



用户手册

V850ES/HE2

32 位单片微控制器

硬件

μ PD70F3700

μ PD70F3701

文档编号: U17720CA2V0UD00 (第二版)
发行日期: 2008 年 1 月 CP(K)

© 日本电气电子株式会社 2008
日本印刷

[备忘录]

① 输入引脚处的电压波形

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

② 未使用的输入引脚的处理

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

③ ESD防护措施

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

④ 初始化之前的状态

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

⑤ 电源开关顺序

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

对于每个设备电源的正确开关顺序必须依据设备的规范说明分别进行判断。

⑥ 电源关闭状态下的输入信号

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

每个设备电源关闭时的信号输入必须依据设备的规范说明分别进行判断。

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

MINICUBE 是 **NEC Electronics Corporation** 在日本和德国的注册商标或在美国的商标。

Applilet 是 **NEC Electronics** 在日本、德国、中国、中国香港、韩国、英国，以及美国的注册商标。

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

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

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

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

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

ITRON 是 **Industrial TRON** 的缩写。.

- 本档所登载的内容有效期截止至 2007 年 11 月，信息先于产品的生产周期发布。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。否则因本档所登载内容引发的错误，本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的，对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或其他人所有的专利、版权以及其它知识产权作出任何明示或默示的许可及授权。
- 本文件中的电路、软件以及相关信息仅用以说明半导体产品的运作和应用实例。用户如在设备设计中应用本文件中的电路、软件以及相关信息，应自行负责。对于用户或其他人因使用了上述电路、软件以及相关信息而引起的任何损失，本公司概不负责。
- 虽然本公司致力于提高半导体产品的质量及可靠性，但用户应同意并知晓，我们仍然无法完全消除出现产品缺陷的可能。为了最大限度地减少因本公司半导体产品故障而引起的对人身、财产造成损害（包括死亡）的危险，用户务必在其设计中采用必要的安全措施，如冗余度、防火和防故障等安全设计。
- 本公司产品质量分为：

“标准等级”、“专业等级”以及“特殊等级”三种质量等级。

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

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

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

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

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

（注）

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

M5 02.11-1

引言

用户 本手册适用于那些希望了解 V850ES/HE2 产品的功能并准备利用 V850ES/HE2 进行应用系统开发的用户。

目的 本手册用于帮助用户了解下面 V850ES/HE2 组件中描述的功能。

组件 V850ES/HE2 产品手册主要分为两个部分：硬件(本手册)和架构（**V850ES 架构用户手册**）。

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

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

如何获悉 V850ES/HE2 的全部功能

→ 请依照 **目录** 阅读本手册。

如何获取某寄存器的详细信息

→ 请参考 **附录 B 寄存器索引**。

如何了解指令的详细功能

→ 请参阅另外提供的 **V850ES 架构用户手册**。

如何了解 V850ES/HE2 产品的电气指标

→ 请参考 **第 25 章 电气特性**。

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

标注“<R>”表示主要修订部分。修订部分可以很方便地通过在 PDF 文件中拷贝“<R>”并通过在“Find what:”中指定来搜索查询。

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

相关文档

本手册中指出的相关文档包括了最初的版本，但未注明。

与 V850ES/HE2 产品相关的文档

文档名称	文档编号
V850ES 架构用户手册	U15943E
V850ES/HE2 硬件用户手册	本手册

与开发工具相关的文档

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

目录

第 1 章	介绍	17
1.1	概要	17
1.2	特性	19
1.3	应用领域	19
1.4	订货信息	20
1.5	引脚图（俯视图）	21
1.6	功能模块配置	23
1.6.1	内部结构框图	23
1.6.2	内部单元	24
第 2 章	引脚功能	26
2.1	引脚功能列表	26
2.2	引脚功能描述	31
2.3	I/O 引脚电路类型以及不使用引脚的推荐连接	37
2.4	I/O 引脚电路	39
2.5	注意事项	40
第 3 章	CPU 功能	41
3.1	特点	41
3.2	CPU 寄存器组	42
3.2.1	程序寄存器组	43
3.2.2	系统寄存器组	44
3.3	操作模式	50
3.3.1	指定操作模式	50
3.4	地址空间	51
3.4.1	CPU 地址空间	51
3.4.2	CPU 地址空间的绕回	52
3.4.3	存储器映射	53
3.4.4	区域	55
3.4.5	地址空间的推荐使用方法	58
3.4.6	周边 I/O 寄存器	60
3.4.7	特殊寄存器	67
3.4.8	注意事项	71
第 4 章	端口功能	74
4.1	特点	74
4.2	端口的基本配置	74
4.3	端口功能	76
4.3.1	端口功能的操作	76
4.3.2	关于设置端口引脚的注释	77
4.3.3	端口 0	78
4.3.4	端口 3	82
4.3.5	端口 4	87
4.3.6	端口 5	90

<R>

4.3.7	端口 7	96
4.3.8	端口 9	99
4.3.9	端口 CM	108
4.3.10	端口 DL	111
4.3.11	复用功能为片上调试功能的端口引脚	113
4.3.12	将端口引脚用作复用功能引脚时的寄存器设置	114
4.4	端口结构图	119
4.5	注意事项	143
4.5.1	设置端口引脚的注意事项	143
第 5 章	时钟产生功能	144
5.1	概述	144
5.2	结构图	145
5.3	寄存器	147
5.4	操作	152
5.4.1	各时钟的操作	152
5.4.2	时钟输出功能	152
5.5	PLL 功能	153
5.5.1	概述	153
5.5.2	寄存器	153
5.5.3	使用方法	157
第 6 章	16 位定时器/事件计数器 P (TMP)	158
6.1	概述	158
6.2	功能	158
6.3	配置	159
6.4	寄存器	161
6.5	操作	175
6.5.1	间隔定时器模式 (TPnMD2 至 TPnMD0 位 = 000)	176
6.5.2	外部事件计数模式 (TPnMD2 至 TPnMD0 位 = 001)	186
6.5.3	外部触发脉冲输出模式 (TPnMD2 至 TPnMD0 位 = 010)	194
6.5.4	单脉冲输出模式 (TPnMD2 至 TPnMD0 位 = 011)	206
6.5.5	PWM 输出模式 (TPnMD2 至 TPnMD0 位 = 100)	213
6.5.6	自由运行定时器模式 (TPnMD2 至 TPnMD0 位 = 101)	222
6.5.7	脉宽测量模式 (TPnMD2 至 TPnMD0 位 = 110)	239
6.5.8	定时器输出操作	245
6.6	定时器调谐操作功能	246
6.7	选择器功能	250
6.8	注意事项	252
第 7 章	16 位定时器/事件计数器 Q (TMQ)	253
7.1	概述	253
7.2	功能	253
7.3	配置	254
7.4	寄存器	257
7.5	操作	275
7.5.1	间隔定时器模式 (TQ0MD2 至 TQ0MD0 位 = 000)	276

7.5.2	外部事件计数模式 (TQ0MD2 至 TQ0MD0 位 = 001)	285
7.5.3	外部触发脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 010)	294
7.5.4	单次脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 011)	307
7.5.5	PWM 输出模式 (TQ0MD2 至 TQ0MD0 位 = 100)	316
7.5.6	自由运行定时器模式 (TQ0MD2 至 TQ0MD0 位 = 101)	327
7.5.7	脉宽测量模式 (TQ0MD2 至 TQ0MD0 位 = 110)	347
7.5.8	三角形波 PWM 模式 (TQ0MD2 至 TQ0MD0 位 = 111)	353
7.5.9	定时器输出操作	354
7.6	定时器调谐操作功能	355
7.7	注意事项	359
第 8 章	16 位间隔定时器 M (TMM)	360
8.1	概述	360
8.2	配制	361
8.3	寄存器	362
8.4	操作	363
8.4.1	间隔定时器模式	363
8.4.2	注意事项	367
第 9 章	钟表定时器功能	368
9.1	功能	368
9.2	配置	369
9.3	寄存器	371
9.4	操作	375
9.4.1	钟表定时器的操作	375
9.4.2	间隔定时器的操作	376
9.4.3	注意事项	377
第 10 章	看门狗定时器 2 的功能	378
10.1	功能	378
10.2	配置	379
10.3	寄存器	380
10.4	操作	383
第 11 章	A/D 转换器	384
11.1	概述	384
11.2	功能	384
11.3	配置	385
11.4	寄存器	388
11.5	操作	396
11.5.1	基本操作	396
11.5.2	触发模式	397
11.5.3	操作模式	399
11.5.4	掉电比较模式	403
11.6	注意事项	408
11.7	转换器特征表的阅读方法	412

第 12 章 异步串行接口 A (UARTA)	416
12.1 特性	416
12.2 配置	417
12.3 寄存器	419
12.4 中断请求信号	425
12.5 操作	426
12.5.1 数据格式	426
12.5.2 发送 / 接收格式	428
12.5.3 SBF 发送	430
12.5.4 SBF 接收	431
12.5.5 UART 发送	432
12.5.6 连续发送的过程	433
12.5.7 UART 接收	435
12.5.8 接收错误	436
12.5.9 校验类型和操作	438
12.5.10 接收数据的噪声过滤器	439
12.6 专用波特率发生器	440
12.7 注意事项	448
第 13 章 3 线长度可变串行 I/O (CSIB)	449
13.1 特征	449
13.2 配制	450
13.3 寄存器	452
13.4 中断请求信号	459
13.5 操作	460
13.5.1 单独发送模式 (主模式, 发送模式)	460
13.5.2 单独发送模式 (主模式, 接收模式)	462
13.5.3 单独发送模式 (主模式, 发送/接收模式)	464
13.5.4 单独发送模式 (从模式, 发送模式)	466
13.5.5 单独发送模式 (从模式, 接收模式)	468
13.5.6 单独发送模式 (从模式, 发送/接收模式)	470
13.5.7 连续发送模式 (主模式, 发送模式)	472
13.5.8 连续发送模式 (主模式, 接收模式)	474
13.5.9 连续发送模式 (主模式, 发送/接收模式)	477
13.5.10 连续发送模式 (从模式, 发送模式)	481
13.5.11 连续发送模式 (从模式, 接收模式)	483
13.5.12 连续发送模式 (从模式, 发送/接收模式)	486
13.5.13 接收错误	490
13.5.14 时钟时序	491
13.6 禁止操作时的输出引脚状态	493
13.7 波特率发生器	494
13.7.1 波特率产生	495
13.8 注意事项	496
第 14 章 中断/异常处理功能	497
14.1 特性	497
14.2 非屏蔽中断	500

14.2.1	操作	502
14.2.2	还原	503
14.2.3	NP 标志	504
14.3	可屏蔽中断	505
14.3.1	操作	505
14.3.2	恢复	507
14.3.3	可屏蔽中断的优先级	508
14.3.4	中断控制寄存器 (xxICn)	512
14.3.5	中断屏蔽寄存器 0 至 2 (IMR0 至 IMR2)	514
14.3.6	正在进行服务优先级寄存器 (ISPR)	515
14.3.7	ID 标志	516
14.3.8	看门狗定时器模式寄存器 2 (WDTM2)	516
14.4	软件异常	517
14.4.1	操作	517
14.4.2	恢复	518
14.4.3	EP 标志	519
14.5	异常陷阱	520
14.5.1	非法操作码定义	520
14.5.2	调试陷阱	522
14.6	外部中断请求输入引脚 (NMI 以及 INTP0 至 INTP7)	524
14.6.1	噪声消除	524
14.6.2	边沿检测	524
14.7	CPU 的中断确认时间	530
14.8	CPU 不对中断进行确认的时期	531
14.9	注意事项	531
第 15 章	按键中断功能	532
15.1	功能	532
15.2	寄存器	533
15.3	注意事项	533
第 16 章	待机功能	534
16.1	概述	534
16.2	寄存器	536
16.3	HALT 模式	539
16.3.1	设置和操作状态	539
16.3.2	释放 HALT 模式	539
16.4	IDLE1 模式	541
16.4.1	设置和操作状态	541
16.4.2	释放 IDLE1 模式	541
16.5	IDLE2 模式	543
16.5.1	设置和操作状态	543
16.5.2	释放 IDLE2 模式	543
16.5.3	当释放 IDLE2 模式时保证设置时间	545
16.6	STOP 模式	546
16.6.1	设置和操作状态	546
16.6.2	释放 STOP 模式	546
16.6.3	当释放 STOP 模式时保证振荡稳定时间	548

16.7	子时钟操作模式	549
16.7.1	设置和操作状态	549
16.7.2	释放子时钟操作模式	549
16.8	子 IDLE 模式	551
16.8.1	设置和操作状态	551
16.8.2	释放子 IDLE 模式	552
第 17 章	复位功能	554
17.1	概述	554
17.2	检测复位源的寄存器	555
17.3	操作	556
17.3.1	通过 RESET 引脚的复位操作	556
17.3.2	通过看门狗定时器 2 执行的复位操作	558
17.3.3	上电清零电路引起的复位操作	559
17.3.4	低电压检测器引起的复位操作	559
17.3.5	时钟监控器引起的复位操作	559
<R> 17.4	复位释放后的操作	560
第 18 章	时钟监控器	562
18.1	功能	562
18.2	结构	562
18.3	寄存器	563
18.4	操作	564
第 19 章	上电清零电路	567
19.1	功能	567
19.2	配置	567
19.3	操作	568
第 20 章	低电压监测器	569
20.1	功能	569
20.2	配制	569
20.3	寄存器	570
20.4	操作	572
20.4.1	用作内部复位信号	572
20.4.2	用作中断	574
20.5	RAM 保持电压检测操作	575
20.6	仿真功能	576
第 21 章	调节器	577
21.1	概述	577
21.2	操作	578
第 22 章	闪存	579
22.1	特性	579
<R> 22.2	存储器配置	580
<R> 22.3	功能概述	581

	22.4 通过专用闪存编程器重写.....	584
	22.4.1 编程环境	584
	22.4.2 通信模式	585
	22.4.3 闪存控制	590
	22.4.4 通讯模式的选择	591
	22.4.5 通讯命令	592
	22.4.6 引脚连接	593
	22.5 通过自编程重写.....	597
	22.5.1 概述	597
	22.5.2 特点	598
	22.5.3 标准自编程流程	599
	22.5.4 闪存功能	600
	22.5.5 引脚处理	600
	22.5.6 使用的内部资源	601
	第 23 章 选项字节功能	602
<R>	第 24 章 片上调试功能	604
	24.1 使用 DCU 进行调试	605
	24.1.1 电路连接示例	605
	24.1.2 接口信号	605
	24.1.3 屏蔽功能	607
	24.1.4 寄存器.....	607
	24.1.5 操作	609
	24.1.6 注意事项	610
	24.2 不使用 DCU 进行调试	611
	24.2.1 电路连接示例	611
	24.2.2 可屏蔽功能	612
	24.2.3 用户资源的获取	613
	24.2.4 注意事项	619
	24.3 ROM 安全功能.....	620
	24.3.1 安全 ID.....	620
	24.3.2 设置	621
	第 25 章 电气特性.....	623
	25.1 电气特性	623
	25.2 电容	625
	25.3 工件条件	625
	25.4 振荡器特性	626
	25.4.1 主时钟振荡器特性.....	626
	25.4.2 子时钟振荡器特性.....	627
	25.4.3 PLL 特性.....	628
	25.4.4 内部振荡器特性	628
	25.5 电压调节器特性	628
	25.6 DC 特性	629
	25.6.1 I/O 电平	629
	25.6.2 引脚泄漏电流.....	630

25.6.3	电源电流	631
25.7	数据保持特性	632
25.8	AC 特性	633
25.8.1	CLKOUT 输出时序	634
25.9	基本操作	635
25.10	闪存编程特性	642
第 26	封装图	643
<R> 第 27 章	推荐焊接条件	644
<R> 附录 A	开发工具	645
A.1	软件包	647
A.2	语言处理软件	647
A.3	控制软件	647
A.4	调试工具（硬件）	648
A.4.1	使用 IECUBE QB-V850ESFX2 时	648
A.4.2	使用 MINICUBE QB-V850MINI 时	650
A.4.3	使用 MINICUBE2 QB-MINI2 时	651
A.5	调试工具（软件）	652
A.6	嵌入软件	653
A.7	闪存写入工具	654
附录 B	寄存器索引	655
附录 C	指令集列表	661
C.1	常规指令	661
C.2	指令集（按字母顺序）	664
<R> 附录 D	注意事项列表	671
<R> 附录 E	修订记录	698
E.1	在此版本中的主要修订	698

第 1 章 介绍

V850ES/HE2 是日电电子公司（NEC Electronics）的 V850 系列单片微控制器产品之一，主要应用在实时控制应用中的低功耗操作。

1.1 概要

V850ES/HE2 是一款 32 位单片微控制器，它包含了 V850ES CPU 内核以及诸如 ROM/RAM、定时器/计数器、串行接口和 A/D 转换器等外围设备功能。

除了实时响应特性和基本指令单时钟执行特点（1-clock-pitch）之外，V850ES/HE2 还具有由硬件乘法器来实现的乘法指令、饱和运算指令、位操作指令等为数字伺服控制应用而优化的指令。

表 1-1 列出了 V850ES/HE2 系列的所有产品。

表 1-1. V850ES/HE2 产品列表

产品代号		μ PD70F3700	μ PD70F3701
内部存储器	Flash 存储器	64 KB	128 KB
	RAM	6 KB	
内存空间	逻辑空间	64 MB	
通用寄存器		32 位 \times 32 个寄存器	
主时钟（振荡频率）		陶瓷/晶体/外部时钟 <ul style="list-style-type: none"> 在 PLL 模式中：$f_x = 4$ 至 5 MHz 在时钟直连模式中：$f_x = 4$ 至 5 MHz 	
子时钟（振荡频率）		晶体/外部时钟： $f_{XT} = 32.768$ kHz RC 振荡：20 kHz	
内部振荡器		$f_R = 200$ kHz（TYP.）	
最短指令执行时间		50 ns（主时钟（ f_{XX} ）= 20 MHz 运算）	
DSP 功能		$32 \times 32 = 64$: 200 至 250 ns（处于 20MHz） $32 \times 32 + 32 = 32$: 300 ns（处于 20MHz） $16 \times 16 = 32$: 50 至 100 ns（处于 20MHz） $16 \times 16 + 32 = 32$: 150 ns（处于 20MHz）	
I/O 端口		I/O: 51	
定时器		16 位定时器/事件计数器 P: 4 个通道 16 位定时器/事件计数器 Q: 1 个通道 16 位内部定时器 M: 1 个通道 看门狗定时器 2: 1 个通道 钟表定时器: 1 个通道	
A/D 转换器		10 位分辨率 \times 10 个通道	
串行接口		CSIB: 2 个通道 UARTA（对于 LIN 的）: 2 个通道	
中断源		外部: 9 (9) ^注 , 内部: 32	
待机功能		HALT/IDLE1/IDLE2/STOP/子时钟/子 IDLE 模式	
复位		RESET 引脚输入, 看门狗定时器 2 (WDT2), 时钟监视器 (CLM), POC 电路, 低压检测器 (LVI)	
DCU		提供的（运行/中止）	
电源工作电压		3.5 至 5.5 V（A/D 转换器: 4.0 至 5.5 V）	
操作环境温度		-40 至 +85°C	
程序包		64-引脚塑料 LQFP（密间距）（10 \times 10 mm）	

注 圆括号内的数字表示可以释放 STOP 模式的外部中断的次数。

1.2 特性

- 最小指令执行时间: 50 ns (使工作在 32 MHz 主时钟 (f_{xx}) 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 (包含上溢及下溢的检测功能)
 - 32 位移位指令: 1 个时钟周期
 - 位操作指令
 - 区分长短格式的读取/存储指令
- 存储器空间: 为程序和数据提供 64 MB 的线性地址空间
 - 内部存储器:
 - RAM: 6 KB
 - Flash 存储器: 64 KB/128 KB (参照表 1-1)
- 中断及异常:
 - 不可屏蔽中断: 2 个源
 - 可屏蔽中断: 39 个源
 - 软件异常: 32 个源
 - 异常陷阱: 2 个源
- I/O 线: I/O 端口: 51
- 定时器功能:
 - 16 位间隔定时器 M (TMM): 1 个通道
 - 16 位 定时器/事件计数器 P (TMP): 4 个通道
 - 16 位 定时器/事件计数器 Q (TMQ): 1 个通道
 - 钟表定时器: 1 个通道
 - 看门狗定时器 2: 1 个通道
- 串行接口:
 - 异步串行接口 A (UARTA)
 - 三线可变长串行接口 B (CSIB)
 - UARTA (支持 LIN): 2 个通道
 - CSIB: 2 个通道
- A/D 转换器: 10 位精度: 10 个通道
- DCU (调试控制单元): JTAG 接口
- 时钟发生器:
 - 在主时钟或子时钟操作过程中
 - 可选择 7 个级别的 CPU 时钟 (f_{xx}, f_{xx}/2, f_{xx}/4, f_{xx}/8, f_{xx}/16, f_{xx}/32, f_{XT})
 - 时钟直连模式/PLL 模式可选
- 内部振荡时钟: 200 kHz (TYP.)
- 节能模式: HALT/IDLE1/IDLE2/STOP/子时钟/子 IDLE 模式
- 封装: 64 引脚塑封 LQFP (密间距) (10 × 10 mm)

1.3 应用领域

消费型电子设备

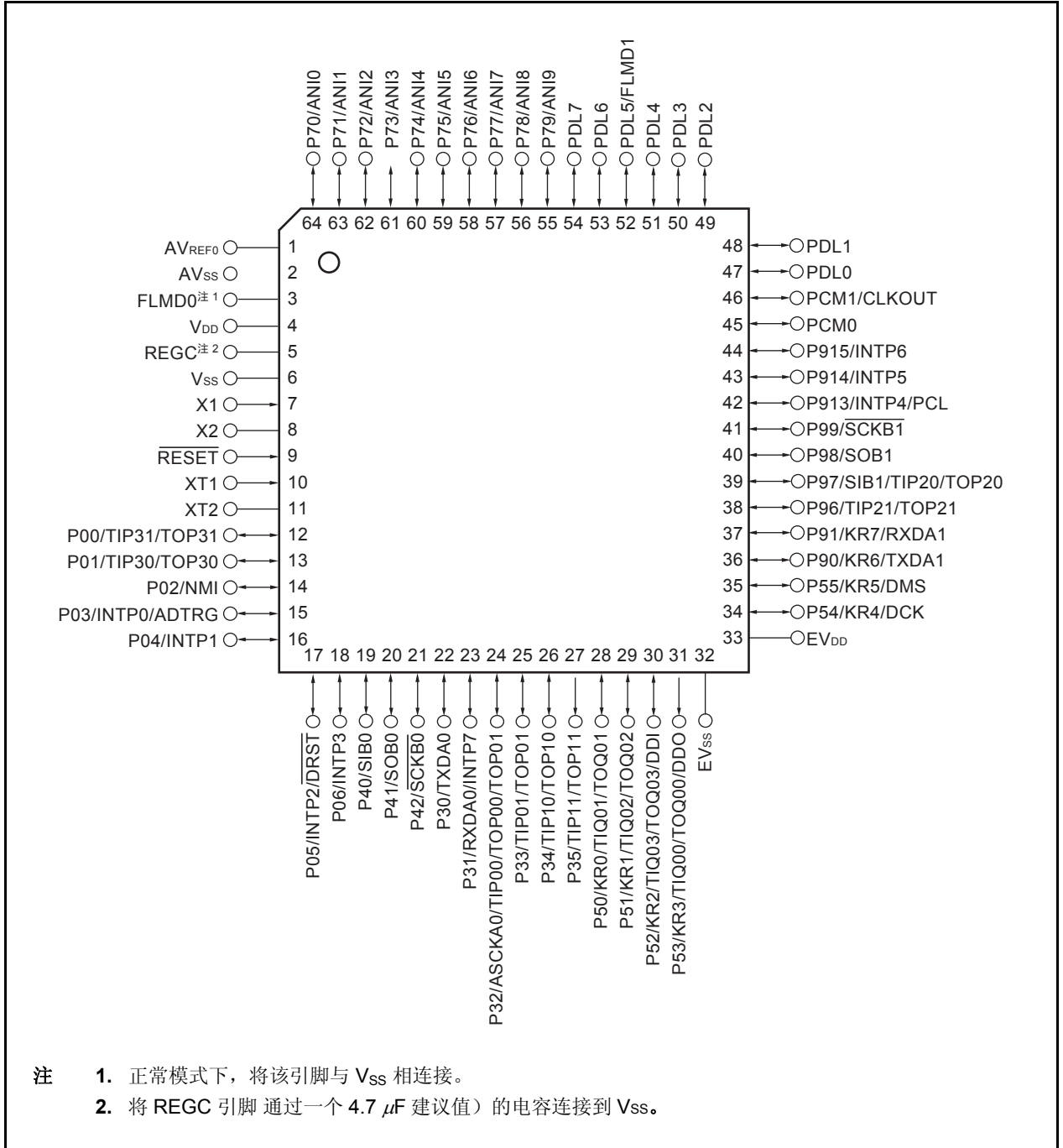
1.4 订货信息

产品代号	封装	片上 Flash 存储器
μ PD70F3700GB-8EU-A	64-引脚塑封 LQFP（密间距）（10 × 10）	64 KB
μ PD70F3701GB-8EU-A	64-引脚塑封 LQFP（密间距）（10 × 10）	128 KB

备注 产品型号的最后带有-A 的产品为无铅产品。

1.5 引脚图（俯视图）

64-引脚塑封 LQFP（密间距）（10 × 10）

 μ PD70F3700GB-8EU-A μ PD70F3701GB-8EU-A

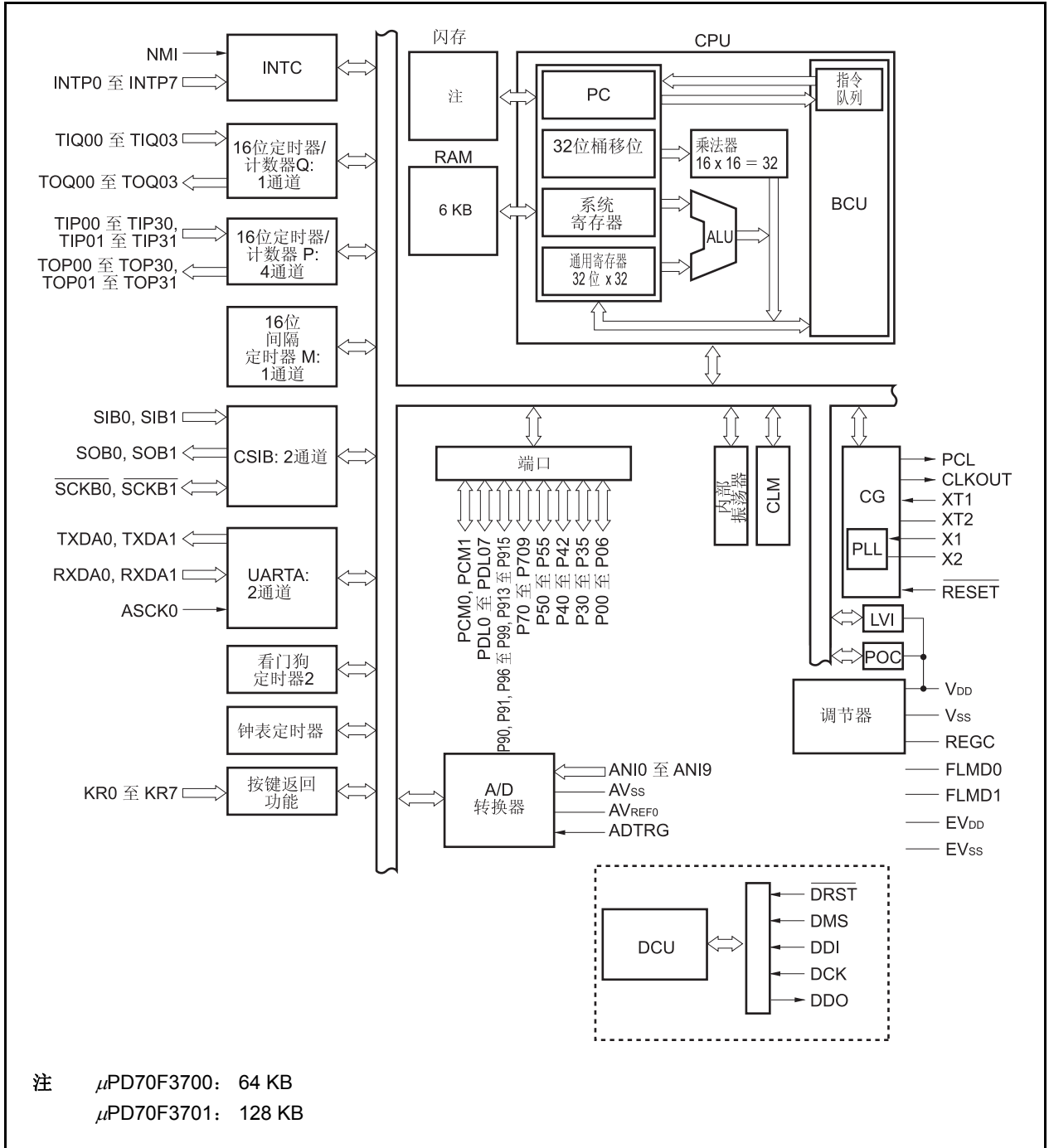
- 注
1. 正常模式下，将该引脚与 V_{SS} 相连接。
 2. 将 REGC 引脚 通过一个 4.7 μ F 建议值) 的电容连接到 V_{SS}。

引脚标识

ADTRG:	A/D 触发信号输入	PCL:	可编程的时钟输出
ANI0 至 ANI9:	模拟输入	PCM0, PCM1:	端口 CM
ASCKA0:	异步串行时钟	PDL0 至 PDL7:	端口 DL
AV _{REF0} :	模拟参考电压	REGC:	整流器控制
AV _{SS} :	模拟 V _{SS}	<u>RESET</u> :	复位
CLKOUT:	时钟输出	RXDA0, RXDA1:	接收数据
DCK:	调试时钟	<u>SCKB0</u> , <u>SCKB1</u> :	串行时钟
DDI:	调试数据输入	SIB0, SIB1:	串行输入
DDO:	调试数据输出	SOB0, SOB1:	串行输出
DMS:	调试模式选择	TIP00, TIP01,	
<u>DRST</u> :	调试复位	TIP10, TIP11,	
EV _{DD} :	端口工作电压	TIP20, TIP21,	
EV _{SS} :	端口地电位	TIP30, TIP31,	
FLMD0, FLMD1:	Flash 编程模式	TIQ00 至 TIQ03:	定时器输入
INTP0 至 INTP7:	外部中断请求	TOP00, TOP01,	
KR0 至 KR7:	按键中断	TOP10, TOP11,	
NMI:	不可屏蔽中断请求	TOP20, TOP21,	
P00 至 P06:	端口 0	TOP30, TOP31,	
P30 至 P35:	端口 3	TOQ00 至 TOQ03:	定时器输出
P40 至 P42:	端口 4	TXDA0, TXDA1:	发送数据
P50 至 P55:	端口 5	V _{DD} :	供电电源引脚
P70 至 P79:	端口 7	V _{SS} :	地电平引脚
P90, P91,		X1, X2:	主时钟晶振
P96 至 P99,		XT1, XT2:	子时钟晶振
P913 至 P915:	端口 9		

1.6 功能模块配置

1.6.1 内部结构框图



1.6.2 内部单元

(1) CPU

CPU 通过使用 5 级流水线控制, 实现了地址计算、算术逻辑运算、数据传输以及其他几乎所有指令的单时钟执行。另外, 乘法器 (16 位 \times 16 位 \rightarrow 32 位) 和桶型移位寄存器 (32 位) 等其它片上专用硬件的集成, 也大大加快了复杂操作处理的速度。

(2) 总线控制单元 (BCU)

BCU 控制了内部总线。

(3) ROM

V850ES/HE2 产品提供 64 KB/128 KB 的片上 Flash 存储器, 映射地址为 0000000H 到 000FFFFH/0000000H 到 001FFFFH。在取指令过程中, 该存储器可由 CPU 在一个时钟周期内访问。

(4) RAM

V850ES/HE2 产品提供 6 KB 的片上 RAM, 映射地址为 3FFD800H 到 3FFEFFFH。在数据访问过程中, 该 RAM 可由 CPU 在一个时钟周期内访问。

(5) 中断控制器 (INTC)

该控制器用来处理由片上周边硬件和外部硬件发出的硬件中断请求 (NMI, INTP0 到 INTP7)。可以对这些中断请求指定 8 个级别的中断优先级, 同时也可进行多重中断服务的处理。

(6) 时钟发生器 (CG)

提供了产生主时钟振荡频率的主时钟振荡器和产生从时钟振荡频率的从时钟振荡器。作为主时钟频率 (f_{xx}), f_x 可以按时钟直连模式和 4 倍频的 PLL 模式使用。

CPU 时钟频率 (f_{CPU}) 可以从下列 7 种形式中选择: f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/8$, $f_{xx}/16$, $f_{xx}/32$ 和 f_{XT} 。

(7) 内部振荡器

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

(8) 定时器/计数器

片上集成了 4 通道 16 位定时器/事件计数器 P (TMP), 单通道 16 位定时器/事件计数器 Q (TMQ) 和单通道 16 位间隔定时器 M (TMM)。

(9) 钟表定时器

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

(10) 看门狗定时器 2

V850ES/SJ3 产品提供一个片上看门狗定时器，该定时器用于检测程序死锁，系统异常等错误状态。

看门狗定时器的时钟源可在内置振荡时钟、主时钟以及子时钟之间自由选择。

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

(11) 串行接口

V850ES/HE2 具有两种形式的串行接口: 异步串行接口 A（UARTA），三线可变量串行接口 B（CSIB）。

使用 UARTA 时，数据通过 TXDAn 到 RXDAn 引脚（n=0, 1）来传输。

使用 CSIB 时，数据通过 SOBn, SIBn 以及 $\overline{\text{SCKBn}}$ 引脚（n=0, 1）来传输。

(12) A/D 转换器

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

(13) 按键中断功能

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

(14) DCU（调试控制单元）

提供了使用 JTAG（联合测试行动小组）通信规范的片上调试功能。正常端口功能与片上调试功能间的转换通过使用控制引脚输入电平以及片上调试模式寄存器（OCDM）来实现。

(15) 端口

提供了以下通用端口功能和控制引脚功能。详细信息请参见第 4 章 端口功能。

第2章 引脚功能

本节说明了 V850ES/HE2 中引脚的名称及功能。

2.1 引脚功能列表

提供了两种类型的引脚 I/O 缓冲式供电电源：AV_{REF0} 和 EV_{DD}。这些供电电源和引脚间的关系如下所示。

表 2-1. 引脚 I/O 缓冲式供电电源

供电电源	对应引脚
AV _{REF0}	端口 7
EV _{DD}	端口 0, 3 至 5, 7, 9, DL, CM, RESET

(1) 端口引脚

表 2-2. 引脚列表（端口引脚）（1/2）

引脚名称	引脚编号	I/O	功能	复用功能
P00	12	I/O	端口 0 7 位 I/O 端口 可以位选输入/输出模式。	TIP31/TOP31
P01	13			TIP30/TOP30
P02	14			NMI
P03	15			INTP0/ADTRG
P04	16			INTP1
P05	17			INTP2/DRST
P06	18			INTP3
P30	22	I/O	端口 3 6 位 I/O 端口 可以位选输入/输出模式。	TXDA0
P31	23			RXDA0/INTP7
P32	24			ASCKA0/TIP00/TOP00/TOP01
P33	25			TIP01/TOP01
P34	26			TIP10/TOP10
P35	27			TIP11/TOP11
P40	19	I/O	端口 4 3 位 I/O 端口 可以位选输入/输出模式。	SIB0
P41	20			SOB0
P42	21			SCKB0
P50	28	I/O	端口 5 6 位 I/O 端口 可以位选输入/输出模式。	KR0/TIQ01/TOQ01
P51	29			KR1/TIQ02/TOQ02
P52	30			KR2/TIQ03/TOQ03/DDI
P53	31			KR3/TIQ00/TOQ00/DDO
P54	34			KR4/DCK
P55	35			KR5/DMS

表 2-2. 引脚列表（端口引脚）（2/2）

引脚名称	引脚编号	I/O	功能	复用功能
P70	64	I/O	端口 7 10 位 I/O 端口 可以位选输入/输出模式。	ANI0
P71	63			ANI1
P72	62			ANI2
P73	61			ANI3
P74	60			ANI4
P75	59			ANI5
P76	58			ANI6
P77	57			ANI7
P78	56			ANI8
P79	55			ANI9
P90	36	I/O	端口 9 9 位 I/O 端口 可以位选输入/输出模式。	KR6/TXDA1
P91	37			KR7/RXDA1
P96	38			TIP21/TOP21
P97	39			SIB1/TIP20/TOP20
P98	40			SOB1
P99	41			SCKB1
P913	42			INTP4/PCL
P914	43			INTP5
P915	44			INTP6
PCM0	45	I/O	端口 CM 2 位 I/O 端口 可以位选输入/输出模式。	—
PCM1	46			CLKOUT
PDL0	47	I/O	端口 DL 8 位 I/O 端口 可以位选输入/输出模式。	—
PDL1	48			
PDL2	49			
PDL3	50			
PDL4	51			
PDL5	52			FLMD1
PDL6	53			—
PDL7	54			

(2) 非端口引脚

表 2-3. 引脚列表（非端口引脚）（1/3）

引脚名称	引脚编号	I/O	功能	复用功能
NMI ^注	14	输入	外部中断输入 外部中断输入（不能屏蔽，模拟噪声消除电路）。	P02
INTP0	15	输入	外部中断请求输入 （可屏蔽，模拟噪声消除电路）	P03/ADTRG
INTP1	16			P04
INTP2	17			P05/ $\overline{\text{DRST}}$
INTP3	18			P06
INTP4	42			P913/PCL
INTP5	43			P914
INTP6	44			P915
INTP7	23			P31/RXDA0
TIP00	24	输入	外部事件/时钟输入（TMP00）	P32/ASCKA0/TOP00/TOP01
TIP01	25		外部事件 输入（TMP01）	P33/TOP01
TIP10	26		外部事件/时钟输入（TMP10）	P34/TOP10
TIP11	27		外部事件 输入（TMP11）	P35/TOP11
TIP20	39		外部事件/时钟输入（TMP20）	P97/SIB1/TOP20
TIP21	38		外部事件 输入（TMP21）	P96/TOP21
TIP30	13		外部事件/时钟输入（TMP30）	P01/TOP30
TIP31	12		外部事件 输入（TMP31）	P00/TOP31
TOP00	24	输出	定时器输出（TMP00）	P32/ASCKA0/TIP00/TOP01
TOP01	24		定时器输出（TMP01）	P32/ASCKA0/TIP00/TOP00
	25			P33/TIP01
TOP10	26		定时器输出（TMP10）	P34/TIP10
TOP11	27		定时器输出（TMP11）	P35/TIP11
TOP20	39		定时器输出（TMP20）	P97/SIB1/TIP20
TOP21	38		定时器输出（TMP21）	P96/TIP21
TOP30	13		定时器输出（TMP30）	P01/TIP30
TOP31	12		定时器输出（TMP31）	P00/TIP31
TIQ00	31	输入	外部事件/时钟输入（TMQ00）	P53/KR3/TOQ00/DDO
TIQ01	28		外部事件输入（TMQ01）	P50/KR0/TOQ01
TIQ02	29		外部事件输入（TMQ02）	P51/KR1/TOQ02
TIQ03	30		外部事件输入（TMQ03）	P52/KR2/TOQ03/DDI
TOQ00	31	输出	定时器输出（TMQ00）	P53/KR3/TIQ00/DDO
TOQ01	28		定时器输出（TMQ01）	P50/KR0/TIQ01
TOQ02	29		定时器输出（TMQ02）	P51/KR1/TIQ02
TOQ03	30		定时器输出（TMQ03）	P52/KR2/TIQ03/DDI

注 引脚 NMI 的复用功能与引脚 P02 相同。复位后，引脚功能为引脚 P02。要使能引脚 NMI，设置 PMC0.PMC02 位为 1。引脚 NMI 的初始设置为“无边沿检测”。通过寄存器 INTF0 和 INTR0 选择引脚 NMI 的有效边沿。

表 2-3. 引脚列表（非端口引脚）（2/3）

引脚名称	引脚编号	I/O	功能	复用功能
SIB0	19	输入	串行接收数据输入（CSIB0）	P40
SIB1	39		串行接收数据输入（CSIB1）	P97/TIP20/TOP20
SOB0	20	输出	串行发送数据输出（CSIB0）	P41
SOB1	40		串行发送数据输出（CSIB1）	P98
$\overline{\text{SCKB0}}$	21	I/O	串行时钟 I/O（CSIB0）	P42
$\overline{\text{SCKB1}}$	41		串行时钟 I/O（CSIB1）	P99
RXDA0	23	输入	串行接收数据输入（UARTA0）	P31/INTP7
RXDA1	37		串行接收数据输入（UARTA1）	P91/KR7
TXDA0	22	输出	串行发送数据输出（UARTA0）	P30
TXDA1	36		串行发送数据输出（UARTA1）	P90/KR6
ASCKA0	24	输入	波特率时钟输入至 UARTA0	P32/TIP00/TOP00/TOP01
ANI0	64	输入	模拟电压输入至 A/D 转换器	P70
ANI1	63			P71
ANI2	62			P72
ANI3	61			P73
ANI4	60			P74
ANI5	59			P75
ANI6	58			P76
ANI7	57			P77
ANI8	56			P78
ANI9	55			P79
AV _{REF0}	1	—	参考电压输入至 A/D 转换器， 提供给复用功能端口 7 的正电源	—
AV _{SS}	2	—	A/D 和 D/A 转换器的地电位（与 V _{SS} 电压相同）	—
ADTRG	15	输入	A/D 转换器外部触发脉冲输入	P03/INTP0
KR0	28	输入	按键中断输入	P50/TIQ01/TOQ01
KR1	29			P51/TIQ02/TOQ02
KR2	30			P52/TIQ03/TOQ03/DDI
KR3	31			P53/TIQ00/TOQ00/DDO
KR4	34			P54/DCK
KR5	35			P55/DMS
KR6	36			P90/TXDA1
KR7	37			P91/RXDA1
DMS	35	输入	调试模式选择	P55/KR5
DDI	30	输入	调试数据输入	P52/KR2/TIQ03/TOQ03
DDO	31	输出	调试数据输出	P53/KR3/TIQ00/TOQ00
DCK	34	输入	调试时钟输入	P54/KR4
$\overline{\text{DRST}}$	17	输入	调试复位输入	P05/INTP2
FLMD0	3	输入	编程模式设置引脚	—
FLMD1	52			PDL5

表 2-3. 引脚列表（非端口引脚）（3/3）

引脚名称	引脚编号	I/O	功能	复用功能
CLKOUT	46	输出	内部系统时钟输出	PCM1
PCL	42	输出	时钟输出（X1 输入时钟和子时钟的时序输出）	P913/INTP4
REGC	5	—	调节器输出稳定电容连接	—
RESET	9	输入	系统复位输入	—
X1	7	输入	主时钟振荡器连接	—
X2	8	—		—
XT1	10	输入		—
XT2	11	—		—
V _{DD}	4	—	内部电路供电电源引脚	—
V _{SS}	6	—	内部电路地电平	—
EV _{DD}	33	—	外部电路供电电源引脚（与 V _{DD} 电压相同）	—
EV _{SS}	32	—	外部电路地电平（与 V _{SS} 电压相同）	—

2.2 引脚功能描述

(1) P00 至 P06 (端口 0) ... 3-状态 I/O

引脚 P00 至 P06 的功能可以被用作 7 位 I/O 端口，其中 7 位 I/O 端口可以被设置成位输入或输出。

除了被用作 I/O 端口的功能外，这些引脚还可以用作 NMI 输入，外部中断请求信号输入，定时器/计数器 I/O，A/D 转换器的外部触发器以及调试复位输入。

该端口可以在端口模式或控制模式中进行位设置。每个引脚的有效沿可以通过 INTR0 和 INTF0 寄存器来指定。

通过使用上拉电阻选项寄存器 0 (PU0)，片上上拉电阻可以被连接到引脚 P00 至 P06 上。

(a) 端口模式

通过使用端口模式寄存器 0 (PM0)，引脚 P00 至 P06 可以在输入或输出模式中进行位设置。

(b) 控制模式

(i) NMI (不可屏蔽中断请求) ... 输入

该引脚输入一个不可屏蔽中断请求信号。

(ii) INTP0 至 INTP3 (外部中断请求) ... 输入

这些引脚输入外部中断请求信号。

(iii) TIP30, TIP31 (定时器输入) ... 输入

这些引脚将一个外部计数时钟输入到定时器 P3 (TMP3)。

(iv) TOP30, TOP31 (定时器输出) ... 输出

这些引脚从定时器 P3 (TMP3) 中输出一个脉冲信号。

(v) ADTRG (A/D trigger 输入) ... 输入

该引脚将一个外部触发信号输入到 A/D 转换器中。它通过使用 A/D 转换模式寄存器 0 (ADA0M0) 来控制。

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

该引脚输入一个调试复位信号，以及一个异步初始化调试控制单元 (DCU) 的负逻辑信号。为了使该信号失效，应复位 DCU 或使 DCU 失效。当没有使用调试功能时使该信号失效。

详细信息请参照第 24 章 片上调试功能。

(2) P30 至 P35 (端口 3) ... 3-状态 I/O

引脚 P30 至 P35 的功能可以被用作 6 位 I/O 端口，其中 6 位 I/O 端口可以被设置成位输入或输出。

除了被用作 I/O 端口的功能外，这些引脚还可以用作外部中断请求信号输入，串行接口 I/O 以及定时器/计数器 I/O。该端口可以在端口模式或控制模式中进行位设置。每个引脚的有效沿可以通过 INTR3L 和 INTF3L 寄存器来指定。

通过使用上拉电阻选项寄存器 3 (PU3)，片上上拉电阻可以被连接到引脚 P30 至 P35 上。

(a) 端口模式

通过使用端口模式寄存器 3 (PM3)，引脚 P30 至 P35 可以在输入或输出模式中进行位设置。

(b) 控制模式**(i) RXDA0 (接收数据) ... 输入**

该引脚输入 UARTA0 的串行接收数据。

(ii) TXDA0 (发送数据) ... 输出

该引脚输出 UARTA0 的串行发送数据。

(iii) ASCKA0 (异步串行时钟) ... 输入

这是 UARTA0 的输入引脚。

(iv) INTP7 (外部中断请求) ... 输入

该引脚输入一个外部中断请求信号。

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

这些是定时器 P0 和 P1 (TMP0 和 TMP1) 的输入引脚。

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

这些是定时器 P0 和 P1 (TMP0 和 TMP1) 的输出引脚。

(3) P40 至 P42 (端口 4) ... 3-状态 I/O

引脚 P40 至 P42 的功能可以被用作 3 位 I/O 端口，其中 3 位 I/O 端口可以被设置成位输入或输出。

除了被用作 I/O 端口的功能外，这些引脚还可以用作串行接口 I/O。该端口可以在端口模式或控制模式中进行位设置。

通过使用上拉电阻选项寄存器 4 (PU4)，片上上拉电阻可以被连接到引脚 P40 至 P42 上。

(a) 端口模式

通过使用端口模式寄存器 4 (PM4)，引脚 P40 至 P42 可以在输入或输出模式中进行位设置。

(b) 控制模式**(i) SIB0 (串行输入) ... 输入**

该引脚输入 CSIB0 的串行接收数据。

(ii) SOB0 (串行输出) ... 输出

该引脚输出 CSIB0 的串行发送数据。

(iii) SCKB0 (串行时钟) ... 3-状态 I/O

该引脚输入/输出 CSIB0 的串行时钟。

(4) P50 至 P55 (端口 5) ... 3-状态 I/O

引脚 P50 至 P55 的功能可以被用作 6 位 I/O 端口，其中 6 位 I/O 端口可以被设置成位输入或输出。

除了被用作 I/O 端口的功能外，这些引脚还可以用作定时器/计数器 I/O，调试功能 I/O 以及按键中断输入。该端口可以在端口模式或控制模式中进行位设置。

通过使用上拉电阻选项寄存器 5 (PU5)，片上上拉电阻可以被连接到引脚 P50 至 P55 上。

(a) 端口模式

通过使用端口模式寄存器 5 (PM5)，引脚 P50 至 P55 可以在输入或输出模式中进行位设置。

(b) 控制模式**(i) KR0 至 KR5 (按键中断) ... 输入**

这些引脚输入一个按键中断。它们的运算可以通过在输入端口模式中使用按键中断模式寄存器 (KRM) 来指定。

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

这些是定时器 Q0 (TMQ0) 的输入引脚。

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

这些是定时器 Q0 (TMQ0) 的输出引脚。

(iv) DDI (调试数据输入) ... 输入

该引脚将调试数据输入到调试控制单元 (DCU) 中。

详细信息请参照第 24 章 片上调试功能。

(v) DDO (调试数据输出) ... 输出

该引脚从 DCU 输出调试数据。

详细信息请参照第 24 章 片上调试功能。

(iv) DCK (调试时钟输入) ... 输入

该引脚将调试时钟输入到 DCU 中。

详细信息请参照第 24 章 片上调试功能。

(vii) DMS (调试模式选择) ... 输入

该引脚选择 DCU 的调试模式。

详细信息请参照第 24 章 片上调试功能。

(5) P70 至 P79 (端口 7) ... 3-状态 I/O

引脚 P70 至 P79 的功能可以被用作 10 位 I/O 端口，其中 10 位 I/O 端口可以被设置成位输入或输出。

除了被用作 I/O 端口的功能外，这些引脚还可以用于控制模式中 A/D 转换器的模拟输入。然而，在使用模拟输入引脚时，应在输入模式中设置该端口。此时，请不要读取端口。

(a) 端口模式

通过使用端口模式寄存器 7L 和 7H (PM7L 和 PM7H)，引脚 P70 至 P79 可以在输入或输出模式中进行位设置。

(b) 控制模式

引脚 P70 至 P79 的复用功能与引脚 ANI0 至 ANI9 相同。

(i) ANI0 至 ANI9 (模拟输入 0 至 9) ... 输入

这些引脚将模拟信号输入到 A/D 转换器中。

(6) P90, P91, P96 至 P99, P913 至 P915 (端口 9) ... 3-状态 I/O

引脚 P90, P91, P96 至 P99 以及 P913 至 P915 的功能可以被用作 9 位 I/O 端口，其中 9 位 I/O 端口可以被设置成位输入或输出。

除了被用作 I/O 端口的功能外，这些引脚还可以用作串行接口 I/O，定时器/计数器 I/O，时钟输出，外部中断请求信号输入以及按键中断输入。该端口可以在端口模式或控制模式中进行位设置。引脚 P90, P91, P96 至 P99 以及 P913 至 P915 的有效沿可以通过 INTR9H 以及 INTF9H 寄存器来指定。

通过使用上拉电阻选项寄存器 9 (PU9)，片上上拉电阻可以被连接到引脚 P90, P91, P96 至 P99 以及 P913 至 P915 上。

(a) 端口模式

通过使用端口模式寄存器 9 (PM9)，引脚 P90, P91, P96 至 P99 以及 P913 至 P915 可以在输入或输出模式中进行位设置。

(b) 控制模式**(i) SIB1 (串行输入) ... 输入**

该引脚输入 CSIB1 的串行接收数据。

(ii) SOB1 (串行输出) ... 输出

该引脚输出 CSIB1 的串行发送数据。

(iii) SCKB1 (串行时钟) ... 3-状态 I/O

该引脚输入/输出 CSIB1 的串行时钟。

(iv) RXDA1 (接收数据) ... 输入

该引脚输入 UARTA1 的串行接收数据。

(v) TXDA1 (发送输出) ... 输出

该引脚输出 UARTA1 的串行发送数据。

(vi) TIP20, TIP21 (定时器输入) ... 输入

这些是定时器 P2 (TMP2) 的输入引脚。

(vii) TOP20, TOP21 (定时器输出) ... 输出

这些是定时器 P2 (TMP2) 的输出引脚。

(viii) PCL (时钟输出) ... 输出

该引脚输出一个时钟。

(ix) INTP4 至 INTP6 (外部中断请求) ... 输入

这些引脚输入一个外部中断请求信号。

(x) KR6, KR7 (按键中断) ... 输入

这些引脚输入一个按键中断。它们的运算可以通过在输入端口模式中使用按键中断模式寄存器 (KRM) 来指定。

(7) PCM0, PCM1 (端口 CM) ... 3-状态 I/O

引脚 PCM0 及 PCM1 的功能可以被用作 2 位 I/O 端口，其中 2 位 I/O 端口可以被设置成位输入或输出。

除了被用作 I/O 端口的功能外，这些引脚还可以用作控制模式中的总线时钟输出。

(a) 端口模式

通过使用端口模式寄存器 CM (PMCM)，引脚 PCM0 及 PCM1 可以在输入或输出模式中进行位设置。

(b) 控制模式**(i) CLKOUT (时钟输出) ... 输出**

该引脚输出一个内部生成的总线时钟。

(8) PDL0 至 PDL7 (端口 DL) ... 3-状态 I/O

引脚 PDL0 至 PDL7 的功能可以被用作 8 位 I/O 端口，其中 8 位 I/O 端口可以被设置成位输入或输出。

当闪存被编程时 PDL5 也可以用作 FLMD1 引脚（当一个高电平被输入到 FLMD0 时）。此时，请确保输入一个低电平到 FLMD1 引脚中。

(a) 端口模式

通过使用端口模式寄存器 DL (PMDL)，引脚 PDL0 至 PDL7 可以在输入或输出模式中进行位设置。

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

$\overline{\text{RESET}}$ 输入是异步输入。当带有固定低电平宽的信号被输入到 $\overline{\text{RESET}}$ 引脚中时，系统将被复位，该操作优先于其他所有操作。

该引脚用于释放待命模式 (HALT, IDLE 或 STOP)，也用于正常初始化/启动。

(10) X1, X2 (主时钟的晶振)

这些引脚用于连接生成系统时钟的谐振器。

(11) XT1, XT2 (子时钟的晶振)

这些引脚用于连接生成子时钟的谐振器。

(12) AVss (模拟地电平)

这是 A/D 转换器以及复用功能端口的地电平引脚。

(13) AVREF0 (模拟参考电压) ... 输入

该引脚为 A/D 转换器以及复用功能端口提供模拟正电源。

它也为 A/D 转换器提供了参考电压。

(14) EVDD (端口供电电源)

该引脚为 I/O 端口和复用功能引脚提供正电源。

(15) EVss (端口地电平)

这是 I/O 端口以及复用功能引脚的地电平引脚。

(16) VDD (供电电源)

该引脚提供了正电源。将所有 VDD 引脚连接到正供电电源上。

(17) Vss (地电平)

这是一个地电平引脚。将所有 Vss 引脚连接到地电平上。

(18) FLMD0 (Flash 编程模式) ... 输入

这是闪存编程模式的信号输入引脚。

在正常操作模式中将该引脚连接到 Vss 上。

(19) REGC (调节器控制) ... 输入

该引脚连接一个调节器的电容。

2.3 I/O 引脚电路类型以及不使用引脚的推荐连接

(1/2)

引脚	引脚编号	I/O 电路类型	推荐连接
P00/TIP31/TOP31	12	5-W	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P01/TIP30/TOP30	13		
P02/NMI	14		
P03/INTP0/ADTRG	15		
P04/INTP1	16		
P05/INTP2/ $\overline{\text{DRST}}$	17	5-AF	输入： 单独连接到 EV _{SS} 输出： 保持开路
P06/INTP3	18	5-W	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P30/TXDA0	22	5-A	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P31/RXDA0/INTP7	23	5-W	
P32/ASCKA0/TIP00/TOP00/TOP01	24		
P33/TIP01/TOP01	25		
P34/TIP10/TOP10	26		
P35/TIP11/TOP11	27		
P40/SIB0	19	5-W	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P41/SOB0	20	5-A	
P42/ $\overline{\text{SCKB0}}$	21	5-W	
P50/KR0/TIQ01/TOQ01	28	5-W	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P51/KR1/TIQ02/TOQ02	29		
P52/KR2/TIQ03/TOQ03/DDI	30		
P53/KR3/TIQ00/TOQ00/DDO	31		
P54/KR4/DCK	34		
P55/KR5/DMS	35		
P70/ANI0 至 P79/ANI9	64 to 55	11-G	输入： 通过一个电阻单独连接到 AV _{REF0} 或 AV _{SS} 输出： 保持开路
P90/KR6/TXDA1	36	5-W	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
P91/KR7/RXDA1	37		
P96/TIP21/TOP21	38		
P97/SIB1/TIP20/TOP20	39		
P98/SOB1	40	5-A	5-W
P99/ $\overline{\text{SCKB1}}$	41		
P913/INTP4/PCL	42		
P914/INTP5	43		
P915/INTP6	44		
PCM0	45	5	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 输出： 保持开路
PCM1/CLKOUT	46		
PDL0 至 PDL4	47 to 51		
PDL5/FLMD1	52		
PDL6, PDL7	53, 54		

引脚	引脚编号	I/O 电路类型	推荐连接
AV _{REF0}	1	—	直接连接到 V _{DD}
AV _{SS}	2	—	—
FLMD0 [‡]	3	—	直接连接到 V _{SS}
REGC	5	—	—
$\overline{\text{RESET}}$	9	2	—
X1	7	—	—
X2	8	—	—
XT1	10	16	通过一个电阻直接连接到 V _{SS}
XT2	11	16	保持开路
V _{DD}	4	—	—
V _{SS}	6	—	—
EV _{DD}	33	—	—
EV _{SS}	32	—	—

注 如果超过噪声消除宽度的噪声在自编程期间被输入到 $\overline{\text{RESET}}$ 引脚中，那么当电容被连接到 FLMD0 引脚时，根据电容充电结束时间可能会进入闪存登入模式。因此，请不要将电容连接到 FLMD0 引脚上。

2.4 I/O 引脚电路

图 2-1. I/O 引脚电路类型 (1/2)

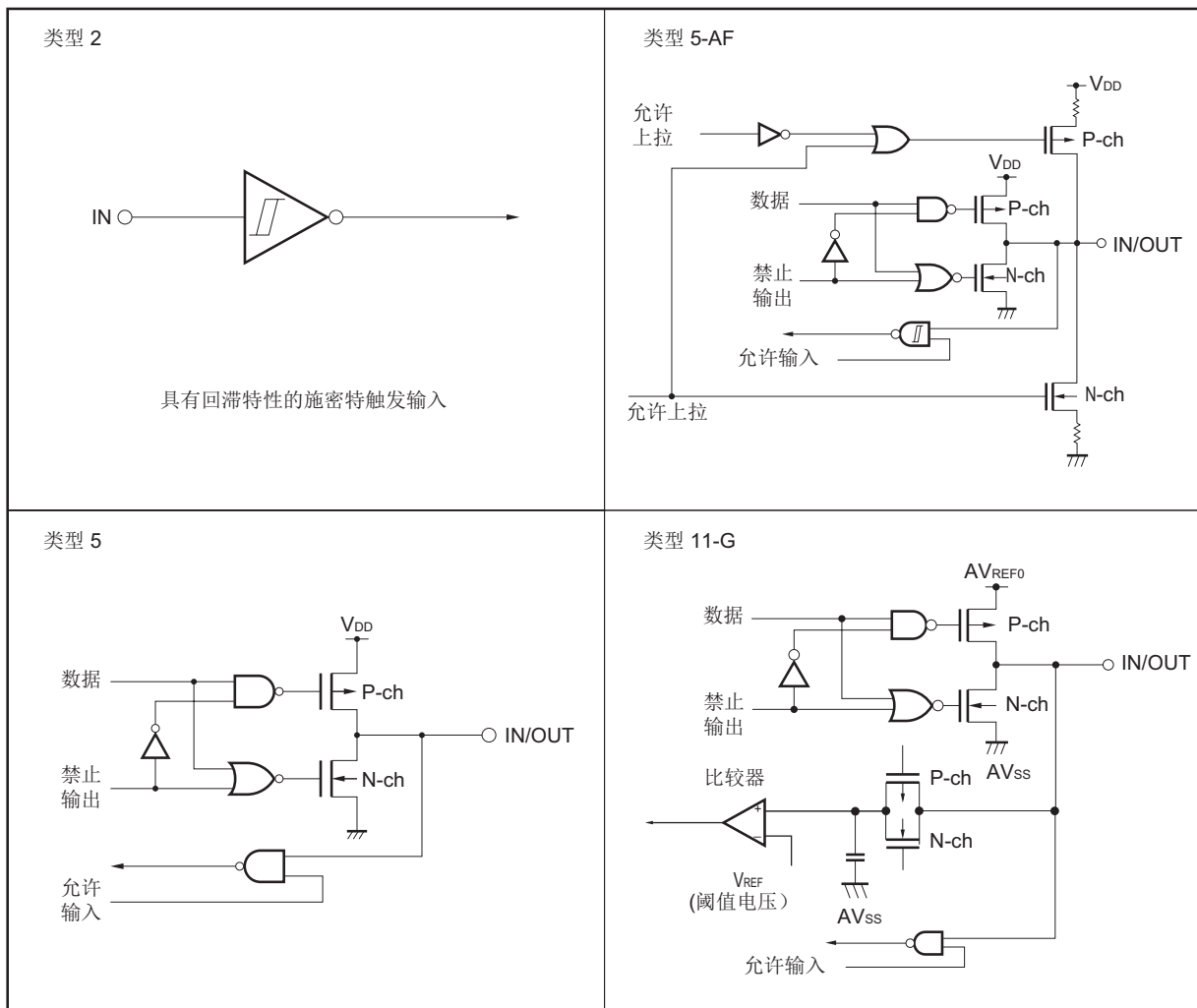
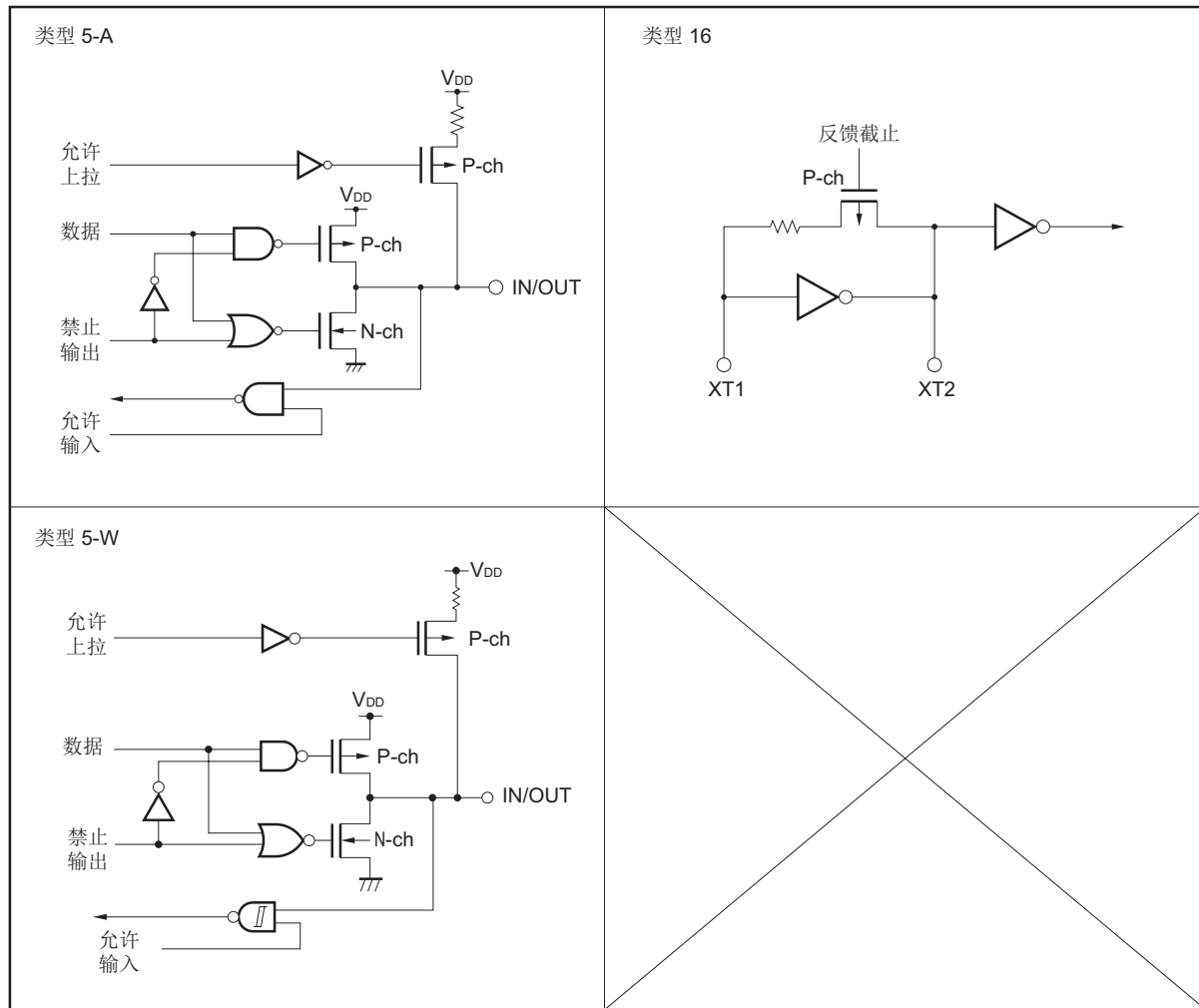


图 2-1. I/O 引脚电路类型 (2/2)



备注 将 V_{DD} 当作 EV_{DD} 来读取。同样地，将 V_{SS} 当作 EV_{SS} 来读取。

<R> 2.5 注意事项

需注意，当打开电源时，下述引脚可能在复位期间临时输出一个不确定的电平。

P53/KR3/TIQ00/TOQ00/DDO 引脚

第3章 CPU 功能

V850ES/HE2 系列产品的 CPU 是基于 RISC 结构设计的，具备 5 级流水线控制，几乎所有指令均可在一个时钟周期内完成。

3.1 特点

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

3.2 CPU 寄存器组

V850ES/HE2 的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。所有的寄存器的宽度均为 32 位。详情请参考 **V850ES** 架构用户手册。

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

3.2.1 程序寄存器组

程序寄存器组包含通用寄存器和程序计数器。

(1) 通用寄存器（r0 到 r31）

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

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

表 3-1. 程序寄存器

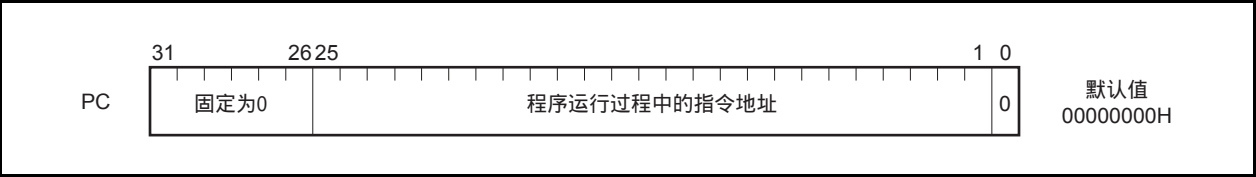
名称	用途	操作
r0	零寄存器	始终保持为零。
r1	汇编器保留寄存器	用于生成 32 位立即数据的工作寄存器
r2	地址/数据寄存器（不被实时操作系统所使用时）	
r3	堆栈指针	当函数被调用时用于生成堆栈结构
r4	全局指针	用于访问数据区域内的全局变量
r5	文本指针	用于表示文本区域（放置程序程序代码的区域）的开始部分的寄存器
r6 至 r29	地址/数据变量寄存器	
r30	元素指针	访问存储器时用于生成地址的基指针
r31	连接指针	用于编译器调用函数时
PC	程序计数器	程序执行时存放指令地址

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

(2) 程序计数器（PC）

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

第 0 位固定为 0，因此，向奇地址的跳转将不会被执行。



3.2.2 系统寄存器组

系统寄存器控制 CPU 的状态并且保存中断信息。

使用载入/存储指令（LDSR 或 STSR）并指定一个系统寄存器编号（如下所示）可以对这些系统寄存器进行读/写操作。

表 3-2. 系统寄存器编号

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

注 1. 由于中断状态保存寄存器只有一组，因此，要实现多重中断需用程序对这组寄存器进行保存。

2. 这些寄存器只能在 DBTRAP 指令和 DBRET 指令之间被访问。

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

备注 √: 允许访问
×: 禁止访问

(1) 中断状态保存寄存器 (EIPC 和 EIPSW)

EIPC 和 EIPSW 用来在可屏蔽中断发生时保存当前的运行状态。

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

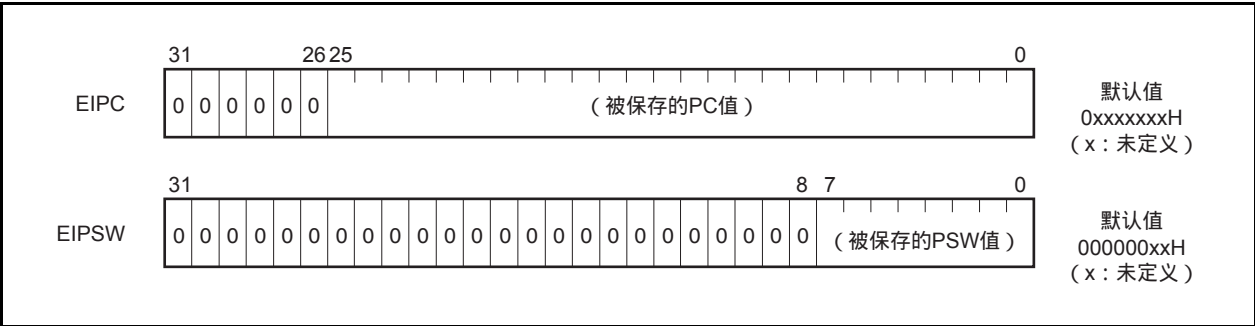
软件异常和可屏蔽中断发生时，当前指令的下一条指令的地址会被保存到 EIPC 中（一些指令除外，详情请参阅 14.8 CPU 不对中断进行确认的时期）。。

当前的 PSW 值被保存在 EIPSW 中。

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

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

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

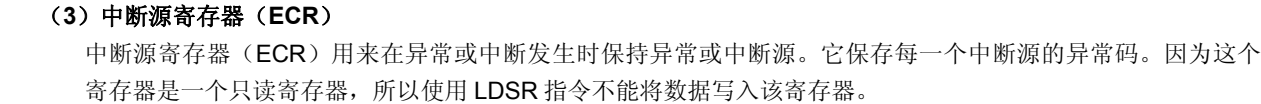


FEPC 和 FEPSW 用来在不可屏蔽中断（NMI）发生时保存当前的运行状态。

不可屏蔽中断发生时，当前指令的下一条指令的地址会被保存到 FEPC 中。

由于中断环境保存寄存器只能对一组中断环境进行保存，因此当多重中断发生时，系统环境需要由程序自行保存。

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



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



(4) 程序状态字 (PSW)

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

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

当用 LDSR 指令对 PSW 进行写操作时, 中断请求响应始终保持无响应状态。

寄存器的第 8 到 31 位为扩展功能保留 (始终为 0)。

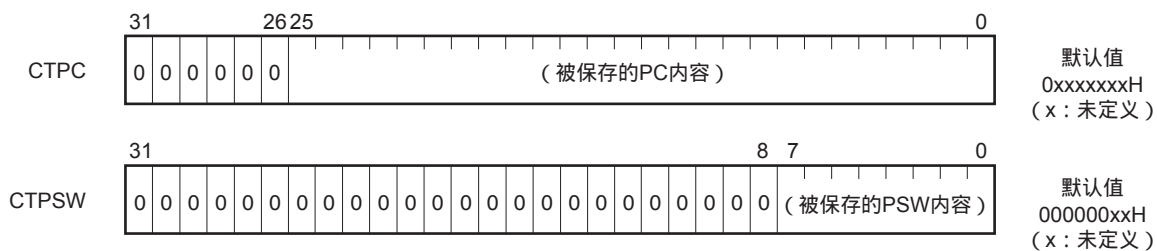
(1/2)

PSW		31	RFU															8	7	6	5	4	3	2	1	0	默认值 00000020H	
																			NP	EP	ID	SAT	CY	OV	S	Z		
位地址	标志名称	含义																										
31 至 8	RFU	保留区域。固定为零。																										
7	NP	表示正在进行非可屏蔽中断 (NMI) 服务。响应 NMI 后该标志被设置, 同时禁止多重中断响应。 0: 未执行 NMI 服务。 1: 正在执行 NMI 服务。																										
6	EP	表示正在进行异常处理。异常产生后该标志被设置。该标志被设置后仍可以对中断请求进行响应。 0: 未执行异常处理。 1: 正在执行异常处理。																										
5	ID	显示可屏蔽中断能否被响应。 0: 允许中断。 1: 禁止中断。																										
4	SAT ^注	显示由于产生了溢出, 饱和运算处理指令的结果是饱和的。因为该标志是累加性标志, 所以运算结果为饱和的饱和运算指令将把该位置 1, 而即使下一条指令的运算结果不为饱和, 该位也不会被清零。向 PSW 中装入数据可将该位清零。注意在普通算术运算过程中, 该位既不会被置 1, 也不会被清零。 0: 不饱和。 1: 饱和。																										
3	CY	运算产生进位或借位时, 该标志会被置 1。 0: 未产生进位或借位。 1: 产生进位或借位。																										
2	OV ^注	运算过程中产生溢出时, 该标志被置 1。 0: 未产生溢出。 1: 产生溢出。																										
1	S ^注	运算结果为负值时, 该标志被置 1。 0: 运算结果为正值或零。 1: 运算结果为负值。																										
0	Z	运算结果为零时, 该标志被设置。 0: 运算结果不为 0。 1: 运算结果为 0。																										

备注 请阅读下页中的注。

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

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



DBPC 和 DBPSW 是异常/调试陷阱状态保存寄存器。

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

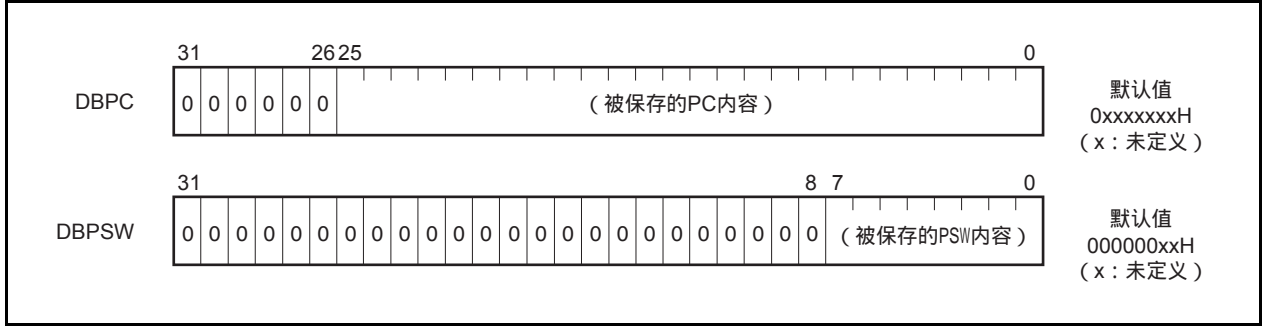
保存在 DBPSW 中的值是当前的 PSW 值。

保存在 DBPSW 中的值是当前的 PSW 值。

该寄存器只有在执行 DBTRAP 指令和 DBRET 指令之间才能够被读写。

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

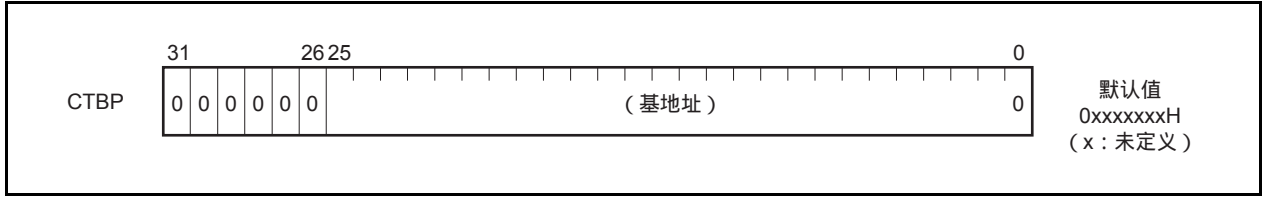
当 DBRETI 指令被执行时，DBPC 和 DBPSW 中的值将被分别恢复到 PC 和 PSW 中。



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

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

第 26 到 31 位为扩展功能保留（始终为 0）。



3.3 操作模式

V850ES/HE2 具有以下几种操作模式。

(1) 正常操作模式

这个模式下，在系统复位被释放后，程序会转至内部 ROM 的复位入口地址，然后指令操作开始执行。

(2) Flash 编程模式

在这个模式下，内部 Flash 可以用 Flash 编程器进行编程。

(3) 片上调试模式

V850ES/HE2 支持 JTAG（联合测试行动小组）格式的在线调试。

详见第 24 章 片上调试功能。

3.3.1 指定操作模式

使用引脚 FLMD0 和 FLMD1 可对器件进行操作模式的指定。

正常模式下，要确保 FLMD0/1 引脚在系统复位释放后为低电平。

Flash 编程模式下，Flash 编程器应该输入 FLMD0 脚高电平信号，但是在自编程模式下这个信号必须通过一个外部电路输入。

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

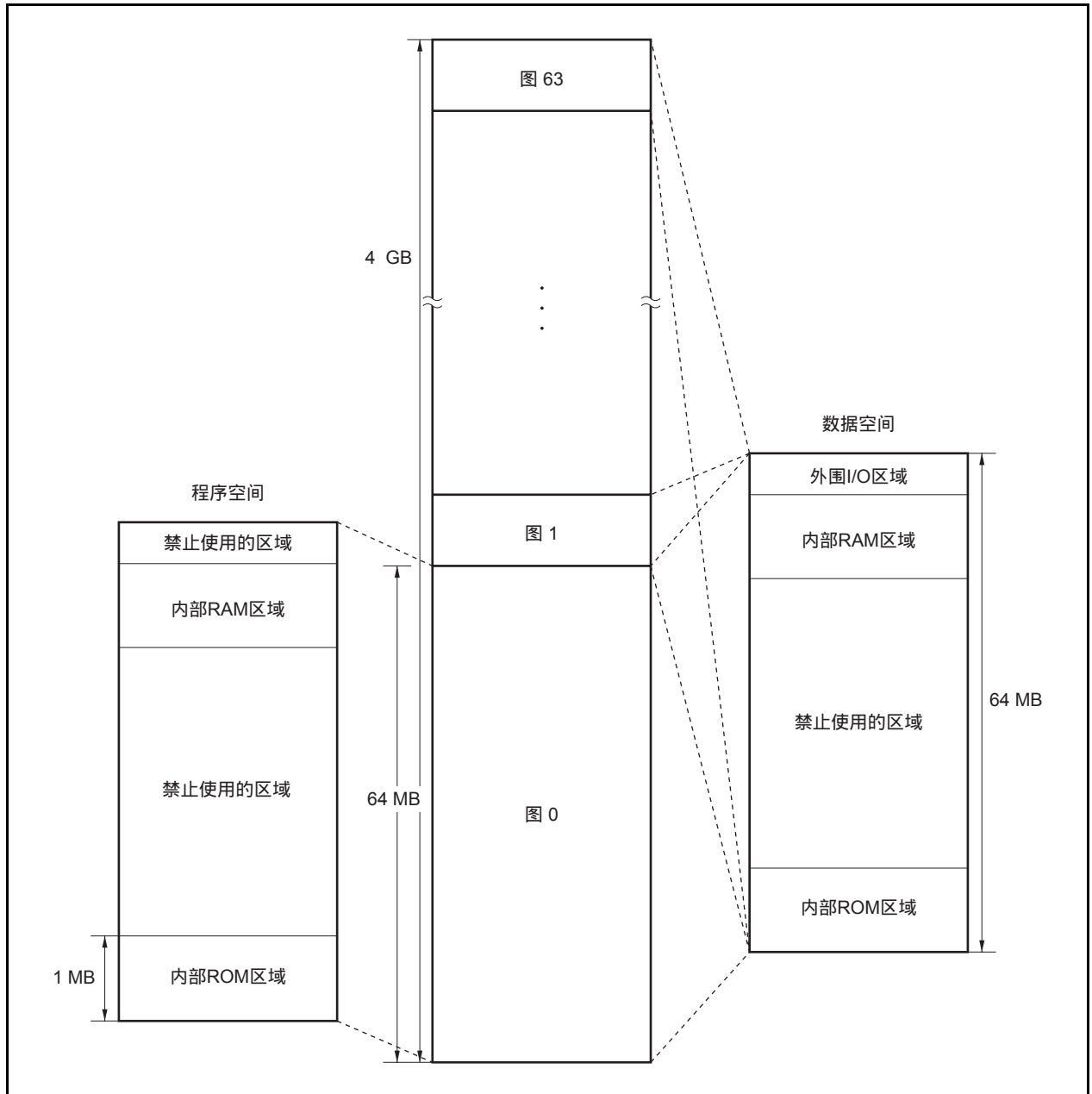
备注 L: 低电平输入
 H: 高电平输入
 ×: 不关心

3.4 地址空间

3.4.1 CPU 地址空间

对于指令地址来说，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 的内置 ROM 区域以及一个内置 RAM 区域。对于操作数地址（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 的地址空间可以被视为 64 个 64MB 的物理地址空间。这就意味着无论地址位的第 26 到 31 位为何值，系统总是在同样的 64MB 的物理地址空间中进行寻址。

图 3-1. 地址空间图

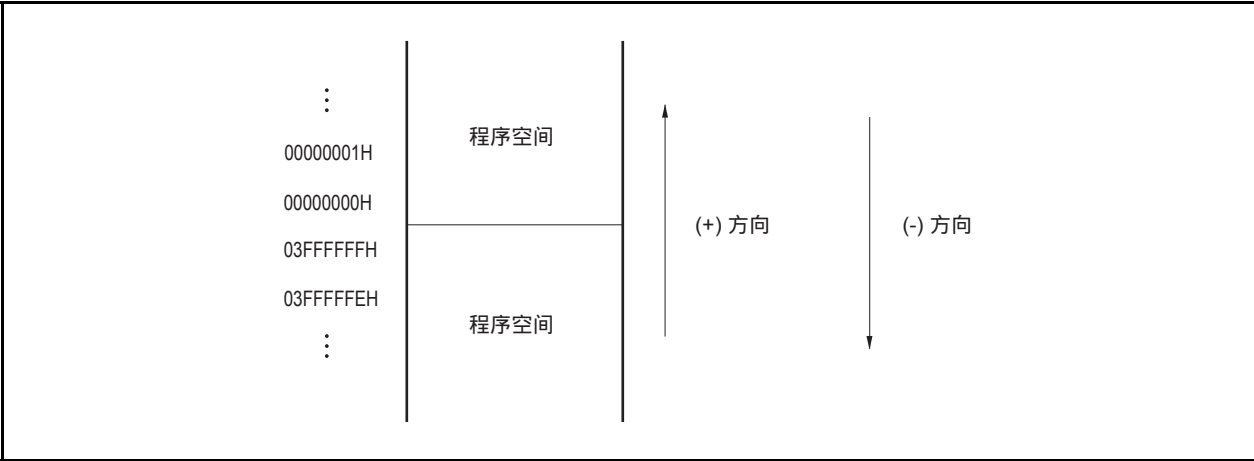


3.4.2 CPU 地址空间的绕回

(1) 程序空间

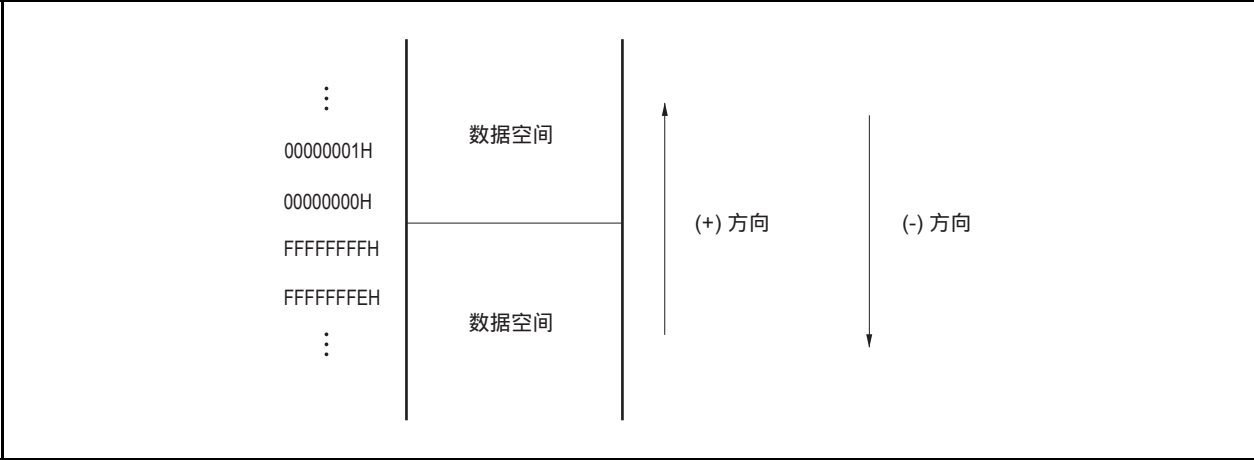
PC（程序计数器）的 32 位中，高 6 位固定为 0，只有低 26 位有效。即使分支地址计算过程中产生第 25 到 26 位的进位或借位，也会被高 6 位忽略。
因此程序空间的地址下限（地址 00000000H）和地址上限（地址 03FFFFFFH）成为连续的地址。绕回所指的就是这种地址下限和地址上限连续的情况。

注意事项 由于从地址 03FFF000H 到 03FFFFFFH 的 4KB 范围是片上外围 I/O 设备区，这个区域不能进行取指令操作。因此，要避免执行可能使分支地址计算结果为此 4KB 中值的操作。



(2) 数据空间

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



3.4.3 存储器映射

V850ES/HE2 存储器的保留区域如下所示。

图 3-2. 数据存储器映射 (物理地址)

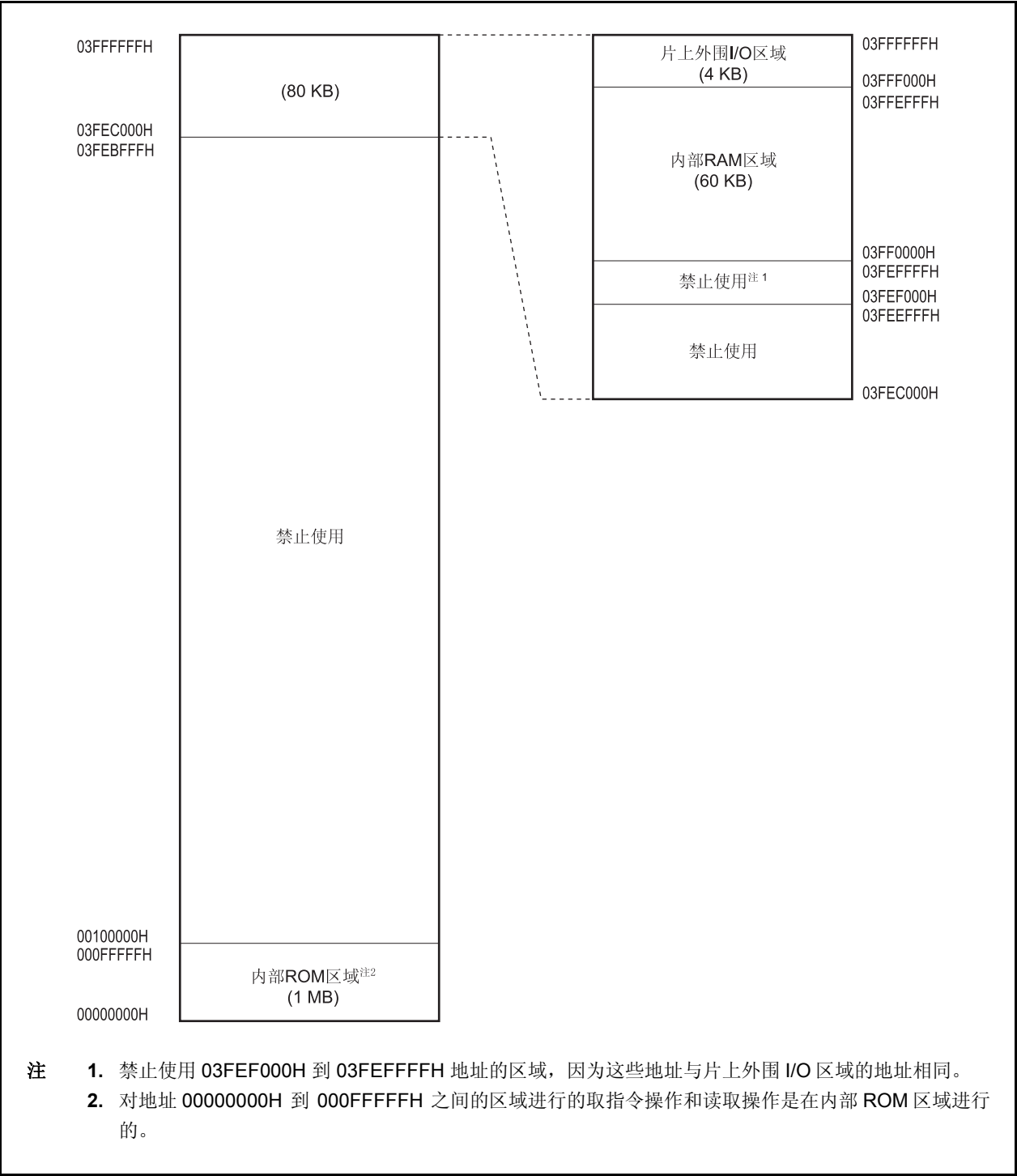
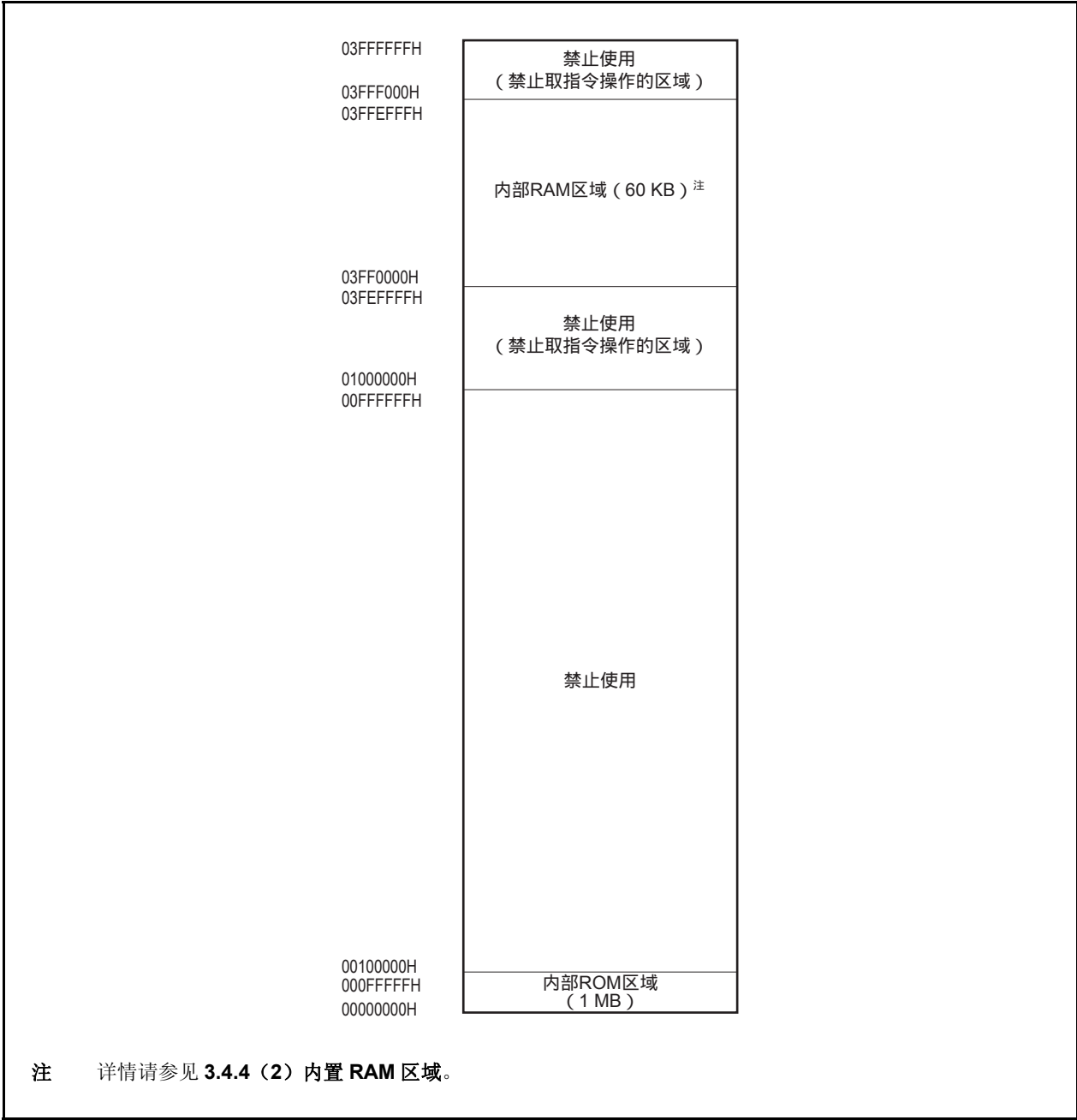


图 3-3. 程序存储器映射



3.4.4 区域

(1) 内部 ROM 区域

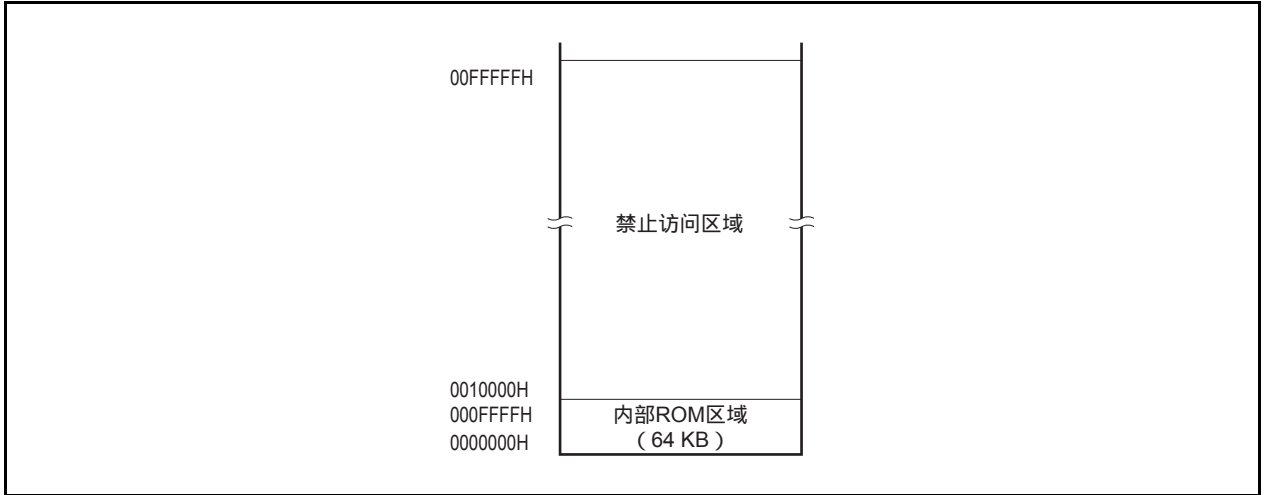
最大为 1MB 的空间被保留为内部 ROM 区域。

(a) 内部 ROM (64 KB)

64KB 的内部 ROM 被分配到 μ PD70F3700 中的地址 0000000H 至 000FFFFH 中。

对地址 0010000H 到 00FFFFFFH 的访问是被禁止的。

图 3-4. 内部 ROM 区域 (64 KB)

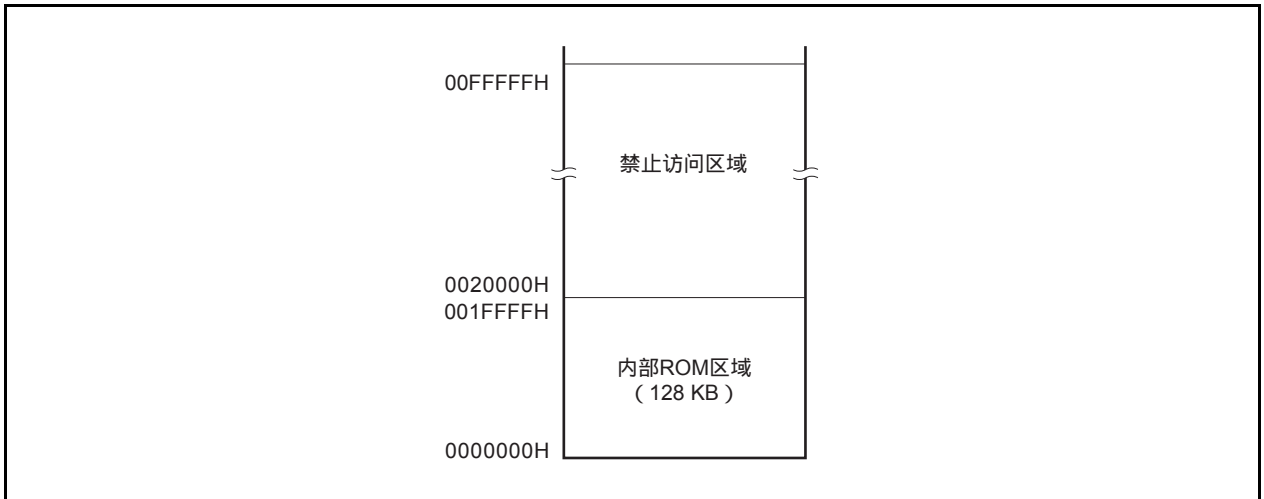


(b) 内部 ROM (128 KB)

128KB 的内部 ROM 被分配到 μ PD70F3701 中的地址 0000000H 至 001FFFFH 中。

对地址 0020000H 到 00FFFFFFH 的访问是被禁止的。

图 3-5. 内部 ROM 区域 (128 KB)



(2) 内部 RAM 区域

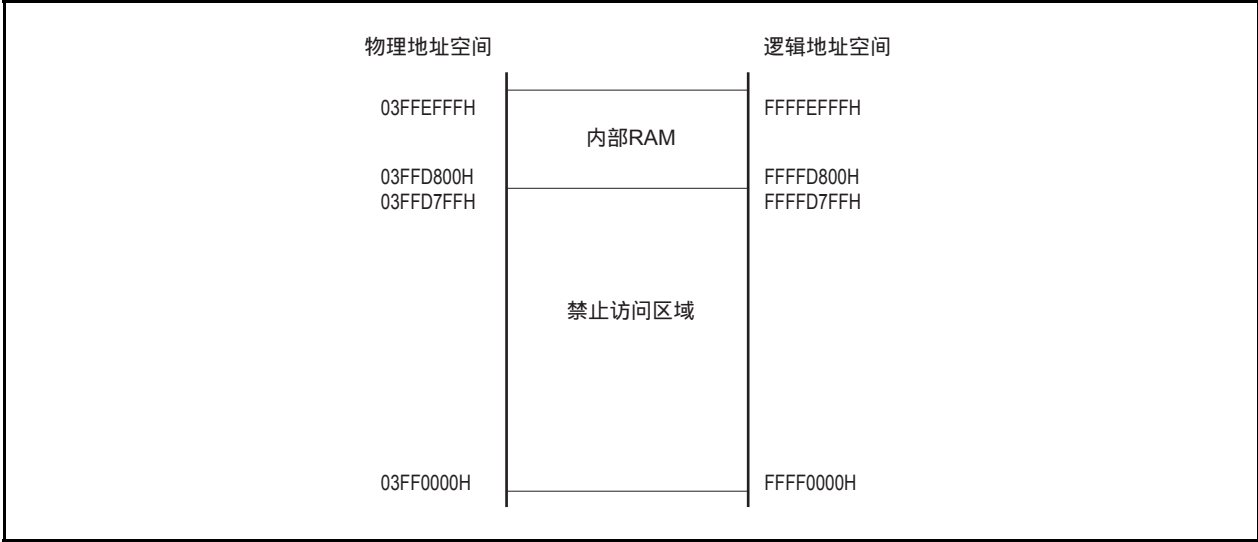
地址 03FF0000H 至 03FFFEFFH 中最大为 60KB 的空间被保留为内部 RAM 区域。

(a) 内部 RAM (6 KB)

6KB 的内部 RAM 被分配到 μ PD70F3700 及 70F3701 中的地址 03FFD800H 至 03FFFEFFH 中。
对地址 03FF0000H 到 03FFD7FFH 的访问是被禁止的。

- μ PD70F3709, 70F3710

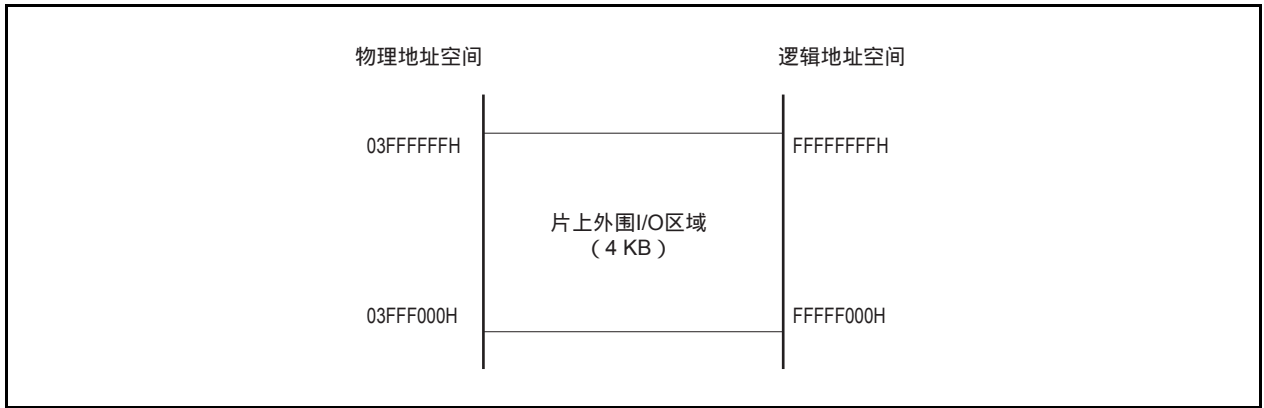
图 3-6. 内部 RAM 区域 (6 KB)



(3) 片上周边 I/O 区域

地址 03FFF000H 至 03FFFFFFH 中 4KB 的空间被保留为片上周边 I/O 区域。

图 3-7. 片上周边 I/O 区域



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

- 注意事项**
1. 当寄存器被一个字操作指令读写时，一个字的空间被分为低 16 位和高 16 位分两次被字节操作先后读写，其中最低 2 位地址被忽略。
 2. 若对只能进行字节操作的寄存器进行半字长(halfword)操作，那么读取时，高 8 位数据无效，写入时，低 8 位数据被写入寄存器。
 3. 未被指定为寄存器的地址为未来扩展保留。对这些地址操作结果的有效性是无法保证的。

3.4.5 地址空间的推荐使用方法

V850ES/HE2 的结构要求在对数据区域中的操作数进行读写时，要确保有一个寄存器可以作为地址指针来使用。这个指针中的地址值的±32KB 的地址范围内的操作数可以被指令直接访问。由于能够用来充当指针的通用寄存器数量有限，当指针值改变引起地址计算时，为了不影响系统的性能，要指定尽可能多的通用寄存器来保存变量值，这样也会减少程序所占的空间。

(1) 程序空间

PC（程序计数器）的 32 位中的高 6 位始终为 0，只有低 26 位有效，因此，从地址 00000000H 开始的 64MB 的连续空间被无条件的映射为程序空间。
要使用内部 RAM 区间作为程序空间，则访问地址 03FFD800H 至 03FFEFFFH（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，那么将不会产生跨越片上外围 I/O 区间的预取址（无效取址）。

(2) 数据空间

在 V850ES/SJ3 系列器件中，4GB 的 CPU 地址空间可被看作 64 个 64MB 的地址空间，因此 26 位地址的最低有效位（bit 25）是作为符号扩展位被扩展至 32 位而被用作地址值的。

(a) 绕回式内存空间的应用举例

如果 R=r0（零寄存器）被用来指定 LD/ST disp16[R]指令，那么 00000000H±32KB 范围的空间都可通过带符号扩展的 16 位偏移量来寻址。包括内部硬件的所有资源都可用一个指针来寻址。
零寄存器（r0）是一个由硬件指定的值固定为零的寄存器，它可以有效地减少指定指针时对寄存器的需求量。

图 3-8. 绕回

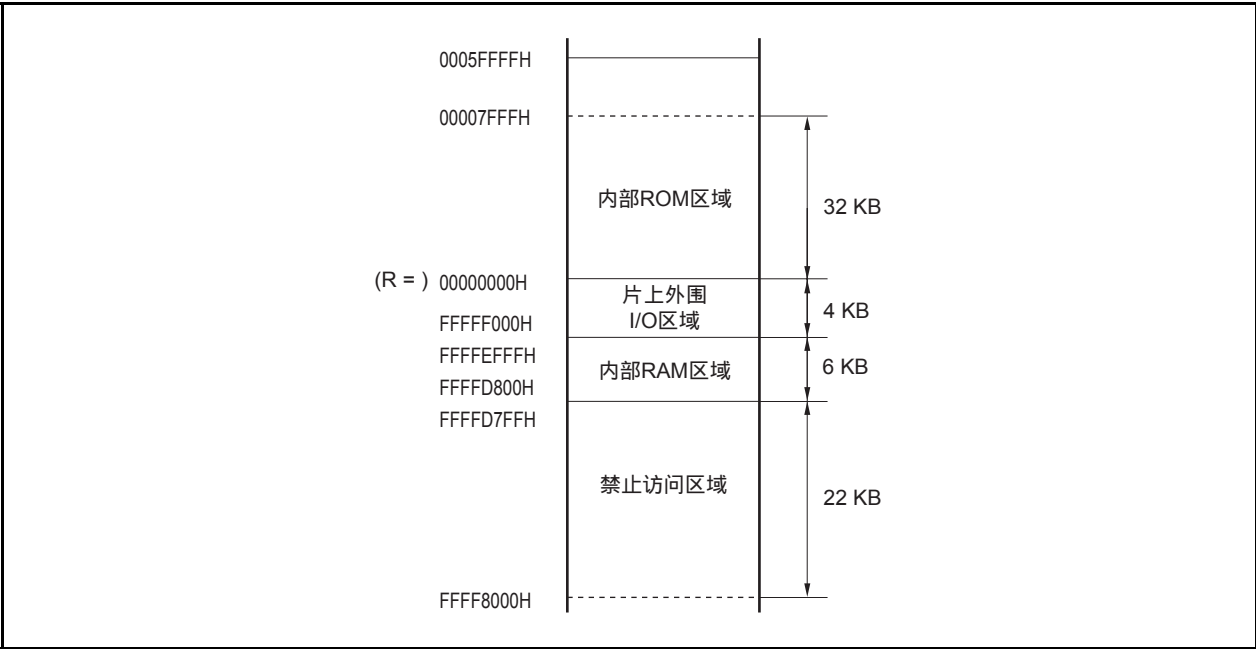
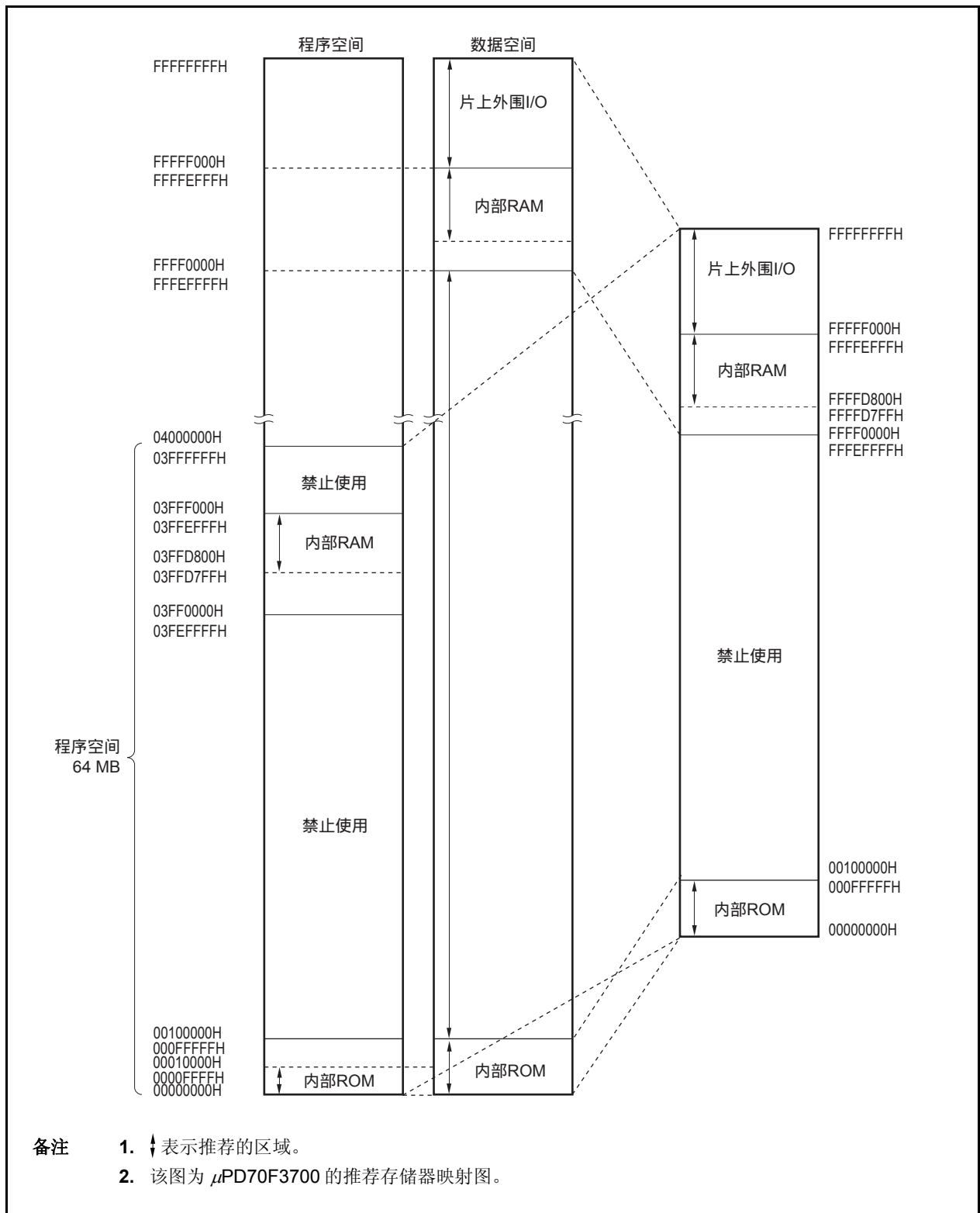


图 3-9. 推荐的存储器映射



3.4.6 周边 I/O 寄存器

(1/7)

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
FFFFF004H	端口 DLL	PDLL	R/W	√	√		未定义
FFFFF00CH	端口 CM	PCM		√	√		未定义
FFFFF024H	端口模式寄存器DLL	PMDLL		√	√		FFH
FFFFF02CH	端口模式寄存器CM	PMCM		√	√		FFH
FFFFF04CH	端口模式控制寄存器CM	PMCCM		√	√		00H
FFFFF06EH	系统等待控制寄存器	VSWC			√		77H
FFFFF100H	中断屏蔽寄存器0	IMR0				√	FFFFH
FFFFF100H	中断屏蔽寄存器0L	IMR0L		√	√		FFH
FFFFF101H	中断屏蔽寄存器0H	IMR0H		√	√		FFH
FFFFF102H	中断屏蔽寄存器1	IMR1				√	FFFFH
FFFFF102H	中断屏蔽寄存器1L	IMR1L		√	√		FFH
FFFFF103H	中断屏蔽寄存器1H	IMR1H		√	√		FFH
FFFFF104H	中断屏蔽寄存器2	IMR2				√	FFFFH
FFFFF104H	中断屏蔽寄存器2L	IMR2L		√	√		FFH
FFFFF105H	中断屏蔽寄存器2H	IMR2H		√	√		FFH
FFFFF110H	中断控制寄存器	LVIIC		√	√		47H
FFFFF112H	中断控制寄存器	PIC0		√	√		47H
FFFFF114H	中断控制寄存器	PIC1		√	√		47H
FFFFF116H	中断控制寄存器	PIC2		√	√		47H
FFFFF118H	中断控制寄存器	PIC3		√	√		47H
FFFFF11AH	中断控制寄存器	PIC4		√	√		47H
FFFFF11CH	中断控制寄存器	PIC5		√	√		47H
FFFFF11EH	中断控制寄存器	PIC6		√	√		47H
FFFFF120H	中断控制寄存器	PIC7		√	√		47H
FFFFF122H	中断控制寄存器	TQ0OVIC		√	√		47H
FFFFF124H	中断控制寄存器	TQ0CCIC0		√	√		47H
FFFFF126H	中断控制寄存器	TQ0CCIC1		√	√		47H
FFFFF128H	中断控制寄存器	TQ0CCIC2		√	√		47H
FFFFF12AH	中断控制寄存器	TQ0CCIC3		√	√		47H
FFFFF12CH	中断控制寄存器	TP0OVIC		√	√		47H
FFFFF12EH	中断控制寄存器	TP0CCIC0		√	√		47H
FFFFF130H	中断控制寄存器	TP0CCIC1		√	√		47H
FFFFF132H	中断控制寄存器	TP1OVIC		√	√		47H
FFFFF134H	中断控制寄存器	TP1CCIC0		√	√		47H
FFFFF136H	中断控制寄存器	TP1CCIC1		√	√		47H
FFFFF138H	中断控制寄存器	TP2OVIC		√	√		47H
FFFFF13AH	中断控制寄存器	TP2CCIC0		√	√		47H
FFFFF13CH	中断控制寄存器	TP2CCIC1		√	√		47H
FFFFF13EH	中断控制寄存器	TP3OVIC		√	√		47H

(2/7)

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
FFFFF140H	中断控制寄存器	TP3CCIC0	R/W	√	√		47H
FFFFF142H	中断控制寄存器	TP3CCIC1		√	√		47H
FFFFF144H	中断控制寄存器	TM0EQIC0		√	√		47H
FFFFF146H	中断控制寄存器	CB0RIC		√	√		47H
FFFFF148H	中断控制寄存器	CB0TIC		√	√		47H
FFFFF14AH	中断控制寄存器	CB1RIC		√	√		47H
FFFFF14CH	中断控制寄存器	CB1TIC		√	√		47H
FFFFF14EH	中断控制寄存器	UA0RIC		√	√		47H
FFFFF150H	中断控制寄存器	UA0TIC		√	√		47H
FFFFF152H	中断控制寄存器	UA1RIC		√	√		47H
FFFFF154H	中断控制寄存器	UA1TIC		√	√		47H
FFFFF156H	中断控制寄存器	ADIC		√	√		47H
FFFFF160H	中断控制寄存器	KRIC		√	√		47H
FFFFF162H	中断控制寄存器	WTIC		√	√		47H
FFFFF164H	中断控制寄存器	WTIC		√	√		47H
FFFFF1FAH	当前服务优先级寄存器	ISPR	R	√	√		00H
FFFFF1FCH	命令寄存器	PRCMD	W		√		未定义
FFFFF1FEH	节能控制寄存器	PSC	R/W	√	√		00H
FFFFF200H	A/D转换器模式寄存器0	ADA0M0		√	√		00H
FFFFF201H	A/D转换器模式寄存器1	ADA0M1		√	√		00H
FFFFF202H	A/D转换器通道指定寄存器 0	ADA0S		√	√		00H
FFFFF203H	A/D转换器模式寄存器2	ADA0M2		√	√		00H
FFFFF204H	掉电比较模式寄存器	ADA0PFM		√	√		00H
FFFFF205H	掉电比较阈值寄存器	ADA0PFT		√	√		00H
FFFFF210H	A/D转换器结果寄存器0	ADA0CR0	R			√	未定义
FFFFF211H	A/D转换器结果寄存器0H	ADA0CR0H			√		未定义
FFFFF212H	A/D转换器结果寄存器1	ADA0CR1				√	未定义
FFFFF213H	A/D转换器结果寄存器1H	ADA0CR1H			√		未定义
FFFFF214H	A/D转换器结果寄存器2	ADA0CR2				√	未定义
FFFFF215H	A/D转换器结果寄存器2H	ADA0CR2H			√		未定义
FFFFF216H	A/D转换器结果寄存器3	ADA0CR3				√	未定义
FFFFF217H	A/D转换器结果寄存器3H	ADA0CR3H			√		未定义
FFFFF218H	A/D转换器结果寄存器4	ADA0CR4				√	未定义
FFFFF219H	A/D转换器结果寄存器4H	ADA0CR4H			√		未定义
FFFFF21AH	A/D转换器结果寄存器5	ADA0CR5				√	未定义
FFFFF21BH	A/D转换器结果寄存器5H	ADA0CR5H			√		未定义
FFFFF21CH	A/D转换器结果寄存器6	ADA0CR6				√	未定义
FFFFF21DH	A/D转换器结果寄存器6H	ADA0CR6H			√		未定义
FFFFF21EH	A/D转换器结果寄存器7	ADA0CR7				√	未定义
FFFFF21FH	A/D转换器结果寄存器7H	ADA0CR7H			√		未定义

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
FFFFF220H	A/D转换器结果寄存器8	ADA0CR8	R			√	未定义
FFFFF221H	A/D转换器结果寄存器8H	ADA0CR8H			√		未定义
FFFFF222H	A/D转换器结果寄存器9	ADA0CR9				√	未定义
FFFFF223H	A/D转换器结果寄存器9H	ADA0CR9H			√		未定义
FFFFF300H	按键中断模式寄存器	KRM	R/W	√	√		00H
FFFFF308H	选择操作控制寄存器0	SELCNT0		√	√		00H
FFFFF318H	噪声消除控制寄存器	NFC		√	√		00H
FFFFF400H	端口 0	P0		√	√		未定义
FFFFF406H	端口 3L	P3L		√	√		未定义
FFFFF408H	端口 4	P4		√	√		未定义
FFFFF40AH	端口 5	P5		√	√		未定义
FFFFF40EH	端口 7L	P7L		√	√		未定义
FFFFF40FH	端口 7H	P7H		√	√		未定义
FFFFF412H	端口 9	P9				√	未定义
FFFFF412H	端口 9L	P9L		√	√		未定义
FFFFF413H	端口 9H	P9H		√	√		未定义
FFFFF420H	端口模式寄存器0	PM0		√	√		FFH
FFFFF426H	端口模式寄存器3L	PM3L		√	√		FFH
FFFFF428H	端口模式寄存器4	PM4		√	√		FFH
FFFFF42AH	端口模式寄存器5	PM5		√	√		FFH
FFFFF42EH	端口模式寄存器7L	PM7L		√	√		FFH
FFFFF42FH	端口模式寄存器7H	PM7H		√	√		FFH
FFFFF432H	端口模式寄存器9	PM9				√	FFFFH
FFFFF432H	端口模式寄存器9L	PM9L		√	√		FFH
FFFFF433H	端口模式寄存器9H	PM9H		√	√		FFH
FFFFF440H	端口模式控制寄存器0	PMC0		√	√		00H
FFFFF446H	端口模式控制寄存器3L	PMC3L		√	√		00H
FFFFF448H	端口模式控制寄存器4	PMC4		√	√		00H
FFFFF44AH	端口模式控制寄存器5	PMC5		√	√		00H
FFFFF452H	端口模式控制寄存器9	PMC9				√	0000H
FFFFF452H	端口模式控制寄存器9L	PMC9L		√	√		00H
FFFFF453H	端口模式控制寄存器9H	PMC9H		√	√		00H
FFFFF460H	端口功能控制寄存器0	PFC0		√	√		00H
FFFFF466H	端口功能控制寄存器3L	PFC3L		√	√		00H
FFFFF46AH	端口功能控制寄存器5	PFC5		√	√		00H
FFFFF472H	端口功能控制寄存器9	PFC9				√	0000H
FFFFF472H	端口功能控制寄存器9L	PFC9L		√	√		00H
FFFFF473H	端口功能控制寄存器9H	PFC9H		√	√		00H
FFFFF540H	TMQ0控制寄存器0	TQ0CTL0		√	√		00H
FFFFF541H	TMQ0控制寄存器1	TQ0CTL1		√	√		00H

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

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
FFFFF5C4H	TMP3 I/O控制寄存器2	TP3IOC2	R/W	√	√		00H
FFFFF5C5H	TMP3选项寄存器0	TP3OPT0		√	√		00H
FFFFF5C6H	TMP3捕捉/比较寄存器0	TP3CCR0				√	0000H
FFFFF5C8H	TMP3捕捉/比较寄存器1	TP3CCR1				√	0000H
FFFFF5CAH	TMP3计数器读取缓冲寄存器	TP3CNT	R			√	0000H
FFFFF680H	钟表定时器操作模式寄存器	WTM	R/W	√	√		00H
FFFFF690H	TMM0控制寄存器0	TM0CTL0		√	√		00H
FFFFF694H	TMM0比较寄存器0	TM0CMP0				√	0000H
FFFFF6C0H	振荡稳定时间选择寄存器	OSTS			√		06H
FFFFF6C1H	PLL 锁止时间指定寄存器	PLLS			√		03H
FFFFF6D0H	看门狗定时器模式寄存器2	WDTM2			√		67H
FFFFF6D1H	看门狗定时器允许寄存器	WDTE			√		9AH
FFFFF706H	端口功能控制扩展寄存器3L	PFCE3L		√	√		00H
FFFFF70AH	端口功能控制扩展寄存器5	PFCE5		√	√		00H
FFFFF712H	端口功能控制扩展寄存器9	PFCE9				√	0000H
FFFFF712H	端口功能控制扩展寄存器9L	PFCE9L		√	√		00H
FFFFF713H	端口功能控制扩展寄存器9H	PFCE9H		√	√		00H
FFFFF802H	系统状态寄存器	SYS		√	√		00H
FFFFF80CH	内部振荡模式寄存器	RCM		√	√		00H
FFFFF820H	节电模式寄存器	PSMR		√	√		00H
FFFFF824H	锁相环锁止状态寄存器	LOCKR	R	√	√		00H
FFFFF828H	处理器时钟控制寄存器	PCC	R/W	√	√		03H
FFFFF82CH	PLL控制寄存器	PLLCTL		√	√		01H
FFFFF82EH	CPU操作时钟状态寄存器	CCLS	R	√	√		00H
FFFFF82FH	可编程时钟模式寄存器	PCLM	R/W	√	√		00H
FFFFF870H	时钟监视器模式寄存器	CLM		√	√		00H
FFFFF888H	复位源标志寄存器	RESF		√	√		00H
FFFFF890H	低电压检测寄存器	LVIM		√	√		00H
FFFFF891H	低电压检测电压选择寄存器	LVIS			√		00H
FFFFF892H	内部RAM 数据状态寄存器	RAMS		√	√		01H
FFFFF8B0H	预分频模式寄存器0	PRSM0		√	√		00H
FFFFF8B1H	预分频器比较寄存器0	PRSCM0			√		00H
FFFFF9FCH	片上调试模式寄存器	OCDM		√	√		01H
FFFFF9FEH	周边仿真寄存器1	PEMU1		√	√		00H
FFFFFA00H	UARTA0控制寄存器0	UA0CTL0		√	√		10H
FFFFFA01H	UARTA0控制寄存器1	UA0CTL1			√		00H
FFFFFA02H	UARTA0控制寄存器2	UA0CTL2			√		FFH
FFFFFA03H	UARTA0选项控制寄存器0	UA0OPT0		√	√		14H
FFFFFA04H	UARTA0状态寄存器	UA0STR		√	√		00H

注意事项 关于 OCDM 寄存器的详细信息，请参照第 24 章 片上调试功能。

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
FFFFFA06H	UARTA0接收数据寄存器	UA0RX	R/W		√		FFH
FFFFFA07H	UARTA0发送数据寄存器	UA0TX			√		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
FFFFFB00H	TIP00引脚噪声消除控制寄存器	P00NFC		√	√		00H
FFFFFB04H	TIP01引脚噪声消除控制寄存器	P01NFC		√	√		00H
FFFFFB08H	TIP10引脚噪声消除控制寄存器	P10NFC		√	√		00H
FFFFFB0CH	TIP11引脚噪声消除控制寄存器	P11NFC		√	√		00H
FFFFFB10H	TIP20引脚噪声消除控制寄存器	P20NFC		√	√		00H
FFFFFB14H	TIP21引脚噪声消除控制寄存器	P21NFC		√	√		00H
FFFFFB18H	TIP30引脚噪声消除控制寄存器	P30NFC		√	√		00H
FFFFFB1CH	TIP31引脚噪声消除控制寄存器	P31NFC		√	√		00H
FFFFFB50H	TIQ00引脚噪声消除控制寄存器	Q00NFC		√	√		00H
FFFFFB54H	TIQ01引脚噪声消除控制寄存器	Q01NFC		√	√		00H
FFFFFB58H	TIQ02引脚噪声消除控制寄存器	Q02NFC		√	√		00H
FFFFFB5CH	TIQ03引脚噪声消除控制寄存器	Q03NFC		√	√		00H
FFFFFC00H	外部中断下降沿指定寄存器 0	INTF0		√	√		00H
FFFFFC06H	外部中断下降沿指定寄存器 3L	INTF3L		√	√		00H
FFFFFC13H	外部中断下降沿指定寄存器 9H	INTF9H		√	√		00H
FFFFFC20H	外部中断上升沿指定寄存器0	INTR0		√	√		00H
FFFFFC26H	外部中断上升沿指定寄存器3L	INTR3L		√	√		00H
FFFFFC33H	外部中断上升沿指定寄存器9H	INTR9H		√	√		00H
FFFFFC40H	上拉电阻选项寄存器0	PU0		√	√		00H
FFFFFC42H	上拉电阻选项寄存器1	PU1		√	√		00H
FFFFFC46H	上拉电阻选项寄存器3L	PU3L		√	√		00H
FFFFFC48H	上拉电阻选项寄存器4	PU4		√	√		00H
FFFFFC4AH	上拉电阻选项寄存器5	PU5		√	√		00H
FFFFFC52H	上拉电阻选项寄存器9	PU9				√	0000H
FFFFFC52H	上拉电阻选项寄存器9L	PU9L		√	√		00H
FFFFFC53H	上拉电阻选项寄存器9H	PU9H		√	√		00H
FFFFFD00H	CSIB0控制寄存器0	CB0CTL0		√	√		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	R/W	√	√		00H

(7/7)

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
FFFFFD04H	CSIB0接收数据寄存器	CB0RX	R			√	0000H
FFFFFD04H	CSIB0接收数据寄存器 L	CB0RXL			√		00H
FFFFFD06H	CSIB0发送数据寄存器	CB0TX	R/W			√	0000H
FFFFFD06H	CSIB0发送数据寄存器 L	CB0TXL			√		00H
FFFFFD10H	CSIB1控制寄存器0	CB1CTL0		√	√		01H
FFFFFD11H	CSIB1控制寄存器1	CB1CTL1		√	√		00H
FFFFFD12H	CSIB1控制寄存器2	CB1CTL2			√		00H
FFFFFD13H	CSIB1状态寄存器	CB1STR		√	√		00H
FFFFFD14H	CSIB1接收数据寄存器	CB1RX	R			√	0000H
FFFFFD14H	CSIB1接收数据寄存器 L	CB1RXL			√		00H
FFFFFD16H	CSIB1发送数据寄存器	CB1TX	R/W			√	0000H
FFFFFD16H	CSIB1发送数据寄存器 L	CB1TXL			√		00H

3.4.7 特殊寄存器

特殊寄存器是为了防止由意外的程序循环所引起的数据非法写入而受到保护的寄存器。V850ES/HE2 中共有以下七种特殊寄存器。

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

此外，器件还提供了一个命令寄存器 (PRCDM) 来拦截对特殊寄存器的非法写操作，这样应用系统就不会因为程序挂起而停止工作了。对特殊寄存器的正常写操作是通过特定的流程实现的，非法的写操作将被记录在系统状态寄存器 (SYS) 中（即使当选项数据（地址：007AH）的读取操作由于噪声，瞬间电压下降等原因而变为非法时，该读取操作也将会被记录）。

<R>

(1) 向特殊寄存器中写入数据

请按照以下流程向特殊寄存器中写入数据。

- <1> 将要向特殊寄存器中写入的数据储存在通用寄存器中。
- <2> 将步骤 <1> 中存储的数据写入 PRCMD 寄存器。
- <3> 向特殊寄存器写入数据（使用以下指令）。
 - 存指令（ST/STT 指令）
 - 位操作指令（SET1/CLR1/NOT1 指令）
- <4>至<8> 插入 NOP 指令（5 条）^注。

[说明示例] 使用 PSC 寄存器时（设置待机模式）

ST.B r11,PSMR[r0] ; PSMR 寄存器调协（IDLE，STOP 模式设置）

- <1> MOV 0x02,r10
 - <2> ST.B r10,PRCMD[r0] ; 写 PRCMD 寄存器
 - <3> ST.B r10,PSC[r0] ; 设置 PSC 寄存器
 - <4> NOP^注 ; 哑指令
 - <5> NOP^注 ; 哑指令
 - <6> NOP^注 ; 哑指令
 - <7> NOP^注 ; 哑指令
 - <8> NOP^注 ; 哑指令
- （下一条指令）

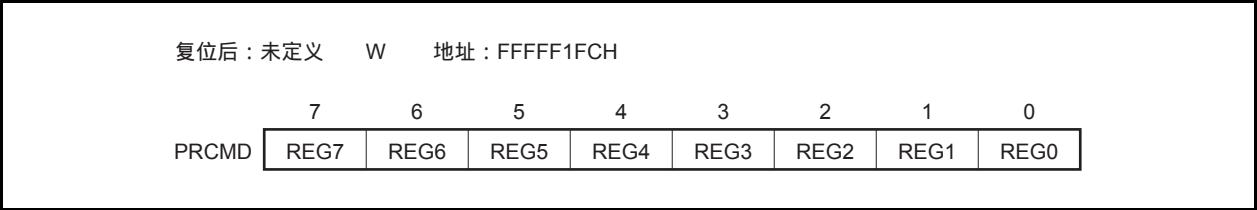
读取特殊寄存器时不需经过特定流程。

注 当转换至 IDLE1，IDLE2，STOP 或子 IDLE 模式时（PSC.STP 位=1），5 条 NOP 指令必须在转换执行后立即被插入。

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

(2) 命令寄存器 (PRCMD)

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



(3) 系统状态寄存器 (SYS)

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

复位后：00H R/W 地址：FFFFF802H

	7	6	5	4	3	2	1	0
SYS	0	0	0	0	0	0	0	PRERR

PRERR	检测保护错误
0	未发生保护错误
1	发生了保护错误

PRERR 标志会在以下情况下改变。

(a) 置位条件 (PRERR 标志 = 1)

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

备注 在写入 PRCMD 寄存器和特殊寄存器的两个操作之间，即使内部 RAM 被访问，例如对片上周边 I/O 寄存器进行读取操作（通过位操作指令读取除外），PRERR 标志位也不会被置 1，数据也可以正常写入特殊寄存器。

(b) 清零条件 (PRERR 标志 = 0)

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

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

3.4.8 注意事项

(1) 需要首先设置的寄存器

在使用 V850ES/HE2 器件时，请务必首先对以下寄存器进行设置。

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

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

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

(a) 系统等待控制寄存器 (VSWC)

VSWC 寄存器用于控制总线访问片上周边 I/O 寄存器的等待时间。

访问一个片上周边 I/O 寄存器需要 3 个时钟周期的时间（不包含等待周期）。V850ES/HE2 需要根据不同的操作频率插入不同的等待周期数。因此，请根据所使用的时钟，将以下对应的数值设置到 VSWC 寄存器中。

可以对 VSWC 寄存器进行 8 位的读写操作。（地址：FFFF06EH，默认值：77H）。

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

(b) 片上调试模式寄存器 (OCDM)

详见 第 24 章 片上调试功能。

(c) 看门狗定时器模式寄存器 2 (WDTM2)

WDTM2 寄存器用于设置看门狗定时器 2 的溢出时间和操作时钟。

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

详见 第 10 章 看门狗定时器 2 的功能。

(2) 访问特定的片上周边 I/O 寄存器

本产品具有两种内部系统总线。

一种是 CPU 总线，另一种是用于与低速周边硬件进行通信的周边总线。

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

对于那些对实时处理要求较高的应用来说，一定要考虑上述的情况。

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

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

周边功能	寄存器名称	访问	k
16 位定时器/事件计数器 P (TMP) (n = 0 至 3)	TPnCNT	读	1 或 2
	TPnCCR0, TPnCCR1	写	<ul style="list-style-type: none"> 第一次访问：无等待 连续写入：3 或 4
		读	1 或 2
16 位定时器/事件计数器 Q (TMQ)	TQ0CNT	读	1 或 2
	TQ0CCR0 至 TQ0CCR3	写	<ul style="list-style-type: none"> 第一次访问：无等待 连续写入：3 或 4
		读	1 或 2
看门狗定时器 2 (WDT2)	WDTM2	写 (在 WDT2 工作时)	3
A/D 转换器	ADA0M0	读	1 或 2
	ADA0CR0 至 ADA0CR9	读	1 或 2
	ADA0CR0H 至 ADA0CR9H	读	1 或 2

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

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

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

备注 i: VSWC 寄存器高 4 位的值 (0 或 1)

j: VSWC 寄存器低 4 位的值 (0 或 1)

(3) sld 指令与中断请求的冲突限制

(a) 描述

如果在下述的种类 <1> 的指令执行完成之前，发生后续的 sld 指令之前的种类 <2> 的指令的解码过程与中断请求的冲突，那么种类 <1> 的指令的执行结果将有可能不被保存在寄存器中。

指令<1>

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

指令<2>

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

<例>

<i> ld.w [r11], r10
:
:
:
<ii> mov r10, r28
<iii> sld.w 0x28, r10

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

(b) 解决方法

- <1> 当使用编译器（CA850）时
- 请使用 CA850 Ver. 2.61 或更高版本，因为这些版本的编译器可以自动避免生成上述情形的指令序列。
- <2> 用汇编程序进行计数测量
- 在指令<ii>之后执行 sld 指令时，可以通过以下的两种方式避免上述的问题。
- 在 sld 指令之前插入一条 nop 指令
 - 在上述的指令<ii>中，不要与 sld 指令使用相同的寄存器

第 4 章 端口功能

4.1 特点

- I/O 端口：51。
- 端口引脚复用功能包含其他的周边 I/O 引脚。
- 可以在输入或输出模式中以 1 位为单位进行设置。

4.2 端口的基本配置

V850ES/HE2 中总共由 51 个 I/O 端口，端口 0，3 至 5，7，9，CM 以及 DL 组成。端口配置情况如下图所示。

图 4-1. 端口配置

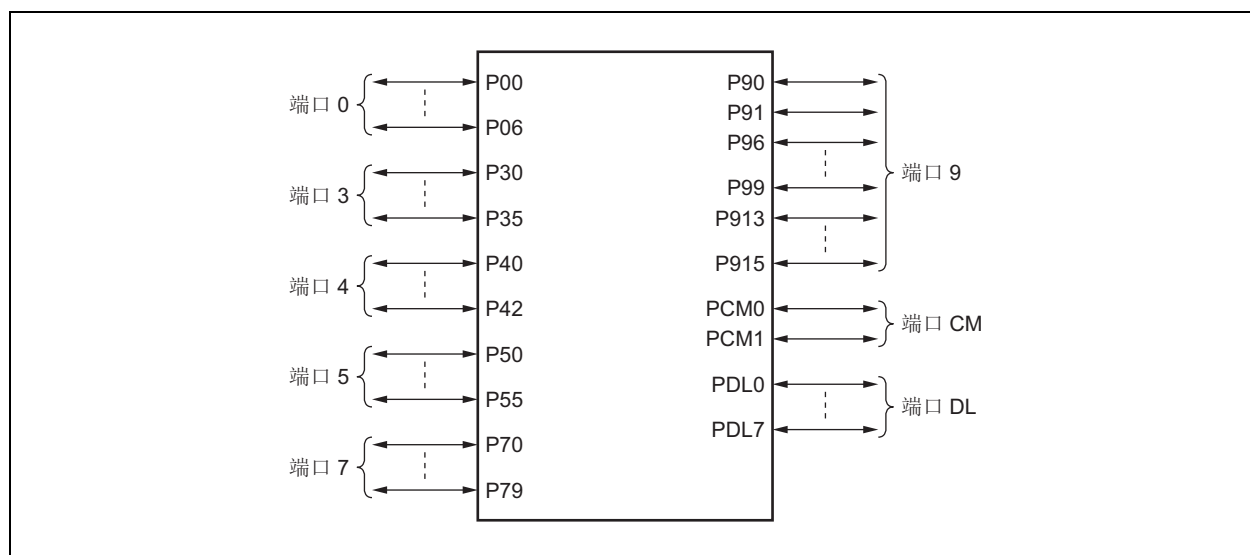


表 4-1. 端口配置

项目	配置
控制寄存器	端口模式寄存器 (PMn: n = 0, 3L, 4, 5, 7L, 7H, 9, CM, 或 DLL)
	端口模式控制寄存器 (PMCn: n = 0, 3L, 4, 5, 9, 或 CM)
	端口功能控制寄存器 (PFCn: n = 0, 3L, 5, 或 9)
	端口功能控制扩展寄存器 (PFCEn: n = 3L, 5, 或 9)
	上拉电阻选项寄存器 (PUn: n = 0, 3L, 4, 5, 或 9)
端口	51

表 4-2. 引脚 I/O 缓冲器电源

电源	对应引脚
AV _{REF0}	端口 7
EV _{DD}	端口 0, 3 至 5, 9, CM, DL, RESET

4.3 端口功能

4.3.1 端口功能的操作

端口的操作根据输入或输出模式中设置的不同而变化，变化情况如下。

(1) 写入 I/O 端口

(a) 在输出模式中

通过使用一个传输指令可以将一个值写入输出锁中。输出锁的内容是从引脚中输出的。一旦数据被写入到输出锁中，它将会被保留直到新的数据被写入到输出锁中。

(b) 在输入模式中

通过使用一个传输指令可以将一个值写入到输出锁中。然而，由于输出缓冲器被关闭，引脚的状态将会保持不变。

一旦数据被写入到输出锁中，它将会被保留直到新的数据被写入到输出锁中。

注意事项 虽然 1 位内存操作指令是操作 1 位的，但它会以 8 位为单位来访问端口。因此，如果一个端口包含了输入引脚及输出引脚的混合物，那么即使引脚不受操作的控制，在输入模式中设置的引脚的输出锁的内容也将会变为未定义的状态。

(2) 从 I/O 端口中读取

(a) 在输出模式中

输出锁的内容可以通过使用一个传输指令来读取。且输出锁的内容是不变的。

(b) 在输入模式中

引脚的状态可以通过使用一个传输指令来读取。且输出锁的内容是不变的。

(3) I/O 端口的操作

(a) 在输出模式中

在输出模式中，操作是在输出锁的内容上进行的，且操作结果会被写入到输出锁中。而输出锁的内容则会从引脚中输出。

一旦数据被写入到输出锁中，它将会被保留直到新的数据被写入到输出锁中。

(b) 在输入模式中

输出锁的内容变为未定义的状态。然而，由于输出缓冲器被关闭，引脚的状态将会保持不变。

注意事项 虽然 1 位内存操作指令是操作 1 位的，但它会以 8 位为单位来访问端口。因此，如果一个端口包含了输入引脚及输出引脚的混合物，那么即使引脚不受操作的控制，在输入模式中设置的引脚的输出锁的内容也将会变为未定义的状态。

4.3.2 关于设置端口引脚的注释

(1) 端口及复用功能的数量根据产品的不同而变化。复位后将会设置与不可用的端口相关的寄存器以及与值相关的复用功能。

(2) 使用以下步骤来设置端口的寄存器。

<1> 设置端口功能控制寄存器 n (PFCn) 和端口功能控制扩展寄存器 n (PFCEn)。

<2> 设置端口模式控制寄存器 n (PMCn)。

<3> 设置外部中断下降沿指定寄存器 n (INTFn) 和外部中断上升沿指定寄存器 n (INTRn)。

如果在设置 PMCn 寄存器后设置 PFCn 以及 PFCEn 寄存器，那么在设置 PFCn 及 PFCEn 寄存器时可能会指定一个意外的周边功能引脚。

(3) PUn 寄存器的 PUnm 位（连接一个片上上拉电阻的位）仅在输入模式（PMn 寄存器的 PMnm 位=1）中有效。在输出模式（PMn 寄存器的 PMnm 位=0）中，片上上拉电阻会通过硬件断开连接。

(4) 引脚电平及端口锁的读取操作通过端口模式寄存器（PMn）来控制。这也会在使用复用功能时被应用。

(5) 施密特触发器（SHMT）输入缓冲器在端口模式中读取时不会当作 SHMT 缓冲器来使用。

4.3.3 端口 0

端口 0 是一个可以以 1 位为单位控制 I/O 设置的 7 位端口（P00 至 P06）。

（1）端口 0 的功能

- 端口的输入/输出数据可以以 1 位为单位进行指定。
通过端口寄存器 0（P0）来指定。
- 端口的输入/输出数据可以以 1 位为单位进行指定。
通过端口模式寄存器 0（PM0）来指定。
- 端口模式或控制模式（复用功能）可以以 1 位为单位进行指定。
通过端口模式控制寄存器 0（PMC0）来指定。
- 控制模式 1 或控制模式 2 可以以 1 位为单位进行指定。
通过端口功能控制寄存器 0（PFC0）来指定。
- 片上上拉电阻可以以 1 位为单位进行连接。
通过上拉电阻选项寄存器 0（PU0）来指定。

端口 0 包含以下复用功能引脚。

表 4-3. 端口 0 的复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	I/O	备注	模块类型
P00	12	TP31/TOP31	I/O	—	G-1
P01	13	TP30/TOP30			G-1
P02	14	NMI ^{※1}			L-1
P03	15	INTP0/ADTRG			N-1
P04	16	INTP1			L-1
P05	17	INTP2/ $\overline{\text{DRST}}$ ^{※2}			AA-1
P06	18	INTP3			L-2

- 注**
1. NMI 引脚的复用功能与 P02 引脚相同。复位后其功能相当于 P02 引脚。
为了允许 NMI 引脚，应将 PMC0.PMC02 位设为 1。NMI 引脚的初始设置是“无边缘检测”。通过使用 INTF0 和 INTR0 寄存器来选择 NMI 引脚的有效沿。
 2. P05 引脚的复用功能是片上调试功能。外部复位后，P05/INTP2/ $\overline{\text{DRST}}$ 引脚将会被初始化为片上调试引脚（ $\overline{\text{DRST}}$ ）。为了将 P05 引脚用作端口引脚而不是片上调试引脚，必须执行以下操作。

<1> 将 OCDM.OCDM0 位（特殊寄存器）清 0。

<2> 将 P05/INTP2/ $\overline{\text{DRST}}$ 引脚固定为低电平直到上述操作被完成。

当片上调试功能没有被使用时，在上述操作被执行前将一个高电平输入 $\overline{\text{DRST}}$ 引脚可能会引起故障（CPU 锁死）。因此，处理 P05 引脚时应特别小心。

当高电平没有被输入到 P05/INTP2/ $\overline{\text{DRST}}$ 引脚时（该引脚固定为低电平时），将没有必要操作 OCDM.OCDM0 位。

由于下拉电阻（30 kΩ TYP.）被连接到 P05/INTP2/ $\overline{\text{DRST}}$ 引脚的缓冲器上，引脚将不必通过一个外部源来固定为低电平。下拉电阻通过将 OCDM0 位清 0 来断开连接。

注意事项 P00 至 P06 引脚在复用功能的输入模式中具有磁滞特性，但在端口模式中则没有磁滞特性。

(2) 寄存器

(a) 端口寄存器 0 (P0)

端口寄存器 0 (P0) 是一个用于控制引脚电平读取操作和输出电平写操作的 8 位寄存器。该寄存器可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

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（输出模式中）（n = 0 至 6）
0	输出 0。
1	输出 1。

- 备注**
1. 输入模式: 当端口 0 (P0) 被读取时, 此时的引脚电平将会被读取。在写操作期间, 写入 P0 中的数据会进行写操作。这不会影响输入引脚。
输出模式: 当端口 0 (P0) 被读取时, P0 的值将会被读取。在写操作期间, 值将会被写入 P0 且写入的值将会立即被输出。
 2. 当 P0 在输入模式中被读取时, 一个未定义的值 (引脚输入电平) 将会在复位后被当作值来读取。
当 P0 在输出模式中被读取时, 00H (输出锁值) 将会被输出。

(b) 端口模式寄存器 0 (PM0)

这是一个用于指定输入或输出模式的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: FFH		R/W		地址: FFFFF420H				
	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	输入/输出模式的控制 (n = 0 至 6)
0	输出模式
1	输入模式

(c) 端口模式控制寄存器 0 (PMC0)

这是一个用于指定端口模式或控制模式的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 00H

R/W

地址: FFFF440H

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

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

PMC05	P05 引脚操作模式的指定
0	I/O 端口
1	INTP2/ $\overline{\text{DRST}}$ 输入

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

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

PMC02	P02 引脚操作模式的指定
0	I/O 端口
1	NMI 输入

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

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

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

(d) 端口功能控制寄存器 0 (PFC0)

这是一个用于指定控制模式 1 或控制模式 2 的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 00H		R/W		地址: FFFFF460H			
	7	6	5	4	3	2	1 0
PFC0	0	0	0	0	PFC03	0	PFC01 PFC00

PFC03	P03 引脚处于控制模式时操作模式的指定
0	INTP0 输入
1	ADTRG 输入

PFC01	P02 引脚处于控制模式时操作模式的指定
0	TIP30 输入
1	TOP30 输出

PFC00	P01 引脚处于控制模式时操作模式的指定
0	TIP31 输入
1	TOP31 输出

(e) 上拉电阻选项寄存器 0 (PU0)

这是一个用于指定片上上拉电阻连接的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

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	片上上拉电阻连接的控制 (n = 0 至 6)
0	未连接
1	已连接

4.3.4 端口 3

端口 3 是一个可以以 1 位为单位控制 I/O 设置的 6 位 端口（P30 至 P35）。

(1) 端口 3 的功能

- 端口的输入/输出数据可以以 1 位为单位进行指定。
通过端口寄存器 3L（P3L）来指定。
- 端口的输入/输出模式可以以 1 位为单位进行指定。
通过端口模式寄存器 3L（PM3L）来指定。
- 端口模式或控制模式（复用功能）可以以 1 位为单位进行指定。
通过端口模式控制寄存器 3L（PMC3L）来指定。
- 控制模式可以以 1 位为单位进行指定。
通过端口功能控制寄存器 3L（PFC3L）和端口功能控制扩展寄存器 3L（PFCE3L）来指定。
- 片上上拉电阻可以以 1 位为单位进行连接。
通过上拉电阻选项寄存器 3L（PU3L）来指定。

端口 3 包含以下复用功能引脚。

表 4-4. 端口 3 的复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	I/O	备注	模块类型
P30	22	TXDA0	I/O	—	E-2
P31	23	RXDA0/INTP7			L-2
P32	24	ASCKA0/TIP00/TOP00/TOP01			U-13
P33	25	TIP01/TOP01			G-1
P34	26	TIP10/TOP10			G-1
P35	27	TIP11/TOP11			G-1

注意事项 P31 至 P35 引脚在复用功能引脚的输入模式中具有磁滞特性，但在端口模式中没有磁滞特性。

(2) 寄存器

(a) 端口寄存器 3L (P3L)

端口寄存器 3L (P3L) 是一个用于控制引脚电平读取操作和输出电平写操作的 8 位寄存器。该寄存器可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 未定义 R/W 地址: FFFFF407H

	7	6	5	4	3	2	1	0
P3L	0	0	P35	P34	P33	P32	P31	P30

P3n	输出数据的控制 (输出模式中) (n = 0 至 5)
0	输出 0。
1	输出 1。

- 备注**
1. 输入模式: 当端口 3L (P3L) 被读取时, 此时的引脚电平将会被读取。在写操作期间, 写入 P3L 中的数据会进行写操作。这不会影响输入引脚。
输出模式: 当端口 3L (P3L) 被读取时, P3L 的值将会被读取。在写操作期间, 值将会被写入 P3L 且写入的值将会立即被输出。
 2. 当 P3L 在输入模式中被读取时, 一个未定义的值 (引脚输入电平) 将会在复位后被当作值来读取。当 P3L 在输出模式中被读取时, 00H (输出锁值) 将会被输出。

(b) 端口模式寄存器 3L (PM3L)

这是一个用于指定输入或输出模式的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: FFH R/W 地址: FFFFF427H

	7	6	5	4	3	2	1	0
PM3L	1	1	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	I/O 模式的控制 (n = 0 至 5)
0	输出模式
1	输入模式

(c) 端口模式控制寄存器 3L (PMC3L)

这是一个用于指定端口模式或控制模式的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 00H		R/W		地址: FFFFF447H			
	7	6	5	4	3	2	1 0
PMC3L	0	0	PMC35	PMC34	PMC33	PMC32	PMC31 PMC30
PMC35		P35 引脚操作模式的指定					
0		I/O 端口					
1		TIP11/TOP11 I/O					
PMC34		P34 引脚操作模式的指定					
0		I/O 端口					
1		TIP10/TOP10 I/O					
PMC33		P33 引脚操作模式的指定					
0		I/O 端口					
1		TIP01/TOP01 I/O					
PMC32		P32 引脚操作模式的指定					
0		I/O 端口					
1		ASCKA0/TIP00/TOP00/TOP01 I/O					
PMC31		P31 引脚操作模式的指定					
0		I/O 端口					
1		RXDA0/INTP7 输入 ^注					
PMC30		P30 引脚操作模式的指定					
0		I/O 端口					
1		TXDA0 输出					

注 INTP7 引脚的功能与 RXDA0 引脚的功能是交互的。要用作 RXDA0 引脚，则使复用功能引脚 INTP7 的边缘检测功能失效（通过将 INTF3L.INTF31 位以及 INTR3L.INTR31 位固定为 0）。要用作 INTP7 引脚，则停止 UARTA0 的接收操作（通过将 UA0CTL0.UA0RXE 位清 0）。

(d) 端口功能控制寄存器 3L (PFC3L)

这是一个用于指定控制模式 1, 2, 3 或 4 的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 00H		R/W		地址: FFFFF466H				
	7	6	5	4	3	2	1	0
PFC3L	0	0	PFC35	PFC34	PFC33	PFC32	0	0

备注 关于如何指定控制模式, 请参见 4.3.4 (2) (f) P3 引脚控制模式的设置。

(e) 端口功能控制扩展寄存器 3L (PFCE3L)

这是一个用于指定控制模式 1, 2, 3 或 4 的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

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

备注 关于如何指定控制模式, 请参见 4.3.4 (2) (f) P3 引脚控制模式的设置。

(f) P3 引脚控制模式的设置

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

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

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

PFCE32	PFC32	P32 引脚控制模式的指定
0	0	ASCKA0 输入
0	1	TOP01 输出
1	0	TIP00 输入
1	1	TOP00 输出

(g) 上拉电阻选项寄存器 3L (PU3L)

这是一个用于指定片上上拉电阻连接的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 00H	R/W	地址: FFFFC47H						
	7	6	5	4	3	2	1	0
PU3L	0	0	PU35	PU34	PU33	PU32	PU31	PU30
PU3n	片上上拉电阻连接的控制 (n = 0 至 5)							
0	未连接							
1	已连接							

4.3.5 端口 4

端口 4 是一个可以以 1 位为单位控制 I/O 设置的 3 位 端口（P40 至 P42）。

（1）端口 4 的功能

- 端口的输入/输出数据可以以 1 位为单位进行指定。
通过端口寄存器 4（P4）来指定。
- 端口的输入/输出模式可以以 1 位为单位进行指定。
通过端口模式寄存器 4（PM4）来指定。
- 端口模式或控制模式（复用功能）可以以 1 位为单位进行指定。
通过端口模式控制寄存器 4（PMC4）来指定。
- 片上上拉电阻可以以 1 位为单位进行连接。
通过上拉电阻选项寄存器 4（PU4）来指定。

端口 4 包含以下复用功能引脚。

表 4-5. 端口 4 的复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	I/O	备注	模块类型
P40	19	SIB0	I/O	—	E-1
P41	20	SOB0			E-2
P42	21	$\overline{\text{SCKB0}}$			E-3

注意事项 P40 至 P42 引脚在复用功能引脚的输入模式中具有磁滞特性，但在端口模式中没有磁滞特性。

(2) 寄存器

(a) 端口寄存器 4 (P4)

端口寄存器 4 (P4) 是一个用于控制引脚电平读取操作和输出电平写操作的 8 位寄存器。该寄存器可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 未定义		R/W		地址: FFFFF408H					
		7	6	5	4	3	2	1	0
P4		0	0	0	0	0	P42	P41	P40

P4n	输出数据的控制（输出模式中）（n = 0 至 2）
0	输出 0。
1	输出 1。

- 备注**
1. 输入模式: 当端口 4 (P4) 被读取时, 此时的引脚电平将会被读取。在写操作期间, 写入 P4 中的数据会进行写操作。这不会影响输入引脚。
输出模式: 当端口 4 (P4) 被读取时, P4 的值将会被读取。在写操作期间, 值将会被写入 P4 且写入的值将会立即被输出。
 2. 当 P4 在输入模式中被读取时, 一个未定义的值 (引脚输入电平) 将会在复位后被当作值来读取。当 P4 在输出模式中被读取时, 00H (输出锁值) 将会被输出。

(b) 端口模式寄存器 4 (PM4)

这是一个用于指定输入或输出模式的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: FFH		R/W		地址: FFFFF428H					
		7	6	5	4	3	2	1	0
PM4		1	1	1	1	1	PM42	PM41	PM40

PM4n	输入/输出模式的控制 (n = 0 至 2)						
0	输出模式						
1	输入模式						

(c) 端口模式控制寄存器 4 (PMC4)

这是一个用于指定端口模式或控制模式的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 00H

R/W

地址: FFFFF448H

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

PMC42	P42 引脚操作模式的指定
0	I/O 端口
1	SCKB0 I/O

PMC41	P41 引脚操作模式的指定
0	I/O 端口
1	SOB0 输出

PMC40	P40 引脚操作模式的指定
0	I/O 端口
1	SIB0 输入

(d) 上拉电阻选项寄存器 4 (PU4)

这是一个用于指定片上上拉电阻连接的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 00H		R/W		地址: FFFFFC48H				
	7	6	5	4	3	2	1	0
PU4	0	0	0	0	0	PU42	PU41	PU40

PU4n	片上上拉电阻连接的控制 (n = 0 至 2)
0	未连接
1	已连接

4.3.6 端口 5

端口 5 是一个可以 1 位为单位控制 I/O 设置的 6 位 端口（P50 至 P55）。

（1）端口 5 的功能

- 端口的输入/输出数据可以以 1 位为单位进行指定。
通过端口寄存器 5（P5）来指定。
- 端口的输入/输出模式可以以 1 位为单位进行指定。
通过端口模式寄存器 5（PM5）来指定。
- 端口模式或控制模式（复用功能）可以以 1 位为单位进行指定。
通过端口模式控制寄存器 5（PMC5）来指定。
- 控制模式可以 1 位为单位进行指定。
通过端口功能控制寄存器 5（PFC5）或端口功能控制扩展寄存器 5（PFCE5）来指定。
- 片上上拉电阻可以以 1 位为单位进行连接。
通过上拉电阻选项寄存器 5（PU5）来指定。

端口 5 包含以下复用功能引脚。

表 4-6. 端口 5 的复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	I/O	备注	模块类型
P50	28	KR0/TIQ01/TOQ01	I/O	—	U-4
P51	29	KR1/TIQ02/TOQ02			U-4
P52	30	KR2/TIQ03/TOQ03/DDI [※]			U-5
P53	31	KR3/TIQ00/TOQ00/DDO [※]			U-6
P54	34	KR4/DCK [※]			G-2
P55	35	KR5/DMS [※]			G-2

注 DDI, DDO, DCK 以及 DMS 引脚用于片上调试功能。要将 DDI, DDO, DCK 以及 DMS 引脚用作端口引脚而不是片上调试引脚，则必须执行以下操作。

- <1> 将 OCDM 寄存器（特殊寄存器）的 OCDM0 位清 0。
- <2> 将 P05/INTP2/ $\overline{\text{DRST}}$ 引脚固定为低电平直到上述操作被执行。

当片上调试功能没有被使用时，在上述操作被执行前将一个高电平输入 $\overline{\text{DRST}}$ 引脚可能会引起故障（CPU 锁死）。因此，处理 P05 引脚时应特别小心。

当高电平没有被输入到 P05/INTP2/ $\overline{\text{DRST}}$ 引脚时（该引脚固定为低电平时），将没有必要操作 OCDM.OCDM0 位。

由于下拉电阻（30 k Ω TYP.）被连接到 P05/INTP2/ $\overline{\text{DRST}}$ 引脚的缓冲器上，引脚将不必通过一个外部源来固定为低电平。下拉电阻通过将 OCDM0 位清 0 来断开连接。

注意事项 P50 至 P55 引脚在复用功能的输入模式中具有磁滞特性，但在端口模式中没有磁滞特性。

(2) 寄存器

(a) 端口寄存器 5 (P5)

端口寄存器 5 (P5) 是一个用于控制引脚电平读取操作和输出电平写操作的 8 位寄存器。该寄存器可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 未定义		R/W	地址: FFFFF40AH					
	7	6	5	4	3	2	1	0
P5	0	0	P55	P54	P53	P52	P51	P50

P5n	输出数据的控制 (输出模式中) (n = 0 至 5)
0	输出 0。
1	输出 1。

- 备注**
1. 输入模式: 当端口 5 (P5) 被读取时, 此时的引脚电平将会被读取。在写操作期间, 写入 P5 中的数据会进行写操作。这不会影响输入引脚。
输出模式: 当端口 5 (P5) 被读取时, P5 的值将会被读取。在写操作期间, 值将会被写入 P5 且写入的值将会立即被输出。
 2. 当 P5 在输入模式中被读取时, 一个未定义的值 (引脚输入电平) 将会在复位后被当作值来读取。
当 P5 在输出模式中被读取时, 00H (输出锁值) 将会被输出。

(b) 端口模式寄存器 5 (PM5)

这是一个用于指定输入或输出模式的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: FFH		R/W	地址: FFFFF42AH					
	7	6	5	4	3	2	1	0
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50

PM5n	I/O 模式的控制 (n = 0 至 5)
0	输出模式
1	输入模式

(c) 端口模式控制寄存器 5 (PMC5)

这是一个用于指定端口模式或控制模式的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

注意事项 如果在 PFC5.PFC5n 以及 PFCE5.PFCE5n 位的值等于默认值 (0) 时通过使用 PMC5 寄存器指定了控制模式，那么输出将会变成未定义的状态。

由于这个原因，应先设置 PFC5.PFC5n 和 PFCE5.PFCE5n 位，然后通过将 PMC5n 位设为 1 来设置控制模式。

复位后: 00H

R/W

地址: FFFFF44AH

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

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

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

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

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

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

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

(d) 端口功能控制寄存器 5 (PFC5)

这是一个用于指定控制模式 1, 2, 3 或 4 的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 00H		R/W	地址: FFFFF46AH					
	7	6	5	4	3	2	1	0
PFC5	0	0	PFC55	PFC54	PFC53	PFC52	PFC51	PFC50

备注 关于如何指定控制模式, 请参见 4.3.6 (2) (f) P5 引脚控制模式的设置。

(e) 端口功能控制扩展寄存器 5 (PFCE5)

这是一个用于指定控制模式 1, 2, 3 或 4 的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 00H		R/W	地址: FFFFF70AH					
	7	6	5	4	3	2	1	0
PFCE5	0	0	0	0	PFCE53	PFCE52	PFCE51	PFCE50

备注 关于如何指定控制模式, 请参见 4.3.6 (2) (f) P5 引脚控制模式的设置。

(f) P5 引脚控制模式的设置

注意事项 如果在 PFC5.PFC5n 以及 PFCE5.PFCE5n 位的值等于默认值 (0) 时通过使用 PMC5 寄存器指定了控制模式, 那么输出将会变成未定义的状态。

由于这个原因, 应先设置 PFC5.PFC5n 和 PFCE5.PFCE5n 位, 然后通过将 PMC5n 位设为 1 来设置控制模式。

PFC55	P55 引脚控制模式的指定
0	禁止设置
1	KR5 输入

PFC54	P54 引脚控制模式的指定
0	禁止设置
1	KR4 输入

PFCE53	PFC53	P53 引脚控制模式的指定
0	0	禁止设置
0	1	TIQ00/KR3 $\overline{\text{TS}}$ 输入
1	0	TOQ00 输出
1	1	禁止设置

PFCE52	PFC52	P52 引脚控制模式的指定
0	0	禁止设置
0	1	TIQ03/KR2 $\overline{\text{TS}}$ 输入
1	0	TOQ03 输出
1	1	禁止设置

PFCE51	PFC51	P51 引脚控制模式的指定
0	0	禁止设置
0	1	TIQ02/KR1 $\overline{\text{TS}}$ 输入
1	0	TOQ02 输出
1	1	禁止设置

PFCE50	PFC50	P50 引脚控制模式的指定
0	0	禁止设置
0	1	TIQ01/KR0 $\overline{\text{TS}}$ 输入
1	0	TOQ01 输出
1	1	禁止设置

注 KRn 引脚的功能与 TIQ0m 引脚的功能是交互的。要将该引脚用作 TIQ0m 引脚，则应使复用功能引脚 KRn 的按键中断检测功能失效（通过将 KRM.KRMn 位清 0）。要将该引脚用作 KRn 引脚，则应使复用功能引脚 TIQ0m 的边缘检测功能失效（n = 0 至 3，m = 0 至 3）。

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

(g) 上拉电阻选项寄存器 5 (PU5)

这是一个用于指定片上上拉电阻连接的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 00H		R/W	地址: FFFFC4AH					
	7	6	5	4	3	2	1	0
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50

PU5n	片上上拉电阻连接的控制 (n = 0 至 5)
0	未连接
1	已连接

4.3.7 端口 7

端口 7 是一个可以以 1 位为单位控制 I/O 设置的 16 位端口（P70 至 P79）。

(1) 端口 7 的功能

- 端口的输入/输出数据可以以 1 位为单位进行指定。
通过端口寄存器 7H, 7L（P7H, P7L）来指定。
- 端口的输入/输出模式可以以 1 位为单位进行指定。
通过端口模式寄存器 7H, 7L（PM7H, PM7L）来指定。

端口 7 包含以下复用功能引脚。

表 4-7. 端口 7 的复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	I/O	备注	模块类型
P70	64	ANI0	I/O	—	A-1
P71	63	ANI1			A-1
P72	62	ANI2			A-1
P73	61	ANI3			A-1
P74	60	ANI4			A-1
P75	59	ANI5			A-1
P76	58	ANI6			A-1
P77	57	ANI7			A-1
P78	56	ANI8			A-1
P79	55	ANI9			A-1

(2) 寄存器

(a) 端口寄存器 7H, 端口寄存器 7L (P7H, P7L)

端口寄存器 7H 和 7L (P7H 和 P7L) 是用于控制引脚电平读取操作和输出电平写操作的 8 位寄存器。该寄存器可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

它们不能以 16 位为单位进行访问。

复位后: 未定义 R/W 地址: FFFFF40FH, FFFFF40EH

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

P7n	输出数据的控制 (输出模式中) (n = 0 至 9)
0	输出 0。
1	输出 1。

注意事项 在 A/D 转换期间不要读取 P7H 和 P7L 寄存器。

- 备注**
1. 输入模式: 当端口 7H 和 7L (P7H 和 P7L) 被读取时, 此时的引脚电平将会被读取。在写操作期间, 写入 P5 中的数据会进行写操作。这不会影响输入引脚。
输出模式: 当端口 7H 和 7L (P7H 和 P7L) 被读取时, 7H 和 7L (P7H 和 P7L) 的值将会被读取。在写操作期间, 值将会被写入 7H 和 7L (P7H 和 P7L) 且写入的值将会立即被输出。
 2. 当 7H 和 7L (P7H 和 P7L) 在输入模式中被读取时, 一个未定义的值 (引脚输入电平) 将会在复位后被当作值来读取。当 7H 和 7L (P7H 和 P7L) 在输出模式中被读取时, 00H (输出锁值) 将会被输出。

(b) 端口模式寄存器 7H, 7L (PM7H, PM7L)

这是用于指定输入或输出模式的 8 位寄存器。它们可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。这两种寄存器不能以 16 位为单位进行访问。

复位后: FFH		R/W		地址: FFFFF42FH, FFFFF42EH				
	7	6	5	4	3	2	1	0
PM7H	1	1	1	1	1	1	PM79	PM78
	7	6	5	4	3	2	1	0
PM7L	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70
PM7n		I/O 模式的控制 (n = 0 至 9)						
0		输出模式						
1		输入模式						

注意事项 要使用 P7n (ANIn) 的复用功能, 则应将 PM7n 设置为 1。

4.3.8 端口 9

端口 9（P90，P91，P96 至 P99，P913 至 P915）是一个可以以 1 位为单位控制 I/O 设置的 9 位端口。

（1）端口 9 的功能

- 端口的输入/输出数据可以以 1 位为单位进行指定。
通过端口寄存器 9（P9）来指定。
- 端口的输入/输出模式可以以 1 位为单位进行指定。
通过端口模式寄存器 9（PM9）来指定。
- 端口模式或控制模式（复用功能）可以以 1 位为单位进行指定。
通过端口模式控制寄存器 9（PMC9）来指定。
- 控制模式可以 1 位为单位进行指定。
通过端口功能控制寄存器 9（PFC9）和端口功能控制扩展寄存器 9（PFCE9）来指定。
- 片上上拉电阻可以以 1 位为单位进行连接。
通过上拉电阻选项寄存器 9（PU9）来指定。

端口 9 包含以下复用功能引脚。

表 4-8. 端口 9 的复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	I/O	备注	模块类型
P90	36	KR6/TXDA1	I/O	—	U-12
P91	37	KR7/RXDA1			U-7
P96	38	TIP21/TOP21			U-9
P97	39	SIB1/TIP20/TOP20			U-8
P98	40	SOB1			G-3
P99	41	SCKB1			G-5
P913	42	INTP4/PCL			W-1
P914	43	INTP5			N-2
P915	44	INTP6			N-2

注意事项 P90，P91，P96，P97，P99, 和 P913 至 P915 引脚在复用功能引脚中的输入模式具有磁滞特性，但在端口模式中没有磁滞特性。

(2) 寄存器

(a) 端口寄存器 9 (P9)

端口寄存器 9 (P9) 是用于控制引脚电平读取操作和输出电平写操作的 16 位寄存器。该寄存器可以以 16 位为单位进行读取操作或写操作。

然而，如果 P9 寄存器的高 8 位被用作 P9H 寄存器，低 8 位被用作 P9L 寄存器，那么这些寄存器将可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后：未定义		R/W	地址： FFFFFFF412H, FFFFFFF413H						
		15	14	13	12	11	10	9	8
P9（P9H ^注 ）	P915	P914	P913	0	0	0	0	P99	P98
	7	6	5	4	3	2	1	0	
（P9L）	P97	P96	0	0	0	0	0	P91	P90
		P9n	输出数据的控制（输出模式中）（n = 0, 1, 6 至 9, 13 至 15）						
		0	输出 0。						
		1	输出 1。						

注 要以 8 位 或 1 位为单位读/写 P9 寄存器的位 8 至 15，需要将这些位指定为 P9H 寄存器的位 0 至 7。

- 备注**
1. 输入模式：

当端口 9 (P9) 被读取时，此时的引脚电平将会被读取。在写操作期间，写入 P9 中的数据会进行写操作。这不会影响输入引脚。
- 输出模式：

当端口 9 (P9) 被读取时，P9 的值将会被读取。在写操作期间，值将会被写入 P9 且写入的值将会立即被输出。
2. 当 P9 在输入模式中被读取时，一个未定义的值（引脚输入电平）将会在复位后被当作值来读取。当 P9 在输出模式中被读取时，00H（输出锁值）将会被输出。

(b) 端口模式寄存器 9 (PM9)

这是一个用于指定输入或输出模式的 16 位寄存器。它们可以以 16 位为单位进行读取操作或写操作。

然而，如果 PM9 寄存器的高 8 位被用作 PM9H 寄存器，低 8 位被用作 PM9L 寄存器，那么这些寄存器将可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: FFFFH

R/W

地址: FFFFF432H, FFFFF433H

	15	14	13	12	11	10	9	8
PM9 (PM9H ^注)	PM915	PM914	PM913	0	0	0	PM99	PM98
	7	6	5	4	3	2	1	0
(PM9L)	PM97	PM96	0	0	0	0	PM91	PM90

PM9n	I/O 模式的控制 (n = 0, 1, 6 至 9, 13 至 15)
0	输出模式
1	输入模式

注 要以 8 位 或 1 位为单位读/写 PM9 寄存器的位 8 至 15，需要将这些位指定为 PM9H 寄存器的位 0 至 7。

(c) 端口模式控制寄存器 9 (PMC9)

这是一个用于指定端口模式或控制模式的 16 位寄存器。它可以以 16 位为单位进行读取操作或写操作。然而，如果 PMC9 寄存器的高 8 位被用作 PMC9H 寄存器，低 8 位被用作 PMC9L 寄存器，那么这些寄存器将可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

注意事项 如果在 PFC9.PFC9n 以及 PFCE9.PFCE9n 位的值等于默认值 (0) 时通过使用 PMC9 寄存器指定了控制模式，那么输出将会变成未定义的状态。
由于这个原因，应先将 PFC9.PFC9n 和 PFCE9.PFCE9n 位设置为 1，然后通过将 PMC9n 位设为 1 来设置控制模式。

(1/2)

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

	15	14	13	12	11	10	9	8
PMC9 (PMC9H [※])	PMC915	PMC914	PMC913	0	0	0	PMC99	PMC98

	7	6	5	4	3	2	1	0
(PMC9L)	PMC97	PMC96	0	0	0	0	PMC91	PMC90

PMC915	P915 引脚操作模式的指定
0	I/O 端口
1	INTP6 输入

PMC914	P914 引脚操作模式的指定
0	I/O 端口
1	INTP5 输入

PMC913	P913 引脚操作模式的指定
0	I/O 端口
1	INTP4/PCL I/O

注 要以 8 位 或 1 位为单位读/写 PMC9 寄存器的位 8 至 15，需要将这些位指定为 PMC9H 寄存器的位 0 至 7。

PMC99	P99 引脚操作模式的指定
0	I/O 端口
1	SCKB1 I/O

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

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

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

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

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

(d) 端口功能控制寄存器 9 (PFC9)

这是一个用于指定控制模式 1, 2, 3 或 4 的 16 位寄存器。它可以以 16 位为单位进行读取操作或写操作。

然而, 如果 PFC9 寄存器的高 8 位被用作 PFC9H 寄存器, 低 8 位被用作 PFC9L 寄存器, 那么这些寄存器将可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

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

	15	14	13	12	11	10	9	8
PFC9 (PFC9H ^注)	PFC915	PFC914	PFC913	0	0	0	PFC99	PFC98
	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	0	0	0	0	PFC91	PFC90

注 要以 8 位或 1 位为单位读/写 PFC9 寄存器的位 8 至 15 , 需要将这些位指定为 PFC9H 寄存器的位 0 至 7。

备注 关于如何指定控制模式, 请参见 4.3.8 (2) (f) P9 引脚控制模式的设置。

(e) 端口功能控制扩展寄存器 9 (PFCE9)

这是一个用于指定控制模式 1, 2, 3 或 4 的 16 位寄存器。它可以以 16 位为单位进行读取操作或写操作。

然而, 如果 PFC9 寄存器的高 8 位被用作 PFC9H 寄存器, 低 8 位被用作 PFC9L 寄存器, 那么这些寄存器将可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

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

	15	14	13	12	11	10	9	8
PFCE9 (PFCE9H ^注)	0	0	PFCE913	0	0	0	0	0
	7	6	5	4	3	2	1	0
(PFCE9L)	PFCE97	PFCE96	0	0	0	0	PFCE91	PFCE90

注 要以 8 位或 1 位为单位读/写 PFCE9 寄存器的位 8 至 15 , 需要将这些位指定为 PFCE9H 寄存器的位 0 至 7。

备注 关于如何指定控制模式, 请参见 4.3.8 (2) (f) P9 引脚控制模式的设置。

(f) P9 引脚控制模式的设置

注意事项 如果在 **PFC9.PFC9n** 以及 **PFCE9.PFCE9n** 位的值等于默认值 (0) 时通过使用 **PMC9** 寄存器指定了控制模式, 那么输出将会变成未定义的状态。

由于这个原因, 应先设置 **PFC9.PFC9n** 和 **PFCE9.PFCE9n** 位, 然后通过将 **PMC9n** 位设为 1 来设置控制模式。

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

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

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

PFC99	P99 引脚控制模式的指定
0	禁止设置
1	$\overline{\text{SCKB1}}$ I/O

PFC98	P98 引脚控制模式的指定
0	禁止设置
1	SOB1 输出

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

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

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

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

注 KR7 引脚和 RXDA1 引脚是复用功能引脚。

当将引脚用作 RXDA1 引脚时，禁止 KR7 引脚的按键中断检测。（将 KRM 寄存器的 KRM7 位清 0。）

同样地，当将引脚用作 KR7 引脚时，建议将 PFC91 位设置为 1 并将 PFCE91 位清 0。

(g) 上拉电阻选项寄存器 9 (PU9)

这是一个用于指定片上上拉电阻连接的 16 位寄存器。它可以以 16 位为单位进行读取操作或写操作。

然而，如果 PU9 寄存器的高 8 位被用作 PU9H 寄存器，低 8 位被用作 PU9L 寄存器，那么这些寄存器将可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

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

注 要以 8 位或 1 位为单位读/写 PU9 寄存器的位 8 至 15，需要将这些位指定为 PU9H 寄存器的位 0 至 7。

4.3.9 端口 CM

端口 CM 是一个可以以 1 位为单位控制 I/O 设置的 2 位端口（PCM0，PCM1）。

（1）端口 CM 的功能

- 端口的输入/输出数据可以以 1 位为单位进行指定。
通过端口寄存器 CM（PCM）来指定。
- 端口的输入/输出模式可以以 1 位为单位进行指定。
通过端口模式寄存器 CM（PMCM）来指定。
- 端口模式或控制模式（复用功能）可以以 1 位为单位进行指定。
通过端口模式控制寄存器 CM（PMCCM）来指定。

端口 CM 包含以下复用功能引脚。

表 4-9. 端口 CM 的复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	I/O	备注	模块类型
PCM0	45	—	I/O	—	B-1
PCM1	46	CLKOUT			D-2

(2) 寄存器

(a) 端口寄存器 CM (PCM)

端口寄存器 CM (PCM) 是一个用于控制引脚电平读取操作和输出电平写操作的 8 位寄存器。该寄存器可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 未定义		R/W		地址: FFFFF00CH			
	7	6	5	4	3	2	1 0
PCM	0	0	0	0	0	0	PCM1 PCM0

PCMn	输出数据的控制 (输出模式中) (n = 0, 1)
0	输出 0。
1	输出 1。

- 备注**
1. 输入模式: 当端口 CM (PCM) 被读取时, 此时的引脚电平将会被读取。在写操作期间, 写入 PCM 中的数据会进行写操作。这不会影响输入引脚。
输出模式: 当端口 CM (PCM) 被读取时, PCM 的值将会被读取。在写操作期间, 值将会被写入 PCM 且写入的值将会立即被输出。
 2. 当 PCM 在输入模式中被读取时, 一个未定义的值 (引脚输入电平) 将会在复位后被当作值来读取。当 PCM 在输出模式中被读取时, 00H (输出锁值) 将会被输出。

(b) 端口模式寄存器 CM (PMCM)

这是一个用于指定输入或输出模式的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: FFH		R/W		地址: FFFFF02CH			
	7	6	5	4	3	2	1 0
PMCM	1	1	1	1	1	1	PMCM1 PMCM0

PMCMn	I/O 模式的控制 (n = 0, 1)
0	输出模式
1	输入模式

(c) 端口模式控制寄存器 CM (PMCCM)

这是一个用于指定端口模式或控制模式的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

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

4.3.10 端口 DL

端口 DL 是一个可以以 1 位为单位控制 I/O 设置的 8 位端口（PDL0 至 PDL7）。

(1) 端口 DL 的功能

- 端口的输入/输出数据可以以 1 位为单位进行指定。
通过端口寄存器 DL（PDL）来指定。
- 端口的输入/输出模式可以以 1 位为单位进行指定。
通过端口模式寄存器 DL（PMDL）来指定。

端口 DL 包含以下复用功能引脚。

表 4-10. 端口 DL 的复用功能引脚

引脚名称	引脚编号	复用功能引脚名称	I/O	备注	模块类型
PDL0	47		I/O	—	B-1
PDL1	48				B-1
PDL2	49				B-1
PDL3	50				B-1
PDL4	51				B-1
PDL5	52	FLMD1 ^注			B-1
PDL6	53				B-1
PDL7	54				B-1

注 由于只在 Flash 存储器编程模式下使用 FLMD1，因此不需要通过使用端口控制寄存器来操作。详细信息请参考第 22 章 Flash 存储器。

(2) 寄存器

(a) 端口寄存器 DLL (PDLL)

端口寄存器 DLL (PDLL) 是一个用于控制引脚电平读取操作和输出电平写操作的 8 位寄存器。该寄存器可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: 未定义 R/W 地址: FFFFF005H

	7	6	5	4	3	2	1	0
PDLL	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

PDLn	输出数据的控制 (输出模式中) (n = 0 至 7)
0	输出 0。
1	输出 1。

- 备注**
1. 输入模式: 当端口 DLL (PDLL) 被读取时, 此时的引脚电平将会被读取。在写操作期间, 写入 PDLL 中的数据会进行写操作。这不会影响输入引脚。
输出模式: 当端口 DLL (PDLL) 被读取时, PDLL 的值将会被读取。在写操作期间, 值将会被写入 PDLL 且写入的值将会立即被输出。
 2. 当 PDLL 在输入模式中被读取时, 一个未定义的值 (引脚输入电平) 将会在复位后被当作值来读取。当 PDLL 在输出模式中被读取时, 00H (输出锁值) 将会被输出。

(b) 端口模式寄存器 DLL (PMDLL)

这是一个用于指定输入或输出模式的 8 位寄存器。它可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。

复位后: FFH R/W 地址: FFFFF025H

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

PMDLn	I/O 模式的控制 (n = 0 至 7)
0	输出模式
1	输入模式

4.3.11 复用功能为片上调试功能的端口引脚

表-11 中显示的复用功能引脚为片上调试引脚。在外部复位后，这些引脚会被初始化为片上调试引脚（ $\overline{\text{DRST}}$ ，DDI，DDO，DCK 和 DMS）。

表 4-11. 片上调试引脚

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

要将这些引脚用作端口引脚而不是片上调试引脚，则必须在外部复位后执行以下操作。

- <1> 将 OCDM 寄存器（特殊寄存器）的 OCDM0 位清 0。
- <2> 将 P05/ $\overline{\text{INTP2/DRST}}$ 引脚固定为低电平直到上述操作被完成。

当片上调试功能没有被使用时，在上述操作被执行前将一个高电平输入 $\overline{\text{DRST}}$ 引脚可能会引起故障（CPU 锁死）。因此，处理 P05 引脚时应特别小心。

当高电平没有被输入到 P05/ $\overline{\text{INTP2/DRST}}$ 引脚时（该引脚固定为低电平时），将没有必要操作 OCDM.OCDM0 位。

由于下拉电阻（30 k Ω TYP.）被连接到 P05/ $\overline{\text{INTP2/DRST}}$ 引脚的缓冲器上，引脚将不必通过一个外部源来固定为低电平。下拉电阻通过将 OCDM0 位清 0 来断开连接。

详细信息参见第 24 章 片上调试功能。

4.3.12 将端口引脚用作复用功能引脚时的寄存器设置

表 4-12. 将端口引脚用作复用功能引脚 (1/5)

引脚名称	复用功能引脚		PMn 寄存器	PMCn 寄存器	PFCm 寄存器	PFCEm 寄存器	其它位 (寄存器)
	名称	I/O					
P00	TIP31	输入	无需设置	PMC00 = 1	PFC00 = 0	—	
	TOP31	输出	无需设置	PMC00 = 1	PFC00 = 1	—	
P01	TIP30	输入	无需设置	PMC01 = 1	PFC01 = 0	—	
	TOP30	输出	无需设置	PMC01 = 1	PFC01 = 1	—	
P02	NMI	输入	无需设置	PMC02 = 1	—	—	
P03	INTP0	输入	无需设置	PMC03 = 1	PFC03 = 0	—	INTx03 (INTx0)
	ADTRG	输出	无需设置	PMC03 = 1	PFC03 = 1	—	
P04	INTP1	输入	无需设置	PMC04 = 1	—	—	INTx04 (INTx0)
P05 [*]	INTP2	输入	无需设置	PMC05 = 1	—	—	INTx05 (INTx0)
	$\overline{\text{DRST}}$	输入	无需设置	无需设置	—	—	OCDM0 (OCDM) = 1
P06	INTP3	输入	无需设置	PMC06 = 1	—	—	INTx06 (INTx0)

注 外部复位后，P05/INTP2/ $\overline{\text{DRST}}$ 引脚被初始化为片上调试引脚 ($\overline{\text{DRST}}$)。若不想将 P05/INTP2/ $\overline{\text{DRST}}$ 引脚用作片上调试引脚，请参见第 24 章 片上调试功能。

备注

1. 当复用功能被使用时端口寄存器 (Pn) 不需要进行设置。
2. INTxn = INTFn, INTRn

表 4-12. 将端口引脚用作复用功能引脚（2/5）

引脚名称	复用功能引脚		PMn 寄存器	PMCn 寄存器	PFCm 寄存器	PFCEm 寄存器	其它位（寄存器）
	名称	I/O					
P30	TXDA0	输出	无需设置	PMC30 = 1	—	—	
P31	RXDA0	输入	无需设置	PMC31 = 1	—	—	注
	INTP7	输入	无需设置	PMC31 = 1	—	—	注, INTx31 (INTx3)
P32	ASCKA0	输入	无需设置	PMC32 = 1	PFC32 = 0	PFCE32 = 0	
	TOP01	输出	无需设置	PMC32 = 1	PFC32 = 1	PFCE32 = 0	
	TIP00	输入	无需设置	PMC32 = 1	PFC32 = 0	PFCE32 = 1	
	TOP00	输出	无需设置	PMC32 = 1	PFC32 = 1	PFCE32 = 1	
P33	TIP01	输入	无需设置	PMC33 = 1	PFC33 = 0	—	
	TOP01	输出	无需设置	PMC33 = 1	PFC33 = 1	—	
P34	TIP10	输入	无需设置	PMC34 = 1	PFC34 = 0	—	
	TOP10	输出	无需设置	PMC34 = 1	PFC34 = 1	—	
P35	TIP11	输入	无需设置	PMC35 = 1	PFC35 = 0	—	
	TOP11	输出	无需设置	PMC35 = 1	PFC35 = 1	—	
P40	SIB0	输入	无需设置	PMC40 = 1	—	—	
P41	SOB0	输出	无需设置	PMC41 = 1	—	—	
P42	SCKB0	I/O	无需设置	PMC42 = 1	—	—	

注 INTP7 引脚的功能与 RXDA0 引脚的功能是交互的。要用作 RXDA0 引脚，则使复用功能引脚 INTP7 的边缘检测功能失效（通过将 INTF3L.INTF31 位以及 INTR3L.INTR31 位清 0）。要用作 INTP7 引脚，则停止 UARTA0 的接收操作（通过将 UA0CTL0.UA0RXE 位清 0）。

备注

1. 当复用功能被使用时端口寄存器（Pn）不需要进行设置。
2. INTxn = INTFn, INTRn

表 4-12. 将端口引脚用作复用功能引脚 (3/5)

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

注 1. KRn 引脚的功能与 TIQ0m 引脚的功能是交互的。要将该引脚用作 TIQ0m 引脚，则应使复用功能引脚 KRn 的按键中断检测功能失效（通过将 KRM.KRMn 位清 0）。要将该引脚用作 KRn 引脚，则应使复用功能引脚 TIQ0m 的边缘检测功能失效（n = 0 至 3，m = 0 至 3）。

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

2. DDI, DDO, DCK 和 DMS 引脚是片上调试引脚。若在外复位后不想将这些引脚用作片上调试引脚，请参见第 24 章 片上调试功能。

注意事项 如果在 PFC5.PFC5n 以及 PFCE5.PFCE5n 位的值等于默认值（0）时通过使用 PMC5 寄存器指定了控制模式，那么输出将会变成未定义的状态。

由于这个原因，应先设置 PFC5.PFC5n 和 PFCE5.PFCE5n 位，然后通过将 PMC5n 位设为 1 来设置控制模式。

备注 1. 当复用功能被使用时端口寄存器 (Pn) 不需要进行设置。

2. INTxn = INTFn, INTRn

表 4-12. 将端口引脚用作复用功能引脚（4/5）

引脚名称	复用功能引脚		PMn 寄存器	PMCn 寄存器	PFCm 寄存器	PFCEm 寄存器	其它位（寄存器）
	名称	I/O					
P70	ANI0	输入	PM70 = 1 ^注	—	—	—	
P71	ANI1	输入	PM71 = 1 ^注	—	—	—	
P72	ANI2	输入	PM72 = 1 ^注	—	—	—	
P73	ANI3	输入	PM73 = 1 ^注	—	—	—	
P74	ANI4	输入	PM74 = 1 ^注	—	—	—	
P75	ANI5	输入	PM75 = 1 ^注	—	—	—	
P76	ANI6	输入	PM76 = 1 ^注	—	—	—	
P77	ANI7	输入	PM77 = 1 ^注	—	—	—	
P78	ANI8	输入	PM78 = 1 ^注	—	—	—	
P79	ANI9	输入	PM79 = 1 ^注	—	—	—	

注 将 PM7n 设为 1 以使用 P7n（ANIn）的复用功能。

备注 1. 当复用功能被使用时端口寄存器（Pn）不需要进行设置。
2. INTxn = INTFn, INTRn

表 4-12. 将端口引脚用作复用功能引脚 (5/5)

引脚名称	复用功能引脚		PMn 寄存器	PMCn 寄存器	PFCm 寄存器	PFCEm 寄存器	其它位 (寄存器)
	名称	I/O					
P90	KR6	输入	无需设置	PMC90 = 1	PFC90 = 1	PFCE90 = 0	
	TXDA1	输出	无需设置	PMC90 = 1	PFC90 = 0	PFCE90 = 1	
P91	KR7 ^{注1}	输入	无需设置	PMC91 = 1	PFC91 = 1	PFCE91 = 0	
					PFC91 = 0	PFCE91 = 1	
	RXDA1 ^{注1}	输入	无需设置	PMC91 = 1	PFC91 = 0	PFCE91 = 1	
P96	TIP21	输入	无需设置	PMC96 = 1	PFC96 = 0	PFCE96 = 1	
	TOP21	输出	无需设置	PMC96 = 1	PFC96 = 1	PFCE96 = 1	
P97	SIB1	输入	无需设置	PMC97 = 1	PFC97 = 1	PFCE97 = 0	
	TIP20	输入	无需设置	PMC97 = 1	PFC97 = 0	PFCE97 = 1	
	TOP20	输出	无需设置	PMC97 = 1	PFC97 = 1	PFCE97 = 1	
P98	SOB1	输出	无需设置	PMC98 = 1	PFC98 = 1	—	
P99	SCKB1	I/O	无需设置	PMC99 = 1	PFC99 = 1	—	
P913	INTP4	输入	无需设置	PMC913 = 1	PFC913 = 1	PFCE913 = 0	INTx913 (INTx9H)
	PCL	输出	无需设置	PMC913 = 1	PFC913 = 0	PFCE913 = 1	
P914	INTP5	输入	无需设置	PMC914 = 1	PFC914 = 1	—	INTx914 (INTx9H)
P915	INTP6	输入	无需设置	PMC915 = 1	PFC915 = 1	—	INTx915 (INTx9H)
PCM1	CLKOUT	输出	无需设置	PMCCM1 = 1	—	—	
PDL5	FLMD1	输入	无需设置	无需设置	—	—	注 2

注 1. KR7 引脚和 RXDA1 引脚是复用功能引脚。

当将引脚用作 RXDA1 引脚时，禁止 KR7 引脚的按键中断检测。（将 KRM.KRM7 位清 0。）同样地，当将引脚用作 KR7 引脚时，建议将 PFC91 位设置为 1 并将 PFCE91 位清 0。

2. 由于 FLMD1 引脚是在 Flash 存储器编程模式中被使用的，因此 FLMD1 引脚不需要通过使用端口控制寄存器来操作。详细信息请参见第 22 章 Flash 存储器。

注意事项 如果在 PFC9.PFC9n 以及 PFCE9.PFCE9n 位的值等于默认值 (0) 时通过使用 PMC9 寄存器指定了控制模式，那么输出将会变成未定义的状态。

由于这个原因，应先将 PFC9.PFC9n 和 PFCE9.PFCE9n 位设置为 1，然后通过将 PMC9n 位设为 1 来设置控制模式。

备注 1. 当复用功能被使用时端口寄存器 (Pn) 不需要进行设置。

2. INTxn = INTFn, INTRn

4.4 端口结构图

图 4-2. A-1 类型的结构图

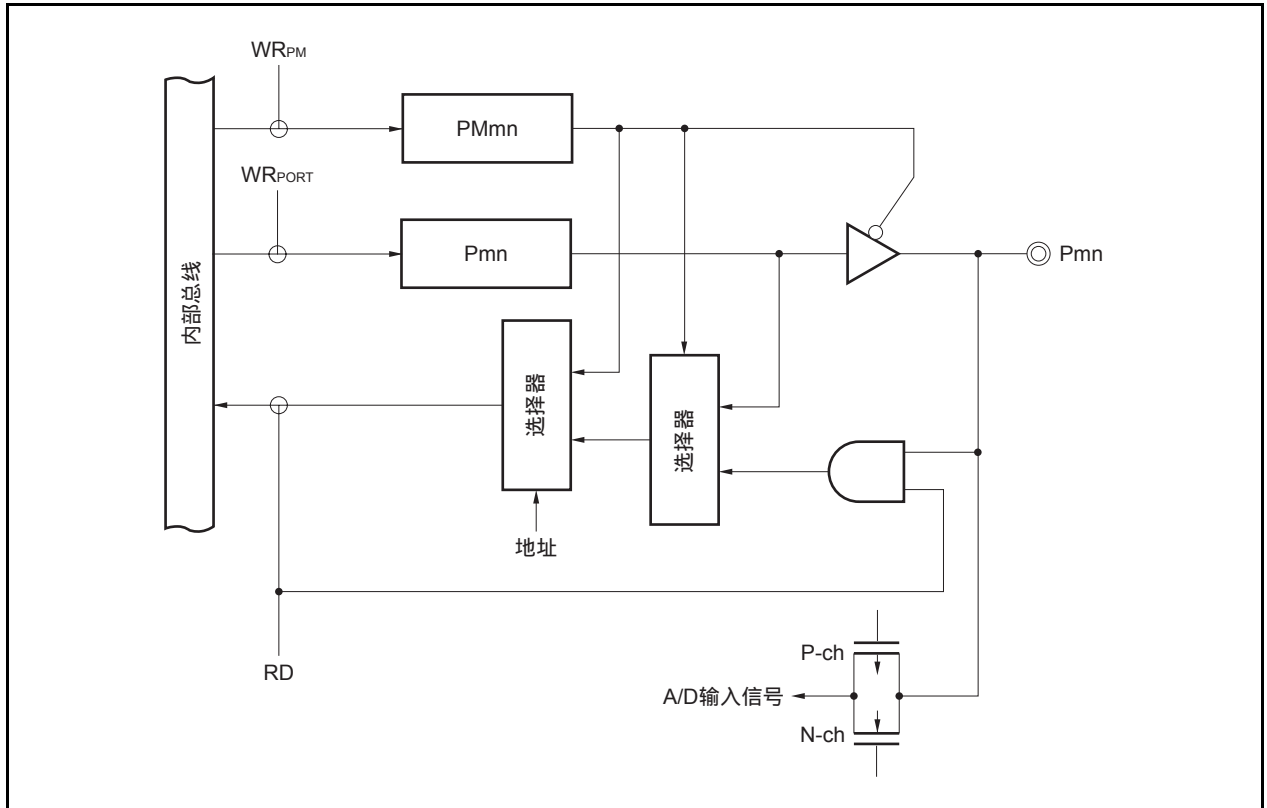


图 4-3. B-1 类型的结构图

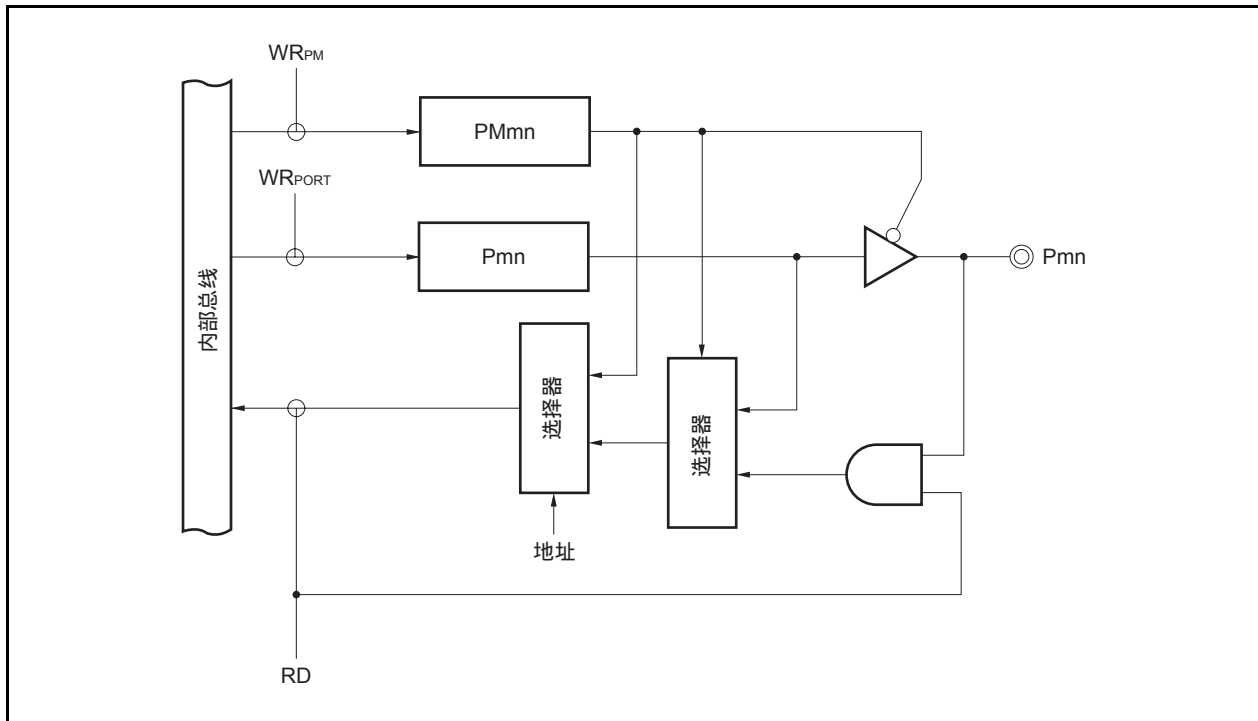


图 4-4. D-2 类型的结构图

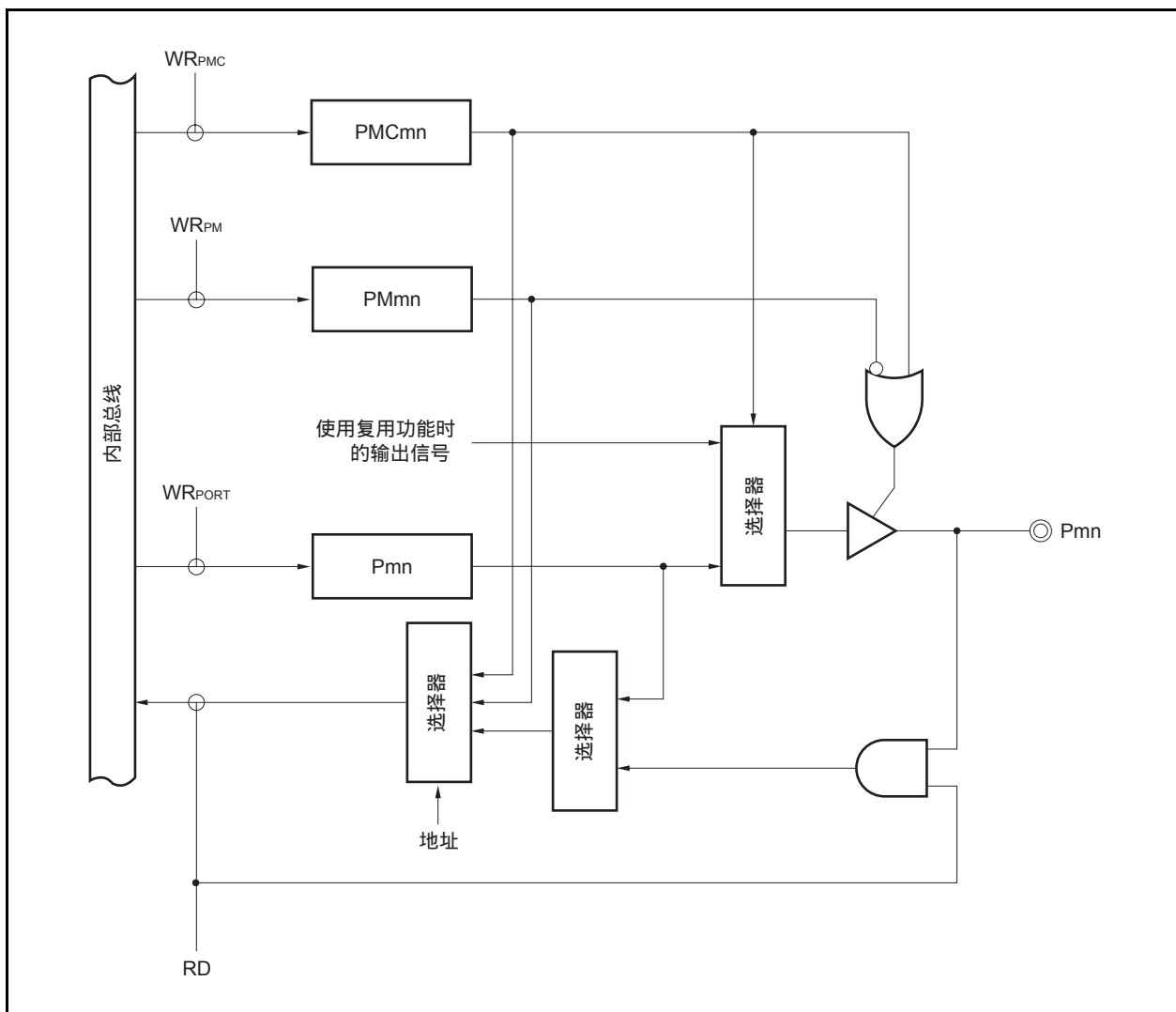


图 4-5. E-1 类型的结构图

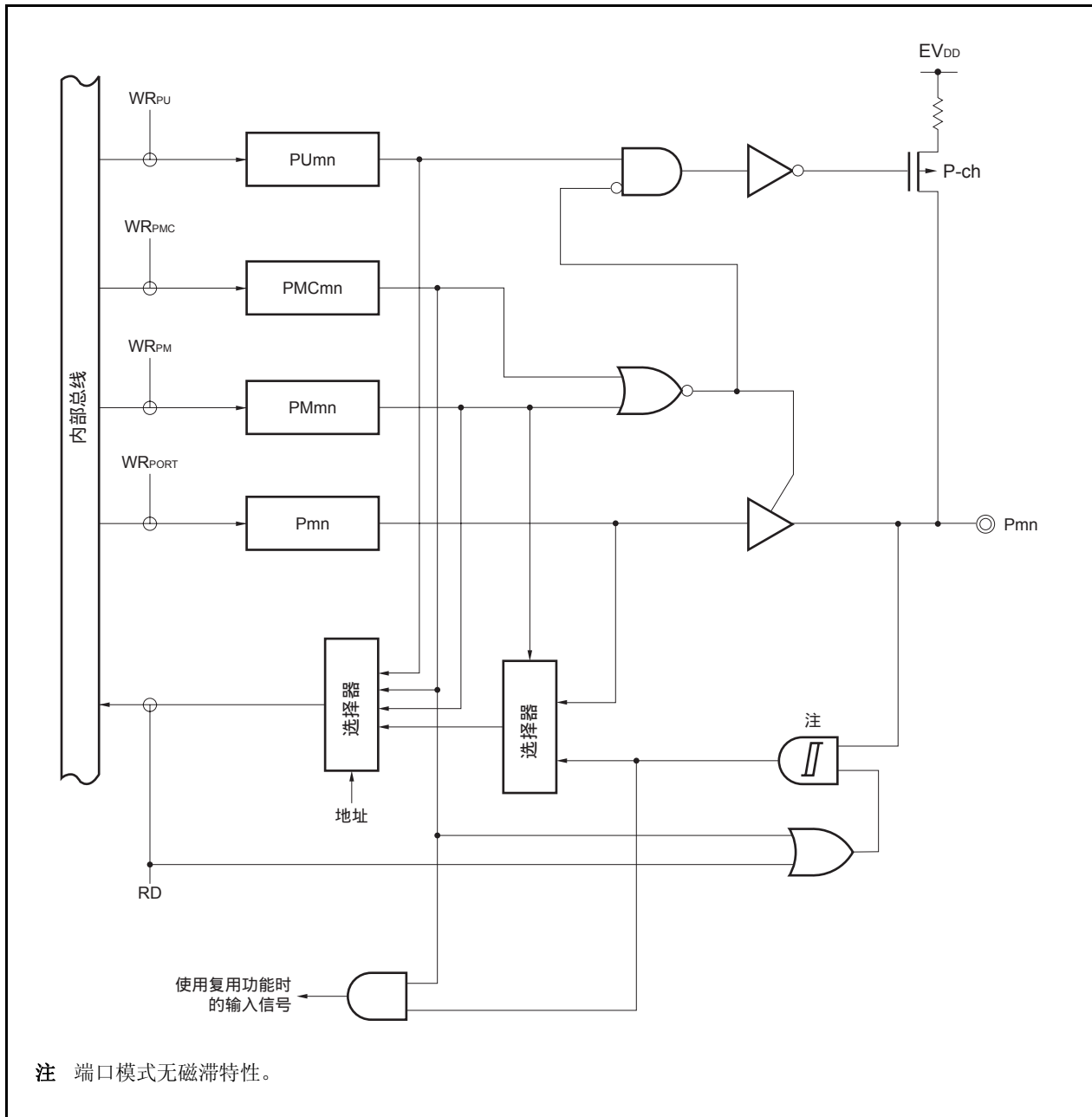


图 4-6. E-2 类型的结构图

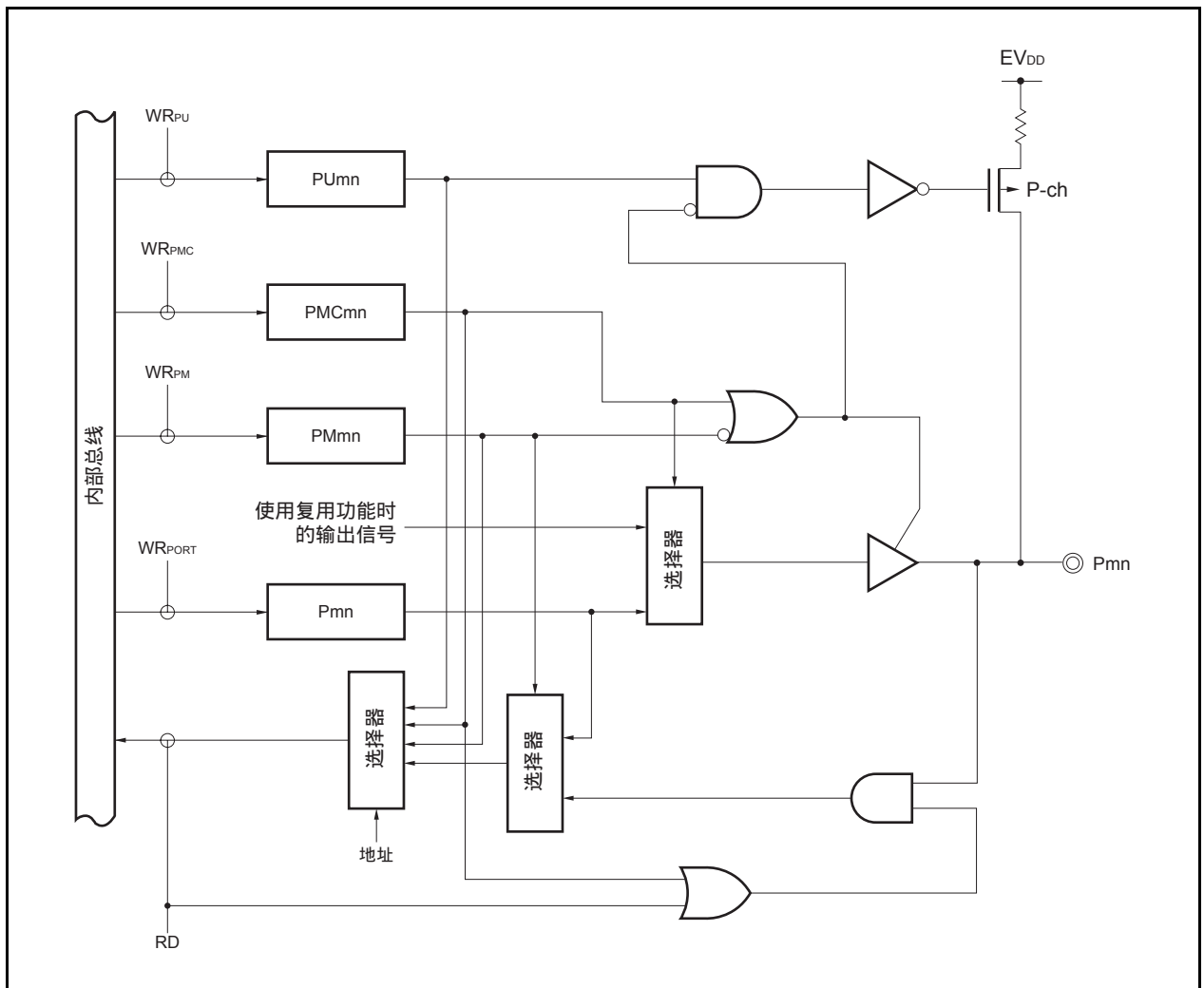


图 4-7. E-3 类型的结构图

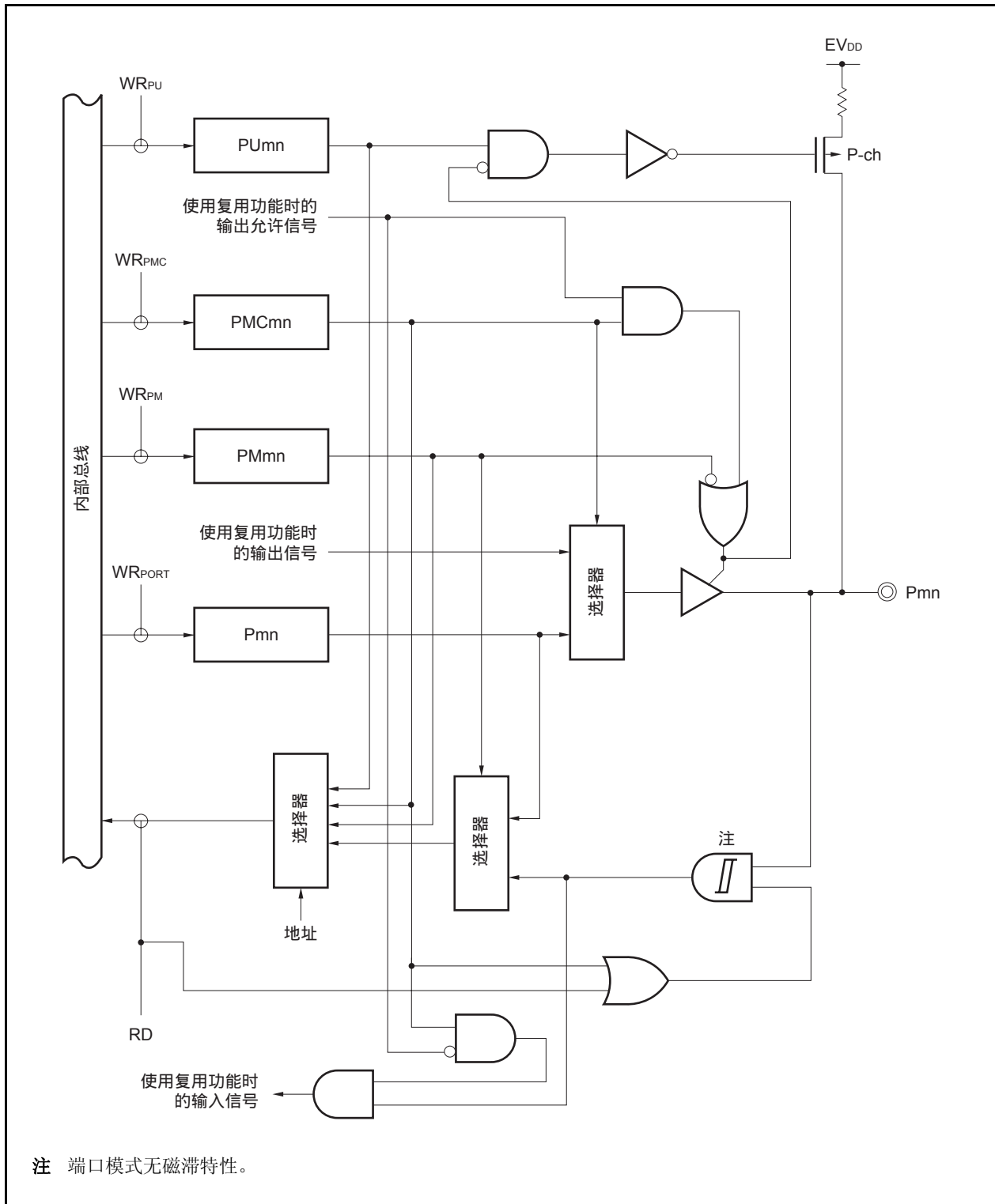


图 4-8. G-1 类型的结构图

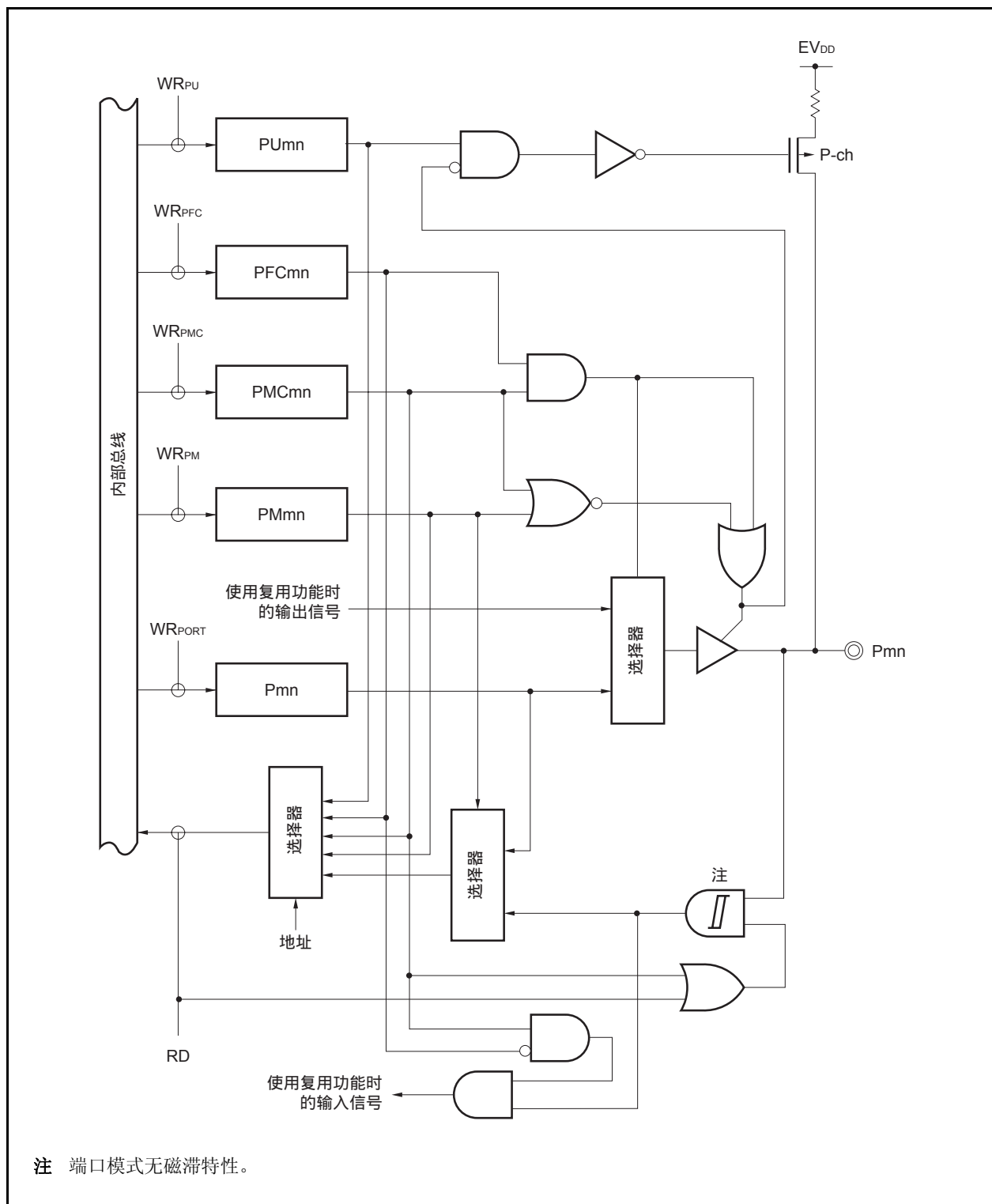


图 4-9. G-2 类型的结构图

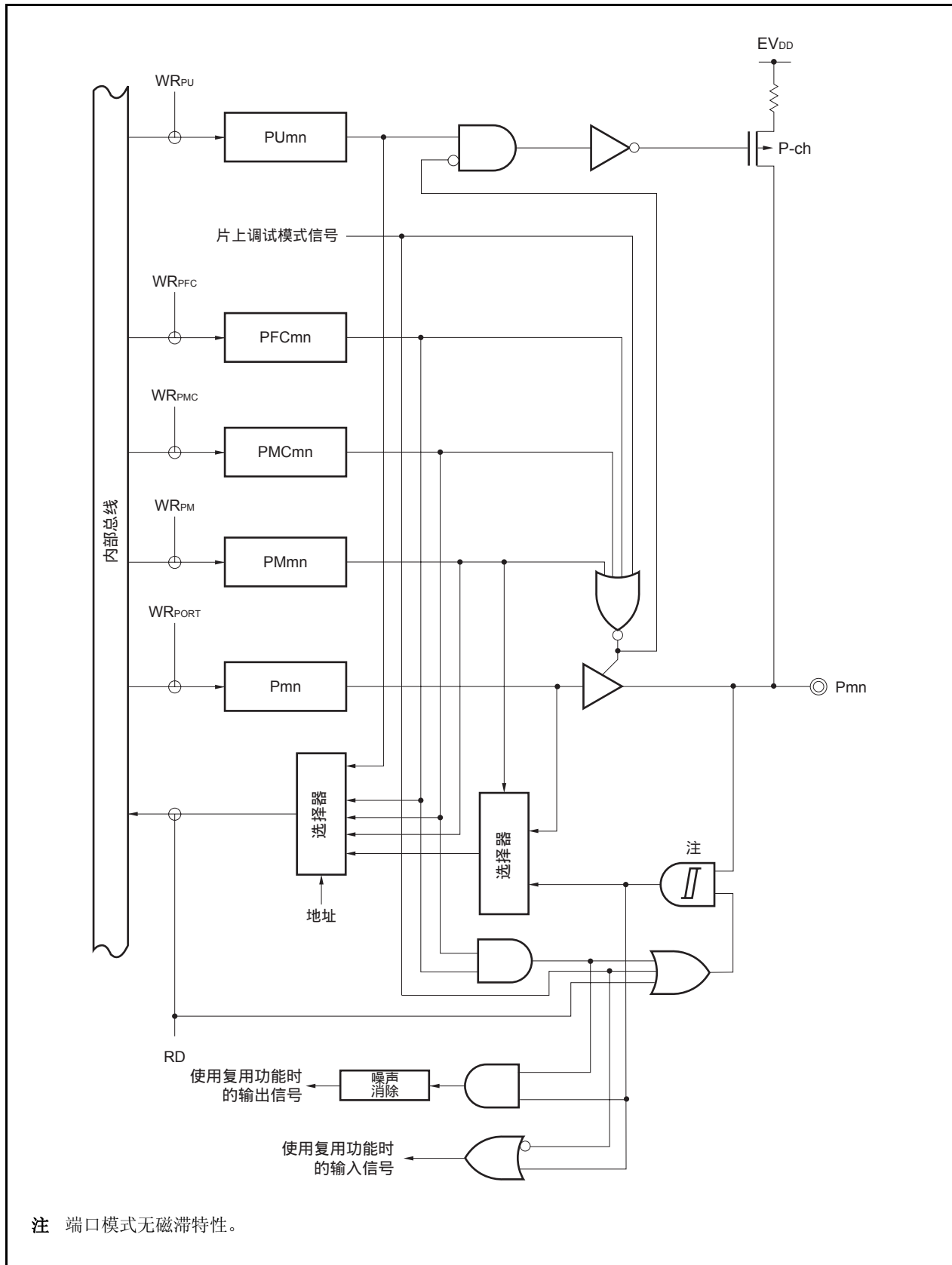


图 4-10. G-3 类型的结构图

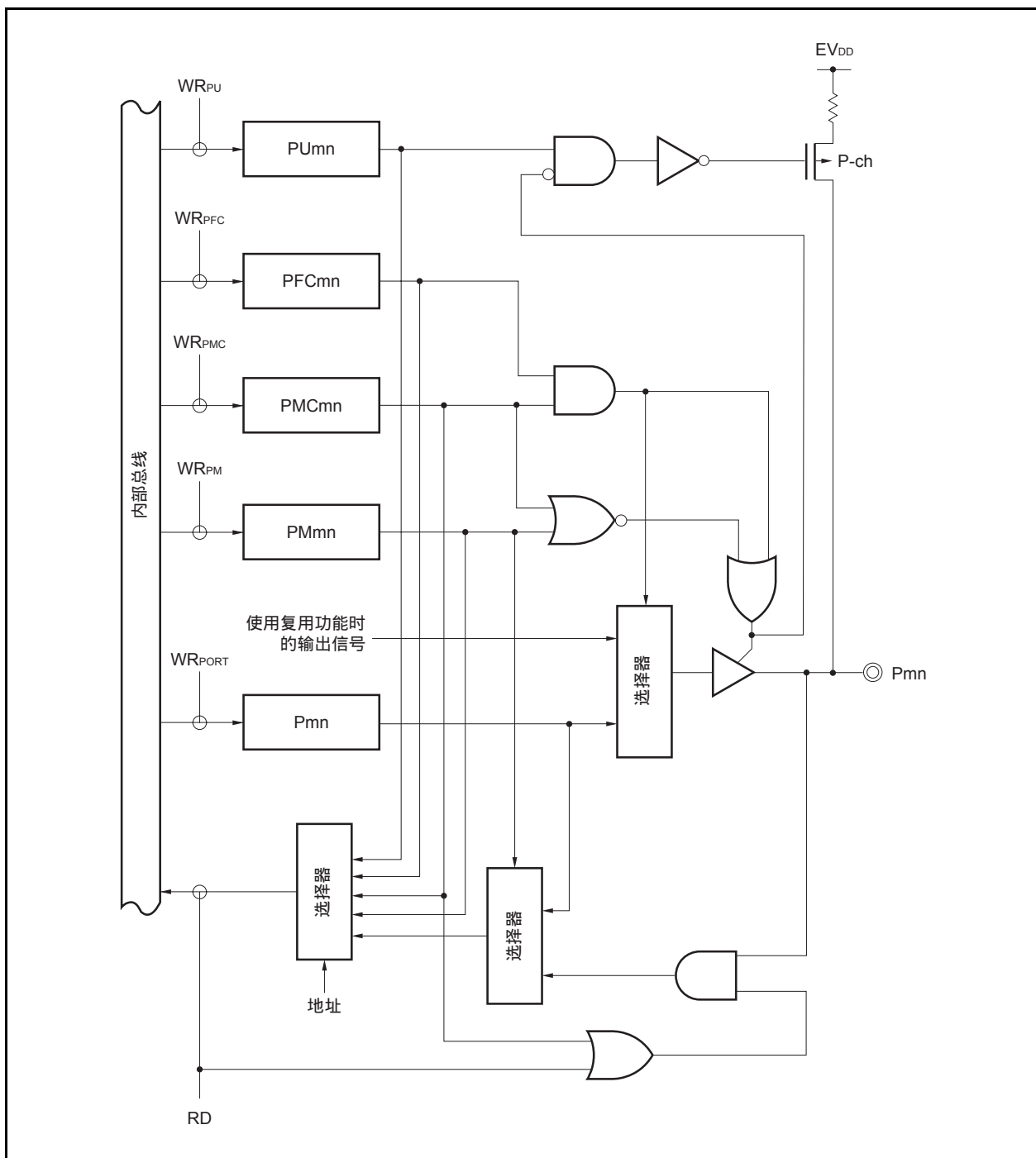
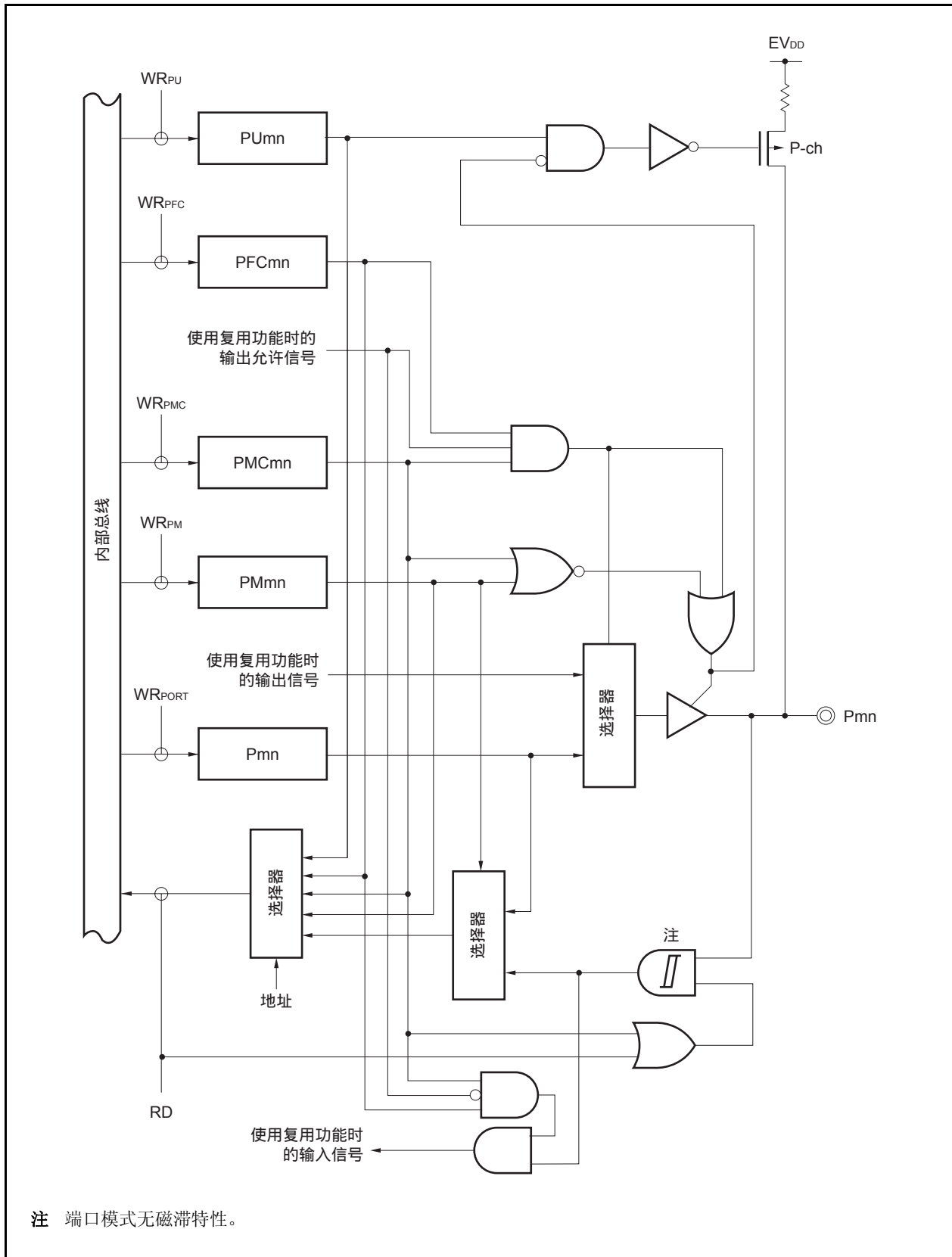


图 4-11. G-5 类型的结构图



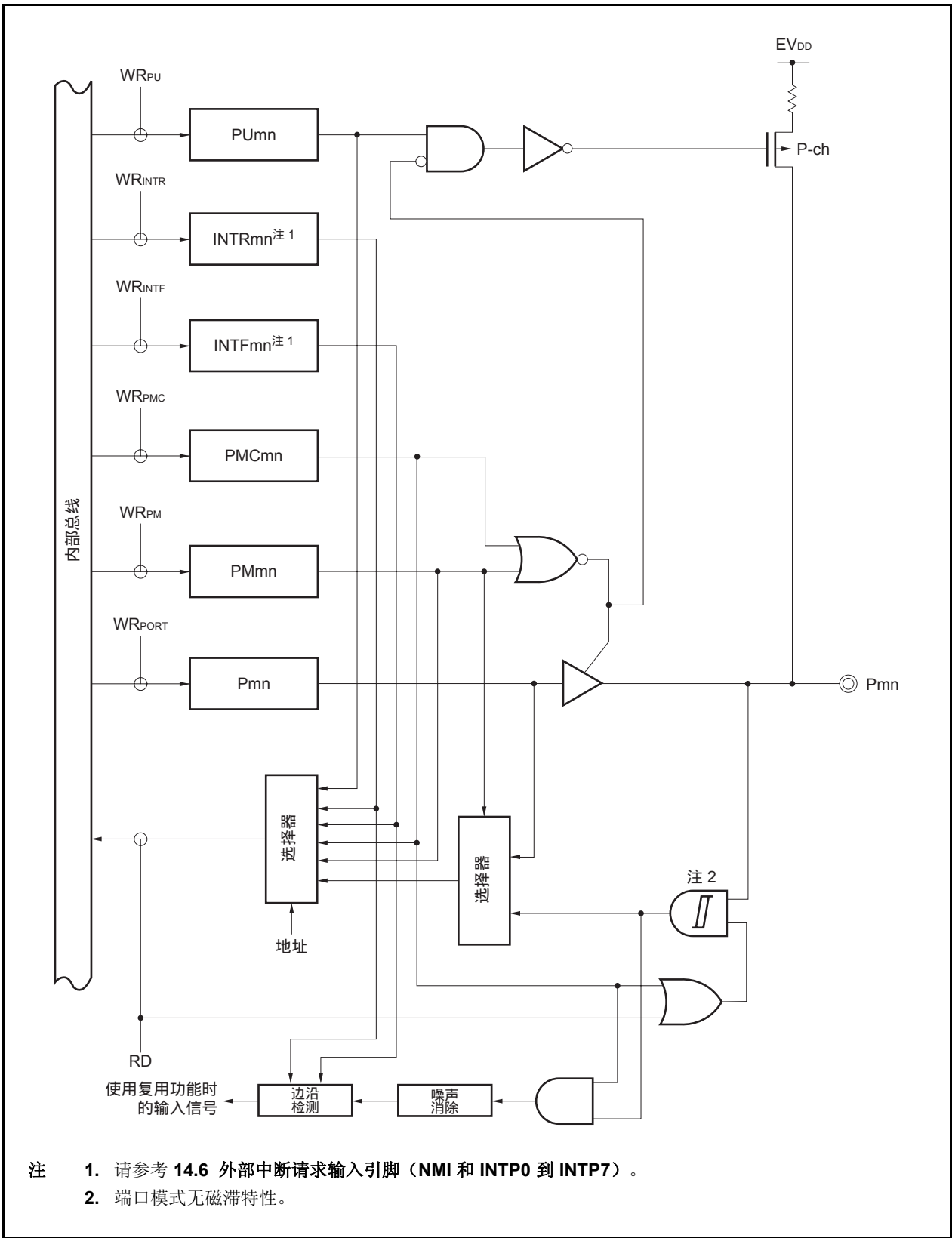
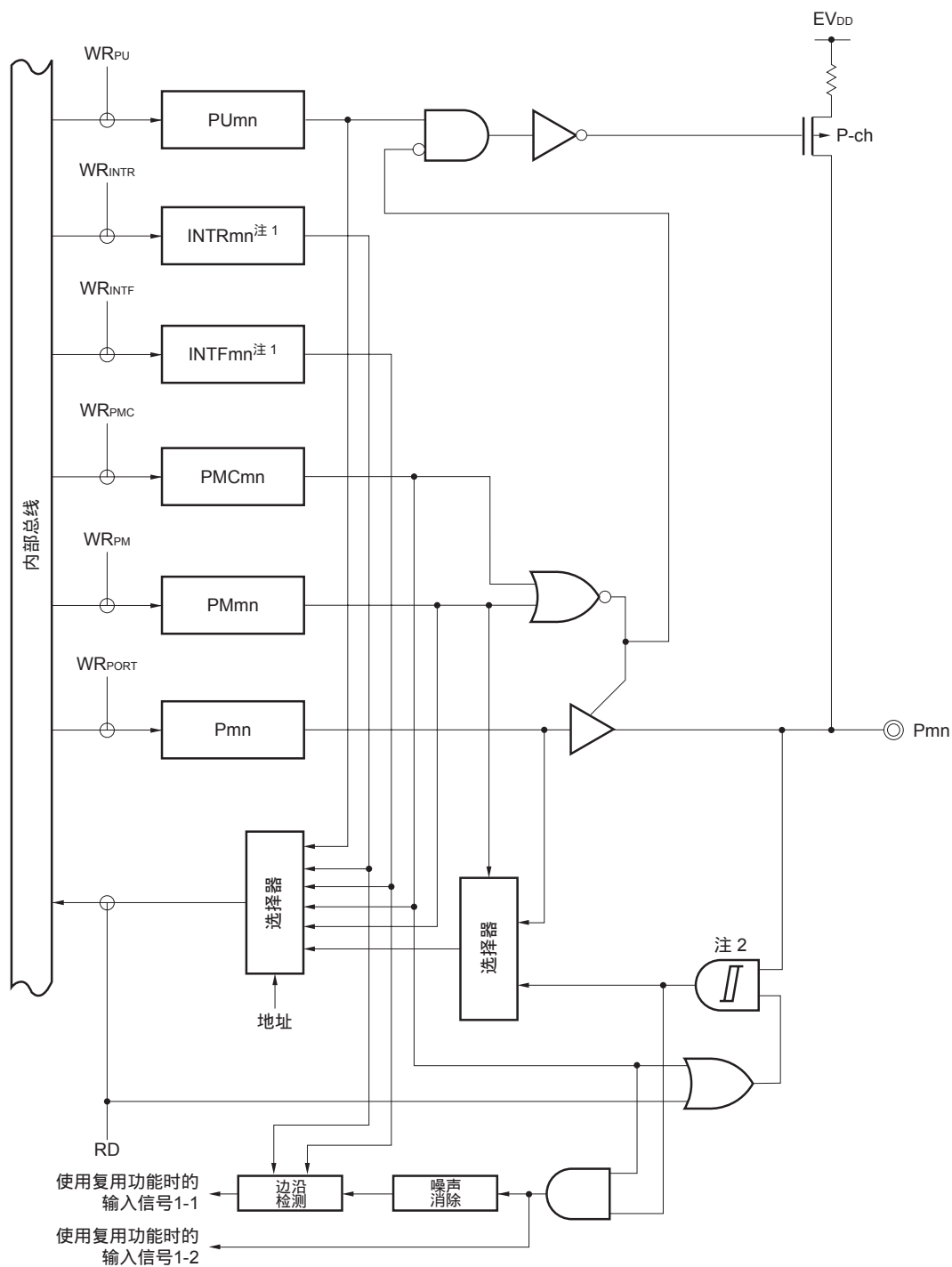


图 4-13. L-2 类型的结构图



- 注 1. 请参考 14.6 外部中断请求输入引脚（NMI 和 INTP0 到 INTP7）。
- 注 2. 端口模式无磁滞特性。

图 4-14. N-1 类型的结构图

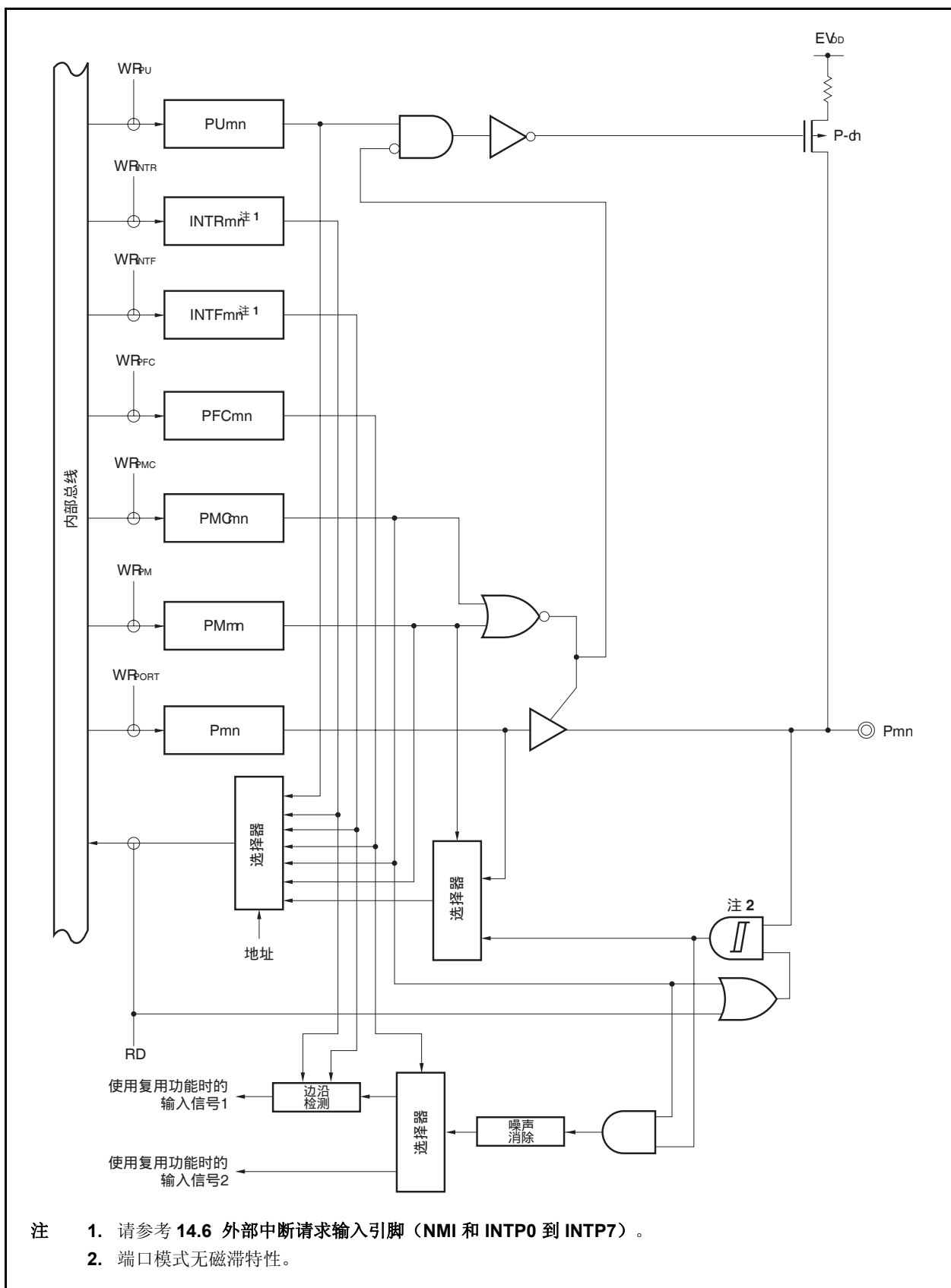
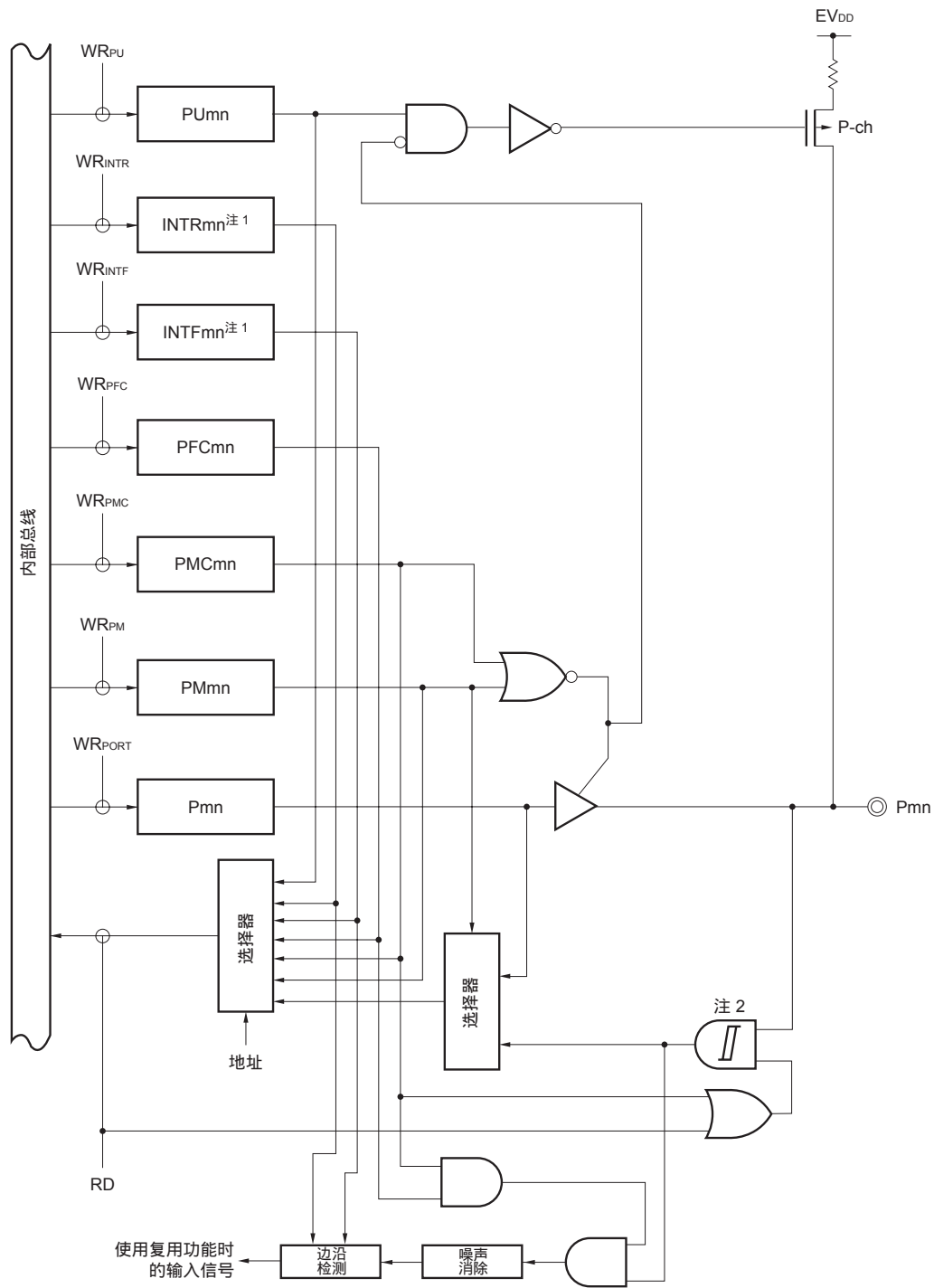


图 4-15. N-2 类型的结构图



- 注 1. 请参考 14.6 外部中断请求输入引脚（NMI 和 INTP0 到 INTP7）。
- 注 2. 端口模式无磁滞特性。

图 4-16. U-4 类型的结构图

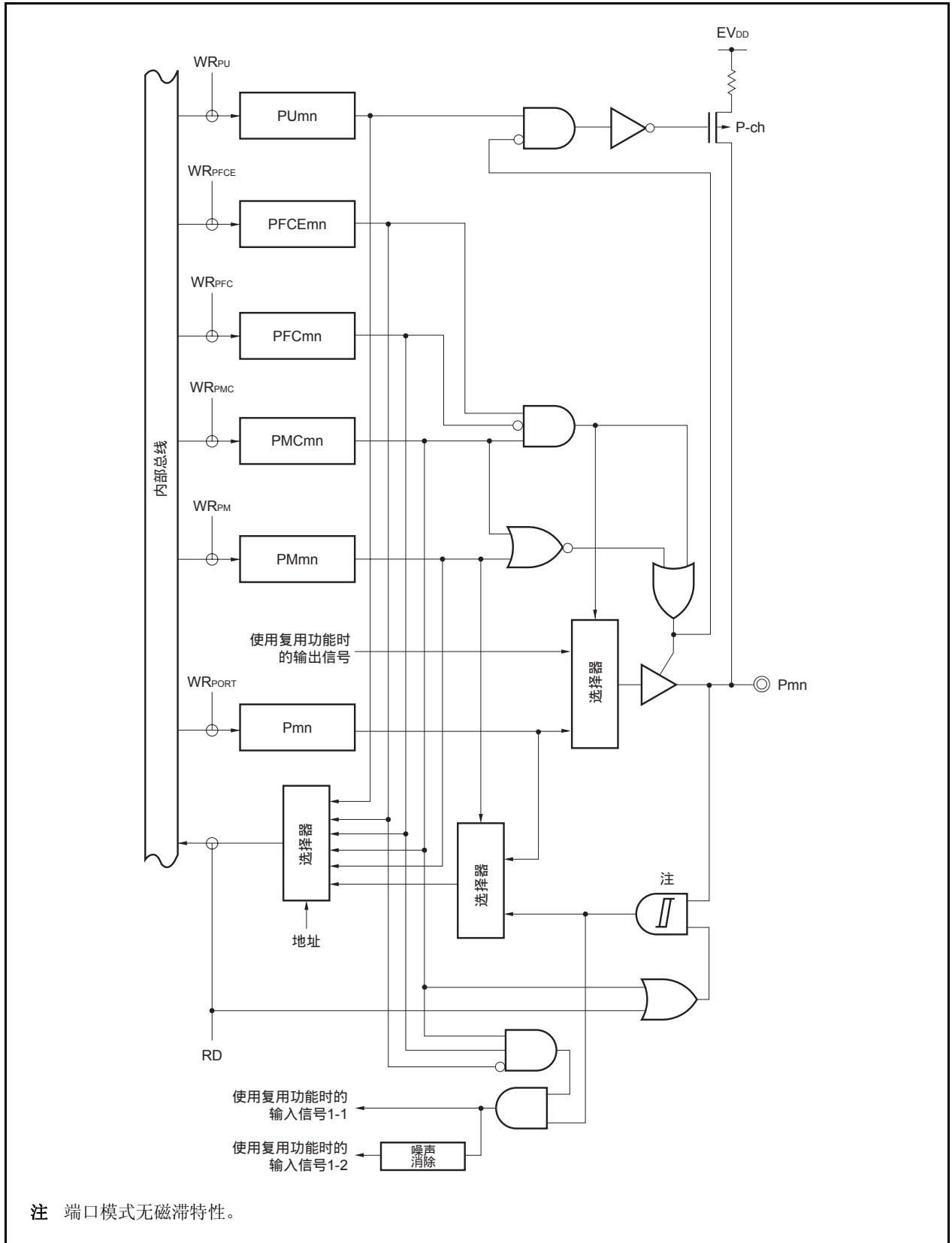
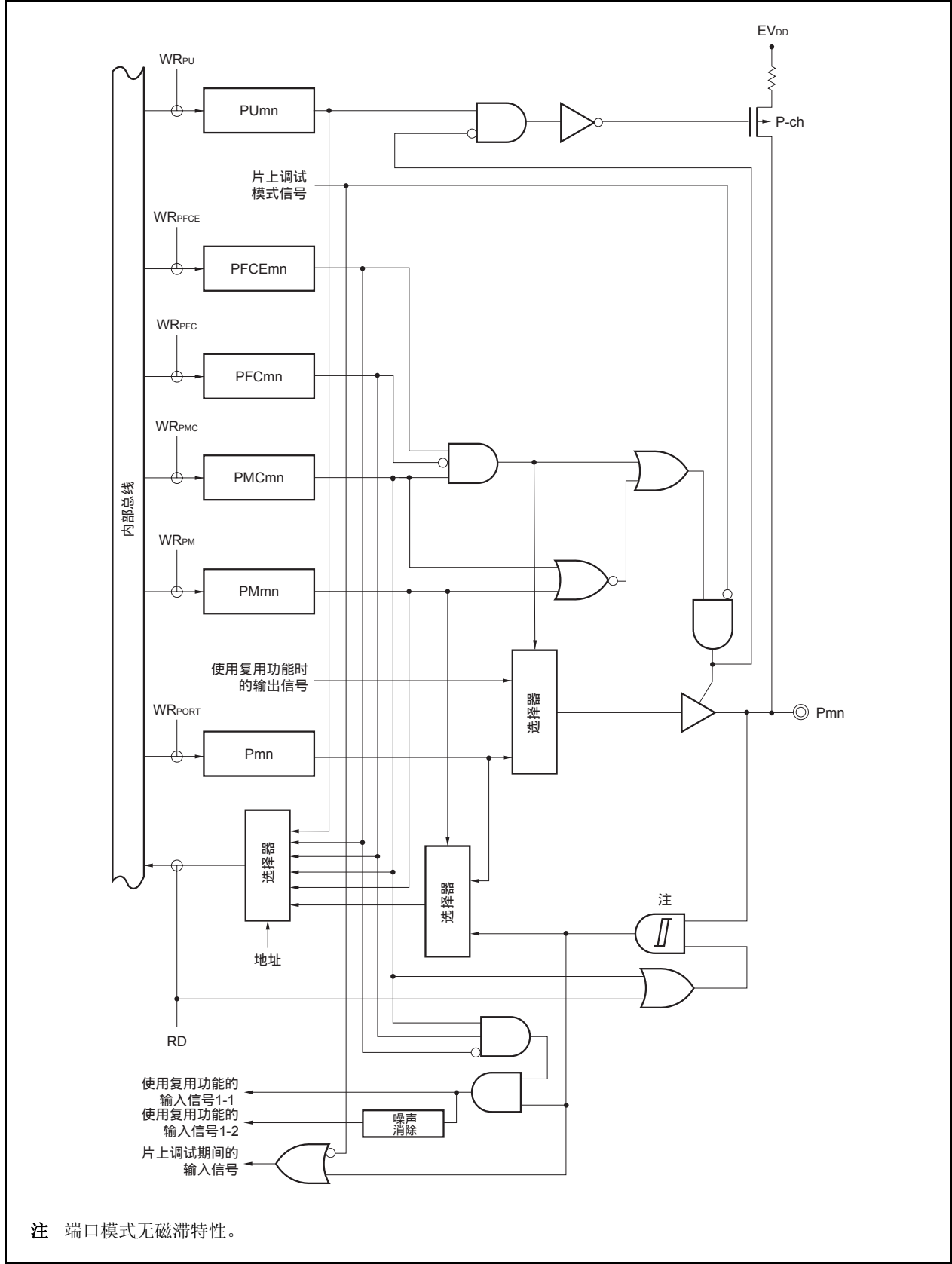


图 4-17. U-5 类型的结构图



注 端口模式无磁滞特性。

图 4-18. U-6 类型的结构图

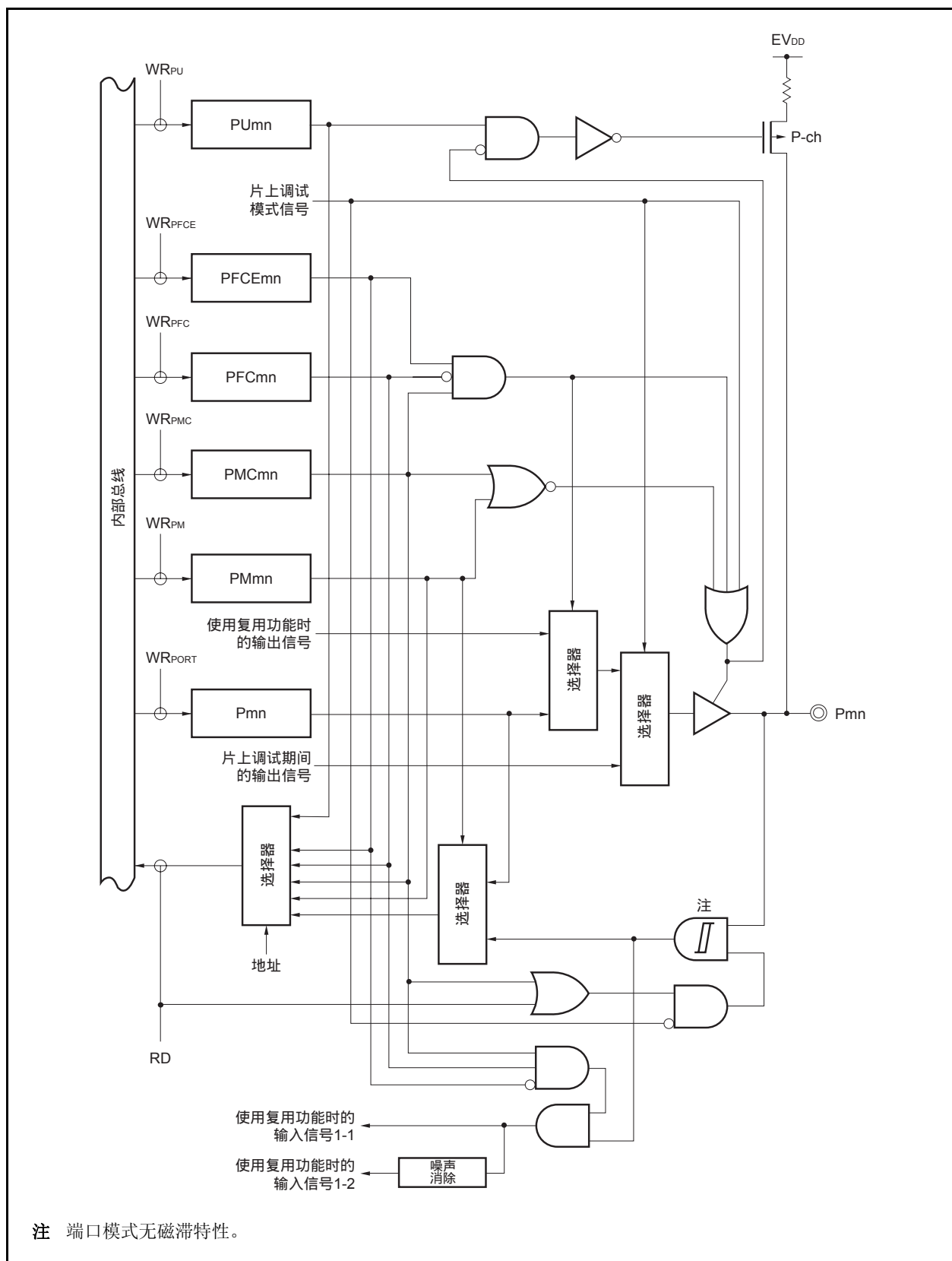


图 4-19. U-7 类型的结构图

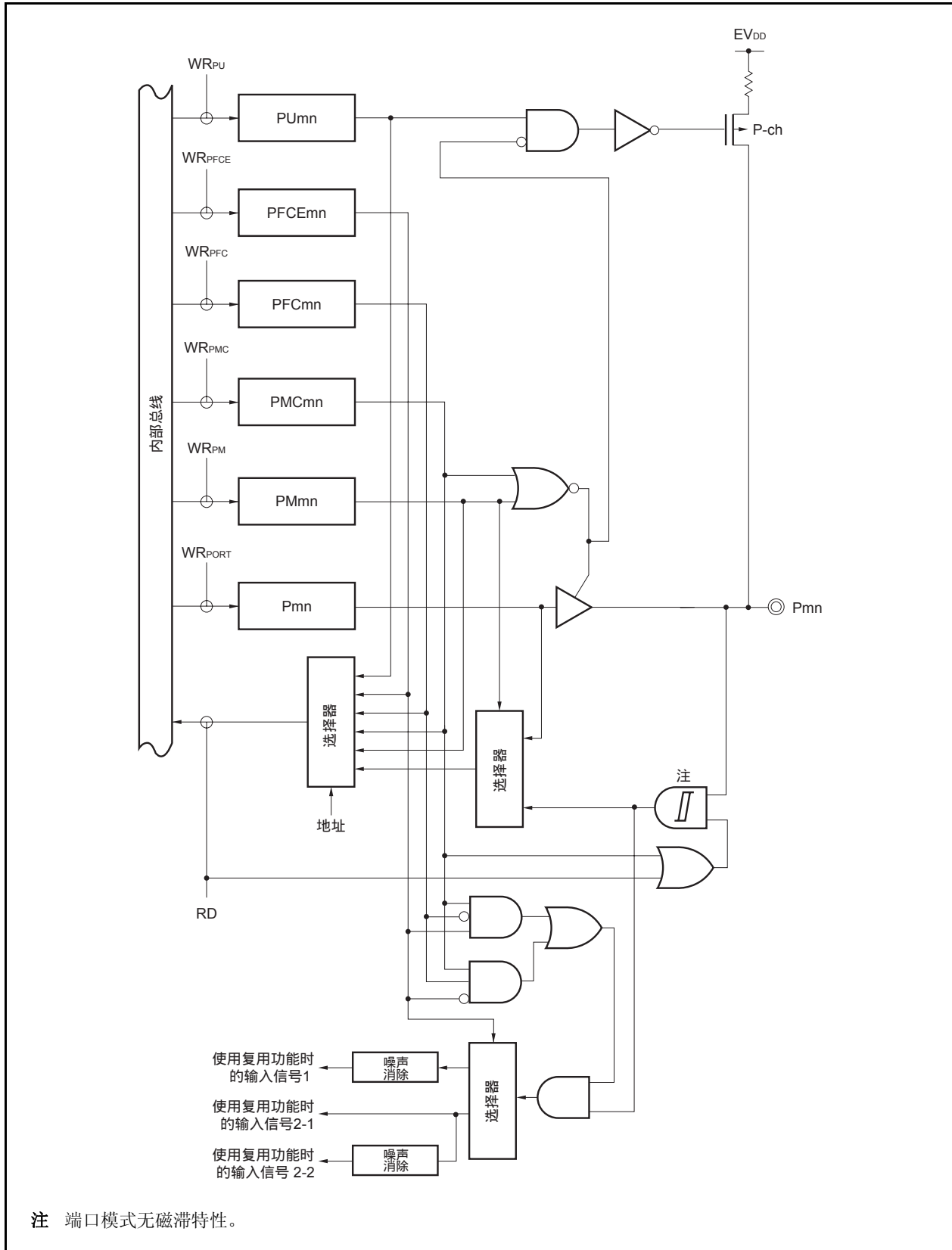


图 4-20. U-8 类型的结构图

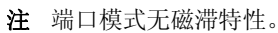


图 4-21. U-9 类型的结构图

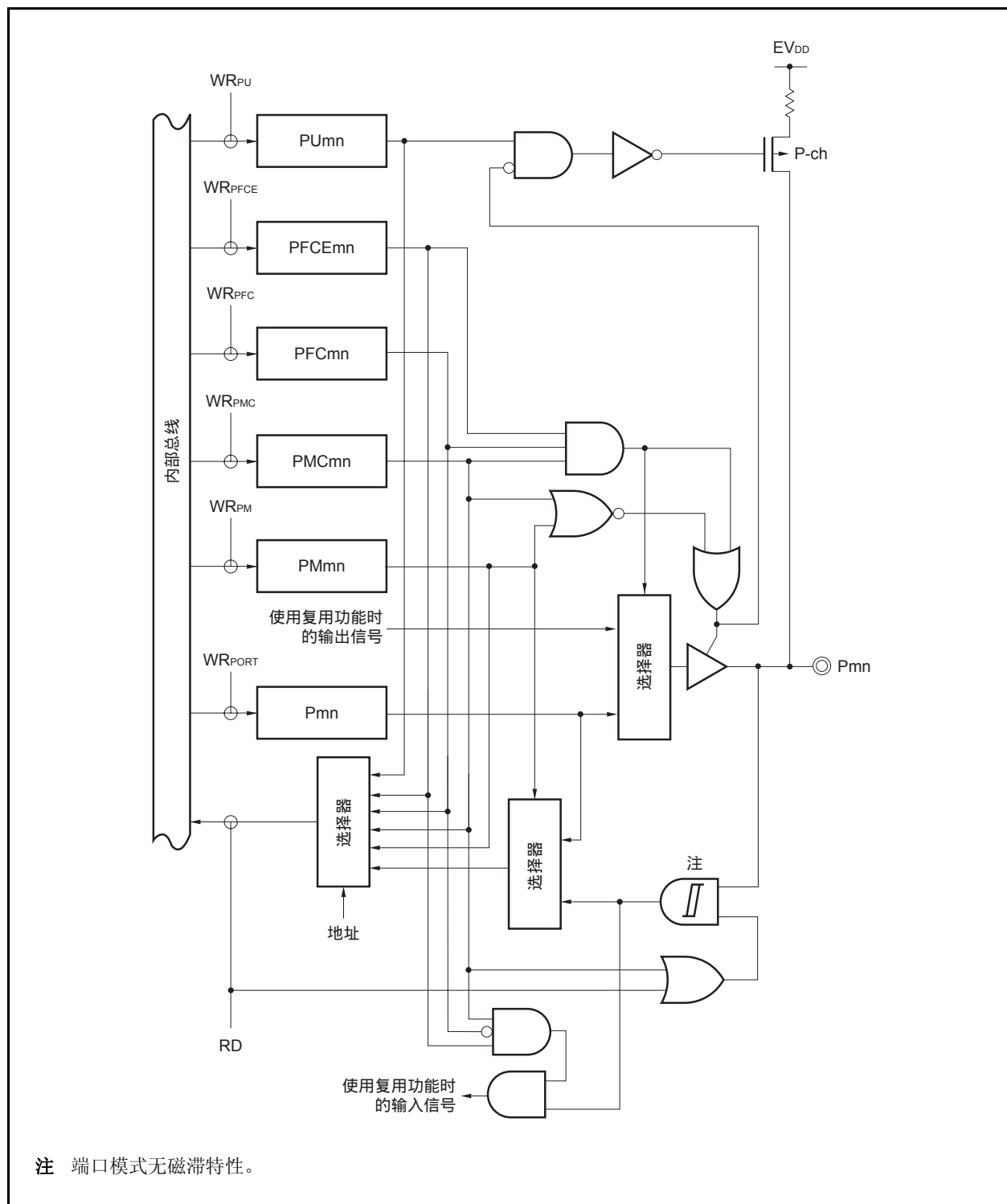


图 4-22. U-12 类型的结构图

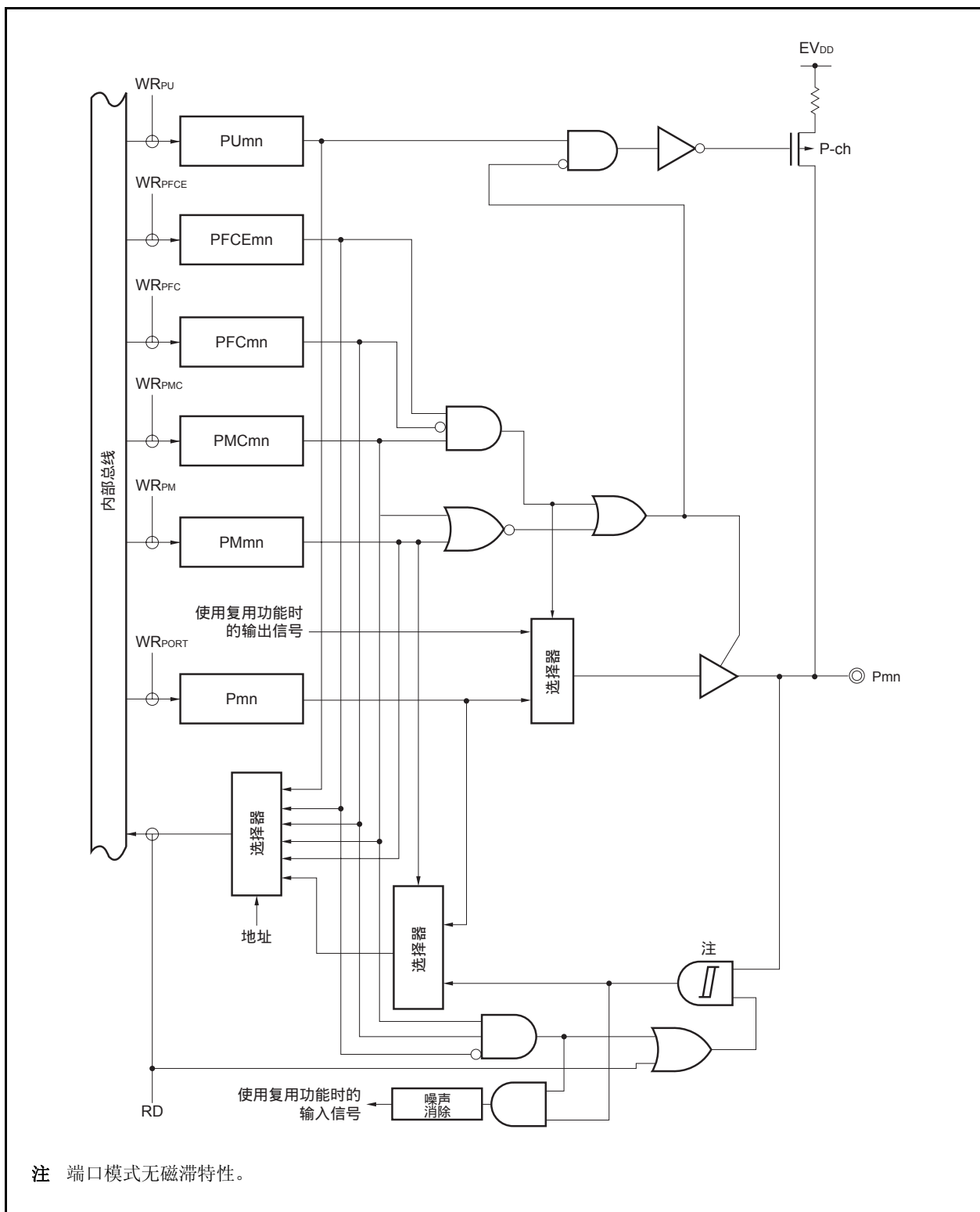


图 4-23. U-13 类型的结构图

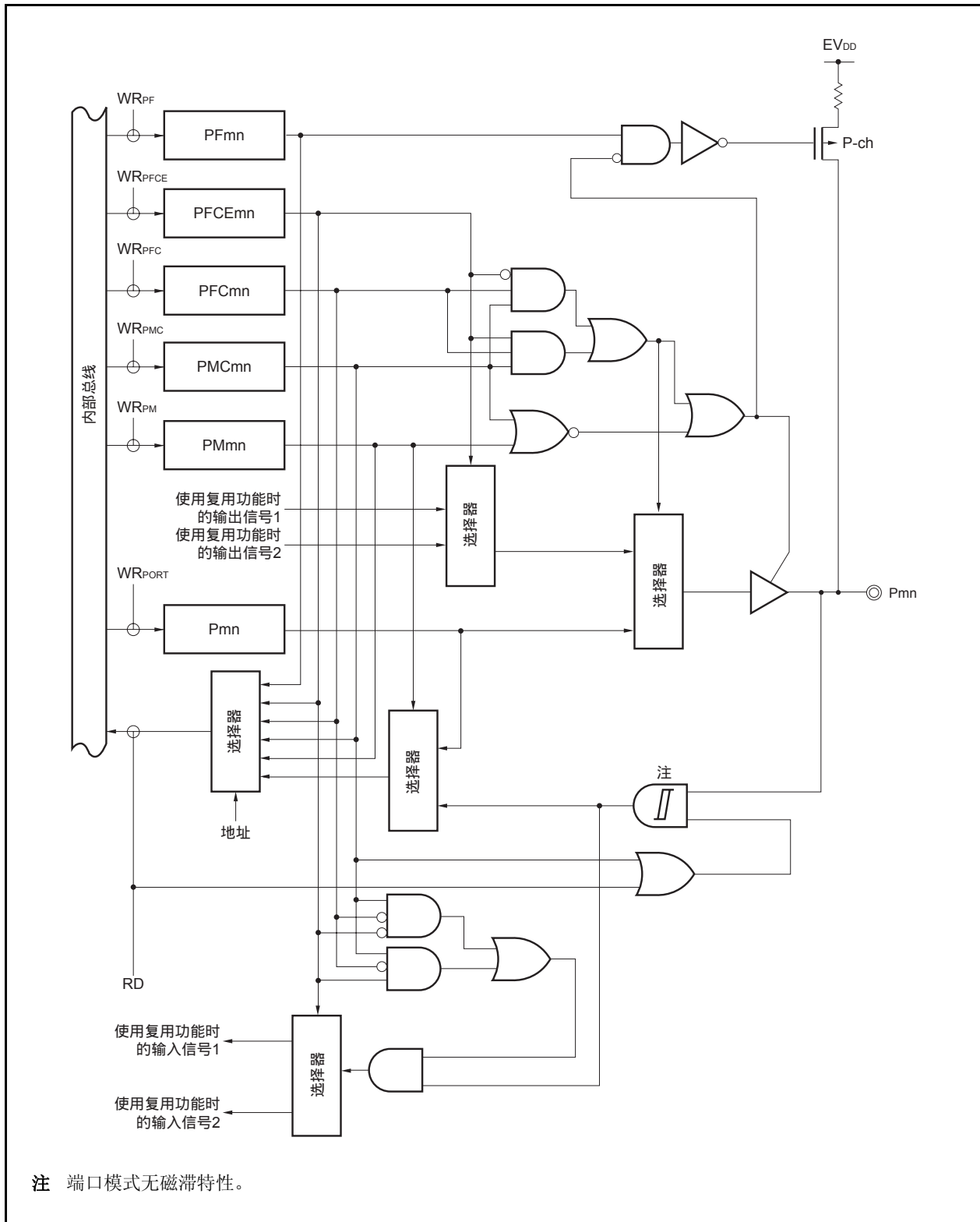


图 4-24. W-1 类型的结构图

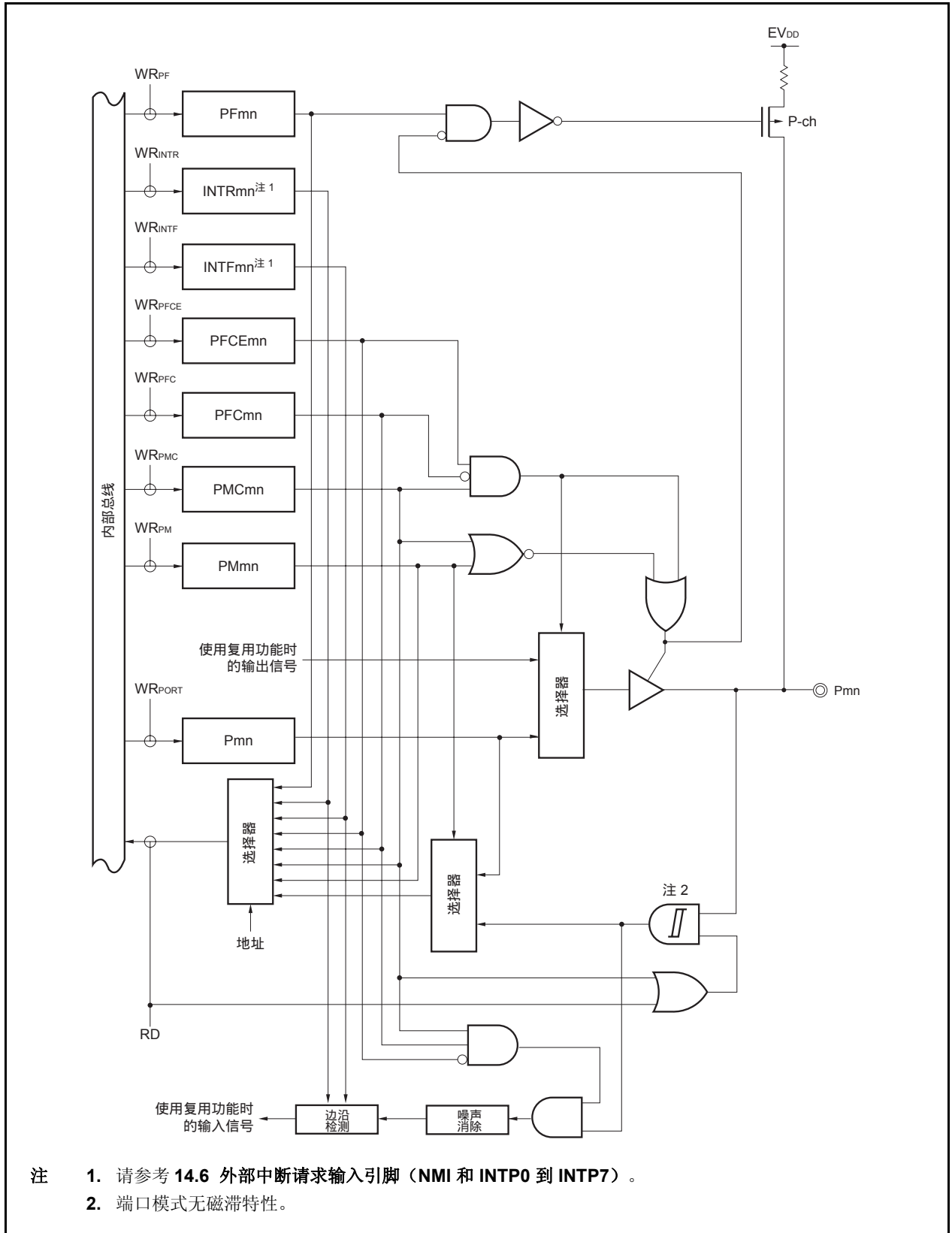
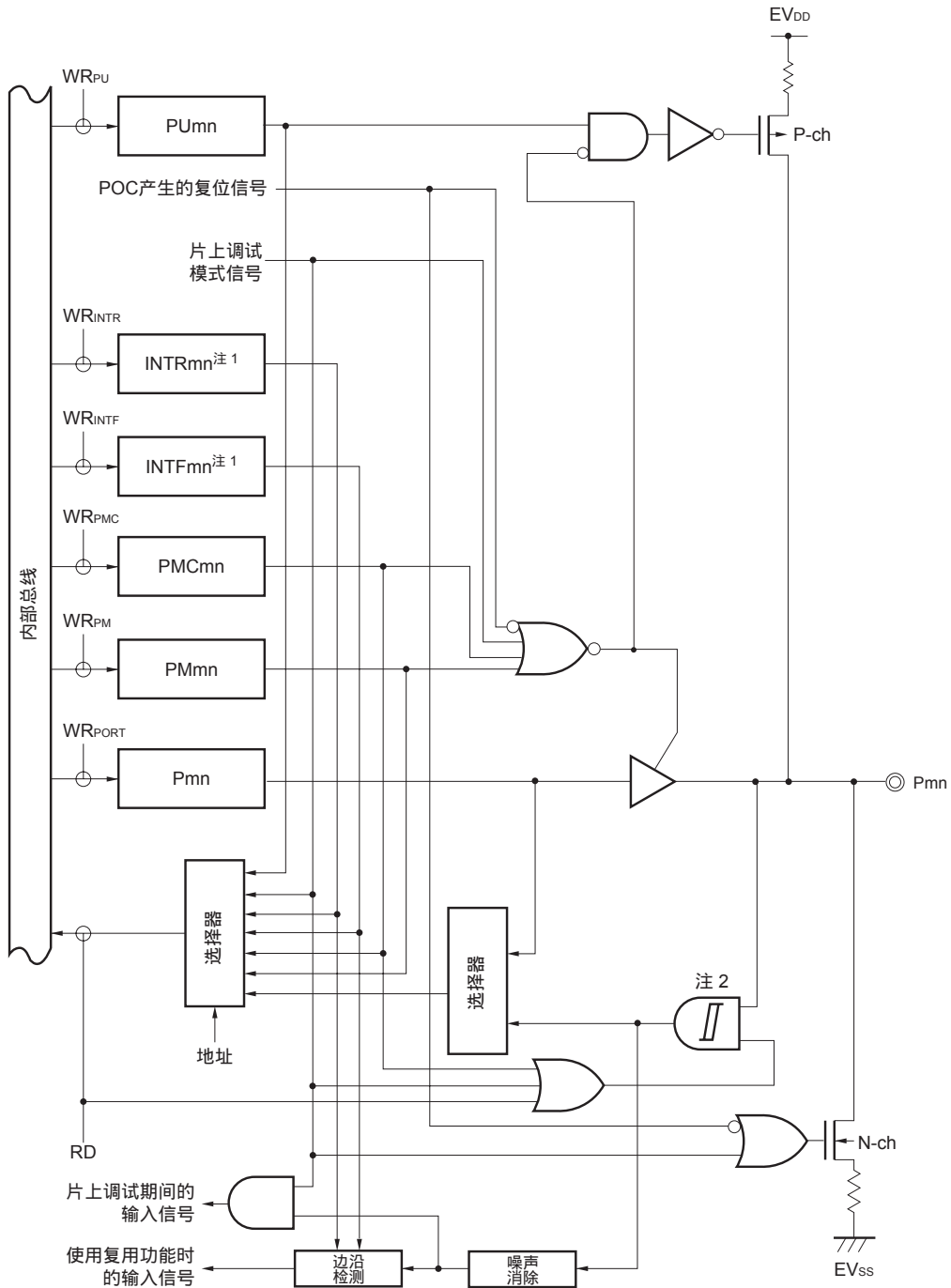


图 4-25. AA-1 类型的结构图



注

1. 请参考 14.6 外部中断请求输入引脚（NMI 和 INTP0 到 INTP7）。
2. 端口模式无磁滞特性。

4.5 注意事项

4.5.1 设置端口引脚的注意事项

(1) 在 V850ES/HE2 系列中，通用端口功能和一些周边功能 I/O 引脚共用一个引脚。要在通用端口（端口模式）和周边功能 I/O 引脚（复用功能模式）之间切换，可以通过设置 PMCn 寄存器来实现。关于此寄存器的设置顺序，需要注意以下几点问题。

(a) 从端口模式切换到复用功能模式的注意事项

要从端口模式切换到复用功能模式需要按照以下顺序执行。

- | | |
|-------------------------|-----------|
| <1>设置 PFCn 和 PFCEn 寄存器： | 复用功能选择 |
| <2>将 PMCn 寄存器相应的位数设为 1： | 切换到复用功能模式 |

如果先设置了 PMCn 寄存器，需要注意，在这个时候或者根据引脚状态的变化，可能会产生意想不到的作用，其中引脚状态的变化依照 PFCn，和 PFCEn 寄存器中的设置。

注意事项 无论是端口模式/复用功能模式，Pn 寄存器按以下方式读写。

- Pn 寄存器的读取：读取端口输出锁值（当 PMn.PMnm 位 = 0 时），或读取引脚状态（PMn.PMnm 位 = 1）。
- Pn 寄存器的写入：写入到端口输出锁。

<R>

(b) 复用功能模式（输入）的注意事项

当 PMCn.PMCnm 位为 0 时，由于 PMCn 寄存器的 AND 输出了设置值和引脚电平，因此输入到复用功能块的输入信号为低电平。因此，根据端口设置和复用功能操作使能定时，可能会产生不可预期的操作。因此，要按照以下顺序在端口模式和复用功能模式间切换。

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

5.1 概述

V850ES/HE2 产品提供以下时钟产生功能。

- 主时钟振荡器
 - 时钟直连模式
 $f_x = 4$ 至 5 MHz ($f_{xx} = 4$ 至 5 MHz)
 - PLL 模式中
 $f_x = 4$ 至 5 MHz ($f_{xx} = 16$ 至 20 MHz)
- 副时钟振荡器 (通过选项字节功能可以选择晶体振荡或 RC 振荡)
 - $f_{XT} = 32.768\text{ kHz}$ (晶体振荡器)
 - $f_{XT} = 20\text{ kHz}$ (RC 振荡器)
- PLL (锁相环) 倍频 ($\times 4$) 功能
 - 可选择时钟直连模式/PLL 模式
- 内置振荡器
 - $f_R = 200\text{ kHz}$ (TYP.)
- 内部系统时钟产生
 - 7 级 (f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/8$, $f_{xx}/16$, $f_{xx}/32$, f_{XT})
- 周边时钟产生功能
- 时钟输出功能
- 可编程时钟 (PCL) 输出功能

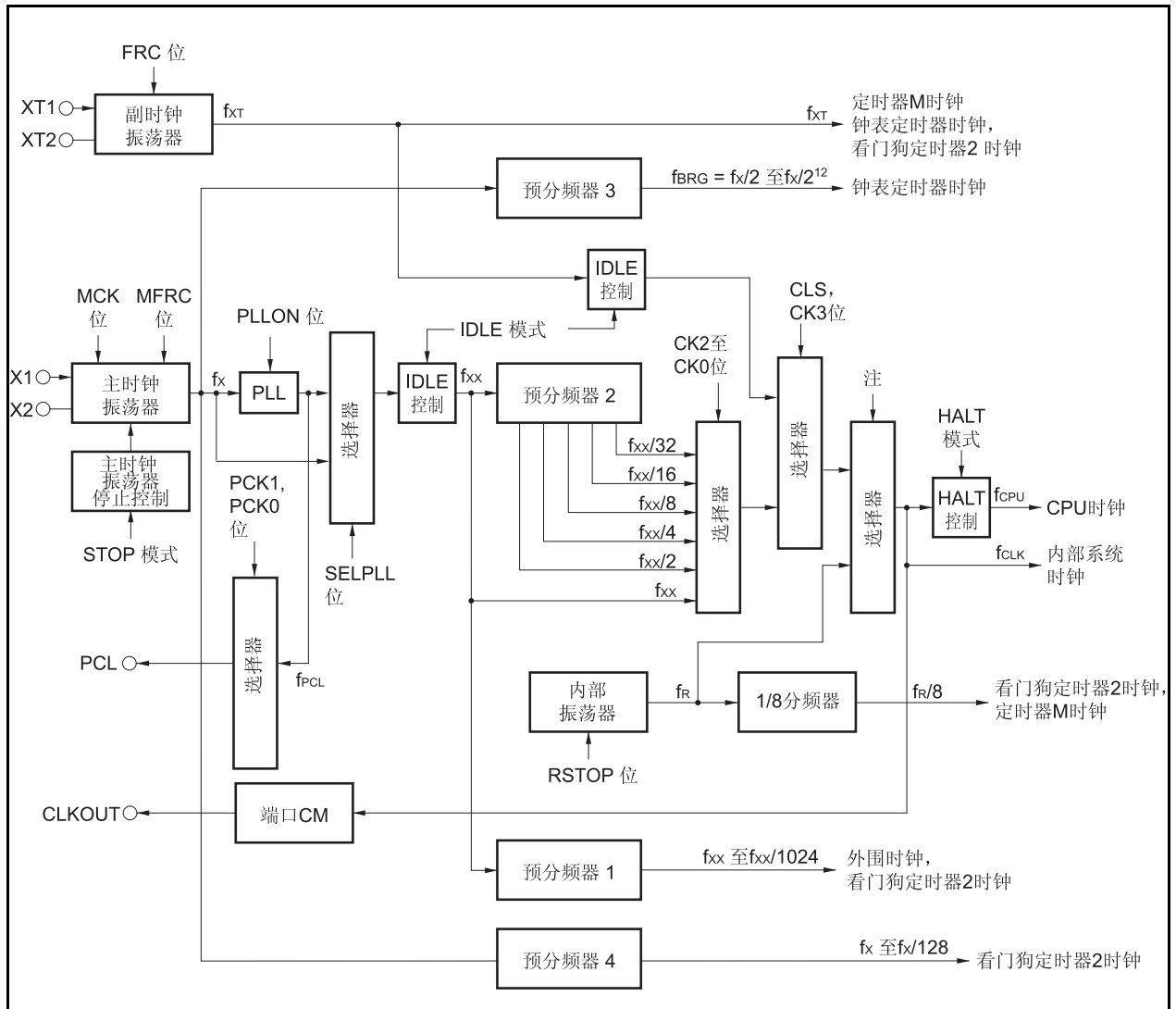
备注

- f_x : 主时钟振荡频率
- f_{xx} : 主时钟频率
- f_R : 内置振荡器时钟频率
- f_{XT} : 副时钟频率

5.2 结构图

<R>

图 5-1. 时钟发生器



注 当看门狗定时器 2 在振荡稳定过程中溢出时，内置振荡器时钟将被选中。

备注

- fx: 主时钟振荡频率
- fxx: 主时钟频率
- fCLK: 内部系统时钟频率
- fXT: 副时钟频率
- fCPU: CPU 时钟频率
- fBRG: 钟表定时器时钟频率
- fR: 内置振荡器时钟频率
- fPCL: 可编程频率

(1) 主时钟振荡器

主振荡器产生以下振荡频率 (f_x)。

- 时钟直连模式中
 $f_x = 4$ 至 5 MHz
- PLL 模式中
 $f_x = 4$ 至 5 MHz ($f_{xx} = 16$ 至 20 MHz)

(2) 副时钟振荡器

子振荡器产生频率为 32.768 kHz 或 20 kHz 的振荡信号 (f_{XT})。

(3) 主时钟振荡器停止控制电路

该电路生成一个可停止主时钟振荡器振荡的控制信号。

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

(4) 内置振荡器

产生一个频率为 220 kHz (TYP.) 的振荡信号 (f_R)。

(5) 预分频器 1

该电路生成提供给以下片上周边器件的时钟信号 (f_{xx} 至 $f_{xx}/1,024$)：TMP0 至 TMP3, TMQ0, TMM0, CSIB0, CSIB1, UARTA0, UARTA1, ADC 以及 WDT2。

(6) 预分频器 2

该电路对主时钟 (f_{xx}) 进行分频。

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

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

(7) 预分频器 3

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

详见第 9 章 钟表定时器功能。

(8) 预分频器 4

<R> 该电路生成提供给以下片上周边器件的时钟信号 (f_x 至 $f_x/128$)。
所提供的模块只有 WDT2。

(9) PLL

该电路对主时钟振荡器产生的时钟信号 (f_x) 进行 4 倍频。

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

5.3 寄存器

(1) 处理器时钟控制寄存器 (PCC)

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

可使用 8 位或 1 位的操作对该寄存器进行读写。

复位输入将把该寄存器置为 03H。

复位后：03H R/W 地址：FFFFF828H

	7	6	5	4	3	2	1	0
PCC	FRC	MCK	MFRC	CLS ^注	CK3	CK2	CK1	CK0

FRC	子时钟片上反馈电阻的使用
0	使用
1	不使用

MCK	主时钟振荡器控制
0	允许振荡
1	停止振荡

- 当系统通过作为CPU时钟的主时钟运行时，即使MCK位被设为（1），主时钟的操作也不会停止。它只有在CPU时钟变为子时钟时才会停止。
- 在将MCK位从0设为1前，应停止通过主时钟实行的片上外围功能。
- 主时钟停止后，器件将工作于副时钟，若将CPU时钟或周边器件的工作时钟切换回主时钟，则需要将MCK位清零并通过软件确保振荡稳定时间。

MFRC	主时钟片上反馈电阻的使用
0	使用
1	不使用

CLS ^注	CPU 时钟的状态 (f _{CPU})
0	工作于主时钟
1	工作于副时钟

CK3	CK2	CK1	CK0	时钟选择 (f _{CLK} /f _{CPU})
0	0	0	0	f _{xx}
0	0	0	1	f _{xx} /2
0	0	1	0	f _{xx} /4
0	0	1	1	f _{xx} /8
0	1	0	0	f _{xx} /16
0	1	0	1	f _{xx} /32
0	1	1	·	禁止设置
1	·	·	·	f _{XT}

(a) 从主时钟操作切换至副时钟操作的举例

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

- 注意事项** 1. 当停止主时钟工作时，也要同时停止 PLL。另外，工作于主时钟频率的片上周边器件也被停止。
2. 如果不满足下面的条件，那么请改变 CK2 到 CK0 位的设置以满足该条件，然后再切换到副时钟工作模式。

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

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

[示例]

```
<1> _SET_SUB_RUN :
    st.b      r0, PRCMD[r0]
    set1      3, PCC[r0]           -- CK3 位  $\leftarrow$  1
<2> _CHECK_CLS :
    tst1      4, PCC[r0]           --等待副时钟工作的开始。
    bz        _CHECK_CLS
<3> _STOP_MAIN_CLOCK :
    st.b      r0, PRCMD[r0]
    set1      6, PCC[r0]           -- MCK 位  $\leftarrow$  1，主时钟停止。
```

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

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

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

```

<1> _START_MAIN_OSC :
    st.b          r0, PRCMD[r0]          --解除对特殊寄存器的保护
    clr1          6, PCC[r0]             --启动主时钟振荡
<2> movea         0x55, r0, r11          --等待振荡稳定时间
    _WAIT_OST :
    nop
    nop
    nop
    addi          -1, r11, r11
    mp            r0, r11
    bne           _PROGRAM_WAIT
<3> st.b          r0, PRCMD[r0]
    clr1          3, PCC[r0]             -- CK3 ← 0
<4> _CHECK_CLS :
    tst1          4, PCC[r0]             --等待主时钟开始工作
    bnz           CHECK_CLS

```

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

(2) 内置振荡模式寄存器 (RCM)

RCM 寄存器是用于设置内置振荡器工作模式的 8 位寄存器。
可以对该寄存器进行 8 位或 1 位的读写操作。
复位输入将把该寄存器置为 00H。

复位后： 00H R/W 地址： FFFFF80CH							
RCM	7	6	5	4	3	2	1
	0	0	0	0	0	0	0
	RSTOP						
RSTOP		振荡/停止内部振荡器					
0		内部振荡器振荡					
1		内部振荡器停止振荡					

(3) CPU 操作时钟状态寄存器 (CCLS)

CCLS 寄存器是用于指示 CPU 操作时钟状态的寄存器。
只能对该寄存器进行 8 位或 1 位的读取操作。
复位数据将把该寄存器置为 00H。

复位后： 00H ^注 R 地址： FFFFF82EH							
CCLS	7	6	5	4	3	2	1
	0	0	0	0	0	0	0
	CCLS F						
CCLS F		CPU操作时钟状态					
0		工作于主时钟（f _x ）或副时钟（f _{XT} ）。					
1		工作于内部振荡时钟（f _R ）。					

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

5.4 操作

5.4.1 各时钟的操作

下表显示了各时钟的操作状态。

表 5-1. 各时钟的操作状态

目标时钟	寄存器设置 以及操作状态	PCC 寄存器								
		CLK 位 = 0, MCK 位 = 0					CLS 位 = 1, MCK 位 = 0		CLS 位 = 1, MCK 位 = 1	
		复位 过程中	振荡稳定 时间等待 过程中	HALT 模 式	IDLE1, IDLE2 模 式	STOP 模 式	副时钟模 式	Sub-IDLE 模式	副时钟模 式	Sub-IDLE 模式
<R>	主时钟振荡器 (f _x)	×	○	○	○	×	○	○	×	×
	主系统时钟 (f _{xx})	×	×	○	×	×	○	×	×	×
	副时钟振荡器 (f _{XT})	○	○	○	○	○	○	○	○	○
	CPU 时钟 (f _{CPU})	×	×	×	×	×	○	×	○	×
	内部系统时钟 (f _{CLK})	×	×	○	×	×	○	×	○	×
	主时钟 (PLL 模式中, f _{xx})	×	注 1	○	注 2	×	○	○	×	×
	周边时钟 (f _{xx} 至 f _{xx} /1,024)	×	×	○	×	×	○	×	×	×
	WT 时钟 (主)	×	×	○	○	×	○	○	×	×
	WT 时钟 (副)	○	○	○	○	○	○	○	○	○
	WDT2 时钟 (内置振荡时钟)	×	○	○	○	○	○	○	○	○
	WDT2 时钟 (主)	×	×	○	×	×	○	×	×	×

- 注
1. 振荡在振荡稳定时间过去一半后开始，且稳定时钟会在入锁时间后提供。
 2. 在 IDLE1 模式中可操作。在 IDLE2 模式中会被停止。

备注

○: 可操作

×: 停止

5.4.2 时钟输出功能

时钟输出功能用于将内部系统时钟 (f_{clk}) 从 CLKOUT 引脚输出。

内部系统时钟 (f_{clk}) 可通过 PCC.CK3 到 PCC.CK0 位进行选择。

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

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

5.5 PLL 功能

5.5.1 概述

在 V850ES/HE2 系列产品中，CPU 和片上周边器件的时钟信号可以选择频率为晶振频率的 4 倍的 PLL 模式和时钟直连模式。

使用 PLL 功能时： 输入时钟 = 4 至 5 MHz（输出：16 至 20 MHz）
 时钟直连模式： 输入时钟 = 4 至 5 MHz（输出：4 至 5 MHz）

5.5.2 寄存器

(1) PLL 控制寄存器 (PLLCTL)

PLLCTL 寄存器是用来控制 PLL 功能的 8 位寄存器。

可以对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器置为 01H。

复位后：01H R/W 地址：FFFFF82CH							
PLLCTL	7	6	5	4	3	2	1
	0	0	0	0	0	0	SELPLL PLLON
PLLON		PLL操作停止寄存器					
0		PLL 停止					
1		PLL 运行 (PLL操作开始后, 需要一个入锁时间来保证振荡频率的稳定)					
SELPLL		CPU操作时钟选择寄存器					
0		时钟直通模式					
1		PLL模式					

注意事项 1. 当 PLLON 位被清零时，SELPLL 位也会被自动清零（进入时钟直连模式）。

2. 只有在 PLL 时钟频率稳定后，SELPLL 位才能够被置 1。如果时钟频率没有稳定（未入锁），那么无论向该位写入的数据为何值，SELPLL 位都会被写入“0”。

(2) 锁定寄存器 (LOCKR)

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

包含 LOCK 位的 LOCKR 寄存器用于反映 PLL 频率稳定状态。

只能对该寄存器进行 8 位或 1 位的读取操作。

复位输入将把该寄存器置为 00H。

复位后： 00H R 地址： FFFFF824H

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

LOCK	检查PLL锁定状态
0	锁定状态
1	未锁定状态

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

[置位条件]

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

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

[清零条件]

- 复位释放后振荡稳定时间（OSTS 寄存器的默认时间（见 16.2 （3） 振荡稳定时间选择寄存器（OSTS）））溢出时。
- 在 PLL 工作状态下设置了 STOP 模式，STOP 模式解除后，振荡稳定定时器溢出时（时间由 OSTS 寄存器设置）。
- 当 PLLCTL.PLLON 位的设置从 0 改变为 1，PLL 入锁时间定时器产生溢出时（时间由 PLLS 寄存器设置）。
- 在 PLL 工作状态下设置了 IDLE2 模式，IDLE2 模式解除时，所插入的设置时间（由 OSTS 寄存器设置的时间）结束后。

(3) PLL 入锁时间指定寄存器 (PLLS)

PLLS 寄存器是用来选择在将 PLLCTL.PLLON 位从 0 变为 1 时的 PLL 入锁时间的 8 位寄存器。
可以对该寄存器进行 8 位的读写操作。
复位输入将把该寄存器设置为 03H。

复位后： 03H R/W 地址： FFFFF6C1H

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

PLLS1	PLLS0	PLL锁定时间的选择
0	0	$2^{10}/f_x$
0	1	$2^{11}/f_x$
1	0	$2^{12}/f_x$
1	1	$2^{13}/f_x$ (默认值)

- 注意事项 1. 将入锁时间设置为 800 μs 或更长。
 2. 入锁过程中不要改变 PLLS 寄存器的设置。

备注 f_x : 主时钟振荡频率

(4) 可编程时钟模式寄存器 (PCLM)

PCLM 寄存器是一个用于控制 PCL 输出的 8 位寄存器。
可以对该寄存器进行 8 位或 1 位的读写操作。

复位后： 00H R/W 地址： FFFFF82FH							
	7	6	5	4	3	2	1 0
PCLM	0	0	0	PCLE	0	0	PCK1 PCK0

PCLE	PCL引脚输出操作的选择
0	禁止PCL引脚输出（PCL引脚被固定为低电平）
1	允许PCL引脚输出

注意事项 先设置与端口相关的控制寄存器（PM，PMC，PFC 以及 PFCE 寄存器等），然后将 PCLE 位设为 1。

<R>

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

注意事项 仅在 PLL 操作过程中将 PCLE 位设为 1。要停止 PLL，则将 PCLE 位清零。

备注 f_{PCL}： 可编程频率

5.5.3 使用方法

(1) 使用 PLL 时

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

(a) 当从始终直连模式切换到 IDLE2 或 STOP 模式时

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

(b) 在保持 PLL 运行模式时转移到 IDLE 2 或 STOP 模式时

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

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

(2) 不使用 PLL 时

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

第 6 章 16 位定时器/事件计数器 P (TMP)

定时器 P (TMP) 是一个 16 位的定时器/事件计数器。

V850ES/HE2 具有 4 个定时器/事件计数器通道, TMP0 到 TMP3。

6.1 概述

以下是 TMPn 的概况。

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

备注 n = 0 至 3

6.2 功能

TMPn 具有以下功能。

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

备注 n = 0 至 3

6.3 配置

TMPn 包含以下硬件。

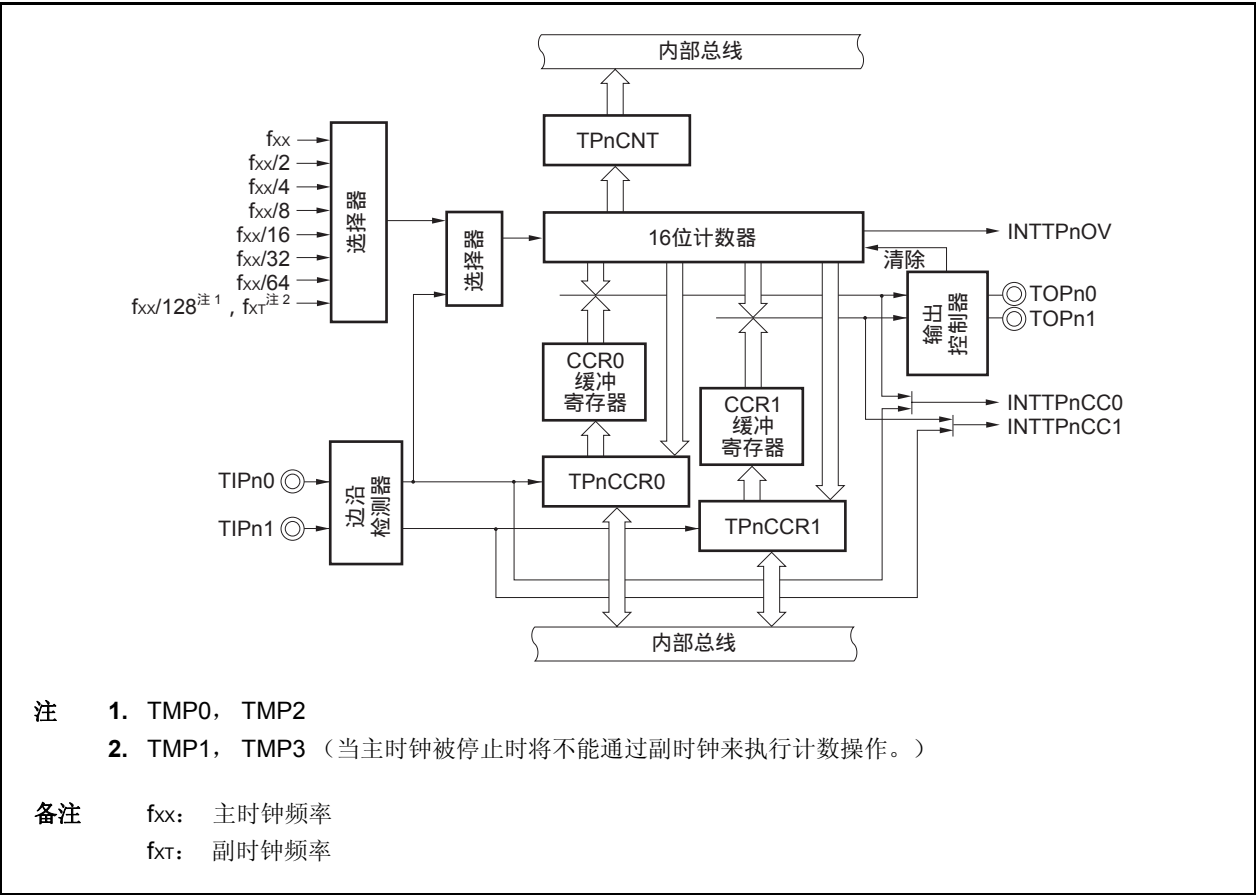
表 6-1. TMPn 的配置

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

注 1. TIPn0 引脚的复用功能包括捕捉触发输入信号，外部事件计数输入信号和外部触发输入信号。
2. 使用 TIPn0, TIPn1, TOPn0 和 TOPn1 引脚功能时，请参照表 4-12 端口引脚作为复用引脚使用。

备注 n = 0 至 3

图 6-1. TMPn 的框图



(1) 16 位计数器

该 16 位计数器可以对内部时钟或外部事件进行计数。

可以使用 TPnCNT 寄存器读取该计数器的计数值。

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

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

(2) CCR0 缓冲寄存器

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

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

CCR0 缓冲寄存器不能被直接读/写。

复位后 CCR0 缓冲寄存器和 TPnCCR0 寄存器都将被清零。

(3) CCR1 缓冲寄存器

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

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

CCR1 缓冲寄存器不能被直接读/写。

复位后 CCR0 缓冲寄存器和 TPnCCR0 寄存器都将被清零。

(4) 边沿检测器

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

(5) 输出控制器

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

(6) 选择器

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

6.4 寄存器

用于控制 TMPn 的寄存器如下所述。

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

备注

1. 使用 TIPn0, TIPn1, TOPn0 和 TOPn1 引脚功能时, 请参照表 4-12 端口引脚作为复用引脚使用。
2. n = 0 至 3

(1) TMPn 控制寄存器 0 (TPnCTL0)

TPnCTL0 寄存器是用于控制 TMPn 操作的 8 位寄存器。

可以对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

可使用软件向 TPnCTL0 寄存器写入相同的数值。

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

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

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

TPnCKS2	TPnCKS1	TPnCKS0	内部计数时钟选择	
			n = 0, 2	n = 1, 3
0	0	0	f _{xx}	
0	0	1	f _{xx} /2	
0	1	0	f _{xx} /4	
0	1	1	f _{xx} /8	
1	0	0	f _{xx} /16	
1	0	1	f _{xx} /32	
1	1	0	f _{xx} /64	
1	1	1	f _{xx} /128	f _{XT} ^{注2}

注 1. TPnOPT0.TPnOVF 位, 16 位计数器, 定时器输出 (TOPn0, TOPn1 引脚)。

2. 当主时钟被停止时将不能通过副时钟来执行计数操作。

注意事项 1. 在 TPnCE 位 = 0 时才可对 TPnCKS2 到 TPnCKS0 位进行设置。
当将 TPnCE 位的值从 0 改变为 1 时, 可同时对 TPnCKS2 到 TPnCKS0 位进行设置。

2. 请务必将第 3 到第 6 位清除为 “0”。

备注 f_{xx}: 主时钟频率
f_{XT}: 副时钟频率

(2) TMPn 控制寄存器 1 (TPnCTL1)

TPnCTL1 寄存器是用于控制 TMPn 操作的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

(1/2)

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

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

TPnSYE	调谐操作模式允许控制														
0	独立操作模式（异步操作模式）														
1	调谐操作模式（从操作的指定） 模式中，定时器P可以与主定时器同步操作。 <table><tr><td colspan="2">主定时器</td><td colspan="2">从定时器</td></tr><tr><td>TMP0</td><td colspan="2">TMP1</td><td>-</td></tr><tr><td>TMP2</td><td colspan="2">TMP3</td><td>TMQ0</td></tr></table> <p>关于调谐操作模式的详细信息，参见 6.6 定时器调谐操作功能。</p> <p>注意事项 确保将TP0SYE及TP2SYE位清零。</p>			主定时器		从定时器		TMP0	TMP1		-	TMP2	TMP3		TMQ0
主定时器		从定时器													
TMP0	TMP1		-												
TMP2	TMP3		TMQ0												

TPnEST	软件触发控制	
0	-	
1	为外部触发输入生成一个有效信号。 <ul style="list-style-type: none">在单脉冲输出模式中： 通过将1写入TPnEST位，单脉冲被输出为触发信号。在外部触发脉冲输出模式中： 通过将1写入TPnEST位，PWM波形被输出为触发信号。	

- 注意事项 1. 对 TPnEST 位的设置只有在外部触发脉冲输出模式或单脉冲输出模式才有效。在其它模式下，对该位的置 1 操作将被忽略。
2. 请务必将第 3, 4 和 7 位清除为“0”。

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

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

- 注意事项**
1. 在外部事件计数模式下，无论 TPnEEE 位如何设置，外部事件计数输入都是被选中的。
 2. 在 TPnCTL0.TPnCE 位 = 0 时才可对 TPnEEE 和 TPnMD2 到 TPnMD0 位进行设置 (TPnCE 位 = 1 时只可以对这些位写入相同的值)。如果在 TPnCE 位 = 1 时对这些位进行了改写，那么操作结果将不能被保证。若由于误操作引起了改写，则要将 TPnCE 位清零然后再次设置这些寄存器位。

<R>

(3) TMPn I/O 控制寄存器 0 (TPnIOC0)

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

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

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

7

6

5

4

3

2

1

0

TPnIOC0

(n = 0至3)

0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0
---	---	---	---	--------	--------	--------	--------

TPnOL1	TOPn1引脚输出电平设置 ^注
0	TOPn1引脚输出从高电平处开始
1	TOPn1引脚输出从低电平处开始

TPnOE1	TOPn1引脚输出设置
0	禁止定时器输出 · TPnOL1位 = 0时：低电平从TOPn1引脚中输出 · TPnOL1位 = 1时：高电平从TOPn1引脚中输出
1	允许定时器输出（从TOPn1引脚中输出一个方波）。

TPnOL0	TOPn0引脚输出电平设置 ^注
0	TOPn0引脚输出从高电平处开始
1	TOPn0引脚输出从低电平处开始

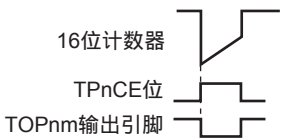
TPnOE0	TOPn0引脚输出设置
0	禁止定时器输出 · TPnOL0位 = 0时：低电平从TOPn0引脚中输出 · TPnOL0位 = 1时：高电平从TOPn0引脚中输出
1	允许定时器输出（从TOPn0引脚中输出一个方波）。

注 由 TPnOLm 位指定的定时器输出管脚 (TOPnm) 输出电平如下所示。

· TPnOLm位= 0时



· TPnOLm位 = 1时



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

备注 n = 0 至 3, m = 0, 1

(4) TMPn I/O 控制寄存器 1 (TPnIOC1)

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

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

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

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

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

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

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

(5) TMPn I/O 控制寄存器 2 (TPnIOC2)

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

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

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

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

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

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

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

(6) TMPn 选项寄存器 0 (TPnOPT0)

TPnOPT0 寄存器是用于设置捕捉/比较操作和检测溢出的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后：00H R/W 地址： TP0OPT0 FFFFF595H, TP1OPT0 FFFFF5A5H, TP2OPT0 FFFFF5B5H, TP3OPT0 FFFFF5C5H

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

TPnCCS1	TPnCCR1寄存器捕捉/比较功能选择
0	选择比较寄存器功能
1	选择捕捉寄存器功能
TPnCCS1位的设置仅在自由运行定时器模式下有效。	

TPnCCS0	TPnCCR0寄存器捕捉/比较功能选择
0	选择比较寄存器功能
1	选择捕捉寄存器功能
TPnCCS0位的设置仅在自由运行定时器模式下有效。	

TPnOVF	TMPn溢出检测标志
设置 (1)	产生溢出
复位 (0)	对TPnOVF位0进行写操作或TPnCTL0.TPnCE位 = 0
<ul style="list-style-type: none"> 在自由运行定时器模式或脉宽测量模式下，16位计数器计数值产生溢出（从FFFFH到0000H）时，TPnOVF位会被重置。 TPnOVF位被置1的同时，会产生一个中断请求信号（INTTPnOV）但除自由运行定时器模式和脉宽测量模式外的其他模式下不会产生INTTPnOV信号。 即使在TPnOVF位 = 1时读取TPnOVF位或TPnOPT0寄存器，TPnOVF位也不会被清除。 可以对TPnOVF位进行读写操作，但无法使用软件将该位设为1。对该位写入1的操作不会对TMPn的工作造成任何影响。 	

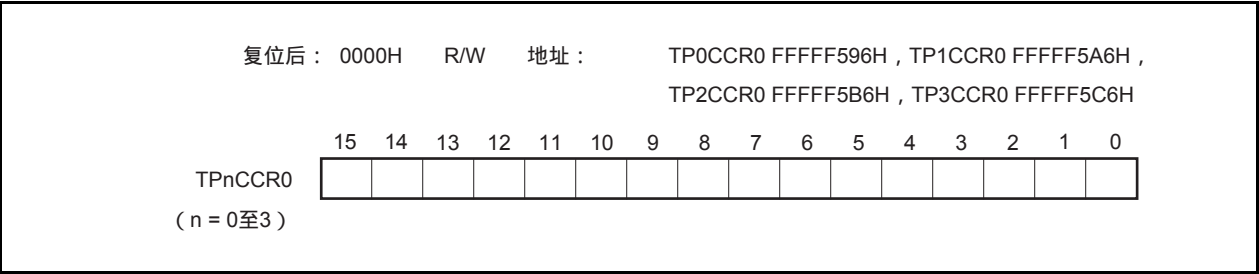
- 注意事项**
1. 在 TPnCTL0.TPnCE 位 = 0 时才可对 TPnCCS1 和 TPnCCS0 位进行改写（TPnCE 位 = 1 时只可以向这些位写入相同的值）。若在 TPnCE 位 = 1 由于误操作引起了改写，则要将 TPnCE 位清零然后再次设置这些寄存器位。
 2. 请务必将第 1 到 3 位，第 6 和第 7 位清除为“0”。

(7) TMPn 捕捉/比较寄存器 0 (TPnCCR0)

根据选择模式的不同，TPnCCR0 可用作捕捉寄存器或比较寄存器。
该寄存器只有在自由运行模式下根据 TPnOPT0.TPnCCS0 位设置的不同，可以用作捕捉寄存器或比较寄存器。
在脉宽测量模式下，TPnCCR0 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。
TPnCCR0 寄存器可以在操作过程中进行读写。
可对该寄存器进行 16 位的读写操作。
复位输入将把该寄存器设置为 0000H。

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

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



(a) 作为比较寄存器时的功能

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

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

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

(b) 作为捕捉寄存器时的功能

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

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

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 6-2. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

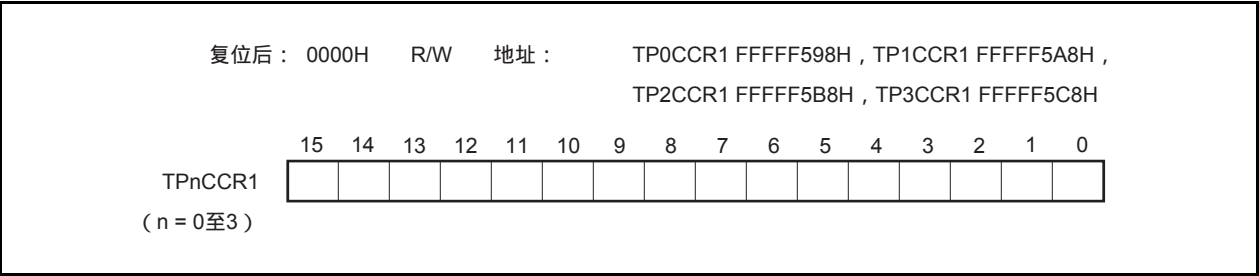
工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	—

(8) TMPn 捕捉/比较寄存器 1 (TPnCCR1)

根据选择模式的不同，TPnCCR1 可用作捕捉寄存器或比较寄存器。
该寄存器只有在自由运行模式下根据 TPnOPT0.TPnCCS1 位设置的不同，可以用作捕捉寄存器或比较寄存器。
在脉宽测量模式下，TPnCCR1 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。
TPnCCR1 寄存器可以在操作过程中进行读写。
可对该寄存器进行 16 位的读写操作。
复位输入将把该寄存器设置为 0000H。

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

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



(a) 作为比较寄存器时的功能

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

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

(b) 作为捕捉寄存器时的功能

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

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

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 6-3. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

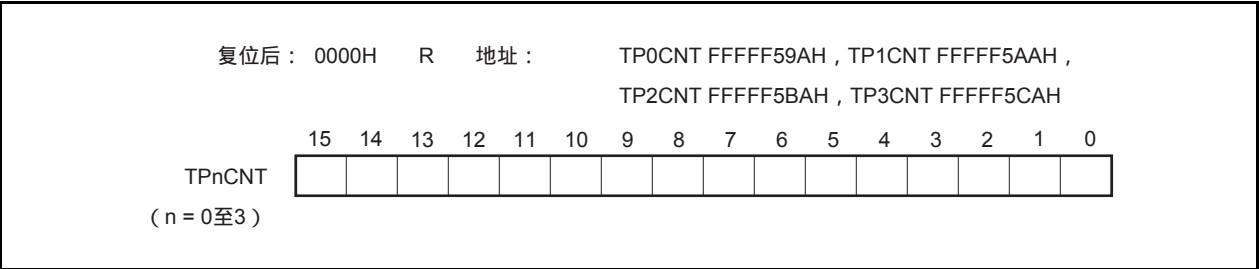
工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	—

(9) TMPn 计数器读取缓冲寄存器 (TPnCNT)

可以通过读取缓冲寄存器 TPnCNT 来读取 16 位计数器的计数值。
如果该寄存器在 TPnCTL0.TPnCE 位 = 1 时被读取，那么读取的值就是 16 位计数器的计数值。
只可以对该寄存器进行 16 位的读取操作。
当 TPnCE 位 = 0 时，TPnCNT 寄存器将被清零。如果此时对 TPnCNT 寄存器进行读取，读取到的值将是 0000H 而不是 16 位计数器的值 (FFFFH)。
复位后，TPnCE 位被清零的同时 TPnCNT 寄存器的值也被清零。

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

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



(10) TIPnm 引脚噪声消除控制寄存器 (PnmNFC)

PnmNFC 寄存器是一个为消除噪声而设置定时器 P 输入引脚的数字噪声滤波器的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后: 00H R/W 地址: P00NFC : FFFFFB00H (TIP00 引脚)
 P01NFC : FFFFFB04H (TIP01 引脚)
 P10NFC : FFFFFB08H (TIP10 引脚)
 P11NFC : FFFFFB0CH (TIP11 引脚)
 P20NFC : FFFFFB10H (TIP20 引脚)
 P21NFC : FFFFFB14H (TIP21 引脚)
 P30NFC : FFFFFB18H (TIP30 引脚)
 P31NFC : FFFFFB1CH (TIP31 引脚)

	7	6	5	4	3	2	1	0
PnmNFC	0	NFSTS	0	0	0	NFC2	NFC1	NFC0

(n = 0 至 3, m =
0, 1)

NFSTS	通过数字噪声滤波器执行的采样次数的设置
0	3 次
1	2 次

NFC2	NFC1	NFC0	采样时钟	
			n = 0, 2	n = 1, 3
0	0	0	f_{xx}	
0	0	1	$f_{xx}/2$	
0	1	0	$f_{xx}/4$	
0	1	1	$f_{xx}/16$	$f_{xx}/8$
1	0	0	$f_{xx}/32$	$f_{xx}/16$
1	0	1	$f_{xx}/64$	f_{XT}
与以上不同			禁止设置	

- 注意事项**
1. 请务必将位 3 至位 5 以及位 7 清除为“0”。
 2. 在设置 PnmNFC 寄存器前, 输入到定时器输入引脚 (TIPnm) 的信号会通过消除的数字噪声来输出。
 因此, 先通过使用 PnmNFC 寄存器来设置采样时钟 (NFC2 至 NFC0) 以及采样的次数 (NFSTS), 然后等待初始化时间 = (采样时钟) × (采样的次数), 再允许定时器运行。

备注 可以被精确消除的噪声的宽度等于 (采样时钟) × (采样的次数 - 1)。如果宽度窄于该噪声宽度的噪声与采样时钟同步, 那么这种噪声可能会引起计算错误。

6.5 操作

TMPn 可进行以下操作。

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

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

备注 n = 0 至 3

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

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

通常情况下，间隔定时器模式中不使用 TPnCCR1 寄存器。

图 6-2. 间隔定时器的配置图

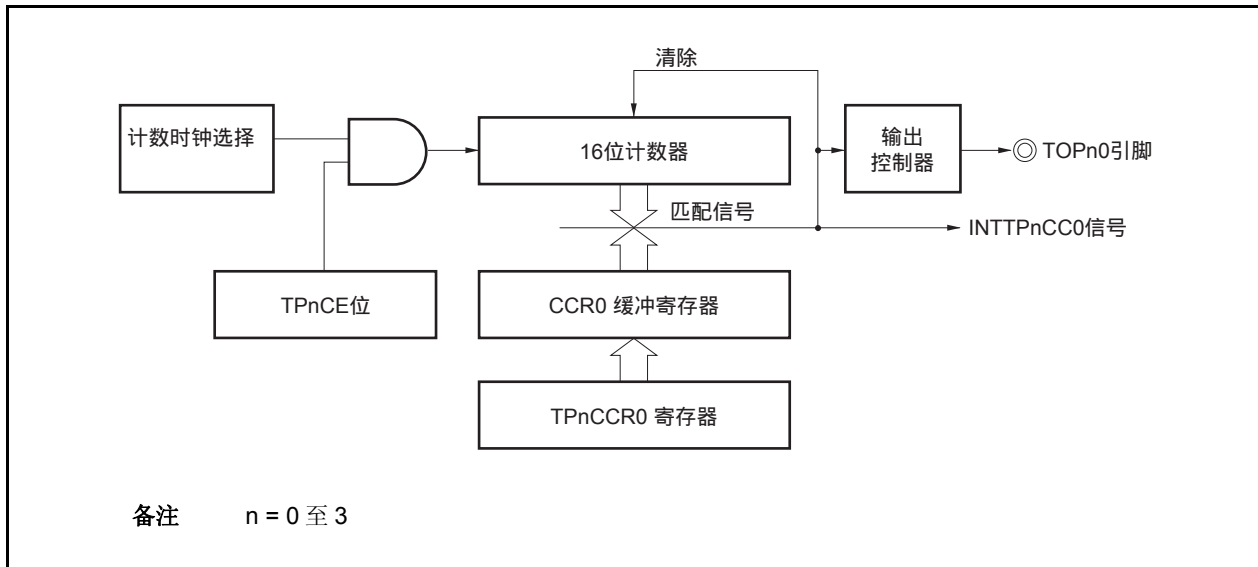
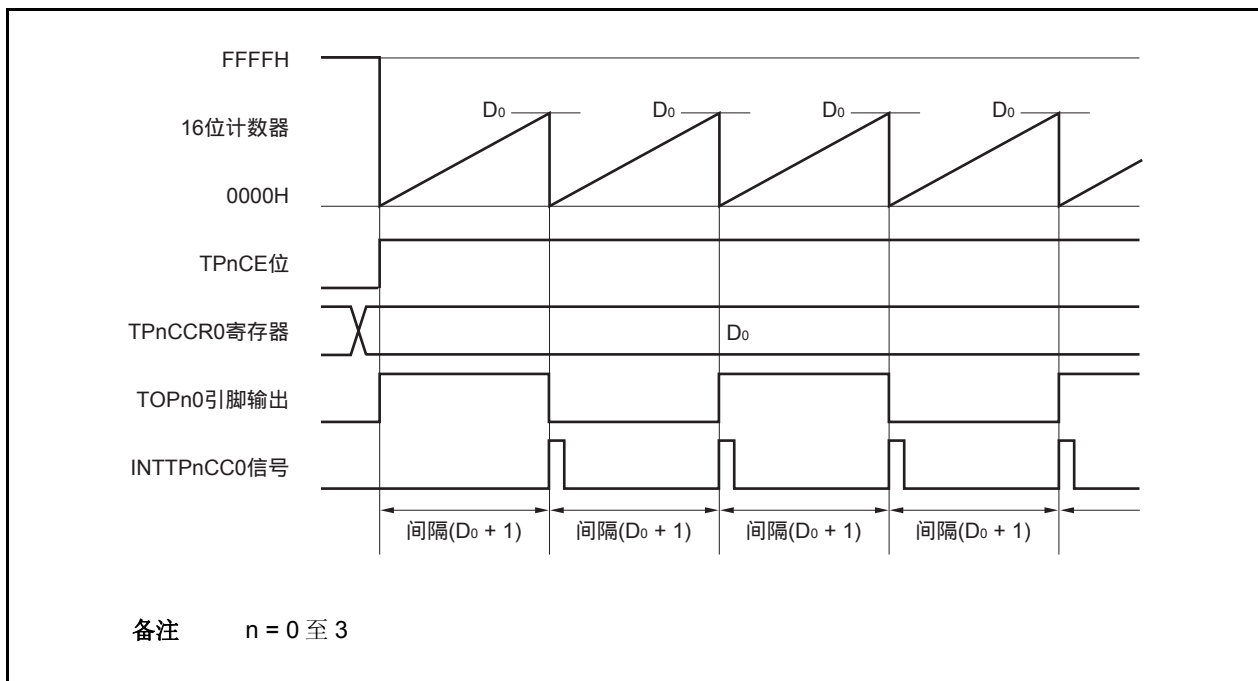


图 6-3. 间隔定时器模式操作的基本时序



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

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

间隔时间可由以下算式求得。

间隔时间=（TPnCCR0 寄存器的设定值+ 1）× 计数时钟周期

备注 n = 0 至 3

图 6-4. 间隔定时器模式操作的寄存器设置（1/2）

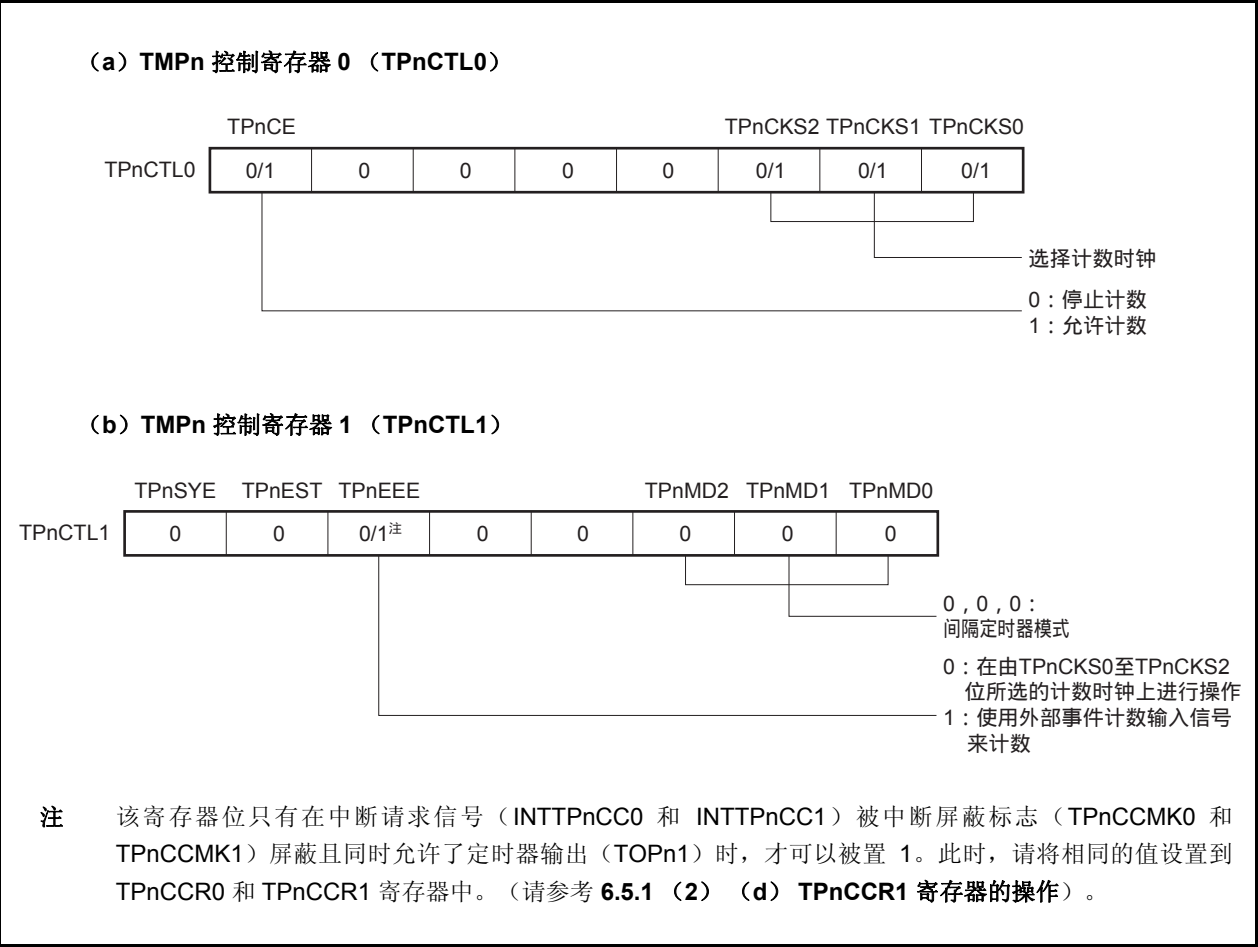
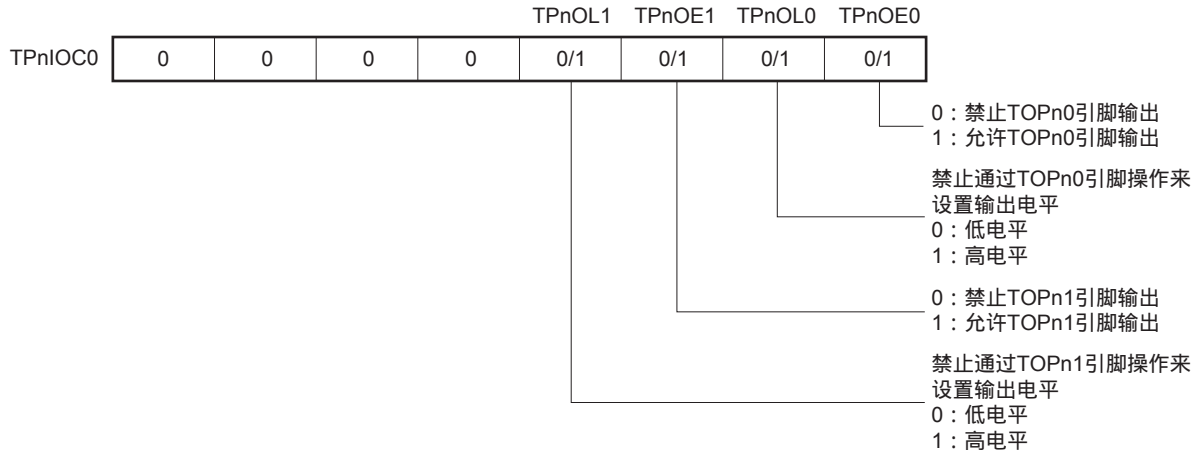


图 6-4. 间隔定时器模式操作的寄存器设置 (2/2)

(c) TMPn I/O 控制寄存器 0 (TPnIOC0)



(d) TMPn 计数器读取缓冲寄存器 (TPnCNT)

可以通过读取 TPnCNT 寄存器来读取 16 位计数器的计数值。

(e) TMPn 捕捉/比较寄存器 0 (TPnCCR0)

如果将 TPnCCR0 寄存器设置为 D_0 ，那么间隔时间如下。

$$\text{间隔时间} = (D_0 + 1) \times \text{计数时钟周期}$$

(f) TMPn 捕捉/比较寄存器 1 (TPnCCR1)

通常情况下，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，也会产生一个比较匹配中断请求信号 (INTTPnCC1)。

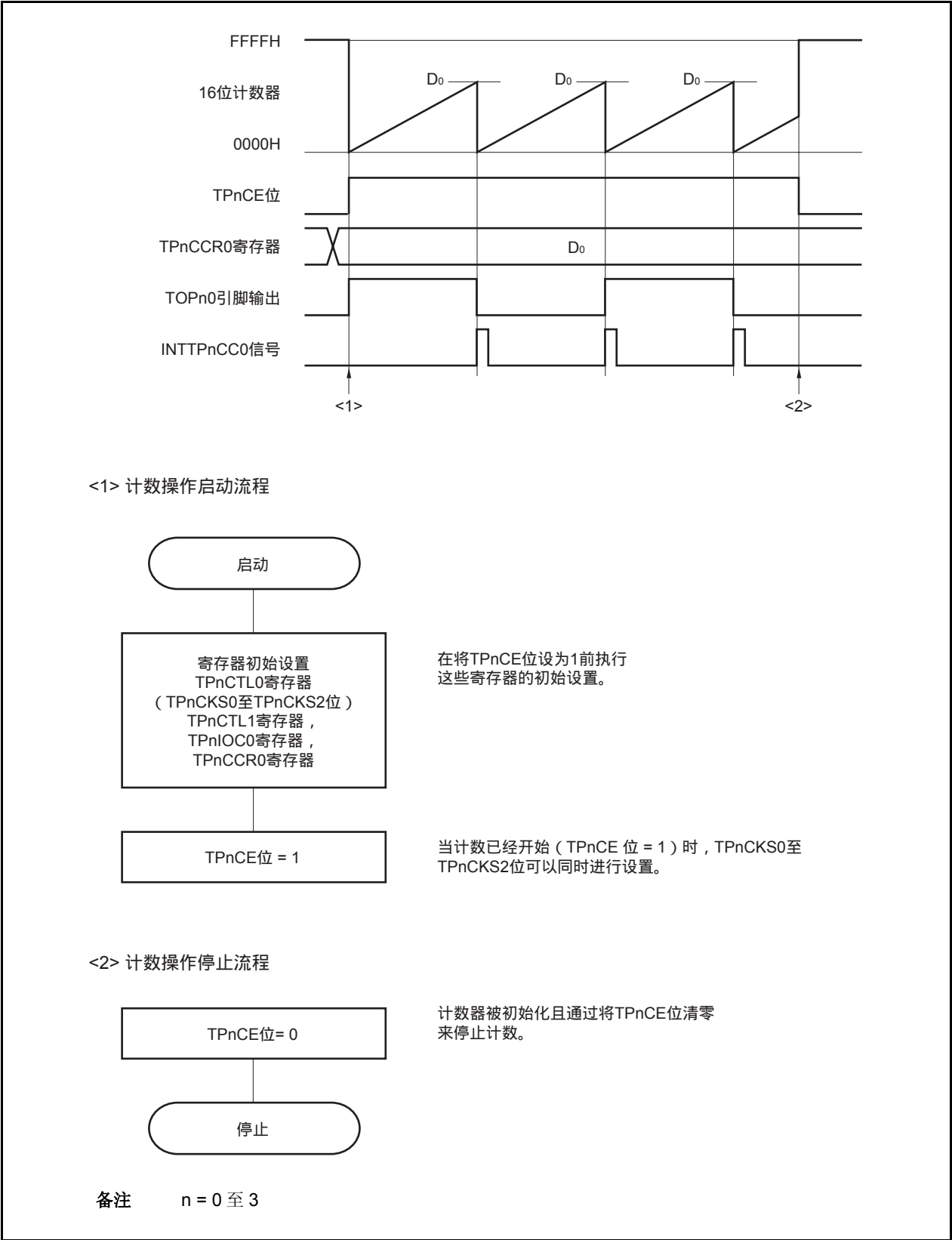
因此，请使用相应的中断屏蔽标志 (TPnCCMK1) 将中断请求屏蔽。

备注 1. 间隔定时器模式下，并未使用 TMPn I/O 控制寄存器 1 (TPnIOC1)，TMPn I/O 控制寄存器 2 (TPnIOC2) 以及 TMPn 选项寄存器 0 (TPnOPT0)。

2. $n = 0$ 至 3

(1) 间隔定时器模式操作流程

图 6-5. 间隔定时器模式下的软件处理流程

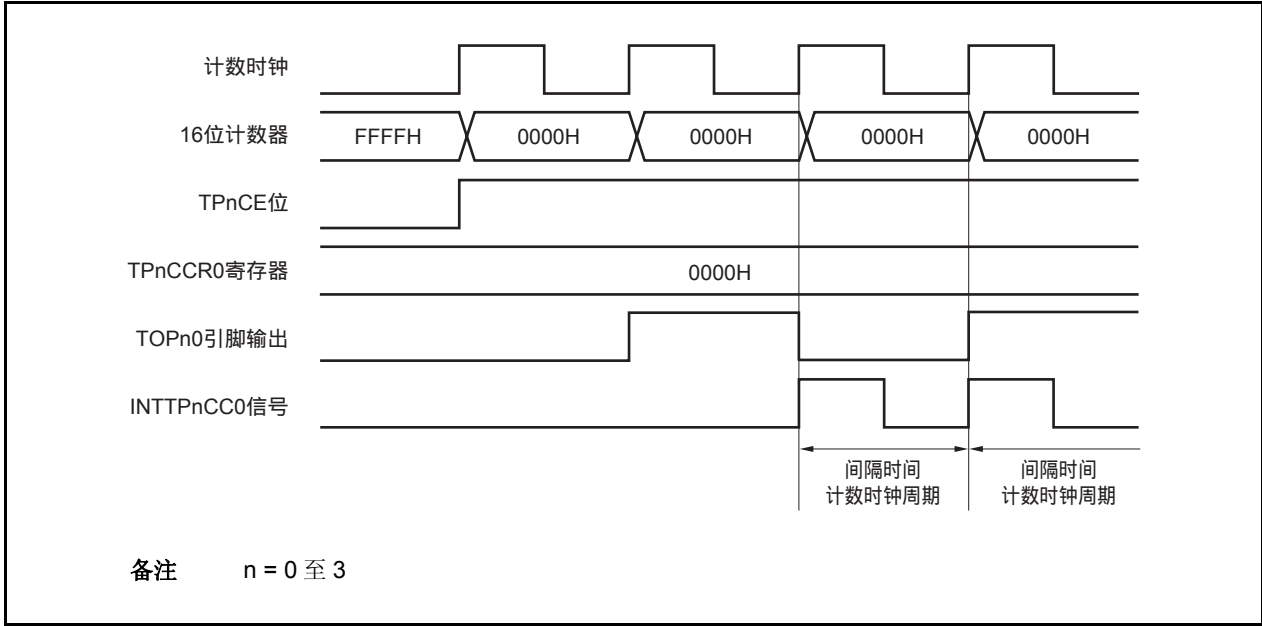


(2) 间隔定时器模式操作时序

(a) TPnCCR0 寄存器设置为 0000H 时的操作

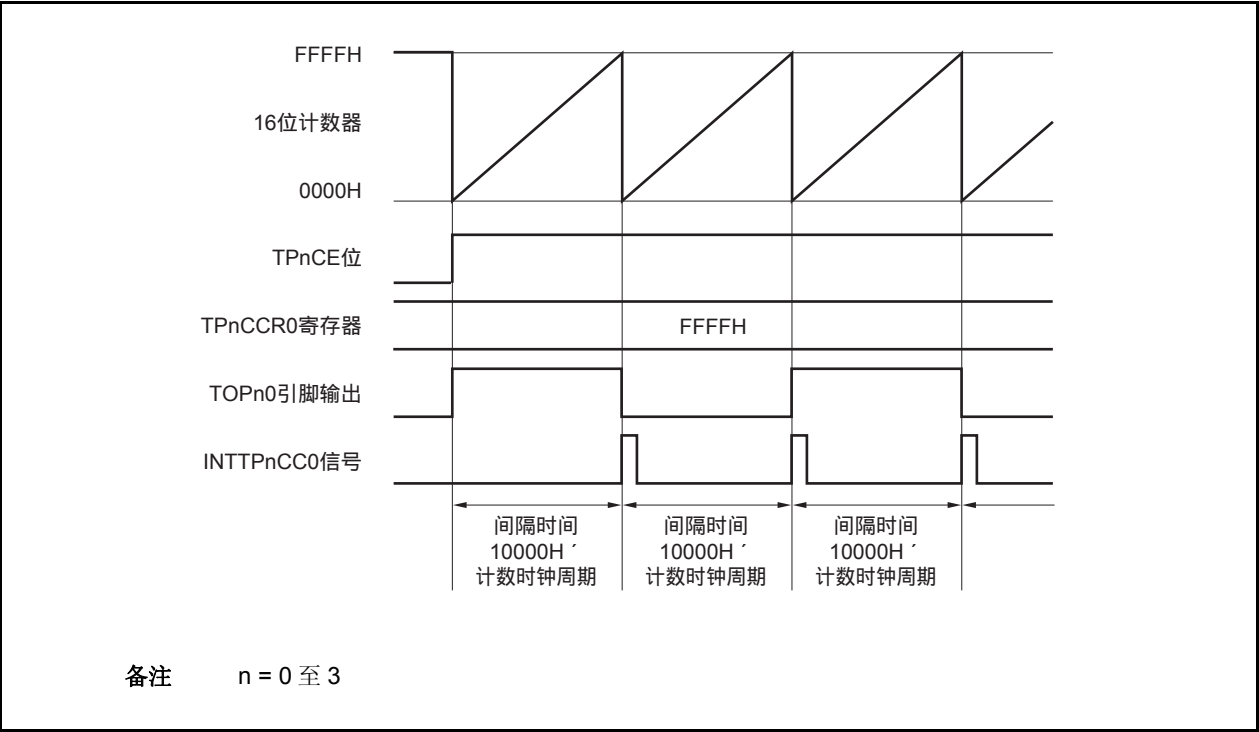
如果 TPnCCR0 寄存器被设成 0000H，在第一个计数时钟之后的每个计数时钟上产生 INTTPnCC0 信号，并且 TOPn0 管脚的输出被取反。

16 位计数器的计数值始终保持为 0000H。



(b) TPnCCR0 寄存器被设置为 FFFFH 时的操作

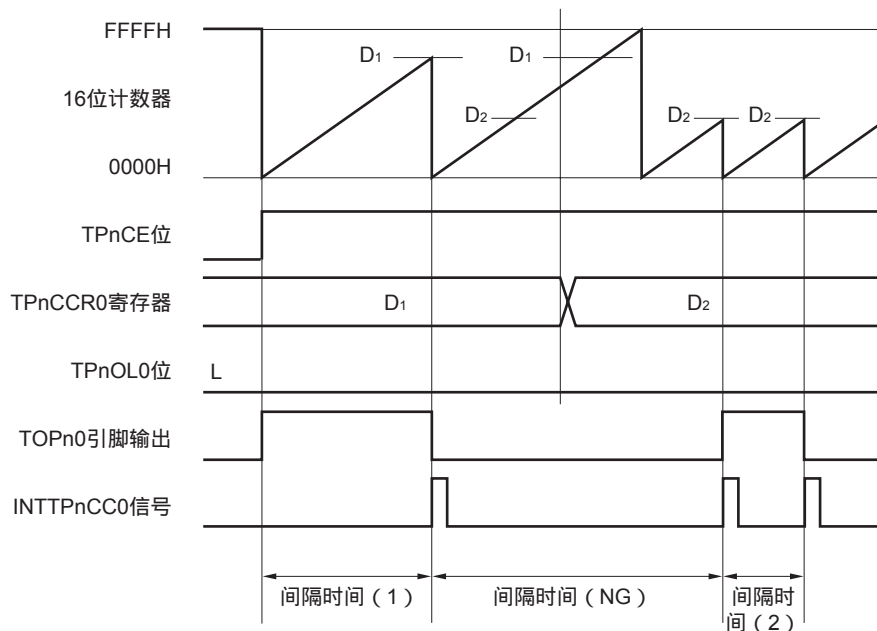
如果 TPnCCR0 寄存器被设置为 FFFFH，那么 16 位计数器将计数到 FFFFH。并在下一个计数时钟时刻同步被清零。同时产生 INTTPnCC0 信号，TOPn0 引脚电平反转。此时，不会产生溢出中断请求信号（INTTPnOV），溢出标志位（TPnOPT0.TPnOVF 位）也不会被置 1。



(c) 改写 TPnCCR0 寄存器时需要注意的事项

将 TPnCCR0 寄存器的值减小时，要先停止计数，再对设定值进行改变。

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



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

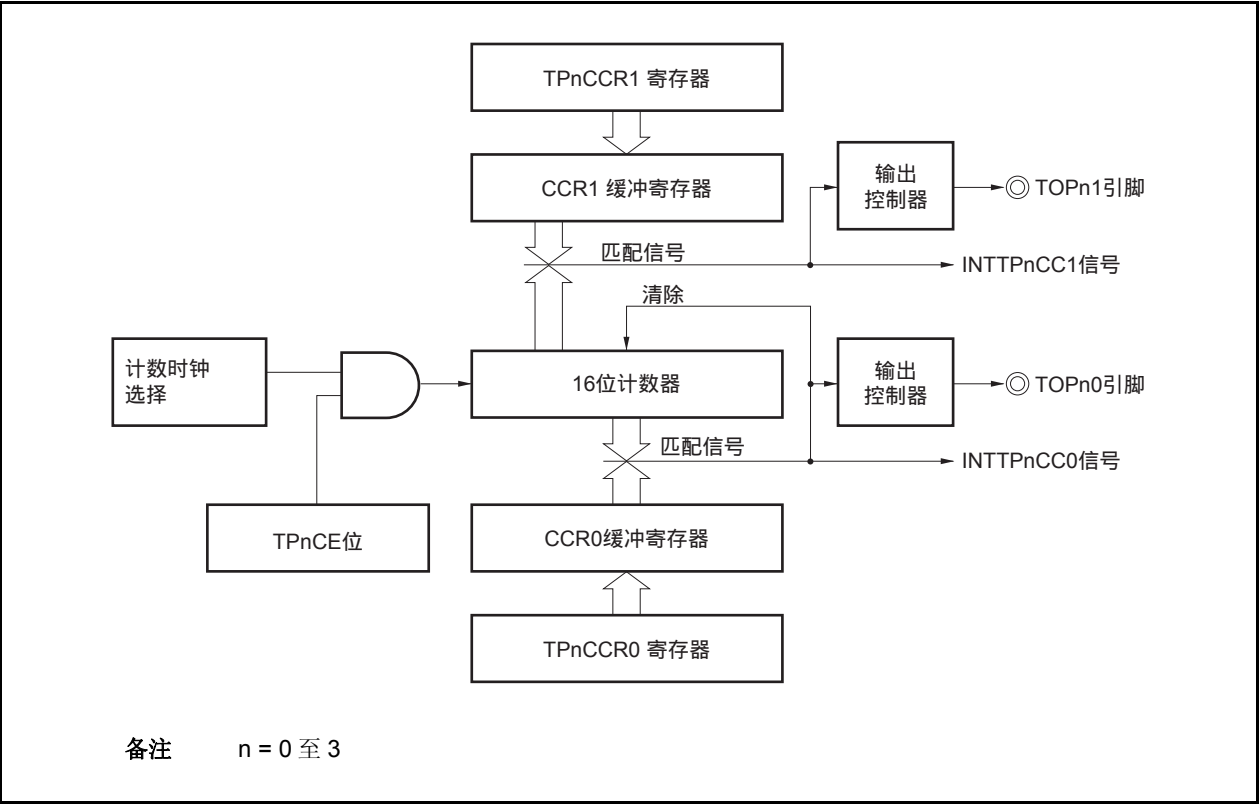
如果当前计数值大于 D_2 但小于 D_1 ，将 TPnCCR0 寄存器的值从 D_1 改变为 D_2 ，新的设定值将在改写后立即从 TPnCCR0 寄存器传送到 CCR0 缓冲寄存器中，那么，16 位计数器的计数值将与 D_2 进行比较。

由于计数值已经超过了 D_2 ，因此 16 位计数器会继续计数到 FFFFH 并产生溢出，之后重新从 0000H 开始计数。在重新计数过程中，计数值与 D_2 匹配时，才会产生 INTTPnCC0 信号，同时 TOPn0 引脚输出电平反转。

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

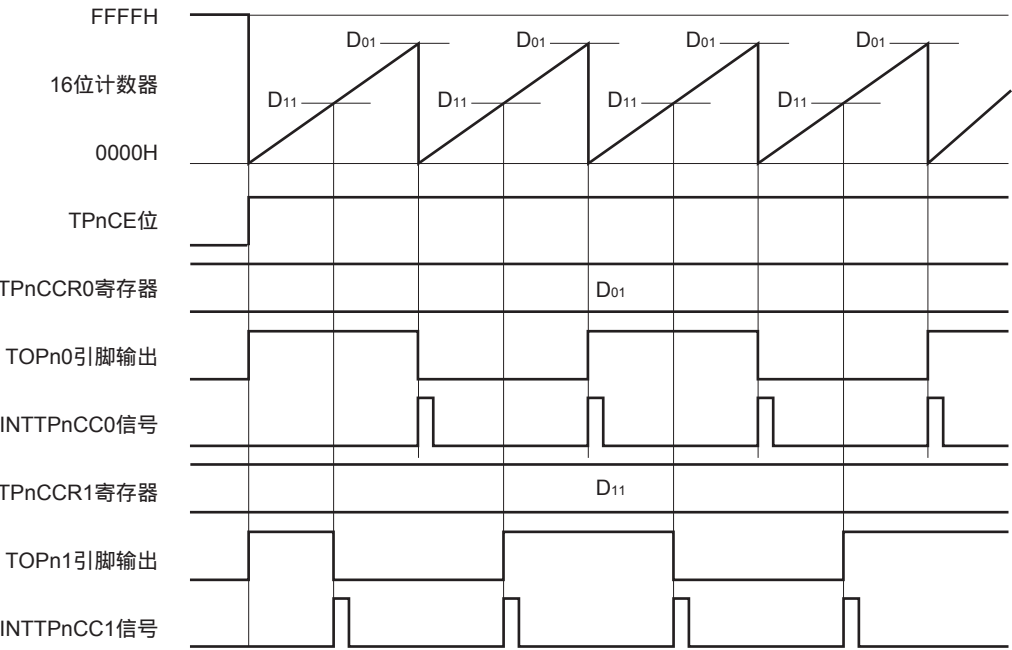
(d) TPnCCR1 寄存器的操作

图 6-6. TPnCCR1 寄存器的配置图



如果 TPnCCR1 寄存器的值小于 TPnCCR0 寄存器的值，那么 INTTPnCC1 信号将每周期产生一次。与此同时，TOPn1 引脚的电平也会反转。
TOPn1 引脚输出的方波周期与 TOPn0 引脚的方波周期相同。

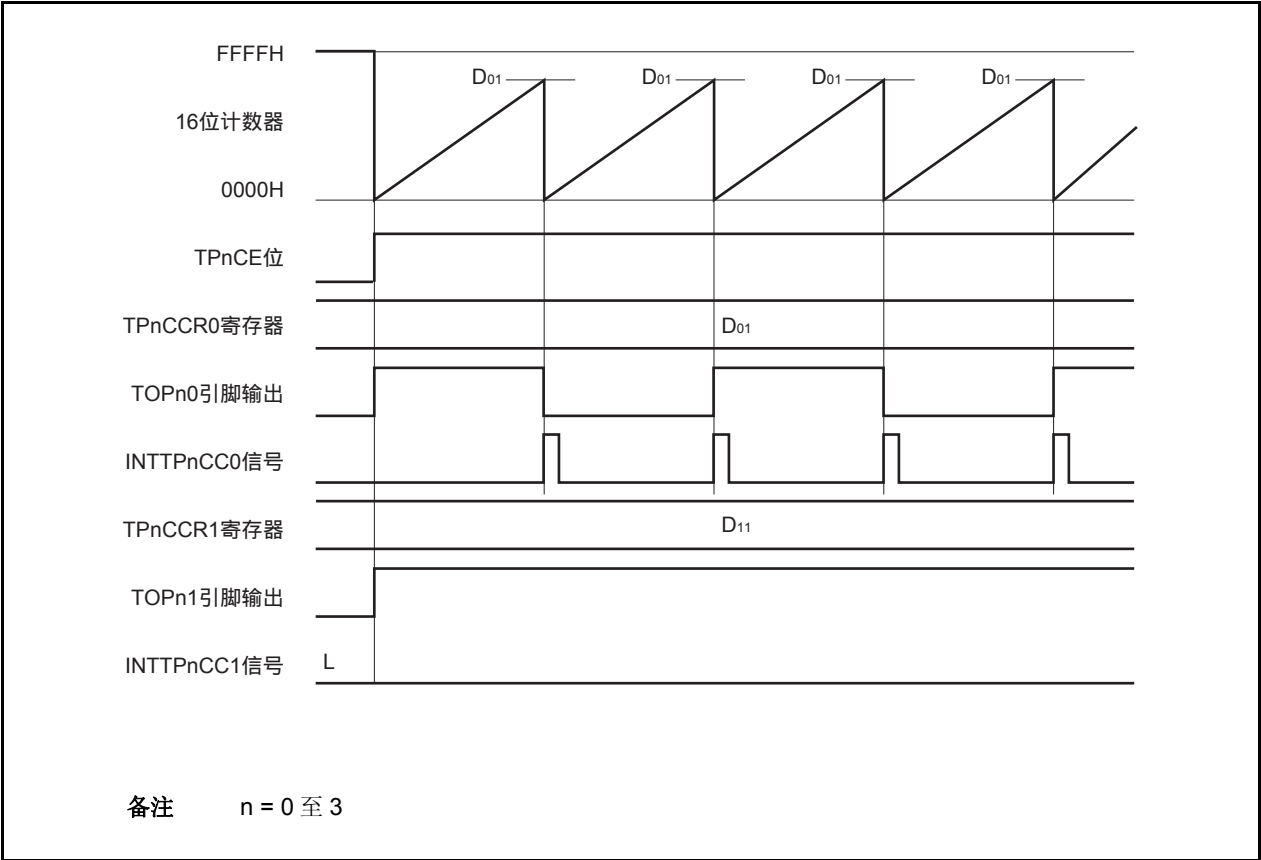
图 6-7. $D_{01} \geq D_{11}$ 时的时序图



备注 n = 0 至 3

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

图 6-8. D01 < D11 时的时序图

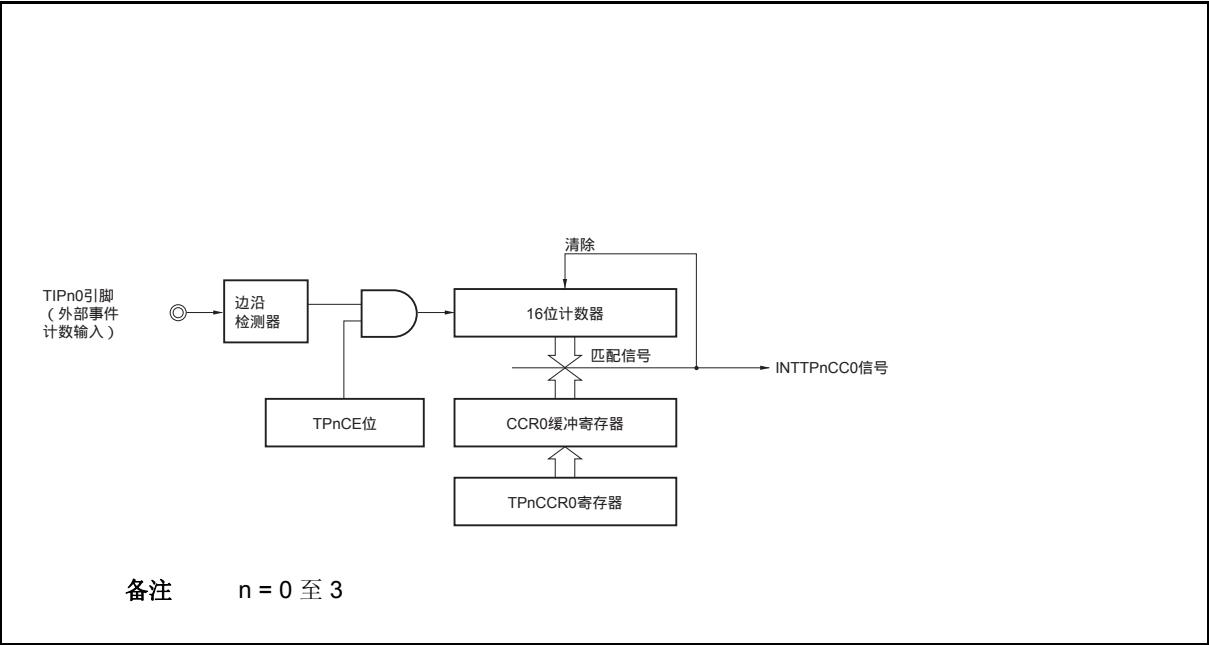


6.5.2 外部事件计数模式 (TPnMD2 至 TPnMD0 位 = 001)

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

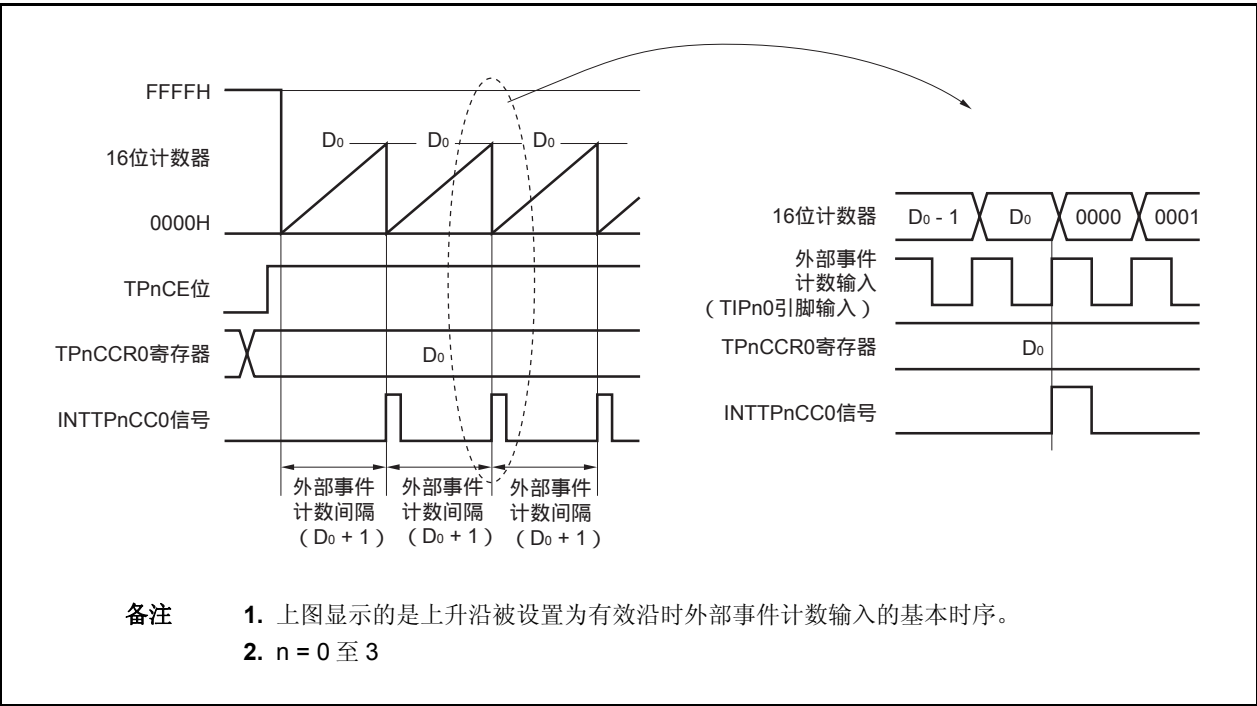
通常情况下，外部事件计数模式下不使用 TPnCCR1 寄存器。

图 6-9. 外部事件计数模式的配置图



备注 n = 0 至 3

图 6-10. 外部事件计数模式的基本时序



备注 1. 上图显示的是上升沿被设置为有效沿时外部事件计数输入的基本时序。
2. n = 0 至 3

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

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

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

图 6-11. 外部事件计数模式操作的寄存器设置 (1/2)

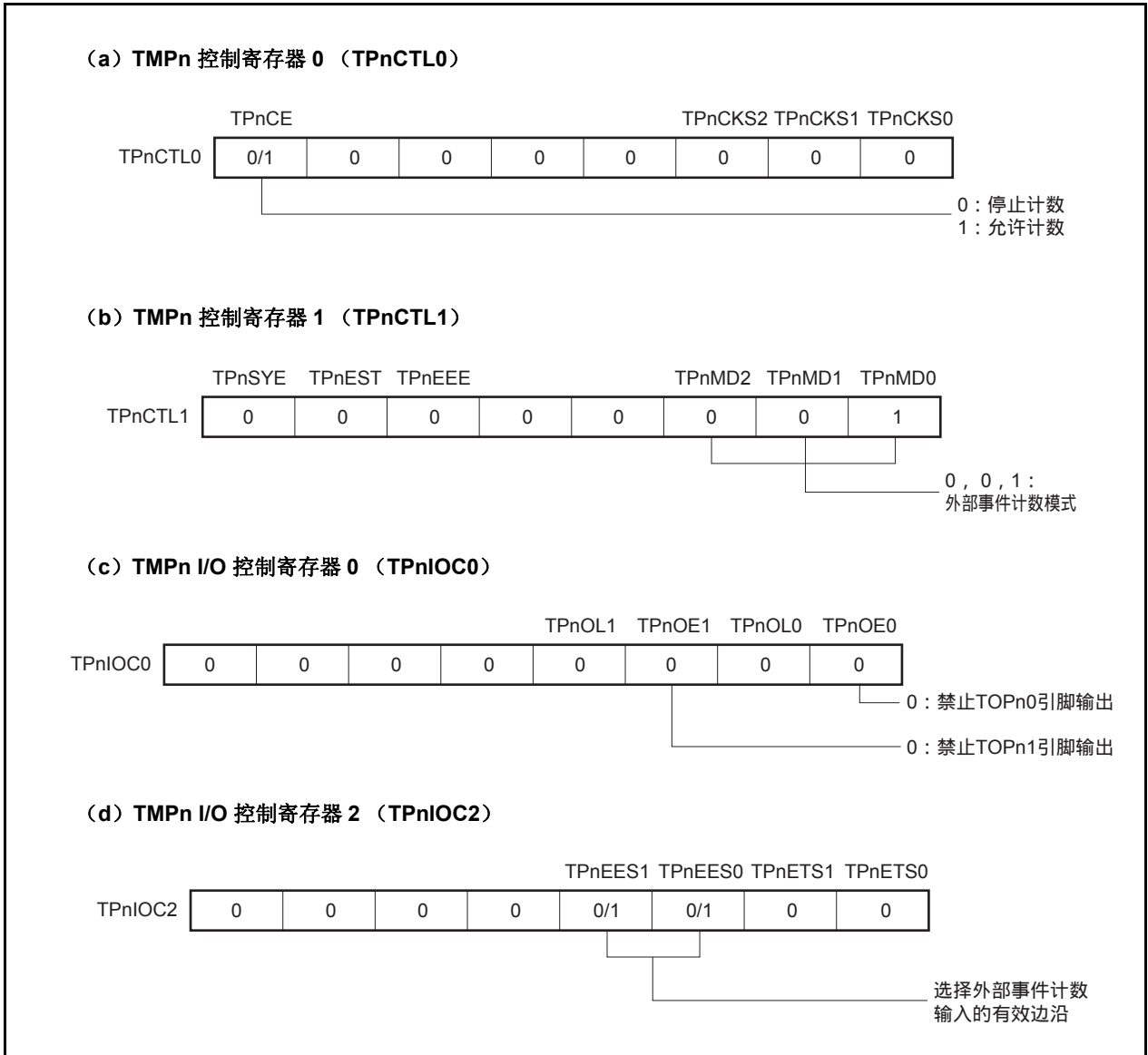


图 6-11. 外部事件计数模式操作的寄存器设置 (2/2)

(e) TMPn 计数器读取缓冲寄存器 (TPnCNT)

可以通过读取 TPnCNT 寄存器来读取 16 位计数器的计数值。

(f) TMPn 捕捉/比较寄存器 0 (TPnCCR0)

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

(g) TMPn 捕捉/比较寄存器 1 (TPnCCR1)

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

因此，请使用相应得中断屏蔽标志 (TPnCCMK1) 将中断请求屏蔽。

注意事项 当使用一个外部时钟作为计数时钟时，外部时钟只能从 TIPn0 管脚输入。这是，须将 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位清零（捕捉触发器输入 (TIPn0 管脚)：未测试到边沿）。

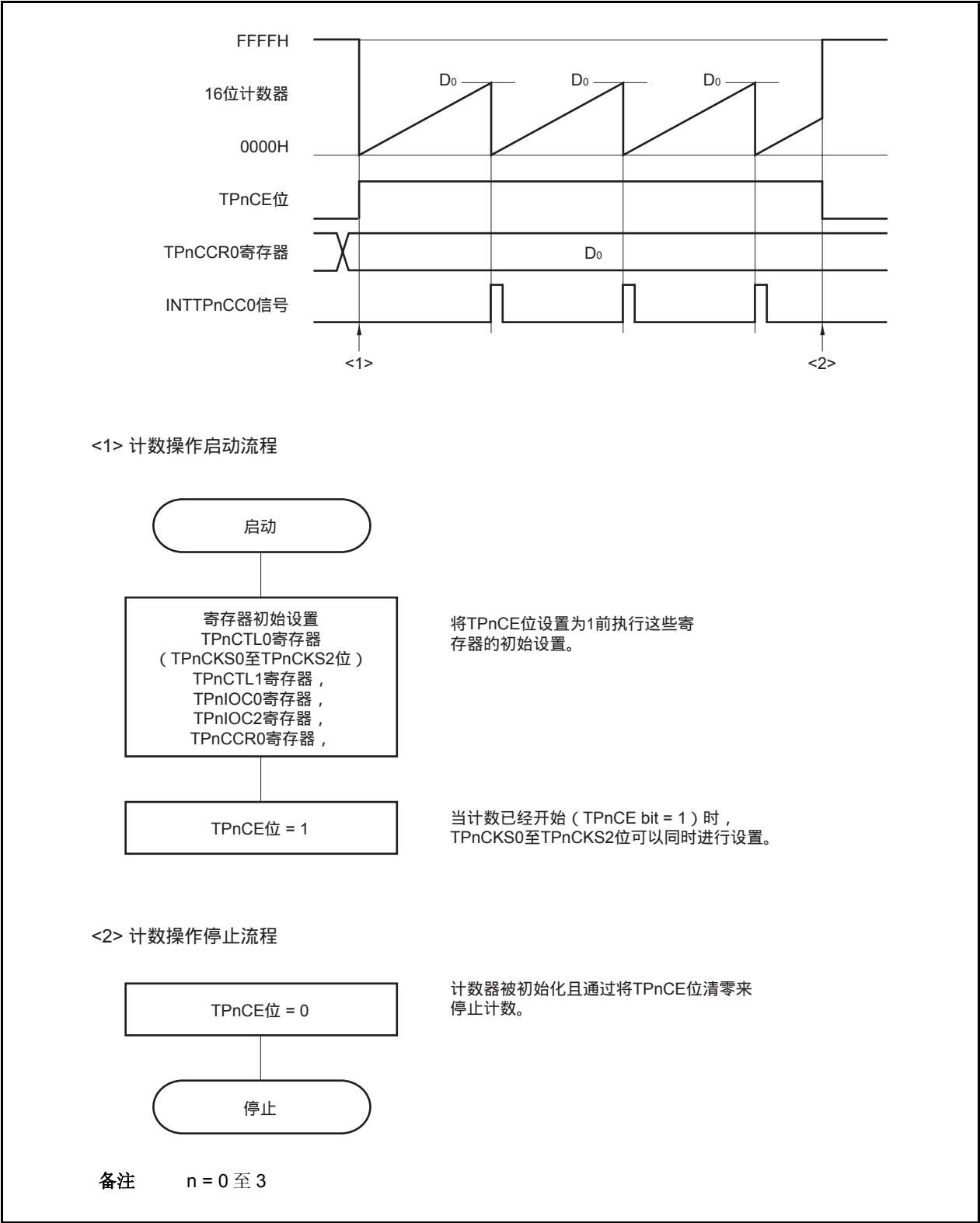
备注

1. 外部事件计数模式下，并未使用 TMPn I/O 控制寄存器 1 (TPnIOC1) 和 TMPn 选项寄存器 0 (TPnOPT0)。
2. $n = 0$ 至 3

<R>

(1) 外部事件计数模式操作流程

图 6-12. 外部事件计数模式的软件处理流程



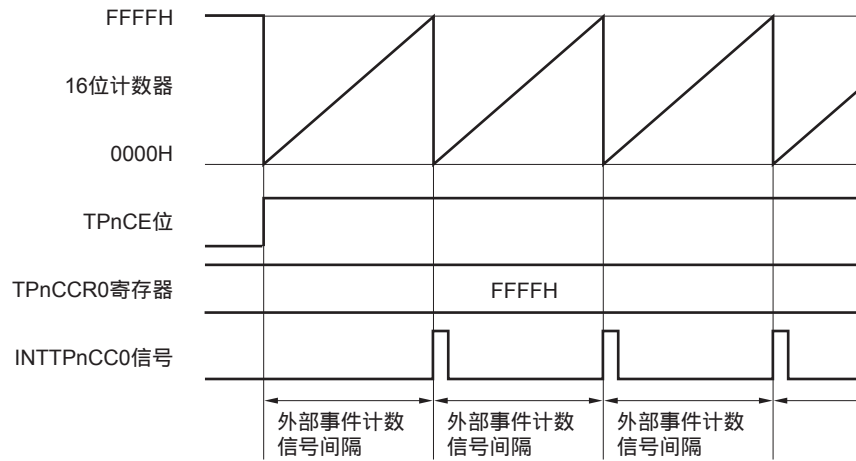
(2) 外部事件计数模式的操作时序

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

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

(a) TPnCCR0 寄存器被设置为 FFFFH 时的操作

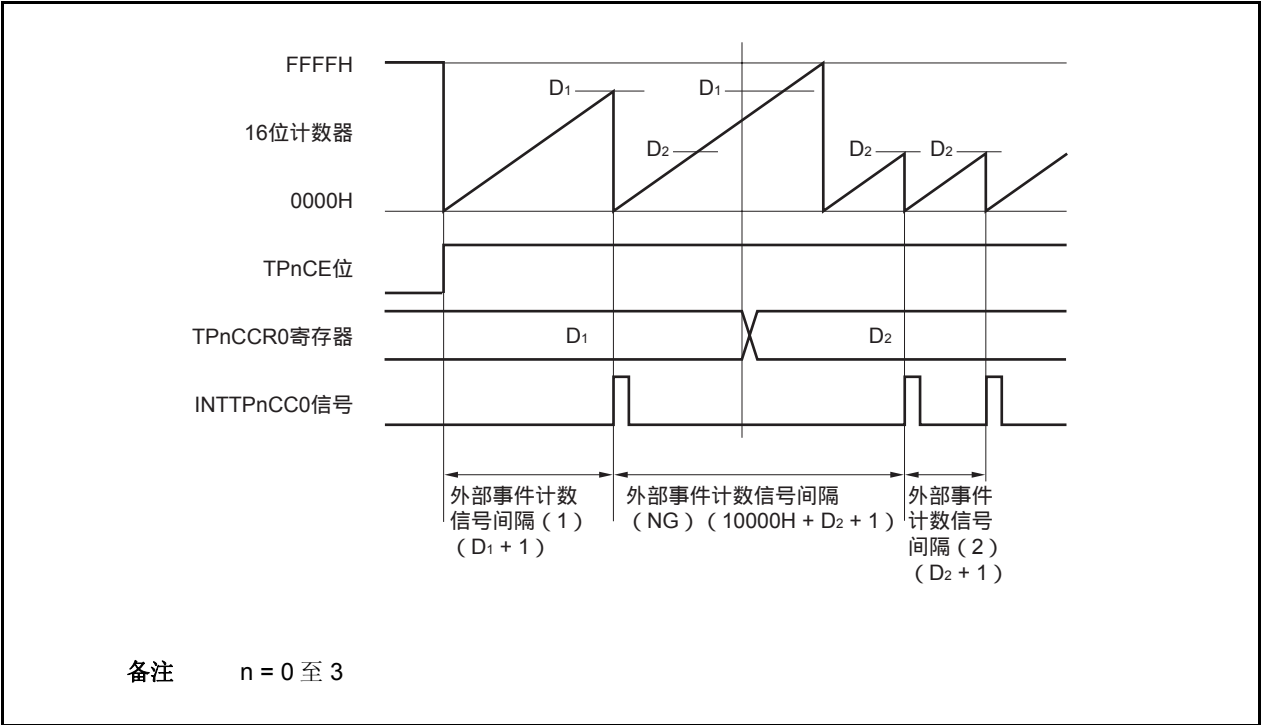
如果 TPnCCR0 寄存器被设置为 FFFFH，那么 16 位计数器将在每次检测到外部事件计数信号的有效沿处计数，直到 FFFFH。并在下一个计数事件时刻被同步清零，同时产生 INTTPnCC0 信号。此时，溢出标志位 (TPnOPT0.TPnOVF 位) 不会被置 1。



备注 n = 0 至 3

(b) 改写 TPnCCR0 寄存器时需要注意的事项

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

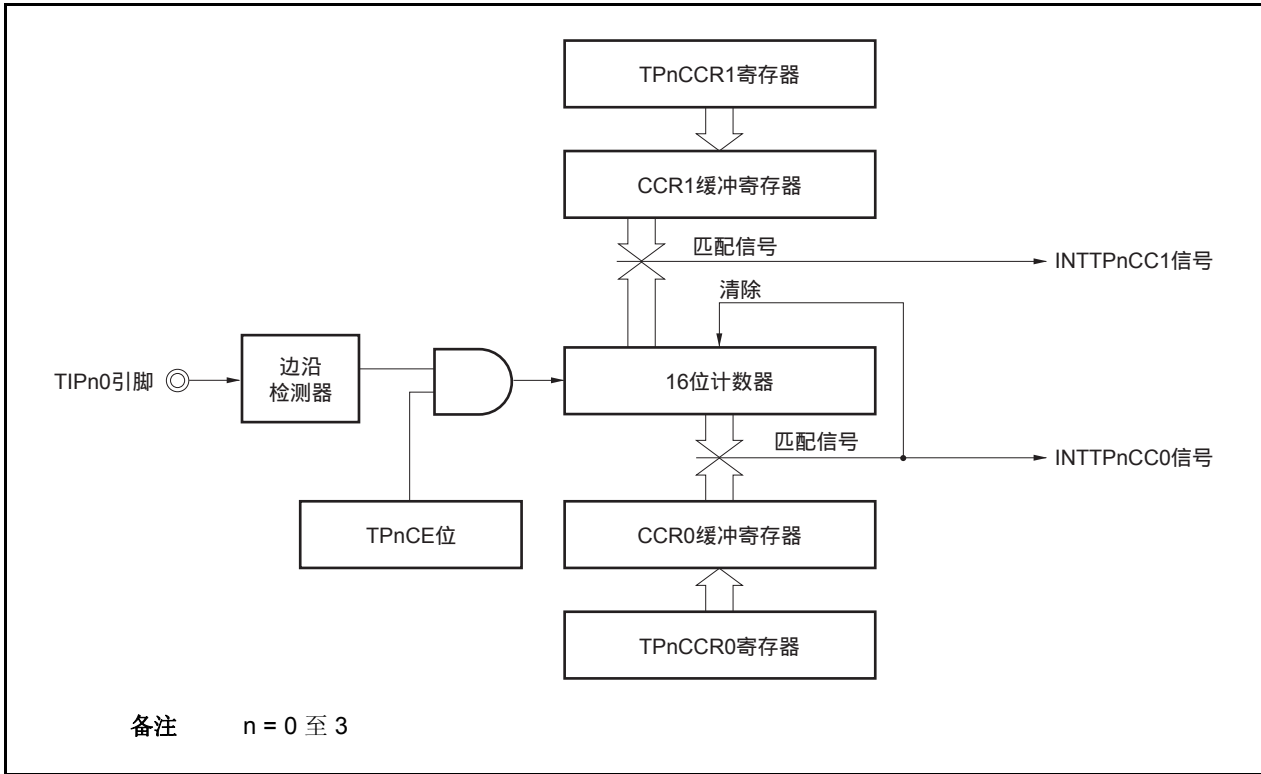


如果当前计数值大于 D_2 但小于 D_1 ，将 TPnCCR0 寄存器的值从 D_1 改变为 D_2 ，新的设定值将在改写后立即从 TPnCCR0 寄存器传送到 CCR0 缓冲寄存器中，那么，16 位计数器的计数值将与 D_2 进行比较。由于计数值已经超过了 D_2 ，因此 16 位计数器会继续计数到 FFFFH 并产生溢出，之后重新从 0000H 开始计数。在重新计数过程中，计数值与 D_2 匹配时，才会产生 INTTPnCC0 信号。

因此可见，INTTPnCC0 信号不会在预期的事件计数次数 “ $(D_1 + 1)$ 次” 或 “ $(D_2 + 1)$ 次” 时产生，而是在一个 “ $(10000H + D_2 + 1)$ 次” 的计数次数时产生。

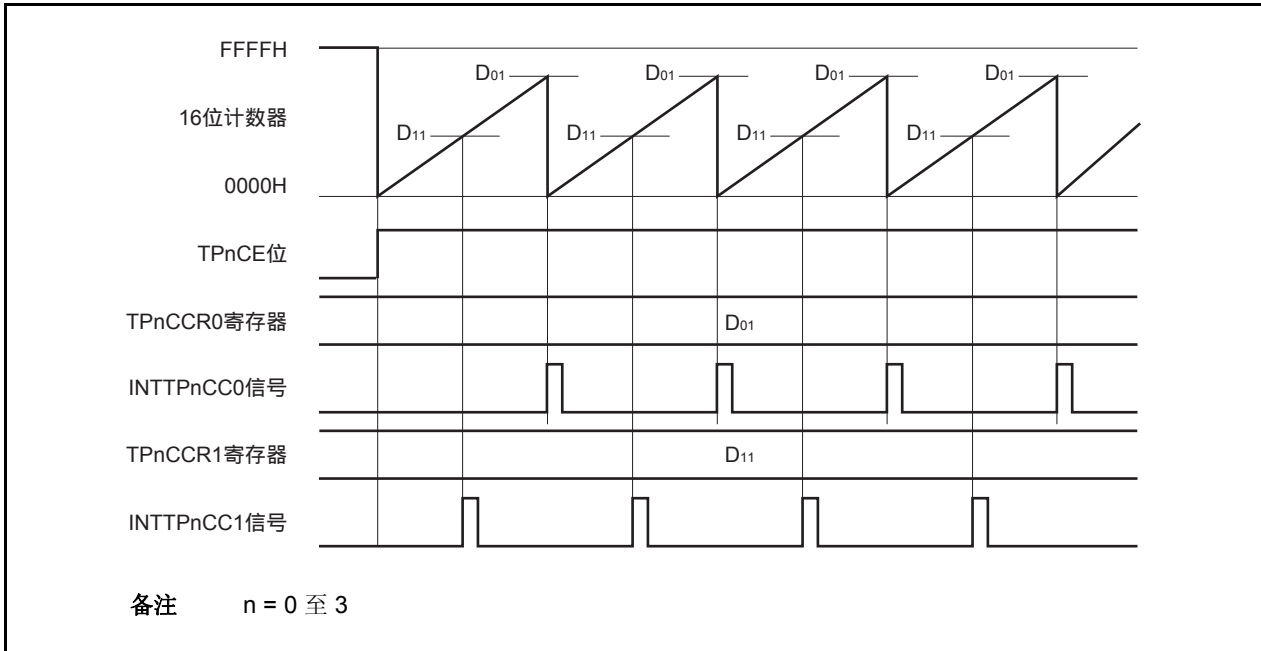
(c) TPnCCR1 寄存器的操作

图 6-13. TPnCCR1 寄存器的配置



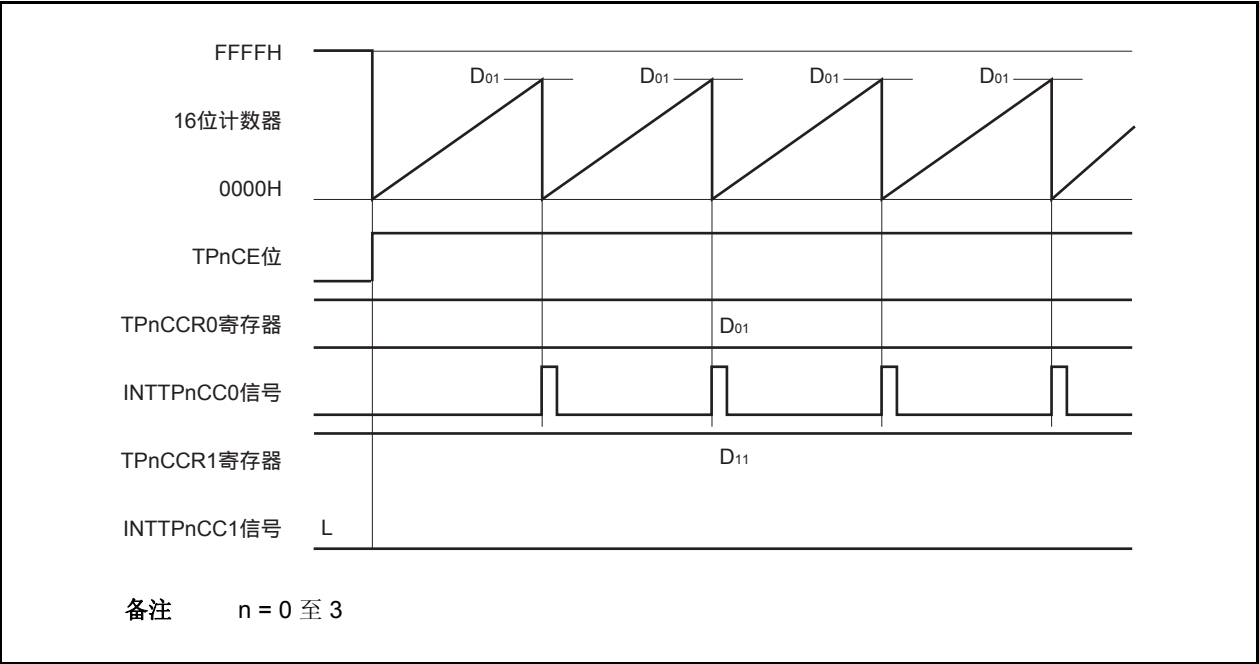
如果 TPnCCR1 寄存器的值小于 TPnCCR0 寄存器的值，那么 INTTPnCC1 信号将每周期产生一次。

图 6-14. $D_{01} \geq D_{11}$ 时的时序图



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

图 6-15. $D_{01} < D_{11}$ 时的时序图



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

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

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

图 6-16. 外部触发脉冲输出模式的配置

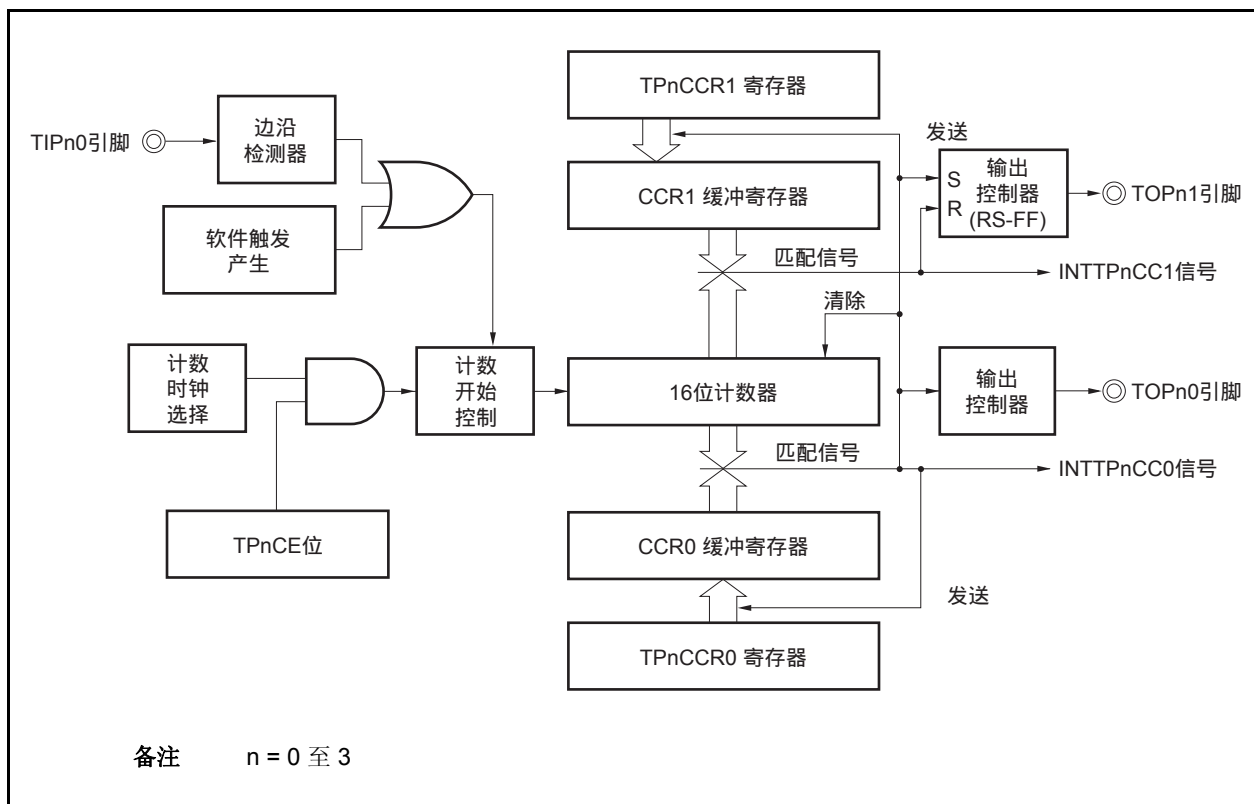
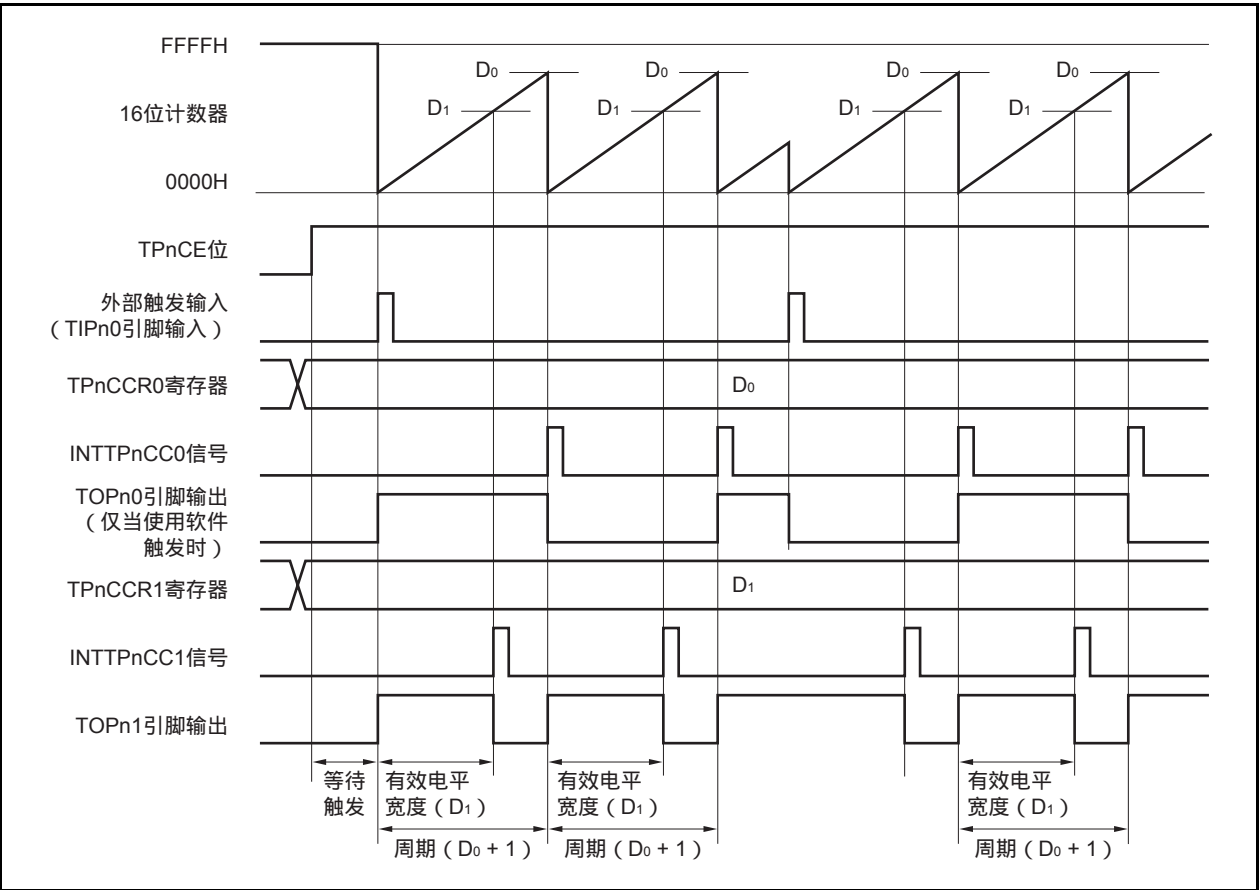


图 6-17. 外部触发脉冲输出模式的基本时序



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

PWM 波形的有效电平宽度，周期以及占空比系数可由以下的算式得出。

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

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

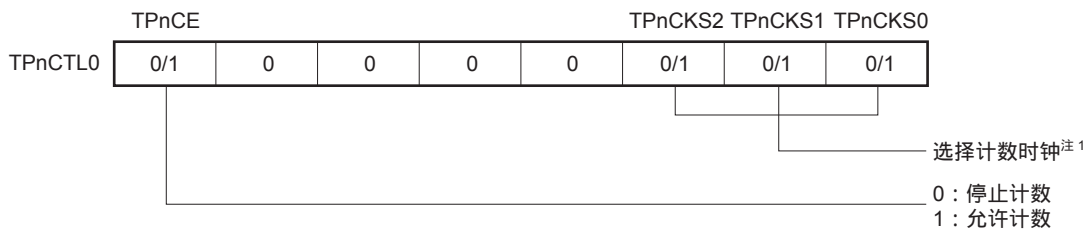
当 16 位计数器的计数值与 CCRm 缓冲寄存器中的值相匹配时，TPnCCRm 寄存器中设置的值会被传送至 CCRm 缓冲寄存器，同时 16 位计数器被清零。

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

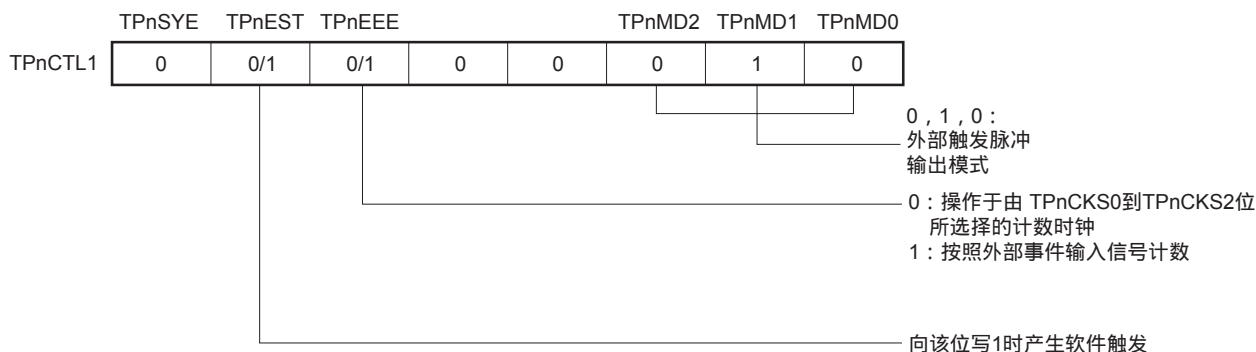
备注 n = 0 至 3, m = 0, 1

图 6-18. 外部触发脉冲输出模式的寄存器设置 (1/2)

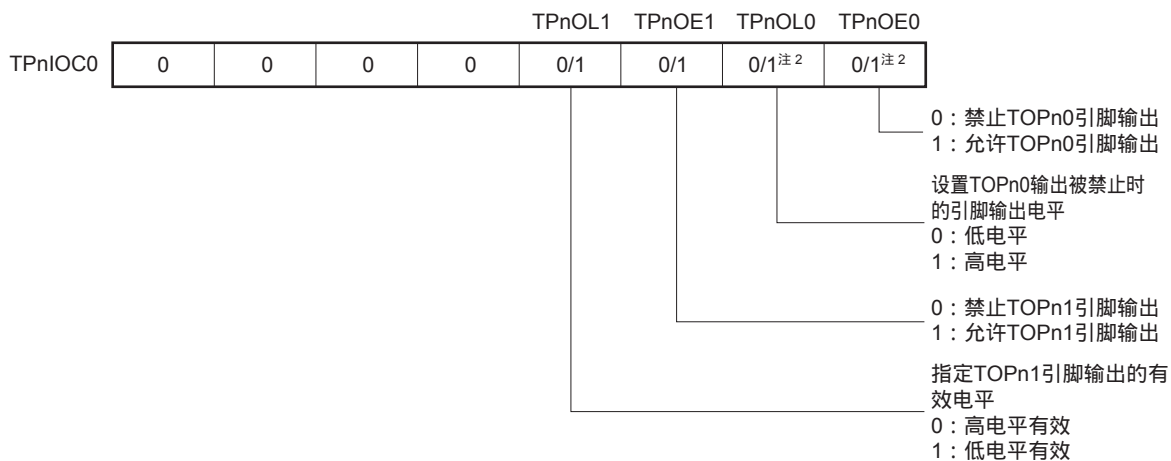
(a) TMPn 控制寄存器 0 (TPnCTL0)



(b) TMPn 控制寄存器 1 (TPnCTL1)



(c) TMPn I/O 控制寄存器 0 (TPnIOC0)



· TPnOL1 位 = 0



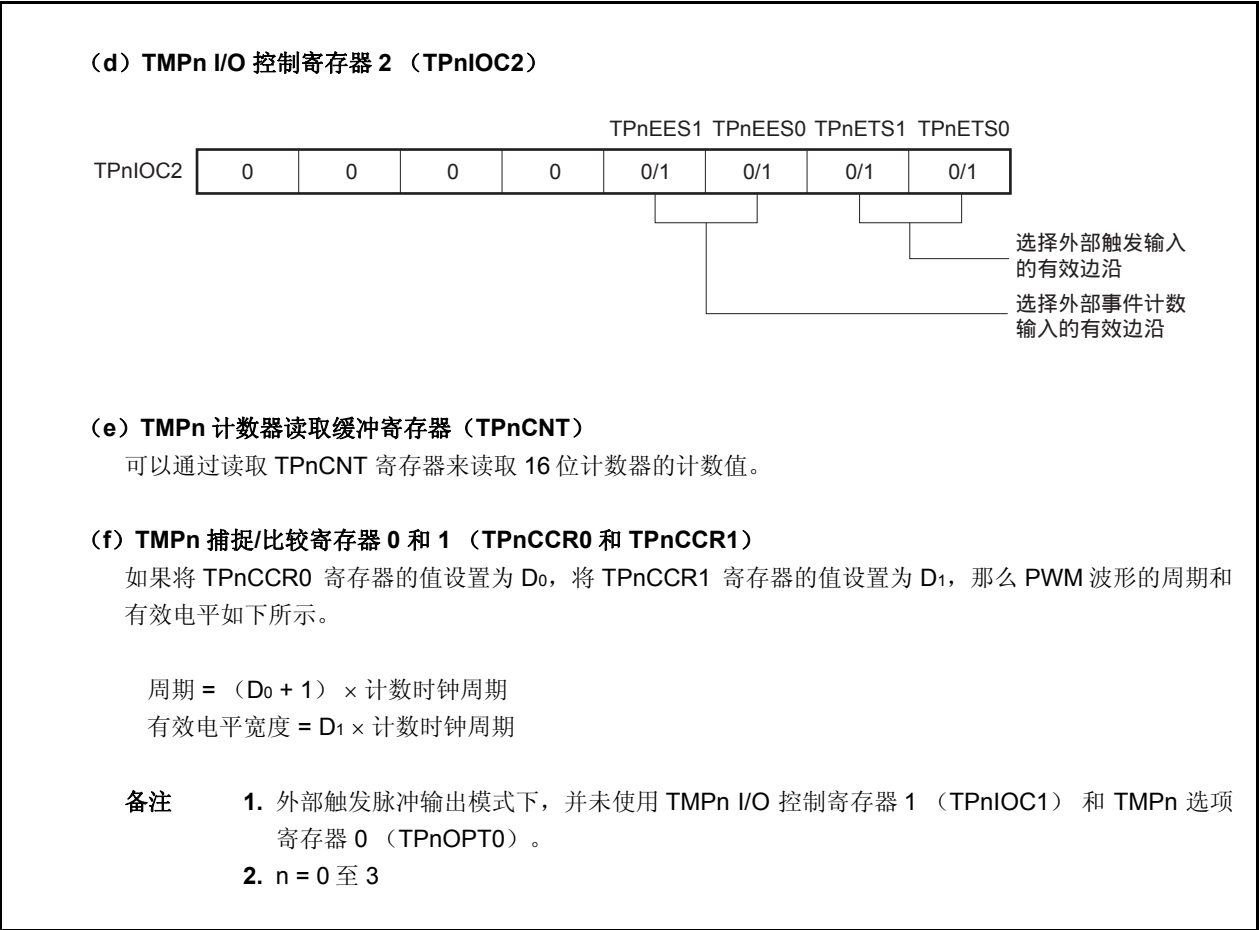
· TPnOL1 位 = 1时



注 1. TPnCTL1.TPnEEE 位 = 1 时该设置无效。

2. 外部触发脉冲输出模式下不使用 TOPn0 引脚时, 要将该位清零。

图 6-18. 外部触发脉冲输出模式的寄存器设置 (2/2)



(1) 外部触发脉冲输出模式的操作流程

图 6-19. 外部触发脉冲输出模式的软件处理流程 (1/2)

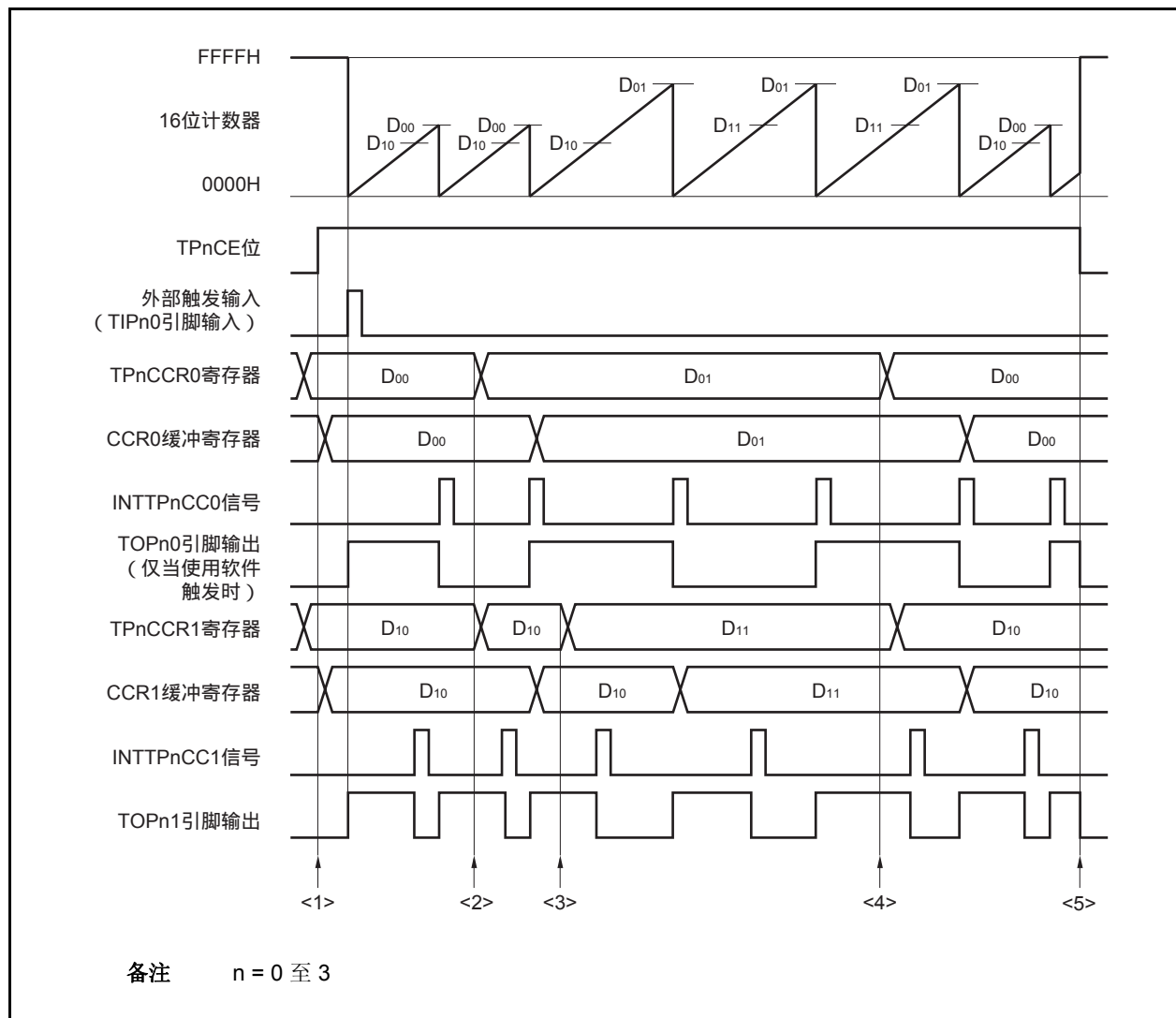
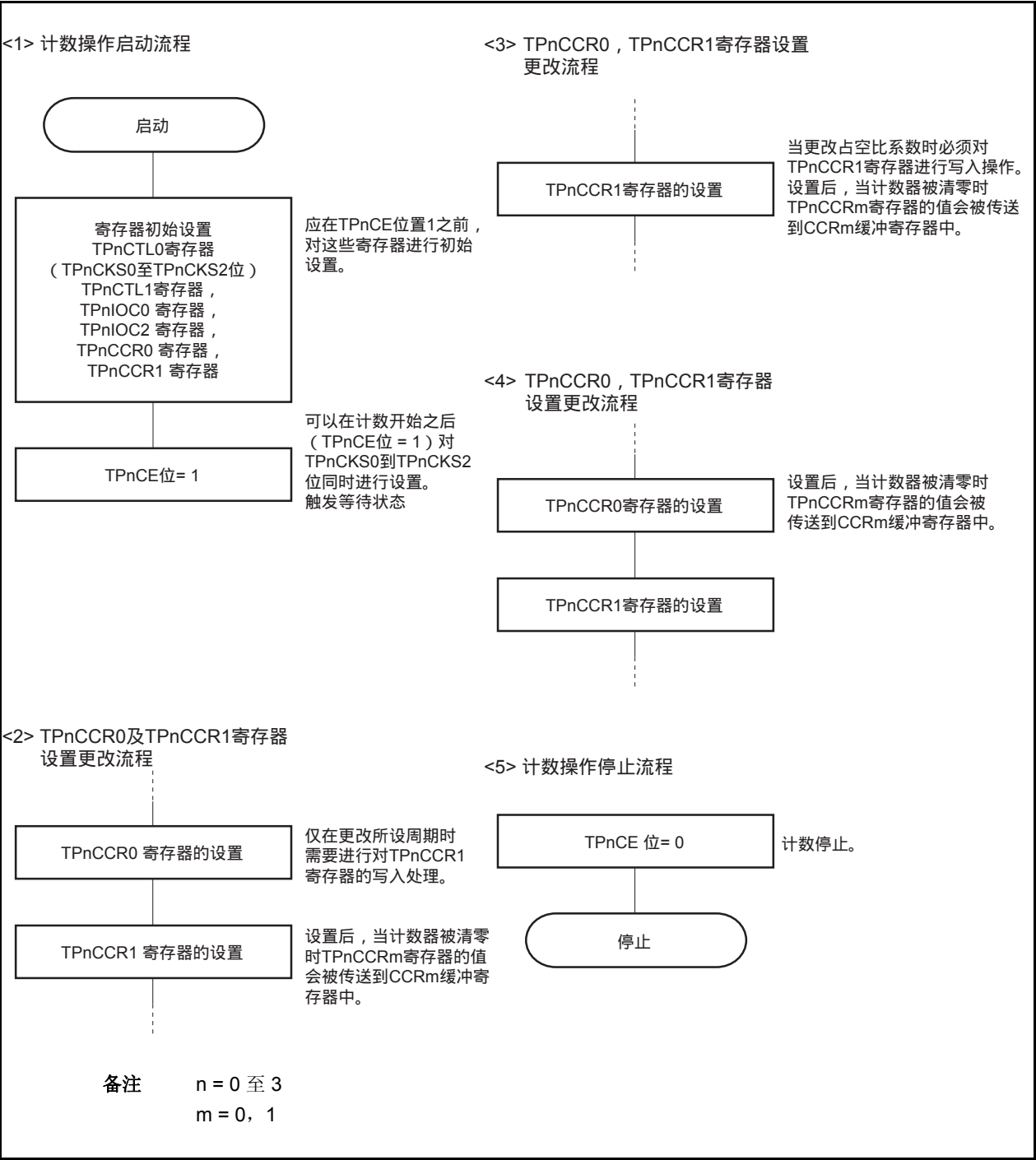


图 6-19. 外部触发脉冲输出模式的软件处理流程（2/2）

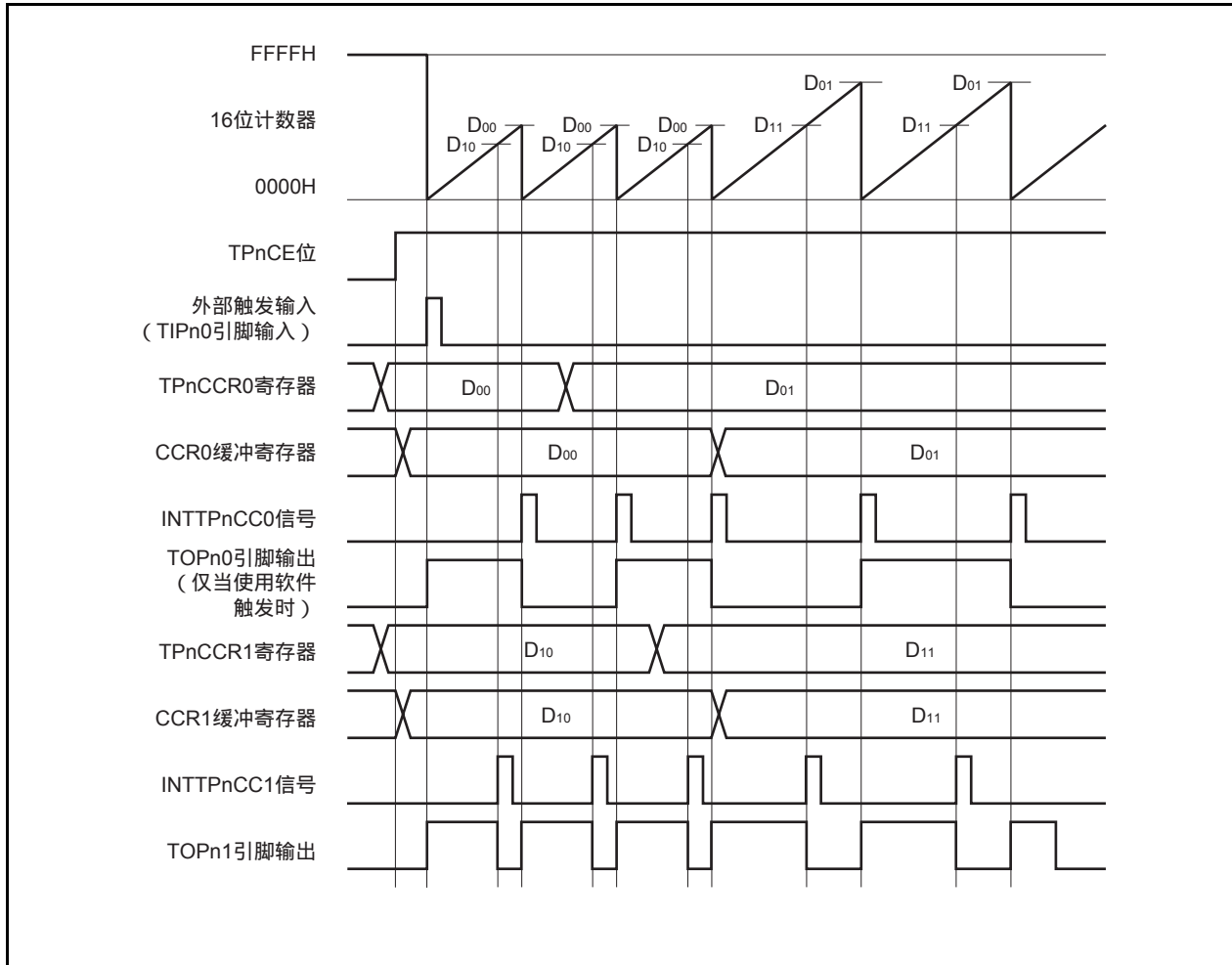


(2) 外部触发脉冲输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

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

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



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

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

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

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

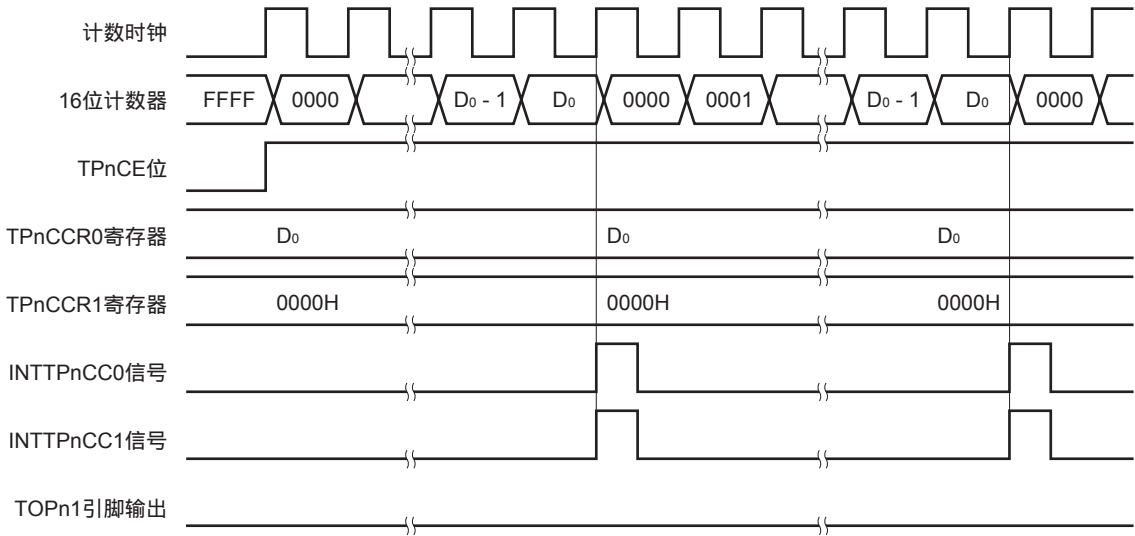
数据被写入 TPnCCR1 寄存器之后，当 16 位计数器被清零时，TPnCCRm 寄存器中的数值才会被传送到 CCRm 缓冲寄存器中。此时 CCRm 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

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

备注 n = 0 至 3
 m = 0, 1

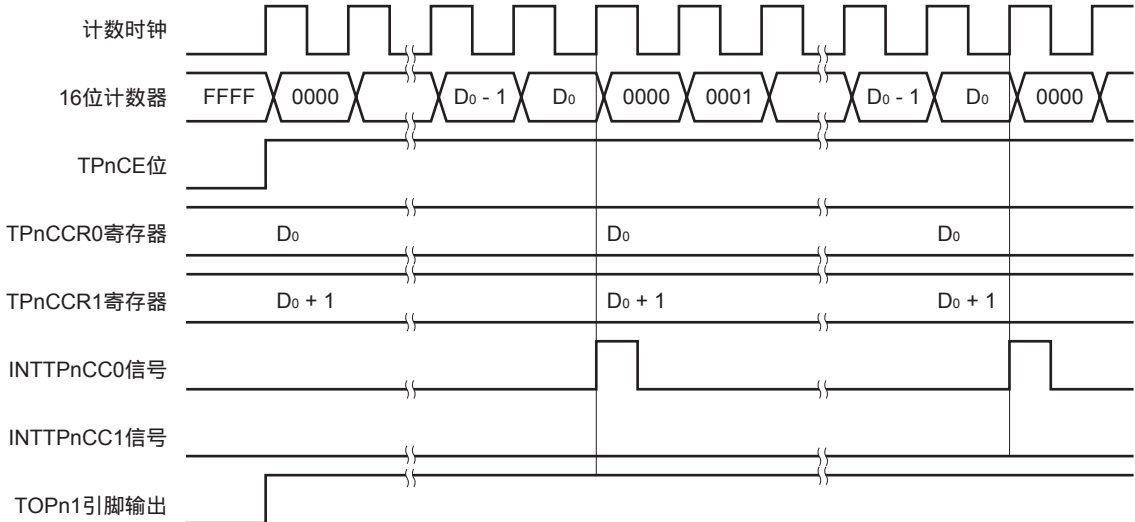
(b) PWM 波形的 0%/100% 输出

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



备注 n = 0 至 3

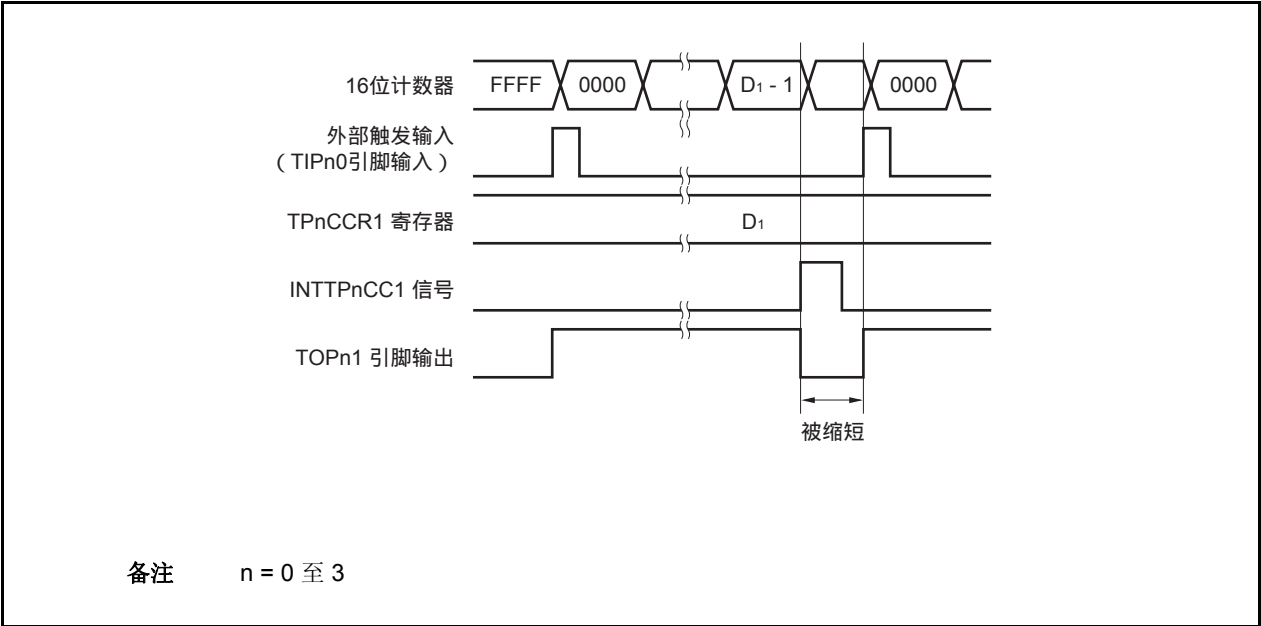
若要输出 100% 波形，则需将 TPnCCR1 寄存器设置为 (TPnCCR0 寄存器值 + 1)。TPnCCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。



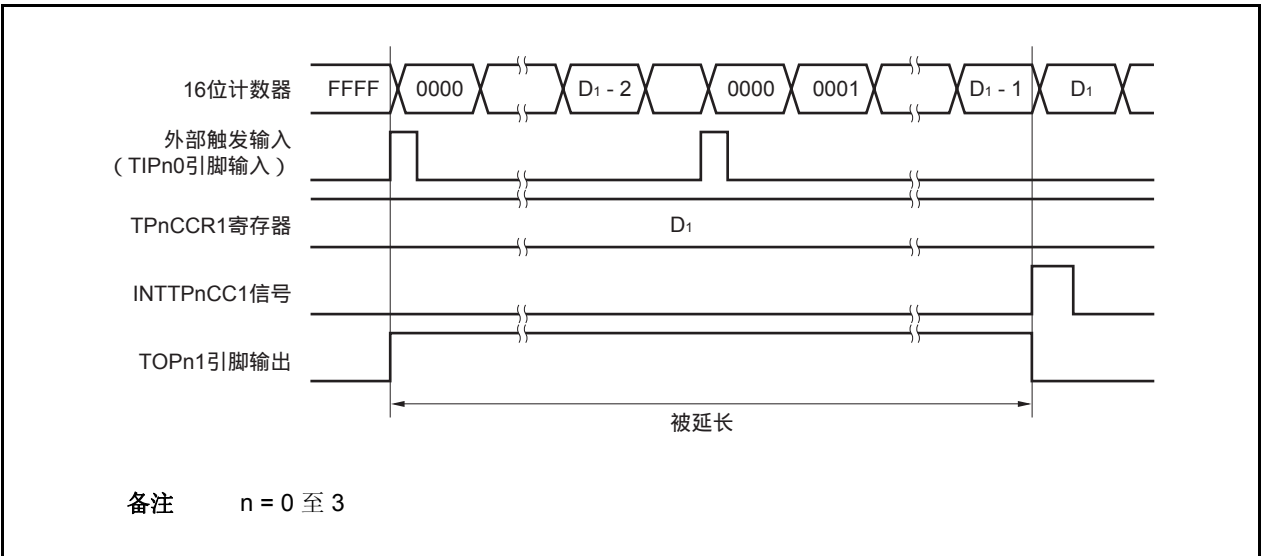
备注 n = 0 至 3

(c) 触发检测和 TPnCCR1 寄存器匹配之间的冲突

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

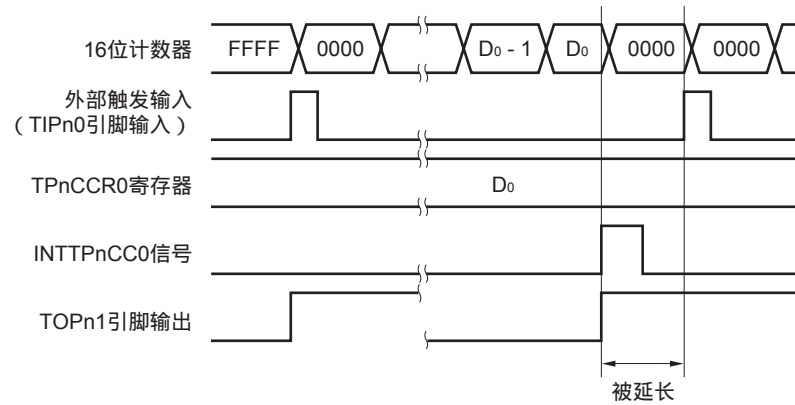


如果在 INTTPnCC1 信号产生之前检测到触发信号，那么 INTTPnCC1 信号将不会产生，此时，16 位计数器被清零并重新开始计数。TOPn1 引脚的输出也会继续保持为有效电平。这种情况下，PWM 波形的有效电平时间就会被延长。



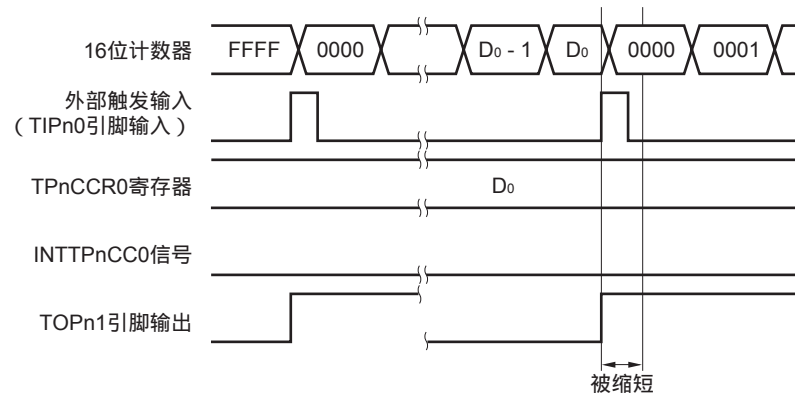
(d) 触发检测和 TPnCCR0 寄存器匹配之间的冲突

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



备注 n = 0 至 3

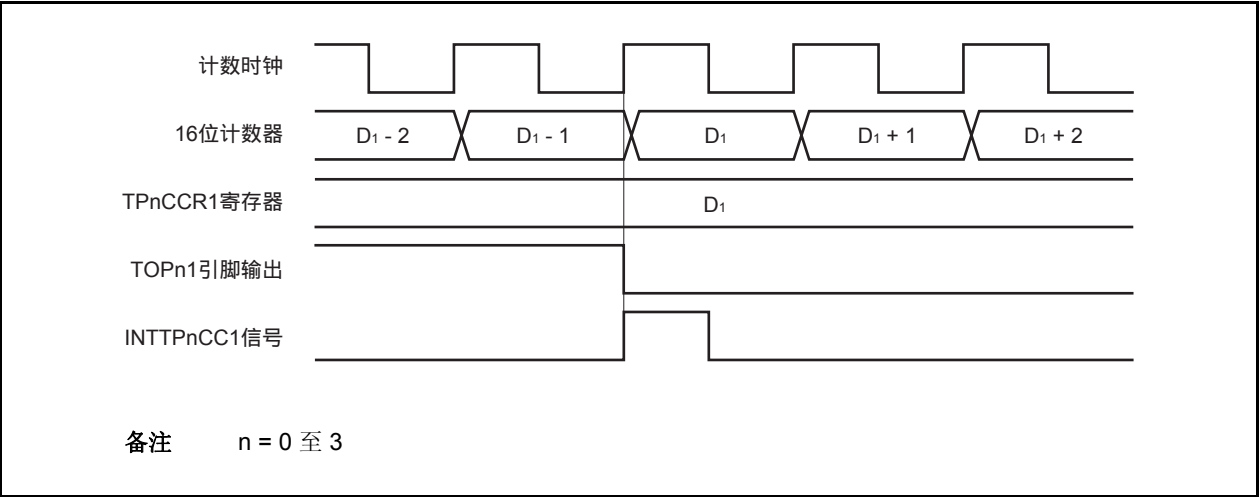
如果在 INTTPnCC0 信号产生之前检测到触发信号，那么 INTTPnCC0 信号将不会产生。16 位计数器将被清零并重新开始计数，TOPn1 引脚输出也会改变为有效电平。这种情况下，PWM 波形的非有效电平时间就会被缩短。



备注 n = 0 至 3

(e) 比较匹配中断请求信号的产生时序 (INTTPnCC1)

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



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

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

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

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

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

图 6-20. 单脉冲输出模式的配置图

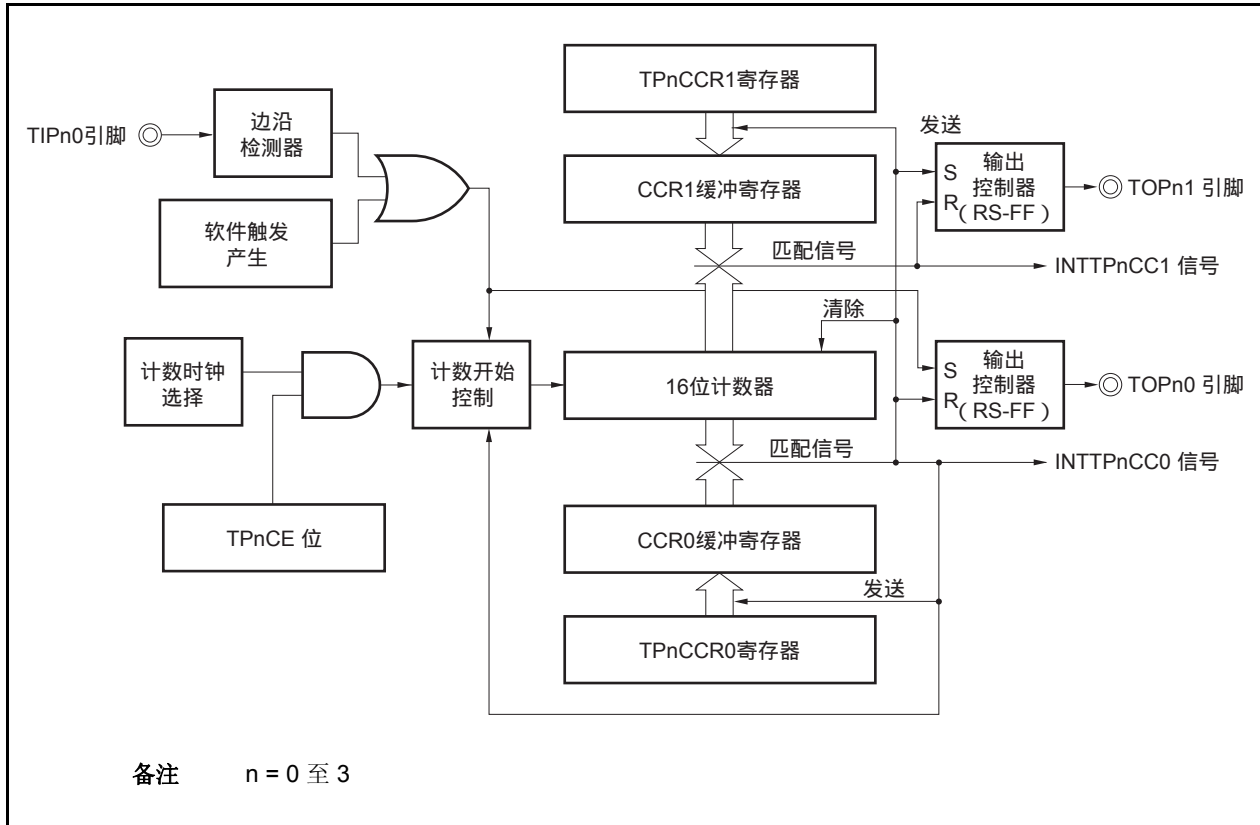
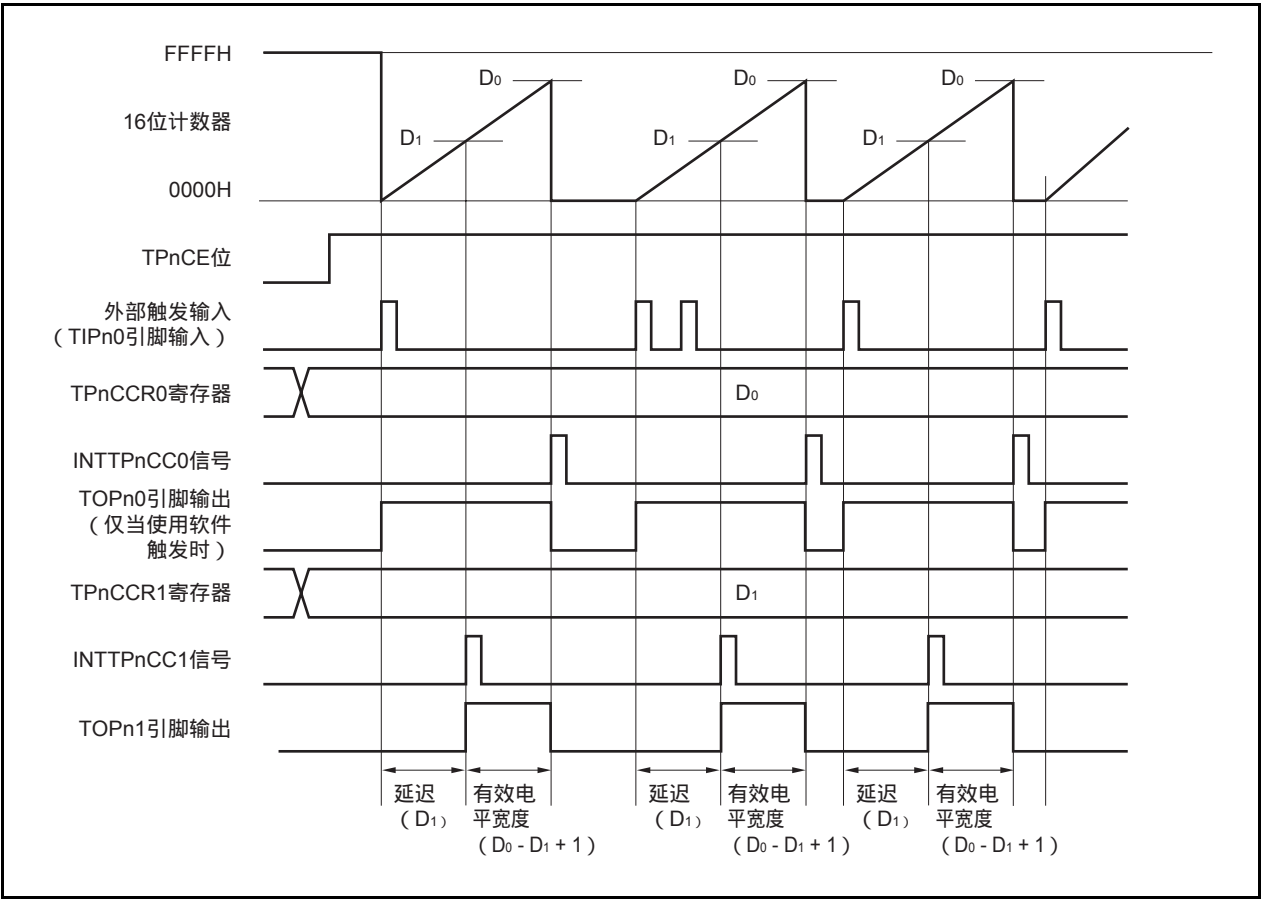


图 6-21. 单脉冲输出模式的基本时序



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

单脉冲波形的输出延迟时间以及有效电平宽度可由以下算式得出。

输出延迟时间 = (TPnCCR1 寄存器的设置值) × 计数时钟周期

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

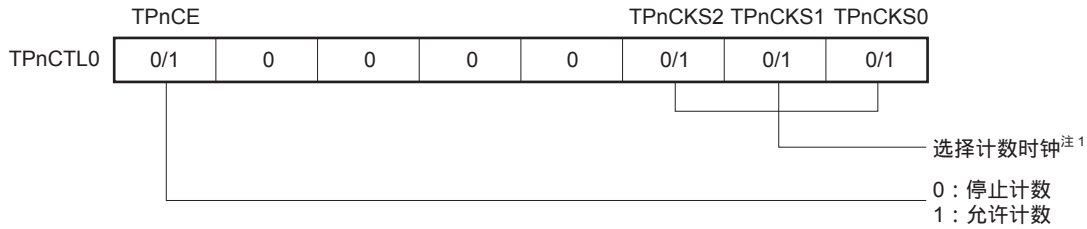
比较匹配中断请求信号 INTTPnCC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器中的值相匹配时的下一个计数时钟时产生。比较匹配中断请求信号 INTTPnCC1 会在 16 位计数器的计数值与 CCR1 缓冲寄存器中的值相匹配时产生。

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

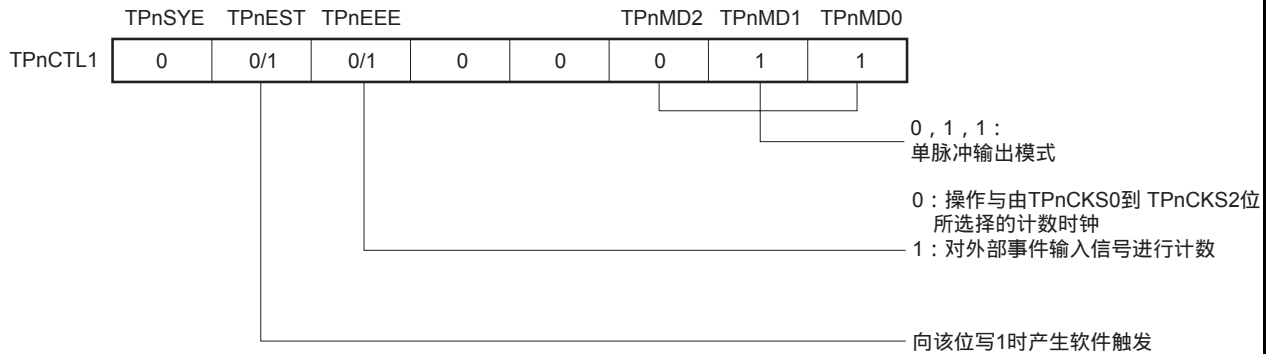
备注 n = 0 至 3
 m = 0, 1

图 6-22. 单脉冲输出模式的寄存器设置 (1/2)

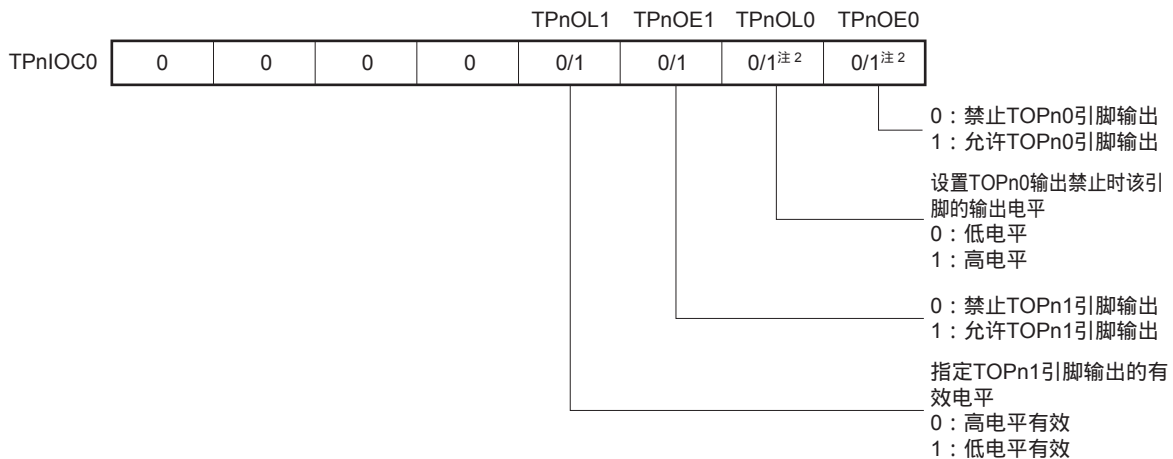
(a) TMPn 控制寄存器 0 (TPnCTL0)



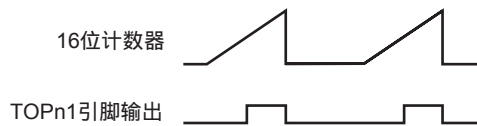
(b) TMPn 控制寄存器 1 (TPnCTL1)



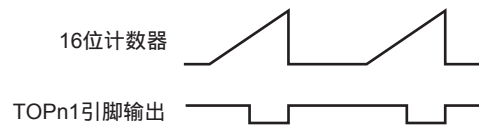
(c) TMPn I/O 控制寄存器 0 (TPnIOC0)



· 当TPnOL1位 = 0时

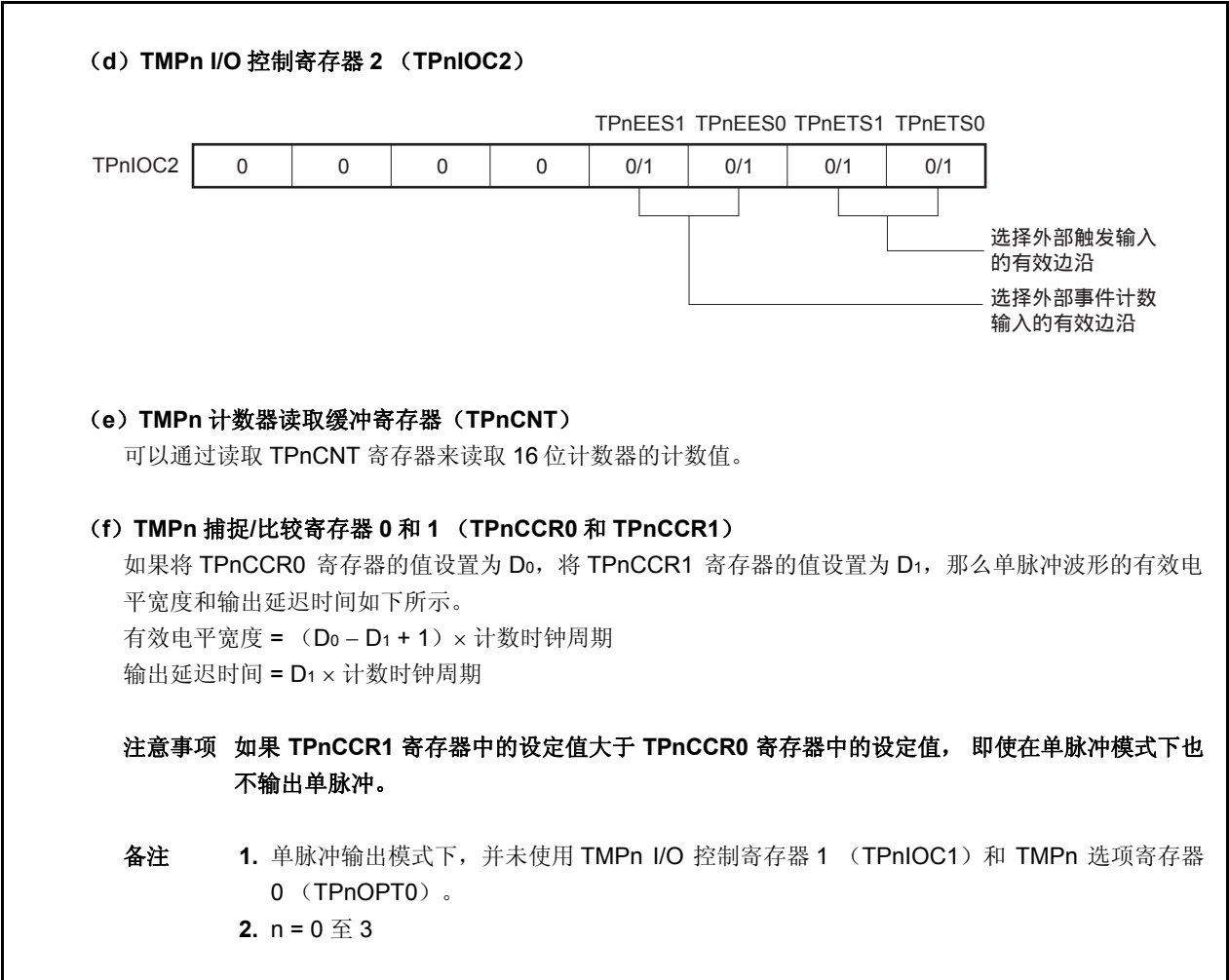


· 当TPnOL1位 = 1



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

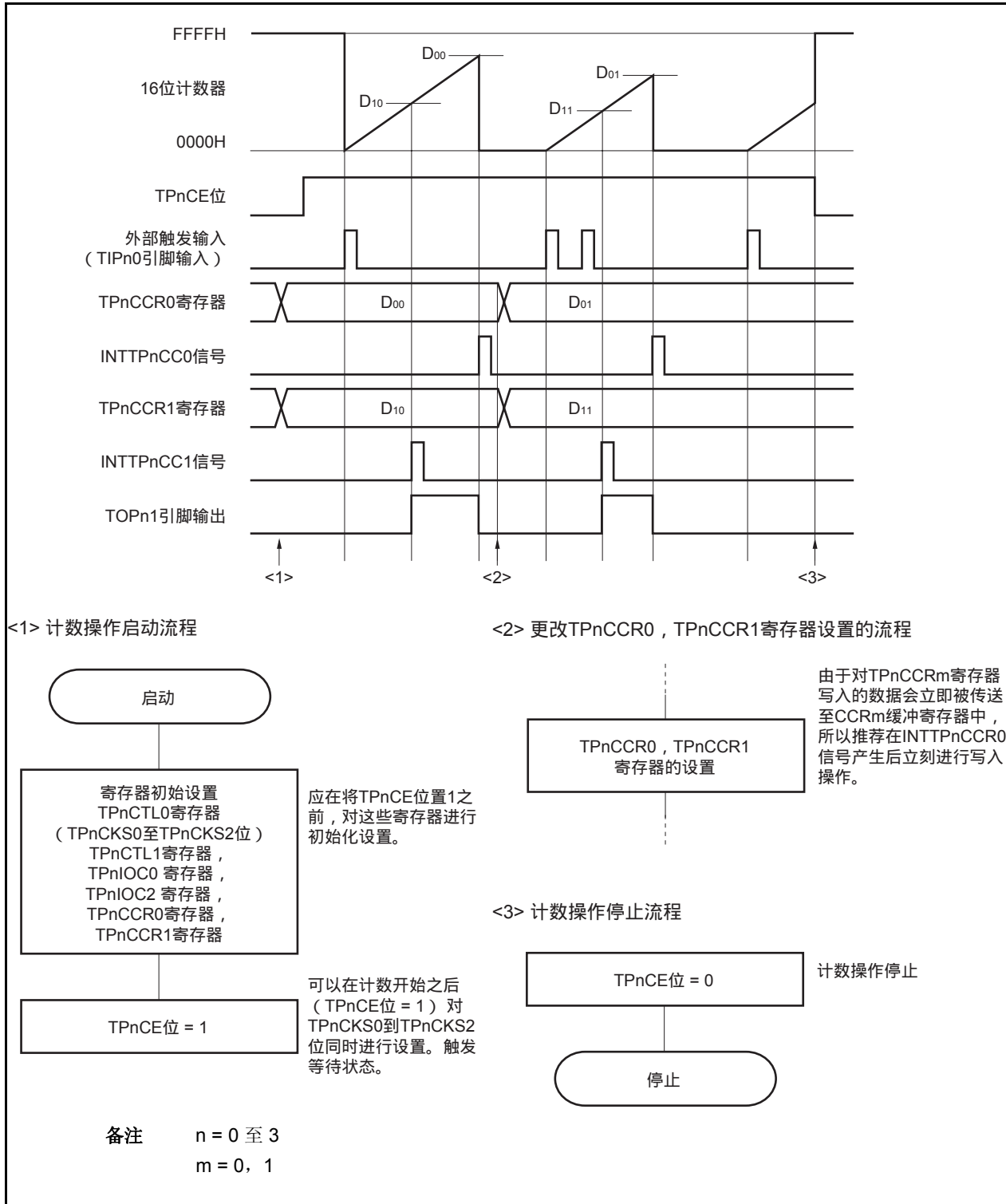
图 6-22. 单脉冲输出模式的寄存器设置 (2/2)



<R>

(1) 单脉冲输出模式的操作流程

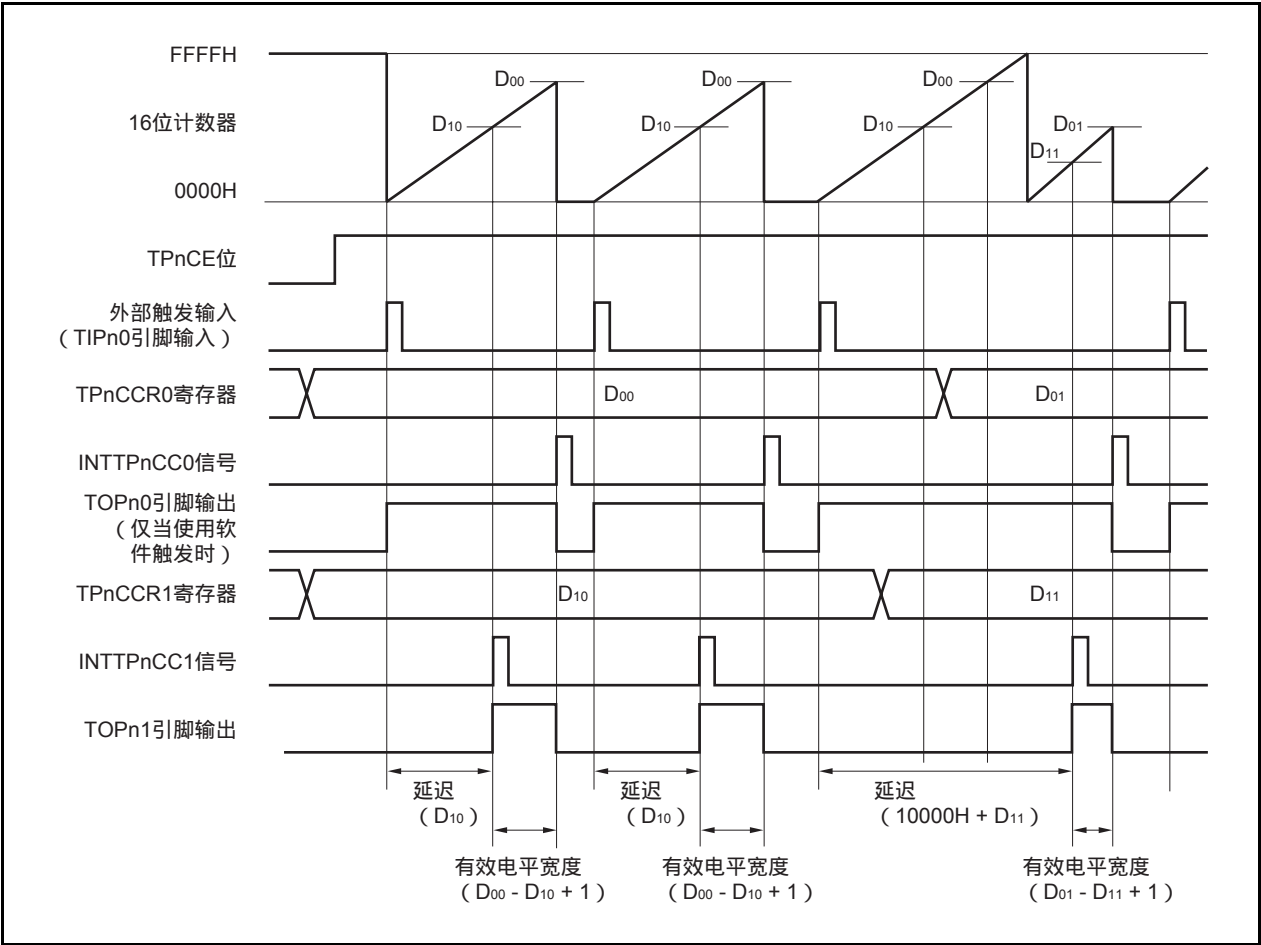
图 6-23. 单脉冲输出模式的软件操作流程



(2) 单脉冲输出模式的操作时序

(a) 改变 TPnCCRm 寄存器时需要注意的事项

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

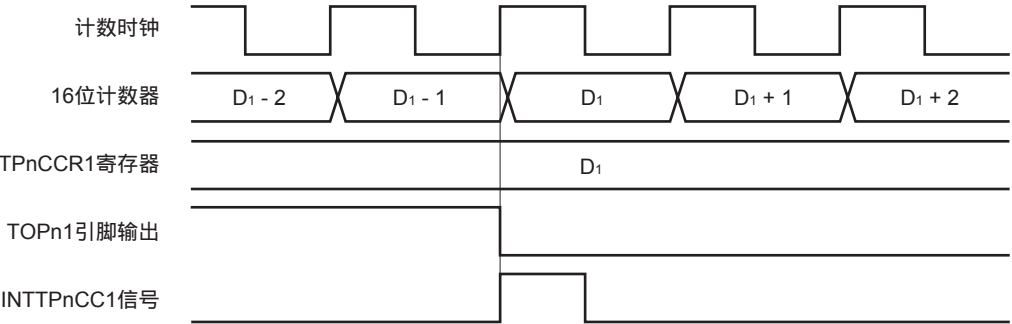


当将 TPnCCR0 寄存器的值从 D00 改写为 D01，TPnCCR1 寄存器的值从 D10 改写为 D11 时，这里假设 $D_{00} > D_{01}$ 且 $D_{10} > D_{11}$ ，如果 TPnCCR1 寄存器被改写时，16 位计数器的计数值大于 D11 且小于 D10，并且 TPnCCR0 寄存器被改写时，计数器的计数值大于 D01 且小于 D00，两个设定值都将在改写后立即生效并与计数值进行比较。此时计数器会计数到 FFFFH 然后从 0000H 重新开始计数。当计数值与 D11 匹配时，将产生 INTTPnCC1 信号，同时 TOPn1 引脚电平变为有效电平。当计数值与 D01 匹配时，将产生 INTTPnCC0 信号，同时 TOPn1 引脚电平变为非有效电平并且计数器停止计数。
因此可见，此时定时器输出的单脉冲波形的延迟时间或有效电平宽度会与预期数值有所差异。

备注 $n = 0$ 至 3
 $m = 0, 1$

(b) 比较匹配中断请求信号的产生时序 (INTTPnCC1)

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



备注 $n = 0$ 至 3

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

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

备注 $n = 0$ 至 3

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

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

图 6-24. PWM 输出模式的配置图

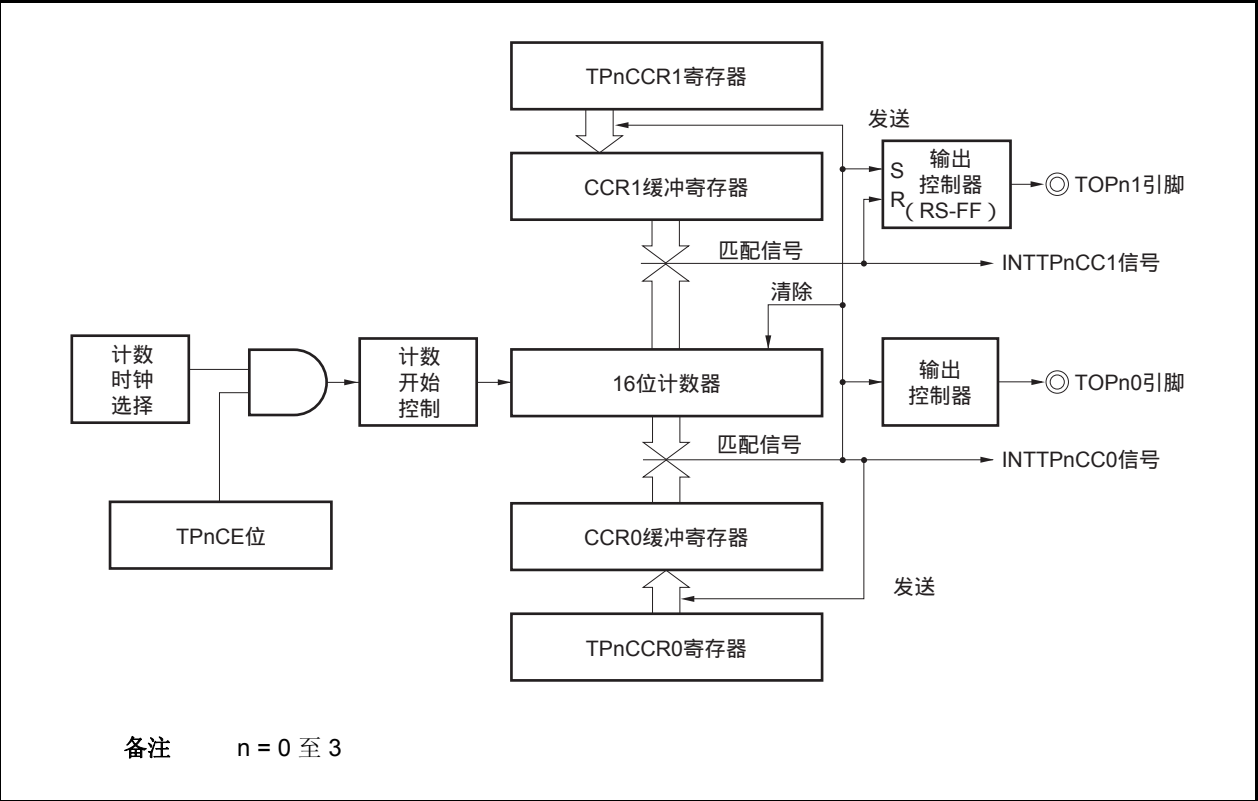
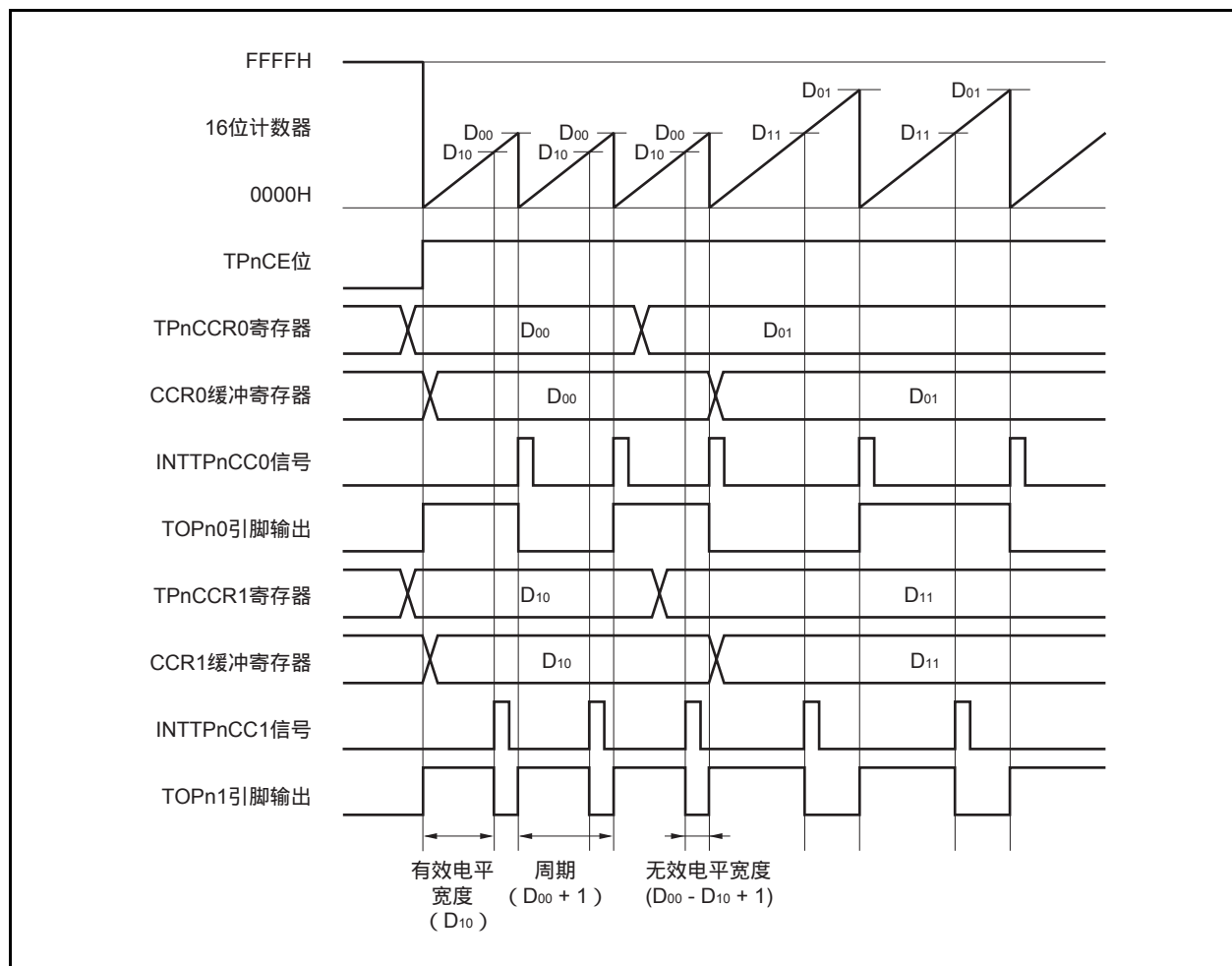


图 6-25. PWM 输出模式的基本时序



TPnCE 位被置 1 后，16 位计数器从 FFFFH 清零为 0000H 并开始计数，同时从 TOPn1 引脚输出一个 PWM 波形。PWM 波形的有效电平宽度，周期以及占空比系数可由以下算式得出。

有效电平宽度 = (TPnCCR1 寄存器的值) × 计数时钟周期

周期 = (TPnCCR0 寄存器的值 + 1) × 计数时钟周期

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

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

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

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

备注 n = 0 至 3, m = 0, 1

图 6-26. PWM 输出模式的寄存器设置 (1/2)

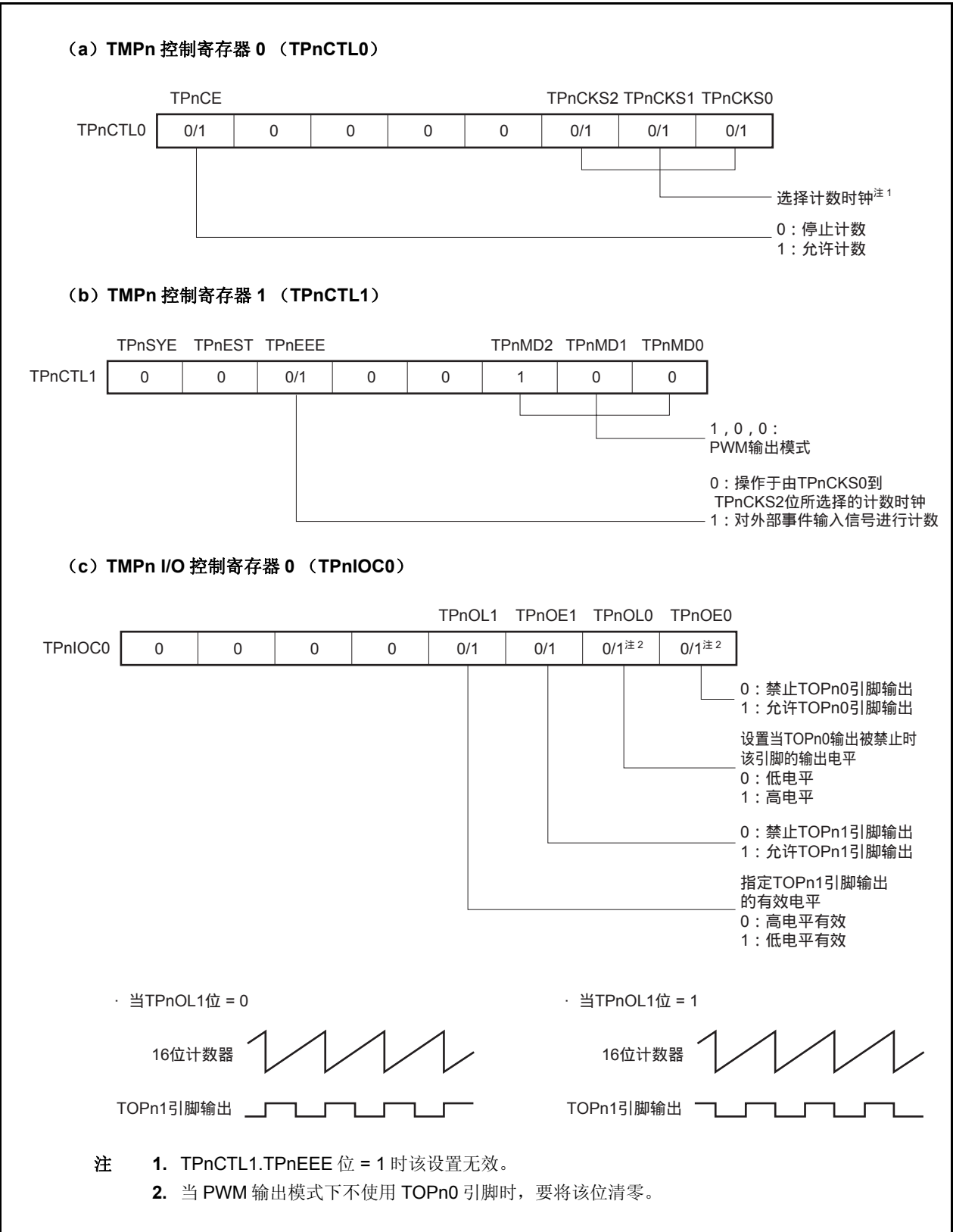
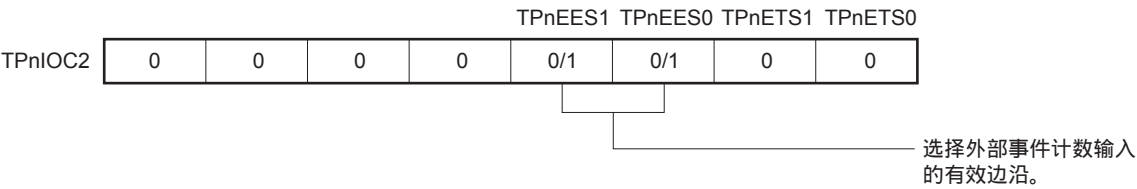


图 6-26. PWM 输出模式的寄存器设置 (2/2)

(d) TMPn I/O 控制寄存器 2 (TPnIOC2)



(e) TMPn 计数器读取缓冲寄存器 (TPnCNT)

可以通过读取 TPnCNT 寄存器来读取 16 位计数器的计数值。

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

如果将 TPnCCR0 寄存器的值设置为 D₀，将 TPnCCR1 寄存器的值设置为 D₁，那么 PWM 波形的周期和有效电平如下所示。

周期= (D₀ + 1) × 计数时钟周期
有效电平宽度 = D₁ × 计数时钟周期

- 备注**
- 1. PWM 输出模式下，并未使用 TMPn I/O 控制寄存器 1 (TPnIOC1) 和 TMPn 选项寄存器 0 (TPnOPT0)。
 - 2. n = 0 至 3

(1) PWM 输出模式的操作流程

图 6-27. PWM 输出模式的软件处理流程 (1/2)

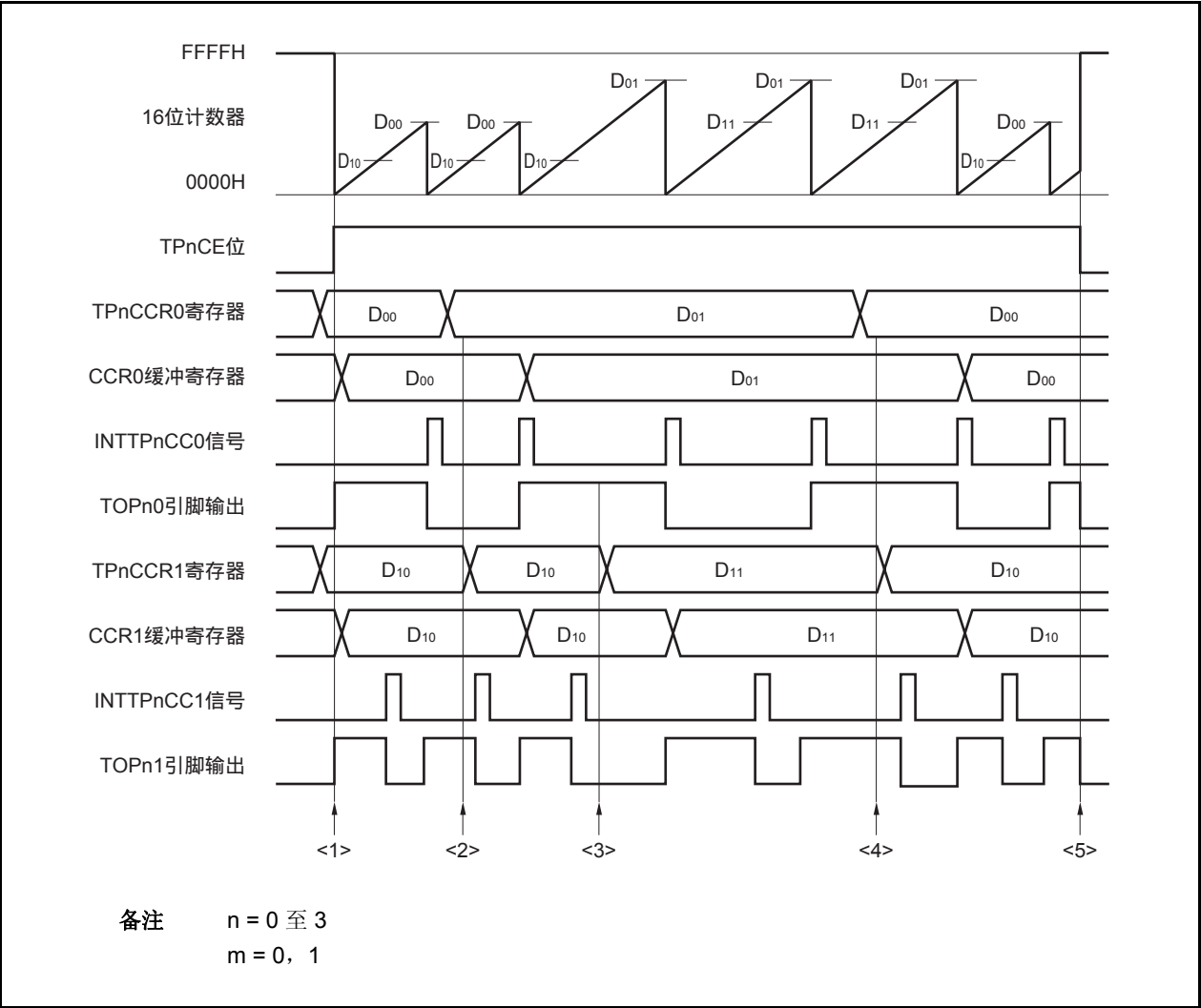
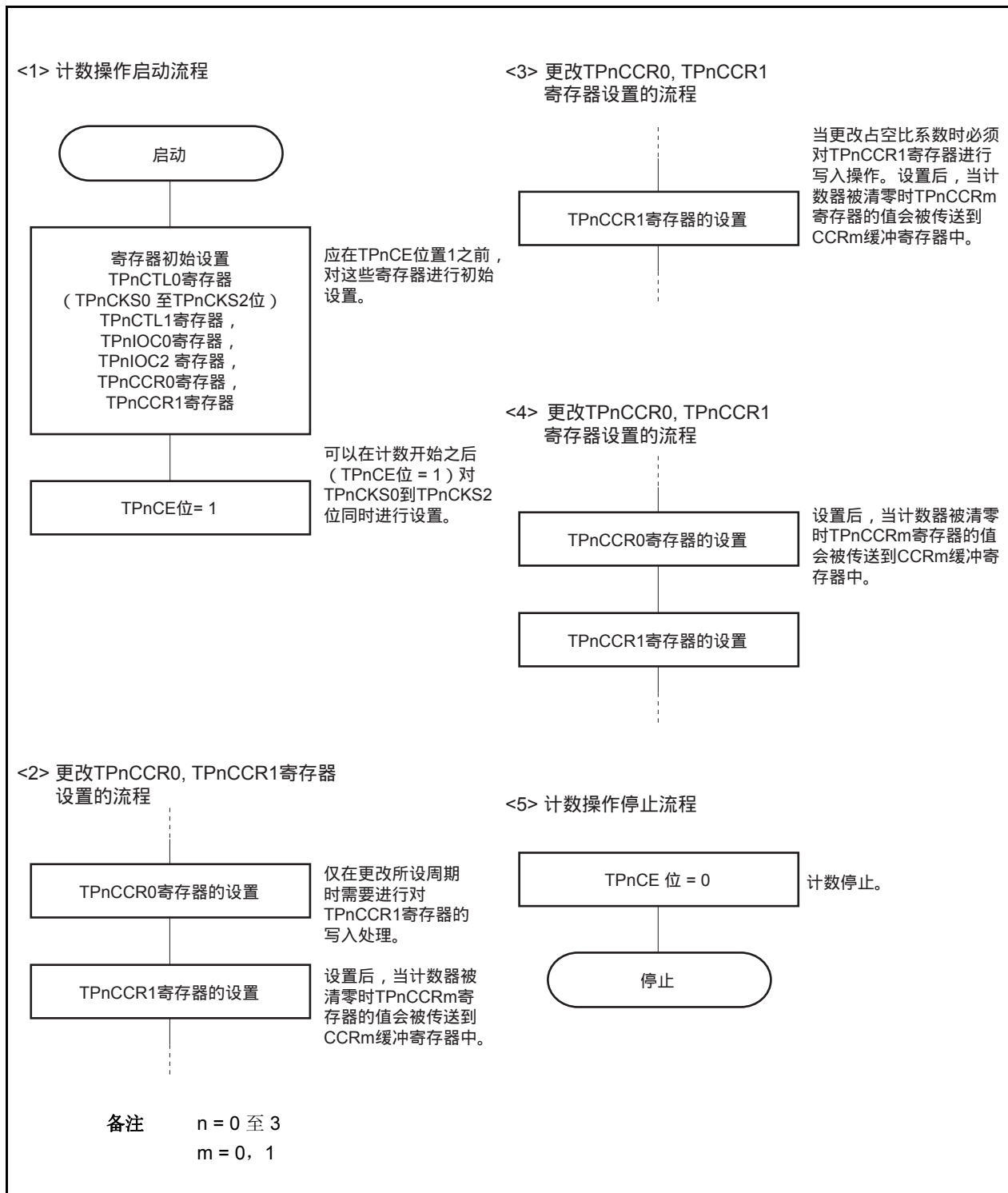


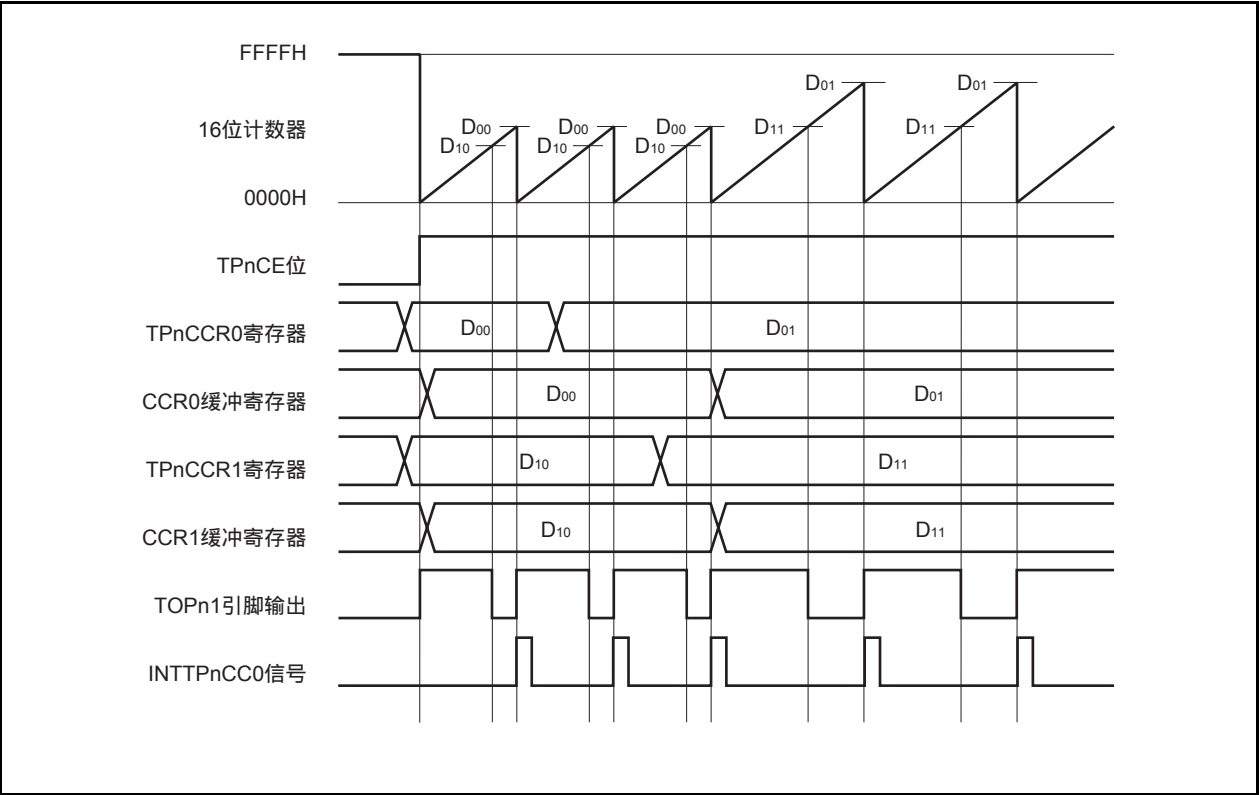
图 6-27. PWM 输出模式的软件处理流程 (2/2)



(2) PWM 输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

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



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

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

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

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

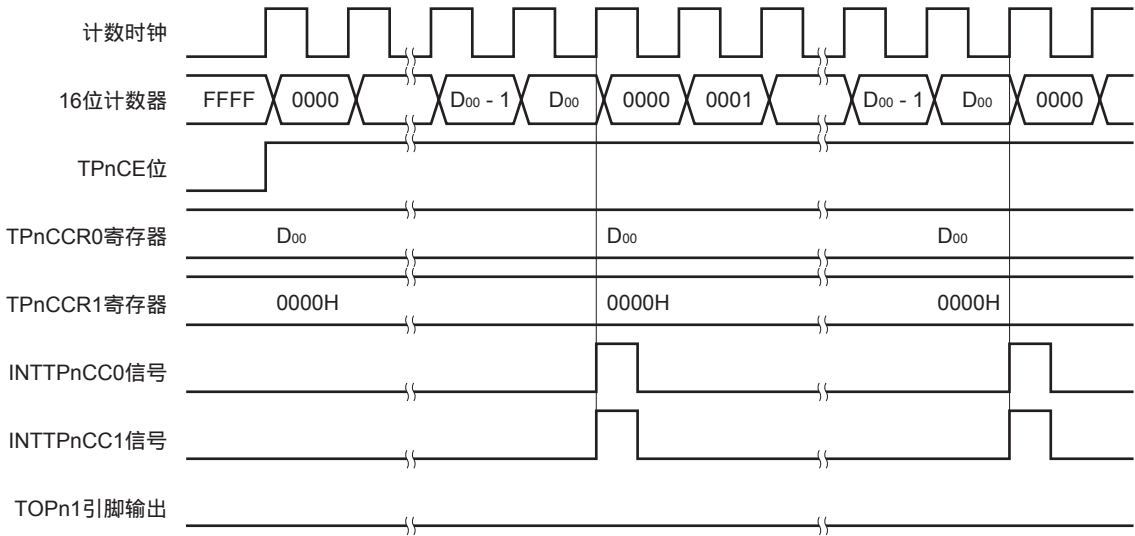
数据被写入 TPnCCR1 寄存器之后，16 位计数器被清零时，TPnCCRm 寄存器中的数值才会被同步传送至 CCRm 缓冲寄存器中，此时 CCRm 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

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

备注 n = 0 至 3, m = 0, 1

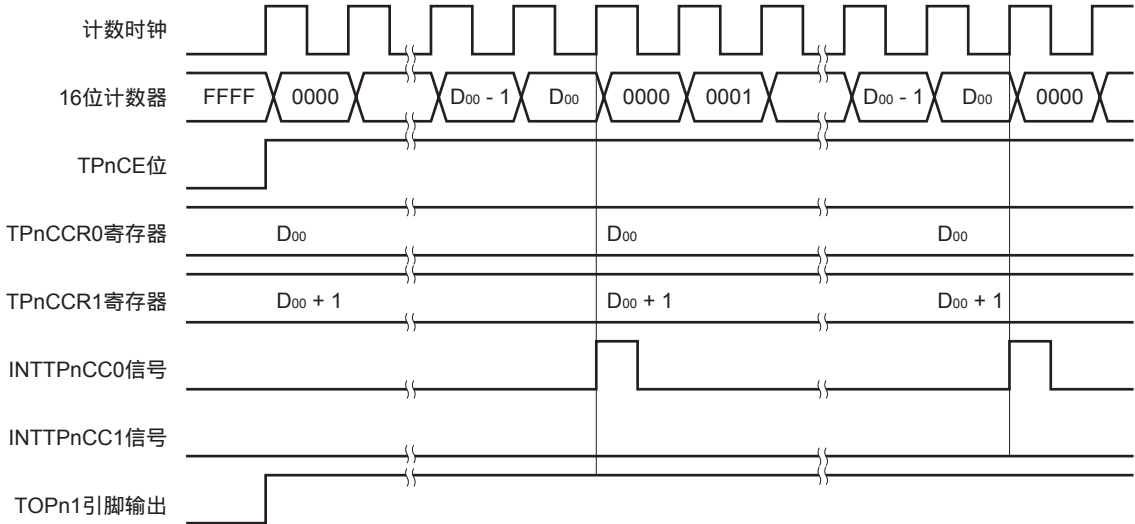
(b) PWM 波形的 0%/100% 输出

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



备注 n = 0 至 3

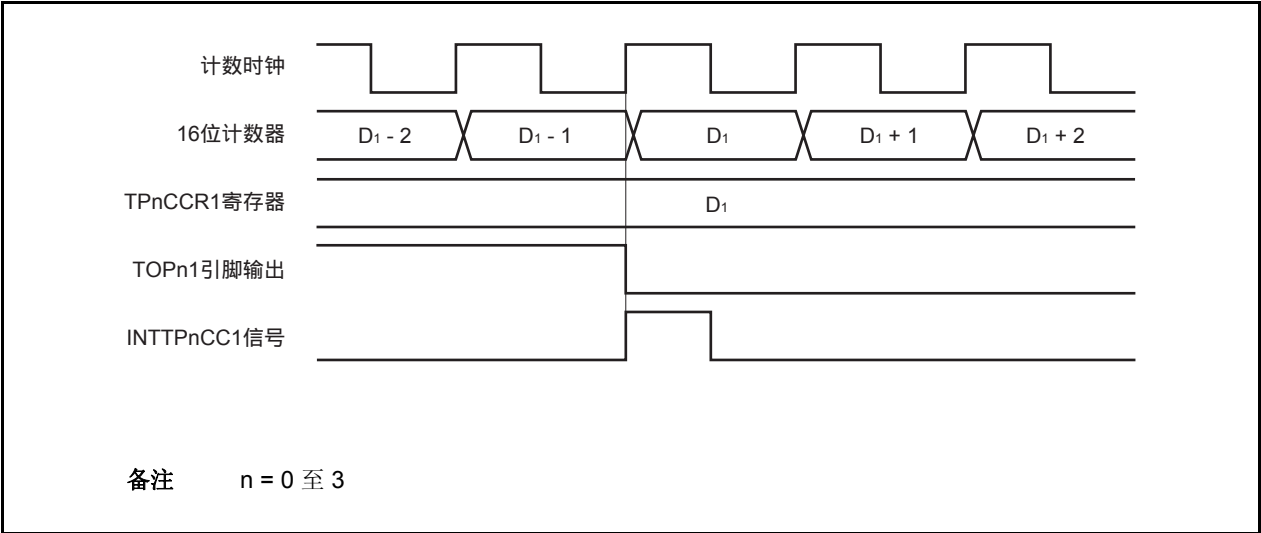
若要输出 100% 波形，则需将 TPnCCR1 寄存器设置为 (TPnCCR0 寄存器值 + 1)。TPnCCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。



备注 n = 0 至 3

(c) 比较匹配中断请求信号的产生时序 (INTTPnCC1)

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



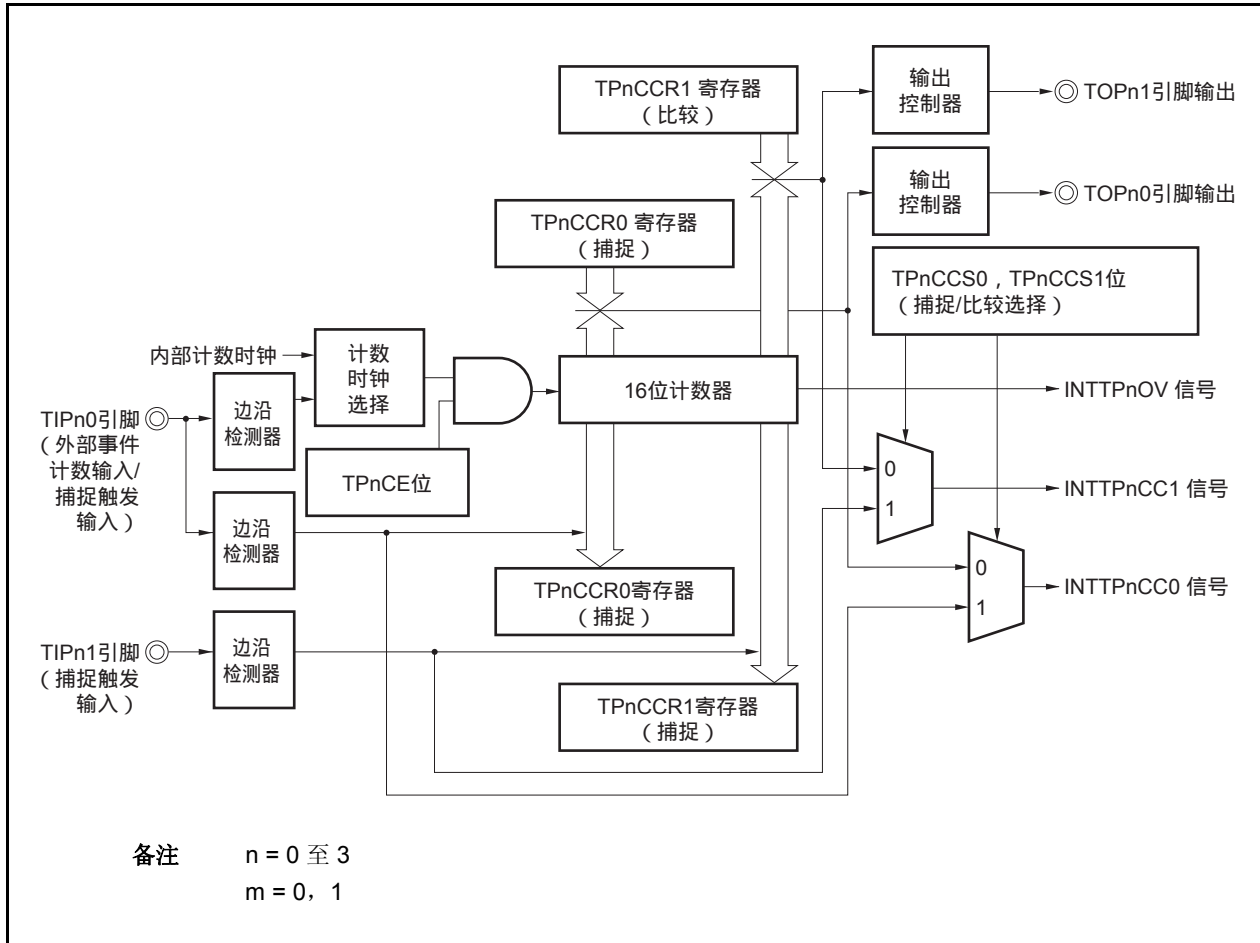
通常情况下, INTTPnCC1 信号会在 16 位计数器计数值与 TPnCCR1 寄存器的值匹配之后的下一个计数时钟同步产生。

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

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

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

图 6-28. 自由运行定时器模式的配置图

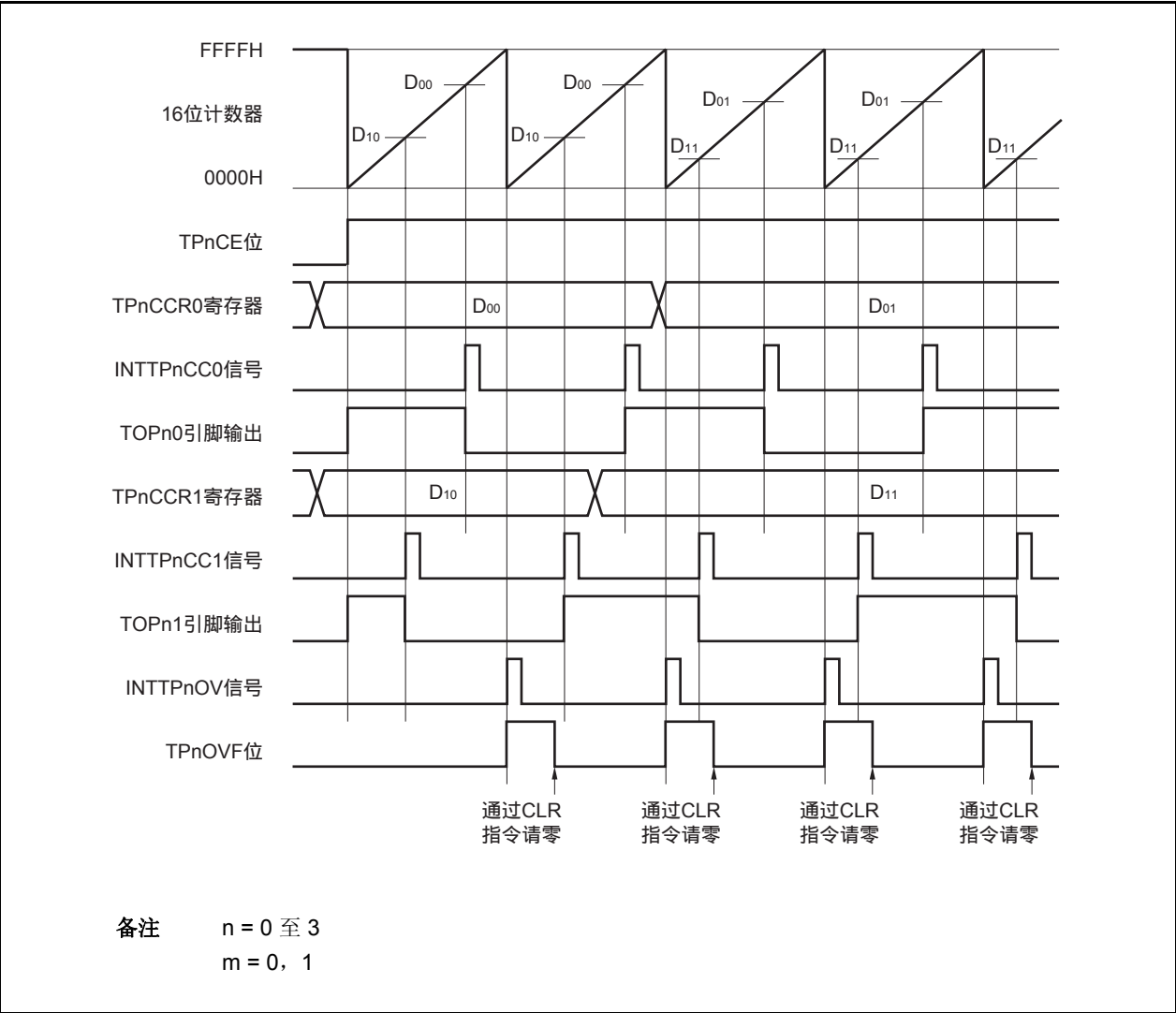


TPnCE 位被置 1 后，16 位计数器开始计数，同时 TOPn0 和 TOPn1 引脚输出信号的电平反转。当 16 位计数器的计数值与 TPnCCRm 寄存器的值匹配时将产生一个比较匹配中断请求信号 (INTTPnCCm)，同时 TOPnm 引脚的输出电平将反转。

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

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

图 6-29. 自由运行定时器模式的基本时序（比较功能）



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

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

图 6-30. 自由运行定时器模式的基本时序 (捕捉功能)

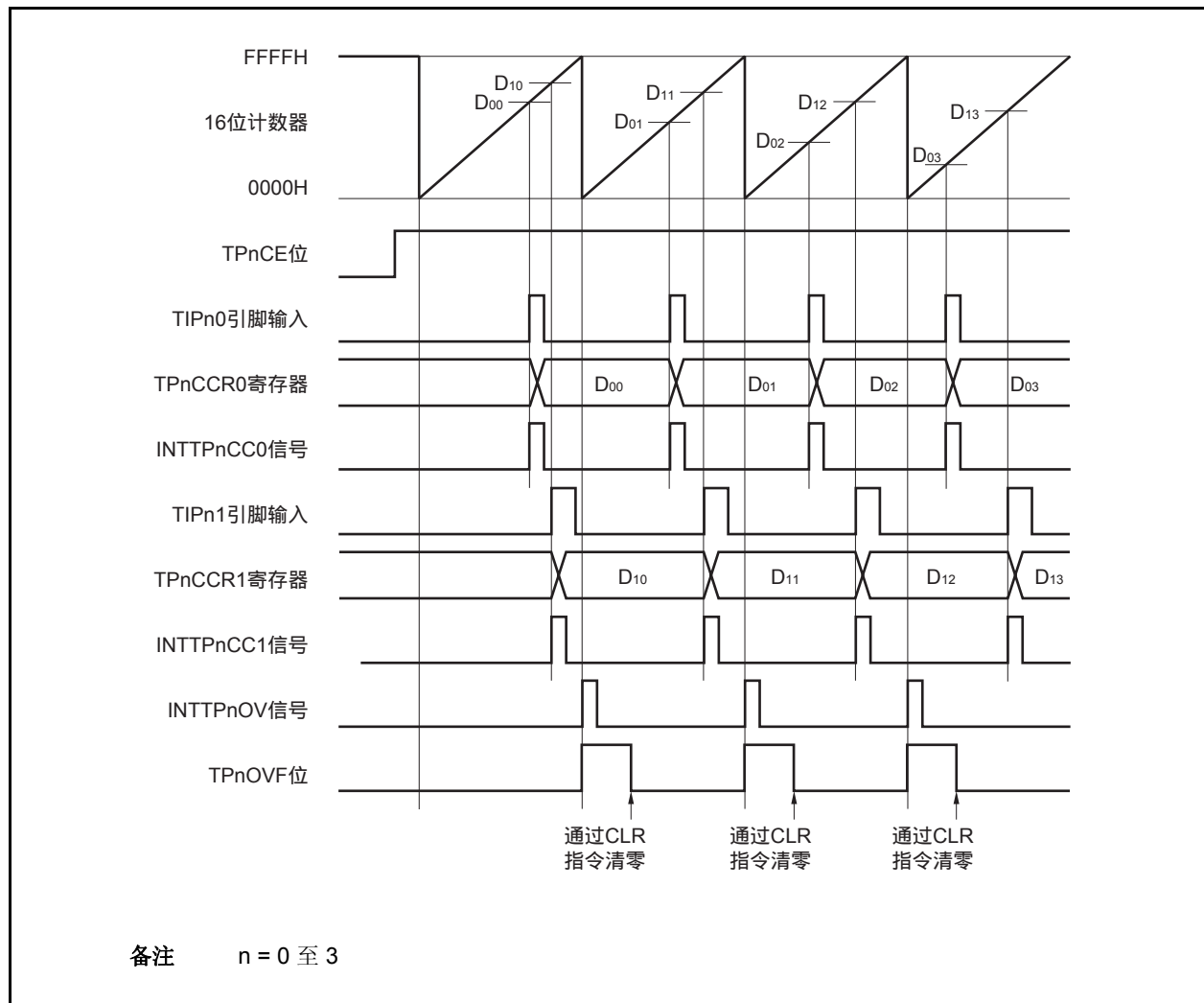


图 6-31. 自由运行定时器模式的寄存器设置 (1/2)

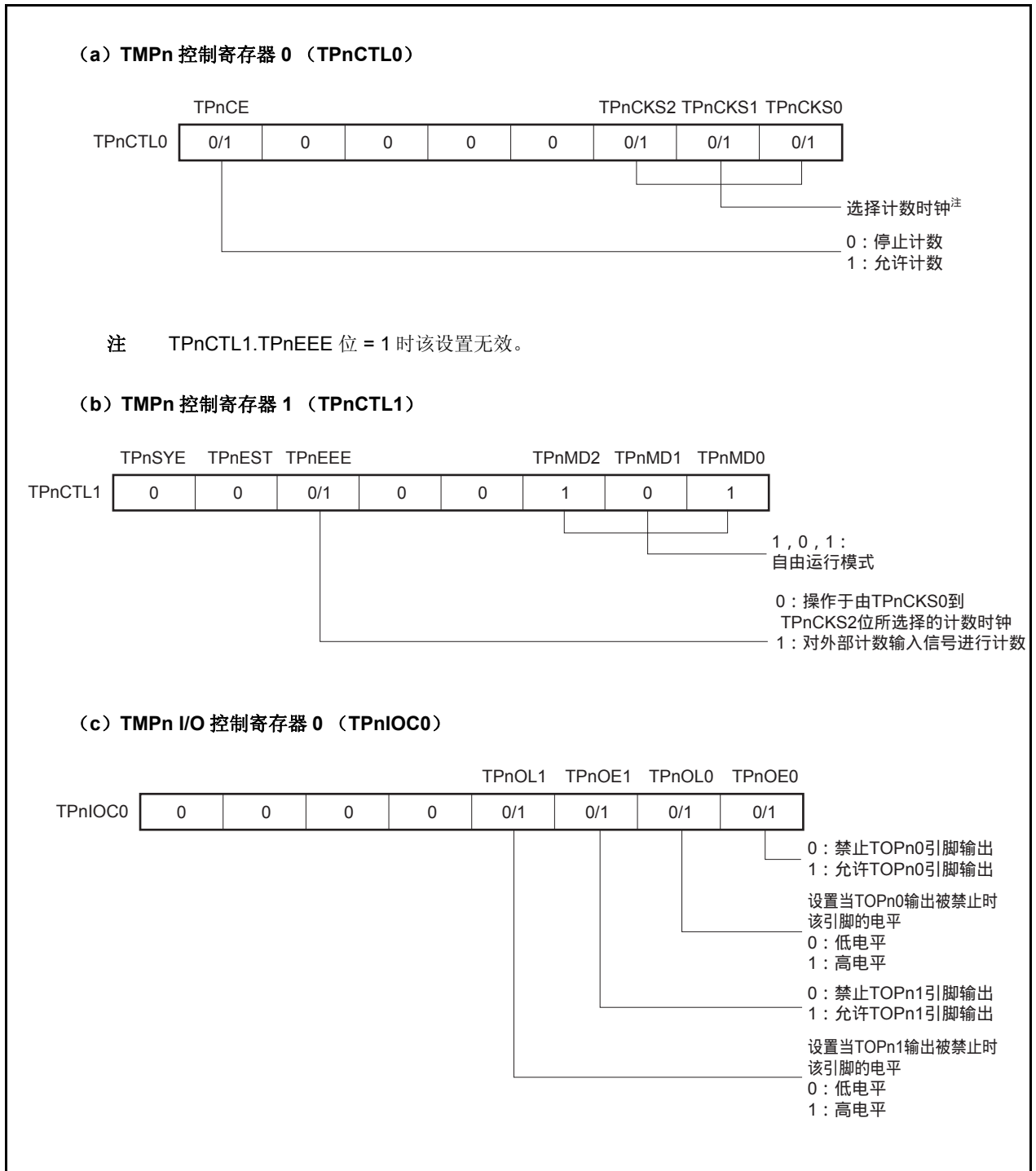
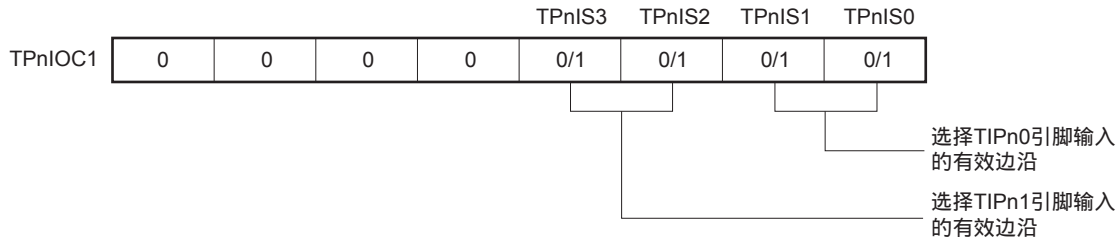
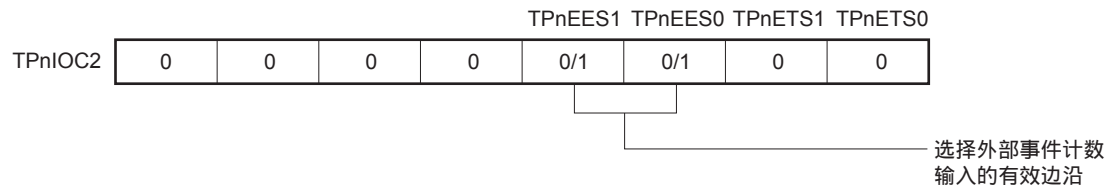


图 6-31. 自由运行定时器模式的寄存器设置 (2/2)

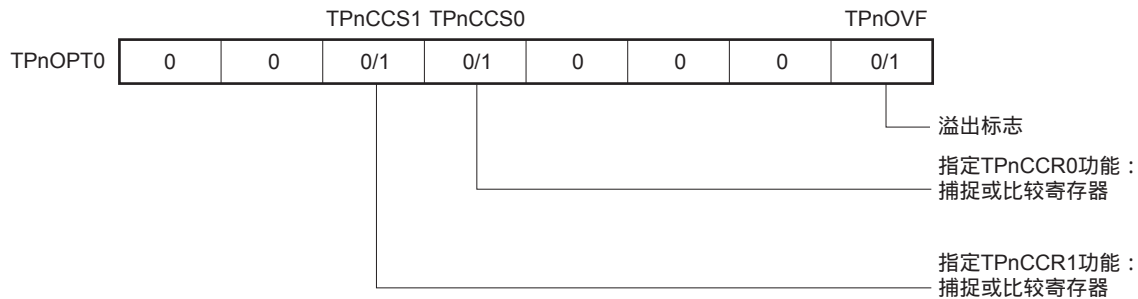
(d) TMPn I/O 控制寄存器 1 (TPnIOC1)



(e) TMPn I/O 控制寄存器 2 (TPnIOC2)



(f) TMPn 选项寄存器 0 (TPnOPT0)



(g) TMPn 计数器读取缓冲寄存器 (TPnCNT)

可以通过 TPnCNT 寄存器读取 16 位定时器的值。

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

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

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

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

备注 n = 0 至 3
 m = 0, 1

(1) 自由运行定时器模式的流程图

(a) 当捕捉/比较寄存器用作比较寄存器时

图 6-32. 自由运行定时器模式的软件处理流程（比较功能）（1/2）

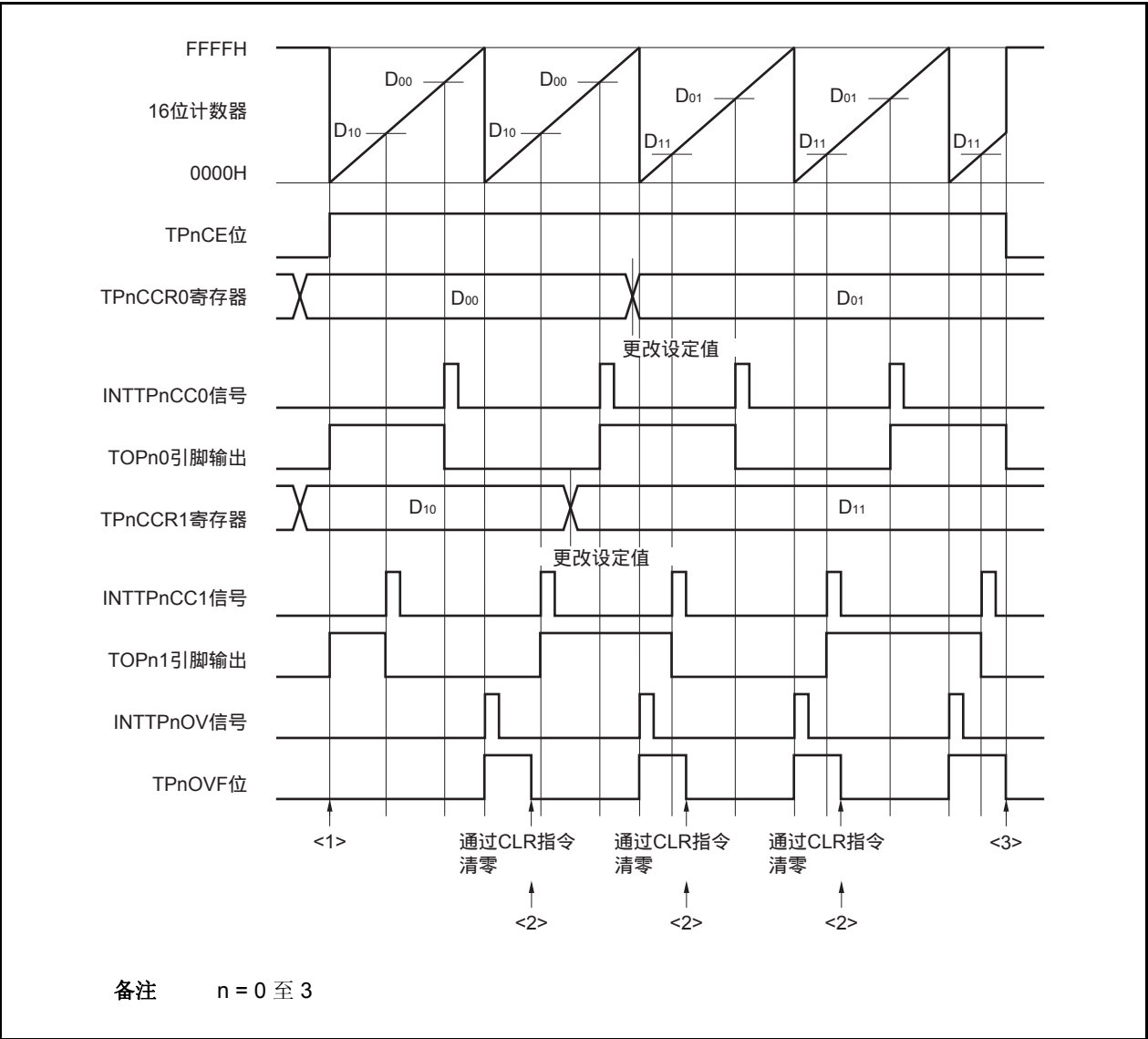
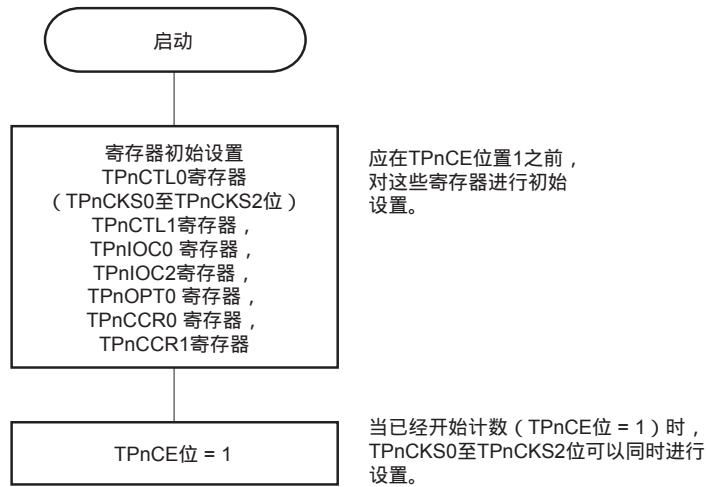
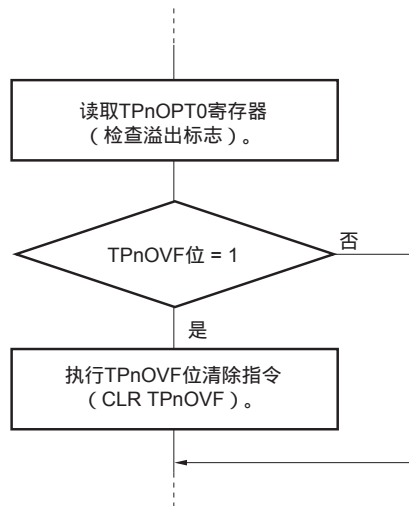


图 6-32. 自由运行定时器模式的软件处理流程（比较功能）（2/2）

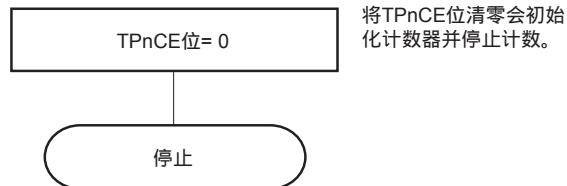
<1> 计数操作启动流程



<2> 溢出标志清除流程



<3> 计数操作停止流程



备注 n = 0 至 3

(b) 当捕捉/比较寄存器用作捕捉寄存器时

图 6-33. 自由运行定时器模式的软件处理流程（捕捉功能）（1/2）

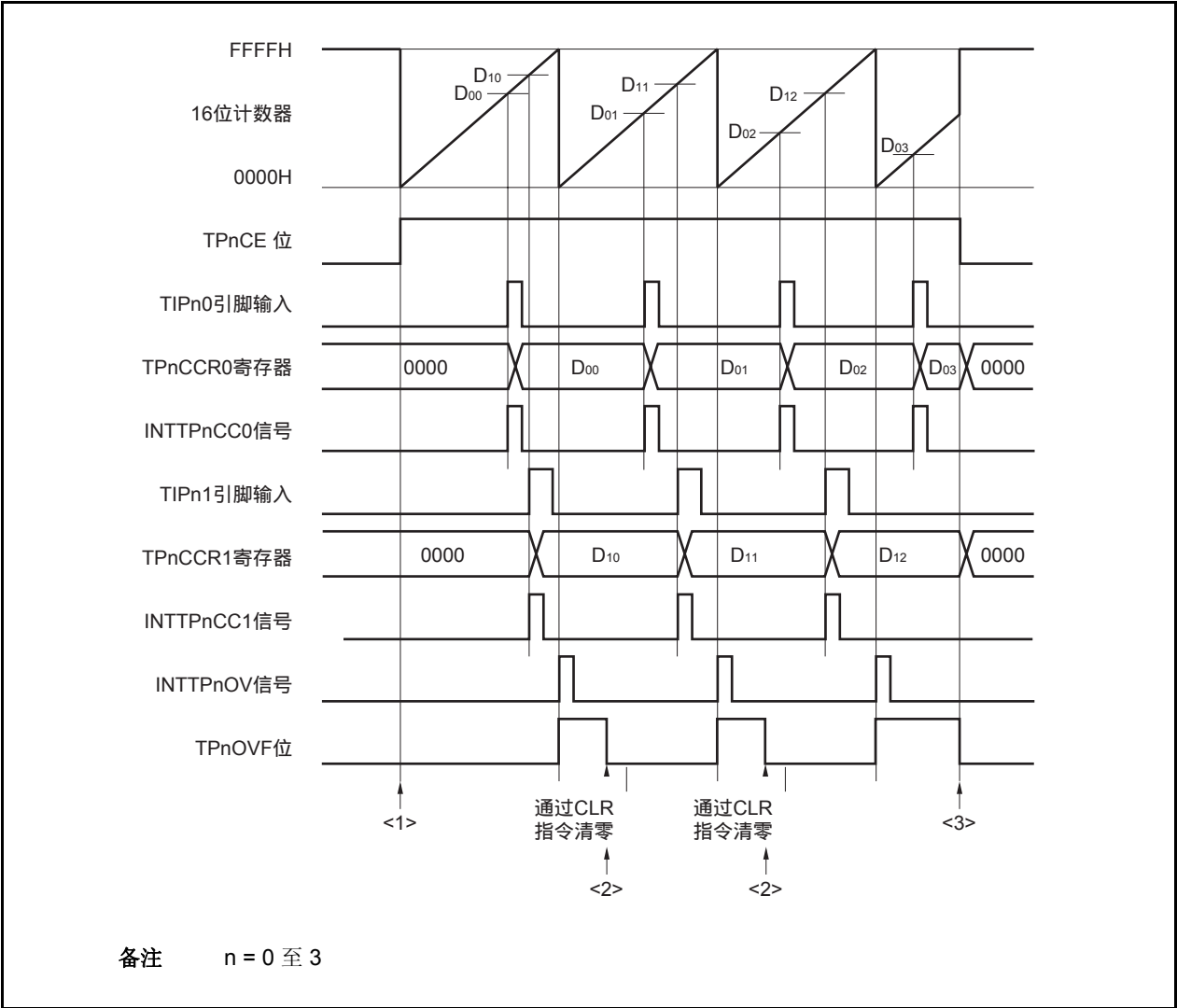
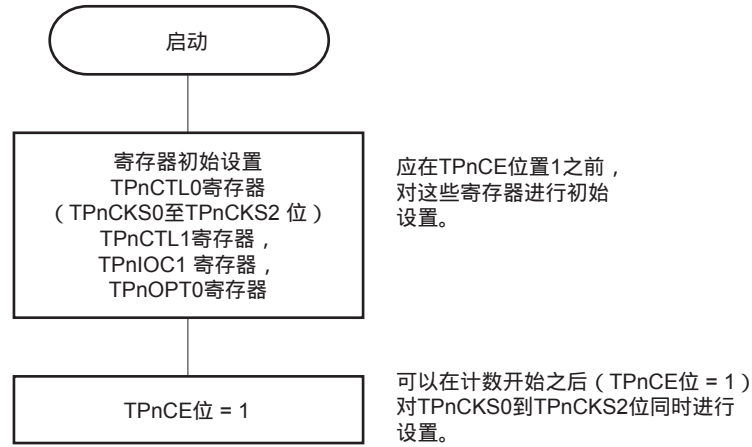
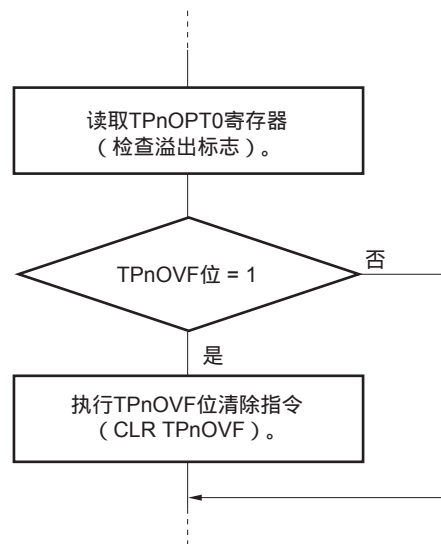


图 6-33. 自由运行定时器模式的软件处理流程（捕捉功能）（2/2）

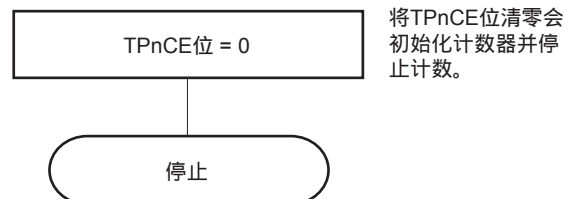
<1> 计数操作启动流程



<2> 溢出标志清除流程



<3> 计数操作停止流程

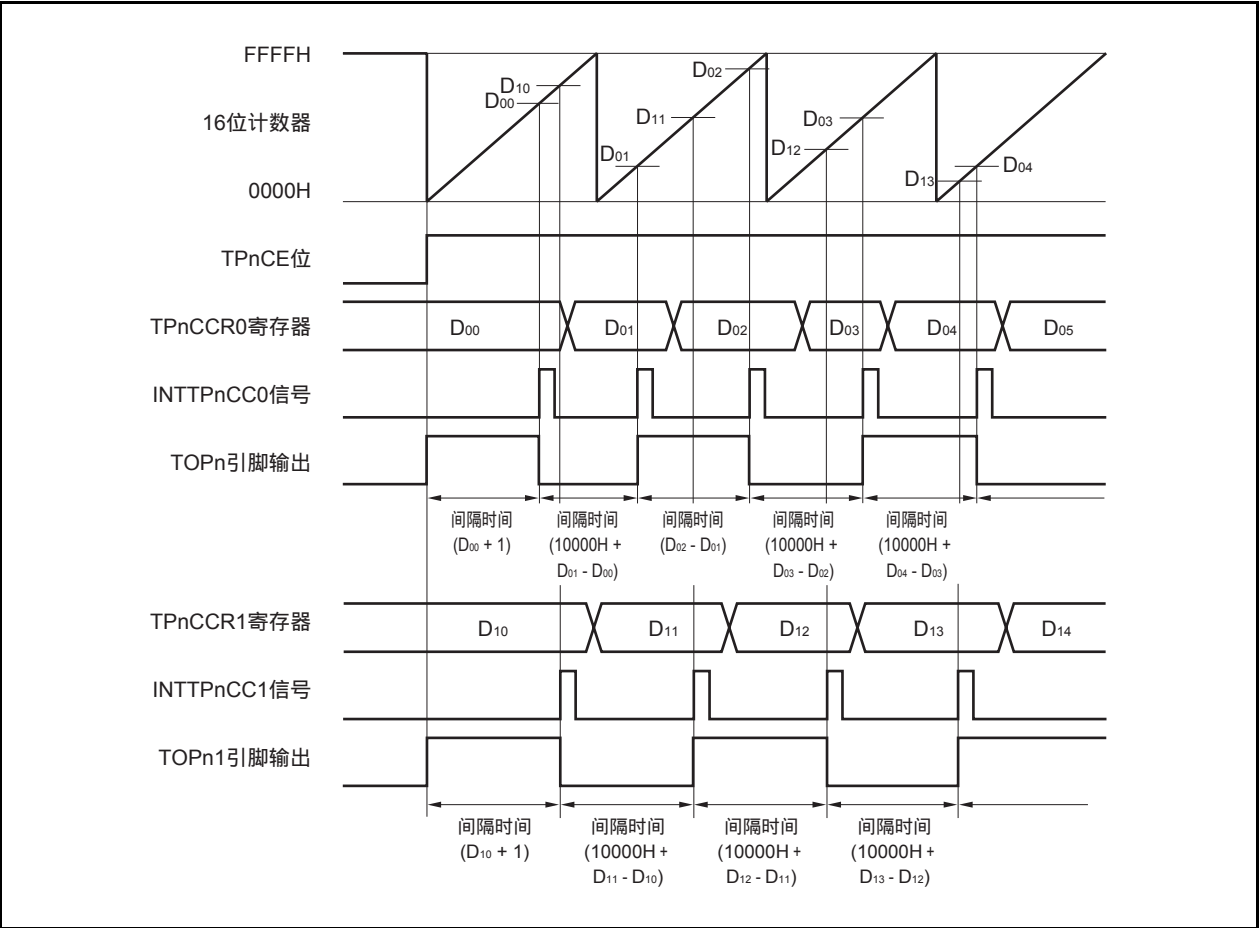


备注 n = 0 至 3

(2) 自由运行定时器模式的操作时序

(a) 使用比较寄存器功能的时间间隔操作

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



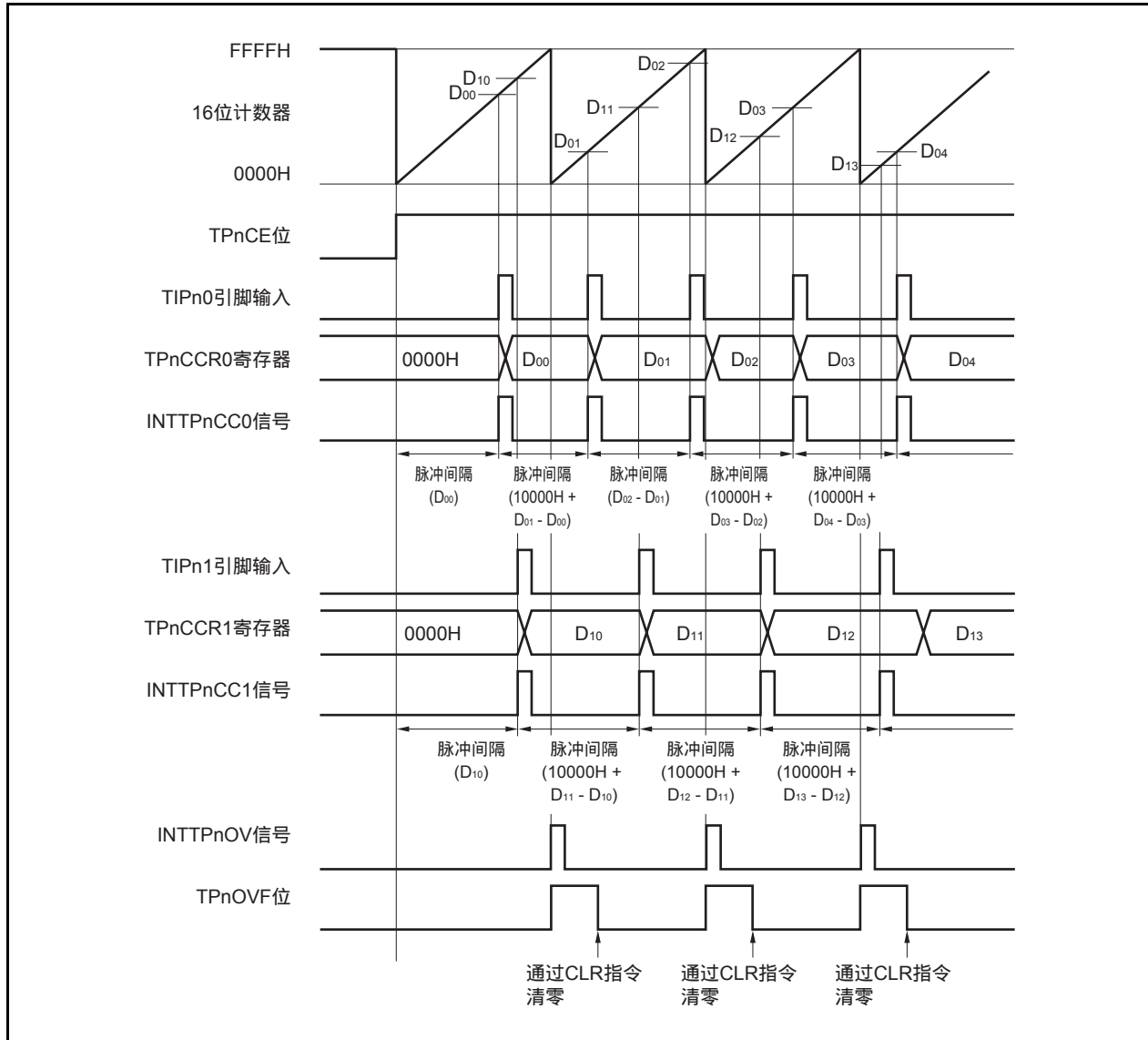
当使用自由运行定时器模式进行间隔时间操作时，可以在同一通道指定两个不同的间隔时间。
进行间隔时间操作时，必须在每次检测到 INTTPnCCm 信号后执行的中断服务程序中将相应的值设置到 TPnCCRm 寄存器。
向 TPnCCRm 寄存器中重新设置的值可由以下算式求出，这里假设间隔时间为“Dm”。

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

备注 n = 0 至 3
 m = 0, 1

(b) 使用捕捉寄存器进行脉宽测量

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

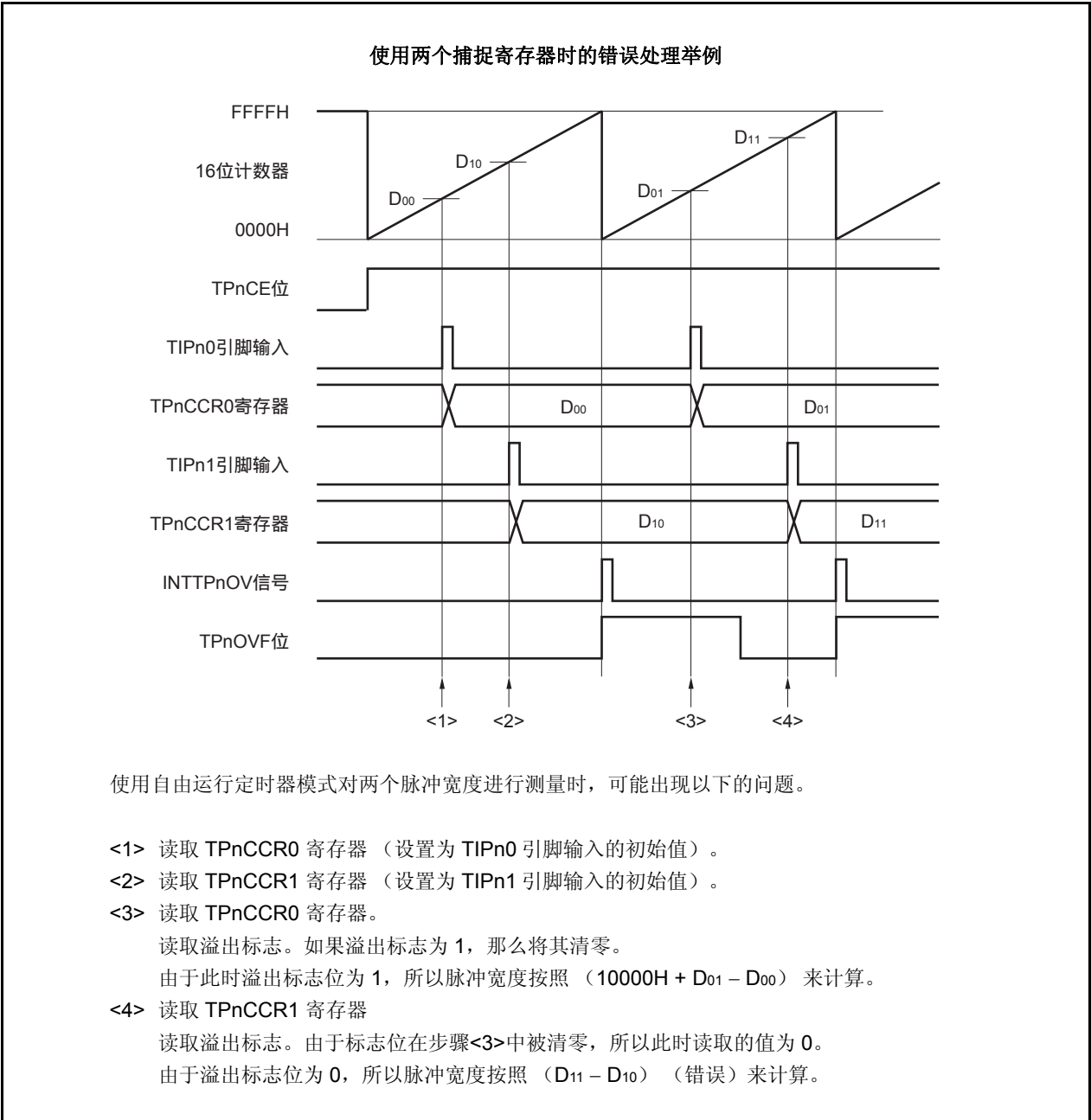


当使用自由运行定时器模式进行脉宽测量操作时, 可以在同一通道对两个不同的脉冲宽度进行测量。通过与 INTTPnCCm 信号同步读取 TPnCCRm 寄存器的值并计算该值与上一次读取值的差, 就可以求得脉冲的宽度。

备注 n = 0 至 3
 m = 0, 1

(c) 使用两个捕捉寄存器产生溢出时的操作

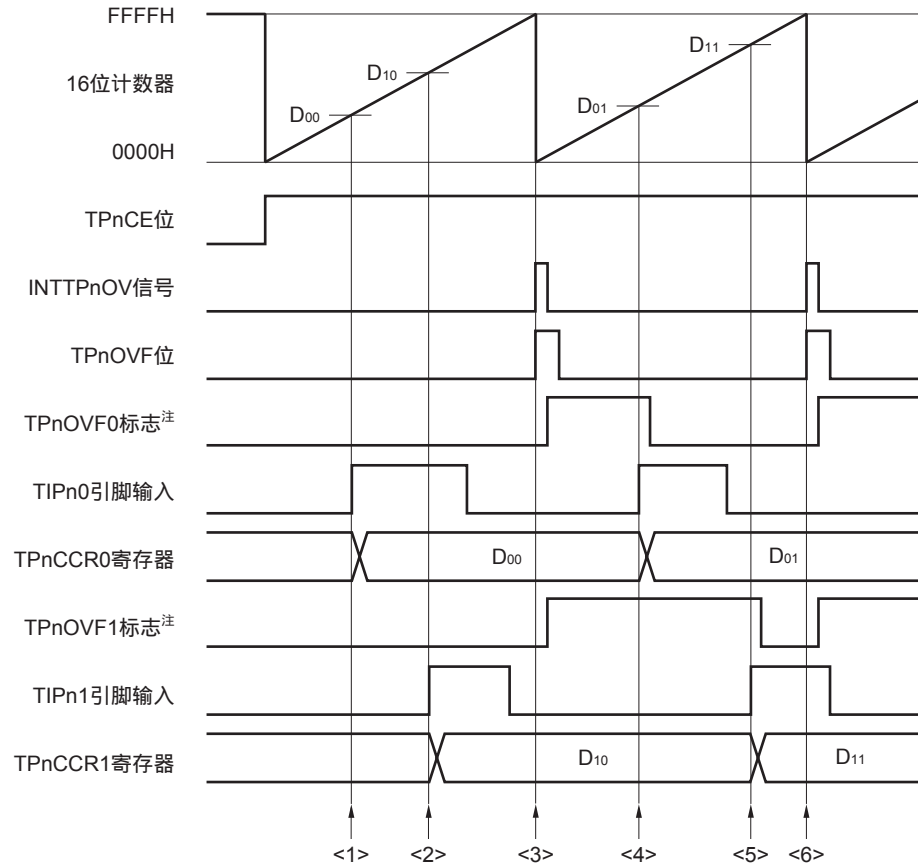
使用两个捕捉寄存器时，请务必慎重处理溢出标志位。首先，给出一个错误处理的例子，如下图所示。



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

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

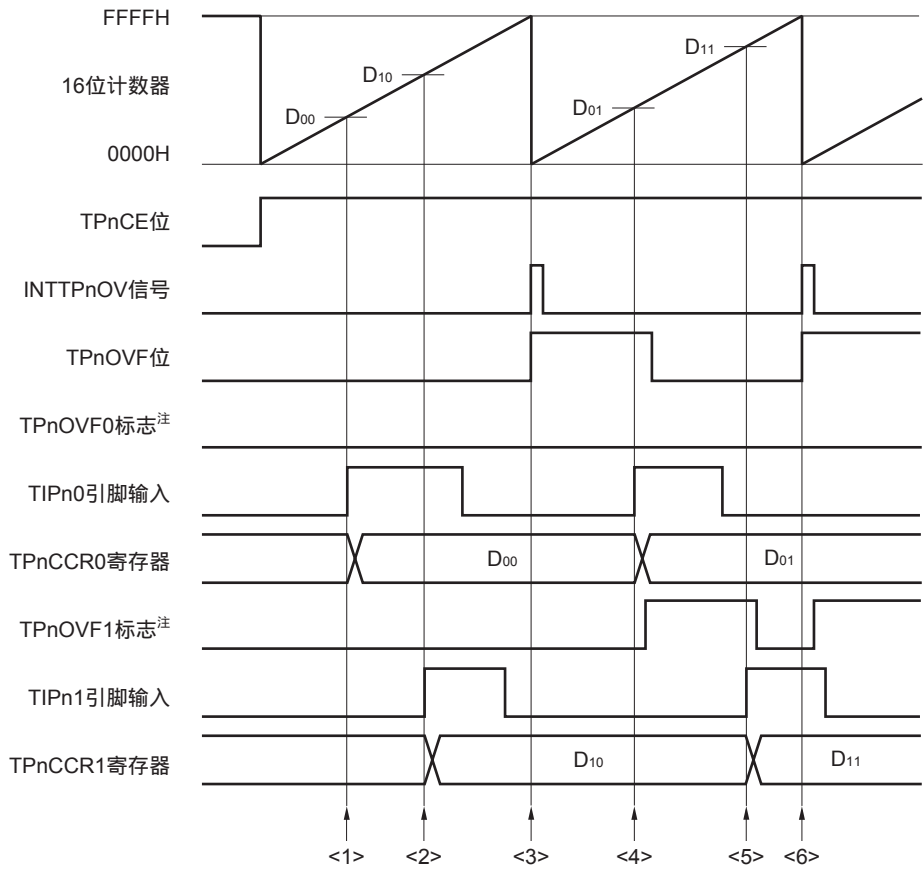
使用两个捕捉寄存器时的处理举例（利用溢出中断）



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

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

使用两个捕捉寄存器时的处理举例（不利用溢出中断）



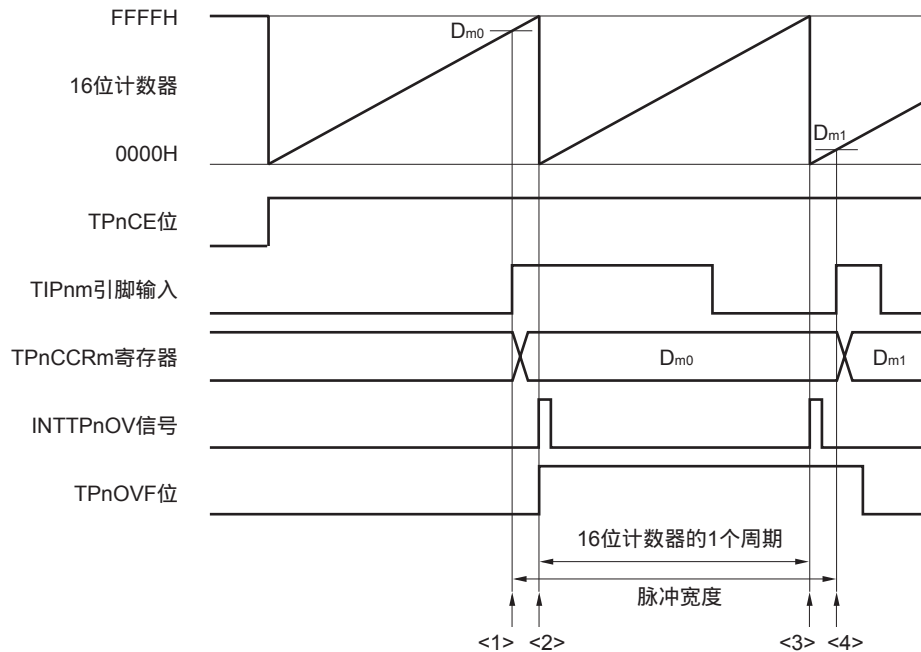
注 这里的 TPnOVF0 和 TPnOVF1 标志是建立于内置 RAM 上的变量，可由软件任意进行设置。

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

(d) 捕捉触发间隔较长时的溢出处理

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

捕捉触发间隔较长时的错误处理举例



使用自由运行定时器模式对较长的脉冲宽度进行测量时，可能出现以下的问题。

- <1> 读取 TPnCCRM 寄存器（设置为 TIPnm 引脚输入的初始值）。
- <2> 产生了溢出。但不进行任何软件处理。
- <3> 产生第二次溢出。但不进行任何软件处理。
- <4> 读取 TPnCCRM 寄存器。

读取溢出标志位。如果溢出标志为 1，那么将其清零。

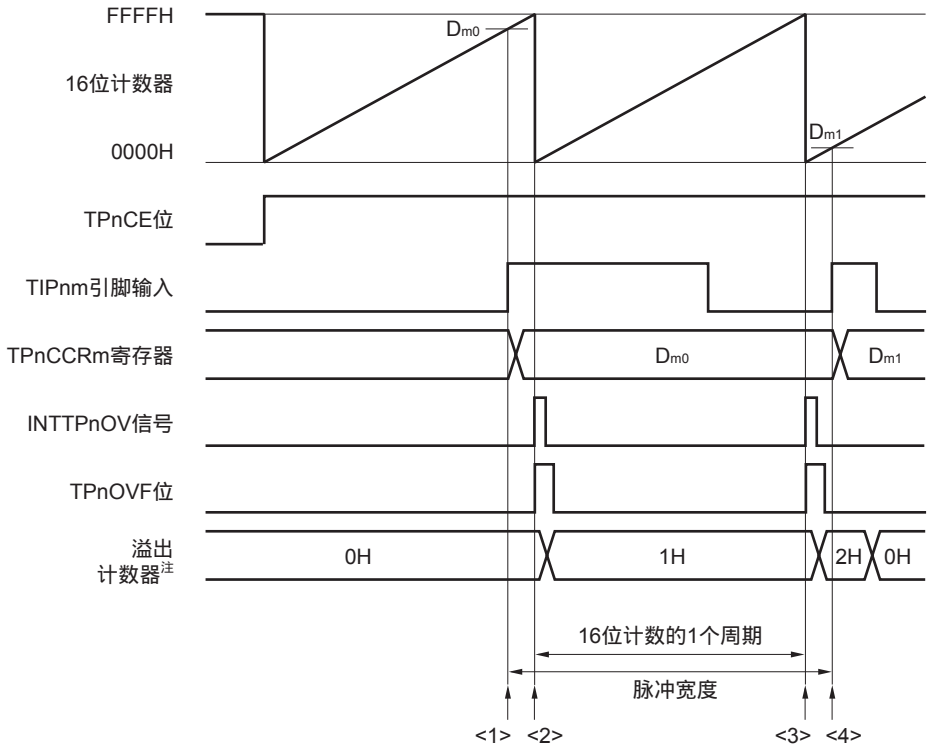
由于溢出标志为 1，脉冲宽度按照 $(10000H + D_{m1} - D_{m0})$ （错误）来计算。

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

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

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

捕捉触发间隔较长时的处理举例



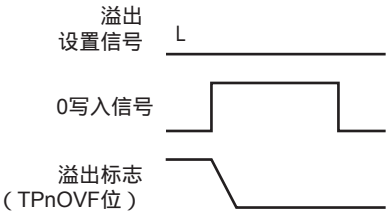
注 这里的溢出次数计数器是建立于内置 RAM 上的变量，可由软件任意进行设置。

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

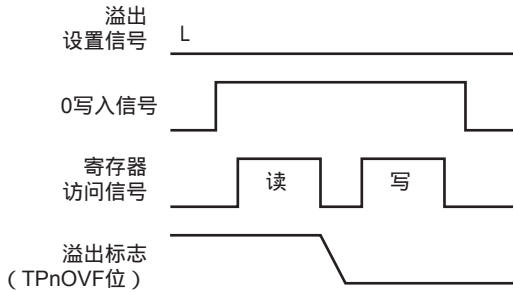
(e) 将溢出标志位清零

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

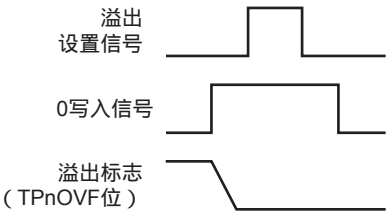
(i) 写0操作（与设置不冲突）



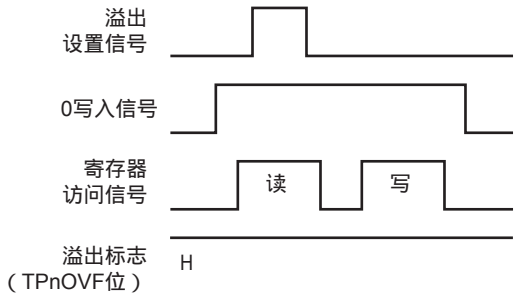
(iii) 清0操作（与设置不冲突）



(ii) 写0操作（与设置冲突）



(iv) 清0操作（与设置冲突）



备注 n = 0 至 3

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

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

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

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

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

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

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

图 6-34. 脉宽测量模式的配置图

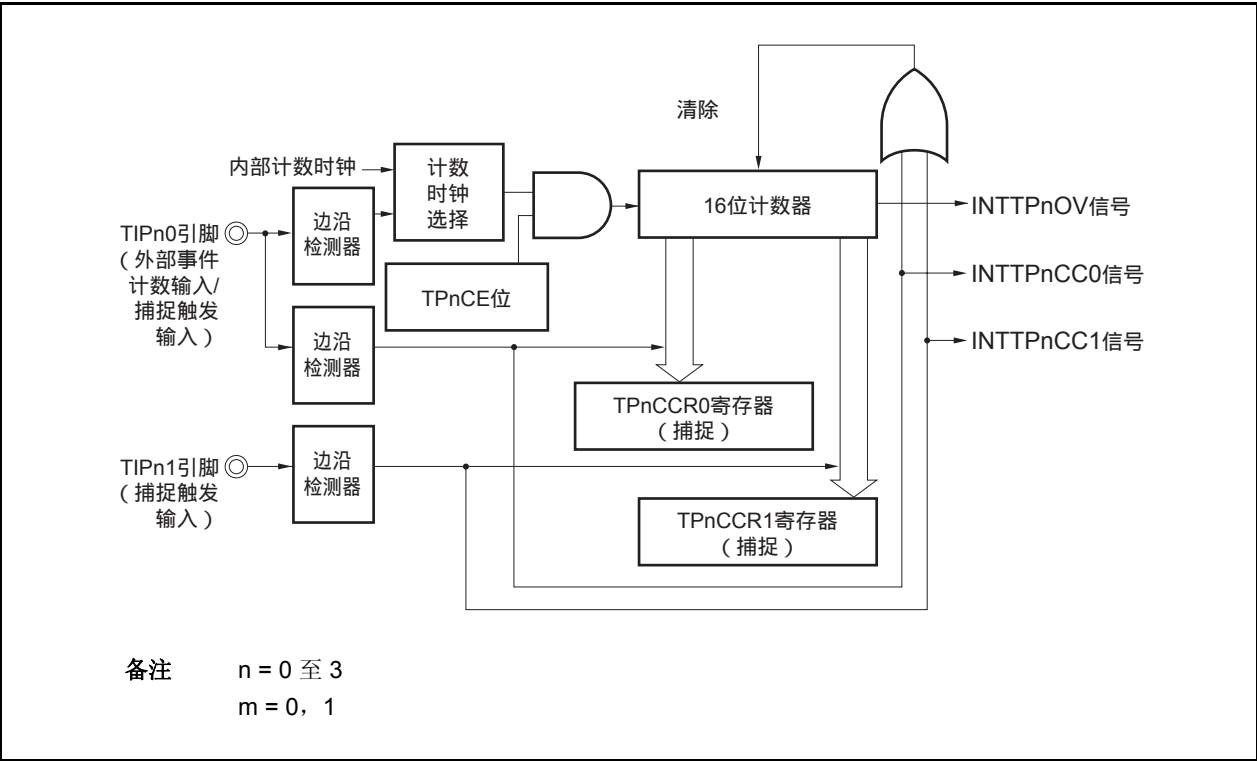
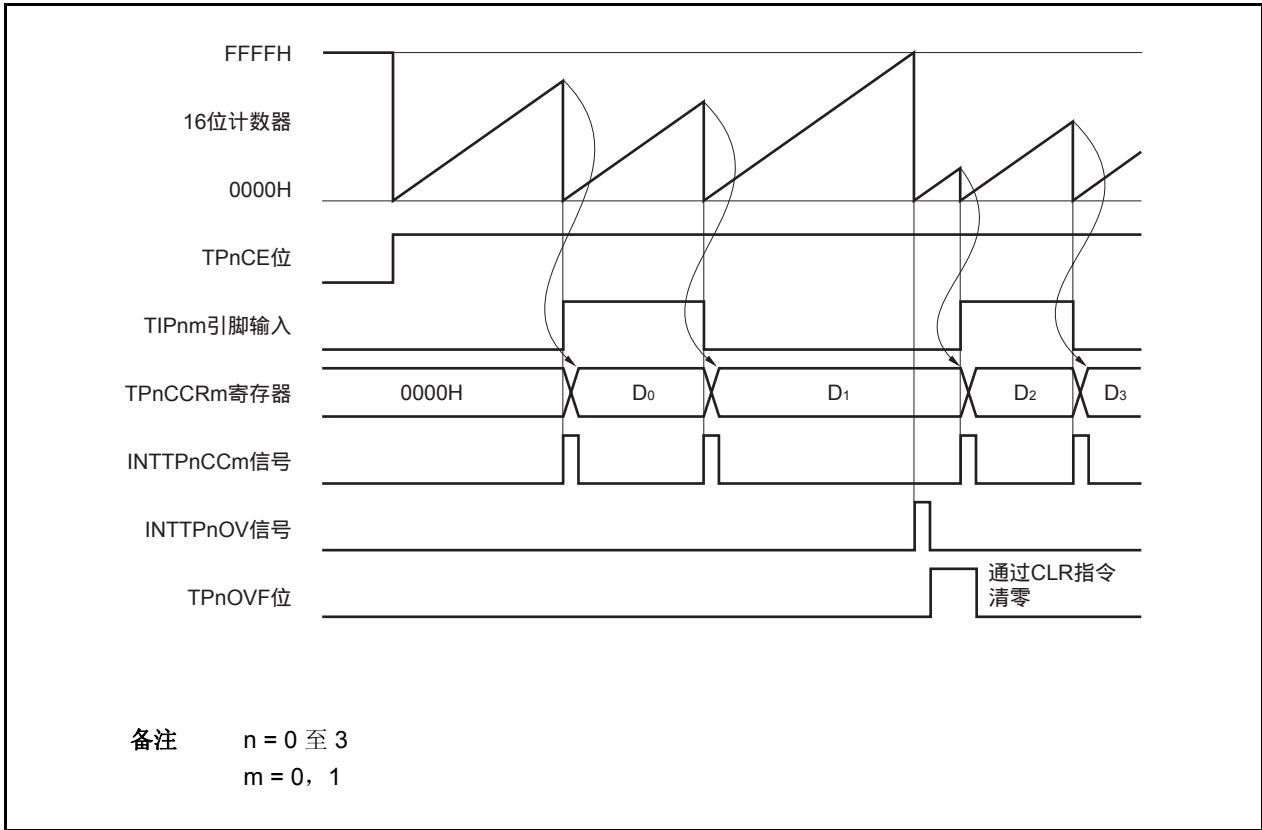


图 6-35. 脉宽测量模式的基本时序



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

脉冲宽度可由下式求得。

脉冲宽度 = 捕捉计数值 × 计数时钟周期

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

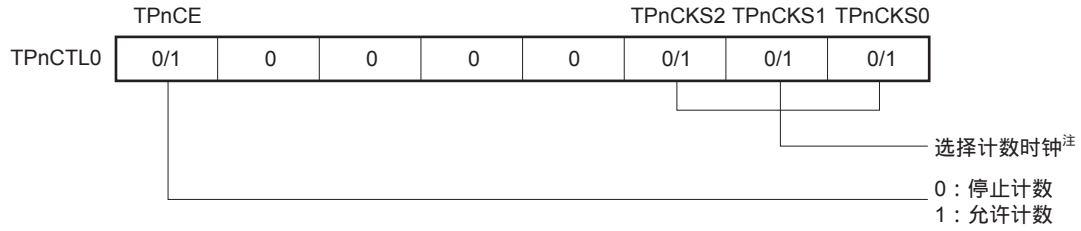
如果溢出标志被设置为 1，那么脉冲宽度可由下式求得。

脉冲宽度 = (10000H × TPnOVF 位被设置为 1 的次数 + 捕捉计数值) × 计数时钟周期

备注 n = 0 至 3
m = 0, 1

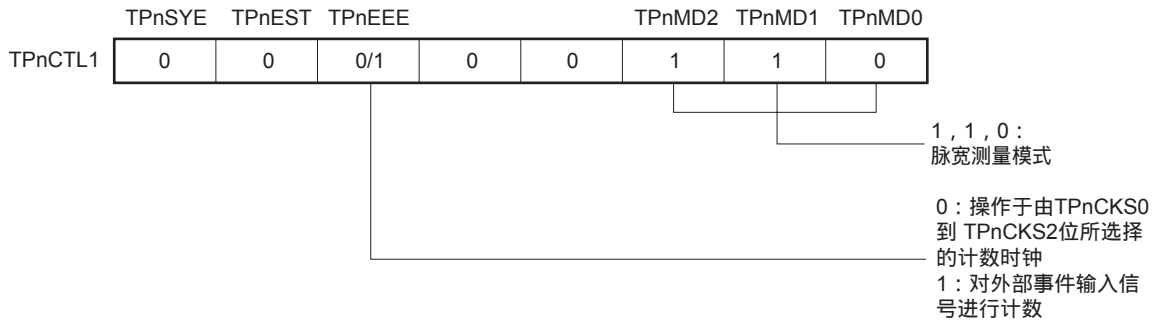
图 6-36. 脉宽测量模式的寄存器设置 (1/2)

(a) TMPn 控制寄存器 0 (TPnCTL0)

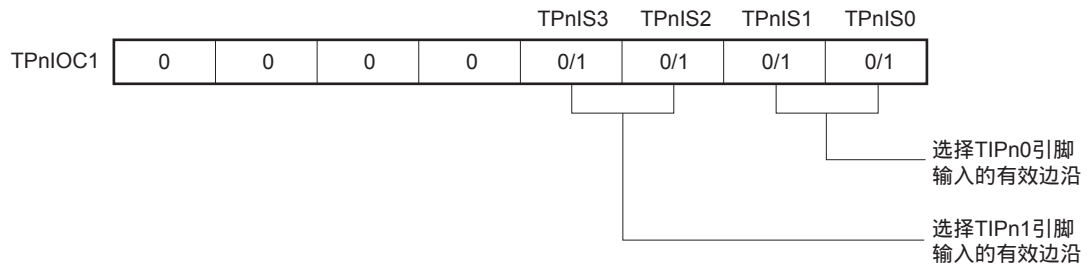


注 TPnEEE 位 = 1 时该设置无效。

(b) TMPn 控制寄存器 1 (TPnCTL1)



(c) TMPn I/O 控制寄存器 1 (TPnIOC1)



(d) TMPn I/O 控制寄存器 2 (TPnIOC2)

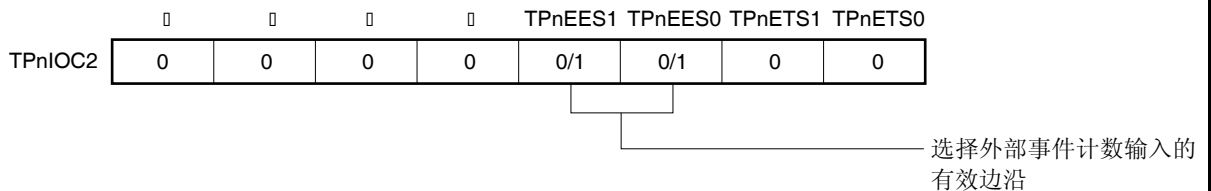
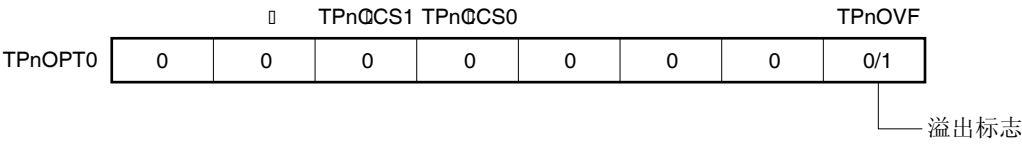


图 6-36. 脉宽测量模式的寄存器设置 (2/2)

(e) TMPn 选项寄存器 0 (TPnOPT0)



(f) TMPn 计数器读取缓冲寄存器 (TPnCNT)

可以通过读取 TPnCNT 寄存器来读取 16 位计数器的计数值。

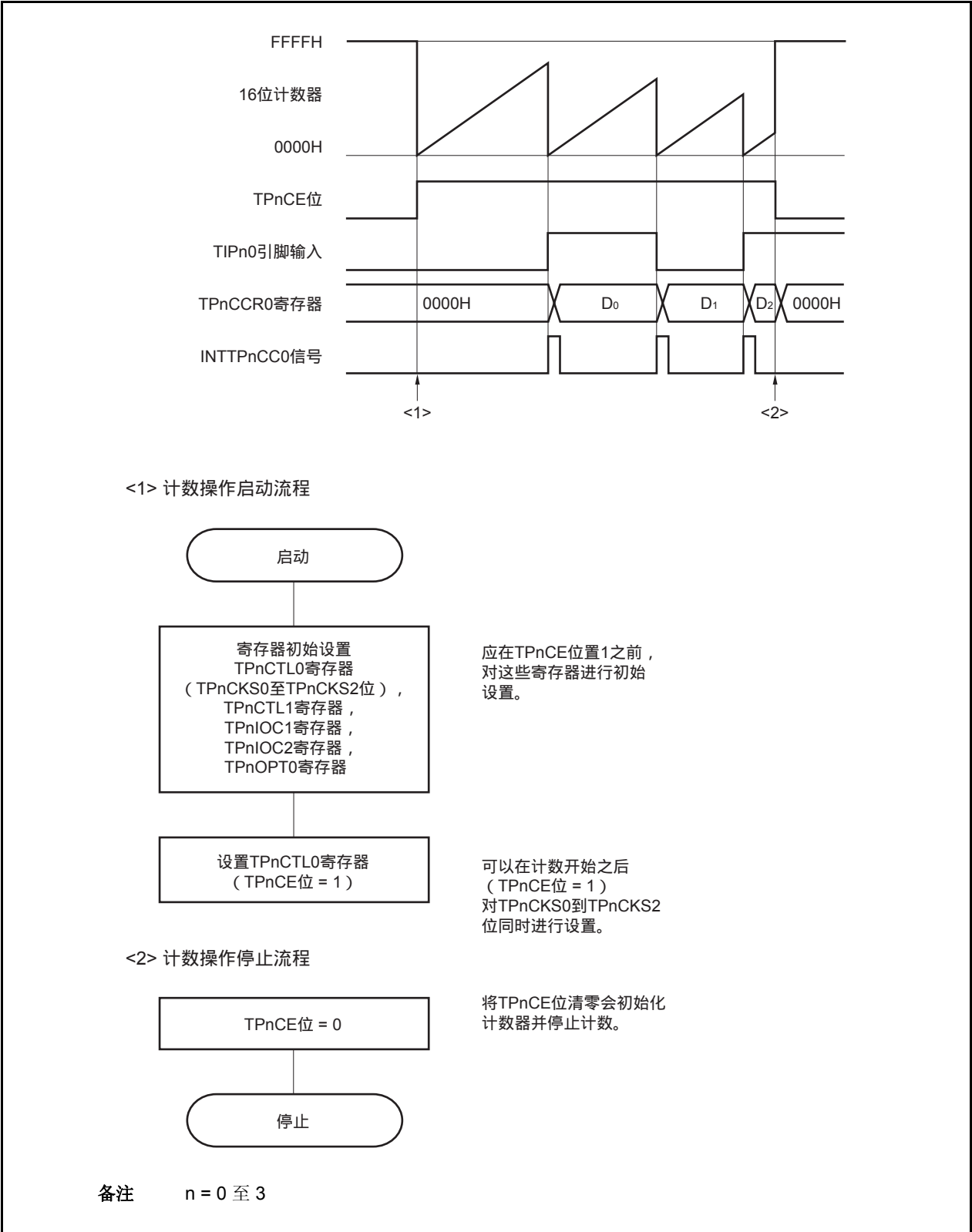
(g) TMPn 捕捉/比较寄存器 0 和 1 (TPnCCR0 和 TPnCCR1)

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

- 备注
- 1. 脉宽测量模式下，并未使用 TMPn I/O 控制寄存器 0 (TPnIOC0)。
 - 2. n = 0 至 3
m = 0, 1

(1) 脉宽测量模式的操作流程

图 6-37. 脉宽测量模式的软件处理流程

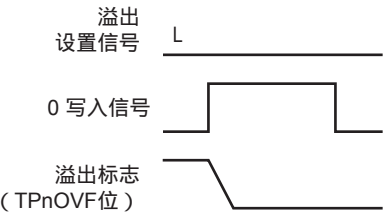


(2) 脉宽测量模式的操作时序

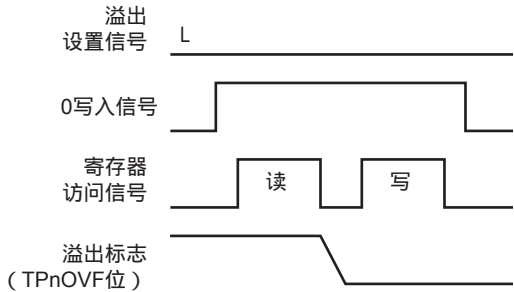
(a) 将溢出标志位清零

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

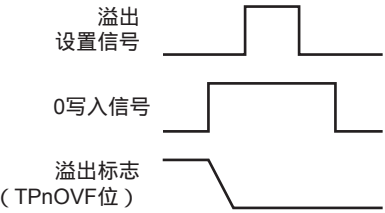
(i) 写0操作（与设置不冲突）



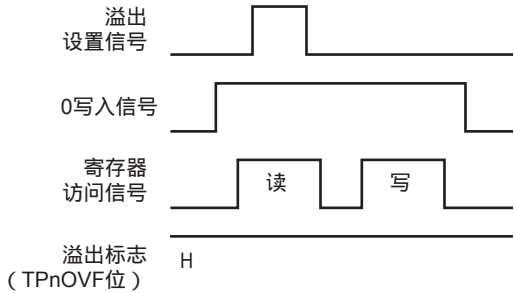
(iii) 清0操作（与设置不冲突）



(ii) 写0操作（与设置冲突）



(iv) 清0操作（与设置冲突）



备注 n = 0 至 3

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

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

6.5.8 定时器输出操作

TOPn0 和 TOPn1 引脚的工作模式与输出电平的关系如下表所示。

表 6-4. 各模式下的定时器输出控制

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

备注 n = 0 至 3

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

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

备注 n = 0 至 3
 m = 0, 1

6.6 定时器调谐操作功能

定时器 P 以及定时器 Q 具有定时器调谐操作功能。

表 6-6 中列出了可以同步的定时器。

表 6-6. 定时器的调谐操作模式

主定时器	从定时器	
TMP0	TMP1	—
TMP2	TMP3	TMQ0

注意事项 1. 通过 TPmCTL1.TPmSYE 及 TQ0CTL1.TQ0SYE 位来允许或禁止调谐操作模式。对于主定时器 TMP2 来说，既可以将 TMP3 或 TMQ0 单独指定为从定时器，也可以将它们同时指定为从定时器。

2. 通过以下步骤来设置调谐操作模式。

<1> 设置从定时器的 TPmCTL1.TPmSYE 及 TQ0CTL1.TQ0SYE 位以允许调谐操作。

将从定时器的 TPmCTL1.TPmMD2 至 TPmCTL1.TPmMD0 位以及 TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位设置为自由运行模式。

<2> 利用 TPnCTL1.TPnMD2 至 TPnCTL1.TPnMD0 位来设置定时器模式。

这时，不要对主定时器的 TPnCTL1.TPnSYE 进行设置。

<3> 设置主定时器及从定时器的比较寄存器值。

<4> 设置从定时器的 TPmCTL0.TPmCE 及 TQ0CTL0.TQ0CE 位以允许操作内置操作时钟。

<5> 设置主定时器的 TPnCTL0.TPnCE 位以允许操作内置操作时钟。

备注 m = 1, 3
n = 0, 2

表 6-7 和 6-8 显示了可以在调谐操作模式中使用的定时器模式（√：可以设置，×：不能设置）。

表 6-7. 可在调谐操作模式中使用的定时器模式

主定时器	自由运行模式	PWM 模式	三角形波 PWM 模式
TMP0	√	√	×
TMP2	√	√	×

表 6-8. 定时器输出功能

调谐通道	定时器	引脚	自由运行模式		PWM 模式		三角形波 PWM 模式	
			调谐关	调谐开	调谐关	调谐开	调谐关	调谐开
Ch0	TMP0 (主)	TOP00	PPG		Toggle		N/A	
		TOP01	PPG		PWM		N/A	
	TMP1 (副)	TOP10	PGP		Toggle	PWM	N/A	
		TOP11	PPG		PWM		N/A	
Ch1	TMP2 (主)	TOP20	PPG		Toggle		N/A	
		TOP21	PPG		PWM		N/A	
	TMP3 (副)	TOP30	PPG		Toggle	PWM	N/A	
		TOP31	PPG		PWM		N/A	
	TMQ0 (副)	TOQ00	PPG		Toggle	PWM	Toggle	N/A
		TOQ01 至 TOQ03	PPG		PWM		三角形波 PWM	N/A

备注 将数据从主定时器的比较寄存器发送至从定时器的比较寄存器的时序如下所示。

PPG: CPU 写入时序。

Toggle, PWM, 三角形波 PWM: 定时器计数器和比较寄存器与 TOPn0 及 TOQ00 匹配时的时序 (n = 0 至 3)。

图 6-38. 调谐操作图 (TMP2, TMP3, TMQ0)

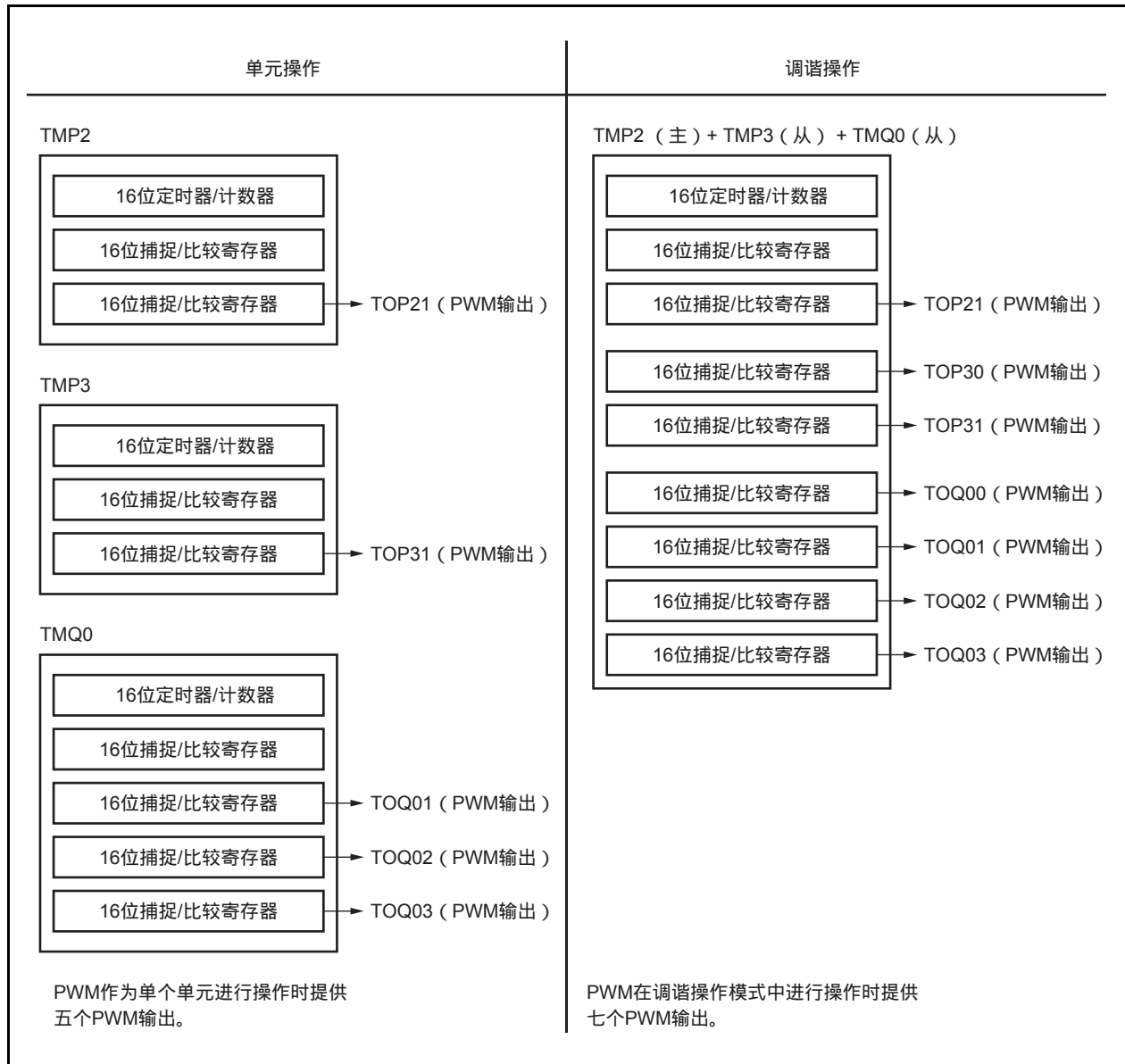
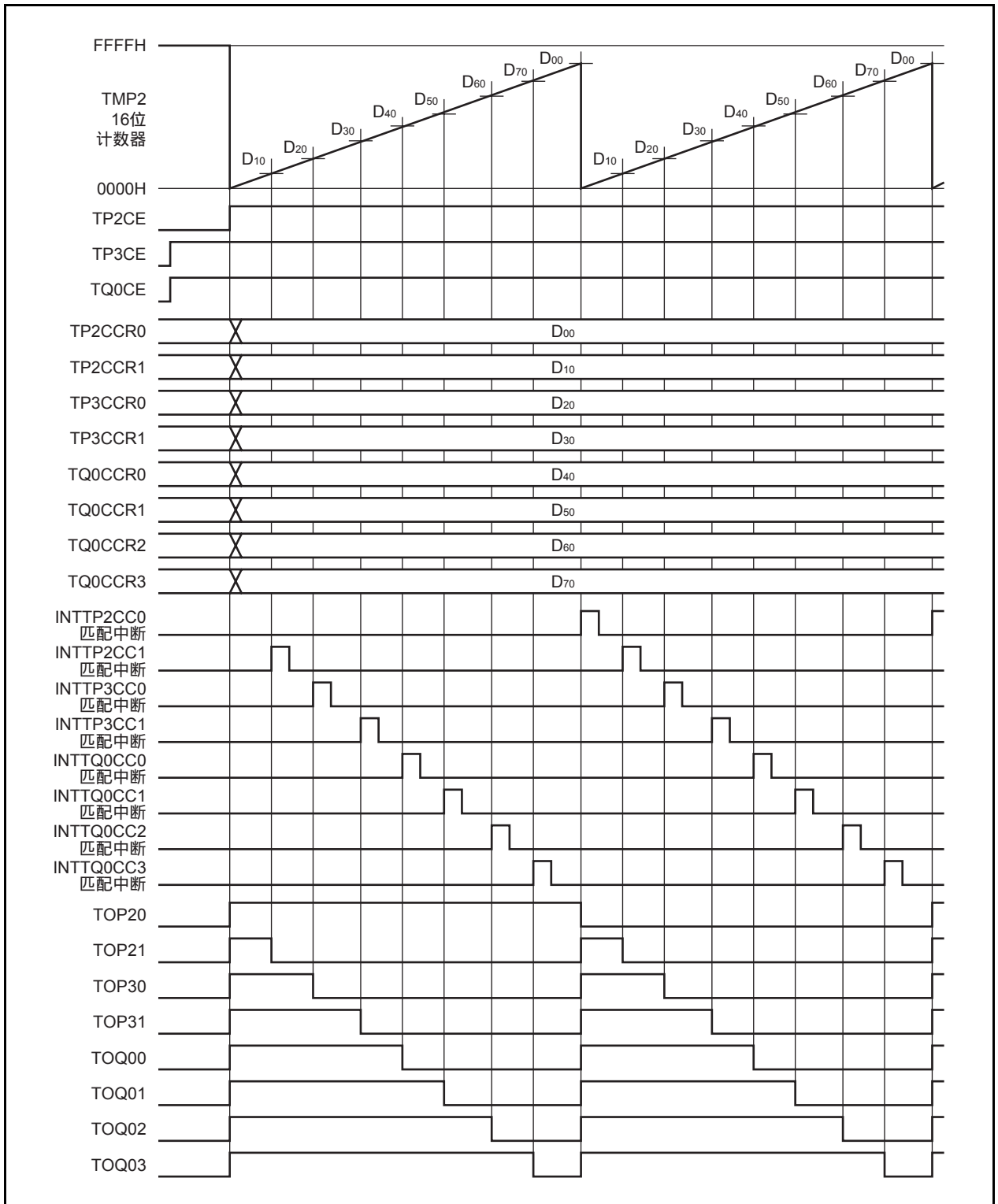


图 6-39. PWM 调谐功能的基本操作时序 (TMP2, TMP3, TMQ0)



6.7 选择器功能

在 V850ES/HE2 中，端口的复用功能引脚和周边 I/O (TMP, TMM0 或 UARTA) 可以用来选择 TMP 的捕捉触发器输入。

通过使用这个功能，下列操作成为可能。

- 可以从端口/定时器复用功能引脚 (TIP10 及 TIP11 引脚) 以及 UARTA 接收复用功能引脚 (RXDA0 及 RXDA1 引脚) 中选择 TMP1 的 TIP10 和 TIP11 输入信号。
→ 当选中 UARTA0 的 RXDA0 信号或 UARTA1 的 RXDA1 时，UARTA LIN 接收传输的波特率误差可以被计算。
- TMP0 的 TIP01 输入信号可以从端口/定时器复用功能引脚 (TIP01 引脚) 以及 TMM0 的 INTTMM0EQ0 信号中选择。

- 注意事项**
1. 当使用选择器功能时，在连接定时器之前设置 TMP 的捕捉触发器输入。
 2. 当设置选择器功能时，首先禁止所连接的外设 I/O (TMP, TMM0 或 UARTA)。

下列寄存器用来指定为选择器功能的捕捉输入。

(1) 选择器操作控制寄存器 0 (SELCNT0)

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

可以对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后：00H R/W 地址：FFFFFF308H

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

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

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

ISEL02 ^注	TIP01输入信号（TMP0）的选择
0	TIP01引脚输入
1	TMM0的INTTM0EQ0中断

注 在以下条件中将 INTTM0EQ0 中断信号用作 TIP01 输入信号。
TMM0 操作时钟 ≥ TMP0 操作时钟 × 4

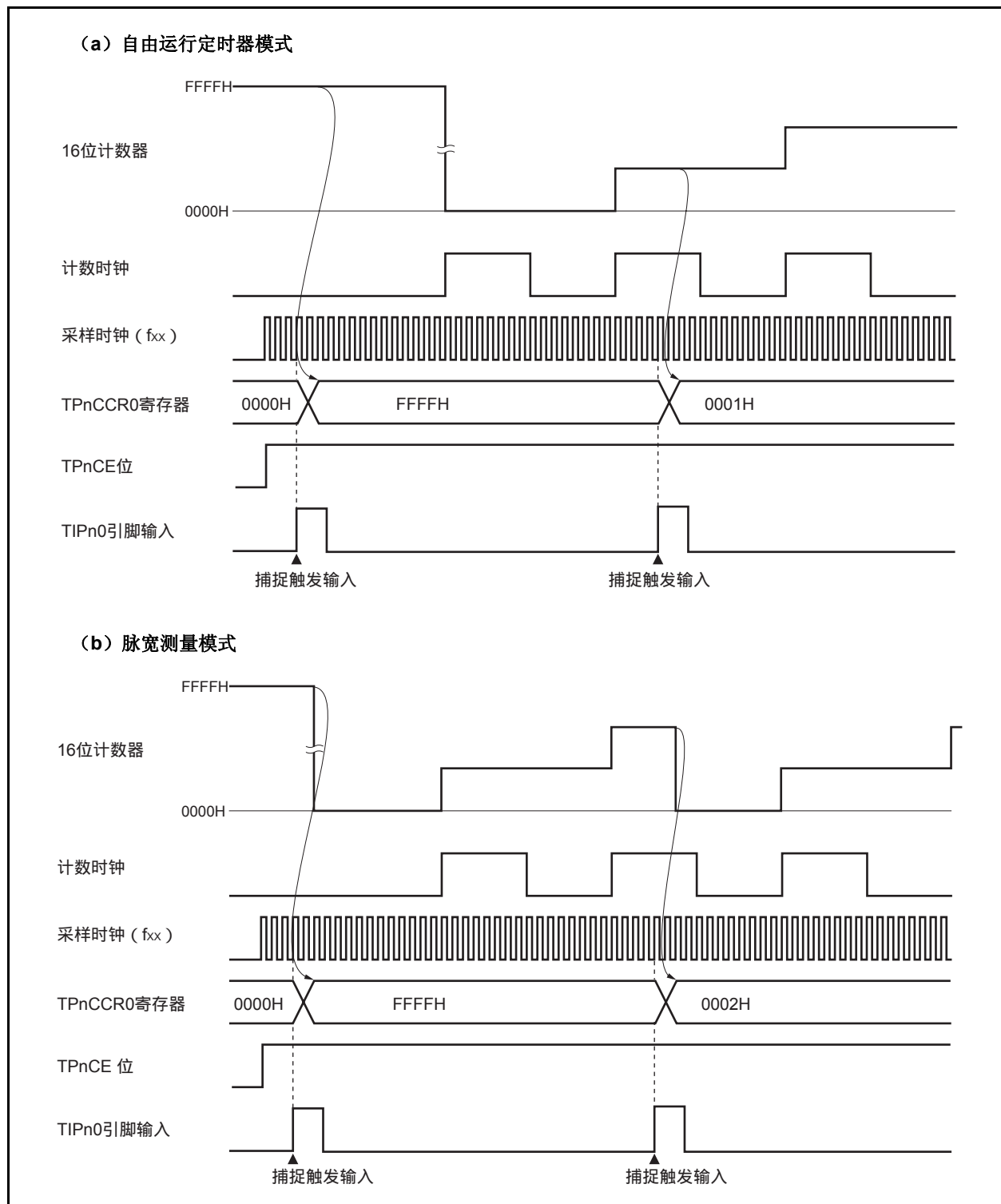
- 注意事项
1. 要将 ISEL02 至 ISEL04 位设为 1，应在捕捉输入模式中对相应引脚进行设置。
 2. 目标器件 (TMP0, TMP1, TMM0, UARTA0 或 UARTA1) 的操作停止时对 ISEL02 至 ISEL04 位进行设置。

<R>

6.8 注意事项

(1) 捕捉操作

当使用捕捉操作并且将低速时钟选择为计数时钟时，如果 TPnCE 位置 1 之后捕捉触发器被立即输入，那么 FFFFH，而不是 0000H，可以被捕捉到 TPnCCR0 和 TPnCCR1 寄存器中。



第 7 章 16 位定时器/事件计数器 Q (TMQ)

定时器 Q (TMQ) 是一个 16 位的定时器/事件计数器。

V850ES/HE2 具有 TMQ0。

7.1 概述

以下是 TMQ0 的概况。

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

7.2 功能

TMQ0 具有以下功能。

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

7.3 配置

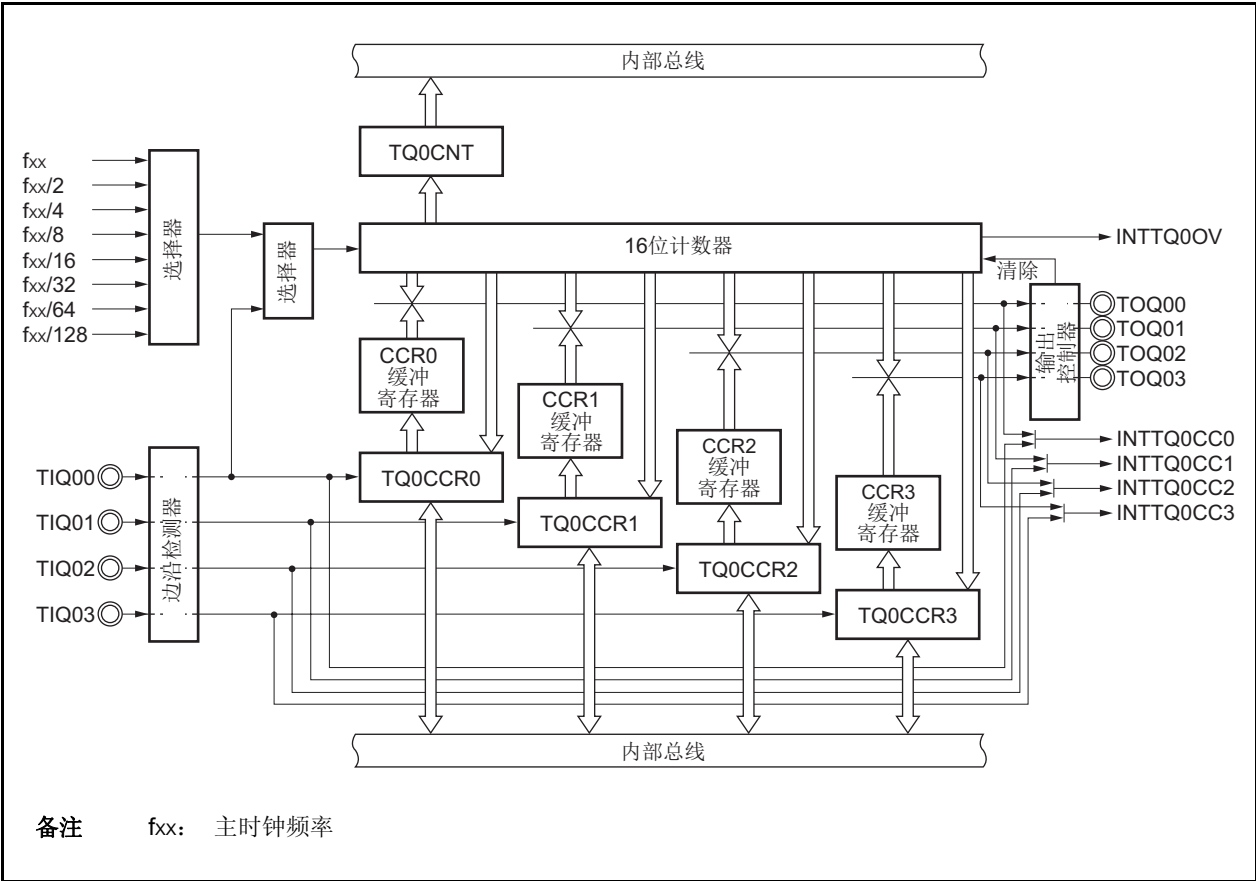
TMQ0 包含以下硬件。

表 7-1. TMQ0 的配置

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

- 注
1. TIQ00 引脚的复用功能包括捕捉触发输入信号，外部事件计数输入信号和外部触发输入信号。
 2. 使用 TIQ00 至 TIQ03 和 TOQ00 至 TOQ03 引脚功能时，请参照表 4-12 端口引脚作为复用引脚使用。

图 7-1. TMQ0 的框图



(1) 16 位计数器

该 16 位计数器可以对内部时钟或外部事件进行计数。

可以使用 TQ0CNT 寄存器读取该计数器的计数值。

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

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

(2) CCR0 缓冲寄存器

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

当 TQ0CCR0 寄存器被作为比较寄存器使用时, 向 TQ0CCR0 寄存器写入的数据将被传送至 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 (INTTQ0CC0)。

CCR0 缓冲寄存器不能被读写。

复位后 CCR0 缓冲寄存器和 TQ0CCR0 寄存器都将被清零。

(3) CCR1 缓冲寄存器

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

当 TQ0CCR1 寄存器被作为比较寄存器使用时, 向 TQ0CCR1 寄存器写入的数据将被传送至 CCR1 缓冲寄存器。

如果 16 位计数器的计数值与 CCR1 缓冲寄存器的值匹配, 会产生一个比较匹配中断请求信号 (INTTQ0CC1)。

CCR1 缓冲寄存器不能被读写。

复位后 CCR1 缓冲寄存器和 TQ0CCR1 寄存器都将被清为 0000H。

(4) CCR2 缓冲寄存器

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

当 TQ0CCR2 寄存器被作为比较寄存器使用时, 向 TQ0CCR2 寄存器写入的数据将被传送至 CCR2 缓冲寄存器。

如果 16 位计数器的计数值与 CCR2 缓冲寄存器的值匹配, 会产生一个比较匹配中断请求信号 (INTTQ0CC2)。

CCR1 缓冲寄存器不能被读写。

复位后 CCR2 缓冲寄存器和 TQ0CCR2 寄存器都将被清为 0000H。

(5) CCR3 缓冲寄存器

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

当 TQ0CCR3 寄存器被作为比较寄存器使用时, 向 TQ0CCR3 寄存器写入的数据将被传送至 CCR3 缓冲寄存器。

如果 16 位计数器的计数值与 CCR3 缓冲寄存器的值匹配, 会产生一个比较匹配中断请求信号 (INTTQ0CC3)。

CCR3 缓冲寄存器不能被读写。

复位后 CCR3 缓冲寄存器和 TQ0CCR3 寄存器都将被清为 0000H。

(6) 边沿检测器

该电路用于检测 TIQ00 和 TIQ03 引脚输入的有效边沿。通过 TQ0IOC1 和 TQ0IOC2 寄存器选择无有效沿, 上升沿, 下降沿或双沿有效。

(7) 输出控制器

该电路用于控制 TOQ00 至 TOQ03 引脚的输出。输出控制器是由 TQ0IOC0 寄存器控制的。

(8) 选择器

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

7.4 寄存器

用于控制 TMQ0 的寄存器如下所述。

- TMQ0 控制寄存器 0 (TQ0CTL0)
- TMQ0 控制寄存器 1 (TQ0CTL1)
- TMQ0 I/O 控制寄存器 0 (TQ0IOC0)
- TMQ0 I/O 控制寄存器 1 (TQ0IOC1)
- TMQ0 I/O 控制寄存器 2 (TQ0IOC2)
- TMQ0 选项寄存器 0 (TQ0OPT0)
- TMQ0 捕捉/比较寄存器 0 (TQ0CCR0)
- TMQ0 捕捉/比较寄存器 1 (TQ0CCR1)
- TMQ0 捕捉/比较寄存器 2 (TQ0CCR2)
- TMQ0 捕捉/比较寄存器 3 (TQ0CCR3)
- TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

备注 使用 TIQ00 至 TIQ03 和 TOQ00 至 TOQ03 引脚功能时，请参照表 4-12 端口引脚作为复用引脚使用。

(1) TMQ0 控制寄存器 0 (TQ0CTL0)

TQ0CTL0 寄存器为用于控制 TMQ0 操作的 8 位寄存器。

可以对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

可使用软件向 TQ0CTL0 寄存器写入相同的数值。

复位后：00H R/W 地址： FFFFF540H

	7	6	5	4	3	2	1	0
TQ0CTL0	TQ0CE	0	0	0	0	TQ0CKS2	TQ0CKS1	TQ0CKS0

TQ0CE	TMQ0操作控制
0	禁止操作TMQ0（TMQ0异步复位 ^注 ）。
1	允许操作TMQ0。开始操作TMQ0。

TQ0CKS2	TQ0CKS1	TQ0CKS0	内部计数时钟选择
0	0	0	f _{xx}
0	0	1	f _{xx} /2
0	1	0	f _{xx} /4
0	1	1	f _{xx} /8
1	0	0	f _{xx} /16
1	0	1	f _{xx} /32
1	1	0	f _{xx} /64
1	1	1	f _{xx} /128

注 TQ0OPT0.TQ0OVF 位，16 位计数器，定时器输出（TOQ00 至 TOQ03 引脚）。

- 注意事项
- 在 TQ0CE 位 = 0 时，设置 TQ0CKS2 至 TQ0CKS0 位。
当将 TQ0CE 位的值从 0 改变为 1 时，可同时对 TQ0CKS2 至 TQ0CKS0 位进行设置。
 - 请务必将第 3 到第 6 位清除为“0”。

备注 fxx: 主时钟频率

(2) TMQ0 控制寄存器 1 (TQ0CTL1)

TQ0CTL1 寄存器是用于控制 TMQ0 操作的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

(1/2)

复位后： 00H R/W 地址： FFFFF541H

	7	6	5	4	3	2	1	0
TQ0CTL1	TQ0SYE	TQ0EST	TQ0EEE	0	0	TQ0MD2	TQ0MD1	TQ0MD0

TQ0SYE	调谐操作模式允许控制							
0	独立操作模式（异步操作模式）							
1	调谐操作模式（从操作的指定） 在该模式中，定时器P可以与主定时器进行同步操作。							
	<table><tr><td>主定时器</td><td colspan="2">从定时器</td></tr><tr><td>TMP2</td><td>TMP3</td><td>TMQ0</td></tr></table>		主定时器	从定时器		TMP2	TMP3	TMQ0
主定时器	从定时器							
TMP2	TMP3	TMQ0						
	关于调谐操作模式的详细信息，参见7.6 定时器调谐操作功能。							

TQ0EST	软件触发控制	
0	-	
1	为外部触发输入产生有效信号。 · 单脉冲输出模式：通过向 TQ0EST 位写1作为触发源来输出单脉冲。 · 外部触发脉冲输出模式：通过向TQ0EST 位写1作为触发源来输出 PWM 波形。	

- 注意事项
1. 对 TQ0EST 位的设置只有在外部触发脉冲输出模式或单脉冲输出模式才有效。在其它模式下，对该位的置 1 操作将被忽略。
 2. 请务必将第 3 位和第 4 位清除为“0”。

TQ0EEE	计数时钟选择
0	禁止通过外部事件计数输入操作。 (通过 TQ0CTL0.TQ0CK0至 TQ0CK2 位选择的计数时钟执行计数。)
1	允许通过外部事件计数输入操作。 (在外部事件计数输入信号的有效沿执行计数。)
TQ0EEE 位选择通过内部计数时钟或外部事件计数输入的有效沿执行计数。	

TQ0MD2	TQ0MD1	TQ0MD0	定时器模式选择
0	0	0	间隔定时器模式
0	0	1	外部事件计数模式
0	1	0	外部触发脉冲输出模式
0	1	1	单脉冲输出模式
1	0	0	PWM输出模式
1	0	1	自由运行定时器模式
1	1	0	脉宽测量模式
1	1	1	三角形波PWM模式

- 注意事项
1. 在外部事件计数模式下，无论 **TQ0EEE** 位如何设置，外部事件计数输入都是被选中的。
 2. 只有在 **TQ0CTL0.TQ0CE** 位 = 0 时才可以对 **TQ0EEE** 以及 **TQ0MD2** 至 **TQ0MD0** 位进行设置 (**TQ0CE** 位 = 1 时，可以对这些位写入相同的值)。如果在 **TQ0CE** 位 = 1 时对这些位进行了改写，那么操作结果将不能被保证。若由于误操作引起了改写，则要将 **TQ0CE** 位清零然后再次设置这些寄存器位。

<R> (3) TMQ0 I/O 控制寄存器 0 (TQ0IOC0)

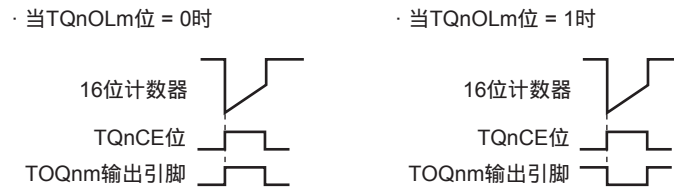
TQ0IOC0 寄存器是用于控制定时器输出的 8 位寄存器 (TOQ00 至 TOQ03 引脚)。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后: 00H	R/W	地址: FFFF542H								
			7	6	5	4	3	2	1	0
TQ0IOC0			TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
			TQ0OLm	TOQ0m引脚输出电平设置 (m = 0至3) 注						
			0	TOQ0m引脚高电平开始						
			1	TOQ0m引脚低电平开始						
			TQ0OEm	TOQ0m引脚输出设置 (m = 0至3)						
			0	禁止定时器输出 · 当TQ0OLm位 = 0时: 从TOQ0m引脚中输出低电平 · 当TQ0OLm位 = 1时: 从TOQ0m引脚中输出高电平						
			1	允许定时器输出 (从TOQ0m引脚输出方波)						

注 由 TQ0OLm 位指定的定时器输出引脚 (TOQ0m) 的输出电平如下列所示。



- 注意事项**
1. 只有在 TQ0CTL0.TQ0CE 位 = 0 时才可以对 TQ0OLm 和 TQ0OEm 位进行改写 (当 TQ0CE 位 = 1 时可以向这些位写入相同的值)。若由于误操作引起了改写, 则要将 TQ0CE 位清零然后再次设置这些寄存器位。
 2. 当 TQ0CE 和 TQ0OEm 位为 0 时, 即使对 TQ0OLm 位进行了操作, TOQ0m 引脚的输出电平也不能确定。

备注 m = 0 至 3

(4) TMQ0 I/O 控制寄存器 1 (TQ0IOC1)

TQ0IOC1 寄存器是用于控制捕捉触发输入信号 (TIQ00 至 TIQ03 引脚) 有效沿的 8 位寄存器。
可对该寄存器进行 8 位或 1 位的读写操作。
复位输入将把该寄存器设置为 00H。

复位后: 00H R/W 地址: FFFFF543H

	7	6	5	4	3	2	1	0
TQ0IOC1	TQ0IS7	TQ0IS6	TQ0IS5	TQ0IS4	TQ0IS3	TQ0IS2	TQ0IS1	TQ0IS0

TQ0IS7	TQ0IS6	捕捉触发输入信号 (TIQ03 引脚) 有效边沿设定
0	0	无边沿检测 (捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升沿及下降沿

TQ0IS5	TQ0IS4	捕捉触发输出信号 (TIQ02 引脚) 有效边沿检测
0	0	无边沿检测 (捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升沿及下降沿

TQ0IS3	TQ0IS2	捕捉触发输出信号 (TIQ01 引脚) 有效边沿设定
0	0	无边沿检测 (捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升沿及下降沿

TQ0IS1	TQ0IS0	捕捉触发输出信号 (TIQ00 引脚) 有效边沿设定
0	0	无边沿检测 (捕捉操作无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升沿及下降沿

- 注意事项
1. 只有在 TQ0CTL0.TQ0CE 位 = 0 时才可以对 TQ0IS7 至 TQ0IS0 进行改写 (TQ0CE 位 = 1 时只可以向这些位写入相同的值)。若由于误操作引起了改写, 则要将 TQ0CE 位清零然后再次设置这些寄存器位。
 2. TQ0IS7 至 TQ0IS0 位只有在自由运行定时器模式和脉宽测量模式下才有效。其它模式下, 无法进行捕捉操作。

(5) TMQ0 I/O 控制寄存器 2 (TQ0IOC2)

TQ0IOC2 寄存器是用于控制外部事件计数输入信号 (TIQ00 引脚) 有效沿和外部触发输入信号 (TIQ00 引脚) 有效沿的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后:	00H	R/W	地址:	FFFF544H				
	7	6	5	4	3	2	1	0
TQ0IOC2	0	0	0	0	TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0

TQ0EES1	TQ0EES0	外部事件计数输入信号 (TIQ00引脚) 有效边沿设定
0	0	无边沿检测 (外部事件计数无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升沿及下降沿

TQ0ETS1	TQ0ETS0	外部触发输入信号 (TIQ00引脚) 有效边沿设定
0	0	无边沿检测 (外部触发无效)
0	1	检测上升沿
1	0	检测下降沿
1	1	检测上升沿及下降沿

- 注意事项**
1. 在 TQ0CTL0.TQ0CE 位 = 0 时才可对 TQ0EES1, TQ0EES0, TQ0ETS1 和 TQ0ETS0 位进行改写 (TQ0CE 位 = 1 时只可以向这些位写入相同的值)。若在 TQ0CE 位 = 1 由于误操作引起了改写, 则要将 TQ0CE 位清零然后再次设置这些寄存器位。
 2. TQ0EES1 和 TQ0EES0 位只有在 TQ0CTL1.TQ0EEE 位 = 1 或设置了外部事件计数模式 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 001) 时才有效。
 3. TQ0ETS1 和 TQ0ETS0 位只有在外部触发脉冲输出模式 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 010) 或单脉冲输出模式 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 = 011) 下才有效。

(6) TMQ0 选项寄存器 0（TQ0OPT0）

TQ0OPT0 寄存器是用于设置捕捉/比较操作和检测溢出的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后： 00H R/W 地址： FFFFF545H

	7	6	5	4	3	2	1	0
TQ0OPT0	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0	0	0	0	TQ0OVF

TQ0CCSm	TQ0CCRm 寄存器捕捉/比较选择
0	选择为比较寄存器
1	选择为捕捉寄存器
只有在自由运行定时器模式下 TQ0CCSm 位的设置才有效。	

TQ0OVF	TMQ0溢出检测
设置（1）	发生溢出
复位（0）	TQ0OVF 位写 0 或 TQ0CTL0.TQ0CE 位 = 07
<ul style="list-style-type: none">• 在自由运行定时器模式下或脉冲宽度测量模式下16位计数器计数值从FFFFH 到 0000H 溢出时TQ0OVF位复位。• TQ0OVF位被置1的同时会产生一个中断请求信号(INTTQ0OV)。 INTTQ0OV信号不会在自由运行定时器模式及脉冲宽度测量模式以外的模式中生成。• TQ0OVF位=1时，即使对TQ0OVF位或TQ0OPT0寄存器进行读取操作，TQ0OVF位也不会被清零。• 可以对TQ0OVF位进行读取和写入操作，但是不能通过软件将TQ0OVF位置1。向其写入1不会对TMQ0的操作造成任何影响。	

- 注意事项
1. 当 **TQ0CTL0.TQ0CE** 位 = 0 时可对 **TQ0CCS3** 到 **TQ0CCS0** 位进行改写（当 **TQ0CE** 位 = 1 时也可以向这些位写入相同的值）。若进行了错误的改写，须先将 **TQ0CE** 位清零，然后再次设置这些位。
 2. 请务必将第 1 到 3 位清除为“0”。

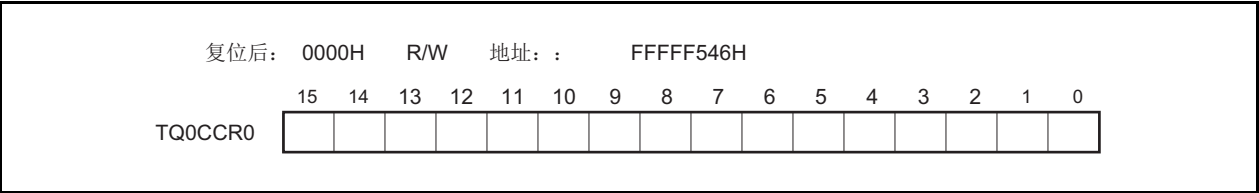
备注 m = 0 至 3

(7) TMQ0 捕捉/比较寄存器 0 (TQ0CCR0)

根据选择模式的不同，TQ0CCR0 寄存器可用作捕捉寄存器或比较寄存器。
该寄存器只有在自由运行模式下根据 TQ0OPT0.TQ0CCS0 位设置的不同，可以用作捕捉寄存器或比较寄存器。
在脉宽测量模式下，TQ0CCR0 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。
TQ0CCR0 寄存器可以在操作过程中进行读写。
可对该寄存器进行 16 位的读写操作。
复位将把该寄存器设置为 0000H。

注意事项 以下情况下禁止访问 TQ0CCR0 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。

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



(a) 作为比较寄存器时的功能

TQ0CCR0 寄存器可以在 TQ0CTL0.TQ0CE 位 = 1 时被改写。

向 TQ0CCR0 寄存器中设置的数值将被传送到 CCR0 缓冲寄存器中。当 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相匹配时，会产生一个比较匹配中断请求信号 (INTTQ0CC0)。如果允许了 TOQ00 引脚输出，那么 TOQ00 引脚输出的电平将被反转。

当 TQ0CCR0 寄存器在间隔定时器模式，外部事件计数模式，外部触发脉冲输出模式，单脉冲输出模式，PWM 输出模式或三角形波 PWM 模式下被用作周期寄存器时，如果 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相匹配，那么 16 位计数器将被清零 (0000H)。

(b) 作为捕捉寄存器时的功能

当 TQ0CCR0 寄存器在自由运行定时器模式下被用作捕捉寄存器时，如果检测到了捕捉触发输入引脚 (TIQ00 引脚) 的有效边沿，那么 16 位计数器的计数值将被保存到 TQ0CCR0 寄存器中。在脉宽测量模式下，如果检测到了捕捉触发输入引脚 (TIQ00 引脚) 的有效边沿，那么 16 位计数器的计数值将被保存到 TQ0CCR0 寄存器中并且同时 16 位计数器被清零 (0000H)。

即使捕捉操作和读取 TQ0CCR0 寄存器操作冲突，仍可正确读出 TQ0CCR0 寄存器中的值。

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

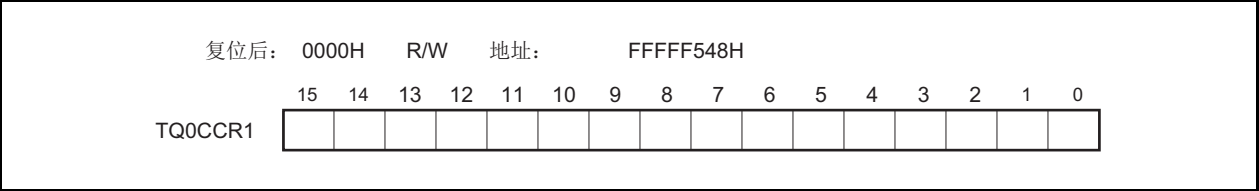
表 7-2. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	—
三角形波 PWM 模式	比较寄存器	同时写入

(8) TMQ0 捕捉/比较寄存器 1 (TQ0CCR1)

根据选择模式的不同，TQ0CCR1 寄存器可用作捕捉寄存器或比较寄存器。
该寄存器只有在自由运行模式下根据 TQ0OPT0.TQ0CCS1 位设置的不同，可以用作捕捉寄存器或比较寄存器。
在脉宽测量模式下，TQ0CCR1 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。
TQ0CCR1 寄存器可以在操作过程中进行读写。
可对该寄存器进行 16 位的读写操作。
复位将把该寄存器设置为 0000H。

- 注意事项** 以下情况下禁止访问 TQ0CCR1 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。
- 当 CPU 工作于副时钟且主时钟振荡停止时
 - 当 CPU 工作于内置振荡时钟时



(a) 作为比较寄存器时的功能

TQ0CCR1 寄存器可以在 TQ0CTL0.TQ0CE 位 = 1 时被改写。
向 TQ0CCR1 寄存器中设置的数值将被传送到 CCR1 缓冲寄存器中。当 16 位计数器的值与 CCR1 缓冲寄存器的值匹配时，产生比较匹配中断请求信号（INTTQ0CC1）。如果此时允许 TOQ01 引脚输出，则 TOQ01 引脚的输出信号的电平将反转。

(b) 作为捕捉寄存器时的功能

在自由运行定时器模式下当 TQ0CCR1 寄存器用作捕捉寄存器时，如果检测到捕捉触发输入引脚（TIQ01 引脚）的有效沿，那么 16 位计数器的计数值将被保存到 TQ0CCR1 寄存器。在脉宽测量模式下，如果检测到捕捉输入引脚（TIQ01 引脚）的有效沿，那么 16 位计数器的计数值将被保存到 TQ0CCR1 寄存器，同时 16 位计数器清为（0000H）。
即使捕捉操作和读取 TQ0CCR1 寄存器的操作产生冲突，仍可正确读出 TQ0CCR1 寄存器中的值。

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

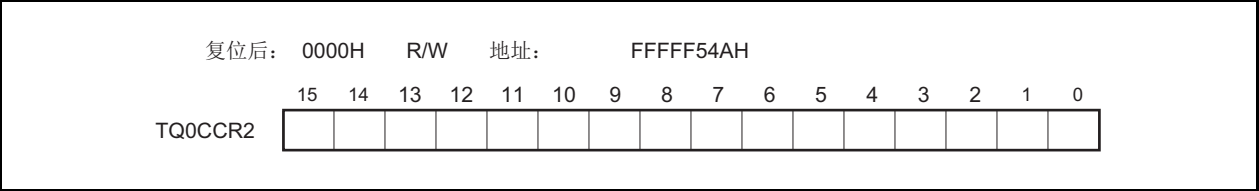
表 7-3. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	—
三角形波 PWM 模式	比较寄存器	同时写入

(9) TMQ0 捕捉/比较寄存器 2 (TQ0CCR2)

根据选择模式的不同，TQ0CCR2 寄存器可用作捕捉寄存器或比较寄存器。
该寄存器只有在自由运行模式下根据 TQ0OPT0.TQ0CCS2 位设置的不同，可以用作捕捉寄存器或比较寄存器。
在脉宽测量模式下，TQ0CCR2 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。
TQ0CCR2 寄存器可以在操作过程中进行读写
可对该寄存器进行 16 位的读写操作。
复位将把该寄存器设置为 0000H。

- 注意事项** 以下情况下禁止访问 TQ0CCR2 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。
- 当 CPU 工作于副时钟且主时钟振荡停止时
 - 当 CPU 工作于内置振荡时钟时



(a) 作为比较寄存器时的功能

TQ0CCR2 寄存器可以在 TQ0CTL0.TQ0CE 位 = 1 时被改写。
向 TQ0CCR2 寄存器中设置的数值将被传送到 CCR2 缓冲寄存器中。当 16 位计数器的值与 CCR2 缓冲寄存器的值匹配时，产生比较匹配中断请求信号（INTTQ0CC2）。如果此时允许 TOQ02 引脚输出，则 TOQ02 引脚的输出信号的电平将反转。

(b) 作为捕捉寄存器时的功能

在自由运行定时器模式下当 TQ0CCR2 寄存器用作捕捉寄存器时，如果检测到捕捉触发输入引脚（TIQ02 引脚）的有效沿，那么 16 位计数器的计数值将被保存到 TQ0CCR2 寄存器。在脉宽测量模式下，如果检测到捕捉输入引脚（TIQ02 引脚）的有效沿，那么 16 位计数器的计数值将被保存到 TQ0CCR2 寄存器，同时 16 位计数器清为（0000H）。
即使捕捉操作和读取 TQ0CCR1 寄存器的操作产生冲突，仍可正确读出 TQ0CCR1 寄存器中的值。

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 7-4. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

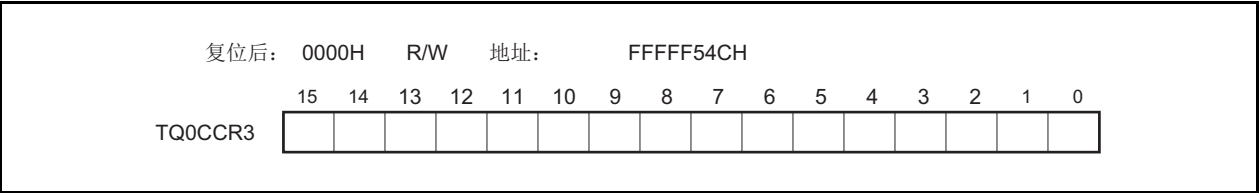
工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	—
三角形波 PWM 模式	比较寄存器	同时写入

(10) TMQ0 捕捉/比较寄存器 3 (TQ0CCR3)

根据选择模式的不同，TQ0CCR3 寄存器可用作捕捉寄存器或比较寄存器。
该寄存器只有在自由运行模式下根据 TQ0OPT0.TQ0CCS3 位设置的不同，可以用作捕捉寄存器或比较寄存器。
在脉宽测量模式下，TQ0CCR3 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。
TQ0CCR3 寄存器可以在操作过程中进行读写
可对该寄存器进行 16 位的读写操作。
复位将把该寄存器设置为 0000H。

注意事项 以下情况下禁止访问 TQ0CCR3 寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。

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



(a) 作为比较寄存器时的功能

TQ0CCR3 寄存器可以在 TQ0CTL0.TQ0CE 位 = 1 时被改写。

向 TQ0CCR3 寄存器中设置的数值将被传送到 CCR3 缓冲寄存器中。当 16 位计数器的值与 CCR3 缓冲寄存器的值匹配时，产生比较匹配中断请求信号 (INTTQ0CC3)。如果此时允许 TOQ03 引脚输出，则 TOQ03 引脚的输出信号的电平将反转。

(b) 作为捕捉寄存器时的功能

在自由运行定时器模式下当 TQ0CCR3 寄存器用作捕捉寄存器时，如果检测到捕捉触发输入引脚 (TIQ03 引脚) 的有效沿，那么 16 位计数器的计数值将被保存到 TQ0CCR3 寄存器。在脉宽测量模式下，如果检测到捕捉输入引脚 (TIQ03 引脚) 的有效沿，那么 16 位计数器的计数值将被保存到 TQ0CCR3 寄存器，同时 16 位计数器清为 (0000H)。

即使捕捉操作和读取 TQ0CCR3 寄存器的操作产生冲突，仍可正确读出 TQ0CCR3 寄存器中的值。

下表中列出了捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 7-5. 捕捉/比较寄存器在各模式下的功能以及对比较寄存器的写入方法

工作模式	捕捉/比较寄存器	比较寄存器的写入方法
间隔定时器	比较寄存器	可随时写入
外部事件计数器	比较寄存器	可随时写入
外部触发脉冲输出	比较寄存器	同时写入
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	同时写入
自由运行定时器	捕捉/比较寄存器	可随时写入
脉宽测量	捕捉寄存器	—
三角形波 PWM 模式	比较寄存器	同时写入

(11) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

可以通过读取缓冲寄存器 TQ0CNT 来读取 16 位计数器的计数值。

如果该寄存器在 TQ0CTL0.TQ0CE 位 = 1 时被读取，那么读取的值就是 16 位计数器的计数值。

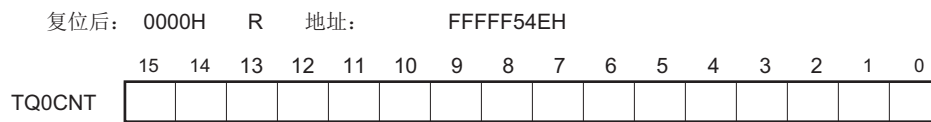
只可以对该寄存器进行 16 位的读取操作。

当 TQ0CE 位 = 0 时，TQ0CNT 寄存器被清为 0000H。如果此时对 TQ0CNT 寄存器进行读取，读取到的值将是 0000H 而不是 16 位计数器的值 (FFFFH)。

复位后，TQ0CE 位被清零的同时 TQ0CNT 寄存器的值也被清零。

注意事项 以下情况下禁止访问 TQ0CNT 寄存器。详细内容请参考 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。

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



(12) TIQ0m 引脚噪声消除控制寄存器 (Q0mNFC)

Q0mNFC 寄存器是一个为消除噪声而设置定时器 Q 输入引脚的数字噪声滤波器的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后: 00H R/W 地址: Q00NFC: FFFFB50H (TIQ00 引脚)

Q01NFC: FFFFB54H (TIQ01 引脚)

Q02NFC: FFFFB58H (TIQ02 引脚)

Q03NFC: FFFFB5CH (TIQ03 引脚)

	7	6	5	4	3	2	1	0
Q0mNFC	0	NFSTS	0	0	0	NFC2	NFC1	NFC0

(m = 0 至 3)

($m = 0$ 至 3)

NFSTS	通过数字噪声滤波器执行的采样次数的设置
0	3 次
1	2 次

NFC2	NFC1	NFC0	采样时钟
0	0	0	f _{xx}
0	0	1	f _{xx} /2
0	1	0	f _{xx} /4
0	1	1	f _{xx} /16
1	0	0	f _{xx} /32
1	0	1	f _{xx} /64
与以上不同			禁止设置

注意事项 1. 请务必将位 3 至位 5 以及位 7 清除为“0”。

2. 在设置 Q0mNFC 寄存器前，输入到定时器输入引脚（TIQ0m）的信号会通过消除的数字噪声来输出。

因此，先通过使用 Q0mNFC 寄存器来设置采样时钟（NFC2 至 NFC0）以及采样的次数（NFSTS），然后等待初始化时间=（采样时钟）×（采样的次数），再允许定时器运行。

备注 可以被精确消除的噪声的宽度等于 (采样时钟) × (采样的次数 - 1)。如果宽度窄于该噪声宽度的噪声与采样时钟同步, 那么这种噪声可能会引起计算错误。

7.5 操作

TMQ0 可进行以下操作。

操作	TQ0CTL1.TQ0EST 位 (软件触发位)	TIQ00 引脚 (外部触发输入)	捕捉/比较寄存器设置	比较寄存器写入
间隔定时器模式	无效	无效	仅比较	可随时写入
外部事件计数模式 ^{注 1}	无效	无效	仅比较	可随时写入
外部触发脉冲输出模式 ^{注 2}	有效	有效	仅比较	同时写入
单脉冲输出模式 ^{注 2}	有效	有效	仅比较	可随时写入
PWM 输出模式	无效	无效	仅比较	同时写入
自由运行定时器模式	无效	无效	可选	可随时写入
脉宽测量模式 ^{注 2}	无效	无效	仅捕捉	不可操作
三角形波 PWM 模式	无效	无效	仅比较	同时写入

- 注
1. 使用外部事件计数模式时，请指定为不检测 TIQ00 引脚捕捉触发输入有效沿（通过将 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位清除为“00”）。
 2. 使用外部触发脉冲输出模式，单脉冲输出模式和脉宽测量模式时，请选择内部时钟作为计数时钟（通过将 TQ0CTL1.TQ0EEE 位清零）。

7.5.1 间隔定时器模式 (TQ0MD2 至 TQ0MD0 位 = 000)

在间隔定时器模式下，当 TQ0CTL0.TQ0CE 位 = 1 时，每隔指定间隔时间都会产生一个中断请求信号 (INTTQ0CC0)，同时可以从 TOQ00 引脚输出一个周期为 2 倍间隔时间的方波。

通常情况下，间隔定时器模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。

图 7-2. 间隔定时器的配置图

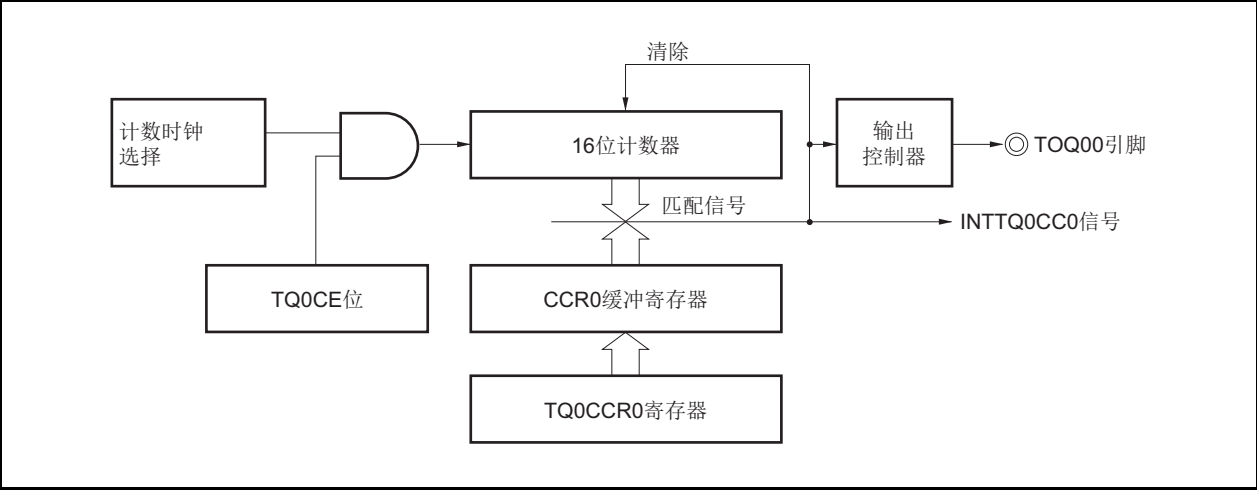
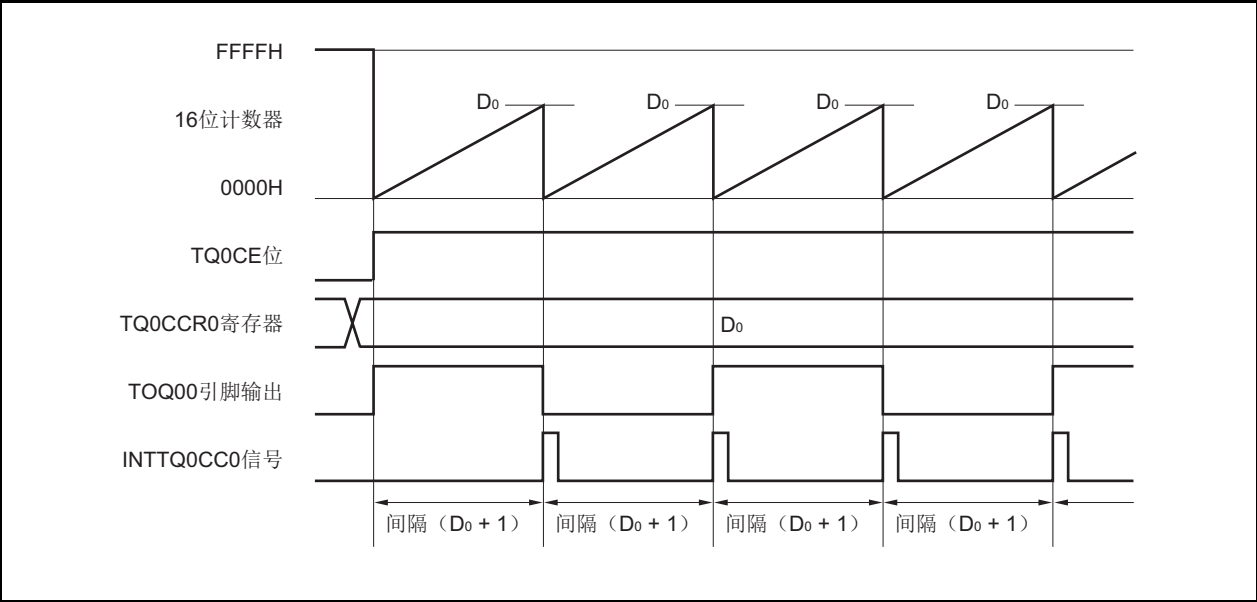


图 7-3. 间隔定时器模式操作的基本时序



16 位计数器将在 TQ0CE 位被置 1 时，与计数时钟同步从 FFFFH 清零为 0000H 并开始计数。此时，TOQ00 引脚的输出电平被反转。另外，TQ0CCR0 寄存器的设置值也会在此时被传送至 CCR0 缓冲寄存器。

当 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值匹配时，16 位计数器清零，TOQ00 引脚输出电平反转并且产生一个比较匹配中断请求信号（INTTQ0CC0）。

间隔时间可由以下算式求得。

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

图 7-4. 间隔定时器模式操作的寄存器设置 (1/2)

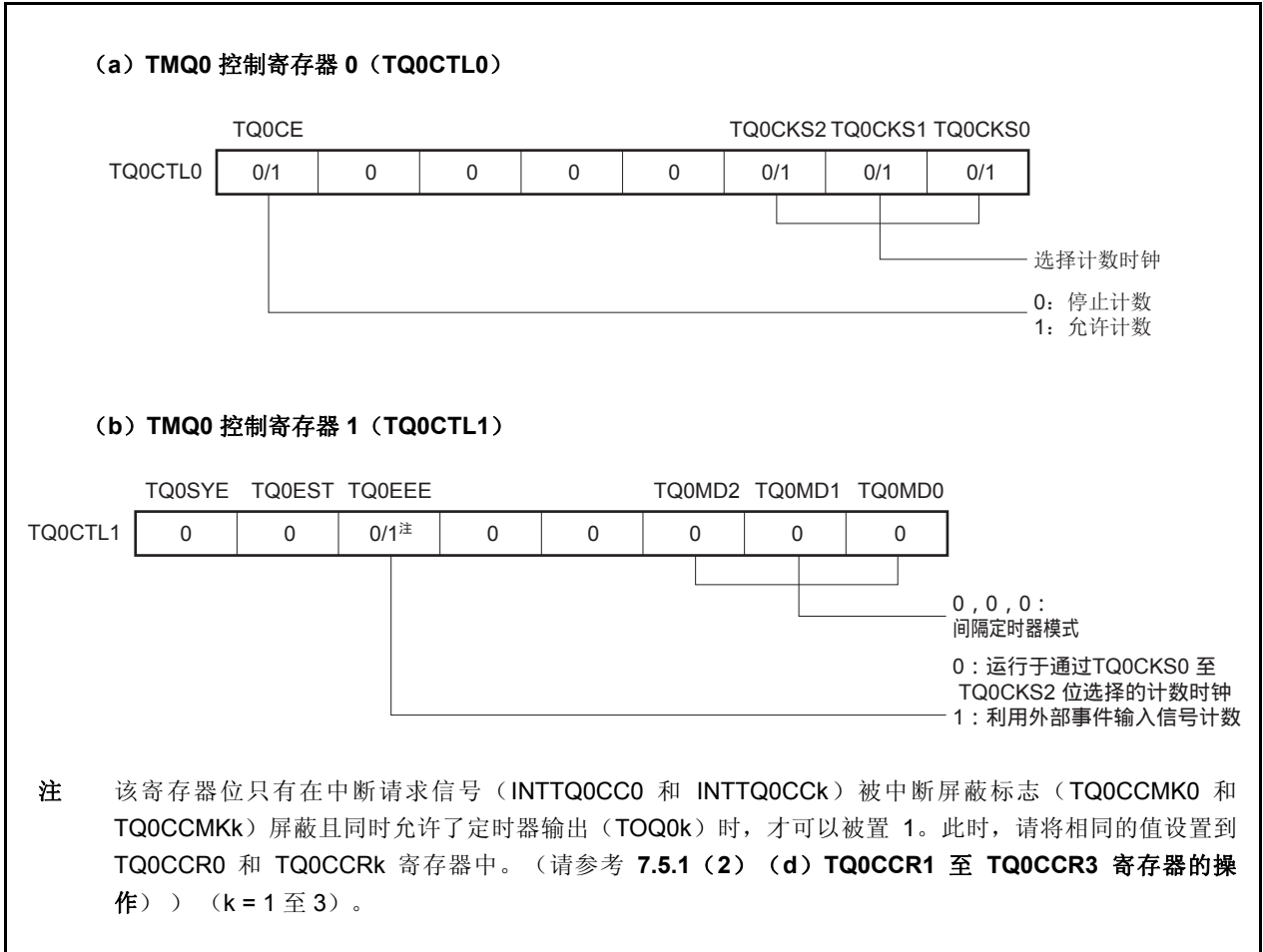
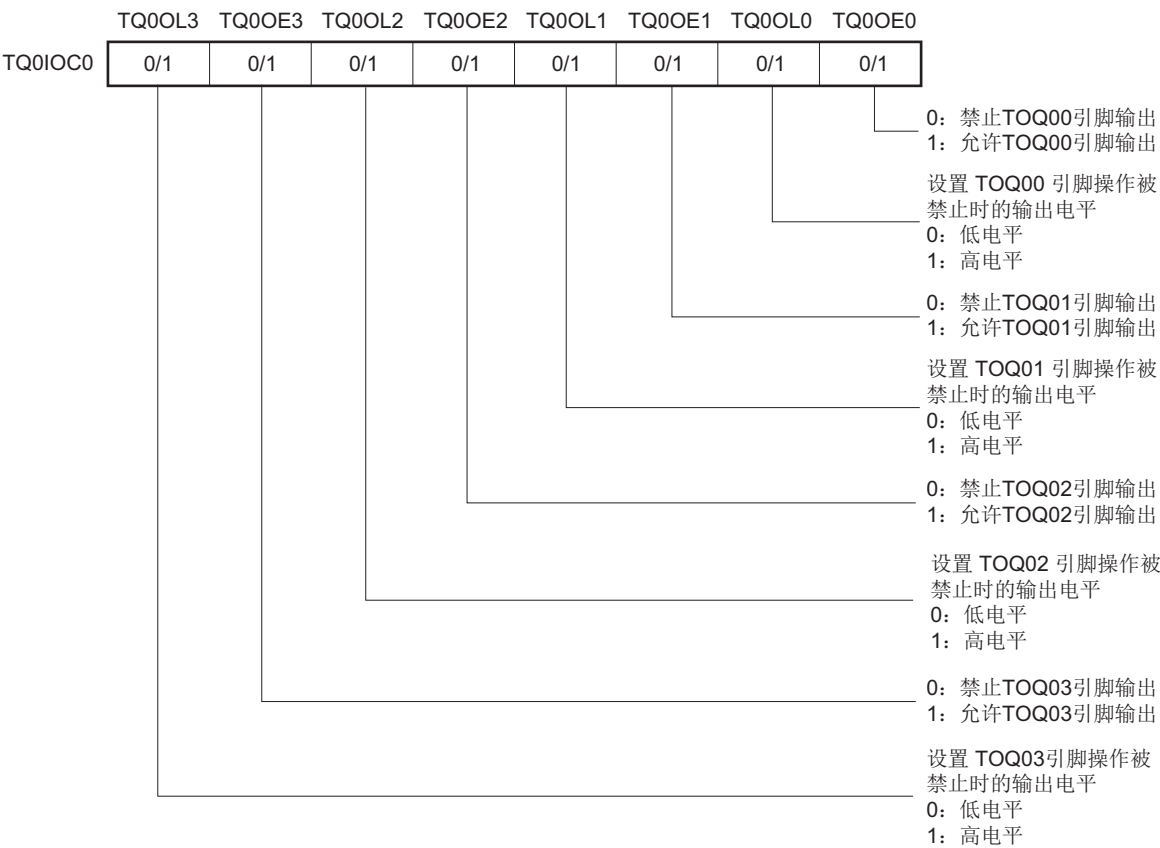


图 7-4. 间隔定时器模式操作的寄存器设置 (2/2)

(c) TMQ0 I/O 控制寄存器 0 (TQ0IOC0)



(d) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

可以通过读取 TQ0CNT 寄存器来读取 16 位计数器的计数值。

(e) TMQ0 捕捉/比较寄存器 0 (TQ0CCR0)

如果将 TQ0CCR0 寄存器设置为 D₀，那么间隔时间如下。

间隔时间 = (D₀ + 1) × 计数时钟周期

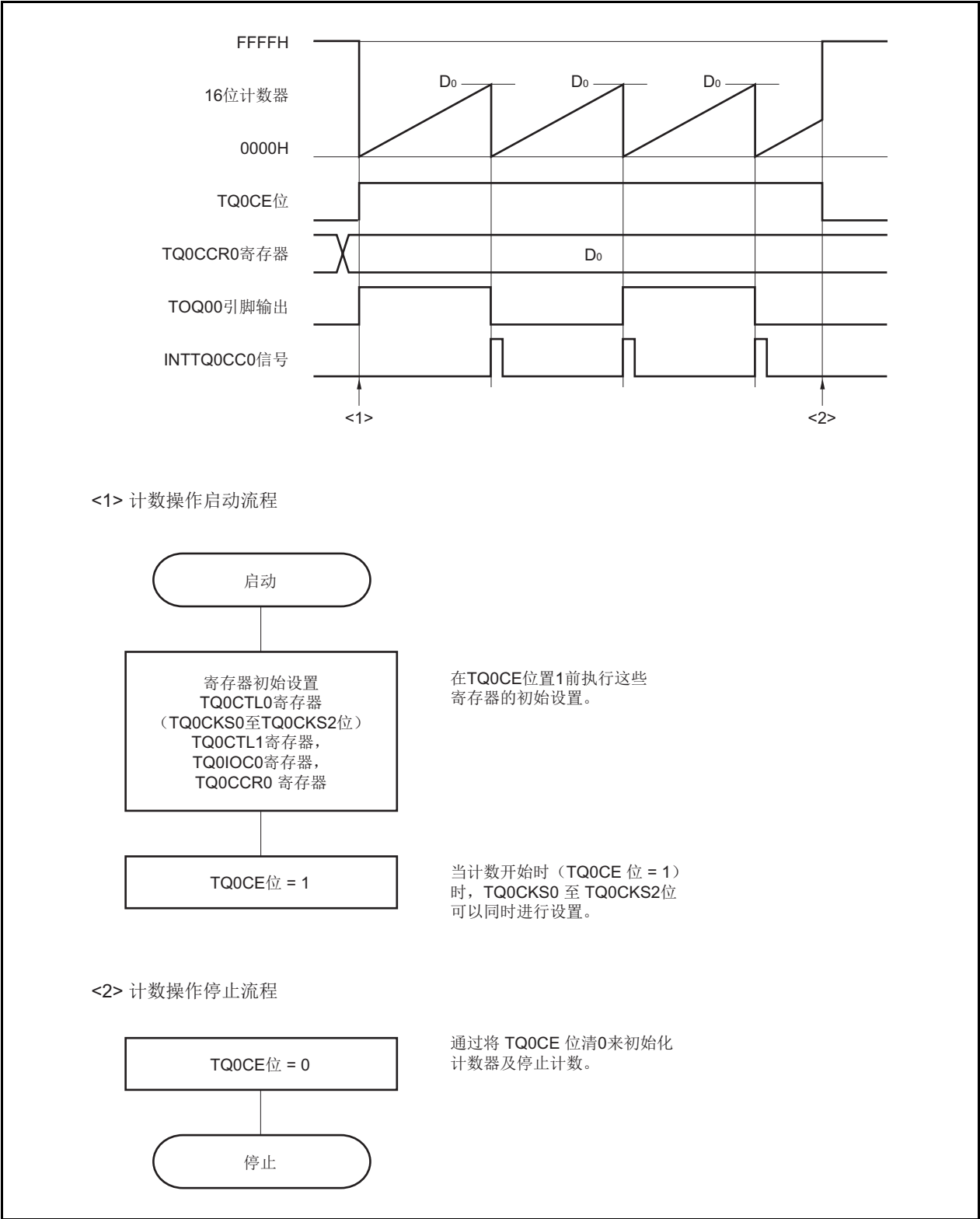
(f) TMQ0 捕捉/比较寄存器 1 至 3 (TQ0CCR1 至 TQ0CCR3)

通常情况下，间隔定时器模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。但对 TQ0CCR1 至 TQ0CCR3 寄存器的设置值也将被传送到 CCR1 至 CCR3 缓冲寄存器。当 16 位计数器的计数值与 CCR1 至 CCR3 缓冲寄存器中的数值匹配时，也会产生一个比较匹配中断请求信号 (INTTQ0CC1 至 INTTQ0CCR3)。因此，请使用相应的中断屏蔽标志 (TQ0CCMK1 至 TQ0CCMK3) 将中断请求屏蔽。

备注 间隔定时器模式下，未使用 TMQ0 I/O 控制寄存器 1 (TQ0IOC1)，TMQ0 I/O 控制寄存器 2 (TQ0IOC2) 以及 TMQ0 选项寄存器 0 (TQ0OPT0)。

(1) 间隔定时器模式操作流程

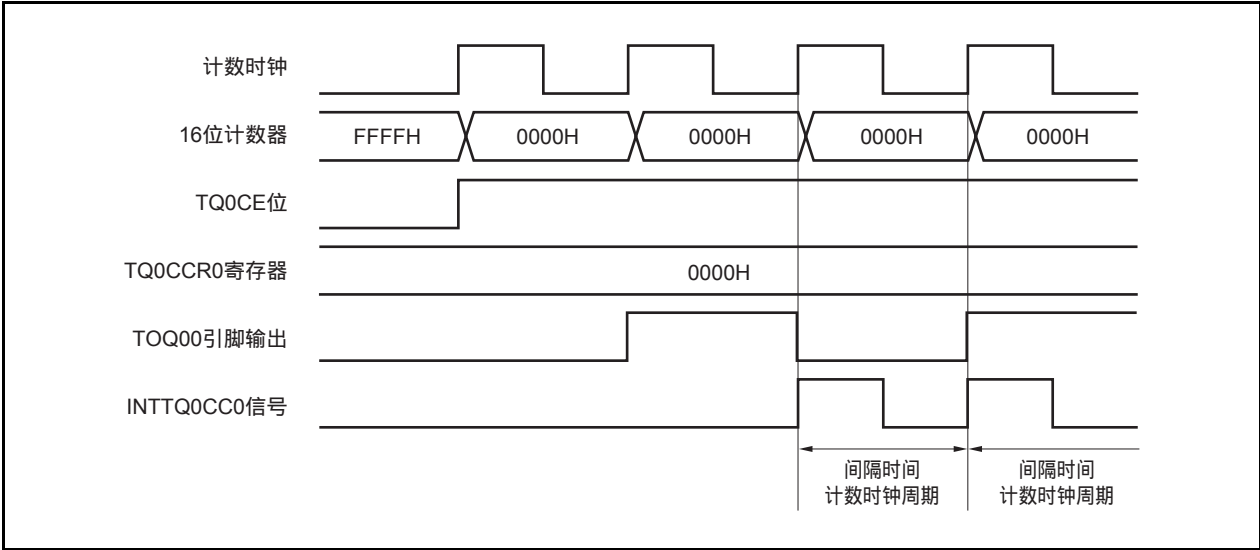
图 7-5. 间隔定时器模式下的软件处理流程



(2) 间隔定时器模式操作时序

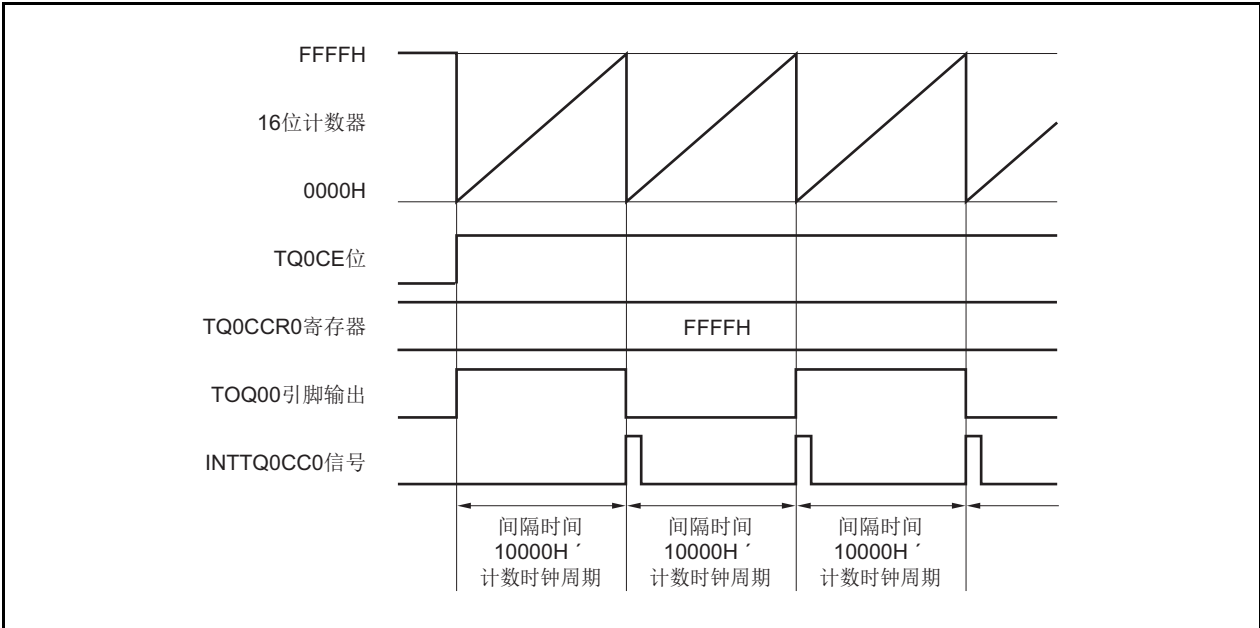
(a) TQ0CCR0 寄存器被设置为 0000H 时的操作

如果 TQ0CCR0 寄存器被设置为 0000H, 那么在第二个计数时钟之后 (含第二个), 都会产生 INTTQ0CC0 信号, 并且 TOQ00 引脚的电平都会反转。
16 位计数器的计数值始终保持为 0000H。



(b) TQ0CCR0 寄存器被设置为 FFFFH 时的操作

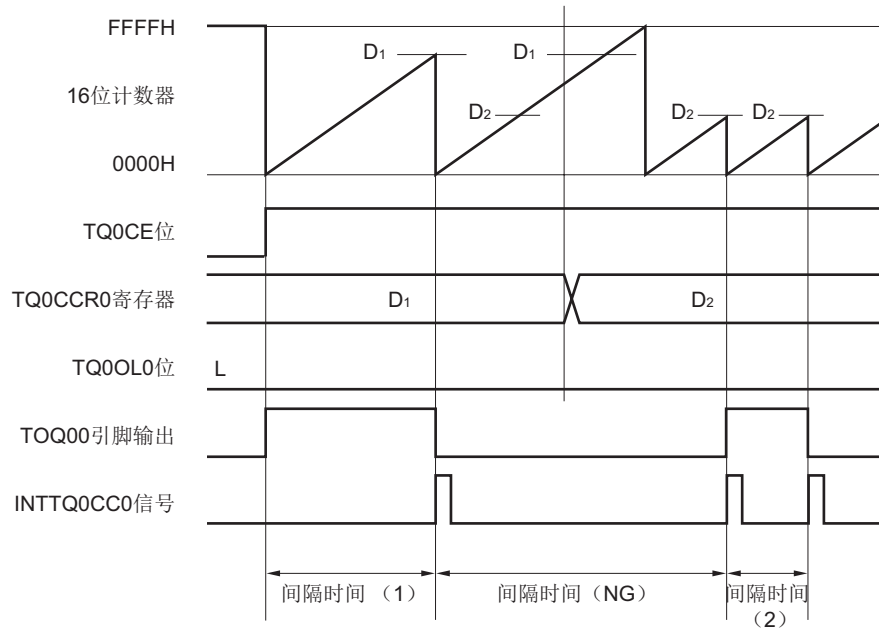
如果 TQ0CCR0 寄存器被设置为 FFFFH, 那么 16 位计数器将计数到 FFFFH。并在下一个计数时钟时刻同步被清零。同时产生 INTTQ0CC0 信号, TOQ00 引脚电平反转。此时, 不会产生溢出中断请求信号 (INTTQ0OV), 溢出标志位 (TQ0OPT0.TQ0OVF 位) 也不会被置 1。



(c) 改写 TQ0CCR0 寄存器时需要注意的事项

要将 TQ0CCR0 寄存器的值改小，首先停止计数然后再更改设置值。

如果在计数时向 TQ0CCR0 寄存器写入更小的值，16 位计数器可能会溢出。



备注

间隔时间 (1) : $(D_1 + 1) \times \text{计数时钟周期}$

间隔时间 (NG) : $(10000H + D_2 + 1) \times \text{计数时钟周期}$

间隔时间 (2) : $(D_2 + 1) \times \text{计数时钟周期}$

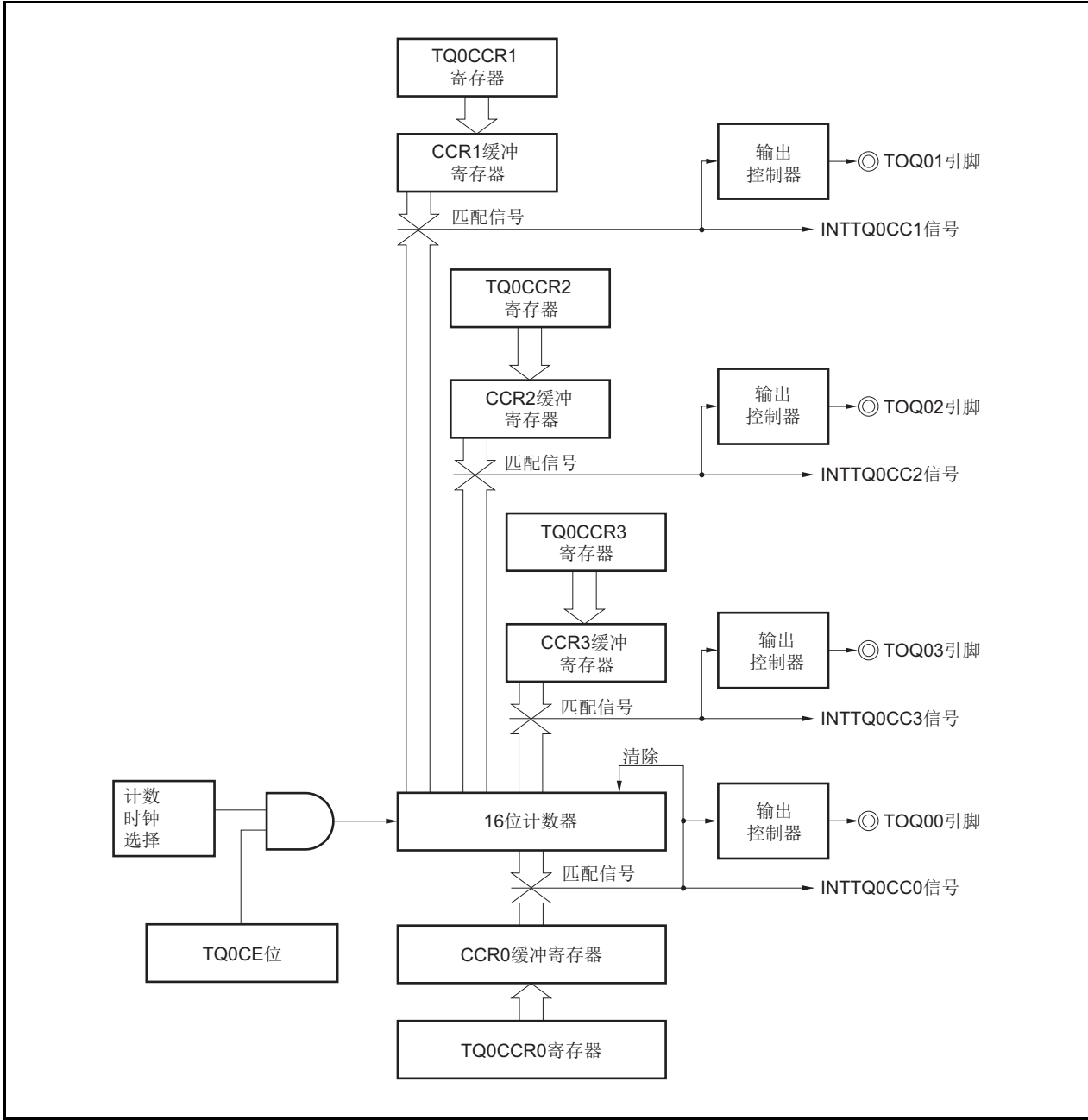
当计数到大于 D_2 小于 D_1 时如果 TQ0CCR0 寄存器的值从 D_1 变为 D_2 ，改写 TQ0CCR0 寄存器后，此值立即传送到 CCR0 缓冲寄存器。因此，16 位计数器的计数值将与 D_2 进行比较。

因为计数值已经超过了 D_2 ，因此 16 位计数器会继续计数到 FFFFH 并产生溢出，然后从 0000H 开始再次计数。在重新计数过程中，当计数值与 D_2 匹配时，产生 INTTQ0CC0 信号，TOQ00 引脚的输出信号的电平反转。

因此，可能不会在最初预期的 “ $(D_1 + 1) \times \text{计数时钟周期}$ ” 或 “ $(D_2 + 1) \times \text{计数时钟周期}$ ” 间隔时间产生 INTTQ0CC0 信号，而是在 “ $(10000H + D_2 + 1) \times \text{计数时钟周期}$ ” 的间隔时间产生。

(d) 操作 TQ0CCR1 至 TQ0CCR3 寄存器

图 7-6. TQ0CCR1 至 TQ0CCR3 寄存器的构成

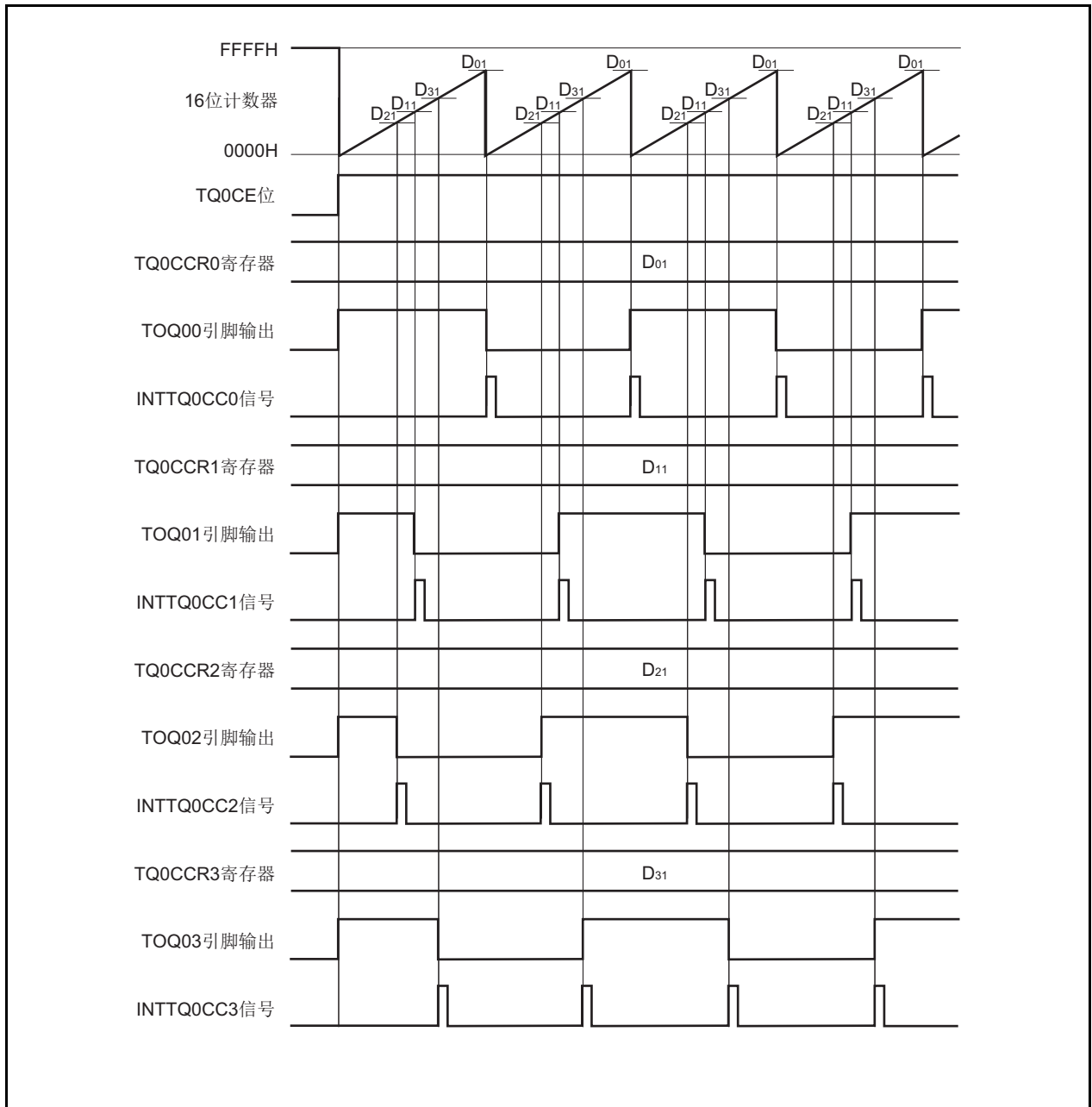


如果 TQ0CCRk 寄存器的设置值小于 TQ0CCR0 寄存器的设置值，则每个周期产生一次 INTTQ0CCk 信号。同时，TOQ0k 引脚的输出信号的电平反转。

TOQ0k 引脚输出的方波周期与 TOQ00 引脚的方波周期相同。

备注 k = 1 至 3

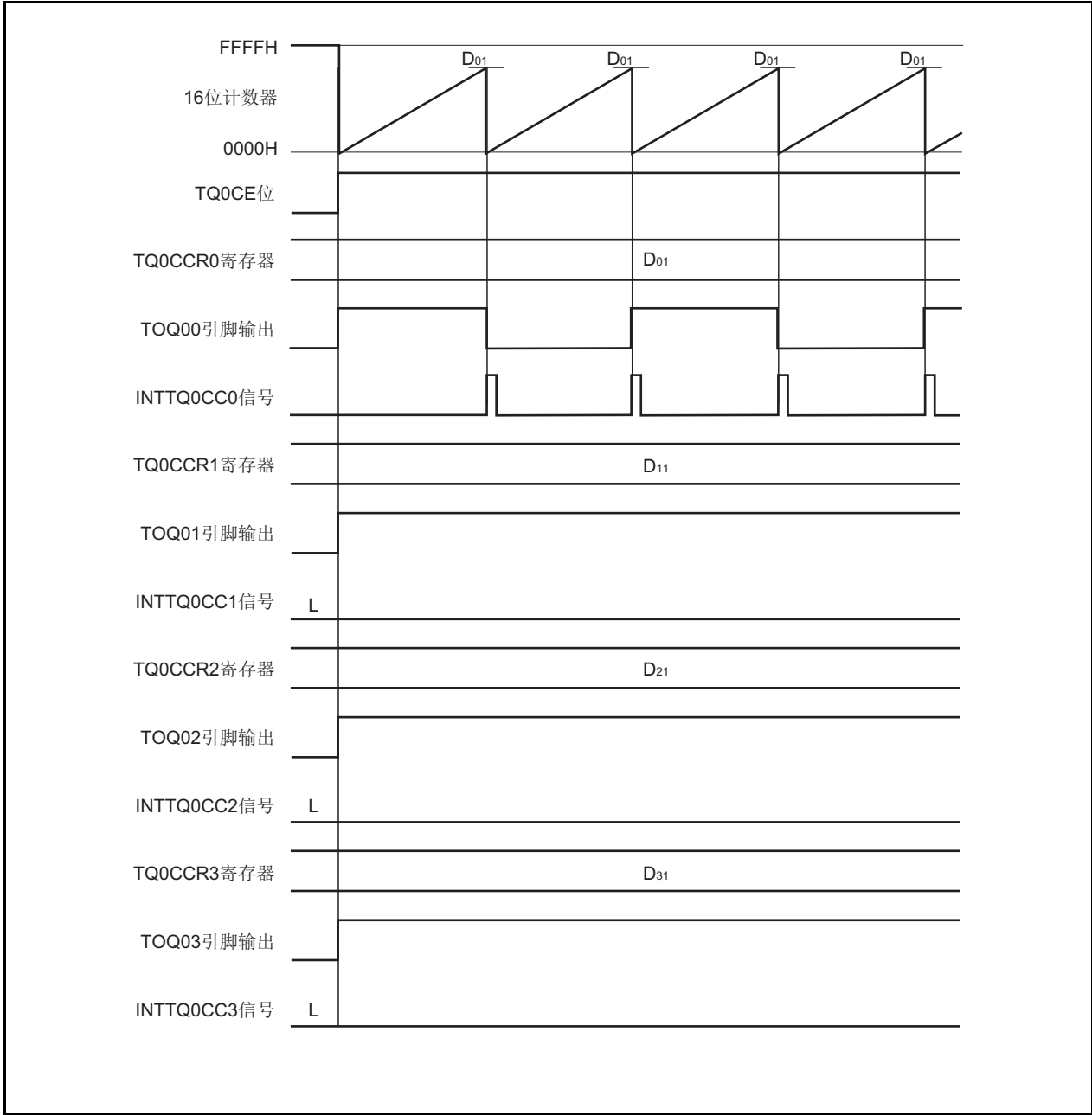
图 7-7. 当 $D_{01} \geq D_{k1}$ 时的时序图



如果 TQ0CCRk 寄存器的设置值大于 TQ0CCR0 寄存器，16 位计数器的计数值不会与 TQ0CCRk 寄存器匹配。因此，不产生 INTTQ0CCk 信号，TOQ0k 引脚的输出也不发生变化。

备注 k = 1 至 3

图 7-8. 当 $D_{01} < D_{k1}$ 时的时序图



7.5.2 外部事件计数模式 (TQ0MD2 至 TQ0MD0 位 = 001)

在外部事件计数模式，只有当 TQ0CTL0.TQ0CE 位置 1 时外部事件计数输入的有效沿才有效，每次计到指定数量的有效沿后都会产生中断请求信号 (INTTQ0CC0)。该模式下不能使用 TOQ00 引脚。

通常，TQ0CCR1 至 TQ0CCR3 寄存器不用于外部事件计数模式。

图 7-9. 外部事件计数模式结构

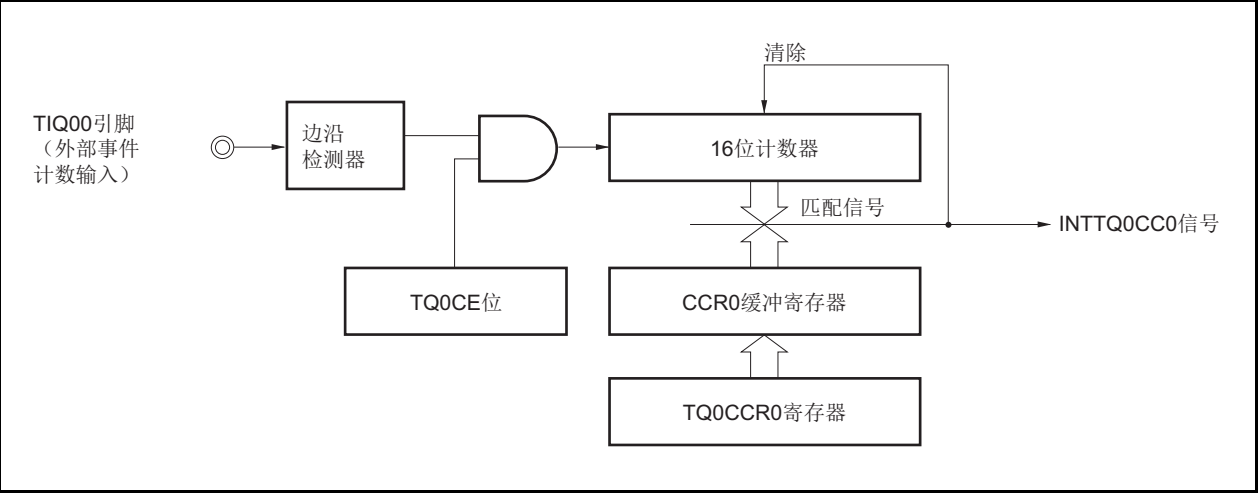
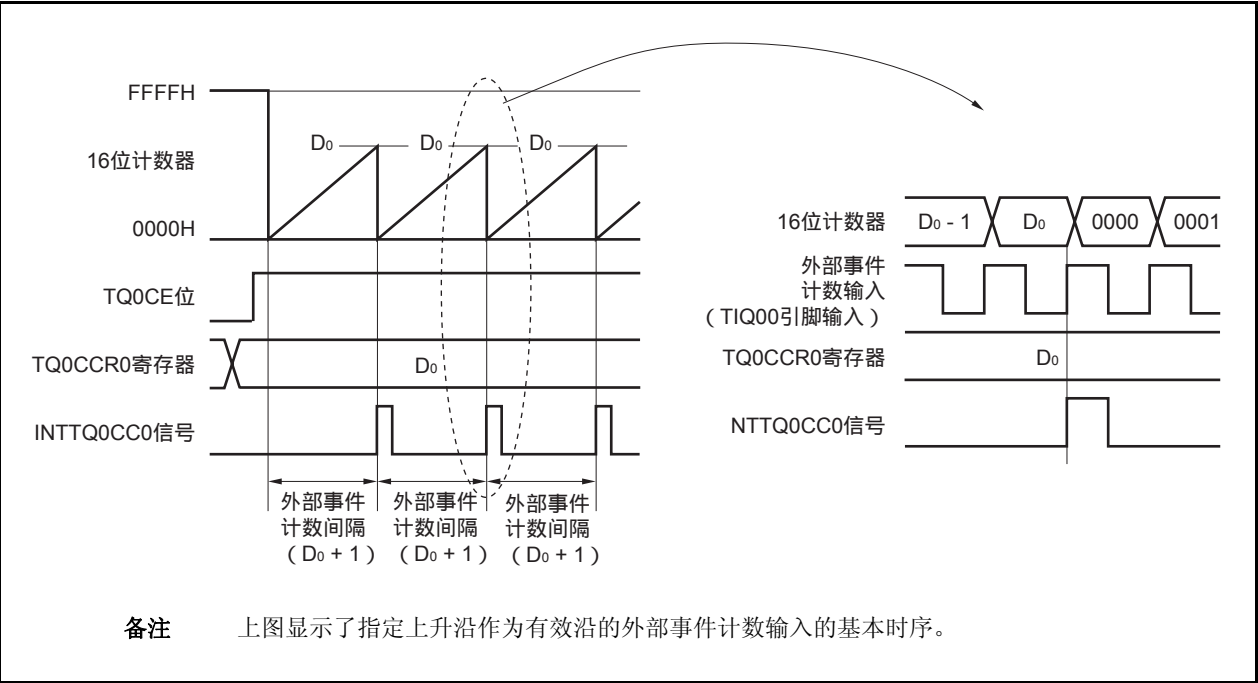


图 7-10. 外部事件计数模式的基本时序



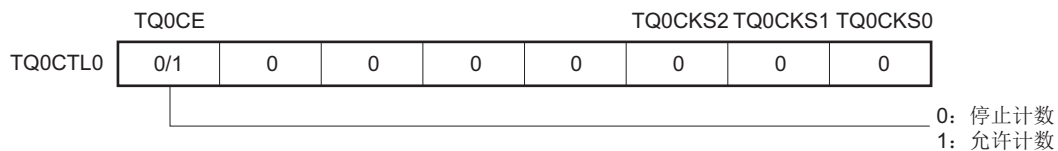
当 TQ0CE 位置 1 时，16 位计数器的值从 FFFFH 清除为 0000H。每次外部事件计数输入的有效沿时计数器计数。此外，TQ0CCR0 寄存器的设置值发送到 CCR0 缓冲寄存器。

当 16 位计数器的计数值与 CCR0 缓冲寄存器匹配时，16 位计数器被清为 0000H，且产生比较匹配中断请求信号 (INTTQ0CC0)。

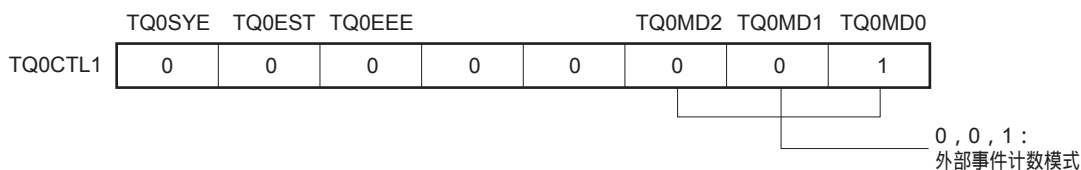
每次检测到外部事件计数输入的有效沿时 (TQ0CCR0 寄存器的设置值+1)，都会产生 INTTQ0CC0 信号。

图 7-11. 外部事件计数模式操作的寄存器设置 (1/2)

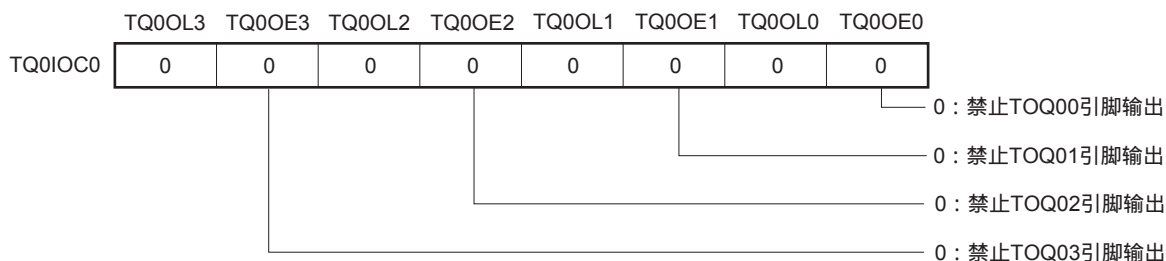
(a) TMQ0 控制寄存器 0 (TQ0CTL0)



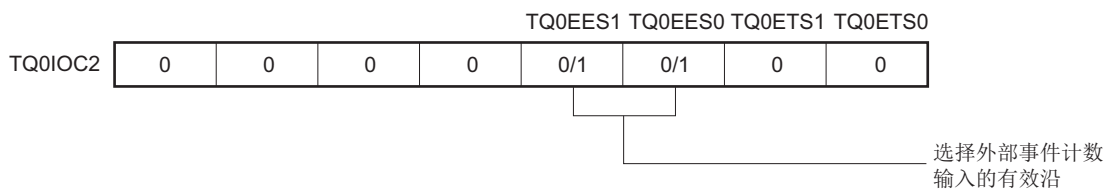
(b) TMQ0 控制寄存器 1 (TQ0CTL1)



(c) TMQ0 I/O 控制寄存器 0 (TQ0IOC0)



(d) TMQ0 I/O 控制寄存器 2 (TQ0IOC2)



(e) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

可以通过读取 TQ0CNT 寄存器来读取 16 位计数器的计数值。

图 7-11. 外部事件计数模式操作的寄存器设置 (2/2)

(f) TMQ0 捕捉/比较寄存器 0 (TQ0CCR0)

如果向 TQ0CCR0 寄存器写入 D_0 ，则计数器被清零，并且将在外部事件计数达到 $(D_0 + 1)$ 时产生比较匹配中断请求信号 (INTTQ0CC0)。

(g) TMQ0 捕捉/比较寄存器 1 至 3 (TQ0CCR1 至 TQ0CCR3)

通常情况下，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，也会产生一个比较匹配中断请求信号 (INTTQ0CC1 至 INTTQ0CC3)。

因此，请使用相应的中断屏蔽标志 (TQ0CCMK1 至 TQ0CCMK3) 将中断请求屏蔽。

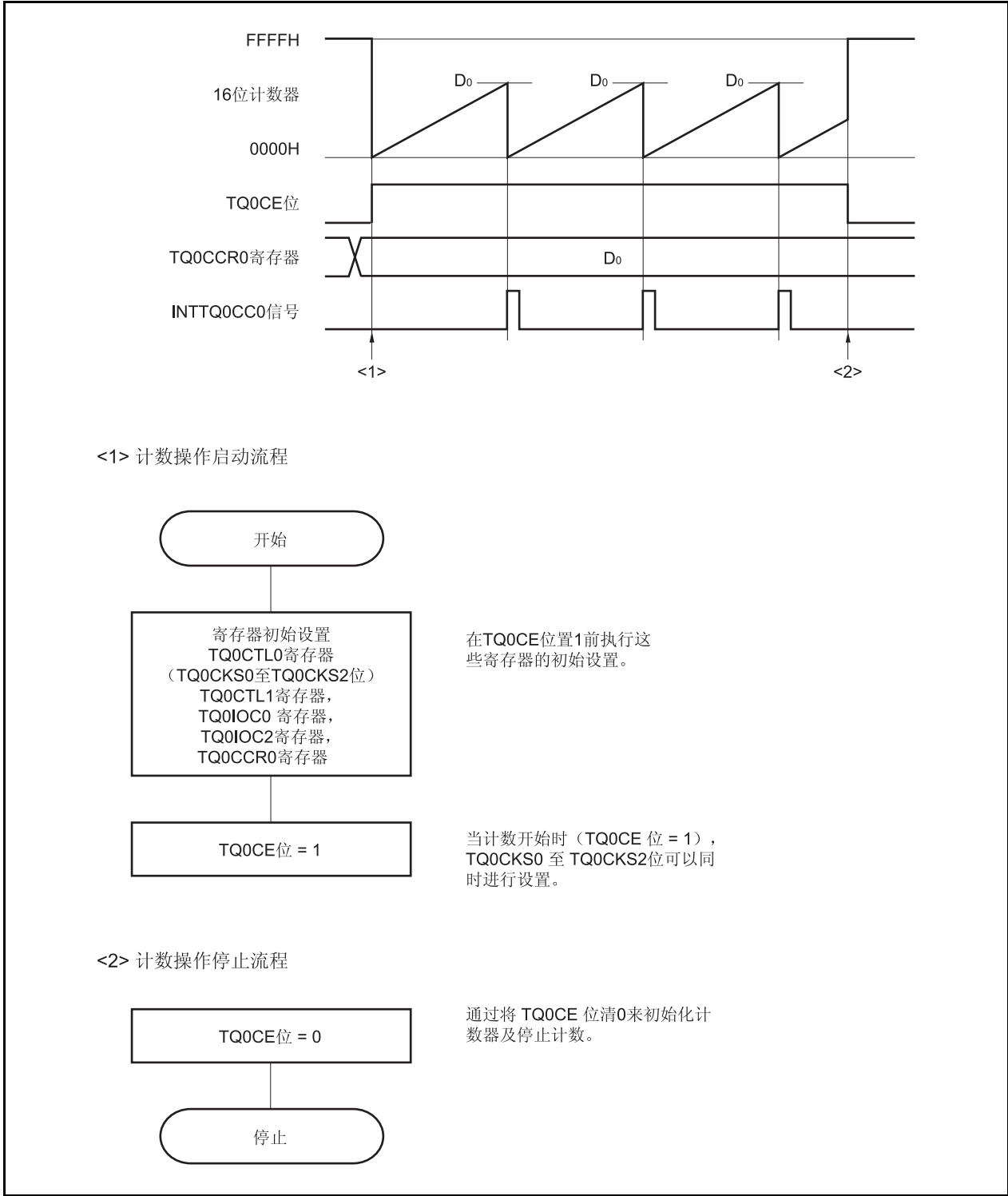
注意事项 当外部时钟用于计数时钟时，外部时钟只能从 TIQ00 引脚输入。这时，将 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 置为 00 (捕捉触发器输入 (TIQ00 引脚)：无边沿检测)

备注 外部事件计数模式下，并未使用 TMQ0 I/O 控制寄存器 1 (TQ0IOC1) 以及 TMQ0 选项寄存器 0 (TQ0OPT0)。

<R>

(1) 外部事件计数模式操作流程

图 7-12. 外部事件计数模式的软件处理流程

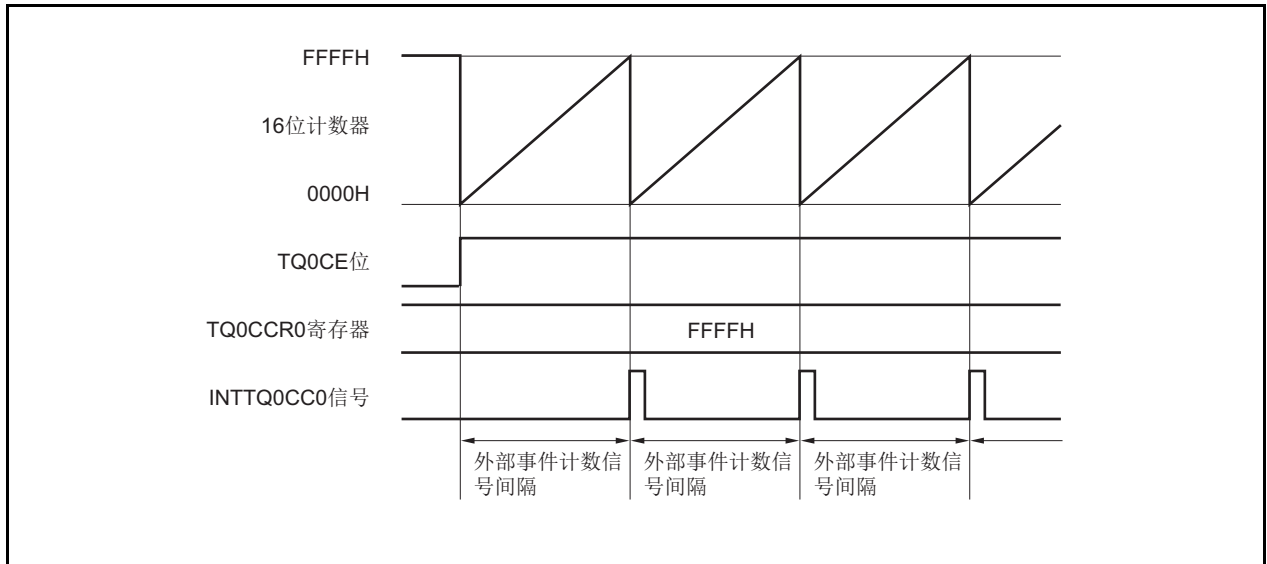


(2) 外部事件计数模式的操作时序

- 注意事项
1. 在外部事件计数模式中，不要将 TQ0CCR0 寄存器设置为 0000H。
 2. 在外部事件计数模式中，禁止使用定时器输出功能。如果要将外部事件计数输入作为定时器输出使用，那么请将工作模式设置为间隔定时器模式并将外部事件计数输入设定为计数时钟 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 000, TQ0CTL1.TQ0EEE 位 = 1)。

(a) TPnCCR0 寄存器被设置为 FFFFH 时的操作

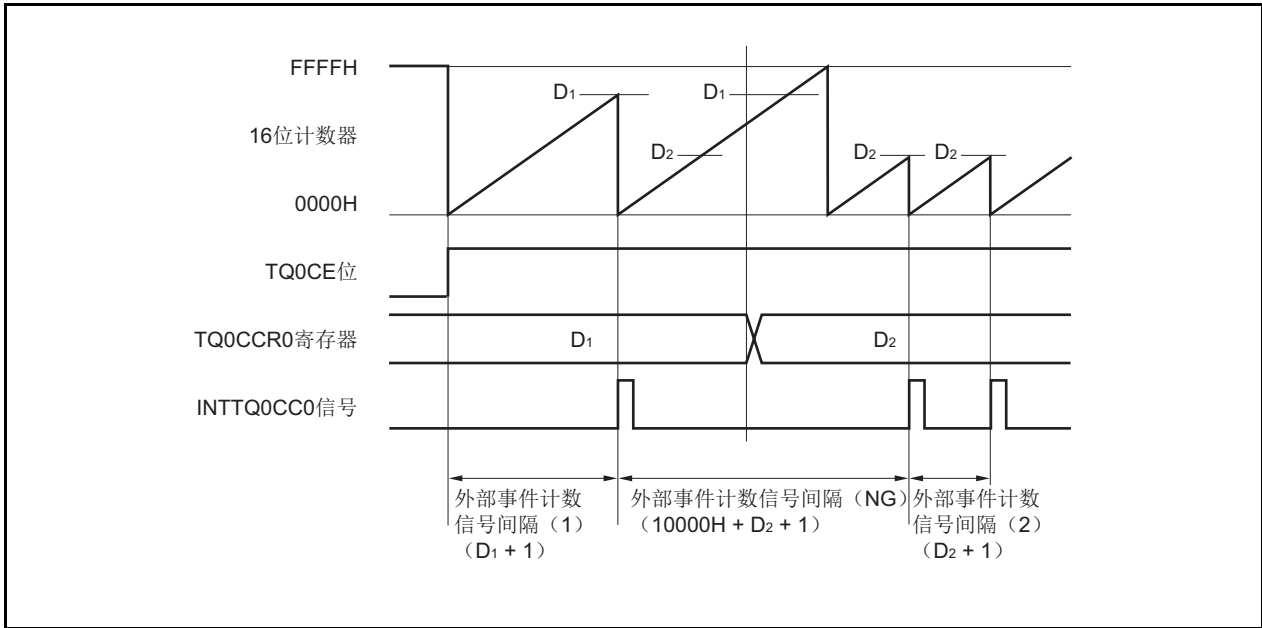
如果 TQ0CCR0 寄存器被设置为 FFFFH，那么 16 位计数器将在每次检测到外部事件计数信号的有效沿处计数，直到 FFFFH。并在下一个计数事件时刻被同步清零，同时产生 INTTQ0CC0 信号。此时，溢出标志位 (TQ0OPT0.TQ0OVF 位) 不会被置 1。



(b) 改写 TQ0CCR0 寄存器时需要注意的事项

将 TQ0CCR0 寄存器的值改小时，要先停止计数，再对设定值进行改变。

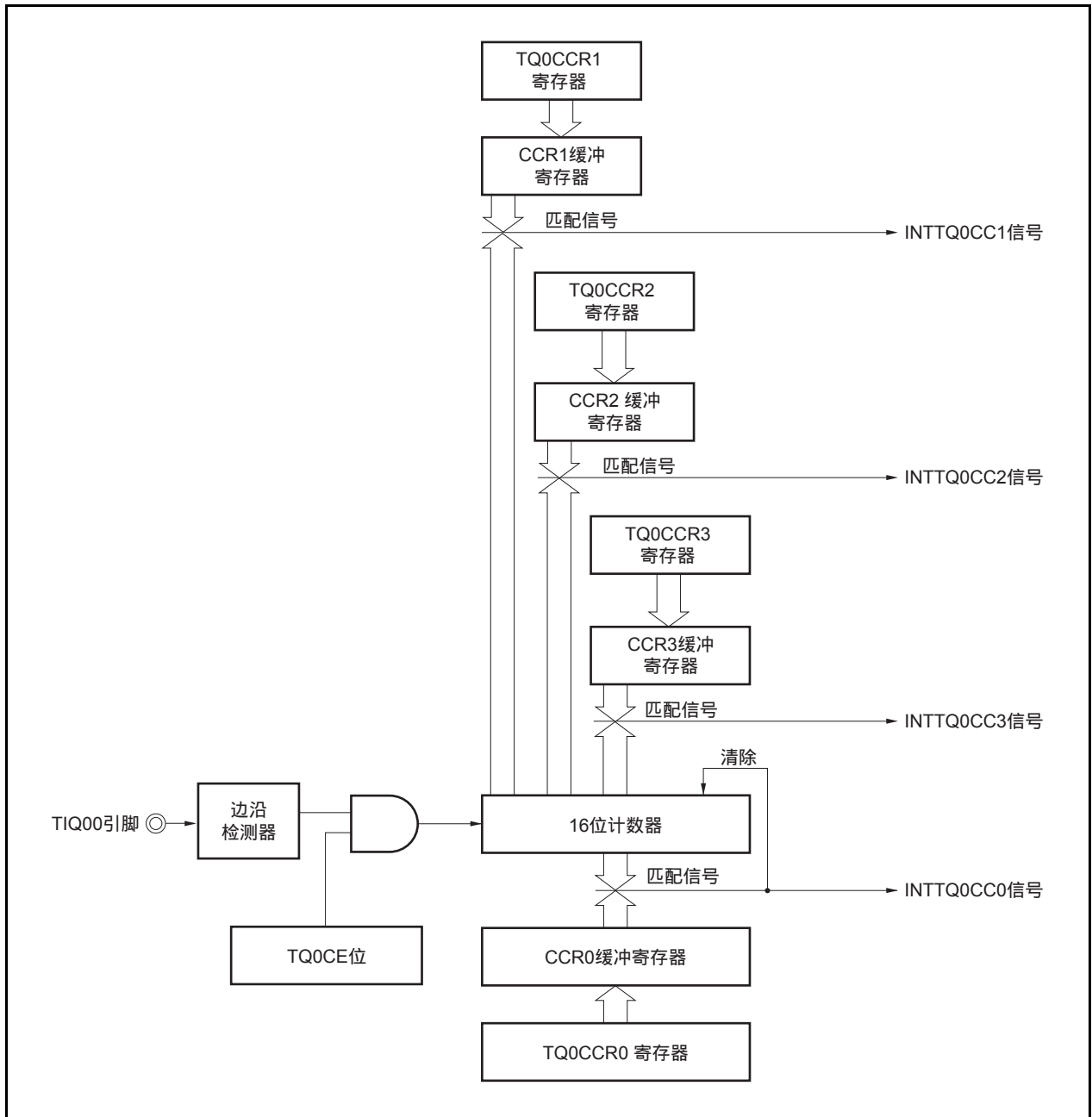
如果在计数过程中将 TQ0CCR0 寄存器的值改小，那么 16 位计数器可能产生溢出。



如果当前计数值大于 D₂ 但小于 D₁，将 TQ0CCR0 寄存器的值从 D₁ 改变为 D₂，新的设定值将在改写后立即从 TQ0CCR0 寄存器传送到 CCR0 缓冲寄存器中，那么，16 位计数器的计数值将与 D₂ 进行比较。由于计数值已经超过了 D₂，因此 16 位计数器会继续计数到 FFFFH 并产生溢出，之后重新从 0000H 开始计数。在重新计数过程中，计数值与 D₂ 匹配时，才会产生 INTTQ0CC0 信号。因此可见，INTTQ0CC0 信号不会在预期的事件计数次数“(D₁ + 1) 次”或“(D₂ + 1) 次”时产生，而是在一个“(10000H + D₂ + 1) 次”的计数次数时产生。

(c) TQ0CCR1 至 TQ0CCR3 寄存器的操作

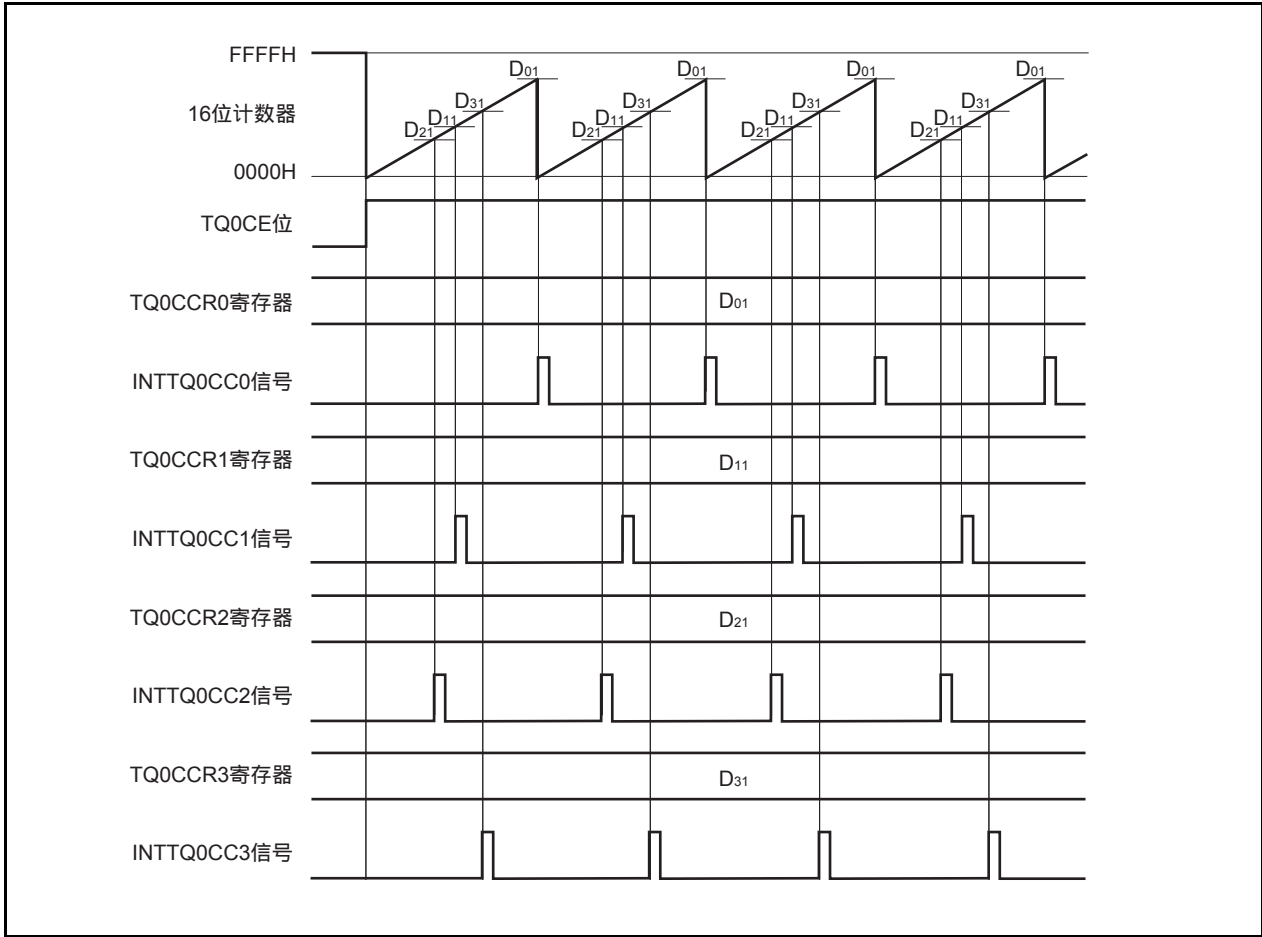
图 7-13. TQ0CCR1 至 TQ0CCR3 寄存器的配置



如果 TQ0CCRk 寄存器的值小于 TQ0CCR0 寄存器的值，那么 INTTQ0CCk 信号将每周产生一次。

备注 k = 1 至 3

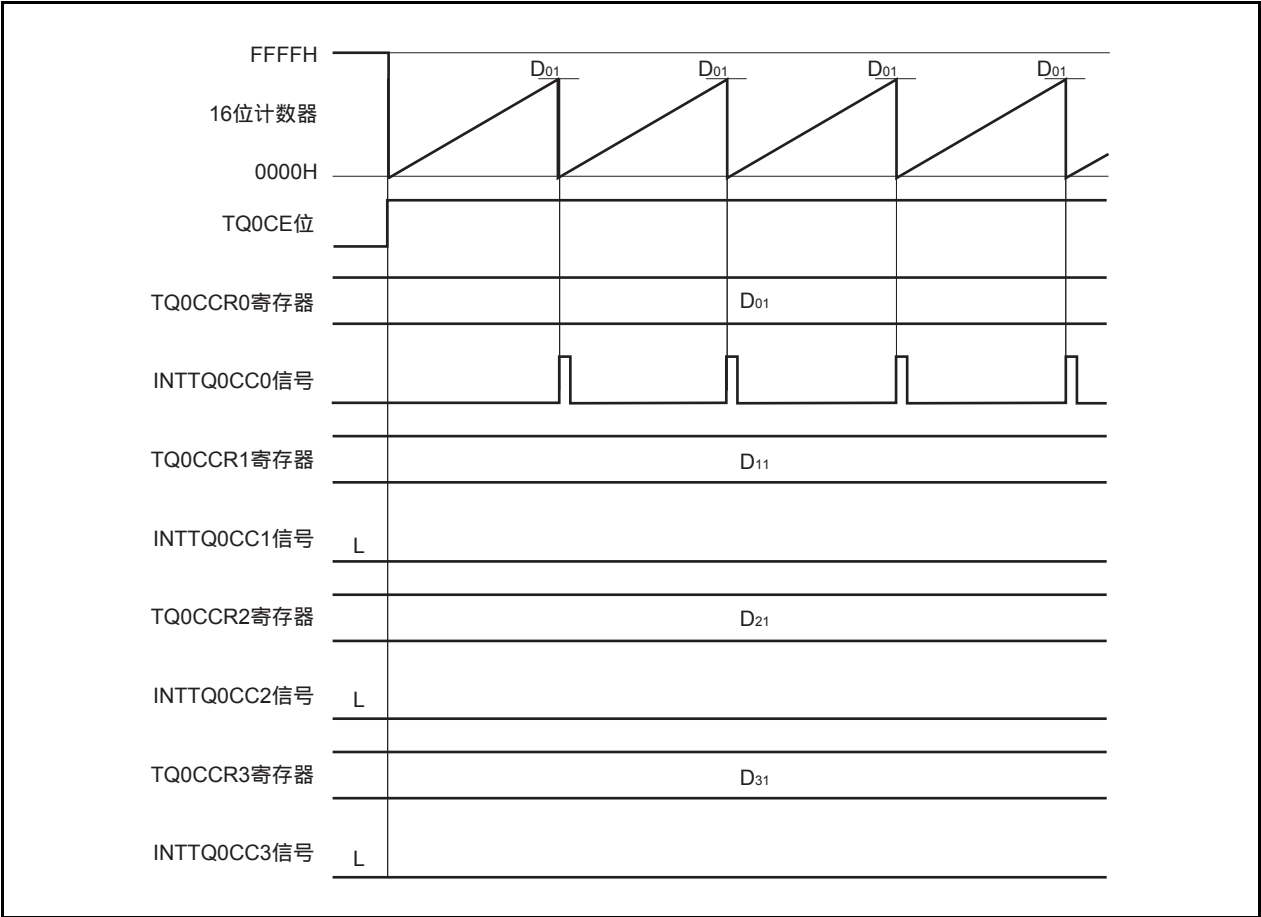
图 7-14. D₀₁ ≥ D_{k1} 时的时序图



如果 TQ0CCRk 寄存器的值大于 TQ0CCR0 寄存器的值，那么 16 位计数器的计数值不会与 TQ0CCRk 寄存器中的值相匹配。因此，不会产生 INTTQ0CCk 信号。

备注 k = 1 至 3

图 7-15. D₀₁ < D_{k1} 时的时序图



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

在外部触发脉冲输出模式中，TQ0CTL0.TQ0CE 位被置 1 后，16 位定时器/事件计数器 Q 进入等待触发信号的状态。当检测到一个外部触发输入信号的有效沿时，16 位定时器/事件计数器 Q 开始计数，并由 TOQ01 至 TOQ03 引脚输出 PWM 波形。

也可以使用软件触发代替外部触发来开始脉冲的输出。当使用软件触发时，还可由 TOQ00 引脚输出一个以 PWM 波形周期的 2 倍为周期的方波。

图 7-16. 外部触发脉冲输出模式的配置

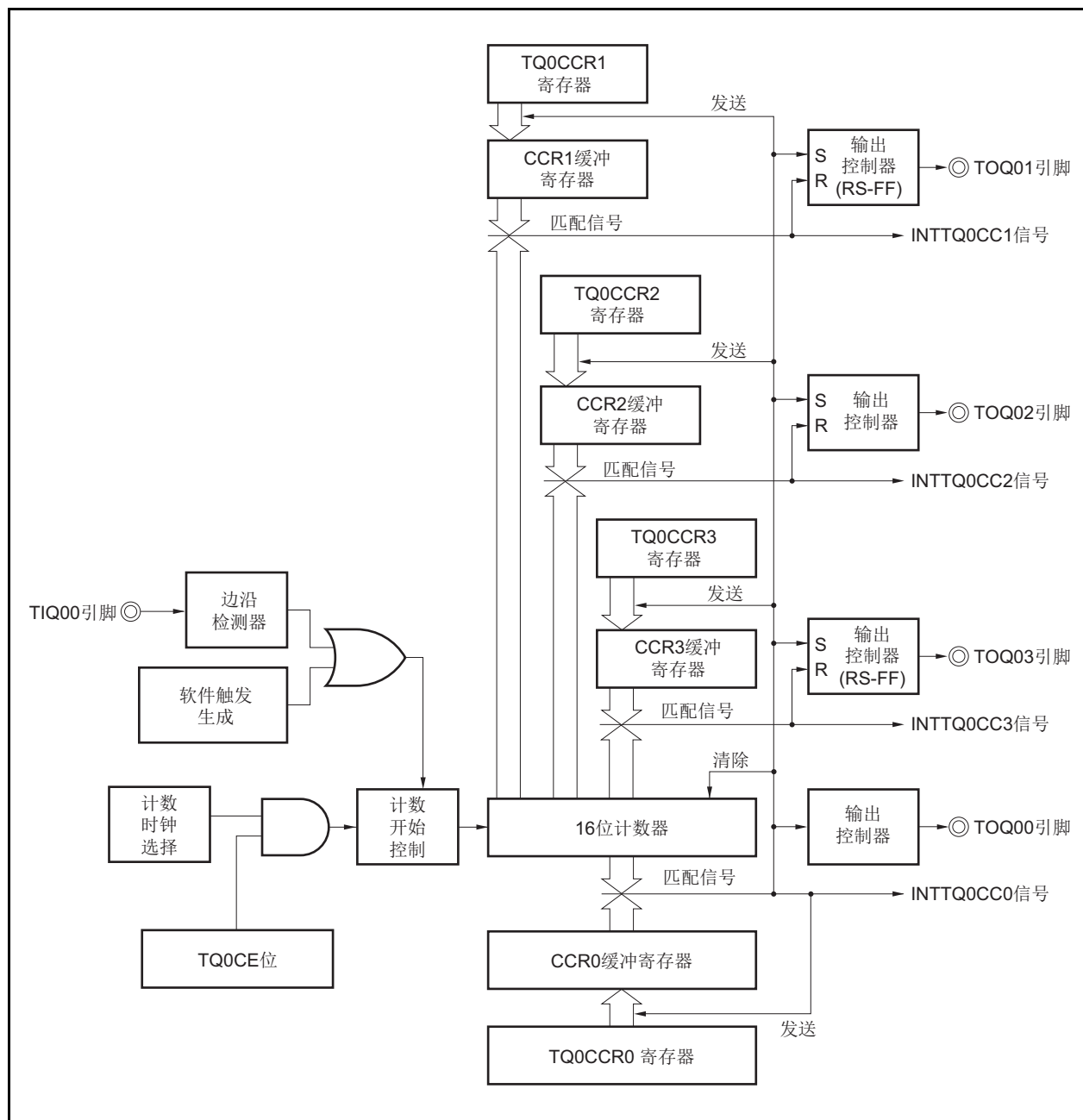
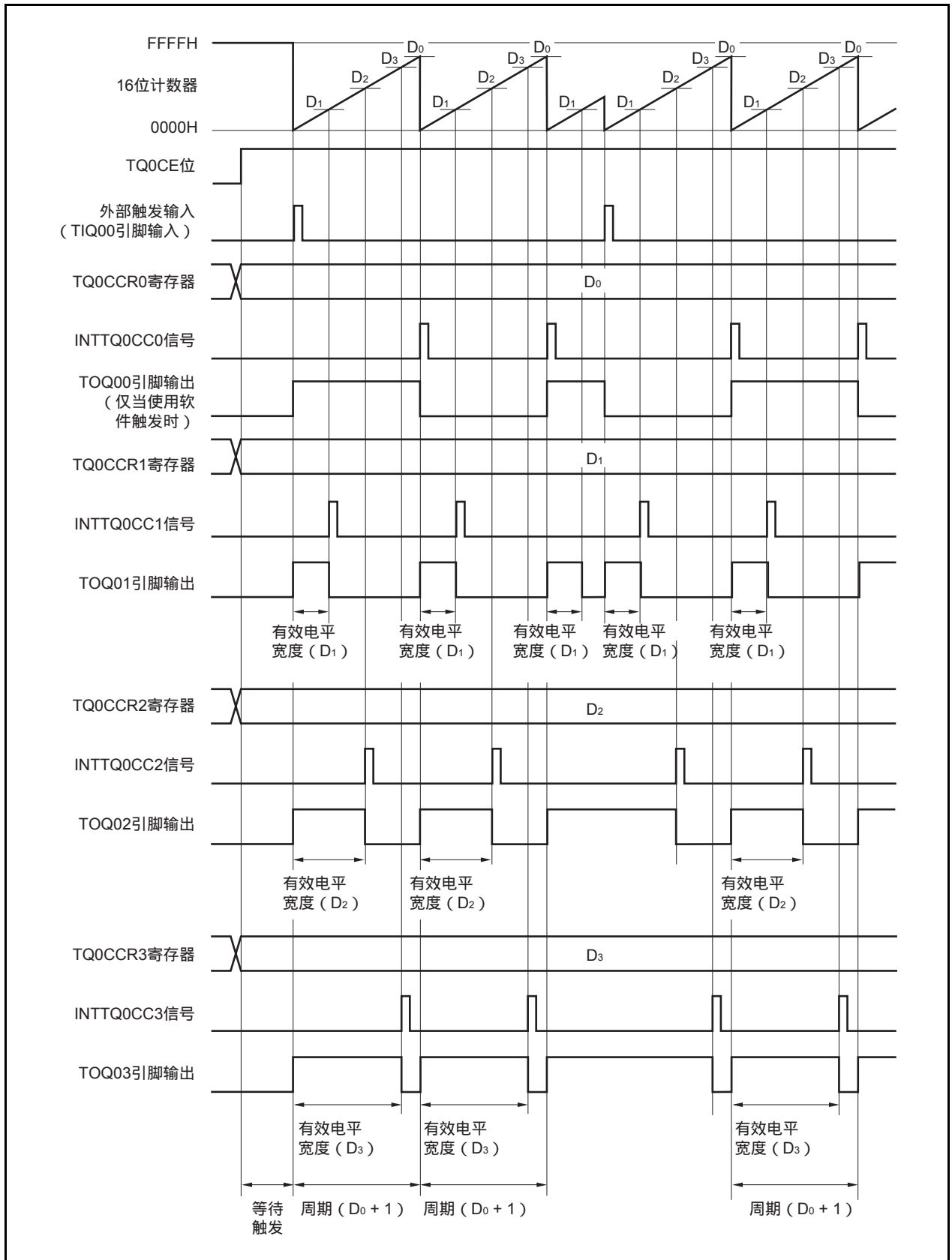


图 7-17. 外部触发脉冲输出模式的基本时序



TQ0CE 位被置 1 后，16 位定时器/事件计数器 Q 进入等待触发信号的状态。触发产生后，16 位计数器从 FFFFH 清零为 0000H，同时开始计数并从 TOQ0k 引脚输出一个 PWM 波形。如果在计数器计数过程中，再次检测到触发信号，那么计数器将被清为 0000H 并重新开始计数。（TOQ00 引脚的输出电平反转。TOQ0k 在触发产生时输出高电平（无论当前状态为高/低）。）

PWM 波形的有效电平宽度，周期以及占空比系数可由以下的算式得出。

有效电平宽度 = (TQ0CCRk 寄存器的设置值) × 计数时钟周期
周期 = (TQ0CCR0 寄存器的值+ 1) × 计数时钟周期
占空比系数 = (TQ0CCRk 寄存器的值) / (TQ0CCR0 寄存器的值+ 1)

比较匹配中断请求信号 INTTQ0CC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器中的值相匹配后产生。，同时 16 位计数器也会被清零。比较匹配中断请求信号 INTTQ0CCk 会在 16 位计数器的计数值与 CCRk 缓冲寄存器中的值相匹配时产生。

当 16 位计数器的计数值与 CCRm 缓冲寄存器中的值相匹配时，TQ0CCRm 寄存器中设置的值会被传送至 CCRm 缓冲寄存器，同时 16 位计数器被清为 0000H。

触发源可以从外部触发输入信号的有效沿和设置软件触发（TQ0CTL1.TQ0EST 位 = 1）的方式中进行选择。

备注 k = 1 至 3， m = 0 至 3

图 7-18. 外部触发脉冲输出模式的寄存器设置 (1/3)

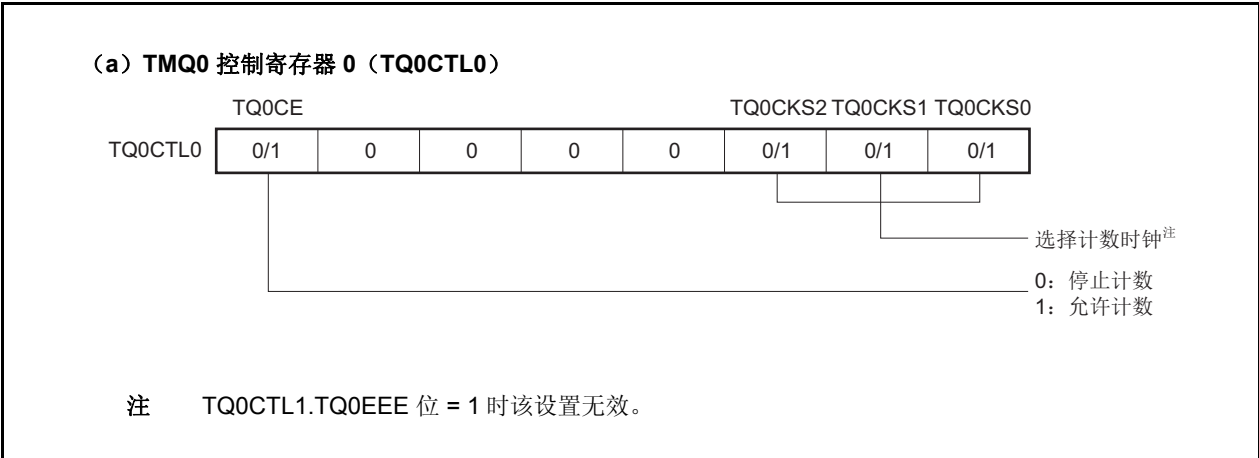
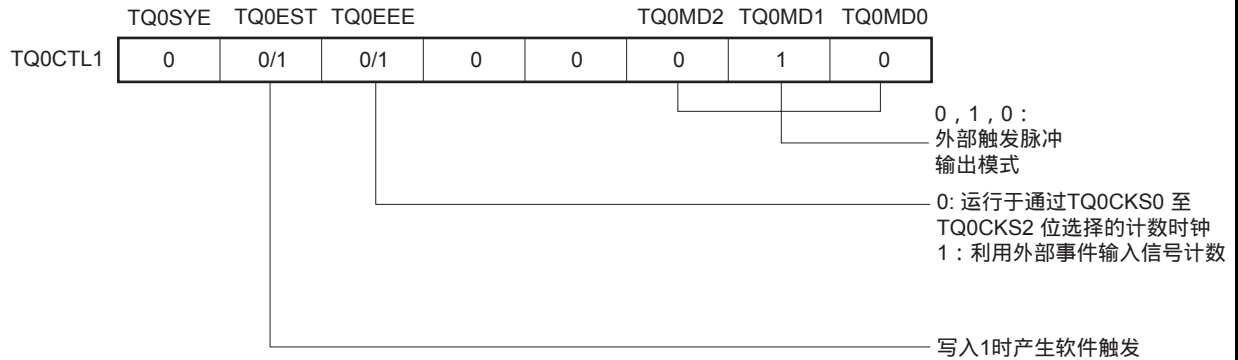
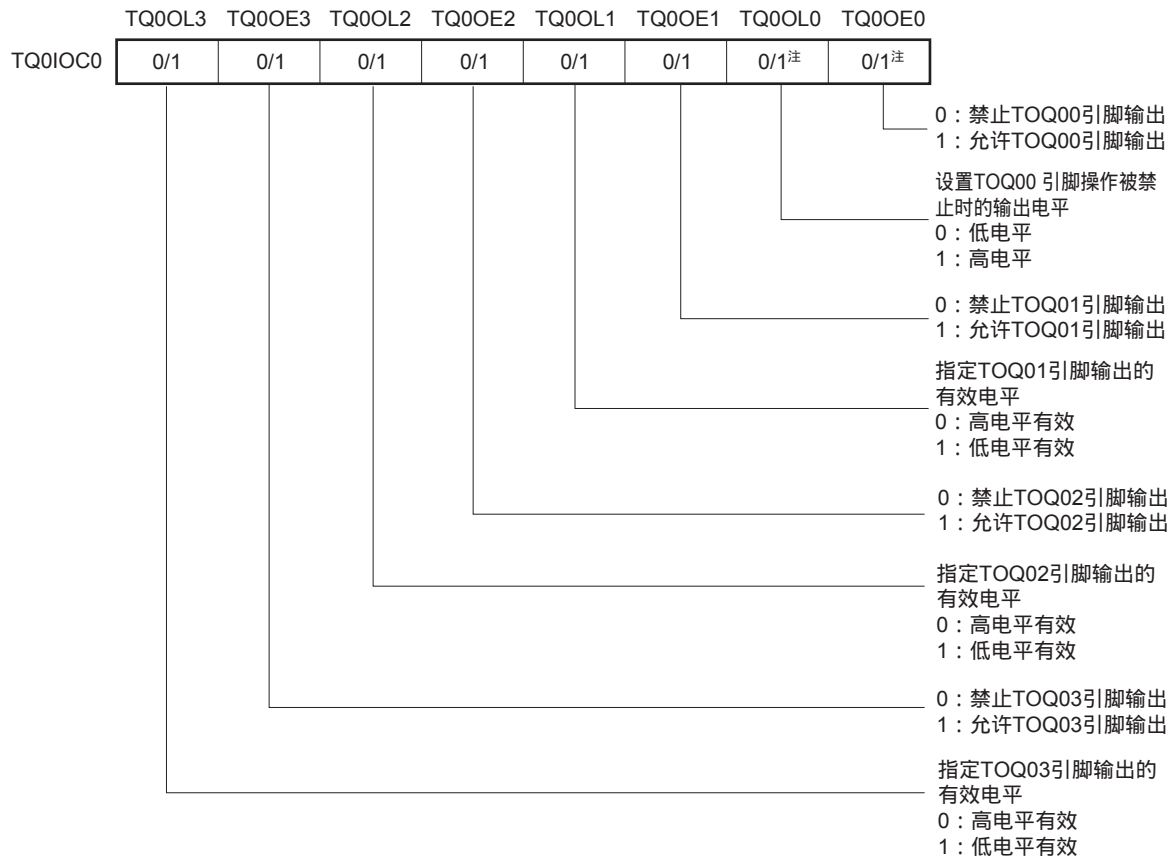


图 7-18. 外部触发脉冲输出模式的寄存器设置 (2/3)

(b) TMQ0 控制寄存器 1 (TQ0CTL1)



(c) TMQ0 I/O 控制寄存器 0 (TQ0IOC0)



· 当TQ0OLk位 = 0



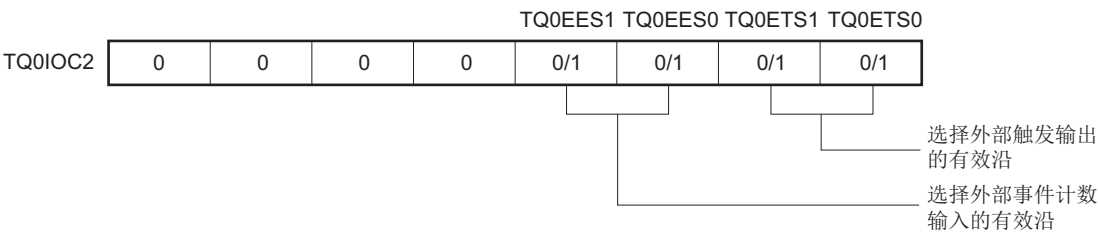
· 当TQ0OLk位 = 1



注 当 TOQ00 引脚不用于外部触发脉冲输出模式时将此位清零。

图 7-18. 外部触发脉冲输出模式的寄存器设置 (3/3)

(d) TMQ0 I/O 控制寄存器 2 (TQ0IOC2)



(e) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

可以通过读取 TQ0CNT 寄存器来读取 16 位计数器的计数值。

(f) TMQ0 捕捉/比较寄存器 0 to 3 (TQ0CCR0 to TQ0CCR3)

如果将 TQ0CCR0 寄存器的值设置为 D_0 ，将 TQ0CCR1 寄存器的值设置为 D_1 ，将 TQ0CCR2 寄存器的值设置为 D_2 ，将 TQ0CCR3 寄存器的值设置为 D_3 ，那么 PWM 波形的周期和有效电平如下所示。

周期 = $(D_0 + 1) \times$ 计数时钟周期

TOQ01 引脚 PWM 波形有效电平宽度 = $D_1 \times$ 计数时钟周期

TOQ02 引脚 PWM 波形有效电平宽度 = $D_2 \times$ 计数时钟周期

TOQ03 引脚 PWM 波形有效电平宽度 = $D_3 \times$ 计数时钟周期

- 备注**
1. 外部触发脉冲输出模式下，并未使用 TMQ0 I/O 控制寄存器 1 (TQ0IOC1) 和 TMQ0 选项寄存器 0 (TQ0OPT0)。
 2. 通过写入 TMQ0 捕捉/比较寄存器 1 (TQ0CCR1)，来使更新 TMQ0 捕捉/比较寄存器 2 (TQ0CCR2) 和 TMQ0 捕捉/比较寄存器 3 (TQ0CCR3) 的操作有效。

(1) 外部触发脉冲输出模式的软件操作流程

图 7-19. 外部触发脉冲输出模式的软件处理流程 (1/2)

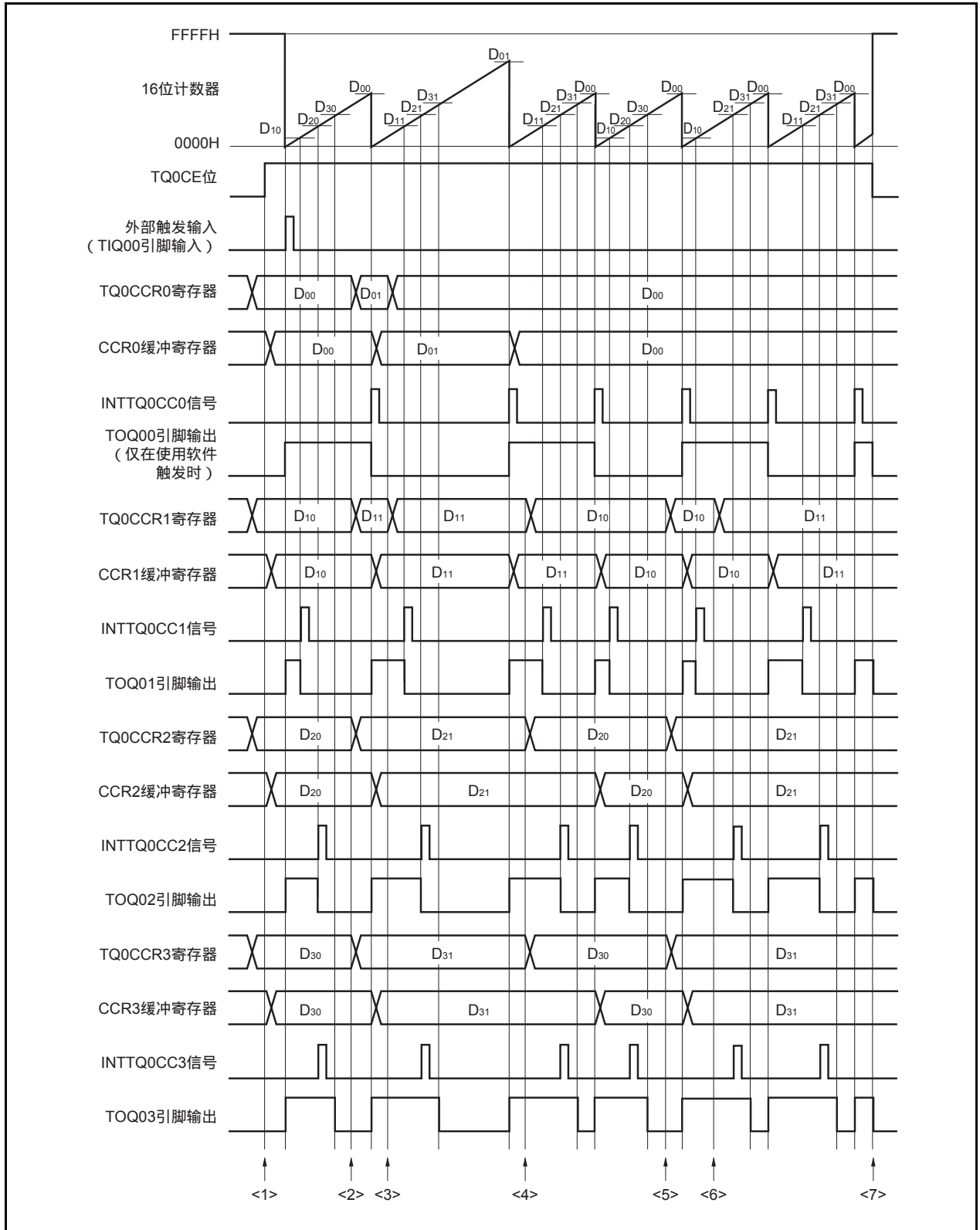
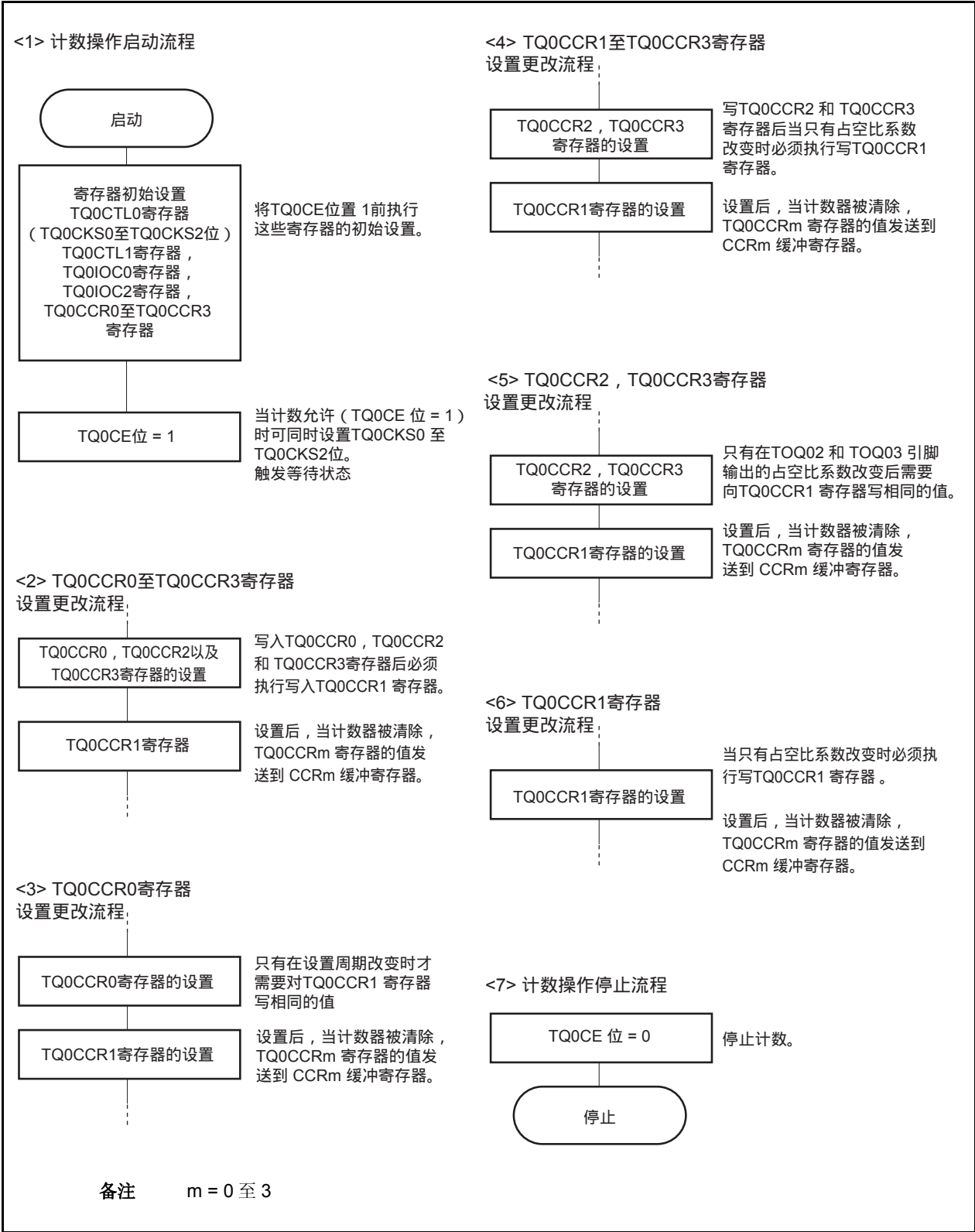


图 7-19. 外部触发脉冲输出模式的软件处理流程 (2/2)

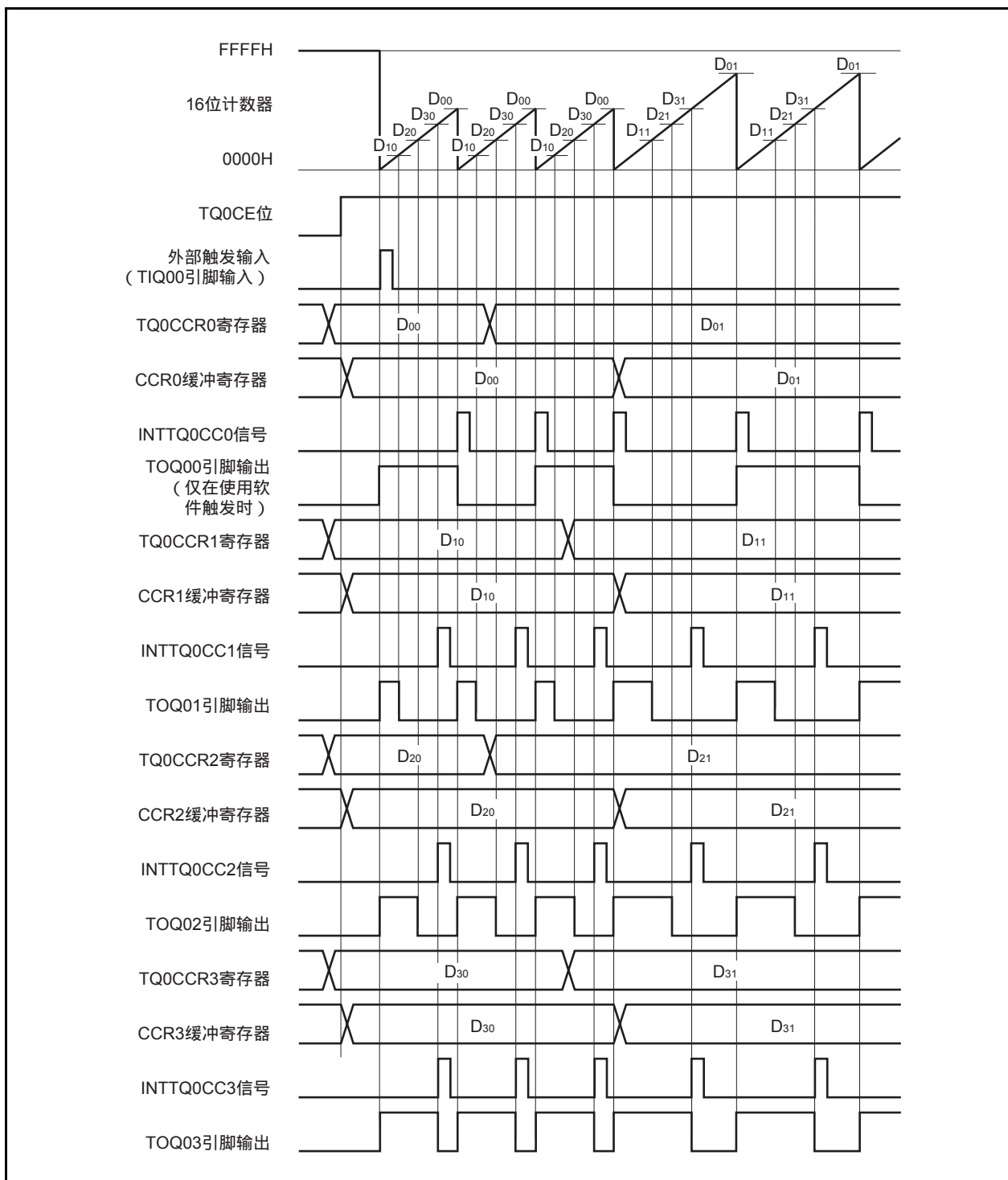


(2) 外部触发脉冲输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

计数器计数过程中，若改变 PWM 波形，则要在最后进行对 TQ0CCR1 寄存器的写入操作。

对 TQ0CCR1 寄存器进行写入操作之后，若需再次更改 TQ0CCRk 寄存器的值，须等待下一个 INTTQ0CC0 信号被检测到后，再进行相关操作。



若要将 TQ0CCRm 寄存器中设置的数据传送至 CCRm 缓冲寄存器，必须对 TQ0CCR1 寄存器进行写入操作。

同时改变 PWM 波形的周期和有效电平时，先将周期值写入 TQ0CCR0 寄存器，再将有效电平宽度值写入 TQ0CCR2 和 TQ0CCR3 寄存器，最后将有效电平写入 TQ0CCR1 寄存器。

只改变 PWM 波形的周期时，先将周期值写入 TQ0CCR0 寄存器，再向 TQ0CCR1 寄存器写入与原来相同的值。

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

若只改变 TOQ01 引脚输出的 PWM 波形的有效电平宽度（占空比系数），只需要设置 TQ0CCR1 寄存器。

若只改变 TOQ02 和 TOQ03 引脚输出的 PWM 波形的有效电平宽度（占空比系数），首先需要向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平宽度，然后向 TQ0CCR1 寄存器写入相同的值。

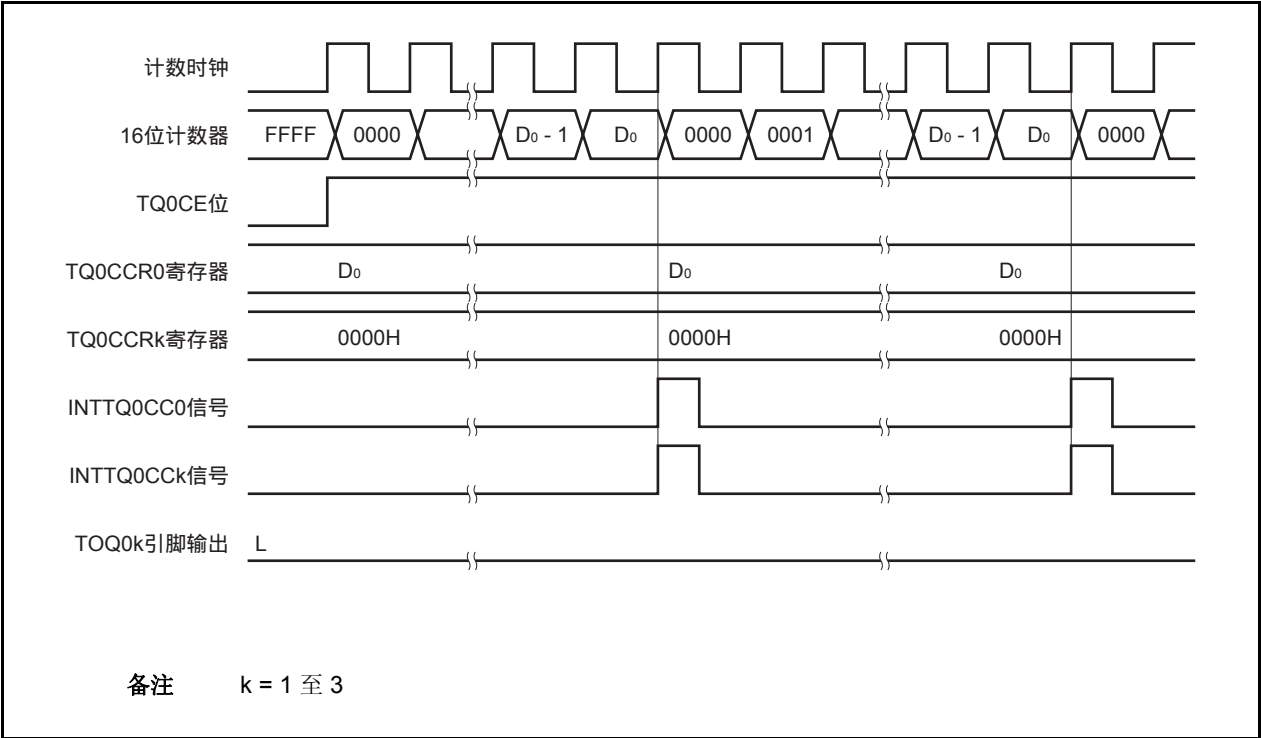
数据被写入 TQ0CCR1 寄存器之后，当 16 位计数器被清零时，TQ0CCRm 寄存器中的数值才会被传送至 CCRm 缓冲寄存器中。此时 CCRm 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

向 TQ0CCR1 寄存器的写入操作完成后，若需再次改写 TQ0CCR0 至 TQ0CCR3 寄存器的值，则要在产生 INTTQ0CC0 信号之后再进行改写。否则，由于数据从 TQ0CCRm 寄存器传送至 CCRm 缓冲寄存器的时序与 TQ0CCRm 寄存器的写入时序相冲突，CCRM 缓冲寄存器中所设置的数值可能无法确定。

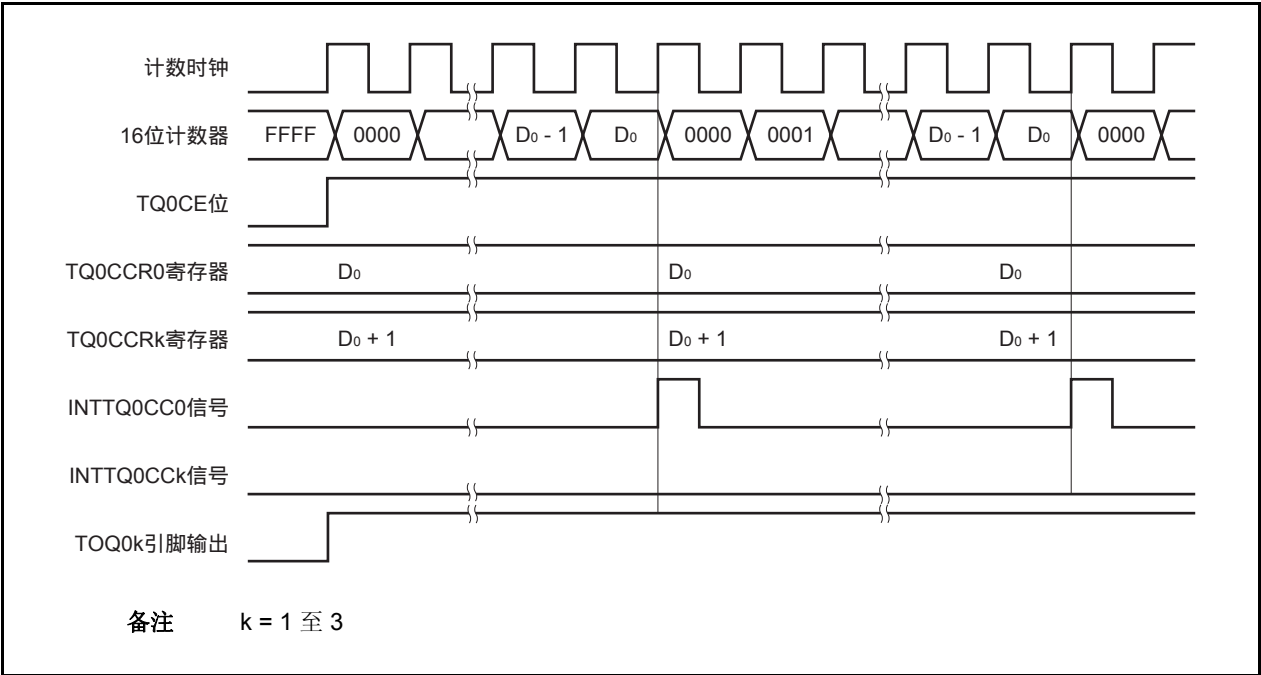
备注 m = 0 至 3

(b) PWM 波形的 0%/100% 输出

若要输出 0% 的波形，则需将 TQ0CCRk 寄存器设置为 0000H。如果 TQ0CCR0 寄存器被设置为 FFFFH，那么 INTTQ0CCK 信号将周期性产生。

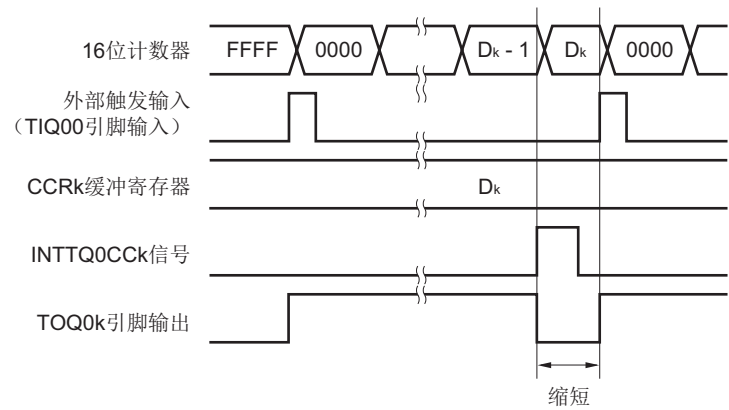


若要输出 100% 波形，则需将 TQ0CCRk 寄存器设置为 (TQ0CCR0 寄存器值 + 1)。TQ0CCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。



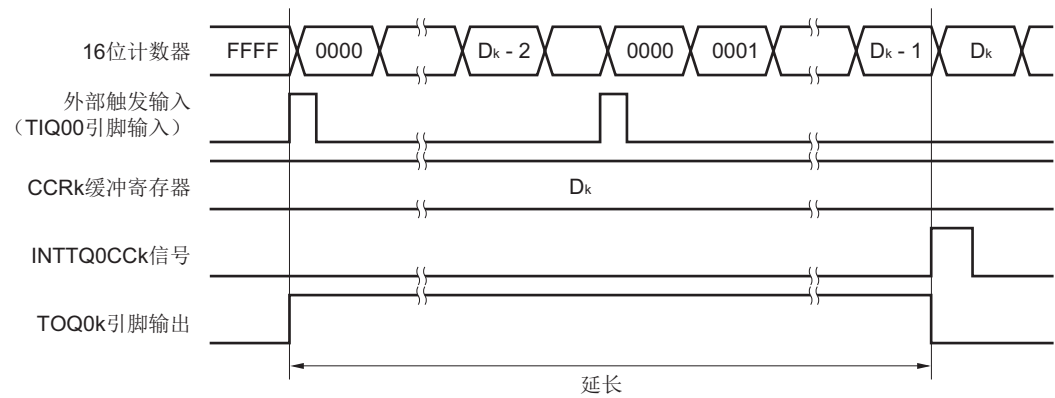
(c) 触发检测和 CCRk 缓冲寄存器匹配之间的冲突

如果触发信号紧随 INTTQ0CCk 中断信号的产生被检测，那么 16 位计数器将被立即清零并重新开始计数，TOQ0k 引脚输出也会改变为有效电平。这种情况下，PWM 波形的非有效电平时间就会被缩短。



备注 $k = 1$ 至 3

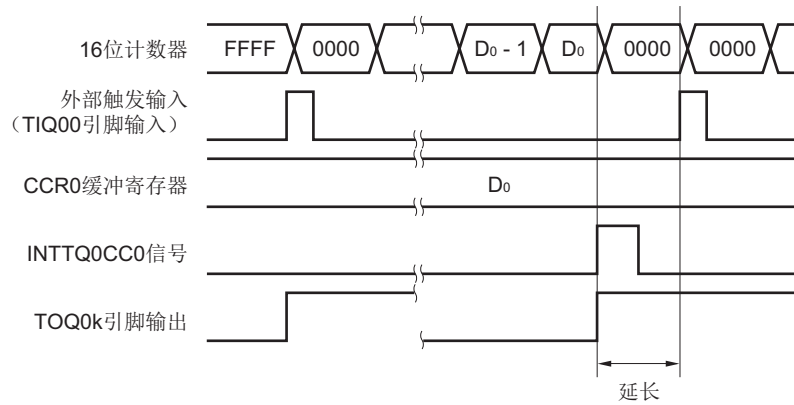
如果在 INTTQ0CCk 信号产生之前检测到触发信号，那么 INTTQ0CCk 信号将不会产生，此时，16 位计数器被清零并重新开始计数。TOQ0k 引脚的输出也会继续保持为有效电平。这种情况下，PWM 波形的有效电平时间就会被延长。



备注 $k = 1$ 至 3

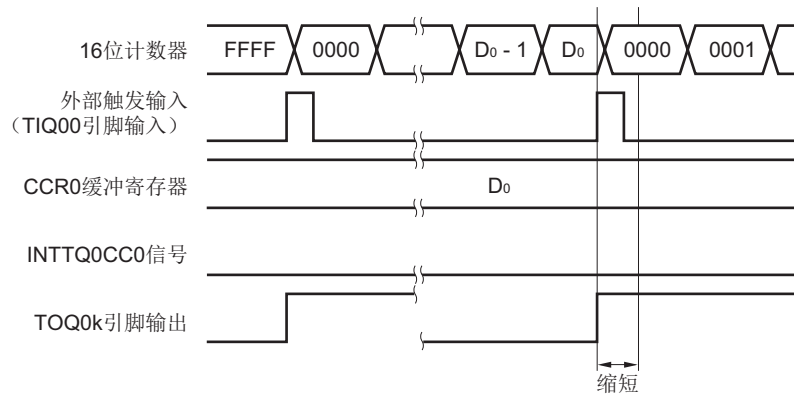
(d) 触发检测和 CCR0 缓冲寄存器匹配之间的冲突

如果触发信号紧随 INTTQ0CC0 中断信号的产生被检测，那么 16 位计数器将被清零并重新开始计数。因此，TOQ0k 引脚输出的有效电平时间就会被延长（从 INTTQ0CC0 信号产生到触发信号被检测的时间）。



备注 k = 1 至 3

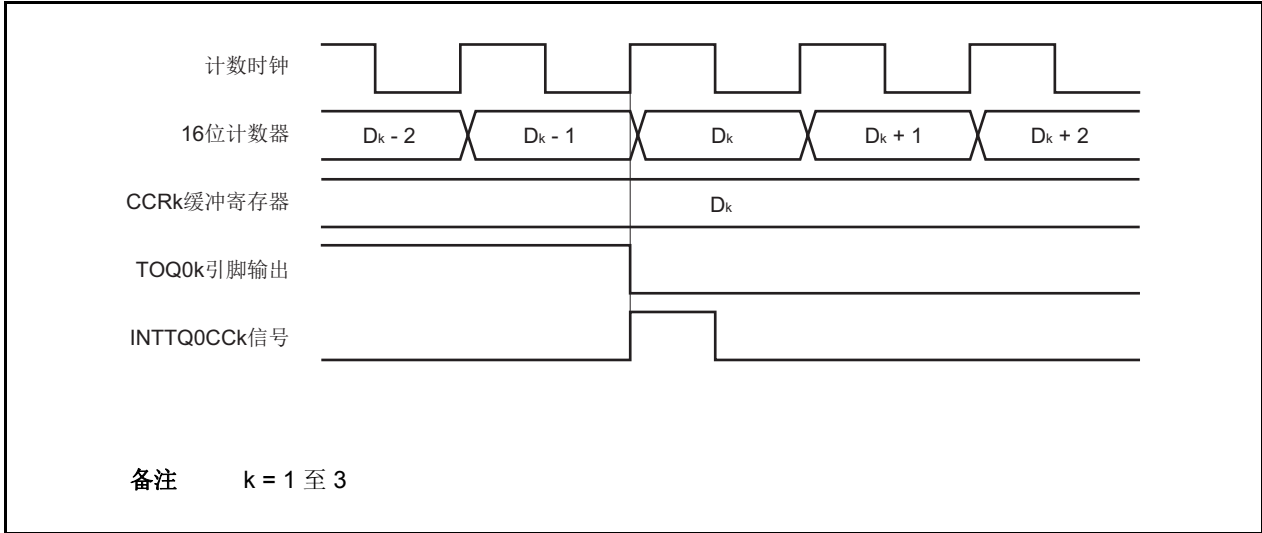
如果在 INTTQ0CC0 信号产生之前检测到触发信号，那么 INTTQ0CC0 信号将不会产生。16 位计数器将被清零并重新开始计数，TOQ0k 引脚输出也会改变为有效电平。这种情况下，PWM 波形的非有效电平时间就会被缩短。



备注 k = 1 至 3

(e) 比较匹配中断请求信号的产生时序 (INTTQ0CCK)

外部触发脉冲输出模式下，INTTQ0CCK 信号的产生时序与其它模式下 INTTQ0CCK 信号的产生时序有所不同，这里，INTTQ0CCK 信号将在 16 位计数器的计数值与 CCRk 缓冲寄存器的值相匹配时产生。



通常情况下，INTTQ0CCK 信号会在 16 位计数器计数值与 CCRk 缓冲寄存器的值匹配之后的下一个计数时钟同步产生。

但在外部触发脉冲输出模式下，INTTQ0CCK 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOQ0k 引脚输出信号变化的时序而改变的。

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

在单次脉冲输出模式中, TQ0CTL0.TQ0CE 位被置 1 后, 16 位定时器/事件计数器 Q 进入等待触发信号的状态。当检测到一个外部触发输入信号的有效沿时, 16 位定时器/事件计数器 Q 开始计数, 并由 TOQ01 至 TOQ03 引脚输出单脉冲波形。

也可以使用软件触发代替外部触发来开始脉冲的输出。当使用软件触发时, TOQ00 引脚会在 16 位计数器计数时输出有效电平, 在计数器停止计数时 (等待触发时) 输出非有效电平。

图 7-20. 单脉冲输出模式的配置图

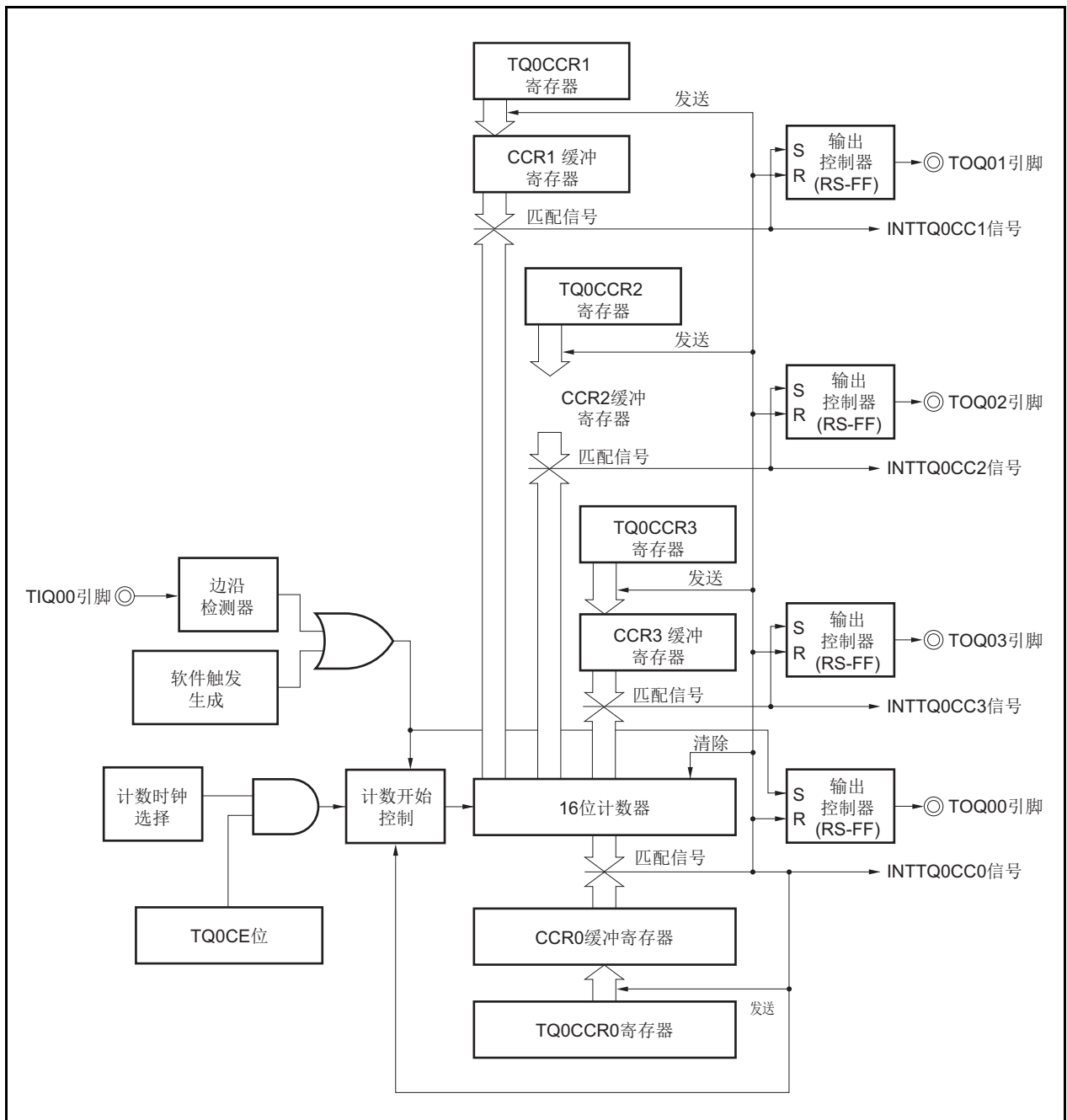
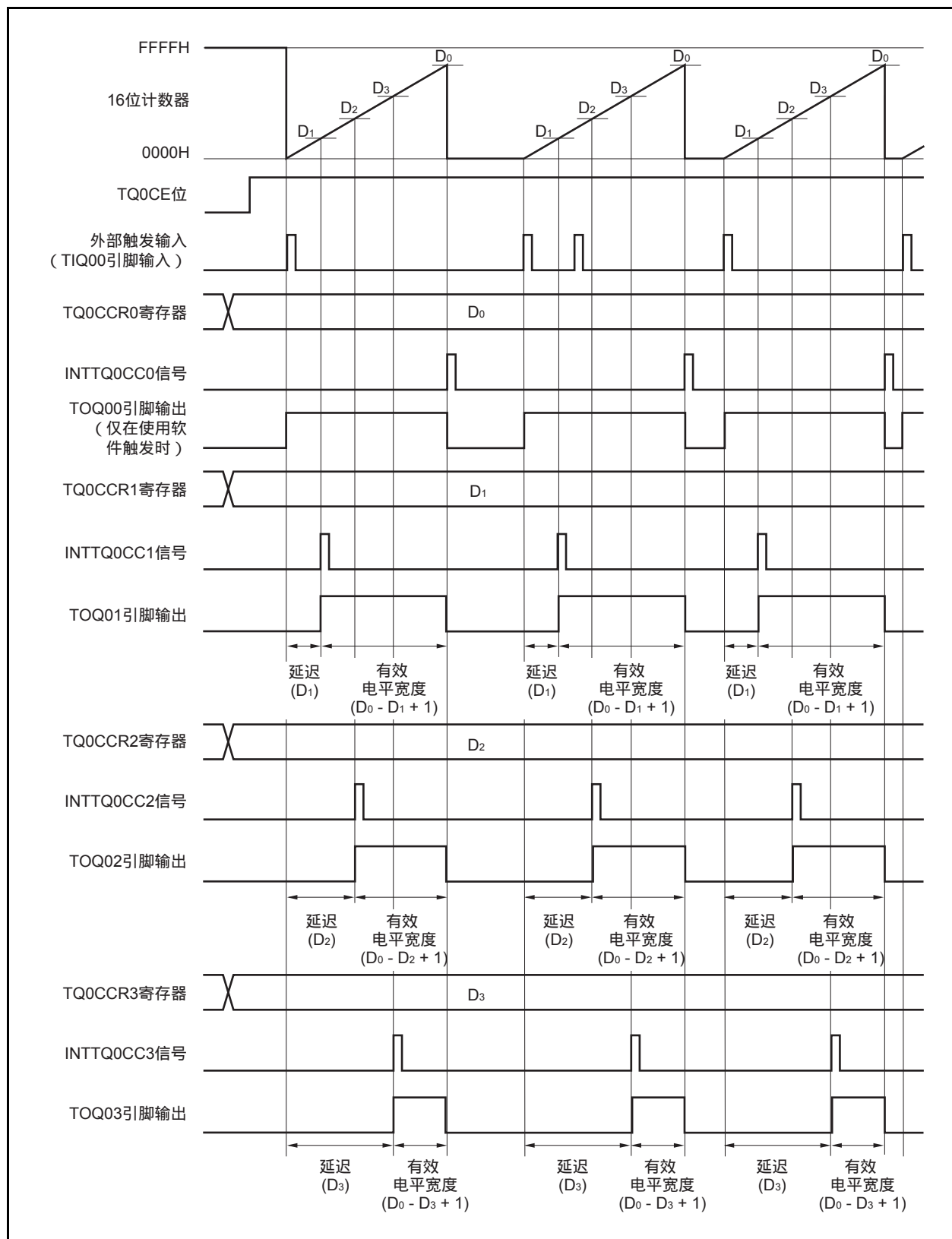


图 7-21. 单脉冲输出模式的基本时序



TQ0CE 位被置 1 后，16 位定时器/事件计数器 Q 进入等待触发信号的状态。触发被检测后，16 位计数器从 FFFFH 清零为 0000H，同时开始计数并从 TOQ0k 引脚输出单脉冲波形。单脉冲波形输出后，16 位计数器将被重置为 FFFFH 并停止计数，16 位定时器/时间计数器 P 重新进入等待触发信号的状态。如果在单脉冲波形输出过程中再次检测到触发信号，那么这个触发信号将被忽略。

单脉冲波形的输出延迟时间以及有效电平宽度可由以下算式得出。

输出延迟时间 = (TQ0CCRk 寄存器的设置值) × 计数时钟周期

有效电平宽度 = (TQ0CCR0 寄存器的值 - TQ0CCRk 寄存器的值 + 1) × 计数时钟周期

比较匹配中断请求信号 INTTQ0CC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器中的值相匹配时的下一个计数时钟时产生。比较匹配中断请求信号 INTTQ0CCk 会在 16 位计数器的计数值与 CCRk 缓冲寄存器中的值相匹配时产生。

触发源可以从外部触发输入信号的有效沿和设置软件触发 (TQ0CTL1.TQ0EST 位 = 1) 置 1 的方式中进行选择。

备注 k = 1 至 3

图 7-22. 单脉冲输出模式的寄存器设置 (1/3)

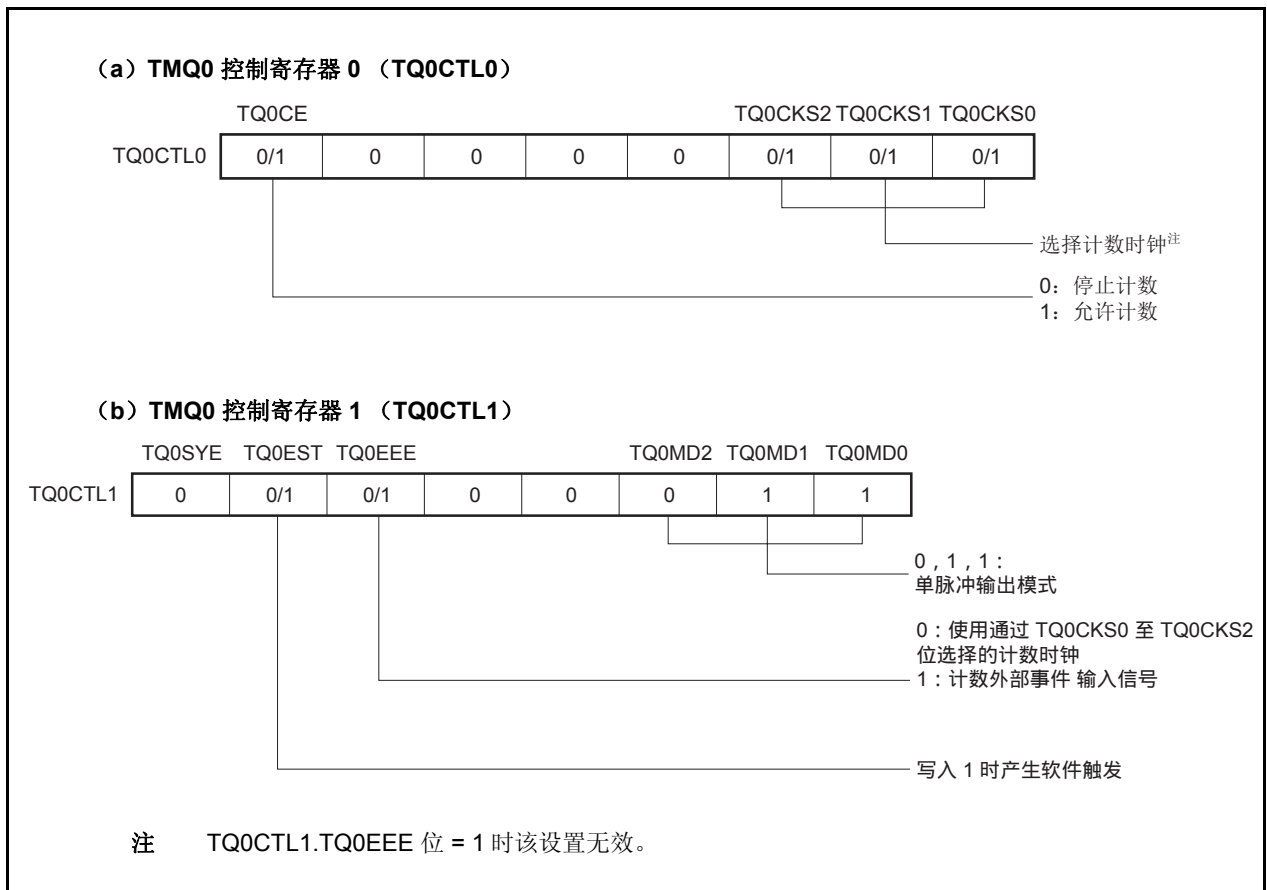
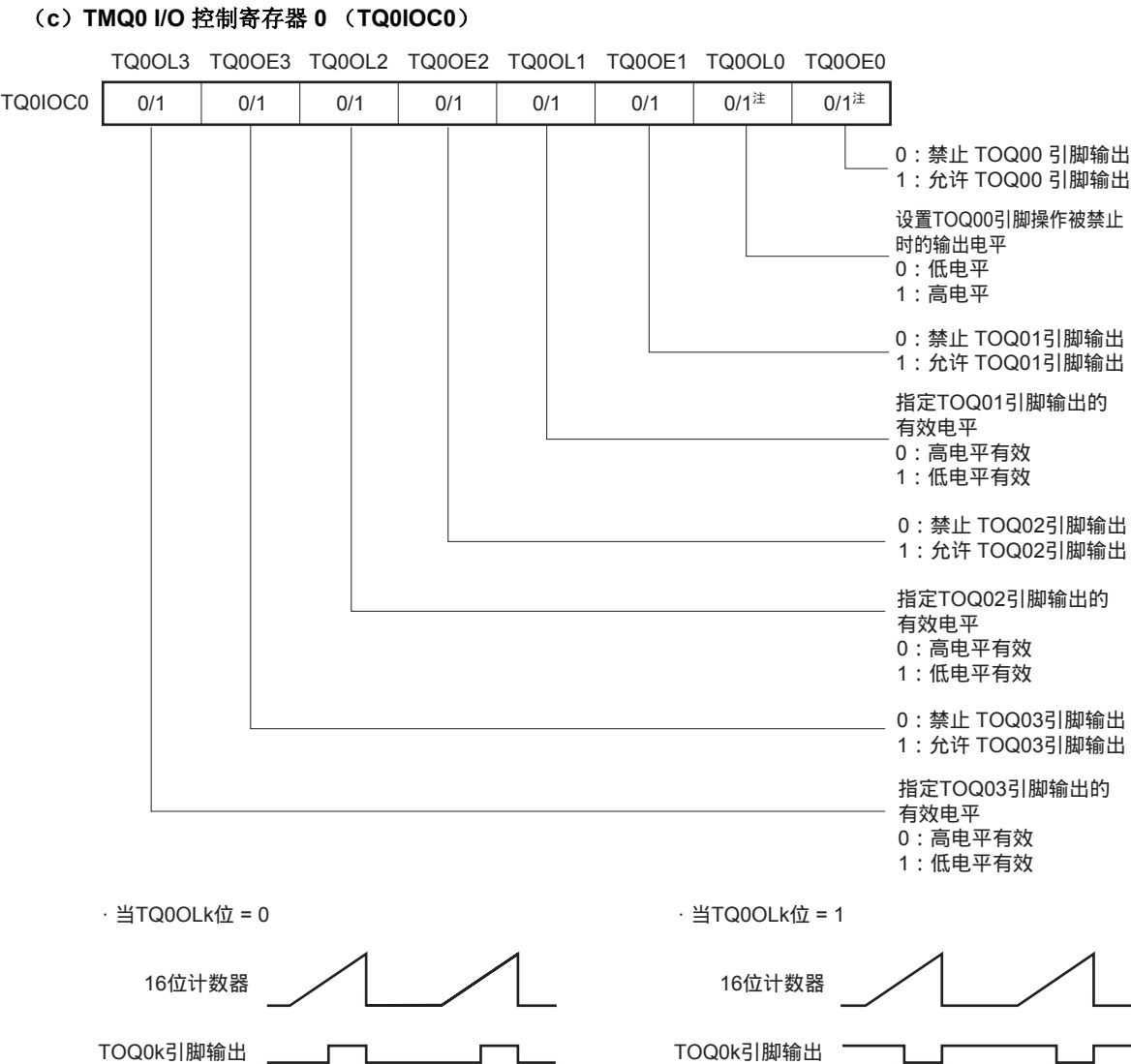
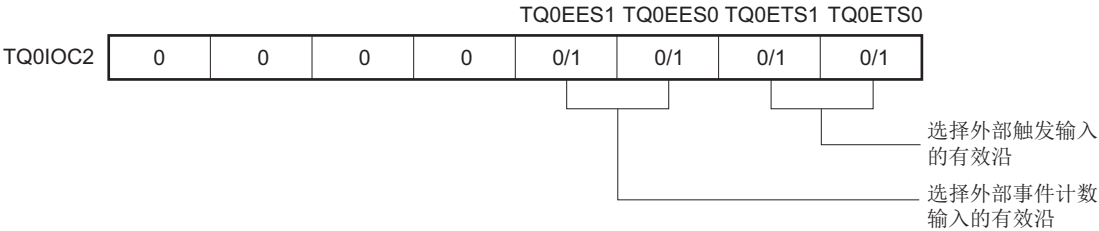


图 7-22. 单脉冲输出模式的寄存器设置 (2/3)



(d) **TMQ0 I/O 控制寄存器 2 (TQ0IOC2)**



注 当单脉冲输出模式下不使用 TOQ00 引脚时，要将该位清零。

图 7-22. 单脉冲输出模式的寄存器设置 (3/3)

(e) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

可以通过读取 TQ0CNT 寄存器来读取 16 位计数器的计数值。

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

如果将 TQ0CCR0 寄存器的值设置为 D0，将 TQ0CCRk 寄存器的值设置为 Dk，那么单脉冲波形的有效电平宽度和输出延迟时间如下所示。

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

输出延迟时间 = $(D_k) \times \text{计数时钟周期}$

注意事项 如果 TQ0CCRk 寄存器的设定值大于 TQ0CCR0 寄存器中的设定值，即使在单脉冲输出模式下也不输出单脉冲。

备注

1. 单脉冲输出模式下，并未使用 TMQ0 I/O 控制寄存器 1 (TQ0IOC1) 和 TMQ0 选项寄存器 0 (TQ0OPT0)。
2. k = 1 至 3

<R>

(1) 单脉冲输出模式的操作流程

图 7-23. 单脉冲输出模式的软件操作流程 (1/2)

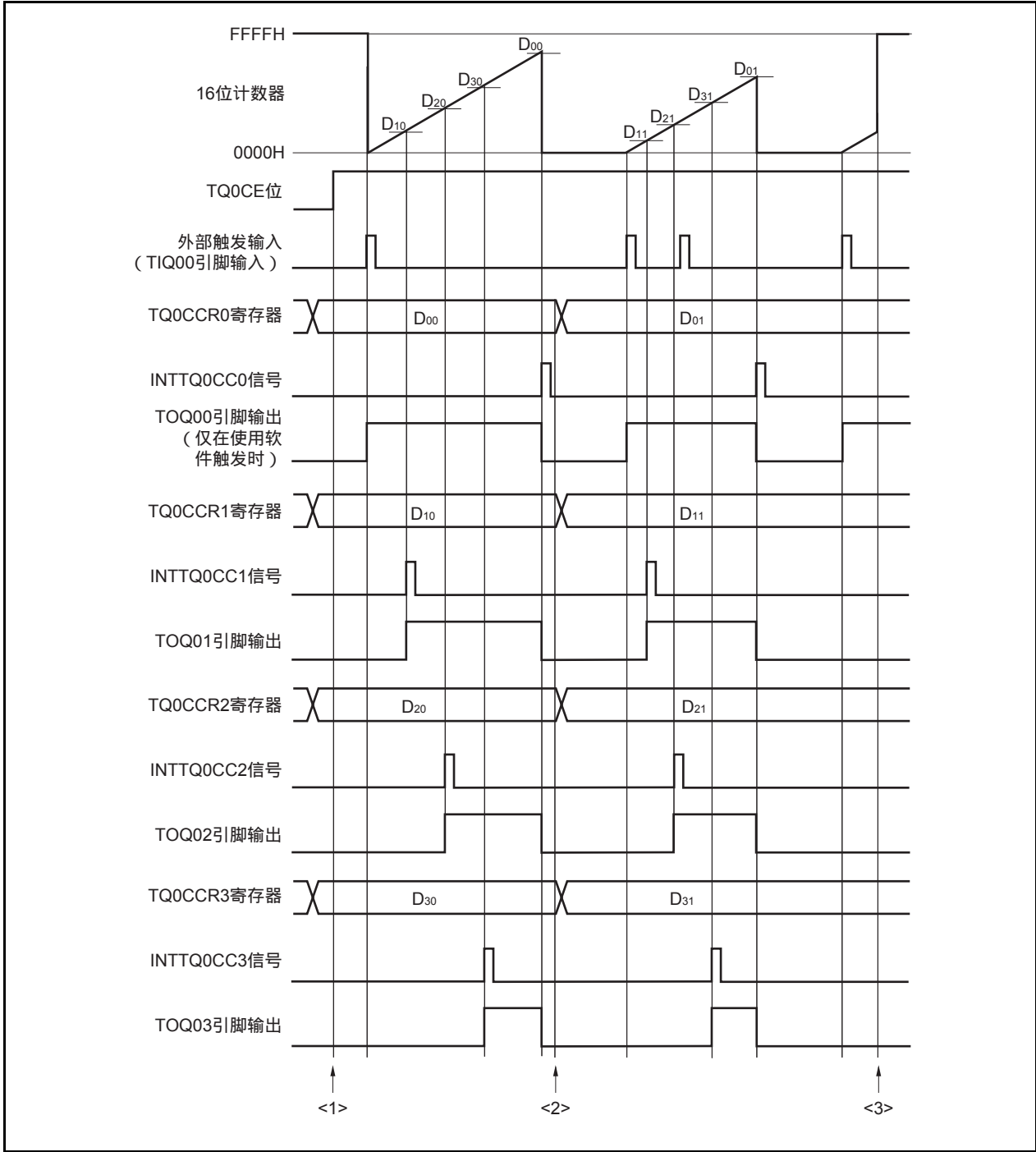
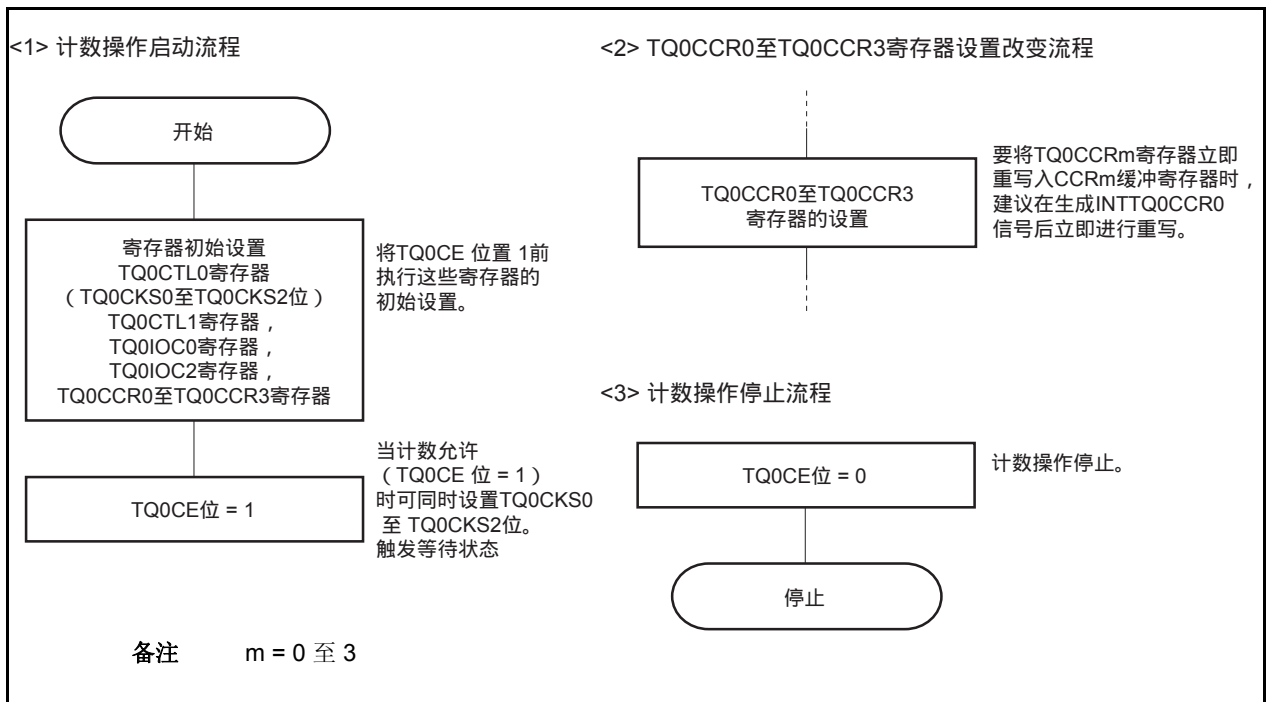


图 7-23. 单脉冲输出模式的软件操作流程 (2/2)

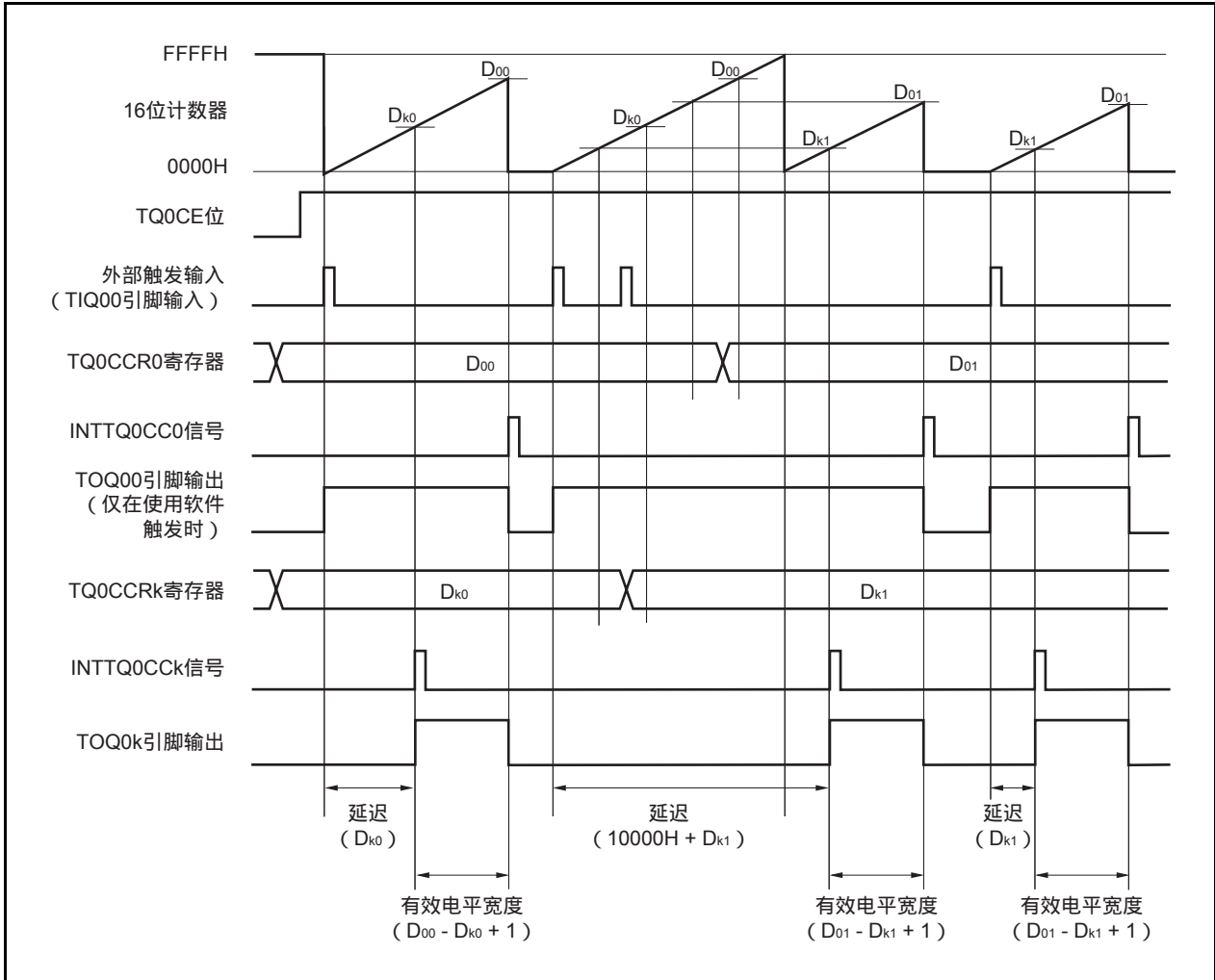


(2) 单脉冲输出模式的操作时序

(a) 改变 TQ0CCRm 寄存器时需要注意的事项

将 TQ0CCRm 寄存器的值减小时，要先停止计数，再对设定值进行改变。

如果在计数过程中将 TQ0CCR0 寄存器的值减小，那么 16 位计数器可能产生溢出。



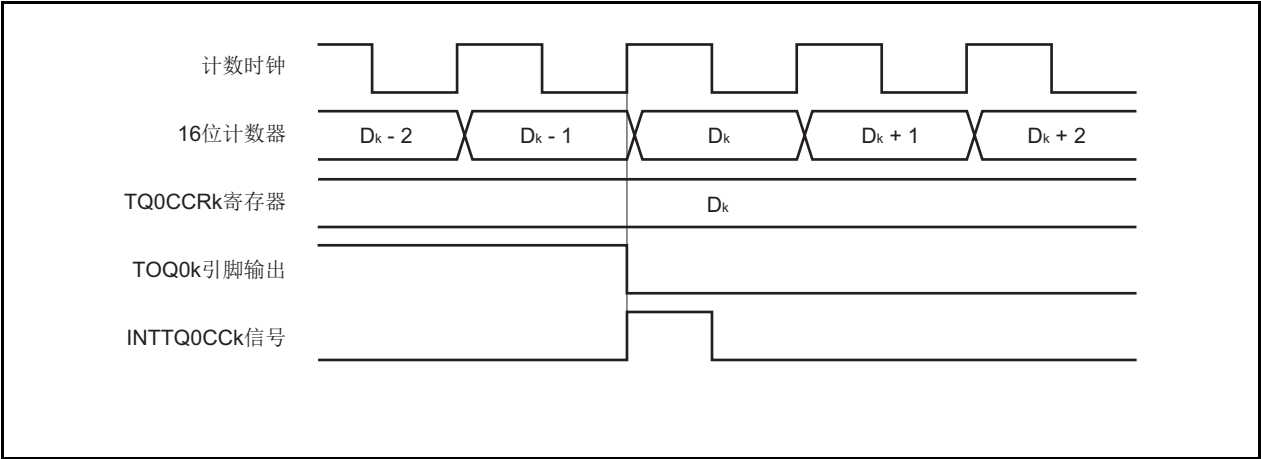
当从 D00 到 D01 和从 Dk0 到 Dk1 分别重写寄存器 TQ0CCR0 和寄存器 TQ0CCRk 时，这里 $D00 > D01$ 和 $Dk0 > Dk1$ ，如果 16 位计数器的计数值大于 Dk1 并且小于 Dk0 时和计数值大于 D01 并且小于 D00 时分别重写寄存器 TQ0CCRk 和 TQ0CCR0，那么每个设定值都会在重写寄存器并与计数值比较时马上表现出来。计数器计数到 FFFFH 后将再次从 0000H 开始计数。当计数值与 Dk1 相符时，计数器产生 INTTQ0CCk 信号并激活 TOQ0k 引脚。当计数值与 D01 相符时，计数器产生 INTTQ0CC0 信号并关闭 TOQ0k 引脚并停止计数。

因此，计数器可以输出一个带一个延迟期或有效高电平期脉冲，这和最初预期的单脉冲的情况不同。

备注 k = 1 至 3

(b) 比较匹配中断请求信号的产生时序 (INTTQ0CCK)

单脉冲输出模式下，INTTQ0CCK 信号的产生时序与其它模式下 INTTQ0CCK 信号的产生时序有所不同。这里，INTTQ0CCK 信号将在 16 位计数器的计数值与 TQ0CCRk 寄存器的值相匹配时产生。



通常情况下，INTTQ0CCK 信号会在 16 位计数器计数值与 TQ0CCRk 寄存器的值匹配之后的下一个计数时钟同步产生。

但在单脉冲输出模式下，INTTQ0CCK 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOQ0k 引脚输出信号变化的时序而改变的。

备注 $k = 1$ 至 3

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

PWM 输出模式下, TQ0CTL0.TQ0CE 位被置 1 后, TOQ01 至 TOQ03 引脚将输出 PWM 波形。

另外, TOQ00 引脚将输出一个以 PWM 波形周期的 2 倍为周期的脉冲信号。

图 7-24. PWM 输出模式的配置图

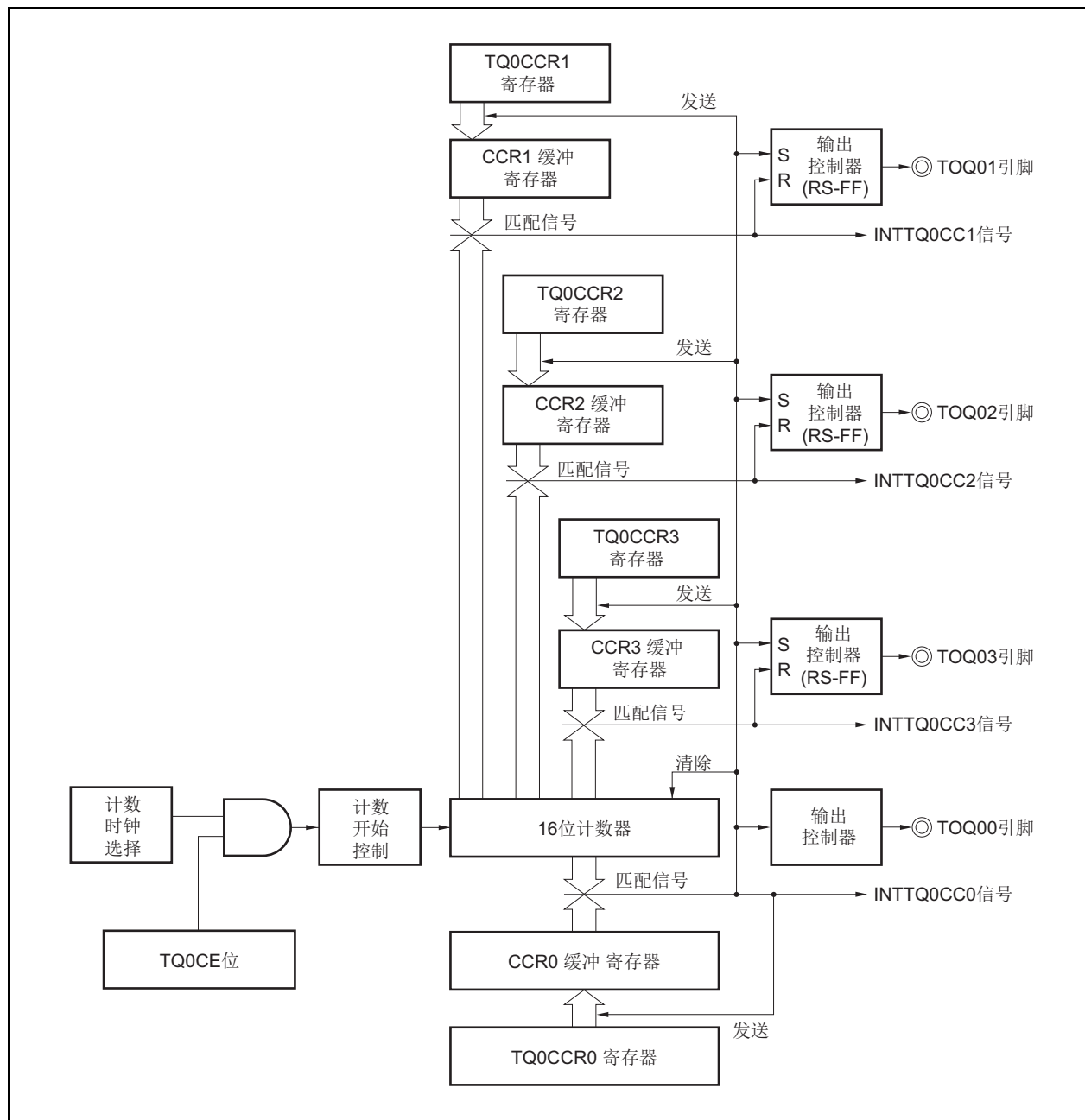
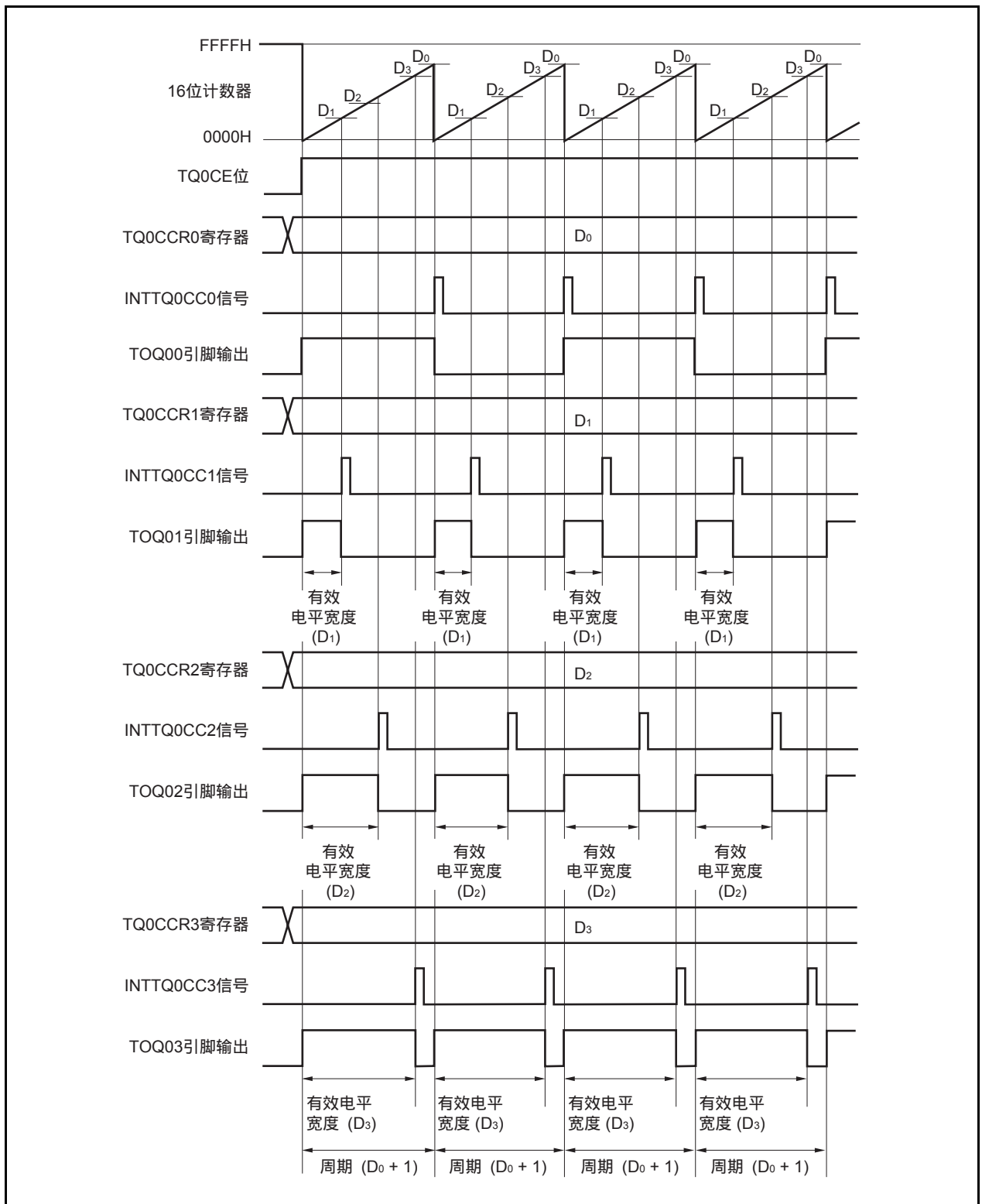


图 7-25. PWM 输出模式的基本时序



TQ0CE 位被置 1 后，16 位计数器从 FFFFH 清零为 0000H 并开始计数，同时从 TOQ0k 引脚输出 PWM 波形。PWM 波形的有效电平宽度，周期以及占空比系数可由以下算式得出。

有效电平宽度 = (TQ0CCRk 寄存器的值) × 计数时钟周期
周期 = (TQ0CCR0 寄存器的值 + 1) × 计数时钟周期
占空比系数 = (TQ0CCRk 寄存器的值) / (TQ0CCR0 寄存器的值 + 1)

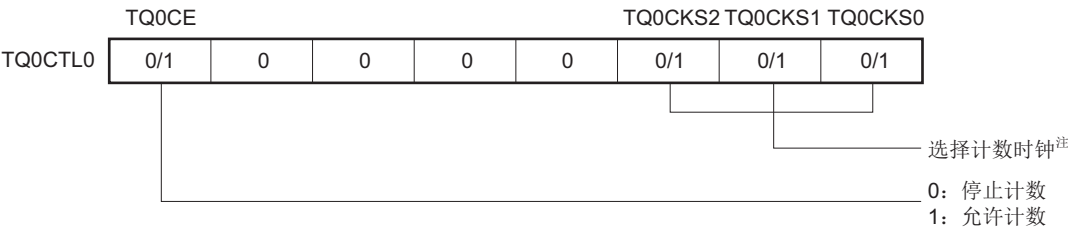
可以在计数器计数过程中，通过改写 TQ0CCRm 寄存器的值来改变 PWM 波形的参数。改写后的数值将在 16 位计数器的计数值与当前 CCR0 缓冲寄存器的值匹配之后生效，同时 16 位计数器也会被清为 0000H。

比较匹配中断请求信号 INTTQ0CC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器的值相匹配后的下一个计数时钟时产生，同时 16 位计数器将被清零。比较匹配中断请求信号 INTTQ0CCk 会在 16 位计数器的计数值与 CCRk 缓冲寄存器的值相匹配时产生。

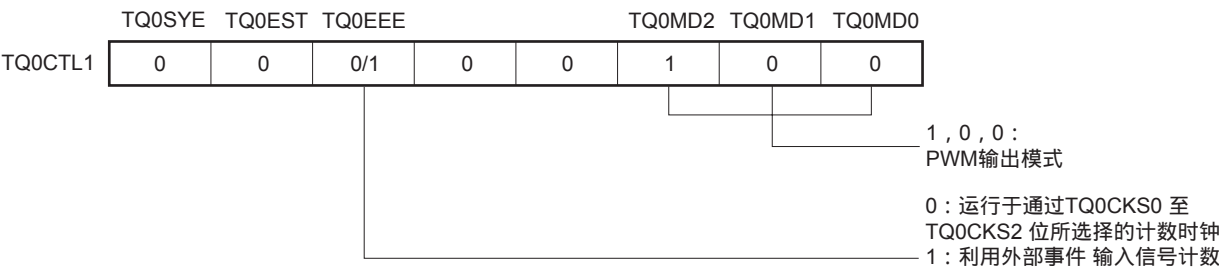
备注 k = 1 至 3， m = 0 至 3

图 7-26. PWM 输出模式的寄存器设置 (1/3)

(a) TMQ0 控制寄存器 0 (TQ0CTL0)



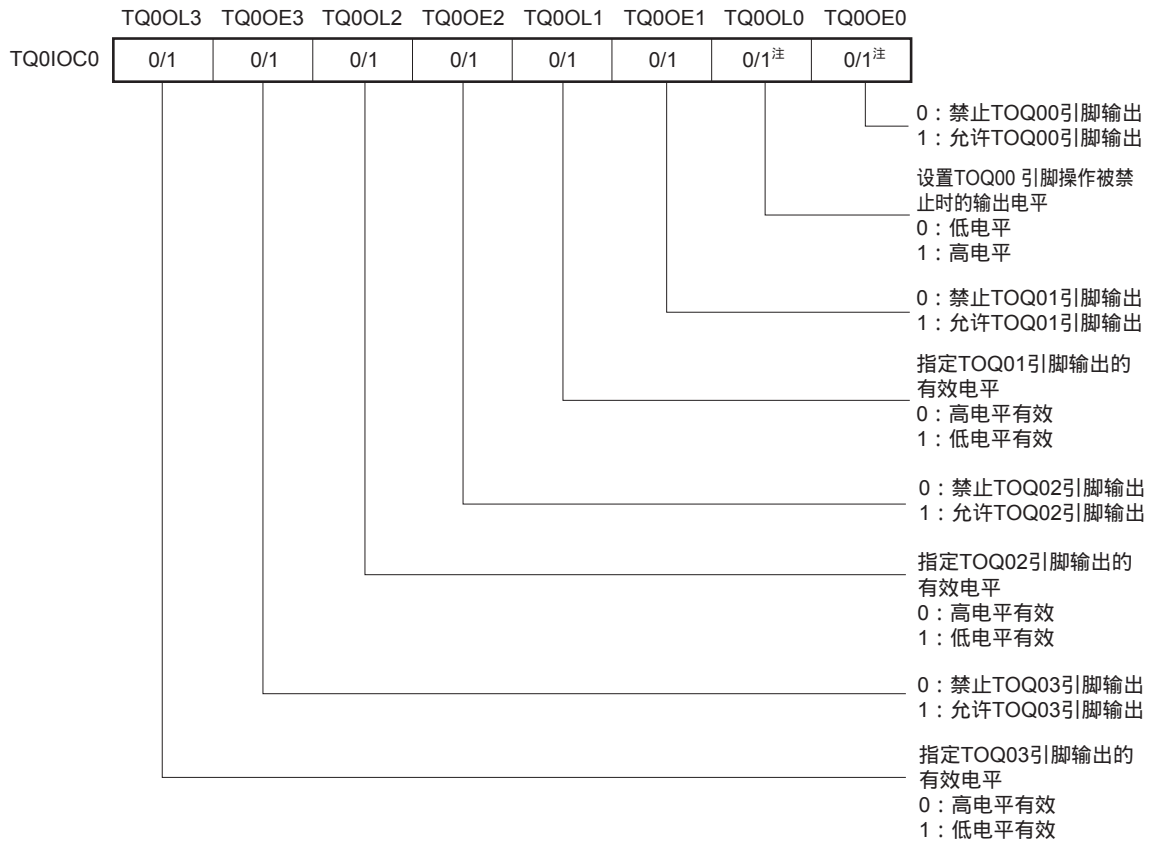
(b) TMQ0 控制寄存器 1 (TQ0CTL1)



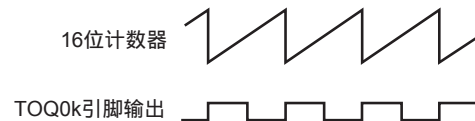
注 TQ0CTL1.TQ0EEE 位 = 1 时该设置无效。

图 7-26. PWM 输出模式的寄存器设置 (2/3)

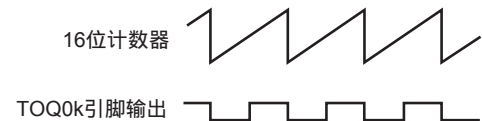
(c) TMQ0 I/O 控制寄存器 0 (TQ0IOC0)



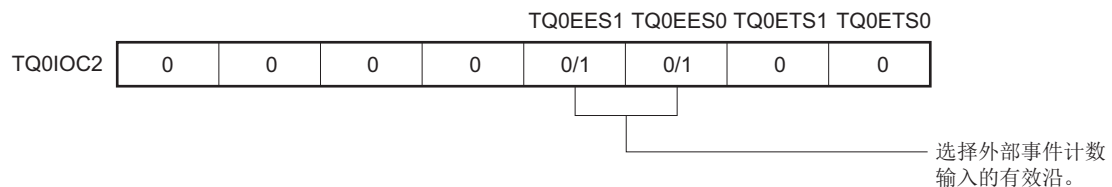
· 当TQ0OLk位 = 0



· 当TQ0OLk位 = 1



(d) TMQ0 I/O 控制寄存器 2 (TQ0IOC2)



(e) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

可以通过读取 TQ0CNT 寄存器来读取 16 位计数器的计数值。

注 当 PWM 输出模式下不使用 TOQ00 引脚时, 要将该位清零。

图 7-26. PWM 输出模式的寄存器设置 (3/3)

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

如果将 TQ0CCR0 寄存器的值设置为 D_0 ，将 TQ0CCR k 寄存器的值设置为 D_k ，那么 PWM 波形的周期和有效电平如下所示。

周期 = $(D_0 + 1) \times$ 计数时钟周期

有效电平宽度 = $D_k \times$ 计数时钟周期

- 备注**
1. PWM 输出模式下，并未使用 TMQ0 I/O 控制寄存器 1 (TQ0IOC1) 和 TMQ0 选项寄存器 0 (TQ0OPT0)。
 2. 通过写 TMQ0 捕捉/比较寄存器 1 (TQ0CCR1) 来使更新 TMQ0 捕捉/比较寄存器 2 (TQ0CCR2) 和 TMQ0 捕捉/比较寄存器 3 (TQ0CCR3) 有效。

(1) PWM 输出模式的操作流程

图 7-27. PWM 输出模式的软件处理流程 (1/2)

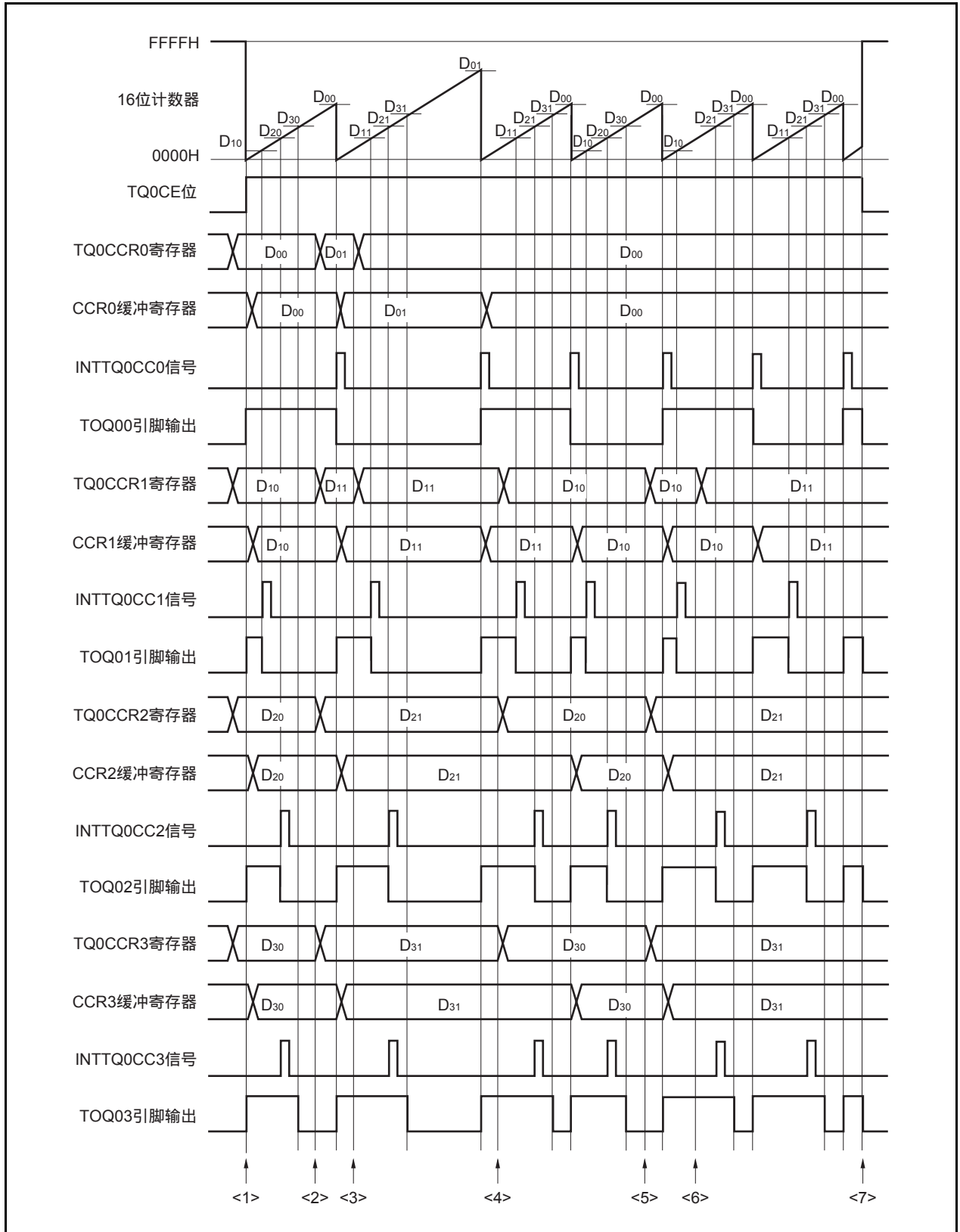
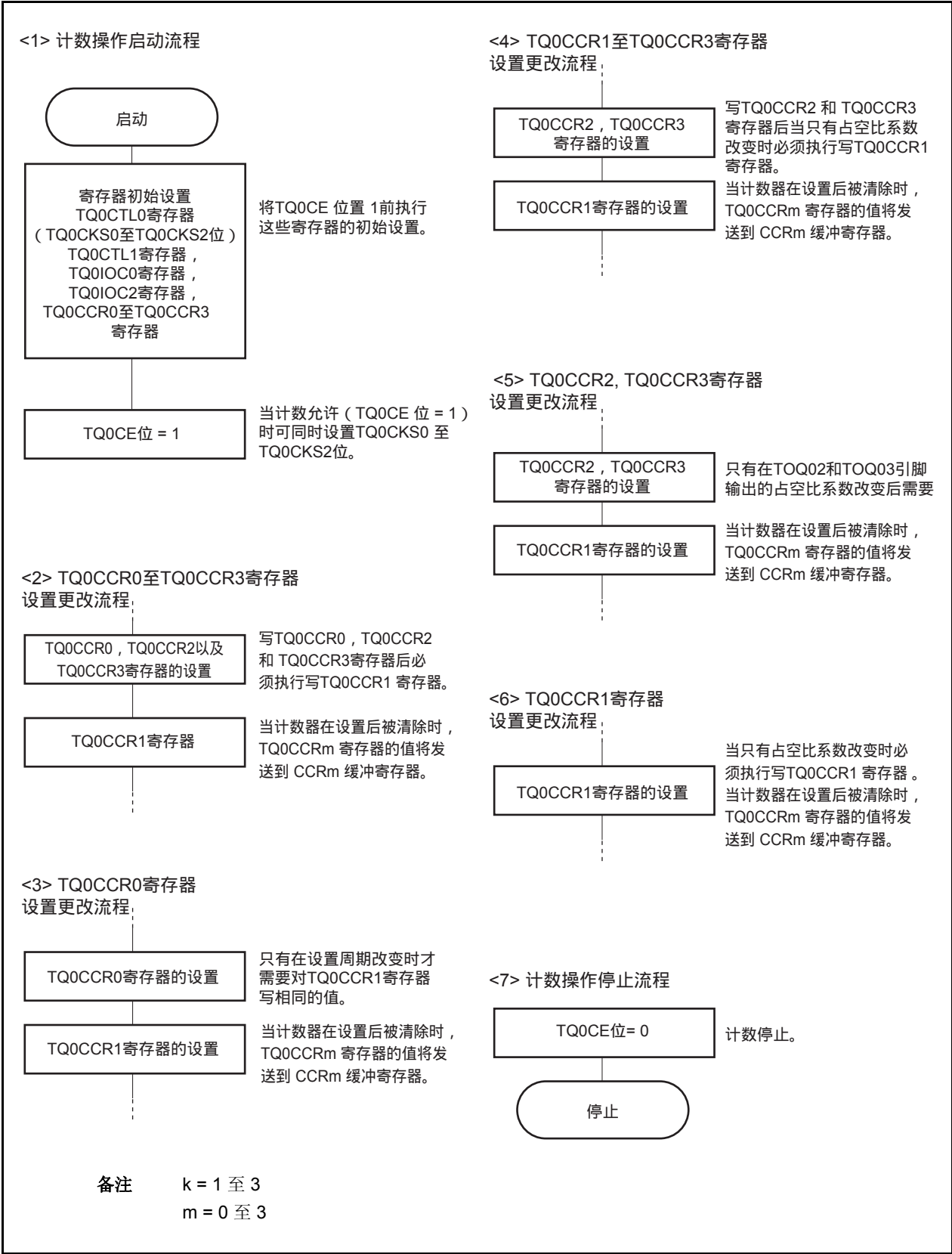


图 7-27. PWM 输出模式的软件处理流程 (2/2)

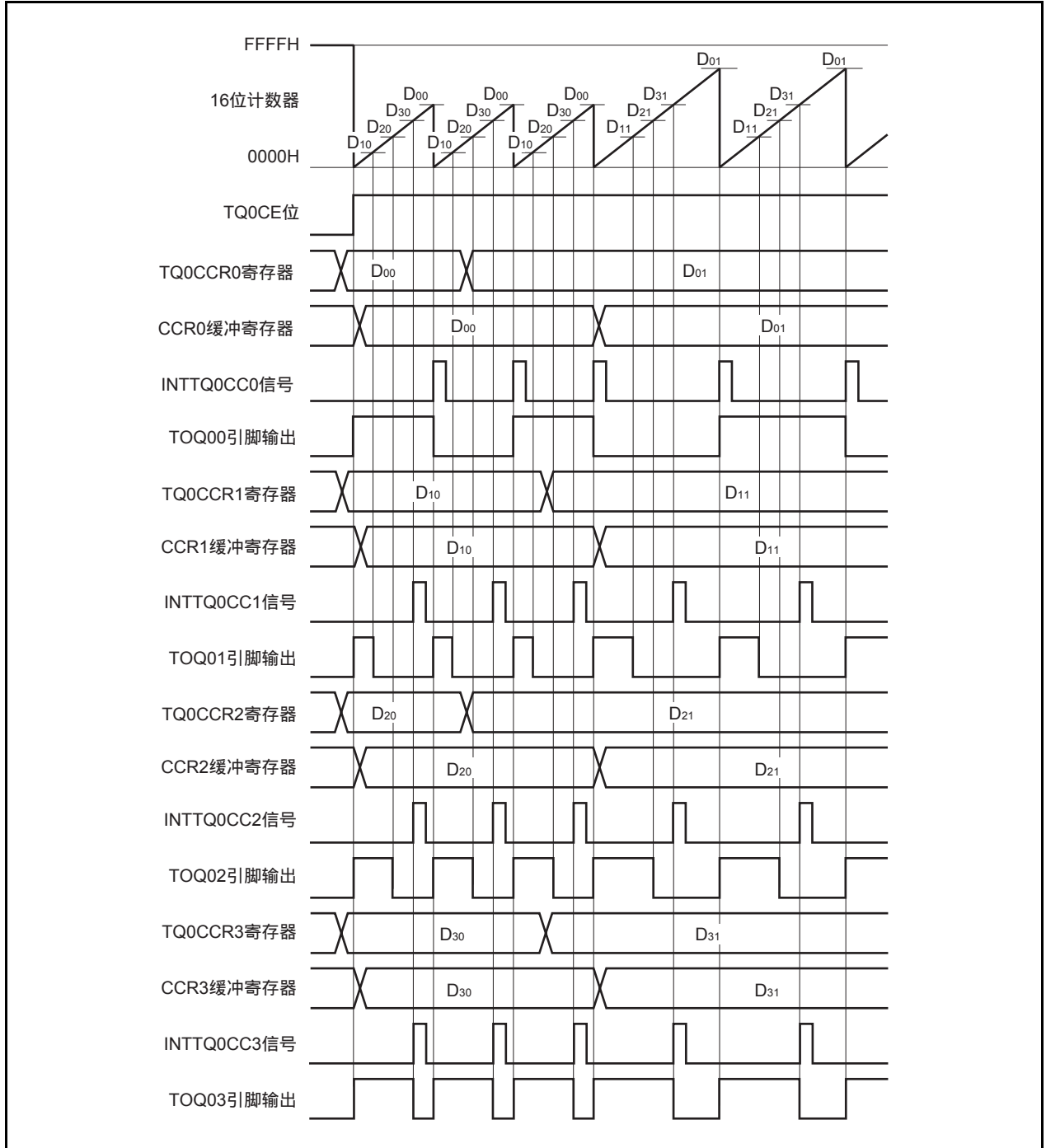


(2) PWM 输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

计数器计数过程中，若改变 PWM 波形，则要在最后进行对 TQ0CCR1 寄存器的写入操作。

对 TQ0CCR1 寄存器进行写入操作之后，若需再次更改 TQ0CCRk 寄存器的值，须等待下一个 INTTQ0CC0 信号被检测到后，再进行相关操作。



若要将 TQ0CCRm 寄存器中设置的数据传送至 CCRm 缓冲寄存器，必须对 TQ0CCR1 寄存器进行写入操作。

同时改变 PWM 波形的周期和有效电平时，先将周期值写入 TQ0CCR0 寄存器，然后向 TQ0CCR2 和 TQ0CCR3 寄存器写入有效电平宽度，最后向 TQ0CCR1 寄存器写入有效电平宽度。

只改变 PWM 波形的有效电平宽度（占空比系数）时，首先向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平然后再向 TQ0CCR1 寄存器设置有效电平。

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

若只改变通过 TOQ02 和 TOQ03 引脚输出的 PWM 波形的有效电平宽度（占空比系数）时，先要向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平宽度，然后向 TQ0CCR1 寄存器写入相同的值。

数据被写入 TQ0CCR1 寄存器之后，16 位计数器被清零时，TQ0CCRm 寄存器中的数值才会被同步传送至 CCRm 缓冲寄存器中，此时 CCRm 缓冲寄存器中的数值将被用来与 16 位计数器的计数值相比较。

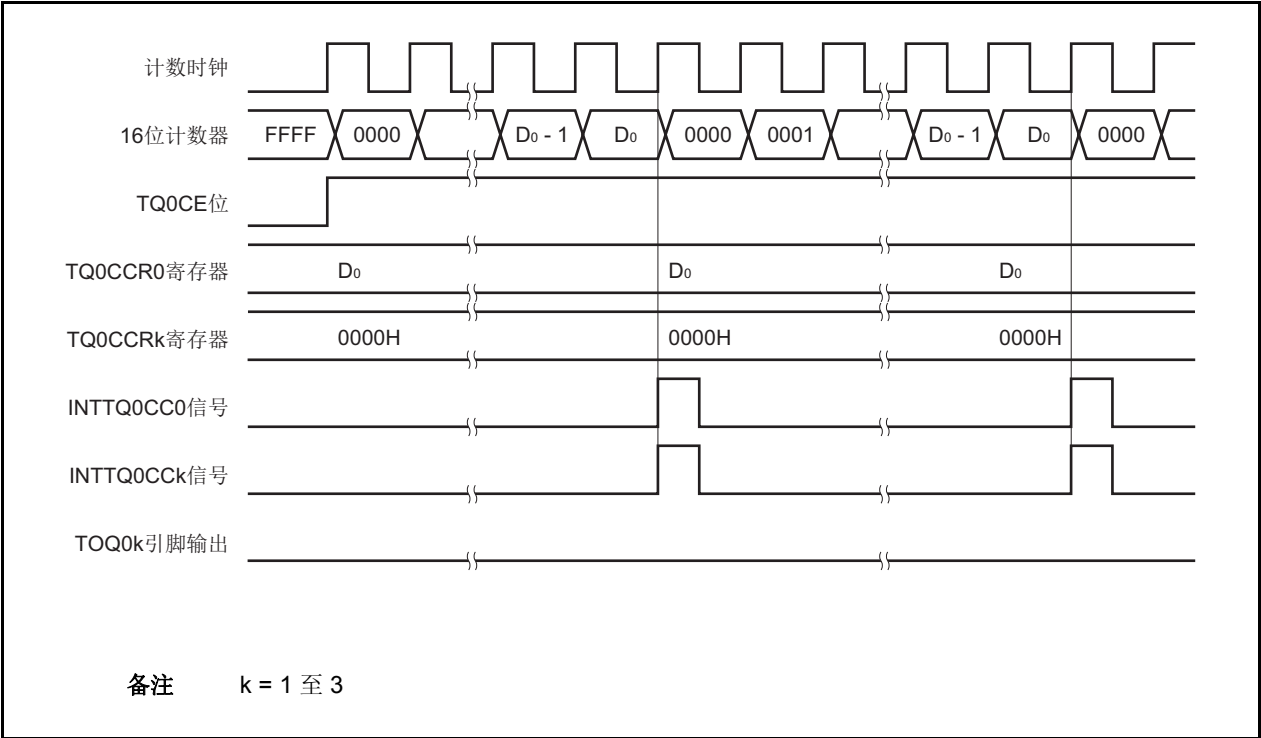
若只改变 PWM 波形的周期，首先要将周期设置于 TQ0CCR0 寄存器，然后向 TQ0CCR1 寄存器写入相同的值。

若要在写入 TQ0CCR1 寄存器后再次写入 TQ0CCR0 至 TQ0CCR3 寄存器，需要在产生 INTTQ0CC0 信号后再进行此步操作。否则，CCRm 缓冲寄存器的值可能无法确定，因为数据从 TQ0CCRm 寄存器传送到 CCRm 缓冲寄存器的时刻与 TQ0CCRm 寄存器的写入时刻相冲突。

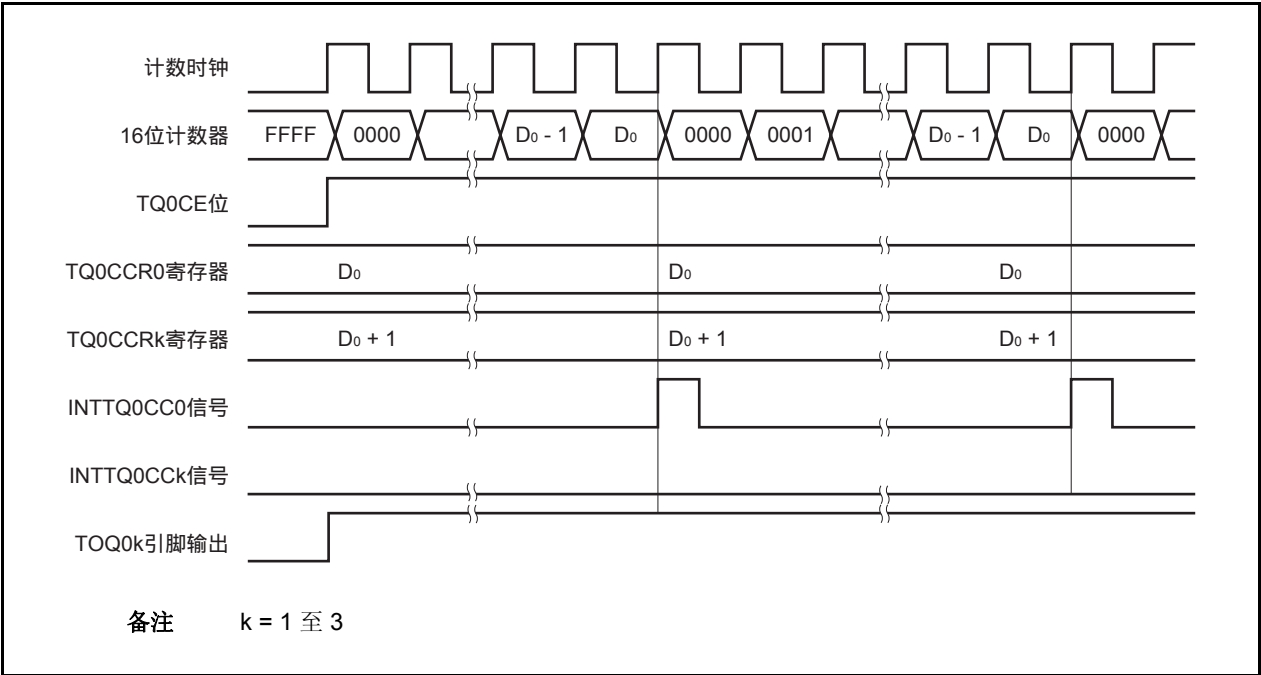
备注 m = 0 至 3

(b) PWM 波形的 0%/100% 输出

若要输出 0% 的波形，则需将 TQ0CCRk 寄存器设置为 0000H。如果 TQ0CCR0 寄存器被设置为 FFFFH，那么 INTTQ0CCk 信号将周期性产生。

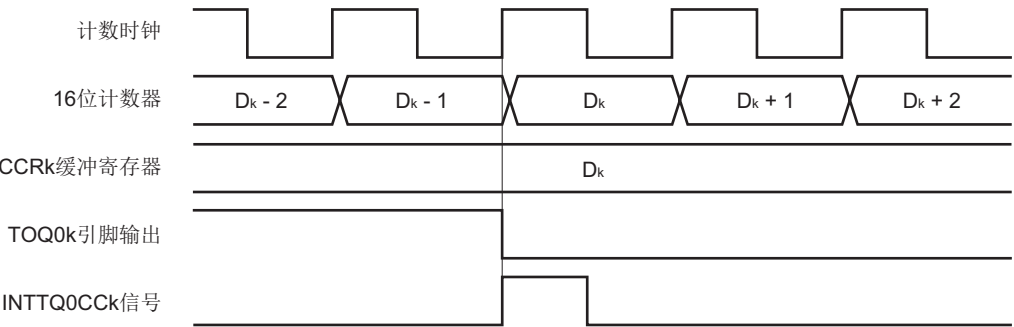


若要输出 100% 波形，则需将 TQ0CCRk 寄存器设置为 (TQ0CCR0 寄存器值 + 1)。TQ0CCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。



(c) 比较匹配中断请求信号的产生时序 (INTTQ0CCK)

PWM 输出模式下，INTTQ0CCK 信号的产生时序与其它模式下 INTTQ0CCK 信号的产生时序有所不同，这里，INTTQ0CCK 信号将在 16 位计数器的计数值与 TQ0CCRk 寄存器的值相匹配时产生。



备注 k = 1 至 3

通常情况下，INTTQ0CCK 信号会在 16 位计数器计数值与 TQ0CCRk 寄存器的值匹配之后的下一个计数时钟同步产生。

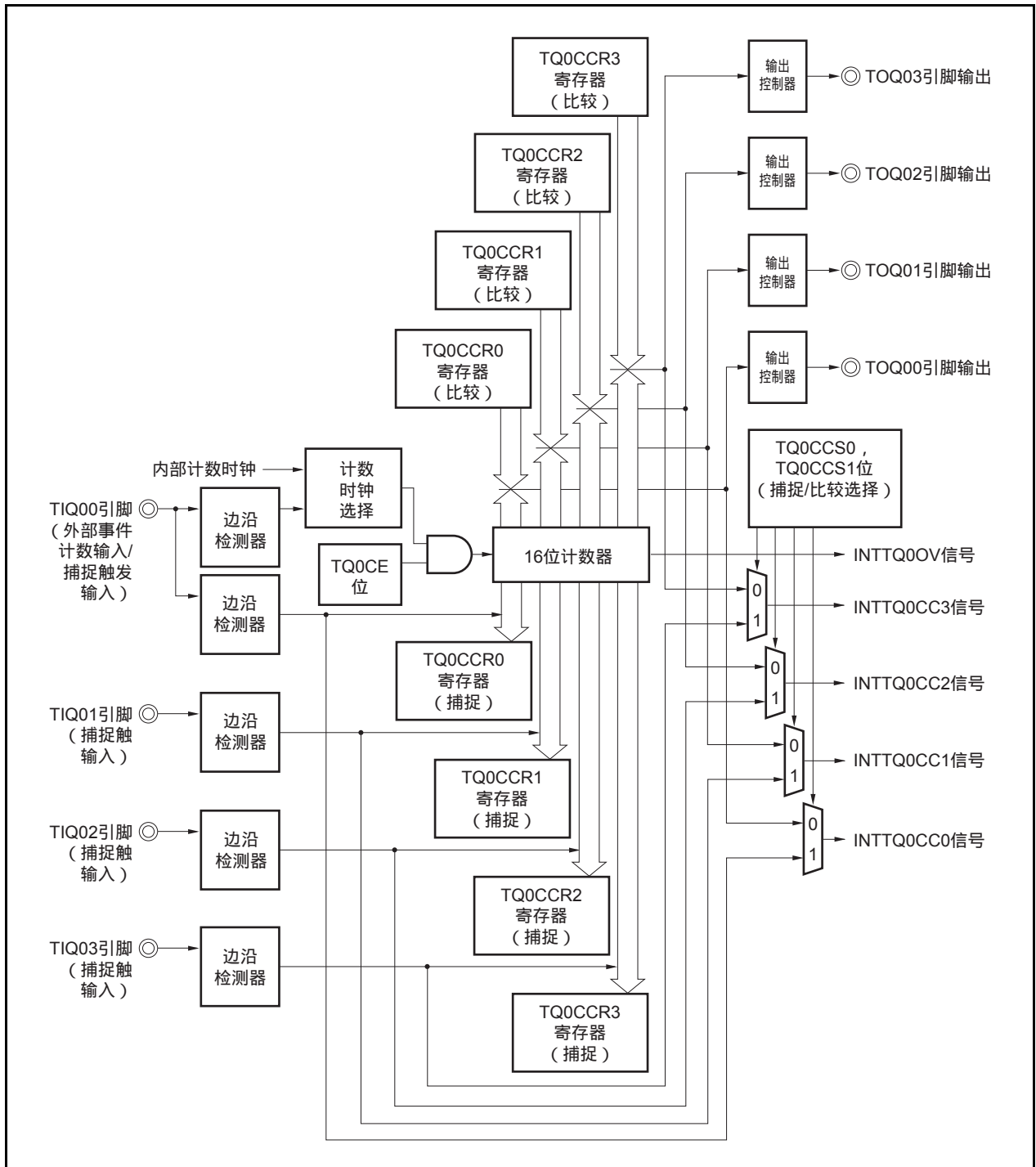
但在 PWM 输出模式下，INTTQ0CCK 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOQ0k 引脚输出信号变化的时序而改变的。

7.5.6 自由运行定时器模式 (TQ0MD2 至 TQ0MD0 位 = 101)

在自由运行定时器模式中, TQ0CTL0.TQ0CE 位被置 1 后, 16 位定时器/时间计数器 Q 开始计数。此时, 根据 TQ0OPT0.TQ0CCS0 和 TQ0OPT0.TQ0CCS1 位的设置 TQ0CCRm 寄存器的功能可以在比较寄存器和捕捉寄存器之间进行选择。

备注 m = 0 至 3

图 7-28. 自由运行定时器模式的配置图

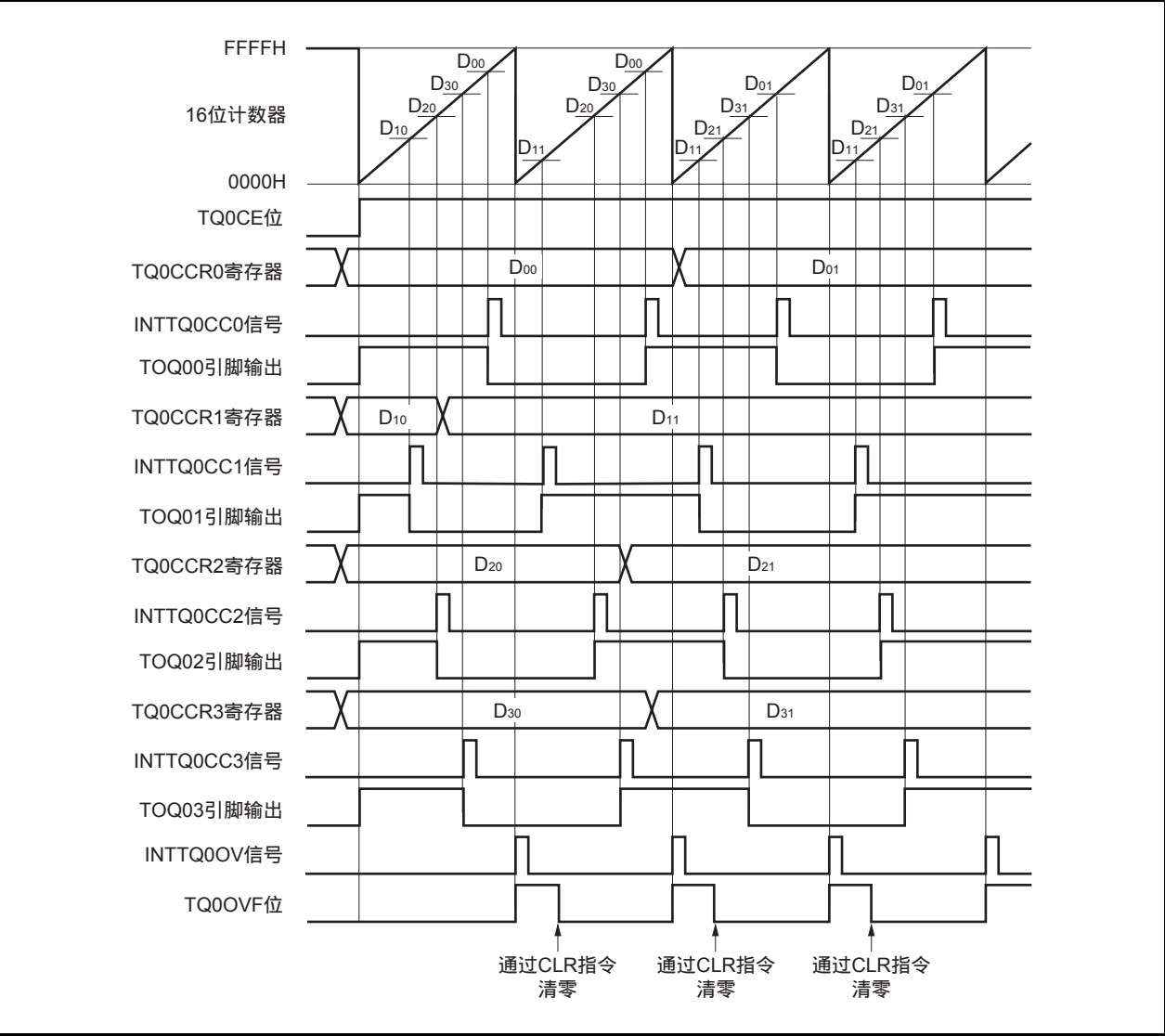


TQ0CE 位被置 1 后，16 位定时/事件计数器 Q 开始计数，同时 TOQ00 至 TOQ03 引脚输出信号的电平反转。当 16 位计数器的计数值与 TQ0CCRm 寄存器的值匹配时将产生一个比较匹配中断请求信号 (INTTQ0CCm)，同时 TOQ0m 引脚的输出电平将反转。

此后，16 位计数器将继续与计数时钟同步计数。当它计数到 FFFFH 时，将在下一个计数时钟时产生一个溢出中断请求信号 (INTTQ0OV)，同时计数器被清零并重新开始计数。此时，溢出标志 (TQ0OPT0.TQ0OVF 位) 也会被置 1。请通过软件使用 CLR 指令将溢出标志位清零。

可以在计数器计数过程中改写 TQ0CCRm 寄存器的值。如果对 TQ0CCRm 寄存器的值进行了改写，那么新的值将立即生效并用于与计数值的比较。

图 7-29. 自由运行定时器模式的基本时序（比较功能）



TQ0CE 位被置 1 后, 16 位计数器开始计数。当检测到 TIQ0m 引脚输入信号的有效沿时, 16 位计数器的计数值被保存在 TQ0CCRm 寄存器中, 同时产生一个捕捉中断请求信号 (INTTQ0CCm)。

此后, 16 位计数器将继续与计数时钟同步计数。当它计数到 FFFFH 时, 将在下一个计数时钟时产生一个溢出中断请求信号 (INTTQ0OV), 同时计数器被清零并重新开始计数。此时, 溢出标志 (TQ0OVF 位) 也会被置 1。请通过软件使用 CLR 指令将溢出标志位清零。

图 7-30. 自由运行定时器模式的基本时序 (捕捉功能)

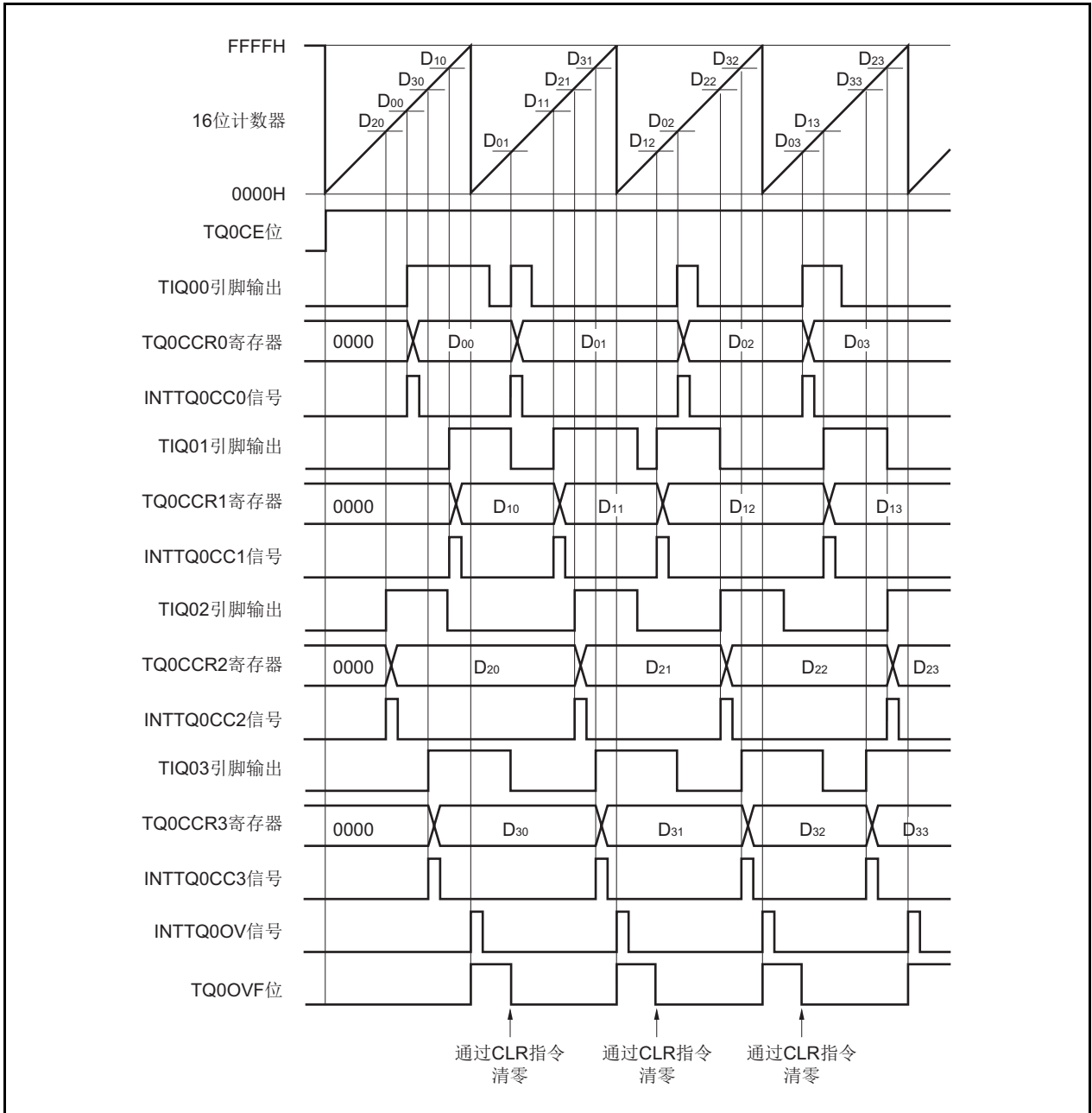
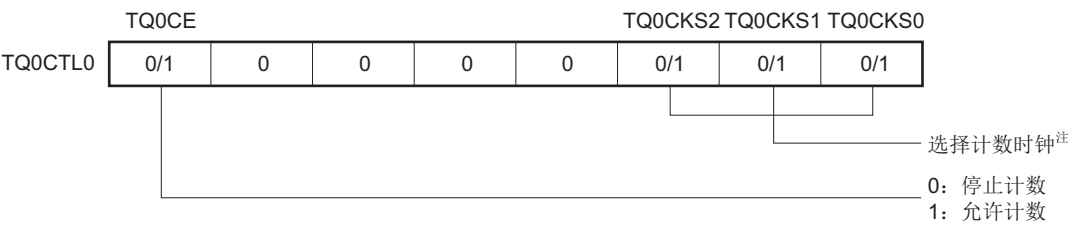


图 7-31. 自由运行定时器模式的寄存器设置 (1/3)

(a) TMQ0 控制寄存器 0 (TQ0CTL0)



注 TQ0CTL1.TQ0EEE 位 = 1 时该设置无效。

(b) TMQ0 控制寄存器 1 (TQ0CTL1)

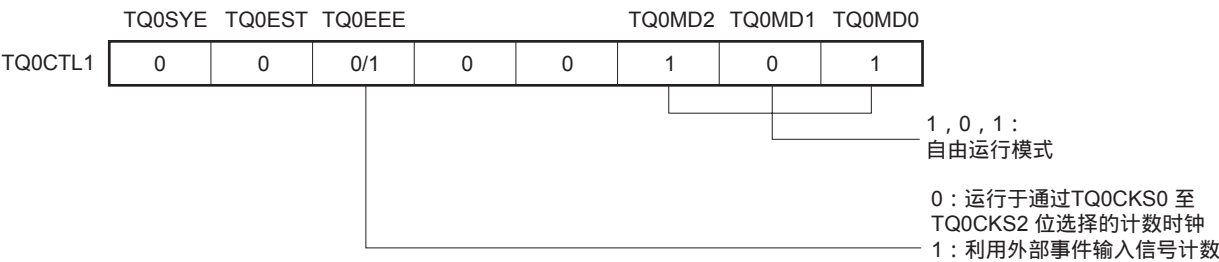


图 7-31. 自由运行定时器模式的寄存器设置 (2/3)

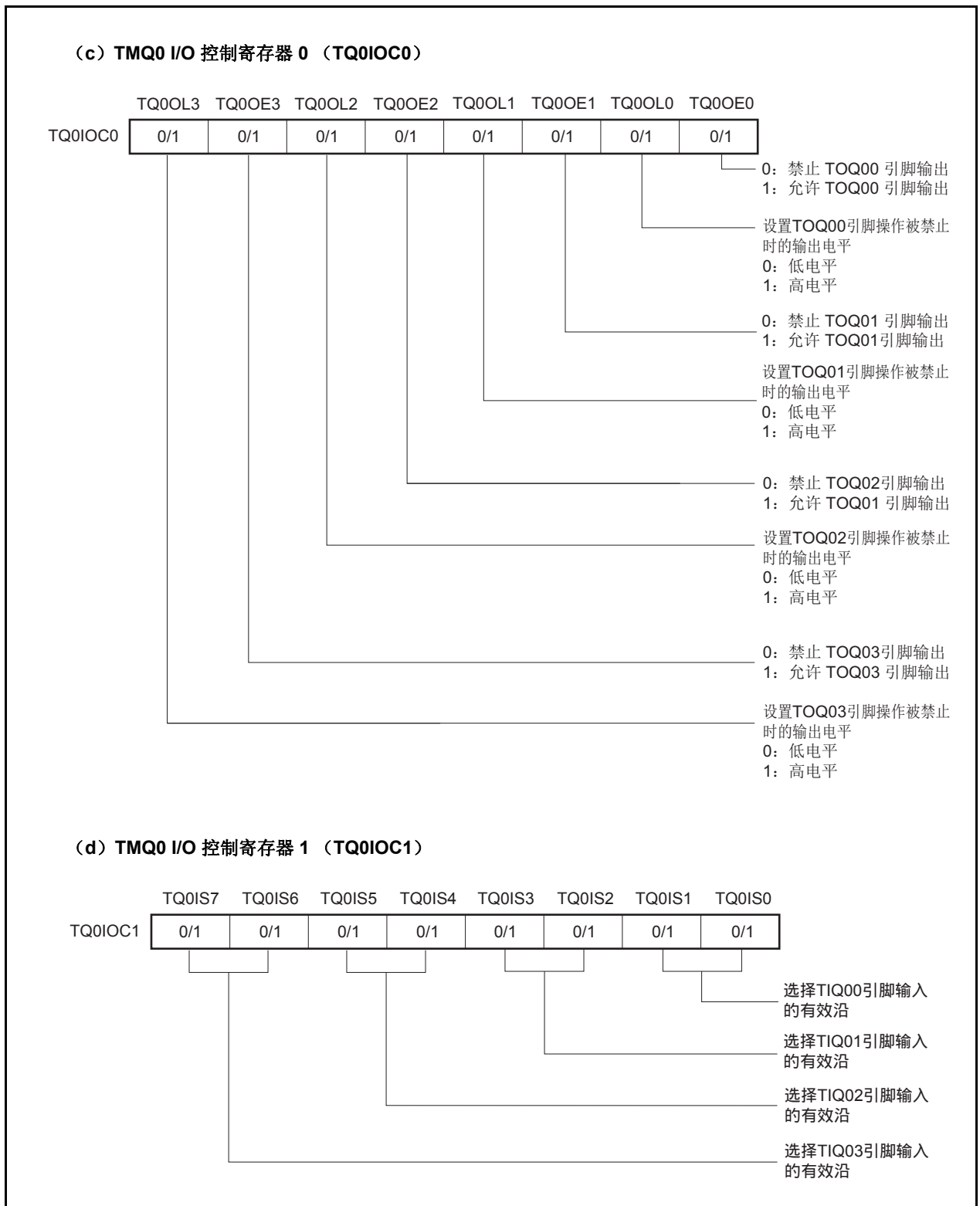
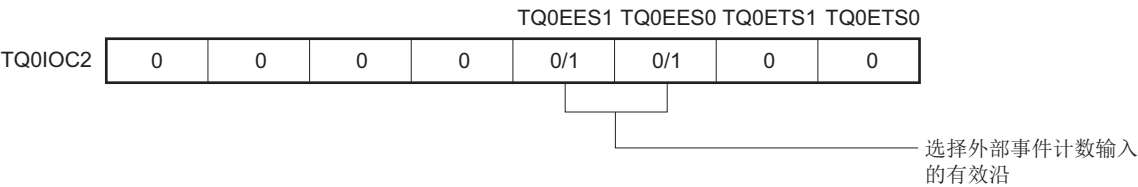
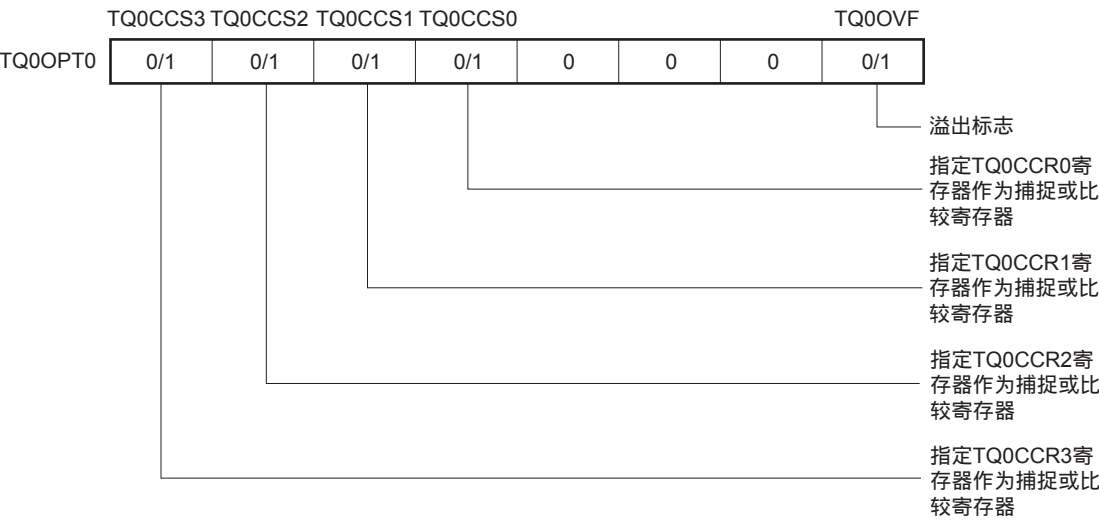


图 7-31. 自由运行定时器模式的寄存器设置 (3/3)

(e) TMQ0 I/O 控制寄存器 2 (TQ0IOC2)



(f) TMQ0 选项寄存器 0 (TQ0OPT0)



(g) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

可以通过读取 TQ0CNT 寄存器来读取 16 位计数器的计数值。

(h) TMQ0 捕捉/比较寄存器 0 至 3 (TQ0CCR0 至 TQ0CCR3)

根据 TQ0OPT0.TQ0CCSm 位的设置，这些寄存器的功能可以在比较寄存器和捕捉寄存器之间进行选择。

当选择了捕捉寄存器功能时，若检测到 TIQ0m 引脚输入信号的有效沿，则 16 位计数器此刻的计数值会被保存到这些寄存器中。

当选择了比较寄存器功能且 TQ0CCRm 寄存器的值被设置为 Dm 时，INTTQ0CCm 信号将在计数器计数到 (Dm + 1) 时产生，同时 TOQ0m 引脚的输出电平反转。

备注 m = 0 至 3

(1) 自由运行定时器模式的操作流程

(a) 当捕捉/比较寄存器用作比较寄存器时

图 7-32. 自由运行定时器模式的软件处理流程（比较功能）（1/2）

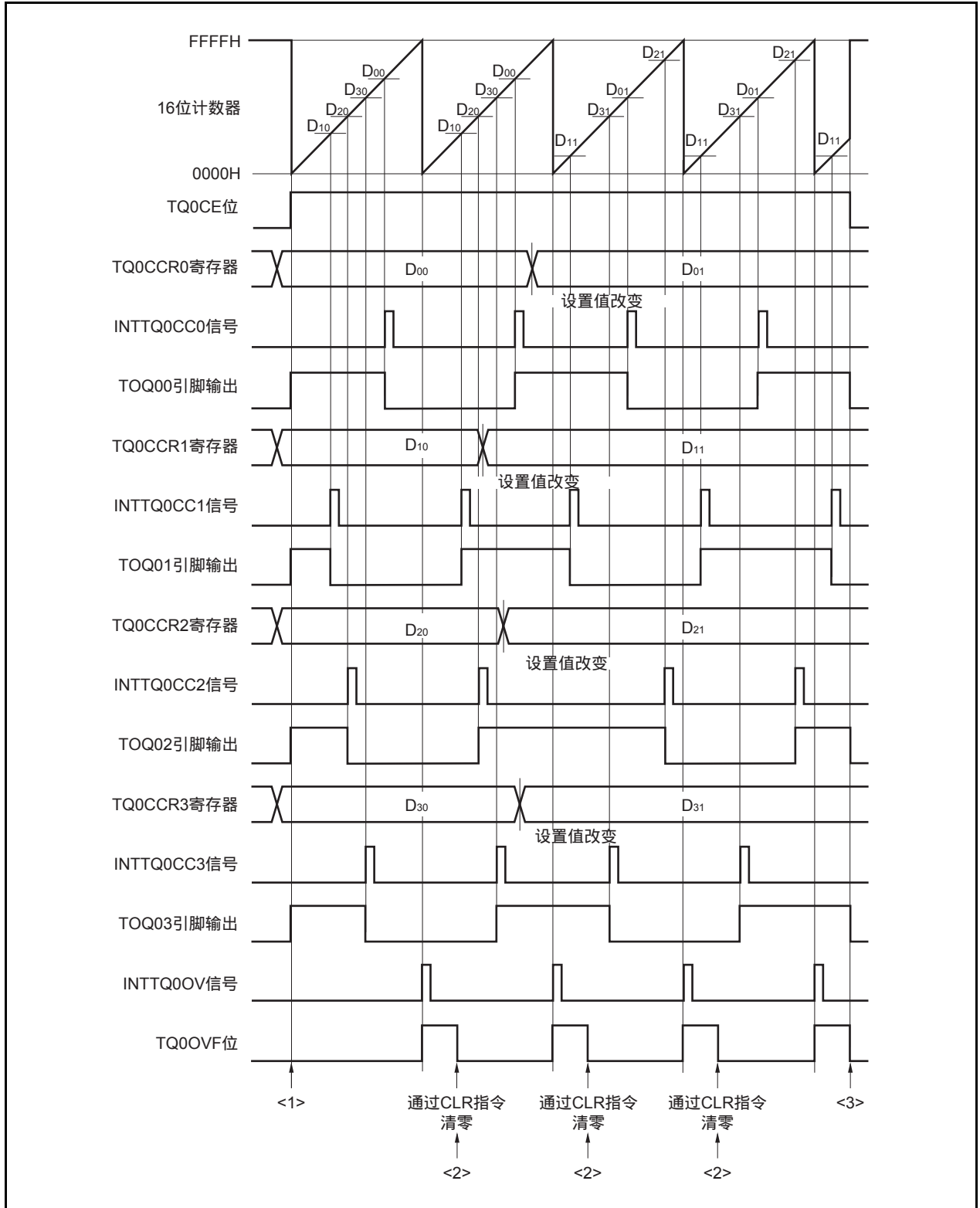
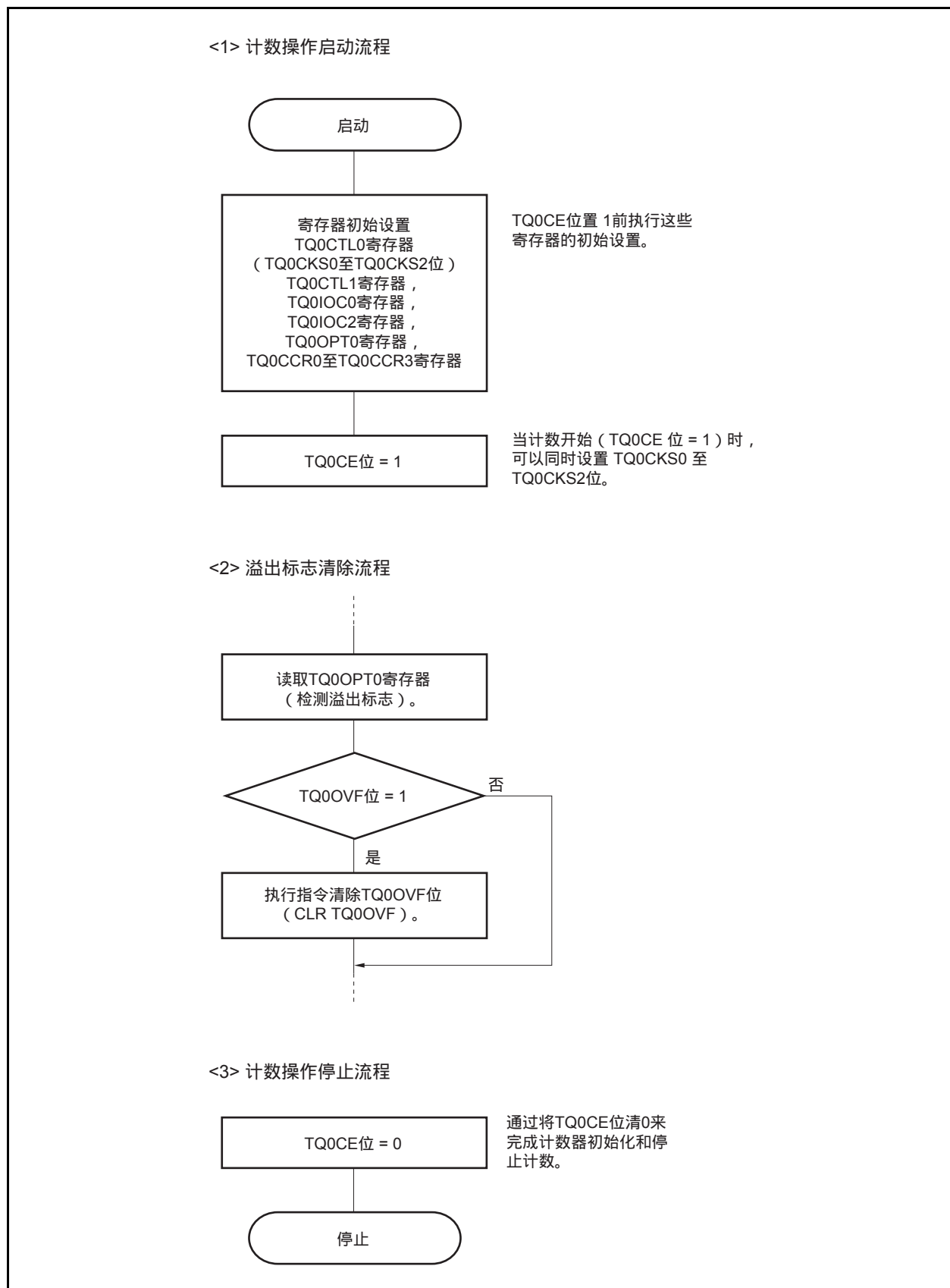


图 7-32. 自由运行定时器模式的软件处理流程（比较功能）（2/2）



(b) 当捕捉/比较寄存器用作捕捉寄存器时

图 7-33. 自由运行定时器模式的软件处理流程（捕捉功能）（1/2）

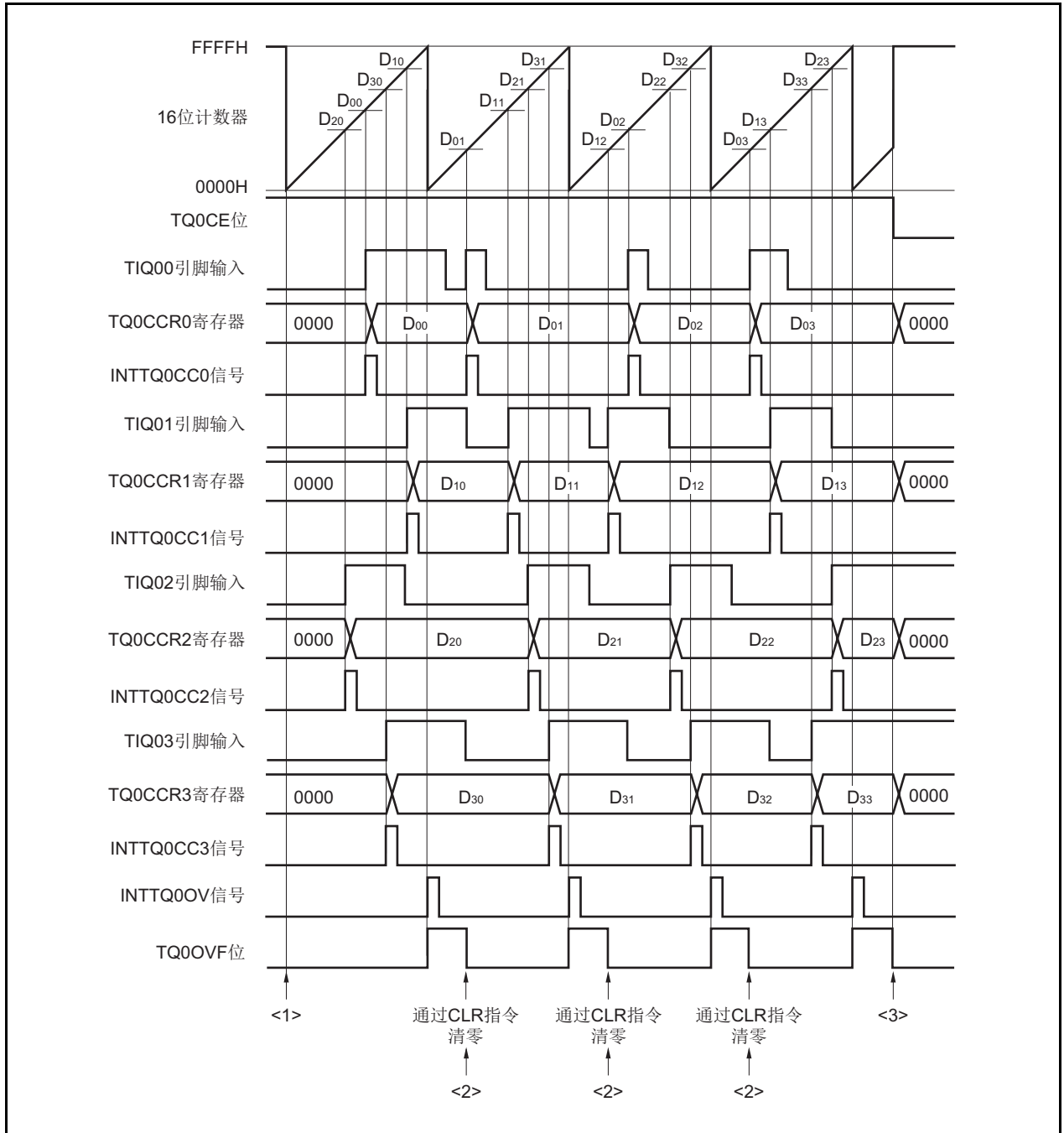
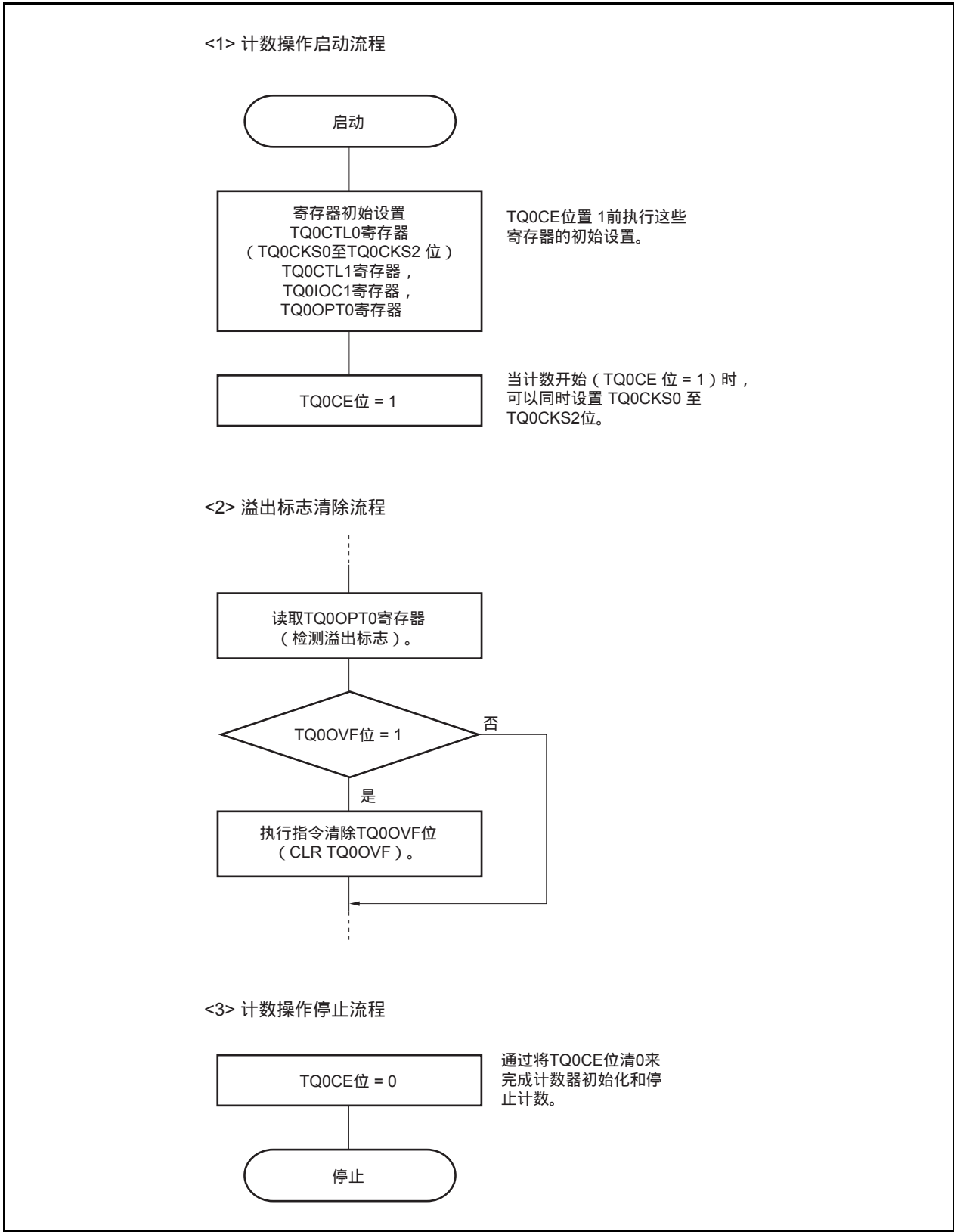


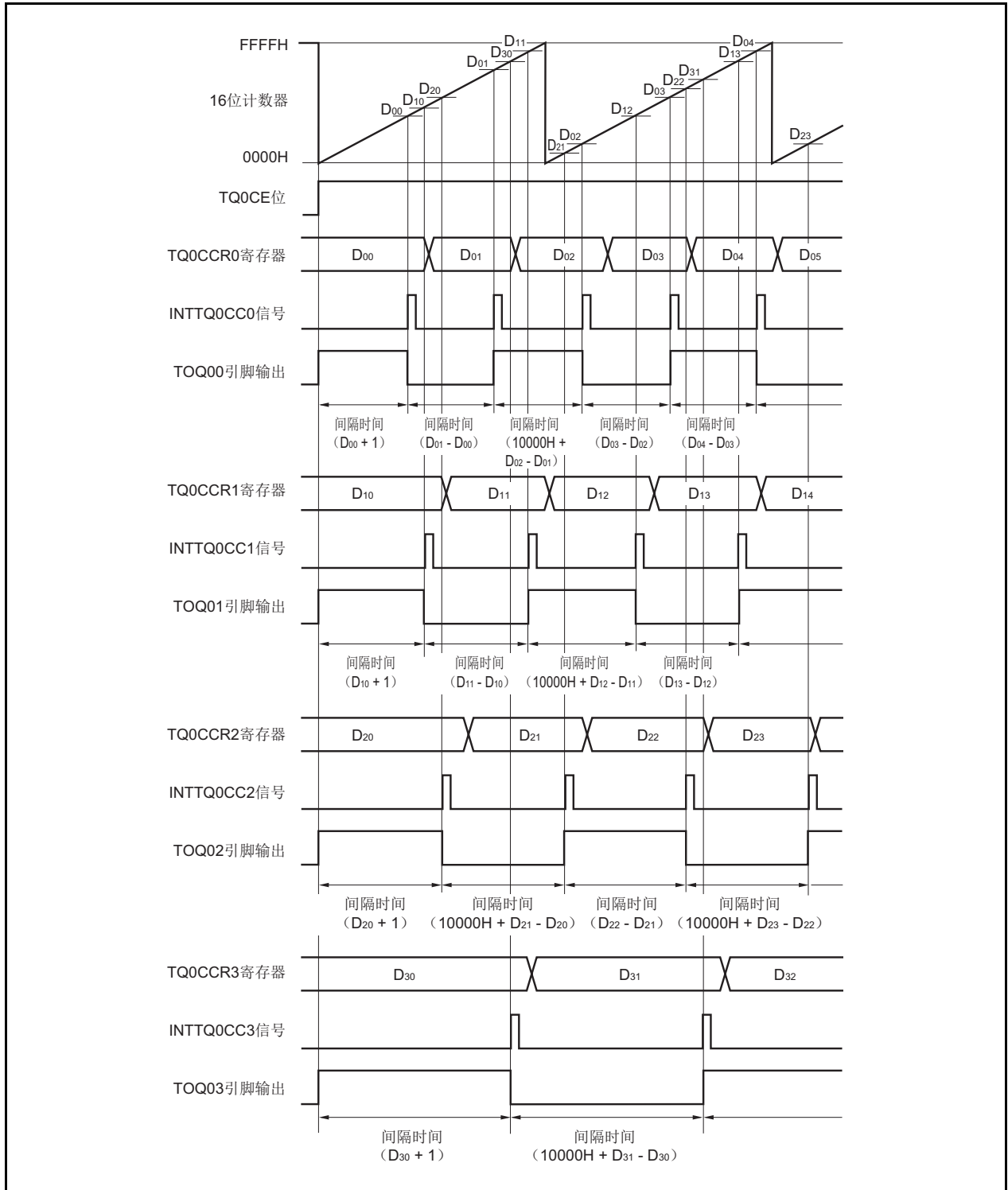
图 7-33. 自由运行定时器模式的软件处理流程（捕捉功能）（2/2）



(2) 自由运行定时器模式的操作时序

(a) 使用比较寄存器功能的时间间隔操作

当 TQ0CCRm 寄存器被作为比较寄存器使用时，16 位定时器/事件计数器 Q 可作为间隔定时器使用，这时，每当检测到 INTTQ0CCm 信号时，都需要通过软件处理的方式来设置比较值以指定下一次中断请求产生的间隔时间。



当使用自由运行定时器模式进行间隔时间操作时，可以在同一通道指定四个不同的间隔时间。
进行间隔时间操作时，必须在每次检测到 INTTQ0CCm 信号后执行的中断服务程序中将相应的值设置到 TQ0CCRm 寄存器。
向 TQ0CCRm 寄存器中重新设置的值可由以下算式求出，这里假设间隔时间为“Dm”。

比较寄存器的最初设置值： $D_m - 1$

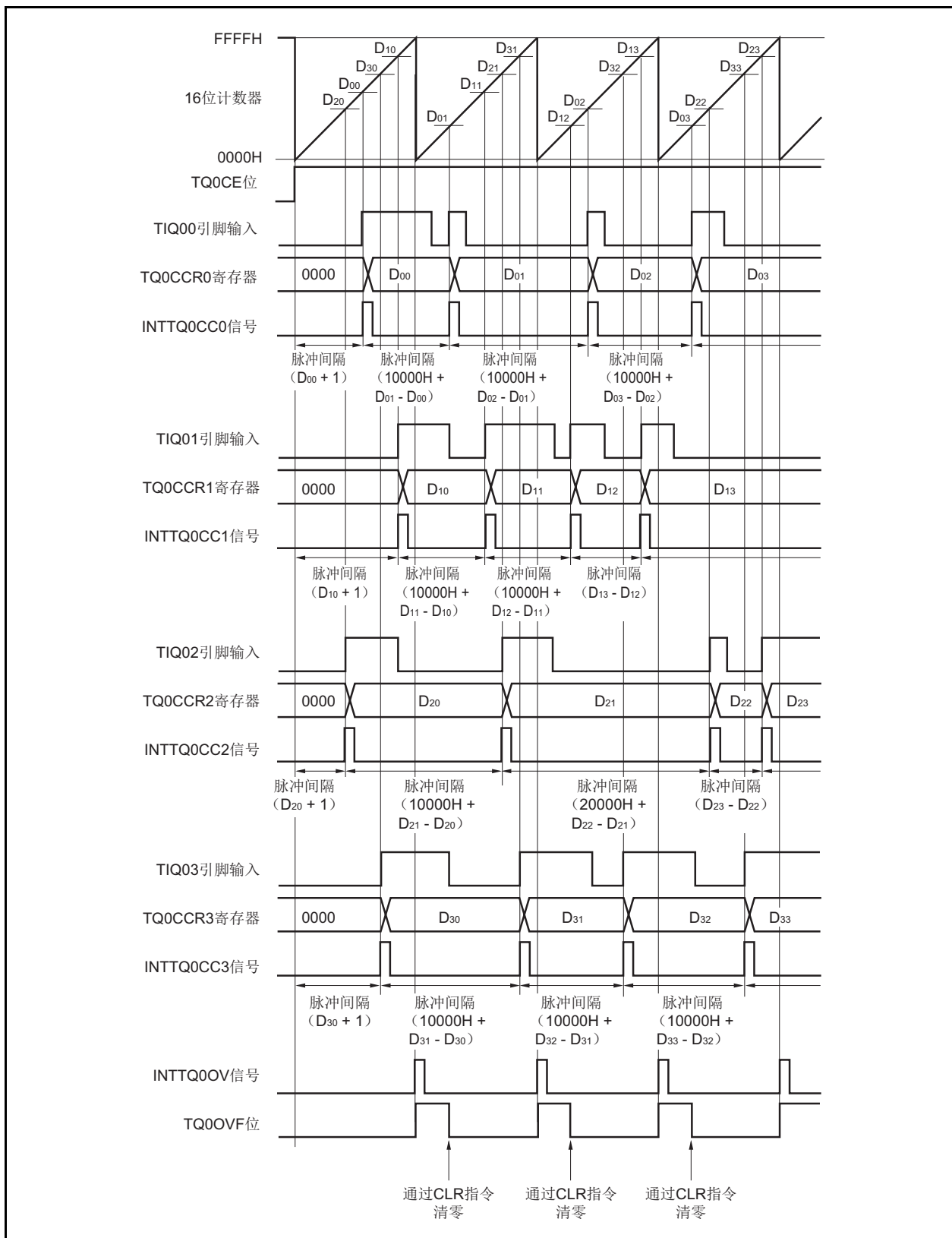
第二次设置比较寄存器以后（含第二次）的设置值：前次设置值 + Dm

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

备注 m = 0 至 3

(b) 使用捕捉寄存器进行脉宽测量

当 TQ0CCRm 寄存器用作捕捉寄存器进行脉宽测量时, 需要通过软件处理在每次检测到中断信号时对捕捉寄存器进行读取并计算间隔时间。

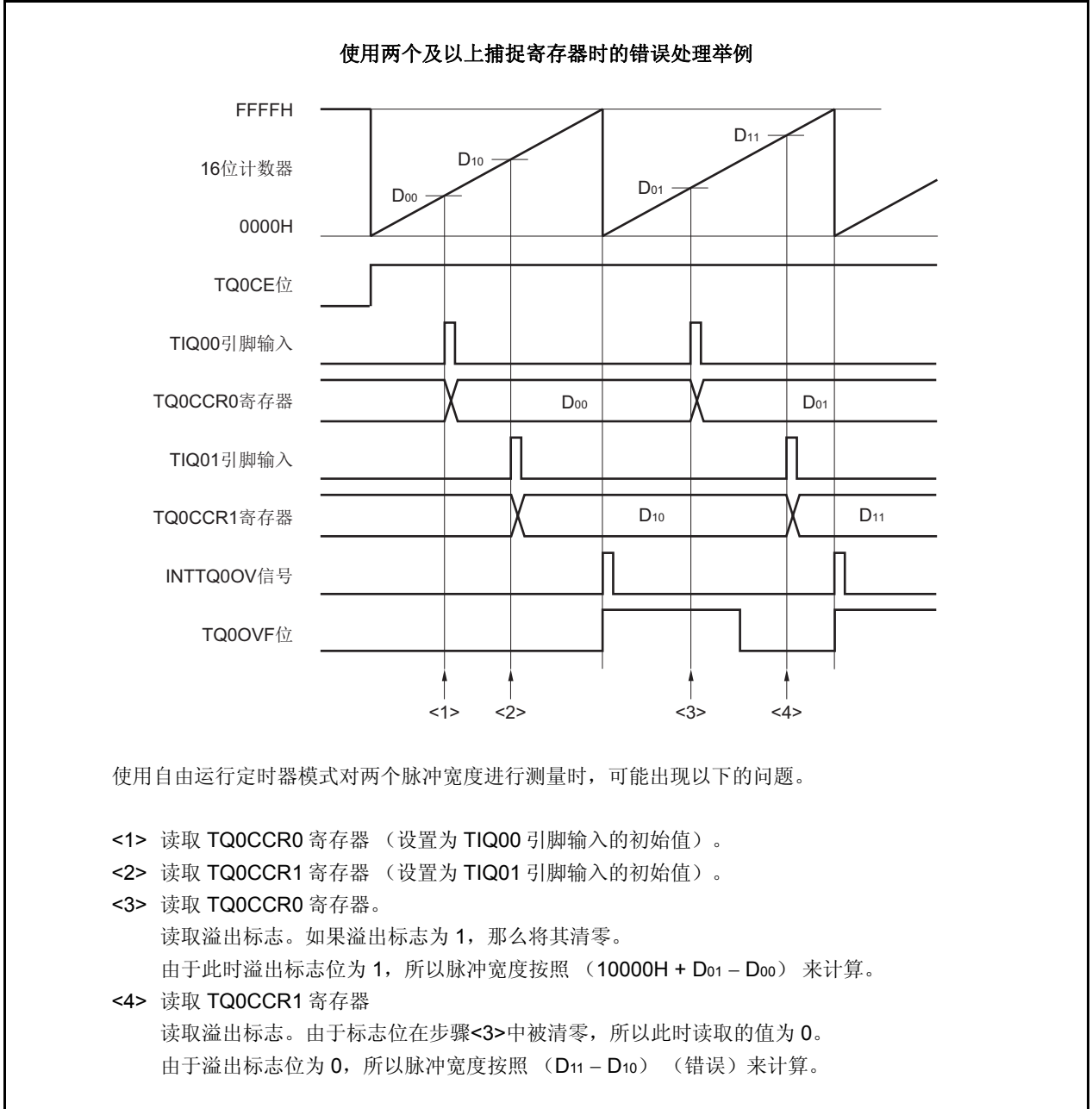


当使用自由运行定时器模式进行脉宽测量操作时，可以在同一通道对四个不同的脉冲宽度进行测量。通过与 INTTQ0CCm 信号同步读取 TQ0CCRm 寄存器的值并计算该值与上一次读取值的差，就可以求得脉冲的宽度。

备注 m = 0 至 3

(c) 使用两个及以上捕捉寄存器产生溢出时的操作

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



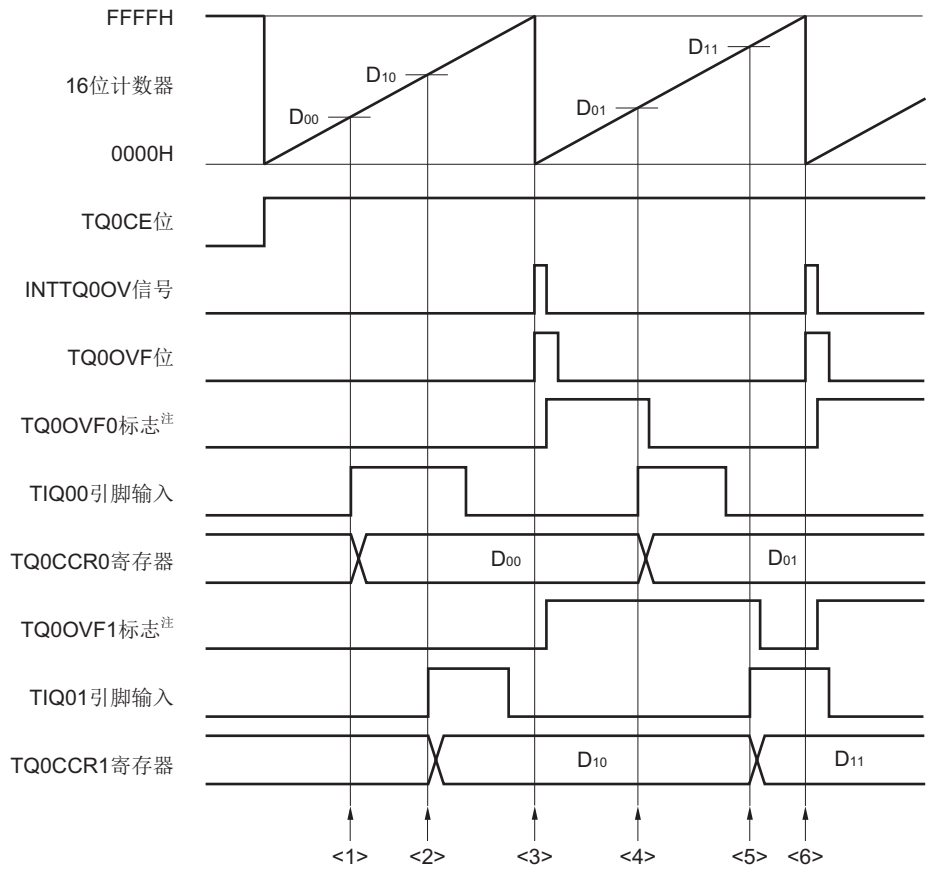
使用自由运行定时器模式对两个脉冲宽度进行测量时，可能出现以下的问题。

- <1> 读取 TQ0CCR0 寄存器（设置为 TIQ00 引脚输入的初始值）。
- <2> 读取 TQ0CCR1 寄存器（设置为 TIQ01 引脚输入的初始值）。
- <3> 读取 TQ0CCR0 寄存器。
读取溢出标志。如果溢出标志为 1，那么将其清零。
由于此时溢出标志位为 1，所以脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。
- <4> 读取 TQ0CCR1 寄存器
读取溢出标志。由于标志位在步骤<3>中被清零，所以此时读取的值为 0。
由于溢出标志位为 0，所以脉冲宽度按照 $(D_{11} - D_{10})$ （错误）来计算。

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

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

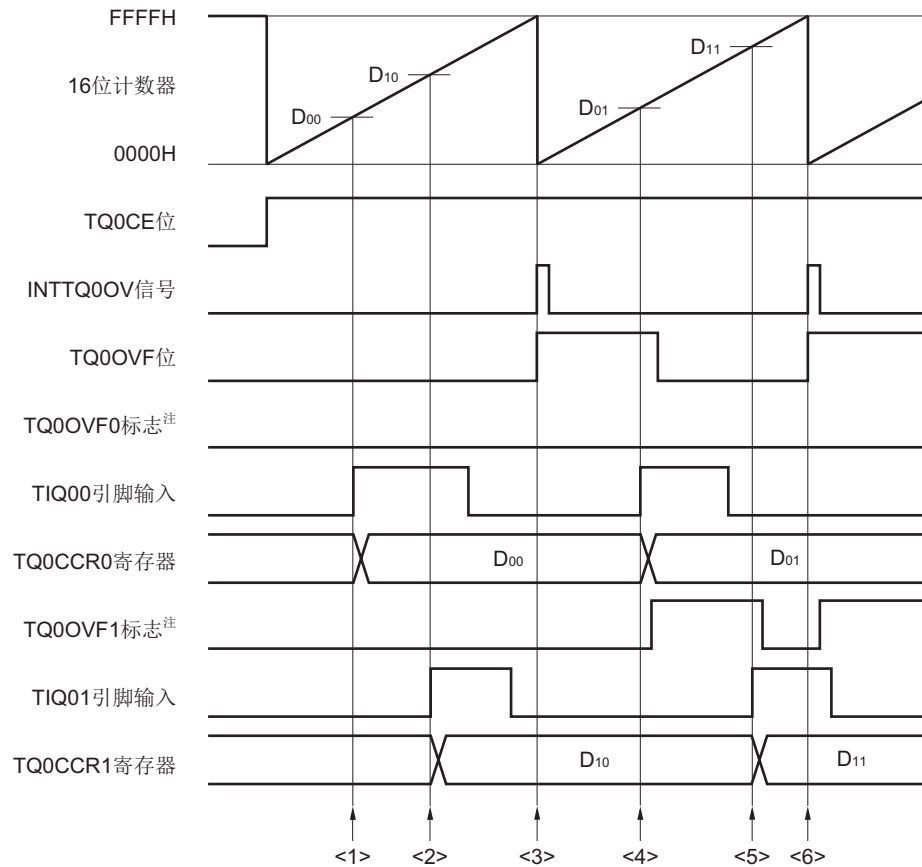
使用两个捕捉寄存器时的处理举例（利用溢出中断）



注 TQ0OVF0 和 TQ0OVF1 标志通过软件设置于内部 RAM。

- <1> 读取 TQ0CCR0 寄存器（设置为 TIQ00 引脚输入的初始值）。
- <2> 读取 TQ0CCR1 寄存器（设置为 TIQ01 引脚输入的初始值）。
- <3> 产生了溢出。在溢出中断服务程序中将 TQ0OVF0 和 TQ0OVF1 标志位置 1，同时将硬件溢出标志位清零。
- <4> 读取 TQ0CCR0 寄存器。
读取 TQ0OVF0 标志位。如果该标志位为 1，那么将其清零。
由于 TQ0OVF0 标志位为 1，脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。
- <5> 读取 TQ0CCR1 寄存器。
读取 TQ0OVF1 标志位，如果该标志位为 1，那么将其清零。（步骤<4>中将 TQ0OVF0 位清零但 TQ0OVF1 标志始终保持为 1。）
由于 TQ0OVF1 标志为 1，脉冲宽度按照 $(10000H + D_{11} - D_{10})$ （正确）来计算。
- <6> 同步步骤<3>

使用两个捕捉寄存器时的处理举例（不利用溢出中断）



注 TQ0OVF0 和 TQ0OVF1 标志通过软件设置于内部 RAM。

<1> 读取 TQ0CCR0 寄存器（设置为 TIQ00 引脚输入的初始值）。

<2> 读取 TQ0CCR1 寄存器（设置为 TIQ01 引脚输入的初始值）。

<3> 产生了溢出。但不进行任何软件处理。

<4> 读取 TQ0CCR0 寄存器。

读取溢出标志。如果溢出标志为 1，那么只把 TQ0OVF1 标志设置为 1，同时将溢出标志清零。

由于 TQ0OVF0 标志位为 1，脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。

<5> 读取 TQ0CCR1 寄存器。

读取溢出标志。由于溢出标志位在步骤<4>中被清零，所以读取结果为 0。

读取 TQ0OVF1 位。如果 TQ0OVF1 标志为 1，那么将其清零。

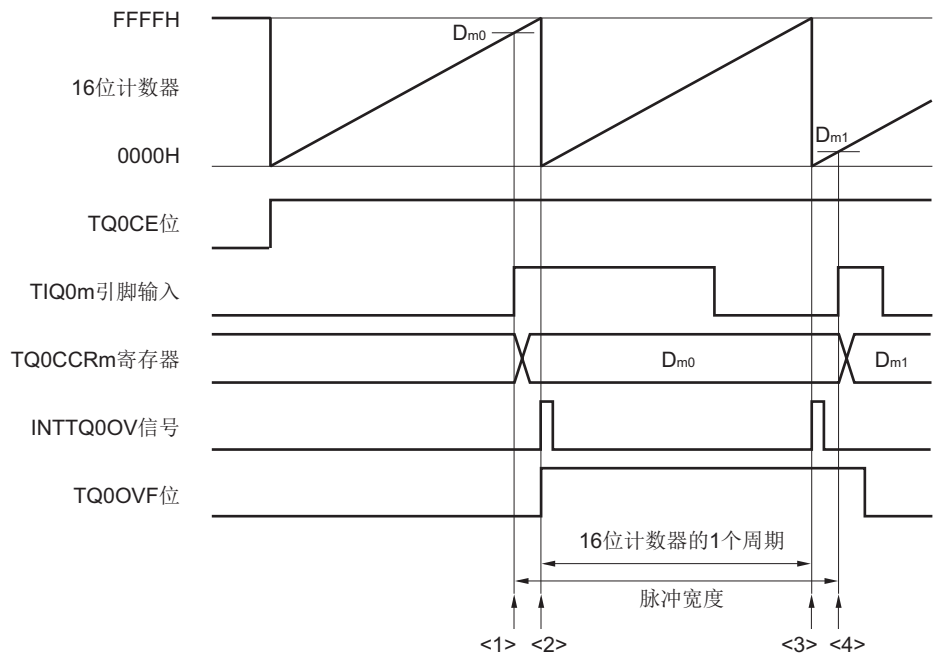
由于 TQ0OVF1 标志为 1，脉冲宽度按照 $(10000H + D_{11} - D_{10})$ （正确）来计算。

<6> 同步骤<3>

(d) 捕捉触发间隔较长时的溢出处理

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

捕捉触发间隔较长时的错误处理举例



使用自由运行定时器模式对较长的脉冲宽度进行测量时，可能出现以下的问题。

- <1> 读取 TQ0CCRm 寄存器（设置为 TIQ0m 引脚输入的初始值）。
- <2> 产生了溢出。但不进行任何软件处理。
- <3> 产生第二次溢出。但不进行任何软件处理。
- <4> 读取 TQ0CCRm 寄存器。

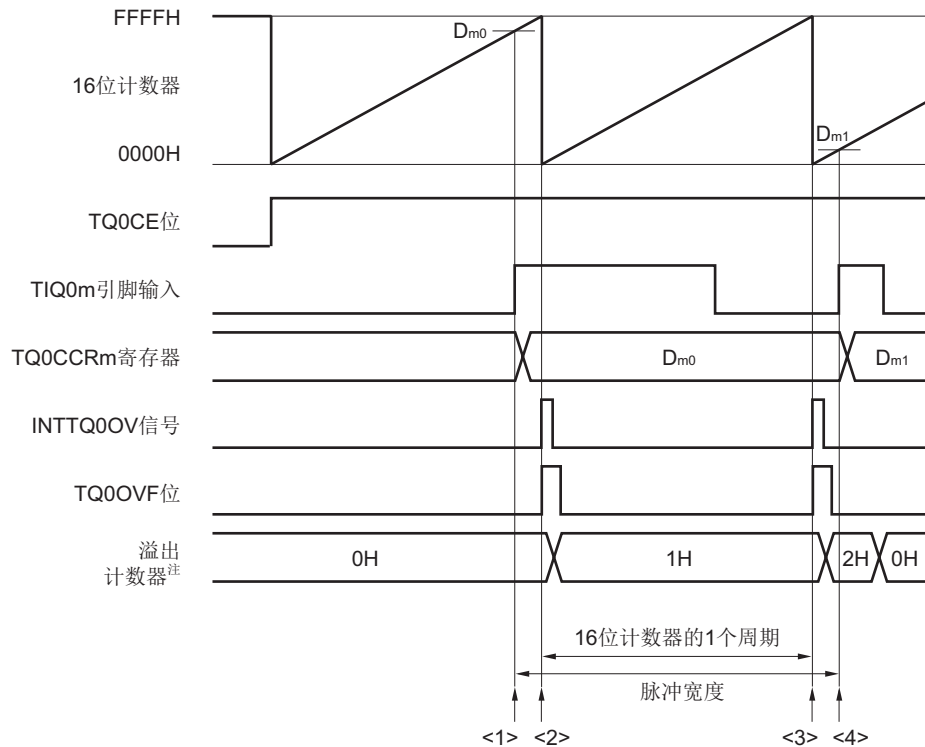
读取溢出标志位。如果溢出标志为 1，那么将其清零。

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

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

如果捕捉触发间隔时间较长以至产生两次或更多次溢出，那么计算结果可能为错误的脉冲宽度。因此，在捕捉触发间隔时间较长时，请减慢 16 位计数器的计数时钟以延长计数周期或者通过软件的方式避免错误的产生。下面给出一个使用软件方式避免错误的例子。

捕捉触发间隔较长时的处理举例



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

<1> 读取 TQ0CCRm 寄存器（设置为 TIQ0m 引脚输入的初始值）。

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

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

<4> 读取 TQ0CCRm 寄存器。

读取溢出次数计数器的值。

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

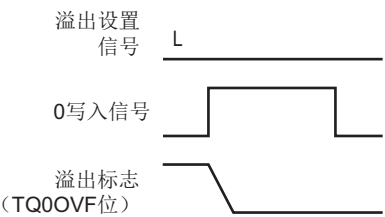
在本例中，由于溢出产生了两次，所以脉冲宽度为 $(20000H + D_{m1} - D_{m0})$ 。

将溢出次数计数器清零（0H）。

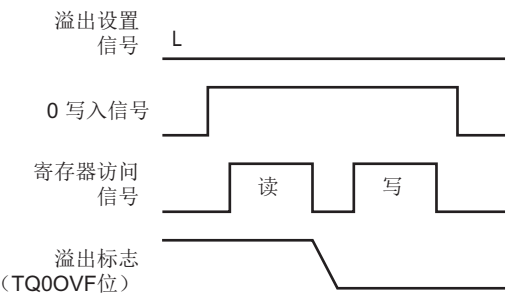
(e) 将溢出标志位清零

可以通过使用 CLR 指令清零 TQ0OVF 位或向 TQ0OPT0 寄存器写入 8 位数据（第 0 位为 0）的方法将溢出标志位清零。为了能够准确检测到溢出的发生，请务必在每次读取到 TQ0OVF 位为 1 后，立即使用位操作指令将该位清零。

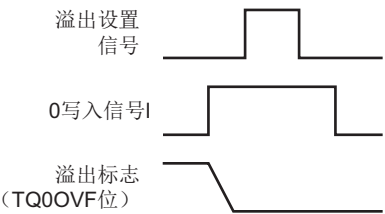
(i) 写0操作（与设置不冲突）



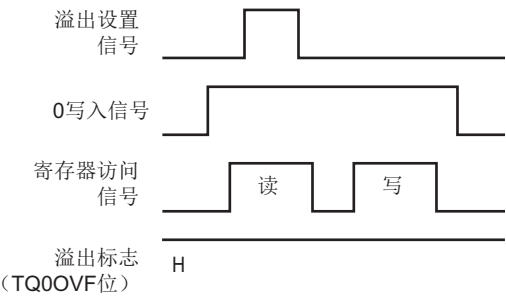
(iii) 清0操作（与设置不冲突）



(ii) 写0操作（与设置冲突）



(iv) 清0操作（与设置冲突）



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

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

7.5.7 脉宽测量模式 (TQ0MD2 至 TQ0MD0 位 = 110)

在脉宽测量模式中，TQ0CTL0.TQ0CE 位被置 1 后，16 位计数器 Q 开始计数。每当检测到 TIQ0m 引脚输入信号的有效沿时，16 位计数器的计数值就会被保存到 TQ0CCRm 寄存器中，同时计数器清零为 0000H。

两个有效沿之间的间隔时间可以在捕捉中断请求信号 (INTTQ0CCm) 产生后通过读取 TQ0CCRm 寄存器的值来计算。

可以选择 TIQ00 至 TIQ03 引脚中的任意一个作为捕捉触发输入引脚。通过 TQ0IOC1 寄存器将未使用的引脚设置为“无边沿检测”。

当使用外部时钟来测量脉宽时，只能使用 TIQ0k 引脚作为捕捉触发输入引脚，因为外部时钟输入已经固定为 TIQ00 引脚。此时要将 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位清为 00（捕捉触发输入 (TIQ00 引脚)：无边沿检测）。

备注 m = 0 至 3

k = 1 至 3

图 7-34. 脉宽测量模式的配置图

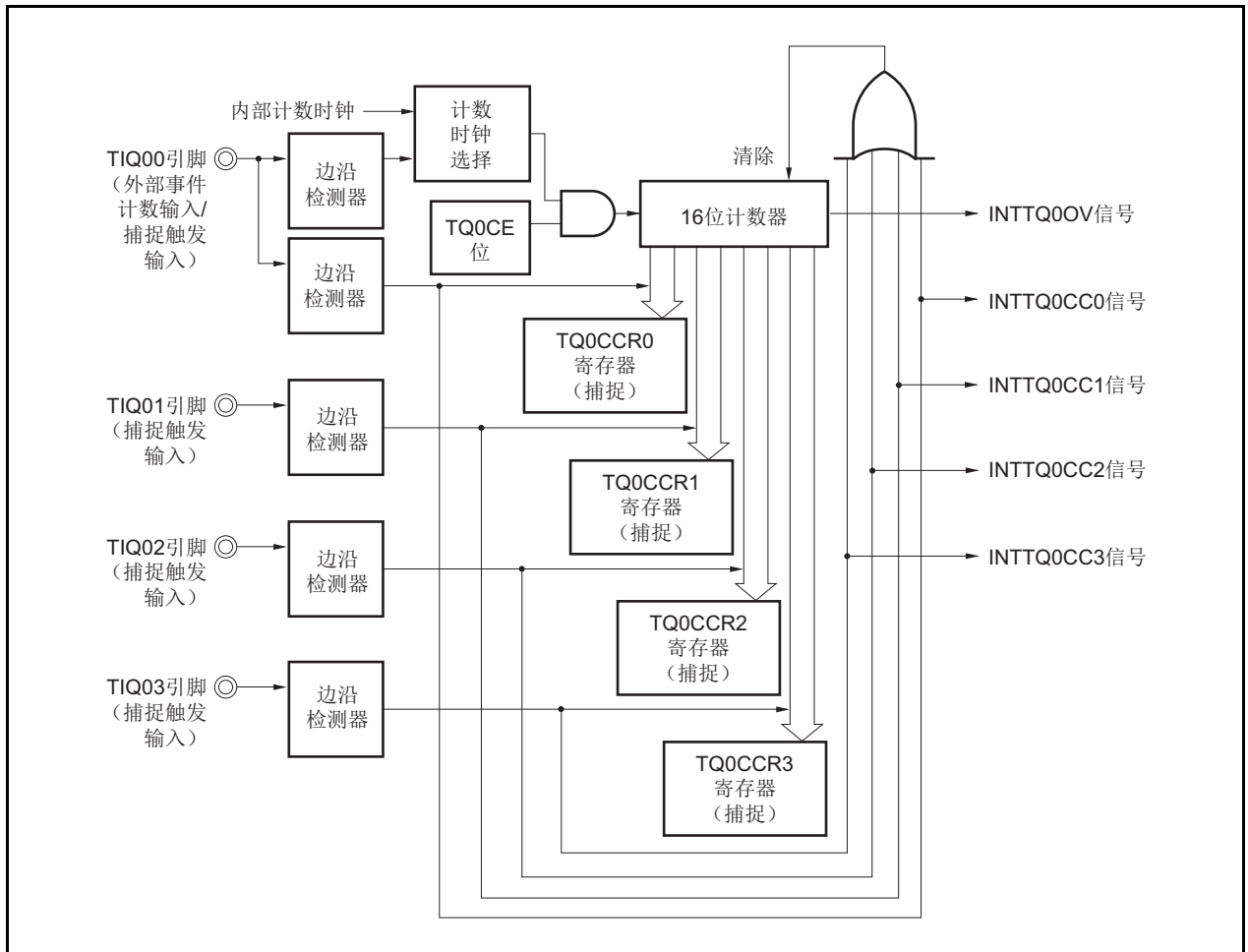
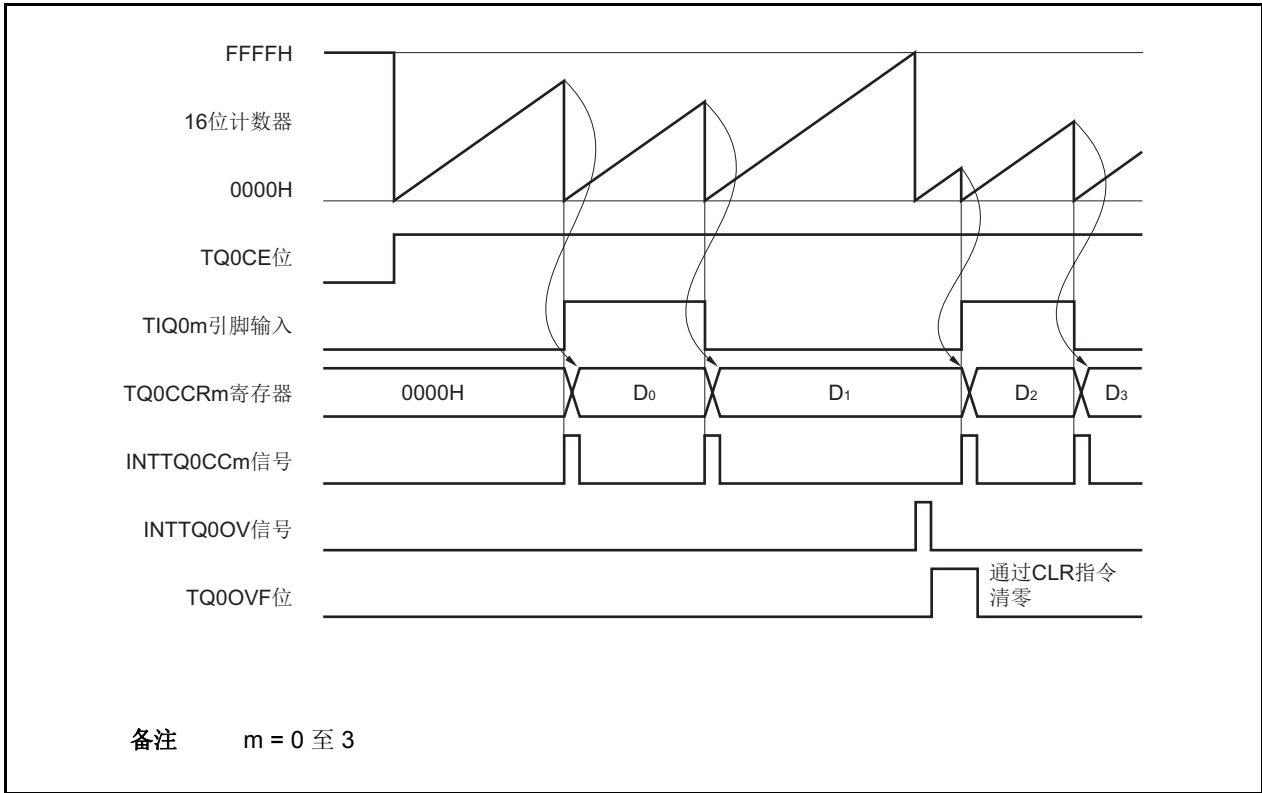


图 7-35. 脉宽测量模式的基本时序



TQ0CE 位被置 1 后，16 位计数器开始计数。当检测到 TIQ0m 引脚的有效沿时，16 位计数器的计数值将被保存到 TQ0CCRm 寄存器中，同时计数器被清零并产生一个捕捉中断请求信号（INTTQ0CCm）。脉冲宽度可由下式求得。

脉冲宽度 = 捕捉计数值 × 计数时钟周期

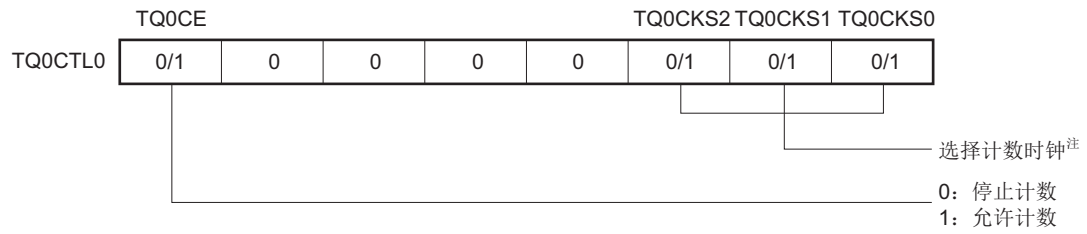
如果直到 16 位计数器计数到 FFFFH 仍没有检测到 TIQ0m 引脚的有效沿，那么将在下一个时钟时产生一个溢出中断请求信号（INTTQ0OV），同时计数器被清零并重新开始计数。这种情况下，溢出标志位（TQ0OPT0.TQ0OVF 位）也将被设置为 1。请通过软件使用 CLR 指令将溢出标志清零。如果溢出标志被设置为 1，那么脉冲宽度可由下式求得。

脉冲宽度 = （10000H × TQ0OVF 位被设置为 1 的次数 + 捕捉计数值） × 计数时钟周期

备注 m = 0 至 3

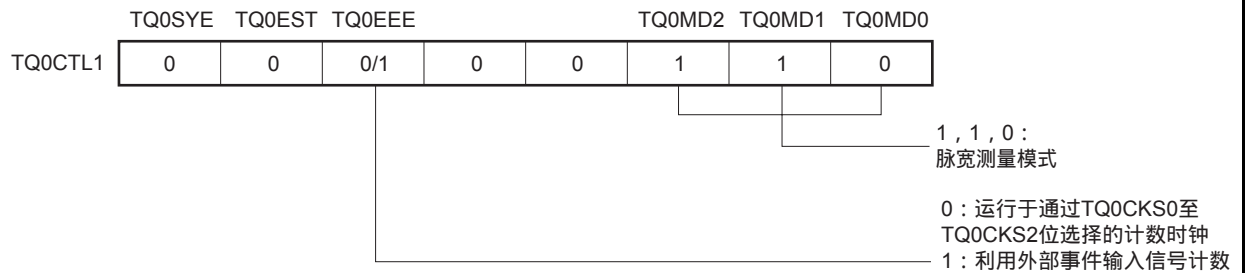
图 7-36. 脉宽测量模式的寄存器设置 (1/2)

(a) TMQ0 控制寄存器 0 (TQ0CTL0)

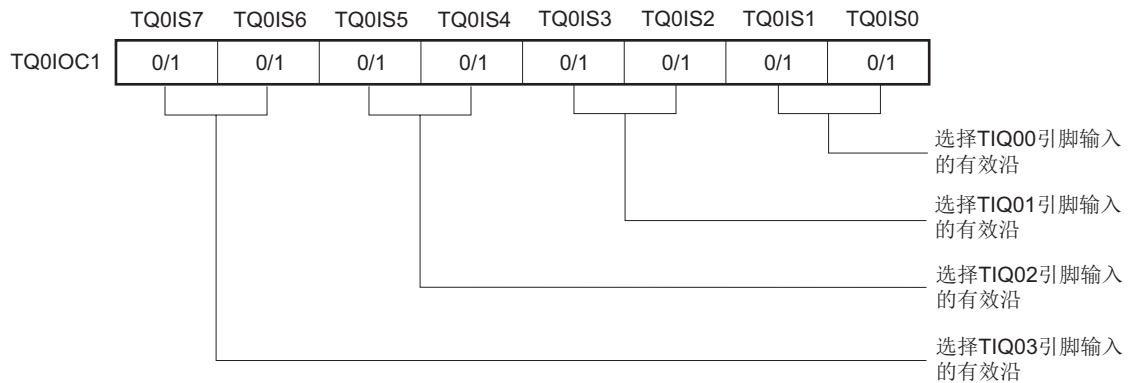


注 TQ0EEE 位 = 1 时该设置无效。

(b) TMQ0 控制寄存器 1 (TQ0CTL1)



(c) TMQ0 I/O 控制寄存器 1 (TQ0IOC1)



(d) TMQ0 I/O 控制寄存器 2 (TQ0IOC2)

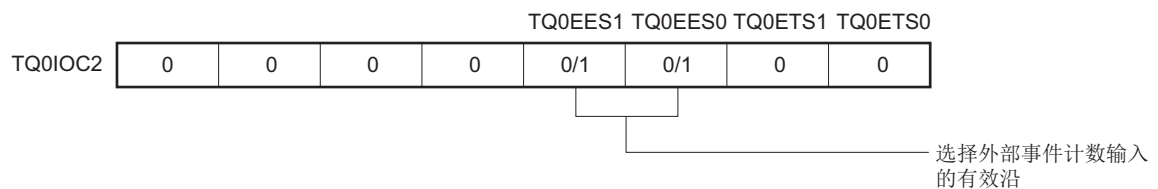
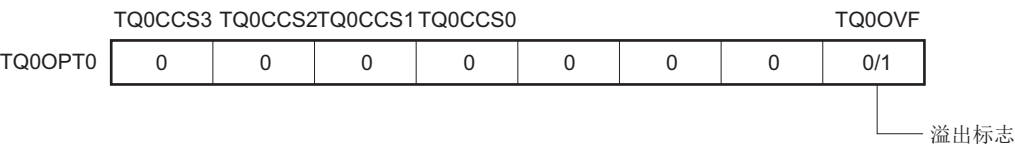


图 7-36. 脉宽测量模式的寄存器设置 (2/2)

(e) **TMQ0 选项寄存器 0 (TQ0OPT0)**



(f) **TMQ0 计数器读取缓冲寄存器 (TQ0CNT)**

可以通过读取 TQ0CNT 寄存器来读取 16 位计数器的计数值。

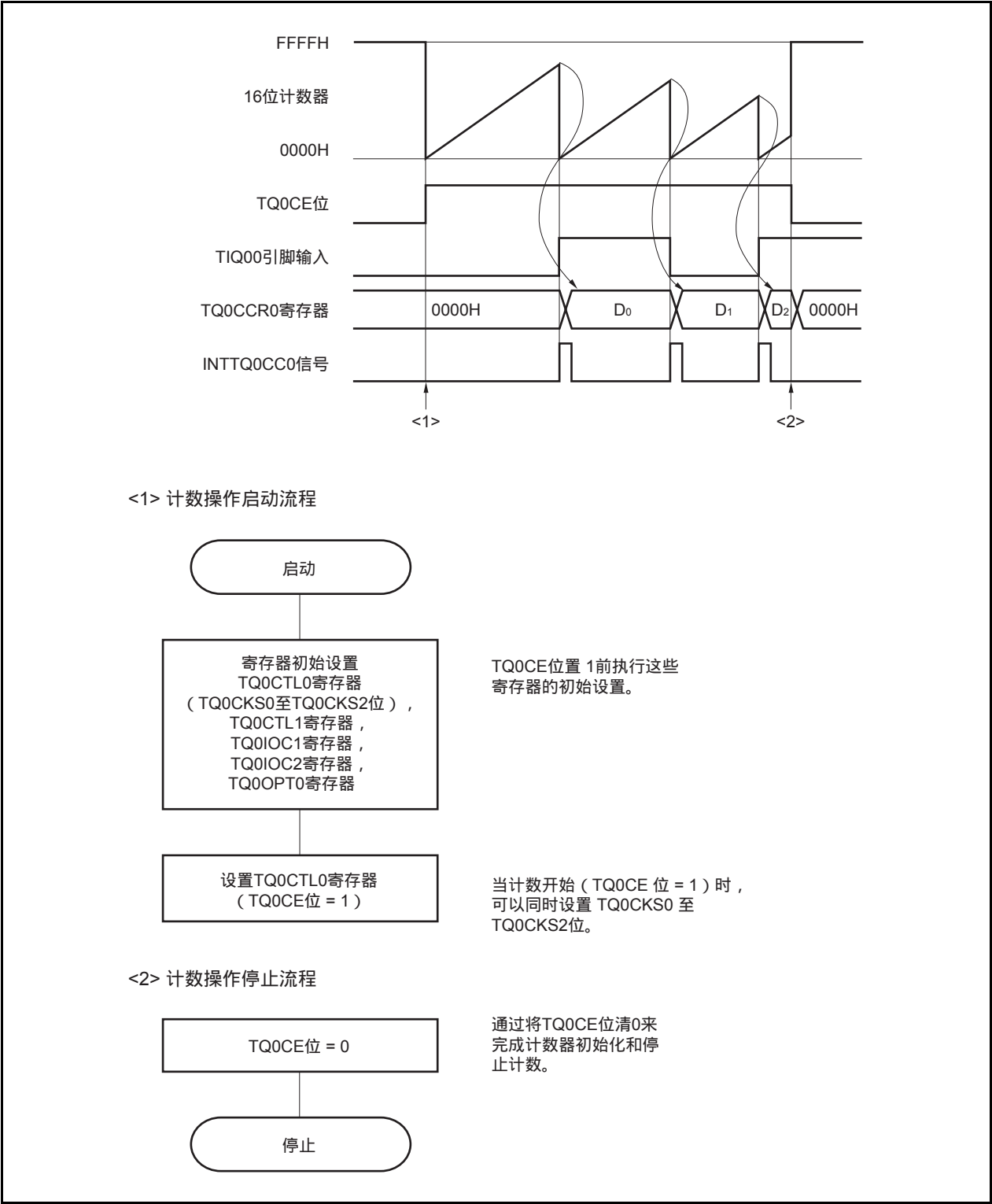
(g) **TMQ0 捕捉/比较寄存器 0 至 3 (TQ0CCR0 至 TQ0CCR3)**

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

- 备注**
- 1. 脉宽测量模式下，并未使用 TMQ0 I/O 控制寄存器 0 (TQ0IOC0)。
 - 2. m = 0 至 3

(1) 脉宽测量模式的操作流程

图 7-37. 脉宽测量模式的软件处理流程

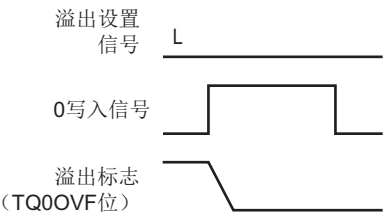


(2) 脉宽测量模式的操作时序

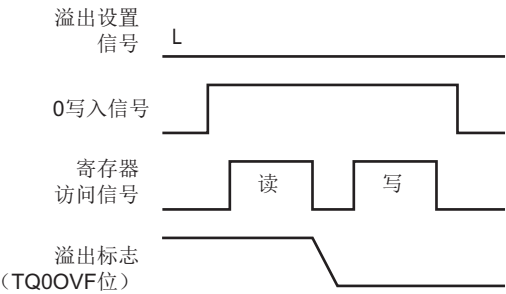
(a) 将溢出标志位清零

可以通过使用 CLR 指令清零 TQ0OVF 位或向 TQ0OPT0 寄存器写入 8 位数据（第 0 位为 0）的方法将溢出标志位清零。为了能够准确检测到溢出的发生，请务必在每次读取到 TQ0OVF 位为 1 后，立即使用位操作指令将该位清零。

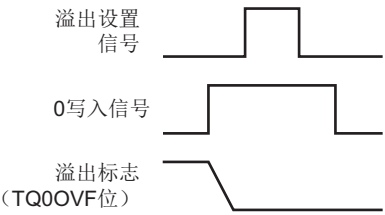
(i) 写0操作（与设置不冲突）



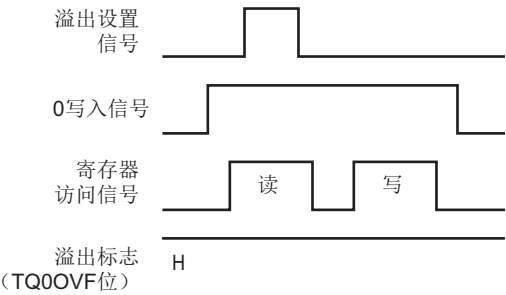
(iii) 清0操作（与设置不冲突）



(ii) 写0操作（与设置冲突）



(iv) 清0操作（与设置冲突）



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

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

7.5.8 三角形波 PWM 模式 (TQ0MD2 至 TQ0MD0 = 111)

在三角形波 PWM 模式中，TMQ0 捕捉/比较寄存器 k (TQ0CCRk) 用于设置占空比系数，而 TMQ0 捕捉/比较寄存器 0 (TQ0CCR0) 则用于设置周期。

通过使用这四种寄存器并操作定时器，将可以输出具有不同周期的三角形波 PWM。

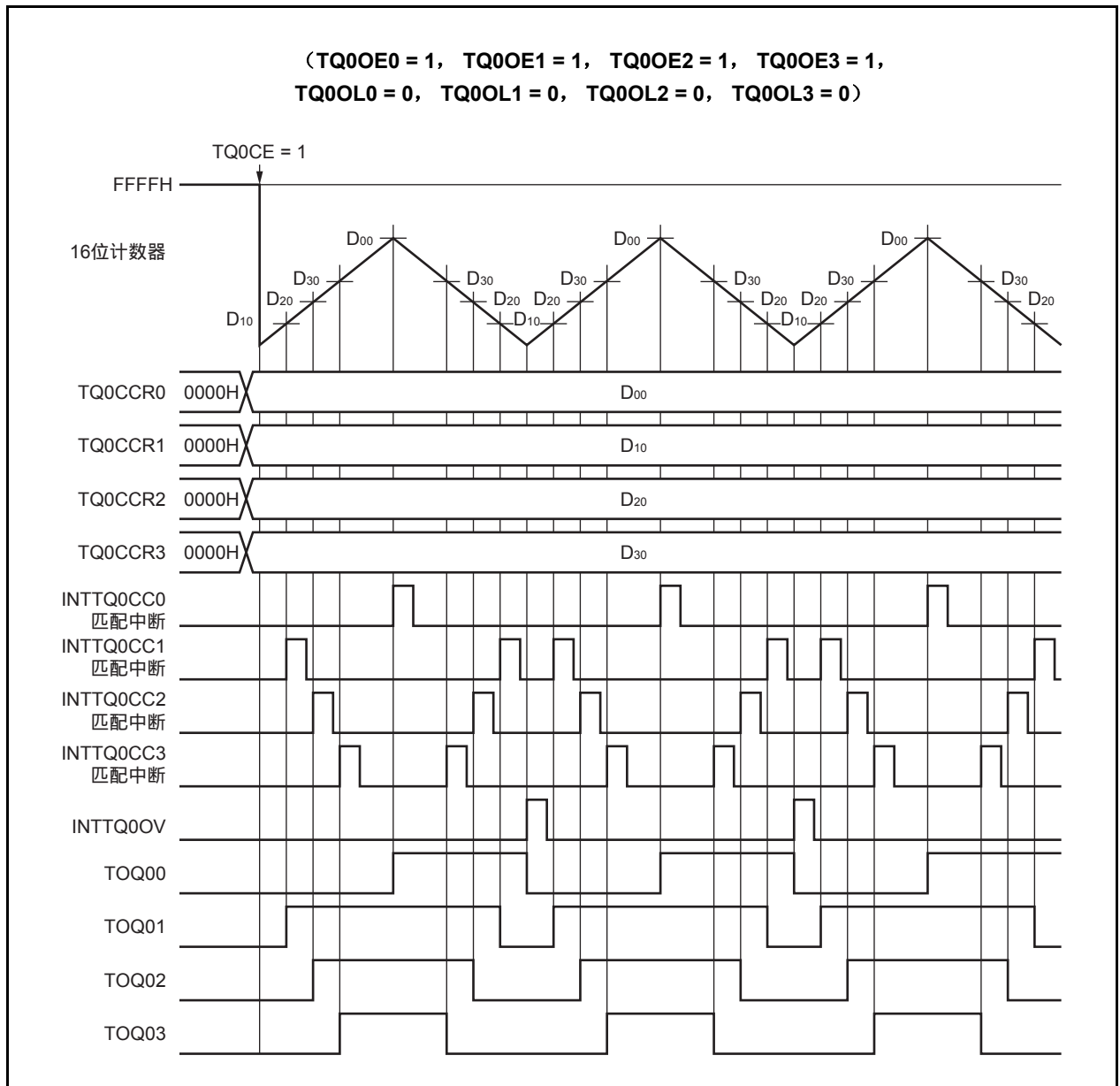
当 TQ0CE = 1 时 TQ0CCRm 寄存器的值可以被改写。

要停止定时器 Q，则将 TQ0CE 清零。PWM 的波形可以从 TOQ0k 引脚中输出。当 16 位计数的值与 TQ0CCR0 寄存器的值匹配且计数器发生下溢时，TOQ00 引脚将会产生触发输出。

注意事项 在 PWM 模式中，由于 TQ0CCRm 寄存器只能被用作比较寄存器，因此它的捕捉功能将不能被使用。

备注 m = 0 至 3, k = 1 至 3

图 7-38. 三角形波 PWM 模式中基本操作的时序



7.5.9 定时器输出操作

TOQ00 至 TOQ03 引脚的工作模式与输出电平的关系如下表所示。

表 7-6. 各模式下的定时器输入控制

操作模式	TOQ00 引脚	TOQ01 引脚	TOQ02 引脚	TOQ03 引脚
间隔定时器模式	方波输出			
外部事件计数模式	方波输出	-		
外部触发脉冲输出模式	方波输出	外部触发脉冲输出	外部触发脉冲输出	外部触发脉冲输出
单脉冲输出模式		单脉冲输出	单脉冲输出	单脉冲输出
PWM 输出模式		PWM 输出	PWM 输出	PWM 输出
自由运行定时器模式	方波输出（仅限使用比较功能时）			
脉宽测量模式	-			
三角形波 PWM 输出模式	方波输出	三角形波 PWM 输出	三角形波 PWM 输出	三角形波 PWM 输出

表 7-7. TOQ00 至 TOQ03 引脚在定时器输出控制位控制下的真值表

TQ0IOC0.TQ0OIm 位	TQ0IOC0.TQ0OEm 位	TQ0CTL0.TQ0CE 位	TOQ0m 引脚的电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前为低电平，计数过程中为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平，计数过程中为低电平

备注 m = 0 至 3

7.6 定时器调谐操作功能

定时器 P 以及定时器 Q 具有定时器调谐操作功能。

表 7-8 中列出了可以同步的定时器。

表 7-8. 定时器的调谐操作模式

主定时器	从定时器	
TMP0	TMP1	—
TMP2	TMP3	TMQ0

- 注意事项**
1. 通过 TPmCTL1.TPmSYE 及 TQ0CTL1.TQ0SYE 位来允许或禁止调谐操作模式。对于 TMQ2 来说，既可以将 TMQ3 或 TMQ0 单独指定为从定时器，也可以将它们同时指定为从定时器。
 2. 通过以下步骤来设置调谐操作模式。
 - <1> 设置从定时器的 TPmCTL1.TPmSYE 及 TQ0CTL1.TQ0SYE 位以允许调谐操作。
将从定时器的 TPmCTL1.TPmMD2 至 TPmCTL1.TPmMD0 位以及 TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位设置为自由运行模式。
 - <2> 利用 TPnCTL1.TPnMD2 至 TPnCTL1.TPnMD0 位来设置定时器模式。
这时，不要对主定时器的 TPnCTL1.TPnSYE 进行设置。
 - <3> 设置主定时器及从定时器的比较寄存器值。
 - <4> 设置从定时器的 TPmCTL0.TPmCE 及 TQ0CTL0.TQ0CE 位以允许操作内置操作时钟。
 - <5> 设置主定时器的 TPnCTL0.TPnCE 位以允许操作内置操作时钟。

备注 m = 1, 3

表 7-9 和 7-10 显示了可以在调谐操作模式中使用的定时器模式（√：可以设置，×：不能设置）。

表 7-9. 可在调谐操作模式中使用的定时器模式

主定时器	自由运行模式	PWM 模式	三角形波 PWM 模式
TMP0	√	√	×
TMP2	√	√	×

表 7-10. 定时器输出功能

调谐通道	定时器	引脚	自由运行模式		PWM 模式		三角形波 PWM 模式	
			调谐关	调谐开	调谐关	调谐开	调谐关	调谐开
Ch0	TMP0 (主)	TOP00	PPG	←	Toggle	←	N/A	←
		TOP01	PPG	←	PWM	←	N/A	←
	TMP1 (从)	TOP10	PPG	←	Toggle	PWM	N/A	←
		TOP11	PPG	←	PWM	←	N/A	←
Ch1	TMP2 (主)	TOP20	PPG	←	Toggle	←	N/A	←
		TOP21	PPG	←	PWM	←	N/A	←
	TMP3 (从)	TOP30	PPG	←	Toggle	PWM	N/A	←
		TOP31	PPG	←	PWM	←	N/A	←
	TMQ0 (从)	TOQ00	PPG	←	Toggle	PWM	Toggle	N/A
		TOQ01 至 TOQ03	PPG	←	PWM	←	三角形波 PWM	N/A

备注 将数据从主定时器的比较寄存器发送至从定时器的比较寄存器的时序如下所示。

PPG:

Toggle, PWM, 三角形波 PWM:

CPU 写入时序。
定时器计数器和比较寄存器与 TOPn0 及 TOQ00 匹配时的时序
(n = 0 至 3)。

图 7-39. 调谐操作图 (TMP2, TMP3, TMQ0)

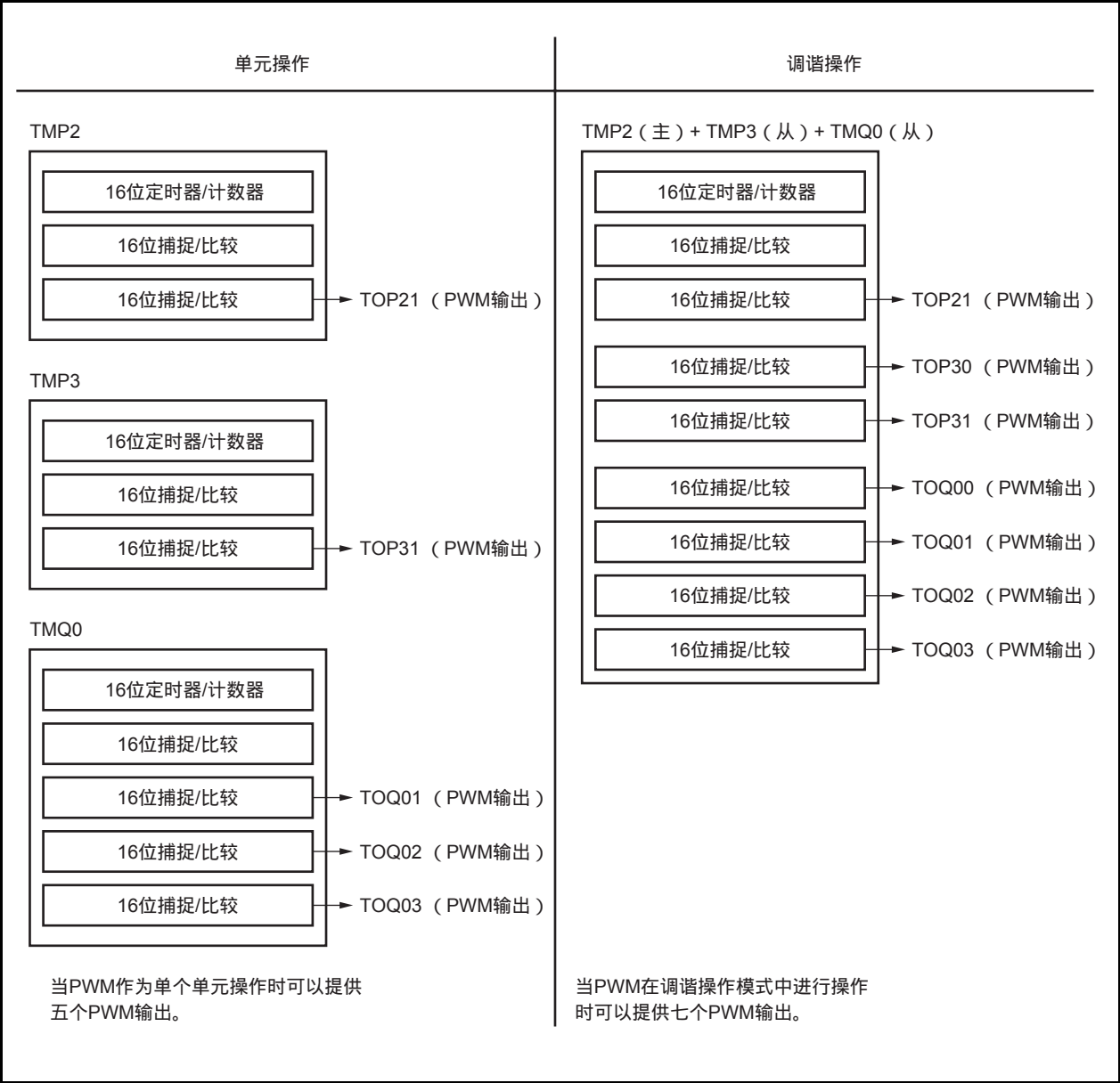
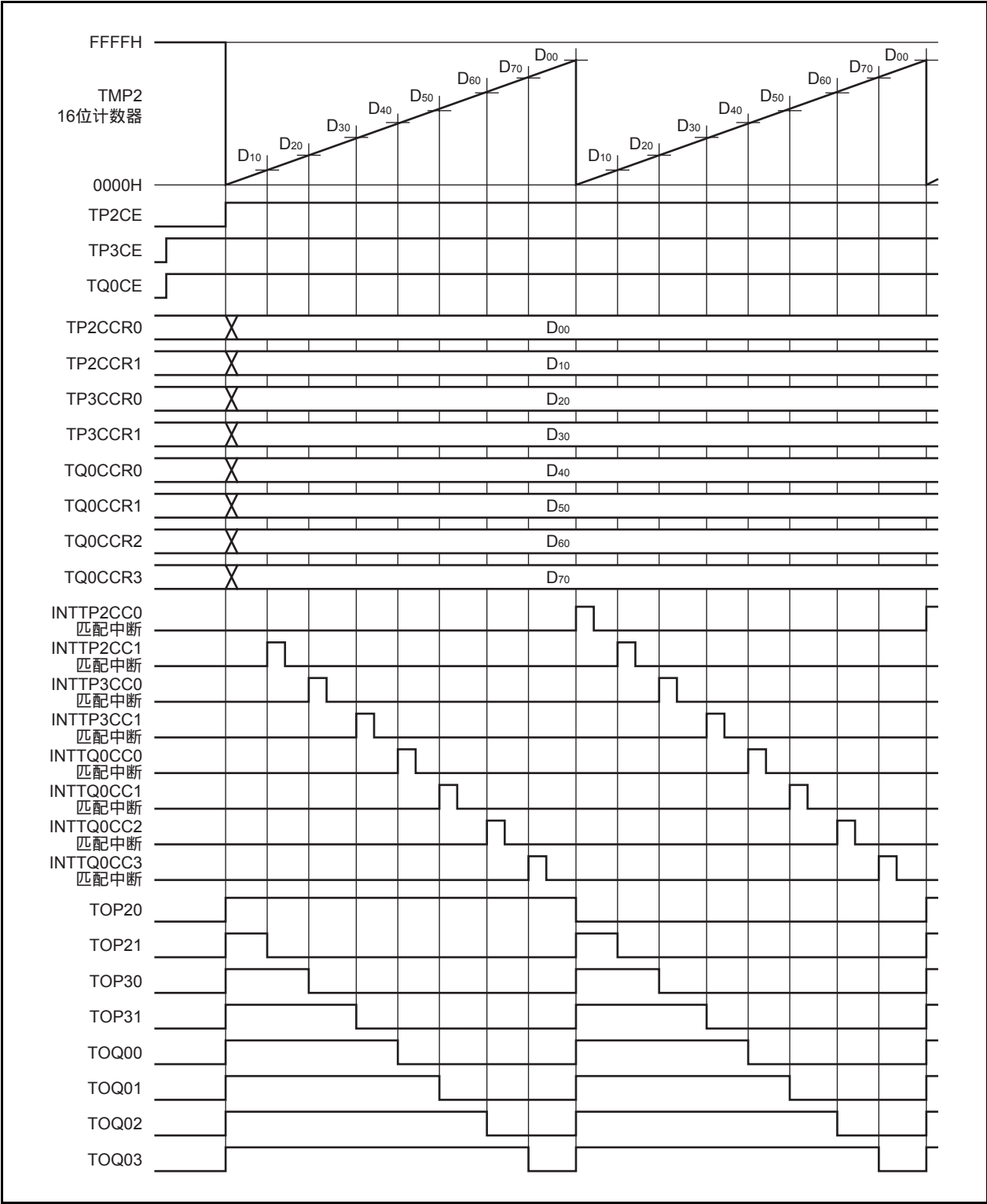


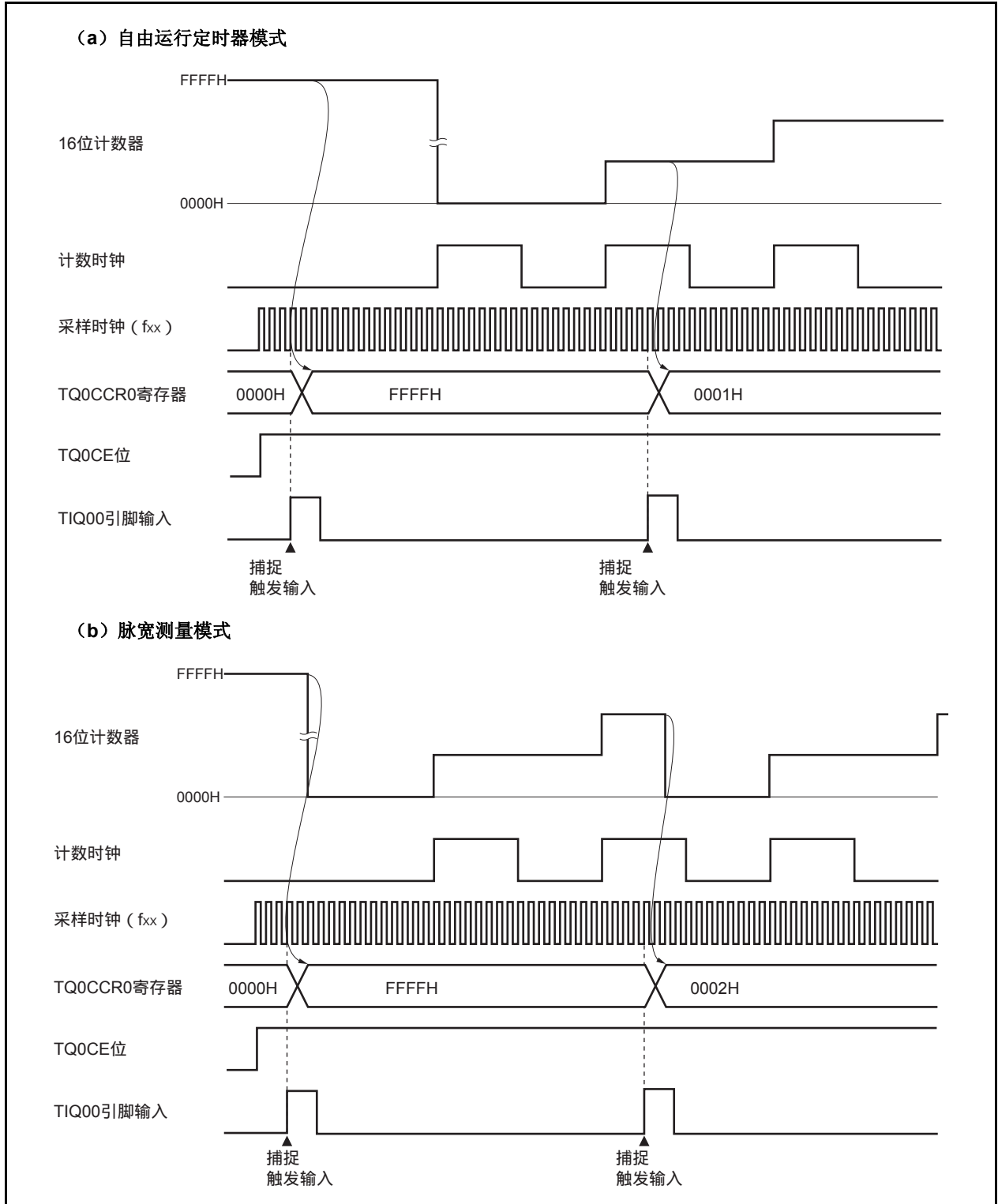
图 7-40. PWM 调谐功能的基本操作时序 (TMP2, TMP3, TMQ0)



7.7 注意事项

(1) 捕捉操作

当使用捕捉操作并且将低速时钟选择为计数时钟时，如果 TQ0CE 位置 1 之后捕捉触发器被立即输入，那么 FFFFH，而不是 0000H，可以被捕捉到 TQ0CCR0，TQ0CCR1，TQ0CCR2 以及 TQ0CCR3 寄存器中。



第 8 章 16 位间隔定时器 M (TMM)

8.1 概述

- 间隔功能
- 8 种时钟可选
- 16 位计数器 × 1
(定时器计数操作过程中不可对 16 位计数器进行读取。)
- 比较寄存器 × 1
(定时器计数操作过程中不可向比较寄存器写入数据。)
- 比较匹配中断 × 1

定时器 M 只支持清除&启动模式。不支持自由运行模式。

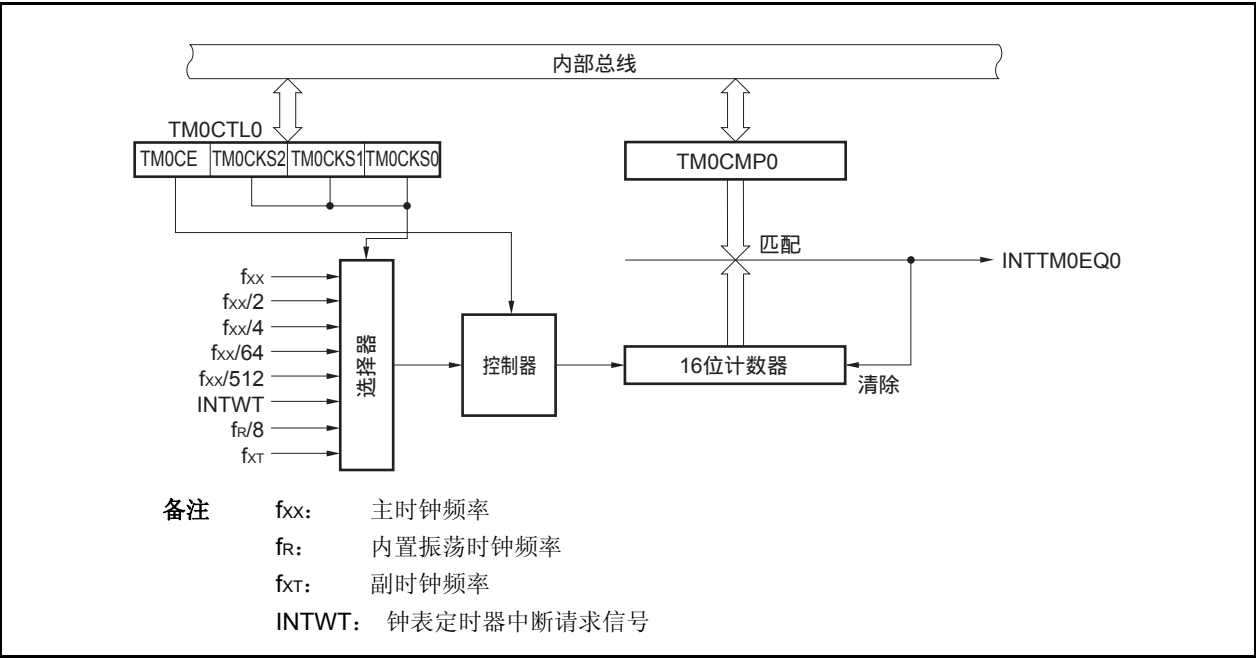
8.2 配置

TMM0 包含以下硬件。

表 8-1. TMM0 的配置

器件	配置
定时器寄存器	16 位计数器
寄存器	TMM0 比较寄存器 0（TM0CMP0）
控制寄存器	TMM0 控制寄存器 0（TM0CTL0）

图 8-1. TMM0 的框图

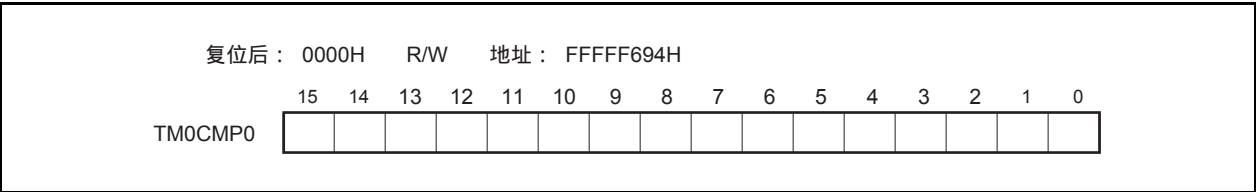


(1) 16 位计数器

该计数器是一个按内部时钟计数的 16 位计数器。
该计数器不可以被读写。

(2) TMM0 比较寄存器 0（TM0CMP0）

TM0CMP0 寄存器是一个 16 位的比较寄存器。
可以对该寄存器进行 16 位的读写操作。
复位输入将把该寄存器设置为 0000H。
可使用软件向 TM0CMP0 寄存器写入相同的数值。
当 TM0CTL0.TM0CE 位 = 1 时禁止改写 TM0CMP0 寄存器的值。



8.3 寄存器

(1) TMM0 控制寄存器 (TM0CTL0)

TM0CTL0 寄存器是用于控制 TMM0 操作的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

可使用软件向 TM0CTL0 寄存器写入相同的数值。定时器运行时，禁止改写该寄存器中除 TM0CE 位外的位。

复位后：00H R/W 地址：FFFF690H

	7	6	5	4	3	2	1	0
TM0CTL0	TM0CE	0	0	0	0	TM0CKS2	TM0CKS1	TM0CKS0

TM0CE	允许/禁止内部时钟操作的指定
0	禁止TMM0操作（16位计数器异步复位）。 停止操作时钟应用。
1	允许TMM0操作。开始操作时钟的应用。TMM0开始工作。
可通过TM0CE位对TMM0的内部时钟控制和内部电路复位进行异步操作。当TM0CE位被清零时，TMM0的内部时钟被停止（固定为低电平）同时16位计数器被异步清零。	

TM0CKS2	TM0CKS1	TM0CKS0	计数时钟选择
0	0	0	f _{xx}
0	0	1	f _{xx} /2
0	1	0	f _{xx} /4
0	1	1	f _{xx} /64
1	0	0	f _{xx} /512
1	0	1	INTWT
1	1	0	f _R /8
1	1	1	f _{XT}

注意事项 1. 在 TM0CE 位 = 0 时设置 TM0CKS2 到 TM0CKS0 位。

当将 TM0CE 的值从 0 改变为 1 时，不可以同时设置 TM0CKS2 到 TM0CKS0 的值。

2. 请务必将第 3 位到第 6 位清除为“0”。

备注

f_{xx}: 主时钟频率

f_R: 内置振荡时钟频率

f_{XT}: 副时钟频率

8.4 操作

注意事项 不可以将 TM0CMP0 寄存器设置为 FFFFH。

8.4.1 间隔定时器模式

在间隔定时器模式中，当 TM0CTL0.TM0CE 位被置 1 后，中断请求信号（INTTM0EQ0）就会按照指定的间隔时间周期性产生。

图 8-2. 间隔定时器的结构图

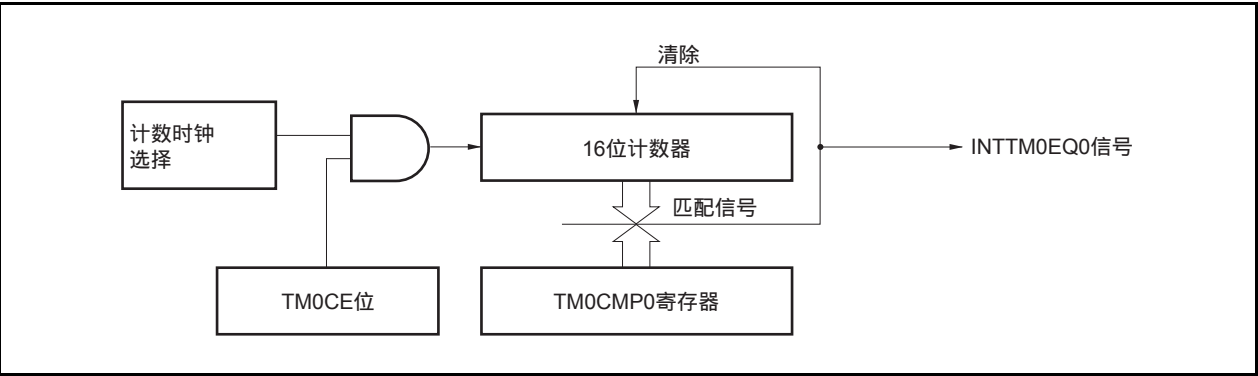
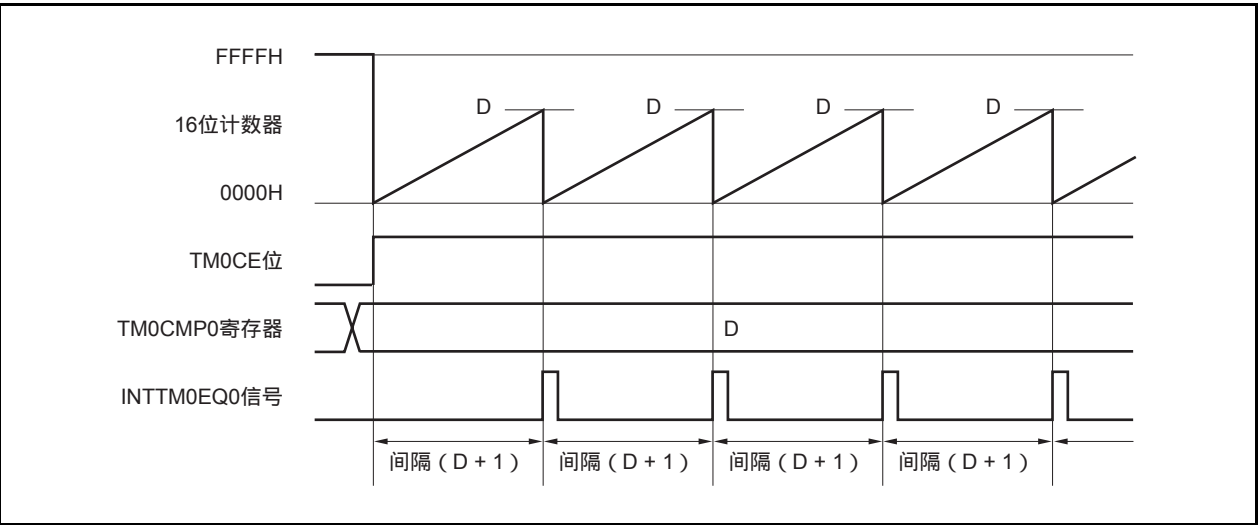


图 8-3. 间隔定时器模式的基本操作时序



当 TM0CE 位被设置为 1，16 位计数器会与计数时钟同步将计数值从 FFFFH 清零为 0000H，并开始计数。

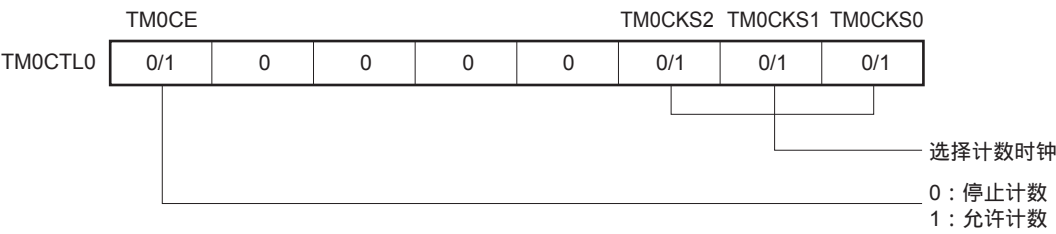
当 16 位计数器的计数值与 TM0CMP0 寄存器中的值相匹配时，16 位计数器被清零为 0000H 并同时产生一个比较匹配中断请求信号（INTTM0EQ0）。

间隔时间可由下列算式求得。

间隔时间 = (TM0CMP0 寄存器的设置值 + 1) × 计数时钟周期

图 8-4. 间隔定时器模式操作的寄存器设置

(a) TMM0 控制寄存器 0 (TM0CTL0)



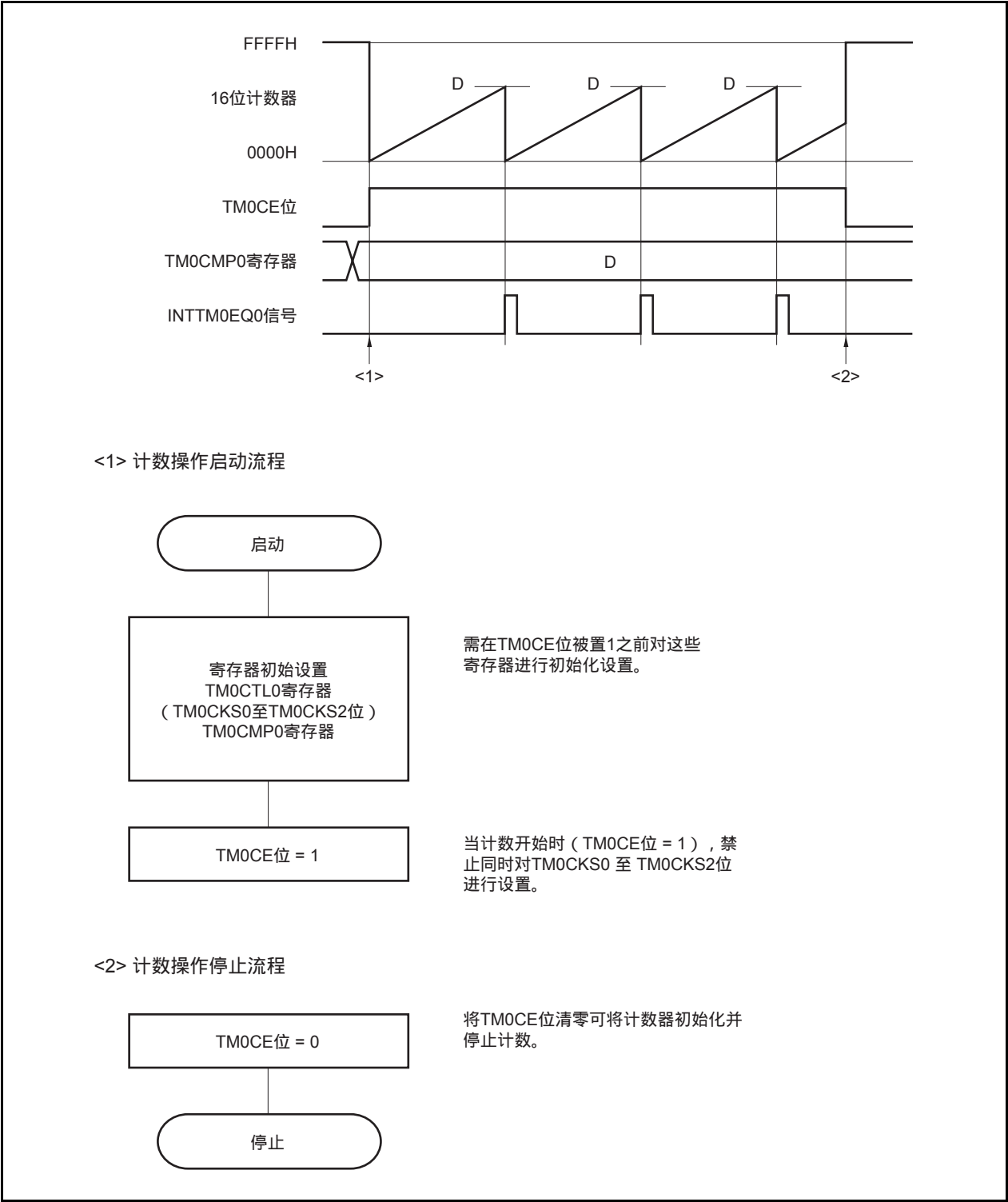
(b) TMM0 比较寄存器 0 (TM0CMP0)

如果 TM0CMP0 寄存器的值被设置为 D，那么间隔时间如下。

间隔时间= (D + 1) × 计数时钟周期

(1) 间隔定时器模式的操作流程

图 8-5. 间隔定时器模式下的软件处理流程

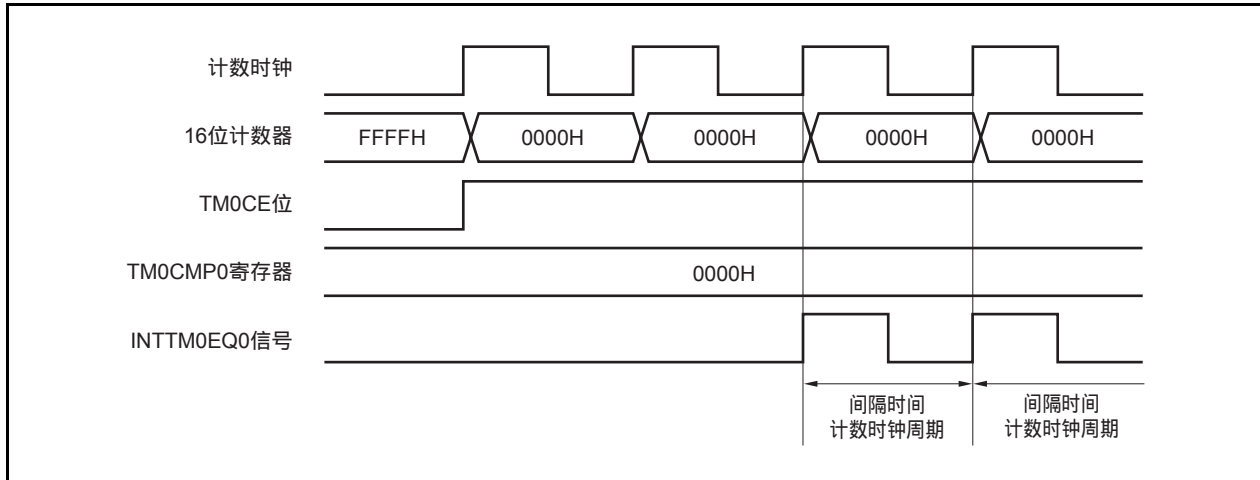


(2) 间隔定时器模式操作时序

注意事项 不要将 TM0CMP0 寄存器设置为 FFFFH。

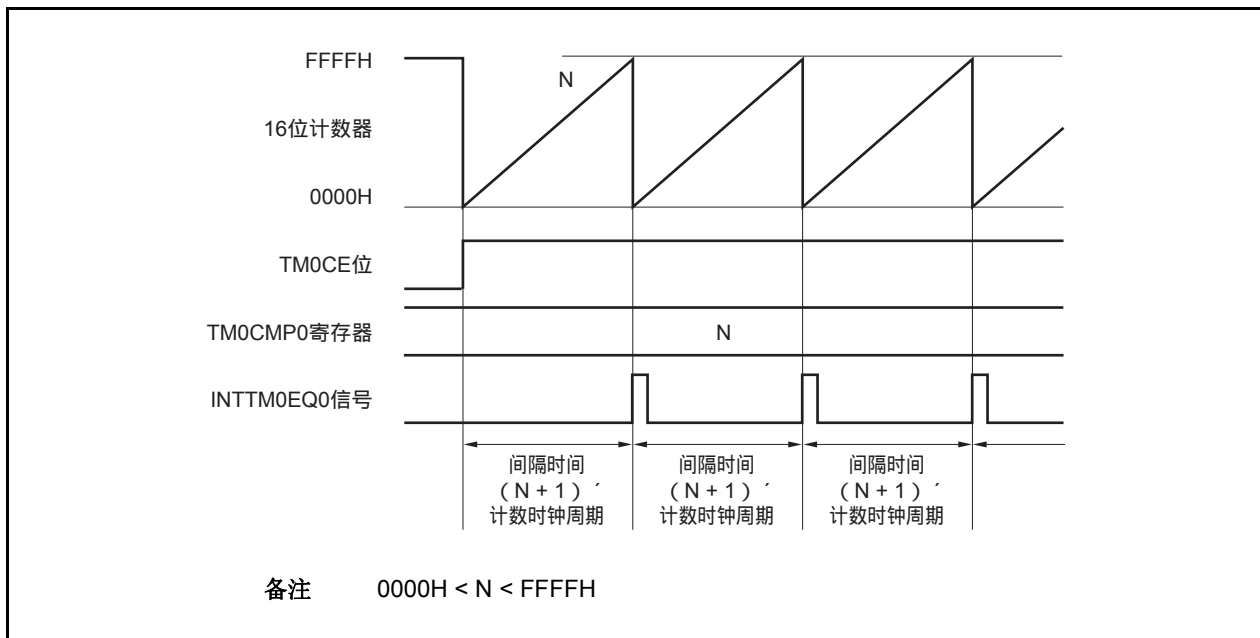
(a) TM0CMP0 寄存器被设置为 0000H 时的操作

如果 TM0CMP0 寄存器被设置为 0000H，那么每个计数时钟到来时刻都会产生 INTTM0EQ0 信号。
16 位计数器的值始终保持为 0000H。



(b) TM0CMP0 寄存器被设置为 N 时的操作

如果 TM0CMP0 寄存器被设置为 N，那么 16 位计数器会累加计数到 N，然后与下一个计数时钟同步被清零并产生 INTTM0EQ0 信号。



8.4.2 注意事项

(1) 根据选择的计数时钟的不同，从 TM0CTL0.TM0CE 位被置 1 到 16 位计数器开始计数的时间间隔如下所示。

所选计数时钟	开始计数前的最大等待时间
f _{XX}	2/f _{XX}
f _{XX} /2	6/f _{XX}
f _{XX} /4	24/f _{XX}
f _{XX} /64	128/f _{XX}
f _{XX} /512	1024/f _{XX}
INTWT	INTWT 信号的第二个上升沿
f _R /8	16/f _R
f _{XT}	2/f _{XT}

(2) TMM0 工作时，禁止改写 TM0CMP0 和 TM0CTL0 寄存器的值。

如果在 TM0CE 位 = 1 时对这两个寄存器进行改写，那么操作结果将不能被保证。
一旦发生上述情况，请先将 TM0CTL0.TM0CE 位清零，之后重置这些寄存器。

第 9 章 钟表定时器功能

9.1 功能

钟表定时器具有以下功能。

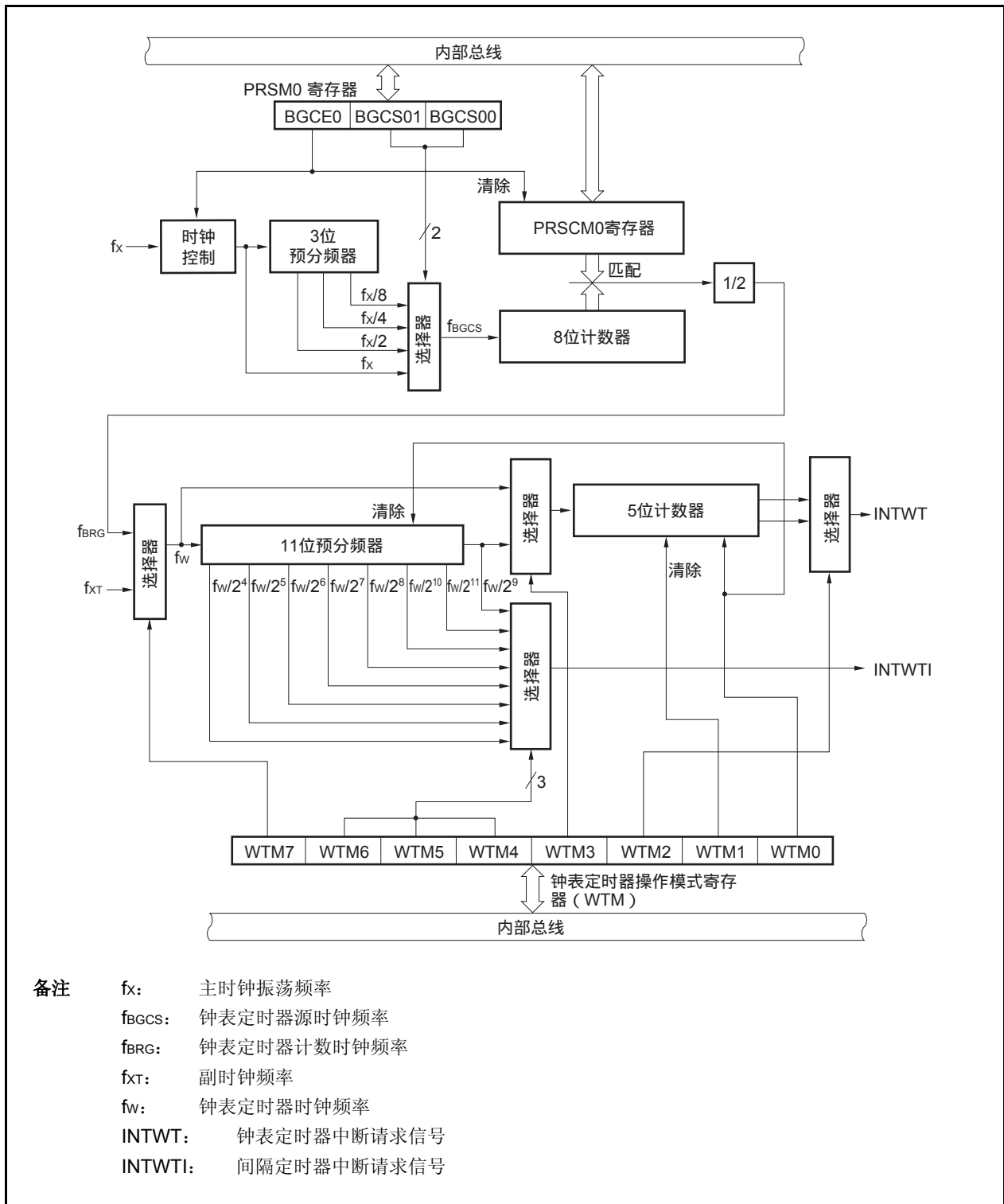
- 钟表定时器： 当使用主时钟或副时钟时，以 0.5 或 0.25 秒的间隔产生中断请求信号（INTWT）。
- 间隔定时器： 以设定的间隔产生中断请求信号（INTWTI）。

钟表定时器和间隔定时器可以同时使用。

9.2 配置

钟表定时器的框图如下。

图 9-1. 钟表定时器的框图



(1) 时钟控制

当钟表定时器工作于主时钟时，该模块用于控制操作时钟（fx）的供给和停止。

(2) 3 位预分频器

fx 被预分频为 $fx/2$ ， $fx/4$ ，或 $fx/8$ 。

(3) 8 位计数器

该 8 位计数器以源时钟（fBGCS）为计数时钟进行计数。

(4) 11 位预分频器

fw 被预分频为 $fw/2^4$ 至 $fw/2^{11}$ 。

(5) 5 位计数器

该计数器以 fw 或 $fw/2^9$ 为计数时钟进行计数，以间隔 $2^4/fw$ ， $2^5/fw$ ， $2^{12}/fw$ ，或 $2^{14}/fw$ 产生一个钟表定时器中断请求信号。

(6) 选择器

钟表定时器有以下 5 种选择器。

- 选择 fx， $fx/2$ ， $fx/4$ 或 $fx/8$ 之一作为钟表定时器源时钟的选择器
- 选择主时钟（fx）或副时钟（fxr）作为钟表定时器时钟的选择器
- 选择 fw 或 $fw/2^9$ 为 5 位计数器的计数时钟频率的选择器
- 选择 $2^4/fw$ ， $2^{13}/fw$ ， $2^5/fw$ 或 $2^{14}/fw$ 作为信号 INTWT 所产生时间间隔的选择器
- 选择 $2^4/fw$ 至 $2^{11}/fw$ 作为间隔定时器中断请求信号（INTWTI）产生时间间隔的选择器。

(7) 寄存器 PRSCM

该寄存器是 8 位比较寄存器，用于设置间隔时间。

(8) 寄存器 PRSM

该寄存器控制提供给钟表定时器的时钟。

(9) 寄存器 WTM

该寄存器是 8 位比较寄存器，用于控制钟表定时器/间隔定时器的操作，并设置中断请求信号产生间隔。

9.3 寄存器

钟表定时器使用下述寄存器。

- 预分频模式寄存器 0（PRSM0）
- 预分频比较寄存器 0（PRSCM0）
- 钟表定时器操作模式寄存器（WTM）

(1) 预分频模式寄存器 0（PRSM0）

该寄存器控制钟表定时器计数时钟的产生。
可对该寄存器进行 1 位或 8 位的读写操作。
复位信号产生将该寄存器设置为 00H。

复位后：00H R/W 地址：FFFFF8B0H

	7	6	5	4	3	2	1	0
PRSM0	0	0	0	BGCE0	0	0	BGCS01	BGCS00

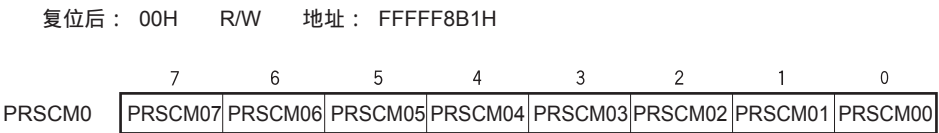
BGCE0	允许主时钟操作	
0	禁止	
1	允许	

BGCS01	BGCS00	钟表定时器源时钟的选择（f _{BGCS} ）		
			5 MHz	4 MHz
0	0	f _x	200 ns	250 ns
0	1	f _x /2	400 ns	500 ns
1	0	f _x /4	800 ns	1 μs
1	1	f _x /8	1.6 μs	2 μs

- 注意事项
1. 钟表定时器操作期间不要改变的 BGCS00 和 BGCS01 位的值。
 2. 在设置 BGCE0 为 1 前，设置寄存器 PRSM0。
 3. 根据主时钟频率，设置寄存器 PRSM0 和 PRSCM0 以获得 32.768 kHz 的 f_{BRG} 频率。

(2) 预分频比较寄存器 0 (PRSCM0)

该寄存器是 8 位比较寄存器。
可对该寄存器进行 8 位的读写操作。
复位信号产生将该寄存器设置为 00H。



- 注意事项
- 1. 钟表定时器操作期间不要对寄存器 **PRSCM0** 进行写入操作。
 - 2. 在设置 **PRSM0.BGCE0** 为 1 前，设置寄存器 **PRCM0**。
 - 3. 根据主时钟频率，设置寄存器 **PRSM0** 和 **PRSCM0** 以获得 32.768 kHz 的 **fBRG** 频率。

fBRG 的计算如下：

$$f_{BRG} = f_{BGCS}/2N$$

备注

fBGCS: 通过寄存器 **PRSM0** 设定的钟表定时器源时钟频率

N: 设置寄存器 **PRSM0** 的值= 1 到 256

但是，仅当 **PRSM0** 的值=00H 时，N = 256。

(3) 钟表定时器操作模式寄存器 (WTM)

该寄存器用于允许/禁止计数时钟以及钟表定时器的操作,设置预分频的间隔时间,控制 5 位计数器的操作,设置钟表标志的确立时间。

在设置寄存器 WTM 前,设置寄存器 PRSM0。

可对该寄存器进行 1 位或 8 位的读写操作。

复位信号产生将该寄存器设置为 00H。

(1/2)

复位后: 00H R/W 地址: FFFFF680H

	7	6	5	4	3	2	1	0
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	WTM6	WTM5	WTM4	预分频器间隔时间的选择
0	0	0	0	$2^4/f_W$ (488 ns: $f_W = f_{XT}$)
0	0	0	1	$2^5/f_W$ (977 ns: $f_W = f_{XT}$)
0	0	1	0	$2^6/f_W$ (1.95 ms: $f_W = f_{XT}$)
0	0	1	1	$2^7/f_W$ (3.91 ms: $f_W = f_{XT}$)
0	1	0	0	$2^8/f_W$ (7.81 ms: $f_W = f_{XT}$)
0	1	0	1	$2^9/f_W$ (15.6 ms: $f_W = f_{XT}$)
0	1	1	0	$2^{10}/f_W$ (31.3 ms: $f_W = f_{XT}$)
0	1	1	1	$2^{11}/f_W$ (62.5 ms: $f_W = f_{XT}$)
1	0	0	0	$2^4/f_W$ (488 ns: $f_W = f_{BRG}$)
1	0	0	1	$2^5/f_W$ (977 ns: $f_W = f_{BRG}$)
1	0	1	0	$2^6/f_W$ (1.95 ms: $f_W = f_{BRG}$)
1	0	1	1	$2^7/f_W$ (3.90 ms: $f_W = f_{BRG}$)
1	1	0	0	$2^8/f_W$ (7.81 ms: $f_W = f_{BRG}$)
1	1	0	1	$2^9/f_W$ (15.6 ms: $f_W = f_{BRG}$)
1	1	1	0	$2^{10}/f_W$ (31.2 ms: $f_W = f_{BRG}$)
1	1	1	1	$2^{11}/f_W$ (62.5 ms: $f_W = f_{BRG}$)

WTM7	WTM3	WTM2	钟表标志设置时间的选择
0	0	0	$2^{14}/f_W$ (0.5 s: $f_W = f_{XT}$)
0	0	1	$2^{13}/f_W$ (0.25 s: $f_W = f_{XT}$)
0	1	0	$2^5/f_W$ (977 ns: $f_W = f_{XT}$)
0	1	1	$2^4/f_W$ (488 ns: $f_W = f_{XT}$)
1	0	0	$2^{14}/f_W$ (0.5 s: $f_W = f_{BRG}$)
1	0	1	$2^{13}/f_W$ (0.25 s: $f_W = f_{BRG}$)
1	1	0	$2^5/f_W$ (977 ns: $f_W = f_{BRG}$)
1	1	1	$2^4/f_W$ (488 ns: $f_W = f_{BRG}$)

WTM1	5位计数器操作的控制
0	操作停止后清除
1	开始

WTM0	允许钟表定时器操作
0	停止操作 (清除预分频器及5位计数器)
1	允许操作

注意事项 当 **WTM0** 和 **WTM1** 均为 0 时才可对 **WTM2** 至 **WTM7** 位进行改写操作。

- 备注
- 1. f_W : 钟表定时器时钟频率
 - 2. 括号中的值应用于 $f_W = 32.768\text{ kHz}$ 时
 - 3. f_{XT} : 副时钟频率
 - 4. f_{BRG} : 钟表定时器计数时钟频率

9.4 操作

9.4.1 钟表定时器的操作

钟表定时器以固定时间间隔产生中断请求信号（INTWT）。使用副时钟（32.768 kHz）或主时钟的 0.5 或 0.25 秒的时间间隔操纵钟表定时器。

当 WTM.WTM1 和 WTM.WTM0 位分别置 1，开始计数操作。当 WTM0 位被清零，11 位预分频器和 5 位计数器被清零，计数操作停止。

当定时器被同时作为钟表定时器和间隔定时器使用时，要通过先清零 WTM1 位再清零 5 位计数器的方法来调整时间。此时，钟表定时器可能产生一个最大 15.6 ms 的误差，而间隔定时器将不受影响。

如果主时钟作为钟表定时器的计数时钟，使用 PRSM0.BGCS01 和 BGCS00 设置计数时钟，使用寄存器 PRSCM0 设置 8 位比较值，以使钟表定时器的计数时钟频率（f_{BRG}）为 32.768 kHz。

当 PRSM0.BGCE0 位置（1），f_{BRG} 被用于钟表定时器。

可以使用下面的表达式计算 f_{BRG}。

$$f_{BRG} = f_x / (2^{m+1} \times N)$$

要设置 f_{BRG} 为 32.768 kHz，进行下面计算，设置 BGCS01 和 BGCS00 位和寄存器 PRSCM0。

- <1> 设置 $N = f_x / 65,536$ 。设置 $m = 0$ 。
- <2> 当 N 的个位的舍入结果为偶数时，在进行舍入之前，请设置 $N = N/2$ ， $m = m + 1$ 。
- <3> 重复步骤 <2> 直到 N 成为奇数或 $m = 3$ 。
- <4> 将舍入后 N 的值（整数部分）设置到 PRSCM0 寄存器，将 m 设置到 BGCS01 和 BGCS00 位中。

示例： 当 $f_x = 4.00 \text{ MHz}$

<1> $N = 4,000,000/65,536 = 61.03\dots$ ， $m = 0$

<2>，<3>由于 N（舍入后的整数部分）为奇数， $N = 61$ ， $m = 0$ 。

<4>设置寄存器 PRSCM0 的值：3DH（61），设置 BGCS01 和 BGCS00 位的值：00

此时，f_{BRG} 的实际频率如下。

$$\begin{aligned} f_{BRG} &= f_x / (2^{m+1} \times N) = 4,000,000 / (2 \times 61) \\ &= 32.787 \text{ kHz} \end{aligned}$$

备注 m：分频值（设置 BGCS01 和 BGCS00 位的值）= 0 到 3

N：设置寄存器 PRSCM0 的值= 1 到 256

但是，仅当 PRSCM0 的值=00H 时，N =00H。

f_x：主时钟振荡频率

9.4.2 间隔定时器的操作

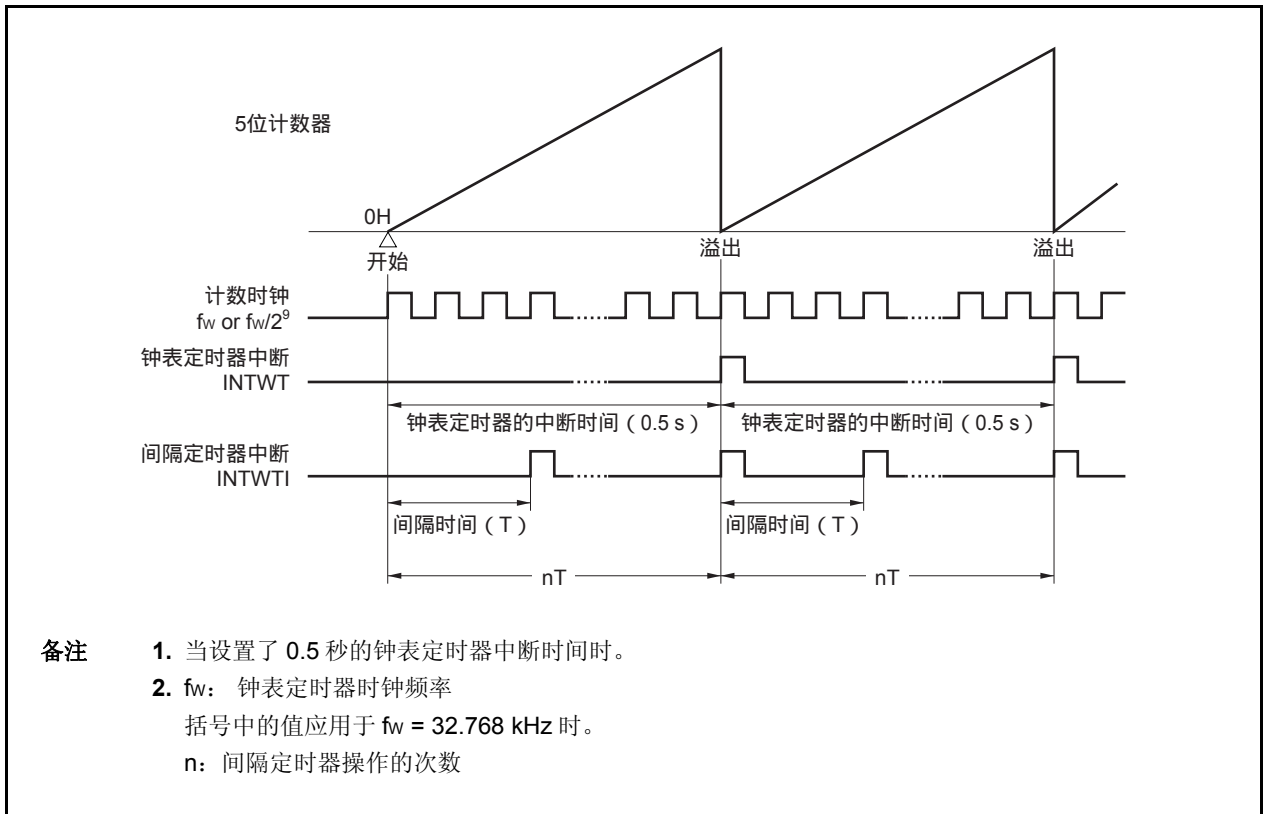
钟表定时器可以用作间隔定时器，并按照预设的数值周期性产生中断请求信号（INTWTI）。
可由寄存器 WTM 的 WTM4 至 WTM7 位选择间隔时间。

表 9-1. 间隔定时器的间隔时间

WTM7	WTM6	WTM5	WTM4	间隔时间	
0	0	0	0	$2^4 \times 1/f_w$	488 μ s （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	0	0	1	$2^5 \times 1/f_w$	977 μ s （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	0	1	0	$2^6 \times 1/f_w$	1.95 ms （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	0	1	1	$2^7 \times 1/f_w$	3.91 ms （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	1	0	0	$2^8 \times 1/f_w$	7.81 ms （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	1	0	1	$2^9 \times 1/f_w$	15.6 ms （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	1	1	0	$2^{10} \times 1/f_w$	31.3 ms （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	1	1	1	$2^{11} \times 1/f_w$	62.5 ms （工作于 $f_w = f_{XT} = 32.768$ kHz）
1	0	0	0	$2^4 \times 1/f_w$	488 μ s （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	0	0	1	$2^5 \times 1/f_w$	977 μ s （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	0	1	0	$2^6 \times 1/f_w$	1.95 ms （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	0	1	1	$2^7 \times 1/f_w$	3.91 ms （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	1	0	0	$2^8 \times 1/f_w$	7.81 ms （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	1	0	1	$2^9 \times 1/f_w$	15.6 ms （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	1	1	0	$2^{10} \times 1/f_w$	31.3 ms （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	1	1	1	$2^{11} \times 1/f_w$	62.5 ms （工作于 $f_w = f_{BRG} = 32.768$ kHz）

备注 f_w : 钟表定时器时钟频率

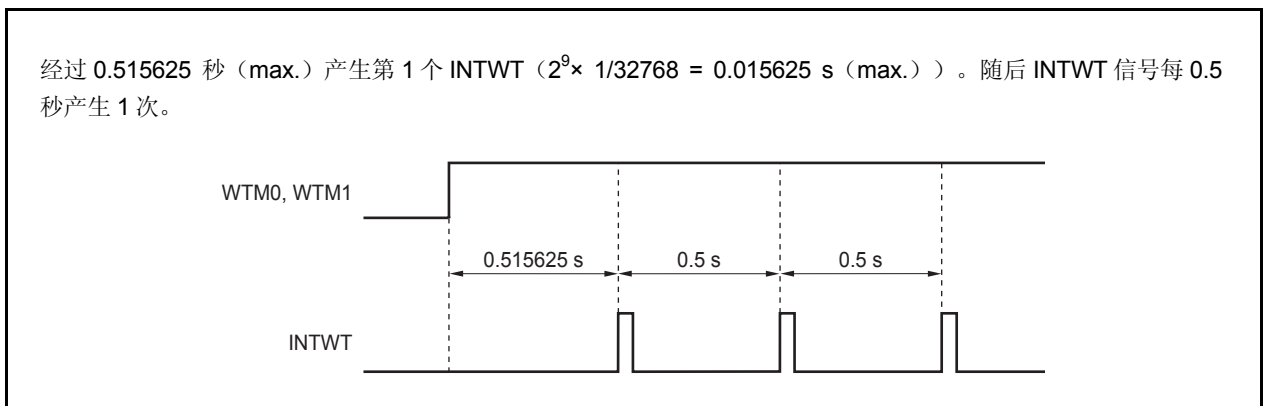
图 9-2. 钟表定时器/间隔定时器的操作时序



9.4.3 注意事项

在允许操作 ($WTM.WTM1$ 和 $WTM.WTM0 = 1$) 后，第一次钟表定时器中断请求信号 ($INTWT$) 产生前需要一段准备时间。

图 9-3. 钟表定时器产生中断请求信号 ($INTWT$) 的示例
(当中断周期= 0.5 s)



第 10 章 看门狗定时器 2 的功能

10.1 功能

看门狗定时器 2 的功能如下。

- 看门狗定时器默认开始工作^{注1}
 - 复位模式：看门狗定时器 2 溢出时复位（产生信号 WDT2RES）
 - 不可屏蔽中断请求模式：看门狗定时器 2 溢出时 NMI 操作（产生信号 INTWDT2）^{注2}
- 源时钟可选主时钟，内部振荡时钟和副时钟

注 1. 复位释放后，看门狗定时器 2 会自动启动。

当不使用看门狗定时器 2，在复位前通过该功能使其停止工作，或清除看门狗定时器 2 并在下一次间隔开始前使其停止工作。

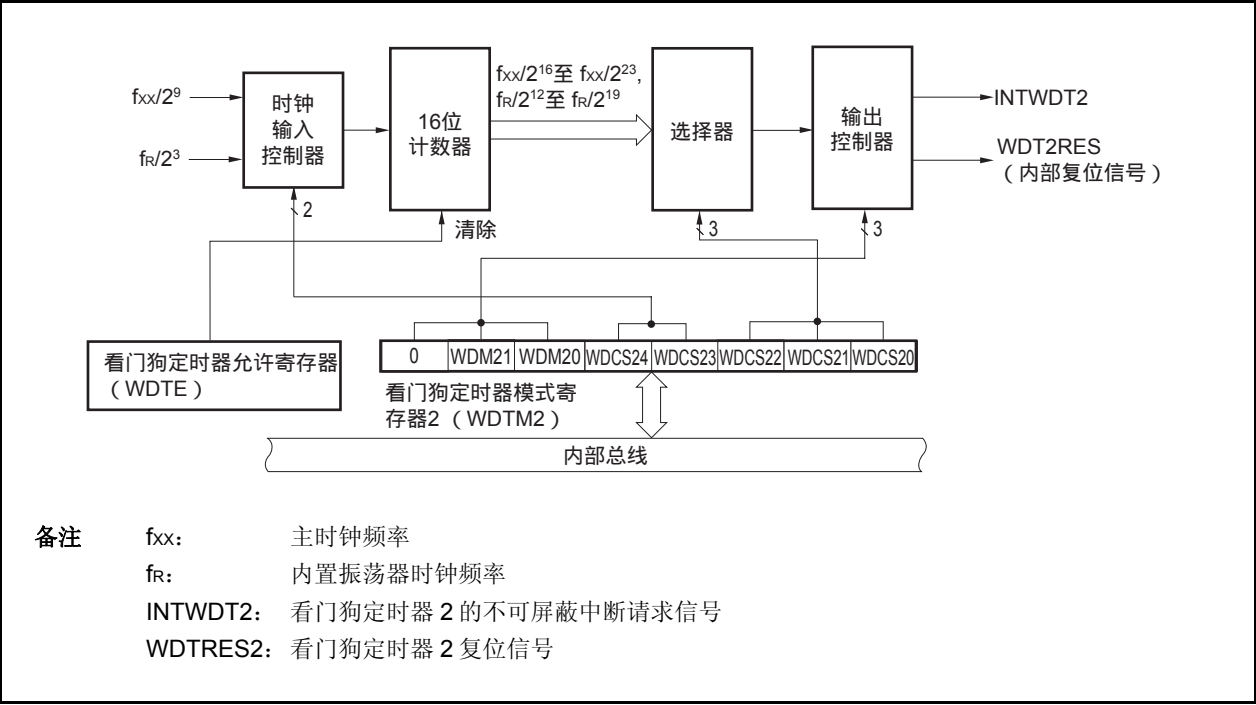
为了确认操作的正确性，即使在不需要改变默认设置（复位模式，间隔时间： $f_R/2^{19}$ ）的情况下，也要对寄存器 WDTM2 执行一次写操作。

2. 由不可屏蔽中断请求信号（INTWDT2）引发的不可屏蔽中断服务，可参见 **14.2.2 （2）信号 INTWDT2**。

10.2 配置

看门狗定时器 2 的框图如下。

图 10-1. 看门狗定时器 2 的框图



看门狗定时器 2 包含以下硬件。

表 10-1. 看门狗定时器 2 的配置

项目	配置
控制寄存器	看门狗定时器模式寄存器 2 (WDTM2) 看门狗定时器使能寄存器 (WDTE)

10.3 寄存器

(1) 看门狗定时器模式寄存器 2 (WDTM2)

该寄存器设置溢出时间和看门狗定时器 2 的操作时钟。
可由 8 位存储器操作指令设置该寄存器。该寄存器可多次读取，但复位释放后只能写入一次。
复位信号产生将该寄存器设置为 67H。

注意事项 下述状态下，禁止访问寄存器 **WDTM2**。详细情况，参见 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。

- 当 CPU 使用副时钟操作，主时钟振荡停止。
- 当 CPU 使用内置振荡时钟进行操作。

复位后： 67H R/W 地址： FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	WDCS24	WDCS23	WDCS22	WDCS21	WDCS20

WDM21	WDM20	看门狗定时器 2 操作模式的选择 ^注
0	0	停止操作
0	1	不可屏蔽中断请求模式 (INTWDT2 信号的生成)
1	—	复位模式 (WDT2RES 信号的生成)

注 如果 OPB1 位通过使用选项字节功能被设为 1（参照第 23 章），那么复位模式将被固定。

- 注意事项**
1. **WDCS20 至 WDCS24** 位的详细信息，请参照表 10-2 看门狗定时器 2 的时钟选择。
 2. 如果寄存器 **WDTM2** 在复位后被复写两次，那么将会强行产生溢出信号且计数器将会复位。
 3. 为了有意产生一个溢出信号，须写入 **WDTM2** 寄存器仅两次或写入一个不是 **ACH** 的值到 **WDTE** 寄存器仅一次。
 4. 要停止看门狗定时器 2 的操作，设置 **RCM.RSTOP** 位为 1（以停止内置晶振）并向寄存器 **WDTM2** 写入 1FH。然而，如果 **OPB1** 位通过使用选项字节功能（参照第 23 章）被设为 1，那么看门狗定时器 2 将不能通过除了复位之外的任务方法来停止。

表 10-2. 看门狗定时器 2 的时钟选择

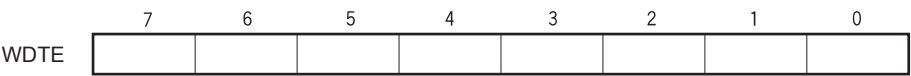
WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	所选择的时钟	100 kHz (MIN.)	200 kHz (TYP.)	400 kHz (MAX.)
0	0	0	0	0	$2^{12}/f_R$	41.0 ms	20.5 ms	10.2 ms
0	0	0	0	1	$2^{13}/f_R$	81.9 ms	41.0 ms	20.5 ms
0	0	0	1	0	$2^{14}/f_R$	163.8 ms	81.9 ms	41.0 ms
0	0	0	1	1	$2^{15}/f_R$	327.7 ms	163.8 ms	81.9 ms
0	0	1	0	0	$2^{16}/f_R$	655.4 ms	327.7 ms	163.8 ms
0	0	1	0	1	$2^{17}/f_R$	1,310.7 ms	655.4 ms	327.7 ms
0	0	1	1	0	$2^{18}/f_R$	2,621.4 ms	1,310.7 ms	655.4 ms
0	0	1	1	1	$2^{19}/f_R$	5,242.9 ms	2,621.4 ms	1,310.7 ms
						f _{xx} = 4 MHz		f _{xx} = 5 MHz
0	1	0	0	0	$2^{16}/f_{xx}$	16.4 ms	13.1 ms	
0	1	0	0	1	$2^{17}/f_{xx}$	32.8 ms	26.2 ms	
0	1	0	1	0	$2^{18}/f_{xx}$	65.5 ms	52.4 ms	
0	1	0	1	1	$2^{19}/f_{xx}$	131.1 ms	104.9 ms	
0	1	1	0	0	$2^{20}/f_{xx}$	262.1 ms	209.7 ms	
0	1	1	0	1	$2^{21}/f_{xx}$	524.3 ms	419.4 ms	
0	1	1	1	0	$2^{22}/f_{xx}$	1,048.6 ms	838.9 ms	
0	1	1	1	1	$2^{23}/f_{xx}$	2,097.2 ms	1,677.7 ms	
1	1	1	1	1	操作停止			

注意事项 如果 **OPB1** 位通过使用选项字节功能被设为 **1**，时钟将会被固定为内置振荡时钟 (f_R)（可以选择 $2^{12}/f_R$ 到 $2^{19}/f_R$ ）。详细信息，请参照第 23 章 选项字节功能。

(2) 看门狗定时器使能寄存器 (WDTE)

将“ACH”写入寄存器 WDTE，则看门狗定时器 2 的计数器被清零，计数重新开始。
可由 8 位存储器操作指令设置该寄存器。
复位信号产生将该寄存器设置为 9AH。

复位后： 9AH R/W 地址： FFFFF6D1H



- 注意事项
- 1. 如果“ACH”以外的数值被写入寄存器 WDTE，必定产生溢出信号。
 - 2. 当向寄存器 WDTE 执行 1 位存储器操作指令，必定产生溢出信号。
 - 3. 为了有意产生一个溢出信号，须写入 WDTM2 寄存器仅两次或写入一个不是“ACH”的值到 WDTE 寄存器仅一次。
 - 4. 读取寄存器 WDTE 的值为“9AH”（与写入值“ACH”不同）。

<R>

10.4 操作

复位释放后，看门狗定时器 2 自动开始运行。

复位后寄存器 WDTM2（使用字节访问操作）只可被写入一次。要使用看门狗定时器 2，使用 8 位操作指令，向寄存器 WDTM2 写入操作模式和间隔时间。之后，将无法停止看门狗定时器 2 的操作。

寄存器 WDTM2 的 WDCS24 至 WDCS20 位被用于选择看门狗定时器 2 循环检测时间间隔。

向寄存器 WDTE 写入 ACH，就会使看门狗定时器 2 的计数器清零，并开始重新计数操作。在计数器开始操作后，循环检测时间间隔内向寄存器 WDTE 写入 ACH。

如果在定时间隔记满时没有对寄存器 WDTE 进行 ACH 写操作，就会根据 WDM21 和 WDTM2.WDM20 的设置值，产生复位信号（WDT2RES）或不可屏蔽中断请求信号（INTWDT2）。

当 WDTM2.WDM21 位设置为 1（复位模式），如果在复位或待机释放后晶振稳定过程中产生 WDT 溢出，将不产生内部复位且 CPU 时钟将转换成内部振荡时钟。

要不使用看门狗定时器 2，向寄存器 WDTM2 写入 1FH。

不可屏蔽中断请求模式下的不可屏蔽中断请求服务可参见 14.2.2（2）通过信号 INTWDT2。

11.1 概述

A/D 转换器用于将模拟输入信号转换为数字信号，具有 10 位分辨率，最多可由 10 个通道（ANI0 至 ANI9）组成。A/D 转换器有以下特点。

- 10 位分辨率
- 通道
- 逐次逼近法
- 操作电压： $AV_{REF0} = 4.0$ 至 5.5 V
- 模拟输入电压： 0 V 至 AV_{REF0}
- 以下为所能提供的操作模式。
 - 连续选择模式
 - 连续扫描模式
 - 单次扫描模式
- 以下功能为所能提供的触发模式。
 - 软件触发模式
 - 外部触发模式（外部，1）
 - 定时器触发模式
- 电压不足监测功能（转换结果比较功能）

11.2 功能

（1）10 位分辨率 A/D 转换

从 ANI0 到 ANI9 选择一个模拟通道，A/D 转换操作以 10 位分辨率重复进行。每当 A/D 转换结束，都产生一个中断请求信号（INTAD）。

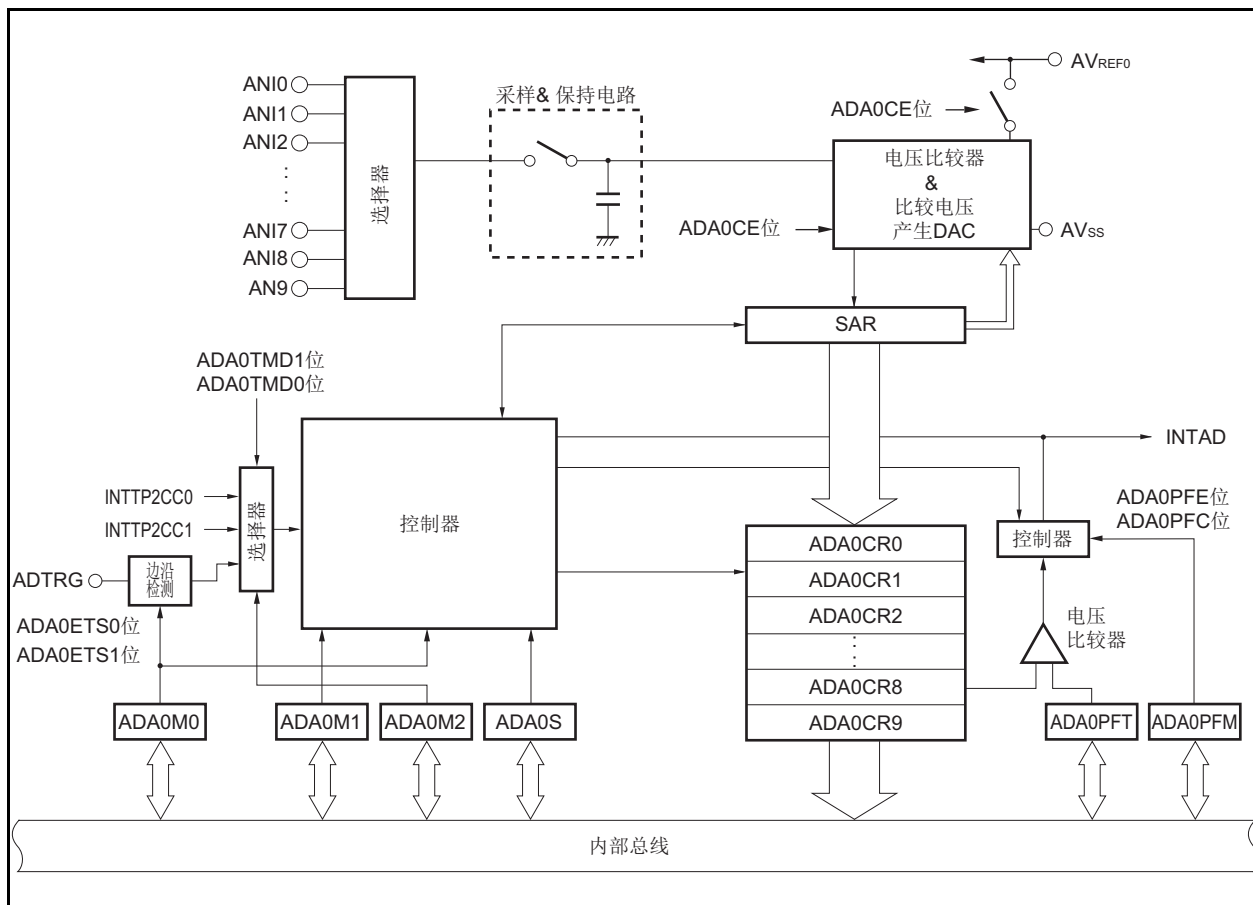
（2）电压不足检测功能

该功能用于检测电池电压的下降。A/D 转换的结果（寄存器 ADA0CRnH 的值）与寄存器 ADA0PFT 的值进行比较，只有当指定的比较条件（ $n = 0$ 至 9 ）满足时，才产生中断信号 INTAD。

11.3 配置

A/D 转换器的框图如下所示。

图 11-1. A/D 转换器的框图



A/D 转换器包括以下硬件。

表 11-1. A/D 转换器的配置

项目	配置
模拟输入	10 通道（ANI0 至 ANI9 引脚）
寄存器	逐次逼近寄存器（SAR） A/D 转换结果寄存器 0 至 9（ADA0CR0 至 ADA0CR9） A/D 转换结果寄存器 0H 至 9H（ADCR0H 至 ADCR9H）：只能读取高 8 位
控制寄存器	A/D 转换器模式寄存器 0 至 2（ADA0M0 至 ADA0M2） A/D 转换器通道选择寄存器 0（ADA0S） 电压不足比较模式寄存器（ADA0PFM） 电压不足比较阈值寄存器（ADA0PFT）

(1) 逐次逼近寄存器 (SAR)

SAR 寄存器将模拟输入信号的电压值和比较电压产生 DAC (比较电压) 的输出电压进行比较, 并从最高有效位 (MSB) 开始保持比较结果。

当比较结果被致以最低有效位 (LSB) (也就是说, 当 A/D 转换完全), 寄存器 SAR 的结果被传送到寄存器 ADA0CRn。

备注 $n = 0$ 至 9

(2) A/D 转换结果寄存器 n (ADA0CRn), A/D 转换结果寄存器 nH (ADA0CRnH)

寄存器 ADA0CRn 是 16 位寄存器, 用于存储 A/D 转换结果。ADA0CRn 由 10 个寄存器组成。根据输入模拟信号, A/D 转换结果存储于 ADA0CRn 的高 10 位。(低 6 位恒为 0。)

(3) A/D 转换器模式寄存器 0 (ADA0M0)

该寄存器设置操作模式并控制 A/D 转换器的转换操作。

(4) A/D 转换器模式寄存器 1 (ADA0M1)

该寄存器设置输入模拟信号被转换的时间。

(5) A/D 转换器模式寄存器 2 (ADA0M2)

该寄存器设置硬件触发模式。

(6) A/D 转换器通道选择寄存器 (ADA0S)

该寄存器设置输入端, 该输入端输入被转换的模拟电压。

(7) 电压不足比较模式寄存器 (ADA0PFM)

该寄存器设置电压不足监测模式。

(8) 电压不足比较阈值寄存器 (ADA0PFT)

寄存器 ADA0PFT 设置一个与 A/D 转换结果寄存器 nH (ADA0CRnH) 比较的阈值。寄存器 ADA0PFT 中所设置的 8 位数据与 A/D 转换结果寄存器 (ADA0CRnH) 比较。

(9) 控制器

当 A/D 转换完成或使用电压不足监测功能时, 控制器将 A/D 转换结果 (ADA0CRnH 的值) 与 ADA0PF 的值进行比较, 只有当指定的比较条件满足时, 才产生中断信号 INTAD。

(10) 采样 & 保存电路

采用与保存电路采集每个模拟输入信号, 发送采样数据到电压比较器。该电路在 A/D 转换期间保持被采样的模拟输入信号的电压。

(11) 电压比较器

电压比较器比较被采样的电压值并保持比较电压产生 DAC 的电压值。

(12) 比较电压产生 DAC

比较电压产生 DAC 连接在 AV_{REF0} 和 AV_{SS} 之间并产生一个电压和模拟输入信号进行比较。

(13) ANI0 至 ANI9 引脚

ANI0 至 ANI9 是 10 个 A/D 转换器通道的引脚，用于输入待转换成数字信号的模拟信号。没有被寄存器 ADA0S 选定作为模拟输入的引脚可以用于普通输入端口。

- 注意事项**
1. 确保输入 ANI0 至 ANI9 的电压不超过额定值。特别是，如果大于或等于 AV_{REF0} 的电压输入某个通道，这个通道的转换结果将不确定，其它通道的转换结果也将受到影响。
 2. 模拟输入引脚（ANI0 至 ANI9）的复用功能与输入商品引脚（P70 至 P79）相同。如果选择 ANI0 至 ANI9 中的任一引脚来执行 A/D 转换，那么请不要在转换期间执行端口 7 的输入指令。如果执行了输入指令，那么转换分辨率将会被降低。

(14) AV_{REF0} 引脚

该引脚用于输入 A/D 转换器的参考电压。即使没有使用 A/D 转换器，也应保持该引脚电压与 V_{DD} 引脚的一致性。输入到引脚 ANI0 至 ANI9 的信号基于引脚 AV_{REF0} 和 AV_{SS} 之间的电压被转换成数字信号。

(15) AV_{SS} 引脚

这是 A/D 转换器的信号地引脚。即便没有使用 A/D 转换器，该引脚的电压也应与 V_{SS} 的电压保持一致。

11.4 寄存器

A/D 转换器使用以下寄存器进行控制。

- A/D 转换器模式寄存器 0, 1, 2 (ADA0M0, ADA0M1, ADA0M2)
- A/D 转换器通道选择寄存器 0 (ADA0S)
- 电压不足比较模式寄存器 (ADA0PFM)

还使用下面的寄存器。

- A/D 转换结果寄存器 n (ADA0CRn)
- A/D 转换结果寄存器 nH (ADA0CRnH)
- 电压不足比较阈值寄存器 (ADA0PFT)

(1) A/D 转换器模式寄存器 0 (ADA0M0)

该寄存器是 8 位寄存器，用于选择操作模式并控制 A/D 转换器的转换操作。可由 1 位或 8 位存储器操作指令读写该寄存器。但是，ADA0EF 位是只读的。

复位使该寄存器清零。

注意事项 下述情况下禁止访问寄存器 **ADA0M0**。详细情况，参见 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。

- 当 CPU 使用副时钟，且主时钟晶振停止振荡
- 当 CPU 使用内部晶振时钟

复位后： 00H R/W 地址： FFFFF200H

	7	6	5	4	3	2	1	0
ADA0M0	ADA0CE	0	ADA0MD1	ADA0MD0	ADA0ETS1	ADA0ETS0	ADA0TMD	ADA0EF

ADA0CE	A/D转换控制
0	停止A/D转换
1	允许A/D转换

ADA0MD1	ADA0MD0	A/D转换操作模式选择
0	0	连接选择模式
0	1	连续扫描模式
1	0	禁止调协
1	1	单脉冲扫描模式

ADA0ETS1	ADA0ETS0	外部触发（引脚ADTRG）输入有效沿选择
0	0	禁止边沿检测
0	1	下降沿检测
1	0	上升沿检测
1	1	上升沿和下降沿检测

ADA0TMD	触发模式选择
0	软件触发模式
1	外部触发模式/定时器触发模式

ADA0EF	A/D转换器状态显示
0	A/D 转换停止
1	A/D 转换正在进行

- 注意事项
1. 0 位的写操作将被忽略。
 2. A/D 转换被允许时（ADA0CE 位= 1）禁止改变 ADA0M1 寄存器的值。
 3. 如果在 A/D 转换（ADA0EF 位= 1）期间 ADA0M0， ADA0M2， ADA0S， ADA0PFM 和 ADA0PFT 寄存器被写入，则将会根据模式执行以下操作。
 - 软件触发模式中
A/D 转换停止并从头开始。
 - 硬件触发模式中
A/D 转换停止并进入触发等待状态。
 4. 当不使用 A/D 转换器时，设置 ADA0CE 位 0 停止操作以减少功耗。
 5. 在 A/D 转换开始后立即输入的引脚的数据第一次转换的分辨率可能会被降低。详细信息，请参见 11.6（7） AV_{REF0} 引脚。

(2) A/D 转换器模式寄存器 1 (ADA0M1)

该寄存器是 8 位寄存器，用于选择转换时间。

可由 1 位或 8 位存储器操作指令读写该寄存器。

复位信号产生将该寄存器清零 (00H)。

复位后: 00H R/W 地址: FFFFF201H							
	7	6	5	4	3	2	1 0
ADA0M1	ADA0HS1	0	0	0	ADA0FR3	ADA0FR2	ADA0FR1 ADA0FR0

注意事项 1. 确保将位 6 至位 4 清除为“0”。

2. 确保将 ADA0HS1 位设为“1”。

备注 A/D 转换时间设置示例，请参见表 11-2。

表 11-2. 转换模式设置示例

ADA0HS1	ADA0FR3 至 ADA0FR0				A/D 转换时间	f _{xx} = 20 MHz	f _{xx} = 16 MHz	f _{xx} = 4 MHz	A/D 稳定时间 ^注
	3	2	1	0					
1	0	0	0	0	31/f _{xx}	禁止设置	禁止设置	7.75 μs	16/f _{xx}
	0	0	0	1	62/f _{xx}	3.10 μs	3.88 μs	15.50 μs	31/f _{xx}
	0	0	1	0	93/f _{xx}	4.65 μs	5.81 μs	禁止设置	47/f _{xx}
	0	0	1	1	124/f _{xx}	6.20 μs	7.75 μs	禁止设置	50/f _{xx}
	0	1	0	0	155/f _{xx}	7.75 μs	9.69 μs	禁止设置	50/f _{xx}
	0	1	0	1	186/f _{xx}	9.30 μs	11.63 μs	禁止设置	50/f _{xx}
	0	1	1	0	217/f _{xx}	10.85 μs	13.56 μs	禁止设置	50/f _{xx}
	0	1	1	1	248/f _{xx}	12.40 μs	15.50 μs	禁止设置	50/f _{xx}
	1	0	0	0	279/f _{xx}	13.95 μs	禁止设置	禁止设置	50/f _{xx}
	1	0	0	1	310/f _{xx}	15.50 μs	禁止设置	禁止设置	50/f _{xx}
	1	0	1	0	341/f _{xx}	禁止设置	禁止设置	禁止设置	50/f _{xx}
	1	0	1	1	372/f _{xx}	禁止设置	禁止设置	禁止设置	50/f _{xx}
	1	1	0	0	403/f _{xx}	禁止设置	禁止设置	禁止设置	50/f _{xx}
	1	1	0	1	434/f _{xx}	禁止设置	禁止设置	禁止设置	50/f _{xx}
	1	1	1	0	465/f _{xx}	禁止设置	禁止设置	禁止设置	50/f _{xx}
	1	1	1	1	496/f _{xx}	禁止设置	禁止设置	禁止设置	50/f _{xx}

注 当 ADA0M0 寄存器的 ADA0CE 位从 0 变为 1 以确保 A/D 转换器的稳定时间时，将会在输入上述时钟值中的一个值后开始第一次 A/D 转换。

<R> **注意事项** 1. 设置 $3.1 \mu\text{s} \leq \text{转换时间} \leq 15.5 \mu\text{s}$ 。

<R> 2. 稳定时间内禁止 ADA0M0，ADA0M2，ADA0S，ADA0PFM 和 ADA0PFT 寄存器的改写，并禁止触发输入。

(3) A/D 转换器模式寄存器 2 (ADA0M2)

该寄存器用于选择硬件触发模式。

可由 1 位或 8 位存储器操作指令读写该寄存器。

复位信号产生将该寄存器清零 (00H)。

复位后：00H R/W 地址：FFFF203H

	7	6	5	4	3	2	1	0
ADA0M2	0	0	0	0	0	0	ADA0TMD1	ADA0TMD0

ADA0TMD1	ADA0TMD0	硬件触发模式选择
0	0	外部触发模式 (当引脚ADTRG有效沿被检测)
0	1	定时器触发模式 0 (当产生中断请求INTTP2CC0)
1	0	定时器触发模式 1 (当产生中断请求INTTP2CC1)
1	1	禁止设置

注意事项 确保将位 7 至位 2 清除为 “0”。

(4) A/D 转换器通道选择寄存器 0 (ADA0S)

该寄存器选择输入模拟电平引脚。该引脚的信号将被转换为数字信号。

可由 1 位或 8 位存储器操作指令读写该寄存器。

复位信号产生将该寄存器清零 (00H)。

复位后： 00H R/W 地址： FFFFF20H								
	7	6	5	4	3	2	1	0
ADA0S	0	0	0	0	ADA0S3	ADA0S2	ADA0S1	ADA0S0

ADA0S3	ADA0S2	ADA0S1	ADA0S0	选择模式	扫描模式
0	0	0	0	ANI0	ANI0
0	0	0	1	ANI1	ANI0， ANI1
0	0	1	0	ANI2	ANI0 至 ANI2
0	0	1	1	ANI3	ANI0 至 ANI3
0	1	0	0	ANI4	ANI0 至 ANI4
0	1	0	1	ANI5	ANI0 至 ANI5
0	1	1	0	ANI6	ANI0 至 ANI6
0	1	1	1	ANI7	ANI0 至 ANI7
1	0	0	0	ANI8	ANI0 至 ANI8
1	0	0	1	ANI9	ANI0 至 ANI9
与上述不同				禁止设置	

(5) A/D 转换结果寄存器 n, nH (ADA0CRn, ADA0CRnH)

该寄存器存储 A/D 转换结果。

该寄存器是只读的，可由 16 位或 8 位存储器操作指令进行读写。但是，由 16 位存储器操作指令访问寄存器 ADA0CRn，8 位存储器操作指令访问寄存器 ADA0CRnH。10 位转换结果存于寄存器 ADA0CRn 的高 10 位，其低 6 位的读取值为 0。转换结果的高 8 位存于 ADA0CRnH。

注意事项 下述状态禁止访问寄存器 ADA0CRn 和 ADA0CRnH。详细情况，参见 3.4.8 (2) 访问特定的片上周边 I/O 寄存器。

- 当 CPU 使用副时钟，且主时钟晶振停止振荡
- 当 CPU 使用内部晶振时钟

复位后：未定义 R 地址： ADA0CR0 FFFFF210H, ADA0CR1 FFFFF212H,
ADA0CR2 FFFFF214H, ADA0CR3 FFFFF216H,
ADA0CR4 FFFFF218H, ADA0CR5 FFFFF21AH,
ADA0CR6 FFFFF21CH, ADA0CR7 FFFFF21EH,
ADA0CR8 FFFFF220H, ADA0CR9 FFFFF222H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADA0CRn	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

复位后：未定义 R 地址： ADA0CR0H FFFFF211H, ADA0CR1H FFFFF213H,
ADA0CR2H FFFFF215H, ADA0CR3H FFFFF217H,
ADA0CR4H FFFFF219H, ADA0CR5H FFFFF21BH,
ADA0CR6H FFFFF21DH, ADA0CR7H FFFFF21FH,
ADA0CR8H FFFFF221H, ADA0CR9H FFFFF223H

	7	6	5	4	3	2	1	0
ADA0CRnH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

备注 n = 0 至 9

注意事项 寄存器 ADA0M0 和 ADA0S 的写操作可能引起 ADA0CRn 内容的不确定。转换后，在写寄存器 ADA0M0 和 ADA0S 前，读取转换结果。如果不按上述操作进行，正确转换结果可能不能读取。

如下为模拟信号输入引脚（ANIO 到 ANI9）和 A/D 转换结果（ADA0CRn 寄存器）的关系。

$$SAR = INT \left(\frac{V_{IN}}{AV_{REF0}} \times 1,024 + 0.5 \right)$$

$$ADA0CR^{\#} = SAR \times 64$$

或，

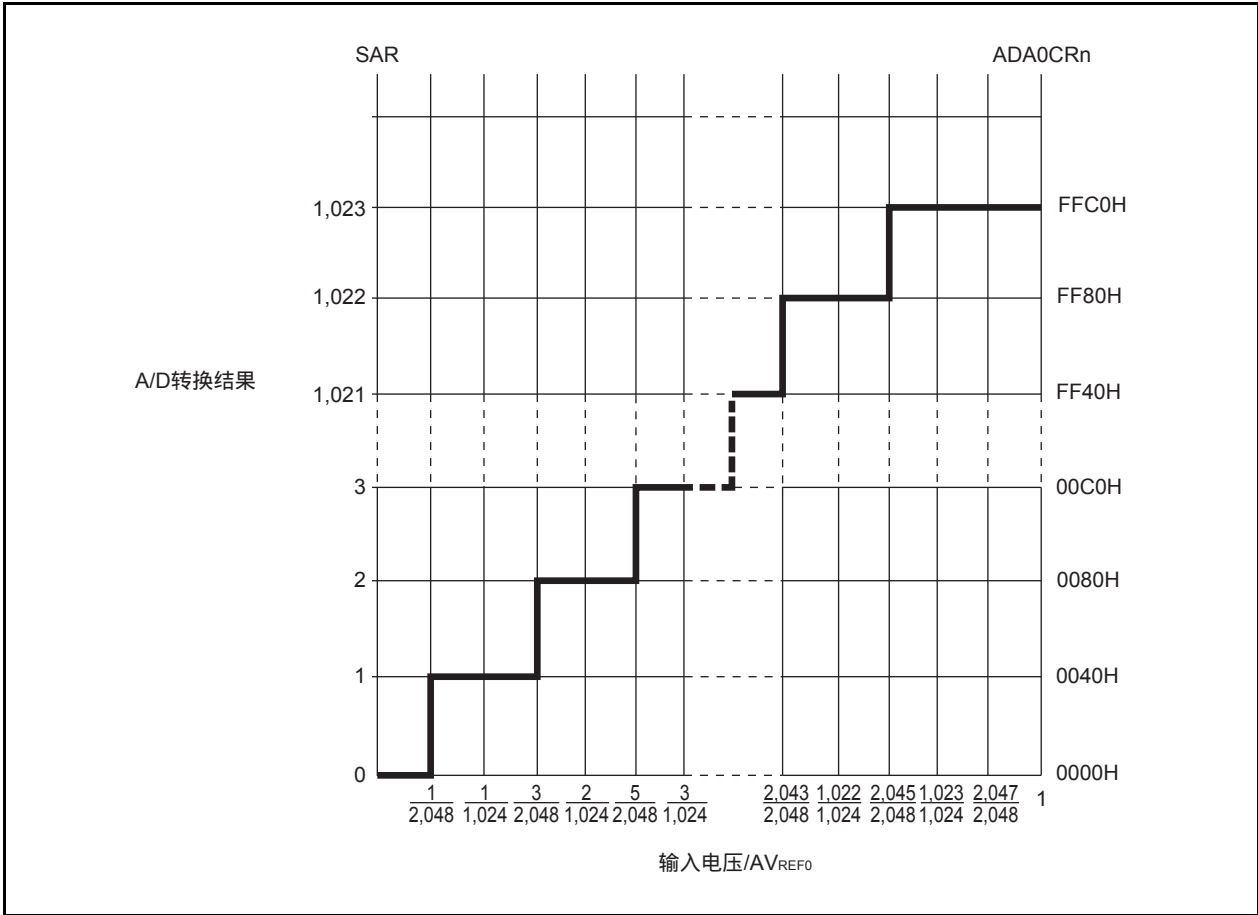
$$(SAR - 0.5) \times \frac{AV_{REF0}}{1,024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{REF0}}{1,024}$$

INT ()：函数，返回 () 中的整数值
VIN：模拟输入电压
AVREF0：AVREF0 引脚电压
ADA0CR：寄存器 ADA0CRn 的值

注 ADA0CRn 的低 6 位恒为 0。

以下显示了模拟输入电压和 A/D 转换结果之间的关系。

图 11-2. 模拟输入电压和 A/D 转换结果之间的关系



(6) 电压不足比较模式寄存器 (ADA0PFM)

该寄存器是 8 位寄存器，用于设置电源电压不足比较模式。

可由 1 位或 8 位存储器操作指令读写该寄存器。

复位使该寄存器清零 (00H)。

复位后：00H R/W 地址：FFFFFF204H

	7	6	5	4	3	2	1	0
ADA0PFM	ADA0PFE	ADA0PFC	0	0	0	0	0	0

ADA0PFE	电压不足比较允许/禁止选择
0	禁止电压不足比较
1	允许电压不足比较

ADA0PFC	电压不足比较模式选择
0	当 $ADA0CRnH \geq ADA0PFT$ 时，产生中断请求信号 (INTAD)
1	当 $ADA0CRnH < ADA0PFT$ 时，产生中断请求信号 (INTAD)

- 注意事项**
1. 选择模式下，寄存器 **ADA0PFT** 所设置的 8 位数据与寄存器 **ADA0CRnH** 的值进行比较，**ADA0CRnH** 由 **ADA0S** 进行设置。如果结果与 **ADA0PFC** 位所设置的条件匹配，转换结果存储于寄存器 **ADA0CRn** 并产生中断信号 **INTAD**。如果不匹配，不产生中断信号 **INTAD**。
 2. 扫描模式下，寄存器 **ADA0PFT** 所设置的 8 位数据与寄存器 **ADA0CR0H** 的值进行比较。如果结果与 **ADA0PFC** 位所设置的条件匹配，转换结果存储于寄存器 **ADA0CR0** 并产生中断信号 **INTAD**。如果不匹配，不产生中断信号 **INTAD**。无论比较结果如何，扫描模式继续工作，转换结果存储于寄存器 **ADA0CRn** 中，直到扫描操作完成。但是，扫描操作完成后不产生中断信号 **INTAD**。

(7) 电压不足比较阈值寄存器 (ADA0PFT)

<R>

与 A/D 转换结果寄存器 nH (ADA0CRnH) 比较时 ADA0PFT 寄存器将会设置阈值。

将在 ADA0PFT 寄存器中设置的 8 位数据与 ADA0CRnH 寄存器 (A/D 转换结果寄存器中的高 8 位) 中的值进行比较。

可由 1 位或 8 位存储器操作指令读写 ADA0PFT 寄存器。

复位使该寄存器清零 (00H)。

复位后：00H R/W 地址：FFFFFF205H

	7	6	5	4	3	2	1	0
ADA0PFT								

11.5 操作

11.5.1 基本操作

- <1> 使用寄存器 ADA0M0, ADA0M1, ADA0M2, 和 ADA0S 设置操作模式, 触发模式, 和执行 A/D 的转换时间。当寄存器的 ADA0CE 位被设置, 软件触发模式下转换开始, 而外部/时间触发模式下等待触发, 才开始 A/D 转换。
- <2> A/D 转换开始后, 采样和保持电路对被选模拟输入通道的输入电压进行采样。
- <3> 当采样和保持电路采集输入信号到指定状态, 电路进入保持状态, 保持输入电压知道 A/D 转换完成。
- <4> 设置逐次逼近寄存器 (SAR) 的第 9 位来设置比较电压生成 DAC 为 $(1/2) AV_{REF0}$ 。
- <5> 比较电压生成 DAC 和模拟输入电压之间的电压差由电压比较器来比较。如果模拟输入电压高于 $(1/2) AV_{REF0}$, 则 SAR 寄存器的 MSB 保持设置。如果模拟输入电压低于 $(1/2) AV_{REF0}$, 则 SAR 寄存器的 MSB 复位。
- <6> 接着, SAR 寄存器的第 8 位被自动设置并且启动下一次比较。根据已经设置好的第 9 位的值, 比较电压生成 DAC 选择如下。
 - 位 9 = 1: $(3/4) AV_{REF0}$
 - 位 9 = 0: $(1/4) AV_{REF0}$这个比较电压与模拟输入电压进行比较, 根据结果, 设置 SAR 的第 8 位, 如下所示。
模拟输入电压 \geq 比较电压: 第 8 位 = 1
模拟输入电压 \leq 比较电压: 第 8 位 = 0
- <7> 按此方式继续进行比较, 直至 SAR 的第 0 位。
- <8> 全部 10 位比较完成后, 在 SAR 中保留一个有效的数值结果, 然后将结果传送至 A/D 转换结果寄存器 ADA0CRn 中。同时也会产生 A/D 转换结束中断请求 (INTAD)。

11.5.2 触发模式

通过设置触发模式确定开始转换操作的时序。触发模式包括软件触发模式和硬件触发模式。硬件触发模式包括定时器触发模式 0 和 1，和外部触发模式。ADA0M0.ADA0TMD 用于设置 ADA0M0.ADA0TMD。ADA0M2.ADA0TMD1 和 ADA0M2.ADA0TMD0 位用于设置硬件 ADA0M0.ADA0TMD。

(1) 软件触发模式

当 ADA0M0.ADA0CE = 1，由寄存器 ADA0S 设置的模拟输入引脚（ANI0 至 ANI9）的信号被转换。当转换完成，结果存储于寄存器 ADA0CRn 中。同时，产生 A/D 转换结束中断请求信号（INTAD）。

如果 ADA0M0.ADA0MD1 和 ADA0M0.ADA0MD0 设置的操作模式是连续选择/扫描模式，下一次转换开始，除非第一次转换完成后设置 ADA0CE = 0。

当转换开始，ADA0M0.ADA0EF = 1（表明转换正在进行）。如果转换过程中对寄存器 ADA0M0，ADA0M2，ADA0S，ADA0PFM，或 ADA0PFT 进行写操作，转换中止并重新开始。

(2) 外部触发模式

该模式下，当外部触发信号输入到引脚 ADTRG，由寄存器 ADA0S 设置的模拟输入引脚（ANI0 至 ANI9）的信号开始转换。ADA0M0.ADA0ETS1 和 ADA0M0.ADA0ETS0 用于设置外部触发的检测边缘（即，上升沿，下降沿，或上升和下降沿）。

无论是否通过 ADA0MD1 和 ADA0MD0 位将连续选择，连续扫描或单次扫描模式设置为操作模式，当转换完成时，结果将被存储在寄存器 ADA0CRn 中。同时，INTAD 信号将被生成，且 A/D 转换器将再次等待下一次触发。

当转换开始，ADA0EF 位将被设为 1（表明转换正在进行）。然而，当 A/D 转换器等待触发时，ADA0EF 位将会被清 0（表明转换停止）。如果在转换过程中输入有效触发，转换将会中止并重新开始。

如果转换过程中对寄存器 ADA0M0，ADA0M2，ADA0S，ADA0PFM 或 ADA0PFT 进行写操作，转换并不会中止，而 A/D 转换器将等待再次触发。

(3) 定时器触发模式

该模式下，当连接到定时器的捕捉/比较寄存器产生比较匹配中断请求信号（INTTP2CC0 或 INTTP2CC1）时，由寄存器 ADA0S 设置的模拟输入引脚（ANI0 至 ANI9）的信号开始转换。使用 ADA0TMD1 和 ADA0TMD0 选择信号 INTTP2CC0 或 INTTP2CC1，在特定的比较匹配中断请求信号上升沿，转换开始。当 ADA0CE 被设为 1 时，A/D 转换器等待触发，并在比较匹配中断请求信号输入时开始转换。

无论是否通过 ADA0MD1 和 ADA0MD0 位将连续选择，连续扫描或单次扫描模式设置为操作模式，当转换完成，结果将存储在寄存器 ADA0CRn 中。同时产生 INTAD 信号，A/D 转换器等待再次触发。

当转换开始，ADA0EF 将被设为 1（表明转换正在进行）。然而，当 A/D 转换器等待触发时，ADA0EF 将会被清 0（表明转换停止）。如果在转换过程中输入有效触发，转换中止并重新开始。

如果转换过程中对寄存器 ADA0M0，ADA0M2，ADA0S，ADA0PFM 或 ADA0PFT 进行写操作，转换将中止，A/D 转换器等待再次触发。

11.5.3 操作模式

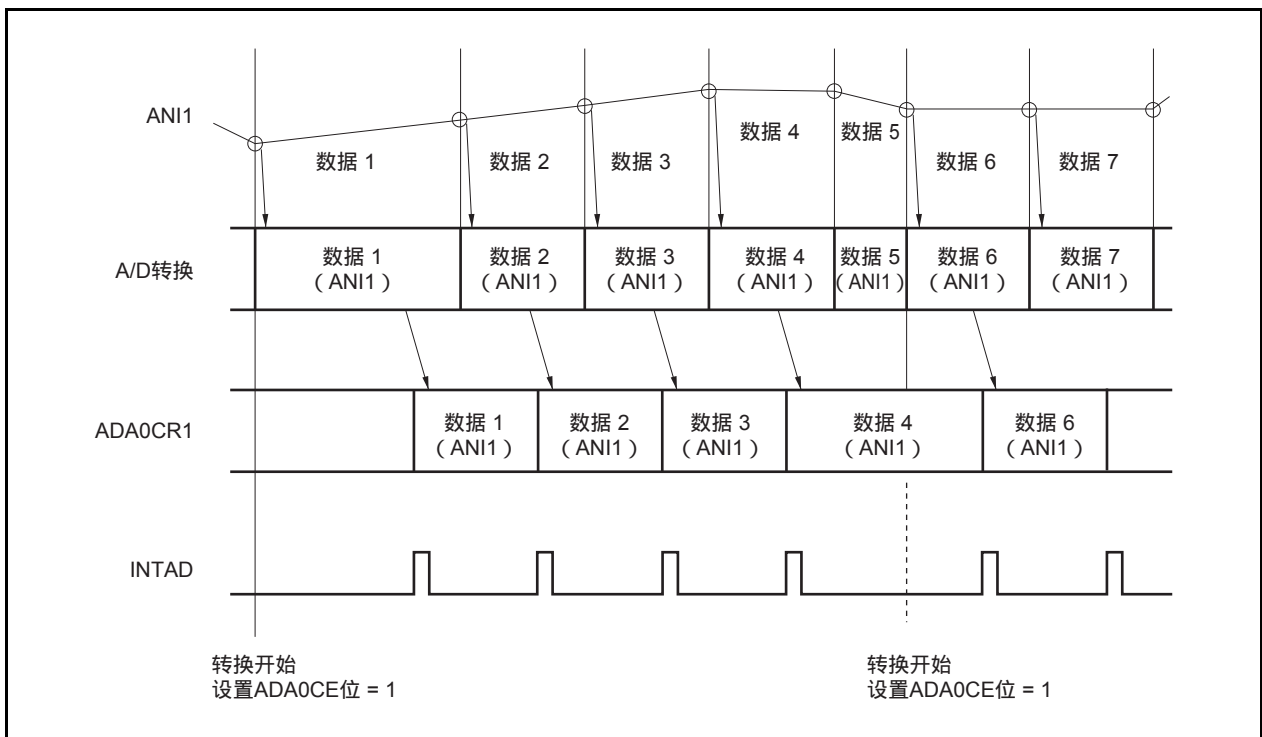
有三种操作模式可用来设置 ANI0 到 ANI9 引脚。连续选择模式，连续扫描模式和单次扫描模式。
使用 ADA0M0.ADA0MD1 和 ADA0M0.ADA0MD0 设置操作模式。

(1) 连续选择模式

该模式下，由寄存器 ADA0S 选择的某个模拟输入引脚的电压连续转换成数字量。

相应模拟输入引脚的转换结果存储于寄存器 ADA0CRn 中。该模式下，模拟输入引脚与寄存器 ADA0CRn 一一对应。每次 A/D 转换完成，产生 A/D 转换结束中断请求信号（INTAD）。转换完成后，除非将 ADA0M0.ADA0CE 位清 0（n = 0 到 0），否则下一次转换开始。

图 11-3. 连续选择模式的操作时序示例（寄存器 ADA0S = 01H）



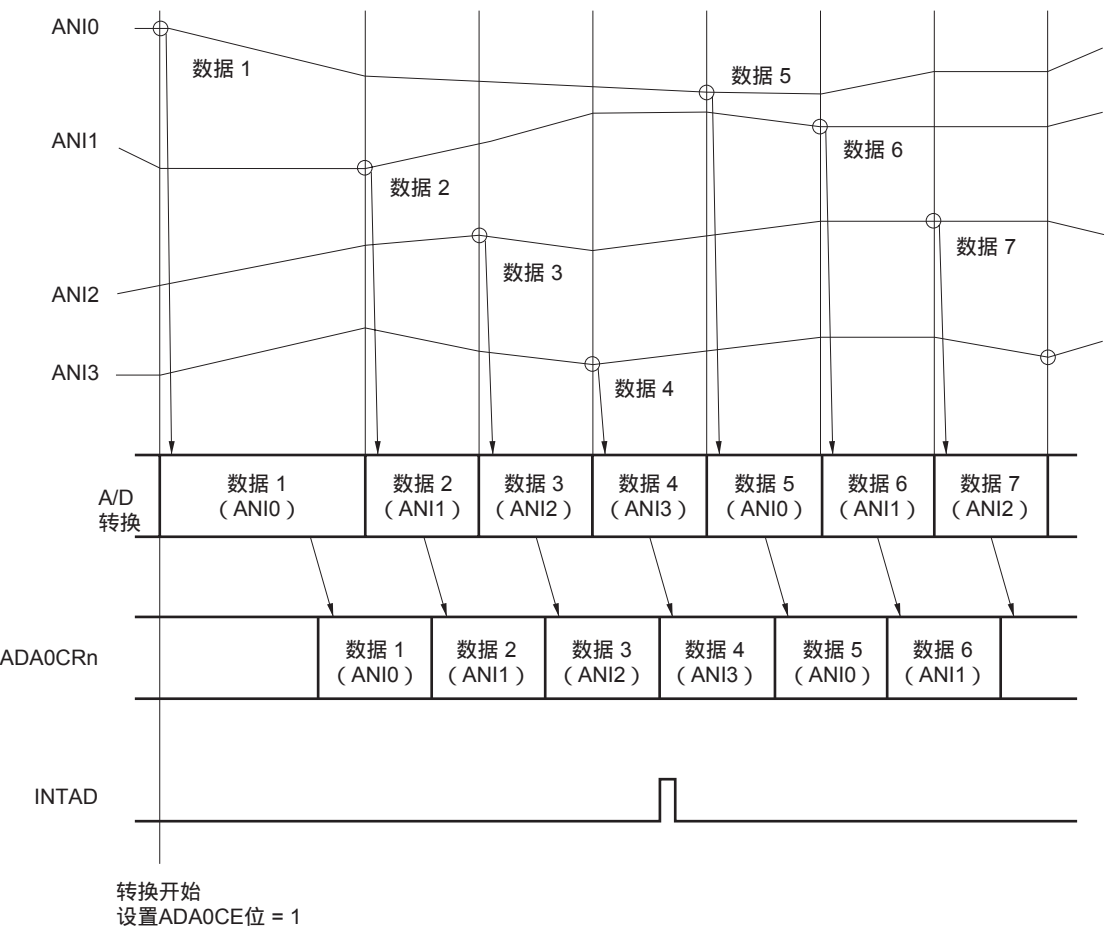
(2) 连续扫描模式

该模式下，模拟输入引脚被依次选择（从引脚 ANI0 到寄存器 ADA0S 所指定的引脚），这些模拟输入值被转换成数字值。

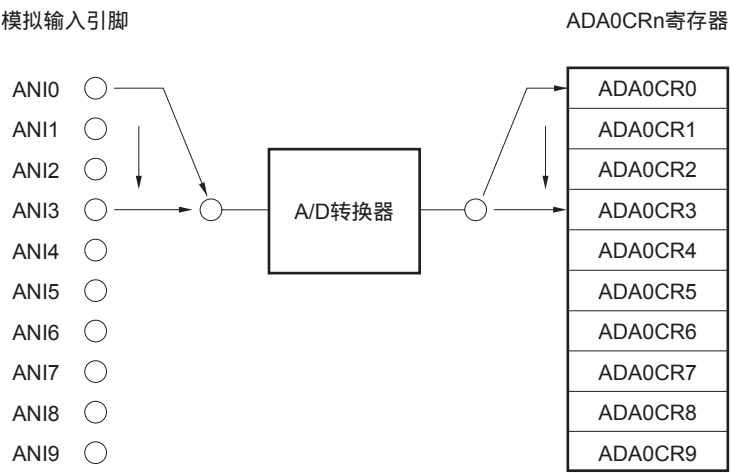
转换结果保存在与相应模拟输入管脚对应的 ADA0CRn 寄存器中。当由 ADA0S 寄存器指定的模拟输入引脚的转换完成之后，会产生 INTAD 信号，然后从 ANI0 引脚的 A/D 转换重新开始，除非 ADA0CE 位被清零（n = 0 至 9）。

图 11-4. 连续扫描模式的操作时序示例（寄存器 ADA0S = 03H）

(a) 时序示例



(b) 框图

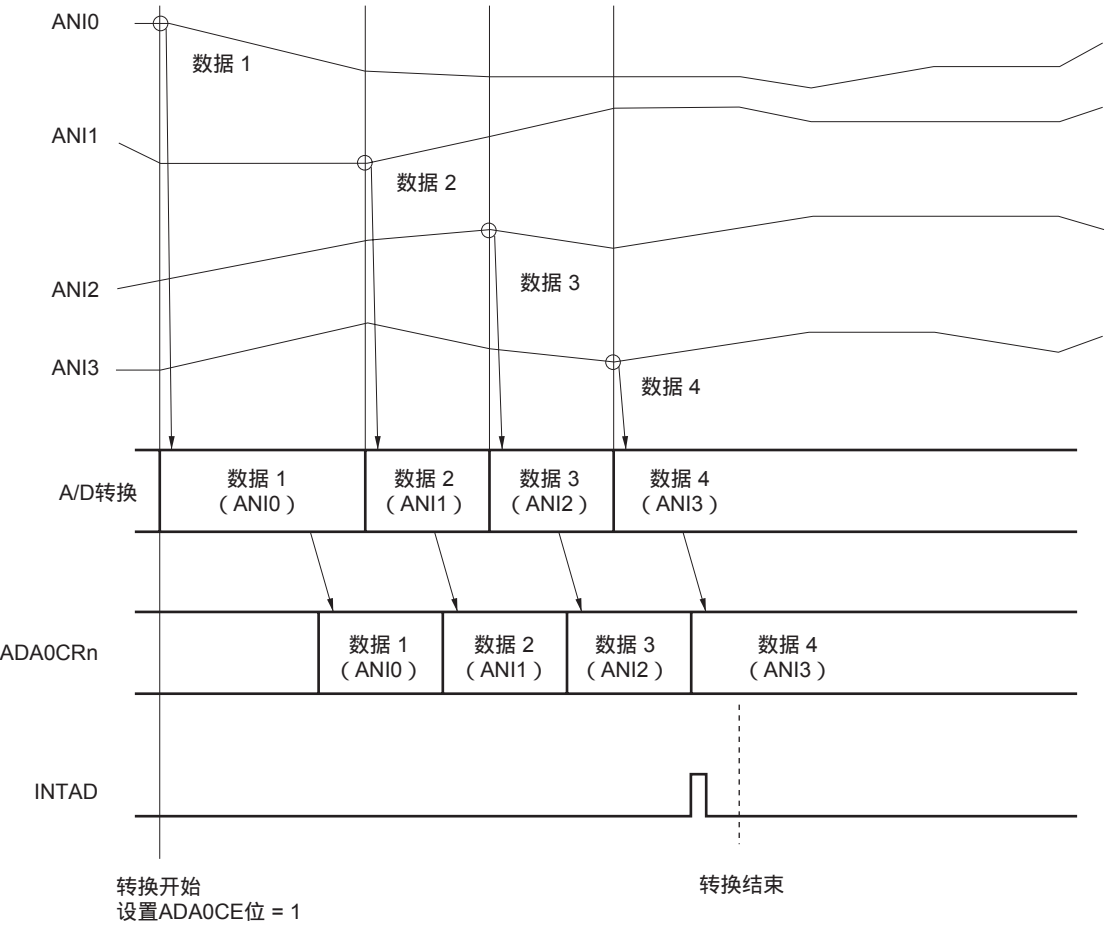


(3) 单脉冲扫描模式

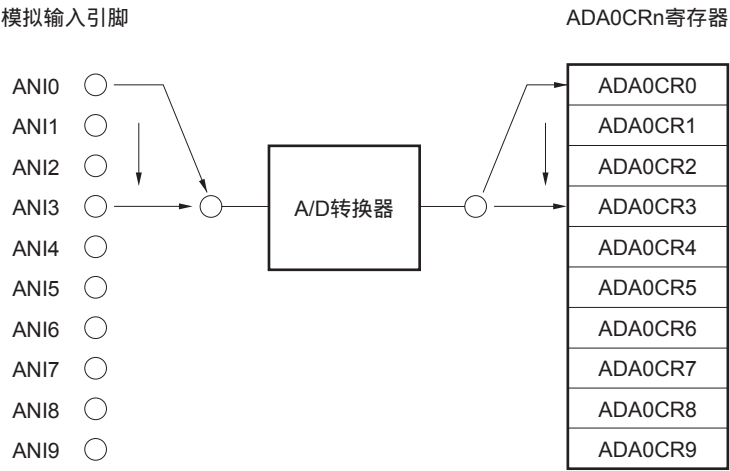
该模式下，模拟输入引脚被依次选择（从引脚 ANI0 到寄存器 ADA0S 所设置的指定引脚），并转换成数字量。每次转换的结果存储于与模拟输入引脚对应的寄存器 ADA0CRn 中。当由寄存器 ADA0S 设置的模拟输入引脚的转换完成时，产生 INTAD 信号。A/D 转换（n = 0 至 9）完成后，转换操作停止。

图 11-5. 单脉冲扫描模式的操作时序示例（寄存器 ADA0S = 03H）

(a) 时序示例



(b) 框图



11.5.4 掉电比较模式

寄存器 ADA0PFM 和 ADA0PFT 用于控制 A/D 转换结束中断请求 (INTAD)。

- 当 ADA0PFM.ADA0PFE = 0 时，每当转换完成时都会产生信号 INTAD（正常使用 A/D 转换器）。
- 当 ADA0PFE = 1 且 ADA0PFM.ADA0PFC = 0 时，当转换完成后，寄存器 ADA0CRnH 的值将与 ADA0PFT 的值进行比较，且只有当 $\text{ADA0CRnH} \geq \text{ADA0PFT}$ 时产生信号 INTAD。
- 当 ADA0PFE = 1 且 ADA0PFC = 1 时，当转换完成后，寄存器 ADA0CRnH 的值将与 ADA0PFT 的值进行比较，且只有当 $\text{ADA0CRnH} < \text{ADA0PFT}$ 时产生信号 INTAD。

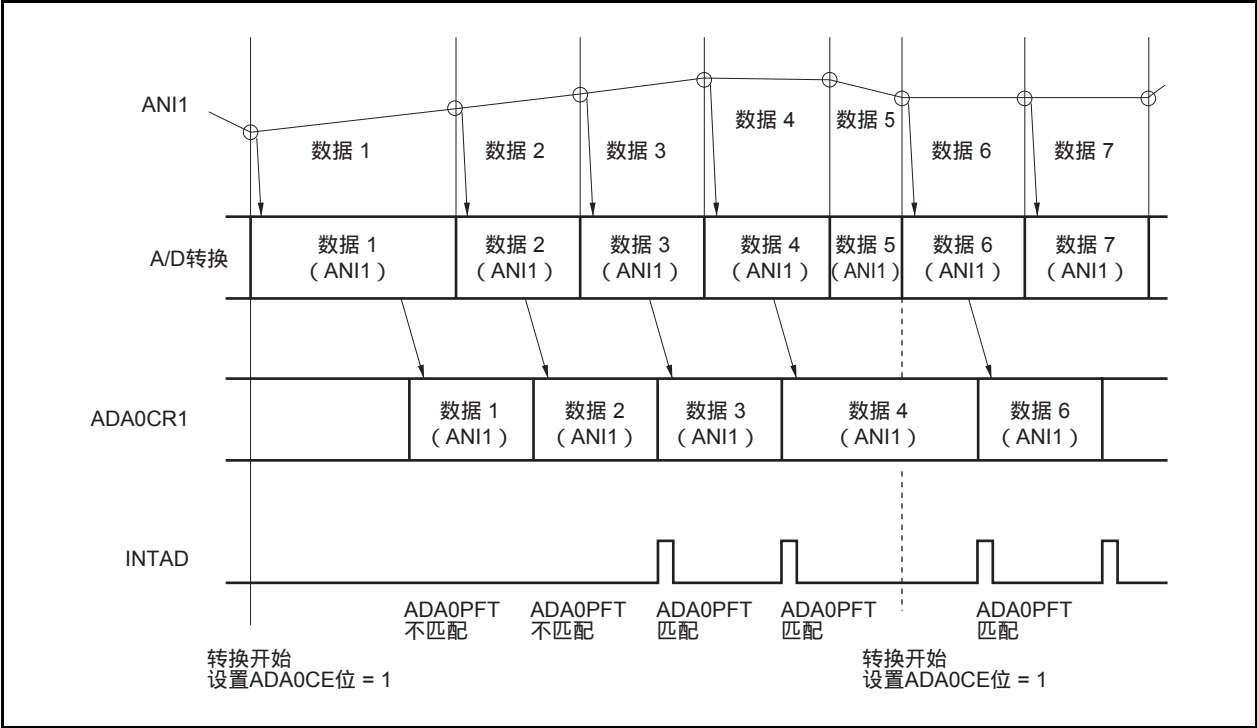
备注 n = 0 至 9

在掉电比较模式下，有三种模式可用来设置 ANI0 到 ANI9 引脚：连续选择模式，连续扫描模式以及单脉冲扫描模式。

(1) 连续选择模式

该模式下，由寄存器 ADA0S 设置的模拟输入引脚的转换电压结果与寄存器 ADA0PFT 的设置值比较。如果掉电比较的结果与 ADA0PFC 设置的条件匹配，转换结果存储于寄存器 ADA0CRn 中，产生 INTAD 信号。如果不匹配，转换结果存储于寄存器 ADA0CRn 中，不产生 INTAD 信号。第一次转换完成后，除非将 ADA0M0.ADA0CE 位清零（n = 0 到 9），否则下一次转换开始。

图 11-6. 连续选择模式的操作时序示例
(掉电比较模式下：寄存器 ADA0S = 01H)

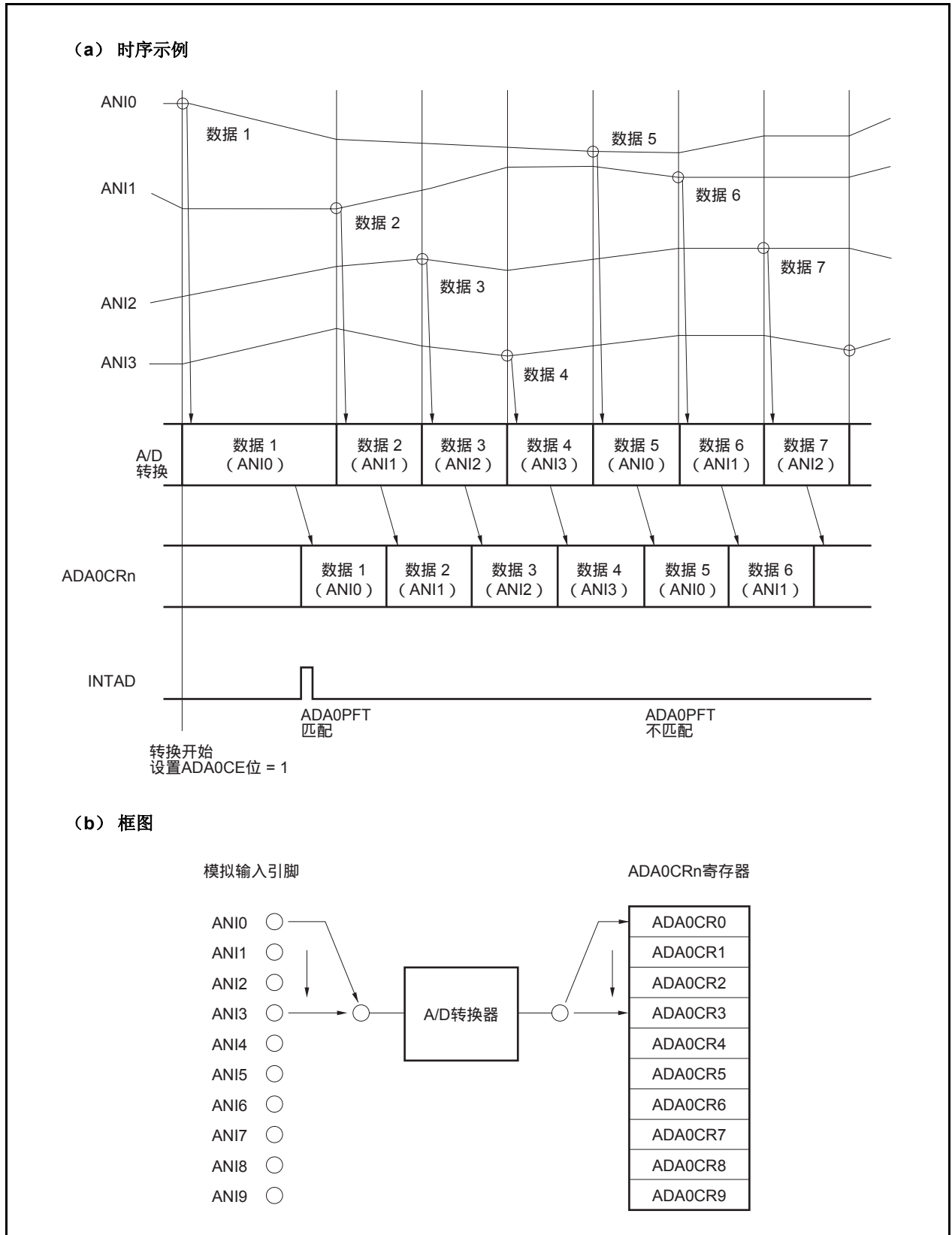


(2) 连续扫描模式

该模式下，模拟输入引脚被依次选择（从引脚 ANI0 到寄存器 ADA0S 所设置的指定引脚），模拟输入引脚的电压转换结果被存储，寄存器 ADA0CR0H 通道 0 的设定值与寄存器 ADA0PFT 的值比较。如果掉电比较的结果与 ADA0PFC 设置的条件匹配，转换结果存储于寄存器 ADA0CRn 中，产生 INTAD 信号。如果不匹配，转换结果存储于寄存器 ADA0CRn 中，不产生 INTAD 信号。

第一次转换的结果存储于寄存器 ADA0CR0 后，模拟输入引脚到寄存器 ADA0S 指定引脚的电压连续转换结果连续被存储。转换完成后，除非 ADA0M0.ADA0CE 位被清零，否则下一次转换开始。

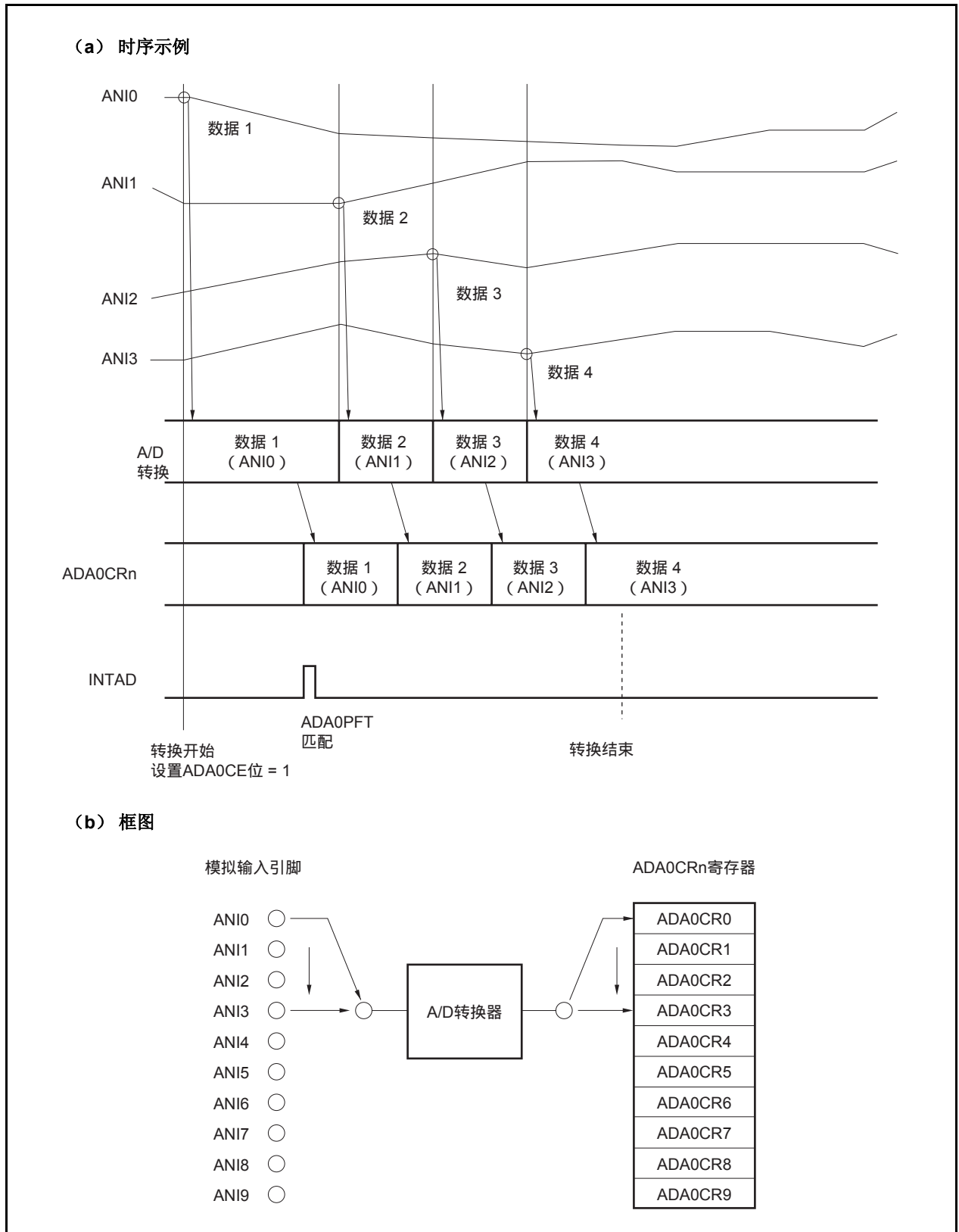
图 11-7. 连续扫描模式的操作时序示例
(掉电比较模式下: 寄存器 ADA0S = 03H)



(3) 单脉冲扫描模式

该模式下，模拟输入引脚被依次选择（从引脚 ANI0 到寄存器 ADA0S 所设置的指定引脚），模拟输入引脚的电压转换结果被存储，寄存器 ADA0CR0H 通道 0 的设定值与寄存器 ADA0PFT 的值比较。如果掉电比较的结果与 ADA0PFC 设置的条件匹配，转换结果存储于寄存器 ADA0CRn 中，产生 INTAD 信号。如果不匹配，转换结果存储于寄存器 ADA0CRn 中，且不产生 INTAD 信号。第一次转换的结果存储于寄存器 ADA0CR0 后，模拟输入引脚到寄存器 ADA0S 指定引脚的电压连续转换结果连续被存储。A/D 转换完成后，转换操作停止。

图 11-8. 单脉冲扫描模式的操作时序示例
(掉电比较模式下: 寄存器 ADA0S = 03H)



11.6 注意事项

(1) 不使用 A/D 转换器时

当不使用 A/D 转换器时，通过将 ADA0M0.ADA0CE 位清零可以使功耗减少。

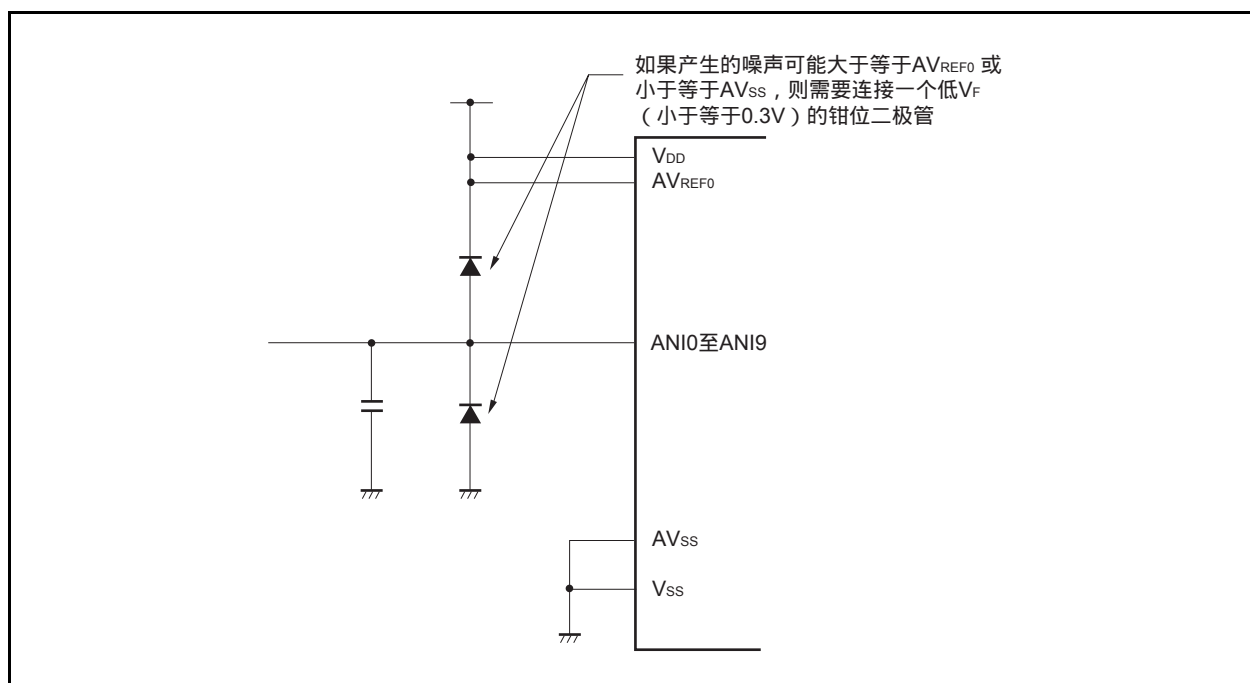
(2) ANI0 至 ANI9 引脚的输入范围

输入指定范围的电压值到引脚 ANI0 至 ANI9。如果大于等于 AV_{REF0} 或小于等于 AV_{SS} （即使在最大绝对值范围内）的电压被输入到这些引脚，则那个通道的转换值不确定，其它通道的转换值也会受到影响。

(3) 抑制噪声的方法

为了确保 10 位分辨率，引脚 ANI0 至 ANI9 必须有效抑制噪声。噪声影响随着模拟输入源的输出阻抗的增加而提高。为了降低噪声，推荐连接一个外部电容，如图 11-9 所示。

图 11-9. 模拟输入引脚的处理



(4) I/O 复用

模拟输入引脚（ANI0 至 ANI9）可功能复用为普通端口引脚。当从引脚 ANI0 至 ANI9 中选择一个执行 A/D 转换时，不要在转换期间执行读/写端口指令，否则转换分辨率将下降。

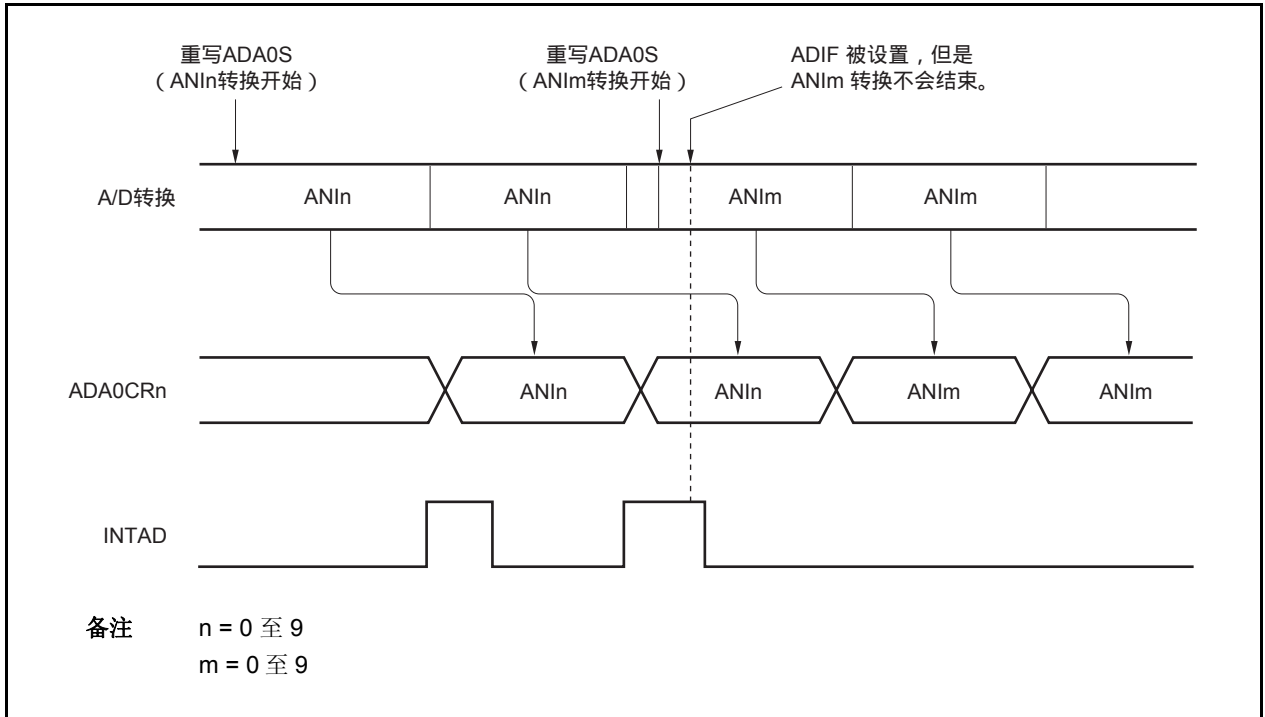
在转换期间设置引脚为输出端口同样会使转换分辨率将下降。其原因是连接端口引脚的外部电路导致输出电流波动。

如果某引脚正在进行 A/D 转换，其临近引脚输入一个数字脉冲，则由于耦合噪声的影响，A/D 转换值可能不准确。因此，在 A/D 转换过程中，确保不使用临近引脚传输脉冲信号。

(5) 中断请求标志 (ADIF)

即使寄存器 ADA0S 的内容改变，中断请求标志 (ADIF) 也不清零。因此，如果在 A/D 转换过程中模拟输入引脚改变，上次所选择的模拟输入信号的转换结果可能被存储，转换结束中断请求标志可能在寄存器 ADA0S 复写前立刻置 1。如果寄存器 ADA0S 复写后立即读取标志 ADIF，标志 ADIF 可能置 1，即使新选择的模拟输入信号的转换还未完成。当 A/D 转换停止，在再次转换前使标志 ADIF 清零。

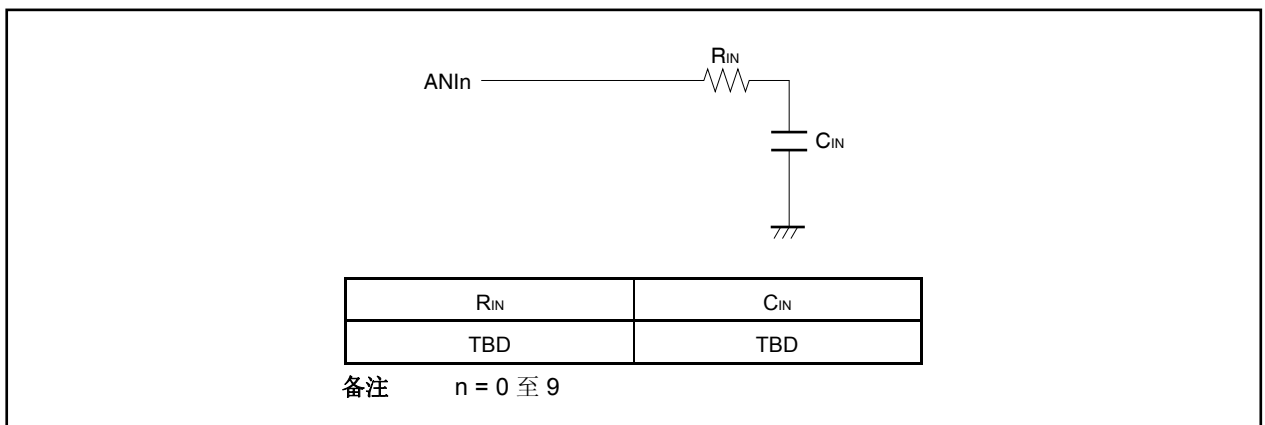
图 11-10. A/D 转换结束中断请求的产生时序



(6) 内部等效电路

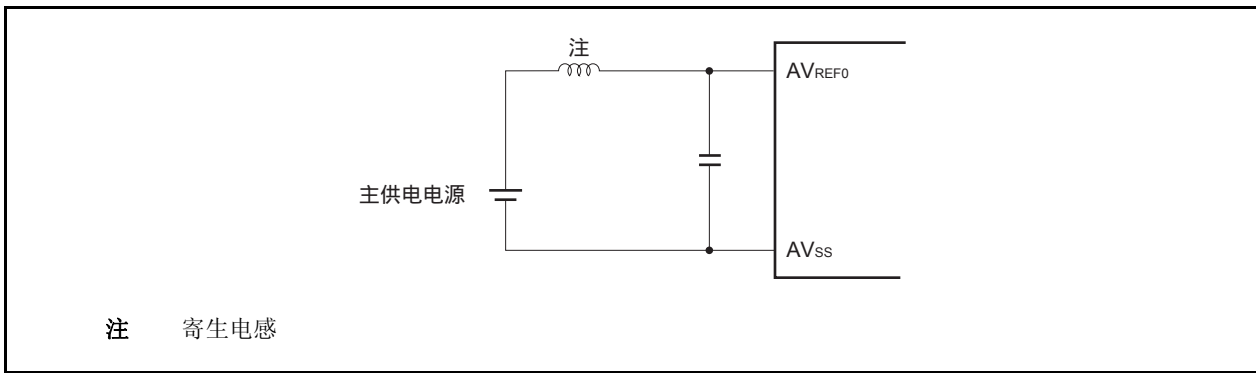
以下显示了模拟输入模块的等效电路。

图 11-11. 引脚 ANIn 的内部等效电路



(7) AVREF0 引脚

- (a) 引脚 AVREF0 被用作 A/D 转换器的供电电压引脚，同时也为复用功能端口供电。当使用备用供电电源时，确保向引脚 AVREF0 提供与 VDD 相同的电压，如图 11-12 所示。
- (b) 引脚 AVREF0 也被用作 A/D 转换器的参考电压引脚。如果提供给引脚 AVREF0 的源供电电源具有高阻抗，或者如果供电电源低电流负载能力低，参考电压可能被转换中的电流影响（特别是，转换操作使能位 ADA0CE 刚刚置 1 之后）。结果，转换精度会下降。如图 11-12 所示，为了避免上述情况，推荐连接一个电容在引脚 AVREF0 和 AVSS 之间以抑制参考电压的波动。
- (c) 如果提供给引脚 AVREF0 的源供电电源具有高直流阻抗（例如，由于插入一个二极管），当转换使能时的电压可能比转换停止时的电压低，因为 A/D 转换电流引起电压失真。

图 11-12. 引脚 AVREF0 处理示例**<R> (8) 读取寄存器 ADA0CRn**

当执行寄存器 ADA0M0 至 ADA0M2, ADA0S, ADA0PFM 或 ADA0PFT 的写指令时，寄存器的 ADA0CRn 内容可能不确定。转换完成后，在写入寄存器 ADA0M0 至 ADA0M2, ADA0S, ADA0PFM 和 ADA0PFT 之前，读取转换结果。另外，当接收到一个外部/定时器触发信号时，ADA0CRn 寄存器的内容可能是不确定的。应在转换完成之后及在下一个外部/定时器触发信号收到之前读取转换结果。在与上述时序不同时，正确转换结果可能无法读取。

(9) A/D 转换结果

如果在模拟输入引脚及参考电压输入引脚上有噪声，那么该噪声可能会产生一个非法转换结果。因此，需要通过软件处理来避免该非法转换结果对于系统的负效应。软件处理的示例如下所示。

- 取若干 A/D 转换的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。
- 如果获得一个可以判断为已经生成了一个系统故障的 A/D 转换结果，那么在执行故障处理前请务必复查系统故障。

<R>

(10) 等待模式

因为 A/D 转换器在 STOP 模式下停止工作，转换结果无效，所以功耗可以降低。释放 STOP 模式后，恢复工作，但 STOP 模式释放后的 A/D 转换结果仍然无效。当 STOP 模式释放后使用 A/D 转换器时，在设置 STOP 模式或释放 STOP 模式之前，使 ADA0M0.ADA0CE 位清 0，然后在释放 STOP 模式后将 ADA0CE 位置 1。

在 IDLE1，IDLE2 或副时钟操作模式下，继续运行。因此，为了减小功耗，应使 ADA0M0.ADA0CE = 0。在 IDLE1，IDLE2 模式下，由于模拟输入电压值不能保留，IDLE1，IDLE2 模式释放后的 A/D 转换结果无效。进入 IDLE1，IDLE2 模式前的转换结果有效。

<R>

(11) 稳定时间内改写寄存器及触发输入

禁止在稳定时间内改写 ADA0M0，ADA0M2，ADA0S，ADA0PFM 和 ADA0PFT 寄存器及触发输入。

(12) A/D 转换结果的误差

A/D 转化的结果会根据供电电压的波动或可能受到干扰而变化。要减小误差，须采取软件抗干扰措施，例如 A/D 转换结果平滑处理。

(13) A/D 转换结果滞后特性

逐次逼近 A/D 转换器维持内部采样和保持电容的模拟输入电压，同时进行 A/D 转换。A/D 转换结束后模拟输入电压保留在内部采样和保持电容中。结果，产生下述现象。

- 当同一个通道用于 A/D 转换，如果电压高于或低于先前 A/D 转换，则出现滞后特征，转换结果受到上次转换值影响。因此，即使同一个模拟输入电压也会产生不同结果。
- 当开关模拟输入通道时，可能出现滞后特征，转换结果受到上次转换通道影响。这是因为只有一个 A/D 转换器用于 A/D 转换。因此，即使同一个模拟输入电压也会产生不同结果。

<R>

因此，要获取更多准确的转换结果，应对同一通道连续执行两次 A/D 转换，并丢弃第一个转换结果。

11.7 A/D 转换器特征表的阅读方法

以下介绍 A/D 转换器中的专用术语。

(1) 分辨率

这是可识别的最小的模拟输入电压，即每位数字输出的模拟输入电压的百分比，称为 1 LSB（最低有效位）。对于满度的 1 LSB 的百分比用 %FSR（满度范围）表示。%FSR 是以百分比形式表示的一定范围的可变模拟输入电压的比率。与分辨率无关，可如下表示。

$$\begin{aligned} 1\%FSR &= (\text{可变模拟输入电压的最大值} - \text{可变模拟输入电压的最小值}) / 100 \\ &= (AV_{REF0} - 0) / 100 \\ &= AV_{REF0} / 100 \end{aligned}$$

当分辨率为 10 位时 1LSB 表示如下：

$$\begin{aligned} 1 \text{ LSB} &= 1/2^{10} = 1/1,024 \\ &= 0.098\%FSR \end{aligned}$$

精度与分辨率无关，而由总误差决定。

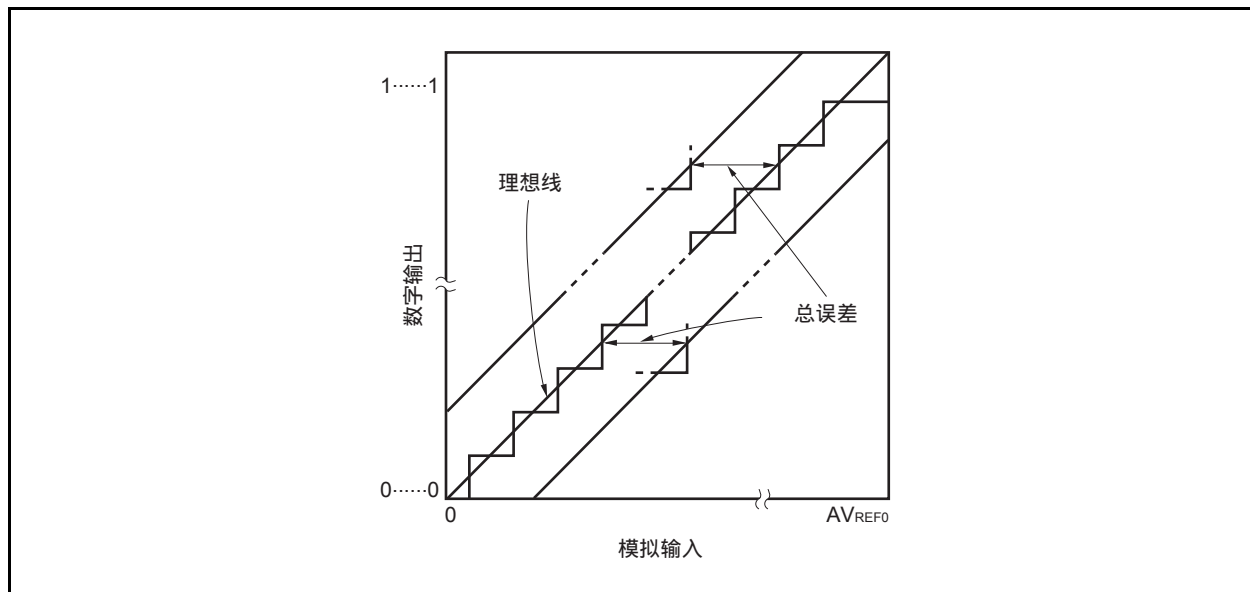
(2) 总误差

总误差是指实际测量值与理论值之间的最大误差。

零度误差，满度误差，积分线性误差和积分线性误差等组合起来表示总误差。

量化误差不属于特征表中总误差的范围。

图 11-13. 总误差

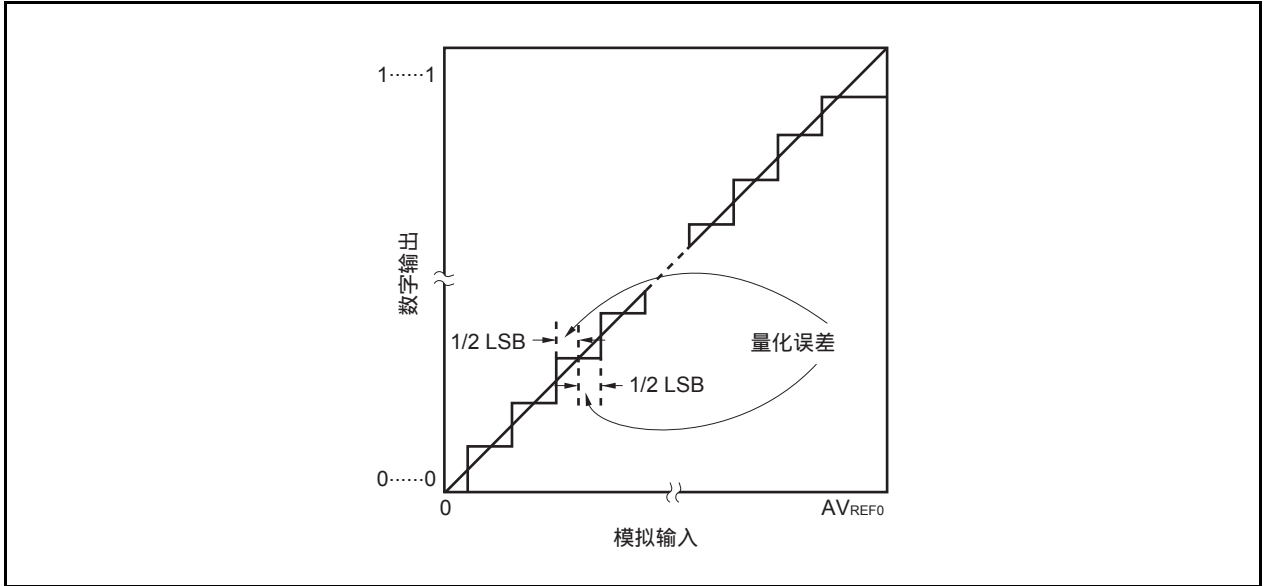


(3) 量化误差

当模拟值转换成数字值时，通常会产生 $\pm 1/2\text{LSB}$ 的误差。在一个 A/D 转换器中，相差 $\pm 1/2\text{LSB}$ 的模拟输入电压被转换成相同的数字代码，因此量化误差不可避免。

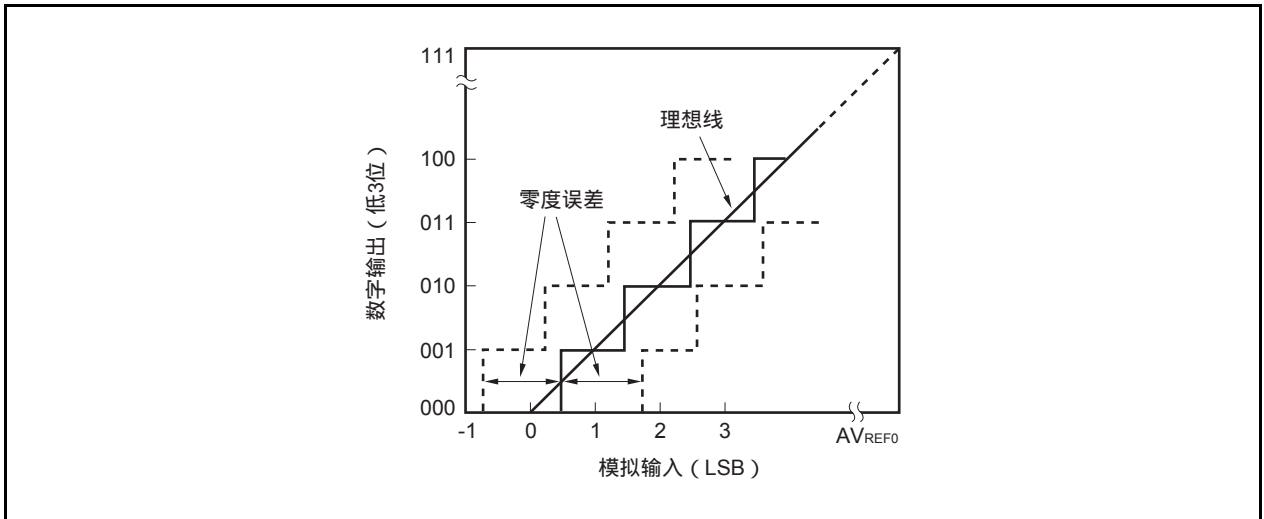
量化误差不属于特征表中总误差、零度误差、满度误差、积分线性误差和微分线性误差的范围。

图 11-14. 量化误差

**(4) 零度误差**

零度误差表示当数字输出范围在 0.....000 至 0.....001 之间时模拟输入电压的实际测量值与理论值 ($1/2\text{LSB}$) 之间的误差。

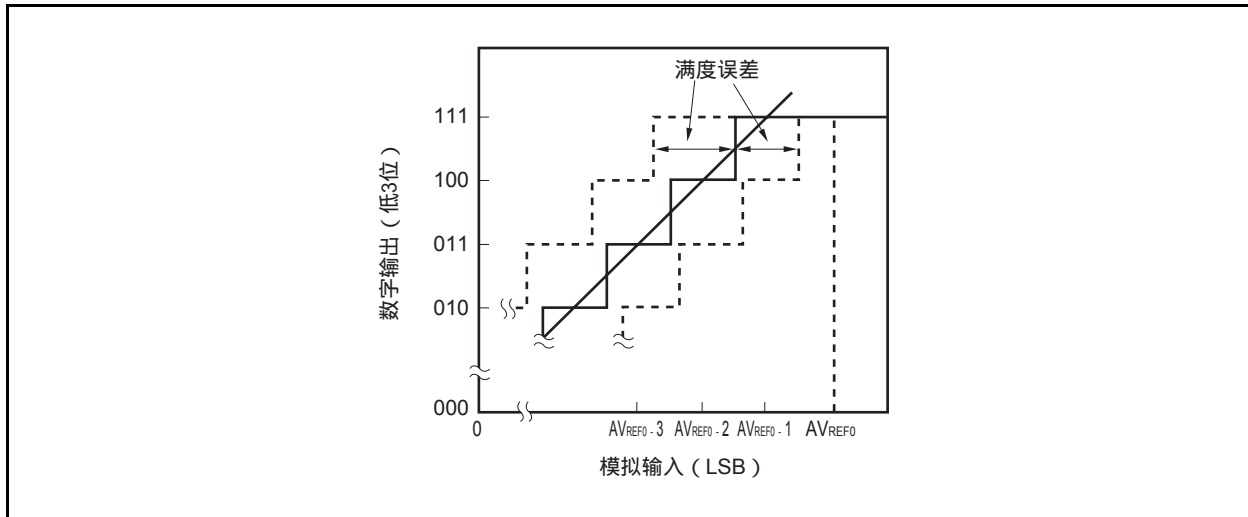
图 11-15. 零度误差



(5) 满度误差

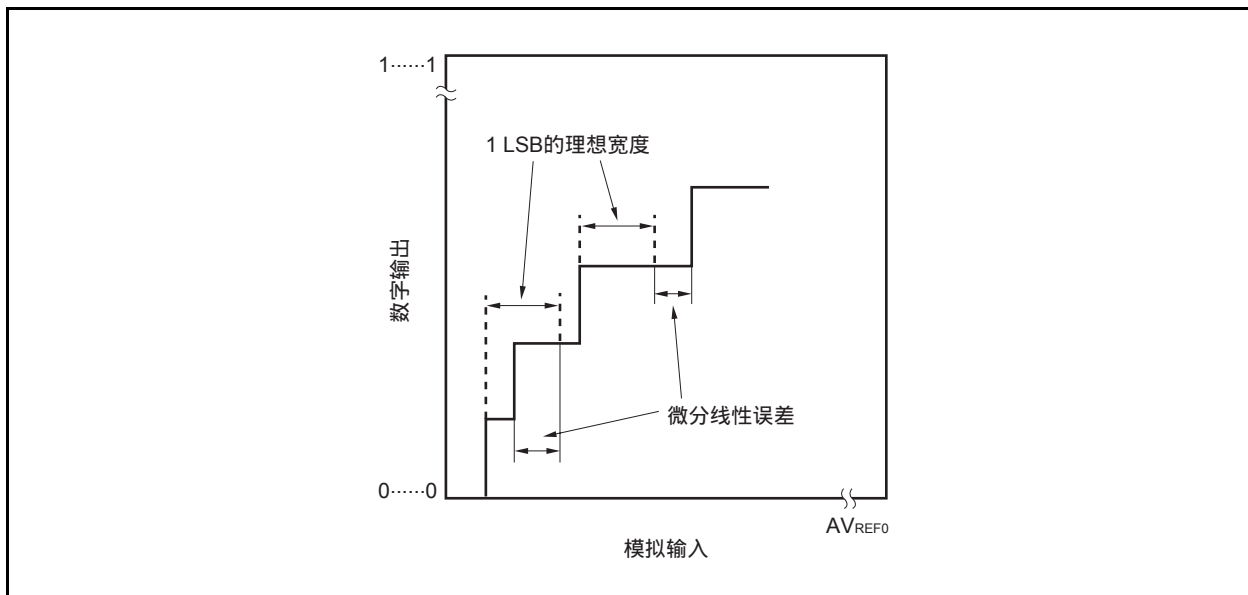
满度误差表示当数字输出范围在 1.....110 至 1.....111 之间时模拟输入电压的实际测量值与理论值（满度 - 3/2LSB）之间的误差。

图 11-16. 满度误差

**(6) 微分线性误差**

理论上，代码输出的宽度为 1LSB。当输出一个特定码，微分线性误差表示实际测量值与理想值之间的差距。当同一个通道的模拟输入引脚的电压一字节的从 AVSS 到 AVREF0 持续增加，这显示了 A/D 转换的基本特征。当输入电压增加或减小，或两个（包括两个以上）被使用，参见 11.7（2）总误差。

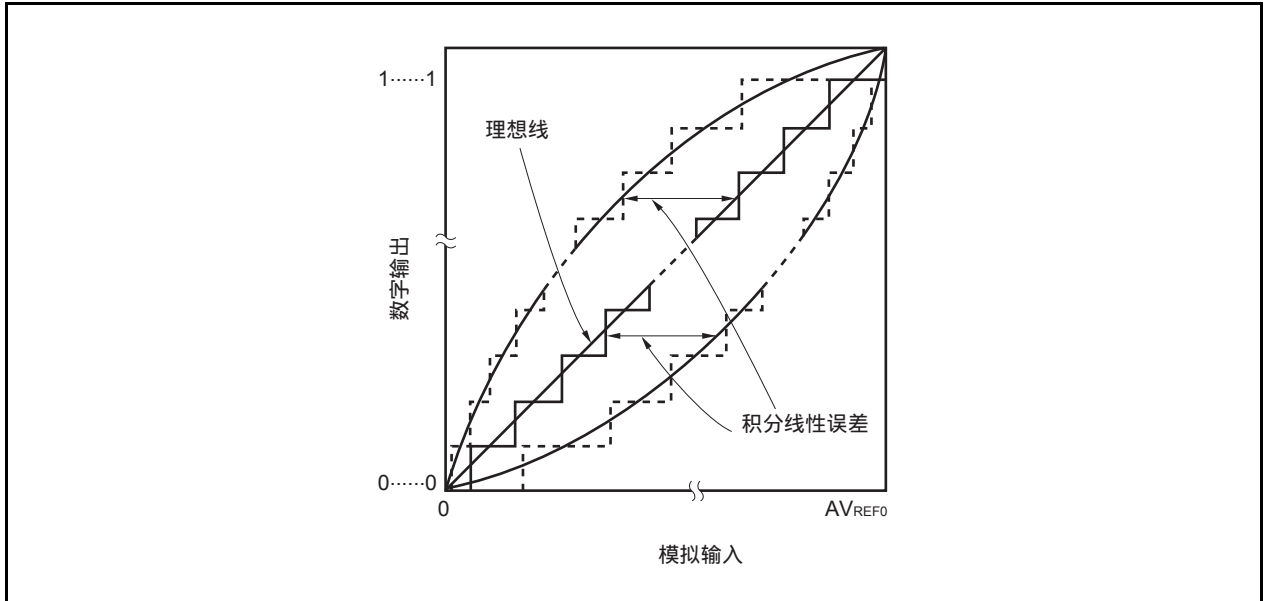
图 11-17. 微分线性误差



(7) 积分线性误差

积分线性误差说明了转换特征偏离理想线性关系的程度。它表示当零度误差和满度误差均为 0 时实际测量值与理想直线之间误差的最大值。

图 11-18. 积分线性误差

**(8) 转换时间**

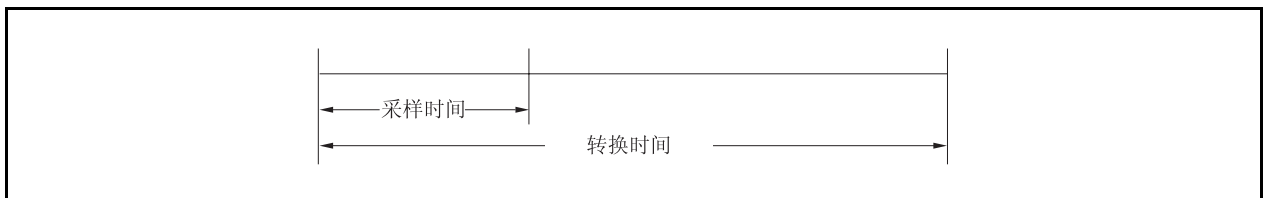
转换时间表示从产生触发到获取数字输出所经历的时间。

采样时间包含在特征表中的转换时间中。

(9) 采样时间

采样时间表示模拟电压的模拟开关被打开到模拟电压被采样 & 保持电路采样所需的时间。

图 11-19. 采样时间



第 12 章 异步串行接口 A (UARTA)

V850ES/HE2 包含异步串行接口 A (UARTA) 的两处通道。

12.1 特性

- 传输速率: 300 bps 至 312.5kbps (使用 20 MHz 内部系统时钟和专用波特率发生器)
- 全双工通信:
 - 内部 UARTAn 数据接收寄存器 (UAnRX)
 - 内部 UARTAn 数据发送寄存器 (UAnTX)
- 2 引脚设置:
 - TXDAn: 数据发送输出引脚
 - RXDAn: 数据接收输入引脚
- 接收错误输出功能
 - 校验错误
 - 帧错误
 - 溢出错误
- 中断源: 2
 - 接收完成中断 (INTUAnR):

通过对接收错误的三种类型进行逻辑或操作在接收使能状态中生成一个中断。串行传输完成后, 当接收数据从接收位移寄存器转移到接收数据寄存器时也会生成一个中断。
 - 发送使能中断 (INTUAnT):

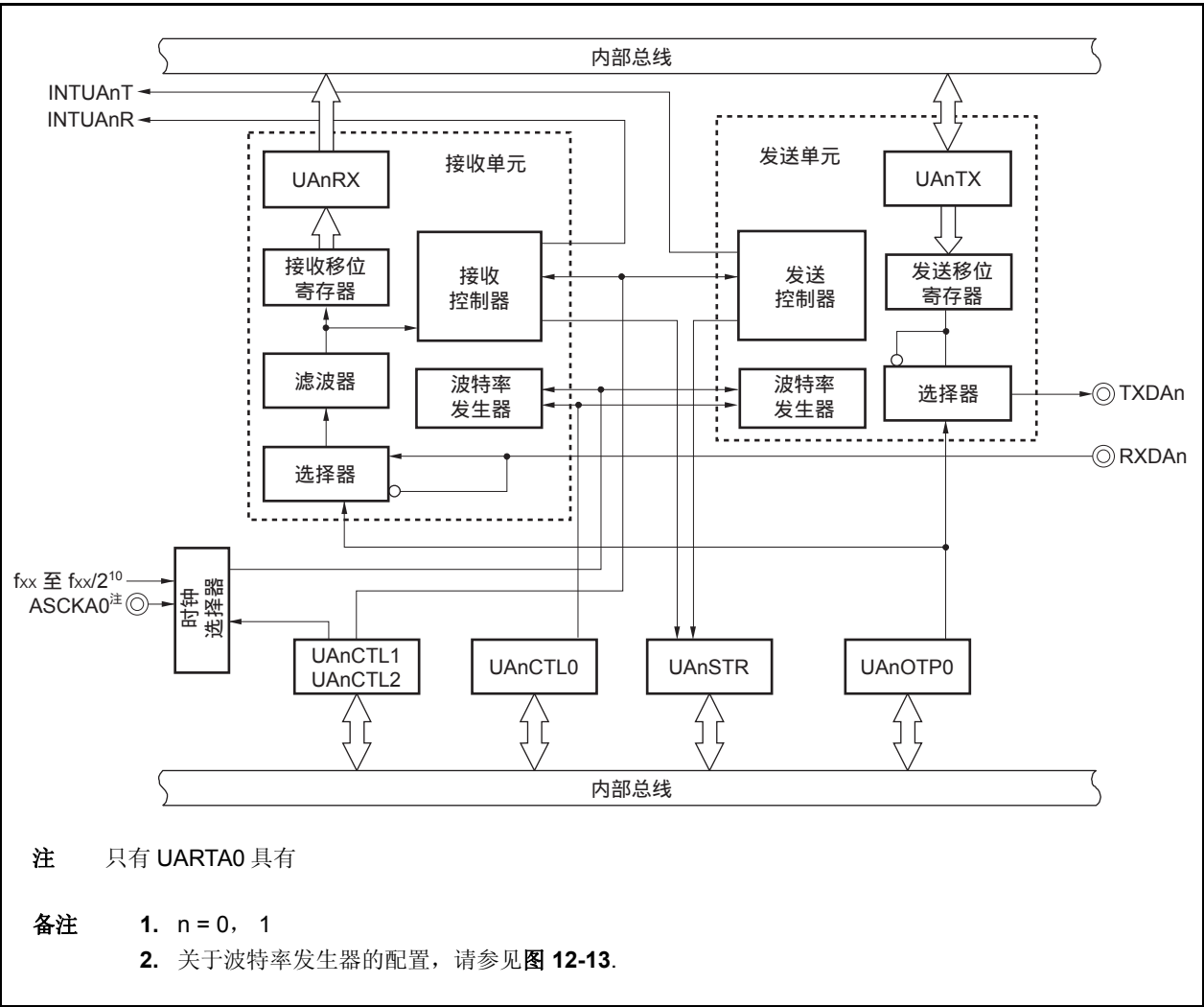
该中断产生于发送使能状态下, 发送数据从数据发送寄存器到发送移位寄存器的转换过程中。
- 字符长度: 7, 8 位
- 校验功能: 奇校验, 偶校验, 零校验, 不输出校验位
- 传输停止位: 1, 2 位
- 片上专用波特率发生器
- 可选 MSB-/LSB-first 通信
- 发送/接收数据反向输入/输出操作
- LIN (局部互联网) 通信模式下 SBF (同步中断区域) 的发送/接收
 - 同步中断区域发送长度为 13 至 20
 - 识别 11 位或更多位的 SBF 接收
 - 提供 SBF 接收标志

备注 n = 0, 1

12.2 配置

以下显示了 UARTAn 的框图。

图 12-1. 异步串行接口 An 的框图



UARTAn 包含以下硬件单元。

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 数据接收寄存器 (UAnRX) UARTAn 发送移位寄存器 UARTAn 数据发送寄存器 (UAnTX)

(1) UARTAn 控制寄存器 0 (UAnCTL0)

该寄存器是 8 位寄存器，用于设定 UARTAn 的操作。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

该寄存器是 8 位寄存器，用于选择 UARTAn 的输入时钟。

(3) UARTAn 控制寄存器 2 (UAnCTL2)

该寄存器是 8 位寄存器，用于控制 UARTAn 的波特率。

(4) UARTAn 选项控制寄存器 0 (UAnOPT0)

该寄存器是 8 位寄存器，用于控制 UARTAn 的串行传输。

(5) UARTAn 状态寄存器 (UAnSTR)

该寄存器包含发生接收错误时指示错误内容的标志。当发生接收错误时，相应的接收错误标志置 1。

(6) UARTAn 接收移位寄存器

该移位寄存器用于将输入到引脚 RXDAn 的串行数据转换为并行数据。当接收到 1 字节的数据并检测到停止位，接收数据被传送到寄存器 UAnRX。

该寄存器不能直接操作。

(7) UARTAn 数据接收寄存器 (UAnRX)

该寄存器是 8 位寄存器，用于保持接收数据。当接收到 7 个位，0 被存储于最高位（当数据以 LSB-first 模式传输）。

在接收使能状态，接收数据以同步方式在每帧移位过程中，从 UARTAn 的接收移位寄存器传送到寄存器 UAnRX。

传输到寄存器 UAnRX 同时引起产生接收完成中断请求信号 (INTUAnR)。

(8) UARTAn 发送移位寄存器

该寄存器是移位寄存器，用于将寄存器 UAnTX 传出的并行数据转换成串行数据。

当 1 字节数据从寄存器 UAnTX 传送，移位寄存器数据从引脚 TXDAn 输出。

该寄存器不能直接操作。

(9) UARTAn 数据发送寄存器 (UAnTX)

该寄存器是 8 位数据传送缓冲。当要传送的数据写入寄存器 UAnTX，传送开始。当数据可以被写入寄存器 UAnTX（当 1 帧的数据从寄存器 UAnTX 传送到发送移位寄存器 UARTAn），产生发送使能中断请求信号 (INTUAnT)。

12.3 寄存器

(1) UARTAn 控制寄存器 0 (UAnCTL0)

该寄存器是 8 位寄存器，用于控制 UARTAn 的串行发送操作。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位将该寄存器设置为 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 复位不同步)
1	允许 UARTAn 操作
UARTAn操作通过UAnPWR位来控制。 通过将UAnPWR位清零 (如果 UAnOPT0.UAnTDL位= 1则固定为低电平)，TXDAn引脚输出被固定为高电平。	

UAnTXE	允许发送操作
0	禁止发送操作
1	允许发送操作
<ul style="list-style-type: none"> · 要开始发送，设置UAnPWR位为1，然后设置 UAnTXE位为1。 · 要停止发送，先将UAnTXE 位清零，然后 将UAnPWR位清零。 · 要初始化发送单元，先将UAnTXE位清零，等待2个基本时钟周期，然后设置 UAnTXE位再次为1。否则，初始化可能不被执行 (关于基本时钟的详细信息，参见12.6 (1) (a) 基本时钟)。 	

UAnRXE	允许接收操作
0	禁止接收操作
1	允许接收操作
<ul style="list-style-type: none"> · 要开始接收，设置UAnPWR位为1，然后设置 UAnRXE位为1。 · 要停止接收，先将UAnRXE 位清零，然后 将UAnPWR位清零。 · 要初始化接收单元，先将UAnRXE位清零，等待2个基本时钟周期，然后设置 UAnRXE位再次为1。否则，初始化可能不被执行 (关于基本时钟的详细信息，参见12.6 (1) (a) 基本时钟)。 	

<R>

UAnDIR	发送方向选择
0	MSB优先发送
1	LSB优先发送
<ul style="list-style-type: none">只有当UAnPWR 位 = 0或UAnTXE 位=UAnRXE位 = 0时，该寄存器可被改写。如果发送和接收都在LIN格式下进行，则将UAnDIR位设为1。	

UAnPS1	UAnPS0	发送期间校验选择	接收期间校验选择
0	0	不输出校验位	无校验接收
0	1	输出零校验	按零校验接收
1	0	输出奇校验	按奇校验判断
1	1	输出偶校验	按偶校验判断
<ul style="list-style-type: none">只有当UAnPWR 位 = 0或UAnTXE 位=UAnRXE位 = 0时，该寄存器可被改写。如果接收时被设置为“按零校验接收”，将不会执行校验判断。因此，UAnSTR.UAnPE位不被设置。如果发送和接收都在LIN格式下进行，则将UAnPS1及UAnPS0位清除为00。			

<R>

UAnCL	指定每帧发送/接收数据的字符长度
0	7位
1	8位
<ul style="list-style-type: none">只有当UAnPWR 位 = 0或UAnTXE 位=UAnRXE位 = 0时，该寄存器可被改写。当发送和接收都在LIN格式下进行，则将UAnCL位设为1。	

UAnSL	指定发送数据停止位的长度
0	1位
1	2位
只有当UAnPWR 位 = 0或UAnTXE 位=UAnRXE位 = 0时，该寄存器可被改写。	

备注 有关校验的详细内容，参见 **12.5.9 校验形式和操作**。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

详细信息请参见 12.6 (2) UARTAn 控制寄存器 1 (UAnCTL1)。

(3) UARTAn 控制寄存器 2 (UAnCTL2)

详细信息请参见 12.6 (3) UARTAn 控制寄存器 2 (UAnCTL2)。

(4) UARTAn 选项控制寄存器 0 (UAnOPT0)

该寄存器是 8 位寄存器，用于控制 UARTAn 的串行传输。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号产生将该寄存器设置为 14H。

(1/2)

复位后：14H R/W 地址： UA0OPT0 FFFFA03H, UA1OPT0 FFFFA13H

	7	6	5	4	3	2	1	0
UAnOPT0	UAnSRF	UAnSRT	UAnSTT	UAnSLS2	UAnSLS1	UAnSLS0	UAnTDL	UAnRDL
(n = 0, 1)								

UAnSRF	SBF接收标志
0	当UAnCTL0.UAnPWR = UAnCTL0.UAnRXE = 0。 或者SBF接收正常结束。
1	SBF接收期间。
<ul style="list-style-type: none"> • SBF（同步中断区）接收只在LIN通信时被判断。 • 当出现SBF接收错误时UAnSRF位保持1，然后SBF接收重新开始。 • UAnSRF位为只读位。 	

UAnSRT	SBF接收触发
0	-
1	SBF接收触发
<ul style="list-style-type: none"> • 该位是LIN通信期间的SBF接收触发位，读取值恒为0。 • 要接收SBF，设置UAnSRT（为1）以允许SBF接收。 • 在设置UAnPWR 位 = UAnRXE 位 = 1之后，设置UAnSRT位。 	

UAnSTT	SBF发送触发
0	-
1	SBF发送触发
<ul style="list-style-type: none"> • 该位是LIN通信期间的SBF接收触发位，读取值恒为0。 • 在设置UAnPWR 位 = UAnTXE 位 = 1之后，设置UAnSTT位。 	

注意事项 在 SBF 接收期间（UAnSRF 位=1），不要将 UAnSRT 和 UAnSTT 置 1。

<R>

UAnSLS2	UAnSLS1	UAnSLS0	SBF发送长度控制
1	0	1	13位输出 (复位值)
1	1	0	14位输出
1	1	1	15位输出
0	0	0	16位输出
0	0	1	17位输出
0	1	0	18位输出
0	1	1	19位输出
1	0	0	20位输出

该寄存器可在UAnPWR 位= 0 或在UAnTXE位 = 0进行设置。

UAnTDL	发送数据电平位
0	发送数据正常输出
1	发送数据反转输出

- 引脚TXDAn输出电平可通过设置UAnTDL位反转。
- 该寄存器可在UAnPWR 位= 0 或在UAnTXE位 = 0进行设置。

UAnRDL	接收数据电平位
0	接收数据正常输出
1	接收数据反转输出

- 引脚RXDAn的输入电平可通过设置UAnRDL位翻转。
- 该寄存器可在UAnPWR 位= 0 或在UAnRXE位 = 0进行设置。

(5) UARTAn 状态寄存器 (UAnSTR)

寄存器是 8 位寄存器，显示了 UARTAn 的传送状态和接收错误原因。

可对该寄存器进行 8 位或 1 位的读写操作。UAnTSF 位是只读的。UAnPE, UAnFE 和 UAnOVE 位可以进行读/写操作。只有写入 0 时，这些位才清零；写入 1 时，这些位不能置 1（即使写入 1，它们也保持原值）。

初始条件如下所示。

寄存器/位	初始条件
UAnSTR 寄存器	<ul style="list-style-type: none"> • 复位 • UAnCTL0.UAnPWR = 0
UAnTSF 位	<ul style="list-style-type: none"> • UAnCTL0.UAnTXE = 0
UAnPE, UAnFE, UAnOVE 位	<ul style="list-style-type: none"> • 写入 0 • UAnCTL0.UAnRXE = 0

复位后：00H R/W 地址： UA0STR FFFFA04H, UA1STR FFFFA14H

	7	6	5	4	3	2	1	0
UAnSTR (n = 0, 1)	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

UAnTSF	发送状态标志
0	<ul style="list-style-type: none"> 当设置 UAnPWR位 = 0 或UAnTXE位 = 0时。 发送完成后，没有数据从寄存器UAnTX发送时。
1	当数据被写入寄存器 UAnTX。
连续发送模式下UAnTSF位恒为1。当初始化发送单元，在执行初始化前查看UAnTSF位= 0。当UAnTSF位= 1时执行初始化，不产生发送数据。	

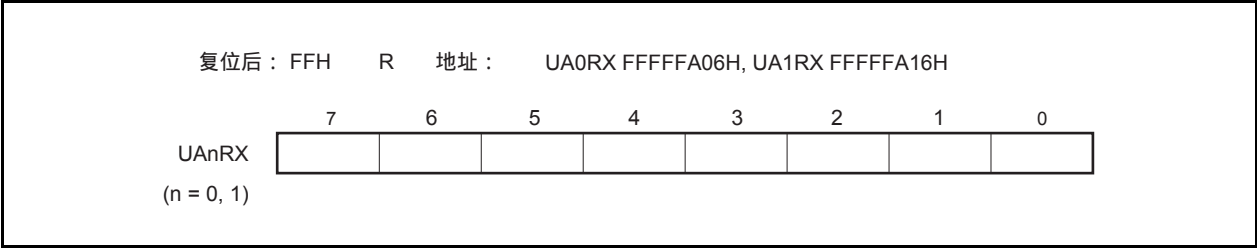
UAnPE	校验错误标志
0	<ul style="list-style-type: none"> 当设置 UAnPWR位 = 0 或UAnRXE位 = 0时。 当已经写入0时。
1	当校验数据与校验位在接收期间不匹配时。
<ul style="list-style-type: none"> UAnPE 位的操作由 UAnCTL0.UAnPS1和UAnCTL0.UAnPS0位的设置来控制。 UAnPE位可读写，但只能通过写0操作清零，不能通过写1操作进行设置。当向该位写1，数值不变。 	

UAnFE	帧错误标志
0	<ul style="list-style-type: none"> 当设置 UAnPWR位 = 0 或UAnRXE位 = 0时。 当已经写入0时。
1	当接收期间未检测到停止位时。
<ul style="list-style-type: none"> 无论UAnCTL0.UAnSL位如何设置，只检测接收数据的第一个停止位。 UAnFE位可读写，但只能通过写0操作清零，不能通过写1操作进行设置。当向该位写1，数值不变。 	

UAnOVE	溢出错误标志
0	<ul style="list-style-type: none"> 当设置 UAnPWR位 = 0 或UAnRXE位 = 0时。 当已经写入0时。
1	当接收到的数据被写入寄存器且在该数据被读取前下一个接收操作被完成。
<ul style="list-style-type: none"> 当产生溢出错误，数据被丢弃，下一个接收到的数据不被写入接收缓存。 UAnOVE位可读写，但只能通过写0操作清零，不能通过写1操作进行设置。当向该位写1，数值不变。 	

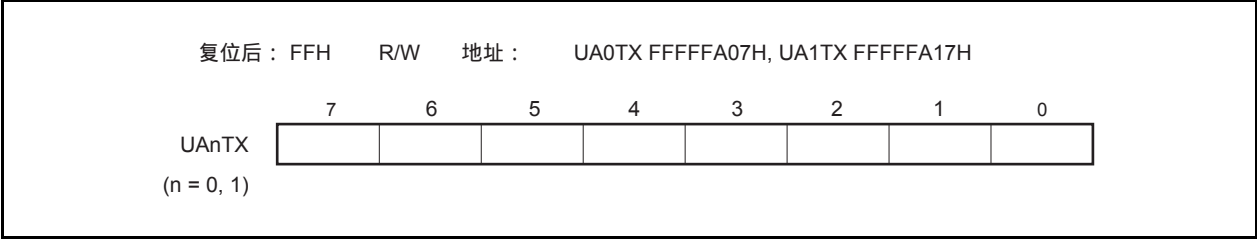
(6) UARTAn 数据接收寄存器 (UAnRX)

该寄存器是 8 位缓冲寄存器，用于存储由接收移位寄存器转换而来的并行数据。
完整接收 1 字节数据后，存储于接收移位寄存器的数据被传送到寄存器 UAnRX。
LSB-first 接收期间，当数据长度设置为 7 位，接收到的数据置于寄存器 UAnRX 的第 0 至 6 位，第 7 位恒为 0。
MSB-first 接收期间，接收到的数据置于寄存器 UAnRX 的第 1 至 7 位，第 0 位恒为 0。
当产生溢出错误 (UAnOVE)，此时接收到的数据不被传送到寄存器 UAnRX 并丢弃。
该寄存器是只读的，可由 8 位存储器操作指令设置该寄存器。
除了复位输入，通过 UAnCTL0.UAnPWR 置 0，寄存器 UAnRX 可置 FFH。



(7) UARTAn 数据发送寄存器 (UAnTX)

该寄存器是 8 位寄存器，用于设置发送数据。
可由 8 位存储器操作指令读/写该寄存器。
复位信号产生将该寄存器设置为 FFH。



12.4 中断请求信号

由 UARTAn 产生下面两个中断请求信号。

- 接收完成中断请求信号 (INTUAnR)
- 发送允许中断请求信号 (INTUAnT)

这两个中断请求信号的默认优先级为：接收完成中断请求信号优先级高，发送允许中断请求信号优先级低。

表 12-2. 中断及其默认优先级

中断	优先级
接收完成	高
允许发送	低

(1) 接收完成中断请求信号 (INTUAnR)

在允许接收状态中，当数据转移到接收移位寄存器并发送到寄存器 UAnRX 时，会输出一个接收完成中断请求信号。

当发生接收错误时也会输出一个接收完成中断请求信号。因此，当接收了一个接收完成中断请求信号并且数据被读取时，将读取寄存器 UAnSTR 并核实接收结果不是错误。

在禁止接收状态中，不产生接收完成中断请求信号。

(2) 发送允许中断请求信号 (INTUAnT)

允许发送时，如果发送的数据从寄存器 UAnTX 发送到发送移位寄存器 UARTAn，则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的接收和发送。

如图 12-2 所示，发送/接收数据的数据帧由起始位，字符位，校验位和停止位组成。

使用寄存器 UAnCTL0 来执行每个数据帧的字符位的长度，校验位选择，停止位长度和 MSB/LSB-first 传送。

此外，对于 TXDAn 位的 UART 输出/反向输出的控制可以通过使用 UAnOPT0.UAnTDL 位来进行。

- 起始位1 位
- 字符位7 位/8 位
- 校验位偶校验/奇校验/零校验/无校验
- 停止位1 位/2 位

图 12-2. UARTA 发送/接收数据格式

(a) 8 位数据宽度，LSB 优先，偶校验，停止位 1 位，通信数据：55H



(b) 8 位数据宽度，MSB 优先，偶校验，停止位 1 位，通信数据：55H



(c) 8 位数据宽度，MSB 优先，偶校验，停止位 1 位，通信数据：55H，TXDAn 反转



(d) 7 位数据宽度，LSB 优先，奇校验，停止位 2 位，通信数据：36H



(e) 8 位数据宽度，LSB 优先，无校验，停止位 1 位，通信数据：87H



12.5.2 SBF 发送/接收格式

V850ES/HE2 具有 SBF（同步中断区域）发送/接收控制功能，用于允许使用 LIN 功能。

备注 LIN 表示局部互联网，它是一个低速（1 至 20 kbps）串行通信协议，用于降低自动网络的成本。

LIN 是单主设备通信方式，一个主设备上最多可以连接 **15** 个从设备。

LIN 的从设备用于控制开关、制动器和传感器，这些设备通过 LIN 网与 LIN 主设备相连。

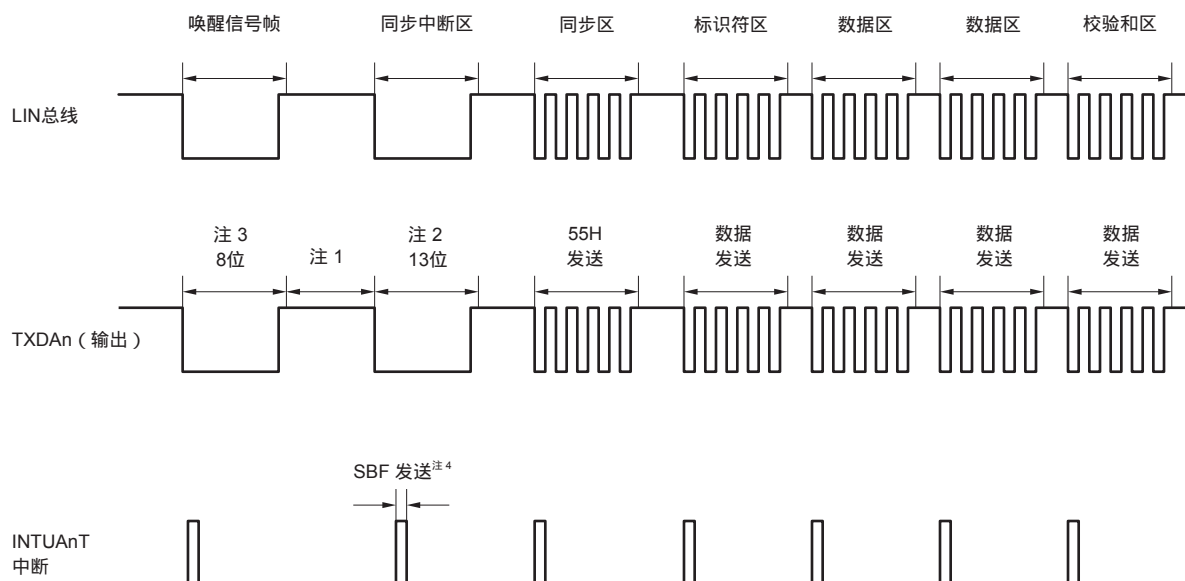
通常，LIN 主设备与一个网络如 CAN（控制器局域网）相连。

此外, LIN 总线采用单线方式, 通过符合 ISO9141 的收发器与各节点相连。

在 LIN 协议中，主设备发送带有波特率信息的一帧数据，从设备接收该数据并校正波特率误差。因此当从设备端的波特率误差在 $\pm 15\%$ 范围内时，可以进行通信。

图 12-3 和 12-4 概括了 LIN 的发送和接收操作。

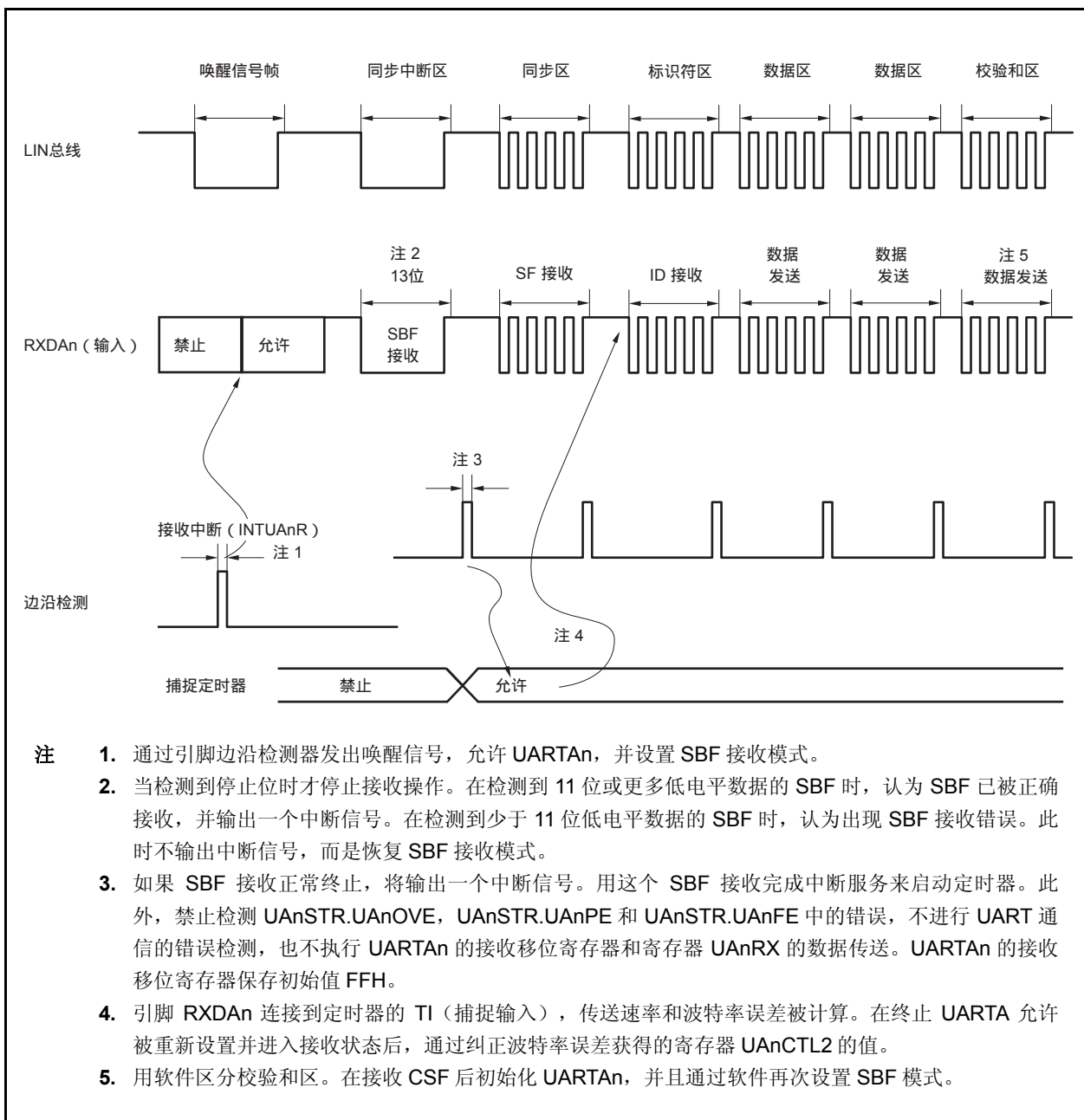
图 12-3. LIN 发送操作概要



注

1. 区域之间的间隔由软件控制。
2. 由硬件控制 SBF 输出。输出宽度为通过 UAnOPT0.UAnSBL2 至 UAnOPT0.UAnSBL0 位设置的位长度。如果要求更合适的宽度调整，可以通过使用 UAnCTLn.UAnBRS7 至 UAnCTLn.UAnBRS0 位来实现。
3. 在 8 位发送模式中，可通过发送 80H 替代唤醒信号帧。
4. 每次开始发送时，输出发送允许中断请求信号（INTUAnT）。每次开始发送 SBF 时，也要输出 INTUAnT 信号。

图 12-4. LIN 接收操作概要



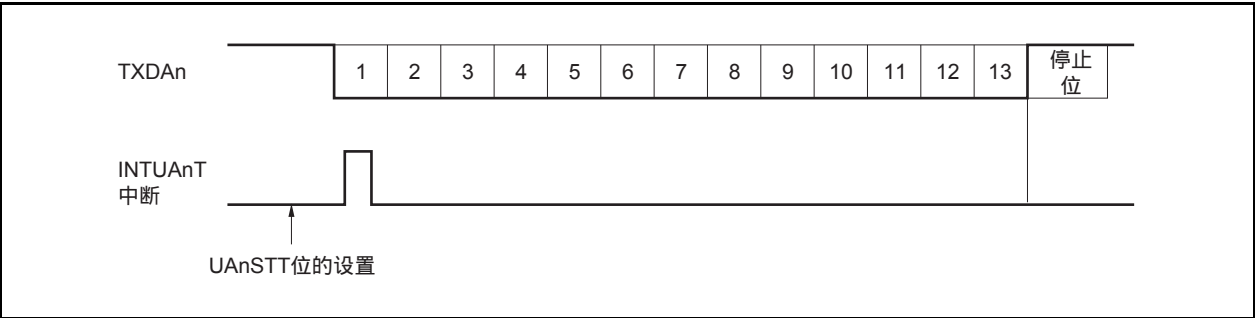
12.5.3 SBF 发送

当 UAnCTL0.UAnPWR 位 =UAnCTL0.UAnTXE 位 =1，进入允许发送状态，通过将 SBF 发送触发 (UAnOPT0.UAnSTT 位) 置 1 来开始 SBF 发送。

之后，输出位 13 至 20 (由 UAnOPT0.UAnSLS2 至 UAnOPT0.UAnSLS0 位指定) 的低电平。SBF 开始发送时，产生发送允许中断请求信号 (INTUAnT)。SBF 发送完之后，UAnSTT 位自动清零。之后，恢复 UART 发送模式。

直到下一个被发送的数据写入寄存器 UAnTX，或直到 SBF 发送触发器 (UAnSTT 位) 被设置，才停止发送。

图 12-5. SBF 发送



12.5.4 SBF 接收

通过设置 UAnCTL0.UAnPWR 位为 1，然后设置 UAnCTL0.UAnRXE 位为 1，来进入允许接收状态。

通过将 SBF 接收触发 (UAnOPT0.UAnSTR 位 1) 设置为 1 来设置 SBF 接收等待状态。

SBF 接收等待状态下，正如 UART 接收等待状态下一样，引脚 RXDAn 被监测，并开始起始位检测。

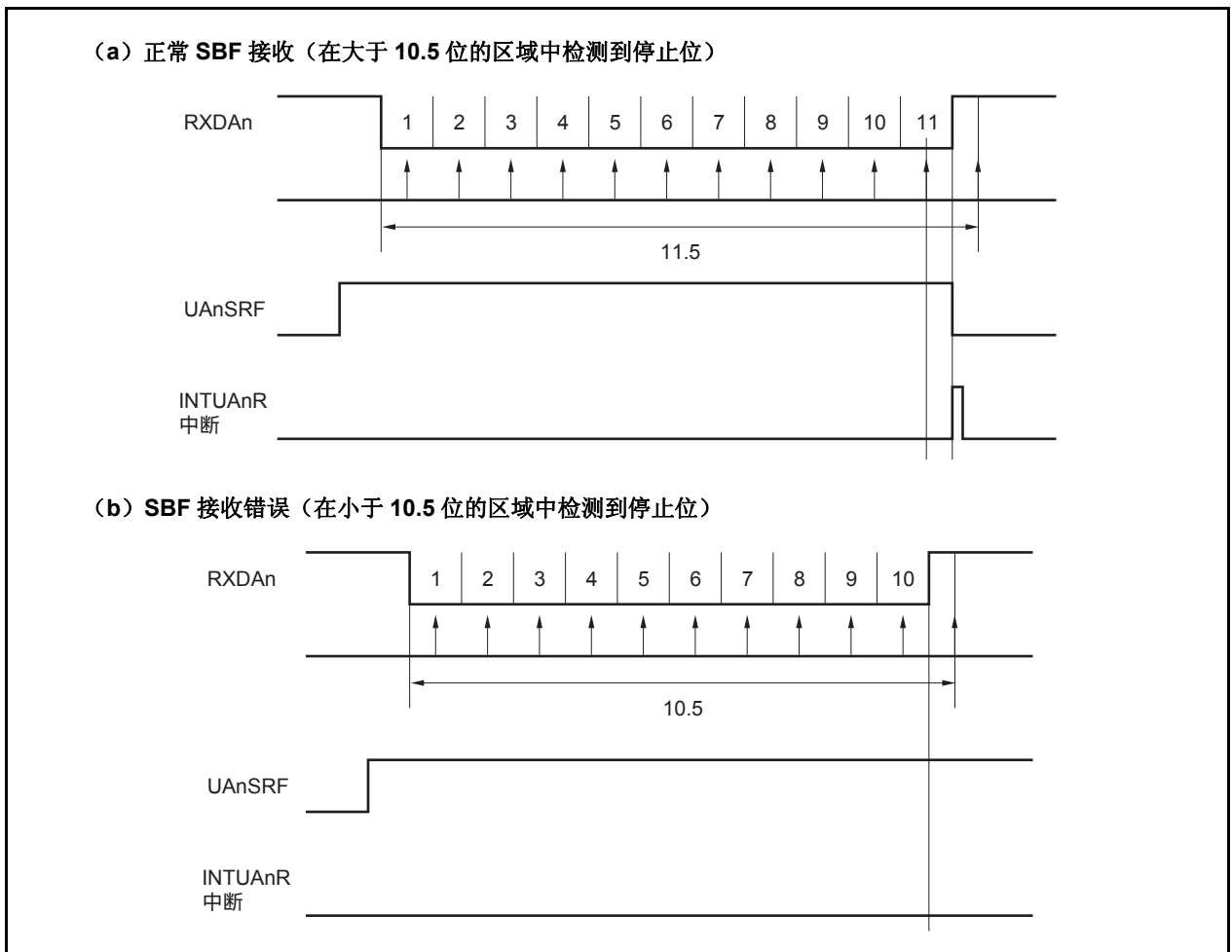
检测到起始位后，开始接收，内部计数器根据设定的波特率计算开始计数。

当接收到停止位时，如果 SBF 的宽度大于等于 11 位，判断为正常操作，并输出接收完成中断请求信号 (INTUAnR)。UAnOPT0.UAnSRF 位自动清零，SBF 接收结束。禁止 UAnSTR.UAnOVE，UAnSTR.UAnPE 和 UAnSTR.UAnFE 中的错误检测，不进行 UART 通信的错误检测。此外，不执行 UARTAn 的接收移位寄存器和寄存器 UAnRX 的数据传送，并且保存初始值 FFH。如果 SBF 的宽度小于等于 10 位，接收终止且错误处理中不输出中断，然后返回 SBF 接收模式。UAnSRF 位此时不清除。

注意事项 1. 如果 SBF 在数据接收期间被传输，则将发生帧错误。

2. 在 SBF 接收期间 (UAnSRF = 1) 请不要将 SBF 接收触发位 (UAnSRT) 及 SBF 发送触发位 (UAnSTT) 设为 1。

图 12-6. SBF 接收



12.5.5 UART 发送

设置 UAnCTL0.UAnPWR 位= 1，使引脚 TXDAn 输出高电平。

之后，通过将 UAnCTL0.UAnTXE 位设置为 1 来进入允许发送状态，通过将发送数据写入寄存器 UAnTX 来开始发送。起始位、校验位和停止位会自动被添加到数据中。

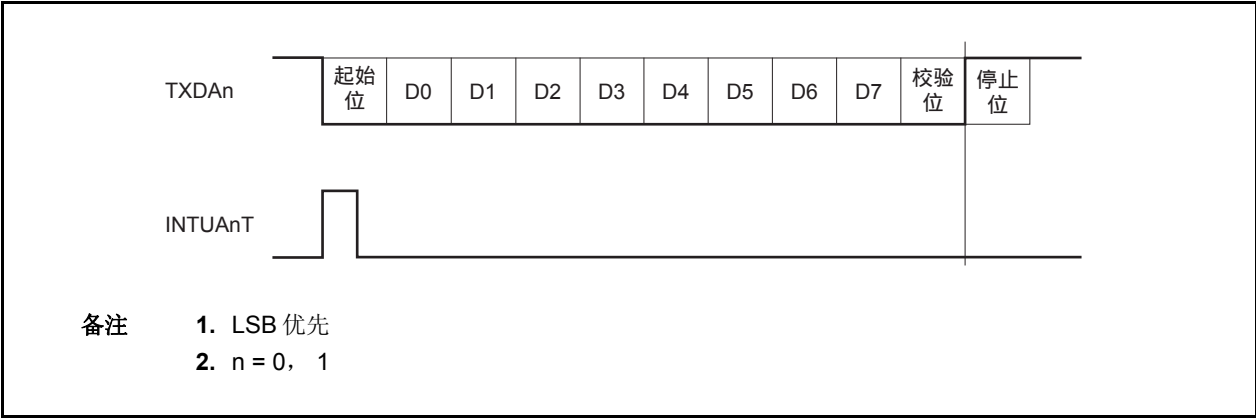
由于在 UARTAn 中没有提供 CTS（发送使能信号）输入引脚，所以使用某个端口检测发送终端的接收状态是否有效。

开始发送时，寄存器 UAnTX 中的数据被传送到发送移位寄存器 UARTAn。

完成寄存器 UAnTX 到发送移位寄存器 UARTAn 的数据的发送时，产生发送允许中断请求信号（INTUAnT），然后，发送移位寄存器 UARTAn 的内容被输出到引脚 TXDAn。

产生信号 INTUAnT 后，允许将下一个传送数据写入寄存器 UAnTX。

图 12-7. UART 发送



12.5.6 连续发送的过程

UARTAn 可以在 UARTAn 的发送移位寄存器开始移位操作时，将下一个发送数据写入寄存器 UAnTX。发送移位寄存器 UARTAn 的发送时序可通过发送允许中断请求信号 (INTUAnT) 判断。

通过发送过程中向寄存器 UAnTX 写入下一个被发送的数据，判断有效通信率。

在连续发送期间，发送数据被写入 UAnTX 寄存器并被转移到 UARTAn 发送移位寄存器后，在生成发送请求中断信号 (INTUAnT) 前不要将下一次的发送数据写入 UAnTX 寄存器。如果在发送请求中断信号生成前将一个值写入 UAnTX 寄存器，那么先前设置的发送数据将会被最近的发送数据覆盖。

注意事项 在连续发送过程中执行发送初始化，确保 UAnSTR.UAnTSF 位为 0，然后进行初始化。当 UAnTSF = 1 时初始化数据，其发送过程不能保证。

在连续发送的情况下，下个数据中从停止位至起始位的通信率将会较正常通信率扩展两个时钟。

图 12-8. 连续发送操作流程

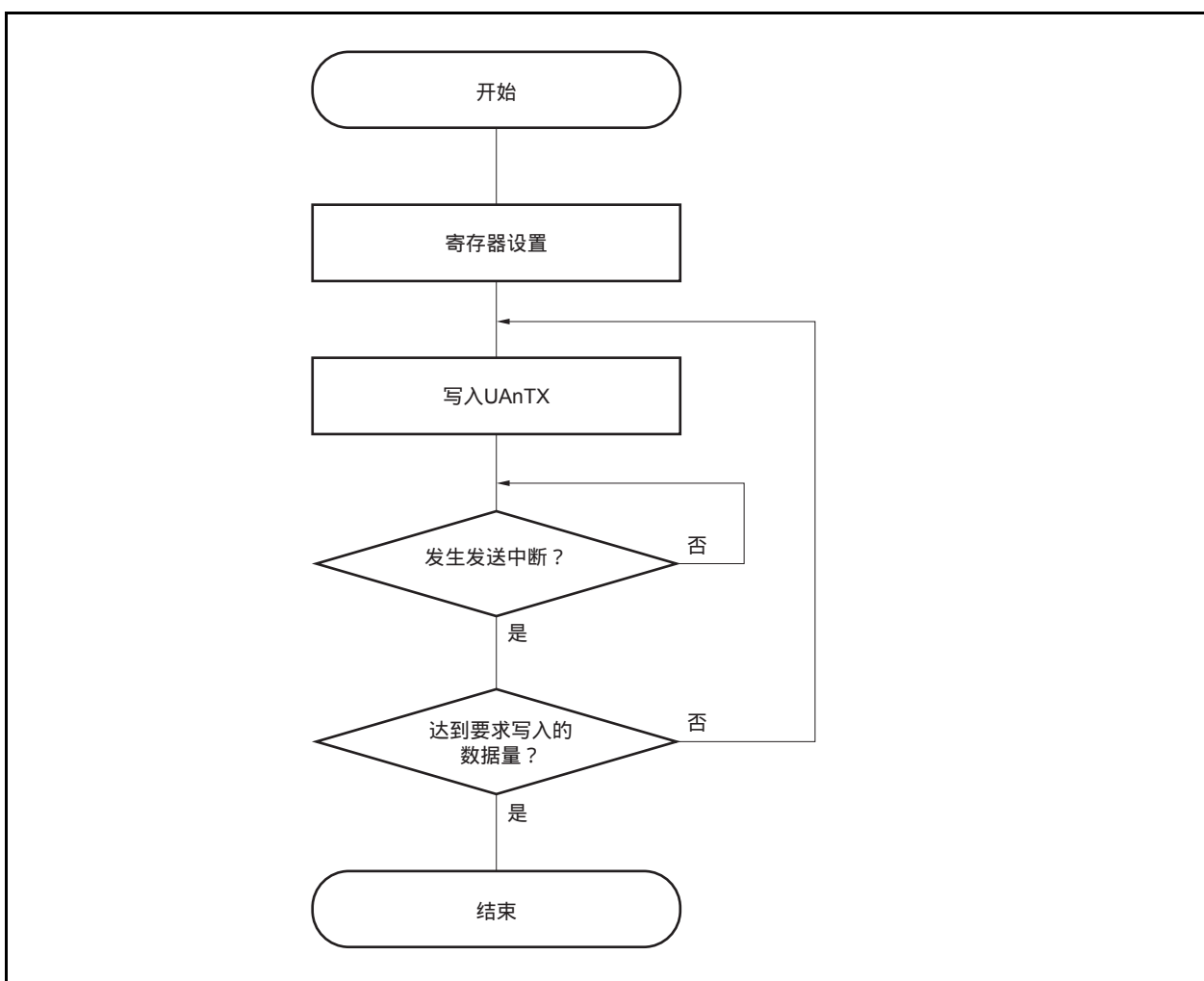
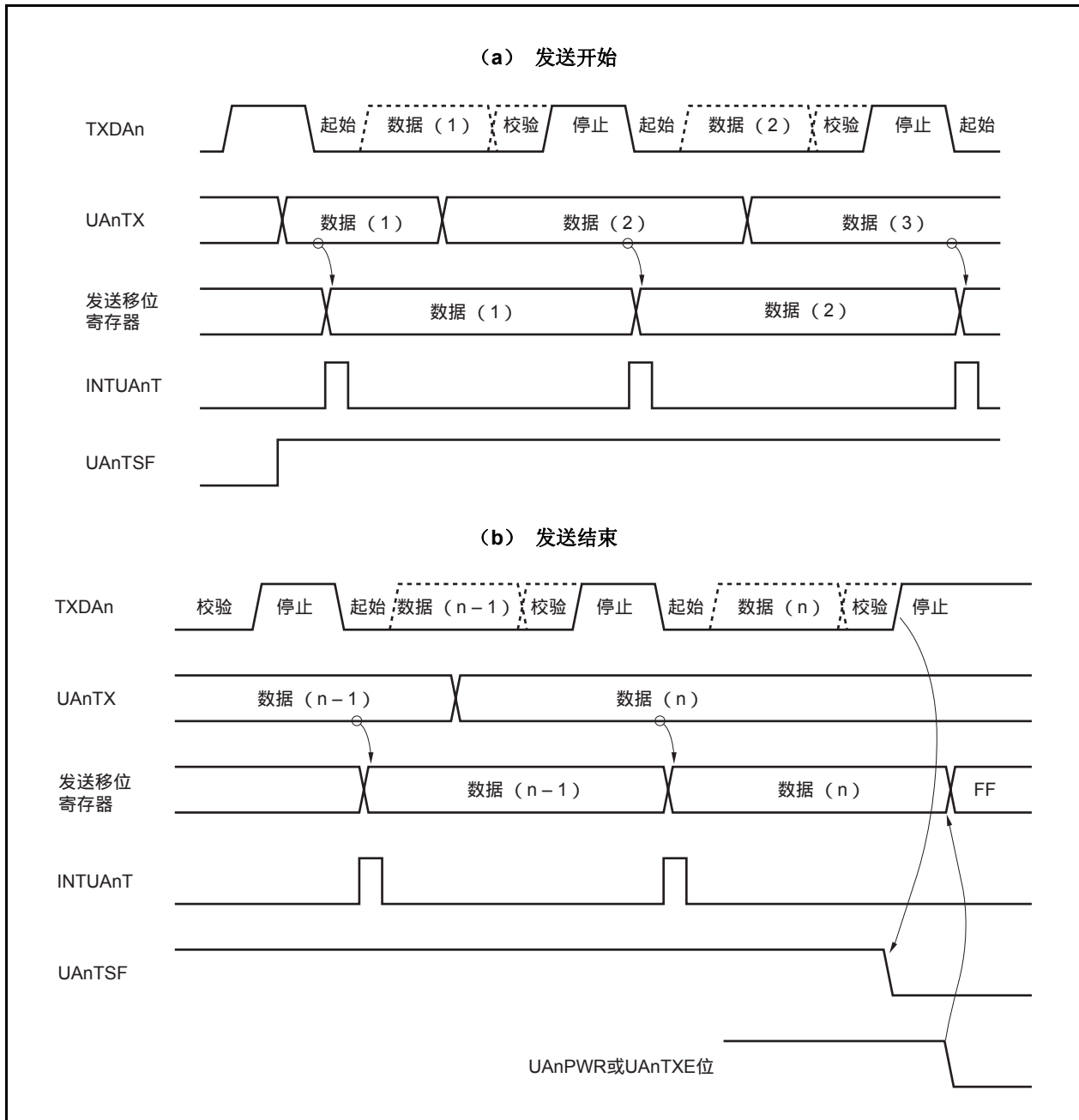


图 12-9. 连续发送操作时序



12.5.7 UART 接收

通过先设置 UAnCTL0.UAnPWR 位为 1，然后设置 UAnCTL0.UAnRXE 位为 1 来进入接收等待状态。接收等待状态下，引脚 RXDAn 被监控并执行检测起始位。

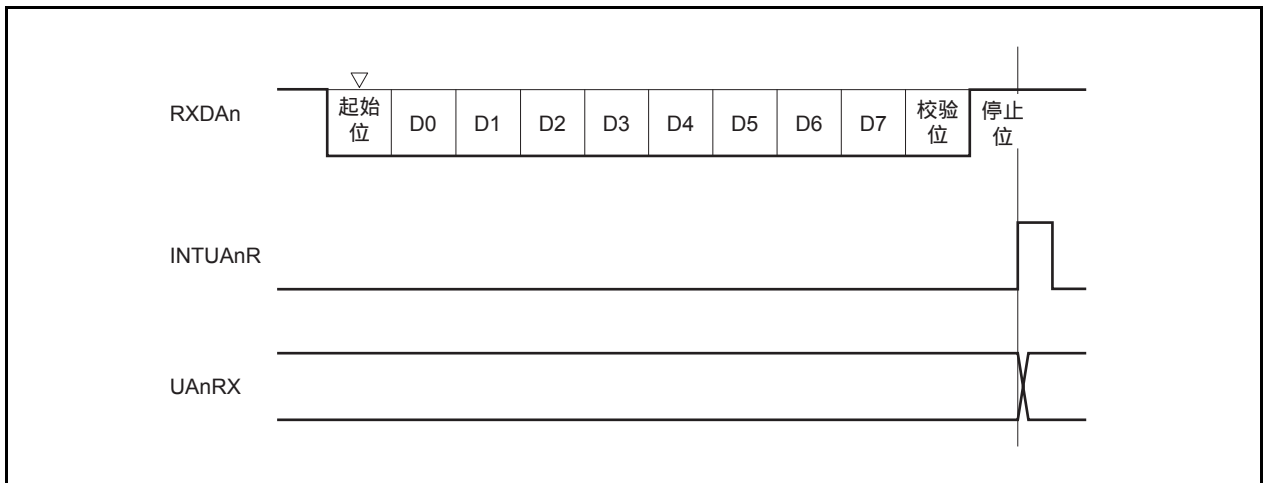
检测起始位使用 2 步检测子程。

首先，检测到引脚 RXDAn 的上升沿，在下降沿开始采样。如果在起始位采样点引脚 RXDAn 为低电平，则认为检测到起始位。检测到起始位后，接收操作开始，串行数据根据设定好的波特率保存到 UARTAn 的接收移位寄存器。

当接收到停止位时，输出接收完成中断请求信号 (INTUAnR)，UARTAn 的接收移位寄存器的数据被写入寄存器 UAnRX。但是，如果产生溢出错误 (UAnSTR.UAnOVE)，此时接收到的数据不被写入寄存器 UAnRX，而是被丢弃。

即使在接收时产生校验错误 (UAnSTR.UAnPE) 或帧错误，继续接收直到接收到第一个停止位，并在接收完全后输出 INTUAnR。

图 12-10. UART 接收



- 注意事项**
1. 即使产生接收错误，也要确保读取寄存器 UAnRX。如果寄存器 UAnRX 不被读取，在接收下一个数据期间会产生溢出错误，而且接收错误会不确定的连续发生。
 2. 在接收期间并假定只有一个停止位时，操作被执行，第二个停止位被忽略。
 3. 在接收完成，产生接收完成中断请求信号后，读取寄存器 UAnRX，并将 UAnPWR 位或 UAnRXE 位清零。如果 UAnPWR 位或 UAnRXE 位在信号 INTUAnR 产生前被清零，寄存器 UAnRX 的读取值不被保证。
 4. 如果 UARTAn 的接收完成处理过程（产生信号 INTUAnR）和 UAnPWR = 0 或 UAnRXE = 0 产生冲突，无论数据是否被存储到寄存器 UAnRX 中都产生信号 INTUAnR。
为了在不等待 INTUAnR 信号产生的情况下完成接收，设置中断控制寄存器 (UAnRIC) 的中断屏蔽标志 (UAnRMK) 为 1，以及 UAnPWR 位 = 0 或 UAnRXE 位 = 0 后，确保寄存器 UAnRIC 的中断请求标志 (UAnRIF) 清零。

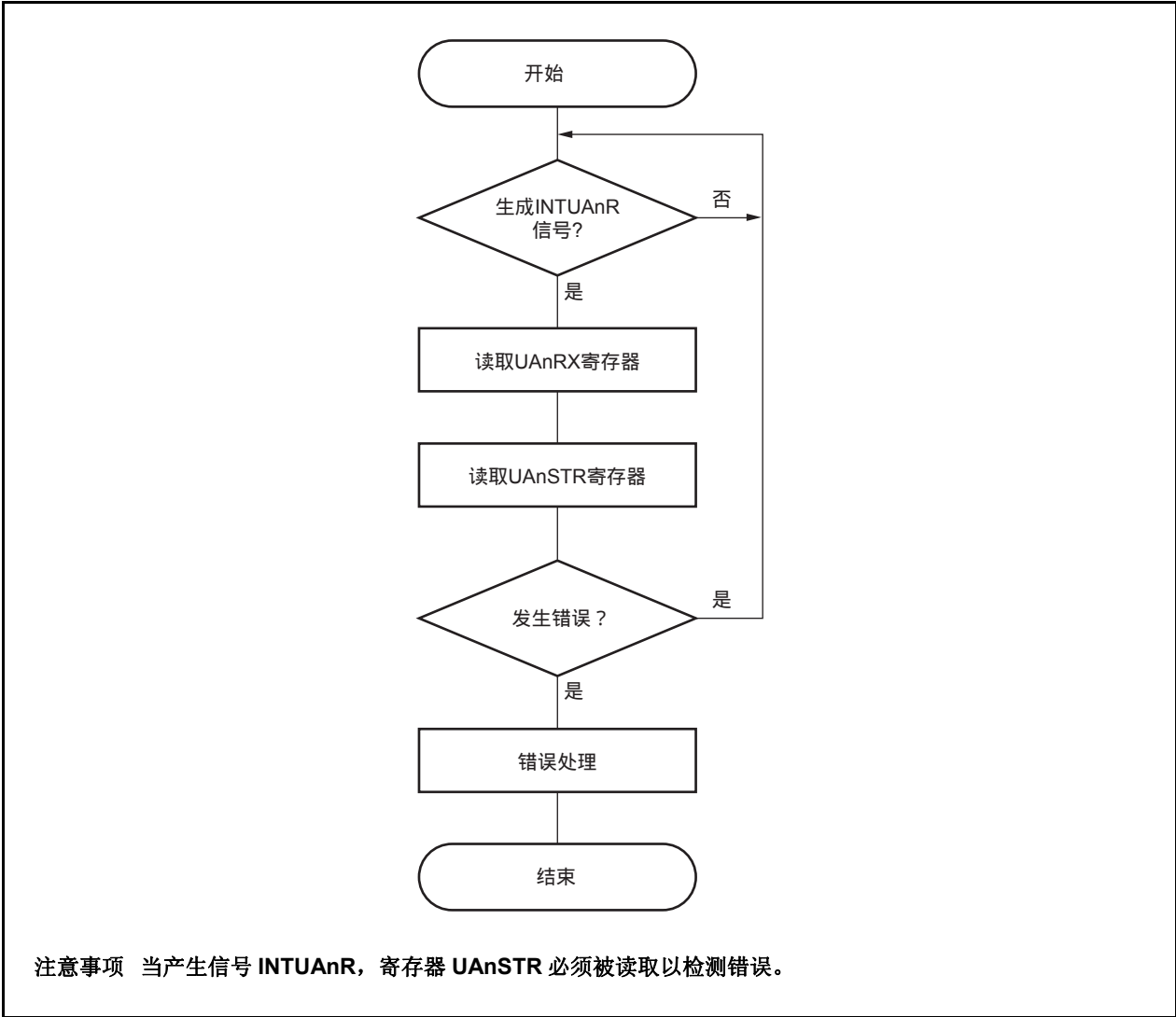
12.5.8 接收错误

接收过程中的错误有三种：校验错误，帧错误和溢出错误。数据接收结果错误标志设置在寄存器 UAnSTR 中，并且当产生错误时，输出接收完成中断请求信号（INTUAnR）。

可以通过读取寄存器 UAnSTR 中的内容判断接收过程出现何种错误。

在读取接收错误标志后，通过对其写入 0 来清除接收错误标志。

- 接收数据读取流程



- 接收错误原因

错误标志	接收错误	原因
UAnPE	校验错误	接收的校验位与设置不匹配
UAnFE	帧错误	没检测到停止位
UAnOVE	溢出错误	在数据被从接收缓冲中读取前，接收到下一个完整数据

当产生接收错误，根据错误类型进行如下的相应处理。

- 校验错误

如果传输过程中由于噪声等原因接收到的错误数据，丢弃该数据并重新发送。

- 帧错误

在发送方和接收方之间可能产生波特率错误，或者起始位可能被错误地检测。由于帧错误是通信格式的致命错误，则应在发送方检查操作停止，并对双方重新初始化，然后重新开始通信。

- 溢出错误

由于在读取接收到的数据前，下一次接收完成，1 帧的数据将被丢弃。如果需要，重传该数据。

注意事项 如果连续接收过程中产生接收错误中断，在下一次接收完成前必须读取寄存器中必需读取的内容，并执行错误处理。

12.5.9 校验类型和操作

注意事项 当使用 LIN 功能，将 UAnCTL0.UAnPS1 及 UAnCTL0.UAnPS0 位固定为 00。

通信数据的校验位用于检测位错误。通常，发送方和接收方使用同种校验方式。

奇校验和偶校验模式下，可以检测到奇数位错误。零校验和无校验模式下，不能检测错误。

(a) 偶校验

(i) 发送期间

通过在传送数据，包括校验位中控制值为“1”的位的个数，使得数据中“1”的个数为偶数。校验位的值如下。

- 如果发送数据有奇数个“1”：1
- 如果发送数据有偶数个“1”：0

(ii) 接收期间

计算接收数据，包括校验位中“1”的个数。如果“1”的个数为奇数，则产生校验错误。

(b) 奇校验

(i) 发送期间

与偶校验相反，通过在传送数据，包括校验位中控制值为“1”的位的个数，使得数据中“1”的个数为奇数。校验位的取值如下。

- 如果发送数据有奇数个“1”：1
- 如果发送数据有偶数个“1”：0

(ii) 接收期间

计算接收数据（包括校验位）中“1”的个数。如果“1”的个数为偶数，则产生校验错误。

(c) 零校验

无论发送数据为何值，发送期间校验位总是为 0。

接收数据期间不检测校验位。因此无论校验位是 0 还是 1，都不会产生校验错误。

(d) 无校验

没有校验位被发送数据中。

在没有校验位的情况下执行接收。由于没有校验位，因此也不会产生校验错。

12.5.10 接收数据的噪声过滤器

使用由预分频模块输出的基准时钟采样 RXDAn 信号。

如果两次采样值相同，则匹配检测器的输出会发生变化，并把采样的 RXDAn 信号作为输入数据。因此，不超过 2 个时钟宽度的数据被判断为噪声，不向内部电路发送（参见图 12-12）。参见 1 12.6 (1) 基本时钟 关于基本时钟。

此外，如图 15-12-11 所示的电路结构，接收操作的内部过程与外部信号状态相比延迟三个时钟才执行。

图 12-11. 噪声过滤器电路

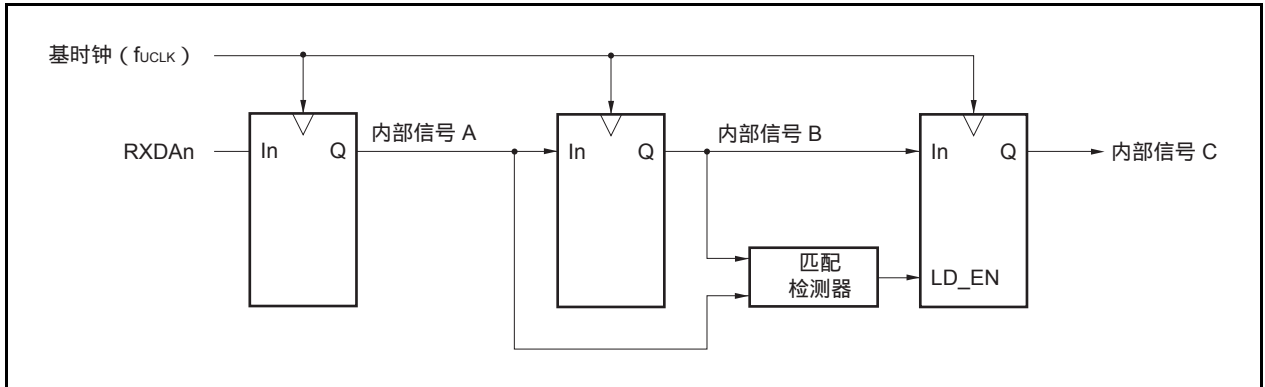
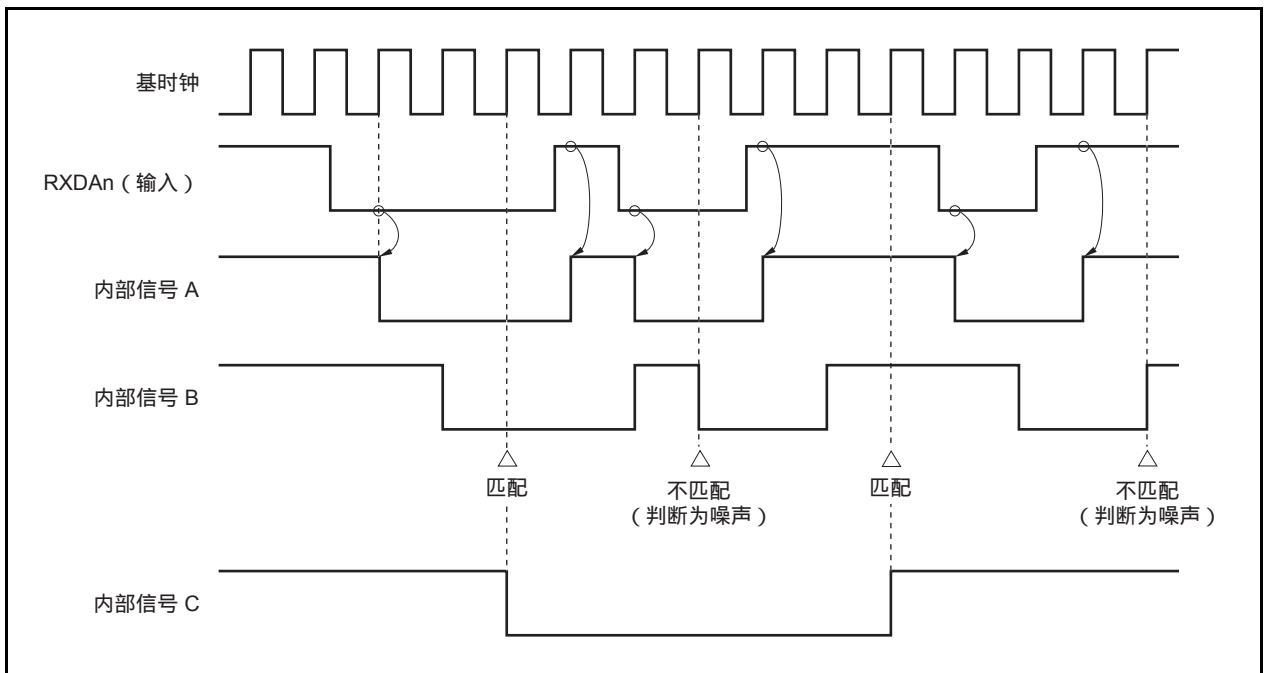


图 12-12. RXDAn 信号判断为噪声的时序



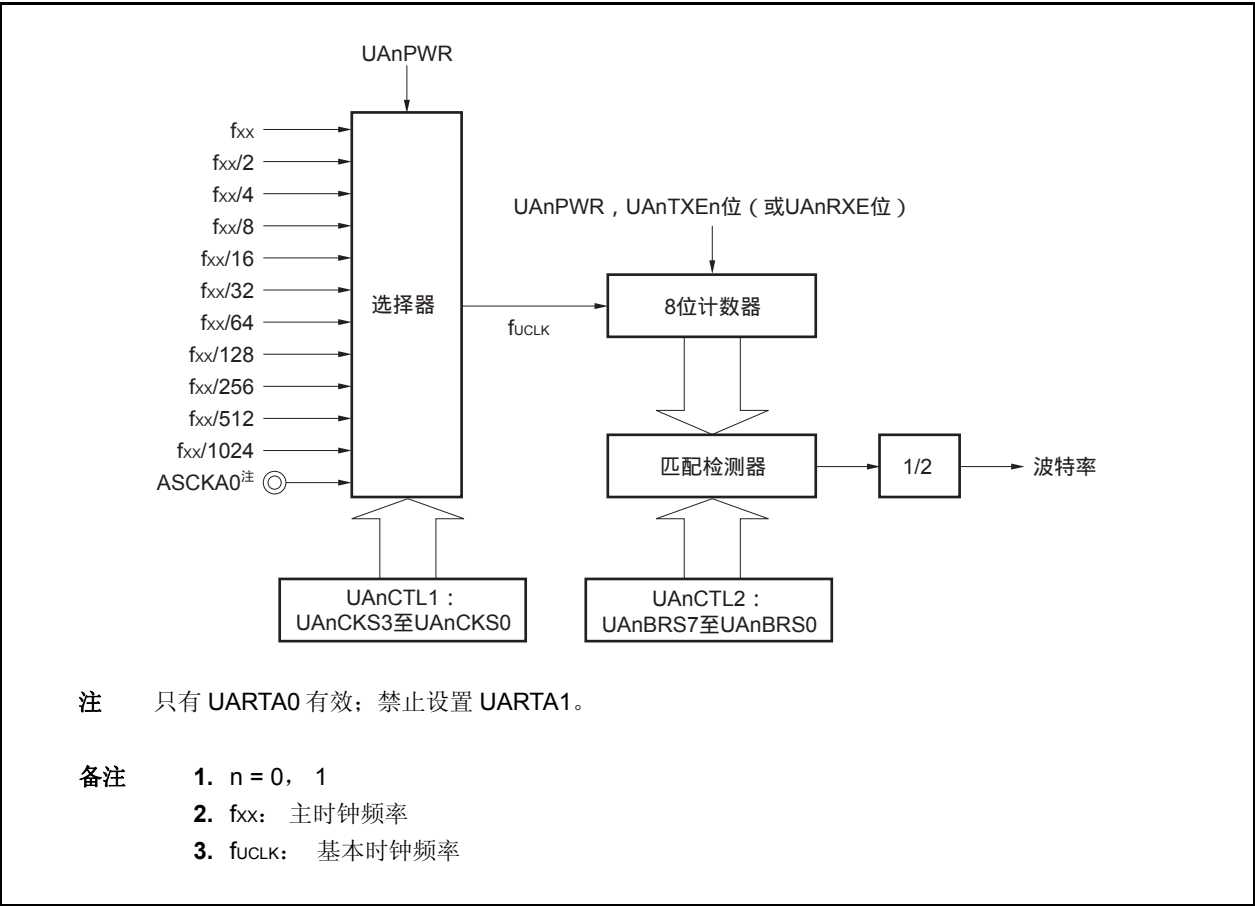
12.6 专用波特率发生器

专用波特率发生器由一个源时钟选择器模块和一个 8 位可编程计数器组成，用于在 UARTAn 的发送/接收期间产生一个串行时钟。关于串行时钟，可以为每个通道选择专用波特率发生器输出。

一个 8 位计数器用于发送，另一个用于接收。

(1) 波特率发生器的配置

图 12-13. 波特率发生器的配置



(a) 基本时钟

当 UAnCTL0.UAnPWR 位置 1，由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择的时钟会提供给 8 位计数器。该时钟称为基本时钟 (f_{UCLK})。

(b) 串行时钟发生器

通过设置寄存器 UAnCTL1 和 UAnCTL2 ($n = 0, 1$) 产生串行时钟。
通过 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择基本时钟。
通过 UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位设置 8 位计数器的分频值。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

该寄存器是 8 位寄存器，用于选择 UARTAn 的基本时钟。

该寄存器可以以 8 位为单位进行读取或写入。

复位后该寄存器被设置为 00H。

注意事项 在复写寄存器 UAnCTL1 前，将 UAnCTL0.UAnPWR 位清零。

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

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	基本时钟 (f _{UCLK}) 选择
0	0	0	0	fxx
0	0	0	1	fxx/2
0	0	1	0	fxx/4
0	0	1	1	fxx/8
0	1	0	0	fxx/16
0	1	0	1	fxx/32
0	1	1	0	fxx/64
0	1	1	1	fxx/128
1	0	0	0	fxx/256
1	0	0	1	fxx/512
1	0	1	0	fxx/1,024
1	0	1	1	外部时钟 ^注 (ASCKA0引脚)
其他情况				禁止设置

注 只有 UARTA0 有效；禁止设置 UARTA1。

备注 1. fxx: 主时钟频率
 2. n = 0, 1

(3) UARTAn 控制寄存器 2 (UAnCTL2)

该寄存器是 8 位寄存器，用于选择 UARTAn 的波特率（串行传送速度）时钟。

该寄存器可以以 8 位为单位进行读取或写入。

复位后该寄存器被设置为 FFH。

注意事项 在复写寄存器 UAnCTL2 前，将 UAnCTL0.UAnPWR 位清零或将 UAnTXE 和 UAnRXE 位清为 00。

复位后：FFH R/W 地址： UA0CTL2 FFFFFFFA02H, UA1CTL2 FFFFFFFA12H

	7	6	5	4	3	2	1	0
UAnCTL2	UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0

UAn BRS7	UAn BRS6	UAn BRS5	UAn BRS4	UAn BRS3	UAn BRS2	UAn BRS1	UAn BRS0	Default (k)	串行 时钟
0	0	0	0	0	0	·	·	·	禁止 设置
0	0	0	0	0	1	0	0	4	f _{UCLK} /4
0	0	0	0	0	1	0	1	5	f _{UCLK} /5
0	0	0	0	0	1	1	0	6	f _{UCLK} /6
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	f _{UCLK} /252
1	1	1	1	1	1	0	1	253	f _{UCLK} /253
1	1	1	1	1	1	1	0	254	f _{UCLK} /254
1	1	1	1	1	1	1	1	255	f _{UCLK} /255

- 备注 1. f_{UCLK}: 由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择的时钟频率
 2. n = 0, 1

(4) 波特率

通过下列公式计算波特率。

$$\text{波特率} = \frac{f_{\text{CLK}}}{2 \times k} [\text{bps}]$$

当使用内部时钟时，公式如下（在 UARTA0 工作时使用引脚 ASCKA0 作为时钟，使用上述公式计算）。

$$\text{波特率} = \frac{f_{\text{xx}}}{2^{m+1} \times k} [\text{bps}]$$

备注 f_{CLK} = 通过 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择的基本时钟频率
 f_{xx} : 主时钟频率
 m = 由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位设置的数值（ $m = 0$ 到 10）
 k = 由 UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位设置的数值（ $k = 4$ 到 255）

通过下列公式计算波特率误差。

$$\begin{aligned} \text{误差}(\%) &= \left(\frac{\text{实际波特率 (有误差的波特率)}}{\text{预期波特率 (正确的波特率)}} - 1 \right) \times 100 [\%] \\ &= \left(\frac{f_{\text{CLK}}}{2 \times k \times \text{预期波特率}} - 1 \right) \times 100 [\%] \end{aligned}$$

当使用内部时钟时，公式如下（在 UARTA0 工作时使用引脚 ASCKA0 作为时钟，使用上述公式计算波特率误差）。

$$\text{误差}(\%) = \left(\frac{f_{\text{xx}}}{2^{m+1} \times k \times \text{预期波特率}} - 1 \right) \times 100 [\%]$$

注意事项 1. 发送期间必须保持波特率误差在接收端允许的误差范围内。
 2. 接收期间，波特率误差必须满足“（5）接收期间允许的波特率范围”中所描述的范围。

为了设置波特率，进行如下计算并设置寄存器 UAnCTL1 和 UAnCTL2（当使用内部时钟）。

<1> 设置 $k = f_{xx} / (2 \times \text{预期波特率})$ 。Set $m = 0$ 。

<2> 设置 $k = k/2$ 和 $m = m + 1$ ，其中 $k \geq 256$ 。

<3> 重复步骤 <2> 直至 $k < 256$ 。

<4> 将 k 取整。

如果取整后 $k = 256$ ，则再次执行步骤 <2>（ k 将会变成 128）。

<5> 将 m 设置到 UAnCTL1 寄存器，并将 k 设置到 UAnCTL2 寄存器。

例：当 $f_{xx} = 20 \text{ MHz}$ 且预期波特率 = 153, 600 bps

<1> $k = 20,000,000 / (2 \times 153,600) = 65.10\dots$ ， $m = 0$

<2>，<3> $k = 65.10\dots < 256$ ， $m = 0$

<4> 设置寄存器 UAnCTL2 的值： $k = 65 = 41\text{H}$ ，设置寄存器 UAnCTL1 的值： $m = 0$

实际波特率 $= 20,000,000 / (2 \times 65)$
 $= 153,846 \text{ [bps]}$

波特率误差 $= \{20,000,000 / (2 \times 65 \times 153,600) - 1\} \times 100$
 $= 0.160 \text{ [%]}$

波特率设置的典型示例如下。

表 12-3. 波特率发生器的数据设置

波特率 (bps)	$f_{xx} = 20 \text{ MHz}$			$f_{xx} = 16 \text{ MHz}$			$f_{xx} = 10 \text{ MHz}$		
	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)
300	08H	82H	0.16	0AH	1AH	0.16	07H	82H	0.16
600	07H	82H	0.16	0AH	0DH	0.16	06H	82H	0.16
1,200	06H	82H	0.16	09H	0DH	0.16	05H	82H	0.16
2,400	05H	82H	0.16	08H	0DH	0.16	04H	82H	0.16
4,800	04H	82H	0.16	07H	0DH	0.16	03H	82H	0.16
9,600	03H	82H	0.16	06H	0DH	0.16	02H	82H	0.16
19,200	02H	82H	0.16	05H	0DH	0.16	01H	82H	0.16
31,250	01H	A0H	0.00	01H	80H	0.00	00H	A0H	0.00
38,400	01H	82H	0.16	00H	D0H	0.16	00H	82H	0.16
76,800	00H	82H	0.16	03H	0DH	0.16	00H	41H	0.16
153,600	00H	41H	0.16	02H	0DH	0.16	00H	21H	-1.36
312,500	00H	20H	0.00	00H	1AH	-1.54	00H	10H	0.00

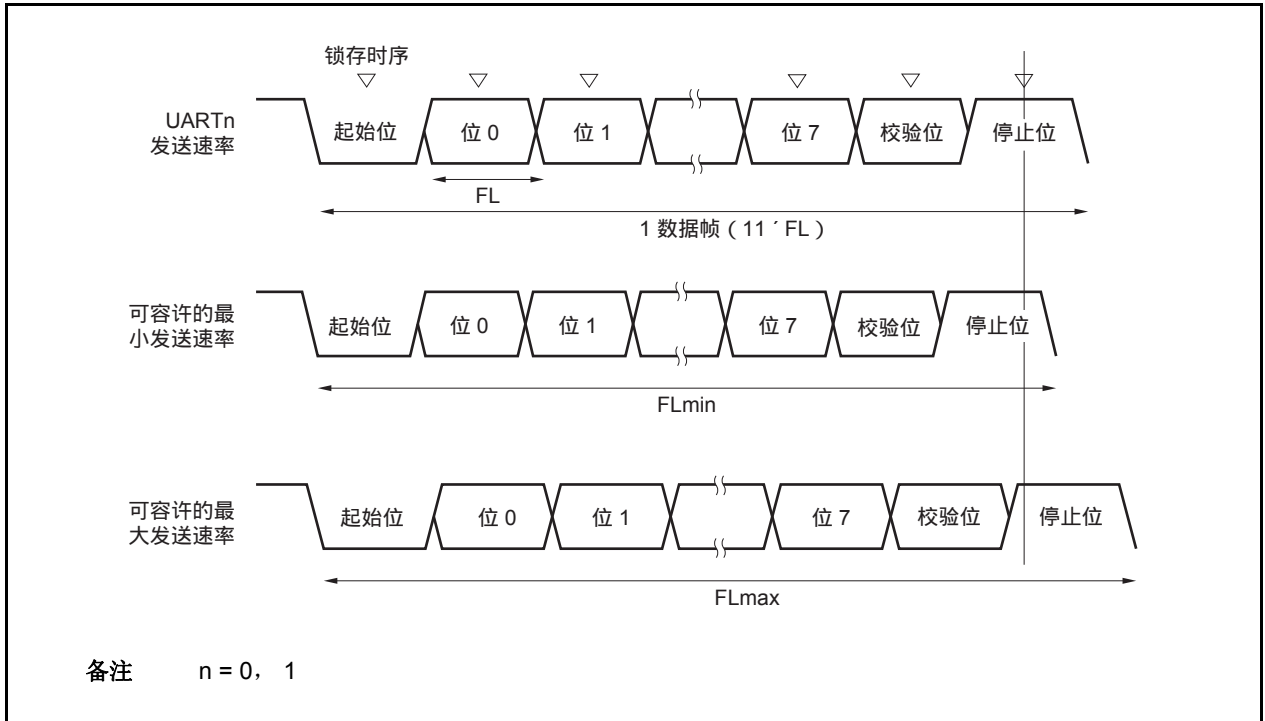
备注 f_{xx} : 主时钟频率
 ERR: 波特率误差 (%)

(5) 接收期间允许的波特率范围

接收期间来自发送端的波特率可允许的误差如下所示。

注意事项 必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。

图 12-14. 接收期间允许的波特率范围



如图 12-14 所示，当检测到起始位后，接收数据的锁存时序由寄存器 UAnCTL2 设置的计数器确定。如果数据的最后一位（停止位）满足该锁存时序，则发送数据可被正确接收。

假定接收数据为 11 位，各项理论值计算如下。

$$FL = (\text{Brate})^{-1}$$

Brate: UARTAn 波特率 (n = 0, 1)

k: UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位的设置值 (n = 0, 1)

FL: 1 位数据长度

锁存时序极限: 2 个时钟

$$\text{可允许的最小数据帧宽度} : FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

因此在发送端设定的可接收的最大波特率如下所示。

$$BR_{\max} = (FL_{\min}/11)^{-1} = \frac{22k}{21k + 2} \text{ Brate}$$

同样的，可以按以下公式获得最大允许传送速率范围。

$$\frac{10}{11} \times FL_{\max} = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FL_{\max} = \frac{21k - 2}{20k} FL \times 11$$

因此在接收端可接收的最小波特率如下所示。

$$BR_{\min} = (FL_{\max}/11)^{-1} = \frac{20k}{21k - 2} \text{ Brate}$$

可通过上述最小和最大波特率公式计算在 UARTAn 与接收端之间允许的波特率误差，如下所示。

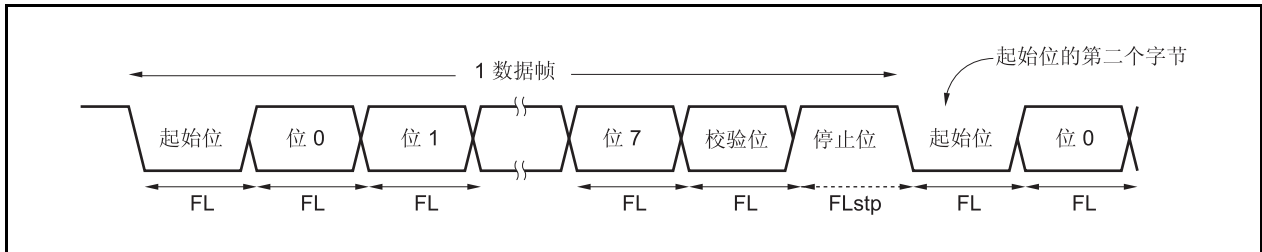
表 12-4. 允许的最大/最小波特率误差

分频比（k）	允许的最大波特率误差	允许的最小波特率误差
4	+2.32%	-2.43%
8	+3.52%	-3.61%
20	+4.26%	-4.30%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.72%

- 备注**
1. 接收准确度取决于每帧上的位数、输入时钟频率和分频比（k）。输入时钟频率和分频比（k）越大，准确度就更高。
 2. k: UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位的设置值（n = 0, 1）。

(6) 连续发送期间的波特率

在连续发送数据期间，从停止位到下一个起始位的传送速率一般为 2 个基本时钟。不过，由于时序初始化是由接收端通过开始位检测来执行，因此传送结果不会受到任何影响。

图 12-15. 连续发送期间的传送速率

假设 1 位数据宽度为 FL；停止位长度：FLstp；基本时钟频率：f_{UCLK}，我们可以得到以下的公式。

$$FLstp = FL + 2/f_{UCLK}$$

因此，连续发送期间的发送速率如下：

$$\text{发送速率} = 11 \times FL + (2/f_{UCLK})$$

12.7 注意事项

- (1) 当 UARTAn 所使用的时钟停止工作（例如，在 IDLE1, IDLE2, 或 STOP 模式中），操作将停止，每个寄存器会在时钟停止前立即保持其所具有的值。TXDAn 引脚输出在时钟停止前也会立即保持和输出它所具有的值。然而，时钟恢复后，操作不被保证。因此，时钟恢复后，应该通过设置 UAnCTL0.UAnPWR, UAnCTL0.UAnRXEn 和 UAnCTL0.UAnTXEn 位为 000 来使电路初始化。
- (2) 引脚 RXDA1 和 KR7 不能同时使用。要 RXDA1 引脚，则不要使用 KR7 引脚。要使用 KR7 引脚，则不要使用 RXDA1 引脚（推荐设置 PFC91 为 1 且 PFCE91 为 0）。
- (3) 在 UARTAn 中，不会产生由于通信错误引起的中断。在通信期间读取 UAnSTR 寄存器以检查错误。
- (4) 按照下述次序启动 UARTAn。
 - <1> 设置 UAnCTL0.UAnPWR 为 1。
 - <2> 设置端口。
 - <3> 设置 UAnCTL0.UAnTXE 位为 1, UAnCTL0.UAnRXE 位为 1。
- (5) 按照下述次序停止 UARTAn。
 - <1> 设置 UAnCTL0.UAnTXE 位为 0, UAnCTL0.UAnRXE 位为 0。
 - <2> 设置端口，设置 UAnCTL0.UAnPWR 位为 0（如果端口设置没有改变则不会有问题）。
- (6) 在发送模式中（UAnCTL0.UAnPWR 位= 1 且 UAnCTL0.UAnTXE 位= 1），不要通过软件将相同的值写到寄存器 UAnTX 中，因为发送操作从值写入寄存器就开始了。要连续发送相同的值，则复写同样的值。
- (7) 在连续发送中，从停止位到下一个起始位的通信速率比正常情况下的延长了 2 个基本时钟。然而，接收端通过检测起始位来初始化时序，所以接收结果不受影响。
- (8) 如果在片上调试（OCD）模式中执行中止命令且如果 UART 接收了数据，那么将会发生高优先中断。

第 13 章 3 线长度可变串行 I/O (CSIB)

V850ES/HE2 具有 3 线串行接口 (CSIB) 的两个通道。

13.1 特征

- 传输速率: 最大 8 Mbps。 ($f_{xx} = 20 \text{ MHz}$, 使用内部时钟)
- 主/从模式均可选
- 8 位至 16 位发送, 3 线串行接口
- 中断请求信号 (INTCBnT, INTCBnR) $\times 2$
- 串行时钟和数据相位可转换
- 可以 1 位为单位在 8 至 16 位之间选择通信数据长度
- 发送数据 MSB-/LSB-first 可切换
- 3 线通信
 - SOBn: 串行数据输出
 - SIBn: 串行数据输入
 - $\overline{\text{SCKBn}}$: 串行时钟 I/O

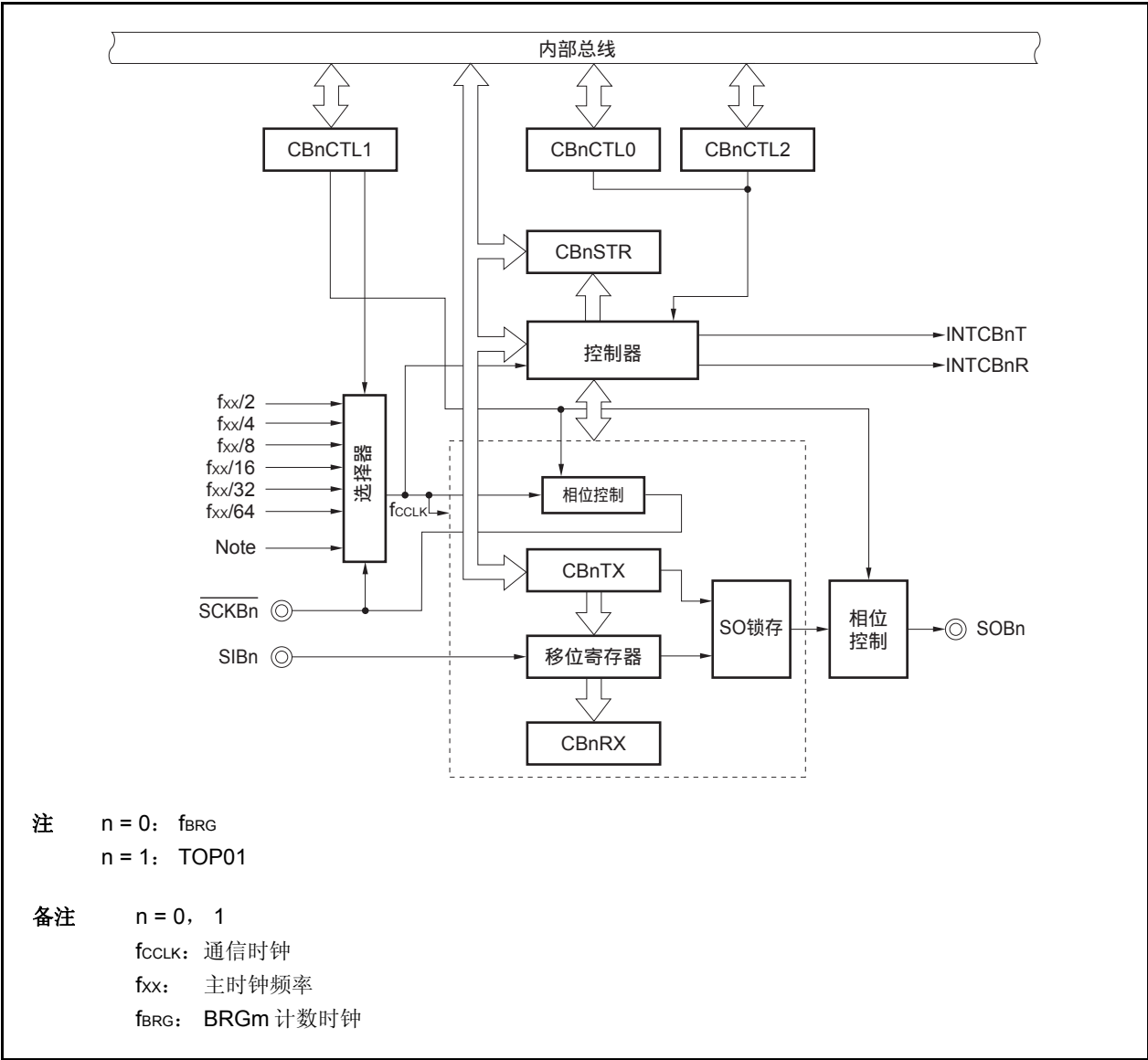
可指定发送模式, 接收模式和发送/接收模式

备注 $n = 0, 1$

13.2 配置

CSIBn 的框图如下。

图 13-1. CSIBn 的框图



CSIBn 包含以下硬件。

表 13-1. CSIBn 的配置

项目	配置
寄存器	CSIBn 数据接收寄存器 (CBnRX) CSIBn 数据发送寄存器 (CBnTX)
控制寄存器	CSIBn 控制寄存器 0 (CBnCTL0) CSIBn 控制寄存器 1 (CBnCTL1) CSIBn 控制寄存器 2 (CBnCTL2) CSIBn 状态寄存器 (CBnSTR)

(1) CSIBn 数据接收寄存器 (CBnRX)

该寄存器是 16 位缓冲寄存器，用于保持接收数据。

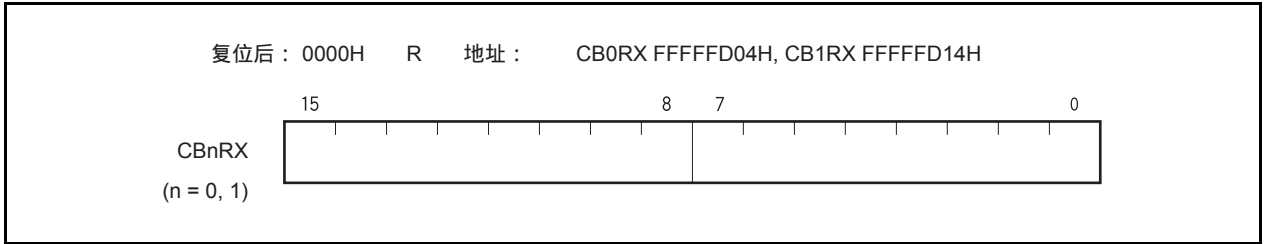
该寄存器是只读的，可以以 16 位为单位进行读取。

在允许接收的状态下，通过读取 CBnRX 寄存器开始接收操作。

如果通信数据的长度是 8 位，该寄存器的低 8 位是只读的，就如同读取 CBnRXL 寄存器，可以以 8 位为单位进行读取。

复位后该寄存器被设置为 0000H。

除了复位信号，通过清除（置 0）寄存器 CBnCTL0 的 CBnPWR 位，寄存器 CBnRX 可被初始化。



(2) CSIB 数据发送寄存器 (CBnTX)

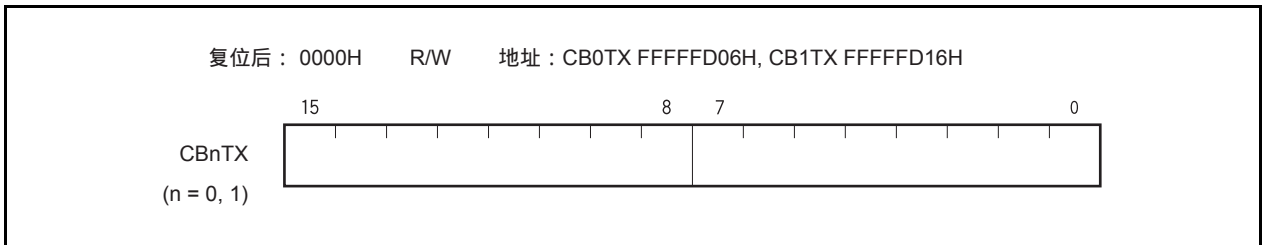
寄存器是 16 位缓冲寄存器，用于写入 CSIBn 以发送数据。

该寄存器可以以 16 位为单位进行读取或写入。

在允许发送的状态下，通过将数据写到 CBnTX 寄存器中开始发送操作。

如果通信数据的长度是 8 位，就如同读取 CBnRXL 寄存器，该寄存器的低 8 位只能以 8 位为单位进行读取。

复位后该寄存器被设置为 0000H。



备注

通信开始条件如下所示。

发送模式（CBnTXE 位 = 1，CBnRXE 位 = 0）：

写入 CBnTX 寄存器

发送/接收模式（CBnTXE 位 = 1，CBnRXE 位 = 1）：

写入 CBnTX 寄存器

接收模式（CBnTXE 位 = 0，CBnRXE 位 = 1）：

从 CBnTX 寄存器中读取

13.3 寄存器

以下寄存器被用于控制 CSIBn。

- CSIBn 控制寄存器 0 (CBnCTL0)
- CSIBn 控制寄存器 1 (CBnCTL1)
- CSIBn 控制寄存器 2 (CBnCTL2)
- CSIBn 状态寄存器 (CBnSTR)

(1) CSIBn 控制寄存器 0 (CBnCTL0)

CBnCTL0 是用于控制 CSIBn 串行通信操作的寄存器。

该寄存器可以以 8 位或 1 位为单位进行读取和写入。

复位后该寄存器被设置为 01H。

(1/3)

复位后：01H R/W 地址：CB0CTL0 FFFFFFFD00H, CB1CTL0 FFFFFFFD10H

	7	6	5	4	3	2	1	0
CBnCTL0	CBnPWR	CBnTXE ^注	CBnRXE ^注	CBnDIR ^注	0	0	CBnTMS ^注	CBnSCE

(n = 0, 1)

CBnPWR	禁止/允许CSIBn操作的指定
0	禁止CSIBn操作并复位CBnSTR寄存器
1	允许CSIBn操作
• CBnPWR位控制CSIBn操作并复位内部电路。	

CBnTXE ^注	禁止/允许发送操作的指定
0	禁止发送操作
1	允许发送操作
• CBnTXE位为0时SOBn输出为低电平。	

CBnRXE ^注	禁止/允许接收操作的指定
0	禁止接收操作
1	允许接收操作
• 当CBnRXE位清零时，即使已经为了禁止接收操作而发送了预分频数据，也不会输出接收完成中断，且接收数据（CBnRX数据）也不会被更新。	

注 这些位只能在 CBnPWR = 0 时复写。但在复写这些位的同时可以设置 CBnPWR = 1。

注意事项 为了强制延缓发送/接收，应将 CBnPWR 位清零，而不是将 CBnRxE 位清零。
这时，时钟输出停止。

CBnDIR ^注	发送方向模式指定 (MSB/LSB)
0	MSB 优先发送
1	LSB 优先发送

CBnTMS ^注	发送模式指定
0	单独发送模式
1	连续发送模式

[单独发送模式中]

生成接收完成中断请求信号 (INTCBnR)。

即使发送使能 (CBnTXE 位 = 1)，不产生发送使能中断 (INTCBnT)。

如果在 期间 (CBnSTR.CBnTSF 位 = 1) 写入下一个发送数据，该数据被忽略，且不开 始下一次通信。同样，如果设置只能接收模式 (CBnTXE 位 = 0, CBnRXE 位 = 1)，即 使接收数据在通信期间 (CBnSTR.CBnTSF 位 = 1) 被读取，也不开始下一次通信。

[连续发送模式中]

在通信期间 (CBnSTR.CBnTSF 位 = 1) 写入下一个发送数据，即可使能连续发送模式。产生发送使能中断 (INTCBnT) 后，才可写入下一个发送数据。

如果在连续发送模式下设置了只能接收模式 (CBnTXE 位 = 0, CBnRXE 位 = 1)，无论是否在产生接收完成中断 (INTCBnR) 后读取寄存器 CBnRX，下一次接收都将立即开始。

因此，要立刻读取寄存器 CBnRX 中接收到的数据。如果延迟执行此操作，将产生溢出错误 (CBnOVE 位 = 1)。

注 这些位只能在 CBnPWR 位 = 0 时复写。然而，在重写这些位的同时也可以设置 CBnPWR 位 = 1。

CBnSCE	禁止/允许开始发送的指定
0	通信开始触发无效
1	通信开始触发有效

• 主模式中
 该位用于禁止或允许通信开始触发。
 (a) 单独发送或发送/接收模式，或连续发送或连续发送/接收模式
 设置CBnSCE位不影响通信操作。
 (b) 单独接收模式
 读取最后接收到的数据前，清零CBnSCE位。因为读取接收数据（寄存器CBnRX），接收已经开始注1。
 (c) 连续接收模式
 在接收到最后一个字节的数据前，清零CBnSCE位。从而禁止接收开始注2。
 • 从模式中
 该位用于使能或禁止通信开始触发。
 设置CBnSCE 位为1。

[CBnSCE位的使用]

• 单独接收模式
 <1> 当INTCBnR中断服务子程序接收到最后一字节数据，在读取寄存器CBnRX前清除CBnSCE 位。
 <2> 确定 CBnSTR.CBnTSF 位= 0后，清零 CBnRXE位以禁止接收。
 要连续接收，设置CBnSCE 位为1，以开始下一次接收，通过虚拟读取寄存器CBnRX。
 • 连续接收模式
 <1> 接收最后一个数据时在INTCBnR中断服务子程序里清除CBnSCE位。
 <2> 读取寄存器CBnRX。
 <3> 响应CBnTIR中断后，从寄存器CBnRX读取最后接收到的数据。
 <4> 确定 CBnSTR.CBnTSF 位= 0后，清零 CBnRXE位以禁止接收。
 要连续接收，设置CBnSCE 位为1，以等待下一次接收，通过虚拟读取寄存器CBnRX。

- 注
1. 如果 CBnSCE 位为 1 时被读取，则下一次通信操作开始。
 2. 如果 CBnSCE 位在最后一个数据接收完成前的一个通信时钟没有被清零，下一次通信操作自动开始。

注意事项 确保将位 2 和 3 位清 0。

(2) CSIBn 控制寄存器 1 (CBnCTL1)

该寄存器是 8 位寄存器，用于控制 CSIBn 串行通信操作。

该寄存器可以以 8 位或 1 位为单位进行读取或写入。

复位后将该寄存器设置为 00H。

<R>

注意事项 只有当 **CBnCTL0.CBnPWR = 0** 或 **CBnCTL0.CBnTXE** 以及 **CBnRXE** 位 = 0 时，寄存器 **CBnCTL1** 可被复写。

复位后: 00H R/W 地址: CB0CTL1 FFFFFFFD01H, CB1CTL1 FFFFFFFD11H

	7	6	5	4	3	2	1	0
CBnCTL1	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

(n = 0, 1)

	CBnCKP	CBnDAP	与 SCKBn 相关的数据 发送/接收时序的设置
通信类型 1	0	0	
通信类型 2	0	1	
通信类型 3	1	0	
通信类型 3	1	1	

CBnCKS2	CBnCKS1	CBnCKS0	通信时钟 (f _{CCLK}) ^{注 1}		模式
			n = 0	n = 1	
0	0	0	f _{xx} /2		主模式
0	0	1	f _{xx} /4		主模式
0	1	0	f _{xx} /8		主模式
0	1	1	f _{xx} /16		主模式
1	0	0	f _{xx} /32		主模式
1	0	1	f _{xx} /64		主模式
1	1	0	f _{BRG} ^{注 2}	TMP0 (TOP01)	主模式
1	1	1	外部时钟 (SCKBn)		从模式

<R>

- 注**
1. 将通信时钟 (f_{CCLK}) 设置到 8 MHz 或更低。
 2. 详细信息请参见 13.7 波特率生成器。

(3) CSIBn 控制寄存器 2 (CBnCTL2)

CBnCTL2 是 8 位寄存器，用于控制 CSIBn 串行通信的字节长度。

该寄存器可以以 8 位为单位进行读取或写入。

复位后将该寄存器设置为 00H。

注意事项 只有当 CBnCTL0.CBnPWR = 0 或者 CBnTXE 和 CBnRXE 位都为 0 时，寄存器 CBnCTL2 可被改写。

复位后：00H R/W 地址：CB0CTL2 FFFFFFFD02H, CB1CTL2 FFFFFFFD12H

	7	6	5	4	3	2	1	0
CBnCTL2	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

(n = 0, 1)

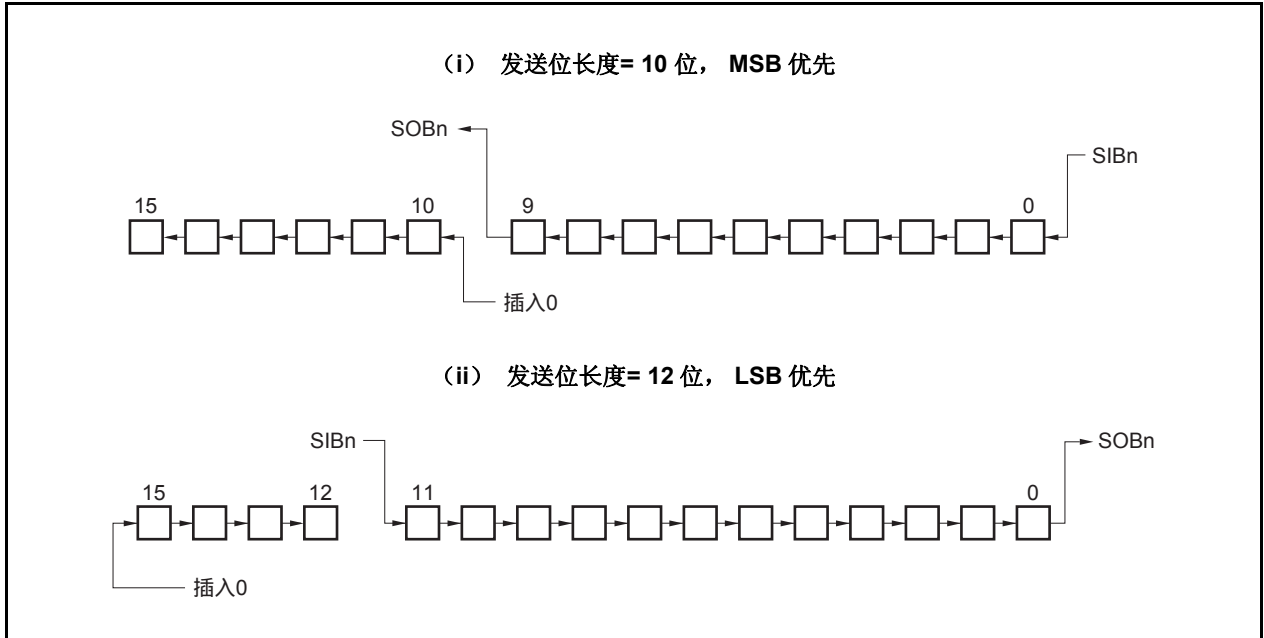
CBnCL3	CBnCL2	CBnCL1	CBnCL0	串行寄存器位长度
0	0	0	0	8位
0	0	0	1	9位
0	0	1	0	10位
0	0	1	1	11位
0	1	0	0	12位
0	1	0	1	13位
0	1	1	0	14位
0	1	1	1	15位
1	×	×	×	16位

- 备注**
1. 如果发送位数不是 8 或 16，则准备和使用来源于寄存器 CBnTX 和 CBnRX 中 LSB 的数据。
 2. ×： 不必考虑

(a) 发送数据长度改变功能

使用 CBnCTL2.CBnCL3 到 CBnCTL2.CBnCL0 位，在 8 位到 16 位间以 1 位为单位设置 CSIBn 发送数据长度。

无论发送起始位是 MSB 还是 LSB，只要当发送位长度设置为非 16 位，则从 LSB 开始将数据设置到寄存器 CBnTX 或 CBnRX。任何数据都可放置到那些不用的高位，但是接收的数据的高位在串行发送后都变为 0。



(4) CSIBn 状态寄存器 (CBnSTR)

该寄存器是 8 位寄存器，用于显示 CSIBn 的状态。
该寄存器可以以 8 位或 1 位为单位进行读取或写入，但 CBnTSF 标志是只读的。
复位后将该寄存器设置为 00H。
除了复位输入，通过清除（置 0）CBnCTL0.CBnPWR 位，该寄存器可被初始化。

复位后：00H R/W 地址：CB0STR FFFFD03H, CB1STR FFFFD13H

	7	6	5	4	3	2	1	0
CBnSTR	CBnTSF	0	0	0	0	0	0	CBnOVE

(n = 0, 1)

CBnTSF	通信状态标志
0	通信停止
1	通信中

• 通信过程中，当数据存入寄存器CBnTX中，该寄存器被设置；接收过程中，当虚拟读取寄存器时，该寄存器被设置。
当发送结束，该标志在最后一个时钟边沿被清零。

CBnOVE	溢出错误标志
0	没有溢出
1	溢出

• 当接收操作完成后 CPU没有读取接收缓存的值就结束下一次接收时，产生溢出错误。CBnOVE 标志显示产生溢出错误状态。
• 单独发送模式下CBnOVE 位依然有效。因此，只使用发送时注意以下方面。
• 不要检查CBnOVE标志。
• 即使不要求读取接收数据，也要读取该位。
• 标志通过写入0来清零。即使写入1也不能被设置。

13.4 中断请求信号

CSIBn 可以生成以下两种类型的中断请求信号。

- 接收完成中断请求信号 (INTCBnR)
- 发送允许中断请求信号 (INTCBnT)

这两个中断请求信号的默认优先级为：接收完成中断请求信号优先级高，发送允许中断请求信号优先级低。

表 13-2. 中断及其默认优先级

中断	优先级
接收完成	高
允许发送	低

(1) 接收完成中断请求信号 (INTCBnR)

在允许接收时，当接收数据被转移到 CBnRX 寄存器时，将会生成接收完成中断请求信号。

发生溢出错误时也可以生成该中断请求信号。

当接收到一个接收完成中断请求信号且数据被读取时，将会读取 CBnSTR 寄存器以检查接收结果是否是错误的。

在单独发送模式中，即使当只有发送操作在执行时，INTCBnR 中断请求信号也只会是在发送完成的基础上被生成。

(2) 发送允许中断请求信号 (INTCBnT)

在连续发送或连续发送/接收模式中，一旦允许对 CBnTX 进行写操作，发送数据将会立即被转移到 CBnTX 寄存器中且会生成一个发送允许中断请求信号。

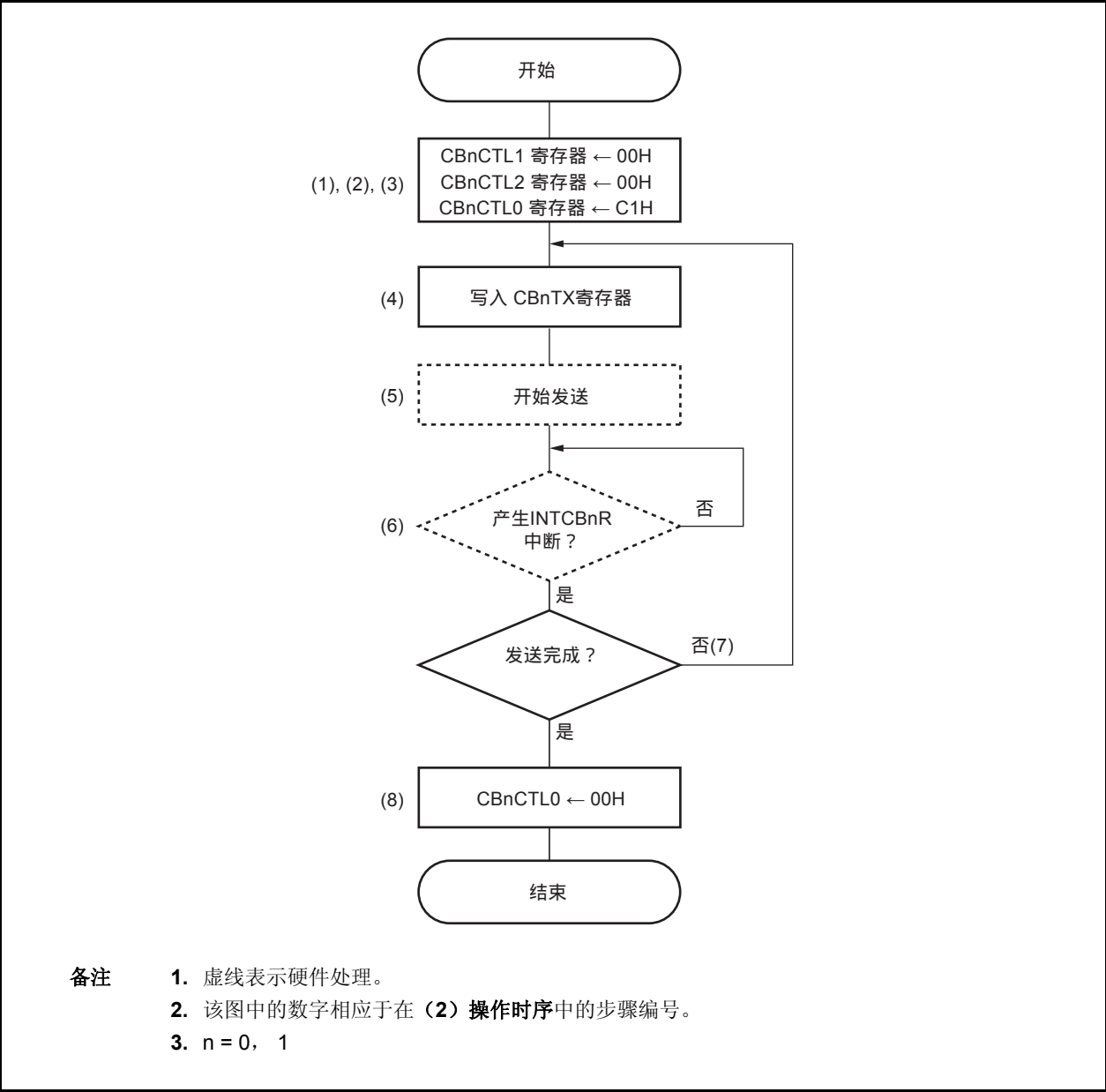
在单独发送模式以及单独发送/接收模式中，不会生成 INTCBnT 中断。

<R> 13.5 操作

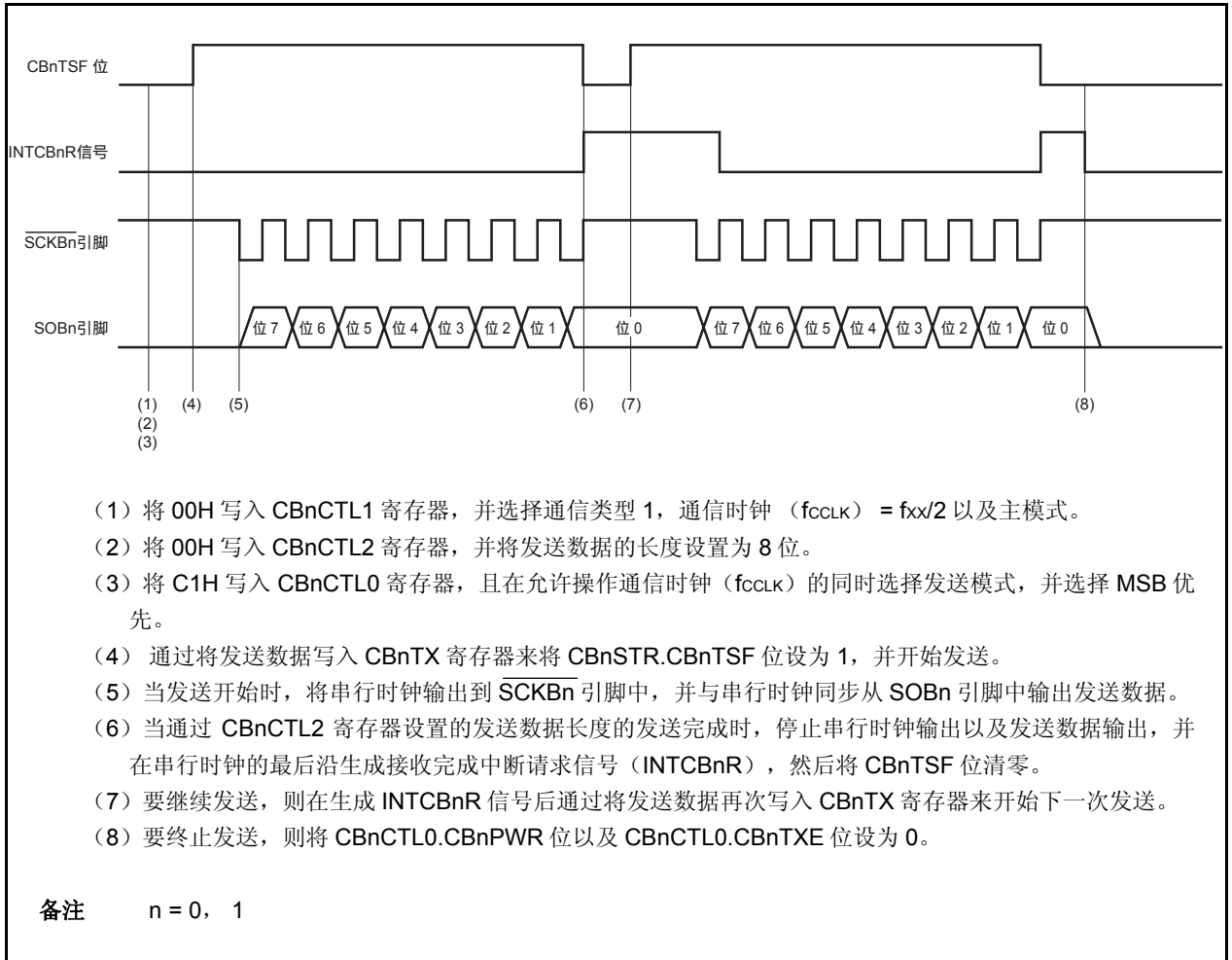
13.5.1 单独发送模式（主模式，发送模式）

MSB 优先（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CLK}）= f_{xx}/2（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000），发送数据的长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）

（1）操作流程



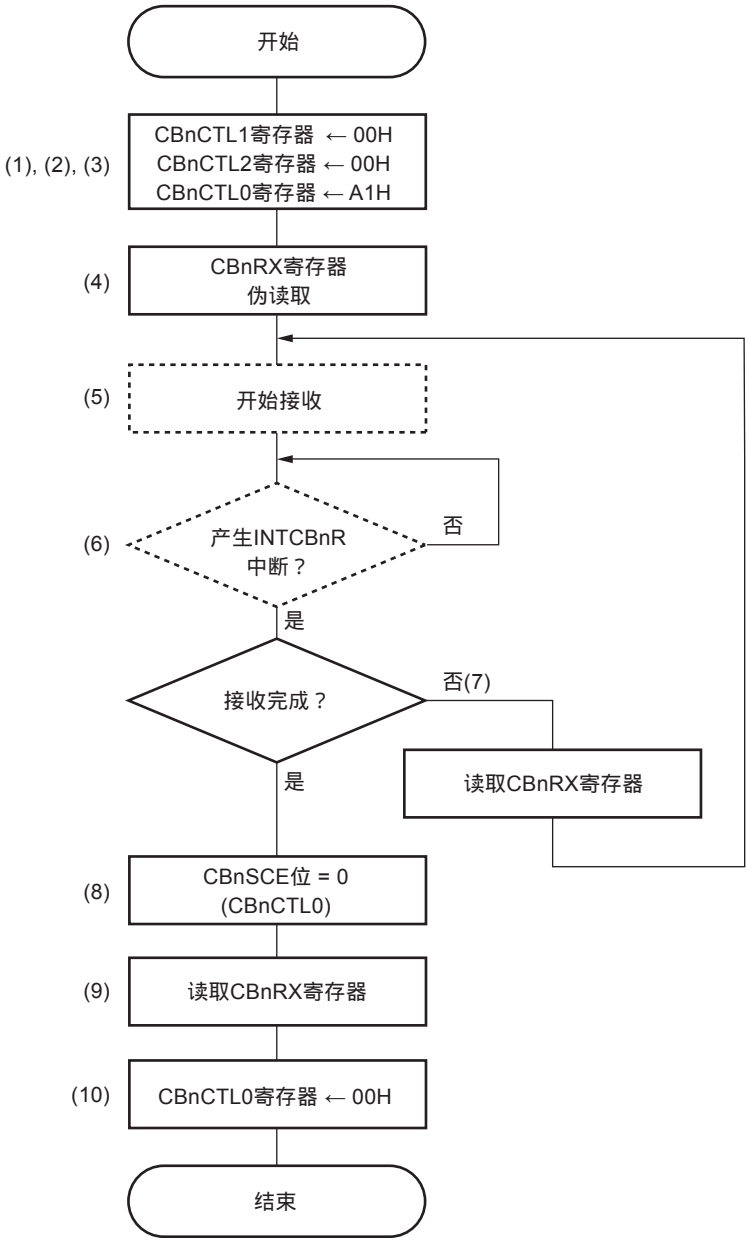
(2) 操作时序



13.5.2 单独发送模式（主模式，接收模式）

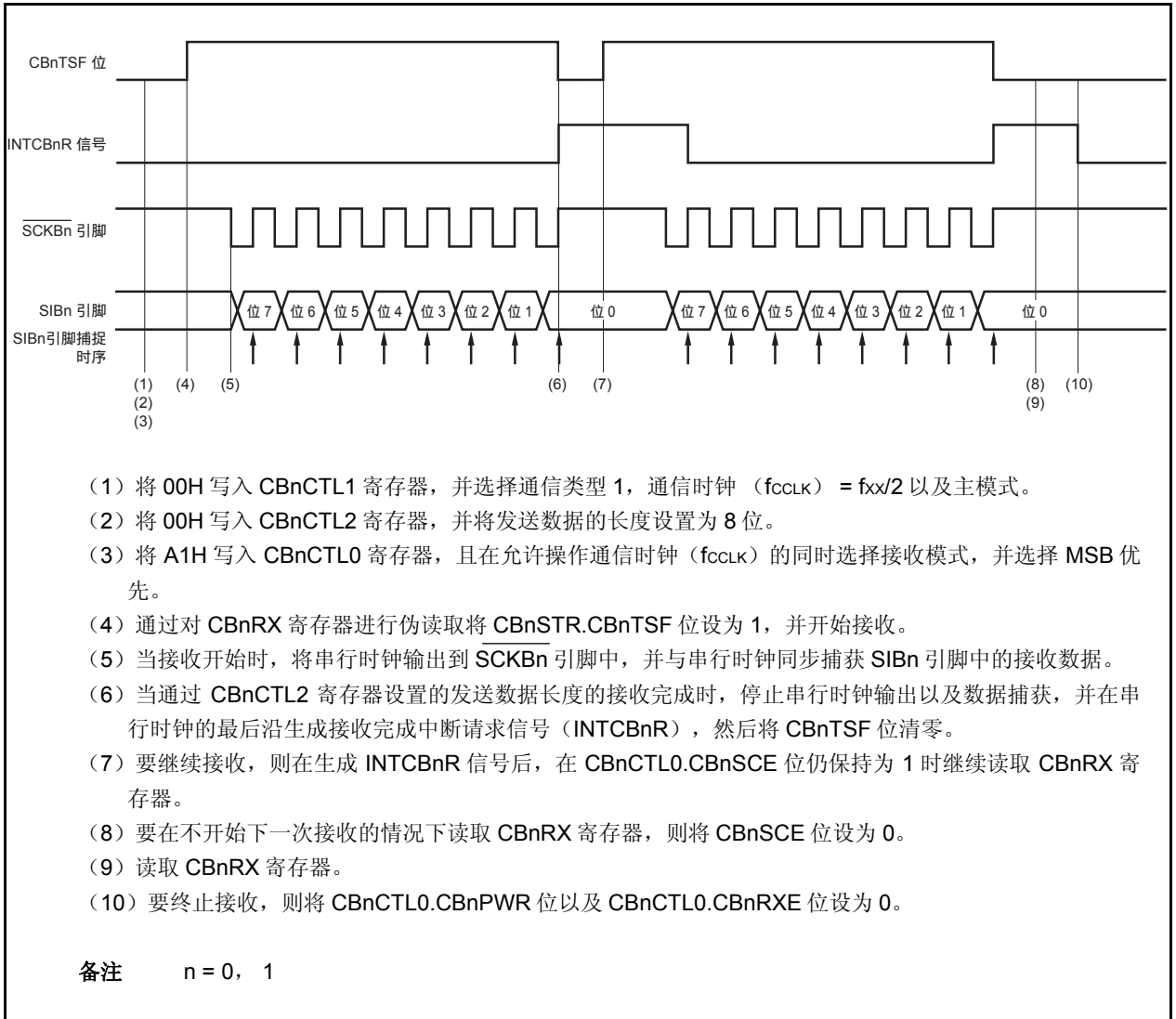
MSB 优先 (CBnCTL0.CBnDIR 位= 0)，通信类型 1 (CBnCTL1.CBnCKP 及 CBnCTL1.CBnDAP 位= 00)，通信时钟 (f_{CCLK}) = $f_{xx}/2$ (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位= 000)，发送数据长度= 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000)

(1) 操作流程



- 备注
- 1. 虚线表示硬件处理。
 - 2. 该图中的数字相应于在 (2) 操作时序中的步骤编号。
 - 3. $n = 0, 1$

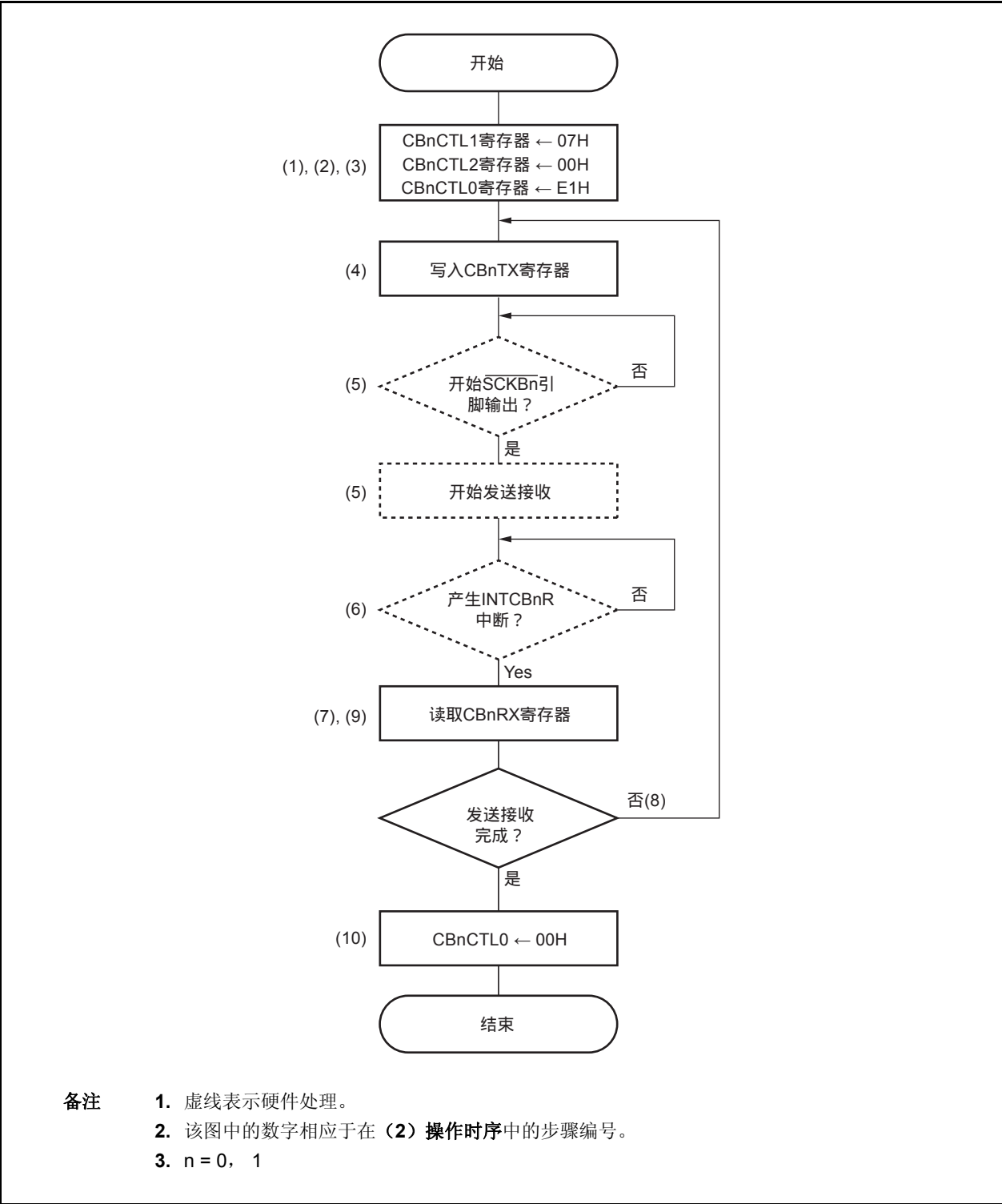
(2) 操作时序



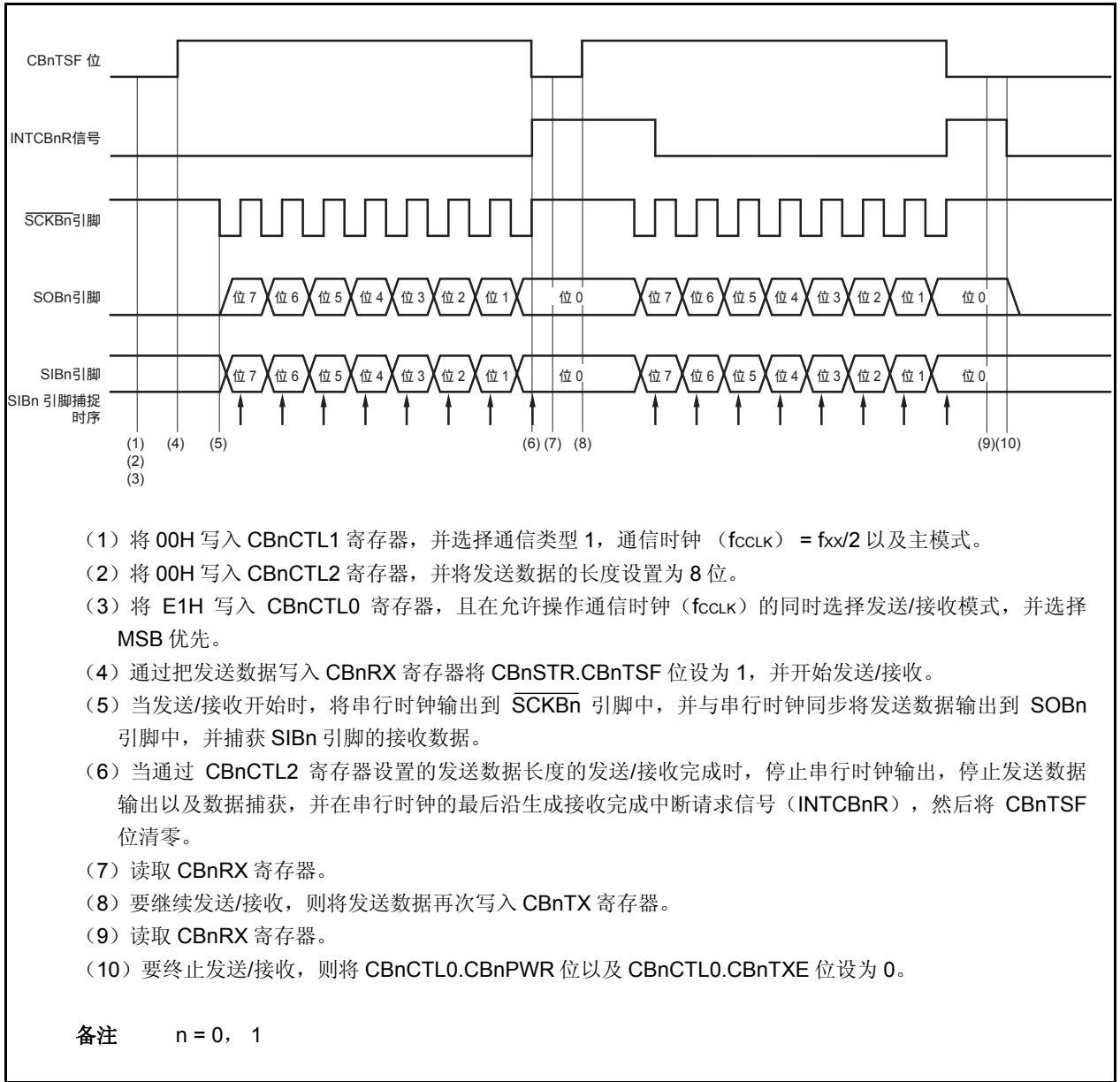
13.5.3 单独发送模式（主模式， 发送/接收模式）

MSB 优先（CBnCTL0.CBnDIR 位= 0）， 通信类型 1（CBnCTL1.CBnCKP 至 CBnCTL1.CBnDAP 位= 00），
通信时钟（f_{CCLK}）= f_{xx}/2（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位= 000）， 发送数据长度 = 8 位
（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000）

（1） 操作流程



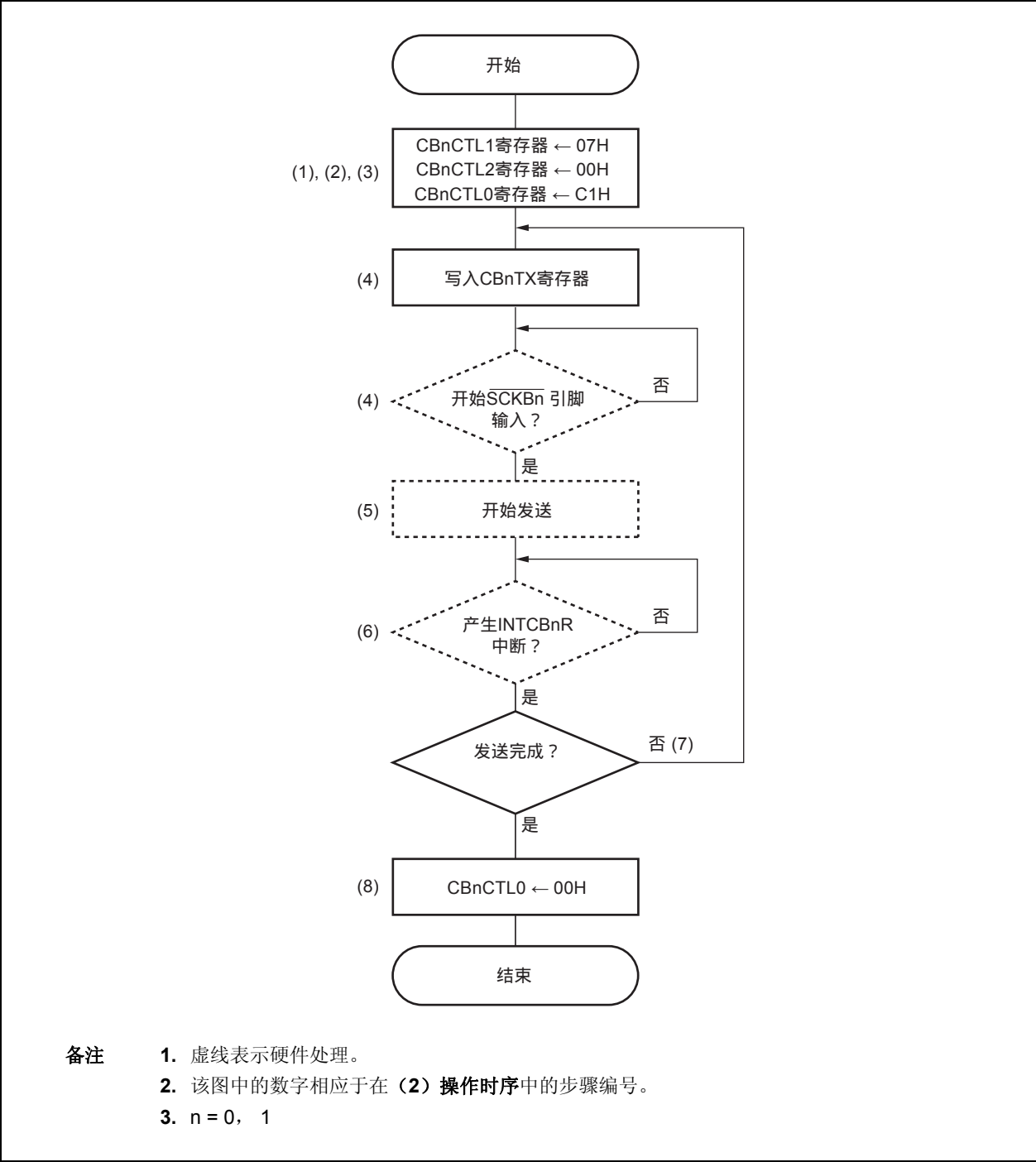
(2) 操作时序



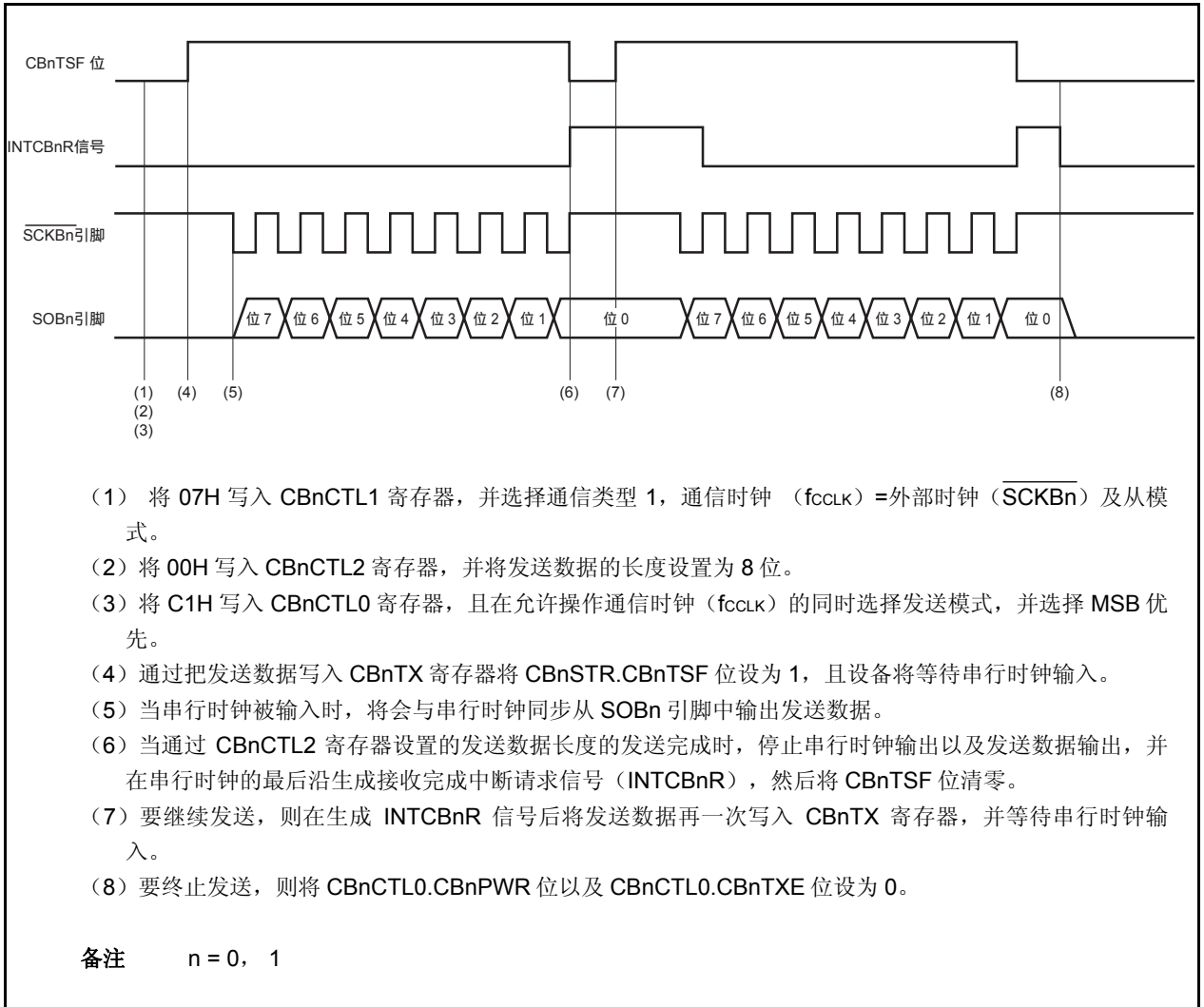
13.5.4 单独发送模式 (从模式, 发送模式)

MSB 优先 (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = 外部时钟 (SCKBn) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



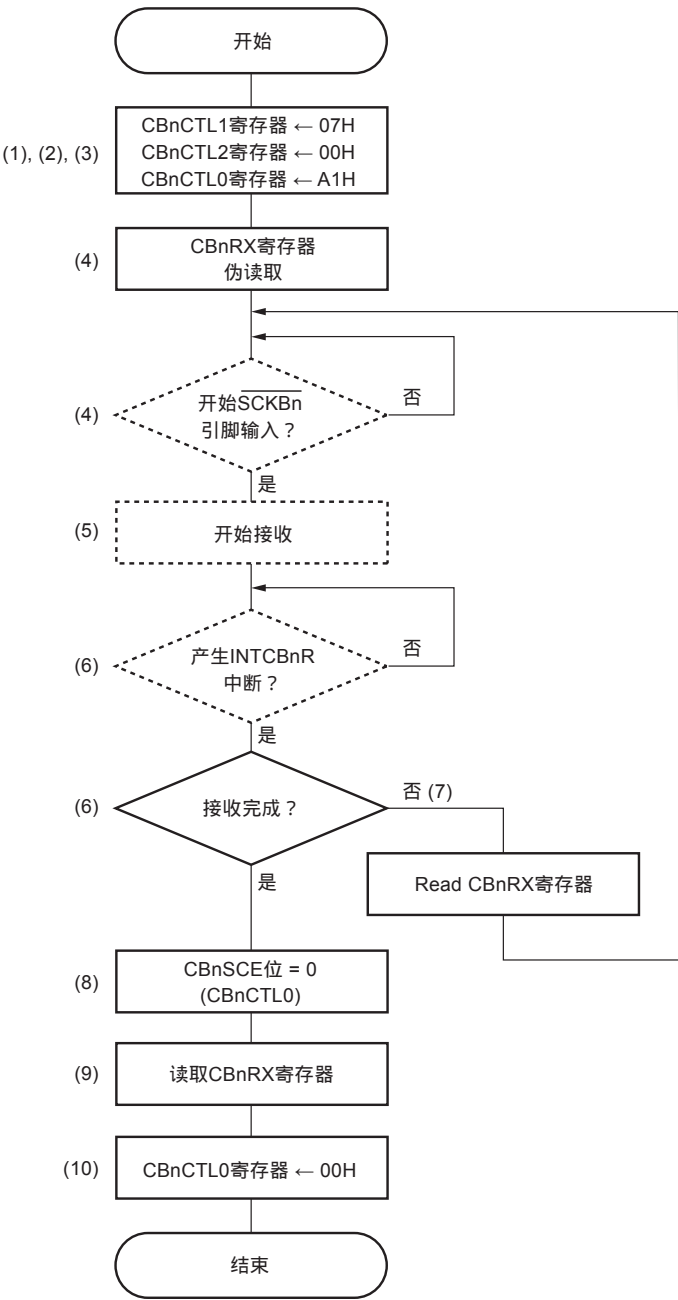
(2) 操作时序



13.5.5 单独发送模式 (从模式, 接收模式)

MSB 优先 (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = 外部时钟 ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

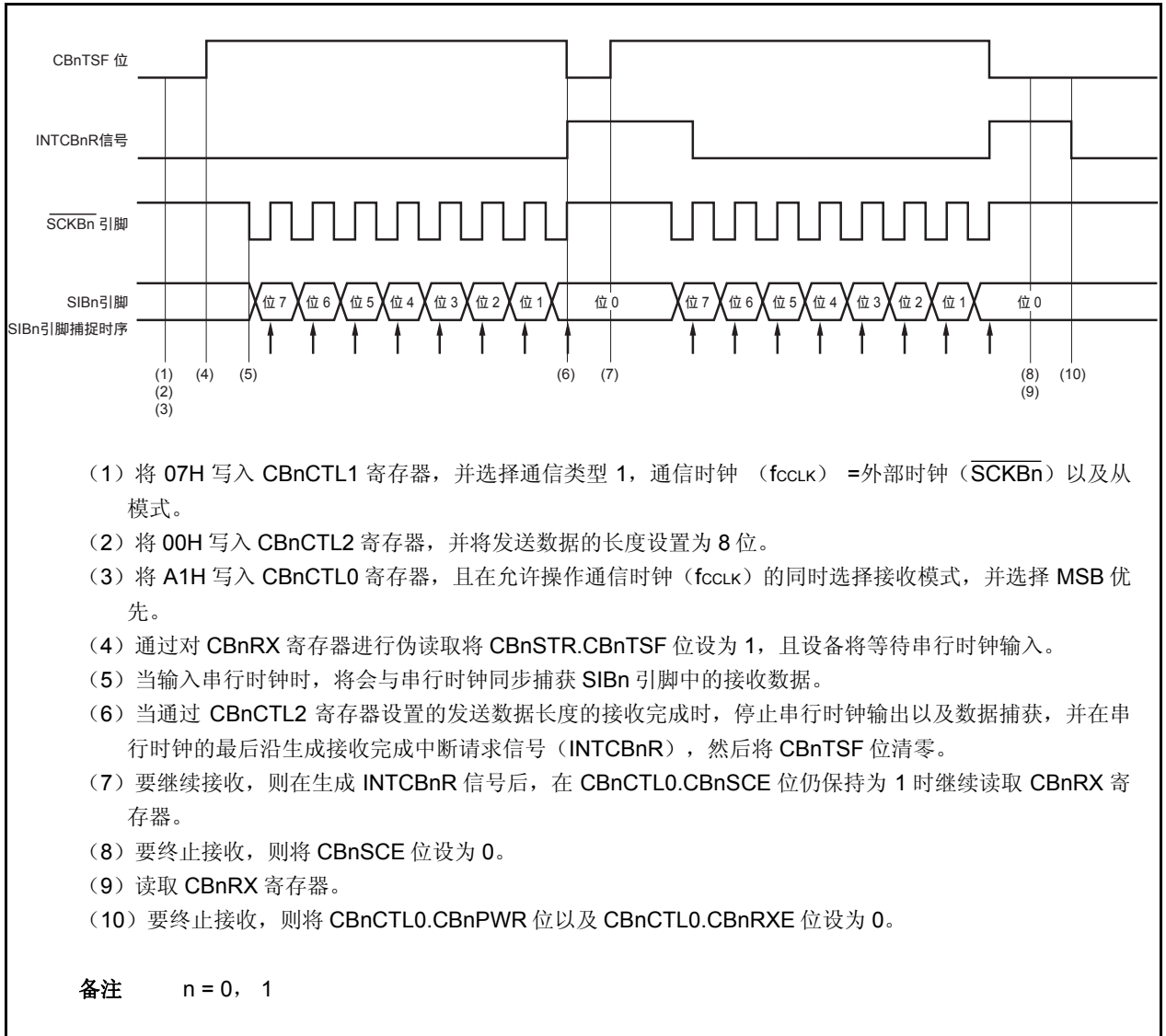
(1) 操作流程



备注

- 1. 虚线表示硬件处理。
- 2. 该图中的数字相应于在 (2) 操作时序中的步骤编号。
- 3. n = 0, 1

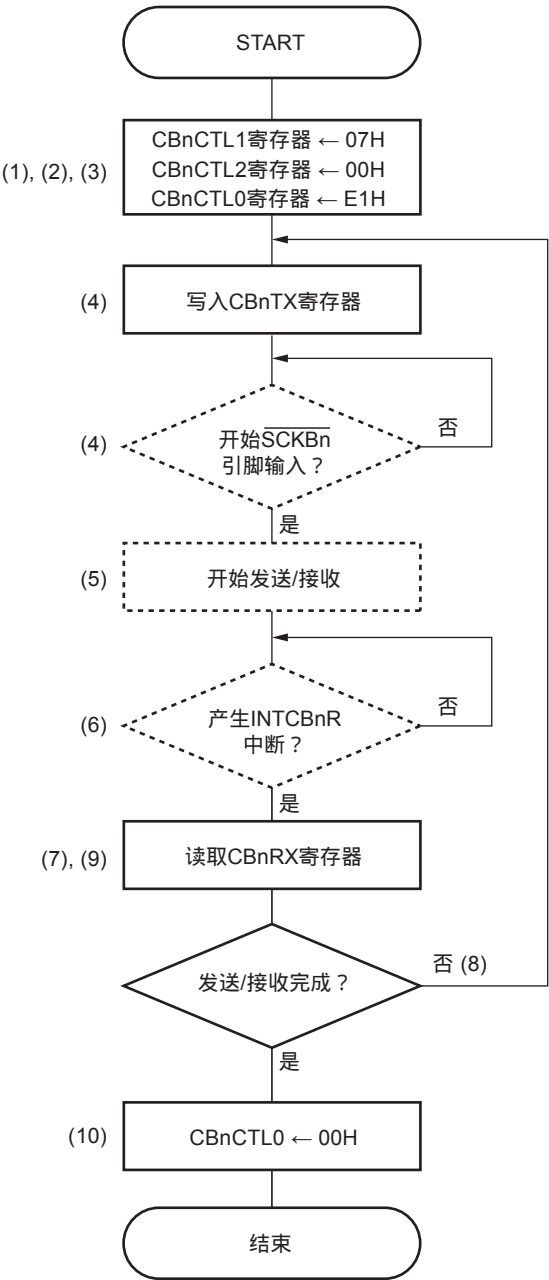
(2) 操作时序



13.5.6 单独发送模式 (从模式, 发送/接收模式)

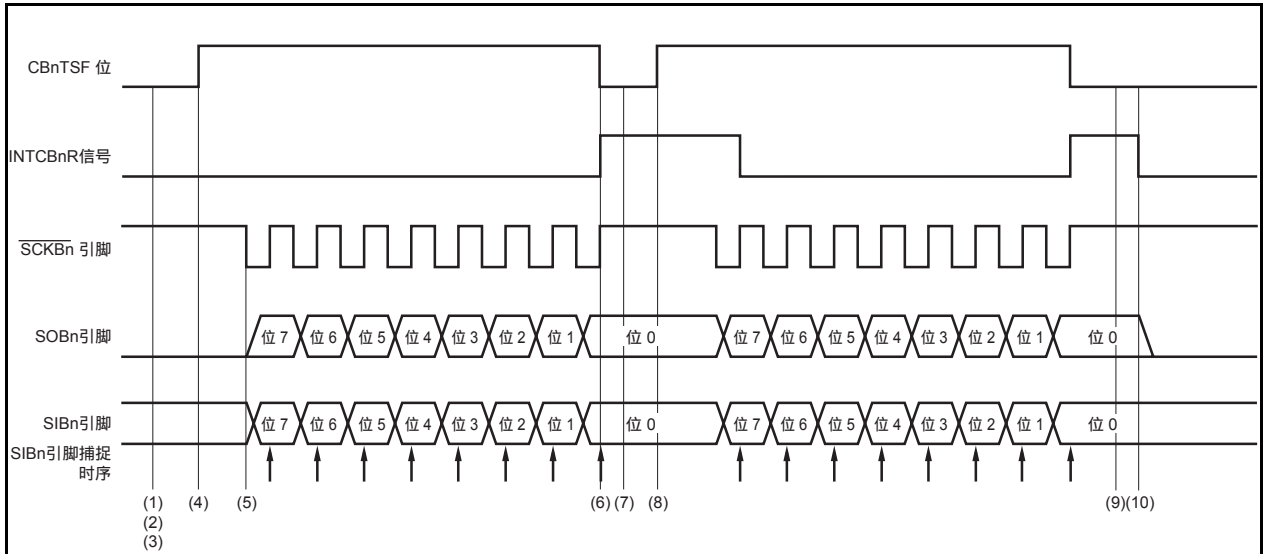
MSB 优先 (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = 外部时钟 ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程



- 备注
- 1. 虚线表示硬件处理。
 - 2. 该图中的数字相应于在 (2) 操作时序中的步骤编号。
 - 3. n = 0, 1

(2) 操作时序



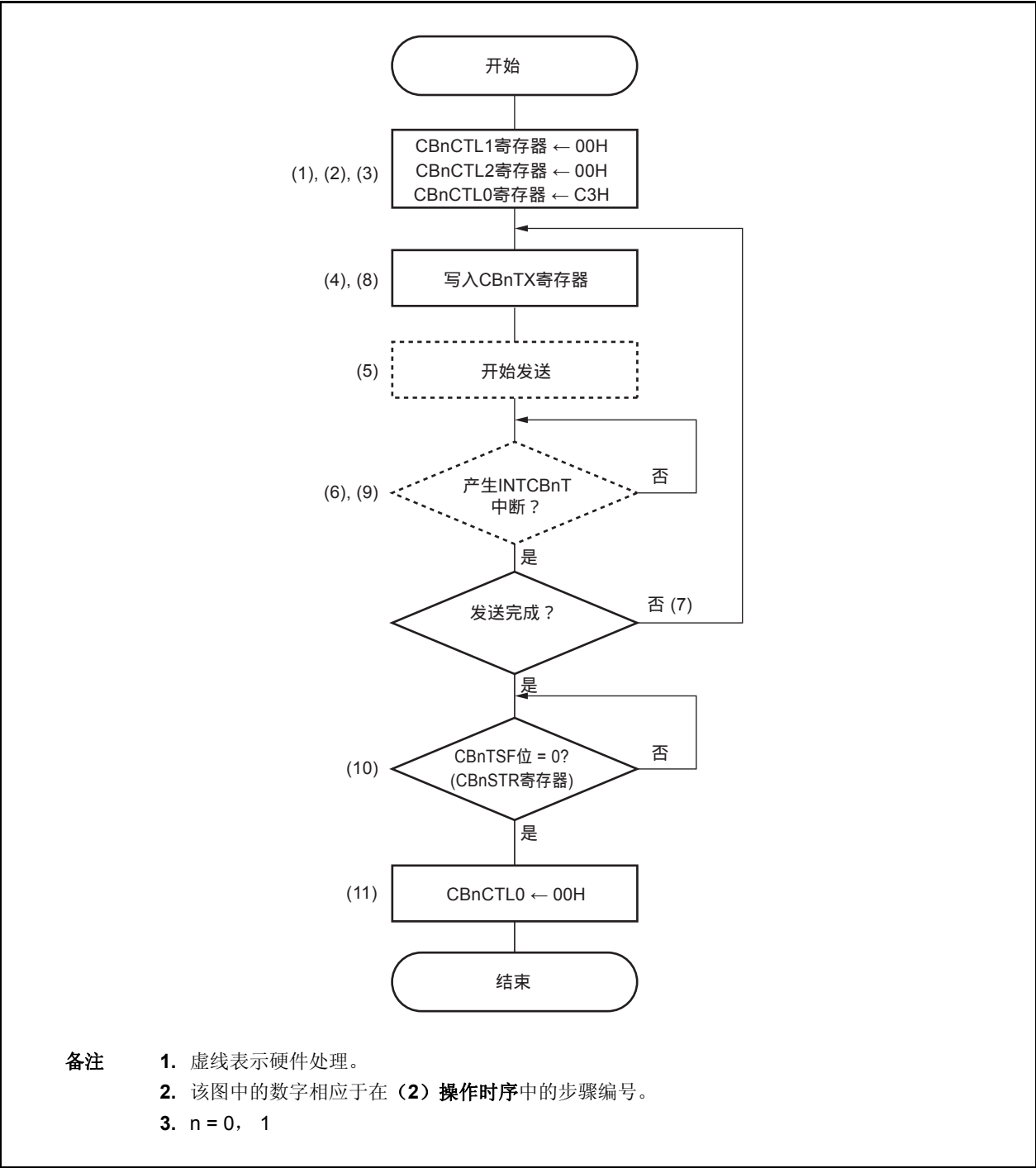
- (1) 将 07H 写入 CBnCTL1 寄存器，并选择通信类型 1，通信时钟 (f_{CCLK}) = 外部时钟 (SCKBn) 以及从模式。
- (2) 将 00H 写入 CBnCTL2 寄存器，并将发送数据的长度设置为 8 位。
- (3) 将 E1H 写入 CBnCTL0 寄存器，且在允许操作通信时钟 (f_{CCLK}) 的同时选择发送/接收模式，并选择 MSB 优先。
- (4) 通过把发送数据写入 CBnRX 寄存器将 CBnSTR.CBnTSF 位设为 1，且设备将会等待串行时钟输入。
- (5) 当输入串行时钟时，将会与串行时钟同步将发送数据输出到 SOBn 引脚中，并捕获 SIBn 引脚的接收数据。
- (6) 当通过 CBnCTL2 寄存器设置的发送数据长度的发送/接收完成时，停止串行时钟输出，停止发送数据输出以及数据捕获，并在串行时钟的最后沿生成接收完成中断请求信号 (INTCBnR)，然后将 CBnTSF 位清零。
- (7) 读取 CBnRX 寄存器。
- (8) 要继续发送/接收，则将发送数据再次写入 CBnTX 寄存器并等待串行时钟输入。
- (9) 读取 CBnRX 寄存器。
- (10) 要终止发送/接收，则将 CBnCTL0.CBnPWR 位以及 CBnCTL0.CBnTXE 位设为 0。

备注 n = 0, 1

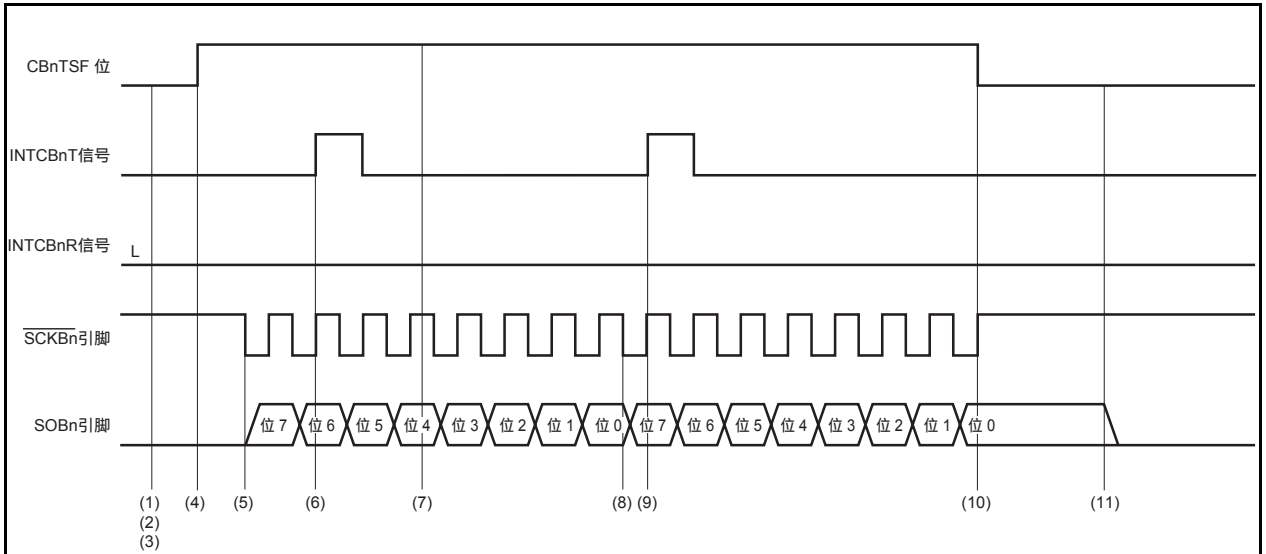
13.5.7 连续发送模式（主模式，发送模式）

MSB 优先（CBnCTL0.CBnDIR 位 = 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00），通信时钟（f_{CCLK}） = f_{xx}/2（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000），发送数据长度 = 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000）

（1）操作流程



(2) 操作时序



- (1) 将 00H 写入 CBnCTL1 寄存器，并选择通信类型 1，通信时钟 (f_{CCLK}) = $f_{xx}/2$ 以及主模式。
- (2) 将 00H 写入 CBnCTL2 寄存器，并将发送数据长度设置为 8 位。
- (3) 将 C3H 写入 CBnCTL0 寄存器，并在允许操作通信时钟 (f_{CCLK}) 的同时选择发送模式，MSN 优先以及连续发送模式。
- (4) 通过将发送数据写入 CBnTX 寄存器将 CBnSTR.CBnTSF 位设为 1，并开始发送。
- (5) 当发送开始时，将串行时钟输出到 SCKBn 引脚中，并与串行时钟同步从 SOBn 引脚中输出发送数据。
- (6) 当完成发送数据从 CBnTX 寄存器到移位寄存器的转移并且允许对 CBnTX 寄存器进行写操作时，将会生成发送允许中断请求信号 (INTCBnT)。
- (7) 要继续发送，则在生成 INTCBnT 信号后将发送数据再次写入 CBnTX 寄存器中。
- (8) 当一个新的发送数据在完成通信前被写入 CBnTX 寄存器，那么在通信完成后将会开始下一次通信。
- (9) 发送数据从 CBnTX 寄存器到移位寄存器的转移被完成并且生成 INTCBnT 信号。要通过当前发送来结果连续发送，则不要对 CBnTX 寄存器进行写操作。
- (10) 当下一个发送数据在发送完成前没有被写入 CBnTX 寄存器时，将会在发送完成后停止将串行时钟输出到 SCKBn 引脚中，并将 CBnTSF 位清零。
- (11) 要释放发送允许状态，则在核实 CBnTSF 位 = 0 后将 CBnCTL0.CBnPWR 位及 CBnCTL0.CBnTXE 位设为 0。

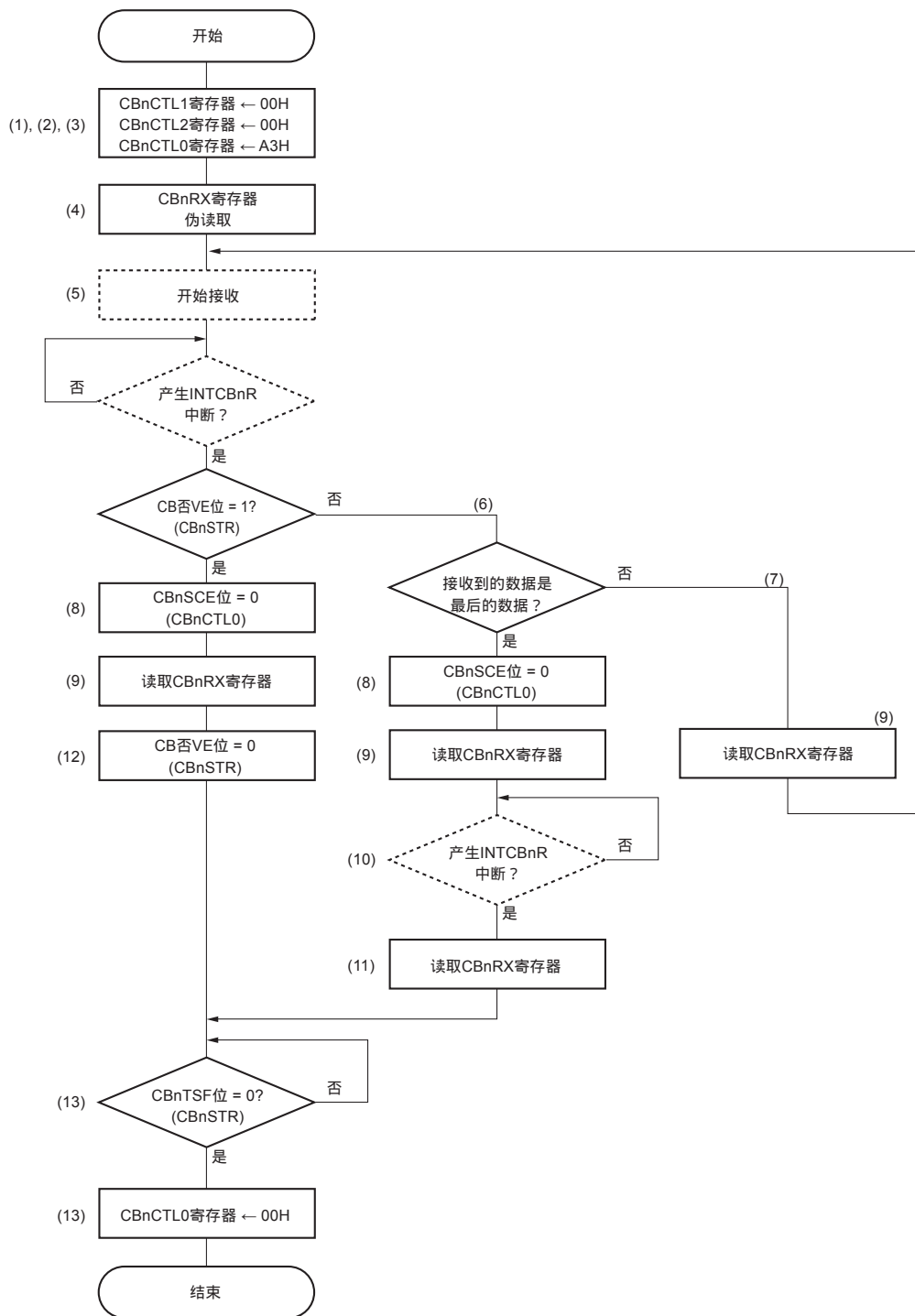
注意事项 在连续发送模式中，接收完成中断请求信号 (INTCBnR) 不会被生成。

备注 n = 0, 1

13.5.8 连续发送模式（主模式，接收模式）

MSB 优先 (CBnCTL0.CBnDIR 位 = 0)，通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通信时钟 (f_{CCLK}) = $f_{xx}/2$ (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000)，发送数据长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

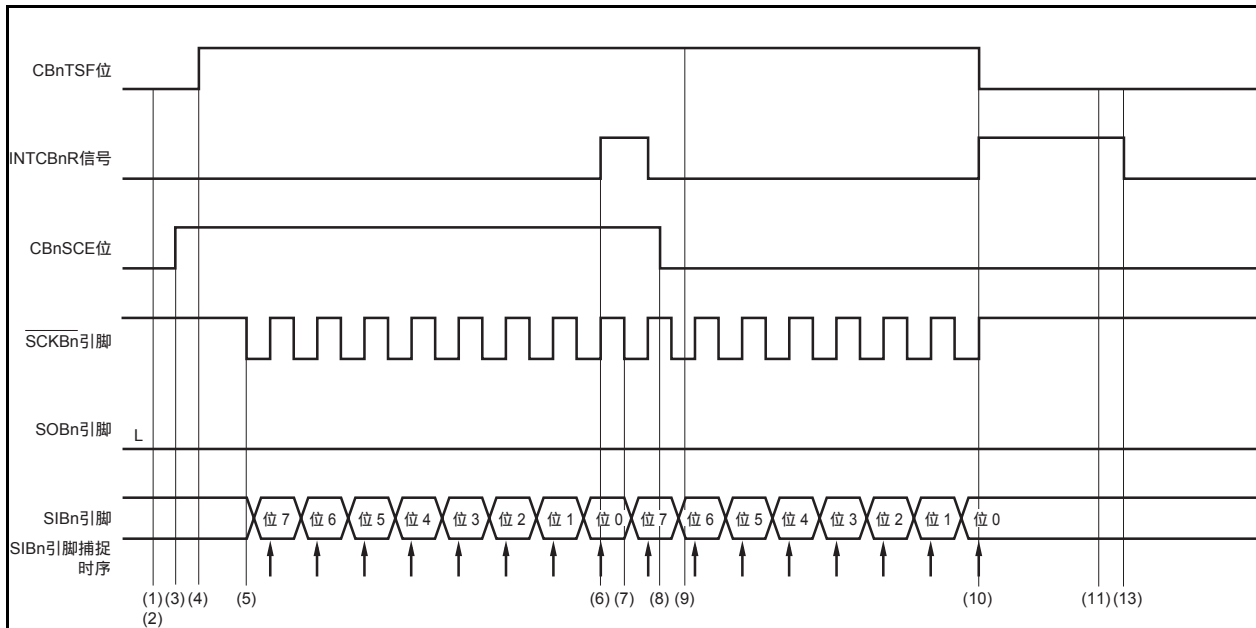
(1) 操作流程



备注

1. 虚线表示硬件处理。
2. 该图中的数字相应于在 (2) 操作时序中的步骤编号。
3. $n = 0, 1$

(2) 操作时序



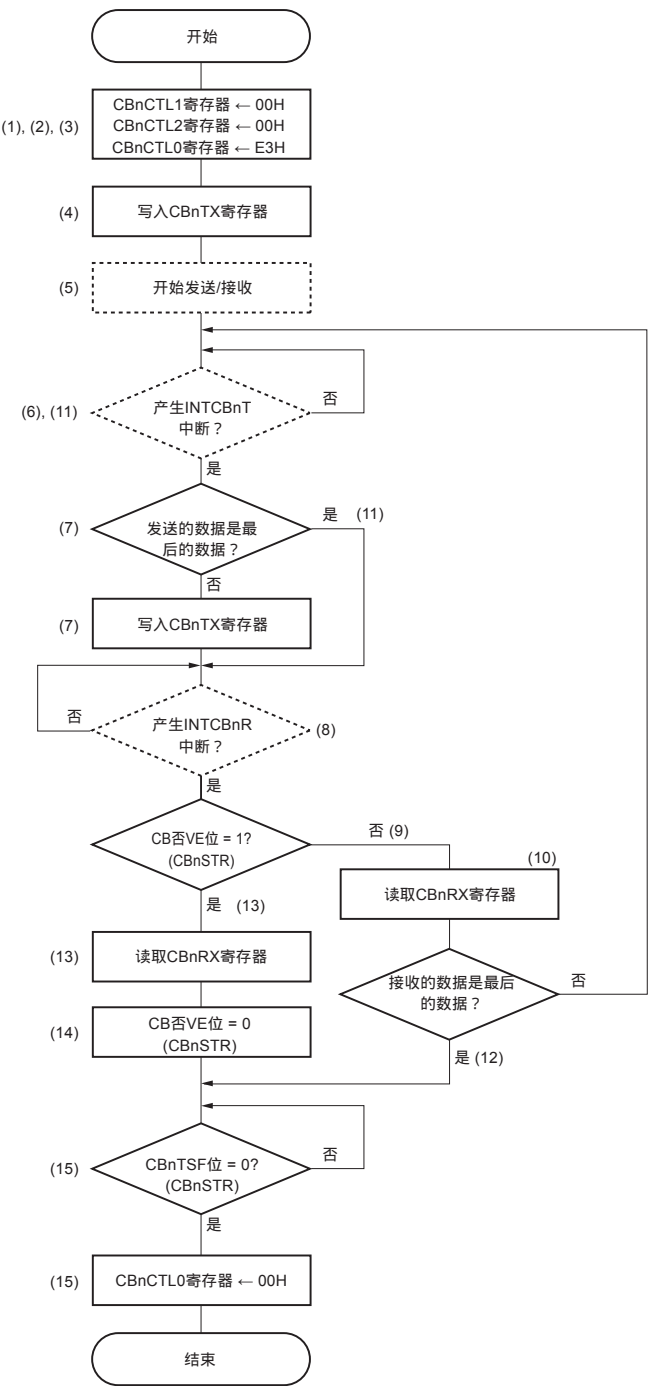
- (1) 将 00H 写入 CBnCTL1 寄存器，并选择通信类型 1，通信时钟 (f_{CCLK}) = $f_x/2$ 以及主模式。
- (2) 将 00H 写入 CBnCTL2 寄存器，并将发送数据长度设置为 8 位。
- (3) 将 A3H 写入 CBnCTL0 寄存器，并在允许操作通信时钟 (f_{CCLK}) 的同时选择接收模式，MSN 优先以及连续发送模式。
- (4) 通过执行对 CBnRX 寄存器的伪读取将 CBnSTR.CBnTSF 设为 1，并开始接收。
- (5) 当接收开始时，将串行时钟输出到 \overline{SCKBn} 引脚中，并与串行时钟同步获取 SIBn 引脚的接收数据。
- (6) 当接收完成时，将会生成接收完成中断请求信号 (INTCBnR)，并允许读取 CBnRX 寄存器。
- (7) 若通信完成时 CBnCTL0.CBnSCE 位=1，那么在通信完成后将会开始下一次通信。
- (8) 要通过当前接收来终止连续接收，则将 CBnSCE 位设为 0。
- (9) 读取 CBnRX 寄存器。
- (10) 当接收完成时，将生成 INTCBnR 信号并允许读取 CBnRX 寄存器。若在通信结束前将 CBnSCE 设为 0，则将停止将串行时钟输出到 \overline{SCKBn} 引脚中并且会将 CBnTSF 位清 0 以结束接收操作。
- (11) 读取 CBnRX 寄存器。
- (12) 如果发生溢出错误，则将会把 CBnSTR.CBnOVE 位设为 0 并清除错误标志。
- (13) 要释放接收允许状态，则在核实 CBnTSF 位= 0 后将 CBnCTL0.CBnPWR 位及 CBnCTL0.CBnRXE 位设为 0。

备注 n = 0, 1

13.5.9 连续发送模式（主模式，发送/接收模式）

MSB 优先（CBnCTL0.CBnDIR 位 = 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00），通信时钟（f_{CCLK}） = f_{xx}/2（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000），发送数据长度 = 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000）

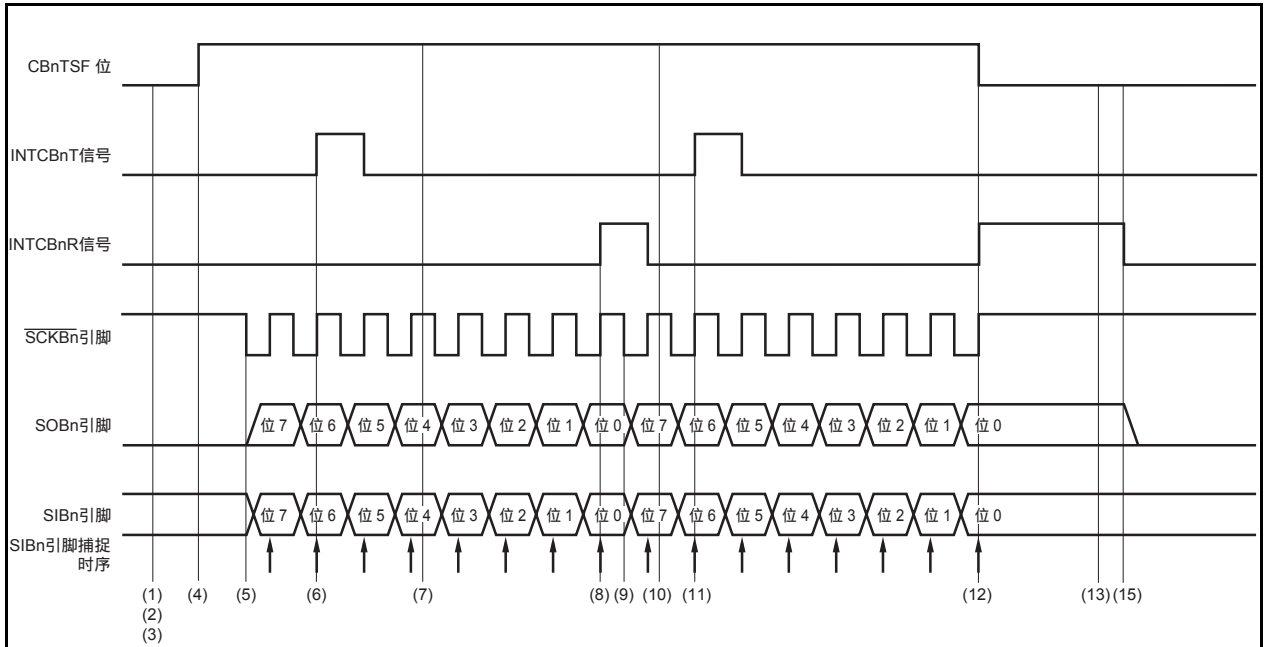
(1) 操作流程



- 备注
- 1. 虚线表示硬件处理。
 - 2. 该图中的数字相应于在 (2) 操作时序中的步骤编号。
 - 3. $n = 0, 1$

(2) 操作时序

(1/2)



- (1) 将 00H 写入 CBnCTL1 寄存器，并选择通信类型 1，通信时钟 (f_{CLK}) = $f_{\text{xx}}/2$ 以及 主模式。
- (2) 将 00H 写入 CBnCTL2 寄存器，并将发送数据长度设置为 8 位。
- (3) 将 E3H 写入 CBnCTL0 寄存器，并在允许操作通信时钟 (f_{CLK}) 的同时选择发送/接收模式，MSN 优先以及连续发送模式。
- (4) 通过将发送数据写入 CBnRX 寄存器将 CBnSTR.CBnTSF 设为 1，并开始发送/接收。
- (5) 当开始发送/接收时，将串行时钟输出到 SCKBn 引脚中，并与串行时钟同步将发送数据输出到 SOBn 引脚中，然后获取 SIBn 引脚的接收数据。
- (6) 当完成发送数据从 CBnTX 寄存器到移位寄存器的转移并且允许对 CBnTX 寄存器进行写操作时，将会生成发送允许中断请求信号 (INTCBnT)。
- (7) 要继续发送/接收，则在生成 INTCBnT 信号后将发送数据再次写入 CBnTX 寄存器。
- (8) 当完成一次发送/接收时，将会生成一个接收完成中断请求信号 (INTCBnR) 并会允许对 CBnTX 寄存器进行读取操作。
- (9) 当一个新的发送数据在通信完成前被写入 CBnTX 寄存器时，在通信完成后将会开始下一次通信。
- (10) 读取 CBnRX 寄存器。

备注 $n = 0, 1$

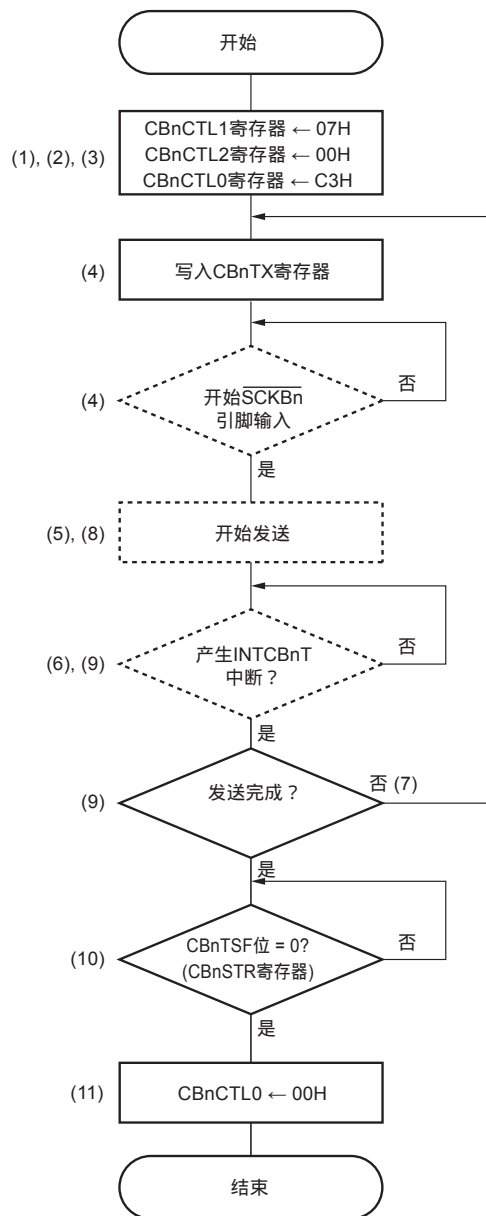
- (11) 发送数据从 CBnTX 寄存器到移位寄存器的转移被完成且生成 INTCBnT 信号。要通过当前发送/接收来结束连续发送/接收，则不要对 CBnTX 寄存器进行写操作。
- (12) 若下一个发送数据在发送完成前没有被写入 CBnTX 寄存器，则将在发送完成后停止将串行时钟输出到 SCKBn 引脚中，并将 CBnTSF 位清零。
- (13) 当接收错误中断请求信号 (INTCBnR) 被生成时，将会读取 CBnRX 寄存器。
- (14) 如果发生溢出错误，则将 CBnSTR.CBnOVE 位设为 0 并清除错误标志。
- (15) 要释放发送/接收允许状态，则在核实 CBnTSF 位 = 0 后将 CBnCTL0.CBnPWR 位及 CBnCTL0.CBnTXE 位设为 0。

备注 n = 0, 1

13.5.10 连续发送模式 (从模式, 发送模式)

MSB 优先 (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CLK}) = 外部时钟 ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

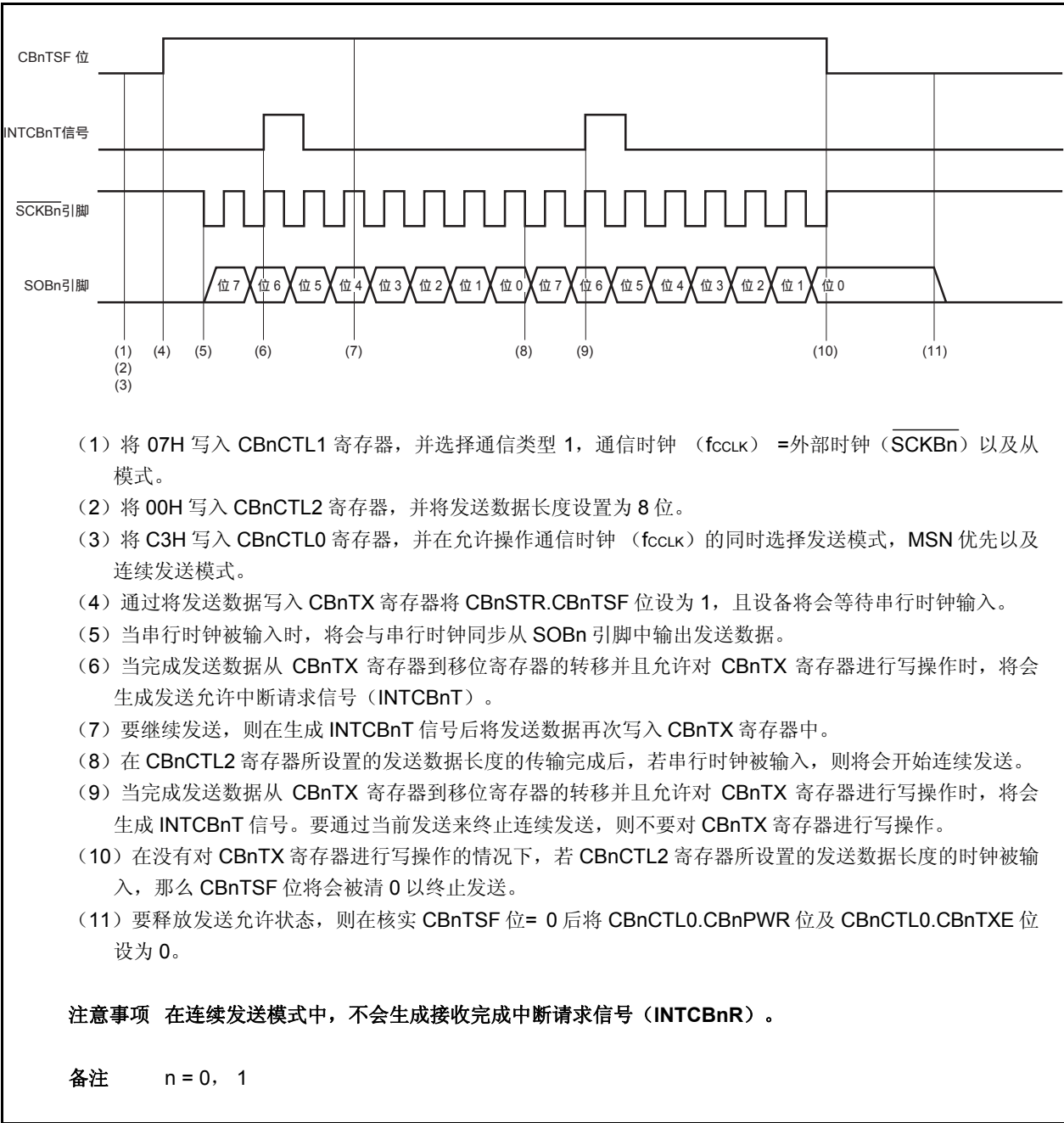
(1) 操作流程



备注

1. 虚线表示硬件处理。
2. 该图中的数字相应于在 (2) 操作时序中的步骤编号。
3. $n = 0, 1$

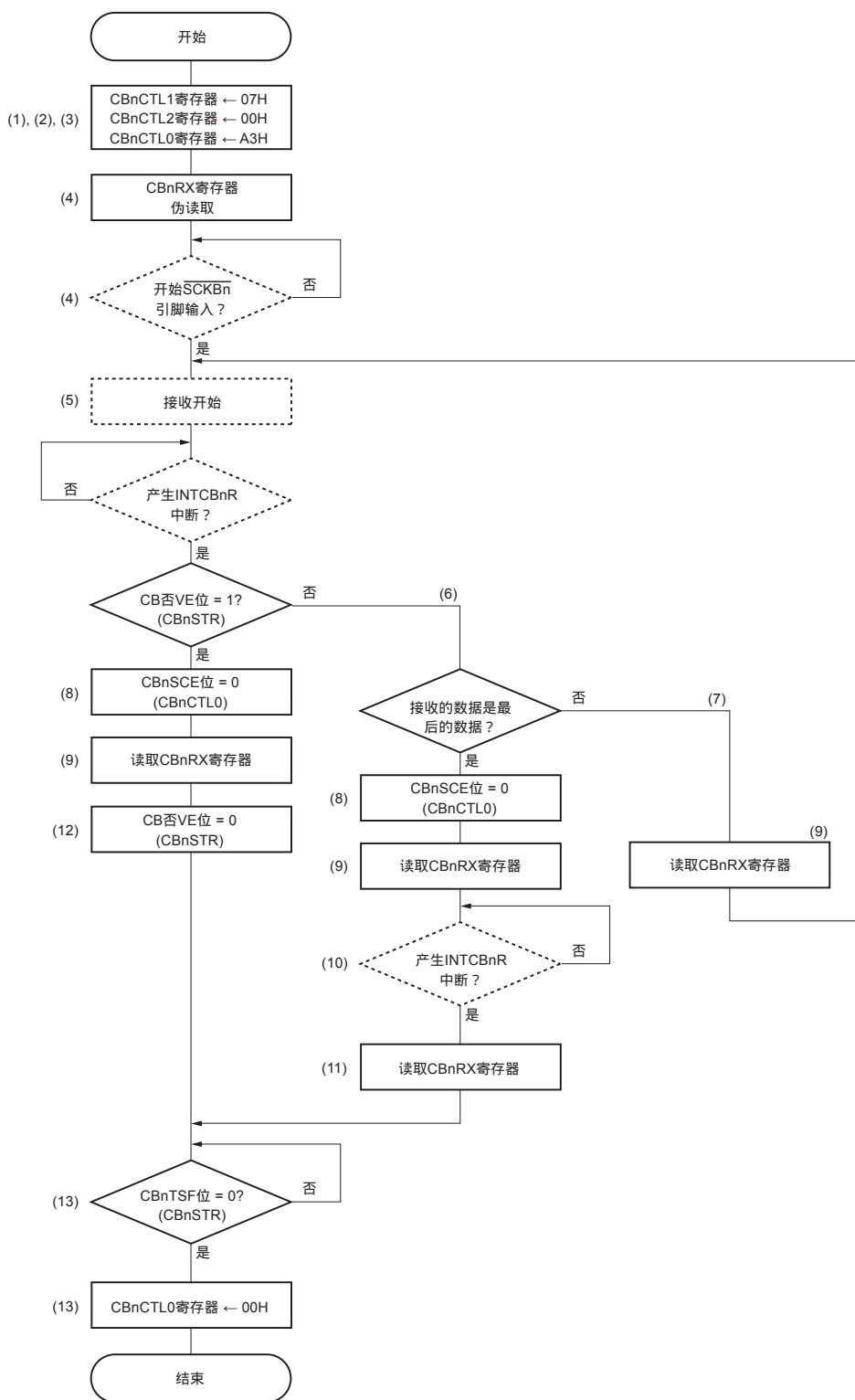
(2) 操作时序



13.5.11 连续发送模式（从模式，接收模式）

MSB 优先 (CBnCTL0.CBnDIR 位 = 0)，通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通信时钟 (f_{CCLK}) = 外部时钟 ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111)，发送数据长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

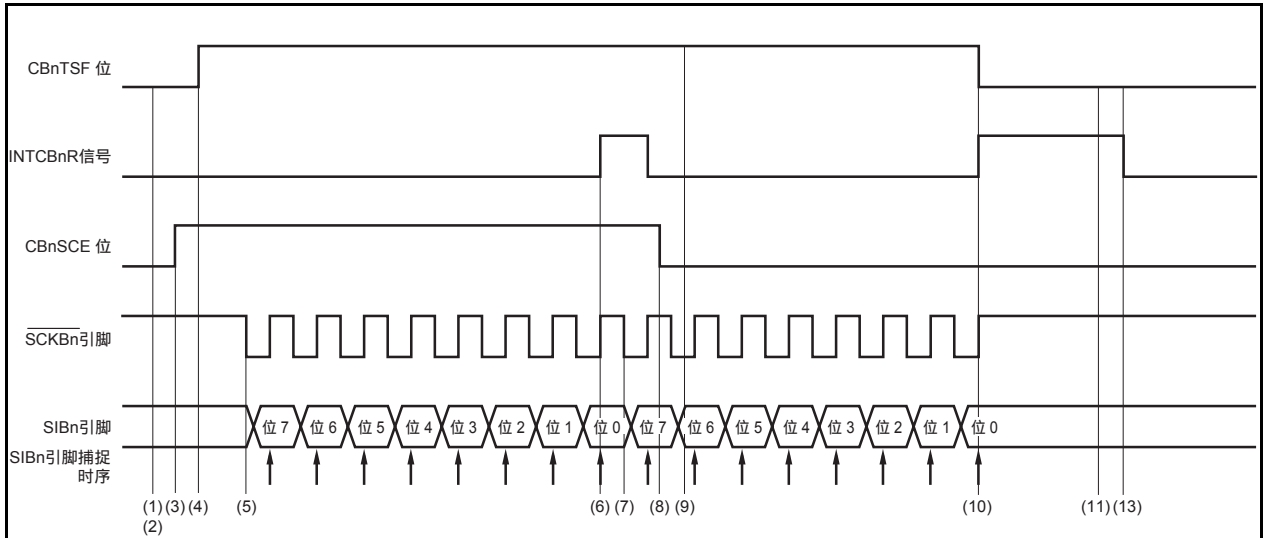
(1) 操作流程



备注

1. 虚线表示硬件处理。
2. 该图中的数字相应于在 (2) 操作时序中的步骤编号。
3. $n = 0, 1$

(2) 操作时序



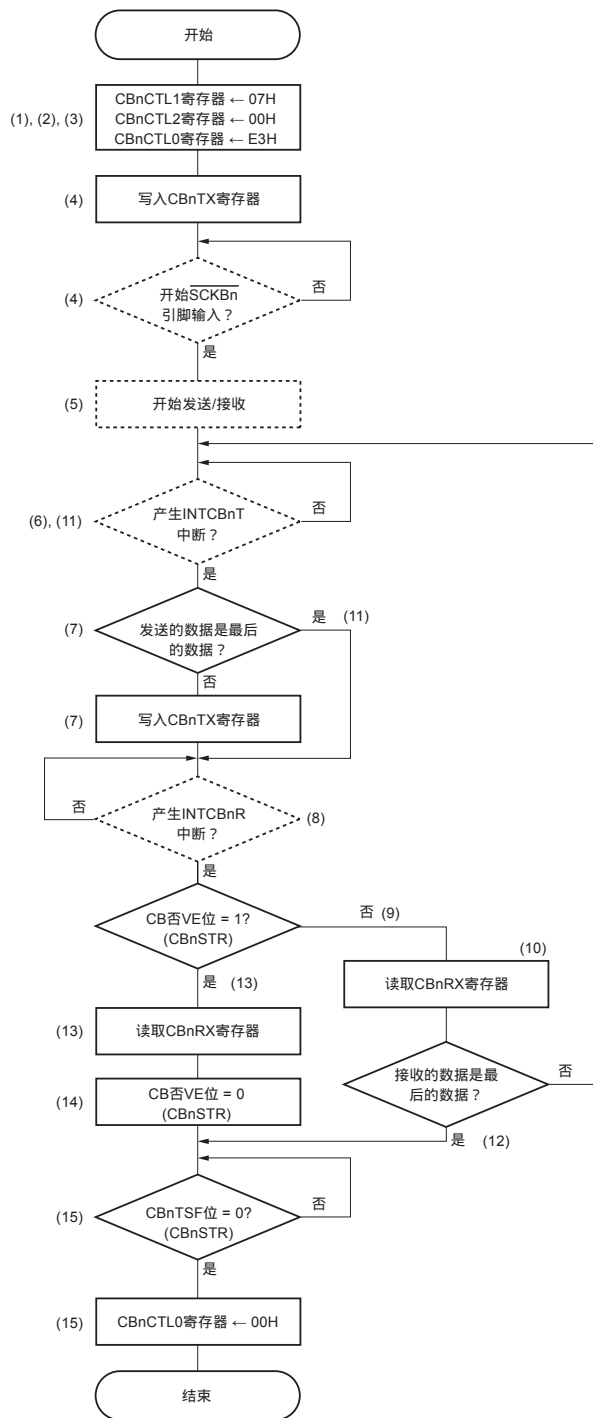
- (1) 将 07H 写入 CBnCTL1 寄存器，并选择通信类型 1，通信时钟 (f_{CLK}) = 外部时钟 (\overline{SCKBn}) 以及从模式。
- (2) 将 00H 写入 CBnCTL2 寄存器，并将发送数据长度设置为 8 位。
- (3) 将 A3H 写入 CBnCTL0 寄存器，并在允许操作通信时钟 (f_{CLK}) 的同时选择接收模式，MSN 优先以及连续发送模式。
- (4) 通过执行对 CBnRX 寄存器的伪读取将 CBnSTR.CBnTSF 设为 1，而设备将会等待串行时钟输入。
- (5) 当输入串行时钟时，将会与串行时钟同步捕获 SIBn 引脚的接收数据。
- (6) 当接收完成时，将会生成接收完成中断请求信号 (INTCBnR)，并允许读取 CBnRX 寄存器。
- (7) 若在 CBnCTL0.CBnSCE 位=1 时输入一个串行时钟，则将会开始连续接收。
- (8) 要通过当前接收来终止连续接收，则将 CBnSCE 位设为 0。
- (9) 读取 CBnRX 寄存器。
- (10) 当接收完成时，将生成 INTCBnR 信号并允许读取 CBnRX 寄存器。若在通信结束前将 CBnSCE 设为 0，那么 CBnTSF 位将会被清 0 以结束接收操作。
- (11) 读取 CBnRX 寄存器。
- (12) 如果发生溢出错误，则将会把 CBnSTR.CBnOVE 位设为 0 并清除错误标志。
- (13) 要释放接收允许状态，则在核实 CBnTSF 位= 0 后将 CBnCTL0.CBnPWR 位及 CBnCTL0.CBnRXE 位设为 0。

备注 $n = 0, 1$

13.5.12 连续发送模式（从模式，发送/接收模式）

MSB 优先 (CBnCTL0.CBnDIR 位 = 0)，通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通信时钟 (f_{CCLK}) = 外部时钟 ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111)，发送数据长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

(1) 操作流程

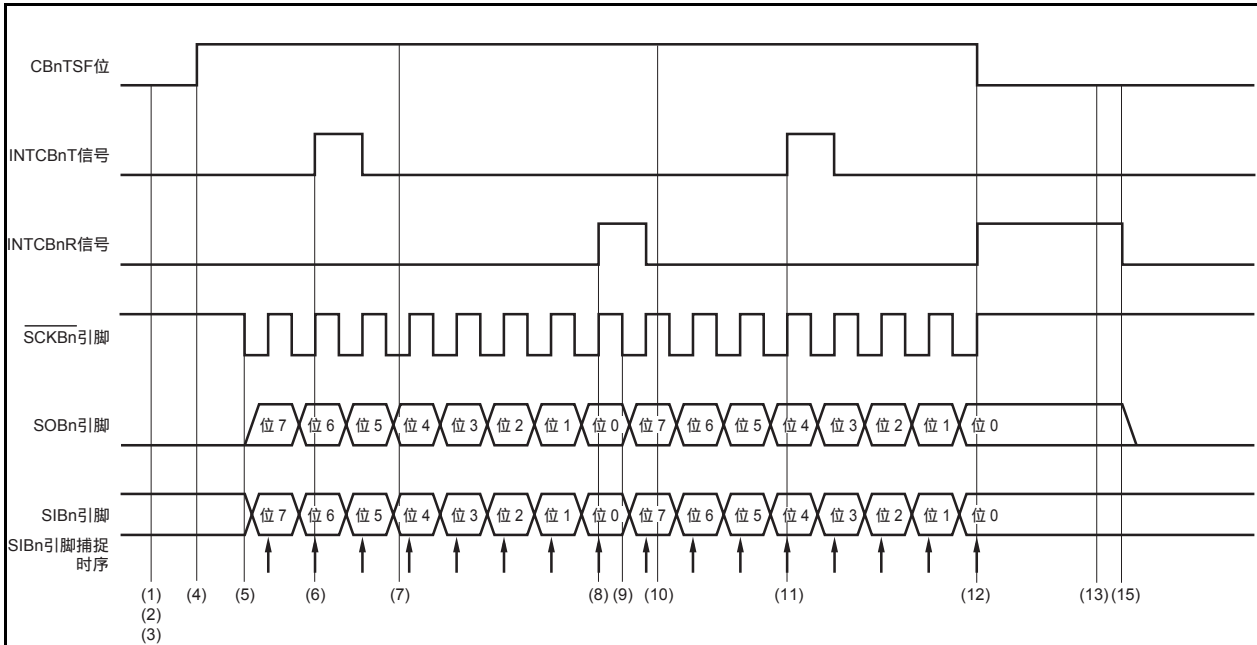


备注

1. 虚线表示硬件处理。
2. 该图中的数字相应于在 (2) 操作时序中的步骤编号。
3. $n = 0, 1$

(2) 操作时序

(1/2)



- (1) 将 07H 写入 CBnCTL1 寄存器，并选择通信类型 1，通信时钟 (f_{CCLK}) = 外部时钟 (SCKBn) 以及从模式。
- (2) 将 00H 写入 CBnCTL2 寄存器，并将发送数据长度设置为 8 位。
- (3) 将 E3H 写入 CBnCTL0 寄存器，并在允许操作通信时钟 (f_{CCLK}) 的同时选择发送/接收模式，MSN 优先以及连续发送模式。
- (4) 通过将发送数据写入 CBnRX 寄存器将 CBnSTR.CBnTSF 设为 1，而设备则将会等待串行时钟输入。
- (5) 当输入串行时钟时，将会与串行时钟同步将发送数据输出到 SOBn 引脚中，然后获取 SIBn 引脚的接收数据。
- (6) 当完成发送数据从 CBnTX 寄存器到移位寄存器的转移并且允许对 CBnTX 寄存器进行写操作时，将会生成发送允许中断请求信号 (INTCBnT)。
- (7) 要继续发送，则在生成 INTCBnT 信号后将发送数据再次写入 CBnTX 寄存器。
- (8) 当 CBnCTL2 寄存器所调协的发送数据长度的接收完成时，将会生成接收完成中断请求信号 (INTCBnR)，并允许对 CBnRX 寄存器进行读取操作。
- (9) 当连续输入另一个串行时钟时，将会开始连续发送/接收。
- (10) 读取 CBnRX 寄存器。
- (11) 当完成发送数据从 CBnTX 寄存器到移位寄存器的转移并且允许对 CBnTX 寄存器进行写操作时，将会生成 INTCBnT 信号。要通过当前发送/接收来结束连续发送/接收，则不要对 CBnTX 寄存器进行写操作。

备注 n = 0, 1

- (12) 在没有对 CBnTX 寄存器进行写操作的情况下，若 CBnCTL2 寄存器所设置的发送数据长度的时钟被输入，则将会生成 INTCBnR 信号。且 CBnTSF 位将会被清 0 以终止发送/接收。
- (13) 当 INTCBnR 被生成时，将会读取 CBnRX 寄存器。
- (14) 如果发生溢出错误，则将 CBnSTR.CBnOVE 位设为 0 并清除错误标志。
- (15) 要释放发送/接收允许状态，则在核实 CBnTSF 位 = 0 后将 CBnCTL0.CBnPWR 位及 CBnCTL0.CBnTXE 位设为 0。

备注 n = 0, 1

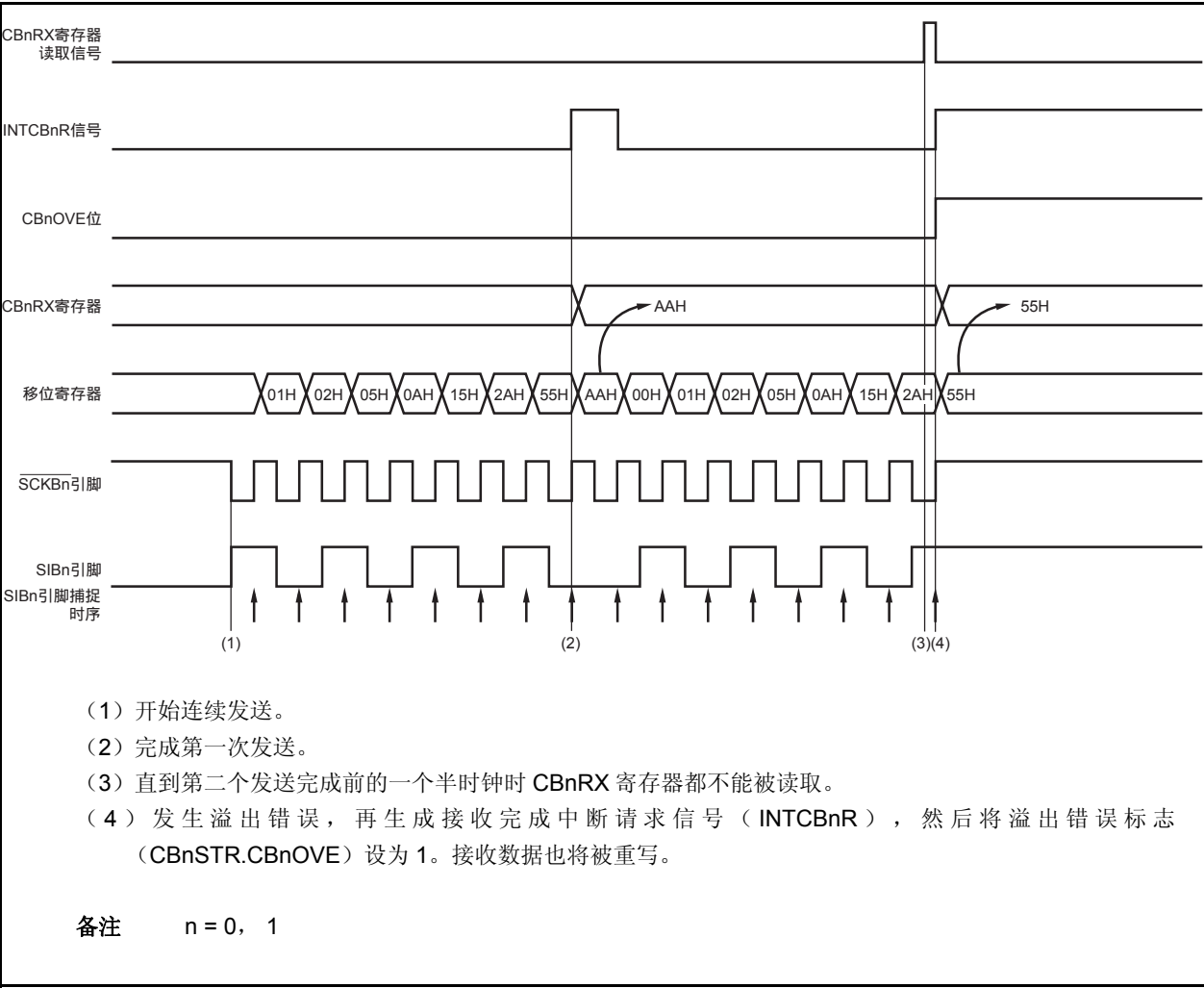
13.5.13 接收错误

在连续发送模式中，当在允许接收（CBnCTL0.CBnRXE bit = 1）的情况下执行发送时，将会在生成 INTCBnR 信号后对 CBnRX 寄存器进行读取操作，而在此之前，接收完成中断请求信号（INTCBnR）则会在下一次操作完成时再次生成，同时，溢出错误标志（CBnSTR.CBnOVE）将会被设为 1。

即使已经发生溢出错误，先前的接收数据也会因为 CBnRX 寄存器的更新而丢失。而即使已经发生接收错误，如果 CBnRX 寄存器没有被读取，在下一次接收完成的情况下也将会生成 INTCBnR 信号。

为了避免溢出错误，应在从 INTCBnR 信号生成中对下一个接收数据的最后一位进行采样前的一个半时钟内完成对 CBnRX 寄存器的读取操作。

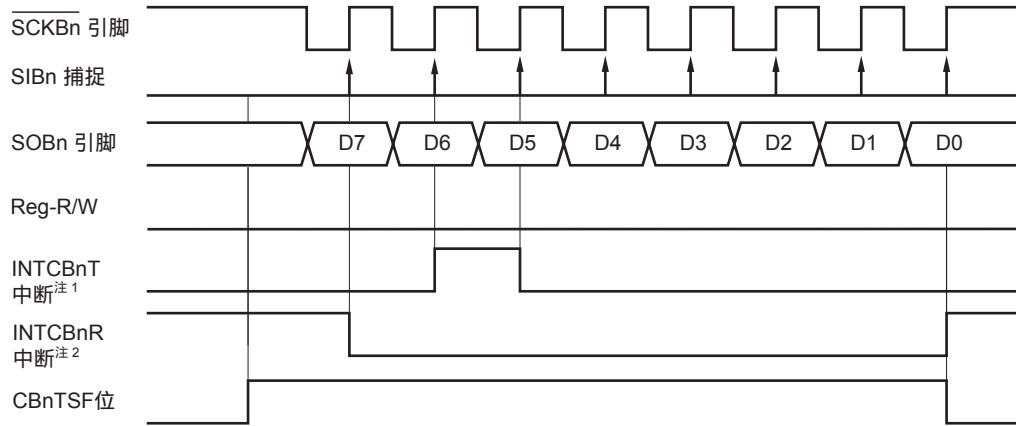
(1) 操作时序



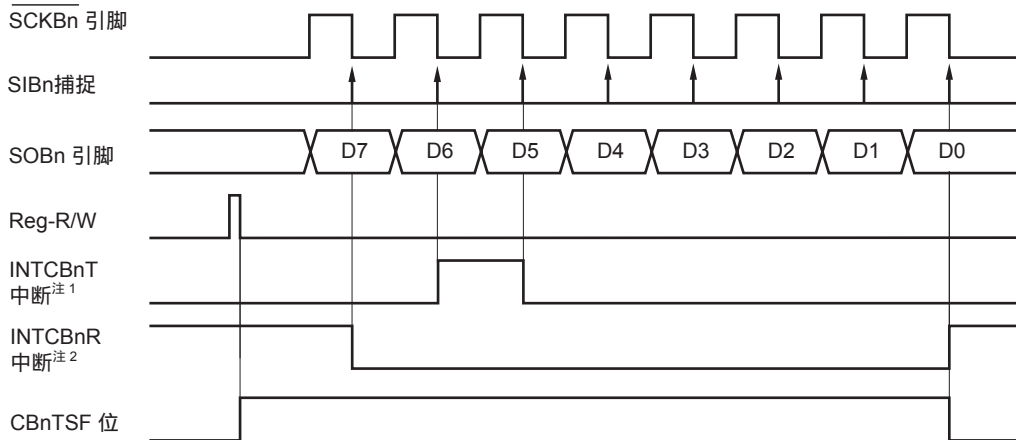
13.5.14 时钟时序

(1/2)

(i) 通信类型 1 (CBnCKP 和 CBnDAP 位 = 00)



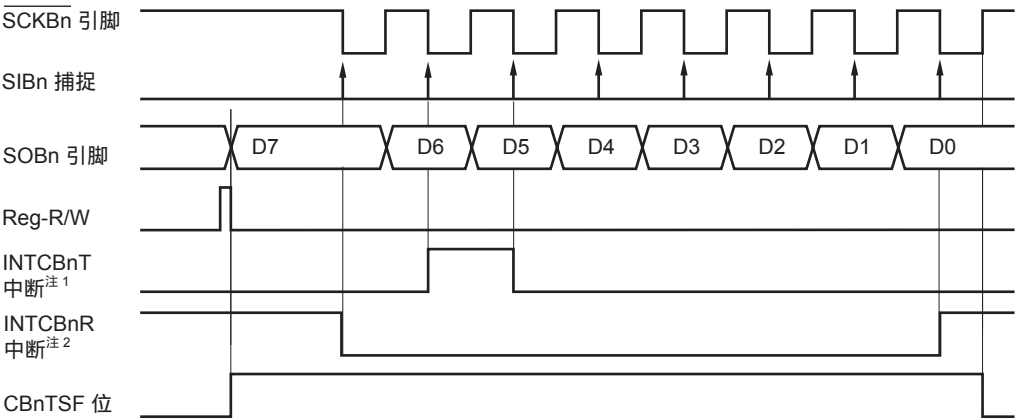
(ii) 通信类型 3 (CBnCKP 和 CBnDAP 位 = 10)



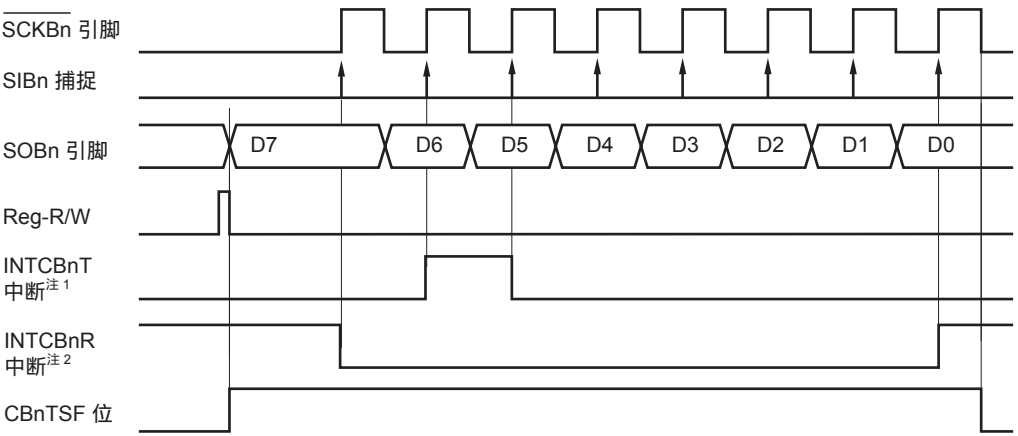
- 注**
1. 连续发送或连续发送/接收模式下，当写入 CBnTX 寄存器的数据被发送到数据移位寄存器时，将会产生 INTCBnT 中断。在单独发送或单独发送/接收模式下，不产生中断请求信号 INTCBnT，但是通信结束后将产生中断请求信号 INTCBnR。
 2. 如果接收准确完成，且允许接收时接收数据已经存在于 CBnRX 寄存器中，则将会发生 INTCBnR 中断。在单独模式下，在通信结束的情况下，即使在发送模式中也会产生 INTCBnR 中断请求信号。

注意事项 在单独发送模式中，将设为 1 的 CBnTSF 位写入 CBnTX 寄存器的操作将被忽略。这在发送期间不会对操作产生任何影响。

(iii) 通信类型 2 (CBnCKP 和 CBnDAP 位= 01)



(iv) 通信类型 4 (CBnCKP 和 CBnDAP 位= 11)



- 注
1. 连续发送或连续发送/接收模式下，当写入 CBnTX 寄存器的数据被发送到数据移位寄存器时，将会产生 INTCBnT 中断。在单独发送或单独发送/接收模式下，不产生中断请求信号 INTCBnT，但是通信结束后将产生中断请求信号 INTCBnR。
 2. 如果接收准确完成，且允许接收时接收数据已经存在于 CBnRX 寄存器中，则将会发生 INTCBnR 中断。在单独模式下，在通信结束的情况下，即使在发送模式中也会产生 INTCBnR 中断请求信号。

注意事项 在单独发送模式中，将设为 1 的 CBnTSF 位写入 CBnTX 寄存器的操作将被忽略。这在发送期间不会对操作产生任何影响。

13.6 禁止操作时的输出引脚状态

(1) $\overline{\text{SCKBn}}$ 引脚

当 CSIBn 的操作被禁止 (CBnCTL0.CBnPWR 位 = 0) 时, $\overline{\text{SCKBn}}$ 引脚输出状态如下所示。

CBnCKS2	CBnCKS1	CBnCKS0	CBnCKP	$\overline{\text{SCKBn}}$ 引脚输出
1	1	1	×	高阻抗
除上述情况			0	固定于高电平
			1	固定于低电平

- 备注**
1. 如果 CBnCTL1.CBnCKP 和 CBnCKS2 至 CBnCKS0 位中的任何位被复写, 引脚 $\overline{\text{SCKBn}}$ 的输出电平将改变。
 2. n = 0, 1
 3. ×: 不用考虑

(2) SOBn 引脚

当 CSIBn 的操作被禁止 (CBnPWR 位 = 0) 时, SOBn 引脚输出状态如下所示。

CBnTXE	CBnDAP	CBnDIR	SOBn 引脚输出
0	×	×	固定于低电平
1	0	×	SOBn 锁存值 (低电平)
	1	0	CBnTX 寄存器值 (MSB)
		1	CBnTX 寄存器值 (LSB)

- 备注**
1. 如果 CBnCTL0.CBnTXE, CBnCTL0.CBnDIR 位以及 CBnCTL1.CBnDAP 位中的一个被复写, 引脚 SOBn 的输出电平将改变。
 2. n = 0, 1
 3. ×: 不用考虑

13.7 波特率发生器

由波特率发生器（预分频器 3）生成的时钟被提供给钟表定时器以及 CSIB0。

(1) 预分频器模式寄存器 0 (PRSM0)

PRSM0 寄存器控制产生 CSIB 的波特率信号。
该寄存器可以以 8 位或 1 位为单位进行读取或写入。
复位后将该寄存器设置为 00H。

复位后：00H R/W 地址：FFFFF8B0H

	7	6	5	4	3	2	1	0
PRSM0	0	0	0	BGCE0	0	0	BGCS01	BGCS00

BGCE0	波特率输出
0	禁止
1	允许

BGCS01	BGCS00	计数时钟选择 (f _{BGCS})		
			5 MHz	4 MHz
0	0	f _x	200 ns	250 ns
0	1	f _x /2	400 ns	500 ns
1	0	f _x /4	800 ns	1 ns
1	1	f _x /8	1.6 ns	2 ns

- 注意事项
1. 在钟表定时器及 CSIB0 操作时请不要复写寄存器 PRSM0。
 2. 在将 BGCE0 位设置为 1 之前设置寄存器 PRSM0。

(2) 预分频器比较寄存器 0 (PRSCM0)

PRSCM0 寄存器是一个 8 位比较寄存器。
该寄存器可以以 8 位为单位进行读取或写入。
复位后将该寄存器设置为 00H。

复位后：00H R/W 地址：FFFFF8B1H

PRSCM0

7	6	5	4	3	2	1	0
PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00

注意事项

1. 在钟表定时器及 CSIB 操作时请不要复写寄存器 PRSCM0。

2. 在将 PRSM0.BGCE0 位设置为 1 之前设置寄存器 PRSCM0。

13.7.1 波特率产生

发送/接收时钟通过划分主时钟来生成。从主时钟中产生的波特率可以通过以下的公式获取。

$$f_{BRG} = \frac{f_{xx}}{2^{k+1} \times N}$$

<R> 注意事项 将 fBRGm 设置为 8 MHz 或更低。

备注 fBRG: BRG 计数时钟
 fxx: 主时钟振动频率
 k: PRSM0 寄存器设置值= 0 至 3
 N: PRSCM0 寄存器设置值 = 1 至 256
 然而，只有当寄存器 PRSCM0 被设为 00H 时，N = 256。

13.8 注意事项

- (1) 考虑到寄存器禁止在操作期间 (CBnCTL0.CBnPWR 位为 1) 复写, 如果操作期间由于错误操作而复写寄存器, 则设置 CBnCTL0.CBnPWR 位为 0, 然后初始化 CSIBn。

禁止在操作期间复写的寄存器如下。

- CBnCTL0 寄存器: CBnTXE, CBnRXE, CBnDIR, CBnTMS 位
- CBnCTL1 寄存器: CBnCKP, CBnDAP, CBnCKS2 至 CBnCKS0 位
- CBnCTL2 寄存器: CBnCL3 至 CBnCL0 位

- (2) 在通信类型 2 或 4 (CBnCTL1.CBnDAP = 1) 中, 产生接收完成中断 (INTCBnR) 后, SCKBn 时钟执行到一半时 CBnSTR.CBnTSF 被清除。

在单独发送模式中, 通信过程中 (CBnTSF = 1) 忽略写入的下一个发送数据, 并且下一次通信不开始。同样, 如果只能接收的通信被设置 (CBnCTL0.CBnTXE = 0, CBnCTL0.CBnRXE = 1), 即使接收到的数据在通信过程中 (CBnTSF = 1) 被读取, 下一次通信也不会开始。

因此, 在通信模式 2 或 4 (CBnDAP = 1) 下的单独发送模式, 特别要注意下述问题。

- 要开始下一次发送, 确保 CBnTSF = 0, 然后向寄存器 CBnTX 写入要发送的数据。
- 只能接收通信模式下 (CBnTXE = 0, CBnRXE = 1), 要连续执行下一次接收, 应确保 CBnTSF = 0, 然后读取寄存器 CBnTX。

或者, 使用连续发送模式来代替单独发送模式。

备注 n = 0, 1

第 14 章 中断/异常处理功能

V850ES/HE2 为中断服务提供一个专门的中断控制器（INTC），它总共可以处理 41 个中断请求。

中断是指不受程序执行影响的事件发生，异常是指依赖于程序执行的事件发生。

V850ES/HE2 可以处理由片上外围硬件和外部事件源引起的中断请求信号。此外，异常处理可以通过 TRAP 指令（软件异常）或者产生一个异常事件（例如：存在非法操作码）（异常陷阱）开始。

14.1 特性

○ 中断

- 非屏蔽中断：2 个中断源
- 可屏蔽中断：外部：8 个中断源，内部：31 个中断源
- 8 个可编程中断优先级级别（针对可屏蔽中断）。
- 依据优先级进行复合中断控制。
- 针对可屏蔽中断，可以使用屏蔽选项。
- 噪声消除，边沿检测，和外部中断请求信号有效边沿说明。

○ 异常

- 软件异常：32 个中断源
- 异常陷阱：2 个异常源（非法的操作码异常，调试陷阱）

中断/异常源列于表 14-1 中。

表 14-1. 中断源列表 (1/2)

类型	级别	默认优先级	名称	触发	产生单元	异常代码	处理器地址	还原的 PC	中断控制寄存器
复位	中断	–	RESET	RESET 引脚输入 通过内部源的复位输入	RESET	0000H	00000000H	未定义	–
非屏蔽中断	中断	–	NMI	NMI 引脚有效边沿输入	引脚	0010H	00000010H	下一个 PC	–
		–	INTWDT2	WDT2 溢出	WDT2	0020H	00000020H	注 1	–
软件异常	异常	–	TRAP0n ^{#2}	TRAP 指令	–	004nH ^{#2}	00000040H	下一个 PC	–
		–	TRAP1n ^{#2}	TRAP 指令	–	005nH ^{#2}	00000050H	下一个 PC	–
异常陷阱	异常	–	ILGOP/ DBG0	非法操作码/ DBTRAP 指令	–	0060H	00000060H	下一个 PC	–
可屏蔽	中断	0	INTLVI	低电压检测	POCLVI	0080H	00000080H	下一个 PC	LVIIIC
		1	INTP0	外部中断引脚输入边沿检测 (INTP0)	引脚	0090H	00000090H	下一个 PC	PIC0
		2	INTP1	外部中断引脚输入边沿检测 (INTP1)	引脚	00A0H	000000A0H	下一个 PC	PIC1
		3	INTP2	外部中断引脚输入边沿检测 (INTP2)	引脚	00B0H	000000B0H	下一个 PC	PIC2
		4	INTP3	外部中断引脚输入边沿检测 (INTP3)	引脚	00C0H	000000C0H	下一个 PC	PIC3
		5	INTP4	外部中断引脚输入边沿检测 (INTP4)	引脚	00D0H	000000D0H	下一个 PC	PIC4
		6	INTP5	外部中断引脚输入边沿检测 (INTP5)	引脚	00E0H	000000E0H	下一个 PC	PIC5
		7	INTP6	外部中断引脚输入边沿检测 (INTP6)	引脚	00F0H	000000F0H	下一个 PC	PIC6
		8	INTP7	外部中断引脚输入边沿检测 (INTP7)	引脚	0100H	00000100H	下一个 PC	PIC7
		9	INTTQ0OV	TMQ0 溢出	TMQ0	0110H	00000110H	下一个 PC	TQ0OVIC
		10	INTTQ0CC0	TMQ0 捕获 0/比较 0 匹配	TMQ0	0120H	00000120H	下一个 PC	TQ0CCIC 0
		11	INTTQ0CC1	TMQ0 捕获 1/比较 1 匹配	TMQ0	0130H	00000130H	下一个 PC	TQ0CCIC 1
		12	INTTQ0CC2	TMQ0 捕获 2/比较 2 匹配	TMQ0	0140H	00000140H	下一个 PC	TQ0CCIC 2
		13	INTTQ0CC3	TMQ0 捕获 3/比较 3 匹配	TMQ0	0150H	00000150H	下一个 PC	TQ0CCIC 3
		14	INTTP0OV	TMP0 溢出	TMP0	0160H	00000160H	下一个 PC	TP0OVIC
		15	INTTP0CC0	TMP0 捕获 0/比较 0 匹配	TMP0	0170H	00000170H	下一个 PC	TP0CCIC0
		16	INTTP0CC1	TMP0 捕获 1/比较 1 匹配	TMP0	0180H	00000180H	下一个 PC	TP0CCIC1
		17	INTTP1OV	TMP1 溢出	TMP1	0190H	00000190H	下一个 PC	TP1OVIC
		18	INTTP1CC0	TMP1 捕获 0/比较 0 匹配	TMP1	01A0H	000001AH	下一个 PC	TP1CCIC0
		19	INTTP1CC1	TMP1 捕获 1/比较 1 匹配	TMP1	01B0H	000001B0H	下一个 PC	TP1CCIC1
		20	INTTP2OV	TMP2 溢出	TMP2	01C0H	000001C0H	下一个 PC	TP2OVIC
		21	INTTP2CC0	TMP2 捕获 0/比较 0 匹配	TMP2	01D0H	000001D0H	下一个 PC	TP2CCIC0

- 注 1. 在 INTWDT2 例子中的还原, 见 14.2.2 (2) 通过 INTWDT2 信号。
2. n = 0H 至 FH

表 14-1. 中断源列表 (2/2)

类型	级别	默认优先级	名称	触发	产生单元	异常代码	处理器地址	还原的 PC	中断控制寄存器
可屏蔽	中断	22	INTTP2CC1	TMP2 捕获 1/比较 1 匹配	TMP2	01E0H	000001E0H	下一个 PC	TP2CCIC1
		23	INTTP3OV	TMP3 溢出	TMP3	01F0H	000001F0H	下一个 PC	TP3OVIC
		24	INTTP3CC0	TMP3 捕获 0/比较 0 匹配	TMP3	0200H	00000200H	下一个 PC	TP3CCIC0
		25	INTTP3CC1	TMP3 捕获 1/比较 1 匹配	TMP3	0210H	00000210H	下一个 PC	TP3CCIC1
		26	INTTM0EQ0	TMM0 比较 匹配	TMM0	0220H	00000220H	下一个 PC	TM0EQIC0
		27	INTCB0R	CSIB0 接收完成	CSIB0	0230H	00000230H	下一个 PC	CB0RIC
		28	INTCB0T	CSIB0 连续传送写入允许	CSIB0	0240H	00000240H	下一个 PC	CB0TIC
		29	INTCB1R	CSIB1 接收完成	CSIB1	0250H	00000250H	下一个 PC	CB1RIC
		30	INTCB1T	CSIB1 连续传送写入允许	CSIB1	0260H	00000260H	下一个 PC	CB1TIC
		31	INTUA0R	UARTA0 接收完成	UARTA0	0270H	00000280H	下一个 PC	UA0RIC
		32	INTUA0T	UARTA0 发送允许	UARTA0	0280H	00000280H	下一个 PC	UA0TIC
		33	INTUA1R	UARTA1 接收完成/UARTA1 接收错误	UARTA1	0290H	00000290H	下一个 PC	UA1RIC
		34	INTUA1T	UARTA1 发送允许	UARTA1	02A0H	000002A0H	下一个 PC	UA1TIC
		35	INTAD	A/D 转换完成	A/D	02BH	000002B0H	下一个 PC	ADIC
		36	INTKR	按键返回中断请求	KR	0300H	00000300H	下一个 PC	KRIC
		37	INTWTI	钟表定时器间隔	WT	0310H	00000310H	下一个 PC	WTIIC
		38	INTWT	钟表定时器参考时间	WT	0320H	00000320H	下一个 PC	WTIC

- 备注**
- 默认优先级: 表示当两个或更多的可屏蔽中断请求同时发生时, 它们的有限顺序。最高值为 0。
非屏蔽中断的优先级顺序为 INTWDT2 > NMI。
还原的 PC: 当中断服务产生时被保存到 EIPC, FEPC, 或是 DBPC 的程序计数器 (PC) 的值。但是, 非屏蔽中断或者可屏蔽中断被确认, 且下列指令之一正在被执行时, 还原的 PC 不会成为下一个 PC (如果在中断执行期间确认中断, 则该操作停止并在中断服务结束后还原)。
 - 载入指令 (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
 - 除法指令 (DIV, DIVH, DIVU, DIVHU)
 - PREPARE, DISPOSE 指令 (仅限中断在堆栈指针更新之前产生的情况)下一个 PC: 在中断/异常处理之后的开始处理的 PC 的值。
 - 当一个非法的操作代码异常发生时, 非法指令的执行地址通过保存的 PC 值减去 4 得出。

14.2 非屏蔽中断

非屏蔽中断请求会被无条件的确认，即使是在禁止中断的条件下（DI）。一个非屏蔽中断不会受到优先级的控制，并且相对于其它的中断请求而言具有绝对的优先权。

在本产品中有下列两种非屏蔽中断请求信号。

- NMI 引脚输入（NMI）
- 非屏蔽中断请求信号可以通过看门狗定时器溢出产生（INTWDT2）

NMI 引脚的有效边沿可以选择以下四种方式：“上升沿”，“下降沿”，“双边沿”和“不检测边沿”。

通过将 PMC0.PMC02 位设为 1，将 INTF0.INTF02 位及 INTR0.INTR02 位设成一个期望值并指定一个期望的有效沿来允许 NMI 引脚的功能。

当 WDTM2.WDM21 位和 WDTM2.WDM20 位设为“01”时，由看门狗定时器 2 溢出而产生的非屏蔽中断请求信号（INTWDT2）生效。

如果两个或是更多的非屏蔽中断请求同时产生，具有较高优先级的中断首先生效，如下所示（具有较低优先级的中断请求信号将被忽略）。

INTWDT2 > NMI

如果在非屏蔽中断正在进行的同时，发出一个新的 NMI 或者 INTWDT2 请求信号，将按照如下方式进行。

（1）如果在非屏蔽中断正在进行的同时，发出一个新的 NMI 请求信号

新的 NMI 请求信号被保持，忽略 PSW.NP 位的值。而新的 NMI 请求信号会在当前的 NMI 运行完毕后被确认（在 RETI 指令执行后）。

（2）如果在非屏蔽中断正在进行的同时，发出一个 INTWDT2 请求信号

当存在正在进行的 NMI 时，如果 NP 位的值是（1），则 INTWDT2 请求信号被保持。被保持的 INTWDT2 请求信号会在当前的 NMI 运行完毕后被确认（在 RETI 指令执行后）。

当存在正在进行的 NMI 时，如果 NP 位为（0），则立即执行新产生的 INTWDT2 请求信号（NMI 服务停止）。

注意事项 关于非屏蔽中断请求信号（INTWDT2）执行的非屏蔽中断服务，参考 14.2.2（2）通过 INTWDT2 信号。

图 14-1. 非屏蔽中断请求信号确认操作（1/2）

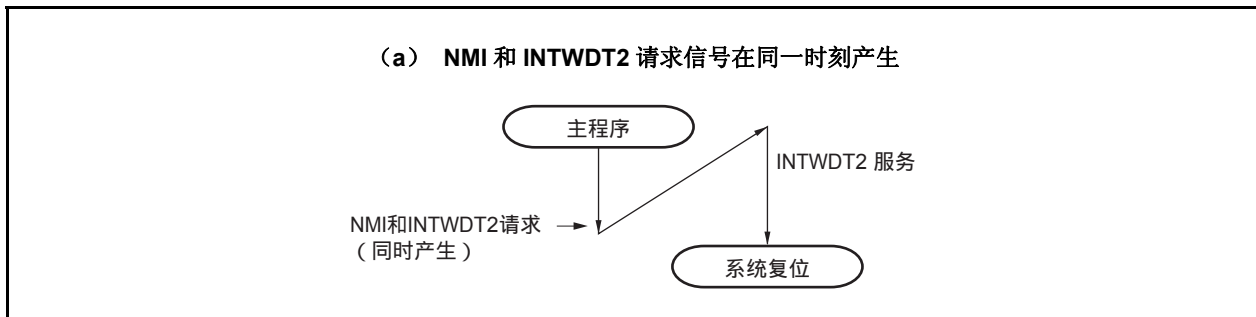
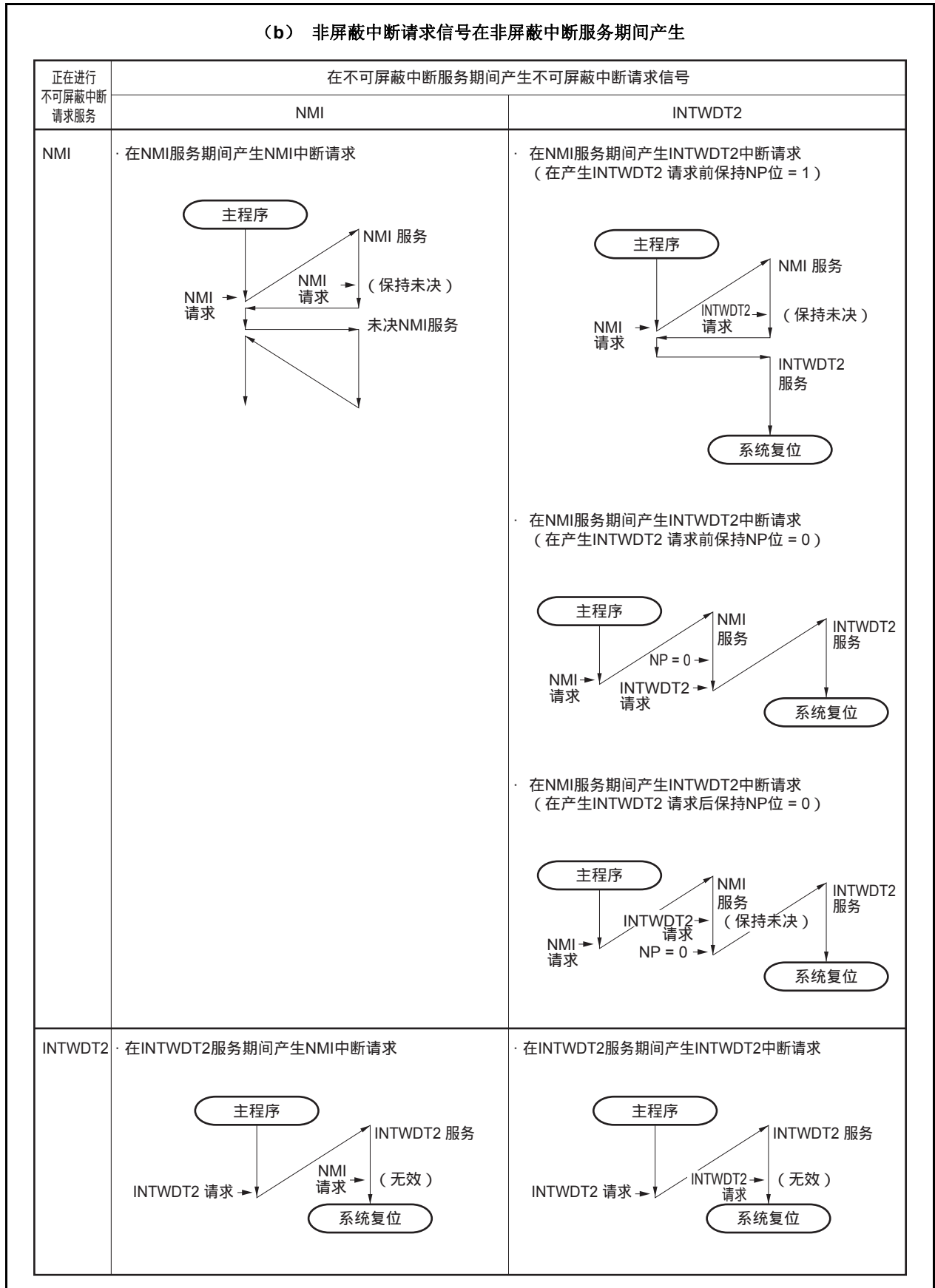


图 14-1. 非屏蔽中断请求信号确认操作 (2/2)



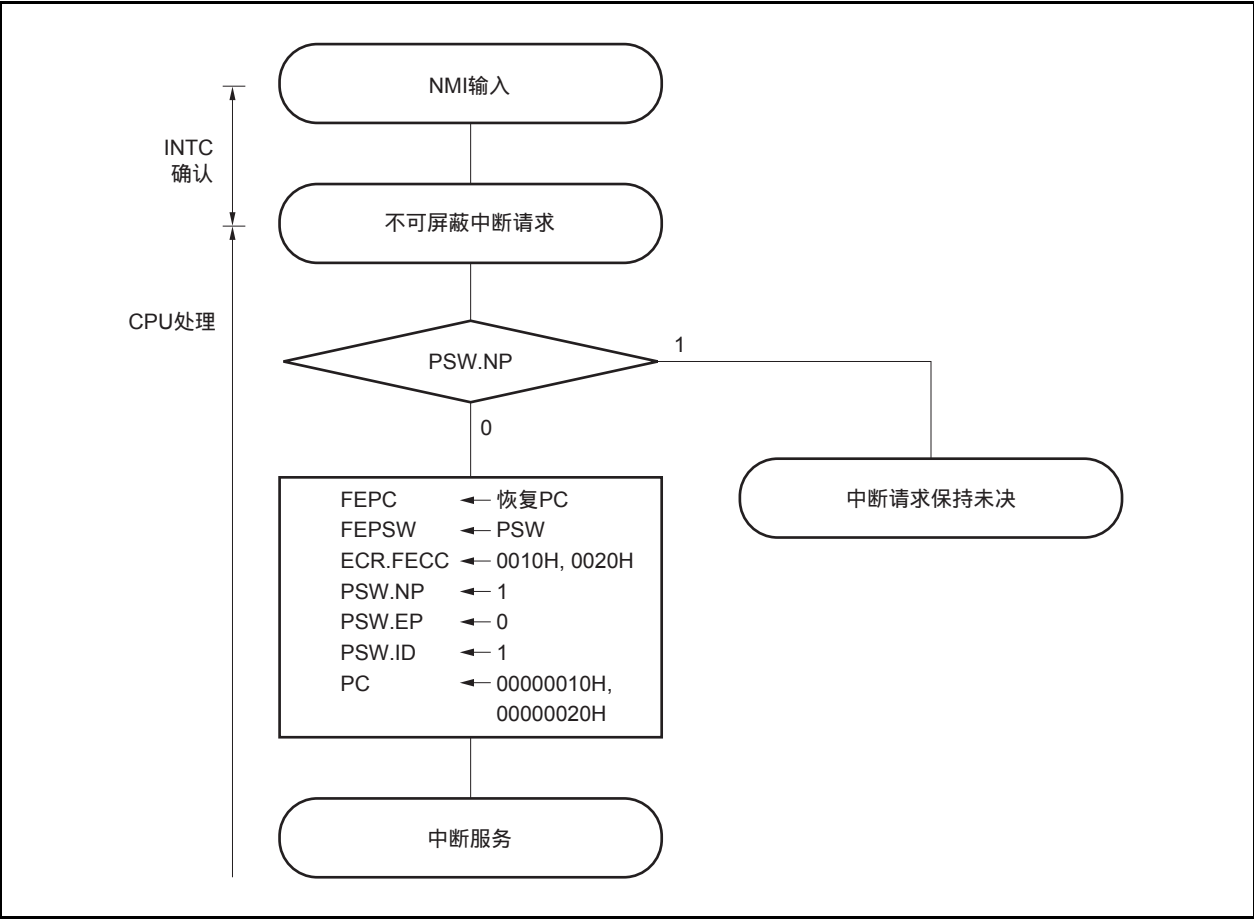
14.2.1 操作

如果产生一个非屏蔽中断请求信号，CPU 执行下列处理，并且将操作权转移给处理例程。

- <1> 将还原 PC 保存到 FEPC。
- <2> 将当前的 PSW 保存到 FEPSW。
- <3> 向 ECR 的高半字 (FECC) 写入异常代码 (0010H, 0020H)。
- <4> 将 PSW.NP 位和 PSW.ID 位置 1 并将 PSW.EP 位清 0。
- <5> 在 PC 中设置相应的非屏蔽中断的处理器地址 (00000010H, 00000020H)，然后转移操作权。

非屏蔽中断的服务过程如图 14-2 所示。

图 14-2. 非屏蔽中断的服务过程



14.2.2 还原

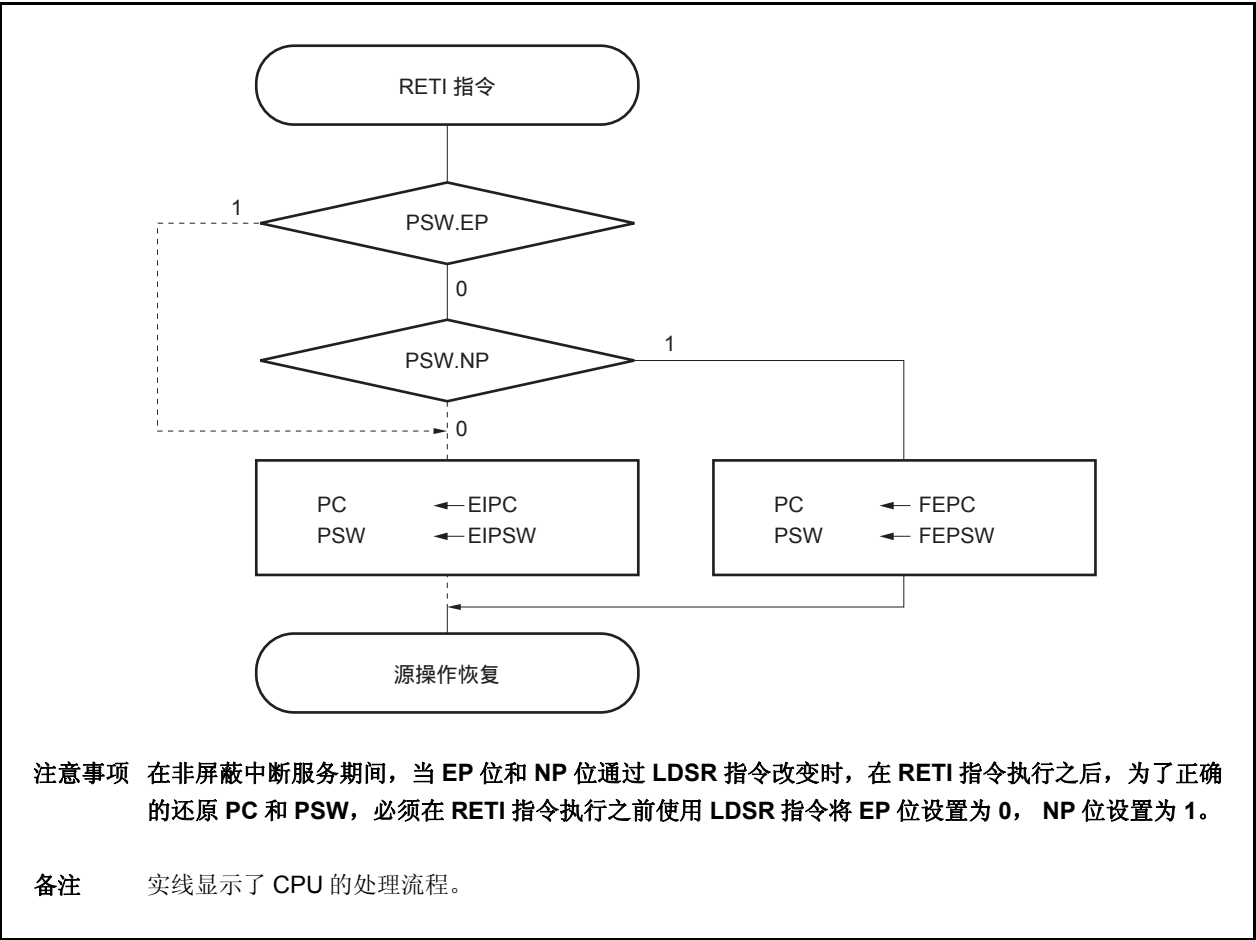
(1) 通过 NMI 引脚输入

通过 RETI 指令从 NMI 服务中执行还原。
当执行 RETI 指令时，CPU 执行下列处理，并且将操作权转移给 PC 存储的地址。

- <1> 分别从 FEPC 和 FEPSW 中加载被保存的 PC 和 PSW 的值，因为 PSW.EP 位的值是 0 且 PSW.NP 的值为 1。
- <2> 操作权转移给还原的 PC 和 PSW 的地址。

图 14-3 举例说明了 RETI 指令是如何处理的。

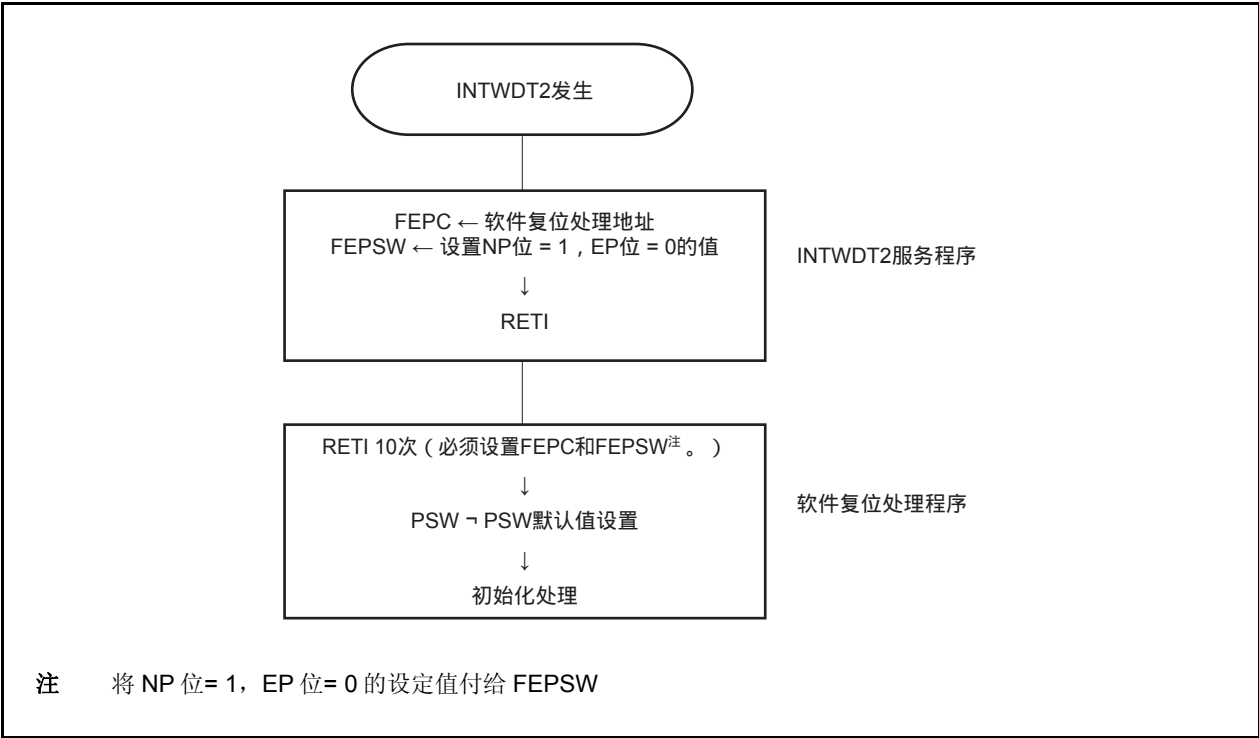
图 14-3. RETI 指令处理过程



(2) 通过 INTWDT2 信号

禁止使用 RETI 指令从非屏蔽中断服务（通过非屏蔽中断请求 INTWDT2 产生）中恢复。执行下列软件复位处理。

图 14-4. 软件复位处理



14.2.3 NP 标志

NP 标志是一个状态标志，它指示了正在执行非屏蔽中断。
当非屏蔽中断请求被确认时会设置该标志，并且还可以保证多重非屏蔽中断的情况出现。

复位后：00000020H

	31							8	7	6	5	4	3	2	1	0
PSW	0								NP	EP	ID	SAT	CY	OV	S	Z

NP	NMI中断服务状态
0	没有NMI中断服务
1	当前正在进行NMI中断服务

14.3 可屏蔽中断

可屏蔽中断请求信号可以通过中断控制寄存器屏蔽。V850ES/HE2 拥有 39 个可屏蔽中断源。

如果有两个或者更多的可屏蔽中断请求信号同时产生，它们将依照缺省优先级的顺序被确认。除了缺省优先级之外，还有八个优先级等级可以使用，这需要使用中断控制寄存器（可变成优先级控制）。

如果一个中断请求信号被确认，将禁止对其它可屏蔽中断请求信号的确认，从而进入了中断禁止状态（DI）。

当在中断服务程序中执行了 EI 指令时，进入中断允许状态（EI），这将允许具有比正在进行处理的当前中断请求信号更高优先级（通过中断控制寄存器设定）的中断。注意只有优先级更高的中断有这种功能，具有相同优先级的中断不能实现嵌套。

但是为了允许多重中断，在执行 EI 指令之前需要将 EIPC 和 EIPSW 保存到内存或是通用目的寄存器当中，并且在 RETI 指令恢复 EIPC 和 EIPSW 的原始值之前执行 DI 指令。

14.3.1 操作

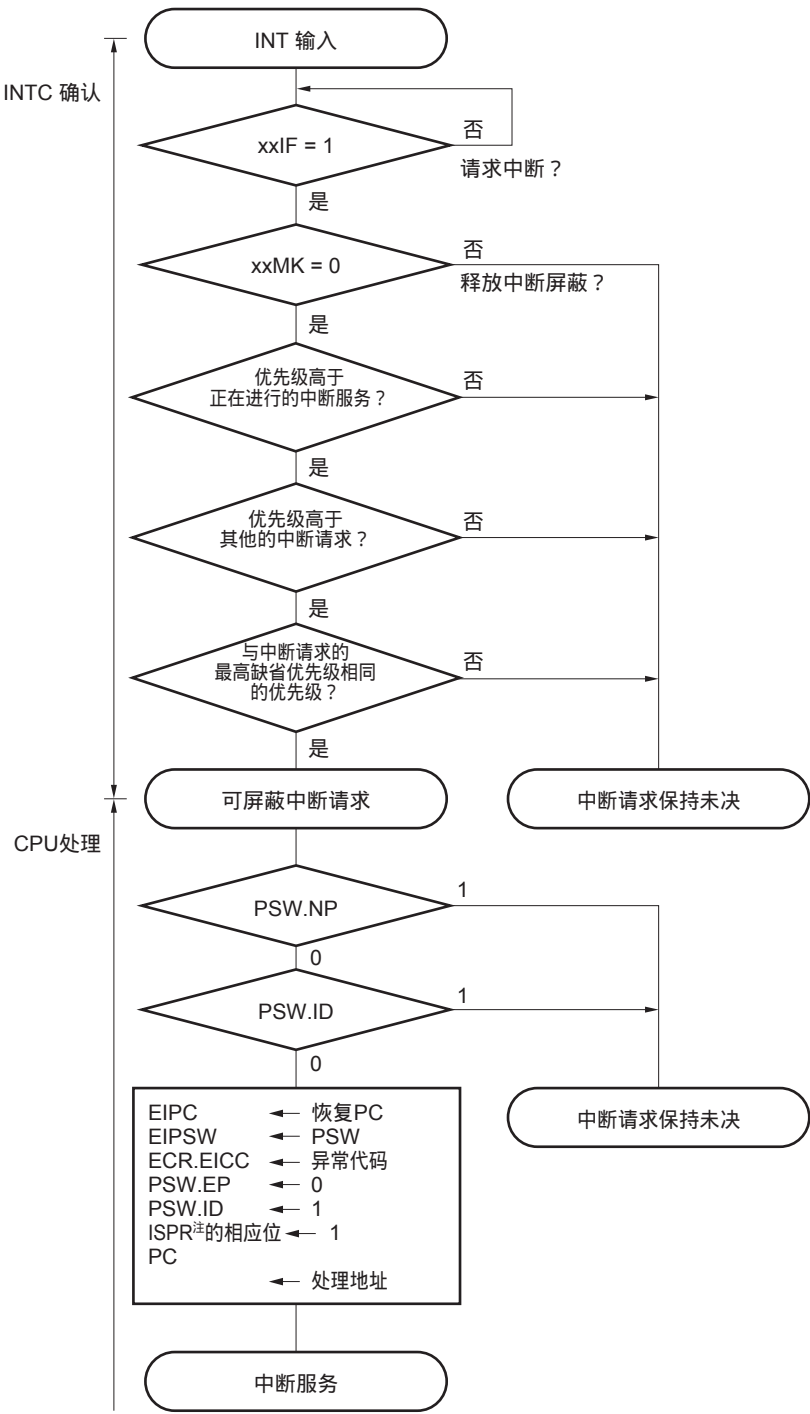
如果产生了可屏蔽中断，CPU 将执行下列处理，并且将操作权转移给处理例程。

- <1> 将恢复的 PC 保存到 EIPC。
- <2> 将当前的 PSW 保存到 EIPSW。
- <3> 向 ECR 的低半字（EICC）写入一个异常代码。
- <4> 将 PSW.ID 位置 1 且将 PSW.EP 位清 0。
- <5> 将相应中断的处理地址设置给 PC，并且转移控制权。

通过 INTC 屏蔽的可屏蔽中断请求信号和在另一个中断正在执行过程中（当 PSW.NP 位=1 或是 PSW.ID 位=1）产生的可屏蔽中断请求信号在 INTC 中被保持。在这种情况下，一个新的可屏蔽中断服务开始是与可屏蔽中断请求的优先级设定是一致的。这需要可屏蔽中断未被屏蔽，而且没有通过 RETI 指令或是 LDSR 指令将 NP 位和 ID 位清 0。

可屏蔽中断是如何执行的将在下面举例说明。

图 14-5. 可屏蔽中断服务



注 关于 ISPR 寄存器，请参考 14.3.6 正在进行服务的优先级寄存器（ISPR）。

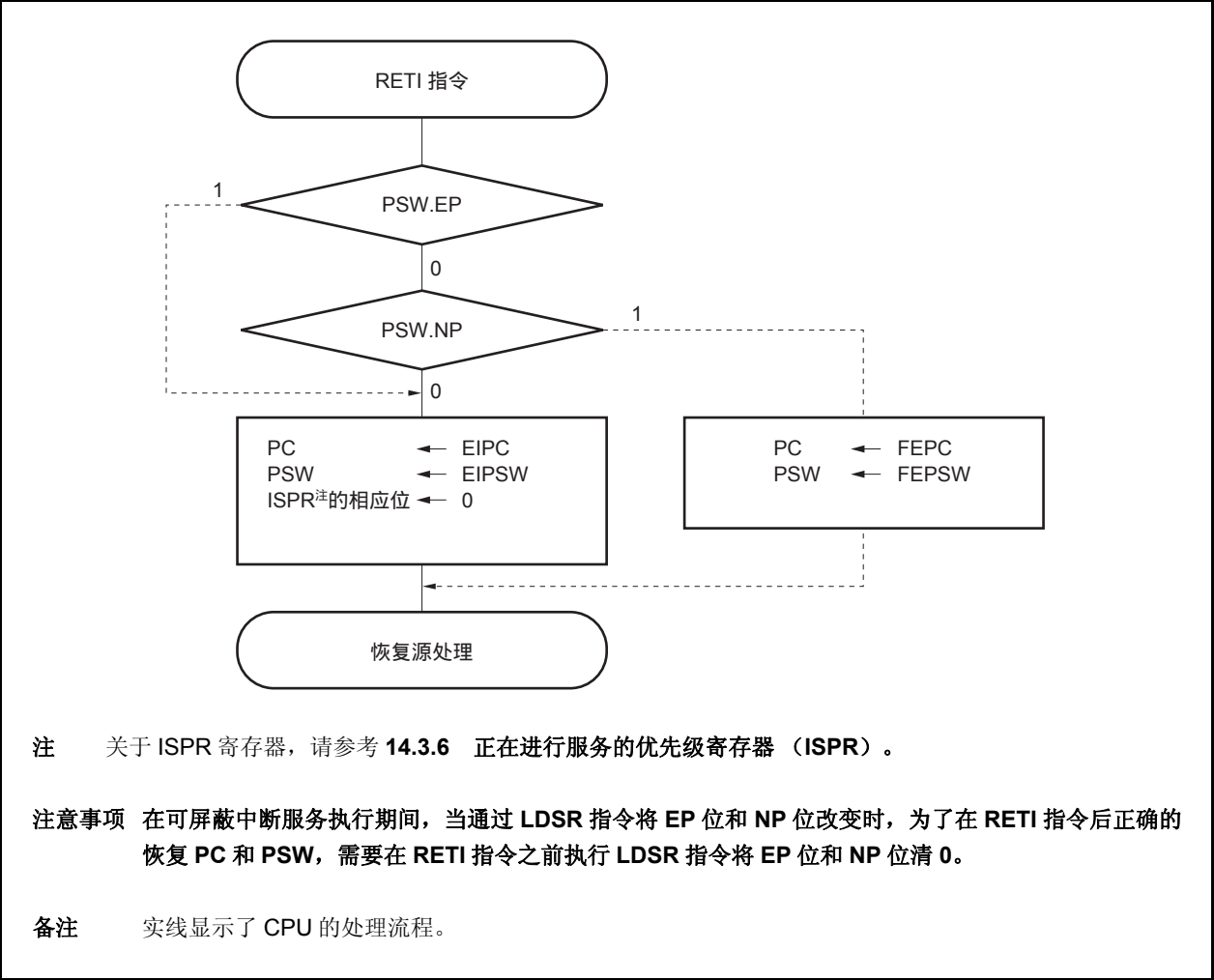
14.3.2 恢复

通过执行 RETI 指令能够从可屏蔽中断服务中恢复。
当 RETI 指令执行后，CPU 将执行下列步骤，且将操作权转移给恢复的 PC 地址。

- <1> 从 EIPC 和 EIPSW 中加载恢复的 PC 和 PSW，这是因为 PSW.EP 位为 0 并且 PSW.NP 位为 0。
- <2> 将操作权转移给恢复 PC 和 PSW 的地址。

图 14-6 举例说明 RETI 指令的处理过程。

图 14-6. 指令处理过程



14.3.3 可屏蔽中断的优先级

INTC 执行多重中断服务，这是指当一个中断服务正在进行的时候确认另外一个中断。多重中断可以通过优先级别来控制。

一共有两种类型的优先级别控制：基于缺省优先级别的控制和基于可编程优先级别的控制，后者是通过在中断控制（xxICn）寄存器的中断优先级指定位（xxPRn）的设置来实现的。当两个具有相同优先级的中断同时产生时，中断请求信号将依照预先指定给每一种中断类型的优先级别（缺省优先级别）顺序进行中断服务。如需更多的信息，参考表 14-1 中断/异常源列表。可编程的优先级通过设定优先级别制定标志分八个等级对用户化的中断请求信号进行控制。

需要注意当响应了中断请求信号，PSW.ID 标志自动设置为 1。因此，当要响应多重中断，需要预先将 ID 标志清零（例如，在中断服务程序设置 EI 指令）来设置中断允许模式。

备注 xx: 表示任意的外围单元的名称（参考表 14-2 中断控制寄存器（xxICn））。
 n: 外围单元编号（参考表 14-2 中断控制寄存器（xxICn））。

图 14-7. 另一个中断请求信号发出时的处理示例
中断服务发生时 (1/2)

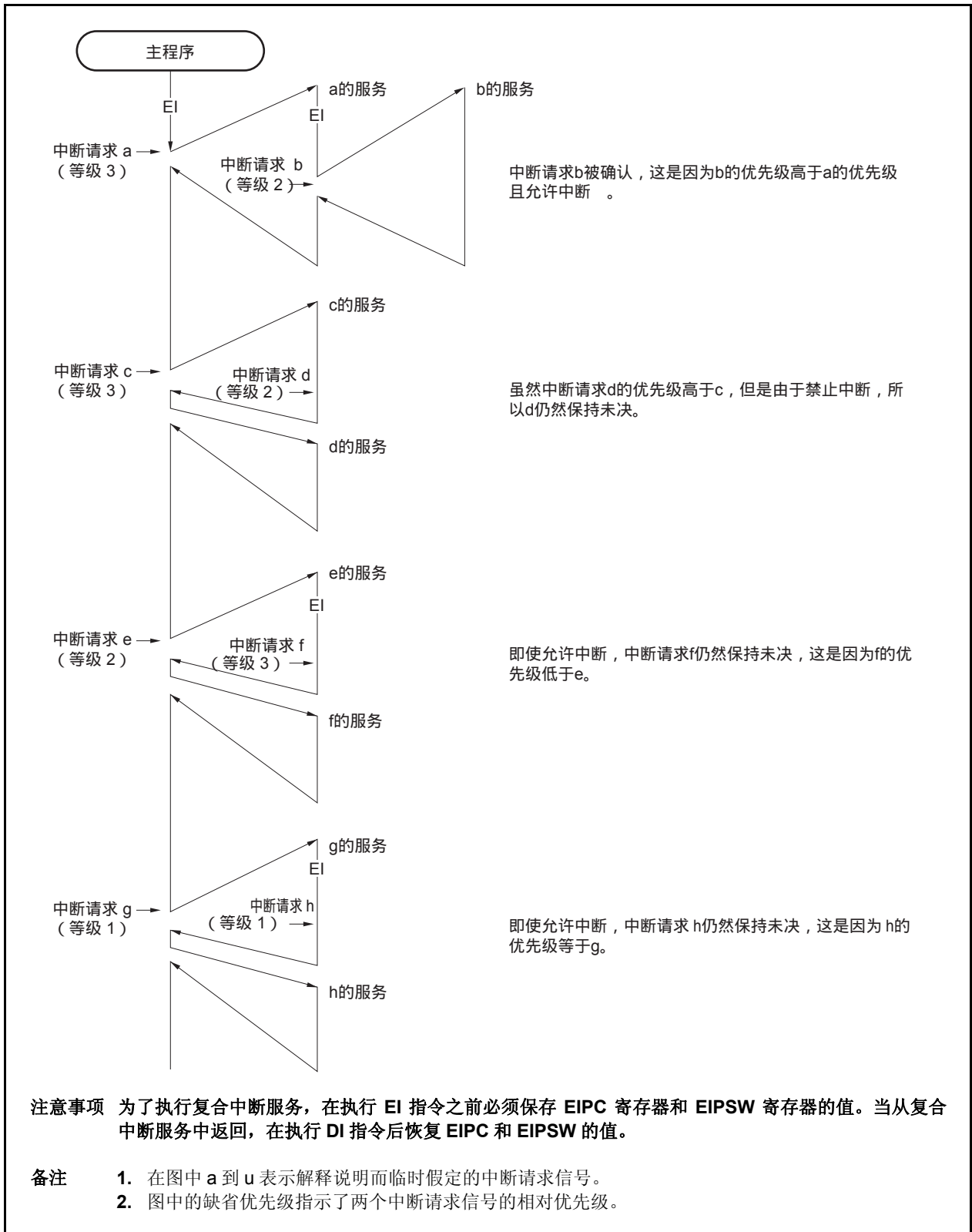
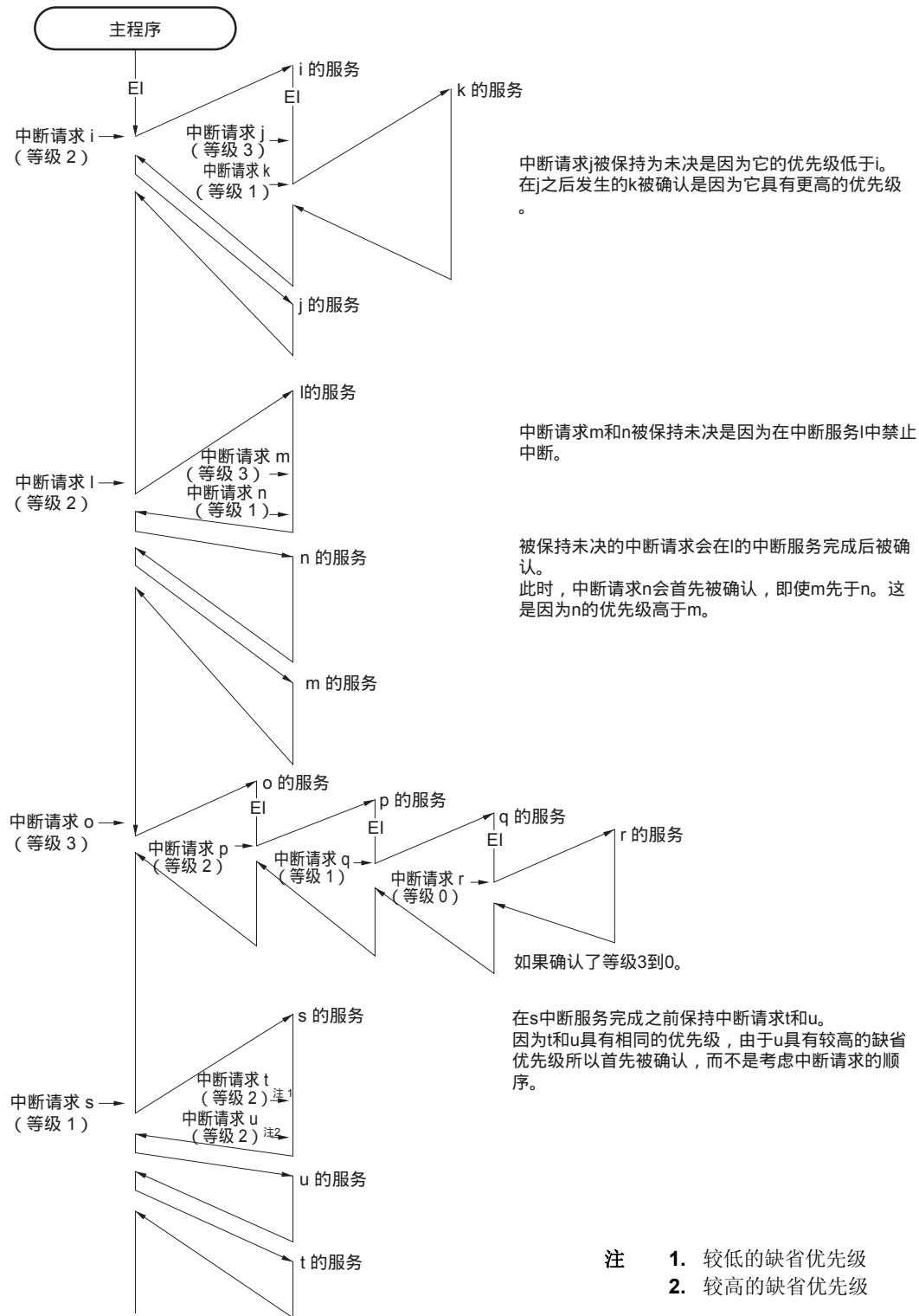
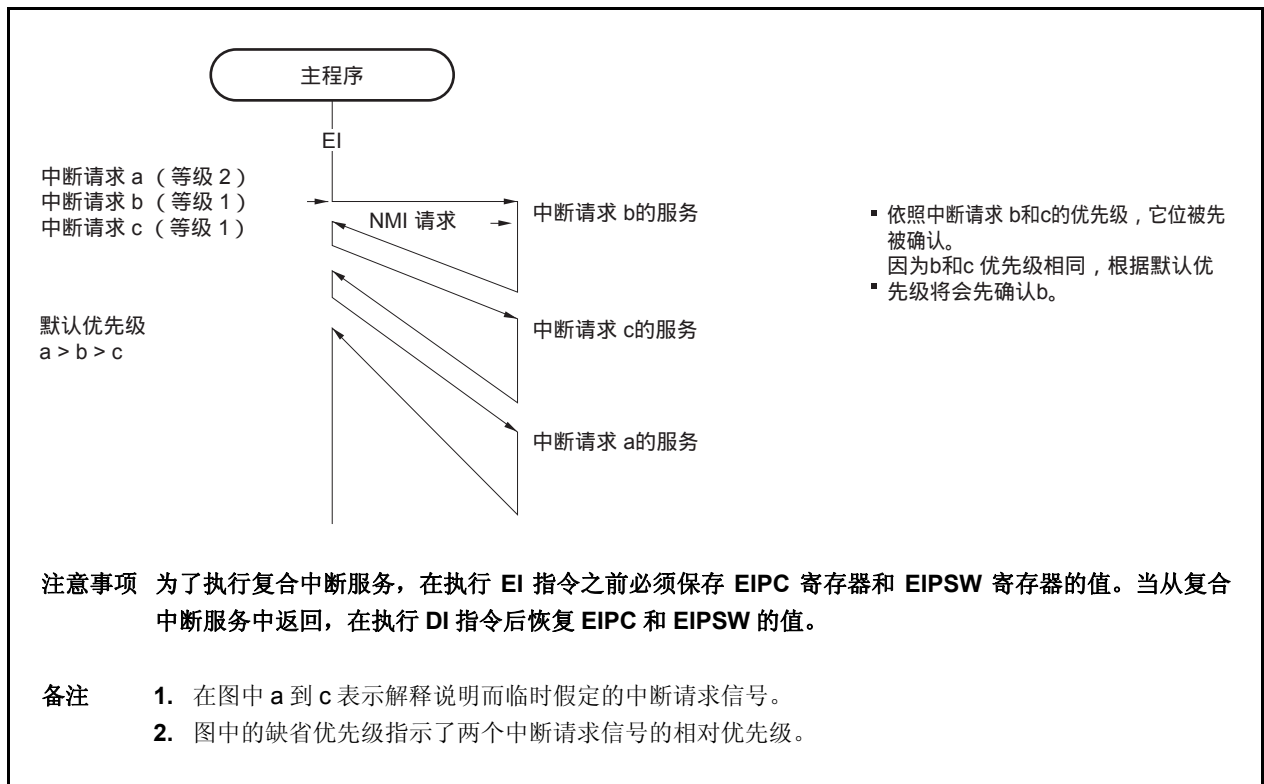


图 14-7. 另一个中断请求信号发出时的处理示例
中断服务发生时 (2/2)



注意事项 为了执行复合中断服务，在执行 EI 指令之前必须保存 EIPC 寄存器和 EIPSW 寄存器的值。当从复合中断服务中返回，在执行 DI 指令后恢复 EIPC 和 EIPSW 的值。

图 14-8. 服务中断请求信号同时发生举例



14.3.4 中断控制寄存器（xxICn）

xxICn 寄存器被分配给每一个中断请求信号（可屏蔽中断）并且设置每一个可屏蔽中断请求控制条件。
该寄存器可以通过 8 位或是 1 位单位读取或是写入。
复位后该寄存器的值为 47H。

注意事项 禁止中断（DI）或者屏蔽中断来读取 xxICn.xxIFn 位。如果在允许中断（EI）或是中断未被屏蔽的时候，读取 xxIFn 位，确认中断且读取该位冲突时数据可能有误。

复位后: 47H R/W 地址: FFFFF110H 到 FFFFF164H

	7	6	5	4	3	2	1	0
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	中断请求标志 ^注
0	不发中断请求
1	发出中断请求

xxMKn	中断屏蔽标志
0	允许中断服务
1	禁止中断服务（未决）

xxPRn2	xxPRn1	xxPRn0	中断优先级指定位
0	0	0	指定等级 0（最高）。
0	0	1	指定等级 1。
0	1	0	指定等级 2。
0	1	1	指定等级 3。
1	0	0	指定等级 4。
1	0	1	指定等级 5。
1	1	0	指定等级 6。
1	1	1	指定等级 7（最低）。

注 如果确认中断信号，xxIFn 标志将通过硬件自动复位。

备注 xx: 表示任意的外围单元的名称（参考表 14-2 中断控制寄存器（xxICn））
 n: 外围单元编号（参考 表 14-2 中断控制寄存器（xxICn））。

中断控制寄存器的位和地址如下所示。

表 14-2. 中断控制寄存器 (xxICn)

地址	寄存器	位							
		7	6	5	4	3	2	1	0
FFFFF110H	LVIIC	LVIIIF	LVIMK	0	0	0	LVIPR2	LVIPR1	LVIPR0
FFFFF112H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF114H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF116H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF118H	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
FFFFF11AH	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
FFFFF11CH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
FFFFF11EH	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60
FFFFF120H	PIC7	PIF7	PMK7	0	0	0	PPR72	PPR71	PPR70
FFFFF122H	TQ0OVIC	TQ0OVIF	TQ0OVMK	0	0	0	TQ0OVPR2	TQ0OVPR1	TQ0OVPR0
FFFFF124H	TQ0CCIC0	TQ0CCIF0	TQ0CCMK0	0	0	0	TQ0CCPR2	TQ0CCPR1	TQ0CCPR0
FFFFF126H	TQ0CCIC1	TQ0CCIF1	TQ0CCMK1	0	0	0	TQ0CCPR12	TQ0CCPR11	TQ0CCPR10
FFFFF128H	TQ0CCIC2	TQ0CCIF2	TQ0CCMK2	0	0	0	TQ0CCPR22	TQ0CCPR21	TQ0CCPR20
FFFFF12AH	TQ0CCIC3	TQ0CCIF3	TQ0CCMK3	0	0	0	TQ0CCPR32	TQ0CCPR31	TQ0CCPR30
FFFFF12CH	TP0OVIC	TP0OVIF	TP0OVMK	0	0	0	TP0OVPR2	TP0OVPR1	TP0OVPR0
FFFFF12EH	TP0CCIC0	TP0CCIF0	TP0CCMK0	0	0	0	TP0CCPR2	TP0CCPR1	TP0CCPR0
FFFFF130H	TP0CCIC1	TP0CCIF1	TP0CCMK1	0	0	0	TP0CCPR12	TP0CCPR11	TP0CCPR10
FFFFF132H	TP1OVIC	TP1OVIF	TP1OVMK	0	0	0	TP1OVPR2	TP1OVPR1	TP1OVPR0
FFFFF134H	TP1CCIC0	TP1CCIF0	TP1CCMK0	0	0	0	TP1CCPR2	TP1CCPR1	TP1CCPR0
FFFFF136H	TP1CCIC1	TP1CCIF1	TP1CCMK1	0	0	0	TP1CCPR12	TP1CCPR11	TP1CCPR10
FFFFF138H	TP2OVIC	TP2OVIF	TP2OVMK	0	0	0	TP2OVPR2	TP2OVPR1	TP2OVPR0
FFFFF13AH	TP2CCIC0	TP2CCIF0	TP2CCMK0	0	0	0	TP2CCPR2	TP2CCPR1	TP2CCPR0
FFFFF13CH	TP2CCIC1	TP2CCIF1	TP2CCMK1	0	0	0	TP2CCPR12	TP2CCPR11	TP2CCPR10
FFFFF13EH	TP3OVIC	TP3OVIF	TP3OVMK	0	0	0	TP3OVPR2	TP3OVPR1	TP3OVPR0
FFFFF140H	TP3CCIC0	TP3CCIF0	TP3CCMK0	0	0	0	TP3CCPR2	TP3CCPR1	TP3CCPR0
FFFFF142H	TP3CCIC1	TP3CCIF1	TP3CCMK1	0	0	0	TP3CCPR12	TP3CCPR11	TP3CCPR10
FFFFF144H	TM0EQIC0	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR2	TM0EQPR1	TM0EQPR0
FFFFF146H	CB0RIC	CB0RIF	CB0RMK	0	0	0	CB0RPR2	CB0RPR1	CB0RPR0
FFFFF148H	CB0TIC	CB0TIF	CB0TMK	0	0	0	CB0TPR2	CB0TPR1	CB0TPR0
FFFFF14AH	CB1RIC	CB1RIF	CB1RMK	0	0	0	CB1RPR2	CB1RPR1	CB1RPR0
FFFFF14CH	CB1TIC	CB1TIF	CB1TMK	0	0	0	CB1TPR2	CB1TPR1	CB1TPR0
FFFFF14EH	UA0RIC	UA0RIF	UA0RMK	0	0	0	UA0RPR2	UA0RPR1	UA0RPR0
FFFFF150H	UA0TIC	UA0TIF	UA0TMK	0	0	0	UA0TPR2	UA0TPR1	UA0TPR0
FFFFF152H	UA1RIC	UA1RIF	UA1RMK	0	0	0	UA1RPR2	UA1RPR1	UA1RPR0
FFFFF154H	UA1TIC	UA1TIF	UA1TMK	0	0	0	UA1TPR2	UA1TPR1	UA1TPR0
FFFFF156H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF160H	KRIC	KRIF	KRMK	0	0	0	KRPR2	KRPR1	KRPR0
FFFFF162H	WTIIC	WTIIF	WTIMK	0	0	0	WTIPR2	WTIPR1	WTIPR0
FFFFF164H	WTIC	WTIF	WTMK	0	0	0	WTPR2	WTPR1	WTPR0

14.3.5 中断屏蔽寄存器 0 至 2（IMR0 至 IMR2）

IMR0 到 IMR2 寄存器为可屏蔽中断设定中断屏蔽状态。IMR0 到 IMR2 寄存器的 **xxMKn** 位相当于 **xxICn.xxMKn** 位。

IMRm 寄存器可以以 16 位为单位读取或是写入（m = 0 至 2）。

如果 IMRm 寄存器的高 8 位用作 IMRmH 寄存器，低 8 位用作 IMRmL 寄存器，这些寄存器可以以 8 位或 1 位为单位进行读取或是写入操作（m = 0 至 2）。

复位后，这些寄存器的值为 FFFFH。

注意事项 设备文件定义了 **xxICn.xxMKn** 位为保留字。如果有其它的位使用 **xxMKn** 的名称进行操作，**xxICn** 寄存器的内容会被写入，以代替 IMRm 寄存器（作为结果，IMRm 寄存器的内容也会被覆盖）。

复位后：FFFFH		R/W	地址：		IMR2 FFFFF104H, IMR2L FFFFF104H, IMR2H FFFFF105H			
	15	14	13	12	11	10	9	8
IMR2 (IMR2H ^注)	1	1	1	1	1	WTMK	WTIMK	KRMK
	7	6	5	4	3	2	1	0
IMR2L	1	1	1	1	ADMK	UA1TMK	UA1RMK	UA0TMK
复位后：FFFFH		R/W	地址：		IMR1 FFFFF102H, IMR1L FFFFF102H, IMR1H FFFFF103H			
	15	14	13	12	11	10	9	8
IMR1 (IMR1H ^注)	UA0RMK	CB1TMK	CB1RMK	CB0TMK	CB0RMK	TM0EQMK0	TP3CCMK1	TP3CCMK0
	7	6	5	4	3	2	1	0
IMR1L	TP3OVMK	TP2CCMK1	TP2CCMK0	TP2OVMK	TP1CCMK1	TP1CCMK0	TP1OVMK	TP0CCMK1
复位后：FFFFH		R/W	地址：		IMR0 FFFFF100H, IMR0L FFFFF100H, IMR0H FFFFF101H			
	15	14	13	12	11	10	9	8
IMR0 (IMR0H ^注)	TP0CCMK0	TP0OVMK	TQ0CCMK3	TQ0CCMK2	TQ0CCMK1	TQ0CCMK0	TQ0OVMK	PMK7
	7	6	5	4	3	2	1	0
IMR0L	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

xxMKn	中断屏蔽标志的设置
0	允许中断服务
1	禁止中断服务

注 为了以 8 位或 1 位为单位读取 IMR0 到 IMR2 寄存器的位 8 到位 15，需要将它们设定为 IMR0H 到 IMR2H 寄存器的位 0 到位 7。

注意事项 将 IMR2 寄存器的位 15 到位 11 以及位 7 至位 4 置“1”。如果这些位的设定值改变，操作将不会得到保证。

备注 **xx**: 表示任意的外围单元的名称（参考表 14-2 中断控制寄存器（xxICn））。
n: 外围单元编号（参考表 14-2 中断控制寄存器（xxICn））。

14.3.6 正在进行服务优先级寄存器 (ISPR)

ISPR 寄存器保持当前被确认的可屏蔽中断的优先级别。当一个中断请求被确认时，中断请求信号的优先级别所对应该寄存器位被设为 1，且在中断服务程序执行期间保持该设定。

当执行 RETI 指令后，具有最高优先级的中断请求信号所对应的位通过硬件自动复位为 0。然而，当操作从非屏蔽中断服务或是异常处理中返回时，该位不会复位为 0。

这是一个只读寄存器，可以以 8 位或是 1 位为单位进行访问。

复位后该寄存器的值为 00H。

注意事项 如果在中断允许状态 (EI) 对 ISPR 寄存器进行读取时确认一个中断，在寄存器的位通过确认中断的方式被设置之后，可以读取 ISPR 寄存器的值。为了在确认中断之前正确的读取 ISPR 寄存器的值，则需在禁止中断 (DI) 时读取寄存器。

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	确定当前中断优先级
0	未确认带有优先级n的中断请求信号
1	已确认带有优先级n的中断请求信号

备注 n = 0 至 7 (优先级)

14.4 软件异常

当 CPU 执行 TRAP 指令的时候会产生一个软件的异常，且总是会被确认。

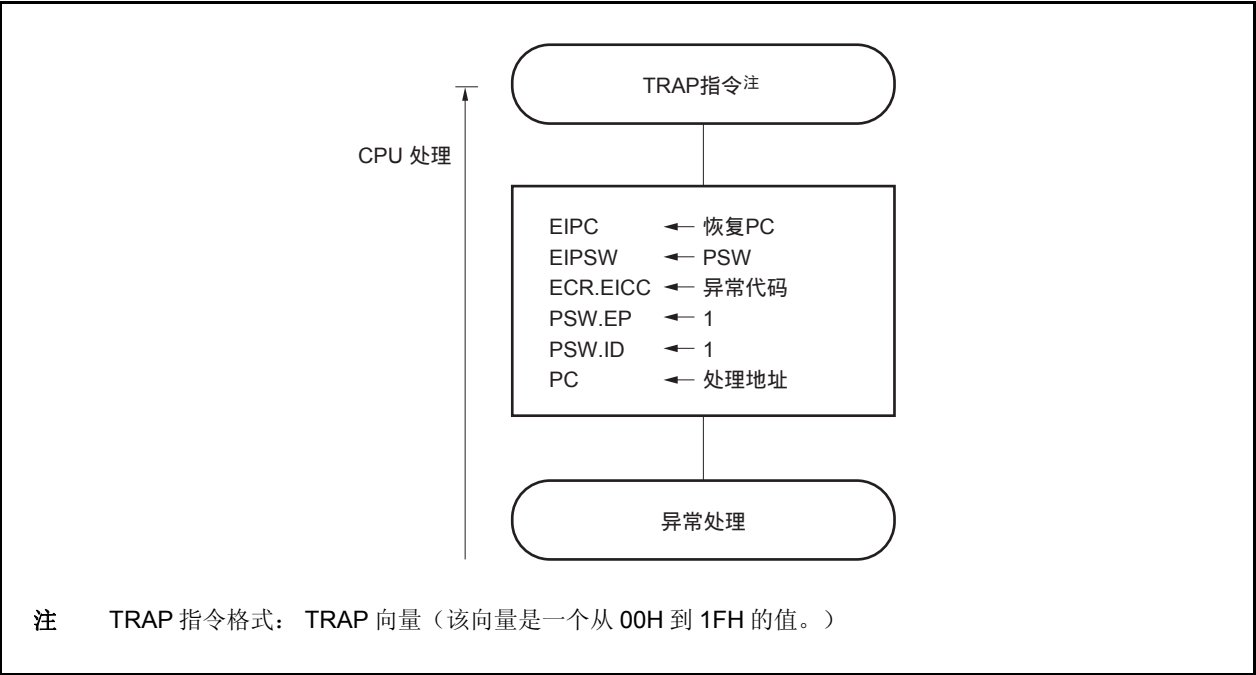
14.4.1 操作

如果一个软件异常产生，CPU 将执行下列处理过程，并且将操作权转移给处理例程。

- <1> 将恢复的 PC 保存在 EIPC 中。
- <2> 将当前的 PSW 保存在 EIPSW 中。
- <3> 在 ECR（中断源）的低 16 位（EICC）中写入一个异常代码。
- <4> 将 PSW.EP 位和 PSW.ID 位置 1。
- <5> 在 PC 中设置软件异常中相应的处理地址（00000040H 或 00000050H），并且转移控制权。

图 14-9 举例说明软件异常的处理过程。

图 14-9. 软件异常处理过程



处理地址是通过 TRAP 指令的操作数（向量）决定的。如果该向量是 00H 到 0FH，它将变为 00000040H；而如果向量是 10H 到 1FH，它将变为 00000050H。

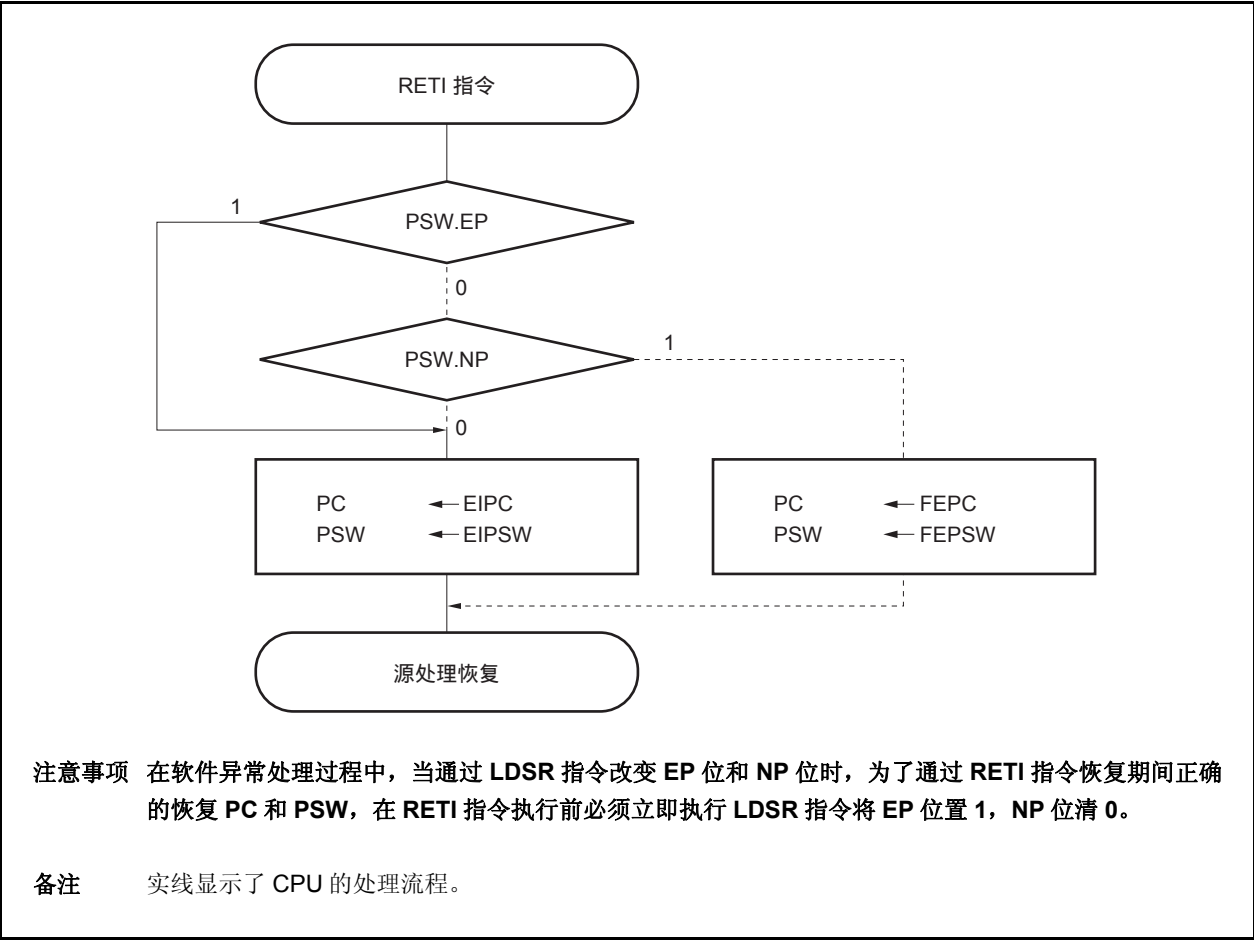
14.4.2 恢复

通过执行 RETI 指令可以从软件异常进程中恢复。
通过执行 RETI 指令，CPU 执行下列处理过程，并且将控制权转移到恢复 PC 的地址。

- <1> 由于 PSW.EP 位为 1，所以从 EIPC 和 EIPSW 中加载恢复的 PC 和 PSW。
- <2> 将控制权转移到恢复 PC 和 PSW 的地址。

图 14-10 举例说明了 RETI 指令的处理过程。

图 14-10. RETI 指令处理过程



14.4.3 EP 标志

EP 标志用于指出异常进程正在进行中。在异常发生时，该位将被设定。

复位后：00000020H

PSW

31876543210

0NP EP ID SAT CY OV S Z

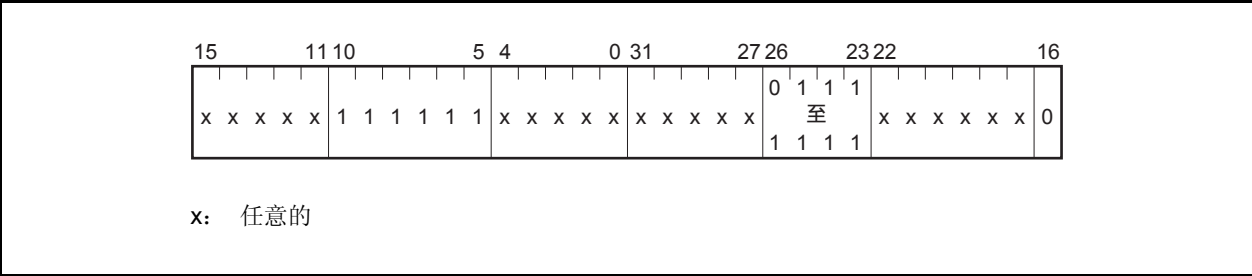
EP	异常处理状态
0	未执行异常处理。
1	正在执行异常处理

14.5 异常陷阱

异常陷阱是一个中断，其在指令的非法操作发生时需要使用。在 V850ES/HE2，一个非法的操作代码异常（ILGOP：非法的操作代码陷阱）被认为是一个异常陷阱。

14.5.1 非法操作码 定义

非法指令位 10 到 5 的操作代码是 111111B，位 26 到 23 的子操作代码是 0111B 到 1111B，位 16 的子操作代码是 0B。当指令应用到这个非法指令执行后，产生一个异常陷阱。



注意事项 由于将来有可能分配该指令到一个非法的操作代码，推荐不要使用它。

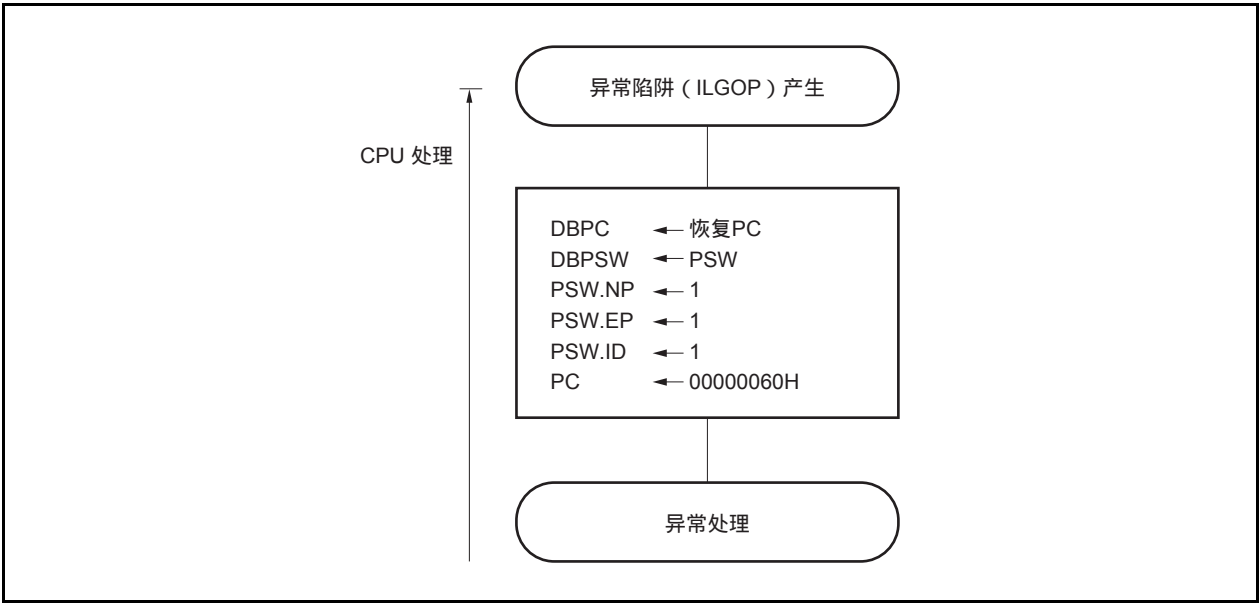
(1) 操作

如果产生一个异常陷阱，CPU 执行下列处理过程，并且将操作权转移给处理例程。

- <1> 将恢复的 PC 保存到 DBPC。
- <2> 将当前的 PSW 保存到 DBPSW。
- <3> 将 PSW.NP 位，PSW.EP 位，和 PSW.ID 位置 1。
- <4> 在 PC 中设置相应异常陷阱的处理地址（00000060H），并且转移控制权。

图 14-11 举例说明异常陷阱的处理过程。

图 14-11. 异常陷阱的处理过程



(2) 恢复

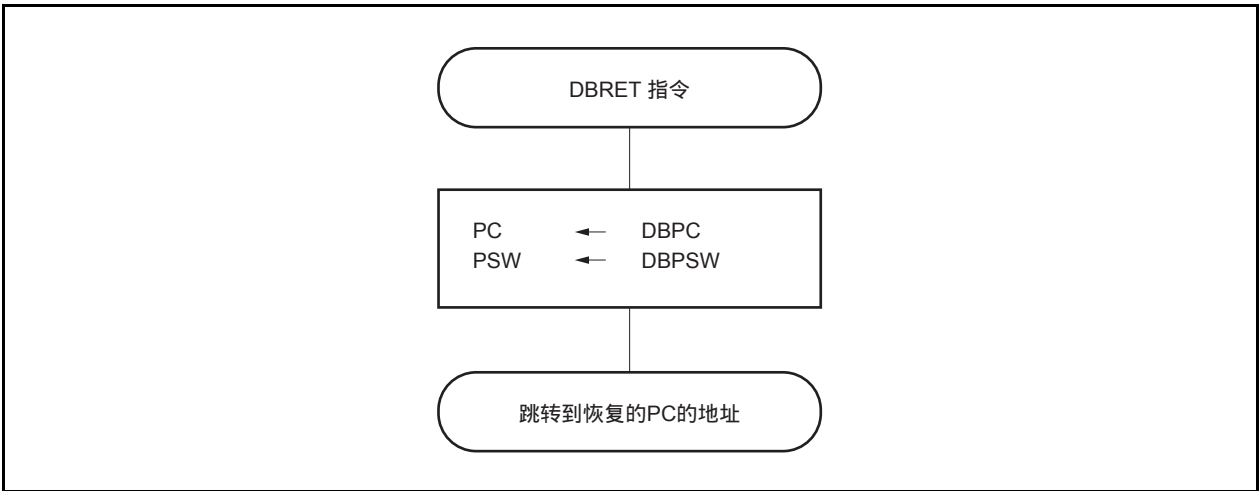
通过执行 DBRET 指令实现从异常陷阱中恢复。通过执行 DBRET 指令，CPU 执行下列处理过程并且控制着恢复 PC 的地址。

- <1> 从 DBPC 和 DBPSW 中加载恢复的 PC 和 PSW。
- <2> 向恢复的 PC 和 PSW 代表的地址转移控制权。

注意事项 只能在执行非法操作码以及 DBRET 指令的间隔时间内访问 DBPC 和 DBPSW。

图 14-12 举例说明从异常陷阱中恢复的处理过程。

图 14-12. 从异常陷阱中恢复的处理过程



14.5.2 调试陷阱

调试陷阱是一个异常，它在 DBTRAP 指令执行时产生，且总是会被确认。

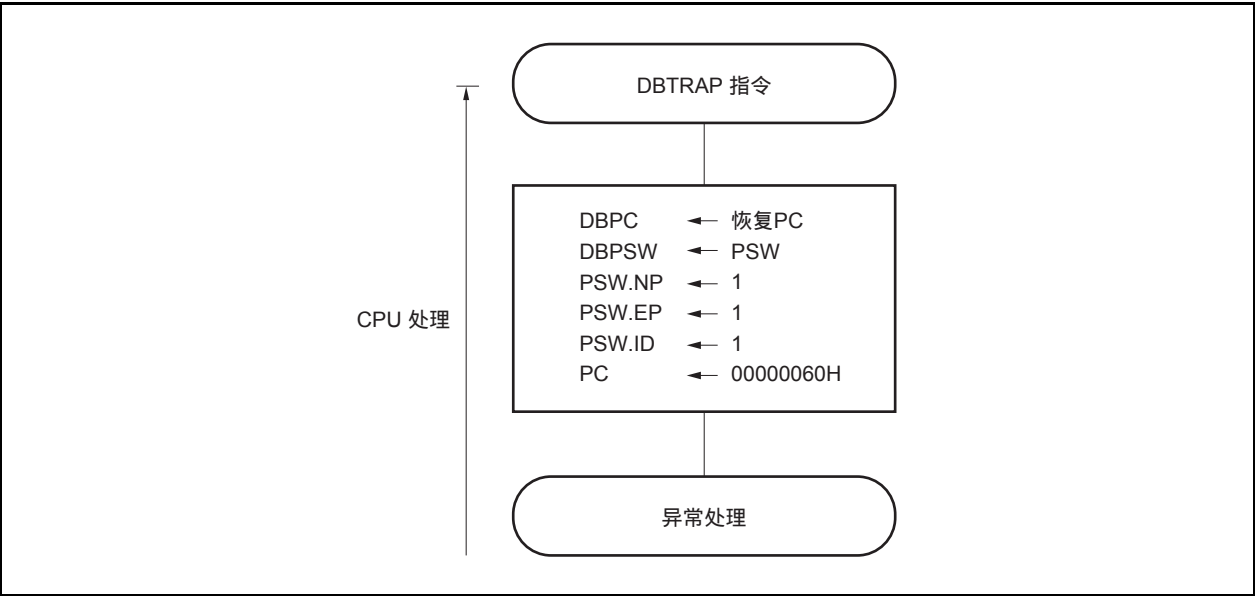
(1) 操作

在出现调试陷阱之后，CPU 将执行如下处理过程。

- <1> 将恢复的 PC 保存到 DBPC。
- <2> 将当前的 PSW 保存到 DBPSW。
- <3> 将 PSW.NP 位，PSW.EP 位，和 PSW.ID 位置 1。
- <4> 在 PC 中设置相应调试陷阱的处理地址（00000060H），并且转移控制权。

图 14-13 显示了调试陷阱的处理格式。

图 14-13. 调试陷阱的处理格式



(2) 恢复

通过执行 DBRET 指令可以从调试陷阱中恢复。

执行 DBRET 指令，CPU 将执行以下步骤，并将控制权转移到恢复 PC 的地址。

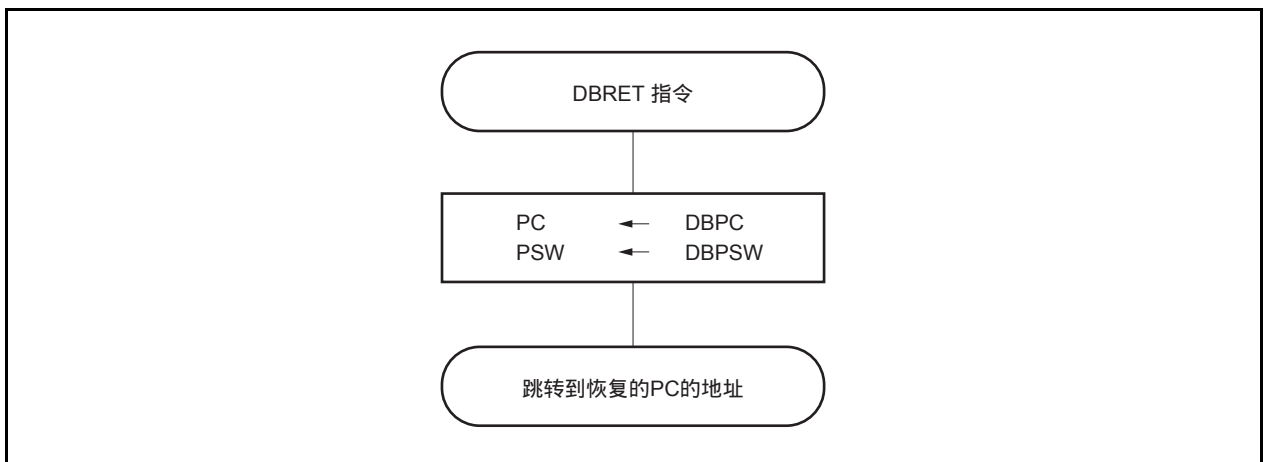
<1> 从 DBPC 和 DBPSW 中加载恢复的 PC 和 PSW。

<2> 向恢复的 PC 和 PSW 代表的地址转移控制权。

注意事项 只能在执行 DBTRAP 指令以及 DBRET 指令间的间隔时间内访问 DBPC 和 DBPSW。

图 14-14 显示了从调试陷阱中恢复的处理格式。

图 14-14. 从调试陷阱中恢复的处理格式



14.6 外部中断请求输入引脚（NMI 以及 INTP0 至 INTP7）

14.6.1 噪声消除

（1）NMI 引脚的噪声消除

NMI 引脚拥有内部噪声消除电路，它使用模拟延迟。因此，NMI 引脚的输入电平不会作为边沿被检测，除非保持一定的时间或者更长。所以只有在特定的时间才会检测边沿。

NMI 引脚可以用来释放 STOP 模式。在 STOP 模式中，使用系统时钟不会消除噪声，因为内部系统时钟是停止的。

（2）INTP0 到 INTP7 引脚的噪声消除

INTP0 到 INTP7 引脚拥有内部噪声消除电路，它使用模拟延迟。因此，这些引脚的输入电平不会作为边沿被检测，除非保持一定的时间或者更长。所以只有在特定的时间才会检测边沿。

14.6.2 边沿检测

每一个 NMI 和 INTP0 到 INTP7 引脚的有效边沿可以在以下四种方式中选择。

- 上升沿
- 下降沿
- 双边沿
- 不检测边沿

在复位后，NMI 引脚不会检测边沿。因此，中断请求信号不会被确认，除非使用 INTF0 和 INTR0 寄存器允许有效边沿（NMI 引脚功能作为一个端口引脚）。

(1) 外部中断下降，上升沿指定寄存器 0 (INTF0, INTR0)

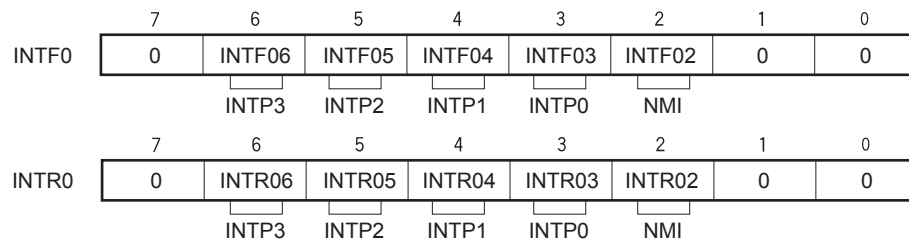
INTF0 和 INTR0 寄存器是 8 位寄存器，它们通过位 2 指定了 NMI 引脚的上升沿和下降沿的检测，通过位 3 到 6 指定了外部中断引脚 (INTP0 至 INTP3) 的上升沿和下降沿的检测。

这些寄存器可以以 8 位或是 1 位为单位进行读取或是写入。

复位后这些寄存器的值为 00H。

注意事项 当引脚的功能从外部中断功能（复用功能）转变到端口功能时，可以进行边沿检测。因此，首先要将 INTF0n 位和 INTR0n 位设定为 00，然后再设定端口模式。

复位后： 00H R/W 地址： INTF0 FFFFC00H, INTR0 FFFFC20H



备注 关于有效边沿设定组合，请参考表 14-3。

表 14-3. 有效边沿设定

INTF0n	INTR0n	有效边沿设定 (n = 2 至 6)
0	0	没有检测到边沿
0	1	上升沿
1	0	下降沿
1	1	双边沿

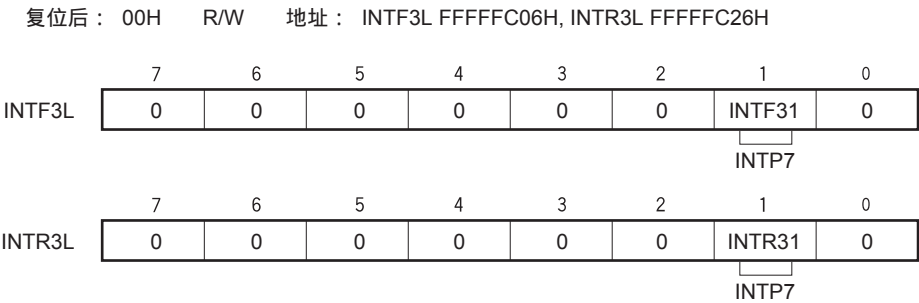
注意事项 如果相应的引脚没有被用作 NMI 或 INTP0 到 INTP3 引脚，那么一定要将 INTF0n 和 INTR0n 位设定为 00。

备注 n = 2: 控制 NMI 引脚
 n = 3 至 6: 控制 INTP0 到 INTP3 引脚

(2) 外部中断上升，下降沿指定寄存器 3L (INTR3L, INTF3L)

INTR3 和 INTF3 寄存器是 8 位寄存器，它指定了外部中断引脚 (INTP7) 的上升沿和下降沿的检测。
这些寄存器可以以 8 位或是 1 位为单位进行读取或是写入。
复位后这些寄存器的值为 00H。

注意事项 当引脚的功能从外部中断功能（复用功能）转变到端口功能时，可以进行边沿检测。因此，首先要将 INTF0n 位和 INTR0n 位设定为 00，然后再设定端口模式。



备注 关于有效边沿设定组合，请参考表 14-4。

表 14-4. 有效边沿设定

INTF31	INTR31	有效边沿设定
0	0	没有检测到边沿
0	1	上升沿
1	0	下降沿
1	1	双边沿

注意事项 如果相应的引脚没有被用作 INTP7 引脚，那么一定要将 INTF31 和 INTR31 位设定为 00。

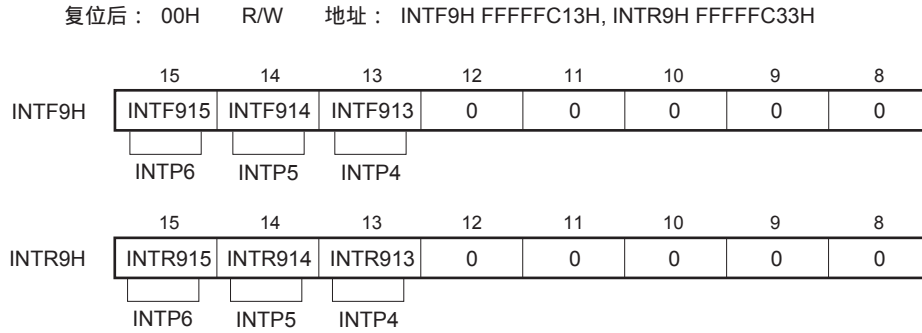
(3) 外部中断下降，上升沿指定寄存器 9H (INTF9H, INTR9H)

INTF9H 和 INTR9H 寄存器是 8 位寄存器，它们指定了外部中断引脚 (INTP4 到 INTP6) 的上升沿和下降沿的检测。

这些寄存器可以以 8 位或是 1 位为单元进行读取或是写。

复位后这些寄存器的值为 00H。

注意事项 当引脚的功能从外部中断功能（复用功能）转变到端口功能时，可以进行边沿检测。因此，首先要将 **INTF9n 位和 INTR9n 位设定为 0**，然后再设定端口模式。



备注 关于有效边沿设定组合，请参考表 14-5。

表 14-5. 有效边沿设定

INTF9n	INTR9n	有效边沿设定 (n = 13 至 15)
0	0	没有检测到边沿
0	1	上升沿
1	0	下降沿
1	1	双边沿

注意事项 如果相应的引脚没有被用作 INTP4 至 INTP6 引脚，则一定要将 INTF9n 和 INTR9n 位设定为 00。

备注 n = 13 至 15: 控制 INTP4 到 INTP6 引脚

(4) 噪声消除控制寄存器 (NFC)

可以为 INTP3 引脚选择数字噪声消除。噪声消除设定通过使用 NFC 寄存器执行。

当选择了数字噪声消除，数字采样的时钟可以从 $f_{xx}/64$ ， $f_{xx}/128$ ， $f_{xx}/256$ ， $f_{xx}/512$ ， $f_{xx}/1,024$ 以及 f_{xt} 中选择。采样计数是通过 NFC.NFSTS 位来设置的。

采样频率是时钟的三倍。

当选择了数字噪声消除时，如果在备用模式中用作采样的时钟被停止，那么 INTP3 中断请求信号将不能用于释放备用模式。当 f_{xt} 用作采样时钟时，INTP3 中断请求信号可以用于副时钟操作模式或 IDLE1/IDLE2/STOP/sub-IDLE 模式。

这些寄存器可以以 8 位为单元进行读取或是写入。

复位后这些寄存器的值为 00H。

注意事项 采样时钟被改变后，一直到数字噪声消除器被初始化，之间将会需要与 NFSTS 位所设置的采样次数和采样时钟的乘积相同的时间。因此，在 NFSTS 位所设置的采样次数和采样时钟的乘积相同的时间过去前，如果 INTP3 的有效边沿在采样时钟改变后被输入，那么可能会生成中断请求信号。因此使用中断功能时应注意以下几点。

- 使用中断功能时，在 NFSTS 位所设置的采样次数和采样时钟的乘积相同的时间过去后，将会在中断请求标志 (PIC3.PIF3 位) 清除后允许中断。

复位后：00H R/W 地址：FFFF318H

	7	6	5	4	3	2	1	0
NFC	NFEN	NFSTS	0	0	0	NFC2	NFC1	NFC0

NFEN	INTP3引脚噪声消除的设置
0	模拟噪声消除
1	数字噪声消除

NFSTS	数字噪声消除采样次数的设置
0	采样次数 × 3次
1	采样次数 × 2次

NFC2	NFC1	NFC0	数字采样时钟
0	0	0	$f_{xx}/64$
0	0	1	$f_{xx}/128$
0	1	0	$f_{xx}/256$
0	1	1	$f_{xx}/512$
1	0	0	$f_{xx}/1,024$
1	0	1	f_{XT} (副时钟)
其他情况			禁止设置

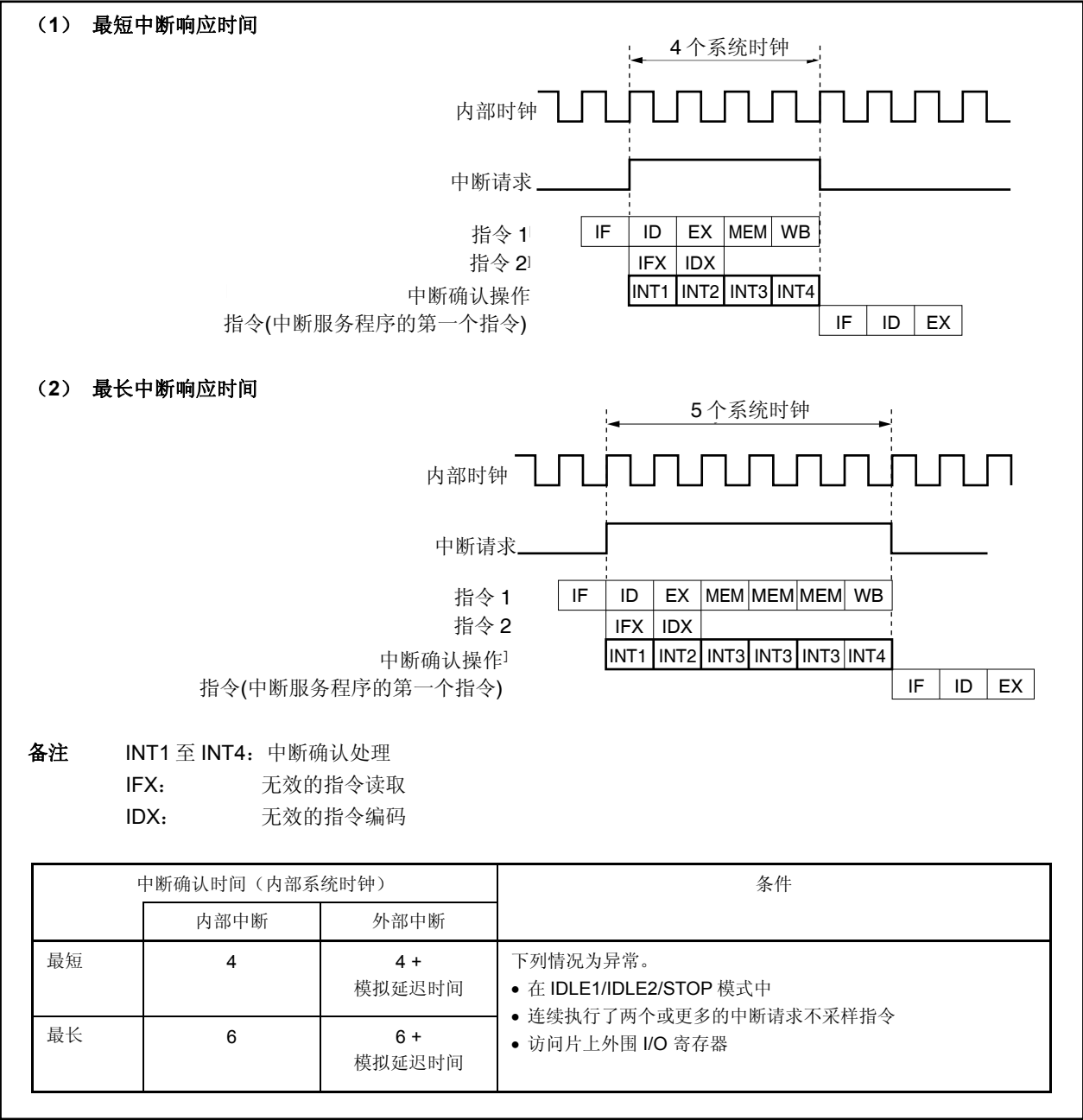
- 备注**
1. 由于执行了 3 次采样，可靠的噪声消除宽度是 2 个采样时钟。
 2. 在噪声的宽度小于 2 个采样时钟的情况下，如果有与采样时钟同步的噪声输入将会产生中断请求信号。

14.7 CPU 的中断确认时间

除非出现下列情况，CPU 的中断确认时间最少需要 4 个时钟周期。为了成功的输入中断请求信号，中断输入的间隔最少应该间隔 5 个时钟。

- 在 IDLE1/IDLE2/STOP 模式中
- 当成功的执行了中断请求不采样指令（参考 14.8 CPU 不对中断进行确认的时期。）
- 当中断控制寄存器被访问时

图 14-15. 在中断请求信号确认过程中的传递操作（概要）



14.8 CPU 不对中断进行确认的时期

当执行指令时，中断会通过 CPU 确认。但是，在两个中断请求不采样指令之间（中断被保持）不会确认任何中断。

中断不采样信号指令如下。

- EI 指令
- DI 指令
- LDSR reg2, 0x5 指令（PSW 的）
- PRCMD 寄存器的存储指令
- 以下寄存器的存储，SET1，NOT1，或是 CLR1 指令。
 - 中断相关寄存器：
 - 中断控制寄存器（xxICn），中断屏蔽寄存器 0 至 2（IMR0 至 IMR2）
 - 节能控制寄存器（PSC）
 - 片上调试模式寄存器（OCDM）
 - 外围仿真寄存器 1（PEMU1）：

备注 xx: 表示任意的单元的名称（参考表 14-2 中断控制寄存器（xxICn））
 n: 外围单元编号（参考表 14-2 中断控制寄存器（xxICn））。

14.9 注意事项

NMI 引脚的复用功能与 P02 引脚相同。复位后它的功能相当于一个正常的端口。为了允许 NMI 引脚，需用通过 PMC0 寄存器使 NMI 引脚有效。NMI 引脚的初始设定是“没有检测到边沿”。使用 INTF0 寄存器和 INTR0 寄存器设定 NMI 引脚的有效边沿。

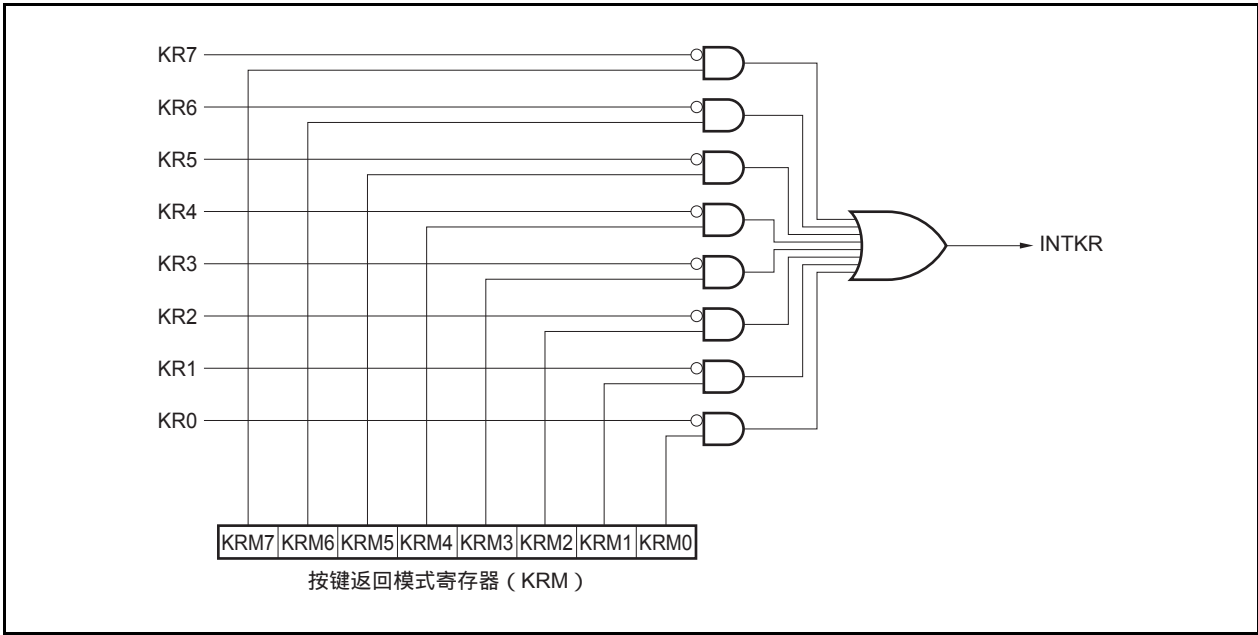
15.1 功能

通过设置 KRM 寄存器，给八个按键输入引脚（KR0 至 KR7）输入下降沿，可以产生按键中断请求信号（INTKR）。

表 15-1. 按键返回检测引脚的任务

标志	引脚描述
KRM0	以 1 位为单位控制 KR0 信号
KRM1	以 1 位为单位控制 KR1 信号
KRM2	以 1 位为单位控制 KR2 信号
KRM3	以 1 位为单位控制 KR3 信号
KRM4	以 1 位为单位控制 KR4 信号
KRM5	以 1 位为单位控制 KR5 信号
KRM6	以 1 位为单位控制 KR6 信号
KRM7	以 1 位为单位控制 KR7 信号

图 15-1. 按键返回框图



15.2 寄存器

(1) 按键返回模式寄存器 (KRM)

KRM 寄存器使用 KR0 至 KR7 信号控制 KRM0 至 KRM7 位。

该寄存器可用 8 位或 1 位指令进行读或写。

复位输入将该寄存器清零 (00H)。

复位后：00H R/W 地址：FFFFFF300H

	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	按键返回模式的控制
0	不检测按键返回信号
1	检测按键返回信号

注意事项 在 KRM 寄存器被清零 (00H) 后，重写 KRM 寄存器。

备注 关于复用功能引脚设置，参见表 4-12 使用端口引脚作为复用功能引脚。

15.3 注意事项

- (1) 如果给 KR0 至 KR7 中的任何一个输入低电平，即使其它引脚输入下降沿，也不会产生 INTKR 信号。
- (2) 不能同时使用 RXDA1 和 KR7 引脚。如果使用 RXDA1 引脚，就不能使用 KR7 引脚。如果使用 KR7 引脚，就不能使用 RXDA1 引脚（推荐设置 PFC91 位为 1 并且将 PFCE91 位清零）。
- (3) 如果改变 KRM 寄存器，则会产生中断请求信号 (INTKR)。为避免这种情况，在禁止中断 (DI) 或屏蔽中断之后改变 KRM 寄存器，然后将中断请求标志 (KRIC、KRIF 位) 清零，再允许中断或清除屏蔽。
- (4) 为了使用按键中断功能，确定设置端口引脚为按键返回引脚，然后使用 KRM 寄存器允许操作。为了从按键返回引脚切换到端口引脚，使用 KRM 寄存器禁止操作，然后设置为端口引脚。

第 16 章 待机功能

16.1 概述

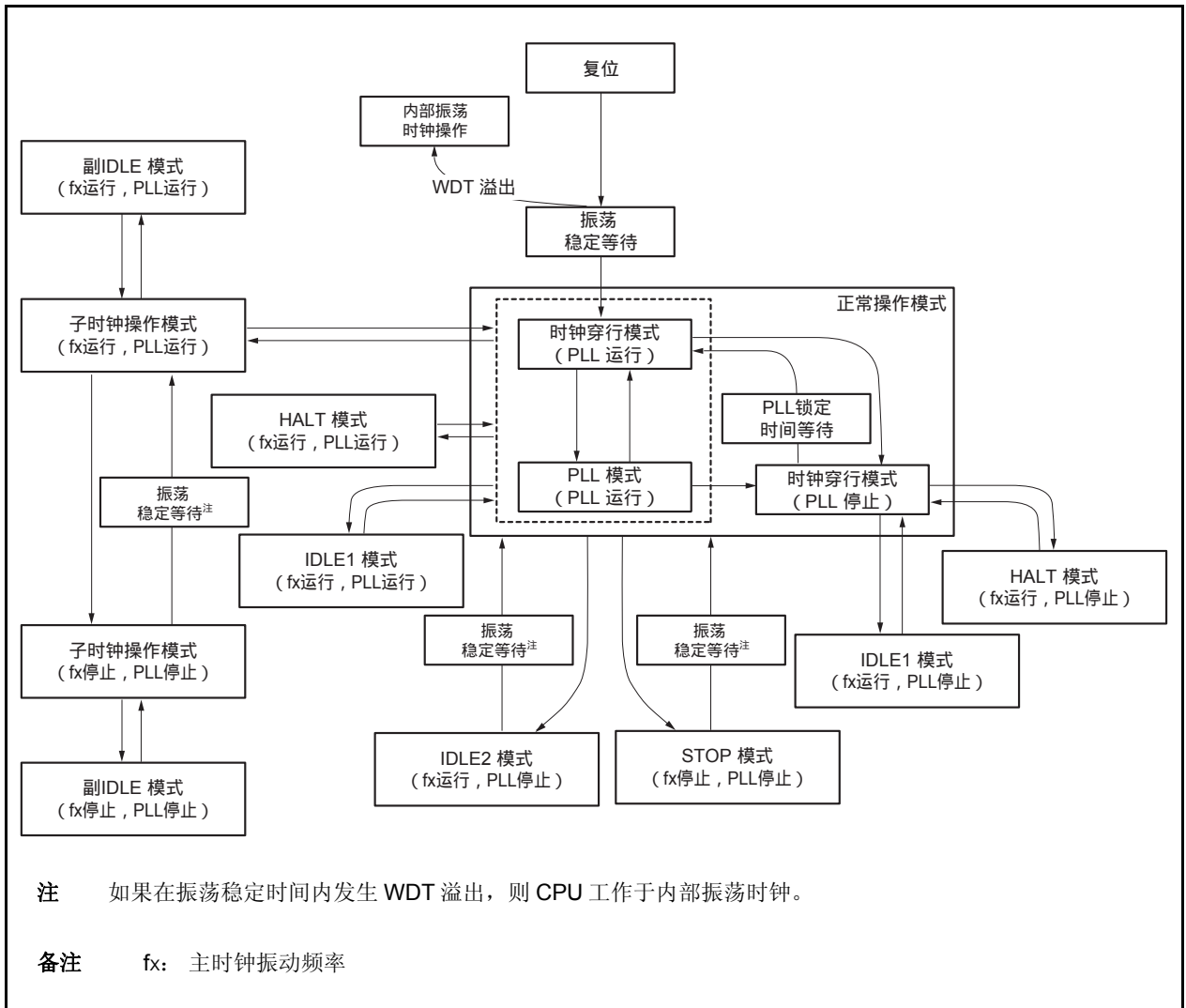
通过结合使用待机模式并选择合适的应用模式，可以有效的降低系统功耗。可用待机模式列于表 16-1。

表 16-1. 待机模式

模式	功能概要
HALT 模式	只有 CPU 的操作时钟被停止的模式。
IDLE1 模式	除了振荡器，PLL [※] 以及内存外，内部电路的其他所有操作都被停止的模式。
IDLE2 模式	除了振荡器外，芯片的其他所有内部操作都被停止的模式。
STOP 模式	除了副时钟振荡器外，芯片的其他所有内部操作都被停止的模式。
副时钟操作模式	副时钟被用作内部系统时钟的模式
副 IDLE 模式	在副时钟操作模式中，除了振荡器外，芯片的其他所有内部操作都被停止的模式。

注 **PLL** 保持预先操作状态。

图 16-1. 状态转换



16.2 寄存器

(1) 节能控制寄存器 (PSC)

PSC 寄存器是控制待机功能的 8 位寄存器。此寄存器的 STP 位被用于指定 STOP 模式。此寄存器是仅能使用特殊顺序组合写入的特殊寄存器（参见 3.4.7 特殊寄存器）。

该寄存器可用 8 位或 1 位指令进行读或写。

复位输入将该寄存器清零（00H）。

复位后： 00H R/W 地址： FFFFF1FEH

	7	6	5	4	3	2	1	0
PSC	0	NMI1M	NMI0M	INTM	0	0	STP	0

NMI1M	通过产生INTWDT2信号释放待机模式
0	允许通过INTWDT2信号释放待机模式
1	禁止通过INTWDT2信号释放待机模式

NMI0M	通过NMI引脚输入控制待机模式的释放
0	允许通过NMI引脚输入释放待机模式
1	禁止通过NMI引脚输入释放待机模式

INTM	通过可屏蔽中断请求信号控制待机模式的释放
0	允许通过可屏蔽中断请求信号释放待机模式
1	禁止通过可屏蔽中断请求信号释放待机模式

STP	待机模式 ^注 设置
0	正常模式
1	待机模式

注 通过 STP 位设置待机模式：IDLE1、IDLE2、STOP 或副 IDLE 模式。

- 注意事项
1. 在设置 IDLE1，IDLE2，STOP 或副 IDLE 模式之前，先设置 PSMR.PSM1 和 PSMR.PSM0 位，然后设置 STP 位。
 2. 当释放 HALT 模式时对 NMI1M，NMI0M 和 INTM 位进行设置是非法的。
 3. 如果 STP 位设为 1 的同时 NMI1M，NMI0M 或 INTM 位也设为 1，那么 NMI1M，NMI0M 或 INTM 位的设置无效。当设置为 IDLE1/IDLE2/STOP 模式时，如果一个未屏蔽的中断请求信号保持为未决定的状态，则将与中断请求信号（NMI1M，NMI0M 或 INTM）相应的位设为 1，然后将 STP 位设为 1。

(2) 功率节省模式寄存器 (PSMR)

PSMR 寄存器是一个 8 位寄存器，其控制省电模式中的操作状态和时钟操作。
该寄存器可用 8 位或 1 位指令进行读或写。
复位输入将该寄存器清零 (00H)。

复位后： 00H R/W 地址： FFFF820H

	7	6	5	4	3	2	1	0
PSMR	0	0	0	0	0	0	PSM1	PSM0

PSM1	PSM0	软件待机模式中的操作说明
0	0	IDLE1
0	1	STOP 模式
1	0	IDLE2，副IDLE模式
1	1	STOP 模式

- 注意事项 1. 确定将第 2 位 至 第 7 位清除为 “0”。
2. 仅当 PSC.STP 位为 1 时，PSM0 和 PSM1 位是有效的。

备注 IDLE1: 在此模式中，除了振荡器之外的所有操作和一些其它的电路（Flash 存储器和 PLL）都被停止。
在 IDLE1 模式被释放后，重新恢复为正常操作模式而无须确保振荡稳定时间，就像 HALT 模式一样。

IDLE2: 在此模式中，除了振荡器操作之外的所有操作都被停止。
在 IDLE2 模式被释放后，等待通过 OSTS 寄存器指定的设置时间后恢复为正常操作模式（Flash 存储器和 PLL）。

STOP: 在此模式中，除了副时钟振荡器操作之外的所有操作都被停止。
在 STOP 模式被释放后，等待通过 OSTS 寄存器指定的振荡稳定时间后恢复为正常操作模式。

副 IDLE: 在此模式中，除了振荡器操作之外的所有其它操作都被暂停。在通过中断请求信号释放 IDLE 模式之后，在等待 12 个副时钟周期之后重新恢复副时钟操作模式。

(3) 振荡稳定时间选择寄存器 (OSTS)

通过 OSTS 寄存器控制 STOP 模式释放后到振荡稳定的等待时间或 IDLE2 模式释放后到内置 flash 存储器稳定的等待时间。

OSTS 寄存器可用 8 位指令进行读或写。

复位输入设置该寄存器为 06H。

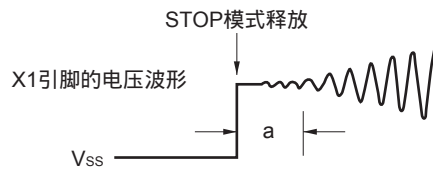
复位后：06H R/W 地址：FFFFF6C0H

	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/设置时间 ^注 的选择	f _x	
				4 MHz	5 MHz
0	0	0	$2^{10}/f_x$	0.256 ms	0.205 ms
0	0	1	$2^{11}/f_x$	0.512 ms	0.410 ms
0	1	0	$2^{12}/f_x$	1.024 ms	0.819 ms
0	1	1	$2^{13}/f_x$	2.048 ms	1.638 ms
1	0	0	$2^{14}/f_x$	4.096 ms	3.277 ms
1	0	1	$2^{15}/f_x$	8.192 ms	6.554 ms
1	1	0	$2^{16}/f_x$	16.38 ms	13.107 ms
1	1	1	禁止设置		

注 当 STOP 模式和 IDLE2 模式被释放后分别要求振荡稳定时间和设置时间。

注意事项 1. STOP 模式释放后的等待时间不包括从释放 STOP 模式到时钟振荡启动这段时间（即下图“a”所示的部分），不管通过复位输入或中断请求信号的产生是否释放了 STOP 模式。



2. 确定将第 3 位至第 7 位清除为“0”。

3. 复位释放后的振荡稳定时间是 $2^{16}/f_x$ （因为 OSTS 寄存器的初始值= 06H）。

备注 f_x = 主时钟振动频率

16.3 HALT 模式

16.3.1 设置和操作状态

当在正常操作模式中执行专用指令（HALT）时，HALT 模式被设置。

在 HALT 模式中，时钟振荡器继续操作。仅仅停止对 CPU 的时钟供应，继续提供给其它内置外围设备功能的时钟供应。

结果，程序停止执行，并且内部 RAM 保持 HALT 模式设置之前的内容。独立于 CPU 指令处理的内置外围设备功能可继续操作。

表 16-3 显示了 HALT 模式中的操作状态。

通过使用 HALT 模式并结合正常操作模式的间歇操作可降低系统的平均电流消耗。

- 注意事项**
1. 在 HALT 指令之后插入五个或更多的 NOP 指令。
 2. 如果在执行 HALT 指令时有一个未屏蔽的中断请求被保持等待响应，则此状态被转移到 HALT 模式中，然后通过这个等待响应的中断请求，HALT 模式被立即释放。

16.3.2 释放 HALT 模式

可通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号），未屏蔽的外部中断请求信号（INTP0 至 INTP7 引脚输入），来自于在 HALT 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，WDT2RES 信号，低电压检测器（LVI）或时钟监视器（CLM）产生的复位）来释放 HALT 模式。在 HALT 模式释放后，恢复正常操作模式。

(1) 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 HALT 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 HALT 模式，而不管中断请求信号的优先级。如果在某个中断服务程序中设置 HALT 模式，则之后产生的中断请求信号按如下所述来执行。

- (a) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 HALT 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- (b) 如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 HALT 模式，并且响应此中断请求信号。

表 16-2. 通过中断请求信号释放 HALT 模式后的操作

释放源	中断允许（EI）状态	中断禁止（DI）状态
不可屏蔽中断请求信号	执行对处理地址的跳转。	
可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令。	执行下一条指令。

(2) 通过复位释放 HALT 模式

执行与正常复位操作相同的操作。

表 16-3. HALT 模式中的操作状态

项目 \ HALT 模式设置		操作状态	
		当不使用副时钟时	当使用副时钟时
主时钟振荡器		允许振荡	
副时钟振荡器		—	允许振荡
内部振荡器		允许振荡	
PLL		可操作	
CPU		停止操作	
中断控制器		可操作	
定时器 P (TMP0 至 TMP3)		可操作	
定时器 Q (TMQ0)		可操作	
定时器 M (TMM0)		当不同于 f_{XT} 时钟被选择作为计数时钟时可操作	可操作
钟表定时器		当 f_x (BRG 分频) 时钟被选择作为计数时钟时可操作	可操作
看门狗定时器 2		可操作	
串行接口	CSIB0, CSIB1	可操作	
	UARTA0, UARTA1	可操作	
A/D 转换器		可操作	
按键中断功能 (KR)		可操作	
端口功能		在 HALT 模式被设置之前保持状态	
内部数据		CPU 寄存器, 状态, 数据和全部其它的内部数据, 例如内部 RAM 的内容, 被保持为 HALT 模式设置之前的状态。	

16.4 IDLE1 模式

16.4.1 设置和操作状态

通过在正常操作模式中对 PSMR.PSM1 和 PSMR.PSM0 位清零，置 PSC.STP 位为 1 来设置 IDLE1 模式。

在 IDLE1 模式中，时钟振荡器，PLL 和 flash 存储器继续操作但停止给 CPU 和其它内置外围设备功能的时钟供应。

结果，停止程序执行，并且保留在设置 IDLE1 模式之前的内部 RAM 的内容。CPU 和其它内置外围设备功能停止操作。但是，使用副时钟或外部时钟的内置外围设备功能继续操作。

表 16-5 显示了在 IDLE1 模式中的操作状态。

因为 IDLE1 模式停止了内置外围设备功能的操作，所以 IDLE1 模式比 HALT 模式更能降低功耗。因为不停止主时钟振荡器，所以当用与释放 HALT 模式相同的方法释放 IDLE1 模式时，无须等待振荡稳定时间就能恢复到正常操作模式。

- 注意事项**
1. 在执行往 PSC 寄存器中存入数据来设置 IDLE1 模式的指令之后，插入五个或更多的 NOP 指令。
 2. 如果设置 IDLE1 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则 IDLE1 模式立即被此等待响应的中断请求释放。

16.4.2 释放 IDLE1 模式

可通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号），未屏蔽的外部中断请求信号（INTP0 至 INTP7 引脚输入），来自于在 IDLE1 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，WDT2RES 信号，上电清零电路（POC），低电压检测器（LVI）或时钟监视器（CLM）产生的复位）来释放 IDLE1 模式。

在 IDLE1 模式释放后，恢复正常操作模式。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE1 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE1 模式，而不管中断请求信号的优先级。

如果在某个中断服务程序中设置 IDLE1 模式，则之后产生的中断请求信号按如下所述来执行。

- 注意事项**
1. 通过设置 PSC.NMI1M, PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号是无效的，IDLE1 模式也不会被释放。
 2. 如果通过使用 NFC 寄存器来选择消除数字噪声的方法且如果采样时钟是从 fxx/64, fxx/128, fxx/256, fxx/512 以及 fxx/1024 中选择的，那么 IDLE1 模式将不能通过 INTP3 引脚的中断请求信号来释放。详细信息请参见 14.6.2（4） 噪声消除控制寄存器（NFC）。

（a）如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 IDLE1 模式，但是不响应此中断请求信号。此中断请求信号被保持。

（b）如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 IDLE1 模式，并且响应此中断请求信号。

表 16-4. 通过中断请求信号释放 IDLE1 模式后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	执行对处理地址的跳转。	
可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令。	执行下一条指令。

(2) 通过复位释放 IDLE1 模式

执行与正常复位操作相同的操作。

表 16-5. IDLE1 模式中的操作状态

IDLE1 模式设置		操作状态	
		当不使用副时钟时	当使用副时钟时
项目			
主时钟振荡器		允许振荡	
副时钟振荡器		—	允许振荡
内部振荡器		允许振荡	
PLL		可操作	
CPU		停止操作	
中断控制器		停止操作 (但是允许释放待机模式)	
定时器 P (TMP0 至 TMP3)		停止操作	
定时器 Q (TMQ0)		停止操作	
定时器 M (TMM0)		当选择 $f_{R/8}$ 作为计数时钟时可操作	当选择 $f_{R/8}$ 或 f_{XT} 作为计数时钟时可操作
钟表定时器		当选择 f_x (BRG 分频) 作为计数时钟时可操作	可操作
看门狗定时器 2		可操作	
串行接口	CSIB0, CSIB1	当选择 \overline{SCKBn} 输入时钟作为计数时钟时可操作 ($n = 0, 1$)	
	UARTA0, UARTA1	停止操作 (但是当选择 $\overline{ASCKA0}$ 输入时钟时是可操作的)	
A/D 转换器		保持操作 (保持转换结果) ^注	
按键中断功能 (KR)		可操作	
端口功能		保持设置 IDLE1 模式之前的状态	
内部数据		CPU 寄存器, 状态, 数据和全部其它的内部数据, 例如内部 RAM 的内容, 被保持为 IDLE1 模式设置之前的状态。	

注 为实现低功耗, 在设置 IDLE1 模式之前停止 A/D 转换器。

16.5 IDLE2 模式

16.5.1 设置和操作状态

通过在正常操作模式中设置 PSMR.PSM1 和 PSMR.PSM0 位为 10，并设置 PSC.STP 位为 1 设置 IDLE2 模式。

在 IDLE2 模式中，时钟振荡器继续操作但停止给 CPU，PLL，flash 存储器和其它内置外围设备功能的时钟供应。

结果，停止程序执行，并且保留在设置 IDLE2 模式之前的内部 RAM 的内容。CPU，PLL 和其它内置外围设备功能停止操作。但是，使用副时钟或外部时钟的内置外围设备功能继续操作。

表 16-7 显示了在 IDLE2 模式中的操作状态。

因为 IDLE2 模式停止了内置外围设备功能，PLL 和 flash 存储器的操作，所以 IDLE2 模式比 IDLE1 模式更能降低功耗。然而，因为 PLL 和 flash 存储器被停止，所以当释放 IDLE2 模式时，对 PLL 和 flash 存储器要求有一段设置时间。

- 注意事项**
1. 在执行往 PSC 寄存器中存入数据来设置 IDLE2 模式的指令之后，插入五个或更多的 NOP 指令。
 2. 如果设置 IDLE2 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则 IDLE2 模式立即被此等待响应的中断请求释放。

16.5.2 释放 IDLE2 模式

可通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号），未屏蔽的外部中断请求信号（INTP0 至 INTP7 引脚输入），来自于在 IDLE2 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，WDT2RES 信号，上电清零电路（POC），低电压检测器（LVI）或时钟监视器（CLM）产生的复位）来释放 IDLE2 模式。在设置 IDLE2 模式之前，PLL 返回到操作状态。

在 IDLE2 模式释放后，恢复正常操作模式。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE2 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE2 模式，而不管中断请求信号的优先级。

如果在某个中断服务程序中设置 IDLE2 模式，则之后产生的中断请求信号按如下所述来执行。

- 注意事项**
1. 通过设置 PSC.NMI1M，PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号是无效的，IDLE2 模式也不会被释放。
 2. 如果通过使用 NFC 寄存器来选择消除数字噪声的方法且如果采样时钟是从 fxx/64，fxx/128，fxx/256，fxx/512 以及 fxx/1024 中选择的，那么 IDLE2 模式将不能通过 INTP3 引脚的中断请求信号来释放。详细信息请参见 14.6.2（4） 噪声消除控制寄存器（NFC）。

（a）如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 IDLE2 模式，但是不响应此中断请求信号。此中断请求信号被保持。

（b）如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 IDLE2 模式，并且响应此中断请求信号。

表 16-6. 通过中断请求信号释放 IDLE2 模式后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	在保证规定的设置时间之后执行对处理地址的跳转。	
可屏蔽中断请求信号	在保证规定的设置时间之后执行对处理地址的跳转或执行下一条指令。	在保证规定的设置时间之后执行下一条指令。

(2) 通过复位释放 IDLE2 模式

执行与正常复位操作相同的操作。

表 16-7. IDLE2 模式中的操作状态

IDLE2 模式设置 项目		操作状态	
		当不使用副时钟时	当使用副时钟时
主时钟振荡器		允许振荡	
副时钟振荡器		—	允许振荡
内部振荡器		允许振荡	
PLL		停止操作	
CPU		停止操作	
中断控制器		停止操作（但是允许释放待机模式）	
定时器 P（TMP0 至 TMP3）		停止操作	
定时器 Q（TMQ0）		停止操作	
定时器 M（TMM0）		当选择 $f_{R/8}$ 作为计数时钟时可操作	当选择 $f_{R/8}$ 或 f_{XT} 作为计数时钟时可操作
钟表定时器		当选择 f_x （BRG 分频）作为计数时钟时可选择	可操作
看门狗定时器 2		可操作	
串行接口	CSIB0, CSIB1	当选择 \overline{SCKBn} 输入时钟作为计数时钟时可操作（ $n = 0, 1$ ）	
	UARTA0, UARTA1	停止操作（但是当选择 ASCKA0 输入时钟时 UARTA0 是可操作的）	
A/D 转换器		保持操作（保持转换结果） ^注	
按键中断功能（KR）		可操作	
端口功能		保持设置 IDLE2 模式之前的状态	
内部数据		CPU 寄存器，状态，数据和全部其它的内部数据，例如内部 RAM 的内容，被保持为 IDLE2 模式设置之前的状态。	

注 为实现低功耗，在设置 IDLE2 模式之前停止 A/D 转换器。

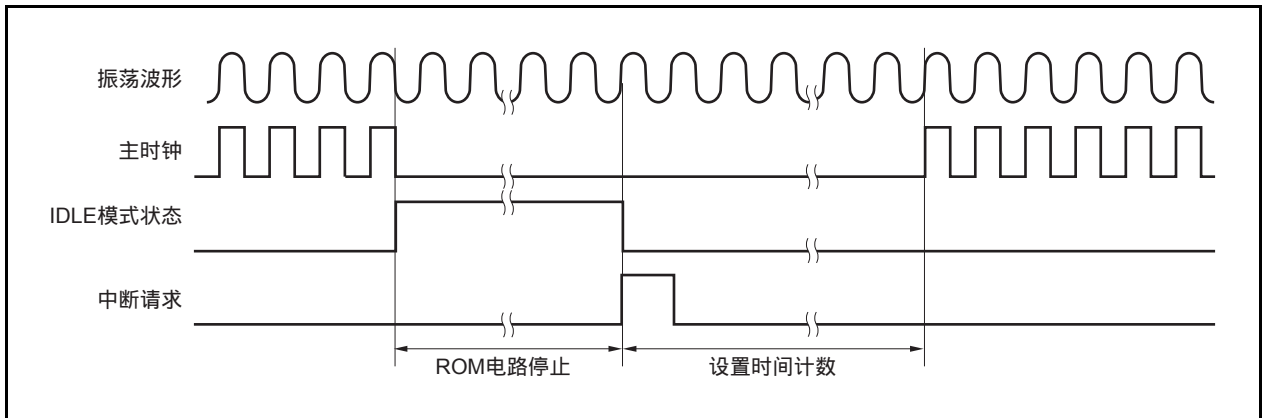
16.5.3 当释放 IDLE2 模式时保证设置时间

在释放 IDLE2 模式之后需保证 ROM（flash 存储器）的设置时间，这是因为在设置 IDLE2 模式之后，除了主时钟振荡器外，block 的操作也被停止了。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE2 模式

通过设置 OSTS 寄存器保证指定的设置时间。

当产生释放源时，根据 OSTS 寄存器的设置，专用的内部定时器开始计数。当它溢出时，正常操作模式被恢复。



（2）通过复位（ $\overline{\text{RESET}}$ 引脚输入，WDT2RES 产生）释放

此操作与正常复位操作相同。

振荡稳定时间是 OSTS 寄存器的初始值， $2^{16}/f_x$ 。

16.6 STOP 模式

16.6.1 设置和操作状态

通过在正常操作模式中设置 PSMR.PSM1 和 PSMR.PSM0 位为 01 或 11，并设置 PSC.STP 位为 1 设置 STOP 模式。

在 STOP 模式中，副时钟振荡器继续操作，主时钟振荡器停止。停止给 CPU 和内置外围设备功能的时钟供应。

结果，停止程序执行，并且保持在设置 STOP 模式之前的内部 RAM 的内容。使用副时钟振荡器或外部时钟的内置外围设备功能继续操作。

表 16-9 显示了在 STOP 模式中的操作状态。

因为 STOP 模式停止了主时钟振荡器的操作，所以 STOP 模式比 IDLE2 模式更能降低功耗。如果不使用副时钟振荡器，内部振荡器和外部时钟，则功耗最低仅为漏电流消耗的功耗。

- 注意事项**
1. 在执行往 PSC 寄存器中存入数据来设置 STOP 模式的指令之后，插入五个或更多的 NOP 指令。
 2. 如果设置 STOP 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则 STOP 模式立即被此等待响应的中断请求释放。

16.6.2 释放 STOP 模式

可通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号），未屏蔽的外部中断请求信号（INTP0 至 INTP7 引脚输入），来自于在 STOP 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，WDT2RES 信号，上电清零电路（POC）或低电压检测器（LVI）产生的复位）来释放 STOP 模式。

在 STOP 模式释放后，在振荡稳定时间被保证之后，恢复正常操作模式。

- 注意事项**
1. 通过设置 PSC.NMI1M，PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号是无效的，STOP 模式也不会被释放。
 2. 如果通过使用 NFC 寄存器来选择消除数字噪声的方法且如果采样时钟是从 fxx/64，fxx/128，fxx/256，fxx/512 以及 fxx/1024 中选择的，那么 STOP 模式将不能通过 INTP3 引脚的中断请求信号来释放。详细信息请参见 14.6.2（4） 噪声消除控制寄存器（NFC）。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 STOP 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 STOP 模式，而不管中断请求信号的优先级。如果在某个中断服务程序中设置 STOP 模式，则之后产生的中断请求信号按如下所述来执行。

- （a）如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放 STOP 模式，但是不响应此中断请求信号。此中断请求信号被保持。
- （b）如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放 STOP 模式，并且响应此中断请求信号。

表 16-8. 通过中断请求信号释放 STOP 模式后的操作

释放源	中断允许 (EI) 状态	中断禁止 (DI) 状态
不可屏蔽中断请求信号	在保证振荡稳定时间之后执行对处理地址的跳转。	
可屏蔽中断请求信号	在保证振荡稳定时间之后执行对处理地址的跳转或执行下一条指令。	在保证振荡稳定时间之后执行下一条指令。

(2) 通过复位释放 STOP 模式

执行与正常复位操作相同的操作。

表 16-9. STOP 模式中的操作状态

STOP 模式设置		操作状态	
		当不使用副时钟时	当使用副时钟时
主时钟振荡器		停止振荡	
副时钟振荡器		—	允许振荡
内部振荡器		允许振荡	
PLL		停止操作	
CPU		停止操作	
中断控制器		停止操作（但是允许释放待机模式）	
定时器 P（TMP0 至 TMP3）		停止操作	
定时器 Q（TMQ0）		停止操作	
定时器 M（TMM0）		当选择 $f_R/8$ 作为计数时钟时可操作	当选择 $f_R/8$ 或 f_{XT} 作为计数时钟时可操作
钟表定时器		停止操作	当选择 f_{XT} 作为计数时钟时可操作
看门狗定时器 2		当选择 f_R 作为计数时钟时可操作	
串行接口	CSIB0, CSIB1	当选择 \overline{SCKBn} 输入时钟作为计数时钟时可操作（ $n = 0, 1$ ）	
	UARTA0, UARTA1	停止操作（但是当选择 $\overline{ASCKA0}$ 输入时钟时，UARTA0 是可操作的）	
A/D 转换器		停止操作（转换结果不确定） ^{注 1, 2}	
按键中断功能（KR）		可操作	
端口功能		保持 STOP 模式设置之前的状态	
内部数据		CPU 寄存器，状态，数据和全部其它的内部数据，例如内部 RAM 的内容，被保持为 STOP 模式设置之前的状态。	

- 注**
1. 当操作 A/D 转换器时，如果设置 STOP 模式，则 A/D 转换器自动停止操作，并且在 STOP 模式释放后再次启动操作。然而，在这种情况下，释放 STOP 模式后的 A/D 转换结果是无效的。所有设置 STOP 模式之前的 A/D 转换结果也是无效的。
 2. 即使在操作 A/D 转换器时设置 STOP 模式，功耗降低也与在设置 STOP 模式之前停止 A/D 转换器时的一样。

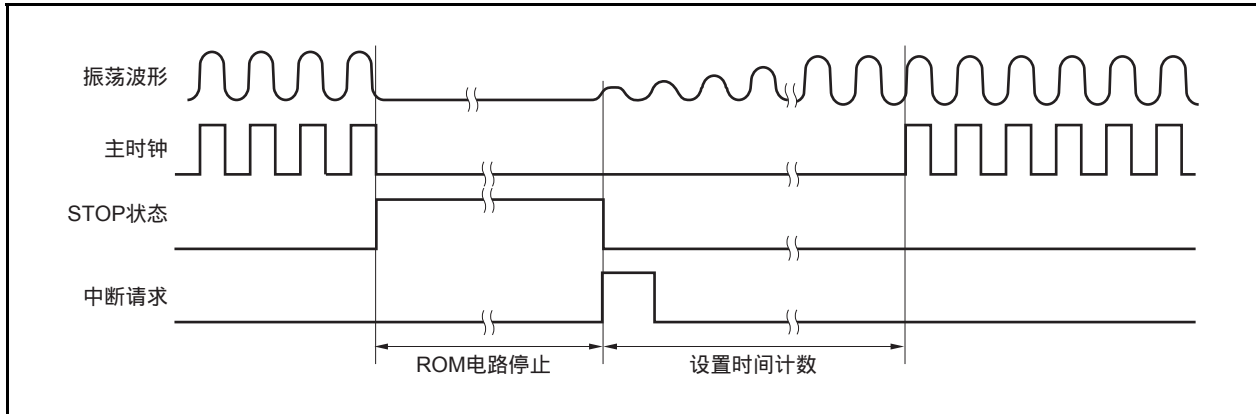
16.6.3 当释放 STOP 模式时保证振荡稳定时间

在释放 STOP 模式后需保证主时钟振荡器的振荡稳定时间，这是因为在设置 STOP 模式之后主时钟振荡器操作停止。

(1) 通过不可屏蔽中断请求信号或未屏蔽的中断请求信号释放 STOP 模式

通过设置 OSTS 寄存器保证振荡稳定时间。

当产生释放源时，根据 OSTS 寄存器的设置，专用内部定时器开始计数。当它溢出时，恢复为正常操作模式。



(2) 通过复位释放

此操作与正常复位操作相同。

振荡稳定时间是 OSTS 寄存器的初始值， $2^{16}/f_x$ 。

16.7 副时钟操作模式

16.7.1 设置和操作状态

通过在正常操作模式中设置 PCC.CK3 位 1 来设置副时钟操作模式。

当副时钟操作模式被设置时，内部系统时钟从主时钟转换为副时钟。通过使用 PCC.CLS 位检测是否时钟已被切换。

当 PCC.MCK 位被设置为 1 时，主时钟振荡器的操作被停止。这样，系统仅以副时钟来操作。

在副时钟操作模式中，与正常操作模式相比，功耗降到一个较低的水平上，这是因为副时钟被用作内部系统时钟。除此之外，通过停止主时钟振荡器的操作，功耗能更进一步降低到 STOP 模式的等级。

表 16-10 显示了副时钟操作模式的状态。

- 注意事项**
1. 当操作 CK3 位时，不要改变 PCC.CK2 至 PCC.CK0 位（推荐使用位操作指令来操作）的设置值。对于 PCC 寄存器的细节描述，参见 5.3（1）处理器时钟控制寄存器（PCC）。
 2. 如果以下的条件不满足，改变 CK2 至 CK0 位的设置以便满足条件并且设置副时钟操作模式。
内部主时钟 (f_{CLK}) > 副时钟 (f_{XT}) × 4

备注 内部系统时钟 (f_{CLK})：由符合 CK2 到 CK0 位设置的主时钟 (f_{xx}) 产生的时钟。

16.7.2 释放副时钟操作模式

当 CK3 位被清零时，可以通过复位信号（通过 $\overline{\text{RESET}}$ 引脚输入，WDT2RES 信号，上电清零电路（POC），低电压检测器（LVI）或时钟检测器（CLM）复位）释放副时钟操作模式。

如果停止主时钟（MCK 位 = 1），设置 MCK 位为 1，通过软件保证主时钟的振荡稳定时间，并将 CK3 位清零。

当副时钟操作模式释放后，恢复为正常操作模式。

- 注意事项**
- 当操作 CK3 位时，不要改变 CK2 至 CK0 位（推荐使用位操作指令来操作）的设置值。
对于 PCC 寄存器的细节描述，参见 5.3（1）处理器时钟控制寄存器（PCC）。

表 16-10. 副时钟操作模式中的操作状态

副时钟操作模式设置 项目		操作状态	
		当主时钟振荡时	当主时钟停止时
副时钟振荡器		允许振荡	
内部振荡器		允许振荡	
PLL		可操作	停止操作 ^註
CPU		可操作	
中断控制器		可操作	
定时器 P (TMP0 至 TMP3)		可操作	停止操作
定时器 Q (TMQ0)		可操作	停止操作
定时器 M (TMM0)		可操作	当选择 $f_R/8$ 或 f_{XT} 作为计数时钟时可操作
钟表定时器		可操作	当选择 f_{XT} 作为计数时钟时可操作
看门狗定时器 2		可操作	当选择 f_R 作为计数时钟时可操作
串行接口	CSIB0, CSIB1	可操作	当选择 \overline{SCKBn} 输入时钟作为计数时钟时可操作 ($n = 0, 1$)
	UARTA0, UARTA1	可操作	停止操作 (但是当选择 ASCKA0 输入时钟时 UARTA0 是可操作的)
A/D 转换器		可操作	停止操作
按键中断功能 (KR)		可操作	
端口功能		Set 表	
内部数据		Set 表	

注 请确保在停止主时钟前停止 PLL (PLLCTL.PLLON = 0)。

注意事项 当 CPU 工作在副时钟并且主时钟振荡停止时，不能访问发生等待的寄存器。如果产生等待，则其只能通过复位来释放 (参见 3.4.8 (2))。

16.8 副 IDLE 模式

16.8.1 设置和操作状态

通过在副时钟操作模式中设置 PSMR.PSM1 和 PSMR.PSM0 位为 10，并设置 PSC.STP 位为 1 来设置副 IDLE 模式。

在此模式中，时钟振荡器继续操作，停止给 CPU，flash 存储器和其它内置外围设备功能的时钟供应。

结果，停止程序执行，并且保持设置副 IDLE 模式之前的内部 RAM 的内容。停止 CPU 和其它内置外围设备功能。然而，可使用副时钟或外部时钟操作的内置外围设备功能继续操作。

因为副 IDLE 模式停止了 CPU，flash 存储器和其它外围设备功能的操作，所以它比副时钟操作模式更能降低功耗。如果在主时钟停止后设置副 IDLE 模式，当前功耗能降低至 STOP 模式的等级。

表 16-12 显示了副 IDLE 模式中的操作状态。

- 注意事项**
1. 为设置副 IDLE 模式而将指令保存到 PSC 寄存器后，插入五个或更多的 NOP 指令。
 2. 如果设置副 IDLE 模式时有一个未被屏蔽的中断请求信号被保持等待响应，则副 IDLE 模式立即被此等待响应的中断请求释放。

16.8.2 释放副 IDLE 模式

可通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号），未屏蔽的外部中断请求信号（INTP0 至 INTP7 引脚输入），来自于在副 IDLE 模式中可操作的外围设备功能的未屏蔽的内部中断请求信号，或复位信号（ $\overline{\text{RESET}}$ 引脚输入，WDT2RES 信号，上电清零电路（POC），低电压检测器（LVI）或时钟监视器（CLM）产生的复位）来释放副 IDLE 模式。在设置副 IDLE 模式之前，PLL 返回到操作状态。

当通过中断请求信号释放副 IDLE 模式时，副时钟操作模式被设置。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放副 IDLE 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放副 IDLE 模式，而不管中断请求信号的优先级。

如果在某个中断服务程序中设置副 IDLE 模式，则之后产生的中断请求信号按如下所述来执行。

- 注意事项**
1. 通过设置 **PSC.NMI1M**，**PSC.NMI0M** 和 **PSC.INTM** 位为 1 来禁止中断请求信号，这样中断请求信号是无效的，副 IDLE 模式也不会被释放。
 2. 当释放副 IDLE 模式时，从产生释放副 IDLE 模式的中断请求信号开始到副 IDLE 模式被释放之间须等待 12 个副时钟周期（大约 366 μs ）。
 3. 如果通过使用 **NFC** 寄存器来选择消除数字噪声的方法且如果采样时钟是从 **fx/64**，**fx/128**，**fx/256**，**fx/512** 以及 **fx/1024** 中选择的，那么副 IDLE 模式将不能通过 **INTP3** 引脚的中断请求信号来释放。详细信息请参见 14.6.2（4） 噪声消除控制寄存器（**NFC**）。

（a）如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较低的优先级，则释放副 IDLE 模式，但是不响应此中断请求信号。此中断请求信号被保持。

（b）如果中断请求信号比当前正在执行的中断请求服务的中断请求具有较高的优先级（包括不可屏蔽中断请求信号），则释放副 IDLE 模式，并且响应此中断请求信号。

表 16-11. 通过中断请求信号释放副 IDLE 模式之后的操作

释放源	中断允许（EI）状态	中断禁止（DI）状态
不可屏蔽中断请求信号	执行对处理地址的跳转。	
可屏蔽中断请求信号	执行对处理地址的跳转或执行下一条指令。	执行下一条指令。

(2) 通过复位释放副 IDLE 模式

执行与正常复位操作相同的操作。

表 16-12. 在副 IDLE 模式中的操作状态

副 IDLE 模式设置 项目		操作状态	
		当主时钟振荡时	当主时钟停止时
副时钟振荡器		允许振荡	
内部振荡器		允许振荡	
PLL		可操作	停止操作 ^{注1}
CPU		停止操作	
中断控制器		停止操作（但是允许释放待机模式）	
定时器 P（TMP0 至 TMP3）		停止操作	
定时器 Q（TMQ0）		停止操作	
定时器 M（TMM0）		当选择 $f_R/8$ 或 f_{XT} 作为计数时钟时可操作	
钟表定时器		停止操作	当选择 f_{XT} 作为计数时钟时可操作
看门狗定时器 2		当选择 f_R 作为计数时钟时可操作	
串行接口	CSIB0, CSIB1	当选择 \overline{SCKBn} 输入时钟作为计数时钟时可操作（ $n = 0, 1$ ）	
	UARTA0, UARTA1	停止操作（但是当选择 $\overline{ASCKA0}$ 输入时钟时 UARTA0 是可操作的）	
A/D 转换器		保持操作（保持转换结果） ^{注2}	
按键中断功能（KR）		可操作	
端口功能		保持设置副 IDLE 模式之前的状态	
内部数据		CPU 寄存器，状态，数据和全部其它的内部数据，例如内部 RAM 的内容，被保持为副 IDLE 模式设置之前的状态。	

- 注
1. 停止主时钟之前确定停止 PLL（PLLCTL.PLLON 位 = 0）。
 2. 为了实现低功耗，在进入副 IDLE 模式之前停止 A/D 转换器。

17.1 概述

以下的复位功能可用。

(1) 四种复位源

- 通过 **RESET** 引脚输入的外部复位
- 通过看门狗定时器 2 (WDT2) 溢出 (WDT2RES) 引起的复位
- 通过低电压检测器 (LV1) 供电电压和检测电压的对比引起的系统复位
- 通过检测到时钟监控器 (CLM) 停止振荡引起的系统复位
- 通过上电清零电路的系统复位

当一个复位信号释放后，复位源可以通过使用复位源标志寄存器 (RESF) 来确认。

(2) 紧急操作模式

复位后在主时钟振荡稳定时间内如果 WDT2 溢出，则判断为一个不规则的主时钟振荡并且 CPU 在内部振荡时钟下开始操作。

注意事项 当以内部振荡时钟来操作 CPU 时，禁止访问生成有等待状态的寄存器。关于生成有等待状态的寄存器的详细信息，请参见 3.4.8 (2) 访问特殊片上外围 I/O 寄存器。

17.2 检测复位源的寄存器

V850ES/HE2 有 4 个复位源。当一个复位释放后，出现的复位源可以通过复位源标志寄存器（RESF）检测。

(1) 复位源标志寄存器（RESF）

RESF 寄存器是一个特殊的寄存器，只能够通过特殊的顺序来写入（参见 3.4.7 特殊寄存器）。

RESF 寄存器表明了复位信号产生的源。

该寄存器可以以 8 位或 1 位为单位来读取或写入。

$\overline{\text{RESET}}$ 引脚输入或 POC 复位将该寄存器清除到 00H。如果复位源不是 $\overline{\text{RESET}}$ 引脚信号，缺省值则有所不同。

复位后： 00H^注 R/W 地址： FFFFF888H

	7	6	5	4	3	2	1	0
RESF	0	0	0	WDT2RF	0	0	CLMRF	LVIRF

WDT2RF	从WDT2产生的复位信号
0	不产生
1	产生

CLMRF	从CLM产生的复位信号
0	不产生
1	产生

LVIRF	从LVI产生的复位信号
0	不产生
1	产生

注 当通过 $\overline{\text{RESET}}$ 引脚执行复位时，RESF 寄存器的值将清为 00H。当通过看门狗定时器 2（WDT2），低电压检测器（LVI），或者时钟监控器（CLM）执行复位时，该寄存器复位标志（WDT2RF 位，CLMRF 位，和 LVIRF 位）将被设置。但是，其它源将被保留。

注意事项 只有“0”可以被写入到该寄存器中的每一位中。如果写入的“0”和设置标志（复位事件）冲突，则设置标志优先。

17.3 操作

17.3.1 通过 $\overline{\text{RESET}}$ 引脚的复位操作

当 $\overline{\text{RESET}}$ 引脚输入一个低电平时，系统将被复位，且所有硬件将被初始化。

当 $\overline{\text{RESET}}$ 引脚的电平由低变高时，复位状态将被释放。

表 17-1. $\overline{\text{RESET}}$ 引脚输入时的硬件状态

项目		复位期间	复位后
主时钟振荡器 (f_x)		停止振荡	开始振荡
子时钟振荡器 (f_{XT})	晶振	继续振荡	
	RC 振荡	停止振荡	开始振荡
内部振荡器		停止振荡	开始振荡
外围时钟 (f_x 至 $f_x/1, 024$)		停止操作	安全振荡稳定时间之后开始振荡
内部系统时钟 (f_{CLK})， CPU 时钟 (f_{CPU})		停止操作	安全振荡稳定时间之后开始操作（初始化到 $f_{xx}/8$ ）
CPU		初始化	安全振荡稳定时间之后开始执行程序
看门狗定时器 2		停止操作（初始化到 0）	开始操作
内部 RAM		如果上电复位或 CPU 访问和复位输入冲突（数据被破坏）则不确定。 否则，值在复位输入后会立即保留 ^{注1} 。	
I/O 线路（端口/复用功能引脚）		高阻抗 ^{注2}	
片上外围 I/O 寄存器		初始化至指定状态，OCDM 寄存器将被设置为（01H）。	
其他片上外围功能		停止操作	安全振荡稳定时间之后可以开始操作

- <R> 注
1. V850ES/HE2 中的固件会在内部系统复位状态被释放后使用部分内部 RAM，这是因为它支持引导置换功能。因此，在上电复位后一些 RAM 区域中的内容将不会被保留。详细信息请参见 17.4 复位释放后的操作。
 2. 当电源开启时，下列引脚可能会在复位时临时输出一个不确定的电平。

- P53/KR3/TIQ00/TOQ00/DDO 引脚

注意事项 OCDM 寄存器通过 $\overline{\text{RESET}}$ 引脚输入来初始化。因此，需要注意的是，如果在 OCDM.OCDM0 的位数被清除前，一个高电平在复位释放后被输入到 P05/DRST 引脚中，则可能会进入片上调试模式。详情请参见第 4 章 端口功能。

图 17-1. RESET 引脚输入引起的复位操作的时序

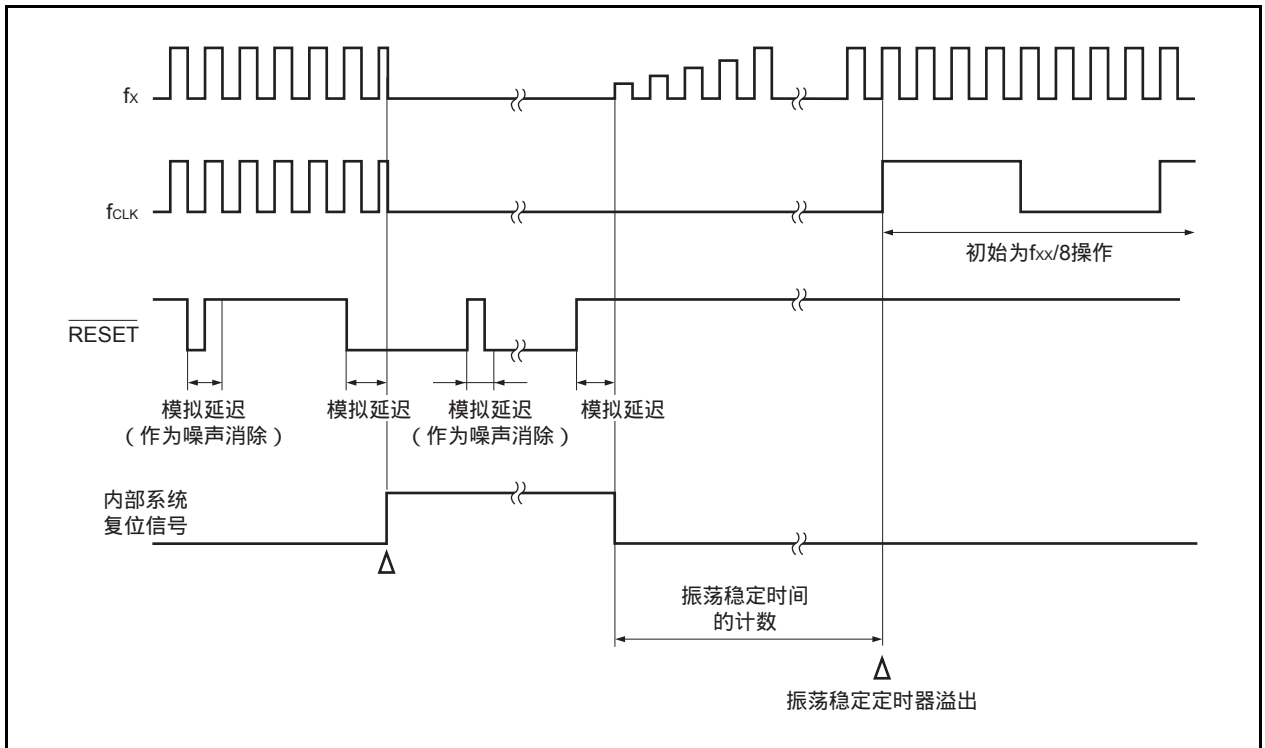
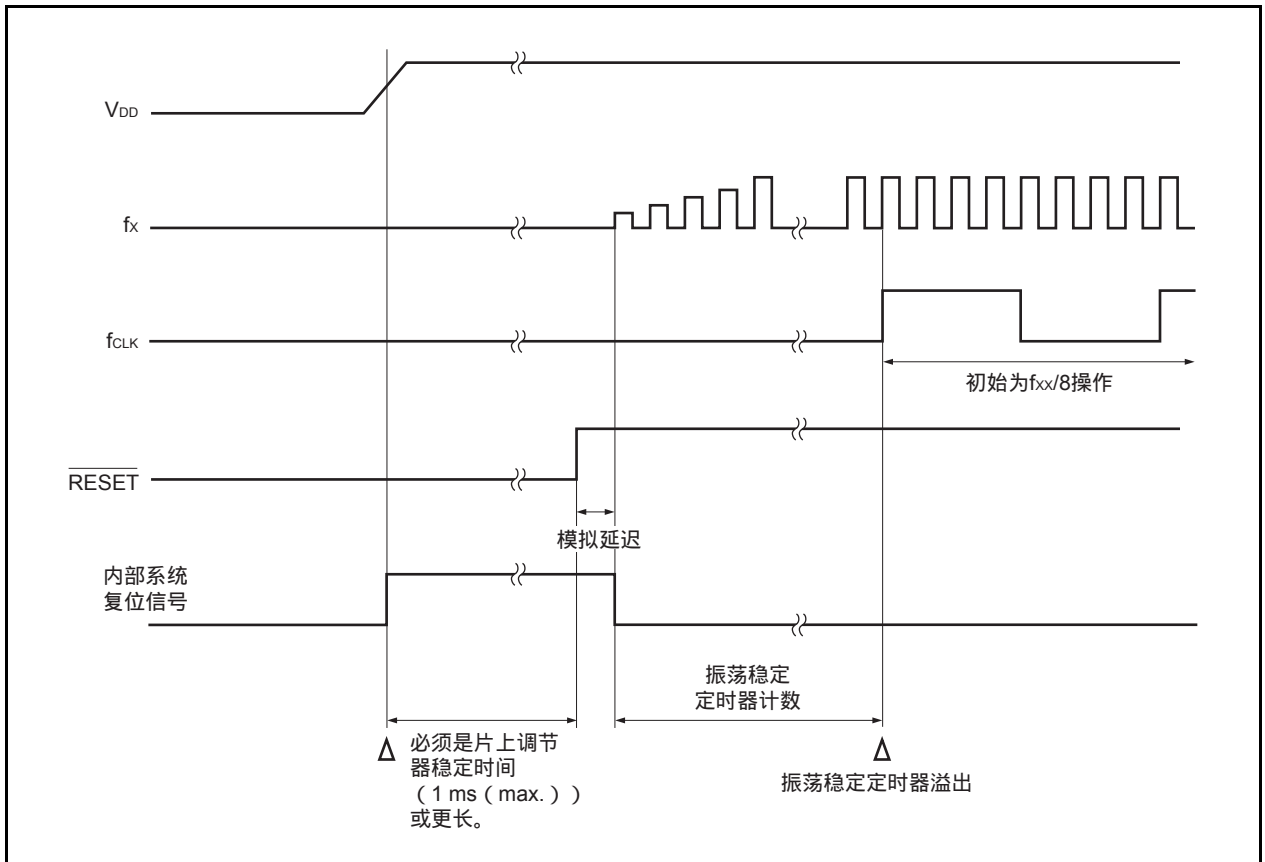


图 17-2. 上电复位操作时序



17.3.2 通过看门狗定时器 2 执行的复位操作

当看门狗定时器 2 由于溢出被设置为复位操作模式时，在看门狗定时器 2 溢出上（产生 WDT2RES 信号），将会执行系统复位且硬件将被初始化到初始状态。

在看门狗定时器 2 溢出后，会进入复位状态并且持续到预定时间（模拟延时），然后复位状态将会自动释放。

在复位期间主时钟振荡停止。

表 17-2. 看门狗定时器 2 复位操作时的硬件状态

项目		复位期间	复位后
主时钟振荡器 (fx)		停止振荡	开始振荡
子时钟振荡器 (fxt)	晶振	继续振荡	
	RC 振荡	停止振荡	开始振荡
内部振荡器		停止振荡	开始振荡
外围时钟 (fxx to fxx/1, 024)		停止操作	安全振荡稳定时间之后开始操作
内部系统时钟 (fxx), CPU 时钟 (fcPU)		停止操作	安全振荡稳定时间之后开始振荡（初始化为 fxx/8）
CPU		初始化	安全振荡稳定时间之后开始执行程序
看门狗定时器 2		停止操作（初始化到 0）	开始操作
内部 RAM		如果上电复位或 CPU 访问和复位输入冲突（数据被破坏）则不确定。否则，值在复位输入后会立即保留 ^注 。	
I/O 线路（端口/复用功能引脚）		高阻抗	
片上外围 I/O 寄存器		初始化至指定状态，OCDM 寄存器将保留其值。	
与上述不同的片上外围功能		停止操作	安全振荡稳定时间之后可以开始操作

<R> 注 V850ES/HE2 中的固件会在内部系统复位状态被释放后使用部分内部 RAM，这是因为它支持引导置换功能。因此，在上电复位后一些 RAM 区域中的内容将不会被保留。详细信息请参见 17.4 复位释放后的操作。

17.3.3 上电清零电路引起的复位操作

工作电压与检测电压会在上电清零操作被允许时进行比较。如果工作电压低于检测电压（包括供应电源时），系统将会被复位且每个硬件单元都会被初始为默认状态。

复位状态会从检测到电压下降开始一直保持到工作电压高于检测电压为止，之后复位状态会被自动清除。复位状态清除后，在主时钟振荡器（OSTS 寄存器的默认值： $2^{16}/f_x$ ）的振荡稳定的时间过去后，CPU 会开始执行程序。详细信息请参见第 19 章 上电清零电路。

17.3.4 低电压检测器引起的复位操作

允许 LVI 操作且 LVIM.LVIMD 位被设为“1”时，工作电压将会与检测电压进行比较。如果工作电压低于检测电压，系统将会被复位且每个硬件单元都会被初始为默认状态。

复位状态会从检测到电压下降开始一直保持到工作电压高于检测电压为止，之后复位状态会被自动清除。复位状态清除后，在主时钟振荡器（OSTS 寄存器的默认值： $2^{16}/f_x$ ）的振荡稳定的时间过去后，CPU 会开始执行程序。

详细信息请参见第 20 章 低压检测器。

17.3.5 时钟监控器引起的复位操作

允许时钟监控器操作时，主时钟会通过使用采样时钟（内部振荡器）来监控。如果检测到主时钟停止，系统将会被复位且每个硬件单元都会被初始为默认状态。

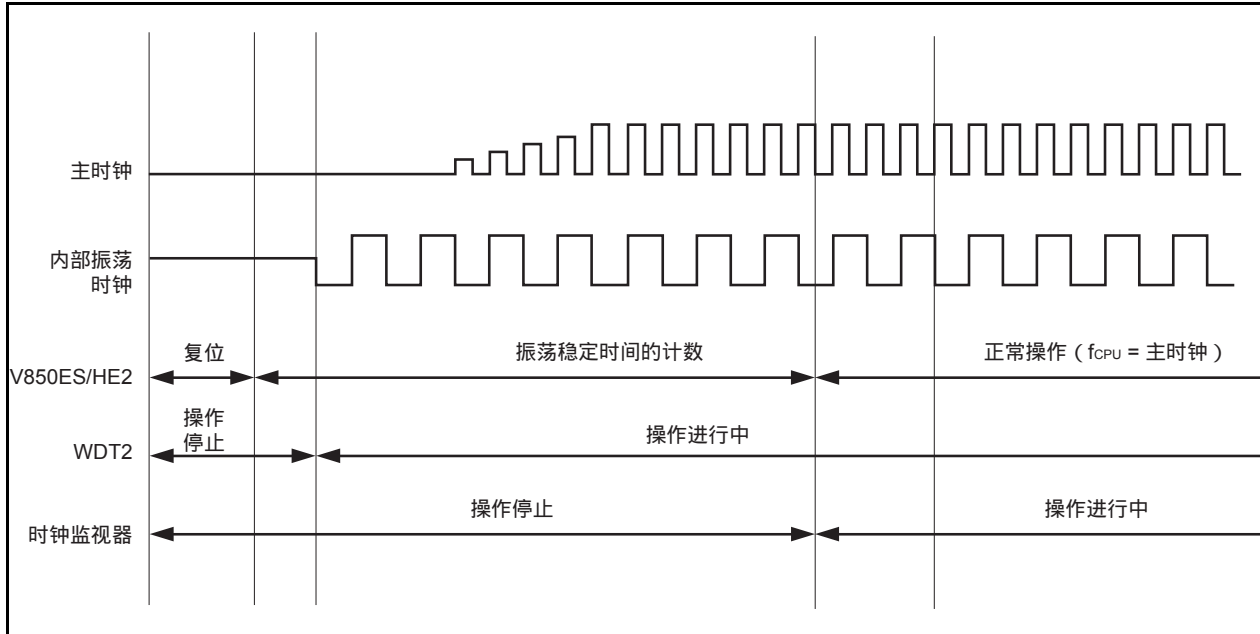
详细信息请参见第 18 章 时钟监控器。

<R> 17.4 复位释放后的操作

复位释放以后，主时钟开始振荡且振荡稳定时间（OSTS 寄存器初始值为： $2^{16}/f_x$ ）可靠，那么 CPU 将会开始执行程序。

在复位释放后，WDT2 会立即以内部振荡时钟为源时钟开始操作。

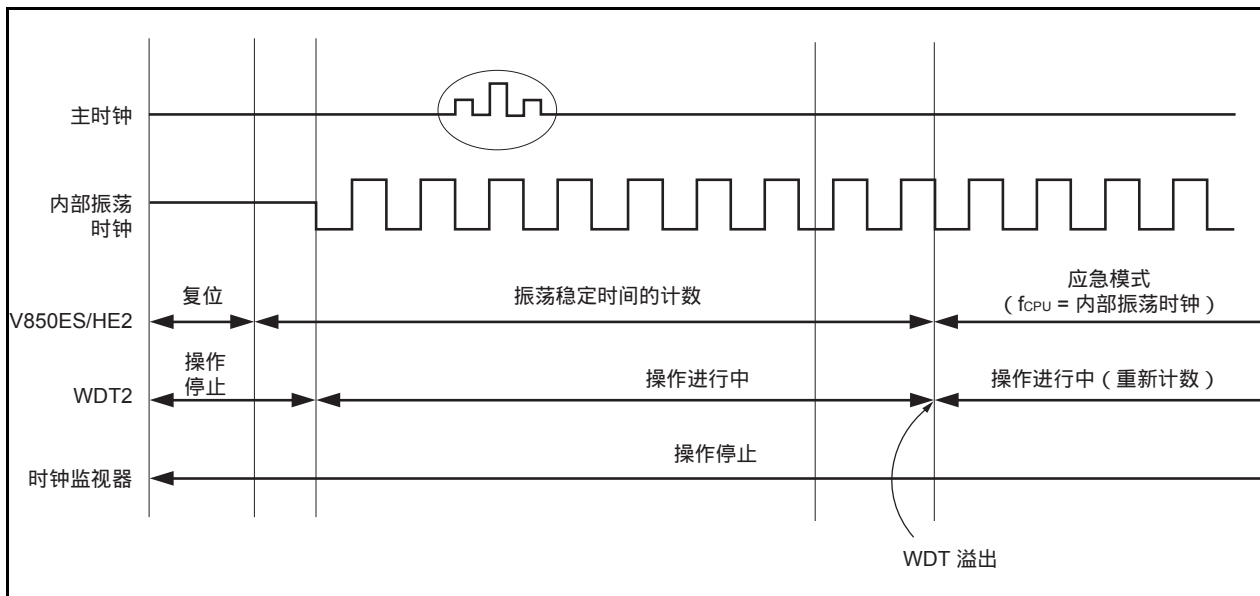
图 17-3. 复位后的操作



(1) 紧急操作模式

如果在振荡稳定时间还不可靠之前，在主时钟中发生异常情况，WDT2 在执行 CPU 程序之前将会溢出。此时，CPU 使用内部振荡时钟作为源时钟开始执行程序。

图 17-4. 复位释放后的操作

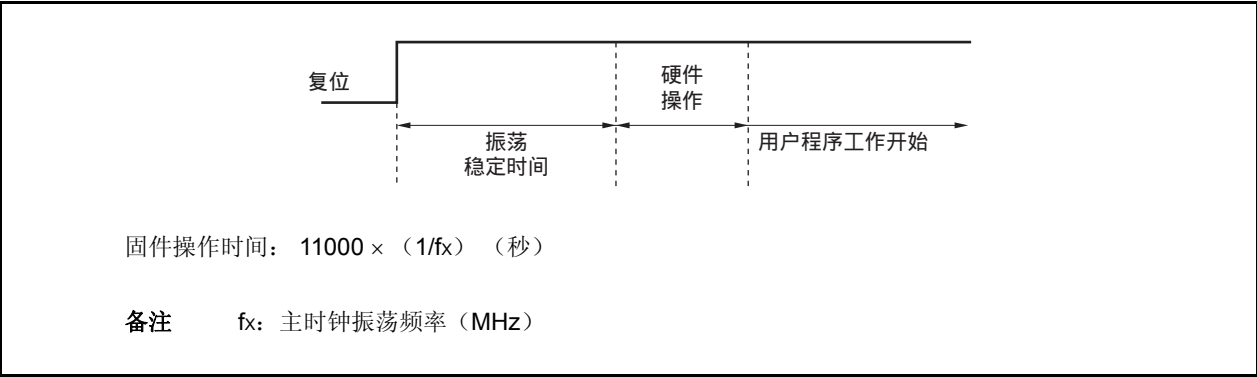


CPU 操作时钟状态可以用 CPU 操作时钟状态寄存器 (CCLS) 来检测。

(2) 固件操作

在 V850ES/HE2 中，复位释放后，片上固件将会在开始用户程序以支持引导置换功能前操作。

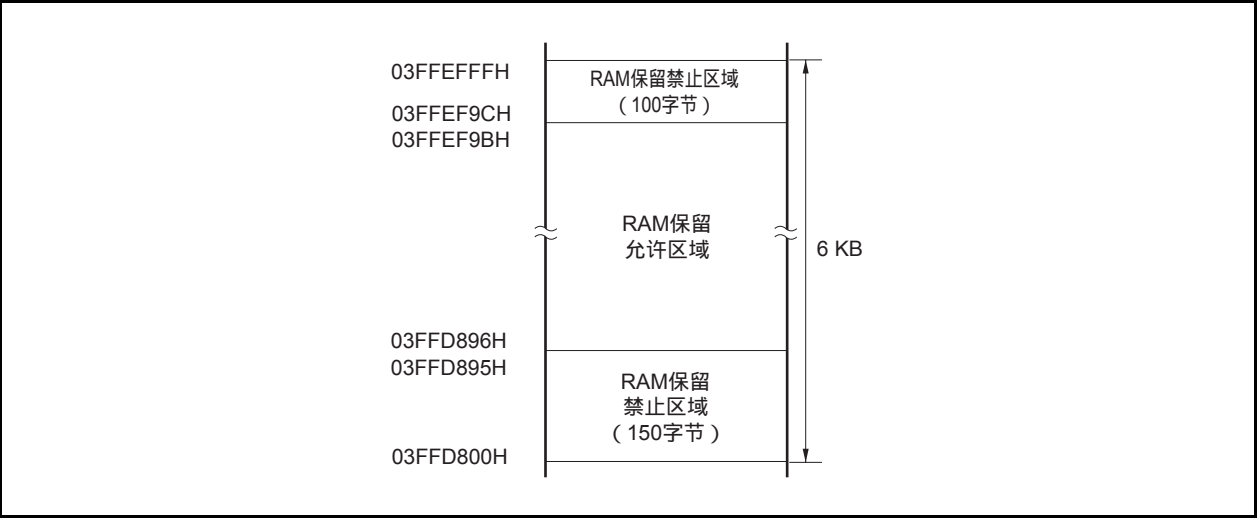
图 17-5. 固件操作



由于固件使用了部分内部 RAM，因此即使在电源打开的状态下，RAM 区域后的内容也不能通过复位来保持。

- 带有 6 KB RAM 的版本： 03FFD800H 至 03FFD895H， 03FFEF9CH 至 03FFEFFFH

图 17-6. RAM 允许保存区域



第 18 章 时钟监控器

18.1 功能

时钟监控器通过使用内部振荡时钟来抽取主时钟，并且在主时钟停止振荡的时候产生一个复位请求信号。

一旦通过操作允许标志允许时钟监控器的操作，除了复位无法通过其余任何方法来清零。

当通过时钟监控器执行复位时，RESF.CLMRF 位将被设置。关于 RESF 寄存器的详细资料，请参见 17.2 检测复位源的寄存器。

时钟监控器在下列条件下自动停止。

- STOP 模式释放后的振荡稳定时间内
- 当主时钟停止时（从副时钟操作 PCC.MCK 位= 1 开始，到主时钟操作 PCC.CLS 位= 0 结束）
- 当取样时钟（内部振荡时钟）停止时
- 当 CPU 使用内部振荡时钟操作时

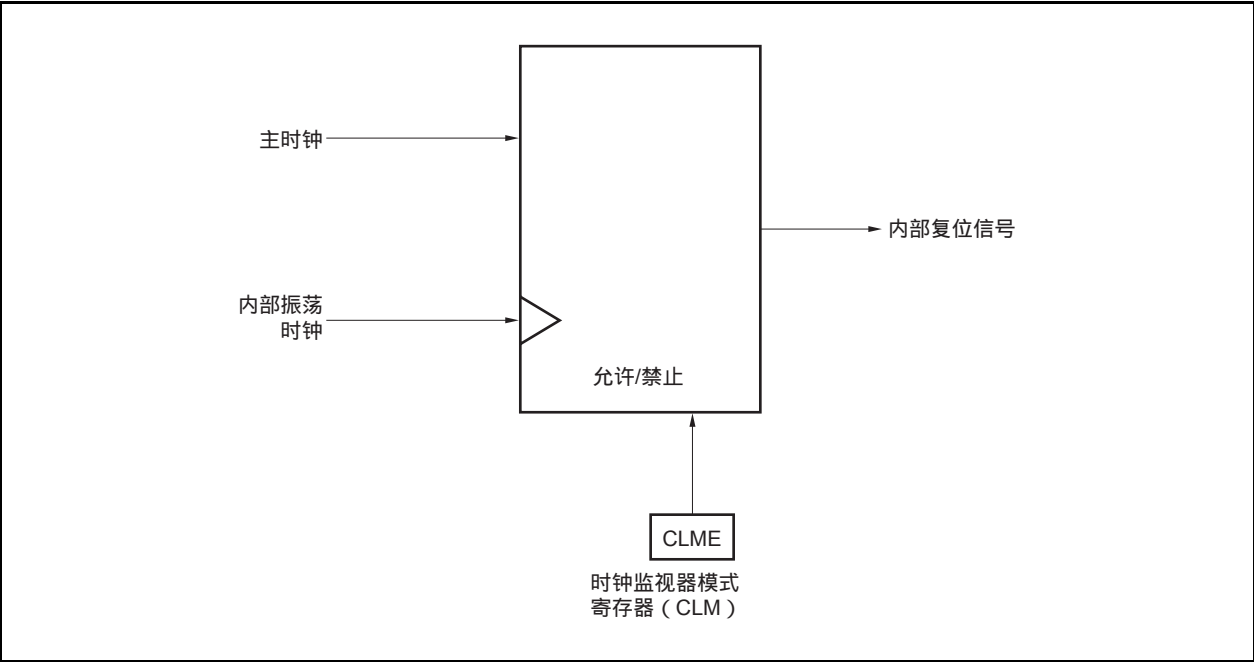
18.2 结构

时钟监控器包含以下硬件。

表 18-1. 时钟监控器的结构

项目	结构
控制寄存器	时钟监控器模式寄存器（CLM）

图 18-1. 时钟监控器的框图



18.3 寄存器

时钟监控器通过时钟监控器模式寄存器（CLM）来控制。

(1) 时钟监控器模式寄存器（CLM）

CLM 寄存器是一个特殊寄存器。它只能够通过特殊顺序的组合来写入（参见 3.4.7 特殊寄存器）。

该寄存器用来设置操作时钟监控器的模式。

该寄存器可以以 8 位或 1 位为单位来读取或写入。

复位操作将该寄存器设为 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		时钟监控器操作允许或禁止						
0		禁止时钟监控器操作。						
1		允许时钟监控器操作。						

- 注意事项
- 一旦 CLME 位被设为 1，除了复位其余任何操作都不能将其清为 0。
 - 当通过时钟监控器执行复位时，CLME 位将被清为 0 并且 RESF.CLMRF 被设置为 1。

18.4 操作

这个部分说明了时钟监控器的功能。开始和停止条件如下所示。

<开始条件>

当 CLM.CLME 位设置为 1 使允许操作。

<停止条件>

- 当 STOP 模式释放后开始计数振荡稳定时间时
- 当主时钟停止（从副时钟操作时 PCC.MCK 位=1 到主时钟操作时 PCC.CLS 位=0 为止）时
- 当取样时钟停止（内部振荡时钟）时
- 当 CPU 使用内部振荡时钟操作时

表 18-2. 时钟监控器的操作状态
(当 CLM.CLME 位=1, 在内部振荡时钟操作时)

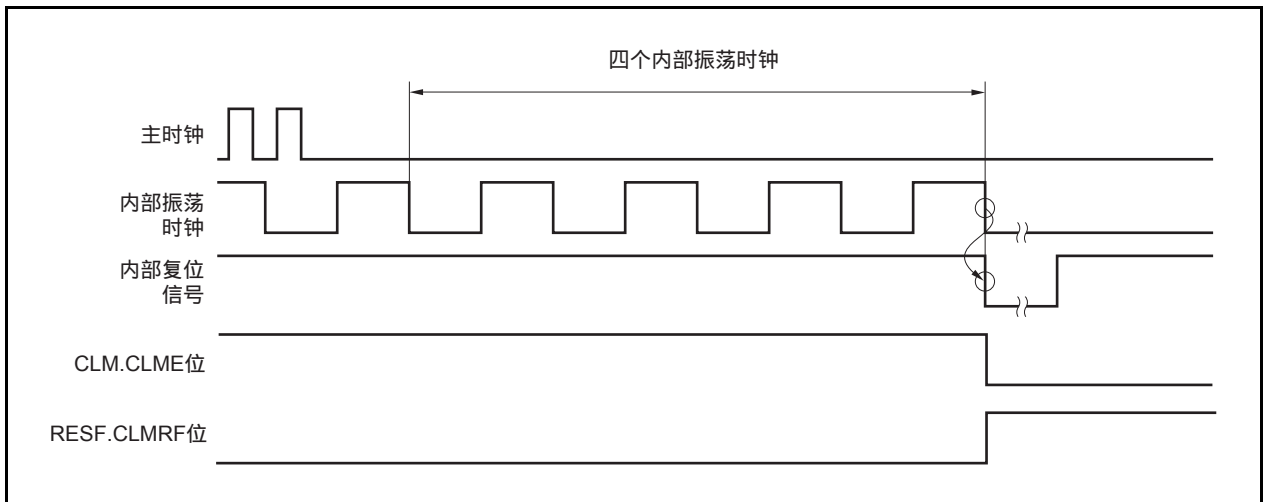
CPU 操作时钟	操作模式	主时钟状态	内部振荡时钟状态	时钟监控器状态
主时钟	HALT 模式	振荡	振荡 ^{注 1}	操作 ^{注 2}
	IDLE1, IDLE2 模式	振荡	振荡 ^{注 1}	操作 ^{注 2}
	STOP 模式	停止	振荡 ^{注 1}	停止
子时钟 (PCC.MCK 位 = 0)	副 IDLE 模式	振荡	振荡 ^{注 1}	操作 ^{注 2}
子时钟 (PCC.MCK 位 = 1)	副 IDLE 模式	停止	振荡 ^{注 1}	停止
内部振荡时钟	—	停止	振荡 ^{注 1}	停止
复位期间	—	停止	停止	停止

- 注
1. 内部振荡器可以通过使用选项字节功能并将 RCM.RSTOP 位设为 1 来停止（参见第 23 章）。
 2. 内部振荡器停止的时候时钟监控器也停止。

(1) 当主时钟振荡器停止时 (CLME 位=1) 的操作

当 CLME 位=1 的时候如果主时钟的振荡停止，一个如图 18-2 所示的内部复位信号将会产生。

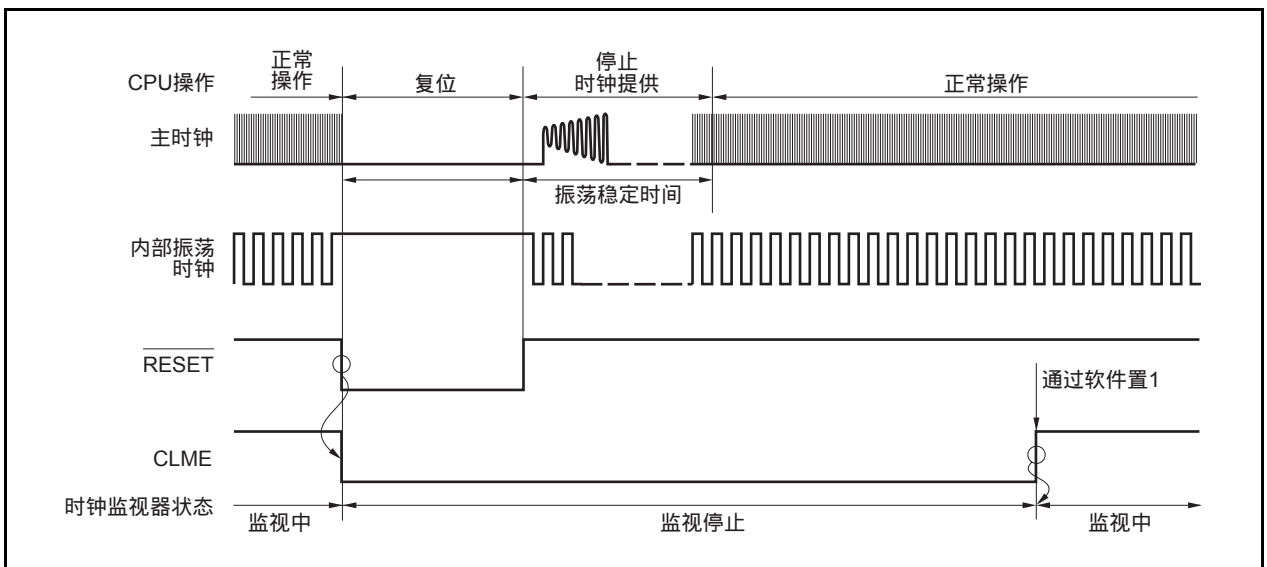
图 18-2. 主时钟振荡停止引起的复位周期

(2) $\overline{\text{RESET}}$ 输入后的时钟监控器状态

$\overline{\text{RESET}}$ 输入将 CLM.CLME 位清除为 0 并且停止时钟监控器的操作。在主时钟振荡稳定以后，当通过软件把 CLME 位设为 1，监控操作开始。

图 18-3. $\overline{\text{RESET}}$ 输入后的时钟监控器状态

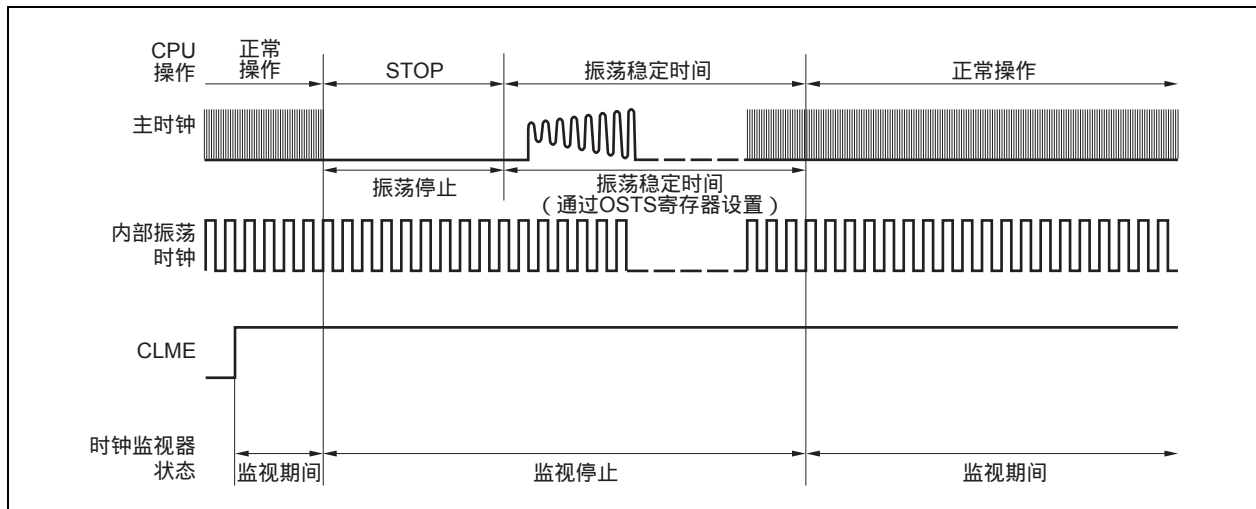
($\overline{\text{RESET}}$ 输入后且在主时钟振荡稳定以后，将设置 CLM.CLME 位=1)



(3) STOP 模式和 STOP 模式释放后的操作

如果在 CLM.CLME 位=1 时设置 STOP 模式，且在振荡稳定时间被计数时，监控操作将在 STOP 模式中停止。振荡稳定时间过后，监控器自动开始操作。

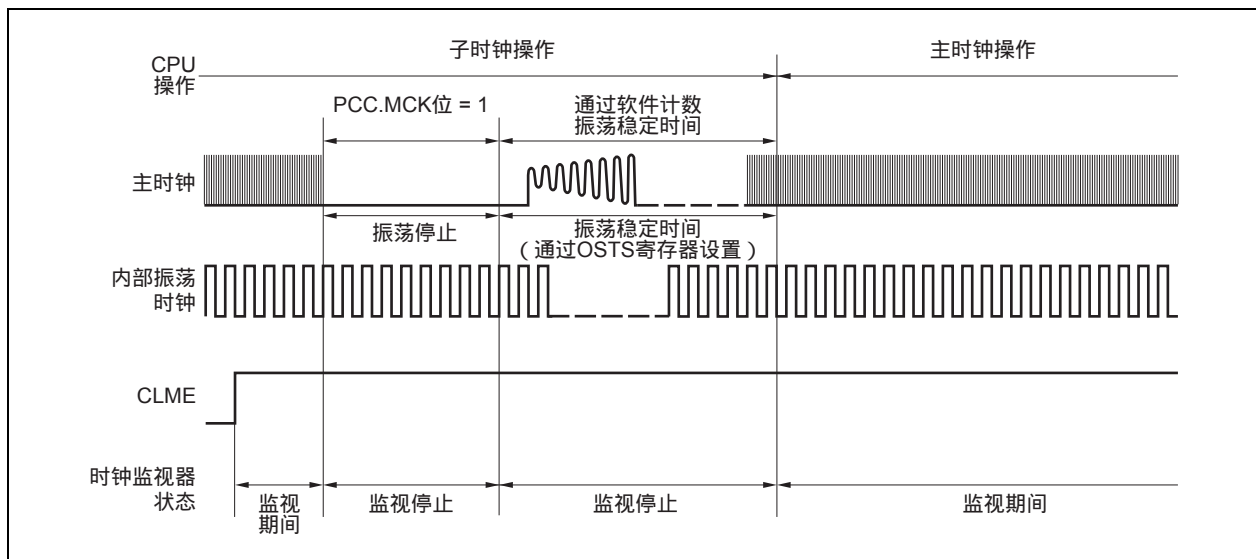
图 18-4. STOP 模式或 STOP 模式释放后的操作



(4) 主时钟停止时的操作（任意时刻）

副时钟操作期间（PCC.CLS 位=1）或者当通过 PCC.MCK 位置 1 来停止主时钟时，监控器操作将停止直到主时钟操作开始（PCC.CLS 位=0）为止。当主时钟操作开始时，监控器操作将自动开始。

图 18-5. 主时钟停止时的操作（任意时刻）



(5) 当 CPU 使用内部振荡时钟上操作（CCLS.CCLS F 位 = 1）的操作。

当 CCLS F 位设为 1 时，监控器的操作不会停止，即使 CLME 位被设为 1。

第 19 章 上电清零电路

19.1 功能

上电清零电路（POC）的功能如下所示。

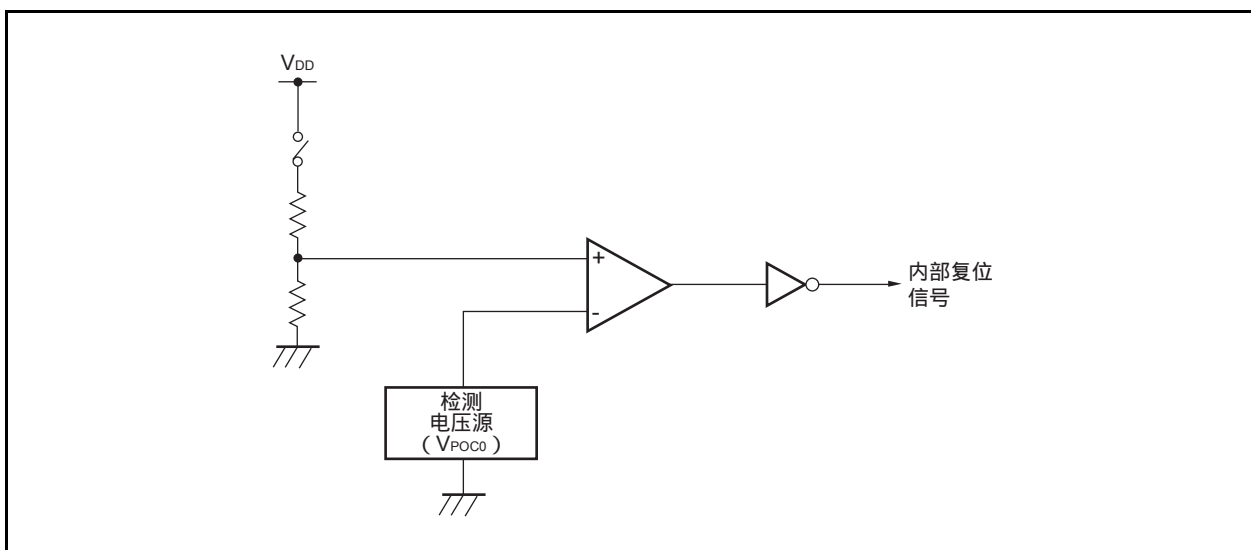
- 在电源打开的基础上生成一个复位信号。
- 将供电电压（ V_{DD} ）与检测电压（ V_{POC0} ）进行比较，并在 $V_{DD} < V_{POC0}$ （检测电压（ V_{POC0} ）： $3.7\text{ V} \pm 0.2\text{ V}$ ）时生成一个复位信号。

- 备注**
1. V850ES/HE2 具有生成内部复位信号的多元内部硬件单元。当系统通过看门狗定时器 2（WDT2RES），低压检测器（LVI）或时钟监控器（CLM）来复位时，相应于复位源的标志将会被分配到复位源标志寄存器（RESF）中。
当内部复位信号通过 WDT2RES，LVI 或时钟监控器来生成且它相应于复位源的标志被设为 1 时，RESF 寄存器将不会被清除。关于 RESF 寄存器的详细信息，参见第 17 章 复位功能。
 2. 如果外部连接的谐振器的操作频率是 5 MHz，那么从电源打开到开始执行程序间的时间等于“从电源打开到释放复位间的时间+ 16 ms”。然而，这个时间会根据外部原因而变化（例如微控制器工件电压的状态以及谐振器的稳定时间）。

19.2 配置

上电清零电路的框图如下所示。

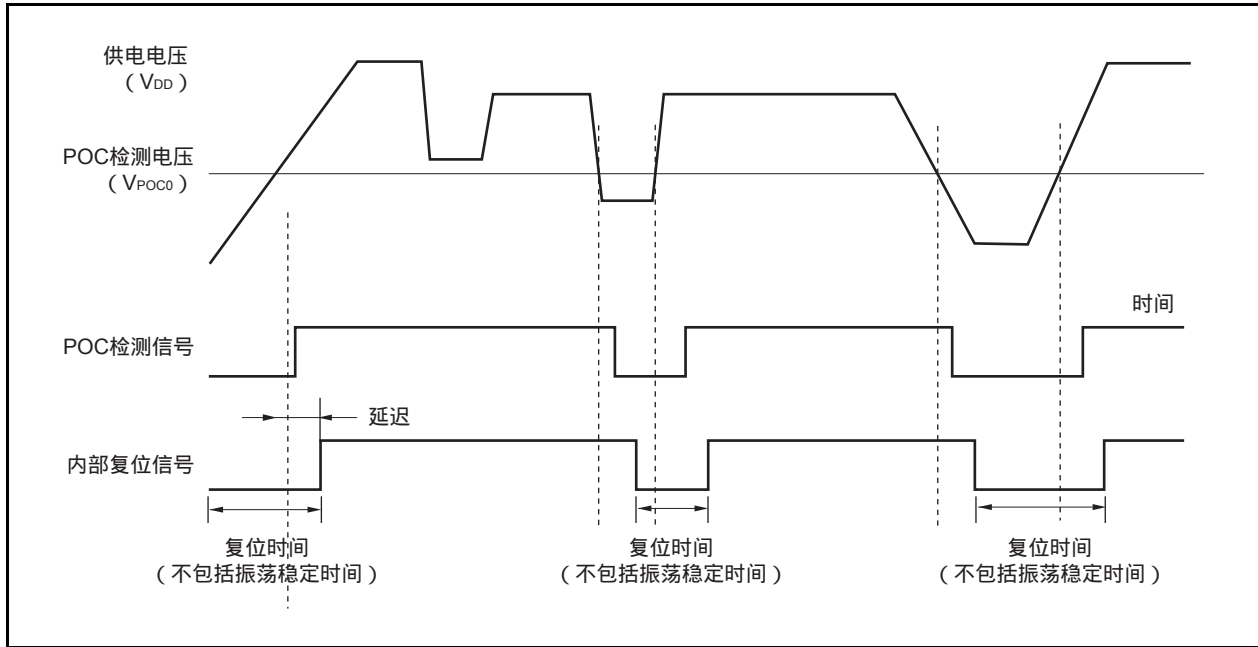
图 19-1. 上电清零电路的框图



19.3 操作

当供电电压与检测电压进行比较且供电电压低于检测电压（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。

图 19-2. 通过上电清零电路生成复位信号的时序



第 20 章 低电压监测器

20.1 功能

低电压监测器 (LVI) 具有以下功能。

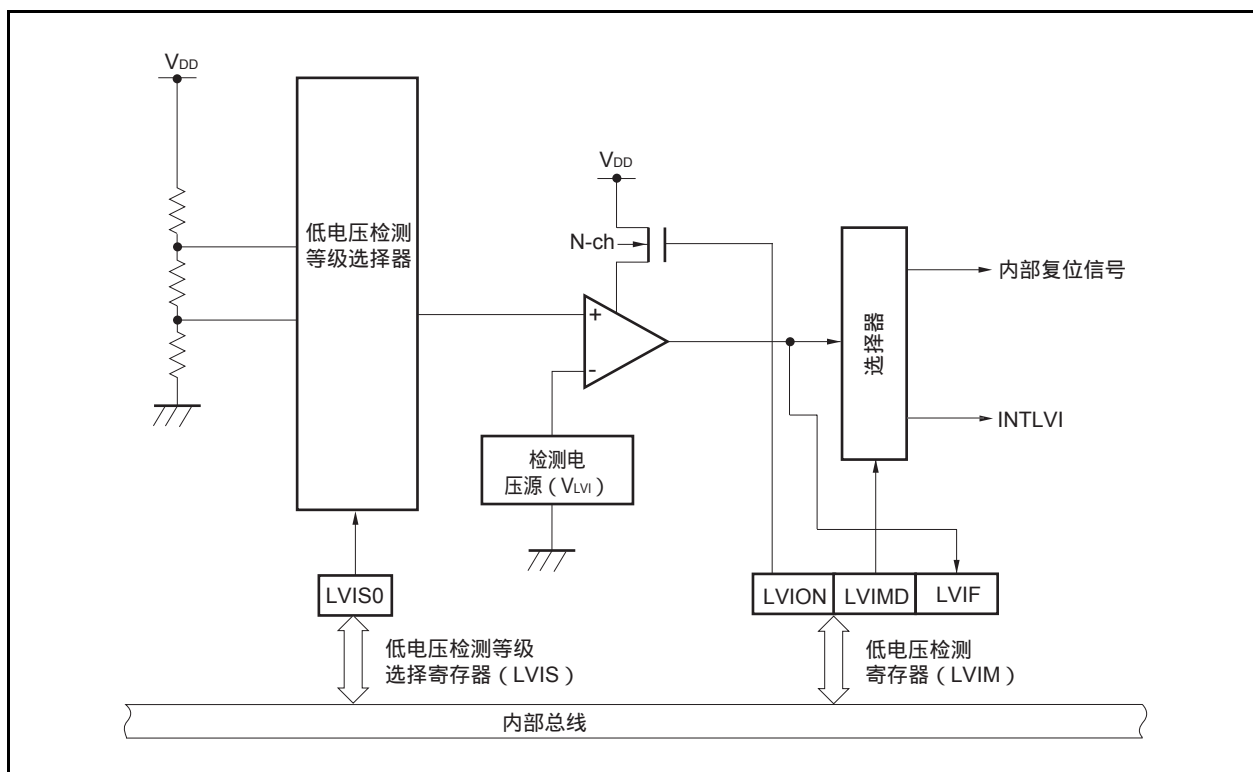
- 将工作电压 (V_{DD}) 与检测电压 (V_{LVI}) 进行比较, 并在 $V_{DD} < V_{LVI}$ 时生成一个中断请求信号或内部复位信号。
- 要检测的工作电压的电平可以通过软件来改变 (分为两个步骤)。
- 可以选择中断请求信号或内部复位信号。
- 可以在 STOP 模式中进行操作。
- 可以通过软件停止操作。

如果低电压监测器用于产生复位信号, 那么当复位信号产生时, 复位控制标志寄存器 RESF 的 LVIRF 位被置 1。要获取 RESF 寄存器的详细信息, 请参考第 17 章 复位功能。

20.2 配置

低电压监测器的结构图如下图所示。

图 20-1. 低电压监测器的结构图



20.3 寄存器

(1) 低电压监测器 (LVIM)

LVIM 寄存器用于允许或禁止低电压监测，并用于设置低电压监测器的操作模式。LVIM 寄存器是一个特殊的寄存器。它只有在特殊的顺序组合时才可以被写入（请参见 3.4.7 特殊寄存器）。该寄存器可以以 8 位或 1 位为单位来读取或写入。然而，位 0 为只读。

复位后: 00H		R/W		地址: FFFFF890H													
7		6		5		4		3		2		1		0			
LVIM		LVION		0		0		0		0		0		LVIMD		LVIF	

LVION		允许低电压检测操作													
0		禁止操作。													
1		允许操作。													

LVIMD		低电压检测操作模式的选择													
0		当供电电压 (V _{DD}) < 检测电压 (V _{LVI}) 时产生中断请求信号 INTLVI。													
1		当供电电压 (V _{DD}) < 检测电压 (V _{LVI}) 时产生内部复位信号 LVIRES。													

LVIF		低电压检测标志													
0		供电电压 (V _{DD}) > 检测电压 (V _{LVI})， 或当禁止操作时													
1		供电电压 (V _{DD}) < 检测电压 (V _{LVI})													

- 注意事项
1. 在将 LVION 位设为 1 后，在通过使用 LVIF 位检测电源前需等待 0.2 ms (MAX.)。
 2. 当 LVION 位= 1 且 LVIMD bit = 0 时，LVIF 标志的值将会被输出为输出信号 INTLVI。
 3. 确保将位 2 至位 6 清除为 “0”。
 4. 低电压监测器不能被停止，直到由于一些不同于低电压检测的复位请求在 LVIM.LVION 和 LVIM.LVIMD 位被设为 1 后被生成为止。

<R>

(2) 低电压检测等级选择寄存器 (LVIS)

LVIS 寄存器用于选择低电压检测等级。

该寄存器可以以 8 位为单位来读取或写入。

复位后: 00H	R/W	地址: FFFFF891H						
	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	0	0	LVIS0

LVIS0	检测等级
0	4.4 V \pm 0.2 V
1	4.2 V \pm 0.2 V

注意事项 1. LVION 和 LVIMD 位设为 1 后, 此寄存器不能写, 直到产生除 LVI 复位以外的复位请求时。

2. 一定要将位 7 至位 1 清除为“0”。

(3) 内部 RAM 数据状态寄存器 (RAMS)

RAMS 寄存器是一个用于显示内部 RAM 是否有效的标志寄存器。RAMS 寄存器是一个特殊寄存器。它只有在特殊的顺序组合时才可以进行写操作 (请看 3.4.7 特殊寄存器)。

关于 RAMS 寄存器的详细信息, 参见 20.5 保持电压检测操作。

此寄存器可以以 8 位或 1 位为单位来读取或写入。

注意事项 复位后以下显示了复位后的特殊顺序。

- 设置条件:
 - 低于检测电压的检测
 - 通过指令来设置
 - 通过看门狗定时器溢出生成复位信号
 - RAM 被访问时复位信号的生成
 - 通过时钟监控器生成复位信号
- 清除条件:
 - 以特定顺序写入 0

复位后: 01H	R/W	地址: FFFFF892H						
	7	6	5	4	3	2	1	0
RAMS	0	0	0	0	0	0	0	RAMF

RAMF	内部有效 RAM/内部无效 RAM
0	有效
1	无效

20.4 操作

依据 LVIM.VIMD 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。

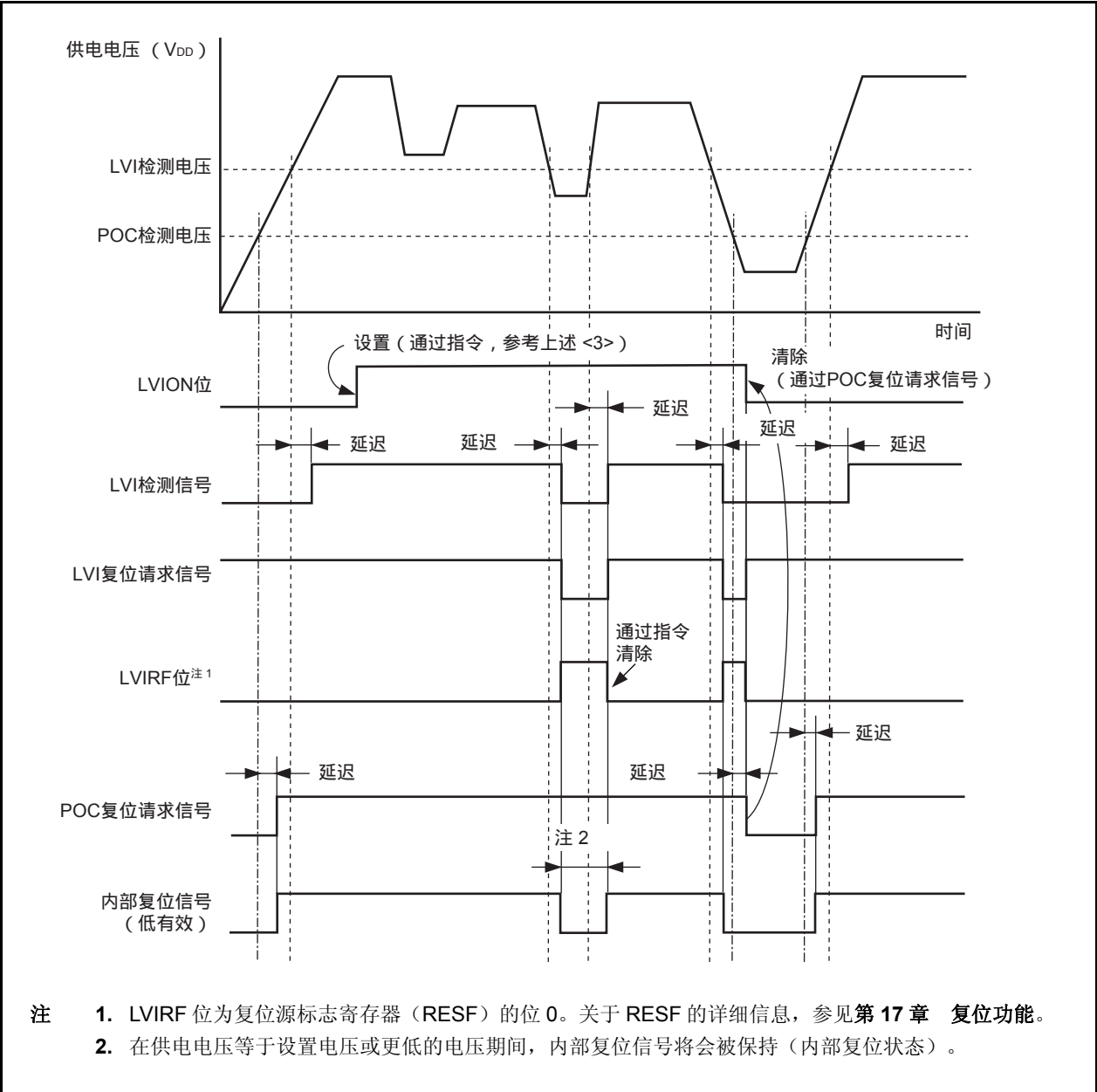
20.4.1 用作内部复位信号

<开始操作>

- <1> 屏蔽 LVI 中断。
- <2> 使用 LVIS.LVIS0 位选择检测的电压。
- <3> 设置 LVIM.LVION 位为 1（允许操作）。
- <4> 由软件插入 0.2 ms（max.）的等待周期。
- <5> 使用 LVIM.LVIF 位，检验供电电源电压是否大于检测电压。
- <6> 设置 LVIM.LVIMD 位为 1（为产生内部复位信号）。

注意事项 如果 LVIMD 位设置为 1，那么 LVIM 和 LVIS 寄存器的内容不能改变，直到 LVI 之外的复位请求产生。

图 20-2. 低电压检测电路的操作时序 (LVIMD 位= 1)



20.4.2 用作中断

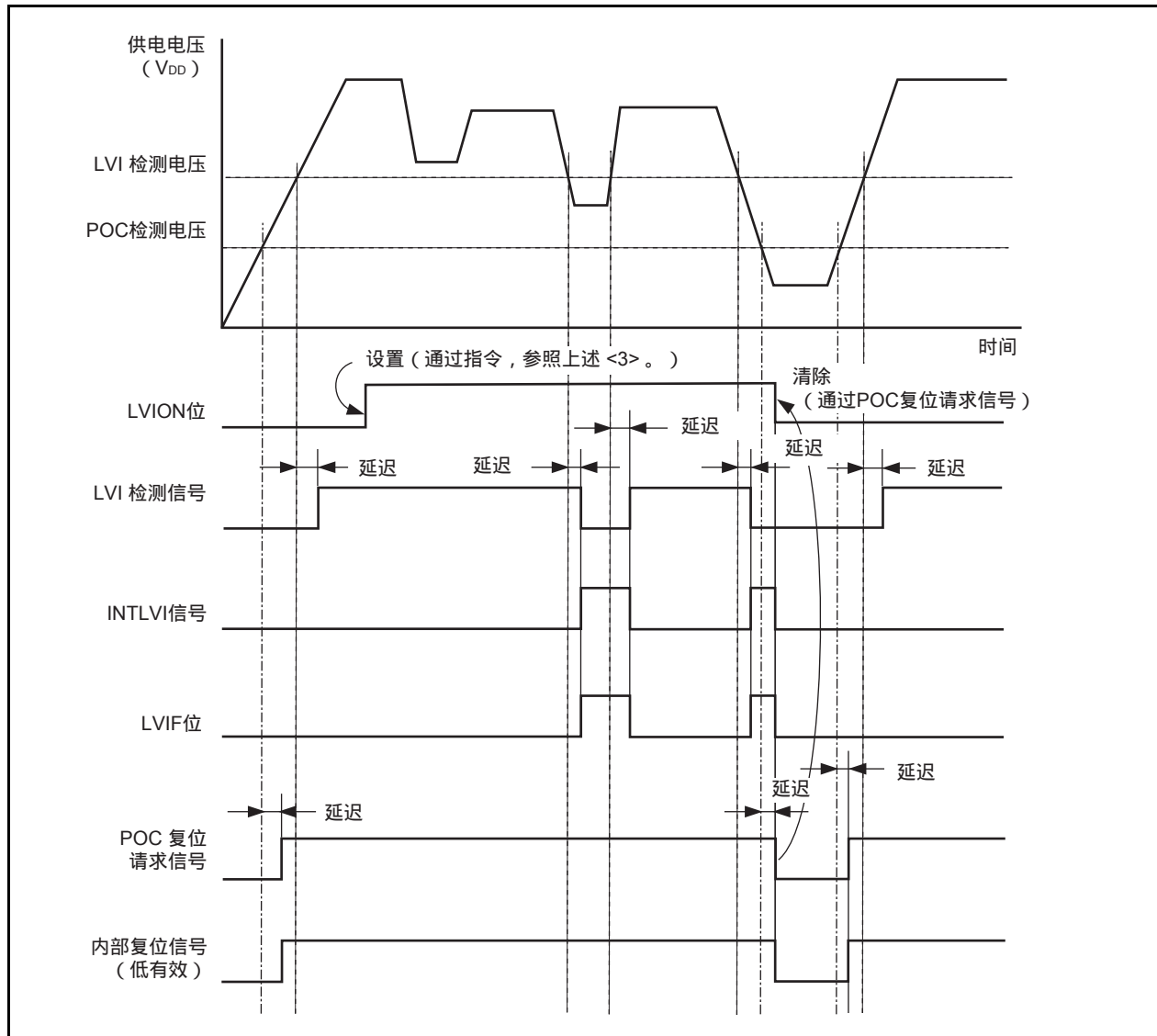
<开始操作>

- <1> 屏蔽 LVI 中断。
- <2> 由 LVIS.LVIS0 位选择检测的电压。
- <3> 设置 LVIM.LVION 位为 1（允许操作）。
- <4> 由软件插入一个 0.2 ms（max.）的等待周期。
- <5> 使用 LVIM.LVIF 位，检验供电电源电压是否大于检测电压。
- <6> 清除 LVI 的中断请求标志。
- <7> 打开 LVI 中断。

<停止操作>

将 LVION 位清 0。

图 20-3. 低电压检测电路的操作时序（LVIM 位 = 0）

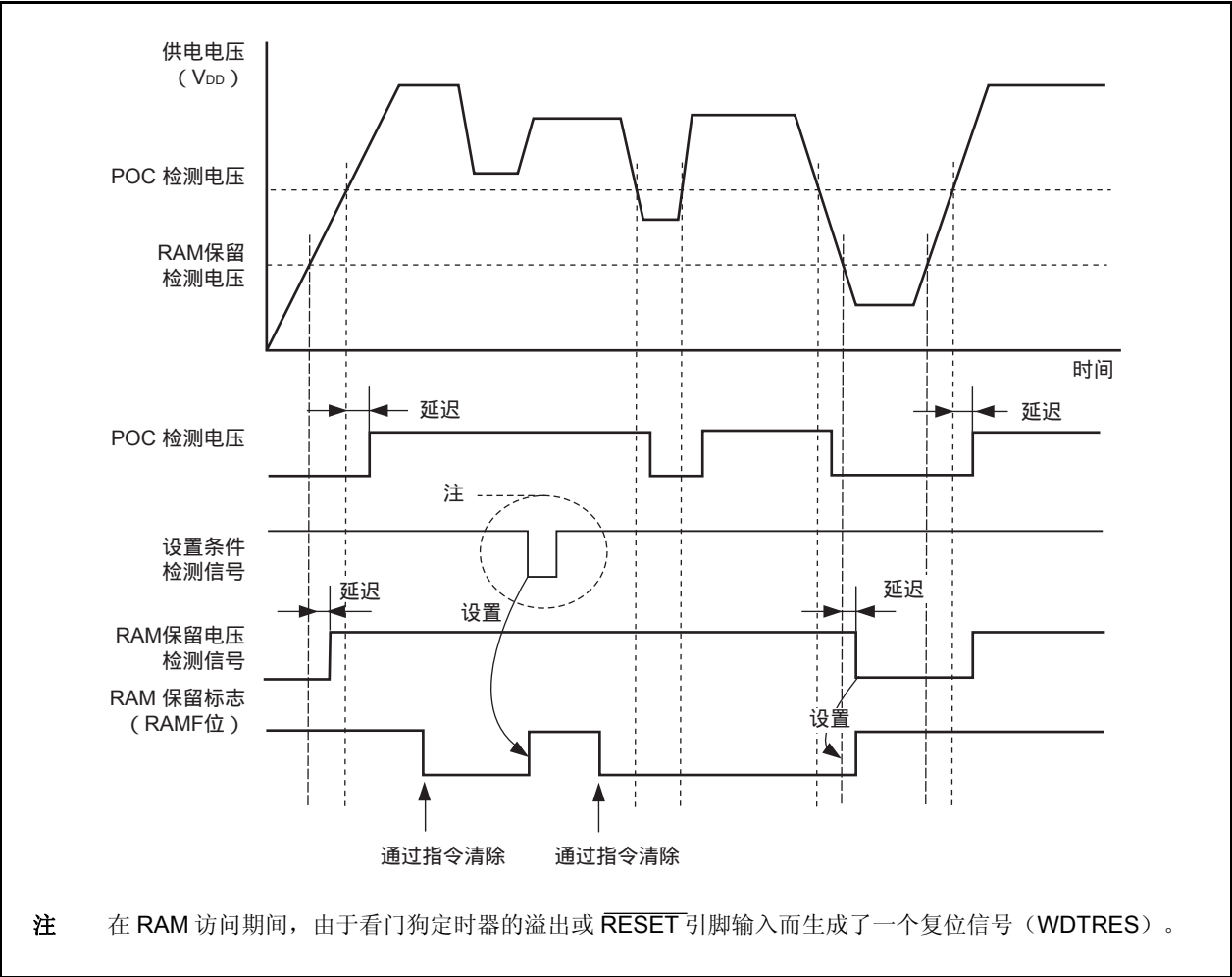


20.5 RAM 保持电压检测操作

比较供电电压和检测电压。当其低于检测电压时（包括上电），RAMS.RAMF 位设置为 1。

当 POC 功能没有并使用而 RAM 保持电压检测功能在使用时，如果检测电压低于操作电压，则一定要输入一个外部复位信号。

图 20-4. RAM 保持电压检测功能的操作时序



20.6 仿真功能

当使用在线仿真器时，RAM 保持标志（RAMS.RAMF 位）的操作可以通过操作调试器上的 PEMU1 寄存器进行伪控制 和仿真。

此寄存器只有在仿真模式下有效，正常模式下无效。

(1) 外围仿真寄存器 1（PEMU1）

复位后：00H

R/W

地址：FFFFFF9FEH

	7	6	5	4	3	2	1	0
PEMU1	0	0	0	0	0	EVARAMIN	0	0

EVARAMIN	RAM 保持电压检测信号的假定
0	不检测低于 RAM 保持电压的电压。
1	检测低于 RAM 保持电压的电压（设置 RAMF 标志）。

注意事项 该位不会自动清除。

[用法]

当使用线上仿真器时，RAMF 的伪仿真通过重写调试器上的寄存器实现。

- <1> CPU 停止（CPU 操作停止）。
- <2> 通过使用寄存器写命令设置 EVARAMIN 位为 1。
通过设置 EVARAMIN 位为 1，硬件上的 RAMF 位设置为 1。（内部 RAM 数据无效）。
- <3> 再次通过寄存器写命令将 EVARAMIN 位清零。
除非在进行此操作（清 EVARAMIN 位为 0），否则 RAMF 位不能由 CPU 操作指令清 0。
- <4> 运行 CPU 并重新仿真。

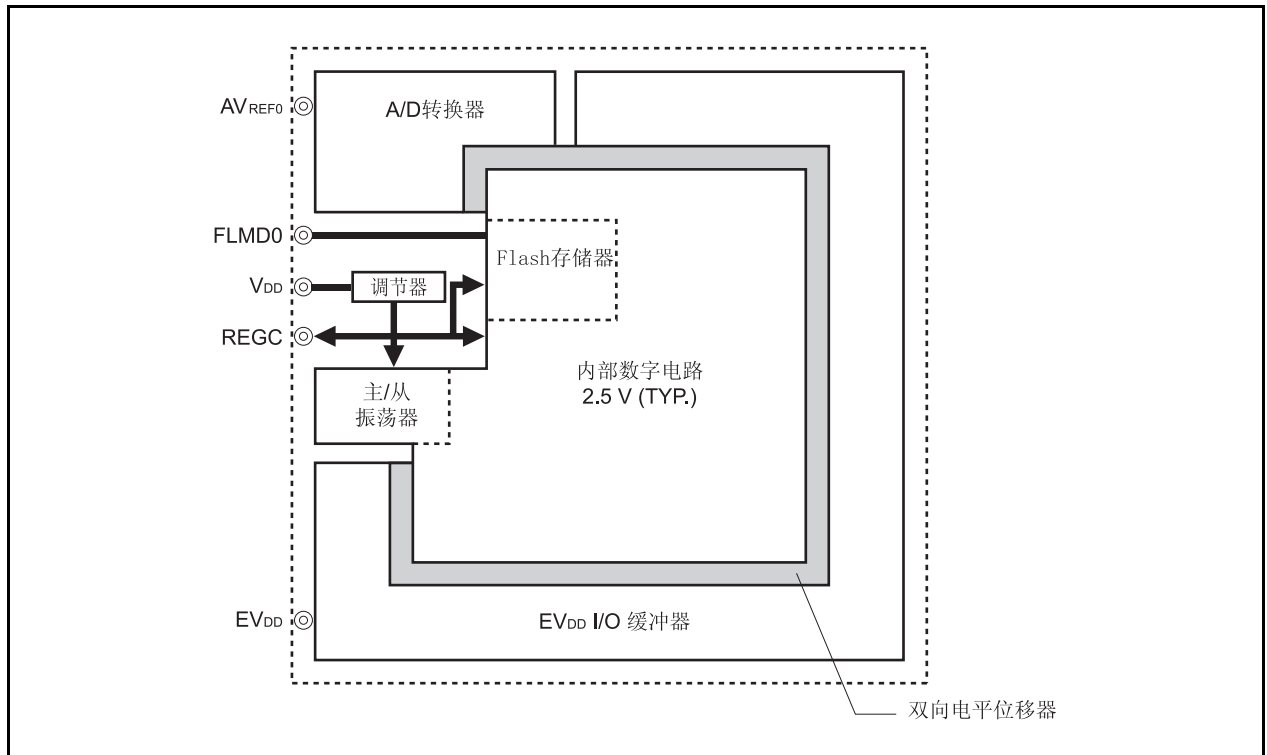
第 21 章 调节器

21.1 概述

V850ES/HE2 包括一个可以降低能耗和噪声的调节器。

该调节器为振荡器模块提供一个分阶下降的 V_{DD} 电源以及内部逻辑电路（除了 A/D 转换器和输出缓冲器）。调节器输出电压设置为 2.5 V (TYP.)。

图 21-1. 调节器



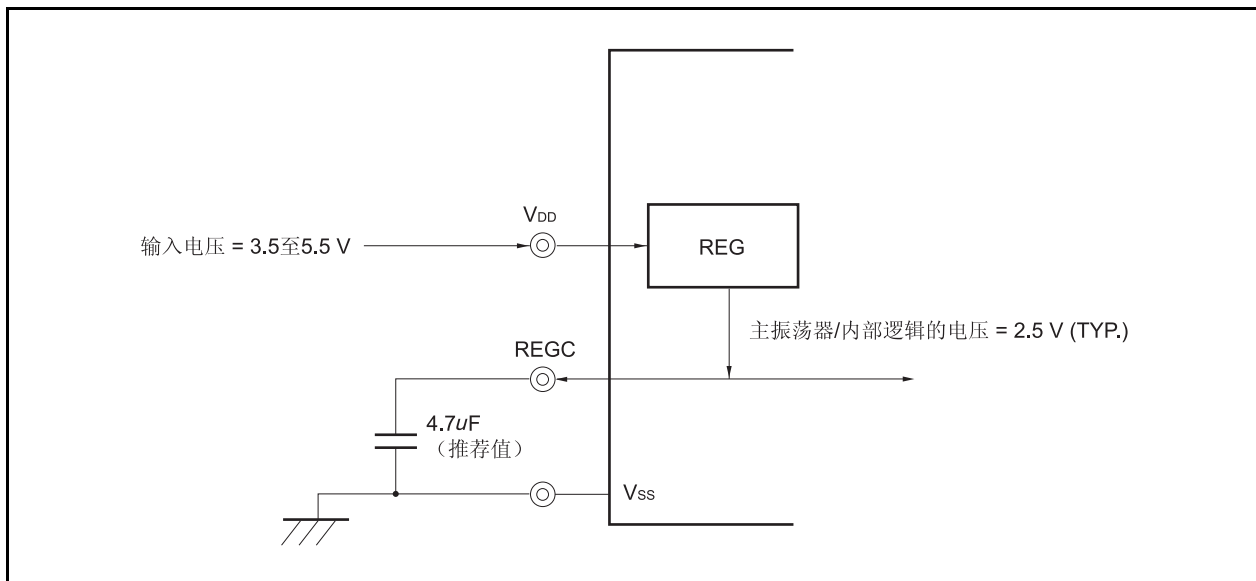
21.2 操作

该产品的调节器总是运行于各种模式下（正常操作模式, HALT 模式, IDLE1 模式, IDLE2 模式, STOP 模式或者复位时）。

请确保给 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。

调节器引脚连接方法图如下所示。

图 21-2. REGC 引脚连接



第 22 章 Flash 存储器

通过使用 Flash 存储器版本，以下内容可以被认为是开发环境和大批量生产应用。

- 在 V850ES/HE2 被安装到目标系统之后可以对软件进行更改
- 量产后的数据调整。
- 在不同模式的小批量生产中依照说明书区分软件。
- 简化存货管理。
- 出库后的软件升级

22.1 特性

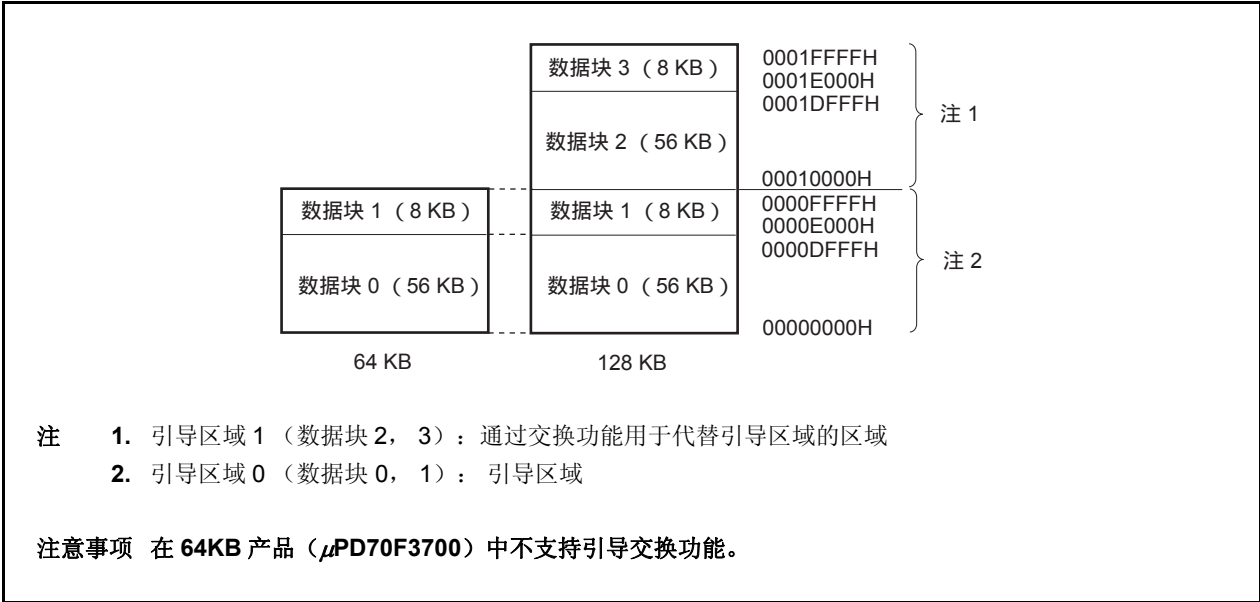
- 4 字节 /1 个时钟访问（当指令取得时）
- 容量：128 KB/64 KB
- 写入电压：独立电源擦除/写入
- 重写方法
 - 通过专用 Flash 存储器 写入器的串行接口进行通信重写（在线/离线编程）
 - 通过用户程序重写 Flash 存储器（自编程）
- 支持 Flash 存储器禁止写入功能（安全功能）
- 由使用引导交换功能的自编程安全重写整个 Flash 存储器区
- 自编程期间可以确认中断。

<R> 22.2 存储器配置

128K 及 64K 内部 Flash 存储器区域分别被分为 4 个及 2 个数据块且可以以数据块为单位进行编程/擦除。

使用引导交换功能时，在引导区域 0 的地址处的物理内存（数据块 0，1）会被处于引导区域 1 的地址处的物理内存（数据块 2，3）所替代。关于引导交换功能的详细信息，参考 22.5 通过自编程重写。

图 22-1. Flash 存储器映射



<R> 22.3 功能概述

V850ES/HE2 的内部 Flash 存储器可使用专用 Flash 存储器编程器的写入功能写入，不论 V850ES/HE2 是否已经被安装到目标系统（离线/在线编程）。

另外，也支持禁止将用户程序写入到内部 Flash 存储器的安全功能，因此程序不能由未经授权的人修改。

使用用户程序（自编程）的重写功能对于那些产品出库后有可能发生改变的目标系统中的应用比较理想。也支持安全重写整个 Flash 存储器的引导交换功能。另外，自编程期间支持中断服务，因此 Flash 存储器可以在不同的条件下写入，例如当与外围设备通讯时。

表 22-1. 重写方法

重写方法	功能概述	操作模式
在线编程	安装到目标系统后通过使用专用 Flash 存储器编程器，Flash 存储器可以被重写。	Flash 存储器编程模式
离线编程	在安装到目标系统之前，可使用专用 Flash 存储器编程器和专用程序适配器（FA 系列）将数据写入 Flash 存储器中。	
自编程	Flash 存储器可以通过执行用在线/离线编程方法预先写入到 Flash 存储器存储器中的用户程序进行重写。（自编程期间，不能获取指令且不能访问内部 Flash 存储器存储器区域的数据。因此，写入程序必须预先发送到内部 RAM 或外部存储器。）	正常操作模式

备注 FA 系列是 Naito Densei Machida Mfg. Co., Ltd.的产品。

表 22-2. 基本功能

功能	功能概述	支持（○：支持的，×：不支持的）	
		在线/离线编程	自编程
数据块的擦除	擦除指定的存储器数据块的内容。	○	○
芯片擦除	迅速擦除整个存储器区域中的内容。	○	×
写入	写入到指定地址，并且校验检查写入级别是否安全地完成。	○	○
校验/校验和	比较 Flash 存储器中读出的数据与从 Flash 存储器编程器发送的数据。	○	× (可由用户程序读取)
空白检测	检查整个存储器的擦除状态。	○	○
安全设置	数据块擦除命令，片擦除命令，程序命令和读取命令可以被禁止。	○	× (仅在设置从允许改为禁止时是支持的。)

下表列出了安全功能。数据块擦除命令禁止功能，芯片擦除命令禁止功能以及编程命令禁止功能在出库时默认为允许，通过在线/离线编程重写来进行安全设置。每种安全功能可以与其他安全功能同时使用。

表 22-3. 安全功能

功能	功能概述
数据块擦除命令禁止	所有数据块禁止执行数据块擦除命令。禁止设置可以通过执行片擦除命令来初始化。
芯片擦除命令禁止	禁止执行数据块擦除和芯片擦除命令。一旦禁止被设置，由于芯片擦除命令不能执行因此禁止不能被初始化。
程序命令禁止	所有的数据块都禁止执行程序和数据擦除命令。禁止设置可以通过执行片擦除命令来初始化。
读取命令禁止	所有的数据块都禁止执行读取命令。禁止设置可以通过执行片擦除命令来初始化。
重写引导区域禁止	不支持。

表 22-4. 安全设置

功能	进行每种安全设置时的擦除，写入，读取操作 (√: 可执行, x: 不可执行, -: 不支持)		安全设置时的注意事项	
	在线/ 离线编程	自编程	在线/ 离线编程	自编程
数据块擦除命令禁止	数据块擦除命令: x 芯片擦除命令: √ 编程命令: √ 读取命令: √	数据块擦除 (Flash 存储器数据块擦除): √ 芯片擦除: - 写入 (Flash 存储器字写入): √ 读取 (Flash 存储器字读取): √	禁止设置可以通过芯片擦除命令来初始化。	保有当设置从允许更改为禁止时是支持的。
芯片擦除命令禁止	数据块擦除命令: x 芯片擦除命令: x 编程命令: √ 读取命令: √	数据块擦除 (Flash 存储器数据块擦除): √ 芯片擦除: - 写入 (Flash 存储器字写入): √ 读取 (Flash 存储器字读取): √	禁止设置不能被初始化。	
编程命令禁止	数据块擦除命令: x 芯片擦除命令: √ 编程命令: x 读取命令: √	数据块擦除 (Flash 存储器数据块擦除): √ 芯片擦除: - 写入 (Flash 存储器字写入): √ 读取 (Flash 存储器字读取): √	禁止设置可以通过芯片擦除命令来初始化。	
读取命令禁止	数据块擦除命令: √ 芯片擦除命令: √ 编程命令: √ 读取命令: x	数据块擦除 (Flash 存储器数据块擦除): √ 芯片擦除: - 写入 (Flash 存储器字写入): √ 读取 (Flash 存储器字读取): √	禁止设置可以通过芯片擦除命令来初始化。	

注 在这种情况下，由于擦除命令无效，因此不能写入与已写入 Flash 存储器中数据的数据。

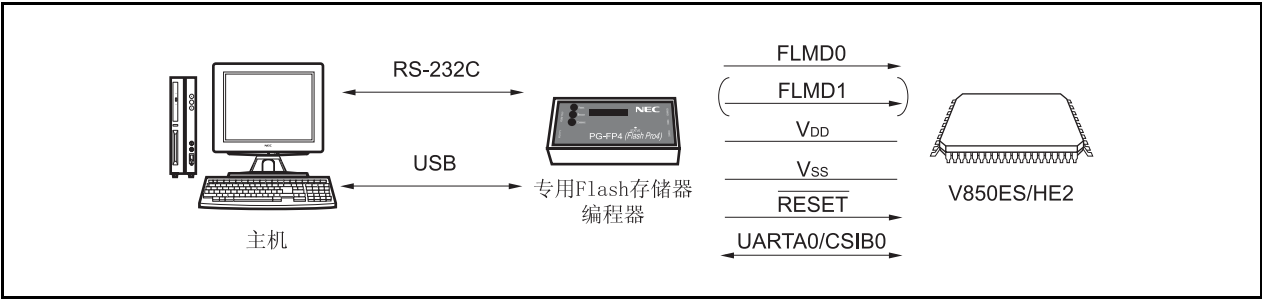
22.4 通过专用 Flash 存储器编程器重写

在 V850ES/HE2 安装到目标系统（在线编程）后可以通过专用 Flash 存储器编程器重写 Flash 存储器。在设备安装到目标系统（离线编程）前也可以通过专用程序适配器（FA 系列）来重写 Flash 存储器。

22.4.1 编程环境

将程序写入 V850ES/HE2 的 Flash 存储器所需的环境如下所示。

图 22-2. 写程序到 Flash 存储器所需要的环境



为控制专用 Flash 存储器编程器需要一个主机。

UARTA0 或 CSIB0 作为专用 Flash 存储器编程器和 V850ES/HE2 的接口执行写入，擦除等。专用程序适配器（FA 系列）需要离线写入。

- FA-70F3701GB-8EU-MX （已经连线的）
- FA-64GB-8EU-A （未连线的：需要写入）

备注 FA 系列是 Naito Densai Machida Mfg. Co., Ltd 的产品

22.4.2 通信模式

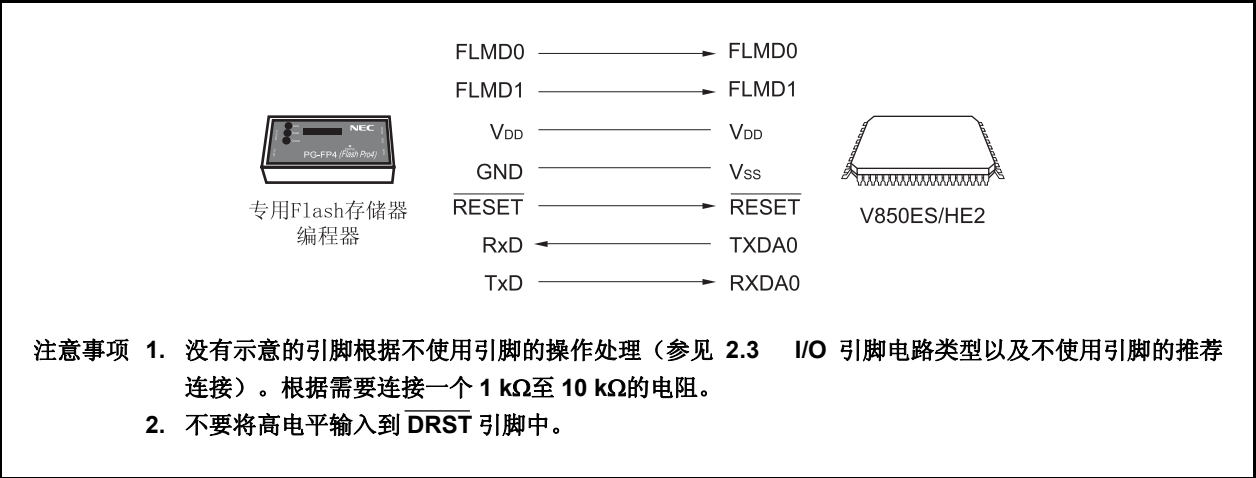
V850ES/HE2 与专用 Flash 存储器编程器间的通讯通过 V850ES/HE2 的串行接口 UARTA0 或 CSIB0 完成。

(1) UARTA0

<R>

发送速率： 9600， 19200， 31250， 38400， 76800， 153600 bps
(不支持 57600， 115200 以及 128000 bps。)

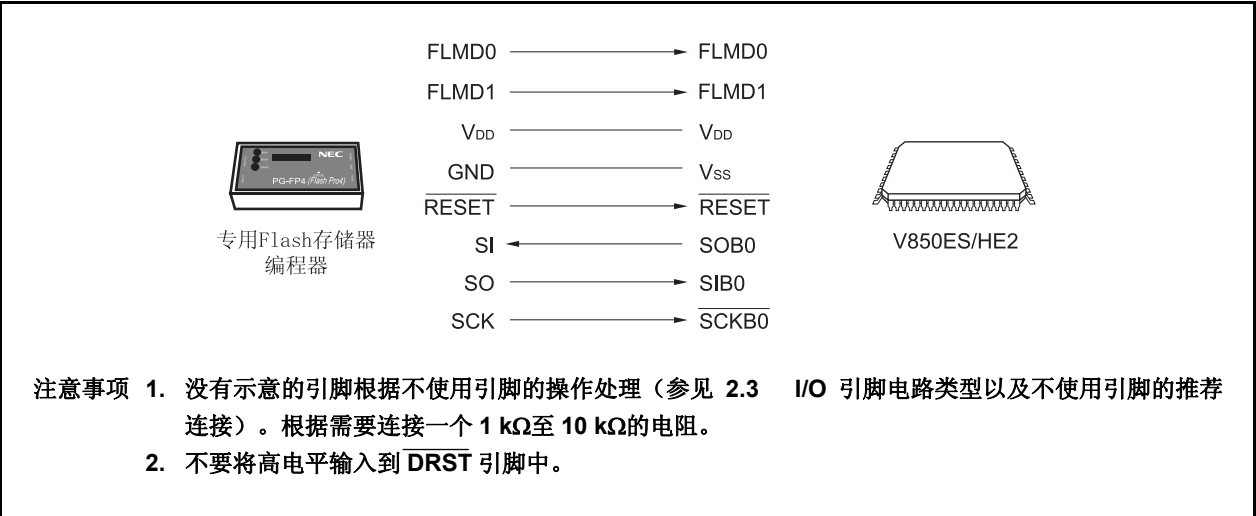
图 22-3. 与专用 Flash 存储器编程器通信 (UARTA0)



(2) CSIB0

串行时钟： 2.4 kHz 至 2.5 MHz （MSB 优先）

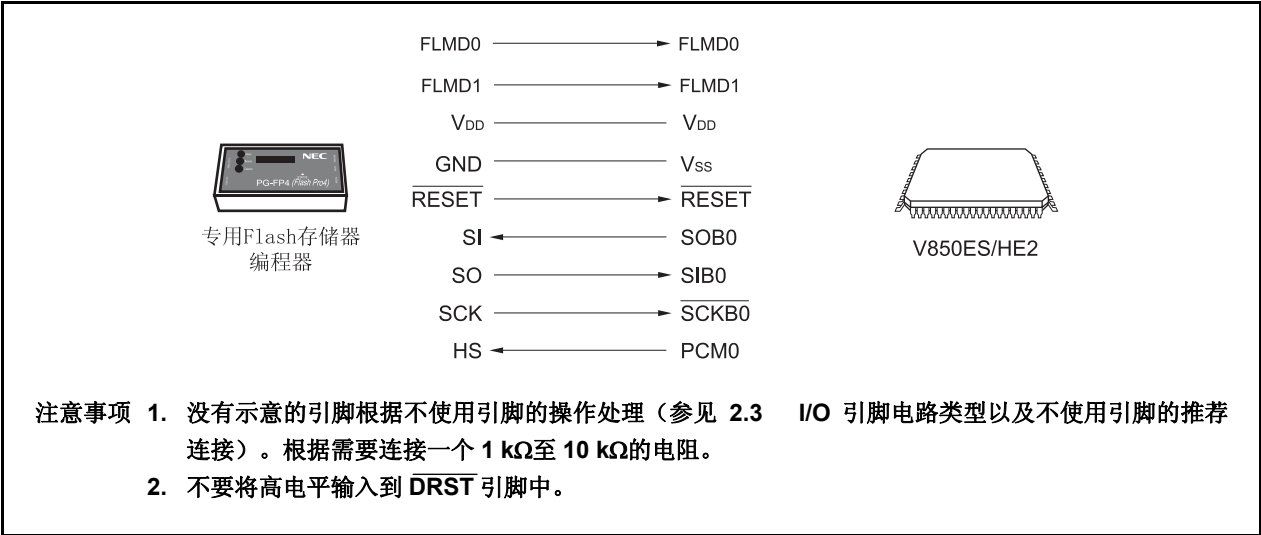
图 22-4. 与专用 Flash 存储器编程器通信 (CSIB0)



(3) CSIB0 + HS

串行时钟： 2.4 kHz 至 2.5 MHz（MSB 优先）

图 22-5. 与专用 Flash 存储器编程器通信（CSIB0 + HS）



专用 Flash 存储器编程器输出发送时钟，V850ES/HE2 作为从设备操作。

当 PG-FP4 用作专用 Flash 存储器编程器时，产生如下信号到 V850ES/ HE2。详细信息， 参见 PG-FP4 用户手册（U15260E）。

表 22-5. 专用 Flash 存储器编程器的信号连接（PG-FP4）

PG-FP4			V850ES/HE2	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} 电压产生/电压监测	V _{DD}	○	○	○
GND	—	接地	V _{SS}	○	○	○
CLK	输出	时钟输出到 V850ES/HE2	X1, X2	× ^{注 2}	× ^{注 2}	× ^{注 2}
RESET	输出	复位信号	RESET	○	○	○
SI/RxD	输入	接收信号	SOB0, TXDA0	○	○	○
SO/TxD	输出	发送信号	SIB0, RXDA0	○	○	○
SCK	输出	发送时钟	SCKB0	×	○	○
HS	输入	给 CSIB0 + HS 握手信号通信	PCM0	×	×	○

- 注
1. 按图 22-6 中所示为这些引脚布线，或在板上经下拉电阻接 GND。
 2. 不能由 Flash 存储器编程器的 CLK 引脚提供时钟。以板上创建振荡器并提供时钟。

备注

○： 必须连接。

×： 不需要连接。

表 22-6. V850ES/HE2 的 Flash 存储器写入适配器的连接 (FA-64GB-8EU-A)

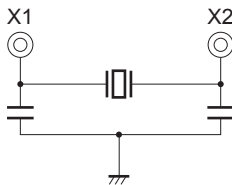
Flash 存储器编程器 (PG-FP4) 连接引脚			FA 板上的 引脚名	当使用 CSIB0 + HS 时		当使用 CSIB0 时		当使用 UARTA0 时	
信号名	I/O	引脚功能		引脚名	引脚 编号	引脚名	引脚 编号	引脚名	引脚 编号
SI/RxD	输入	接收信号	SI	P41/SOB0	20	P41/SOB0	20	P30/TXDA0	22
SO/TxD	输出	发送信号	SO	P40/SIB0	19	P40/SIB0	19	P31/RXDA0/INTP7	23
SCK	输出	发送时钟	SCK	P42/ $\overline{\text{SCKB0}}$	21	P42/ $\overline{\text{SCKB0}}$	21	不必要	—
CLK	输出	V850ES/HE2 的时钟	X1	不必要	—	不必要	—	不必要	—
			X2	不必要	—	不必要	—	不必要	—
/RESET	输出	复位信号	/RESET	$\overline{\text{RESET}}$	9	$\overline{\text{RESET}}$	9	$\overline{\text{RESET}}$	9
FLMD0	输出	写入电压	FLMD0	FLMD0	3	FLMD0	3	FLMD0	3
FLMD1	输出	写入电压	FLMD1	PDL5/FLMD1	52	PDL5/FLMD1	52	PDL5/FLMD1	52
HS	输入	CSI0+ HS 通信 的握手信号	RESERVE /HS	PCM0	45	不必要	—	不必要	—
VDD	—	VDD 电压产生/ 电压监视器	VDD	V _{DD}	4	V _{DD}	4	V _{DD}	4
				EV _{DD}	33	EV _{DD}	33	EV _{DD}	33
				AV _{REF0}	1	AV _{REF0}	1	AV _{REF0}	1
GND	—	接地	GND	V _{SS}	6	V _{SS}	6	V _{SS}	6
				AV _{SS}	2	AV _{SS}	2	AV _{SS}	2
				EV _{SS}	32	EV _{SS}	32	EV _{SS}	32

- 注意事项 1. 确保 REGC 引脚经 4.7 μ F (推荐值) 电容器接地。
2. 时钟不能由 Flash 存储器编程器的 CLK 引脚提供。在板上创建一个振荡器并从振荡器中提供时钟。

图 22-6. V850ES/HE2 Flash 存储器写入适配器连接示例 (FA-64GB-8EU-A)
(在 CSIB0 + HS 模式中) (2/2)

- 注
1. 按下面所示连接 FLMD1 引脚，或通过下拉电阻将其连接到 GND。
 2. 使用 UARTA0 时所使用的引脚。
 3. 通过在 Flash 存储器写入适配器（用虚线围绕的）上创建一个振荡器来提供一个时钟。这是一个振荡器示例。

示例



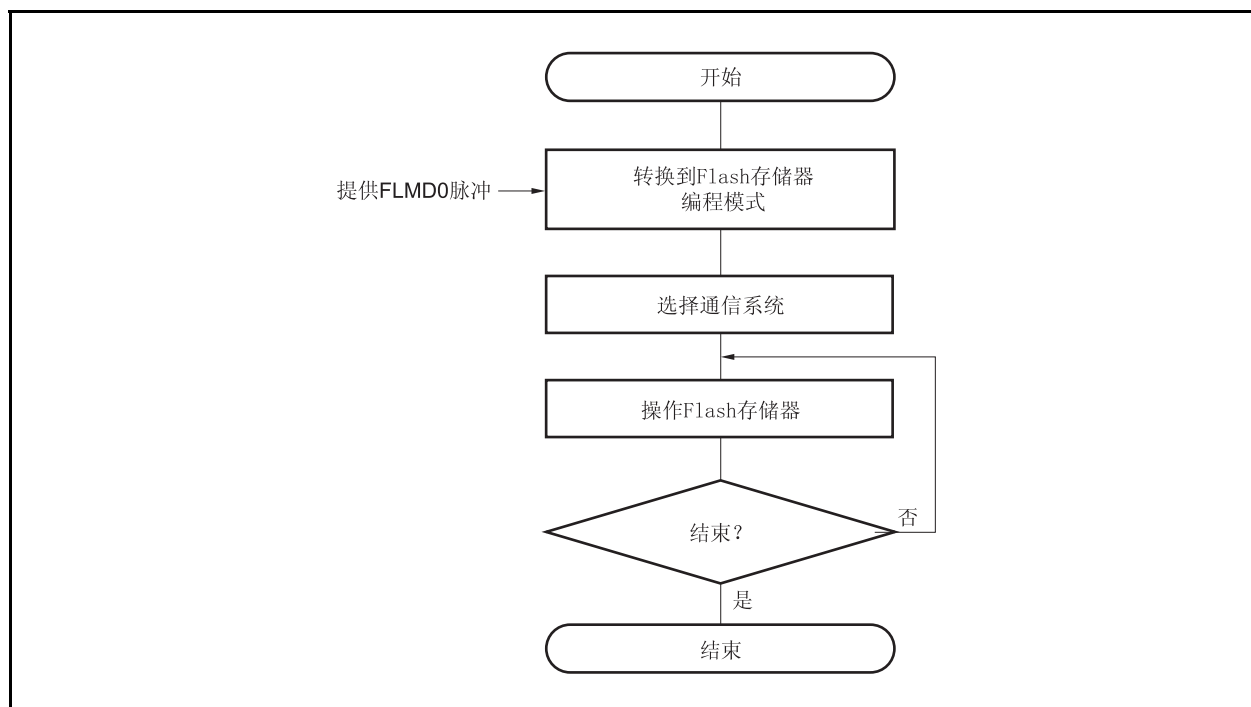
注意事项 不要将高电平输入到 $\overline{\text{DRST}}$ 引脚中。

- 备注
1. 没有示意的引脚根据不使用引脚的操作处理（参见 2.3 I/O 引脚电路类型以及不使用引脚的推荐连接）。
 2. 该适配器用于 64 引脚塑封 LQFP 封装。

22.4.3 Flash 存储器控制

下图显示了 Flash 存储器操作过程。

图 22-7. Flash 存储器操作过程

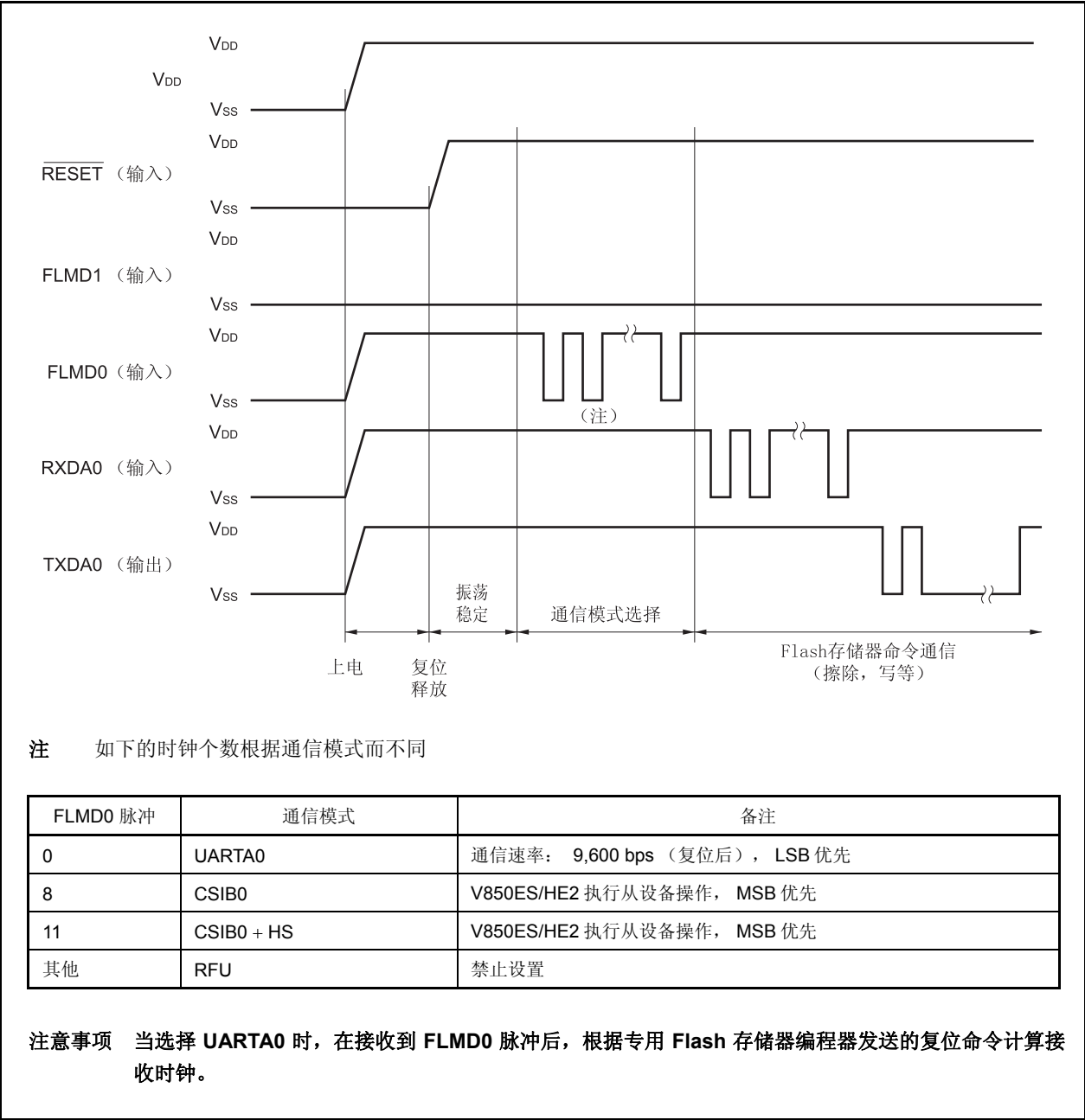


22.4.4 通讯模式的选择

在 V850ES/HE2 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 存储器编程器产生这些 FLMD0 脉冲。

下图显示了脉冲个数与通信模式之间的关系。

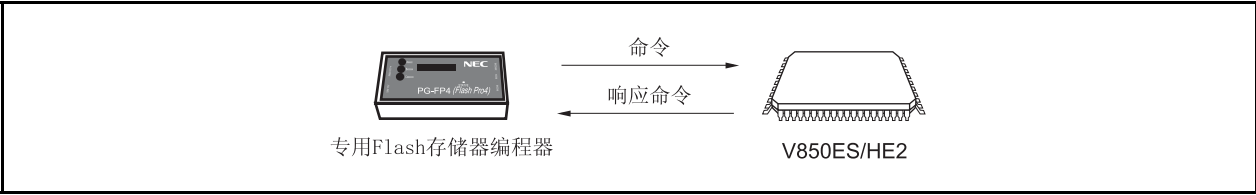
图 22-8. 通讯模式的选择



22.4.5 通信命令

V850ES/HE2 使用命令与专用 Flash 存储器编程器进行通信。从 Flash 存储器编程器发往 V850ES/HE2 的信号称为“命令”，从 V850ES/HE2 发往专用 Flash 存储器编程器的命令称为“响应命令”。

图 22-9. 通信命令



下面所示为在 V850ES/HE2 中 Flash 存储器的控制命令。所有的这些命令从专用 Flash 存储器编程器发出，V850ES/HE2 根据这些命令运行处理。

表 22-7. Flash 存储器控制命令

类别	命令名称	支持			功能
		CSIB0	CSIB0 + HS	UARTA0	
空白检测	数据块空白检测命令	√	√	√	检查存储器指定数据块是否被正确地擦除。
擦除	芯片擦除命令	√	√	√	擦除整个存储器的内容。
	数据块擦除命令	√	√	√	擦除存储器指定数据块的内容。
写入	编程命令	√	√	√	写入指定地址范围，并执行内容校验检查。
校验	校验命令	√	√	√	比较存储器指定地址范围的内容与 Flash 存储器编程器发送的数据。
	校验和命令	√	√	√	在指定地址范围读校验和。
系统设置，控制	硅标记命令	√	√	√	读取硅标记信息。
	安全设置命令	√	√	√	禁止芯片擦除命令，数据块擦除命令，编程命令和读取命令。

22.4.6 引脚连接

进行在线写入操作时，必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，也需带有在线功能用来实现从正常操作模式到 Flash 存储器编程模式的转换。

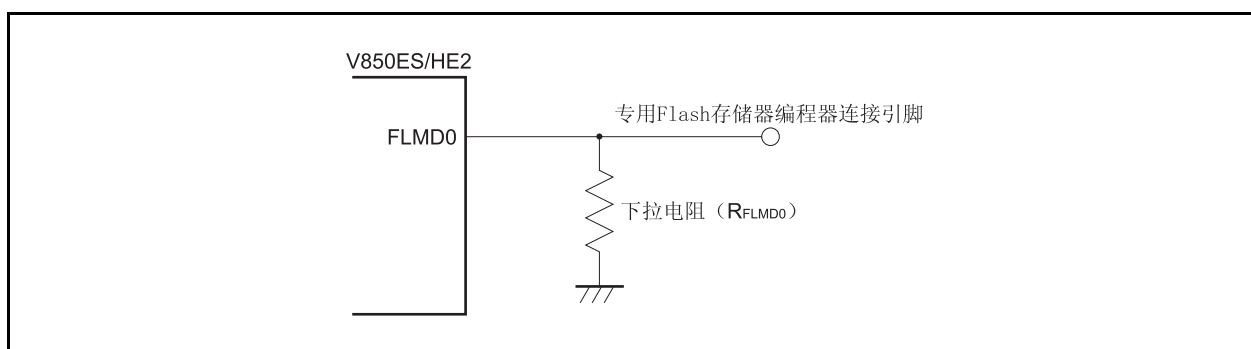
在 Flash 存储器编程模式，所有那些不用于 Flash 存储器存储器编程的引脚状态与复位后的状态相同。因此如果外部设备不能立即识别复位后的状态，则必须采用如下方式连接引脚。

(1) FLMD0 引脚

在普通工作模式下，输入一个 V_{SS} 电平的电压到 FLMD0 引脚。在 Flash 存储器存储器编程模式中，将 V_{DD} 电平的写电压提供给 FLMD0 引脚。

因为在自编程模式下 FLMD0 引脚用作写保护引脚，因此在写入 Flash 存储器前， V_{DD} 电平的电压必须经过端口控制供给 FLMD0 引脚，等。详细信息，参见 22.5.5 (1) FLMD0 引脚。

图 22-10. FLMD0 引脚连接示例



(2) FLMD1 引脚

当 FLMD0 引脚输入 0 V，FLMD1 引脚不起作用。当 V_{DD} 供给 FLMD0 引脚，会进入 Flash 存储器编程模式，因此 0 V 必须输入到 FLMD1 引脚。FLMD1 引脚的连接示意图如下所示。

图 22-11. FLMD1 引脚连接示例

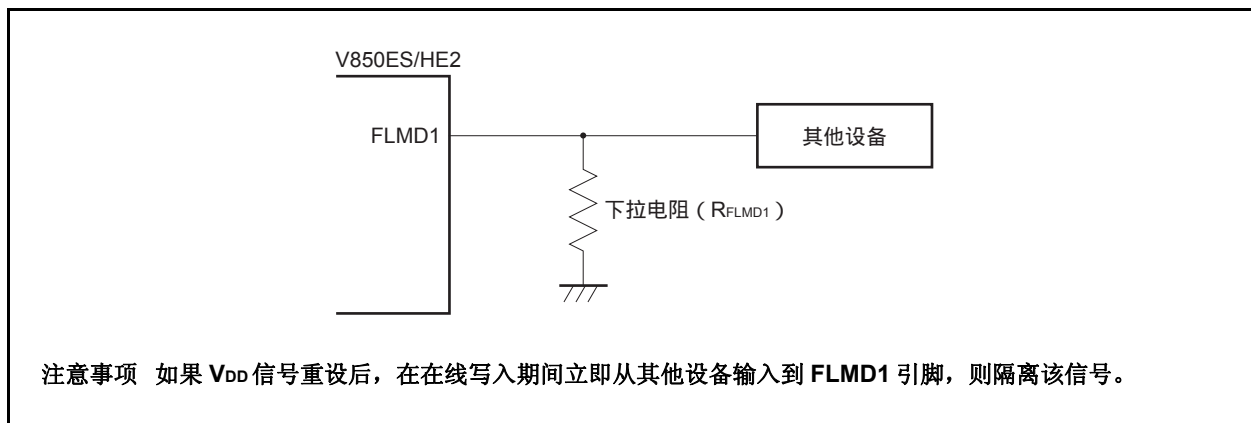


表 22-8. FLMD0 和 FLMD1 引脚的关系和复位释放时的工作模式

FLMD0	FLMD1	工作模式
0	忽略	普通工作模式
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

串行接口使用的引脚如下所示。

表 22-9. 串行接口使用的引脚

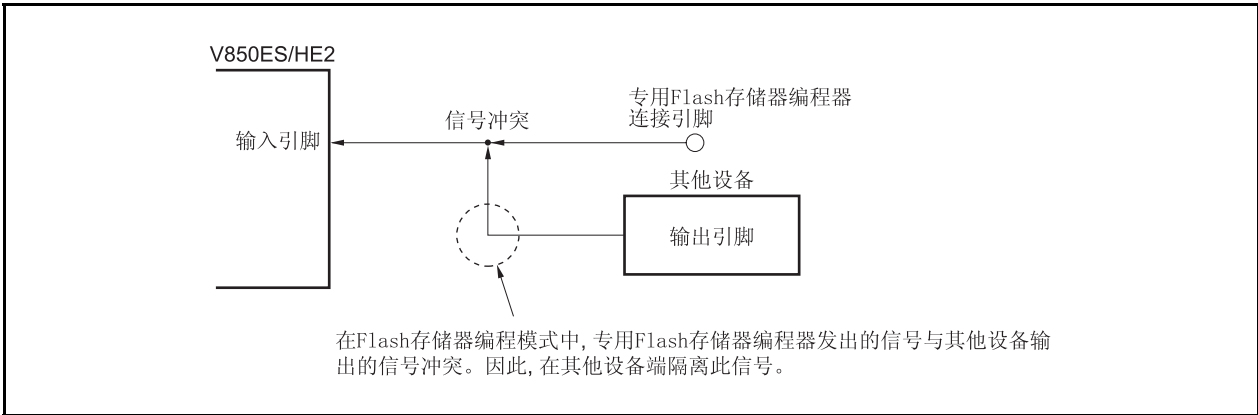
串行接口	使用引脚
UARTA0	TXDA0, RXDA0
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, PCM0

将专用 Flash 存储器编程器和连接在板上其它设备串行接口的引脚相连时，必须特别注意以避免信号之间冲突和其他设备的故障。

(a) 信号冲突

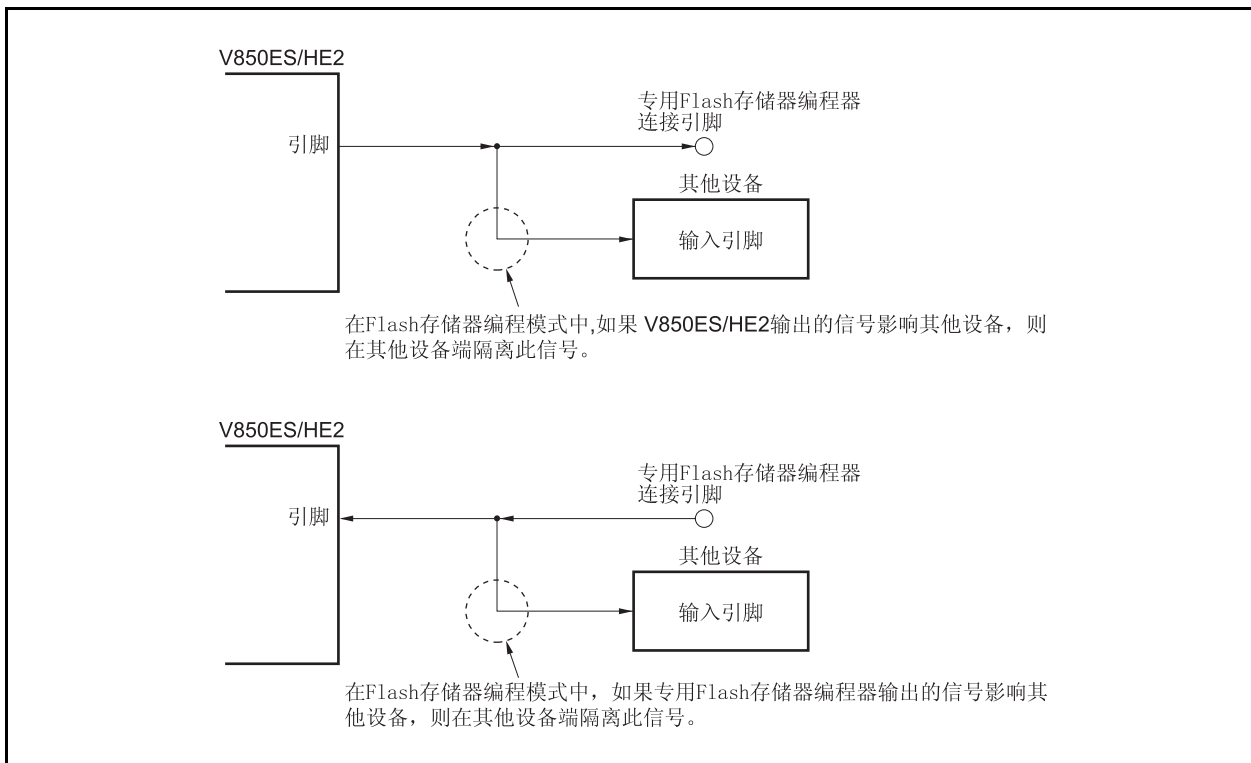
如果专用 Flash 存储器编程器（输出端）与串行接口（已连接到另一个设备的输出端）的一个输入引脚相连，这时会产生信号冲突。为了避免这种情况，应隔离与另一个设备的连接，或者使另一个设备处于高阻抗状态。

图 22-12. 信号冲突（串行接口 输入引脚）



(b) 其他设备故障

如果专用 Flash 存储器编程器（输出或输入）与已连接到另一个设备的输入端的串行接口引脚（输入或输出）相连，则信号可能会输出到另一个设备，从而引起该设备的故障。为了避免这种情况，应隔离与该设备的连接。

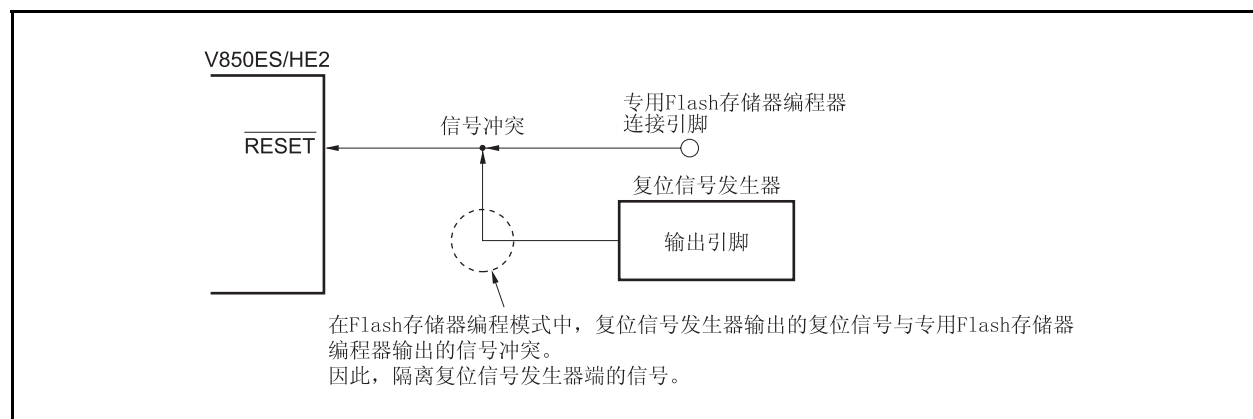
图 22-13. 其他设备故障

(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，则会产生信号冲突。为了避免这种情况，应隔离与复位信号发生器的连接。

在 Flash 存储器存储器编程模式下，如果从用户系统输入复位信号，则不能对 Flash 存储器存储器进行正确编程。因此除了专用 Flash 存储器编程器的复位信号外，不要输入其它信号。

图 22-14. 信号冲突 (RESET 引脚)



(5) 端口引脚 (包括 NMI)

当系统切换到 Flash 存储器编程模式时，复位后所有不用于 Flash 存储器编程的引脚状态相同。因此，如果外部设备（与端口相连）不能立即识别复位后的状态，那么引脚需要诸如通过一个电阻将端口引脚连接到 V_{DD} 或 V_{SS} 这样的合适操作。

(6) 其他信号引脚

在与普通操作模式中的状态相同的状态下连接 X1, X2, XT1, XT2。

在 Flash 存储器编程期间，输入低电平到 DRST 引脚或置为开路。不要输入高电平。

(7) 供电电源

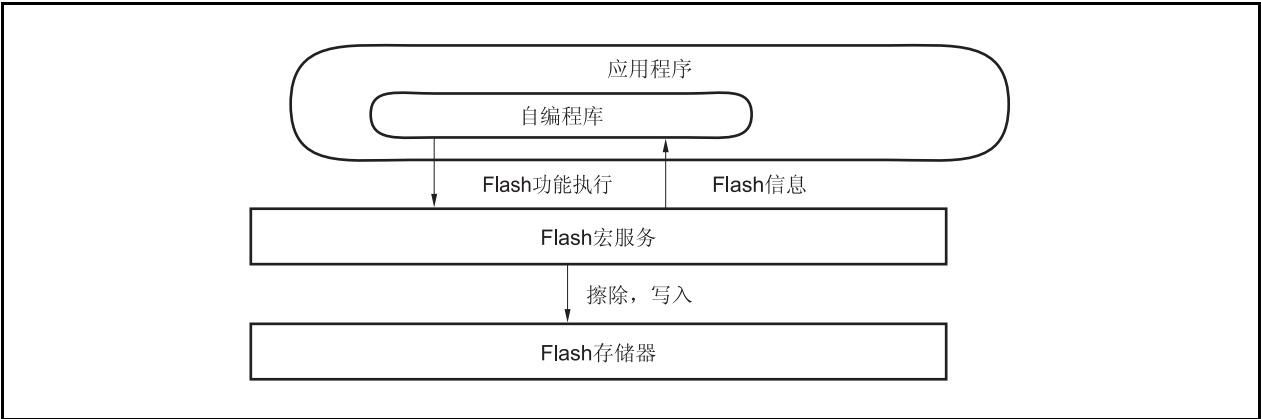
与普通工作模式下供电电源 (V_{DD} , V_{SS} , EV_{DD} , EV_{SS} , AV_{REF0} , AV_{SS} , $REGC$) 相同。

22.5 通过自编程重写

22.5.1 概述

V850ES/HE2 支持 Flash 存储器宏指令服务，允许用户程序自己重写内部 Flash 存储器。通过使用此接口和一个实现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。总之，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 22-15. 自编程的概念



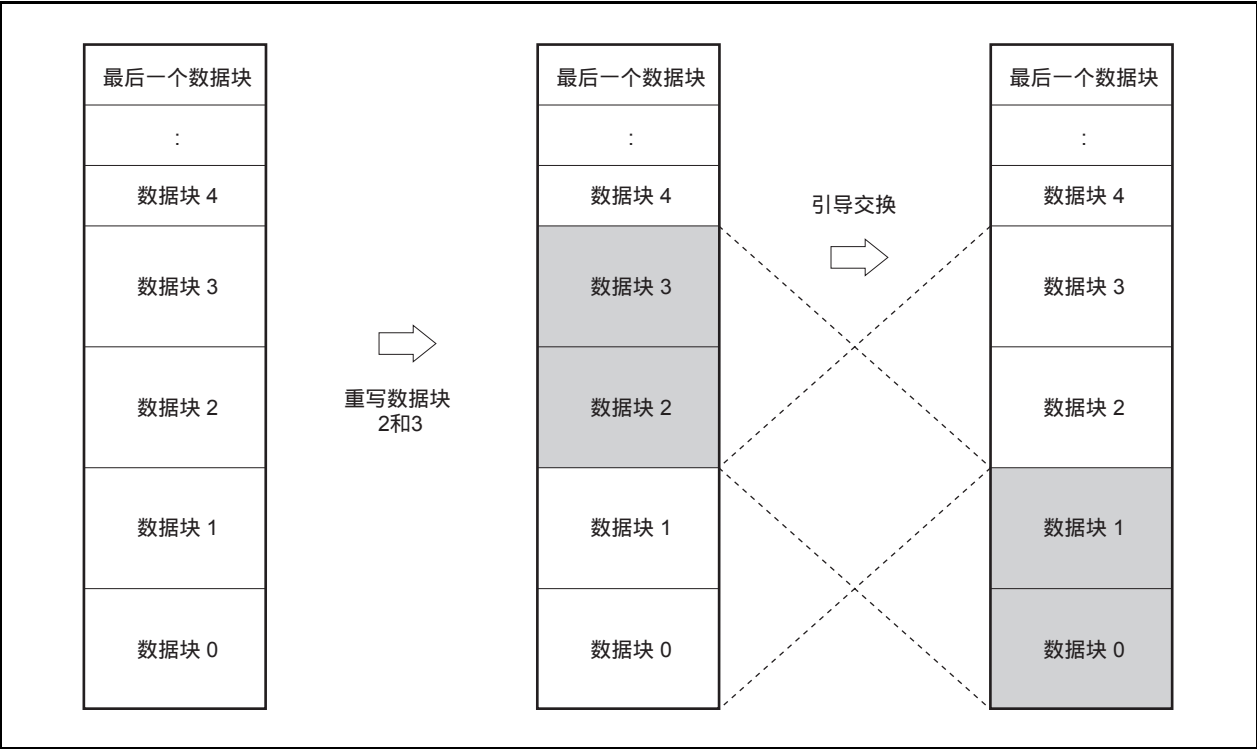
22.5.2 特点

(1) 可靠的自编程（引导交换功能）

μPD70F3701 支持引导交换功能，该功能将物理存储器的数据块 0 到 1 与数据块 2 到 3 进行交换。事先在数据块 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数据块 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。

注意事项 μPD70F3700 中不支持引导交换功能，因此只有数据块 0 和 1 是内存区域。

图 22-16. 重写整个内存区域（引导交换）



(2) 中断支持

自编程期间不能从 Flash 存储器获取指令。通常，即使发生中断也不能使用写入 Flash 存储器的用户句柄。因此，在 V850ES/HE2 中自编程期间使用中断，处理过程会跳转到内部 RAM 中的特殊地址^注。调用跳转指令使处理过程跳转到内部 RAM 中特殊地址^注上的用户中断服务程序。

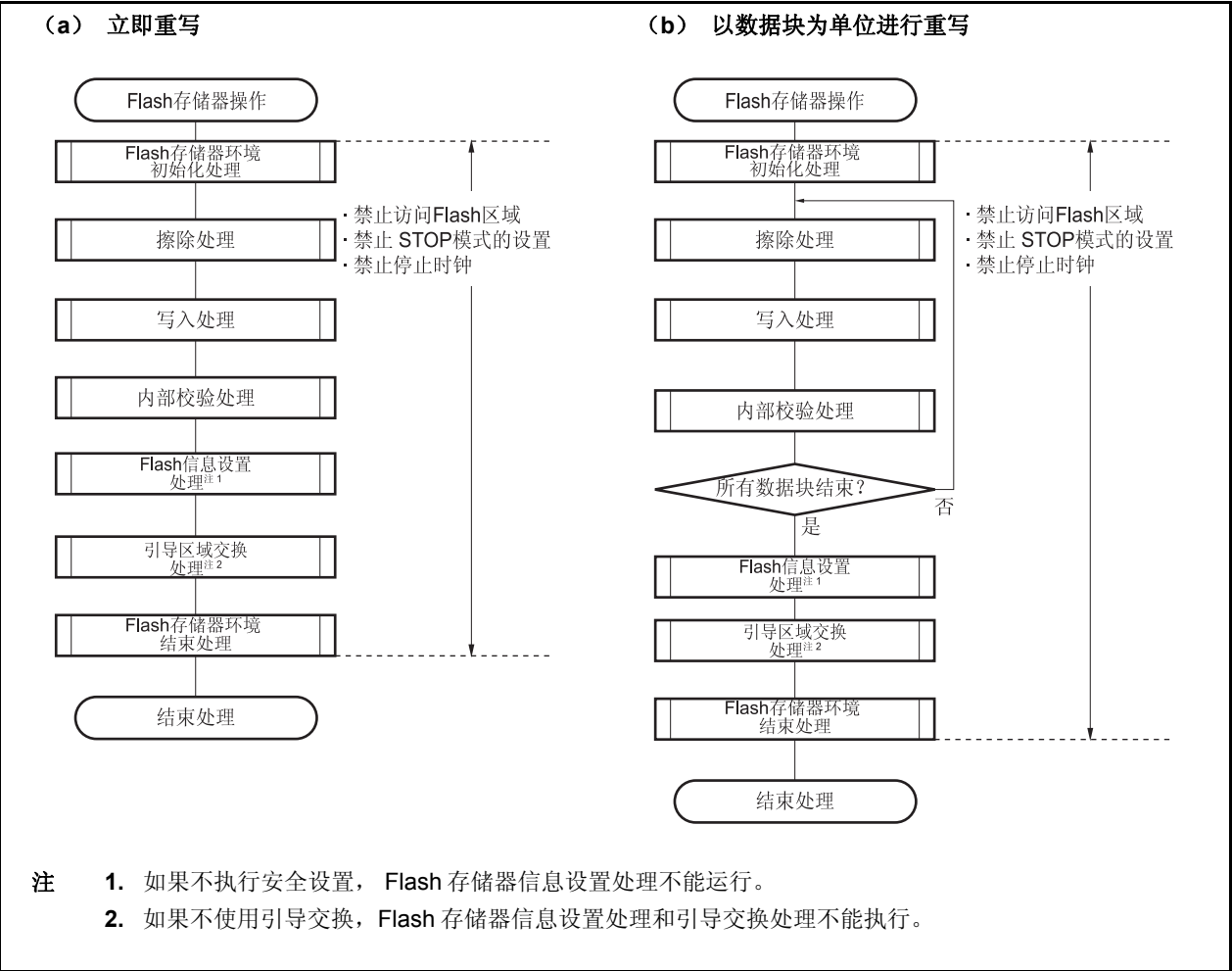
注 不可屏蔽中断： 内部 RAM 的开始地址。
 可屏蔽中断： 内部 RAM 的开始地址+ 4 个地址

22.5.3 标准自编程流程

由 Flash 存储器自编程重写 Flash 存储器的完整过程如下图所示。

<R>

图 22-17. 标准自编程流程



22.5.4 Flash 存储器功能

表 22-10. Flash 存储器功能列表

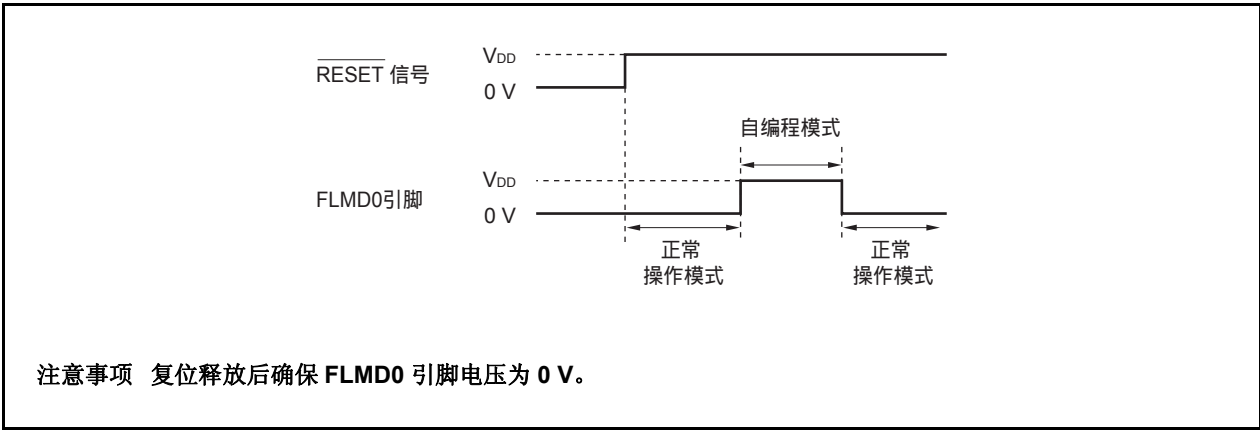
功能名称	概要	支持
FlashEnv	Flash 存储器控制宏初始化	√
FlashBlockErase	擦除指定数据块	√
FlashWordWrite	从指定地址中写入	√
FlashBlockIVerify	指定数据块的内部校验	√
FlashBlockBlankCheck	指定数据块空白检查	√
FlashFLMDCheck	FLMD 引脚的检查	√
FlashStatusCheck	指定操作前的状态检查	√
FlashGetInfo	Flash 存储器信息的读取	√
FlashSetInfo	Flash 存储器信息的设置	√
FlashBootSwap	引导区域的交换	√
FlashWordRead	从指定地址中读取数据	√
FlashSetUserHandler	用户中断处理程序注册功能	√

22.5.5 引脚处理

(1) FLMD0 引脚

FLMD0 引脚用于当复位释放时设置工作模式和在自写入时保护 Flash 存储器。因此当复位释放和普通工作模式时要保持 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 V_{DD} 。自编程完成后，FLMD0 引脚的电压必须返回 0 V。

图 22-18. 模式改变时序



22.5.6 使用的内部资源

下表所列为可用于自编程的内部资源。这些内部资源除了自编程也可以用于其他用途。

表 22-11. 使用的内部资源

<R>

资源名	描述
堆栈区域（用户堆栈 + 300 字节）	通过库使用的由用户应用的扩展堆栈（内部 RAM 和外部 RAM 均可使用）。
程序库代码（大约 2,500 字节）	库的程序实体（除了修改 Flash 存储器数据块外可以用于任何地方）。
应用程序	作为用户应用程序执行 称为 flash 功能
可屏蔽中断	可以用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这种中断，由于处理过程要跳转到内部 RAM 开始地址+4 的地址，需事先调用跳转指令，使处理过程跳转到内部开始地址+4 地址中的用户中断服务程序。
NMI 中断	可以用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这种中断，由于处理过程要跳转到内部 RAM 开始地址的地址，需事先调用跳转指令，使处理过程跳转到内部开始地址地址中的用户中断服务程序。

第 23 章 选项字节功能

选项字节被当作 8 位数据保存在内部 Flash 存储器（内部 ROM 区域）的地址 000007AH 中。

当将程序写入 V850ES/HE2 时，确保将相应于以下处于地址 000007AH 处的程序中的选项数据设置为默认数据。
在程序执行期间不能对该区域中的数据进行重写。

地址：000007AH

7	6	5	4	3	2	1	0
OPB7	OPB6	-	-	-	-	OPB1	OPB0

OPB7	OPB6	副时钟操作模式设置
0	0	晶体谐振器模式
1	1	RC 振荡器模式

OPB1	看门狗定时器 2 模式设置
0	操作时钟 (f_x/f_R) 可选 INTWDT2 模式/WDTRES 模式可选
1	固定为内部振荡时钟 (f_R) 固定为 WDTRES 模式

OPB0	允许/禁止停止内部振荡器
0	允许停止
1	禁止停止

<R> 使用 CA850 时的试样程序如下所示。

[试样程序]

```
#-----  
# OPTION_BYTES  
#-----  
.section "OPTION_BYTES"  
.byte 0b00000001 -- 0x7a  
.byte 0b00000000 -- 0x7b  
.byte 0b00000000 -- 0x7c  
.byte 0b00000000 -- 0x7d  
.byte 0b00000000 -- 0x7e  
.byte 0b00000000 -- 0x7f
```

注意事项 确保为该节写入 6 个字节。如果少于 6 个字节，那么在链接器工作期间将会发生错误。
错误消息： **F4112**：非法“**OPTION_BYTES**”节尺寸。

备注 将 0x00 设置到地址 007BH 至 007FH 中。

V850ES/HE2 片上调试功能可以通过以下两种方法来实行。

- 使用 DCU（调试控制单元）
通过将 $\overline{\text{DRST}}$ ，DCK，DMS，DDI 及 DDO 引脚用作调试接口引脚，片上调试功能可以通过 V850ES/HE2 中的片上 DCU 来实行。
- 不使用 DCU
片上调试功能不通过 DCU 单元来实行，取而代之的是通过 MINICUBE2 或类似使用用户资源的方法来实行。

下表显示了两种片上调试功能的特点。

表 24-1. 片上调试功能的特点

		使用 DCU 进行调试	不使用 DCU 进行调试
调试接口引脚		$\overline{\text{DRST}}$ ，DCK，DMS，DDI，DDO	<ul style="list-style-type: none">• 使用 UARTA0 时 RXD0，TXD0• 使用 CSIB0 时 SIB0，SOB0，$\overline{\text{SCKB0}}$，HS (PCM0)
取得用户资源		不需要	需要
硬件中断功能		2 点	2 点
软件中断功能	内部 ROM 区域	4 点	4 点
	内部 RAM 区域	2000 点	2000 点
实时 RAM 监视功能 ^{註 1}		可用	可用
动态存储器修改 (DMM) 功能 ^{註 2}		可用	可用
屏东功能		复位，NMI，INTWDT2	$\overline{\text{RESET}}$ 引脚
ROM 安全功能		10 字节 ID 代码认证	10 字节 ID 代码认证
使用的硬件		NINICUBE [®] 等	NINICUBE2 等
追踪功能		不支持。	不支持。
调试中断接口功能 (DBINT)		不支持。	不支持。

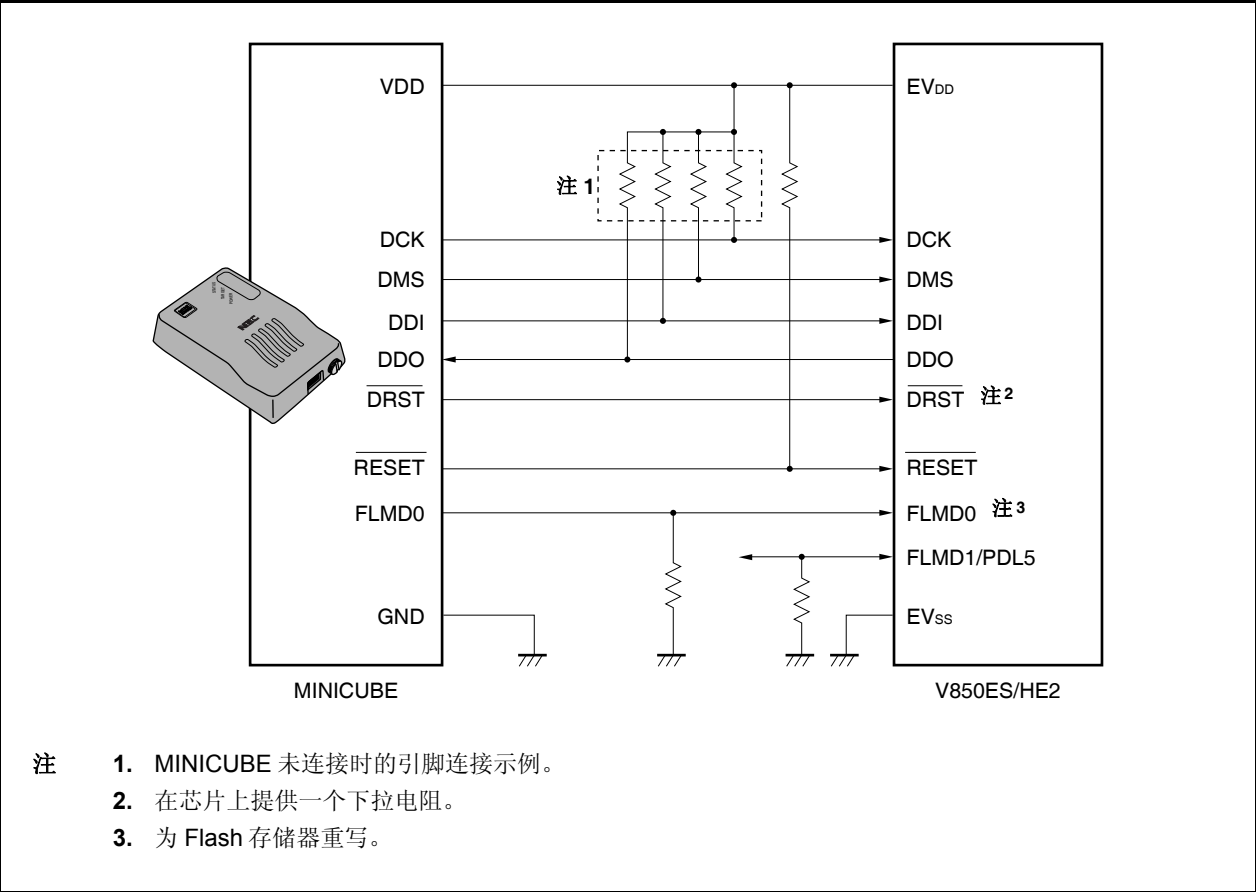
注 1. 这是用于在程序执行期间读取存储内容的功能。
 2. 这是用于在程序执行期间重写 RAM 内容的功能。

24.1 使用 DCU 进行调试

程序可以通过使用调试接口引脚（ $\overline{\text{DRST}}$ ，DCK，DMS，DDI 和 DDO）来调试，用以连接到片上调试仿真器（MINICUBE）上。

24.1.1 电路连接示例

图 24-1. 调试端口引脚用于通信接口时的电路连接示例



24.1.2 接口信号

接口信号说明如下。

(1) $\overline{\text{DRST}}$

对于片上调试单元这是一个复位输入信号。它是异步初始化调试控制单元的负逻辑信号。

调试器启动后，当片上调试模拟器探测到目标系统的 V_{DD} 时，它会生成 $\overline{\text{DRST}}$ 信号，并且启动设备的片上调试单元。

当 $\overline{\text{DRST}}$ 信号拉高时，在 CPU 中同样会生成一个复位信号。

当启动集成调试器来开始调试时，通常会产生 CPU 复位。

(2) DCK

这是一个时钟输入信号。它从片上调试模拟器中提供一个 20MHz 的时钟。在片上调试单元中，DMS 和 DDI 信号是在 DCK 信号的上升沿采样，数据 DDO 是在它的下降沿处输出。

(3) DMS

这是发送模式选择信号。调试单元的发送状态根据 DMS 信号的电平改变。

(4) DDI

这是一个数据输入信号。DCK 上升沿时在片上调试单元采样。

(5) DDO

这是一个数据输出信号。DCK 信号下降沿时从片上调试单元输出。

(6) EV_{DD}

这是一个用于检测目标系统 VDD 的信号。如果目标系统中的 VDD 没有被检测到，那么从片上调试模拟器中输出的信号 ($\overline{\text{DRST}}$, DCK, DMS, DDI, FLMD0 以及 $\overline{\text{RESET}}$) 会进入高阻抗状态。

(7) FLMD0

Flash 存储器自编程功能用于通过集成调试器下载数据到 Flash 存储器。在 Flash 存储器自编程期间，FLMD0 引脚必须保持为高。另外，将下拉电阻连接到 FLMD0 引脚。

FLMD0 引脚也可以用如下两种方式控制。

<1> 从片上调试模拟器中控制

将片上调试模拟器中的 FLMD0 信号连接到 FLMD0 引脚。

在正常模式下，片上调试模拟器（高阻抗）不能进行任何驱动。

在暂停期间，执行集成调试器的下载功能时片上调试模拟器会使 FLMD0 引脚升为高电平的。

<2> 通过端口控制

连接设备的任意端口到 FLMD0 引脚。

通过用户程序实现 Flash 存储器自编程功能任意端口都可以使用。

在集成调试器的控制台上，执行下载功能前设置端口引脚为高电平，或者执行完下载功能后置端口引脚为低。

详细内容，请参阅 ID850QB Ver. 3.10 集成调试器用户操作手册（U17435E）。

(8) $\overline{\text{RESET}}$

这是一个系统复位输入引脚。如果 $\overline{\text{DRST}}$ 引脚由用户程序设置的 OCDM 寄存器的 OCDM0 位的值无效，那么不能执行片上调试功能。因此，重设功能受片上调试模拟器的影响，使用 $\overline{\text{RESET}}$ 引脚以使 $\overline{\text{DRST}}$ 引脚有效（初始化）。

24.1.3 屏蔽功能

复位，NMI 以及 INTWDT2 信号可以被屏蔽。
以下列出了调试器（ID850QB）的屏蔽功能以及相应的 V850ES/HE2 的功能。

表 24-2. 屏蔽功能

ID850QB 的屏蔽功能	相应的 V850ES/HE2 的功能
NMI0	NMI 引脚输入
NMI2	生成不可屏蔽中断请求信号（INTWDT2）
STOP	—
HOLD	—
RESET	RESET 引脚输入引起的复位信号生成，低电压监视器，时钟监视器，上电清零电路或看门狗定时（WDT2）溢出
WAIT	—

24.1.4 寄存器

（1）片上调试模式寄存器（OCDM）

OCDM 寄存器用于选择正常操作模式或片上调试模式。该寄存器是特殊寄存器，并且只有在对规定的顺序组合下才能写入（参见 3.4.7 特殊寄存器）。

此寄存器也用来指定提供片上调试功能的引脚是作为片上调试引脚还是作为一般的端口/外围功能引脚。也可用于断 P05/INTP2/ $\overline{\text{DRST}}$ 引脚的内部下拉电阻。

OCDM 寄存器仅当 $\overline{\text{DRST}}$ 引脚输入为低电平时可以写入。

该寄存器可用 8 位或 1 位指令读写。

复位后: 01H^注 R/W 地址: FFFFFFF9FCH

	7	6	5	4	3	2	1	0
OCDM	0	0	0	0	0	0	0	OCDM0

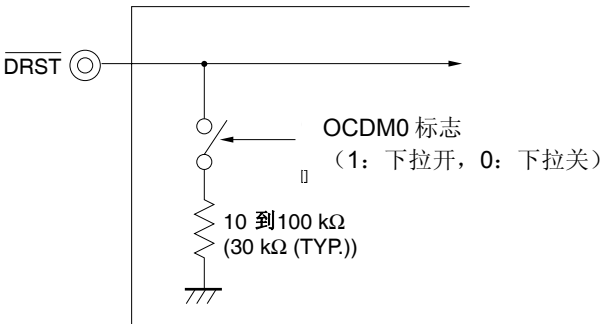
OCDM0	操作模式
0	选择正常操作模式（在此模式下，片上调试功能引脚可用作端口/外部功能引脚），同时断开 P05/INTP2/DRST 引脚的片上下拉电阻。
1	当 DRST 引脚位于低： 正常操作模式（在此模式下，片上调试功能引脚可用作端口/外部功能引脚） 当 DRST 引脚位于高： 片上调试模式（在此模式下，片上调试功能引脚可用作片上调试模式引脚）

注 RESET 输入设置此寄存器为 01H。通过 WDT2RES 信号复位后，时钟监视器（CLM），或低电压检测器（LVI），上电清零电路（POC），OCDM 寄存器的值仍被保留。

注意事项 1. 外部复位后，当 DDI，DDO，DCK，和 DMS 引脚不用于片上调试引脚而作为端口引脚时，可以进行下面的任何一项操作。

- 输入低电平到 P05/INTP2/DRST 引脚。
- 设置 OCDM0 位。在这种情况下，进行如下操作。
 - <1> 将 OCDM0 位清零。
 - <2> 将 P05/INTP2/DRST 引脚固定为低电平直到 <1> 完成。

2. DRST 引脚有片上下拉电阻。这个电阻当 OCDM0 标志清 0 时断开。



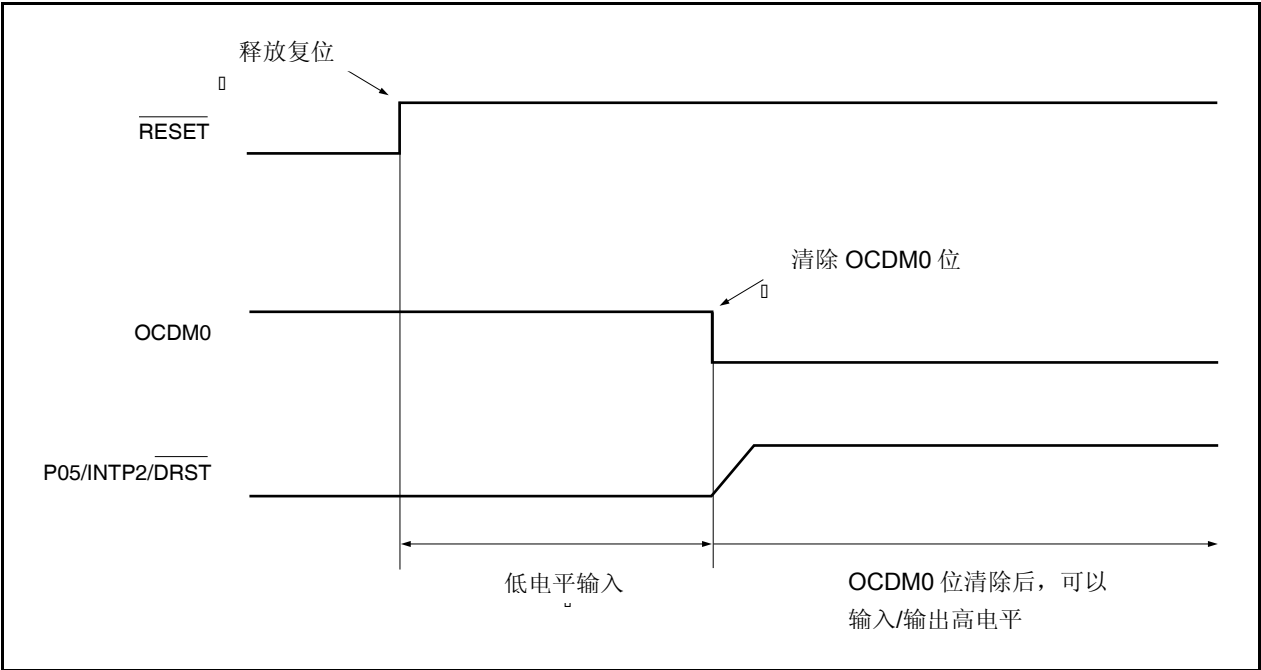
24.1.5 操作

在下表中列出的条件下片上调试功能无效。
当不使用此功能时，保持 $\overline{\text{DRST}}$ 引脚为低直到 OCDM.OCDM0 标志被清零。

<div>OCDM0 标志</div> <div>$\overline{\text{DRST}}$ 引脚</div>	0	1
L	无效	无效
H	无效	有效

备注 L: 低电平输入
 H: 高电平输入

图 24-2. 不使用片上调试功能时的时序



24.1.6 注意事项

- (1) 如果复位信号在 RUN（程序执行）期间被输入（从目标系统或内部复位源的复位信号中），那么中断功能可能会发生故障。
- (2) 如果复位信号从引脚中输入，那么即使复位信号通过屏蔽功能被屏蔽，I/O 缓冲器（端口引脚）也可能被复位。
- (3) 由于在内部 Flash 存储器中设置的软件中断点会通过目标复位或通过由看门狗定时器 2 生成的内部复位被暂时设为无效的。因此，当硬件中断或强行中断发生时，中断点将会变成有效的，但在这个过程中不会发生软件中断。
- (4) 中断过程中引脚复位被屏蔽，且 CPU 及外围 I/O 不会被复位。执行用户程序时，一旦 Flash 存储器通过 DMM 进行重写或通过 RAM 监视功能进行读取，将会立即生成引脚复位或内部复位，此时 CPU 和外围 I/O 可能不会被正确复位。
- (5) 满足以下条件（a）和（b）且仿真器（IECUBE[®]， MINICUBE）上的操作由于中断等原因而停止时，看门狗定时器 2 不会停止且会发生复位或不可屏蔽中断。当复位发生时，调试器将会被挂起。

(a) 为看门狗定时器 2 将主时钟或子时钟用作源时钟。

(b) 内部振荡时钟被停止（RCM.RSTOP 位 = 1）。

为了避免这种情况，应执行以下任一操作。

- 当使用仿真器时，将内部振荡时钟用作源时钟。
- 当使用仿真器时，不要停止内部振荡器。

- (6) 满足以下条件（a）和（b）且仿真器（IECUBE， MINICUBE）上的操作由于中断等原因而停止时，即使外围中断功能被设置为“Break”，TMM 也不会停止。

(a) INTWLT，内部振荡时钟（f_R/8）或子时钟被选为 TMM 源时钟。

(b) 主时钟被停止。

为了避免这种情况，应执行以下任一操作。

- 当使用仿真器时，主时钟（f_{xx}， f_{xx}/2， f_{xx}/4， f_{xx}/64， f_{xx}/512）被用作源时钟。
- 当使用仿真器时，禁止主时钟振荡。

- (7) 在片上调试模式中，DDO 引脚被强行设为高电平输出。

24.2 不使用 DCU 进行调试

以下描述了在不使用 DCU 的情况下，通过将带有 UARTA0 (RXDA0 和 TXDA0) 的引脚的 MINICUBE2 或带有 CSIB0 (SIB0, $\overline{\text{SOB0}}$, SCKB0 和 HS (PMC0)) 的引脚的 MINICUBE2 用作调试接口来实现片上调试功能的方法。

24.2.1 电路连接示例

图 24-3. UARTA0/CSIB0 用于通信接口时的电路连接示例

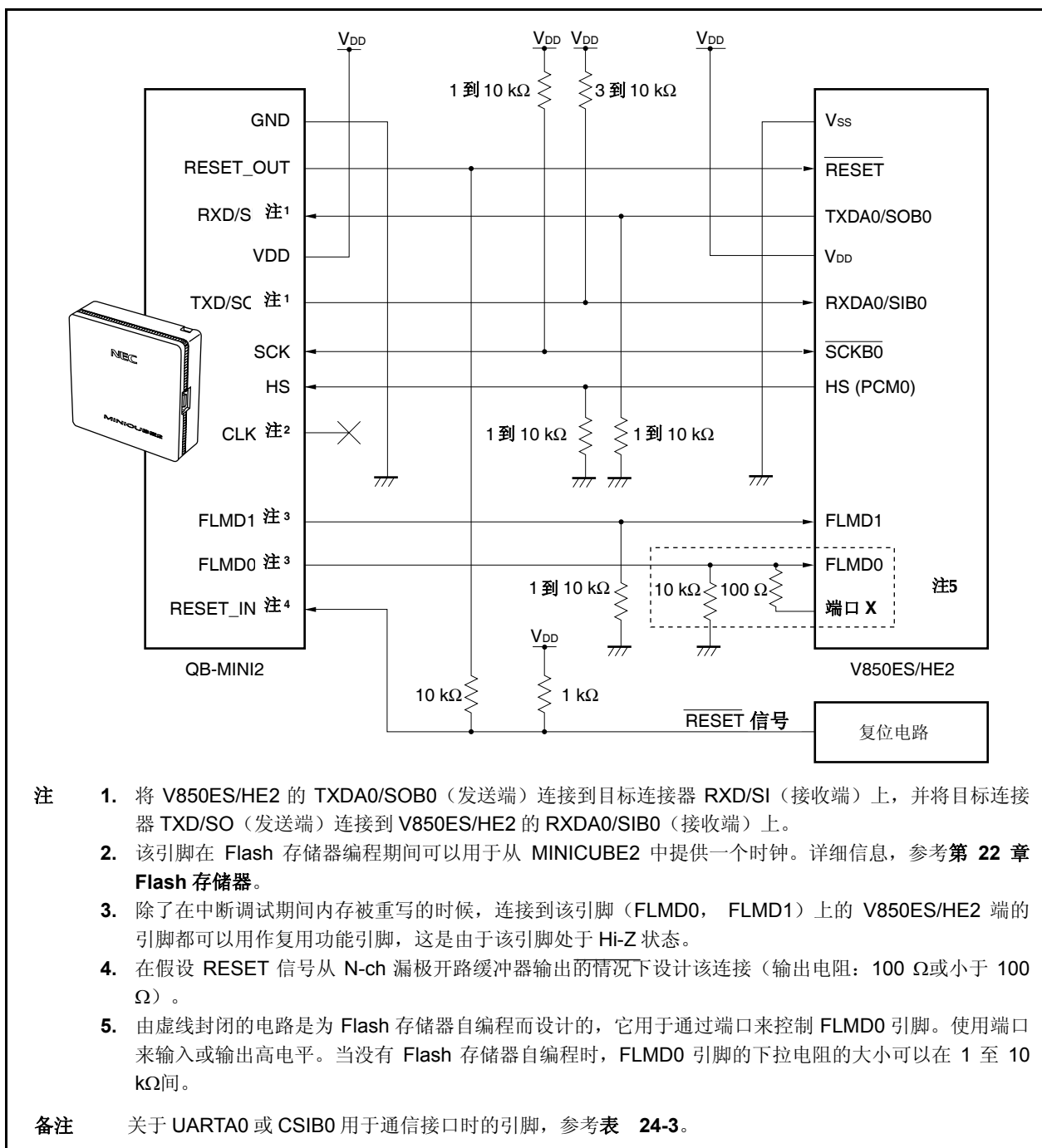


表 24-3. V850ES/HE2 与 MINICUBE2 间的连线

MINICUBE2 (QB-MINI2) 的引脚配置			使用 CSIB0-HS 时		使用 UARTA0 时	
信号名称	I/O	引脚功能	引脚名	引脚号	引脚名	引脚号
SI/RxD	输入	用于从 V850ES/HE2 中接收命令和数据的引脚	P41/SOB0	20	P30/TXD0	22
SO/TxD	输出	用于将命令和数据发送到 V850ES/HE2 的引脚	P40/SIB0	19	P31/RXD0	23
SCK	输出	用于 3 线串行通信的时钟输出引脚	P42/SCKB0	21	不需要	–
CLK ^注	输出	V850ES/HE2 的时钟输出引脚	Not needed ^注	–	不需要 ^注	–
			Not needed ^注	–	不需要 ^注	–
RESET_OUT	输出	V850ES/HE2 的复位输出引脚	RESET	9	RESET	9
FLMD0	输出	用于将 V850ES/HE2 设为调试模式或编程模式输出引脚	FLMD0	3	FLMD0	3
FLMD1	输出	用于设置编程模式的输出引脚	PDL5/FLMD1	52	PDL5/FLMD1	52
HS	输入	CSI0 + HS 通信的商议讯号	PCM0	45	不需要	–
GND	–	接地	V _{SS}	6	V _{SS}	6
			AV _{SS}	2	AV _{SS}	2
			EV _{SS}	32	EV _{SS}	32
RESET_IN	输入	目标系统中的复位输入引脚				

注 为 MINICUBE2 将其用作 Flash 存储器编程器的时钟输出。详细信息，参考第 22 章 Flash 存储器。

24.2.2 可屏蔽功能

只有复位信号可以屏蔽。

以下列出了调试器 (ID850QB) 的可屏蔽功能以及相应的 V850ES/HE2 的功能。

表 24-4. 可屏蔽功能

ID850QB 的可屏蔽功能	相应的 V850ES/HE2 的功能
NMI0	–
NMI1	–
NMI2	–
STOP	–
HOLD	–
RESET	通过 RESET 引脚输入生成复位信号
WAIT	–

24.2.3 用户资源的获取

用户必须为在 MINICUBE2 与目标设备间进行通信而准备以下内容并实行每个调试功能。这些项目需要在用户程序中进行设置或使用编译程序选项。

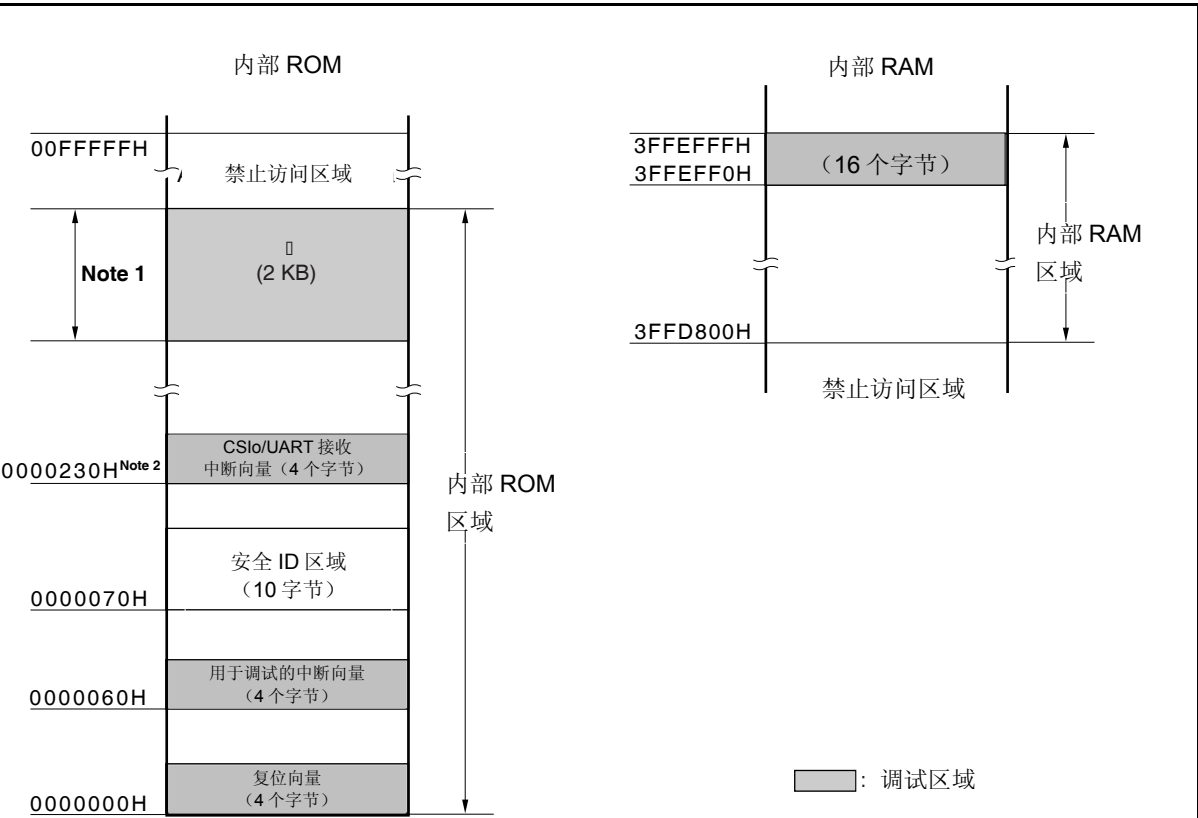
(1) 取得内存空间

图 24-4 中的阴影部分是为放置调试监视程序而保留的区域，因此用户程序和数据不能被分配到这些空间中。必须获取这些空间以免被用户程序所使用。

(2) 安全 ID 设置

ID 代码必须被嵌入到图 24-4 中 0000070H 和 0000079H 间的区域中，以用于保护存储器不被未经授权者读取。详细信息，参考 **24.3 ROM 安全功能**。

图 24-4. 调试监视程序被分配到的内存空间



- 注 1. 地址值根据产品的不同而变化。
- | | 内部 ROM 尺寸 | 地址 |
|------------|-----------|---------------------|
| μPD70F3700 | 64 KB | 000F800H 至 000FFFFH |
| μPD70F3701 | 128 KB | 001F800H 至 001FFFFH |
2. 这是使用 CSIB0 时的地址。使用 UARTA0 时，它从 0000270H 处开始。

(3) 复位微量

复位向量包括用于调试监视程序的跳转指令。

[获取该区域的方法]

不需要刻意获取该区域。然而，在下载程序时，调试器会根据以下情况对复位向量进行重写。如果重写模式与以下情况不匹配，那么调试将会生成一个错误（使用 ID850QB 时为 F0C34）。

(a) 从地址 0 连续插入 nop 指令时

重写前		重写后
0x0 nop	→	在 0x0 处跳转到调试监视程序
0x2 nop		0x4 xxxx
0x4 xxxx		

(b) 从地址 0 连续插入两个 0xFFFF（已经擦除的设备）时

重写前		重写后
0x0 0xFFFF	→	在 0x0 处跳转到调试监视程序
0x2 0xFFFF		0x4 xxxx
0x4 xxxx		

(c) 在地址 0 处插入 jr 指令时（使用 CA850 时）

重写前		重写后
0x0 jr disp22	→	在 0x0 处跳转到调试监视程序
		0x4 jr disp22 - 4

(d) 从地址 0 连续插入 mov32 和 jmp 时（使用 IAR 编译程序 ICCV850 时）

重写前		重写后
0x0 mov imm32, reg1	→	在 0x0 处跳转到调试监视程序
0x6 jmp [reg1]		0x4 mov imm32, reg1
		0xa jmp [reg1]

(e) 在地址 0 处插入调试监视程序的跳转指令

重写前		重写后
在 0x0 处跳转到调试监视程序	→	没有变化

(4) 调试监视程序区域的获取

图 24-4 中的阴影部分是调试监视程序分配到的区域。监视程序为调试通信接口执行初始化处理并为 CPU 执行中断处理。内部 ROM 区域必须用 0xFF 填满。该区域不必通过用户程序来重写。

[获取该区域的方法]

如果用户程序没有使用该区域，则不需要获取该区域。

然而，为了避免在调试器运行期间可能产生的问题，推荐使用编译器事先获取该区域。

以下显示了使用 NEC 的编译器 CA850 获取该区域的示例。以下也显示了添加汇编源文件和链接导引代码的情况。

- 汇编源（将以下代码添加为汇编源文件。）

```
-- 为监视 ROM 节获取 2KB 的空间
.section "MonitorROM", const
.space    0x800, 0xff

--为调试获取中断向量
.section "DBG0"
.space 4, 0xff

--为串行通信获取中断向量
--依照所使用的串行通信模式更改节名
.section "INTCB0R"
.space 4, 0xff

--为监视 RAM 节获取 16 字节的空间
.section "MonitorRAM", bss
.lcomm monitorramsym, 16, 4 -- 定义符号 monitorramsym
```

- 链接导引（将以下代码添加到链接导引文件中。）

以下显示了当内部 ROM 具有 128KB 的空间（结束地址为 003FFFFH）以及内部 RAM 具有 6KB 的空间（结束地址为 3FFEFFFH）时的示例。

```
MROMSEG      : !LOAD ?R V0x01f800{
               MonitorROM    = $PROGBITS    ?A MonitorROM;
};

MRAMSEG       : !LOAD ?RW V0x03ffeff0{
               MonitorRAM     = $NOBITS      ?AW MonitorRAM;
};
```

(5) 通信串行接口的获取

UARTA0 或 CSIB0 用于 MINICUBE2 和目标系统间的通信。与串行接口模式相关的设置通过调试监视程序来执行，但如果设置通过用户程序被更改，那么可能会发生通信错误。
为了防止发生这个情况，通信串行接口必须从用户程序中获取。

[获取通信串行接口的方法]

- 片上调试模式寄存器（OCDM）
对于使用 UARTA0 或 CSIB0 的片上调试功能，应将 OCDM 寄存器功能设为正常模式。请确保按如下进行设置。
 - 输入低电平到 P05/INTP2/ $\overline{\text{DRST}}$ 引脚中。
 - 按如下所示对 OCDM0 位进行设置。
 - <1> 将 OCDM0 位清 0。
 - <2> 将 P05/INTP2/ $\overline{\text{DRST}}$ 引脚输入固定为低电平直到完成步骤<1>为止。
- 串行接口寄存器
不要在用户程序中对与 CSIB0 或 UARTA0 相关的寄存器进行设置。
- 中断屏蔽寄存器
使用 CSIB0 时，不要屏蔽发送终止中断（INTCB0R）。使用 UARTA0 时，不要屏蔽接收终止中断（INTUA0R）。

(a) 使用 CSIB0 时								
	7	6	5	4	3	2	1	0
CB0RIC	×	0	×	×	×	×	×	×
(b) 使用 UARTA0 时								
	7	6	5	4	3	2	1	0
UA0RIC	×	0	×	×	×	×	×	×
备注	×: 不考虑							

• 使用 UARTA0 时的端口寄存器

使用 UARTA0 时，端口寄存器通过调试监视程序被设置用于使 TXDA0 和 RXDA0 变成有效的。在调试期间不要使用用户程序更改以下寄存器的设置。（相同的值可以被覆盖。）

	7	6	5	4	3	2	1	0
PMC3L	×	×	×	×	×	×	1	1

备注 ×：不考虑

• 使用 CSIB0 时的端口寄存器

使用 CSIB0 时，端口寄存器通过调试监视程序被设置用于使 SIB0，SOB0， $\overline{\text{SCKB0}}$ 和 HS（PMC0）引脚变成有效的。在调试期间不要使用用户程序更改以下寄存器的设置。（相同的值可以被覆盖。）

(a) SIB0，SOB0 和 $\overline{\text{SCKB0}}$ 的设置								
	7	6	5	4	3	2	1	0
PMC4	×	×	×	×	×	1	1	1

(b) HS（PMC0 引脚）的设置								
	7	6	5	4	3	2	1	0
PMCM	×	×	×	×	×	×	×	0

	7	6	5	4	3	2	1	0
PCM	×	×	×	×	×	×	×	注

注 禁止对该位进行写操作。
相应于 HS 引脚的端口值依照调试器状态通过监视程序来改变。要以 8 位为单位来执行端口寄存器的设置，用户程序通常可以使用读取-修改-写入操作。然而，如果在写入前发生调试中断，则将会执行一个意外操作。

备注 ×：不考虑

24.2.4 注意事项

(1) 用于调试的设备的处理

不要将用于调试的设备装载到大批量生产的产品上，因为 Flash 存储器会在调试期间被重写且 Flash 存储器重写的次数不能被保证。此外，不要将调试监视程序嵌入到大批量生产的产品中。

(2) 不能执行中止时

如果以下任一条件被满足，则将不能执行强行中止。

- 禁止中断 (DI)。
- 为串行接口发出的用于 MINICUBE2 及目标设备间通信的中断被屏蔽。
- 禁止在通过可屏蔽中断释放待命时进入待机模式。
- MINICUBE2 与目标设备间的通信模式是 UARTA0，且主时钟已经被停止。

(3) 没有执行伪实时 RAM 监视 (RRM) 功能及 DMM 功能 时

如果以下任一条件被满足，则将不能执行伪 RRM 功能以及 DMM 功能。

- 禁止中断 (DI)。
- 为串行接口发出的用于 MINICUBE2 及目标设备间通信的中断被屏蔽。
- 禁止在通过可屏蔽中断释放待命时进入待机模式。
- MINICUBE2 与目标设备间的通信模式是 UARTA0，且主时钟已经被停止。
- MINICUBE2 与目标设备间的通信模式是 UARTA0，且不同于在调试器中所指定的时钟被用于通信。

(4) 允许使用伪 RRM 及 DMM 功能来释放等待

如果以下任一条件被满足，那么待机模式将通过伪 RRM 功能和 DMM 功能来释放。

- MINICUBE2 与目标设备间的通信模式是 CSIB0。
- MINICUBE2 与目标设备间的通信模式是 UARTA0，且已经提供主时钟。

(5) 使用 DMM 功能对需要特定写入顺序的外围 I/O 寄存器进行写操作

需要特定写入顺序的外围 I/O 寄存器不能使用 DMM 功能来写入。

(6) 调试程序启动变慢的设备

第一次启动调试程序时会执行芯片擦除以及监视程序的写操作，但执行该操作大概需要十二秒。

(7) 调试监视程序的写操作

当 CPU 操作时钟的设置通过调试器改变时，调试器将会重写监视程序。所需要的时间与上述 (6) 中的相同。

对于集成调试器 ID850QB 来说，这会在配置对话框中的 Clock 列的设置被改变时应用。

(8) Flash 存储器自编程

如果调试监视程序被分配到的空间通过 Flash 存储器自编程进行重写，那么调试器将不能正常运行。

24.3 ROM 安全功能

24.3.1 安全 ID

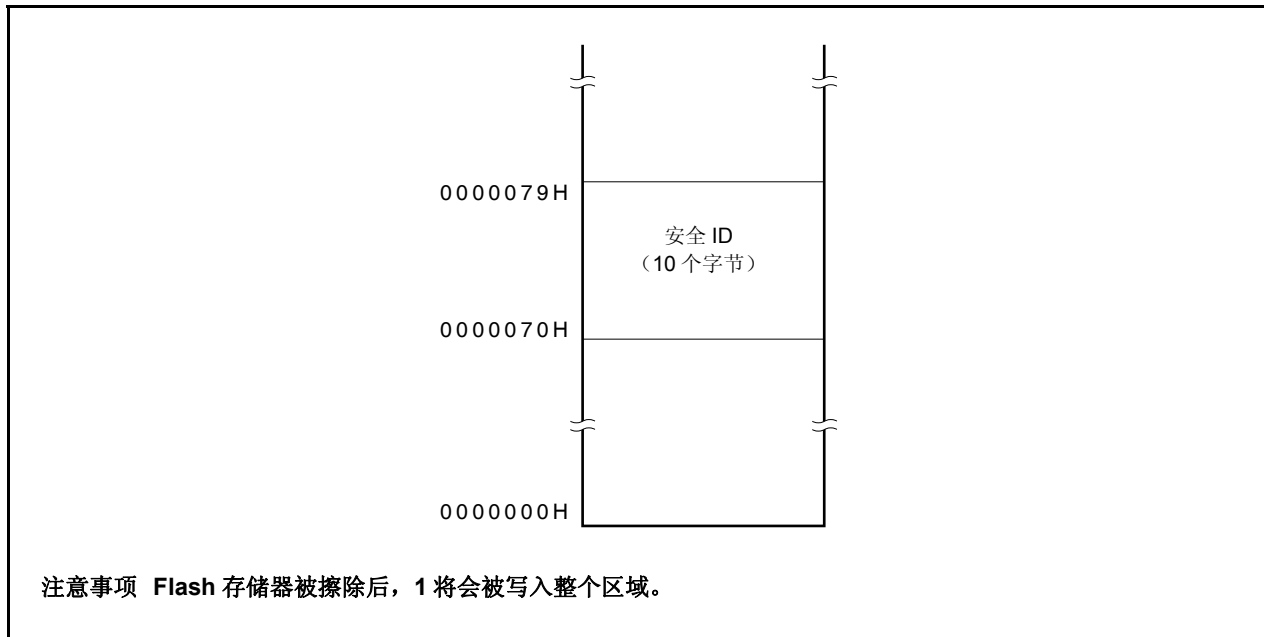
V850ES/HE2 的 Flash 存储器版使用 10 个字节的 ID 码来进行加密，以防在片上调试期间 Flash 存储器的内容被没有授权的人通过片上调试仿真器读取。

从 0000070H 到 0000079H 设置片上 Flash 存储器区域的 10 位 ID 密码域进行设置，以允许调试器进行 ID 加密。

如果 IDs 匹配，那么就释放安全保密功能，允许读取 Flash 存储器和使用片上调试模拟器。

- 在 0000070H 至 0000079H 之间设置 10 个字节的 ID 号码。
- 0000079H 的位 7 是片上调试模拟器的允许标志。
(0: 禁止, 1: 允许)
- 当片上调试模拟器启动时，调试器要求输入 ID 号码。当输入调试器的 ID 号码与设置在 0000070H 到 0000079H 的 ID 号码一致时，调试器开始工作。
- 如果片上调试器允许标志为 0，即使 ID 号码匹配，也不能进行调试。

图 24-5. 安全 ID 区域



24.3.2 设置

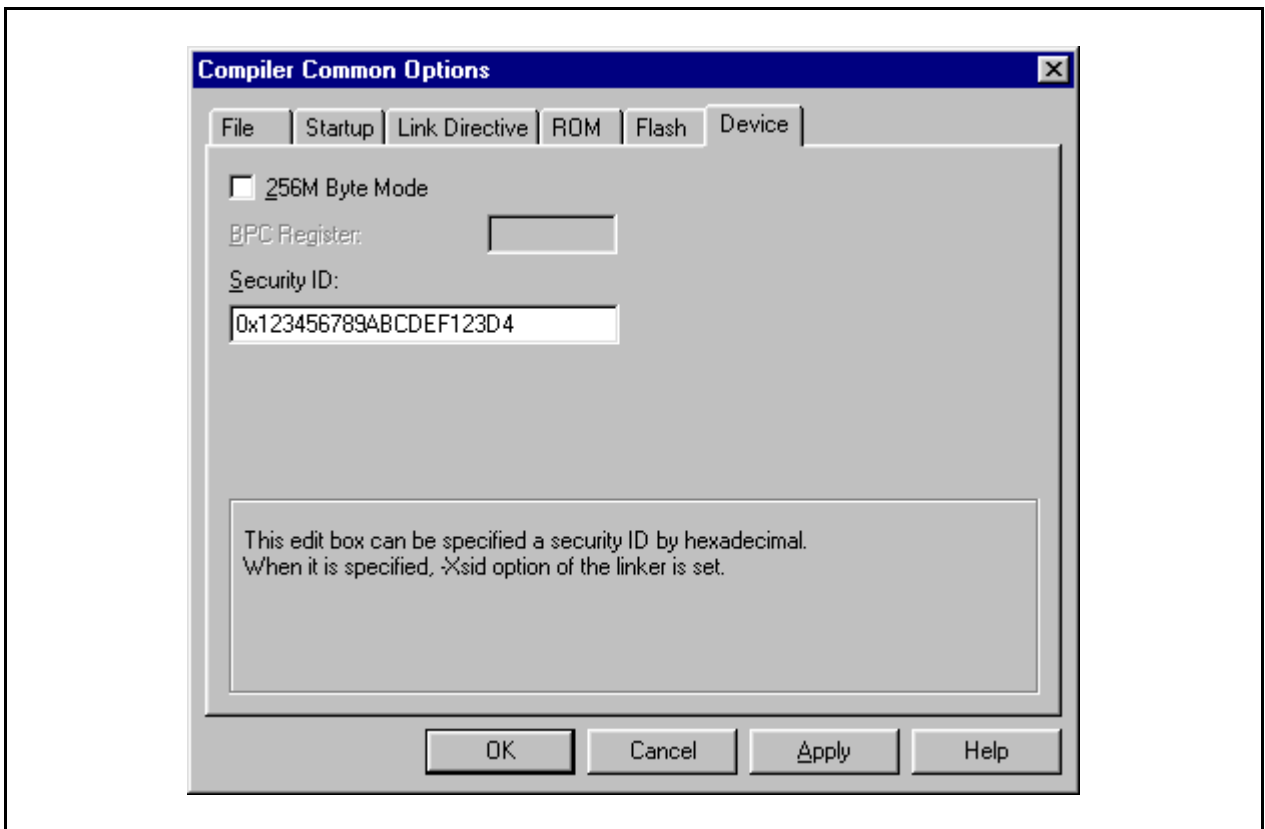
如表 24-5 所示，以下显示了 ID 代码的设置方法。

当 ID 代码按表 24-5 中所显示的进行设置时，在 ID850QB 的配置对话框中输入的 ID 代码将会是“123456789ABCDEF123D4”（ID 代码不区分大小写）。

表 24-5. ID 代码

地址	值
0x70	0x12
0x71	0x34
0x72	0x56
0x73	0x78
0x74	0x9A
0x75	0xBC
0x76	0xDE
0x77	0XF1
0x78	0x23
0x79	0xD4

ID 代码可以为支持 CA850 Ver. 3.10 或更高版本的的设备文件来指定，也可以为使用 PM+编译程序公共选项设置的安全 ID 来指定。



[程序示例（使用 CA850 Ver. 3.10 或更高版本时）]

```
#-----  
#      SECURITYID  
#-----  
      .section      "SECURITY_ID"      --中断处理地址 0x70  
      .word          0x78563412        --0-3 字节的代码  
      .word          0xF1DEBC9A        --4-7 字节的代码  
      .hword         0xD423            --8-9 字节的代码
```

备注

将以上程序示例添加到启动文件中。

第 25 章 电气特性

25.1 电气特性

最大绝对额定值 ($T_A = 25^\circ\text{C}$) (1/2)

参数	符号	条件	额定值	单位
电源电压	V_{DD}	$V_{DD} = EV_{DD}$	-0.5 至 +6.5	V
	EV_{DD}	$V_{DD} = EV_{DD}$	-0.5 至 +6.5	V
	AV_{REF0}		-0.5 至 +6.5	V
	V_{SS}	$V_{SS} = EV_{SS} = AV_{SS}$	-0.5 至 +0.5	V
	AV_{SS}	$V_{SS} = EV_{SS} = AV_{SS}$	-0.5 至 +0.5	V
	EV_{SS}	$V_{SS} = EV_{SS} = AV_{SS}$	-0.5 至 +0.5	V
输入电压	V_{I1}	P00 至 P06, P30 至 P35, P40 至 P42, P50 至 P55, P90, P91, P96 至 P99, P913 至 P915, PCM0, PCM1, PDL0 至 PDL7, $\overline{\text{RESET}}$, FLMD0	-0.5 至 $EV_{DD} + 0.5^{\#}$	V
	V_{I3}	X1, X2, XT1, XT2	-0.5 至 $V_{RO} + 0.5$	V
模拟输入电压	V_{IAN}	P70 至 P79	-0.5 至 $AV_{REF0} + 0.5^{\#}$	V

注 确保不要超过每个电源电压的最大额定值（最大值）。

- 注意事项**
1. 避免 IC 设备输出（或 I/O）引脚间的连接以及 V_{DD} 或 V_{CC} 与 GND 连接。
 2. 任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。
DC 特性和 AC 特性中指出的额定值和条件是正常工作的质量保证。
 3. 当直接将外部电路连接到变成高阻状态的引脚时，必须这样设置时序以避免外部电路上的输出冲突。

备注 除非另外说明，复用功能引脚的特性与端口引脚的相同。

最大绝对额定值 ($T_A = 25^\circ\text{C}$) (2/2)

参数	符号	条件		额定值	单位
输出电流，低	I _{OL}	P00 至 P06， P30 至 P35， P40 至 P42， P50 至 P55， P90， P91， P96 至 P99， P913 至 P915， PCM0， PCM1， PDL0 至 PDL7	每个引脚	4	mA
			所有引脚总和	50	mA
		P70 至 P79	每个引脚	4	mA
			所有引脚总和	20	mA
输出电流，高	I _{OH}	P00 至 P06， P30 至 P35， P40 至 P42， P50 至 P55， P90， P91， P96 至 P99， P913 至 P915， PCM0， PCM1， PDL0 至 PDL7	每个引脚	−4	mA
			所有引脚总和	−50	mA
		P70 至 P79	每个引脚	−4	mA
			所有引脚总和	−20	mA
工作环境温度	T _A	正常操作模式		−40 至 +85	°C
		Flash 存储器编程模式			
存储温度	T _{stg}			−40 至 +125	°C

- 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，或连接到 V_{DD} , V_{CC} , 和 GND 。
 2. 任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。
DC 特性和 AC 特性中指出的额定值和条件是正常工作的质量保证。
 3. 当直接将外部电路连接到变成高阻状态的引脚时，必须这样设置时序以避免外部电路上的输出冲突。

备注 除非另外说明，复用功能引脚的特性与端口引脚的相同。

25.2 电容

(TA = 25°C, VDD = EVDD = AVREF0 = VSS = EVSS = AVSS = 0 V)

参数	符号	条件	最小.	典型值.	最大.	单位
I/O 电容	C _{IO}	f _x = 1 MHz, 不可测量的引脚返回 0 V。			10	pF

25.3 工件条件

(TA = 40 至 +85°C, VDD = EVDD = 3.5 V 至 5.5 V, 4.0 V ≤ AVREF0 ≤ 5.5 V, VSS = EVSS = AVSS = 0 V)

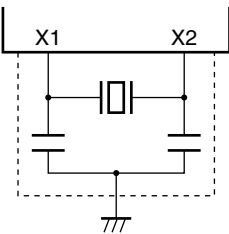
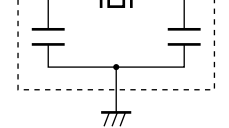
参数	符号	条件	最小.	典型值.	最大.	单位
内部系统时钟频率	f _{CLK}	REGC = 4.7 μF, 通过主时钟操作时	4		20	MHz
		REGC = 4.7 μF, 通过副时钟操作时 (晶体谐振器)	32		35	kHz
		REGC = 4.7 μF, 通过副时钟操作时 (RC 谐振器)	12.5 [※]		27.5 [※]	kHz

注 内部系统时钟频率是振荡频率的一半。

25.4 振荡器特性

25.4.1 主时钟振荡器特性

($T_A = 40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

谐振器	推荐电路	参数	条件	最小.	典型值.	最大.	单位
陶瓷谐振器		振荡频率 (fx) ^{註 1}		4		5	MHz
		振荡稳定时间 ^{註 2}	复位释放后		$2^{16}/f_x$		s
			STOP 模式释放后	0.5 ^{註 3}	註 4		ms
			IDLE2 模式释放后	0.35 ^{註 3}	註 4		ms
晶体谐振器		振荡频率 (fx) ^{註 1}		4		5	MHz
		振荡稳定时间 ^{註 2}	复位释放后		$2^{16}/f_x$		s
			STOP 模式释放后	0.5 ^{註 3}	註 4		ms
			IDLE2 模式释放后	0.35 ^{註 3}	註 4		ms

- 注
1. 只显示振荡器特性。
 2. 复位或 STOP 模式被释放后振荡稳定所需的时间。
 3. 访问稳定内部 Flash 存储器所需的时间。
 4. 值根据 OSTS 寄存器设置的不同而改变。

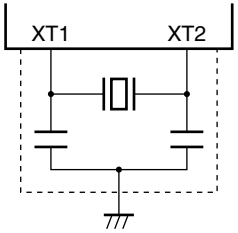
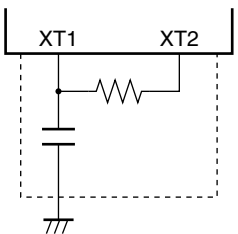
注意事项 1. 当使用主时钟振荡器，上图中被虚线包围的部分的配线应如下布线方法布线，以防止连接电容产生不利影响。

- 连接线越短越好。
- 连接线不应与其他信号线交叉。
- 流经的电流变化较大的信号线不要在振荡器周围布线。
- 要保持振荡器电容器的接地点电压与 V_{SS} 相同。
- 不要将电容的地信号接入大电流地。
- 不要从振荡器获取信号。

2. 当主时钟停止而副时钟运行时，在转换回主时钟前将一直等待直到振荡稳定时间由程序保证为止。

25.4.2 副时钟振荡器特性

($T_A = 40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

谐振器	推荐电路	参数	条件	最小.	典型值.	最大.	单位
晶体谐振器		振荡频率 (f_{XT}) ^{注 1}		32	32.768	35	kHz
		振荡稳定时间 ^{注 2}				10	s
RC 谐振器		振荡频率 ^{注 1, 4}	$R = 390\text{ k}\Omega \pm 5\%$ ^{注 3} $C = 47\text{ pF} \pm 10\%$ ^{注 3}	25	40	55	kHz
		振荡稳定时间 ^{注 2}				100	μs

- 注
1. 只显示振荡器特性。关于 CPU 操作时钟的详细信息，参见 **25.8 AC 特性**。
 2. 从 V_{DD} 达到振荡电压范围（最小.: 3.5 V ）开时直到振荡稳定所需的时间。
 3. 为了避免连接线电容所产生的不利影响，应使连接线的长度尽可能的短。
 4. RC 振荡频率为 40 kHz （典型值.）。该时钟在内部被除以 2。使用 RC 谐振器时，内部系统时钟频率是振荡频率的一半；最小. = 12.5 kHz ，典型值. = 20 kHz ，最大. = 27.5 kHz 。

注意事项 1. 当使用副时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。

- 连接线越短越好。
 - 连接线不应与其他信号线交叉。
 - 流经的电流变化较大的信号线不要在振荡器周围布线。
 - 要保持振荡器电容器的接地点电压与 V_{SS} 相同。
 - 不要将电容的地信号接入大电流地。
 - 不要从振荡器获取信号。
2. 为了缩减能耗，副时钟振荡器被设计为低幅电路，并且它比主时钟振荡器更容易受到噪声的影响。因此在使用副时钟时更要注意布线方法。

25.4.3 PLL 特性

($T_A = 40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小.	典型值.	最大.	单位
输入频率	f_x		4		5	MHz
输出频率	f_{xx}		16		20	MHz
锁时间	t_{PLL}	V_{DD} 达到最小.: 3.5 V 后			800	μs

25.4.4 内部振荡器特性

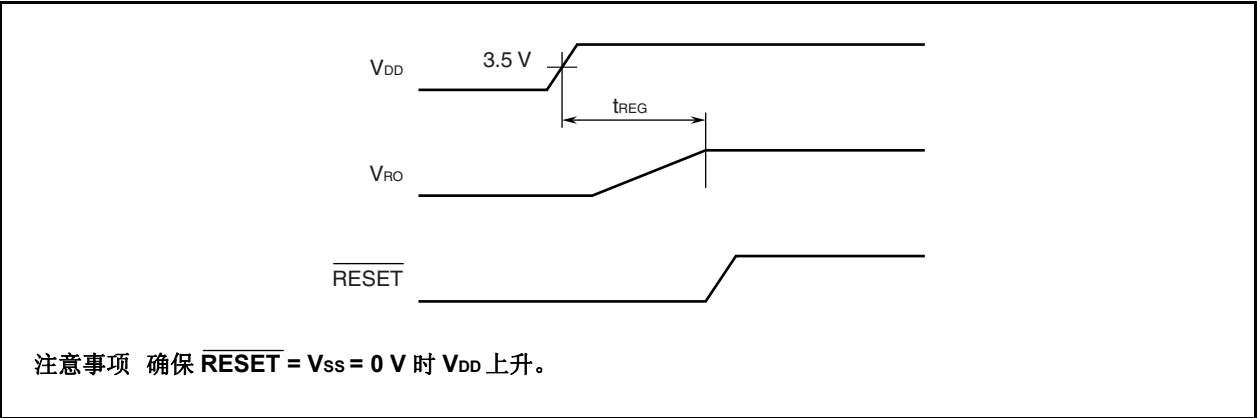
($T_A = 40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小.	典型值.	最大.	单位
输出频率	f_R		100	200	400	kHz

25.5 电压调节器特性

($T_A = 40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小.	典型值.	最大.	单位
输入频率	V_{DD}		3.5		5.5	V
输出频率	V_{RO}			2.5		V
锁时间	t_{REG}	V_{DD} 达到最小.: 3.5 V 后, $C = 4.7\text{ }\mu\text{F} \pm 20\%$ 连接到 REGC 引脚			1	ms



25.6 DC 特性

25.6.1 I/O 电平

($T_A = 40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

(1/2)

参数	符号	条件	最小.	典型值.	最大.	单位
输入电压, 高	V_{IH1}	P30, P34, P41, P98, PCM0, PCM1, PDL0 至 PDL7	$0.7EV_{DD}$		EV_{DD}	V
	V_{IH2}	P00 至 P06, P31 至 P33, P35, P40, P42, P50 至 P55, P90, P91, P96, P97, P99, P913 至 P915	$0.8EV_{DD}$		EV_{DD}	V
	V_{IH4}	P70 至 P79	$0.7AV_{REF0}$		AV_{REF0}	V
	V_{IH5}	RESET, FLMD0	$0.8EV_{DD}$		EV_{DD}	V
输入电压, 低	V_{IL1}	P30, P34, P41, P98, PCM0, PCM1, PDL0 至 PDL7	EV_{SS}		$0.3EV_{DD}$	V
	V_{IL2}	P00 至 P06, P31 至 P33, P35, P40, P42, P50 至 P55, P90, P91, P96, P97, P99, P913 至 P915	EV_{SS}		$0.2EV_{DD}$	V
	V_{IL4}	P70 至 P79	AV_{SS}		$0.3AV_{REF0}$	V
	V_{IL5}	RESET, FLMD0	EV_{SS}		$0.2EV_{DD}$	V

备注 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

(T_A = 40 至 +85°C, V_{DD} = E_{VDD} = 3.5 V 至 5.5 V, 4.0 V ≤ A_{VREF0} ≤ 5.5 V, V_{SS} = E_{VSS} = A_{VSS} = 0 V)

(2/2)

参数	符号	条件	最小.	典型值.	最大.	单位
输出电压, 高 ^{注 1}	V _{OH1}	P00 至 P06, P30 至 P35, P40 至 P42, P50 至 P55, P90, P91, P96 至 P99, P913 至 P915, PCM0, PCM1, PDL0 至 PDL7	I _{OH} = -1.0 mA	E _{VDD} - 1.0	E _{VDD}	V
			I _{OH} = -0.1 mA	E _{VDD} - 0.5	E _{VDD}	V
	V _{OH3}	P70 至 P79	I _{OH} = -1.0 mA	A _{VREF0} - 1.0	A _{VREF0}	V
			I _{OH} = -0.1 mA	A _{VREF0} - 0.5	A _{VREF0}	V
输出电压, 低 ^{注 1}	V _{OL1}	P00 至 P06, P30 至 P35, P40 至 P42, P50 至 P55, P90, P91, P96 至 P99, P913 至 P915, PCM0, PCM1, PDL0 至 PDL7	I _{OL} = 1.0 mA	0	0.4	V
	V _{OL3}	P70 至 P79	I _{OL} = 1.0 mA	0	0.4	V
上拉电阻	R ₁	V _I = 0 V	10	30	100	kΩ
下拉电阻 ^{注 2}	R ₂	V _I = V _{DD}	10	30	100	kΩ

注 1. 对于每个电源 (E_{VDD}, A_{VREF0}) 来说所有 I_{OH}/I_{OL} 中的最大值为 20 mA/-20 mA。
 2. 只有 DRST 引脚

备注 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

25.6.2 引脚泄漏电流

(T_A = 40 至 +85°C, V_{DD} = E_{VDD} = 3.5 V 至 5.5 V, 4.0 V ≤ A_{VREF0} ≤ 5.5 V, V_{SS} = E_{VSS} = A_{VSS} = 0 V)

参数	符号	条件	最小.	典型值.	最大.	单位
输入泄漏电流, 高 ^注	I _{LIH1}	V _{IN} = V _{DD}	模拟引脚		+0.2	μA
			不是模拟引脚		+0.5	
输入泄漏电流, 低 ^注	I _{LIL1}	V _{IN} = 0 V	模拟引脚		-0.2	μA
			不是模拟引脚		-0.5	
输出泄漏电流, 高	I _{LOH1}	V _O = V _{DD}	模拟引脚		+0.2	μA
			不是模拟引脚		+0.5	
输出泄漏电流, 低	I _{LOL1}	V _O = 0 V	模拟引脚		-0.2	μA
			不是模拟引脚		-0.5	

注 FLMD0 引脚的值如下所示。
 • 输入泄漏电流, 高: 2 μA (最大.)
 • 输入泄漏电流, 低: -2 μA (最大.)

25.6.3 电源电流

(TA = 40 至 +85°C, VDD = EVDD = 3.5 V 至 5.5 V, 4.0 V ≤ AVREF0 ≤ 5.5 V, VSS = EVSS = AVSS = 0 V)

参数	符号	条件		最小.	典型值.	最大.	单位	
电源电流 ^{註 1}	IDD1	正常操作模式	fxx = 20 MHz (fx = 5 MHz)	运行所有外围功能		25	40	mA
				停止所有外围功能		20		mA
	IDD2	HALT 模式	fxx = 20 MHz (fx = 5 MHz)	运行所有外围功能		14	24	mA
					停止所有外围功能		9	
	IDD3	IDLE1 模式	fxx = 5 MHz (fx = 5 MHz) , PLL 关			0.6	0.9	mA
	IDD4	IDLE2 模式	fxx = 5 MHz (fx = 5 MHz) , PLL 关			0.25	0.7	mA
	IDD5	副时钟操作模式 ^{註 2, 3}	晶体谐振器 (fxT = 32.768 kHz)			200	400	μA
			RC 谐振器 (fxT = 40 kHz ^{註 4})			200	400	μA
	IDD6	副 IDLE 模式 ^{註 2, 3}	晶体谐振器 (fxT = 32.768 kHz)			20	120	μA
			RC 谐振器 (fxT = 40 kHz ^{註 4})			35	140	μA
	IDD7	Stop 模式 ^{註 2, 5}	POC 停止, 内部振荡器停止			7	50	μA
			POC 运行, 内部振荡器停止			10	55	μA
			POC 停止, 内部振荡器运行			15	65	μA
			POC 运行, 内部振荡器运行			18	70	μA

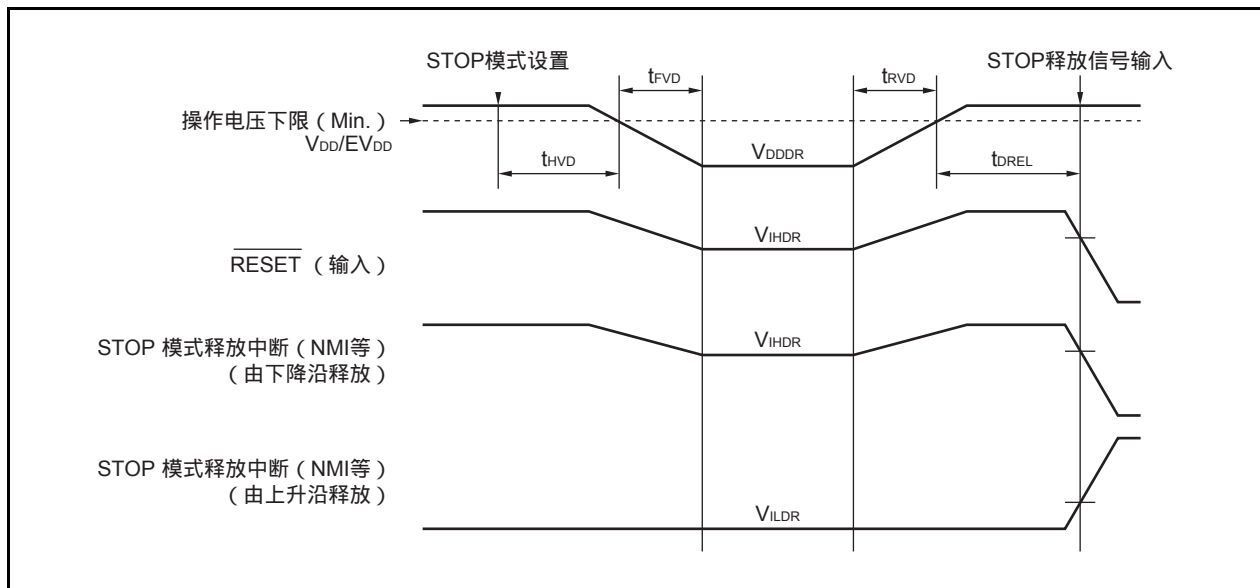
- 注
1. VDD 和 EVDD 的总电流（所有端口都被停止）。不包括 AVREF0 的电流以及包括流过片上上拉/下拉电阻的电流的端口缓冲电流。
 2. 当主时钟振荡停止时。
 3. PCK 运行, 内部振荡器运行。
 4. RC 振荡频率为 40 kHz（典型值.）。该时钟在内部被除以 2。
 5. 没有使用副时钟振荡时。

25.7 数据保持特性

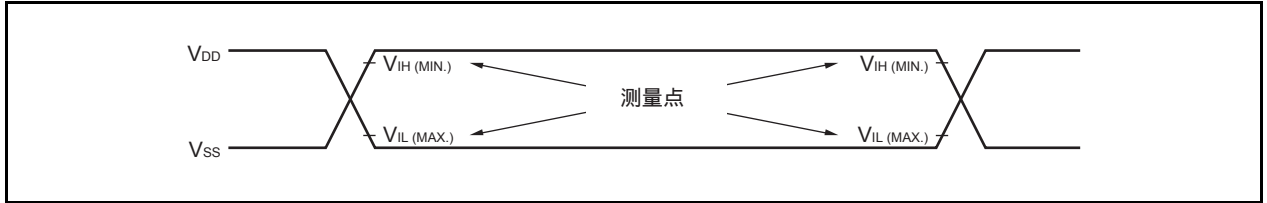
STOP 模式 ($T_A = 40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 1.9\text{ V}$ 至 5.5 V , $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小.	典型值.	最大.	单位
数据保持电压	V_{DDDR}	STOP 模式中 (所有功能被停止)	1.9		5.5	V
数据保持电流	I_{DDDR}	$V_{DDDR} = 2.0\text{ V}$ (所有功能被停止)		6	45	μA
电源电压上升时间	t_{rVD}		1			μs
电源电压下降时间	t_{fVD}		1			μs
电源电压保持时间	t_{HVD}	STOP 模式释放后	0			ms
STOP 释放信号输入时间	t_{DREL}	V_{DD} 达到最小.: 3.5 V 后	0			ms
数据保持输入电压, 高	V_{IHDR}	所有输入端口	$0.9V_{DDDR}$		V_{DDDR}	V
数据保持输入电压, 低	V_{ILDR}	所有输入端口	0		$0.1V_{DDDR}$	V

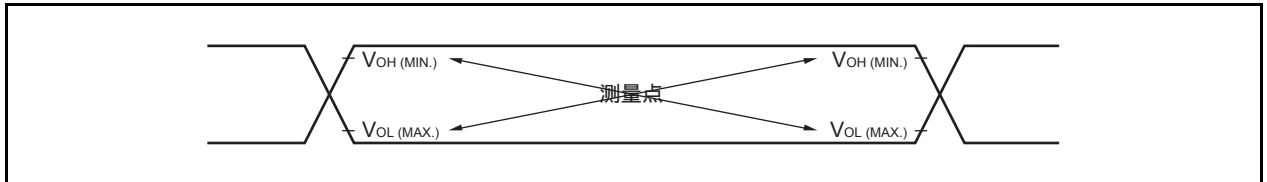
注意事项 转换到 STOP 模式并从 STOP 模式还原必须在额定的工作范围完成。



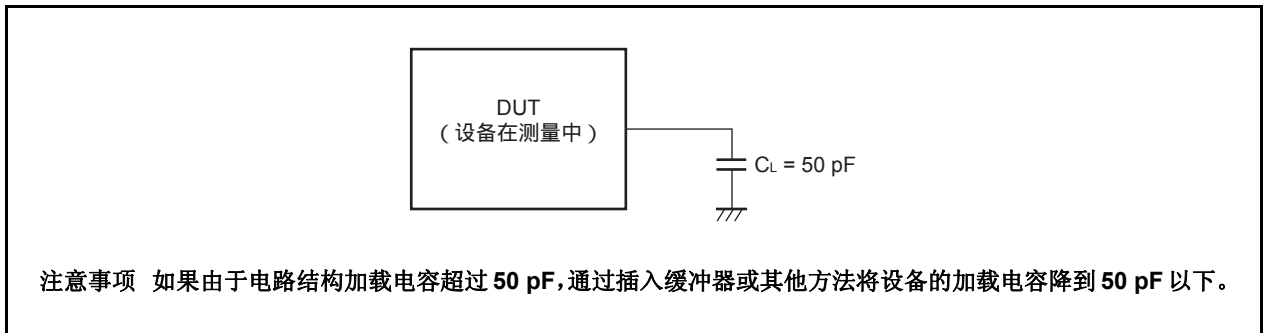
25.8 AC 特性

(1) AC 输入测试点 (V_{DD} , AV_{REF0} , EV_{DD})

(2) AC 输出测试点



(3) 加载条件

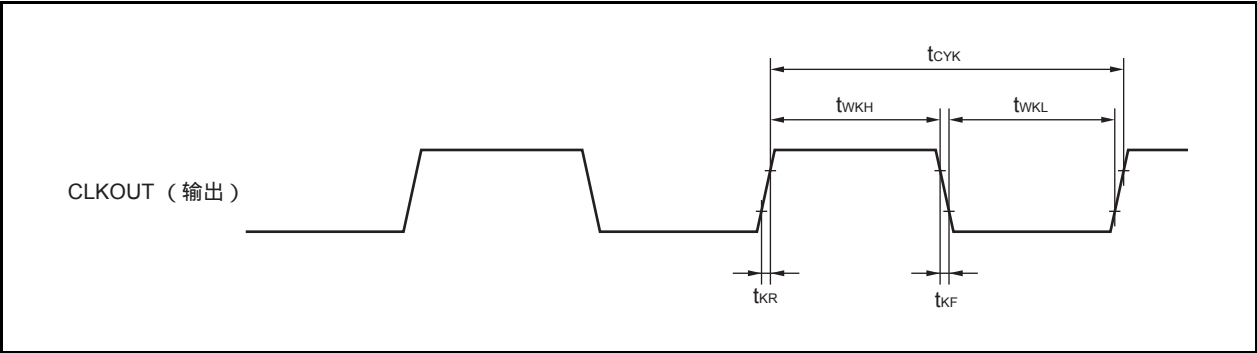


25.8.1 CLKOUT 输出时序

($T_A = 40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小.	最大.	单位
输出周期	t_{CYK}		50 ns	80 μs	
高电平宽度	t_{WKH}		$t_{CYK}/2 - 15$		ns
低电平宽度	t_{WKL}		$t_{CYK}/2 - 15$		ns
上升时间	t_{KR}			15	ns
下降时间	t_{KF}			15	ns

时钟时序



25.9 基本操作

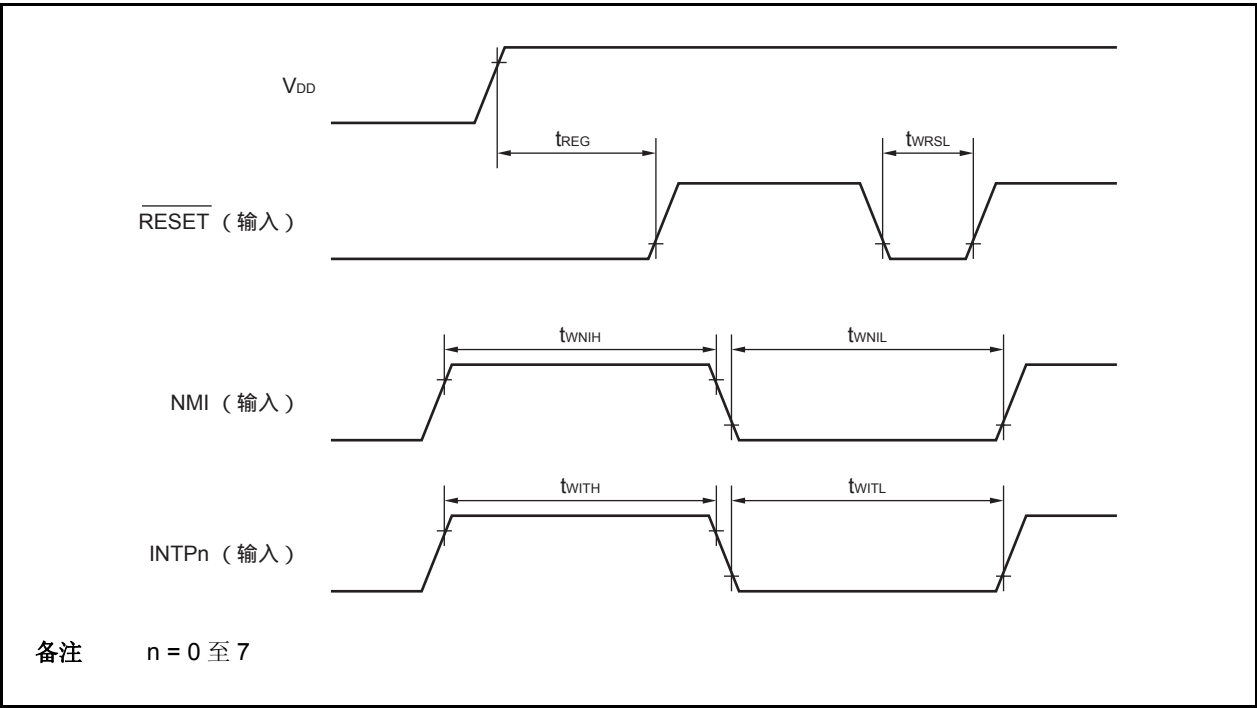
(1) 复位，中断时序

($T_A = 40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小.	最大.	单位
$\overline{\text{RESET}}$ 低电平宽度	t_{WRSL}		500		ns
NMI 高电平宽度	t_{WNIH}	模拟噪声消除	500		ns
NMI 低电平宽度	t_{WNIL}	模拟噪声消除	500		ns
INTPn ^{※1} 高电平宽度	t_{WITH}	模拟噪声消除 (n = 0 至 7)	500		ns
		数字噪声消除 (n = 3)	注 2		ns
INTPn ^{※1} 低电平宽度	t_{WITL}	模拟噪声消除 (n = 0 至 7)	500		ns
		数字噪声消除 (n = 3)	注 2		ns

- 注 1. 在使用 ADTRG 引脚的情况下会应用与 INTP0/P03 引脚相同的值。在使用 $\overline{\text{DRST}}$ 引脚的情况下则会应用与 INTP2/P05 引脚相同的值。
2. $2T_{\text{samp}} + 20$ 或 $3T_{\text{samp}} + 20$
 T_{samp} : 用于噪声消除的采样时钟

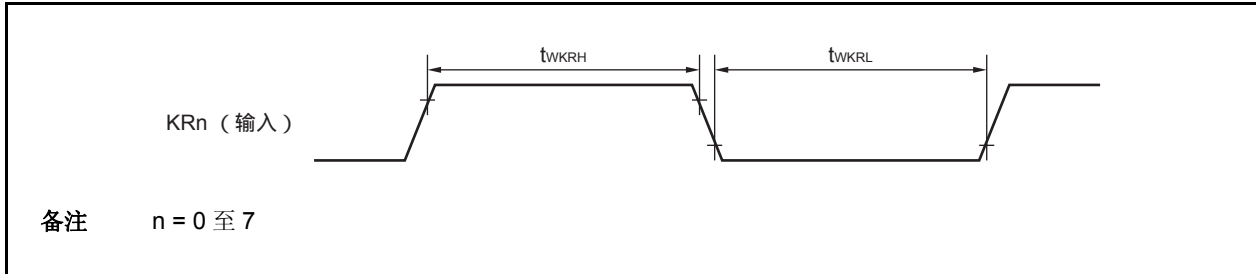
复位/中断



(2) 按键中断时序

($T_A = 40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小.	最大.	单位
KRn 输入高电平宽度	t_{WKRH}	模拟噪声消除 ($n = 0$ 至 7)	500		ns
KRn 输入低电平宽度	t_{WKRL}		500		ns

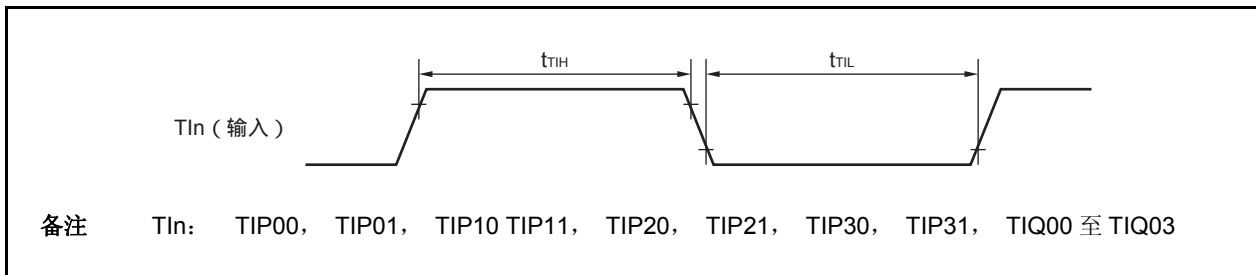


(3) 定时器输入时序

($T_A = 40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小.	最大.	单位
TIn 高电平宽度	t_{TIH}	TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, TIP31, TIQ00 至 TIQ03 ^{注 1}	注 2		ns
TIn 低电平宽度	t_{TIL}		注 2		ns

- 注
1. TIP00, TIP10, TIP20, TIP30 以及 TIQ00 引脚上的噪声只有在捕获信号被输入时才能消除。
当外部触发信号或外部事件计数信号被输入时噪声将不能被消除。
 2. $2T_{\text{samp}} + 20$ 或 $3T_{\text{samp}} + 20$
 T_{samp} : 用于噪声消除的采样时钟



(4) CSIB 时序

(a) 主模式

($T_A = 40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小.	最大.	单位
SCKBn 循环时间	t_{KCYn}		125		ns
SCKBn 高电平宽度	t_{KHn}		$t_{KCYn}/2 - 15$		ns
SCKBn 低电平宽度	t_{KLn}		$t_{KCYn}/2 - 15$		ns
SIBn 建立时间 (至 SCKBn \uparrow)	t_{SIKn}		30		ns
SIBn 保持 (从 SCKBn \uparrow)	t_{SIn}		25		ns
从 SCKBn \downarrow 至 SOBn 的输出延迟时间	t_{KSON}			25	ns

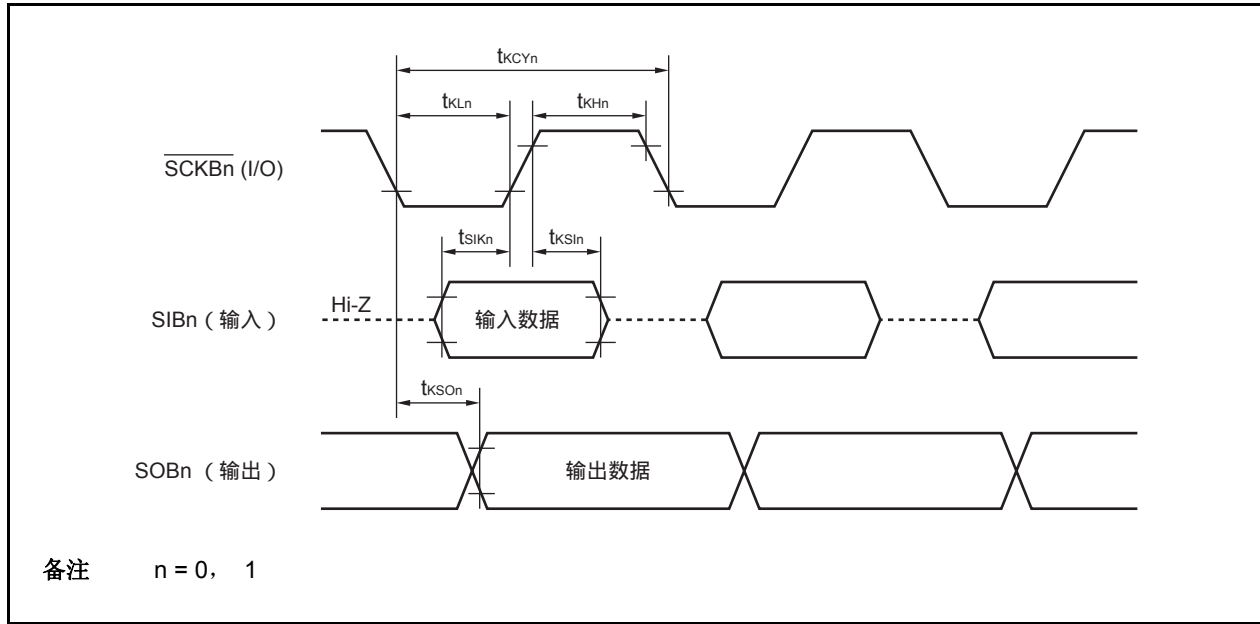
备注 $n = 0, 1$

(b) 从模式

($T_A = 40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小.	最大.	单位
SCKBn 循环时间	t_{KCYn}		200		ns
SCKBn 高电平宽度	t_{KHn}		90		ns
SCKBn 低电平宽度	t_{KLn}		90		ns
SIBn 建立时间 (至 SCKBn \uparrow)	t_{SIKn}		50		ns
SIBn 保持时间 (从 SCKBn \uparrow)	t_{SIn}		50		ns
从 SCKBn \downarrow 到 SOBn 的输出延迟时间	t_{KSON}			50	ns

备注 $n = 0, 1$



(5) UARTA 时序

($T_A = 40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小.	最大.	单位
通信速率				312.5	kbps
ASCK0 循环时间				10	MHz

(6) A/D 转换器

($T_A = 40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小.	典型值.	最大.	单位
分辨率					10	位
总误差 [✱]		$4.0 \leq AV_{REF0} \leq 5.5\text{ V}$		± 0.15	± 0.3	%FSR
置换时间	t_{CONV}		3.1		16	μs
模拟输入电压	V_{IAN}		AV_{SS}		AV_{REF0}	V
AVREF0 电流	I_{AREF0}	使用 A/D 转换器时		5	10	mA
		不使用 A/D 转换器时		1	10	μA

注 不包括量化误差 ($\pm 0.05\%$ FSR)。表明满度值比率 (%FSR)。

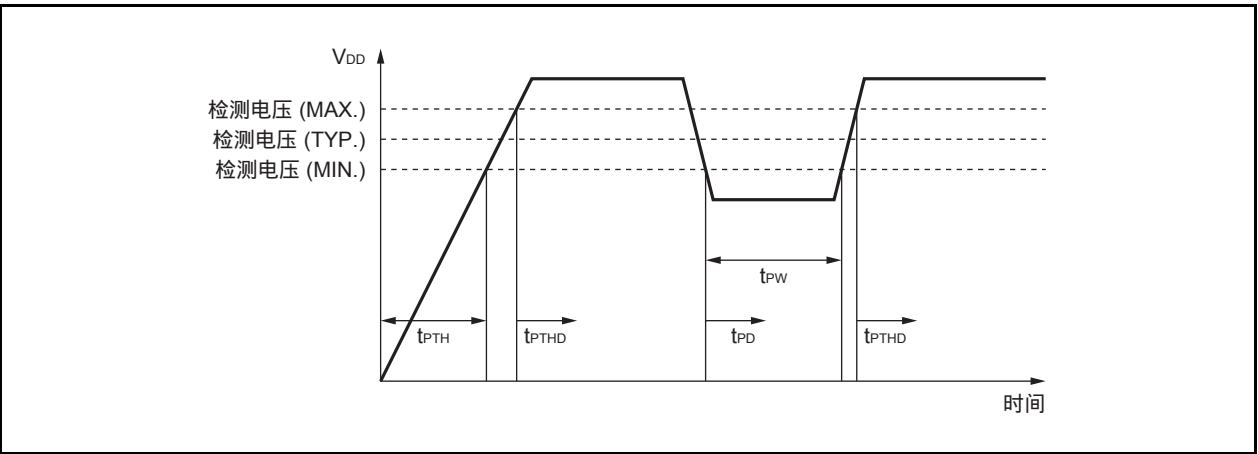
备注 FSR: 满度范围

(7) POC 电路特性

($T_A = 40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小.	典型值.	最大.	单位
检测电压	V_{POC0}		3.5	3.7	3.9	V
电源启动时间	t_{PTH}	$V_{DD} = 0\text{ V} \rightarrow 3.5\text{ V}$	0.002			ms
响应延迟时间 1 ^{注 1}	t_{PTHD}	电源应用时 V_{DD} 达到 3.9 V 后			3.0	ms
响应延迟时间 2 ^{注 2}	t_{PD}	电源下降时 V_{DD} 降到 3.5 V 以下后			1	ms
最小 V_{DD} 宽度	t_{PW}		0.2			ms

- 注
1. 检测到检测电压后释放一个复位所需的时间。
 2. 检测到检测电压后输出一个复位所需的时间。

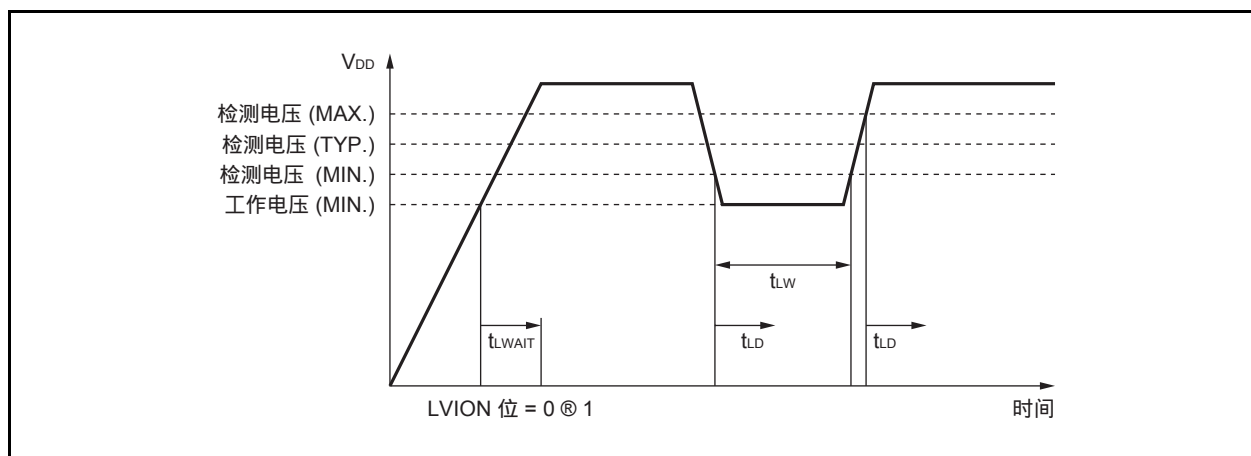


(8) LVI 电路特性

($T_A = 40$ 至 $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小.	典型值.	最大.	单位
检测电压	V_{LV10}		4.2	4.4	4.6	V
	V_{LV11}		4.0	4.2	4.4	V
响应时间 ^{注 1}	t_{LD}	在 V_{DD} 达到 V_{LV10}/V_{LV11} (最大.) 后 或 V_{DD} 下降到 V_{LV10}/V_{LV11} (最小.) 后		0.2	2	ms
最小 V_{DD} 宽度	t_{LW}		0.2			ms
参考电压稳定等待时间 ^{注 2}	t_{LWAIT}	V_{DD} 达到 3.5 V 后或 $LVION$ 位 ($LVIM$.位 7) 从 0 变为 1 后		0.1	0.2	ms

- 注 1. 检测到检测电压后输出一个中断/复位所需的时间。
2. 使用 POC 功能时不需要。

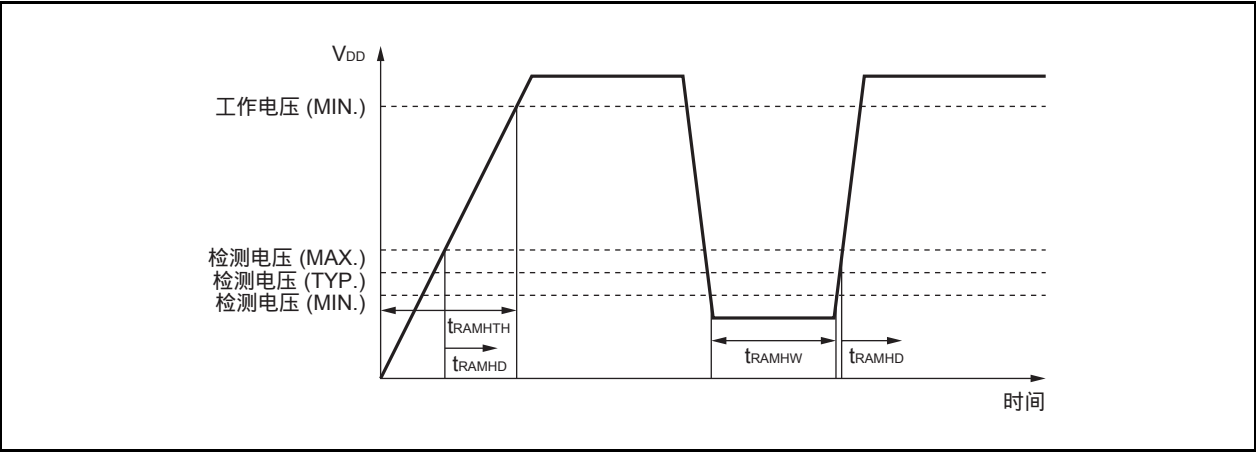


(9) RAM 保持标志特性

($T_A = 40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小.	典型值.	最大.	单位
检测电压	V_{RAMH}		1.9	2.0	2.1	V
电源电压上升时间	t_{RAMHTH}	$V_{DD} = 0\text{ V} \rightarrow 3.5\text{ V}$	0.002		1800	ms
响应时间 ^注	t_{RAMHD}	电源电压达到检测电压（最大.）后		0.2	2.0	ms
最小 V_{DD} 宽度	t_{RAMHW}		0.2			ms

注 检测到检测电压后设置 RAMF 位所需的时间。



25.10 Flash 存储器编程特性

(1) 基本特性

($T_A = 40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小.	典型值.	最大.	单位
工作频率	f_{CPU}		4		20	MHz
电源电压	V_{DD}		3.5		5.5	V
写入次数	$C_{WRT}^{\#}$				100	Times
输入电压, 高	V_{IH}	FLMD0	$0.8EV_{DD}$		EV_{DD}	V
输入电压, 低	V_{IL}	FLMD0	EV_{SS}		$0.2EV_{SS}$	V
写入时间 + 擦除时间	$t_{WRT} + t_{ERASE}$				TBD	s
编程温度	t_{PRG}		-40		+85	$^{\circ}\text{C}$

注 当对出厂产品进行初次写入时, 将“擦除写入”和“只写入”两个过程按一次重写来计数。

例 (P: 写入, E: 擦除)

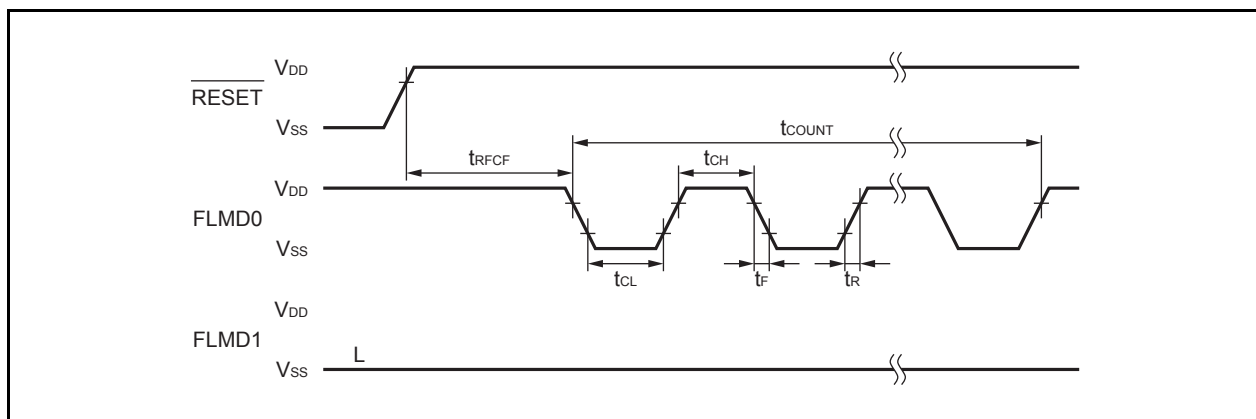
出厂产品 $\rightarrow P \rightarrow E \rightarrow P \rightarrow E \rightarrow P$: 3 次重写

出厂产品 $\rightarrow E \rightarrow P \rightarrow E \rightarrow P \rightarrow E \rightarrow P$: 3 次重写

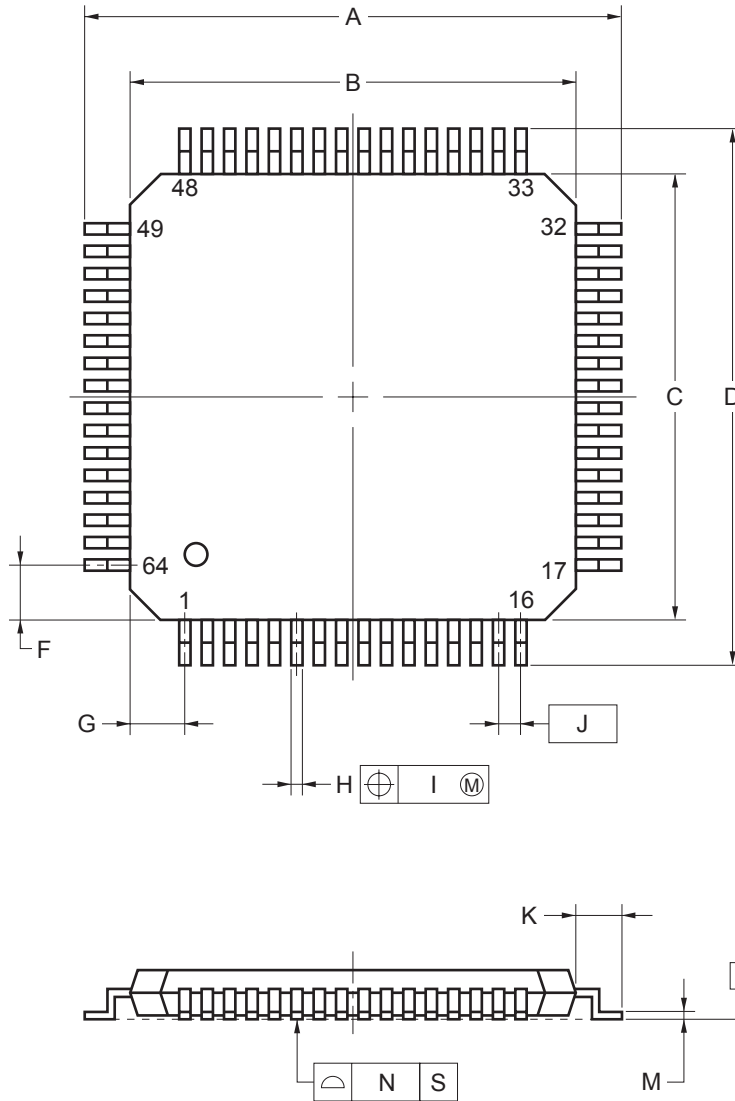
(2) 串行写入操作特性

($T_A = 40$ 至 $+85^{\circ}\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V}$ 至 5.5 V , $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

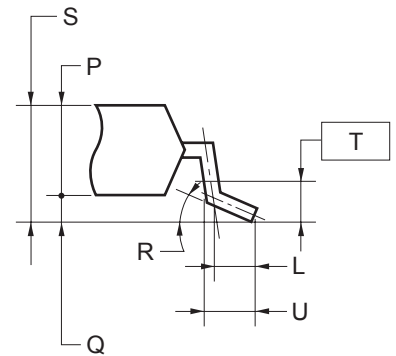
参数	符号	条件	最小.	典型值.	最大.	单位
从 $\overline{\text{RESET}}\uparrow$ 开始的 FLMD0 建立时间	t_{RFCF}		70536/ f_x			s
计数执行时间	t_{COUNT}				3	ms
FLMD0 高电平宽度	t_{CH}		10		100	μs
FLMD0 低电平宽度	t_{CL}		10		100	μs
FLMD0 上升时间	t_R				50	ns
FLMD0 下降时间	t_F				50	ns



64引脚塑封LQFP（10x10）



引脚详端图



尺寸

该结构在最大使用材料情况下，每条引脚的中心线位于其实际位置的0.08mm(T.P.) 内。

项目	尺寸
A	12.0±0.2
B	10.0±0.2
C	10.0±0.2
D	12.0±0.2
F	1.25
G	1.25
H	0.22±0.05
I	0.08
J	0.5 (T.P.)
K	1.0±0.2
L	0.5
M	0.17 ^{+0.03} _{-0.07}
N	0.08
P	1.4
Q	0.1±0.05
R	3° ^{+4°} _{-3°}
S	1.5±0.10
T	0.25
U	0.6±0.15

S64GB-50-8EU-2

The V850ES/HE2 应在如下推荐条件下焊接和装配。

关于技术信息，请参照下面的网址。

半导体设备装配手册 (<http://www.necel.com/pkg/en/mount/index.html>)

表 27-1. 贴装焊接条件

μPD70F3700GB-8EU-A: 64-引脚塑封 LQFP (密脚距) (10 × 10)

μPD70F3701GB-8EU-A: 64-引脚塑封 LQFP (密脚距) (10 × 10)

焊接方法	焊接条件	推荐条件符号
红外线	包装最高温度: 260°C, 时间: 最多 60 秒 (220°C 时或更高), 计数: 三次或更少, 暴露限制: 7 天 [※] (之后在 125°C 预烘 20 至 72 个小时)	IR60-207-3
局部加热	引脚温度: 最高 350°C, 时间: 最多 3 秒 (每引脚排)	—

注 打开干燥的包装后, 存储在低于 25°C 和 65% 相对湿度的条件下, 或减少存储时间。

注意事项 不要同时使用不同的焊接方式 (除局部加热外)。

备注

1. 部件号码以 -A 结尾的产品为无铅产品。
2. 有关不同于以上所推荐的焊接方法和条件, 请与日电销售代理联系。

在使用 V850ES/HE2 的系统开发中可应用如下开发工具。

图 A-1 所示为开发工具的组成。

- **支持 PC98-NX 系列**

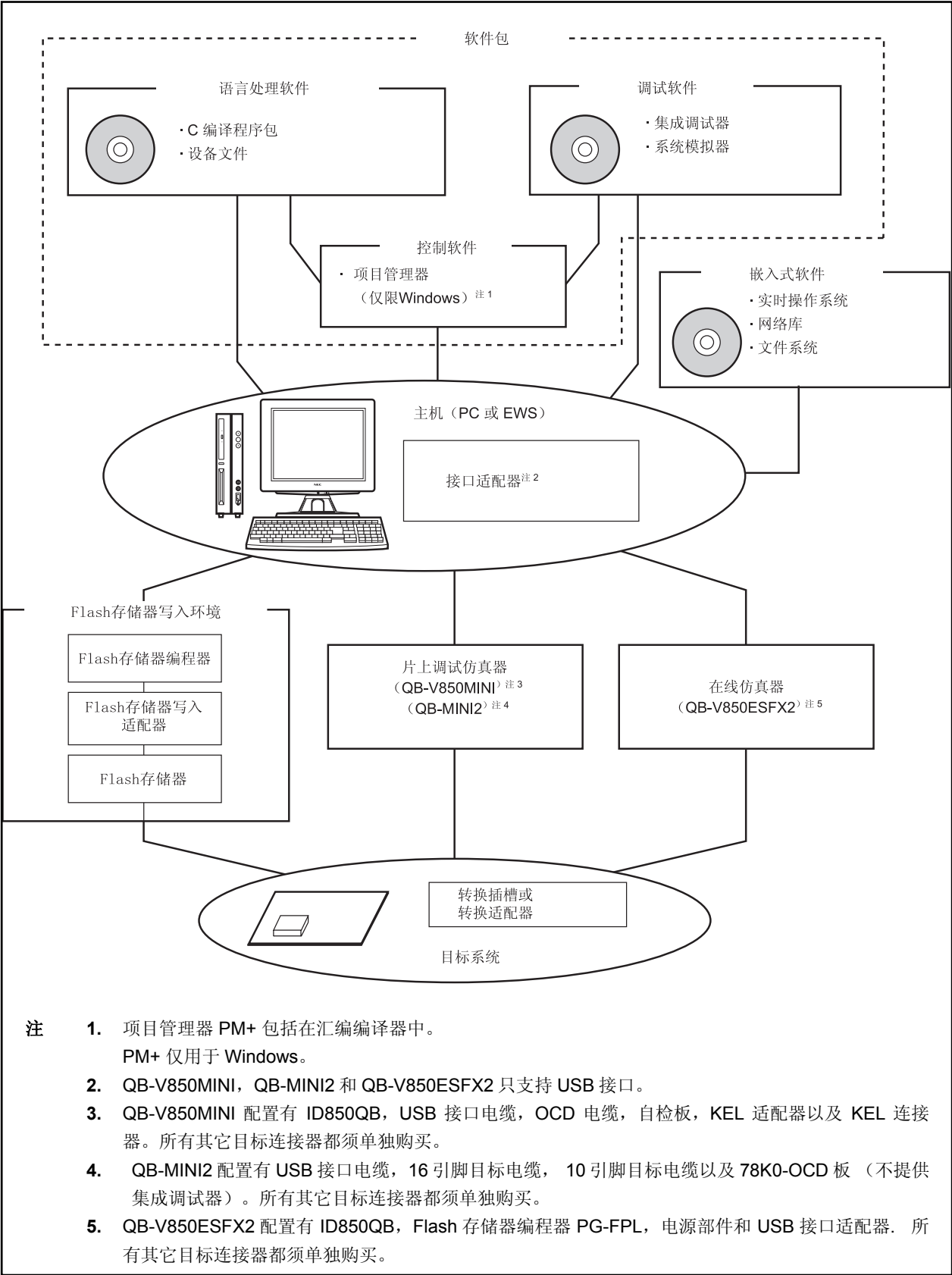
除非特别说明，IBM PC/AT™ 兼容机支持的产品与 PC98-NX 系列计算机是兼容的。当使用 PC98-NX 系列计算机时，参阅 IBM PC/AT 兼容机的使用说明。

- **Windows™**

除非特别说明，“Windows”指以下几种操作系统。

- Windows 98, 2000
- Windows Me
- Windows XP
- Windows NT™ Ver. 4.0

图 A-1. 开发工具的组成



A.1 软件包

SP850 V850 微控制器的软件包	该软件包包含适用于 V850 微控制器的开发工具（软件）。
	产品型号： μ SxxxxSP850

备注 产品型号中的 xxxx 随主机和使用的 OS 而变化。

μ SxxxxSP850

xxxx	主机	OS	存储介质
AB17	PC-9800 系列， IBM PC/AT 兼容机	Windows（日文版）	CD-ROM
BB17		Windows（英文版）	

A.2 语言处理软件

CA850 C 编译程序包	此编译器将 C 语言程序转换为微控制器可执行的目标代码。此编译器由项目管理器 PM+ 启动。
	产品型号： μ SxxxxCA703000
DF703724 设备文件	该文件包含设备特有的信息。 该设备文件应结合工具（CA850，SM+ for V850ES/Hx2,或 ID850QB）一起使用。 相应的 OS 和主机随使用工具而变化。

备注 产品型号中的 xxxx 随主机和使用的 OS 而变化。

μ SxxxxCA703000

xxxx	主机	OS	存储介质
AB17	PC-9800 系列， IBM PC/AT 兼容机	Windows（日文版）	CD-ROM
BB17		Windows（英文版）	
3K17	SPARCstation™	SunOS™（Rel. 4.1.4）， Solaris™（Rel. 2.5.1）	

A.3 控制软件

PM+ 项目管理器	这是一个控制软件，可使用户在 Windows 环境下有效地进行软件开发。用户程序开发中进行的任何操作，如启动编辑器、构造程序和启动调试器，都可以由 PM+ 执行。 <注意事项> PM+ 包含在 C 编译器软件包 CA850。 仅在 Windows 下使用。
--------------	--

A.4 调试工具（硬件）

A.4.1 使用 IECUBE QB-V850ESFX2 时

QB-V850ESFX2 连接到主机（PC-9821 系列，PC/AT 兼容机）时的系统配置如下所示。即使没有可选产品，也可以进行连接。

图 A-2. 系统配置（使用 QB-V850ESFX2 时）（1/2）

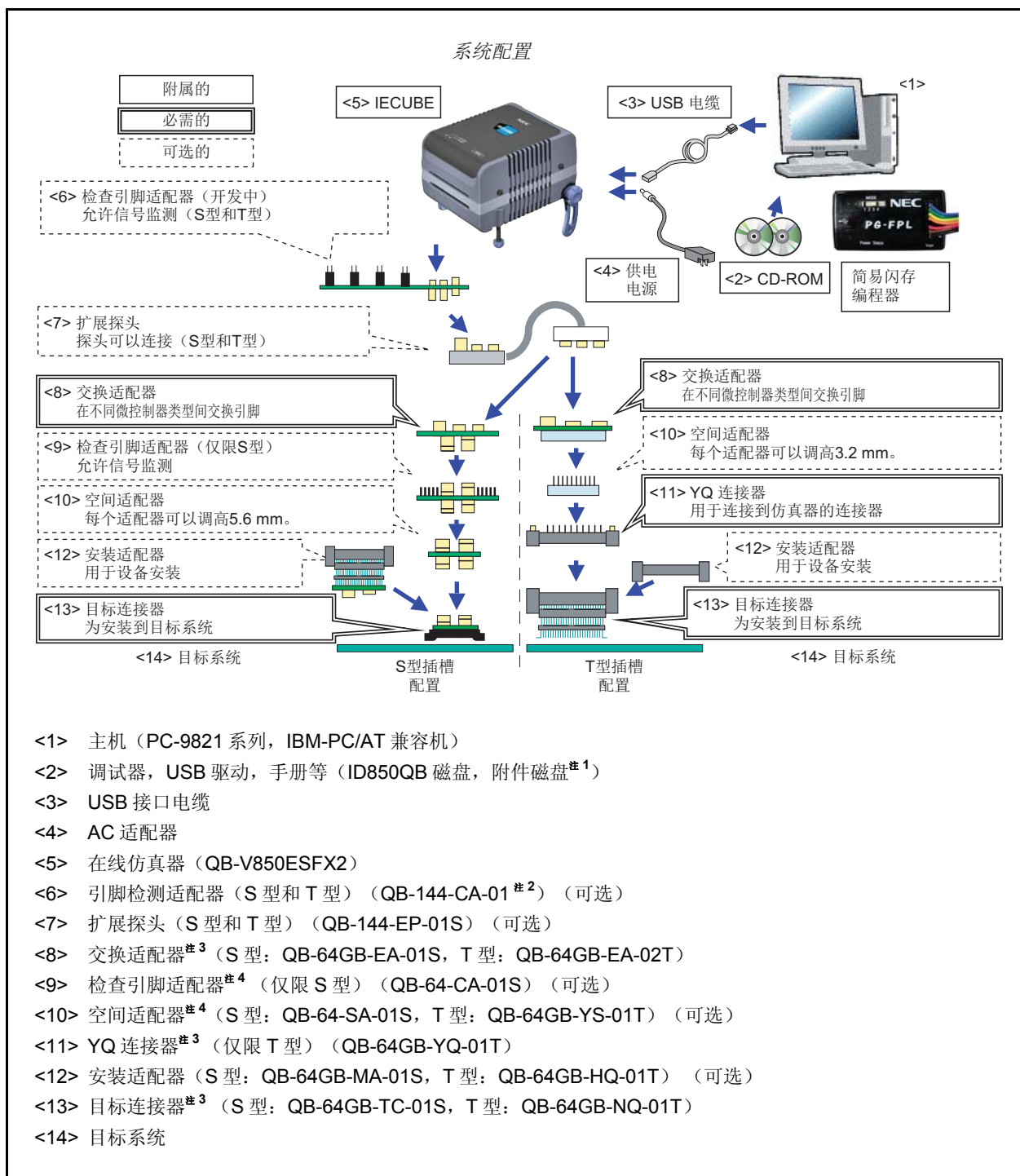


图 A-2. 系统配置（使用 QB-V850ESFX2 时）（2/2）

- 注
1. 从 NEC Electronics 网下载设备文件。
<http://www.necel.com/micro/ods/eng/>
 2. 开发中
 3. 依照订购的数字提供设备。
 - 订购 QB-V850ESFX2-ZZZ 时
不提供交换适配器和目标连接器。
 - 订购 QB-V850ESFX2-S64GB 时
提供 QB-64GB-EA-01S 和 QB-64GB-TC-01S。
 - 订购 QB-V850ESFX2-T64GB 时
提供 QB-64GB-EA-02T, QB-64GB-YQ-01T 以及 QB-64GB-NQ-01T。
 4. 当同时使用<9>和<10>时，不必关心<9>和<10>之间的顺序。

<5> QB-V850ESFX2 [※] 在线仿真器	当使用 V850ES/HE2 系列产品进行应用系统开发时，在线仿真器用来调试硬件和软件。它支持整合调试工具 ID850QB。该仿真器应与电源和仿真探头结合使用。通过 USB 接口电缆将仿真器连接到主机上。
<3> USB 接口电缆	连接主机和 QB-V850ESFX2 的电缆。
<4> AC 适配器	通过更换交流电源插座，可支持 100 到 240 V 的交流供电电压。
<8> QB-64GB-EA-01S QB-64GB-EA-02T 交换适配器	用于执行引脚转换的适配器。
<9> QB-64-CA-01S 引脚检测适配器	在方波监控中使用的适配器需要用到示波境，等。
<10> QB-64-SA-01S QB-64GB-YS-01T 空间适配器	用于调整最高点的适配器。
<11> QB-64GB-YQ-01T YQ 连接器	连接目标连接器和交换适配器的连接器。
<12> QB-64GB-MA-01S QB-64GB-HQ-01T 安装适配器	带有插槽的 V850ES/HE2 的适配器。
<13> QB-64GB-TC-01S QB-64GB-NQ-01T 目标连接器	目标系统焊接的连接器。

注 QB-V850ESFX2 提供电源单元，USB 接口线缆，和简易编程器 PG-FPL。同时也提供集成调试器 ID850QB 作为控制工具。

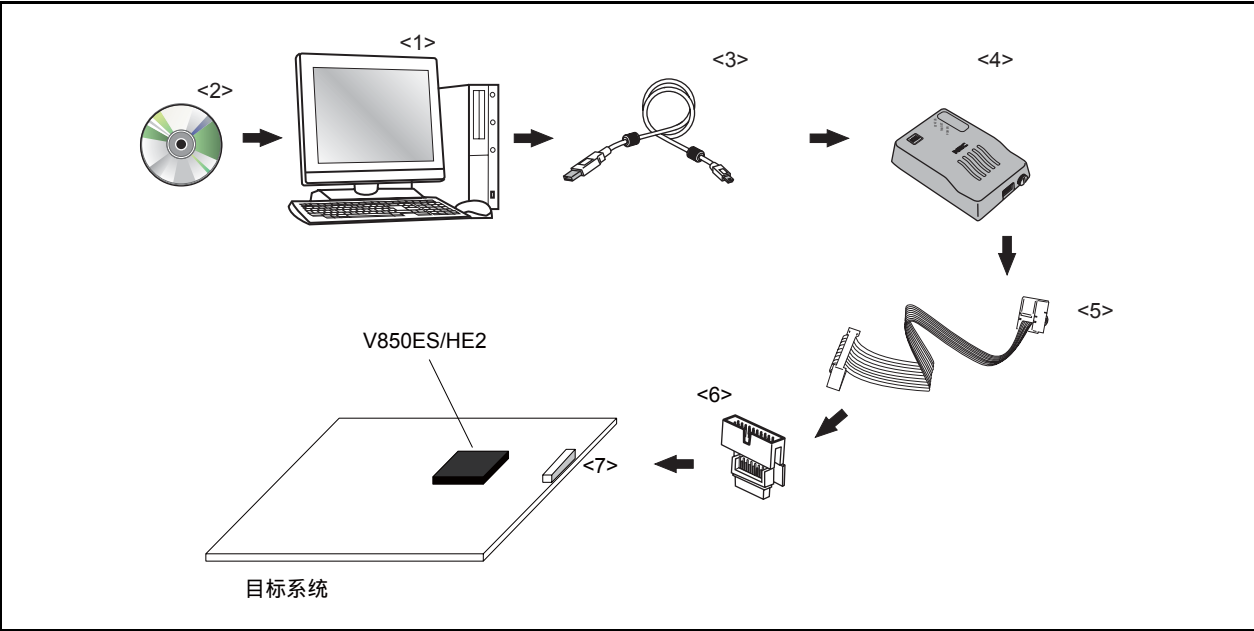
备注 方括号中的数字对应图 A-2 中的数字。

A.4.2 使用 MINICUBE QB-V850MINI 时

(1) 使用 MINICUBE 的片上仿真

以下显示了将 MINICUBE 连接到主机（PC-9821 系列，PC/AT 兼容机）时的系统结构。

图 A-3. 片上仿真系统结构



<1> 主机	带有 USB 端口的 PC。
<2> CD-ROM ^{#1}	CD-ROM 包含有诸如集成调试器 ID850QB，N 线检测器，设备驱动以及文件的内容。它与 MINICUBE 一起提供。
<3> USB 接口电缆	用于连接主机和 MINICUBE 的 USB 电缆。它与 MINICUBE 一起提供。电缆长度大概为 2m。
<4> MINICUBE 片上调试仿真器	使用 V850ES/HE2 开发应用系统时，片上调试仿真器用于调试硬件和软件。它支持集成调试器 ID850QB。
<5> OCD 电缆	用于连接 MINICUBE 和目标系统的电缆。它与 MINICUBE 一起提供。电缆长度大概为 20m。
<6> 连接器转换卡 KEL 适配器	该转换卡与 MINICUBE 一起提供。
<7> MINICUBE 连接器 KEL 连接器 ^{#2}	8830E-026-170S （与 MINICUBE 一起提供。） 8830E-026-170L （需另外购买）

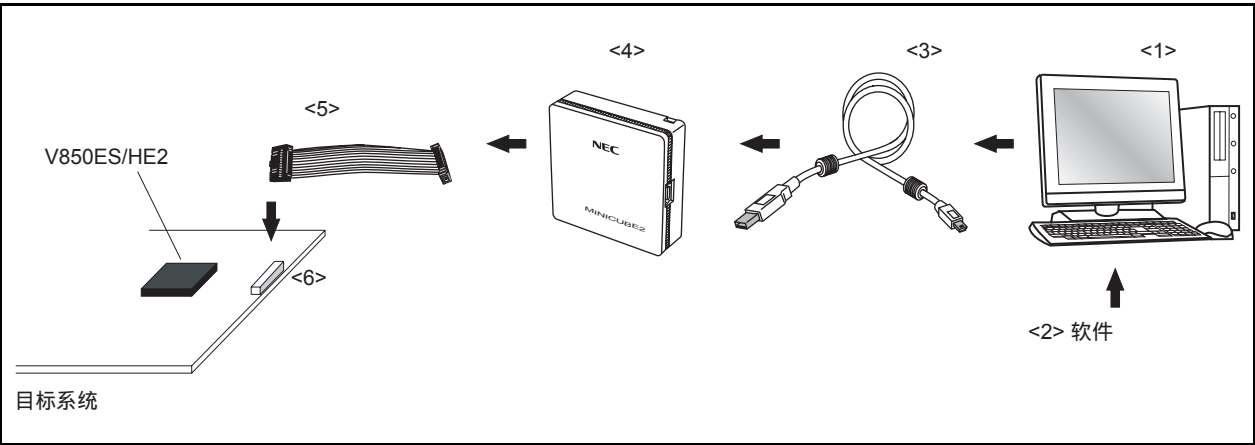
注 1. 从 NEC Electronics 网页下载设备文件。
<http://www.necel.com/micro/ods/eng/index.html>
2. KEL 公司的产品。

备注 尖括号中的数字对应于图 A-3 中的数字。

A.4.3 使用 MINICUBE2 QB-MINI2 时

以下显示了将 MINICUBE2 连接到主机（PC-9821 系列，PC/AT 兼容机）时的系统结构。

图 A-4. 片上仿真系统的系统结构



<1> 主机	带有 USB 端口的 PC。
<2> 软件	集成调试器 ID850QB，设备文件等。 NEC Electronics 网页下载设备文件。 http://www.necel.com/micro/ods/eng/
<3> USB 接口电缆	用于连接主机和 MINICUBE 的 USB 电缆。它与 MINICUBE 一起提供。电缆长度大概为 2m。
<4> MINICUBE2 片上调试仿真器	使用 V850ES/HE2 开发应用系统时，片上调试仿真器用于调试硬件和软件。它支持集成调试器 ID850QB。
<5> 16 引脚目标电缆	用于连接 MINICUBE2 和目标系统的电缆。 它与 MINICUBE 一起提供。电缆长度大概为 15m。
<6> 目标连接器（需另外购买）	使用带有 2.54 mm 间距的 16 引脚通信连接器。

备注 尖括号中的数字对应于图 A-4 中的数字。

A.5 调试工具（软件）

V850ES/Hx2 的 SM+（开发中） 系统模拟器	该系统模拟器同 V850 微控制器一同使用。 V850ES/Hx2 的 SM+ 是基于 Windows 的软件。 在主机上执行模拟目标系统操作时期间可以调试 C 源程序或汇编程序。 通过使用 V850ES/Hx2 的 SM+，可以独立于硬件开发完成逻辑验证和性能验证的应用。因此，改进了开发效率以及软件质量。它也可以同设备文件一起使用中。
	产品型号：μSxxxxSM703712-B
ID850QB 集成调试器	支持 V850 微控制器的在线仿真器。ID850 和 ID850QB 是基于 Windows 的软件。 它拥有改良的 C 语言编译调试功能，并且能够通过结合源程序的集成窗口功能，分解显示，和存储器显示来显示追踪源程序的结果。 它应该同设备文件一起使用中。 .
	产品型号：μSxxxx ID703000-QB（ID850QB）

备注 产品型号中的 xxxx 随主机和使用的 OS 而变化。

μSxxxxID703000-QB

xxxx	主机	OS	存储介质
AB17	PC-9800 系列，	Windows（日文版）	CD-ROM
BB17	IBM PC/AT 兼容机	Windows（英文版）	

A.6 嵌入软件

RX850, RX850 Pro 实时操作系统	RX850 和 RX850 Pro 是依照 μ ITRON 3.0 规范的实时操作系统。 提供一个为了产生多个信息表格的工具（配置器）。 RX850 Pro 比 RX850 具有更多的功能。
	产品型号： μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$ (RX850) μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$ (RX850 Pro)
Applilet [®]	这是一个为 V850ES/HE2 自动生成采样程序的驱动配置程序。
RX-FS850 (文件系统)	这是 FAT 文件系统功能 此文件系统支持 CD-ROM 文件系统功能。 此文件系统用在实时 OS RX850 Pro。

注 关于如何获取 Applilet，请咨询 NEC Electronics 的销售代表。

注意事项 要购买 RX850 或 RX850 Pro，首先需要填写购买申请表并且签署用户协议。

备注 产品型号中的 xxxx 和 $\Delta\Delta\Delta\Delta$ 随主机和使用的 OS 而变化。

μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$

μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$

$\Delta\Delta\Delta\Delta$	产品概述	用于大规模生产的上限
001	评价目标	不用于批量生产。
100K	大批量生产目标	0.1 万单元
001M		1 万单元
010M		10 万单元
S01	源程序	量产的目标源程序

xxxx	主机	OS	存储介质
AB17	PC-9800 系列， IBM PC/AT 兼容机	Windows（日文版）	CD-ROM
BB17		Windows（英文版）	
3K17	SPARCstation	Solaris（Rel. 2.5.1）	

A.7 Flash 存储器写入工具

Flashpro IV (产品型号: PG-FP4) Flash 存储器编程器	Flash 存储器编程器专用于有片上 Flash 存储器存储器的微控制器。
QB-MINI2 (MINICUBE2)	具有编程功能的片上调试仿真器。
FA-64GB-8EU-A Flash 存储器写入适配器	Flash 存储器写入适配器用于连接 Flashpro IV 等 (未连线)。
FA-70F3701GB-8EU-MX Flash 存储器写入适配器	Flash 存储器写入适配器用于连接 Flashpro IV 等 (已连线)。

备注 FA-64GB-8EU-A 和 FA-70F3701GB-8EU-MX 是 Naito Densei Machida Mfg. Co., Ltd 的产品。
TEL: +81-42-750-4172

附录 B 寄存器索引

(1/6)

符号	名称	单位	所在页
ADA0CR0	A/D 转换结果寄存器 0	ADC	393
ADA0CR0H	A/D 转换结果寄存器 0H	ADC	393
ADA0CR1	A/D 转换结果寄存器 1	ADC	393
ADA0CR1H	A/D 转换结果寄存器 1H	ADC	393
ADA0CR2	A/D 转换结果寄存器 2	ADC	393
ADA0CR2H	A/D 转换结果寄存器 2H	ADC	393
ADA0CR3	A/D 转换结果寄存器 3	ADC	393
ADA0CR3H	A/D 转换结果寄存器 3H	ADC	393
ADA0CR4	A/D 转换结果寄存器 4	ADC	393
ADA0CR4H	A/D 转换结果寄存器 4H	ADC	393
ADA0CR5	A/D 转换结果寄存器 5	ADC	393
ADA0CR5H	A/D 转换结果寄存器 5H	ADC	393
ADA0CR6	A/D 转换结果寄存器 6	ADC	393
ADA0CR6H	A/D 转换结果寄存器 6H	ADC	393
ADA0CR7	A/D 转换结果寄存器 7	ADC	393
ADA0CR7H	A/D 转换结果寄存器 7H	ADC	393
ADA0CR8	A/D 转换结果寄存器 8	ADC	393
ADA0CR8H	A/D 转换结果寄存器 8H	ADC	393
ADA0CR9	A/D 转换结果寄存器 9	ADC	393
ADA0CR9H	A/D 转换结果寄存器 9H	ADC	393
ADA0M0	A/D 转换模式寄存器 0	ADC	388
ADA0M1	A/D 转换模式寄存器 1	ADC	390
ADA0M2	A/D 转换模式寄存器 2	ADC	391
ADA0PFM	掉电比较模式寄存器	ADC	395
ADA0PFT	掉电比较阈值寄存器	ADC	395
ADA0S	A/D 转换器通道指定寄存器 0	ADC	392
ADIC	中断控制寄存器	INTC	512
CB0CTL0	CSIB0 控制寄存器 0	CSI	452
CB0CTL1	CSIB0 控制寄存器 1	CSI	455
CB0CTL2	CSIB0 控制寄存器 2	CSI	456
CB0RIC	中断控制寄存器	INTC	512
CB0RX	CSIB0 接收数据寄存器	CSI	451
CB0RXL	CSIB0 接收数据寄存器 L	CSI	451
CB0STR	CSIB0 状态寄存器	CSI	458
CB0TIC	中断控制寄存器	INTC	512
CB0TX	CSIB0 发送数据寄存器	CSI	451
CB0TXL	CSIB0 发送数据寄存器 L	CSI	451
CB1CTL0	CSIB1 控制寄存器 0	CSI	452
CB1CTL1	CSIB1 控制寄存器 1	CSI	455
CB1CTL2	CSIB1 控制寄存器 2	CSI	456

(2/6)

符号	名称	单位	所在页
CB1RIC	中断控制寄存器	INTC	512
CB1RX	CSIB1 接收数据寄存器	CSI	451
CB1RXL	CSIB1 接收数据寄存器 L	CSI	451
CB1STR	CSIB1 状态寄存器	CSI	458
CB1TIC	中断控制寄存器	INTC	512
CB1TX	CSIB1 发送数据寄存器	CSI	451
CB1TXL	CSIB1 发送数据寄存器 L	CSI	451
CCLS	CPU 操作时钟状态寄存器	CG	151
CLM	时钟监视模式寄存器	CLM	563
CTBP	CALLT 基指针	CPU	49
CTPC	CALLT 执行状态保存寄存器	CPU	48
CTPSW	CALLT 执行状态保存寄存器	CPU	48
DBPC	异常/调试陷阱状态保存寄存器	CPU	49
DBPSW	异常/调试陷阱状态保存寄存器	CPU	49
ECR	中断源寄存器	CPU	46
EIPC	中断状态保存源寄存器	CPU	45
EIPSW	中断状态保存源寄存器	CPU	45
FEPC	NMI 状态保存源寄存器	CPU	46
FEPSW	NMI 状态保存源寄存器	CPU	46
IMR0	中断屏蔽源寄存器 0	INTC	514
IMR0H	中断屏蔽源寄存器 0H	INTC	514
IMR0L	中断屏蔽源寄存器 0L	INTC	514
IMR1	中断屏蔽源寄存器 1	INTC	514
IMR1H	中断屏蔽源寄存器 1H	INTC	514
IMR1L	中断屏蔽源寄存器 1L	INTC	514
IMR2	中断屏蔽源寄存器 2	INTC	514
IMR2H	中断屏蔽源寄存器 2H	INTC	514
IMR2L	中断屏蔽源寄存器 2L	INTC	514
INTF0	外部中断下降沿指定寄存器 0	INTC	525
INTF3L	外部中断下降沿指定寄存器 3L	INTC	526
INTF9H	外部中断下降沿指定寄存器 9H	INTC	527
INTR0	外部中断上升沿指定寄存器 0	INTC	525
INTR3L	外部中断上升沿指定寄存器 3L	INTC	526
INTR9H	外部中断上升沿指定寄存器 9H	INTC	527
ISPR	正在服务优先级寄存器	INTC	515
KRIC	中断控制寄存器	INTC	512
KRM	按键返回模式寄存器	KR	533
LOCKR	锁定寄存器	CG	154
LVIIC	中断控制寄存器	INTC	512
LVIM	低电压检测寄存器	LVI	570
LVIS	低电压检测电平选择寄存器	LVI	571
NFC	噪声消除控制寄存器	INTC	528
OCDM	片上调试模式寄存器	DCU	607
OSTS	振荡稳定时间选择寄存器	WDT	538

符号	名称	单位	所在页
P0	端口 0	端口	79
P00NFC	TIP00 引脚噪声消除控制寄存器	定时器	174
P01NFC	TIP01 引脚噪声消除控制寄存器	定时器	174
P10NFC	TIP10 引脚噪声消除控制寄存器	定时器	174
P11NFC	TIP11 引脚噪声消除控制寄存器	定时器	174
P20NFC	TIP20 引脚噪声消除控制寄存器	定时器	174
P21NFC	TIP21 引脚噪声消除控制寄存器	定时器	174
P30NFC	TIP30 引脚噪声消除控制寄存器	定时器	174
P31NFC	TIP31 引脚噪声消除控制寄存器	定时器	174
P3L	端口 3L	端口	83
P4	端口 4	端口	88
P5	端口 5	端口	91
P7H	端口 7H	端口	97
P7L	端口 7L	端口	97
P9	端口 9	端口	100
P9H	端口 9H	端口	100
P9L	端口 9L	端口	100
PC	程序计数器	CPU	43
PCC	处理器时钟控制寄存器	CG	147
PCLM	可编程时钟模式寄存器	CG	156
PCM	端口 CM	端口	109
PDLL	端口 DLL	端口	112
PEMU1	外围仿真寄存器 1	LVI	576
PFC0	端口功能控制寄存器 0	端口	81
PFC3L	端口功能控制寄存器 3L	端口	85
PFC5	端口功能控制寄存器 5	端口	93
PFC9	端口功能控制寄存器 9	端口	104
PFC9H	端口功能控制寄存器 9H	端口	104
PFC9L	端口功能控制寄存器 9L	端口	104
PFCE3L	端口功能控制扩展寄存器 3L	端口	85
PFCE5	端口功能控制扩展寄存器 5	端口	93
PFCE9	端口功能控制扩展寄存器 9	端口	104
PFCE9H	端口功能控制扩展寄存器 9H	端口	104
PFCE9L	端口功能控制扩展寄存器 9L	端口	104
PIC0	中断控制寄存器	INTC	512
PIC1	中断控制寄存器	INTC	512
PIC2	中断控制寄存器	INTC	512
PIC3	中断控制寄存器	INTC	512
PIC4	中断控制寄存器	INTC	512
PIC5	中断控制寄存器	INTC	512
PIC6	中断控制寄存器	INTC	512
PIC7	中断控制寄存器	INTC	512
PLLCTL	PLL 控制寄存器	CG	153
PLLS	PLL 锁定时间指定寄存器	CG	155

符号	名称	单位	所在页
PM0	端口模式寄存器 0	端口	79
PM3L	端口模式寄存器 3L	端口	83
PM4	端口模式寄存器 4	端口	88
PM5	端口模式寄存器 5	端口	91
PM7H	端口模式寄存器 7H	端口	98
PM7L	端口模式寄存器 7L	端口	98
PM9	端口模式寄存器 9	端口	101
PM9H	端口模式寄存器 9H	端口	101
PM9L	端口模式寄存器 9L	端口	101
PMC0	端口模式控制寄存器 0	端口	80
PMC3L	端口模式控制寄存器 3L	端口	84
PMC4	端口模式控制寄存器 4	端口	89
PMC5	端口模式控制寄存器 5	端口	92
PMC9	端口模式控制寄存器 9	端口	102
PMC9H	端口模式控制寄存器 9H	端口	102
PMC9L	端口模式控制寄存器 9L	端口	102
PMCCM	端口模式控制寄存器 CM	端口	110
PMCM	端口模式寄存器 CM	端口	109
PMDLL	端口模式寄存器 DLL	端口	112
PRCMD	命令寄存器	CPU	69
PRSCM0	预分频比较寄存器 0	WT	372, 495
PRSM0	预分频模式寄存器 0	WT	371, 494
PSC	节电控制寄存器	CG	536
PSMR	节电模式寄存器	CG	537
PSW	程序状态字	CPU	47
PU0	上拉电阻选项寄存器 0	端口	81
PU3L	上拉电阻选项寄存器 3L	端口	86
PU4	上拉电阻选项寄存器 4	端口	89
PU5	上拉电阻选项寄存器 5	端口	95
PU9	上拉电阻选项寄存器 9	端口	107
PU9H	上拉电阻选项寄存器 9H	端口	107
PU9L	上拉电阻选项寄存器 9L	端口	107
Q00NFC	TIQ00 引脚噪声消除控制寄存器	定时器	274
Q01NFC	TIQ01 引脚噪声消除控制寄存器	定时器	274
Q02NFC	TIQ02 引脚噪声消除控制寄存器	定时器	274
Q03NFC	TIQ03 引脚噪声消除控制寄存器	定时器	274
r0 to r31	通用寄存器	CPU	43
RAMS	内部 RAM 数据状态寄存器	CG	571
RCM	内部振荡模式寄存器	CG	151
RESF	复位源标志寄存器	Reset	555
SELCNT0	选择器操作控制寄存器 0	定时器	251
SYS	系统状态寄存器	CPU	70
TM0CMP0	TMM0 比较寄存器 0	定时器	361
TM0CTL0	TMM0 控制寄存器 0	定时器	362

符号	名称	单位	所在页
TM0EQIC0	中断控制寄存器	INTC	512
TP0CCIC0	中断控制寄存器	INTC	512
TP0CCIC1	中断控制寄存器	INTC	512
TP0CCR0	TMP0 捕捉/比较寄存器 0	定时器	169
TP0CCR1	TMP0 捕捉/比较寄存器 1	定时器	171
TP0CNT	TMP0 计数器读取缓冲寄存器	定时器	173
TP0CTL0	TMP0 控制寄存器 0	定时器	162
TP0CTL1	TMP0 控制寄存器 1	定时器	163
TP0IOC0	TMP0 I/O 控制寄存器 0	定时器	165
TP0IOC1	TMP0 I/O 控制寄存器 1	定时器	166
TP0IOC2	TMP0 I/O 控制寄存器 2	定时器	167
TP0OPT0	TMP0 选项寄存器 0	定时器	168
TP0OVIC	中断控制寄存器	INTC	512
TP1CCIC0	中断控制寄存器	INTC	512
TP1CCIC1	中断控制寄存器	INTC	512
TP1CCR0	TMP1 捕捉/比较寄存器 0	定时器	169
TP1CCR1	TMP1 捕捉/比较寄存器 1	定时器	171
TP1CNT	TMP1 计数器读取缓冲寄存器	定时器	173
TP1CTL0	TMP1 控制寄存器 0	定时器	162
TP1CTL1	TMP1 控制寄存器 1	定时器	163
TP1IOC0	TMP1 I/O 控制寄存器 0	定时器	165
TP1IOC1	TMP1 I/O 控制寄存器 1	定时器	166
TP1IOC2	TMP1 I/O 控制寄存器 2	定时器	167
TP1OPT0	TMP1 选项寄存器 0	定时器	168
TP1OVIC	中断控制寄存器	INTC	512
TP2CCIC0	中断控制寄存器	INTC	512
TP2CCIC1	中断控制寄存器	INTC	512
TP2CCR0	TMP2 捕捉/比较寄存器 0	定时器	169
TP2CCR1	TMP2 捕捉/比较寄存器 1	定时器	171
TP2CNT	TMP2 计数器读取缓冲寄存器	定时器	173
TP2CTL0	TMP2 控制寄存器 0	定时器	162
TP2CTL1	TMP2 控制寄存器 1	定时器	163
TP2IOC0	TMP2 I/O 控制寄存器 0	定时器	165
TP2IOC1	TMP2 I/O 控制寄存器 1	定时器	166
TP2IOC2	TMP2 I/O 控制寄存器 2	定时器	167
TP2OPT0	TMP2 选项寄存器 0	定时器	168
TP2OVIC	中断控制寄存器	INTC	512
TP3CCIC0	中断控制寄存器	INTC	512
TP3CCIC1	中断控制寄存器	INTC	512
TP3CCR0	TMP3 捕捉/比较寄存器 0	定时器	169
TP3CCR1	TMP3 捕捉/比较寄存器 1	定时器	171
TP3CNT	TMP3 计数器读取缓冲寄存器	定时器	173
TP3CTL0	TMP3 控制寄存器 0	定时器	162
TP3CTL1	TMP3 控制寄存器 1	定时器	163

符号	名称	单位	所在页
TP3IOC0	TMP3 I/O 控制寄存器 0	定时器	165
TP3IOC1	TMP3 I/O 控制寄存器 1	定时器	166
TP3IOC2	TMP3 I/O 控制寄存器 2	定时器	167
TP3OPT0	TMP3 选项寄存器 0	定时器	168
TP3OVIC	中断控制寄存器	INTC	512
TQ0CCIC0	中断控制寄存器	INTC	512
TQ0CCIC1	中断控制寄存器	INTC	512
TQ0CCIC2	中断控制寄存器	INTC	512
TQ0CCIC3	中断控制寄存器	INTC	512
TQ0CCR0	TMQ0 捕捉/比较寄存器 0	定时器	265
TQ0CCR1	TMQ0 捕捉/比较寄存器 1	定时器	267
TQ0CCR2	TMQ0 捕捉/比较寄存器 2	定时器	269
TQ0CCR3	TMQ0 捕捉/比较寄存器 3	定时器	271
TQ0CNT	TMQ0 计数器读取缓冲寄存器	定时器	273
TQ0CTL0	TMQ0 控制寄存器 0	定时器	258
TQ0CTL1	TMQ0 控制寄存器 1	定时器	259
TQ0IOC0	TMQ0 I/O 控制寄存器 0	定时器	261
TQ0IOC1	TMQ0 I/O 控制寄存器 1	定时器	262
TQ0IOC2	TMQ0 I/O 控制寄存器 2	定时器	263
TQ0OPT0	TMQ0 选项寄存器 0	定时器	264
TQ0OVIC	中断控制寄存器	INTC	512
UA0CTL0	UARTA0 控制寄存器 0	UART	419
UA0CTL1	UARTA0 控制寄存器 1	UART	421
UA0CTL2	UARTA0 控制寄存器 2	UART	421
UA0OPT0	UARTA0 选项控制寄存器 0	UART	421
UA0RIC	中断控制寄存器	INTC	512
UA0RX	UARTA0 接收数据寄存器	UART	424
UA0STR	UARTA0 状态寄存器	UART	422
UA0TIC	中断控制寄存器	INTC	512
UA0TX	UARTA0 发送数据寄存器	UART	424
UA1CTL0	UARTA1 控制寄存器 0	UART	419
UA1CTL1	UARTA1 控制寄存器 1	UART	421
UA1CTL2	UARTA1 控制寄存器 2	UART	421
UA1OPT0	UARTA1 选项控制寄存器 0	UART	421
UA1RIC	中断控制寄存器	INTC	512
UA1RX	UARTA1 接收数据寄存器	UART	424
UA1STR	UARTA1 状态寄存器	UART	422
UA1TIC	中断控制寄存器	INTC	512
UA1TX	UARTA1 发送数据寄存器	UART	424
VSWC	系统等待控制寄存器	CPU	71
WDT	看门狗定时器允许寄存器	WDT	382
WDTM2	看门狗定时器 模式寄存器 2	WDT	380, 516
WTIC	中断控制寄存器	INTC	512
WTIIC	中断控制寄存器	INTC	512
WTM	钟表定时器 操作模式寄存器	WT	373

附录 C 指令集列表

C.1 常规指令

(1) 用于描述操作数的寄存器符号

寄存器符号	解释
reg1	通用寄存器：用作源寄存器。
reg2	通用寄存器：主要用作目的寄存器。在有些指令中也用作源寄存器。
reg3	通用寄存器：主要用作存放除法运算结果的余数或乘法运算结果的高 32 位。
bit#3	用于指定位编号的 3 位数据
immX	X 位立即数据
dispX	X 位偏移量数据
regID	系统寄存器编号
vector	用于指定陷阱向量的 5 位数据（00H 到 1FH）
cccc	表示条件代码的 4 位数据
sp	堆栈指针（r3）
ep	元素指针（r30）
listX	X 条目寄存器列表

(2) 用于描述操作码的寄存器符号

寄存器符号	解释
R	指定 reg1 或 regID 的 1 位数据码
r	指定 reg2 的 1 位数据码
w	指定 reg3 的 1 位数据码
d	1 位位移数据
l	1 位立即数据（指示立即数据的高位）
i	1 位立即数据
cccc	表示条件代码的 4 位数据
CCCC	表示 Bcond 指令的条件代码的 4 位数据
bbb	用于指定位编号的 3 位数据
L	在寄存器列表中指定程序寄存器的 1 位数据

(3) 用于描述操作数的寄存器符号

寄存器符号	解释
←	输入到
GR[]	通用寄存器
SR[]	系统寄存器
零扩展 (n)	用 0 来扩展 n 个位直到一个字边界
符号扩展 (n)	用符号来扩展 n 个位直到一个字边界。
装载存储器 (a, b)	从地址 a 读取位 b.
保存存储器 (a, b, c)	写长度为 c 的数据 b 到地址 a..
装载存储器-bit (a, b)	读取地址 a 的 b 位
保存存储器位 (a, b, c)	写入 c 到 地址 a 的 b 位.
饱和运算 (n)	执行 n 的饱和处理 (n 是 2 的余数) 作为计算的结果, 如果 n ≥ 7FFFFFFFH, 则 n 设为 7FFFFFFFH. n ≤ 80000000H, 则 n 设为 80000000H.
结果	在标记中反映结果。
Byte	字节 (8 位)
Halfword	半字 (16 位)
Word	字 (32 位)
+	加
−	减
	位连接
×	乘法
÷	除法
%	取余
AND	逻辑与
OR	逻辑或
XOR	异或
NOT	逻辑非
logically shift left by	逻辑左移
logically shift right by	逻辑右移
arithmetically shift right by	算术右移

(4) 用于操作时钟的寄存器符号

寄存器符号	解释
i	在指令执行后立即执行另一条指令 (issue)。
r	在指令执行后立即重复同一指令 (repeat)。
l	在指令执行后在指令中立即使用执行结果 (latency)。

(5) 用于标志操作的寄存器符号

标识符	解释
(Blank)	不变
0	清零
X	依据结果设置或清零。
R	恢复预先保存的值。

(6) 条件代码

条件代码 (cccc)	条件公式	解释
0 0 0 0	$OV = 1$	溢出
1 0 0 0	$OV = 0$	没有溢出
0 0 0 1	$CY = 1$	进位 小于 (小于)
1 0 0 1	$CY = 0$	没有进位 不低于 (大于或等于)
0 0 1 0	$Z = 1$	零
1 0 1 0	$Z = 0$	非零
0 0 1 1	$(CY \text{ or } Z) = 1$	不大于 (小于或等于)
1 0 1 1	$(CY \text{ or } Z) = 0$	大于 (大于)
0 1 0 0	$S = 1$	负
1 1 0 0	$S = 0$	正
0 1 0 1	—	一直 (无论什么情况下)
1 1 0 1	$SAT = 1$	饱和
0 1 1 0	$(S \text{ xor } OV) = 1$	小于带符号的数
1 1 1 0	$(S \text{ xor } OV) = 0$	大于或等于带符号的数
0 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 1$	小于或等于带符号的数
1 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 0$	大于带符号的数

C.2 指令集（按字母顺序）

(1/6)

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SA T
ADD	reg1, reg2	rrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]	1	1	1	×	×	×	×	
	imm5, reg2	rrrr010010iiii	GR[reg2]←GR[reg2]+符号扩展（imm5）	1	1	1	×	×	×	×	
ADDI	imm16, reg1, reg2	rrrr110000RRRRR iiiiiiiiiiii	GR[reg2]←GR[reg1]+符号扩展（imm16）	1	1	1	×	×	×	×	
AND	reg1, reg2	rrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]	1	1	1		0	×	×	
ANDI	imm16, reg1, reg2	rrrr110110RRRRR iiiiiiiiiiii	GR[reg2]←GR[reg1]AND 零扩展（imm16）	1	1	1		0	×	×	
Bcond	disp9	dddd1011ddcccc 注 1	如果条件满足 则 PC←PC+符号扩展（disp9）	2 注 2	2 注 2	2 注 2					
			当条件不满足时	1	1	1					
BSH	reg2, reg3	rrrr11111100000 www01101000010	GR[reg3]←GR[reg2]（23：16）∥GR[reg2] （31：24）∥ GR[reg2]（7：0）∥GR[reg2]（15：8）	1	1	1	×	0	×	×	
BSW	reg2, reg3	rrrr11111100000 www01101000000	GR[reg3]←GR[reg2]（7：0）∥GR[reg2]（15：8）∥GR [reg2]（23：16）∥GR[reg2]（31：24）	1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiii	CTPC←PC+2（返回PC） CTPSW←PSW adr←CTBP+零扩展（imm6 逻辑左移1位） PC←CTBP+零扩展（装载存储器（adr， Halfword））	4	4	4					
CLR1	bit#3, disp16[reg1]	10bbb111110RRRRR dddddddddddddd	adr←GR[reg1]+符号扩展（disp16） Z 标志←Not（装载存储器位（adr，bit#3）） 存储存储器位（adr，bit#3，0）	3 注 3	3 注 3	3 注 3				×	
	reg2, [reg1]	rrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z 标志←Not（装载存储器位（adr，reg2）） 存储存储器位（adr，reg2，0）	3 注 3	3 注 3	3 注 3				×	
CMOV	cccc, imm5, reg2, reg3	rrrr111111iiii www011000cccc0	如果条件满足 则 GR[reg3]←符号扩展 ed（imm5） 否则 GR[reg3]←GR[reg2]	1	1	1					
	cccc, reg1, reg2, reg3	rrrr111111RRRR www011001cccc0	如果条件满足 则 GR[reg3]←GR[reg1] 否则 GR[reg3]←GR[reg2]	1	1	1					
CMP	reg1, reg2	rrrr001111RRRRR	结果←GR[reg2]−GR[reg1]	1	1	1	×	×	×	×	
	imm5, reg2	rrrr010011iiii	结果←GR[reg2]−符号扩展（imm5）	1	1	1	×	×	×	×	
CTRET		000001111100000 0000000101000100	PC←CTPC PSW←CTPSW	3	3	3	R	R	R	R	R
DBRET		000001111100000 0000000101000110	PC←DBPC PSW←DBPSW	3	3	3	R	R	R	R	R

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SA T
DBTRAP		1111100001000000	DBPC←PC+2 (被恢复的 PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3					
DI		000001111110000000 00000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5, list12	0000011001iiiiLLLLLL LLLLLL00000	sp←sp+零扩展 (imm5 逻辑左移 2) GR[列表 12 中的 reg]←装载存储器 (sp, Word) sp←sp+4 重复以上 2 个步骤直到在列表 12 中的所有 reg 被装载	n+1 注 4	n+1 注 4	n+1 注 4					
	imm5, list12, [reg1]	0000011001iiiiLLLLLL LLLLLLRRRRR 注 5	sp←sp+零扩展 (imm5 逻辑左移 2) GR[列表 12 中的 reg]←装载存储器 (sp, Word) sp←sp+4 重复以上 2 个步骤直到在列表 12 中的所有 reg 被装载 PC←GR[reg1]	n+3 注 4	n+3 注 4	n+3 注 4					
DIV	reg1, reg2, reg3	rrrrr111111RRRRRww www01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVH	reg1, reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1] ^{注 6}	35	35	35		×	×	×	
	reg1, reg2, reg3	rrrrr111111RRRRRww www01010000000	GR[reg2]←GR[reg2]÷GR[reg1] ^{注 6} GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1, reg2, reg3	rrrrr111111RRRRRww www01010000010	GR[reg2]←GR[reg2]÷GR[reg1] ^{注 6} GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1, reg2, reg3	rrrrr111111RRRRRww www01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		100001111110000000 00000101100000	PSW.ID←0	1	1	1					
HALT		000001111110000000 00000100100000	停止	1	1	1					
HSW	reg2, reg3	rrrrr11111100000www ww01101000100	GR[reg3]←GR[reg2] (15 : 0) GR[reg2] (31 : 16)	1	1	1	×	0	×	×	
JARL	disp22, reg2	rrrrr11110ddddddddd ddddddddddd0 注 7	GR[reg2]←PC+4 PC←PC+符号扩展 (disp22)	2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110ddddddd ddddddddddd0 注 7	PC←PC+符号扩展 (disp22)	2	2	2					
LD.B	disp16[reg1], reg2	rrrrr111000RRRRR ddddddddddddddd	adr←GR[reg1]+符号扩展 (disp16) GR[reg2]←符号扩展 (装载存储器 (adr, Byte))	1	1	注 11					
LD.BU	disp16[reg1], reg2	rrrrr11110bRRRRRdd ddddddddddd1 注 8, 10	adr←GR[reg1]+符号扩展 (disp16) GR[reg2]←零扩展 (装载存储器 (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	SA T	
LD.H	disp16[reg1], reg2	rrrrr111001RRRRRdd ddddddddddddd0 注 8	adr←GR[reg1]+符号扩展（disp16） GR[reg2]←符号扩展（装载存储器（adr， Halfword））		1	1	注 11					
LDSR	reg2, regID	rrrrr111111RRRRR00 00000000100000 注 12	SR[regID]←GR[reg2]	除 regID = PSW 外的情况	1	1	1					
			regID = PSW		1	1	1	×	×	×	×	×
LD.HU	disp16[reg1], reg2	rrrrr111111RRRRRdd ddddddddddddd1 注 8	adr←GR[reg1]+符号扩展（disp16） GR[reg2]←零扩展（装载存储器（adr，Halfword））		1	1	注 11					
LD.W	disp16[reg1], reg2	rrrrr111001RRRRRdd ddddddddddddd1 注 8	adr←GR[reg1]+符号扩展（disp16） GR[reg2]←装载存储器（adr，Word）		1	1	注 11					
MOV	reg1, reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]		1	1	1					
	imm5, reg2	rrrrr010000iiii	GR[reg2]←符号扩展（imm5）		1	1	1					
	imm32, reg1	00000110001RRRRR iiiiiiiiiiii IIIIIIIIIIII	GR[reg1]←imm32		2	2	2					
MOVEA	imm16, reg1, reg2	rrrrr110001RRRRRiiiiii iiiiiiii	GR[reg2]←GR[reg1]+符号扩展（imm16）		1	1	1					
MOVHI	imm16, reg1, reg2	rrrrr110010RRRRRiiiiii iiiiiiii	GR[reg2]←GR[reg1]+（imm16 Ⅱ 0 ¹⁶ ）		1	1	1					
MUL	reg1, reg2, reg3	rrrrr111111RRRRRww www01000100000	GR[reg3] Ⅱ GR[reg2]←GR[reg2]xGR[reg1] 注 14		1	4	5					
	imm9, reg2, reg3	rrrrr111111iiiiwwwww 01001IIII00 注 13	GR[reg3] Ⅱ GR[reg2]←GR[reg2]x 符号扩展（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} x 符号扩展（imm5）		1	1	2					
MULHI	imm16, reg1, reg2	rrrrr110111RRRRRiiiiii iiiiiiii	GR[reg2]←GR[reg1] ^{※6} ximm16		1	1	2					
MULU	reg1, reg2, reg3	rrrrr111111RRRRRww www01000100010	GR[reg3] Ⅱ GR[reg2]←GR[reg2]xGR[reg1] 注 14		1	4	5					
	imm9, reg2, reg3	rrrrr111111iiiiwwwww 01001IIII10 注 13	GR[reg3] Ⅱ GR[reg2]←GR[reg2]x 零扩展（imm9）		1	4	5					
NOP		0000000000000000	要经过至少一个时钟周期的空操作。		1	1	1					
NOT	reg1, reg2	rrrrr000001RRRRR	GR[reg2]←NOT（GR[reg1]）		1	1	1		0	×	×	
NOT1	bit#3, disp16[reg1]	01bbb111110RRRRR ddddddddddddd	adr←GR[reg1]+符号扩展（disp16） Z 标志←Not（装载存储器位（adr，bit#3）） 保存存储器位（adr，bit#3，Z 标志）		3 注 3	3 注 3	3 注 3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z 标志←Not（装载存储器位（adr，reg2）） 保存存储器位（adr，reg2，Z 标志）		3 注 3	3 注 3	3 注 3				×	

助记符	操作数	操作码	操作	执行时钟		标志					
				i	r	l	CY	OV	S	Z	SA T
OR	reg1, reg2	rrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16, reg1, reg2	rrrr110100RRRRRiiiiii iiiiiii	GR[reg2]←GR[reg1]OR 零扩展 (imm16)	1	1	1		0	×	×	
PREPARE	list12, imm5	0000011110iiiiLLLLLL LLLLLL00001	保存存储器 (sp-4, GR[列表 12 中的 reg], Word) sp←sp-4 重复以上的 1 个步骤直到列表 12 中的所有 reg 被保存 sp←sp-零扩展 (imm5)	n+1 注 4	n+1 注 4	n+1 注 4					
	list12, imm5, sp/imm ^{注 15}	0000011110iiiiLLLLLL LLLLLLf011imm16/im m32 注 16	保存存储器 (sp-4, GR[列表 12 中的 reg], Word) sp←sp+4 重复以上的 1 个步骤直到列表 12 中的所有 reg 被保存 sp←sp-零扩展 (imm5) ep←sp/imm	n+2 注 4 注 17	n+2 注 4 注 17	n+2 注 4 注 17					
RETI		00000111110000000 00000101000000	如果 PSW.EP=1 则 PC ←EIPC PSW ←EIPSW 否则 if 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	rrrr111111RRRRR00 00000010100000	GR[reg2]←GR[reg2]算数右移 by GR[reg1]	1	1	1	×	0	×	×	
	imm5, reg2	rrrr010101iiii	GR[reg2]←GR[reg2]算数右移 by 零扩展 (imm5)	1	1	1	×	0	×	×	
SASF	cccc, reg2	rrrr1111110cccc0000 001000000000	如果条件满足 则 GR[reg2]←(GR[reg2]逻辑左移 1) OR 00000001H 否则 GR[reg2]←(GR[reg2]逻辑左移 1) OR 00000000H	1	1	1					
SATADD	reg1, reg2	rrrr000110RRRRR	GR[reg2]←饱和运算 (GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5, reg2	rrrr010001iiii	GR[reg2]←饱和运算 (GR[reg2]+符号扩展 (imm5))	1	1	1	×	×	×	×	×
SATSUB	reg1, reg2	rrrr000101RRRRR	GR[reg2]←饱和运算 (GR[reg2]-GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16, reg1, reg2	rrrr110011RRRRRiiiiii iiiiiii	GR[reg2]←饱和运算 (GR[reg1]-符号扩展 (imm16))	1	1	1	×	×	×	×	×
SATSUBR	reg1, reg2	rrrr000100RRRRR	GR[reg2]←饱和运算 (GR[reg1]-GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc, reg2	rrrr1111110cccc0000 000000000000	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	SA T
SET1	bit#3, disp16[reg1]	00bbb111110RRRRR dddddddddddddd	adr←GR[reg1]+符号扩展（disp16） Z 标志←Not（装载存储器位（adr, bit#3）） 保存存储器位（adr, bit#3, 1）	3 注 3	3 注 3	3 注 3				×	
	reg2, [reg1]	rrrr111111RRRRR00 00000011100000	adr←GR[reg1] Z 标志←Not（装载存储器位（adr, reg2）） 保存存储器位（adr, reg2, 1）	3 注 3	3 注 3	3 注 3				×	
SHL	reg1, reg2	rrrr111111RRRRR00 00000011000000	GR[reg2]←GR[reg2]逻辑左移 GR[reg1]	1	1	1	×	0	×	×	
	imm5, reg2	rrrrr010110iiii	GR[reg2]←GR[reg2] 逻辑左移零扩展（imm5）	1	1	1	×	0	×	×	
SHR	reg1, reg2	rrrr111111RRRRR00 00000010000000	GR[reg2]←GR[reg2] 逻辑右移 GR[reg1]	1	1	1	×	0	×	×	
	imm5, reg2	rrrrr010100iiii	GR[reg2]←GR[reg2] l 逻辑右移零扩展（imm5）	1	1	1	×	0	×	×	
SLD.B	disp7[ep], reg2	rrrrr0110dddddd	adr←ep+零扩展（disp7） GR[reg2]←符号扩展（装载存储器（adr, Byte））	1	1	注 9					
SLD.BU	disp4[ep], reg2	rrrrr0000110ddd 注 18	adr←ep+零扩展（disp4） GR[reg2]←零扩展（装载存储器（adr, Byte））	1	1	注 9					
SLD.H	disp8[ep], reg2	rrrrr1000dddddd 注 19	adr←ep+零扩展（disp8） GR[reg2]←符号扩展（装载存储器（adr, Halfword））	1	1	注 9					
SLD.HU	disp5[ep], reg2	rrrrr0000111ddd 注 18, 20	adr←ep+零扩展（disp5） GR[reg2]←零扩展（装载存储器（adr, Halfword））	1	1	注 9					
SLD.W	disp8[ep], reg2	rrrrr1010dddddd0 注 21	adr←ep+零扩展（disp8） GR[reg2]←装载存储器（adr, Word）	1	1	注 9					
SST.B	reg2, disp7[ep]	rrrrr0111dddddd	adr←ep+零扩展（disp7） 保存存储器（adr, GR[reg2], Byte）	1	1	1					
SST.H	reg2, disp8[ep]	rrrrr1001dddddd 注 19	adr←ep+零扩展（disp8） 保存存储器（adr, GR[reg2], Halfword）	1	1	1					
SST.W	reg2, disp8[ep]	rrrrr1010dddddd1 注 21	adr←ep+零扩展（disp8） 保存存储器（adr, GR[reg2], Word）	1	1	1					
ST.B	reg2, disp16[reg1]	rrrrr111010RRRRRdd dddddddddddddd	adr←GR[reg1]+符号扩展（disp16） 保存存储器（adr, GR[reg2], Byte）	1	1	1					
ST.H	reg2, disp16[reg1]	rrrrr111011RRRRRdd dddddddddddddd0 注 8	adr←GR[reg1]+符号扩展（disp16） 保存存储器（adr, GR[reg2], Halfword）	1	1	1					
ST.W	reg2, disp16[reg1]	rrrrr111011RRRRRdd dddddddddddddd1 注 8	adr←GR[reg1]+符号扩展（disp16） 保存存储器（adr, GR[reg2], Word）	1	1	1					
STSR	regID, reg2	rrrrr111111RRRRR00 00000001000000	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	SA T
SUB	reg1, reg2	rrrr001101RRRRR	GR[reg2]←GR[reg2]-GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1, reg2	rrrr001100RRRRR	GR[reg2]←GR[reg1]-GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	00000000010RRRRR	adr←(PC+2)+(GR[reg1]逻辑左移1) PC←(PC+2)+(符号扩展 (装载存储器 (adr, Halfword)) 逻辑左移1)	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←符号扩展 (GR[reg1] (7:0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←符号扩展 (GR[reg1] (15:0))	1	1	1					
TRAP	vector	0000011111iiii00000 001000000000	EIPC ←PC+4 (被恢复的 PC) EIPSW ←PSW ECR.EICC ←中断代码 PSW.EP ←1 PSW.ID ←1 PC ←00000040H (当向量是 00H 到 0FH) 00000050H (当向量是 10H 到 1FH)	3	3	3					
TST	reg1, reg2	rrrr001011RRRRR	结果←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3, disp16[reg1]	11bbb111110RRRRR dddddddddddddd	adr←GR[reg1]+符号扩展 (disp16) Z 标志←Not (装载存储器位 (adr, bit#3))	3 注 3	3 注 3	3 注 3				×	
	reg2, [reg1]	rrrr111111RRRRR00 00000011100110	adr←GR[reg1] Z 标志←Not (装载存储器位 (adr, reg2))	3 注 3	3 注 3	3 注 3				×	
XOR	reg1, reg2	rrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16, reg1, reg2	rrrr110101RRRRRiiiiii iiiiiiii	GR[reg2]←GR[reg1] XOR 零扩展 (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←零扩展 (GR[reg1] (7:0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←零扩展 (GR[reg1] (15:0))	1	1	1					

- 注
1. dddddddd: disp9 的高 8 位。
 2. 如果有重写即刻的 PSW 内容的指令，那么时钟数为 3。
 3. 如果没有等待状态 (3+读取的等待状态数)。
 4. n 是 list12 中装载的寄存器的总量 (依照等待状态而得。如果没有等待状态，n 是列表 12 中寄存器的总数。如果 n=0，会执行和 n=1 时同样的操作。))。
 5. RRRRR: 00000 除外。
 6. 仅低半字数据有效。
 7. ddddddddddddddddddd: disp22 的高 21 位。
 8. ddddddddddddddd: disp16 的高 15 位。
 9. 根据等待状态的数目 (如果没有等待状态则为 1)。
 10. b: disp16 的第 0 位。
 11. 根据等待状态的数目 (如果没有等待状态则为 2)。

- 注 12. 在该指令中，为了记忆描述的方便，将由 **reg2** 构成源寄存器，而 **reg1** 区则在操作码中使用。因此，在记忆描述和操作码中寄存器规定的含义与其它指令不同。
- rrrrr = regID 指定
RRRRR = reg2 指定
13. iiii: imm9 的低 5 位。
IIII: imm9 的高 4 位。
14. 对于通用寄存器 **reg1** 和 **reg3**，不要指定相同的寄存器。
15. sp/imm: 由副操作码的位 19 和位 20 指定。
16. ff = 00: 在 ep 中装载 SP
01: 在 ep 中装载扩展符号的 16 位立即数（位 47 至 32）。
10: 在 ep 中装载 16 位的逻辑左移 16 位立即数（位 47 至 32）。
11: 在 ep 中加载 32 位立即数据（位 63 至 32）。
17. 如果 imm = imm32, n + 3 个时钟。
18. rrrrr: 00000 除外。
19. ddddddd: disp8 的高 7 位。
20. dddd: disp5 的高 4 位。
21. ddddd: disp8 的高 6 位。

附录 D 注意事项列表

该附录列出了在本文档中所描述的注意事项。

表格中的“分类（硬件/软件）”如下所示。

硬件：微处理器内部/外部硬件的注意事项。

软件：诸如寄存器设置或程序的软件的事项。

(1/27)

章节	分类	功能	功能的详细信息	注意事项	所在页
第1章	硬件	介绍	FLMD0	正常模式下，将该引脚与Vss 相连接。	p. 21 <input type="checkbox"/>
			REGC	将REGC 引脚 通过一个4.7（ μ F建议值）的电容连接到Vss。	p. 21 <input type="checkbox"/>
第2章	软件	引脚功能	NMI	引脚NMI的复用功能与引脚P02相同。复位后，引脚功能为引脚P02。要能使引脚NMI，设置PMC0.PMC02位为1。引脚NMI的初始设置为“无边沿检测”。通过寄存器INTF0和INTR0选择引脚NMI的有效边沿。	p. 28 <input type="checkbox"/>
	硬件		FLMD0	如果超过噪声消除宽度的噪声在自编程期间被输入到RESET引脚中，那么当电容被连接到FLMD0引脚时，根据电容充电结束时间可能会进入Flash存储器登入模式。因此，请不要将电容连接到FLMD0引脚上。	p. 38 <input type="checkbox"/>
	硬件		电源开启时	需注意，当打开电源时，下述引脚可能在复位期间临时输出一个不确定的电平。P53/KR3/TIQ00/TOQ00/DDO引脚	p. 40 <input type="checkbox"/>
第3章	软件	CPU 功能	EIPC寄存器，EIPSW 寄存器，FEPC寄存器，FEPSW 寄存器	由于中断状态保存寄存器只有一组，因此，要实现多重中断需用程序对这组寄存器进行保存。	p. 44 <input type="checkbox"/>
			EIPC， FEPC	即使EIPC， FEPC或CTPC的第0位被LDSR指令置1，在中断处理程序执行后程序由RETI指令返回时，第0位仍然被忽略（这是因为PC的第0位固定为0）。因此请使用偶数数值（第0位为0）来设置EIPC， FEPC或CTPC。	p. 44 <input type="checkbox"/>
			程序空间	由于从地址03FFF000H到03FFFFFFFH的4KB范围是片上外围I/O设备区，这个区域不能进行取指令操作。因此，要避免执行可能使分支地址计算结果为此4KB中值的操作。	p. 52 <input type="checkbox"/>
			片上外围I/O 区域	当寄存器被一个字操作指令读写时，一个字的空间被分为低16位和高16位分两次被字节操作先后读写，其中最低2位地址被忽略。	p. 57 <input type="checkbox"/>
				若对只能进行字节操作的寄存器进行半字长（halfword）操作，那么读取时，高8位数据无效，写入时，低8位数据被写入寄存器。	p. 57 <input type="checkbox"/>
				未被指定为寄存器的地址为未来扩展保留。对这些地址操作结果的有效性是无法保证的。	p. 57 <input type="checkbox"/>
			内部RAM区域	如果分支指令在内部RAM区间的上沿地址处发生，那么将不会产生跨越片上外围I/O区间的预取址（无效取址）。	p. 58 <input type="checkbox"/>
将数据设置到特殊寄存器中	当转换至IDLE1， IDLE2， STOP或副IDLE模式时（PSC.STP位=1），5条NOP指令必须在转换执行后立即被插入。	p. 68 <input type="checkbox"/>			

(2/27)

章节	分类	功能	功能的详细信息	注意事项	所在页
第3章	软件	CPU	将数据设置到特殊寄存器中	当对命令寄存器进行写操作时，系统将不响应中断。存储指令应该是按上述的第<2>和<3>步骤连续执行的。若在步骤<2>和<3>之间执行了其他指令，而这个指令又进行了中断的响应，则将打乱上述的顺序导致设备误操作。	p. 68 <input type="checkbox"/>
				尽管对PRCMD写入的数据为虚数据，但还是请使用与设置特殊寄存器（例子中的<3>）的通用寄存器相同的寄存器来向PRCMD寄存器（例子中的<2>）写入虚数据。这个过程也同样可以用于使用通用寄存器进行寻址的操作中。	p. 68 <input type="checkbox"/>
			SYS寄存器	如果在对PRCMD寄存器进行写入之后，向SYS寄存器（并非特殊寄存器）的PRERR位写入0，那么PRERR位将被清零。（写入命令优先）。	p. 70 <input type="checkbox"/>
				如果在对PRCMD寄存器进行写入之后，又执行了对PRCMD寄存器（并非特殊寄存器）的写入操作，那么PRERR位将被置1。	p. 70 <input type="checkbox"/>
			要先进行设置的寄存器	在使用V850ES/HE2器件时，请务必首先对以下寄存器进行设置。 • 系统等待控制寄存器（VSWC） • 片上调试模式寄存器（OCDM） • 看门狗定时器模式寄存器2（WDTM2）	p. 71 <input type="checkbox"/>
			VSWC寄存器	访问一个片上周边I/O寄存器需要3个时钟周期的时间（不包含等待周期）。V850ES/HE2需要根据不同的操作频率插入不同的等待周期数。因此，请根据所使用的时钟，将以下对应的数值设置到VSWC寄存器中。	p. 71 <input type="checkbox"/>
第4章	硬件	端口功能	访问特殊片上外围I/O寄存器	以下状态中，禁止访问上述的寄存器。如果这时产生了等待状态，那么只有复位才可退出等待状态。 • 当主时钟振荡停止，系统工作于副时钟时 • 当CPU工作于内部振荡器时钟时	p. 72 <input type="checkbox"/>
			端口功能	虽然1-位内存操作指令是操作1位的，但它会以8-位为单位来访问端口。因此，如果一个端口包含了输入引脚及输出引脚的混合物，那么即使引脚不受操作的控制，在输入模式中设置的引脚的输出锁的内容也将会变为未定义的状态。	p. 76 <input type="checkbox"/>
			端口 0	NMI引脚的复用功能与P02引脚相同。复位后其功能相当于P02引脚。为了允许NMI引脚，应将PMC0.PMC02位设为1。NMI引脚的初始设置是“无边缘检测”。通过使用INTF0和INTR0寄存器来选择NMI引脚的有效沿。 P05引脚的复用功能是片上调试功能。外部复位后，P05/INTP2/DRST引脚将会被初始化为片上调试引脚（DRST）。为了将P05引脚用作端口引脚而不是片上调试引脚，必须执行以下操作。 <1> 将OCDM.OCDM0位（特殊寄存器）清0。 <2> 将P05/INTP2/DRST引脚固定为低电平直到上述操作被完成。 当片上调试功能没有被使用时，在上述操作被执行前将一个高电平输入DRST引脚可能会引起故障（CPU锁死）。因此，处理P05引脚时应特别小心。 当高电平没有被输入到P05/INTP2/DRST引脚时（该引脚固定为低电平时），将没有必要操作OCDM.OCDM0位。 由于下拉电阻（30 kΩ TYP.）被连接到P05/INTP2/DRST引脚的缓冲器上，引脚将不必通过一个外部源来固定为低电平。下拉电阻通过将OCDM0位清0来断开连接。	p. 78 <input type="checkbox"/> p. 78 <input type="checkbox"/>

(3/27)

章节	分类	功能	功能的详细信息	注意事项	所在页	
第4章	硬件	端口功能	端口 0	P00 至 P06 引脚在复用功能的输入模式中具有磁滞特性，但在端口模式中则没有磁滞特性。	p. 79	<input type="checkbox"/>
	软件		PMC0寄存器	当OCDM.OCDM0 位为1时无论PMC05位的值如何， P05/INTP2/DRST 引脚都会变为DRST 引脚。	p. 80	<input type="checkbox"/>
	硬件		端口 3	P31 至 P35引脚在复用功能引脚的输入模式中具有磁滞特性，但在端口模式中没有磁滞特性。	p. 82	<input type="checkbox"/>
	软件		PMC3L寄存器	INTP7引脚的功能与RXDA0引脚的功能是交互的。要用作RXDA0引脚，则使复用功能引脚INTP7的边缘检测功能失效（通过将INTF3L.INTF31位以及INTR3L.INTR31位固定为0）。要用作INTP7引脚，则停止UARTA0的接收操作（通过将UA0CTL0.UA0RXE位清0）。	p. 84	<input type="checkbox"/>
	硬件		端口 4	P40 至 P42 引脚在复用功能引脚的输入模式中具有磁滞特性，但在端口模式中没有磁滞特性。	p. 87	<input type="checkbox"/>
	硬件， 软件		端口 5	DDI, DDO, DCK以及DMS引脚用于片上调试功能。要将DDI, DDO, DCK以及DMS引脚用作端口引脚而不是片上调试引脚，则必须执行以下操作。 <1> 将OCDM寄存器（特殊寄存器）的OCDM0位清0。 <2> 将P05/INTP2/DRST引脚固定为低电平直到上述操作被执行。 当片上调试功能没有被使用时，在上述操作被执行前将一个高电平输入DRST引脚可能会引起故障（CPU锁死）。因此，处理P05引脚时应特别小心。 当高电平没有被输入到P05/INTP2/DRST引脚时（该引脚固定为低电平时），将没有必要操作OCDM.OCDM0位。 由于下拉电阻（30 kΩ TYP.）被连接到P05/INTP2/DRST引脚的缓冲器上，引脚将不必通过一个外部来源固定为低电平。下拉电阻通过将OCDM0位清0来断开连接。	p. 90	<input type="checkbox"/>
				P50 至 P55 引脚在复用功能的输入模式中具有磁滞特性，但在端口模式中没有磁滞特性。	p. 90	<input type="checkbox"/>
	软件		PMC5寄存器	如果在PFC5.PFC5n以及PFCE5.PFCE5n位的值等于默认值（0）时通过使用PMC5寄存器指定了控制模式，那么输出将会变成未定义的状态。 由于这个原因，应先设置PFC5.PFC5n和PFCE5.PFCE5n位，然后通过将PMC5n位设为1来设置控制模式。	pp. 92, 93	<input type="checkbox"/>
			端口 5复用功能指定	KRn引脚的功能与TIQ0m引脚的功能是交互的。要将该引脚用作TIQ0m引脚，则应使复用功能引脚KRn的按键中断检测功能失效（通过将KRM.KRMn位清0）。要将该引脚用作KRn引脚，则应使复用功能引脚TIQ0m的边缘检测功能失效（n = 0至3, m = 0至3）。	p. 94	<input type="checkbox"/>
			P7H寄存器，P7L寄存器	在A/D 转换期间不要读取P7H 和 P7L 寄存器。	p. 97	<input type="checkbox"/>
			PM7H寄存器，PM7L寄存器	要使用P7n（ANIn）的复用功能，则应将PM7n设置为1。	p. 98	<input type="checkbox"/>
			端口 9	P90, P91, P96, P97, P99, 和 P913 至 P915 引脚在复用功能引脚中的输入模式具有磁滞特性，但在端口模式中没有磁滞特性。	p. 99	<input type="checkbox"/>
	软件		P9寄存器	要以8位 或 1位为单位读/写P9寄存器的位 8 至 15，需要将这些位指定为P9H寄存器的位 0 至 7。	p. 100	<input type="checkbox"/>
			PM9寄存器	要以8位 或 1位为单位读/写PM9寄存器的位 8 至 15，需要将这些位指定为PM9H寄存器的位 0 至 7。	p. 101	<input type="checkbox"/>

章节	分类	功能	功能的详细信息	注意事项	所在页
第4章	软件	端口功能	PMC9寄存器	如果在PFC9.PFC9n以及PFCE9.PFCE9n位的值等于默认值（0）时通过使用PMC9寄存器指定了控制模式，那么输出将会变成未定义的状态。由于这个原因，应先将PFC9.PFC9n和PFCE9.PFCE9n位设置为1，然后通过将PMC9n位设为1来设置控制模式。	p. 102 <input type="checkbox"/>
				要以8位 或 1位为单位读/写PMC9寄存器的位8至15，需要将这些位指定为PMC9H寄存器的位0至7。	p. 102 <input type="checkbox"/>
			PFC9寄存器	要以8位或1位为单位读/写PFC9寄存器的位8至15，需要将这些位指定为PFC9H寄存器的位0至7。	p. 104 <input type="checkbox"/>
			PFCE9寄存器	要以8位或1位为单位读/写PFCE9寄存器的位8至15，需要将这些位指定为PFCE9H寄存器的位0至7。	p. 104 <input type="checkbox"/>
			端口9复用功能的指定	如果在PFC9.PFC9n以及PFCE9.PFCE9n位的值等于默认值（0）时通过使用PMC9寄存器指定了控制模式，那么输出将会变成未定义的状态。由于这个原因，应先将PFC9.PFC9n和PFCE9.PFCE9n位，然后通过将PMC9n位设为1来设置控制模式。	p. 105 <input type="checkbox"/>
				KR7引脚和RXDA1引脚是复用功能引脚。当将引脚用作RXDA1引脚时，禁止KR7引脚的按键中断检测。（将KRM寄存器的KRM7位清0。）同样地，当将引脚用作KR7引脚时，建议将PFC91位设置为1并将PFCE91位清0。	p. 106 <input type="checkbox"/>
			PU9寄存器	要以8位或1位为单位读/写PU9寄存器的位8至15，需要将这些位指定为PU9H寄存器的位0至7。	p. 107 <input type="checkbox"/>
			端口 DL	由于只在Flash存储器编程模式下使用FLMD1，因此不需要通过使用端口控制寄存器来操作。详细信息请参考第22章 Flash存储器。	p. 111 <input type="checkbox"/>
			将端口引脚用作复用功能引脚	外部复位后，P05/INTP2/DRST引脚被初始化为片上调试引脚（DRST）。若不想将P05/INTP2/DRST引脚用作片上调试引脚，请参见第24章 片上调试功能。	p. 114 <input type="checkbox"/>
				INTP7引脚的功能与RXDA0引脚的功能是交互的。要用作RXDA0引脚，则使复用功能引脚INTP7的边缘检测功能失效（通过将INTF3L.INTF31位以及INTR3L.INTR31位清0）。要用作INTP7引脚，则停止UARTA0的接收操作（通过将UA0CTL0.UA0RXE位清0）。	p. 115 <input type="checkbox"/>
				KRn引脚的功能与TIQ0m引脚的功能是交互的。要将该引脚用作TIQ0m引脚，则应使复用功能引脚KRn的按键中断检测功能失效（通过将KRM.KRMn位清0）。要将该引脚用作KRn引脚，则应使复用功能引脚TIQ0m的边缘检测功能失效（n = 0至3，m = 0至3）。	p. 116 <input type="checkbox"/>
				DDI, DDO, DCK和DMS引脚是片上调试引脚。若在外部分复位后不想将这些引脚用作片上调试引脚，请参见第24章 片上调试功能。	p. 116 <input type="checkbox"/>
				如果在PFC5.PFC5n以及PFCE5.PFCE5n位的值等于默认值（0）时通过使用PMC5寄存器指定了控制模式，那么输出将会变成未定义的状态。由于这个原因，应先将PFC5.PFC5n和PFCE5.PFCE5n位，然后通过将PMC5n位设为1来设置控制模式。	p. 116 <input type="checkbox"/>
				将PM7n设为1以使用P7n（ANIn）的复用功能。	p. 117 <input type="checkbox"/>
				KR7引脚和RXDA1引脚是复用功能引脚。当将引脚用作RXDA1引脚时，禁止KR7引脚的按键中断检测。（将KRM.KRM7位清0。）同样地，当将引脚用作KR7引脚时，建议将PFC91位设置为1并将PFCE91位清0。	p. 118 <input type="checkbox"/>

章节	分类	功能	功能的详细信息	注意事项	所在页
第4章	软件	端口功能	将端口引脚用作复用功能引脚	由于FLMD1引脚是在Flash存储器编程模式中被使用的，因此FLMD1引脚不需要通过使用端口控制寄存器来操作。详细信息请参见第22章 Flash存储器。	p. 118 <input type="checkbox"/>
				如果在PFC9.PFC9n以及PFCE9.PFCE9n位的值等于默认值（0）时通过使用PMC9寄存器指定了控制模式，那么输出将会变成未定义的状态。由于这个原因，应先将PFC9.PFC9n和PFCE9.PFCE9n位设置为1，然后通过将PMC9n位设为1来设置控制模式。	p. 118 <input type="checkbox"/>
			从端口模式转换为复用功能模式的注意事项	从端口模式切换到复用功能模式的注意事项。 <1> 设置PFCn 和 PFCEn 寄存器：复用功能选择 <2> 将PMCn 寄存器相应的位数设为1：切换到复用功能模式 如果先设置了PMCn 寄存器，需要注意，在这个时候或者根据引脚状态的变化，可能会产生意想不到的作用，其中引脚状态的变化依照PFCn，和 PFCEn 寄存器中的设置。	p. 143 <input type="checkbox"/>
				无论是端口模式/复用功能模式，Pn寄存器按以下方式读写。 • Pn 寄存器的读取：读取端口输出锁值（当 PMn.PMnm 位 = 0 时），或读取引脚状态（PMn.PMnm 位 = 1）。 • Pn 寄存器的写入：写入到端口输出锁。	p. 143 <input type="checkbox"/>
			复用功能模式（输入）的注意事项	当 PMCn.PMCnm 位为0时，由于PMCn 寄存器的AND输出了设置值和引脚电平，因此输入到复用功能块的输入信号为低电平。因此，根据端口设置和复用功能操作使能定时，可能会产生不可预期的操作。因此，要按照以下顺序在端口模式和复用功能模式间切换。 • 要从端口模式切换到复用功能模式（输入） 使用 PMCn 寄存器将引脚设置为复用功能模式，然后允许复用功能操作。 • 要从复用功能模式（输入）切换到端口模式 停止复用功能操作然后将引脚切换到端口模式。	p. 143 <input type="checkbox"/>
第5章	软件	时钟产生功能	时钟发生器	当看门狗定时器2在振荡稳定过程中溢出时，内置振荡器时钟将被选中。	p. 145 <input type="checkbox"/>
			PCC寄存器	在通过CLKOUT向外输出时钟信号时，不要改变CPU时钟的频率（通过CK3到CK0位）。	p. 148 <input type="checkbox"/>
				对CK3位操作时，请使用位操作指令。如果使用8位操作指令，不要改变CK2到CK0位的值。	p. 148 <input type="checkbox"/>
				当停止主时钟工作时，也要同时停止PLL。另外，工作于主时钟频率的片上周边器件也被停止。	p. 149 <input type="checkbox"/>
				如果不满足下面的条件，那么请改变CK2到CK0位的设置以满足该条件，然后再切换到副时钟工作模式。 内部系统时钟（fCLK）> 副时钟（fxT）× 4	p. 149 <input type="checkbox"/>
				只有在主时钟振荡稳定之后才可以使工作于主时钟频率的片上周边器件开始工作。如果它们在振荡稳定之前就被启动，则可能引起误操作。	p. 150 <input type="checkbox"/>
			RCM寄存器	RCM通过设置选项字节所得到的寄存器的设置是有效的。 详细信息，请参见第23章 选项字节功能。	p. 151 <input type="checkbox"/>
				CPU工作于内置振荡器时钟时（CCLS.CCLSF 位 = 1），不可以停止内置振荡器的工作。不要将RSTOP位置1。	p. 151 <input type="checkbox"/>
				即使RSTOP位为1，如果CCLS.CCLSF位被置1内置振荡器也会开始振荡。（振荡稳定时间中产生WDT溢出时）此时，RSTOP位仍为1。	p. 151 <input type="checkbox"/>

章节	分类	功能	功能的详细信息	注意事项	所在页
第5章	软件	时钟产生功能	CCLS寄存器	如果复位释放后在振荡稳定时间中产生WDT的溢出，那么CCLSF位将被置1，并且CCLS寄存器的复位值变为01H。	p. 151 <input type="checkbox"/>
			PLLCTL寄存器	当PLLON位被清零时，SELPLL位也会被自动清零（进入时钟直连模式）。	p. 153 <input type="checkbox"/>
				只有在PLL时钟频率稳定后，SELPLL位才能够被置1。如果时钟频率没有稳定（未入锁），那么无论向该位写入的数据为何值，SELPLL位都会被写入“0”。	p. 153 <input type="checkbox"/>
			LOCKR寄存器	LOCK寄存器不能实时反映PLL的锁定状态。置位和清零条件如下。	p. 154 <input type="checkbox"/>
			PLLS寄存器	将入锁时间设置为800 μs 或更长。	p. 155 <input type="checkbox"/>
				入锁过程中不要改变PLLS寄存器的设置。	p. 155 <input type="checkbox"/>
			PCLM寄存器	先设置与端口相关的控制寄存器（PM，PMC，PFC以及PFCE 寄存器等），然后将PCLE位设为1。	p. 156 <input type="checkbox"/>
				仅在PLL操作过程中将PCLE位设为1。要停止PLL，则将PCLE位清零。	p. 156 <input type="checkbox"/>
			16位定时器/事件计数器P（TMP）	在TPnCE位 = 0时才可对TPnCKS2 到 TPnCKS0 位进行设置。当将TPnCE 位的值从0改变为1时，可同时对TPnCKS2 到 TPnCKS0位进行设置。请务必将第3到第6位清除为“0”。	p. 162 <input type="checkbox"/>
				请务必将TP0SYE和 TP2SYE位清零。	p. 163 <input type="checkbox"/>
				对TPnEST位的设置只有在外部触发脉冲输出模式或单脉冲输出模式才有效。在其它模式下，对该位的置1操作将被忽略。	p. 163 <input type="checkbox"/>
				请务必将第3位和第4位清除为“0”。	p. 163 <input type="checkbox"/>
				在外部事件计数模式下，无论TPnEEE位如何设置，外部事件计数输入都是被选中的。	p. 164 <input type="checkbox"/>
				在TPnCTL0.TPnCE位 = 0时才可对TPnEEE和TPnMD2 到 TPnMD0 位进行设置（TPnCE 位 = 1时只可以向这些位写入相同的值）。如果在TPnCE位 = 1时对这些位进行了改写，那么操作结果将不能被保证。若由于误操作引起了改写，则要将TPnCE位清零然后再次设置这些寄存器位。	p. 164 <input type="checkbox"/>
				在TPnCTL0.TPnCE 位 = 0时才可对TPnOL1， TPnOE1， TPnOL0和 TPnOE0 位进行改写（TPnCE位 = 1时只可以向这些位写入相同的值）。若在TPnCE位 = 1由于误操作引起了改写，则要将TPnCE位清零然后再次设置这些寄存器位。	p. 165 <input type="checkbox"/>
				在TPnCE位和TPnOEm位为0时，即使对TPnOLm位进行了操作，TOPnm引脚的输出电平也不能确定。	p. 165 <input type="checkbox"/>
				在TPnCTL0.TPnCE 位 = 0时才可对TPnIS3 到TPnIS0位进行改写（TPnCE位 = 1时只可以向这些位写入相同的值）。若在TPnCE位 = 1由于误操作引起了改写，则要将TPnCE位清零然后再次设置这些寄存器位。	p. 166 <input type="checkbox"/>
				TPnIS3 到 TPnIS0 位只有在自由运行定时器模式和脉宽测量模式下才有效。其它模式下，无法进行捕捉操作。	p. 166 <input type="checkbox"/>
			TPnIOC2寄存器	在TPnCTL0.TPnCE 位 = 0时才可对TPnEES1， TPnEES0， TPnETS1和 TPnETS0位进行改写（TPnCE位 = 1时只可以向这些位写入相同的值）。若在TPnCE位 = 1由于误操作引起了改写，则要将TPnCE位清零然后再次设置这些寄存器位。	p. 167 <input type="checkbox"/>
				TPnEES1 和 TPnEES0 位只有在TPnCTL1.TPnEEE 位 = 1 或设置了外部事件计数模式（TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0 位 = 001）时才有效。	p. 167 <input type="checkbox"/>
				TPnETS1 和 TPnETS0 位只有在外部触发脉冲输出模式（TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0 位 = 010）或单脉冲输出模式（TPnCTL1.TPnMD2 到 TPnCTL1.TPnMD0 = 011）下才有效。	p. 167 <input type="checkbox"/>

章节	分类	功能	功能的详细信息	注意事项	所在页
第6章	软件	16位定时器/事件计数器P (TMP)	TPnOPT0寄存器	在TPnCTL0.TPnCE 位 = 0时才可对TPnCCS1和TPnCCS0位进行改写 (TPnCE位 = 1时只可以向这些位写入相同的值)。若在TPnCE位 = 1由于误操作引起了改写, 则要将TPnCE位清零然后再次设置这些寄存器位。 请务必将第1到3位, 第6和第7位清除为“0”。	p. 168 <input type="checkbox"/>
			TPnCCR0寄存器	以下情况下禁止访问TPnCCR0寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边I/O寄存器。 • 当CPU工作于副时钟且主时钟振荡停止时 • 当CPU工作于内置振荡时钟时	p. 169 <input type="checkbox"/>
			TPnCCR1寄存器	以下情况下禁止访问TPnCCR1寄存器。详细内容请参考 3.4.8 (2) 访问特定的片上周边I/O寄存器。 • 当CPU工作于副时钟且主时钟振荡停止时 • 当CPU工作于内置振荡时钟时	p. 171 <input type="checkbox"/>
			TPnCNT寄存器	以下情况下禁止访问TPnCNT寄存器。详细内容请参考3.4.8 (2) 访问特定的片上周边I/O寄存器。 • 当CPU工作于副时钟且主时钟振荡停止时 • 当CPU工作于内置振荡时钟时	p. 173 <input type="checkbox"/>
			PnmNFC寄存器	请务必将位3至位5以及位7清除为“0”。	p. 174 <input type="checkbox"/>
				在设置PnmNFC寄存器前, 输入到定时器输入引脚 (TIPnm) 的信号会通过消除的数字噪声来输出。 因此, 先通过使用PnmNFC寄存器来设置采样时钟 (NFC2至NFC0) 以及采样的次数 (NFSTS), 然后等待初始化时间 = (采样时钟) × (采样的次数), 再允许定时器运行。	p. 174 <input type="checkbox"/>
			操作	使用外部事件计数模式时, 请关闭对TIPn0引脚捕捉触发输入的有效边沿的检测 (通过将TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位清零)。	p. 175 <input type="checkbox"/>
				使用外部触发脉冲输出模式, 单脉冲输出模式和脉宽测量模式时, 请选择内部时钟作为计数时钟 (通过将TPnCTL1.TPnEEE位清零)。	p. 175 <input type="checkbox"/>
			TPnCTL1.TPnEEE位	该寄存器位只有在中断请求信号 (INTTPnCC0 和 INTTPnCC1) 被中断屏蔽标志 (TPnCCMK0 和TPnCCMK1) 屏蔽且同时允许了定时器输出 (TOPn1) 时, 才可以被置1。此时, 请将相同的值设置到TPnCCR0 和 TPnCCR1 寄存器中。(请参考 6.5.1 (2) (d) TPnCCR1 寄存器的操作)。	p. 177 <input type="checkbox"/>
			改写TPnCCR0寄存器的注意事项	将TPnCCR0寄存器的值减小时, 要先停止计数, 再对设定值进行改变。 如果在计数过程中将TPnCCR0寄存器的值减小, 那么16位计数器可能产生溢出。	p. 182 <input type="checkbox"/>
			外部事件计数模式操作的寄存器设置	当使用一个外部时钟作为计数时钟时, 外部时钟只能从 TIPn0 管脚输入。这是, 须将 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位清零 (捕捉触发器输入 (TIPn0 管脚): 未测试到边沿)。	p. 188 <input type="checkbox"/>
			外部事件计数模式	在外部事件计数模式中, 不要将TPnCCR0 寄存器设置为0000H。	p. 190 <input type="checkbox"/>
				在外部事件计数模式中, 禁止使用定时器输出功能。如果要将外部事件计数输入作为定时器输出使用, 那么请将工作模式设置为间隔定时器模式并将外部事件计数输入设定为计数时钟 (TPnCTL1.TPnMD2至TPnCTL1.TPnMD0 位 = 000, TPnCTL1.TPnEEE 位 = 1)。	p. 190 <input type="checkbox"/>

章节	分类	功能	功能的详细信息	注意事项	所在页
第6章	软件	16位定时器/事件计数器P (TMP)	改写TPnCCR0寄存器时需要注意的事项	将TPnCCR0寄存器的值减小时，要先停止计数，再对设定值进行改变。 如果在计数过程中将TPnCCR0寄存器的值减小，那么16位计数器可能产生溢出。	p. 191 <input type="checkbox"/>
			TPnIOC0.TPnOE0, TPnOL0位	外部触发脉冲输出模式下不使用TOPn0引脚时，要将该位清零。	p. 196 <input type="checkbox"/>
			操作过程中改变脉冲宽度时需要注意的事项	计数器计数过程中，若改变PWM波形，则要在最后进行对TPnCCR1寄存器的写入操作。 对TPnCCR1寄存器进行写入操作之后，若需再次更改TPnCCRm寄存器的值，须等待下一个INTTPnCC0信号被检测到后，再进行相关操作。	p. 200 <input type="checkbox"/>
			TPnIOC0.TPnOE0, TPnOL0位	当单脉冲输出模式下不使用TOPn0引脚时，要将该位清零。	p. 208 <input type="checkbox"/>
			单脉冲输出模式的寄存器设置	如果 TPnCCR1 寄存器中的设定值大于 TPnCCR0 寄存器中的设定值，即使在单脉冲模式下也不输出单脉冲。	p. 209 <input type="checkbox"/>
			改变TPnCCRm寄存器时需要注意的事项	将TPnCCRm寄存器的值减小时，要先停止计数，再对设定值进行改变。 如果在计数过程中将TPnCCRm寄存器的值减小，那么16位计数器可能产生溢出。	p. 211 <input type="checkbox"/>
			TPnIOC0.TPnOE0, TPnOL0位	当PWM输出模式下不使用TOPn0引脚时，要将该位清零。	p. 215 <input type="checkbox"/>
			定时器调谐操作功能	通过TPmCTL1.TPmSYE及TQ0CTL1.TQ0SYE位来允许或禁止调谐操作模式。对于主定时器TMP2来说，既可以将TMP3或TMQ0单独指定为从定时器，也可以将它们同时指定为从定时器。	p. 246 <input type="checkbox"/>
				通过以下步骤来设置调谐操作模式。 <1> 设置从定时器的TPmCTL1.TPmSYE及TQ0CTL1.TQ0SYE位以允许调谐操作。 将从定时器的TPmCTL1.TPmMD2至TPmCTL1.TPmMD0位以及TQ0CTL1.TQ0MD2至TQ0CTL1.TQ0MD0位设置为自由运行模式。 <2> 利用TPnCTL1.TPnMD2至TPnCTL1.TPnMD0位来设置定时器模式。 这时，不要对主定时器的TPnCTL1.TPnSYE进行设置。 <3> 设置主定时器及从定时器的比较寄存器值。 <4> 设置从定时器的TPmCTL0.TPmCE及TQ0CTL0.TQ0CE位以允许操作内置操作时钟。 <5> 设置主定时器的TPnCTL0.TPnCE位以允许操作内置操作时钟。	p. 246 <input type="checkbox"/>
			选择器功能	当使用选择器功能时，在连接定时器之前设置 TMP 的捕捉触发器输入。	p. 250 <input type="checkbox"/>
				当设置选择器功能时，首先禁止所连接的外设I/O（TMP，TMM0或UARTA）。	p. 250 <input type="checkbox"/>
			SELCNT0寄存器	在以下条件中将INTTMM0EQ0中断信号用作TIP01输入信号。 TMM0操作时钟 ≥ TMP0操作时钟 × 4	p. 251 <input type="checkbox"/>
				要将ISEL02至ISEL04位设为1，应在捕捉输入模式中对相应引脚进行设置。	p. 251 <input type="checkbox"/>
				目标器件（TMP0，TMP1，TMM0，UARTA0或UARTA1）的操作停止时对ISEL02至ISEL04位进行设置。	p. 251 <input type="checkbox"/>
			捕捉操作	当使用捕捉操作并且将低速时钟选择为计数时钟时，如果 TPnCE 位置1之后捕捉触发器被立即输入，那么FFFFH，而不是 0000H，可以被捕捉到 TPnCCR0 和 TPnCCR1 寄存器中。	p. 252 <input type="checkbox"/>

章节	分类	功能	功能的详细信息	注意事项	所在页
第7章	软件	16位定时器/事件计数器 Q (TMQ)	TQ0CTL0寄存器	在 TQ0CE 位 = 0时, 设置TQ0CKS2 至 TQ0CKS0 位。当将TQ0CE位的值从0改变为1时, 可同时对TQ0CKS2 至 TQ0CKS0位进行设置。 请务必将第3到第6位清除为“0”。	p. 258 <input type="checkbox"/>
			TQ0CTL1寄存器	对TQ0EST位的设置只有在外部触发脉冲输出模式或单脉冲输出模式才有效。在其它模式下, 对该位的置1操作将被忽略。 请务必将第3位和第4位清除为“0”。	p. 259 <input type="checkbox"/>
				在外部事件计数模式下, 无论TQ0EEE位如何设置, 外部事件计数输入都是被选中的。	p. 260 <input type="checkbox"/>
				只有在TQ0CTL0.TQ0CE 位 = 0时才可以对TQ0EEE 以及 TQ0MD2 至 TQ0MD0位进行设置 (TQ0CE 位 = 1时, 可以对这些位写入相同的值)。如果在TQ0CE位 = 1时对这些位进行了改写, 那么操作结果将不能被保证。若由于误操作引起了改写, 则要将TQ0CE位清零然后再次设置这些寄存器位。	p. 260 <input type="checkbox"/>
			TQ0IOC0寄存器	只有在TQ0CTL0.TQ0CE 位 = 0时才可以对TQ0OLm 和 TQ0OEm位进行改写 (当 TQ0CE 位 = 1时可以向这些位写入相同的值)。若由于误操作引起了改写, 则要将TQ0CE位清零然后再次设置这些寄存器位。	p. 261 <input type="checkbox"/>
				当 TQ0CE 和 TQ0OEm 位为0时, 即使对TQ0OLm位进行了操作, TQ0Om 引脚的输出电平也不能确定。	p. 261 <input type="checkbox"/>
			TQ0IOC1寄存器	只有在 TQ0CTL0.TQ0CE 位 = 0时才可以对TQ0IS7 至 TQ0IS0进行改写 (TQ0CE 位 = 1时只可以向这些位写入相同的值)。若由于误操作引起了改写, 则要将TQ0CE位清零然后再次设置这些寄存器位。	p. 262 <input type="checkbox"/>
				TQ0IS7 至 TQ0IS0位只有在自由运行定时器模式和脉宽测量模式下才有效。其它模式下, 无法进行捕捉操作。	p. 262 <input type="checkbox"/>
			TQ0IOC2寄存器	在TQ0CTL0.TQ0CE 位 = 0时才可对TQ0EES1, TQ0EES0, TQ0ETS1和 TQ0ETS0位进行改写 (TQ0CE位 = 1时只可以向这些位写入相同的值)。若在 TQ0CE位 = 1由于误操作引起了改写, 则要将TQ0CE位清零然后再次设置这些寄存器位。	p. 263 <input type="checkbox"/>
				TQ0EES1 和 TQ0EES0 位只有在TQ0CTL1.TQ0EEE 位 = 1 或设置了外部事件计数模式 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0位 = 001) 时才有效。	p. 263 <input type="checkbox"/>
				TQ0ETS1 和 TQ0ETS0 位只有在外部触发脉冲输出模式 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 010) 或单脉冲输出模式 (TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 = 011) 下才有效。	p. 263 <input type="checkbox"/>
			TQ0OPT0寄存器	当TQ0CTL0.TQ0CE 位 = 0时可对TQ0CCS3到TQ0CCS0位进行改写 (当TQ0CE 位 = 1时也可以向这些位写入相同的值)。若进行了错误的改写, 须先将TQ0CE位清零, 然后再次设置这些位。 请务必将第1到3位清除为“0”。	p. 264 <input type="checkbox"/>
			TQ0CCR0寄存器	以下情况下禁止访问TQ0CCR0寄存器。详细内容请参考3.4.8 (2) 访问特定的片上周边I/O寄存器。 • 当CPU工作于副时钟且主时钟振荡停止时 • 当CPU工作于内置振荡时钟时	p. 265 <input type="checkbox"/>
			TQ0CCR1寄存器	以下情况下禁止访问TQ0CCR1寄存器。详细内容请参考3.4.8 (2) 访问特定的片上周边I/O寄存器。 • 当CPU工作于副时钟且主时钟振荡停止时 • 当CPU工作于内置振荡时钟时	p. 267 <input type="checkbox"/>

(10/27)

章节	分类	功能	功能的详细信息	注意事项	所在页
第7章	软件	16位定时器/事件计数器 Q (TMQ)	TQ0CCR2寄存器	以下情况下禁止访问TQ0CCR2寄存器。详细内容请参考3.4.8 (2) 访问特定的片上周边I/O寄存器。 • 当CPU工作于副时钟且主时钟振荡停止时 • 当CPU工作于内置振荡时钟时	p. 269 <input type="checkbox"/>
			TQ0CCR3寄存器	以下情况下禁止访问TQ0CCR3寄存器。详细内容请参考3.4.8 (2) 访问特定的片上周边I/O寄存器。 • 当CPU工作于副时钟且主时钟振荡停止时 • 当CPU工作于内置振荡时钟时	p. 271 <input type="checkbox"/>
			TQ0CNT寄存器	以下情况下禁止访问TQ0CNT寄存器。详细内容请参考 3.4.9 (2) 访问特定的片上周边I/O寄存器。 • 当CPU工作于副时钟且主时钟振荡停止时 • 当CPU工作于内置振荡时钟时	p. 273 <input type="checkbox"/>
			Q0mNFC寄存器	请务必将位3至位5以及位7清除为“0”。	p. 274 <input type="checkbox"/>
				在设置Q0mNFC寄存器前，输入到定时器输入引脚 (TIQ0m) 的信号会通过消除的数字噪声来输出。 因此，先通过使用Q0mNFC寄存器来设置采样时钟 (NFC2至NFC0) 以及采样的次数 (NFSTS)，然后等待初始化时间= (采样时钟) × (采样的次数)，再允许定时器运行。	p. 274 <input type="checkbox"/>
			外部事件计数模式	使用外部事件计数模式时，请指定为不检测TIQ00 引脚捕捉触发输入有效沿 (通过将TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位清除为“00”)。	p. 275 <input type="checkbox"/>
			外部触发脉冲输出模式，单脉冲输出模式，脉宽测量模式	使用外部触发脉冲输出模式，单脉冲输出模式和脉宽测量模式时，请选择内部时钟作为计数时钟 (通过将TQ0CTL1.TQ0EEE 位清零)。	p. 275 <input type="checkbox"/>
			TQ0CTL1.TQ0EEE位	该寄存器位只有在中断请求信号 (INTTQ0CC0 和 INTTQ0CCk) 被中断屏蔽标志 (TQ0CCMK0 和 TQ0CCMKk) 屏蔽且同时允许了定时器输出 (TOQ0k) 时，才可以被置1。此时，请将相同的值设置到TQ0CCR0 和TQ0CCRk 寄存器中。(请参考7.5.1 (2) (d) TQ0CCR1 至 TQ0CCR3 寄存器的操作) (k = 1 至 3)。	p. 277 <input type="checkbox"/>
			改写TQ0CCR0寄存器时需要注意的事项	要将TQ0CCR0 寄存器的值改小，首先停止计数然后再更改设置值。 如果在计数时向 TQ0CCR0 寄存器写入更小的值，16位计数器可能会溢出。	pp. 281, 290 <input type="checkbox"/>
			外部事件计数模式操作的寄存器设置	当外部时钟用于计数时钟时，外部时钟只能从TIQ00引脚输入。这时，将TQ0IOC1.TQ0IS1和TQ0IOC1.TQ0IS0置为00 (捕捉触发器输入 (TIQ00 引脚)：无边沿检测)。	p. 287 <input type="checkbox"/>
			外部事件计数模式	在外部事件计数模式中，不要将 TQ0CCR0寄存器设置为0000H。	p. 289 <input type="checkbox"/>
				在外部事件计数模式中，禁止使用定时器输出功能。如果要将外部事件计数输入作为定时器输出使用，那么请将工作模式设置为间隔定时器模式并将外部事件计数输入设定为计数时钟 (TQ0CTL1.TQ0MD2 至TQ0CTL1.TQ0MD0 位 = 000，TQ0CTL1.TQ0EEE 位 = 1)。	p. 289 <input type="checkbox"/>

(11/27)

(12/27)

章节	分类	功能	功能的详细信息	注意事项	所在页
第8章	软件	16位间隔定时器 M (TMM)	TM0CMP0, TM0CTL0寄存器s	TMM0工作时, 禁止改写TM0CMP0 和 TM0CTL0 寄存器的值。 如果在TM0CE位 = 1时对这两个寄存器进行改写, 那么操作结果将不能被保证。 一旦发生上述情况, 请先将TM0CTL0.TM0CE位清零, 之后重置这些寄存器。	p. 367 <input type="checkbox"/>
第9章	软件	钟表定时器功能	PRSM0寄存器	钟表定时器操作期间不要改变的BGCS00 和 BGCS01位的值。	p. 371 <input type="checkbox"/>
				在设置BGCE0为1前, 设置寄存器PRSM0。	p. 371 <input type="checkbox"/>
				根据主时钟频率, 设置寄存器PRSM0和 PRSCM0以获得32.768 kHz的fBRG频率。	p. 371 <input type="checkbox"/>
			PRSCM0寄存器	钟表定时器操作期间不要对寄存器PRSCM0进行写入操作。	p. 372 <input type="checkbox"/>
				在设置PRSM0.BGCE0为1前, 设置寄存器PRCM0。	p. 372 <input type="checkbox"/>
				根据主时钟频率, 设置寄存器PRSM0和 PRSCM0以获得32.768 kHz的fBRG频率。	p. 372 <input type="checkbox"/>
	WTM寄存器	当WTM0 和WTM1均为0时才可对WTM2至WTM7位进行改写操作。	p. 374 <input type="checkbox"/>		
	硬件	注意事项	在允许操作 (WTM.WTM1 和WTM.WTM0 = 1) 后, 第一次钟表定时器中断请求信号 (INTWT) 产生前需要一段准备时间。	p. 377 <input type="checkbox"/>	
经过0.515625 秒 (max.) 产生第1个INTWT (2 ⁹ × 1/32768 = 0.015625 s (max.))。随后INTWT信号每0.5 秒产生1次。			p. 377 <input type="checkbox"/>		
第10章	软件	看门狗定时器2 功能	功能	复位释放后, 看门狗定时器2会自动启动。 当不使用看门狗定时器2, 在复位前通过该功能使其停止工作, 或清除看门狗定时器2并在下一次间隔开始前使其停止工作。 为了确认操作的正确性, 即使在不需要改变默认设置 (复位模式, 间隔时间: fr/2 ¹⁹) 的情况下, 也要对寄存器WDTM2执行一次写操作。	p. 378 <input type="checkbox"/>
				由不可屏蔽中断请求信号 (INTWDT2) 引发的不可屏蔽中断服务, 可参见14.2.2 (2) 信号INTWDT2。	p. 378 <input type="checkbox"/>
			WDTM2寄存器	下述状态下, 禁止访问寄存器WDTM2。详细情况, 参见3.4.8 (2) 访问特定的片上周边I/O寄存器。 ● 当CPU工作于副时钟且主时钟振荡停止时 ● 当CPU工作于内置振荡时钟时	p. 380 <input type="checkbox"/>
				如果OPB1位通过使用选项字节功能被设为1 (参照第23章), 那么复位模式将被固定。	p. 380 <input type="checkbox"/>
				WDCS20至WDCS24位的详细信息, 请参照表10-2 看门狗定时器2的时钟选择。	p. 380 <input type="checkbox"/>
				如果寄存器WDTM2在复位后被复写两次, 那么将会强行产生溢出信号且计数器将会复位。	p. 380 <input type="checkbox"/>
				为了有意产生一个溢出信号, 须写入WDTM2寄存器仅两次或写入一个不是ACH的值到WDTE寄存器仅一次。 然而, 当看门狗定时器2被设置成停止操作时, 即使写入WDTM2寄存器仅两次或写入一个不是“ACH”的值到WDTE寄存器仅一次也不会产生溢出信号。	p. 380 <input type="checkbox"/>
				要停止看门狗定时器2的操作, 设置RCM.RSTOP位为1 (以停止内置晶振) 并向寄存器WDTM2写入1FH。然而, 如果OPB1位通过使用选项字节功能 (参照第23章) 被设为1, 那么看门狗定时器2将不能通过除了复位之外的任务方法来停止。	p. 380 <input type="checkbox"/>

章节	分类	功能	功能的详细信息	注意事项	所在页
第10章	软件	看门狗定时器2 功能	WDTM2寄存器	如果OPB1位通过使用选项字节功能被设为1，时钟将会被固定为内置振荡时钟(f _R)（可以选择2 ¹² /f _R 到2 ¹⁹ /f _R ）。详细信息，请参照第23章 选项字节功能。	p. 381 <input type="checkbox"/>
			WDTE寄存器	如果“ACH”以外的数值被写入寄存器WDTE，必定产生溢出信号。	p. 382 <input type="checkbox"/>
				当向寄存器WDTE执行1位存储器操作指令，必定产生溢出信号。	p. 382 <input type="checkbox"/>
				为了有意产生一个溢出信号，须写入WDTM2寄存器仅两次或写入一个不是“ACH”的值到WDTE寄存器仅一次。	p. 382 <input type="checkbox"/>
				然而，当看门狗定时器2被设置成停止操作时，即使写入WDTM2寄存器仅两次或写入一个不是“ACH”的值到WDTE寄存器仅一次也不会产生溢出信号。	
				读取寄存器WDTE的值为“9AH”（与写入值“ACH”不同）。	p. 382 <input type="checkbox"/>
第11章	硬件	A/D 转换器	ANIO至ANI9 引脚	确保输入ANIO 至ANI9的电压不超过额定值。特别是，如果大于或等于AVREF0的电压输入某个通道，这个通道的转换结果将不确定，其它通道的转换结果也将受到影响。	p. 387 <input type="checkbox"/>
				模拟输入引脚（ANIO 至 ANI9）的复用功能与输入商品引脚（P70至 P79）相同。如果选择ANIO 至 ANI9中的任一引脚来执行A/D转换，那么请不要在转换期间执行端口7的输入指令。如果执行了输入指令，那么转换分辨率将会被降低。	p. 387 <input type="checkbox"/>
	软件		ADA0M0寄存器	下述情况下禁止访问寄存器ADA0M0。详细情况，参见3.4.8（2）访问特定的片上周边I/O寄存器。	p. 388 <input type="checkbox"/>
				• 当CPU工作于副时钟且主时钟振荡停止时	
				• 当CPU工作于内置振荡时钟时	
				0位的写操作将被忽略。	p. 389 <input type="checkbox"/>
				A/D转换被允许时（ADA0CE 位= 1）禁止改变ADA0M1寄存器的值。	p. 389 <input type="checkbox"/>
				如果在A/D转换（ADA0EF 位= 1）期间ADA0M0，ADA0M2，ADA0S，ADA0PFM和ADA0PFT寄存器被写入，则将会根据模式执行以下操作。	p. 389 <input type="checkbox"/>
				• 软件触发模式中 A/D转换停止并从头开始。	
				• 硬件触发模式中 A/D转换停止并进入触发等待状态。	
				当不使用A/D转换器时，设置ADA0CE位0停止操作以减少功耗。	p. 389 <input type="checkbox"/>
				在A/D转换开始后立即输入的引脚的数据第一次转换的分辨率可能会被降低。详细信息，请参见11.6（7）AVREF0引脚。	p. 389 <input type="checkbox"/>
			ADA0M1寄存器	确保将位6至位4清除为“0”。	p. 390 <input type="checkbox"/>
				确保将ADA0HS1位设为“1”。	p. 390 <input type="checkbox"/>
		转换模式设置示例		设置 3.1 μs ≤ 转换时间 ≤ 15.5 μs。	p. 390 <input type="checkbox"/>
				稳定时间内禁止ADA0M0，ADA0M2，ADA0S，ADA0PFM和ADA0PFT寄存器的改写，并禁止触发输入。	p. 390 <input type="checkbox"/>
		ADA0M2寄存器		确保将位7至位2清除为“0”。	p. 391 <input type="checkbox"/>

章节	分类	功能	功能的详细信息	注意事项	所在页
第11章	软件	A/D 转换器	ADA0CRn, ADA0CRnH寄存器s	<p>下述状态禁止访问寄存器ADA0CRn和ADA0CRnH。详细情况, 参见3.4.8 (2) 访问特定的片上周边I/O寄存器。</p> <ul style="list-style-type: none"> 当CPU工作于副时钟且主时钟振荡停止时 当CPU工作于内置振荡时钟时 <p>寄存器ADA0M0 和 ADA0S的写操作可能引起ADA0CRn内容的不确定。转换后, 在写寄存器ADA0M0 和 ADA0S前, 读取转换结果。如果不按上述操作进行, 正确转换结果可能不能读取。</p>	p. 393 <input type="checkbox"/>
			ADA0PFM寄存器	<p>选择模式下, 寄存器ADA0PFT所设置的8位数据与寄存器ADA0CRnH的值进行比较, ADA0CRnH由ADA0S进行设置。如果结果与ADA0PFC位所设置的条件匹配, 转换结果存储于寄存器ADA0CRn并产生中断信号INTAD。如果不匹配, 不产生中断信号INTAD。</p> <p>扫描模式下, 寄存器ADA0PFT所设置的8位数据与寄存器ADA0CR0H的值进行比较。如果结果与ADA0PFC位所设置的条件匹配, 转换结果存储于寄存器ADA0CR0并产生中断信号INTAD。如果不匹配, 不产生中断信号INTAD。无论比较结果如何, 扫描模式继续工作, 转换结果存储于寄存器ADA0CRn中, 直到扫描操作完成。但是, 扫描操作完成后不产生中断信号INTAD。</p>	p. 393 <input type="checkbox"/> p. 395 <input type="checkbox"/>
			不使用A/D转换器时	当不使用A/D转换器时, 通过将ADA0M0.ADA0CE 位清零可以使功耗减少。	p. 408 <input type="checkbox"/>
			ANI0 至 ANI9 引脚的输入范围	输入指定范围的电压值到引脚ANI0 至 ANI9。如果大于等于AVREF0 或小于等于AVss (即使在最大绝对值范围内) 的电压被输入到这些引脚, 则那个通道的转换值不确定, 其它通道的转换值也会受到影响。	p. 408 <input type="checkbox"/>
			抑制噪声的方法	为了确保10位分辨率, 引脚ANI0 至ANI9必须有效抑制噪声。噪声影响随着模拟输入源的输出阻抗的增加而提高。为了降低噪声, 推荐连接一个外部电容, 如图11-9所示。	p. 408 <input type="checkbox"/>
			I/O复用	<p>模拟输入引脚 (ANI0 至 ANI9) 可功能复用为普通端口引脚。当从引脚ANI0 至 ANI9中选择一个执行A/D转换时, 不要在转换期间执行读/写端口指令, 否则转换分辨率将下降。</p> <p>在转换期间设置引脚为输出端口同样会使转换分辨率将下降。其原因是连接端口引脚的外部电路导致输出电流波动。</p> <p>如果某引脚正在进行A/D转换, 其临近引脚输入一个数字脉冲, 则由于耦合噪声的影响, A/D转换值可能不准确。因此, 在A/D转换过程中, 确保不使用临近引脚传输脉冲信号。</p>	p. 408 <input type="checkbox"/>
			中断请求标志 (ADIF)	即使寄存器ADA0S的内容改变, 中断请求标志 (ADIF) 也不清零。因此, 如果在A/D转换过程中模拟输入引脚改变, 上次所选择的模拟输入信号的转换结果可能被存储, 转换结束中断请求标志可能在寄存器ADA0S复写前立刻置1。如果寄存器ADA0S复写后立即读取标志ADIF, 标志ADIF可能置1, 即使新选择的模拟输入信号的转换还未完成。当A/D转换停止, 在再次转换前使标志ADIF清零。	p. 409 <input type="checkbox"/>

章节	分类	功能	功能的详细信息	注意事项	所在页
第11章	硬件	A/D 转换器	AV _{REF0} 引脚	<p>(a) 引脚AV_{REF0}被用作A/D转换器的供电电压引脚，同时也为复用功能端口供电。当使用备用供电电源时，确保向引脚AV_{REF0}提供与VDD相同的电压，如图11-12所示。</p> <p>(b) 引脚AV_{REF0}也被用作A/D转换器的参考电压引脚。如果提供给引脚AV_{REF0}的源供电电源具有高阻抗，或者如果供电电源低电流负载能力低，参考电压可能被转换中的电流影响（特别是，转换操作使能位ADA0CE刚刚置1之后）。结果，转换精度会下降。如图11-12所示，为了避免上述情况，推荐连接一个电容在引脚AV_{REF0}和AV_{SS}之间以抑制参考电压的波动。</p> <p>(c) 如果提供给引脚AV_{REF0}的源供电电源具有高直流阻抗（例如，由于插入一个二极管），当转换使能时的电压可能比转换停止时的电压低，因为A/D转换电流引起电压失真。</p>	p. 410 <input type="checkbox"/>
	软件		读取ADA0CRn寄存器	当执行寄存器ADA0M0至ADA0M2，ADA0S，ADA0PFM或ADA0PFT的写指令时，寄存器的ADA0CRn内容可能不确定。转换完成后，在写入寄存器ADA0M0至ADA0M2，ADA0S，ADA0PFM和ADA0PFT之前，读取转换结果。另外，当接收到一个外部/定时器触发信号时，ADA0CRn寄存器的内容可能是不确定的。应在转换完成之后及在下一个外部/定时器触发信号收到之前读取转换结果。在与上述时序不同时，正确转换结果可能无法读取。	p. 410 <input type="checkbox"/>
			A/D转换结果	如果在模拟输入引脚及参考电压输入引脚上有噪声，那么该噪声可能会产生一个非法转换结果。因此，需要通过软件处理来避免该非法转换结果对于系统的负效应。软件处理的示例如下所示。 <ul style="list-style-type: none"> 取若干A/D转换的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。 如果获得一个可以判断为已经生成了一个系统故障的A/D转换结果，那么在执行故障处理前请务必复查系统故障。 	p. 410 <input type="checkbox"/>
			等待模式	因为A/D转换器在STOP模式下停止工作，转换结果无效，所以功耗可以降低。释放STOP模式后，恢复工作，但STOP模式释放后的A/D转换结果仍然无效。当STOP模式释放后使用A/D转换器时，在设置STOP模式或释放STOP模式之前，使ADA0M0.ADA0CE 位清 0，然后在释放STOP模式后将ADA0CE 位置 1。在IDLE1，IDLE2或副时钟操作模式下，继续运行。因此，为了减小功耗，应使ADA0M0.ADA0CE = 0。在IDLE1，IDLE2模式下，由于模拟输入电压值不能保留，IDLE1，IDLE2模式释放后的A/D转换结果无效。进入IDLE1，IDLE2模式前的转换结果有效。	p. 411 <input type="checkbox"/>
			稳定时间内改写寄存器及触发输入	禁止在稳定时间内改写ADA0M0，ADA0M2，ADA0S，ADA0PFM和ADA0PFT寄存器及触发输入。	p. 411 <input type="checkbox"/>

章节	分类	功能	功能的详细信息	注意事项	所在页
第11章	软件	A/D 转换器	A/D 转换结果的误差	A/D 转化的结果会根据供电电压的波动或可能受到干扰而变化。要减小误差，须采取软件抗干扰措施，例如A/D转换结果平滑处理。	p. 411 <input type="checkbox"/>
	硬件		A/D转换结果滞后特性	逐次逼近A/D转换器维持内部采样和保持电容的模拟输入电压，同时进行A/D转换。A/D转换结束后模拟输入电压保留在内部采样和保持电容中。结果，产生下述现象。 <ul style="list-style-type: none"> 当同一个通道用于A/D转换，如果电压高于或低于先前A/D转换，则出现滞后特征，转换结果受到上次转换值影响。因此，即使同一个模拟输入电压也会产生不同结果。 当开关模拟输入通道时，可能出现滞后特征，转换结果受到上次转换通道影响。这是因为只有一个A/D转换器用于A/D转换。因此，即使同一个模拟输入电压也会产生不同结果。 因此，要获取更多准确的转换结果，应对同一通道连续执行两次A/D转换，并丢弃第一个转换结果。	p. 411 <input type="checkbox"/>
第12章	软件	异步串行接口 A (UARTA)	UAnOPT0寄存器	在SBF接收期间 (UAnSRF位=1), 不要将UAnSRT和UAnSTT置1。	p. 421 <input type="checkbox"/>
			SBF 接收	如果SBF在数据接收期间被传输，则将发生帧错误。	p. 431 <input type="checkbox"/>
				在SBF接收期间 (UAnSRF = 1) 请不要将SBF接收触发位 (UAnSRT) 及SBF发送触发位 (UAnSTT) 设为1。	p. 431 <input type="checkbox"/>
			连续发送	在连续发送过程中执行发送初始化，确保UAnSTR.UAnTSF 位为 0，然后进行初始化。当UAnTSF = 1时初始化数据，其发送过程不能保证。在连续发送的情况下，下个数据中从停止位至起始位的通信率将会较正常通信率扩展两个时钟。	p. 433 <input type="checkbox"/>
			UART接收	即使产生接收错误，也要确保读取寄存器UAnRX。如果寄存器UAnRX不被读取，在接收下一个数据期间会产生溢出错误，而且接收错误会不确定的连续发生。	p. 435 <input type="checkbox"/>
				在接收期间并假定只有一个停止位时，操作被执行，第二个停止位被忽略。	p. 435 <input type="checkbox"/>
				在接收完成，产生接收完成中断请求信号后，读取寄存器UAnRX，并将UAnPWR位或UAnRXE位清零。如果UAnPWR位或UAnRXE位在信号INTUAnR产生前被清零，寄存器UAnRX的读取值不被保证。	p. 435 <input type="checkbox"/>
				如果UARTAn的接收完成处理过程（产生信号INTUAnR）和UAnPWR = 0或UAnRXE = 0产生冲突，无论数据是否被存储到寄存器UAnRX中都产生信号INTUAnR。为了在不等待INTUAnR信号产生的情况下完成接收，设置中断控制寄存器 (UAnRIC) 的中断屏蔽标志 (UAnRMK) 为1，以及UAnPWR位 = 0 或UAnRXE位 = 0后，确保寄存器UAnRIC的中断请求标志 (UAnRIF) 清零。	p. 435 <input type="checkbox"/>
			接收错误	当产生信号INTUAnR，寄存器UAnSTR必须被读取以检测错误。	p. 436 <input type="checkbox"/>
				如果连续接收过程中产生接收错误中断，在下次接收完成前必须读取寄存器中必需读取的内容，并执行错误处理。	p. 437 <input type="checkbox"/>
			LIN 功能	当使用LIN功能，将UAnCTL0.UAnPS1 及 UAnCTL0.UAnPS0 位固定为00。	p. 438 <input type="checkbox"/>

章节	分类	功能	功能的详细信息	注意事项	所在页
第12章	软件	异步串行接口 A (UARTA)	UAnCTL1寄存器	在复写寄存器UAnCTL1前，将UAnCTL0.UAnPWR位清零。	p. 441 <input type="checkbox"/>
			UAnCTL2寄存器	在复写寄存器UAnCTL2前，将UAnCTL0.UAnPWR位清零或将UAnTXE 和 UAnRXE位清为00。	p. 442 <input type="checkbox"/>
			波特率误差	发送期间必须保持波特率误差在接收端允许的误差范围内。	p. 443 <input type="checkbox"/>
				接收期间，波特率误差必须满足“（5）接收期间允许的波特率范围”中所描述的范围。	p. 443 <input type="checkbox"/>
			接收期间允许的波特率范围	必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。	p. 445 <input type="checkbox"/>
			当提供给UARTAn的时钟停止时	当UARTAn所使用的时钟停止工作（例如，在IDLE1，IDLE2，或STOP模式中），操作将停止，每个寄存器会在时钟停止前立即保持其所具有的值。TXDAn引脚输出在时钟停止前也会立即保持和输出它所具有的值。然而，时钟恢复后，操作不被保证。因此，时钟恢复后，应该通过设置UAnCTL0.UAnPWR，UAnCTL0.UAnRXEn 和UAnCTL0.UAnTXEn 位为 000来使电路初始化。	p. 448 <input type="checkbox"/>
			RXDA1引脚 KR7引脚	引脚RXDA1 和KR7不能同时使用。要RXDA1引脚，则不要使用KR7引脚。要使用KR7引脚，则不要使用RXDA1引脚（推荐设置PFC91 为1且PFCE91 为0）。	p. 448 <input type="checkbox"/>
			接收数据发送	在UARTAn中，不会产生由于通信错误引起的中断。在通信期间读取UAnSTR寄存器以检查错误。	p. 448 <input type="checkbox"/>
			UARTAn启动	按照下述次序启动UARTAn。 <1> 设置UAnCTL0.UAnPWR为1。 <2> 设置端口。 <3> 设置UAnCTL0.UAnTXE位为 1，UAnCTL0.UAnRXE位为 1。	p. 448 <input type="checkbox"/>
			UARTAn停止	按照下述次序停止UARTAn。 <1> 设置 UAnCTL0.UAnTXE位为0，UAnCTL0.UAnRXE 位为 0。 <2> 设置端口，设置UAnCTL0.UAnPWR 位为0（如果端口设置没有改变则不会有问題）。	p. 448 <input type="checkbox"/>
			发送模式	在发送模式中（UAnCTL0.UAnPWR 位= 1 且 UAnCTL0.UAnTXE 位= 1），不要通过软件将相同的值写到寄存器UAnTX中，因为发送操作从值写入寄存器就开始了。要连续发送相同的值，则复写同样的值。	p. 448 <input type="checkbox"/>
			连续发送	在连续发送中，从停止位到下一个起始位的通信速率比正常情况下的延长了2个基本时钟。然而，接收端通过检测起始位来初始化时序，所以接收结果不受影响。	p. 448 <input type="checkbox"/>
第13章	软件	3线长度可变串行I/O (CSIB)	CBnCTL0寄存器	为了强制延缓发送/接收，应将CBnPWR位清零，而不是将CBnRXE位清零。这时，时钟输出停止。	p. 452 <input type="checkbox"/>
				确保将位2和3位清0。	p. 454 <input type="checkbox"/>
			CBnCTL1寄存器	只有当CBnCTL0.CBnPWR = 0或CBnCTL0.CBnTXE以及CBnRXE位 = 0时，寄存器CBnCTL1可被复写。	p. 455 <input type="checkbox"/>
				将通信时钟（fCLK）设置到8 MHz或更低。	p. 455 <input type="checkbox"/>
			CBnCTL2寄存器	只有当CBnCTL0.CBnPWR = 0或者CBnTXE 和 CBnRXE位都为0时，寄存器CBnCTL2可被改写。	p. 456 <input type="checkbox"/>

章节	分类	功能	功能的详细信息	注意事项	所在页
第13章	软件	3线长度可变串行I/O (CSIB)	连续发送模式 (主模式, 发送模式)	在连续发送模式中, 接收完成中断请求信号 (INTCBnR) 不会被生成。	p. 473 <input type="checkbox"/>
			连续发送模式 (从模式, 发送模式)	在连续发送模式中, 不会生成接收完成中断请求信号 (INTCBnR)。	p. 482 <input type="checkbox"/>
			时钟时序	在单独发送模式中, 将设为1的CBnTSF位写入CBnTX寄存器的操作将被忽略。这在发送期间不会对操作产生任何影响。	pp. 491, 492 <input type="checkbox"/>
			PRSM0寄存器	在钟表定时器及CSIB0操作时请不要复写寄存器PRSM0。	p. 494 <input type="checkbox"/>
				在将BGCE0位设置为1之前设置寄存器PRSM0。	p. 494 <input type="checkbox"/>
			PRSCM0寄存器	在钟表定时器及CSIB操作时请不要复写寄存器PRSCM0。	p. 495 <input type="checkbox"/>
				在将PRSM0.BGCE0位设置为1之前设置寄存器PRSCM0。	p. 495 <input type="checkbox"/>
			波特率产生	将fBRGm设置为8 MHz或更低。	p. 495 <input type="checkbox"/>
			CBnCTL0寄存器 CBnCTL1寄存器 CBnCTL2寄存器	考虑到寄存器禁止在操作期间 (CBnCTL0.CBnPWR位为1) 复写, 如果操作期间由于错误操作而复写寄存器, 则设置CBnCTL0.CBnPWR位为0, 然后初始化CSIBn。 禁止在操作期间复写的寄存器如下。 • CBnCTL0寄存器: CBnTXE, CBnRXE, CBnDIR, CBnTMS 位 • CBnCTL1寄存器: CBnCKP, CBnDAP, CBnCKS2至CBnCKS0 位 • CBnCTL2寄存器: CBnCL3至CBnCL0位	p. 496 <input type="checkbox"/>
第14章	软件	中断/异常处理功能	不可屏蔽中断	在通信类型2或4 (CBnCTL1.CBnDAP = 1) 中, 产生接收完成中断 (INTCBnR) 后, \overline{SCKBn} 时钟执行到一半时CBnSTR.CBnTSF被清除。在单独发送模式中, 通信过程中 (CBnTSF = 1) 忽略写入的下一个发送数据, 并且下一次通信不开始。同样, 如果只能接收的通信被设置 (CBnCTL0.CBnTXE = 0, CBnCTL0.CBnRXE = 1), 即使接收到的数据在通信过程中 (CBnTSF = 1) 被读取, 下一次通信也不会开始。 因此, 在通信模式2或4 (CBnDAP = 1) 下的单独发送模式, 特别要注意下述问题。 • 要开始下一次发送, 确保CBnTSF = 0, 然后向寄存器CBnTX写入要发送的数据。 • 只能接收通信模式下 (CBnTXE = 0, CBnRXE = 1), 要连续执行下一次接收, 应确保CBnTSF = 0, 然后读取寄存器CBnTX。 或者, 使用连续发送模式来代替单独发送模式。	p. 496 <input type="checkbox"/>
			可屏蔽中断	在INTWDT2例子中的还原, 见14.2.2 (2) 通过INTWDT2 信号。	p. 498 <input type="checkbox"/>
				在不可屏蔽中断服务期间, 当EP位和NP位通过LDSR指令改变时, 在RETI指令执行之后, 为了正确的还原PC和PSW, 必须在RETI指令执行之前使用LDSR指令将EP位设置为0, NP位设置为1。	p. 503 <input type="checkbox"/>
			可屏蔽中断	在可屏蔽中断服务执行期间, 当通过LDSR指令将EP位和NP位改变时, 为了在RETI指令后正确的恢复PC和PSW, 需要在RETI指令之前执行LDSR指令将EP位和NP位清0。	p. 507 <input type="checkbox"/>

(19/27)

章节	分类	功能	功能的详细信息	注意事项	所在页
第14章	软件	中断/异常处理功能	多次中断	为了执行复合中断服务，在执行EI指令之前必须保存EIPC寄存器和EIPSW寄存器的值。当从复合中断服务中返回，在执行DI指令后恢复EIPC和EIPSW的值。	pp. 509至511 <input type="checkbox"/>
			中断控制寄存器	禁止中断（DI）或者屏蔽中断来读取xxICn.xxIFn位。如果在允许中断（EI）或是中断未被屏蔽的时候，读取xxIFn位，确认中断且读取该位冲突时数据可能有误。	p. 512 <input type="checkbox"/>
				如果确认中断信号，xxIFn标志将通过硬件自动复位。	p. 512 <input type="checkbox"/>
			IMR0至IMR2寄存器	设备文件定义了xxICn.xxMKn位为保留字。如果有其它的位使用xxMKn的名称进行操作，xxICn寄存器的内容会被写入，以代替IMRm寄存器（作为结果，IMRm寄存器的内容也会被覆盖）。	p. 514 <input type="checkbox"/>
				为了以8位或1位为单位读取IMR0到IMR2寄存器的位8到15，需要将它们设定为IMR0H到IMR2H寄存器的位0到7。	p. 514 <input type="checkbox"/>
				将IMR2寄存器的位15到11以及位7至4置“1”。如果这些位的设定值改变，操作将不会得到保证。	p. 514 <input type="checkbox"/>
			ISPR寄存器	如果在中断允许状态（EI）对ISPR寄存器进行读取时确认一个中断，在寄存器的位通过确认中断的方式被设置之后，可以读取 ISPR寄存器的值。为了在确认中断之前正确的读取ISPR寄存器的值，则需在禁止中断（DI）时读取寄存器。	p. 515 <input type="checkbox"/>
			从软件异常处理中恢复	在软件异常处理过程中，当通过LDSR指令改变EP位和NP位时，为了通过RETI指令恢复期间正确的恢复PC和PSW，在RETI指令执行前必须立即执行LDSR指令将EP位置1，NP位清0。	p. 518 <input type="checkbox"/>
			非法操作码定义	由于将来有可能分配该指令到一个非法的操作代码，推荐不要使用它。	p. 520 <input type="checkbox"/>
			从非法操作码中恢复	只能在执行非法操作码以及DBRET指令的间隔时间内访问DBPC和DBPSW。	p. 521 <input type="checkbox"/>
			从调试陷阱中恢复	只能在执行DBTRAP指令以及DBRET指令间的间隔时间内访问DBPC和DBPSW。	p. 523 <input type="checkbox"/>
			INTF0, INTR0寄存器s	当引脚的功能从外部中断功能（复用功能）转变到端口功能时，可以进行边沿检测。因此，首先要将INTF0n位和INTR0n位设定为00，然后再设定端口模式。	p. 525 <input type="checkbox"/>
				如果相应的引脚没有被用作NMI或INTP0到INTP3引脚，那么一定要将INTF0n和INTR0n位设定为00。	p. 525 <input type="checkbox"/>
			INTF3, INTR3寄存器s	当引脚的功能从外部中断功能（复用功能）转变到端口功能时，可以进行边沿检测。因此，首先要将INTF0n位和INTR0n位设定为00，然后再设定端口模式。	p. 526 <input type="checkbox"/>
				如果相应的引脚没有被用作INTP7 引脚，那么一定要将INTF31和INTR31位设定为00。	p. 526 <input type="checkbox"/>
			INTF9H, INTR9H寄存器s	当引脚的功能从外部中断功能（复用功能）转变到端口功能时，可以进行边沿检测。因此，首先要将INTF9n位和INTR9n位设定为0，然后再设定端口模式。	p. 527 <input type="checkbox"/>
				如果相应的引脚没有被用作INTP4至INTP6引脚，则一定要将INTF9n和INTR9n位设定为00。	p. 527 <input type="checkbox"/>

(20/27)

章节	分类	功能	功能的详细信息	注意事项	所在页
第14章	软件	中断/异常处理功能	NFC寄存器	采样时钟被改变后，一直到数字噪声消除器被初始化，之间将会需要与NFSTS位所设置的采样次数和采样时钟的乘积相同的时间。因此，在NFSTS位所设置的采样次数和采样时钟的乘积相同的时间过去前，如果INTP3的有效边沿在采样时钟改变后被输入，那么可能会生成中断请求信号。因此使用中断功能时应注意以下几点。 • 使用中断功能时，在NFSTS位所设置的采样次数和采样时钟的乘积相同的时间过去后，将会在中断请求标志（PIC3.PIF3位）清除后允许中断。	p. 528 <input type="checkbox"/>
			NMI引脚	NMI引脚的复用功能与P02引脚相同。复位后它的功能相当于一个正常的端口。为了允许NMI引脚，需用通过PMC0寄存器使NMI引脚有效。NMI引脚的初始设定是“没有检测到边沿”。使用INTF0寄存器和INTR0寄存器设定NMI引脚的有效边沿。	p. 531 <input type="checkbox"/>
第15章	软件	按键中断功能	KRM寄存器	在KRM寄存器被清零（00H）后，重写KRM寄存器。	p. 533 <input type="checkbox"/>
				如果改变KRM寄存器，则会产生中断请求信号（INTKR）。为避免这种情况，在禁止中断（DI）或屏蔽中断之后改变KRM寄存器，然后将中断请求标志（KRIC、KRIF位）清零，再允许中断或清除屏蔽。	p. 533 <input type="checkbox"/>
			KR0至KR7引脚	如果给KR0至KR7中的任何一个输入低电平，即使其它引脚输入下降沿，也不会产生INTKR信号。	p. 533 <input type="checkbox"/>
			RXDA1引脚 KR7引脚	不能同时使用RXDA1和KR7引脚。如果使用RXDA1引脚，就不能使用KR7引脚。如果使用KR7引脚，就不能使用RXDA1引脚（推荐设置PFC91位为1并且将PFCE91位清零）。	p. 533 <input type="checkbox"/>
			使用按键中断功能	为了使用按键中断功能，确定设置端口引脚为按键返回引脚，然后使用KRM寄存器允许操作。为了从按键返回引脚切换到端口引脚，使用KRM寄存器禁止操作，然后设置为端口引脚。	p. 533 <input type="checkbox"/>
第16章	软件	待机功能	PSC寄存器	在设置IDLE1、IDLE2、STOP或副IDLE模式之前，先设置PSMR.PSM1和PSMR.PSM0位，然后设置STP位。	p. 536 <input type="checkbox"/>
				当释放HALT模式时对NMI1M、NMI0M和INTM位进行设置是非法的。	p. 536 <input type="checkbox"/>
				如果STP位设为1的同时NMI1M、NMI0M或INTM位也设为1，那么NMI1M、NMI0M或INTM位的设置无效。当设置为IDLE1/IDLE2/STOP模式时，如果一个未屏蔽的中断请求信号保持为未决定的状态，则将与中断请求信号（NMI1M、NMI0M或INTM）相应的位设为1，然后将STP位设为1。	p. 536 <input type="checkbox"/>
			PSMR寄存器	确定将第2位至第7位清除为“0”。	p. 537
				仅当PSC.STP位为1时，PSM0和PSM1位是有效的。	p. 537
			OSTS寄存器	STOP模式释放后的等待时间不包括从释放STOP模式到时钟振荡启动这段时间（即下图“a”所示的部分），不管通过复位输入或中断请求信号的产生是否释放了STOP模式。	p. 538
				确定将第3位至第7位清除为“0”。	p. 538
				复位释放后的振荡稳定时间是 $2^{16}/f_x$ （因为OSTS寄存器的初始值=06H）。	p. 538
			HALT模式	在HALT指令之后插入五个或更多的NOP指令。	p. 539
				如果在执行HALT指令时有一个未屏蔽的中断请求被保持等待响应，则此状态被转移到HALT模式中，然后通过这个等待响应的中断请求，HALT模式被立即释放。	p. 539

(21/27)

章节	分类	功能	功能的详细信息	注意事项	所在页
第16章	软件	待机功能	IDLE1模式	在执行往PSC寄存器中存入数据来设置IDLE1模式的指令之后，插入五个或更多的NOP指令。	p. 541
				如果设置IDLE1模式时有一个未被屏蔽的中断请求信号被保持等待响应，则IDLE1模式立即被此等待响应的中断请求释放。	p. 541
			释放IDLE1模式	通过设置PSC.NMI1M, PSC.NMI0M和PSC.INTM位为1来禁止中断请求信号，这样中断请求信号是无效的，IDLE1模式也不会被释放。	p. 541
				如果通过使用NFC寄存器来选择消除数字噪声的方法且如果采样时钟是从fxx/64, fxx/128, fxx/256, fxx/512以及fxx/1024中选择的，那么IDLE1模式将不能通过INTP3引脚的中断请求信号来释放。详细信息请参见14.6.2 (4) 噪声消除控制寄存器 (NFC)。	p. 541
			IDLE2模式	在执行往PSC寄存器中存入数据来设置IDLE2模式的指令之后，插入五个或更多的NOP指令。	p. 543
				如果设置IDLE2模式时有一个未被屏蔽的中断请求信号被保持等待响应，则IDLE2模式立即被此等待响应的中断请求释放。	p. 543
			释放IDLE2模式	通过设置PSC.NMI1M, PSC.NMI0M和PSC.INTM位为1来禁止中断请求信号，这样中断请求信号是无效的，IDLE2模式也不会被释放。	p. 543
				如果通过使用NFC寄存器来选择消除数字噪声的方法且如果采样时钟是从fxx/64, fxx/128, fxx/256, fxx/512以及fxx/1024中选择的，那么IDLE2模式将不能通过INTP3引脚的中断请求信号来释放。详细信息请参见14.6.2 (4) 噪声消除控制寄存器 (NFC)。	p. 543
			STOP 模式	在执行往PSC寄存器中存入数据来设置STOP模式的指令之后，插入五个或更多的NOP指令。	p. 546
				如果设置STOP模式时有一个未被屏蔽的中断请求信号被保持等待响应，则STOP模式立即被此等待响应的中断请求释放。	p. 546
			释放STOP 模式	通过设置PSC.NMI1M, PSC.NMI0M和PSC.INTM位为1来禁止中断请求信号，这样中断请求信号是无效的，STOP模式也不会被释放。	p. 546 <input type="checkbox"/>
				如果通过使用NFC寄存器来选择消除数字噪声的方法且如果采样时钟是从fXX/64, fXX/128, fXX/256, fXX/512以及fXX/1024中选择的，那么STOP模式将不能通过INTP3引脚的中断请求信号来释放。详细信息请参见14.6.2 (4) 噪声消除控制寄存器 (NFC)。	p. 546 <input type="checkbox"/>
			STOP 模式中的操作状态	当操作A/D转换器时，如果设置STOP模式，则A/D转换器自动停止操作，并且在STOP模式释放后再次启动操作。然而，在这种情况下，释放STOP模式后的A/D转换结果是无效的。所有设置STOP模式之前的A/D转换结果也是无效的。	p. 547 <input type="checkbox"/>
				即使在操作A/D转换器时设置STOP模式，功耗降低也与在设置STOP模式之前停止A/D转换器时的一样。	p. 547 <input type="checkbox"/>
			副时钟操作模式	当操作CK3位时，不要改变PCC.CK2至PCC.CK0位（推荐使用位操作指令来操作）的设置值。对于PCC寄存器的细节描述，参见5.3 (1) 处理器时钟控制寄存器 (PCC)。	p. 549 <input type="checkbox"/>
				如果以下的条件不满足，改变CK2至CK0位的设置以便满足条件并且设置副时钟操作模式。 内部主时钟 (fCLK) > 副时钟 (fXT) ×4	p. 549 <input type="checkbox"/>

(22/27)

章节	分类	功能	功能的详细信息	注意事项	所在页
第16章	软件	待机功能	释放副时钟操作模式	当操作CK3位时，不要改变CK2至CK0位（推荐使用位操作指令来操作）的设置值。 对于PCC寄存器的细节描述，参见5.3（1）处理器时钟控制寄存器（PCC）。	p. 549 <input type="checkbox"/>
			副时钟操作模式中的操作状态	请确保在停止主时钟前停止PLL（PLLCTL.PLLON = 0）。	p. 550 <input type="checkbox"/>
				当CPU工作在副时钟并且主时钟振荡停止时，不能访问发生等待的寄存器。如果产生等待，则其只能通过复位来释放（参见3.4.8（2））。	p. 550 <input type="checkbox"/>
			副IDLE模式	为设置副IDLE模式而将指令保存到PSC寄存器后，插入五个或更多的NOP指令。	p. 551 <input type="checkbox"/>
				如果设置副IDLE模式时有一个未被屏蔽的中断请求信号被保持等待响应，则副IDLE模式立即被此等待响应的中断请求释放。	p. 551 <input type="checkbox"/>
			释放副IDLE模式	通过设置PSC.NMI1M，PSC.NMI0M和PSC.INTM位为1来禁止中断请求信号，这样中断请求信号是无效的，副IDLE模式也不会被释放。	p. 552 <input type="checkbox"/>
				当释放副IDLE模式时，从产生释放副IDLE模式的中断请求信号开始到副IDLE模式被释放之间须等待12个副时钟周期（大约366 μs）。	p. 552 <input type="checkbox"/>
				如果通过使用NFC寄存器来选择消除数字噪声的方法且如果采样时钟是从fxx/64，fxx/128，fxx/256，fxx/512以及fxx/1024中选择的，那么副IDLE模式将不能通过INTP3引脚的中断请求信号来释放。详细信息请参见14.6.2（4）噪声消除控制寄存器（NFC）。	p. 552 <input type="checkbox"/>
			在副IDLE模式中的操作状态	停止主时钟之前确定停止PLL（PLLCTL.PLLON位 = 0）。	p. 553 <input type="checkbox"/>
				为了实现低功耗，在进入副IDLE模式之前停止A/D转换器。	p. 553 <input type="checkbox"/>
第17章	软件	复位功能	紧急操作模式	当以内部振荡时钟来操作CPU时，禁止访问生成有等待状态的寄存器。关于生成有等待状态的寄存器的详细信息，请参见3.4.8（2）访问特殊片上外围I/O寄存器。	p. 554 <input type="checkbox"/>
			RESF寄存器	只有“0”可以被写入到该寄存器中的每一位中。如果写入的“0”和设置标志（复位事件）冲突，则设置标志优先。	p. 555 <input type="checkbox"/>
	硬件		复位后的内部RAM状态	V850ES/HE2中的固件会在内部系统复位状态被释放后使用部分内部RAM，这是因为它支持引导置换功能。因此，在上电复位后一些RAM区域中的内容将不会被保留。详细信息请参见17.4 复位释放后的操作。	pp. 556, 558 <input type="checkbox"/>
			RESET引脚的硬件状态输入	当电源开启时，下列引脚可能会在复位时临时输出一个不确定的电平。 ● P53/KR3/TIQ00/TOQ00/DDO引脚	p. 556 <input type="checkbox"/>
				OCDM寄存器通过RESET引脚输入来初始化。因此，需要注意的是，如果在OCDM.OCDM0的位数被清除前，一个高电平在复位释放后被输入到P05/DRST引脚中，则可能会进入片上调试模式。详情请参见第4章 端口功能。	p. 556 <input type="checkbox"/>
第18章	软件	时钟监控器	CLM寄存器	一旦CLME位被设为1，除了复位其余任何操作都不能将其清为0。	p. 563 <input type="checkbox"/>
				当通过时钟监控器执行复位时，CLME位将被清为0并且RESF.CLMRF被设置为1。	p. 563 <input type="checkbox"/>
			内部振荡器	内部振荡器可以通过使用选项字节功能并将RCM.RSTOP位设为1来停止（参见第23章）。	p. 564 <input type="checkbox"/>
				内部振荡器停止的时候时钟监控器也停止。	p. 564 <input type="checkbox"/>

(23/27)

章节	分类	功能	功能的详细信息	注意事项	所在页
第20章	软件	低电压监测器	LVIM寄存器	在将LVION位设为1后，在通过使用LVIF位检测电源前需等待0.2 ms（MAX.）。	p. 570 <input type="checkbox"/>
				当LVION位= 1 且LVIMD bit = 0时，LVIF标志的值将会被输出为输出信号INTLVI。	p. 570 <input type="checkbox"/>
				确保将位2至位6清除为“0”。	p. 570 <input type="checkbox"/>
				低电压监测器不能被停止，直到由于一些不同于低电压检测的复位请求在LVIM.LVION和LVIM.LVIMD位被设为1后被生成为止。	p. 570 <input type="checkbox"/>
			LVIS寄存器	LVION和LVIMD位设为1后，此寄存器不能写，直到产生除LVI复位以外的复位请求时。	p. 571 <input type="checkbox"/>
				一定要将位 7至位 1 清除为“0”。	p. 571 <input type="checkbox"/>
			RAMS寄存器	复位后以下显示了复位后的特殊顺序。 ● 设置条件：低于检测电压的电压的检测 通过指令来设置 通过看门狗定时器溢出生成复位信号 RAM被访问时复位信号的生成 通过时钟监控器生成复位信号 ● 清除条件：以特定顺序写入0	p. 571 <input type="checkbox"/>
				用作内部复位信号	如果LVIMD 位设置为1，那么LVIM 和 LVIS 寄存器的内容不能改变，直到LVI 之外的复位请求产生。
		PEMU1寄存器	该位不会自动清除。	p. 576 <input type="checkbox"/>	
		第22章	硬件	Flash存储器	Flash存储器映射
通信模式	没有示意的引脚根据不使用引脚的操作处理（参见2.3 I/O引脚电路类型以及不使用引脚的推荐连接）。根据需要连接一个1 kΩ至10 kΩ的电阻。				pp. 585, 586 <input type="checkbox"/>
	不要将高电平输入到DRST引脚中。				pp. 585, 586 <input type="checkbox"/>
PG-FP4	按图 22-6 中所示为这些引脚布线，或在板上经下拉电阻接GND。				p. 586 <input type="checkbox"/>
	不能由Flash存储器编程器的CLK引脚提供时钟。以板上创建振荡器并提供时钟。				p. 586 <input type="checkbox"/>
FA-64GB-8EU-A	确保REGC引脚经 4.7μF（推荐值）电容器接地。				p. 587 <input type="checkbox"/>
	时钟不能由Flash存储器编程器的CLK引脚提供。在板上创建一个振荡器并从振荡器中提供时钟。				p. 587 <input type="checkbox"/>
FA-64GB-8EU-A （在CSIB0 + HS 模式中）	按下面所示连接FLMD1引脚，或通过下拉电阻将其连接到GND。				p. 589 <input type="checkbox"/>
	通过在Flash存储器写入适配器（用虚线围绕的）上创建一个振荡器来提供一个时钟。				p. 589 <input type="checkbox"/>
通讯模式的选择	当选择UARTA0时，在接收到FLMD0脉冲后，根据专用Flash存储器编程器发送的复位命令计算接收时钟。				p. 591 <input type="checkbox"/>
FLMD1引脚处理	如果VDD信号重设后，在在线写入期间立即从其他设备输入到FLMD1引脚，则隔离该信号。	p. 593 <input type="checkbox"/>			

(24/27)

章节	分类	功能	功能的详细信息	注意事项	所在页
第22章	硬件	Flash存储器	可靠的自编程 (引导交换功能)	μ PD70F3700中不支持引导交换功能, 因此只有数据块0和1是内存区域。	p. 598 <input type="checkbox"/>
			FLMD0引脚处理	复位释放后确保FLMD0 引脚电压为0 V。	p. 600 <input type="checkbox"/>
第23章	硬件	Option byte 功能	使用CA850时的试样程序	确保为该节写入6个字节。如果少于6个字节, 那么在链接器工作期间将会发生错误。 错误消息: F4112: 非法“OPTION_BYTES”节尺寸。	p. 603 <input type="checkbox"/>
第24章	软件, 硬件	片上调试功能	OCDM寄存器	外部复位后, 当DDI, DDO, DCK, 和DMS 引脚不用于片上调试引脚而作为端口引脚时, 可以进行下面的任何一项操作。 • 输入低电平到 P05/INTP2/DRST引脚。 • 设置ODCM0 位。在这种情况下, 进行如下操作。 <1> 将ODCM0 位清零。 <2> 将P05/INTP2/DRST引脚固定为低电平直到 <1> 完成。	p. 608 <input type="checkbox"/>
				DRST 引脚有片上下拉电阻。这个电阻当OCDM0 标志清0时断开。	p. 608 <input type="checkbox"/>
	软件	注意事项 (使用DCU)		如果复位信号在RUN (程序执行) 期间被输入 (从目标系统或内部复位源的复位信号中), 那么中断功能可能会发生故障。	p. 610 <input type="checkbox"/>
				如果复位信号从引脚中输入, 那么即使复位信号通过屏蔽功能被屏蔽, I/O缓冲器 (端口引脚) 也可能被复位。	p. 610 <input type="checkbox"/>
				由于在内部Flash存储器中设置的软件中断点会通过目标复位或通过由看门狗定时器2生成的内部复位被暂时设为无效的。因此, 当硬件中断或强行中断发生时, 中断点将会变成有效的, 但在这个过程中不会发生软件中断。	p. 610 <input type="checkbox"/>
				中断过程中引脚复位被屏蔽, 且CPU及外围I/O不会被复位。执行用户程序时, 一旦Flash存储器通过DMM进行重写或通过RAM监视功能进行读取, 将会立即生成引脚复位或内部复位, 此时CPU和外围I/O可能不会被正确复位。	p. 610 <input type="checkbox"/>
				满足以下条件 (a) 和 (b) 且仿真器 (IECUBE®, MINICUBE) 上的操作由于中断等原因而停止时, 看门狗定时器2不会停止且会发生复位或不可屏蔽中断。当复位发生时, 调试器将会被挂起。 (a) 为看门狗定时器2将主时钟或副时钟用作源时钟。 (b) 内部振荡时钟被停止 (RCM.RSTOP 位 = 1)。 为了避免这种情况, 应执行以下任一操作。 • 当使用仿真器时, 将内部振荡时钟用作源时钟。 • 当使用仿真器时, 不要停止内部振荡器。	p. 610 <input type="checkbox"/>
				满足以下条件 (a) 和 (b) 且仿真器 (IECUBE, MINICUBE) 上的操作由于中断等原因而停止时, 即使外围中断功能被设置为 “Break”, TMM也不会停止。 (a) INTWT, 内部振荡时钟 (fr/8) 或副时钟被选为TMM源时钟。 (b) 主时钟被停止。 为了避免这种情况, 应执行以下任一操作。 • 当使用仿真器时, 主时钟 (fxx, fxx/2, fxx/4, fxx/64, fxx/512) 被用作源时钟。 • 当使用仿真器时, 禁止主时钟振荡。	p. 610 <input type="checkbox"/>
				在片上调试模式中, DDO引脚被强行设为高电平输出。	p. 610 <input type="checkbox"/>
	硬件				

(25/27)

章节	分类	功能	功能的详细信息	注意事项	所在页	
第24章	硬件	片上调试功能	注意事项 (不使用DCU)	不要将用于调试的设备装载到大批量生产的产品上，因为Flash存储器会在调试期间被重写且Flash存储器重写的次数不能被保证。 此外，不要将调试监视程序嵌入到大批量生产的产品中。	p. 619 <input type="checkbox"/>	
				如果以下任一的条件被满足，则将不能执行强行中止。 ● 禁止中断（DI）。 ● 为串行接口发出的用于MINICUBE2及目标设备间通信的中断被屏蔽。 ● 禁止在通过可屏蔽中断释放待命时进入待机模式。 ● MINICUBE2与目标设备间的通信模式是UARTA0，且主时钟已经被停止。	p.619 <input type="checkbox"/>	
	软件			如果以下任一的条件被满足，则将不能执行伪RRM功能以及DMM功能。 ● 禁止中断（DI）。 ● 为串行接口发出的用于MINICUBE2及目标设备间通信的中断被屏蔽。 ● 禁止在通过可屏蔽中断释放待命时进入待机模式。 ● MINICUBE2与目标设备间的通信模式是UARTA0，且主时钟已经被停止。 ● MINICUBE2与目标设备间的通信模式是UARTA0，且不同于在调试器中所指定的时钟被用于通信。	p. 619 <input type="checkbox"/>	
				如果以下任一的条件被满足，那么待机模式将通过伪RMM功能和DMM功能来释放。 ● MINICUBE2与目标设备间的通信模式是CSIB0。 ● MINICUBE2与目标设备间的通信模式是UARTA0，且已经提供主时钟。	p. 619 <input type="checkbox"/>	
				需要特定写入顺序的外围I/O寄存器不能使用DMM功能来写入。	p. 619 <input type="checkbox"/>	
				第一次启动调试程序时会执行芯片擦除以及监视程序的写操作，但执行该操作大概需要十二秒。	p. 619 <input type="checkbox"/>	
				当CPU操作时钟的设置通过调试器改变时，调试器将会重写监视程序。所需要的时间与上述（6）中的相同。对于集成调试器ID850QB来说，这会在配置对话框中的Clock列的设置被改变时应用。	p. 619 <input type="checkbox"/>	
				如果调试监视程序被分配到的空间通过Flash存储器自编程进行重写，那么调试器将不能正常运行。	p. 619 <input type="checkbox"/>	
				安全ID	Flash存储器被擦除后，1将会被写入整个区域。	p. 620 <input type="checkbox"/>
				硬件	电气特性	最大绝对额定值
避免IC设备输出（或I/O）引脚间的连接以及VDD或Vcc与GND连接。	pp. 623, 624 <input type="checkbox"/>					
任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。 DC特性和AC特性中指出的额定值和条件是正常工作的质量保证。	pp. 623, 624 <input type="checkbox"/>					

(26/27)

章节	分类	功能	功能的详细信息	注意事项	所在页
第25章	硬件	电气特性	最大绝对额定值	当直接将外部电路连接到变成高阻状态的引脚时，必须这样设置时序以避免外部电路上的输出冲突。	pp. <input type="checkbox"/> 623, 624
			主时钟振荡器特性	当使用主时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。 <ul style="list-style-type: none">• 连接线越短越好。• 连接线不应与其他信号线交叉。• 流经的电流变化较大的信号线不要在振荡器周围布线。• 要保持振荡器电容器的接地点电压与Vss相同。• 不要将电容的地信号接入大电流地。• 不要从振荡器获取信号。	p. 626 <input type="checkbox"/>
			当主时钟停止而副时钟运行时，在转换回主时钟前将一直等待直到振荡稳定时间由程序保证为止。	p. 626 <input type="checkbox"/>	
		硬件	副时钟振荡器特性	当使用副时钟振荡器，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接线电容产生不利影响。 <ul style="list-style-type: none">• 连接线越短越好。• 连接线不应与其他信号线交叉。• 流经的电流变化较大的信号线不要在振荡器周围布线。• 要保持振荡器电容器的接地点电压与Vss相同。• 不要将电容的地信号接入大电流地。• 不要从振荡器获取信号。	p. 627 <input type="checkbox"/>
	为了缩减能耗，副时钟振荡器被设计为低幅电路，并且它比主时钟振荡器更容易受到噪声的影响。因此在使用副时钟时更要注意布线方法。			p. 627 <input type="checkbox"/>	
	电压调节器特性		确保RESET = Vss = 0 V时VDD上升。	p. 628 <input type="checkbox"/>	
	引脚泄漏电流		FLMDO引脚的值如下所示。 <ul style="list-style-type: none">• 输入泄漏电流，高： 2 μA （最大.）• 输入泄漏电流，低： -2 μA （最大.）	p. 630 <input type="checkbox"/>	
	软件	数据保持特性	转换到STOP模式并从 STOP模式还原必须在额定的工作范围完成。	p. 632 <input type="checkbox"/>	
		AC 特性	如果由于电路结构加载电容超过50 pF，通过插入缓冲器或其他方法将设备的加载电容降到50 pF以下。	p. 633 <input type="checkbox"/>	
		编程特性	当对出厂产品进行初次写入时，将“擦除写入”和“只写入”两个过程按一次重写来计数。 示例（P：写入， E：擦除） 出厂产品 → P → E → P → E → P： 3次重写 出厂产品 → E→P → E → P → E → P： 3次重写	p. 642 <input type="checkbox"/>	

(27/27)

章节	分类	功能	功能的详细信息	注意事项	所在页
第27	硬件	推荐焊接条件	推荐焊接条件	不要同时使用不同的焊接方式（除局部加热外）。	p. 644 <input type="checkbox"/>
附录 A	软件	开发工具	RX850, RX850 Pro	要购买RX850 或 RX850 Pro，首先需要填写购买申请表并且签署用户协议。	p. 653 <input type="checkbox"/>
附录 C	软件	指令集列表	指令集	对于通用寄存器reg1 和 reg3，不要指定相同的寄存器。	p. 670 <input type="checkbox"/>

E.1 在此版本中的主要修订

(1/2)

所在页	修订说明
p. 40	添加了 2.5 注意事项
p. 67	添加了对 3.4.7 特殊寄存器 的描述
p. 143	在 4.5.1 (b) 复用功能模式 (输入) 中添加了注意事项
p. 145	修改了图 5-1 时钟发生器
p. 146	修改了 5.2 (8) 预分频器 4
p. 152	修改了表 5-1 各时钟的操作状态
p. 156	修改了 5.5.2 (4) 可编程时钟模式寄存器 (PCLM)
p. 165	修改了 6.4 (3) TMPn I/O 控制寄存器 0 (TPnIOC0)
p. 188	在图 6-11 外部事件计数模式操作的寄存器设置中添加了注意事项
p. 209	在图 6-22 单脉冲输出模式的寄存器设置中添加了注意事项
p. 251	在 6.7 (1) 选择器操作控制寄存器 0 (SELCNT0) 中添加了注意事项 2
p. 261	在 7.4 (3) TMQ0 I/O 控制寄存器 0 (TQ0IOC0) 中添加了注
p. 287	在图 7-11 外部事件计数模式操作的寄存器设置中添加了注意事项
p. 311	在图 7-22 单脉冲输出模式的寄存器设置中添加了注意事项
p. 380	修改了 10.3 (1) 看门狗定时器模式寄存器 2 (WDTM2) 中的注意事项 3, 4
p. 382	修改了 10.3 (2) 看门狗定时器允许寄存器 (WDTE) 中的注意事项 3
p. 390	在表 11-2 转换模式设置示例中添加了注意事项
p. 395	修改了 11.4 (7) 电压不足比较阈值寄存器 (ADA0PFT) 中的说明
p. 410	修改了 11.6 (8) 读取 ADA0CRn 结果 中的说明
p. 411	添加了 11.6 (10) 待机模式
p. 411	添加了 11.6 (11) 稳定时间内改写寄存器及触发输入
p. 411	添加了对 11.6 (13) A/D 转换结果滞后特性 的说明
p. 420	添加了对 12.3 (1) UARTAn 控制寄存器 0 (UAnCTL0) 的说明
p. 421	添加了对 12.3 (4) UARTAn 选项控制寄存器 0 (UAnOPT0) 的说明
p. 425	添加了对 12.4 (1) 接收完成中断请求信号 (INTUAnR) 的说明
p. 431	在 12.5.4 SBF 接收 中添加了注意事项
p. 452	修改了 13.3 (1) CSIBn 控制寄存器 0 (CBnCTL0) 中的注意事项
p. 455	修改了 13.3 (2) CSIBn 控制寄存器 1 (CBnCTL1) 中的注意事项并在其中添加了注 1。
pp. 460 至 492	修改了 13.5 操作
p. 495	在 13.7.1 波特率产生 中添加了注意事项
p. 528	修改了 14.6.2 (4) 噪声消除控制寄存器 (NFC) 中的说明
p. 536	在 16.2 (1) 节能控制寄存器 (PSC) 中添加了注意事项 3
p. 556	修改了表 17-1 RESET 引脚输入时的硬件状态 中的注 1
p. 558	修改了表 17-2 看门狗定时器 2 复位操作时的硬件状态 中的注
pp. 560, 561	添加了 17.4 复位释放后的操作
p. 570	在 20.3 (1) 低电压监测寄存器 (LVIM) 中添加了注意事项 4

(2/2)

所在页	修订说明
p. 573	修改了图 20-2 低电压检测电路的操作时序 (LVIMD 位 = 1)
p. 580	修改了 22.2 存储器配置
pp. 581 至 583	添加了 22.3 功能概述
p. 585	修改了 22.4.2 (1) UARTA0 中的发送速率
p. 592	修改了表 22-7 Flash 存储器控制命令
p. 599	修改了图 22-17 标准自编程流程
p. 601	修改了表 22-11 使用的内部资源
p. 603	添加了对 第 23 章 选项字节功能的说明
pp. 604 至 622	修改了第 24 章 片上调试功能
p. 628	在 25.5 电压调节器特性中添加了注意事项
p. 644	添加了第 27 章 推荐焊接条件
p. 645	添加了附录 A 开发工具
p. 671	添加了附录 D 注意事项列表
p. 698	添加了附录 E 修订记录

详细信息请联系:

(中国区)

网址:

<http://www.cn.necel.com/>

<http://www.necel.com/>

[北京]

日电电子(中国)有限公司
中国北京市海淀区知春路 27 号
量子芯座 7, 8, 9, 15 层
电话: (+86)10-8235-1155
传真: (+86)10-8235-7679

[深圳]

日电电子(中国)有限公司深圳分公司
深圳市福田区益田路卓越时代广场大厦 39
楼 3901, 3902, 3909 室
电话: (+86)755-8282-9800
传真: (+86)755-8282-9899

[上海]

日电电子(中国)有限公司上海分公司
中国上海市浦东新区银城中路 200 号
中银大厦 2409-2412 和 2509-2510 室
电话: (+86)21-5888-5400
传真: (+86)21-5888-5230

[香港]

香港日电电子有限公司
香港九龙旺角太子道西 193 号新世纪广场
第 2 座 16 楼 1601-1613 室
电话: (+852)2886-9318
传真: (+852)2886-9022
2886-9044

上海恩益禧电子国际贸易有限公司
中国上海市浦东新区银城中路 200 号
中银大厦 2511-2512 室
电话: (+86)21-5888-5400
传真: (+86)21-5888-5230