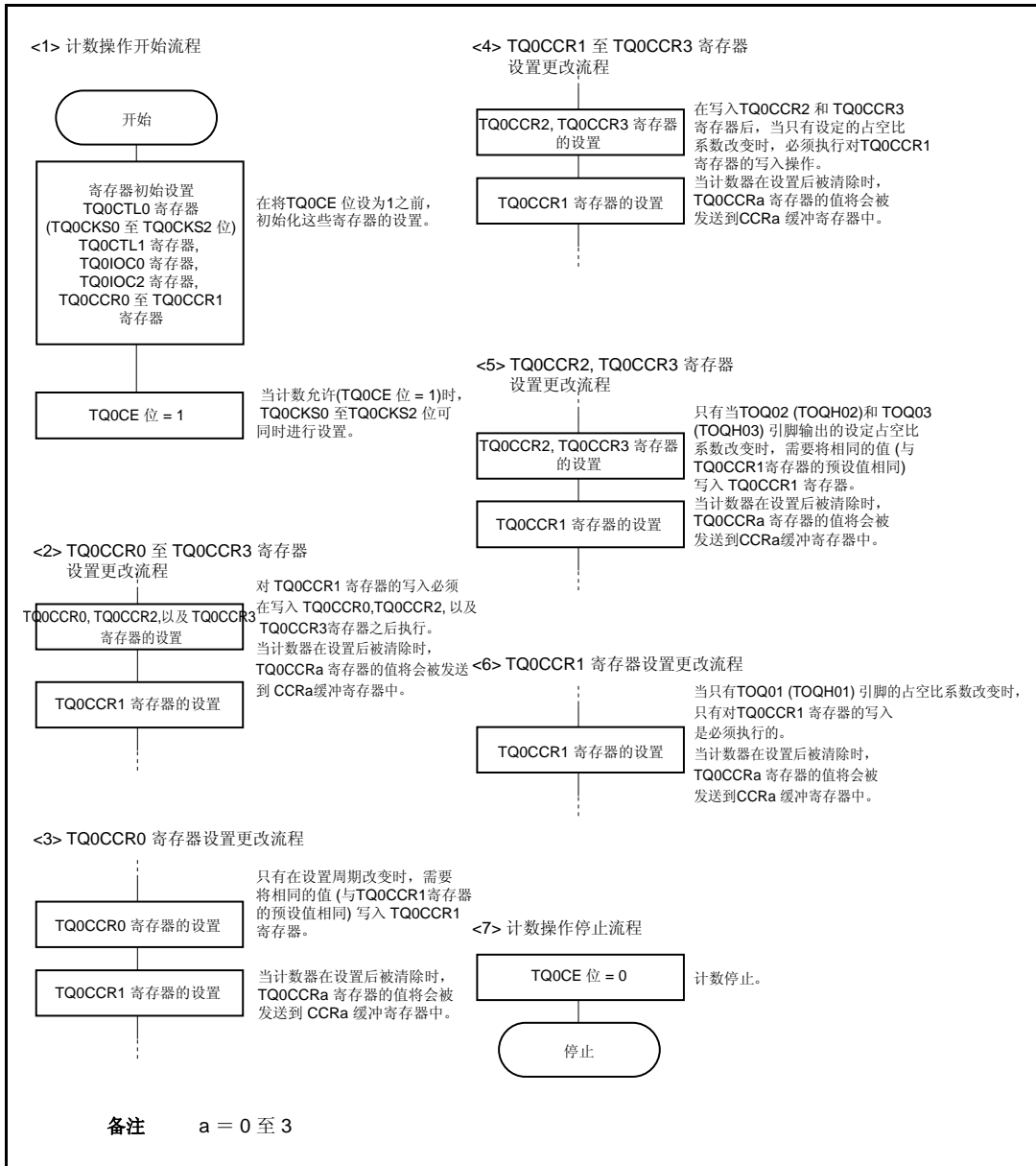


图 7-33. 脉宽调制信号输出模式时的软件处理流程 (2/2)

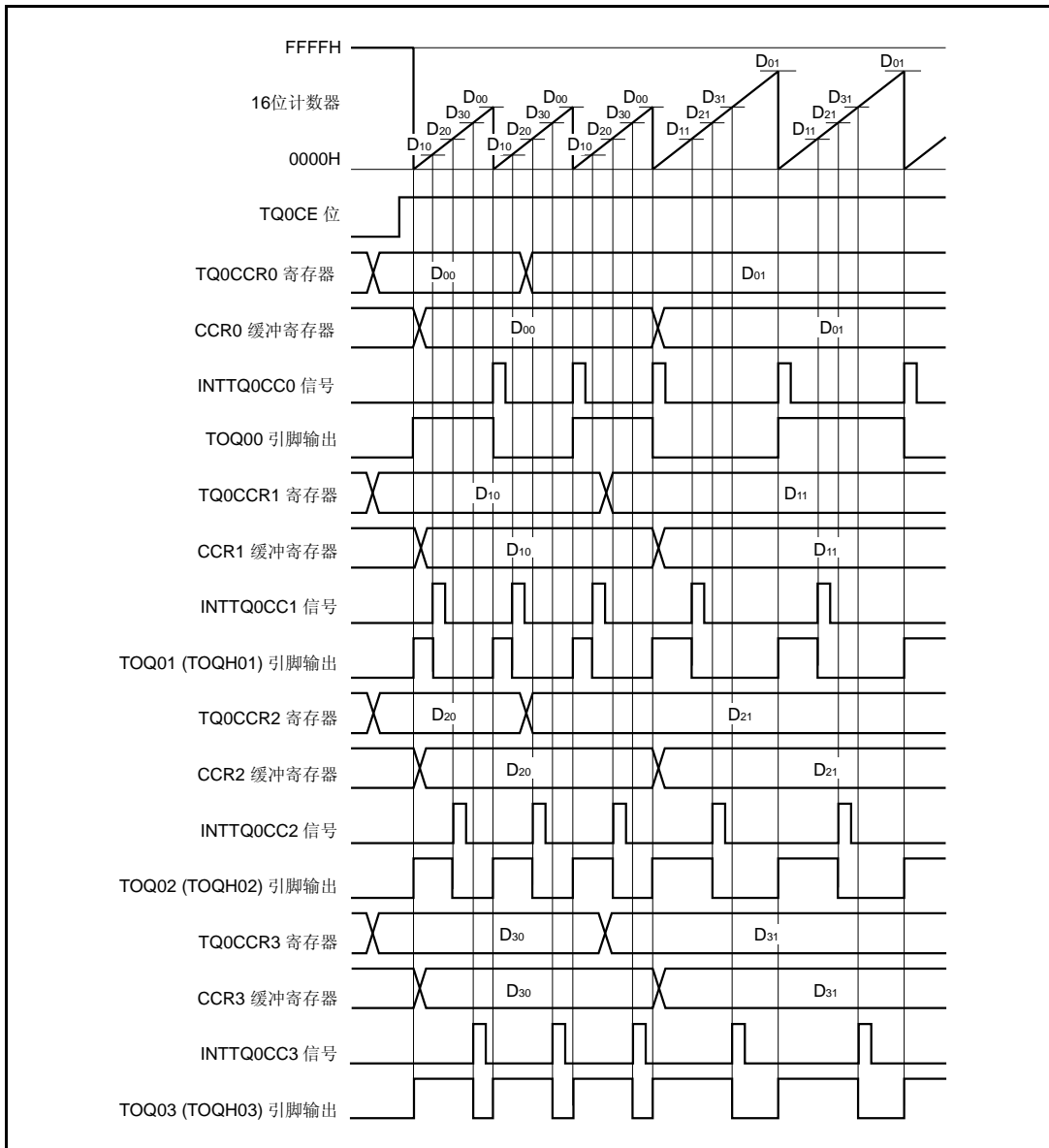


(2) 脉宽调制信号输出模式的操作时序

(a) 操作过程中改变脉冲宽度

计数器计数过程中，若要改变脉宽调制信号波形，应最后写入 TQ0CCR1 寄存器。

在检测到 INTTQ0CC1 信号后，将其写入 TQ0CCR1 寄存器，然后再重写 TQ0CCRa 寄存器。



要把数据从 TQ0CCRa 寄存器传送到 CCRa 缓冲寄存器，必须对 TQ0CCR1 寄存器进行写入操作。

要同时改变脉宽调制信号波形的周期和有效电平宽度，应首先将周期设定给 TQ0CCR0 寄存器，将有效电平宽度设定给 TQ0CCR2 和 TQ0CCR3 寄存器，然后将有效电平设定给 TQ0CCR1 寄存器。

如果只改变脉宽调制信号波形的周期，应首先把周期设置给 TQ0CCR0 寄存器，然后将相同值（与 TQ0CCR1 寄存器预设值相同的值）写入 TQ0CCR1 寄存器。

只改变脉宽调制信号波形的有效电平宽度（占空比系数）时，首先需要向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平值，然后再向 TQ0CCR1 寄存器设置有效电平。

若只改变通过 TOQ01 (TOQH01) 引脚输出的脉宽调制信号波形的有效电平宽度（占空比系数）时，只需要设置 TQ0CCR1 寄存器。

若只改变通过 TOQ02 和 TOQ03 引脚输出的脉宽调制信号波形的有效电平宽度（占空比系数）时，首先要向 TQ0CCR2 (TOQH02) 和 TQ0CCR3 (TOQH03) 寄存器设置有效电平宽度，然后向 TQ0CCR1 寄存器写入相同的值（等于 TQ0CCR1 寄存器的预置值）。

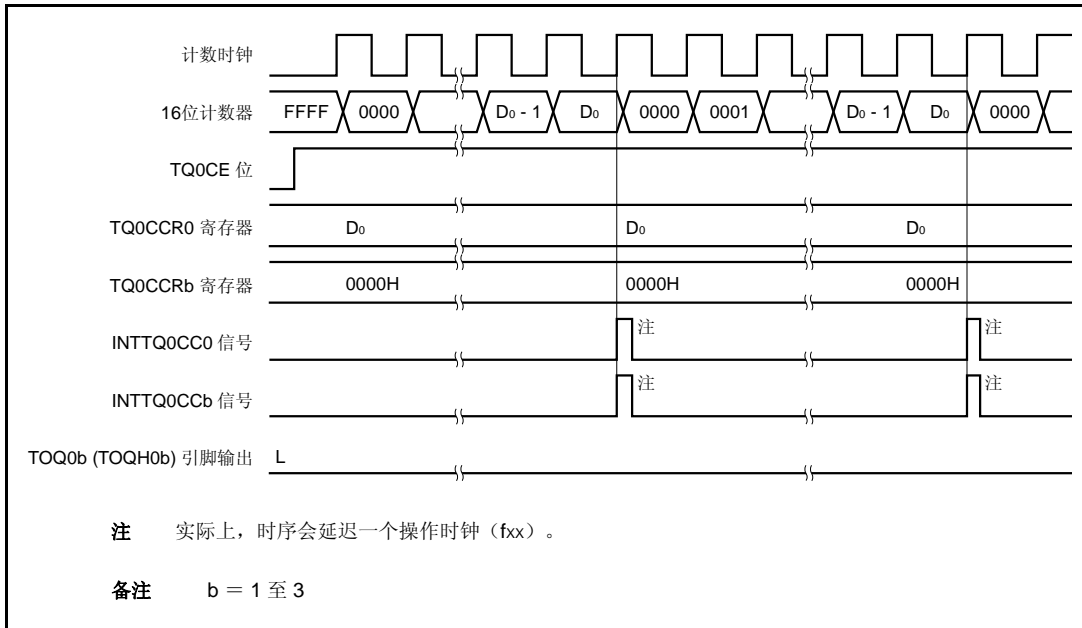
在数据写入 TQ0CCR1 寄存器后，写入 TQ0CCRa 寄存器的值会传送到 CCRa 缓冲寄存器中，与 16 位计数器清零同步，并作为与 16 位计数器进行比较的值。

要在写入 TQ0CCR1 寄存器一次之后，再次写入 TQ0CCR0 至 TQ0CCR3 寄存器，应在 INTTQ0CC0 信号产生之后写入，否则，CCRa 缓冲寄存器的值可能不确定，因为把数据从 TQ0CCRa 寄存器传送到 CCRa 缓冲寄存器的时间与写入 TQ0CCRa 寄存器相冲突。

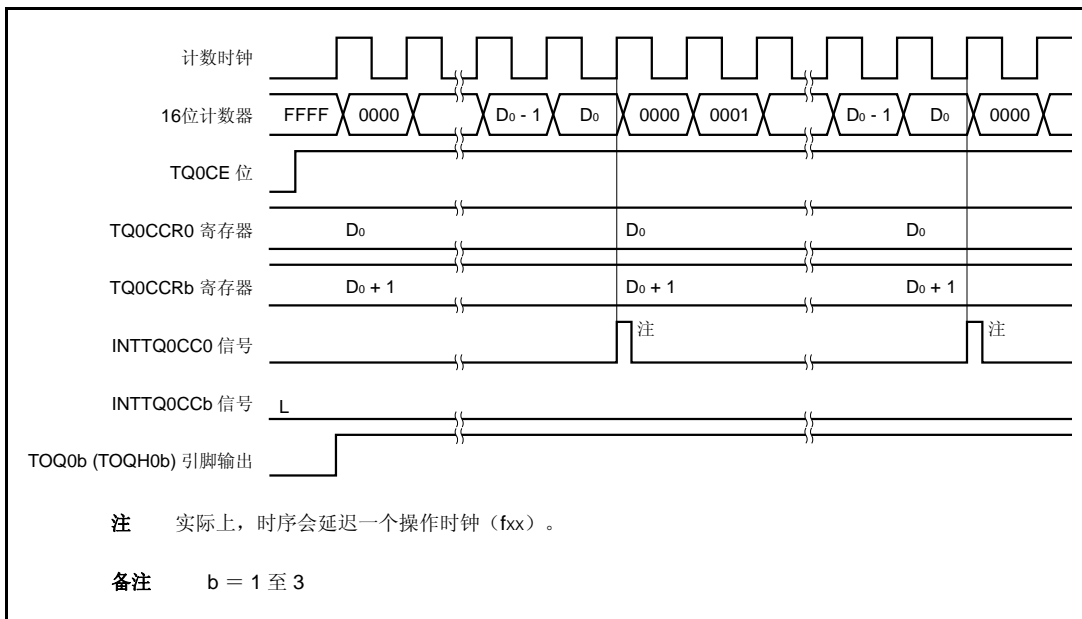
备注 a = 0 至 3

(b) 脉宽调制信号波形的 0% 和 100% 输出

若要输出 0% 的波形，则需将 TQ0CCRb 寄存器设置为 0000H。在下次 16 位计数器的值与 CCR0 缓冲寄存器的值匹配时，16 位计数器将会被清除为 0000H 且会生成 INTTQ0CC0 和 INTTQ0CCb 信号。



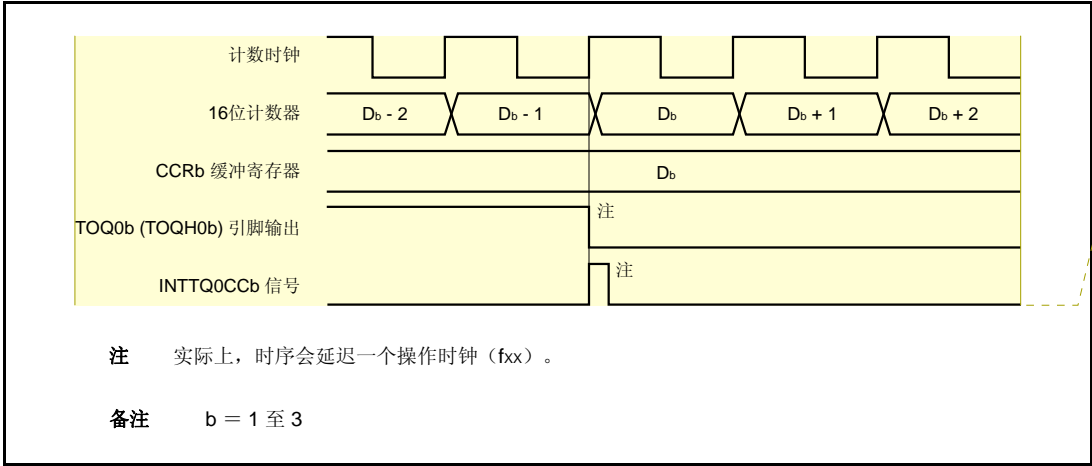
若要输出 100% 波形，则需将 TQ0CCRb 寄存器设置为 (TQ0CCR0 寄存器的设定值 + 1)。TQ0CCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。



<R>

(c) 比较匹配中断请求信号 (INTTQ0CCb) 的发生时序

在脉宽调制信号输出模式下，比较匹配中断请求信号 INTTQ0CCb 的发生时序与其它模式下 INTTQ0CCb 信号的时序有所不同；当 16 位计数器的计数值与 TQ0CCRb 寄存器的值匹配时，产生 INTTQ0CCb 信号。



批注 [任远1]: 图中口去掉

通常情况下，在 16 位计数器的当前计数值与 TQ0CCRb 寄存器的值匹配之后，INTTQ0CCb 信号与和下一次计数同步发生。

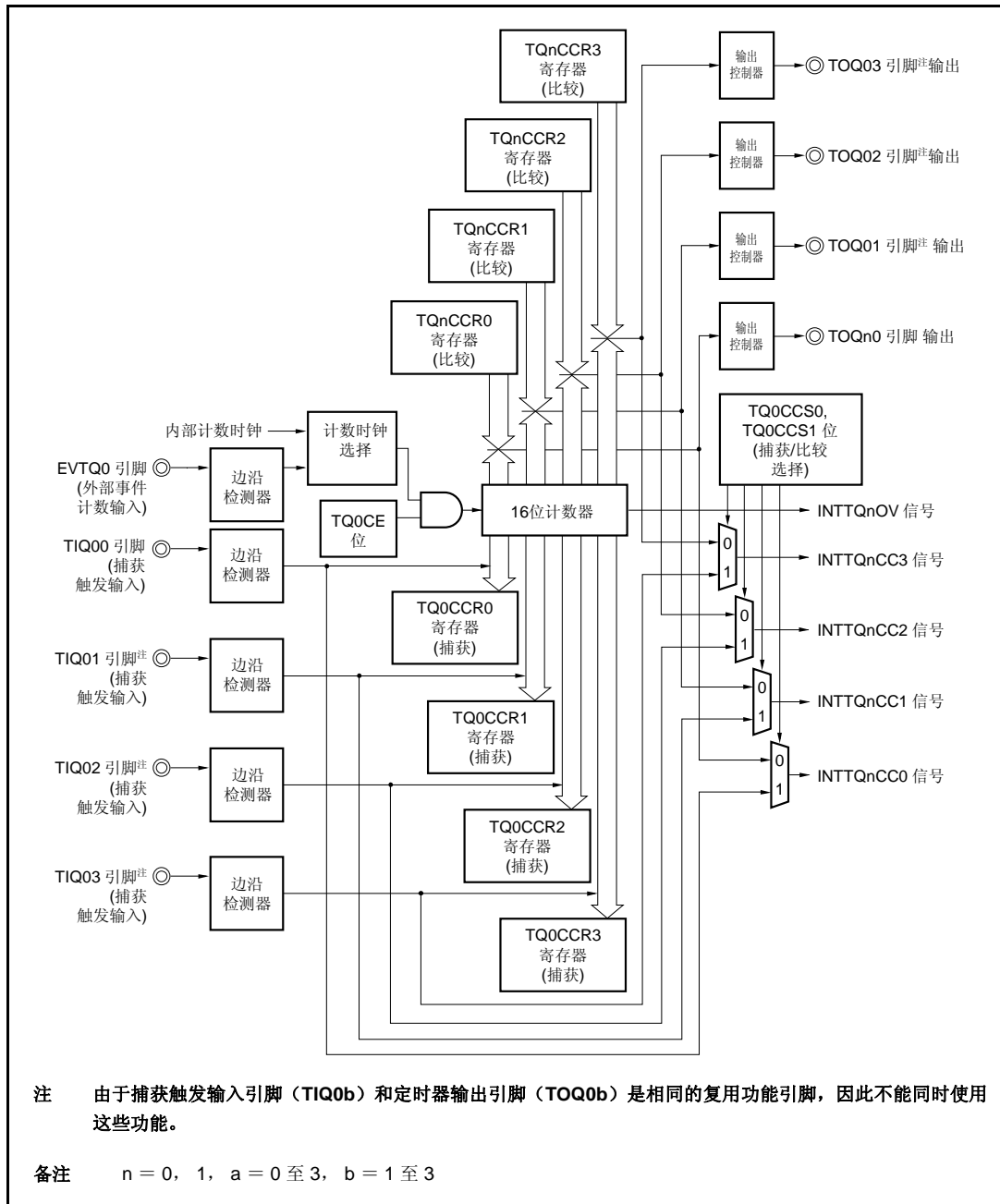
但是，在脉宽调制信号输出模式下，INTTQ0CC1 信号会提早一个计数时钟周期产生。这是由于为了匹配 TOQ0b (TOQH0b) 引脚输出信号的变化，时序发生了变化。

7.6.6 自由运行定时器模式 (TQnMD2 至 TQnMD0 位 = 101)

比较功能在 TMQ0 和 TMQ1 中均有效。捕获功能仅在 TMQ0 中有效。

在自由运行定时器模式下，当 TQnCTL0.TQnCE 位被置为 1 时，16 位定时器/事件计数器 Q 开始计数。此时，TQ0CCR_a 寄存器根据 TQ0OPT0.TQ0CCSa 位的设置，可以用作比较寄存器或捕获寄存器。

图 7-34. 自由运行定时器模式的配置

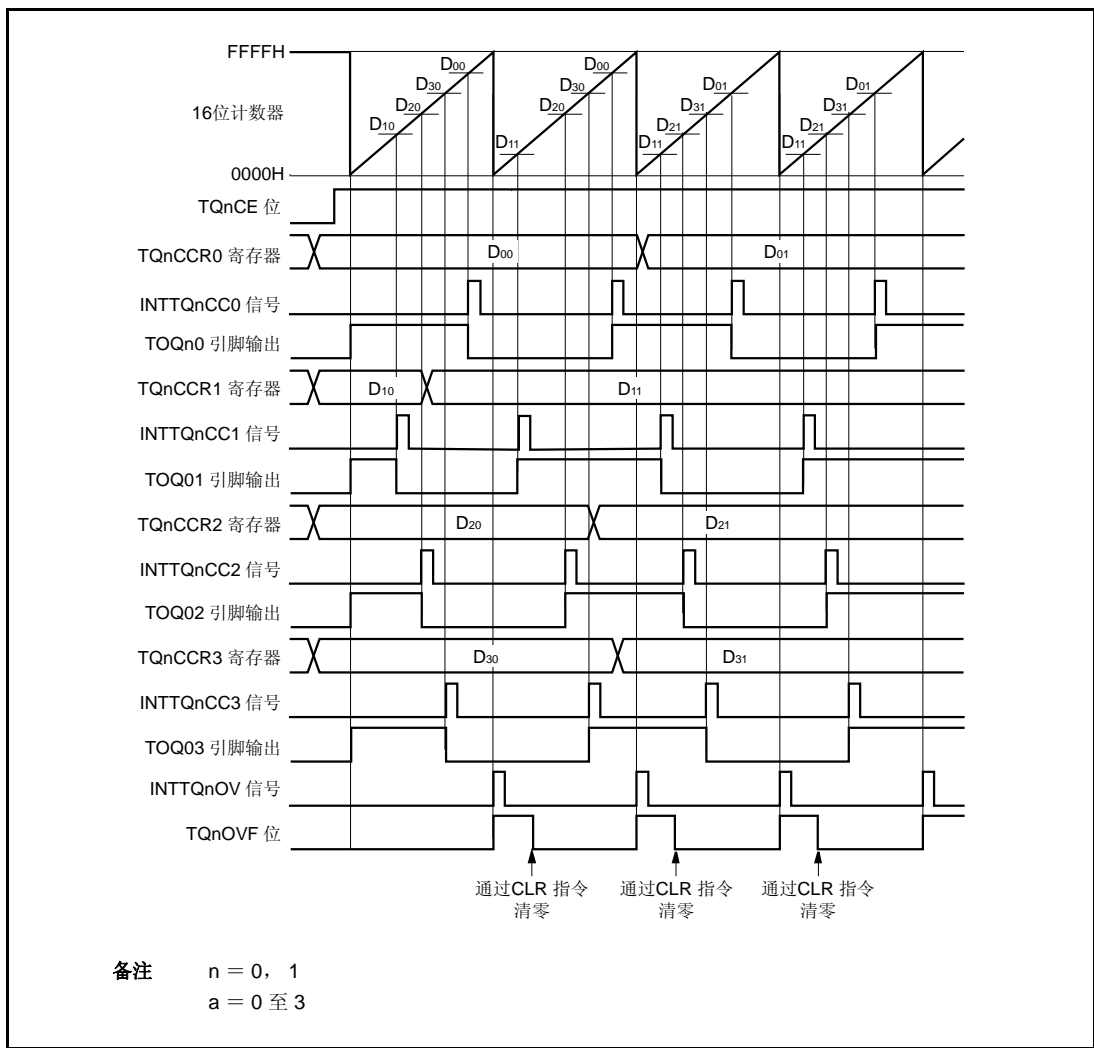


• 比较操作

当 TQnCE 位被设置为 1 时，16 位定时器/事件计数器 Q 开始计数，TOQ03 和 TOQ10 引脚的输出信号反向。此后当 16 位计数器的计数值与 TQnCCRa 寄存器的设置值匹配时，产生一个比较匹配中断请求信号 (INTTQnCCa)，并且 TOQ00 至 TOQ03 和 TOQ10 引脚的输出信号反向。16 位计数器继续与计数时钟同步进行计数。当它计数到 FFFFH 时，将在下一个计数时钟时产生一个溢出中断请求信号 (INTTQnOV)，计数器被清除为 0000H，并重新开始计数。这种情况下，溢出标志位 (TQnOPT0.TQnOVF 位) 也将被设置为 1。在确认溢出标志被设为 1 后通过软件使用 CLR 指令将溢出标志清零。

当计数器工作时，TQnCCRa 寄存器可以重写。如果对寄存器进行了重写，则新写入的值将随时生效并与计数值进行比较。

图 7-35. 自由运行定时器模式下的基本时序 (比较功能)



• 捕获操作

当 TQ0CE 位被设置 1 时，16 位计数器开始计数。当检测到 TIQ0a 引脚输入信号的有效沿时，16 位计数器的计数值被保存在 TQ0CCRa 寄存器中，同时产生一个捕获中断请求信号 (INTTQ0CCa)。

16 位计数器继续与计数时钟同步进行计数。

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，并继续进行计数。这种情况下，溢出标志位 (TQ0OVF 位) 也将被设置为 1。在确认溢出标志被设为 1 后通过软件使用 CLR 指令将溢出标志清零。

图 7-36. 自由运行定时器模式下的基本时序 (捕获功能)

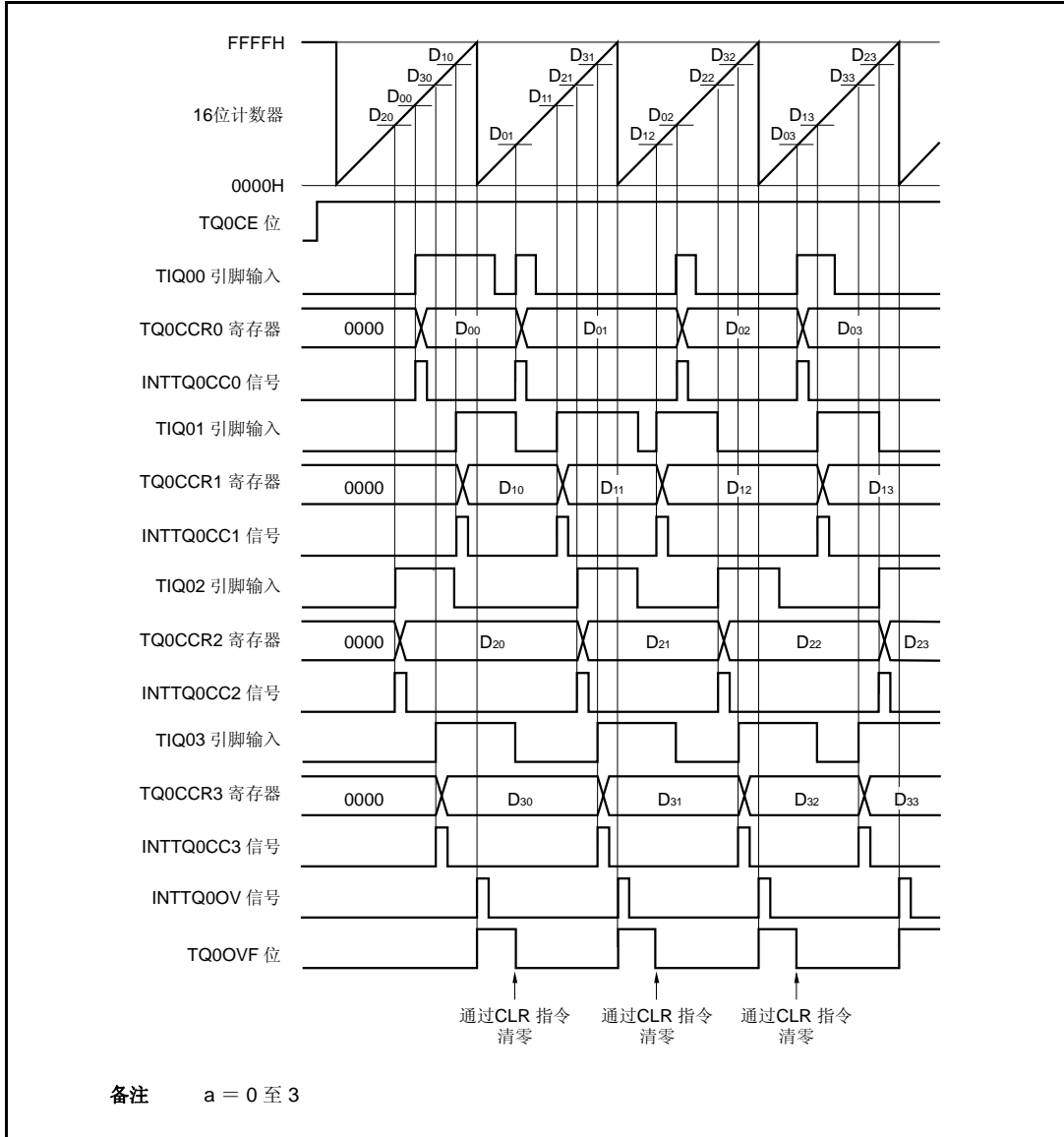


图 7-37. 自由运行定时器模式下的寄存器设置 (1/3)

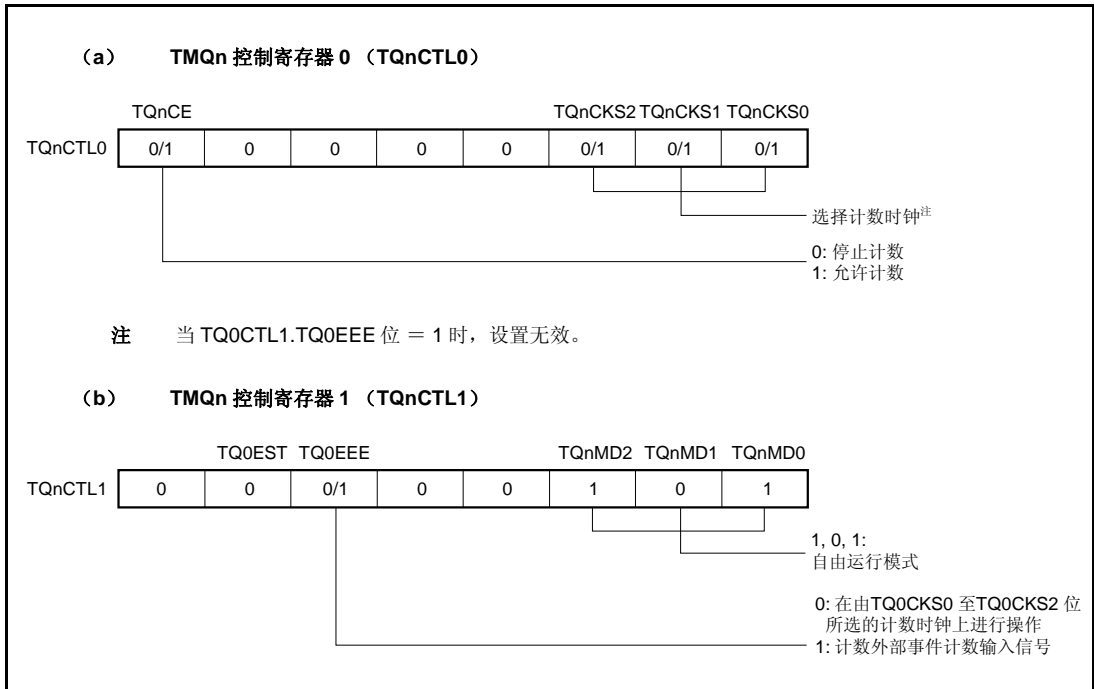


图 7-37. 自由运行定时器模式下的寄存器设置 (2/3)

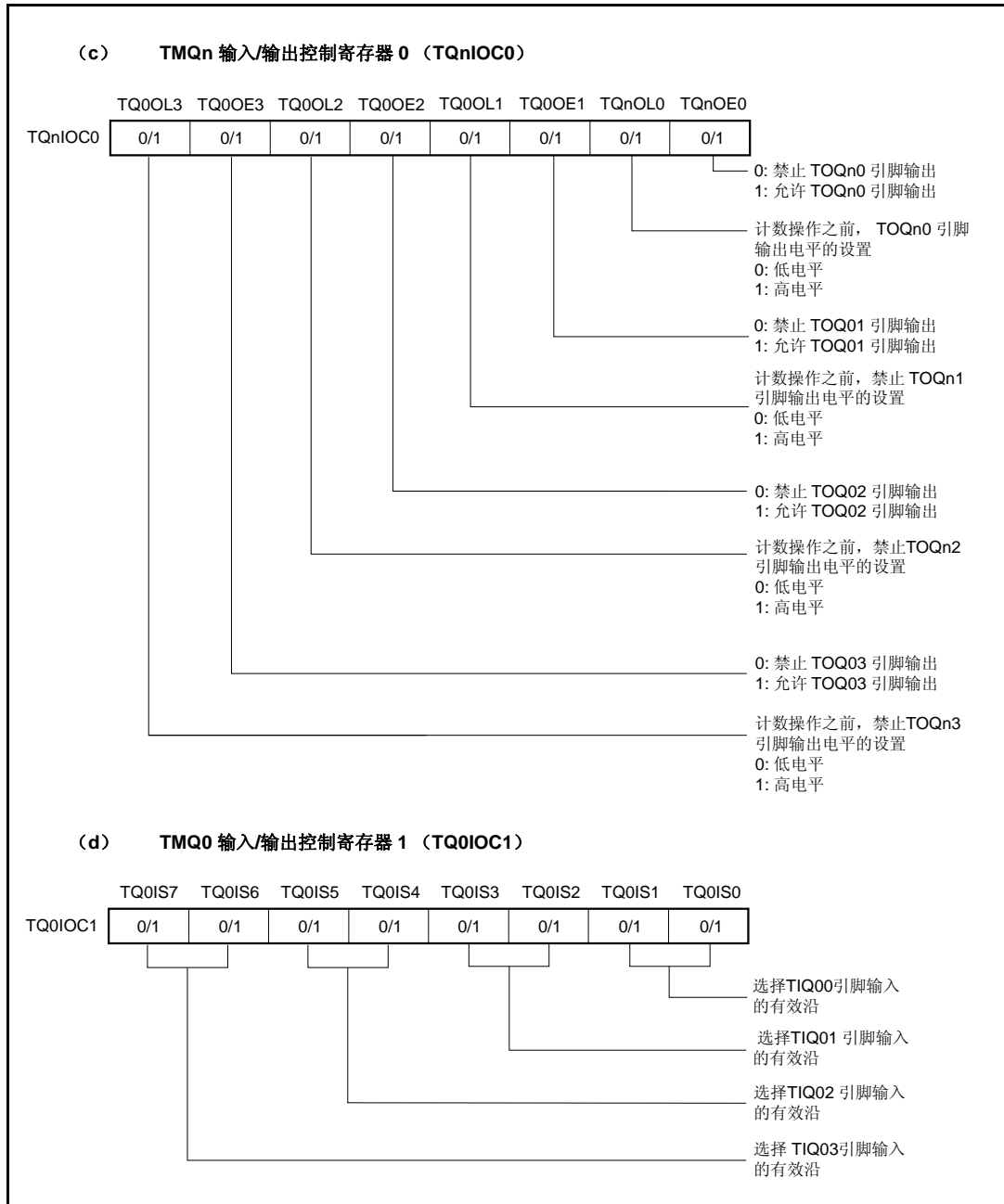
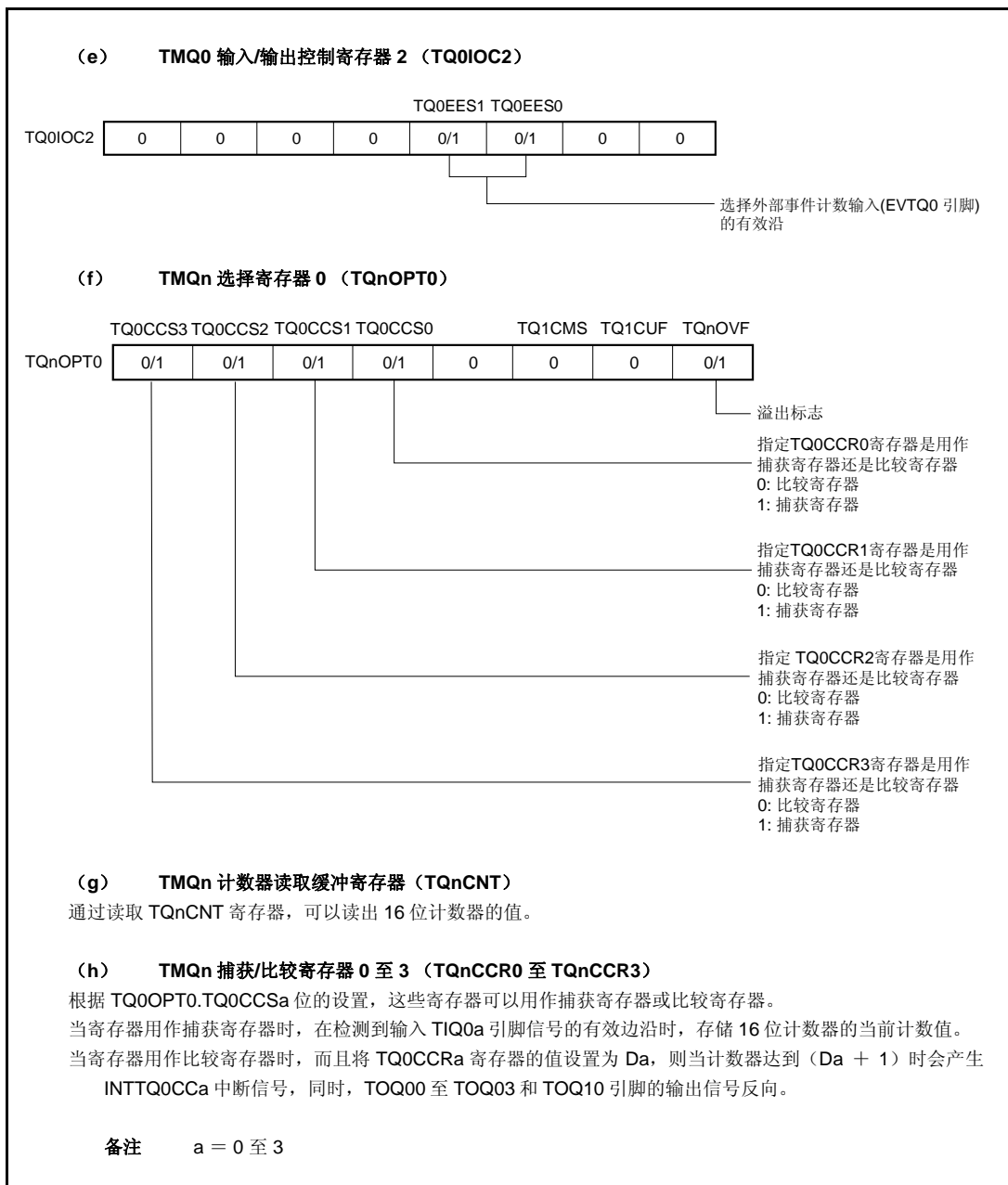


图 7-37. 自由运行定时器模式下的寄存器设置 (3/3)



(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 7-38. 自由运行定时器模式时的软件处理流程 (比较功能) (1/2)

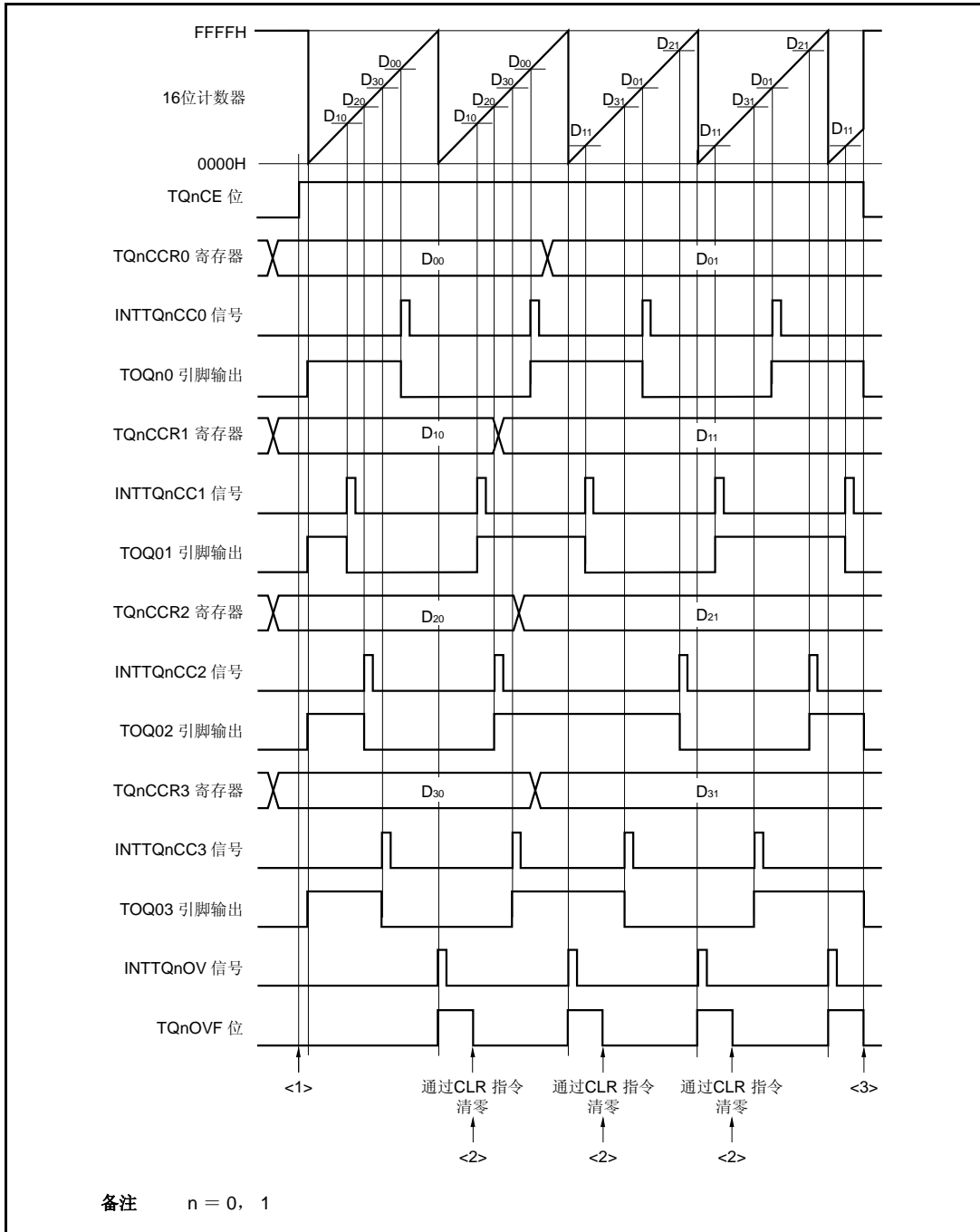
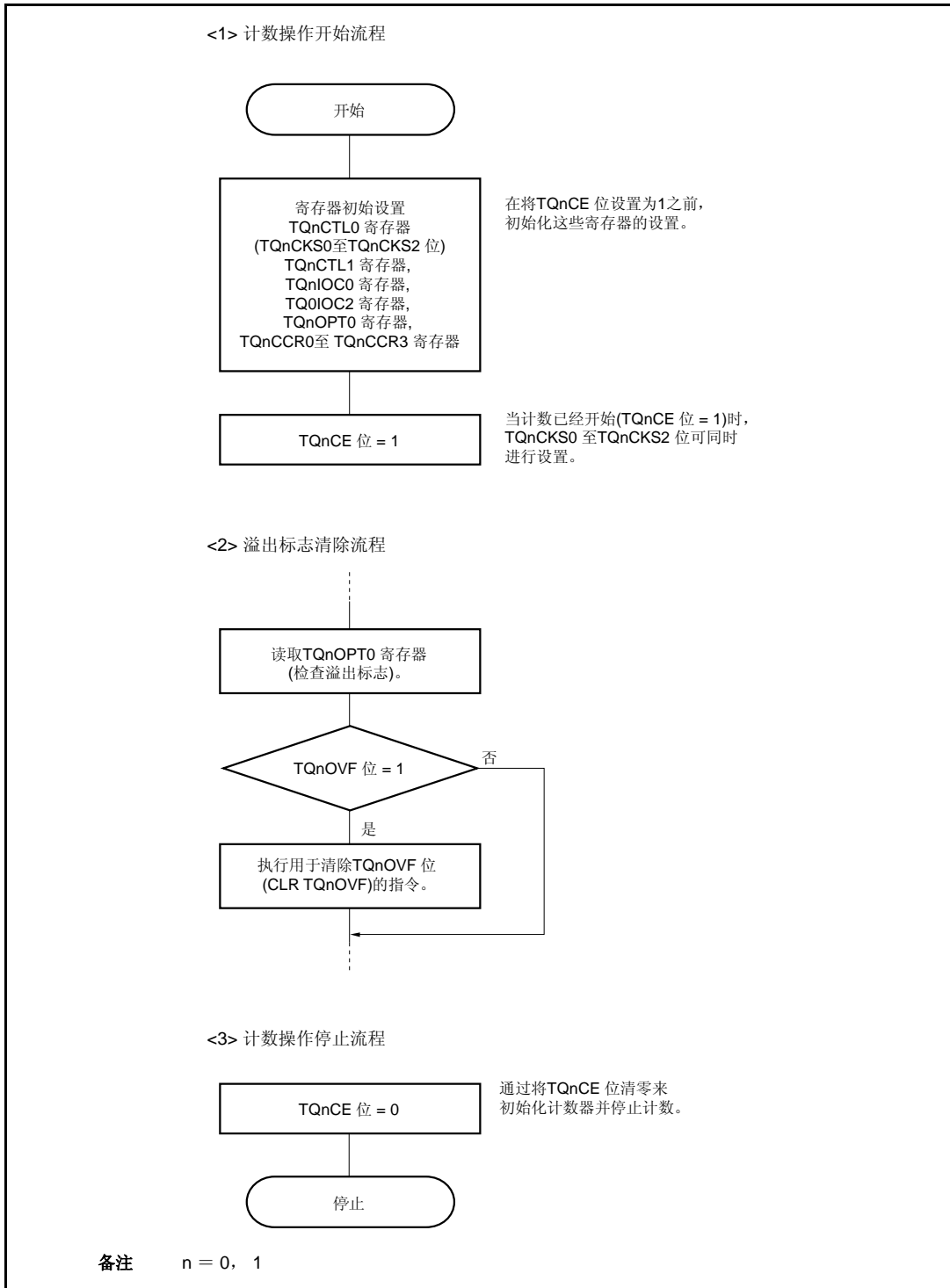


图 7-38. 自由运行定时器模式时的软件处理流程 (比较功能) (2/2)



(b) 当捕获/比较寄存器用作捕获寄存器时

图 7-39. 自由运行定时器模式时的软件处理流程 (捕获功能) (1/2)

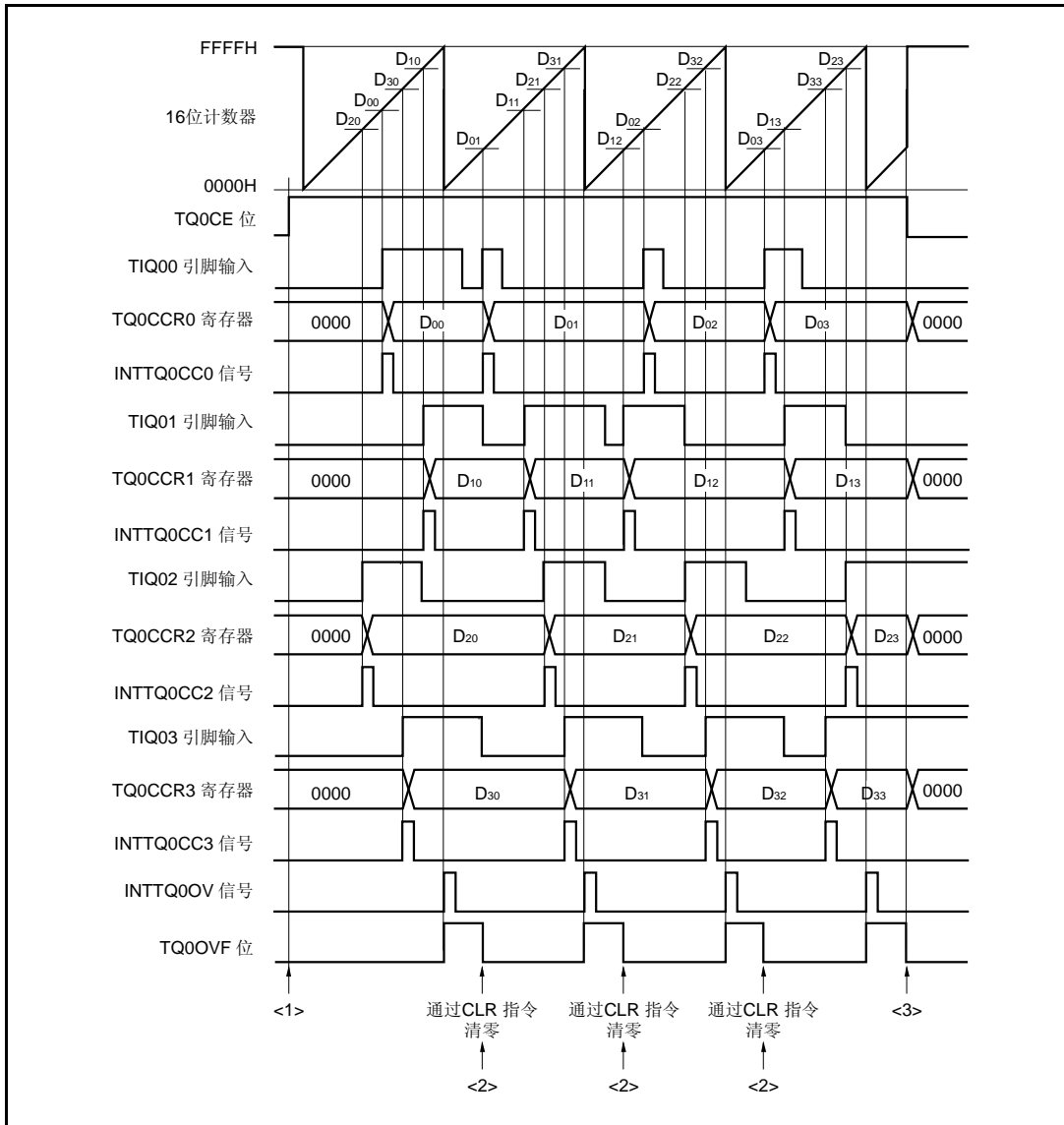
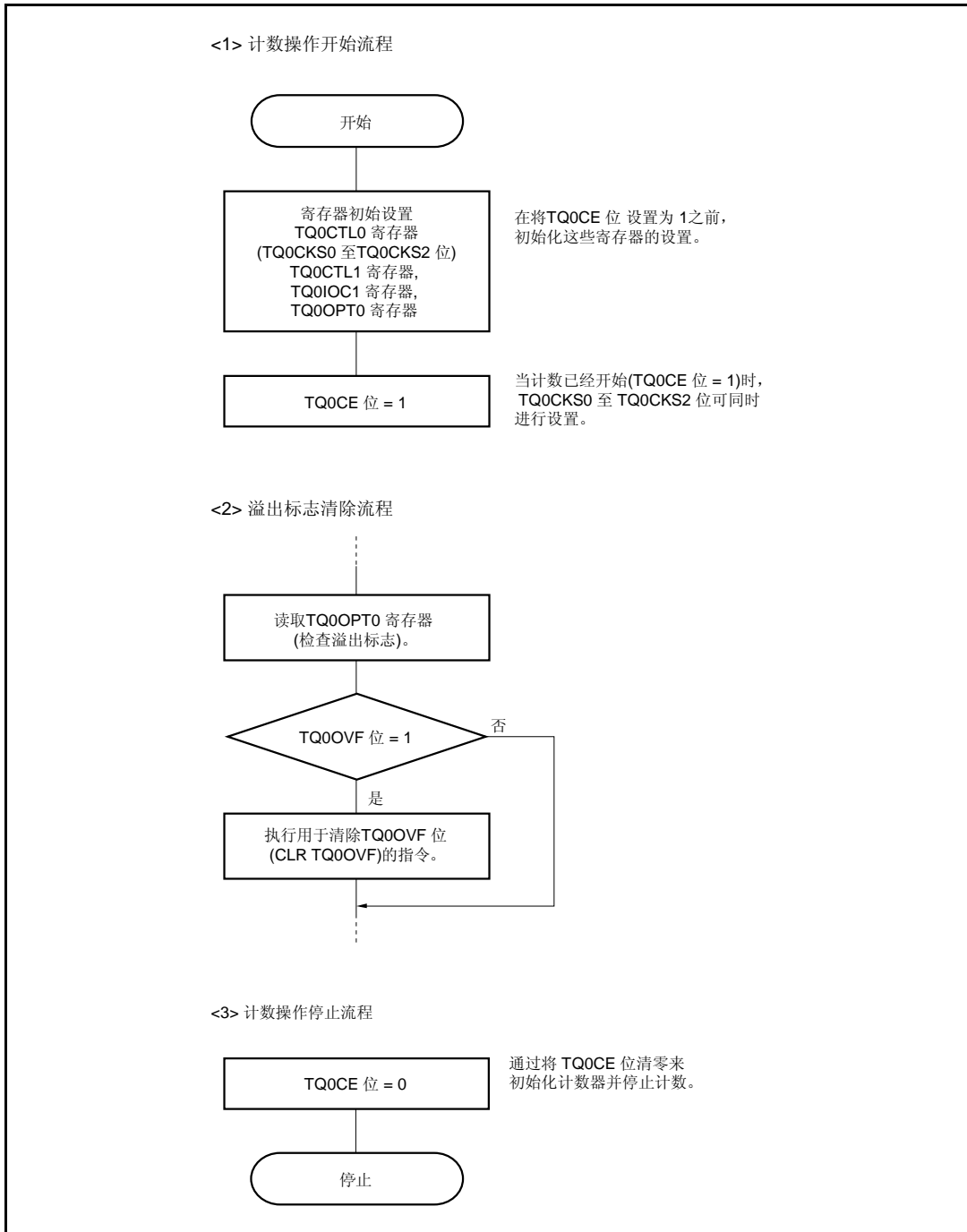


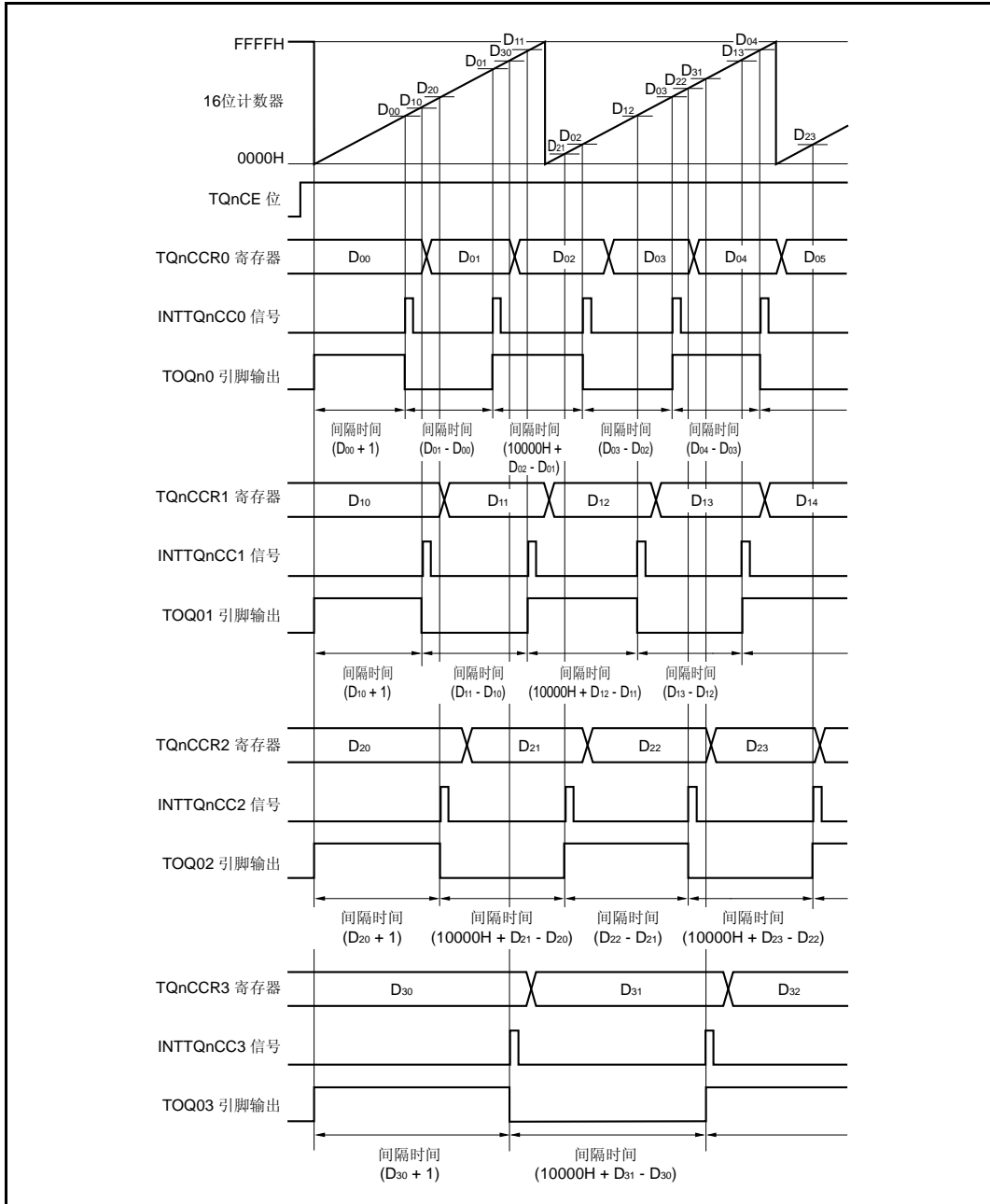
图 7-39. 自由运行定时器模式时的软件处理流程 (捕获功能) (2/2)



(2) 自由运行定时器模式时的工作时序

(a) 使用比较寄存器的间隔定时操作

当 16 位定时器/事件计数器 Q 用作间隔定时器而 TQnCCRa 寄存器用作比较寄存器时，在每次检测到 INTTQnCCa 信号后，都需要利用软件操作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 INTTQnCCa 信号后，必须在中断服务程序中重置相应的 TQnCCRa 寄存器。

重置 TQnCCRa 寄存器的设置值可由下式计算，其中“Da”是间隔时间：

比较寄存器默认值： $D_a - 1$

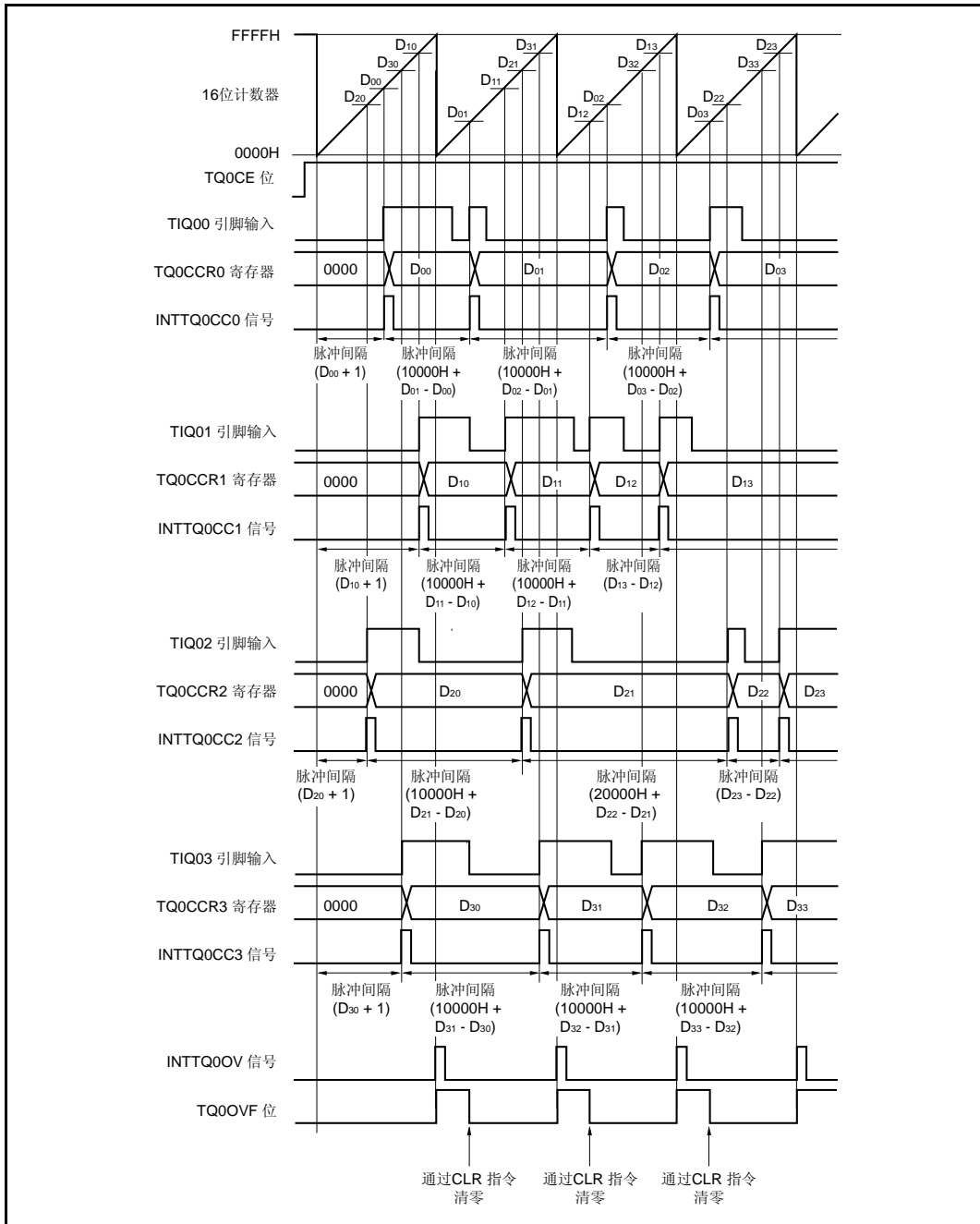
比较寄存器的第二次及其以后的设置值：前次设置值 + D_a

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

备注 n = 0, 1
 a = 0 至 3

(b) 使用捕获寄存器进行脉冲宽度测量

在用 TQ0CCRa 寄存器作为捕获寄存器进行脉宽测量时，每次检测到 INTTQ0CCa 信号时读取捕获寄存器及计算时间间隔，都需要进行软件处理。



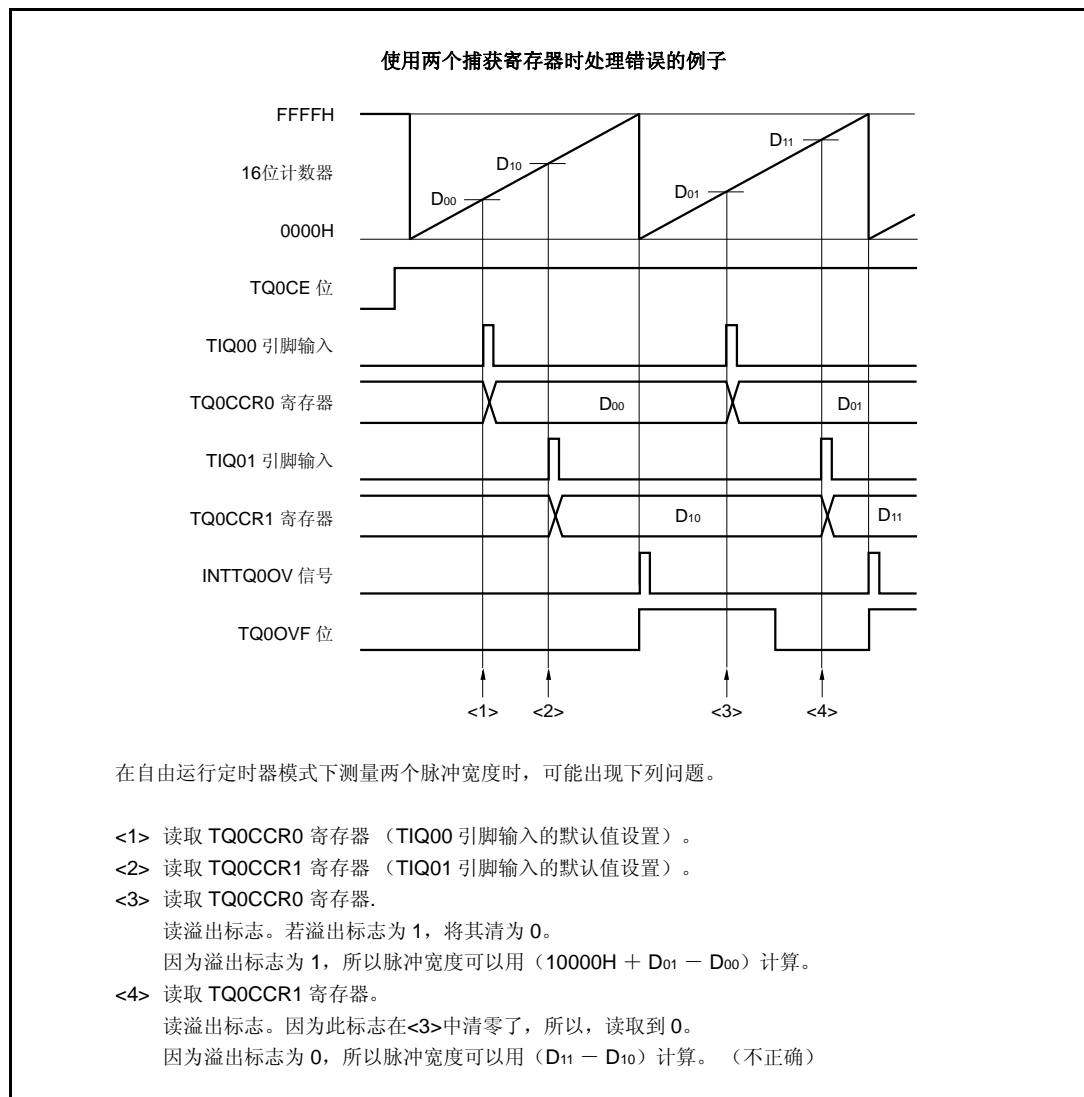
在自由运行定时器模式下执行脉宽测量时，可以用一个通道测量四个脉冲宽度。

要测量脉冲宽度的话，可以通过与 INTTQ0CCa 信号同步读取 TQ0CCRa 寄存器的值，并计算本次读取值与上一次读取值之间的差，计算得到脉冲宽度。

备注 a = 0 至 3

(c) 使用两个捕获寄存器时对溢出的处理

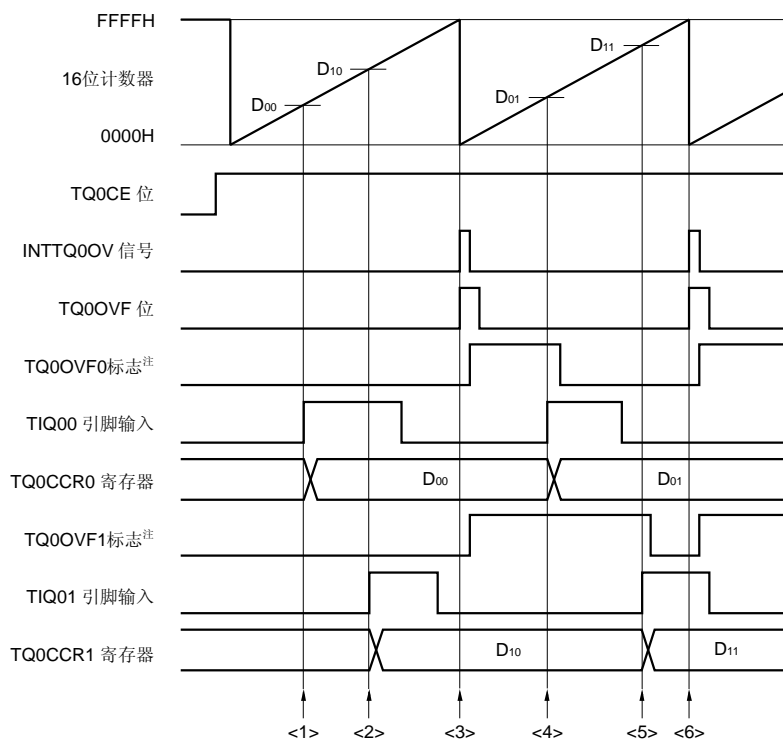
当使用两个捕获寄存器时必须格外小心。首先，下图所示为一个不正确处理的例子。



当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不会得到正确的脉冲宽度。

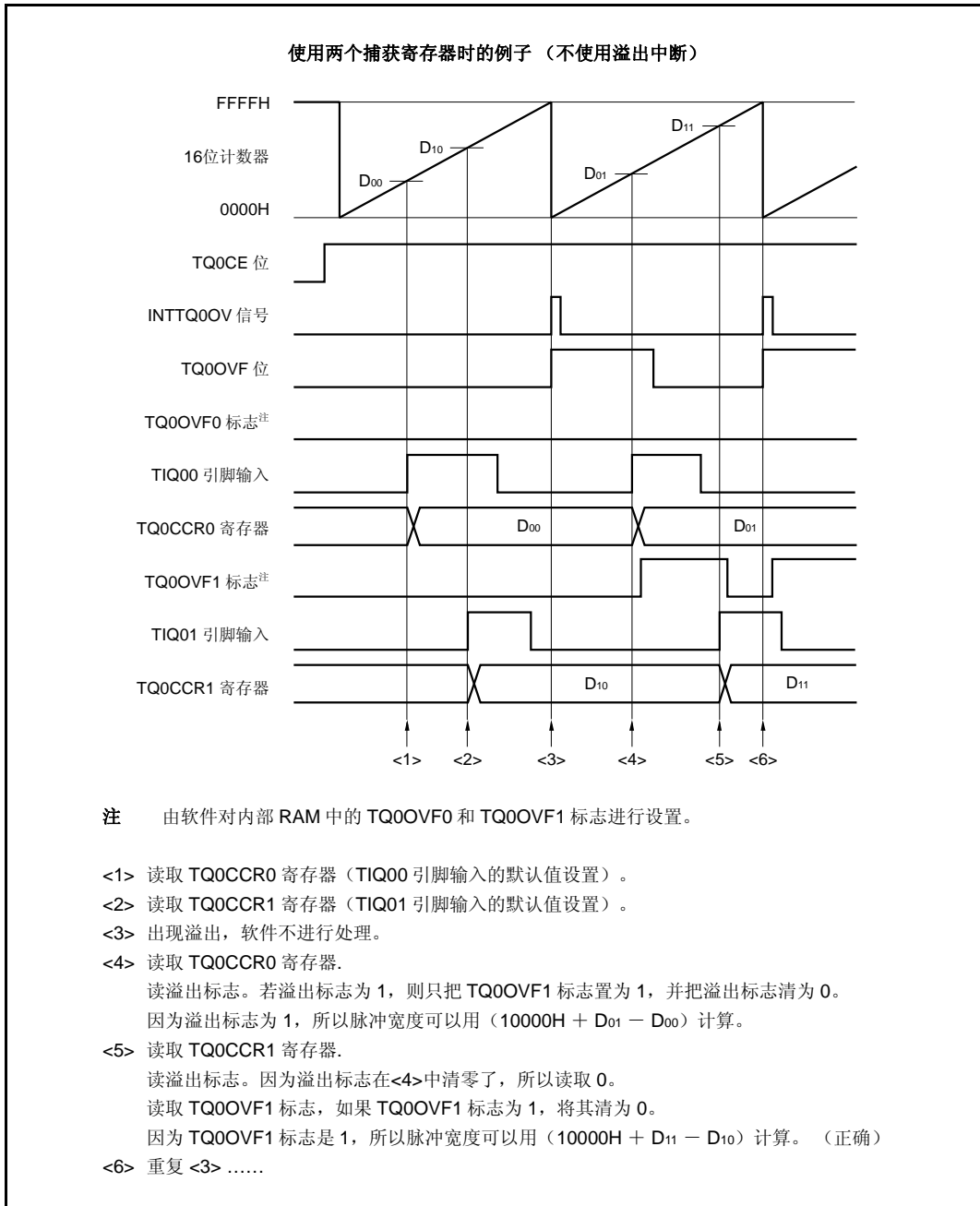
当使用两个捕获寄存器时，使用软件解决。关于如何使用软件，示例如下。

使用两个捕获寄存器时的例子 (使用溢出中断)



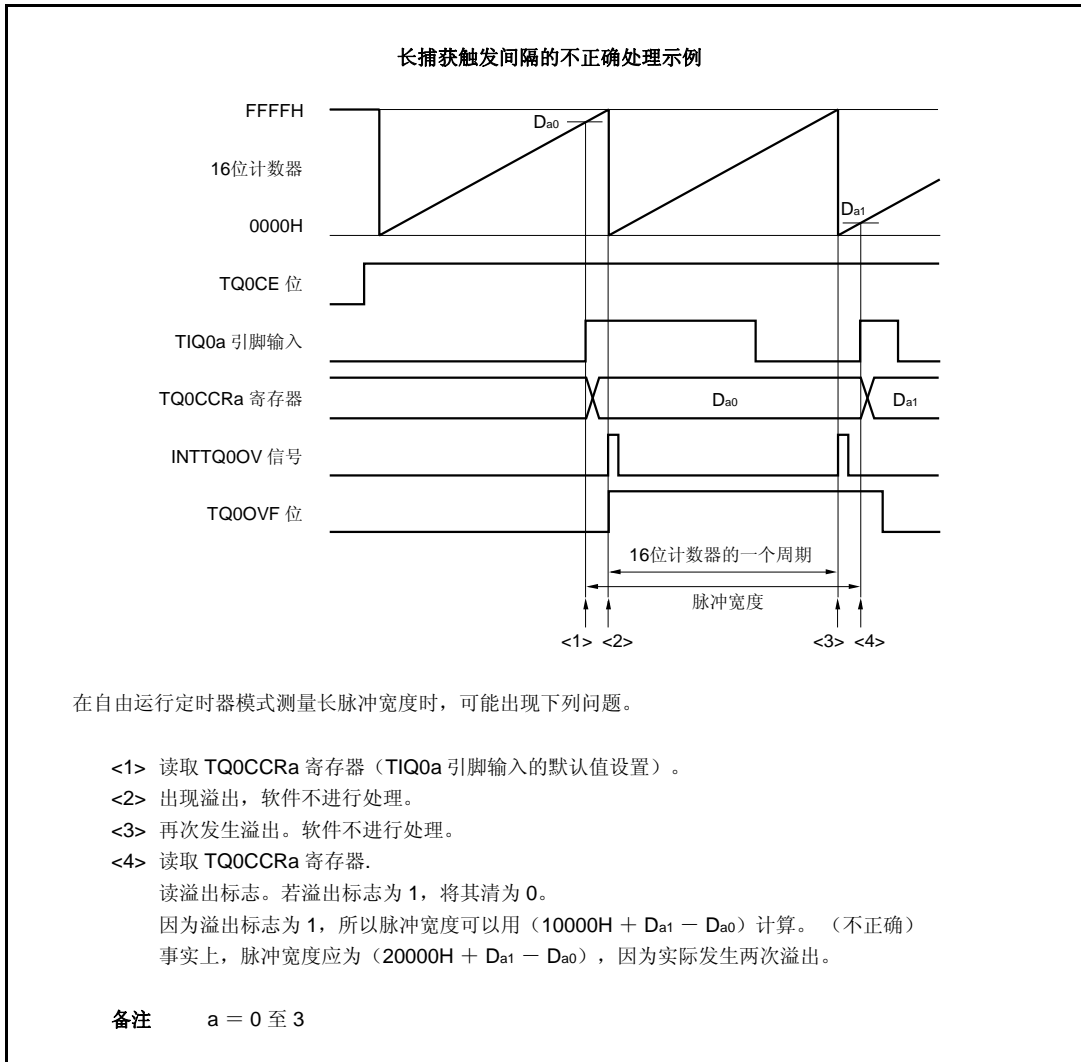
注 由软件对内部 RAM 中的 TQ0OVF0 和 TQ0OVF1 标志进行设置。

- <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> 一个溢出发生, 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, 将其清除为 0。
因为 TQ0OVF0 标志为 1, 所以脉冲宽度可以利用 $(10000H + D_{01} - D_{00})$ 计算得到。
- <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, 而且 TQ0OVF1 标志仍然保持为 1)。
因为 TQ0OVF1 标志是 1, 所以脉冲宽度可以用 $(10000H + D_{11} - D_{10})$ 计算。(正确)
- <6> 重复 <3>



(d) 长捕获触发间隔时溢出的处理

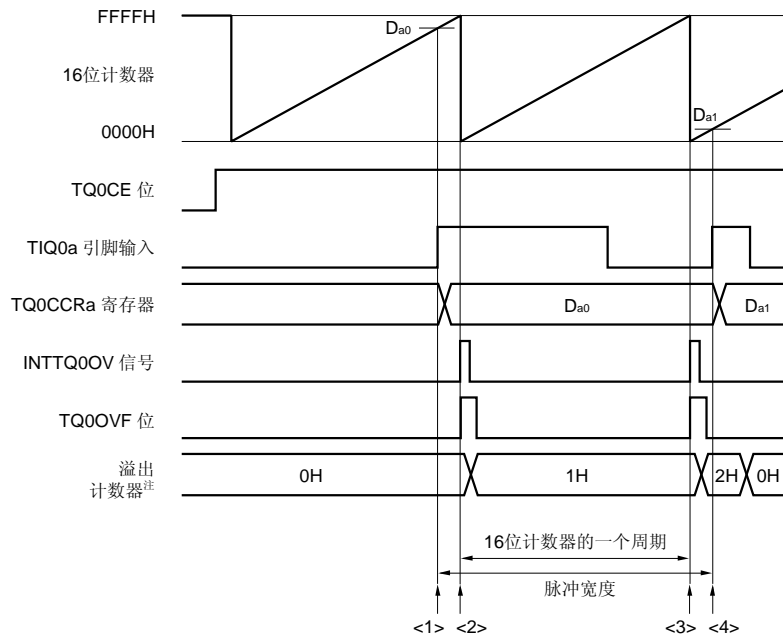
如果脉冲宽度大于 16 位计数器的一个计数循环，则必须加以注意，因为从第一个捕获触发到下一个之间可能发生不止一次溢出。首先，如下所示，是一个不正确处理的例子。



如果长捕获触发间隔期间，发生了两次或更多次的溢出，则无法获得正确的脉冲宽度。

如果捕获触发间隔为长间隔（相对于计数循环），应减慢计数时钟以延长 16 位计数器的循环周期，或者使用软件加以解决。关于如何使用软件，示例如下。

长捕获触发间隔时的举例



注 内部 RAM 中的溢出计数器由软件任意设置。

- <1> 读取 TQ0CCRa 寄存器 (TIQ0a 引脚输入的默认值设置)。
- <2> 发生溢出, 增加 (+1) 溢出计数器的值, 并在溢出中断服务程序中把溢出标志清为 0。
- <3> 再次发生溢出。增加 (+1) 溢出计数器的值, 并在溢出中断服务程序中把溢出标志清为 0。
- <4> 读取 TQ0CCRa 寄存器。

读取溢出计数器。

→当溢出计数器为“N”时, 脉冲宽度可用 $(N \times 10000H + D_{a1} - D_{a0})$ 计算。

在本例中, 脉冲宽度为 $(20000H + D_{a1} - D_{a0})$, 因为溢出发生了两次。

溢出计数器清零 (0H)。

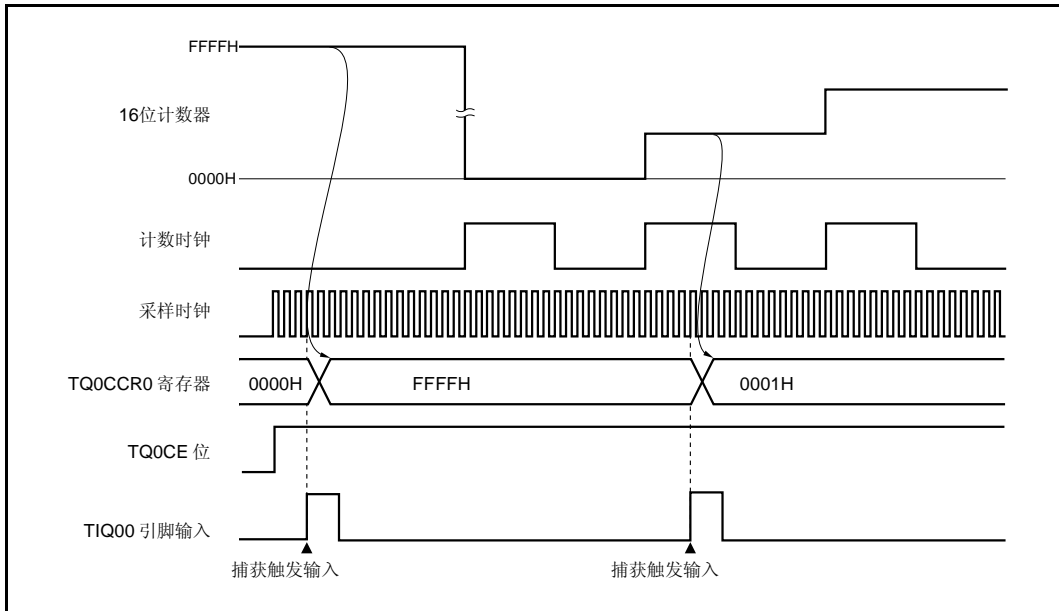
备注 a = 0 至 3

(e) 清除溢出标志

在 TQnOVF 位为 1 时读取该位后, 使用 CLR 指令将 TQnOVF 位清零可以将溢出标志清零, 而在 TQnOVF 位为 1 时读取该位后, 通过将 8 位数据 (位 0 为 0) 写入 TQnOPT0 寄存器也可以将溢出标志清零。

(3) 捕获操作的注意事项

如果使用捕获操作并且将低速时钟选择为计数时钟时，如果在 TQ0CTL0.TQ0CE 位置 1 之后捕获触发器立即输入，那么可以被捕获到 TQ0CCRa 寄存器中的是 FFFFH，而不是 0000H。



7.6.7 脉宽测量模式 (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 TIQ0a 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 TQ0CCRa 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 (INTTQ0CCa) 之后，可以通过读取 TQ0CCRa 寄存器，测量有效边缘的间隔。

如图 7-41 所示，选择 TIQ00 至 TIQ03 引脚中的任何一个引脚作为捕获触发输入引脚。通过使用 TQ0IOC1 寄存器将未使用的引脚设置为“无边沿检测”。

图 7-40. 脉冲宽度测量模式的配置

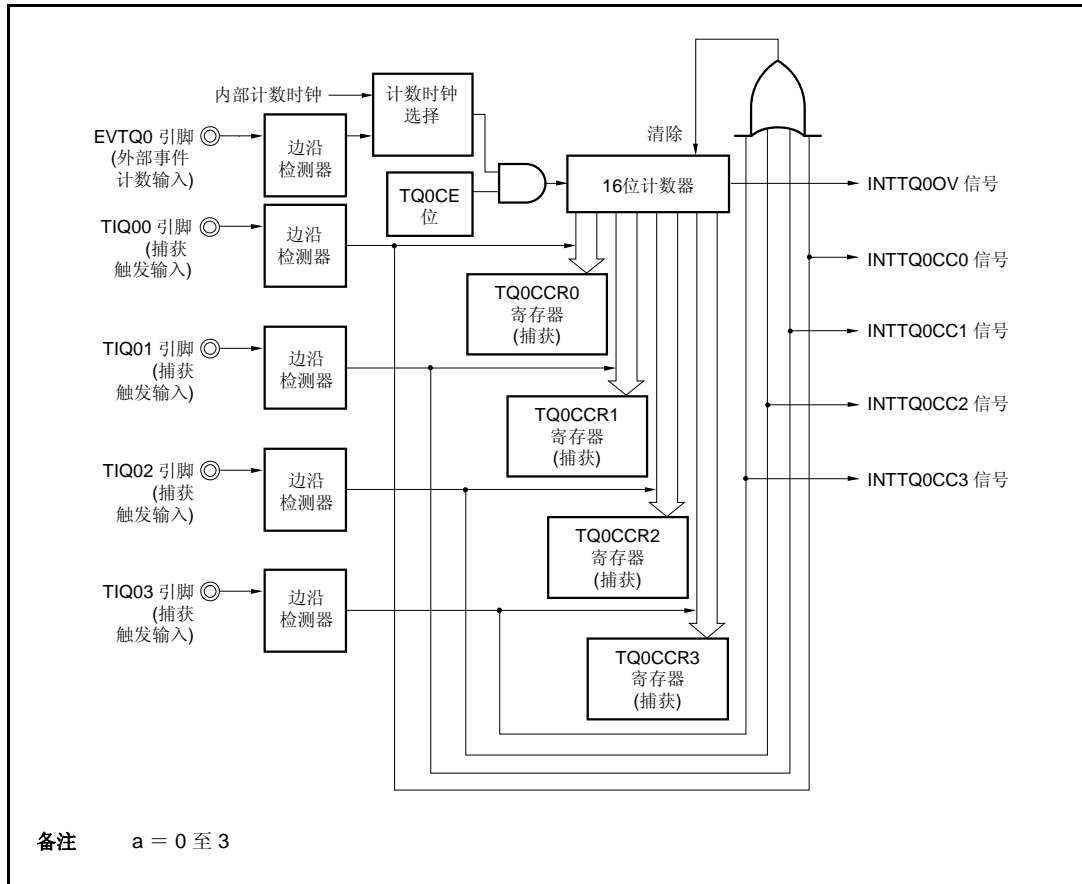
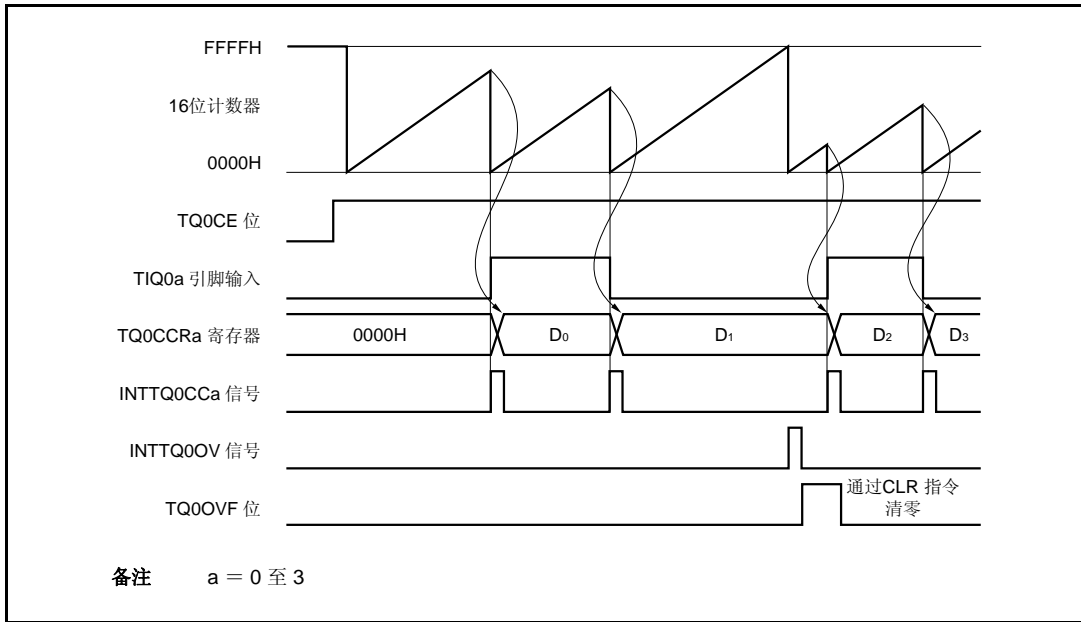


图 7-41. 脉冲宽度测量模式的基本时序



当 TQ0CE 位被置为 1 时，16 位计数器开始计数。此后当检测到 TIQ0a 引脚输入信号的有效边沿时，16 位计数器的计数值将被存储在 TQ0CCRa 寄存器中，16 位计数器清零为 0000H，并生成一个捕获中断请求信号（INTTQ0CCa）。脉冲宽度计算如下。

<R>

$$\text{脉冲宽度} = \text{捕获值} \times \text{计数时钟周期}$$

即使 16 位计数器一直计数到 FFFFH，TIQ0m 引脚输入信号一直没有有效边沿，仍然会在下一个计数时钟时产生溢出中断请求信号（INTTQ0OV），且计数器清零为 0000H，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。

若溢出标志设置为 1，则脉冲宽度可计算如下。

<R>

$$\text{脉冲宽度} = (10000H \times \text{TQ0OVF 位被置为 1 的次数} + \text{捕获值}) \times \text{计数时钟周期}$$

备注 a = 0 至 3

图 7-42. 脉冲宽度测量模式时的寄存器设置 (1/2)

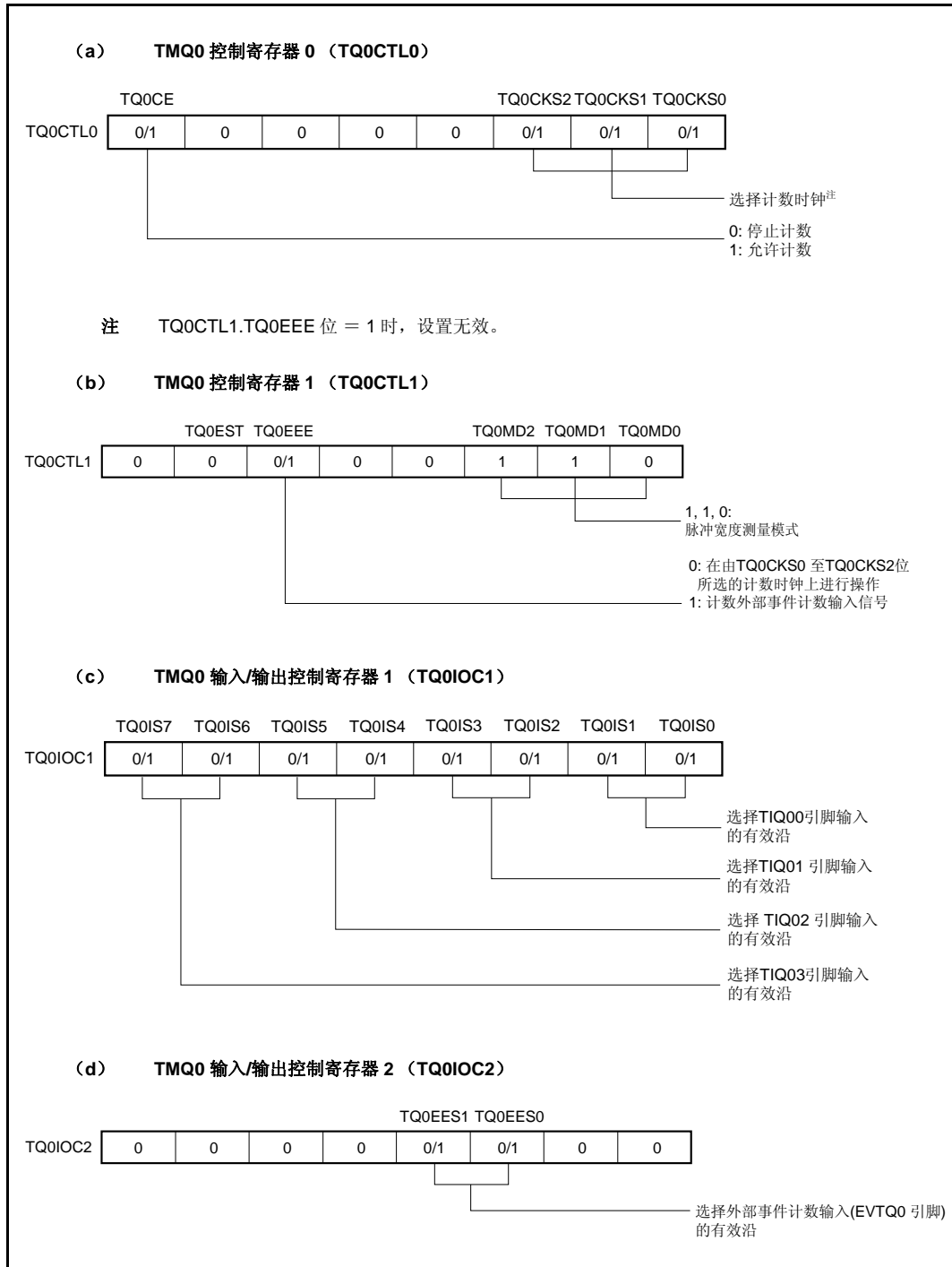
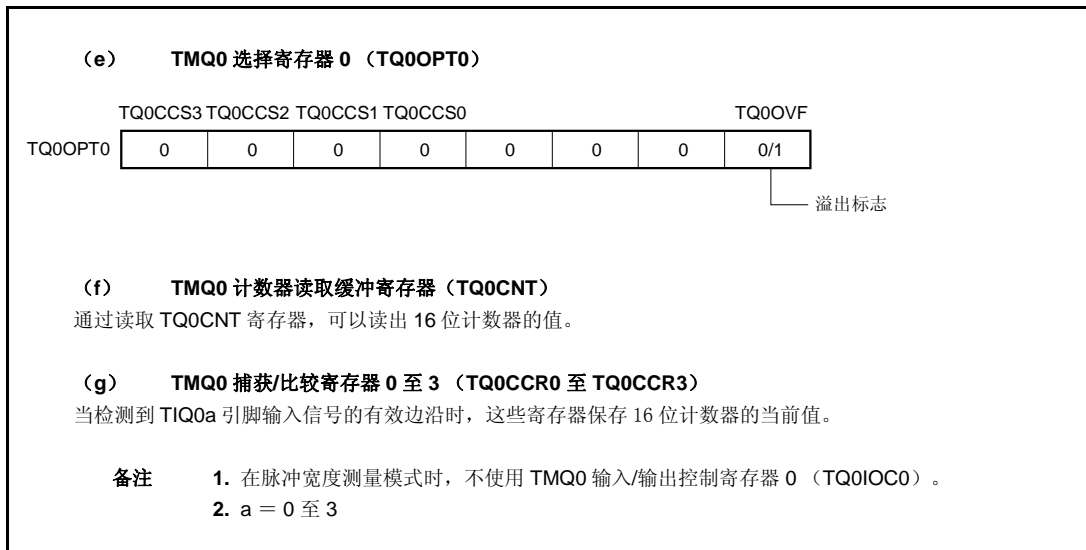
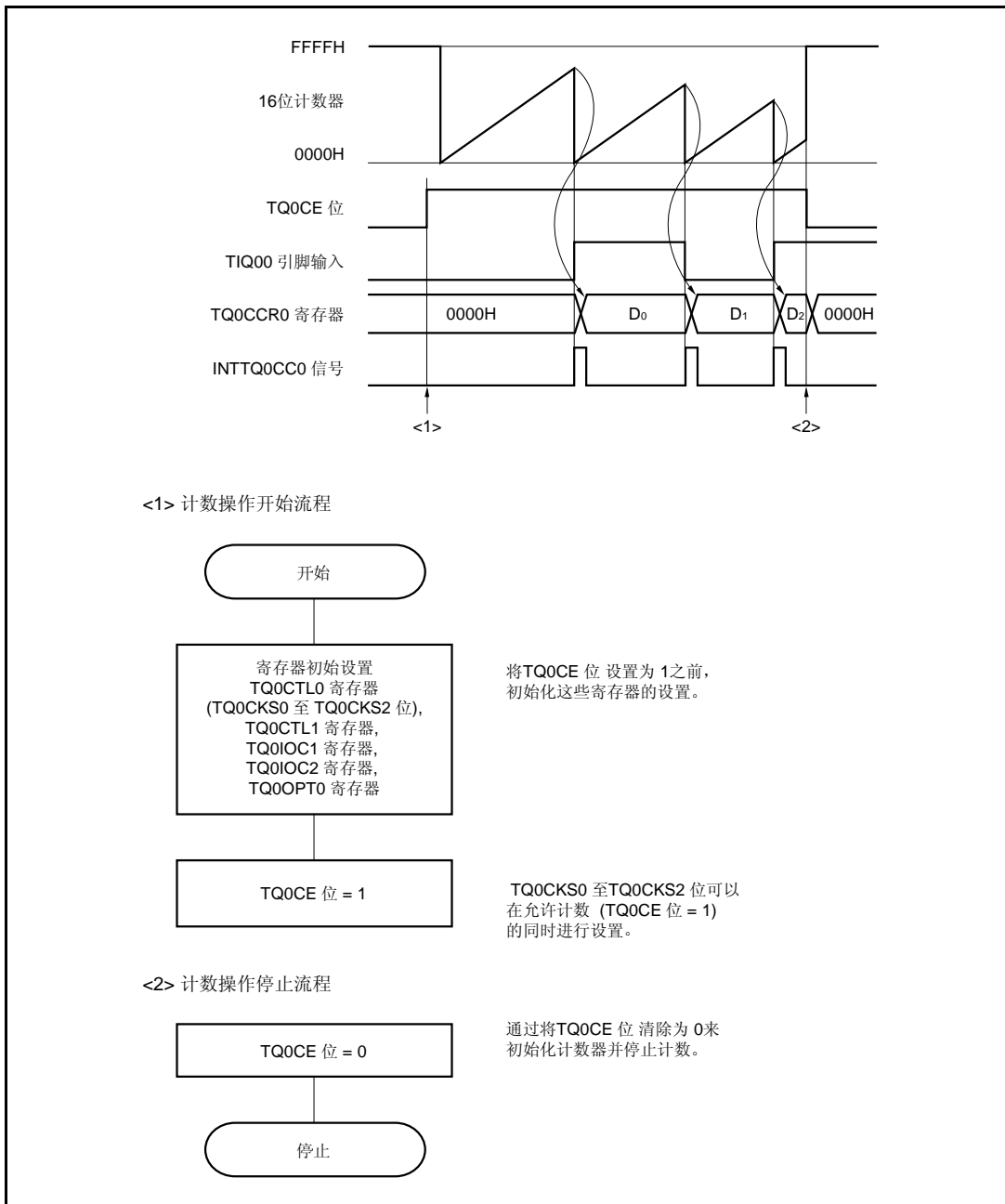


图 7-42. 脉冲宽度测量模式时的寄存器设置 (2/2)



(1) 脉冲宽度测量模式时的工作流程

图 7-43. 脉冲宽度测量模式时的软件处理流程



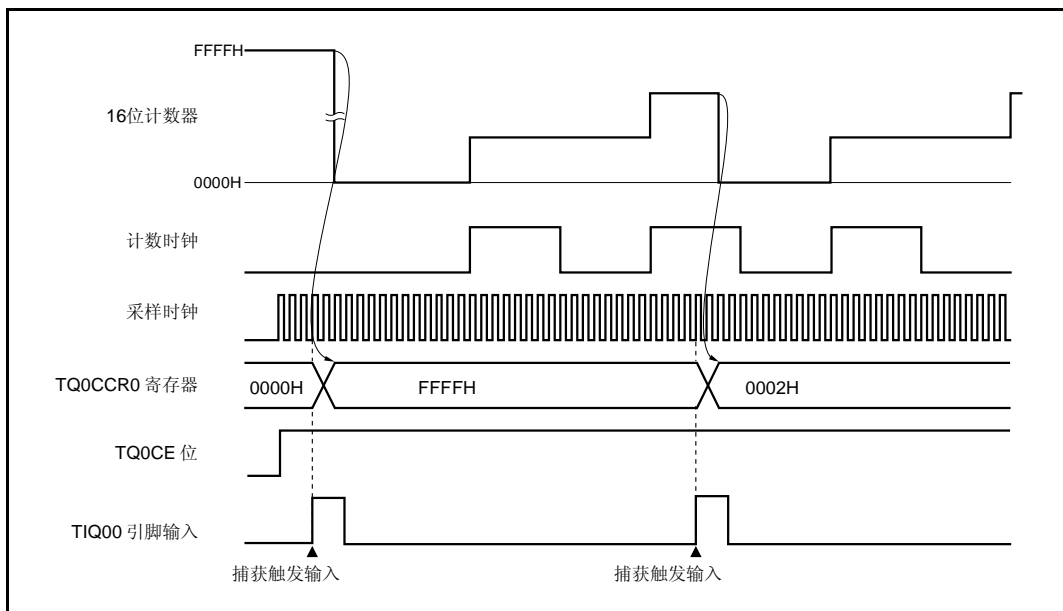
(2) 脉冲宽度测量模式时的工作时序

(a) 清除溢出标志

在 TQ0OVF 位为 1 时读取该位后，使用 CLR 指令将 TQ0OVF 位清零可以将溢出标志清零，而在 TQ0OVF 位为 1 时读取该位后，通过将 8 位数据（位 0 为 0）写入 TQ0OPT0 寄存器也可以将溢出标志清零。

(3) 注

如果将低速时钟选择为计数时钟，那么在 TQ0CTL0.TQ0CE 位被置为 1 之后（a = 0 至 3），如果捕获触发器被立即输入，那么可以被捕获到 TQ0CCRa 寄存器中的是 FFFFH，而不是 0000H。



8.1 概述

- 间隔定时功能
- 8 种时钟可选
- 16 位计数器 × 1
(定时器计数操作过程中不可对 16 位计数器进行读取。)
- 比较寄存器 × 1
(定时器计数操作过程中不可向比较寄存器写入数据。)
- 比较匹配中断 × 1

定时器 M 只支持清零&启动模式。不支持自由运行定时器模式。

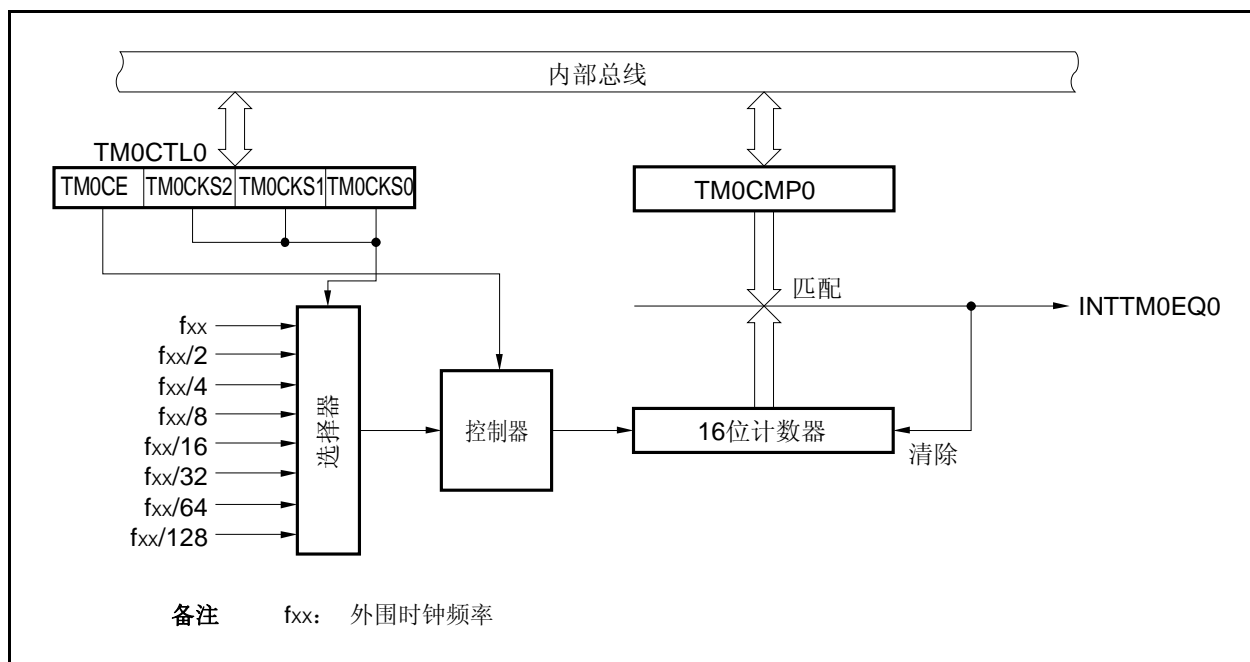
8.2 配置

TMM0 包括以下硬件：

表 8-1. TMM0 的配置

项目	配置
定时器寄存器	16 位计数器
寄存器	TMM0 比较寄存器 0 (TM0CMP0)
控制寄存器	TMM0 控制寄存器 0 (TM0CTL0)

图 8-1. TMM0 框图

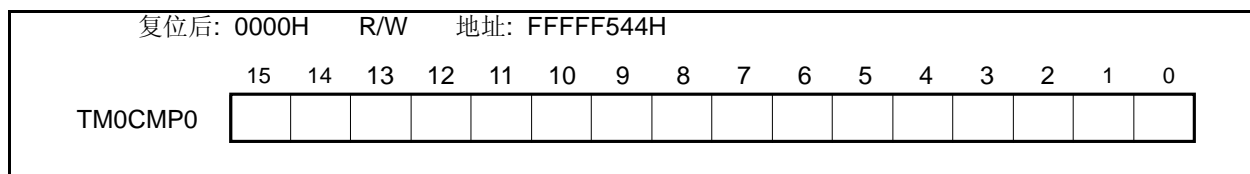


(1) 16 位计数器

该计数器是一个计数内部时钟的 16 位计数器。
该计数器不可以被读写。

(2) TMM0 比较寄存器 0 (TM0CMP0)

TM0CMP0 寄存器是一个 16 位的比较寄存器。
该寄存器可按 16 位进行读写。
复位输入将寄存器清为 0000H。
总是可以使用软件将相同的值写入 TM0CMP0 寄存器。
禁止在 TMM0 操作期间 (TM0CTL0.TMOCE 位 = 1) 对 TM0CMP0 寄存器进行重写。



8.3 控制寄存器

(1) TMM0 控制寄存器 0 (TM0CTL0)

TM0CTL0 寄存器是控制定时器 TMM0 的操作的 8 位寄存器。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

总是可以使用软件将相同的值写入 TM0CTL0 寄存器。

复位后: 00H R/W 地址: FFFFFFF540H

	<7>	6	5	4	3	2	1	0
TM0CTL0	TM0CE	0	0	0	0	TM0CKS2	TM0CKS1	TM0CKS0

TM0CE	内部设置操作允许/禁止 规范
0	禁止TMM0 操作(16位计数器异步复位)
1	允许TMM0操作。开始操作时钟供给。开始 TMM0 操作。
TMM0的内部时钟控制和内部电路复位与TM0CE位异步。 TM0CE 位清除为 0时, TMM0的内部时钟停止 (固定为低电平), 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} /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

注意事项 1. TM0CE 位=0 时, 设置 TM0CKS2 至 TM0CKS0 位。

然而, 当将 TM0CE 的值从 0 改变为 1 时, 不可以同时设置 TM0CKS2 到 TM0CKS0 位的值。

2. 一定要将位 3 至 6 清除为“0”。

备注 f_{xx}: 外围时钟频率

8.4 操作

8.4.1 间隔定时器模式

在间隔定时器模式中，如果 `TM0CTL0.TM0CE` 位被设为 1，那么中断请求信号 (`INTTM0EQ0`) 就会在 `TM0CMP0` 寄存器所设定的间隔中生成。

图 8-2. 间隔定时器的配置

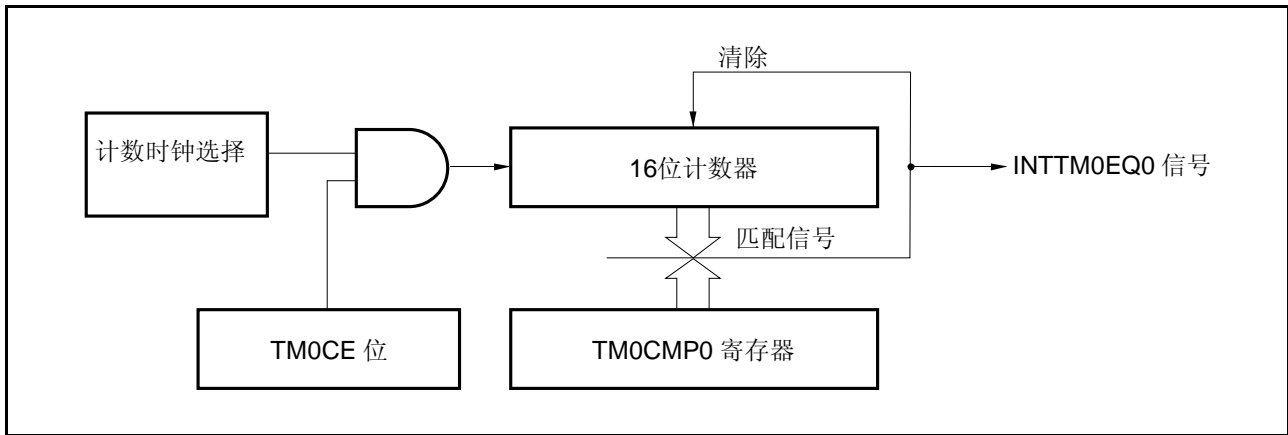
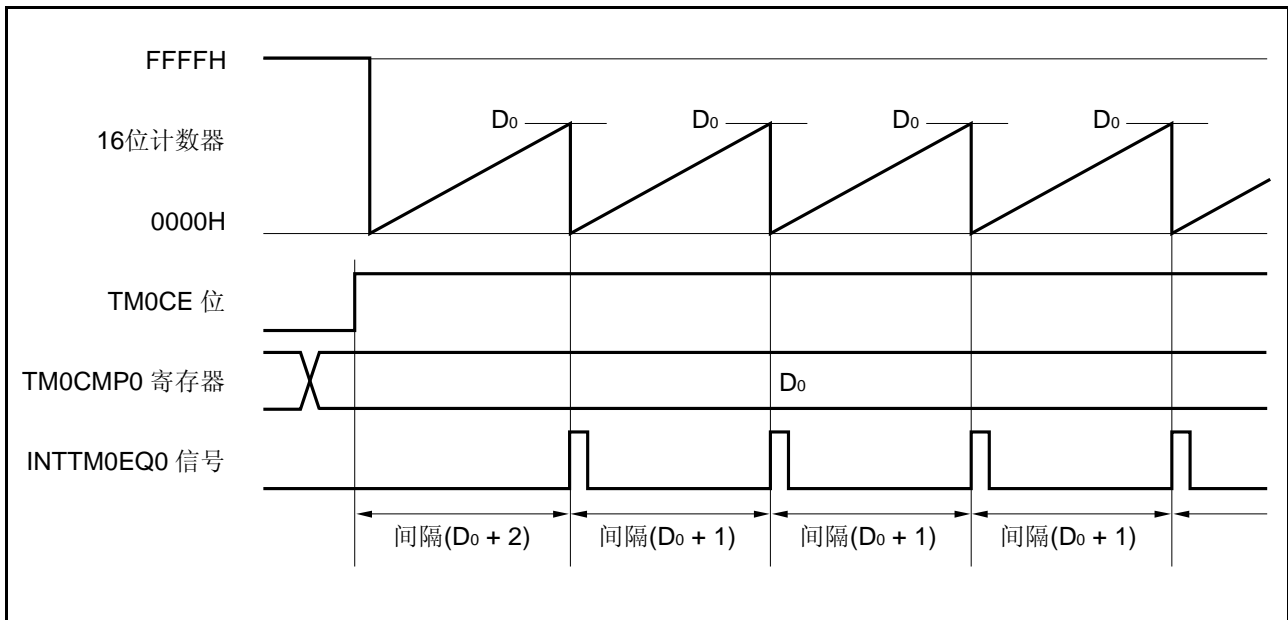


图 8-3. 间隔定时器工作的基本时序



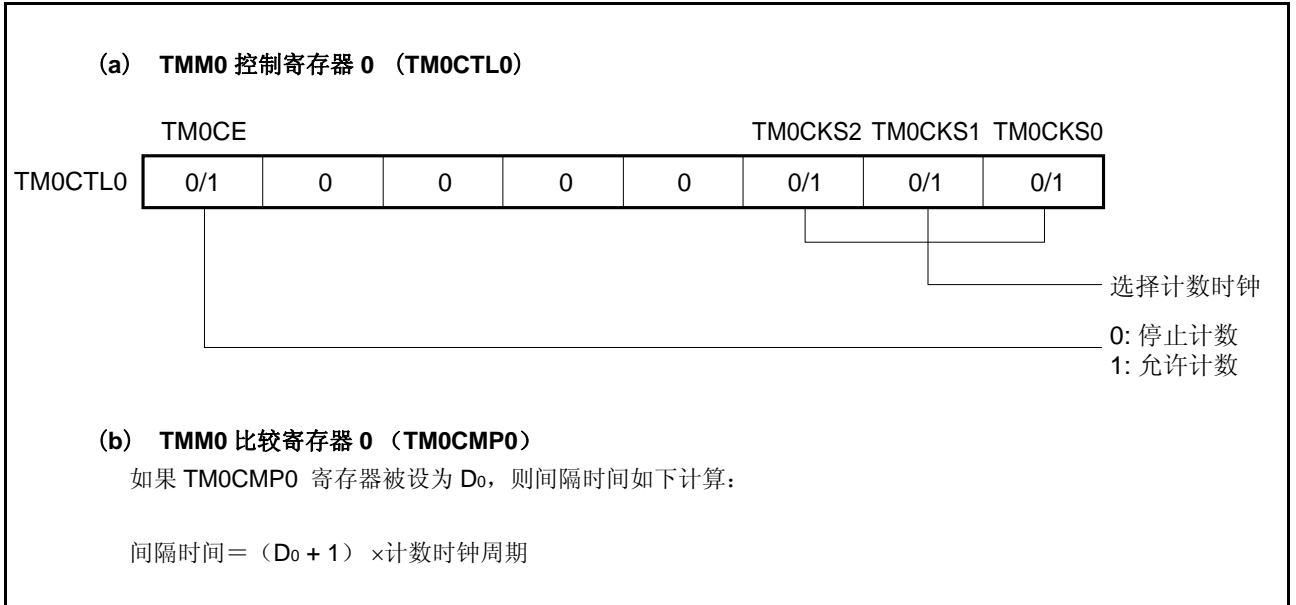
当 **TM0CE** 位被设置为 1 时，16 位计数器会与计数时钟同步将计数值从 **FFFFH** 清零为 **0000H**，并开始计数。

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**)。

时间间隔可用下式计算。

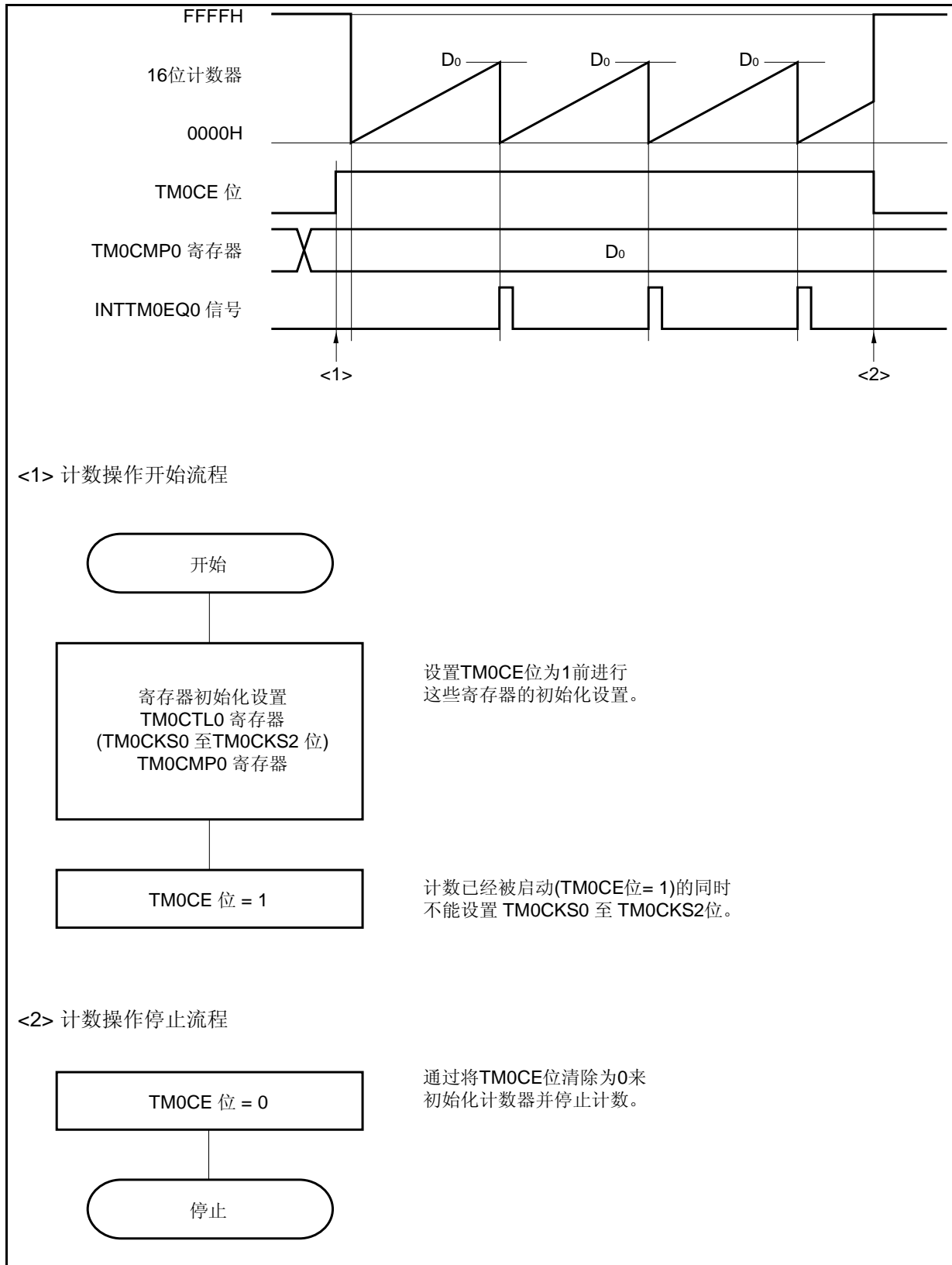
$$\text{时间间隔} = (\text{TM0CMP0 寄存器的设定值} + 1) \times \text{计数时钟周期}$$

图 8-4. 间隔定时器模式操作时的寄存器设置



(1) 间隔定时器模式工作流程

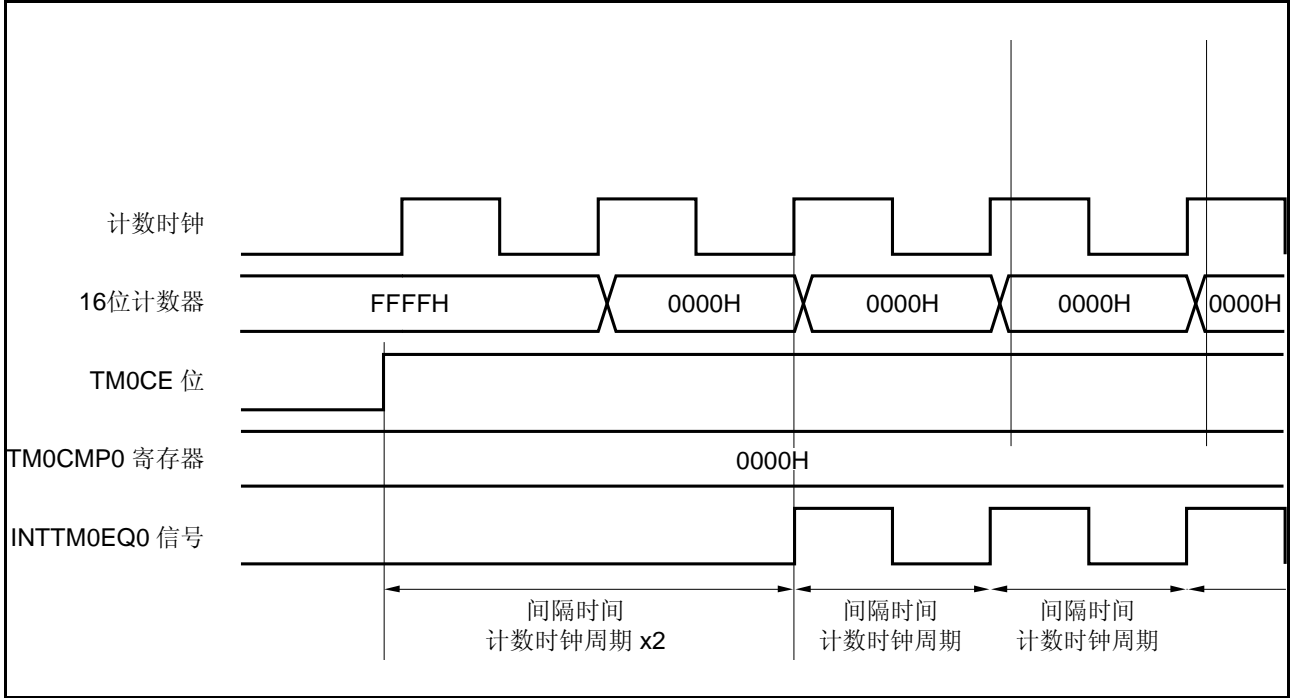
图 8-5. 间隔定时器模式下的软件处理流程



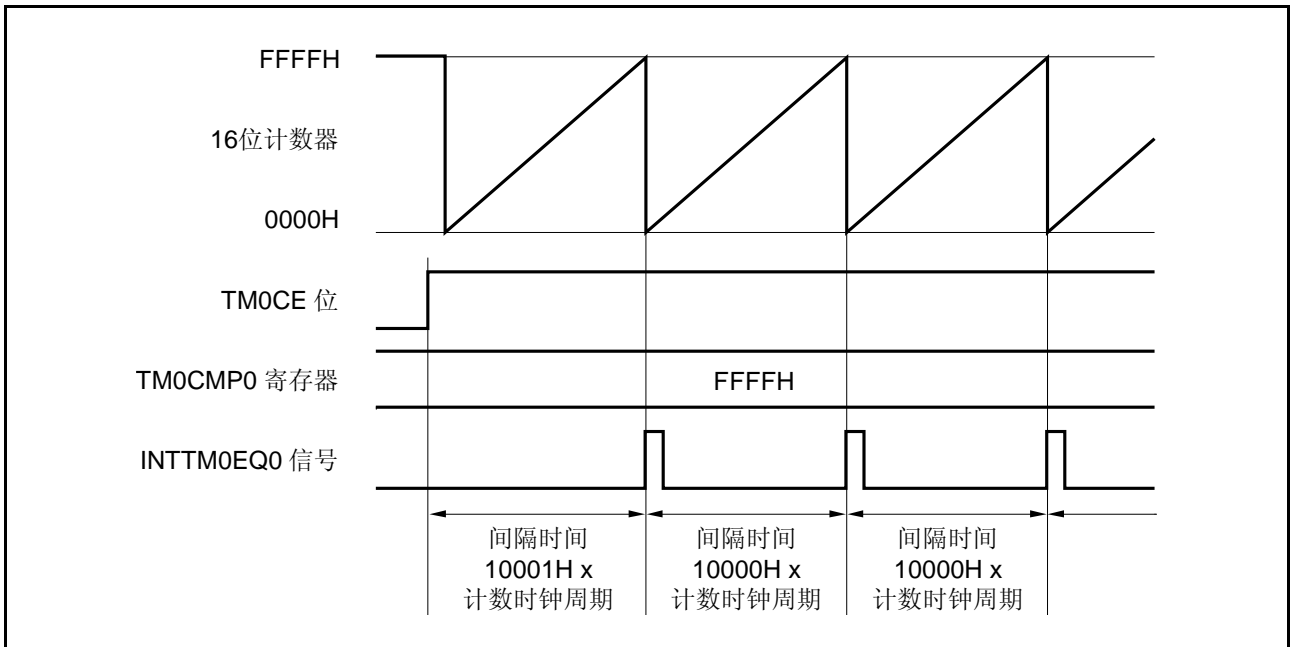
(2) 间隔定时器模式工作时序

(a) **TM0CMP0 寄存器设置为 0000H 时的操作**

如果 TM0CMP0 寄存器被设置为 0000H，那么每个计数时钟都会产生 INTTM0EQ0 信号。
16 位计数器的值总是 0000H。

(b) **TM0CMP0 寄存器被设置为 FFFFH 时的操作**

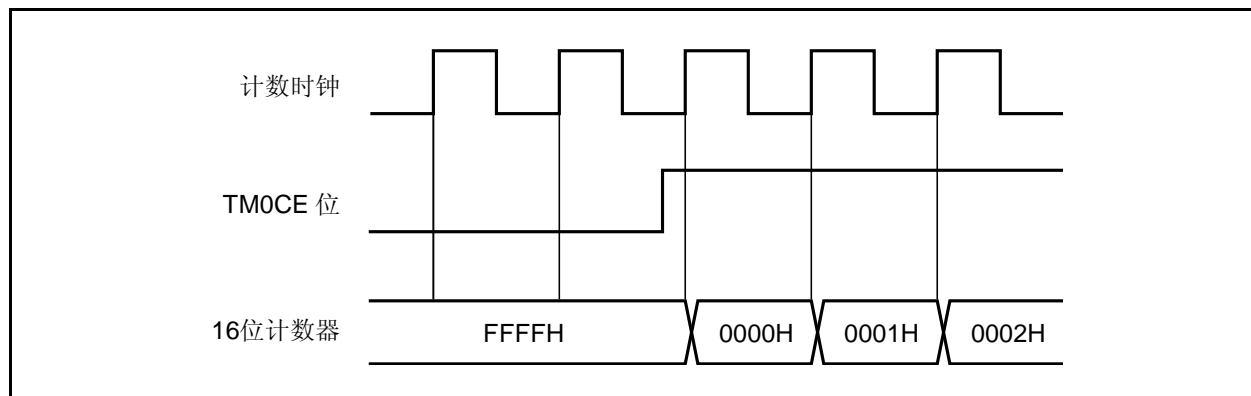
如果 TM0CMP0 寄存器被设置为 FFFFH，16 位计数器可以计数直至 FFFFH。该计数器在下一个计数时刻同步被清零。INTTM0EQ0 信号也将产生。



8.5 注意事项

(1) 定时器启动误差

在 `TM0CTL0.TM0CE` 位设为 1 且 TMM0 被启动后，需要用一個时钟的时间来生成第一个比较匹配中断请求信号 (`INTTM0EQ0`)。这是因为在 `TM0CE` 位 = 0 且 TMM0 同步开始计数时钟时，16 位计数器的值为 `FFFFH`。

(2) TMM0 工作时，禁止重写 `TM0CMP0` 和 `TM0CTL0` 寄存器的值。

如果在 `TM0CTL0.TM0CE` 位 = 1 时对这两个寄存器进行重写，那么操作结果将得不到保证。

如果这些寄存器被错误地改写，先将 `TM0CE` 位清零，然后重置这些寄存器。

第九章 马达控制功能

9.1 功能概述

定时器 Q1 (TMQ1) 和 TMQ1 选项 (TMQOP1) 可用作控制马达的变频器功能。它可以使用定时器 P1 (TMP1) 来执行调制功能，并且当 TMQ1 的值与 TMP1 的值匹配时，可以启动 A/D 转换器 0 和 1 的 A/D 转换。以下操作可以用于实现马达控制功能。

- 16 位精度的 6 相 PWM 输出功能（带有死区时间，实现上臂和下臂调制）。
- 定时器调制操作功能（使用 TMP1 时可调）。
- 周期设置功能（在峰值或谷值中断操作期间可以改变周期）。
- 比较寄存器重写：随时重写，批量重写，或间隙重写（在 TMQ1 操作过程中可选）。
- 中断及传送选择功能。
- 死区时间设置功能。
- A/D 转换器 0 和 1 的 A/D 触发时序功能（可以产生四种类型的时序）。
- 0%输出和 100%输出可用。
- 通过波峰中断和波谷中断可以选择 0%输出和 100%输出。
- 强行输出停止功能。
 - 通过外部引脚输入（TOQ1OFF、TOP2OFF、TOP3OFF、TOQH0OFF）检测到有效沿时。
 - 通过时钟监视器功能检测到主时钟振荡停止时。

9.2 配置

马达控制功能包含以下硬件：

项目	配置
定时器寄存器	死区时间计数器 m
比较寄存器	TMQ1 死区时间比较寄存器 (TQ1DTC 寄存器)
控制寄存器	TMQ1 选项寄存器 0 (TQ1OPT0) TMQ1 选项寄存器 1 (TQ1OPT1) TMQ1 选项寄存器 2 (TQ1OPT2) TMQ1 选项寄存器 3 (TQ1OPT3) TMQ1 I/O 控制寄存器 3 (TQ1IOC3) 高阻输出控制寄存器 0, 1 (HZA1CTL0, HZAaCTL1)

备注 m = 0 至 3
 a = 0, 1

- 通过使用 TMQ1 (TOQ11, TOQ12, TOQ13) 的输出可以产生带有死区时间的 6 相 PWM 输出。
- 6 相 PWM 输出的输出电平可以进行单独设置。
- TMQ1 的 16 位定时器 / 计数器递增/递减计数三角波。当定时器 / 计数器下溢且发生周期匹配时会生成一个中断。然而，中断生成被限制为 31 次。
- TMP1 可以与 TMQ1 (定时器调制操作功能) 同时执行计数操作。TMP1 可以按四种方式进行设置，它可以生成两种类型的 A/D 触发源 (INTTP1CC0 和 INTTP1CC1) 以及两种类型的中断：下溢中断 (INTTQ1OV) 和周期匹配中断 (INTTQ1CC0)。

图 9-1. 马达控制的框图

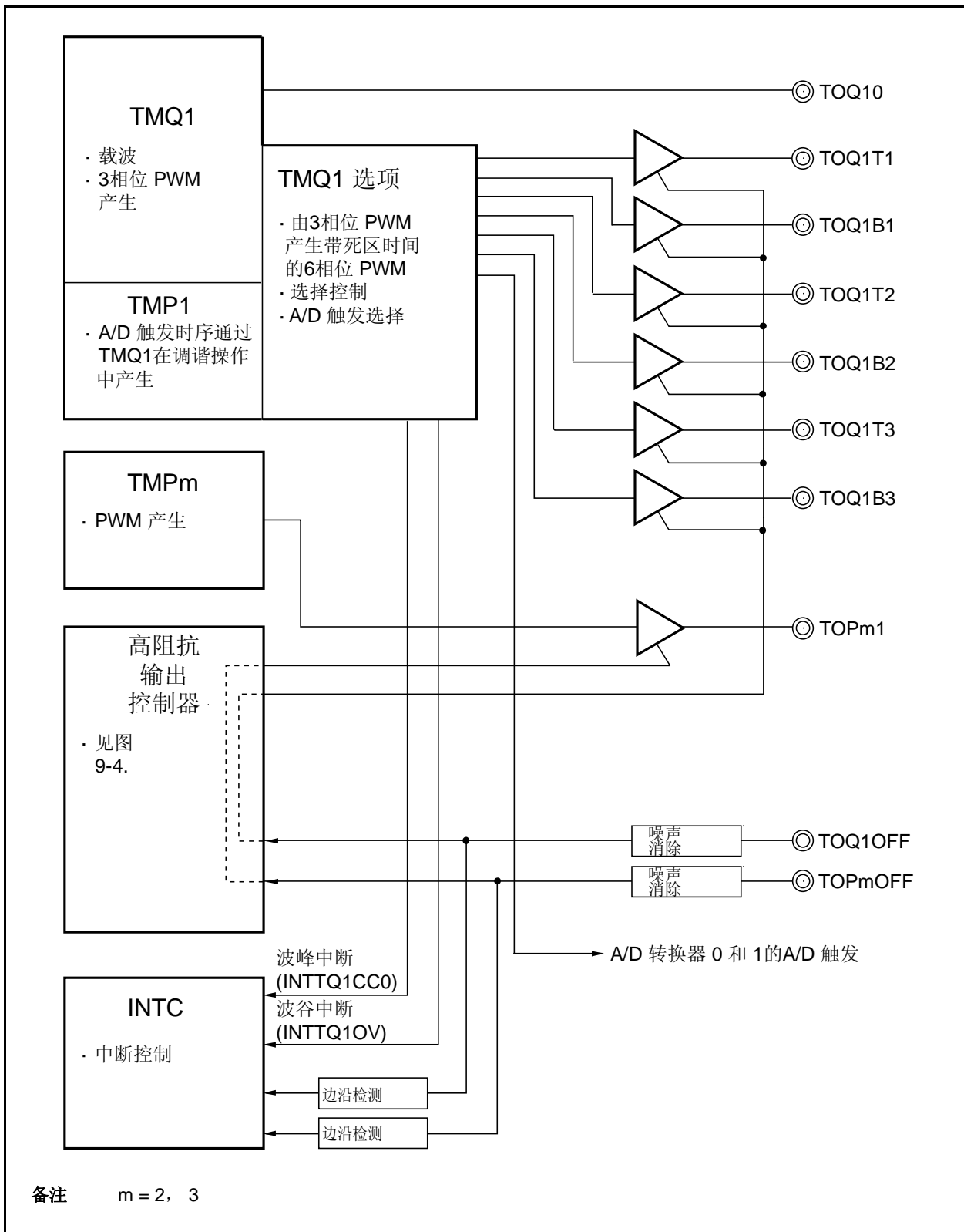
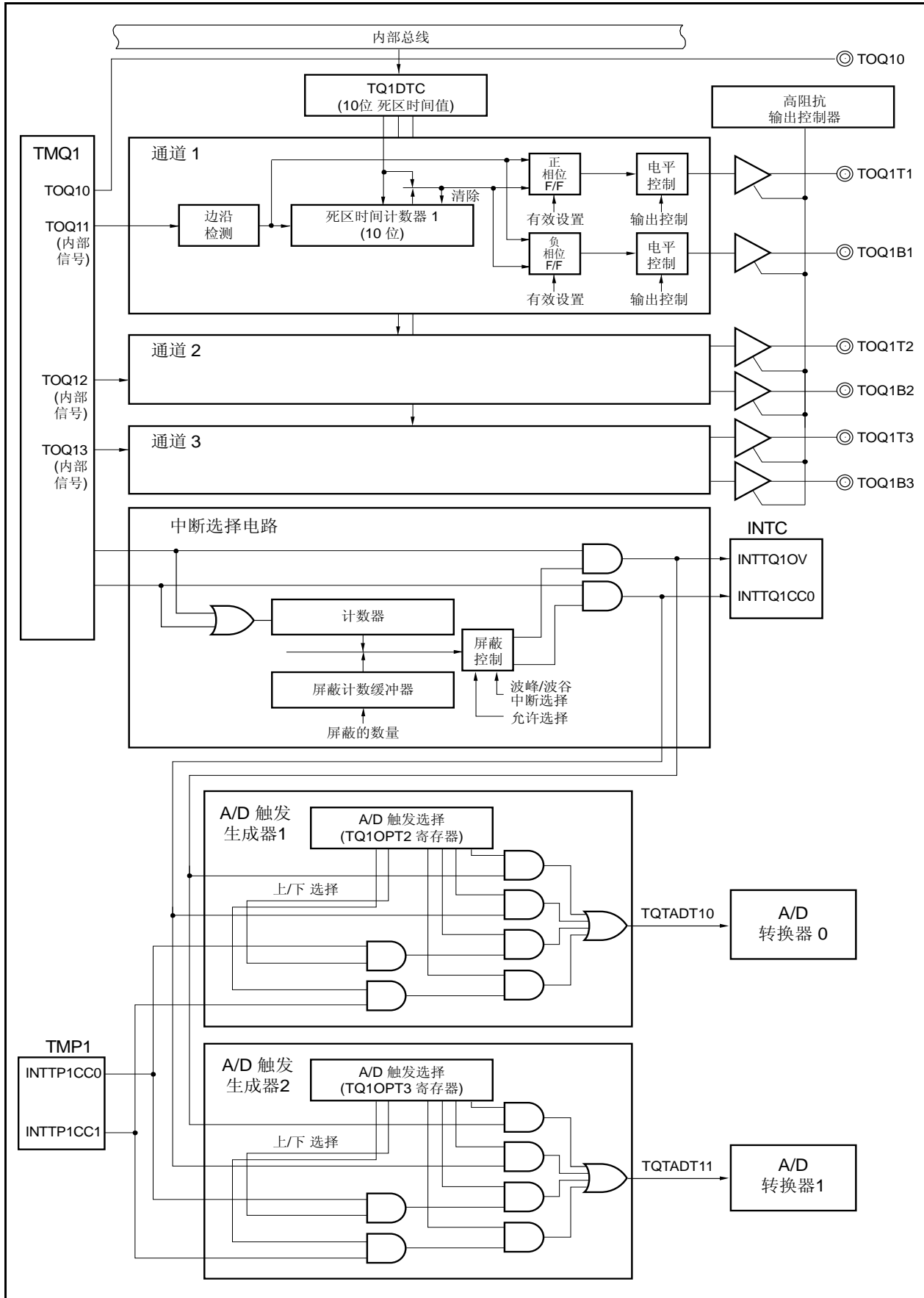


图 9-2. TMQ1 选项



(1) TMQ1 死区时间比较寄存器 (TQ1DTC)

TQ1DTC 寄存器是用于指定死区时间的 10 位比较寄存器。

当 TQ1CTL0.TQ1CE 位 = 1 时禁止重写该寄存器。

该寄存器可按 16 位进行读写。

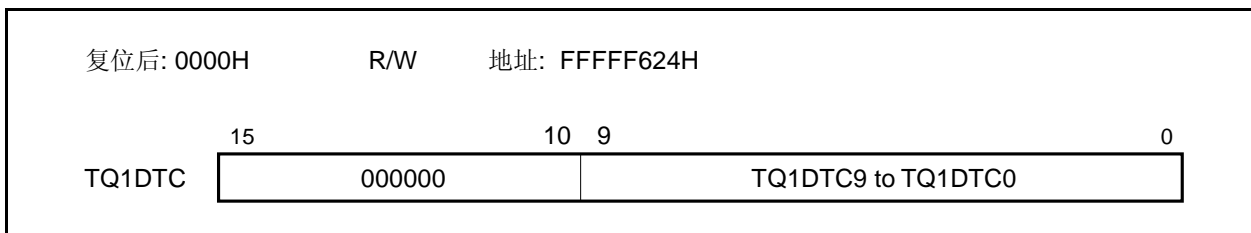
复位设置将寄存器清为 0000H。

<R>

注意事项 要产生死区时间，需将 TQ1DTC 寄存器设置为 1 或更大的数值。

操作停止 (TQ1CTL0.TQ1CE 位 = 0) 时，不会产生死区时间期间并且 TOQ1T1 至 TOQ1T3 引脚以及 TOQ1B1 至 TOQ1B3 引脚的输出电平会处于初始状态。因此考虑到系统保护，在操作停止之前，将 TOQ1T1 至 TOQ1T3 引脚以及 TOQ1B1 至 TOQ1B3 引脚设置为高阻状态，或设置这些引脚的输出电平并将它们转换为端口模式。

如果不需要死区时间，将 TQ1DTC 寄存器设置为 0。

**(2) 死区时间计数器 1 至 3**

死区时间计数器是用于计数死区时间的 10 位计数器。

这些寄存器会在 TOQ1m 输出信号的上升沿或下降沿处通过 TMQ1 来清除或计数，并且它们会在其计数值与 TQ1DTC 寄存器的值匹配时被清除或停止。这些计数器的计数时钟同 TMQ1 的 TQ1CTL0.TQ1CKS2 至 TQ1CTL0.TQ1CKS0 位所设置的计数时钟相同。

- 备注**
1. TQ1OPT2.TQ1DTM 位 = 1 时的操作是不同的。详细信息，请参见 9.4.2 (4) 自动死区时间宽度缩小功能 (TQ1OPT2.TQ1DTM 位 = 1)。
 2. m = 1 至 3

9.3 控制寄存器

(1) TMQ1 选项寄存器 0 (TQ1OPT0)

TQ1OPT0 寄存器是用来控制定时器 Q1 选项功能的 8 位寄存器。

该寄存器可进行 8 位或 1 位读写。但是, TQ1CUF 位只读。

系统复位后该寄存器被设为 00H。

注意事项 TQ1CMS 和 TQ1CUF 位只能在 6 相 PWM 输出模式下进行设置。单独使用 TMQ1 时一定要将这些位清零。

复位后: 00H R/W 地址: FFFFF605H

	7	6	5	4	3	<2>	<1>	<0>
TQ1OPT0	0	0	0	0	0	TQ1CMS	TQ1CUF	TQ1OVF ^注

TQ1CMS	比较寄存器重写模式选择
0	批量重写模式(传送操作)
1	随时重写模式
<ul style="list-style-type: none"> · 只有设置6相位 PWM 输出模式(TQ1CTL1.TQ1MD2 至 TQ1CTL1.TQ1MD0 位 = 111时)时 TQ1CMS 位才有效。其它任何模式下都将 TQ1CMS 位清 0。 · 定时器操作(TQ1CTL0.TQ1CE 位 = 1时)时 TQ1CMS位可被重写。 · 以下寄存器在批量写入模式下被重写: TQ1CCR0 至 TQ1CCR3, TP1CCR0, TP1CCR1以及 TQ1OPT1 寄存器 	

TQ1CUF	定时器 Q1的向上计数/向下计数 标志
0	定时器 Q1 正在向上计数。
1	定时器 Q1 正在向下计数。
只有设置6相位 PWM 输出模式(TQ1CTL1.TQ1MD2 至 TQ1CTL1.TQ1MD0 位 = 111时)时 TQ1CUF位才有效。	

注 关于 TQ1OVF 位的详细情况, 参见第七章 16 位定时器/事件 计数器 Q (TMQ)。

注意事项 一定要将位 7 至 3 清除为“0”。

(2) TMQ1 选项寄存器 1 (TQ1OPT1)

TQ1OPT1 寄存器是用来控制由定时器 Q1 选项功能所生成的中断请求信号的 8 位寄存器。

<R> TQ1OPT1 寄存器生成的信号输出到中断选择电路、A/D 触发发生器 1 以及 A/D 触发发生器 2，如图 9-2。

该寄存器可以在 TQ1CTL0.TQ1CE 位为 1 时进行重写。

根据 TQ1OPT0.TQ1CMS 位的设置有两种重写模式（批量写入模式和随时写入模式）可选择。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

复位后: 00H R/W 地址: FFFFF620H

	<7>	<6>	5	4	3	2	1	0
TQ1OPT1	TQ1ICE	TQ1IOE	0	TQ1ID4	TQ1ID3	TQ1ID2	TQ1ID1	TQ1ID0

<R>

TQ1ICE	允许波峰中断(INTTQ1CC0 信号) ^注
0	不使用 INTTQ1CC0 信号 (不要将其用作中断选择的计数信号)。
1	使用 INTTQ1CC0 信号 (将其用作中断选择的计数信号)。

<R>

TQ1IOE	允许波谷中断 (INTTQ1OV 信号) ^注
0	不使用 INTTQ1OV 信号 (不要将其用作中断选择的计数信号)。
1	使用 INTTQ1OV 信号 (将其用作中断选择的计数信号)。

TQ1ID4	TQ1ID3	TQ1ID2	TQ1ID1	TQ1ID0	中断的次数
0	0	0	0	0	未选择 (所有中断都被输出)
0	0	0	0	1	1 次屏蔽 (1/2 中断被输出)
0	0	0	1	0	2 次屏蔽 (1/3 中断被输出)
0	0	0	1	1	3 次屏蔽 (1/4 中断被输出)
:	:	:	:	:	:
1	1	1	0	0	28 次屏蔽 (1/29 中断被输出)
1	1	1	0	1	29 次屏蔽 (1/30 中断被输出)
1	1	1	1	0	30 次屏蔽 (1/31 中断被输出)
1	1	1	1	1	31 次屏蔽 (1/32 中断被输出)

<R> **注** 使用波峰中断 (INTTQ1CC0 信号) 和波谷中断 (INTTQ1OV 信号) 作为中断选择的计数信号或 A/D 触发信号时, 需将该信号设置为 1。
A/D 触发信号在选择中断时序处产生。

(3) TMQ1 选项寄存器 2 (TQ1OPT2)

TQ1OPT2 寄存器是用来控制定时器 Q1 选项功能的 8 位寄存器。

该寄存器可以在 TQ1CTL0.TQ1CE 位为 1 时进行重写。然而，在 TQ1CE 为 1 时禁止重写 TQ1DTM 位。相同的值可以进行重写。

该寄存器可进行 8 位或 1 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: FFFFF621H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TQ1OPT2	TQ1RDE	TQ1DTM	TQ1ATM03	TQ1ATM02	TQ1AT03	TQ1AT02	TQ1AT01	TQ1AT00

TQ1RDE	允许传送选择
0	不选择传送 (传送时序会在每次的波峰和波谷处产生)。
1	在TQ1OPT1 寄存器设置的中断选择的同一间隔处选择传送。

TQ1DTM	死区时间计数器操作模式选择
0	死区时间计数器正常向上计数，如果TMQ1的TOQ1m输出处在窄的间隔(TOQ1m 输出宽度< 死区时间宽度), 死区时间计数器被清除并重新向上计数。
1	死区时间计数器正常向上计数，如果TMQ1的TOQ1m输出处在窄的间隔(TOQ1m 输出宽度< 死区时间宽度),死区时间计数器向下计数并且死区控制宽度被自动变窄。
定时器操作期间禁止重写 TQ1DTM 位。如果被错误重写，通过将TQ1CE 位清0停止定时器操作，并重设TQ1DTM 位。	

注意事项 1. 使用中断选择 (TQ1OPT1.TQ1ID4 至 TQ1OPT1.TQ1ID0 位设置为非 00000) 时，确保将 TQ1RDE 位设置为 1。

因此，中断和传送可以同时生成。中断和传送不能分开设置。如果中断和传送分开设置 (TQ1RDE 位 = 0)，则传送将不能正常进行。

2. 要产生死区时间期间，需将 TQ1DTC 寄存器设置为 1 或更大的数值。

操作停止 (TQ1CTL0.TQ1CE 位 = 0) 时，不会产生死区时间期间并且 TOQ1T1 至 TOQ1T3 引脚以及 TOQ1B1 至 TOQ1B3 引脚输出电平会处于初始状态。因此考虑到系统保护，在操作停止之前，将 TOQ1T1 至 TOQ1T3 引脚以及 TOQ1B1 至 TOQ1B3 引脚设置为高阻状态，或设置这些引脚的输出电平并将它们转换为端口模式。

如果不需要死区时间期间，将 TQ1DTC 寄存器设置为 0。

备注 m = 1 至 3

<R>

TQ1ATM03	TQ1ATM03 模式选择
0	死区时间计数器正在向上计数时输出TTP1CC1 中断的 A/D触发信号(TQTADT10)。
1	死区时间计数器正在向下计数时输出TTP1CC1 中断的 A/D触发信号(TQTADT10)。

TQ1ATM02	TQ1ATM02 模式选择
0	死区时间计数器正在向上计数时输出 INTTP1CC0 中断的 A/D触发信号(TQTADT10) 。
1	死区时间计数器正在向下计数时输出 INTTP1CC0 中断的 A/D触发信号(TQTADT10) 。

TQ1AT03 ^注	A/D 触发输出控制3
0	禁止 INTTP1CC1 中断的A/D触发信号(TQTADT10)输出。
1	允许INTTP1CC1 中断的A/D触发信号(TQTADT10)输出。

TQ1AT02 ^注	A/D 触发输出控制2
0	禁止 INTTP1CC0 中断的A/D触发信号(TQTADT10)输出。
1	允许INTTP1CC0 中断的A/D触发信号(TQTADT10)输出。

TQ1AT01 ^注	A/D 触发输出控制1
0	禁止 INTTP1CC0 中断的A/D触发信号(TQTADT10)输出 (波峰中断)。
1	允许 INTTP1CC0 中断的A/D触发信号(TQTADT10)输出 (波峰中断)。

TQ1AT00 ^注	A/D 触发输出控制0
0	禁止 INTTQ1OV 中断的A/D触发信号(TQTADT10)输出 (波谷中断)。
1	允许 INTTQ1OV 中断的A/D触发信号(TQTADT10)输出 (波谷中断)。

注 关于 TQ1AT03 至 TQ1AT00 位设置的详细信息，请参见第十一章 **A/D 转换器 0 和 1**。

(4) TMQ1 选项寄存器 3 (TQ1OPT3)

TQ1OPT3 寄存器是用来控制定时器 Q1 选项功能的 8 位寄存器。

该寄存器可以在 TQ1CTL0.TQ1CE 位为 1 时进行重写。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

复位后: 00H		R/W	地址: FFFFF623H							
			7	6	<5>	<4>	<3>	<2>	<1>	<0>
TQ1OPT3	0	0	TQ1ATM13	TQ1ATM12	TQ1AT13	TQ1AT12	TQ1AT11	TQ1AT10		
TQ1ATM13	TQ1ATM3 模式选择									
0	死区时间计数器正在向上计数时输出 INTTP1CC1 中断的 A/D 触发信号 (TQTADT11)。									
1	死区时间计数器正在向下计数时输出 INTTP1CC1 中断的 A/D 触发信号 (TQTADT11)。									
TQ1ATM12	TQ1ATM2 模式选择									
0	死区时间计数器正在向上计数时输出 INTTP1CC0 中断的 A/D 触发信号 (TQTADT11)。									
1	死区时间计数器正在向下计数时输出 INTTP1CC0 中断的 A/D 触发信号 (TQTADT11)。									
TQ1AT13 ^注	A/D 触发输出控制3									
0	禁止 INTTP1CC1 中断的 A/D 触发信号 (TQTADT11) 输出。									
1	允许 INTTP1CC1 中断的 A/D 触发信号 (TQTADT11) 输出。									
TQ1AT12 ^注	A/D 触发输出控制2									
0	禁止 INTTP1CC0 中断的 A/D 触发信号 (TQTADT11) 输出。									
1	允许 INTTP1CC0 中断的 A/D 触发信号 (TQTADT11) 输出。									
TQ1AT11 ^注	A/D 触发输出控制1									
0	禁止 INTTQ1CC0 中断的 A/D 触发信号 (TQTADT11) 输出 (波峰中断)。									
1	允许 INTTQ1CC0 中断的 A/D 触发信号 (TQTADT11) 输出 (波峰中断)。									
TQ1AT10 ^注	A/D 触发输出控制0									
0	禁止 INTTQ1OV 中断的 A/D 触发信号 (TQTADT11) 输出 (波谷中断)。									
1	允许 INTTQ1OV 中断的 A/D 触发信号 (TQTADT11) 输出 (波谷中断)。									

注 关于 TQ1AT13 至 TQ1AT10 位设置的详细信息，请参见第十一章 A/D 转换器 0 和 1。

(5) TMQ1 输入输出控制寄存器 3 (TQ1IOC3)

TQ1IOC3 寄存器是用来控制定时器 Q1 选项功能输出的 8 位寄存器。

若要从 TOQ1Tm 引脚中输出, 应先将 TQ1IOC0.TQ1OEm 位设为 1, 然后对 TQ1IOC3 寄存器进行设置。

只有在 TQ1CTL0.TQ1CE 位为 0 时, 才可以重写 TQ1IOC3 寄存器。

TQ1CTL0.TQ1CE 位为 1 时禁止对 TQ1IOC3 寄存器中的各位进行重写; 然而当 TQ1CTL0.TQ1CE 位为 1 时相同值可以被重写至 TQ1IOC3 寄存器的每个位中。

该寄存器可进行 8 位或 1 位读写。

复位将寄存器清为 A8H。

注意事项 在不同于 6 相 PWM 输出模式下使用定时器时, 应将 TQ1IOC3 寄存器设置为默认值 (A8H)。

备注 通过 TQ1IOC0 寄存器来设置 TOQ1Tm 引脚的输出电平。

复位后: A8H	R/W	地址: FFFFF622H							
		<7>	<6>	<5>	<4>	<3>	<2>	1	0
TQ1IOC3		TQ1OLB3	TQ1OEB3	TQ1OLB2	TQ1OEB2	TQ1OLB1	TQ1OEB1	0	0
	TQ1OLBm	TOQ1Bm 引脚输出电平的设置(m = 1 至 3)							
	0	禁止 TOQ1Bm 引脚输出翻转							
	1	允许 TOQ1Bm 引脚输出翻转							
	TQ1OEBm	TOQ1Bm 引脚输出(m = 1 至 3)							
	0	禁止 TOQ1Bm 引脚输出。 · TQ1OLBm 位 = 0 时, 从 TOQ1Bm 引脚输出低电平。 · TQ1OLBm 位 = 1 时, 从 TOQ1Bm 引脚输出高电平。							
	1	允许 TOQ1Bm 引脚输出。							

(a) 从 TOQ1Tm 和 TOQ1Bm 引脚中输出

TOQ1Tm 引脚输出可以通过 TQ1IOC0.TQ1OLm 和 TQ1IOC0.TQ1OEm 位来控制。TOQ1Bm 引脚输出可以通过 TQ1IOC3.TQ1OLBm 和 TQ1IOC3.TQ1OEBm 位来控制。

以下显示了 6 相 PWM 输出模式下各种设置的定时器输出。

<R>

图 9-3. TOQ1Tm 和 TOQ1Bm 引脚的输出控制（不带死区时间）

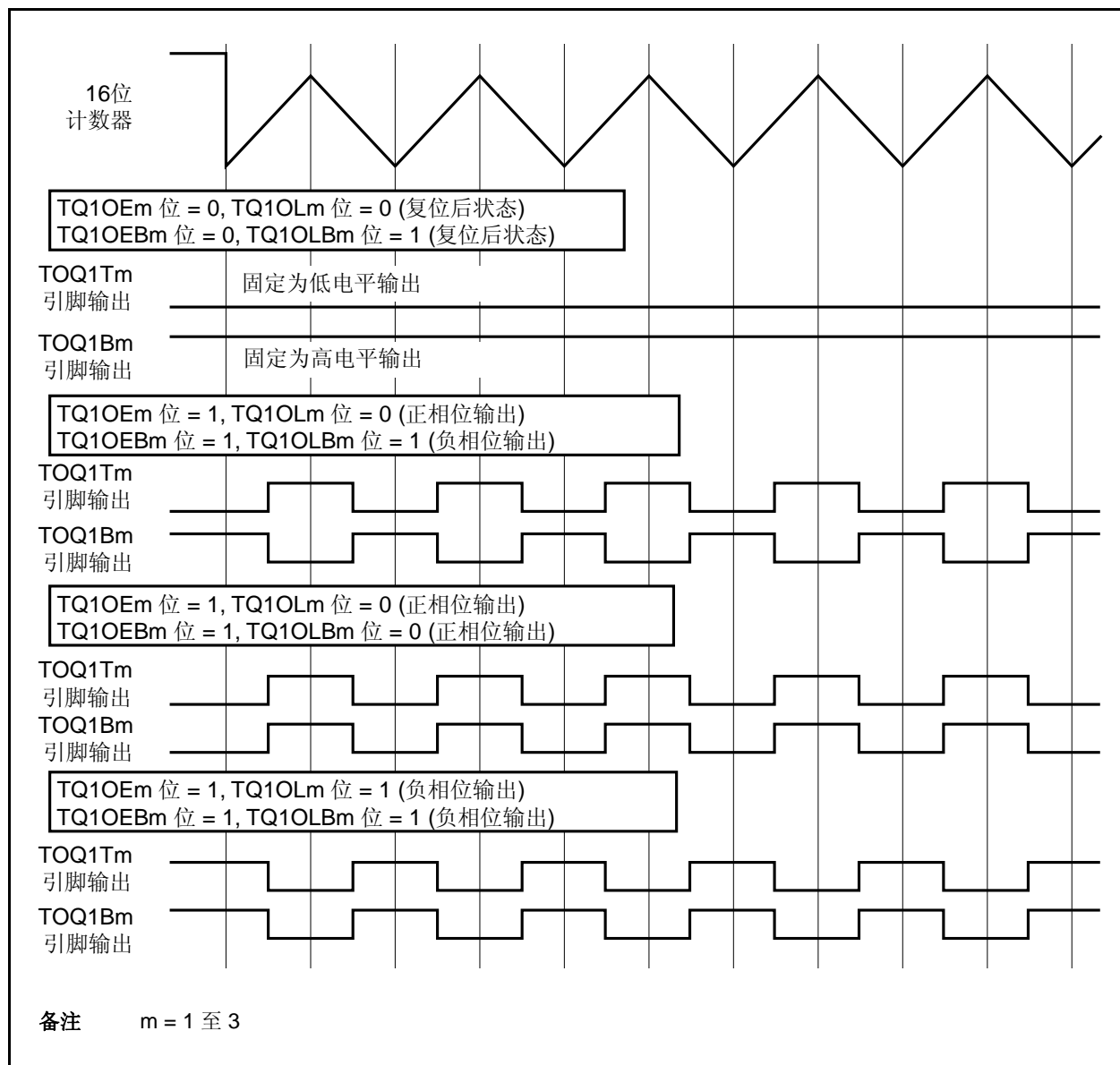


表 9-1. TOQ1Tm 引脚输出

TQ1OLm 位	TQ1OEm 位	TQ1CE 位	TOQ1Tm 引脚输出
0	0	x	低电平输出
	1	0	低电平输出
		1	TOQ1Tm 正相输出
1	0	x	高电平输出
	1	0	高电平输出
		1	TOQ1Tm 负相输出

备注 m = 1 至 3

表 9-2. TOQ1Bm 引脚输出

TQ1OLBm 位	TQ1OEBm 位	TQ1CE 位	TOQ1Bm 引脚输出
0	0	x	低电平输出
	1	0	低电平输出
		1	TOQ1Bm 正相输出
1	0	x	高电平输出
	1	0	高电平输出
		1	TOQ1Bm 负相输出

备注 m = 1 至 3

(6) 高阻输出控制寄存器 00, 01, 10, 11 (HZAyCTLn)

HZAyCTLn 寄存器是用于控制输出缓冲的高阻抗状态的 8 位寄存器。

这些寄存器可以按 8 位或按位读写。然而, HZAyDCF1 位为只读位且不能被写入。

不能进行 16 位访问。

系统复位后, 这些寄存器被置为 00H。

可以通过软件向 HZAyCTLn 寄存器写入相同的数值。

以下显示了检测系数和控制寄存器之间的关系。

用于高阻抗输出的引脚	高阻抗控制系数 (外部引脚)	控制寄存器
输出 TOQH01 至 TOQH03 时	TOQH0OFF	HZA0CTL0
输出 TOP21 时	TOP2OFF	HZA0CTL1
输出 TOQ1T1 至 TOQ1T3 时 输出 TOQ1B1 至 TOQ1B3 时	TOQ1OFF	HZA1CTL0
输出 TOP31 时	TOP3OFF	HZA1CTL1

注意事项 只有在端口引脚功能被设置为上表中所示的功能时才可以进行高阻抗输出。

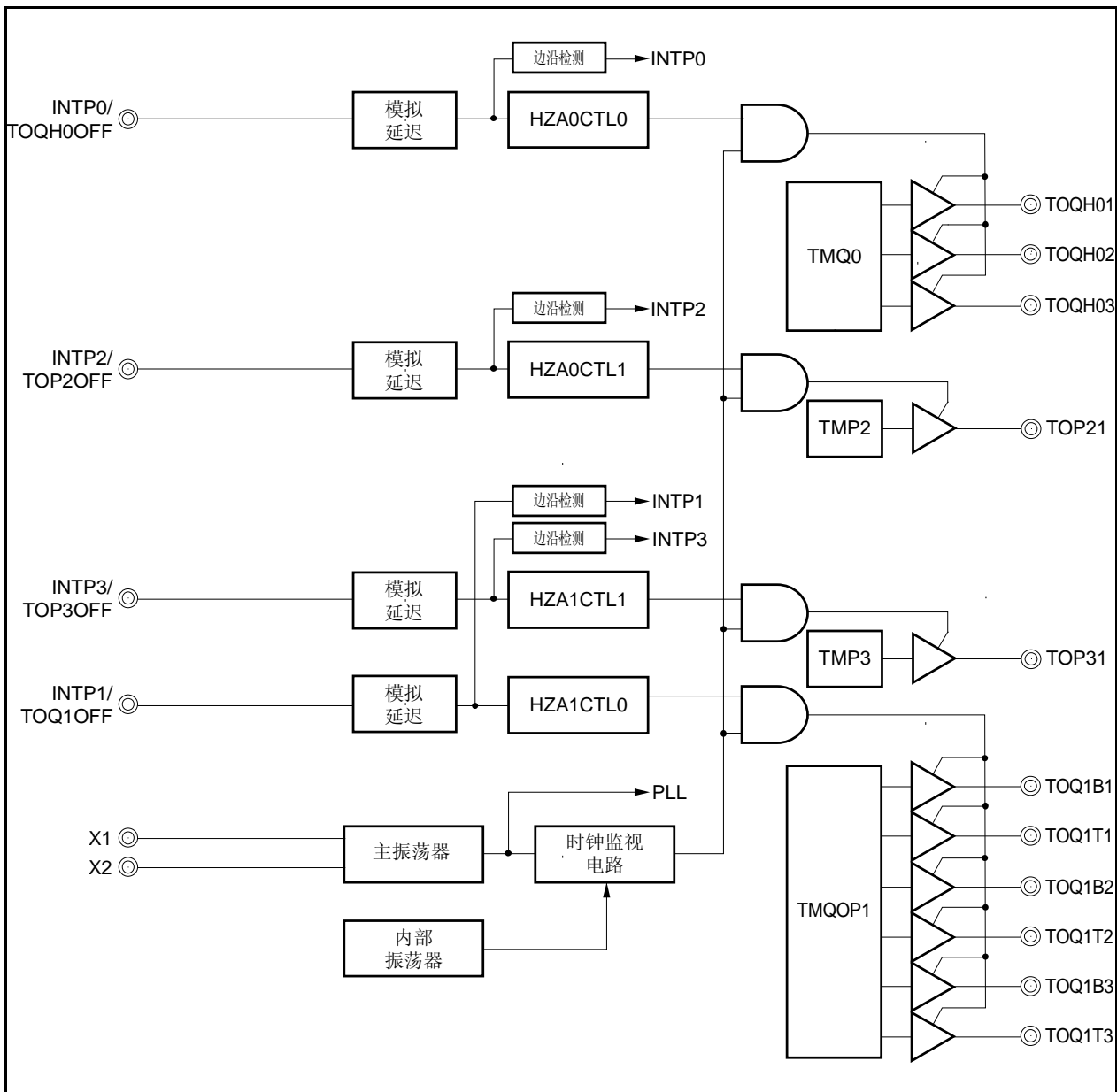
HZAyDCCn	高阻抗输出控制清除位
0	不操作
1	进入高阻抗状态的引脚处于软件允许的输 出且 HZAyDCFn 位清除为 0。

· HZAyDCM 位 = 0时, 无论外部引脚^注的状态如何, 引脚可用作输出功能。
 · HZAyDCM 位 = 1时, 如果一个表示异常的边沿输入到外部引脚^注
 (由HZAyDCNn和HZAyDCPn 位设置), HZAyDCCn 位即使设为1也无效。
 · 读取 HZAyDCCn 位时总为 0。
 · HZAyDCEn 位 = 0时即使HZAyDCCn位设为1也无效。
 · 禁止同时将 HZAyDCTn 和HZAyDCCn 位设为 1。

HZAyDCFn	高阻抗输出状态标志
0	表示允许引脚的输出。 · HZAyDCEn 位 = 0时, 此位清0。 · HZAyDCEn 位 = 1时, 此位清0。
1	表示引脚进入高阻抗状态。 · HZAyDCTn 位 = 1时, 此位设为1。 · 当一个表示异常的边沿输入到外部引脚 ^注 (根据HZAyDCNn 和HZAyDCPn 位设置的检测)时, 此位设为1。

注 HZA0CTL0: TOQH0OFF 引脚, HZA0CTL1: TOP2OFF 引脚
HZA1CTL0: TOQ1OFF 引脚, HZA1CTL1: TOP3OFF 引脚

图 9-4. 高阻抗输出控制器的配置



(a) 设置步骤:

(i) 高阻抗控制操作的设置

- <1> 设置 HZAyDCMn, HZAyDCNn, 以及 HZAyDCP1 位。
- <2> 设置 HZAyDCEn 位为 1 (允许高阻抗控制)。

(ii) 允许高阻抗控制操作后更改设置

- <1> 将 HZAyDCEn 位清零 (用来停止高阻抗控制操作)。
- <2> 更改 HZAyDCMn, HZAyDCNn 以及 HZAyDCP1 位的设置。
- <3> 将 HZAyDCEn 位设为 1 (用来再次允许高阻抗控制操作)。

(iii) 引脚处于高阻抗状态时恢复输出

如果 HZAyDCMn 位为 1, 则在检测到外部引脚[Ⓜ]的有效沿后将 HZAyDCCn 位设为 1 以清除高阻抗状态。然而, 除非在外部引脚[Ⓜ]的输入电平为非激活状态时设置该位, 否则高阻抗状态将不能被清除。

- <1> 将 HZAyDCCn 位设为 1 (命令信号清除高阻抗状态)。
- <2> 读取 HZAyDCFn 位并检查标志状态。
- <3> 如果 HZAyDCFn 位为 1 则返回<1>。且必须检查外部引脚[Ⓜ]的输入电平。
如果 HZAyDCFn 位为 0, 则引脚可以用作输出引脚。

(iv) 通过软件使引脚进入高阻抗状态

外部引脚[Ⓜ]的输入电平为非激活状态时必须通过软件将 HZAyDCTn 位设为 1 以使引脚进入高阻抗状态。示例中步骤如下所示, 其中的设置独立于 HZAyDCMn 位的设置。

- <1> 将 HZAyDCTn 位设为 1 (高阻抗输出命令)。
- <2> 读取 HZAyDCFn 位以检查标志状态。
- <3> 如果 HZAyDCFn 位为 0 则返回<1>。且必须检查外部引脚[Ⓜ]的输入电平。
如果 HZAyDCFn 位为 1, 那么引脚将处于高阻抗状态。

然而, 如果没有使用外部引脚[Ⓜ]且 HZAyDCP1 位和 HZAyDCNn 位的被清为 0, 那么在 HZAyDCTn 位被设为 1 时引脚将进入高阻抗状态。

注 HZA0CTL0: TOQH0OFF 引脚, HZA0CTL1: TOP2OFF 引脚, HZA1CTL0: TOQ1OFF 引脚, HZA1CTL1: TOP3OFF 引脚

9.4 操作

9.4.1 系统概述

(1) 6 相 PWM 输出概述

通过结合使用 TMQ1 和 TMQ1 选项，6 相 PWM 输出模式可以用于生成一个 6 相 PWM 输出波形。

通过将 TMQ1 的 TQ1CTL1.TQ1MD2 至 TQ1CTL1.TQ1MD0 位设为“111”可以使能 6 相 PWM 输出模式。

TMQ1 中的一个 16 位计数器和四个 16 位比较寄存器用于生成一个基本的 3 相波形。

比较寄存器的功能如下：

TMP1 可以通过 TMQ1 来未完成调制操作，用来为 A/D 转换器 0 和 1 启动转换触发源。

比较寄存器	功能	设置范围
TQ1CCR0 寄存器	设置周期	$0002H \leq m \leq FFEH$
TQ1CCR1 寄存器	指定相 U 的输出宽度	$0000H \leq i \leq m + 1$
TQ1CCR2 寄存器	指定相 V 的输出宽度	$0000H \leq j \leq m + 1$
TQ1CCR3 寄存器	指定相 W 的输出宽度	$0000H \leq k \leq m + 1$

备注

- m = TQ1CCR0 寄存器的设定值
- i = TQ1CCR1 寄存器的设定值
- j = TQ1CCR2 寄存器的设定值
- k = TQ1CCR3 寄存器的设定值

死区时间间隔从基本 3 相波形中生成，该波形通过使用三个 10 位死区时间计数器和一个用于创建与基本 3 相波形相位相反的波形的比较寄存器来生成。这样就会生成一个 6 相 PWM 输出波形 (U, \bar{U} , V, \bar{V} , W 以及 \bar{W})。

用于生成基本 3 相波形的 16 位计数器向上计数或向下计数。操作开始后，该寄存器将开始向上计数。当它的计数值与 TQ1CCR0 寄存器设置的周期匹配时，计数器将开始向下计数。当计数值与 0001H 匹配时，计数器将再次开始向上计数。这意味着载波周期等于两倍于“TQ1CCR0 寄存器中设定值+ 1”的值。

生成死区时间间隔的 10 位死区时间计数器 1 至 3 向上计数。因此，TMQ1 死区时间比较寄存器 (TQ1DTC) 中的设定值被用作死区时间值。由于使用了三个计数器，因此死区时间可以在相 U, V, W 中独立地生成。然而，由于只有一个指定死区时间值 (TQ1DTC) 的寄存器，因此在三种相中使用同一个死区时间值。

<R>

图 9-5. 6 相 PWM 输出模式的概要

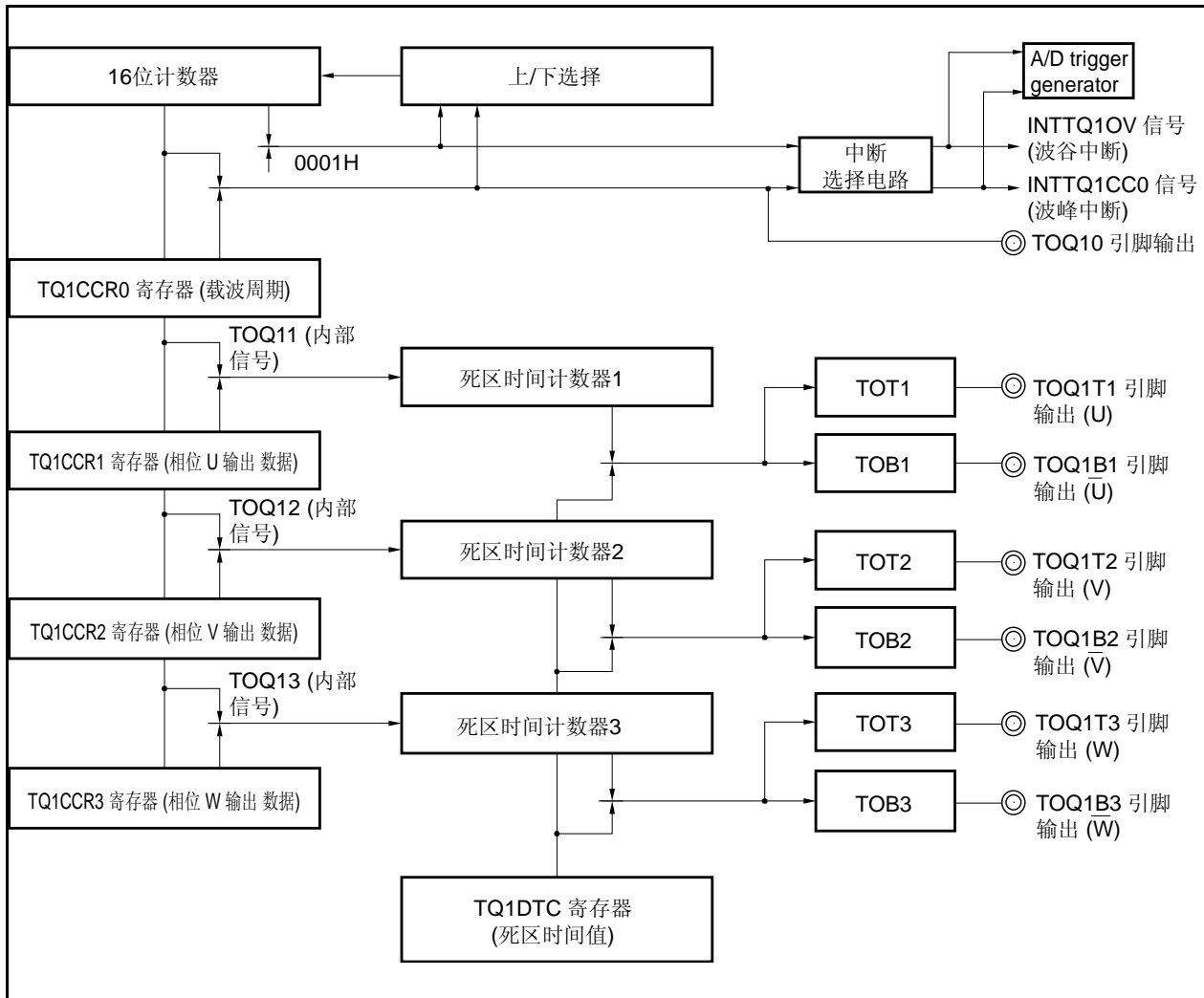
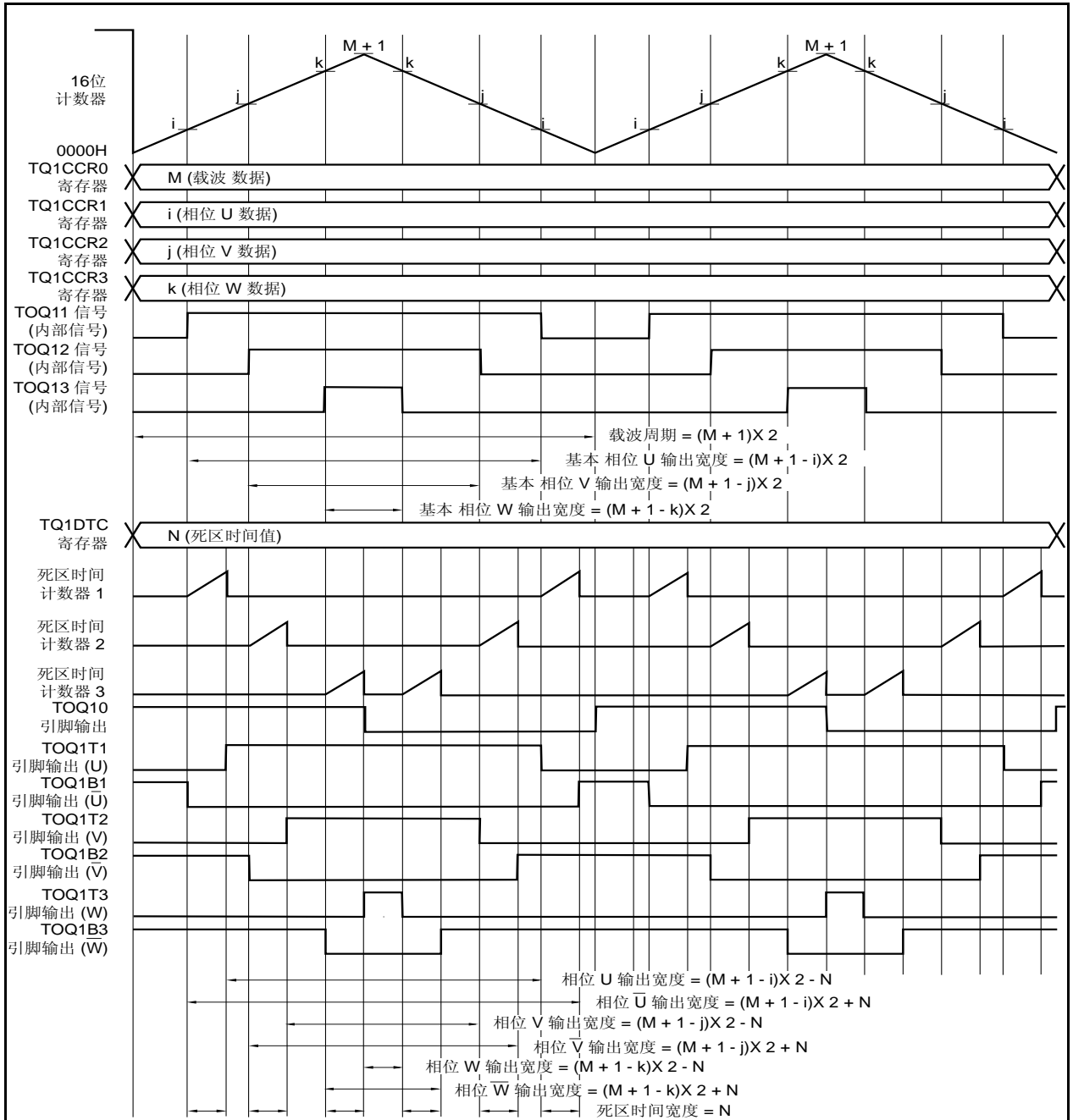


图 9-6. 6 相 PWM 输出模式的时序图



- 注意事项**
- 在 6 相 PWM 输出模式中，应将 TQ1CCR0 寄存器的值“M”设定在 $0002H \leq M \leq FFEH$ 的范围内。
 - 只有最大到“M + 1”值可以被设置到 TQ1CCR1，TQ1CCR2 以及 TQ1CCR3 寄存器中。
 - 如果“0000H”被设置到 TQ1CCR1，TQ1CCR2 以及 TQ1CCR3 寄存器中，将会输出 100%。如果“M + 1”被设置到 TQ1CCR1，TQ1CCR2 以及 TQ1CCR3 寄存器中，将会输出 0%。如果“M + 2”或更高的值被设置到 TQ1CCR1，TQ1CCR2 以及 TQ1CCR3 寄存器中，输出（50% 占空比）会在 16 位计数器的峰值（M + 1）处上升，在谷值（0000H）处下降。
 - 如果相 U，V 和 W 的输出宽度的等式（例如 $(M + 1 - i) \times 2 - N$ ）的操作值为 0 或低于 0，它将会收敛为 0（100% 输出）。如果操作值高于“ $(M + 1) \times 2$ ”，它将会收敛为 $(M + 1) \times 2$ （0% 输出）。

(2) 中断请求

有两种类型的中断请求可用：INTTQ1CC0（波峰中断）信号和INTTQ1OV（波谷中断）信号。

INTTQ1CC0和INTTQ1OV信号可以使用TQ1OPT1寄存器来进行选择。

关于选择中断的详细信息，请参见**9.4.3 中断选择功能**。

- INTTQ1CC0（波峰中断）信号： 显示用于向上计数的16位计数器的值与TQ1CCR0寄存器的值间匹配的中断信号
- INTTQ1OV（波谷中断）信号： 显示用于向下计数的16位计数器的值与0001H值间匹配的中断信号

(3) 定时器操作过程中重写寄存器

以下寄存器具有一个缓冲寄存器并可以在随时重写模式，批量重写模式或间隙批量重写模式中进行重写。

相关单元	寄存器
定时器 P1	TMP1 捕捉 / 比较寄存器 0 (TP1CCR0) TMP1 捕捉 / 比较寄存器 1 (TP1CCR1)
定时器 Q1	TMQ1 捕捉 / 比较寄存器 0 (TQ1CCR0) TMQ1 捕捉 / 比较寄存器 1 (TQ1CCR1) TMQ1 捕捉 / 比较寄存器 2 (TQ1CCR2) TMQ1 捕捉 / 比较寄存器 3 (TQ1CCR3)
定时器 Q1 选项	TMQ1 选项寄存器 1 (TQ1OPT1)

关于比较寄存器传送功能的详细信息，请参见**9.4.4 使用传送功能重写寄存器的操作**。

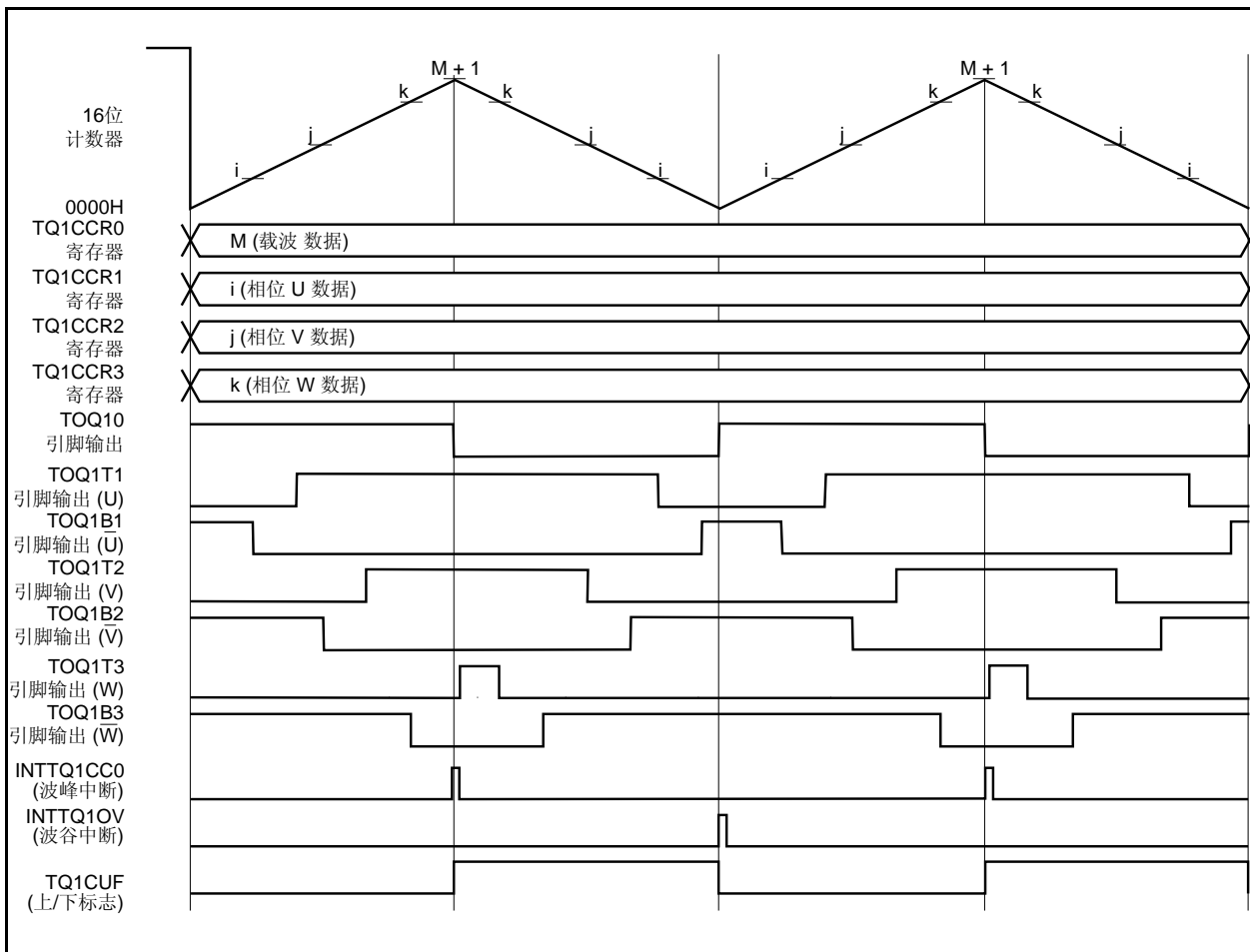
(4) 16 位计数器的向上 / 向下计数操作

16 位计数器的操作状态可以通过使用 TMQ1 选项寄存器 0 (TQ1OPT0) 的 TQ1CUF 位来检查。

TQ1CUF 位的状态	16 位计数器的状态	16 位计数器值的范围
TQ1CUF 位 = 0	向上计数	0000H – m
TQ1CUF 位 = 1	向下计数	(m + 1) – 0001H

备注 m = TQ1CCR0 寄存器的设定值

图 9-7. 中断及向上 / 向下标志



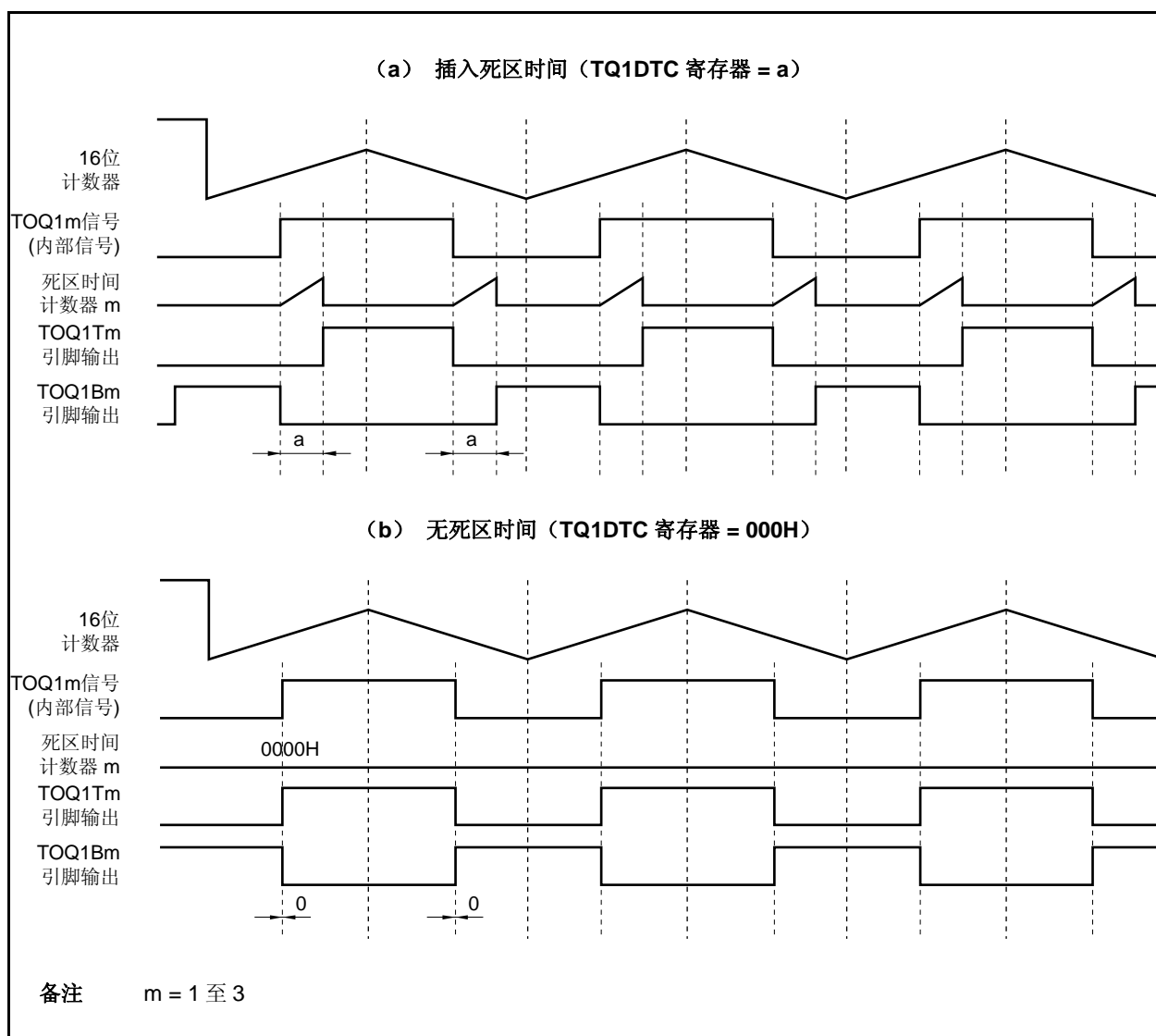
9.4.2 死区时间控制（负相波形信号的产生）

(1) 死区时间控制原理

在 6 相 PWM 输出模式中，比较寄存器 1 至 3 (TQ1CCR1, TQ1CCR2 以及 TQ1CCR3) 用于设置占空比系数，而比较寄存器 0 (TQ1CCR0) 则用于设置周期。通过设置这四种寄存器并启动 TMQ 操作，可以生成三种带有可变占空比的 PWM 输出波形（基本 3 相波形）。这三种 PWM 输出波形被输入到定时器 Q 选项单元 (TMQOP1) 中，并创建带有死区时间的反转信号以生成三组（六个）PWM 波形。

TMQOP1 单元由三个 10 位计数器（死区时间计数器 1 至 3）组成，这三个计数器与 TMQ1 的计数时钟以及指定死区时间的 TMQ1 死区时间比较寄存器 (TQ1DTC) 同步操作。如果 TQ1DTC 寄存器的值被设置为“a”，那么死区时间的值为“a”且间隔“a”将会创建在正相波形和负相波形之间。

图 9-8. 带有死区时间的 PWM 输出波形 (1)



(2) 0%/100%的 PWM 输出

V850ES/IE2 可以实现 PWM 输出的 0%波形输出和 100%波形输出。

低电平会作为 0%波形输出从 TOQ1Tm 引脚中连续输出。高电平会作为 100%波形输出从 TOQ1Tm 引脚中连续输出。

TQ1CCR0 寄存器 = M 时，通过将 TQ1CCRm 寄存器设为“M + 1”来输出 0%波形。

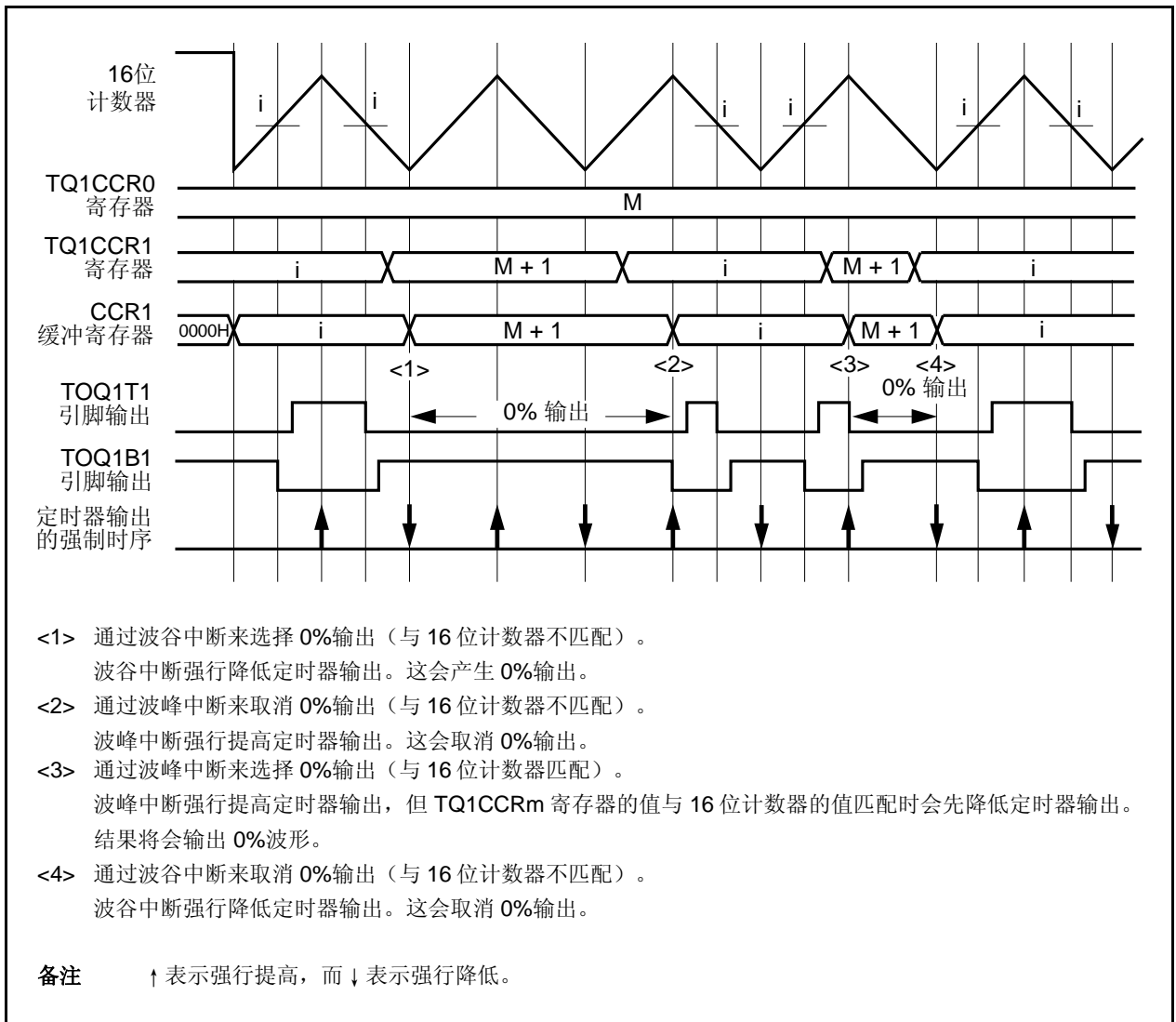
通过将 TQ1CCRm 寄存器设为“0000H”来输出 100%波形。

定时器进行操作时允许对 TQ1CCRm 寄存器进行重写，0%波形输出或 100%波形输出可以在发生波峰中断 (INTTQ1CC0) 和波谷中断 (INTTQ1OV) 时进行选择。

备注 m = 1 至 3

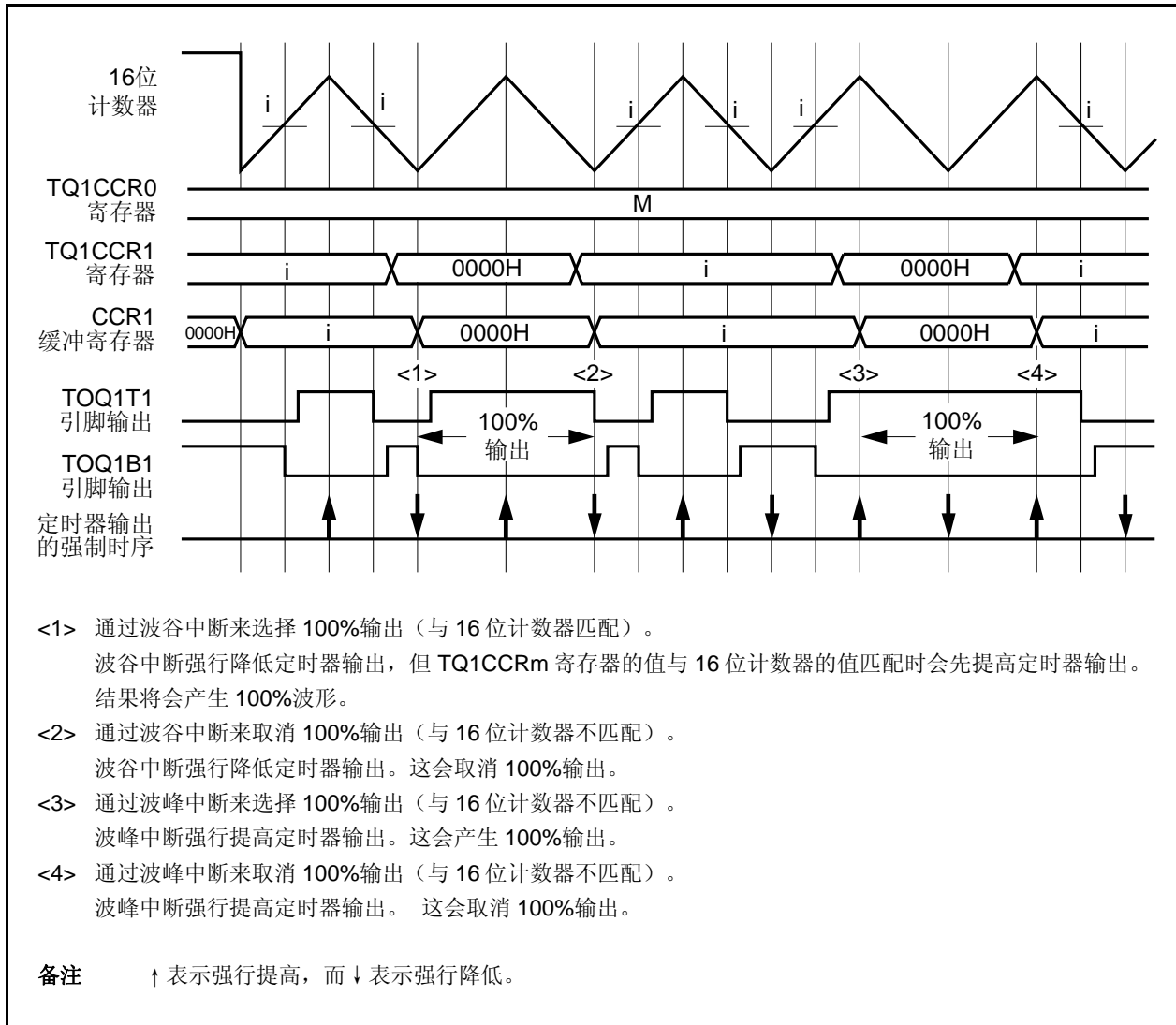
<R>

图 9-9. 0%PWM 输出波形 (带有死区时间)



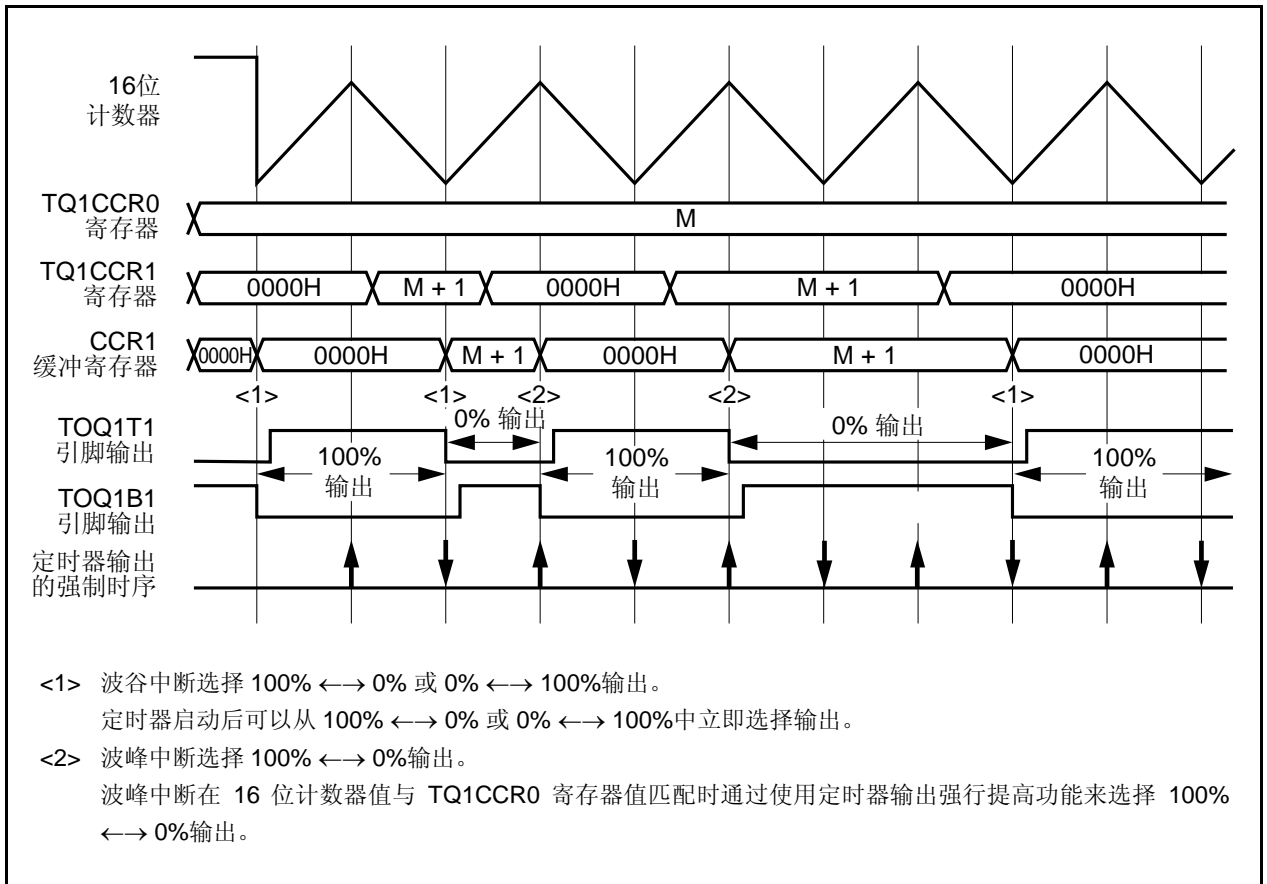
<R>

图 9-10. 100%PWM 输出波形（带有死区时间）



<R>

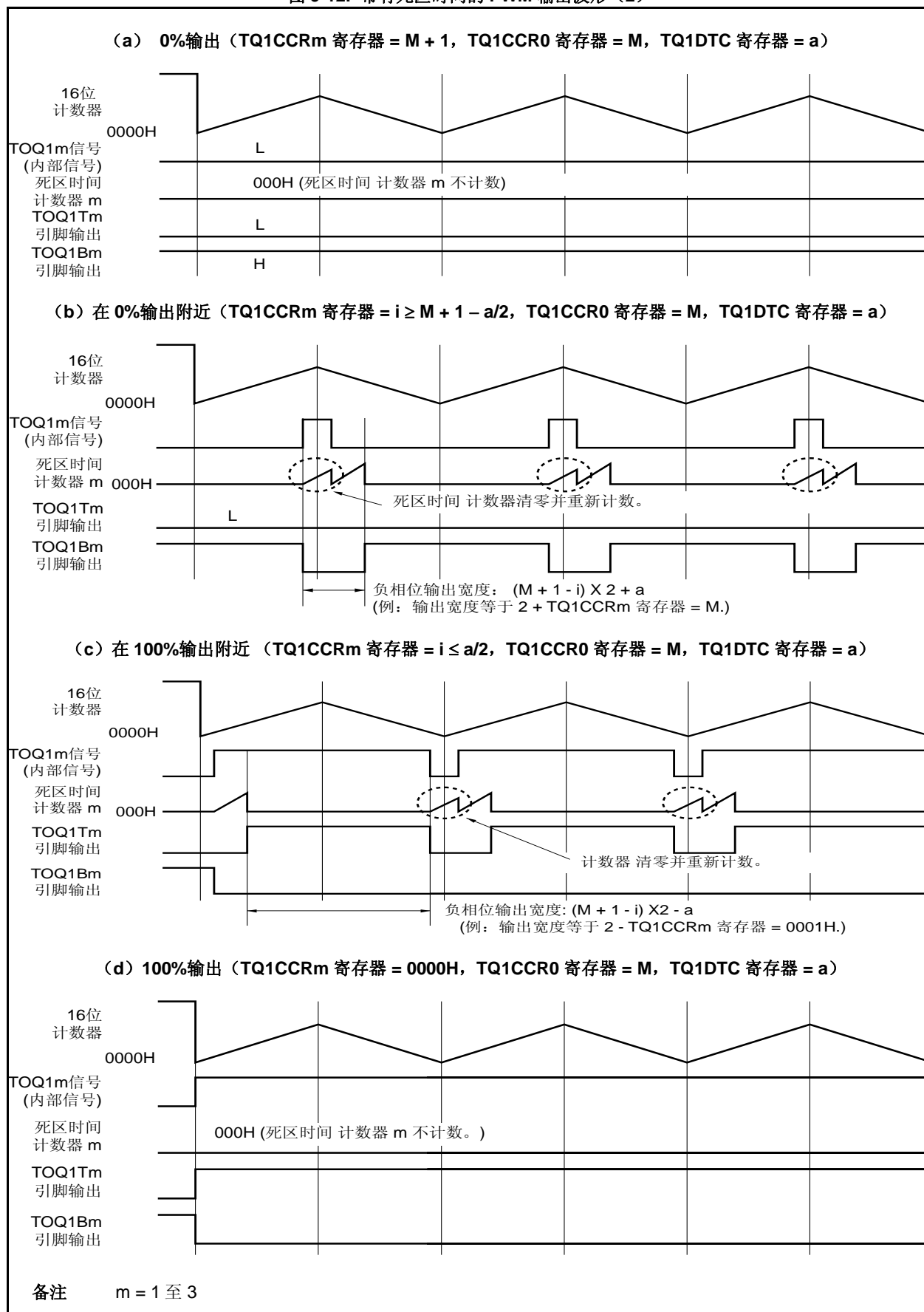
图 9-11. 从 0% 到 100% 以及从 100% 到 0% 的 PWM 输出波形（带有死区时间）

**(3) 在 0% 和 100% 输出附近的输出波形**

如果由于 16 位计数器的值与比较寄存器的值在死区时间计数时相匹配而生成了中断，那么死区时间计数器将会被清除并再次开始计数操作。

以下显示了在 0% 和 100% 输出附近的死区时间控制的输出波形。

图 9-12. 带有死区时间的 PWM 输出波形 (2)



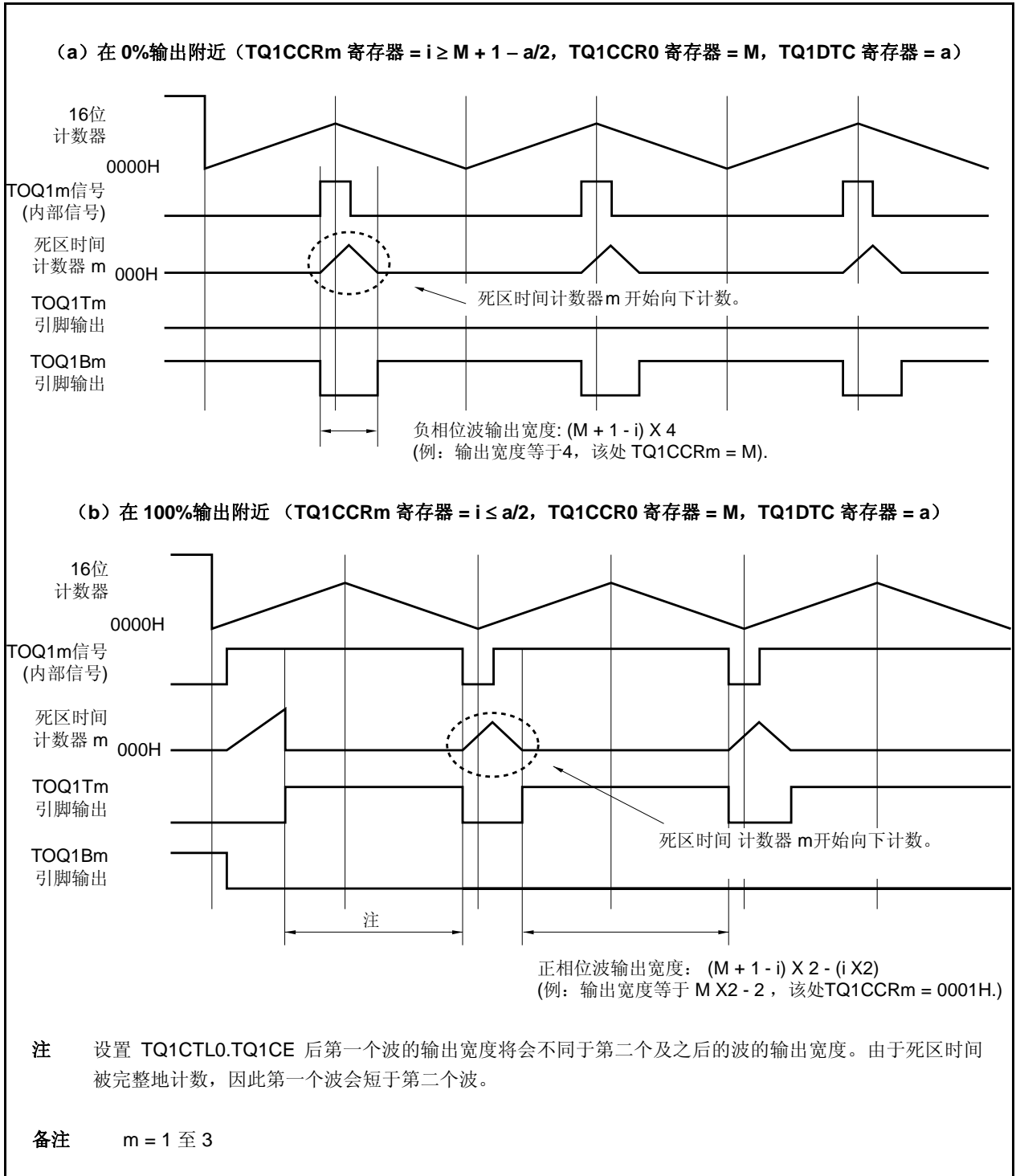
(4) 自动缩小死区时间宽度功能 (TQ1OPT2.TQ1DTM 位 = 1)

通过将 TQ1OPT2.TQ1DTM 位设为 1 可以在 0%输出或 100%输出的附近自动缩小死区时间宽度。

通过将 TQ1DTM 位设为 1 不能清除死区时间计数器，但如果在死区时间计数过程中改变定时器 Q 的 TOQ1m (内部信号) 输出，则将会开始向下计数。

以下时序图说明了 TQ1DTM 位设为 1 时死区计数器的操作。

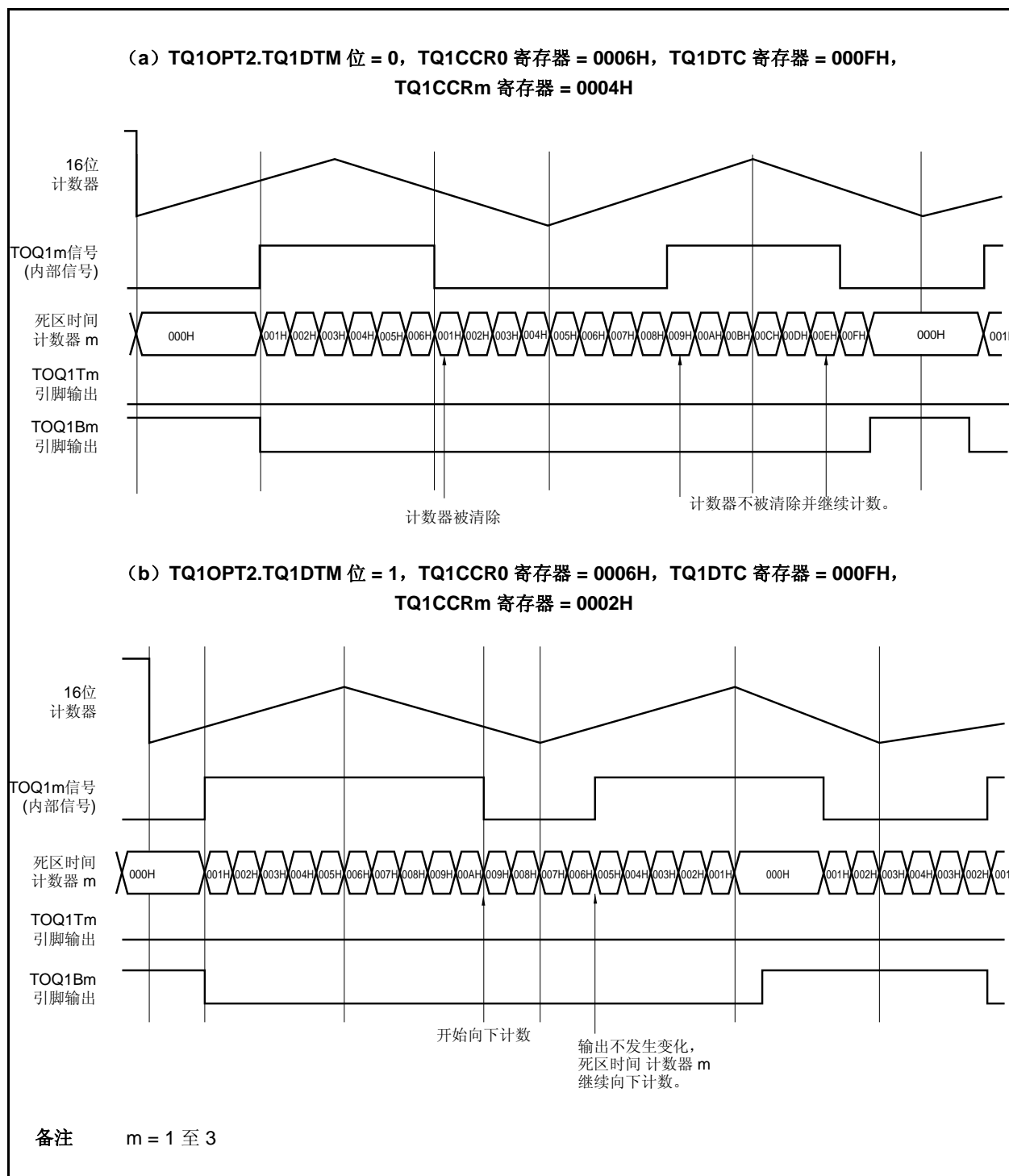
图 9-13. 死区时间计数器 m 的操作 (1)



(5) 设置不正确时死区时间的控制

通常情况下，在死区时间计数过程中，TMQ1 的 TOQ1m（内部信号）输出只会在 0% 和 100% 输出附近发生变化。本节介绍 TQ1CCR0 寄存器（载波周期）和 TQ1DTC 寄存器（死区时间值）设置错误时的例子。如果这些寄存器没有被正确设置，那么在死区时间计数过程中，TMQ1 的 TOQ1m（内部信号）输出将不只变化一次。以下的流程图显示了在这种情况下的 6 相 PWM 输出波形。

图 9-14. 死区时间计数器 m 的操作 (2)



9.4.3 中断选择功能

<R>

- 可以选择的中断为 INTTQ1CC0（波峰中断）以及 INTTQ1OV（波谷中断）。
- TQ1OPT1.TQ1ICE 位用于允许 INTTQ1CC0 中断的输出以及中断要进行选择的次数。
- TQ1OPT1.TQ1IOE 位用于允许 INTTQ1OV 中断的输出以及中断要进行选择的次数。
- TQ1OPT1.TQ1ID4 至 TQ1OPT1.TQ1ID0 位用于指定进行选择计数的中断的次数。
指定选择计数的中断被屏蔽且屏蔽的中断在下一个时序发生。
- TQ1OPT2 的 TQ1RDE 位用于指定是否要对传送进行选择。
如果指定要对传送进行选择，那么选择后传送将会与中断输出同时序进行。如果指定不对传送进行选择，那么在 TQ1CCR1 寄存器被写入后将会在传送时序进行传送。
- TQ1OPT0.TQ1CMS 位用于指定带有传送功能的寄存器是执行批量重写还是随时重写。
TQ1CMS 位为 0 时，寄存器的值会在传送的同时进行更新。当 TQ1CMS 位为 1 时，寄存器的值则会在寄存器写入一个新的值后立即更新。
从 TQ1CCRm 寄存器至 CCRm 缓冲寄存器的传送操作与中断选择时序同步进行。

- 注意事项**
1. 在批量重写模式（传送模式）中使用中断选择功能时，将会按间隙批量重写模式（传送选择模式）来执行操作。
 2. 完成选择后将会生成中断。

(1) 中断选择操作

图 9-15. TQ1OPT1.TQ1ICE 位 = 1, TQ1OPT1.TQ1IOE 位 = 1, TQ1OPT2.TQ1RDE 位 = 1 时的中断选择操作 (波峰 / 波谷中断输出)

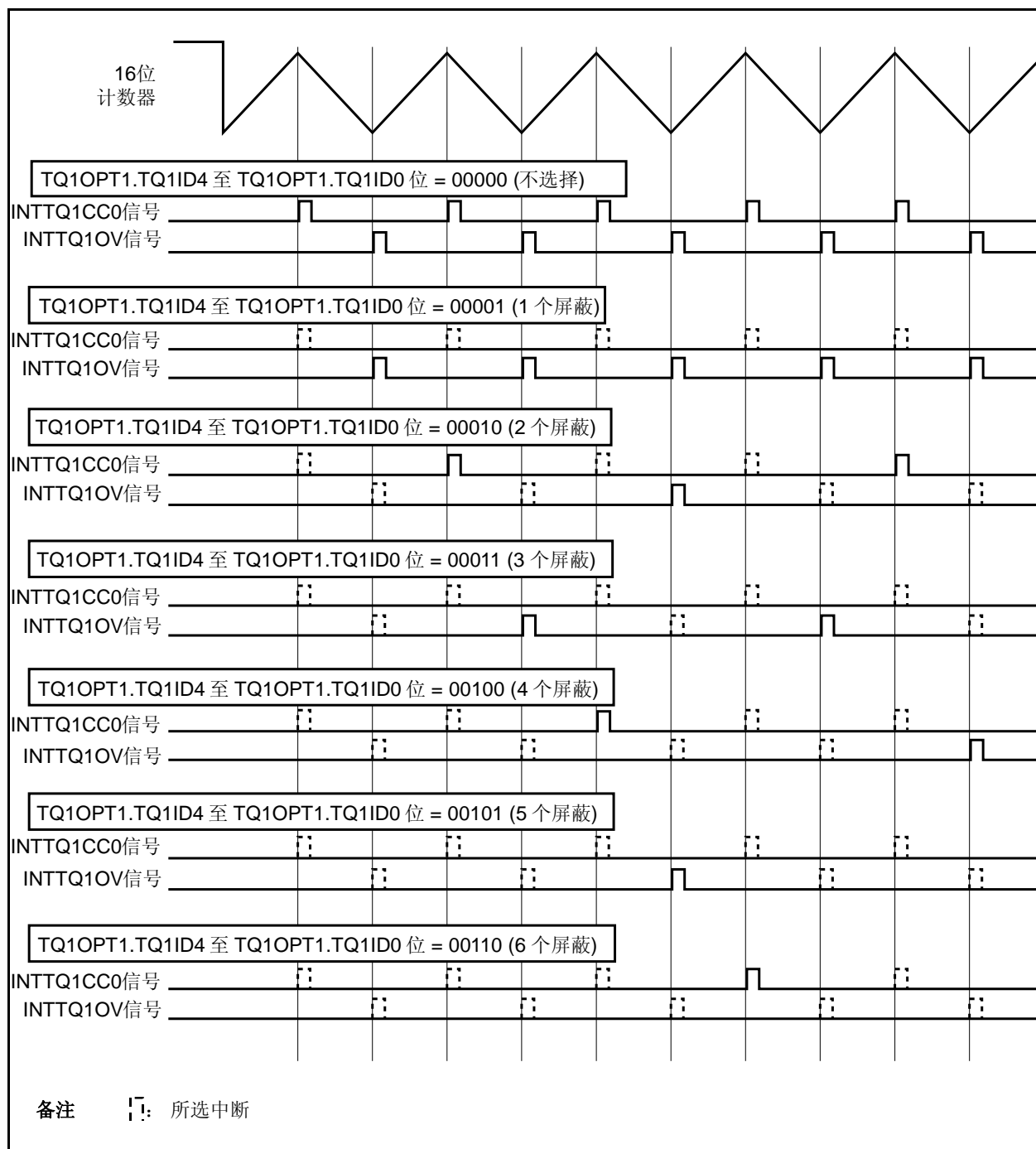


图 9-16. TQ1OPT1.TQ1ICE 位 = 1, TQ1OPT1.TQ1IOE 位 = 0, TQ1OPT2.TQ1RDE 位 = 1 时的中断选择操作 (波峰中断输出)

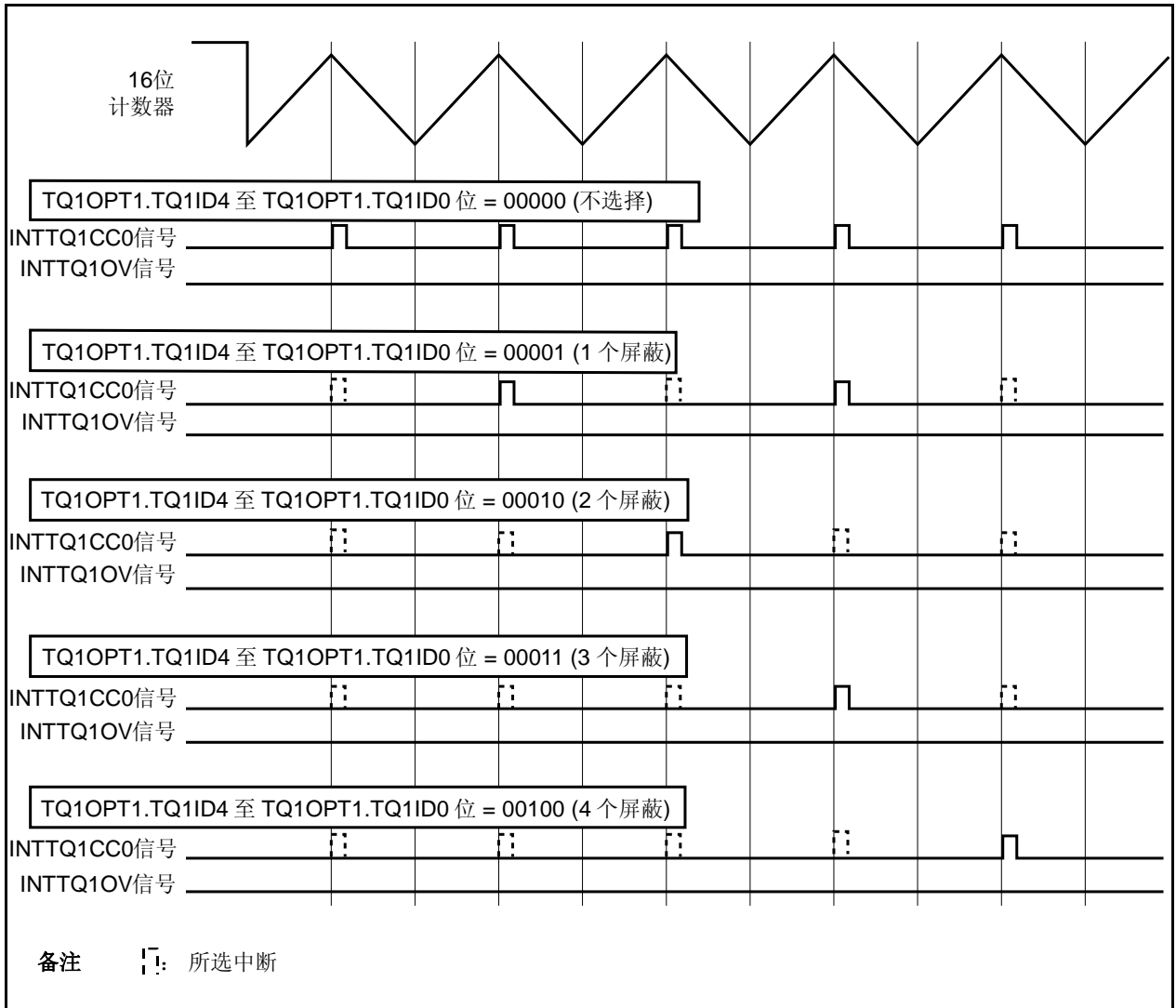
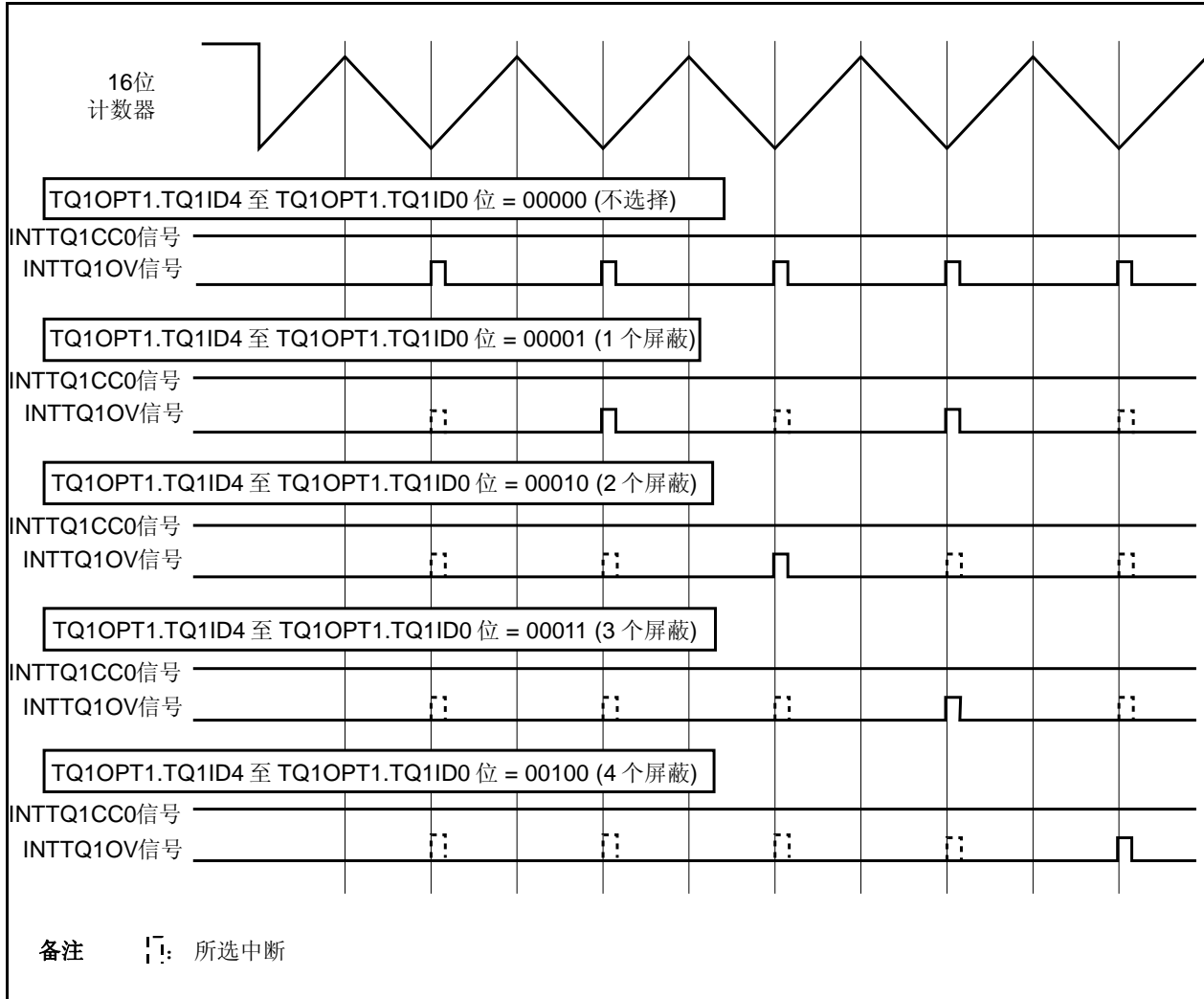


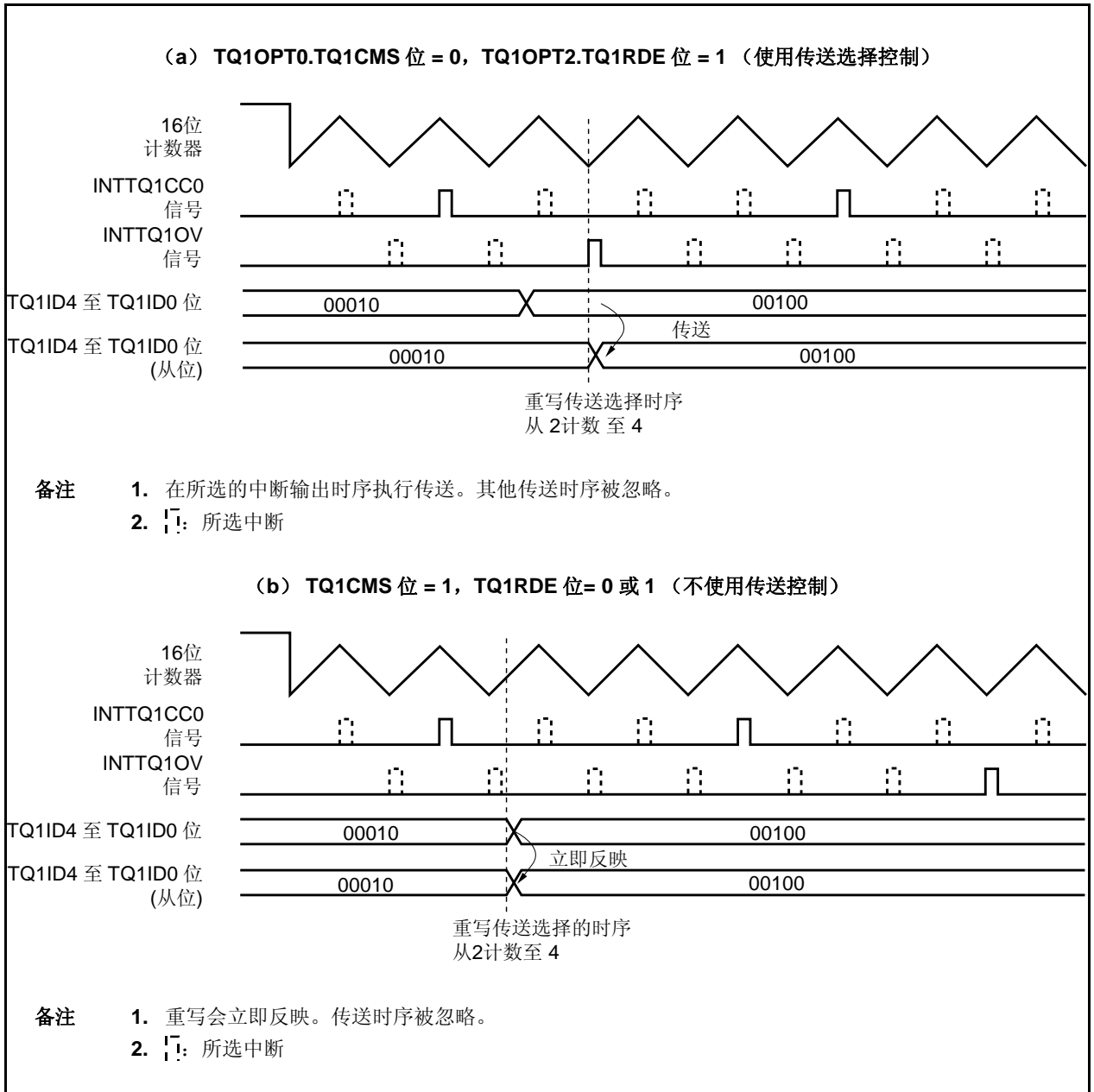
图 9-17. TQ1OPT1.TQ1ICE 位 = 0, TQ1OPT1.TQ1IOE 位 = 1, TQ1OPT2.TQ1RDE 位 = 1 时的中断选择操作 (波谷中断输出)



(2) 交替输出波峰中断 (INTTQ1CC0) 和波谷中断 (INTTQ1OV)

若要交替输出波峰及波谷中断, 则应将 TQ1OPT1.TQ1ICE 和 TQ1OPT1.TQ1IOE 位都设为 1。

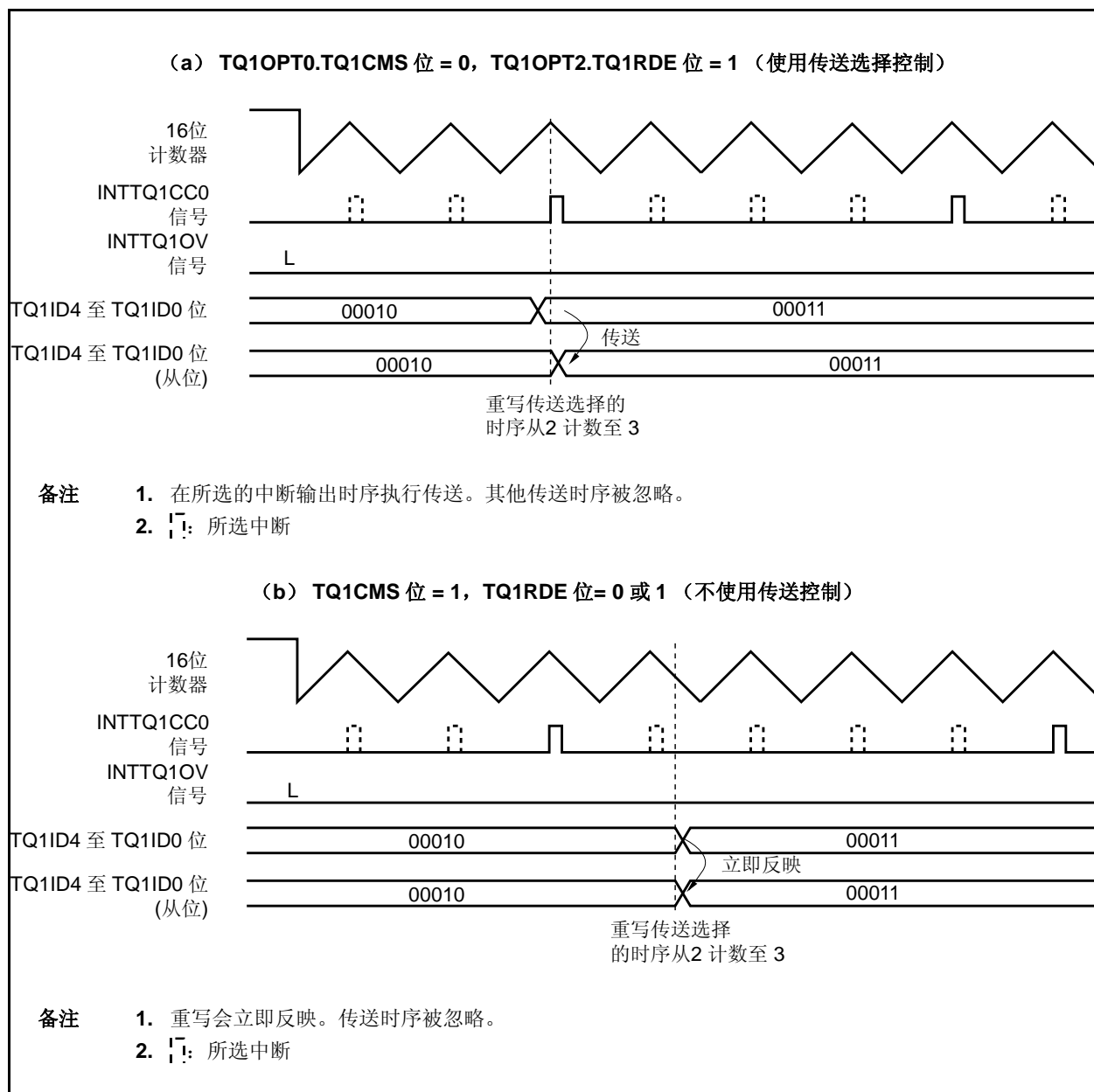
图 9-18. 波峰 / 波谷中断输出



(3) 只输出波峰中断 (INTTQ1CC0)

将 TQ1OPT1.TQ1ICE 位设为 1 并将 TQ1OPT1.TQ1IOE 位清零。

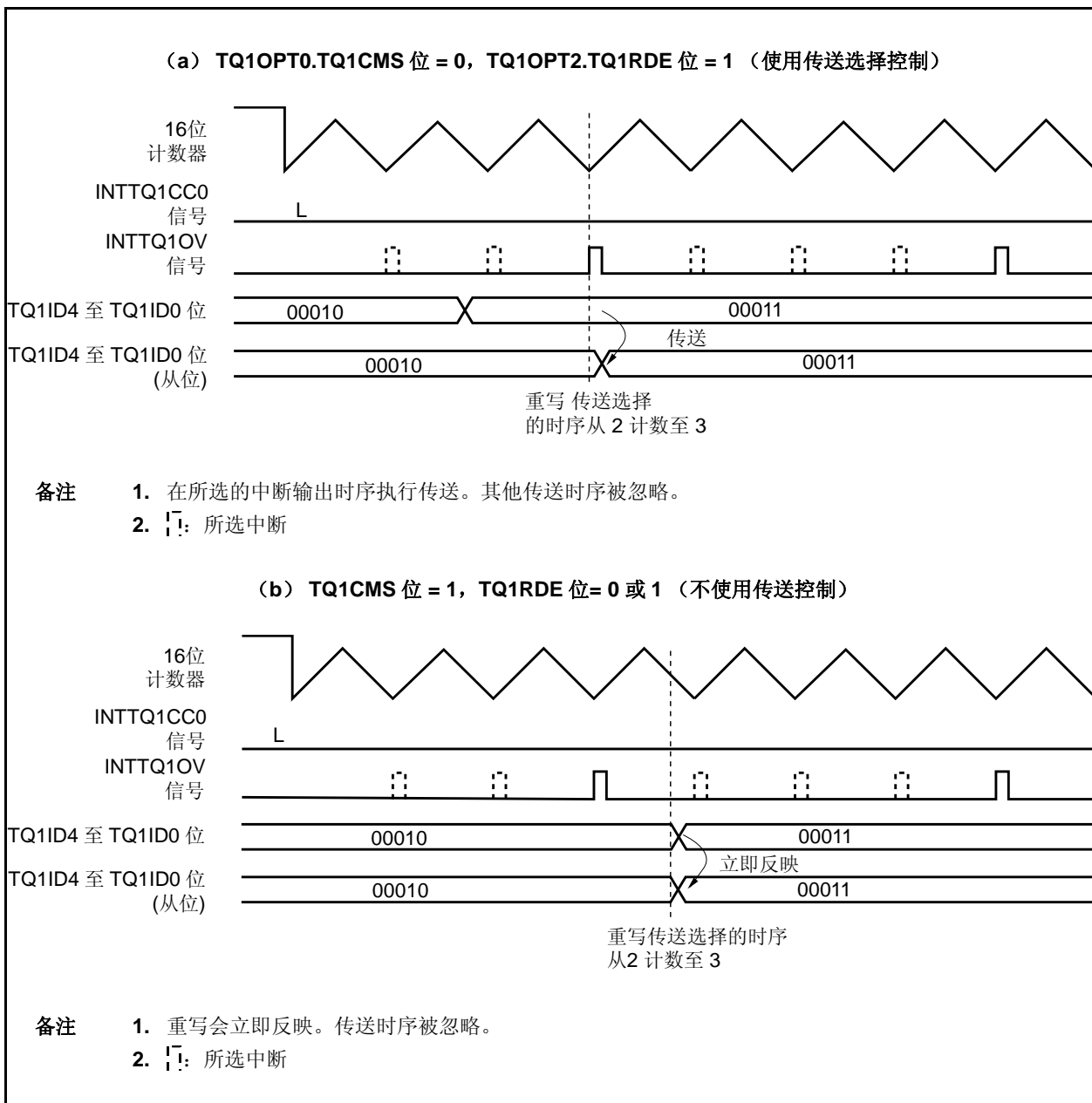
图 9-19. 波峰中断输出



(4) 只输出波谷中断 (INTTQ1OV)

将 TQ1OPT1.TQ1ICE 位清除为 0 并将 TQ1OPT1.TQ1IOE 位设置为 1。

图 9-20. 波谷中断输出



9.4.4 重写带有传送功能的寄存器的操作

以下七种寄存器提供了传送功能并用于控制马达。每个寄存器都有一个缓冲寄存器。

- TQ1CCR0: 指定 16 位计数器 (TMQ) 周期的寄存器。
- TQ1CCR1: 指定 TOQ1T1 (U) 及 TOQ1B1 (\bar{U}) 的占空比的寄存器。
- TQ1CCR2: 指定 TOQ1T2 (V) 及 TOQ1B2 (\bar{V}) 的占空比的寄存器。
- TQ1CCR3: 指定 TOQ1T3 (W) 及 TOQ1B3 (\bar{W}) 的占空比的寄存器。
- TQ1OPT1: 指定中断选择的寄存器。
- TP1CCR0: 指定 A/D 转换开始触发生成时序 (调制操作中的 TMP1) 的寄存器。
- TP1CCR1: 指定 A/D 转换开始触发生成时序 (调制操作中的 TMP1) 的寄存器。

带有传送功能的寄存器中包含以下三种重写模式:

- 随时重写模式

该模式通过将 TQ1OPT0.TQ1CMS 位置为 1 来设置。TQ1OPT2.TQ1RDE 位的设置被忽略。

在该模式中, 每个比较寄存器都会独立地更新, 而比较寄存器的值则会在写入新的值后立即更新。

- 批量重写模式 (传送模式)

该模式通过将 TQ1OPT0 的 TQ1CMS 位清除为 0, TQ1OPT1.TQ1ID4 至 TQ1OPT1.TQ1ID0 清除为 00000, 以及 TQ1OPT2.TQ1RDE 位清除为 0 来进行设置。当数据被写入 TQ1CCR1 寄存器时, 这七种寄存器会在下一个传送时序立即传送到缓冲寄存器中。除非 TQ1CCR1 寄存器被重写, 否则即使其他六个寄存器被重写传送操作也不会被执行。

不考虑中断, 传送时序是每个波峰 (16 位计数器值与 TQ1CCR0 寄存器值匹配) 与波谷 (16 位计数器值与 0001H 匹配) 的时序。

- 间隙批量重写模式 (传送选择模式)

该模式可以通过将 TQ1OPT0.TQ1CMS 位清零并将 TQ1OPT2.TQ1RDE 位设为 1 来进行设置。

当数据被写入 TQ1CCR1 寄存器时, 这七种寄存器会在下一个传送时序时立即传送到缓冲寄存器中。除非 TQ1CCR1 寄存器被重写, 否则即使其他六个寄存器被重写传送操作也不会被执行。

如果通过 TQ1OPT1 寄存器来指定中断选择, 那么当中断被选择时传送时序也将会被选择, 而这七种寄存器也会在波峰中断 (16 位计数器值与 TQ1CCR0 寄存器值匹配) 或波谷中断 (16 位计数器值与 0001H 匹配) 选中时序处立即被传送。

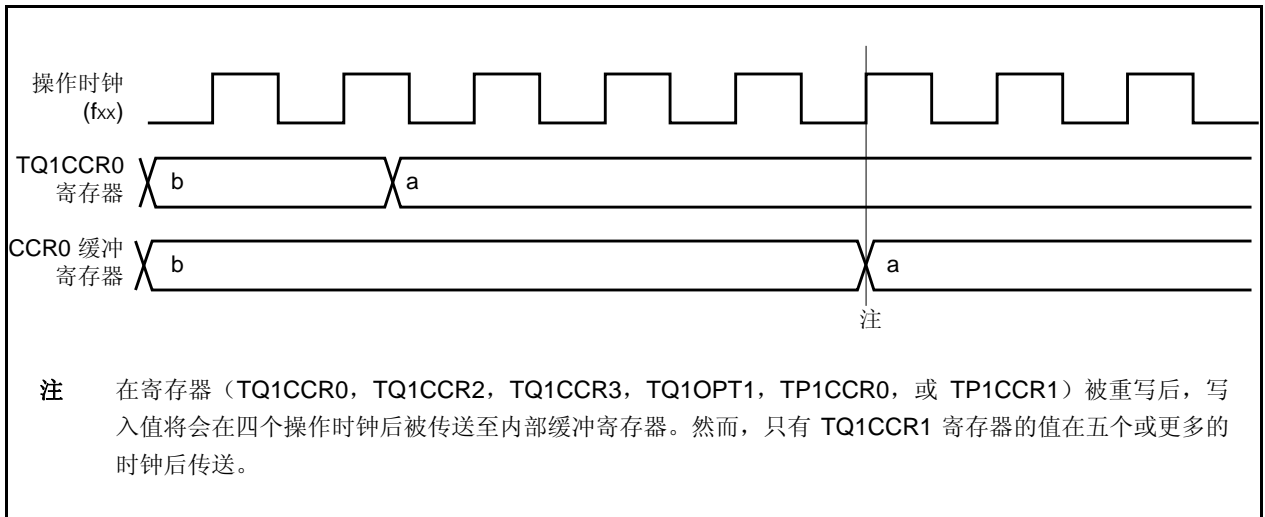
关于中断选择功能的详细信息, 请参见 9.4.3 中断选择功能。

(1) 随时重写模式

该模式通过将 TQ1OPT0.TQ1CMS 位置为 1 来设置。TQ1OPT2.TQ1RDE 位的设置被忽略。

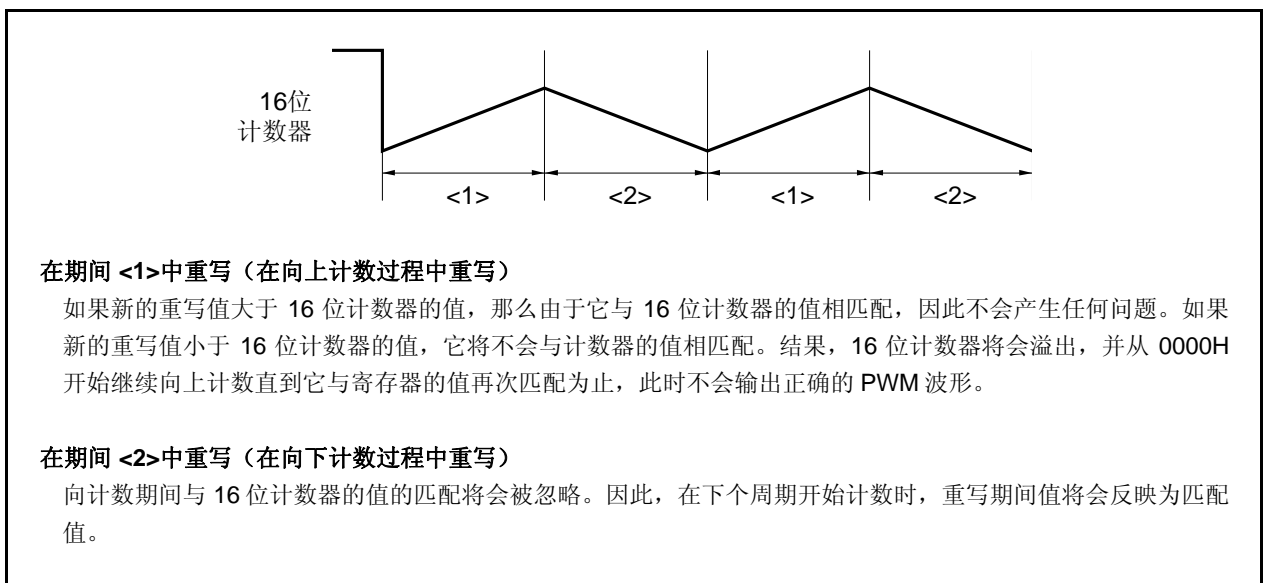
在该模式中，写入带有传送功能的各个寄存器中的值被立即传送到内部缓冲寄存器中并会与计数器的值进行比较。在 16 位计数器的计数值与 TQ1CCRm 寄存器的值匹配后，若带有传送功能的寄存器在该模式中被重写，由于第一次匹配发生后忽略下一次匹配，所以重写值将不会被反映出来。如果寄存器在向上计数过程中被重写，那么在计数器开始向下计数后新的寄存器值将会变为有效的。

图 9-21. 反映重写值的时序

**(a) 重写 TQ1CCR0 寄存器**

即使在随时重写模式中对 TQ1CCR0 寄存器进行重写，在某些情况下，新的值也不能反映出来。

图 9-22. 重写 TQ1CCR0 寄存器的示例



(b) 重写 TQ1CCRm 寄存器

图 9-24 显示了 16 位计数器的值与 TQ1CCRm 寄存器的值匹配前的重写时序（图 9-23 中的<1>），图 9-25 显示了 16 位计数器的值与 TQ1CCRm 寄存器的值匹配后的重写时序（图 9-23 中的<2>）。

图 9-23. 16 位计数器与 TQ1CCRm 寄存器的基本操作

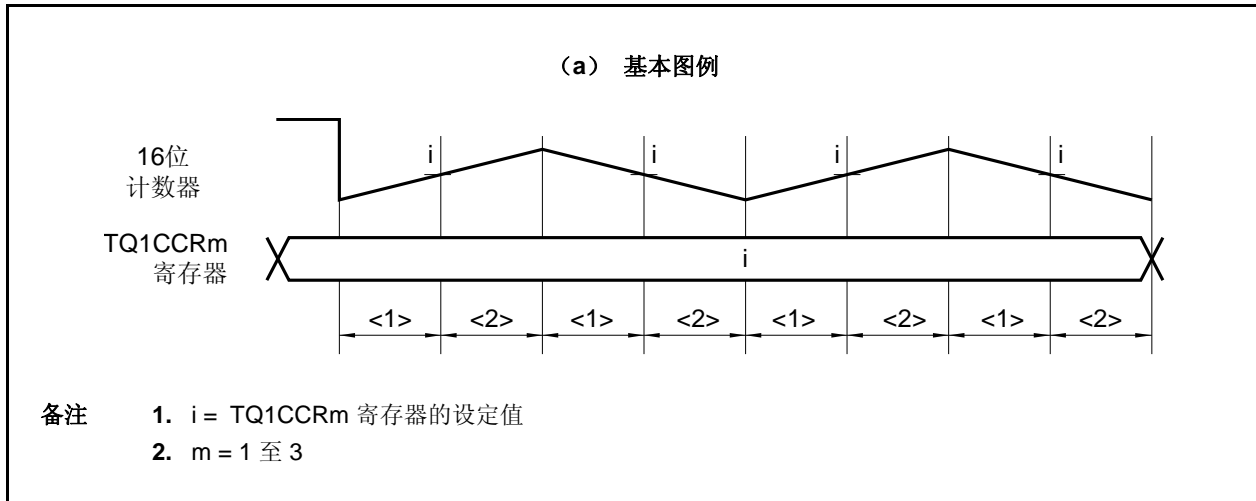


图 9-24. 重写 TQ1CCR1 至 TQ1CCR3 寄存器的示例（匹配发生前重写）

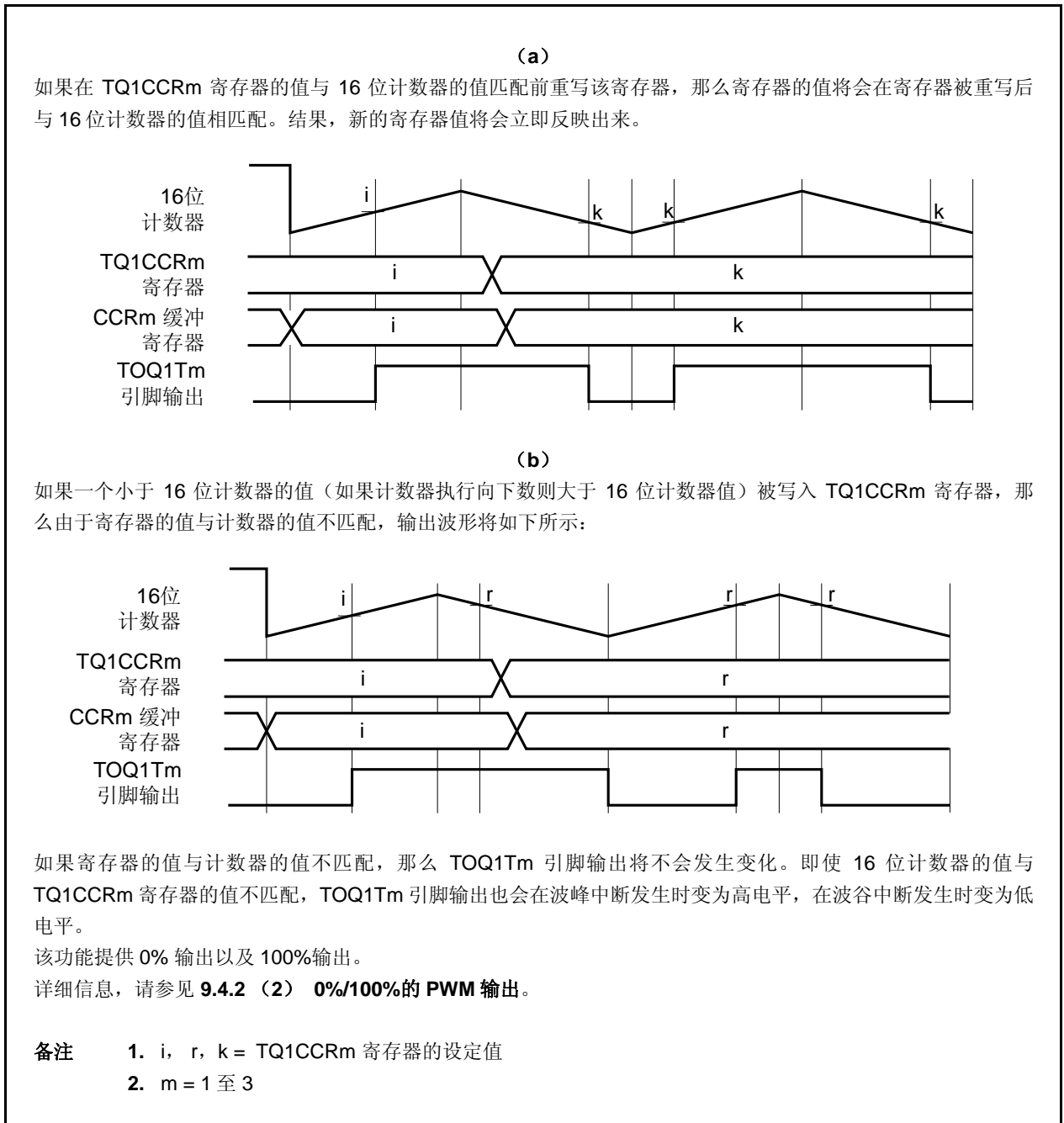
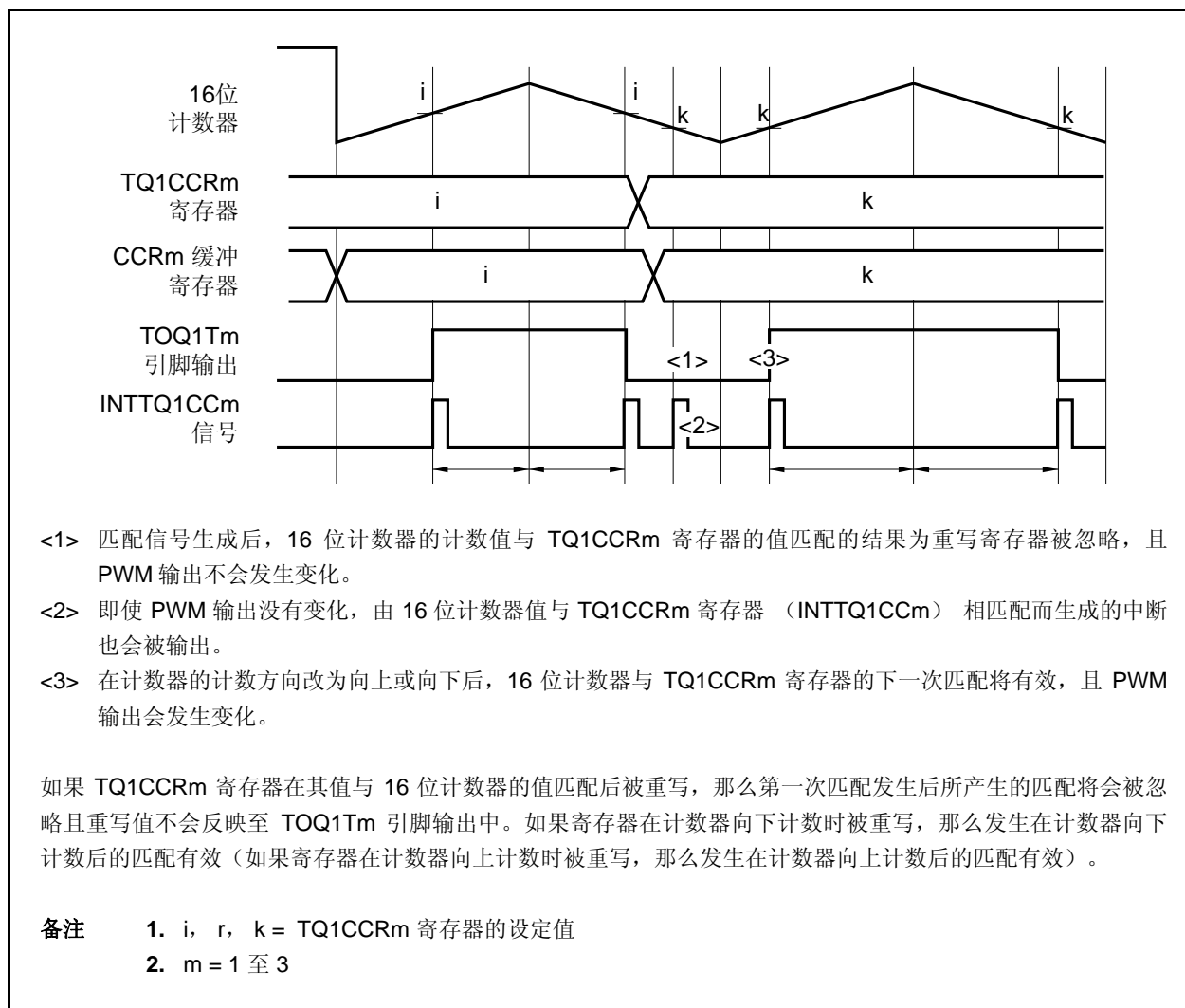


图 9-25. 重写 TQ1CCR1 至 TQ1CCR3 寄存器的示例（匹配后重写）

**(c) 重写 TQ1OPT1 寄存器**

中断选择计数器在 TQ1OPT1 寄存器写入时被清除。中断选择计数器被清除后，所检测到的已发生的中断次数被丢弃。结果，中断产生间隔会暂时延长。

为了避免该操作，应在间隙批量重写模式（传送选择模式）中重写 TQ1OPT1 寄存器。

关于重写 TQ1OPT1 寄存器的详细信息，请参见 9.4.3 中断选择功能。

(2) 批量重写模式（传送模式）

通过将 TQ1OPT0.TQ1CMS 位清 0，TQ1OPT1.TQ1ID4 至 TQ1OPT1.TQ1ID0 位清除为 00000 以及 TQ1OPT2.TQ1RDE 位清 0 可以设置该模式。

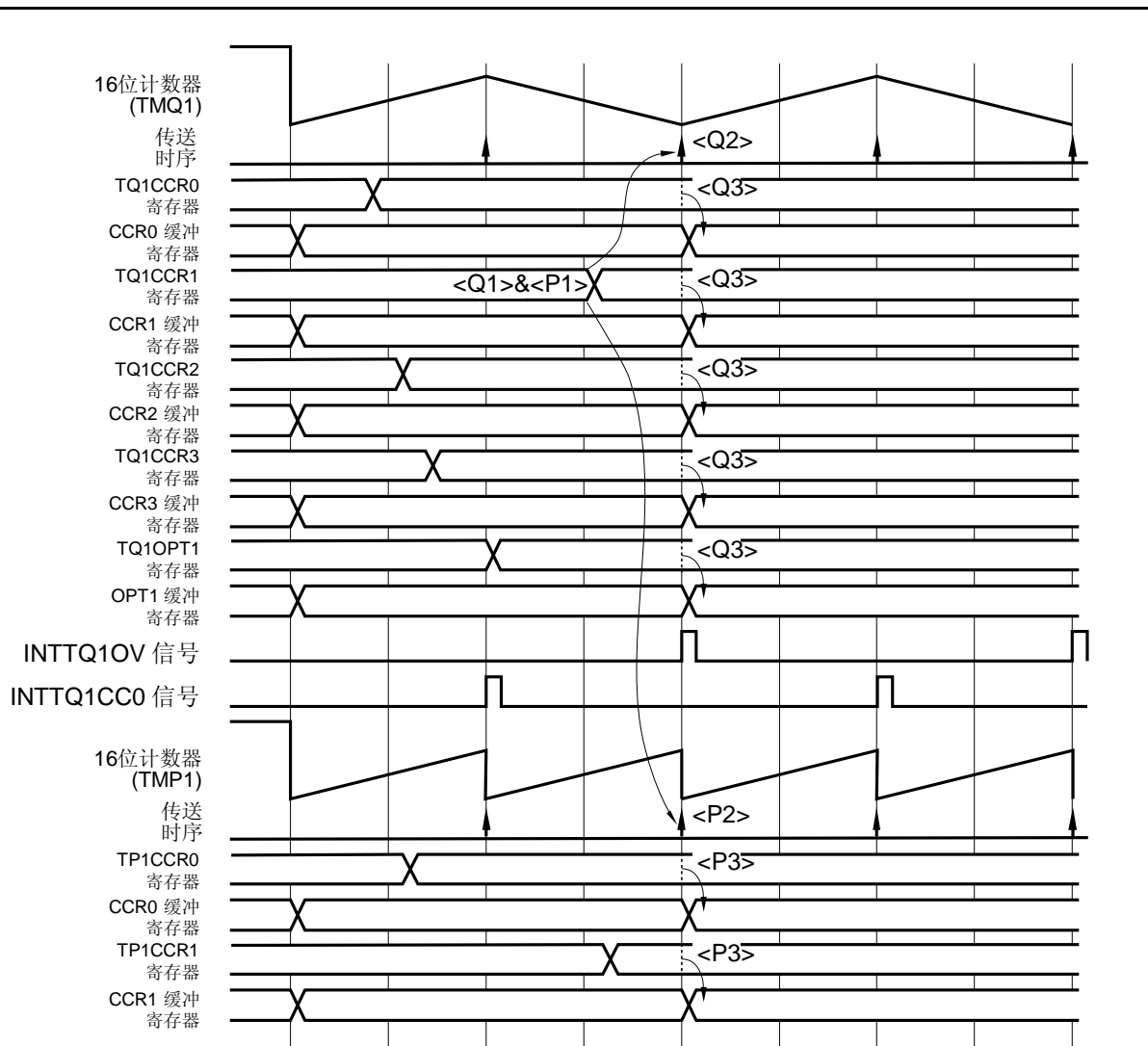
在该模式中，在传送时序处写入各个比较寄存器的值会被立即传送到内部缓冲寄存器中并会与计数器值进行比较。

(a) 重写步骤

如果数据被写入 TQ1CCR1 寄存器，那么在下一个传送时序处，设置在 TQ1CCR0 至 TQ1CCR3，TQ1OPT1，TP1CCR0 以及 TP1CCR1 寄存器中的值会被立即传送到内部缓冲寄存器中。因此，应最后写入 TQ1CCR1 寄存器。传送时序生成前，禁止在写入 TQ1CCR1 寄存器后对寄存器进行写操作（直到波峰（16 位计数器值与 TQ1CCR0 寄存器值匹配）或波谷（16 位计数器值与 0001H 匹配））。操作步骤如下：

- <1> 对 TQ1CCR0，TQ1CCR2，TQ1CCR3，TQ1OPT1，TP1CCR0 以及 TP1CCR1 寄存器进行重写。
不要对没必要重写的寄存器进行重写。
- <2> 对 TQ1CCR1 寄存器进行重写。
即使没有必要重写 TQ1CCR1 寄存器，也应将相同的值重写到该寄存器中。
- <3> 保持下次重写为未决状态直至传送时序生成。
发生 INTTQ1OV 或 INTTQ1CC0 后对寄存器再次进行重写。
- <4> 返回步骤<1>。

图 9-26. 批量写入模式中的基本操作



[TMQ1 的操作]

- $\langle Q1 \rangle$ 对 TQ1CCR1 寄存器进行写操作。
- $\langle Q2 \rangle$ 在对 TQ1CCR1 寄存器执行写操作后，目标时序将会是第一个传送时序。
- $\langle Q3 \rangle$ 在传送时序处数值会立即被传送。

[TMP1 的操作]

- $\langle P1 \rangle$ 对 TP1CCR1 寄存器进行写操作。
- $\langle P2 \rangle$ 在对 TP1CCR1 寄存器执行写操作后，目标时序将会是第一个传送时序。
- $\langle P3 \rangle$ 在传送时序处数值会立即被传送。

(b) 重写 TQ1CCR0 寄存器

在批量重写模式中对 TQ1CCR0 寄存器进行重写时，根据传送是发生在波峰（16 位计数器值与 TQ1CCR0 寄存器值匹配）还是在波谷（16 位计数器值与 0001H 匹配），输出波形会有所不同。通常情况下，建议在 16 位计数器进行向下计数时重写 TQ1CCR0 寄存器，并在波谷的传送时序处传送寄存器的值。

图 9-28 显示了 16 位计数器向上计数时对 TQ1CCR0 寄存器进行重写的例子（图 9-27 的期间<1>中）。图 9-29 显示了 16 位计数器向下计数时对 TQ1CCR0 寄存器进行重写的例子（图 9-27 的期间<2>中）。

图 9-27. 16 位计数器的基本操作

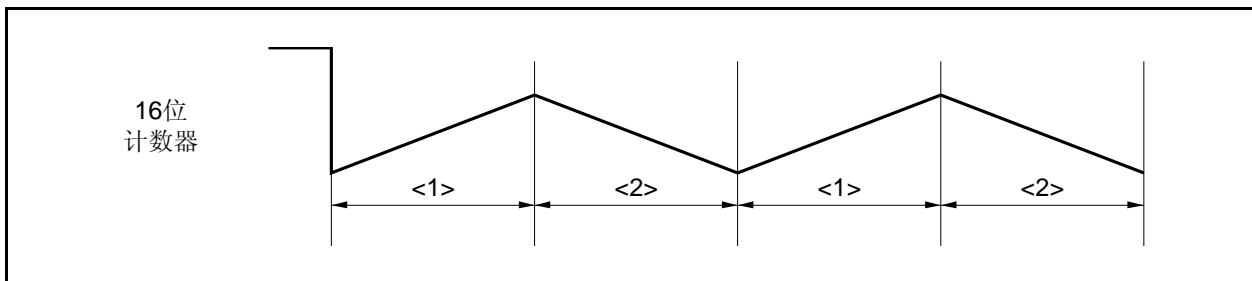


图 9-28 中的传送时序是发生波峰时序的位置。16 位计数器向下计数时，周期将会发生变化并会输出一个不对称的三角波。由于周期发生变化，因此需要重写占空比（电压数据值）。

图 9-28. 重写 TQ1CCR0 寄存器的示例（向上计数期间）

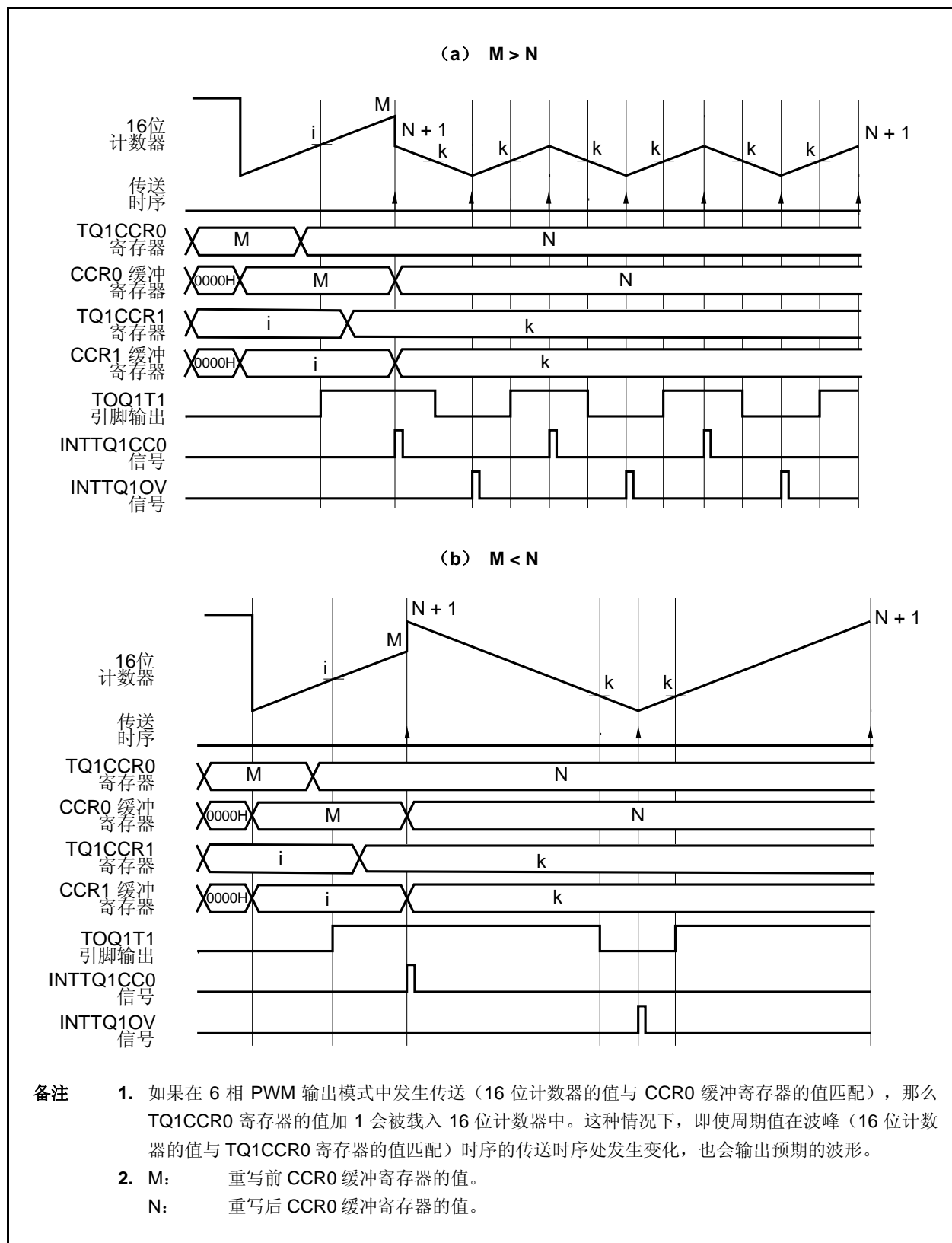
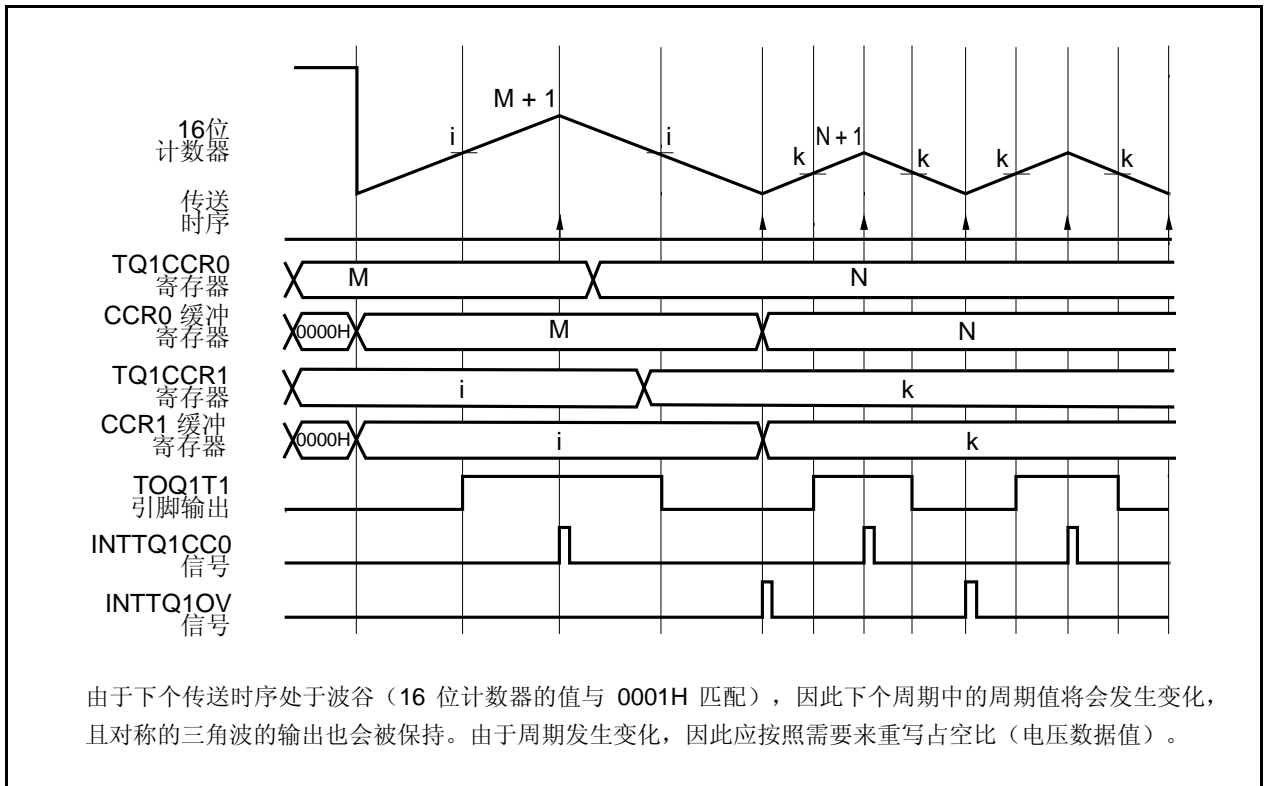
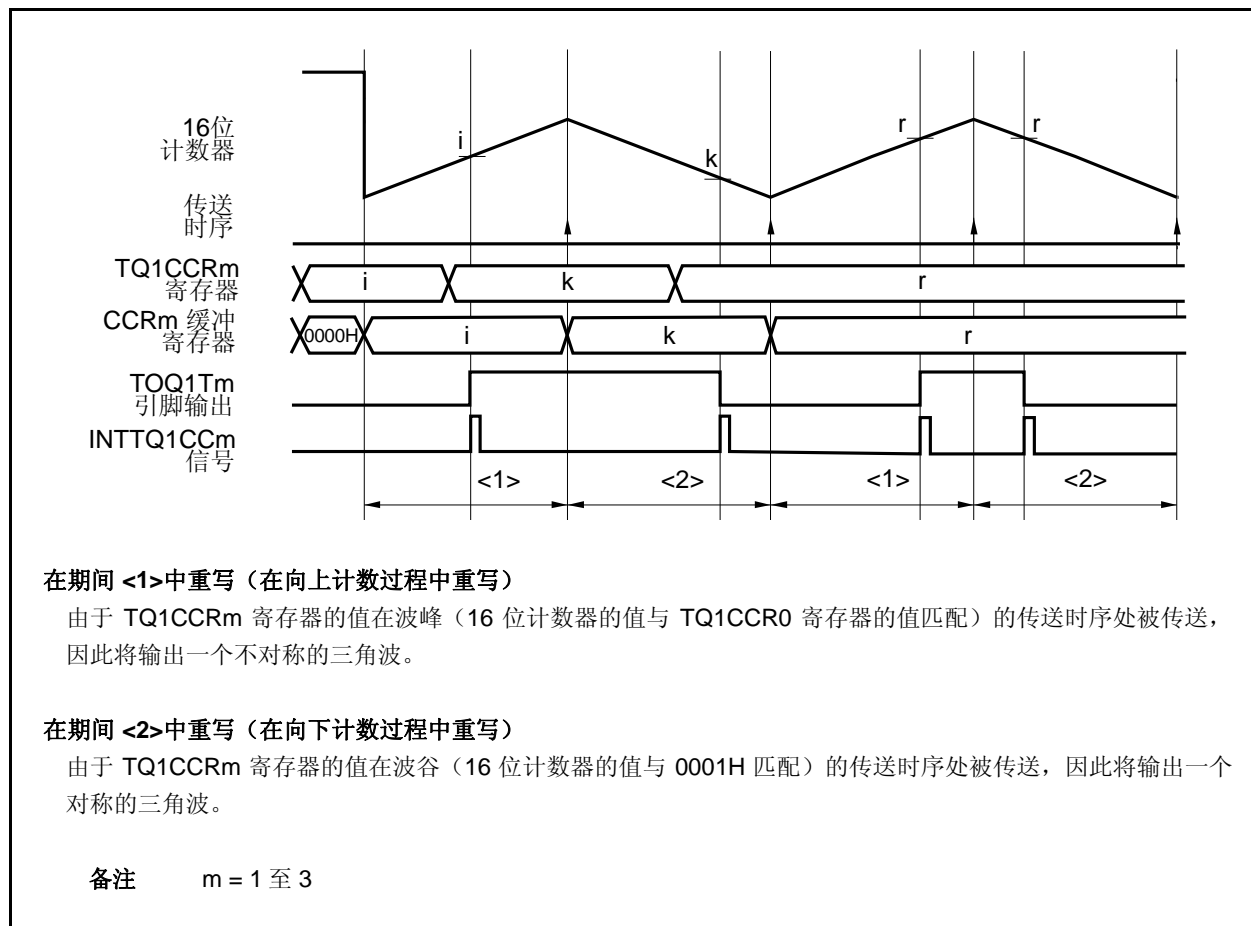


图 9-29. 重写 TQ1CCR0 寄存器的示例（向下计数期间）



(c) 重写 TQ1CCRm 寄存器

图 9-30. 重写 TQ1CCRm 寄存器的示例



(d) 传送 TQ1OPT1 寄存器值

不要将 TQ1OPT1.TQ1ID4 至 TQ1OPT1.TQ1ID0 位设为不同于 00000B 的值。使用中断选择功能时, 应在间隙批量重写模式 (传送选择模式) 中重写 TQ1OPT1 寄存器。

关于重写 TQ1OPT1 寄存器的详细信息, 请参见 9.4.3 中断选择功能。

(3) 间隙批量重写模式（传送选择模式）

该模式可以通过将 TQ1OPT0.TQ1CMS 位清零并将 TQ1OPT2.TQ1RDE 位设为 1 来进行设置。

在该模式中，在选择传送时序处写入各个比较寄存器的值会被立即传送到内部缓冲寄存器中并会与计数器值进行比较。传送时序是通过中断选择生成中断（INTTQ1CC0，INTTQ1OV）时的时序。

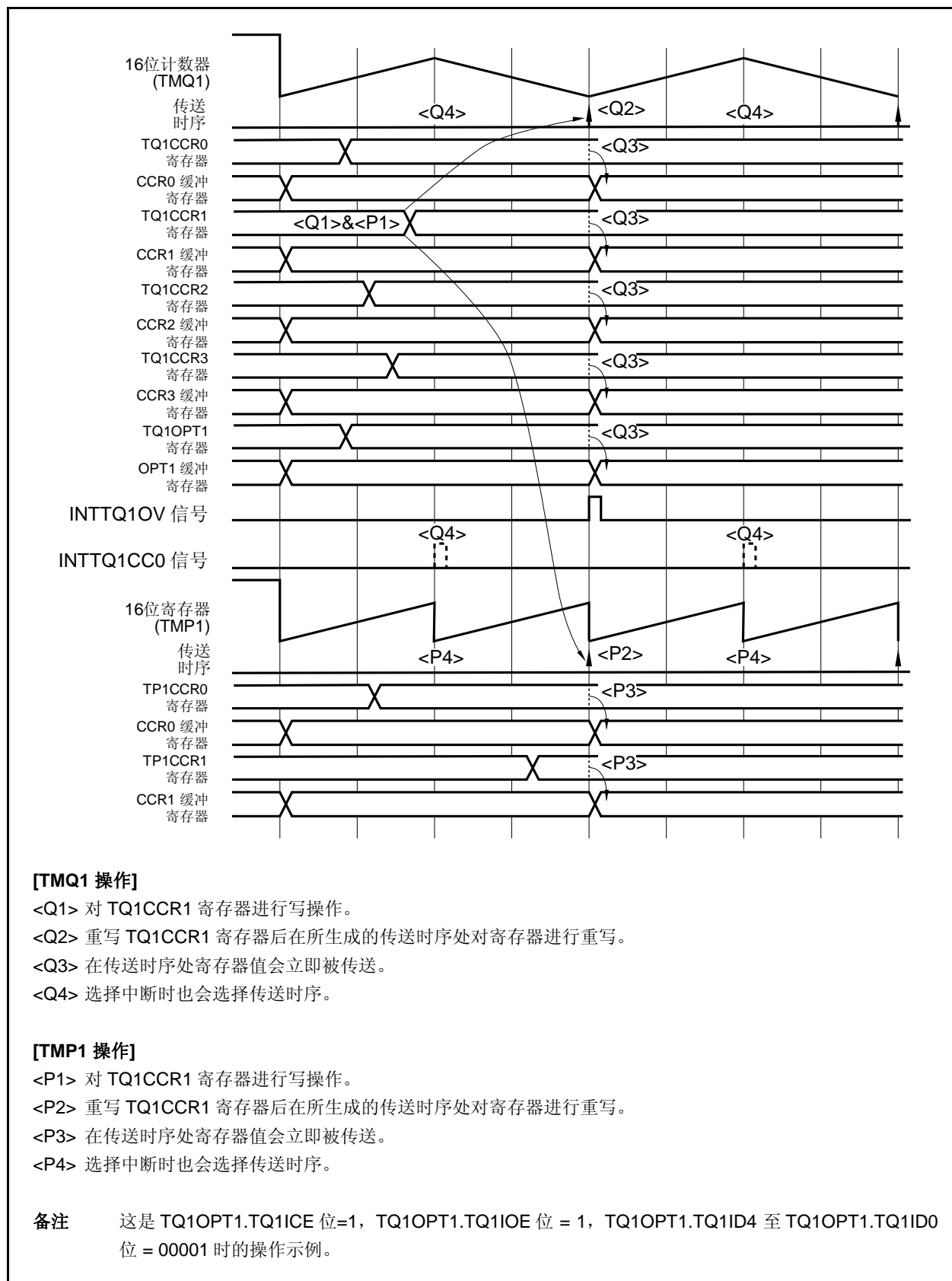
关于中断选择功能的详细信息，请参见 9.4.3 中断选择功能。

(a) 重写步骤

如果数据被写入 TQ1CCR1 寄存器，那么在下一个传送时序处，TQ1CCR0 至 TQ1CCR3，TQ1OPT1，TP1CCR0 以及 TP1CCR1 寄存器中的值会被立即传送到内部缓冲寄存器中。因此，应最后写入 TQ1CCR1 寄存器。写入 TQ1CCR1 寄存器后禁止写入寄存器直到生成传送时序为止（直到产生 INTTQ1OV 或 INTTQ1CC0 中断）。操作步骤如下：

- <1> 对 TQ1CCR0，TQ1CCR2，TQ1CCR3，TQ1OPT1，TP1CCR0 以及 TP1CCR1 寄存器进行写操作。
不要对没必要重写的寄存器进行重写。
- <2> 对 TQ1CCR1 寄存器进行重写。
即使没有必要重写 TQ1CCR1 寄存器，也应将相同的值重写到该寄存器中。
- <3> 保持下次重写为未决状态直至传送时序生成。
发生 INTTQ1OV 或 INTTQ1CC0 中断后对寄存器再次进行重写。
- <4> 返回步骤<1>。

图 9-31. 间隙批量写入模式中的基本操作



(b) 重写 TQ1CCR0 寄存器

在间隙批量写入模式中重写 TQ1CCR0 寄存器时，输出波形会根据中断选择设置所指定的波峰或波谷中断发生的位置而变化。下图说明了中断被选择时输出波形的变化。

图 9-32. 重写 TQ1CCR0 寄存器（设为波峰中断时）

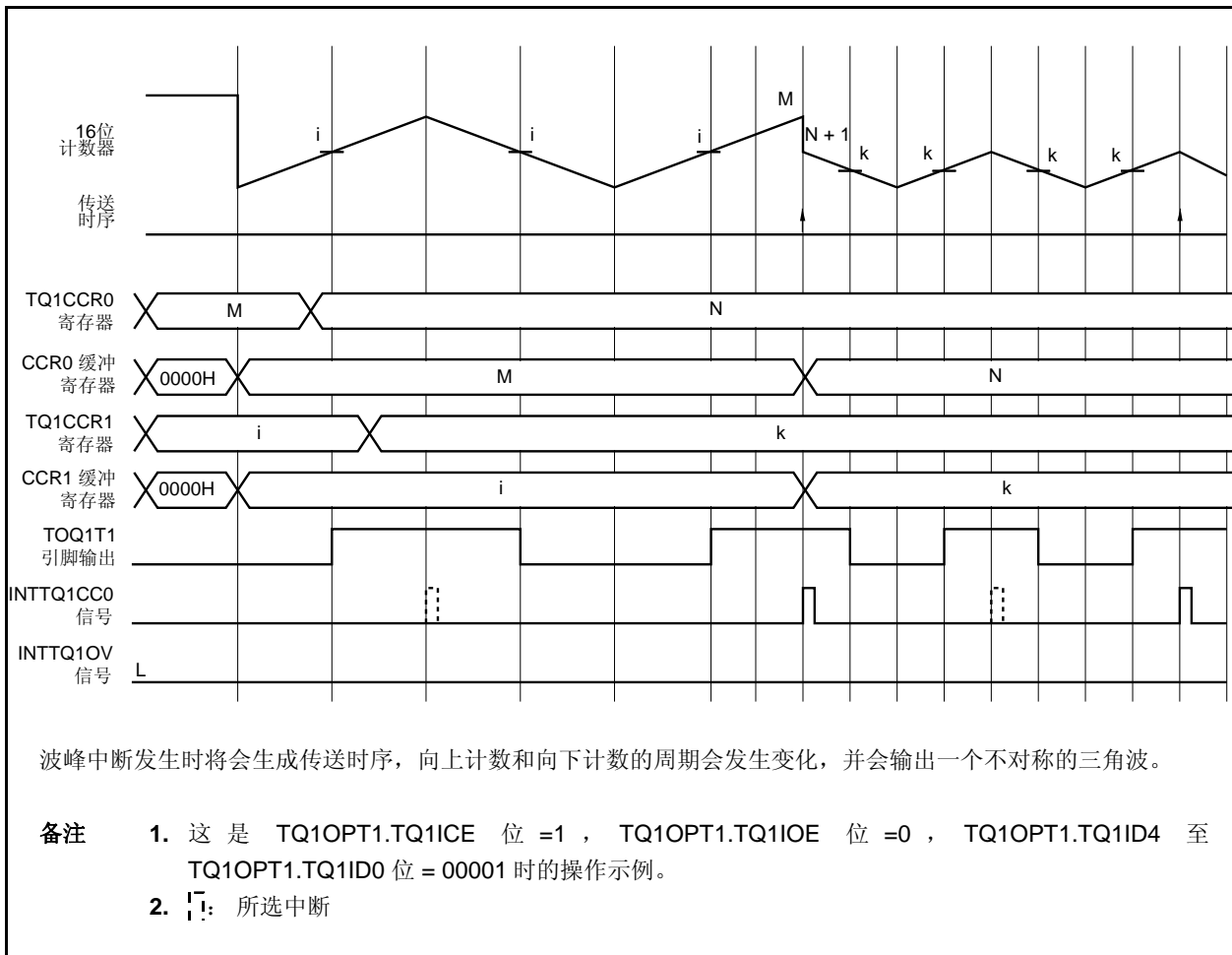
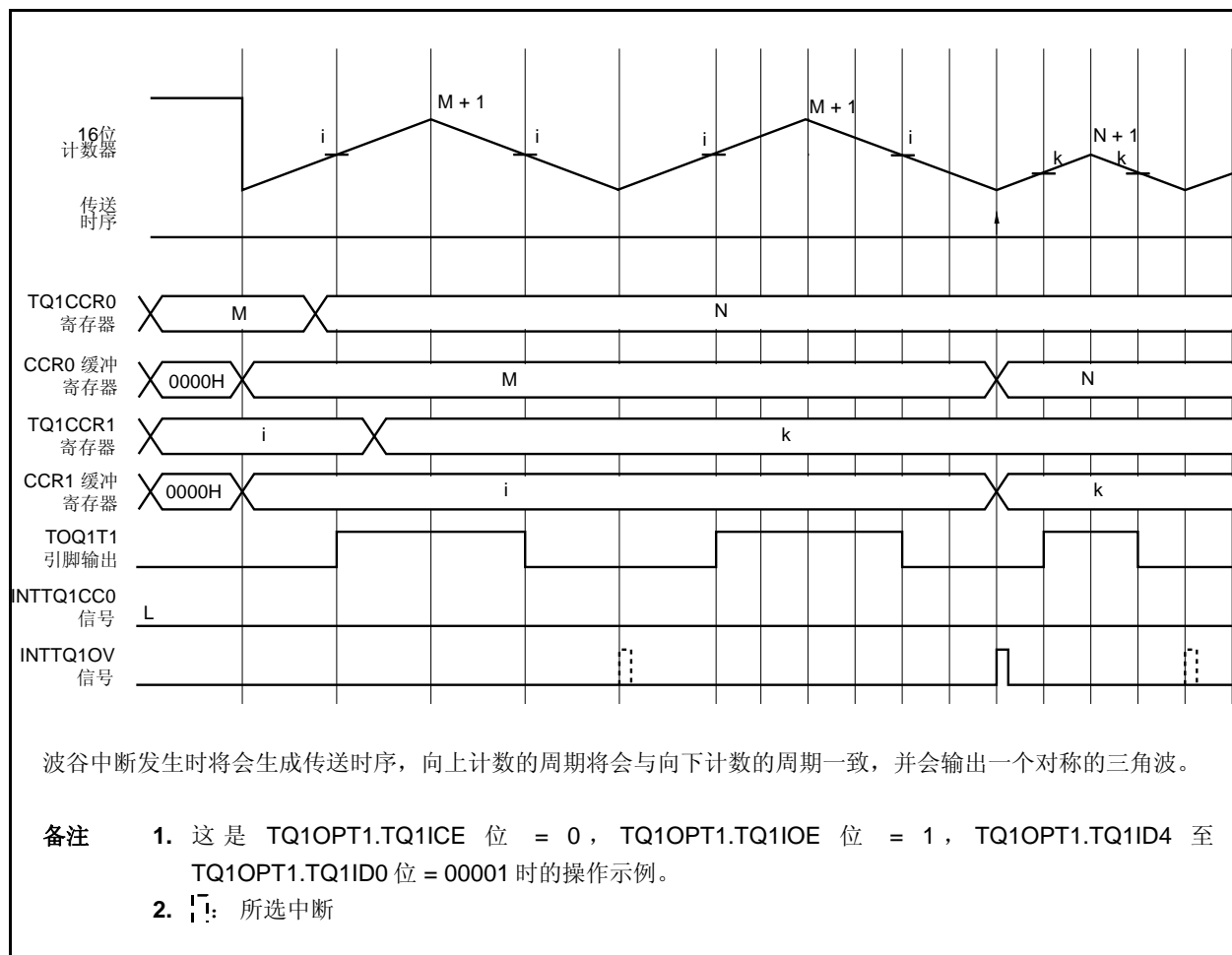


图 9-33. 重写 TQ1CCR0 寄存器（设为波谷中断时）



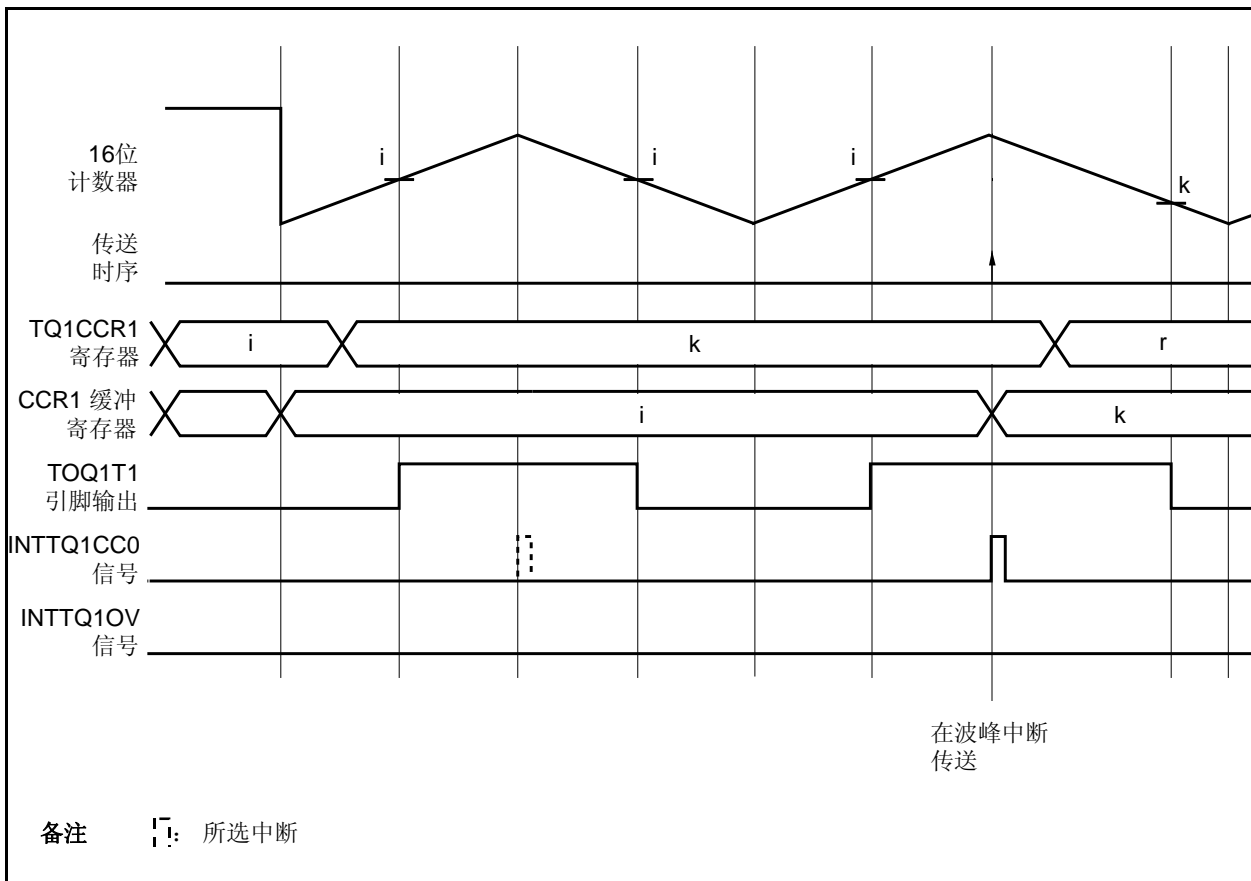
(c) 重写 TQ1CCR1 至 TQ1CCR3 寄存器

- 设为波峰中断时在波峰传送

由于寄存器在波峰中断的发送时序处被传送，因此会输出一个不对称的三角波。

图 9-34. 重写 TQ1CCR1 寄存器

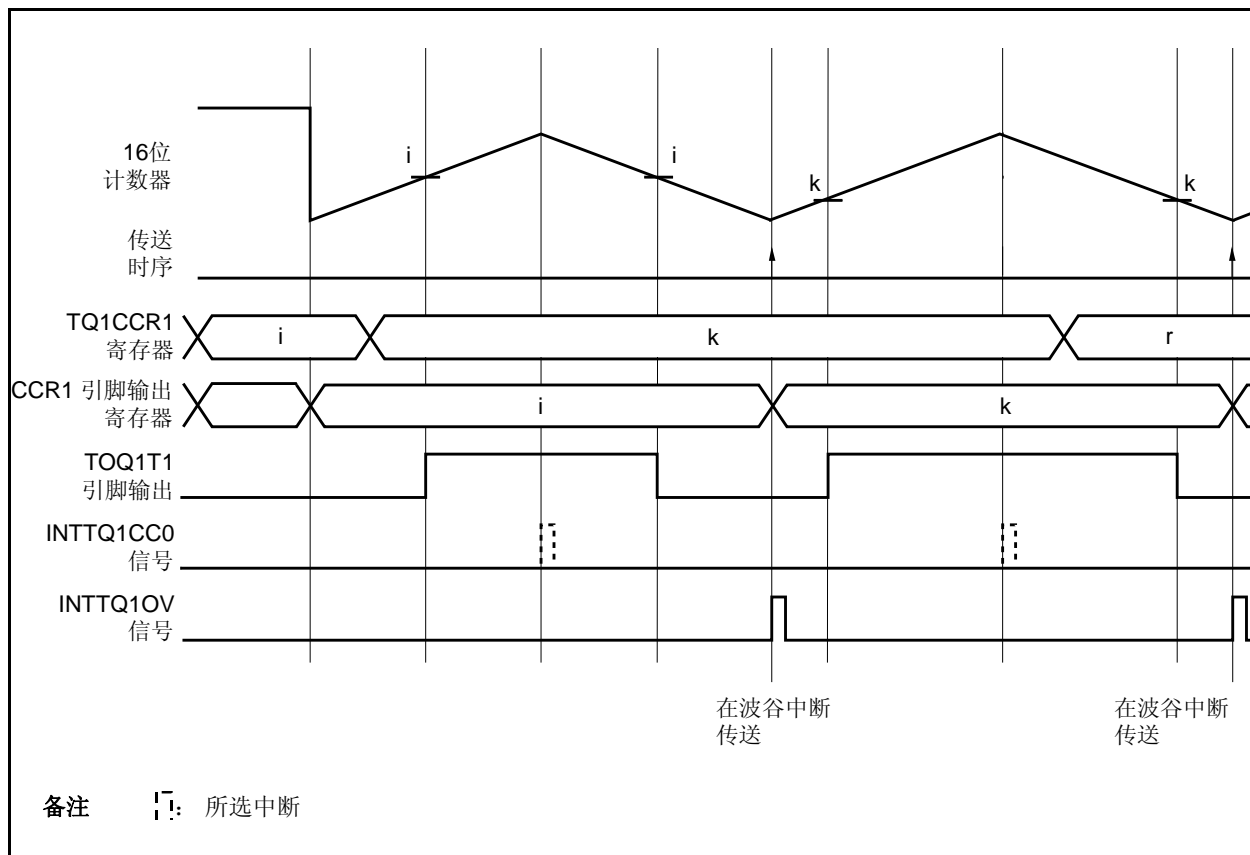
(TQ1OPT1.TQ1ICE 位 = 1, TQ1OPT1.TQ1IOE 位 = 0, TQ1OPT1.TQ1ID4 至 TQ1OPT1.TQ1ID0 = 00001)



- 设为波谷中断时在波谷传送
由于寄存器在波谷中断的发送时序处被传送，因此会输出一个对称的三角波。

图 9-35. 重写 TQ1CCR1 寄存器

(TQ1OPT1.TQ1ICE 位 = 1, TQ1OPT1.TQ1IOE 位 = 1, TQ1OPT1.TQ1ID4 至 TQ1OPT1.TQ1ID0 = 00001)



(d) 重写 TQ1OPT1 寄存器

由于新的中断选择值会在中断选择计数器的值与 16 位计数器的值匹配时被传送，因此下次中断以及之后的中断会在设定的间隔时间发生。

关于重写 TQ1OPT1 寄存器的详细信息，请参见 9.4.3 中断选择功能。

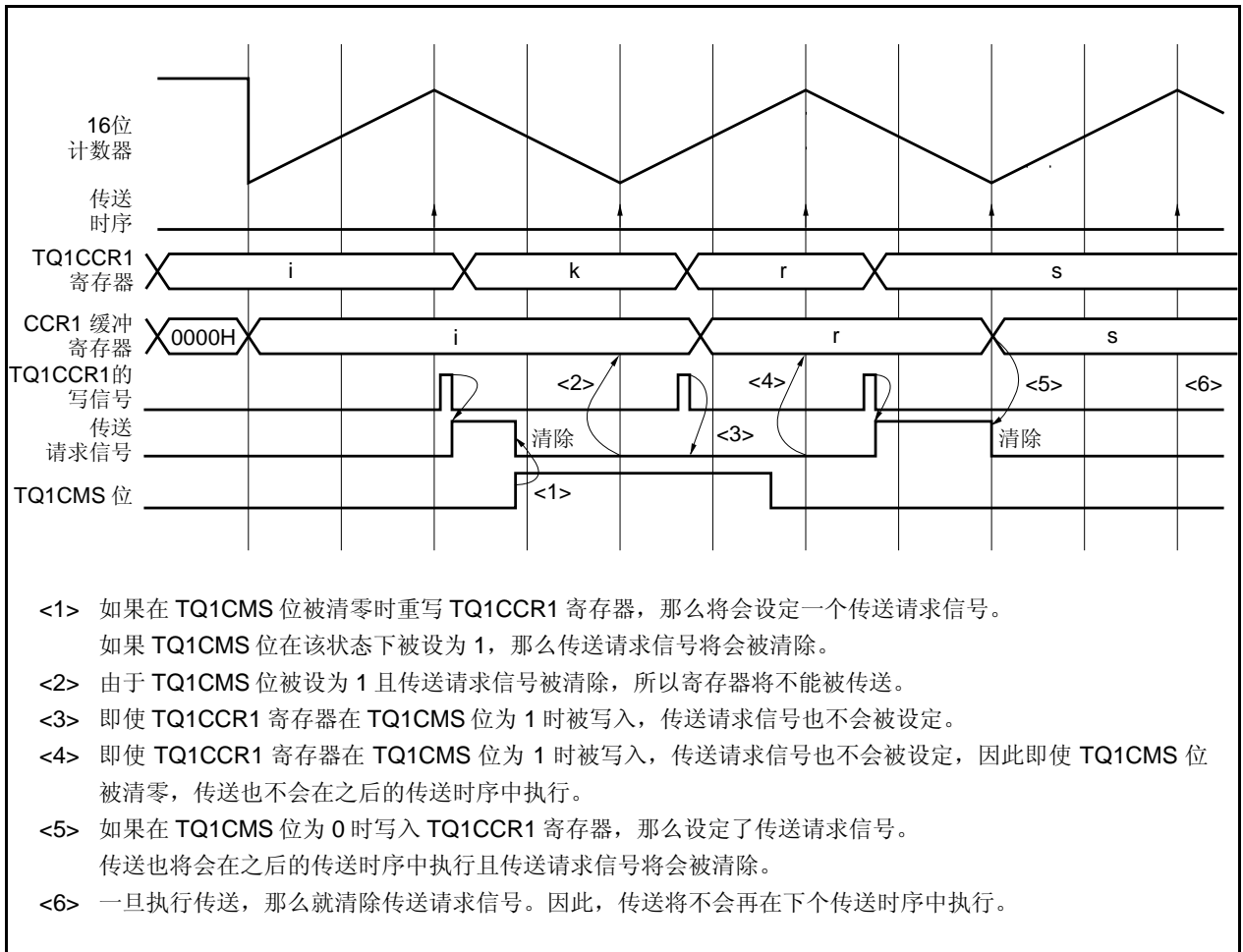
(4) 重写 TQ1OPT0.TQ1CMS 位

TQ1CMS 位可以选择随时重写模式和批量重写模式。该位可以在定时器操作过程中 (TQ1CTL0.TQ1CE 位 = 1 时) 进行重写。然而, 图 9-31 中说明的操作和注意事项是必需的。

如果在 TQ1CMS 位被清零时写入 TQ1CCR1 寄存器, 那么将会设定一个传送请求信号 (内部信号)。

当传送请求信号被设定时, 寄存器将会在下个传送时序处发送, 且传送请求信号将会被清除。当 TQ1CMS 位被设为 1 时该传送请求信号也将会被清除。

图 9-36. 重写 TQ1CMS 位



9.4.5 A/D转换开始触发信号输出的TMP1 调制操作

本节介绍在 6 相 PWM 输出模式中 TMP1 和 TMQ1 的调制操作。

在 6 相 PWM 输出模式中，TMQ1 作为主定时器而 TMP1 作为从定时器来执行调制操作。A/D 转换器 0 和 1 的转换开始触发信号可以通过 TMP1 的 INTTP1CC0 和 INTTP1CC1 信号以及 TMQ1 的 INTTQ1OV 和 INTTQ1CC0 信号来设为 A/D 转换开始触发源。

(1) 调制操作开始步骤

应该使用以下步骤来设置 TMP1 和 TMQ1 寄存器用来完成调制操作。

(a) TMP1 寄存器的设置（停止 TMQ1 和 TMP1 的操作（通过将 TQ1CTL0.TQ1CE 位和 TP1CTL0.TP1CE 位清零））

- 将 TP1CTL1 寄存器设为 85H（设置调制操作从模式以及自运行定时器模式）。
- 将 TP1OPT0 寄存器清除为 00H（选择比较寄存器）。
- 为 TP1CCR0 和 TP1CCR1 寄存器设置一个适当的值（为开始操作的比较设定默认值）。

(b) TMQ1 寄存器的设置

- 将 TQ1CTL1 寄存器设为 07H（设置主模式以及 6 相 PWM 输出模式）。
- 为 TQ1IOC0 寄存器设置一个适当的值（设置 TOQ1T1 至 TOQ1T3 的输出模式）。然而，应将 TQ1OL0 位清零并将 TQ1OE0 位设为 1（允许正相输出）。除非完成该设置，否则波峰中断（INTTQ1CC0）和波谷中断（INTTQ1OV）都不能产生。因此，不能正确生成 A/D 转换器 0 和 1 的转换开始触发信号。
- 将 TQ1OPT0 寄存器清除为 00H（选择比较寄存器）。
- 为 TQ1CCR0 至 TQ1CCR3 寄存器设置一个适当的值（为开始操作的比较设定默认值）。
- 将 TQ1CTL0 寄存器设为 0xH（将 TQ1CE 位清零并设置 TMQ1 的操作时钟）。
由 TQ1CTL0 寄存器设置的 TMQ1 的操作时钟也会提供给 TMP1，并会在相同时序执行计数操作。忽略由 TP1CTL0 寄存器设置的 TMP1 的操作时钟。

(c) TMQOP1（TMQ1 选项）寄存器的设置

- 为 TQ1OPT1 和 TQ1OPT2 寄存器设置一个适当的值。
- 为 TQ1IOC3 寄存器设置一个适当的值（在输出模式中设置 TOQ1B1 至 TOQ1B3）。
- 为 TQ1DTC 寄存器设置一个适当的值（为开始操作的比较设定默认值）。

(d) 复用功能的设置

- 通过设置端口控制模式来设置端口的复用功能。

(e) 开始 6 相 PWM 输出操作后立即将 TP1CE 位和 TQ1CE 位设为 1

禁止在操作期间对 TQ1CTL0, TQ1CTL1, TP1CTL0, 以及 TP1CTL1 进行重写。如果在操作期间对这些寄存器中的任一寄存器进行重写，操作以及 PWM 输出波形将不能被保证。然而，允许对 TQ1CTL0.TQ1CE 位进行清零的重写操作。禁止操作（读取 / 写入）其他 TMQ1, TMP1 以及 TMQ1 选项寄存器直到先将 TP1CTL0.TP1CE 位设为 1，再将 TQ1CE 位设为 1 为止。

(2) 调制操作清除步骤

要清除调制操作并退出 6 相 PWM 输出模式，应使用以下步骤来设置 TMP1 和 TMQ1 寄存器。

- <1> 将 TQ1CTL0.TQ1CE 位清零并停止定时器操作。
- <2> 将 TP1CTL0.TP1CE 位清零使得可以分离 TMP1。
- <3> 通过使用 TQ1IOC0 寄存器来停止定时器输出。
- <4> 将 TP1CTL1.TP1SYE 位清零以清除调制操作。

注意事项 禁止操作（读取 / 写入）其他 **TMQ1**，**TMP1** 以及 **TMQ1** 选项寄存器直到先将 **TQ1CE** 位设为 **0**，再将 **TP1CE** 位设为 **0** 为止。

(3) 没有调制 TMP1 时

当 TMP1 的匹配中断信号不需要作为启动 A/D 转换器 0 和 1 的转换触发源时，TMP1 可以用作一个不需要调制的独立的定时器。在这种情况下，TMP1 的匹配中断信号不能在 6 相 PWM 输出模式中用作启动 A/D 转换的触发源。因此，应将 TQ1OPT2.TQ1AT00 至 TQ1OPT2.TQ1AT03 位以及 TQ1OPT3.TQ1AT10 至 TQ1OPT3.TQ1AT13 位固定为 0。

其他控制位可以按调制 TMP1 时的相同的方式来使用。

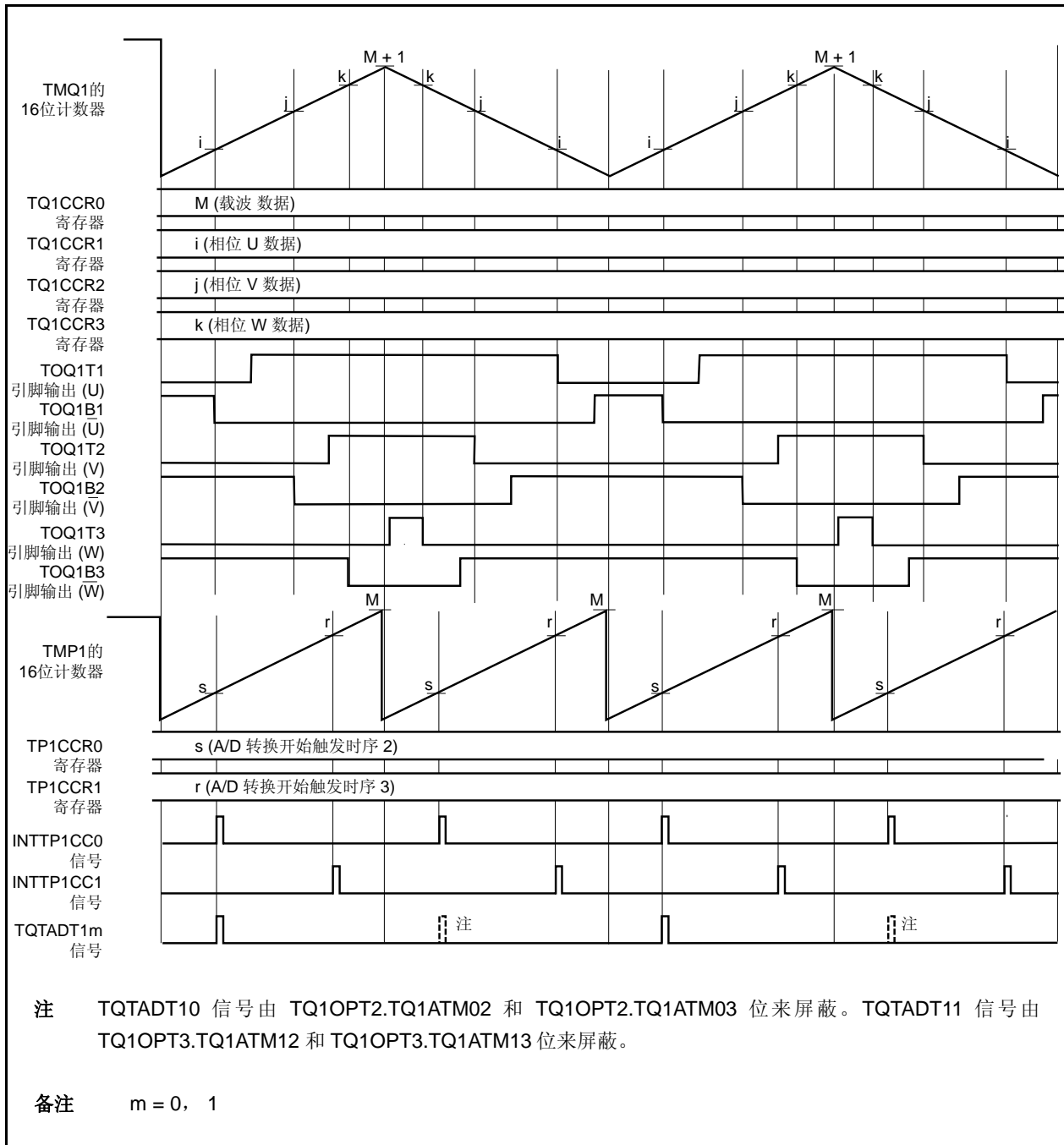
如果 TMP1 没有被调制，那么 TMP1 的比较寄存器（TP1CCR0 和 TP1CCR1）将不会受到 TQ1OPT0.TQ1CMS 和 TQ1OPT2.TQ1RDE 位的设置的影响。关于 TMP1 没有被调制时的初始化步骤，请参见 **9.4.5 (1) 调制操作开始步骤** 中的 (b) 至 (e)。不需要步骤 (a)，因为它是用于为调制操作设置 TMP1 的步骤。

(4) 调制操作过程中 TMP1 的基本操作

TMP1 的 16 位计数器只能向上计数。16 位计数器通过 TQ1CCR0 寄存器的设定周期值来清除，并从 0000H 开始再次计数。当该计数器开始计数时，它的计数值与 TMQ1 的 16 位计数器的值相同。然而，它的计数值在 TMQ1 的 16 位计数器向下计数时将不同于 16 位计数器的值。

- TMQ1 向上计数时（相同的值）
 - TMQ1 的 16 位计数器：0000H → M（向上计数）
 - TMP1 的 16 位计数器：0000H → M（向上计数）
- TMQ1 向下计数时（不相同的值）
 - TMQ1 的 16 位计数器：M + 1 → 0001H（向下计数）
 - TMP1 的 16 位计数器：0000H → M（向上计数）

图 9-37. 调制操作期间的 TMP1



9.4.6 A/D转换开始触发输出功能

V850ES/IE2 具有四种可选触发源（INTTQ1OV, INTTQ1CC0, INTTP1CC0, INTTP1CC1）以生成 A/D 转换器 0 和 1 的 A/D 转换开始触发信号（TQTADT00, TQTADT01）。

触发源通过 TQ1OPT2.TQ1AT00 至 TQ1OPT2.TQ1AT03 位以及 TQ1OPT3.TQ1AT10 至 TQ1OPT3.TQ1AT13 位来指定。

- TQ1AT00, TQ1AT10 位 = 1:
发生 INTTQ1OV（计数器下溢）时生成 A/D 转换开始触发信号。
- TQ1AT01, TQ1AT11 位 = 1:
发生 INTTQ1CC0（周期匹配）时生成 A/D 转换开始触发信号。
- TQ1AT02, TQ1AT12 位 = 1:
发生 INTTP1CC0（调制操作期间 TMP1 的 TP1CCR0 寄存器的匹配）时生成 A/D 转换开始触发信号。
- TQ1AT03, TQ1AT13 位 = 1:
发生 INTTP1CC1（调制操作期间 TMP1 的 TP1CCR1 寄存器的匹配）时生成 A/D 转换开始触发信号。

由 TQ1AT00 至 TQ1AT03 以及 TQ1AT10 至 TQ1AT13 位选择的 A/D 转换开始触发信号是只读的并且会被输出。因此，可以同时指定两个或更多的触发源。

由 TQ1AT00, TQ1AT01, TQ1AT10 以及 TQ1AT11 位选择的 INTTQ1OV 和 INTTQ1CC0 信号为选中的中断信号。

因此，这些信号会在中断被选择后输出，并且除非允许中断输出（TQ1OPT1.TQ1ICE, TQ1OPT1.TQ1IOE 位），否则将不会输出 A/D 转换开始触发信号。

如果 TQ1AT02, TQ1AT03, TQ1AT12 以及 TQ1AT13 位进行如此设置，则 TMP1 的触发源（INTTP1CC0 和 INTTP1CC1）具有根据 16 位计数器向上计数 / 向下计数的状态来屏蔽 A/D 转换开始触发信号的功能。

- TQ1ATM02, TQ1ATM12 位:
相应于 TQ1AT02 和 TQ1AT12 位，并且控制 TMP1 的 INTTP1CC0（匹配中断信号）。
- TQ1ATM02, TQ1ATM12 位=0
A/D 转换开始触发信号在 16 位计数器向上计数时（TQ1OPT0.TQ1CUF 位 = 0）输出，A/D 转换开始触发信号在 16 位计数器向下计数时（TQ1OPT0.TQ1CUF 位 = 1）不会输出。
- TQ1ATM02, TQ1ATM12 位=1
A/D 转换开始触发信号在 16 位计数器向下计数时（TQ1OPT0.TQ1CUF 位 = 1）输出，A/D 转换开始触发信号在 16 位计数器向上计数时（TQ1OPT0.TQ1CUF 位 = 0）不会输出。
- TQ1ATM03, TQ1ATM13 位:
相应于 TQ1AT03 和 TQ1AT13 位，并且控制 TMP1 的 INTTP1CC1（匹配中断信号）。
- TQ1ATM03, TQ1ATM13 位= 0
A/D 转换开始触发信号在 16 位计数器向上计数时（TQ1OPT0.TQ1CUF 位 = 0）输出，A/D 转换开始触发信号在 16 位计数器向下计数时（TQ1OPT0.TQ1CUF 位 = 1）不会输出。
- TQ1ATM03, TQ1ATM13 位= 1
A/D 转换开始触发信号在 16 位计数器向下计数时（TQ1OPT0.TQ1CUF 位 = 1）输出，A/D 转换开始触发信号在 16 位计数器向上计数时（TQ1OPT0.TQ1CUF 位 = 0）不会输出。

TQ1ATM03, TQ1ATM02, TQ1AT03 至 TQ1AT00, TQ1ATM13, TQ1ATM12 以及 TQ1AT13 至 TQ1AT10 位可以在定时器操作过程中进行重写。如果设置 A/D 转换开始触发信号的位在定时器操作时被重写, 那么新的设置会立即反映在 A/D 转换开始触发信号的输出状态上。这些控制位不具有传送功能, 并且只能在随时重写模式中使用。

- 注意事项**
1. 由 TQ1AT02, TQ1AT03, TQ1AT12 以及 TQ1AT13 位设置的 A/D 转换开始触发信号输出只能在 TMP1 作为 TMQ1 的从定时器来执行调制功能时使用。如果 TMQ1 和 TMP1 都没有执行调制操作, 或者使用了除了 6 相 PWM 输出模式的模式, 那么将无法保证输出。
 2. TOQ10 信号输出用于在内部识别 16 位计数器是向上计数还是向下计数。因此, 通过将 TQ1IOC0.TQ1OL0 位清零并将 TQ1IOC0.TQ1OE0 位设为 1 可以允许 TOQ10 引脚输出。

图 9-38. A/D 转换开始触发 (TQTADT10) 信号输出的示例 (TQ1OPT1.TQ1ICE 位 = 1, TQ1OPT1.TQ1IOE 位 = 1, TQ1OPT1.TQ1ID4 至 TQ1OPT1.TQ1ID0 位 = 00000: 没有中断选择)

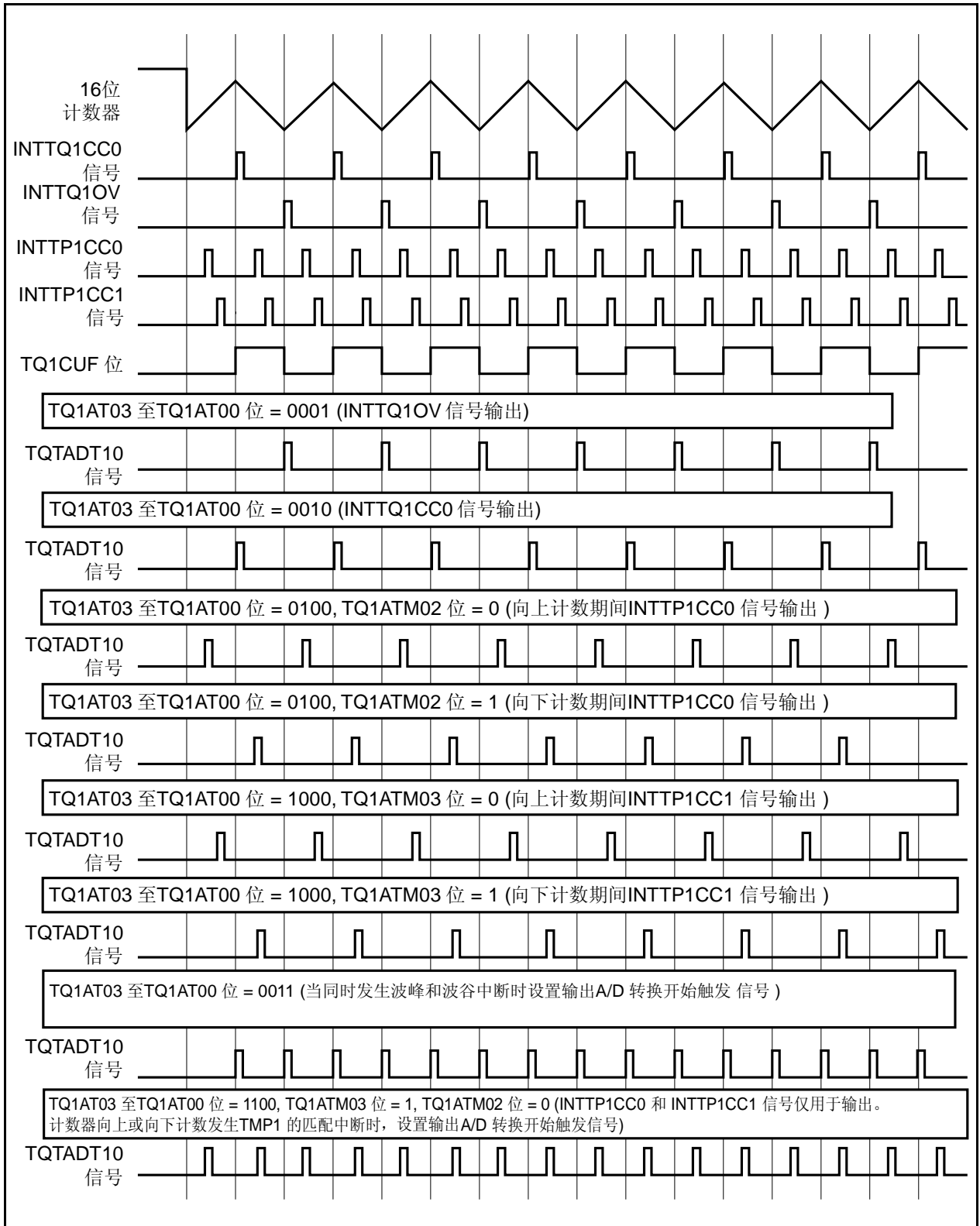


图 9-39. A/D 转换开始触发 (TQTADT10) 信号输出的示例 (TQ1OPT1.TQ1ICE 位 = 0, TQ1OPT1.TQ1IOE 位 = 1, TQ1OPT1.TQ1ID4 至 TQ1OPT1.Q1ID0 位 = 00010: 带有中断选择) (1)

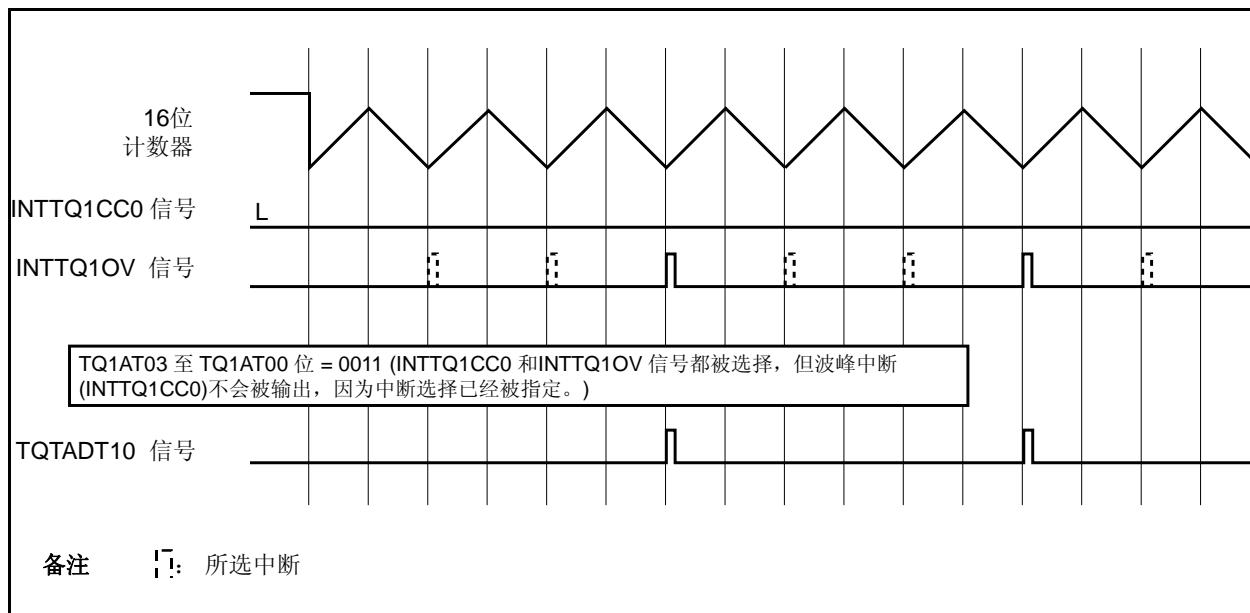
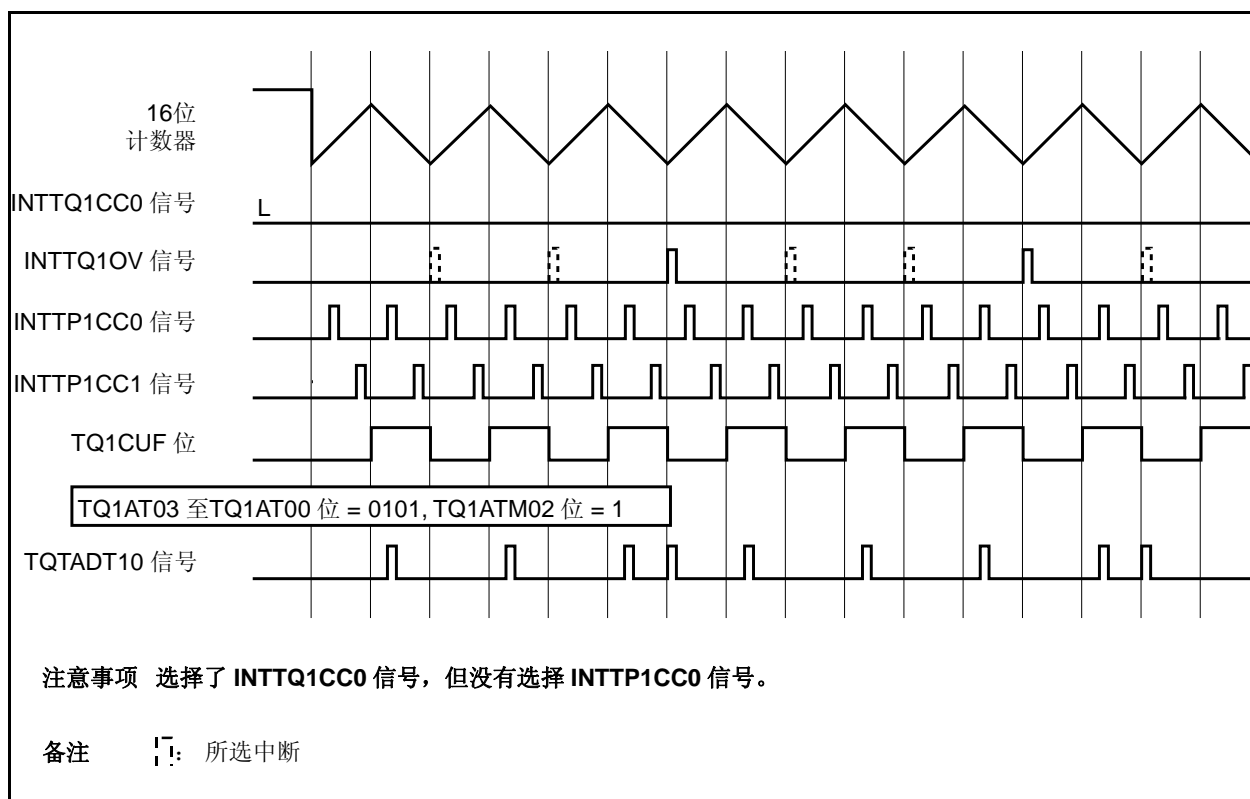


图 9-40. A/D 转换开始触发 (TQTADT10) 信号输出的示例 (TQ1OPT1.TQ1ICE 位 = 0, TQ1OPT1.TQ1IOE 位 = 1, TQ1OPT1.TQ1ID4 至 TQ1OPT1.TQ1ID0 位 = 00010: 带有中断选择) (2)



(1) 边界条件下的操作 (16 位计数器与 INTTP1CC0 信号匹配时的操作)

表 9-3. TQ1CCR0 寄存器 = M, TQ1ATm2 位 = 1, TQ1ATMm2 位 = 0 时的操作
(选择向上计数期间)

TP1CCR0 寄存器的值	TMQ1 的 16 位计数器的值	TMP1 的 16 位计数器的值	TMQ1 的 16 位计数器的状态	由 INTTP1CC0 信号输出的 TQTADT1m 信号
0000H	0000H	0000H	-	输出
0000H	M + 1	0000H	-	不输出
0001H	0001H	0001H	向上计数	输出
0001H	M	0001H	向下计数	不输出
M	M	M	向上计数	输出
M	0001H	M	向下计数	不输出

表 9-4. TQ1CCR0 寄存器 = M, TQ1ATm2 位 = 1, TQ1ATMm2 位 = 1 时的操作
(选择向下计数期间)

TP1CCR0 寄存器的值	TMQ1 的 16 位计数器的值	TMP1 的 16 位计数器的值	TMQ1 的 16 位计数器的状态	由 INTTP1CC0 信号输出的 TQTADT1m 信号
0000H	0000H	0000H	-	不输出
0000H	M + 1	0000H	-	输出
0001H	0001H	0001H	向上计数	不输出
0001H	M	0001H	向下计数	输出
M	M	M	向上计数	不输出
M	0001H	M	向下计数	输出

注意事项 TQ1CCR0 寄存器 = M 时 TP1CCRm 寄存器允许“0”至“M”的设置。但禁止“M + 1”及更高的值的设置。如果设置了一个高于“M + 1”的值，则 TMP1 的 16 位计数器将会通过“M”来清除。因此，将不会输出 TQTADT1m 信号。

备注 m = 0, 1

第十章 看门狗定时器功能

10.1 功能

看门狗定时器具有以下功能：

- 复位模式：看门狗定时器溢出时产生的复位操作（产生 WDTRES 信号）。
- 不可屏蔽中断请求模式：
看门狗定时器溢出时产生的不可屏蔽中断操作（产生 INTWDT 信号）。

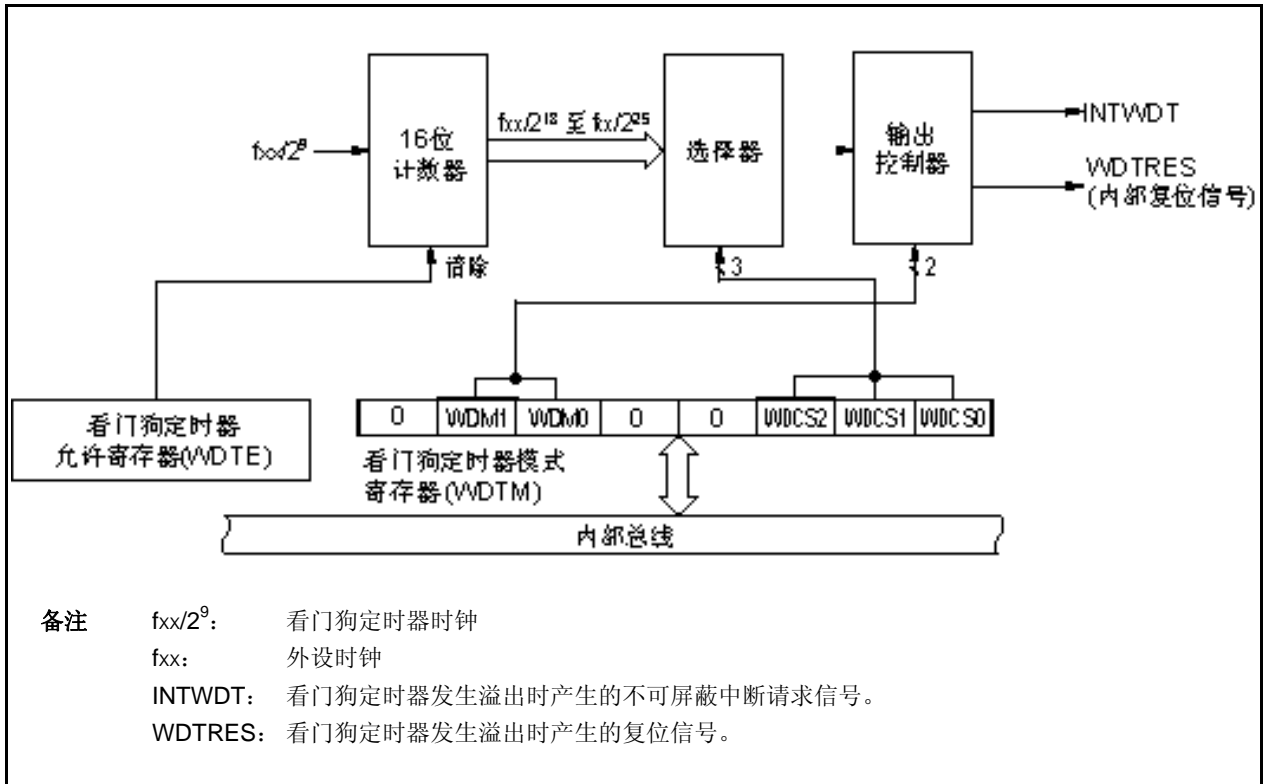
注意事项 复位解除后看门狗定时器将停止。

当“ACH”写入 WDTM 寄存器时启动操作。为了确认操作的正确性，即使在不需要改变默认设置（复位模式，间隔时间： $2^{25}/f_{xx}$ ）的情况下，也要对寄存器 WDTM 执行一次写操作。

10.2 配置

看门狗定时器的框图如下所示：

图 10-1. 看门狗定时器的框图



看门狗定时器包含以下硬件：

表 10-1. 看门狗定时器的配置

名称	配置
控制寄存器	看门狗定时器模式寄存器 (WDTM) 看门狗定时器使能寄存器 (WDTE)

10.3 控制寄存器

(1) 看门狗定时器模式寄存器 (WDTM)

WDTM 寄存器用于设置看门狗定时器的溢出时间和工作时钟。

该寄存器可按字节进行读写。该寄存器可读取任意次，但在复位解除后仅能执行一次写入操作，不能进行两次或更多次写入。

复位输入将该寄存器设置为 67H。

复位后: 67H R/W 地址: FFFFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM	0	WDM1	WDM0	0	0	WDCS2	WDCS1	WDCS0

WDM1	WDM0	看门狗定时器操作模式的选择
0	0	停止操作
0	1	不可屏蔽中断请求模式 (产生INTWDT 信号)
1	X	复位模式(产生 WDTRES 信号)

- 注意事项
1. 有关 WDCS2 位至 WDCS0 位之详情，请参见表 10-2 溢出时间。
 2. 一定要将位 3，位 4 和位 7 清为“0”。

表 10-2. 溢出时间

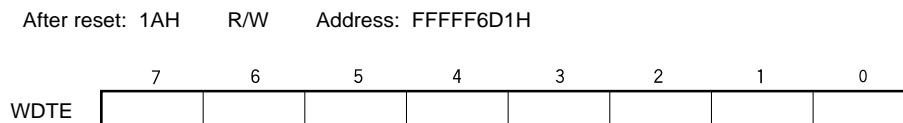
WDCS2	WDCS1	WDCS0	溢出时间	$f_{xx} = 20 \text{ MHz}$
0	0	0	$2^{18}/f_{xx}$	13.1 ms
0	0	1	$2^{19}/f_{xx}$	26.2 ms
0	1	0	$2^{20}/f_{xx}$	52.4 ms
0	1	1	$2^{21}/f_{xx}$	104.9 ms
1	0	0	$2^{22}/f_{xx}$	209.7 ms
1	0	1	$2^{23}/f_{xx}$	419.4 ms
1	1	0	$2^{24}/f_{xx}$	838.9 ms
1	1	1	$2^{25}/f_{xx}$	1677.7 ms

(2) 看门狗定时器使能寄存器(WDTE)

将“ACH”写入 WDTE 寄存器，则看门狗定时器的计数器清零，计数重新开始。

WDTE 寄存器可按字节进行读写。

复位输入将该寄存器设为 1AH。



- 注意事项**
1. 如果先将“ACH”写入 WDTE 寄存器以使能看门狗定时器的操作，然后将不同于“ACH”的值写入 WDTE 寄存器，那么根据 WDTM.WDM1 和 WDTM.WDM0 位的规范，由于看门狗定时器发生了溢出，则将会产生不可屏蔽中断请求信号（INTWDT）或复位信号（WDTRES）。
 2. 当按位读取或写入 WDTE 寄存器时，将输出内部复位信号。
 3. 看门狗定时器工作前，WDTE 寄存器的读取值为“1AH”，工作后其值为“9AH”。从该寄存器中读取的值不同于写入值（ACH）。

10.4 操作

复位解除后看门狗定时器停止工作。

复位解除后，仅能对 WDTM 寄存器进行一次写入操作。如果看门狗定时器开始工作后，第二次写入该寄存器，那么根据 WDTM、WDM1 和 WDTM.WDM0 位的规范，由于看门狗定时器发生了溢出，则将会产生不可屏蔽中断请求信号（INTWDT）或复位信号（WDTRES）。如果至 WDTM 寄存器写入相同值，也将产生 INTWDT 或 WDTRES 信号。如果写入 WDTM 寄存器三次或更多次，则不能保证操作的有效性。

要使用看门狗定时器，应以按字节将操作模式以及间隔时间写入 WDTM 寄存器。之后，将无法停止看门狗定时器的操作。

若不使用看门狗定时器，则将 00H 写入 WDTM 寄存器。

10.5 注意事项

如果在看门狗定时器没有清“0”时连续产生 INTWDT 信号，由于看门狗定时器溢出而产生的不可屏蔽中断请求信号（INTWDT），其周期可以根据“设置至 WDTM 寄存器的间隔时间+ 2⁷ 外设时钟脉宽”来计算。

需注意：从看门狗定时器启动后直到产生第一个中断请求信号时的脉宽不包括在内。

11.1 特性

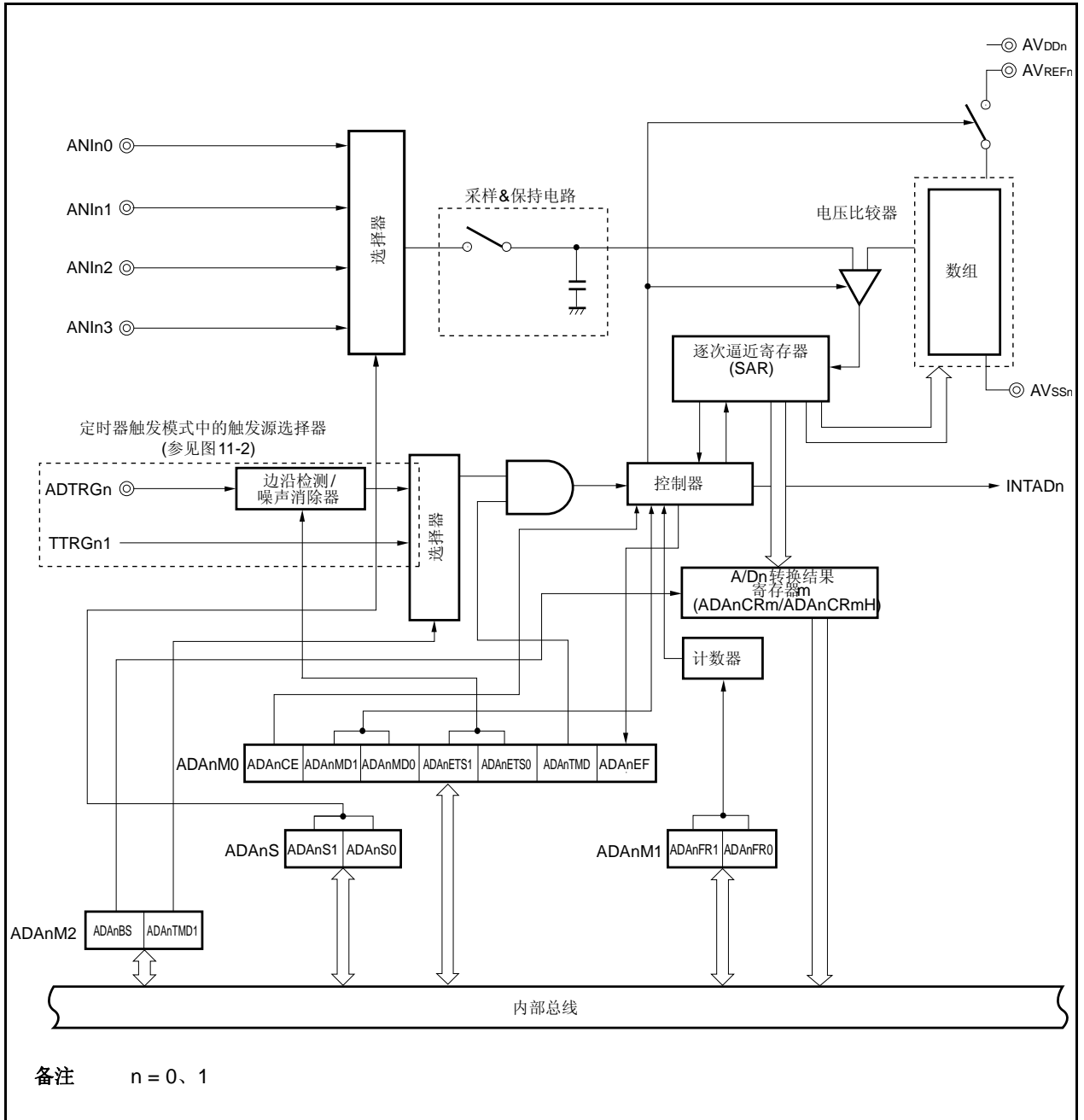
- 两个 10 位分辨率 A/D 转换器电路 (A/D 转换器 0 和 1)
可以对两路输入同时采样
- 模拟输入
两路输入, 总共八个通道
A/D 转换器 0: ANI00 至 ANI03 (4 个通道)
A/D 转换器 1: ANI10 至 ANI13 (4 个通道)
- A/D 转换结果寄存器 0m 和 1m (ADA0CRm 和 ADA1CRm)
10 位 $\times 4 \times 2$
- A/D 转换触发模式
 - 软件触发模式
 - 硬件触发模式
外部触发模式
定时器触发模式
- A/D 转换操作模式
连续选择模式
连续扫描模式
单脉冲选择模式
单脉冲扫描模式
- 缓冲区模式
1 缓冲区模式
4 缓冲区模式
- 逐次逼近法
- 操作电压范围
 $V_{DD} = EV_{DD} = AV_{DDn} = AV_{REFn} = 4.5$ 至 5.5 V

Remark m = 0 至 3
n = 0、1

11.2 配置

框图如下所示：

图 11-1. A/D 转换器 0 和 1 的框图



注意事项 1. 如果在模拟输入引脚 (ANIn0 至 ANIn3) 或 A/D 转换器参考电压输入引脚 (AVREFn) 处有噪声, 那么噪声可能会产生不正确的转换结果 (n = 0, 1)。

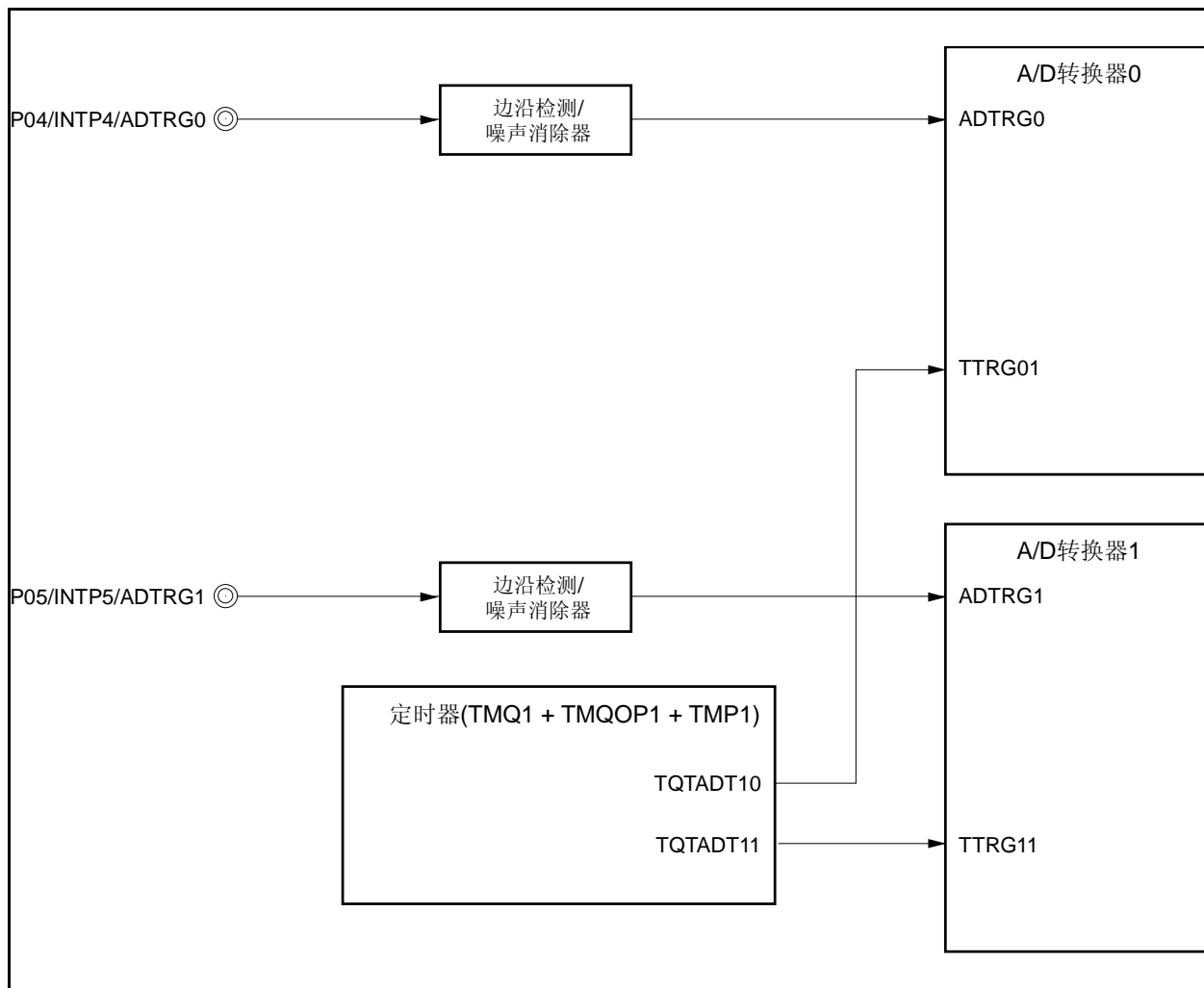
因此, 需要通过软件处理来避免该不正确的转换结果对系统的负效应。

软件处理实例如下所示:

- 取若干 A/D 转换结果的平均值并将其作为 A/D 转换结果。
- 连续执行若干 A/D 转换并使用转换结果, 同时忽略可能得到的异常结果。
- 如果从获得 A/D 转换结果能够判断已经产生了系统故障, 那么在执行故障处理前请务必复查系统故障。

2. 不要将超出 AVSSn 至 AVREFn 范围的电压应用于用作 A/D 转换器 0 和 1 的输入引脚的引脚。

图 11-2. 定时器触发模式中触发源选择器的框图



A/D 转换器 0 和 1 由以下硬件组成：

表 11-1. A/D 转换器 0 和 1 的配置

名称	配置
模拟输入	ANI00 至 ANI03, ANI10 至 ANI13 (两个电路, 总共八个通道)
寄存器	逐次逼近寄存器 (SAR) A/Dn 转换结果寄存器 0 至 3 (ADAnCR0 至 ADAnCR3) A/Dn 转换结果寄存器 0H 至 3H (ADAnCR0H 至 ADAnCR3H)
控制寄存器	A/D 转换器 n 模式寄存器 0 (ADAnM0) A/D 转换器 n 模式寄存器 1 (ADAnM1) A/D 转换器 n 模式寄存器 2 (ADAnM2) A/D 转换器 n 通道指定寄存器 0 (ADAnS)

备注 n = 0,1

(1) 选择器

输入电路根据 ADAnM0, ADAnM1, ADAnM2, 以及 ADAnS 寄存器设置的模式来选择模拟输入引脚 (ANIn0 至 ANIn3), 并且将输入传送到采样&保持电路 (n = 0、1)。

(2) 采样保持电路

采样保持电路采集每个从输入电路连续发出的模拟输入电压, 并发送采样数据到电压比较器。 A/D 转换期间, 该电路保持所采样模拟输入的电压。

(3) 电压比较器

该比较器用于将自数组抽头电压中产生的电压与模拟输入电压进行比较。 如果发现比较的结果是模拟输入电压大于参考电压 ($1/2AV_{REFn}$), 则将会设置逐次逼近寄存器 (SAR) 的最高有效位 (MSB)。 如果模拟输入电压小于参考电压 ($1/2AV_{REFn}$), 那么逐次逼近寄存器 (SAR) 的最高有效位 (MSB) 复位。接着, 自动设置 SAR 寄存器的位 8, 并启动下一次比较。 由位 9 的值来选择数组的抽头电压, 其结果 (位 9 的值) 已经设定。

位 9 = 0: ($1/4AV_{REFn}$)
位 9 = 1: ($3/4AV_{REFn}$)

将数组的抽头电压和模拟输入电压进行比较, 且根据比较结果生成 SAR 的位 8。

模拟输入电压 \geq 数组的抽头电压: 位 8 = 1
模拟输入电压 \leq 数组的抽头电压: 位 8 = 0

这样的比较会持续至 SAR 的位 0。

(4) 数组

数组通过自模拟输入引脚 (ANIn0 至 ANIn3) (n = 0, 1) 的电压输入来产生比较电压。

(5) 逐次逼近寄存器 (SAR)

SAR 为 10 位寄存器，用于设置抽头电压数据，自数组的抽头电压值与模拟输入引脚的电压值作比较，从最高有效位 (MSB) 开始，每次移 1 位。

如果数据设置至 SAR 中的最低有效位 (LSB) (A/D 转换结束)，那么 SAR (转换结果) 将会保存到 A/Dn 转换结果寄存器 0 至 3 中 (ADAnCR0 至 ADAnCR3) ($n = 0, 1$)。当所有指定的 A/D 转换操作都结束时，将产生 A/Dn 转换结束中断请求信号 (INTADn)。

(6) A/Dn 转换结果寄存器 0 至 3 (ADAnCR0 至 ADAnCR3)、A/Dn 转换结果寄存器 0H 至 3H (ADAnCR0H 至 ADAnCR3H) ($n = 0, 1$)

ADAnCR0 至 ADAnCR3 以及 ADAnCR0H 至 ADAnCR3H 寄存器是用于保存 A/D 转换结果的寄存器。每当 A/D 转换结束时，将会从逐次逼近寄存器 (SAR) 中装载转换结果并存储到 ADAnCR0 至 ADAnCR3 寄存器的高 10 位中。读取时，这些寄存器中的低 6 位始终为 0。

A/D 转换结果的高 8 位从 ADAnCR0H 至 ADAnCR3H 寄存器中读取。要以 16 位为单位读取 A/D 转换结果，应读取 ADAnCR0 至 ADAnCR3 寄存器。要读取高 8 位，则应指读取 ADAnCR0H 至 ADAnCR3H 寄存器。

(7) A/D 转换器 n 模式寄存器 0 (ADAnM0) ($n = 0, 1$)

该寄存器用于指定操作模式并控制转换操作。

(8) A/D 转换器 n 模式寄存器 1 (ADAnM1) ($n = 0, 1$)

该寄存器用于设置将要进行 A/D 转换的模拟输入的转换时钟数目。

(9) A/D 转换器 n 通道指定寄存器 (ADAnS) ($n = 0, 1$)

该寄存器用于指定进行 A/D 转换的模拟输入引脚。

(10) A/D 转换器 n 模式寄存器 2 (ADA2M2) ($n = 0, 1$)

该寄存器用于指定缓冲模式并在硬件触发模式中指定模式。

(11) ANIn0 至 ANIn3 引脚 ($n = 0, 1$)

ANIn0 至 ANIn3 是 A/D 转换器 0 和 1 的模拟输入引脚，用于输入要进行 A/D 转换的模拟信号。

注意事项 确保输入 ANIn0 至 ANIn3 的电压不超过额定值。如果大于等于 AV_{REFn} 或小于等于 AV_{SSn} (即使在最大绝对额定值范围内) 的电压被输入到通道中，那么通道的转换值将不确定，而其它通道的转换值也会受到影响。

(12) AV_{REF0} 和 AV_{REF1} 引脚

该引脚用于输入 A/D 转换器 0 和 1 的参考电压，其根据施加在 AV_{REFn} 和 AV_{SSn} ($n = 0, 1$) 之间的电压将 ANIn0 至 ANIn3 引脚的输入信号转换为数字信号。即使没有使用 A/D 转换器 0 和 1，也应保持 AV_{REFn} 引脚电位与 EV_{DD} 引脚电位一致。

AV_{REFn} 引脚的工作电压范围为 $V_{DD} = EV_{DD} = AV_{DDn} = AV_{REFn} = 4.5$ 至 5.5 V。

(13) AVSS0 和 AVSS1 引脚

这些引脚是 A/D 转换器 0 和 1 的接地引脚。即使没有使用 A/D 转换器 0 和 1，也应始终保持 AVSSn 引脚电位与 EVSS 引脚电位一致。

(14) AVDD0 和 AVDD1 引脚

这些引脚是 A/D 转换器 0 和 1 的模拟电源引脚。

为 AVDD0 和 AVDD1 引脚提供相同的电势。

即使没有使用 A/D 转换器 0 和 1，也应保持该 AVDDn 引脚电位与 EVDD 引脚电位一致。

AVDDn 引脚的操作电压范围为 $V_{DD} = E_{VDD} = A_{VREFn} = A_{VDDn} = 4.5$ 至 5.5 V。

(15) 控制器

该电路用于执行诸如允许/禁止 A/D 转换器 0 和 1 和选择操作模式以及触发模式之类的控制操作。

11.3 控制寄存器

A/D 转换器 0 和 1 由下列寄存器来控制：

- A/D 转换器 n 模式寄存器 0 至 2 (ADAnM0 至 ADAnM2)
- A/D 转换器 n 通道指定寄存器 (ADAnS)

也使用下面的寄存器：

- A/Dn 转换结果寄存器 0 至 3 (ADAnCR0 至 ADAnCR3)
- A/Dn 转换结果寄存器 0H 至 3H (ADAnCR0H 至 ADAnCR3H)

(1) A/D 转换器 n 模式寄存器 0 (ADAnM0)

ADAnM0 寄存器是 8 位寄存器，用于指定操作模式并控制转换操作。

该寄存器可按字节或位进行读写。但是，位 0 是只读位。忽略对位 0 执行的写操作。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: ADA0M0 FFFFF200H, ADA1M0 FFFFF220H

	<7>	6	5	4	3	2	1	0
ADAnM0 (n = 0, 1)	ADAnCE	0	ADAnMD1	ADAnMD0	ADAnETS1	ADAnETS0	ADAnTMD	ADAnEF
	ADAnCE	A/D 转换操作的控制						
	0	停止转换操作						
	1	开始转换操作						
	ADAnMD1	ADAnMD0	操作模式说明					
	0	0	连续选择模式					
	0	1	连续扫描模式					
	1	0	单脉冲选择模式					
	1	1	单脉冲扫描模式					
	ADAnETS1	ADAnETS0	外部触发(ADTRGn)有效沿的说明					
	0	0	没有边沿检测(外部触发无效)					
	0	1	下降沿					
	1	0	上升沿					
	1	1	双边沿					
	ADAnTMD	定时器模式说明						
	0	软件触发模式						
	1	硬件触发模式 ^注						
	ADAnEF	A/D转换器 n的状态						
	0	A/D 转换停止						
	1	A/D 转换进行中						

注 选择硬件触发模式时，使用 ADAnM2.ADAnTMD1 位来选择触发模式。

- 注意事项 1. 在软件触发模式中，当向 ADAnCE 位写入 1 时触发转换操作。
 在硬件触发模式（外部触发模式或定时器触发模式）中，当向 ADAnCE 位写入 1 时触发信号将待命。
 在所有模式中，即使都产生 A/Dn 转换结束中断请求信号（INTADn）后，ADAnCE 位也不会清零。若要停止 A/D 转换操作，应将 0 写入 ADAnCE 位
2. 如果在 A/D 转换（ADAnEF 位 = 1）期间写入 ADAnM0，ADAnM2，以及 ADAnS 寄存器，那么在各个模式中执行如下操作：
- 软件触发模式中：
A/D 转换停止并再次从头开始执行。
 - 硬件触发模式中：
A/D 转换停止且再次恢复到触发等待状态。

(2) A/D 转换模式寄存器 1 (ADAnM1)

ADAnM1 寄存器是用于指定转换时钟数目的 8 位寄存器。

转换时钟数目包括采样时钟数目。

该寄存器可按字节或位进行读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: ADA0M1 FFFFF201H, ADA1M1 FFFFF221H

	7	6	5	4	3	2	1	0
ADAnM1 (n = 0, 1)	0	0	0	0	0	0	ADAnFR1	ADAnFR0

注意事项

1. 参见表 11-2 ADAnFR1 位和 ADAnFR0 位的转换时钟数目。
2. ADAnM0.ADAnCE 位 = 1 (转换允许) 时, 禁止改变 ADAnFR1 位和 ADAnFR0 位。
3. 一定要将位 2 至 7 清为“0”。

表 11-2. 转换时钟的数目

ADAnFR1	ADAnFR0	转换时钟的数目 (f _{xx} = 20 MHz)	稳定时钟的数目 (f _{xx} = 20 MHz)	触发响应时钟的数目 (f _{xx} = 20 MHz)
0	0	设置禁止	-	-
0	1	62 (3.10 μs)	33	6
1	0	93 (4.65 μs)	50	7
1	1	124 (6.20 μs)	54	8

• A/D 转换时钟的总数

触发模式:		操作模式	ADAnCE 位设为 1 后由第一次触发生成的 A/D 转换时钟的总数	ADAnCE 位设为 1 后由第二次或之后的触发生成的 A/D 转换时钟总数	
软件触发	连续选择	1 缓冲区	稳定时钟的数目 + 触发响应时钟的数目 + 转换时钟的数目	转换时钟的数目	
		4 缓冲区			
	连续扫描	1 缓冲区		- (一次转换后结束转换)	
	单脉冲选择	1 缓冲区			转换时钟的数目
		4 缓冲区			
单脉冲扫描	1 缓冲区	转换时钟的数目 ^{※1}			
硬件触发	定时器触发	连续选择/ 单脉冲选择	触发响应时钟的数目 + 转换时钟的数目 ^{※2}	- (一次转换后结束转换)	
		1 缓冲区		转换时钟的数目	
		4 缓冲区			
	外部触发	连续扫描/ 单脉冲扫描	噪声消除时间 + 触发响应时钟的数目 + 转换时钟的数目 ^{※2}	转换时钟的数目 ^{※1}	
		连续选择/ 单脉冲选择		1 缓冲区	- (一次转换后结束转换)
				4 缓冲区	
连续扫描/ 单脉冲扫描	1 缓冲区	转换时钟的数目 ^{※1}			

- 注**
1. 扫描两个或更多通道时 (ADAnS 寄存器 ≠ 00H)
 2. ADAnCE 位从 0 设为 1 后, 等待稳定时钟数目的时间。如果在这段时间内输入触发, 则稳定时间过后响应触发。结果, A/D 转换时钟的最大数目会大于“上述稳定时间加上稳定时钟的数目”。

(3) A/D 转换器 n 通道指定寄存器 (ADAnS)

ADAnS 是用于指定模拟输入引脚的 8 位寄存器。

该寄存器可按字节或位进行读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: ADA0S FFFFF202H, ADA1S FFFFF222H

	7	6	5	4	3	2	1	0
ADAnS (n = 0, 1)	0	0	0	0	0	0	ADAnS1	ADAnS0

ADAnS1	ADAnS0	选择模式	扫描模式
0	0	ANIn0	ANIn0
0	1	ANIn1	ANIn0, ANIn1
1	0	ANIn2	ANIn0 至 ANIn2
1	1	ANIn3	ANIn0 至 ANIn3

注意事项 1. 如果在 A/D 转换 (ADAnM0.ADAnEF = 1) 期间写入 ADAnS 寄存器, 那么在各个模式中进行如下操作:

- 软件触发模式中:
A/D 转换停止并再次从头开始执行。
- 硬件触发模式中:
A/D 转换停止且再次恢复到触发等待状态。

2. 一定要将位 2 至 7 清为“0”。

(4) 转换器 n 模式寄存器 2 (ADAnM2)

ADAnM2 寄存器是 8 位寄存器，用于指定缓冲模式以及硬件触发模式。

该寄存器可按字节或位进行读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: ADA0M2 FFFFF203H, ADA1M2 FFFFF223H

	7	6	5	4	3	2	1	0
ADAnM2 (n = 0, 1)	ADAnBS	0	0	0	0	0	ADAnTMD1	0

ADAnBS	缓冲模式说明
0	1缓冲区模式
1	4缓冲区模式 ^注

ADAnTMD1	硬件触发模式说明
0	外部触发模式
1	定时器触发模式

注 在 4 缓冲区模式中，不考虑所选的模拟输入引脚，将会按 ADAnCR0 → ADAnCR1 → ADAnCR2 → ADAnCR3 的顺序存储 A/D 转换结果。

在连续扫描模式和单脉冲扫描模式中禁止 4 缓冲区模式的设置。

- 注意事项**
1. A/D 转换器 0 和 1 的外部触发分别从 P04/INTP4/ADTRG0 引脚和 P05/INTP5/ADTRG1 引脚输入。因此，若要使用外部触发模式，一定要将 PMC0.PMC04 位和 PMC0.PMC05 位设为 1。
 2. A/D 转换器的定时器触发是定时器（马达控制功能）的 A/D 转换开始触发信号（TQTADT1n）。将 TQTADT1n 信号连接至 A/D 转换器 n 的 TTRGn1 信号（参见图 11-2）。

- A/D 转换器 0 的定时器触发
定时器触发模式中：TQTADT10
- A/D 转换器 1 的定时器触发
定时器触发模式中：TQTADT11

TQTADT1n 信号通过使用 TMQ1 选项寄存器 2 (TQ1OPT2) 的 TQ1AT00 至 TQ1AT03 位以及 TMQ1 选项寄存器 3 (TQ1OPT3) 的 TQ1AT10 至 TQ1AT13 位来设置。可选为 A/D 转换开始触发（定时器触发）的马达控制功能，其触发源是 INTTP1CC0, INTTP1CC1, INTTQ1CC0, 以及 INTTQ1OV 信号（可以选择两个或更多的信号）。

3. A/D 转换 (ADAnM0.ADAnEF = 1) 期间，如果写入 ADAnM2 寄存器，那么在各个模式中将进行如下操作：
 - 软件触发模式中：
A/D 转换停止并再次从头开始执行。
 - 硬件触发模式中：
A/D 转换停止且再次恢复到触发等待状态。

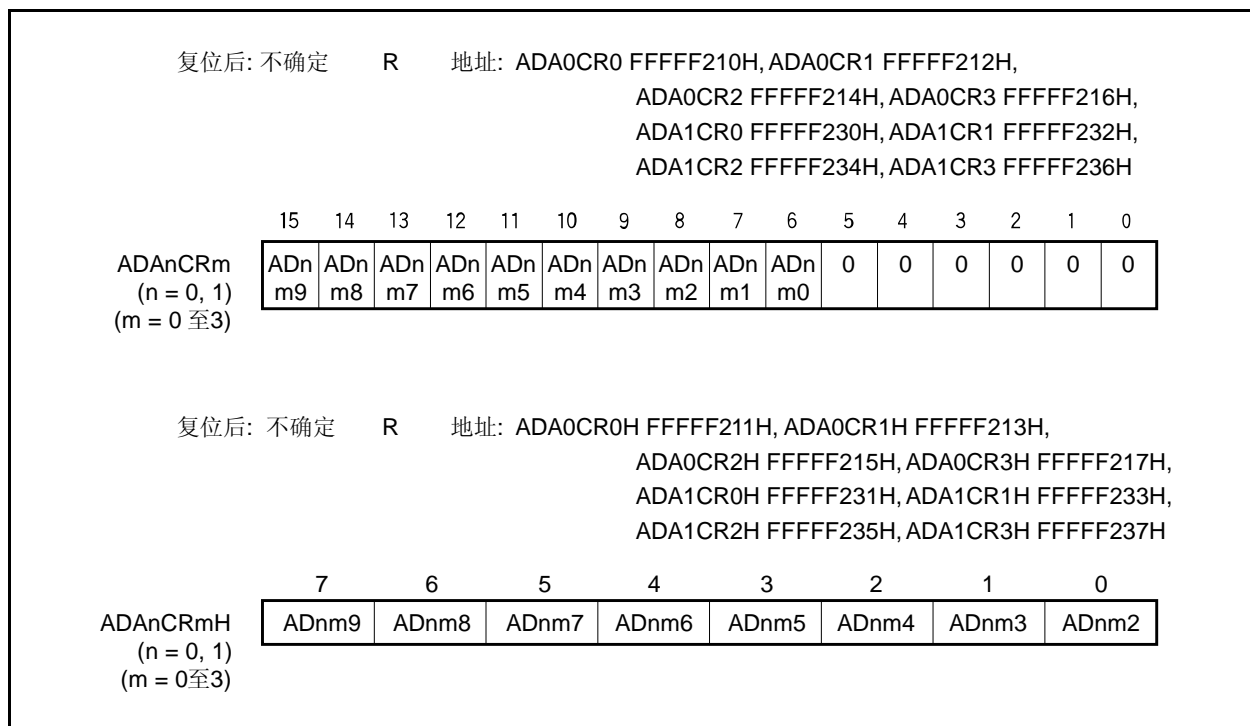
(5) A/Dn 转换结果寄存器 0 至 3, 0H 至 3H (ADAnCR0 至 ADAnCR3, ADAnCR0H 至 ADAnCR3H)

ADAnCRm 和 ADAnCRH 是用于保存 A/D 转换结果的寄存器。每个电路都提供了这四种寄存器，且有两个电路可用。每当 A/D 转换结束时，将从逐次逼近寄存器（SAR）中加载转换结果并保存到 ADAnCRm 寄存器的高 10 位。读取时，这些寄存器中的低 6 位始终为 0。

A/D 转换结果的高 8 位读入 ADAnCRmH 寄存器。

这些寄存器是以字节或字为单位的只读寄存器。当以 16 位为单位读取 A/D 转换结果时，指定 ADAnCRm 寄存器，而当读取高 8 位时，则指定 ADAnCRmH 寄存器。

复位后，这些寄存器的值不确定。



模拟输入引脚与 ADAnCRm 寄存器和 ADAnCRmH 寄存器的对应关系如下所示：

表 11-3. 模拟输入引脚与 ADAnCRm 和 ADAnCRmH 寄存器的对应关系

A/D 转换器	模拟输入引脚:	A/D 转换结果寄存器
A/D 转换器 0	ANI00	ADA0CR0、ADA0CR0H
	ANI01	ADA0CR1、ADA0CR1H
	ANI02	ADA0CR2、ADA0CR2H
	ANI03	ADA0CR3、ADA0CR3H
A/D 转换器 1	ANI10	ADA1CR0、ADA1CR0H
	ANI11	ADA1CR1、ADA1CR1H
	ANI12	ADA1CR2、ADA1CR2H
	ANI13	ADA1CR3、ADA1CR3H

输入模拟输入引脚 (ANInm) 的模拟电压与 A/D 转换结果 (A/Dn 转换结果寄存器 m (ADAnCRm) 的结果) 的关系如下:

$$SAR = INT \left(\frac{V_{IN}}{AV_{REF}} \times 1,024 + 0.5 \right)$$

$$ADCR^* = SAR \times 64$$

或,

$$(SAR - 0.5) \times \frac{AV_{REF}}{1,024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{REF}}{1,024}$$

INT(): 函数, 返回 () 中的整数值

V_{IN}: 模拟输入电压

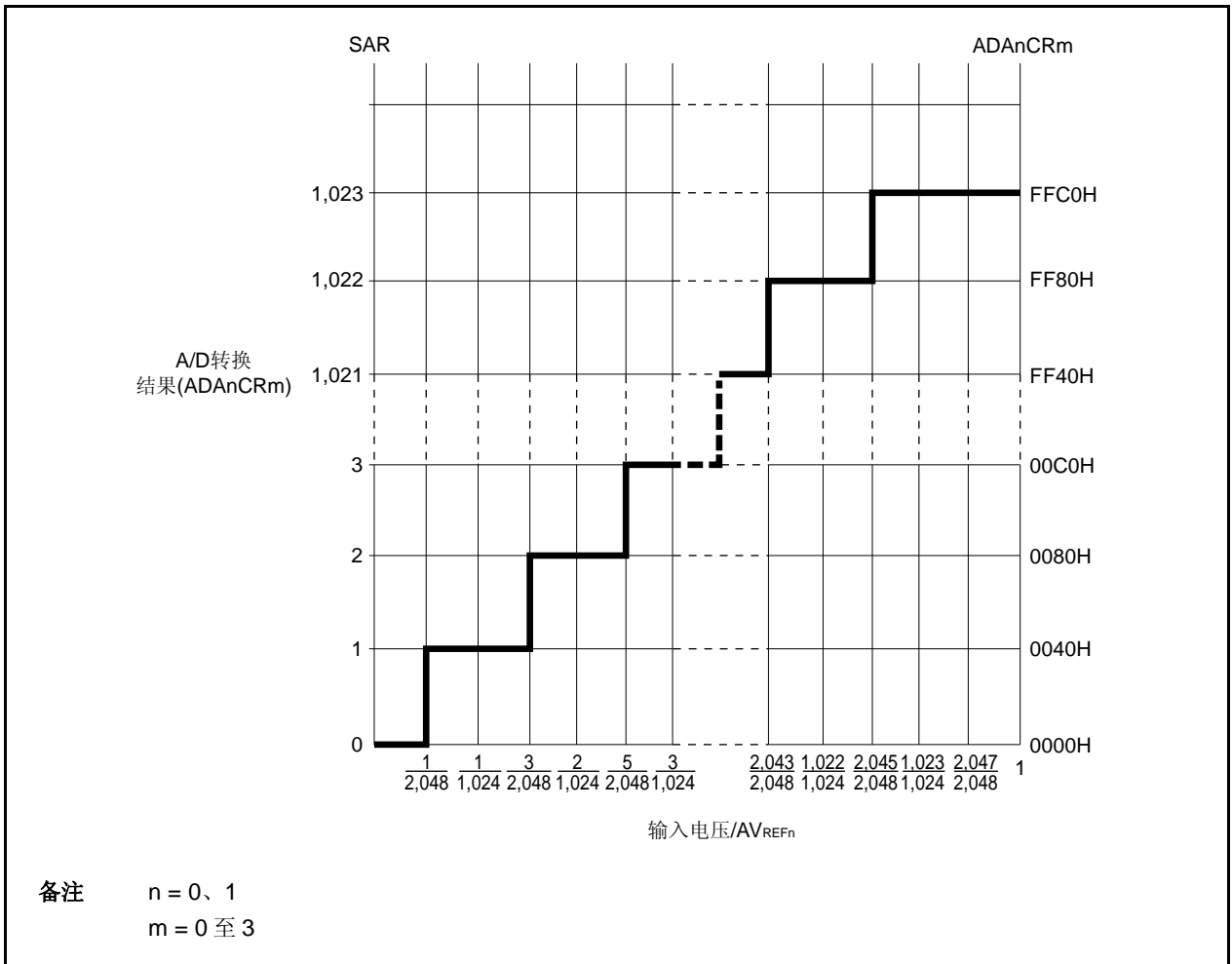
AV_{REF}: AV_{REFn} 引脚电压

ADCR: A/Dn 转换结果寄存器 m (ADAnCRm) 的值

注 ADAnCRm 寄存器的低 6 位固定为 0。

模拟输入电压与 A/D 转换结果之间的关系如下所示。

图 11-3. 模拟输入电压和 A/D 转换结果的关系



11.4 操作

注意事项 A/D 转换器 0 和 1 可以同时采样两个电路。

11.4.1 基本操作

按以下步骤执行 A/D 转换：

- (1) 使用 ADAnM0, ADAnM1, ADAnM2, 以及 ADAnS 寄存器^注 ($n = 0, 1$) 来选择模拟输入引脚, 操作模式以及触发模式。A/D 转换使能后, 应立即由 ADAnM1.ADAnFR0 和 ADAnM1.ADAnFR1 位的规范确定稳定时钟个数的设置。

<R> **注** 在 A/D 转换期间, 写入 ADAnM0, ADAnM2, 以及 ADAnS 寄存器时, 如果输入无效触发信号, 则转换结果将不会正确存储到 ADAnCRm 寄存器 ($m = 0$ 至 3) 中, 且转换操作会在发生变化前被初始化并重新开始执行。

- (2) 在软件触发模式中, 稳定时钟过后 ($n = 0, 1$), 将 ADAnM0.ADAnCE 位设为 1 以启动 A/D 转换。在硬件触发模式 (外部触发模式, 定时器触发模式) 中, 如果将 ADAnCE 位设为 1, 则 A/D 转换器将会进入触发待命状态。详情请参见 11.3 (2) A/D 转换器 n 模式寄存器 1 (ADAnM1)。
- (3) 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。
- (4) 在特定时间执行采样时, 采样&保持电路将会进入保持状态, 并会保持输入模拟电压直到 A/D 转换结束。
- (5) 设置逐次逼近寄存器 (SAR) 的位 9。抽头选择器将数组抽头电压的电平改为参考电压 ($1/2AV_{REFn}$)。
- (6) 通过比较器将数组抽头电压所产生的电压与模拟输入电压进行比较。如果发现比较的结果是模拟输入电压大于参考电压 ($1/2AV_{REFn}$), 那么逐次逼近寄存器 (SAR) 的最高有效位 (MSB) 保持置位。如果模拟输入电压小于参考电压 ($1/2AV_{REFn}$), 那么逐次逼近寄存器 (SAR) 的最高有效位 (MSB) 复位。
- (7) 接着, 自动设置逐次逼近寄存器 (SAR) 的位 8 并启动下一次比较。根据位 9 的值选择数组的抽头电压, 其结果 (位 9 的值) 已经设定。

位 9 = 0: ($1/4AV_{REFn}$)

位 9 = 1: ($3/4AV_{REFn}$)

将数组的抽头电压和模拟输入电压进行比较, 而 SAR 的位 8 则根据比较结果来执行操作。

模拟输入电压 \geq 数组的抽头电压: 位 8 = 1

模拟输入电压 \leq 数组的抽头电压: 位 8 = 0

这样的比较会持续至 SAR 的位 0。

(8) 完成 10 位比较后，有效数字值结果将会保留在逐次逼近寄存器（SAR）中。该值传输至 A/Dn 转换结果寄存器 m（ADAnCRm）中，且转换结果也存储在该寄存器中（n = 0、1，m = 0 至 3）。当指定次数的 A/D 转换操作结束时，将产生 A/Dn 转换结束中断请求信号（INTADn）。

11.4.2 操作模式和触发模式

通过指定操作模式和触发模式可以为 A/D 转换器 0 和 1 指定各种转换操作。操作模式和触发模式通过 ADAnM0，ADAnM1，ADAnM2，以及 ADAnS 寄存器来设置。

以下显示了操作模式与触发模式之间的关系：

备注 n = 0、1

触发模式		操作模式		设置值			
				ADAnM0	ADAnM1	ADAnM2	ADAnS
软件触发	连续选择	1 缓冲区	X000XX0XB	000000XXB	00000000B	00000XXXB	
		4 缓冲区	X000XX0XB	000000XXB	10000000B	00000XXXB	
	连续扫描		X001XX0XB	000000XXB	00000000B	00000XXXB	
	单脉冲选择	1 缓冲区	X010XX0XB	000000XXB	00000000B	00000XXXB	
		4 缓冲区	X010XX0XB	000000XXB	10000000B	00000XXXB	
	单脉冲扫描		X011XX0XB	000000XXB	00000000B	00000XXXB	
硬件触发	外部触发	连续选择	1 缓冲区 ^{注 1}	X000XX1XB	000000XXB	00000000B	00000XXXB
			4 缓冲区 ^{注 2}	X000XX1XB	000000XXB	10000000B	00000XXXB
		连续扫描 ^{注 3}		X001XX1XB	000000XXB	00000000B	00000XXXB
		单脉冲选择	1 缓冲区 ^{注 1}	X010XX1XB	000000XXB	00000000B	00000XXXB
			4 缓冲区 ^{注 2}	X010XX1XB	000000XXB	10000000B	00000XXXB
		单脉冲扫描 ^{注 3}		X011XX1XB	000000XXB	00000000B	00000XXXB
	定时器触发	连续选择	1 缓冲区 ^{注 1}	X000XX1XB	000000XXB	00000010B	00000XXXB
			4 缓冲区 ^{注 2}	X000XX1XB	000000XXB	10000010B	00000XXXB
		连续扫描 ^{注 3}		X001XX1XB	000000XXB	00000010B	00000XXXB
		单脉冲选择	1 缓冲区 ^{注 1}	X010XX1XB	000000XXB	00000010B	00000XXXB
			4 缓冲区 ^{注 2}	X010XX1XB	000000XXB	10000010B	00000XXXB
		单脉冲扫描 ^{注 3}		X011XX1XB	000000XXB	00000010B	00000XXXB

- 注
1. 执行相同的操作，与触发类型无关。
 2. 执行相同的操作，与触发类型无关。
 3. 执行相同的操作，与触发类型无关。

(1) 触发模式

有两种用作 A/D 转换操作启动时序的触发模式：软件触发模式和硬件触发模式。硬件触发模式有两种类型：外部触发模式和定时器触发模式。

这些触发模式由 ADAnM0 和 ADAnM2 寄存器来设置。

备注 n = 0、1

(a) 软件触发模式

对于 ANIn0 至 ANIn3 引脚，当 ADAnM0.ADAnCE 位在该模式中被设为 1 时，由 ADAnS.ADAnS1 和 ADAnS.ADAnS0 指定的模拟输入引脚用于 A/D 转换启动时序。

A/D 转换结束后，转换结果将保存在 A/Dn 转换结果寄存器 m (ADAnCRm) (m = 0 至 3) 中。同时，产生 A/Dn 转换结束中断请求信号 (INTADn)。

如果由 ADAnM0.ADAnMD1 和 ADAnM0.ADAnMD0 位设置的操作模式是连续选择模式或连续扫描模式，除非 ADAnM0.ADAnCE 位清零，否则将重复转换操作。在单脉冲选择模式或单脉冲扫描模式中，A/D 转换结束后就会停止转换操作。

转换开始时，ADAnM0.ADAnEF 设为 1 (转换正在进行)。

A/D 转换期间，如果写入 ADAnM0, ADAnM2, 和 ADAnS 寄存器，那么转换将会停止并重新开始执行。

备注 n = 0、1
 m = 0 至 3

(b) 定时器触发模式

对于 ANIn0 至 ANIn3 引脚，由 ADAnS.ADAnS1 位和 ADAnS.ADAnS0 位指定的模拟输入引脚在该模式中用于 A/D 转换。定时器（马达控制功能）用于 A/D 转换启动时序。

A/D 转换器 n 的定时器触发信号为定时器（马达控制功能）的定时器中断请求信号（TQTADT10，TQTADT11）。将 TQTADT10 信号和 TQTADT11 信号连接至 A/D 转换器 n 的 TTRG01 信号和 TTRG11 信号（参见图 11-2）。

- A/D 转换器 0 的定时器触发：TQTADT10
- A/D 转换器 1 的定时器触发：TQTADT11

通过使用 TMQ1 选项寄存器 2（TQ1OPT2）的 TQ1AT00 位至 TQ1AT03 位以及 TMQ1 选项寄存器 3（TQ1OPT3）的 TQ1AT10 位至 TQ1AT13 位来设置 TQTADT10 信号和 TQTADT11 信号。可选为定时器触发信号的马达控制功能，其中断请求信号为 INTTP1CC0，INTTP1CC1，INTTQ1CC0，和 INTTQ1OV 信号（可以选择两个或更多的信号）。

当 ADAnM2.ADAnTMD1 位设为 1 时，A/D 转换将开始于为马达控制功能而设置的定时器中断请求信号（TQTADT10 或 TQTADT11）的上升沿处。

当 ADAnM0.ADAnCE 位设为 1 时，A/D 转换器等待触发，当输入定时器中断请求信号后，开始 A/D 转换。A/D 转换结束后，转换结果将会保存在 A/Dn 转换结果寄存器 m（ADAnCRm）中，同时产生 A/Dn 转换结束中断请求信号（INTADn）。

如果由 ADAnM0.ADAnMD1 和 ADAnM0.ADAnMD0 位设置的操作模式为连续选择模式或连续扫描模式，除非 ADAnM0.ADAnCE 位清零，否则将重复转换操作，此时随后的定时器中断请求信号将会用作触发。在单脉冲选择模式或单脉冲扫描模式中，A/D 转换器将等待触发。

转换开始时，ADAnM0.ADAnEF 位设为 1（转换正在进行）。然而，在转换器等待触发时，ADAnM0.ADAnEF 位=0（转换停止）。

如果在 A/D 转换期间输入有效触发，转换将会停止并从头开始执行。如果在 A/D 转换期间写入 ADAnM0，ADAnM2，和 ADAnS 寄存器，那么转换器停止转换并等待下一次触发。

注意事项 在定时器触发模式中，应确保在间隔时间内不产生定时器中断请求信号（A/D 转换启动时序），该间隔时间短于由 ADAnM1.ADAnFR1 和 ADAnM1.ADAnFR0 位所指定的转换时钟的最小个数。如果在间隔时间内产生了中断请求信号，且间隔时间短于转换时钟的最小个数，那么最近一次触发是有效的。

备注 n = 0、1
m = 0 至 3

(c) 外部触发模式

对于 ANIn0 至 ANIn3 引脚，由 ADAnS.ADAnS1 位和 ADAnS.ADAnS0 位指定的模拟输入引脚在该模式中用于 A/D 转换。ADTRGn 引脚用于 A/D 转换启动时序。

ADTRG0 引脚与 P04/INTP4 引脚功能复用，而 ADTRG1 引脚与 P05/INTP5 引脚功能复用。要设置外部触发模式，将端口模式控制寄存器 0 (PMC0) 的 PMC04 和 PMC05 位设为 1，且将 ADAnM2.ADAnTMD1 位设为 0。

根据 ADAnM0.ADAnETS1 位和 ADAnM0.ADAnETS0 位的设置，外部触发模式中，外部输入信号的有效沿可以从上升沿，下降沿或双边沿中进行选择。

ADAnM0.ADAnCE 位设为 1 时，A/D 转换器等待触发，且当触发信号自 ADTRGn 引脚输入时开始转换。

转换结束后，转换结果将会保存在 A/Dn 转换结果寄存器 m (ADAnCRm) 中，同时产生 A/Dn 转换结束中断请求信号 (INTADn)。

如果由 ADAnM0.ADAnMD1 和 ADAnM0.ADAnMD0 位设置的操作模式为连续选择模式或连续扫描模式，除非 ADAnM0.ADAnCE 位清零，否则将重复转换操作，此时随后的 ADTRGn 信号将会用作触发。在单脉冲选择模式或单脉冲扫描模式中，A/D 转换器将等待触发。

转换开始时，ADAnM0.ADAnEF 位设为 1 (转换正在进行)。然而，在转换器等待触发时，ADAnEF 位=0 (转换停止)。

如果在 A/D 转换期间输入有效触发，转换将会停止并从头开始执行。如果在 A/D 转换期间写入 ADAnM0，ADAnM2，和 ADAnS 寄存器，那么转换器停止转换并等待下一次触发。

注意事项 在外部触发模式中，应确保在间隔时间不会产生 ADTRGn 信号 (A/D 转换开始时序)，该间隔时间短于由 ADAnM1.ADAnFR1 和 ADAnM1.ADAnFR0 位所指定的转换时钟的最小个数。如果在间隔时间内产生 ADTRGn 信号，且间隔时间短于转换时钟的最小个数，那么最近一次触发是有效的。

备注 n = 0、1
m = 0 至 3

(2) 操作模式

ANIn0 至 ANIn3 引脚可以在四种操作模式中进行设置：连续选择模式，连续扫描模式，单脉冲选择模式以及单脉冲扫描模式。连续选择模式和单脉冲选择模式由 1 缓冲区模式和 4 缓冲区模式两种子模式组成。这些模式由 ADAnM0 和 ADAnM2 寄存器来设置。

备注 n = 0、1

(a) 连续选择模式

在该模式中，由 ADAnS 寄存器指定的模拟输入引脚（ANInm）连续进行 A/D 转换。转换结果保存在与 ANInm 引脚相对应的 A/Dn 转换结果寄存器（ADAnCRm）中。在该模式中，1 缓冲区模式和 4 缓冲区模式用于存储 A/D 转换结果。

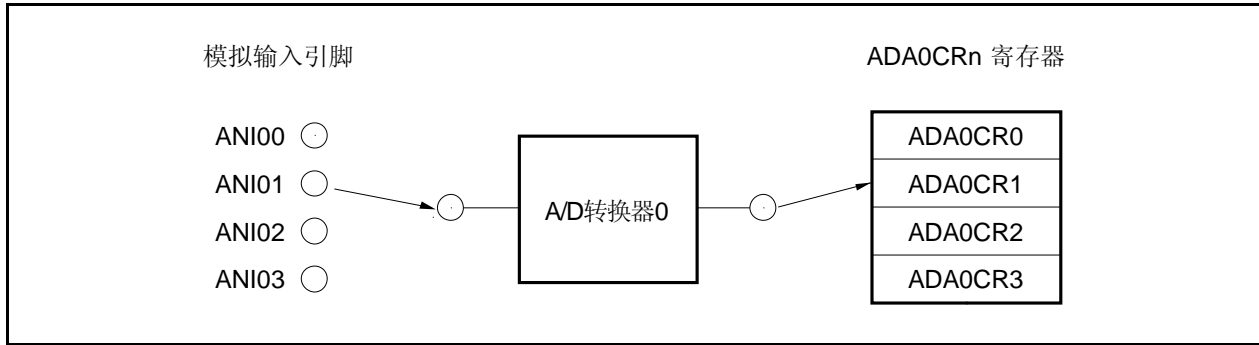
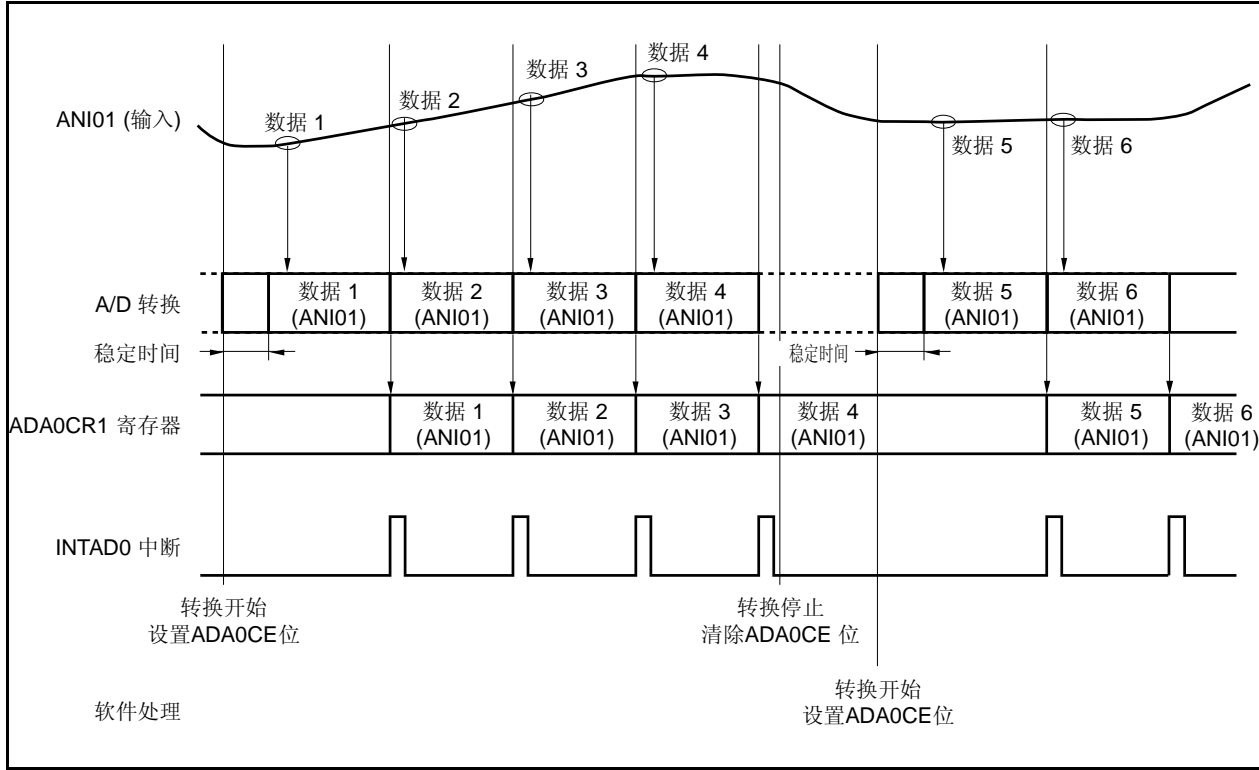
- **1 缓冲区模式**

在该模式中，由 ADAnS 寄存器指定的模拟输入引脚电压（ANInm）进行 A/D 转换。转换结果被保存在与 ANInm 引脚相对应的 ADAnCRm 寄存器中。ANInm 引脚和 ADAnCRm 寄存器一一对应，且每次 A/D 转换结束后都会产生 A/Dn 转换结束中断请求信号（INTADn）。

A/D 转换结束后，除非 ADAnM0.ADAnCE 位清零，否则转换会再次重复。

备注 n = 0、1
 m = 0 至 3

图 11-4. 连续选择 1 缓冲区模式的操作时序
 (当 ADA0M0.ADA0MD1 位和 ADA0M0.ADA0MD0 位= 00、
 ADA0M2.ADA0BS 位 = 0、ADA0S.ADA0S1 位和 ADA0S.ADA0S0 位= 01 时)

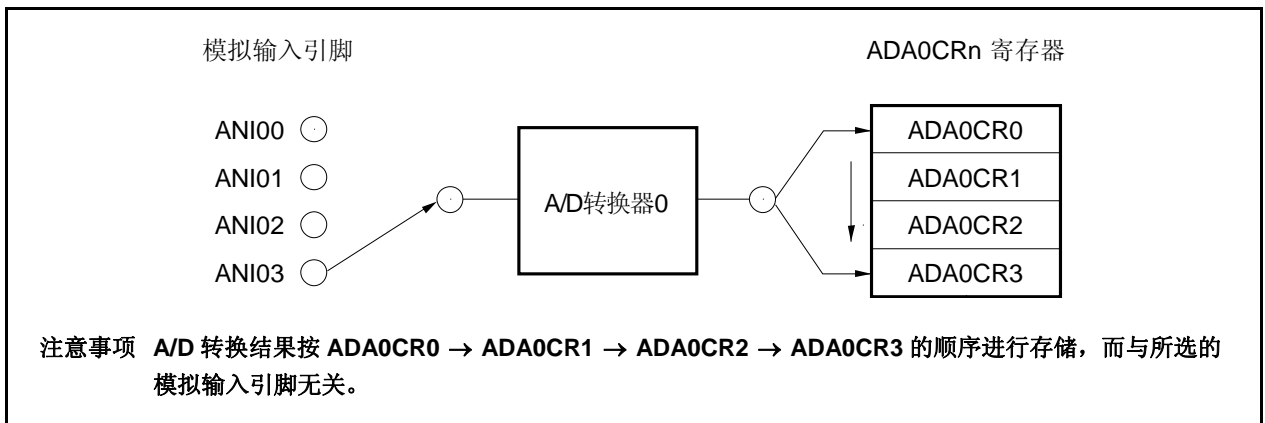
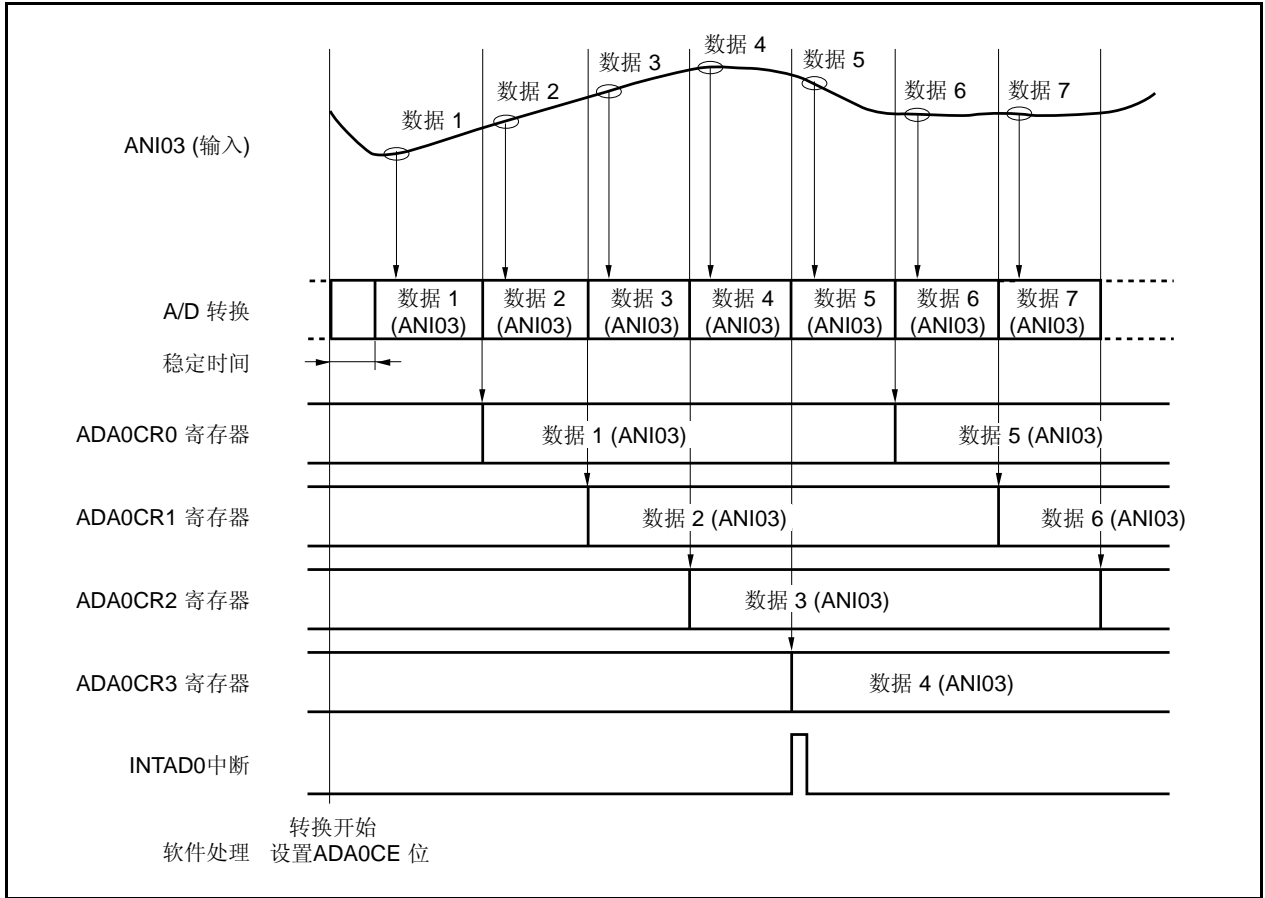


• 4 缓冲区模式

在该模式中，一个模拟输入引脚（ANInm）的电压将会进行四次 A/D 转换，其结果将会保存在 ADAnCRm 寄存器中。四次 A/D 转换结束后，将产生 A/Dn 转换结束中断请求信号（INTADn）。A/D 转换结束后，除非 ADAnM0.ADAnCE 位清零，否则将再次从头开始转换。

备注 n = 0、1, m = 0 至 3

图 11-5. 连续选择 4 缓冲区模式的操作时序
 (当 ADA0M0.ADA0MD1 位和 ADA0M0.ADA0MD0 位 = 00、
 ADA0M2.ADA0BS 位 = 1, ADA0S.ADA0S1 位和 ADA0S.ADA0S0 位 = 11)



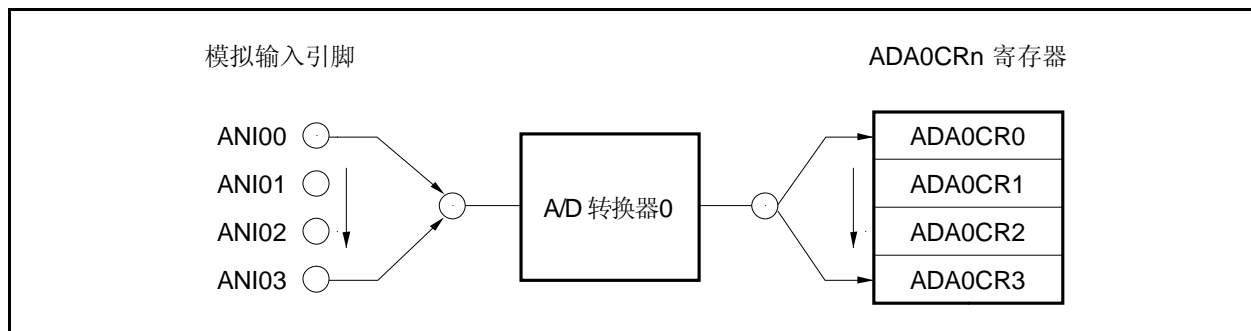
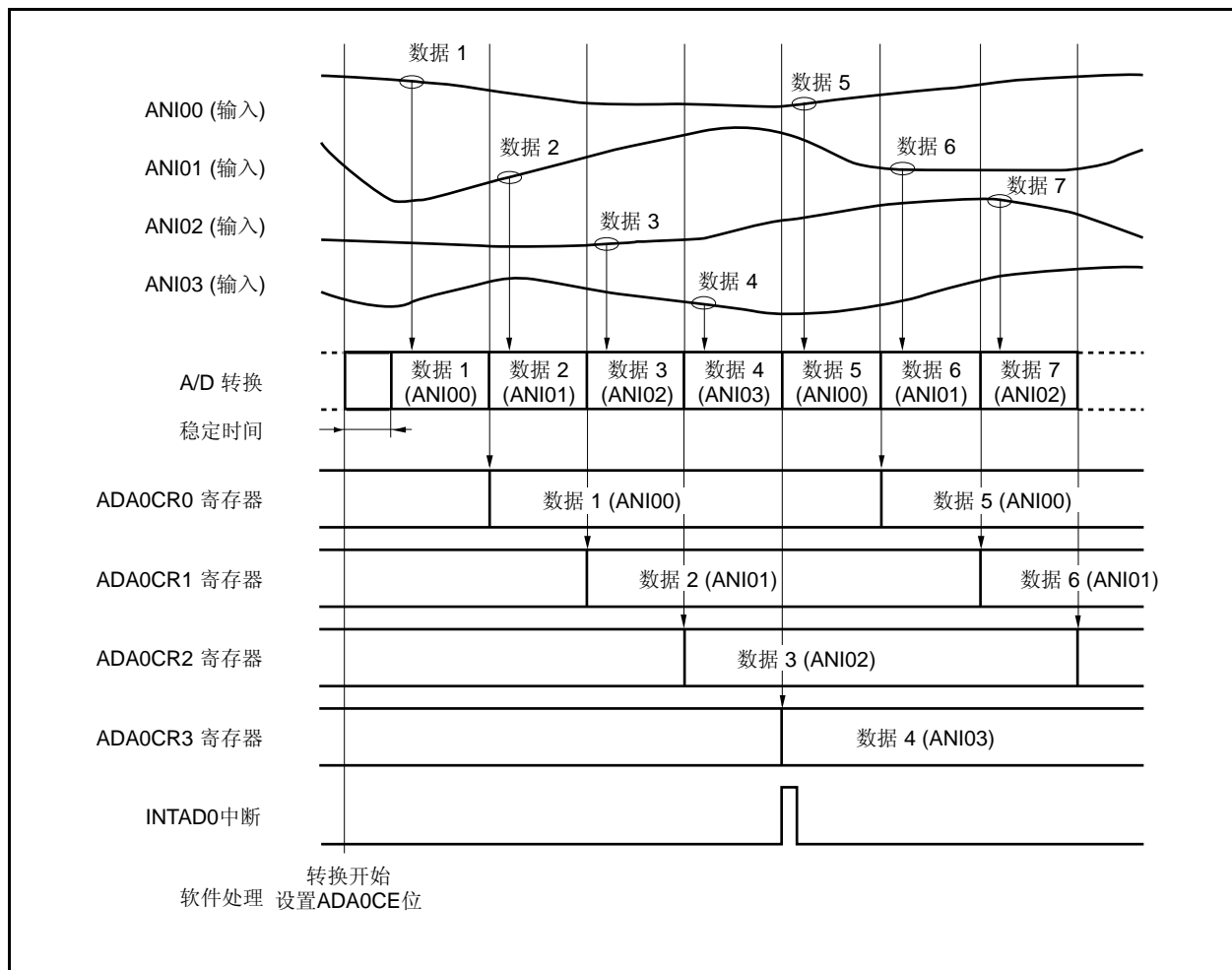
(b) 连续扫描模式

在该模式中，从 ANIn0 引脚处连续选择由 ADAnS 寄存器所指定的模拟输入引脚 (ANInm)，并且连续执行 A/D 转换。A/D 转换结果保存在与模拟输入引脚相对应的 ADAnCRm 寄存器中。当所有指定的模拟输入引脚都结束转换时，将产生 A/Dn 转换结束中断请求信号 (INTADn)。A/D 转换结束后，除非 ADAnM0.ADAnCE 位清零，否则转换会从 ANIn0 引脚再次开始。

备注 n = 0, 1, m = 0 至 3

图 11-6. 连续扫描模式的操作时序

(当 ADA0M0.ADA0MD1 位和 ADA0M0.ADA0MD0 位 = 01、ADA0S.ADA0S1 位和 ADA0S.ADA0S0 位 = 11 时)



(c) 单脉冲选择模式

在该模式中，由 ADAnS 寄存器指定的模拟输入引脚（ANInm）只会进行一次 A/D 转换。转换结果保存在与 ANInm 引脚相对应的 A/Dn 转换结果寄存器（ADAnCRm）中。在该模式中，1 缓冲区模式和 4 缓冲区模式用于存储 A/D 转换结果。

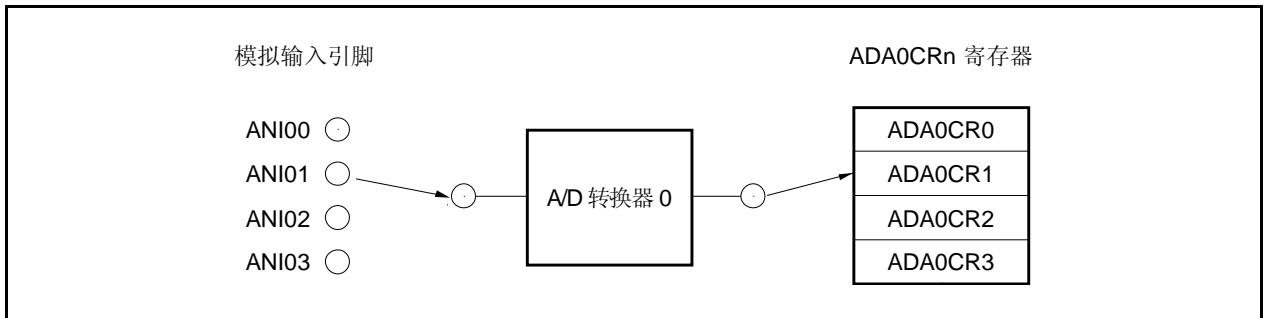
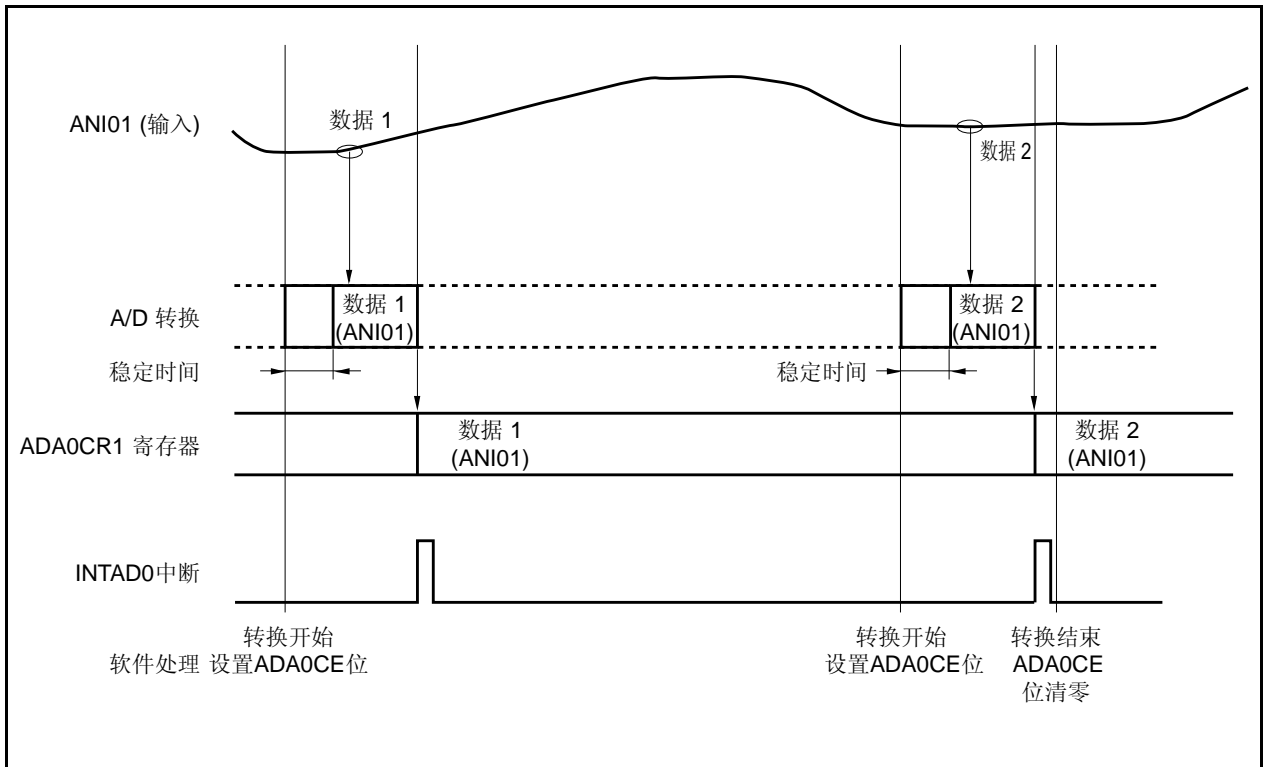
• 1 缓冲区模式

在该模式中，由 ADAnS 寄存器指定的模拟输入引脚的电压（ANInm）进行 A/D 转换。转换结果被保存在与 ANInm 引脚相对应的 ADAnCRm 寄存器中。ANInm 引脚和 ADAnCRm 寄存器一一对应，且每次 A/D 转换结束后都会产生 A/Dn 转换结束中断请求信号（INTADn）。

A/D 转换结束后，转换操作停止。

备注 n = 0、1, m = 0 至 3

图 11-7. 单脉冲选择 1 缓冲区模式的操作时序
 (当 ADA0M0.ADA0MD1 位和 ADA0M0.ADA0MD0 位 = 10、
 ADA0M2.ADA0BS 位 = 0、ADA0S.ADA0S1 位和 ADA0S.ADA0S0 位 = 01)

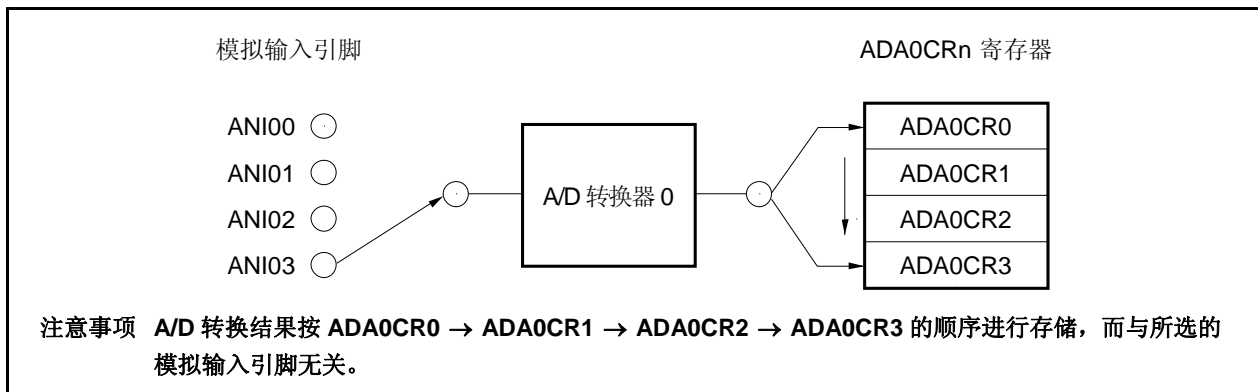
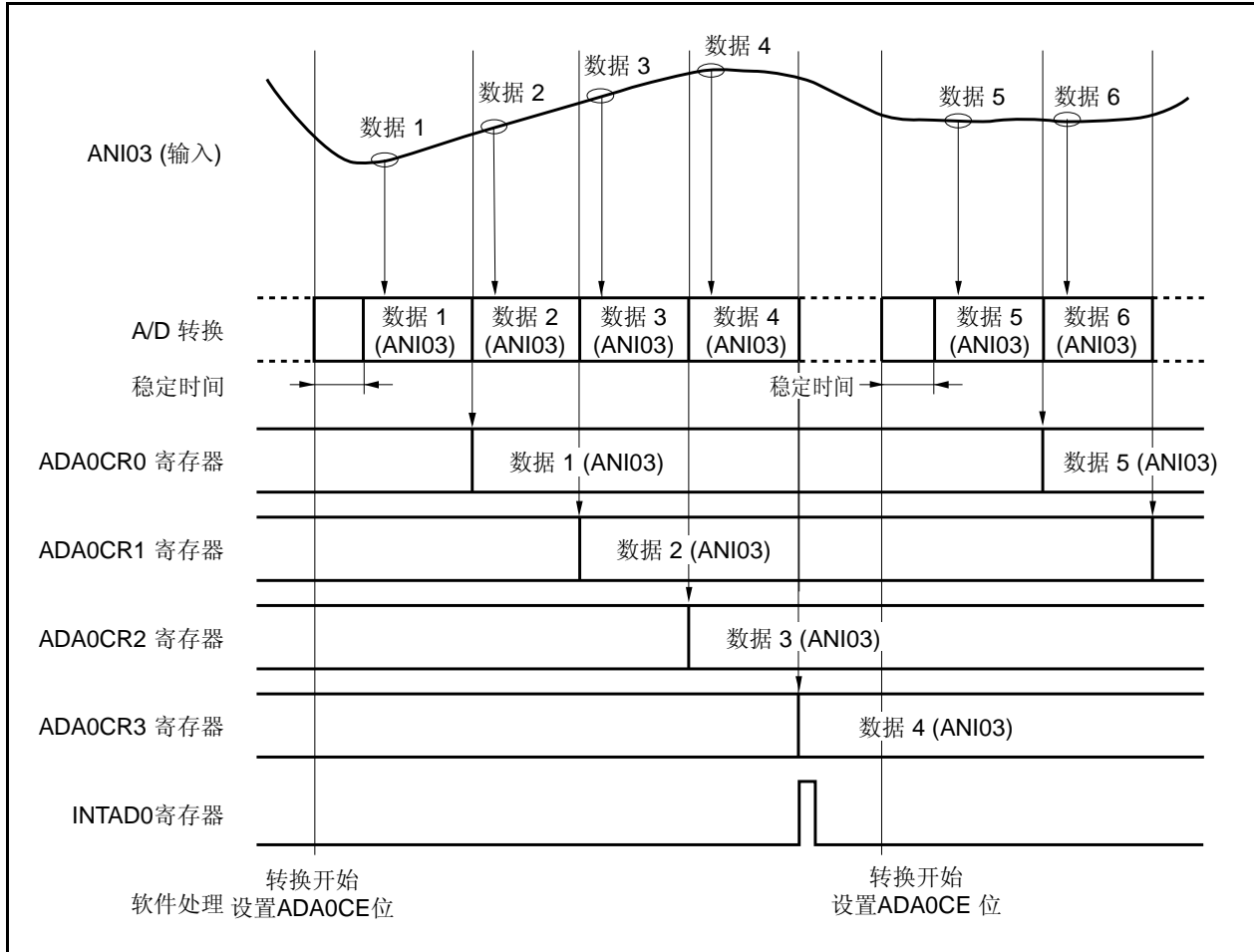


• 4 缓冲区模式

在该模式中，一个模拟输入引脚（ANInm）的电压将会进行四次 A/D 转换，其结果将会保存在 ADAnCRm 寄存器中。四次 A/D 转换结束后，将产生 A/Dn 转换结束中断请求信号（INTADn）。A/D 转换结束后，转换操作停止。

备注 n = 0、1，m = 0 至 3

图 11-8. 单脉冲选择 4 缓冲区模式的操作时序
 (当 ADA0M0.ADA0MD1 位和 ADA0M0.ADA0MD0 位 = 10、
 ADA0M2.ADA0BS 位 = 1、ADA0S.ADA0S1 位和 ADA0S.ADA0S0 位 = 11 时)



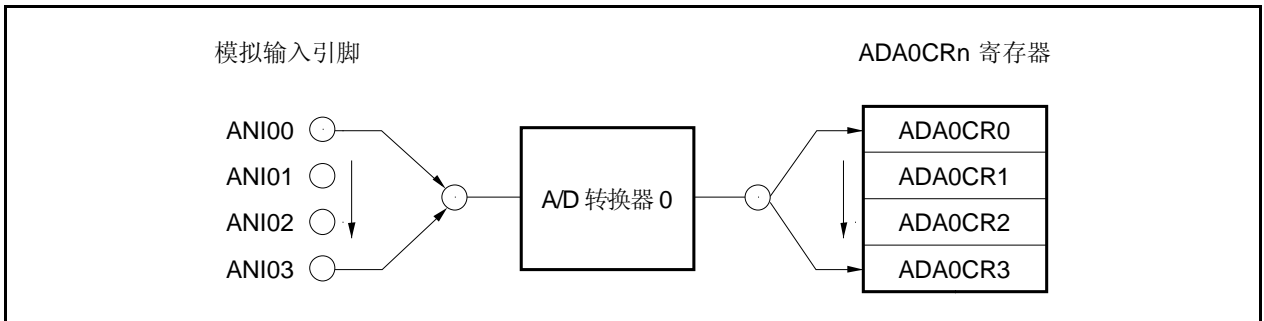
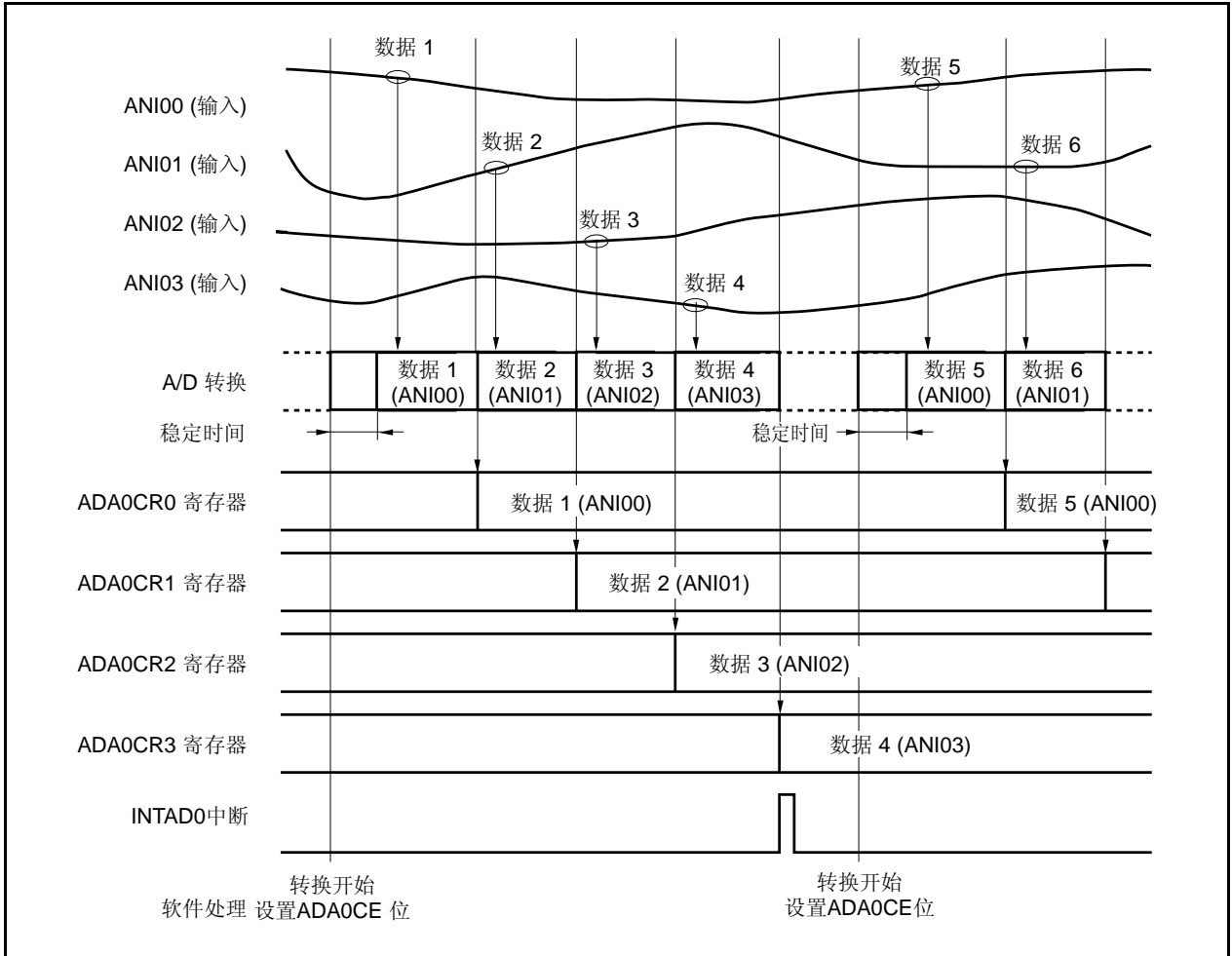
(d) 单脉冲扫描模式

在该模式中，从 ANIn0 引脚处连续选择由 ADAnS 寄存器所指定的模拟输入引脚 (ANInm)，并执行 A/D 转换。A/D 转换结果保存在与模拟输入引脚相对应的 ADAnCRm 寄存器中。当所有指定的模拟输入引脚都结束转换时，将产生 A/Dn 转换结束中断请求信号 (INTADn)。A/D 转换结束后，转换操作停止。

备注 n = 0、1
m = 0 至 3

图 11-9. 单脉冲扫描模式的操作时序

(当 ADA0M0.ADA0MD1 位和 ADA0M0.ADA0MD0 位 = 11、ADA0S.ADA0S1 位和 ADA0S.ADA0S0 位 = 11 时)



11.5 软件触发模式中的操作

ADAnM0.ADAnCE 位设为 (1) 时, 启动 A/D 转换。

A/D 转换开始后, ADAnM0.ADAnEF 位 = 1 (转换正在进行)。

A/D 转换期间, 如果写入 ADAnM0, ADAnM2, 和 ADAnS 寄存器, 那么转换将会停止并重新开始执行。

备注 n = 0、1

11.5.1 连续选择模式的操作

在该模式中, 由 ADAnS 寄存器指定的模拟输入引脚 (ANInm) 连续进行 A/D 转换。转换结果被保存在 ADAnCRm 寄存器中。在连接选择模式中, 根据存储 A/D 转换结果的方法, 可支持 1 缓冲区模式和 4 缓冲区模式。

备注 n = 0、1
 m = 0 至 3

(1) 1 缓冲区模式（软件触发连续选择：1 缓冲区）

在该模式中，模拟输入引脚（ANInm）的电压将执行一次 A/D 转换。转换结果保存在 ADAnCRm 寄存器中。ANInm 引脚与 ADAnCRm 寄存器一一对应。

每次执行 A/D 转换时，都会产生 A/Dn 转换结束中断请求信号（INTADn），并结束 A/D 转换。A/D 转换结束后，除非 ADAnM0.ADAnCE 位清零，否则转换会再次重复。

重新启动 A/D 转换^注时，无需至 ADAnM0.ADAnCE 位置（1）。

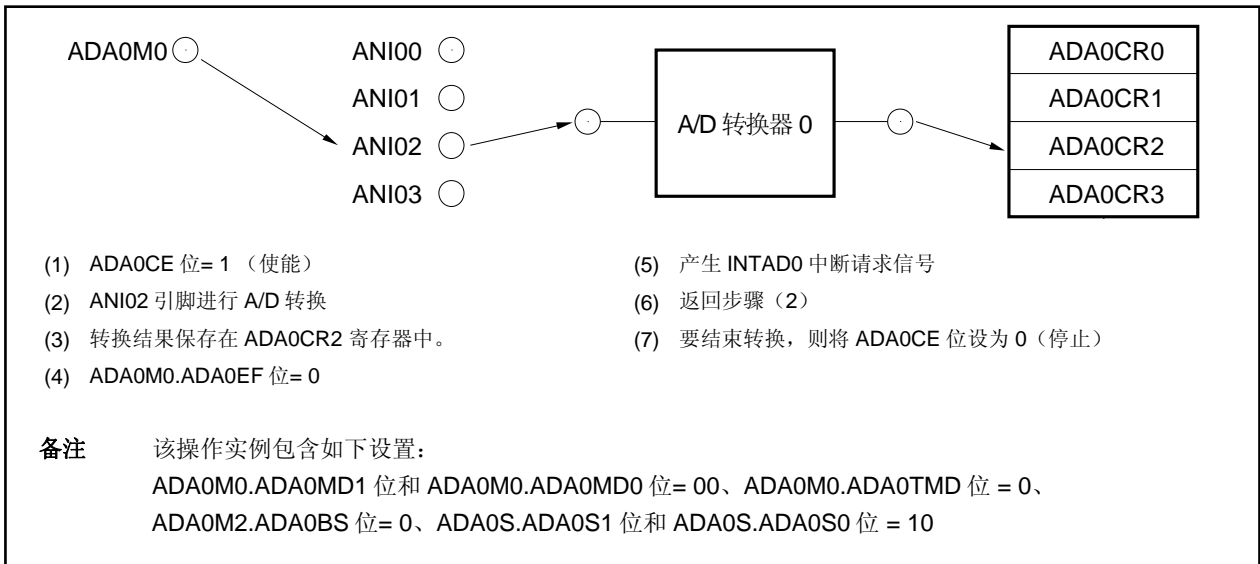
注 在软件触发连续选择的 1 缓冲区模式中，除非 ADAnM0.ADAnCE 清零，否则 A/D 转换操作不会停止。如果在下一次 A/D 转换结束前没有读取 ADAnCRm 寄存器，那么其内容会被重写。

该模式适用于读取单个模拟输入引脚的 A/D 转换值。

模拟输入引脚	A/D 转换结果寄存器
ANInm	ADAnCRm

备注 n = 0、1
m = 0 至 3

图 11-10. 1 缓冲区模式操作举例（软件触发连续选择：1 缓冲区）



(2) 4 缓冲区模式（软件触发连续选择：4 缓冲区）

在该模式中，一个模拟输入引脚（ANInm）的电压将会进行四次 A/D 转换，其结果将会保存在 ADAnCRm 寄存器中。

第 4 次 A/D 转换操作结束时，将产生 A/Dn 转换结束中断请求信号（INTADn）。A/D 转换结束后，除非将 ADAnM0.ADAnCE 位清零，否则转换会重新开始。要重新启动 A/D 转换^注，无需置位 ADAnM0.ADAnCE（1）。

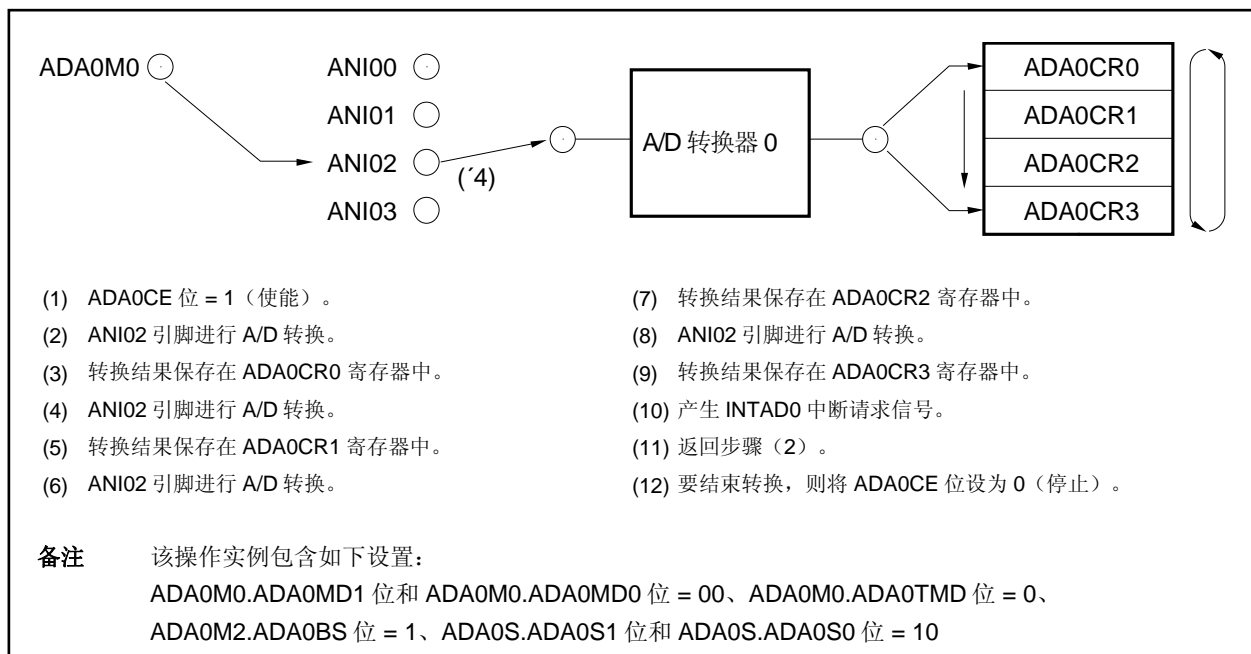
注 在软件触发连续选择的 4 缓冲区模式中，除非 ADAnM0.ADAnCE 清零，否则 A/D 转换操作不会停止。如果在下一次 A/D 转换结束前没有读取 ADAnCRm 寄存器，那么其内容会被重写。

该模式适用于计算模拟输入引脚的 A/D 转换结果平均值的应用操作。

模拟输入引脚	A/D 转换结果寄存器
ANInm	ADAnCR0
ANInm	ADAnCR1
ANInm	ADAnCR2
ANInm	ADAnCR3

备注 n = 0、1
m = 0 至 3

图 11-11. 4 缓冲区模式操作举例（软件触发连续选择：4 缓冲区）



11.5.2 连续扫描模式的操作

在该模式中，从 ANIn0 引脚处连续选择由 ADAnS 寄存器所指定的模拟输入引脚（ANInm），并且连续执行 A/D 转换。A/D 转换结果保存在与模拟输入引脚相对应的 ADAnCRm 寄存器中。

当所有指定的模拟输入引脚都结束转换时，将产生 A/Dn 转换结束中断请求信号（INTADn）。A/D 转换结束后，除非将 ADAnM0.ADAnCE 位清零，否则转换会从 ANIn0 引脚重新开始。要重新启动 A/D 转换^注，无需置位 ADAnM0.ADAnCE（1）。

在连接扫描模式中，仅支持 1 缓冲区模式。

注 在软件触发连续扫描模式中，除非 ADAnM0.ADAnCE 清零，否则 A/D 转换操作不会停止。如果在下一次 A/D 转换结束前没有读取 ADAnCRm 寄存器，那么其内容会被重写。

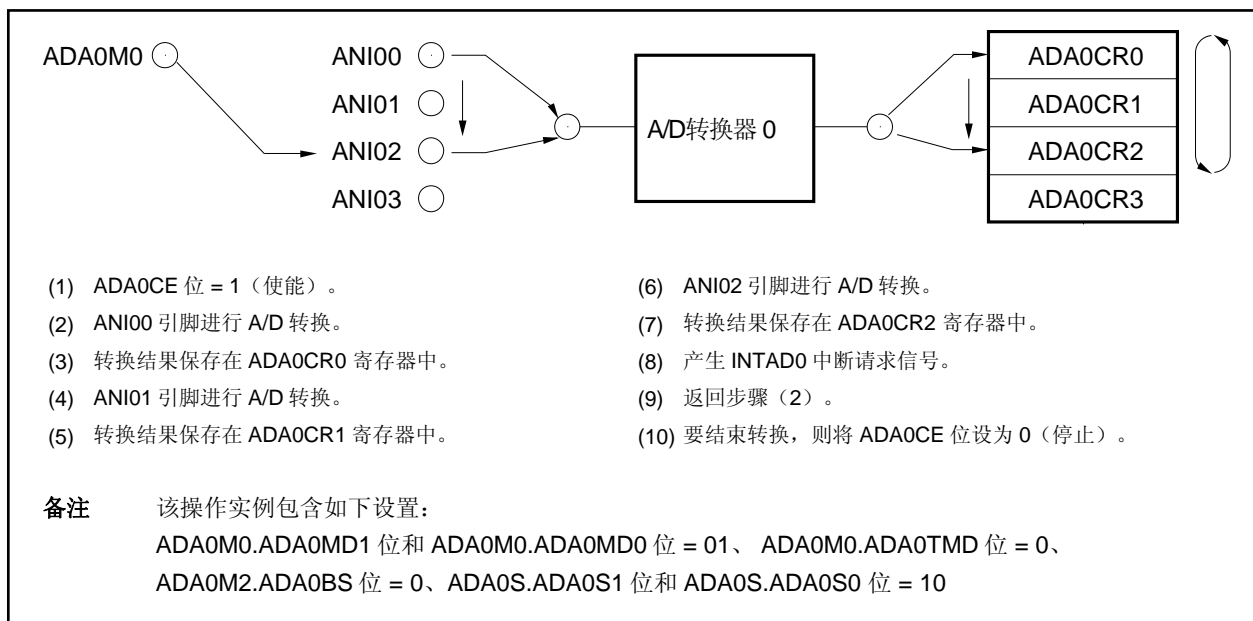
该模式适用于持续监视多个模拟输入的应用。

模拟输入引脚	A/D 转换结果寄存器
ANIn0	ADAnCR0
⋮	⋮
ANInm ^注	ADAnCRm

注 由 ADAnS.ADAnS0 位和 ADAnS.ADAnS1 位进行设置。

备注 n = 0、1
m = 0 至 3

图 11-12. 连续扫描模式操作实例（软件触发连续扫描）



11.5.3 单脉冲选择模式的操作

在该模式中，由 ADAnS 寄存器指定的模拟输入引脚 (ANInm) 连续进行 A/D 转换。转换结果保存在 ADAnCRm 寄存器中。在单脉冲选择模式中，根据存储 A/D 转换结果的方法，可支持 1 缓冲区模式和 4 缓冲区模式。

备注 n = 0、1
m = 0 至 3

(1) 1 缓冲区模式 (软件触发单脉冲选择: 1 缓冲区)

在该模式中，模拟输入引脚 (ANInm) 的电压将执行一次 A/D 转换。转换结果保存在 ADAnCRm 寄存器中。ANInm 引脚与 ADAnCRm 寄存器一一对应。

每次执行 A/D 转换时，都会产生 A/Dn 转换结束中断请求信号 (INTADn)，并结束 A/D 转换。A/D 转换结束后，转换操作停止。

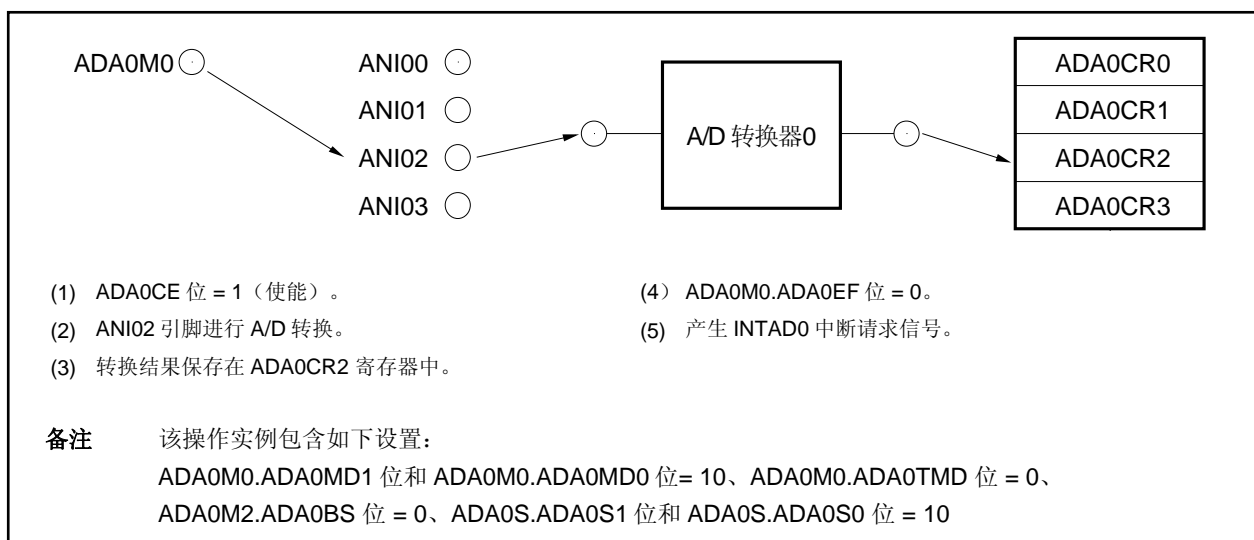
ADAnM0.ADAnCE 位设为 (1) 时，可重新启动 A/D 转换。

该模式适用于读取每个第一次 A/D 转换结果的应用操作。

模拟输入引脚	A/D 转换结果寄存器
ANInm	ADAnCRm

备注 n = 0、1
m = 0 至 3

图 11-13. 1 缓冲区模式操作举例 (软件触发单脉冲选择: 1 缓冲区)



(2) 4 缓冲区模式（软件触发单脉冲选择：4 缓冲区）

在该模式中，一个模拟输入引脚（ANInm）的电压将会进行四次 A/D 转换，其结果将会保存在 ADAnCRm 寄存器中。

第 4 次 A/D 转换操作结束时，将产生 A/Dn 转换结束中断请求信号（INTADn）。A/D 转换结束后，转换操作停止。

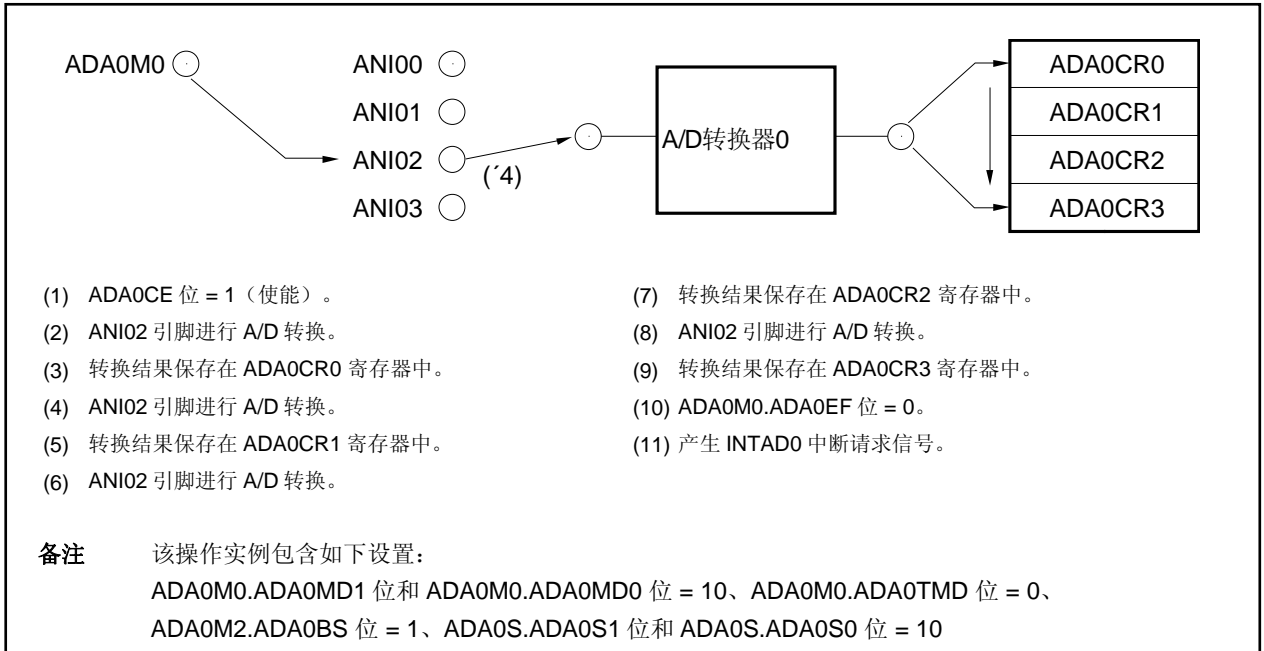
ADAnM0.ADAnCE 位设为（1）时，可重新启动 A/D 转换。

该模式适用于计算 A/D 转换结果平均值的应用操作。

模拟输入引脚	A/D 转换结果寄存器
ANInm	ADAnCR0
ANInm	ADAnCR1
ANInm	ADAnCR2
ANInm	ADAnCR3

备注 n = 0、1
m = 0 至 3

图 11-14. 4 缓冲区模式操作举例（软件触发单脉冲选择：4 缓冲区）



11.5.4 单脉冲扫描模式的操作

在该模式中，从 ANIn0 引脚处连续选择由 ADAnS 寄存器所指定的模拟输入引脚（ANInm），并且连续执行 A/D 转换。A/D 转换结果保存在与模拟输入引脚相对应的 ADAnCRm 寄存器中。

当所有指定的模拟输入引脚都结束转换时，将产生 A/Dn 转换结束中断请求信号（INTADn）。A/D 转换结束后，转换操作停止。

ADAnM0.ADAnCE 位设为（1）时，可重新启动 A/D 转换。

在单脉冲扫描模式中，仅支持 1 缓冲区模式。

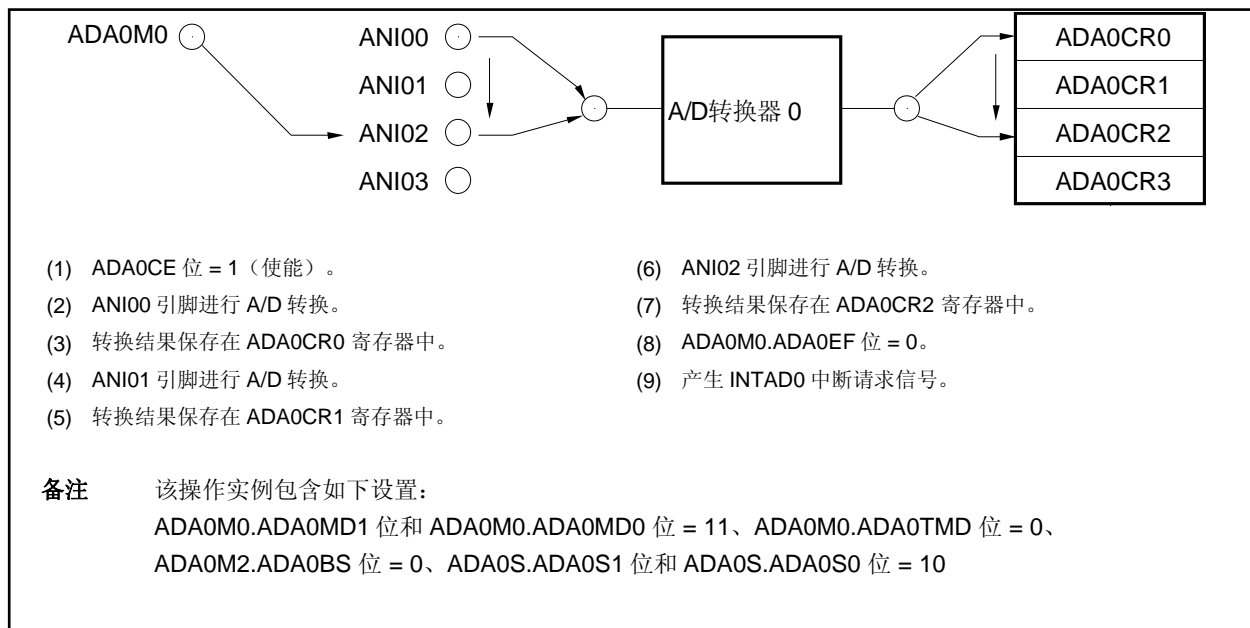
该模式适用于持续监视多个模拟输入的应用。

模拟输入引脚	A/D 转换结果寄存器
ANIn0	ADAnCR0
⋮	⋮
ANInm ^注	ADAnCRm

注 由 ADAnS.ADAnS0 位和 ADAnS.ADAnS1 位进行设置。

备注 n = 0、1
m = 0 至 3

图 11-15. 单脉冲扫描模式操作实例（软件触发单脉冲扫描）



11.6 定时器触发模式中的操作

使用 A/D 转换器 n 时，利用自定时器（马达控制功能）的 A/D 转换开始触发信号（TQTADT1 n ）来指定转换时序（参见图 11-2）。

- A/D 转换器 0 的定时器触发：TQTADT10
- A/D 转换器 1 的定时器触发：TQTADT11

TQTADT1 n 信号通过使用 TMQ1 选项寄存器 2（TQ1OPT2）的 TQ1AT00 至 TQ1AT03 位以及 TMQ1 选项寄存器 3（TQ1OPT3）的 TQ1AT10 至 TQ1AT13 位来设置。 可选为 A/D 转换开始触发（定时器触发）的马达控制功能，其触发源是 INTTP1CC0，INTTP1CC1，INTTQ1CC0，以及 INTTQ1OV 信号（可以选择两个或更多的信号）。

当 ADAnM2.ADAnTMD1 位设为 1 时，在为马达控制功能所选择的 A/D 转换开始触发信号（TQTADT1 n ）的上升沿处开始 A/D 转换。

ADAnM0.ADAnCE 位设为 1 时，A/D 转换器等待触发并在 A/D 转换开始触发信号输入时启动转换。

A/D 转换结束后，转换结果将会保存在 A/D n 转换结果寄存器 m （ADAnCR m ）中，同时产生 A/D n 转换结束中断请求信号（INTAD n ）。

A/D 转换结束后，A/D 转换器等待触发而与通过 ADAnM0.ADAnMD1 位和 ADAnM0.ADAnMD0 位所设置的操作模式无关。

转换开始时，ADAnM0.ADAnEF 位设为 1（转换正在进行）。 然而，在转换器等待触发时，ADAnM0.ADAnEF 位 = 0（转换停止）。

如果在 A/D 转换期间输入有效触发，转换操作停止并从头开始。 如果在 A/D 转换期间写入 ADAnM0，ADAnM2，和 ADAnS 寄存器，那么 A/D 转换器停止转换并等待下一次触发。

注意事项 在定时器触发模式中，应确保在短于由 ADAnM1.ADAnFR1 位和 ADAnM1.ADAnFR0 位所指定的转换时钟的最小个数的间隔时间内，不产生 A/D 转换开始触发信号（A/D 转换启动时序）。如果在间隔时间内产生了 A/D 转换开始触发信号，且间隔时间短于转换时钟的最小个数，那么最近一次触发将会是有效的。

备注 $n = 0, 1$
 $m = 0$ 至 3

11.6.1 连续选择模式/单脉冲选择模式的操作

在该模式中，由 ADAnS 寄存器指定的模拟输入引脚的电压（ANInm）进行 A/D 转换。转换结果保存在 ADAnCRm 寄存器中。在连接选择模式或单脉冲选择模式中，根据存储 A/D 转换结果的方法，可支持 1 缓冲区模式和 4 缓冲区模式。

备注 n = 0、1
m = 0 至 3

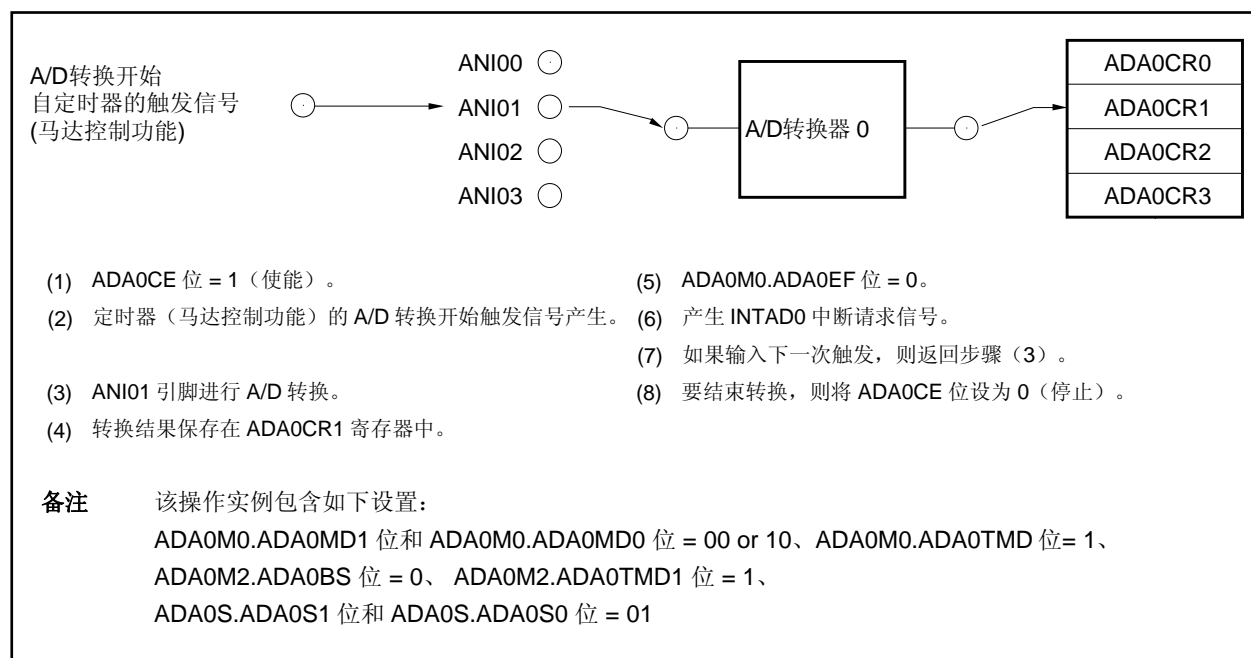
(1) 1 缓冲区模式操作（通过定时器触发进行连续选择/单脉冲选择的 1 缓冲区模式）

在该模式中，将自定时器（马达控制功能）的 A/D 转换开始触发信号作为触发，模拟输入引脚（ANInm）的电压则执行一次 A/D 转换，且结果保存在 ADAnCRm 寄存器中。ANInm 引脚与 ADAnCRm 寄存器一一对应。每次 A/D 转换均生成 A/Dn 转换结束中断请求信号（INTADn）。A/D 转换结束后，A/D 转换器将等待触发。该模式适用于读取每个第一次 A/D 转换结果的应用操作。

模拟输入引脚	A/D 转换结果寄存器
ANInm	ADAnCRm

备注 n = 0、1
m = 0 至 3

图 11-16. 1 缓冲模式操作实例
(通过定时器触发进行连续选择/单脉冲选择的 1 缓冲模式)



(2) 4 缓冲区模式操作（通过定时器触发进行连续选择/单脉冲选择的 4 缓冲区模式）

在该模式中，将自定时器（马达控制功能）的 A/D 转换开始触发信号用作触发，模拟输入引脚（ANInm）的电压则执行四次 A/D 转换，且结果都保存在 ADAnCRm 寄存器中。

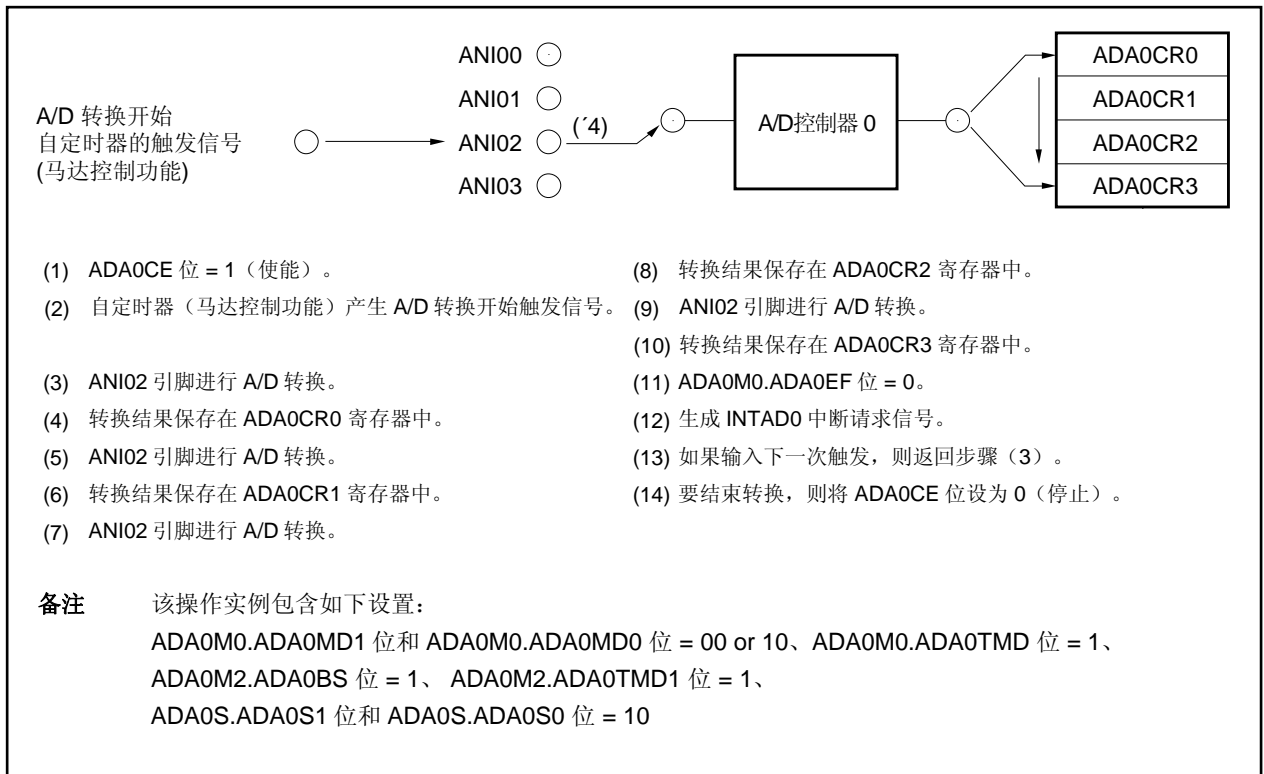
四次 A/D 转换结束后，将产生 A/Dn 转换结束中断请求信号（INTADn）。A/D 转换结束后，A/D 转换器将等待触发。

该模式适用于计算 A/D 转换结果平均值的应用操作。

模拟输入引脚	A/D 转换结果寄存器
ANInm	ADAnCR0
ANInm	ADAnCR1
ANInm	ADAnCR2
ANInm	ADAnCR3

备注 n = 0、1
 m = 0 至 3

图 11-17. 4 缓冲模式操作实例
(通过定时器触发进行连续选择/单脉冲选择的 4 缓冲模式)



11.6.2 连续扫描模式/单脉冲扫描模式的操作

在该模式中，由 ADAnS 寄存器指定的模拟输入引脚（ANInm）会从 ANIn0 引脚中连续进行选择，ANIn0 引脚将定时器（马达控制功能）的 A/D 转换开始触发信号用作触发，并连续进行 A/D 转换。A/D 转换结果保存在与模拟输入引脚相对应的 ADAnCRm 寄存器中。

当所有指定的模拟输入引脚都结束转换时，将产生 A/Dn 转换结束中断请求信号（INTADn）。A/D 转换结束后，A/D 转换器将等待触发。

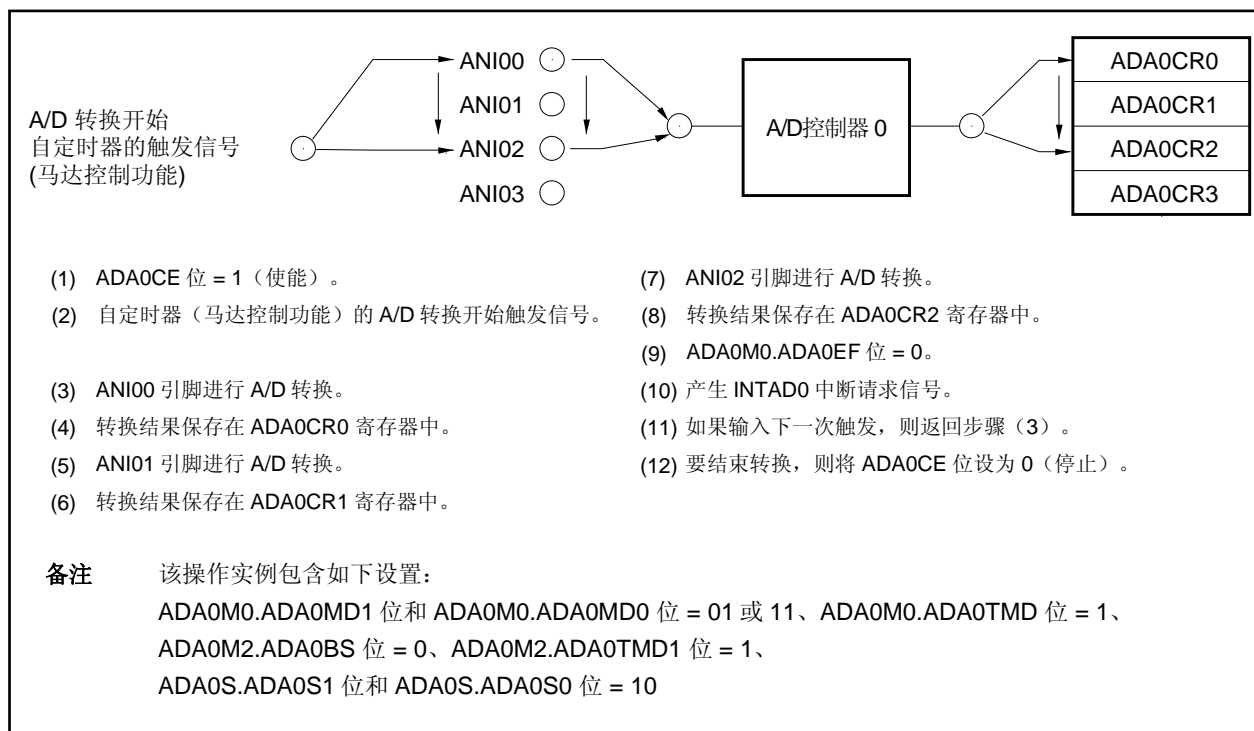
该模式适用于持续监视多个模拟输入引脚的应用操作。

在连续扫描模式或单脉冲扫描模式中，仅支持 1 缓冲区模式。

模拟输入引脚	A/D 转换结果寄存器
ANIn0	ADAnCR0
ANIn1	ADAnCR1
ANIn2	ADAnCR2
ANIn3	ADAnCR3

备注 n = 0、1
m = 0 至 3

图 11-18. 扫描模式操作实例（通过定时器触发进行连续扫描/单脉冲扫描）



11.7 外部触发模式中的操作

在外部触发模式中，模拟输入引脚（ANIn0 至 ANIn3）将在 ADTRGn 引脚输入时序时进行 A/D 转换。

ADTRG0 引脚的复用功能相当于 P04/INTP4 引脚，而 ADTRG1 引脚的复用功能则相当于 P05/INTP5 引脚。要设置外部触发模式，则使用 A/D 转换器 0 将端口模式控制寄存器 0（PMC0）的 PMC04 位设为 1 并将 ADA0M2.ADA0TMD1 位设为 0。使用 A/D 转换器 1 时，则应将端口模式控制寄存器 0（PMC0）的 PMC05 位设为 1 并 ADA1M2.ADA1TMD1 位设为 0。

外部触发模式中，通过设置 ADAnM0.ADAnETS1 位和 ADAnM0.ADAnETS0 位，可以为外部输入信号的有效沿指定上升沿，下降沿或双边沿。

ADAnM0.ADAnCE 位置(1)时，A/D 转换器将会等待触发，并在触发信号从 ADTRGn 引脚中输入时开始转换。

A/D 转换结束后，转换结果将会保存在 A/Dn 转换结果寄存器 m（ADAnCRm）中，同时产生 A/Dn 转换结束中断请求信号（INTADn）。

A/D 转换结束后，A/D 转换器等待触发而与通过 ADAnM0.ADAnMD1 位和 ADAnM0.ADAnMD0 位所设置的操作模式无关。

转换开始时，ADAnM0.ADAnEF 位设为 1（转换正在进行）。然而，在转换器等待触发时，ADAnEF 位=0（转换停止）。

如果在 A/D 转换期间输入有效触发，转换操作停止并从头开始。如果在 A/D 转换期间写入 ADAnM0，ADAnM2，和 ADAnS 寄存器，那么 A/D 转换器停止转换并等待下一次触发。

备注 n = 0、1
 m = 0 至 3

11.7.1 连续选择模式/单脉冲选择模式的操作

该模式中，由 ADAnS 寄存器指定的模拟输入引脚（ANInm）进行 A/D 转换。转换结果保存在 ADAnCRm 寄存器中。在连续选择模式或单脉冲选择模式中，有两种选择模式：根据保存 A/D 转换结果的方式，分为 1 缓冲区模式和 4 缓冲区模式。

备注 n = 0、1
 m = 0 至 3

(1) 1 缓冲区模式（通过外部触发进行连续选择/单脉冲选择的 1 缓冲区模式）

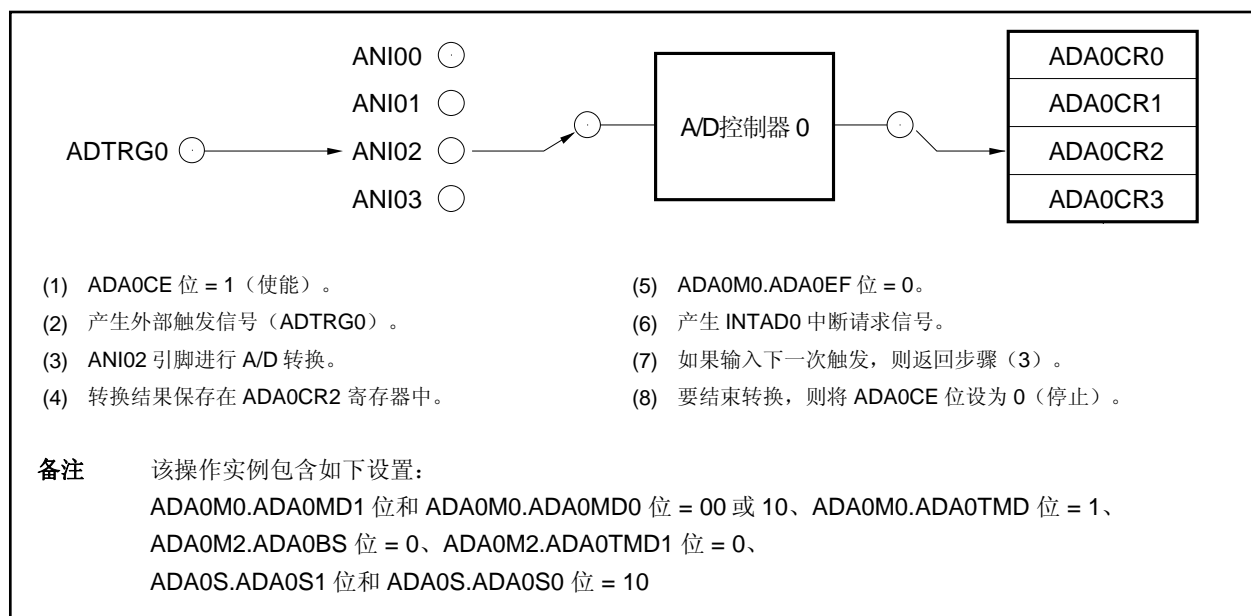
在该模式中，将 ADTRGn 信号用作触发信号，模拟输入引脚（ANInm）的电压将进行一次 A/D 转换。转换结果保存在 ADAnCRm 寄存器中。ANInm 引脚与 ADAnCRm 寄存器一一对应。

每次 A/D 转换均生成 A/Dn 转换结束中断请求信号（INTADn）。A/D 转换结束后，A/D 转换器将等待触发。该模式适用于读取每个第一次 A/D 转换结果的应用操作。

模拟输入引脚	A/D 转换结果寄存器
ANInm	ADAnCRm

备注 n = 0、1
 m = 0 至 3

图 11-19. 1 缓冲区模式操作实例
(通过外部触发进行连续选择/单脉冲选择的 1 缓冲区模式)



(2) 4 缓冲区模式（通过外部触发进行连续选择/单脉冲选择的 4 缓冲区模式）

在该模式中，通过将 ADTRGn 信号用作触发信号，一个模拟输入引脚（ANInm）的电压将会进行四次 A/D 转换，其结果将会保存在 ADAnCRm 寄存器中（n = 0、1，m = 0 至 3）。

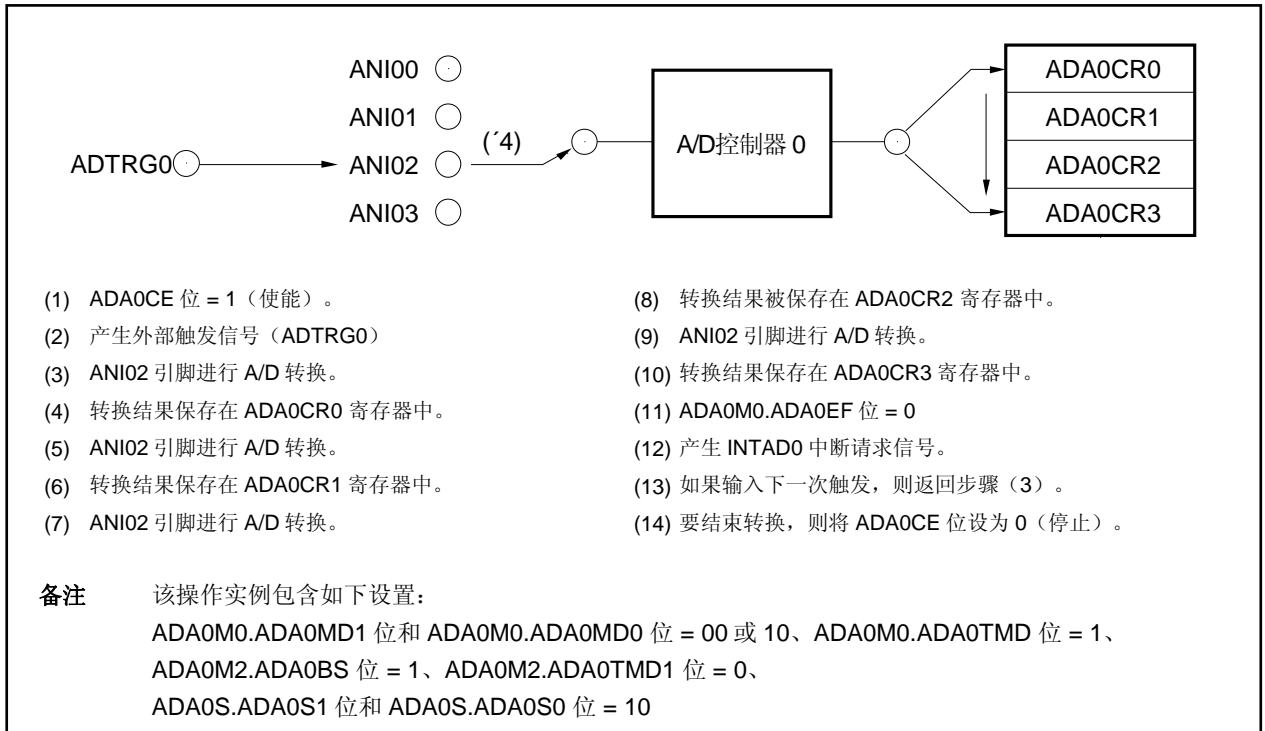
四次 A/D 转换结束后，将产生 A/Dn 转换结束中断请求信号（INTADn）。 A/D 转换结束后，A/D 转换器将等待触发。

该模式适用于计算 A/D 转换结果平均值的应用操作。

模拟输入引脚	A/D 转换结果寄存器
ANInm	ADAnCR0
ANInm	ADAnCR1
ANInm	ADAnCR2
ANInm	ADAnCR3

备注 n = 0、1
 m = 0 至 3

图 11-20. 4 缓冲模式操作实例
(通过外部触发进行连续选择/单脉冲选择的 4 缓冲区模式)



11.7.2 连续扫描模式/单脉冲扫描模式的操作

在该模式中，从 ANIn0 引脚（该引脚将 ADTRGn 信号用作触发信号）处连续选择由 ADAnS 寄存器所指定的模拟输入引脚（ANInm），并连续执行 A/D 转换。A/D 转换结果保存在与模拟输入引脚相对应的 ADAnCRm 寄存器中。

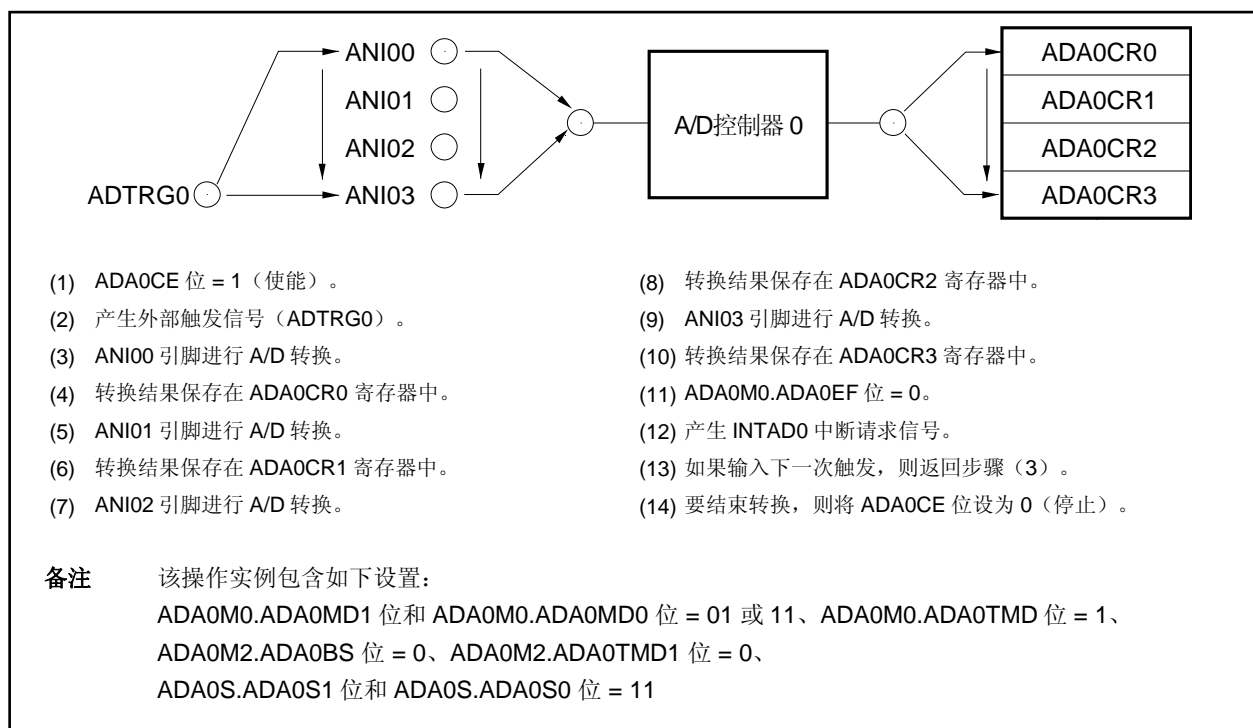
当所有指定的模拟输入引脚都结束转换时，将产生 A/Dn 转换结束中断请求信号（INTADn）。A/D 转换结束后，A/D 转换器将等待触发。

该模式适用于持续监视多个模拟输入的应用。

模拟输入引脚	A/D 转换结果寄存器
ANIn0	ADAnCR0
ANIn1	ADAnCR1
ANIn2	ADAnCR2
ANIn3	ADAnCR3

备注 n = 0、1
m = 0 至 3

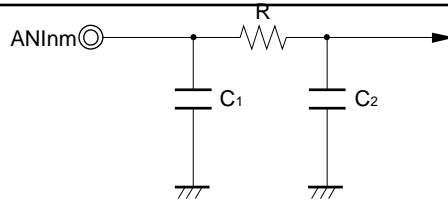
图 11-21. 扫描模式操作实例（通过外部触发进行连续扫描/单脉冲扫描）



11.8 内部等效电路

下图显示了模拟输入模块的等效电路：

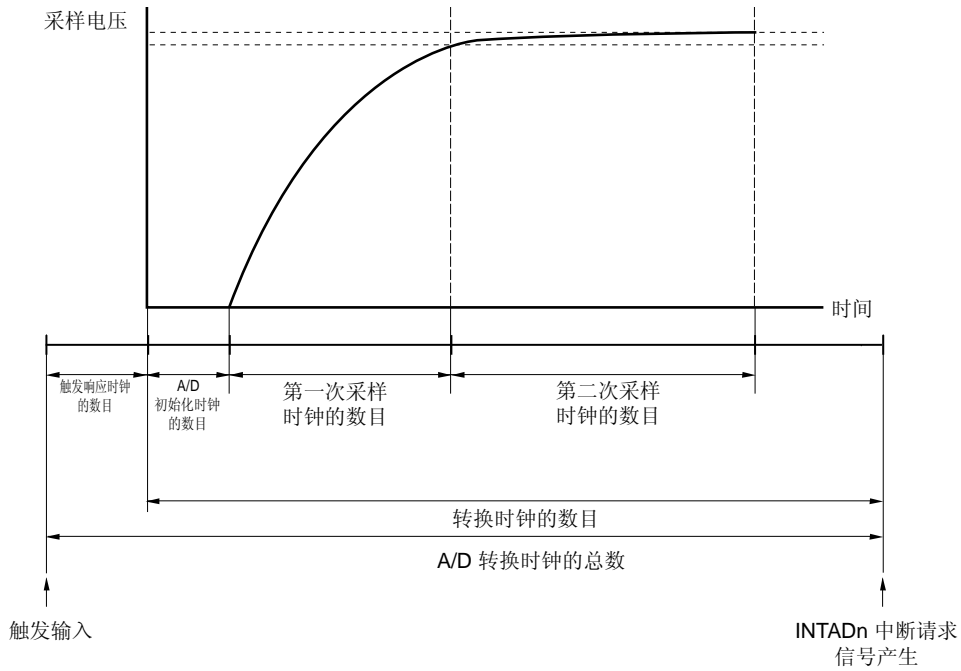
图 11-22. ANInm 引脚的内部等效电路



R	C ₁	C ₂
4.0 kΩ	15 pF	4.2 pF

- 备注
1. 显示了最大值（参考值）。
 2. n = 0、1
m = 0 至 3

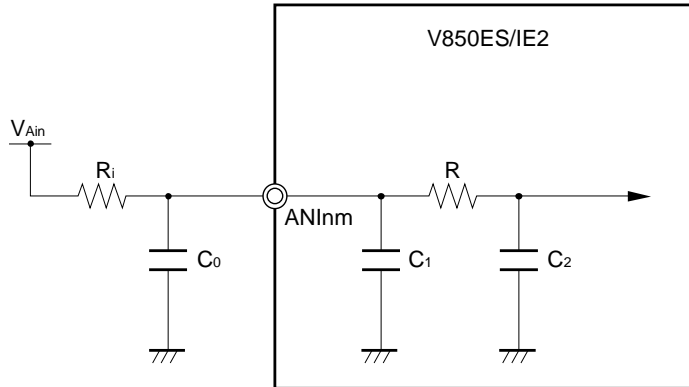
注意事项 当 A/D 初始化后，A/D 转换开始时，A/D 转换器 0 和 1 执行第一次采样。此期间，采样电源几乎完全充电。因此，可以在这种条件下计算采样误差。在此之后，转换器执行第二次采样以纠正误差。A/D 转换器 0 和 1 的采样周期是从转换开始直到第二次采样结束的时间。此期间，应避免噪声。



ADAnM1 寄存器		A/D 初始化时钟的数目	第一次采样时钟的数目	第二次采样时钟的数目
ADAnFR1 位	ADAnFR0 位			
0	1	2	16	18
1	0	3	24	27
1	1	4	32	36

计算 A/D 转换器 0 和 1 的总体误差实例显示如下：

图 11-23. 计算 A/D 转换器 0 和 1 的总体误差实例



f_{xx} (MHz)	转换时间 (μs)	A/D 初始化时钟 (μs)	第一次采样 (μs)	第二次采样 (μs)	R (k Ω)	C ₁ (pF)	C ₂ (pF)	C ₀ (pF)	R _i (k Ω)	采样误差 (LSB) ^注
20	3.10 (62 个时 钟)	0.1 (2 个时钟)	0.8 (16 个时 钟)	0.9 (18 个时 钟)	4.0	15	4.2	100	1.0	18.4
								100	0.5	0.2
								100	0.25	0.1 或更低
								100	0.1	0.1 或更低
								50	1.0	1.0
								50	0.5	0.1 或更低
								50	0.25	0.1 或更低
								50	0.125	0.1 或更低

注 考虑信号源阻抗时的误差为“采样误差+总体误差”。

- 备注
1. 这些值是在施加到 V_{Ain} 上的电压从 0 V 至 5 V 时，由 R_i 和 C_0 模拟 C_2 电压计算的参考值。
 2. $n = 0、1$
 $m = 0$ 至 3
 3. f_{xx} : 系统时钟频率

11.9 操作时的注意事项

11.9.1 停止转换操作

转换操作期间，当 ADAnM0.ADAnCE 位清零时，转换操作停止且转换结果将不会保存在 A/Dn 转换结果寄存器 m (ADAnCRm) 中。

在所有模式中，即使都产生 A/Dn 转换结束中断请求信号 (INTADn) 后，ADAnCE 位也不会清零。

备注 n = 0、1, m = 0 至 3

11.9.2 定时器/外部触发间隔

在定时器触发模式或外部触发模式中，确保触发发生间隔大于由 ADAnM1.ADAnFR1 和 ADAnM1.ADAnFR0 位指定的转换时钟的总数（参见表 11-2 转换时钟的数目）。

(1) 当 $0 < \text{触发发生间隔} < \text{A/D 转换时钟的总数}$

转换操作期间，输入定时器/外部触发后，转换操作异常终止，且根据最后输入的定时器/外部触发重新开始转换。

当转换操作终止时，之前转换操作的转换结果将不会保存在 ADAnCRm 寄存器中。因此，注意：不能保证 INTADn 信号的产生和保存结果至 ADAnCRm 寄存器中操作的有效性。

备注 n = 0、1, m = 0 至 3

(2) 当触发发生间隔 \geq A/D 转换时钟的总数时

产生 INTADn 信号，且转换结束时的值将正确地保存到 ADAnCRm 寄存器中。设计以使触发发生间隔等于或大于 A/D 转换时钟的总数。

备注 n = 0、1, m = 0 至 3

11.9.3 待机模式中的操作

(1) HALT 模式

该模式中，A/D 转换继续进行。

(2) IDLE 模式，STOP 模式

当至 A/D 转换器 0 和 1 的时钟供给停止时，不执行转换操作。

通过可屏蔽中断请求信号输入引脚^注解除这些模式时，ADAnM0, ADAnM1, ADAnM2, ADAnS 寄存器以及 A/Dn 转换结果寄存器 m (ADAnCRm) 继续保持原有值。然而，若在转换操作期间设置 IDLE 或 STOP 模式，转换操作将挂起。这时，如果通过可屏蔽中断请求信号输入引脚^注解除了该模式，那么将恢复转换操作。同时，可能产生 A/Dn 转换结束中断请求信号 (INTADn)，但写入 ADAnCRm 寄存器中的转换结果不确定。

注 INTPO 至 INTP5

备注 n = 0、1, m = 0 至 3

11.9.4 定时器触发模式中的定时器中断请求信号

定时器中断请求信号 (TQTADT1n) 变为 A/D 转换开始的触发信号, 并开始进行转换操作。当这种情况发生时, 定时器中断请求信号也将用作 CPU 的中断。为了产生避免 CPU 中断, 应利用中断控制寄存器的屏蔽位禁止中断。

备注 n = 0、1

11.9.5 稳定时间内重新转换时的启动触发输入

如果稳定时间结束时序和寄存器写入冲突, 或者稳定时间结束时序与触发输入冲突, 那么将再次插入稳定时间。

11.9.6 A/D 转换结果的误差

根据电压的波动或噪声的影响, A/D 转换结果可能有所变化。为了减小变化, 在程序中应用抵消措施, 例如平均 A/D 转换结果。

11.9.7 A/D 转换结果滞后特性

逐次逼近 A/D 转换器维持内部采样&保持电容的模拟输入电压, 同时进行 A/D 转换。A/D 转换完成后, 模拟输入电压仍然保留在内部采样&保持电容器上。因此, 如果自模拟输入源的输出阻抗太高, 则可能产生下述现象。

- 当相同通道用于 A/D 转换时, 如果电压比先前 A/D 转换的电压高或低, 受到先前值影响, 转换结果可能出现滞后特性。即使在相同电位执行转换, 结果也可能会变化。
- 当转换模拟输入通道时, 受到先前通道值影响, 转换结果可能出现滞后特性。这是因为只有一个 A/D 转换器用于 A/D 转换。即使要在相同电位执行转换, 结果也可能会变化。

要获取更多准确的转换结果, 降低模拟输入源的输出阻抗或在同一通道连续执行两次 A/D 转换, 并丢弃第一个转换结果。

11.9.8 设置单脉冲模式和软件触发模式的限制

如果在单脉冲选择模式和软件触发模式（ADAnM0 寄存器 = 1010XX0XB）或单脉冲扫描模式和软件触发模式（ADAnM0 寄存器 = 1011XX0XB）中设置 A/D 转换器 0 和 1，那么在 A/D 转换操作完成的基础上对 ADAnM0，ADAnM2，和 ADAnS 寄存器中的任一寄存器写入数据时，应该在新的条件下重新执行转换操作。然而，并不执行重新转换操作，而是允许转换操作使能（ADAnM0.ADAnCE 位 = 1）且停止转换（ADAnM0.ADAnEF 位 = 0）。不产生 A/Dn 转换结束中断请求信号（INTADn），也不保存最后的 A/D 转换结果。然而，数据将正确写入 ADAnM0，ADAnM2，和 ADAnS 寄存器中的任一寄存器。

如果发生这种情况，将 ADAnM0.ADAnCE 位设为 1 可以恢复正常的操作。

例如，如果 ANIn0 和 ANIn1 引脚在扫描模式设置（ADAnS 寄存器 = 00000001B），在单脉冲模式和软件触发模式（ADAnM0 寄存器 = 1011XX0XB）中，A/D 转换操作完成后将数据写入 ADAnM0 寄存器，那么 ANIn0 引脚的信号将会正确地转换且转换结果正确地保存在 ADAnCR0 寄存器中。然而，在 A/D 转换完成前对 ANIn1 引脚的信号执行转换的结果不会保存在 ADAnCR1 寄存器，也不会产生 INTADn 中断请求信号。

[相应措施]

执行以下步骤<1>至 <3>中的任一步骤可以避免以上的限制。

- <1> 对 ADAnM0，ADAnM2，和 ADAnS 寄存器中的任一寄存器执行写操作前，应确认 A/D 转换已停止（ADAnM0.ADAnEF 位 = 0）。
- <2> 禁止中断（PSW.ID 位 = 1）后，连续执行对 ADAnM0，ADAnM2，和 ADAnS 寄存器中的任一寄存器写入数据的指令和将 ADAnM0.ADAnCE 位设为 1 的指令，然后允许中断（PSW.ID 位 = 0）。此操作是用于避免 A/D 转换操作完成和写入 ADAnM0，ADAnM2，和 ADAnS 寄存器同时发生。例如，如果执行写指令的同时 A/D 转换操作完成，那么 A/D 转换将会停止，可以通过将 ADAnCE 位设为 1 来重新开始 A/D 转换。如果 ADAnM0.ADAnCE 位 = 1，那么 ADAnCE 位也要再次设为 1。
- <3> 通过将 ADAnCE 位清零来禁止 A/D 转换操作，先将数据写入 ADAnM0，ADAnM2，和 ADAnS 寄存器中的任一寄存器，再通过将 ADAnCE 位设为 1 来允许 A/D 转换操作，并启动 A/D 转换。

11.10 A/D转换器特征表的阅读方法

在这里，解释 A/D 转换器的专用术语

(1) 分辨率

这是可以识别的最小模拟输入电压。也就是说，数字输出中每一位的模拟输入电压的百分比称为 **LSB**（最低有效位）。对于满量程的 1 LSB 的百分比用 %FSR（满量程范围）表示。%FSR 表示可以转换为百分比的模拟输入电压的比值，不管分辨率如何，总可以通过以下的公式表示：

$$\begin{aligned} 1\%FSR &= (\text{可以转换的模拟输入电压的最大值} - \text{可以转换的模拟输入电压的最小值}) / 100 \\ &= (AV_{REFn} - 0) / 100 \\ &= AV_{REFn} / 100 \end{aligned}$$

当分辨率为 10 位时，1 LSB 表示如下：

$$\begin{aligned} 1 \text{ LSB} &= 1/2^{10} = 1/1,024 \\ &= 0.098\%FSR \end{aligned}$$

精确度和分辨率无关，而是由总体的误差决定的。

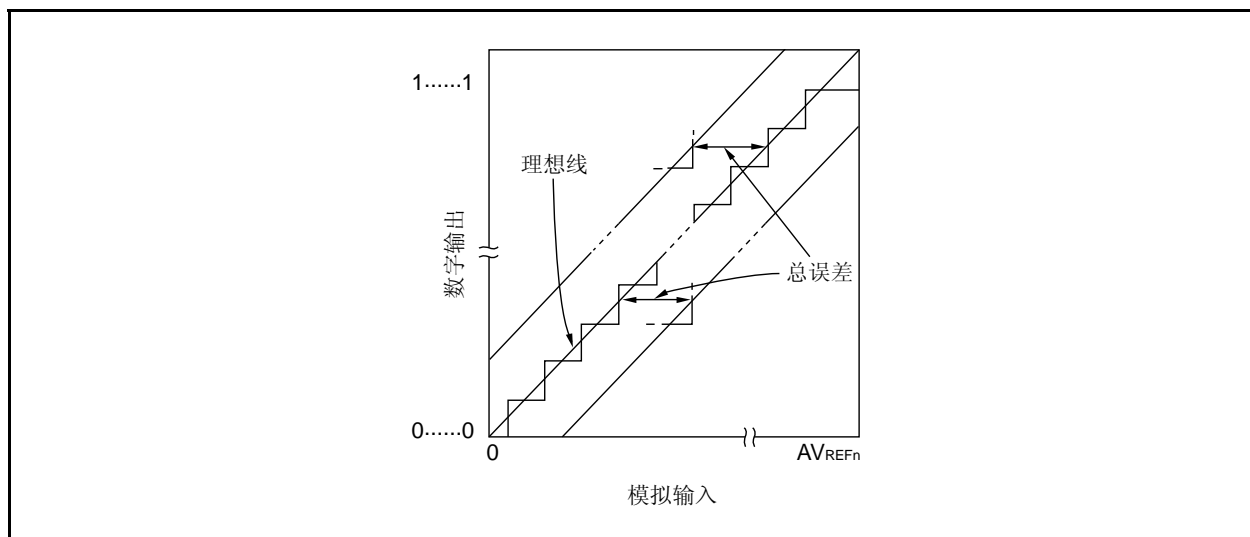
(2) 总误差

此项显示了实际测量值和理论值的最大误差值。

零标称误差、满量程误差、线形误差以及这些误差的综合构成总误差。

注意：在参数表的总体误差里没有包含量化误差。

图 11-24. 总误差

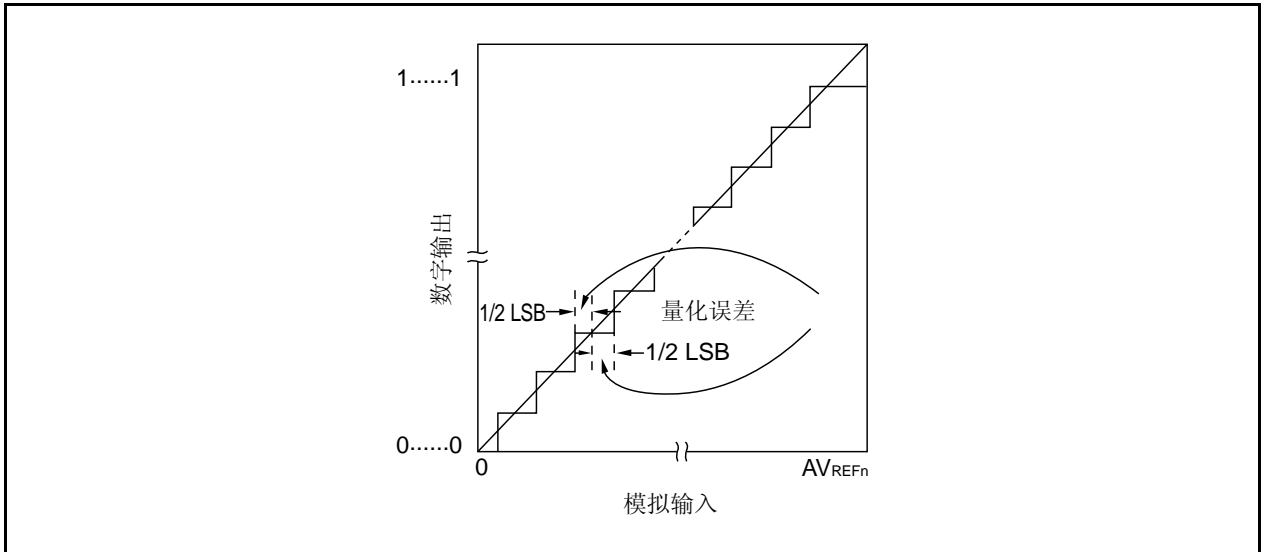


(3) 量化误差

当模拟值转换为数字值时，会产生 $\pm 1/2$ LSB 的误差。A/D 转换器中，在 $\pm 1/2$ LSB 范围内的模拟输入电压转换成相同的数字编码，所以不可能避免量化误差。

注意：量化误差不包括在参数表中的整体误差、零标称误差、满量程误差、积分线性误差和微分线性误差之中。

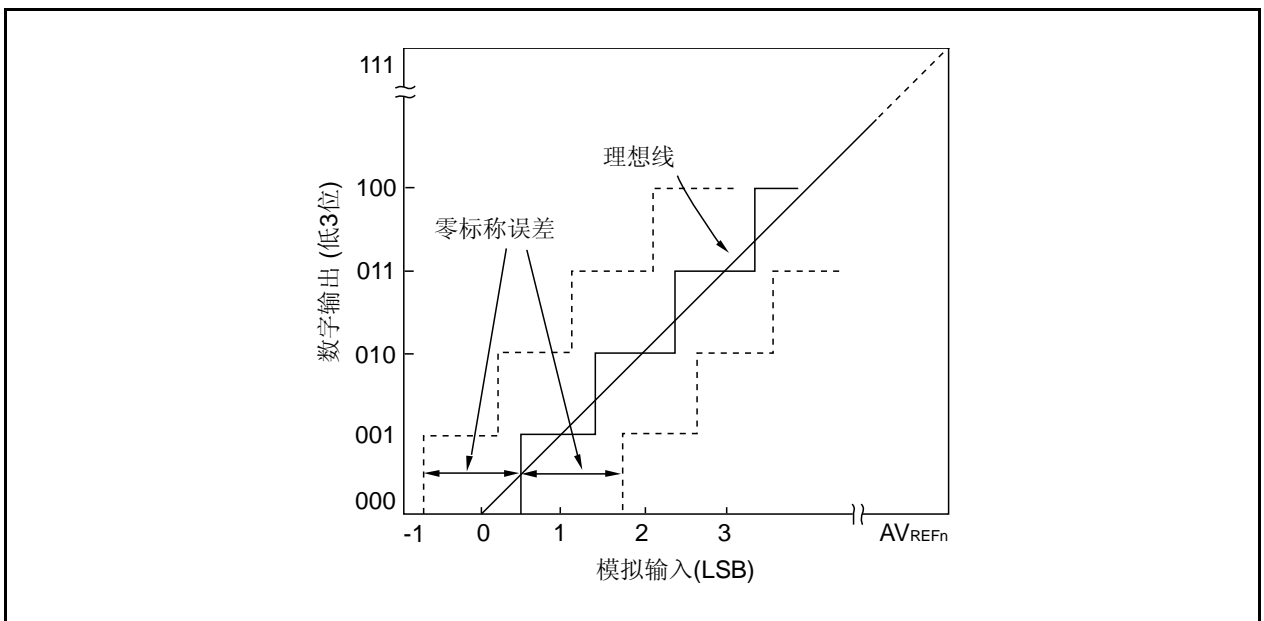
图 11-25. 量化误差



(4) 零标称误差

零度误差显示了数字输出从 0.....000 变为 0.....001 时，模拟输入电压的实际测量值与理论值（1/2 LSB）之间的差别。

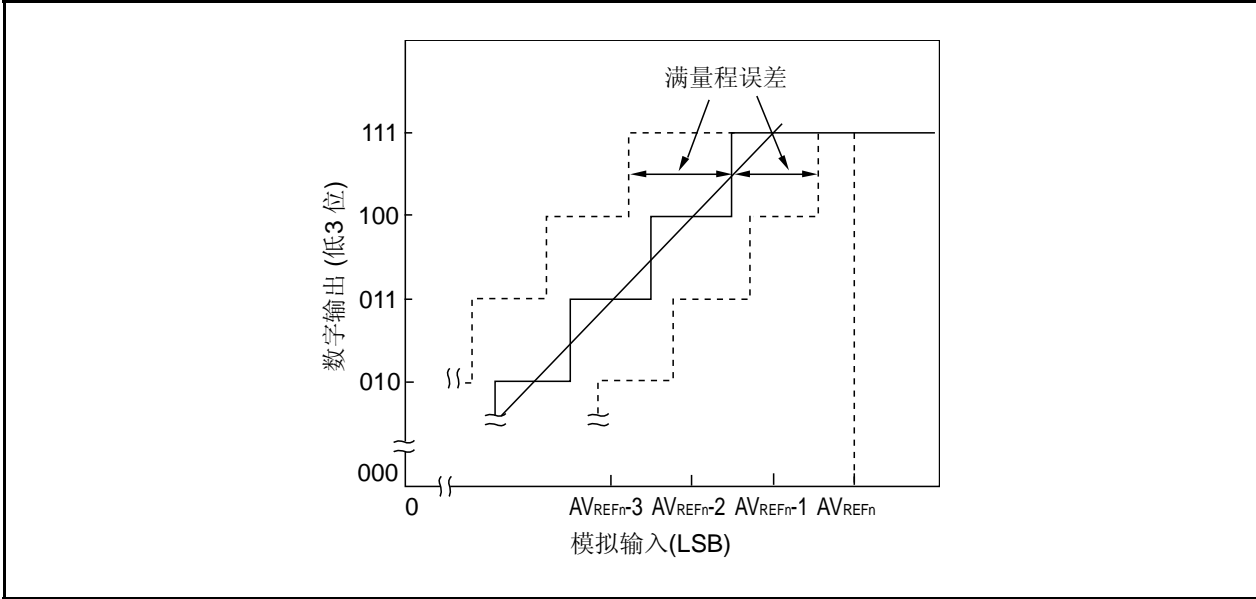
图 11-26. 零标称误差



(5) 满量程误差

满度误差显示了数字输出从 1.....110 变为 1.....111 时，模拟输入电压的实际测量值与理论值（满度值 - 3/2 LSB）之间的差别。

图 11-27. 满量程误差

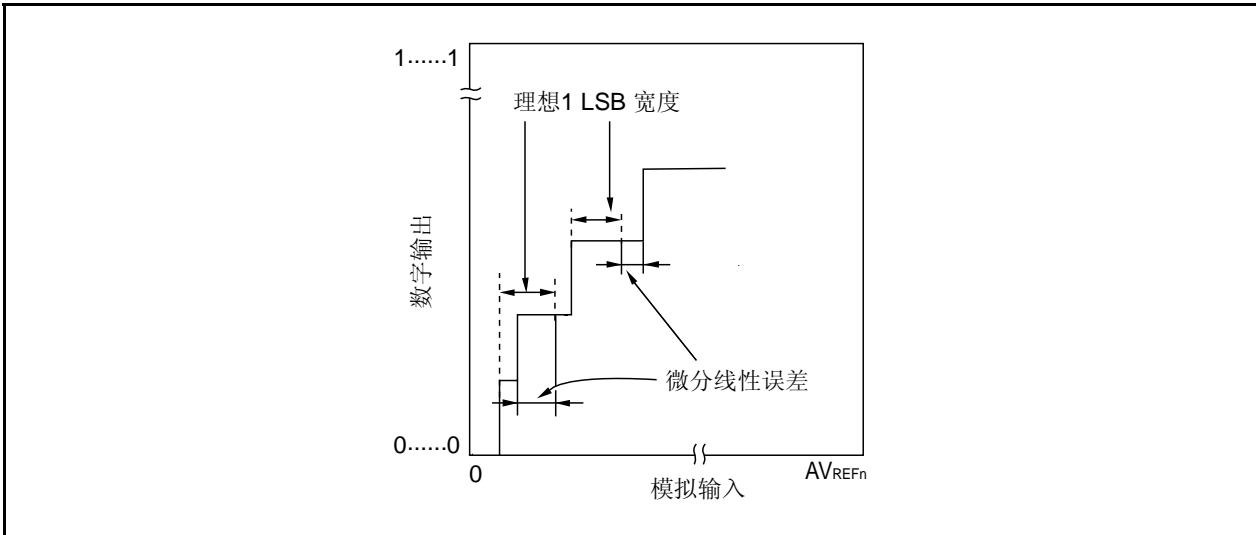


(6) 微分线性误差

当代码输出的理想宽度为 1 LSB 时，微分线性误差表示实际测量值与理想值间的差别。

当输入到相同通道的模拟输入引脚的电压从 AV_{SSn} 一位一位增加到 AV_{REFn} ，这显示了 A/D 转换的基本特性。关于输入电压增加或减少时的详细信息，或使用两个或更多个通道时的详细信息，请参见 11.10 (2) 总误差。

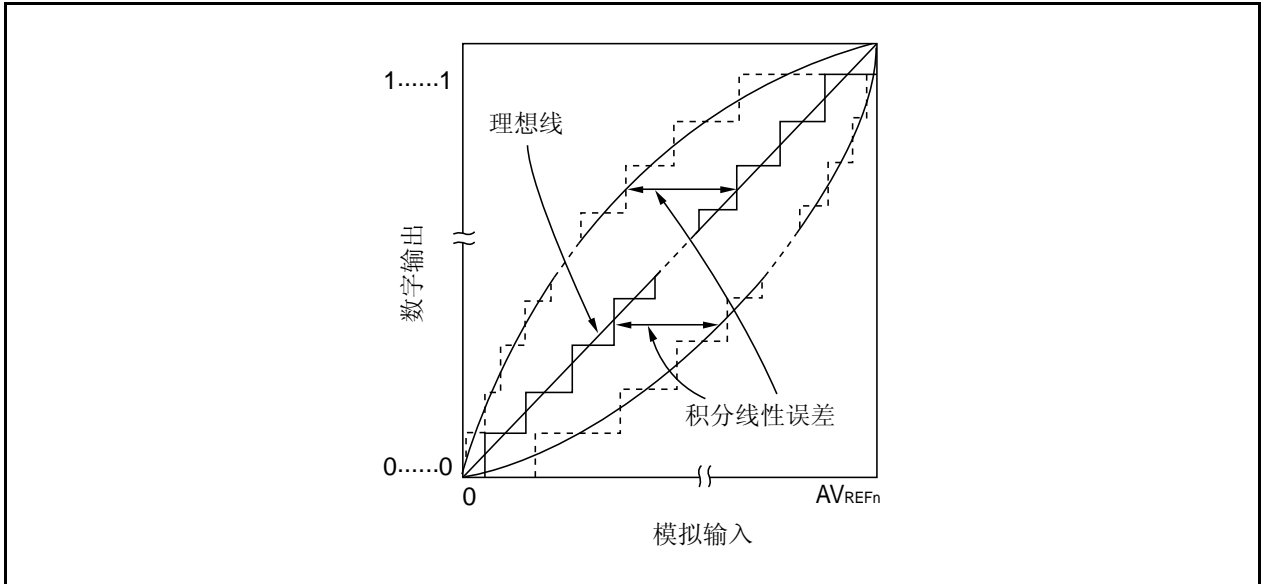
图 11-28. 微分线性误差



(7) 积分线性误差

积分线性误差说明了转换特征偏离理想线性关系的程度。 它表示当零标称误差和满量程误差均为 0 时，实际测量值和理想直线之间的最大误差。

图 11-29. 积分线性误差



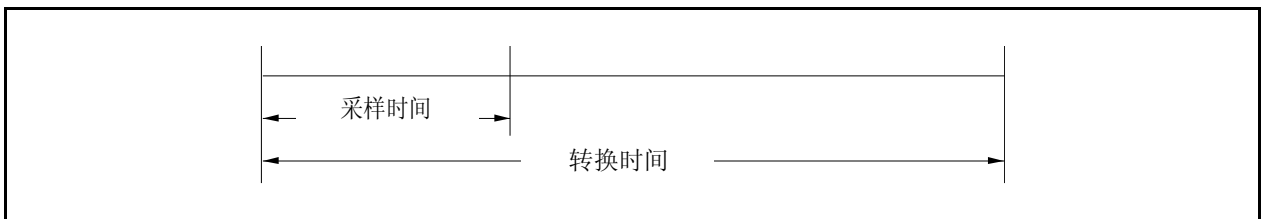
(8) 转换时间

转换时间表示从触发产生开始直到获取数字输出时的时间。
在特性表中，采样时间包括在转换时间内。

(9) 采样时间

采样时间即：模拟开关合上由'采样&保持'电路进行模拟电压采样的时间。

图 11-30. 采样时间



第十二章 异步串行接口A (UARTA)

V850ES/IE2 包括 UARTA0 和 UARTA1。

12.1 特性

- 传输速率: 300 bps 至 1.25 Mbps (使用 20 MHz 外部时钟 (f_{xx}) 和专用的波特率发生器)
- 全双工通信: 内部 UARTA 数据接收寄存器 n (UAnRX)
内部 UARTA 数据发送寄存器 n (UAnTX)
- 2 引脚配置: TXDAn: 数据发送输出引脚
RXDAn: 数据接收输入引脚
- 接收错误输出功能
 - 奇偶校验错误
 - 帧错误
 - 溢出错误
- 中断源: 3
 - 接收错误中断 (INTUAnRE): 该中断由 3 种接收中断的‘或运算’产生。
 - 接收结束中断 (INTUAnR): 在接收允许状态下, 该中断在串行传输结束后, 数据从移位寄存器传输至 UAnRX 寄存器时产生。
 - 发送使能中断 (INTUAnT): 在发送使能状态下, 该中断于数据从 UAnTX 寄存器传输至移位寄存器时产生。
- 字符长度: 7, 8 位
- 奇偶检验功能: 奇检验、偶校验、0 校验、无校验。
- 传输停止位: 1, 2 位
- 片内专用波特率发生器。
- 可选择 MSB-/LSB-先行传输。
- 发送/接收数据可以反向输入/输出。

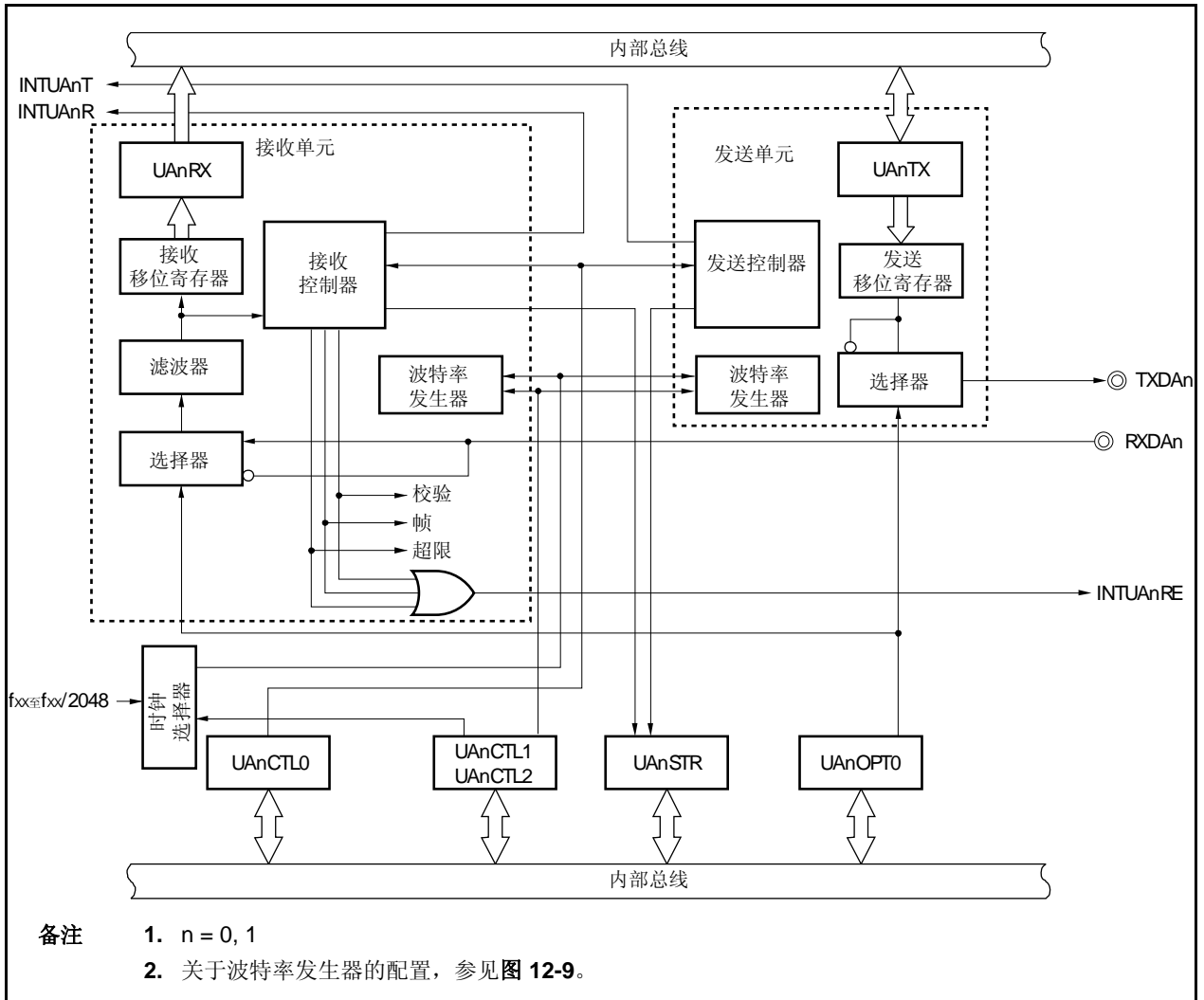
备注 n = 0, 1

12.2 配置

UARTAn 的功能框图如下所示：

<R>

图 12-1. UARTAn 的功能框图



UARTAn 包括以下硬件单元。

表 12-1. UARTAn 的配置

项目	配置
寄存器	UARTAn 控制寄存器 0 (UAnCTL0) UARTAn 控制寄存器 1 (UAnCTL1) UARTAn 控制寄存器 2 (UAnCTL2) UARTAn 选项控制寄存器 0 (UAnOPT0) UARTAn 状态寄存器 (UAnSTR) UARTAn 接收移位寄存器 UARTAn 接收数据寄存器 (UAnRX) UARTAn 发送移位寄存器 UARTAn 发送数据寄存器 (UAnTX)

(1) UARTAn 控制寄存器 0 (UAnCTL0)

UAnCTL0 寄存器为 8 位寄存器，用来指定 UARTAn 的操作。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

UAnCTL1 寄存器为 8 位寄存器，用来选择 UARTAn 的基准时钟(fuclk)。

(3) UARTAn 控制寄存器 2 (UAnCTL2)

UAnCTL2 寄存器为 8 位寄存器，用来控制 UARTAn 的波特率。

(4) UARTAn 选项控制寄存器 0 (UAnOPT0)

UAnOPT0 寄存器为 8 位寄存器，用来控制 UARTAn 的串行传输。

<R> (5) UARTAn 状态寄存器 (UAnSTR)

UAnSTRn 寄存器由标志所组成，当接收错误发生时，指明错误内容。当发生一个接收错误时，接收错误标志之一被置位（为 1）。

(6) UARTAn 接收移位寄存器

该寄存器为移位寄存器，用来将输入至 RXDAn 引脚的串行数据转换为并行数据。当接收 1 字节数据并检测到停止位时，接收数据传输至 UAnRX 寄存器。

该寄存器不能直接操作。

(7) UARTAn 接收数据寄存器 (UAnRX)

UAnRX 为 8 位寄存器，用来保持接收数据。当接收 7 位字节时，最高位存为 0 (当数据以 LSB 先行方式接收)。

在接收使能状态下，与 1 帧移入操作完毕同步，接收数据从 UARTAn 接收移位寄存器传输至 UAnRX 寄存器。

传输至 UAnRX 寄存器还导致输出接收结束中断请求信号 (INTUAnR)。

(8) UARTAn 发送移位寄存器

UARTAn 发送移位寄存器用来将来自于 UAnTX 寄存器的并行数据转换为串行数据。

当 1 字节数据从 UAnTX 寄存器传输发出。UARTAn 发送移位寄存器的数据由 TXDAn 引脚输出。

该寄存器不能直接操作。

(9) UARTAn 发送数据寄存器 (UAnTX)

UAnTX 寄存器为 8 位发送数据缓存器。当数据写入 UAnTX 寄存器后启动发送。当数据可以被写入 UAnTX 寄存器时 (当一帧数据从 UAnTX 寄存器传输至 UARTAn 发送移位寄存器时)，产生发送允许中断请求信号 (INTUAnT)。

12.3 控制寄存器

(1) UARTAn 控制寄存器 0 (UAnCTL0)

UAnCTL0 寄存器为 8 位寄存器，用来控制 UARTAn 的串行传输操作。

该寄存器可以按 8 位或 1 位为单位进行读写。

复位后该寄存器设置为 10H。

(1/2)

复位后: 10H R/W 地址: UA0CTL0 FFFFFFFA00H, UA1CTL0 FFFFFFFA10H

	<7>	<6>	<5>	<4>	3	2	1	0
UAnCTL0	UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL

(n = 0, 1)

UAnPWR	UARTAn 操作控制
0	禁止 UARTAn 操作 (UARTAn 异步复位)
1	允许 UARTAn 操作
UARTAn 的操作 UAnPWR 位控制. 将 UAnPWR 位清 0, TXDAn 输出固定为高电平 (UAnOPT0.UAnTDL 位 = 1 时固定为低电平).	

UAnTXE	发送操作使能
0	禁止发送
1	允许发送

<R>

- 设置 UAnPWR 位为 1 然后设置 UAnTXE 位 1, 启动发送.
- 为了初始化发送单元, 清除 UAnTXE 位为 0, 等待两个基础时钟 (fucx) 周期, 然后再次设置 UAnTXE 位为 1. 否则, 初始化可能不执行. (关于基础时钟, 见 12.6 (1) (a) 基础时钟).
- 当操作允许 (UAnPWR 位 = 1) 时, 发送操作从 UAnTXE = 1 起经过 2 个以上基础时钟周期之后被允许操作.
- 当 UAnPWR 位清除为 0, 即使 UAnTXE 位为 1, 内部电路状态与 UAnTXE 位 = 0 时状态相同, 当 UAnPWR 位再次设为 1 时, 允许发送.

UAnRXE	接收操作使能
0	禁止接收
1	允许接收

<R>

<R>

- 设置 UAnPWR 位为 1 然后设置 UAnRXE 位 1, 启动接收.
- 为了初始化接收单元, 清除 UAnRXE 位为 0, 等待两个基础时钟 (fucx) 周期, 然后再次设置 UAnRXE 位为 1. 否则, 初始化可能不执行. (关于基础时钟, 见 12.6 (1) (a) 基础时钟).
- 当操作允许 (UAnPWR 位 = 1) 时, 接收操作从 UAnRXE = 1 起经过 2 个以上基础时钟周期之后被允许操作. 当接收允许前收到起始位, 则被忽略.
- 当 UAnPWR 位清除为 0, 即使 UAnRXE 位为 1, 内部电路状态与 UAnTXE 位 = 0 时状态相同, 当 UAnPWR 位再次设为 1 时, 允许接收.

UAnDIR ^注	传输方向选择
0	MSB-先行传输
1	LSB-先行传输

UAnPSI ^注	UAnPSJ ^注	发送时选择校验位	接收时选择校验位
0	0	不输出校验位	无校验位接收
0	1	0 校验位输出	0校验位接收
1	0	奇校验位输出	奇校验检测
1	1	偶校验位输出	偶校验检测

如果接收时选择“使用0校验位接收”,不执行检验检测。
所以,既然UAnSTRUAnPE位不置位,就不输出校验错误中断。

UAnCL ^注	指定发送/接收数据帧字符长度
0	7 位
1	8 位

UAnSL ^注	指定发送数据的停止位长度
0	1 位
1	2 位

不管UAnSL位取值如何,只有第一个接收数据的停止位被检测。

注 仅当 UAnPWR 位 = 0 或 UAnTXE 位 = UAnRXE 位 = 0 时, 该寄存器才能被改写。然而, 可以同时设置 UAnPWR、UAnTXE 和 UAnRXE 位部分或全部为 1。

备注 关于奇偶检验的细节, 参见 12.5.6 奇偶校验类型和操作。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

关于细节, 参见 12.6 (2) UARTAn 控制寄存器 1 (UAnCTL1)。

(3) UARTAn 控制寄存器 2 (UAnCTL2)

关于细节, 参见 12.6 (3) UARTAn 控制寄存器 2 (UAnCTL2)。

(4) UARTAn 选项控制寄存器 0 (UAnOPT0)

UAnOPT0 寄存器为 8 位寄存器，用来控制 UARTAn 的串行传输操作。
 该寄存器可以按 8 位或 1 位为单位进行读写。
 复位后该寄存器设置为 14H。

复位后: 14H R/W 地址: UA0OPT0 FFFFA03H, UA1OPT0 FFFFA13H

	7	6	5	4	3	2	1	0
UAnOPT0 (n = 0, 1)	0	0	0	1	0	1	UAnTDL	UAnRDL

UAnTDL	发送数据电平位
0	传输数据正常输出
1	传输数据输出翻转

- TXDAn 引脚输出电平可以利用 UAnTDL 位进行翻转。
- UAnCTL0.UAnPWR 位 = 0 或 UAnCTL0.UAnTXE 位 = 0 时可设置本寄存器。

UAnRDL	接收数据电平位
0	传输数据正常输入
1	传输数据输入翻转

- RXDAn 引脚输入电平可以利用 UAnRDL 位进行翻转。
- UAnPWR 位 = 0 或 the UAnCTL0.UAnRXE 位 = 0 时可设置本寄存器。
- 当 UAnRDL 位设为 1 (输入数据翻转) 时, 必须在接收开始时设定数据接收引脚为 UART 接收引脚之后使能接收 (UAnRXE bit = 1)。若引脚模式在接收使能后进行改变, 当引脚电平为高时, 会错误检测起始位。

注意事项 务必将位 3 和 5 清除为 0，而将位 2 和 4 置为 1，使用其它的设置时，操作不能保证。

(5) UARTAn 状态寄存器 (UAnSTR)

UAnSTR 寄存器为 8 位寄存器，显示 UARTAn 的传输状态和接收错误的内容。
 该寄存器可以按 8 位或 1 位为单位进行读写，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 清除，它们不能通过写 1 来置位(即便写入 1，其值仍保持不变)。
 初始状态如下所示：

寄存器/位	初始状态
UAnSTR 寄存器	<ul style="list-style-type: none"> • 复位后 • UAnCTL0.UAnPWR 位 = 0
UAnTSF 位	<ul style="list-style-type: none"> • UAnCTL0.UAnTXE 位 = 0
UAnPE, UAnFE, UAnOVE 位	<ul style="list-style-type: none"> • 0 写入 • UAnCTL0.UAnRXE 位 = 0

注意事项 务必读取并检查 UAnPE、UAnFE 和 UAnOVE 位的错误标志，并通过写入‘0’清除。

复位后: 00H R/W 地址: UA0STR FFFFFFFA04H, UA1STR FFFFFFFA14H

	<7>	6	5	4	3	<2>	<1>	<0>
UAnSTR (n = 0, 1)	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

UAnTSF	传输状态标志
0	<ul style="list-style-type: none"> 当已设置UAnPWR位 = 0 或 UAnTXE位 = 0. 当传输结束后UAnTX内没有下一个传输数据.
1	写入UAnTXB位
<p>当进行连续发送时, UAnTSF位总是1. 当初始化发送单元时, 事前检查UAnTSF位= 0. 当UAnTSF位= 1时进行初始化, 传输数据得不到保证.</p>	

UAnPE	校验错误标志
0	<ul style="list-style-type: none"> 当UAnPWR位 = 0 或 UAnRXE位 = 0 已设置. 当0已写入.
1	当接收中数据校验和校验位不匹配.
<ul style="list-style-type: none"> UAnPE位的操作由设置 UAnCTL0.UAnPS1 和UAnCTL0.UAnPS0 位控制. UAnPE位可读可写, 但只能通过写入0进行清除, 而不能写入1进行置位. 当写入1时, 原值保留. 	

UAnFE	帧错误标志
0	<ul style="list-style-type: none"> 当 UAnPWR位 = 0 或 UAnRXE位 = 0 已设置. 当0已写入
1	接收之中未检测到停止位
<ul style="list-style-type: none"> 不管UAnCTL0.UAnSL位取值如何, 只检测接收数据中的第一个停止位. UAnFE位可读可写, 但只能通过写入0进行清除, 而不能写入1进行置位. 当写入1时, 原值保留. 	

UAnOVE	超限错误标志
0	<ul style="list-style-type: none"> 当UAnPWR位 = 0 或 UAnRXE位 = 0 已设置. 当0已写入
1	当接收数据已置入UAnRXB寄存器而于读取接收数据之前下一次接收操作就已经结束.
<ul style="list-style-type: none"> 发生超限错误时, 数据被放弃, 下一接收数据不写入UAnRX寄存器. UAnOVE位可读可写, 但只能通过写入0进行清除, 而不能写入1进行置位. 当写入1时, 原值保留. 	

(6) UARTAn 接收数据寄存器 (UAnRX)

UAnRX 寄存器为 8 位缓存寄存器，用来保存由 UARTAn 接收移位寄存器的数据转换得到的并行数据。

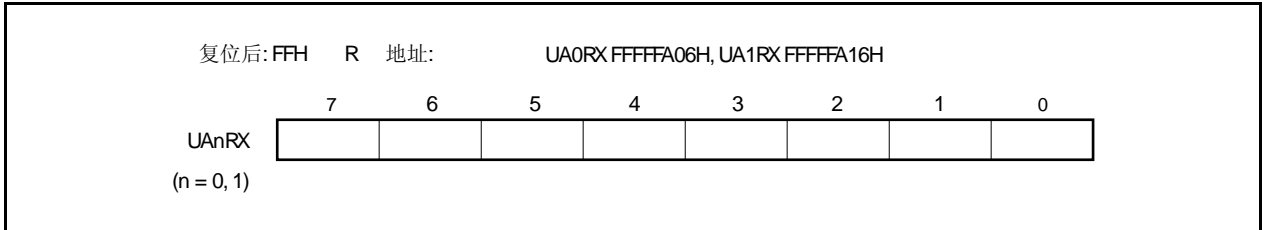
在 1 字节数据接收结束时，存储于 UARTAn 接收移位寄存器的数据传输至 UAnRX 寄存器。此时，产生一个接收结束中断请求信号(INTUAnR)。

当数据长度指定为 7 位，LSB-先行模式接收当中，接收数据传输至 UAnRX 寄存器的位 6 至位 0，而 MSB 总是为 0。MSB-先行模式接收当中，接收数据传输至 UAnRX 寄存器的位 7 至位 1，而 LSB 总是为 0。

当产生溢出错误时 (UAnSTR.UAnOVE 位 = 1)，此时接收的数据不传输至 UAnRX 寄存器中，并被放弃。

该寄存器为 8 位只读寄存器。

除复位之外，UAnRX 寄存器可以通过清除 UAnCTL0.UAnPWR 位为 0 从而设置为 FFH。



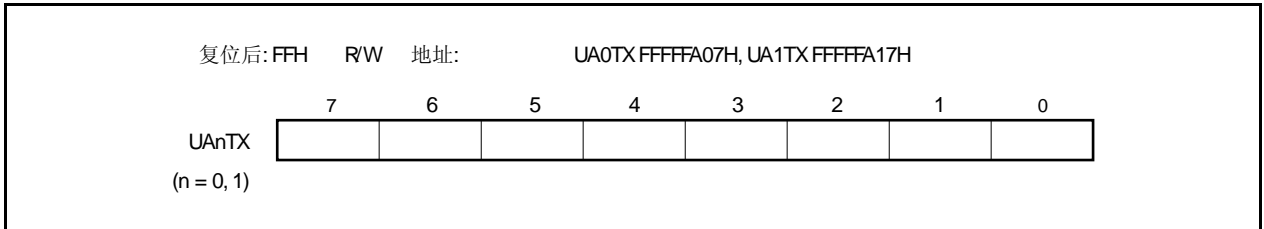
(7) UARTAn 发送数据寄存器 (UAnTX)

UAnTX 寄存器为 8 位寄存器，用来设置发送数据。

处于发送允许状态(UAnCTL0.UAnTXE 位 = 1)时，当发送数据写入 UAnTX 寄存器之时启动发送。当 UAnTX 寄存器的数据已经传输至 UARTAn 发送移位寄存器，产生发送允许中断请求信号 (INTUAnT)。

该寄存器可以按 8 位或 1 位为单位进行读写。

复位后该寄存器设置为 FFH。



12.4 中断请求信号

UARTAn 产生以下 3 种中断请求信号：

- 接收错误中断请求信号 (INTUAnRE)
- 接收结束中断请求信号 (INTUAnR)
- 发送允许中断请求信号 (INTUAnT)

在这三种中断请求信号之中，接收错误中断请求信号具有最高优先级，而接收结束中断请求信号和发送允许中断请求信号依序优先级较低。

表 12-2. 中断与其默认优先级

中断	优先级
接收错误	高
接收结束	↓
发送允许	低

(1) 接收错误中断请求信号 (INTUAnRE)

在接收允许时，接收错误中断请求信号由 3 种接收错误(校验错误、帧错误以及溢出错误)进行或运算产生，在 UAnSTR 寄存器一节予以说明。

(2) 接收结束中断请求信号 (INTUAnR)

接收允许状态下，当数据移位至 UARTAn 接收移位寄存器并且传输至 UAnRX 寄存器时，会输出接收结束中断请求信号。

在接收禁止状态下，不产生接收结束中断请求信号。

(3) 发送允许中断请求信号 (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，产生发送允许中断请求信号。

12.5 操作

12.5.1 数据格式

执行全双工串行数据接收和发送。

如图 12-2 所示，发送/接收数据帧的数据由起始位、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-先行传输方式。

另外，使用 UAnOPT0.UAnTDL 位实现对 UARTAn 的 TXDAn 位输出/反向的输出控制。

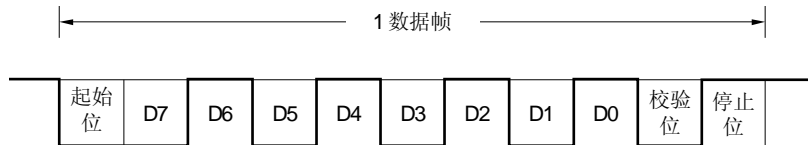
- 起始位 1 位
- 字符位 7 位/8 位
- 校验位 奇检验/偶校验/0 校验/无校验
- 停止位 1 位/2 位

图 12-2. UARTA 发送/接收数据格式

(a) 8 位数据长度，LSB 先行，偶校验，一个停止位，传输数据：55H



(b) 8 位数据长度，MSB 先行，偶校验，一个停止位，传输数据：



(c) 8 位数据长度，MSB 先行，偶校验，一个停止位，传输数据：55H，TXDAn 反向



(d) 7 位数据长度，LSB 先行，奇校验，2 个停止位，传输数据：36H



(e) 8 位数据长度，LSB 先行，无校验，一个停止位，传输数据：87H



12.5.2 UART 发送

通过设置 UAnCTL0.UAnPWR 位为 1，使 TXDAn 引脚输出高电平。

之后，通过设置 UAnCTL0.UAnTXE 位为 1 设定发送允许状态，并且，通过写入发送数据至 UAnTX 寄存器启动发送。起始位、校验位以及停止位自动添加。

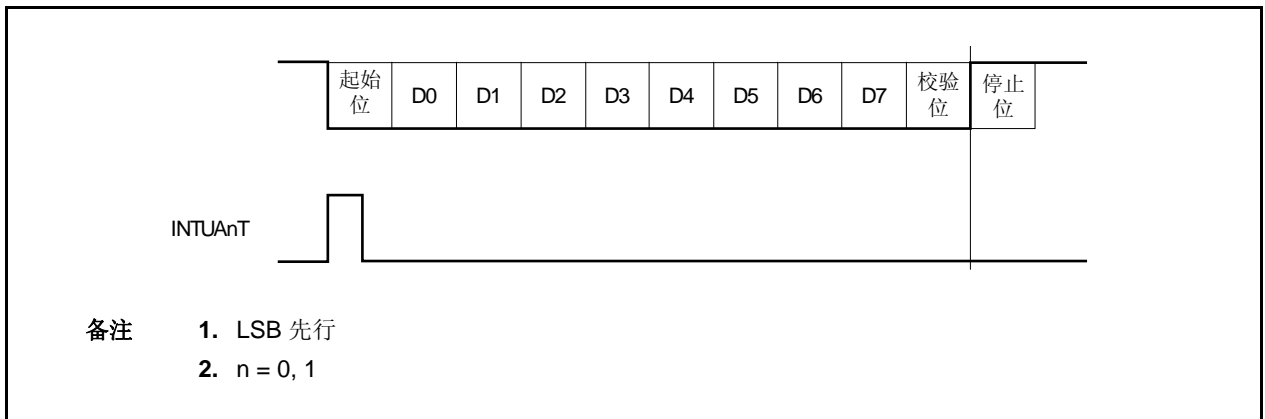
由于 UARTAn 中不提供 CTS (发送允许信号) 输入引脚，故而使用一个端口检查发送目的方的接收允许信号。

当发送操作启动之时，UAnTX 寄存器内的数据传输至 UARTAn 发送移位寄存器。

当 UAnTX 寄存器内的数据发送到 UARTAn 发送移位寄存器结束之时，产生发送允许中断请求信号 (INTUAnT)，且在此之后 UARTAn 发送移位寄存器的内容从 TXDAn 引脚输出。

通过产生 INTUAnT 信号，允许向 UAnTX 寄存器写入下一个发送数据。

图 12-3. UART 发送



12.5.3 连续发送过程

当 UARTAn 发送移位寄存器启动移位操作时，UARTAn 可以写下一个数据至 UAnTX 寄存器，可以由发送允许中断请求信号 (INTUAnT)来判断 UARTAn 发送移位寄存器的发送时序。

通信速率的效能依赖于在传输中将下一个待发送数据写入 UAnTX 寄存器。

注意事项 连续发送执行过程之中，检查到 UAnSTR.UAnTSF 位为 0 后进行初始化。而在 UAnTSF 为 1 时进行初始化发送数据不能保证。

备注 n = 0, 1

图 12-4. 连续发送处理流程

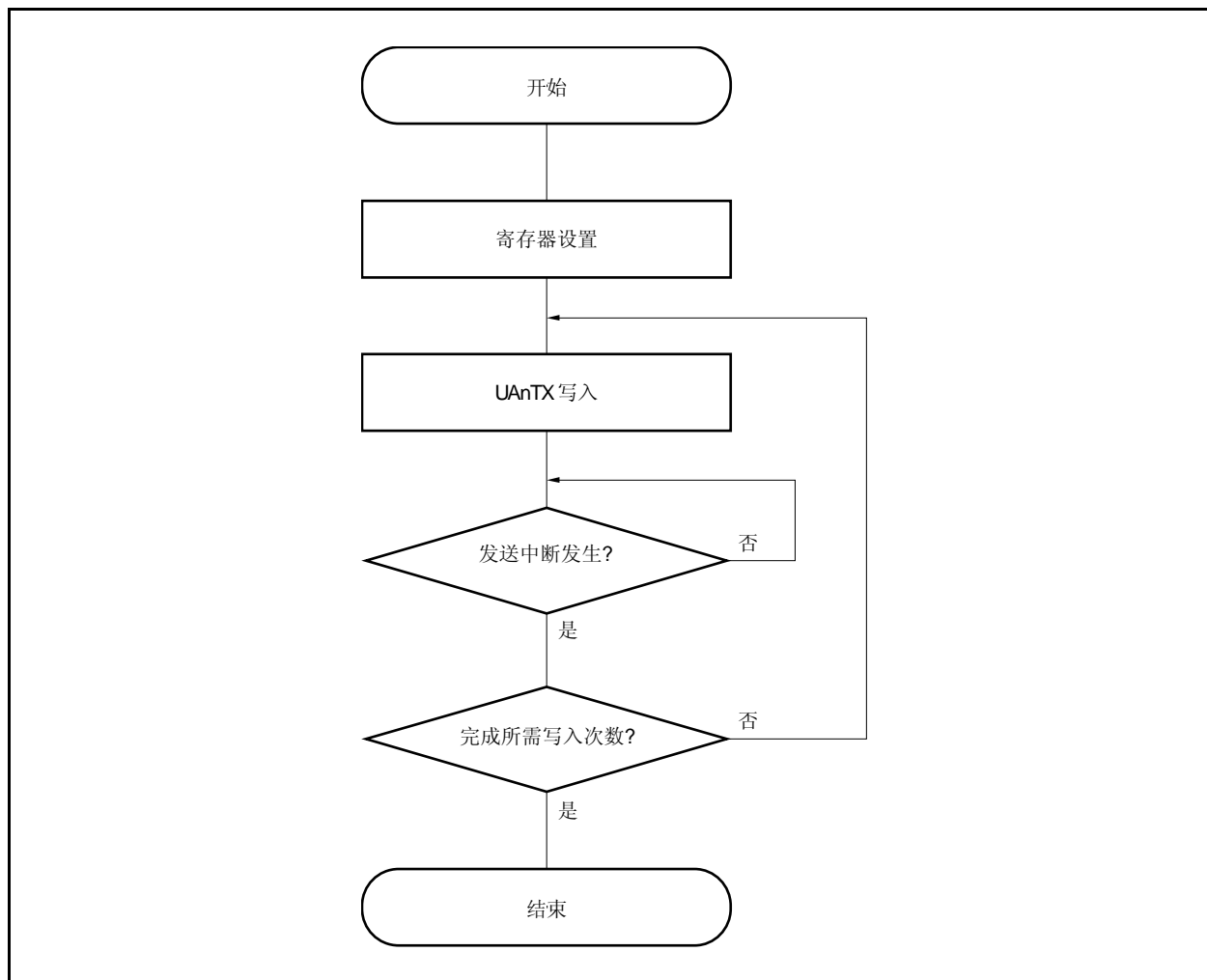
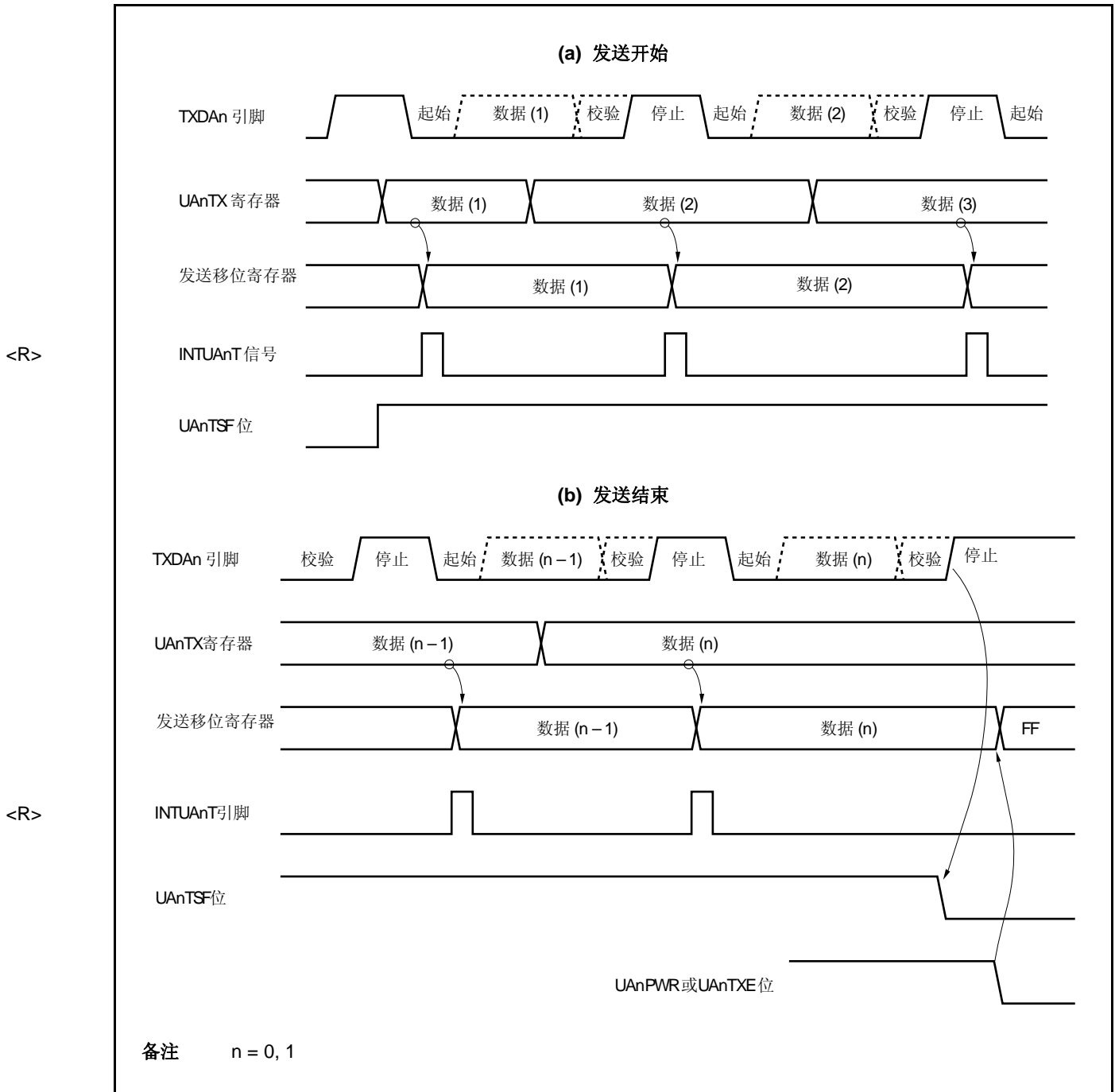


图 12-5. 连续发送操作时序



12.5.4 UART 接收

通过设置 UAnCTL0.UAnPWR 位为 1 并随之设置 UAnCTL0.UAnRXE 位为 1 来设置接收等待状态。在接收等待状态下，执行 RXDAn 引脚监控以及起始位检测。

使用两步检测程序进行起始位检测。

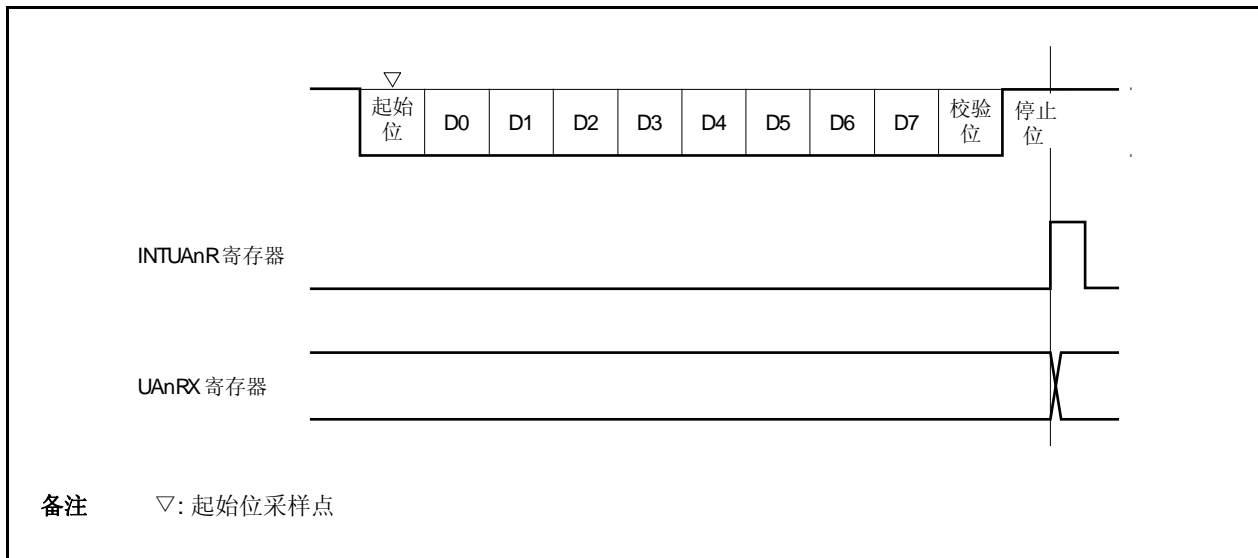
首先，检测 RXDAn 引脚的下降沿并在下降沿开始采样。如果 RXDAn 引脚在起始位采样点上为低电平，确认为起始位。起始位得到确认以后，开始接收操作，按照设定的波特率，串行数据被存入 UARTAn 接收移位寄存器。

接收停止位中，当接收结束中断请求信号 (INTUAnR) 输出时，UARTAn 接收移位寄存器的数据写入 UAnRX 寄存器。然而，如果发生溢出错误(UAnSTR.UAnOVE 位= 1)，此时，接收数据不写入 UAnRX 寄存器，而被丢弃。

即使在接收中发生奇偶校验错误(UAnSTR.UAnPE 位= 1) 或帧错误(UAnSTR.UAnFE 位= 1)，接收仍进行直到收到第一个停止位位置为止，并随着接收结束输出 INTUAnRE 信号。

备注 n = 0, 1

图 12-6. UART 接收



- 注意事项
1. 即使发生接收错误，也要确保读取 UAnRX 寄存器。如果未读取 UAnRX 寄存器，接收下一个数据时将发生溢出错误，并且连续不定地发生接收错误。
 2. 执行接收时的操作假定只存在一个停止位，第二个停止位将被忽略。
 3. 当接收完毕后，于接收结束中断请求信号 (INTUAnR) 已经产生后读取 UAnRX 寄存器，并将 UAnPWR 或 UAnRXE 位清 0。如果 UAnPWR 或 UAnRXE 位在 INTUAnR 信号产生之前清 0，不能保证对 UAnRX 寄存器值的正确读取。
 4. 如果 UARTAn 的接收结束处理(INTUAnR 信号产生) 和 UAnPWR 位= 0 或 UAnRXE 位 = 0 相冲突，则即使 UAnRX 寄存器内没有存储数据，INTUAnR 信号也可能产生。为了结束接收而不等待 INTUAnR 信号产生，务必将中断请求标志 (UAnRIC.UAnRIF) 清 0，将中断屏蔽标志 (UAnRIC.UAnRMK) 设置为 1，并随之设置 UAnPWR 位= 0 或 UAnRXE 位 = 0。

12.5.5 接收错误

接收过程中的错误为 3 类：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设置在 UAnSTR 寄存器内，当错误发生时，输出接收错误中断请求信号 (INTUAnRE)。

通过读取 UAnSTR 寄存器的内容，可以确定在接收过程中发生了那种错误。

读取完毕后，写入 0 来清除接收错误标志。

注意事项 接收结束中断请求信号 (INTUAnR)和接收错误中断请求信号 (INTUAnRE)不同时产生。INTUAnR 信号在接收正常结束时产生，在接收错误时，产生 INTUAnRE 信号，而不产生 INTUAnR 信号。

备注 n = 0, 1

- 接收错误原因

错误标志	接收错误	原因
UAnPE	奇偶检验错误	接收的校验位和设定的不匹配。
UAnFE	帧错误	未检测到停止位。
UAnOVE	溢出错误	从 UAnRX 寄存器读取数据前完成下一个数据接收。

12.5.6 检验类型和操作

奇偶校验位用来检测通信数据中的位错误。通常情况下，发送端和接收端应用相同的检验方式。

在使用奇校验和偶校验情况下，可以检测到奇数个位错误。而在 0 奇偶校验和无校验情况下，不能检测到错误。

(a) 偶校验

(i) 发送过程中

发送数据之中值为 1 的比特数，包括上校验位，控制其数目为偶数。奇偶校验位的值计算如下：

- 发送数据中值为“1”的比特数目为奇数： 1
- 发送数据中值为“1”的比特数目为偶数： 0

(ii) 接收过程中

发送数据中值为“1”的比特数，包括校验位一并计算，如果数目为奇数，则输出奇偶校验错误。

(b) 奇校验

(i) 发送过程中

与偶校验相对，发送数据之中值为 1 的比特数，包括上校验位，控制其数目为奇数。奇偶校验位的值计算如下：

- 发送数据中值为“1”的比特数目为奇数： 0
- 发送数据中值为“1”的比特数目为偶数： 1

(ii) 接收过程中

发送数据中值为“1”的比特数，包括校验位一并计算，如果数目为偶数，则输出奇偶校验错误。

(c) 0 奇偶校验

发送过程中，不管传输数据怎样，奇偶校验位总是为 0。

接收过程中，不进行奇偶校验检查，所以，不管奇偶校验位是 0 还是 1，都不发生奇偶校验错误。

(d) 无校验

发送数据上不附加奇偶校验位。

接收时认为没有奇偶校验位，既然没有奇偶校验位，所以，也不发生奇偶校验错误。

12.5.7 接收数据噪声滤波器

该滤波器使用预分频器输出的基准时钟(f_{CLK})采样 RXDAn 引脚。

如果相同的采样值被读到两次，则匹配检测器输出改变，并且，RXDAn 信号被采样为输入数据。所以，不超过两个时钟宽度的数据被判定为噪声，而不会提供给内部电路(参见图 12-8)。有关基准时钟，参见 12.6 (1) (a) 基准时钟。

另外，由于电路如图 12-7 所示，所以，接收操作的内部处理过程与外部信号状态相比，延迟了 3 个时钟周期。

备注 n = 0, 1

图 12-7. 噪声滤波器电路

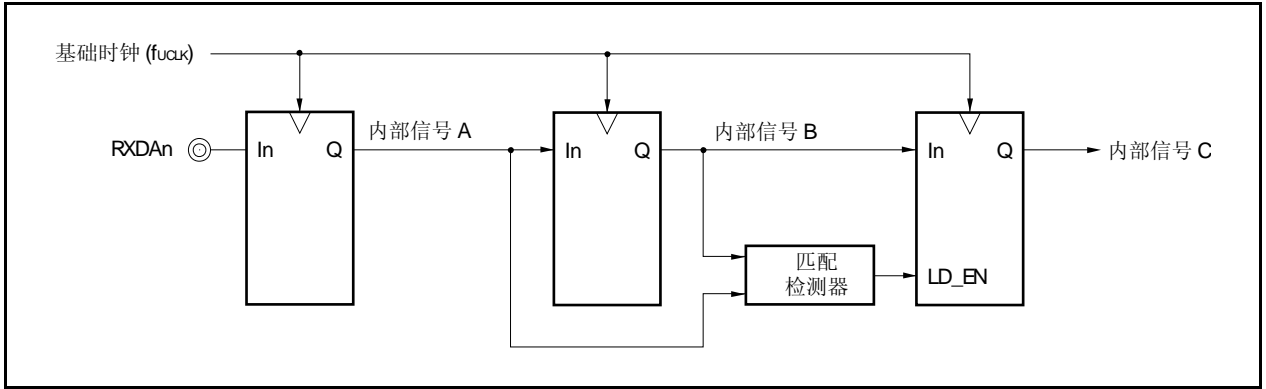
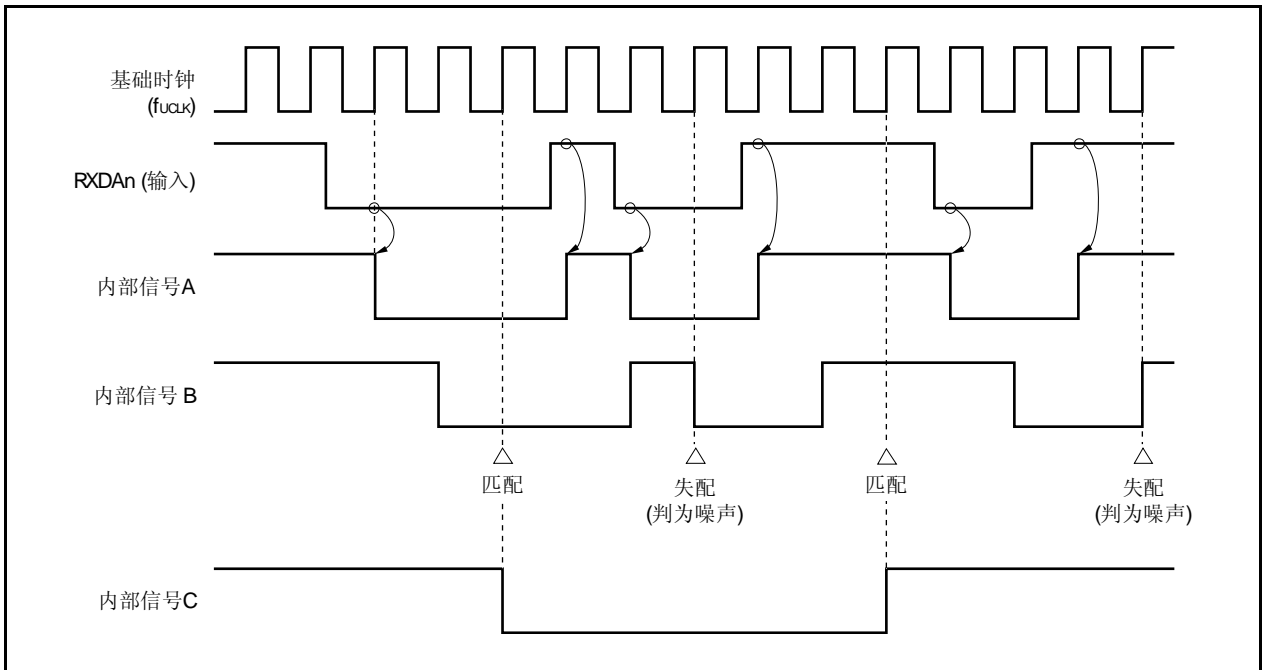


图 12-8. RXDAn 信号被判决为噪声的时序



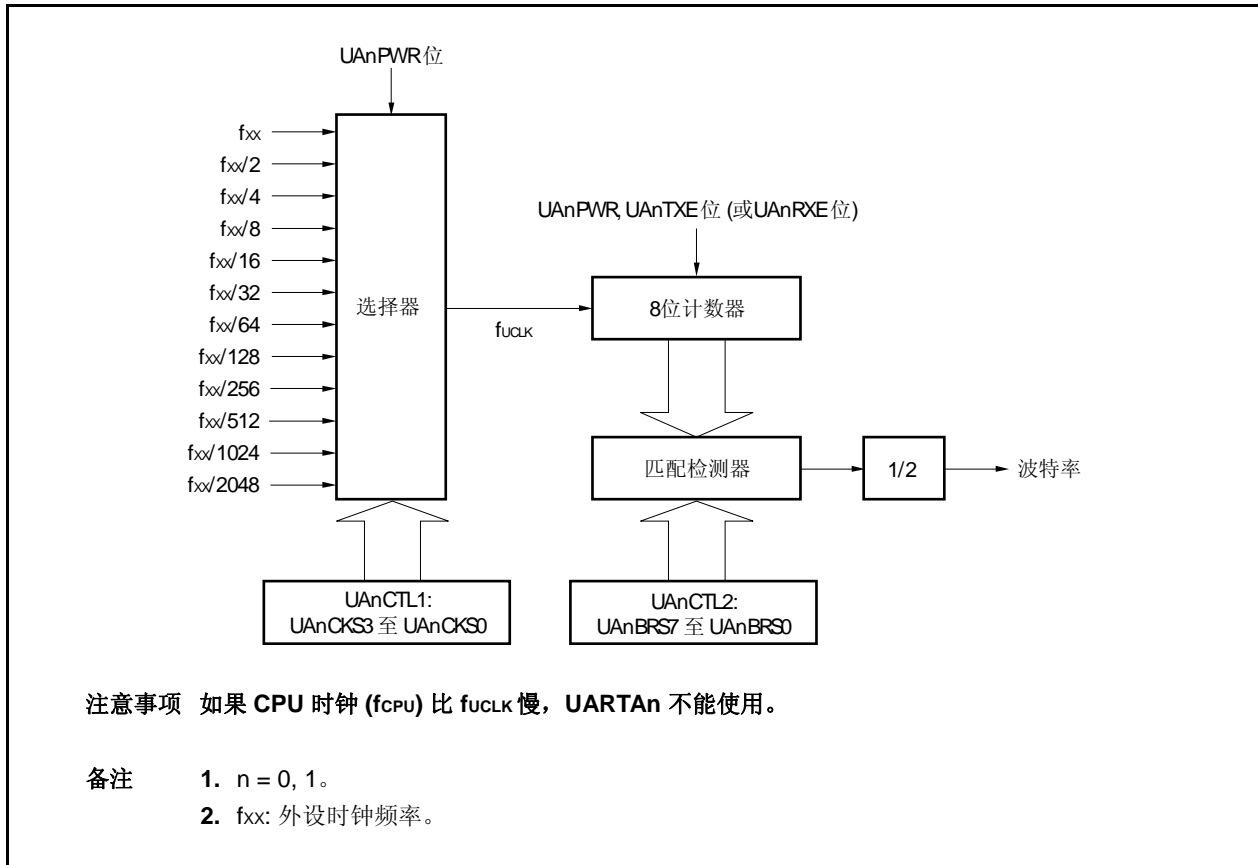
12.6 专用波特率发生器

专用波特率发生器由时钟源选择器模块和一个 8 位可编程计数器构成，它在使用 UARTAn 进行发送和接收时产生一个串行时钟。专用波特率发生器的输出可以为每一个通道选择串行时钟。

一个 8 位计数器用于发送，另一个用于接收。

(1) 波特率发生器的配置

图 12-9. 波特率发生器的配置



(a) 基准时钟

当 UAnCTL0.UAnPWR 位为 1 时，由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位所选择的时钟提供给 8 位计数器。本时钟称之为基准时钟(f_{uCLK})。当 UAnPWR 位 = 0， f_{uCLK} 固定为低电平。

(b) 串行时钟的产生

通过设置 UAnCTL1 寄存器和 UAnCTL2 寄存器来产生一个串行时钟。

基准时钟(f_{uCLK}) 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) UARTAn 控制寄存器 1 (UAnCTL1)

UAnCTL1 寄存器为 8 位寄存器，用来选择 UARTAn 基准时钟。

该寄存器可以按 8 位进行读写。

复位后该寄存器设置为 00H。

注意事项 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							
			7	6	5	4	3	2	1	0
UAnCTL1			0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0
(n = 0, 1)										
			UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	基础时钟 (f _{clk}) 选择			
			0	0	0	0	f _{clk}			
			0	0	0	1	f _{clk} /2			
			0	0	1	0	f _{clk} /4			
			0	0	1	1	f _{clk} /8			
			0	1	0	0	f _{clk} /16			
			0	1	0	1	f _{clk} /32			
			0	1	1	0	f _{clk} /64			
			0	1	1	1	f _{clk} /128			
			1	0	0	0	f _{clk} /256			
			1	0	0	1	f _{clk} /512			
			1	0	1	0	f _{clk} /1,024			
			1	0	1	1	f _{clk} /2,048			
			其它				禁止设置			

备注 f_{clk}: 外设时钟频率

(3) UARTAn 控制寄存器 2 (UAnCTL2)

UAnCTL2 寄存器为 8 位寄存器，用来选择 UARTAn 的波特率（串行传输速率）时钟。

该寄存器可以按 8 位进行读写。

复位后该寄存器设置为 FFH。

注意事项 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 00。

复位后: FFH R/W 地址: UA0CTL2 FFFFFFFA02H, UA1CTL2 FFFFFFFA12H

	7	6	5	4	3	2	1	0
UAnCTL2	UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0

(n = 0, 1)

UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0	默认 (k)	串行时钟
0	0	0	0	0	0	'	'	'	禁止设置
0	0	0	0	0	1	0	0	4	f _{CLK} /4
0	0	0	0	0	1	0	1	5	f _{CLK} /5
0	0	0	0	0	1	1	0	6	f _{CLK} /6
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	f _{CLK} /252
1	1	1	1	1	1	0	1	253	f _{CLK} /253
1	1	1	1	1	1	1	0	254	f _{CLK} /254
1	1	1	1	1	1	1	1	255	f _{CLK} /255

备注 f_{CLK}: 由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择的基准时钟频率

(4) 波特率

波特率由以下算式计算得到:

$$\text{波特率} = \frac{f_{\text{CLK}}}{2 \times k} \text{ [bps]}$$

f_{CLK} : 由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择的基准时钟频率

k: 用 UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位所设定的值(k = 4, 5, 6, ..., 255)

(5) 波特率误差

波特率误差可以由以下算式计算得到:

$$\text{误差 (\%)} = \left[\frac{\text{实际波特率 (误差波特率)}}{\text{目标波特率 (正确波特率)}} - 1 \right] \times 100 \text{ [\%]}$$

- 注意事项**
1. 发送过程中, 波特率误差必须限于接收端的误差容限之内。
 2. 接收时的波特率误差必须满足由一定范围, 该范围在‘段落(7)接收中允许的波特率误差范围’中指明。

举例

外设时钟频率 = 20 MHz = 20,000,000 Hz

UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位的设定值 = 0000B ($f_{\text{CLK}} = 20,000,000$ Hz)

UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位的设定值 = 01000001B (k = 65)

目标波特率 = 153,600

$$\text{波特率} = 20,000,000 / (2 \times 65) = 153,846 \text{ [bps]}$$

$$\begin{aligned} \text{误差} &= (153,846 / 153,600 - 1) \times 100 \\ &= 0.160 \text{ [\%]} \end{aligned}$$

(6) 波特率设置举例

表 12-3. 波特率发生器设置数据

波特率 (bps)	f _{xx} = 20 MHz		
	UAnCTL1	UAnCTL2	ERR (%)
300	08H	82H	0.16
600	07H	82H	0.16
1,200	06H	82H	0.16
2,400	05H	82H	0.16
4,800	04H	82H	0.16
9,600	03H	82H	0.16
19,200	02H	82H	0.16
31,250	01H	A0H	0
38,400	01H	82H	0.16
76,800	00H	82H	0.16
153,600	00H	41H	0.16
312,500	00H	20H	0
625,000	00H	10H	0
1,250,000	00H	8H	0

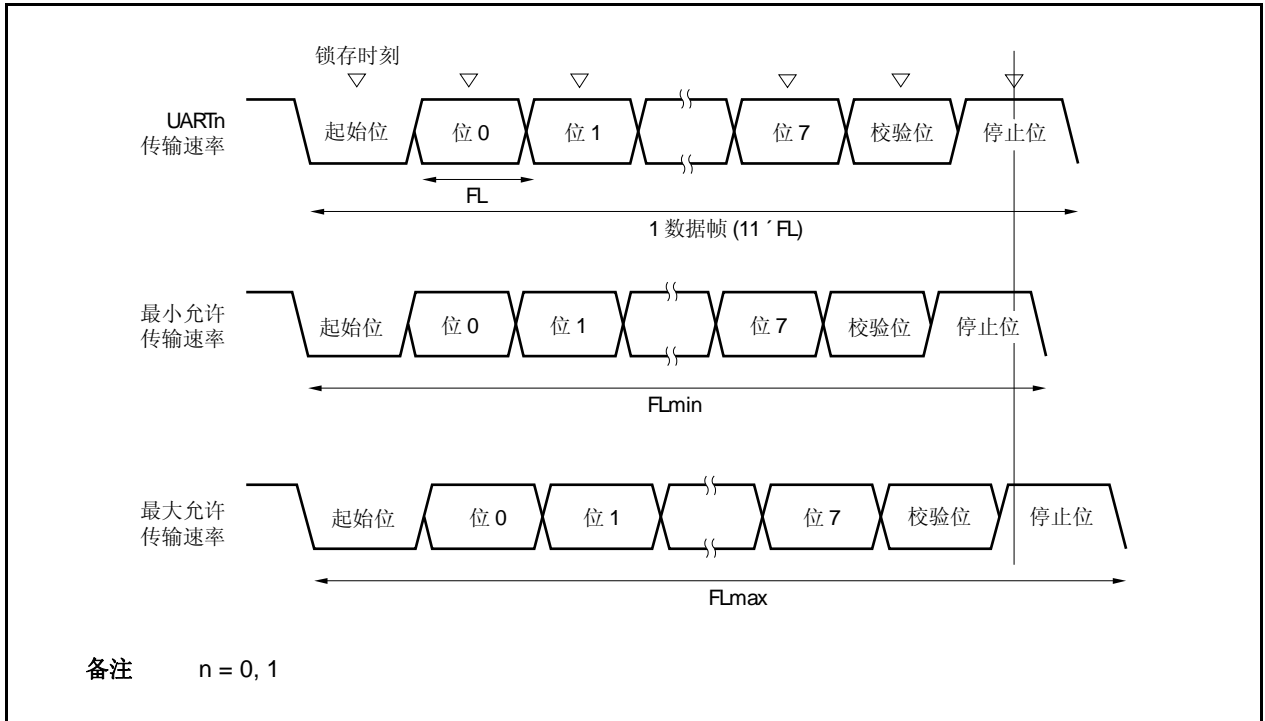
备注 f_{xx}: 外设时钟频率。
ERR: 波特率误差 (%)。

(7) 接收中允许的波特率误差范围

目的端接收中允许的波特率误差范围如下所示：

注意事项 接收时的波特率误差必须使用以下等式设置在允许的范围之内。

图 12-10. 接收中允许的波特率误差范围



如图 12-10 所示，接收数据的锁存时刻取决于检测到起始位后使用 UAnCTL2 寄存器对计数器的设置。如果在此锁存时刻可以一直如期接收到最末数据（停止位），发送数据就可以正常接收。

当应用于 11 位接收时，以下为理论结果：

$$FL = (\text{Brate})^{-1}$$

Brate: UARTAn 波特率 (n = 0, 1)

k: UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位的设定值 (n = 0, 1)

FL: 1-位数据长度

锁存时间宽度: 2 时钟周期

$$\text{最小允许传输速率: } FL_{\min} = 11 \times FL - \frac{21k + 2}{2k} \times FL = \frac{k - 2}{2k} FL$$

所以，可以由目的端接收的最大波特率如下所示：

$$BR_{max} = (FL_{min}/11)^{-1} = \frac{22k}{21k + 2} \text{ Brate}$$

相似地，遵从以下算式计算获得最大许可波特率：

$$\frac{10}{11} \times FL_{max} = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FL_{max} = \frac{21k - 2}{20 k} FL \times 11$$

所以，可以由目的端接收的最小波特率如下所示：

$$BR_{min} = (FL_{max}/11)^{-1} = \frac{20k}{21k - 2} \text{ Brate}$$

由上述等式获得 UARTAn 以及目的端的允许波特率误差，以此遵从以下表格得到最大波特率值。

表 12-4. 最大/最小允许波特率误差

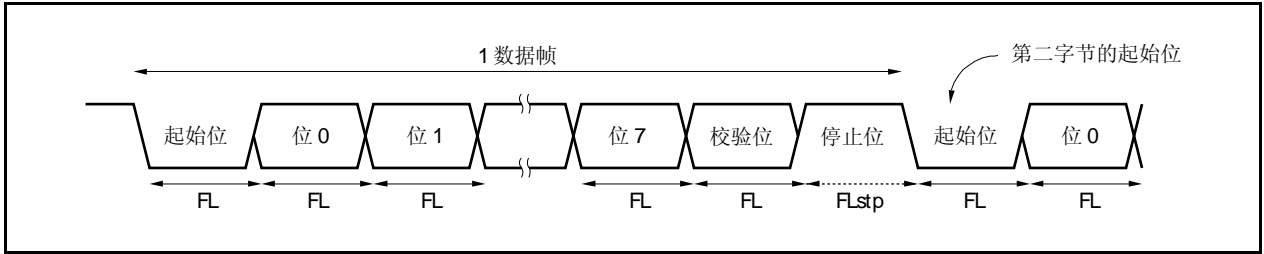
分频率 (k)	最大允许波特率误差	最小允许波特率误差
4	+2.32%	-2.43%
8	+3.52%	-3.61%
20	+4.26%	-4.30%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.72%

- 备注**
1. 接收精度依赖于 1 帧内的比特数、输入时钟频率以及分频率 (k)。输入时钟频率越高、分频率越大，精度就越高。
 2. k: UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位的设定值(n = 0, 1)。

(8) 连续发送时的传输速率

在连续发送之中，从停止位到下一个起始位的转换速度通常为 2 个基准时钟长度。而初始化由收端通过检测起始位进行，故而这对传输结果没有影响。

图 12-11. 连续发送时的转换速度



假设 1 比特数据长度：FL；停止位长度：FLstp；基准时钟频率：f_{UCLK}，我们得到如下等式：

$$FLstp = FL + 2/f_{UCLK}$$

所以，连续发送时的传输速率如下：

$$\text{传输速率} = 11 \times FL + (2/f_{UCLK})$$

12.7 注意事项

当供给 UARTAn 的时钟 停止(如：处于 IDLE1 或 STOP 模式中)，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 引脚也保持并输出时钟供应停止前的瞬间值。然而，时钟恢复之后不能保证操作正常。所以，时钟恢复之后，应该通过设置 UAnCTL0.UAnPWR、UAnCTL0.UAnRXE 和 UAnCTL0.UAnTXE 位为 000，对电路进行初始化。

备注 n = 0, 1

第十三章 3-线可变长度串行 I/O (CSIB)

V850ES/IE2 内部集成 CSIB0。

13.1 特性

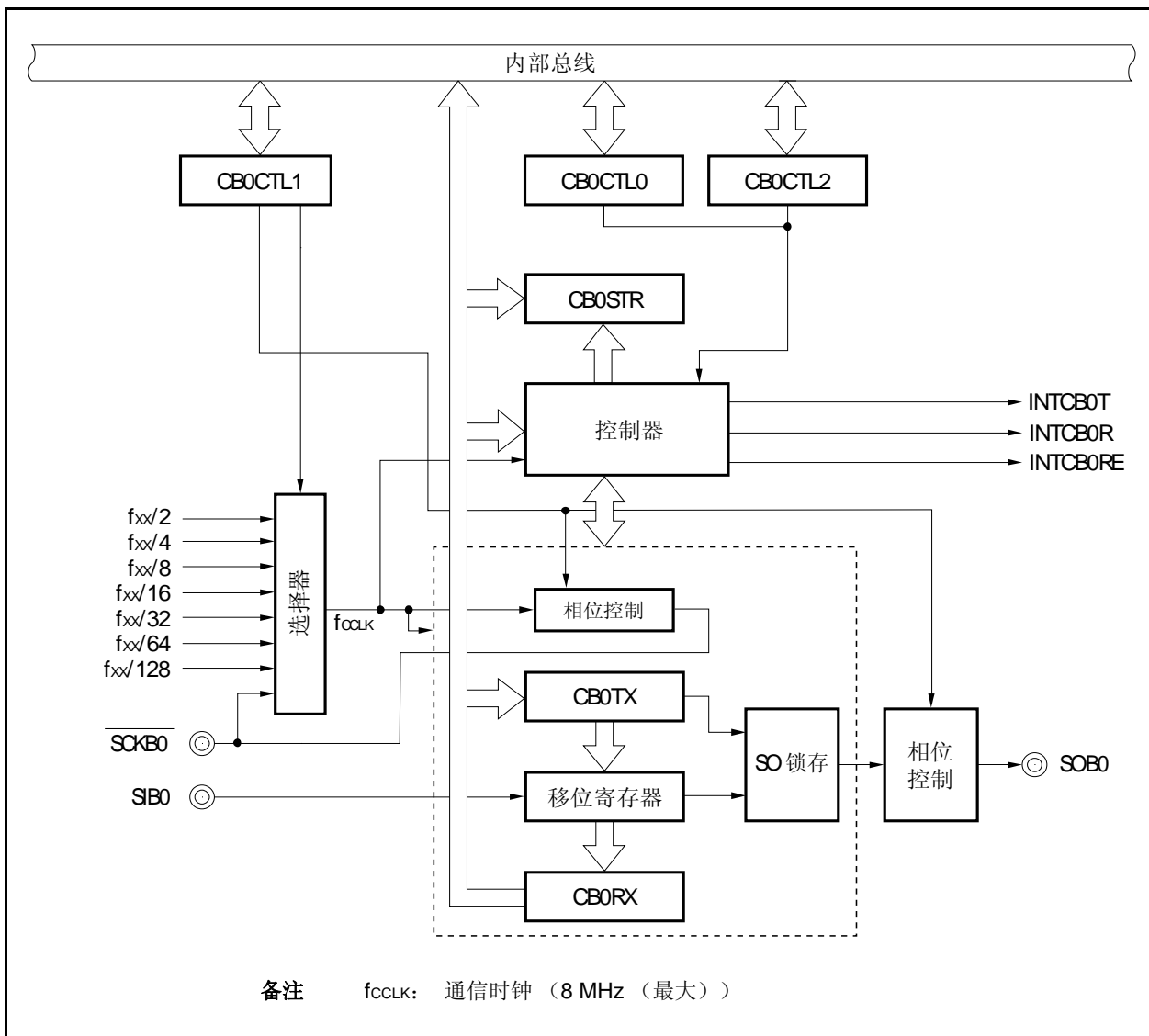
- 传输速率: 8 Mbps (使用内部时钟)
- 主机、从机模式可选
- 8-位至 16-位 传输, 3-线串行接口
- 中断请求信号 (INTCB0RE, INTCB0T, INTCB0R)
- 串行时钟和数据相位可转换
- 可以 1 位为单位, 选择 8 至 16 位数据之间选择数据长度
- 可切换 MSB 先行/LSB 先行传输数据
- 3 线传输
 - SOB0: 串行数据输出
 - SIB0: 串行数据输入
 - $\overline{\text{SCKB0}}$: 串行时钟 I/O

可指定发送模式、接收模式以及发送/接收模式。

13.2 配置

以下所示为 CSIB0 的功能框图。

图 13-1. CSIB0 框图



CSIB0 包括以下硬件。

表 13-1. CSIB0 的配置

项目	配置
寄存器	CSIB0 数据接收寄存器 (CB0RX) CSIB0 数据发送寄存器 (CB0TX)
控制寄存器	CSIB0 控制寄存器 0 (CB0CTL0) CSIB0 控制寄存器 1 (CB0CTL1) CSIB0 控制寄存器 2 (CB0CTL2) CSIB0 状态寄存器 (CB0STR)

(1) CSIB0 数据接收寄存器 (CB0RX)

CB0RX 是一个 16 位缓存寄存器，用来存放接收数据。

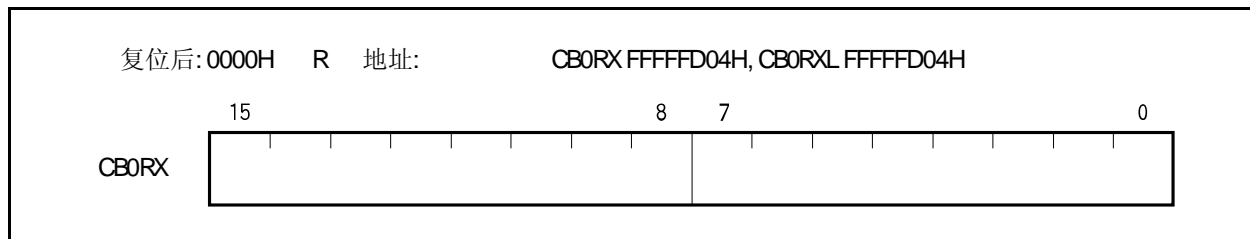
本寄存器为只读寄存器，以 16 位为单位。

在接收允许状态下，通过读取 CB0RX 寄存器开始接收操作。

如果传输数据长度为 8 位，本寄存器的低 8 位作为寄存器 CB0RXL，其只能以 8 位为单位进行读取。

复位后本寄存器被置为 0000H。

除复位之外，CB0RX 寄存器可以通过清除（清为 0）CB0CTL0.CB0PWR 位进行初始化。

**(2) CSIB0 数据发送寄存器 (CB0TX)**

CB0TX 寄存器为 16 位缓存寄存器，用来写入 CSIB0 以发送数据。

本寄存器可以以 16 位为单位进行读出或写入。

在发送允许的情况下，通过向 CB0TX 寄存器写入数据启动发送操作。

如果传输数据长度为 8 位，本寄存器的低 8 位作为寄存器 CB0TXL，其只能以 8 位为单位进行写入。

复位后本寄存器被置为 0000H。



备注 通信开始状态如下所示。

发送模式 (CB0TXE 位 = 1, CB0RXE 位 = 0):

写入 CB0TX 寄存器

发送/接收 (CB0TXE 位 = 1, CB0RXE 位 = 1):

写入 CB0TX 寄存器

接收模式 (CB0TXE 位 = 0, CB0RXE 位 = 1):

读取 CB0RX 寄存器

13.3 控制寄存器

以下寄存器用来控制 CSIB0。

- CSIB0 控制寄存器 0 (CB0CTL0)
- CSIB0 控制寄存器 1 (CB0CTL1)
- CSIB0 控制寄存器 2 (CB0CTL2)
- CSIB0 状态寄存器 (CB0STR)

(1) CSIB0 控制寄存器 0 (CB0CTL0)

CB0CTL0 寄存器控制 CSIB0 串行传输操作。

该寄存器可以以 8 位或 16 位为单位进行读取或写入。

复位后该寄存器被置为 01H。

(1/2)

复位后: 01H	R/W	地址:	FFFFFFD00H					
CB0CTL0	<7>	<6>	<5>	<4>	3	2	1	<0>
	CB0PWR	CB0TXE [‡]	CB0RXE [‡]	CB0DIR [‡]	0	0	CB0TMS [‡]	CB0SCE
	CB0PWR	指定 CSIB0 禁止/允许操作						
	0	禁止 CSIB0 操作并复位 CB0STR 寄存器						
	1	允许 CSIB0 操作						
	• CB0PWR 位 CSIB0 的操作以及复位内部电路							
	CB0TXE [‡]	指定禁止/允许发送操作						
	0	禁止发送操作						
	1	允许发送操作						
	• 当 CB0TXE 位为 0 时，SOB0 输出低电平。							
	CB0RXE [‡]	指定禁止/允许接收操作						
	0	禁止接收操作						
	1	允许接收操作						
	• 当 CB0RXE 位清 0，为了禁止接收操作，即使已经传输了预定数据，仍不会产生接收结束中断，并且接收数据 (CB0RX 寄存器) 不会更新。							
<p>注 当 CB0PWR 位 = 0 时，这些位仅可写入。而重写这些位时，与此同时也可设置 CB0PWR 位 = 1。</p> <p>注意事项 务必将位 3 和位 2 清为“0”。</p>								

CB0DIR ^{注1}	指定产生方向模式 (MSB/LSB)
0	MSB 先行
1	LSB 先行

CB0TMS ^{注1}	指定传输模式
0	单次传输模式
1	连续传输模式

• 当使用单次传输或发送/接收模式，且通信类型2或4时，检查CB0STRCB0TSF位为0后，写入传输数据至CB0TX寄存器。

CB0SCE	指定传输禁止/允许启动传输
0	通信启动触发无效
1	通信启动触发有效

• 主机模式
本位允许或禁止通信启动触发。
(a) 单次接收模式
读取接收数据(CB0RX寄存器)^{注2}之前清除CB0SCE位为0。
(b) 连续接收模式
接收最后数据终止^{注3}前一个通信时钟，将CB0SCE位清0。

• 从机模式
本位允许或禁止通信启动触发。
(a) 单次接收模式或连续接收模式
设置CB0SCE位为1^{注4}。

• 单次发送或发送/接收模式，或者，连续发送或发送接收模式
CB0SCE位的功能无效，建议设置本位为1。

- 注**
1. 这些位仅可当CB0PWR位=0时重写。重写这些位时，也可设置CB0PWR位为1。
 2. 如果当CB0SCE值为1时被读取，则开始下一次通信操作。
 3. 如果CB0SCE位在最后一个数据接收结束之前没有被清0，下一次通信操作自动开始。
若要在读取最后一个数据后再次开始通信，则应将CB0SCE位设为1并对CB0RX寄存器进行伪读取。
 4. 为了启动接收，必需进行一次伪读取。

(a) 如何使用 CB0SCE 位**(i) 在单收模式时**

- <1> 当 INTCB0R 中断服务程序接收完成最后一个数据时，将 CB0SCE 位清为 0，然后读取 CB0RX 寄存器。
- <2> 当完成最后一个数据接收之后禁止接收时，应确保 CB0STR.CB0TSF 位为 0，然后将 CB0PWR 和 CB0RXE 位清为 0。如果继续通信，将 CB0SCE 位设置为 1，并通过对 CB0RX 寄存器执行一次伪读取启动下一次通信操作。

(ii) 在连续接收模式时

- <1> 通过最后接收的前一次 INTCB0R 中断服务在最后数据接收期间将 CB0SCE 位清为 0，然后读取 CB0RX 寄存器。
- <2> 在接收到最后一次 INTCB0R 信号之后，从 CB0RX 寄存器中读取最后的数据。
- <3> 当完成最后一个数据接收之后禁止接收时，检查 CB0STR.CB0TSF 位为 0，然后将 CB0PWR 和 CB0RXE 位清为 0。如果继续通信，将 CB0SCE 位设置为 1，并通过对 CB0RX 寄存器执行一次伪读取启动下一次通信操作。

注意事项 连续接收模式时，当通过伪读取启动接收后，CB0SCE 位清 0 完毕，直到接收执行期间串行时钟不停止。

(2) CSIB0 控制寄存器 1 (CB0CTL1)

CB0CTL1 为 8 位寄存器，控制 CSIB0 串行传输操作。

该寄存器可以以 8 位或 16 位为单位进行读取或写入。

复位后本寄存器被置为 01H。

注意事项 CB0CTL1 寄存器仅当 CB0CTL0.CB0PWR 位=0 是才可以被改写。

复位后: 00H R/W 地址: FFFFFFFD01H

	7	6	5	4	3	2	1	0
CB0CTL1	0	0	0	CB0CKP	CB0DAP	CB0CKS2	CB0CKS1	CB0CKS0

	CB0CKP	CB0DAP	指定与 SCKB0S 相关的数据发送接收时序
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	

CB0CKS2	CB0CKS1	CB0CKS0	通信时钟	模式
0	0	0	$f_{\text{clk}}/2$	主机模式
0	0	1	$f_{\text{clk}}/4$	主机模式
0	1	0	$f_{\text{clk}}/8$	主机模式
0	1	1	$f_{\text{clk}}/16$	主机模式
1	0	0	$f_{\text{clk}}/32$	主机模式
1	0	1	$f_{\text{clk}}/64$	主机模式
1	1	0	$f_{\text{clk}}/128$	主机模式
1	1	1	外部时钟(SCKB0)	从机模式

注意事项 将 f_{cCLK} 设为 8 MHz 或更低。

(3) CSIB0 控制寄存器 2 (CB0CTL2)

CB0CTL2 为一个 8 位寄存器，用来控制 CSIB0 串行传输的比特数目。

该寄存器可以以 8 位为单位进行读取或写入。

复位后本寄存器被置为 01H。

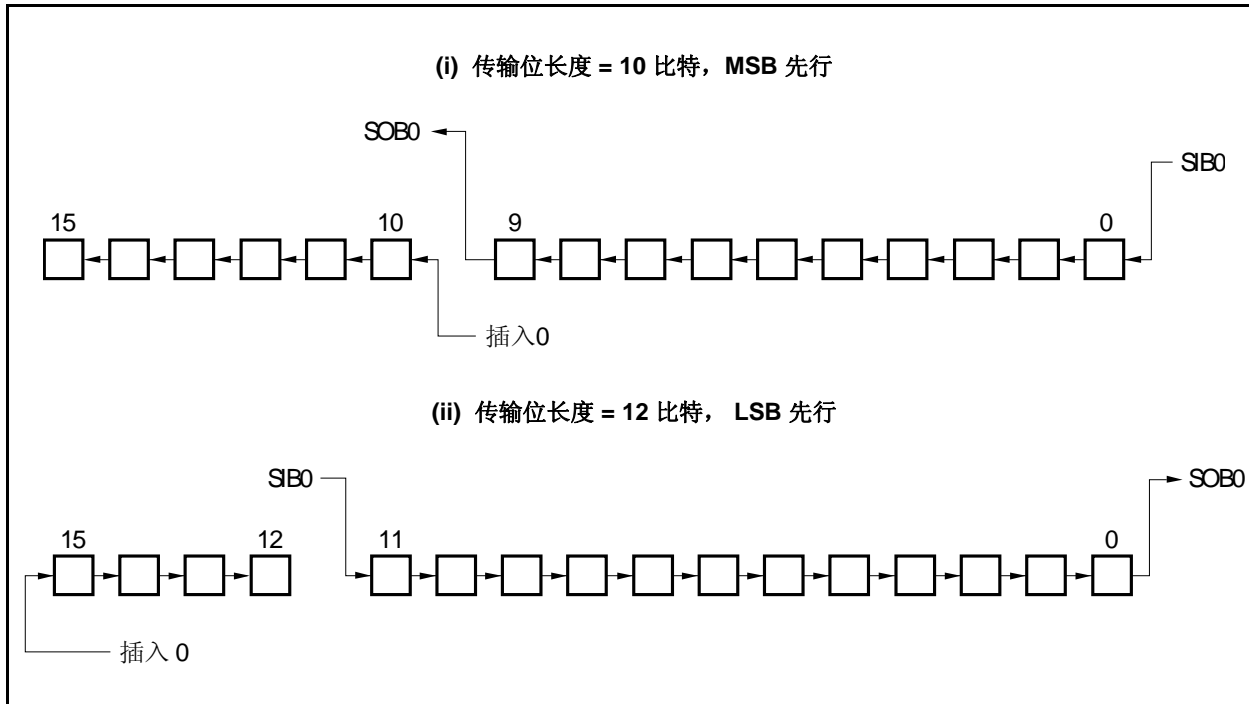
注意事项 CB0CTL2 寄存器仅当 CB0CTL0.CB0PWR 位= 0 或 CB0TXE 和 CB0RXE 位均为 0 时才可以写入。

复位后: 00H	R/W	地址:	FFFFFFD02H					
	7	6	5	4	3	2	1	0
CB0CTL2	0	0	0	0	CB0CL3	CB0CL2	CB0CL1	CB0CL0
	CB0CL3	CB0CL2	CB0CL1	CB0CL0	串行寄存器位长度			
	0	0	0	0	8 位			
	0	0	0	1	9 位			
	0	0	1	0	10 位			
	0	0	1	1	11 位			
	0	1	0	0	12 位			
	0	1	0	1	13 位			
	0	1	1	0	14 位			
	0	1	1	1	15 位			
	1	·	·	·	16 位			
备注	如果传输比特数不是 8 或 16，请准备并使用填充数据，从 CB0TX 和 CB0RX 寄存器的 LSB 起进行填充。							

(a) 传输数据长度改变功能

CSIB0 传输数据长度可以使用 CB0CTL2.CB0CL3 至 CB0CTL2.CB0CL0 位，在 8 位和 16 位之间以 1 位为单位进行设置。

当传输比特长度设置为非 16 位时，将 CB0TX 或 CB0RX 数据设置为由 LSB 开始，不管传输起始位是 MSB 还是 LSB。没有使用的高位可以设置为任意数据，但是，串行传输之后所接收的相应数据变为 0。



(4) CSIB0 状态寄存器 (CB0STR)

CB0STR 为一个 8 位寄存器，用来显示 CSIB0 的状态。

本寄存器可以以 8 位或 16 位为单位进行读取或写入，但是，CB0TSF 标志是只读的。

复位后本寄存器被置为 01H。

除复位之外，CB0STR 可以通过清除 CB0CTL0.CB0PWR 位（为 0）来初始化。

复位后: 00H R/W 地址: FFFFFFFD03H

	<7>	6	5	4	3	2	1	<0>
CB0STR	CB0TSF	0	0	0	0	0	0	CB0OVE

CB0TSF	通信状态标志
0	通信停止
1	通信中
<ul style="list-style-type: none"> 发送过程中,当数据在CB0TX寄存器中准备好时,本寄存器置位,而在接收过程中,当对CB0RX寄存器执行伪写入时,本寄存器置位。当输出结束时,在时钟的最末边沿本标志清0。 	

CB0OVE	超限错误标志
0	未超限
1	超限
<ul style="list-style-type: none"> 接收操作结束时,当不执行对CB0RX的CPU读取就启动下一次接收,则发生超限错误。这种情况下,CB0OVE标志显示超限错误发生状态。 写入0清除CB0OVE标志,即使写入1,也不能使之置位。 	

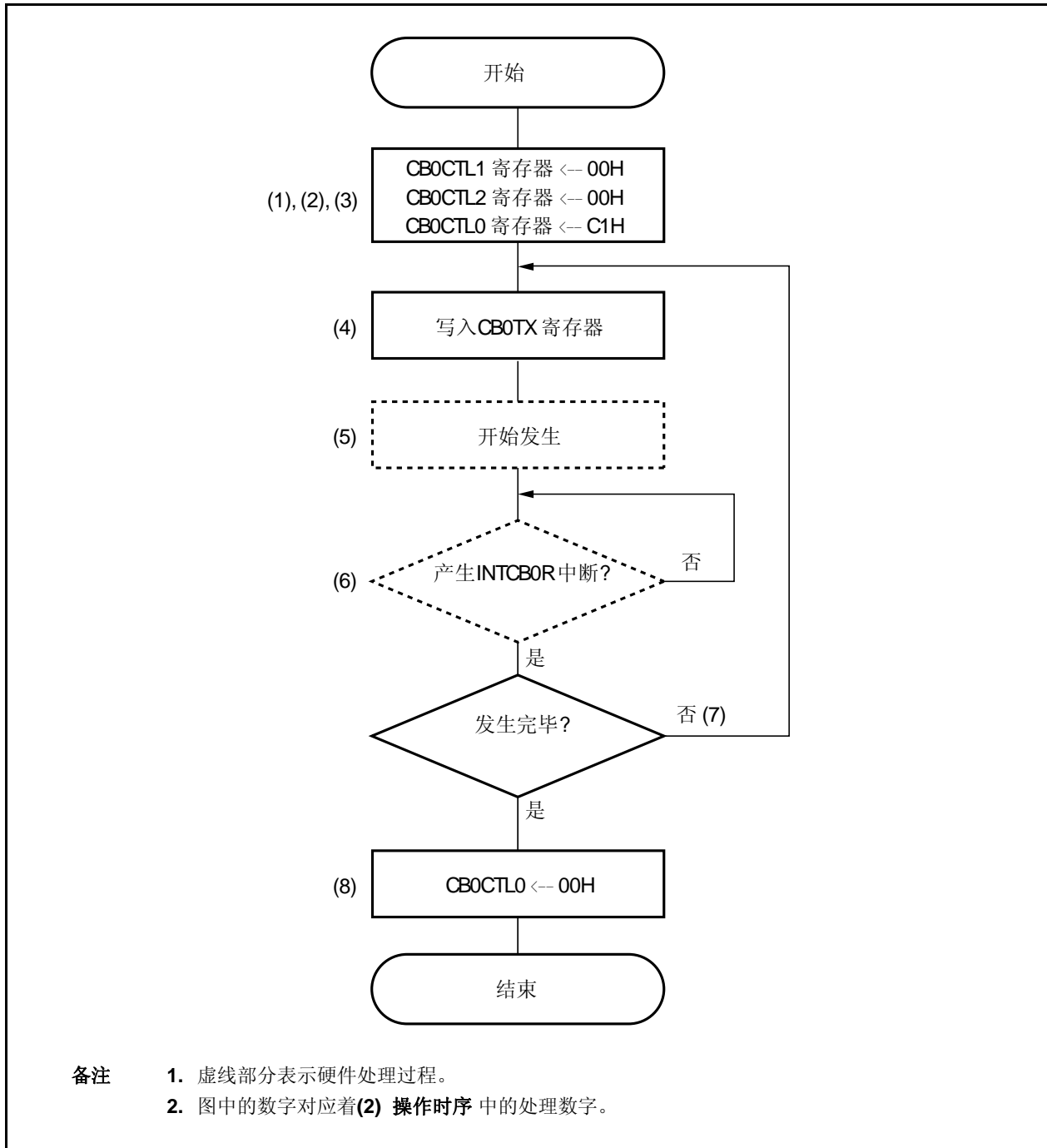
注意事项 单独传输模式中，CB0TSF 位置为 1 时写入 CB0TX 寄存器将被忽略。在传输操作过程中不产生任何影响。
使用连续传输模式，而不是单独传输模式。

13.4 操作

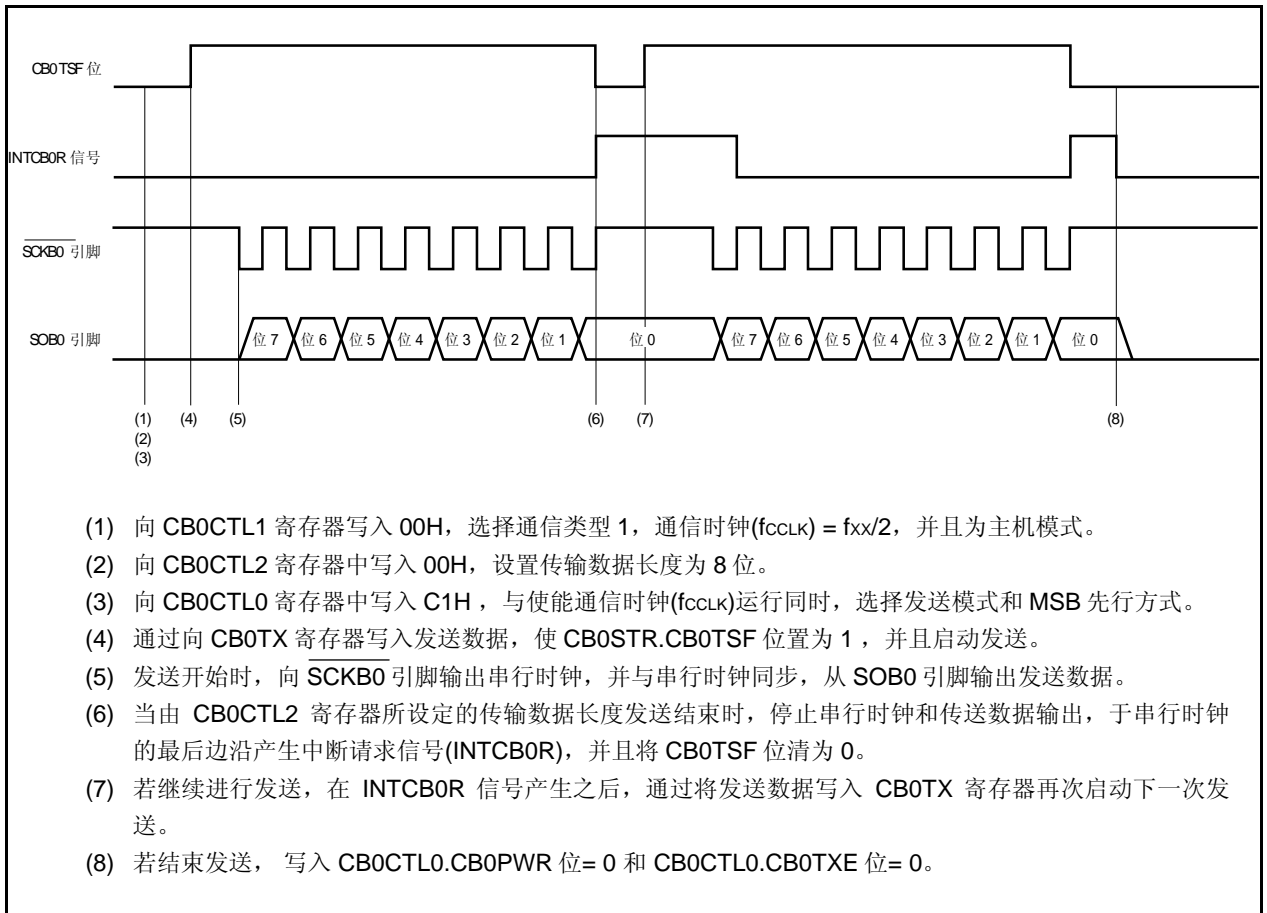
13.4.1 单独传送模式 (主机模式, 发送模式)

MSB 先行 (CB0CTL0.CB0DIR 位 = 0), 通信类型 1 (CB0CTL1.CB0CKP 和 CB0CTL1.CB0DAP 位 = 00), 通信时钟 (f_{CLK}) = $f_x/2$ (CB0CTL1.CB0CKS2 至 CB0CTL1.CB0CKS0 位 = 000), 传输数据长度 = 8 位 (CB0CTL2.CB0CL3 至 CB0CTL2.CB0CL0 位 = 0000)。

(1) 操作流程



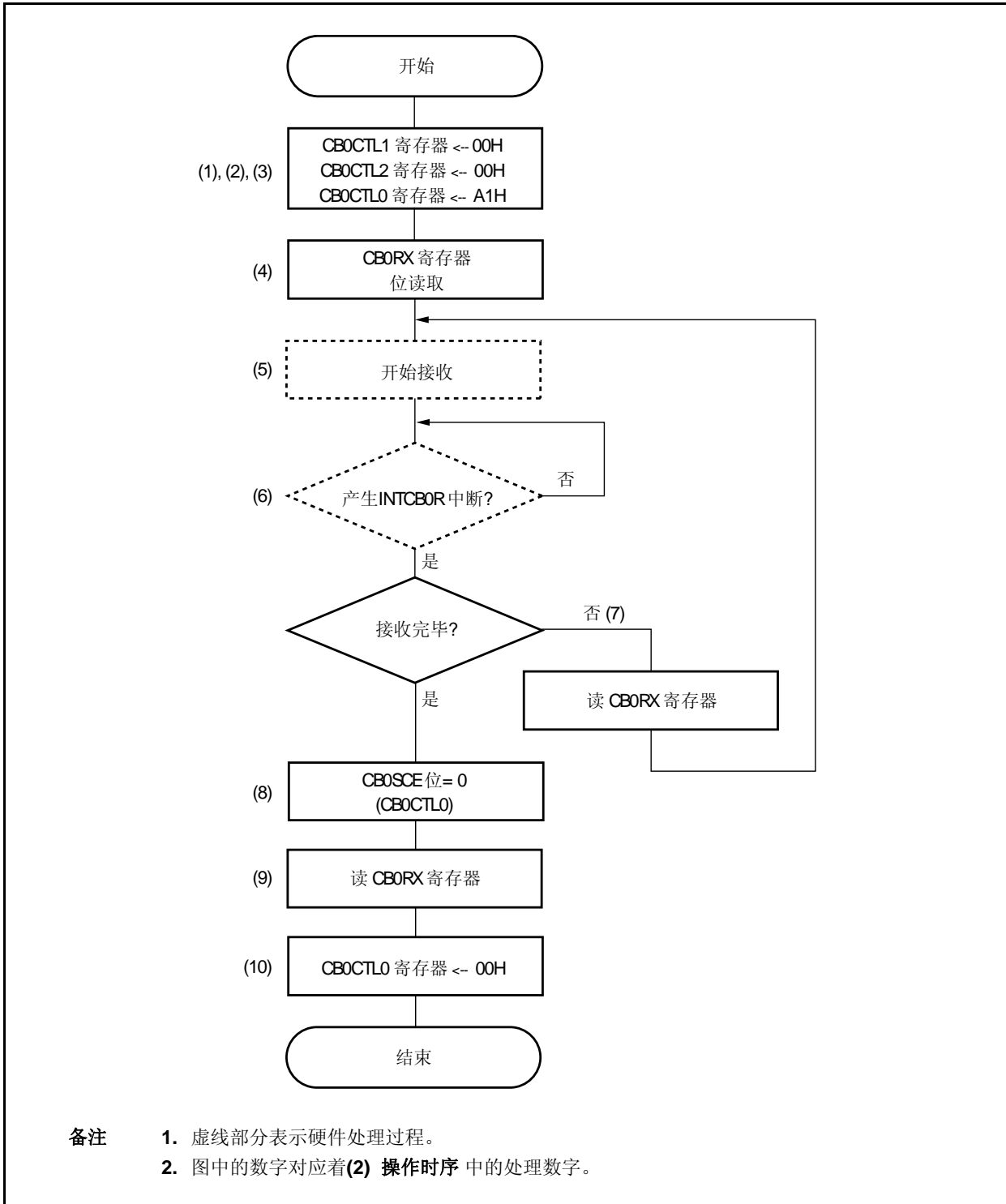
(2) 操作时序



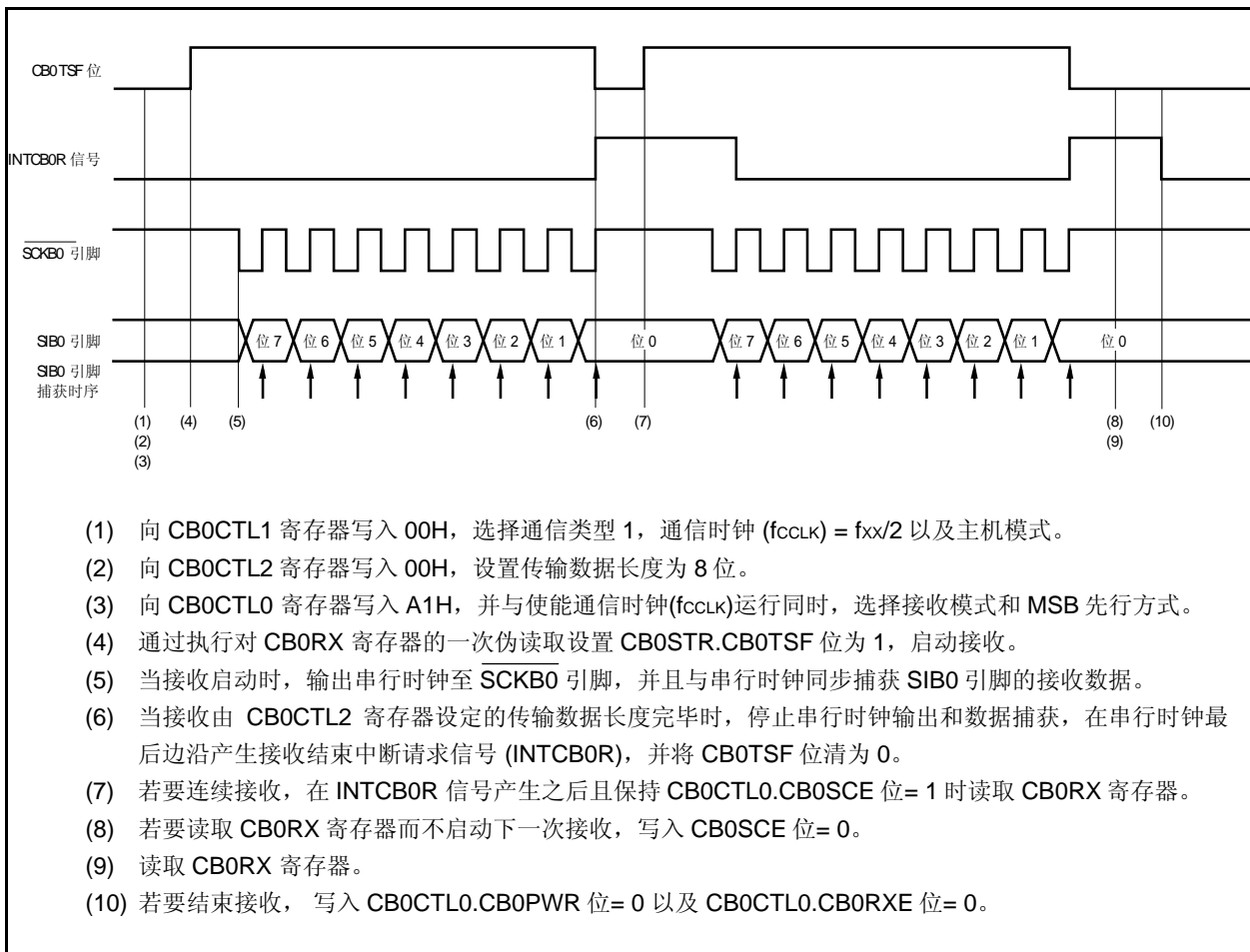
13.4.2 单独发送模式 (主机模式, 接收模式)

MSB 先行 (CB0CTL0.CB0DIR 位 = 0), 通信类型 1 (CB0CTL1.CB0CKP 和 CB0CTL1.CB0DAP 位 = 00), 通信时钟 (f_{CLK}) = f_{xx}/2 (CB0CTL1.CB0CKS2 至 CB0CTL1.CB0CKS0 位 = 000), 传输数据长度 = 8 位 (CB0CTL2.CB0CL3 至 CB0CTL2.CB0CL0 位 = 0000)。

(1) 操作流程



(2) 操作时序

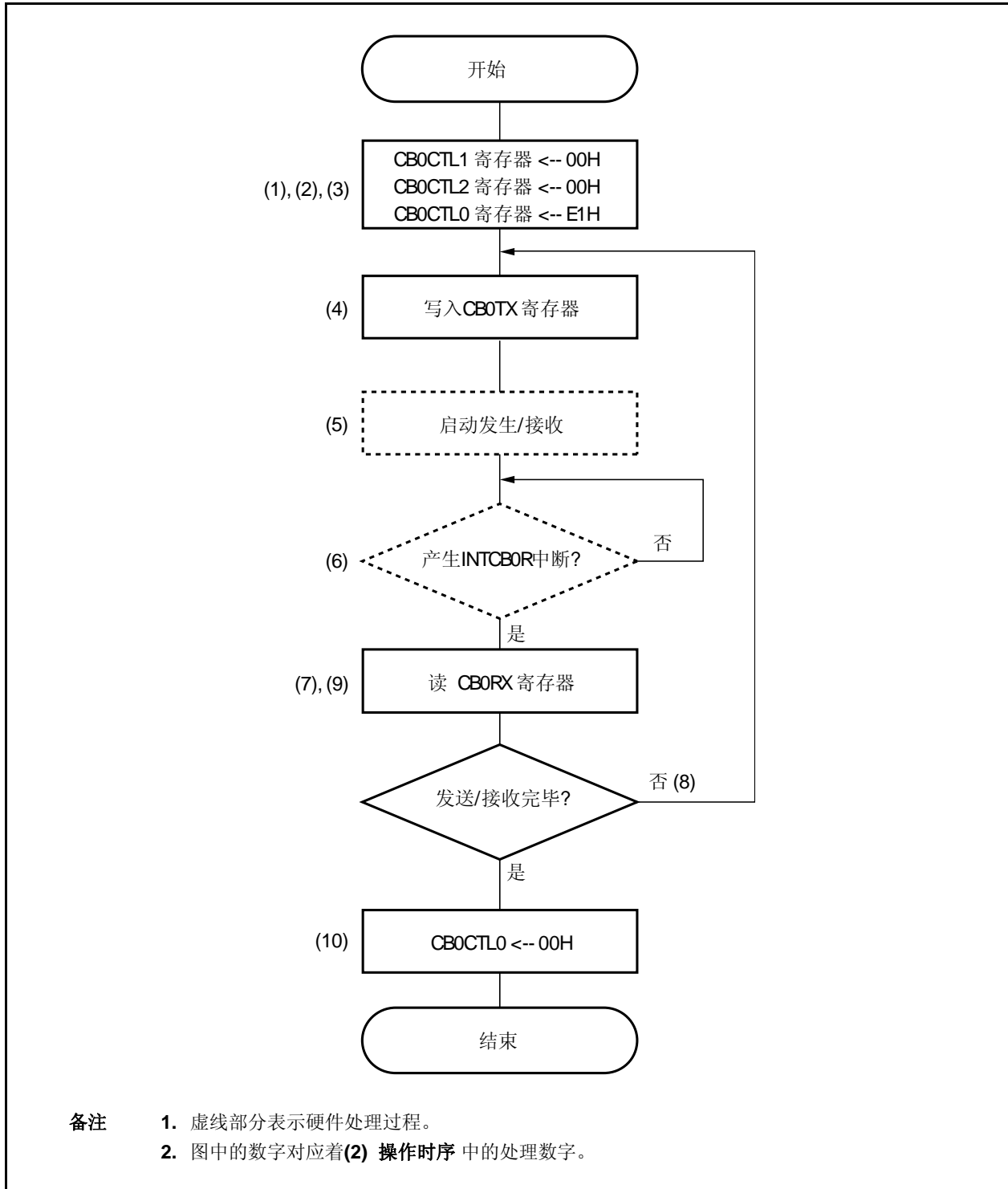


- (1) 向 **CB0CTL1** 寄存器写入 **00H**，选择通信类型 1，通信时钟 ($f_{cclk} = f_{xx}/2$) 以及主机模式。
- (2) 向 **CB0CTL2** 寄存器写入 **00H**，设置传输数据长度为 8 位。
- (3) 向 **CB0CTL0** 寄存器写入 **A1H**，并与使能通信时钟(f_{cclk})运行同时，选择接收模式和 MSB 先行方式。
- (4) 通过执行对 **CB0RX** 寄存器的一次伪读取设置 **CB0STR.CB0TSF** 位为 1，启动接收。
- (5) 当接收启动时，输出串行时钟至 **SCKB0** 引脚，并且与串行时钟同步捕获 **SIB0** 引脚的接收数据。
- (6) 当接收由 **CB0CTL2** 寄存器设定的传输数据长度完毕时，停止串行时钟输出和数据捕获，在串行时钟最后边沿产生接收结束中断请求信号 (**INTCB0R**)，并将 **CB0TSF** 位清为 0。
- (7) 若要连续接收，在 **INTCB0R** 信号产生之后且保持 **CB0CTL0.CB0SCE** 位= 1 时读取 **CB0RX** 寄存器。
- (8) 若要读取 **CB0RX** 寄存器而不启动下一次接收，写入 **CB0SCE** 位= 0。
- (9) 读取 **CB0RX** 寄存器。
- (10) 若要结束接收，写入 **CB0CTL0.CB0PWR** 位= 0 以及 **CB0CTL0.CB0RXE** 位= 0。

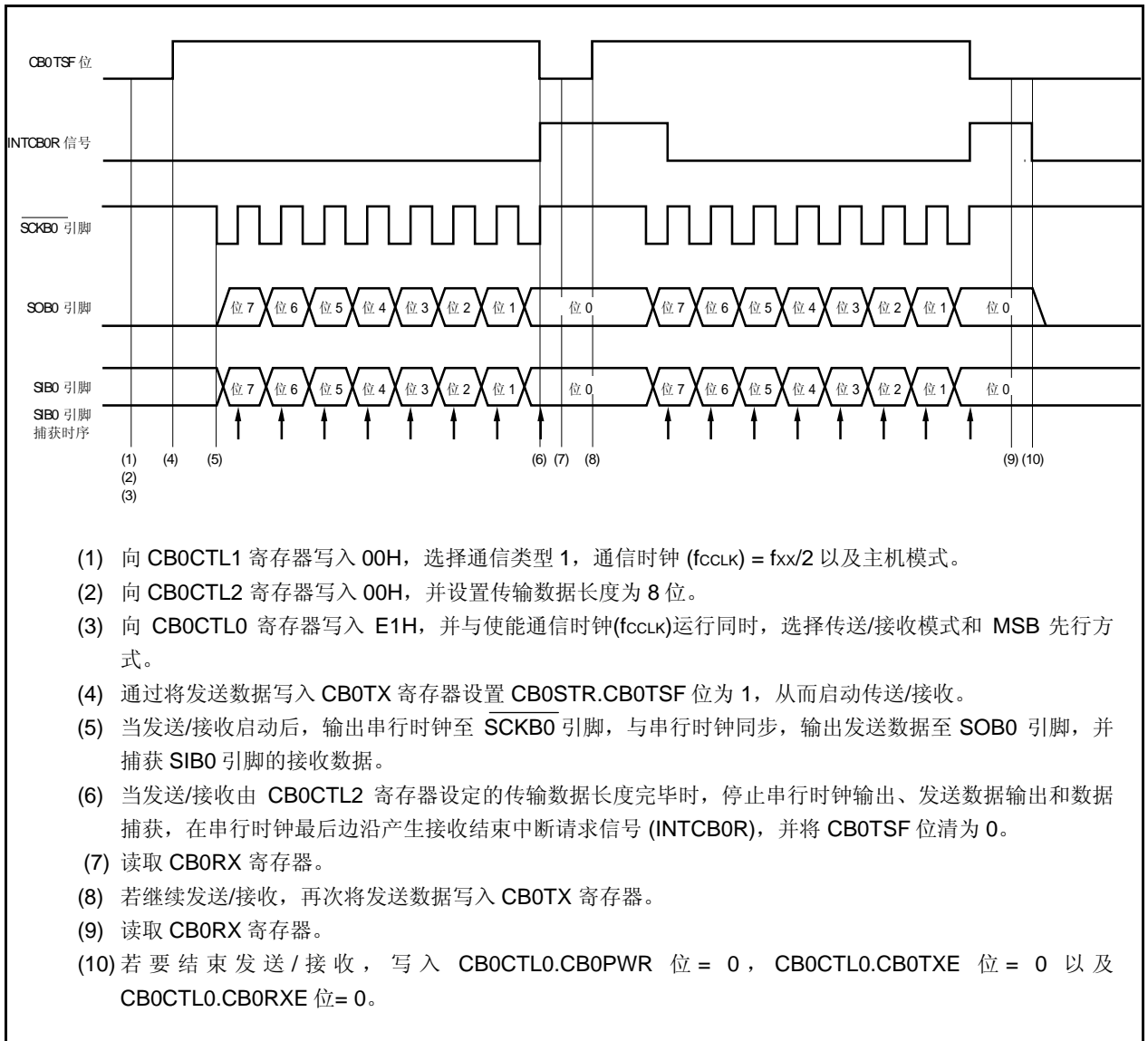
13.4.3 单独传送模式 (主机模式, 发送/接收模式)

MSB 先行(CB0CTL0.CB0DIR 位= 0), 通信类型 1 (CB0CTL1.CB0CKP 和 CB0CTL1.CB0DAP 位= 00), 通信时钟 (f_{CCLK}) = f_{xx}/2 (CB0CTL1.CB0CKS2 至 CB0CTL1.CB0CKS0 位= 000), 传输数据长度= 8 位(CB0CTL2.CB0CL3 至 CB0CTL2.CB0CL0 位= 0000)。

(1) 操作流程



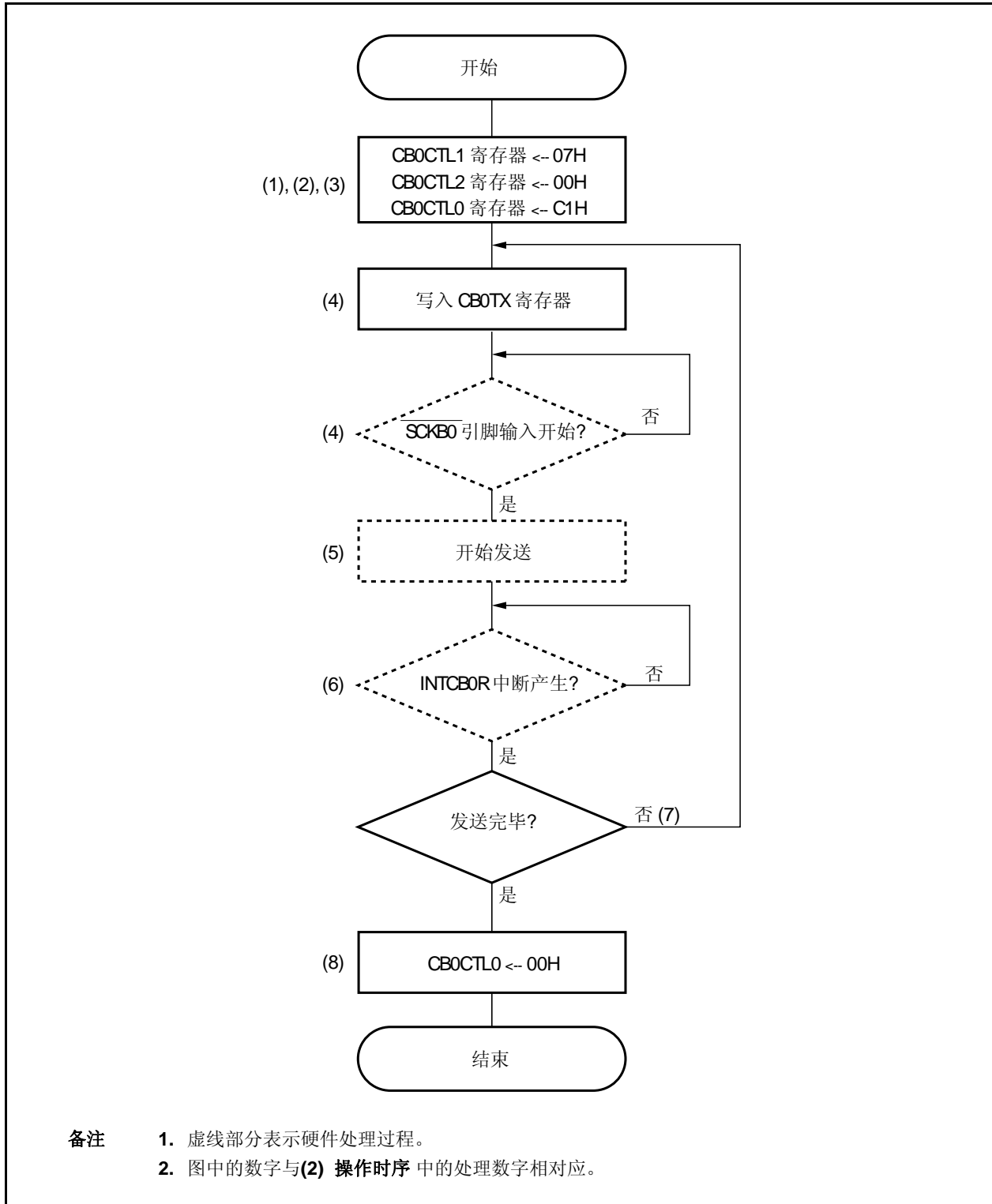
(2) 操作时序



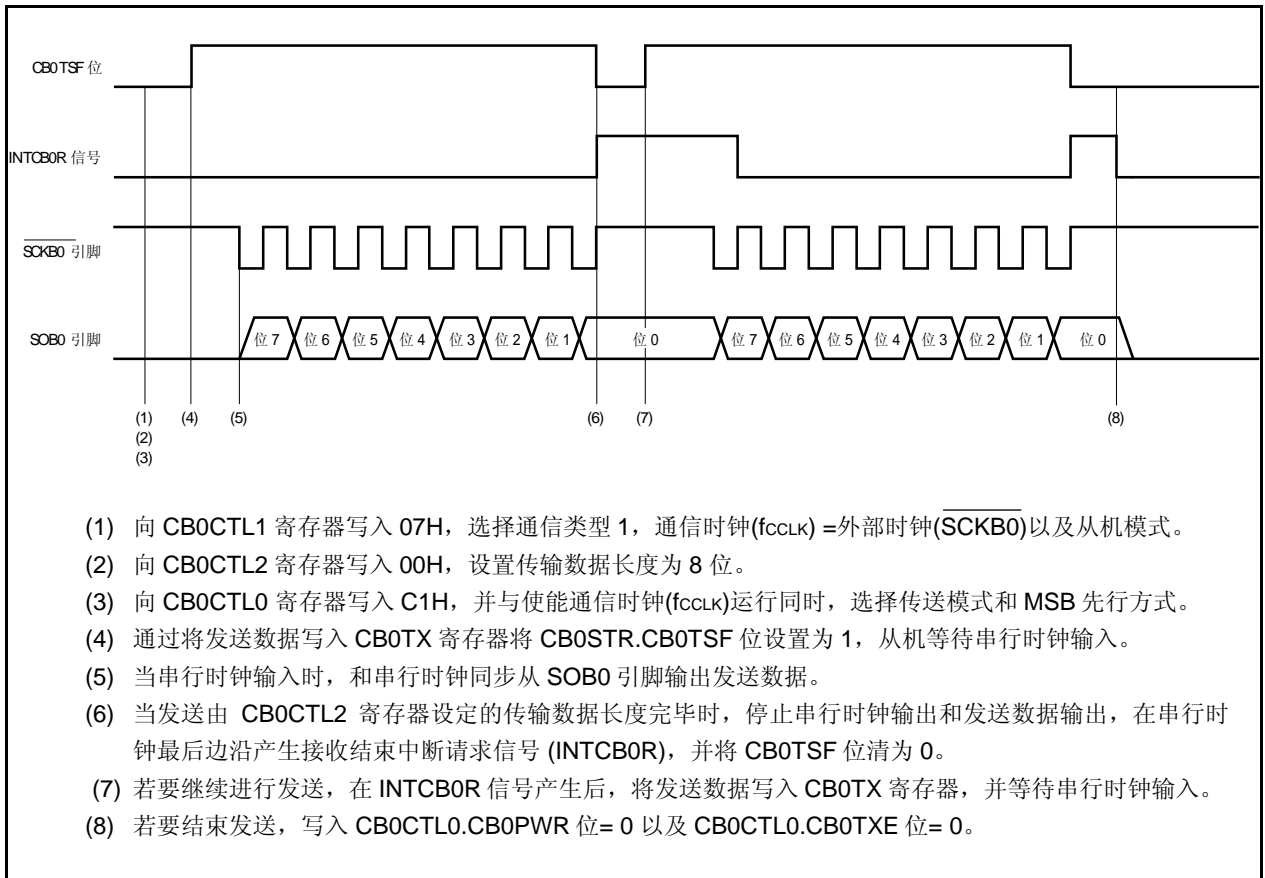
13.4.4 单独传送模式 (从机模式, 发送模式)

MSB 先行(CB0CTL0.CB0DIR 位= 0), 通信类型 1 (CB0CTL1.CB0CKP 和 CB0CTL1.CB0DAP 位= 00), 通信时钟 (f_{clock}) = 外部时钟(SCKB0) (CB0CTL1.CB0CKS2 至 CB0CTL1.CB0CKS0 位= 111), 传输数据长度 = 8 位 (CB0CTL2.CB0CL3 至 CB0CTL2.CB0CL0 位 = 0000)。

(1) 操作流程



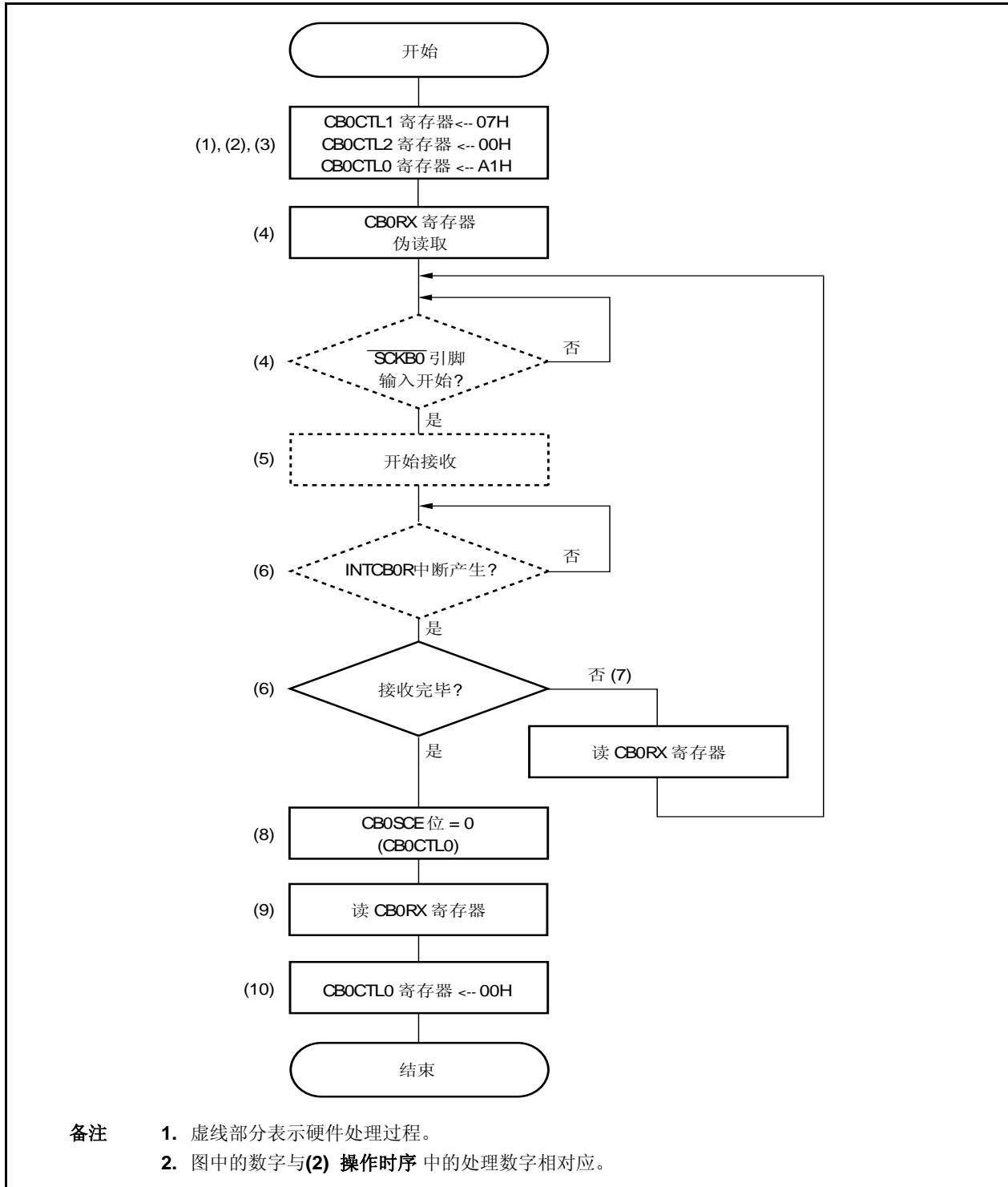
(2) 操作时序



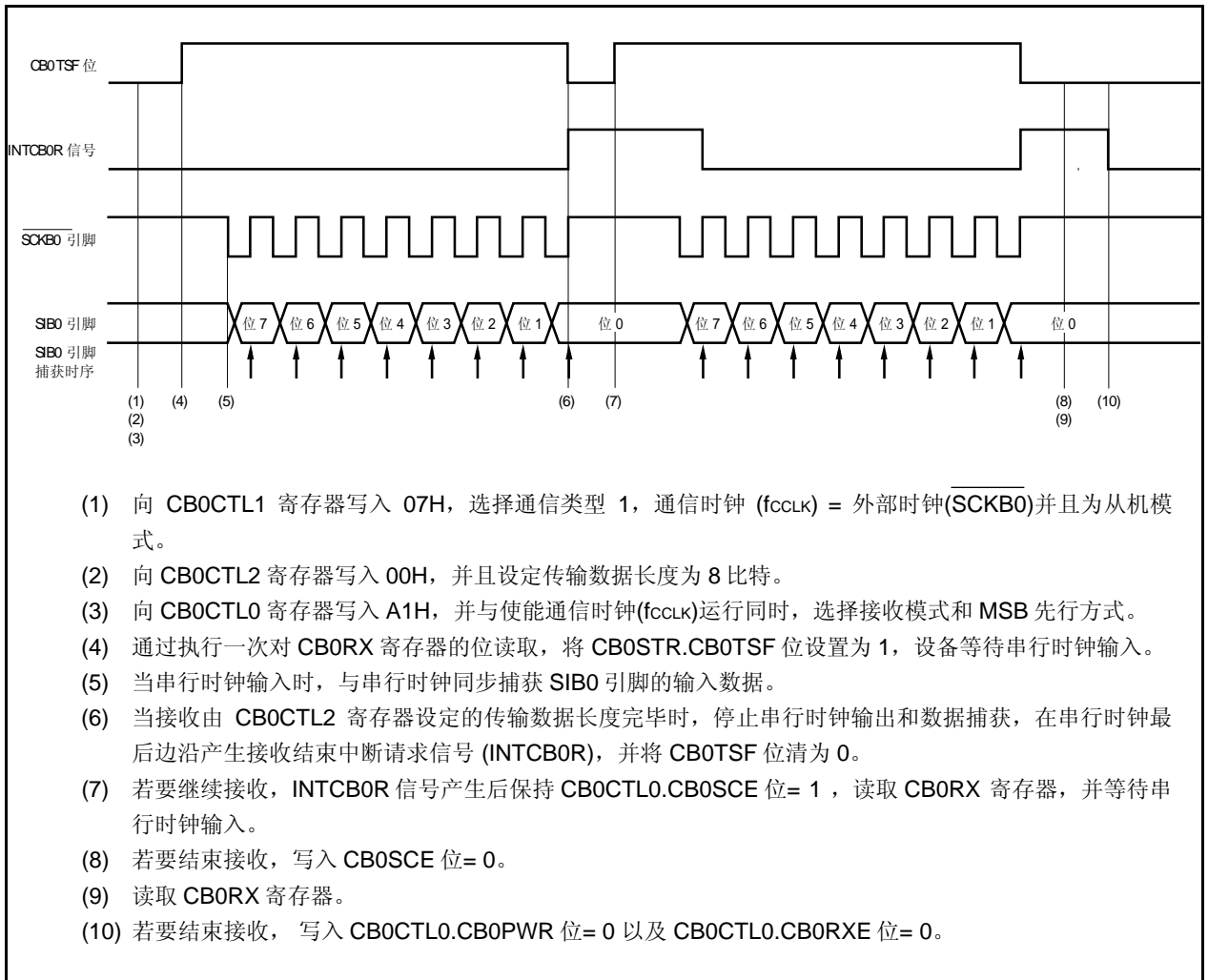
13.4.5 单独传送模式 (从机模式, 接收模式)

MSB 先行 (CB0CTL0.CB0DIR 位 = 0), 通信类型 1 (CB0CTL1.CB0CKP 和 CB0CTL1.CB0DAP 位 = 00), 通信时钟 (f_{CLK}) = 外部时钟 (SCKB0) (CB0CTL1.CB0CKS2 至 CB0CTL1.CB0CKS0 位 = 111), 传输数据长度 = 8 位 (CB0CTL2.CB0CL3 至 CB0CTL2.CB0CL0 位 = 0000)。

(1) 操作流程



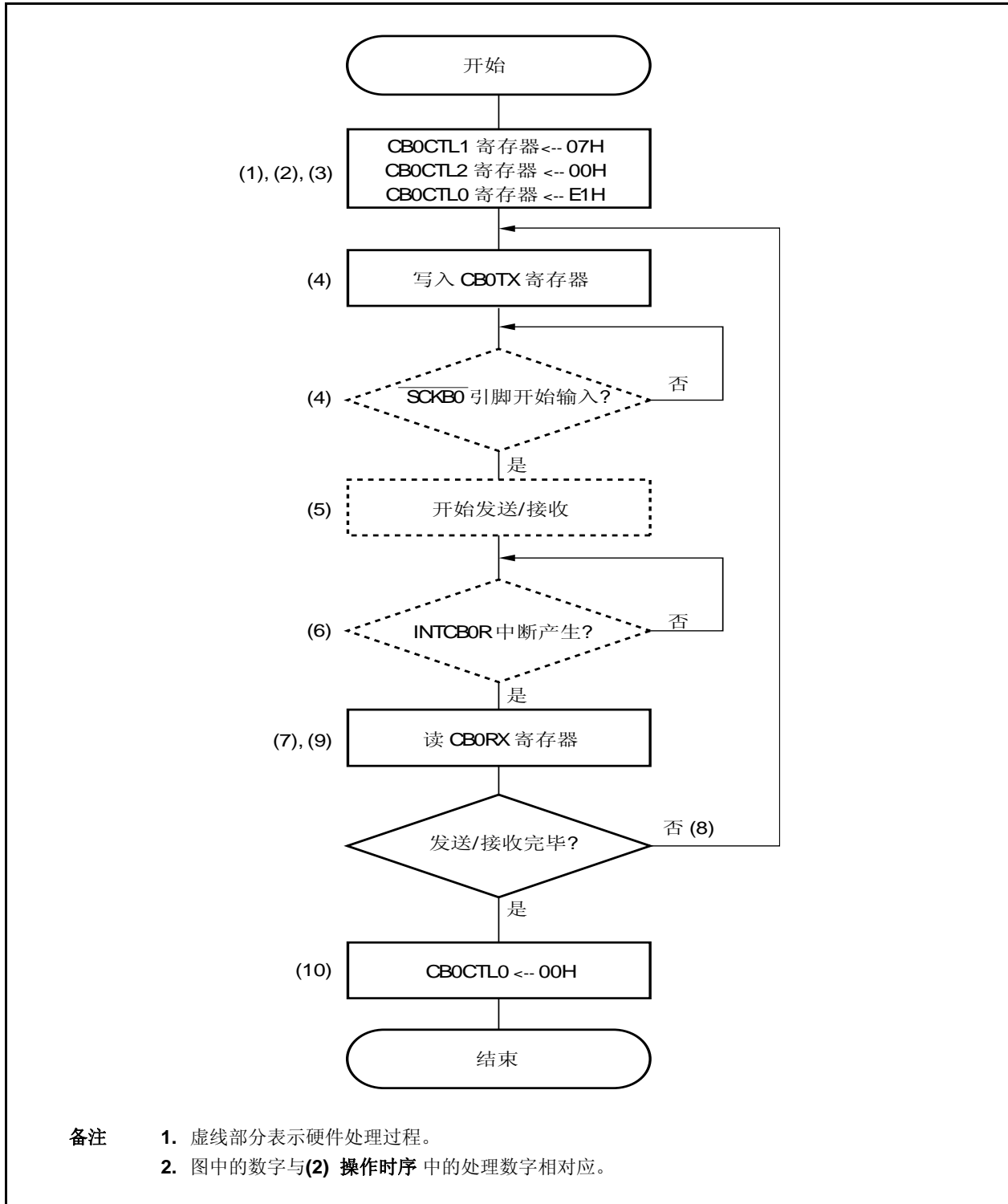
(2) 操作时序



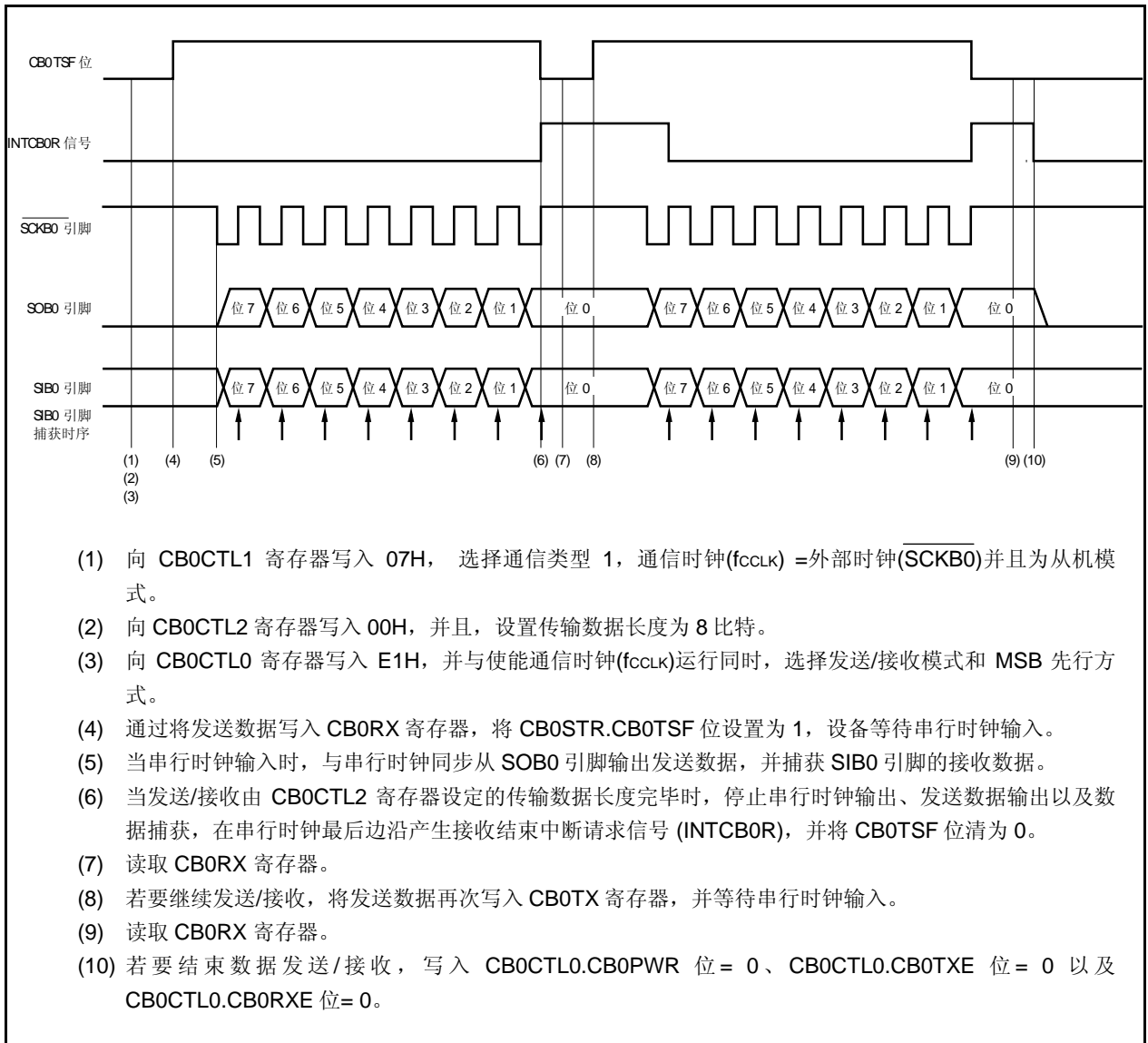
13.4.6 单独传送模式(从机模式, 发送/接收模式)

MSB 先行(CB0CTL0.CB0DIR 位= 0), 通信类型 1 (CB0CTL1.CB0CKP 和 CB0CTL1.CB0DAP 位= 00), 通信时钟 (f_{CCLK}) = 外部时钟(SCKB0) (CB0CTL1.CB0CKS2 至 CB0CTL1.CB0CKS0 位 = 111), 传输数据长度 = 8 比特 (CB0CTL2.CB0CL3 至 CB0CTL2.CB0CL0 位 = 0000)。

(1) 操作流程



(2) 操作时序

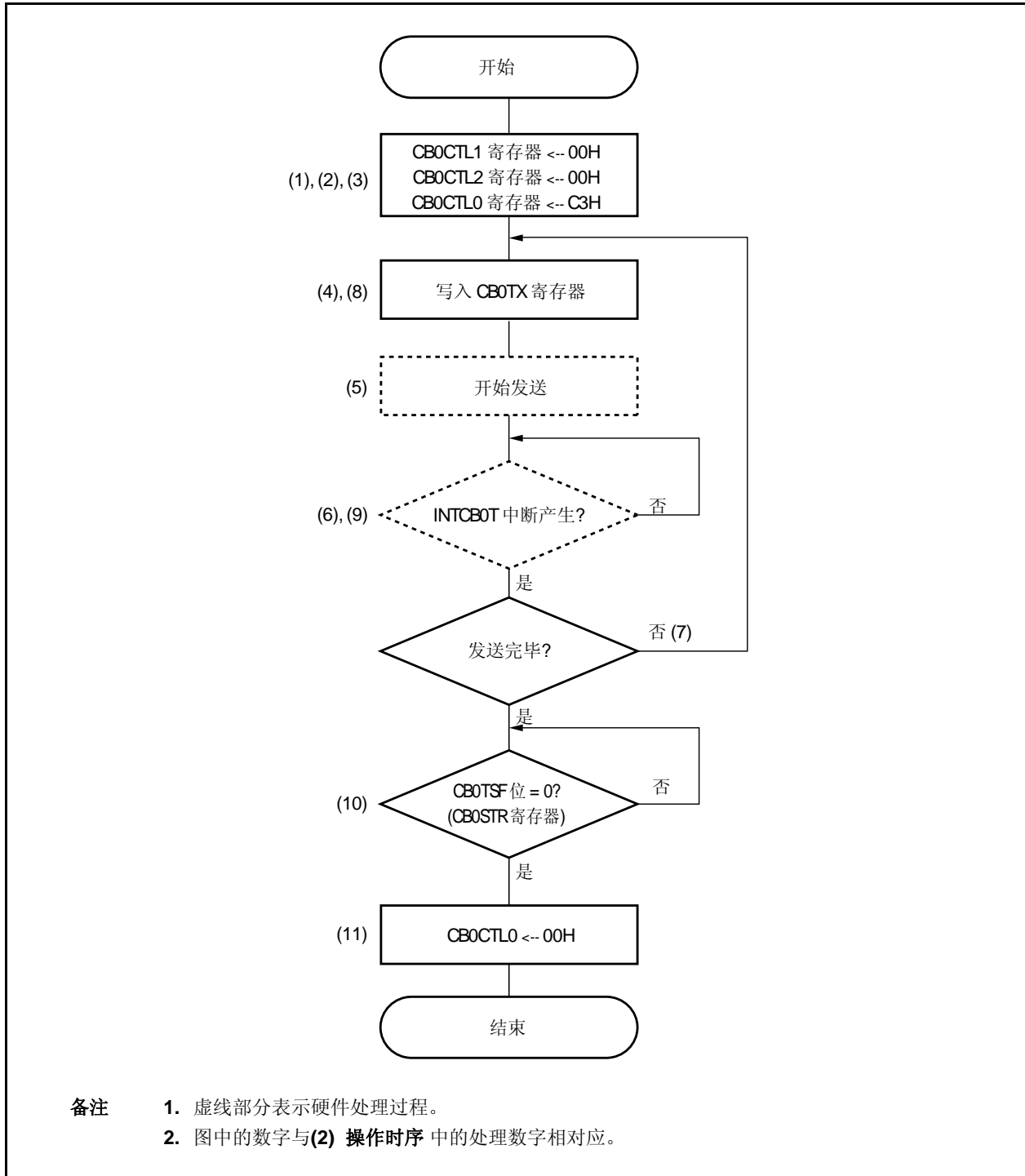


- (1) 向 CB0CTL1 寄存器写入 07H，选择通信类型 1，通信时钟(f_{cclk}) =外部时钟(SCKB0)并且为从机模式。
- (2) 向 CB0CTL2 寄存器写入 00H，并且，设置传输数据长度为 8 比特。
- (3) 向 CB0CTL0 寄存器写入 E1H，并与使能通信时钟(f_{cclk})运行同时，选择发送/接收模式和 MSB 先行方式。
- (4) 通过将发送数据写入 CB0RX 寄存器，将 CB0STR.CB0TSF 位设置为 1，设备等待串行时钟输入。
- (5) 当串行时钟输入时，与串行时钟同步从 SOB0 引脚输出发送数据，并捕获 SIB0 引脚的接收数据。
- (6) 当发送/接收由 CB0CTL2 寄存器设定的传输数据长度完毕时，停止串行时钟输出、发送数据输出以及数据捕获，在串行时钟最后边沿产生接收结束中断请求信号 (INTCB0R)，并将 CB0TSF 位清为 0。
- (7) 读取 CB0RX 寄存器。
- (8) 若要继续发送/接收，将发送数据再次写入 CB0TX 寄存器，并等待串行时钟输入。
- (9) 读取 CB0RX 寄存器。
- (10) 若要结束数据发送/接收，写入 CB0CTL0.CB0PWR 位 = 0、CB0CTL0.CB0TXE 位 = 0 以及 CB0CTL0.CB0RXE 位 = 0。

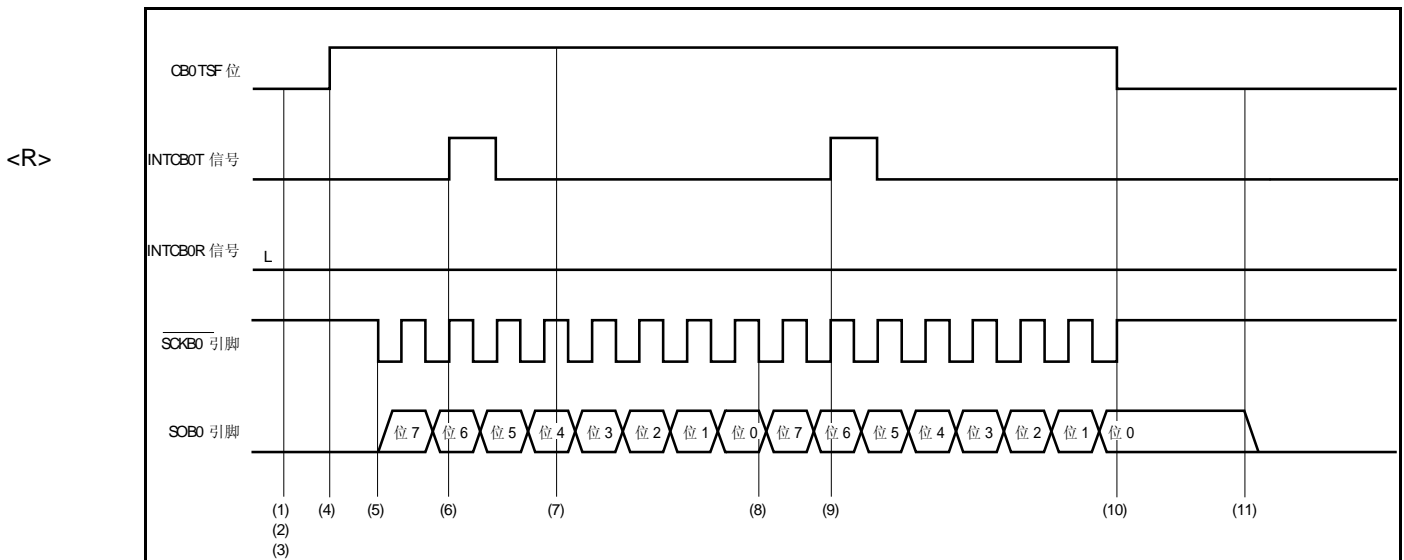
13.4.7 连续传送模式 (主机模式, 发送模式)

MSB 先行(CB0CTL0.CB0DIR 位= 0), 通信类型 1 (CB0CTL1.CB0CKP 和 CB0CTL1.CB0DAP 位 = 00), 通信时钟 ($f_{\text{CCLK}} = f_{\text{xx}}/2$ (CB0CTL1.CB0CKS2 至 CB0CTL1.CB0CKS0 位= 000)), 传输数据长度= 8 位(CB0CTL2.CB0CL3 至 CB0CTL2.CB0CL0 位= 0000)。

(1) 操作流程



(2) 操作时序



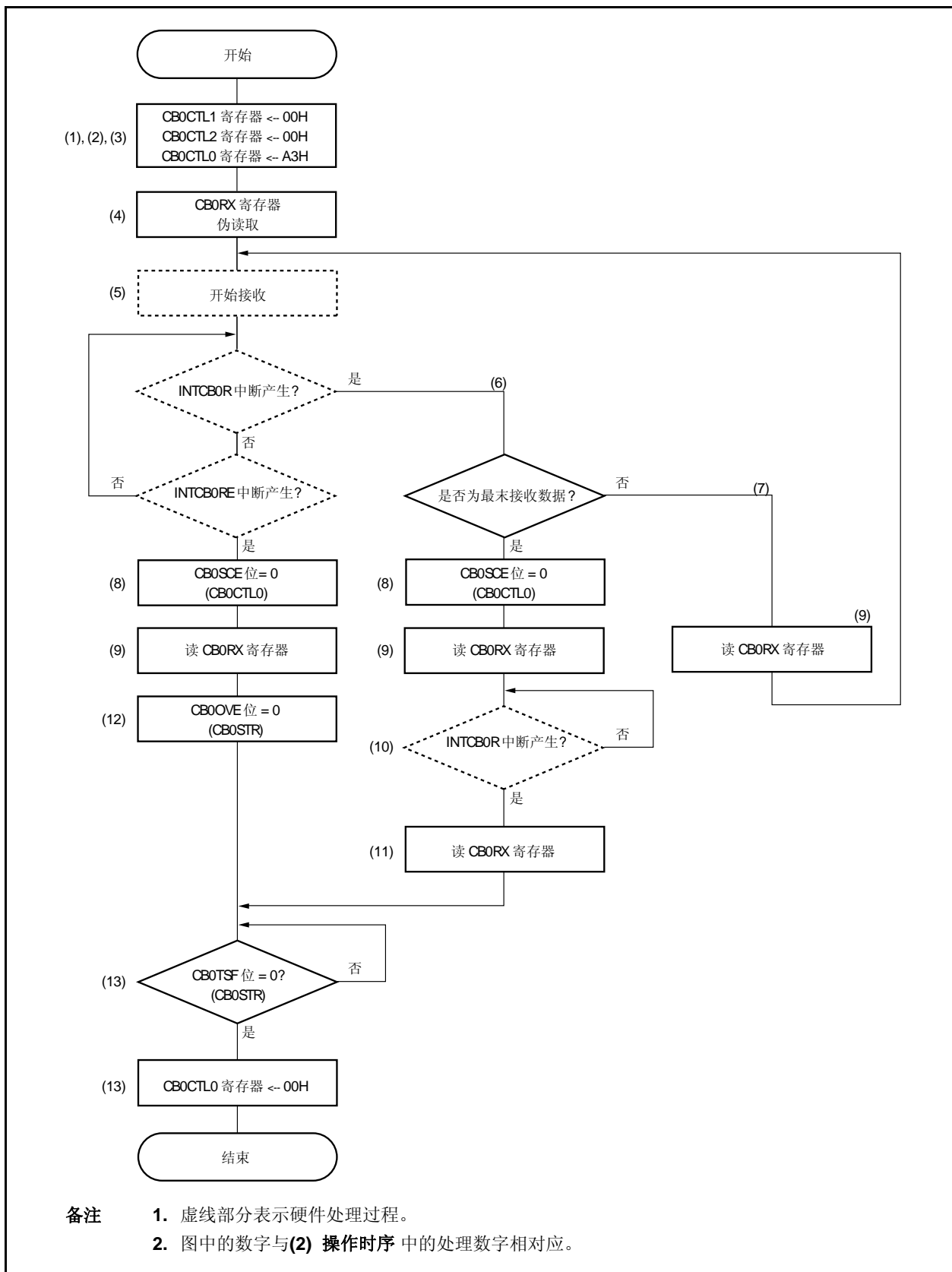
- (1) 向 CB0CTL1 寄存器写入 00H, 设置通信类型 1, 通信时钟 ($f_{cclk} = f_{xx}/2$), 及主机模式。
- (2) 向 CB0CTL2 寄存器写入 00H, 并设定传输数据长度为 8 比特。
- (3) 向 CB0CTL0 寄存器写入 C3H, 并与使能通信时钟(f_{cclk})运行同时, 选择发送模式、MSB 先行方式和连续传输模式。
- (4) 通过将发送数据写入 CB0RX 寄存器, 将 CB0STR.CB0TSF 位设置为 1, 发送开始。
- (5) 当发送开始后, 输出串行时钟至 SCKB0 引脚, 并与串行时钟同步, 发送数据由 SOB0 引脚输出。
- (6) 当从 CB0TX 寄存器中将发送数据传输至移位寄存器完毕, 并且允许写入 CB0TX 寄存器时, 产生发送使能中断请求信号 (INTCB0T)。
- (7) 若要继续发送, 于 INTCB0T 信号产生之后, 将发送数据再次写入 CB0TX 寄存器。
- (8) 当通信完毕之前, CB0TX 寄存器写入一个新的发送数据, 则紧随通信结束开始下一次通信。
- (9) 从 CB0TX 寄存器中将发送数据传输至移位寄存器完毕后并且已经产生 INTCB0T 信号。为了当前的连续发送传输, 不要写入 CB0TX 寄存器。
- (10) 当传输结束之前下一个发送数据没有写入 CB0TX 寄存器, 传输完毕后停止向 SCKB0 引脚输出串行时钟, 并将 CB0TSF 位清除为 0。
- (11) 若要解除发送使能状态, 检查 CB0TSF 位 = 0 之后, 写入 CB0CTL0.CB0PWR 位 = 0 和 CB0CTL0.CB0TXE 位 = 0。

注意事项 连续发送模式中, 不产生接收结束中断请求信号(INTCB0R)。

13.4.8 连续传输模式 (主机模式, 接收模式)

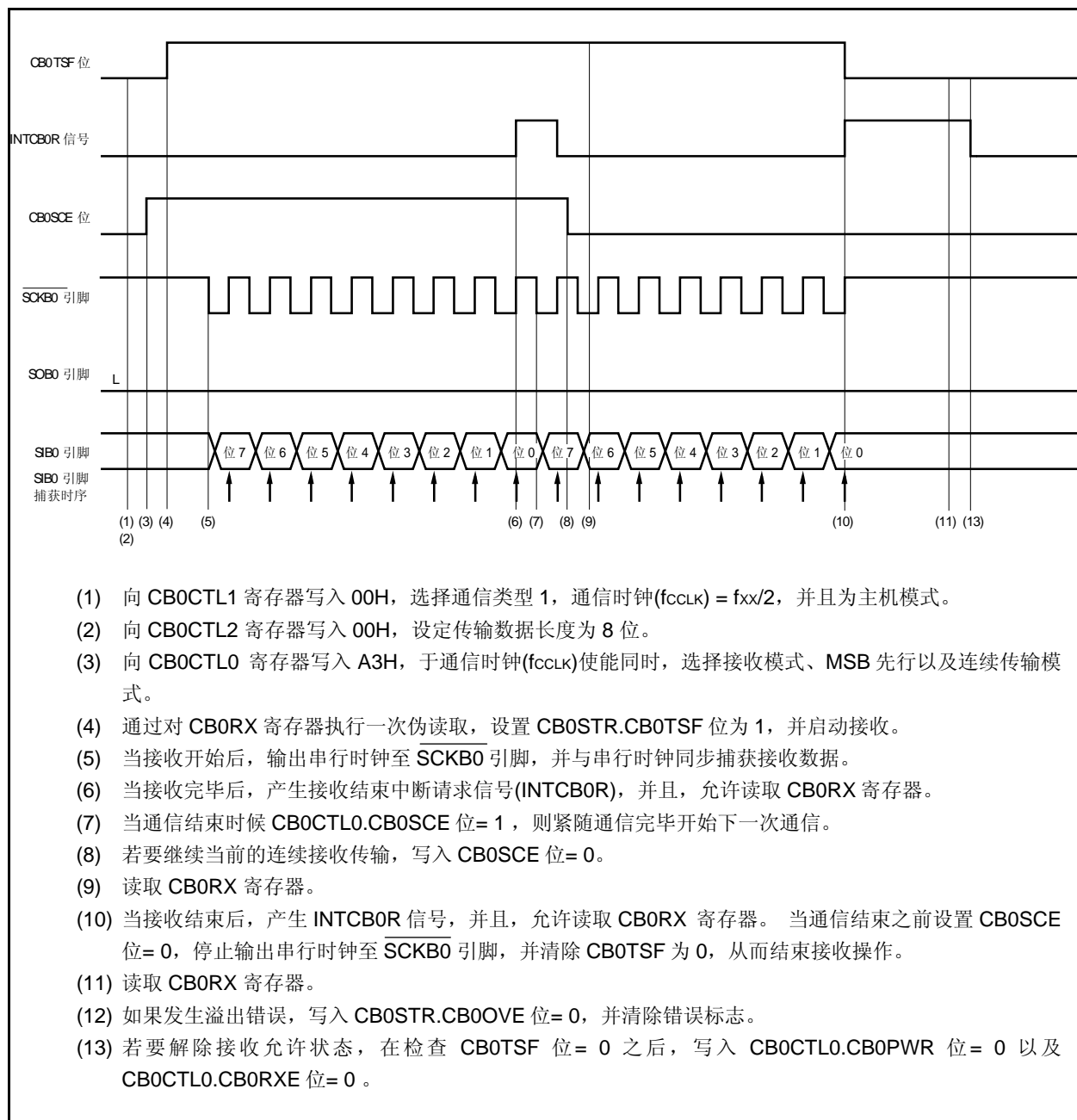
MSB 先行(CB0CTL0.CB0DIR 位= 0), 通信类型 1 (CB0CTL1.CB0CKP 和 CB0CTL1.CB0DAP 位= 00), 通信时钟 (f_{CCLK}) = $f_{\text{xx}}/2$ (CB0CTL1.CB0CKS2 至 CB0CTL1.CB0CKS0 位= 000), 传输数据长度= 8 比特(CB0CTL2.CB0CL3 至 CB0CTL2.CB0CL0 位= 0000)。

(1) 操作流程



备注 1. 虚线部分表示硬件处理过程。
2. 图中的数字与(2) 操作时序 中的处理数字相对应。

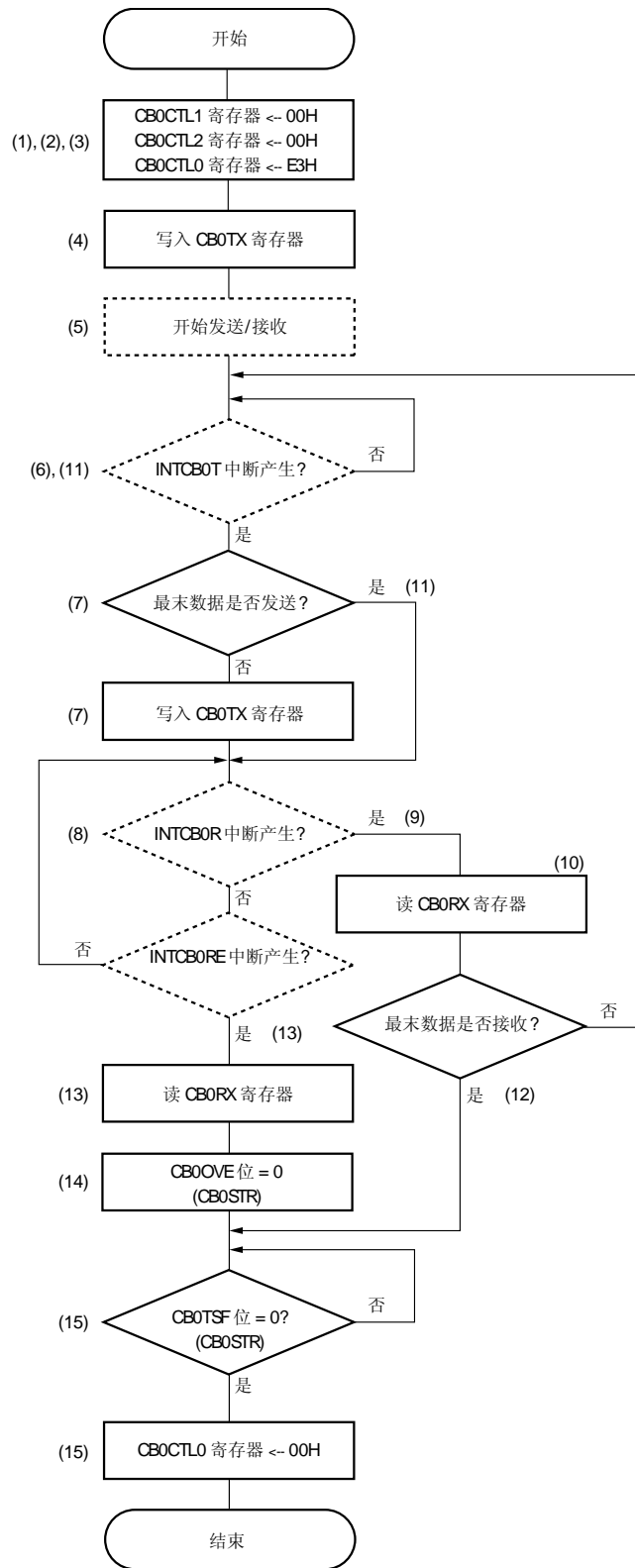
(2) 操作时序



13.4.9 连续传输模式（主机模式，发送/接收模式）

MSB 先行 (CB0CTL0.CB0DIR 位= 0)，通信类型 1 (CB0CTL1.CB0CKP 和 CB0CTL1.CB0DAP 位= 00)，通信时钟 (f_{CCLK}) = f_{xx}/2 (CB0CTL1.CB0CKS2 至 CB0CTL1.CB0CKS0 位 = 000)，传输数据长度= 8 位 (CB0CTL2.CB0CL3 至 CB0CTL2.CB0CL0 位= 0000)。

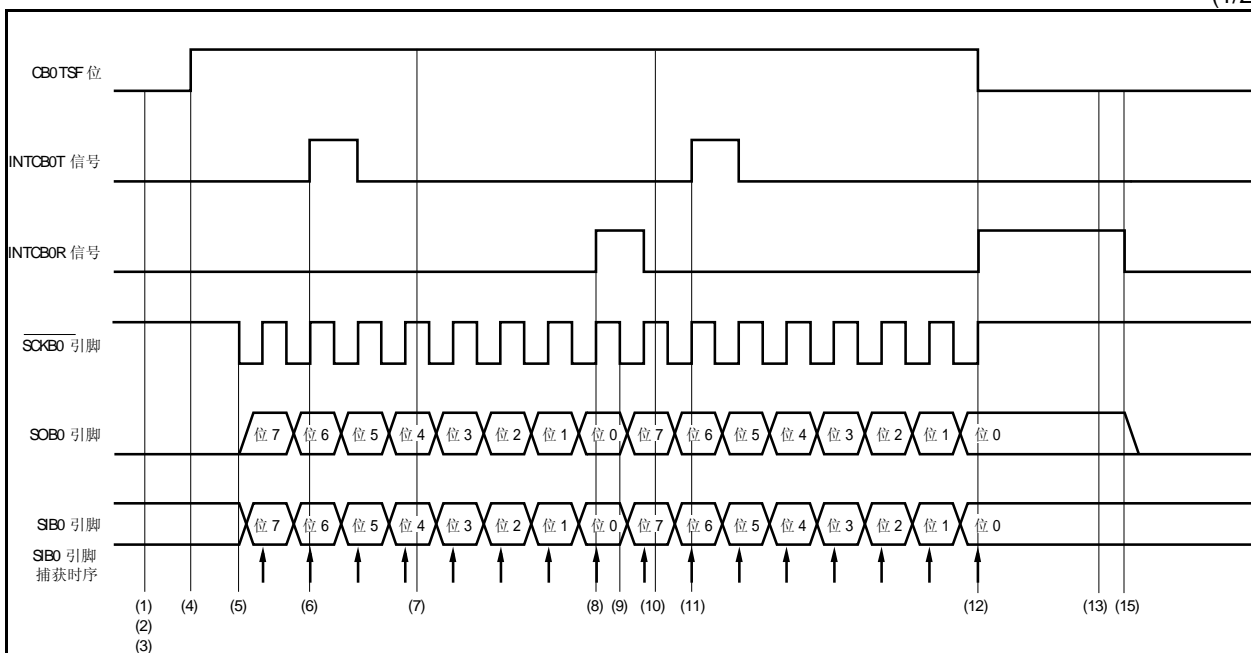
(1) 操作流程



- 备注
1. 虚线部分表示硬件处理过程。
 2. 图中的数字与(2) 操作时序 中的处理数字相对应。

(2) 操作时序

(1/2)



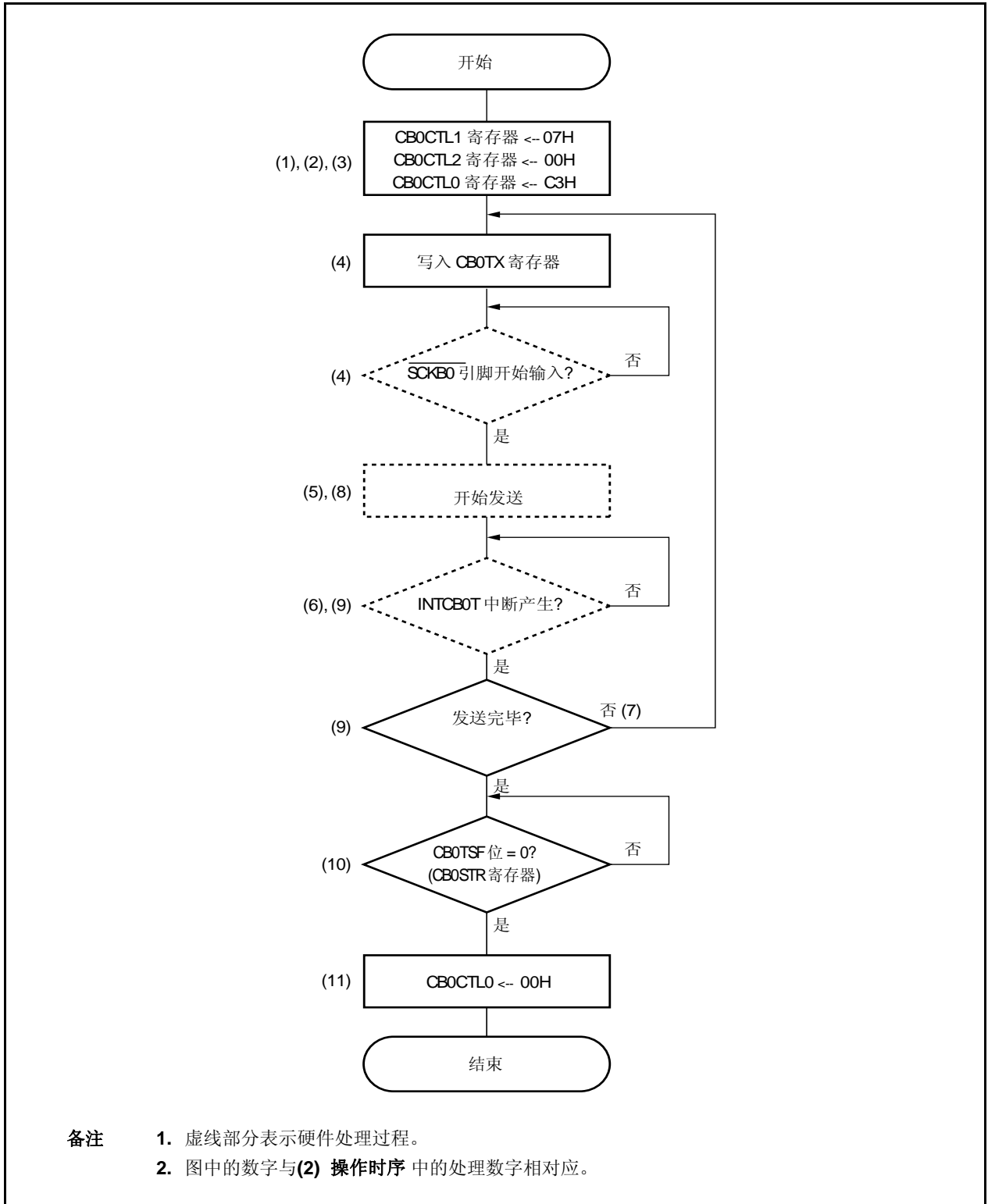
- (1) 向 CB0CTL1 寄存器写入 00H, 选择通信类型 1, 通信时钟(f_{cclk}) = $f_{xx}/2$, 并且为主机模式。
- (2) 向 CB0CTL2 寄存器写入 00H, 设定传输数据长度为 8 比特。
- (3) 向 CB0CTL0 寄存器写入 E3H, 于通信时钟(f_{cclk})操作使能同时, 选择发送/接收模式、MSB 先行以及连续接收模式。
- (4) 通过将发送数据写入 CB0TX 寄存器, 将 CB0STR.CB0TSF 位置为 1, 启动发送/接收。
- (5) 当发送/接收启动之后, 输出串行时钟至 SCKB0 引脚, 与串行时钟同步输出发送数据至 SOB0 引脚, 并且, 捕获 SIB0 引脚的接收数据。
- (6) 当由 CB0TX 寄存器传输发送数据至移位寄存器完毕, 并且允许写入 CB0TX 寄存器, 则产生发送允许中断请求信号 (INTCB0T)。
- (7) 若要继续发送/接收传输, INTCB0T 信号产生之后, 再次将发送数据写入 CB0TX 寄存器。
- (8) 当一次发送/接收完成后, 产生接收结束中断请求信号 (INTCB0R), 并且, 允许读取 CB0RX 寄存器。
- (9) 当通信完成之前, 新的发送数据已经写入 CB0TX 寄存器, 则紧随通信完成开始下一次通信。
- (10) 读取 CB0RX 寄存器。

- (11) 从 CB0TX 寄存器传输发送数据至移位寄存器完毕，且产生 INTCB0T 信号。若要结束当前的连续发送/接收传输，不要写入 CB0TX 寄存器。
- (12) 当传输完毕之前下一个发送数据未写入 CB0TX 寄存器，传输完毕后，则停止输出串行时钟至 $\overline{\text{SCKB0}}$ 引脚，并将 CB0TSF 位清除为 0。
- (13) 当产生接收错误中断请求信号 (INTCB0RE) 时，读取 CB0RX 寄存器。
- (14) 如果发生溢出错误，写入 CB0STR.CB0OVE 位= 0，并清除错误标志。
- (15) 若要解除发送/接收使能状态，在检查 CB0TSF 位= 0 之后，写入 CB0CTL0.CB0PWR 位= 0、CB0CTL0.CB0TXE 位= 0 和 CB0CTL0.CB0RXE 位= 0。

13.4.10 连续传输模式 (从机模式, 发送模式)

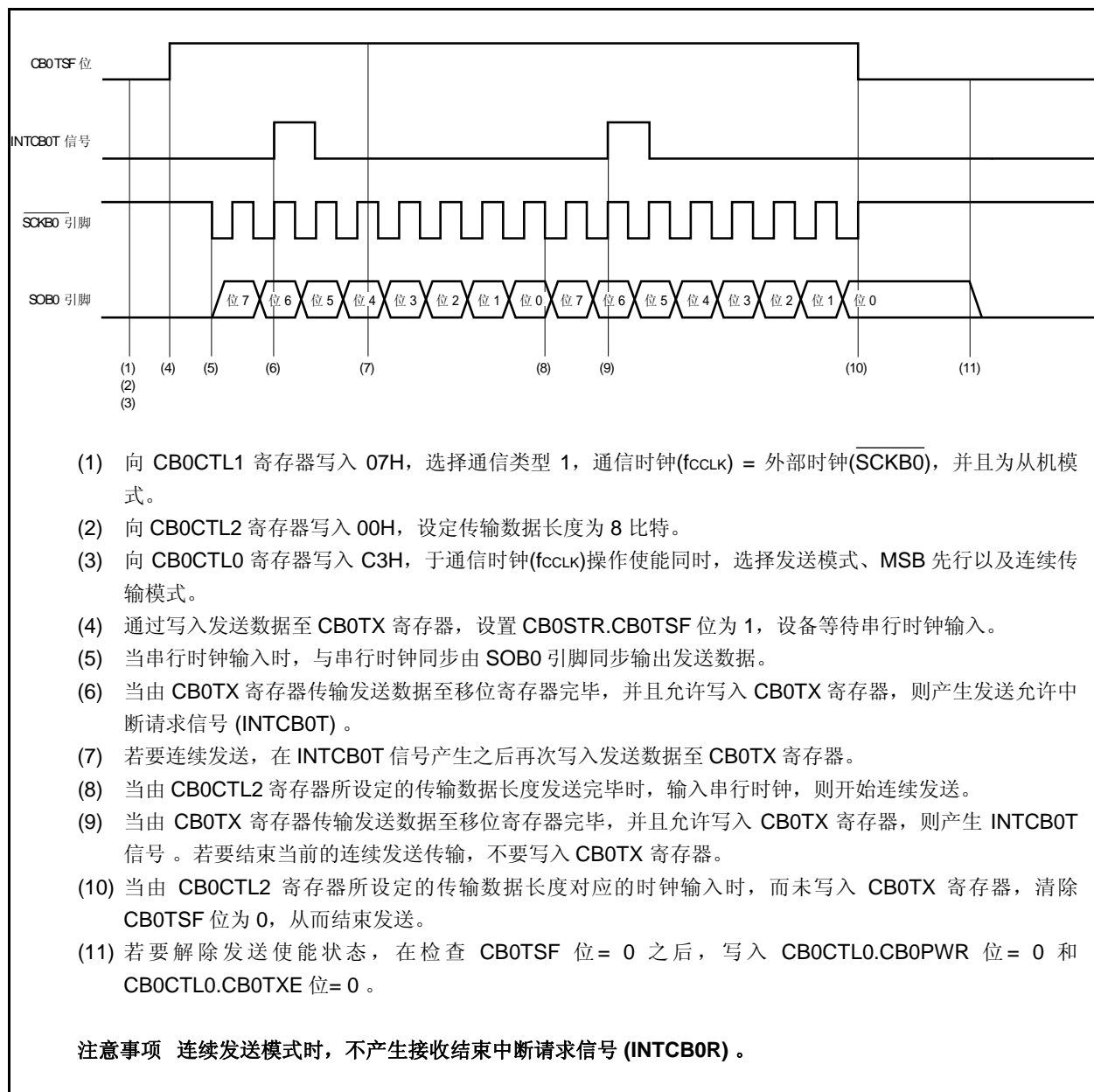
MSB 先行(CB0CTL0.CB0DIR 位= 0), 通信类型 1 (CB0CTL1.CB0CKP 和 CB0CTL1.CB0DAP 位= 00), 通信时钟 (f_{CCLK}) = 外部时钟(SCKB0) (CB0CTL1.CB0CKS2 至 CB0CTL1.CB0CKS0 位= 111), 传输数据长度= 8 比特 (CB0CTL2.CB0CL3 至 CB0CTL2.CB0CL0 位= 0000)。

(1) 操作流程



- 备注
1. 虚线部分表示硬件处理过程。
 2. 图中的数字与(2) 操作时序 中的处理数字相对应。

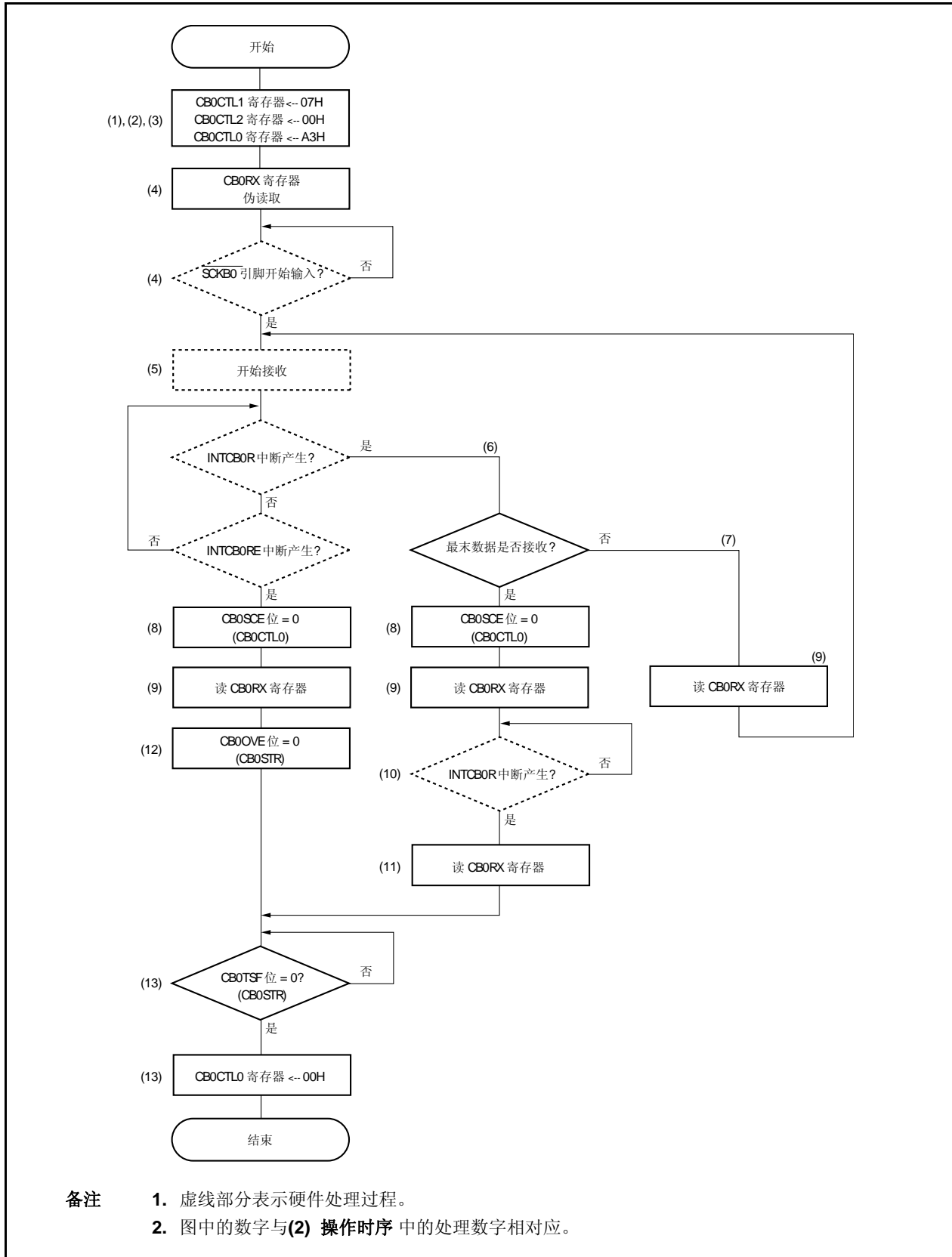
(2) 操作时序



13.4.11 连续传输模式(从机模式, 接收模式)

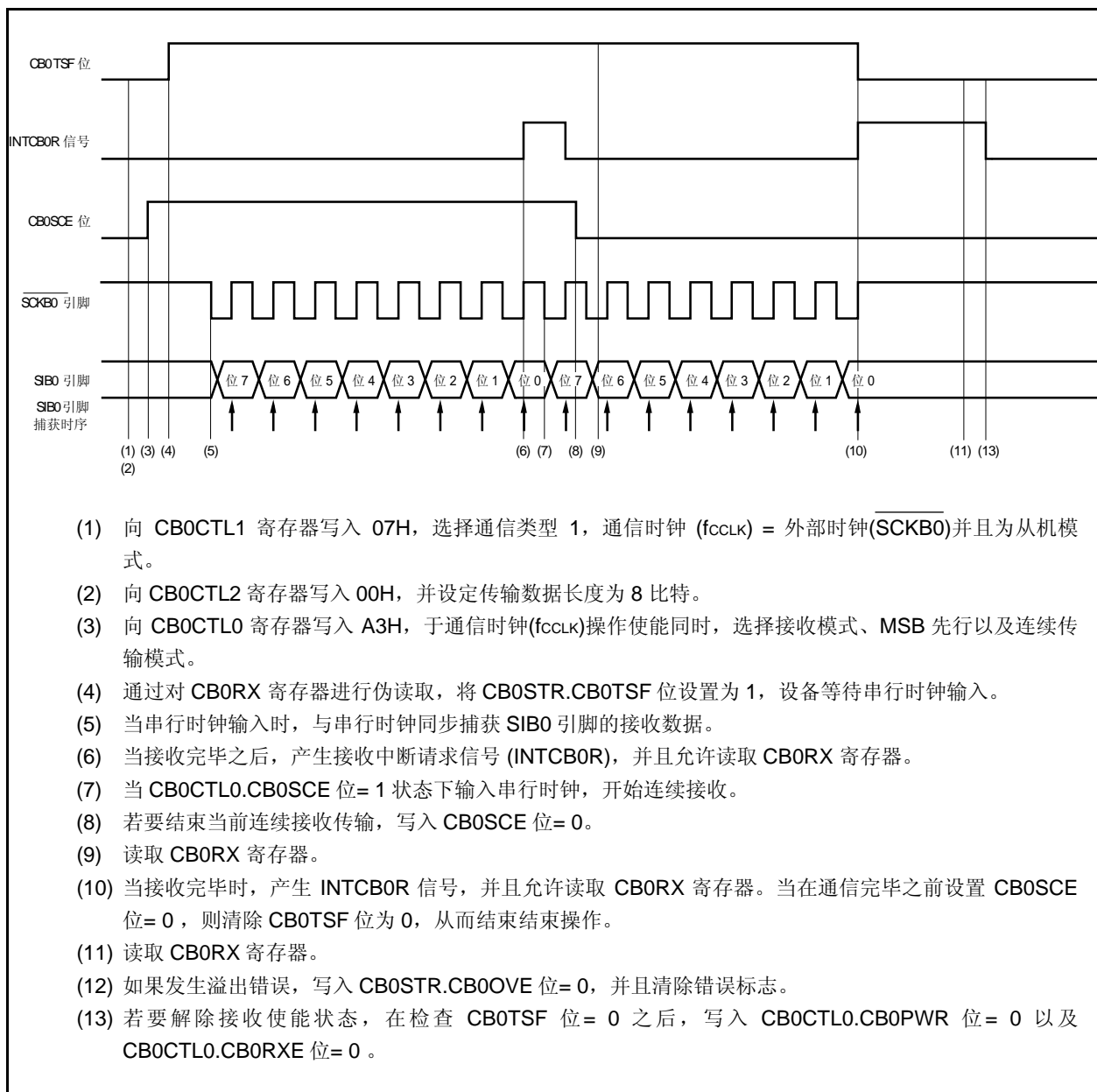
MSB 先行(CB0CTL0.CB0DIR 位= 0), 通信类型 1 (CB0CTL1.CB0CKP 和 CB0CTL1.CB0DAP 位= 00), 通信时钟 (f_{CCLK}) = 外部时钟(SCKB0) (CB0CTL1.CB0CKS2 至 CB0CTL1.CB0CKS0 位= 111), 传输数据长度 = 8 比特 (CB0CTL2.CB0CL3 至 CB0CTL2.CB0CL0 位= 0000)。

(1) 操作流程



备注 1. 虚线部分表示硬件处理过程。
2. 图中的数字与(2) 操作时序 中的处理数字相对应。

(2) 操作时序

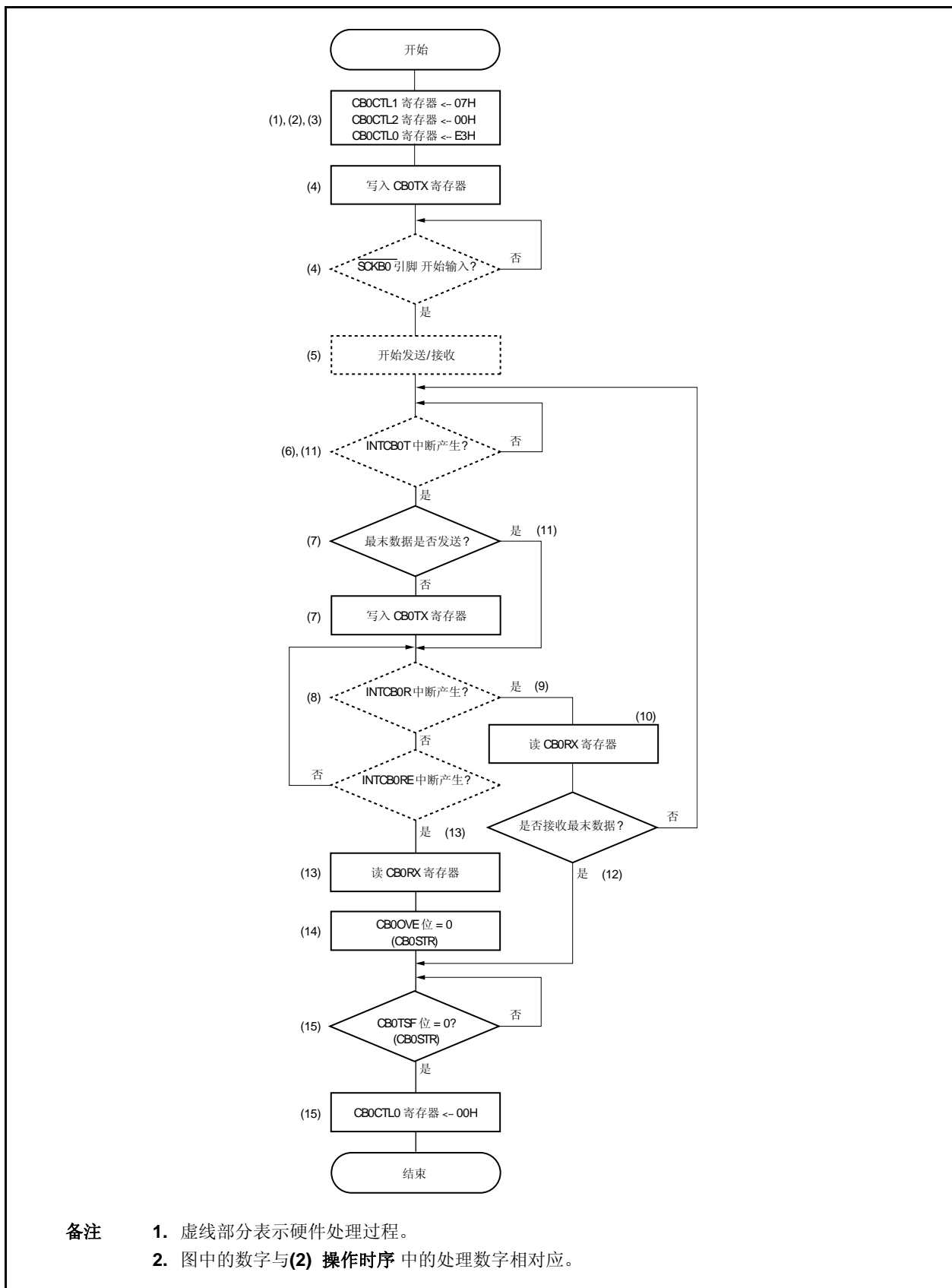


- (1) 向 CB0CTL1 寄存器写入 07H, 选择通信类型 1, 通信时钟 (f_{CCLK}) = 外部时钟(SCKB0)并且为从机模式。
- (2) 向 CB0CTL2 寄存器写入 00H, 并设定传输数据长度为 8 比特。
- (3) 向 CB0CTL0 寄存器写入 A3H, 于通信时钟(f_{CCLK})操作使能同时, 选择接收模式、MSB 先行以及连续传输模式。
- (4) 通过对 CB0RX 寄存器进行伪读取, 将 CB0STR.CB0TSF 位设置为 1, 设备等待串行时钟输入。
- (5) 当串行时钟输入时, 与串行时钟同步捕获 SIB0 引脚的接收数据。
- (6) 当接收完毕之后, 产生接收中断请求信号 (INTCB0R), 并且允许读取 CB0RX 寄存器。
- (7) 当 CB0CTL0.CB0SCE 位= 1 状态下输入串行时钟, 开始连续接收。
- (8) 若要结束当前连续接收传输, 写入 CB0SCE 位= 0。
- (9) 读取 CB0RX 寄存器。
- (10) 当接收完毕时, 产生 INTCB0R 信号, 并且允许读取 CB0RX 寄存器。当在通信完毕之前设置 CB0SCE 位= 0, 则清除 CB0TSF 位为 0, 从而结束操作。
- (11) 读取 CB0RX 寄存器。
- (12) 如果发生溢出错误, 写入 CB0STR.CB0OVE 位= 0, 并且清除错误标志。
- (13) 若要解除接收使能状态, 在检查 CB0TSF 位= 0 之后, 写入 CB0CTL0.CB0PWR 位= 0 以及 CB0CTL0.CB0RXE 位= 0。

13.4.12 连续传输模式 (从机模式, 发送/接收模式)

MSB 先行(CB0CTL0.CB0DIR 位= 0), 通信类型 1 (CB0CTL1.CB0CKP 和 CB0CTL1.CB0DAP 位= 00), 通信时钟 (f_{CCLK}) = 外部时钟 (SCKB0) (CB0CTL1.CB0CKS2 至 CB0CTL1.CB0CKS0 位 = 111), 传输长度 = 8 比特 (CB0CTL2.CB0CL3 至 CB0CTL2.CB0CL0 位 = 0000)。

(1) 操作流程

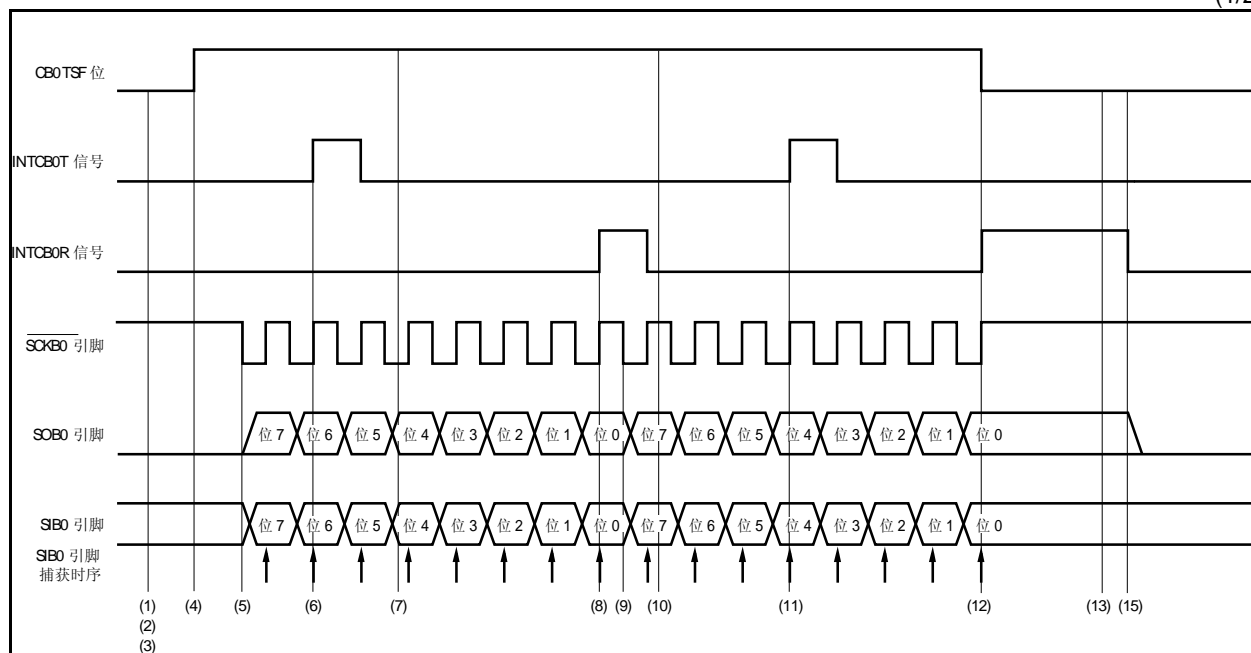


备注

1. 虚线部分表示硬件处理过程。
2. 图中的数字与(2) 操作时序 中的处理数字相对应。

(2) 操作时序

(1/2)



- (1) 向 CB0CTL1 寄存器写入 07H, 选择通信类型 1, 通信时钟(f_{cclk}) = 外部时钟 (SCKB0), 并且为从机模式。
- (2) 向 CB0CTL2 寄存器写入 00H, 并设定传输数据长度为 8 比特。
- (3) 向 CB0CTL0 寄存器写入 E3H, 并且, 与通信时钟(f_{cclk})使能同时选择发送/接收模式、MSB 先行模式以及连续传输模式。
- (4) 通过将发送数据写入 CB0TX 寄存器, 设置 CB0STR.CB0TSF 位为 1, 设备等待串行时钟输入。
- (5) 当串行时钟输入时, 与串行时钟同步, 输出发送数据至 SOB0 引脚, 并且, 捕获 SIB0 引脚的接收数据。
- (6) 当发送数据从 CB0TX 寄存器传输至移位寄存器完毕, 且允许写入 CB0TX 寄存器时, 产生发送允许中断请求信号 (INTCB0T)。
- (7) 若要连续发送, 在产生 INTCB0T 信号之后, 再次将发送数据写入 CB0TX 寄存器。
- (8) 当由 CB0CTL2 寄存器所设定的传输数据长度接收完毕后, 产生接收结束中断请求信号 (INTCB0R), 并且, 允许读取 CB0RX 寄存器。
- (9) 当串行时钟连续输入, 开始连续发送/接收。
- (10) 读取 CB0RX 寄存器。
- (11) 当发送数据从 CB0TX 寄存器传输至移位寄存器完毕, 并且, 允许写入 CB0TX 寄存器, 产生 INTCB0T 信号。若要结束当前的发送/接收传输, 则不写入 CB0TX 寄存器。

- (12) 当由 CB0CTL2 所设定的传输数据长度相应的时钟输入时，而不写入 CB0TX 寄存器，则产生 INTCB0R 信号。将 CB0TSF 位清除为 0 从而结束发送/接收。
- (13) 当产生接收错误中断请求信号(INTCB0RE)时，读取 CB0RX 寄存器。
- (14) 如果发生溢出错误，写入 CB0STR.CB0OVE 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 CB0TSF 位= 0 之后，写入 CB0CTL0.CB0PWR 位= 0，CB0CTL0.CB0TXE 位= 0 以及 CB0CTL0.CB0RXE 位= 0。

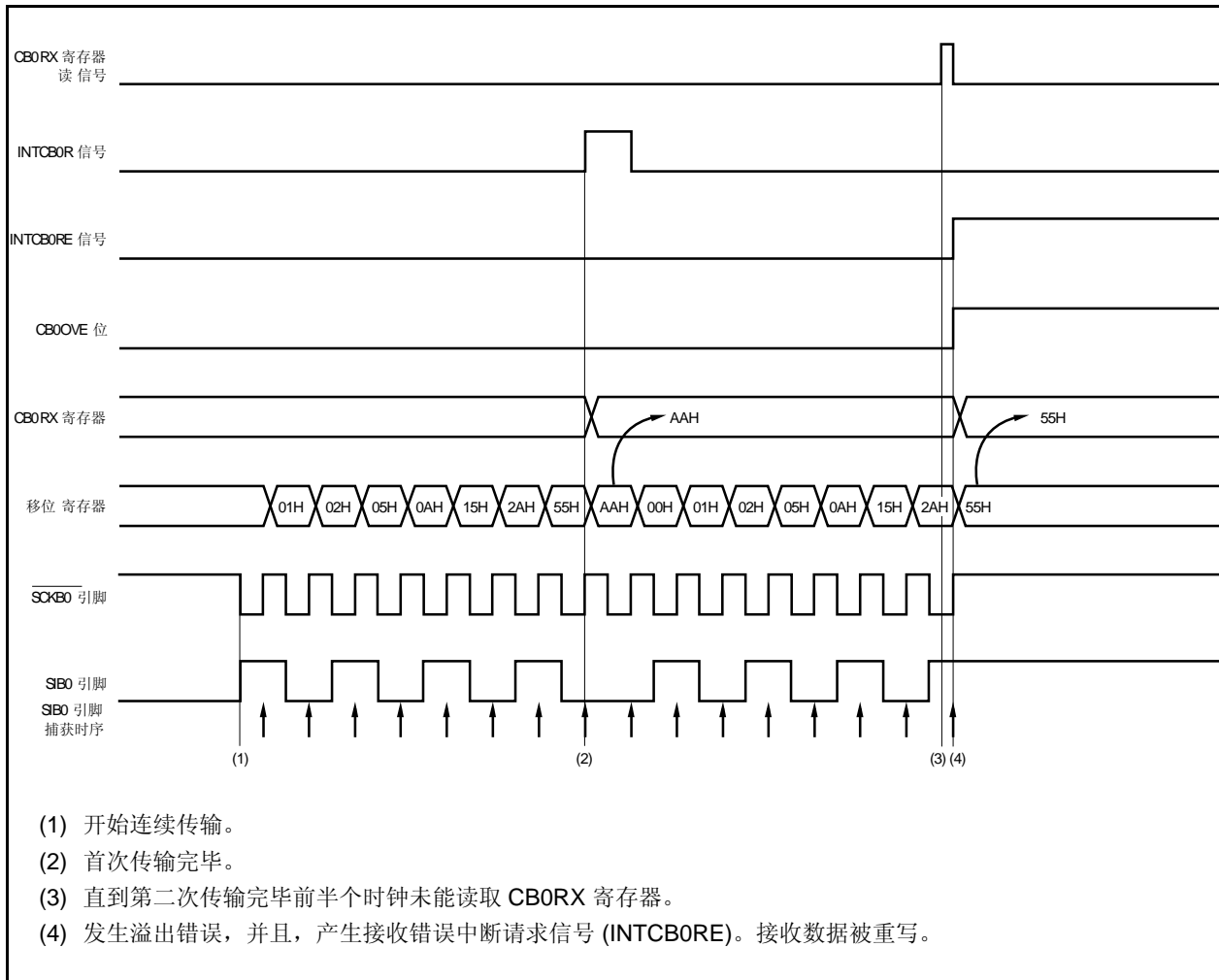
13.4.13 接收错误

在连续传输模式中，在接收允许(CB0CTL0.CB0RXE 位= 1)时执行传输，当接收结束中断请求信号(INTCB0R)产生后，在读取 CB0RX 寄存器之前进行下一次接收操作，则产生接收错误中断信号 (INTCB0RE)，并且，溢出错误标志 (CB0STR.CB0OVE)被设置为 1。

即使发生了溢出错误，由于 CB0RX 已经更新，先前的接收数据因而丢失。即使发生了溢出错误，如果 CB0RX 寄存器没被读取，则下次接收完毕时产生 INTCB0RE 信号。

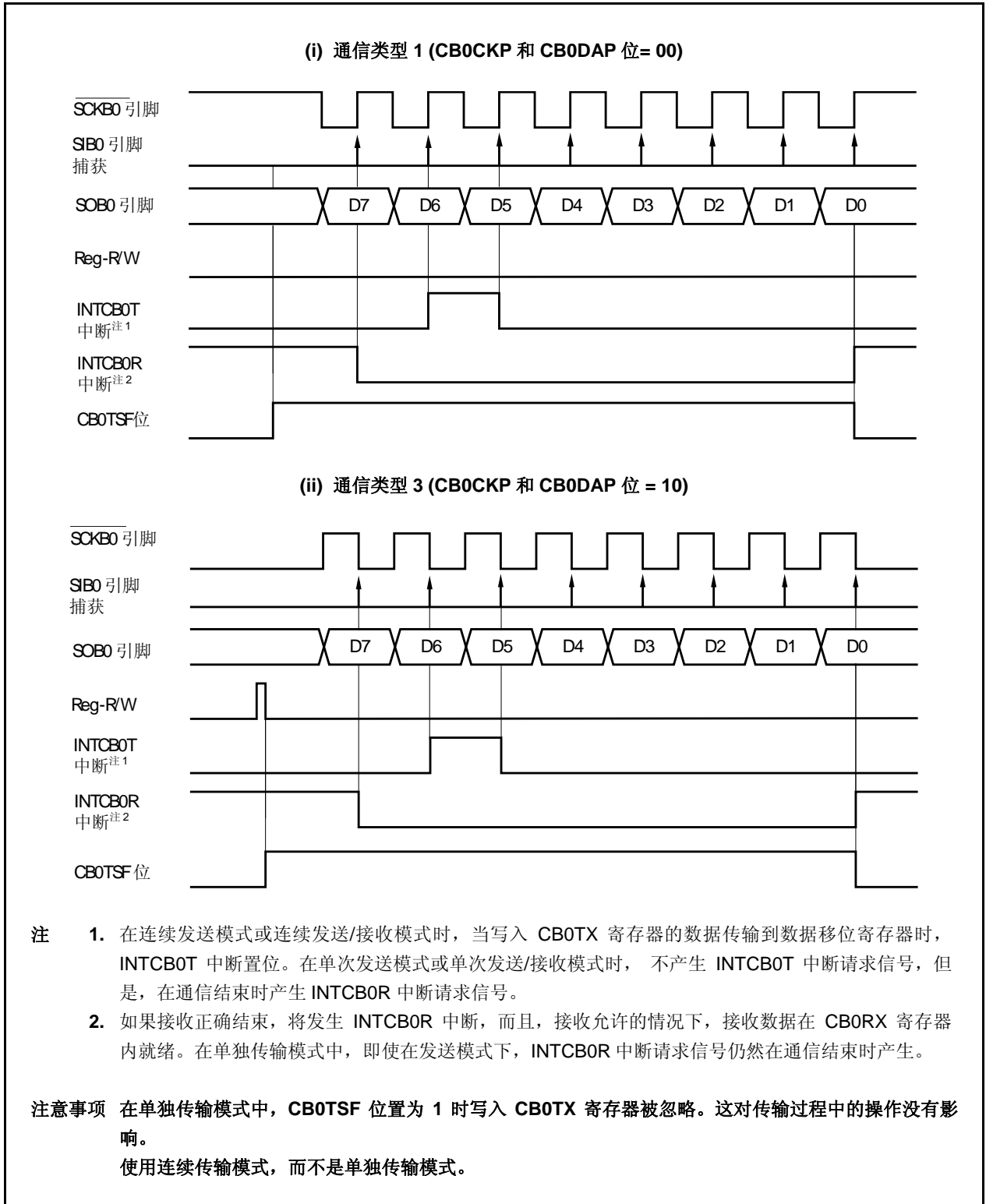
为了避免发送溢出错误，从 INTCB0R 信号产生起至采样下次接收最末位数据前半个时钟，应在此之前完成读取 CB0RX 寄存器。

(1) 操作时序

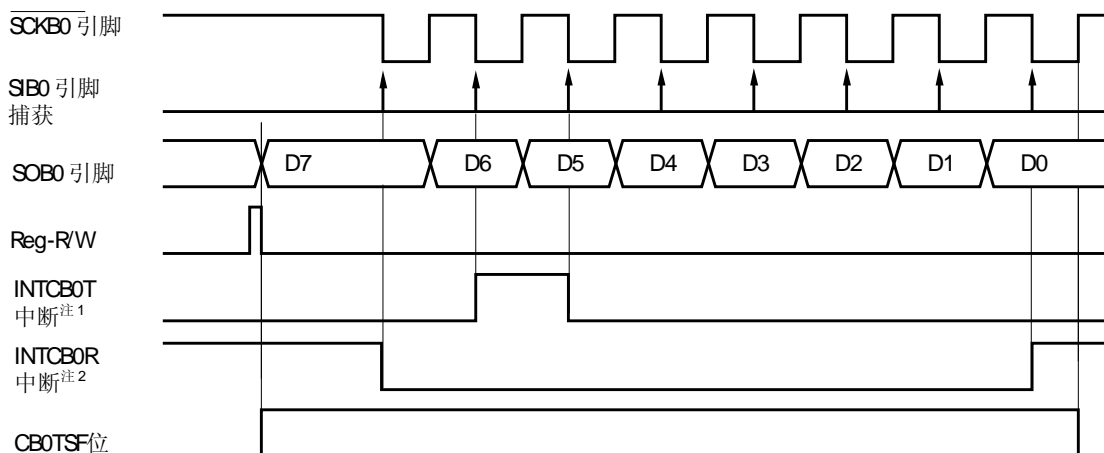


13.4.14 时钟时序

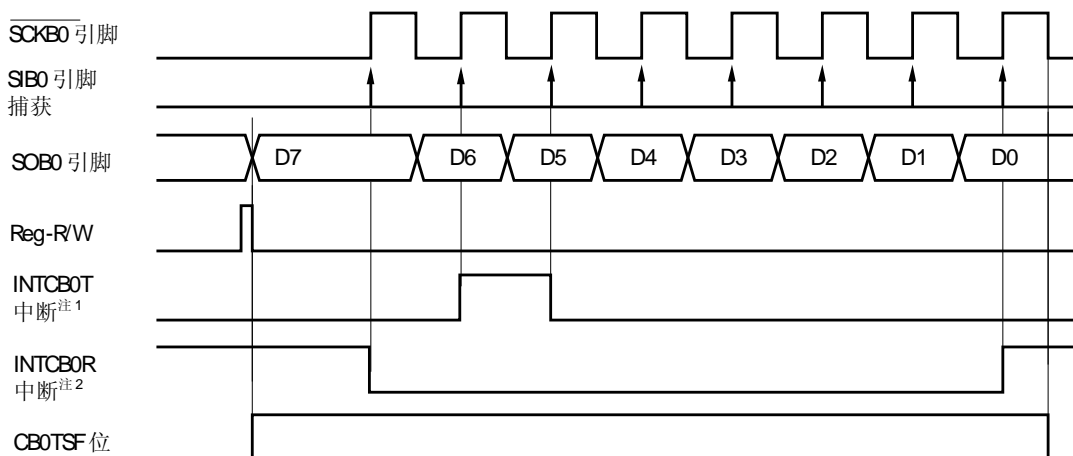
(1/2)



(iii) 通信类型 2 (CB0CKP 和 CB0DAP 位= 01)



(iv) 通信类型 4 (CB0CKP 和 CB0DAP 位 = 11)



- 注
1. 在连续发送模式或连续发送/接收模式时，当写入 CB0TX 寄存器的数据传输到数据移位寄存器时，INTCB0T 中断置位。在单次发送模式或单次发送/接收模式时，不产生 INTCB0T 中断请求信号，但是，在通信结束时产生 INTCB0R 中断请求信号。
 2. 如果接收正确结束，将发生 INTCB0R 中断，而且，接收允许的情况下，接收数据在 CB0RX 寄存器内就绪。在单次模式中，即使在发送模式下，INTCB0R 中断请求信号仍然在通信结束时产生。

注意事项 在单独传输模式中，CB0TSF 位置为 1 时写入 CB0TX 寄存器被忽略。这对传输过程中的操作没有影响。

使用连续传输模式，而不是单独传输模式。

13.5 输出引脚

(1) $\overline{\text{SCKB0}}$ 引脚

当 CSIB0 操作禁止 (CB0CTL0.CB0PWR 位= 0)， $\overline{\text{SCKB0}}$ 引脚输出状态如下所示。

CB0CKP	CB0CKS2	CB0CKS1	CB0CKS0	$\overline{\text{SCKB0}}$ 引脚输出
0	1	1	1	高阻。
	上述之外			固定高电平。
1	1	1	1	高阻。
	上述之外			固定低电平。

备注 如果 CB0CTL1.CB0CKP 和 CB0CKS2 至 CB0CKS0 位任何位被重新， $\overline{\text{SCKB0}}$ 引脚输出改变。

(2) SOB0 引脚

当 CSIB0 操作禁止 (CB0PWR 位= 0)，SOB0 引脚输出状态如下所示：

CB0TXE	CB0DAP	CB0DIR	SOB0 引脚输出
0	x	x	固定低电平。
1	0	x	SOB0 锁存值(低电平)。
	1	0	CB0TX0 值 (MSB)。
		1	CB0TX0 值 (LSB)。

- 备注**
1. 如果 CB0CTL0.CB0TXE，CB0CTL0.CB0DIR，或 CB0CTL1.CB0DAP 任何位被重写，SOB0 引脚输出改变。
 2. x: 无意义。

第十四章 中断/异常处理功能

V850ES/IE2 为中断服务提供专用的中断控制器 (INTC)，可以处理多达 43 个中断请求。

所谓中断，是其发生独立于程序执行的事件，而异常则是其发生与程序执行相关的事件。

V850ES/IE2 能够处理来自于外设和外部中断源的中断信号，不仅如此，异常处理可以由 TRAP 指令（软件异常）启动或者由异常事件启动（如：非法取指）（异常陷阱）。

14.1 特性

○ 中断

- 不可屏蔽中断: 1 个中断源 (外部: 无, 内部: 1 个中断源)
- 可屏蔽中断:
外部: 7 个中断源, 内部: 35 个中断源。
- 8 级可编程优先级 (可屏蔽中断)。
- 依照优先级多重中断控制。
- 可以为每一个可屏蔽中断指定屏蔽字。
- 噪声抑制、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 个产生源
- 异常陷阱: 2 个产生源 (非法取指陷阱和调试陷阱)

中断发生源列表如下:

表 14-1. 中断源列表 (1/3)

类型	类别	默认 优先级	名称	产生源	产生 单元	异常码	服务程序 句柄地址	恢复 PC	中断控制寄 存器
复位	中断	-	RESET	RESET 引脚输入	引脚	0000H	00000000H	未定义	-
				WDT 溢出 (WDTRES)	WDT				
				低电压检测(LVIRE)	POC/LVI				
不可屏蔽 中断	中断	-	INTWDT	WDT 溢出	WDT	0010H	0000010H	下一个PC值	-
软件异常	异常	-	TRAP0n ^{#1}	TRAP指令	-	004nH	0000040H	下一个PC值	-
	异常	-	TRAP1n ^{#1}	TRAP指令	-	005nH	0000050H	下一个PC值	-
异常陷阱	异常	-	ILGOP/ DBG0	无效指令代码/ DBTRAP指令	-	0060H	0000060H	下一个PC值	-
可屏蔽	中断	0	INTP0	INTP0 引脚有效边沿输入	引脚	0080H	00000080H	下一个PC值	PIC0
	中断	1	INTP1	INTP1 引脚有效边沿输入	引脚	0090H	00000090H	v	PIC1
	中断	2	INTP2	引脚	引脚	00A0H	000000A0H	下一个PC值	PIC2
	中断	3	INTP3	INTP3 引脚有效边沿输入	引脚	00B0H	000000B0H	下一个PC值	PIC3
	中断	4	INTP4	INTP4 引脚有效边沿输入	引脚	00C0H	000000C0H	下一个PC值	PIC4
	中断	5	INTP5	INTP5 引脚有效边沿输入	引脚	00D0H	000000D0H	下一个PC值	PIC5
	中断	6	INTP6	INTP6 引脚有效边沿输入	引脚	00E0H	000000E0H	下一个PC值	PIC6
	中断	7	INTLVI	低电压检测	POC/LVI	00F0H	000000F0H	下一个PC值	LVIIC
	中断	-	-	未使用	-	-	00000100H	-	-
	中断	-	-	未使用	-	-	00000110H	-	-
	中断	8	INTTQ0OV	TMQ0 溢出	TMQ0	0120H	00000120H	下一个PC值	TQ0OVIC
	中断	9	INTTQ0CC0	TQ0CCR0 捕获输入/ 比较匹配	TMQ0	0130H	00000130H	下一个PC值	TQ0CCIC0
	中断	10	INTTQ0CC1	TQ0CCR1 捕获输入/ 比较匹配	TMQ0	0140H	00000140H	下一个PC值	TQ0CCIC1
	中断	11	INTTQ0CC2	TQ0CCR2捕获输入/ 比较匹配	TMQ0	0150H	00000150H	下一个PC值	TQ0CCIC2
	中断	12	INTTQ0CC3	TQ0CCR3捕获输入/ 比较匹配	TMQ0	0160H	00000160H	下一个PC值	TQ0CCIC3
	中断	13	INTTQ1OV	TMQ1 溢出 ^{#2}	TMQ1	0170H	00000170H	下一个PC值	TQ1OVIC
	中断	14	INTTQ1CC0	TQ1CCR0比较匹配 ^{#3}	TMQ1	0180H	00000180H	下一个PC值	TQ1CCIC0
	中断	15	INTTQ1CC1	TQ1CCR1比较匹配	TMQ1	0190H	00000190H	下一个PC值	TQ1CCIC1
	中断	16	INTTQ1CC2	TQ1CCR2比较匹配	TMQ1	01A0H	000001A0H	下一个PC值	TQ1CCIC2
	中断	17	INTTQ1CC3	TQ1CCR3比较匹配	TMQ1	01B0H	000001B0H	下一个PC值	TQ1CCIC3
	中断	-	-	未使用	-	-	000001C0H	-	-
	中断	-	-	未使用	-	-	000001D0H	-	-

- 注
1. n取值范围为 0 至 FH。
 2. 当 TMQ1 用于 6 相 PWM 输出模式时，其功能为来自于 TMQ1 选项(TMQOP1)的 INTTQ1OV (谷底中断)。
 3. 当 TMQ1 用于 6 相 PWM 输出模式时，其功能为来自于 TMQ1 选项(TMQOP1)的 INTTQ1CC0 (峰顶中断)。

表 14-1. 中断源列表 (2/3)

类型	类别	默认 优先级	名称	产生源	产生 单元	异常码	服务程序 句柄地址	恢复 PC	中断控制寄 存器
可屏蔽	中断	-	-	未使用	-	-	000001E0H	-	-
	中断	-	-	未使用	-	-	000001F0H	-	-
	中断	-	-	未使用	-	-	00000200H	-	-
	中断	-	-	未使用	-	-	00000210H	-	-
	中断	-	-	未使用	-	-	00000220H	-	-
	中断	-	-	未使用	-	-	00000230H	-	-
	中断	18	INTTP0OV	TMP0 溢出	TMP0	0240H	00000240H	下一个PC值	TP0OVIC
	中断	19	INTTP0CC0	TP0CCR0 捕获输入/ 比较匹配	TMP0	0250H	00000250H	下一个PC值	TP0CCIC0
	中断	20	INTTP0CC1	TP0CCR1 捕获输入/ 比较匹配	TMP0	0260H	00000260H	下一个PC值	TP0CCIC1
	中断	21	INTTP1OV	TMP1 溢出	TMP1	0270H	00000270H	下一个PC值	TP1OVIC
	中断	22	INTTP1CC0	TP1CCR0 比较匹配	TMP1	0280H	00000280H	下一个PC值	TP1CCIC0
	中断	23	INTTP1CC1	TP1CCR1 比较匹配	TMP1	0290H	00000290H	下一个PC值	TP1CCIC1
	中断	24	INTTP2OV	TMP2 溢出	TMP2	02A0H	000002A0H	下一个PC值	TP2OVIC
	中断	25	INTTP2CC0	TP2CCR0 捕获输入/ 比较匹配	TMP2	02B0H	000002B0H	下一个PC值	TP2CCIC0
	中断	26	INTTP2CC1	TP2CCR1 捕获输入/ 比较匹配	TMP2	02C0H	000002C0H	下一个PC值	TP2CCIC1
	中断	27	INTTP3OV	TMP3 溢出	TMP3	02D0H	000002D0H	下一个PC值	TP3OVIC
	中断	28	INTTP3CC0	TP3CCR0 比较匹配	TMP3	02E0H	000002E0H	下一个PC值 (nextPC)	TP3CCIC0
	中断	29	INTTP3CC1	TP3CCR1 比较匹配	TMP3	02F0H	000002F0H	下一个PC值	TP3CCIC1
	中断	-	-	未使用	-	-	00000300H	-	-
	中断	-	-	未使用	-	-	00000310H	-	-
	中断	-	-	未使用	-	-	00000320H	-	-
	中断	-	-	未使用	-	-	00000330H	-	-
	中断	30	INTUA0RE	UARTA0 接收错误	UARTA0	0340H	00000340H	下一个PC值	UA0REIC
	中断	31	INTUA0R	UARTA0 接收结束	UARTA0	0350H	00000350H	下一个PC值	UA0RIC
	中断	32	INTUA0T	UARTA0 接收允许	UARTA0	0360H	00000360H	下一个PC值	UA0TIC
	中断	33	INTCB0RE	CSIB0 接收错误	CSIB0	0370H	00000370H	下一个PC值	CB0REIC
	中断	34	INTCB0R	CSIB0 发送/接收完毕	CSIB0	0380H	00000380H	下一个PC值	CB0RIC
	中断	35	INTCB0T	CSIB0 连续发送使能	CSIB0	0390H	00000390H	下一个PC值	CB0TIC
	中断	36	INTUA1RE	UARTA1 接收错误	UARTA1	03A0H	000003A0H	下一个PC值	UA1REIC
	中断	37	INTUA1R	UARTA1 接收结束	UARTA1	03B0H	000003B0H	下一个PC值	UA1RIC
	中断	38	INTUA1T	UARTA1 发送使能	UARTA1	03C0H	000003C0H	下一个PC值	UA1TIC

表 14-1. 中断源列表 (3/3)

类型	类别	默认 优先级	名称	产生源	产生 单元	异常码	服务程序 句柄地址	恢复 PC	中断控制 寄存器
可屏蔽	中断	-	-	未使用	-	-	000003D0H	-	-
	中断	-	-	未使用	-	-	000003E0H	-	-
	中断	-	-	未使用	-	-	000003F0H	-	-
	中断	39	INTAD0	ADC0 转换完毕	ADC0	0400H	00000400H	下一个PC值	AD0IC
	中断	40	INTAD1	ADC1 转换完毕	ADC1	0410H	00000410H	下一个PC值	AD1IC
	中断	-	-	未使用	-	-	00000420H	-	-
	中断	41	INTTM0EQ0	TM0CMP0 比较匹配	TMM0	0430H	00000430H	下一个PC值	TM0EQIC 0

- 备注**
- 默认优先级： 当同一时间产生两个或更多的可屏蔽中断请求时，中断优先级同时产生，最高优先级 0。
 存储的 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：当执行以下指令之一，程序响应一个不可屏蔽中断或可屏蔽中断时，则恢复 PC 不是下一个 PC 值。（如果在中断执行当中响应一个中断，执行停止，而之后在中断服务完成后恢复执行。这种情况下，被中止的指令其地址是恢复 PC。）
 - Load 指令 (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)。
 - 除法指令 (DIV, DIVH, DIVU, DIVHU)
 - PREPARE, DISPOSE 指令 (仅当在堆栈指针更新前产生中断)
 下一个 PC 值 (nextPC)： 中断/异常处理后随之由此进行处理的 PC 值。
 - 当代码非法异常发生时，非法指令的执行地址如此计算：(恢复 PC - 4)。

14.2 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无条件地得到响应。不可屏蔽中断并不遵从优先级控制，而且，比所有其他中断请求信号更为优先。

V850ES/IE2 的不可屏蔽中断信号是由看门狗定时器(INTWDT)所产生的不可屏蔽中断请求信号。

当 WDTM.WDM1 和 WDTM.WDM0 位设置为“01”时，INTWDT 功能有效。

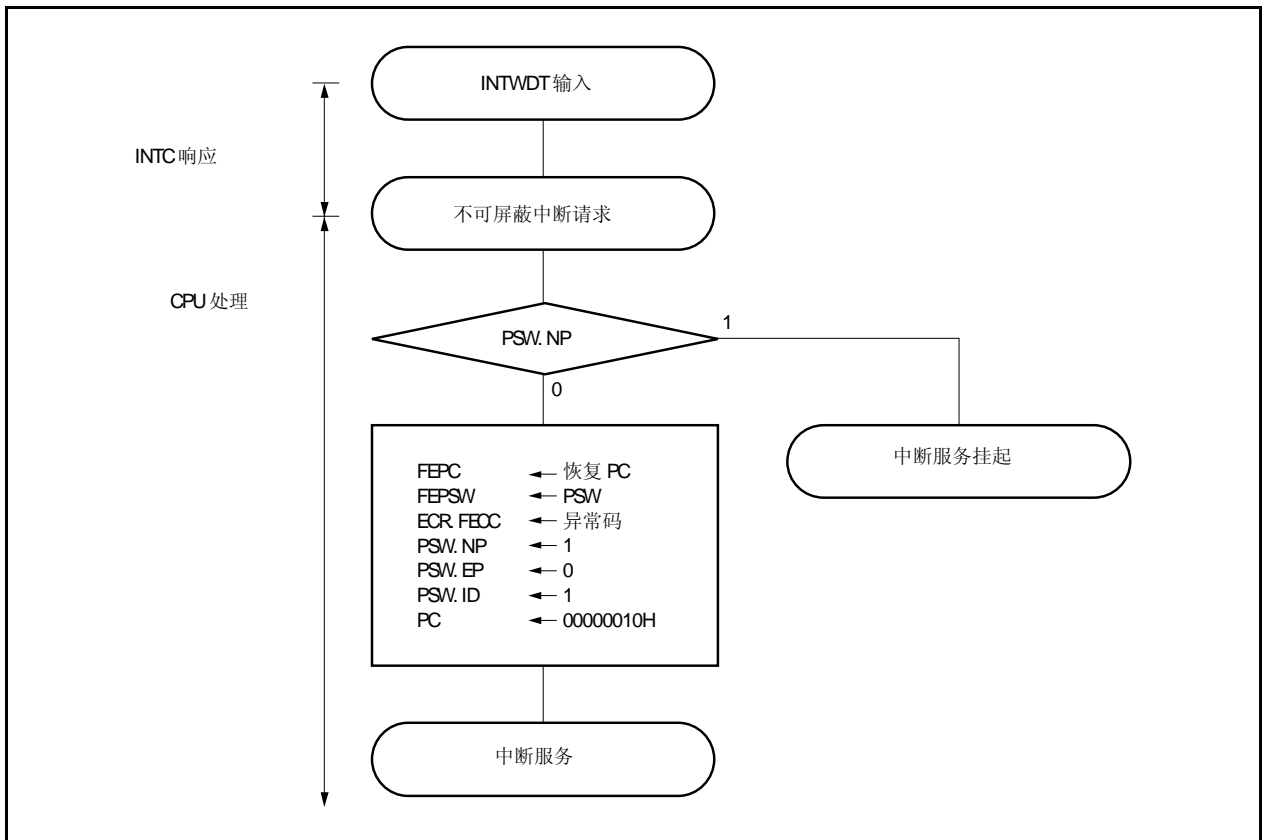
14.2.1 操作

如果一个不可屏蔽中断请求信号(INTWDT)产生，CPU 执行以下处理过程，并且，控制转移至句柄处理程序：

- (1) 将存储的 PC 保存至 FEPC。
- (2) 将当前 PSW 保存至 FEPSW。
- (3) 将异常码(0010H)写入 ECR 的高半字 (FECC)。
- (4) 置位 PSW.NP 和 PSW.ID 位(为 1) 并清除 PSW.EP 位(为 0)。
- (5) 载入不可屏蔽中断处理程序句柄地址(00000010H) 至 PC，并且，控制转移。

以下显示不可屏蔽中断 的服务处理：

图 14-1. 不可屏蔽中断服务处理



14.2.2 恢复

通过 RETI 指令从不可屏蔽中断服务中恢复执行。

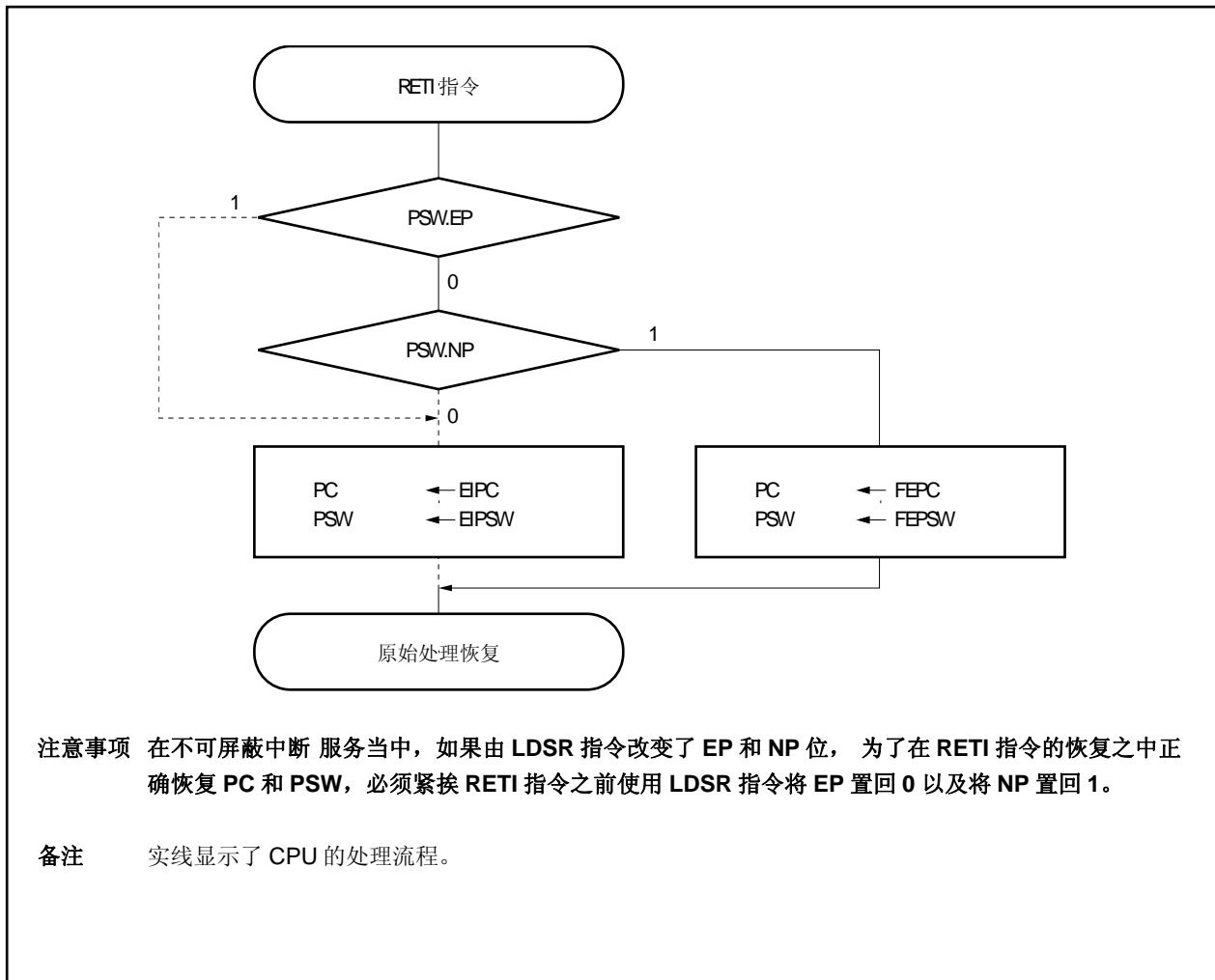
当 RETI 指令执行时，CPU 执行以下处理过程，并且，控制转移到恢复 PC 的地址。

<1> 由于 PSW.EP 位为 0 以及 PSW.NP 位为 1，因此，由 FEPC 和 FEPSW 恢复各自恢复 PC 和 PSW。

<2> 控制转移回到恢复的 PC 以及 PSW 的地址。

以下介绍 RETI 指令如何处理：

图 14-2. RETI 指令处理过程

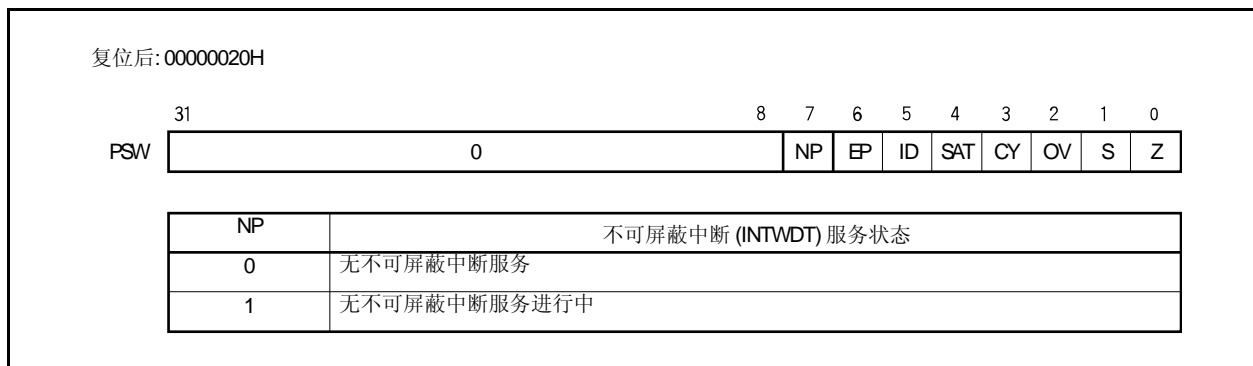


注意事项 在不可屏蔽中断服务当中，如果由 LDSR 指令改变了 EP 和 NP 位，为了在 RETI 指令的恢复之中正确恢复 PC 和 PSW，必须紧挨 RETI 指令之前使用 LDSR 指令将 EP 置回 0 以及将 NP 置回 1。

备注 实线显示了 CPU 的处理流程。

14.2.3 不可屏蔽中断状态标志

NP 标志为一个状态标志，它指示不可屏蔽中断 (INTWDT)服务正在处理之中。NP 标志位于 PSW 之内。
 当 INTWDT 中断信号被响应之后，本标志置位，并且，屏蔽所有中断及异常请求，以便禁止响应多重中断。
 复位之后本标志被清除为 00000020H。



14.3 可屏蔽中断

可屏蔽中断请求信号可由中断控制寄存器进行屏蔽，V850ES/IE2 具有 42 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，按照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，使得其它中断服务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。注意，只有具有较高优先级的中断才有这种能力；具有相同优先级等级的中断不能用作多重中断。

而且，为了允许多重中断服务，在执行 EI 指令前先将 EIPC 和 EIPSW 存入内存或寄存器中，并在使用 RETI 指令恢复 EIPC 和 EIPSW 原始值前执行 DI 指令。

14.3.1 操作

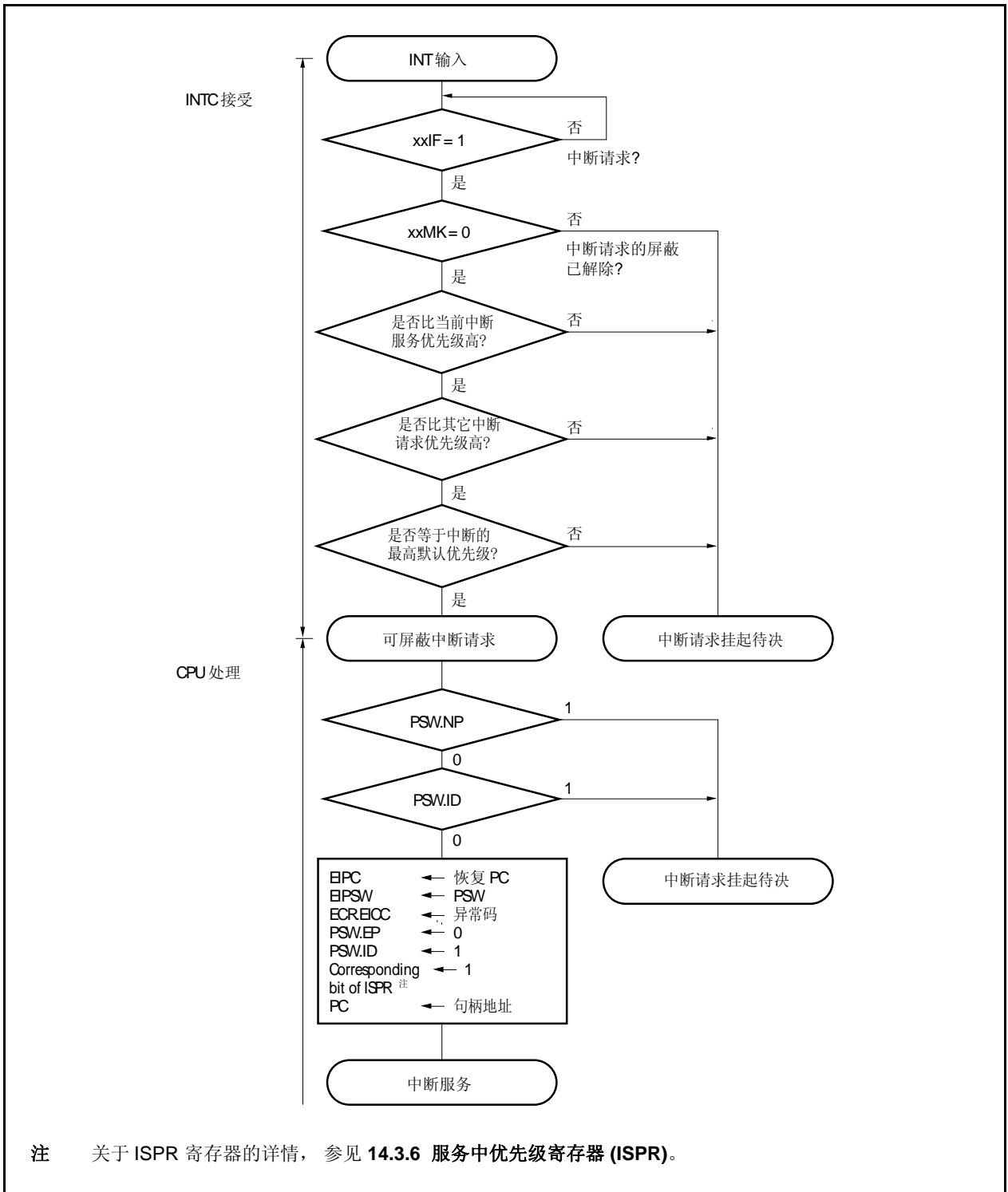
如果发生可屏蔽中断，CPU 执行以下处理过程，并控制转移至处理程序句柄。

- <1> 保存恢复 PC 至 EIPC。
- <2> 保存当前 PSW 至 EIPSW。
- <3> 将异常码写入 ECR (EICC)的低半字。
- <4> 置位 PSW.ID 位为 1 并清除 PSW.EP 位为 0。
- <5> 设置对应于 PC 各中断的处理程序句柄，并进行控制转移。

由中断控制寄存器(INTC)所屏蔽的可屏蔽中断请求信号，和于另外中断服务之中(当 PSW.NP 位 = 1 或 ID 位 = 1)时产生的可屏蔽中断请求信号，在 INTC 内保持未决状态。这种情况下，如果可屏蔽中断被取消屏蔽或使用 RETI 或 LDSR 指令将 NP 或 ID 清除为 0，将根据未决的可屏蔽中断请求之优先级来启动一个新的可屏蔽中断。

以下介绍可屏蔽中断的服务过程：

图 14-3. 可屏蔽中断服务过程



14.3.2 恢复

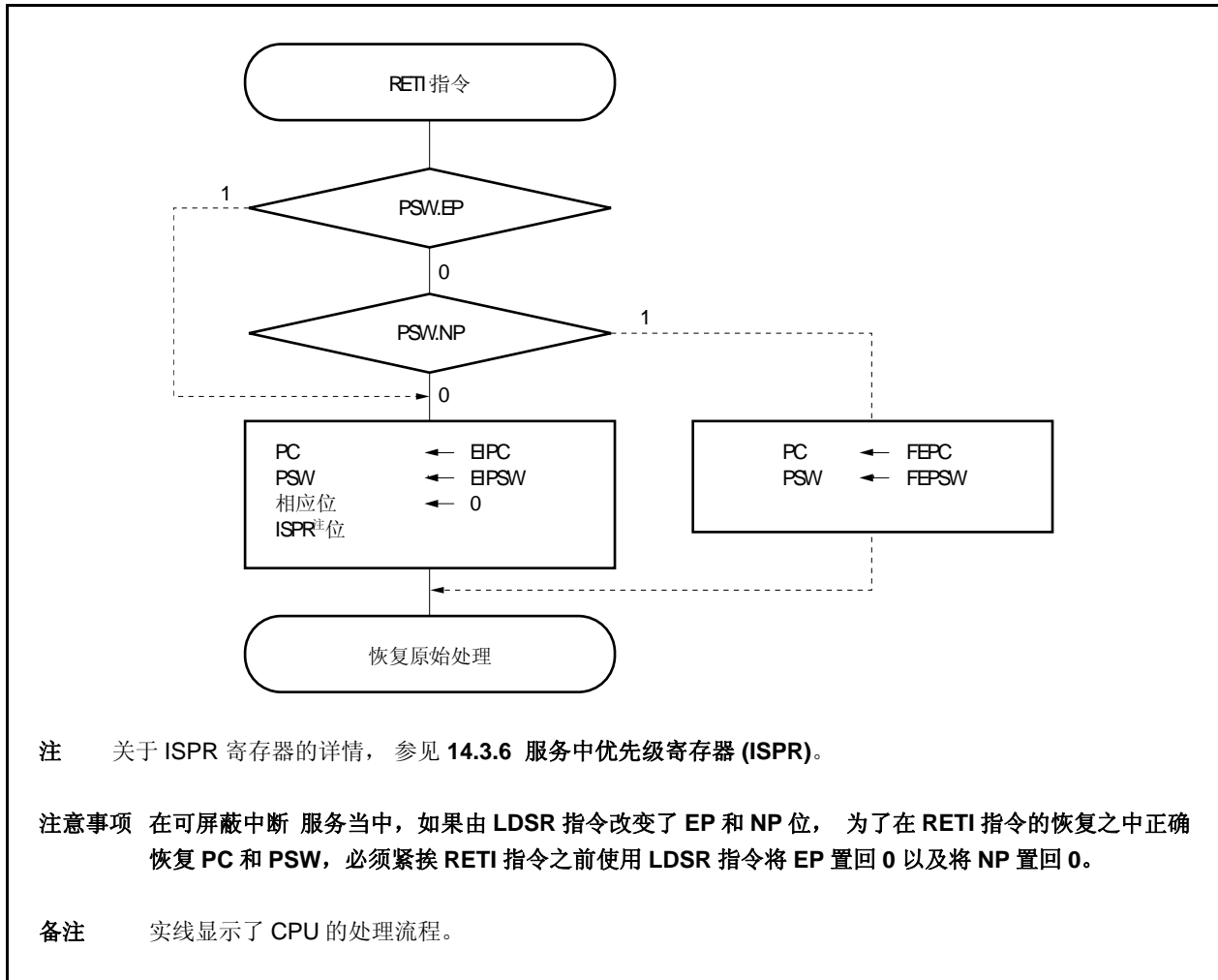
从可屏蔽中断服务中恢复，由 RETI 指令完成。

当执行 RETI 指令时，CPU 执行以下步骤，并且，控制转移至恢复 PC 的地址。

- <1> 由于 PSW.EP 位为 0 以及 PSW.NP 位为 0，由 EIPC 和 EIPSW 恢复 PC 和 PSW 的值。
- <2> 控制转移返回到恢复 PC 和 PSW 的地址。

RETI 指令的处理过程如下所示：

图 14-4. RETI 指令处理过程



14.3.3 可屏蔽中断的优先级

INTC 提供多重中断服务，多重中断中在一个中断处于服务之际响应另一个中断。多重中断可由优先等级进行控制。

有两种优先等级控制类型：基于默认优先等级控制以及基于由中断控制寄存器(xxICn)的(xxPRn)位所指定的可编程优先等级控制。当两个或更多的具有由 xxPRn 位所指定的相同优先级的中断同时发生，中断将按预先分配给各自请求信号的类型优先级（默认优先级）次序进行服务。更多信息参见 **表 14-1 中断源列表**。通过设置中断优先级指定标志，可编程优先控制将中断请求信号定制为 8 个等级。

注意，当一个中断请求信号得到响应，PSW.ID 标志自动设置为 1。所以，当使用多重中断时，预先将 ID 标志清为 0（如：通过在中断服务程序中安排 EI 指令）从而设置中断允许模式。

备注 xx: 各个外设单元的标识名 (参见 **表 14-2**)
 n: 外设单元号 (参见 **表 14-2**)

图 14-5. 一个中断服务当中而另一个中断请求发生的处理举例(1/2)

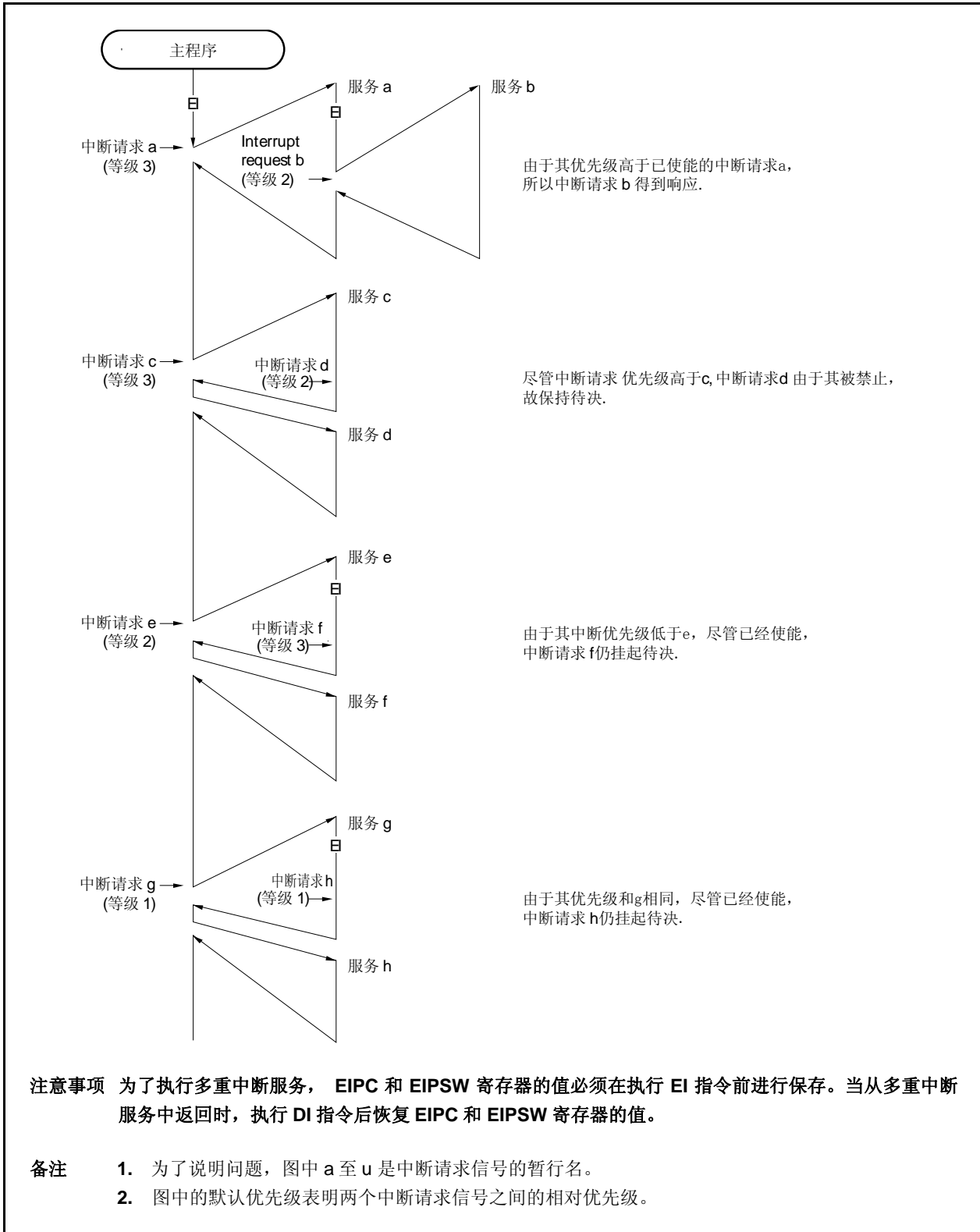


图 14-5. 一个中断服务当中而另一个中断请求发生的处理举例(2/2)

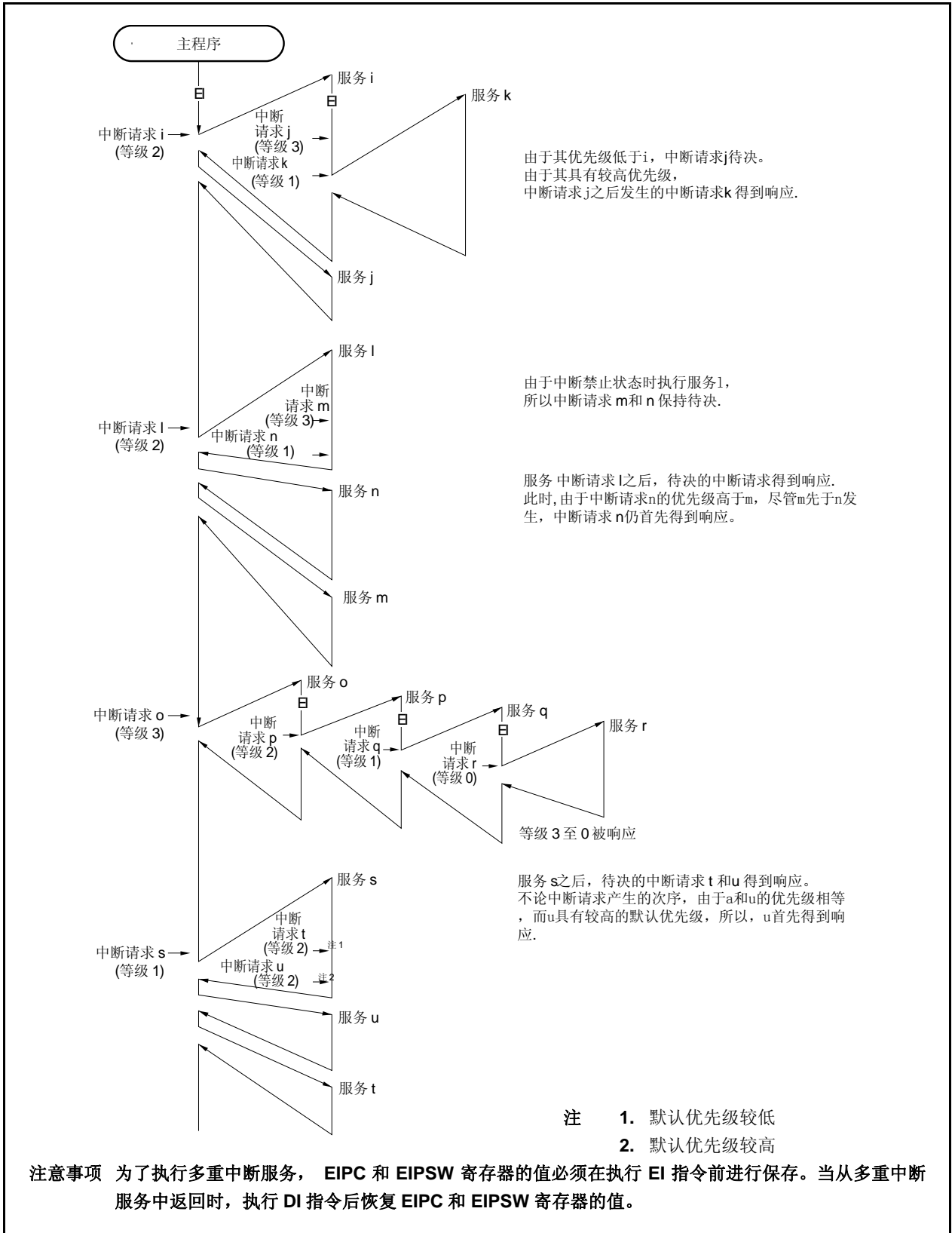
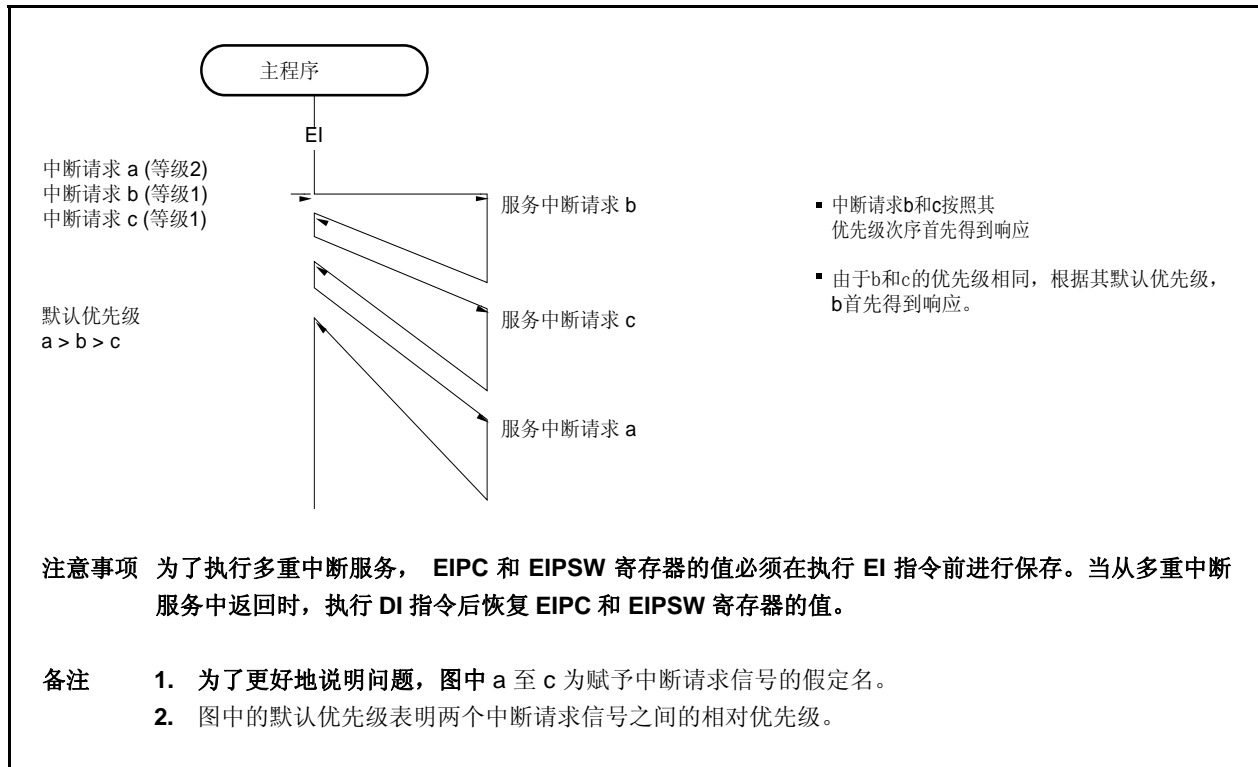


图 14-6. 对同时发生的中断请求信号服务举例



14.3.4 中断控制寄存器 (xxICn)

每一个中断请求信号(可屏蔽中断)都被赋予一个 xxICn 寄存器，用来设置每一个可屏蔽中断请求的控制条件。这些寄存器可以按 8 位或 16 位进行读写。
复位后这些寄存器设为 47H。

- 注意事项**
1. 读取 xxICn.xxIFn 位时禁止中断(DI)。如果在中断允许(EI)的情况下读取 xxIFn 位，当响应一个中断与读取该位相冲突时，可能读不出正确值。
 2. 当在有可能发生中断请求（包括中断被禁止(DI)状态下）时处理 xxICn.xxMKn 位，务必使用位处理指令或使用 IMRm (m = 0 to 3)寄存器的 xxMKn 位。

<R>

复位后: 47H R/W 地址: FFFFF110H to FFFFF186H

	<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	无中断请求
1	有中断请求

xxMKn	中断屏蔽标志
0	中断服务使能
1	中断服务禁止(挂起)

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: 各个外设单元的标识名(参见表 14-2)
 n: 外设单元号 (参见表 14-2)

中断控制寄存器的地址及各位如下所示:

表 14-2. 中断控制寄存器的地址及各位 (1/2)

地址	寄存器	位							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF112H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF114H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF116H	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
FFFFF118H	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
FFFFF11AH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
FFFFF11CH	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60
FFFFF11EH	LVIIIC	LVIIIF	LVIMK	0	0	0	LVIPR2	LVIPR1	LVIPR0
FFFFF120H	未使用	-	-	-	-	-	-	-	-
FFFFF122H	未使用	-	-	-	-	-	-	-	-
FFFFF124H	TQ0OVIC	TQ0OVIF	TQ0OVMK	0	0	0	TQ0OVPR2	TQ0OVPR1	TQ0OVPR0
FFFFF126H	TQ0CCIC0	TQ0CCIF0	TQ0CCMK0	0	0	0	TQ0CCPR02	TQ0CCPR01	TQ0CCPR00
FFFFF128H	TQ0CCIC1	TQ0CCIF1	TQ0CCMK1	0	0	0	TQ0CCPR12	TQ0CCPR11	TQ0CCPR10
FFFFF12AH	TQ0CCIC2	TQ0CCIF2	TQ0CCMK2	0	0	0	TQ0CCPR22	TQ0CCPR21	TQ0CCPR20
FFFFF12CH	TQ0CCIC3	TQ0CCIF3	TQ0CCMK3	0	0	0	TQ0CCPR32	TQ0CCPR31	TQ0CCPR30
FFFFF12EH	TQ1OVIC	TQ1OVIF	TQ1OVMK	0	0	0	TQ1OVPR2	TQ1OVPR1	TQ1OVPR0
FFFFF130H	TQ1CCIC0	TQ1CCIF0	TQ1CCMK0	0	0	0	TQ1CCPR02	TQ1CCPR01	TQ1CCPR00
FFFFF132H	TQ1CCIC1	TQ1CCIF1	TQ1CCMK1	0	0	0	TQ1CCPR12	TQ1CCPR11	TQ1CCPR10
FFFFF134H	TQ1CCIC2	TQ1CCIF2	TQ1CCMK2	0	0	0	TQ1CCPR22	TQ1CCPR21	TQ1CCPR20
FFFFF136H	TQ1CCIC3	TQ1CCIF3	TQ1CCMK3	0	0	0	TQ1CCPR32	TQ1CCPR31	TQ1CCPR30
FFFFF138H	未使用	-	-	-	-	-	-	-	-
FFFFF13AH	未使用	-	-	-	-	-	-	-	-
FFFFF13CH	未使用	-	-	-	-	-	-	-	-
FFFFF13EH	未使用	-	-	-	-	-	-	-	-
FFFFF140H	未使用	-	-	-	-	-	-	-	-
FFFFF142H	未使用	-	-	-	-	-	-	-	-
FFFFF144H	未使用	-	-	-	-	-	-	-	-
FFFFF146H	未使用	-	-	-	-	-	-	-	-
FFFFF148H	TP0OVIC	TP0OVIF	TP0OVMK	0	0	0	TP0OVPR2	TP0OVPR1	TP0OVPR0
FFFFF14AH	TP0CCIC0	TP0CCIF0	TP0CCMK0	0	0	0	TP0CCPR02	TP0CCPR01	TP0CCPR00
FFFFF14CH	TP0CCIC1	TP0CCIF1	TP0CCMK1	0	0	0	TP0CCPR12	TP0CCPR11	TP0CCPR10
FFFFF14EH	TP1OVIC	TP1OVIF	TP1OVMK	0	0	0	TP1OVPR2	TP1OVPR1	TP1OVPR0
FFFFF150H	TP1CCIC0	TP1CCIF0	TP1CCMK0	0	0	0	TP1CCPR02	TP1CCPR01	TP1CCPR00
FFFFF152H	TP1CCIC1	TP1CCIF1	TP1CCMK1	0	0	0	TP1CCPR12	TP1CCPR11	TP1CCPR10
FFFFF154H	TP2OVIC	TP2OVIF	TP2OVMK	0	0	0	TP2OVPR2	TP2OVPR1	TP2OVPR0
FFFFF156H	TP2CCIC0	TP2CCIF0	TP2CCMK0	0	0	0	TP2CCPR02	TP2CCPR01	TP2CCPR00
FFFFF158H	TP2CCIC1	TP2CCIF1	TP2CCMK1	0	0	0	TP2CCPR12	TP2CCPR11	TP2CCPR10
FFFFF15AH	TP3OVIC	TP3OVIF	TP3OVMK	0	0	0	TP3OVPR2	TP3OVPR1	TP3OVPR0
FFFFF15CH	TP3CCIC0	TP3CCIF0	TP3CCMK0	0	0	0	TP3CCPR02	TP3CCPR01	TP3CCPR00
FFFFF15EH	TP3CCIC1	TP3CCIF1	TP3CCMK1	0	0	0	TP3CCPR12	TP3CCPR11	TP3CCPR10
FFFFF160H	未使用	-	-	-	-	-	-	-	-

表 14-2. 中断控制寄存器的地址及各位(2/2)

地址	寄存器	位							
		<7>	<6>	5	4	3	2	1	0
FFFFF162H	未使用	-	-	-	-	-	-	-	-
FFFFF164H	未使用	-	-	-	-	-	-	-	-
FFFFF166H	未使用	-	-	-	-	-	-	-	-
FFFFF168H	UA0REIC	UA0REIF	UA0REMK	0	0	0	UA0REPR2	UA0REPR1	UA0REPR0
FFFFF16AH	UA0RIC	UA0RIF	UA0RMK	0	0	0	UA0RPR2	UA0RPR1	UA0RPR0
FFFFF16CH	UA0TIC	UA0TIF	UA0TMK	0	0	0	UA0TPR2	UA0TPR1	UA0TPR0
FFFFF16EH	CB0REIC	CB0REIF	CB0REMK	0	0	0	CB0REPR2	CB0REPR1	CB0REPR0
FFFFF170H	CB0RIC	CB0RIF	CB0RMK	0	0	0	CB0RPR2	CB0RPR1	CB0RPR0
FFFFF172H	CB0TIC	CB0TIF	CB0TMK	0	0	0	CB0TPR2	CB0TPR1	CB0TPR0
FFFFF174H	UA1REIC	UA1REIF	UA1REMK	0	0	0	UA1REPR2	UA1REPR1	UA1REPR0
FFFFF176H	UA1RIC	UA1RIF	UA1RMK	0	0	0	UA1RPR2	UA1RPR1	UA1RPR0
FFFFF178H	UA1TIC	UA1TIF	UA1TMK	0	0	0	UA1TPR2	UA1TPR1	UA1TPR0
FFFFF17AH	未使用	-	-	-	-	-	-	-	-
FFFFF17CH	未使用	-	-	-	-	-	-	-	-
FFFFF17EH	未使用	-	-	-	-	-	-	-	-
FFFFF180H	AD0IC	AD0IF	AD0MK	0	0	0	AD0PR2	AD0PR1	AD0PR0
FFFFF182H	AD1IC	AD1IF	AD1MK	0	0	0	AD1PR2	AD1PR1	AD1PR0
FFFFF184H	未使用	-	-	-	-	-	-	-	-
FFFFF186H	TM0EQIC0	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR02	TM0EQPR01	TM0EQPR00

14.3.5 中断屏蔽寄存器 0 至 3 (IMR0 至 IMR3)

IMR0 至 IMR3 寄存器设置可屏蔽中断的中断屏蔽状态。IMR0.xxMKn 至 IMR3.xxMKn 位等同于 xxICn.xxMKn 位。

IMRm 寄存器 (m = 0 至 3) 可以按 16 位为单位读取或写入。

如果 IMRm 寄存器的高 8 位用作 IMRmH 寄存器，而其低 8 位用作 IMRmL 寄存器，则这些寄存器可以按 8 位或按 1 位为单位进行读写。

复位后这些寄存器设置为 FFFFH。

注意事项 设备文件中将 xxICn.xxMKn 位定义为保留字。如果使用 xxMKn 处理一位，将是 xxICn 寄存器的内容而不是 IMRm 寄存器被改写(结果，IMRm 寄存器的内容也被改写)。

复位后: FFFFH R/W 地址: IMR3 FFFF106H
IMR3L FFFF106H, IMR3H FFFF107H

	15	14	13	12	11	10	9	8
IMR3 (IMR3H注)	1	1	1	1	TM0EQMK0	1	AD1MK	AD0MK
	7	6	5	4	3	2	1	0
(IMR3L)	1	1	1	UA1TMK	UA1RMK	UA1REMK	CB0TMK	CB0RMK

复位后: FFFFH R/W 地址: IMR2 FFFF104H
IMR2L FFFF104H, IMR2H FFFF105H

	15	14	13	12	11	10	9	8
IMR2 (IMR2H注)	CB0REMK	UA0TMK	UA0RMK	UA0REMK	1	1	1	1
	7	6	5	4	3	2	1	0
(IMR2L)	TP3CCMK1	TP3CCMK0	TP3OVMK	TP2CCMK1	TP2CCMK0	TP2OVMK	TP1CCMK1	TP1CCMK0

复位后: FFFFH R/W 地址: IMR1 FFFF102H
IMR1L FFFF102H, IMR1H FFFF103H

	15	14	13	12	11	10	9	8
IMR1 (IMR1H注)	TP1OVMK	TP0CCMK1	TP0CCMK0	TP0OVMK	1	1	1	1
	7	6	5	4	3	2	1	0
(IMR1L)	1	1	1	1	TQ1CCMK3	TQ1CCMK2	TQ1CCMK1	TQ1CCMK0

复位后: FFFFH R/W 地址: IMR0 FFFF100H
IMR0L FFFF100H, IMR0H FFFF101H

	15	14	13	12	11	10	9	8
IMR0 (IMR0H注)	TQ1OVMK	TQ0CCMK3	TQ0CCMK2	TQ0CCMK1	TQ0CCMK0	TQ0OVMK	1	1
	7	6	5	4	3	2	1	0
(IMR0L)	LVIMK	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0

xxMKn	中断屏蔽标志设定
0	中断服务允许
1	中断服务禁止

注 当按 8 位或 1 位为单位读取/写入 IMR0 至 IMR3 寄存器的位 15 至 8 时, 指定这些位为 IMR0H 至 IMR3H 寄存器的位 7 至 0。

注意事项 设置 IMR0 寄存器的位 9 和 8 (IMR0H 寄存器的位 1 和 0)、IMR1 寄存器的位 11 至 4 (IMR1H 寄存器的位 3 至 0 以及 IMR1L 寄存器的位 7 至 4)、IMR2 寄存器的位 11 至 8 (IMR2H 寄存器的位 3 至 0)、IMR3 寄存器的位 15 至 12、位 10 以及位 7 至 5 (IMR3H 寄存器的位 7 至 4、位 2 以及 IMR3L 寄存器的位 7 至 5) 为 1。当这些设置发生改变, 则操作无法保证。

备注 xx: 各个外设单元的标识名 (参见表 14-2)
n: 外设单元号 (参见表 14-2)

14.3.6 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，本寄存器与该中断请求信号的优先级相对应的位设置为 1，并且，在中断服务期间一直保持。

当执行 RETI 指令时，与具有最高优先级的中断请求信号相对应的位由硬件自动清 0。然而，当由不可屏蔽中断服务或异常处理中返回时，并不清 0。

本寄存器为只读寄存器，按 8 位或 1 位为单位读取。

复位后本寄存器设置为 00H。

注意事项 在中断使能(EI) 状态下，如果在读取 ISPR 寄存器期间响应一个中断，ISPR 的读取值可能为由此中断响应将该位设为 1 后的值，为了在中断响应前读取 ISPR 寄存器的正确值，应在中断禁止(DI) 状态下读取。

复位后:00H R 地址:FFFF1FAH

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	当前响应中断的优先级
0	具有优先级n的中断请求未得到响应
1	具有优先级n的中断请求正在响应

备注 n: 0 至 7 (优先等级)

14.4 外部中断请求输入引脚 (INTP0 至 INTP6)

14.4.1 噪声抑制

(1) INTP0 至 INTP5 引脚的噪声抑制

INTP0 至 INTP5 引脚具有使用模拟延迟的噪声抑制器，因此，各个引脚的输入电平需保持一定长的时间，否则，其边沿将不能被检测到。只有经过一定的时间之后，才检测到边沿。

(2) INTP6 引脚的噪声抑制

INTP6 引脚集成有一个数字噪声抑制器。

负责数字采样的采样时钟可以从 $f_{xx}/2$ 、 $f_{xx}/4$ 、 $f_{xx}/8$ 和 $f_{xx}/16$ 之中选择。

在 IDLE 和 STOP 模式时，系统时钟停止，所以，INTP6 引脚不能用来取消 IDLE 和 STOP 模式。

(a) 外部中断噪声抑制控制寄存器 (INTPNRC)

INTPNRC 寄存器用来选择采样时钟，该时钟用作消除 INTP6 引脚上的数字噪声。如果不能接连 5 次检测到相同电平，则该信号作为噪声被消除。

本寄存器可以按 8 位或 1 位为单位进行读写。

复位后本寄存器被设置为 00H。

- 注意事项**
1. 如果输入脉冲持续 4 至 5 个时钟周期，则不能确定其是被作为有效边沿或是被当作噪声消除。所以，脉冲若要确实被检测为有效边沿，其相同脉冲电平输入必须保持 5 个时钟周期或更长。
 2. 如果噪声是和采样时钟同步产生，于输入引脚接入一个滤波器以抑制噪声。
 3. 如果引脚用作普通输入 端口引脚，不能消除噪声。

复位后: 00H	R/W	地址:	FFFFFF310H							
			7	6	5	4	3	2	1	0
INTPNRC			0	0	0	0	0	0	INTPNRC1	INTPNRC0
			INTPNRC1		INTPNRC0		采样时钟选择			
			0	0	f _{xx} /16					
			0	1	f _{xx} /8					
			1	0	f _{xx} /4					
			1	1	f _{xx} /2					

14.4.2 边沿检测

INTP0 至 INTP6 引脚的有效边沿可以通过编程选择。以下边沿之一可以选择为有效边沿。

- 上升沿
- 下降沿
- 上升沿和下降沿二者均有效

INTP0 至 INTP6 信号的边沿检测可以成为中断源。

有效边沿由 INTR0 和 INTF0 寄存器指定。

(1) 外部中断上升沿指定寄存器 0 (INTR0)，外部中断下降沿指定寄存器 0 (INTF0)

INTR0 和 INTF0 寄存器为 8 位寄存器，用来指明 INTP0 至 INTP6 引脚的触发模式，并且，可以对各个引脚独立地设置有效边沿 (上升沿、下降沿以及双边沿)。

这些寄存器可以按 8 位或按 1 位为单位进行读写。

复位后这些寄存器设置为 00H。

注意事项 当引脚功能由外部中断功能（复用功能）变为端口功能，可能会检测到一个边沿。所以，务必将 INTF0n 和 INTR0n 位清为 00，然后再设置端口模式。

复位后: 00H	R/W	地址: FFFF20H							
		7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
INTR0		0	INTR06	INTR05	INTR04	INTR03	INTR02	INTR01	INTR00
复位后: 00H	R/W	地址: FFFF00H							
		7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
INTF0		0	INTF06	INTF05	INTF04	INTF03	INTF02	INTF01	INTF00
备注	关于有效边沿设定，参见表 14-3。								

表 14-3. 有效边沿设定

INTF0n	INTR0n	有效边沿设定 (n = 0 至 6)
0	0	不检测边沿
0	1	上升沿
1	0	下降沿
1	1	上升沿和下降沿均有效

注意事项 当不将这些引脚用作 INTP0 至 INTP6 引脚，务必将 INTF0n 和 INTR0n 位清为 00。

14.5 软件异常

当 CPU 执行 TRAP 指令时，产生软件异常，并且总能够得到响应。

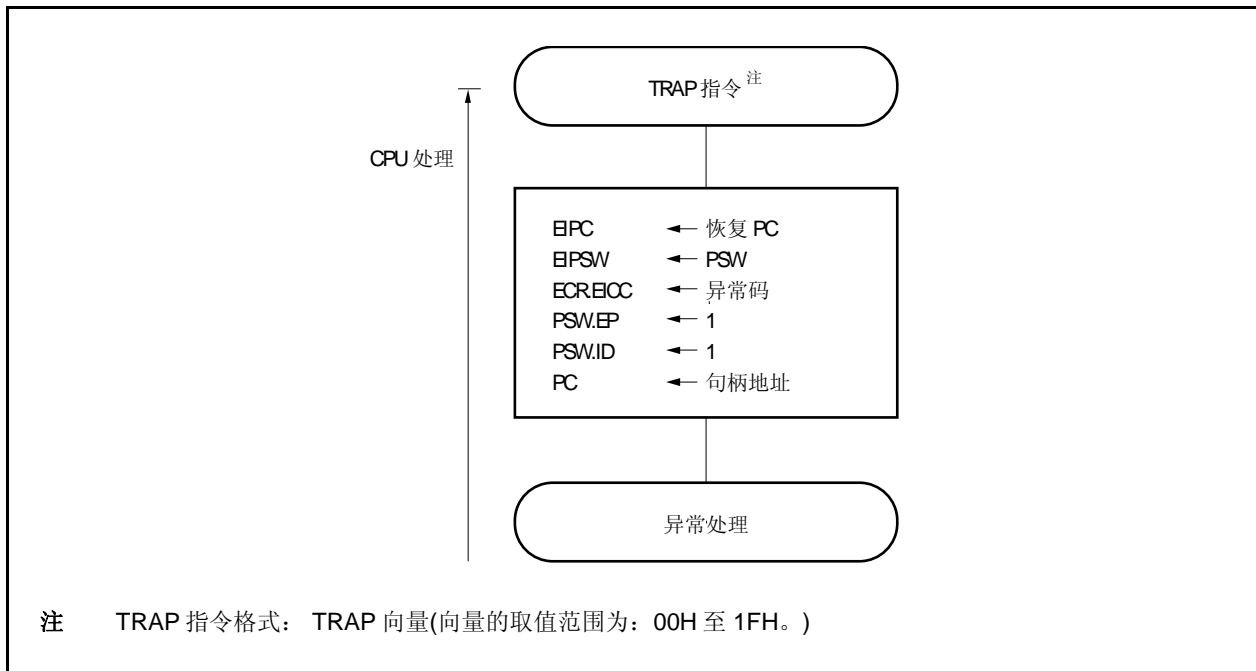
14.5.1 操作

如果发生软件异常，CPU 执行以下处理，并控制转移到处理程序句柄。

- <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(EICC) 的低 16 位。
- <4> 置位 PSW.EP 和 PSW.ID 位 (为 1)。
- <5> 设置 PC 软件异常的处理程序句柄地址(00000040H or 00000050H)，并进行控制转移。

软件异常的处理过程如下所示：

图 14-7. 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，它成为 00000040H，而如果向量为 10H 至 1FH，它成为 00000050H。

14.5.2 恢复

由 RETI 指令负责从软件异常处理之中进行恢复。

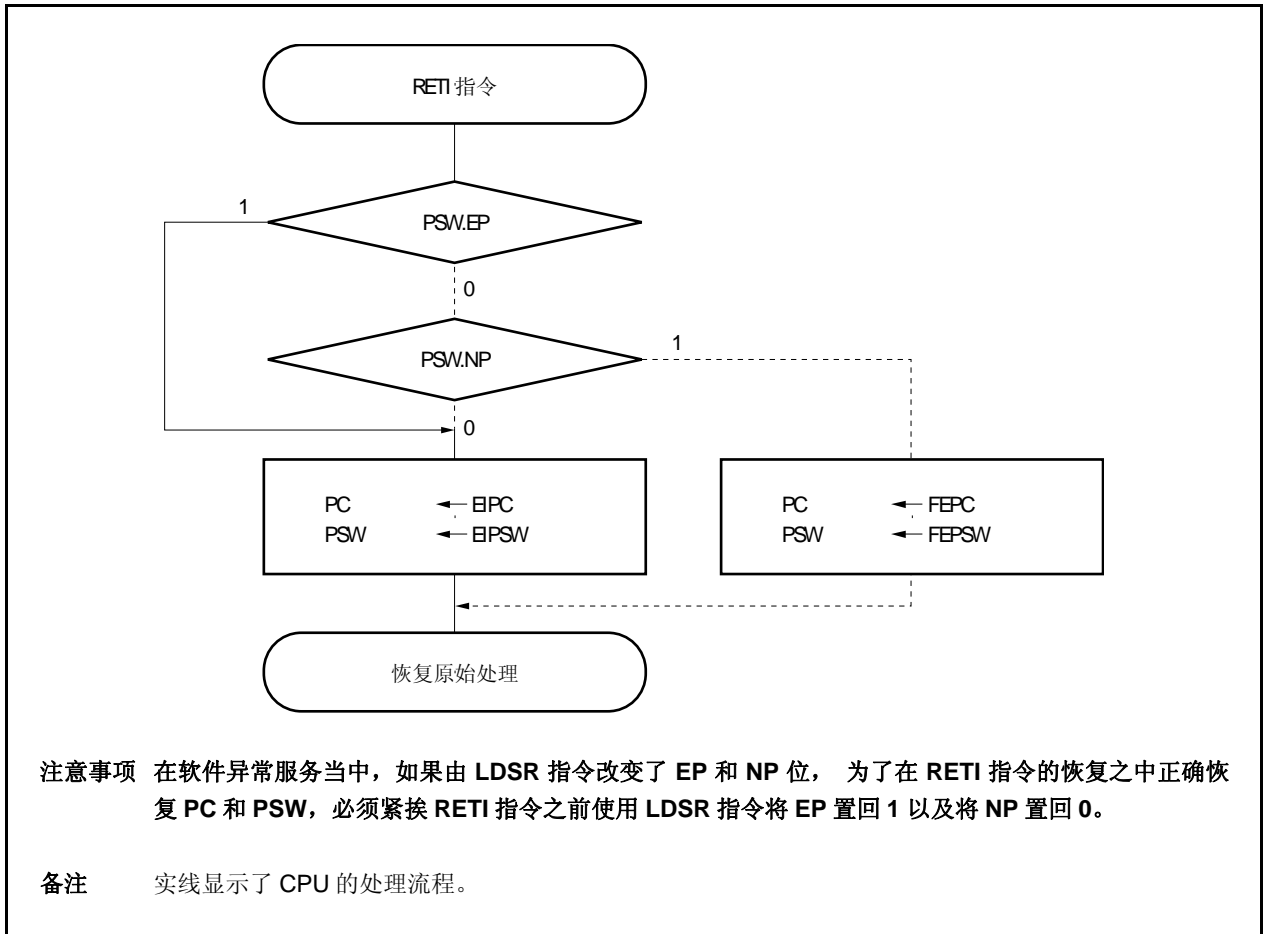
通过执行 RETI 指令，CPU 执行以下处理并控制转移至恢复 PC 的地址。

<1> 由于 PSW.EP 位为 1，由 EIPC 和 EIPSW 载入恢复 PC 和 PSW。

<2> 控制转移至恢复 PC 和 PSW 的地址。

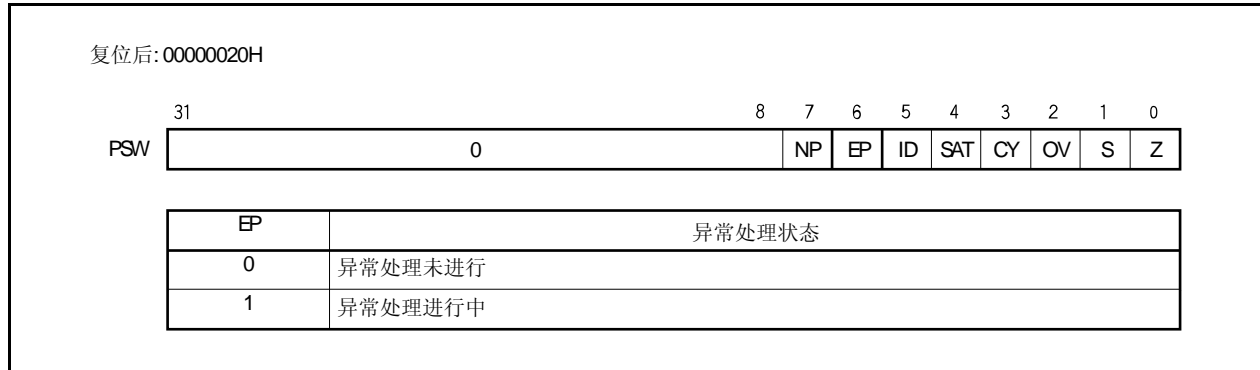
RETI 指令的处理过程如下所示：

图 14-8. RETI 指令处理过程



14.5.3 异常状态标志(EP)

EP 为一个状态标志，用来指示异常处理正在处理当中。复位之后该标志设置为 00000020H。

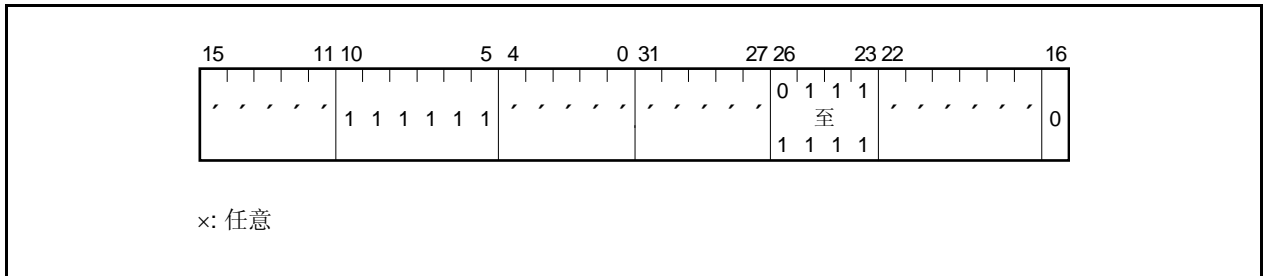


14.6 异常陷阱

异常陷阱为一个中断，当执行非法指令发生时产生中断请求。在 V850ES/IE2 中，一个非法操作码陷阱(ILGOP: 非法操作码陷阱)被认为是一个异常陷阱。

14.6.1 非法操作码定义

非法指令具有一个操作码，其(位 10 至 5)为 111111B，一个次操作码，其(位 26 至 23)为 0111B 至 1111B，以及一个次操作码，其(位 16)为 0B。当使用这样的非法指令进行执行时，产生异常陷阱。



注意事项 由于以后可能将本指令赋为非法操作码，所以，建议不要使用。

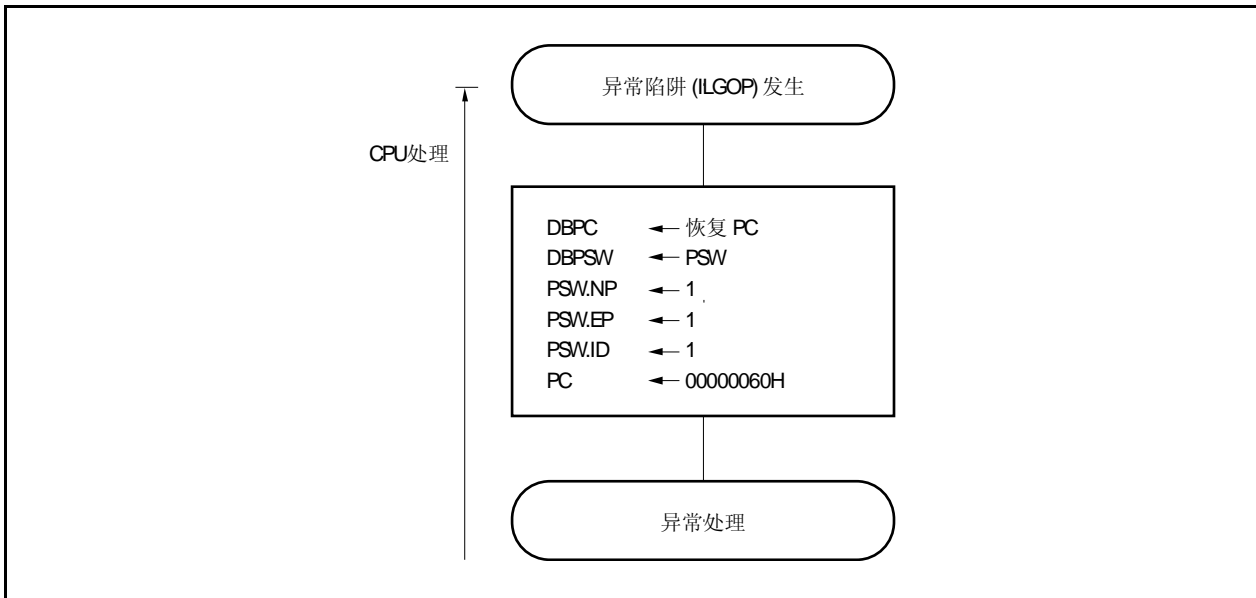
(1) 操作

当发生异常陷阱时，CPU 执行以下处理，并控制转移到处理程序句柄：

- <1> 将恢复 PC 保存至 DBPC。
- <2> 将当前 PSW 保存至 DBPSW。
- <3> 设置 PSW.NP、PSW.EP 和 PSW.ID 位 (为 1)。
- <4> 设置对应于 PC 异常陷阱的句柄地址 (00000060H)，并进行控制转移。

异常陷阱的处理过程如下所示：

图 14-9. 异常陷阱 处理过程



(2) 恢复

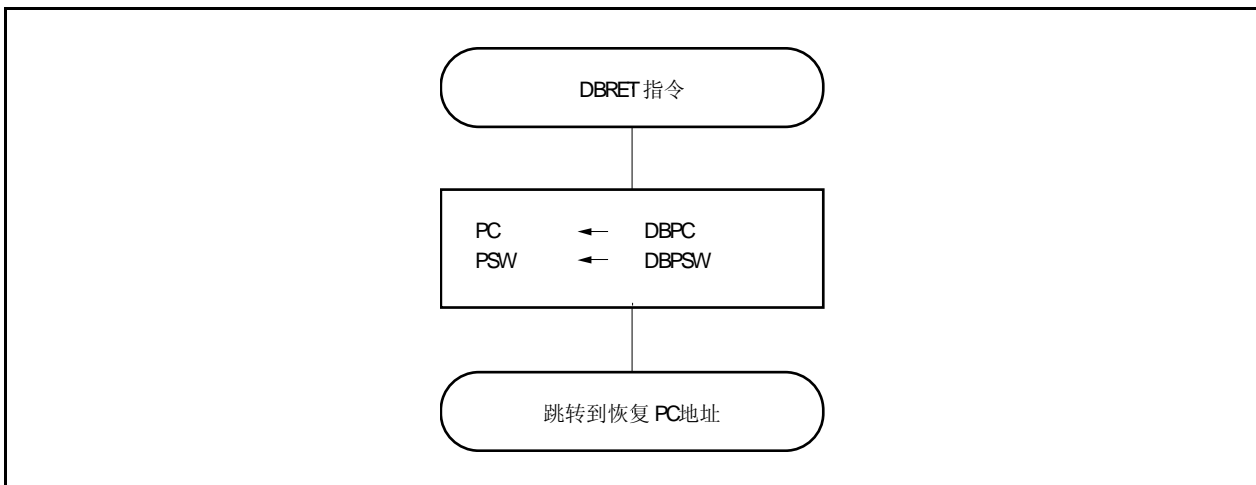
由 DBRET 指令负责从异常陷阱中进行恢复。通过执行 DBRET 指令，CPU 进行以下处理并控制转移至恢复 PC 的地址。

- <1> 从 DBPC 和 DBPSW 中载入恢复 PC 和 PSW。
- <2> 控制转移至恢复 PC 和 PSW 指示的地址。

<R> **注意事项** 只有在非法指令执行时到 DBRET 指令执行之间，才可以访问 DBPC 和 DBPSW。

从异常陷阱之中的恢复处理过程如下所示：

图 14-10. 从异常陷阱中恢复



14.6.2 调试陷阱

调试陷阱是一个异常，由执行 DBTRAP 指令产生，可以在任何时候都得到响应。

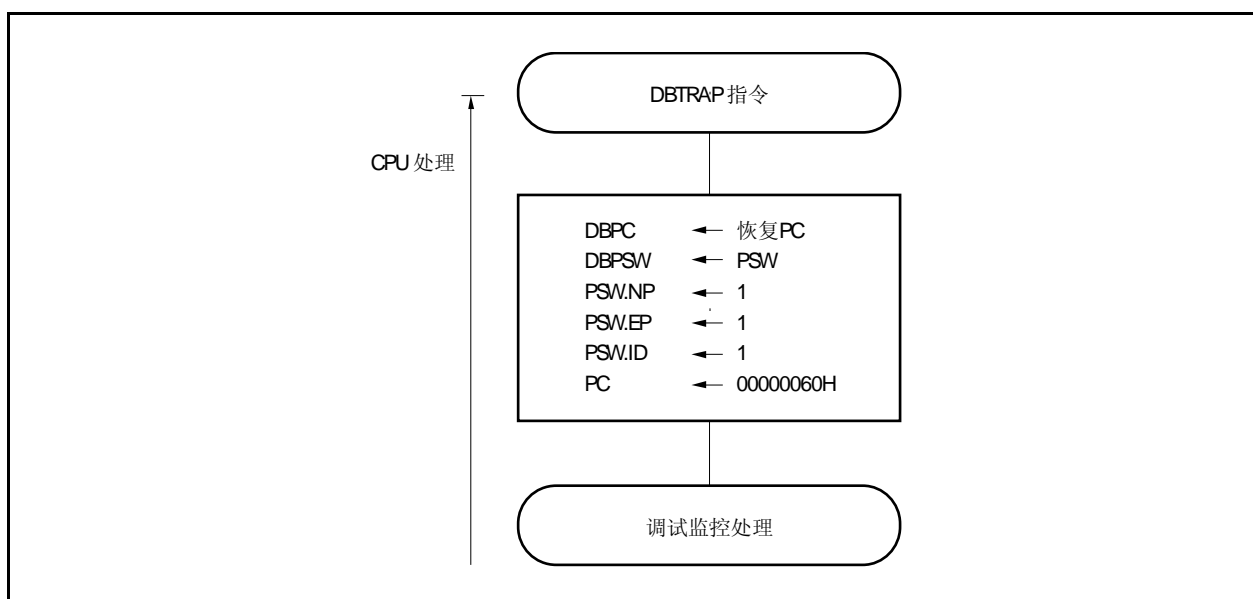
当调试陷阱产生时，CPU 执行以下处理：

(1) 操作

- <1> 将恢复 PC 保存至 DBPC。
- <2> 将当前 PSW 保存至 DBPSW。
- <3> 设置 PSW.NP、PSW.EP 和 PSW.ID 位 (为 1)。
- <4> 设置对应 PC 调试陷阱的句柄地址 (00000060H)，并执行控制转移。

调试陷阱的处理过程如下所示：

图 14-11. 调试陷阱处理过程



(2) 恢复

由 DBRET 指令负责从调试陷阱中进行恢复，通过执行 DBRET 指令，CPU 完成以下处理并控制转移至恢复 PC 地址。

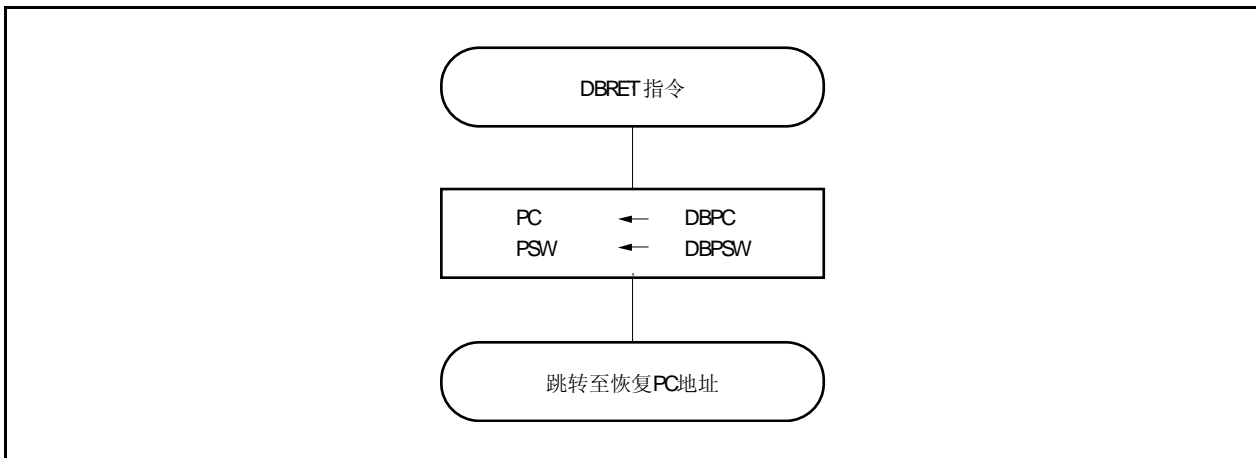
<1> 从 DBPC 和 DBPSW 中载入恢复 PC 和 PSW。

<2> 控制转移至恢复 PC 和 PSW 所指定的地址。

注意事项 在 DBTRAP 执行时到 DBRET 指令执行之间，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 14-12. 从调试陷阱中恢复的处理过程



14.7 多重中断服务控制

多重中断服务控制是这样一种处理机制，根据这个机制，如果存在一个具有较高优先级的中断请求，则低优先级的中断请求在服务期间可以被中断。而且，较高优先级的中断请求信号首先得到响应和服务。

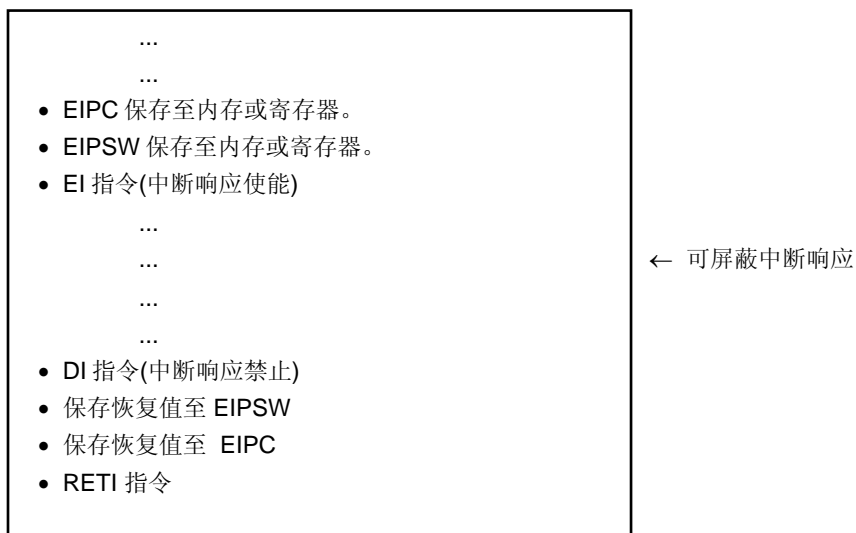
如果有一个中断请求信号其优先级低于当前中断请求的服务，这时，该中断信号保持未决。

当中断使能后(PSW.ID 位 = 0)，执行可屏蔽中断的多重中断服务控制。这就是说，为了执行多重中断，即使在中断服务程序当中，也必须设置为中断使能状态(PSW.ID 位 = 0)。

如果在可屏蔽中断或软件异常服务程序中可屏蔽中断使能或产生一个软件异常，必需保存 EIPC 和 EIPSW。这由以下处理过程完成。

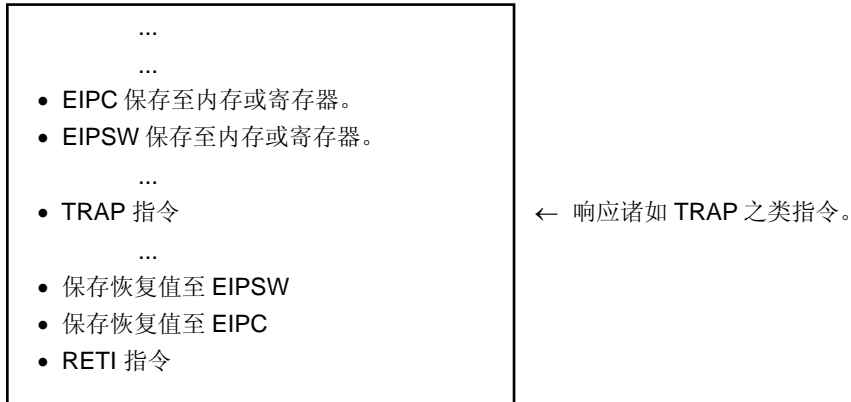
(1) 服务程序中响应可屏蔽中断信号

可屏蔽中断或异常的服务程序



(2) 服务程序中产生异常

可屏蔽中断或异常的服务程序



所有可屏蔽中断请求信号，其多重中断服务控制优先级等级都有 8 级，从 0 至 7 (0 为最高优先级)，可以通过软件设置。每一个可屏蔽中断请求，都可以使用中断控制寄存器(xxCn)的 xxPRn0 至 xxPRn2 位设置优先级等级。系统复位之后，由 xxMKn 位屏蔽中断信号，由 xxPRn0 至 xxPRn2 位将其中断优先级设为 7。

可屏蔽中断的优先级等级如下所示。

(高) 等级 0 > 等级 1 > 等级 2 > 等级 3 > 等级 4 > 等级 5 > 等级 6 > 等级 7 (低)

较高优先级的中断请求信号执行完成，并且 RETI 指令已经执行以后，由多重中断服务控制所挂起未决的中断服务恢复执行。

当前中断服务已完成，并且 RETI 指令已经执行，响应挂起的中断请求信号。

注意事项 在不可屏蔽中断服务程序中(直到执行 RETI 指令时)，可屏蔽中断被挂起，不被响应。

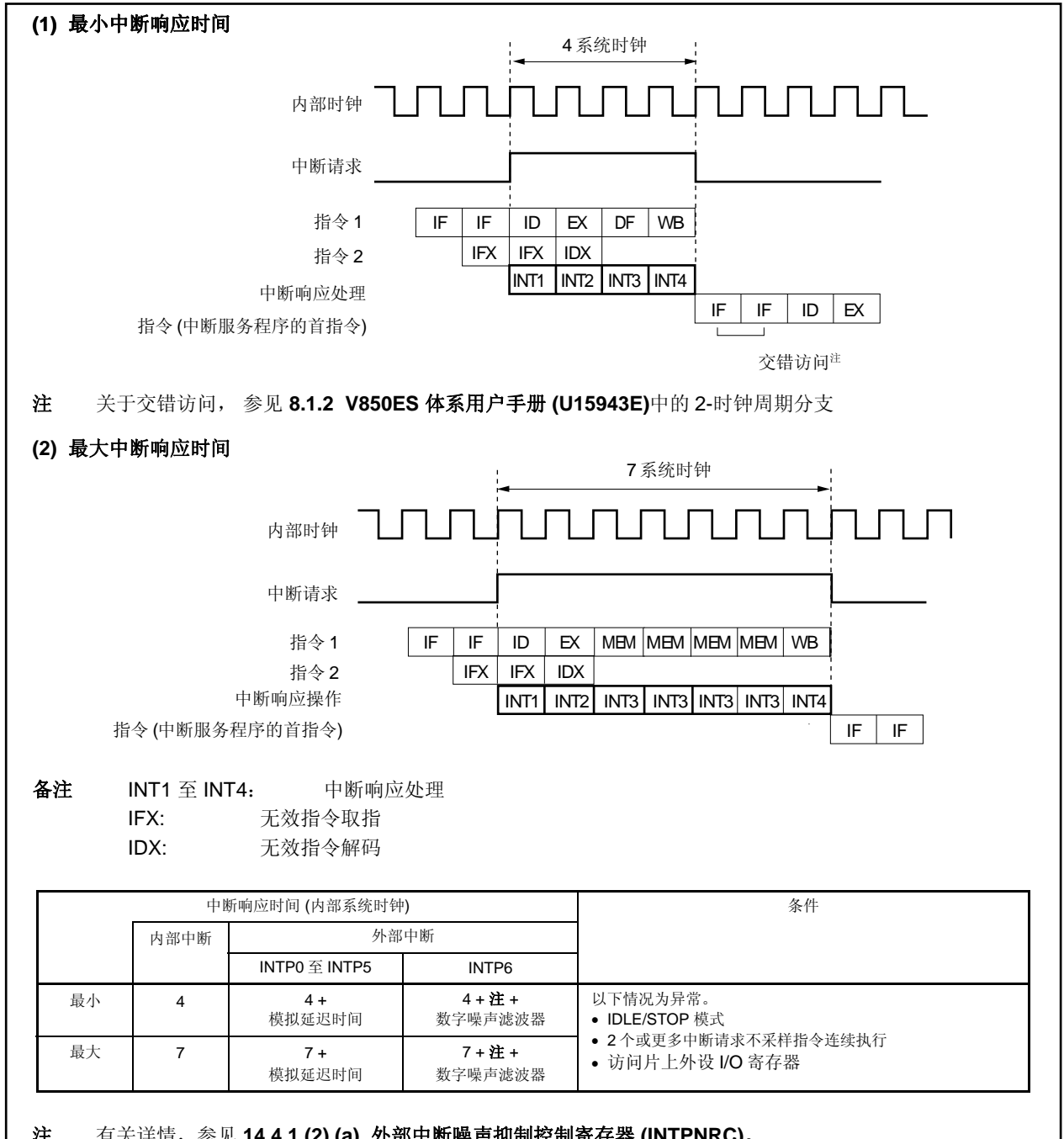
备注 xx: 各个外设单元的标识名 (参见 表 14-2)
n: 外设单元号 (参见 表 14-2)

14.8 CPU的中断响应时间

除了以下情况之外，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 4 个时钟周期输入下一中断请求信号。

- IDLE/STOP 模式
- 不采样指令的中断请求连续执行(参见 14.9 CPU 不响应中断的时期。)
- 访问片上外设 I/O 寄存器。

图 14-13. 中断请求信号响应的流水线操作 (概述)



14.9 CPU不响应中断的时期

当指令正在执行时，CPU 可以响应中断，然而，在中断请求不采样指令和下一条指令之间不会响应中断（中断保持未决）。

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)
- 命令寄存器 (PRCMD) 和命令寄存器 2 (PRCMD2)的存储指令。
- 存储指令或位处理指令，除去对以下寄存器的 tst1 指令。
 - 中断有关寄存器：
中断控制寄存器(xxlCn)和中断屏蔽寄存器 0 至 3 (IMR0 至 IMR3)。
 - 节电控制寄存器 (PSC)
 - 内存大小设置寄存器 (IMS)

备注 xx: 各个外设单元的标识名 (参见 表 14-2)

 n: 外设单元号 (参见 表 14-2)

14.10 注意事项

注意，如果一个端口设置为外部中断请求输入 (INTPn)，作为复用功能的定时器/计数器中断和 A/D 转换中断，则不会发生 (n = 0 至 7)。

第十五章 待机功能

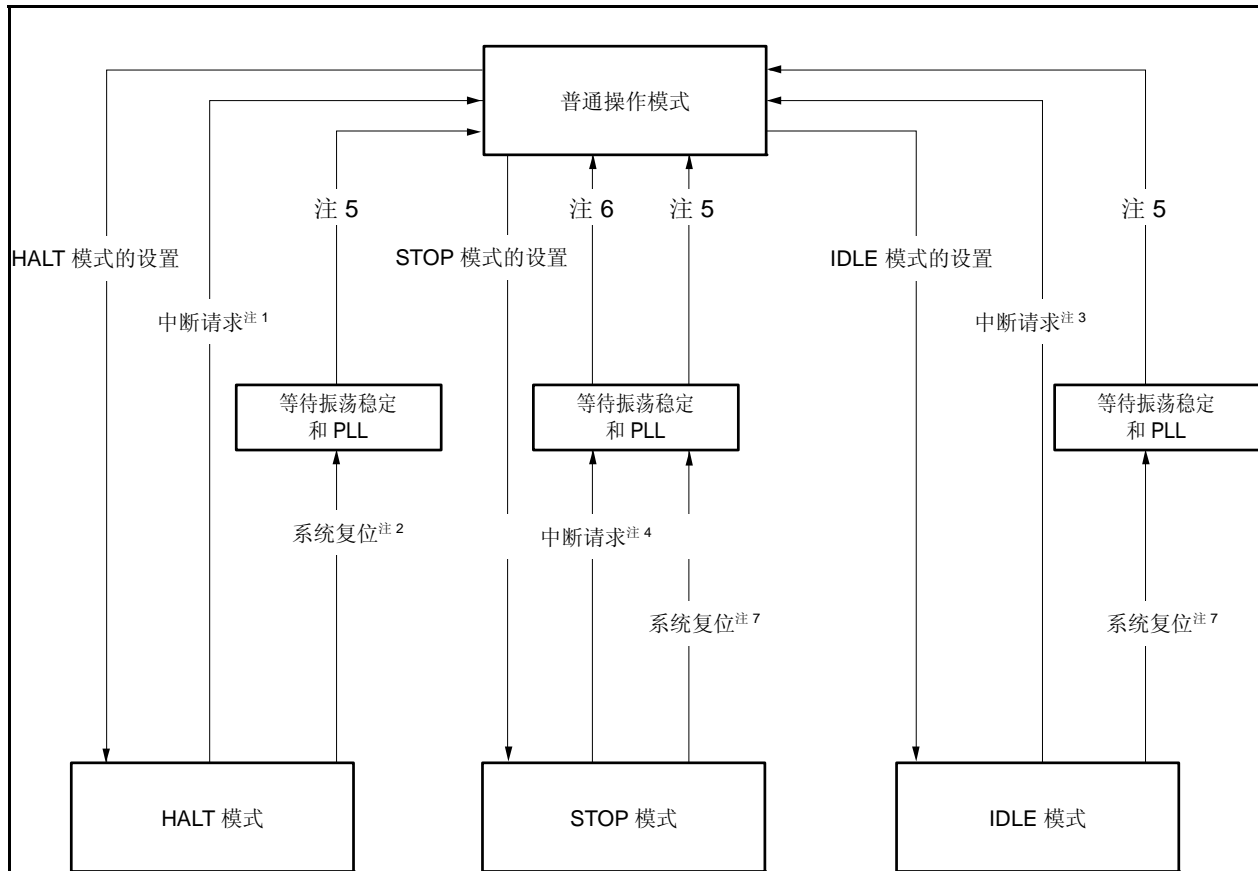
15.1 概述

通过结合运用待机模式并为应用程序选择恰当模式，可以有效地减少系统的功耗。可用待机模式列于表 15-1 中：

表 15-1. 待机模式

方式	功能概述
模式	此模式下，仅停止 CPU 的工作时钟
IDLE 模式	停止内部电路所有操作的模式除低电压检测器（LVI），上电清除电路（POC），振荡器，PLL 和从模式中的 CSIB
STOP 模式	停止内部电路所有操作除低电压检测器（LVI），上电清除电路（POC）和从模式中的 CSIB

图 15-1. 状态转换



- 注
1. 不可屏蔽中断请求信号 (INTWDT) 或未屏蔽的可屏蔽中断请求信号。
 2. $\overline{\text{RESET}}$ 引脚输入, 由看门狗定时器溢出 (WDTRES) 所生成的复位信号, 由低电压检测 (LVIREs) 所生成的复位信号, 或由上电清除 (POCREs) 所生成的复位信号。
 3. 未屏蔽的外部中断请求信号 (INTP0 至 INTP5), 未屏蔽的内部中断请求信号 (INTLVI), 或在 IDLE 模式中可进行操作的外围功能所产生的未屏蔽的内部中断请求信号 (在从模式中与 CSIB 相关的中断请求信号)。
 4. 未屏蔽的外部中断请求信号 (INTP0 至 INTP5), 未屏蔽的内部中断请求信号 (INTLVI), 或在 STOP 模式中可进行操作的外围功能所产生的未屏蔽的内部中断请求信号 (在从模式中与 CSIB 相关的中断请求信号)。
 5. 由振荡稳定时间等待控制 (OST) 计数的振荡稳定时间。
由于 PLL 通过复位被初始化, 因此在复位释放后振荡稳定时间是必需的。稳定时间是默认设置所决定的时间。
 6. 由振荡稳定时间等待控制 (OST) 计数的振荡稳定时间。
稳定时间是由 OSTs 寄存器的设置所决定的。
 7. $\overline{\text{RESET}}$ 引脚输入, 由低电压检测 (LVIREs) 所生成的复位信号, 或由上电清除 (POCREs) 所生成的复位信号。

15.2 控制寄存器

(1) 节电控制寄存器 (PSC)

PSC 寄存器是控制待机功能的 8 位寄存器。该寄存器的 STB 位用来指定待机模式。该寄存器是特殊寄存器（参见 3.4.7 特殊寄存器）。该寄存器只能通过特定顺序的组合来写入。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

复位后: 00H R/W 地址: FFFFF1FEH

	7	6	5	<4>	3	2	<1>	0
PSC	0	0	0	INTM	0	0	STB	0

INTM	通过可屏蔽中断请求控制待机模式 (INTxx ^注)
0	允许 INTxx 请求解除待机模式
1	禁止 INTxx 请求解除待机模式

STB	操作模式设置
0	普通模式
1	待机模式

注 详细信息请参见表 14-1 中断源列表。

注意事项 1. 一定要将位 0、2、3 以及 5 至 7 清为“0”。

2. 在通过将 STB 位设为 1 来设置待机模式前，务必先将 PCC 寄存器设为 03H，然后将 STB 位设为 1。否则，待机模式可能不被设置或者释放。释放待机模式后，将 PCC 寄存器的值改为期望值。

3. 要设置为 IDLE 模式或 STOP 模式，则应按顺序先将 PCC 寄存器设为 03H，再设置 PSMR.PSM0 位，然后将 STB 位设为 1。

4. 写入 PSC 寄存器时，需使用命令寄存器 PRCMD。

<R>

(2) 节电模式寄存器 (PSMR)

PSMR 寄存器是控制软件待机模式中操作的 8 位寄存器。

该寄存器可进行 8 位或 1 位读写。

系统复位后该寄存器被设为 00H。

复位后: 00H R/W 地址: FFFFFFF820H

	7	6	5	4	3	2	1	<0>
PSMR	0	0	0	0	0	0	0	PSM0

PSM0	软件待机模式中的操作规范
0	IDLE 模式
1	STOP 模式

- 注意事项 1. 一定要将位 1 至 7 清除为“0”。
2. 仅当 PSC.STB 位为 1 时，PSM0 位才有效。

15.3 HALT 模式

15.3.1 设置和操作状态

当在正常操作模式下执行一个专用指令（HALT）时，HALT 模式被设置。

当 HALT 模式被设置时，只会停止为 CPU 提供时钟。时钟发生器和 PLL 将继续进行操作。对其他片上外围功能提供时钟的操作也会继续。

因此，程序停止执行，内部 RAM 的内容保持设置为 HALT 模式之前的内容。独立于 CPU 指令处理的片上外围功能继续工作。

表 15-3 显示了 HALT 模式下的操作状态。

通过在 HALT 模式下结合使用正常操作模式作为间歇，可以减少系统的平均功耗。

- 注意事项**
1. 应在 HALT 指令后插入五个或更多 NOP 指令。
 2. 如果有一个未屏蔽的中断请求信号被保持等待响应时执行 HALT 指令，则 HALT 模式被设置但立即被此等待响应的中断请求释放。

15.3.2 释放 HALT 模式

HALT 模式通过不可屏蔽中断请求信号（INTWDT），未屏蔽的可屏蔽中断请求信号，以及复位信号（ $\overline{\text{RESET}}$ 引脚输入，由看门狗定时器溢出（WDTRES）所生成的复位信号），由低电压检测（LVIRES）生成的复位信号以及由上电清除（POCRES）生成的复位信号）来释放。

HALT 模式释放后，正常操作模式恢复。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 HALT 模式

通过不可屏蔽中断请求信号（INTWDT）或未屏蔽的可屏蔽中断请求信号，可释放 HALT 模式，且与中断请求信号的优先级无关。但是，若 HALT 模式被设置在一个中断服务程序中，则之后发出的中断请求信号按如下所述来执行：

注意事项 当 PSC.INTM 位被设为 1 时，HALT 模式将不能通过未屏蔽的可屏蔽中断请求信号来释放。

- (a) 如果生成一个比当前正在执行的中断请求信号的优先级低的或相等的中断请求信号，则释放 HALT 模式，但是不会响应新生成的中断请求信号。该中断请求信号被保持。因此，操作将会在执行 HALT 指令后下个指令处开始。
- (b) 如果发出一个比当前正在执行的中断请求服务的中断请求的优先级高的中断请求信号（包括不可屏蔽中断请求信号），则释放 HALT 模式，并且响应此中断请求信号。因此，将执行对处理地址的跳转。

表 15-2. 用中断请求信号释放 HALT 模式后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	执行对处理地址的跳转。	
未屏蔽的可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令	执行下一条指令。

(2) 通过 **RESET** 引脚输入，生成 **WDTRES** 信号，生成 **LVIRES** 信号或生成 **POCRES** 信号来释放 **HALT** 模式执行与正常的复位操作相同的操作。

表 15-3. HALT 模式下的操作状态

HALT 模式的设置		操作状态
项目		
时钟产生器, PLL		运行
系统时钟 (f _{xx})		提供
CPU		停止运行
中断控制器		可运行
定时器	TMM0	可运行
	TMP0 至 TMP3	可运行
	TMQ0, TMQ1	可运行
看门狗定时器		可运行
串行接口	CSIB0	可运行
	UARTA0, UARTA1	可运行
A/D 转换器 0 和 1		可运行
低电压检测器 (LVI)		使用 LVI 时可运行
上电清除电路 (POC)		可运行
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。

15.4 IDLE 模式

15.4.1 设置和操作状态

IDLE 模式通过在正常操作模式中将 PSMR.PSM0 位清零并将 PSC.STB 设为 1 来进行设置。

在 IDLE 模式下，时钟发生器和 PLL 继续操作但停止给 CPU 和其它片上外设功能的时钟供应。

因此，程序停止执行，且内部 RAM 的内容保持与设置为 IDLE 模式前的内容一致。CPU 及其它片上外设功能停止工作。然而，可使用外部时钟操作的片上外设功继续操作。

表 15-5 显示了 IDLE 模式的操作状态。

与 HALT 模式相比，IDLE 模式更能减少功耗，因为停止了片上外设功能的运行。由于时钟发生器和 PLL 不停止，因此，IDLE 模式释放后，无需等待振荡稳定时间就可以恢复到正常操作模式，HALT 模式释放时亦如此。

注意事项 应在 PSC 寄存器存储数据指令后插入五个或更多 NOP 指令来设置 IDLE 模式。

15.4.2 释放IDLE 模式

IDLE 模式通过未屏蔽的外部中断请求信号（INTP0 至 INTP5 引脚输入），未屏蔽的内部中断请求信号（INTLVI），在 IDLE 模式中可操作的外围功能所生成的未屏蔽的内部中断请求信号（在从模式中与 CSIB 相关的中断请求信号），或复位信号（RESET 引脚输入，由低电压检测生成的复位信号（LVIRES），以及由上电清除生成的复位信号（POCRES））来释放。

释放 IDLE 模式后，恢复到正常操作模式。

(1) 通过未屏蔽的可屏蔽中断请求信号释放 IDLE 模式

IDLE 模式通过未屏蔽的可屏蔽中断请求信号来释放，而不考虑中断请求的优先级。如果在某个中断服务程序中设置 IDLE 模式，则之后产生的中断请求信号按如下所述来进行处理：

注意事项 当 PSC.INTM 位被设为 1 时，IDLE 模式将不能通过未屏蔽的可屏蔽中断请求信号来释放。

- (a) 如果生成一个比当前正在执行的中断请求信号的优先级低的或相等的中断请求信号，则释放 IDLE 模式，但是不会响应新生成的中断请求信号。该中断请求信号被保持。因此，操作将会在执行 IDLE 指令后下个指令处开始。
- (b) 如果发出一个比当前正在执行的中断请求服务请求的优先级高的中断请求信号，则将释放 IDLE 模式，但是不响应此中断请求信号。因此，将执行对处理地址的跳转。

表 15-4.通过中断请求信号释放 IDLE 模式后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
未屏蔽的可屏蔽中断请求	执行对处理地址的跳转或执行下一条指令。	执行下一条指令。

(2) 通过 $\overline{\text{RESET}}$ 引脚输入，生成 $\overline{\text{LVIRES}}$ 信号以及生成 $\overline{\text{POCRES}}$ 信号来释放 IDLE 模式
执行与正常的复位操作相同的操作。

表 15-5. 在 IDLE 模式下的操作状态

IDLE 模式的设置		操作状态
项目		
时钟产生器, PLL		运行
系统时钟 (f _{xx})		停止提供
CPU		停止运行
中断控制器		停止运行
定时器	TMM0	停止运行
	TMP0 至 TMP3	停止运行
	TMQ0, TMQ1	停止运行
看门狗定时器		停止运行
串行接口	$\overline{\text{CSIB0}}$	$\overline{\text{SCKB0}}$ 输入时钟被选为计数时钟时 (在从模式中) 可运行
	UARTA0, UARTA1	停止运行
A/D 转换器 0 和 1		停止运行
低压检测器 (LVI)		使用 LVI 时可运行
上电清除电路 (POC)		可运行
端口功能		保持设置为 IDLE 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 (如内部 RAM 的内容) 与设置为 IDLE 模式前相比保持原样。

15.5 STOP 模式

15.5.1 设置和操作状态

STOP 模式通过在正常操作模式中将 PSMR.PSM0 位设为 1 并将 PSC.STB 设为 1 来进行设置。

在 STOP 模式中，时钟发生器停止操作。停止对 CPU 和片上外设功能的时钟供给。

因此，程序停止执行，且内部 RAM 的内容保留为设置 STOP 模式前的内容。CPU 及其它片上外设功能停止工作。

<R> 可使用外部时钟操作的片上外设功能继续操作。

表 15-7 显示了 STOP 模式的操作状态。

因为 STOP 模式停止了时钟发生器的操作，所以 STOP 模式比 IDLE 模式更能降低功耗。因此，如果不使用外部时钟，则功耗只通过泄漏电流来最小化。

注意事项 应在 PSC 寄存器存储数据指令后插入五个或更多 NOP 指令来设置为 STOP 模式。

15.5.2 释放 STOP 模式

STOP 模式通过未屏蔽的外部中断请求信号（INTP0 至 INTP5 引脚输入），未屏蔽的内部中断请求信号（INTLVI），在 STOP 模式中可操作的外围功能所生成的未屏蔽的内部中断请求信号（在从模式中与 CSIB 相关的中断请求信号），或复位信号（RESET 引脚输入，由低电压检测生成的复位信号（LVIRES），以及由上电清除生成的复位信号（POCRES））来释放。

在 STOP 模式释放后，正常操作模式在振荡稳定时间得到保证后恢复。

(1) 通过未屏蔽的可屏蔽中断请求信号释放 STOP 模式

STOP 模式通过未屏蔽的可屏蔽中断请求信号来释放，而不考虑中断请求的优先级。但是，若 STOP 模式被设置在一个中断服务程序中，则之后发出的中断请求信号按如下所述来执行：

注意事项 当 PSC.INTM 位被设为 1 时，STOP 模式将不能通过未屏蔽的可屏蔽中断请求信号来释放。

- (a) 如果生成一个比当前正在执行的中断请求信号的优先级低的或相等的中断请求信号，则释放 STOP 模式，但是不会响应新生成的中断请求信号。该中断请求信号被保持。因此，操作将会在执行 STOP 指令后下个指令处开始。
- (b) 如果发出比当前正在进行服务的中断请求信号的优先级更高的中断请求信号，则释放 STOP 模式并响应此中断请求信号。因此，将执行对处理地址的跳转。

表 15-6. 通过中断请求信号释放 STOP 模式的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
未屏蔽的可屏蔽中断请求信号	在保证振荡稳定时间之后执行对处理地址的跳转或执行下一条指令。	在保证振荡稳定时间之后执行下一条指令。

- (2) 通过 **RESET** 引脚输入，生成 **LVIRE** 信号以及生成 **POCRE** 信号来释放 STOP 模式
执行与正常的复位操作相同的操作。

表 15-7. STOP 模式的操作状态

STOP 模式的设置		操作状态
项目		
时钟产生器, PLL		停止运行
系统时钟 (f _{xx})		停止提供
CPU		停止运行
中断控制器		停止运行
定时器	TMM0	停止运行
	TMP0 至 TMP3	停止运行
	TMQ0, TMQ1	停止运行
看门狗定时器		停止运行
串行接口	CSIB0	$\overline{\text{SCKB0}}$ 输入时钟被选为计数时钟时 (在从模式中) 可运行
	UARTA0, UARTA1	停止运行
A/D 转换器 0 和 1		停止运行
低电压检测器 (LVI)		使用 LVI 时可运行
上电清除电路 (POC)		可运行
时钟监视器		停止运行 (然而, 内部振荡器继续运行)
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、数据及其它内部数据 (如内部 RAM 的内容) 与设置为 STOP 模式前相比保持原样。

15.6 保证振荡稳定时间

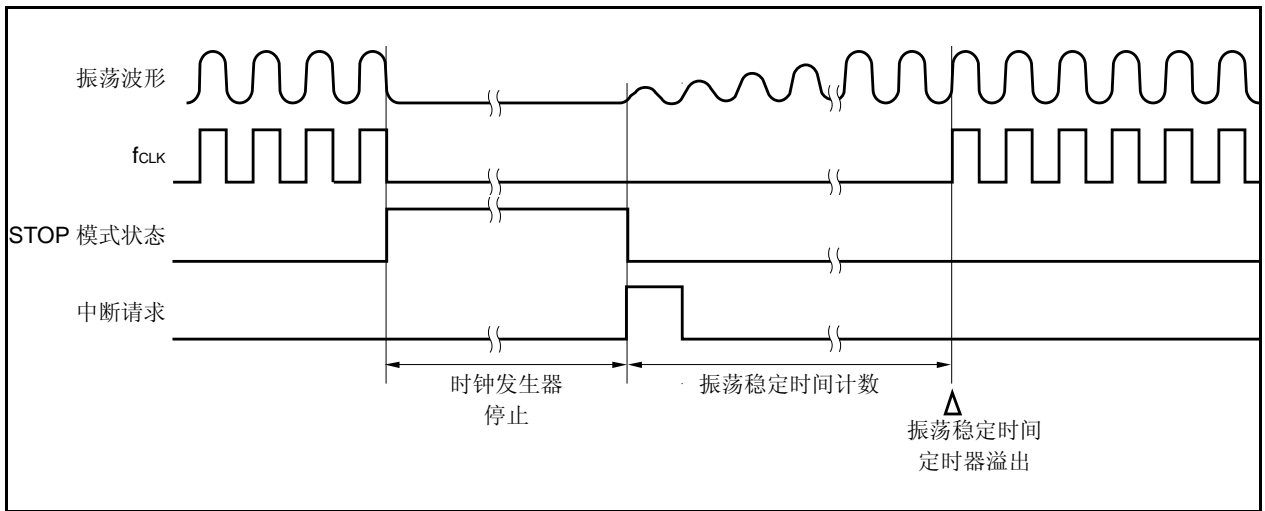
STOP 模式释放时，消耗由 OSTS 寄存器设置的振荡稳定时间。如果 STOP 模式通过复位信号（RESET 输入，由低电压检测生成的复位信号（LVIRES）以及由上电清除生成的复位信号（POCRES））被释放，那么振荡稳定时间将会是 OSTS 寄存器的复位值， $2^{16}/f_x$ （ $f_x = 2.5 \text{ MHz}$ 时 26.2 ms ）。

然而，实际振荡稳定时间是该值的一半（复位后： $2^{15}/f_x$ （ $f_x = 2.5 \text{ MHz}$ 时 13.1 ms ），而另一半时间则是 PLL 的稳定时间。STOP 模式被释放时将振荡稳定时间设为振荡器所用振荡稳定时间的两倍。

用于计数振荡稳定时间的定时器确保振荡稳定时间等于看门狗定时器的溢出时间。

由中断请求信号释放 STOP 模式所执行操作如下图所示：

图 15-2. 振荡稳定时间



注意事项 关于 OSTS 寄存器的详细信息，请参见 5.3（6）振荡稳定时间选择寄存器（OSTS）。

第十六章 复位功能

16.1 概述

有下列复位功能：

- 通过 $\overline{\text{RESET}}$ 引脚输入来复位
- 通过看门狗定时器溢出来复位 (WDTRES)
- 通过低电压检测器 (LVI) 复位系统 (LVIRES)
- 通过上电清除电路 (POC) 复位系统 (POCRES)

16.2 检测复位源的寄存器

(1) 复位源标志寄存器 (RESF)

RESF 寄存器是一个特殊的寄存器，只能够通过特殊的顺序来写入（参见 3.4.7 特殊寄存器）。

RESF 寄存器表明复位信号是由看门狗定时器 (WDT) 生成的。

LVIRF 和 WDT2RF 位通过 $\overline{\text{RESET}}$ 引脚进行复位或通过位操作指令或存储指令（对 LVIRF 和 WDT2RF 位写入 0）来清零。

本寄存器能按 8 位进行读取。但是，位 0 是只能写入。

该寄存器通过 $\overline{\text{RESET}}$ 引脚输入来清除为 00H，并通过上电清除电路 (POC) 来复位。如果复位源不同于这些，缺省值会有所不同。

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	0	LVIRF

WDT2RF	来自 WDT 的复位信号
0	没有生成/被清除
1	已生成

LVIRF	RESF2.LVIRFS 位的清除
0	清除
1	禁止写入

注 该寄存器的值在通过 $\overline{\text{RESET}}$ 引脚输入或上电清除电路 (POC) 复位后被清除为 00H。当通过看门狗定时器溢出执行复位后，该寄存器被设为 10H 或 11H。通过低电压检测器 (LVI) 复位时，位 4 将保持复位前的值，且位 0 未定义的。

注意事项 1. 只有“0”能写入该寄存器的各位。如果写“0”与标志的设置冲突（发生复位），标志的设置会优先。
2. 写入 RESF 寄存器时，需使用命令寄存器 PRCMD。

<R>

(2) 复位源标志寄存器 2 (RESF2)

RESF2 寄存器表明复位信号是由低电压检测器 (LVI) 生成的。

该寄存器为只读寄存器，按 8 位读取。

该寄存器通过 $\overline{\text{RESET}}$ 引脚输入和上电清除电路 (POC) 来清除为 00H 或设置为 10H。通过其他方式执行复位时的默认值是不相同的。

复位后: 注 1	R	地址: FFFFF8A8H						
RESF2	7	6	5	4	3	2	1	0
	0	0	0	未定义	0	0	0	LVIRFS ^{注2}

LVIRFS ^{注2}	来自 LVI 的复位信号
0	没有生成
1	已生成

注

1. 通过 $\overline{\text{RESET}}$ 引脚输入或上电清除电路 (POC) 复位后，该寄存器将被设为 00H 或 10H。通过低电压检测器 (LVI) 复位后，该寄存器将会被设为 01H 或 11H。通过看门狗定时器溢出复位后，复位前的寄存器值将会保持。
2. 向 RESF.LVIRF 位写入 0 用来将 LVIRFS 标志清零。

表 16-1. 复位源生成时 RESF 和 RESF2 寄存器中位 0 和 4 的值

复位源	寄存器	RESF 寄存器		RESF2 寄存器	
		位 4	位 0	位 4	位 0
通过 $\overline{\text{RESET}}$ 引脚输入的复位功能		0	0	未定义	0
通过上电清除电路 (POC) 复位系统 (POCRES)		0	0	未定义	0
通过 WDT 溢出的复位功能 (WDTRES)		1	未定义	保持	保持
通过低电压检测器 (LVI) 复位系统 (LVIRES)		保持	未定义	未定义	1

16.3 操作

16.3.1 通过RESET引脚实现的复位操作

当 RESET 引脚输入一个低电平时，系统复位，且所有硬件将被初始化。

当 RESET 引脚的电平由低变高时，复位状态将被解除。

如果通过 RESET 引脚输入解除复位状态，那么将先会经过振荡器的振荡稳定时间（OSTS 寄存器的复位值： $2^{16}/f_x$ ），然后 CPU 开始执行程序。

表 16-2. RESET 引脚输入时的硬件状态

项目	复位期间	复位解除后
振荡器 (fx)	振荡停止	振荡开始
内部振荡器	振荡停止	振荡开始
外围时钟 (fx 至 fx/2, 048)	停止运行	在振荡稳定时间得到保证后开始运行
内部系统时钟 (fclk), CPU 时钟 (fcPU)	停止运行	在振荡稳定时间得到保证后开始运行 (初始化为 fx/8)
CPU	初始化	在振荡稳定时间得到保证后开始执行程序 ^{#1}
内部 RAM	上电复位或数据被写入 RAM (通过 CPU) 时为未定义, 且复位冲突 (数据被破坏) 否则, 数值在复位后会立即保留 ^{#2} 。	
I/O 线路 (端口/复用功能引脚)	高阻抗	
片上外设 I/O 寄存器	初始化为指定状态	
其它片上外设功能	停止运行	在振荡稳定时间得到保证后开始运行

- 注
1. 使用 μ PD70F3714 时, 引导交换内部处理的插入会延迟程序的执行。
 2. 内部系统复位操作被解除后, μ PD70F3714 中的固件会使用部分内部 RAM (所使用的 RAM 区域: 3FFE000H 至 3FFE095H, 3FFEFBAH 至 3FFEFFFH), 这是因为它支持引导交换功能。因此, 在上电复位时一些 RAM 区域中的内容将不会被保留。

图 16-1. RESET 引脚输入引起的复位操作的时序

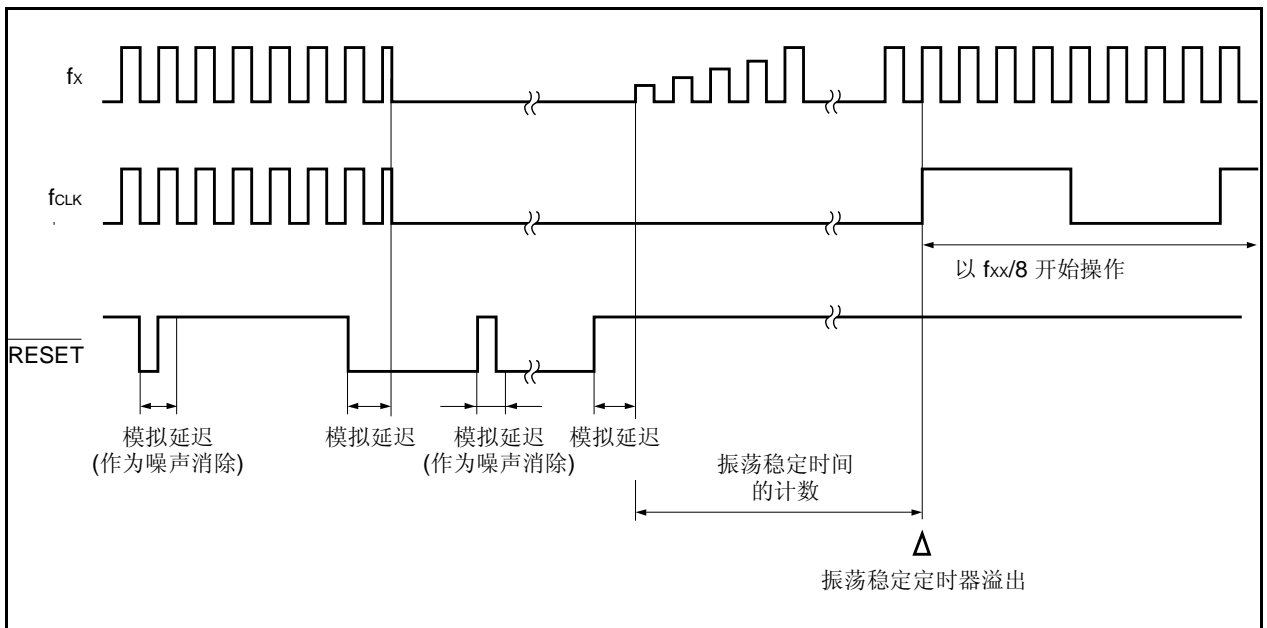
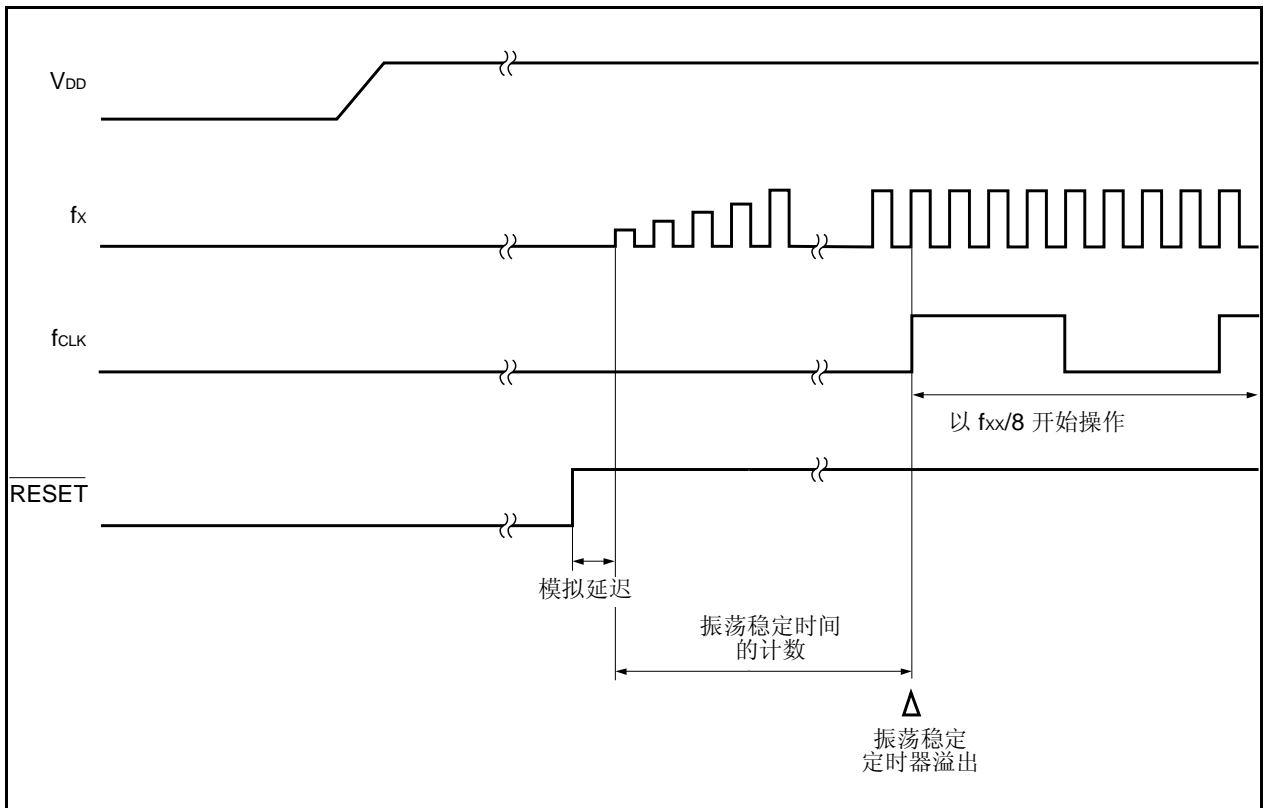


图 16-2. 上电（供应电源）复位操作时序



16.3.2 通过看门狗定时器（WDT）溢出实现的复位操作（WDTRES）

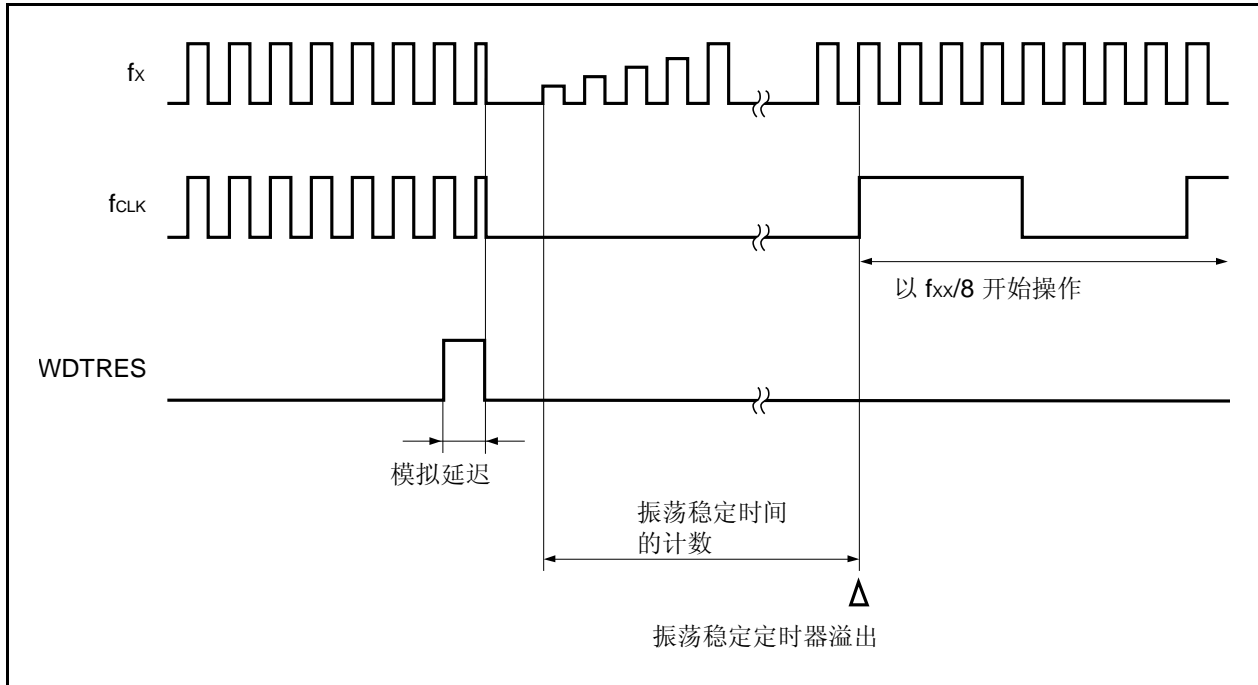
当看门狗定时器由于溢出被设置为复位操作模式时，在看门狗定时器溢出的基础上（产生 WDTRES 信号），将会执行系统复位且硬件将被初始化。

在看门狗定时器溢出后，会进入复位状态并且持续到预定时间（模拟延迟），然后复位状态将会自动解除。复位解除后，在保证振荡器的振荡稳定时间（OSTS 寄存器的初始值： $2^{16}/f_x$ ）后 CPU 将开始执行程序。

振荡器会在复位期间停止以保证振荡稳定时间。

在复位期间以及复位解除后各硬件的状态与 $\overline{\text{RESET}}$ 引脚实现复位操作的状态相同（参见 16.3.1 通过 $\overline{\text{RESET}}$ 引脚实现的复位操作）。

图 16-3. WDTRES 信号生成所引起的复位操作的时序



16.3.3 低电压检测器 (LVI)

(1) 功能

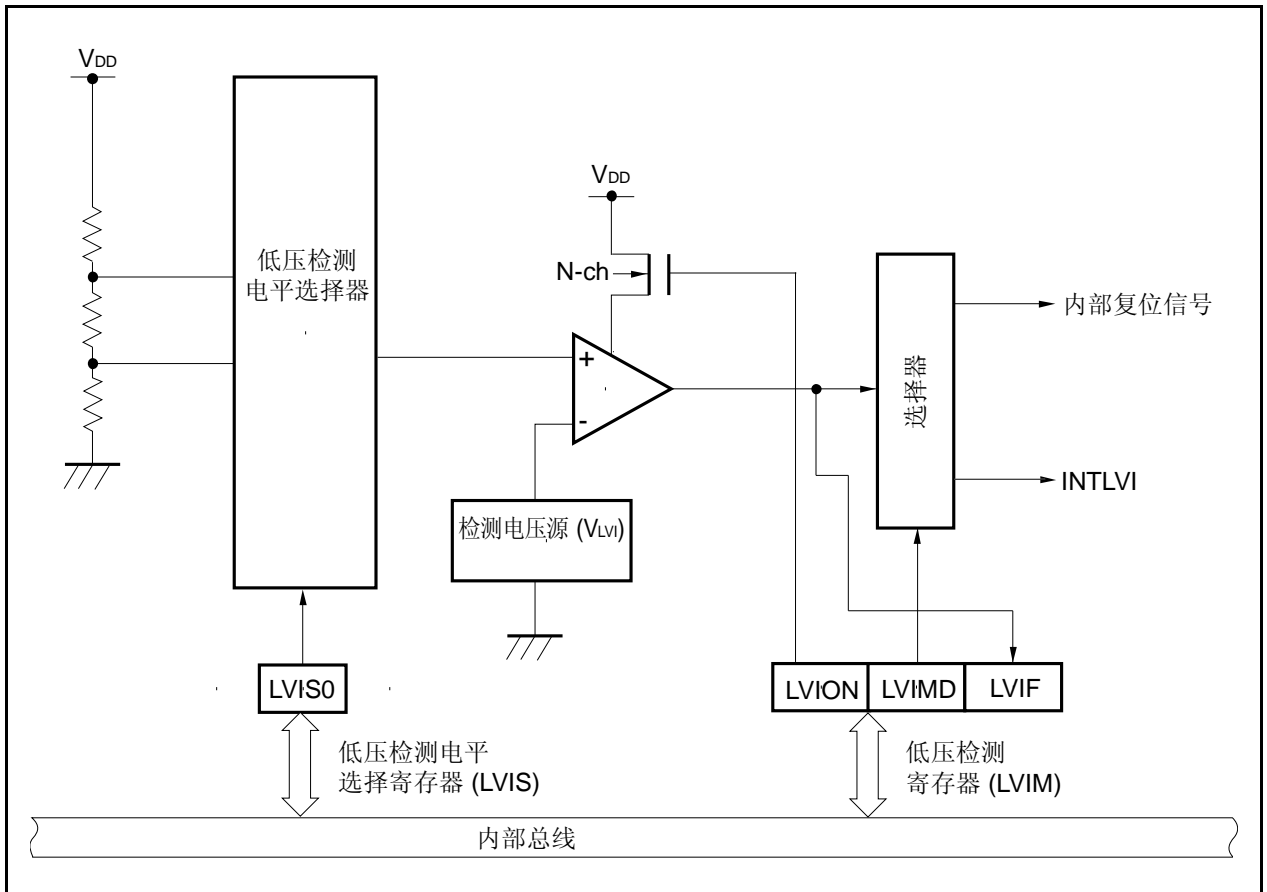
低电压监测器 (LVI) 具有以下功能:

- 将供应电压 (V_{DD}) 与检测电压 (V_{Lvi}) 进行比较, 并在 $V_{DD} < V_{Lvi}$ 时生成一个中断请求信号或内部复位信号。
- 需要检测的供应电压的电平可以通过两步来改变。
- 可以选择中断请求信号或内部复位信号。
- 可以在 HALT/IDLE/STOP 模式中进行操作。
- 可以通过软件停止操作。

(2) 配置

功能框图如下:

图 16-4. 低电压检测器的框图



(3) 控制寄存器

(a) 低电压检测寄存器 (LVIM)

LVIM 寄存器用于允许或禁止低电压检测，并用于设置低电压检测器的操作模式。LVIM 寄存器是一个特殊寄存器。只有通过结合特定顺序才能写入该寄存器（见 3.4.7 特殊寄存器）。

该寄存器可进行 8 位或 1 位读写。但是，位 0 是只能读取。

该寄存器通过由 RESET 引脚输入，上电清除电路（POC）以及看门狗定时器溢出所实现的复位清除为 00H。该寄存器在通过低电压检测器（LVI）实现复位后被设为 82H。

复位后: 00H [*]		R/W	地址: FFFFF890H						
	<7>	6	5	4	3	2	<1>	<0>	
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF	

LVION	允许或禁止低电压检测操作
0	禁止操作。
1	允许操作。

LVIMD	低电压检测操作模式的选择
0	当 供应电压 < 检测电压时生成中断请求信号 INTLVI。
1	当 供应电压 < 检测电压时生成内部复位信号 LVIRES。

LVIF	低电压检测标志
0	供应电压 > 检测电压，或禁止操作。
1	供应电压 < 检测电压。

注 该寄存器通过由 RESET 引脚输入，上电清除电路（POC）以及看门狗定时器溢出所实现的复位清除为 00H。该寄存器在通过低电压检测器（LVI）实现复位后被设为 82H。

- 注意事项
1. 将 LVION 位设为 1 后，在使用 LVIF 位检查电压前需等待 0.1 ms（TYP）（目标值）。
 2. 当 LVION 位 = 1 且 LVIMD 位 = 0 时，LVIF 标志的值将作为输出信号 INTLVI 输出。
 3. 一定要将位 2 至 6 清除为“0”。
 4. 写入 LVIM 寄存器时，需使用命令寄存器 PRCMD2。

<R>

(b) 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等级。

该寄存器可进行 8 位或 1 位读写。

该寄存器通过由 **RESET** 引脚输入，上电清除电路 (POC) 以及看门狗定时器溢出所实现的复位清除为 00H。该寄存器的值在通过低电压检测器 (LVI) 复位后被保持。

复位后: 00H^注 R/W 地址: FFFFF891H

	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	0	0	LVIS0

LVIS0	检测等级
0	4.4 V ±0.2 V
1	4.2 V ±0.2 V

注 该寄存器通过由 **RESET** 引脚输入，上电清除电路 (POC) 以及看门狗定时器溢出所实现的复位清除为 00H。该寄存器的值在通过低电压检测器 (LVI) 复位后被保持。

注意事项 一定要将位 1 至 7 清除为“0”。

(c) 内部 RAM 数据状态寄存器 (RAMS)

RAMS 寄存器是一个用于表明内部 RAM 是否有效的标志寄存器。RAMS 寄存器是一个特殊寄存器。只有通过结合特定顺序才能写入该寄存器（见 3.4.7 特殊寄存器）。

关于 RAMS 寄存器的详细信息，参见 (5) RAM 保持电压检测操作。

该寄存器可进行 8 位或 1 位读写。

该寄存器可以通过 RESET 引脚输入（只有在 RAM 访问期间）以及看门狗定时器溢出来设为 01H。通过其他原因复位时值会被保持。

<R> **注意事项 1. 写入 RAMS 寄存器时，需使用命令寄存器 PRCMD2。**

2. 以下表明了复位后的特殊顺序。

- **设置条件:**
 - 检测电压低于指定电平 (2.0 V ± 0.1 V)
 - 通过特定顺序写入 1
 - 通过看门狗定时器溢出生成复位信号
 - 访问在 RAM 时输入 RESET 引脚
- **清除条件:**
 - 以特定顺序写入 0

	复位后: 01H ^注	R/W	地址: FFFFF892H					
	7	6	5	4	3	2	1	<0>
RAMS	0	0	0	0	0	0	0	RAMF
	RAMF	内部 RAM 数据有效/无效						
	0	有效						
	1	无效						

注 该寄存器可以通过 $\overline{\text{RESET}}$ 引脚输入（只有在 RAM 访问期间）以及看门狗定时器溢出来设为 01H。通过其他原因复位时值会被保持。

(4) 操作

根据 LVIM.LVIMD 位的设置，会产生一个中断请求信号（INTLVI）或者一个内部复位信号。

(a) 用作内部复位信号

如果在允许 LVI 操作时供应电压低于通过低电压检测器检测到的电压，那么将会对系统进行复位（LVIM.LVIMD 位被设为 1 时），且硬件将会被初始化。

复位状态会从检测到供应电压降低开始一直持续到供应电压高于 LVI 检测电压为止。复位解除后，在保证振荡器的振荡稳定时间（OSTS 寄存器的初始值： $2^{16}/f_x$ ）后 CPU 将开始执行程序。

主时钟振荡器会在复位期间停止以保证振荡稳定时间。

在复位期间以及复位解除后各硬件的状态与 **RESET** 引脚实现复位操作的状态相同（参见 **16.3.1 通过 RESET 引脚实现的复位操作**）。以下显示了操作设置方法及时序图。

<准备开始操作>

<1> 屏蔽 LVI 中断。

<2> 由 LVIS.LVIS0 位选择检测的电压。

<3> 设置 LVIM.LVION 位为 1（以允许操作）。

<4> 通过软件插入一个 0.1 ms（TYP）（目标值）或更长的等待周期。

<5> 使用 LVIM.LVIF 位，检验供应电压是否大于检测电压。

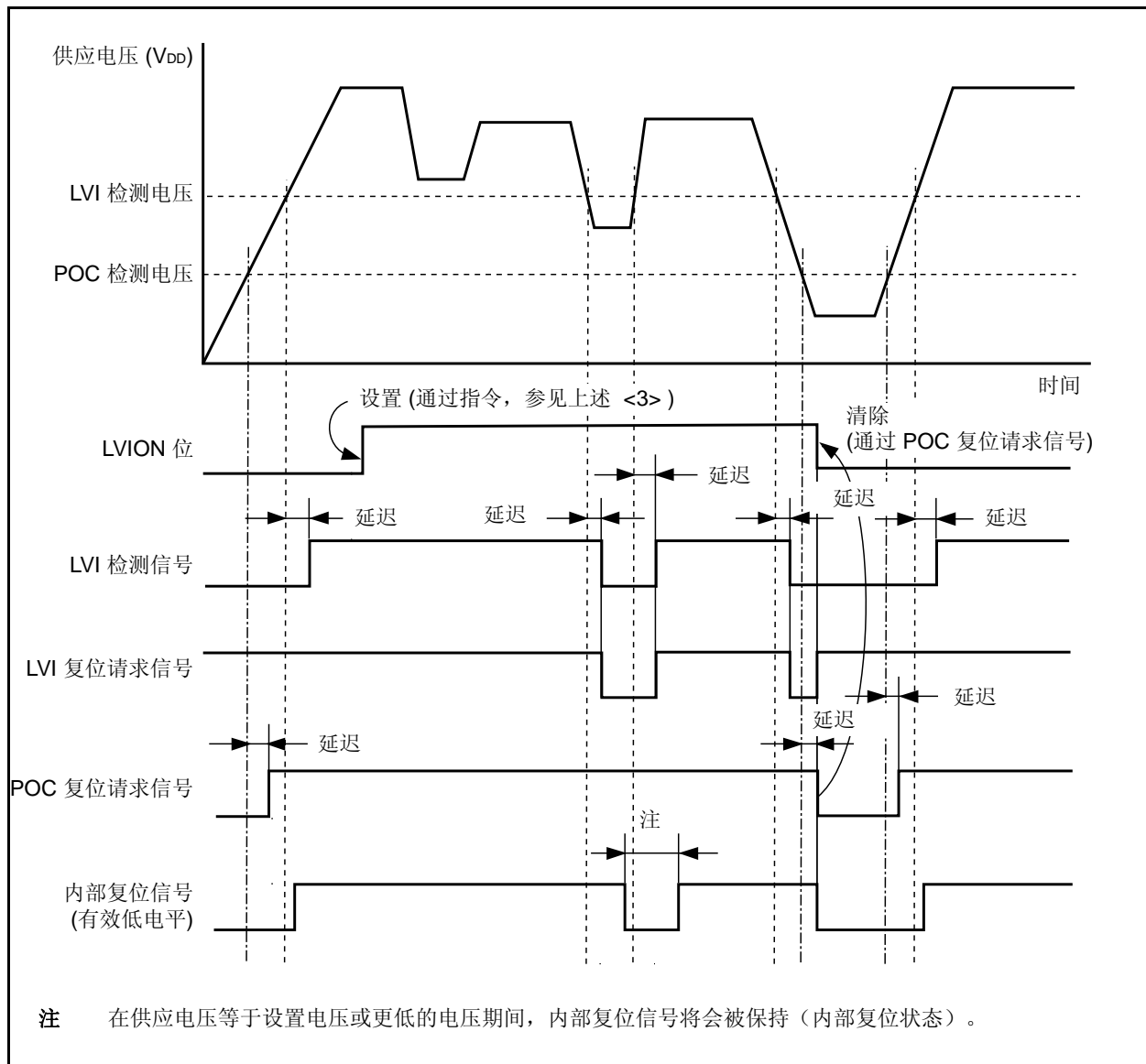
<6> 设置 LVIM.LVIMD 位为 1（以产生内部复位信号）。

注意事项 如果 LVIMD 位设置为 1，那么 LVIM 和 LVIS 寄存器的内容不能改变，直到 LVI 之外的复位请求产生。

<准备停止操作>

LVI 操作不能被停止直到从不同于 LVI 的复位请求生成。

图 16-5. 低电压检测器的操作时序 (LVIMD 位= 1)



(b) 用作中断

当允许 LVI 操作时，供应电压会与检测电压进行比较。如果供应电压低于检测电压，则会生成一个中断请求信号 (INTLVI) (当 LVIM.LVIMD 位被清零时)。

以下显示了操作设置方法及时序图。

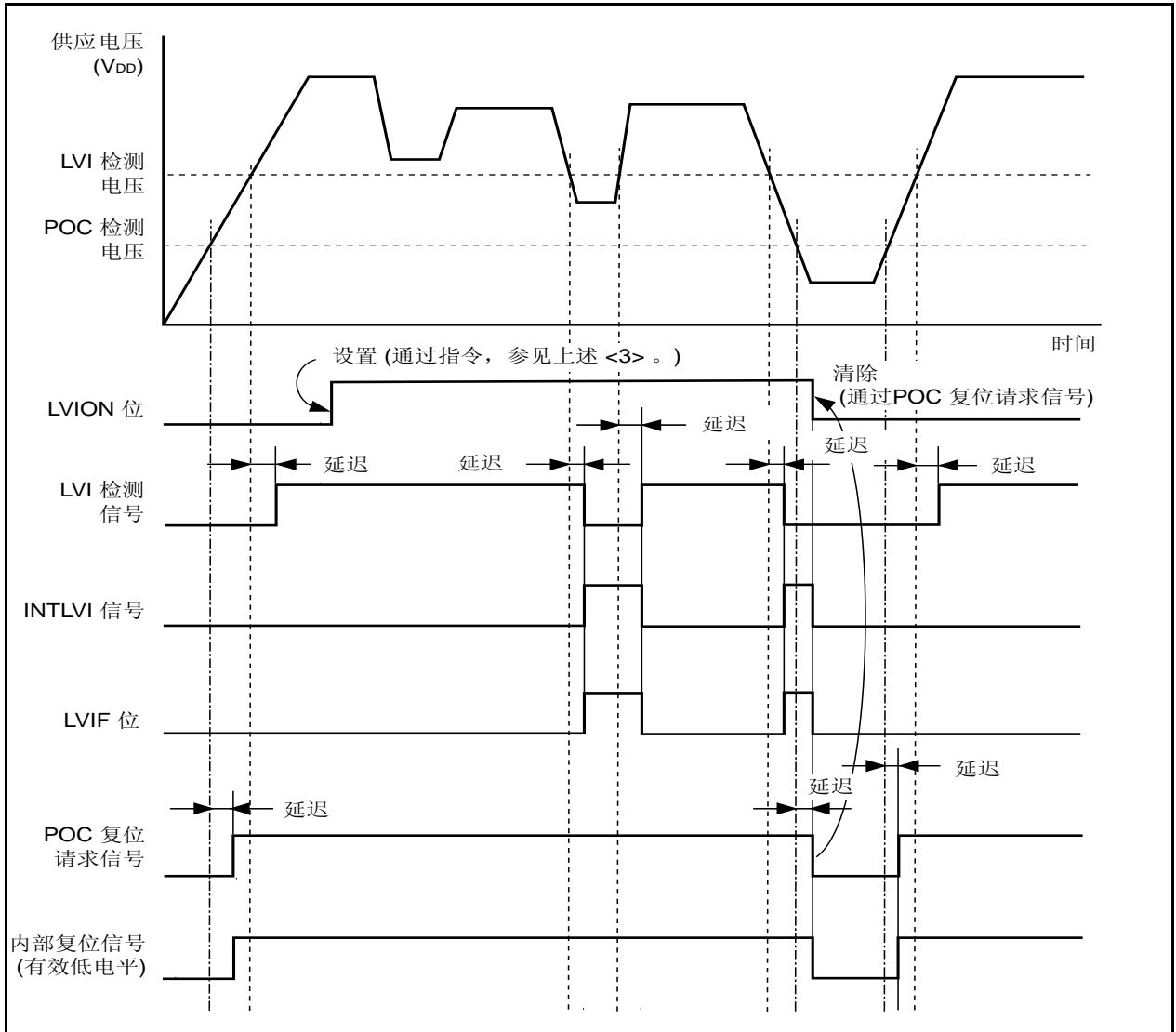
<准备开始操作>

- <1> 屏蔽 LVI 中断。
- <2> 由 LVIS.LVIS0 位选择检测的电压。
- <3> 设置 LVIM.LVION 位为 1 (以允许操作)。
- <4> 通过软件插入一个 0.1 ms (TYP) (目标值) 或更长的等待周期。
- <5> 使用 LVIM.LVIF 位，检验供应电压是否大于检测电压。
- <6> 清除 LVI 的中断请求标志。
- <7> 不屏蔽 LVI 中断。

<准备停止操作>

将 LVION 位清 0。

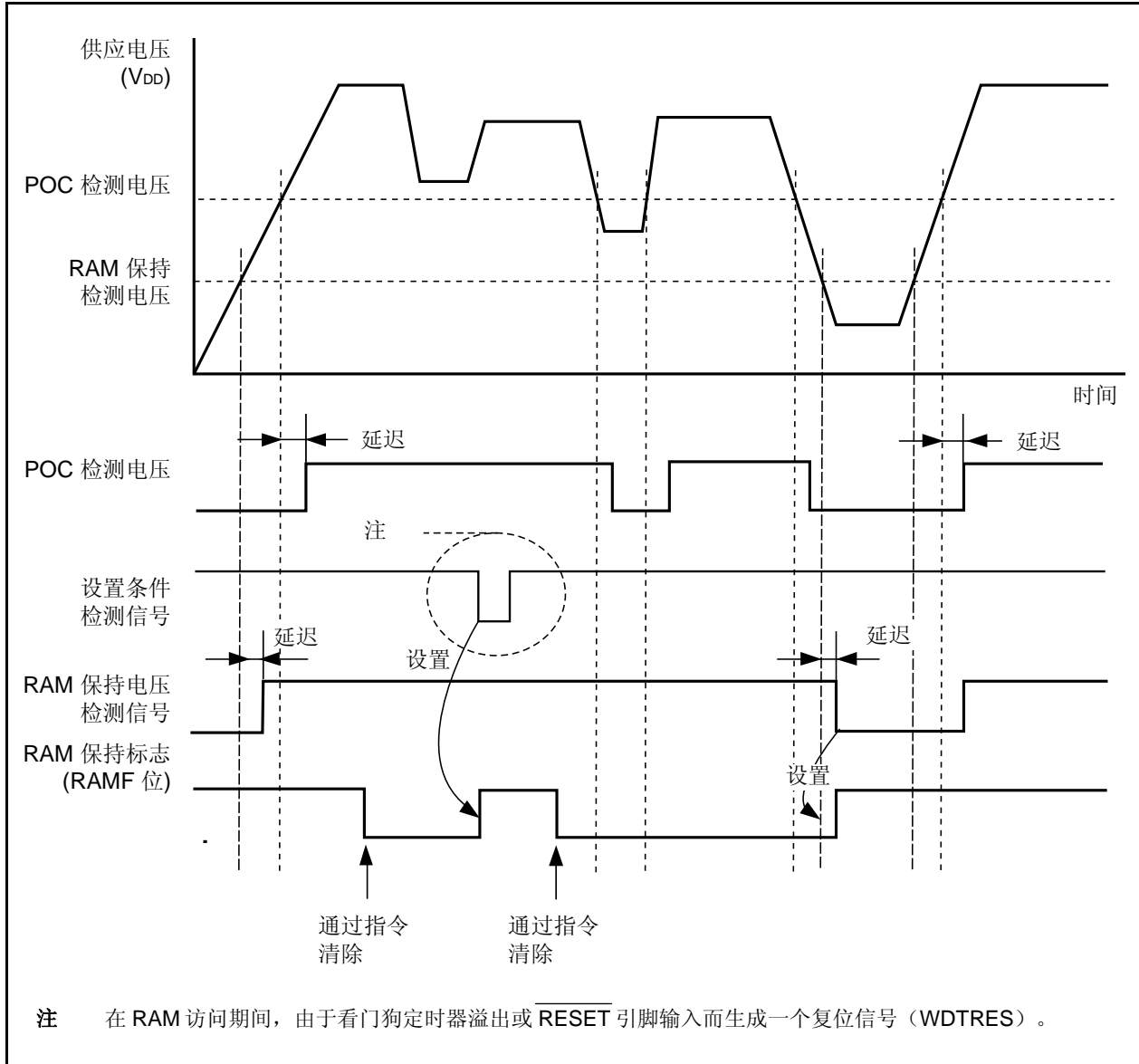
图 16-6. 低电压检测器的操作时序 (LVIMD 位= 0)



(5) RAM 保持电压检测操作

供应电压与检测电压 ($V_{RAMH} = 2.0\text{ V} \pm 0.1\text{ V}$) 进行比较。当供应电压降低低于检测电压时 (包括上电), RAMS.RAMF 位设置为 1。

图 16-7. RAM 保持电压检测功能的操作时序



16.3.4 上电清除电路 (POC)

(1) 概述

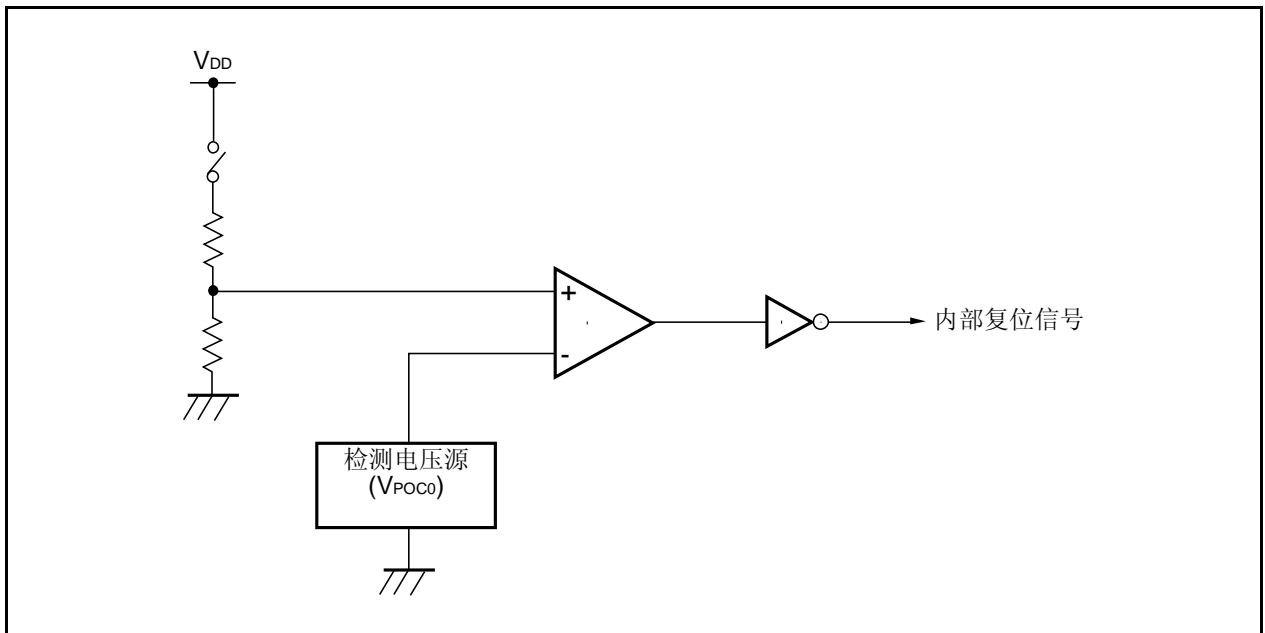
下面是上电清零电路 (POC) 概述。

- 在电源打开时生成一个复位信号。
- 将供应电压 (V_{DD}) 与检测电压 (V_{POC0}) 进行比较, 并在 $V_{DD} < V_{POC0}$ 时生成一个复位信号。

(2) 配置

功能框图如下:

图 16-8. 上电清除电路的框图



(3) 操作

当供应电压与检测电压进行比较且供应电压低于检测电压（包括电源打开时）时，系统将会被复位且每个硬件都将会返回到特定状态。

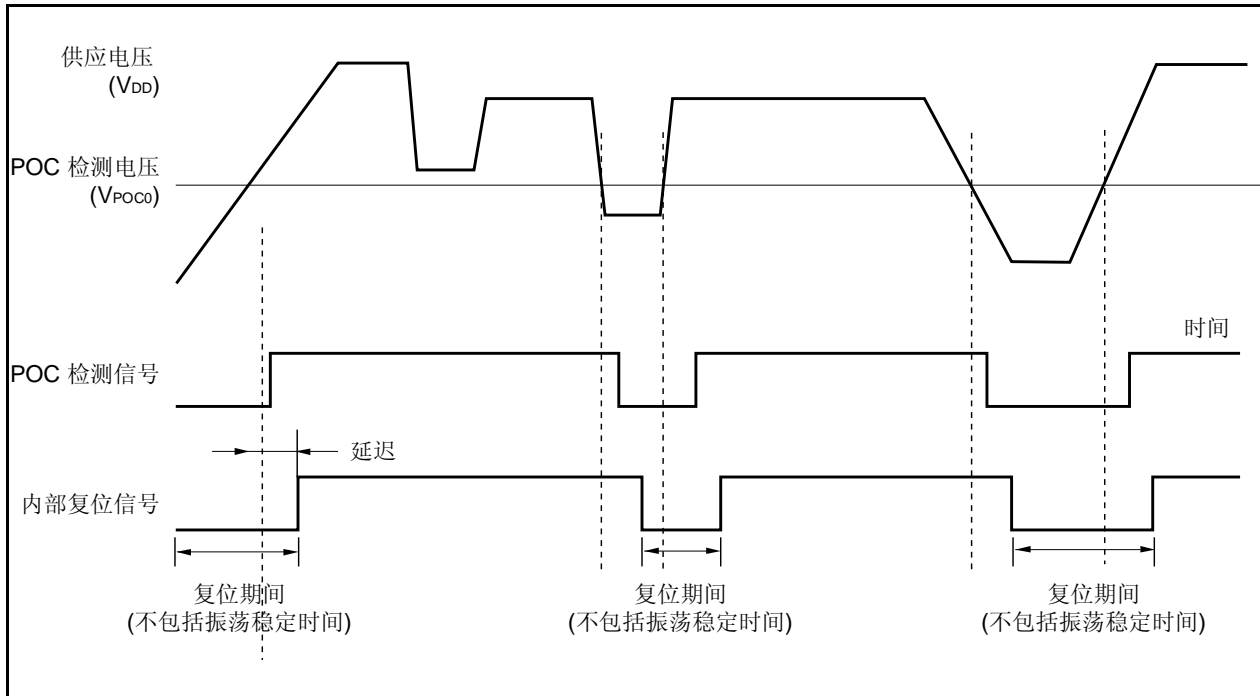
系统从检测到低电压时开始处于复位状态直到供应电压高于检测电压为止。复位解除后，在振荡器的振荡稳定时间（OSTS 寄存器的默认值： $2^{16}/f_x$ ）过后，CPU 将开始执行程序。

振荡器会在复位期间停止以保证振荡稳定时间。

在复位期间以及复位解除后各硬件的状态与 **RESET** 引脚实现复位操作的状态相同（参见 16.3.1 通过 **RESET** 引脚实现的复位操作）。

以下显示了时序图。

图 16-9. 通过上电清除电路生成复位信号的时序



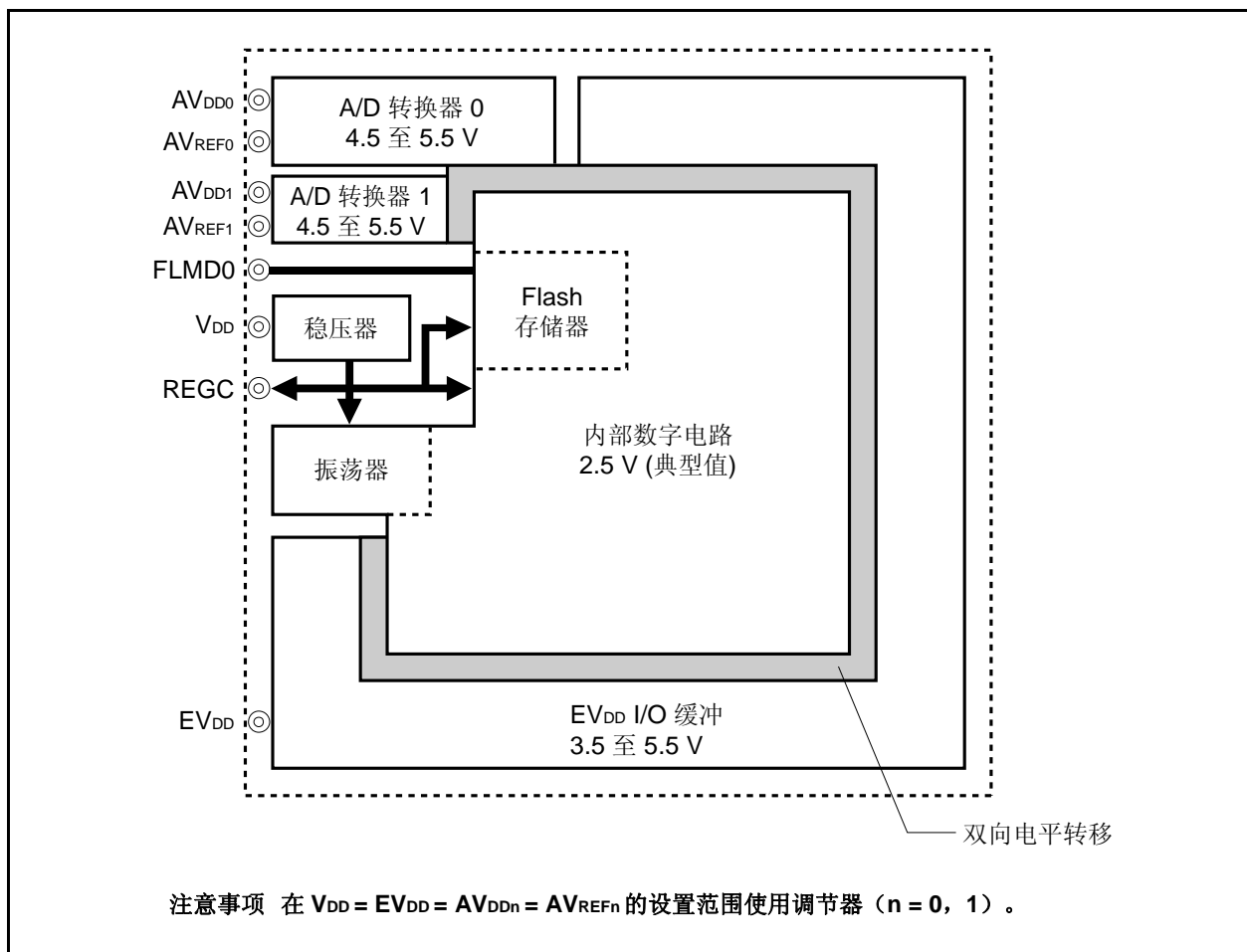
第十七章 调节器

17.1 概述

V850ES/IE2 集成了一个调节器以降低功耗和噪声。

该调节器为振荡器和内部逻辑电路（A/D 转换器和输入输出缓冲器除外）提供降压电源供电电压 V_{DD} 。该调节器输出电压（REGC 引脚）设置为 2.5V（典型值）。

图 17-1. 调节器



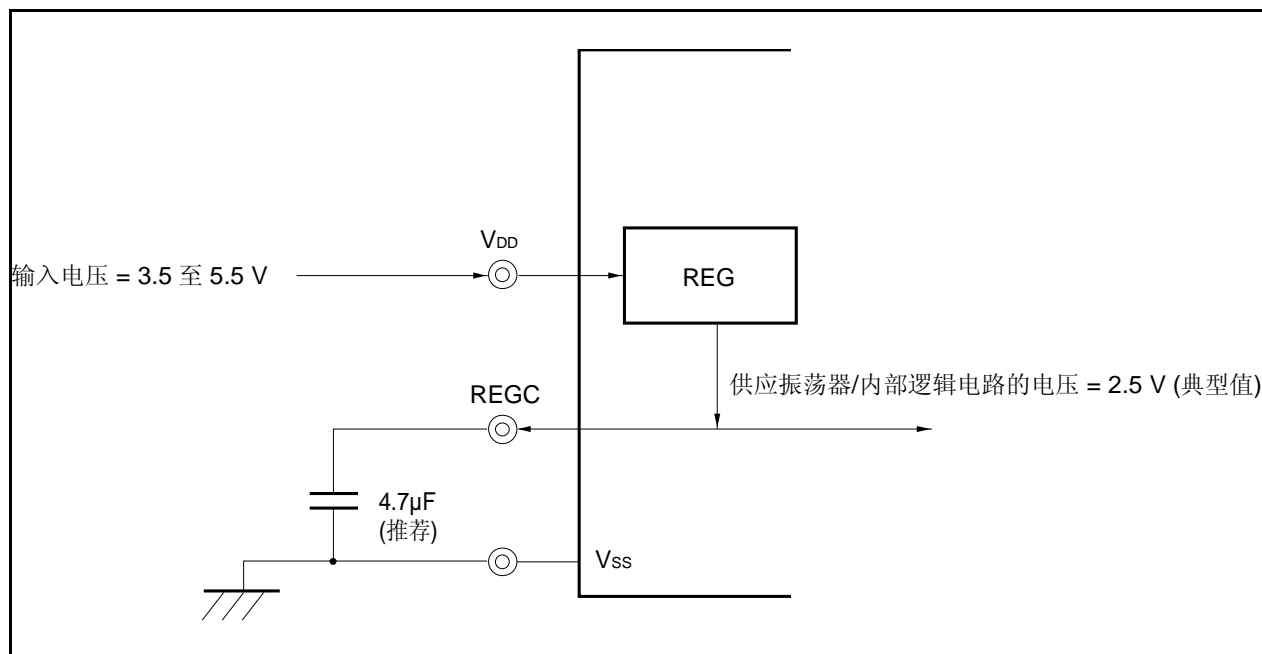
17.2 操作

该产品的调节器运行于各种模式下（正常操作模式，HALT 模式，IDLE 模式，STOP 模式或者复位时）。

确保给 REGC 引脚连接一个电容（ $4.7\ \mu\text{F}$ （推荐值））来稳定调节器输出。

调节器的引脚连接如下图所示：

图 17-2. REGC 引脚连接



第十八章 FLASH 存储器

μ PD70F3713 和 70F3714 包含内部 ROM 的大小为 64 KB 和 128 KB 的 flash 存储器。

注意事项 有关存储器相关的电气特性，参见第十九章 电气特性。

V850ES/IE2 通常应用于以下开发环境和批量生产场合：

- 用于 V850ES/IE2 单片机焊接到目标系统后进行软件修改。
- 用于开始批量生产时进行数据调整。
- 用于根据不同模型的小批量生产的要求区分软件。
- 便于库存管理。
- 用于交付之后的软件更新。

flash 存储器的取指令可以在一个时钟内访问 4 个字节。

可以通过安装在目标系统（板上）上完成对 flash 存储器的写入。使用连接到目标系统上的专用 flash 存储器编程器。

18.1 特点

- 访问速度：4 字节/每时钟（取指时）
- 容量：128 KB/64 KB
- 写入电压：独立电源擦除 / 写入
- 重写方法
 - 通过串行接口跟专用 flash 存储器编程器通信进行重写（在板/脱板编程）
 - 通过用户程序重写 flash 存储器（自编程）（仅限于 μ PD70F3714）。
- 支持 Flash 存储器写入禁止功能（保护功能）
- 通过自编程（使用引导区交换功能）实现整个 flash 存储器区域的安全重写^注（仅限于 μ PD70F3714）。

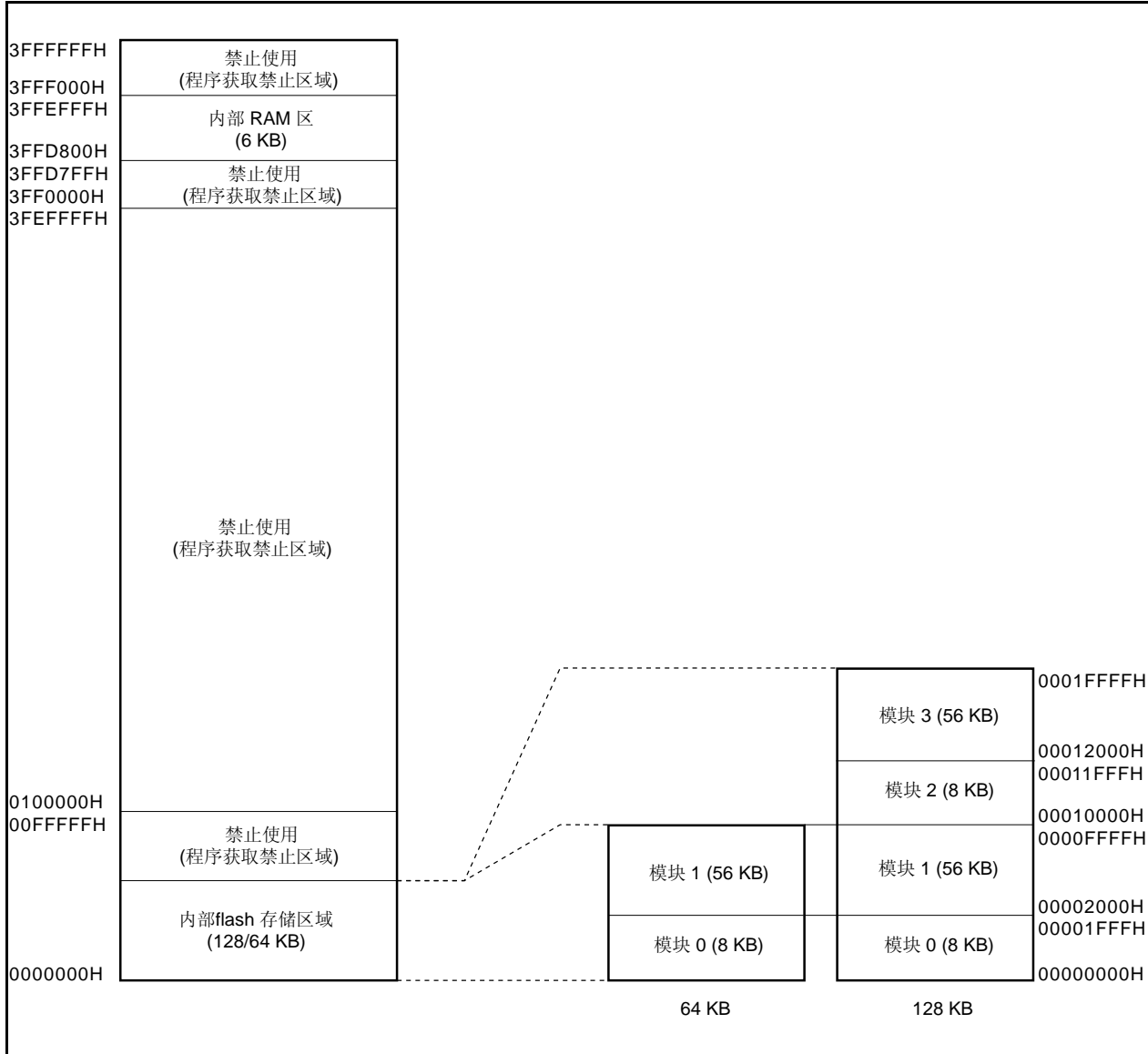
注 在 μ PD70F3713 中，不能使用引导交换功能，但可以通过自编程对整个 flash 存储器区域进行安全重写。

- 自编程期间可以响应中断。

18.2 存储器配置

128 KB/64 KB 内部 flash 存储器区域被划分为 4/2 模块，并且以模块为单位可以进行编程/擦除。所有模块也可以一次性全部擦除。

图 18-1. Flash 存储器映射



18.3 功能概述

V850ES/IE2 单片机的内部 flash 存储器可以使用专用 flash 存储器编程器的重写功能进行重写，不管单片机是不是已经安装到目标系统之上（在板/脱板编程）。

另外，也支持保护功能，该保护功能禁止重写被写入内部 flash 存储器中的用户程序，这样，程序就不会被未经授权的人改变。

使用用户程序（自编程（仅限 μ PD70F3714））的重写功能对于那些产品出库后有可能发生改变的目标系统中的应用比较理想。也支持安全重写整个 flash 存储器的引导交换功能（仅限 μ PD70F3714）。另外，在自编程期间支持中断服务，这样，flash 存储器可以在不同场合下进行重写，比如在和一个外部设备进行通信时。

表 18-1. 重写方式

重写方式	功能概述	工作模式
在板编程	设备安装到目标系统之上后，通过使用一个专用的 flash 存储器编程器对 Flash 存储器进行重写。	Flash 存储器编程模式
脱板编程	设备安装到目标系统上之前，通过使用一个专用 flash 存储器编程器和一个专用的程序适配板（FA 系列）对 Flash 存储器进行重写。	
自编程（仅限 μ PD70F3714）	执行由在板/脱板编程的方式预先写入 Flash 存储器的一个用户程序，对 Flash 存储器进行重写。（自编程期间，不能获取指令且不能访问内部 flash 存储器区域的数据。因此，重写程序必须预先发送到内部 RAM 或外部存储器）。	正常工作模式

备注 FA 系列是 Naito Densei Machida Mfg.Co., Ltd 的产品。

表 18-2. 基本功能

功能	功能概述	支持 (√: 支持, ×: 不支持)	
		在板/脱板编程	自编程
块擦除	指定存储模块的内容被擦除。	√	√
片擦除	存储区的内容一次性整体擦除。	√	×
写	写入指定地址, 并且执行校验, 检查写入电平是否安全。	√	√
校验/校验和	从 flash 存储器读出的数据和从 flash 存储器编程器传输的数据相比较。	√	× (可以由用户程序读取)
空白检查	检查整个存储区的擦除状态	√	√
<R> 安全设置	禁止使用块擦除命令、片擦除命令、编程命令、读命令以及引导区重写命令。	√	× (仅支持从允许到禁止的设置)

以下表格列举了安全功能。块擦除命令禁止、片擦除命令禁止以及编程命令禁止等功能在本产品交货时默认为允许, 可以由在板/脱板编程进行安全设置。每一项安全功能都可以同时联合其它安全功能使用。

表 18-3. 安全功能

功能	功能概述
块擦除命令禁止	所有模块禁止执行一个块擦除命令。 可以通过执行片擦除命令来初始化禁止的设置。
片擦除命令禁止	对所有模块的块擦除命令以及片擦除命令的执行被禁止。一旦该禁止功能被设置, 禁止的设置就无法再次初始化, 因为片擦除命令不能执行。
编程命令禁止	对所有模块的编程和块擦除命令被禁止执行。可以通过执行片擦除命令来初始化禁止的设置。
读取命令禁止	对所有模块的读取命令被禁止执行。 可以通过执行片擦除命令来初始化禁止的设置。
引导区重写禁止	从模块 0 至指定的最末模块的引导区不被保护。被保护的引导区不能进行重写(擦除和写入)。执行片擦除命令不能初始化禁止的设置。

<R>

表 18-4. 安全设置

功能	各项安全功能设置时的擦除、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	
	在板/脱板编程	自编程	在板/脱板编程	自编程
块擦除命令禁止	块擦除命令: × 片擦除命令: √ 编程命令: √ 读取命令: √	块擦除 (flash 块擦除): √ 片擦除: - 写入 (flash 字写入): √ 读取 (flash 字读取): √	通过片擦除命令使禁止的设置初始化	仅支持从允许到禁止的设置
片擦除命令禁止	块擦除命令: × 片擦除命令: × 编程命令: √ ¹ 读取命令: √	块擦除 (flash 块擦除): √ 片擦除: - 写入 (flash 字写入): √ 读取 (flash 字读取): √	禁止的设置不能被初始化	
编程命令禁止	块擦除命令: × 片擦除命令: √ 编程命令: × 读取命令: √	块擦除 (flash 块擦除): √ 片擦除: - 写入 (flash 字写入): √ 读取 (flash 字读取): √	通过片擦除命令使禁止的设置初始化	
读取命令禁止	块擦除命令: √ 片擦除命令: √ 编程命令: √ 读取命令: ×	块擦除 (flash 块擦除): √ 片擦除: - 写入 (flash 字写入): √ 读取 (flash 字读取): √	通过片擦除命令使禁止的设置初始化	
引导区重写禁止	块擦除命令: √ ² 片擦除命令: × 编程命令: √ ² 读取命令: √	块擦除 (flash 块擦除): √ 片擦除: - 写入 (flash 字写入): √ 读取 (flash 字读取): √	禁止的设置不能被初始化	

注 1. 这种情况下, 由于擦除命令无效, 与已经写入 Flash 存储器的数据不同的数据不能被写入。
2. 对引导区的重写禁止是无效的。

<R> (1) 由 PG-FP5 进行安全设置（安全标志设置）

当禁止读取命令时（读禁止），要提高安全水平，建议也要禁止块擦除命令（块擦除禁止）和编程命令（编程禁止）。

另外，如果产品像掩模产品一样不需进行重新编程，则可以将片擦除命令禁止（片擦除禁止）。

The screenshot shows the 'Properties' dialog box with the 'Advanced' tab selected. The 'Command options' section is highlighted with a dashed box and labeled 'Note 1'. The 'Security flag settings' section is also highlighted with a dashed box and labeled 'Note 2'. The 'Program download/upload' section has 'Enable two Program Areas' checked.

Note 1 points to the 'Command options' section, which includes:

- Blank check before Erase
- Read verify after Program
- Security flag after Program
- Checksum after Program

Note 2 points to the 'Security flag settings' section, which includes:

- Disable Chip Erase
- Disable Block Erase
- Disable Program
- Disable Read
- Disable Boot block cluster reprogramming

The 'Reset vector' is set to 000000 h. The 'End Boot block' is set to 0. The 'Show Address' checkbox is unchecked.

Note

1. 根据使用情况，设置虚线框内的“供电电压”“编程下载/上传”以及“命令选项”。
2. 为了禁止重写引导区（引导区模块簇的设置），在“安全标志设置”里选择“禁止引导区模块簇再编程”并选择禁止重写的引导区的最后模块。

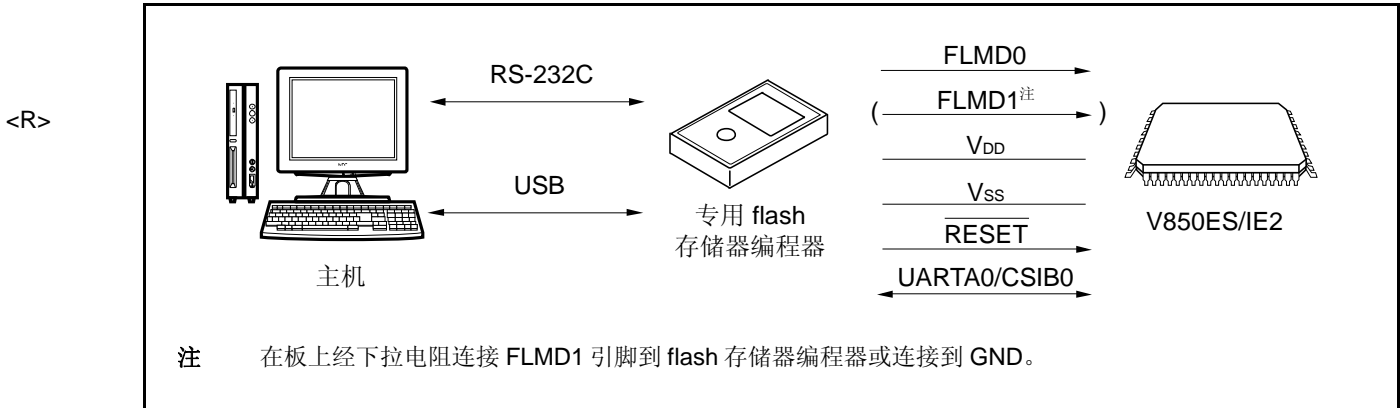
18.4 通过专用Flash 存储器编程器重写

V850ES/IE2 单片机安装于目标系统之上后（在板编程），可以使用专用 flash 存储器编程器对 flash 存储器进行重写。flash 存储器也可以使用专用的编程适配器（FA 系列），于设备安装在目标系统之前被重写（脱板编程）。

18.4.1 编程环境

以下显示将程序写入 V850ES/IE2 单片机的 flash 存储器时所需的环境：

图 18-2. Flash 存储器编程所需环境



需要一台主机控制专用 flash 存储器编程器的工作。

UARTA0 或 CSIB0 用作专用 flash 存储器编程器和 V850ES/IE2 单片机之间的接口，通过它执行对单片机的写入、擦除等。脱板写入则需要一个专用的编程适配器（FA 系列）。

备注 FA 系列是 Naito Densai Machida Mfg.Co., Ltd 的产品。

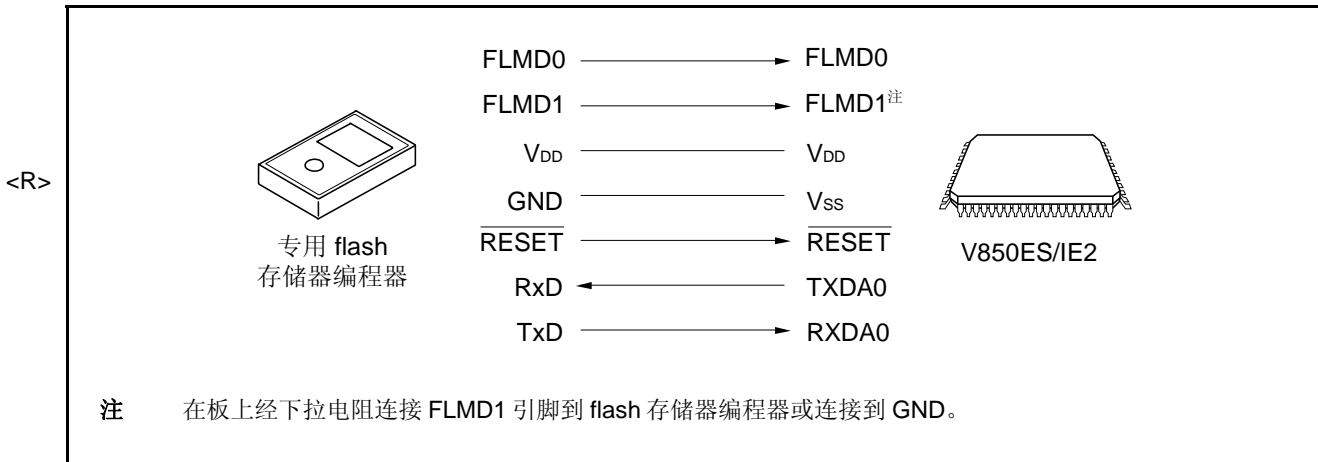
18.4.2 通信模式

使用 V850ES/IE2 单片机的 UARTA0 或 CSIB0 接口，通过串行通信的方式在单片机与专用 flash 存储器编程器间进行通信。

(1) UARTA0

传输速率： 9, 600 至 153, 600 bps

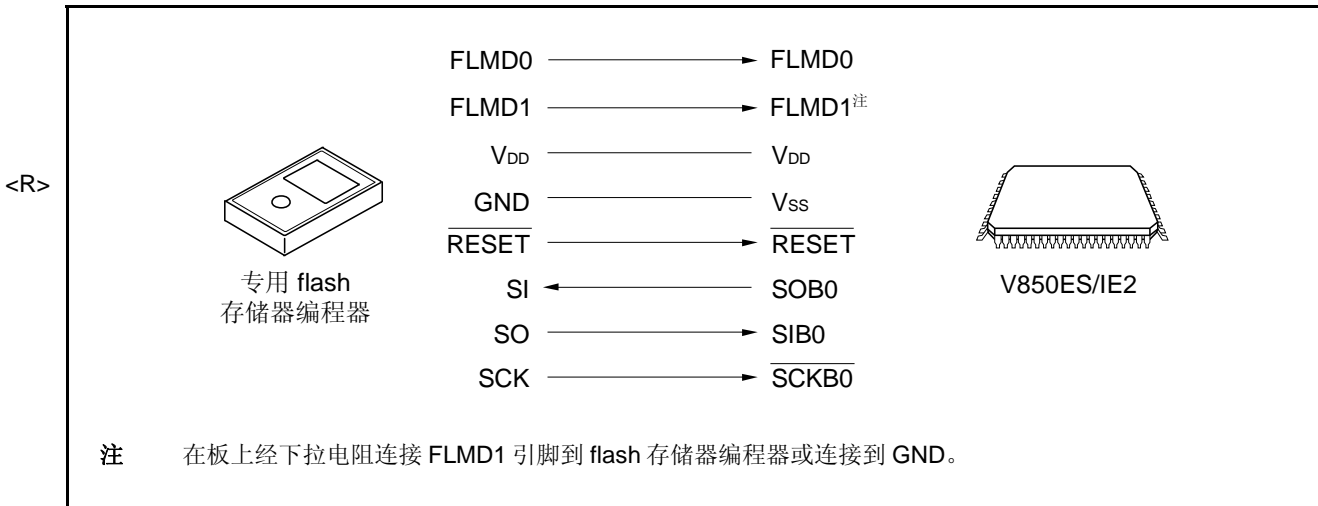
图 18-3. 与专用 Flash 存储器编程器的通信 (UARTA0)



(2) CSIB0

串行时钟： 2.4 kHz 至 2.5 MHz (MSB 优先)

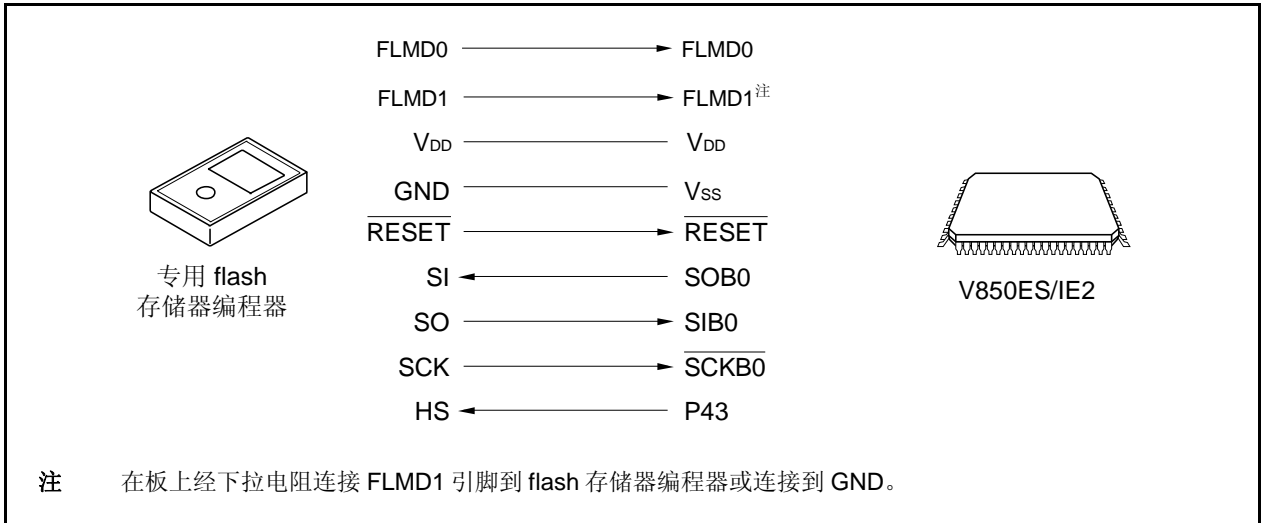
图 18-4. 与专用 Flash 存储器编程器的通信 (CSIB0)



(3) CSIB0 + HS

串行时钟： 2.4 kHz 至 2.5 MHz（MSB 在先）

图 18-5. 与专用 Flash 存储器编程器的通信（CSIB0 + HS）



专用 flash 存储器编程器输出传输时钟，并且 V850ES/IE2 单片机工作于从机模式。

当 PG-FP4 或 PG-FP5 用作专用 flash 存储器编程器时，其将发送以下信号至 V850ES/IE2。有关细节，参见 PG-FP4 用户手册（U15260E）和 PG-FP5 用户手册（U18865E）。

表 18-5. 专用 Flash 存储器编程器（PG-FP4 和 PG-FP5）的连接信号

PG-FP4 和 PG-FP5			V850ES/IE2	连接处理		
信号名称	I/O	引脚功能	引脚名称	UARTA0	CSIB0	CSIB0 + HS
FLMD0	输出	写入允许/禁止	FLMD0	○	○	○
FLMD1	输出	写入允许/禁止	FLMD1	○ ^{注 1}	○ ^{注 1}	○ ^{注 1}
VDD	-	V _{DD} 电压发生/电压监视	V _{DD} , EV _{DD}	○ ^{注 2}	○ ^{注 2}	○ ^{注 2}
GND	-	地	V _{SS}	○	○	○
CLK ^{注 3}	输出	-	X1 ^{注 3} , X2 ^{注 3}	-	-	-
RESET	输出	复位信号	RESET	○	○	○
SI/RxD	输入	接收信号	SOB0/TXDA0	○	○	○
SO/TxD	输出	发送信号	SIB0/RXDA0	○	○	○
SCK	输出	发送时钟	SCKB0	x	○	○
HS	输入	给 CSIB0 + HS 握手信号通信	P43	x	x	○

- 注
1. 连接引脚如图 18-6 或通过一个下拉电阻将它们连接到板上 GND。
 2. 从 PG-FP4 和 PG-FP5 中提供电源时连接至该引脚。
如果使用了板上电源，则不必连接该引脚。
 3. 在 V850ES/IE2 中，禁止外部时钟输入。 创建一个板上振荡器并通过该振荡器来提供时钟。

备注
 ○： 必须连接
 x： 不必连接

表 18-6. V850ES/IE2 Flash 写入适配器 (FA-64GC-8BS-A) 的布线

Flash 存储器编程器 (PG-FP4 和 PG-FP5) 连接引脚			FA 板上的 引脚名	使用CSIB0 + HS		使用CSIB0		使用UARTA0	
信号名称	I/O	引脚功能		引脚名称	引脚编号	引脚名称	引脚编号	引脚名称	引脚编号
SI/RxD	输入	接收信号	SI	P41/SOB0	39	P41/SOB0	39	P31/TXDA0	43
SO/TxD	输出	发送信号	SO	P40/SIB0	40	P40/SIB0	40	P30/RXDA0	44
SCK	输出	发送时钟	SCK	P42/SCKB0	38	P42/SCKB0	38	不需要	-
CLK [‡]	输出	V850ES/IE2的 时钟	X1 [‡]	X1 [‡]	6	X1 [‡]	6	X1 [‡]	6
			X2 [‡]	X2 [‡]	7	X2 [‡]	7	X2 [‡]	7
/RESET	输出	复位信号	/RESET	RESET	5	RESET	5	RESET	5
FLMD0	输入	写入电压	FLMD0	FLMD0	25	FLMD0	25	FLMD0	25
FLMD1	输入	写入电压	FLMD1	PDL5/FLMD1	30	PDL5/FLMD1	30	PDL5/FLMD1	30
HS	输入	CSI0 + HS的 握手信号	RESERVE/ HS	P43/TOP00/ TIP00	37	不需要	-	不需要	-
VDD	-	VDD电压产生 /电压监视	VDD	VDD	9	VDD	9	VDD	9
				EVDD	26, 47	EVDD	26, 47	EVDD	26, 47
				AVREF0	64	AVREF0	64	AVREF0	64
				AVREF1	59	AVREF1	59	AVREF1	59
				AVDD0	63	AVDD0	63	AVDD0	63
			AVDD1	60	AVDD1	60	AVDD1	60	
GND	-	接地	GND	VSS	8	VSS	8	VSS	8
				EVSS	27, 48	EVSS	27, 48	EVSS	27, 48
				AVSS0	62	AVSS0	62	AVSS0	62
				AVSS1	61	AVSS1	61	AVSS1	61

注 flash 存储器编程器的 CLK 引脚不能提供时钟。
创建一个板上振荡器并通过该振荡器来提供时钟。

注意事项 确保以下列两种方式之一连接 REGC 引脚。

- 通过 4.7 μ F 电容连接 GND
- 直接连接 VDD

图 18-6. V850ES/IE2 单片机 Flash 写入适配器 (FA-64GC-8BS-A) 的布线举例 (1/2)

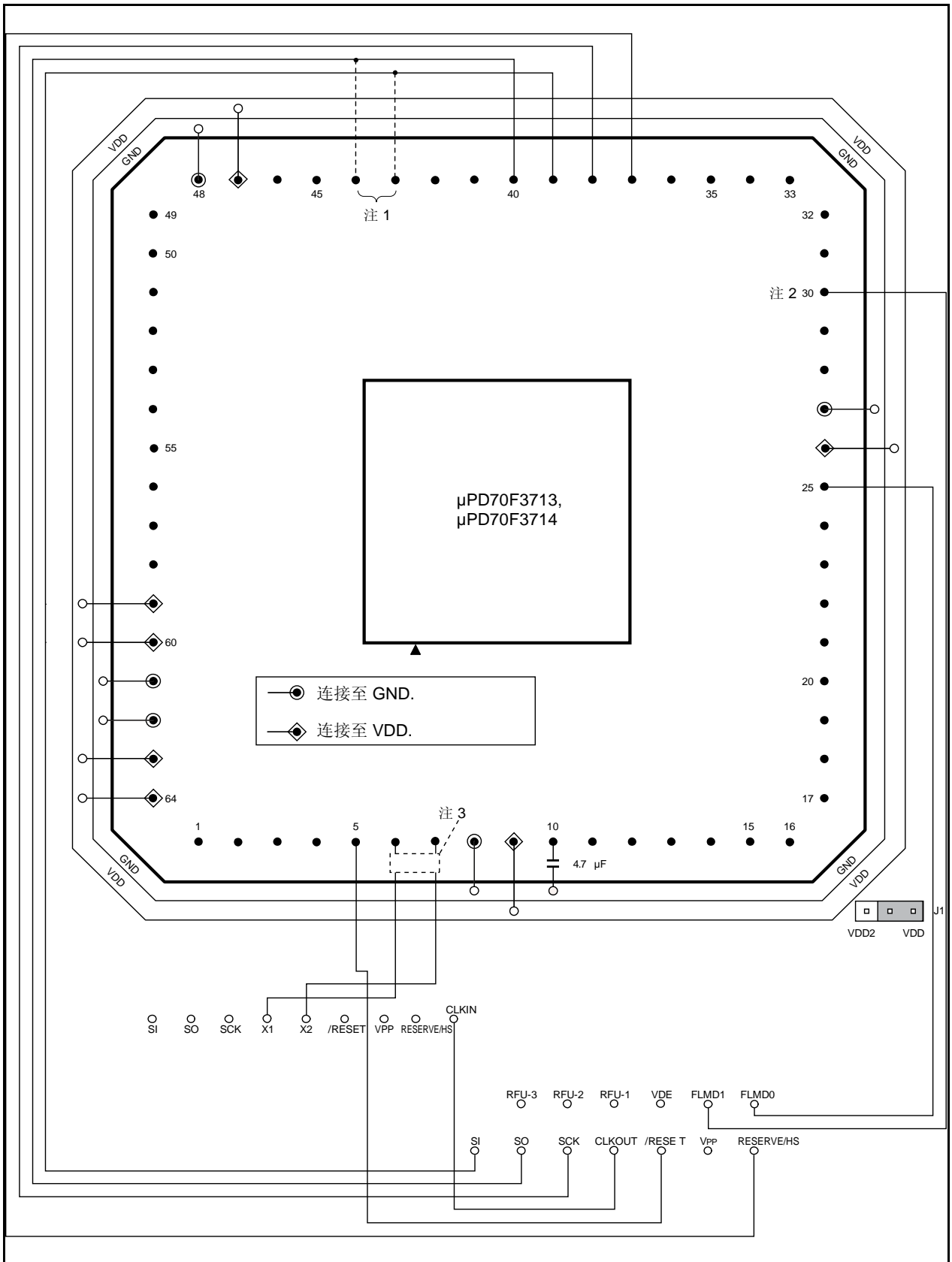
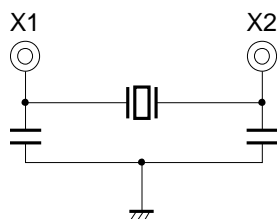


图 18-6. V850ES/IE2 单片机 Flash 写入适配器 (FA-64GC-8BS-A) 的布线举例 (2/2)

- 注
1. 使用 UARTA0 时的相应引脚。
 2. 如下图中所示连接 FLMD1 引脚，或通过一个下拉电阻将它连接到板上 GND。
 3. 在 flash 写入适配器（以虚线表示）上创建一个振荡器并提供一个时钟。
这是一个振荡器示例。

例：



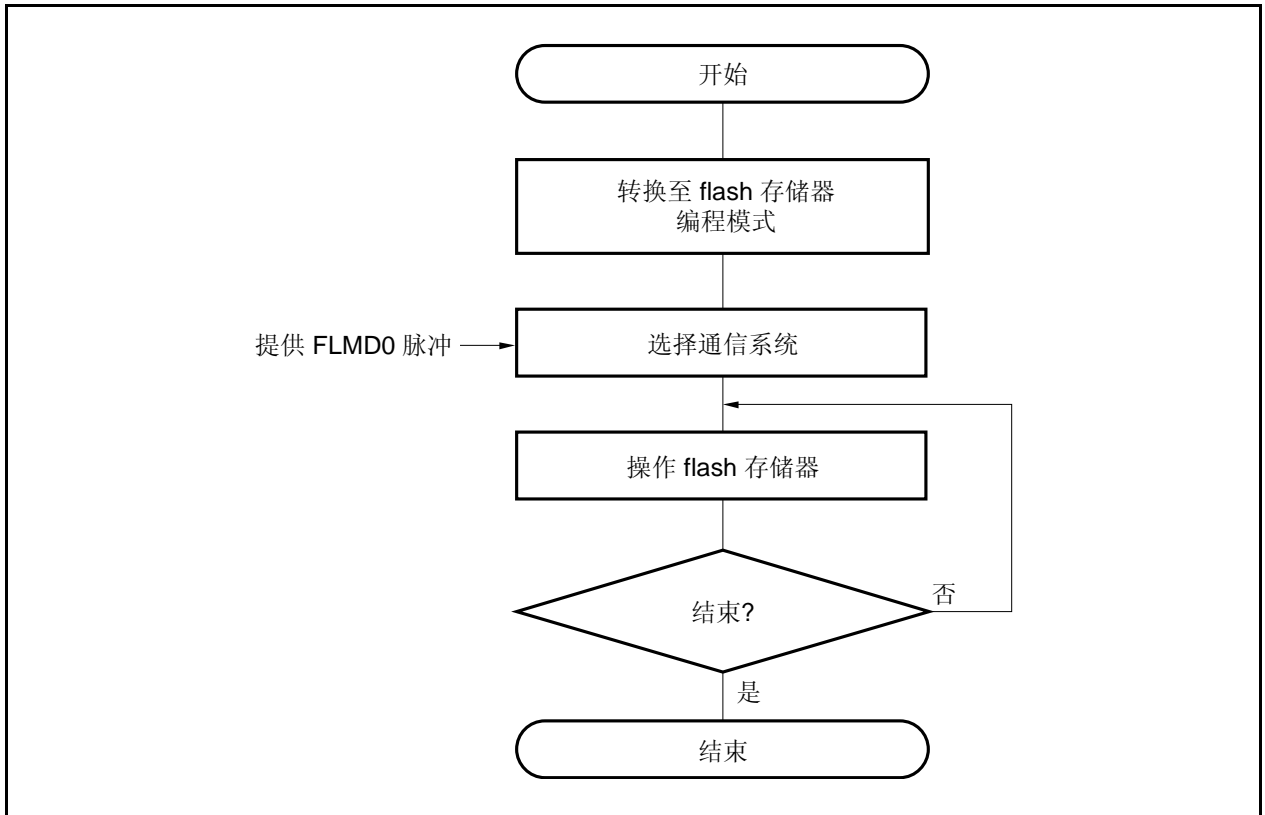
- 备注
1. 根据未使用引脚的处理来处理没有显示的引脚（参见 2.2 引脚 I/O 电路类型和未使用引脚的连接）。建议使用一个 1 k Ω 至 10 k Ω 的电阻。
 2. 该图显示了使用 CSIB0 支持握手操作时的布线。

18.4.3 Flash 存储器控制

下图显示了 flash 存储器的操作流程:

<R>

图 18-7. Flash 存储器的操作流程

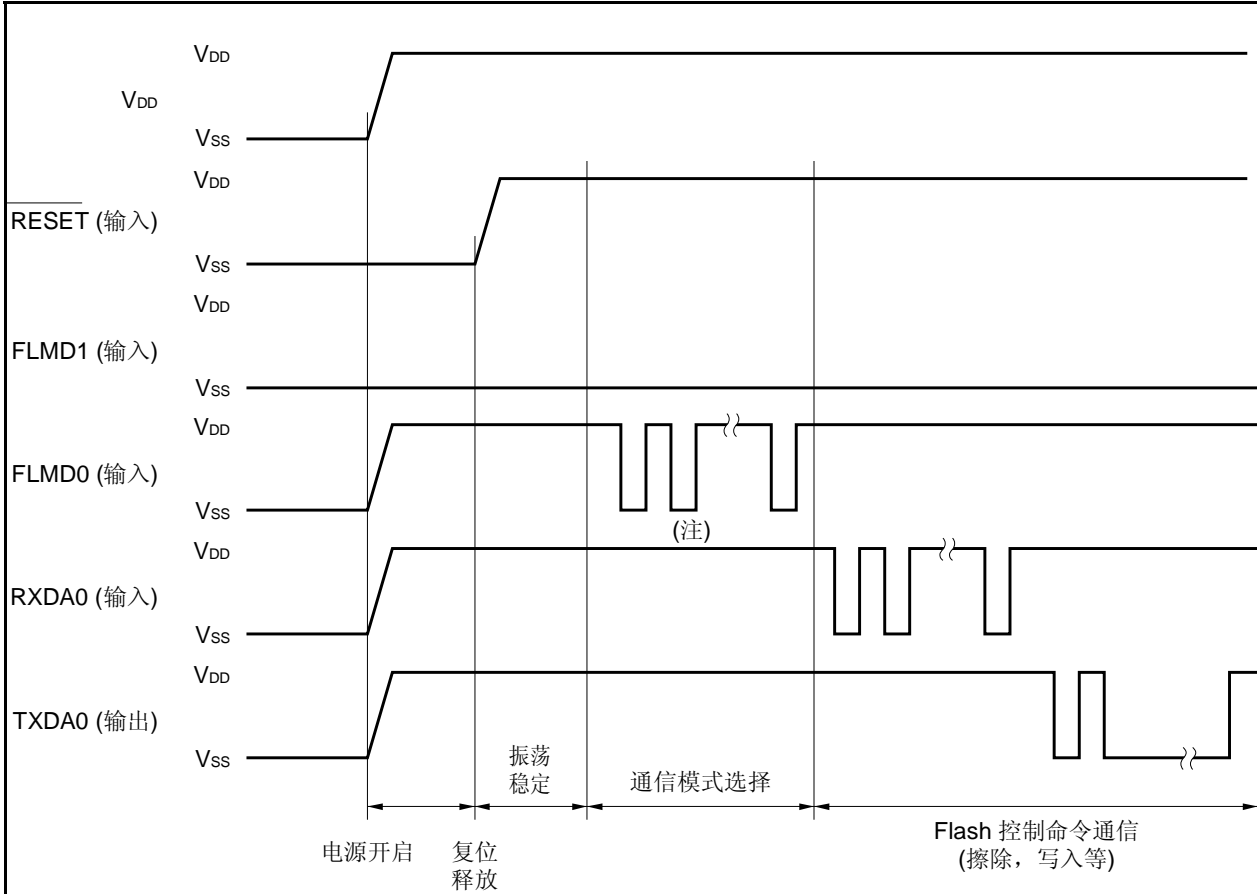


18.4.4 通信模式的选择

在 V850ES/IE2 中，进入 flash 存储器编程模式后，通过将脉冲（可多达 11 个脉冲）输入到 FLMD0 引脚，来选择通信模式。由专用 flash 存储器编程器产生 FLMD0 脉冲。

以下显示脉冲数目和通信模式之间的关系：

图 18-8. 通信模式的选择



注 根据不同的通信模式，时钟数目如下：

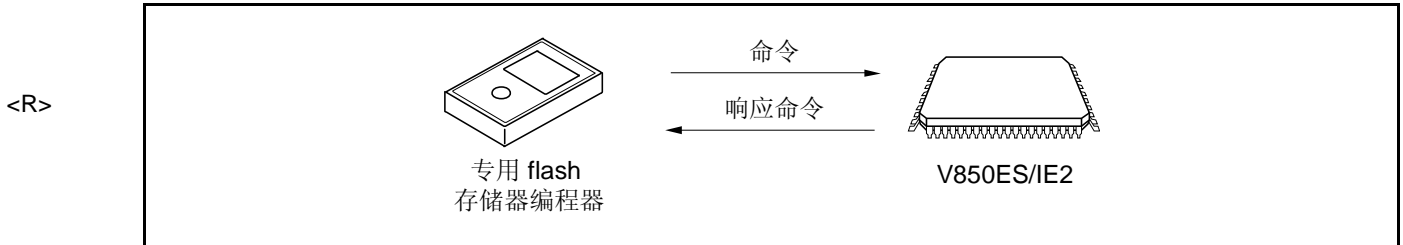
FLMD0 脉冲	通信模式	备注
0	UARTA0	通信速率： 9, 600 bps（复位后），LSB 在先
8	CSIB0	V850ES/IE2 执行从机操作，MSB 在先
11	CSIB0 + HS	V850ES/IE2 执行从机操作，MSB 在先
其它	RFU	设置禁止

注意事项 当选择 UARTA0 后，接收时钟根据收到 FLMD0 脉冲后从专用 flash 存储器编程器发出的复位命令来计算得到。

18.4.5 通信命令

V850ES/IE2 单片机通过命令和专用 flash 存储器编程器进行通信。从专用 flash 存储器编程器发送到 V850ES/IE2 中的信号被称作“命令”。从 V850ES/IE2 发送到专用 flash 存储器编程器中的响应信号被称作“响应命令”。

图 18-9. 通信命令



下面所示为在 V850ES/IE2 中 flash 存储器控制的命令。所有的这些命令从专用 flash 存储器编程器发出，V850ES/IE2 根据这些命令执行处理。

表 18-7. Flash 存储器控制命令

分类	命令名称	支持			功能
		CSIB0	CSIB0 + HS	UARTA0	
空白检查	块空白检查命令	√	√	√	检查指定块的存储内容是否被正确擦除
擦除	片擦除命令	√	√	√	擦除整个存储器内容
	块擦除命令	√	√	√	擦除指定块的存储内容
写	编程命令	√	√	√	写入指定的地址范围并执行内容校验检查
验证	验证命令	√	√	√	将 Flash 存储器编程器输出的数据和指定地址范围的存储内容进行比较
	校验和命令	√	√	√	读取指定地址范围存储数据的校验和
系统设置, 控制	“硅签字”命令	√	√	√	读取“硅签字”信息
	安全设置命令	√	√	√	禁止片擦除命令, 块擦除命令, 编程命令, 读取命令以及引导区重写。

18.4.6 引脚连接

进行板上写入操作时，需在目标系统上装配一个连接器用来连接专用 flash 存储器编程器。同时，也需带有板上从正常操作模式到 flash 存储器编程模式转换的功能。

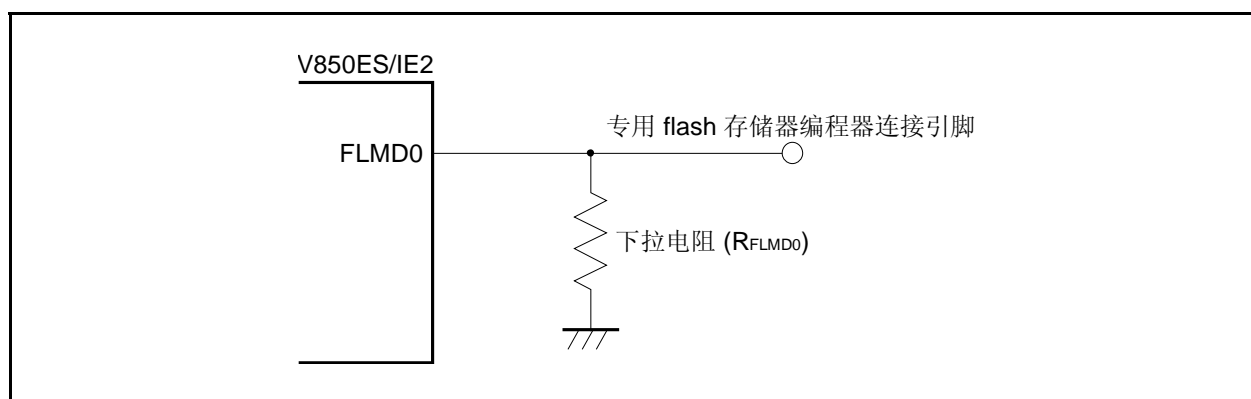
在 flash 存储器编程模式，所有那些不用于 flash 存储器存储器编程的引脚状态与复位后的状态相同。因此外部设备不能识别立即复位后的状态时，则必须采用如下方式连接引脚。

(1) FLMD0 引脚

在正常操作模式中，FLMD0 引脚输入 V_{SS} 电压。在 Flash 存储器编程模式中，FLMD0 引脚被供应一个 V_{DD} 写入电压。

因为 FLMD0 引脚在自编程模式下（仅限于 μ PD70F3714）作为写保护引脚，写入 flash 存储器之前，V_{DD} 电压必须通过端口控制以及其它方式提供给 FLMD0 引脚。有关细节，参见 18.5.5 (1) FLMD0 引脚。

图 18-10. FLMD0 引脚连接举例



(2) FLMD1 引脚

当 FLMD0 引脚输入 0 V 时，FLMD1 引脚不起作用。当 FLMD0 引脚输入 V_{DD} 电压时，进入 flash 存储器编程模式，因而 FLMD1 引脚必须输入 0 V 电压。以下显示 FLMD1 引脚连接举例：

图 18-11. FLMD1 引脚连接举例

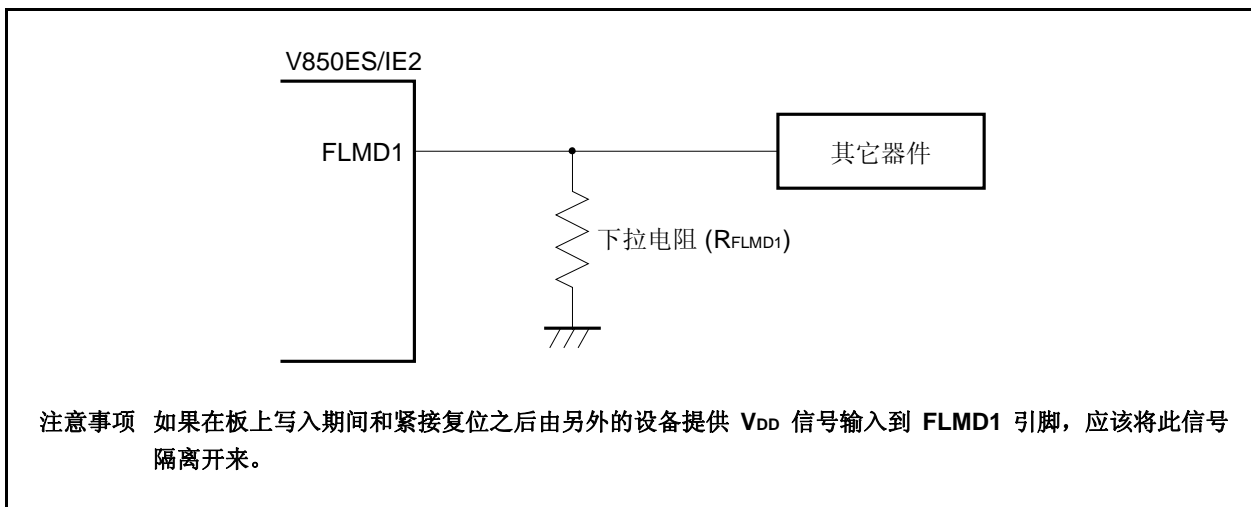


表 18-8. FLMD0 引脚和 FLMD1 引脚间的关系以及复位解除后的操作模式

FLMD0	FLMD1	工作模式
0	无需理会	正常工作模式
V_{DD}	0	Flash 存储器编程模式
V_{DD}	V_{DD}	设置禁止

(3) 串行接口引脚

以下显示了用于各串口的引脚：

表 18-9. 用于串行接口的引脚

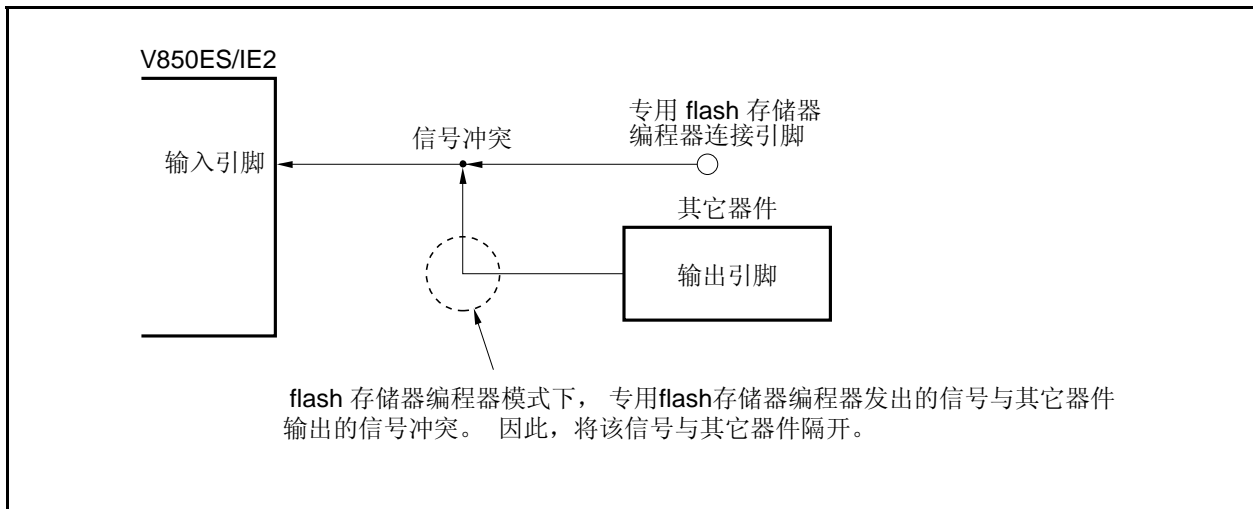
串行接口	所用引脚
UARTA0	TXDA0, RXDA0
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, P43

当专用 flash 存储器编程器连接到一个串口引脚时，而该串行接口已经连接到板上其它器件，必须注意避免信号冲突以及其它器件故障。

(a) 信号冲突

当专用 flash 存储器编程器（输出）连接一个已经连接另外器件（输出）的串行接口引脚时，会发生信号冲突。为了避免信号冲突，应该隔离与其它器件的连接或设置其它器件为输出高阻状态。

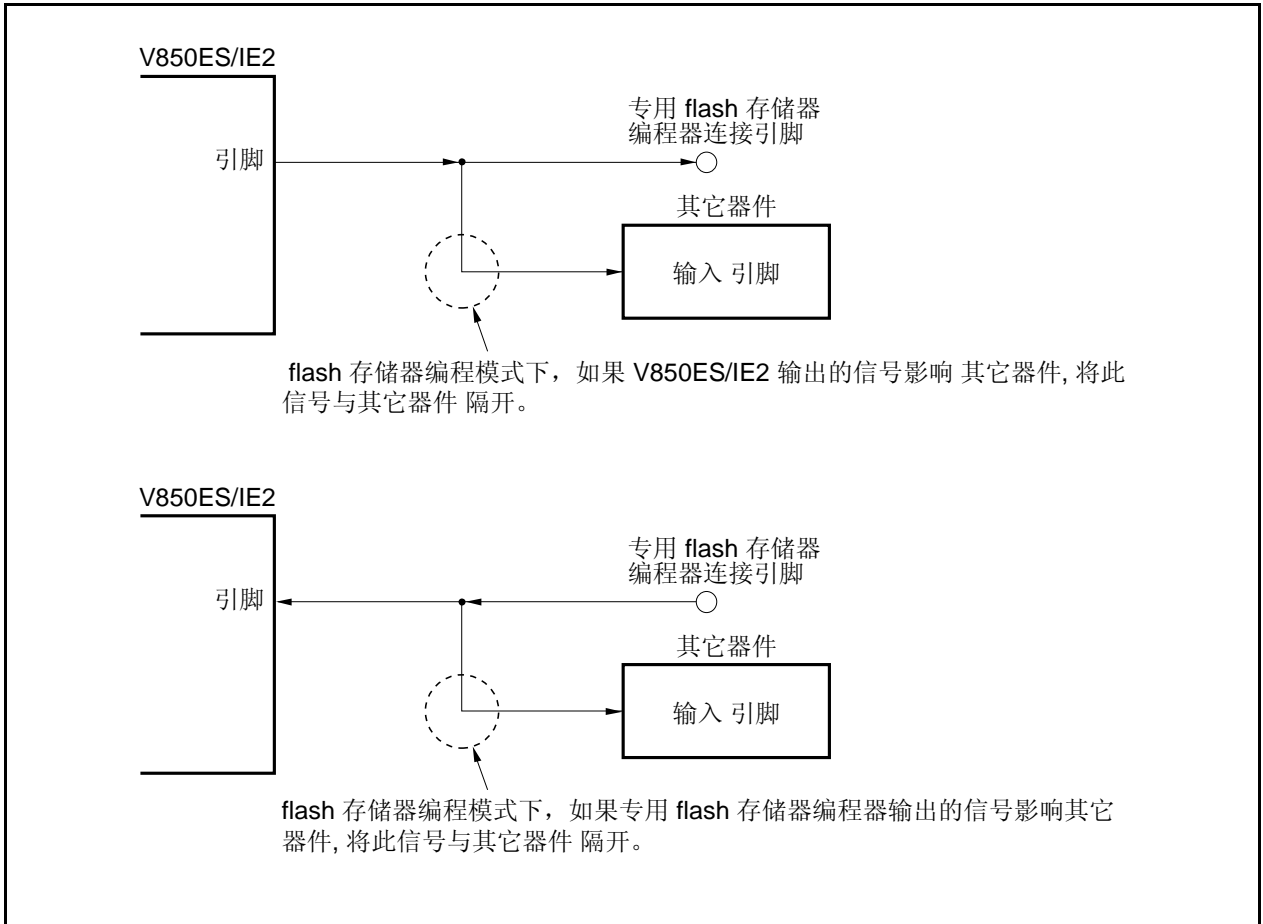
图 18-12. 信号冲突（串行接口输入引脚）



(b) 其它器件的故障

当专用 Flash 存储器编程器（输入或输出） 连接到串行接口引脚（输入或输出）， 而该串行接口还被连接到另外一个器件（输入）， 此时信号会输出到其它器件， 引起其它器件故障。为了避免这种情况发生， 隔离与其它器件的连接。

表 18-13. 其它器件的故障

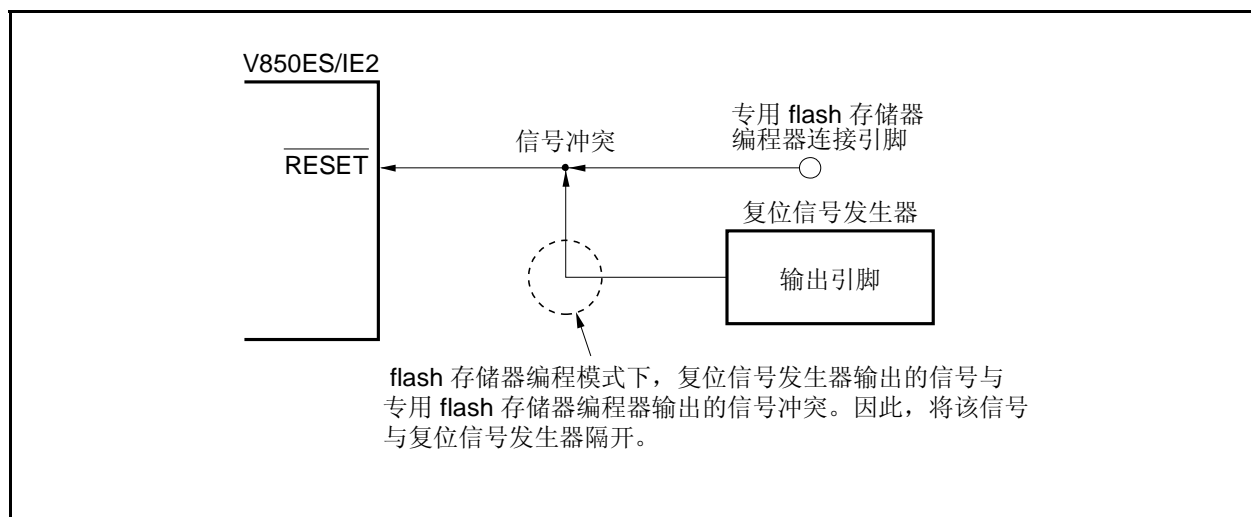


(4) RESET 引脚

专用 flash 存储器编程器的复位信号连接到 RESET 引脚时，该引脚已连接到板上的复位信号发生器时，会产生信号冲突。为了避免信号冲突，应隔离与复位信号发生器的连接。

在 flash 存储器编程模式下，当一个复位信号由用户系统输入时，编程操作将不会正确执行。因此，除了来自于专用 flash 存储器编程器的复位信号以外，不要输入其它信号。

图 18-14. 信号冲突 (RESET 引脚)



(5) 端口引脚

当系统转移到 Flash 存储器编程模式时，所有不用于 Flash 存储器编程的引脚处于与刚刚复位后相同的状态。如果连接各端口的设备不能识别端口刚刚复位后的状态，需要对引脚进行恰当的处理，比如通过一个电阻连接 V_{DD} 或通过一个电阻连接 V_{SS}。

(6) 其它信号引脚

同正常操作模式一样的情况连接 X1、X2 和 REGC 引脚。

(7) 供电电压

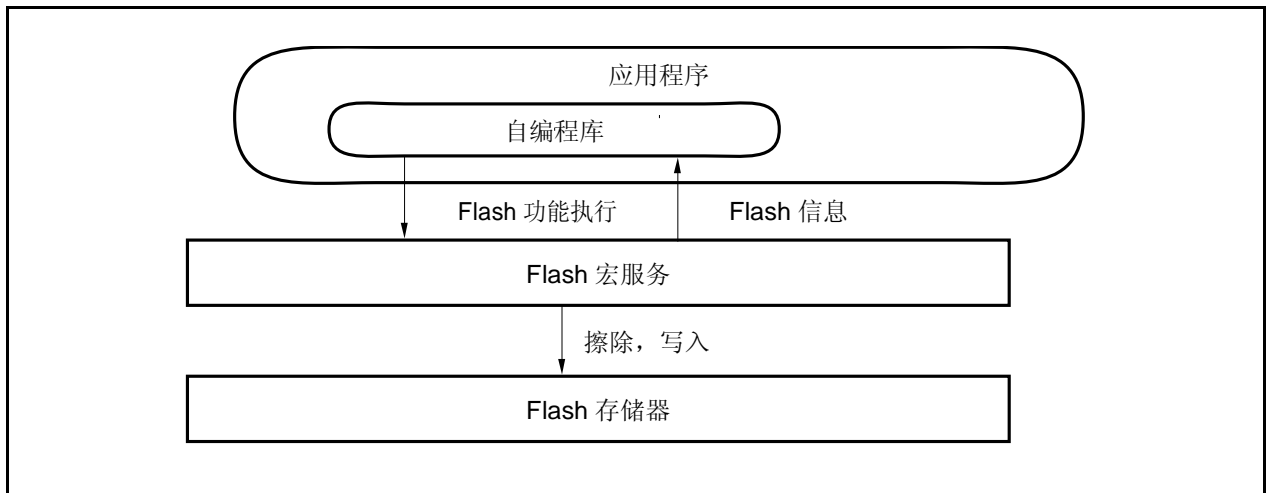
同正常操作模式一样，使用相同的电源 (V_{DD}、V_{SS}、EV_{DD}、EV_{SS}、AV_{SS0}、AV_{SS1}、AV_{DD0}、AV_{DD1}、AV_{REF0}、AV_{REF1}) 供电。

18.5 通过自编程进行重写（仅限 μ PD70F3714）

18.5.1 概述

V850ES/IE2 单片机支持 flash 宏服务，该服务允许用户程序自身重写内部 Flash 存储器。通过使用此接口和一个使用用户应用程序重写 flash 存储器的自编程库，flash 存储器可以通过预先传送到内部 RAM 或外部存储器中的用户应用程序进行重写。因此，在这个区域里用户程序可以更新，并且可以重写固定数据。

图 18-15. 自编程的概念



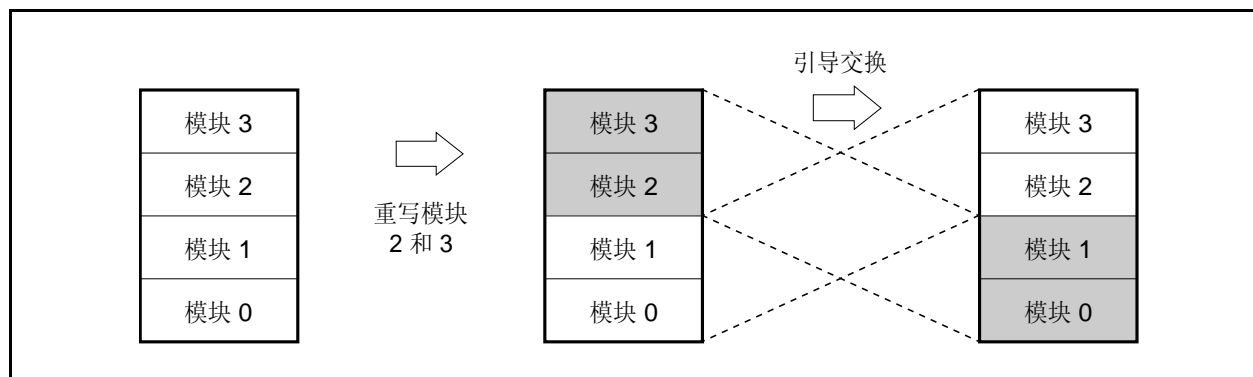
18.5.2 特点

(1) 保证自编程（引导交换功能）（仅限于 μ PD70F3714）

μ PD70F3714 支持引导交换功能，该功能将模块 0 到 1 的物理存储与模块 2 到 3 的物理存储进行交换。

预先在模块 2 到 3 中写入一个要被重写的开始程序，然后交换物理存储器，因为正确的用户程序总是存在于模块 0 到 1 中，因此即使在重写过程中发生掉电，整个区域也可以被安全地重写。

图 18-16. 整个存储区域重写（引导区交换）



(2) 中断支持

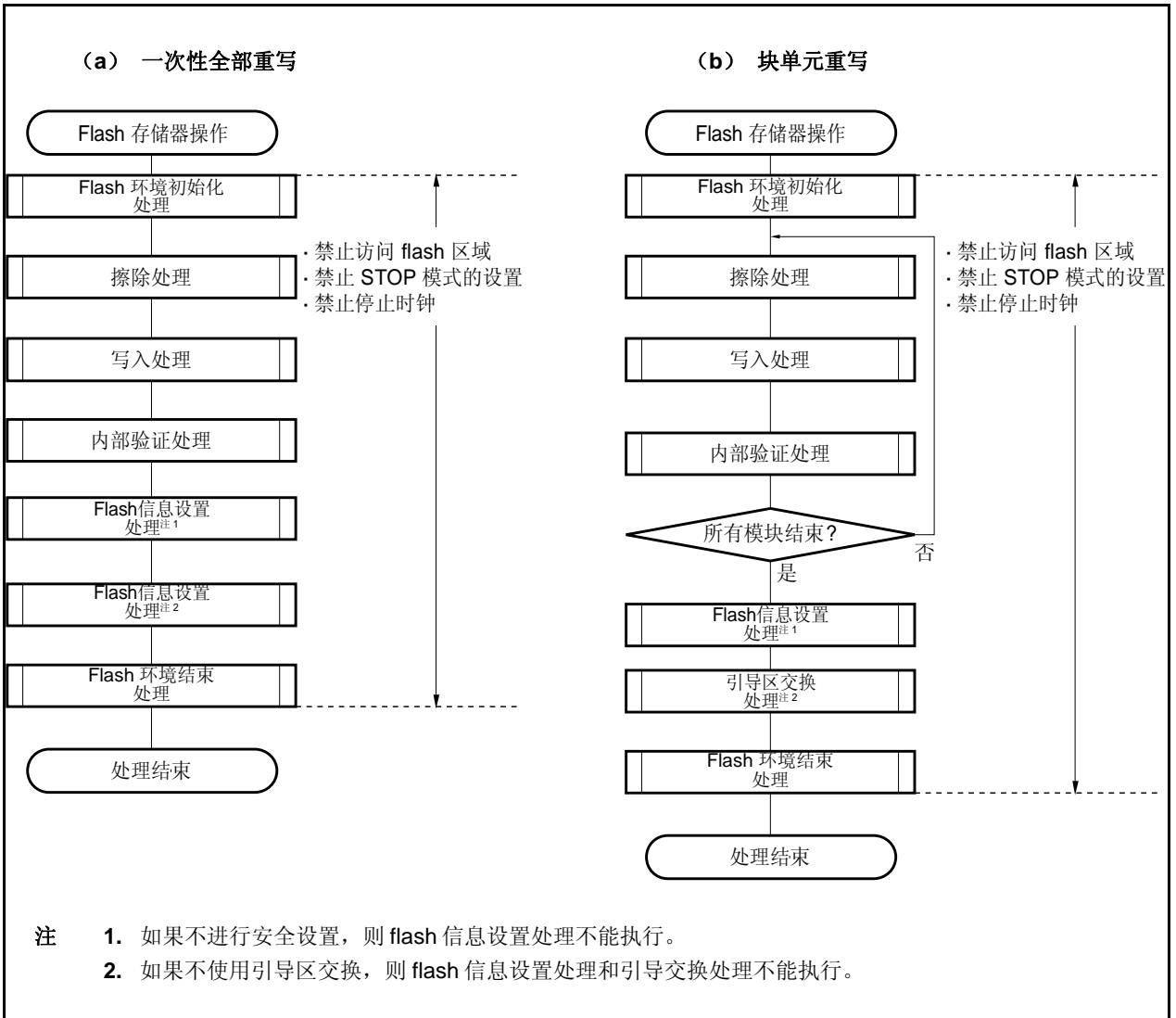
自编程期间不能从 flash 存储器获取指令。因此，通常情况下，即使发生中断也不能使用写入 flash 存储器的用户处理程序。使用 V850ES/IE2 时，用户处理程序可以使用库函数寄存在入口 RAM 区域，因此中断服务可以通过内部 RAM 或外部存储器执行来完成。

18.5.3 标准自编程流程

使用 flash 自编程对 Flash 存储器进行重写的全部处理过程描述如下：

<R>

图 18-17. 标准自编程流程



18.5.4 Flash 存储器功能

<R>

表 18-10. Flash 功能列表

功能名称	概要	支持
FlashEnv	flash 控制宏的初始化	√
FlashBlockErase	仅指定一个模块的擦除	√
FlashWordWrite	由指定地址写入	√
FlashBlockVerify	指定模块的内部校验	√
FlashBlockBlankCheck	指定模块的空白检查	√
FlashFLMDCheck	FLMD 引脚的检查	√
FlashGetInfo	读取 flash 信息	√
FlashSetInfo	设置 flash 信息	√
FlashBootSwap [※]	引导区交换	√
FlashWordRead	从指定地址读取数据	√

注 仅限于 μ PD70F3714。

备注 有关细节，参见 V850 系列 Flash 存储器自编程（单电源供电 Flash 存储器）用户手册。
与 NEC Electronics 的产品代理商联系以取得以上手册。

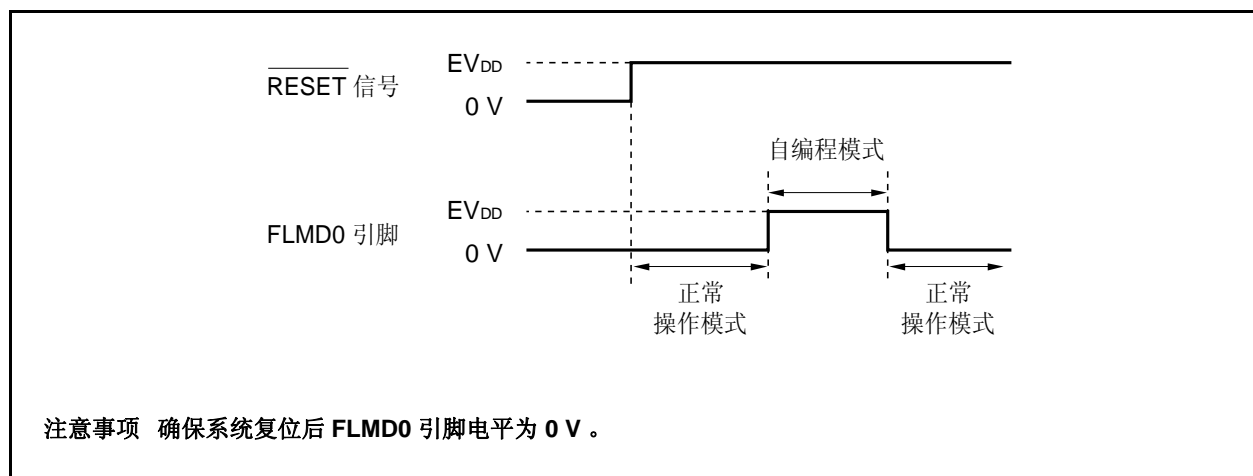
18.5.5 引脚处理

(1) FLMD0 引脚

当系统复位完成后，FLMD0 引脚用来设置操作模式并且防止自编程重写期间 Flash 存储器被重写。因此当复位释放和正常工作模式时要保持 FLMD0 引脚电压为 0 V。重写存储器前，在自编程模式中也需要经由端口控制供给 FLMD0 引脚电压 E_{VDD} 。

当自编程结束后，FLMD0 引脚的电压输入必须返回 0 V。

图 18-18. 模式改变时序



18.5.6 使用的内部资源

下表所列为用于自编程的内部资源。 这些内部资源也可以用于除了自编程外的其他用途。

表 18-11. 使用的内部资源

资源名称	描述说明
入口 RAM 区域 (内部 RAM / 外部 RAM 尺寸 ^註)	用于 flash 模块服务的程序和参数被置于该区域。 入口程序和缺省参数通过调用库初始化函数被拷贝到这里。
堆栈区 (堆栈尺寸 ^註)	用户使用的扩展堆栈被库使用 (在内部 RAM 和外部 RAM 中都可使用)
库代码 (代码尺寸 ^註)	库实体程序 (除处理 Flash 存储器模块以外, 可以在任何其它地方使用)
应用程序	执行用户应用程序。 调用 flash 函数
可屏蔽中断	可以在用户应用执行状态或自编程状态。要在自编程状态中使用该中断, 那么中断服务开始地址必须预先通过注册功能进行注册。

注 有关细节, 参见 V850 系列 Flash 存储器自编程 (单电源供电 Flash 存储器) 用户手册。
与 NEC Electronics 的产品代理商联系以取得以上手册。

第十九章 电气特性

绝对最大额定值 (T_A = 25°C)

参数	符号	条件		额定值	单位
供电电压	V _{DD}	V _{DD} = EV _{DD}		-0.5 至+6.5	V
	V _{SS}	V _{SS} = EV _{SS} = AV _{SS0} = AV _{SS1}		-0.5 至+0.5	V
	EV _{DD}	V _{DD} = EV _{DD}		-0.5 至+6.5	V
	EV _{SS}	V _{SS} = EV _{SS} = AV _{SS0} = AV _{SS1}		-0.5 至+0.5	V
	AV _{DD}	EV _{DD} = AV _{DD0} = AV _{DD1}		-0.5 至+6.5	V
	AV _{SS}	V _{SS} = EV _{SS} = AV _{SS0} = AV _{SS1}		-0.5 至+0.5	V
输入电压	V _{I1}	注 1		-0.5 至 EV _{DD} + 0.5 ^{‡2}	V
	V _{I2}	X1, X2		-0.5 至 VRO ^{‡3} + 0.5	V
输出电流, 低	I _{OL}	P20 至 P25	每个引脚	18	mA
		除 P20 至 P25 之外的引脚	每个引脚	4	mA
		P00 至 P06, P10 至 P14, P16, P17, P30 至 P33, P40 至 P44, PDL0 至 PDL7	所有引脚总和	50	mA
		P20 至 P27	所有引脚总和	50	mA
输出电流, 高	I _{OH}	所有引脚	每个引脚	-4.0	mA
		P00 至 P06, P10 至 P14, P16, P17, P30 至 P33, P40 至 P44, PDL0 至 PDL7	所有引脚总和	-40	mA
		P20 至 P27	所有引脚总和	-15	mA
模拟输入电压	V _{IAN}	ANI00 至 ANI03, ANI10 至 ANI13		-0.5 至 AV _{DD} + 0.5 ^{‡2}	V
模拟参考输入电压	V _{IREF}	AV _{REF0} , AV _{REF1}		-0.5 至 AV _{DD} + 0.5 ^{‡2}	V
工作环境温度	T _A	正常工作时		-40 至+85	°C
		Flash 存储器编程时		-40 至+85	°C
存储温度	T _{stg}			-40 至+125	°C

- 注
1. P00 至 P06, P10 至 P14, P16, P17, P20 至 P27, P30 至 P33, P40 至 P44, PDL0 至 PDL7, RESET, FLMD0
 2. 确保不要超过各电源的最大额定值 (最大值)。
 3. VRO: 稳压器输出电压 (2.5 V (TYP.))。

- 注意事项 1. 不要直接将 IC 产品的输出（或 I/O）引脚相互连接起来，或连接至 V_{DD}，EV_{DD}，和 GND。但是，漏极开路或集电极开路引脚可以直接互连。如果输出引脚能够被设置为高阻状态，并且外部电路的输出时序设计为避免输出冲突形式，那么 IC 产品和外部电路间的输出引脚才可以直接连接。
2. 任何参数，哪怕瞬间超出其最大额定值，都可能使产品性能受损。就是说，工作在最大额定值时产品处于物理损坏的边缘，因此，产品必须在不超出其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正常工作期间性能得到保障的范围。

电容 (T_A = 25°C, V_{DD} = V_{SS} = EV_{DD} = EV_{SS} = AV_{DD0} = AV_{SS0} = AV_{DD1} = AV_{SS1} = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
输入电容	C _i	f _c = 1 MHz 不可测量的引脚返回 0 V。			15	pF
I/O 电容	C _{io}				15	pF
输出电容	C _o				15	pF

操作条件 (T_A = -40 至+85°C, V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1} = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
系统时钟频率	f _{xx}	PLL 模式	20		20	MHz
		时钟直通模式	2.5		2.5	MHz
CPU 时钟频率	f _{CPU}	PLL 模式	2.5		20	MHz
		时钟直通模式	0.3125		2.5	MHz
V _{DD} , EV _{DD} 电压	V _{DD} , EV _{DD}		3.5		5.5	V
AV _{DD0} , AV _{DD1} 电压	AV _{DD}	EV _{DD} 为 4.5 V 或更低时，不能保证操作的有效性。	4.5		EV _{DD}	V

时钟振荡器的特性

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = 3.5$ 至 5.5 V (内部 $V_{DD} = 2.5\text{ V}$))

谐振器	推荐电路	参数	条件	MIN.	TYP.	MAX.	单位
陶瓷谐振器/晶体谐振器		振荡频率 (f_x)		2.5		2.5	MHz
		振荡稳定时间	复位解除后		$2^{16}/f_x$		ms
			STOP 模式解除后		注		ms

注 该值随振荡稳定时间选择寄存器 (OSTS) 设置的不同而异。

- 注意事项
1. 尽可能使振荡器接近 X1 和 X2 引脚。
 2. 不要让信号线穿过虚线封闭的区域。
 3. 振荡波形的占空比必须在 45%至 55%内。
 4. 禁止至 V850ES/IE2 输入外部时钟。

<R> (i) Murata Mfg. Co., Ltd.: 陶瓷谐振器 ($T_A = -40$ 至 $+85^\circ\text{C}$)

类型	部件编号	振荡频率 f_x (MHz)	推荐的电路常量			振荡电压范围	
			C1	C2	Rd (k Ω)	MIN. (V)	MAX. (V)
外部安装	CSTCC2M50G56-R0	2.5	片上(47 pF)	片上(47 pF)	1	V_{POCO}	5.5

注意事项 这些振荡器常量是基于谐振器厂商在特殊环境下进行鉴定的参考值。实际应用中，有必要优化振荡器特性时，请从谐振器厂商申请关于安装电路的鉴定。
振荡电压和振荡频率仅指示振荡器特性，因此，对于内部操作条件，请根据 DC 特性和 AC 特性来使用 V850E/IE2。

DC 特性

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ 至 5.5 V, $AV_{DD0} = AV_{DD1} = 4.5$ 至 5.5 V, $V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1} = 0$ V)
(1/2)

参数	符号	条件		MIN.	TYP.	MAX.	单位
输入电压, 高	V_{IH1}	注 1		$0.7EV_{DD}$		EV_{DD}	V
	V_{IH2}	注 2		$0.8EV_{DD}$		EV_{DD}	V
输入电压, 低	V_{IL1}	注 1		EV_{SS}		$0.3EV_{DD}$	V
	V_{IL2}	注 2		EV_{SS}		$0.2EV_{DD}$	V
输入漏电流, 高	I_{LIH1}	$V_i = V_{DD} = EV_{DD}$	除 X1 外			5	μA
	I_{LIH2}		X1			20	μA
输入漏电流, 低	I_{LIL1}	$V_i = 0$ V	除 X1 外			-5	μA
	I_{LIL2}		X1			-20	μA
输出漏电流, 高	I_{LOH}	$V_o = V_{DD} = EV_{DD}$				5	μA
输出漏电流, 低	I_{LOL}	$V_o = 0$ V				-5	μA
输出电压, 高	V_{OH}	注 3	每个引脚 $I_{OH} = -1.0$ mA	所有引脚 -31 mA	$EV_{DD} - 1.0$		V
		注 4	每个引脚 $I_{OH} = -1.0$ mA	所有引脚 -8 mA	$EV_{DD} - 1.0$		V
输出电压, 低	V_{OL1}	注 5, 注 6	每个引脚 $I_{OL} = 15$ mA	所有引脚 45 mA		2.0	V
		注 3	每个引脚 $I_{OL} = 1.0$ mA	所有引脚 31 mA		0.4	V
	注 4	每个引脚 $I_{OL} = 1.0$ mA	所有引脚 8 mA		0.4	V	
上拉电阻	R_1			10	30	100	k Ω

- 注
1. P20 至 P27, P31, P33, P41, 和 PDL0 至 PDL7 引脚。
 2. P00 至 P06, P10 至 P14, P16, P17, P30, P32, P40, P42 至 P44, $\overline{\text{RESET}}$, 和 FLMD0 引脚。
 3. P00 至 P06, P10 至 P14, P16, P17, P30 至 P33, P40 至 P44, and PDL0 至 PDL7 引脚。
 4. P20 至 P27 引脚
 5. P20 至 P25 引脚
 6. 最多有三个引脚可以同时输出低电平（其它三个必须输出高电平或高阻抗）。

- 备注
1. 除非另外说明, 复用功能引脚的特性与端口引脚的相同。
 2. 当 I_{OH} 和 I_{OL} 条件没有满足单个引脚但满足所有引脚的总值时, 则仅该引脚不满足 DC 特性。

DC 特性

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ 至 5.5 V, $AV_{DD0} = AV_{DD1} = 4.5$ 至 5.5 V, $V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1} = 0$ V)
(2/2)

	参数	符号	条件	MIN.	TYP. ^{注2}	MAX.	单位
<R>	电源电流 ^{注1}	I _{DD1}	正常操作, $f_{xx} = 20$ MHz		38	60	mA
<R>		I _{DD2}	HALT 模式, $f_{xx} = 20$ MHz		22	35	mA
<R>		I _{DD3}	IDLE 模式, $f_{xx} = 20$ MHz		6	10	mA
		I _{DD4}	STOP 模式		30	200	μA

- 注
1. 不包括流过输出缓冲区的电流以及上拉电阻。
 2. TYP.值是 $V_{DD} = 5.0$ V 时的参考值。

数据保持特性

STOP 模式($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1} = 0\text{ V}$)

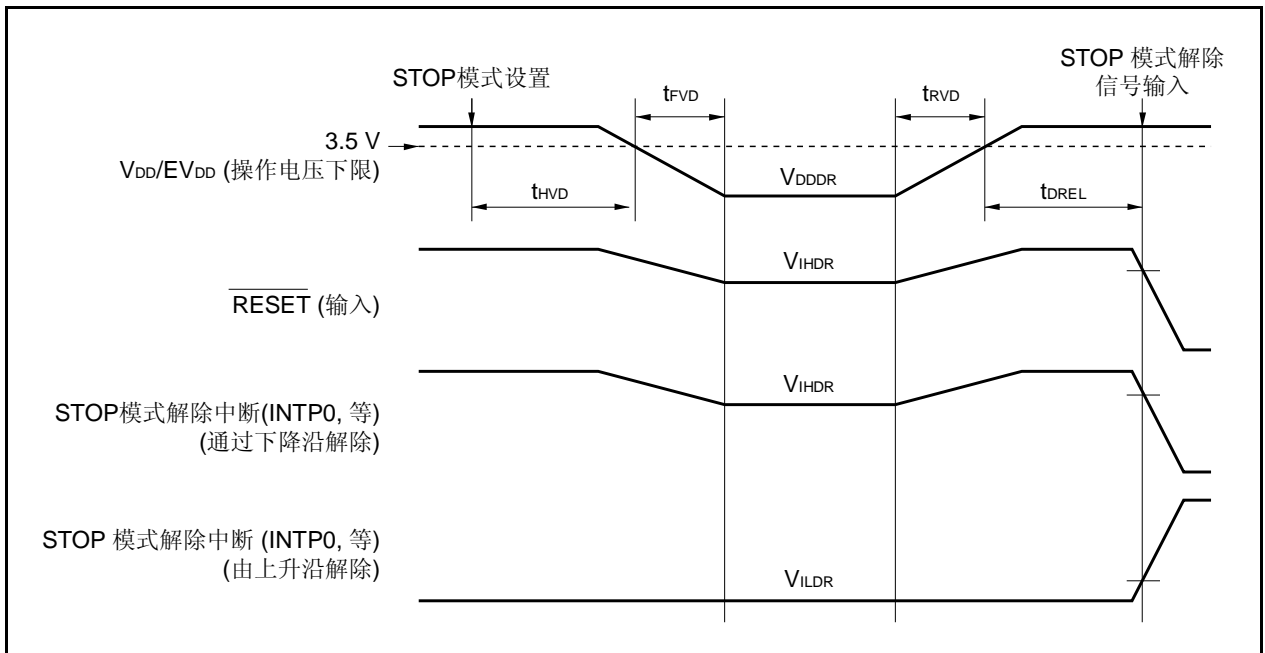
参数	符号	条件	MIN.	TYP.	MAX.	单位
数据保持电压	V_{DDDR}	在 STOP 模式中	V_{DD}, EV_{DD}	注	5.5	V
		复位后不同于供电电压	V_{DD}, EV_{DD}	1.9	注	V
数据保持电流	I_{DDDR}	$V_{DD} = V_{DDDR}$		30	200	μA
电压上升时间	t_{RVD}	V_{DD}	1			μs
电压下降时间	t_{FVD}	V_{DD}	1			μs
电压保持时间 (对于 STOP 模式的设置)	t_{HVD}		0			ms
STOP 模式解除信号输入时间	t_{DREL}		0			ms
数据保持输入电压, 高	V_{IHDR}	所有输入引脚	$0.9V_{DDDR}$		V_{DDDR}	V
数据保持输入电压, 低	V_{ILDR}	所有输入引脚	0		$0.1V_{DDDR}$	V

注 没有使用低电压检测器 (LVI) 的复位模式 (LVIM.LVIMD 位 = 0) 时:
 POC 检测电压 (V_{POC0})
 使用低电压检测器 (LVI) 的复位模式 (LVIM.LVIMD 位 = 1) 时:
 LVI 检测电压 (V_{LV10}/V_{LV11})

注意事项 转换到 STOP 模式以及从 STOP 模式中恢复必须在额定的工作范围内进行。

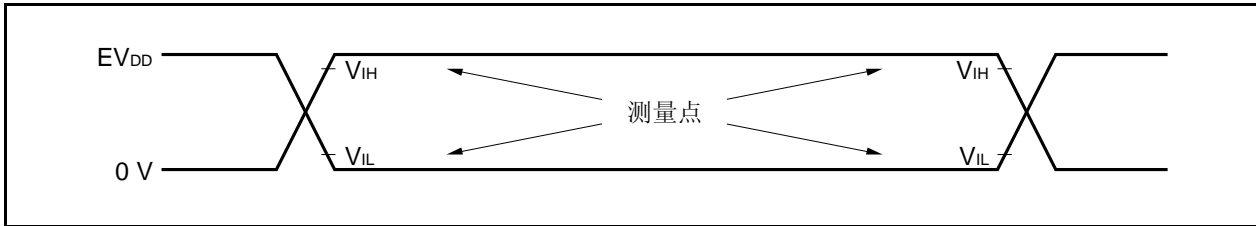
- 当没有使用 LVI (复位模式) 时: POC 检测电压 (V_{POC0}) 或更高
- 当使用 LVI (复位模式) 时: LVI 检测电压 (V_{LV10}/V_{LV11}) 或更高

如果在没有使用 LVI (复位模式) 时电压达到 POC 检测电压 (V_{POC0}) 或者在使用 LVI (复位模式) 时达到检测电压 (V_{LV10}/V_{LV11}), 因为内部复位 (POCRES/LVIREs) 没有受影响, 所以 STOP 模式将会自动解除。

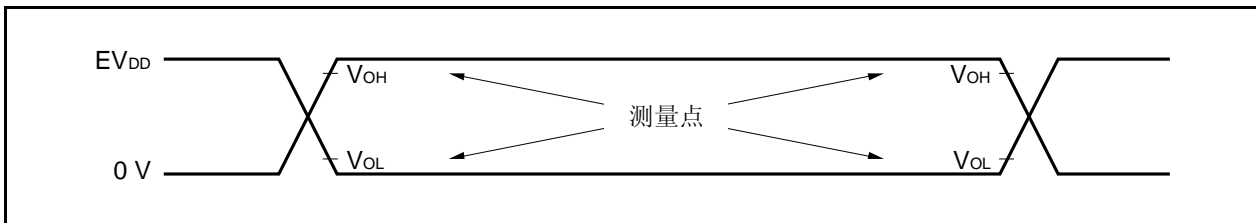


AC 特性

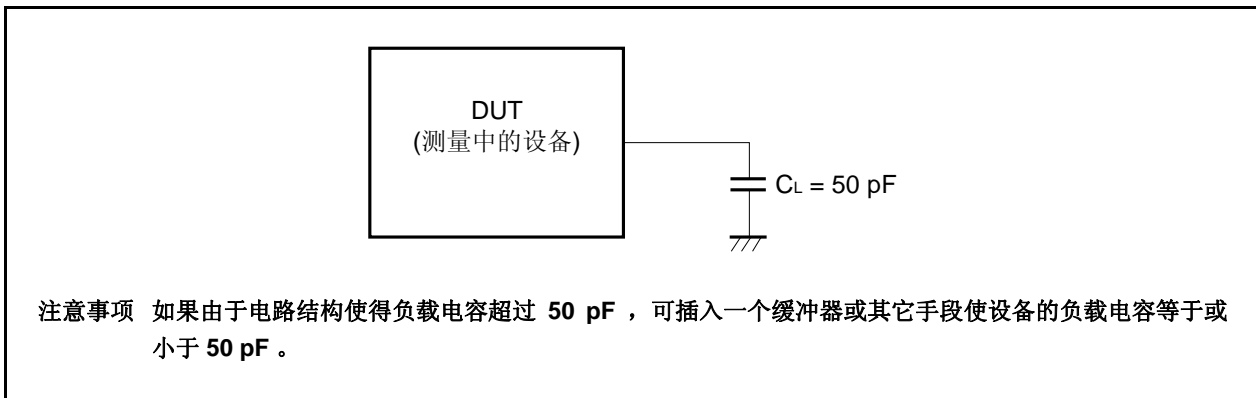
AC 测试输入测量点



AC 测试输出测量点



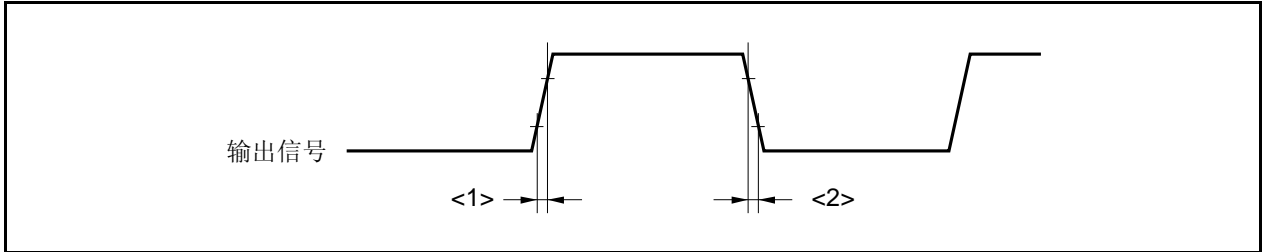
负载条件



输出信号时序

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ 至 5.5 V , $V_{SS} = EV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号		条件	MIN.	MAX.	单位
输出上升时间	t _{OR}	<1>			15	ns
输出下降时间	t _{OF}	<2>			15	ns

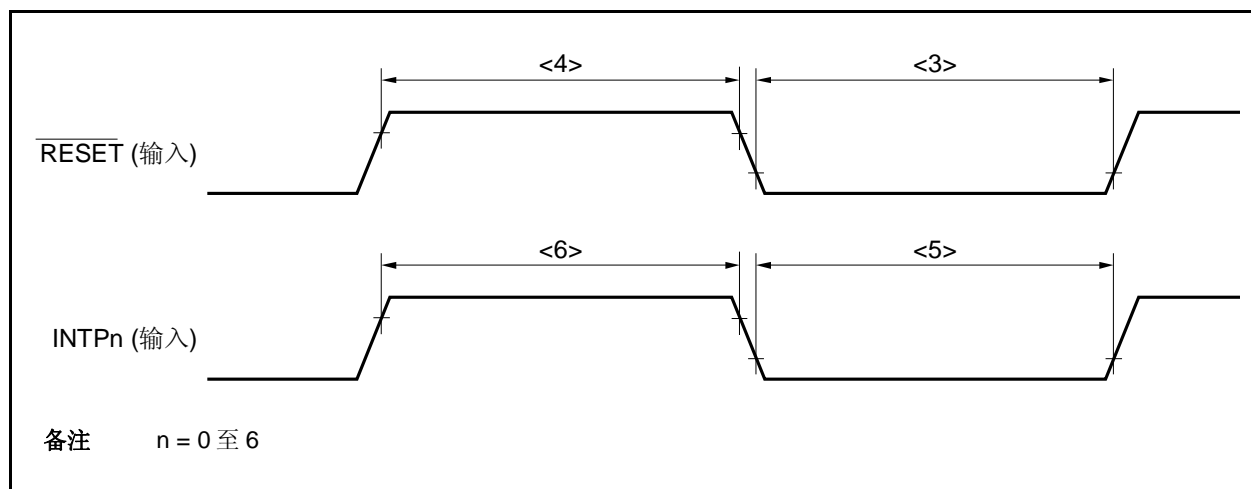


复位，外部中断时序

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ 至 5.5 V, $V_{SS} = EV_{SS} = 0$ V, $C_L = 50$ pF)

参数	符号	条件	MIN.	MAX.	单位
RESET 低电平宽度	t_{WRSL}	<3>	500		ns
RESET 高电平宽度	t_{WRSH}	<4>	500		ns
INTPn 低电平宽度	t_{WITL}	n = 0 至 5 (模拟噪声消除)	500		ns
		n = 6 (数字噪声消除)	$5T_{smp} + 10$		ns
INTPn 高电平宽度	t_{WITH}	n = 0 至 5 (模拟噪声消除)	500		ns
		n = 6 (数字噪声消除)	$5T_{smp} + 10$		ns

备注 T_{smp} : 噪声消除采样时钟周期 (由 INTPNRC 寄存器设置)



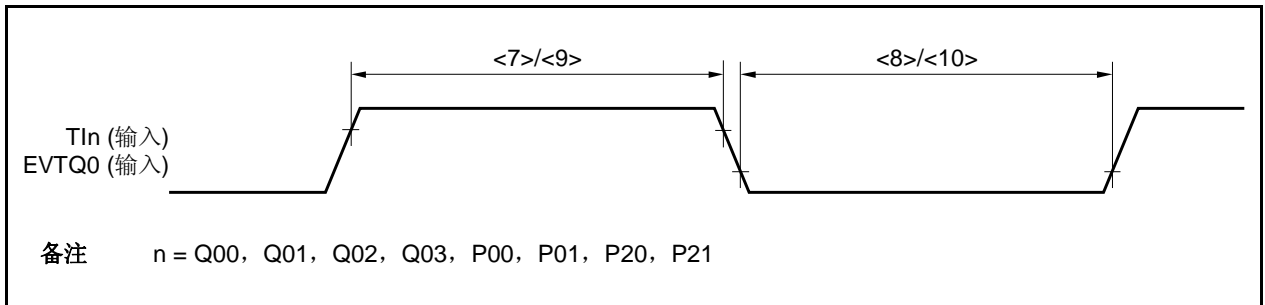
定时器时序

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ 至 5.5 V , $V_{SS} = EV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
TIn 高电平宽度 ^{注1}	t_{WTHn}	<7>	$10T + 10$		ns
TIn 低电平宽度 ^{注1}	t_{WTILn}	<8>	$10T + 10$		ns
EVTQ0 高电平宽度 ^{注1}	t_{WEVHn}	<9>	$10T + 10$		ns
EVTQ0 低电平宽度 ^{注1}	t_{WEVLn}	<10>	$10T + 10$		ns

注 $T = 1/f_{xx}$

- 备注
1. $n = Q00, Q01, Q02, Q03, P00, P01, P20, P21$
 2. 以上规范显示了被正确检测为有效沿的脉冲宽度。因此，即使输入一个比上述规范窄的脉冲，也会将其检测为有效沿。



高阻抗控制时序

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ 至 5.5 V , $AV_{DD0} = AV_{DD1} = 4.5$ 至 5.5 V , $V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	MAX.	单位
振荡停止 → 定时器输出高阻抗	t_{CLM}	时钟监视器工作时		65	μs
TOQH0OFF 输入 → 定时器输出高阻抗	t_{HTQH0}			300	ns
TOQ1OFF 输入 → 定时器输出高阻抗	t_{HTQ1}			300	ns
TOPnOFF 输入 → 定时器输出高阻抗	t_{HTPn}			300	ns

备注 $n = 2, 3$

CSIB Timing

(1) 主模式

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ 至 5.5 V, $V_{SS} = EV_{SS} = 0$ V, $C_L = 50$ pF)

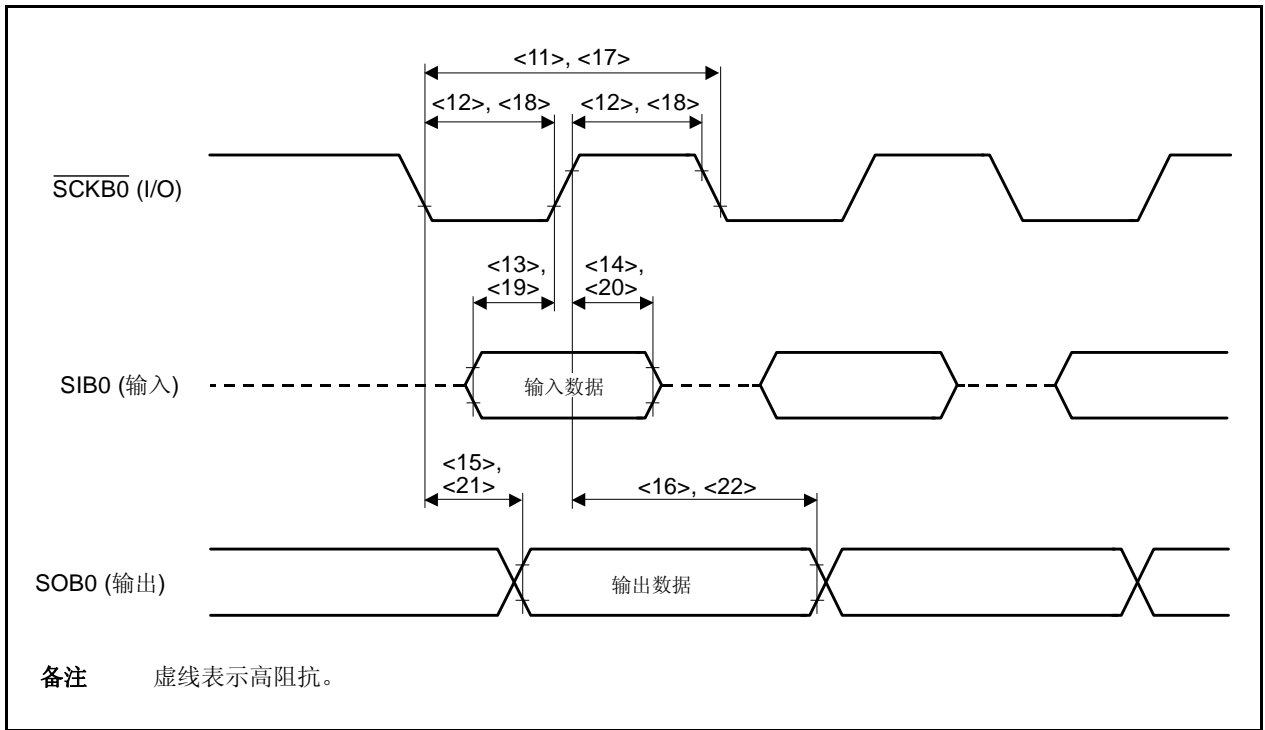
参数	符号	条件	MIN.	MAX.	单位
$\overline{\text{SCKB0}}$ 周期	t_{KCYM}	<11>	125		ns
$\overline{\text{SCKB0}}$ 高/低电平宽度	t_{KWHM} , t_{KWLM}	<12>	$t_{\text{KCYM}}/2 - 10$		ns
SIB0 设置时间 (至 $\overline{\text{SCKB0}}\uparrow$)	t_{SSIM}	<13>	30		ns
SIB0 保持时间 (自 $\overline{\text{SCKB0}}\uparrow$)	t_{HSIM}	<14>	30		ns
SOB0 输出延迟时间 (自 $\overline{\text{SCKB0}}\downarrow$)	t_{DSOM}	<15>		30	ns
SOB0 输出延迟时间 (自 $\overline{\text{SCKB0}}\uparrow$)				30	ns
SOB0 输出保持时间 (自 $\overline{\text{SCKB0}}\uparrow$)	t_{HSOM}	<16>	$t_{\text{KCYM}}/2 - 10$		ns
SOB0 输出保持时间 (自 $\overline{\text{SCKB0}}\downarrow$)			$t_{\text{KCYM}}/2 - 10$		ns

(2) 从模式

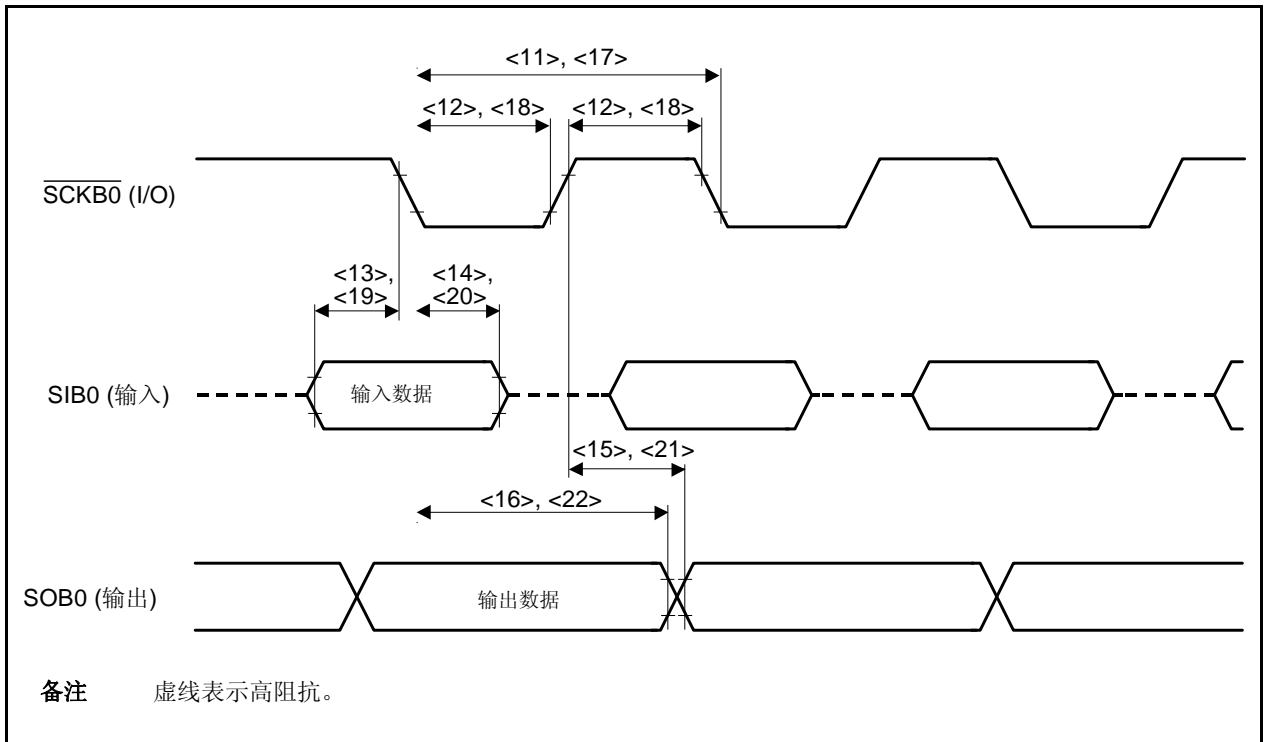
($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ 至 5.5 V, $V_{SS} = EV_{SS} = 0$ V, $C_L = 50$ pF)

参数	符号	条件	MIN.	MAX.	单位
$\overline{\text{SCKB0}}$ 周期	t_{KCYS}	<17>	125		ns
$\overline{\text{SCKB0}}$ 高/低电平宽度	t_{KWHS} , t_{KWLS}	<18>	$t_{\text{KCYS}}/2 - 10$		ns
SIB0 设置时间 (至 $\overline{\text{SCKB0}}\uparrow$)	t_{SSIS}	<19>	30		ns
SIB0 保持时间 (自 $\overline{\text{SCKB0}}\uparrow$)	t_{HSIS}	<20>	30		ns
SOB0 输出延迟时间 (自 $\overline{\text{SCKB0}}\downarrow$)	t_{DSOS}	<21>		30	ns
SOB0 输出延迟时间 (自 $\overline{\text{SCKB0}}\uparrow$)				30	ns
SOB0 输出保持时间 (自 $\overline{\text{SCKB0}}\uparrow$)	t_{HSOS}	<22>	$t_{\text{KCYS}}/2 - 10$		ns
SOB0 输出保持时间 (自 $\overline{\text{SCKB0}}\downarrow$)			$t_{\text{KCYS}}/2 - 10$		ns

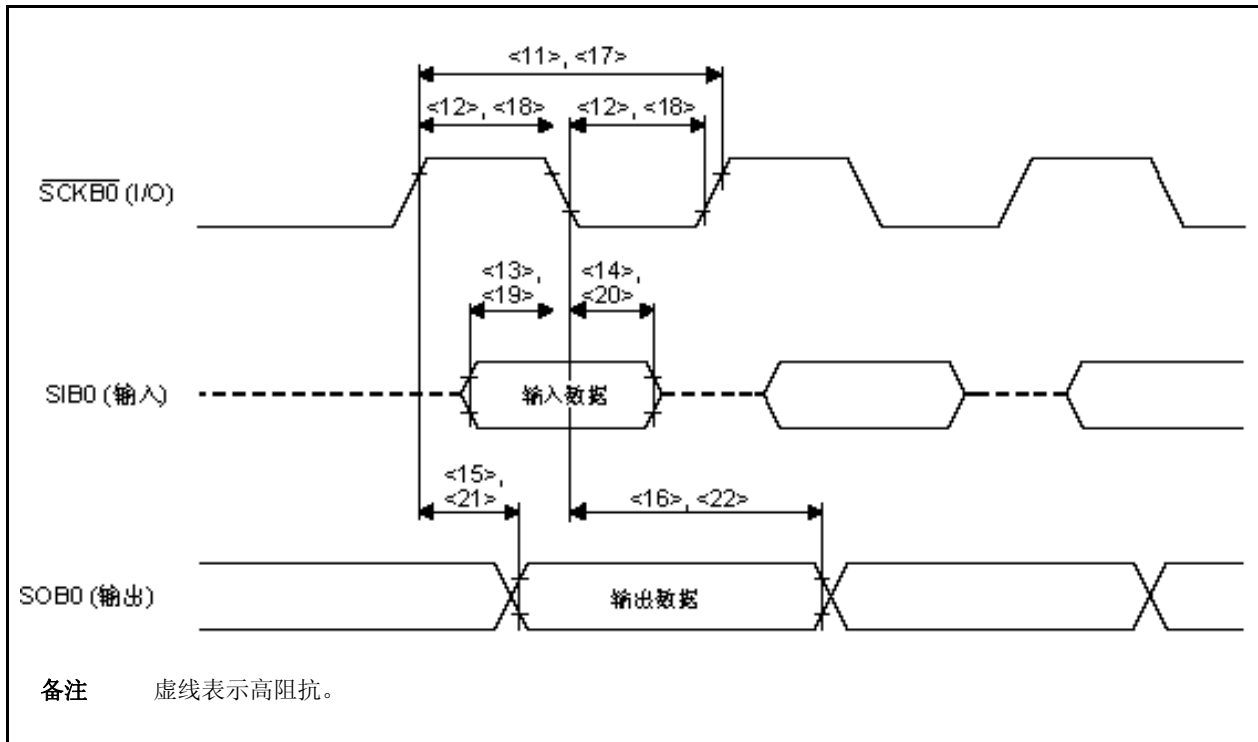
CSIB 时序: CB0CTL1.CB0CKP、CB0CTL1.CB0DAP 位 = 00



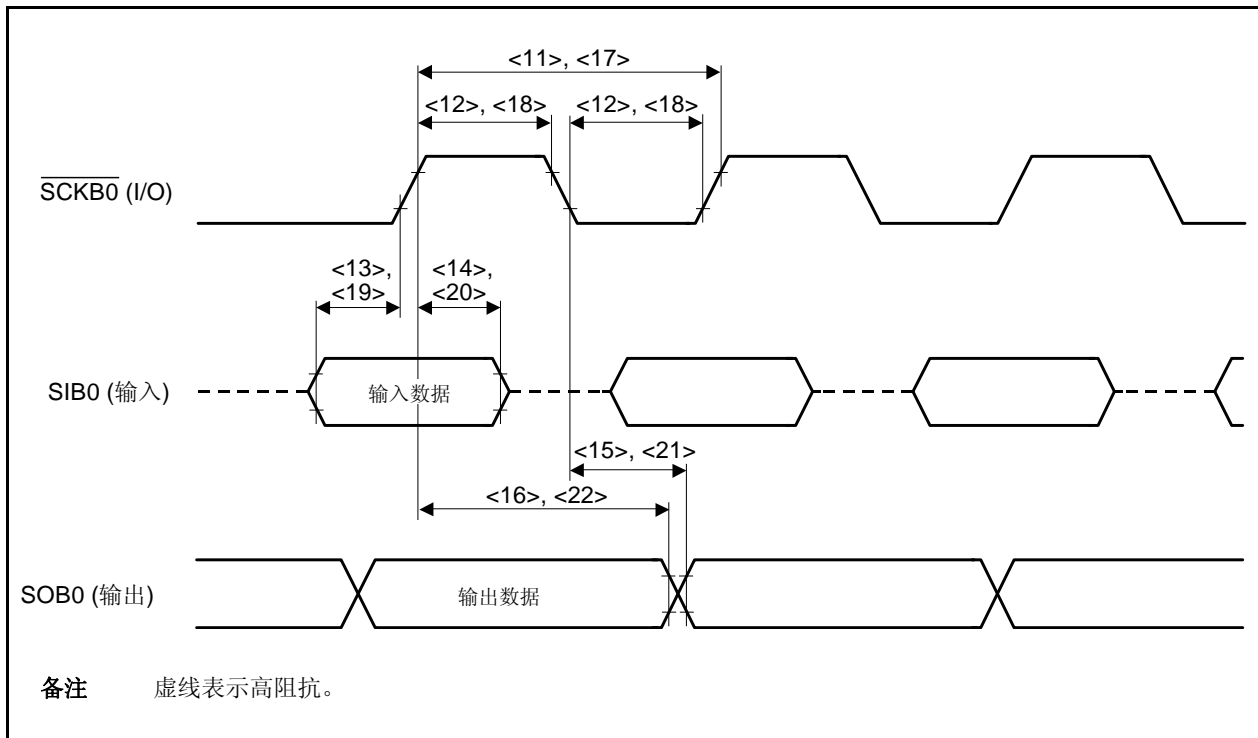
CSIB 时序: CB0CTL1.CB0CKP、CB0CTL1.CB0DAP 位 = 01



C



CSIB 时序: CB0CTL1.CB0CKP、CB0CTL1.CB0DAP 位 = 11



A/D Converters 0、1 的特性

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 4.5$ 至 5.5 V, $AV_{DD0} = AV_{DD1} = 4.5$ 至 5.5 V, $V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1} = 0$ V, $C_L = 50$ pF)

参数	符号	条件	MIN.	TYP.	MAX.	单位
分辨率			10	10	10	bit
总误差 ^{注1}					±4.0	LSB
转换时间	t _{CONV}		2		10	μs
零刻度误差 ^{注1}					±4.0	LSB
满量程误差 ^{注1}					±4.0	LSB
积分线性误差 ^{注1}					±4.0	LSB
微分线性误差 ^{注1}					-1 至+2	LSB
模拟参考电压	AV _{REF}	AV _{REF0} = AV _{REF1} = AV _{DD0} = AV _{DD1}	4.5		5.5	V
模拟输入电压	V _{IAN}		AV _{SS}		AV _{DD}	V
AV _{DD0} , AV _{DD1} 供电电流 ^{注2}	AI _{DD}	操作期间		6	10	mA
	AI _{DDS}	在 STOP 模式中 ^{注3}		0.5	25	μA

- 注**
1. 量化误差除外 (±0.5 LSB)
 2. 该值为 A/D 转换器 0 或 A/D 转换器 1 的一个周期。
 3. 在设为 STOP 模式前停止 A/D 转换器 0 和 1 的操作 (ADAnM0.ADAnCE 位 = 0)。

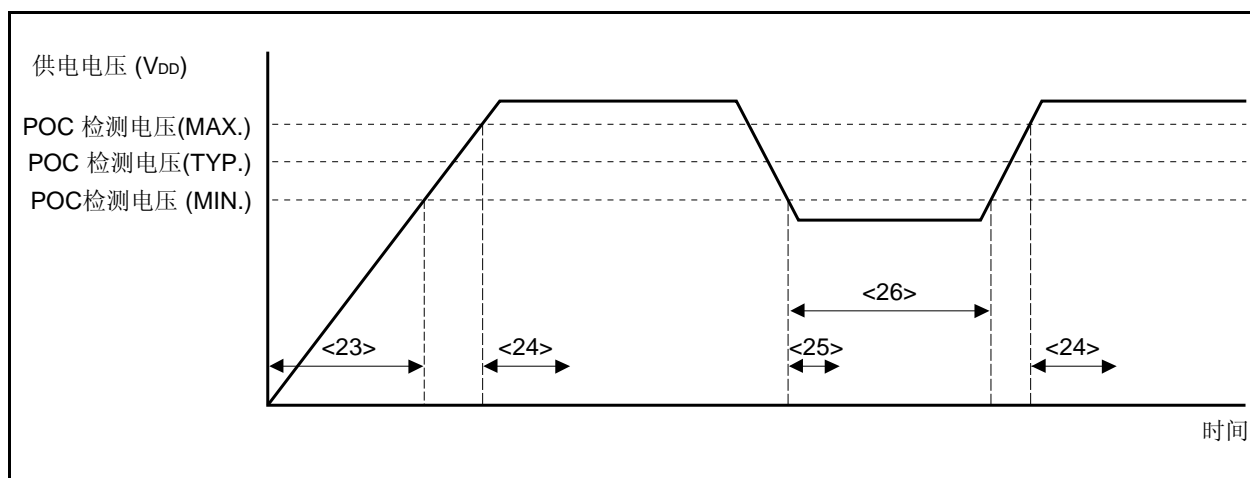
- 备注**
1. LSB: 最低有效位
 2. n = 0, 1

加电清除电路 (POC)

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ 至 5.5 V , $V_{SS} = EV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
POC 检测电压	V_{POC0}		3.5	3.7	3.9	V
电压上升时间	t_{PTH}	<23>	$V_{DD} = 0$ 至 3.5 V	0.002		ms
响应时间 1 ^{注1}	t_{PTHd}	<24>	上电时, V_{DD} 达到 POC 检测电压后		3.0	ms
响应时间 2 ^{注2}	t_{PD}	<25>	断电时, V_{DD} 低于 POC 检测电压后		1.0	ms
最小 V_{DD} 宽度	t_{PW}	<26>		0.2		ms

- 注
1. 检测到 POC 检测电压后, 解除一个复位信号 (POCRES) 所需的时间。
 2. 检测到 POC 检测电压后, 输出一个复位信号 (POCRES) 所需的时间。



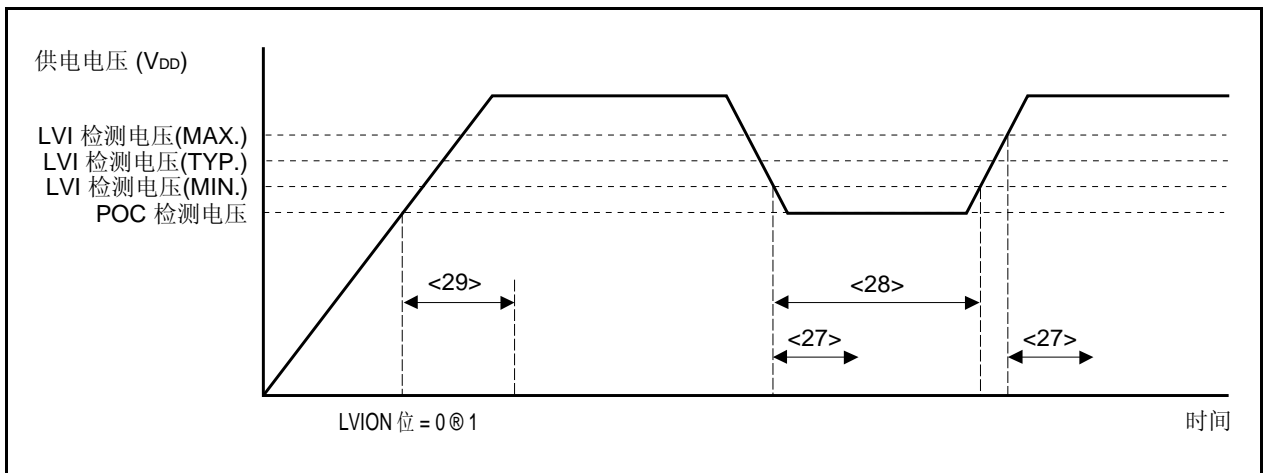
低电压监测器(LVI)

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ 至 5.5 V , $V_{SS} = EV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
LVI 检测电压	V_{LV10}	LVIS.LVIS0 位= 0	4.2	4.4	4.6	V
	V_{LV11}	LVIS.LVIS0 位= 1	4.0	4.2	4.4	V
响应时间 1#	t_{LD}	<27> V_{DD} 达到 V_{LV10}/V_{LV11} (MAX.)或下降至 V_{LV10}/V_{LV11} (MIN.)后		0.2	2.0	ms
最小 V_{DD} 宽度	t_{LW}	<28>	0.2			ms
参考电压稳定等待时间	t_{LWAIT}	<29> V_{DD} 达到 POC 检测电压, 且 LVIM.LVION 自 0 改变至 1 后		0.1	0.2	ms

<R>

注 检测到检测电压后, 输出一个中断请求信号 (INTLVI) 或中断复位信号 (LVIRES) 所需的时间。



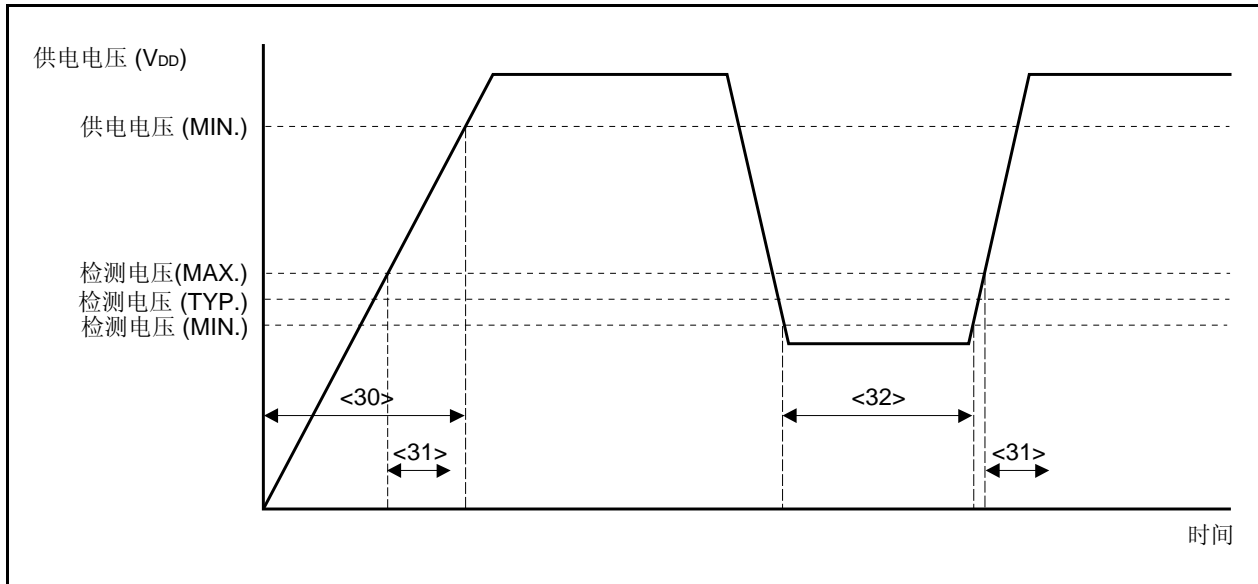
<R>

RAM 保持电压检测

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ 至 5.5 V , $V_{SS} = EV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
检测电压	V_{RAMH}		1.9	2.0	2.1	V
电压上升时间	t_{RAMHTH}	<30> $V_{DD} = 0$ 至 3.5 V	0.002		1800	ms
响应时间 1 ^注	t_{RAMHD}	<31> V_{DD} 达到 2.1 V		0.2	2.0	ms
最小 V_{DD} 宽度	t_{RAMHW}	<32>	0.2			ms

注 检测到检测电压后，将 RAMS.RAMF 位设为 1 所需的时间。



Flash 存储器编程特性

($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ 至 5.5 V , $V_{SS} = EV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

(1) 基本特性

参数	符号	条件	MIN.	TYP.	MAX.	单位
工作频率	f _{CPU}		2.5		20	MHz
供电电压	V _{DD}		3.5		5.5	V
重写时间	C _{ERWR}	注			100	次数
重写温度	t _{PRG}		-40		85	°C

注 当对出厂产品进行初次写入时，将“擦除写入”和“只写”两个过程按一次重写来计数。

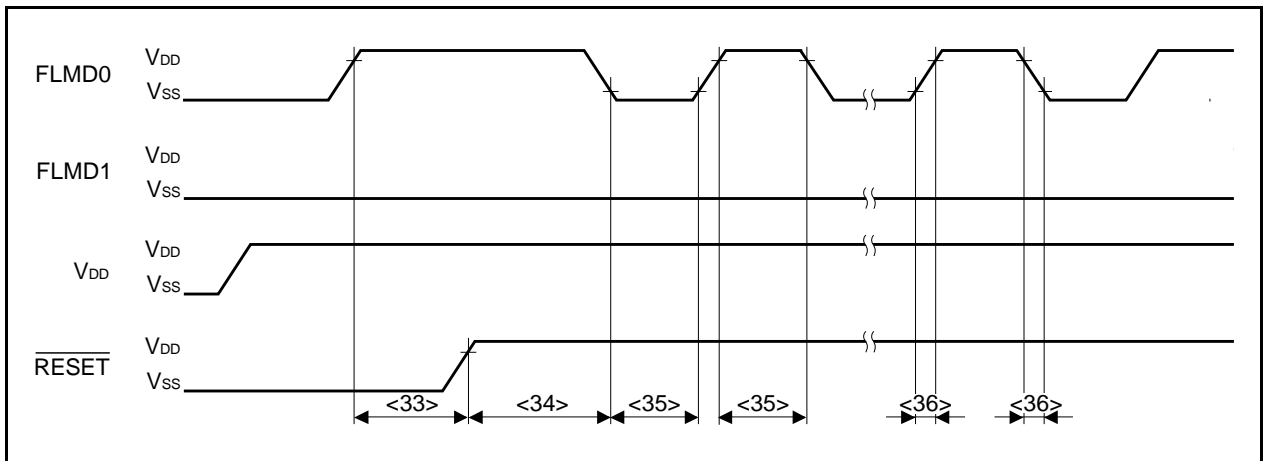
例如 (P: 写入, E: 擦除)

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

(2) 串行写入操作特性

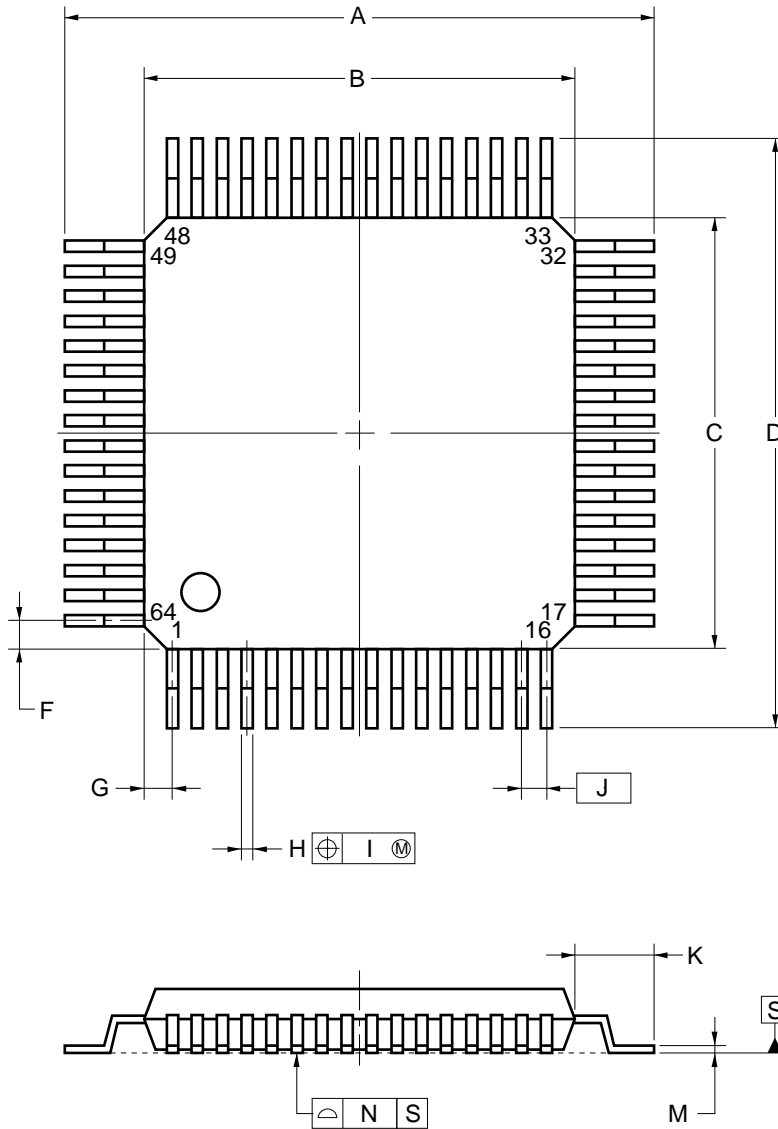
参数	符号	条件	MIN.	TYP.	MAX.	单位
FLMD0, FLMD1 设置时间	t _{MDSET} <33>		2		3000	s
自 RESET↑的 FLMD0 脉冲输入起始时间	t _{RFCF} <34>		30560/f _x + 振荡稳定时间			ns
FLMD0 脉冲宽度	t _{CH/tCL} <35>		10	100		μs
FLMD0 脉冲上升/下降时间	t _{R/tF} <36>				50	ns



<R> (3) 编程特性

参数	符号	条件	TYP.	TYP.-Worst	MAX.	单位
块擦除		f _{xx} = 20 MHz, 8 KB	0.78		5.78	s
		f _{xx} = 20 MHz, 56 KB	3.9		35.0	s
写入 (128 个字节)		f _{xx} = 20 MHz	5.3		43.1	ms
块内部验证		f _{xx} = 20 MHz, 8 KB	80.3		3000	ms
		f _{xx} = 20 MHz, 56 KB	562		3000	ms
块空白检查		f _{xx} = 20 MHz, 8 KB	35.5		44.4	ms
		f _{xx} = 20 MHz, 56 KB	249		311	ms
Flash 信息设置		f _{xx} = 20 MHz	1.32		10.8	ms

64-引脚 LQFP 塑料封装(14x14)



注
每个引脚实际位置(T.P.)偏离理论中心最多 0.20 mm。

项目	mm
A	17.2±0.2
B	14.0±0.2
C	14.0±0.2
D	17.2±0.2
F	1.0
G	1.0
H	0.37 ^{+0.08} _{-0.07}
I	0.20
J	0.8 (T.P.)
K	1.6±0.2
L	0.8
M	0.17 ^{+0.03} _{-0.06}
N	0.10
P	1.4±0.1
Q	0.127±0.075
R	3° ^{+4°} _{-3°}
S	1.7 MAX.
T	0.25
U	0.886±0.15

P64GC-80-8BS

这些器件应该按照以下所建议的条件进行焊接和安装。

关于技术方面的信息，参考以下网站。

半导体器件安装手册(<http://www.necel.com/pkg/en/mount/index.html>)

表 21-1. 表面贴装类型焊接条件

μPD70F3713GC-8BS-A: 64-引脚 LQFP 塑料封装 (14 × 14)

μPD70F3714GC-8BS-A: 64-引脚 LQFP 塑料封装(14 × 14)

焊接方法	焊接条件	建议条件符号
回流焊	封装耐温极限: 260°C, 时间: 60 最大 60 秒 (220°C 或更高), 次数: 最多 3 次; 暴露极限: 3 天 ^注 (之后, 需要在 125°C 下经过 20 至 72 小时的预烘培)。	IR60-203-3
波峰焊	相关细节, 请和 NEC 电子的销售代理商联系。	—
局部加热	引脚温度: 最大 350°C, 时间: 最多 3 秒 (每排引脚)。	—

注 打开干燥包装后, 在许可的储存期内, 保存在温度低于 25°C 而湿度低于 65% RH 环境中。

- 备注**
1. 器件编号末尾有-A 的产品为无铅产品。
 2. 如果使用以上建议之外的焊接方法和条件, 请和 NEC 电子的销售代理商联系。

附录 A 注意事项

A.1 sld指令和中断请求之间冲突时的限制

A.1.1 描述

如果 <2>中的指令紧靠 sld 指令之前，并处于<1>中的一条指令之后，其解码操作与<1>中的指令完毕前发生的中断请求相冲突，那么，<1>中的指令执行结果可能不会存入寄存器中。

指令 <1>

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

指令 <2>

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

<举例>

<i> ld.w [r11], r10	⋮	如果 mov 指令<ii>的解码操作刚好在 sld 指令<iii>前，并且，在 ld 指令<i>完毕之前与中断请求相冲突，则指令 <i>的执行结果可能不会存入寄存器中。
<ii> mov r10, r28		
<iii> sld.w 0x28, r10		

A.1.2 应对方法

(1) 当使用(CA850)编译器时

请使用 CA850 Ver. 2.61 或更新版本，因为它可以自动避免产生相应的指令序列。

(2) 当使用汇编器时

当紧随指令<ii>之后执行 sld 指令时，使用以下两种办法之一以避免以上操作。

- 紧靠 sld 指令之前插入一条 nop 指令。
- 在以上紧靠 sld 指令执行的指令<ii>中，不要使用相同的寄存器作为 sld 指令的目的寄存器。

附录 B 寄存器索引

(1/5)

符号	名称	单元	页码
AD0IC	中断控制寄存器	INTC	559
AD1IC	中断控制寄存器	INTC	559
ADA0CR0	A/D0 转换结果寄存器 0	ADC0	436
ADA0CR0H	A/D0 转换结果寄存器 0H	ADC0	436
ADA0CR1	A/D0 转换结果寄存器 1	ADC0	436
ADA0CR1H	A/D0 转换结果寄存器 1H	ADC0	436
ADA0CR2	A/D0 转换结果寄存器 2	ADC0	436
ADA0CR2H	A/D0 转换结果寄存器 2H	ADC0	436
ADA0CR3	A/D0 转换结果寄存器 3	ADC0	436
ADA0CR3H	A/D0 转换结果寄存器 3H	ADC0	436
ADA0M0	A/D 转换器 0 模式寄存器 0	ADC0	431
ADA0M1	A/D 转换器 0 模式寄存器 1	ADC0	433
ADA0M2	A/D 转换器 0 模式寄存器 2	ADC0	435
ADA0S	A/D 转换器 0 通道指定寄存器	ADC0	434
ADA1CR0	A/D1 转换结果寄存器 0	ADC1	436
ADA1CR0H	A/D1 转换结果寄存器 0H	ADC1	436
ADA1CR1	A/D1 转换结果寄存器 1	ADC1	436
ADA1CR1H	A/D1 转换结果寄存器 1H	ADC1	436
ADA1CR2	A/D1 转换结果寄存器 2	ADC1	436
ADA1CR2H	A/D1 转换结果寄存器 2H	ADC1	436
ADA1CR3	A/D1 转换结果寄存器 3	ADC1	436
ADA1CR3H	A/D1 转换结果寄存器 3H	ADC1	436
ADA1M0	A/D 转换器 1 模式寄存器 0	ADC1	431
ADA1M1	A/D 转换器 1 模式寄存器 1	ADC1	433
ADA1M2	A/D 转换器 1 模式寄存器 2	ADC1	435
ADA1S	A/D 转换器 1 通道指定寄存器	ADC1	434
CB0CTL0	CSIB0 控制寄存器 0	CSIB0	503
CB0CTL1	CSIB0 控制寄存器 1	CSIB0	506
CB0CTL2	CSIB0 控制寄存器 2	CSIB0	507
CB0REIC	中断控制寄存器	INTC	559
CB0RIC	中断控制寄存器	INTC	559
CB0RX	CSIB0 接收数据寄存器	CSIB0	502
CB0RXL	CSIB0 接收数据寄存器 L	CSIB0	502
CB0STR	CSIB0 状态寄存器	CSIB0	509
CB0TIC	中断控制寄存器	INTC	559
CB0TX	CSIB0 发送数据寄存器	CSIB0	502
CB0TXL	CSIB0 发送数据寄存器 L	CSIB0	502
CKC	时钟控制寄存器	CG	125
CLM	时钟监控模式寄存器	CG	129
HZA0CTL0	高阻输出控制寄存器 00	Timer	372
HZA0CTL1	高阻输出控制寄存器 01	Timer	372

符号	名称	单元	页码
HZA1CTL0	高阻输出 控制 寄存器 10	定时器	372
HZA1CTL1	高阻输出 控制 寄存器 11	定时器	372
IMR0	中断 屏蔽 寄存器 0	INTC	562
IMR0H	中断 屏蔽 寄存器 0H	INTC	562
IMR0L	中断 屏蔽 寄存器 0L	INTC	562
IMR1	中断 屏蔽 寄存器 1	INTC	562
IMR1H	中断 屏蔽 寄存器 1H	INTC	562
IMR1L	中断 屏蔽 寄存器 1L	INTC	562
IMR2	中断 屏蔽 寄存器 2	INTC	562
IMR2H	中断 屏蔽 寄存器 2H	INTC	562
IMR2L	中断 屏蔽 寄存器 2L	INTC	562
IMR3	中断 屏蔽 寄存器 3	INTC	562
IMR3H	中断 屏蔽 寄存器 3H	INTC	562
IMR3L	中断 屏蔽 寄存器 3L	INTC	562
IMS	内存大小设定寄存器	CPU	45
INTF0	外部 中断 下降沿 指定 寄存器 0	INTC	567
INTPNRC	外部 中断 噪声抑制 控制 寄存器	INTC	566
INTR0	外部 中断 上升沿 指定 寄存器 0	INTC	567
ISPR	服务中优先级寄存器	INTC	564
LVIIC	中断 控制 寄存器	INTC	559
LVIM	低电压检测寄存器	POC/LVI	596
LVIS	低电压检测电平选择 寄存器	POC/LVI	597
OSTS	振荡稳定时间选择 寄存器	CG	128
P0	端口 0 寄存器	端口	69
P1	端口 1 寄存器	端口	74
P2	端口 2 寄存器	端口	86
P3	端口 3 寄存器	端口	91
P4	端口 4 寄存器	端口	98
PCC	处理器 时钟 控制 寄存器	CG	125
PDLL	端口 DL 寄存器 L	端口	106
PFC1	端口 1 功能 控制 寄存器	端口	76
PFC3	端口 3 功能 控制 寄存器	端口	92
PFC4	端口 4 功能 控制 寄存器	端口	99
PFCE1	端口 1 功能 控制扩展寄存器	端口	76
PIC0	中断 控制 寄存器	INTC	559
PIC1	中断 控制 寄存器	INTC	559
PIC2	中断 控制 寄存器	INTC	559
PIC3	中断 控制 寄存器	INTC	559
PIC4	中断 控制 寄存器	INTC	559
PIC5	中断 控制 寄存器	INTC	559
PIC6	中断 控制 寄存器	INTC	559
PLLCTL	PLL 控制 寄存器	CG	124
PM0	端口 0 模式 寄存器	端口	69

符号	名称	单元	页码
PM1	端口 1 模式 寄存器	端口	74
PM2	端口 2 模式 寄存器	端口	86
PM3	端口 3 模式 寄存器	端口	91
PM4	端口 4 模式 寄存器	端口	98
PMC0	端口 0 模式 控制 寄存器	端口	70
PMC1	端口 1 模式 控制 寄存器	端口	75
PMC2	端口 2 模式 控制 寄存器	端口	87
PMC3	端口 3 模式 控制 寄存器	端口	92
PMC4	端口 4 模式 控制 寄存器	端口	99
PMDLL	端口 DL 模式 寄存器 L	端口	106
PRCMD	命令寄存器	CPU	58
PRCMD2	命令寄存器 2	CPU	58
PSC	节电控制 寄存器	CPU	126, 581
PSMR	节电 模式 寄存器	CPU	127, 582
PU0	上拉电阻选项寄存器 0	端口	70
PU1	上拉电阻选项寄存器 1	端口	78
PU2	上拉电阻选项寄存器 2	端口	88
PU3	上拉电阻选项 寄存器 3	端口	93
PU4	上拉电阻选项寄存器 4	端口	100
PUDLL	上拉电阻选项寄存器 DLL	端口	106
RAMS	内部 RAM 数据 状态 寄存器	POC/LVI	598
RESF	复位源标志寄存器	Reset	590
RESF2	复位源标志寄存器 2	Reset	591
SYS	系统 状态 寄存器	CPU	59
SYS2	系统 状态 寄存器 2	CPU	59
TM0CMP0	TMM0 比较寄存器 0	定时器	352
TM0CTL0	TMM0 控制 寄存器 0	定时器	353
TM0EQIC0	中断 控制 寄存器	INTC	559
TP0CCIC0	中断 控制 寄存器	INTC	559
TP0CCIC1	中断 控制 寄存器	INTC	559
TP0CCR0	TMP0 捕获/比较寄存器 0	定时器	151
TP0CCR1	TMP0 捕获/比较寄存器 1	定时器	153
TP0CNT	TMP0 计数器读取缓冲寄存器	定时器	155
TP0CTL0	TMP0 控制 寄存器 0	定时器	143
TP0CTL1	TMP0 控制 寄存器 1	定时器	144
TP0IOC0	TMP0 I/O 控制 寄存器 0	定时器	146
TP0IOC1	TMP0 I/O 控制 寄存器 1	定时器	148
TP0IOC2	TMP0 I/O 控制 寄存器 2	定时器	149
TP0OPT0	TMP0 选项寄存器 0	定时器	150
TP0OVIC	中断 控制 寄存器	INTC	559
TP1CCIC0	中断 控制 寄存器	INTC	559
TP1CCIC1	中断 控制 寄存器	INTC	559
TP1CCR0	TMP1 捕获/比较寄存器 0	定时器	151

符号	名称	单元	页码
TP1CCR1	TMP1 捕获/比较寄存器 1	定时器	153
TP1CNT	TMP1 计数器读取缓冲寄存器	定时器	155
TP1CTL0	TMP1 控制寄存器 0	定时器	143
TP1CTL1	TMP1 控制寄存器 1	定时器	144
TP1OPT0	TMP1 选项寄存器 0	定时器	150
TP1OVIC	中断控制寄存器	INTC	559
TP2CCIC0	中断控制寄存器	INTC	559
TP2CCIC1	中断控制寄存器	INTC	559
TP2CCR0	TMP2 捕获/比较寄存器 0	定时器	151
TP2CCR1	TMP2 捕获/比较寄存器 1	定时器	153
TP2CNT	TMP2 计数器读取缓冲寄存器	定时器	155
TP2CTL0	TMP2 控制寄存器 0	定时器	143
TP2CTL1	TMP2 控制寄存器 1	定时器	144
TP2IOC0	TMP2 I/O 控制寄存器 0	定时器	146
TP2IOC1	TMP2 I/O 控制寄存器 1	定时器	148
TP2IOC2	TMP2 I/O 控制寄存器 2	定时器	149
TP2OPT0	TMP2 选项寄存器 0	定时器	150
TP2OVIC	中断控制寄存器	INTC	559
TP3CCIC0	中断控制寄存器	INTC	559
TP3CCIC1	中断控制寄存器	INTC	559
TP3CCR0	TMP3 捕获/比较寄存器 0	定时器	151
TP3CCR1	TMP3 捕获/比较寄存器 1	定时器	153
TP3CNT	TMP3 计数器读取缓冲寄存器	定时器	155
TP3CTL0	TMP3 控制寄存器 0	定时器	143
TP3CTL1	TMP3 控制寄存器 1	定时器	144
TP3IOC0	TMP3 I/O 控制寄存器 0	定时器	146
TP3OPT0	TMP3 选项寄存器 0	定时器	150
TP3OVIC	中断控制寄存器	INTC	559
TQ0CCIC0	中断控制寄存器	INTC	559
TQ0CCIC1	中断控制寄存器	INTC	559
TQ0CCIC2	中断控制寄存器	INTC	559
TQ0CCIC3	中断控制寄存器	INTC	559
TQ0CCR0	TMQ0 捕获/比较寄存器 0	定时器	251
TQ0CCR1	TMQ0 捕获/比较寄存器 1	定时器	253
TQ0CCR2	TMQ0 捕获/比较寄存器 2	定时器	255
TQ0CCR3	TMQ0 捕获/比较寄存器 3	定时器	257
TQ0CNT	TMQ0 计数器读取缓冲寄存器	定时器	259
TQ0CTL0	TMQ0 控制寄存器 0	定时器	244
TQ0CTL1	TMQ0 控制寄存器 1	定时器	244
TQ0IOC0	TMQ0 I/O 控制寄存器 0	定时器	246
TQ0IOC1	TMQ0 I/O 控制寄存器 1	定时器	248
TQ0IOC2	TMQ0 I/O 控制寄存器 2	定时器	249
TQ0OPT0	TMQ0 选项寄存器 0	定时器	250

符号	名称	单元	页码
TQ0OVIC	中断控制寄存器	INTC	559
TQ1CCIC0	中断控制寄存器	INTC	559
TQ1CCIC1	中断控制寄存器	INTC	559
TQ1CCIC2	中断控制寄存器	INTC	559
TQ1CCIC3	中断控制寄存器	INTC	559
TQ1CCR0	TMQ1 捕获/比较寄存器 0	定时器	251
TQ1CCR1	TMQ1 捕获/比较寄存器 1	定时器	253
TQ1CCR2	TMQ1 捕获/比较寄存器 2	定时器	255
TQ1CCR3	TMQ1 捕获/比较寄存器 3	定时器	257
TQ1CNT	TMQ1 计数器读取缓冲寄存器	定时器	259
TQ1CTL0	TMQ1 控制寄存器 0	定时器	244
TQ1CTL1	TMQ1 控制寄存器 1	定时器	244
TQ1DTC	TMQ1 死期比较寄存器	定时器	363
TQ1IOC0	TMQ1 I/O 控制寄存器 0	定时器	246
TQ1IOC3	TMQ1 I/O 控制寄存器 3	定时器	369
TQ1OPT0	TMQ1 选项寄存器 0	定时器	250, 364
TQ1OPT1	TMQ1 选项寄存器 1	定时器	365
TQ1OPT2	TMQ1 选项寄存器 2	定时器	366
TQ1OPT3	TMQ1 选项寄存器 3	定时器	368
TQ1OVIC	中断控制寄存器	INTC	559
UA0CTL0	UARTA0 控制寄存器 0	UARTA0	477
UA0CTL1	UARTA0 控制寄存器 1	UARTA0	478
UA0CTL2	UARTA0 控制寄存器 2	UARTA0	478
UA0OPT0	UARTA0 选项控制寄存器 0	UARTA0	479
UA0REIC	中断控制寄存器	INTC	559
UA0RIC	中断控制寄存器	INTC	559
UA0RX	UARTA0 接收数据寄存器	UARTA0	481
UA0STR	UARTA0 状态寄存器	UARTA0	479
UA0TIC	中断控制寄存器	INTC	559
UA0TX	UARTA0 发送数据寄存器	UARTA0	481
UA1CTL0	UARTA1 控制寄存器 0	UARTA1	477
UA1CTL1	UARTA1 控制寄存器 1	UARTA1	478
UA1CTL2	UARTA1 控制寄存器 2	UARTA1	478
UA1OPT0	UARTA1 选项控制寄存器 0	UARTA1	479
UA1REIC	中断控制寄存器	INTC	559
UA1RIC	中断控制寄存器	INTC	559
UA1RX	UARTA1 接收数据寄存器	UARTA1	481
UA1STR	UARTA1 状态寄存器	UARTA1	479
UA1TIC	中断控制寄存器	INTC	559
UA1TX	UARTA1 发送数据寄存器	UARTA1	481
VSWC	系统等待控制寄存器	BCU	61
WDTE	看门狗定时器使能寄存器	WDT	425
WDTM	看门狗定时器模式寄存器	WDT	424

附录 C 指令集列表

C.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。
×	根据结果置位或清 0。
R	恢复先前的保存值。

(6) 条件码

条件名称 (cond)	条件码 (cccc)	条件公式	解释
V	0 0 0 0	$OV = 1$	溢出
NV	1 0 0 0	$OV = 0$	无溢出
C/L	0 0 0 1	$CY = 1$	进位 小(相比小)
NC/NL	1 0 0 1	$CY = 0$	无进位 不小(相比大于或等于)
Z	0 0 1 0	$Z = 1$	0
NZ	1 0 1 0	$Z = 0$	非 0
NH	0 0 1 1	$(CY \text{ or } Z) = 1$	不高(相比小于或等于 1)
H	1 0 1 1	$(CY \text{ or } Z) = 0$	高(相比大于)
S/N	0 1 0 0	$S = 1$	负
NS/P	1 1 0 0	$S = 0$	正
T	0 1 0 1	-	恒为(无条件的)
SA	1 1 0 1	$SAT = 1$	饱和
LT	0 1 1 0	$(S \text{ xor } OV) = 1$	小于有符号数。
GE	1 1 1 0	$(S \text{ xor } OV) = 0$	大于或等于有符号数。
LE	0 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 1$	小于或等于有符号数。
GT	1 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 0$	大于有符号数。

C.2 指令集 (按字母先后次序)

(1/6)

助记符	操作数	操作码	操作		执行时钟			标志				
					i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]		1	1	1	x	x	x	x	
	imm5,reg2	rrrrr010010iiii	GR[reg2]←GR[reg2]+sign-extend(imm5)		1	1	1	x	x	x	x	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1	x	x	x	x	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]		1	1	1		0	x	x	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)		1	1	1		0	x	x	
Bcond	disp9	dddd1011dddcccc 注 1	如果条件满足	如果条件满足	2	2	2					
			则: PC←PC+sign-extend(disp9)	如果条件不满足	注2	注2	注2					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) GR[reg2] (31 : 24) GR[reg2] (7 : 0) GR[reg2] (15 : 8)		1	1	1	x	0	x	x	
			GR[reg3]←GR[reg2] (7 : 0) GR[reg2] (15 : 8) GR[reg2] (23 : 16) GR[reg2] (31 : 24)		1	1	1	x	0	x	x	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) GR[reg2] (15 : 8) GR[reg2] (23 : 16) GR[reg2] (31 : 24)		1	1	1	x	0	x	x	
CALLT	imm6	0000001000iiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Halfword))		4	4	4					
CLR1	bit#3,disp16[reg1]	10bbb111110RRRRR dddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)		3	3	3				x	
	reg2,[reg1]	rrrrr111111RRRRR 000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)		3	3	3				x	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiii wwwww011000cccc0	如果条件满足 则 GR[reg3]←sign-extended(imm5) 否则 GR[reg3]←GR[reg2]		1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRRR wwwww011001cccc0	如果条件满足 则 GR[reg3]←GR[reg1] 否则 GR[reg3]←GR[reg2]		1	1	1					
CMP	reg1,reg2	rrrrr001111RRRRR	result←GR[reg2]-GR[reg1]		1	1	1	x	x	x	x	
	imm5,reg2	rrrrr010011iiii	result←GR[reg2]-sign-extend(imm5)		1	1	1	x	x	x	x	
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

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 LLLLLLLLLLLL000000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 重复以上两个步骤直到表 12 中所有寄存器被载入。	n+1 注4	n+1 注4	n+1 注4						
	imm5,list12,[reg1]	0000011001iiiiL LLLLLLLLLLLLRRRRR 注5	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 重复以上两个步骤直到表 12 中所有寄存器载入 PC←GR[reg1]	n+3 注4	n+3 注4	n+3 注4						
DIV	reg1,reg2,reg3	rrrrr11111RRRRR wwwww01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×		
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	rrrrr11110dddddd dddddddddddddd0 注7	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2						
JMP	[reg1]	0000000011RRRRR	PC←GR[reg1]	3	3	3						
JR	disp22	0000011110dddddd dddddddddddddd0 注7	PC←PC+sign-extend(disp22)	2	2	2						
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	注11						
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR ddddddddddddddd1 注8, 10	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	注11						

助记符	操作数	操作码	操作	执行时钟			标志					
				i	r	l	CY	OV	S	Z	SAT	
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR dddddddddddddd0 注 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	注 11						
LDSR	reg2,regID	rrrrr111111RRRRR 000000000100000 注 12	SR[regID]←GR[reg2]	除非 regID = PSW	1	1	1					
			regID = PSW	1	1	1	x	x	x	x	x	
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR dddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	注 11						
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR dddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)	1	1	注 11						
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]	1	1	1						
	imm5,reg2	rrrrr010000iiii	GR[reg2]←sign-extend(imm5)	1	1	1						
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii iiiiiiiiiiiiiiii	GR[reg1]←imm32	2	2	2						
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1						
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 0 ¹⁶)	1	1	1						
MUL ^{注 22}	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] GR[reg2]←GR[reg2]xGR[reg1] 注 14	1	4	5						
	imm9,reg2,reg3	rrrrr111111iiii wwwww01001111100 注 13	GR[reg3] GR[reg2]←GR[reg2]xsign-extend(imm9)	1	4	5						
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2] ⁶ xGR[reg1] ⁶	1	1	2						
	imm5,reg2	rrrrr010111iiii	GR[reg2]←GR[reg2] ⁶ xsign-extend(imm5)	1	1	2						
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] ⁶ ximm16	1	1	2						
MULU ^{注 22}	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] GR[reg2]←GR[reg2]xGR[reg1] 注 14	1	4	5						
	imm9,reg2,reg3	rrrrr111111iiii wwwww0100111110 注 13	GR[reg3] GR[reg2]←GR[reg2]xzero-extend(imm9)	1	4	5						
NOP		0000000000000000	至少一个时钟周期内不作任何操作。	1	1	1						
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])	1	1	1		0	x	x		
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)	3 注 3	3 注 3	3 注 3					x	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)	3 注 3	3 注 3	3 注 3					x	

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiL LLLLLLLLLLLL00001	Store-memory(sp-4,GR[reg in list12],Word) sp←sp-4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 注4	n+1 注4	n+1 注4					
	list12,imm5, sp/imm ^{#15}	0000011110iiiiL LLLLLLLLLLLLff011 imm16/imm32 注 16	Store-memory(sp-4,GR[reg in list12],Word) sp←sp+4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend (imm5) ep←sp/imm	n+2 注4 注17	n+2 注4 注17	n+2 注4 注17					
RETI		000001111100000 0000000101000000	如果 PSW.EP=1 那么 PC ←EIPC PSW ←EIPSW 否则, 如果 PSW.NP=1 则 PC ←FEPC PSW ←FEPSW 否则 PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R
SAR	reg1,reg2	rrrrr111111RRRRR 000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr1111110cccc 0000001000000000	如果条件满足 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	rrrrr1111110cccc 0000000000000000	如果条件满足 则 GR[reg2]←00000001H 否则 GR[reg2]←00000000H	1	1	1					

助记符	操作数	操作码	操作	执行时钟			标志					
				i	r	l	CY	OV	S	Z	SAT	
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 注3	3 注3	3 注3					x	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 注3	3 注3	3 注3					x	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	x	0	x	x		
	imm5,reg2	rrrrr010110iiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	x	0	x	x		
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	x	0	x	x		
	imm5,reg2	rrrrr010100iiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	x	0	x	x		
SLD.B	disp7[ep],reg2	rrrrr0110ddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	注9						
SLD.BU	disp4[ep],reg2	rrrrr0000110ddd 注 18	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	注9						
SLD.H	disp8[ep],reg2	rrrrr1000ddddd 注 19	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	注9						
SLD.HU	disp5[ep],reg2	rrrrr0000111ddd 注 s 18, 20	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	注9						
SLD.W	disp8[ep],reg2	rrrrr1010ddddd0 注 21	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	注9						
SST.B	reg2,disp7[ep]	rrrrr0111ddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1						
SST.H	reg2,disp8[ep]	rrrrr1001ddddd 注 19	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Halfword)	1	1	1						
SST.W	reg2,disp8[ep]	rrrrr1010ddddd1 注 21	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1						
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1						
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd0 注 8	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Halfword)	1	1	1						
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd1 注 8	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Word)	1	1	1						
STSR	regID,reg2	rrrrr111111RRRRR 000000001000000	GR[reg2]←SR[regID]	1	1	1						

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]-GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]-GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	0000000010RRRRR	adr←-(PC+2) + (GR [reg1] logically shift left by 1) PC←-(PC+2) + (sign-extend (Load-memory (adr,Halfword)) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15 : 0))	1	1	1					
TRAP	vector	000001111111iiii 0000000100000000	EIPC ←PC+4 (Restored PC) EIPSW ←PSW ECR.EICC ←Interrupt code PSW.EP ←-1 PSW.ID ←-1 PC ←-00000040H (when vector is 00H to 0FH) 00000050H (when vector is 10H to 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1	0	×	×		
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr,bit#3))	3 注3	3 注3	3 注3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3 注3	3 注3	3 注3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1	0	×	×		
XORI	imm16,reg1,reg2	rrrrr110101RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1	0	×	×		
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15 : 0))	1	1	1					

- 注
1. 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 `regID`
`RRRRR` = 指定 `reg2`
13. `iiii`: `imm9` 的低 5 位。
`IIII`: `imm9` 的高 4 位。
14. 不要将为通用寄存器 `reg1` 和 `reg3` 指定同一寄存器。
15. `sp/imm`: 由次操作码的位 19 和位 20 指定。
16. `ff = 00`: 将 `sp` 载入 `ep`。
01: 将符号扩展的 16 位立即数(位 47 至 32)载入 `ep`。
10: 将经过 16 位逻辑左移的 16 位立即数 (位 47 至 32) 载入 `ep`。
11: 将 32 位立即数 (位 63 至 32) 载入 `ep`。
17. `if imm = imm32, n + 3` 时钟。
18. `rrrrr`: 非 00000。
19. `dddddd`: `disp8` 的高 8 位。
20. `dddd`: `disp5` 的高 4 位。
21. `dddddd`: `disp8` 的高 6 位。
22. 当执行“`MUL reg1, reg2, reg3`”和“`MULU reg1, reg2, reg3`”指令时，不要使寄存器组合满足以下所有条件。如果满足以下条件的指令执行，则操作 无法保证。
- `reg1 = reg3`
 - `reg1 ≠ reg2`
 - `reg1 ≠ r0`
 - `reg3 ≠ r0`

D.1 本辑主要修订之处

(1/3)

页码	描述
遍及全文	<ul style="list-style-type: none"> • 所有产品从 开发之中 → 开发完毕所应有的更改 • 增加了 PG-FP5
p. 23	给 2.1 (1) 端口引脚 增加了说明
p. 37	更改了 3.2.2 (6) 异常/调试陷阱状态存储寄存器(DBPC, DBPSW) 中的说明
p. 56	增加了对 3.4.7 特殊寄存器 的说明
p. 58	增加了对 3.4.7 (2) (a) 命令寄存器 (PRCMD) 的说明
p. 58	增加了对 3.4.7 (2) (b) 命令寄存器 2 (PRCMD2) 的说明
p. 117	更改了 表 4-12 噪声抑制器 的说明
p. 118	增加了 图 4-22 噪声消除时序举例
p. 119	增加了 4.6 (1) 外部中断噪声消除控制寄存器 (INTPNRC)
p. 125	增加了对 5.3 (2) 时钟控制寄存器 (CKC) 的说明
p. 125	增加了对 5.3 (3) 处理器时钟控制寄存器 (PCC) 的说明
p. 126	增加了对 5.3 (4) 节电控制寄存器 (PSC) 的说明
p. 129	增加了对 5.3 (7) 时钟监控模式寄存器 (CLM) 的说明
p. 137	增加了对 表 6-1 TMPn 概览 的说明
p. 145	更改了 6.4 (2) TMPn 控制寄存器 1 (TPnCTL1) 的说明
p. 147	更改了 6.4 (3) TMPm I/O 控制寄存器 0 (TPmIOC0) 的说明
p. 159	更改了 6.6 (1) (a) 计数器启动操作 的说明
pp. 166 至 168	更改了 图 6-11 间隔定时器模式操作的寄存器设置 的说明
p. 169	增加了对 图 6-12 间隔定时器模式软件处理过程 的说明
p. 176	增加了 6.6.1 (3) 由外部事件计数器输入操作(TIPk0)
p. 177	增加了对 6.6.2 外部事件计数器模式(TPkMD2 至 TPkMD0 位 = 001) 的说明
p. 180	增加了对 图 6-18 外部事件计数器模式中操作的寄存器设置 的说明
p. 182	增加了对 6.6.2 (2) 外部事件计数器模式的操作时序 的说明
p. 186	更改了 图 6-23 外部触发脉冲输出模式的配置 的说明
p. 187	更改了 图 6-24 外部触发脉冲输出模式基本时序
p. 187	增加了对 6.6.3 外部触发脉冲输出模式(TPmMD2 至 TPmMD0 位 = 010) 的说明
pp. 188, 189	更改了 图 6-25 外部触发脉冲输出模式的寄存器设置 的说明
p. 195	更改了 6.6.3 (2) (b) 0%/100% PWM 输出波形 的图示
p. 199	更改了 图 6-27 单次脉冲输出模式的配置
pp. 201, 202	更改了 图 6-29 单次脉冲输出模式的寄存器配置
p. 203	更改了 图 6-30 单次脉冲输出模式的软件处理流程
p. 204	更改了 6.6.4 (2) (a) 重写 TPmCCRa 寄存器的附注 的图示
p. 208	增加了对 图 6-33 PWM 输出模式的寄存器设置 的说明

页码	描述
p. 213	更改了 6.6.5 (2) (b) 0%/100% PWM 的输出波形 的说明
p. 232	更改了 图 6-41 脉冲宽度测量模式的配置 的说明
p. 233	更改了 6.6.7 脉冲宽度测量模式 (TPkMD2 至 TPkMD0 位 = 110) 的说明
pp. 234, 235	更改了 图 6-43 脉冲宽度测量模式的寄存器配置 的说明
p. 236	删除了 图 6-44 脉冲宽度测量模式的软件处理流程 中的说明
p. 238	增加了对 表 7-1 TMQn 概况 的说明
p. 246	更改了 7.4 (3) TMQn I/O 控制寄存器 0 (TQnIOC0) 的说明
p. 262	更改了 7.6 (1) (a) 计数器启动操作 的说明
pp. 270 to 272	更改了 图 7-10 间隔定时器模式操作的寄存器设置 的说明
p. 273	增加了对 图 7-11 间隔定时器模式的软件处理流程 的说明
p. 278	更改了 7.6.1 (2) (d) TQnCCR1 至 TQnCCR3 寄存器的操作 的说明
p. 280	增加了对 7.6.1 (3) 由外部事件计数器输入的操作(EVTQ0)
p. 281	增加了对 7.6.2 外部事件计数器模式 (TQ0MD2 至 TQ0MD0 位 = 001) 的说明
p. 284	增加了对 图 7-17 外部事件计数器模式操作的寄存器设置 的说明
p. 286	增加了对 7.6.2 (2) 外部事件计数器的操作时序 的说明
p. 292	更改了 图 7-23 外部触发脉冲输出模式的基本时序 中的图示说明
p. 293	增加了对 7.6.3 外部触发脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 010) 的说明
p. 301	更改了 7.6.3 (2) (b) 0%/100% PWM 输出波形 的说明
p. 309	更改了 图 7-28 单次脉冲输出模式中的寄存器设置 的说明
pp. 310, 311	更改了 图 7-29 单次脉冲输出模式中的软件处理流程 的说明
p. 323	更改了 7.6.5 (2) (b) 0%/100% PWM 输出波形 的说明
p. 346	更改了 7.6.7 脉冲宽度测量模式(TQ0MD2 至 TQ0MD0 位= 110)
p. 363	增加了对 9.2 (1) TMQ1 死期比较寄存器 (TQ1DTC) 的说明
p. 365	增加了对 9.3 (2) TMQ1 选项寄存器 1 (TQ1OPT1) 的说明
p. 366	增加了对 9.3 (3) TMQ1 选项寄存器 2 (TQ1OPT2) 的说明
p. 370	增加了对 图 9-3 TOQ1Tm 和 TOQ1Bm 引脚的输出控制 (无死期) 的说明
p. 378	更改了 图 9-5 6-相位 PWM 输出模式的概况
p. 383	更改了 图 9-9 0% PWM 输出波形 (有死期)
p. 384	更改了 图 9-10 100% PWM 输出波形 (有死期)
p. 385	更改了 图 9-11 PWM 输出波形从 0% 至 100% 以及从 100%至 0% (有死期)
p. 389	增加了对 9.4.3 中断逸选功能 的说明
p. 438	更改了 11.4.1 基本操作
p. 475	更改了 图 12-1 UARTAn 的功能框图
p. 476	更改了 12.2 (5) UARTAn 状态寄存器(UAnSTR) 的说明
p. 477	更改了 12.3 (1) UARTAn 控制寄存器 0 (UAnCTL0) 的说明
p. 487	更改了 图 12-5 连续发送操作时序
p. 523	更改了 13.4.7 (2) 操作时序 的图示说明
p. 559	增加了对 14.3.4 中断控制寄存器 (xxICn) 的说明
p. 572	增加了对 14.6.1 (2) 恢复 的说明

页码	描述
p. 581	增加了对 15.2 (1) 节电控制寄存器 (PSC) 的说明
p. 587	增加了对 15.5.1 设置和操作状态 的说明
p. 590	增加了对 16.2 (1) 复位源标志寄存器 (RESF) 的说明
p. 596	增加了对 16.3.3 (3) (a) 低电压检测寄存器(LVIM) 的说明
p. 598	增加了对 16.3.3 (3) (c) 内部 RAM 数据状态寄存器 (RAMS) 的说明
p. 610	更改了 表 18-2 基本功能 的说明
p. 610	更改了 表 18-3 安全功能
p. 611	增加了 表 18-4 安全设置
p. 612	增加了 18.3 (1) 通过 PG-FP4 和 PG-FP5 (安全标志设置)的安全设置
p. 619	更改了 图 18-7 操作 Flash 存储器的过程
p. 621	更改了 表 18-7 Flash 存储器控制命令 的说明
p. 629	更改了 图 18-17 标准自编程流程 的说明
p. 630	更改了 表 18-10 Flash 功能列表 的说明
p. 634	增加了 第 19 章 电气参数 中 时钟振荡器特性 (i) Murata Mfg. Co., Ltd.: 陶瓷谐振器 (TA = -40 to +85°C)
p. 636	增加了对 第 19 章 电气参数 中 DC 特性 的说明
p. 642	更改了 第 19 章 电气参数 中 CSIB 时序 的说明
p. 645	更改了 第 19 章 电气参数 中 A/D 转换器 0, 1 的指标 的说明
p. 647	增加了对 第 19 章 电气参数 中 低电压检测器 (LVI) 的说明
p. 650	更改了 第 19 章 电气参数 中(3) 编程特性 之 Flash 存储编程特性 的说明
p. 652	增加了 第 21 章 建议焊接条件
p. 669	增加了 附录 D 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

成都市二环路南三段 15 号天华大厦 7 楼 703 室

电话：(+86)28-8512-5224

传真：(+86)28-8512-5334