

用户手册

V850ES/SG2, V850ES/SG2-H

32 位单片微控制器

硬件

V850ES/SG2:

μPD703260	μPD70F3263	μPD70F3271	μPD703283
μPD703260Y	μPD70F3263Y	μPD70F3271Y	μPD703283Y
μPD703261	μPD703270	μPD70F3273	μPD70F3281
μPD703261Y	μPD703270Y	μPD70F3273Y	μPD70F3281Y
μPD703262	μPD703271	μPD703280	μPD70F3283
μPD703262Y	μPD703271Y	μPD703280Y	μPD70F3283Y
μPD703263	μPD703272	μPD703281	
μPD703263Y	μPD703272Y	μPD703281Y	
μPD70F3261	μPD703273	μPD703282	
μPD70F3261Y	μPD703273Y	μPD703282Y	

V850ES/SG2-H:

μPD703262HY	μPD70F3263HY	μPD703273HY	μPD703282HY
μPD703263HY	μPD703272HY	μPD70F3273HY	μPD703283HY
			μPD70F3283HY

文件编号 U16541CA5V1UD00 (第五版)
出版日期 2008 年 4 月 N

[备忘录]

① 输入引脚处的电压波形

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

② 未使用的输入引脚的处理

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

③ ESD防护措施

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

④ 初始化之前的状态

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

⑤ 电源开关顺序

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

对于每个设备电源的正确开关顺序必须依据设备的规范说明分别进行判断。

⑥ 电源关闭状态下的输入信号

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

每个设备电源关闭时的信号输入必须依据设备的规范说明分别进行判断。

IECUBE 是 NEC Corporation 在日本和德国的注册商标。

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

EEPROM, IEBus 和 Inter Equipment Bus 是 NEC Electronics Corporation 的商标。

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

PC/AT 是 International Business Machines Corporation 的商标。

SPARCstation 是SPARC International, Inc的商标。

Solaris 和 SunOS 是Sun Microsystems, Inc的商标。

这些商品、技术或软件在出口时必须遵守出口国的出口管理法规。

禁止违反该国法律。

- 本档所登载的内容有效期截止至 2008 年 4 月，信息先于产品的生产周期发布。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。否则因本档所登载内容引发的错误，本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的，对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或其他人所有的专利、版权以及其它知识产权作出任何明示或默示的许可及授权。
- 本文件中的电路、软件以及相关信息仅用以说明半导体产品的运作和应用实例。用户如在设备设计中应用本文件中的电路、软件以及相关信息，应自行负责。对于用户或其他人因使用了上述电路、软件以及相关信息而引起的任何损失，本公司概不负责。
- 虽然本公司致力于提高半导体产品的质量及可靠性，但用户应同意并知晓，我们仍然无法完全消除出现产品缺陷的可能。为了最大限度地减少因本公司半导体产品故障而引起的对人身、财产造成损害（包括死亡）的危险，用户务必在其设计中采用必要的安全措施，如冗余度、防火和防故障等安全设计。
- 本公司产品质量分为：

“标准等级”、“专业等级”以及“特殊等级”三种质量等级。

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

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

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

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

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

（注）

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

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

前言

读者	本手册适用于那些希望了解 V850ES/SG2 和 V850ES/SG2-H 产品的功能并准备利用这些产品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/SG2 和 V850ES/SG2-H 结构中描述的功能。
结构	本手册被分为两部分：硬件（本手册）和体系结构（V850ES 体系结构用户手册）。

硬件	体系结构
• 引脚功能 • CPU 功能 • 片上外围功能 • Flash 存储器编程 • 电气规范	• 数据类型 • 寄存器集合 • 指令格式和指令集 • 中断和异常 • 流水线操作

如何阅读本手册 假定本手册的读者具有电气工程、逻辑电路和微控制器的基本知识。

要理解 V850ES/SG2 和 V850ES/SG2-H 的一般功能

→按照**目录**阅读本手册。

要找到已知名称的寄存器的细节

→使用**附录 C 寄存器索引**。

寄存器格式

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

要理解指令的详细功能

→参阅单独的**V850ES 体系结构用户手册**。

要了解 V850ES/SG2 和 V850ES/SG2-H 的电气规范

→见**第 32 章 电气规范**。

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“查找”中指定来搜索查询。

约定

数据重要性:	高位数字在左边，低位数字在右边	
低有效表示法:	$\overline{\text{xxx}}$ （引脚和信号名上的线）	
存储器映射地址:	高地址在上面，低地址在下面	
注:	文中用 注 标记的项目的脚注	
注意事项:	需要特别注意的信息	
备注:	增补的信息	
数字表示法:	二进制	xxxx 或 xxxxB
	十进制	xxxx
	十六进制	xxxxH
2 的幂表示		
（地址空间，存储器容量）:	K（kilo）:	$2^{10} = 1,024$
	M（mega）:	$2^{20} = 1,024^2$
	G（giga）:	$2^{30} = 1,024^3$

相关文档

本手册中指出的相关文档可能包括了最初的版本，但未注明。

V850ES/SG2, V850ES/SG2-H 相关的文档

文档名称	文档编号
V850ES 体系结构用户手册	U15943E
V850ES/SG2, V850ES/SG2-H 硬件用户手册	本手册

开发工具相关的文档

文档名称		文档编号
IE-V850ES-G1 （在线仿真器）		U16313E
IE-703288-G1-EM1 （在线仿真器选项板）		U16697E
IE-V850E1-CD-NW （PCMCIA 卡型片上调试仿真器）		U16647E
QB-V850ESSX2 （在线仿真器）		U17091E
QB-V850MINI （片上调试仿真器）		U17638E
QB-MINI2 （带编程功能的片上调试仿真器）		U18371E
CA850 Ver. 3.00 C 编译器包	操作	U17293E
	C 语言	U17291E
	汇编语言	U17292E
	连接指导	U17294E
PM+ Ver. 6.30 项目管理器		U18416E
ID850 Ver. 3.00 集成调试器	操作	U17358E
ID850QB Ver. 3.40 集成调试器	操作	U18604E
TW850 Ver. 2.00 性能分析调节工具		U17241E
SM+ 系统仿真	操作	U18601E
	用户开放接口	U18212E
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. 3.21 实时操作系统	基础	U18165E
	安装	U17421E
	技术	U13772E
	任务调试器	U17422E
AZ850 Ver. 3.30 系统性能分析器		U17423E
PG-FP4 Flash 存储器编程器		U15260E
PG-FP5 Flash 存储器编程器		U18865E

目录

第 1 章 介绍	22
1.1 概要	22
1.2 特性	26
1.3 应用领域	27
1.4 订货信息	28
1.5 引脚图（俯视图）	30
1.6 功能模块配置	35
1.6.1 内部结构框图	35
1.6.2 内部单元	36
第 2 章 引脚功能	39
2.1 引脚功能列表	39
2.2 引脚状态	49
2.3 I/O 引脚电路类型，I/O 缓冲供电电平，和不使用引脚的连接	50
2.4 注意事项	54
第 3 章 CPU 功能	55
3.1 特性	55
3.2 CPU 寄存器组	56
3.2.1 程序寄存器组	57
3.2.2 系统寄存器组	58
3.3 操作模式	64
3.3.1 指定操作模式	64
3.4 地址空间	65
3.4.1 CPU 地址空间	65
3.4.2 CPU 地址空间的饶回	66
3.4.3 存储器映射	67
3.4.4 区域	69
3.4.5 地址空间的推荐使用方法	75
3.4.6 周边 I/O 寄存器	78
3.4.7 可编程周边 I/O 寄存器	89
3.4.8 特殊寄存器	89
3.4.9 注意事项	93
第 4 章 端口功能	98
4.1 特性	98
4.2 端口的基本配置	98
4.3 端口配置	99
4.3.1 端口 0	104
4.3.2 端口 1	107
4.3.3 端口 3	108
4.3.4 端口 4	115
4.3.5 端口 5	118
4.3.6 端口 7	122

4.3.7	端口 9	124
4.3.8	端口 CM	132
4.3.9	端口 CT	134
4.3.10	端口 DH	136
4.3.11	端口 DL	138
4.4	结构图	141
4.5	当使用复用功能时的端口寄存器设置	171
4.6	注意事项	179
4.6.1	设置端口引脚的注意事项	179
4.6.2	端口 n 寄存器 (Pn) 位操作指令的注意事项	182
4.6.3	片上调试引脚的注意事项	183
4.6.4	P05/INTP2/ $\overline{\text{DRST}}$ 引脚的注意事项	183
4.6.5	当电源打开时 P10, P11, 和 P53 引脚的注意事项	183
4.6.6	磁滞特性	183
<R> 4.6.7	分离总线模式注意事项	183
第 5 章	总线控制功能	184
5.1	特征	184
5.2	总线控制引脚	185
5.2.1	当内置 ROM, 内置 RAM 或片上周边 I/O 被访问时的引脚状态	185
5.2.2	各操作模式中的引脚状态	185
5.3	存储器分块功能	186
5.4	外部总线接口模式控制功能	187
5.5	总线访问	188
5.5.1	访问所需的时钟数	188
5.5.2	总线尺寸设置功能	189
5.5.3	按总线尺寸访问	190
5.6	等待功能	197
5.6.1	可编程等待功能	197
5.6.2	外部等待功能	198
5.6.3	可编程等待和外部等待之间的关系	199
5.6.4	可编程地址等待功能	200
5.7	空闲状态插入功能	201
5.8	总线保持功能	202
5.8.1	功能概述	202
5.8.2	总线保持流程	203
5.8.3	节能模式下的操作	203
5.9	总线优先级	204
5.10	总线时序	205
第 6 章	时钟产生功能	211
6.1	概述	211
6.2	结构图	212
6.3	寄存器	214
6.4	操作	219
6.4.1	各时钟的操作	219

6.4.2	时钟输出功能	219
6.5	PLL 功能	220
6.5.1	概述	220
6.5.2	寄存器	220
6.5.3	使用方法	223
第 7 章	16 位定时器/事件计数器 P (TMP)	224
7.1	概述	224
7.2	功能	224
7.3	配置	225
7.4	寄存器	227
7.5	定时器输出操作	240
7.6	操作	241
7.6.1	间隔定时器模式 (TPnMD2 到 TPnMD0 位 = 000)	248
7.6.2	外部事件计数模式 (TPnMD2 到 TPnMD0 位 = 001)	260
7.6.3	外部触发脉冲输出模式 (TPnMD2 到 TPnMD0 位 = 010)	269
7.6.4	单脉冲输出模式 (TPnMD2 到 TPnMD0 位 = 011)	281
7.6.5	PWM 输出模式 (TPnMD2 到 TPnMD0 位 = 100)	288
7.6.6	自由运行定时器模式 (TPnMD2 到 TPnMD0 位 = 101)	297
7.6.7	脉宽测量模式 (TPnMD2 到 TPnMD0 位 = 110)	315
7.7	选择器功能	321
第 8 章	16 位定时器 / 事件计数器 Q (TMQ)	323
8.1	概述	323
8.2	功能	323
8.3	配置	324
8.4	寄存器	327
8.5	定时器输出操作	343
8.6	操作	344
8.6.1	间隔定时器模式 (TQ0MD2 至 TQ0MD0 位 = 000)	351
8.6.2	外部事件计数模式 (TQ0MD2 至 TQ0MD0 位 = 001)	362
8.6.3	外部触发脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 010)	372
8.6.4	单次脉冲输出模式 (TQ0MD2 至 TQ0MD0 位 = 011)	385
8.6.5	PWM 输出模式 (TQ0MD2 至 TQ0MD0 位 = 100)	394
8.6.6	自由运行定时器模式 (TQ0MD2 至 TQ0MD0 位 = 101)	405
8.6.7	脉宽测量模式 (TQ0MD2 至 TQ0MD0 位 = 110)	426
8.7	选择器功能	431
第 9 章	116 位间隔定时器 M (TMM)	432
9.1	概述	432
9.2	配置	433
9.3	寄存器	434
9.4	操作	435
9.4.1	间隔定时器模式	435
9.4.2	注意事项	439

第 10 章 钟表定时器功能	440
10.1 功能	440
10.2 配置	441
10.3 控制寄存器.....	443
10.4 操作	447
10.4.1 作为钟表定时器的操作	447
10.4.2 作为间隔定时器的操作	448
10.4.3 注意事项	449
第 11 章 看门狗定时器 2 的功能	450
11.1 功能	450
11.2 配置	451
11.3 寄存器	452
11.4 操作	454
第 12 章 实时输出功能（RTO）	455
12.1 功能	455
12.2 配置	456
12.3 寄存器	458
12.4 操作	460
12.5 使用.....	461
12.6 注意事项	461
第 13 章 A/D 转换器	462
13.1 概述	462
13.2 功能	462
13.3 配置	463
13.4 寄存器	466
13.5 操作	477
13.5.1 基本操作	477
13.5.2 转换操作时序	478
13.5.3 触发模式	479
13.5.4 操作模式	481
13.5.5 掉电比较模式	485
13.6 注意事项	490
13.7 A/D 转换器特征表的阅读方法.....	495
第 14 章 D/A 转换器	499
14.1 功能	499
14.2 配置	499
14.3 寄存器	500
14.4 操作	502
14.4.1 正常模式下的操作	502
14.4.2 实时输出模式下的操作	502

	14.4.3 注意事项	503
第 15 章	异步串行接口 A (UARTA)	504
15.1	UARTA 和其它串行接口的模式转换	504
15.1.1	CSIB4 和 UARTA0 模式转换	504
15.1.2	UARTA2 和 I ² C00 模式转换	505
15.1.3	UARTA1 和 I ² C02 模式转换	506
15.2	特性	507
15.3	配置	508
15.4	寄存器	510
15.5	中断请求信号	517
15.6	操作	518
15.6.1	数据格式	518
15.6.2	SBF 发送/接收格式	520
15.6.3	SBF 发送	522
15.6.4	SBF 接收	523
15.6.5	UART 发送	525
15.6.6	连续发送	526
15.6.7	UART 接收	528
15.6.8	接收错误	529
15.6.9	校验类型和操作	531
15.6.10	接收数据的噪声过滤器	532
15.7	专用波特率发生器	533
15.8	注意事项	541
第 16 章	3 线长度可变串行 I/O (CSIB)	542
16.1	CSIB 和其他接口的模式转换	542
16.1.1	CSIB4 和 UARTA0 的模式转换	542
16.1.2	CSIB0 和 I ² C01 的模式转换	543
16.2	特点	544
16.3	配置	545
16.4	寄存器	547
16.5	中断请求信号	555
16.6	操 作	556
<R>	16.6.1 单独发送模式 (主模式, 发送模式)	556
<R>	16.6.2 单独发送模式 (主模式, 接收模式)	558
	16.6.3 单独发送模式 (主模式, 发送/接收模式)	560
<R>	16.6.4 单独发送模式 (从模式, 发送模式)	562
<R>	16.6.5 单独发送模式 (从模式, 接收模式)	564
<R>	16.6.6 单独发送模式 (从模式, 发送/接收模式)	566
<R>	16.6.7 连续发送模式 (主模式, 发送模式)	568
	16.6.8 连续发送模式 (主模式, 接收模式)	570
	16.6.9 连续发送模式 (主模式, 发送/接收模式)	573
<R>	16.6.10 连续发送模式 (从模式, 发送模式)	577
	16.6.11 连续发送模式 (从模式, 接收模式)	579
	16.6.12 连续发送模式 (从模式, 发送/接收模式)	582
	16.6.13 接收错误	586

16.6.14	时钟时序	587
16.7	输出引脚	589
16.8	波特率发生器	590
16.8.1	波特率产生	591
16.9	注意事项	592
第 17 章	I²C 总线	593
17.1	I2C 总线和其它串行接口的模式切换	593
17.1.1	UARTA2 和 I ² C00 模式的模式切换	593
17.1.2	CSIB0 和 I ² C01 的模式转换	594
17.1.3	UARTA1 和 I ² C02 的模式切换	595
17.2	特点	596
17.3	配置	597
17.4	寄存器	601
17.5	I²C 总线模式功能	617
17.5.1	引脚配置	617
17.6	I²C 总线定义和控制方法	618
17.6.1	开始条件	618
17.6.2	地址	619
17.6.3	指定传输方向	620
17.6.4	ACK	621
17.6.5	停止条件	622
17.6.6	等待状态	623
17.6.7	等待状态取消方法	625
17.7	I²C 中断请求信号 (INTIICn)	626
17.7.1	主设备操作	627
17.7.2	从设备操作 (当接收从地址时 (地址匹配))	630
17.7.3	从设备操作 (当接收扩展码时)	634
17.7.4	无通信时的操作	638
17.7.5	仲裁失败操作 (仲裁失败后作为从设备操作)	639
17.7.6	仲裁失败时的操作 (仲裁失败后无通信)	641
17.8	中断请求信号 (INTIICn) 的产生时序和等待控制	648
17.9	地址匹配检测方法	649
17.10	错误检测	649
17.11	扩展码	650
17.12	仲裁	651
17.13	唤醒功能	652
17.14	通信保留	653
17.14.1	通信保留功能使能时 (IICFn.IICRSVn 位 = 0)	653
17.14.2	通信保留功能禁止时 (IICFn.IICRSVn 位 = 1)	657
17.15	注意事项	658
17.16	通信操作	659
17.16.1	在单个主系统中的主设备操作	660
17.16.2	在多主控系统中的主设备操作	661
17.16.3	从设备操作	664
17.17	数据通信的时序	668

第 18 章 IEBus 控制器	675
18.1 功 能	675
18.1.1 IEBus 总线通信协议.....	675
18.1.2 总线控制权的决定（仲裁）	676
18.1.3 通信模式	676
18.1.4 通信地址	676
18.1.5 广播通信	677
18.1.6 总线的传输格式	677
18.1.7 传输数据	687
18.1.8 位格式.....	689
18.2 配置	690
18.3 寄存器	692
18.4 IEBus 总线控制器的中断操作	722
18.4.1 中断控制模块	722
18.4.2 确定中断的实例	725
18.4.3 中断源列表	728
18.4.4 通信错误源处理列表	729
18.5 中断请求信号产生时序和主 CPU 处理	731
18.5.1 主传输.....	731
18.5.2 主接收.....	733
18.5.3 从设备传输	735
18.5.4 从接收.....	737
18.5.5 IEBus 总线控制的中断请求信号产生间隔	739
 第 19 章 CAN 控制器	 743
19.1 概要	743
19.1.1 特点	743
19.1.2 功能概要	744
19.1.3 配置	745
19.2 CAN 协议	746
19.2.1 帧的格式	746
19.2.2 帧类型.....	747
19.2.3 数据帧和远程帧的格式.....	747
19.2.4 错误帧.....	754
19.2.5 过载帧.....	755
19.3 功能	756
19.3.1 决定总线优先级	756
19.3.2 位填充.....	756
19.3.3 多重控制	756
19.3.4 多点传送	756
19.3.5 CAN SLEEP 模式 / CAN STOP 模式功能.....	757
19.3.6 错误控制功能	757
19.3.7 波特率控制功	762
19.4 与目标系统连接	766
19.5 CAN 控制器的内部寄存器	767
19.5.1 CAN 控制器的配置	767
19.5.2 寄存器访问类型	768

	19.5.3 寄存器位配置	785
	19.6 寄存器	789
	19.7 位设置 / 清除功能.....	824
	19.8 CAN 控制器初始化	826
	19.8.1 控制器初始化	826
	19.8.2 信息缓存的初始化	826
	19.8.3 信息缓存的重定义	826
	19.8.4 从初始化模式转换到操作模式	828
	19.8.5 CAN 模块的复位误差计数器	828
	19.9 信息接收	829
<R>	19.9.1 信息接收	829
	19.9.2 读取接收数据	830
	19.9.3 接收历史列表功能	831
	19.9.4 屏蔽功能	833
	19.9.5 复合缓存接收块功能	835
	19.9.6 远程帧接收	836
	19.10 信息发送	837
	19.10.1 信息发送	837
	19.10.2 发送历史列表功能	839
	19.10.3 自动块发送 (ABT)	841
	19.10.4 发送中止进程	842
	19.10.5 远程帧发送	843
	19.11 节能模式	844
	19.11.1 CAN 休眠模式	844
	19.11.2 CAN 停止模式	846
	19.11.3 使用节能模式举例	847
	19.12 中断功能	848
	19.13 诊断功能和特殊的操作模式	849
	19.13.1 只接收模式	849
	19.13.2 单脉冲模式	850
	19.13.3 自测试模式	851
<R>	19.13.4 在每种操作模式下的发送 / 接收操作	852
	19.14 时间戳功能.....	853
	19.14.1 时间戳功能	853
	19.15 波特率设定.....	855
	19.15.1 位速率设定条件	855
	19.15.2 波特率设定的典型实例	859
	19.16 CAN 控制器的操作	863
	 第 20 章 DMA 功能 (DMA 控制器)	 889
	20.1 特点	889
	20.2 配置	890
	20.3 寄存器	891
	20.4 传送目标	899
	20.5 传送模式	899
	20.6 传送类型	900
	20.7 DMA 通道优先级.....	901

20.8	DMA 传送的相关时间	901
20.9	DMA 传送开始的因素	902
20.10	DMA 中止因素	903
20.11	DMA 传送结束	903
20.12	操作时序	903
20.13	注意事项	908
第 21 章	CRC 功能	912
21.1	功能	912
21.2	配置	912
21.3	寄存器	913
21.4	操作	914
21.5	用途	915
第 22 章	中断/ 异常处理功能	917
22.1	特点	917
22.2	不可屏蔽中断	921
22.2.1	操作	923
22.2.2	恢复	924
22.2.3	NP 标志	925
22.3	可屏蔽中断	926
22.3.1	操作	926
22.3.2	恢复	928
22.3.3	可屏蔽中断的优先级	929
22.3.4	中断控制寄存器 (xxICn)	933
22.3.5	中断屏蔽寄存器 0 至 3 (IMR0 至 IMR3)	936
22.3.6	正在进行服务优先级寄存器 (ISPR)	938
22.3.7	ID 标志	939
22.3.8	看门狗定时器模式寄存器 2 (WDTM2)	939
22.4	软件异常	940
22.4.1	操作	940
22.4.2	恢复	941
22.4.3	EP 标志	942
22.5	异常陷阱	943
22.5.1	非法操作码定义	943
22.5.2	调试陷阱	945
22.6	外部中断请求输入引脚 (NMI 以及 INTP0 至 INTP7)	947
22.6.1	噪声消除	947
22.6.2	边沿检测	947
22.7	CPU 的中断确认时间	952
22.8	CPU 不对中断进行确认的时期	955
22.9	注意事项	955
第 23 章	按键中断功能	956
23.1	功能	956
23.2	寄存器	957

23.3	注意事项	957
第 24 章	待机功能	958
24.1	概述	958
24.2	寄存器	960
24.3	HALT 模式	963
24.3.1	设置和操作状态	963
24.3.2	释放 HALT 模式	963
24.4	IDLE1 模式	965
24.4.1	设置和操作状态	965
24.4.2	释放 IDLE1 模式	966
24.5	IDLE2 模式	968
24.5.1	设置和操作状态	968
24.5.2	释放 IDLE2 模式	968
24.5.3	当释放 IDLE2 模式时保证设置时间	970
24.6	STOP 模式	971
24.6.1	设置和操作状态	971
24.6.2	释放 STOP 模式	971
24.6.3	当释放 STOP 模式时保证振荡稳定时间	974
24.7	子时钟操作模式	975
24.7.1	设置和操作状态	975
24.7.2	释放子时钟操作模式	975
24.8	副 IDLE 模式	977
24.8.1	设置和操作状态	977
24.8.2	释放副 IDLE 模式	978
第 25 章	复位功能	980
25.1	概述	980
25.2	检测复位源的寄存器	982
25.3	操作	983
25.3.1	通过 $\overline{\text{RESET}}$ 引脚的复位操作	983
25.3.2	通过看门狗定时器 2 执行的复位操作 (WDT2RES)	985
25.3.3	通过低压检测器执行的复位操作 (LVIRES) (仅限 V850ES/SG2)	987
25.3.4	时钟监控器引起的复位操作 (CLMRES)	988
25.3.5	复位释放后的操作	990
25.3.6	复位功能操作流程	992
25.4	内部 RAM 数据的有效/无效	993
第 26 章	时钟监视器	994
26.1	功能	994
26.2	配置	994
26.3	寄存器	995
26.4	操作	996
第 27 章	低电压检测器	999

27.1	功能	999
27.2	配置	999
27.3	寄存器	1000
27.4	操作	1002
27.4.1	用作内部复位信号 (LVIREs)	1002
27.4.2	用作中断 (INTLVI)	1003
27.5	RAM 保持电压检测操作 (为 V850ES/SG2 和 V850ES/SG2-H 提供)	1004
27.6	仿真功能 (为 V850ES/SG2 和 V850ES/SG2-H 所提供)	1005
第 28 章	调节器	1006
28.1	概要	1006
28.2	操作	1007
第 29 章	ROM 修正功能	1008
29.1	概述	1008
29.2	寄存器	1009
29.3	ROM 修正操作和程序流程	1011
29.4	注意事项	1013
第 30 章	FLASH 存储器	1014
30.1	特点	1014
30.2	存储器配置	1015
30.3	功能概述	1018
30.4	通过专用 Flash 编程器重写	1021
30.4.1	编程环境	1021
30.4.2	通信模式	1022
30.4.3	Flash 存储器控制	1031
30.4.4	通讯模式的选择	1032
30.4.5	通信命令	1033
30.4.6	引脚连接	1034
30.5	通过自编程重写	1038
30.5.1	概述	1038
30.5.2	特点	1039
30.5.3	标准自编程流程	1040
30.5.4	Flash 存储器功能	1041
30.5.5	引脚处理	1041
30.5.6	使用的内部资源	1042
第 31 章	片上调试功能	1043
31.1	特性	1043
31.2	连接电路示例	1044
31.3	接口信号	1044
31.4	寄存器	1046
31.5	操作 1048	
31.6	ROM 安全功能	1049
31.6.1	安全 ID	1049

31.6.2	设置	1050
31.7	注意事项	1052
第 32	电气规范.....	1053
32.1	绝对最大额定值	1053
32.2	电容	1055
32.3	工作条件	1055
32.4	振荡器特性.....	1056
32.4.1	主时钟振荡器特性	1056
32.4.2	子时钟振荡器特性	1059
32.4.3	PLL 特性	1060
32.4.4	内部振荡器特性	1060
32.5	调节器特性.....	1061
32.6	DC 特性	1062
32.6.1	I/O 电平	1062
32.6.2	源电流	1064
32.7	数据保持特性	1066
32.8	AC 特性	1067
32.8.1	CLKOUT 输出顺序	1068
32.8.2	总线时序	1068
32.9	基本操作	1082
32.10	Flash 存储器编程特性	1091
第 33	章 封装图.....	1094
第 34	章 推荐焊接条件.....	1096
附录 A	开发工具	1098
A.1	软件包	1103
A.2	语言处理软件	1103
A.3	控制软件	1103
A.4	调试工具（硬件）	1104
A.4.1	当使用在线方针器 IE-V850ES-G1 时	1104
A.4.2	当使用 ECUBE QB-V850ESSX2 时	1106
<R> A.4.3	当使用片上调试仿真器 IE-V850E1-CD-NW 时	1109
<R> A.4.4	当使用 MINICUBE® QB-V850MINI 时.....	1110
A.5	调试工具（软件）	1111
A.6	嵌入式软件.....	1112
A.7	flash 存储器写入工具	1112
<R> 附录 B	V850ES/SG2 和 V850ES/SG2-H 之间的主要差别	1113
附录 C	寄存器索引.....	1115

附录 D 指令集列表	1127
D.1 常规指令	1127
D.2 指令集（按字母顺序）	1130
附录 E 再版修订记录	1137
E.1 在此版本中的主要修订	1137
E.2 先前版本的修订记录	1147

第 1 章 介绍

<R> V850ES/SG2 和 V850ES/SG2-H 是日电电子公司（日电电子）的 V850 系列单片微控制器产品，适用于实时控制应用中的低功耗操作。

<R> 1.1 概要

V850ES/SG2 和 V850ES/SG2-H 是 32 位单片微控制器，它包含了 V850ES CPU 内核以及外围设备功能，诸如 ROM/RAM、定时器/计数器、串行接口、A/D 转换器和 D/A 转换器等。V850ES/SG2 和 V850ES/SG2-H 系列中的某些产品拥有 IEBusTM（内部设备总线TM）或者 CAN 总线可以用作汽车的网络。

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/SG2 还具有硬件乘法器来实现的乘法指令、饱和运算指令、位操作指令等为数字伺服控制应用而优化的指令。此外在实时控制系统中，V850ES/SG2 可以以极高的性能-功耗比实现那些需要低功耗的应用，比如音频系统和车载音频系统。

表 1-1 列出了 V850ES/SG2 和 V850ES/SG2-H 的产品。

还有 V850ES/SJ2 和 V850ES/SJ2-H 系列产品，在 V850ES/SG2 和 V850ES/SG2-H 基础上并带有扩展 I/O，定时器/计数器，串行接口等功能模块。见表 1-2 V850ES/SJ2，V850ES/SJ2-H 产品列表。

<R>

表 1-1 V850ES/SJ2, V850ES/SJ2-H 产品列表。

功能 产品编号	ROM		RAM 容量	工作频率 (最高)	I ² C	IEBus 总线	CAN	可屏蔽中断		不可屏蔽中 断:
	类型	容量						外部	内部	
μPD703260	掩模 ROM	256 KB	24 KB	20 MHz	无	无	无	8	47	2
μPD703260Y					片上					
μPD703261		384 KB	32 KB		无					
μPD703261Y					片上					
μPD70F3261	Flash 存储器				无					
μPD70F3261Y					片上					
μPD703262	掩模 ROM	512 KB	40 KB		无					
μPD703262Y					片上					
μPD703262HY				32 MHz	46					
μPD703263		640 KB	48 KB	20 MHz	无				47	
μPD703263Y				片上						
μPD703263HY				32 MHz	46					
μPD70F3263	Flash 存储器			20 MHz	无				47	
μPD70F3263Y				片上						
μPD70F3263HY				32 MHz	46					
μPD703270	掩模 ROM	256 KB	24 KB	20 MHz	无	片上		51		
μPD703270Y					片上					
μPD703271		384 KB	32 KB		无					
μPD703271Y					片上					
μPD70F3271	Flash 存储器				无					
μPD70F3271Y					片上					
μPD703272	掩模 ROM	512 KB	40 KB		无					
μPD703272Y					片上					
μPD703272HY				32 MHz	50					
μPD703273		640 KB	48 KB	20 MHz	无				51	
μPD703273Y				片上						
μPD703273HY				32 MHz	50					
μPD70F3273	Flash 存储器			20 MHz	无				51	
μPD70F3273Y				片上						
μPD70F3273HY				32 MHz	50					
μPD703280	掩模 ROM	256 KB	24 KB	20 MHz	无	无	片上	51		
μPD703280Y					片上					
μPD703281		384 KB	32 KB		无					
μPD703281Y					片上					
μPD70F3281	Flash 存储器				无					
μPD70F3281Y					片上					
μPD703282	掩模 ROM	512 KB	40 KB		无					
μPD703282Y					片上					
μPD703282HY				32 MHz	50					
μPD703283		640 KB	48 KB	20 MHz	无				51	
μPD703283Y				片上						
μPD703283HY				32 MHz	50					
μPD70F3283	Flash 存储器			20 MHz	无				51	
μPD70F3283Y				片上						
μPD70F3283HY				32 MHz	50					

备注 本手册中使用的 V850ES/SG2 和 V850ES/SG2-H 系列产品的产品代号如下所述。

- <R> • V850ES/SG2
 μ PD703260, 703260Y, 703261, 703261Y, 703262, 703262Y, 703263, 703263Y, 703270, 703270Y, 703271, 703271Y, 703272, 703272Y, 703273, 703273Y, 703280, 703280Y, 703281, 703281Y, 703282, 703282Y, 703283, 703283Y, 70F3261, 70F3261Y, 70F3263, 70F3263Y, 70F3271, 70F3271Y, 70F3273, 70F3273Y, 70F3281, 70F3281Y, 70F3283, 70F3283Y
- <R> • V850ES/SG2-H
 μ PD703262HY, 703263HY, 703272HY, 703273HY, 703282HY, 703283HY, 70F3263HY, 70F3273HY,
- <R> 70F3283HY
- 掩模 ROM
 μ PD703260, 703260Y, 703261, 703261Y, 703262, 703262Y, 703262HY, 703263, 703263Y, 703263HY, 703270, 703270Y, 703271, 703271Y, 703272, 703272Y, 703272HY, 703273, 703273Y, 703273HY, 703280, 703280Y, 703281, 703281Y, 703282, 703282Y, 703282HY, 703283, 703283Y, 703283HY
- <R> • Flash 存储器版本
 μ PD70F3261, 70F3261Y, 70F3263, 70F3263Y, 70F3263HY, 70F3271, 70F3271Y, 70F3273, 70F3273Y, 70F3273HY, 70F3281, 70F3281Y, 70F3283, 70F3283Y, 70F3283HY
- <R> • I²C 总线版本 (Y 版本) 所有 V850ES/SG2-H 产品都有一条片上 I²C 总线。
 μ PD703260Y, 703261Y, 703262Y, 703262HY, 703263Y, 703263HY, 703270Y, 703271Y, 703272Y, 703272HY, 703273Y, 703273HY, 703280Y, 703281Y, 703282Y, 703282HY, 703283Y, 703283HY, 70F3261Y, 70F3263Y, 70F3263HY, 70F3271Y, 70F3273Y, 70F3273HY, 70F3281Y, 70F3283Y, 70F3283HY
- <R> • 通用寄存器版本
 μ PD703260, 703260Y, 703261, 703261Y, 703262, 703262Y, 703262HY, 703263, 703263Y, 703263HY, 70F3261, 70F3261Y, 70F3263, 70F3263Y, 70F3263HY
- <R> • IEBus 总线控制器版本
 μ PD703270, 703270Y, 703271, 703271Y, 703272, 703272Y, 703272HY, 703273, 703273Y, 703273HY, 70F3271, 70F3271Y, 70F3273, 70F3273Y, 70F3273HY
- <R> • CAN 总线控制器版本
 μ PD703280, 703280Y, 703281, 703281Y, 703282, 703282Y, 703282HY, 703283, 703283Y, 703283HY, 70F3281, 70F3281Y, 70F3283, 70F3283Y, 70F3283HY

<R>

表 1-2 V850ES/SJ2, V850ES/SJ2-H 产品列表

产品代号\功能	ROM		RAM 容量	工作频率 (最高)	I ² C	IEBus 总线	CAN	可屏蔽中断		不可屏蔽中 断:
	类型	容量						外部	内部	
μPD703264	掩模 ROM	384 KB	32 KB	20 MHz	无	无	无	9	60	2
μPD703264Y					片上					
μPD70F3264	flash 存储器				无					
μPD70F3264Y					片上					
μPD703265	掩模 ROM	512 KB	40 KB	20 MHz	无					
μPD703265Y					片上					
μPD703265HY					32 MHz					
μPD703266					640 KB					
μPD703266Y	片上									
μPD703266HY	32 MHz									
μPD70F3266	flash 存储器	20 MHz								
μPD70F3266Y		片上								
μPD70F3266HY		32 MHz								
μPD703274	掩模 ROM	384 KB	32 KB	20 MHz	无	片上			64	
μPD703274Y					片上					
μPD70F3274	flash 存储器				无					
μPD70F3274Y					片上					
μPD703275	掩模 ROM	512 KB	40 KB	20 MHz	无					
μPD703275Y					片上					
μPD703275HY					32 MHz					
μPD703276					640 KB					48 KB
μPD703276Y	片上									
μPD703276HY	32 MHz									
μPD70F3276	flash 存储器	20 MHz								
μPD70F3276Y		片上								
μPD70F3276HY		32 MHz								
μPD703284	掩模 ROM	384 KB	32 KB	20 MHz	无	无	1 通道		64	
μPD703284Y					片上					
μPD70F3284	flash 存储器				无					
μPD70F3284Y					片上					
μPD703285	掩模 ROM	512 KB	40 KB	20 MHz	无					
μPD703285Y					片上					
μPD703285HY					32 MHz					
μPD703286					640 KB					48 KB
μPD703286Y	片上									
μPD703286HY	32 MHz									
μPD70F3286	flash 存储器	20 MHz								
μPD70F3286Y		片上								
μPD70F3286HY		32 MHz								
μPD703287	掩模 ROM	512 KB	40 KB	20 MHz	无	2 ch			68	
μPD703287Y					片上					
μPD703287HY					32 MHz					
μPD703288					640 KB					48 KB
μPD703288Y	片上									
μPD703288HY	32 MHz									
μPD70F3288	flash 存储器	20 MHz								
μPD70F3288Y		片上								
μPD70F3286HY		32 MHz								

1.2 特性

	○ 最小指令执行时间:	V850ES/SG2 50 ns (工作在 20 MHz 主时钟 (f _{xx}) 下)
<R>		V850ES/SG2 -H 31.25 ns (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 到 2 时钟 带符号乘法 (32 × 32 → 64): 1 到 5 时钟 饱和操作 (包含上溢和下溢的检测功能) 32-位移位指令: 1 时钟周期 位操作指令 区分长短格式的读取/存储指令
<R>	○ 存储器空间:	为程序和数据提供 64 MB 的线性地址空间 外部扩展: 多达 4 MB
	• 内部存储器:	RAM: 24/32/40/48 KB (见 表 1-1) 掩模 ROM: 256/384/512/640 KB (见 表 1-1) flash 存储器: 384/640 KB (见 表 1-1)
	• 外部总线接口:	分离总线/复用总线输出选择: 可选择 8/16 位数据总线宽度 等待功能 • 可编程等待功能 • 外部等待功能 空闲状态功能 总线保持功能
<R>	○ 中断与异常:	不可屏蔽中断: 2 个 可屏蔽中断: 54/55/58/59 个中断源 (见 表 1-1) 软件异常: 32 个 异常陷阱: 2 个
	○ I/O 线:	I/O 端口: 84
	○ 定时器功能:	16-位间隔定时器 M (TMM): 1 通道 16-位 定时器/事件计数器 P (TMP): 6 通道 16-位 定时器/事件计数器 Q (TMQ): 1 通道 钟表定时器: 1 通道 看门狗定时器: 1 通道
	○ 实时输出端口:	6 位 × 1 通道
	○ 串行接口:	异步串行接口 A (UARTA) 3-线可变长度串行接口 B (CSIB) I ² C 总线接口 (I ² C) (只适用于 I ² C 总线模式 (Y 产品)) UARTA/CSIB: 1 通道 UARTA/I ² C: 2 通道 CSIB/I ² C: 1 通道 CSIB: 3 通道
	○ IEBus 控制器:	1 通道 (IEBus 总线控制器)
	○ CAN 控制器:	1 通道 (CAN 总线控制器)
	○ A/D 转换器:	10 位: 12 通道
	○ D/A 转换器:	8 位: 2 通道
	○ DMA 控制器	4 通道
	○ CRC 功能:	为 8 位数据生成 16 位错误检测码

- <R>
- 片上调试功能:

○ ROM 修正:

○ 时钟产生器

○ 内置振荡时钟:

○ 节能模式:

○ 封装

JTAG 接口 (flash 存储器)

可指定 4 个修正地址

在主时钟或副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})

时钟直通模式/PLL 模式可选

200 kHz (典型值)

HALT/IDLE1/IDLE2/STOP/副时钟/sub-IDLE 模式

100-pin 塑封 QFP (14 × 20) (仅适用于 V850ES/SG2 (见 1.4 订货信息))

100-pin 塑封 LQFP (细间距) (14 × 14)

1.3 应用领域

音响, 汽车音响, 其它消费电子设备

<R> 1.4 订货信息

(1) V850ES/SG2

产品代号	封装	内置 ROM
μ PD703260GC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	256 KB (掩模 ROM)
μ PD703260GF-xxx-JBT-A	100-pin 塑封 QFP (14 × 20)	256 KB (掩模 ROM)
μ PD703260YGC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	256 KB (掩模 ROM)
μ PD703260YGF-xxx-JBT-A	100-pin 塑封 QFP (14 × 20)	256 KB (掩模 ROM)
μ PD703261GC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	384 KB (掩模 ROM)
μ PD703261GF-xxx-JBT-A	100-pin 塑封 QFP (14 × 20)	384 KB (掩模 ROM)
μ PD703261YGC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	384 KB (掩模 ROM)
μ PD703261YGF-xxx-JBT-A	100-pin 塑封 QFP (14 × 20)	384 KB (掩模 ROM)
μ PD703262GC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	512 KB (掩模 ROM)
μ PD703262YGC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	512 KB (掩模 ROM)
μ PD703263GC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	640 KB (掩模 ROM)
μ PD703263YGC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	640 KB (掩模 ROM)
μ PD703270GC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	256 KB (掩模 ROM)
μ PD703270GF-xxx-JBT-A	100-pin 塑封 QFP (14 × 20)	256 KB (掩模 ROM)
μ PD703270YGC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	256 KB (掩模 ROM)
μ PD703270YGF-xxx-JBT-A	100-pin 塑封 QFP (14 × 20)	256 KB (掩模 ROM)
μ PD703271GC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	384 KB (掩模 ROM)
μ PD703271GF-xxx-JBT-A	100-pin 塑封 QFP (14 × 20)	384 KB (掩模 ROM)
μ PD703271YGC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	384 KB (掩模 ROM)
μ PD703271YGF-xxx-JBT-A	100-pin 塑封 QFP (14 × 20)	384 KB (掩模 ROM)
μ PD703272GC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	512 KB (掩模 ROM)
μ PD703272YGC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	512 KB (掩模 ROM)
μ PD703273GC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	640 KB (掩模 ROM)
μ PD703273YGC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	640 KB (掩模 ROM)
μ PD703280GC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	256 KB (掩模 ROM)
μ PD703280YGC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	256 KB (掩模 ROM)
μ PD703281GC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	384 KB (掩模 ROM)
μ PD703281YGC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	384 KB (掩模 ROM)
μ PD703282GC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	512 KB (掩模 ROM)
μ PD703282YGC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	512 KB (掩模 ROM)
μ PD703283GC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	640 KB (掩模 ROM)
μ PD703283YGC-xxx-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	640 KB (掩模 ROM)

- 备注
1. Xxx 表示 ROM 代码的后缀。
 2. V850ES/SG2 微控制器是无铅产品。

产品代号	封装	内置 ROM
μ PD70F3261GC-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	384 KB (flash 存储器)
μ PD70F3261GF-JBT-A	100-pin 塑封 QFP (14 × 20)	384 KB (flash 存储器)
μ PD70F3261YGC-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	384 KB (flash 存储器)
μ PD70F3261YGF-JBT-A	100-pin 塑封 QFP (14 × 20)	384 KB (flash 存储器)
μ PD70F3263GC-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	640 KB (flash 存储器)
μ PD70F3263YGC-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	640 KB (flash 存储器)
μ PD70F3271GC-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	384 KB (flash 存储器)
μ PD70F3271GF-JBT-A	100-pin 塑封 QFP (14 × 20)	384 KB (flash 存储器)
μ PD70F3271YGC-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	384 KB (flash 存储器)
μ PD70F3271YGF-JBT-A	100-pin 塑封 QFP (14 × 20)	384 KB (flash 存储器)
μ PD70F3273GC-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	640 KB (flash 存储器)
μ PD70F3273YGC-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	640 KB (flash 存储器)
μ PD70F3281GC-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	384 KB (flash 存储器)
μ PD70F3281YGC-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	384 KB (flash 存储器)
μ PD70F3283GC-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	640 KB (flash 存储器)
μ PD70F3283YGC-8EA-A	100-pin 塑封 LQFP (细间距) (14 × 14)	640 KB (flash 存储器)

备注 V850ES/SG2 微控制器是无铅产品。

(2) V850ES/SG2-H

产品代号	封装	内置 ROM
μ PD703262HYGC-xxx-8EA-A	100-pin 塑封 QFP (14 × 20)	512 KB (掩模 ROM)
μ PD703263HYGC-xxx-8EA-A	100-pin 塑封 QFP (14 × 20)	640 KB (掩模 ROM)
μ PD703272HYGC-xxx-8EA-A	100-pin 塑封 QFP (14 × 20)	512 KB (掩模 ROM)
μ PD703273HYGC-xxx-8EA-A	100-pin 塑封 QFP (14 × 20)	640 KB (掩模 ROM)
μ PD703282HYGC-xxx-8EA-A	100-pin 塑封 QFP (14 × 20)	512 KB (掩模 ROM)
μ PD703283HYGC-xxx-8EA-A	100-pin 塑封 QFP (14 × 20)	640 KB (掩模 ROM)
μ PD70F3263HYGC-8EA-A	100-pin 塑封 QFP (14 × 20)	640 KB (flash 存储器)
μ PD70F3273HYGC-8EA-A	100-pin 塑封 QFP (14 × 20)	640 KB (flash 存储器)
μ PD70F3283HYGC-8EA-A	100-pin 塑封 QFP (14 × 20)	640 KB (flash 存储器)

备注 1. Xxx 表示 ROM 代码的后缀。
2. V850ES/SG2 微控制器是无铅产品。

1.5 引脚图（俯视图）

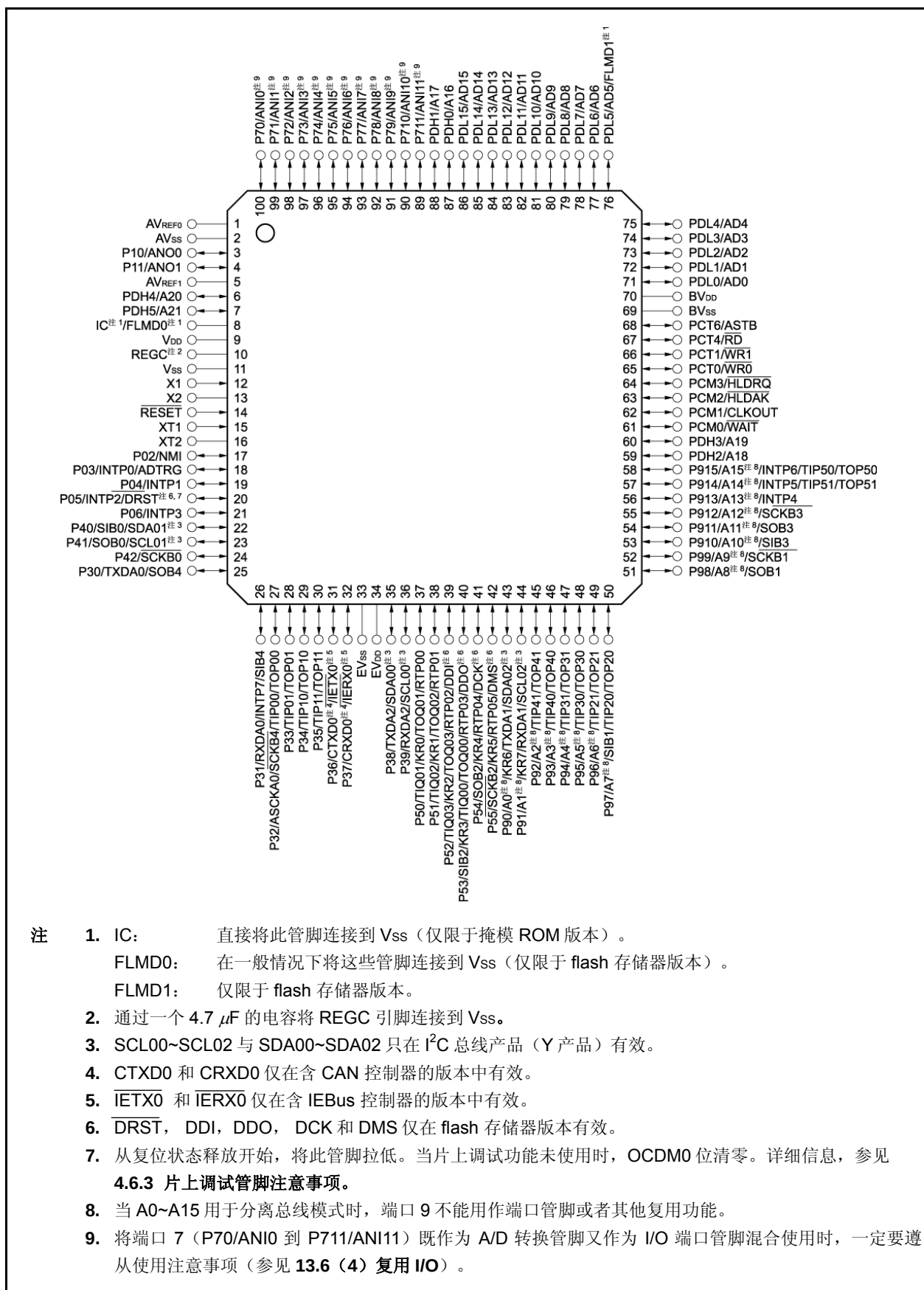
100-pin 塑封 LQFP（细间距）（14 × 14）

• V850ES/SG2

<R>	μPD703260GC-xxx-8EA-A	μPD703270GC-xxx-8EA-A	μPD703280GC-xxx-8EA-A
<R>	μPD703260YGC-xxx-8EA-A	μPD703270YGC-xxx-8EA-A	μPD703280YGC-xxx-8EA-A
<R>	μPD703261GC-xxx-8EA-A	μPD703271GC-xxx-8EA-A	μPD703281GC-xxx-8EA-A
<R>	μPD703261YGC-xxx-8EA-A	μPD703271YGC-xxx-8EA-A	μPD703281YGC-xxx-8EA-A
<R>	μPD703262GC-xxx-8EA-A	μPD703272GC-xxx-8EA-A	μPD703282GC-xxx-8EA-A
<R>	μPD703262YGC-xxx-8EA-A	μPD703272YGC-xxx-8EA-A	μPD703282YGC-xxx-8EA-A
<R>	μPD703263GC-xxx-8EA-A	μPD703273GC-xxx-8EA-A	μPD703283GC-xxx-8EA-A
<R>	μPD703263YGC-xxx-8EA-A	μPD703273YGC-xxx-8EA-A	μPD703283YGC-xxx-8EA-A
<R>	μPD70F3261GC-8EA-A	μPD70F3271GC-8EA-A	μPD70F3281GC-8EA-A
<R>	μPD70F3261YGC-8EA-A	μPD70F3271YGC-8EA-A	μPD70F3281YGC-8EA-A
<R>	μPD70F3263GC-8EA-A	μPD70F3273GC-8EA-A	μPD70F3283GC-8EA-A
<R>	μPD70F3263YGC-8EA-A	μPD70F3273YGC-8EA-A	μPD70F3283YGC-8EA-A

• V850ES/SG2-H

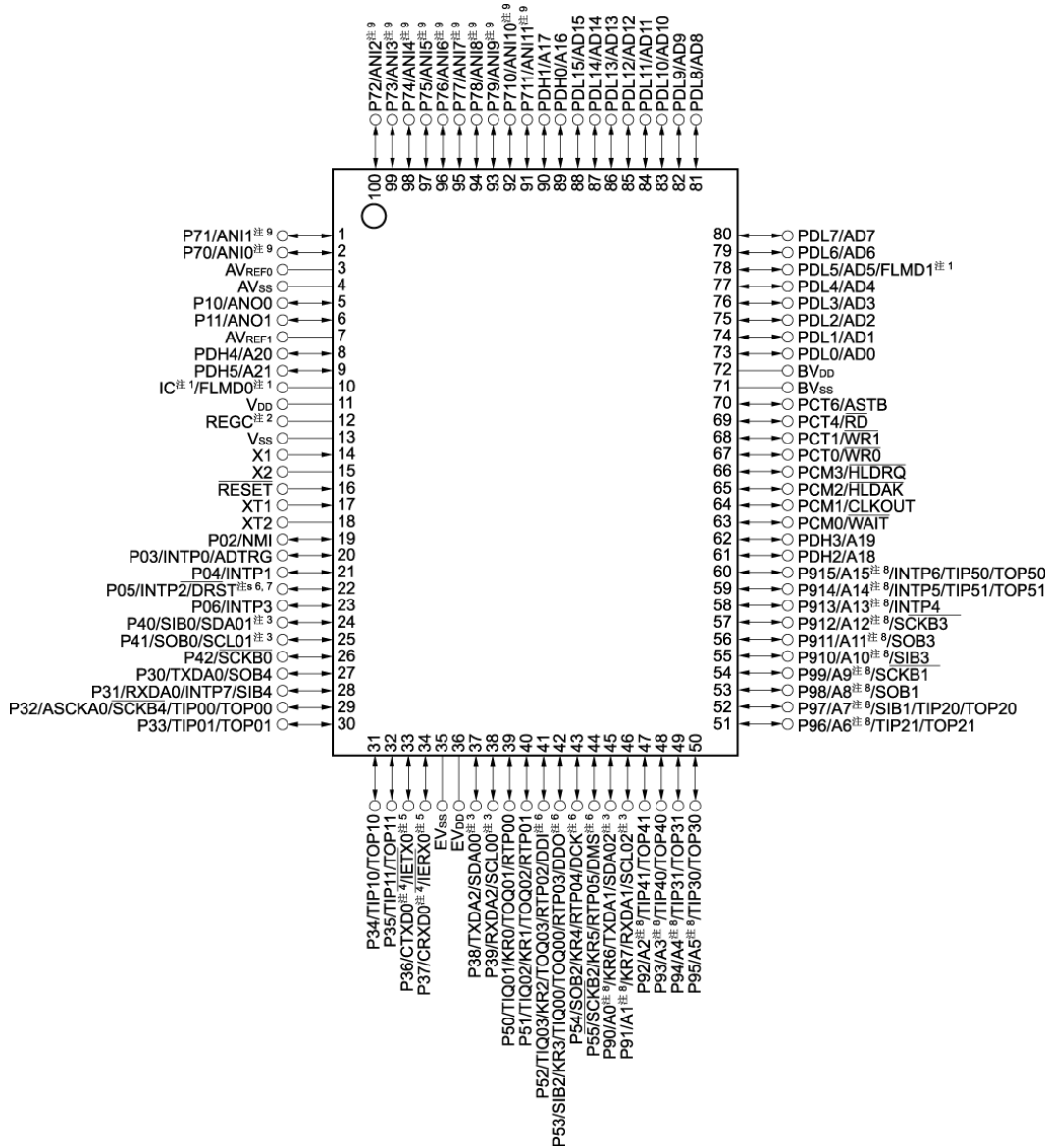
<R>	μPD703262HYGC-xxx-8EA-A	μPD703272HYGC-xxx-8EA-A	μPD703282HYGC-xxx-8EA-A
<R>	μPD703263HYGC-xxx-8EA-A	μPD703273HYGC-xxx-8EA-A	μPD703283HYGC-xxx-8EA-A
<R>	μPD70F3263HYGC-8EA-A	μPD70F3273HYGC-8EA-A	μPD70F3283HYGC-8EA-A



- 注 1. IC: 直接将此管脚连接到 Vss (仅限于掩模 ROM 版本)。
 FLMD0: 在一般情况下将这些管脚连接到 Vss (仅限于 flash 存储器版本)。
 FLMD1: 仅限于 flash 存储器版本。
2. 通过一个 4.7 μ F 的电容器将 REGC 引脚连接到 Vss。
3. SCL00~SCL02 与 SDA00~SDA02 只在 I²C 总线产品 (Y 产品) 有效。
4. CTXD0 和 CRXD0 仅在含 CAN 控制器的版本中有效。
5. IETX0 和 IERX0 仅在含 IEBus 控制器的版本中有效。
6. DRST, DDI, DDO, DCK 和 DMS 仅在 flash 存储器版本有效。
7. 从复位状态释放开始, 将此管脚拉低。当片上调试功能未使用时, OCDM0 位清零。详细信息, 参见 4.6.3 片上调试管脚注意事项。
8. 当 A0~A15 用于分离总线模式时, 端口 9 不能用作端口管脚或者其他复用功能。
9. 将端口 7 (P70/ANI0 到 P711/ANI11) 既作为 A/D 转换管脚又作为 I/O 端口管脚混合使用时, 一定要遵从使用注意事项 (参见 13.6 (4) 复用 I/O)。

100-pin 塑封 QFP (14 × 20)

<R>	μPD703260GF-xxx-JBT-A	μPD703270GF-xxx-JBT-A	μPD70F3261GF-JBT-A
<R>	μPD703260YGF-xxx-JBT-A	μPD703270YGF-xxx-JBT-A	μPD70F3261YGF-JBT-A
<R>	μPD703261GF-xxx-JBT-A	μPD703271GF-xxx-JBT-A	μPD70F3271GF-JBT-A
<R>	μPD703261YGF-xxx-JBT-A	μPD703271YGF-xxx-JBT-A	μPD70F3271YGF-JBT-A



引脚名称

A0 到 A21:	地址线	PCM0 到 PCM3:	端口 CM
AD0 到 AD15:	地址 / 数据线	PCT0, PCT1,	
ADTRG:	A/D 触发输入	PCT4, PCT6:	端口 CT
ANI0 到 ANI11:	模拟输入	PDH0 到 PDH5:	端口 DH
ANO0、ANO1:	模拟输出	PDL0 到 PDL15:	端口 DL
ASCKA0:	异步串行时钟	$\overline{RD}$:	读探针
ASTB:	地址探针	REGC:	电阻控制
AVREF0、AVREF1:	模拟参考电压	$\overline{RESET}$:	复位
AVss:	模拟 Vss	RTP00 到 RTP05:	实时输出端口
BVDD:	总线接口供电电压	RXDA0 到 RXDA2:	接收数据
BVss:	总线接口的地	$\overline{SCKB0}$ 到 $\overline{SCKB4}$:	串行时钟
CLKOUT:	时钟输出	SCL00 到 SCL02:	串行时钟
CRXD0:	CAN 接收数据	SDA00 到 SDA02:	串行数据
CTXD0:	CAN 发送数据	SIB0 到 SIB4:	串行输入
DCK:	调试时钟	SOB0 到 SOB4:	串行输出
DDI:	调试数据输入	TIP00、TIP01,	
DDO:	调试数据输出	TIP10、TIP11,	
DMS:	调试模式选择	TIP20、TIP21,	
$\overline{DRST}$:	调试模式复位	TIP30、TIP31,	
EVDD:	端口供电电压	TIP40、TIP41,	
EVss:	端口的地	TIP50、TIP51,	
FLMD0、FLMD1:	flash 存储器编程模式	TIQ00 到 TIQ03:	定时器输入
HLD $\overline{AK}$:	保持确认	TOP00、TOP01,	
HLD $\overline{RQ}$:	保持请求	TOP10、TOP11,	
IC:	内部连接	TOP20、TOP21,	
$\overline{IERX0}$:	IEBus 接收数据	TOP30、TOP31,	
$\overline{IETX0}$:	IEBus 发送数据	TOP40、TOP41,	
INTP0 到 INTP7:	外部中断输入	TOP50、TOP51,	
KR0 到 KR7:	键值返回	TOQ00 到 TOQ03:	定时器输出
NMI:	非屏蔽中断请求	TXDA0 到 TXDA2:	发送数据
P02 到 P06:	端口 0	VDD:	供电电源
P10、P11:	端口 1	Vss:	地
P30 到 P39:	端口 3	$\overline{WAIT}$:	等待
P40 到 P42:	端口 4	$\overline{WR0}$:	写探针的低字节
P50 到 P55:	端口 5	$\overline{WR1}$:	写探针的高字节
P70 到 P711:	端口 7	X1、X2:	主时钟晶振
P90 到 P915:	端口 9	XT1、XT2:	副时钟晶振

1.6.2 内部单元

(1) CPU

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

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

(2) 总线控制单元（BCU）

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

(3) ROM

这是一个 640/512/384/256 KB 大小的掩模 ROM 或 flash 存储器映射地址空间为 0000000H 到 009FFFFH/0000000H 到 007FFFFH/0000000H 到 005FFFFH/0000000H 到 003FFFFH。在数据访问过程中，该 RAM 可由 CPU 在一个时钟周期内访问。

(4) RAM

V850ES/SG2 产品提供 48/40/32/24KB 的片上 RAM，映射地址为 3FF3000H~3FF5000H/ 3FF5000H~3FF5000H/ 3FF7000H~3FF7000H/ 3FF9000H~3FF9000H。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 ~ 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）

一个主时钟振荡器和一个副时钟振荡器能够分别提供主时钟振荡器频率（fx）和副时钟振荡器频率（fxT）。有两种模式。时钟交叉模式，fx 作为主时钟频率（fxx）。在 PLL 模式，将 fx 4 倍频或者 8 倍频。

CPU 时钟可以在 fxx, fxx/2, fxx/4, fxx/8, fxx/16, fxx/32 和 fxT 之间进行选择。

(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）。内置振荡器将提供给看门狗定时器 2 和定时器 M 作为时钟信号。

(8) 定时器/计数器

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

(9) 钟表定时器：

该定时器可以使用 32.768KHz 的副时钟或预分频器 3 提供的 32.768KHz 的 fBRG 作为计数时钟，提供用于钟表定时的参考时间间隔（0.5s）。钟表定时器也可以工作于主时钟，来实现间隔定时器功能。

(10) 看门狗定时器

V850ES/SG2 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、主时钟以及副时钟之间自由选择。

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

<R>

V850ES/SG2 和 V850ES/SG2-H 包括 3 种串行接口。异步串行接口 A（UARTA），三线可变长串行接口 B（CSIB）和一个 I²C 总线接口（I²C）。

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

使用 CSIB 时，数据通过 SOB0 ~ SOB4 引脚，SIB0 ~ SIB4 引脚以及 $\overline{\text{SCKB0}} \sim \overline{\text{SCKB4}}$ 引脚传输。

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

I²C 仅提供与 I²C 总线版本（Y 产品）（见表 1-1）。

(12) IEBus 控制器

IEBus 控制器是用于在器件间传输数字数据的小型数据传输系统。

只有含 IEBus 控制器的版本中带有 IEBus 控制器模块（见表 1-1）。

(13) CAN 控制器

CAN 控制器是用于在器件间传输数字数据的小型数据传输系统。

只有含 CAN 控制器的版本中带有 CAN 控制器模块（见表 1-1）。

(14) A/D 转换器

10 位 A/D 转换器包含 12 个模拟输入管脚。转换工作与连续逼近模式。

(15) D/A 转换器

V850ES/SG2 产品提供一个 2 通道 8 位精度的 D/A 转换器，该转换器采用 R-2R 梯形转换法进行转换。

(16) DMA 控制器

V850ES/SG2 产品提供一个 4 通道的 DMA 控制器。该控制器根据片上周边 I/O 器件发出的中断请求在内置 RAM 和片上周边 I/O 设备间传输数据。

(17) ROM 修正

V850ES/SG2 产品提供用于将掩模 ROM 上的程序替换为内部 RAM 程序的 ROM 修正功能。最多可设置 4 处修正地址。

(18) 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）。

(19) 实时输出功能

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

(20) CRC 功能

V850ES/SG2 产品提供一个可以针对指定的 8 位数据生成 16 位 CRC 码 (Cyclic Redundancy Check) 的 CRC 工作电路。

(21) 片上调试功能

V850ES/SG2 产品提供了符合 JTAG (Joint Test Action Group) 通信标准的片上调试功能, 可通过片上调试仿真器进行调试。通过控制引脚的输入电平和片上调试模式设置寄存器 (OCDM) 可以实现正常端口功能和片上调试功能间的切换。

片上调试功能仅能用于 flash 存储器版本。

(22) 端口

可使用以下通用端口功能和控制引脚功能。

端口	I/O	可转换功能
P0	5-位 I/O	NMI, 外部中断, A/D 转换器触发信号, 调试复位
P1	2-位 I/O	D/A 转换器模拟输入
P3	10-位 I/O	外部中断, 串行接口, 定时器 I/O, CAN 数据 I/O, IEBus 数据 I/O
P4	3-位 I/O	串行接口
P5	6-位 I/O	定时器 I/O, 实时输出, 按键中断输入, 串行接口, 调试 I/O
P7	12-位 I/O	A/D 转换器模拟输入
P9	16-位 I/O	外部地址总线, 串行接口, 按键中断输入, 定时器 I/O, 外部中断
PCM	4-位 I/O	外部控制信号
PCT	4-位 I/O	外部控制信号
PDH	6-位 I/O	外部地址总线
PDL	16-位 I/O	外部地址 / 数据总线

第 2 章 引脚功能

2.1 引脚功能列表

<R> V850ES/SG2 与 V850ES/SG2-H 引脚名称及功能描述如下。

有四种类型的引脚 I/O 缓冲式供电电源：AVREF0，AVREF1，BVDD，和 EVDD。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
AVREF0	端口 7
AVREF1	端口 1
BVDD	端口 CM，CT，DH（第 0 位 ~ 第 3 位），DL
EVDD	RESET，端口 0，3 到 5，9，DH（位 4，5）

(1) 端口引脚

(1/4)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
P02	19	17	I/O	端口 0 5-位 I/O 端口 可以位选输入 / 输出模式。 可以位选 N 沟道开漏输出。 5 V 耐压。	NMI
P03	20	18			INTP0/ADTRG
P04	21	19			INTP1
P05 ^{※1}	22	20			INTP2/DRST ^{※2}
P06	23	21			INTP3
P10	5	3	I/O	端口 1 2-位 I/O 端口 可以位选输入 / 输出模式。	ANO0
P11	6	4			ANO1

<R> 注 1. 从复位状态释放直到 OCDM.OCDM0 位清零，只要未使用片上调试功能，此管脚必须拉低。详情参见 4.6.3 片上调试管脚注意事项。集成了一个下拉电阻。它可以通过清除 OCDM.OCDM0 位来断开下拉电阻的连接。

2. 仅限 flash 存储器版本。

备注 GF: 100-pin 塑封 QFP（14 × 20）
GC: 100-pin 塑封 LQFP（细间距）（14 × 14）

(2/4)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
P30	27	25	I/O	端口 3 10-位 I/O 端口 可以位选输入/输出模式。 可以位选 N 沟道开漏输出。 5 V 耐压。	TXDA0/SOB4
P31	28	26			RXDA0/INTP7/SIB4
P32	29	27			ASCKA0/SCKB4/TIP00/TOP00
P33	30	28			TIP01/TOP01
P34	31	29			TIP10/TOP10
P35	32	30			TIP11/TOP11
P36	33	31			CTXD0 ^{注1} /IETX0 ^{注2}
P37	34	32			CRXD0 ^{注1} /IERX0 ^{注2}
P38	37	35			TXDA2/SDA00 ^{注2}
P39	38	36			RXDA2/SCL00 ^{注2}
P40	24	22	I/O	端口 4 3-位 I/O 端口 可以位选输入/输出模式。 可以位选 N 沟道开漏输出。 5 V 耐压。	SIB0/SDA01 ^{注2}
P41	25	23			SOB0/SCL01 ^{注2}
P42	26	24			SCKB0
P50	39	37	I/O	端口 5 6-位 I/O 端口 可以位选输入/输出模式。 可以位选 N 沟道开漏输出。 5 V 耐压。	TIQ01/KR0/TOQ01/RTP00
P51	40	38			TIQ02/KR1/TOQ02/RTP01
P52	41	39			TIQ03/KR2/TOQ03/RTP02/DDI ^{注3}
P53	42	40			SIB2/KR3/TIQ00/TOQ00/RTP03/DDO ^{注3}
P54	43	41			SOB2/KR4/RTP04/DCK ^{注4}
P55	44	42			SCKB2/KR5/RTP05/DMS ^{注4}
P70	2	100	I/O	端口 7 12-位 I/O 端口 可以位选输入/输出模式。	ANI0
P71	1	99			ANI1
P72	100	98			ANI2
P73	99	97			ANI3
P74	98	96			ANI4
P75	97	95			ANI5
P76	96	94			ANI6
P77	95	93			ANI7
P78	94	92			ANI8
P79	93	91			ANI9
P710	92	90			ANI10
P711	91	89			ANI11

- 注
1. 仅适用于 CAN 控制器版本
 2. 仅适用于内置 IEBus 控制器版本
 3. 仅适用于 I²C 总线版本 (Y 产品)
 4. 仅限 flash 存储器版本。

备注

GF: 100-pin 塑封 QFP (14 × 20)

GC: 100-pin 塑封 LQFP (细间距) (14 × 14)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
P90	45	43	I/O	端口 9 16-位 I/O 端口 可以位选输入/输出模式。 可以位选 N 沟道开漏输出。 5 V 耐压。	A0/KR6/TXDA1/SDA02 [※]
P91	46	44			A1/KR7/RXDA1/SCL02 [※]
P92	47	45			A2/TIP41/TOP41
P93	48	46			A3/TIP40/TOP40
P94	49	47			A4/TIP31/TOP31
P95	50	48			A5/TIP30/TOP30
P96	51	49			A6/TIP21/TOP21
P97	52	50			A7/SIB1/TIP20/TOP20
P98	53	51			A8/SOB1
P99	54	52			A9/SCKB1
P910	55	53			A10/SIB3
P911	56	54			A11/SOB3
P912	57	55			A12/SCKB3
P913	58	56			A13/INTP4
P914	59	57			A14/INTP5/TIP51/TOP51
P915	60	58			A15/INTP6/TIP50/TOP50
PCM0	63	61	I/O	端口 CM 4-位 I/O 端口 可以位选输入/输出模式。	WAIT
PCM1	64	62			CLKOUT
PCM2	65	63			HLDK
PCM3	66	64			HLDRQ
PCT0	67	65	I/O	端口 CT 4-位 I/O 端口 可以位选输入/输出模式。	WR0
PCT1	68	66			WR1
PCT4	69	67			RD
PCT6	70	68			ASTB
PDH0	89	87	I/O	端口 DH 6-位 I/O 端口 可以位选输入/输出模式。	A16
PDH1	90	88			A17
PDH2	61	59			A18
PDH3	62	60			A19
PDH4	8	6			A20
PDH5	9	7			A21

注 仅适用于 I²C 总线版本 (Y 产品)

备注 GF: 100-pin 塑封 QFP (14 × 20)
GC: 100-pin 塑封 LQFP (细间距) (14 × 14)

(4/4)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
PDL0	73	71	I/O	端口 DL 16-位 I/O 端口 可以位选输入/输出模式。	AD0
PDL1	74	72			AD1
PDL2	75	73			AD2
PDL3	76	74			AD3
PDL4	77	75			AD4
PDL5	78	76			AD5/FLMD1 [‡]
PDL6	79	77			AD6
PDL7	80	78			AD7
PDL8	81	79			AD8
PDL9	82	80			AD9
PDL10	83	81			AD10
PDL11	84	82			AD11
PDL12	85	83			AD12
PDL13	86	84			AD13
PDL14	87	85			AD14
PDL15	88	86			AD15

注 仅限 flash 存储器版本。

备注 GF: 100-pin 塑封 QFP (14 × 20)
GC: 100-pin 塑封 LQFP (细间距) (14 × 14)

(2) 非端口引脚

(1/6)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
<R>	A0	45	43	输出 访问外部存储器的地址总线 (当使用分离总线时) 当 A0 到 A15 用于分离总线模式时, 端口 9 不能用作端口管脚或其他复用功能管脚。 可以选择 N-通道开漏输出。 5 V 耐压。	P90/KR6/TXDA1/SDA02 ^{註1}
<R>	A1	46	44		P91/KR7/RXDA1/SCL02 ^{註1}
<R>	A2	47	45		P92/TIP41/TOP41
<R>	A3	48	46		P93/TIP40/TOP40
<R>	A4	49	47		P94/TIP31/TOP31
<R>	A5	50	48		P95/TIP30/TOP30
<R>	A6	51	49		P96/TIP21/TOP21
<R>	A7	52	50		P97/SIB1/TIP20/TOP20
<R>	A8	53	51		P98/SOB1
<R>	A9	54	52		P99/SCKB1
<R>	A10	55	53		P910/SIB3
<R>	A11	56	54		P911/SOB3
<R>	A12	57	55		P912/SCKB3
<R>	A13	58	56		P913/INTP4
<R>	A14	59	57		P914/INTP5/TIP51/TOP51
<R>	A15	60	58		P915/INTP6/TIP50/TOP50
	A16	89	87	输出	PDH0
	A17	90	88		PDH1
	A18	61	59		PDH2
	A19	62	60		PDH3
	A20	8	6		PDH4
	A21	9	7		PDH5
	AD0	73	71	I/O	PDL0
	AD1	74	72		PDL1
	AD2	75	73		PDL2
	AD3	76	74		PDL3
	AD4	77	75		PDL4
	AD5	78	76		PDL5/FLMD1 ^{註2}
	AD6	79	77		PDL6
	AD7	80	78		PDL7
	AD8	81	79		PDL8
	AD9	82	80		PDL9
	AD10	83	81		PDL10
	AD11	84	82		PDL11
	AD12	85	83		PDL12
	AD13	86	84		PDL13
	AD14	87	85		PDL14
	AD15	88	86		PDL15

- 注
1. 仅适用于 I²C 总线版本 (Y 产品)
 2. 仅限 flash 存储器版本。

备注

GF: 100-pin 塑封 QFP (14 × 20)

GC: 100-pin 塑封 LQFP (细间距) (14 × 14)

(2/6)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
ADTRG	20	18	输入	A/D 转换器外部触发输入。5 V 耐压。	P03/INTP0
ANI0	2	100	输入	A/D 转换器的模拟电压输入	P70
ANI1	1	99			P71
ANI2	100	98			P72
ANI3	99	97			P73
ANI4	98	96			P74
ANI5	97	95			P75
ANI6	96	94			P76
ANI7	95	93			P77
ANI8	94	92			P78
ANI9	93	91			P79
ANI10	92	90			P710
ANI11	91	89			P711
ANO0	5	3	输出	转换器的模拟电压输出	P10
ANO1	6	4			P11
ASCKA0	29	27	输入	UARTA0 波特率时钟输入。5 V 耐压。	P32/SCKB4/TIP00/TOP00
ASTB	70	68	输出	外部寄存器的地址触发信号输出	PCT6
AVREF0	3	1	—	A/D 转换器的参考电压输入 提供给复用功能端口 7 的正电源	—
AVREF1	7	5		A/D 转换器的参考电压输入 提供给复用功能端口 1 的正电源	—
AVSS	4	2	—	A/D 和 D/A 转换器的地电位 (与 VSS 电压相同)	—
BVDD	72	70	—	总线接口和复用功能端口的供电引脚	—
BVSS	71	69	—	G 总线接口和复用功能端口的地电平	—
CLKOUT	64	62	输出	内部系统时钟输出	PCM1
CRXD0 ^{注1}	34	32	输入	CAN 接受数据输入。5 V 耐压。	P37/IERX0 ^{注2}
CTXD0 ^{注1}	33	31	输出	CAN 传送数据输出 N-通道 开漏输出选择。5 V 耐压。	P36/IETX0 ^{注2}
DCK ^{注3}	43	41	输入	调试时钟输入。5 V 耐压。	P54/SOB2/KR4/RTP04
DDI ^{注3}	41	39	输入	调试数据输入。5 V 耐压。	P52/TIQ03/KR2/TOQ03/RTP02
DDO ^{注3, 4}	42	40	输出	调试数据输出 N-通道 开漏输出选择。5 V 耐压。	P53/SIB2/KR3/TIQ00/TOQ00/ RTP03
DMS ^{注2}	44	42	输入	调试模式选择输入。5 V 耐压。	P55/SCKB2/KR5/RTP05
DRST ^{注3}	22	20	输入	调试复位输入。5 V 耐压。	P05/INTP2
EVDD	36	34	—	外部供电电源 (与 VDD 电压相同)	—
EVSS	35	33	—	外部地电平 (与 VSS 电压相同)	—
FLMD0 ^{注3}	10	8	输入	flash 存储器编程模式设置引脚	—
FLMD1 ^{注3}	78	76			PDL5/AD5

- 注
1. 仅适用于 CAN 控制器版本
 2. 仅适用于内置 IEBus 控制器版本
 3. 仅限 flash 存储器版本。
 4. 在片上调试模式中，端口被强制设置为高电平。

备注 GF: 100-pin 塑封 QFP (14 × 20)
GC: 100-pin 塑封 LQFP (细间距) (14 × 14)

(3/6)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
HLD $\overline{\text{AK}}$	65	63	输出	总线保持应答输出	PCM2
HLD $\overline{\text{RQ}}$	66	64	输入	总线保持请求输入	PCM3
IC ^{注1}	10	8	—	内部连接	—
I $\overline{\text{ERX0}}$ ^{注2}	34	32	输入	IEBus 接受数据输入。5 V 耐压。	P37/CRXD0 ^{注3}
I $\overline{\text{ETX0}}$ ^{注2}	33	31	输出	IEBus 传送数据输出 N-通道开漏输出选择。5 V 耐压。	5 V 耐压 P36/CTXD0 ^{注3}
INTP0	20	18	输入	5 V 耐压。 外部中断请求输入（可屏蔽，模拟噪声消除电路）。 引脚 INTP3 可选模拟噪声消除电路或数字噪声消除电路。 5 V 耐压。	P03/ADTRG
INTP1	21	19			P04
INTP2	22	20			P05/ $\overline{\text{DRST}}$ ^{注4}
INTP3	23	21			P06
INTP4	58	56			P913/A13
INTP5	59	57			P914/A14/TIP51/TOP51
INTP6	60	58			P915/A15/TIP50/TOP50
INTP7	28	26			P31/RXDA0/SIB4
KR0 ^{注5}	39	37	输入	键盘中断输入（片上模拟噪声消除器）。 5 V 耐压。	P50/TIQ01/TOQ01/RTP00
KR1 ^{注5}	40	38			P51/TIQ02/TOQ02/RTP01
KR2 ^{注5}	41	39			P52/TIQ03/TOQ03/ RTP02/DDI ^{注4}
KR3 ^{注5}	42	40			P53/SIB2/TIQ00/TOQ00/ RTP03/DDO ^{注4}
KR4 ^{注5}	43	41			P54/SOB2/RTP04/DCK ^{注4}
KR5 ^{注5}	44	42			P55/ $\overline{\text{SCKB2}}$ /RTP05/DMS ^{注4}
KR6 ^{注5}	45	43			P90/A0/TXDA1/SDA02 ^{注6}
KR7 ^{注5}	46	44			P91/A1/RXDA1/SCL02 ^{注6}
NMI ^{注7}	19	17	输入	外部中断输入（可屏蔽，模拟噪声消除电路）。5 V 耐压。	P02
$\overline{\text{RD}}$	69	67	输出	信号外部存储器的读取触发信号输出	PCT4
REGC	12	10	—	调节器输出稳压电容的连接（4.7 μF ）	—
$\overline{\text{RESET}}$	16	14	输入	系统复位输入	—

- 注
1. 仅限 Mask ROM 版本
 2. 仅适用于内置 IEBus 控制器版本
 3. 仅适用于 CAN 控制器版本
 4. 仅限 flash 存储器版本。
 5. 外部上拉该引脚
 6. 仅适用于 I²C 总线版本（Y 产品）
 7. NMI 管脚作为 P02 管脚复用功能。复位之后实现 P02 管脚功能。要使能引脚 NMI，设置 PMC0.PMC02 为 1。引脚 NMI 的初始设置为“无边沿检测”。通过 INTF0 和 INTR0 寄存器选择引脚 NMI 的有效边沿。

备注 GF: 100-pin 塑封 QFP（14 × 20）
GC: 100-pin 塑封 LQFP（细间距）（14 × 14）

(4/6)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
RTP00	39	37	输出	实时输出端口。 N-通道开漏输出选择。 5 V 耐压。	5 V 耐压 P50/TIQ01/KR0/TOQ01
RTP01	40	38			P51/TIQ02/KR1/TOQ02
RTP02	41	39			P52/TIQ03/KR2/TOQ03/DDI ^{‡1}
RTP03	42	40			P53/SIB2/KR3/TIQ00/TOQ00/ DDO ^{‡1}
RTP04	43	41			P54/SOB2/KR4/DCK ^{‡1}
RTP05	44	42			P55/ $\overline{\text{SCKB2}}$ /KR5/DMS ^{‡1}
RXDA0	28	26	输入	串行接收数据输入 (UARTA0 ~ UARTA2) 5 V 耐压。	P31/INTP7/SIB4
RXDA1	46	44			P91/A1/KR7/SCL02 ^{‡2}
RXDA2	38	36			P39/SCL00 ^{‡2}
$\overline{\text{SCKB0}}$	26	24	I/O	串行时钟 I/O (CSIB0 ~ CSIB4) 可以位选 N 沟道开漏输出。 5 V 耐压。	P42
$\overline{\text{SCKB1}}$	54	52			P99/A9
$\overline{\text{SCKB2}}$	44	42			P55/KR5/RTP05/DMS ^{‡1}
$\overline{\text{SCKB3}}$	57	55			P912/A12
$\overline{\text{SCKB4}}$	29	27			P32/ASCKA0/TIP00/TOP00
SCL00 ^{‡2}	38	36	I/O	串行时钟 I/O () 可以位选 N 沟道开漏输出。 5 V 耐压。	P39/RXDA2
SCL01 ^{‡2}	25	23			P41/SOB0
SCL02 ^{‡2}	46	44			P91/A1/KR7/RXDA1
SDA00 ^{‡2}	37	35	I/O	串行发送/接收 I/O (I ² C00 到 I ² C02) 可以位选 N 沟道开漏输出。 5 V 耐压。	P38/TXDA2
SDA01 ^{‡2}	24	22			P40/SIB0
SDA02 ^{‡2}	45	43			P90/A0/KR6/TXDA1
SIB0	24	22	输入	串行接收数据输入 (CSIB0 到 CSIB4) 5 V 耐压。	P40/SDA01 ^{‡2}
SIB1	52	50			P97/A7/TIP20/TOP20
SIB2	42	40			P53/KR3/TIQ00/TOQ00/ RTP03/DDO ^{‡1}
SIB3	55	53			P910/A10
SIB4	28	26			P31/RXDA0/INTP7
SOB0	25	23	输出	串行发送数据输出 (CSIB0 到 CSIB4) 可以位选 N 沟道开漏输出。 5 V 耐压。	P41/SCL01 ^{‡2}
SOB1	53	51			P98/A8
SOB2	43	41			P54/KR4/RTP04/DCK ^{‡1}
SOB3	56	54			P911/A11
SOB4	27	25			P30/TXDA0

- 注
1. 仅限 flash 存储器版本。
 2. 仅适用于 I²C 总线版本 (Y 产品)

备注

GF: 100-pin 塑封 QFP (14 × 20)

GC: 100-pin 塑封 LQFP (细间距) (14 × 14)

(5/6)

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

注 仅限 flash 存储器版本。

备注 GF: 100-pin 塑封 QFP (14 × 20)
GC: 100-pin 塑封 LQFP (细间距) (14 × 14)

(6/6)

引脚名称	引脚编号		I/O	功能	复用功能
	GF	GC			
TOP00	29	27	输出	定时器输出 (TMP0)	P32/ASCKA0/SCKB4/TIP00
TOP01	30	28		可选 N 沟道开漏输出。5 V 耐压。	P33/TIP01
TOP10	31	29		定时器输出 (TMP1)	P34/TIP10
TOP11	32	30		可选 N 沟道开漏输出。5 V 耐压。	P35/TIP11
TOP20	52	50		定时器输出 (TMP2)	P97/A7/SIB1/TIP20
TOP21	51	49		可选 N 沟道开漏输出。5 V 耐压。	P96/A6/TIP21
TOP30	50	48		定时器输出 (TMP3)	P95/A5/TIP30
TOP31	49	47		可选 N 沟道开漏输出。5 V 耐压。	P94/A4/TIP31
TOP40	48	46		定时器输出 (TMP4)	P93/A3/TIP40
TOP41	47	45		可选 N 沟道开漏输出。5 V 耐压。	P92/A2/TIP41
TOP50	60	58		定时器输出 (TMP5)	P915/A15/INTP6/TIP50
TOP51	59	57		可选 N 沟道开漏输出。5 V 耐压。	P914/A14/INTP5/TIP51
TOQ00	42	40	输出	定时器输出 (TMQ0)	P53/SIB2/KR3/TIQ00/RTP03/ DDO ^{‡1}
TOQ01	39	37		可选 N 沟道开漏输出。5 V 耐压。	P50/TIQ01/KR0/RTP00
TOQ02	40	38			P51/TIQ02/KR1/RTP01
TOQ03	41	39			P52/TIQ03/KR2/RTP02/DDI ^{‡1}
TXDA0	27	25	输出	串行发送数据输出 (UARTA0 到 UARTA4)	P30/SOB4
TXDA1	45	43		可以位选 N 沟道开漏输出。	P90/A0/KR6/SDA02 ^{‡2}
TXDA2	37	35		5 V 耐压。	P38/SDA00 ^{‡2}
V _{DD}	11	9	—	内部供电电源引脚	—
V _{SS}	13	11	—	内部地电平	—
WAIT	63	61	输入	外部等待输入	PCM0
WR0	67	65	输出	外部存储器写触发 (低 8 位)	PCT0
WR1	68	66		外部存储器写触发 (高 8 位)	PCT1
X1	14	12	输入	连接主时钟振荡器	—
X2	15	13	—		—
XT1	17	15	输入	连接副时钟振荡器	—
XT2	18	16	—		—

- 注
1. 仅限 flash 存储器版本。
 2. 仅适用于 I²C 总线版本 (Y 产品)

备注 GF: 100-pin 塑封 QFP (14 × 20)
GC: 100-pin 塑封 LQFP (细间距) (14 × 14)

2.2 引脚状态

如下为各种模式下的引脚操作状态。

<R>

表 2-2. 各种模式下的引脚操作状态

引脚名称	复位期间 (在刚上电后)	复位期间 (除 去刚上电后时)	HALT 模式 ^{注2}	IDLE1, IDLE2, Sub- IDLE 模式 ^{注2}	STOP 模式 ^{注2}	Idle 模式 ^{注3}	总线保持
P05/DRST ^{注4}	拉低	拉低 ^{注5}	保持	保持	保持	保持	保持
保持 P10/ANO0, P11/ANO1	未定义 ^{注1}	Hi-Z	保持	保持	注 11	保持	保持
P53/DDO ^{注4}		Hi-Z ^{注6}	保持	保持	保持	保持	保持
AD0 到 AD15	Hi-Z ^{注7}	Hi-Z ^{注7}	注 8, 9	Hi-Z	Hi-Z	保持	Hi-Z
A0 到 A15			未定义 ^{注8, 10}				
A16 到 A21			未定义 ^{注8}				
WAIT			—	—	—	—	—
CLKOUT			操作	L	L	操作	操作
WR0, WR1 输入:			H ^{注8}	H	H	H	Hi-Z
RD							
ASTB			操作 ^{注8}	—	—	—	L
HLDAR							操作
HLDAR			—	—	—	—	操作
其它端口引脚	Hi-Z	Hi-Z	保持	保持	保持	保持	保持

<R>

- 注
1. 上电时这些管脚可能瞬时输出为未知电平。
 2. 使用复用功能时的操作。
 3. 分离总线模式下，在显示 T2 状态后空闲状态下的引脚状态被插入。复用总线模式下，在显示 T3 状态后空闲状态下的引脚状态被插入。
 4. 仅限 flash 存储器版本。
 5. 外部复位期间置低。在看门狗定时器、时钟监测器等引发内部复位期间，引脚状态根据 OCDM.OCDM0 位的设置不同而不同。
 6. DDO 输出在片上调试模式中定义。
 7. 总线控制引脚功能复用为端口引脚，所以被初始化为输入模式（端口模式）。
 8. HALT 模式下继续操作，在 DMA 操作期间。
 9. 分立总线模式中：Hi-Z
复用总线模式中：未定义
 10. 分离总线模式中
 11. 在端口模式：保持
当功能复用使用中：Hi-Z

<R>

备注

Hi-Z: 高阻态
保持: 在之前的外部总线周期期间状态保持。
L: 低电平输出
H: 高电平输出
—: 无采样输入（无应答）

2.3 I/O引脚电路类型, I/O缓冲供电电平, 和不使用引脚的连接

(1/3)

引脚	复用功能	引脚编号		I/O 电路类型	推荐连接	
		GF	GC			
P02	NMI	19	17	10-D	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 输出： 保持开路。	
P03	INTP0/ADTRG	20	18			
P04	INTP1	21	19			
P05	INTP2/ $\overline{\text{DRST}}$ ^{‡1}	22	20	10-N	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 禁止固定到 V _{DD} 电平。 输出： 保持开路。 $\overline{\text{RESET}}$ 引脚复位后内部置低。	
P06	INTP3	23	21	10-D	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 输出： 保持开路。	
P10, P11	ANO0, ANO1	5, 6	3, 4	12-D	输入： 通过一个电阻单独连接到 AV _{REF1} 或 AV _{SS} 。 输出： 保持开路。	
<R>	P30	TXDA0/SOB4	27	25	10-G	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 输出： 保持开路。
	P31	RXDA0/INTP7/SIB4	28	26	10-D	
	P32	ASCKA0/ $\overline{\text{SCKB4}}$ /TIP00	29	27		
	P33	TIP01/TOP01	30	28		
	P34	TIP10/TOP10	31	29		
	P35	TIP11/TOP11	32	30		
<R>	P36	CTXD0 ^{‡2} / $\overline{\text{IETX0}}$ ^{‡3}	33	31	10-G	10-D
	P37	CRXD0 ^{‡2} / $\overline{\text{IERX0}}$ ^{‡3}	34	32		
	P38	TXDA2/SDA00 ^{‡4}	37	35		
	P39	RXDA2/SCL00 ^{‡4}	38	36		
	P40	SIB0/SDA01 ^{‡4}	24	22		
	P41	SOB0/SCL01 ^{‡4}	25	23		
	P42	$\overline{\text{SCKB0}}$	26	24		

- 注
1. 仅限 flash 存储器版本。
 2. 仅适用于 CAN 控制器版本
 3. 仅适用于 IEBus 控制器版本
 4. 仅适用于 I²C 总线版本 (Y 产品)

备注 GF: 100-pin 塑封 QFP (14 × 20)
GC: 100-pin 塑封 LQFP (细间距) (14 × 14)

(2/3)

引脚	复用功能	引脚编号		I/O 电路类型	推荐连接
		GF	GC		
P50	TIQ01/KR0/TOQ01/RTP00	39	37	10-D	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 输出： 保持开路。
P51	TIQ02/KR1/TOQ02/RTP01	40	38		
P52	TIQ03/KR2/TOQ03/RTP02/DDI ^{‡1}	41	39		
P53	SIB2/KR3/TIQ00/TOQ00/RTP03/DDO ^{‡1}	42	40		
P54	SOB2/KR4/RTP04/DCK ^{‡1}	43	41		
P55	$\overline{\text{SCKB2}}$ /KR5/RTP05/DMS ^{‡1}	44	42		
P70 到 P711	ANI0 到 ANI11	2, 1, 100 到 91	100 到 89	11-G	输入： 通过一个电阻单独连接到 AV _{REF0} 或 AV _{SS} 。 输出： 保持开路。
P90	A0/KR6/TDXA1/SDA02 [‡]	45	43	10-D	输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 输出： 保持开路。
P91	A1/KR7/RXDA1/SCL02 ^{‡2}	46	44		
P92	A2/TIP41/TOP41	47	45		
P93	A3/TIP40/TOP40	48	46		
P94	A4/TIP31/TOP31	49	47		
P95	A5/TIP30/TOP30	50	48		
P96	A6/TIP21/TOP21	51	49		
P97	A7/SIB1/TIP20/TOP20	52	50		
P98	A8/SOB1	53	51	10-G	
P99	A9/ $\overline{\text{SCKB1}}$	54	52	10-D	
P910	A10/SIB3	55	53		
P911	A11/SOB3	56	54	10-G	
P912	A12/ $\overline{\text{SCKB3}}$	57	55	10-D	
P913	A13/INTP4	58	56		
P914	A14/INTP5/TIP51/TOP51	59	57		
P915	A15/INTP6/TIP50/TOP50	60	58		

- 注
1. 仅限 flash 存储器版本。
 2. 仅适用于 I²C 总线版本（Y 产品）

备注

GF: 100-pin 塑封 QFP（14 × 20）

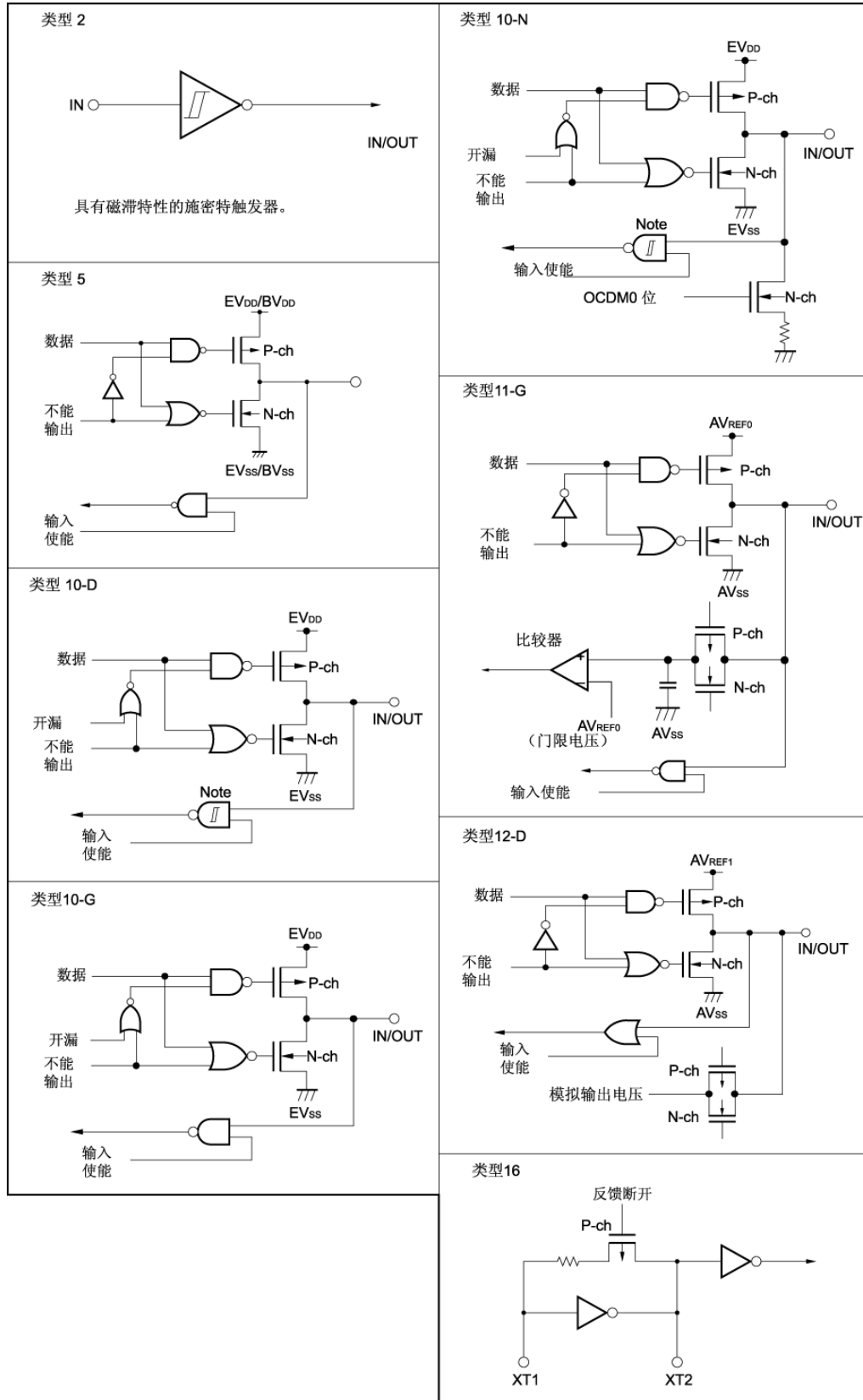
GC: 100-pin 塑封 LQFP（细间距）（14 × 14）

引脚	复用功能	引脚编号		I/O 电路类型	推荐连接
		GF	GC		
PCM0	WAIT	63	61	5	输入： 通过一个电阻单独连接到 BV _{DD} 或 BV _{SS} 。 输出： 保持开路。
PCM1	CLKOUT	64	62		
PCM2	HLD $\overline{\text{AK}}$	65	63		
PCM3	HLD $\overline{\text{RQ}}$	66	64		
PCT0, PCT1	WR $\overline{0}$, WR1	67, 68	65, 66		
PCT4	RD $\overline{}$	69	67		
PCT6	ASTB	70	68		
PDH0 到 PDH3	A16 到 A19	89, 90, 61, 62	87, 88, 59, 60		
PDH4, PDH5	A20, A21	8, 9	6, 7		输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 输出： 保持开路。
PDL0 到 PDL4	AD0 到 AD4	73 到 77	71 到 75		输入： 通过一个电阻单独连接到 EV _{DD} 或 EV _{SS} 。 输出： 保持开路。
PDL5	AD5/FLMD1 ^{‡1}	78	76		
PDL6 到 PDL15	AD6-AD15	79 到 88	77 到 86		
PDL16 到 PDL31	AD16-AD31	89 到 98	87 到 96		
AV _{REF0}	—	3	1	—	直接输出 到 V _{DD} 并且提供电源。
AV _{REF1}	—	7	5	—	直接输出 到 V _{DD} 并且提供电源。
AV _{SS}	—	4	2	—	直接输出 到 V _{SS} 并且提供电源。
BV _{DD}	—	72	70	—	直接输出 到 V _{DD} 并且提供电源。
BV _{SS}	—	71	69	—	直接输出 到 V _{SS} 并且提供电源。
EV _{DD}	—	36	34	—	—
EV _{SS}	—	35	33	—	—
FLMD0 ^{‡1}	—	10	8	—	在非 flash 存储器编程模式下直接输出 到 V _{SS} 。
IC ^{‡2}	—	10	8	—	直接输出到 V _{SS} 。
REGC	—	12	10	—	连接标准输出稳压电容 (4.7 μ F)。
RESET	—	16	14	2	—
V _{DD}	—	11	9	—	—
V _{SS}	—	13	11	—	—
X1	—	14	12	—	—
X2	—	15	13	—	—
XT1	—	17	15	16	连接到 V _{SS} 。
XT2	—	18	16	16	保持开路。

注 1. 仅限 flash 存储器版本。
2. 仅限 Mask ROM 版本

备注 GF: 100-pin 塑封 QFP (14 × 20)
GC: 100-pin 塑封 LQFP (细间距) (14 × 14)

图 2-1. I/O 引脚电路



注 普通端口模式中不会有磁滞特性。

2.4 注意事项

(1) 上电注意事项

<R> 当上电时，下列管脚瞬时输出为未知电平。

- P10/ANO0 管脚
- P11/ANO1 pin
- P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO[※]管脚

注 DDO 管脚仅在含 flash 存储器的版本中提供。

(2) FLMD0 管脚注意事项

<R> 为了准确启动用户程序的操作，从复位释放状态直到振荡器稳定并且固件操作完成，都要保证 FLMD0 管脚拉低。对于详细的固件操作，参见 25.3.5 (2) 固件操作（仅限 flash 存储器版本）。

<R> V850ES/ SG2 和 V850ES/SG2-H 系列产品的 CPU 是基于 RISC 结构设计的，具备 5 级流水线结构，几乎所有指令均可在一个时钟周期内完成。

3.1 特点

- <R>

○ 最小指令执行时间:

V850ES/SG2:

50ns（主时钟（f_{xx}）= 20 MHz）

V850ES/SG2-H:

31.25ns（主时钟（f_{xx}）= 32 MHz）
- 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 (load / store)指令
- 4 种位操作指令

• SET1

• CLR1

• NOT1

• TST1

3.2 CPU 寄存器组

<R> V850ES/SG2 和 V850ES/SG2-H 寄存器可分为两组：通用程序寄存器组和专用系统寄存器组。所有的寄存器的宽度均为 32 位。

详情请参考 **V850ES 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 运行状态保存寄存器)
r11		CTPSW	(CALLT 运行状态保存寄存器)
r12			
r13			
r14		DBPC	(异常/调试陷阱运行状态保存寄存器)
r15		DBPSW	(异常/调试陷阱运行状态保存寄存器)
r16			
r17		CTBP	(CALLT基指针)
r18			
r19			
r20			
r21			
r22			
r23			
r24			
r25			
r26			
r27			
r28			
r29			
r30	(元素指针(EP))		
r31	(连接指针(LP))		
31	0		
PC	(程序计数器)		

3.2.1 程序寄存器组

程序寄存器组包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 这两个寄存器是被指令隐含使用的，因此要慎用使用。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~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

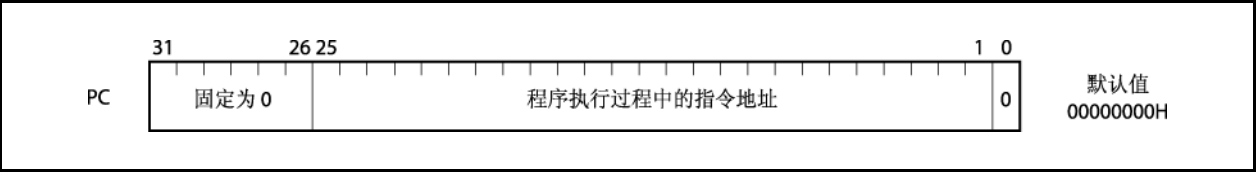
名称	用途	操作
r0	零寄存器	始终保持为零
r1	汇编器保留寄存器	用于生成 32 位立即数据的工作寄存器
r2	地址 / 数据寄存器(不被实时操作系统所使用时)	
r3	堆栈指针	当函数被调用时用于生成堆栈结构
r4	全局指针	用于访问数据区域内的全局变量
r5	文本指针	用于表示文本区域(放置程序程序代码的区域)的开始部分的寄存器
r6 ~ r29	地址 / 数据变量寄存器	
r30	元素指针	访问存储器时用于生成地址的基指针
r31	连接指针	用于编译器调用函数时
PC	程序计数器	程序执行时存放指令地址

备注 汇编器和 C 编译器所使用的 r1, r3 ~ 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 32 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，跳往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	操作数规范	
		LDSR 指令	STSR 指令
0	中断状态保存寄存器(EIPC) ^{註 1}	√	√
1	中断状态保存寄存器(EIPSW) ^{註 1}	√	√
2	NMI 状态保存寄存器 (FEPC) ^{註 1}	√	√
3	NMI 状态保存寄存器 (FEPSW) ^{註 1}	√	√
4	中断源寄存器 (ECR)	×	√
5	程序状态字 (PSW)	√	√
6 ~ 15	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)	√ ^{註 2}	√ ^{註 2}
19	异常陷阱状态保存寄存器(DBPSW)	√ ^{註 2}	√ ^{註 2}
20	CALLT 基址针 (CTBP)	√	√
21 ~ 31	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。

<R>

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(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)。

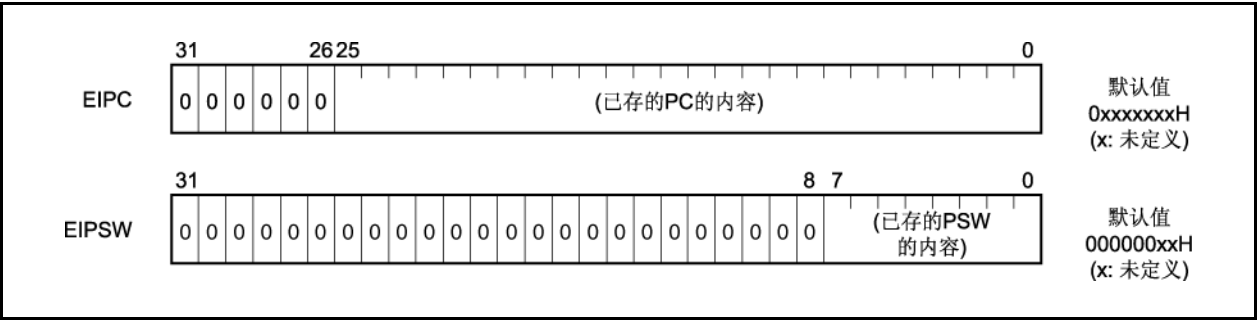
软件异常和可屏蔽中断发生时，当前指令的下一条指令的地址会被保存到 EIPC 中(一些指令除外，详情请参阅 22.8 CPU 不对中断进行确认的时期)。

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。



(2) 不可屏蔽中断状态保存寄存器 (FEPC 和 FEPSW)

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，PC 的值被保存在 FEPC 中，程序状态字 PSW 的值被保存在 FEPSW 中。

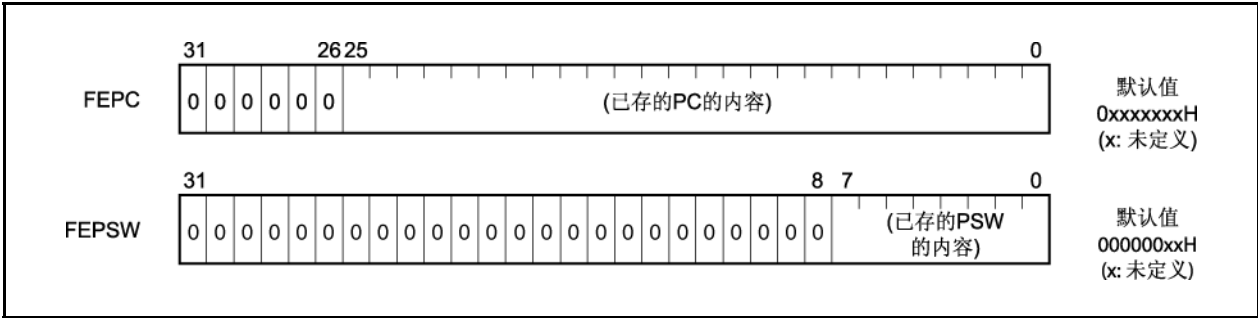
不可屏蔽中断发生时，当前指令的下一条指令的地址会被保存到 FEPC 中。

当前的 PSW 值被保存在 FEPSW 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。

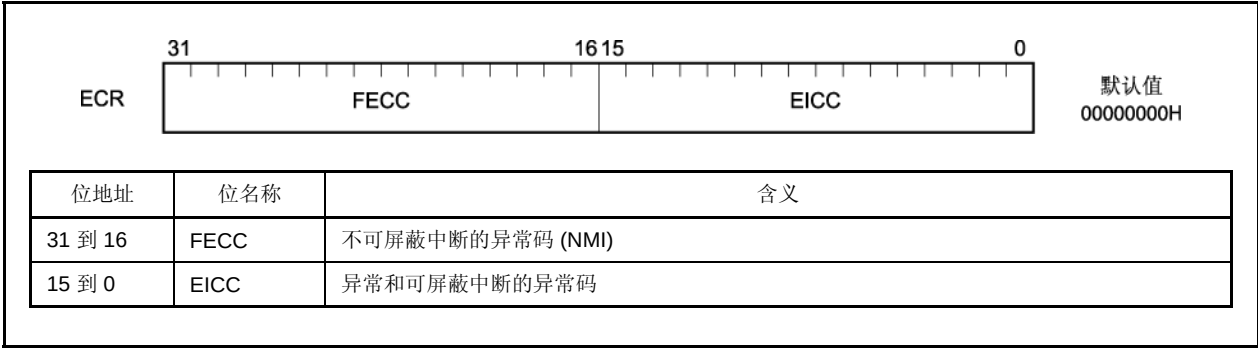
FEPC 的第 26 到 31 位和 FEPSW 的第 8 到 31 位为扩展功能保留(这些位始终为 0)。

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



(3) 中断源寄存器 (ECR)

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 8 7 6 5 4 3 2 1 0																默认值 00000020H							
		RFU																NP	EP	ID	SAT	CY	OV	S	Z

位地址	标志名称	含义
31 到 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

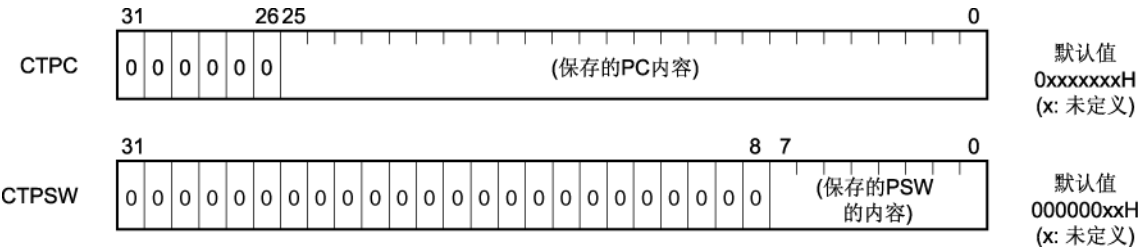
请阅读下页中的注

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

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

(5) CALLT 执行状态保存寄存器 (CTPC 和 CTPSW)

CTPC 和 CTPSW 是 CALLT 执行状态保存寄存器。
当 CALLT 指令执行时，PC 值被保存在 CTPC 中，PSW 值被保存在 CTPSW 中。
被保存在 CTPC 中的值是 CALLT 下一条指令的地址。
保存在 CTPSW 中的值是当前的 PSW 值。
CTPC 的第 26 到 31 位和 CTPSW 的第 8 到 31 位为扩展功能保留(始终为 0)。



(6) 异常 / 调试陷阱状态保存寄存器 (DBPC 和 DBPSW)

DBPC 和 DBPSW 是异常 / 调试陷阱状态保存寄存器。

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

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

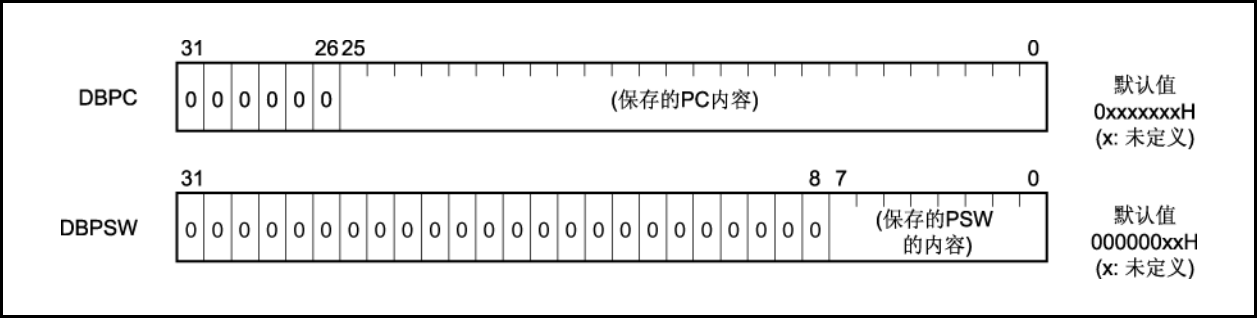
保存在 DBPSW 中的值是当前的 PSW 值。

该寄存器只有在执行 DBTRAP 指令和 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。

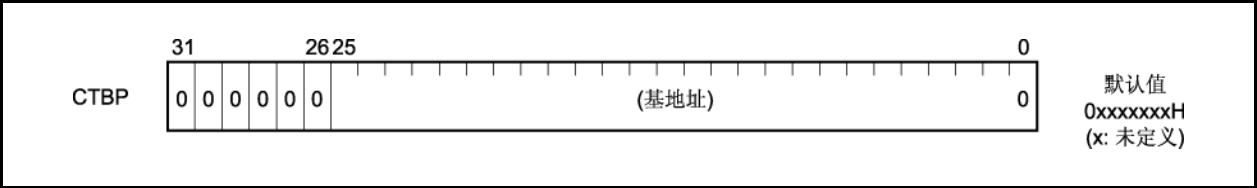
<R>



(7) CALLT 基址针 (CTBP)

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

第 26 到 31 位为扩展功能保留(始终为 0)。



3.3 操作模式

<R> V850ES/SG2 与 V850ES/SG2-H 有如下的操作模式。

(1) 正常操作模式

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

(2) flash 存储器编程模式

在这个模式下，内部 flash 存储器可以用 flash 存储器编程器进行烧写。

<R> 下列产品是 V850ES/SG2 与 V850ES/SG2-H 的片上 flash 存储器版本。

- <R>
- μ PD70F3261, 70F3261Y, 70F3263, 70F3263Y, 70F3271, 70F3271Y, 70F3273, 70F3273Y, 70F3281, 70F3281Y, 70F3283, 70F3283Y, 70F3263HY, 70F3273HY, 70F3283HY

(3) 片上调试模式

V850ES/ SG2 与 V850ES/SG2-H 支持 JTAG(Joint Test Action Group)格式的在线调试，可通过在线仿真器对器件进行在线调试。

只有在 flash 存储器版本才提供片上调试功能。

详见 第 31 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。

<R> 正常模式下，在复位释放之后，振荡器稳定之前，给 FLMD0/IC 管脚输入一个低电平，固件操作完成。

Flash 存储器编程模式下，flash 存储器编程器应该输入 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 存储器编程模式
H	H	禁止设置

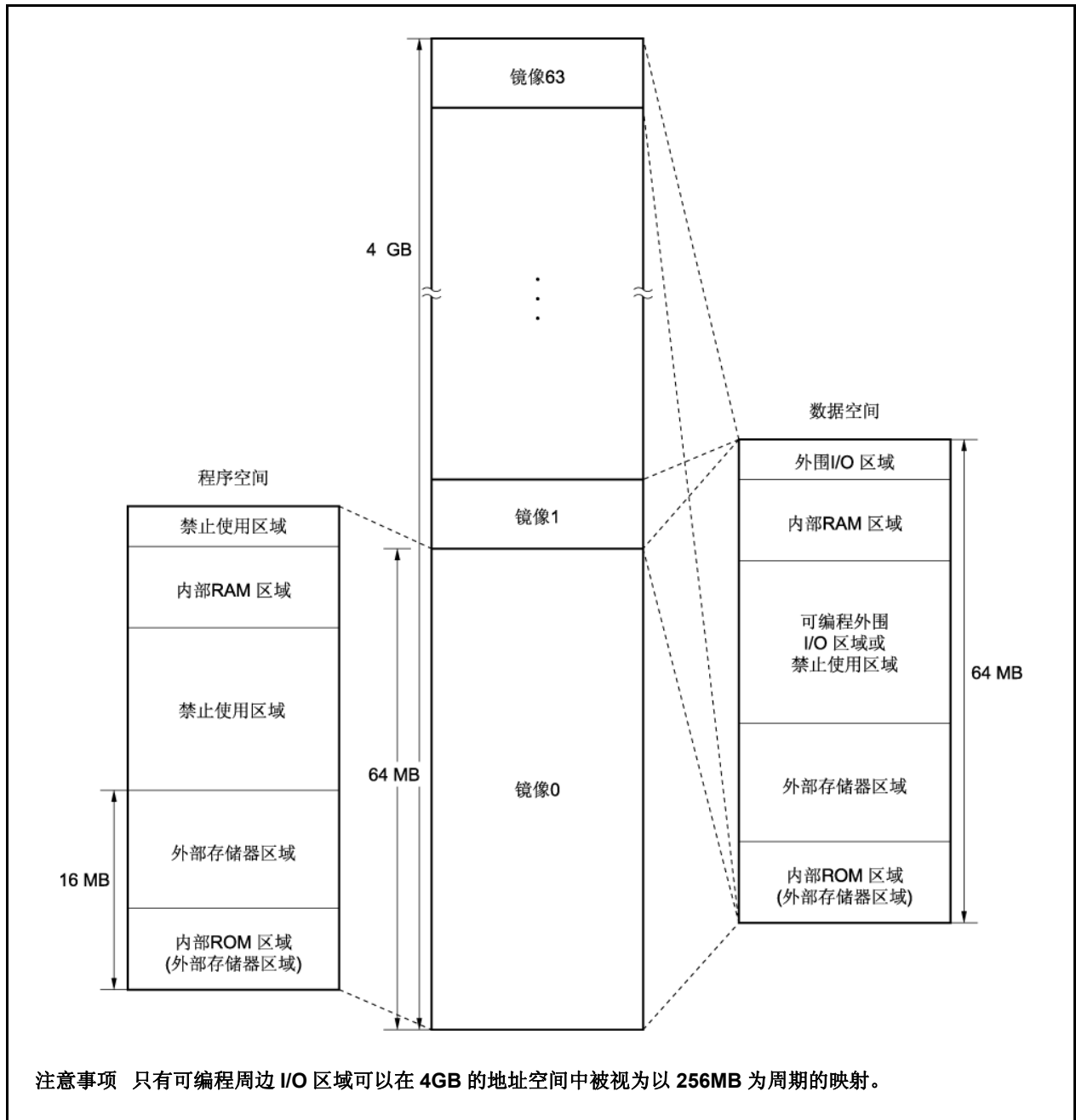
备注 L:低电平输入
 H:高电平输入
 ×: 不关心

3.4 地址空间

3.4.1 CPU 地址空间

对于寻址指令，16 MB 外部存储区和内部 ROM 区，外加内部 RAM 区加起来构成 64MB 的线性地址空间（程序空间）。对于寻址操作（数据访问），支持高达 4GB 的线性地址空间（数据空间）。4GB 地址空间，可以看作 64 个 64MB 的物理地址空间。这意味着无论地址位的第 26 到 31 位为何值，系统总是在同样的 64MB 的物理地址空间中进行寻址。

图 3-1. 地址空间图



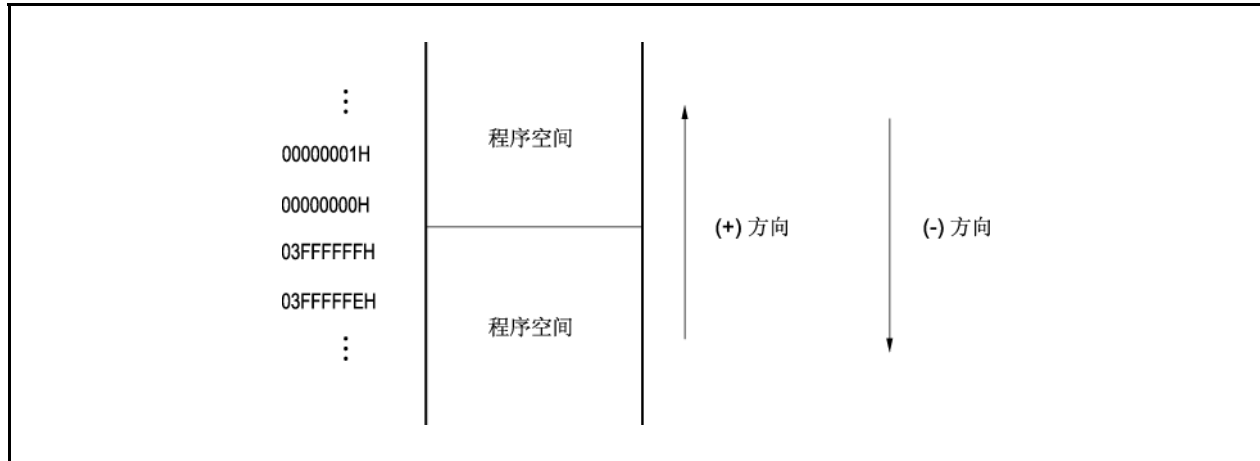
3.4.2 CPU 地址空间的绕回

(1) 程序空间

PC(程序计数器)的 32 位中, 高 6 位固定为 0, 只有低 26 位有效。即使分支地址计算过程中产生第 25 到 26 位的进位或借位, 也会被高 6 位忽略。

因此程序空间的地址下限(地址 00000000H)和地址上限(地址 03FFFFFFH)成为连续的地址。绕回所指的就是这种地址下限和地址上限连续的情况。

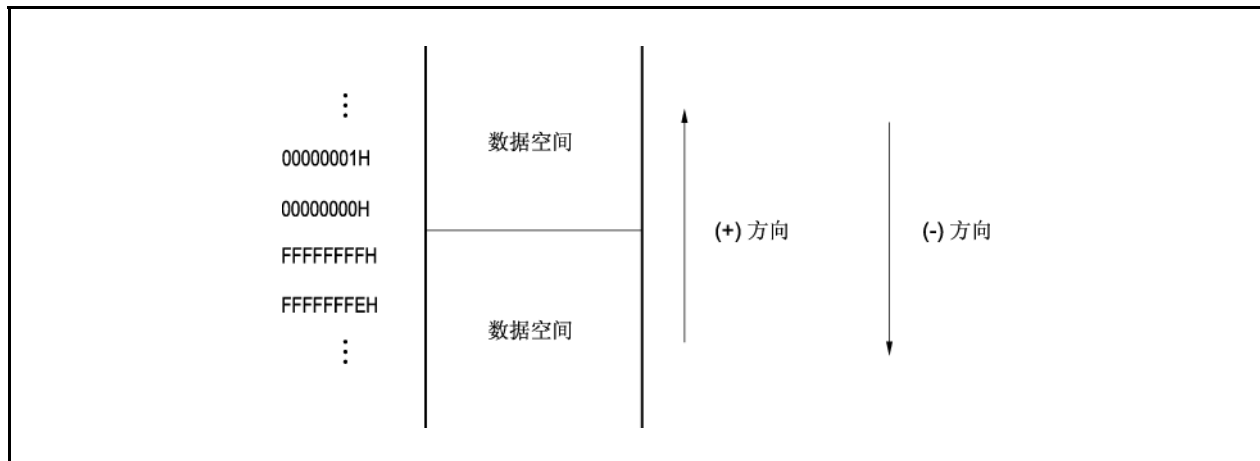
注意事项 由于从地址 03FFF000H 到 03FFFFFFH 的 4KB 范围是片上外围 I/O 设备区域, 不能从这个区域进行取指令操作。因此, 对于跳转地址计算结果会影响此区域的操作, 不要执行。



(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些地址的边界处。



3.4.3 存储器映射

<R> 如下是 V850ES/SG2 与 V850ES/SG2-H 地址空间。

图 3-2. 数据存储映射 (物理地址)

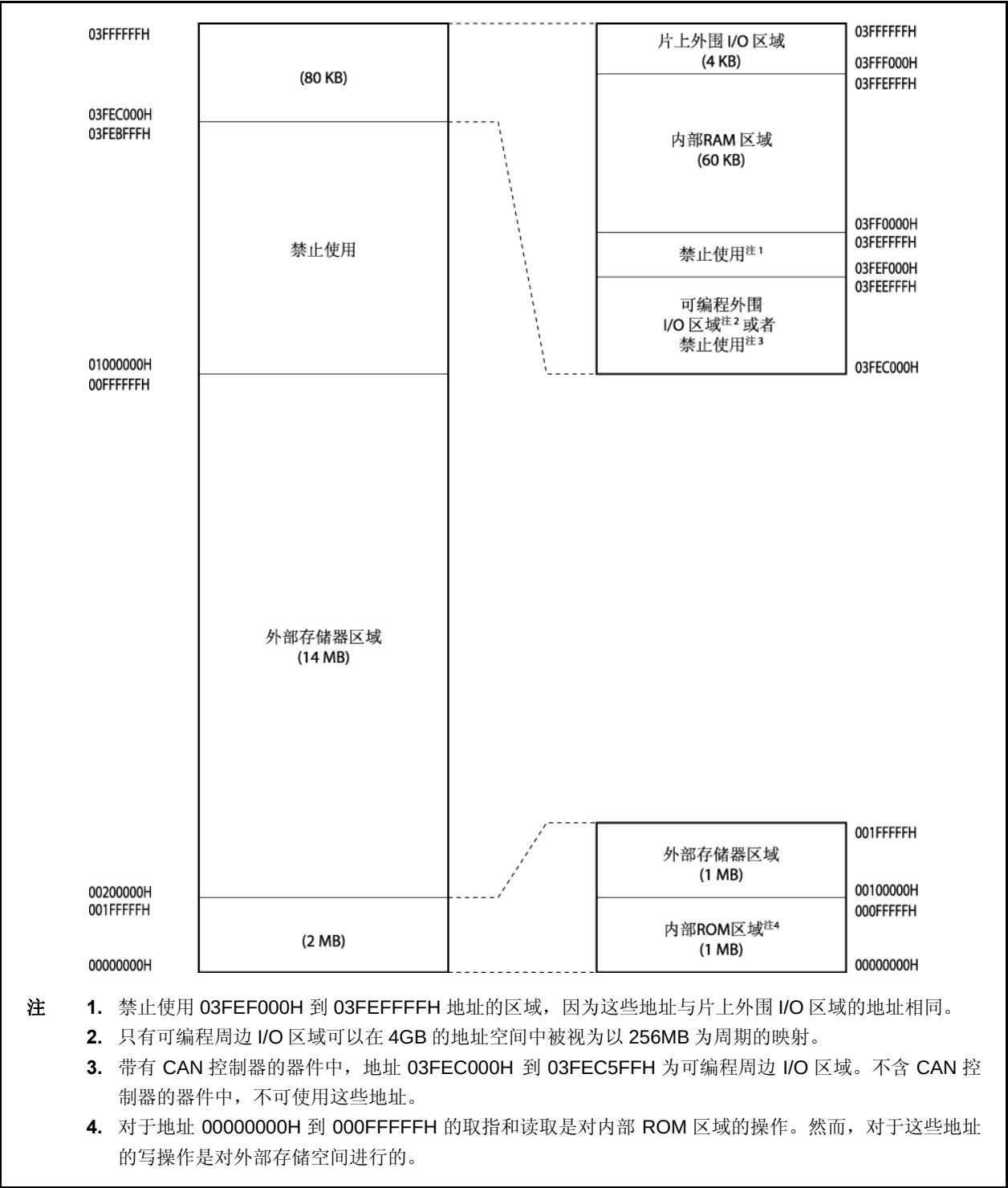
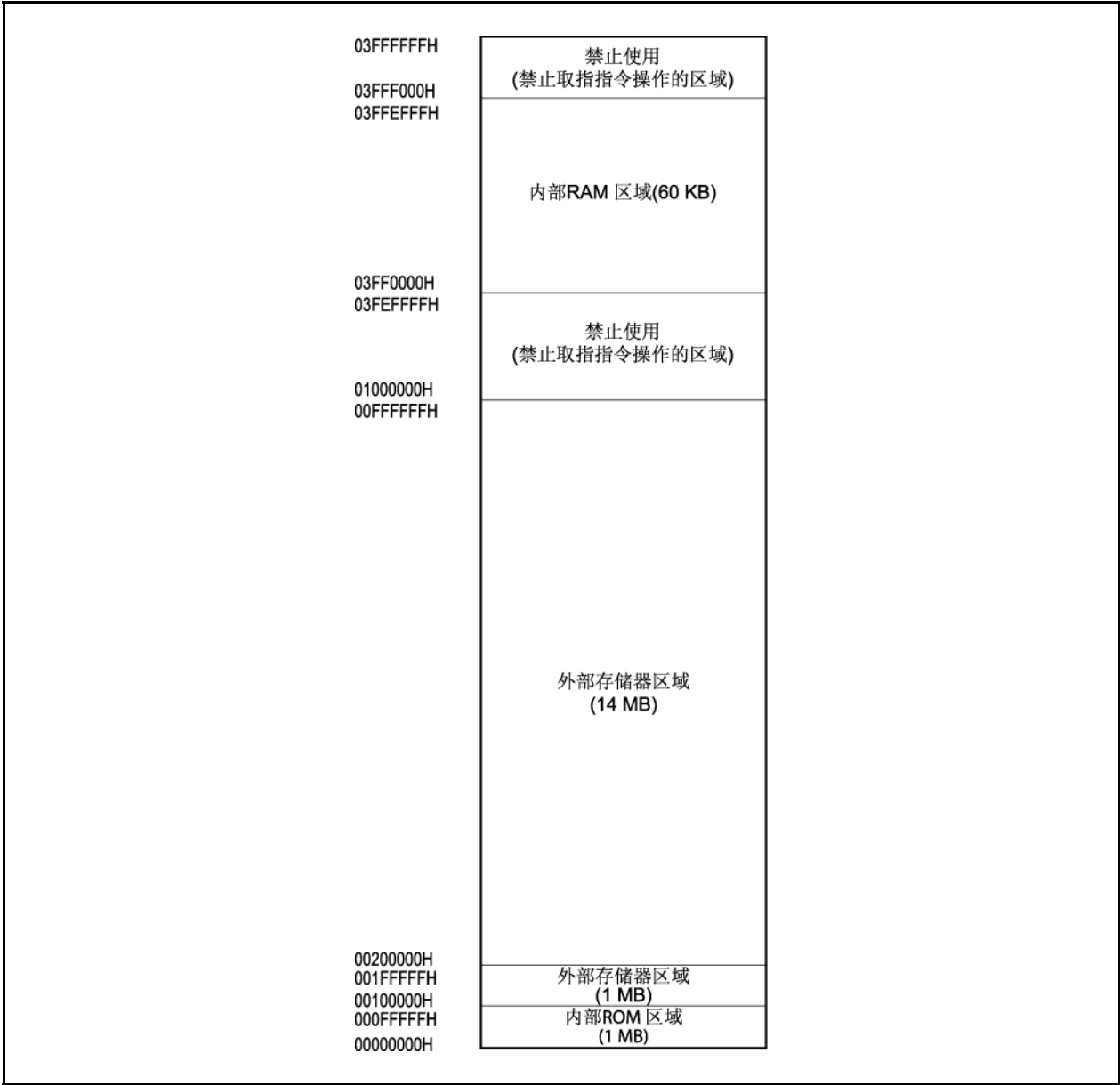


图 3-3. 程序存储器映射



3.4.4 区域

(1) 内部 ROM 区域

最高有 1MB 的空间被保留为内部 ROM 区域。

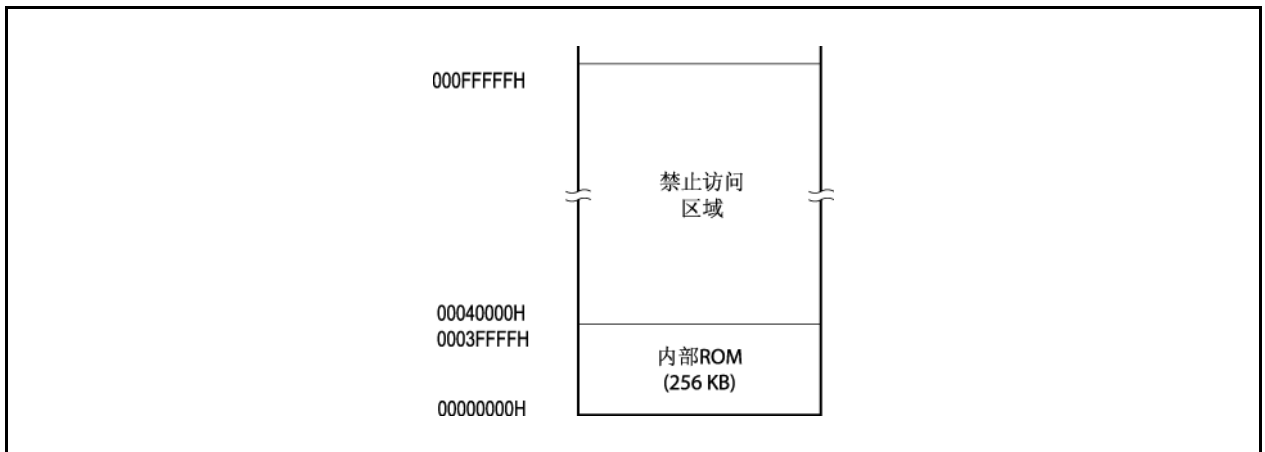
(a) 内部 ROM (256 KB)

以下版本的芯片的 256KB Mask ROM 分配在地址 00000000H 到 0003FFFFH。

对地址 00040000H 到 000FFFFFFH 的操作是被禁止的。

- μ PD703260, 703260Y, 703270, 703270Y, 703280, 703280Y

图 3-4. 内部 ROM 区域(256 KB)



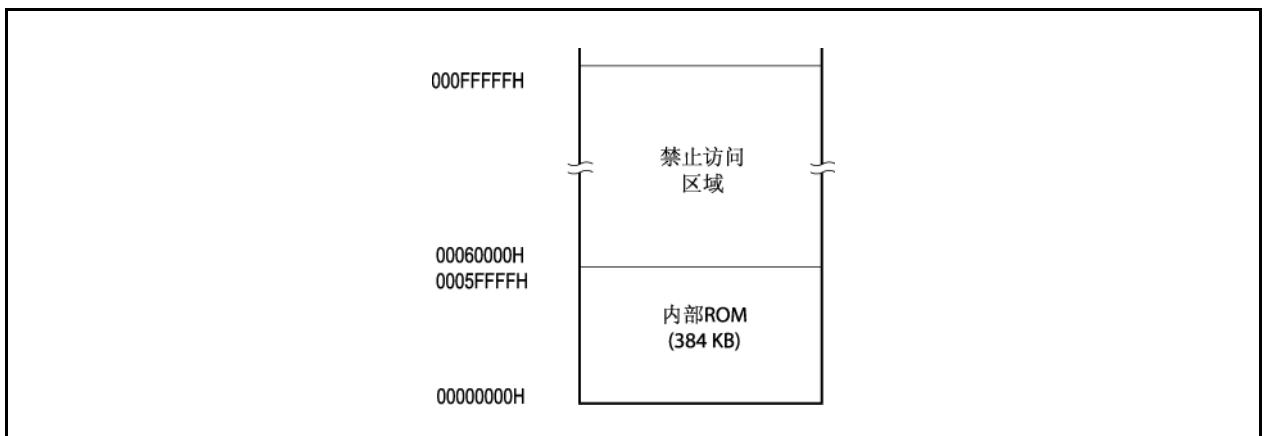
(b) 内部 ROM (384 KB)

以下版本的芯片的 384KB Mask ROM 或 Flash 存储器分配在地址 00000000H 到 0005FFFFH。

对地址 00600000H 到 000FFFFFFH 的操作是被禁止的。

- μ PD703261, 703261Y, 703271, 703271Y, 703281, 703281Y, 70F3261, 70F3261Y, 70F3271, 70F3271Y, 70F3281, 70F3281Y

图 3-5. 内部 ROM 区域(384 KB)



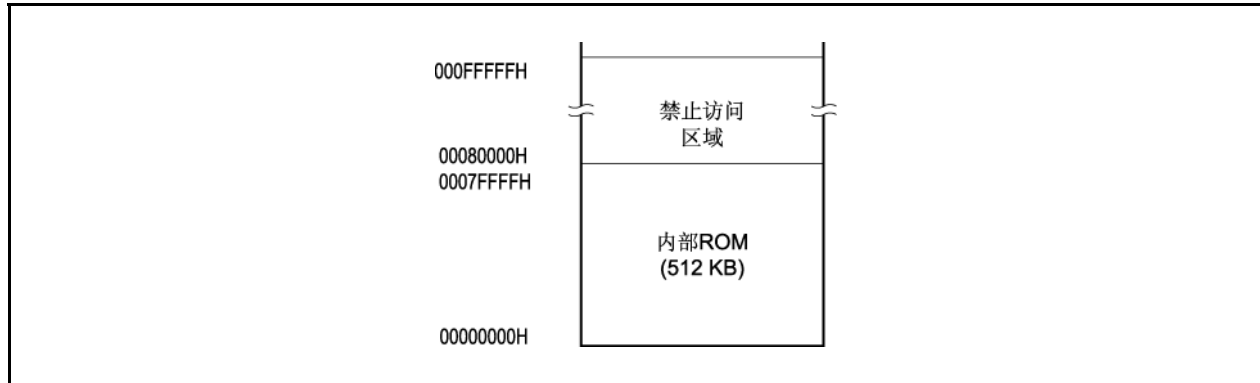
(c) 内部 ROM (512 KB)

以下版本的芯片的 512KB Mask ROM 分配在地址 00000000H 到 0007FFFFH。
对地址 00080000H 到 000FFFFFFH 的操作是被禁止的。

<R>

- μ PD703262, 703262Y, 703272, 703272Y, 703282, 703282Y, 703262HY, 703272HY, 703282HY

图 3-6. 内部 ROM 区域(512 KB)

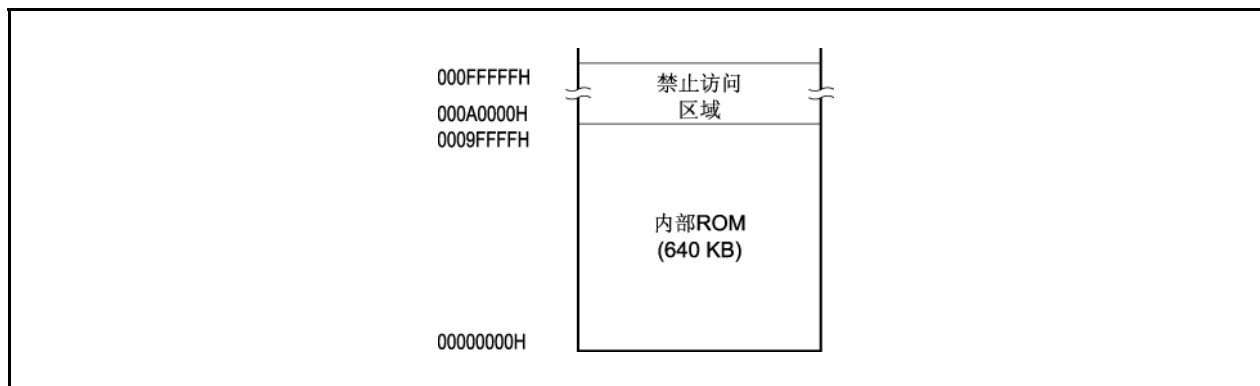
**(d) 内部 ROM (640 KB)**

以下版本的芯片的 640KB Mask ROM 或 Flash 存储器分配在地址 00000000H 到 0009FFFFH。
对地址 000A0000H 到 000FFFFFFH 的操作是被禁止的。

<R>

- μ PD703263, 703263Y, 703273, 703273Y, 703283, 703283Y, 70F3263, 70F3263Y, 70F3273, 70F3273Y, 70F3283, 70F3283Y, 703263HY, 703273HY, 703283HY, 70F3263HY, 70F3273HY, 70F3283HY

图 3-7. 内部 ROM 区域(640 KB)



(2) 内部 RAM 区域

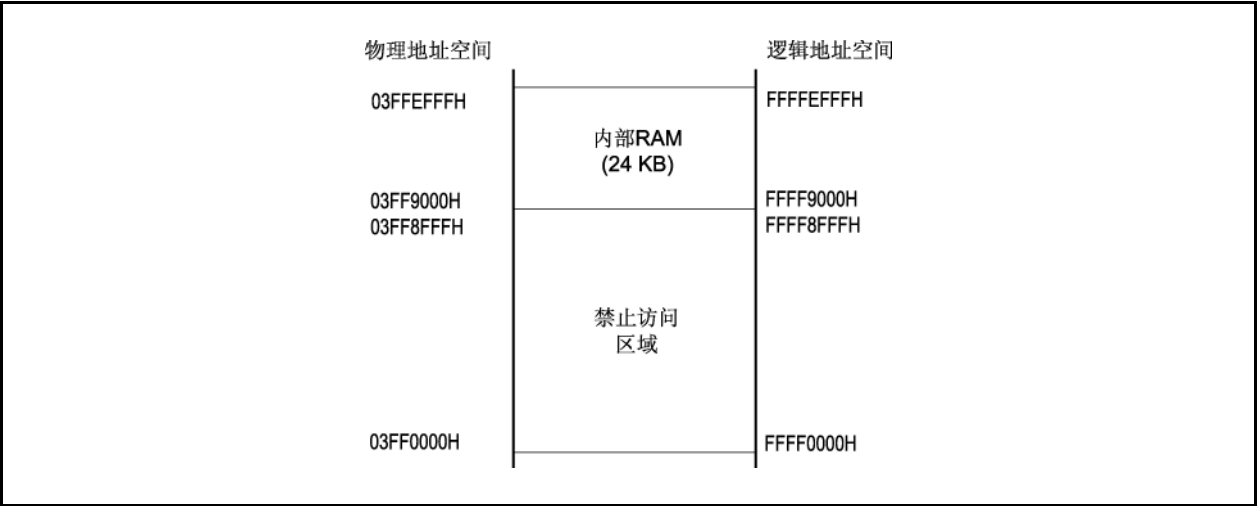
最多有 60KB 的空间被保留为内部 RAM 区域。

(a) 内部 RAM 区域 (24 KB)

以下版本的芯片的 24 KB RAM 分配在地址 03FF9000H 到 03FFFFFFFFH。
禁止对地址 03FF0000H 到 03FF8FFFFH 进行操作。

- μ PD703260, 703260Y, 703270, 703270Y, 703280, 703280Y

图 3-8. 内部 RAM 区域 (24 KB)

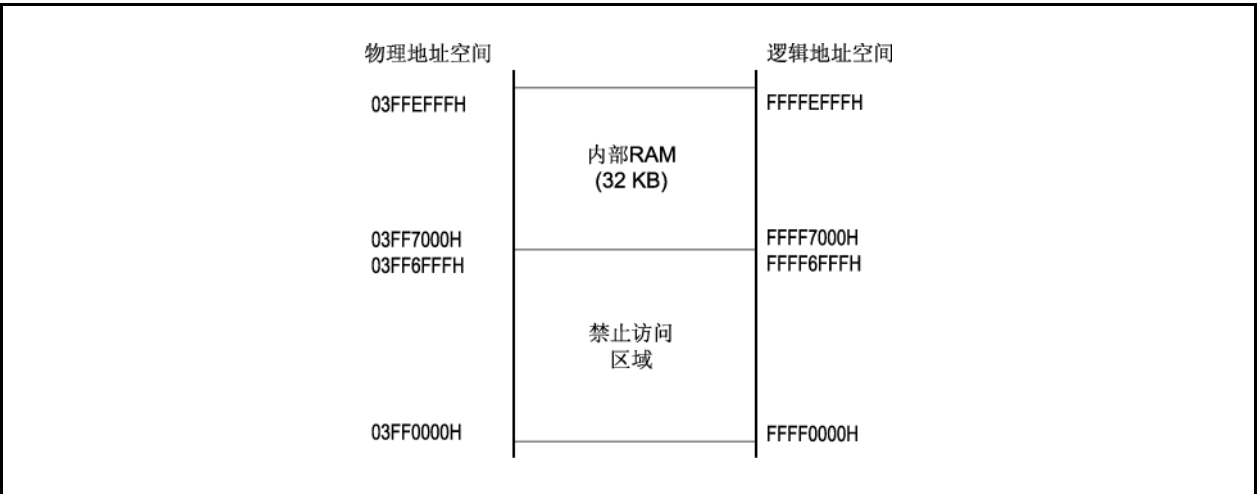


(b) 内部 RAM 区域 (32 KB)

以下版本的芯片的 32 KB RAM 分配在地址 03FF7000H 到 03FFFFFFFFH。
禁止对地址 03FF0000H 到 03FF6FFFFH 进行操作。

- μ PD703261, 703261Y, 703271, 703271Y, 703281, 703281Y, 70F3261, 70F3261Y, 70F3271, 70F3271Y, 70F3281, 70F3281Y

图 3-9. 内部 RAM 区域 (32 KB)



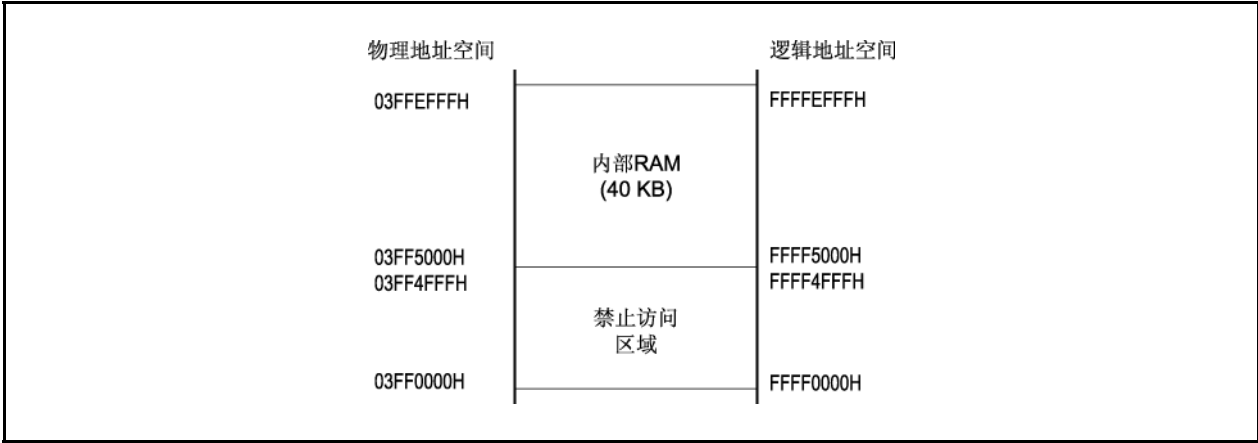
(c) 内部 RAM 区域 (40 KB)

以下版本的芯片的 40 KB RAM 分配在地址 03FF5000H 到 03FFFFFFH。
禁止对地址 03FF0000H 到 03FF4FFFH 进行操作。

<R>

- μ PD703262, 703262Y, 703272, 703272Y, 703282, 703282Y, 703262HY, 703272HY, 703282HY

图 3-10. 内部 RAM 区域 (40 KB)



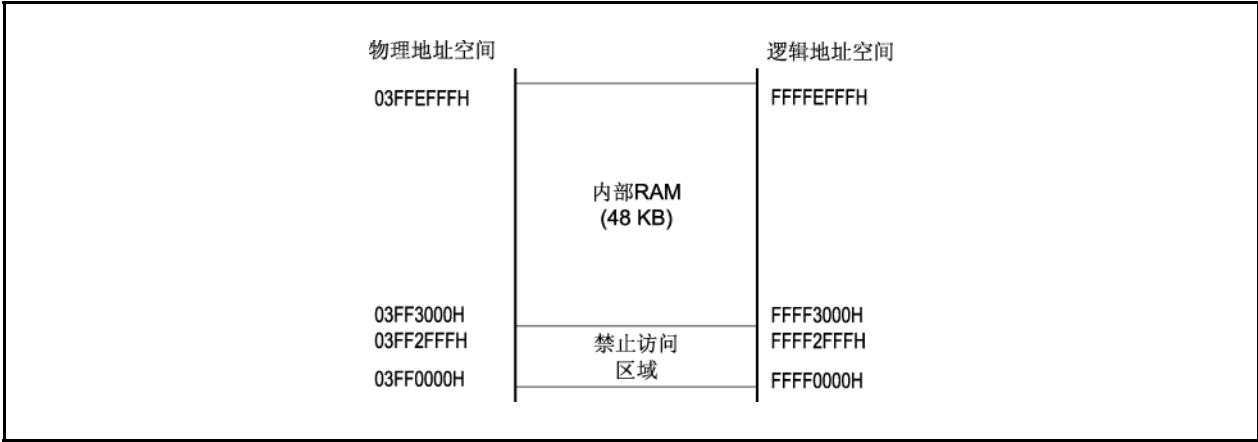
(d) 内部 RAM 区域 (48 KB)

以下版本的芯片的 48KB RAM 分配在地址 03FF3000H 到 03FF7FFFH。
禁止对地址 03FF0000H 到 03FF2FFFH 进行操作。

<R>

- μ 03FF3000PD703263, 703263Y, 703273, 703273Y, 703283, 703283Y, 70F3263, 70F3263Y, 70F3273, 70F3273Y, 70F3283, 70F3283Y, 703263HY, 703273HY, 703283HY, 70F3263HY, 70F3273HY, 70F3283HY

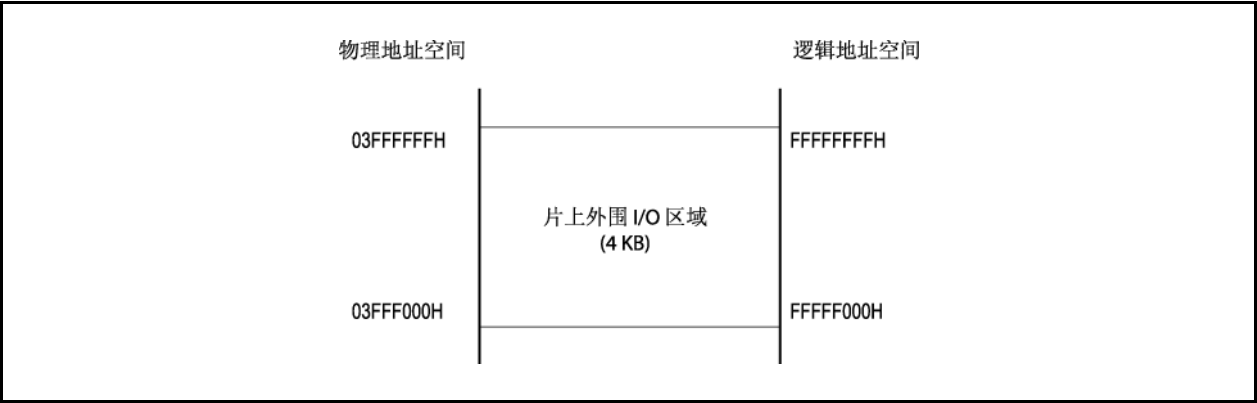
图 3-11. 内部 RAM 区域 (48 KB)



(3) 片上周边 I/O 区域

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

图 3-12. 片上周边 I/O 区域



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

- 注意事项
- 1. 当寄存器被字操作指令读写时，一个字的空间被分为低 16 位半字和高 16 位半字，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。
 - 4. 内置 ROM/RAM 区域和片上外围 I/O 区域被分配到连续的地址上。当通过使用诸如指针操作的增量或减量地址来访问内置 ROM/RAM 时，需小心以免错误地访问到遍布于内置 ROM/RAM 区域边界上的片上外围 I/O 区域。

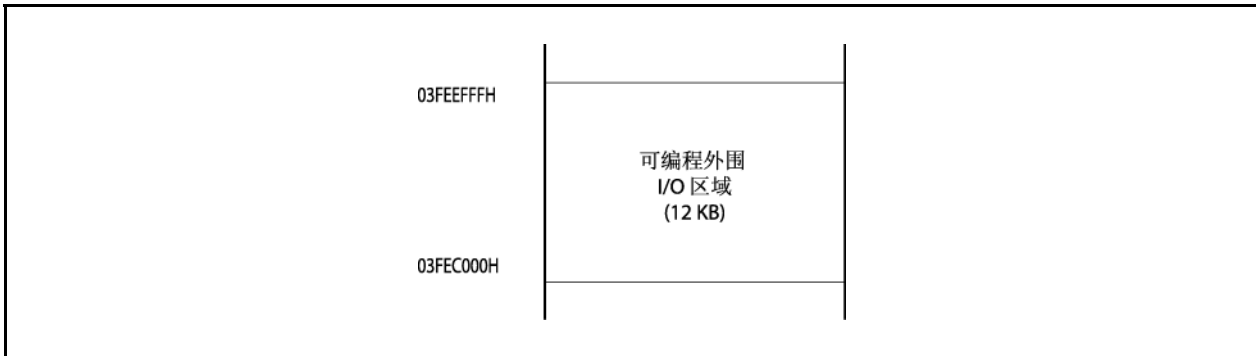
<R>

(4) 可编程周边 I/O 区域

- 注意事项**
1. 只有含有在 **CAN** 控制器的器件中才存在可编程周边 I/O 区域。不含 **CAN** 控制器的器件无法使用该区域。
 2. 只有可编程周边 I/O 区域可以在 **4GB** 的地址空间中被视为以 **256MB** 为周期的映射。

地址 **03FEC000H** 到 **03FEEFFFFH** 的 **12KB** 的空间被保留为可编程周边 I/O 区域。

图 3-13. 可编程周边 I/O 区域

**(5) 外部存储器区域**

外部存储区域是地址 **0100000H** 到 **0FFFFFFH** 的 **15MB** 空间。详见 **第 5 章 总线控制功能**。

- <R> **注意事项** **V850ES/SG2** 和 **V850ES/SG2-H** 有 **22** 根地址线 (**A0** 到 **A21**)，所以外部存储区域也是类似的 **4MB** 映射。当使用 **A20** 和 **A21** 管脚时，必须保证设置 $EV_{DD} = BV_{DD} = V_{DD}$ 。

3.4.5 地址空间的推荐使用方法

<R> V850ES/ SG2 和 V850ES/SG2-H 的架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，高 6 位固定为 0，只有低 26 位有效。考虑程序空间，64MB 的连续地址空间开始于 00000000H，无条件对应存储器映射。
可使用以下地址的内部 RAM 区间作为程序空间。

注意事项 如果分支指令在内部 RAM 区间的上沿地址处发生，那么跨越片上外围 I/O 区间的预取址操作(无效取址)将不会产生。

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

(2) 数据空间

<R> V850ES/SG2 与 V850ES/SG2-H 在 4GB 的 CPU 地址空间上有 64 块 64MB 大小的地址空间。所以，26 位地址中的最高位（位 25）符号扩展到 32 位，然后来定位一个地址。

(a) 绕回式内存空间的应用举例

如果 R=r0(零寄存器)被指定用于 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，它可以有效地减少专用于指针的寄存器需求量。

例如 μ PD703261Y

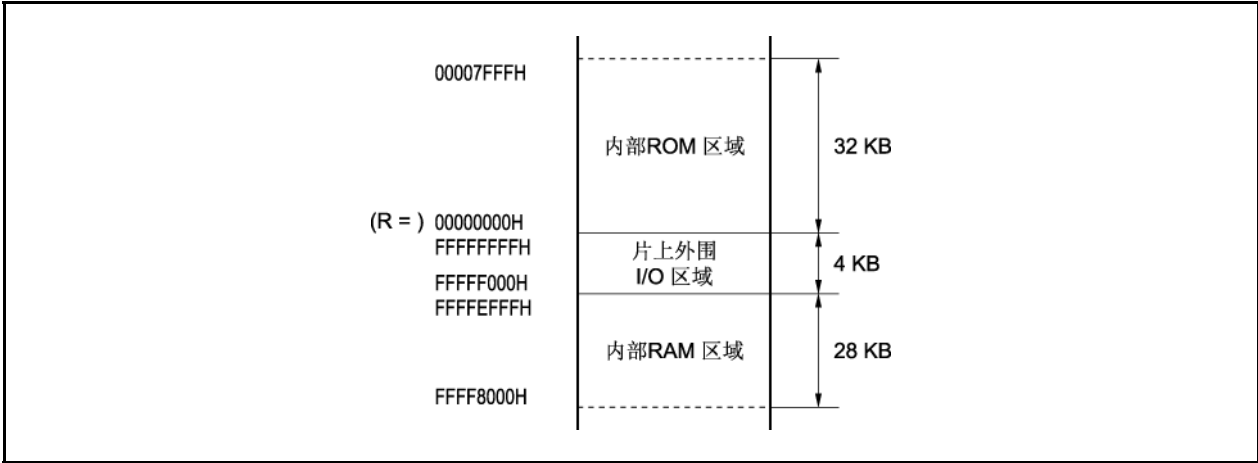
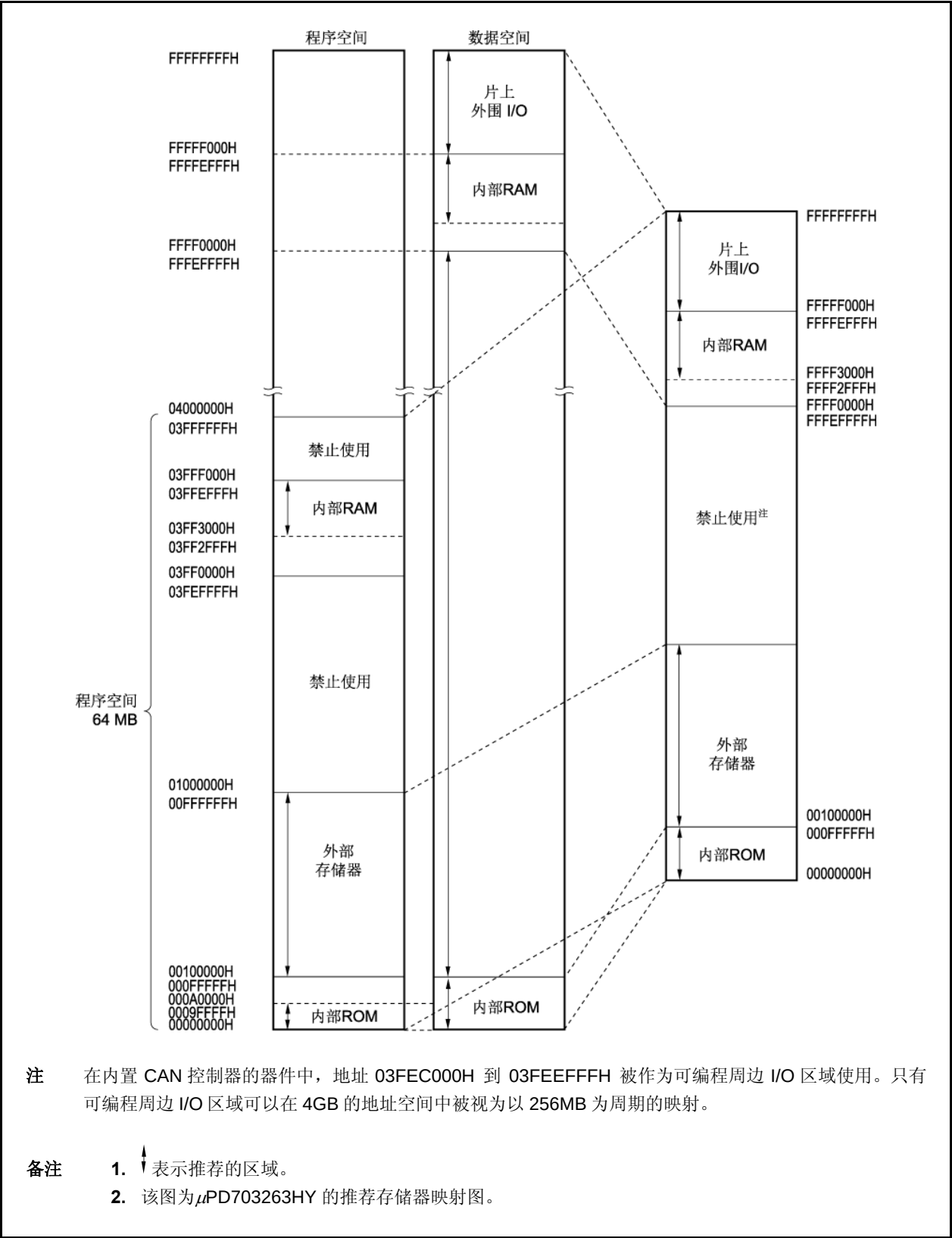


图 3-14. 推荐的存储器映射



3.4.6 周边 I/O 寄存器

(1/11)

地址	寄存器功能名	符号	R/W	可使用操作位数			默认值
				1	8	16	
FFFFF004H	端口 DL 寄存器	PDL	R/W			√	0000H ^{注1}
FFFFF004H	端口 DL 寄存器 L	PDLL		√	√		00H ^{注1}
FFFFF005H	端口 DL 寄存器 H	PDLH		√	√		00H ^{注1}
FFFFF006H	端口 DH 寄存器	PDH		√	√		00H ^{注1}
FFFFF00AH	端口 CT 寄存器	PCT		√	√		00H ^{注1}
FFFFF00CH	端口 CM 寄存器	PCM		√	√		00H ^{注1}
FFFFF024H	端口 DL 模式寄存器	PMDL				√	FFFFH
FFFFF024H	端口 DL 模式寄存器 L	PMDLL		√	√		FFH
FFFFF025H	端口 DL 模式寄存器 H	PMDLH		√	√		FFH
FFFFF026H	端口 DH 模式寄存器	PMDH		√	√		FFH
FFFFF02AH	端口 CT 模式寄存器	PMCT		√	√		FFH
FFFFF02CH	端口 CM 模式寄存器	PMCM		√	√		FFH
FFFFF044H	端口 DL 模式控制寄存器	PMCDL				√	0000H
FFFFF044H	端口 DL 模式控制寄存器 L	PMCDLL		√	√		00H
FFFFF045H	端口 DL 模式控制寄存器 H	PMCDLH		√	√		00H
FFFFF046H	端口 DH 模式控制寄存器	PMCDH		√	√		00H
FFFFF04AH	端口 CT 模式控制寄存器	PM CCT		√	√		00H
FFFFF04CH	端口 CM 模式控制寄存器	PMCCM		√	√		00H
FFFFF064H	周边 I/O 区域选择控制寄存器	BPC ^{注2}				√	0000H
FFFFF066H	总线宽度设置寄存器	BSC				√	5555H
FFFFF06EH	系统等待控制寄存器	VSWC			√		77H
FFFFF080H	DMA 源地址寄存器 0L	DSA0L				√	未定义
FFFFF082H	DMA 源地址寄存器 0H	DSA0H				√	未定义
FFFFF084H	DMA 目的地址寄存器 0L	DDA0L				√	未定义
FFFFF086H	DMA 目的地址寄存器 0H	DDA0H				√	未定义
FFFFF088H	DMA 源地址寄存器 1L	DSA1L				√	未定义
FFFFF08AH	DMA 源地址寄存器 1H	DSA1H				√	未定义
FFFFF08CH	DMA 目的地址寄存器 1L	DDA1L				√	未定义
FFFFF08EH	DMA 目的地址寄存器 1H	DDA1H				√	未定义
FFFFF090H	DMA 源地址寄存器 2L	DSA2L				√	未定义
FFFFF092H	DMA 源地址寄存器 2H	DSA2H				√	未定义
FFFFF094H	DMA 目的地址寄存器 2L	DDA2L				√	未定义
FFFFF096H	DMA 目的地址寄存器 2H	DDA2H				√	未定义
FFFFF098H	DMA 源地址寄存器 3L	DSA3L				√	未定义
FFFFF09AH	DMA 源地址寄存器 3H	DSA3H				√	未定义
FFFFF09CH	DMA 目的地址寄存器 3L	DDA3L				√	未定义
FFFFF09EH	DMA 目的地址寄存器 3H	DDA3H				√	未定义
FFFFF0C0H	DMA 传输计数寄存器 0	DBC0				√	未定义
FFFFF0C2H	DMA 传输计数寄存器 1	DBC1				√	未定义
FFFFF0C4H	DMA 传输计数寄存器 2	DBC2				√	未定义

注 1. 输出所存是 00H 或者 0000H。当这些寄存器处于输入模式时，管脚的状态可读。

2. 仅限内置 CAN 控制器的版本

(2/11)

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	
FFFFF0C6H	DMA 传输计数寄存器 3	DBC3	R/W			√	未定义
FFFFF0D0H	DMA 寻址控制寄存器 0	DADC0				√	0000H
FFFFF0D2H	DMA 寻址控制寄存器 1	DADC1				√	0000H
FFFFF0D4H	DMA 寻址控制寄存器 2	DADC2				√	0000H
FFFFF0D6H	DMA 寻址控制寄存器 3	DADC3				√	0000H
FFFFF0E0H	DMA 通道控制寄存器 0	DCHC0		√	√		00H
FFFFF0E2H	DMA 通道控制寄存器 1	DCHC1		√	√		00H
FFFFF0E4H	DMA 通道控制寄存器 2	DCHC2		√	√		00H
FFFFF0E6H	DMA 通道控制寄存器 3	DCHC3		√	√		00H
FFFFF100H	中断屏蔽寄存器 0	IMR0				√	FFFFH
FFFFF100H	中断屏蔽寄存器 0L	IMR0L		√	√		FFH
FFFFF101H	中断屏蔽寄存器 0H	IMR0H		√	√		FFH
FFFFF102H	中断屏蔽寄存器 1	IMR1				√	FFFFH
FFFFF102H	中断屏蔽寄存器 1L	IMR1L		√	√		FFH
FFFFF103H	中断屏蔽寄存器 1H	IMR1H		√	√		FFH
FFFFF104H	中断屏蔽寄存器 2	IMR2				√	FFFFH
FFFFF104H	中断屏蔽寄存器 2L	IMR2L		√	√		FFH
FFFFF105H	中断屏蔽寄存器 2H	IMR2H		√	√		FFH
FFFFF106H	中断屏蔽寄存器 3	IMR3				√	FFFFH
FFFFF106H	中断屏蔽寄存器 3L	IMR3L		√	√		FFH
FFFFF107H	中断屏蔽寄存器 3H	IMR3H		√	√		FFH
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

<R>

<R>

注 只对 V850ES/SG2 版本。

(3/11)

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	
FFFFF13AH	中断控制寄存器	TP2CCIC0	R/W	√	√		47H
FFFFF13CH	中断控制寄存器	TP2CCIC1		√	√		47H
FFFFF13EH	中断控制寄存器	TP3OVIC		√	√		47H
FFFFF140H	中断控制寄存器	TP3CCIC0		√	√		47H
FFFFF142H	中断控制寄存器	TP3CCIC1		√	√		47H
FFFFF144H	中断控制寄存器	TP4OVIC		√	√		47H
FFFFF146H	中断控制寄存器	TP4CCIC0		√	√		47H
FFFFF148H	中断控制寄存器	TP4CCIC1		√	√		47H
FFFFF14AH	中断控制寄存器	TP5OVIC		√	√		47H
FFFFF14CH	中断控制寄存器	TP5CCIC0		√	√		47H
FFFFF14EH	中断控制寄存器	TP5CCIC1		√	√		47H
FFFFF150H	中断控制寄存器	TM0EQIC0		√	√		47H
FFFFF152H	中断控制寄存器	CB0RIC/IICIC1 ^注		√	√		47H
FFFFF154H	中断控制寄存器	CB0TIC		√	√		47H
FFFFF156H	中断控制寄存器	CB1RIC		√	√		47H
FFFFF158H	中断控制寄存器	CB1TIC		√	√		47H
FFFFF15AH	中断控制寄存器	CB2RIC		√	√		47H
FFFFF15CH	中断控制寄存器	CB2TIC		√	√		47H
FFFFF15EH	中断控制寄存器	CB3RIC		√	√		47H
FFFFF160H	中断控制寄存器	CB3TIC		√	√		47H
FFFFF162H	中断控制寄存器	UA0RIC/CB4RIC		√	√		47H
FFFFF164H	中断控制寄存器	UA0TIC/CB4TIC		√	√		47H
FFFFF166H	中断控制寄存器	UA1RIC/IICIC2 ^注		√	√		47H
FFFFF168H	中断控制寄存器	UA1TIC		√	√		47H
FFFFF16AH	中断控制寄存器	UA2RIC/IICIC0 ^注		√	√		47H
FFFFF16CH	中断控制寄存器	UA2TIC		√	√		47H
FFFFF16EH	中断控制寄存器	ADIC		√	√		47H
FFFFF170H	中断控制寄存器	DMAIC0		√	√		47H
FFFFF172H	中断控制寄存器	DMAIC1		√	√		47H
FFFFF174H	中断控制寄存器	DMAIC2		√	√		47H
FFFFF176H	中断控制寄存器	DMAIC3		√	√		47H
FFFFF178H	中断控制寄存器	KRIC		√	√		47H
FFFFF17AH	中断控制寄存器	WTIIC		√	√		47H
FFFFF17CH	中断控制寄存器	WTIC		√	√		47H
FFFFF17EH	中断控制寄存器	ERRIC0 ^{注2/}		√	√		47H
FFFFF180H	中断控制寄存器	WUPIC0 ^{注2/}		√	√		47H
FFFFF182H	中断控制寄存器	ERRIC0 ^{注2/}		√	√		47H

- 注
1. 仅限内置 I²C 总线版本 (Y 产品)
 2. 仅限内置 CAN 控制器的版本
 3. 仅限内置 IEBus 控制器的版本

(4/11)

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	
FFFFF184H	中断控制寄存器	TRXIC0 ^{※1/}	R/W	√	√		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	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
FFFFF220H	A/D 转换器结果寄存器 8	ADA0CR8				√	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
FFFFF224H	A/D 转换器结果寄存器 10	ADA0CR10				√	未定义
FFFFF225H	A/D 转换器结果寄存器 10H	ADA0CR10H			√		未定义
FFFFF226H	A/D 转换器结果寄存器 11	ADA0CR11				√	未定义
FFFFF227H	A/D 转换器结果寄存器 11H	ADA0CR11H			√		未定义
FFFFF280H	D/A 转换器转换值设置寄存器 0	DA0CS0	R/W		√		00H
FFFFF281H	D/A 转换器转换值设置寄存器 1	DA0CS1			√		00H
FFFFF282H	D/A 转换器模式寄存器	DA0M		√	√		00H
FFFFF300H	按键中断模式寄存器	KRM		√	√		00H
FFFFF308H	选择操作控制寄存器	SELCNT0		√	√		00H
FFFFF310H	CRC 输入寄存器	CRCIN			√		00H

- 注
1. 仅限内置 CAN 控制器的版本
 2. 仅限内置 IEBus 控制器的版本

(5/11)

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	
FFFFF312H	CRC 数据寄存器	CRCD	R/W			√	0000H
FFFFF318H	噪声消除控制寄存器	NFC			√		00H
FFFFF320H	BRG1 预分频器模式寄存器	PRSM1		√	√		00H
FFFFF321H	BRG1 预分频器比较寄存器	PRSCM1			√		00H
FFFFF324H	BRG2 预分频器模式寄存器	PRSM2		√	√		00H
FFFFF325H	BRG2 预分频器比较寄存器	PRSCM2			√		00H
FFFFF328H	BRG3 预分频器模式寄存器	PRSM3		√	√		00H
FFFFF329H	BRG3 预分频器比较寄存器	PRSCM3			√		00H
FFFFF340H	IIC 分频时钟选择寄存器	OCKS0 ^{注1}			√		00H
FFFFF344H	IIC 分频时钟选择寄存器	OCKS1 ^{注1}			√		00H
FFFFF348H	IEBus 时钟选择寄存器	OCKS2 ^{注2}			√		00H
FFFFF360H	IEBus 控制寄存器	BCR ^{注2}		√	√		00H
FFFFF361H	IEBus 节能寄存器	PSR ^{注2}		√	√		00H
FFFFF362H	IEBus 从设备状态寄存器	SSR ^{注2}	R	√	√		81H
FFFFF363H	IEBus 单元状态寄存器	USR ^{注2}	R/W	√	√		00H
FFFFF364H	IEBus 中断状态寄存器	ISR ^{注2}		√	√		00H
FFFFF365H	IEBus 错误状态寄存器	ESR ^{注2}		√	√		00H
FFFFF366H	IEBus 单元地址寄存器	UAR ^{注2}				√	0000H
FFFFF368H	IEBus 从设备地址寄存器	SAR ^{注2}				√	0000H
FFFFF36AH	IEBus 伙伴地址寄存器	PAR ^{注2}				√	0000H
FFFFF36CH	IEBus 接收从设备地址寄存器	RSA ^{注2}	R/W			√	0000H
FFFFF36EH	IEBus 控制数据寄存器	CDR ^{注2}			√		00H
FFFFF36FH	IEBus 电报长度寄存器	DLR ^{注2}			√		01H
FFFFF370H	IEBus 数据寄存器	DR ^{注2}			√		00H
FFFFF371H	IEBus 场状态寄存器	FSR ^{注2}			√		00H
FFFFF372H	IEBus 成功计数寄存器	SCR ^{注2}			√		01H
FFFFF373H	IEBus 通信计数寄存器	CCR ^{注2}	R/W		√		20H
FFFFF400H	端口 0 寄存器	P0		√	√		00H ^{注3}
FFFFF402H	端口 1 寄存器	P1		√	√		00H ^{注3}
FFFFF406H	端口 3 寄存器	P3				√	0000H ^{注3}
FFFFF406H	端口 3 寄存器 L	P3L		√	√		00H ^{注3}
FFFFF407H	端口 3 寄存器 H	P3H		√	√		00H ^{注3}
FFFFF408H	端口 4 寄存器	P4		√	√		00H ^{注3}
FFFFF40AH	端口 5 寄存器	P5		√	√		00H ^{注3}
FFFFF40EH	端口 7 寄存器 L	P7L		√	√		00H ^{注3}
FFFFF40FH	端口 7 寄存器 H	P7H		√	√		00H ^{注3}
FFFFF412H	端口 9 寄存器	P9				√	0000H ^{注3}
FFFFF412H	端口 9 寄存器 L	P9L		√	√		00H ^{注3}
FFFFF413H	端口 9 寄存器 H	P9H		√	√		00H ^{注3}
FFFFF420H	端口 0 模式寄存器	PM0		√	√		FFH

- 注
1. 仅限内置 I²C 总线版本（Y 产品）
 2. 仅限内置 CAN 控制器的版本
 3. 输出所存是 00H 或者 0000H。当这些寄存器处于输入模式时，管脚的状态可读。

(6/11)

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	
FFFFFF422H	端口 1 模式寄存器	PM1	R/W	√	√		FFH
FFFFFF426H	端口 3 模式寄存器	PM3				√	FFFFH
FFFFFF426H	端口 3 模式寄存器 L	PM3L		√	√		FFH
FFFFFF427H	端口 3 模式寄存器 H	PM3H		√	√		FFH
FFFFFF428H	端口 4 模式寄存器	PM4		√	√		FFH
FFFFFF42AH	端口 5 模式寄存器	PM5		√	√		FFH
FFFFFF42EH	端口 7 模式寄存器 L	PM7L		√	√		FFH
FFFFFF42FH	端口 7 模式寄存器 H	PM7H		√	√		FFH
FFFFFF432H	端口 9 模式寄存器	PM9				√	FFFFH
FFFFFF432H	端口 9 模式寄存器 L	PM9L		√	√		FFH
FFFFFF433H	端口 9 模式寄存器 H	PM9H		√	√		FFH
FFFFFF440H	端口 0 模式控制寄存器	PMC0		√	√		00H
FFFFFF446H	端口 3 模式控制寄存器	PMC3				√	0000H
FFFFFF446H	端口 3 模式控制寄存器 L	PMC3L		√	√		00H
FFFFFF447H	端口 3 模式控制寄存器 H	PMC3H		√	√		00H
FFFFFF448H	端口 4 模式控制寄存器	PMC4		√	√		00H
FFFFFF44AH	端口 5 模式控制寄存器	PMC5		√	√		00H
FFFFFF452H	端口 9 模式控制寄存器	PMC9				√	0000H
FFFFFF452H	端口 9 模式控制寄存器 L	PMC9L		√	√		00H
FFFFFF453H	端口 9 模式控制寄存器 H	PMC9H		√	√		00H
FFFFFF460H	端口 0 功能控制寄存器	PFC0		√	√		00H
FFFFFF466H	端口 3 功能控制寄存器	PFC3				√	0000H
FFFFFF466H	端口 3 功能控制寄存器 L	PFC3L		√	√		00H
FFFFFF467H	端口 3 功能控制寄存器 H	PFC3H		√	√		00H
FFFFFF468H	端口 4 功能控制寄存器	PFC4		√	√		00H
FFFFFF46AH	端口 5 功能控制寄存器	PFC5		√	√		00H
FFFFFF472H	端口 9 功能控制寄存器	PFC9				√	0000H
FFFFFF472H	端口 9 功能控制寄存器 L	PFC9L		√	√		00H
FFFFFF473H	端口 9 功能控制寄存器 H	PFC9H		√	√		00H
FFFFFF484H	数据等待控制寄存器 0	DWC0				√	7777H
FFFFFF488H	地址等待控制寄存器	AWC				√	FFFFH
FFFFFF48AH	总线周期控制寄存器	BCC				√	AAAAH
FFFFFF540H	TMQ0 控制寄存器 0	TQ0CTL0		√	√		00H
FFFFFF541H	TMQ0 控制寄存器 1	TQ0CTL1		√	√		00H
FFFFFF542H	TMQ0 I/O 控制寄存器 0	TQ0IOC0		√	√		00H
FFFFFF543H	TMQ0 I/O 控制寄存器 1	TQ0IOC1		√	√		00H
FFFFFF544H	TMQ0 I/O 控制寄存器 2	TQ0IOC2		√	√		00H
FFFFFF545H	TMQ0 选项寄存器 0	TQ0OPT0		√	√		00H
FFFFFF546H	TMQ0 捕捉 / 比较寄存器 0	TQ0CCR0				√	0000H
FFFFFF548H	TMQ0 捕捉 / 比较寄存器 1	TQ0CCR1				√	0000H
FFFFFF54AH	TMQ0 捕捉 / 比较寄存器 2	TQ0CCR2				√	0000H
FFFFFF54CH	TMQ0 捕捉 / 比较寄存器 3	TQ0CCR3				√	0000H
FFFFFF54EH	TMQ0 计数器读取缓冲寄存器	TQ0CNT	R			√	0000H

(7/11)

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	
FFFFF590H	TMP0 控制寄存器 0	TP0CTL0	R/W	√	√		00H
FFFFF591H	TMP0 控制寄存器 1	TP0CTL1		√	√		00H
FFFFF592H	TMP0I/O 控制寄存器 0	TP0IOC0		√	√		00H
FFFFF593H	TMP0I/O 控制寄存器 1	TP0IOC1		√	√		00H
FFFFF594H	TMP0I/O 控制寄存器 2	TP0IOC2		√	√		00H
FFFFF595H	TMP0 选项寄存器 0	TP0OPT0		√	√		00H
FFFFF596H	TMP0 捕捉 / 比较寄存器 0	TP0CCR0				√	0000H
FFFFF598H	TMP0 捕捉 / 比较寄存器 1	TP0CCR1				√	0000H
FFFFF59AH	TMP0 计数器读取缓冲寄存器	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	TMP1I/O 控制寄存器 0	TP1IOC0		√	√		00H
FFFFF5A3H	TMP1I/O 控制寄存器 1	TP1IOC1		√	√		00H
FFFFF5A4H	TMP1I/O 控制寄存器 2	TP1IOC2		√	√		00H
FFFFF5A5H	TMP1 选项寄存器 0	TP1OPT0		√	√		00H
FFFFF5A6H	TMP1 捕捉 / 比较寄存器 0	TP1CCR0				√	0000H
FFFFF5A8H	TMP1 捕捉 / 比较寄存器 1	TP1CCR1				√	0000H
FFFFF5AAH	TMP1 计数器读取缓冲寄存器	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	TMP2I/O 控制寄存器 0	TP2IOC0		√	√		00H
FFFFF5B3H	TMP2I/O 控制寄存器 1	TP2IOC1		√	√		00H
FFFFF5B4H	TMP2I/O 控制寄存器 2	TP2IOC2		√	√		00H
FFFFF5B5H	TMP2 选项寄存器 0	TP2OPT0		√	√		00H
FFFFF5B6H	TMP2 捕捉 / 比较寄存器 0	TP2CCR0				√	0000H
FFFFF5B8H	TMP2 捕捉 / 比较寄存器 1	TP2CCR1				√	0000H
FFFFF5BAH	TMP2 计数器读取缓冲寄存器	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	TMP3I/O 控制寄存器 0	TP3IOC0		√	√		00H
FFFFF5C3H	TMP3I/O 控制寄存器 1	TP3IOC1		√	√		00H
FFFFF5C4H	TMP3I/O 控制寄存器 2	TP3IOC2		√	√		00H
FFFFF5C5H	TMP3 选项寄存器 0	TP3OPT0		√	√		00H
FFFFF5C6H	TMP3 捕捉 / 比较寄存器 0	TP3CCR0				√	0000H
FFFFF5C8H	TMP3 捕捉 / 比较寄存器 1	TP3CCR1				√	0000H
FFFFF5CAH	TMP3 计数器读取缓冲寄存器	TP3CNT	R			√	0000H
FFFFF5D0H	TMP4 控制寄存器 0	TP4CTL0	R/W	√	√		00H
FFFFF5D1H	TMP4 控制寄存器 1	TP4CTL1		√	√		00H
FFFFF5D2H	TMP4I/O 控制寄存器 0	TP4IOC0		√	√		00H
FFFFF5D3H	TMP4I/O 控制寄存器 1	TP4IOC1		√	√		00H
FFFFF5D4H	TMP4I/O 控制寄存器 2	TP4IOC2		√	√		00H
FFFFF5D5H	TMP4 选项寄存器 0	TP4OPT0		√	√		00H
FFFFF5D6H	TMP4 捕捉/比较寄存器 0	TP4CCR0				√	0000H
FFFFF5D8H	TMP4 捕捉/比较寄存器 1	TP4CCR1				√	0000H

(8/11)

地址	寄存器功能名	符号	R/W	可使用操作位数				默认值
				1	8	16	32	
FFFFF5DAH	TMP4 计数器读取缓冲 寄存器	TP4CNT	R			√		0000H
FFFFF5E0H	TMP5 控制寄存器 0	TP5CTL0	R/W	√	√			00H
FFFFF5E1H	TMP5 控制寄存器 1	TP5CTL1		√	√			00H
FFFFF5E2H	TMP5I/O 控制寄存器 0	TP5IOC0		√	√			00H
FFFFF5E3H	TMP5I/O 控制寄存器 1	TP5IOC1		√	√			00H
FFFFF5E4H	TMP5I/O 控制寄存器 2	TP5IOC2		√	√			00H
FFFFF5E5H	TMP5 选项寄存器 0	TP5OPT0		√	√			00H
FFFFF5E6H	TMP5 捕捉 / 比较寄存器 0	TP5CCR0				√		0000H
FFFFF5E8H	TMP5 捕捉 / 比较寄存器 1	TP5CCR1				√		0000H
FFFFF5EAH	TMP5 计数器读取缓冲寄存器	TP5CNT	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
FFFFF6E0H	实时输出缓冲寄存器 0L	RTBL0		√	√			00H
FFFFF6E2H	实时输出缓冲寄存器 0H	RTBH0		√	√			00H
FFFFF6E4H	实时输出端口模式寄存器 0	RTPM0		√	√			00H
FFFFF6E5H	实时输出端口控制寄存器 0	RTPC0		√	√			00H
FFFFF706H	端口 3 功能控制扩展寄存器 L	PFCE3L		√	√			00H
FFFFF70AH	端口 5 功能控制扩展寄存器	PFCE5		√	√			00H
FFFFF712H	端口 9 功能控制扩展寄存器	PFCE9				√		0000H
FFFFF712H	端口 9 功能控制扩展寄存器 L	PFCE9L		√	√			00H
FFFFF713H	端口 9 功能控制扩展寄存器 H	PFCE9H		√	√			00H
FFFFF802H	系统状态寄存器	SYS		√	√			00H
FFFFF80CH	内部振荡模式寄存器	RCM		√	√			00H
FFFFF810H	DMA 触发源寄存器 0	DTFR0		√	√			00H
FFFFF812H	DMA 触发源寄存器 1	DTFR1		√	√			00H
FFFFF814H	DMA 触发源寄存器 2	DTFR2		√	√			00H
FFFFF816H	DMA 触发源寄存器 3	DTFR3		√	√			00H
FFFFF820H	节电模式寄存器	PSMR		√	√			00H
FFFFF822H	时钟控制寄存器	CKC		√	√			0AH
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
FFFFF840H	修正地址寄存器 0	CORAD0	R/W				√	00000000H
FFFFF840H	修正地址寄存器 0L	CORAD0L				√		0000H
FFFFF842H	修正地址寄存器 0H	CORAD0H				√		0000H

(9/11)

	地址	寄存器功能名	符号	R/W	可使用操作位数				默认值
					1	8	16	32	
<R>	FFFFF844H	修正地址寄存器 1	CORAD1	R/W				√	00000000H
	FFFFF844H	修正地址寄存器 1L	CORAD1L				√		0000H
	FFFFF846H	修正地址寄存器 1H	CORAD1H				√		0000H
<R>	FFFFF848H	修正地址寄存器 2	CORAD2	R/W				√	00000000H
	FFFFF848H	修正地址寄存器 2L	CORAD2L				√		0000H
	FFFFF84AH	修正地址寄存器 2H	CORAD2H				√		0000H
<R>	FFFFF84CH	修正地址寄存器 3	CORAD3	R/W				√	00000000H
	FFFFF84CH	修正地址寄存器 3L	CORAD3L				√		0000H
	FFFFF84EH	修正地址寄存器 3H	CORAD3H				√		0000H
	FFFFF870H	时钟监视器模式寄存器	CLM		√	√			00H
	FFFFF880H	修正控制寄存器	CORCN		√	√			00H
	FFFFF888H	复位源标志寄存器	RESF		√	√			00H
	FFFFF890H	低电压检测寄存器	LVIM ^{※1}		√	√			00H
	FFFFF891H	低电压检测电压选择寄存器	LVIS ^{※1}			√			00H
	FFFFF892H	内部 RAM 数据状态寄存器	RAMS		√	√			01H
	FFFFF8B0H	预分频模式寄存器	PRSM0		√	√			00H
	FFFFF8B1H	预分频器比较寄存器	PRSCM0			√			00H
	FFFFF9FCH	片上调试模式寄存器	OCDM		√	√			01H
	FFFFF9FEH	周边仿真寄存器 1	PEMU1 ^{※2}		√	√			00H
	FFFFFA00H	UARTA0 控制寄存器 0	UA0CTL0		√	√			10H
	FFFFFA01H	UARTA0 控制寄存器 1	UA0CTL1			√			00H
	FFFFFA02H	UARTA0 控制寄存器 2	UA0CTL2			√			FFH
	FFFFFA03H	UARTA0 选项控制寄存器 0	UA0OPT0		√	√			14H
	FFFFFA04H	UARTA0 状态寄存器	UA0STR		√	√			00H
	FFFFFA06H	UARTA0 接收数据寄存器	UA0RX	R		√			FFH
	FFFFFA07H	UARTA0 发送数据寄存器	UA0TX	R/W		√			FFH
	FFFFFA10H	UARTA1 控制寄存器 0	UA1CTL0		√	√			10H
	FFFFFA11H	UARTA1 控制寄存器 1	UA1CTL1			√			00H
	FFFFFA12H	UARTA1 控制寄存器 2	UA1CTL2			√			FFH
	FFFFFA13H	UARTA1 选项控制寄存器 0	UA1OPT0		√	√			14H
	FFFFFA14H	UARTA1 状态寄存器	UA1STR		√	√			00H
	FFFFFA16H	UARTA1 接收数据寄存器	UA1RX	R		√			FFH
	FFFFFA17H	UARTA1 发送数据寄存器	UA1TX	R/W		√			FFH
	FFFFFA20H	UARTA2 控制寄存器 0	UA2CTL0		√	√			10H
	FFFFFA21H	UARTA2 控制寄存器 1	UA2CTL1			√			00H
	FFFFFA22H	UARTA2 控制寄存器 2	UA2CTL2			√			FFH
	FFFFFA23H	UARTA2 选项控制寄存器 0	UA2OPT0		√	√			14H
	FFFFFA24H	UARTA2 状态寄存器	UA2STR		√	√			00H
	FFFFFA26H	UARTA2 接收数据寄存器	UA2RX	R		√			FFH
	FFFFFA27H	UARTA2 发送数据寄存器	UA2TX	R/W		√			FFH
	FFFFFC00H	外部中断下降沿指定寄存器 0	INTF0		√	√			00H

<R> 注 1. 仅限于 V850ES/SG2:
2. 仅限于仿真器。

(10/11)

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	
FFFFFC06H	外部中断下降沿指定寄存器 3	INTF3	R/W	√	√		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 0H	INTR0		√	√		00H
FFFFFC26H	外部中断上升沿指定寄存器 3H	INTR3		√	√		00H
FFFFFC33H	外部中断上升沿指定寄存器 9H	INTR9H		√	√		00H
FFFFFC60H	端口 0 功能控制寄存器	PF0		√	√		00H
FFFFFC66H	端口 3 功能控制寄存器	PF3				√	0000H
FFFFFC66H	端口 3 功能控制寄存器 L	PF3L		√	√		00H
FFFFFC67H	端口 3 功能控制寄存器 H	PF3H		√	√		00H
FFFFFC68H	端口 4 功能控制寄存器	PF4		√	√		00H
FFFFFC6AH	端口 5 功能控制寄存器	PF5		√	√		00H
FFFFFC72H	端口 9 功能控制寄存器	PF9				√	0000H
FFFFFC72H	端口 9 功能控制寄存器 L	PF9L		√	√		00H
FFFFFC73H	端口 9 功能控制寄存器 H	PF9H		√	√		00H
FFFFFD00H	CSIB0 控制寄存器 0	CB0CTL0	R	√	√		01H
FFFFFD01H	CSIB0 控制寄存器 1	CB0CTL1		√	√		00H
FFFFFD02H	CSIB0 控制寄存器 2	CB0CTL2	R/W		√		00H
FFFFFD03H	CSIB0 状态寄存器	CB0STR		√	√		00H
FFFFFD04H	CSIB0 接收数据寄存器	CB0RX	R			√	0000H
FFFFFD04H	CSIB0 接收数据寄存器 L	CB0RXL			√		00H
FFFFFD06H	CSIB0 发送数据寄存器	CB0TX	R/W			√	0000H
FFFFFD06H	CSIB0 发送数据寄存器 L	CB0TXL			√		00H
FFFFFD10H	CSIB1 控制寄存器 0	CB1CTL0	R	√	√		01H
FFFFFD11H	CSIB1 控制寄存器 1	CB1CTL1		√	√		00H
FFFFFD12H	CSIB1 控制寄存器 2	CB1CTL2	R/W		√		00H
FFFFFD13H	CSIB1 状态寄存器	CB1STR		√	√		00H
FFFFFD14H	CSIB1 接收数据寄存器	CB1RX	R			√	0000H
FFFFFD14H	CSIB1 接收数据寄存器 L	CB1RXL			√		00H
FFFFFD16H	CSIB1 发送数据寄存器	CB1TX	R/W			√	0000H
FFFFFD16H	CSIB1 发送数据寄存器 L	CB1TXL			√		00H
FFFFFD20H	CSIB2 控制寄存器 0	CB2CTL0	R	√	√		01H
FFFFFD21H	CSIB2 控制寄存器 1	CB2CTL1		√	√		00H
FFFFFD22H	CSIB2 控制寄存器 2	CB2CTL2	R/W		√		00H
FFFFFD23H	CSIB2 状态寄存器	CB2STR		√	√		00H
FFFFFD24H	CSIB2 接收数据寄存器	CB2RX	R			√	0000H
FFFFFD24H	CSIB2 接收数据寄存器 L	CB2RXL			√		00H
FFFFFD26H	CSIB2 发送数据寄存器	CB2TX	R/W			√	0000H
FFFFFD26H	CSIB2 发送数据寄存器 L	CB2TXL			√		00H
FFFFFD30H	CSIB3 控制寄存器 0	CB3CTL0	R	√	√		01H
FFFFFD31H	CSIB3 控制寄存器 1	CB3CTL1		√	√		00H
FFFFFD32H	CSIB3 控制寄存器 2	CB3CTL2	R/W		√		00H
FFFFFD33H	CSIB3 状态寄存器	CB3STR		√	√		00H
FFFFFD34H	CSIB3 接收数据寄存器	CB3RX	R			√	0000H
FFFFFD34H	CSIB3 接收数据寄存器 L	CB3RXL			√		00H

(11/11)

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	
FFFFFD36H	CSIB3 发送数据寄存器	CB3TX	R/W			√	0000H
FFFFFD36H	CSIB3 发送数据寄存器 L	CB3TXL			√		00H
FFFFFD40H	CSIB4 控制寄存器 0	CB4CTL0		√	√		01H
FFFFFD41H	CSIB4 控制寄存器 1	CB4CTL1		√	√		00H
FFFFFD42H	CSIB4 控制寄存器 2	CB4CTL2			√		00H
FFFFFD43H	CSIB4 状态寄存器	CB4STR		√	√		00H
FFFFFD44H	CSIB4 接收数据寄存器	CB4RX	R			√	0000H
FFFFFD44H	CSIB4 接收数据寄存器 L	CB4RXL			√		00H
FFFFFD46H	CSIB4 发送数据寄存器	CB4TX	R/W			√	0000H
FFFFFD46H	CSIB4 发送数据寄存器 L	CB4TXL			√		00H
FFFFFD80H	IIC 移位寄存器 0	IIC0 ^注			√		00H
FFFFFD82H	IIC 控制寄存器 0	IICCO ^注		√	√		00H
FFFFFD83H	IIC 从设备地址寄存器 0	SVA0 ^注			√		00H
FFFFFD84H	IIC 时钟选择寄存器 0	IICCL0 ^注		√	√		00H
FFFFFD85H	IIC 功能扩展寄存器 0	IICX0 ^注		√	√		00H
FFFFFD86H	IIC 状态寄存器 0	IICS0 ^注	R	√	√		00H
FFFFFD8AH	IIC 标志寄存器 0	IICF0 ^注	R/W	√	√		00H
FFFFFD90H	IIC 移位寄存器 1	IIC1 ^注			√		00H
FFFFFD92H	IIC 控制寄存器 1	IICC1 ^注		√	√		00H
FFFFFD93H	IIC 从设备地址寄存器 1	SVA1 ^注			√		00H
FFFFFD94H	IIC 时钟选择寄存器 1	IICCL1 ^注		√	√		00H
FFFFFD95H	IIC 功能扩展寄存器 1	IICX1 ^注		√	√		00H
FFFFFD96H	IIC 状态寄存器 1	IICS1 ^注	R	√	√		00H
FFFFFD9AH	IIC 标志寄存器 1	IICF1 ^注	R/W	√	√		00H
FFFFFDA0H	IIC 移位寄存器 2	IIC2 ^注			√		00H
FFFFFDA2H	IIC 控制寄存器 2	IICC2 ^注		√	√		00H
FFFFFDA3H	IIC 从设备地址寄存器 2	SVA2 ^注			√		00H
FFFFFDA4H	IIC 时钟选择寄存器 2	IICCL2 ^注		√	√		00H
FFFFFDA5H	IIC 功能扩展寄存器 2	IICX2 ^注		√	√		00H
FFFFFDA6H	IIC 状态寄存器 2	IICS2 ^注	R	√	√		00H
FFFFDDAAH	IIC 标志寄存器 2	IICF2 ^注	R/W	√	√		00H
FFFFDDBEH	外部总线接口模式控制寄存器	EXIMC		√	√		00H

注 仅限内置 I²C 总线版本（Y 产品）

3.4.7 可编程周边 I/O 寄存器

BPC 寄存器是用来选择可编程周边 I/O 寄存器区域的寄存器。

(1) 周边 I/O 区域选择控制寄存器 (BPC)

BPC 寄存器可用 16 位指令进行读 / 写操作。
复位信号的输入将把该寄存器置为 0000H。

1514131211109876543210地址默认值

BPC

PA150PA13PA12PA11PA10PA09PA08PA07PA06PA05PA04PA03PA02PA01PA00

FFFFF064H0000H

位地址	位名称	功能						
15	PA15	允许 / 禁止使用可编程周边 I/O 区域 <table><tr><td>PA15</td><td>使用可编程周边 I/O 区域</td></tr><tr><td>0</td><td>禁止使用可编程周边 I/O 区域</td></tr><tr><td>1</td><td>允许使用可编程周边 I/O 区域</td></tr></table>	PA15	使用可编程周边 I/O 区域	0	禁止使用可编程周边 I/O 区域	1	允许使用可编程周边 I/O 区域
PA15	使用可编程周边 I/O 区域							
0	禁止使用可编程周边 I/O 区域							
1	允许使用可编程周边 I/O 区域							
13 到 0	PA13 到 PA00	指定可编程周边 I/O 区域中的一个地址 (分别与 A27 到 A14 对应)。						

注意事项

当将 PA15 设置为 1 时, 请务必将 BPC 寄存器设置为 8FFBH。
当将 PA15 设置为 0 时, 请务必将 BPC 寄存器设置为 0000H。

可编程周边 I/O 寄存器区域的详细列表见 表 19-16 寄存器访问类型。

3.4.8 特殊寄存器

特殊寄存器是为了防止由程序挂起引起的数据非法写入而受到保护的寄存器。V850ES/SG2 有 8 个特殊寄存器，
<R> V850ES/SG2-H 有 7 个特殊寄存器。

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

<R> 注 仅限于 V850ES/SG2:

此外，器件还提供了一个命令寄存器（PRCDM）来保护特殊寄存器，避免对其非法写操作，这样应用系统就不会因为程序挂起而意外停止工作了。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。

(1) 向特殊寄存器中写入数据

请按照以下流程向特殊寄存器中写入数据。

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

[例] 写 PSC 寄存器（设置待机模式）

```

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

```

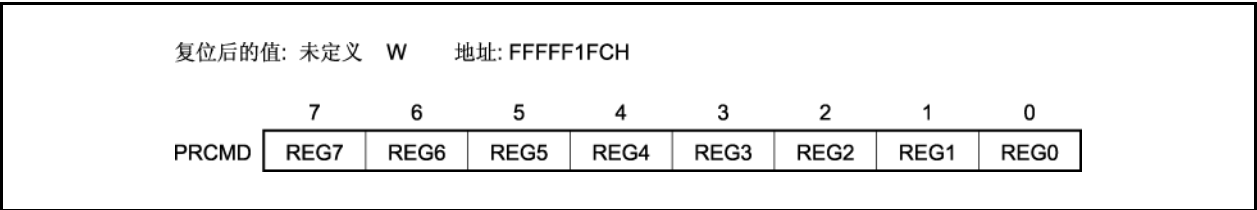
读取特殊寄存器时，无需经过特定流程。

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

- 注意事项**
1. 当执行存储指令向命令寄存器中存储数据的操作时，系统将不响应中断。存储指令应该是按上述的第 <3> 和 <4> 步骤连续执行的。若在步骤 <3> 和 <4> 之间执行了其他指令，而这个指令又进行了中断的响应，则上述操作可能无法实现，导致错误。
 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<4>)的通用寄存器来向 PRCMD 寄存器(例子中的<3>)写入虚数据。这个过程也同样适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程序挂起而停止工作。除非数据先被写入 PRCMD，否则向特殊寄存器中写入的数据都是无效的。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，则读取的数据不确定）。



(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读写。
复位输入将把该寄存器置为 00H。

复位后的值: 00H		R/W	地址: FFFF802H					
SYS	7	6	5	4	3	2	1	<0>
	0	0	0	0	0	0	0	PRERR
PRERR		检测保护错误						
0		保护错误未发生						
1		保护错误发生						

PRERR 标志会在以下情况下改变。

(a) 置位条件（PRERR 标志 = 1）

- (i) 当向特殊寄存器写入数据之前没有对 PRCMD 寄存器进行写入时 (在执行 3.4.8 (1) 向特殊寄存器中写入数据中的步骤 <4> 之前没有执行步骤 <3>)
- (ii) 当对 PRCMD 寄存器进行写入操作后，没有进行对特殊寄存器的写入，而是写入了片上周边 I/O 寄存器时 (如果 3.4.8 (1) 向特殊寄存器中写入数据中的步骤<4>不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，若插入了对片上周边 I/O 寄存器的读取操作(位操作除外)，PRERR 标志位不会被置 1，数据也可以正常写入特殊寄存器。

(b) 清除条件（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，马上向 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。

3.4.9 注意事项

(1) 需要首先设置的寄存器

<R> 在使用 V850ES/SG2 与 V850ES/SG2-H 器件时，请务必首先对以下寄存器进行设置。

- 系统等待寄存器 (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，再设置各端口相关的寄存器，将总线引脚设置为复用功能中的总线控制引脚模式。

(a) 系统等待寄存器 (VSWC)

VSWC 寄存器用于控制总线访问片上周边 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/SG2 与 V850ES/SG2-H 要求的等待周期取决于运行频率。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。然而，如果 f_{CLK} 高于 20MHz，则只能使用 V850ES/SG2-H。
可以对 VSWC 寄存器进行 8 位的读写操作。(地址：FFFFF06EH, 默认值:77H)。

<R>
<R>

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} < 25 MHz	01H	1
25 MHz ≤ f _{CLK} ≤ 32 MHz	11H	2

<R>
<R>

(b) 片上调试模式寄存器 (OCDM)

详见 第 31 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 11 章 看门狗定时器 2 的功能。

(2) 访问特定的片上周边 I/O 寄存器

本产品具有两种内部系统总线。

一种是 CPU 总线，另一种是用于与低速周边硬件进行通信的外围总线。

CPU 总线和外围总线的时钟是异步的。如果发生 CPU 访问和外围硬件访问的冲突，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，对外围硬件进行访问时，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。

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

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。

(1/2)

周边功能	寄存器名称	访问	k
16 位定时器 / 事件计数器 P (TMP) (n = 0 到 5)	TPnCNT	读	1 或 2
	TPnCCR0, TPnCCR1	写	- 第一次访问: 无等待 - 连续写入: 3 或 4
		读	1 或 2
16-位定时器 / 事件计数器 Q (TMQ)	TQ0CNT	读	1 或 2
	TQ0CCR0 到 TQ0CCR3	写	- 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
实时输出功能(RTO)	RTBL0	写 (RTPC0.RTPOE0 位= 0)	1
	RTBH0	写 (RTPC0.RTPOE0 位= 0)	1
A/D 转换器	ADA0M0	读	1 或 2
	ADA0CR0 到 ADA0CR11	读	1 或 2
	ADA0CR0H 到 ADA0CR11H	读	1 或 2
<R> I ² C00 至 I ² C02 ^{※1}	IICS0 到 IICS2 读	读	1

(2/2)

周边功能	寄存器名称	访问	k
<R>	CAN 控制器 ^{注2} (m = 0 到 31, a = 1 到 4)	读 / 写	$(f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MIN.) ^{注3} $(2 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MAX.) ^{注3}
	C0GMABT, C0GMABTD, C0MASKaL, C0MASKaH, C0LEC, C0INFO, C0ERC, C0IE, C0INTS, C0BRP, C0BTR, C0TS		
<R>	C0GMCTRL, C0GMCS, C0CTRL	读 / 写	$(f_{XX}/f_{CAN} + 1)/(2 + j)$ (MIN.) ^{注3} $(2 \times f_{XX}/f_{CAN} + 1)/(2 + j)$ (MAX.) ^{注3}
	C0RGPT, C0TGPT	写	$(f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MIN.) ^{注3} $(2 \times f_{XX}/f_{CANMOD} + 1)/(2 + j)$ (MAX.) ^{注3}
<R>		读	$(3 \times f_{XX}/f_{CANMODE} + 1)/(2 + j)$ (MIN.) ^{注3} $(4 \times f_{XX}/f_{CANMODE} + 1)/(2 + j)$ (MAX.) ^{注3}
	C0LIPT, C0LOPT	读	$(3 \times f_{XX}/f_{CANMODE} + 1)/(2 + j)$ (MIN.) ^{注3} $(4 \times f_{XX}/f_{CANMODE} + 1)/(2 + j)$ (MAX.) ^{注3}
<R>	C0MCTRLm	写	$(4 \times f_{XX}/f_{CAN} + 1)/(2 + j)$ (MIN.) ^{注3} $(5 \times f_{XX}/f_{CAN} + 1)/(2 + j)$ (MAX.) ^{注3}
		读	$(3 \times f_{XX}/f_{CAN} + 1)/(2 + j)$ (MIN.) ^{注3} $(4 \times f_{XX}/f_{CAN} + 1)/(2 + j)$ (MAX.) ^{注3}
<R>	C0MDATA01m, C0MDATA0m, C0MDATA1m, C0MDATA23m, C0MDATA2m, C0MDATA3m, C0MDATA45m, C0MDATA4m, C0MDATA5m, C0MDATA67m, C0MDATA6m, C0MDATA7m, C0MDLCm, C0MCONFm, C0MIDLm, C0MIDHm	写(8 bits)	$(4 \times f_{XX}/f_{CANMODE} + 1)/(2 + j)$ (MIN.) ^{注3} $(5 \times f_{XX}/f_{CANMODE} + 1)/(2 + j)$ (MAX.) ^{注3}
		写(16 bits)	$(2 \times f_{XX}/f_{CANMODE} + 1)/(2 + j)$ (MIN.) ^{注3} $(3 \times f_{XX}/f_{CANMODE} + 1)/(2 + j)$ (MAX.) ^{注3}
<R>		读(8/16 bits)	$(3 \times f_{XX}/f_{CANMODE} + 1)/(2 + j)$ (MIN.) ^{注3} $(4 \times f_{XX}/f_{CANMODE} + 1)/(2 + j)$ (MAX.) ^{注3}
CRC	CRCD	写	1

访问需要的时钟数 $= 3 + i + j + (2 + j) \times k$

- 注
1. 仅限内置 I²C 总线版本 (Y 产品)
 2. 仅限内置 CAN 控制器的版本
 3. 小数点后的位需进行舍入处理。

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，系统工作于副时钟时
- ~
- 当 CPU 工作于内部振荡器时钟时

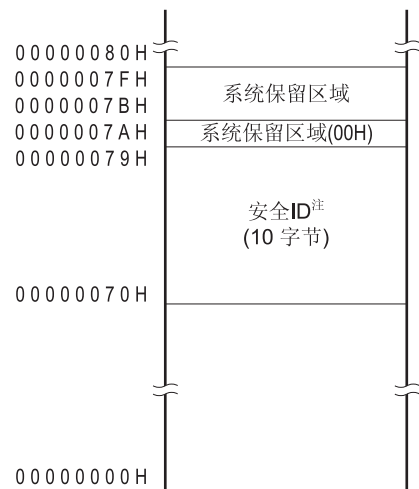
- 备注 f_{xx}: 主时钟频率= f_{xx}
 f_{CANMOD}: CAN 模块系统时钟
 f_{CAN}: 给 CAN 提供的时钟
 i: VSWC 寄存器高 4 位的值 (0 或 1)
 j: VSWC 寄存器低 4 位的值 (0 或 1) r

(3) 系统保留区域

<R> 在 V850ES/SG2 与 V850ES/SG2-H 的 flash 存储器版本中，0000007AH 到 0000007FH 是扩展功能保留区域，因此不推荐使用这块区域。
然而，在下列产品的特殊版本中，保证要设置 0000007AH 为 00H。

- μPD70F3261, 70F3261Y, 70F3271, 70F3271Y, 70F3281, 70F3281Y: 版本. 1.0
- <R> • μPD70F3263, 70F3263Y, 70F3263HY, 70F3273, 70F3273Y, 70F3273HY, 70F3283, 70F3283Y, 70F3283HY:无保留版本

- 备注 1. 出去上述的产品（包括所有手册中列出的产品），即使将 0000007AH 设为 00H 也不会影响操作。
2. 通过封装标记第三行 DS, ES,或 CS 之后号码来检查产品版本。如果印的名称为 V850ES/SG2, V850ES/SG2-H，上述约束不起作用。
- <R> 更多信息，请联系日电电子的销售代表。



- 注 对于安全 ID，参见 31.6.1 安全 ID。
- 注意事项 当 flash 存储器中数据被删除时，所有位置为 1。

(4) 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，在指令<1>之后 sld 指令<iii>之前的指令<2>的解码过程与中断请求发生冲突，那么种类 <1> 的指令的执行结果将有可能不会保存在寄存器中。

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
- 乘指令: mul, mulh, mulhi, mulu

指令<2>

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

<例>

<i> ld.w [r11], r10	如果 ld 指令 <i> 的执行完成之前，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。
•	
•	
<ii> mov r10, r28	
<iii> sld.w 0x28, r10	

(b) 解决方法

- <1> 当使用编译器（CA850）时
使用 CA850 版本。请使用 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> 对于汇编器
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： 84
 - 5 V 容限/ N 沟道开漏输出可选： 40（端口 0， 3 至 5， 9）
- 输入或者输出可以逐位指定

4.2 端口的基本配置

<R> V850ES/SG2 与 V850ES/SG2-H 由端口 0， 1， 3 至 5， 7， 9， CM， CT， DH 以及 DL 共 84 个 I/O 端口组成。端口配置情况如下图所示。

图 4-1. 端口配置图

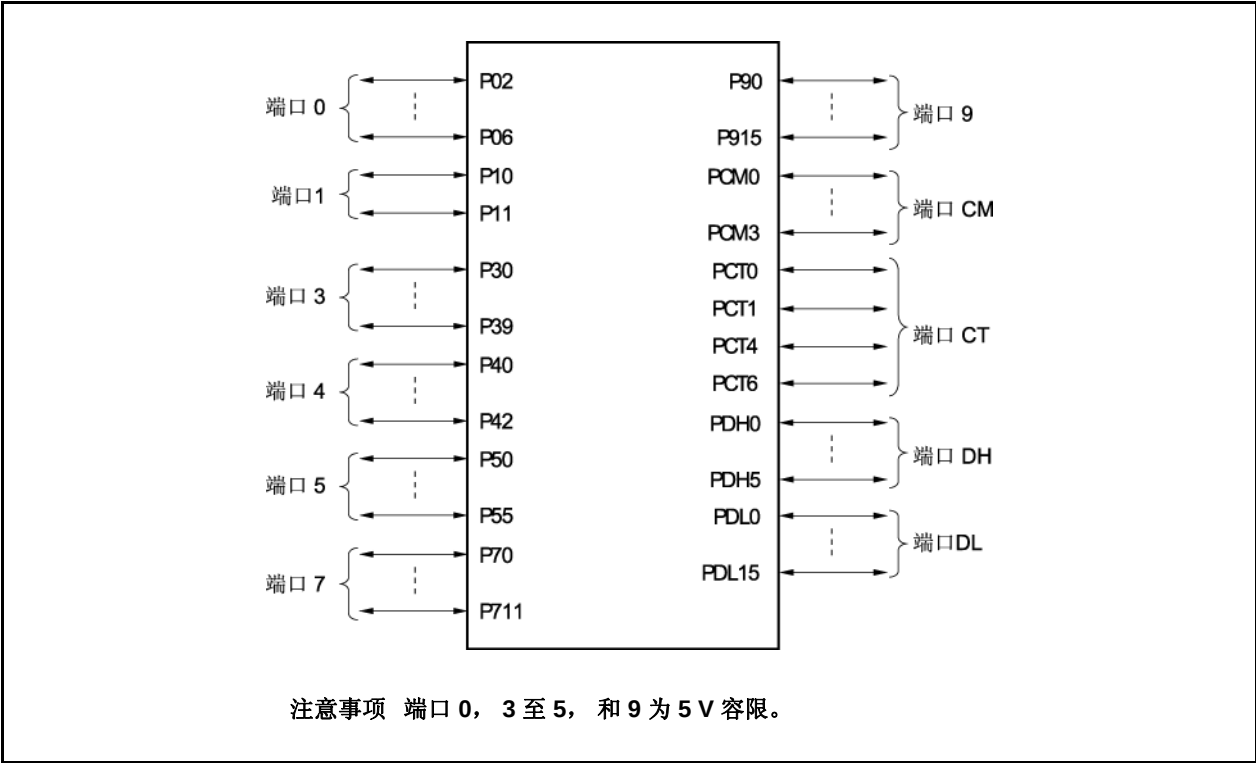


表 4-1. 引脚的 I/O 缓冲器电源

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

4.3 端口配置

表 4-2. 端口配置

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

(1) 端口 n 寄存器（Pn）

通过读 / 写 Pn 寄存器，将数据输出到外部设备或从外部设备输入。
Pn 寄存器包括保持输出数据的端口锁存器，以及读取引脚状态的电路。
Pn 寄存器的每一位都与端口 n 的一个引脚对应。该寄存器可以以 1 位为单位读取或写入。

复位之后: 00H (输出锁存) R/W								
Pn	7	6	5	4	3	2	1	0
	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0

Pnm	输出数据控制（输出模式）
0	输出0.
1	输出1.

无论 PMCn 寄存器的设置如何，从 Pn 寄存器读写数据如下。

表 4-3. 读取 / 写入 Pn 寄存器

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

注 除非有新的值写入，否则写入到输出锁存的值会继续保持。

(2) 端口 n 模式寄存器 (PMn)

PMn 寄存器指定对应端口引脚作为输入模式或输出模式。
此寄存器的每一位都与端口 n 的一个引脚对应，可以以 1 位为单位指定输入输出模式。

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

(3) 端口 n 模式控制寄存器 (PMCn)

PMCn 寄存器指定端口模式或复用功能。
此寄存器的每一位都与端口 n 的一个引脚对应，可以以 1 位为单位指定端口模式。

复位后: 00H R/W

	7	6	5	4	3	2	1	0
PMCn	PMCn7	PMCn6	PMCn5	PMCn4	PMCn3	PMCn2	PMCn1	PMCn0

PMCnm	指定操作模式
0	端口模式
1	复用功能模式

(4) 端口 n 功能控制寄存器 (PFCn)

如果引脚有两个复用功能，PFCn 寄存器指定引脚要使用的复用功能。
此寄存器的每一位都与端口 n 的一个引脚对应，可以以 1 位为单位指定端口引脚的复用功能。

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

(5) 端口 n 功能控制扩展寄存器 (PFCEn)

如果引脚有三个或更多的复用功能，PFCEn 寄存器指定引脚要使用的复用功能。
此寄存器的每一位都与端口 n 的一个引脚对应，可以以 1 位为单位指定端口引脚的复用功能。

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

(6) 端口 n 功能寄存器 (PFn)

PFn 寄存器指定正常输出或 N 沟道开漏输出。
此寄存器的每一位都与端口 n 的一个引脚对应，可以以 1 位为单位指定端口引脚的输出模式。

复位后: 00H R/W								
	7	6	5	4	3	2	1	0
PFn	PFn7	PFn6	PFn5	PFn4	PFn3	PFn2	PFn1	PFn0

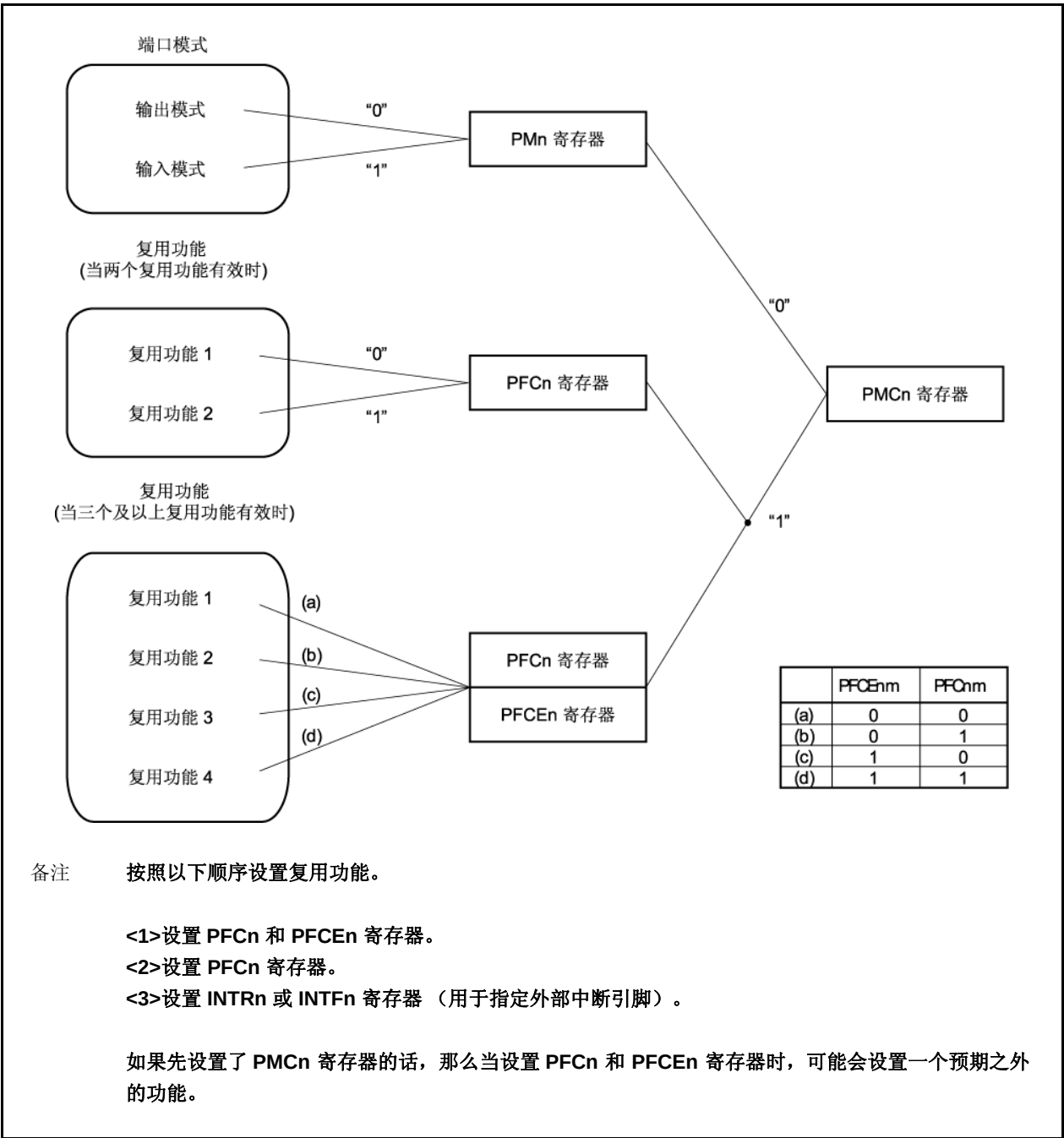
PFnm ^注	正常输出/开漏输出控制
0	正常输出 (CMOS 输出)
1	N沟开漏输出

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

(7) 端口设置

按照下图设置端口。

图 4-2. 每个寄存器和引脚功能的设置



4.3.1 端口 0

端口 0 是可以以 1 位为单位控制 I/O 设置的 5 位端口。
端口 0 包含以下复用功能引脚。

表 4-4. 端口 0 复用功能引脚

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

注 DRST 引脚用作片上调试（仅限于 flash 存储器版本）。

如果不使用片上调试功能，从 RESET 引脚复位释放到 OCDM.OCDM0 位清除（0）期间，需要将 P05/INTP2/DRST 引脚固定为低电平。虽然掩模 ROM 版本不支持片上调试模式，但还是内置有一个片上下拉电阻。对 P05/INTP2 的控制跟 flash 存储器版本的情况相同。

详细信息请参考 4.6.3 片上调试引脚中的注意事项。

注意事项 P02 至 P06 引脚在复用功能的输入模式中具有磁滞特性，但在端口模式下不具有磁滞特性。

备注 GF100-pin 塑封 QFP（14 × 20）
GC100-pin 塑封 LQFP（细间距）（14 × 14）

(1) 端口 0 寄存器（P0）

复位后: 00H (输出锁存) R/W 地址: FFFFF400H

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

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

(2) 端口 0 模式寄存器 (PM0)

复位后: FFH R/W 地址: FFFFF420H

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

PM0n	I/O 模式控制 (n = 2 到 6)
0	输出模式
1	输入模式

(3) 端口 0 模式控制寄存器 (PMC0)

复位后: 00H R/W 地址: FFFFF440H

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

PMC06	指定 P06 引脚的操作模式
0	I/O 端口
1	INTP3 输入

PMC05	指定 P05 引脚的操作模式
0	I/O 端口
1	INTP2 输入

PMC04	指定 P04 引脚的操作模式
0	I/O 端口
1	INTP1 输入

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

PMC02	指定 P02 引脚的操作模式
0	I/O 端口
1	NMI 输入

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

(4) 端口 0 功能控制寄存器 (PFC0)

复位后: 00H R/W 地址: FFFFF460H

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

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

(5) 端口 0 功能寄存器 (PF0)

复位后: 00H R/W 地址: FFFFFC60H								
	7	6	5	4	3	2	1	0
PF0	0	PF06	PF05	PF04	PF03	PF02	0	0

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

注意事项 为了将输出引脚上拉到 EV_{DD} 或者更高，保证对应的 PF0n 位设置为 1。

4.3.2 端口 1

端口 1 是可以以 1 位为单位控制 I/O 设置的 2 位端口。
端口 1 包含以下复用功能引脚。

表 4-5. 端口 1 复用功能引脚

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

<R> 注意事项 当上电时，P10 与 P11 引脚瞬时输出未知。

备注 GF100-pin 塑封 QFP（14 × 20）
GC100-pin 塑封 LQFP（细间距）（14 × 14）

(1) 端口 1 寄存器（P1）

复位后: 00H (输出锁存) R/W 地址: FFFFF402H

P1

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

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

<R> 注意事项 请不要在 D/A 转换过程中对 P1 寄存器进行读 / 写（见 14.4.3 注意事项）。

(2) 端口 1 模式寄存器（PM1）

复位后: FFH R/W 地址: FFFFF422H

PM1

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

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

注意事项 1. 当使用 P1n 作为复用功能（ANOn 引脚输出）时，需要将 PM1n 设置成 1。
2. 当使用 P10 和 P11 其中一个引脚作为 I/O 端口，另一个作为 D/A 输出引脚时，也需要在应用程序中将 PM1n 设置成 1，在 D/A 输出过程中，应用程序保持端口 I/O 电平不改变。

4.3.3 端口 3

端口 3 是可以以 1 位为单位控制 I/O 设置的 10 位端口。

端口 3 包含以下复用功能引脚。

表 4-6. 端口 3 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
P30	27	25	TXDA0/SOB4	输出	可选作 N 沟道开漏输出	G-3
P31	28	26	RXDA0/INTP7/SIB4	输入		N-3
P32	29	27	ASCKA0/SCKB4/TIP00/TOP00	I/O		U-1
P33	30	28	TIP01/TOP01	I/O		G-1
P34	31	29	TIP10/TOP10	I/O		G-1
P35	32	30	TIP11/TOP11	I/O		G-1
P36	33	31	CTXD0 ^{注1} /IETX0 ^{注2}	输出		G-3
P37	34	32	CRXD0 ^{注1} /IERX0 ^{注2}	输入		G-4
P38	37	35	TXDA2/SDA00 ^{注3}	I/O		G-12
P39	38	36	RXDA2/SCL00 ^{注3}	I/O		G-6

- 注
1. 仅限内置 CAN 控制器的版本
 2. 仅限内置 IEBus 控制器的版本
 3. 仅限内置 I²C 总线的版本（Y 产品）

<R> 注意事项 P31 至 P35 和 P37 至 P39 引脚在复用功能引脚的输入模式中具有磁滞特性，但在端口模式中不具有磁滞特性。

备注 GF100-pin 塑封 QFP（14 × 20）
GC100-pin 塑封 LQFP（细间距）（14 × 14）

(1) 端口 3 寄存器 (P3)

复位后: 0000H (输出锁存)		R/W	地址:		P3 FFFFF406H, P3L FFFFF406H, P3H FFFFF407H			
	15	14	13	12	11	10	9	8
P3 (P3H)	0	0	0	0	0	0	P39	P38
	7	6	5	4	3	2	1	0
(P3L)	P37	P36	P35	P34	P33	P32	P31	P30

P3n	输出数据控制 (输出模式下) (n = 0 到 9)
0	输出 0
1	输出 1

备注

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

(2) 端口 3 模式寄存器 (PM3)

复位后: FFFFH		R/W	地址:		PM3 FFFFF426H, PM3L FFFFF426H, PM3H FFFFF427H			
	15	14	13	12	11	10	9	8
PM3 (PM3H)	1	1	1	1	1	1	PM39	PM38
	7	6	5	4	3	2	1	0
(PM3L)	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	I/O 模式控制 (n = 0 到 9)
0	输出模式
1	输入模式

备注

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

复位后：0000H R/W 地址：
PMC3 FFFFF446H,
PMC3L FFFFF446H, PMC3H FFFFF447H

	7	6	5	4	3	2	1	0
(PMC3L)	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

PMC39	指定 P39引脚的操作模式
0	I/O 端口
1	RXDA2 输入/SCL00 I/O

PMC38	指定 P38引脚的操作模式
0	I/O 端口
1	TXDA2 输出/SDA00 I/O

PMC37	指定 P37 引脚的操作模式
0	I/O 端口
1	CRXD0 输入/IERX0 输入

PMC36	指定 P36 引脚的操作模式
0	I/O 端口
1	CTXD0 输出/IETX0 输出

PMC35	指定 P35 引脚的操作模式
0	I/O 端口
1	TIP11 输入/TOP11 输出

PMC34	指定 P34 引脚的操作模式
0	I/O 端口
1	TIP10 输入/TOP10 输出

PMC33	指定 P33 引脚的操作模式
0	I/O 端口
1	TIP01 输入/TOP01 输出

PMC32	指定 P32 引脚的操作模式
0	I/O 端口
1	ASCKA0 输入/SCKB4 I/O/TIP00 输入/TOP00 输出

PMC31	指定 P31 引脚的操作模式
0	I/O 端口
1	RXDA0 输入/SIB4 输入/INTP7 输入

PMC30	指定 P30 引脚的操作模式
0	I/O 端口
1	TXDA0 输出/SOB4 输出

备注

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

(4) 端口 3 功能控制寄存器 (PFC3)

复位后: 0000H		R/W	地址:		PFC3 FFFFF466H, PFC3L FFFFF466H, PFC3L FFFFF467H					
			15	14	13	12	11	10	9	8
PFC3 (PFC3H)			0	0	0	0	0	0	PFC39	PFC38
			7	6	5	4	3	2	1	0
(PFC3L)			PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

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

(5) 端口 3 功能控制扩展寄存器 L (PFCE3L)

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

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

(6) 端口 3 复用功能指定

PFC39	P39 引脚复用功能的指定
0	RXDA2 输入
1	SCL00 输入

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

PFC37	P37 引脚复用功能的指定
0	CRXD0 输入
1	IERX0 输入

PFC36	P36 引脚复用功能的指定
0	CTXD0 输出
1	IETX0 输出

PFC35	P35 引脚复用功能的指定
0	TIP11 输入
1	TOP11 输出

PFC34	P34 引脚复用功能的指定
0	TIP10 输入
1	TOP10 输出

PFC33	P33 引脚复用功能的指定
0	TIP01 输入
1	TOP01 输出

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

PFC31	P31 引脚复用功能的指定
0	RXDA0 input/INTP7 [※] 输入
1	SIB4 输入

PFC30	P30 引脚复用功能的指定
0	TXDA0 输出
1	SOB4 输出

注 INTP7 引脚和 RXDA0 引脚为一组复用功能引脚。当该引脚用作 RXDA0 引脚时，应关闭 INTP7 复用功能的边沿检测。（清除 INTF3.INTF31 位和 INTR3.INTR31 位为 0。）当该引脚用作 INTP7 时，停止 UARTA0 接收。（清除 UA0CTL0.UA0RXE 位为 0。）

(7) 端口 3 功能寄存器 (PF3)

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

<R> 注意事项 为了将输出引脚上拉到 EV_{DD} 或者更高，保证对应的 PF3n 位设置为 1。

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

4.3.4 端口 4

端口 4 是可以 1 位为单位控制 I/O 的 3 位端口。
端口 4 包含以下复用功能引脚。

表 4-7. 端口 4 复用功能引脚

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

注 仅限内置 I²C 总线的版本（Y 产品）

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，但在端口模式中不具有磁滞特性。

备注 GF100-pin 塑封 QFP（14 × 20）
GC100-pin 塑封 LQFP（细间距）（14 × 14）

（1）端口 4 寄存器（P4）

复位后: 00H (输出锁存) R/W 地址: FFFFF408H								
P4	7	6	5	4	3	2	1	0
	0	0	0	0	0	P42	P41	P40
P4n		输出数据控制 (输出模式下) (n = 0 到 2)						
0		输出 0						
1		输出 1						

（2）端口 4 模式寄存器（PM4）

复位后: FFH R/W 地址: FFFFF428H								
PM4	7	6	5	4	3	2	1	0
	1	1	1	1	1	PM42	PM41	PM40
PM4n		I/O 模式控制 (n = 0 到 2)						
0		输出模式						
1		输入模式						

(3) 端口 4 模式控制寄存器 (PMC4)

复位后: 00H R/W 地址: FFFFF448H

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

PMC42	指定 P42 引脚的操作模式	
0	I/O 端口	
1	SCKB0 I/O	
PMC41	指定 P41 引脚的操作模式	
0	I/O 端口	
1	SOB0 输出/SCL01 I/O	
PMC40	指定 P40 引脚的操作模式	
0	I/O 端口	
1	SIB0 输入/SDA01 I/O	

(4) 端口 4 功能控制寄存器 (PFC4)

复位后: 00H R/W 地址: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	0	0	0	0	0	0	PFC41	PFC40

PFC41	指定 P41 引脚的操作模式	
0	SOB0 输出	
1	SCL01 I/O	
PFC40	指定 P40 引脚的操作模式	
0	SIB0 输入	
1	SDA01 I/O	

(5) 端口 4 功能寄存器 (PF4)

复位后: 00H R/W 地址: FFFFC68H

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

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

<R>

注意事项 为了将输出引脚上拉到 EV_{DD} 或者更高，保证对应的 PF4n 位设置为 1。

4.3.5 端口 5

端口 5 是可以 1 位为单位控制 I/O 的 6 位 端口。

端口 5 包含以下复用功能引脚。

表 4-8. 端口 5 复用功能引脚

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

注 DDI, DDO, DCK, 和 DMS 引脚用作片上调试（针对 flash 存储器版本）。

如果不使用片上调试功能，从 $\overline{\text{RESET}}$ 引脚复位释放到 OCDM.OCDM0 位清除（0）期间，需要将 P05/INTP2/DRST 引脚固定为低电平。虽然掩模 ROM 版本不支持片上调试模式，但还是内置有一个片上下拉电阻。对 P05/INTP2 的控制跟 flash 存储器版本相同。

详细信息请参考 4.6.3 片上调试引脚中的注意事项。

- <R> 注意事项 1. 当上电时，P53 引脚可能瞬时输出不定电平。
2. P50 至 P55 引脚在复用功能的输入模式中具有磁滞特性，但在端口模式中不具有磁滞特性。

备注 GF100-pin 塑封 QFP（14 × 20）
GC100-pin 塑封 LQFP（细间距）（14 × 14）

（1）端口 5 寄存器（P5）

复位后: 00H (输出锁存) R/W 地址: FFFFF40AH

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

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

(2) 端口 5 模式寄存器 (PM5)

复位后: FFH R/W 地址: FFFFF42AH

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

PM5n	I/O模式控制 (n = 0 到 5)
0	输出模式
1	输入模式

(3) 端口 5 模式控制寄存器 (PMC5)

复位后: 00H R/W 地址: FFFFF44AH

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

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

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

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

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

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

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

(4) 端口 5 功能控制寄存器 (PFC5)

复位后: 00H R/W 地址: FFFFF46AH

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

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

(5) 端口 5 功能控制扩展寄存器 (PFCE5)

复位后: 00H R/W 地址: FFFFF70AH

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

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

(6) 端口 5 复用功能指定

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

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

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

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

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

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

注 KRn 引脚和 TIQ0m 引脚为一组复用功能引脚。当该引脚用作 TIQ0m 引脚时，关闭 KRn 引脚按键返回检测无效，这是一个复用功能。（清除 KRM.KRMn 位为 0。）同样，当该引脚用作 KRn 引脚时，使 TIQ0m 引脚沿检测功能无效，这是个复用功能（n=0 到 3，m=0 到 3）。

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

(7) 端口 5 功能寄存器（PF5）

复位后: 00H R/W 地址: FFFFC6AH

PF5

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

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

<R> 注意事项 为了将输出引脚上拉到 EVDD 或者更高，保证对应的 PF5n 位设置为 1。

4.3.6 端口 7

端口 7 是可以以 1 位为单位控制 I/O 设置的 12 位端口。
端口 7 包含以下复用功能引脚。

表 4-9. 端口 7 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
P70	2	100	ANI0	输入	-	A-1
P71	1	99	ANI1	输入		A-1
P72	100	98	ANI2	输入		A-1
P73	99	97	ANI3	输入		A-1
P74	98	96	ANI4	输入		A-1
P77	97	95	ANI5	输入		A-1
P76	96	94	ANI6	输入		A-1
P77	95	93	ANI7	输入		A-1
P78	94	92	ANI8	输入		A-1
P79	93	91	ANI9	输入		A-1
P710	92	90	ANI10	输入		A-1
P711	91	89	ANI11	输入		A-1

备注 GF100-pin 塑封 QFP（14 × 20）
GC100-pin 塑封 LQFP（细间距）（14 × 14）

（1）端口 7 寄存器 H，端口 7 寄存器 L（P7H，P7L）

<R>

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

P7H	7	6	5	4	3	2	1	0
	0	0	0	0	P711	P710	P79	P78
P7L	7	6	5	4	3	2	1	0
	P77	P76	P75	P74	P73	P72	P71	P70
P7n	输出数据控制 (输出模式下) (n = 0 to 11)							
0	输出 0							
1	输出 1							

<R>

注意事项 在 A/D 转换期间，不要读 / 写 P7H 和 P7L 寄存器（见 13.6 （4）复用 I/O）。

备注 不能以 16 位为单位访问 P7 寄存器。只能以 8 位或 1 位为单位读 / 写 P7H 和 P7L 寄存器。

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

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

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

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

注意事项 当使用 P7n 引脚的复用功能（ANIn 引脚）时，需要将 PM7n 位置 1。

备注 不能以 16 位为单位访问 PM7 寄存器。只能以 8 位或 1 位为单位读 / 写 P7H 和 P7L 寄存器。

4.3.7 端口 9

端口 9 是可以以 1 位为单位控制 I/O 设置的 16 位端口。

端口 9 包含以下复用功能引脚。

表 4-10. 端口 9 复用功能引脚

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

注 仅限内置 I²C 总线的版本（Y 产品）

<R> 注意事项 P90 至 P97，P99，P910，和 P912 至 P915 引脚在复用功能引脚中的输入模式具有磁滞特性，但在端口模式中不具有磁滞特性。

备注 GF100-pin 塑封 QFP（14 × 20）
GC100-pin 塑封 LQFP（细间距）（14 × 14）

(1) 端口 9 寄存器 (P9)

复位后: 0000H (输出锁存)		R/W	地址:		P9 FFFFF412H, P9L FFFFF412H, P9H FFFFF413H			
P9 (P9H)	15	14	13	12	11	10	9	8
	P915	P914	P913	P912	P911	P910	P99	P98
(P9L)	7	6	5	4	3	2	1	0
	P97	P96	P95	P94	P93	P92	P91	P90
P9n	输出数据控制 (输出模式下) (n = 0 到 15)							
0	输出 0							
1	输出 1							

备注

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

(2) 端口 9 模式寄存器 (PM9)

复位后: FFFFH		R/W	地址:		PM9 FFFFF432H, PM9L FFFFF432H, PM9H FFFFF433H			
PM9 (PM9H)	15	14	13	12	11	10	9	8
	PM915	PM914	PM913	PM912	PM911	PM910	PM99	PM98
(PM9L)	7	6	5	4	3	2	1	0
	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90
PM9n	I/O 模式控制 (n = 0 到 15)							
0	输出模式							
1	输入模式							

备注

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

(3) 端口 9 模式控制寄存器 (PMC9)

(1/2)

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

PMC9 (PMC9H)	15	14	13	12	11	10	9	8
	PMC915	PMC914	PMC913	PMC912	PMC911	PMC910	PMC99	PMC98
(PMC9L)	7	6	5	4	3	2	1	0
	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90
PMC915	指定 P915 引脚的操作模式							
0	I/O 端口							
1	A15 输出/INTP6 输入/TIP50 输入/TOP50 输出							
PMC914	指定 P914 引脚的操作模式							
0	I/O 端口							
1	A14 输出/INTP5 输入/TIP51 输入/TOP51 输出							
PMC913	指定 P913 引脚的操作模式							
0	I/O 端口							
1	A13 输出/INTP4 输入							
PMC912	指定 P912 引脚的操作模式							
0	I/O 端口							
1	A12 输出/SCKB3 I/O							
PMC911	指定 P911 引脚的操作模式							
0	I/O 端口							
1	A11 输出/SOB3 输出							
PMC910	指定 P910 引脚的操作模式							
0	I/O 端口							
1	A10 输出/SIB3 输入							
PMC99	指定 P99 引脚的操作模式							
0	I/O 端口							
1	A9 输出/SCKB1 I/O							

备注

1. 可以以 16 位为单位读 / 写 PMC9 寄存器。
此外, 当使用 PMC9 寄存器的高 8 位作为 PMC9H 寄存器, 低 8 位作为 PMC9L 寄存器时, PMC9 可以以 8 位或 1 位为单位读 / 写。
2. 要以 8 位 或 1 位为单位读 / 写 PMC9 寄存器的第 8 位至第 15 位, 需要将他们指定为 PMC9H 寄存器的第 0 位至第 7 位。

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

<R>

注意事项 即使 A0 到 A15 只有一个用于分离总线模式，端口 9 也不能作为端口引脚或者是功能复用引脚。将寄存器 PFC9 和 PFCE9 设置为 0000H 之后，寄存器 PMC9 的 16 位全都一次设置为 FFFFH。即使 A0 到 A15 其中一个没有用于分离总线模式，端口 9 也可以作为端口引脚或者是功能复用引脚。

(4) 端口 9 功能控制寄存器 (PFC9)

<R> 注意事项 即使 A0 到 A15 只有一个用于分离总线模式，端口 9 也不能作为端口引脚或者是功能复用引脚。将寄存器 PFC9 与 PFCE9 设为 0000H 之后，寄存器 PMC9 的所有 16 位全都一次设置为 FFFFH。即使 A0 到 A15 其中一个没有用于分离总线模式，端口 9 也可以作为端口引脚或者是功能复用引脚。

复位后: 0000H		R/W	地址:		PFC9 FFFFF472H, PFC9L FFFFF472H, PFC9H FFFFF473H			
PFC9 (PFC9H)	15	14	13	12	11	10	9	8
	PFC915	PFC914	PFC913	PFC912	PFC911	PFC910	PFC99	PFC98
(PFC9L)	7	6	5	4	3	2	1	0
	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

- 备注**
1. 有关指定复用功能的详细内容，请参考 4.3.9 (6) 端口 9 复用功能指定。
 2. 可以以 16 位为单位读 / 写 PFC9 寄存器。
此外，当使用 PFC9 寄存器的高 8 位作为 PFC9H 寄存器，低 8 位作为 PFC9L 寄存器时，PFC9 可以以 8 位或 1 位为单位读 / 写。
 3. 要以 8 位或 1 位为单位读 / 写 PFC9 寄存器的第 8 位至第 15 位，需要将他们指定为 PFC9H 寄存器的第 0 位至第 7 位。

(5) 端口 9 功能控制扩展寄存器 (PFCE9)

复位后: 0000H		R/W	地址:		PFCE9 FFFFF712H, PFCE9L FFFFF712H, PFCE9H FFFFF713H			
PFCE9 (PFCE9H)	15	14	13	12	11	10	9	8
	PFCE915	PFCE914	0	0	0	0	0	0
(PFCE9L)	7	6	5	4	3	2	1	0
	PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

- 备注**
1. 有关指定复用功能的详细内容，请参考 4.3.9 (6) 端口 9 复用功能指定。
 2. 可以以 16 位为单位读 / 写 PFCE9 寄存器。
此外，当使用 PFCE9 寄存器的高 8 位作为 PFCE9H 寄存器，低 8 位作为 PFCE9L 寄存器时，PFCE9 可以以 8 位或 1 位为单位读 / 写。
 3. 要以 8 位或 1 位为单位读 / 写 PFCE9 寄存器的第 8 位至第 15 位，需要将他们指定为 PFCE9H 寄存器的第 0 位至第 7 位。

(6) 端口 9 复用功能指定

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

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

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

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

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

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

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

PFC98	P98 引脚复用功能的指定
0	A8 输出
1	SOB1 输出

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

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

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

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

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

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

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

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

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

(7) 端口 9 功能寄存器 (PF9)

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

<R>

注意事项 为了将输出引脚上拉到 EV_{DD} 或者更高，保证对应的 PF9n 位设置为 1。

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

4.3.8 端口 CM

端口 CM 为可以以位为单位控制 I/O 设置的 4 位端口。

端口 CM 包含以下复用功能引脚。

表 4-11. 端口 CM 复用功能引脚

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

备注 GF100-pin 塑封 QFP (14 × 20)
GC100-pin 塑封 LQFP (细间距) (14 × 14)

(1) 端口 CM 寄存器 (PCM)

复位后: 00H (输出锁存) R/W 地址: FFFF00CH

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

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

(2) 端口 CM 模式寄存器 (PMCM)

复位后: FFH R/W 地址: FFFF02CH

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

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

(3) 端口 CM 模式控制寄存器 (PMCCM)

复位后: 00H R/W 地址: FFFFF04CH

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

PMCCM3	指定 PCM3 引脚的操作模式
0	I/O 端口
1	$\overline{\text{HLDRQ}}$ 输入

PMCCM2	指定PCM2 引脚的操作模式
0	I/O 端口
1	$\overline{\text{HLDAK}}$ 输出

PMCCM1	指定 PCM1 引脚的操作模式
0	I/O 端口
1	CLKOUT 输出

PMCCM0	指定 PCM0 引脚的操作模式
0	I/O 端口
1	$\overline{\text{WAIT}}$ 输入

4.3.9 端口 CT

端口 CT 是可以以 1 位来控制 I/O 设置的 4 位端口。

端口 CT 包含以下复用功能引脚。

表 4-12. 端口 CT 复用功能引脚

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

备注 GF100-pin 塑封 QFP (14 × 20)
GC100-pin 塑封 LQFP (细间距) (14 × 14)

(1) 端口 CT 寄存器 (PCT)

复位后: 00H (输出锁存) R/W 地址: FFFFF00AH

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

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

(2) 端口 CT 模式寄存器 (PMCT)

复位后: FFH R/W 地址: FFFF02AH

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

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

(3) 端口 CT 模式控制寄存器 (PMCCT)

复位后: 00H R/W 地址: FFFF04AH

	7	6	5	4	3	2	1	0
PMOCT	0	PMOCT6	0	PMOCT4	0	0	PMOCT1	PMOCT0

PMOCT6	指定 PCT6 引脚的操作模式
0	I/O 端口
1	ASTB 输出

PMOCT4	指定 PCT4 引脚的操作模式
0	I/O 端口
1	$\overline{\text{RD}}$ 输出

PMOCT1	指定 PCT1 引脚的操作模式
0	I/O 端口
1	$\overline{\text{WR1}}$ 输出

PMOCT0	指定 PCT0 引脚的操作模式
0	I/O 端口
1	$\overline{\text{WR0}}$ 输出

4.3.10 端口DH

端口 CM 为可以以位为单位控制 I/O 设置的 6 位端口。

端口 DH 包含以下复用功能引脚。

表 4-13. 端口 DH 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
PDH0	89	87	A16	输出	—	D-2
PDH1	90	88	A17	输出		D-2
PDH2	61	59	A18	输出		D-2
PDH3	62	60	A19	输出		D-2
PDH4	8	6	A20	输出		D-2
PDH5	9	7	A21	输出		D-2

备注 GF100-pin 塑封 QFP (14 × 20)
GC100-pin 塑封 LQFP (细间距) (14 × 14)

(1) 端口 DH 寄存器 (PDH)

复位后: 00H (输出锁存) R/W 地址: FFFFF006H

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

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

(2) 端口 DH 模式寄存器 (PMDH)

复位后: FFH R/W 地址: FFFFF026H

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

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

(3) 端口 DH 模式控制寄存器 (PMCDH)

复位后: 00H R/W 地址: FFFFF046H								
PMCDH	7	6	5	4	3	2	1	0
	0	0	PMCDH5	PMCDH4	PMCDH3	PMCDH2	PMCDH1	PMCDH0
PMCDHn		指定 PDHn 引脚的操作模式 (n = 0到 5)						
0		I/O 端口						
1		Am 输出 (地址总线输出) (m = 16 到 21)						

4.3.11 端口DL

端口 DL 为可以以位为单位控制 I/O 设置的 16 位端口。

端口 DL 包含以下复用功能引脚。

表 4-14. 端口 DL 复用功能引脚

引脚名称	引脚编号		复用功能引脚名称	I/O	备注	模块类型
	GF	GC				
PDL0	73	71	AD0	I/O	—	D-3
PDL1	74	72	AD1	I/O		D-3
PDL2	75	73	AD2	I/O		D-3
PDL3	76	74	AD3	I/O		D-3
PDL4	77	75	AD4	I/O		D-3
PDL5	78	76	AD5/FLMD1 [※]	I/O		D-3
PDL6	79	77	AD6	I/O		D-3
PDL7	80	78	AD7	I/O		D-3
PDL8	81	79	AD8	I/O		D-3
PDL9	82	80	AD9	I/O		D-3
PDL10	83	81	AD10	I/O		D-3
PDL11	84	82	AD11	I/O		D-3
PDL12	85	83	AD12	I/O		D-3
PDL13	86	84	AD13	I/O		D-3
PDL14	87	85	AD14	I/O		D-3
PDL15	88	86	AD15	I/O		D-3

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

备注 GF100-pin 塑封 QFP (14 × 20)
GC100-pin 塑封 LQFP (细间距) (14 × 14)

(1) 端口 DL 寄存器 (PDL)

复位后: 0000H (输出锁存)

R/W

地址:

PDL FFFFF004H, PDLL FFFFF004H, PDLH FFFFF005H

	15	14	13	12	11	10	9	8
PDL (PDLH)	PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8

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

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

备注

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

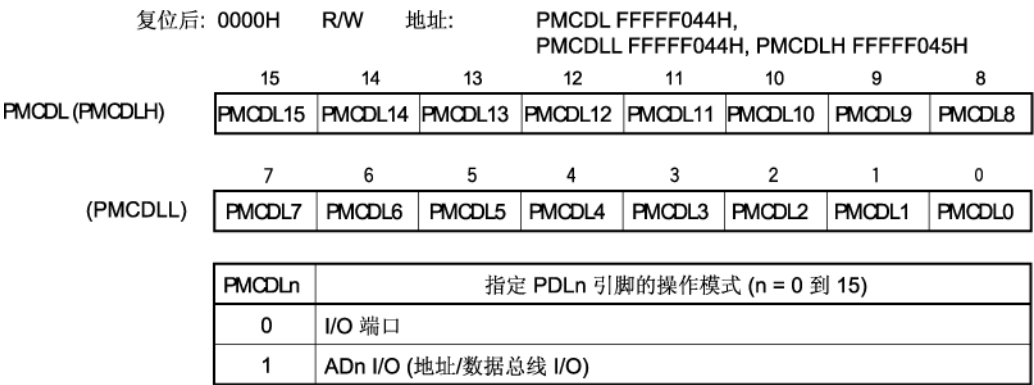
(2) 端口 DL 模式寄存器 (PMDL)

复位后: FFFFH		R/W	地址:		PMDL FFFFF024H, PMDLL FFFFF024H, PMDLH FFFFF025H			
PMDL (PMDLH)	15	14	13	12	11	10	9	8
	PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8
(PMDLL)	7	6	5	4	3	2	1	0
	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0
PMDLn		I/O 模式控制 (n = 0 到 15)						
0		输出模式						
1		输入模式						

备注

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

(3) 端口 DL 模式控制寄存器 (PMCDL)



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

备注

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

4.4 结构框图

图 4-3. A-1 类型的结构框图

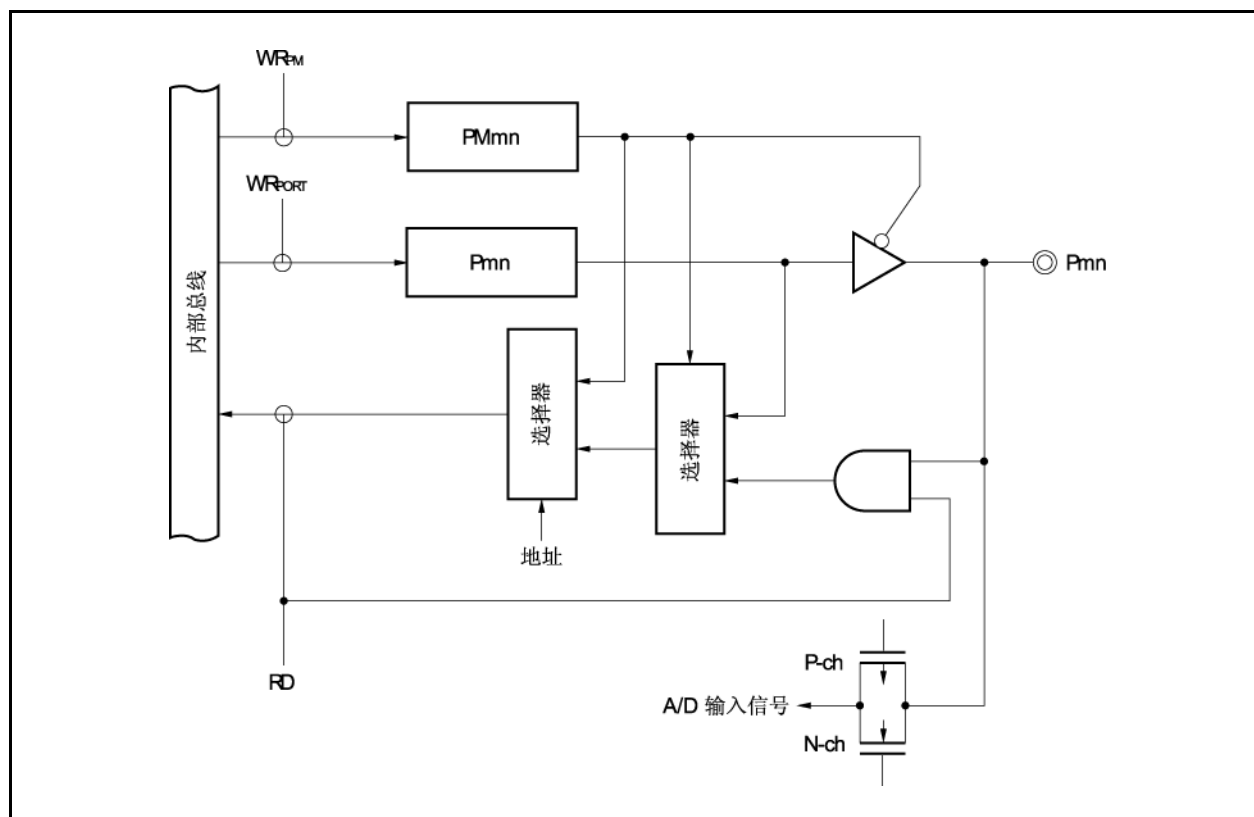


图 4-4. A-2 类型的结构框图

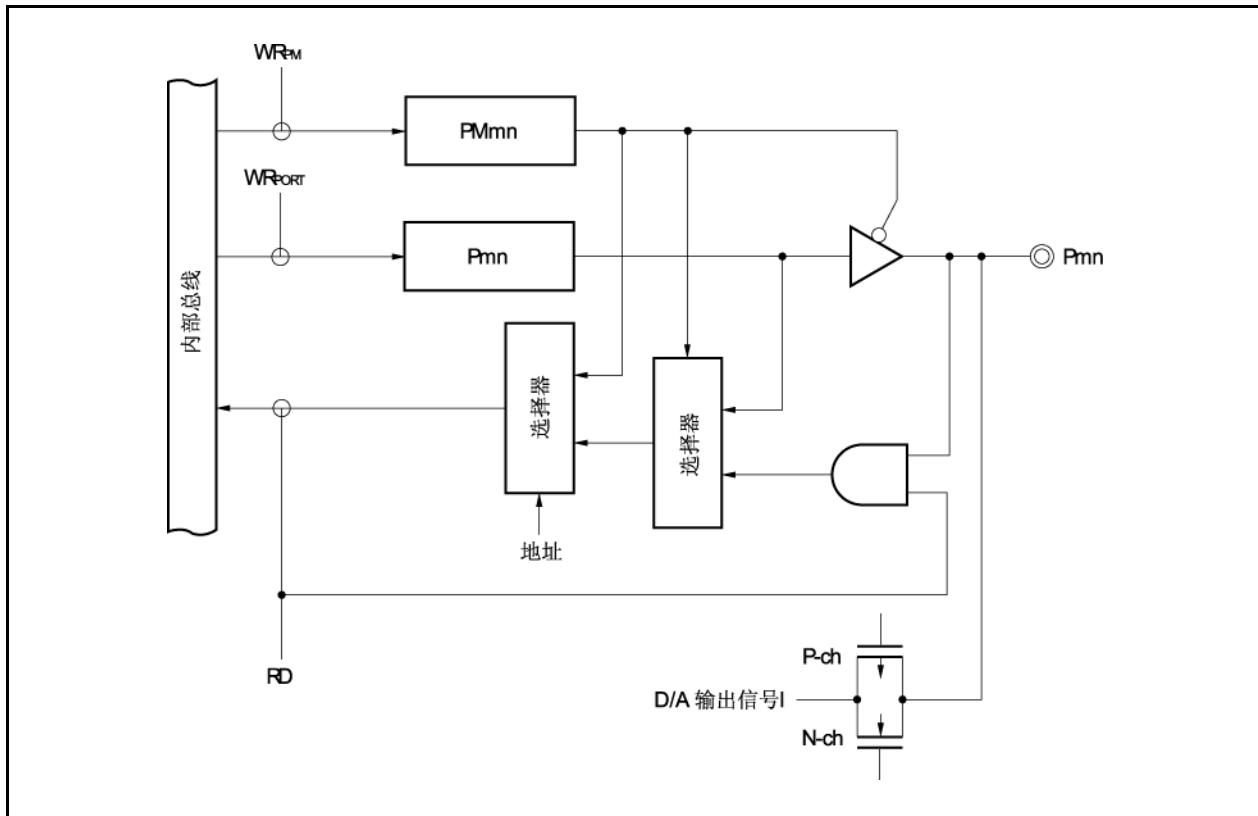


图 4-5. D-1 类型的结构框图

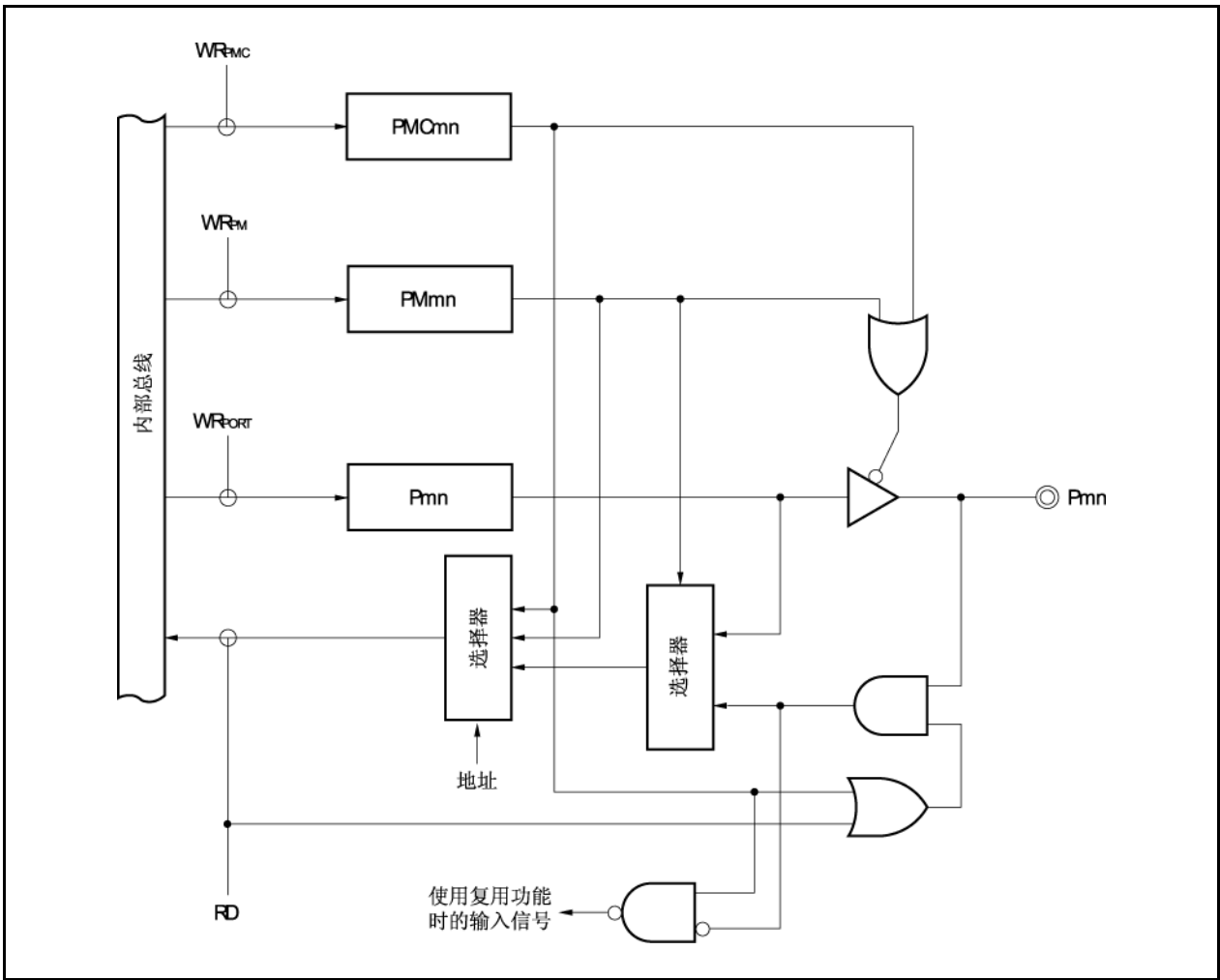


图 4-6. D-2 类型的结构框图

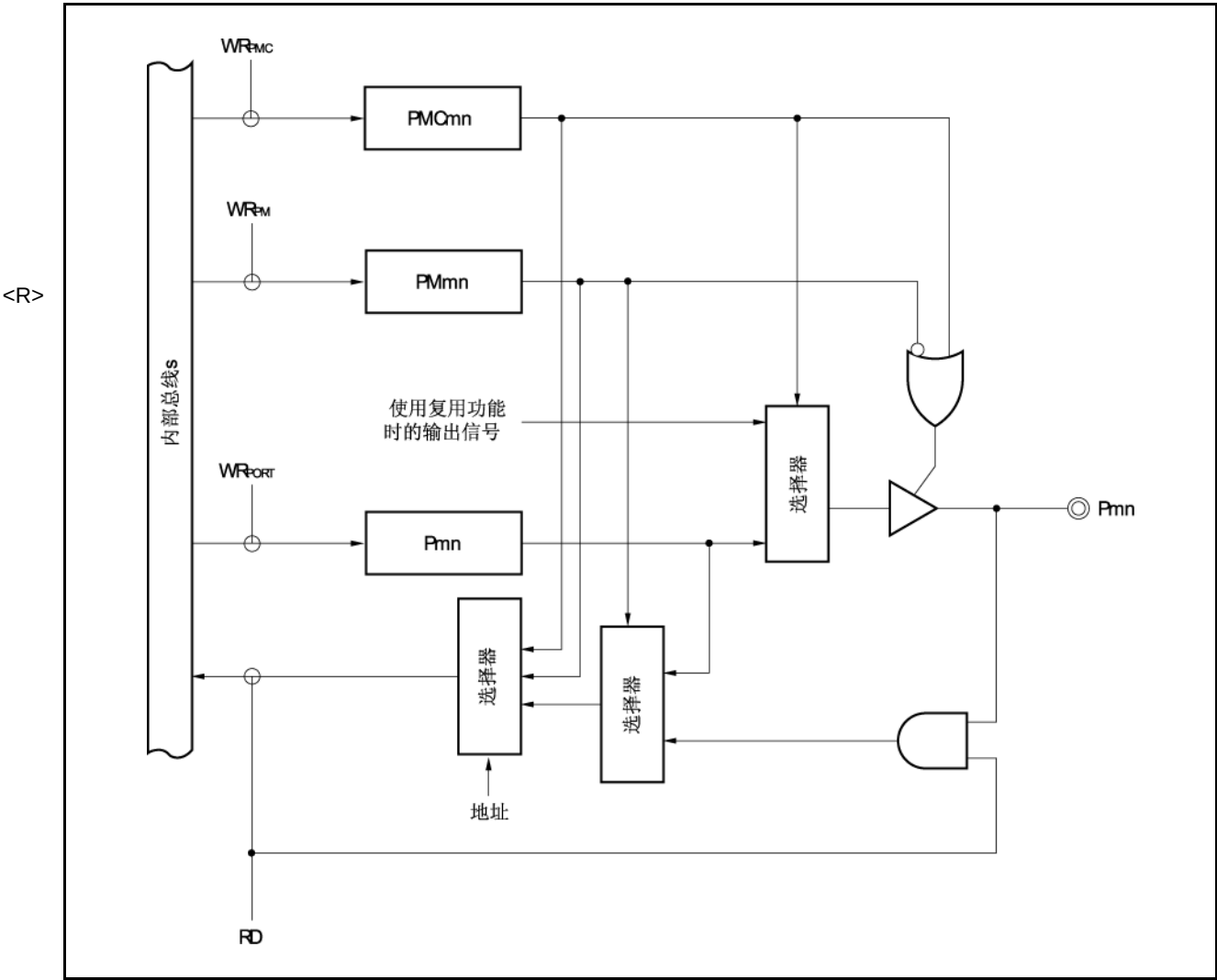


图 4-7. D-3 类型的结构框图

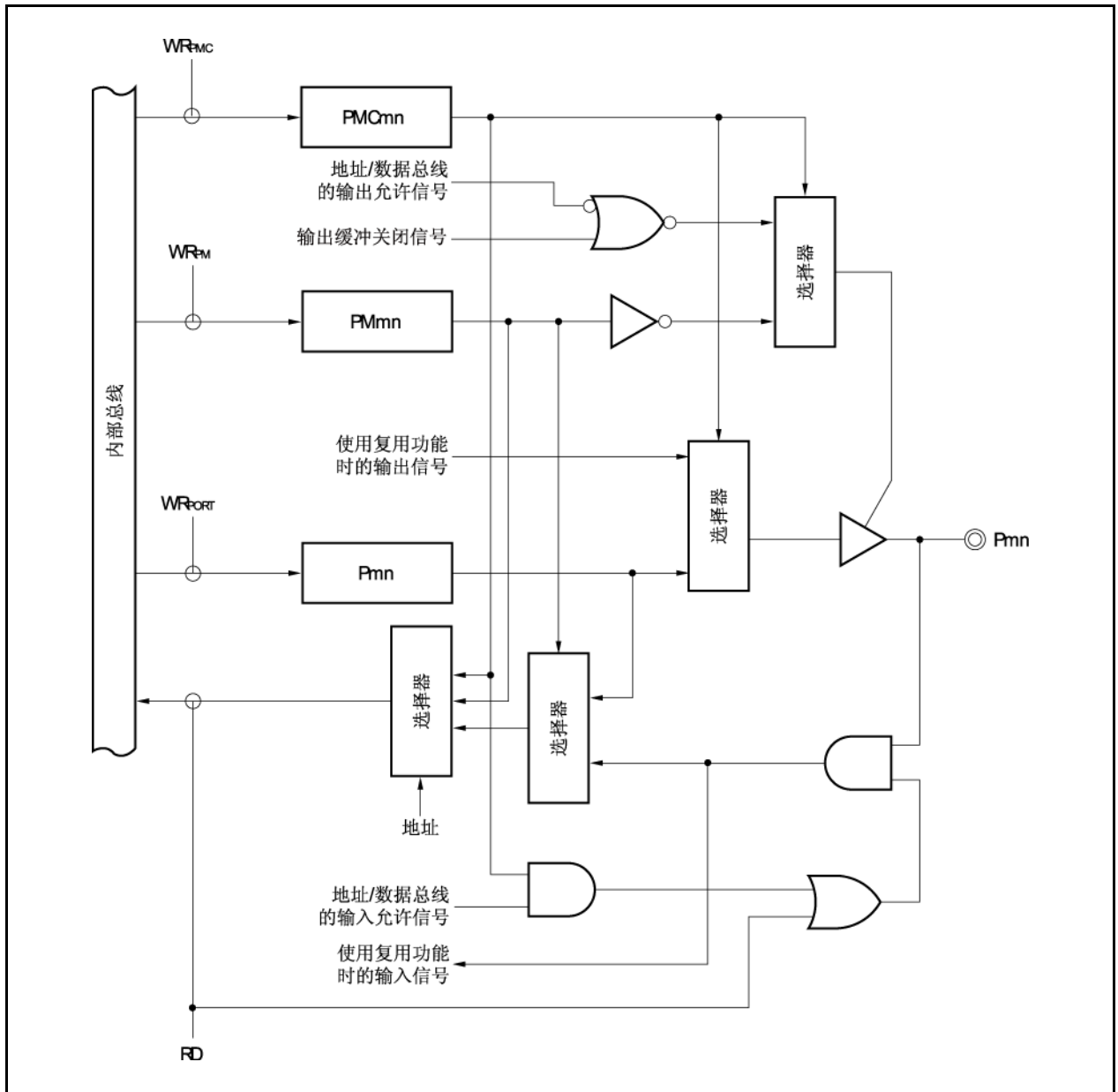


图 4-8. E-3 类型的结构框图

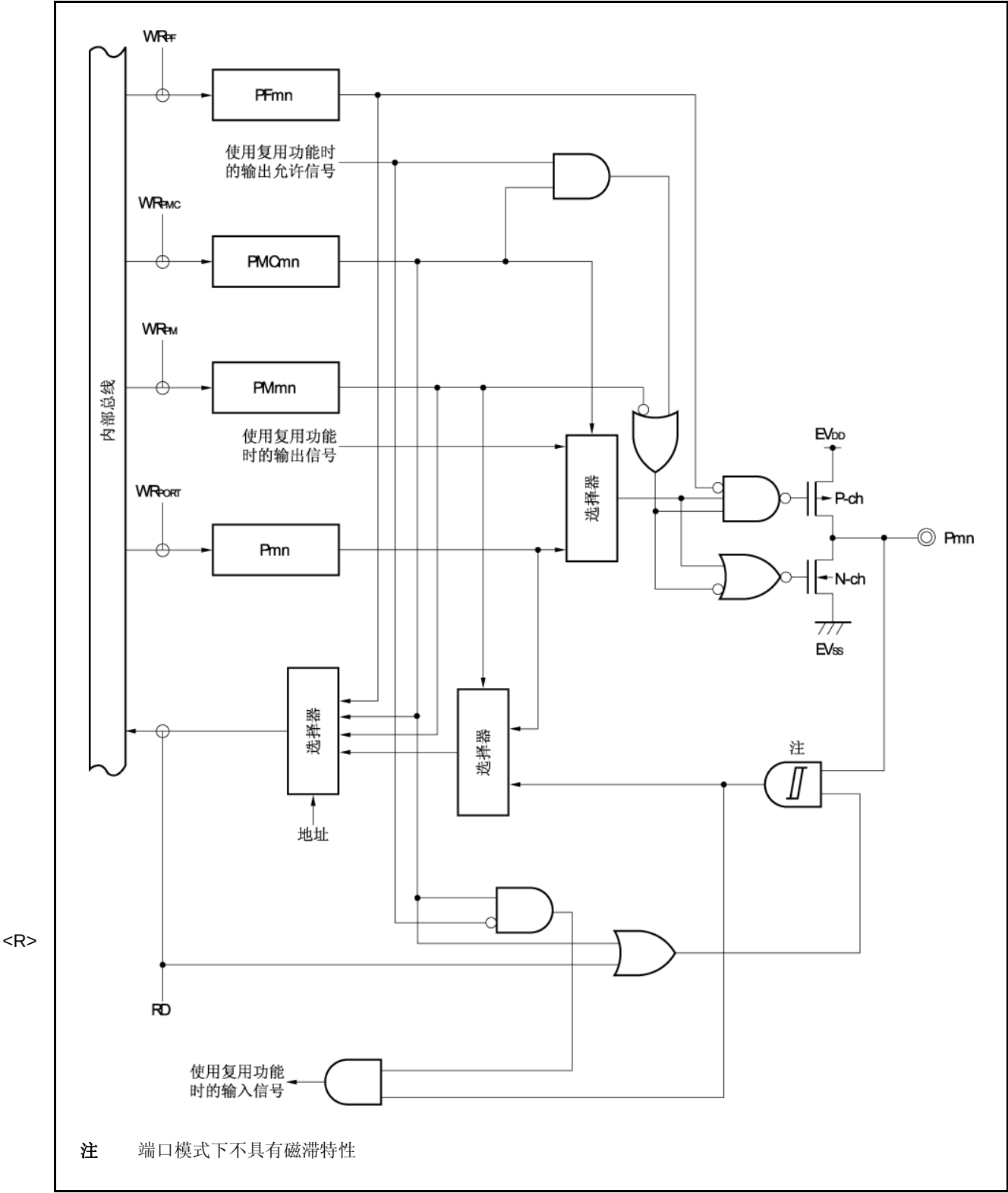


图 4-9. G-1 类型的结构框图



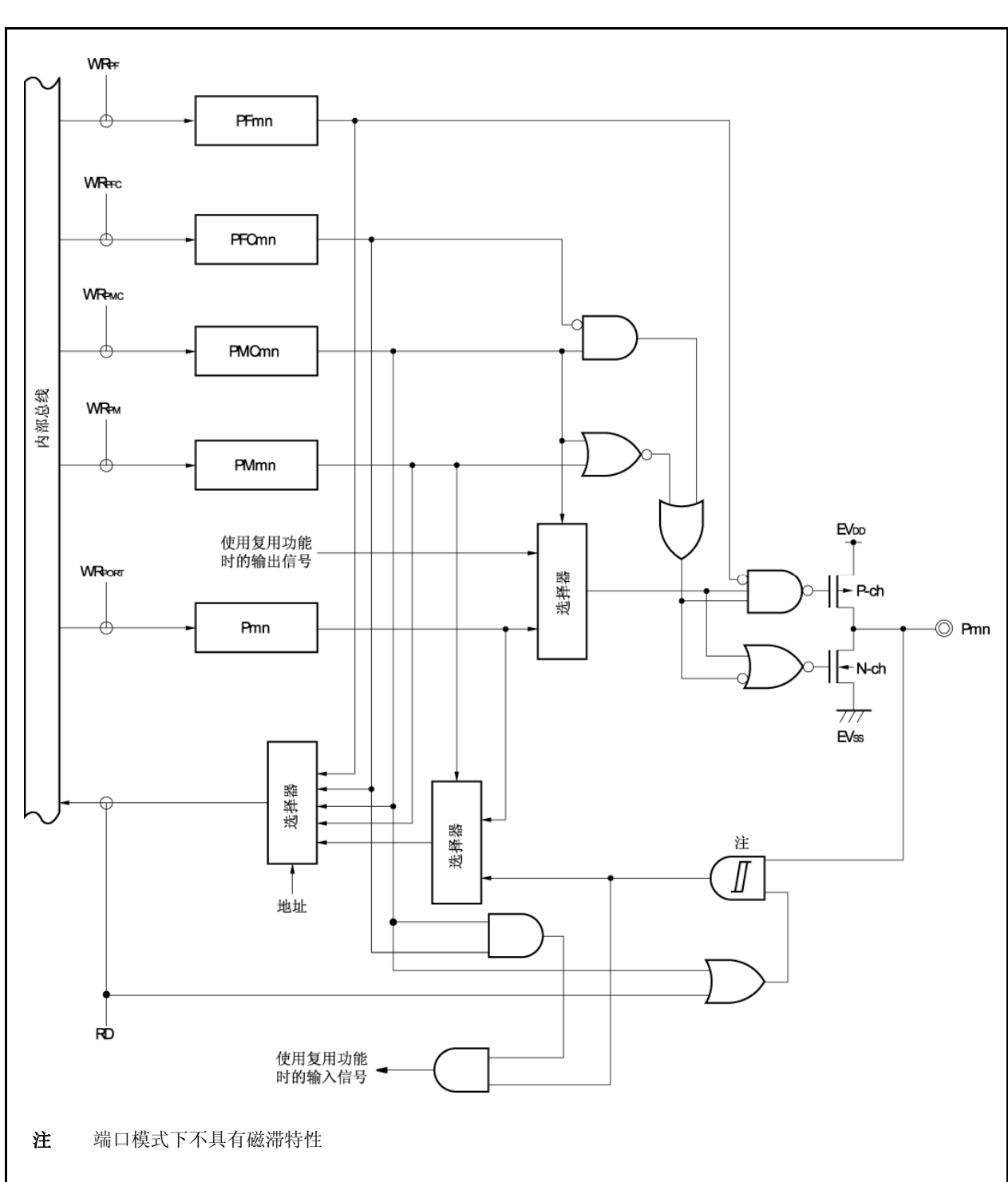


图 4-11. G-3 类型的结构框图

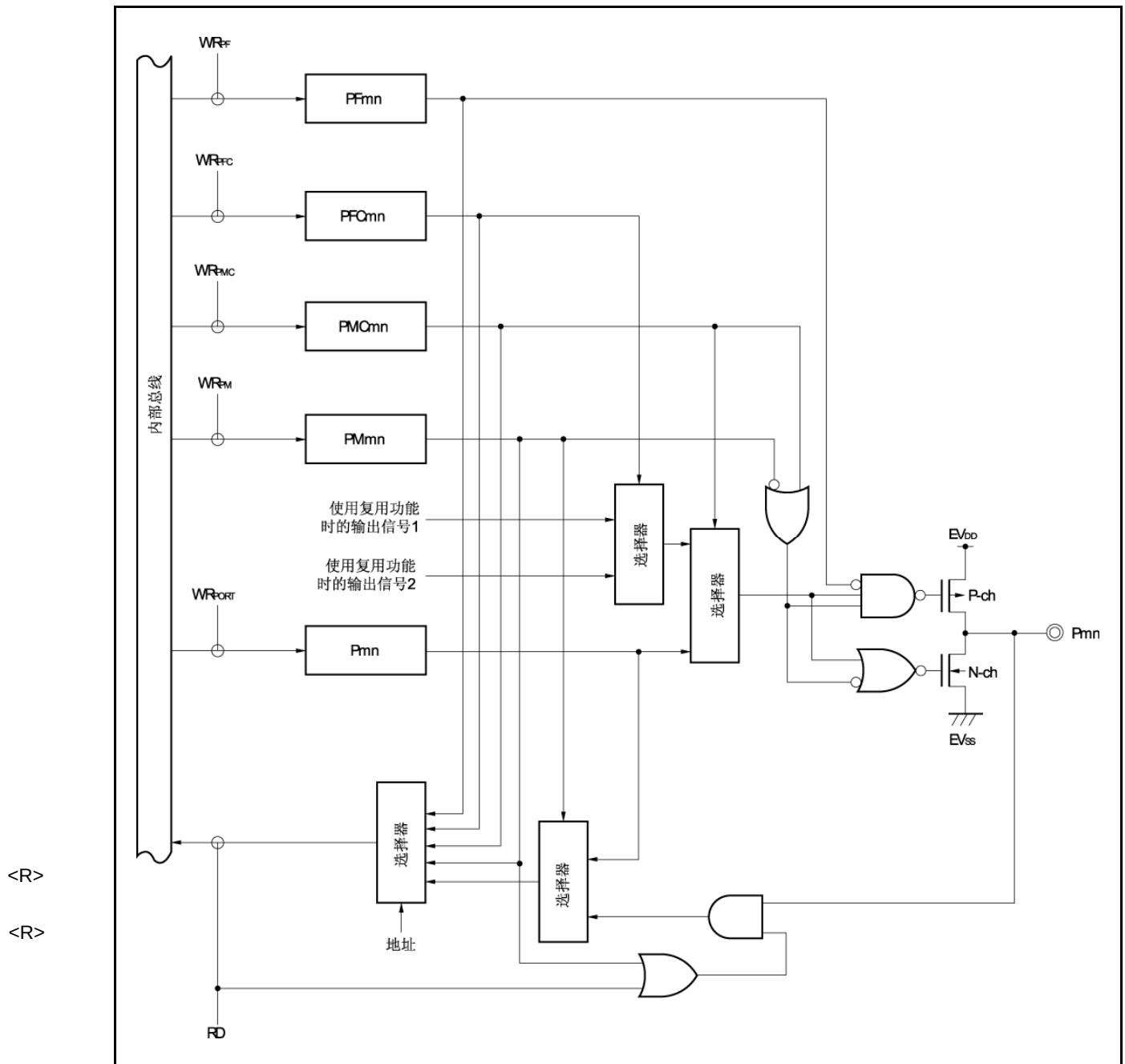


图 4-12. G-4 类型的结构框图

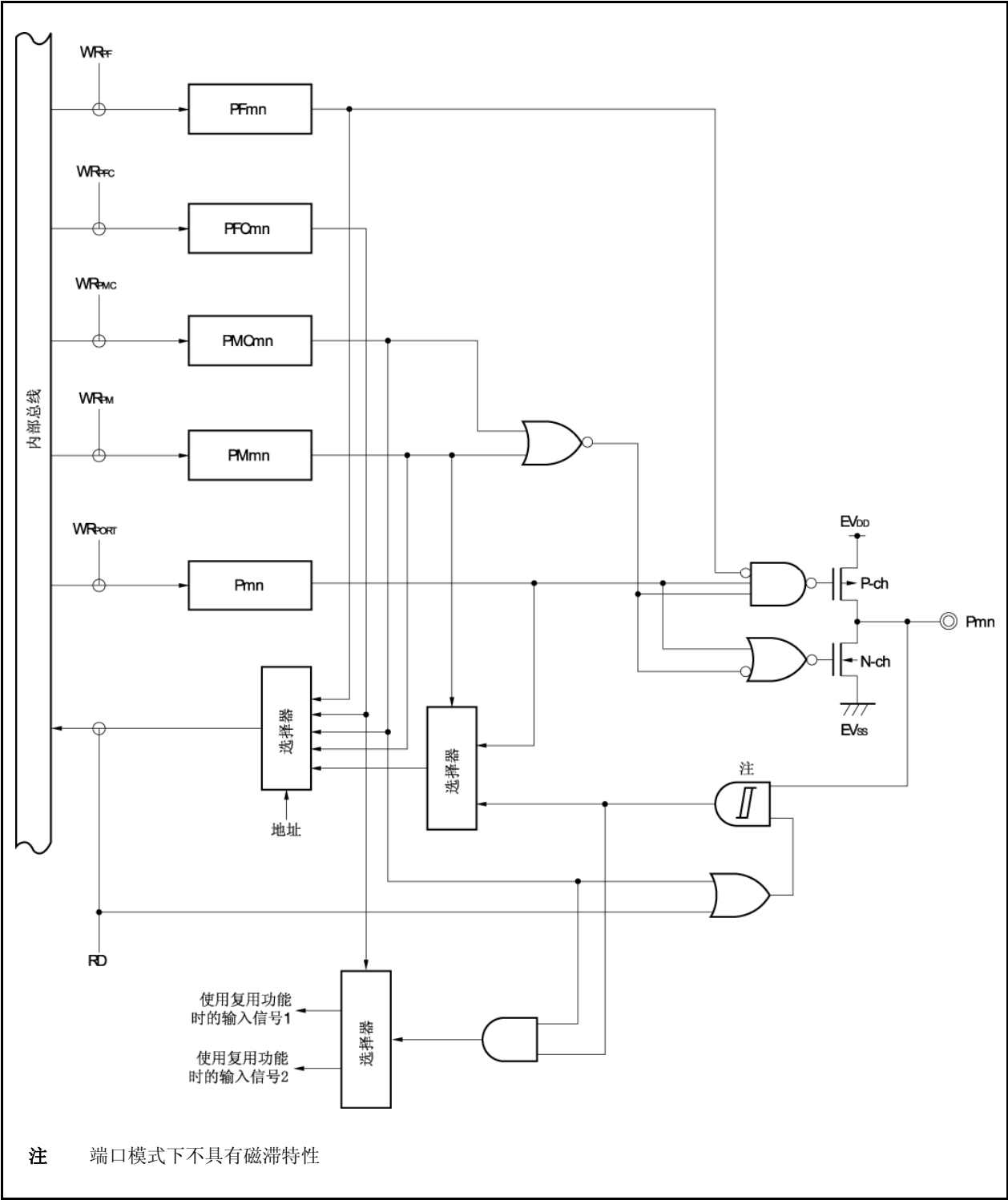


图 4-13. G-5 类型的结构框图



图 4-14. G-6 类型的结构框图

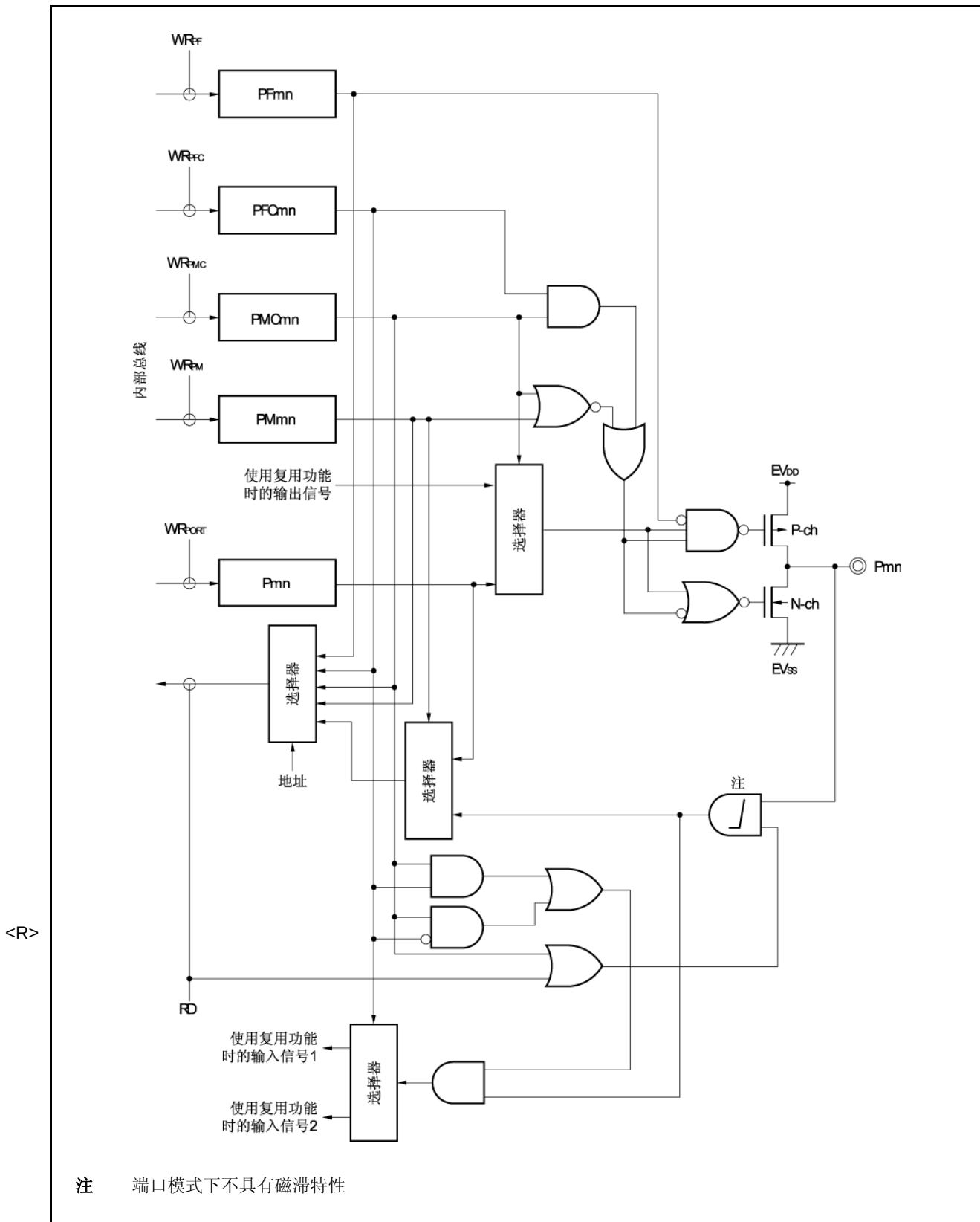


图 4-15. G-12 类型的结构框图



图 4-16. L-1 类型的结构框图

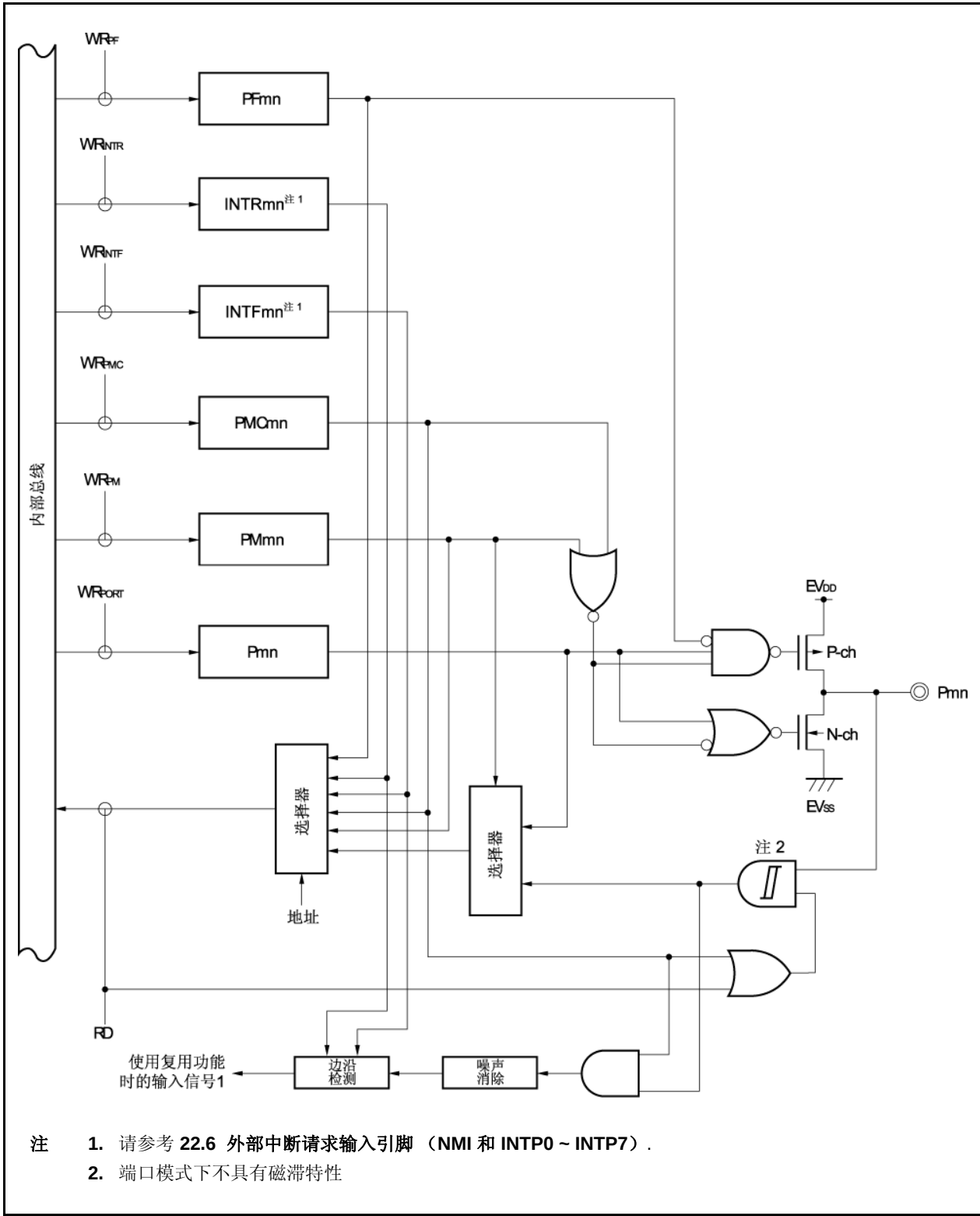


图 4-17. N-1 类型的结构框图

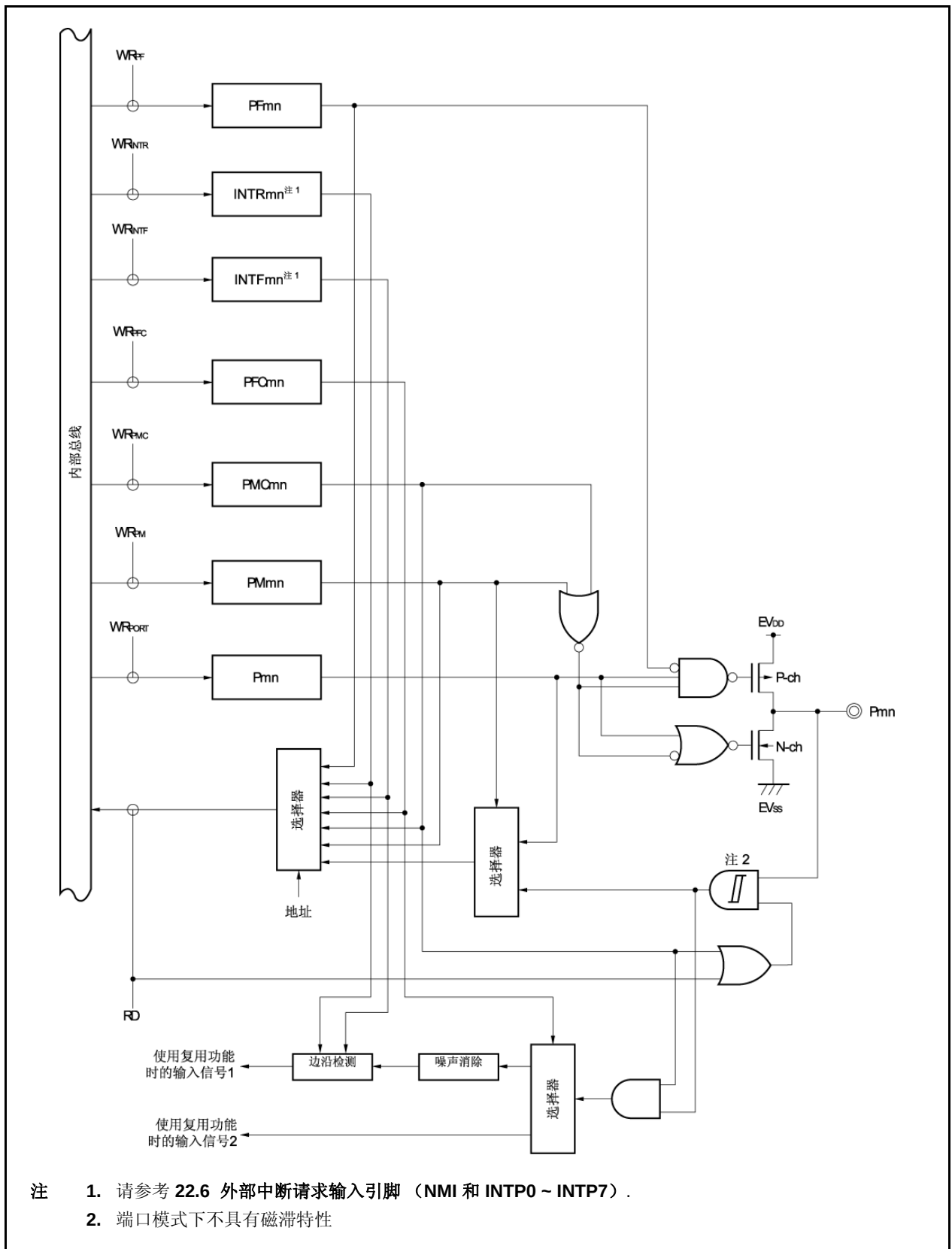


图 4-18. N-2 类型的结构框图



2. 端口模式下不具有磁滞特性

Journal of Management Education 36(7) 809–824

图 4-19. N-3 类型的结构框图

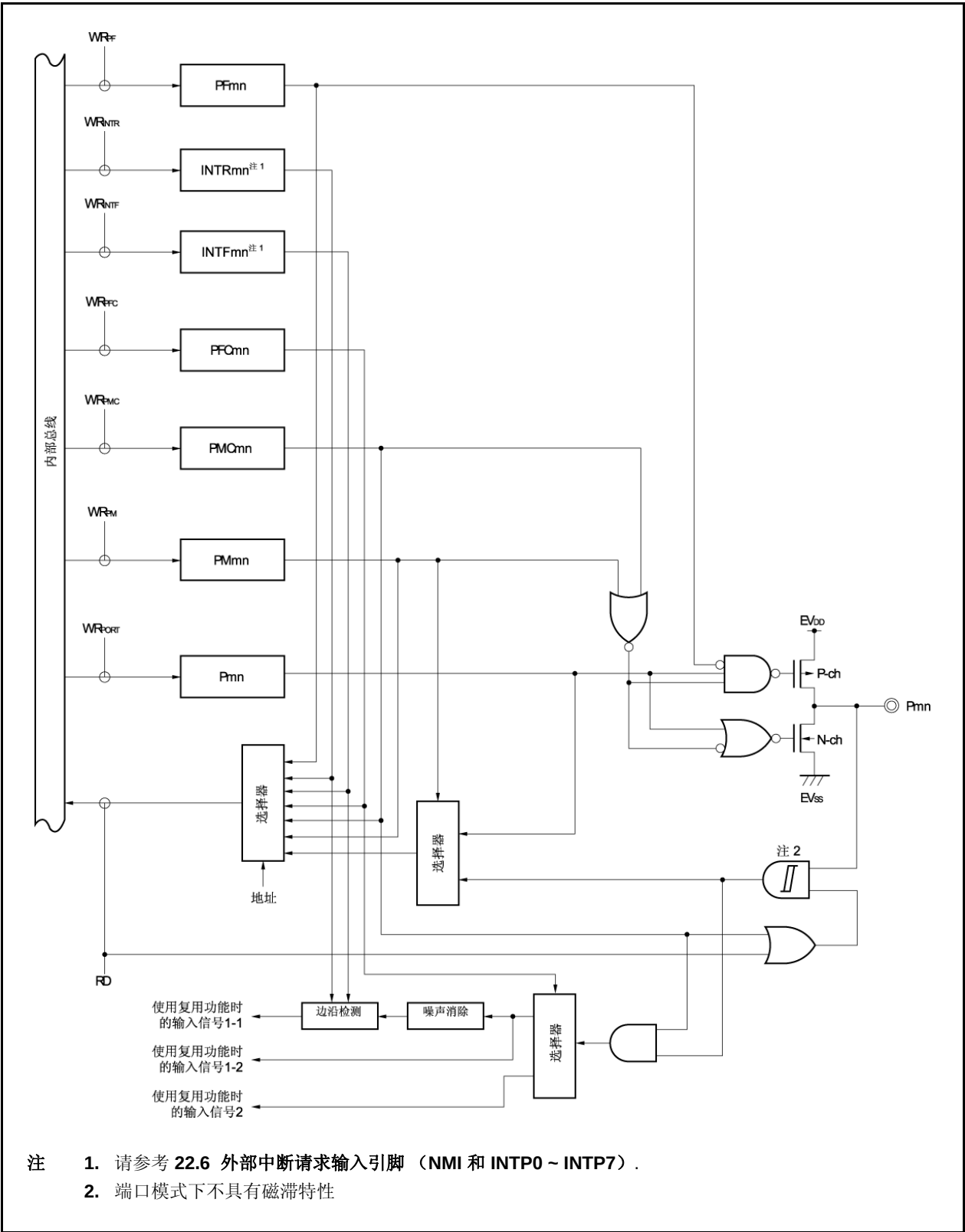


图 4-20. U-1 类型的结构框图

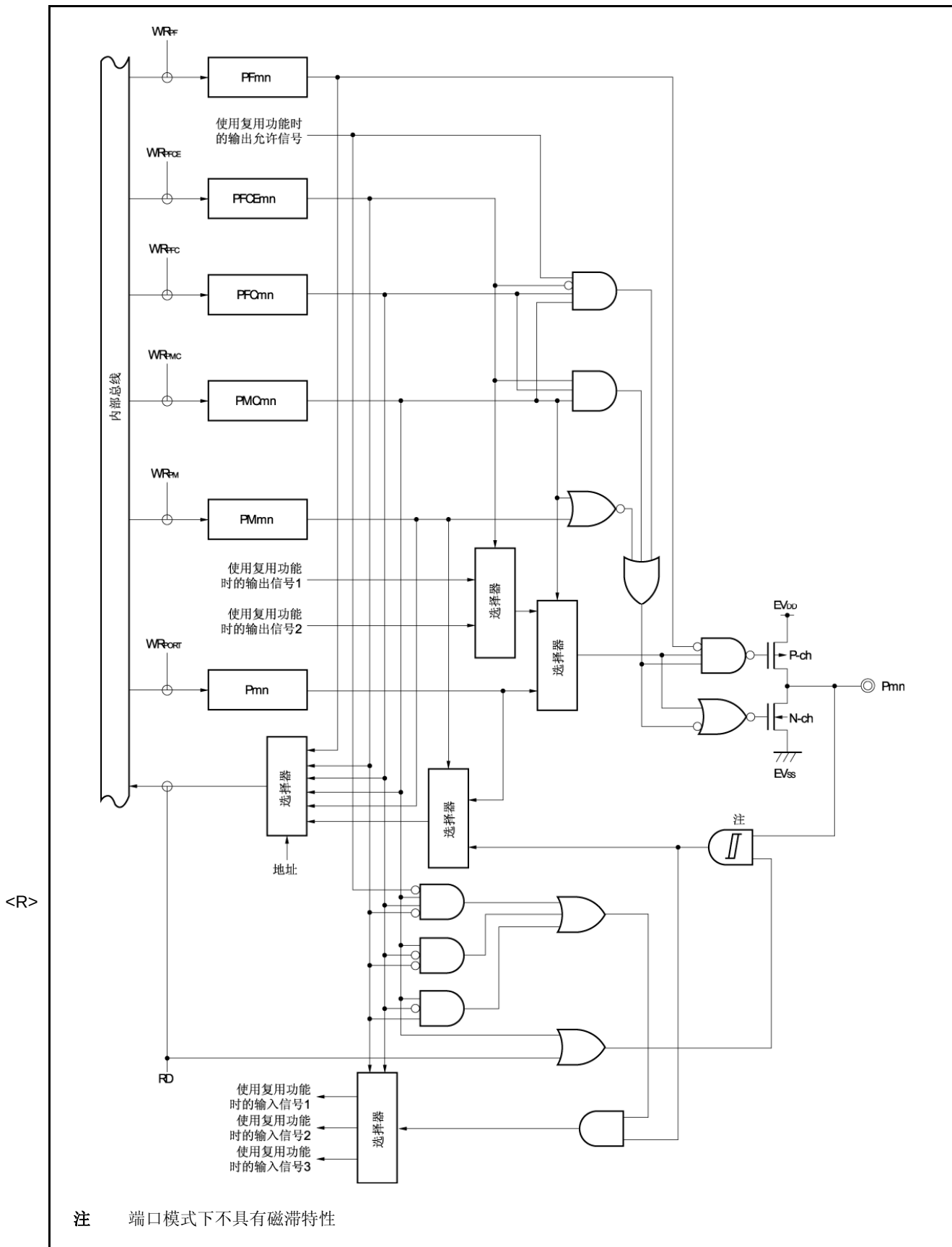


图 4-21. U-5 类型的结构框图

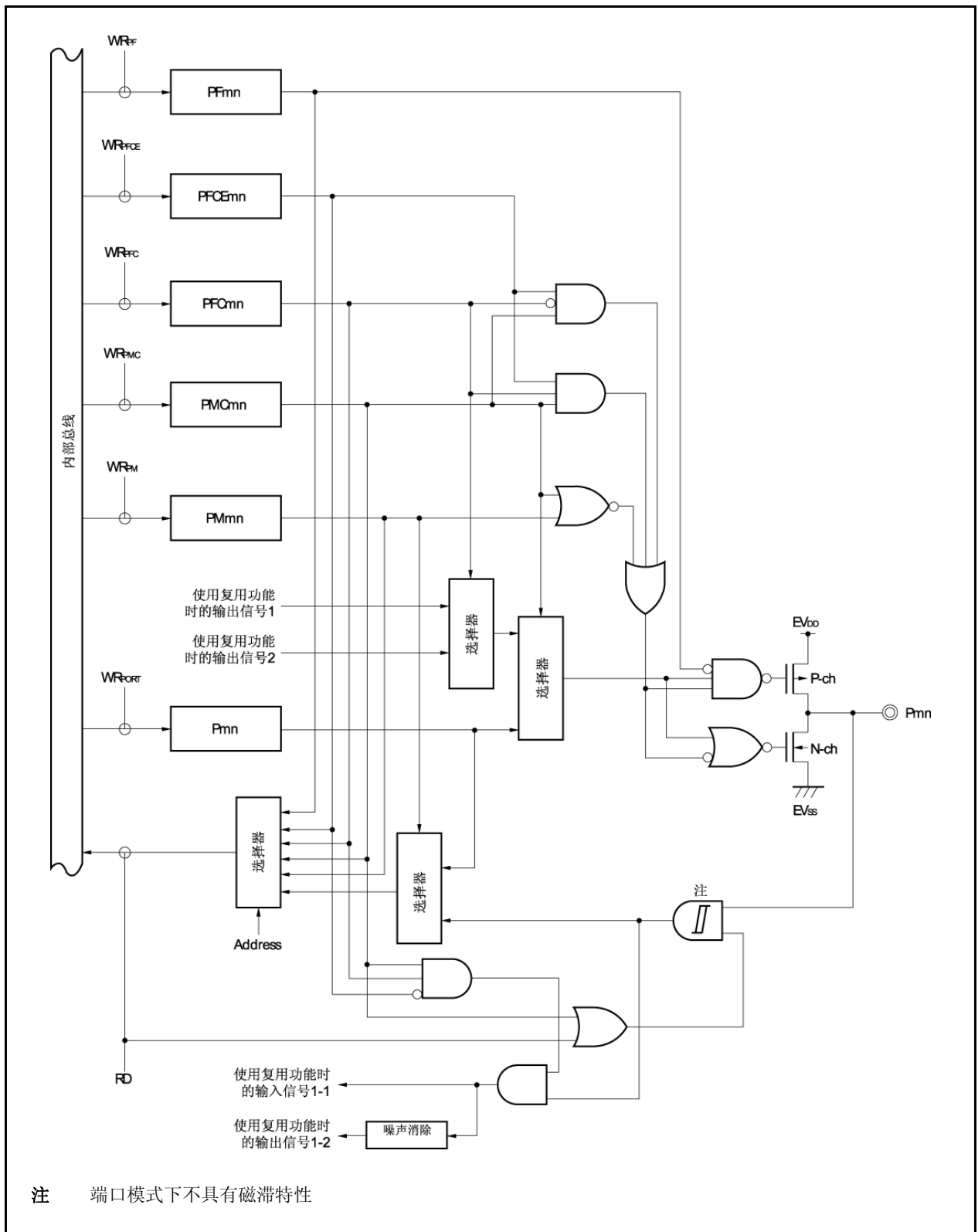


图 4-22. U-6 类型的结构框图

图 4-23. U-7 类型的结构框图

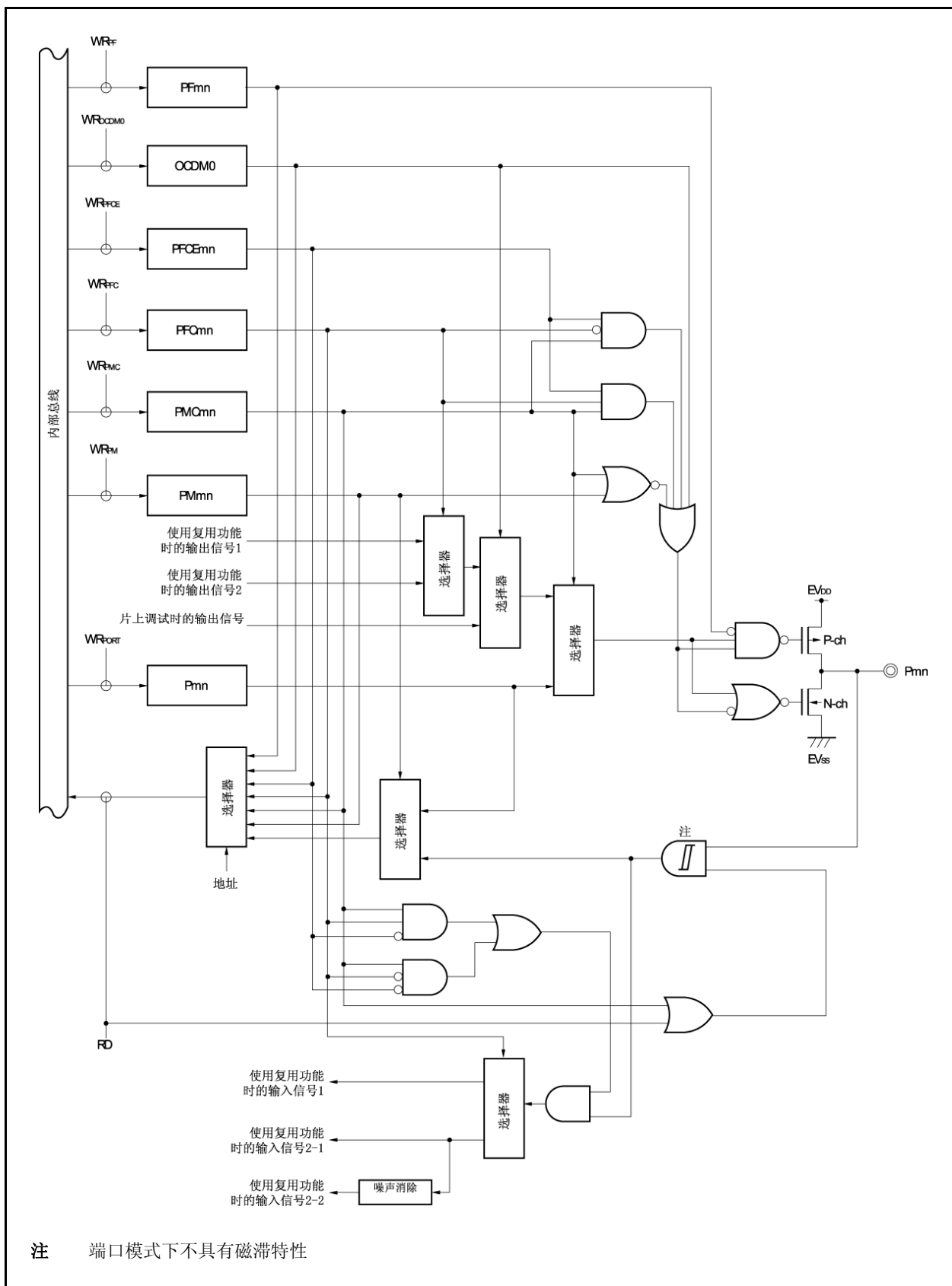


图 4-24. U-8 类型的结构框图

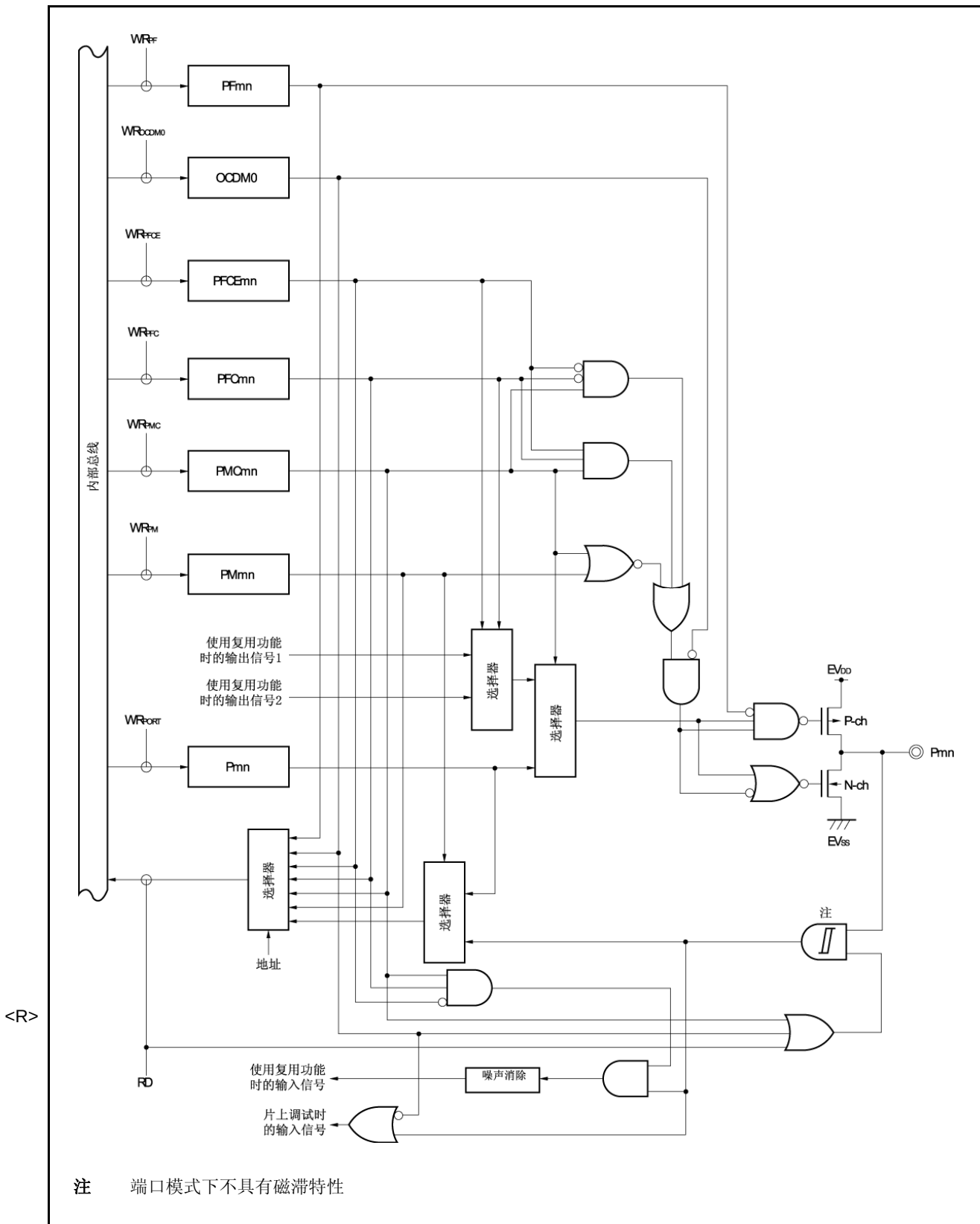


图 4-25. U-9 类型的结构框图

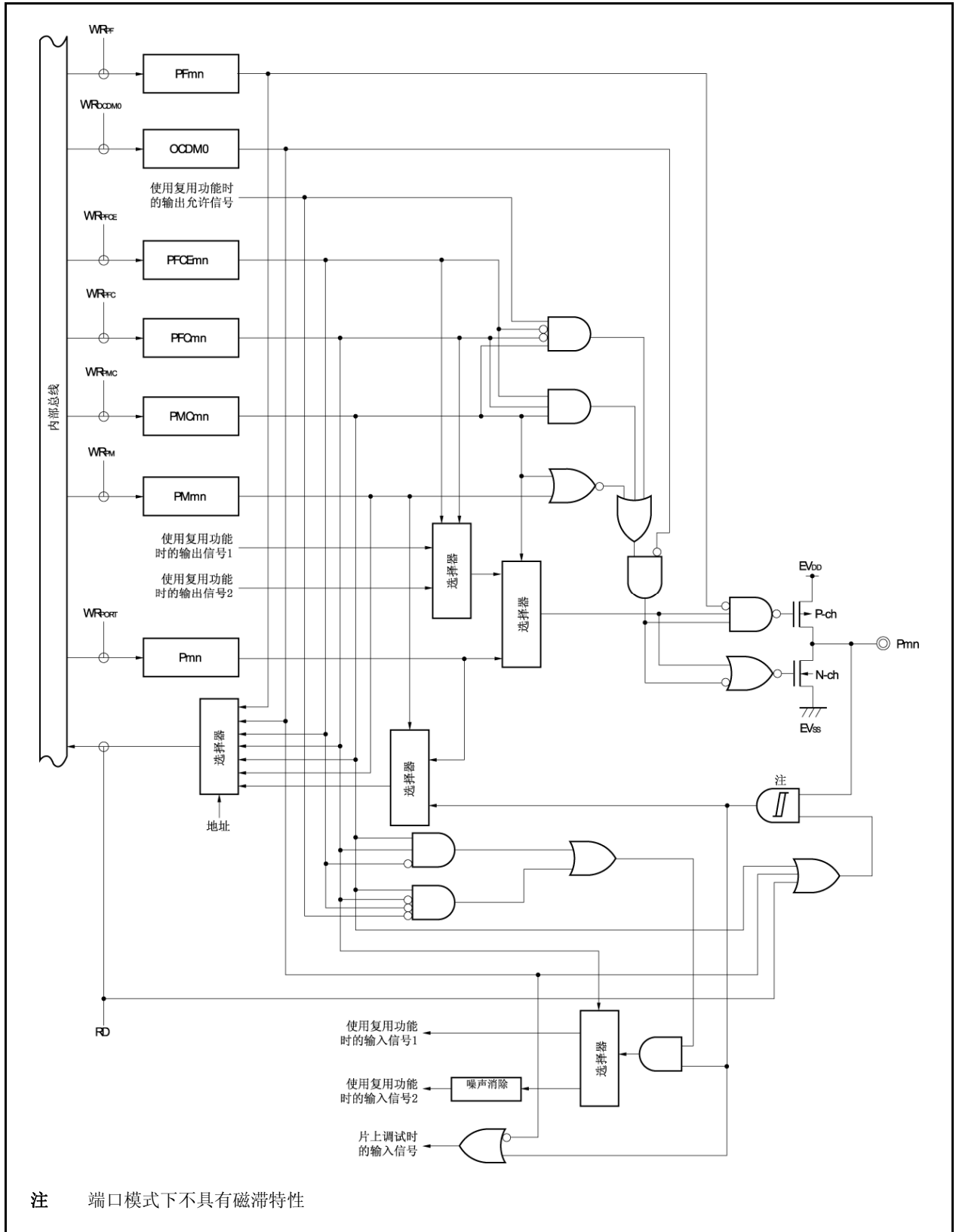


图 4-26. U-10 类型的结构框图

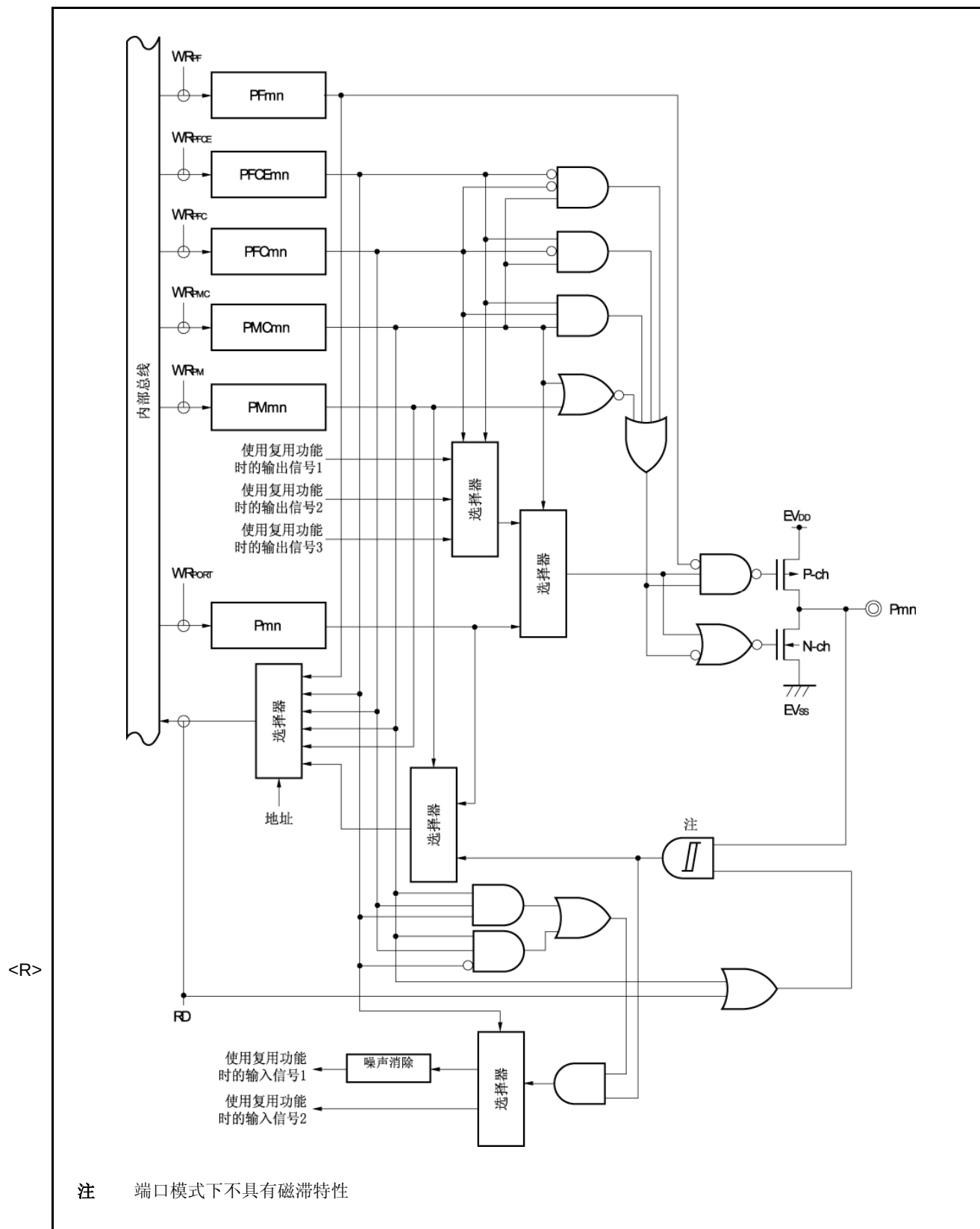


图 4-27. U-11 类型的结构框图



图 4-28. U-12 类型的结构框图

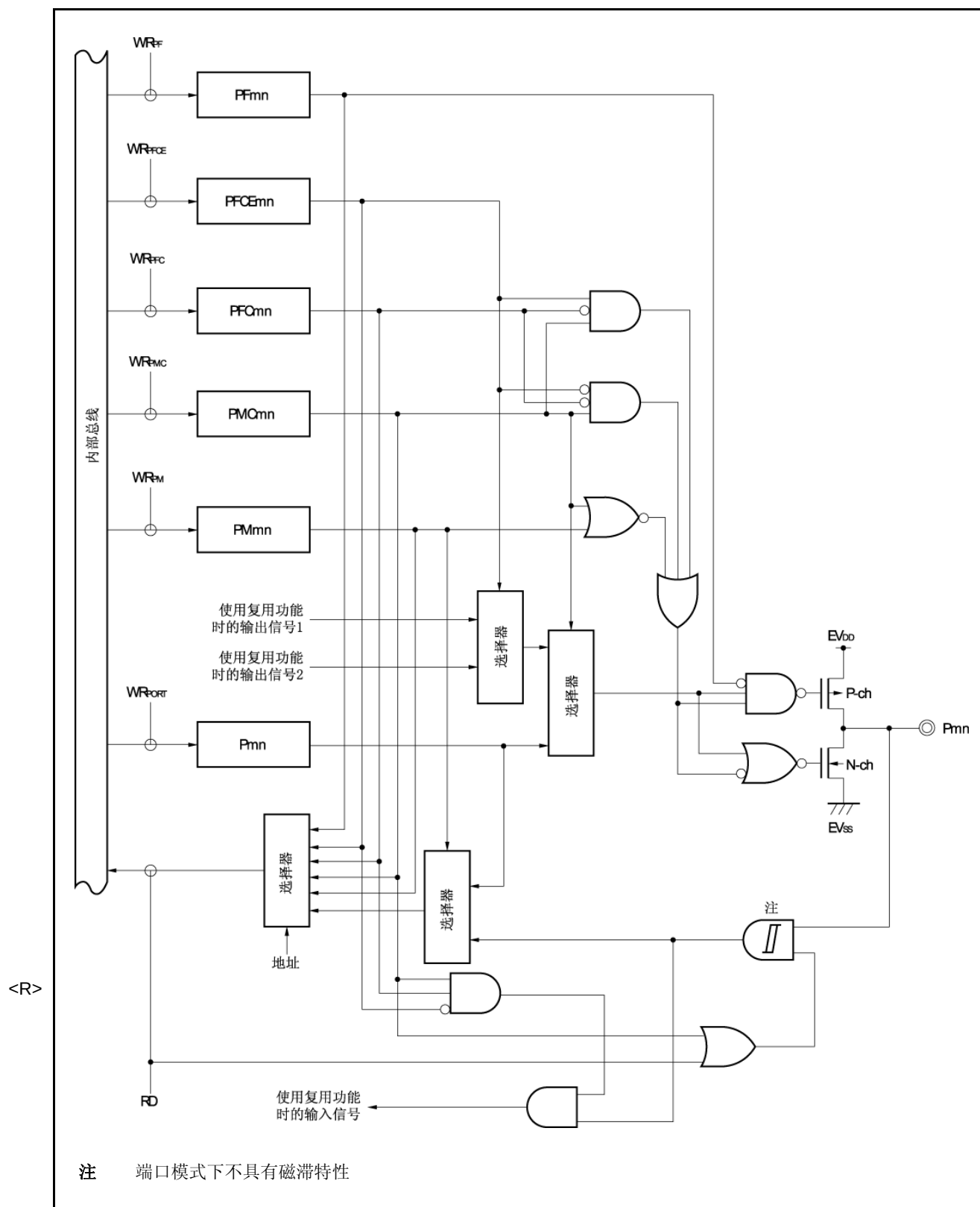


图 4-29. U-13 类型的结构框图

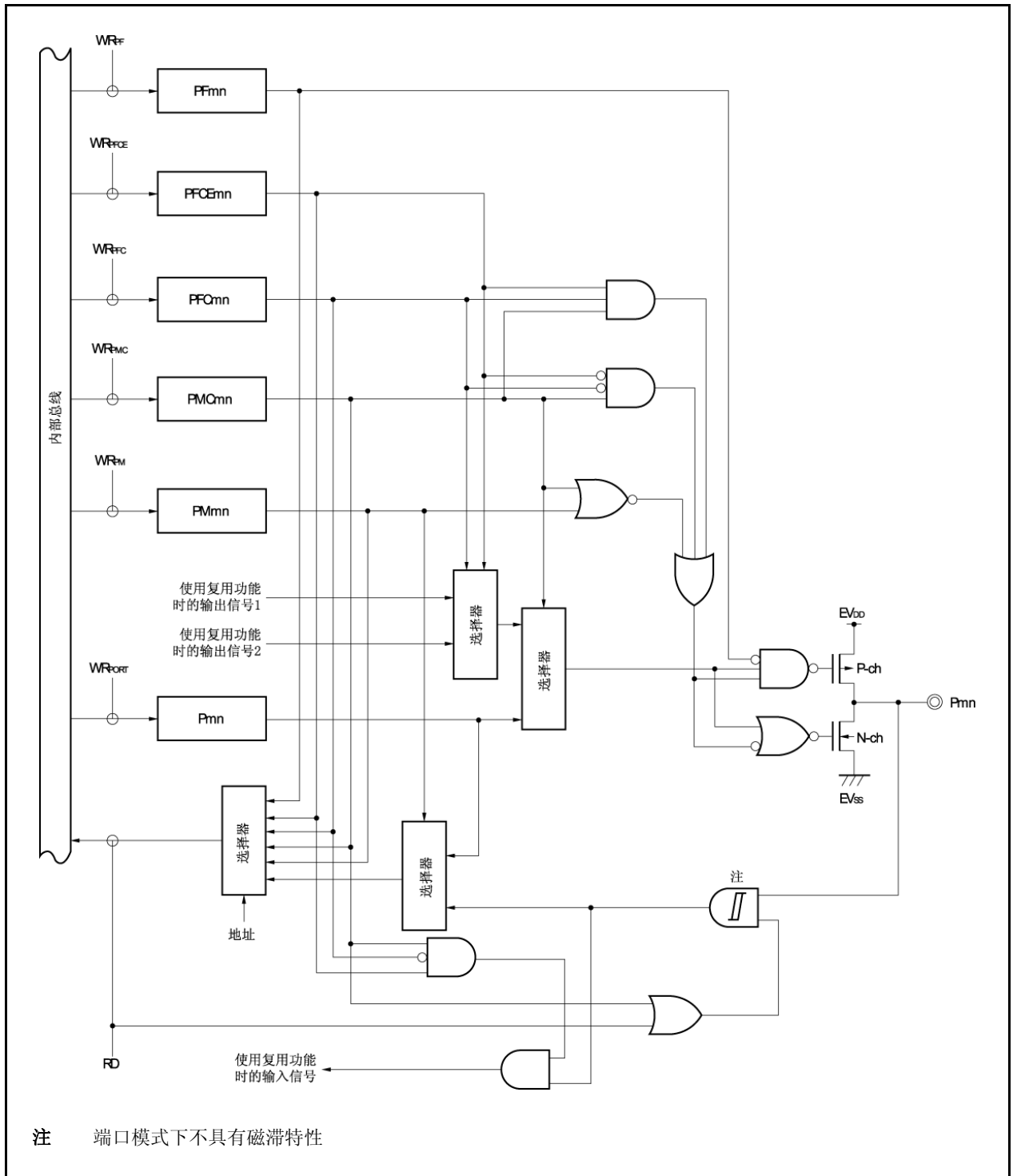


图 4-30. U-14 类型的结构框图

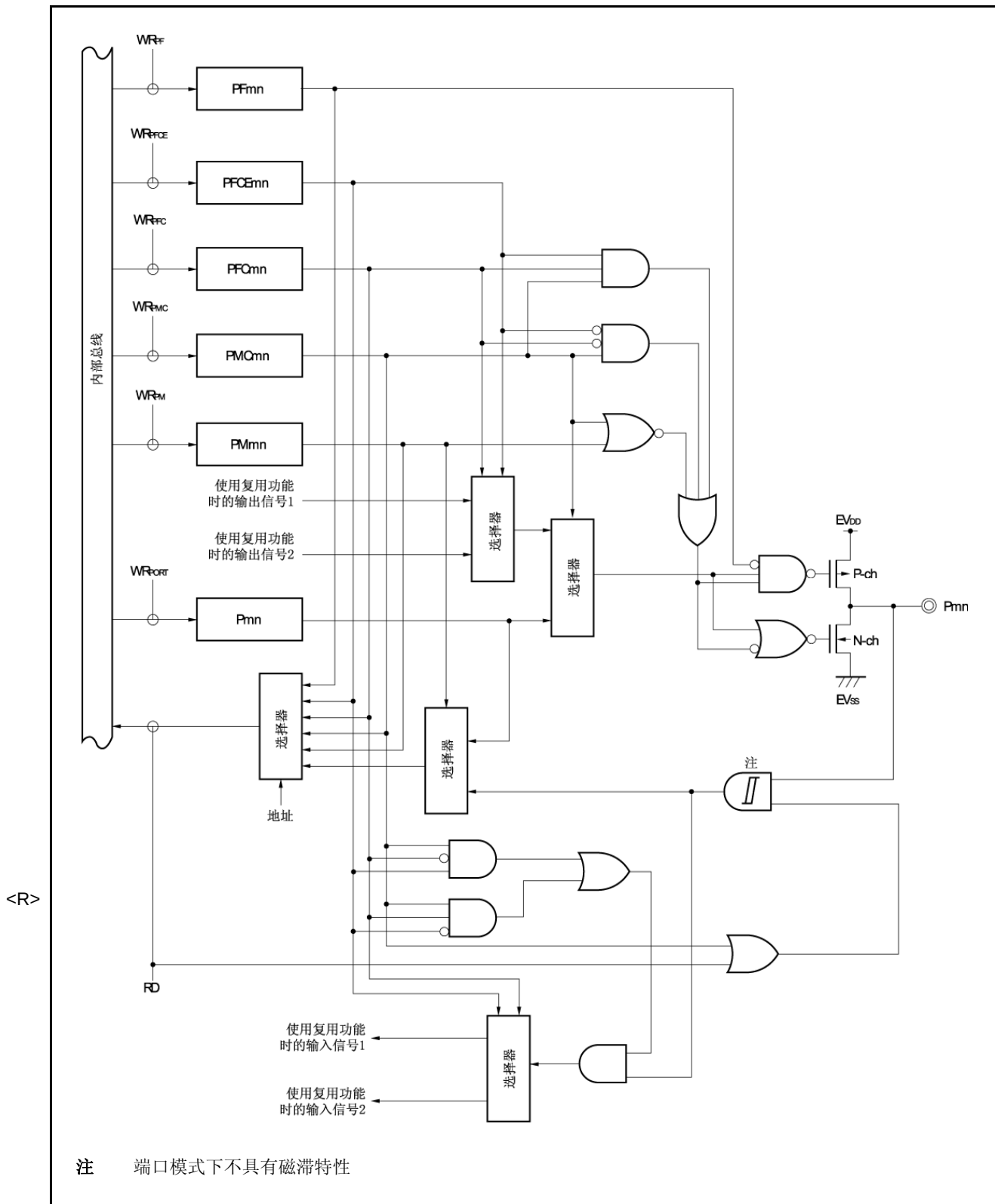
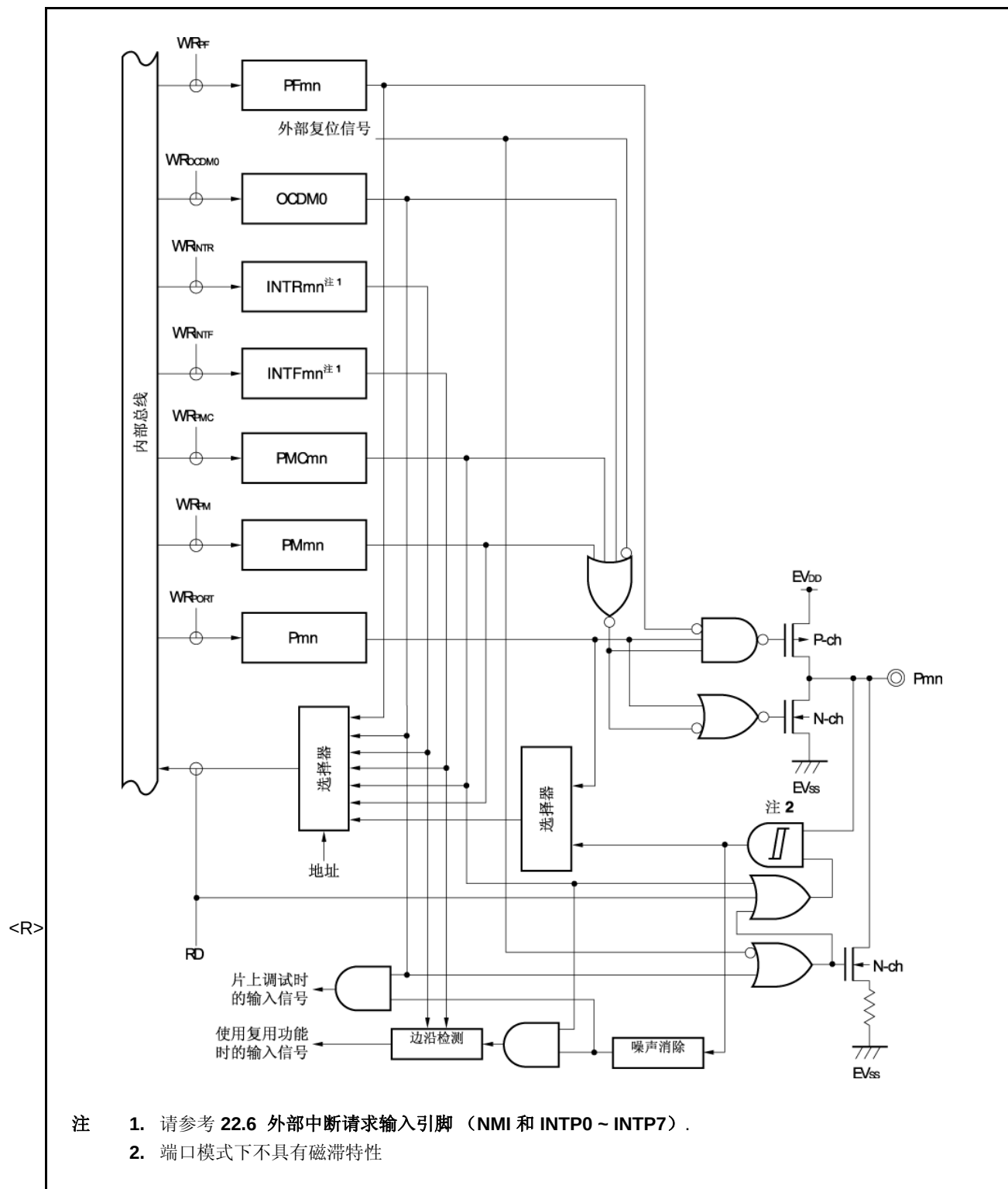


图 4-31. U-15 类型的结构框图



图 4-32. AA-1 类型的结构框图



4.5 当使用复用功能时的端口寄存器设置

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

表 4-15. 端口引脚用作复用功能引脚 (1/7)

管脚名称	复用功能		Pn 寄存器的 Pnx 位	PM 寄存器的 PMx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其他位 (寄存器)
	名称	I/O						
P02	NMI	输入	P02 = 不要求设置	PM02 = 不要求设置	PMC02 = 1	-	-	
P03	INTP0	输入	P03 = 不要求设置	PM03 = 不要求设置	PMC03 = 1	-	PFC03 = 0	
	ADTRG	输入	P03 = 不要求设置	PM03 = 不要求设置	PMC03 = 1	-	PFC03 = 1	
P04	INTP1	输入	P04 = 不要求设置	PM04 = 不要求设置	PMC04 = 1	-	-	
P05	INTP2	输入	P05 = 不要求设置	PM05 = 不要求设置	PMC05 = 1	-	-	
	DRST ^{注 1}	输入	P05 = 不要求设置	PM05 = 不要求设置	PMC05 = 不要求设置	-	-	OCDM0 (OCDM) = 1
P06	INTP3	输入	P06 = 不要求设置	PM06 = 不要求设置	PMC06 = 1	-	-	
P10	AN00	输出	P10 = 不要求设置	PM10 = 1	-	-	-	
P11	AN01	输出	P11 = 不要求设置	PM11 = 1	-	-	-	
P30	TXDA0	输出	P30 = 不要求设置	PM30 = 不要求设置	PMC30 = 1	-	PFC30 = 0	
	SOB4	输出	P30 = 不要求设置	PM30 = 不要求设置	PMC30 = 1	-	PFC30 = 1	
P31	RXDA0	输入	P31 = 不要求设置	PM31 = 不要求设置	PMC31 = 1	-	注 2, PFC31 = 0	
	INTP7	输入	P31 = 不要求设置	PM31 = 不要求设置	PMC31 = 1	-	注 2, PFC31 = 0	
	SIB4	输入	P31 = 不要求设置	PM31 = 不要求设置	PMC31 = 1	-	PFC31 = 1	
P32	ASCKA0	输入	P32 = 不要求设置	PM32 = 不要求设置	PMC32 = 1	PFCE32 = 0	PFC32 = 0	
	SCKB4	I/O	P32 = 不要求设置	PM32 = 不要求设置	PMC32 = 1	PFCE32 = 0	PFC32 = 1	
	TIP00	输入	P32 = 不要求设置	PM32 = 不要求设置	PMC32 = 1	PFCE32 = 1	PFC32 = 0	
	TOP00	输出	P32 = 不要求设置	PM32 = 不要求设置	PMC32 = 1	PFCE32 = 1	PFC32 = 1	
P33	TIP01	输入	P33 = 不要求设置	PM33 = 不要求设置	PMC33 = 1	-	PFC33 = 0	
	TOP01	输出	P33 = 不要求设置	PM33 = 不要求设置	PMC33 = 1	-	PFC33 = 1	
P34	TIP10	输入	P34 = 不要求设置	PM34 = 不要求设置	PMC34 = 1	-	PFC34 = 0	
	TOP10	输出	P34 = 不要求设置	PM34 = 不要求设置	PMC34 = 1	-	PFC34 = 1	
P35	TIP11	输入	P35 = 不要求设置	PM35 = 不要求设置	PMC35 = 1	-	PFC35 = 0	
	TOP11	输出	P35 = 不要求设置	PM35 = 不要求设置	PMC35 = 1	-	PFC35 = 1	

注 1. 仅限于 flash 存储器版本

2. INTP7 引脚和 RXDA0 引脚为一组复用功能引脚。当用作 RXDA0 引脚功能时，需要禁止 INTP7 复用功能引脚的边沿检测（将 INTF3.INTF31 位和 INTR3.INTR31 位清 0）。当用作 INTP8 引脚功能时，需要停止 UARTA3 接收（将 UA0CTL3.UA3RXE 位清 0）。

注意事项 当使用引脚 P10 和 P11 作为 I/O 端口或者 D/A 输出引脚（AN00，AN01），D/A 输出电平不改变。

表 4-15. 端口引脚用作复用功能引脚 (2/7)

管脚名称	功能复用		Pn寄存器 的Pnx位	PMn寄存器 的PMnx位	PMCn寄存器 的PMCnx位	PFCEn寄存器 的PFCEnx位	PFCn寄存器 的PFCnx位	其它位 (寄存器)
	名称	I/O						
P36	CTXD0 ^{注1}	输出	P36 = 无需设置	PM36 = 无需设置	PMC36 = 1	-	PFC36 = 0	
	IETX0 ^{注2}	输出	P36 = 无需设置	PM36 = 无需设置	PMC36 = 1	-	PFC36 = 1	
P37	CRXD0 ^{注1}	输入	P37 = 无需设置	PM37 = 无需设置	PMC37 = 1	-	PFC37 = 0	
	IERX0 ^{注2}	输入	P37 = 无需设置	PM37 = 无需设置	PMC37 = 1	-	PFC37 = 1	
P38	TXDA2	输出	P38 = 无需设置	PM38 = 无需设置	PMC38 = 1	-	PFC38 = 0	
	SDA00 ^{注3}	I/O	P38 = 无需设置	PM38 = 无需设置	PMC38 = 1	-	PFC38 = 1	PF38 (PF3) = 1
P39	RXDA2	输入	P39 = 无需设置	PM39 = 无需设置	PMC39 = 1	-	PFC39 = 0	
	SCL00 ^{注3}	I/O	P39 = 无需设置	PM39 = 无需设置	PMC39 = 1	-	PFC39 = 1	PF39 (PF3) = 1
P40	SIB0	输入	P40 = 无需设置	PM40 = 无需设置	PMC40 = 1	-	PFC40 = 0	
	SDA01 ^{注3}	I/O	P40 = 无需设置	PM40 = 无需设置	PMC40 = 1	-	PFC40 = 1	PF40 (PF4) = 1
P41	SOB0	输出	P41 = 无需设置	PM41 = 无需设置	PMC41 = 1	-	PFC41 = 0	
	SCL01 ^{注3}	I/O	P41 = 无需设置	PM41 = 无需设置	PMC41 = 1	-	PFC41 = 1	PF41 (PF4) = 1
P42	SCKB0	I/O	P42 = 无需设置	PM42 = 无需设置	PMC42 = 1	-	-	
P50	TIQ01	输入	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 0	PFC50 = 1	KRM0 (KRM) = 0
	KR0	输入	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 0	PFC50 = 1	TQ0TIG2, TQ0TIG3 (TQ0IOC1) = 0
	TOQ01	输出	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 1	PFC50 = 0	
	RTP00	输出	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 1	PFC50 = 1	
	TIQ02	输入	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 0	PFC51 = 1	KRM1 (KRM) = 0
P51	KR1	输入	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 0	PFC51 = 1	TQ0TIG4, TQ0TIG5 (TQ0IOC1) = 0
	TOQ02	输出	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 1	PFC51 = 0	
	RTP01	输出	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 1	PFC51 = 1	

注 1. 仅限内置 CAN 控制器的版本

2. 仅限内置 CAN 控制器的版本

3. 仅限内置 I²C 总线版本 (Y 产品)

表 4-15. 端口引脚用作复用功能引脚 (3/7)

Pin 管脚名称	功能复用		Pn寄存器 的Pnx位	PMn寄存器 的PMnx位	PMCn寄存器 的PMCnx位	PFCEn寄存器 的PFCEnx位	PFCn寄存器 的PFCnx位	其它位 (寄存器)
	名称	I/O						
P52	TIQ03	输入	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 0	PFC52 = 1	KRM2 (KRM) = 0
	KR2	输入	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 0	PFC52 = 1	TQ0TIG6, TQ0TIG7 (TQ0IOC1) = 0
	TOO03	输出	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 1	PFC52 = 0	
	RTP02	输出	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 1	PFC52 = 1	
	DDI ^注	输入	P52 = 无需设置	PM52 = 无需设置	PMC52 = 无需设置	PFCE52 = 无需设置	PFC52 = 无需设置	OCDM0 (OCDM) = 1
	SIB2	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 0	PFC53 = 0	
P53	TIQ00	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 0	PFC53 = 1	KRM3 (KRM) = 0
	KR3	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 0	PFC53 = 1	TQ0TIG0, TQ0TIG1 (TQ0IOC1) = 0, TQ0EES0, TQ0EES1 (TQ0IOC2) = 0, TQ0ETS0, TQ0ETS1 (TQ0IOC2) = 0
	TOQ00	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 1	PFC53 = 0	
	RTP03	输出	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 1	PFC53 = 1	
	DDO ^注	输出	P53 = 无需设置	PM53 = 无需设置	PMC53 = 无需设置	PFCE53 = 无需设置	PFC53 = 无需设置	OCDM0 (OCDM) = 1
	SOB2	输出	P54 = 无需设置	PM54 = 无需设置	PMC54 = 1	PFCE54 = 0	PFC54 = 0	
P54	KR4	输入	P54 = 无需设置	PM54 = 无需设置	PMC54 = 1	PFCE54 = 0	PFC54 = 1	
	RTP04	输出	P54 = 无需设置	PM54 = 无需设置	PMC54 = 1	PFCE54 = 1	PFC54 = 1	
	DCK ^注	输入	P54 = 无需设置	PM54 = 无需设置	PMC54 = 无需设置	PFCE54 = 无需设置	PFC54 = 无需设置	OCDM0 (OCDM) = 1
	SCKB2	I/O	P55 = 无需设置	PM55 = 无需设置	PMC55 = 1	PFCE55 = 0	PFC55 = 0	
	KR5	输入	P55 = 无需设置	PM55 = 无需设置	PMC55 = 1	PFCE55 = 0	PFC55 = 1	
	RTP05	输出	P55 = 无需设置	PM55 = 无需设置	PMC55 = 1	PFCE55 = 1	PFC55 = 1	
P55	DMS ^注	输入	P56 = 无需设置	PM56 = 无需设置	PMC56 = 无需设置	PFCE56 = 无需设置	PFC56 = 无需设置	OCDM0 (OCDM) = 1

注 仅限于 flash 存储器版本

表 4-15. 端口引脚用作复用功能引脚 (4/17)

管脚名称	功能复用		Pn 寄存器 的 Pnx 位	PMn 寄存器 的 PMnx 位	PMCn 寄存器 的 PMCnx 位	PFCEn 寄存器 的 PFCEnx 位	PFCn 寄存器 的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P70	AN10	输入	P70 = 无需设置	PM70 = 1	-	-	-	
P71	AN11	输入	P71 = 无需设置	PM71 = 1	-	-	-	
P72	AN12	输入	P72 = 无需设置	PM72 = 1	-	-	-	
P73	AN13	输入	P73 = 无需设置	PM73 = 1	-	-	-	
P74	AN14	输入	P74 = 无需设置	PM74 = 1	-	-	-	
P75	AN15	输入	P75 = 无需设置	PM75 = 1	-	-	-	
P76	AN16	输入	P76 = 无需设置	PM76 = 1	-	-	-	
P77	AN17	输入	P77 = 无需设置	PM77 = 1	-	-	-	
P78	AN18	输入	P78 = 无需设置	PM78 = 1	-	-	-	
P79	AN19	输入	P79 = 无需设置	PM79 = 1	-	-	-	
P710	AN110	输入	P710 = 无需设置	PM710 = 1	-	-	-	
P711	AN111	输入	P711 = 无需设置	PM711 = 1	-	-	-	
P90	A0	输出	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 0	PFC90 = 0	注 1
P91	KR6	输入	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 0	PFC90 = 1	
	TXDA1	输出	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 1	PFC90 = 0	
	SDA02 ^{注 2}	I/O	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 1	PFC90 = 1	PF90 (PF9) = 1
	A1	输出	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 0	PFC91 = 0	注 1
	KR7	输入	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 0	PFC91 = 1	
	RXDA1/KR7 ^{注 3}	输入	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 1	PFC91 = 0	
	SCL02 ^{注 2}	I/O	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 1	PFC91 = 1	PF91 (PF9) = 1

<R> 注 1. 端口 9 的引脚不能用作端口引脚或其他功能复用引脚，即使 A0 到 A15 只有一个用于分离总线模式。在设置寄存器 PFC9 和 PFCE9 为 0000H 之后，

PMC9 寄存器 16 位一起设置为 FFFFH。

2. 仅限内置 I²C 总线版本 (Y 产品)

3. RXDA1 和 KR7 引脚不能同时使用。使用 RXDA1 引脚时，不要使用 KR7 引脚。使用 KR7 引脚时，不要使用 RXDA1 引脚（建议使用 PFC91 的位数设为 1，并将 PFCE91 的位数清 0）。

表 4-15. 端口引脚用作复用功能引脚 (5/7)

管脚名称	功能复用		Pn寄存器 的Pnx位	PMn寄存器 的PMnx位	PMCn寄存器 的PMCnx位	PFCEn寄存器 的PFCEnx位	PFCn寄存器 的PFCnx位	其它位 (寄存器)
	名称	I/O						
P92	A2	输出	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 0	PFC92 = 0	注
	TIP41	输入	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 0	PFC92 = 1	
	TOP41	输出	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 1	PFC92 = 0	
P93	A3	输出	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 0	PFC93 = 0	注
	TIP40	输入	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 0	PFC93 = 1	
	TOP40	输出	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 1	PFC93 = 0	
P94	A4	输出	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 0	PFC94 = 0	注
	TIP31	输入	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 0	PFC94 = 1	
	TOP31	输出	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 1	PFC94 = 0	
P95	A5	输出	P95 = 无需设置	PM95 = 无需设置	PMC95 = 1	PFCE95 = 0	PFC95 = 0	注
	TIP30	输入	P95 = 无需设置	PM95 = 无需设置	PMC95 = 1	PFCE95 = 0	PFC95 = 1	
	TOP30	输出	P95 = 无需设置	PM95 = 无需设置	PMC95 = 1	PFCE95 = 1	PFC95 = 0	
P96	A6	输出	P96 = 无需设置	PM96 = 无需设置	PMC96 = 1	PFCE96 = 0	PFC96 = 0	注
	TIP21	输入	P96 = 无需设置	PM96 = 无需设置	PMC96 = 1	PFCE96 = 1	PFC96 = 0	
	TOP21	输出	P96 = 无需设置	PM96 = 无需设置	PMC96 = 1	PFCE96 = 1	PFC96 = 1	
P97	A7	输出	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 0	PFC97 = 0	注
	SIB1	输入	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 0	PFC97 = 1	
	TIP20	输入	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 1	PFC97 = 0	
P98	TOP20	输出	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 1	PFC97 = 1	
	A8	输出	P98 = 无需设置	PM98 = 无需设置	PMC98 = 1	-	PFC98 = 0	注
	SOB1	输出	P98 = 无需设置	PM98 = 无需设置	PMC98 = 1	-	PFC98 = 1	
P99	A9	输出	P99 = 无需设置	PM99 = 无需设置	PMC99 = 1	-	PFC99 = 0	注
	SCKB1	I/O	P99 = 无需设置	PM99 = 无需设置	PMC99 = 1	-	PFC99 = 1	

注 端口 9 的引脚不能用作端口引脚或其他功能复用引脚，即使 A0 到 A15 只有一个用于分离总线模式。在设置寄存器 PFC9 和 PFCE9 为 0000H 之后，PMC9 寄存器 16 位一起设置为 FFFFH。

表 4-15. 端口引脚用作复用功能引脚 (617)

管脚名称	功能复用		Pn寄存器 的Pnx位	PMn寄存器 的PMnx位	PMCn寄存器 的PMCnx位	PFCEn寄存器 的PFCEnx位	PFCn寄存器 的PFCnx位	其它位 (寄存器)
	名称	I/O						
P910	A10	输出	P910 = 无需设置	PM910 = 无需设置	PMC910 = 1	-	PFC910 = 0	注
	SIB3	输入	P910 = 无需设置	PM910 = 无需设置	PMC910 = 1	-	PFC910 = 1	
	A11	输出	P911 = 无需设置	PM911 = 无需设置	PMC911 = 1	-	PFC911 = 0	注
P911	SOB3	输出	P911 = 无需设置	PM911 = 无需设置	PMC911 = 1	-	PFC911 = 1	
	A12	输出	P912 = 无需设置	PM912 = 无需设置	PMC912 = 1	-	PFC912 = 0	注
	SCKB3	I/O	P912 = 无需设置	PM912 = 无需设置	PMC912 = 1	-	PFC912 = 1	
P912	A13	输出	P913 = 无需设置	PM913 = 无需设置	PMC913 = 1	-	PFC913 = 0	注
	INTP4	输入	P913 = 无需设置	PM913 = 无需设置	PMC913 = 1	-	PFC913 = 1	
	A14	输出	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 0	PFC914 = 0	注
P913	INTP5	输入	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 0	PFC914 = 1	
	TIP51	输入	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 1	PFC914 = 0	
	TOP51	输出	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 1	PFC914 = 1	
P914	A15	输出	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 0	PFC915 = 0	注
	INTP6	输入	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 0	PFC915 = 1	
	TIP50	输入	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 1	PFC915 = 0	
P915	TOP50	输出	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 1	PFC915 = 1	
	WAIT	输入	PCM0 = 无需设置	PMCM0 = 无需设置	PMCCM0 = 1	-	-	
	CLKOUT	输出	PCM1 = 无需设置	PMCM1 = 无需设置	PMCCM1 = 1	-	-	
PCM0	HLDAK	输出	PCM2 = 无需设置	PMCM2 = 无需设置	PMCCM2 = 1	-	-	
PCM1	HLDRQ	输入	PCM3 = 无需设置	PMCM3 = 无需设置	PMCCM3 = 1	-	-	
PCM2	WR0	输出	PCT0 = 无需设置	PMCT0 = 无需设置	PMCCCT0 = 1	-	-	
PCM3	WR1	输出	PCT1 = 无需设置	PMCT1 = 无需设置	PMCCCT1 = 1	-	-	
PCT0	RD	输出	PCT4 = 无需设置	PMCT4 = 无需设置	PMCCCT4 = 1	-	-	
PCT1	ASTB	输出	PCT6 = 无需设置	PMCT6 = 无需设置	PMCCCT6 = 1	-	-	
PCT4								
PCT6								

注 端口 9 的引脚不能用作端口引脚或其他功能复用引脚，即使 A0 到 A15 只有一个用于分离总线模式。在设置寄存器 PFC9 和 PFCE9 为 0000H 之后，PMC9 寄存器 16 位一起设置为 FFFFH。

<R>

表 4-15. 端口引脚用作复用功能引脚（717）

管脚名称	功能复用		Pn寄存器的Pnx位	PMn寄存器的PMnx位	PMCn寄存器的PMCnx位	PFCEn寄存器的PFCEnx位	PFCn寄存器的PFCnx位	其它位 (寄存器)
	名称	I/O						
PDH0	A16	输出	PDH0 = 无需设置	PMDH0 = 无需设置	PMCDH0 = 1	-	-	
PDH1	A17	输出	PDH1 = 无需设置	PMDH1 = 无需设置	PMCDH1 = 1	-	-	
PDH2	A18	输出	PDH2 = 无需设置	PMDH2 = 无需设置	PMCDH2 = 1	-	-	
PDH3	A19	输出	PDH3 = 无需设置	PMDH3 = 无需设置	PMCDH3 = 1	-	-	
PDH4	A20	输出	PDH4 = 无需设置	PMDH4 = 无需设置	PMCDH4 = 1	-	-	
PDH5	A21	输出	PDH5 = 无需设置	PMDH5 = 无需设置	PMCDH5 = 1	-	-	
PDL0	AD0	I/O	PDL0 = 无需设置	PMDL0 = 无需设置	PMCDL0 = 1	-	-	
PDL1	AD1	I/O	PDL1 = 无需设置	PMDL1 = 无需设置	PMCDL1 = 1	-	-	
PDL2	AD2	I/O	PDL2 = 无需设置	PMDL2 = 无需设置	PMCDL2 = 1	-	-	
PDL3	AD3	I/O	PDL3 = 无需设置	PMDL3 = 无需设置	PMCDL3 = 1	-	-	
PDL4	AD4	I/O	PDL4 = 无需设置	PMDL4 = 无需设置	PMCDL4 = 1	-	-	
PDL5	AD5	I/O	PDL5 = 无需设置	PMDL5 = 无需设置	PMCDL5 = 1	-	-	
	FLMD1 ^注	输入	PDL5 = 无需设置	PMDL5 = 无需设置	PMCDL5 = 无需设置	-	-	
PDL6	AD6	I/O	PDL6 = 无需设置	PMDL6 = 无需设置	PMCDL6 = 1	-	-	
PDL7	AD7	I/O	PDL7 = 无需设置	PMDL7 = 无需设置	PMCDL7 = 1	-	-	
PDL8	AD8	I/O	PDL8 = 无需设置	PMDL8 = 无需设置	PMCDL8 = 1	-	-	
PDL9	AD9	I/O	PDL9 = 无需设置	PMDL9 = 无需设置	PMCDL9 = 1	-	-	
PDL10	AD10	I/O	PDL10 = 无需设置	PMDL10 = 无需设置	PMCDL10 = 1	-	-	
PDL11	AD11	I/O	PDL11 = 无需设置	PMDL11 = 无需设置	PMCDL11 = 1	-	-	
PDL12	AD12	I/O	PDL12 = 无需设置	PMDL12 = 无需设置	PMCDL12 = 1	-	-	
PDL13	AD13	I/O	PDL13 = 无需设置	PMDL13 = 无需设置	PMCDL13 = 1	-	-	
PDL14	AD14	I/O	PDL14 = 无需设置	PMDL14 = 无需设置	PMCDL14 = 1	-	-	
PDL15	AD15	I/O	PDL15 = 无需设置	PMDL15 = 无需设置	PMCDL15 = 1	-	-	

注 因为这个引脚被设置成 flash 存储器编程模式，因此不需要用端口控制寄存器进行操作。详细内容，参见第 30 章 flash 存储器

4.6 注意事项

4.6.1 设置端口引脚的注意事项

<R> (1) 在 V850ES/SG2 和 V850ES/SG2-H 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 PFn 寄存器^注：

<2>设置 PFCn 和 PFCEn 寄存器：

<3>将 PMCn 寄存器相应的位数设为 1：
- N 沟道开漏设置

复用功能选择

切换到复用功能模式

需要注意，如果先设置了 PMCn 寄存器，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按照 PFn，PFCn，和 PFCEn 寄存器中的设置。
具体示例如下。

注 针对 N 沟道开漏输出引脚

注意事项 无论是端口模式/复用功能模式，Pn 寄存器按以下方式来读写。

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

[例] SCL01 引脚设置样例
SCL01 引脚与 P41/SOB0 引脚复用。使用 PMC4，PFC4，和 PF4 寄存器来选择对应的引脚功能。

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

将 P41 引脚切换到 SCL01 引脚时，可能会发生的故障设置顺序如下所示。

设置顺序	设定内容	引脚状态	引脚电平
<1>	初始值 (PMC41 位 = 0, PFC41 位 = 0, PF41 位 = 0)	端口模式 (输入)	Hi-Z
<2>	PMC41 bit ← 1	SOB0 输出	低电平 (基于 CSIB0 设定为高电平)
<3>	PFC41 bit ← 1	SCL01 I/O	高电平 (CMOS 输出)
<4>	PF41 bit ← 1	SCL01 I/O	Hi-Z (N 沟道开漏输出)

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

(b) 复用功能模式 (输入) 的注意事项

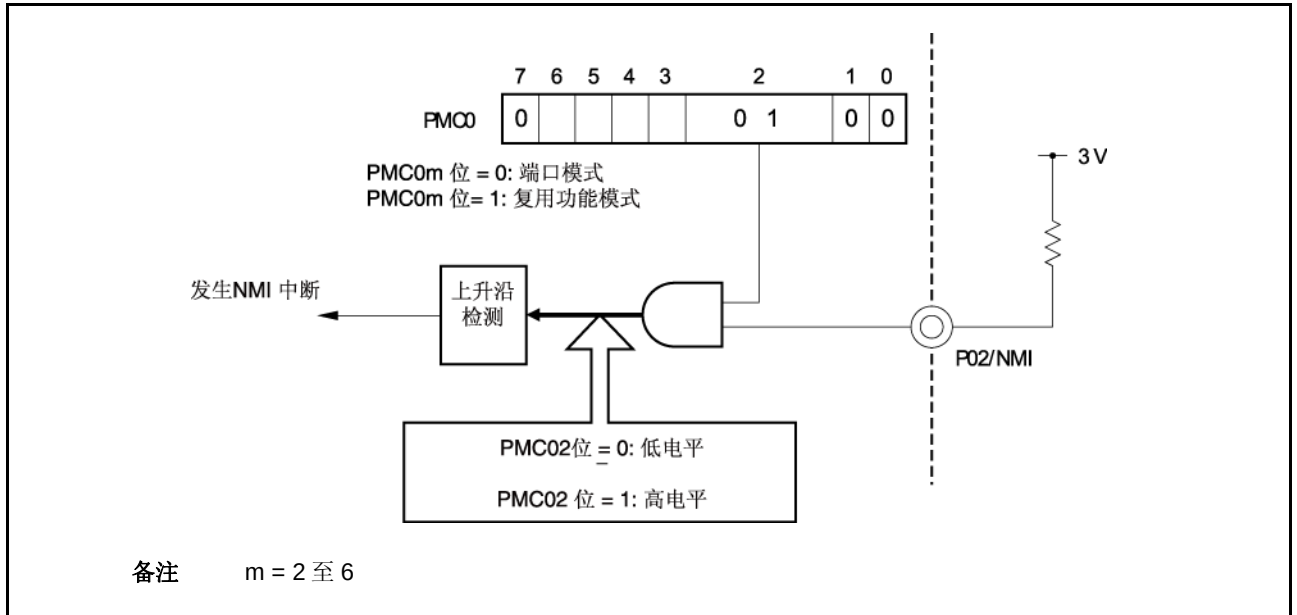
当 PMCn.PMCnm 位为 0 时，由于 PMCn 寄存器的设置值和引脚电平与操作 (AND) 后输出，因此输入到复用功能块的信号为低电平。因此，根据端口设置和复用功能操作使能时序，可能会产生不可预期的操作。因此，要按照以下顺序在端口模式和复用功能模式间切换。

- 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。

具体示例如例 1 和例 2 所示

[例 1] 从通用端口 (P02) 切换到外部中断引脚 (NMI)
如图 4-33 所示，当 P02/NMI 引脚上拉，并在 NMI 引脚边沿检测设置中指定上升沿时，即使在 P02 引脚切换到 NMI 引脚期间，有高电平 (PMC02 位 = 0 → 1) 连续输入到 NMI 引脚中，但仍旧被认为是上升沿，像从低电平变为高电平一样，产生 NMI 中断。
要避免以上情况发生，需在从 P02 引脚切换到 NMI 引脚后，再设置 NMI 引脚的有效沿。

图 4-33. 从 P02 切换到 NMI 的例子（不正确）



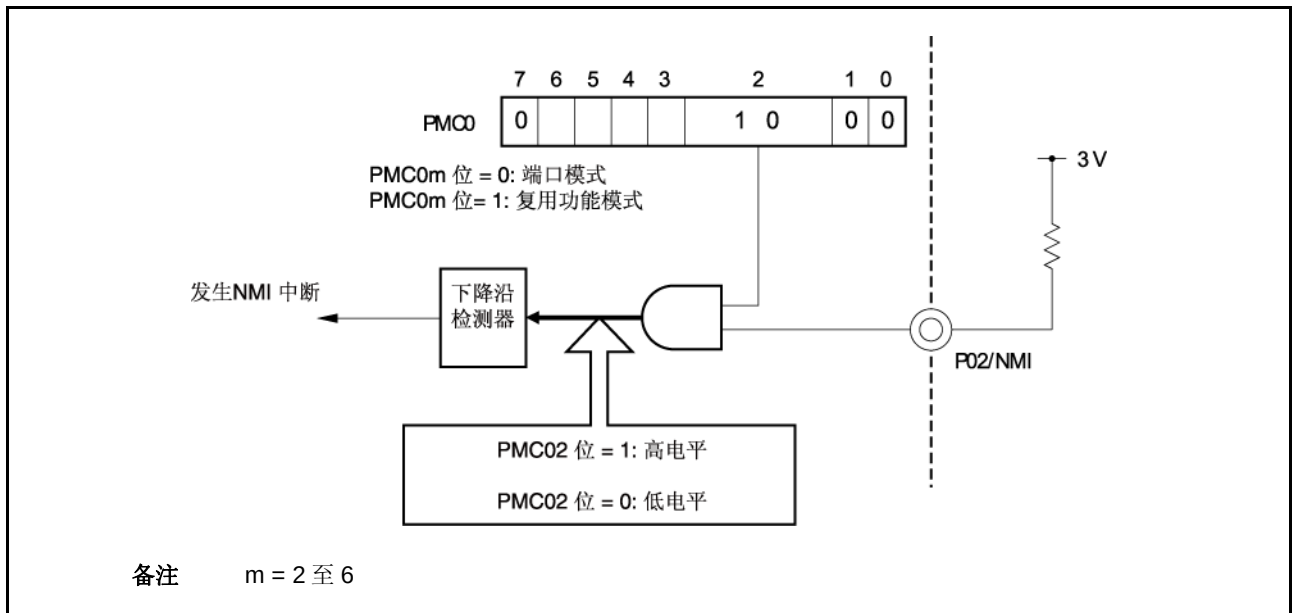
[例 2]

从外部引脚（NMI）切换到通用端口（P02）

如图 4-34 所示，当 P02/NMI 引脚上拉，且在 NMI 引脚边沿检测设置中指定下降沿时，即使在 NMI 引脚切换到 P02 引脚期间，有高电平（PMC02 位 = 1 → 0）连续输入到 NMI 引脚中，但仍旧认为是下降沿，像从高电平变为低电平一样，产生 NMI 中断。

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

图 4-34. 从 NMI 切换到 P02 的例子（不正确）



- (2) 在端口模式中，PFn.PFnm 位只在输出模式（PMn.PMnm 位 = 0）中有效。在输入模式（PMnm 位 = 1）中，PFnm 位的值不会影响到缓冲器。

4.6.2 端口 n 寄存器 (Pn) 位操作指令的注意事项

当一个支持输入和输出功能的端口上执行位操作指令时，除了目标位以外，输入端口的输出锁存的值也可能被意外写入。

因此，当端口从输入模式切换到输出模式时，推荐重写输出锁存。

<例> 当 P90 引脚为输出端口，P91 至 P97 引脚为输入端口（所有的引脚状态均为高电平），且端口锁存的值为 00H 时，如果通过位操作指令将 P90 引脚的输出从低电平变为高电平，则端口锁存的值为 FFH。

说明：从 PMnm 位 = 1 的端口 Pn 寄存器的写入对象是输出锁存，读取对象的是引脚状态。

V850ES/SG2 和 V850ES/SG2-H 中位操作指令按照以下顺序执行。

<R>

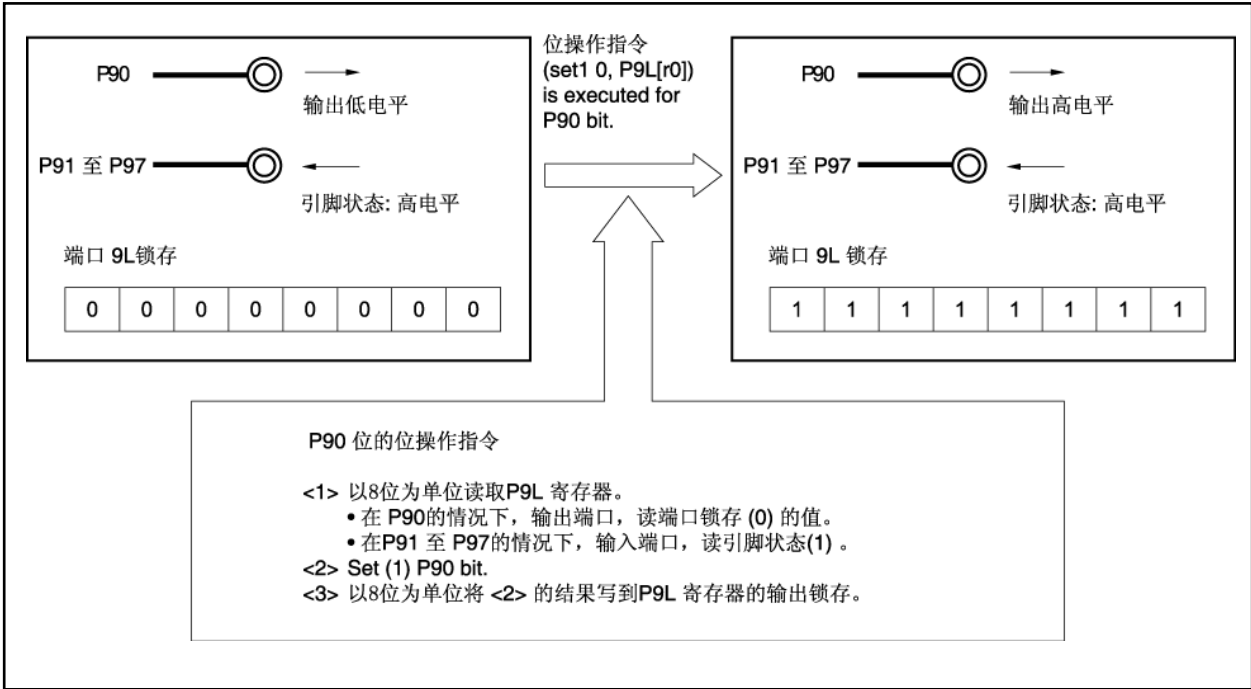
- <1> 以 8 位为单位读取 Pn 寄存器。
- <2> 操作目标位。
- <3> 以 8 位为单位写入 Pn 寄存器。

在步骤 <1>中，当作为输入端口的 P91 至 P97 引脚状态被读取时，作为输出端口的 P90 引脚的输出锁存（0）的值被读取。如果此时 P91 至 P97 引脚的状态为高电平，则读取的值为 FEH。

通过在步骤<2>中操作，此值变为 FFH。

通过在步骤<3>中操作，FFH 被写到输出锁存中。

图 4-35. 位操作指令（P90 引脚）



4.6.3 片上调试引脚的注意事项

$\overline{\text{DRST}}$, DCK, DMS, DDI 和 DDO 引脚是片上调试引脚（这些引脚仅存在于 flash 存储器版本中）。

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

如果不使用片上调试, 则需要采取以下措施。

- 将 OCDM 寄存器（特殊寄存器）的 OCDM0 位清（0）

此时, 从通过 $\overline{\text{RESET}}$ 引脚复位释放直到采取以上措施期间, 一直将 P05/INTP2/ $\overline{\text{DRST}}$ 引脚固定在低电平上。

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

注意事项 当由看门狗定时器溢出引发复位, 低电压检测器（LVI）产生复位信号（LVIREN），或者由时钟监测器（CLM）（只有 V850ES/SG2 可以低电压检测器（LVI）复位）产生复位信号（CLMRES）时, 引脚 P05/INTP2/ $\overline{\text{DRST}}$ 不会被初始化作为片上调试引脚（ $\overline{\text{DRST}}$ ）。OCDM 寄存器保持当前值。

<R>

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

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

4.6.5 当电源打开时P10, P11, 和P53 引脚的注意事项

<R>

上电时, 下列引脚可能会瞬时输出不定电平。

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

注 只有存储器版本才有引脚 DDO。

4.6.6 磁滞特性

在端口模式, 以下端口引脚没有磁滞特性。

P02 到 P06
 P31 至 P35, P37 至 P39
 P40 至 P42
 P50 至 P55
 P90 至 P97, P99, P910, P912 至 P915

4.6.7 分离总线模式注意事项

端口 9 的引脚不能用作端口引脚或其他功能复用引脚, 即使 A0 到 A15 只有一个用于分离总线模式。在设置寄存器 PFC9 和 PFCE9 为 0000H 之后, PMC9 寄存器 16 位被一起设置为 FFFFH。即使引脚 A0 到 A15 有一个不是用于分离总线模式, 端口 9 也可以用作端口引脚或其他功能复用引脚。

第 5 章 总线控制功能

<R> V850ES/SG2 和 V850ES/SG2-H 系列产品提供了一个外部总线接口功能，通过该功能，可以连接外部存储器诸如 ROM 和 RAM 以及外扩的 I/O 器件。

5.1 特征

- 输出可以选择复用总线或分离总线，复用总线最小传输时间为 3 个总线周期，分离总线最小传输时间为 2 个总线周期。
- 可选择 8 位/16 位数据总线
- 等待功能
 - 最大支持 7 个状态的可编程等待功能
 - 使用 $\overline{\text{WAIT}}$ 引脚的外部等待功能
- 空闲状态功能
- 总线保持功能
- <R> ○ 最大可连接 4MB 物理存储器
- 当 $\text{BV}_{\text{DD}} \leq \text{EV}_{\text{DD}} = \text{V}_{\text{DD}}$ 时，可以使用不同于工作电压的电压来控制总线。然而，在分离总线模式或者 A20 和 A21 引脚被使用时，设置 $\text{BV}_{\text{DD}} = \text{EV}_{\text{DD}} = \text{V}_{\text{DD}}$ 。

5.2 总线控制引脚

用于连接外部设备的引脚如下表所示。

表 5-1. 总线控制引脚（复用总线）

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

表 5-2. 外部控制引脚（分离总线）

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

5.2.1 当内置ROM，内置RAM或片上周边I/O被访问时的引脚状态

当内置 ROM，内置 RAM 或片上周边 I/O 被访问时，各引脚的状态如下表所示。

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

分离总线模式		复用总线模式	
地址总线（A21 到 A0）	不确定	地址总线（A21 到 A16）	不确定
数据总线（AD15 到 AD0）	Hi-Z	地址/数据总线（AD15 to AD0）	不确定
控制信号（ $\overline{\text{RD}}$, $\overline{\text{WR0}}$, $\overline{\text{WR1}}$ ）	高电平	控制信号（ $\overline{\text{RD}}$, $\overline{\text{WR0}}$, $\overline{\text{WR1}}$, ASTB）	高电平

注意事项 对内置 ROM 区域进行写访问时，地址、数据以及控制信号的激活和访问外部存储器区域的方式相同。

5.2.2 各操作模式中的引脚状态

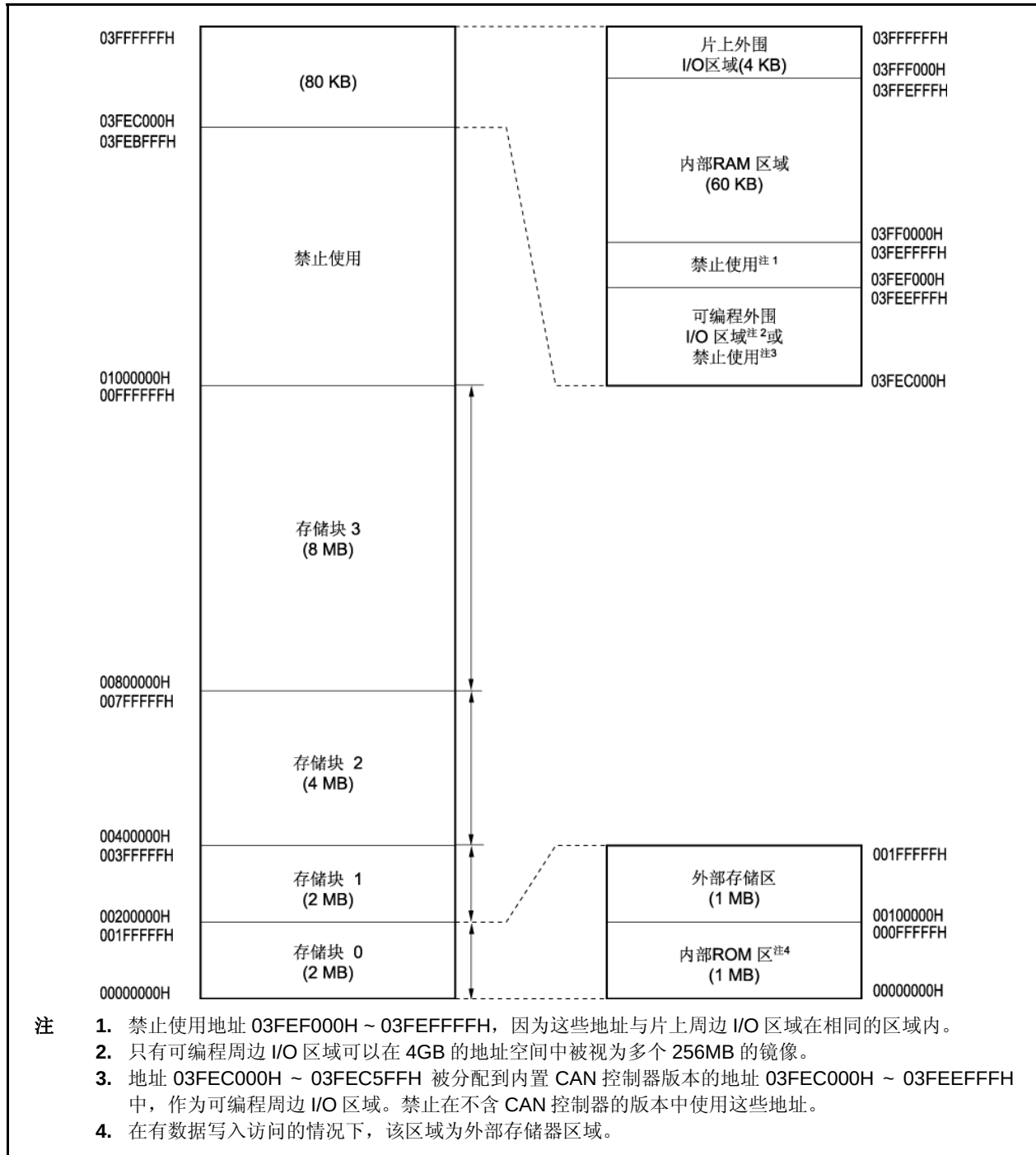
<R>

V850ES/SG2 和 V850ES/SG2-H 产品在各操作模式下引脚的状态，参见 2.2 引脚状态。

5.3 存储器分块功能

16MB 的外部存储器空间由低到高被分为 2MB, 2MB, 4MB 和 8MB 共 4 个存储器块。这些存储器模块中每个模块的可编程等待功能和总线周期操作模式可以以块为单位单独进行控制。

图 5-1. 数据存储器映射：物理地址



5.4 外部总线接口模式控制功能

<R> V850ES/SG2 和 V850ES/SG2-H 有如下两种外围总线接口模式。

- 复用总线模式
- 分离总线模式

可以使用 EXIMC 寄存器在这两种模式间进行选择。

(1) 外部总线接口模式控制寄存器 (EXIMC)

EXIMC 寄存器可以以 8 位或 1 位为单位来读取或写入。

复位输入将把该寄存器设置为 00H。

复位之后: 00H R/W 地址: FFFFFFFBEH							
EXIMC	7	6	5	4	3	2	1
	0	0	0	0	0	0	0
SMSEL							
模式选择							
0 复用总线 模式							
1 分离总线模式							

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

5.5 总线访问

5.5.1 访问所需的时钟数

下表中列出了访问每种资源时所需的基本时钟数。

(1) 在 V850ES/SG2 中

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

- 注
1. 如果与数据访问发生冲突，则增加 1。
 2. 当选择分离总线模式时，所需时钟数为 $2 + n$ (n: 等待状态的数量)。

备注 单位: 时钟/访问

<R> (2) 在 V850ES/SG2-H 中

区域 (总线宽度) 总线周期类型	内置 ROM (32 位)	内置 RAM (32 位)	外部存储器 (16 位)
取指 (正常访问)	1	1 ^{注 1}	3 + n ^{注 2}
取指 (分支)	3	2 ^{注 1}	3 + n ^{注 2}
操作数数据访问	4	1	3 + n ^{注 2}

- 注
1. 如果与数据访问发生冲突，则增加 1。
 2. 当选择分离总线模式时，所需时钟数为 $2 + n$ (n: 等待状态的数量)。

备注 单位: 时钟/访问

5.5.2 总线宽度设置功能

每个通过内存块 n 选择的外部存储器区域可以通过 BSC 寄存器来设置。但总线宽度只可以设置为 8 位和 16 位。

V850ES/SG2 和 V850ES/SG2-H 的外部存储区域可以选择存储块 0 到 3。

<R>

(1) 总线宽度配置寄存器 (BSC)

BSC 寄存器可以以 16 位为单位来读取或写入。

复位输入将把该寄存器设置为 5555H。

注意事项 请在复位后对 BSC 寄存器进行设置，然后不要更改设置值。此外，直到 BSC 寄存器初始化设置完成后才可以访问外部存储器区域。

复位之后: 5555H R/W 地址: FFFFF066H

	15	14	13	12	11	10	9	8
BSC	0	1	0	1	0	1	0	1
	7	6	5	4	3	2	1	0
	0	BS30	0	BS20	0	BS10	0	BS00
		存储块 3		存储块 2		存储块 1		存储块 0
BSn0	存储块 n 空间数据总线宽度(n = 0 到3)							
0	8 位							
1	16位							

注意事项 请务必将第 14, 12, 10 和 8 位设置为“1”，并将第 15, 13, 11, 9, 7, 5, 3 和 1 位清“0”。

5.5.3 按总线宽度访问

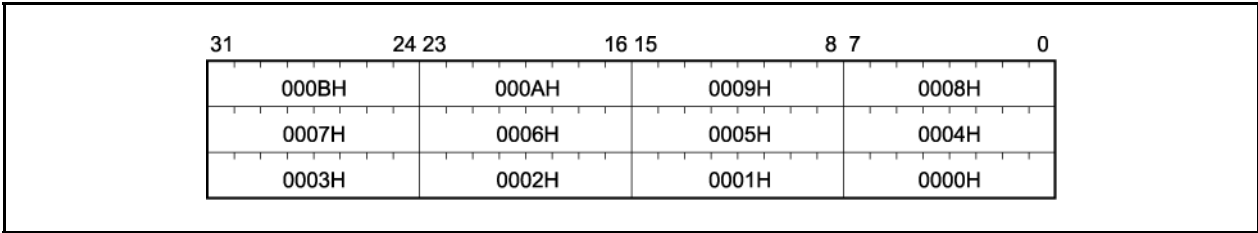
<R> V850ES/SG2 和 V850ES/SG2-H 系列器件以 8 位、16 位或 32 位为单位对片上周边 I/O 和外部存储器进行访问。总线宽度如下所示。

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

以下描述了上述每种操作的访问操作。所有数据的访问都是由低端进行的。

<R> V850ES/SG2 和 V850ES/SG2-H 只支持小端格式。

图 5-2. 以字为单位的小端地址



(1) 数据空间

<R> V850ES/SG2 和 V850ES/SG2-H 有地址匹配错位功能。通过该功能，不论什么格式的数据（字数据或字节数据）都可以被放置在任何位置的地址上。但是，如果字数据或字节数据字数据或半字数据没有和边界对齐，那么访问它们时至少需要产生两次总线周期，导致总线效率降低。

(a) 半字长度的数据访问

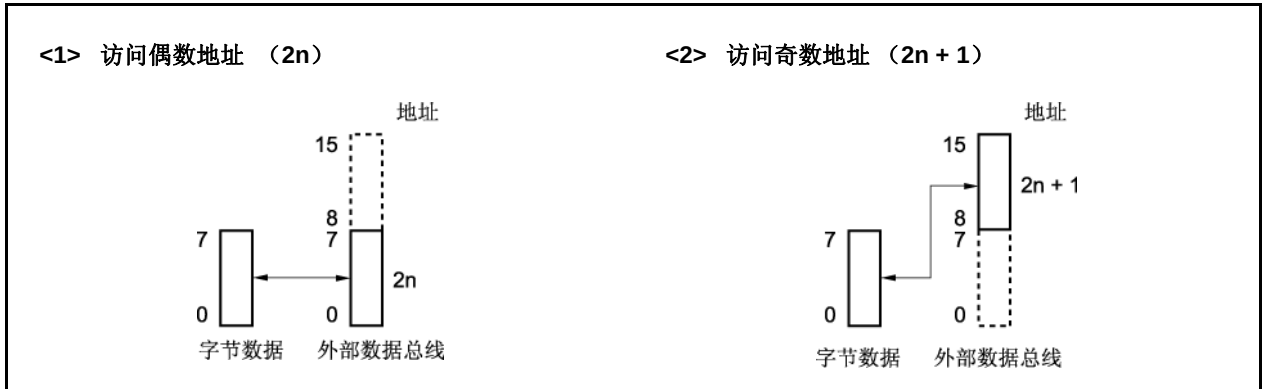
如果地址的最低有效位为 1，那么对其访问时将产生两次字节长度的总线周期。

(b) 字长度的数据访问

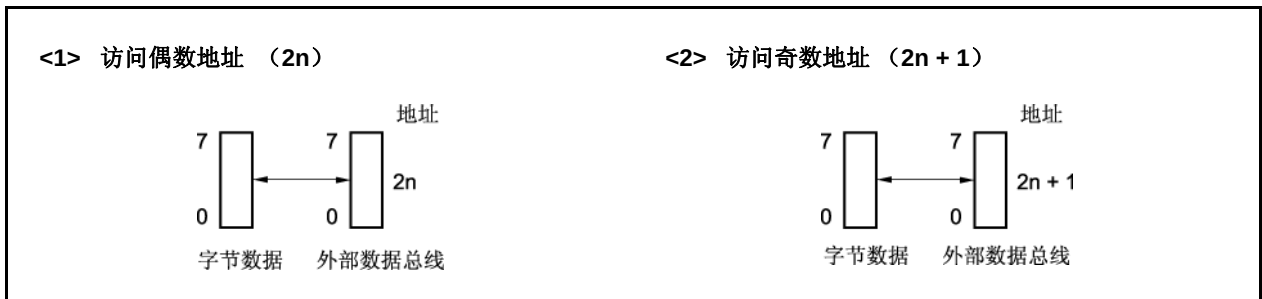
- (i) 如果地址的最低有效位为 1，那么对其访问时将按顺序产生字节长度的总线周期、半字长度的总线周期和字节长度的总线周期。
- (ii) 如果该地址最低 2 位是 10，那么对其访问时将产生两次半字长度的总线周期。

(2) 字节访问 (8 位)

(a) 16 位数据总线宽度

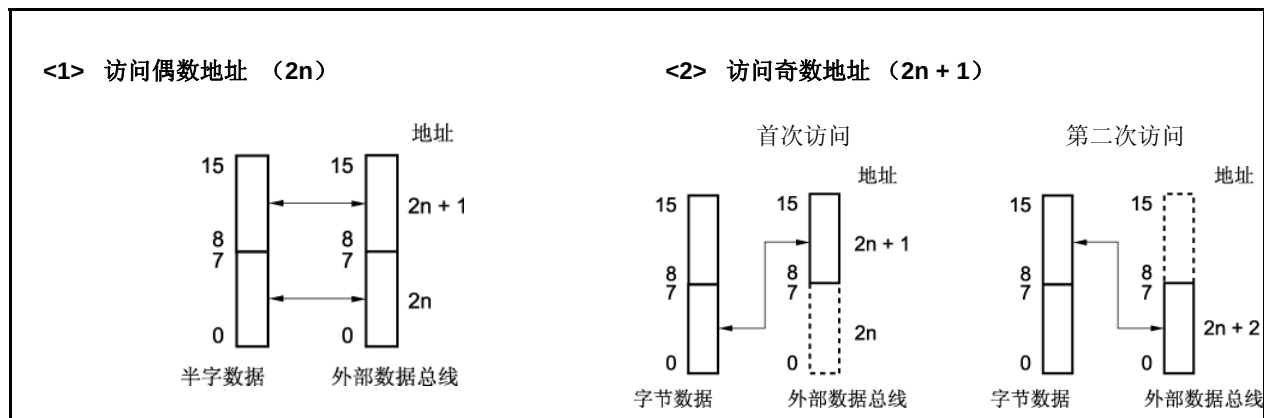


(b) 8 位数据总线宽度

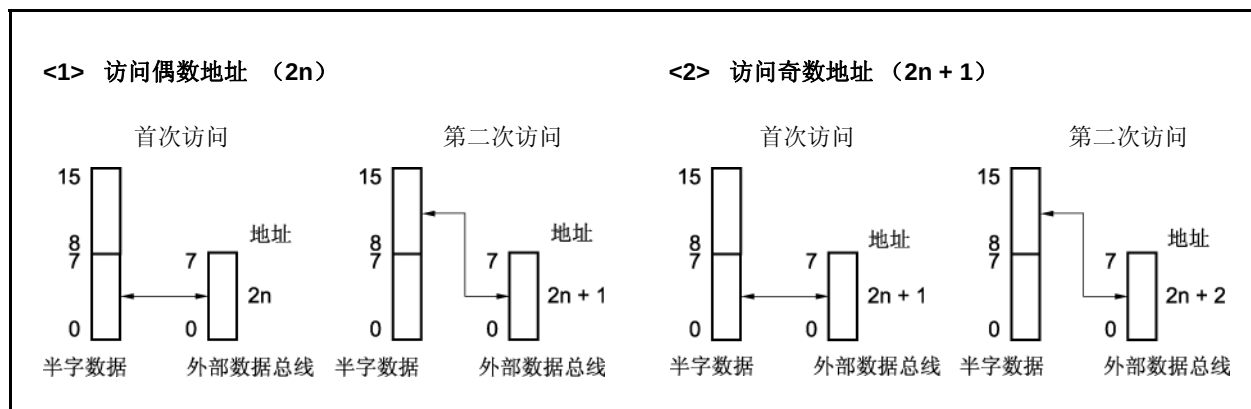


(3) 半字访问 (16 位)

(a) 16 位数据总线宽度



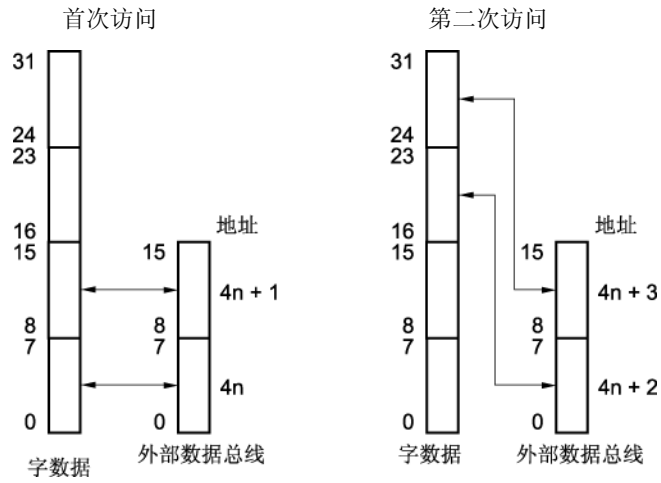
(b) 8 位数据总线宽度



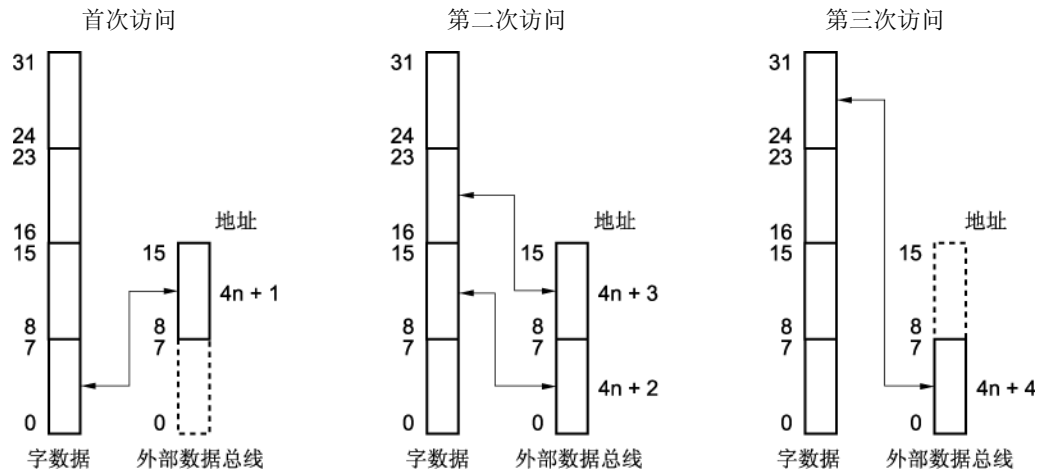
(4) 字访问 (32 bits)

(a) 16 位数据总线宽度 (1/2)

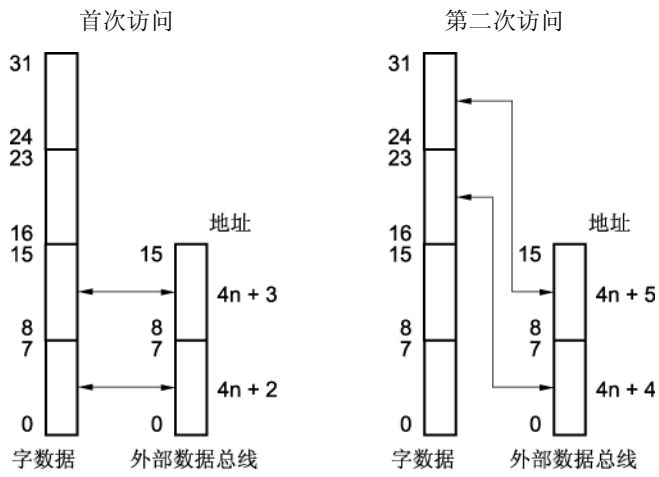
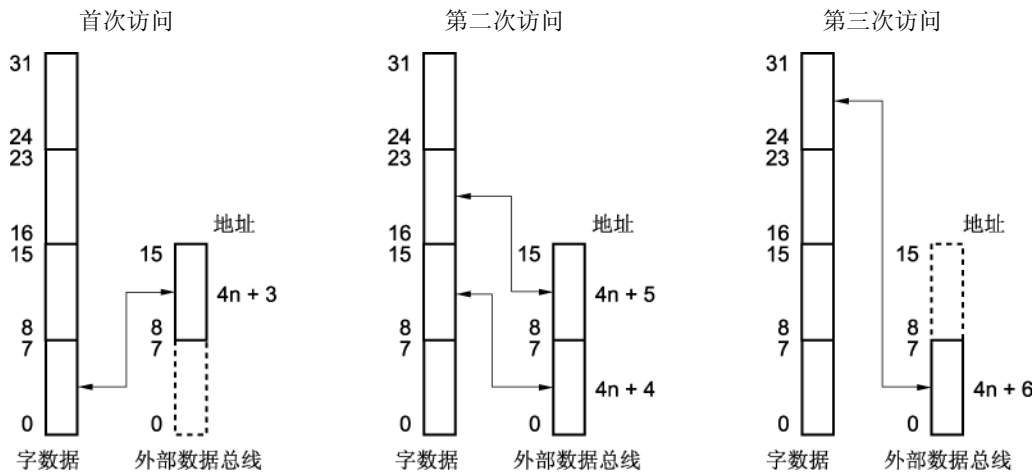
<1> 访问地址 ($4n$)



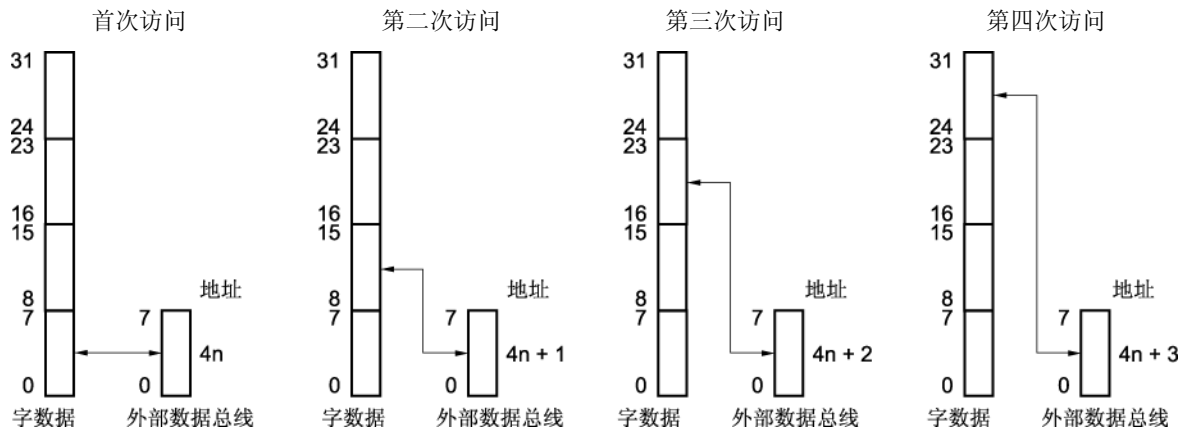
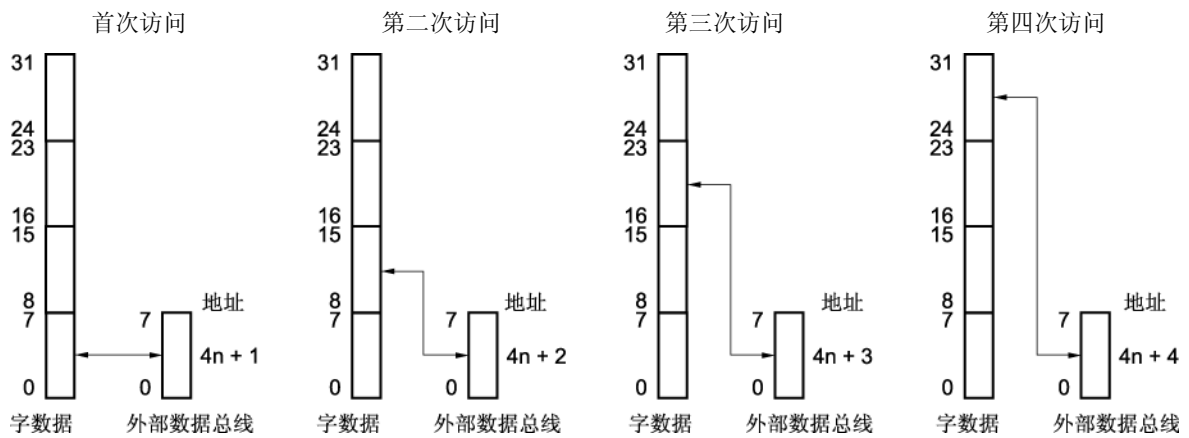
<2> 访问地址 ($4n + 1$)



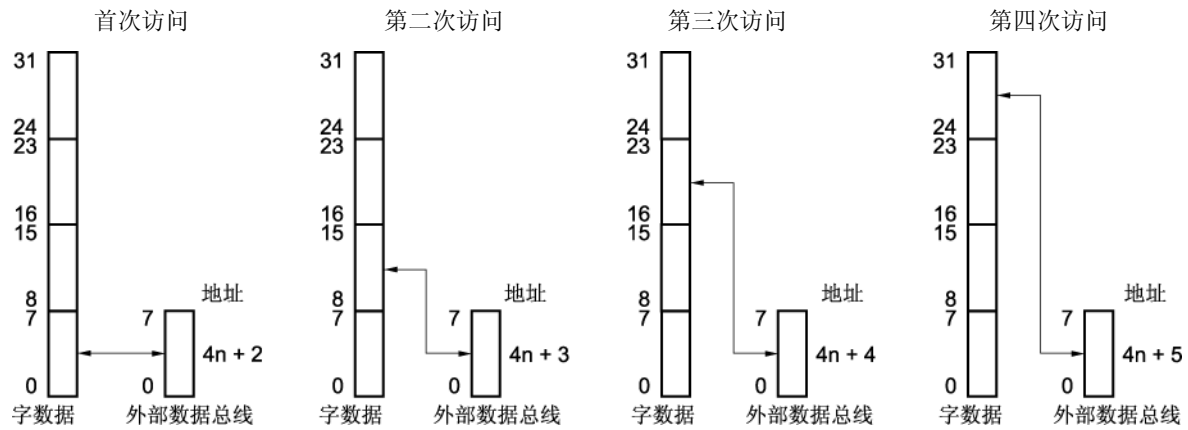
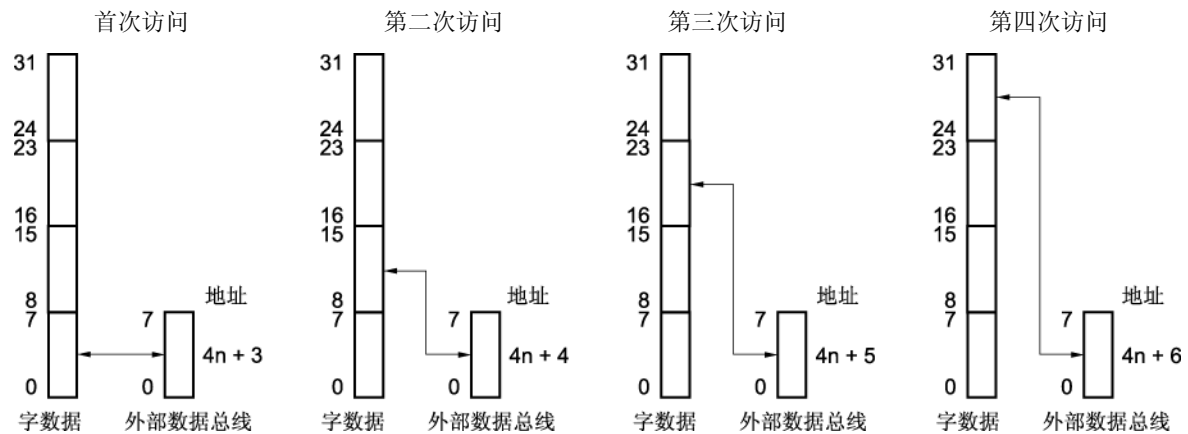
(a) 16 位数据总线宽度 (2/2)

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

(b) 8 位数据总线宽度 (1/2)

<1> 访问地址 ($4n$)<2> 访问地址 ($4n + 1$)

(b) 8 位数据总线宽度 (2/2)

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

5.6.2 外部等待功能

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

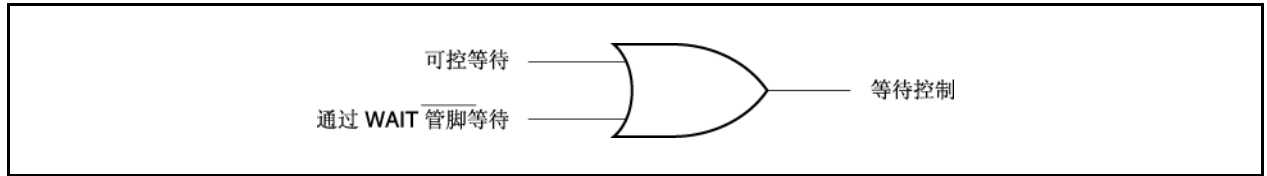
PCM0 引脚设置为复用功能时，外部等待功能有效。

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

$\overline{\text{WAIT}}$ 信号可以异步输入到 CLKOUT 中，在复用总线模式下，WAIT 信号在总线周期的 T2 和 TW 状态的时钟下降沿被采样。在分离总线模式下，WAIT 信号会在紧随总线周期的 T1 和 TW 状态之后的时钟上升沿被采样。如果没有满足采样时间中的建立/保持时间，那么在下一状态前插入一个等待状态，或者根本无需插入。

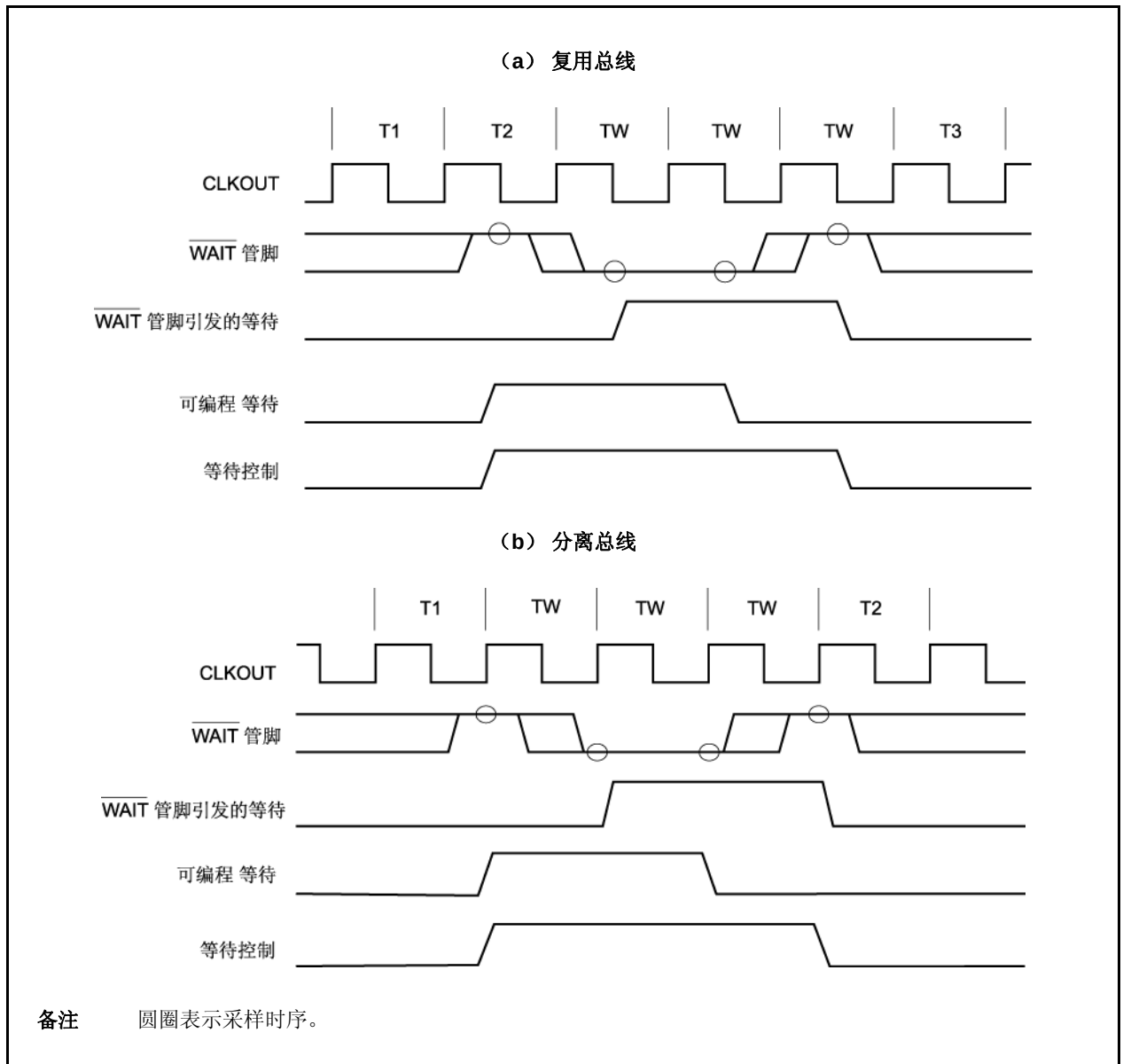
5.6.3 可编程等待和外部等待之间的关系

插入的等待周期是或操作（OR）的结果，将可编程等待中的设置指定的等待周期和 $\overline{\text{WAIT}}$ 引脚控制的等待周期进行或操作。



例如，如果可编程等待的时间和 $\overline{\text{WAIT}}$ 引脚信号如下图所示，那么将在总线周期中插入 3 个等待状态。

图 5-3. 插入等待举例



5.6.4 可编程地址等待功能

在每个总线周期中的地址建立（address-setup）或地址保持（address-hold）插入的等待周期可以通过 AWC 寄存器来设置。在每个存储块空间（存储块 0 到 3）都插入地址等待。

如果插入了一个地址建立等待，这意味 T1 状态的时钟高电平被延长了 1 个时钟周期。如果插入了一个地址保持等待，这意味着 T1 状态的时钟低电平被延长了 1 个时钟周期。

（1）地址等待控制寄存器（AWC）

AWC 寄存器可以以 16 位为单位进行读取或写入。

复位操作将把该寄存器设置为 FFFFH。

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

2. 请在复位后对 AWC 寄存器进行设置，并且在设置完成后不要更改设置值。此外，直到完成 DWC0 寄存器的初始化设置完成后，才可以访问外部存储器区域。

3. 当 V850ES/SG2-H 在 $f_{xx} > 20 \text{ MHz}$ 的情况下运行时，请确保插入地址保持等待和地址设立等待。

复位之后: FFFFH R/W 地址: FFFFF488H

	15	14	13	12	11	10	9	8
AWC	1	1	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
	AHW3		ASW3		AHW2		ASW2	
	AHW1		ASW1		AHW0		ASW0	
	存储块 3		存储块 2		存储块 1		存储块 0	

AHWn	制定插入地址等待保持周期 (n = 0 到 3)	
	$f_{xx} \leq 20 \text{ MHz}$	$f_{xx} > 20 \text{ MHz}$
0	不插入	禁止写入
1	插入	插入

ASWn	制定插入地址等待保持周期 (n = 0 到 3)	
	$f_{xx} \leq 20 \text{ MHz}$	$f_{xx} > 20 \text{ MHz}$
0	不插入	禁止写入
1	插入	插入

注意事项 请务必将第 15 位到第 8 位设置为 1。

5.7 空闲状态插入功能

为了顺利地与低速存储器进行接口访问，可以在总线周期的 T3 状态后插入一个空闲状态（TI），这是在复用地址/数据总线模式中针对通过存储块所选择的每个空间来执行的。分离总线模式下，可在 T2 状态后插入一个空闲状态（TI）。通过插入一个空闲状态，在读取访问期间（无法在写入访问时插入空闲状态）可以确保存储器在数据输出时添加对应的浮动延迟时间（float delay time）。

可以通过 BCC 寄存器来设置是否插入空闲状态。

系统复位后会立即向所有区域插入一个空闲状态。

（1）总线周期控制寄存器（BCC）

BCC 寄存器可以以 16 位为单位进行读取或写入。

复位输入将把该寄存器设置为 AAAAH。

注意事项 1. 内置 ROM，内置 RAM 和片上周边 I/O 区域不受空闲状态插入的影响。

2. 请在复位后对 BCC 寄存器进行设置，并且在设置完成后不要更改设置值。此外，直到完成 BCC 寄存器的初始设置完成后，才可以访问外部存储器区域。

复位之后: AAAAH		R/W	地址: FFFFF48AH					
	15	14	13	12	11	10	9	8
BCC	1	0	1	0	1	0	1	0
	7	6	5	4	3	2	1	0
BC31	0	BC21	0	BC11	0	BC01	0	
存储块 3		存储块 2		存储块 1		存储块 0		
BCn1	制定插入空闲状态(n = 0 到3)							
0	不插入							
1	插入							

注意事项 请务必将第 15, 13, 11 和 9 位设置为“1”，并将第 14, 12, 10, 8, 6, 4, 2 和 0 位清“0”。

5.8 总线保持功能

5.8.1 功能概述

如果 PCM2 和 PCM3 引脚被设置为复用功能，可以使 $\overline{\text{HLDRQ}}$ 和 $\overline{\text{HLDK}}$ 功能生效。

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

在总线保持期间，内置 ROM 和内置 RAM 上的程序会继续运行，直到片上周边 I/O 寄存器或外部存储器被访问。

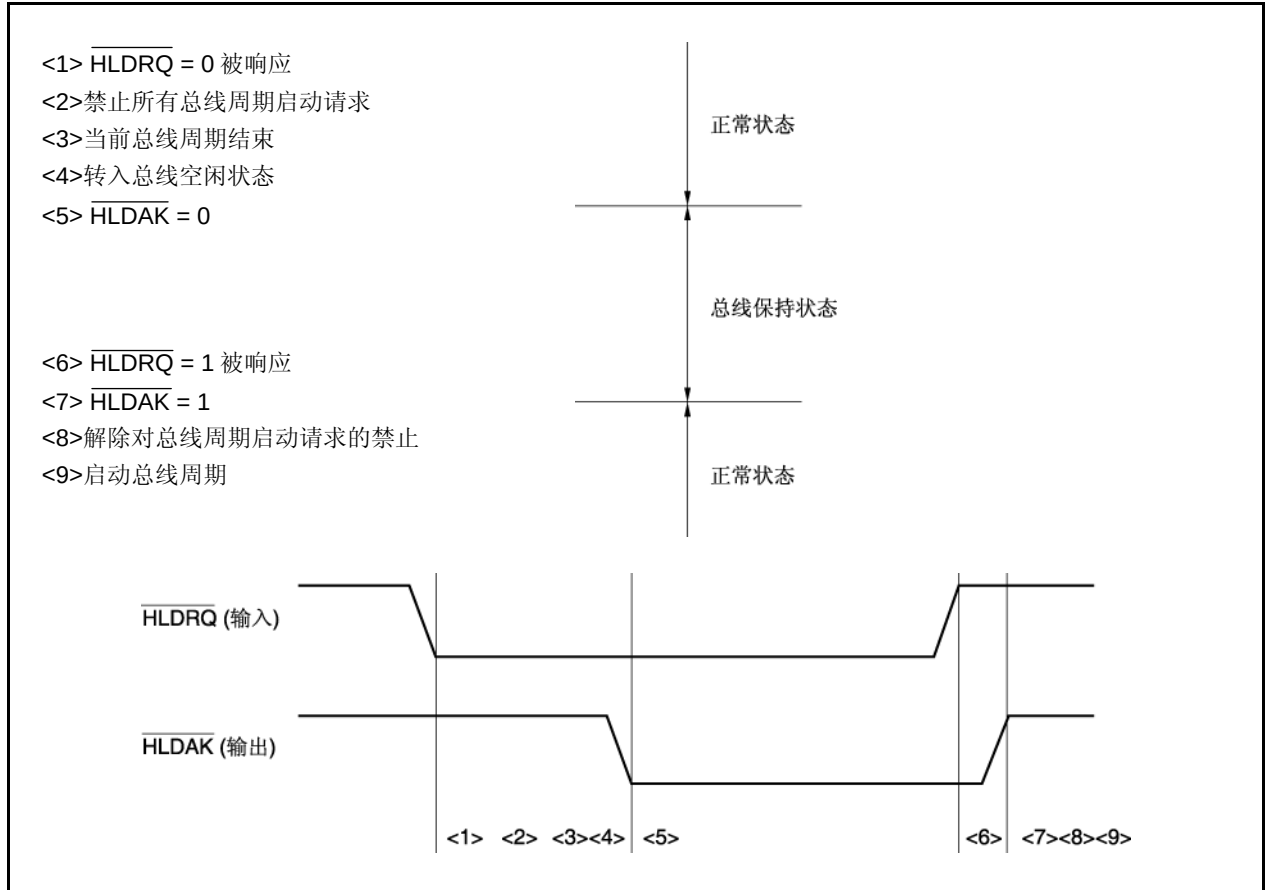
$\overline{\text{HLDK}}$ 引脚的低电平会指示总线保持状态。总线保持功能使内含 2 个或更多总线主设备的多处理器系统的配置成为可能。

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

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

5.8.2 总线保持流程

总线保持状态转换过程如下所示。



5.8.3 节能模式下的操作

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

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

5.9 总线优先级

外部总线周期中执行的操作有总线保持、操作数数据访问、取指（分支）、以及取指（连续）。

总线保持有最高优先级，接下去依次是 DMA 转移，操作数数据访问，取指（分支）以及取指（连续）。

在读取-修改-写入访问过程中，读取访问和写入访问之间可能插入另一条取指操作。

如果某条指令执行了 2 次或更多次访问操作，那么由于总线宽度的限制在访问过程中将不会插入取指或总线保持操作。

表 5-4. 总线优先级

优先级	外部总线周期	总线主设备
高 ↑ ↓ 低	总线保持	外部设备
	DMA 转移	DMAC
	操作数数据访问	CPU
	取指（分支）	CPU
	取指（连续）	CPU

5.10 总线时序

图 5-4. 复用总线读取时序（总线宽度：16 位，16 位访问）

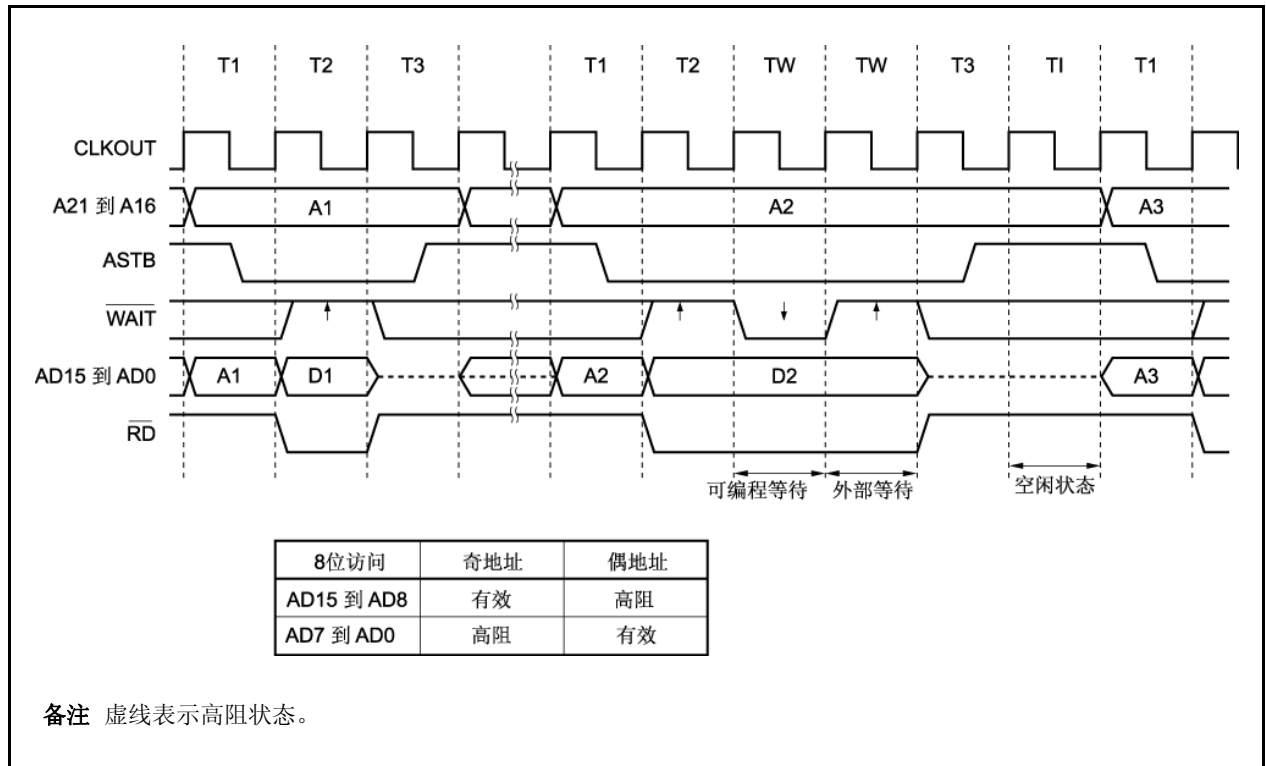


图 5-5. 复用总线读取时序（总线宽度：8 位）

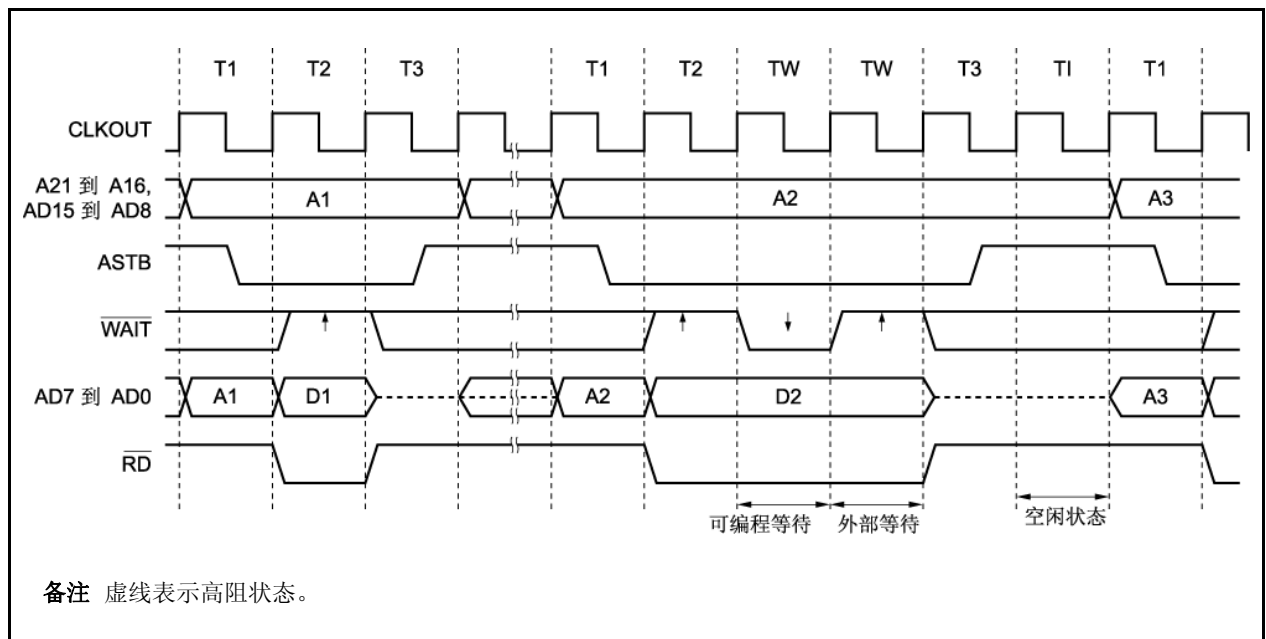


图 5-6. 复用总线写入时序（总线宽度：16 位，16 位访问）

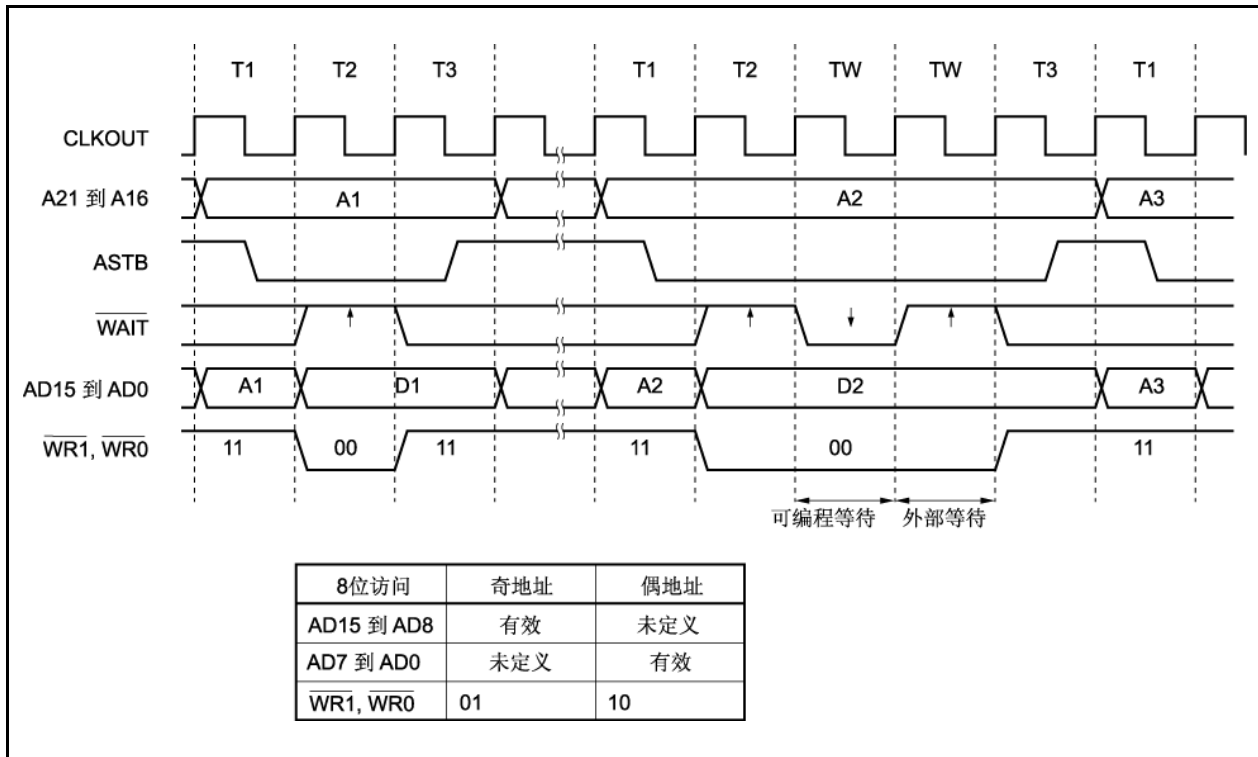


图 5-7. 复用总线写入时序（总线宽度：8 位）

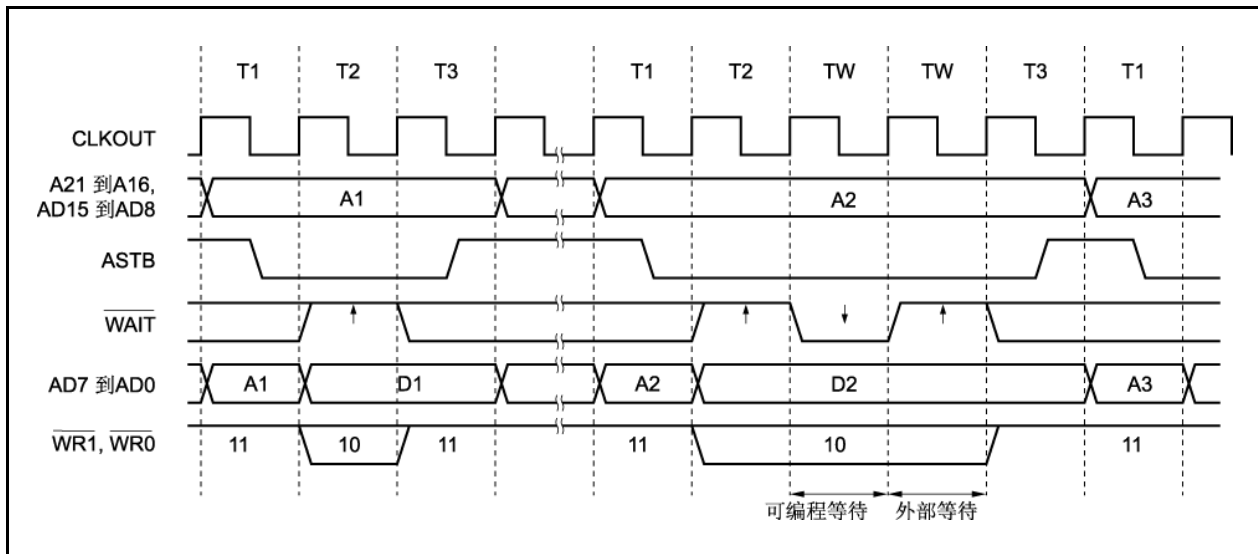


图 5-8. 复用总线保持时序（总线宽度：16 位，16 位访问）

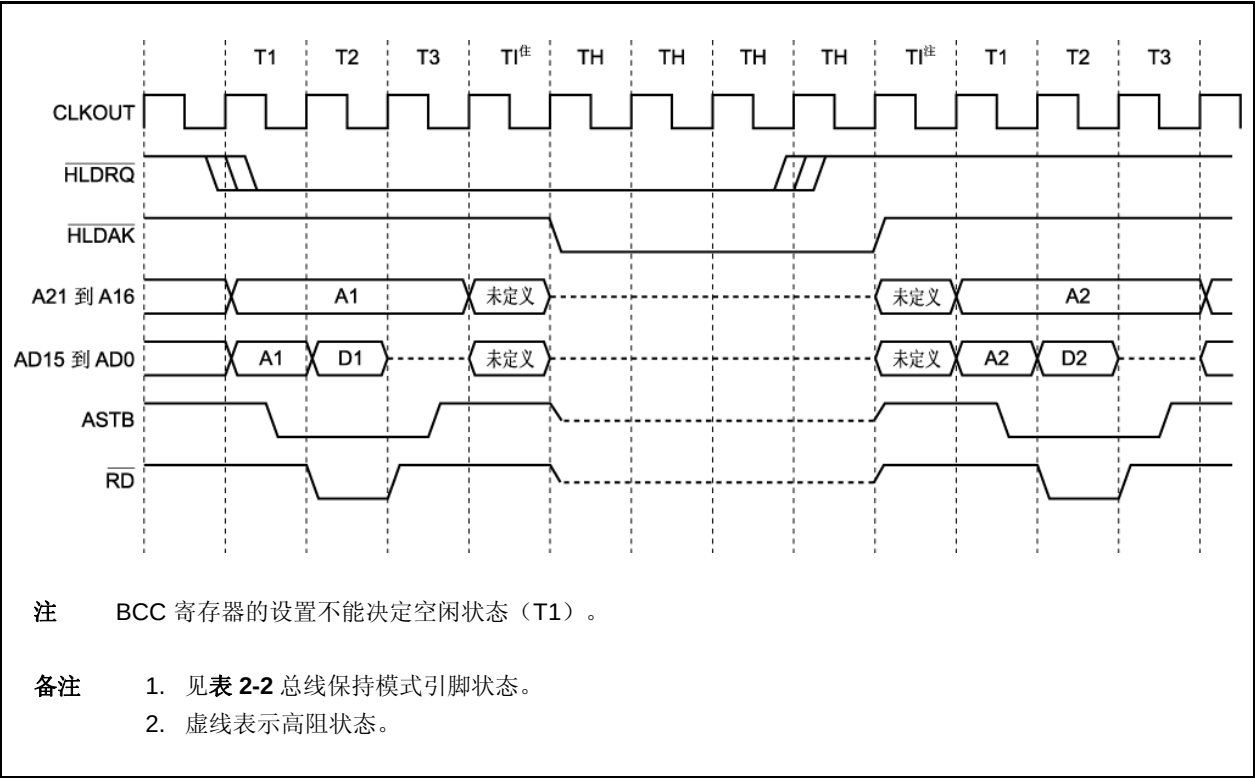
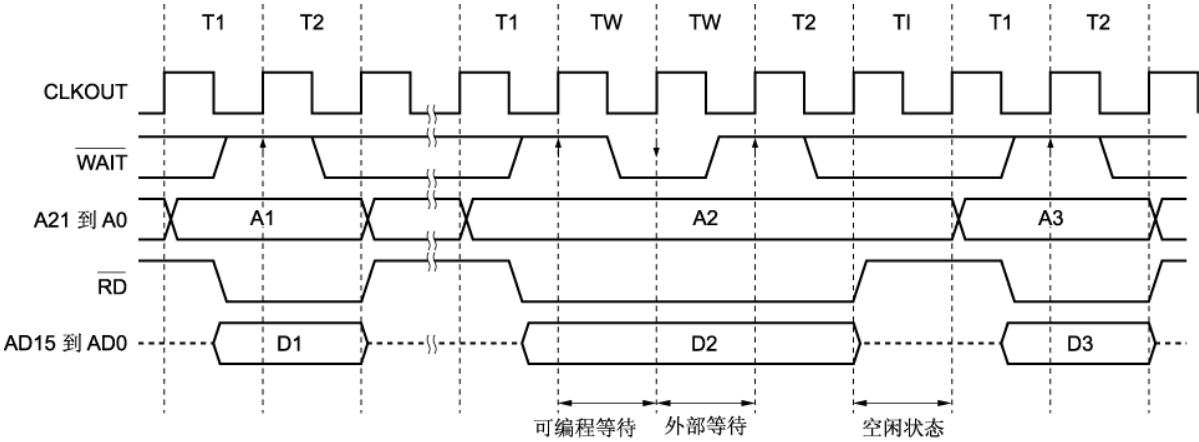


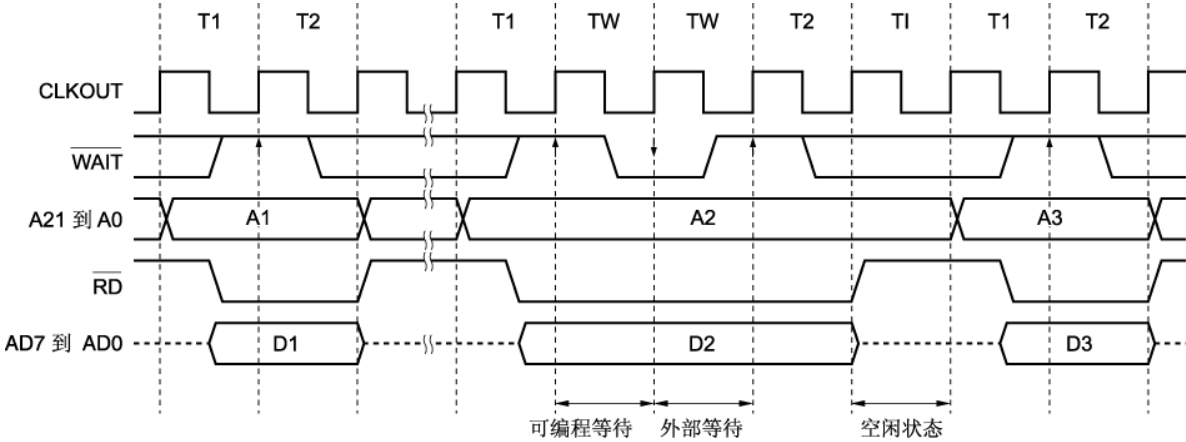
图 5-9. 分离总线读取时序(总线宽度：16 位，16 位访问)



8位访问	奇地址	偶地址
AD15 到 AD8	有效	高阻
AD7 到 AD0	高阻	有效

备注 虚线表示高阻状态。

图 5-10. 分离总线读取时序（总线宽度：8 位）



备注 虚线表示高阻状态。

图 5-11. 分离总线写入时序（总线宽度：16 位，16 位访问）

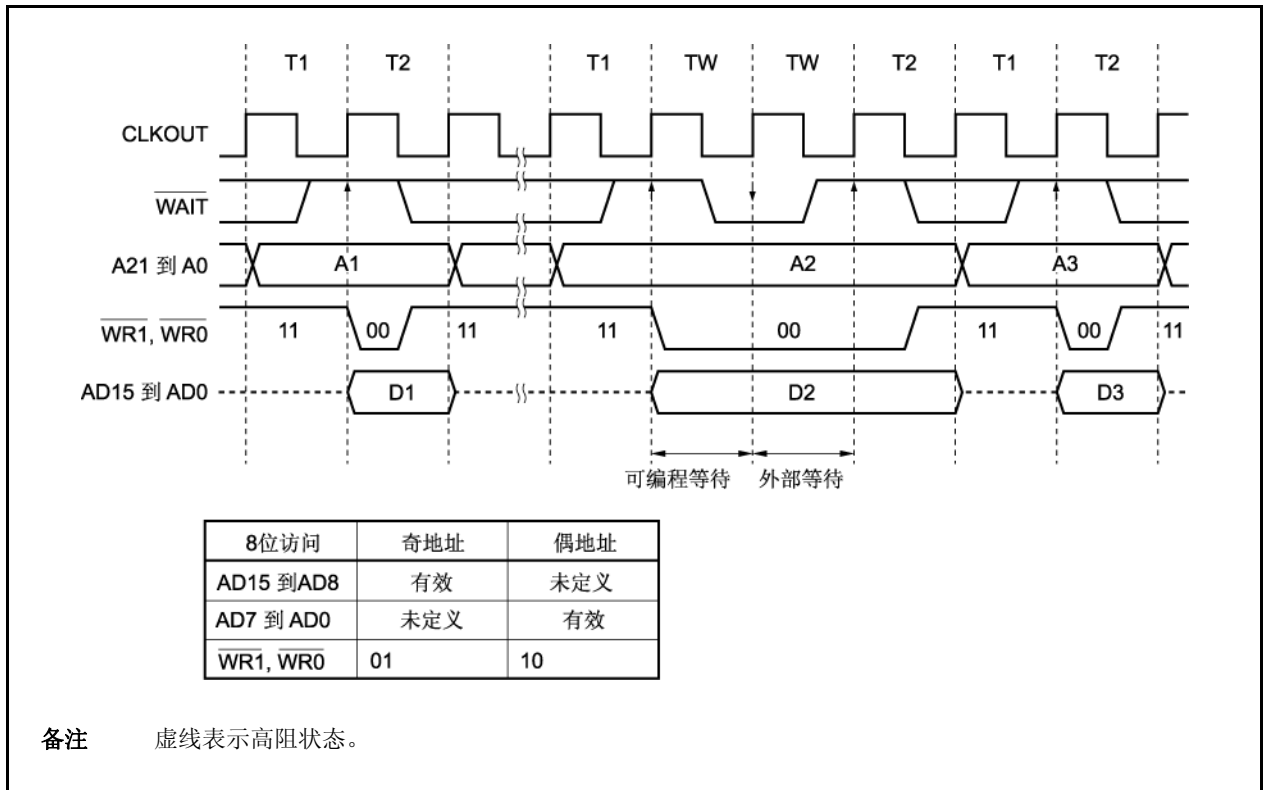


图 5-12. 分离总线写入时序（总线宽度：8 位）

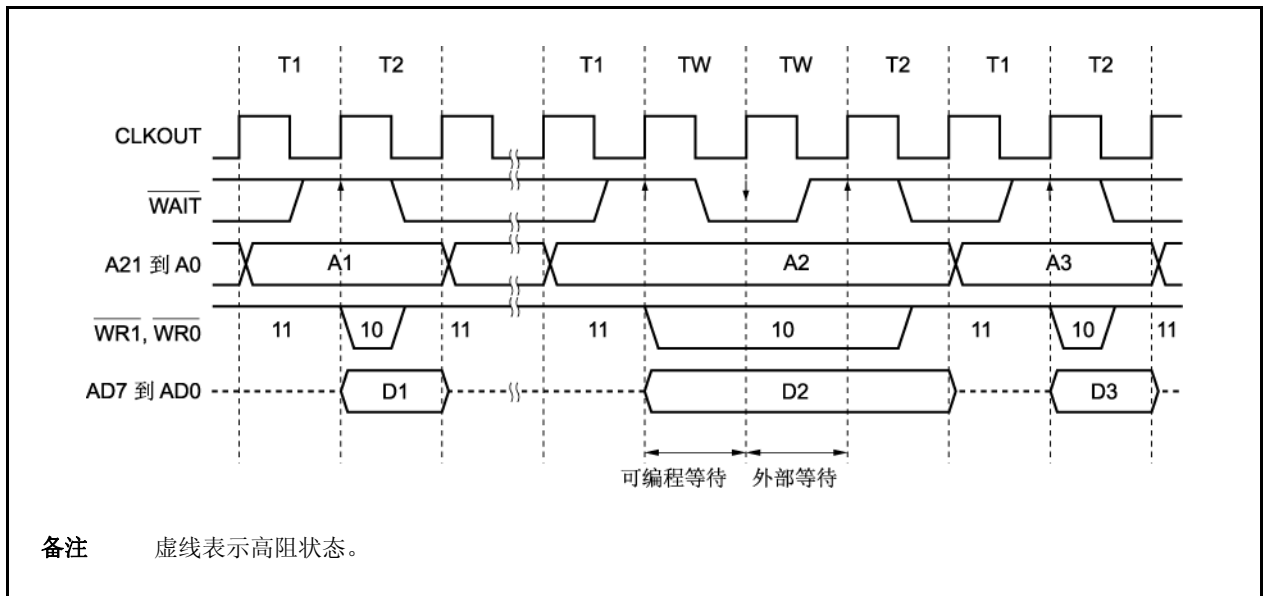
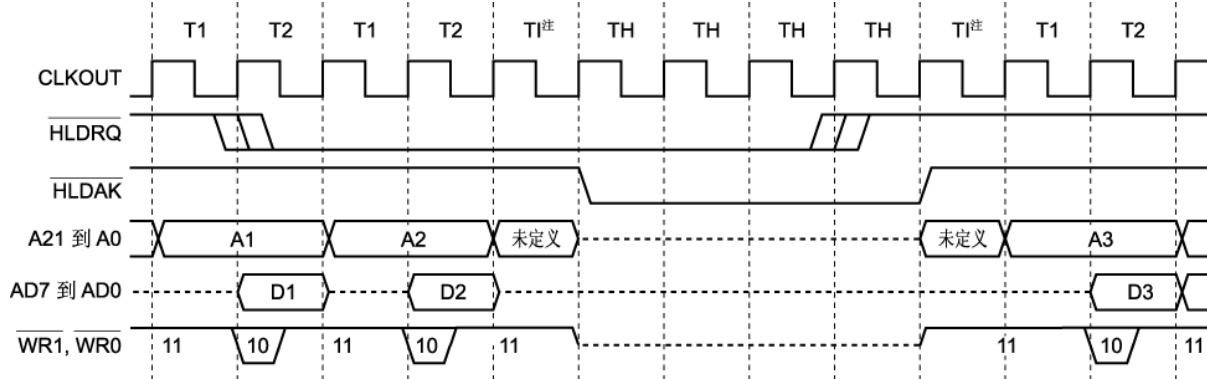


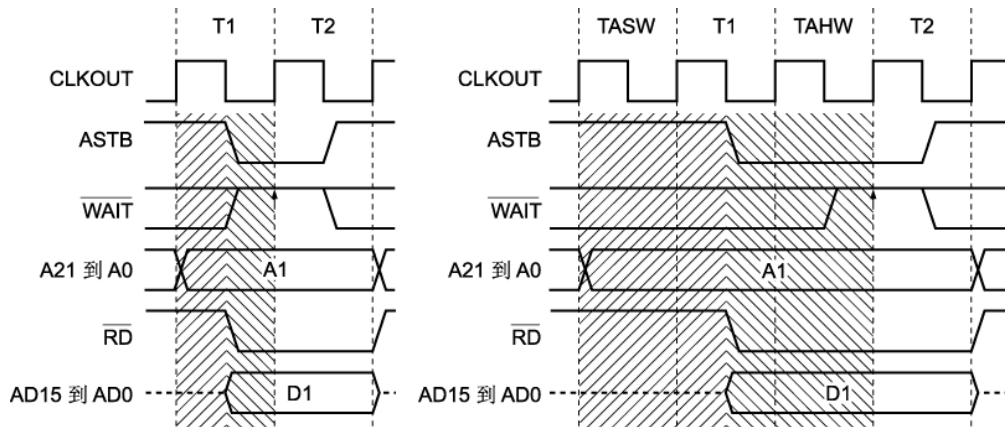
图 5-13. 分离总线保持时序（总线宽度：8 位，写入）



注 BCC 寄存器的设置不能决定空闲状态（T1）。

备注 虚线表示高阻状态。

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



- 备注
1. TASW (地址设立等待): 扩展了 T1 状态的高电平宽度的映像。
 2. TAHW (地址保持等待): 扩展了 T1 状态的高电平宽度的映像。
 3. 虚线表示高阻状态。

6.1 概述

V850ES/SG2 产品提供以下时钟产生功能。

<R>

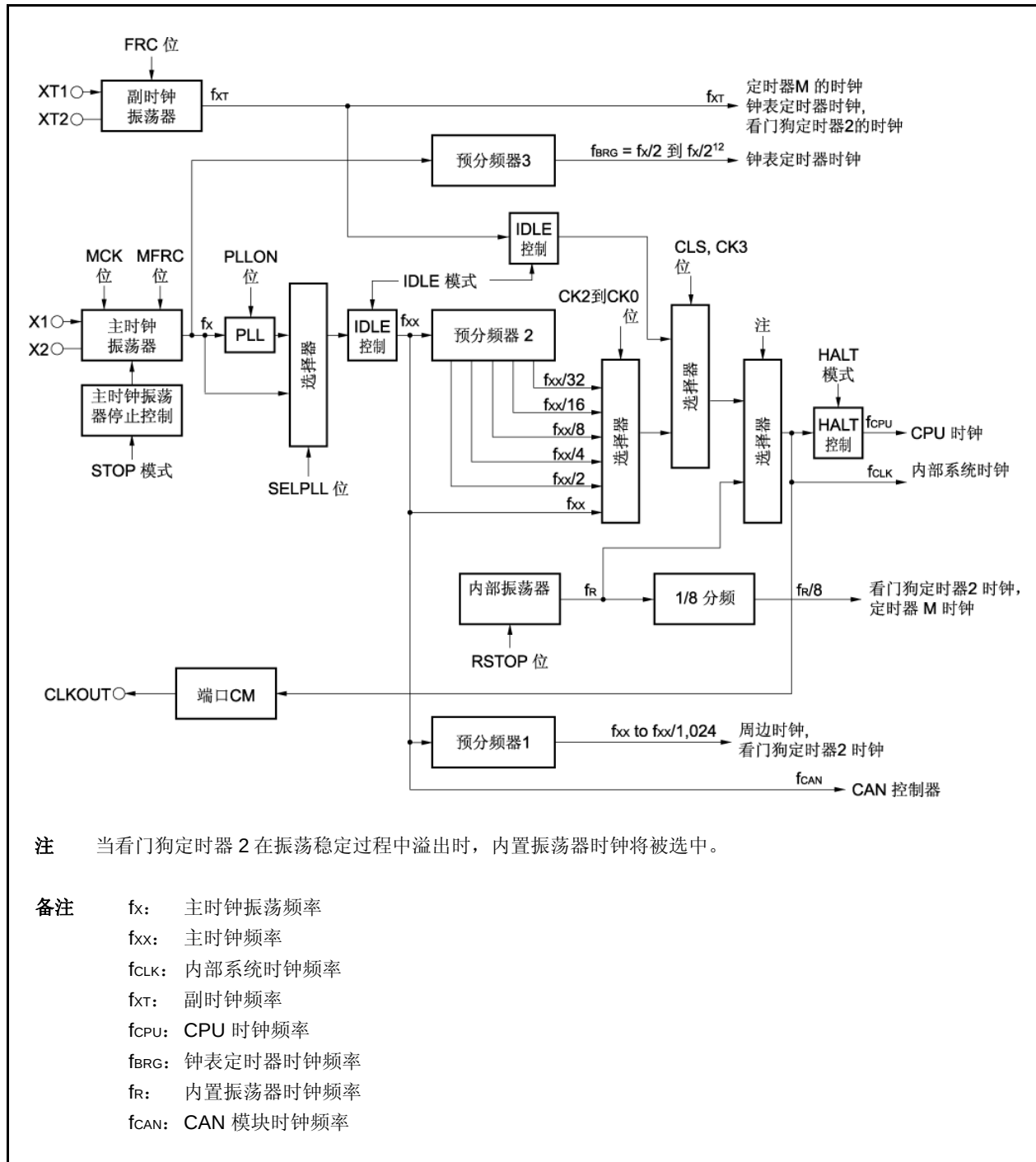
<R>

- 主时钟振荡器
 - 时钟直通模式
 - V850ES/SG2: $f_x = 2.5$ 到 10 MHz ($f_{xx} = 2.5$ 到 10 MHz)
 - V850ES/SG2-H: $f_x = 2.5$ 到 8 MHz ($f_{xx} = 2.5$ 到 8 MHz)
 - PLL 模式
 - V850ES/SG2: $f_x = 2.5$ 到 5 MHz ($f_{xx} = 10$ 到 20 MHz)
 - V850ES/SG2-H: $f_x = 2.5$ 到 5 MHz ($\times 4$: $f_{xx} = 10$ 到 20 MHz)
 $f_x = 2.5$ 到 4 MHz ($\times 8$: $f_{xx} = 20$ 到 32 MHz)
- 副时钟振荡器
 - $f_{XT} = 32.768\text{ kHz}$
- PLL (锁相环) 倍频 ($\times 4/\times 8$) 功能
 - 可选择时钟直通模式/ PLL 模式
- 内置振荡器
 - $f_R = 200\text{ kHz}$ (典型.)
- 内部系统时钟产生
 - 7 级 (f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/8$, $f_{xx}/16$, $f_{xx}/32$, f_{XT})
- 周边时钟产生功能
- 时钟输出功能

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

6.2 结构图

图 6-1. 时钟发生器



(1) 主时钟振荡器

主振荡器产生以下振荡频率 (f_x)。

- 时钟直通模式
 - V850ES/SG2: $f_x = 2.5$ 到 10 MHz
 - V850ES/SG2-H: $f_x = 2.5$ 到 8 MHz
- PLL 模式
 - V850ES/SG2: $f_x = 2.5$ 到 5 MHz
 - V850ES/SG2-H: $f_x = 2.5$ 到 5 MHz ($\times 4$)
 - $f_x = 2.5$ 到 4 MHz ($\times 8$)

<R>

<R>

(2) 副时钟振荡器

子振荡器产生频率为 32.768 kHz 的振荡信号 (f_{XT})。

(3) 主时钟振荡器停止控制电路

该电路生成一个控制信号，可停止主时钟振荡器振荡。

主时钟振荡器的振荡在 STOP 模式下停止，或当 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 200 kHz (TYP.) 的振荡信号 (f_R)。

(5) 预分频器 1

该预分频器生成的时钟信号 (f_{XX} 到 $f_{XX}/1,024$) 将提供给以下片上周边器件: TMP0 到 TMP5, TMQ0, TMM0, CSIB0 到 CSIB4, UARTA0 到 UARTA2, I²C00 到 I²C02^{※1}, ADC, WDT2, CAN0^{※2}, 以及 IEBus^{※3}

<R>

- 注
1. 仅限内置 I²C 总线的版本 (Y 产品)
 2. 仅限内置 CAN 控制器的版本
 3. 仅限内置 IEBus 的版本

(6) 预分频器 2

该电路对主时钟 (f_{XX}) 进行分频。

预分频器 2 生成的时钟信号 (f_{XX} 到 $f_{XX}/32$) 将提供给选择器，用于产生 CPU 时钟 (f_{CPU}) 和内部系统时钟 (f_{CLK})。

f_{CLK} 的时钟信号提供给 INTC, ROM 修正, ROM 和 RAM 模块，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，分至指定的频率 (32.768 kHz) 并将该时钟频率提供给钟表定时器模块。

详见第 10 章 钟表定时器功能。

(8) PLL

该电路对主时钟振荡器产生的时钟信号 (f_x) 进行 4 或 8 倍频。

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

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

6.3 寄存器**(1) 处理器时钟控制寄存器 (PCC)**

PCC 寄存器是一个特殊寄存器。只有按照特定流程才能将数据写入该寄存器（见 **3.4.8 特殊寄存器**）。

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 03H。

复位之后: 03H R/W 地址: FFFF828H

7<6>5<4><3>210

PCC

FRC	MCK	MFRC	CLS ^注	CK3	CK2	CK1	CK0
-----	-----	------	------------------	-----	-----	-----	-----

FRC	副时钟片上反馈电阻的使用						
0	使用						
1	不使用						

MCK	主时钟振荡器控制						
0	振荡器使能						
1	振荡器停止						
- 当系统以主时钟作为CPU时钟工作时，即使MCK位设为1，主时钟也不会停止工作。在CPU时钟变为副时钟后才会停止。 - 将MCK位的设置从0更改为1之前，需要将使用主时钟频率工作的周边器件关闭。 - 主时钟停止后，器件将工作于副时钟，若将CPU时钟或周边器件的工作时钟切换回主时钟，则需要将MCK位清零并通过软件确保振荡稳定时间。							

MFRC	主时钟片上反馈电阻的使用						
0	使用						
1	不使用						

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

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

注 CLS 位为只读位。

注意事项 1. 在通过 CLKOUT 向外输出时钟信号时，不要改变 CPU 时钟的频率（通过 CK3 到 CK0 位）。
 2. 对 CK3 位操作时，请使用位操作指令。如果使用 8 位操作指令，不要改变 CK2 到 CK0 位的值。

备注 ×: 不必考虑

(a) 从主时钟操作切换至副时钟操作的举例

- <1> CK3 位 ← 1: 推荐使用位指令操作。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 ← 1: 只有在需要停止主时钟工作的时候才会将 MCK 位置 1。

注意事项 1. 当停止主时钟工作时, PLL 也要停止。另外, 使用主时钟工作的片上周边器件也要停止。
2. 如果下面的条件不满足, 那么请改变 CK2 到 CK0 位的设置以满足条件要求, 然后再切换到副时钟工作模式。

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

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

[示例]

```

_DMA_DISABLE:
    crl      0, DCHCn[r0]      --禁止 DMA 操作。n = 0 到 3
<1> _SET_SUB_RUN :
    st.b     r0, PRCMD[r0]
    set1     3, PCC[r0]        -- CK3 位 ← 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 ← 1, 主时钟停止
_DMA_ENABLE:
    setl     0, DCHCn[r0]      --允许 DMA 操作。n = 0 到 3

```

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

(b) 从副时钟操作切换至主时钟操作的举例

- | | | |
|-----|-----------------------|--|
| <1> | MCK 位 ← 1: | 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 ← 1: | 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，使用主时钟工作的片上周边器件开始工作。如果它们在振荡稳定之前就被启动，则可能引起误操作。

[示例]

```

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

```

备注 以上的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。

复位之后: 00H		R/W	地址: FFFFF80CH				
RCM	7	6	5	4	3	2	1
	0	0	0	0	0	0	0
<0>							
RSTOP							
内部振荡器的振荡控制							
0 内部振荡器振荡							
1 内部振荡器停止							

- 注意事项
- 1. CPU 工作于内置振荡器时钟 (CCLS.CCLS_F 位 = 1) 时, 不可以停止内置振荡器的工作。不要将 RSTOP 位置 1。
 - 2. 即使 RSTOP 位为 1 时, 如果 CCLS.CCLS_F 位被置 1 (振荡稳定时间中产生 WDT 溢出时), 内置振荡器也会开始振荡。此时, RSTOP 位仍被设为 1。

(3) CPU 操作时钟状态寄存器 (CCLS)

CCLS 寄存器是用于指示 CPU 操作时钟状态的寄存器。
只能对该寄存器进行 8 位或 1 位的读取操作。
复位输入将把该寄存器置为 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							
CPU操作时钟状态							
0 工作于主时钟 (f _x) 或副时钟(f _{xτ}).							
1 工作于内部振荡时钟 (f _R).							

<R> 注 如果在复位释放后的振荡器稳定期间发生 WDT 溢出, CPU 使用内部振荡时钟 (f_R) 操作。这时, CCLS_F 位置为 1, 且复位值为 01H。

6.4 操作

6.4.1 各时钟的操作

下表显示了各时钟的操作状态。

表 6-1. 各时钟的操作状态

寄存器设置 以及操作状态 目标时钟	PCC 寄存器								
	CLS 位= 0, MCK 位= 0					CLS 位 = 1, MCK 位 = 0		CLS 位 = 1, MCK 位 = 1	
	复位过程中	振荡稳定 时间等待 过程中	HALT 模式	IDLE1, IDLE2 模式	STOP 模式	副时钟 模式	Sub-IDLE 模式	副时钟 模式	Sub-IDLE 模式
主时钟振荡器 (f_X)	×	√	√	√	×	√	√	×	×
副时钟振荡器 (f_{XT})	√	√	√	√	√	√	√	√	√
CPU 时钟 (f_{CPU})	×	×	×	×	×	√	×	√	×
内部系统时钟 (f_{CLK})	×	×	√	×	×	√	×	√	×
主时钟 (PLL 模式, f_{XX})	×	√ ^注	√	×	×	√	√	×	×
周边时钟 (f_{XX} 到 $f_{XX}/1, 024$)	×	×	√	×	×	√	×	×	×
WT 时钟 (主)	×	√	√	√	×	√	√	×	×
WT 时钟 (子)	√	√	√	√	√	√	√	√	√
WDT2 时钟 (内置振荡时钟)	×	√	√	√	√	√	√	√	√
WDT2 时钟 (主)	×	×	√	×	×	√	×	×	×
WDT2 时钟 (子)	√	√	√	√	√	√	√	√	√

注 锁定所需时间

备注 √: 可操作
 ×: 停止

6.4.2 时钟输出功能

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

内部系统时钟 (f_{CLK}) 可通过 PCC.CK3 ~ 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，因此使用之前请按照需求设置端口 CM 的控制寄存器。

CLKOUT 引脚的状态与表 6-1 中所示的内部系统时钟的状态是一致的，在内部系统时钟为可操作状态时，该引脚可以输出时钟信号。当内部系统时钟处于停止状态时，CLKOUT 引脚输出为低电平。然而，在复位之后直到它被设置为输出模式之前，管脚 CLKOUT 处于端口模式 (PCM1 管脚：输入模式)。因此，这个管脚的状态为高阻态。

6.5 PLL 功能

6.5.1 概述

<R> 在 V850ES/ SG2 和 V850ES/ SG2-H 系列产品中，CPU 和片上周边器件的时钟信号可以选择频率为晶振频率的 4 或 8 倍的 PLL 模式和时钟直通模式。

- V850ES/SG2
使用 PLL 时： 输入时钟 = 2.5 到 5 MHz（输出：10 到 20 MHz）
时钟直通模式： 输入时钟 = 2.5 到 10 MHz（输出：2.5 到 10 MHz）

- <R>
- V850ES/SG2-H
使用 PLL 功能时（x4）： 输入时钟 = 2.5 到 5 MHz（输出：10 到 20 MHz）
使用 PLL 功能时（x8）： 输入时钟 = 2.5 到 4 MHz（输出：20 到 32 MHz）
时钟直通模式： 输入时钟 = 2.5 到 8 MHz（输出：2.5 到 8 MHz）

6.5.2 寄存器

（1）PLL 控制寄存器（PLLCTL）

PLLCTL 寄存器是用来控制 PLL 功能的 8 位寄存器。
可以对该寄存器进行 8 位或 1 位的读写操作。
复位输入将把该寄存器置为 01H。

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

- <R>
- 注意事项
1. 要停止 PLL 的工作，首先设置时钟直通模式（SELPLL 位=0），等待至少 8 个时钟，然后停止 PLL（PLLON 位=0）。当 PLLON 位被清零，SELPLL 位被自动清零（时钟直通模式），但在上述过程中，请保证停止 PLL。
 2. 只有在 PLL 时钟频率稳定后，SELPLL 位才能够被置 1。如果时钟频率没有稳定（未入锁），那么无论向该位写入的数据为何值，SELPLL 位都会被写入"0"。

(2) 时钟控制寄存器 (CKC)

CKC 寄存器是一个特殊寄存器。只有按照特定流程才能将数据写入该寄存器（见 3.4.8 特殊寄存器）。

CKC 寄存器用于控制 PLL 模式下的内部系统时钟。

可以对该寄存器进行 8 位或 1 位的读写操作。

复位操作将把该寄存器设置为 0AH。

复位后: 0AH R/W 地址: FFFF822H

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

CKDIV0	PLL模式下系统内部时钟 (fxx) in PLL mode
0	$f_{xx} = 4 \times f_x$ ($f_x = 2.5$ 到 5.0 MHz)
1	V850ES/SG2: $f_{xx} = 8 \times f_x$ ($f_x = 2.5$ MHz) V850ES/SG2-H: $f_{xx} = 8 \times f_x$ ($f_x = 2.5$ 到 4.0 MHz)

注意事项 1. PLL 模式不能用于下列振荡器频率。

- V850ES/SG2: $5.0\text{ MHz} < f_x \leq 10.0\text{ MHz}$
- V850ES/SG2-H: $5.0\text{ MHz} < f_x \leq 8.0\text{ MHz}$

2. 通过 CKC 寄存器改变倍频系数为 4 倍或 8 倍间之前, 要先设置时钟为直通模式并停止 PLL 工作。
3. 请务必将第 3 位和第 1 位置 1, 将第 7~4 位和第 2 位清零。

备注 CKC 寄存器设置会影响 CPU 时钟和片上周边时钟的分频, 而 PCC 寄存器设置仅会影响 CPU 时钟的分频。

(3) 锁定寄存器 (LOCKR)

上电后或 STOP 模式解除后在给定频率会发生相位锁定，入锁时间（频率稳定时间）是指这个过程稳定下来所需要的时间。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。

复位之后: 00H R 地址: FFFFF824H

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

LOCK	PLL 锁定状态检查
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^注
- 在 IDLE2 或 STOP 模式下。
- 设置 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 位都置 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]

- 复位释放后的振荡稳定时间（OSTS 寄存器的默认时间（见 24.2 (3) 振荡稳定时间选择寄存器 (OSTS)））溢出时。
- 在 PLL 操作状态下设置了 STOP 模式，STOP 模式解除后，振荡稳定定时器溢出（时间由 OSTS 寄存器设置）时。
- 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 模式解除过程中插入的设置时间（由 OSTS 寄存器设置的时间）结束后。

(4) PLL 入锁时间指定寄存器 (PLLS)

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

可以对该寄存器进行 8 位的读写操作。

复位输入将把该寄存器设置为 03H。

复位后: 03H R/W 地址: FFFFF6C1H

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

PLLS1	PLLS0	选择PLL入锁时间
0	0	$2^{10}/f_x$
0	1	$2^{11}/f_x$
1	0	$2^{12}/f_x$
1	1	$2^{13}/f_x$ (默认值)

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

6.5.3 使用方法**(1) 使用 PLL 时**

- 复位信号被释放之后, PLL 处于工作状态 (PLLCTL.PLLON 位 = 1), 但由于默认模式为时钟直通模式 (PLLCTL.SELPLL 位 = 0), 为使 PLL 有效, 请选择 PLL 模式 (SELPLL 位 = 1)。
- 要使能 PLL 操作: 首先要将 PLLON 位置 1, 然后在 LOCKR.LOCK 位 = 0 之后将 SELPLL 位置 1。停止 PLL 工作: 首先选择时钟直通模式 (SELPLL 位 = 0), 等待 8 个或更多个时钟周期, 然后停止 PLL 工作 (PLLON 位 = 0)。
- 系统状态切换至 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) 在时钟直通模式下设置 IDLE2/STOP 模式

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

(b) 在 PLL 工作模式设置 IDLE2/STOP 模式

- STOP 模式: 设置 OSTS 寄存器使振荡稳定时间为 1 ms (最小值) 或更长。
- IDLE2 模式: 设置 OSTS 寄存器使设置时间为 800 μ s (最小值) 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, 默认选中时钟直通模式 (SELPLL 位 = 0), 但 PLL 处于工作状态 (PLLON 位 = 1), 所以请务必手动关闭 PLL (PLLON 位 = 0)。

<R>

第 7 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。

<R> V850ES/SG2 和 V850ES/SG2-H 有 6 个定时器 / 事件计数器通道，TMP0 到 TMP5。

7.1 概述

以下是 TMPn 的概况。

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

备注 n = 0 到 5

7.2 功能

TMPn 具有以下功能。

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

备注 n = 0 到 5

7.3 配置

TMPn 包含以下硬件。

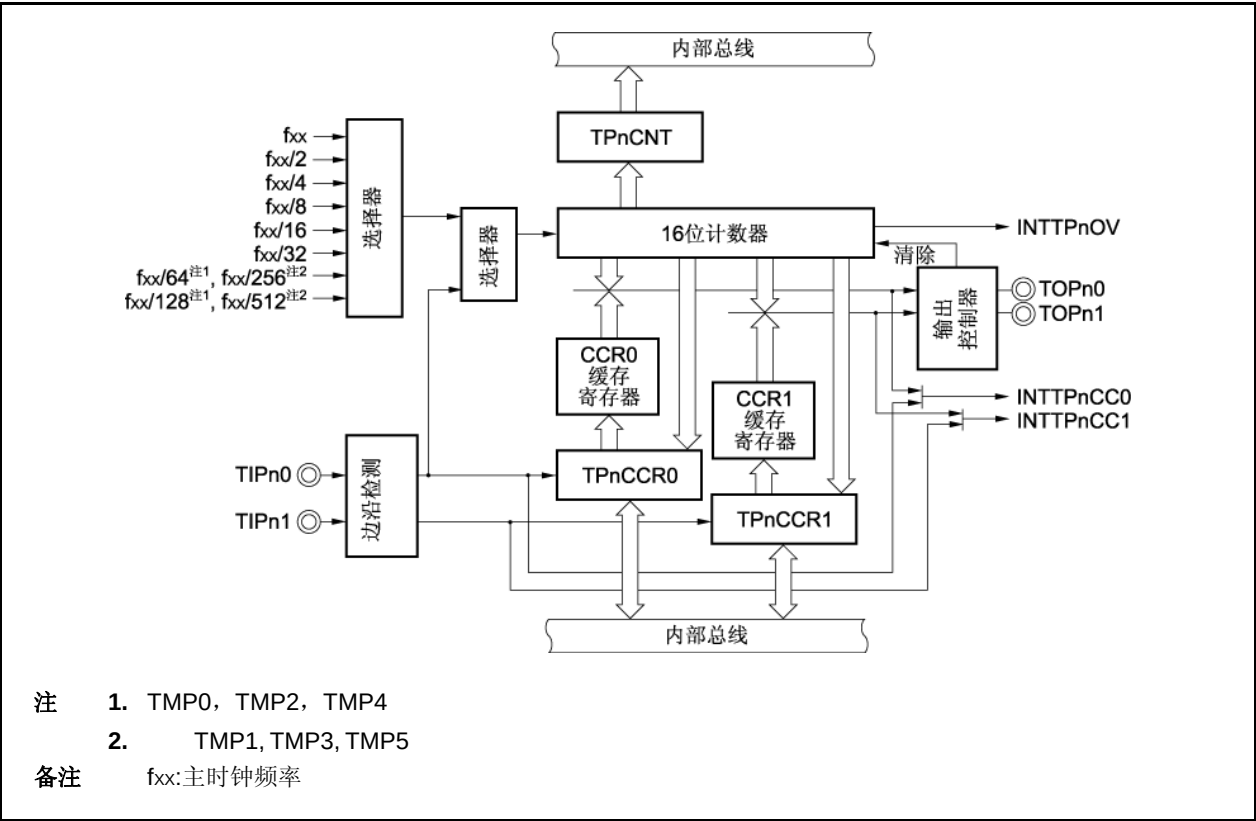
表 7-1. TMPn 的配置

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

注 1. TIPn0 引脚的复用功能包括捕捉触发输入信号，外部事件计数输入信号和外部触发输入信号。
2. 使用 TIPn0, TIPn1, TOPn0 和 TOPn1 引脚功能时，请参照表 4-15 端口引脚作为复用引脚使用。

备注 n = 0 到 5

图 7-1. TMPn 的框图



(1) 16 位计数器

该 16 位计数器可以对内部时钟或外部事件进行计数。

可以使用 TPnCNT 寄存器读取该计数器的计数值。

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

复位输入将 TPnCE 位清零。

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

CCR0 缓冲寄存器不能被读 / 写。

复位后 CCR0 缓冲寄存器和 TPnCCR0 寄存器都将被清零。

(3) CCR1 缓冲寄存器

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

当 TPnCCR1 寄存器被作为比较寄存器使用时, 向 TPnCCR1 寄存器写入的数据将被传送至 CCR1 缓冲寄存器。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)。

CCR1 缓冲寄存器不能被读 / 写。

复位后 CCR1 缓冲寄存器和 TPnCCR1 寄存器都将被清零。

(4) 边沿检测器

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

(5) 输出控制器

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

(6) 选择器

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

7.4 寄存器

用于控制 TMPn 的寄存器如下所述。

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

备注

1. 使用 TIPn0, TIPn1, TOPn0 和 TOPn1 引脚功能时, 请参照表 4-15 端口引脚作为复用引脚使用。
2. n = 0 到 5

(1) TMPn 控制寄存器 0 (TPnCTL0)

TPnCTL0 寄存器是用于控制 TMPn 操作的 8 位寄存器。

可对该寄存器进行 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 读/写 地址: TP0CTL0 FFFFF590H, TP1CTL0 FFFFF5A0H,
TP2CTL0 FFFFF5B0H, TP3CTL0 FFFFF5C0H,
TP4CTL0 FFFFF5D0H, TP5CTL0 FFFFF5E0H

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

TPnCE	TMPn 操作控制
0	TMPn 操作禁止 (TMPn 异步复位 ^注)。
1	TMPn 操作使能 TMPn 操作启动。

TPnCKS2	TPnCKS1	TPnCKS0	内部计数时钟选择	
			n = 0, 2, 4	n = 1, 3, 5
0	0	0	fxx	
0	0	1	fxx/2	
0	1	0	fxx/4	
0	1	1	fxx/8	
1	0	0	fxx/16	
1	0	1	fxx/32	
1	1	0	fxx/64	fxx/256
1	1	1	fxx/128	fxx/512

<R> 注 TPnOPT0.TPnOVF 位和 16 位计数器同时复位。而且，当 16 位计数器复位时定时器输出管脚 (TOPn0 和 TOPn1 管脚) 复位到被 TPnIOC0 寄存器设置的状态。

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

备注 fxx:主时钟频率

(2) TMPn 控制寄存器 1 (TPnCTL1)

TPnCTL1 寄存器是用于控制 TMPn 操作的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

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

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

TPnEST	软件触发控制
0	-
1	为外部触发输入产生有效信号 · 单脉冲输出模式中: 向TPnEST位写1输出单脉冲波形作为触发信号。 · 外部触发脉冲输出模式: 向TPnEST位写1输出PWM波形作为触发信号。
读 TPnEST 位总为 0.	

TPnEEE	计数时钟选择
0	禁止外部事件输入操作 (TPn0 引脚). (对TPnCTL0.TPnCKS0到TPnCKS2 位选择的计数时钟进行计数)
1	使能外部事件输入操作 (TPn0 引脚). (对外部事件计数输入信号 (TPn0 引脚)计数)
TPnEEE 位选择对内部计数时钟或外部事件计数输入的有效沿进行计数。	

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

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

(3) TMPn I/O 控制寄存器 0 (TPnIOC0)

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

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

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

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

<R>

TPnOL1	TOPn1 引脚输出电平设置 ^注
0	TOPn1 引脚开始时高电平
1	TOPn1 引脚开始时低电平

TPnOE1	TOPn1 引脚输出设置
0	禁止定时器输出 · TPnOL1 位= 0: TOPn1 引脚输出低电平 · TPnOL1 位= 1: TOPn1 引脚输出高电平
1	使能定时器输出(TOPn1 引脚输出脉冲).

<R>

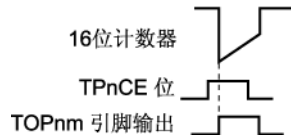
TPnOL0	TOPn0 引脚输出电平设置 ^注
0	TOPn0 引脚开始时高电平
1	TOPn0 引脚开始时低电平

TPnOE0	TOPn0 引脚输出设置
0	禁止定时器输出 · TPnOL0 位= 0: TOPn1 引脚输出低电平 · TPnOL0 位= 1: TOPn1 引脚输出高电平
1	使能定时器输出(TOPn0引脚输出脉冲).

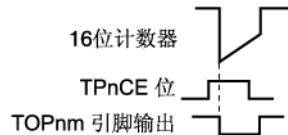
<R>

注 由 TPnOLm 位指定的定时器输出管脚 (TOPn0, TOPn1) 输出电平如下所示 (m = 0, 1)。

· 当 TPnOLm 位= 0



· 当 TPnOLm 位= 1



<R>

- 注意事项
1. 当端口设置为输出 TOPn0 和 TOPn1 时, 如果寄存器 TPnIOC0 的设置被重写, 管脚输出改变。因此, 通过设置此端口为输入模式并让管脚的输出状态位高阻态来记录管脚状态的变化。
 2. 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。
 3. 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。(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。

<R>

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 读/写 地址: TP0IOC1 FFFFF593H, TP1IOC1 FFFFF5A3H,
TP2IOC1 FFFFF5B3H, TP3IOC1 FFFFF5C3H,
TP4IOC1 FFFFF5D3H, TP5IOC1 FFFFF5E3H

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

TPnIS3	TPnIS2	捕捉触发输入信号(TIPn1 引脚) 有效沿设置
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) 有效沿设置
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 (仅当 TPnOPT0.TPnCCS1, TPnCCS0 位 = 1) 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 读/写 地址: TP0IOC2 FFFFF594H, TP1IOC2 FFFFF5A4H,
TP2IOC2 FFFFF5B4H, TP3IOC2 FFFFF5C4H,
TP4IOC2 FFFFF5D4H, TP5IOC2 FFFFF5E4H

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

TPnEES1	TPnEES0	外部事件计数输入信号 (TIPn0 引脚) 有效沿设置
0	0	无边沿检测 (外部事件技术无效)
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) 有效沿设置
0	0	无边沿检测 (外部事件技术无效)
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 位 = 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 读/写 地址: TP0OPT0 FFFFF595H, TP1OPT0 FFFFF5A5H,
TP2OPT0 FFFFF5B5H, TP3OPT0 FFFFF5C5H,
TP4OPT0 FFFFF5D5H, TP5OPT0 FFFFF5E5H

	7	6	5	4	3	2	1	<0>
TPnOPT0	0	0	TPnCCS1	TPnCCS0	0	0	0	TPnOVF

(n = 0 到 5)

TPnCCS1	TPnCCR1 寄存器捕捉/比较选择
0	选择比较寄存器
1	选择捕捉寄存器 (设置 TPnCTL0.TPnCE 位= 0清除)
仅在自由运行定时器模式下 TPnCCS1 位设置有效。	

TPnCCS0	TPnCCR0 寄存器捕捉/比较选择
0	选择比较寄存器
1	选择捕捉寄存器 (设置 TPnCTL0.TPnCE 位= 0清除)
仅在自由运行定时器模式下 TPnCCS0 位设置有效。	

TPnOVF	TMPn 溢出检测标志
置位 (1)	发生溢出
复位(0)	TPnOVF 位写 0 或 TPnCTL0.TPnCE 位= 0
- 自由运行模式或脉宽测量模式下16位计数器计数值从FFFFH到0000H溢出时TPnOVF位置1。 - TPnOVF位置1的同时产生一个溢出中断请求信号(INTTPnOV)。INTTPnOV 信号在自由运行定时器模式和脉宽测量模式以外的模式下不产生。 - 当TPnOVF 位= 1时即使读TPnOVF位或TPnOPT0 寄存器, TPnOVF位不清零。 - 在INTTPnOV信号产生后 TPnOVF 位清零前, 确认TPnOVF位置1。 - TPnOVF 位能被读写, 但是 TPnOVF 位不能被软件置1. 写1对TMPn操作无影响。	

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。

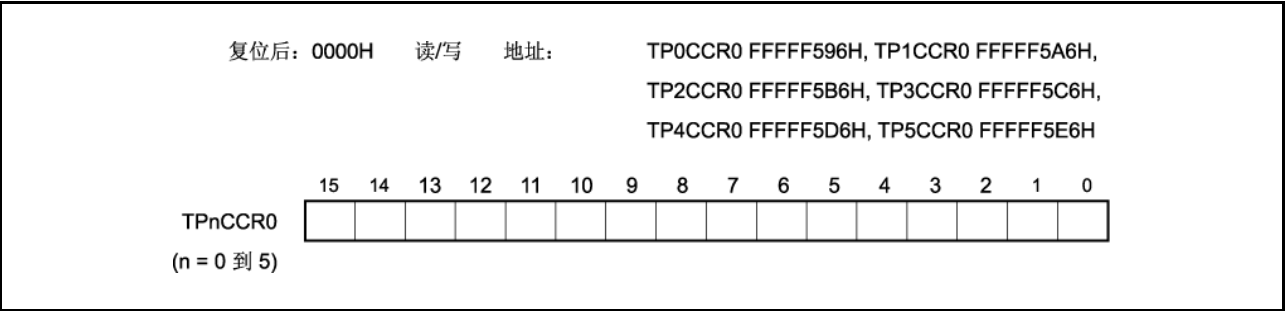
2. 请务必将第 1 到 3 位, 第 6 和第 7 位清零。

(7) TMPn 捕捉 / 比较寄存器 0 (TPnCCR0)

根据选择模式的不同，TPnCCR0 可用作 16 位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。详细内容请参阅 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。

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



(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）。

<R> 当 TPnCTL0.TPnCE 位=0 时比较寄存器不被清零。

(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。

当 TPnCTL0.TPnCE 位=0 时捕捉寄存器不被清零。

<R> 备注 n = 0 到 5

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
<R> 外部触发脉冲输出	比较寄存器	批量写入 [※]
单脉冲输出	比较寄存器	可随时写入
<R> PWM 输出	比较寄存器	批量写入 [※]
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	无

<R> 注 写寄存器 TPnCCR1 触发。

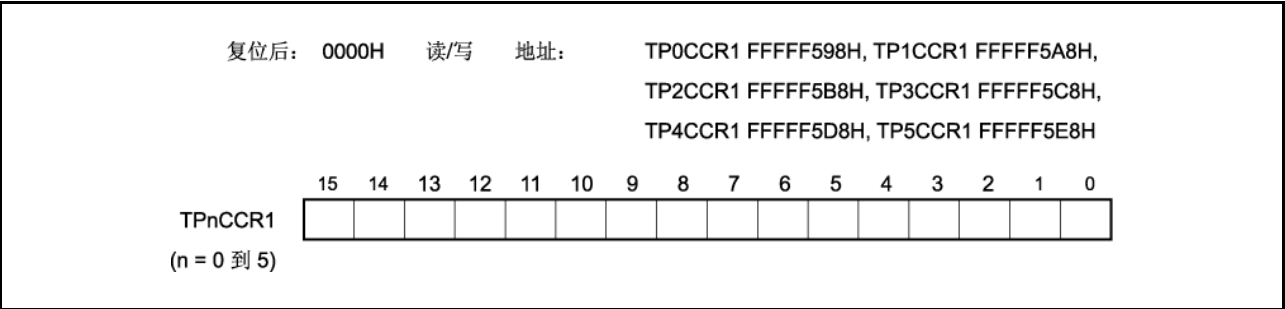
<R> 备注 关于随时写入和批量写入的细节，参阅 7.6（2）随时写入和批量写入。

(8) TMPn 捕捉 / 比较寄存器 1 (TPnCCR1)

根据选择模式的不同，TPnCCR1 可用作 16 位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。详细内容请参阅 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。

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



(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。
当 TPnCTL0.TPnCE 位=0 时，比较寄存器不被清零。

<R>

(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。
当 TPnCTL0.TPnCE 位=0 时捕捉寄存器不被清零。

<R>

备注 n = 0 到 5

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	批量写入 [※]
单脉冲输出	比较寄存器	可随时写入
PWM 输出	比较寄存器	批量写入 [※]
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	无

<R>

注 写寄存器 TPnCCR1 触发。

<R>

备注 关于随时写入和批量写入，参阅 7.6（2）随时写入和批量写入。

7.5 定时器输出操作

下表列出了 TOPn0 和 TOPn1 管脚的输出电平和操作。

表 7-4. 各模式下定时器输出控制

工作模式	TOPn1 管脚	TOPn0 管脚
间隔定时器模式	方波输出	
外部事件计数模式	无	
外部触发脉冲输出模式	外部触发脉冲输出	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（仅当比较功能被使用时）	
脉宽测量模式	无	

备注 n = 0 到 5

表 7-5. 定时器输出控制位控制下的 TOPn0 和 TOPn1 管脚真值表

TPnIOC0.TPnOLm 位	TPnIOC0.TPnOEm 位	TPnCTL0.TPnCE 位	TOPnm 管脚电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数开始前低电平，计数开始后高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数开始前高电平，计数开始后低电平

备注 n = 0 到 5
 m = 0, 1

7.6 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	批量写入
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	批量写入
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，单脉冲输出模式或脉宽测量模式时，请选择内部时钟作为计数时钟（通过将 TPnCTL1.TPnEEE 位清零）。

备注 n = 0 到 5

<R> (1) 计数基本操作

本节介绍 16 位计数器的基本操作。关于细节，参见各个模式下操作说明。

备注 n = 0 到 5

(a) 计数器开始操作

在所有模式下 TMPn 的 16 位计数器从缺省值 FFFFH 开始计数。

它从 FFFFH 到 0000H, 0001H, 0002H, 0003H.....向上计数。

(b) 清零操作

当 16 位计数器的值与比较寄存器的值匹配时被清零，当被捕捉时也清零。在计数器开始计数后或当计数器溢出时从 FFFFH 变到 0000H 的计数操作不是清零操作。因此，INTTPnCC0 和 INTTPnCC1 中断信号不会产生。

(c) 溢出操作

在自由运行定时器模式或脉宽测量模式下，计数器从 FFFFH 向上记到 0000H 时 16 位计数器溢出。如果计数器溢出，TPnOPT0.TPnOVF 位被置 1，中断请求信号（INTTPnOV）产生。注意在以下情况下 INTTPnOV 信号不被产生。

- 计数操作已经启动后
- 如果计数器值匹配比较器值 FFFFH 并被清零
- 当脉宽测量模式下 FFFFH 被捕捉并清零以及计数器从 FFFFH 向上记到 0000H。

注意事项 溢出中断请求信号（INTTPnOV）产生后，确认检查溢出标志（TPnOVF 位）被置为 1。

(d) 计数操作期间计数器读操作

计数操作时可以通过读取 TPnCNT 寄存器来读取 TMPn 的 16 位计数器的计数值。当 TPnCTL0.TPnCE 位 =1 时，可以通过读取 TPnCNT 寄存器来读取 16 位计数器的计数值。当 TPnCTL0.TPnCE 位=0 时，16 位计数器为 FFFFH，TPnCNT 寄存器位 0000H。

(e) 中断操作

TMPn 产生以下三种中断请求信号。

- INTTPnCC0 中断： 此信号作为 CCR0 缓存寄存器的匹配中断请求信号和 TPnCCR0 寄存器的捕捉中断请求信号。
- INTTPnCC1 中断： 此信号作为 CCR1 缓存寄存器的匹配中断请求信号和 TPnCCR1 寄存器的捕捉中断请求信号。
- INTTPnOV 中断： 此信号作为溢出中断请求信号。

<R>

(2) 随时写入和批量写入

TMPn 的 TPnCCR0 和 TPnCCR1 寄存器在计数器操作期间 (TPnCTL0.TPnCE 位=1) 可重写, 但 CCR0 和 CCR1 缓存寄存器的写入方法 (随时写入, 批量写入) 因模式而异。

(a) 随时写入

此模式下, 定时器操作期间随时从 TPnCCR0 和 TPnCCR1 寄存器传输数据到 CCR0 和 CCR1 缓存寄存器。n = 0 到 5

图 7-2. 随时写入基本操作流程

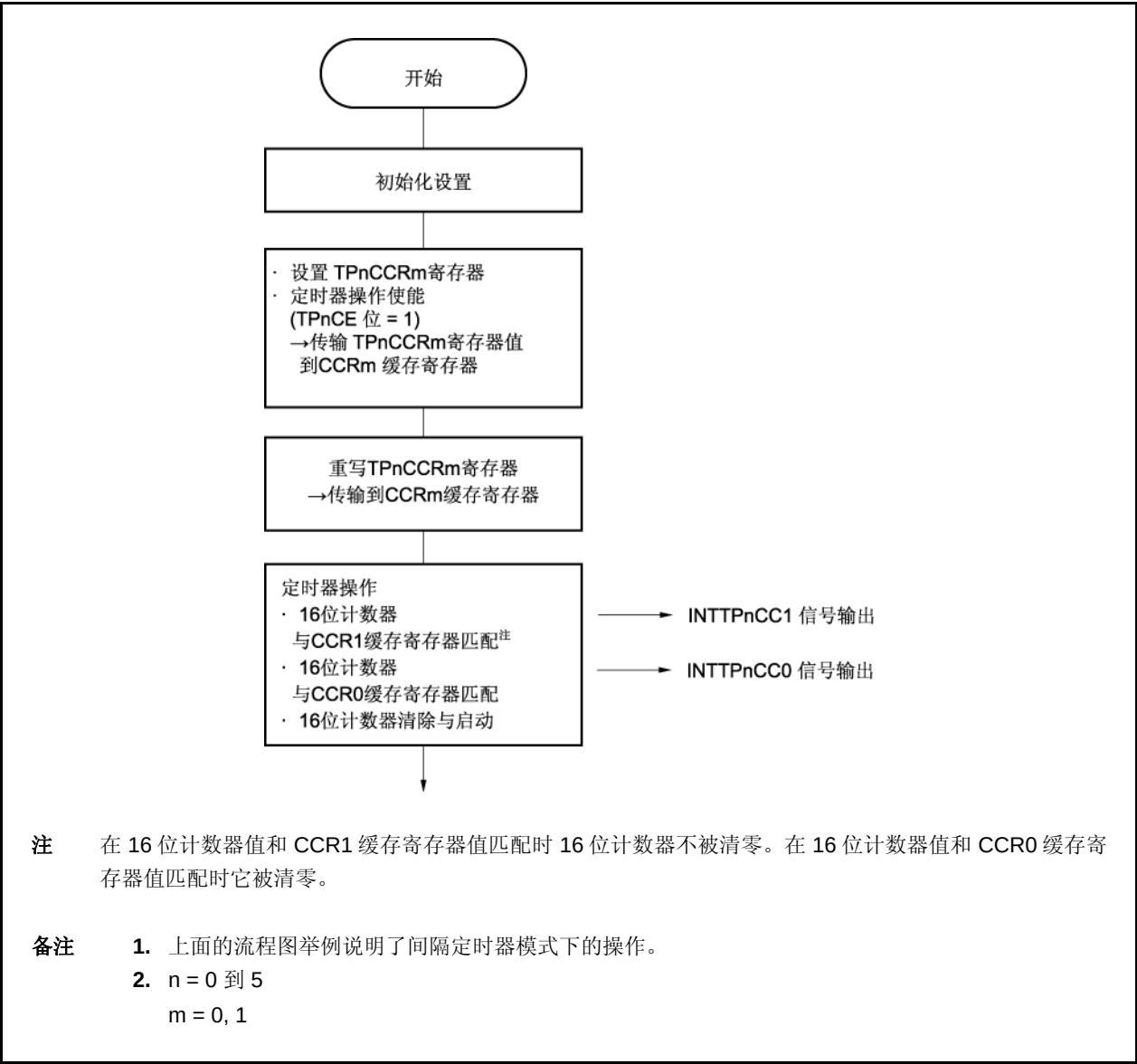
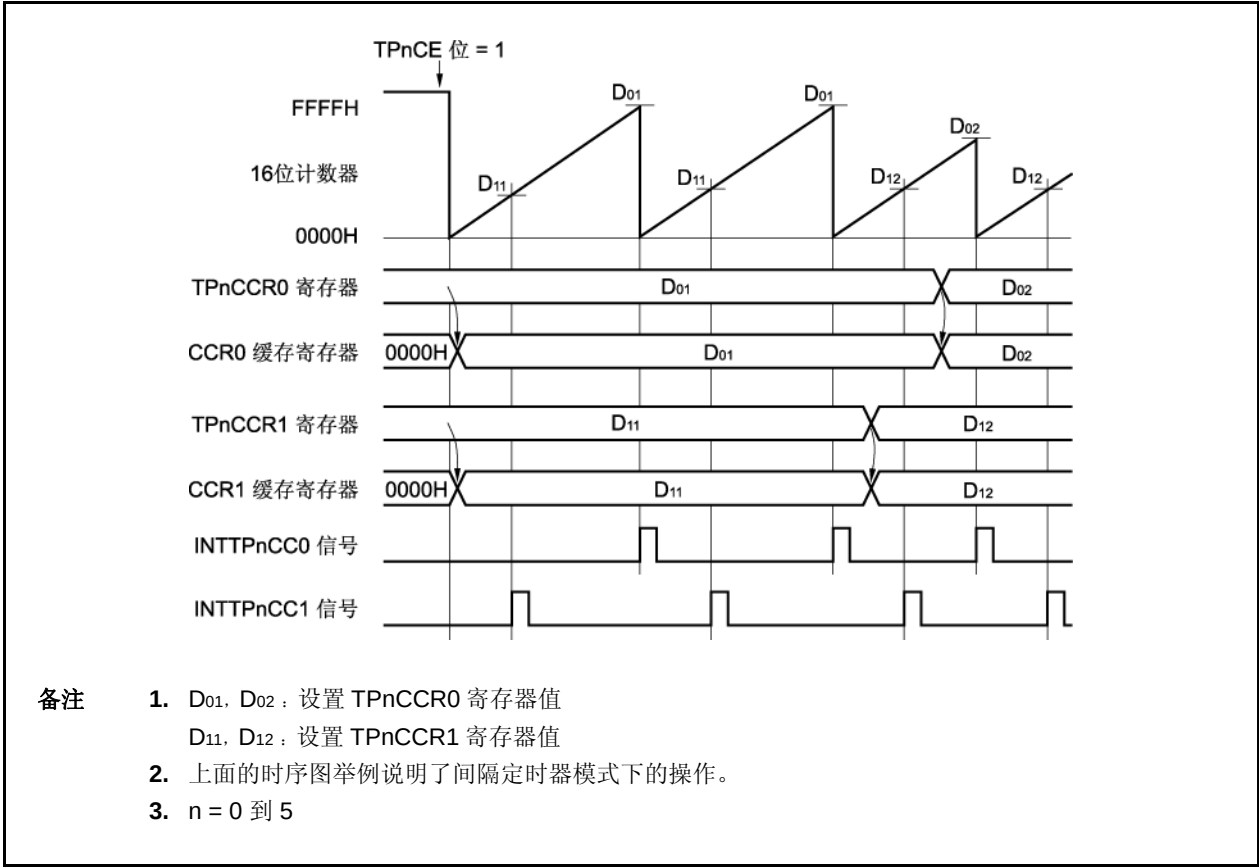


图 7-3. 随时写入时序



(b) 批量写入

此模式下，定时器操作期间每次传输全部数据从 TPnCCR0 和 TPnCCR1 寄存器到 CCR0 和 CCR1 缓存寄存器。当 CCR0 缓存寄存器的值与 16 位计数器的值匹配时，数据传输。通过写 TPnCCR1 寄存器使能传输。

使能或禁止下次传输取决于是否写 TPnCCR1 寄存器。

为了将 TPnCCR0 和 TPnCCR1 寄存器重写为 16 位计数器比较值的配置值（也就是说，为了传输到 CCR0 和 CCR1 缓存寄存器的值），有必要在 16 位计数器值和 CCR0 缓存寄存器值匹配前先重写 TPnCCR0 寄存器后写 TPnCCR1 寄存器。因此，当 16 位计数器的值和 CCR0 缓存寄存器的值匹配时，传输 TPnCCR0 和 TPnCCR1 寄存器的值到 CCR0 和 CCR1 缓存寄存器。因此即使仅希望重写 TPnCCR0 寄存器的值，也要写相同的值（和 TPnCCR1 寄存器当前值相同）到 TPnCCR1 寄存器。

图 7-4. 批量写入基本操作流程

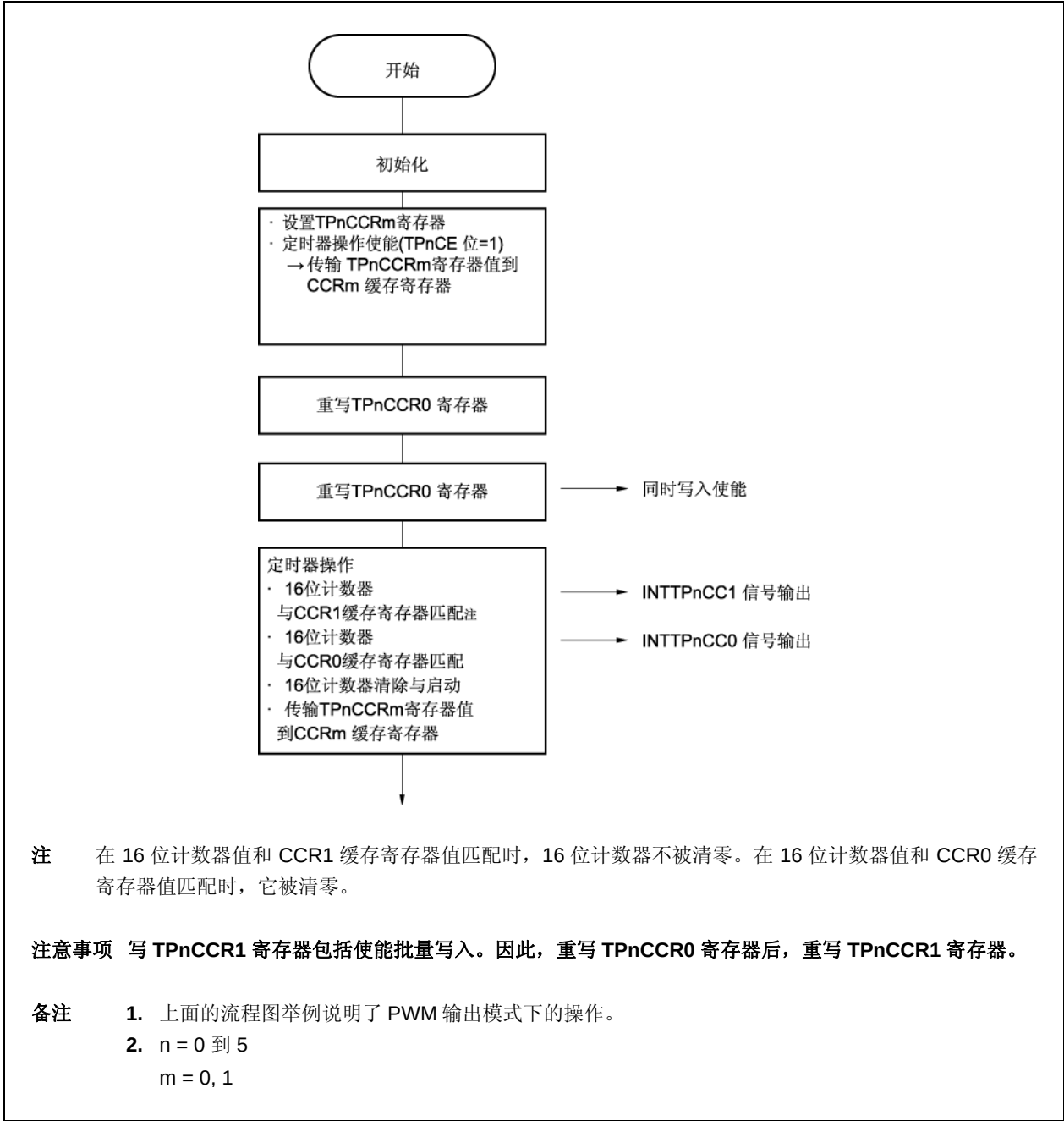
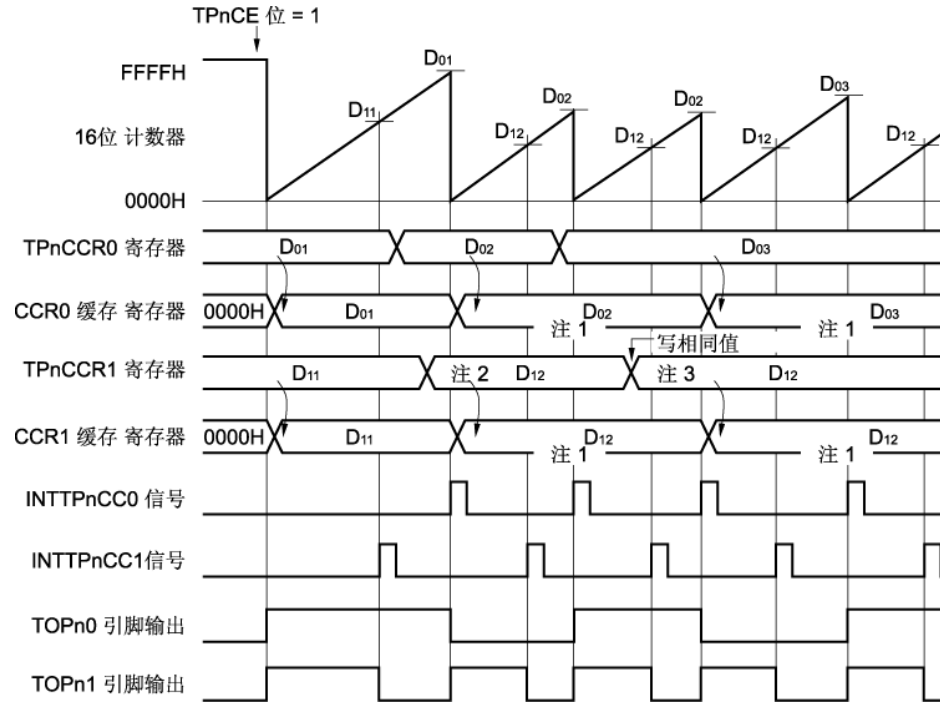


图 7-5. 批量写入时序



- 注**
1. 由于 TPnCCR1 寄存器没有重写，D03 没被传输。
 2. 由于 TPnCCR1 寄存器被写入 (D12)，当 16 位计数器的值和 TPnCCR0 寄存器 (D01) 的值匹配时，数据传输到 CCR1 缓存寄存器。
 3. 由于 TPnCCR1 寄存器被写入 (D12)，当 16 位计数器的值和 TPnCCR0 寄存器 (D02) 的值匹配时，数据传输到 CCR1 缓存寄存器。

- 备注**
1. D01, D02, D03: 设置 TPnCCR0 寄存器值
D11, D12: 设置 TPnCCR1 寄存器值
 2. 上面的时序图举例说明了 PWM 输出模式下的操作。
 3. n = 0 到 5

<R> 7.6.1 间隔定时器模式 (TPnMD2 到 TPnMD0 位 = 000)

在间隔定时器模式下，当 TPnCTL0.TPnCE 位 = 1 时，每隔 TPnCCR0 寄存器设置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。

间隔定时器模式中不使用 TPnCCR1 寄存器。然而，TPnCCR1 寄存器的设置值传输到 CCR1 缓存寄存器，当 16 位计数器的计数值匹配 CCR1 缓存寄存器的值时，产生一个比较匹配中断请求信号 (INTTPnCC1)。另外，当 INTTPnCC1 信号产生时，转换的占空比为 50% 的方波从 TOPn1 管脚输出。

即使当时器操作时，也能重写 TPnCCR0 和 TPnCCR1 寄存器的值。

图 7-6. 间隔定时器的配置图

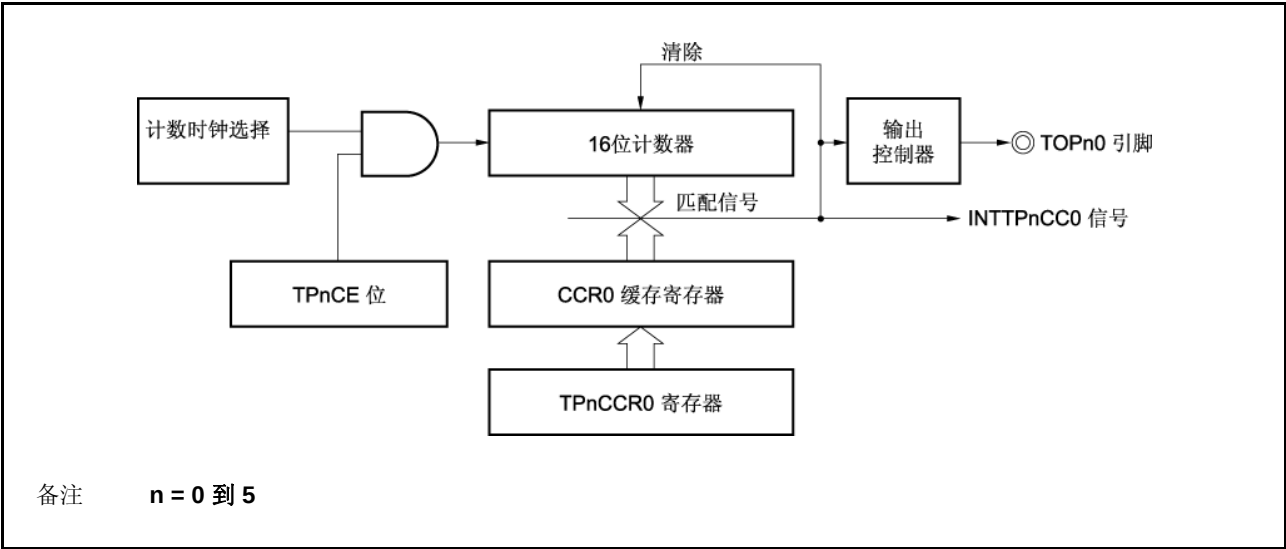
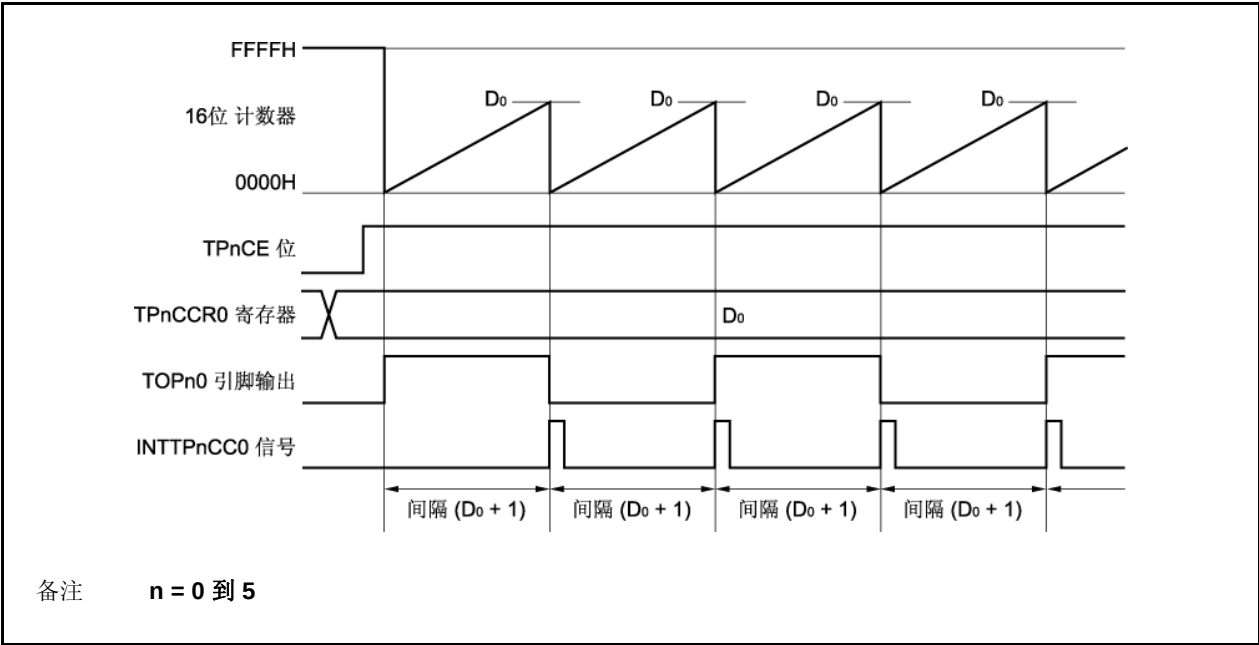


图 7-7. 间隔定时器模式操作的基本时序



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

间隔时间可由以下算式求得。

间隔时间 = (TPnCCR0 寄存器的设定值 + 1) × 计数时钟周期

备注 n = 0 到 5

图 7-8. 间隔定时器模式操作的寄存器设置 (1/3)

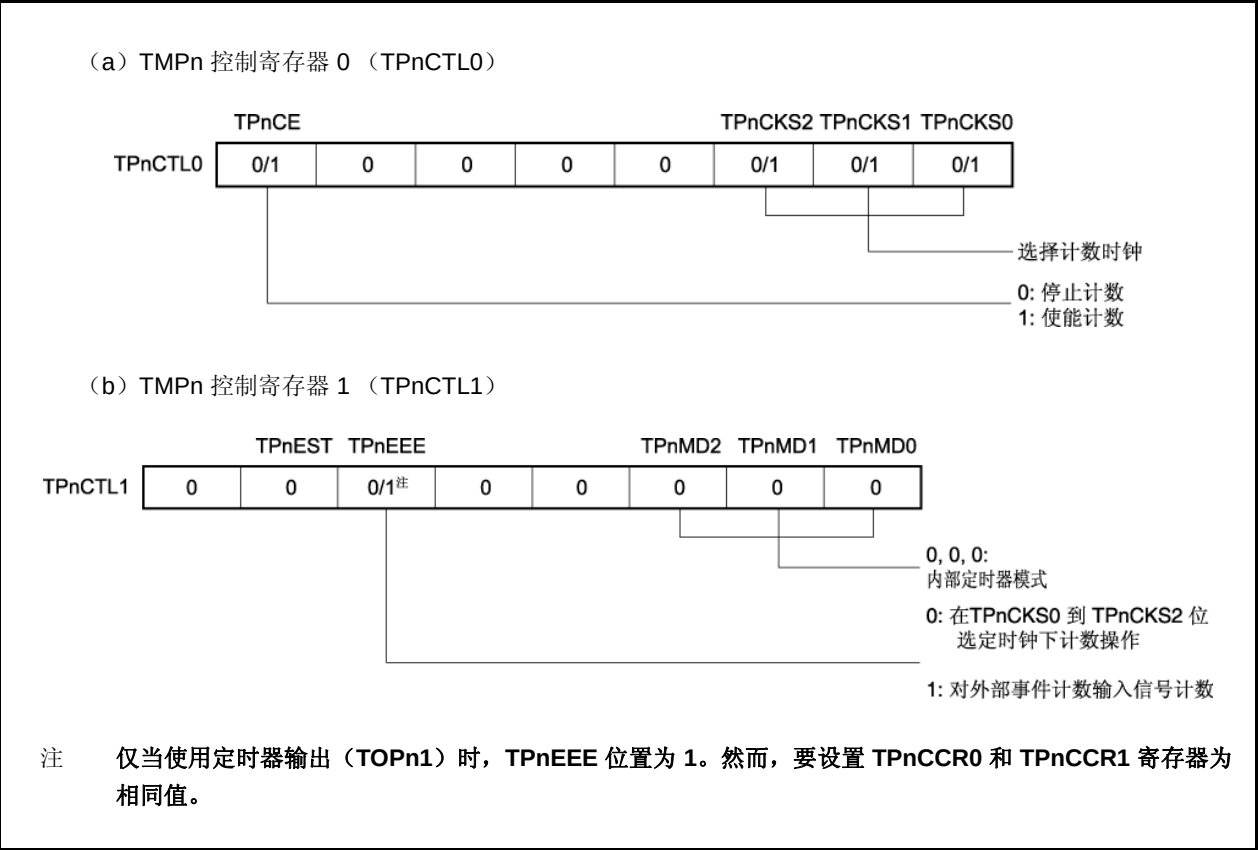
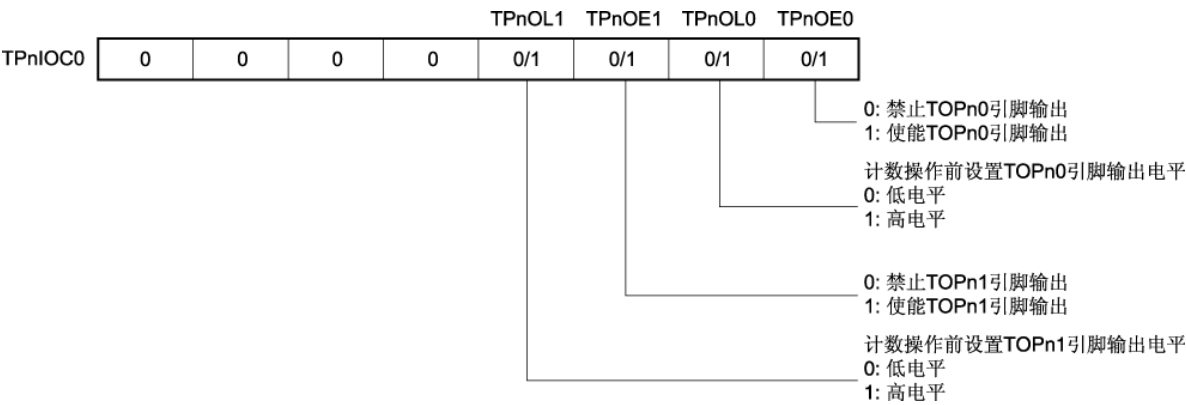


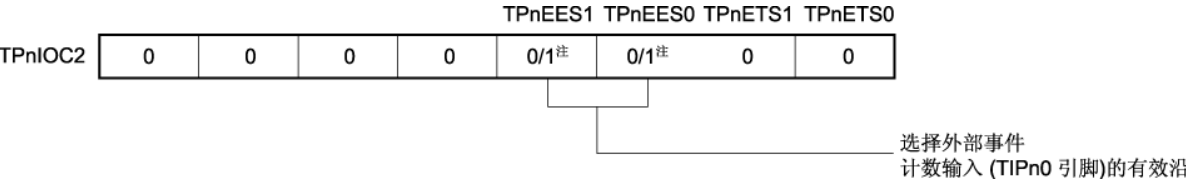
图 7-8. 间隔定时器模式操作的寄存器设置 (2/3)

(c) TMPn I/O 控制寄存器 0 (TPnIOC0)



<R>

(d) TMPn I/O 控制寄存器 2 (TPnIOC2)



注 仅当使用定时器输出 (TOPn1) 时, TPnEES1 和 TPnEES0 位能设置。然而, 要设置 TPnCCR0 和 TPnCCR1 寄存器为相同值。

(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)

如果将 TPnCCR0 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

备注 n = 0 到 5

图 7-8. 间隔定时器模式操作的寄存器设置 (3/3)

<R>

(g) TMPn 捕捉 / 比较寄存器 1 (TPnCCR1)

间隔定时器模式中不使用 TPnCCR1 寄存器。但对 TPnCCR1 寄存器设置的数值也将被传送到 CCR1 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 缓冲寄存器中的数值匹配时，TOPn1 管脚输出翻转，同时产生一个比较匹配中断请求信号 (INTTPnCC1)。

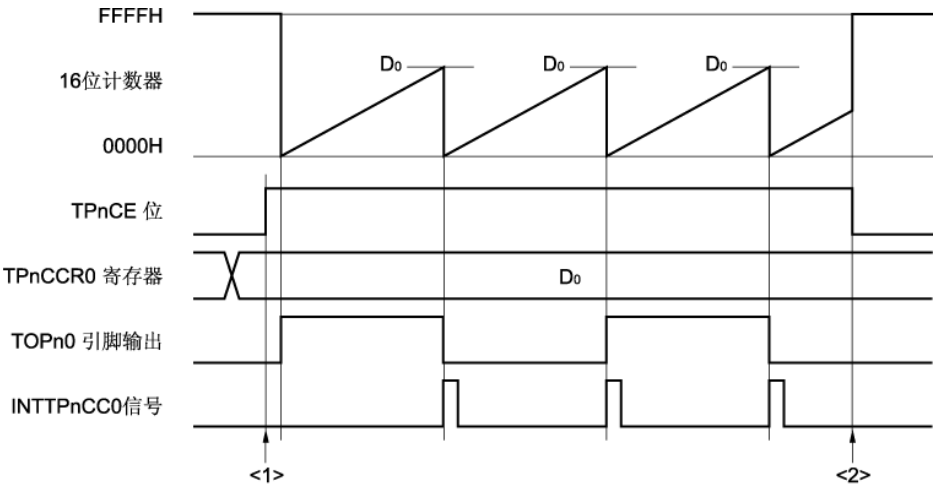
通过设置此寄存器为与 TPnCCR0 寄存器相同的值，从 TOPn1 管脚输出占空比为 50% 的方波。

当未使用 TPnCCR1 寄存器时，推荐将它的值设为 FFFFH。也通过中断屏蔽标志 (TPnCCIC1.TPnCCMK1) 屏蔽此寄存器。

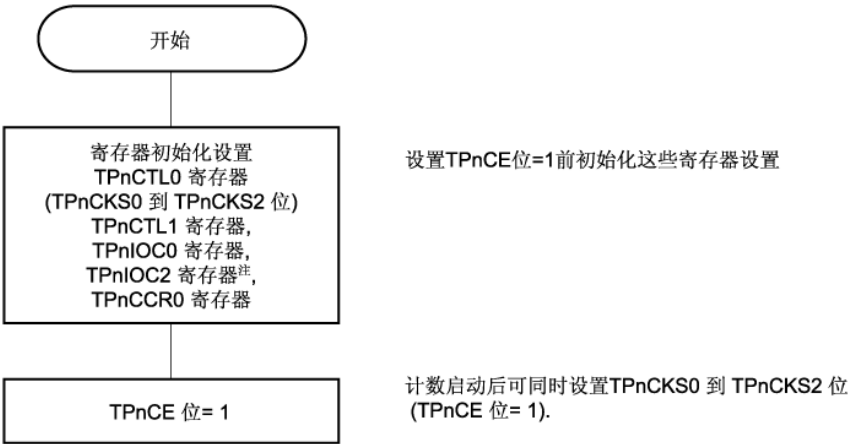
- 备注
1. 间隔定时器模式下，并未使用 TMPn I/O 控制寄存器 1 (TPnIOC1) 和 TMPn 选项寄存器 0 (TPnOPT0)。
 2. n = 0 到 5

(1) 间隔定时器模式操作流程

图 7-9. 间隔定时器模式下的软件处理流程

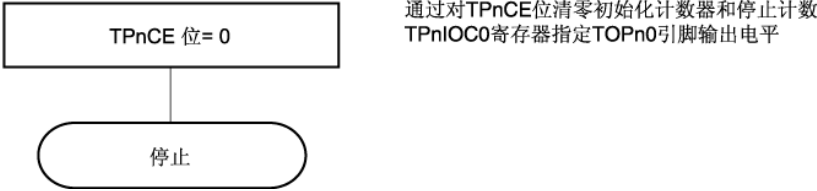


<1> 按数操作启动流程



注 仅当使用定时器输出 (TOPn1) 时能设置 TPnEES1 和 TPnEES0 位
然而,设置 TPnCCR0 和 TPnCCR1 寄存器为相同值.

<2> 计数操作停止流程



备注 n = 0 到 5

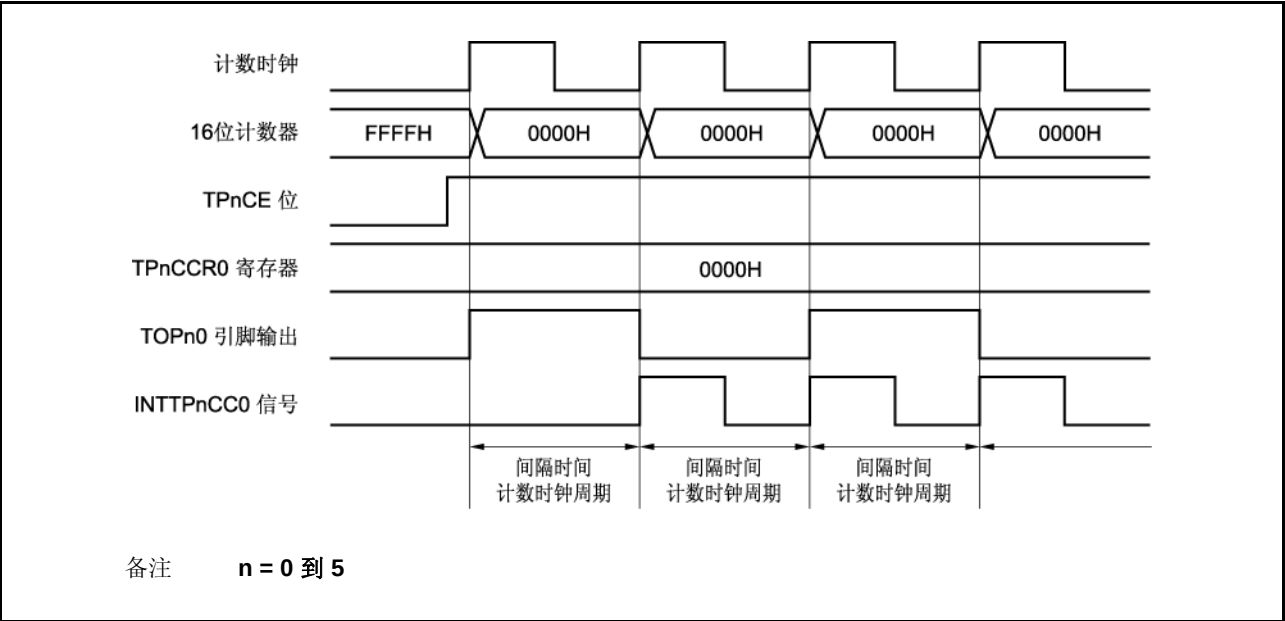
(2) 间隔定时器模式操作时序

<R>

(a) TPnCCR0 寄存器设置为 0000H 时的操作

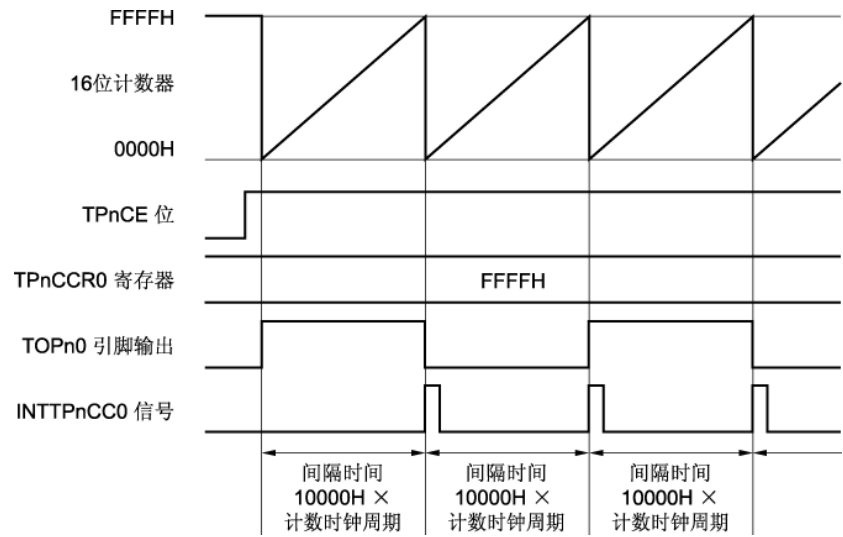
如果 TPnCCR0 寄存器被设成 0000H，在每个计数时钟上产生 INTTPnCC0 信号，并且 TOPn0 管脚的输出被取反。

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

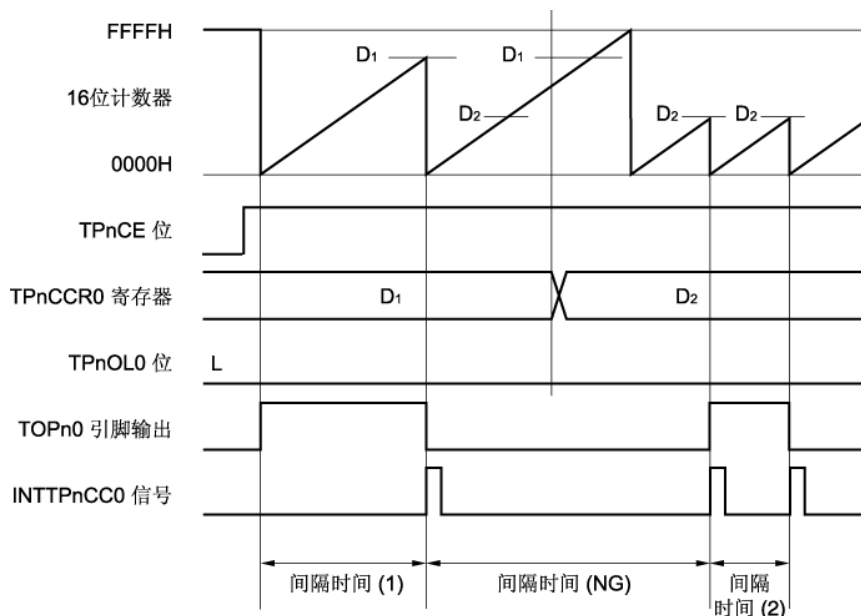
如果 TPnCCR0 寄存器被设置为 FFFFH，那么 16 位计数器将计数到 FFFFH。并在下一个计数时钟时刻同步被清零。同时产生 INTTPnCC0 信号，TOPn0 引脚电平反转。此时，不会产生溢出中断请求信号（INTTPnOV），溢出标志位（TPnOPT0.TPnOVF 位）也不会被置 1。



备注 n = 0 到 5

(c) 改写 TPnCCR0 寄存器时需要注意的事项

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。如果可能溢出，停止计数并改变设置值。



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

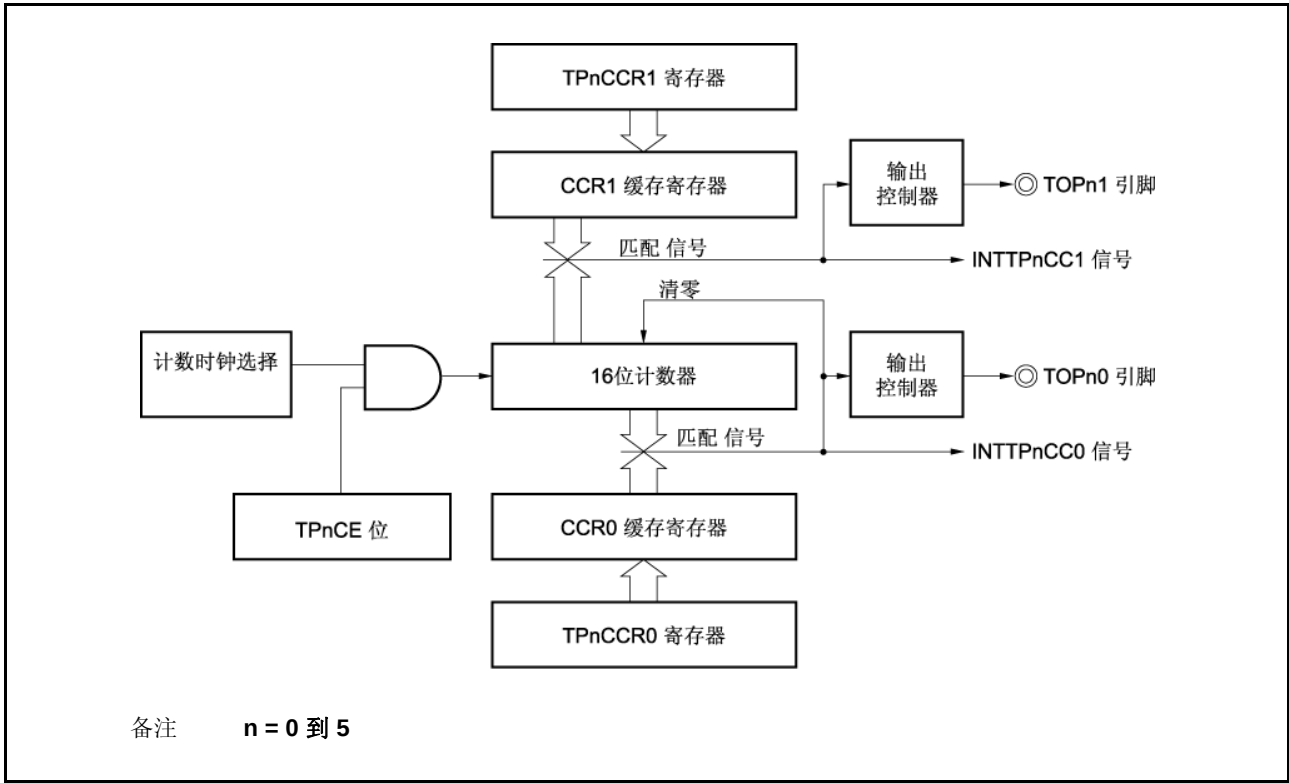
如果当前计数值大于 D_2 但小于 D_1 ，将 TPnCCR0 寄存器的值从 D_1 改变为 D_2 ，新的设定值将在改写后立即从 TPnCCR0 寄存器传送到 CCR0 缓冲寄存器中。因此，16 位计数器的值将与 D_2 比较。

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 7-10. TPnCCR1 寄存器的配置



<R>

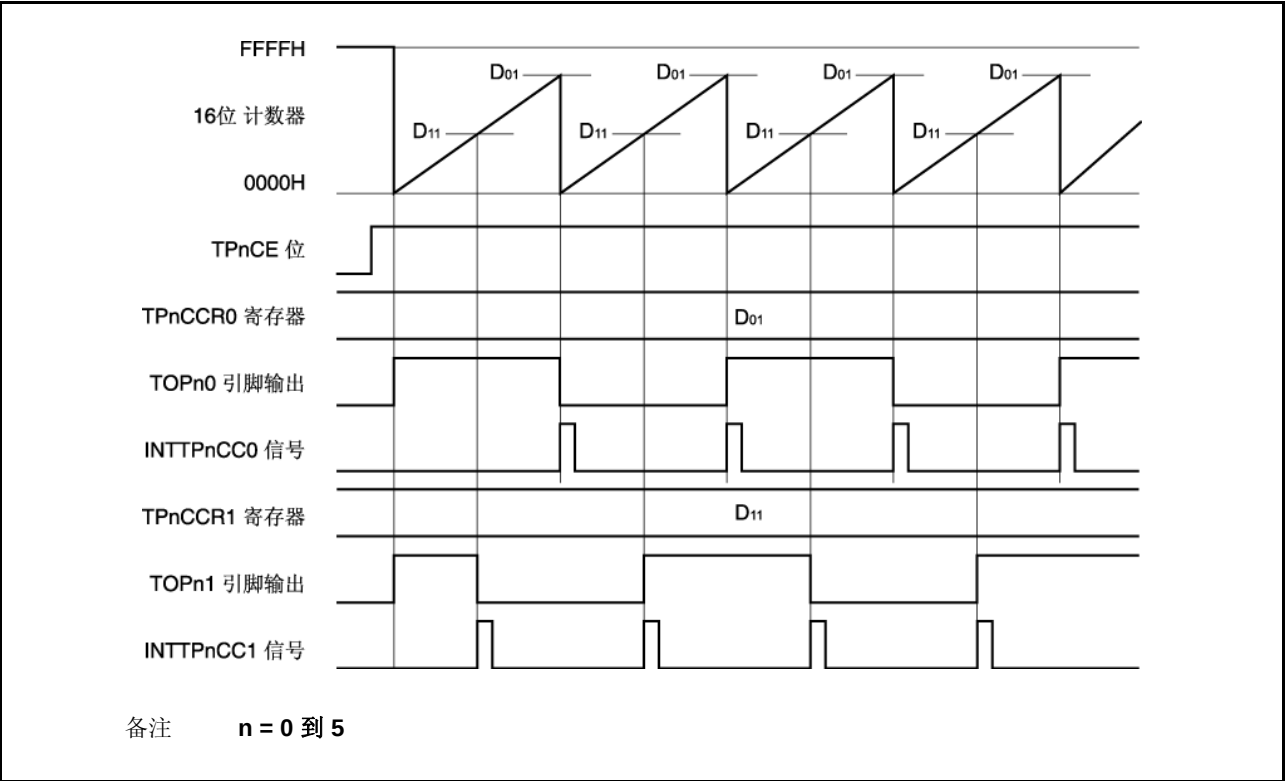
当 TPnCCR1 寄存器设置为 TPnCCR0 寄存器相同的值时，INTTPnCC1 信号与 INTTPnCC0 信号同时产生，TOPn1 管脚输出反转。也就是说，占空比为 50%的方波从 TOPn1 管脚输出。

下面显示出了 TPnCCR1 寄存器设置为与 TPnCCR0 寄存器不同值时的操作。

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 管脚输出一个短脉冲后输出占空比为 50%的方波。

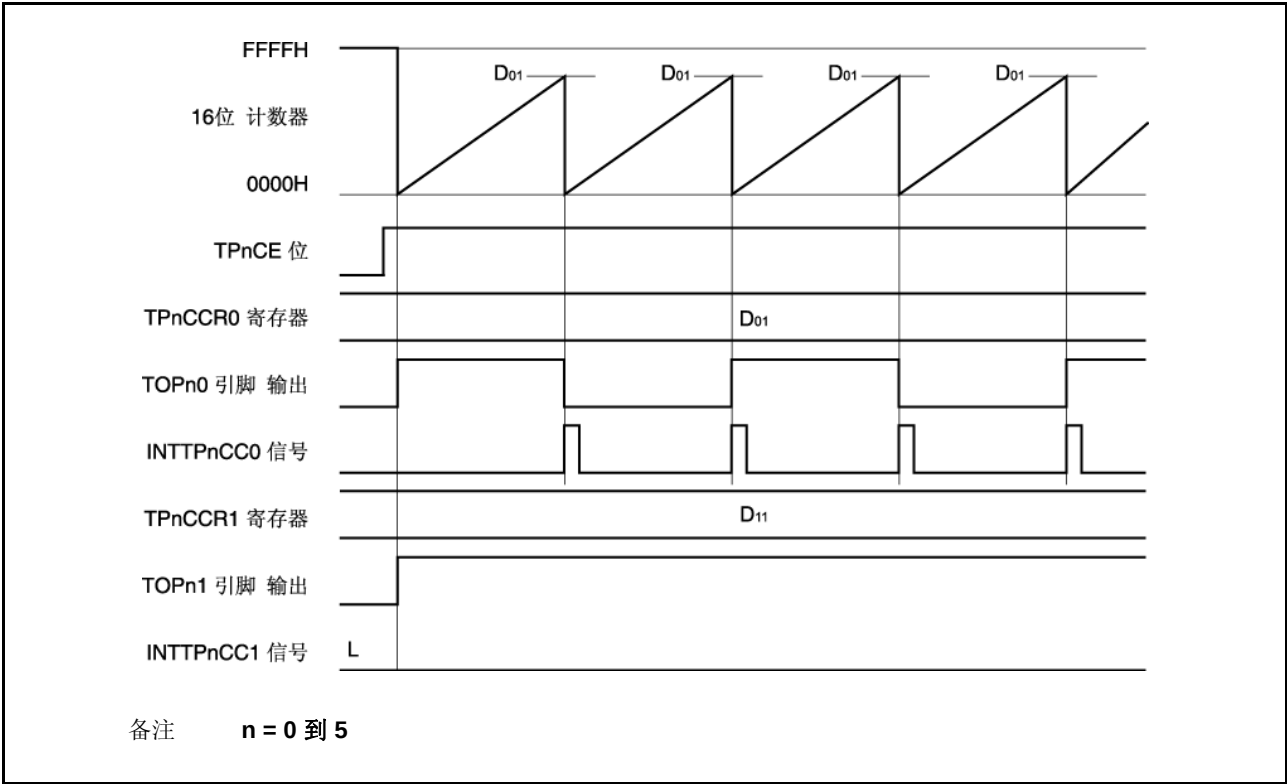
图 7-11. $D_{01} \geq D_{11}$ 时的时序图



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

<R> 当未使用 TPnCCR1 寄存器时，推荐将它的值设为 FFFFH。

图 7-12. D₀₁ < D₁₁ 时的时序图



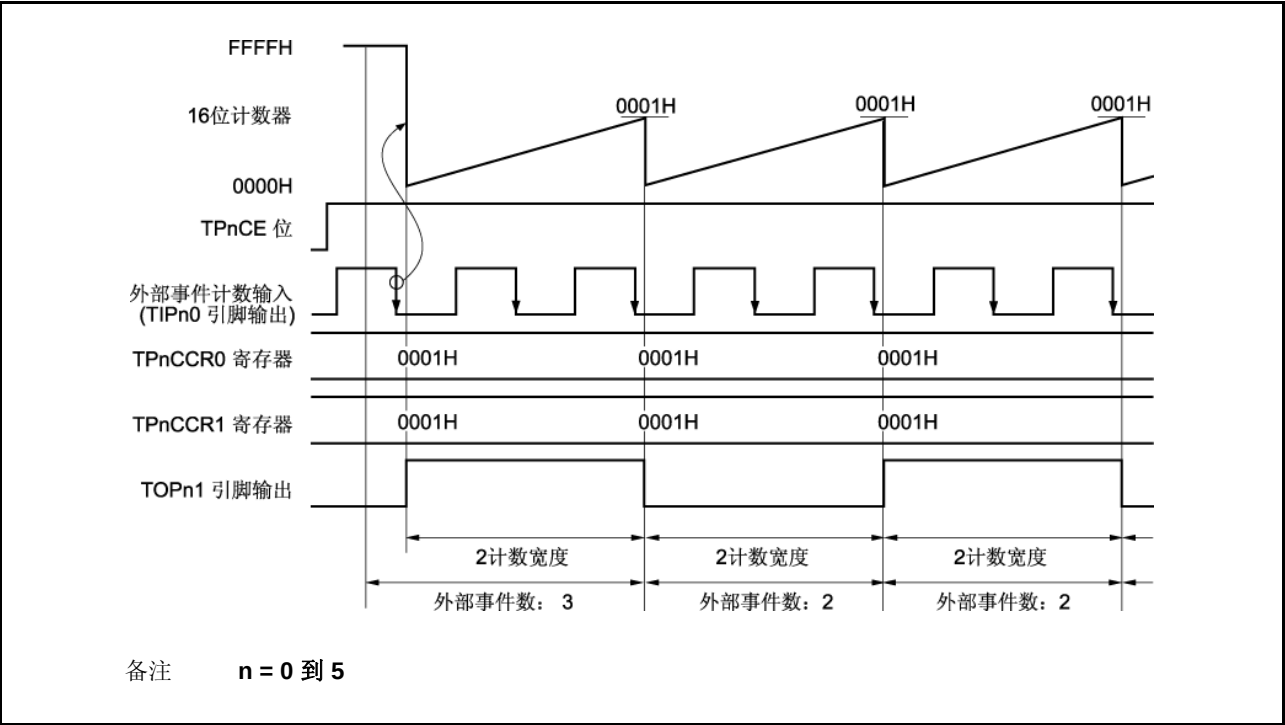
<R> (3) 外部事件计数输入 (TIPn0) 操作

(a) 操作

间隔定时器模式下为了在外部事件计数输入 (TIPn0) 的有效沿开始计数, 外部事件计数输入的有效沿是必需的, 因为 TPnCE 位从 0 置为 1 后 16 位计数器从 FFFFH 清零。

当 TPnCCR0 和 TPnCCR1 寄存器设置为 0001H 时, 每次 16 位计数器计数加倍时 TOPn1 管脚输出反转。

间隔定时器模式下, 仅当定时器输出 (TOPn1) 用于外部事件计数输入时, TPnCTL1.TPnEEE 位置 1。



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

外部事件计数模式下，当 TPnCTL0.TPnCE 位=1 时，计数外部事件计数输入 (TIPn0) 的有效沿。每次计数到 <R> TPnCCR0 寄存器设置的值时产生中断请求信号 (INTTPnCC0)。该模式下不可使用 TOPn0 和 TOPn1 引脚。当 TOPn1 引脚用于外部事件计数输入时，在间隔定时器模式下，设置 TPnCTL1.TPnEEE 位为 1 (参阅 7.6.1 (3) 外部事件计数输入 (TIPn0) 操作)。

外部事件计数模式中不使用 TPnCCR1 寄存器。

<R> 注意事项 外部事件计数模式下，TPnCCR0 和 TPnCCR1 寄存器不能清零。

图 7-13. 外部事件计数模式的配置图

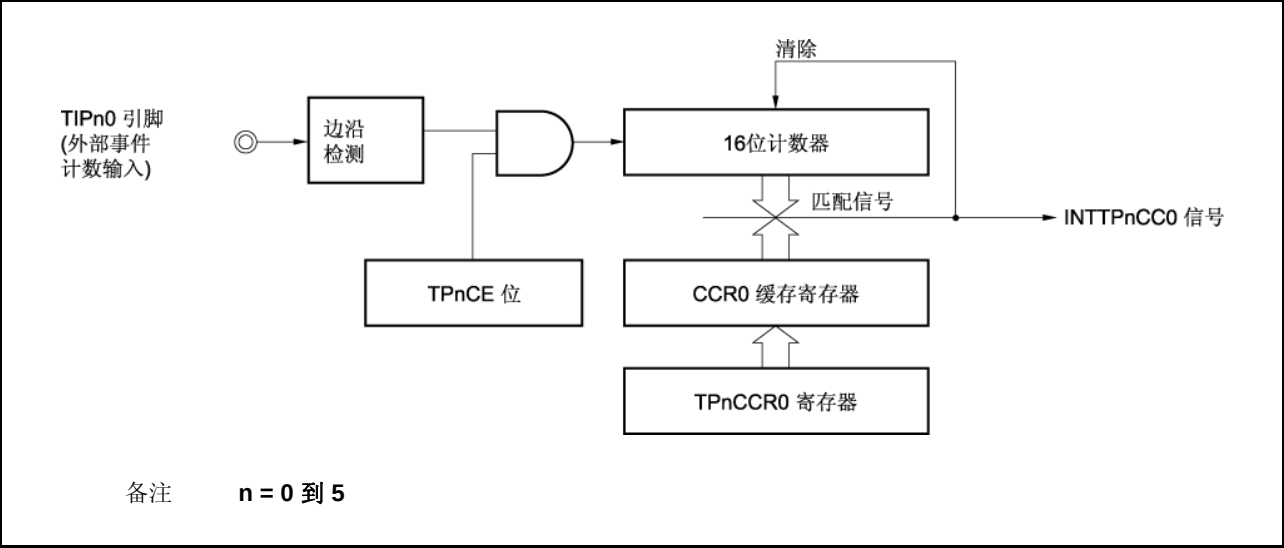
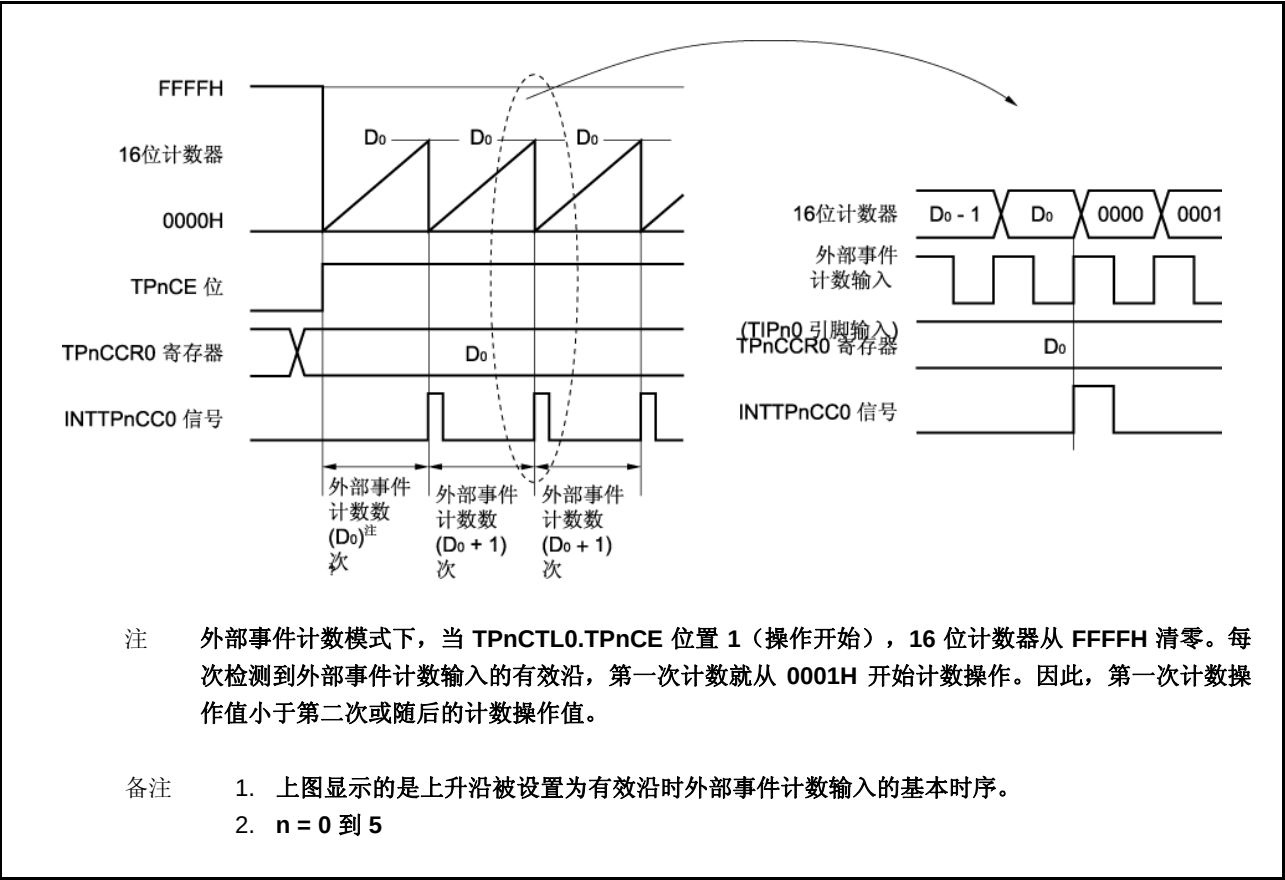


图 7-14. 外部事件计数模式的基本时序



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

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

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

图 7-15. 外部事件计数模式操作的寄存器设置（1/2）

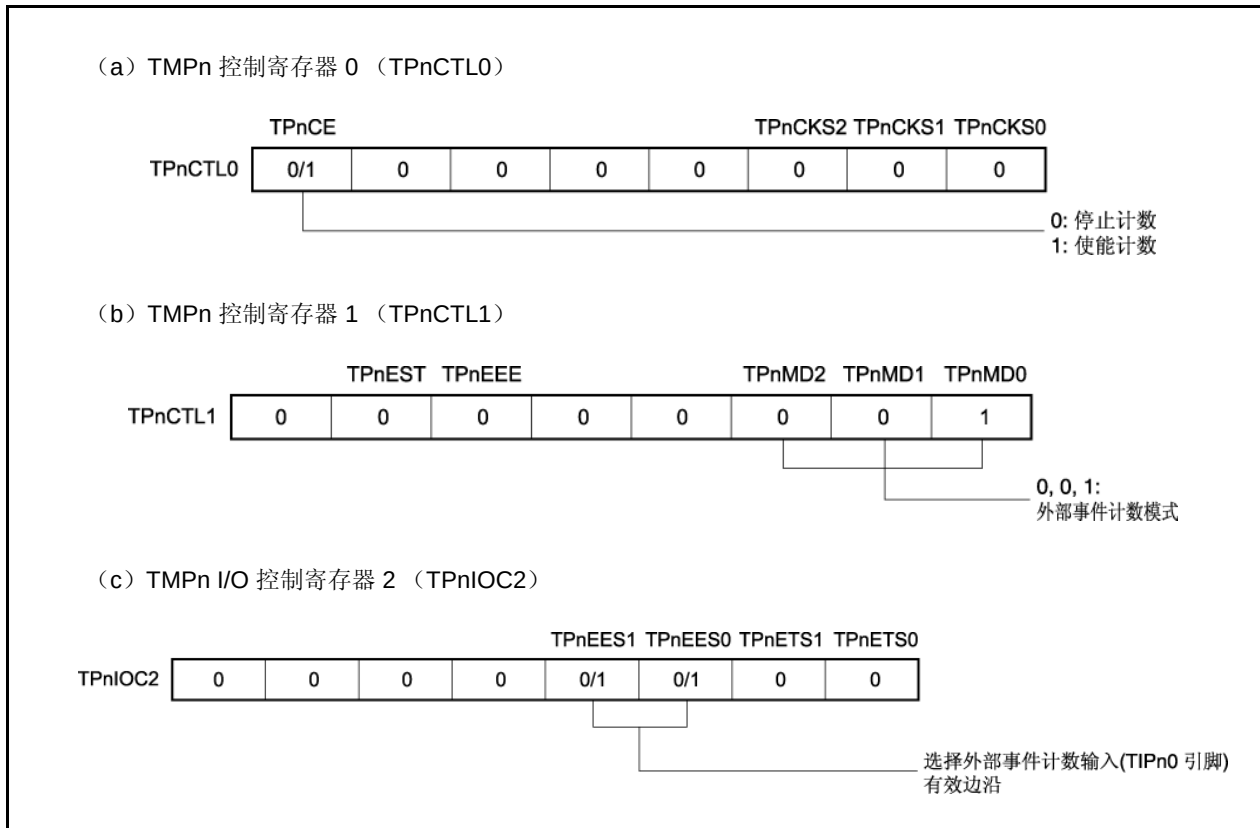


图 7-15. 外部事件计数模式操作的寄存器设置 (2/2)

(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。

<R> (e) TMPn 捕捉 / 比较寄存器 0 (TPnCCR0)
如果 TPnCCR0 寄存器设为 D₀, 当外部时间达到 (D₀) 次, 计数清零, 产生第一个比较匹配中断请求信号 (INTTPnCC0)。当外部事件数达到 (D₀+1) 次时, 产生第二次比较匹配中断请求信号 (INTTPnCC0)。

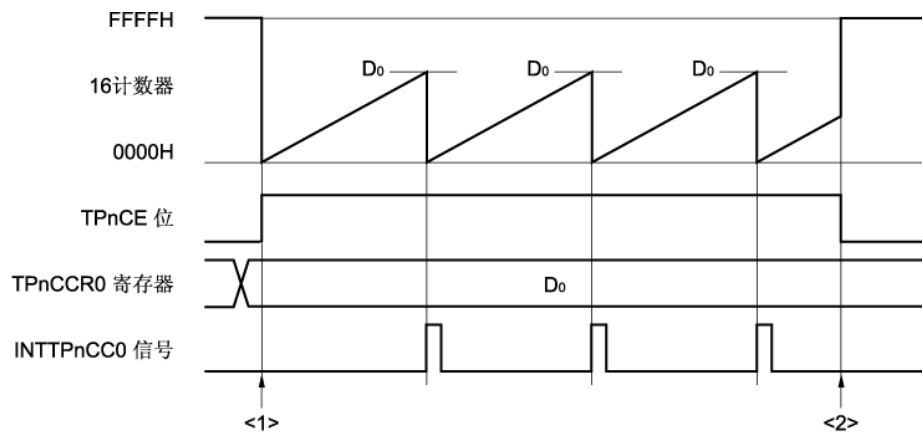
<R> (f) TMPn 捕捉 / 比较寄存器 1 (TPnCCR1)
外部事件计数模式中不使用 TPnCCR1 寄存器。但对 TPnCCR1 寄存器设置的数值也将被传送到 CCR1 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 缓冲寄存器中的数值匹配时, 也会产生一个比较匹配中断请求信号 (INTTPnCC1)。
当未使用 TPnCCR1 寄存器时, 推荐将它的值设为 FFFFH。也通过中断屏蔽标志 (TPnCCIC1.TPnCCMK1) 屏蔽此寄存器。

<R> 注意事项 1. 设置 TPnIOC0 寄存器为 00H。
2. 当使用外部时钟作为计数时钟时, 外部时钟只能从 TIPn0 管脚输入。这时, 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)。

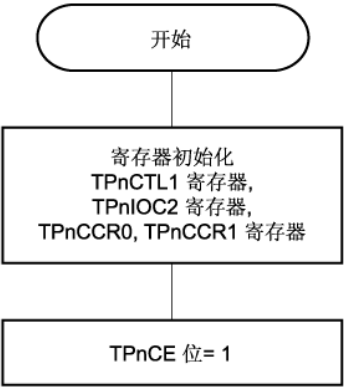
备注 1. 外部事件计数模式下, 并未使用 TMPn I/O 控制寄存器 1 (TPnIOC1) 和 TMPn 选项寄存器 0 (TPnOPT0)。
2. n = 0 到 5

(1) 外部事件计数模式操作流程

图 7-16. 外部事件计数模式的软件处理流程



<1> 计数操作启动流程



在TPnCE 位置1前
初始化这些寄存器。

<2> 计数操作停止流程



通过TPnCE位清零来
初始化计数器和停止计数。

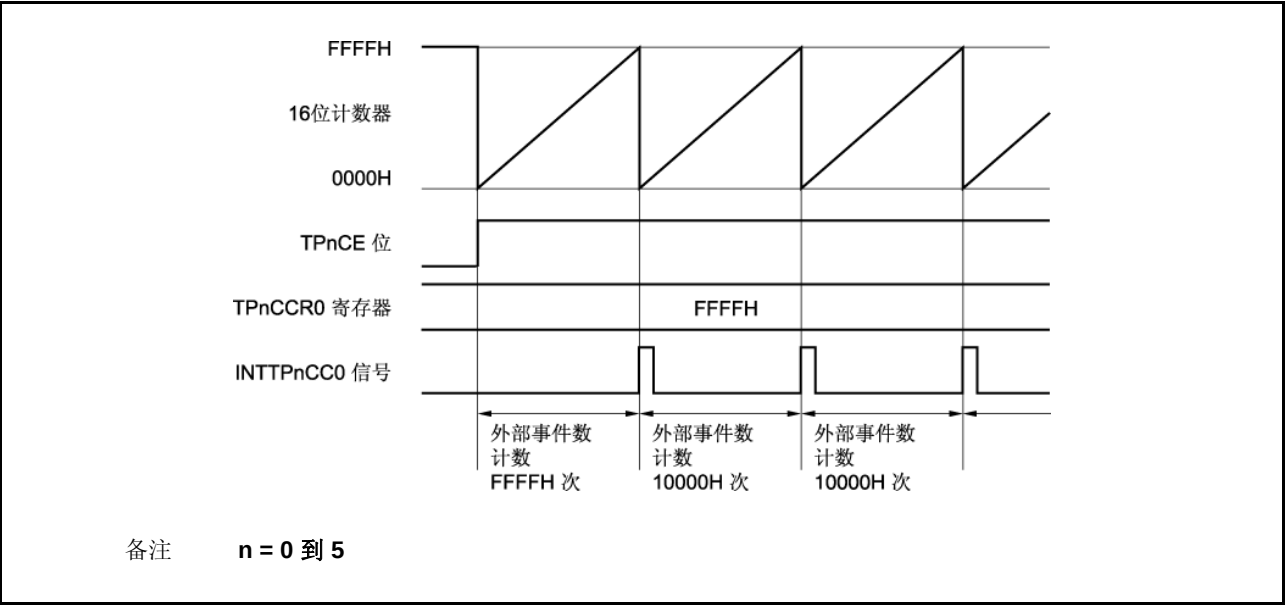
备注 **n = 0 到 5**

(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 和 TPnCCR1 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（TOPn0，TOPn1）功能。如果用外部事件计数输入（TIPn0）作为定时器输出（TOPn1），设置间隔定时器模式，通过外部事件输入作为计数时钟设置操作来使能操作（TPnCTL.TPnEEE 位=1）（参见 7.6.1（3）外部事件计数输入操作（TIPn0））。

(a) TPnCCR0 寄存器被设置为 FFFFH 时的操作

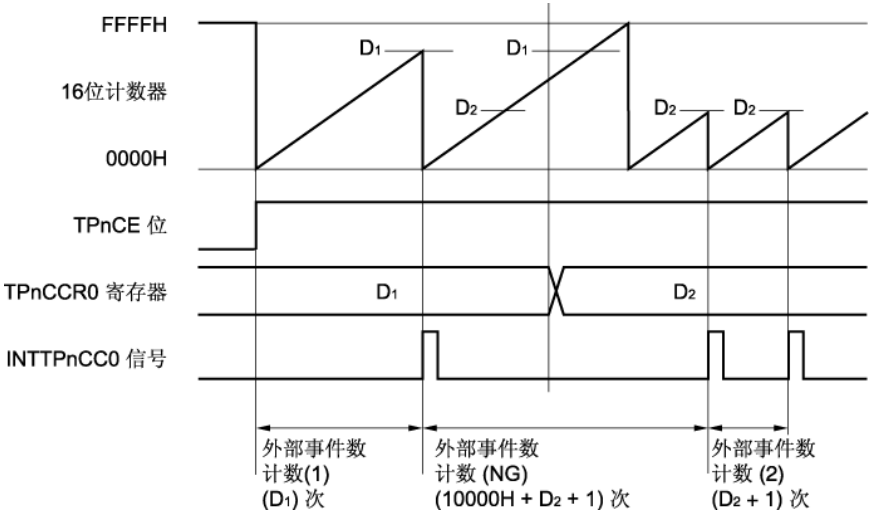
如果 TPnCCR0 寄存器被设置为 FFFFH，那么 16 位计数器将在每次检测到外部事件计数信号的有效沿处计数，直到 FFFFH。并在下一个计数事件时刻被同步清零，同时产生 INTTPnCC0 信号。此时，溢出标志位（TPnOPT0.TPnOVF 位）不会被置 1。



(b) 改写 TPnCCR0 寄存器时需要注意的事项

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。如果可能溢出，停止计数并改变设置值。

<R>

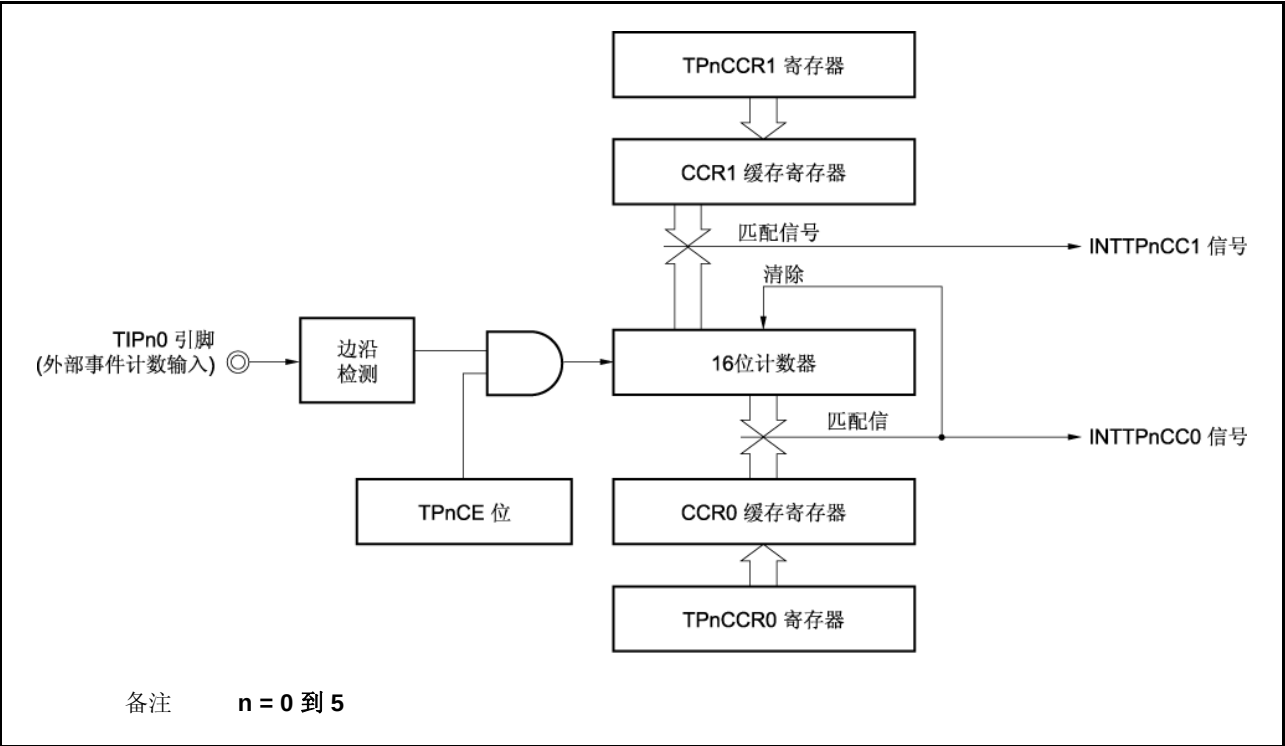


备注 n = 0 到 5

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

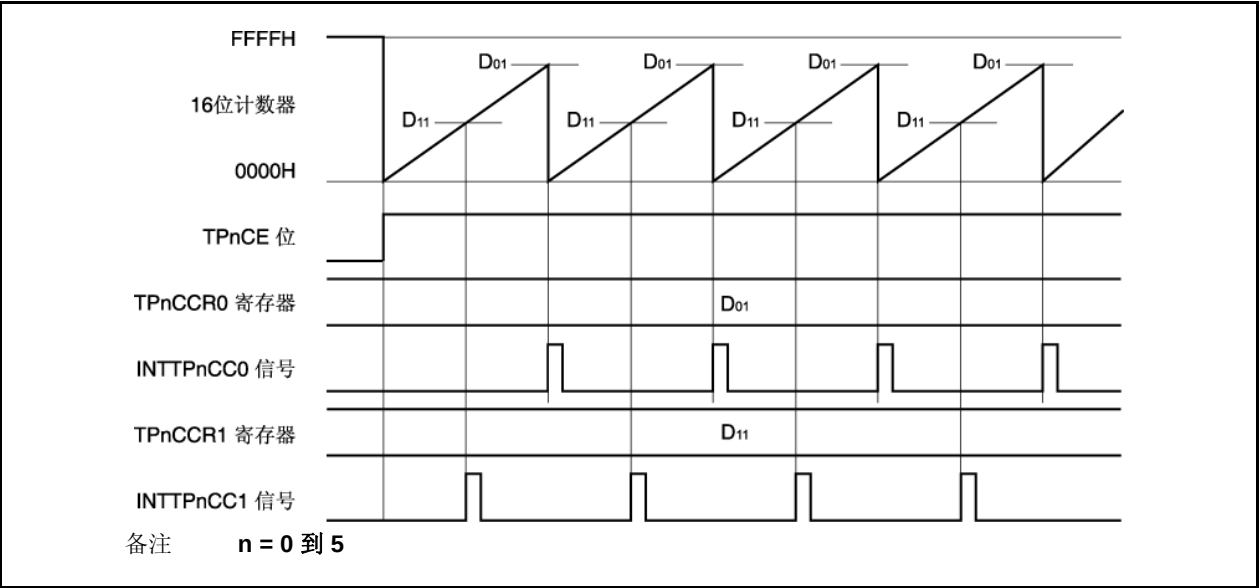
(c) TPnCCR1 寄存器的操作

图 7-17. TPnCCR1 寄存器的配置



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

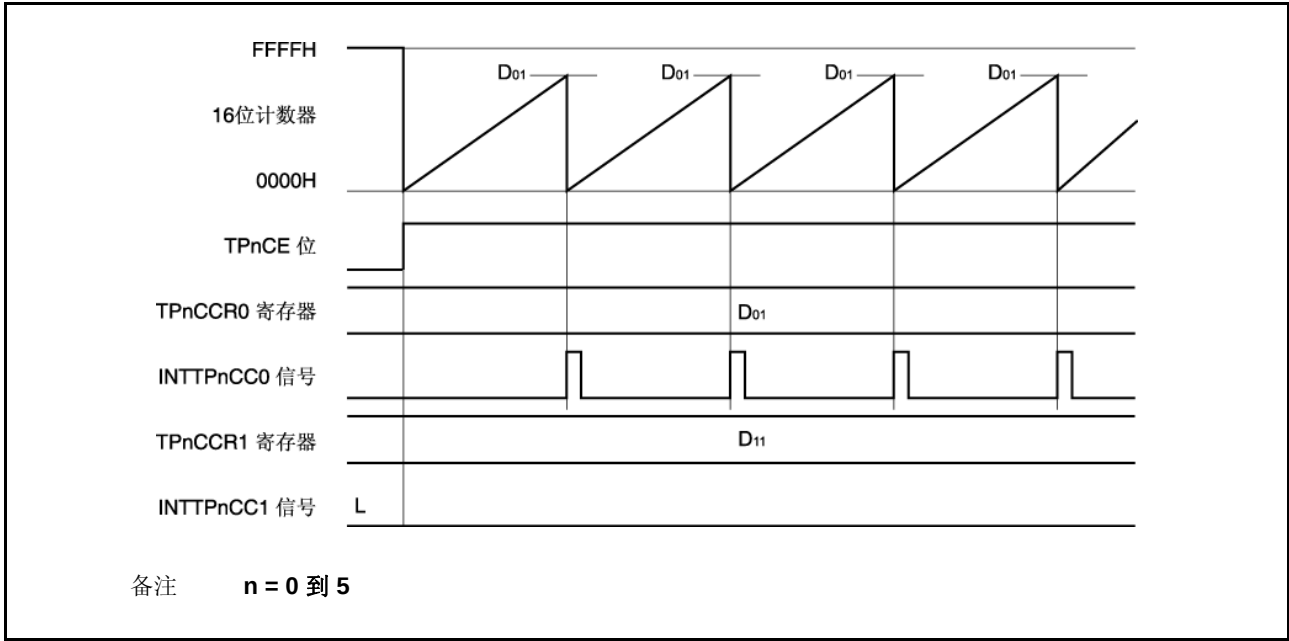
图 7-18. $D_{01} \geq D_{11}$ 时的时序图



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。

<R> 当 TPnCCR1 寄存器未使用时推荐设置 TPnCCR1 寄存器为 FFFFH。

图 7-19. D01 < D11 时的时序图



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

在外部触发脉冲输出模式中，TPnCTL0.TPnCE 位被置 1 后，16 位定时器 / 事件计数器 P 进入等待触发信号的状态。当检测到一个外部触发输入信号 (TIPn0) 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。使用软件触发器时，半周期为 (TPnCCR0 寄存器设置值 +1)、占空比为 50% 的方波也从 TOPn0 引脚输出。

图 7-20. 外部触发脉冲输出模式的配置

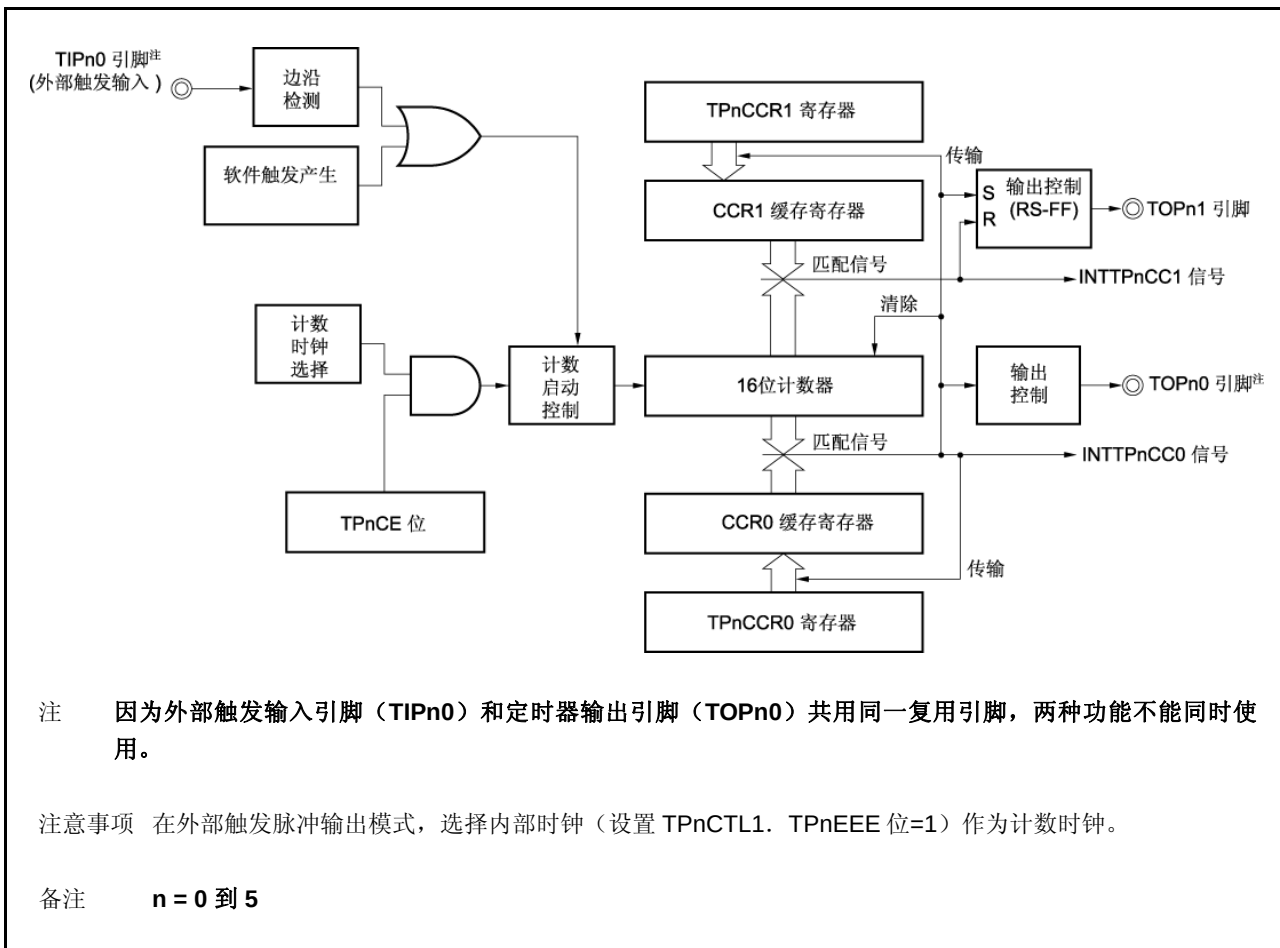
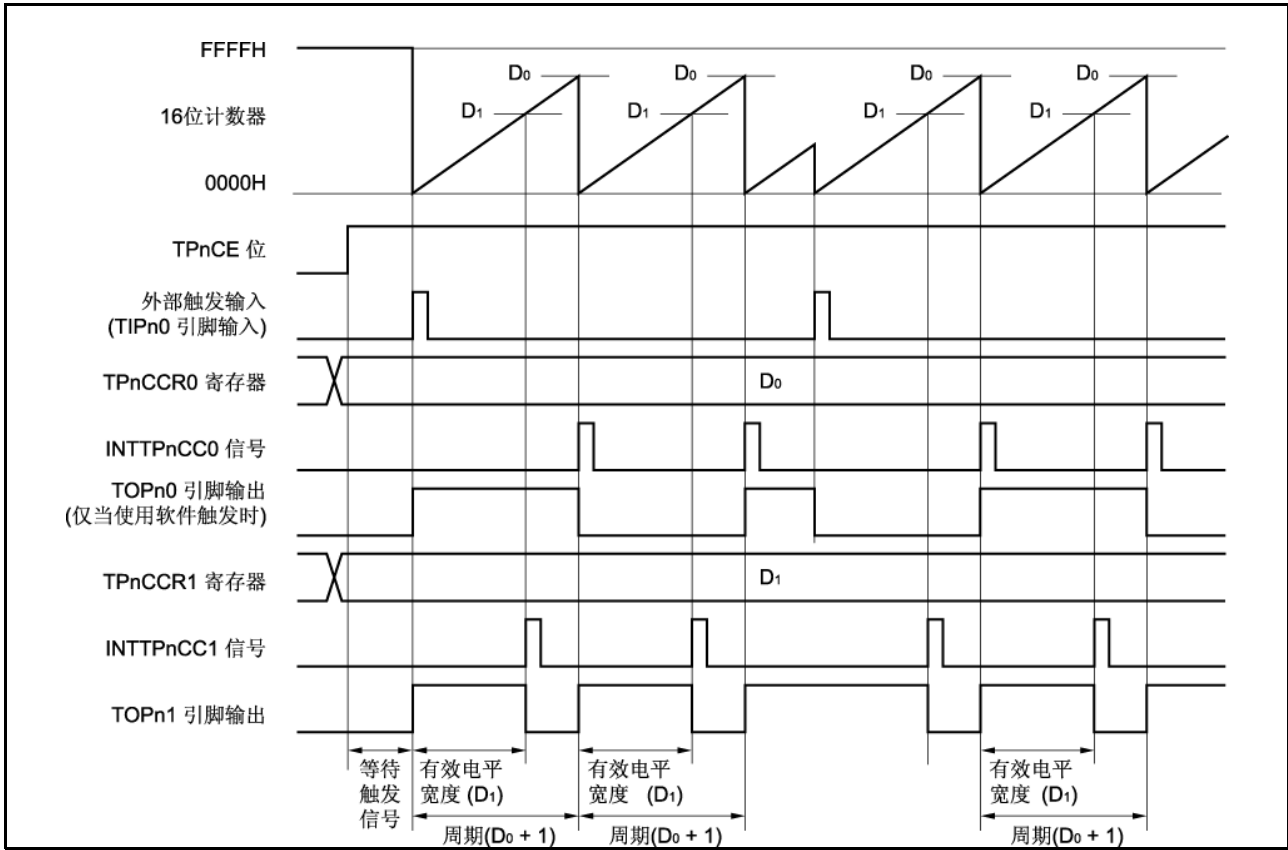


图 7-21. 外部触发脉冲输出模式的基本时序



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 (TIPn0) 信号的有效沿和设置软件触发 (TPnCTL1.TPnEST 位 = 1) 的方式中进行选择。

备注 n = 0 到 5, m = 0, 1

图 7-22. 外部触发脉冲输出模式的寄存器设置 (1/2)

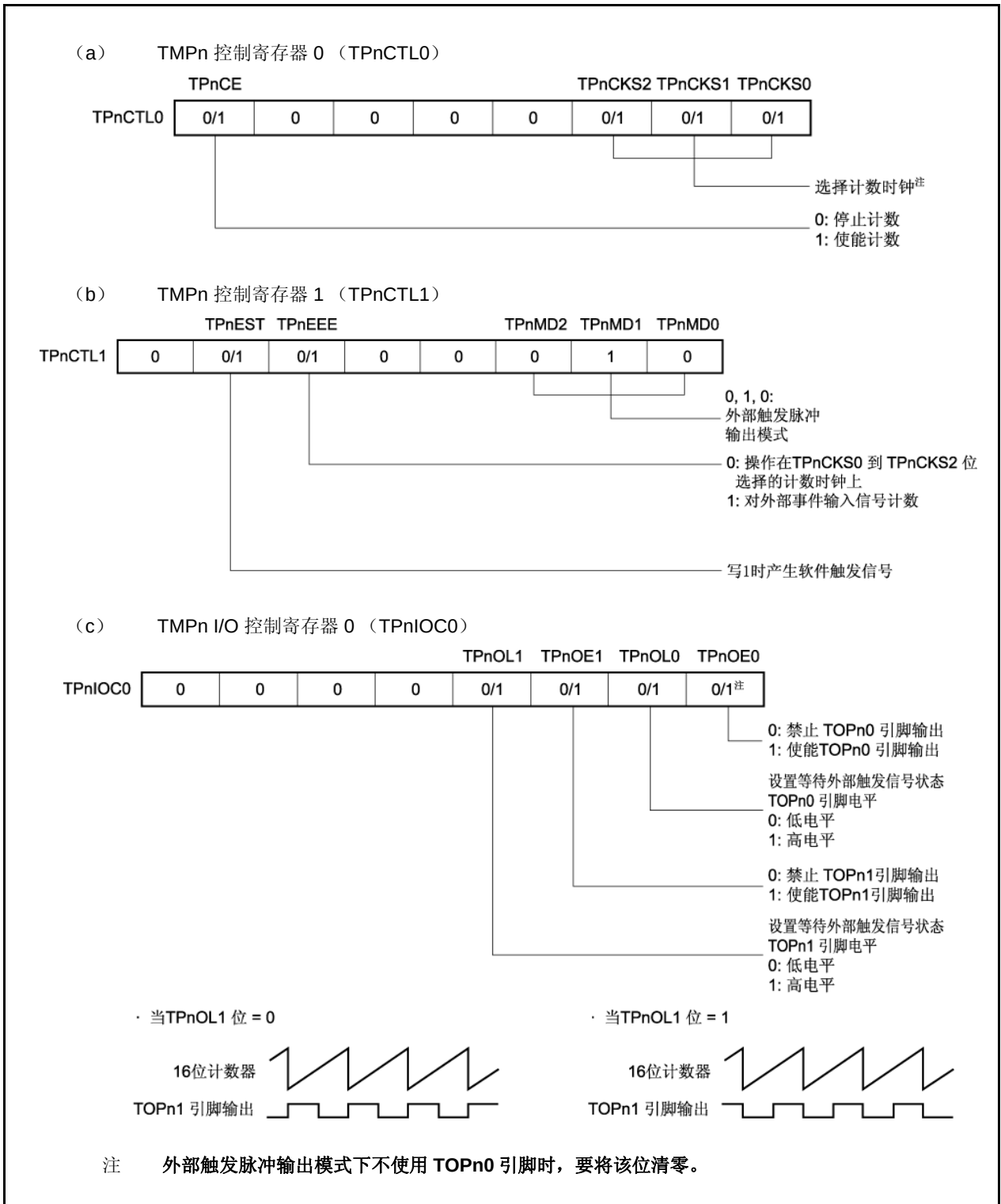
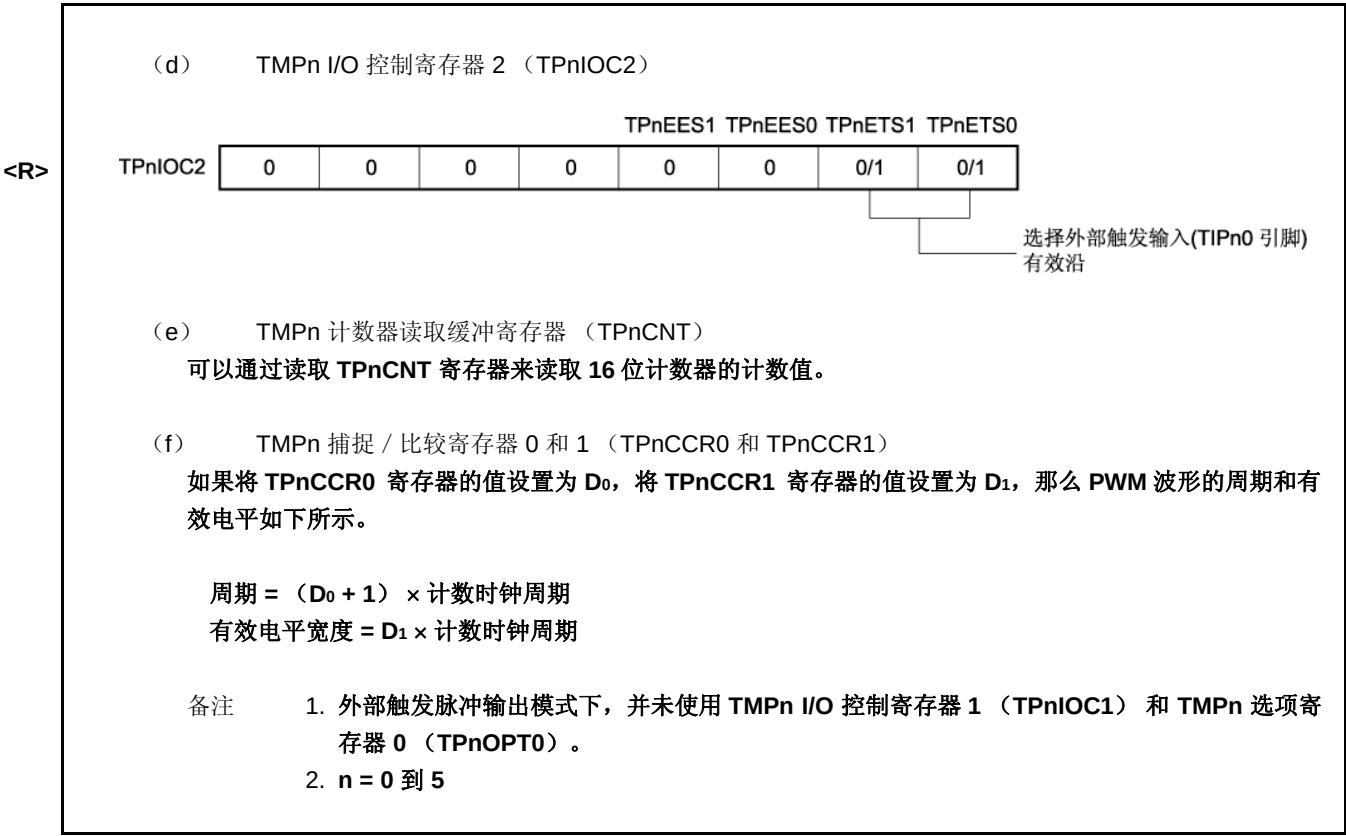


图 7-22. 外部触发脉冲输出模式的寄存器设置 (2/2)



(1) 外部触发脉冲输出模式的操作系统

图 7-23. 外部触发脉冲输出模式的软件处理流程 (1/2)

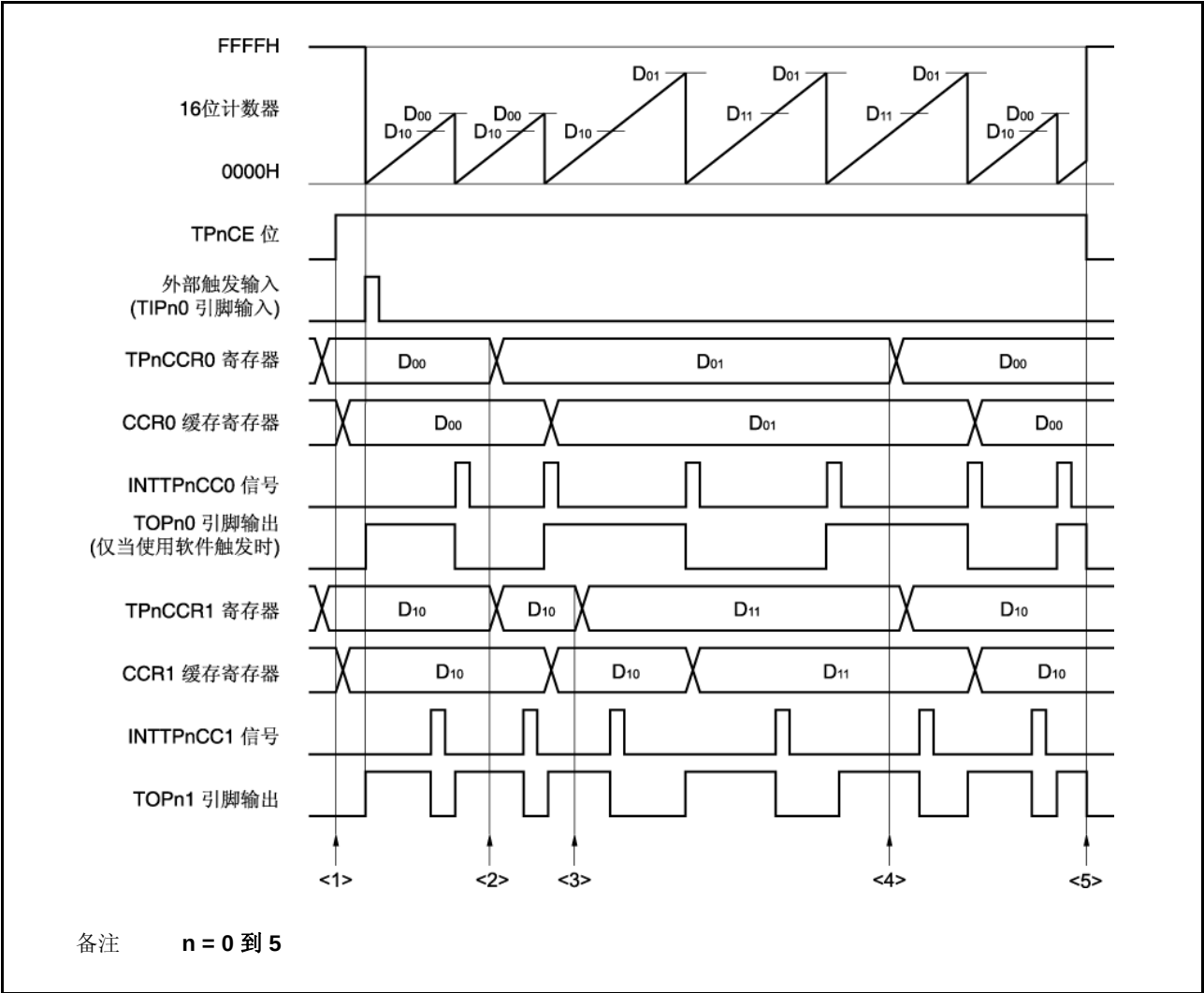
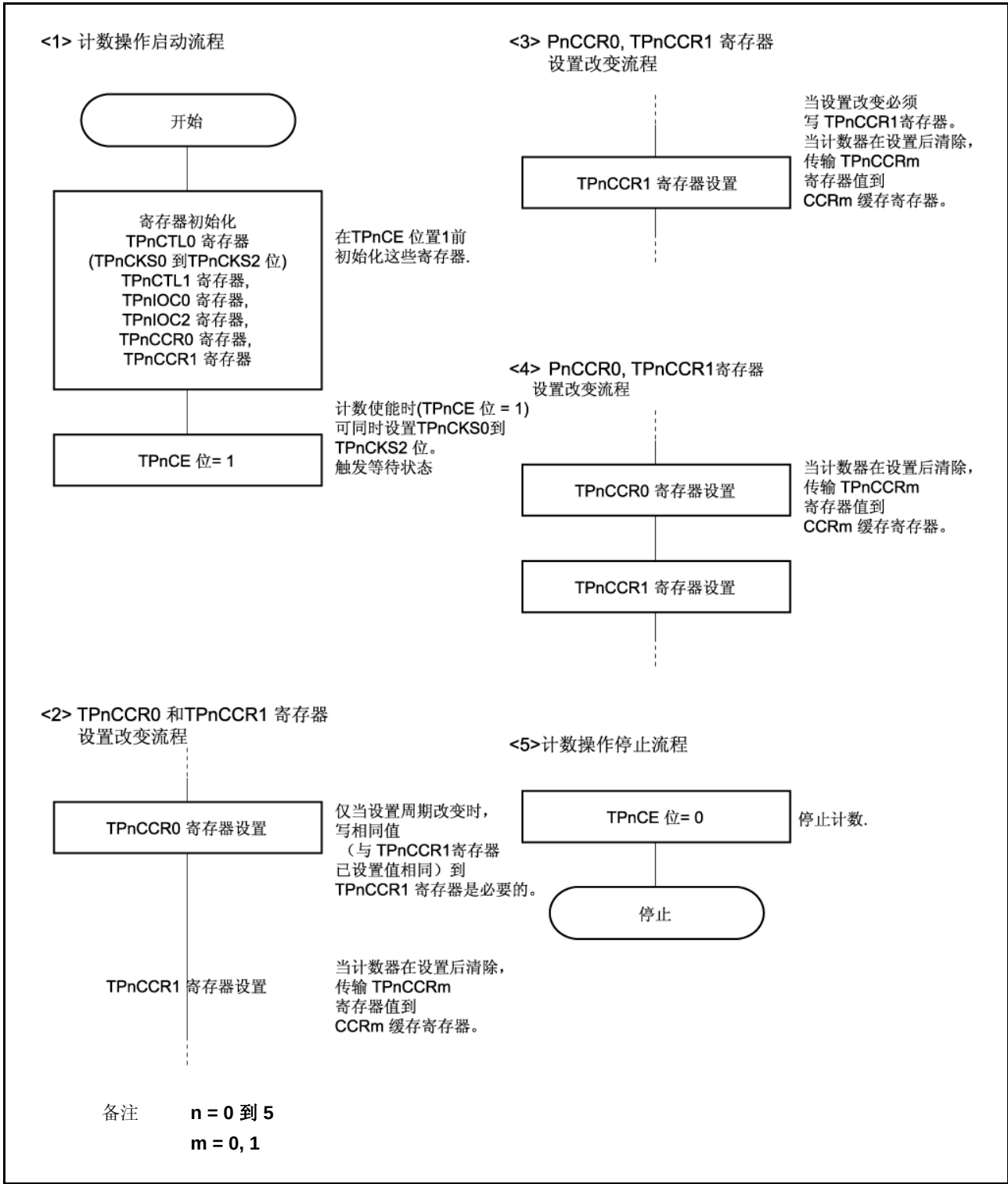


图 7-23. 外部触发脉冲输出模式的软件处理流程 (2/2)

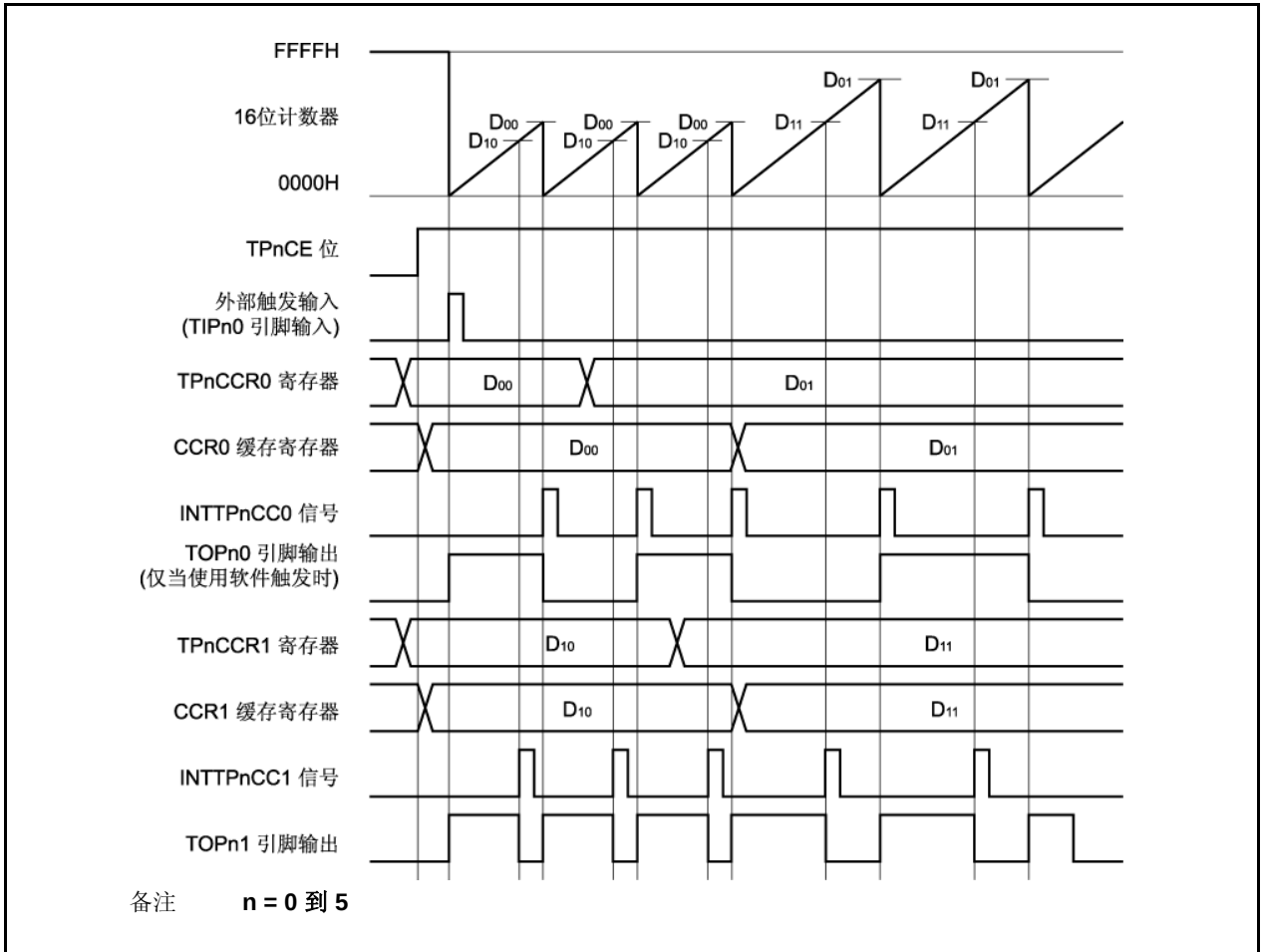


(2) 外部触发脉冲输出模式的操作时序

(a) 操作过程中改变脉冲宽度时需要注意的事项

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

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



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

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

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

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

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

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

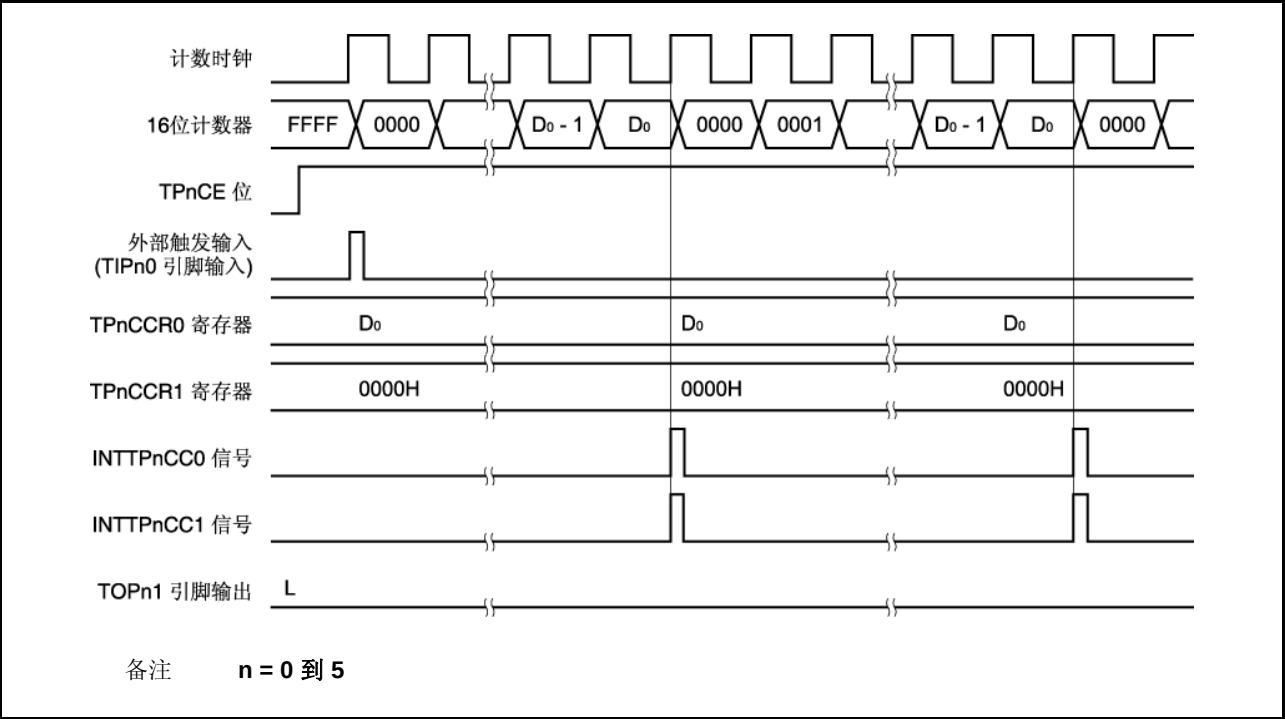
备注 n = 0 到 5
 m = 0, 1

(b) PWM 波形的 0%/100% 输出

<R>

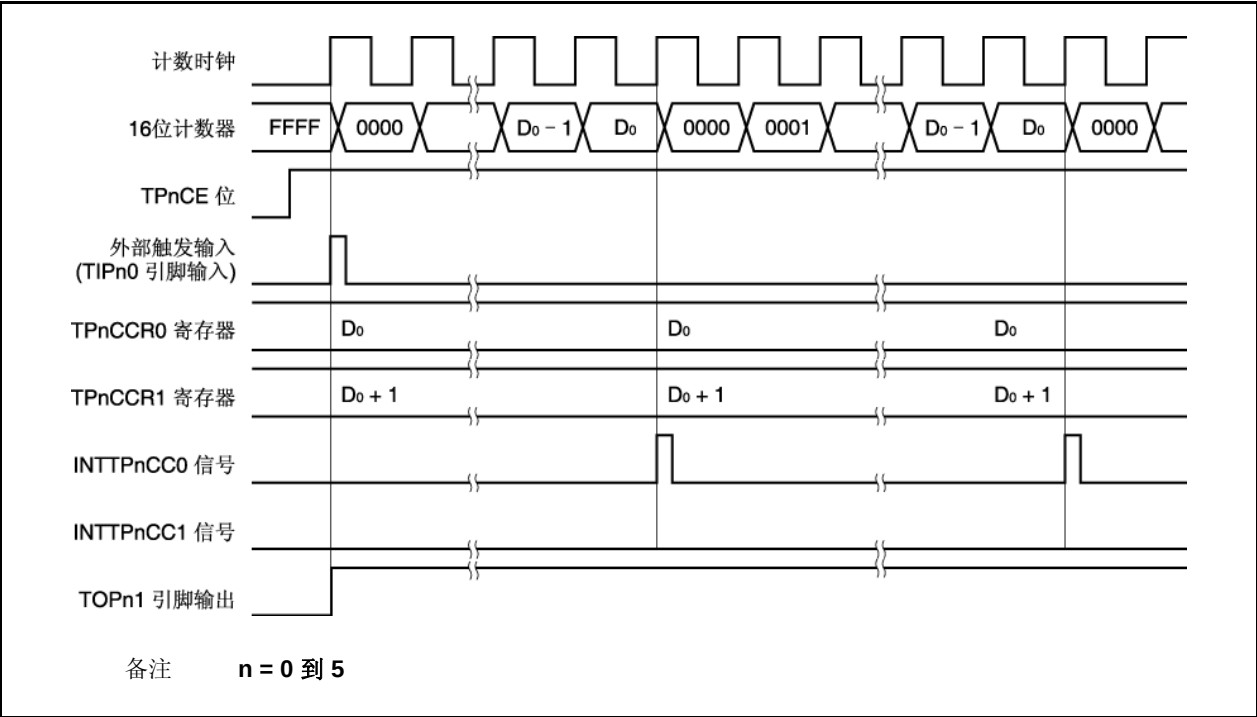
若要输出 0% 的波形，则需将 TPnCCR1 寄存器设置为 0000H。16 位计数器的值与 CCR0 缓存寄存器的值匹配后的下一时刻 16 位计数器被清零，并产生 INTTPnCO0 和 INTTPnCC1 信号。

<R>



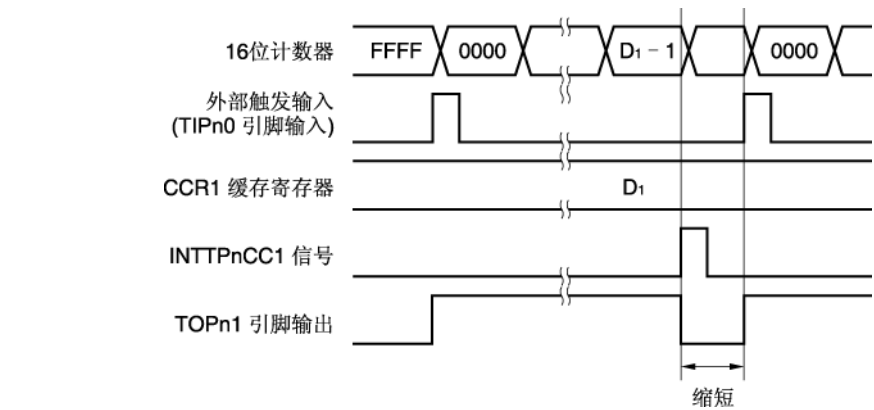
若要输出 100% 波形，则需将 TPnCCR1 寄存器设置为 (TPnCCR0 寄存器值 + 1)。TPnCCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。

<R>



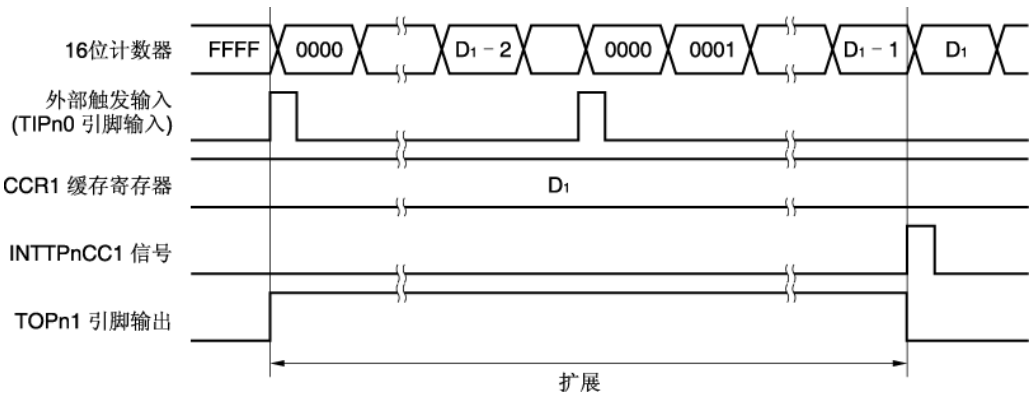
(c) 触发检测和 CCR1 寄存器匹配之间的冲突

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。



备注 n = 0 到 5

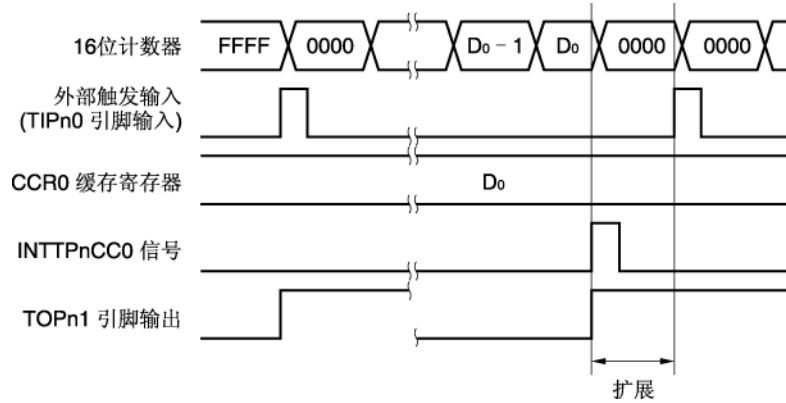
如果在 INTTPnCC1 信号产生之前检测到触发信号，那么 INTTPnCC1 信号将不会产生，此时，16 位计数器被清零并重新开始计数。TOPn1 引脚的输出也会继续保持为有效电平。这种情况下，PWM 波形的有效电平时间就会被延长。



备注 n = 0 到 5

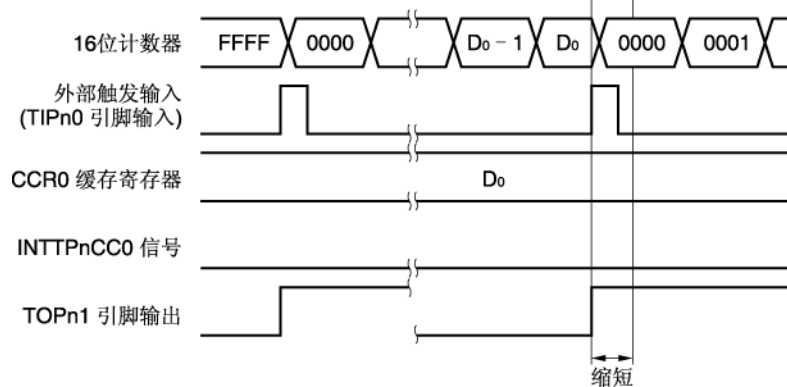
(d) 触发检测和 CCR0 寄存器匹配之间的冲突

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 到 5

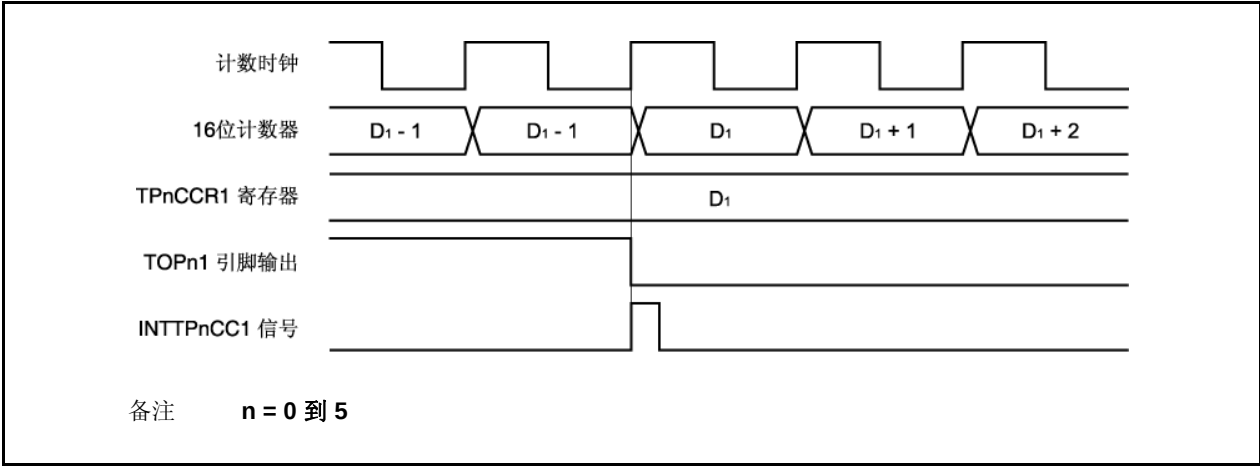
如果在 INTTPnCC0 信号产生之前检测到触发信号，那么 INTTPnCC0 信号将不会产生。16 位计数器将被清零并重新开始计数，TOPn1 引脚输出也会改变为有效电平。这种情况下，PWM 波形的非有效电平时间就会被缩短。



备注 n = 0 到 5

(e) 比较匹配中断请求信号的产生时序 (INTTPnCC1)

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



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

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

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

在单脉冲输出模式中，TPnCTL0.TPnCE 位被置 1 后，16 位定时器/事件计数器 P 进入等待触发信号的状态。当检测到一个外部触发输入信号（TIPn0）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 (TIPn0) 来开始脉冲的输出。当使用软件触发时, TOPn0 引脚会在 16 位计数器计数时输出有效电平, 在计数器停止计数时 (等待触发时) 输出非有效电平。

图 7-24. 单脉冲输出模式的配置图

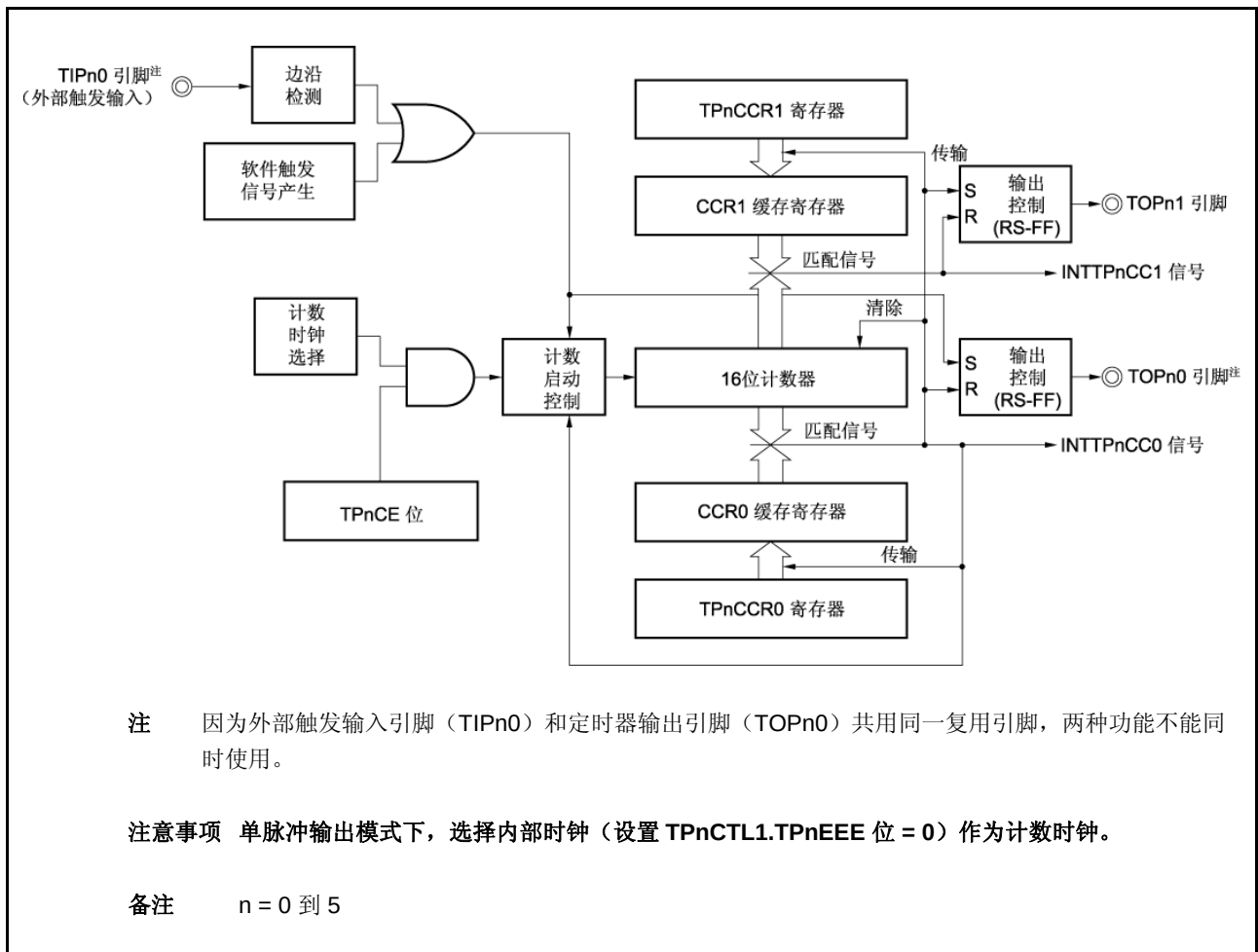
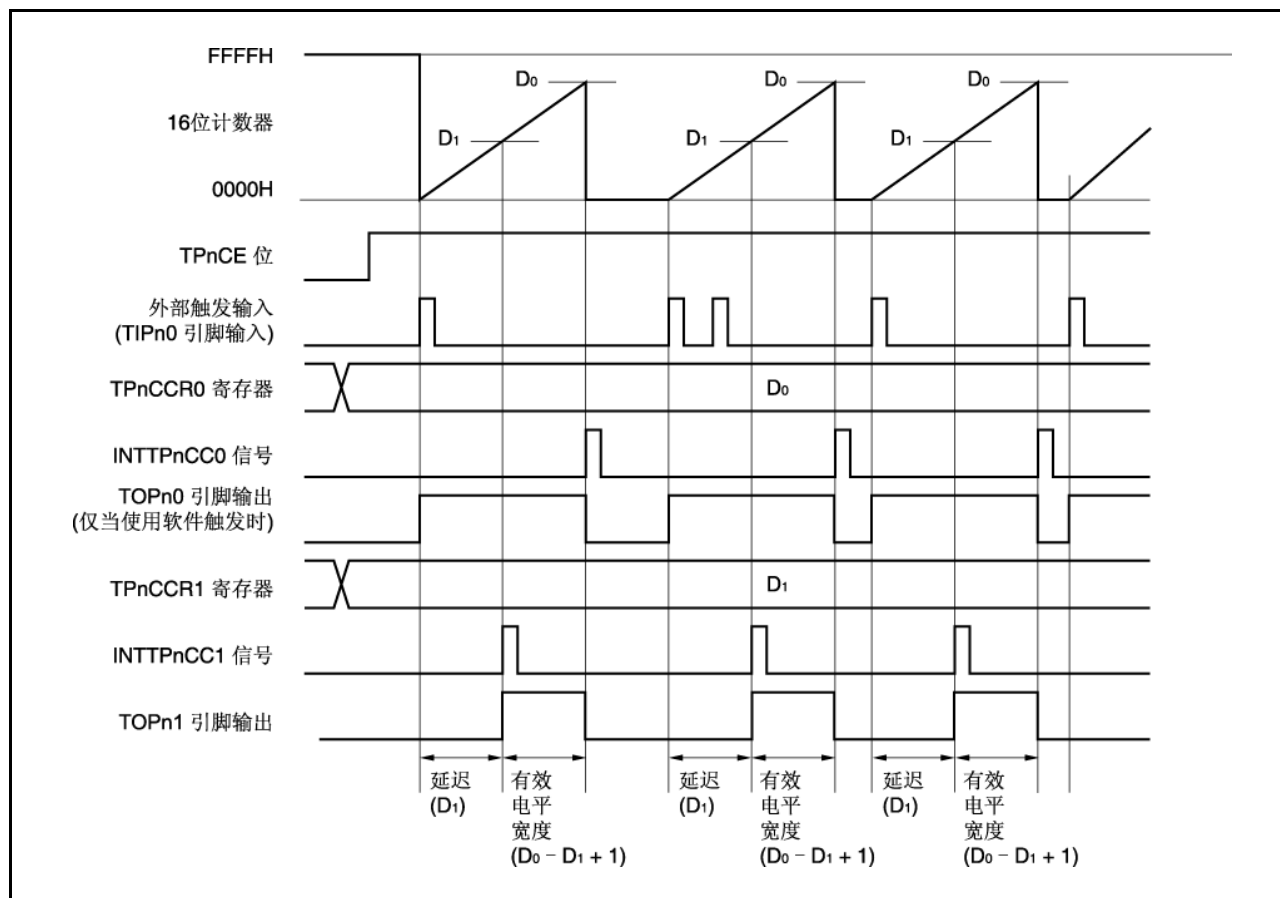


图 7-25. 单脉冲输出模式的基本时序



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 0000H 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。当触发信号再次产生时, 16 位计数器从 0000H 开始计数。如果在单脉冲波形输出过程中再次检测到触发信号, 那么这个触发信号将被忽略。

单脉冲波形的输出延迟时间以及有效电平宽度可由以下算式得出。

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

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

比较匹配中断请求信号 INTTPnCC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器中的值相匹配之后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（TIPn0 引脚）的有效沿和设置软件触发（TPnCTL1.TPnEST 位 = 1）的方式进行选择。

备注 $n = 0$ 到 5
 $m = 0, 1$

图 7-26. 单脉冲输出模式的寄存器设置 (1/2)

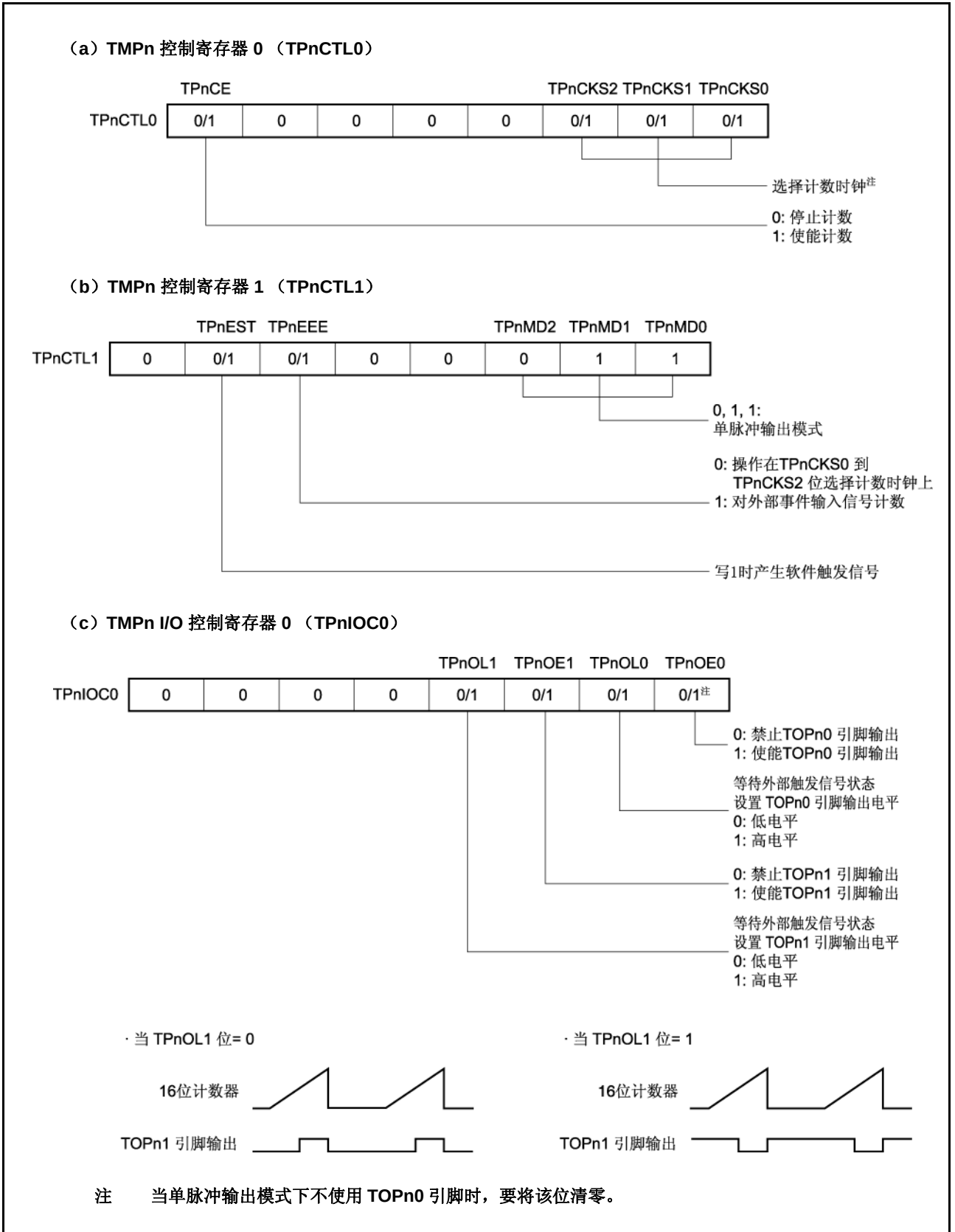
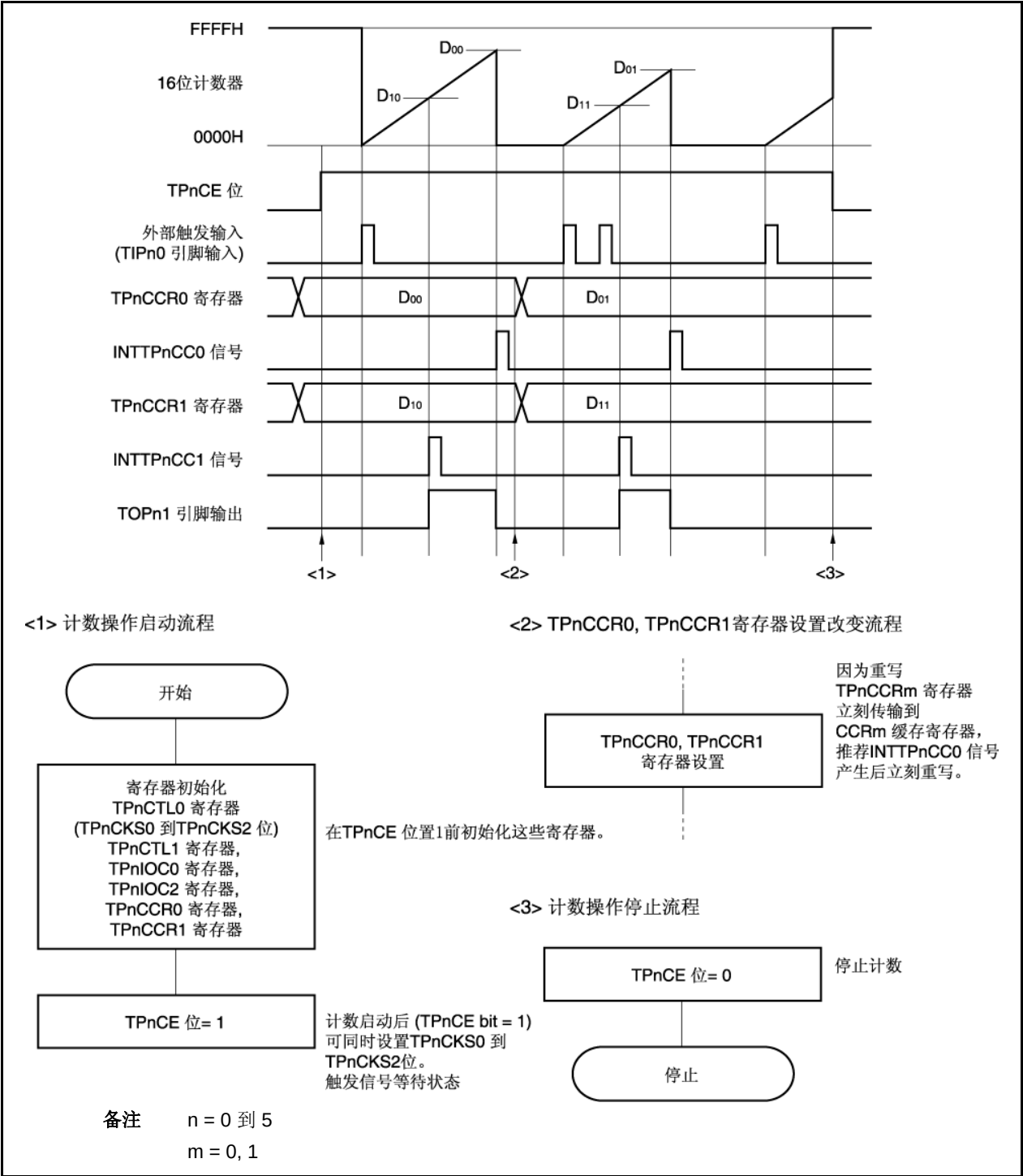


图 7-26. 单脉冲输出模式的寄存器设置 (2/2)



(1) 单脉冲输出模式的操作流程

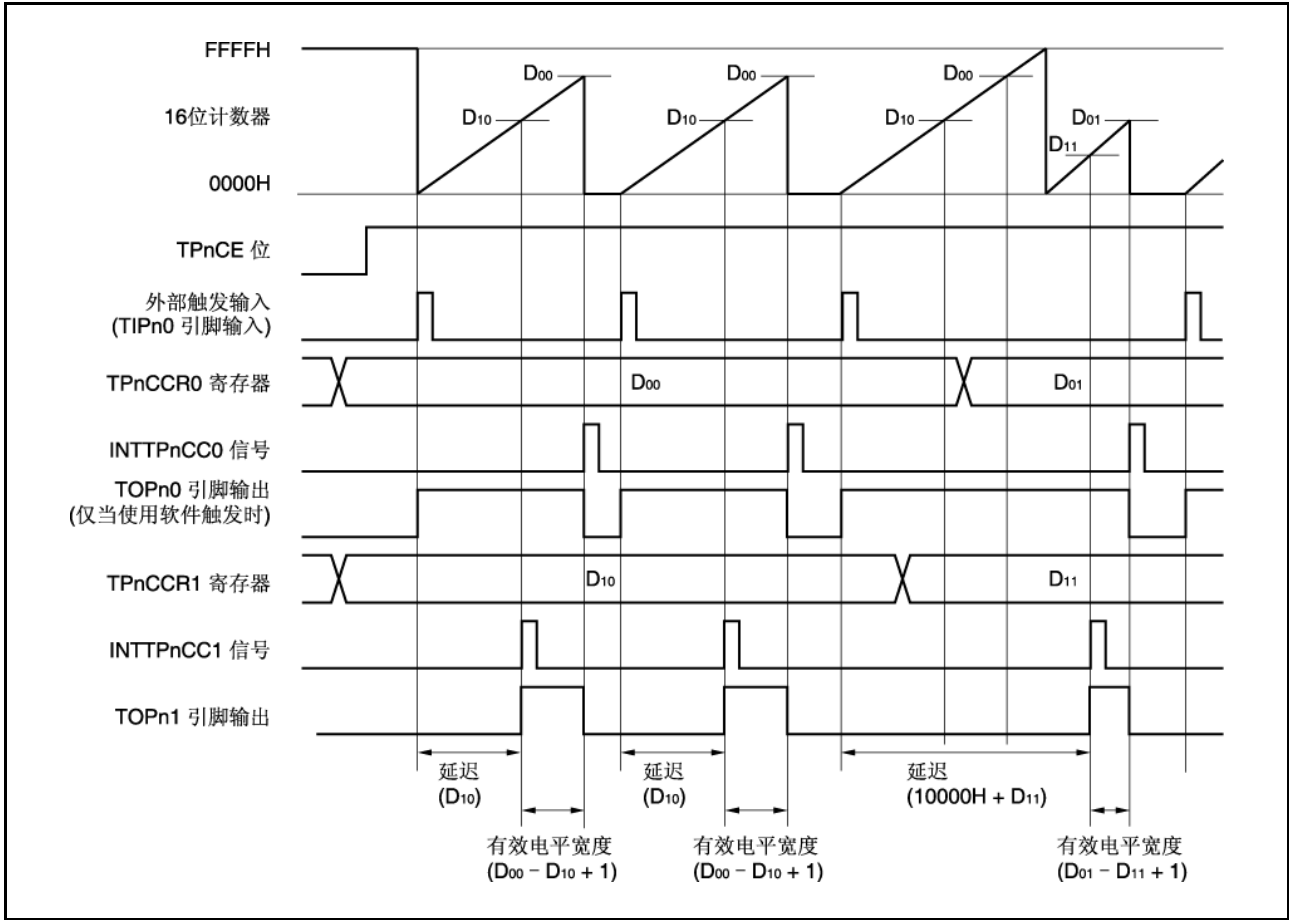
图 7-27. 单脉冲输出模式的软件操作流程



(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

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。如果可能溢出，停止计数并改变设置值。



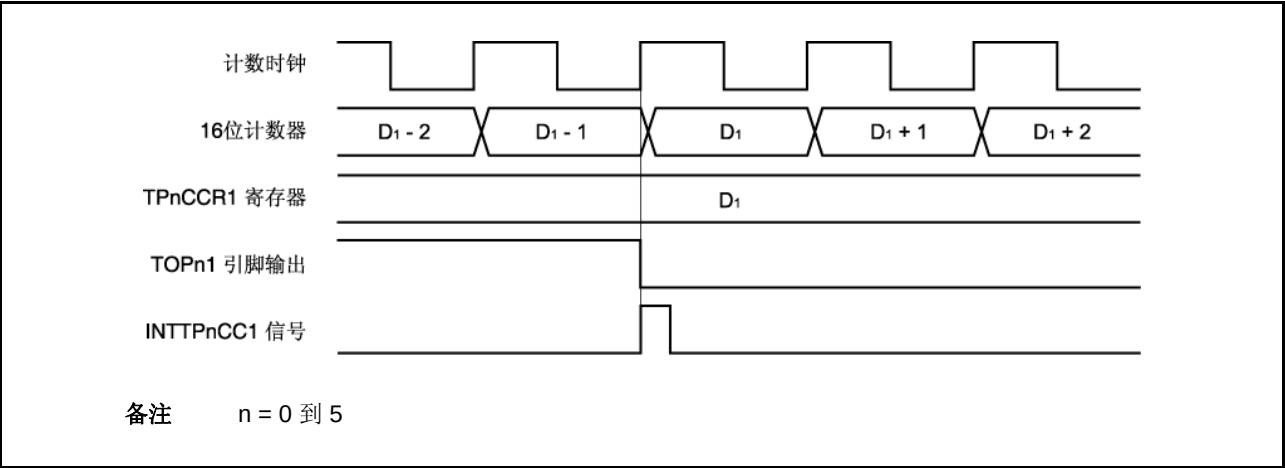
当将 TPnCCR0 寄存器的值从 D00 改写为 D01，TPnCCR1 寄存器的值从 D10 改写为 D11 时，这里假设 $D_{00} > D_{01}$ 且 $D_{10} > D_{11}$ ，如果 TPnCCR1 寄存器被改写时，16 位计数器的计数值大于 D11 且小于 D10，并且 TPnCCR0 寄存器被改写时，计数器的计数值大于 D01 且小于 D00，两个设定值都将在改写后立即生效并与计数值进行比较。此时计数器会计数到 FFFFH 然后从 0000H 重新开始计数。当计数值与 D11 匹配时，将产生 INTTPnCC1 信号，同时 TOPn1 引脚电平变为有效电平。当计数值与 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 $n = 0$ 到 5
 $m = 0, 1$

(b) 比较匹配中断请求信号的产生时序 (INTTPnCC1)

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



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

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

备注 $n = 0$ 到 5

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

PWM 输出模式下, TPnCTL0.TPnCE 位被置 1 后, TOPn1 引脚将输出一个 PWM 波形。
另外, TOPn0 引脚输出半周期为 (TPnCCR0 寄存器值+1)、占空比为 50%的方波。

<R>

图 7-28. PWM 输出模式的配置图

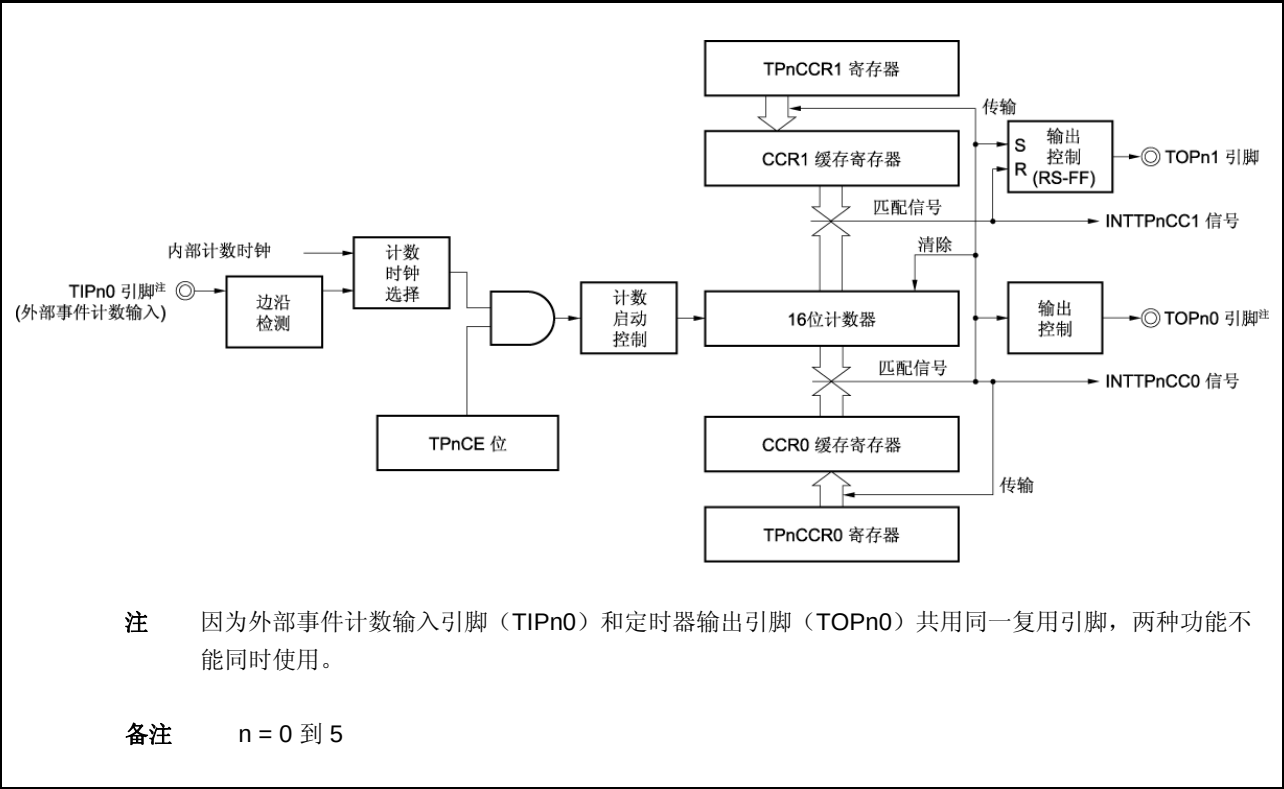
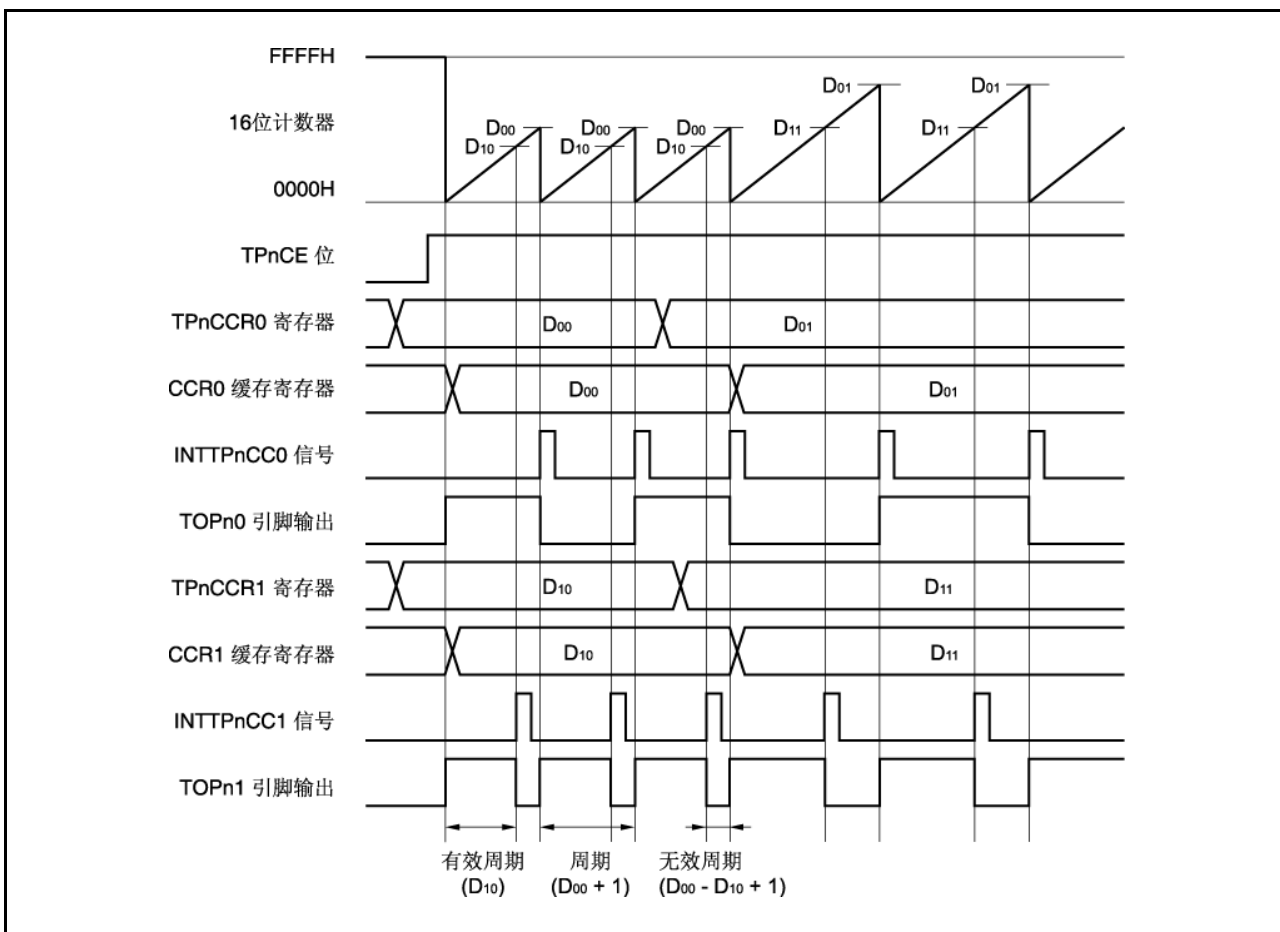


图 7-29. PWM 输出模式的基本时序



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)

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。

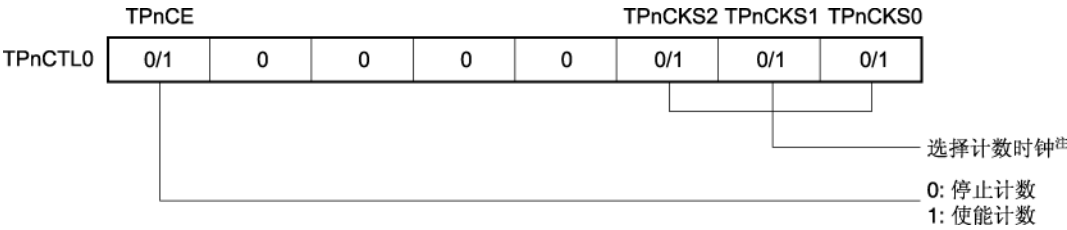
比较匹配中断请求信号 INTTPnCC0 会在 16 位计数器的计数值与 CCR0 缓冲寄存器的值相匹配后的下一个计数时钟时产生，同时 16 位计数器将被清零。比较匹配中断请求信号 INTTPnCC1 会在 16 位计数器的计数值与 CCR1 缓冲寄存器中的值相匹配时产生。

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。

备注 n = 0 到 5, m = 0, 1

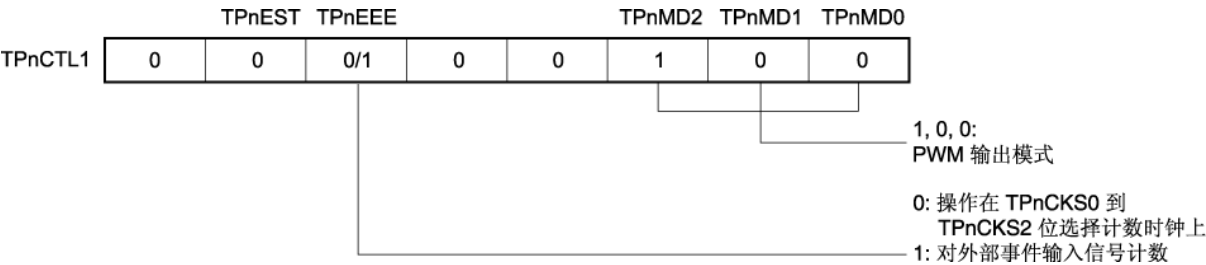
图 7-30. PWM 输出模式的寄存器设置 (1/2)

(a) TMPn 控制寄存器 0 (TPnCTL0)

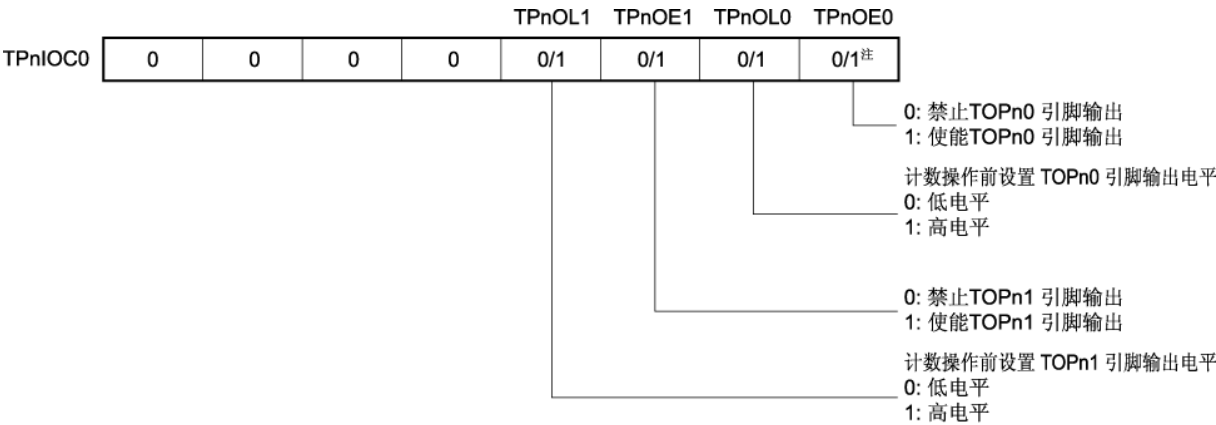


注 TPnCTL1.TPnEEE 位 = 1 时该设置无效。

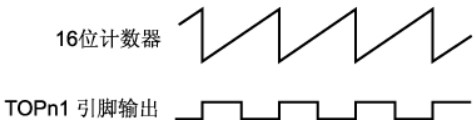
(b) TMPn 控制寄存器 1 (TPnCTL1)



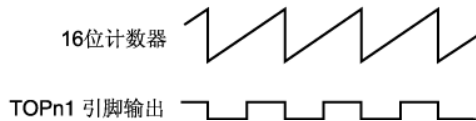
(c) TMPn I/O 控制寄存器 0 (TPnIOC0)



· 当 TPnOL1 位 = 0



· 当 TPnOL1 位 = 1



注 当 PWM 输出模式下不使用 TOPn0 引脚时，要将该位清零。

图 7-30. PWM 输出模式的寄存器设置 (2/2)



(1) PWM 输出模式的操作流程

图 7-31. PWM 输出模式的软件处理流程 (1/2)

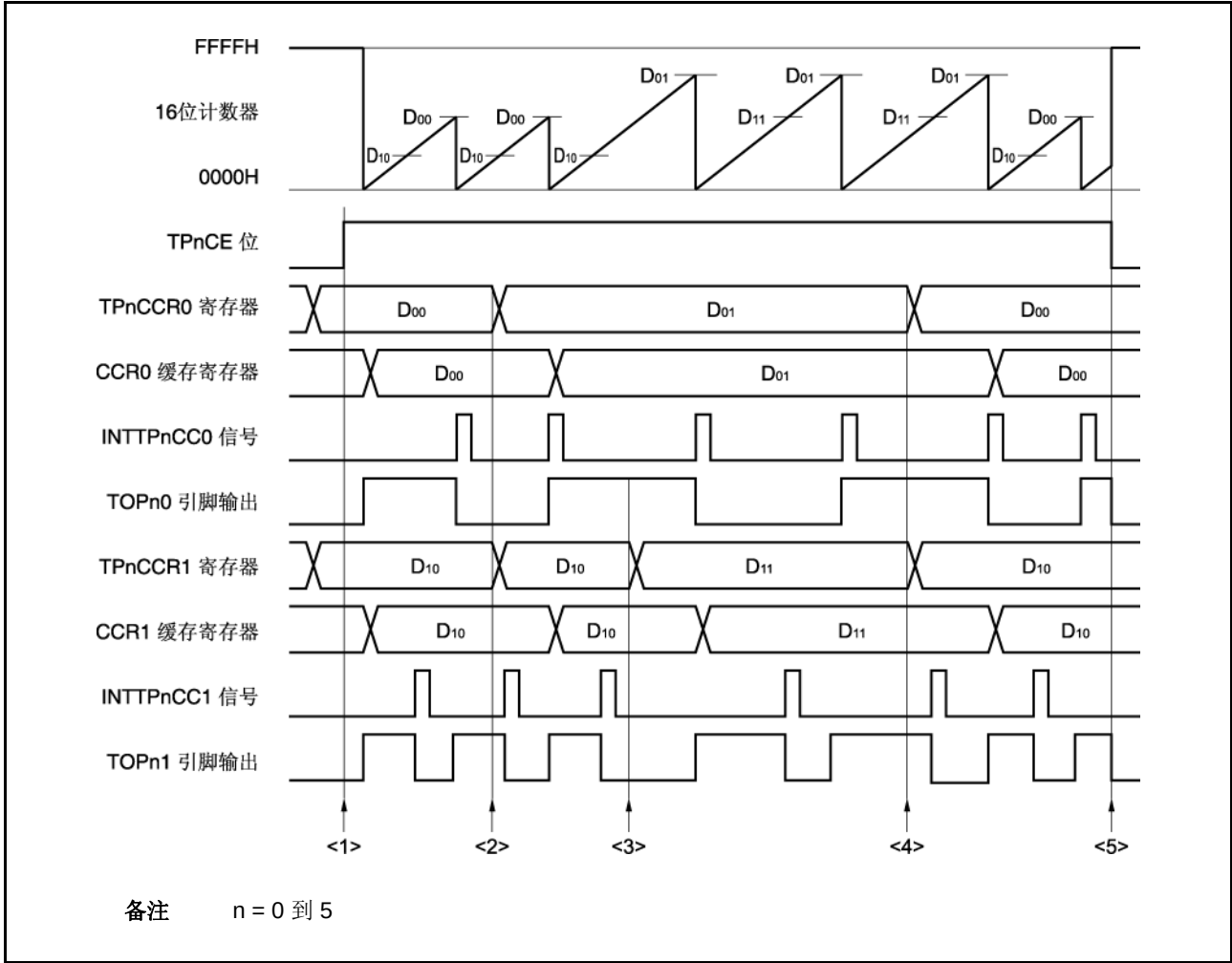
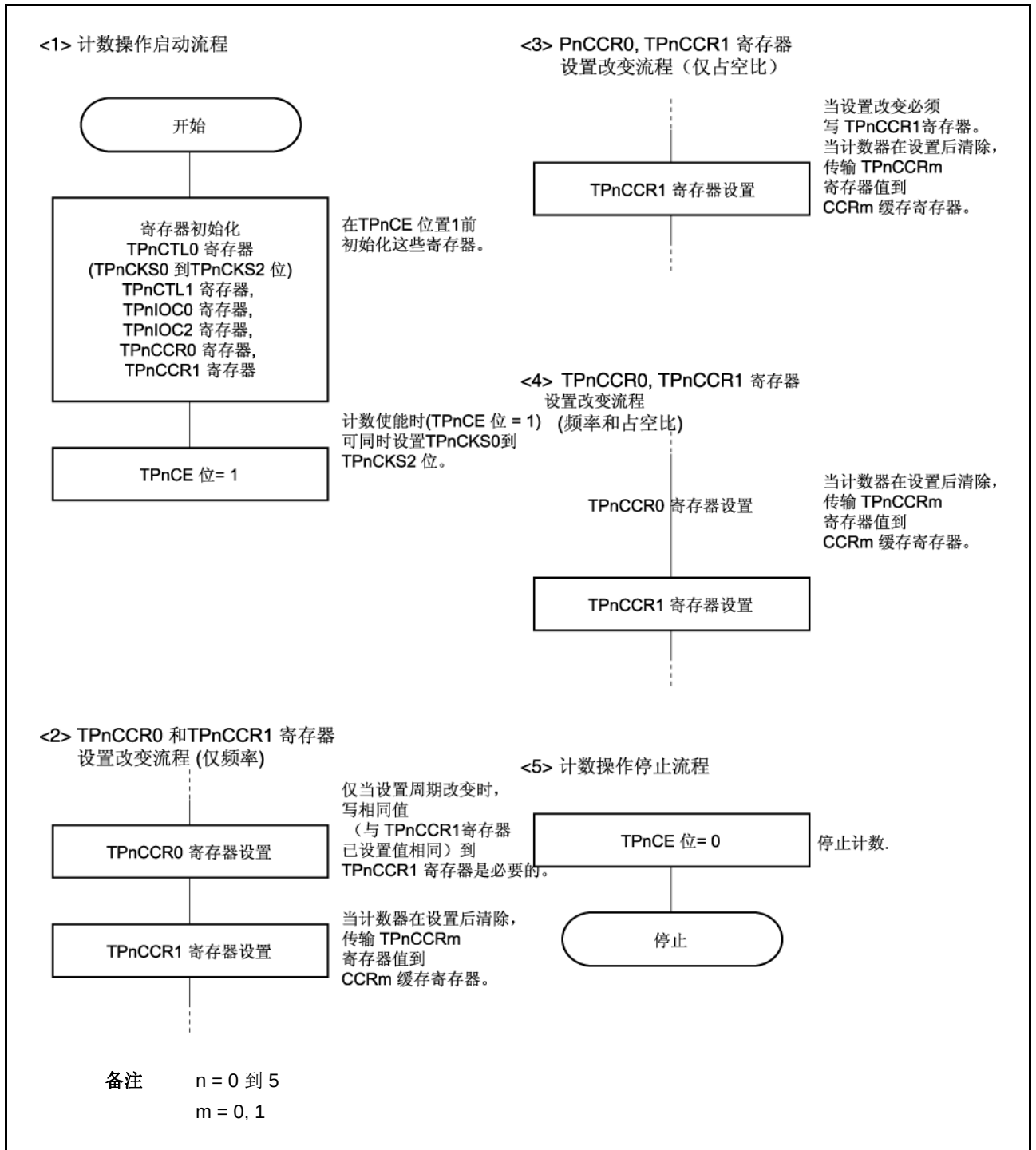


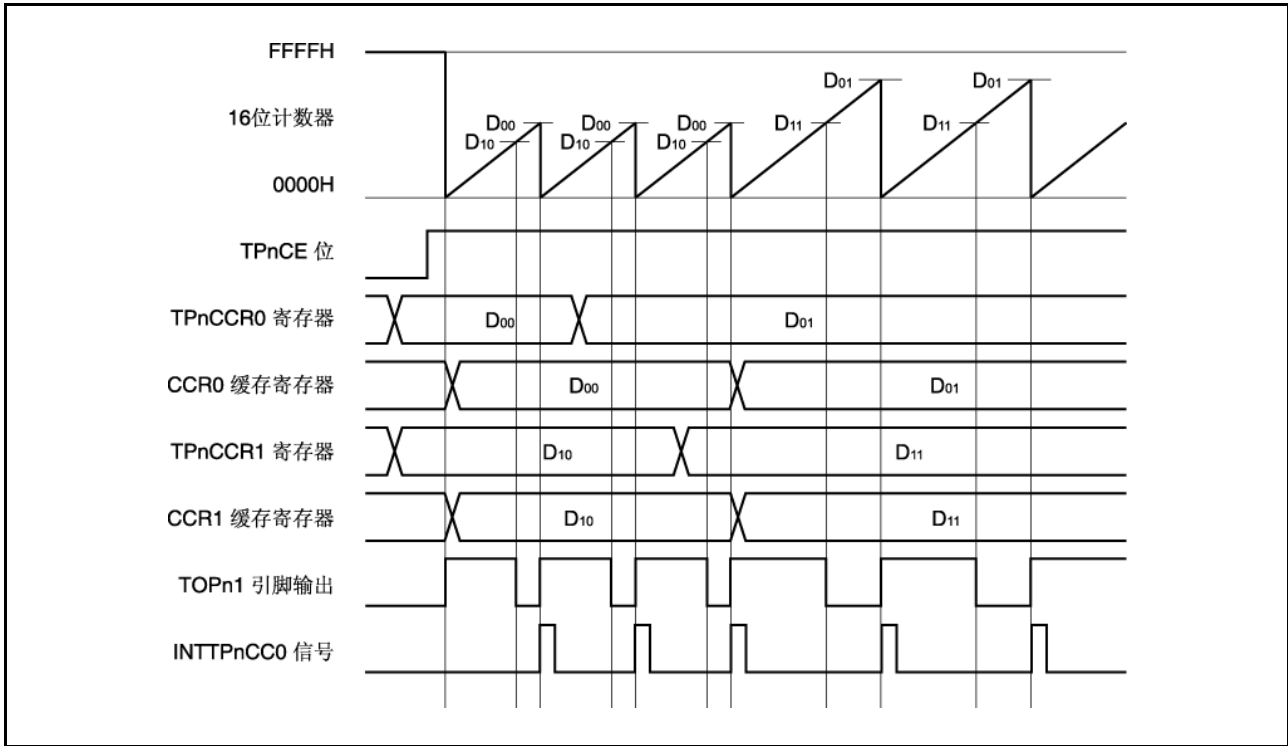
图 7-31. PWM 输出模式的软件处理流程 (2/2)



(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 TPnCCRa 寄存器的值，须等待下一个 INTTPnCC1 信号被检测到后，再进行相关操作。



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

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

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

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

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

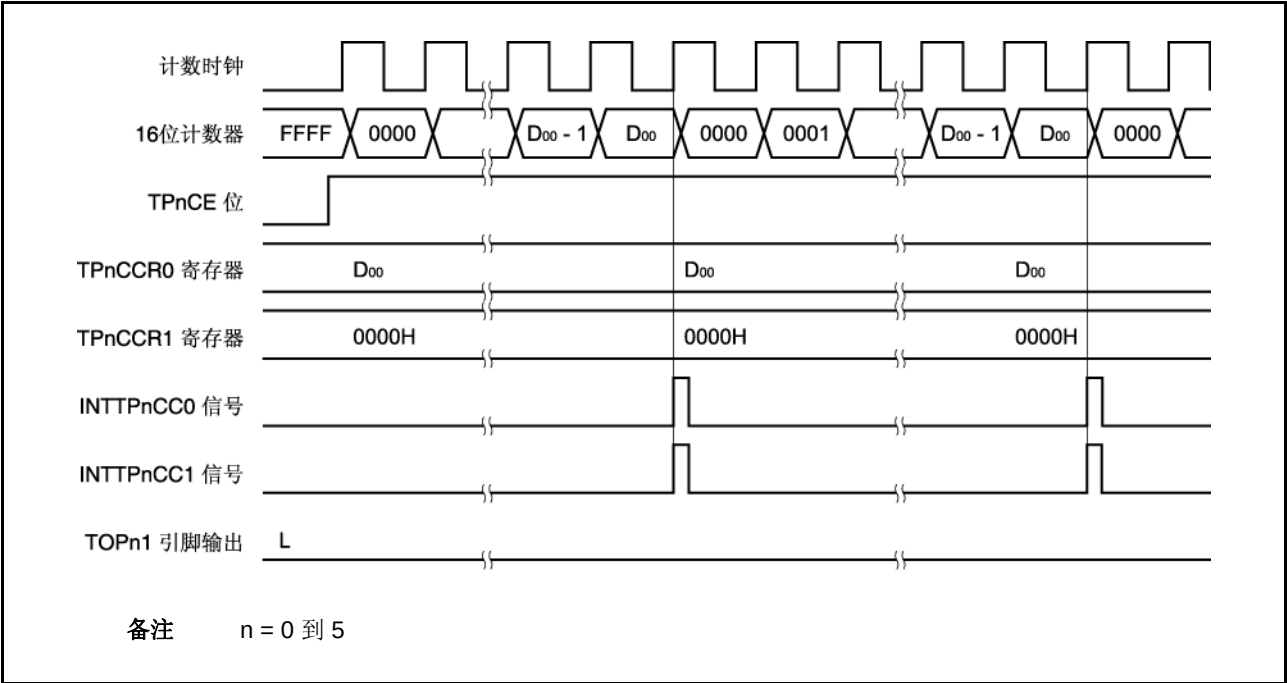
向 TPnCCR1 寄存器的写入操作完成后，若需再次改写 TPnCCR0 或 TPnCCR1 寄存器的值，则要在产生 INTTPnCC0 信号之后再进行改写。否则，由于数据从 TPnCCRm 寄存器传送至 CCRm 缓冲寄存器的时序与 TPnCCRm 寄存器的写入时序相冲突，CCRm 缓冲寄存器中所设置的数值可能无法确定。

备注 n = 0 到 5, m = 0, 1

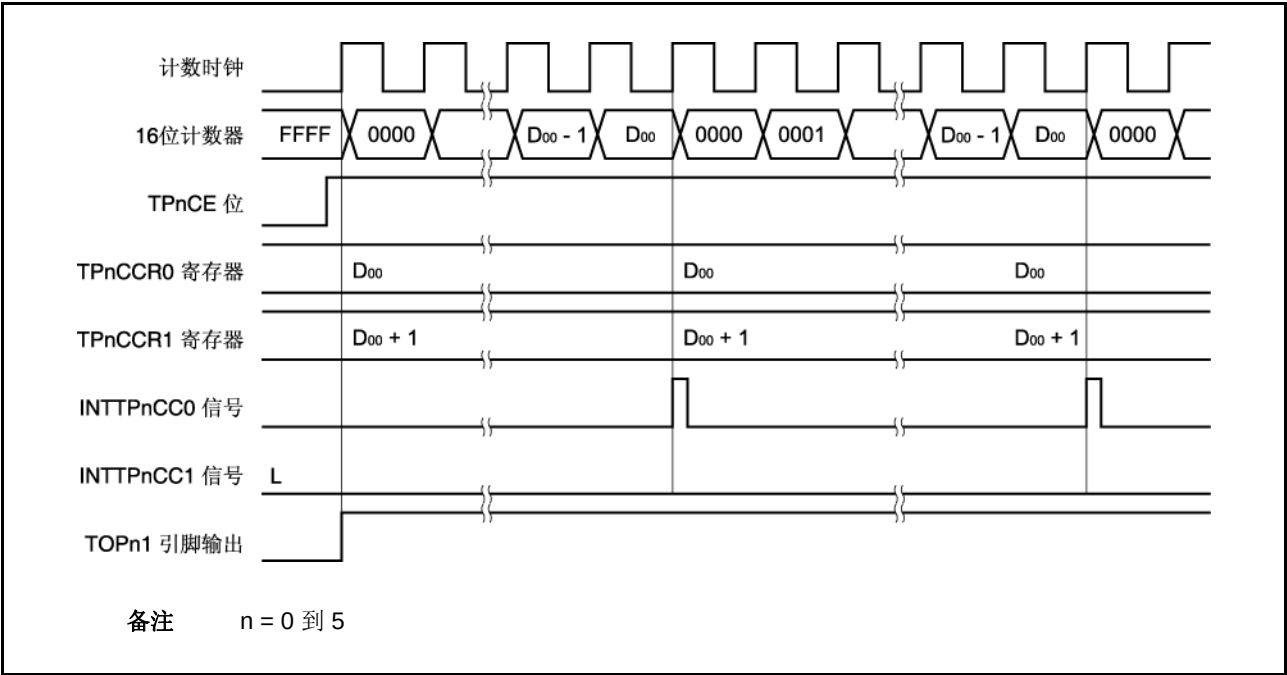
(b) PWM 波形的 0%/100% 输出

<R>

若要输出 0% 的波形，则需将 TPnCCR1 寄存器设置为 0000H。当 16 位计数器的值与 CCR0 缓存寄存器的值匹配后的下一时刻，16 位计数器被清零，产生 INTTPnCC0 和 INTTPnCC1 信号。

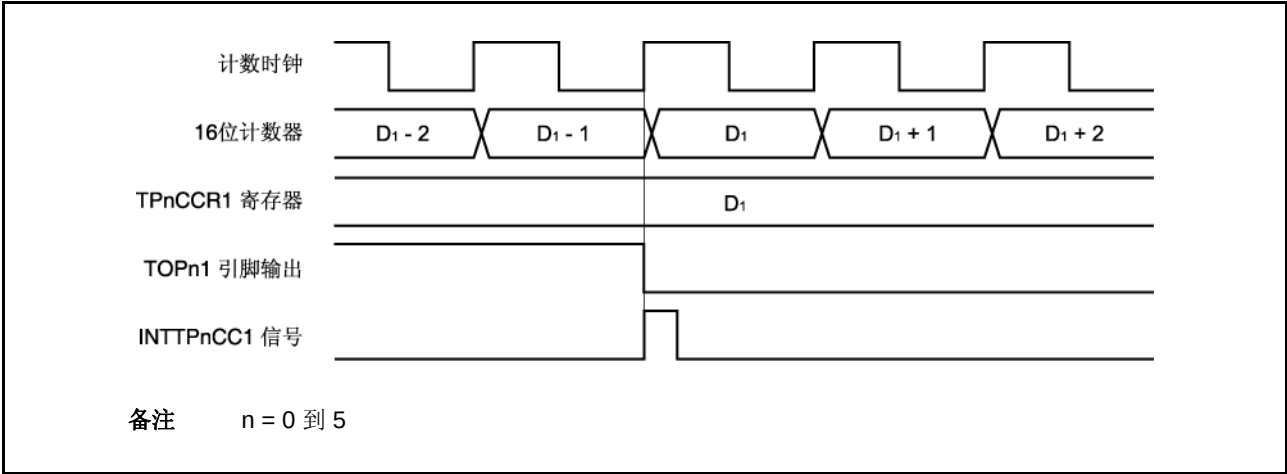


若要输出 100% 波形，则需将 TPnCCR1 寄存器设置为 (TPnCCR0 寄存器值 + 1)。TPnCCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。



(c) 比较匹配中断请求信号的产生时序 (INTTPnCC1)

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



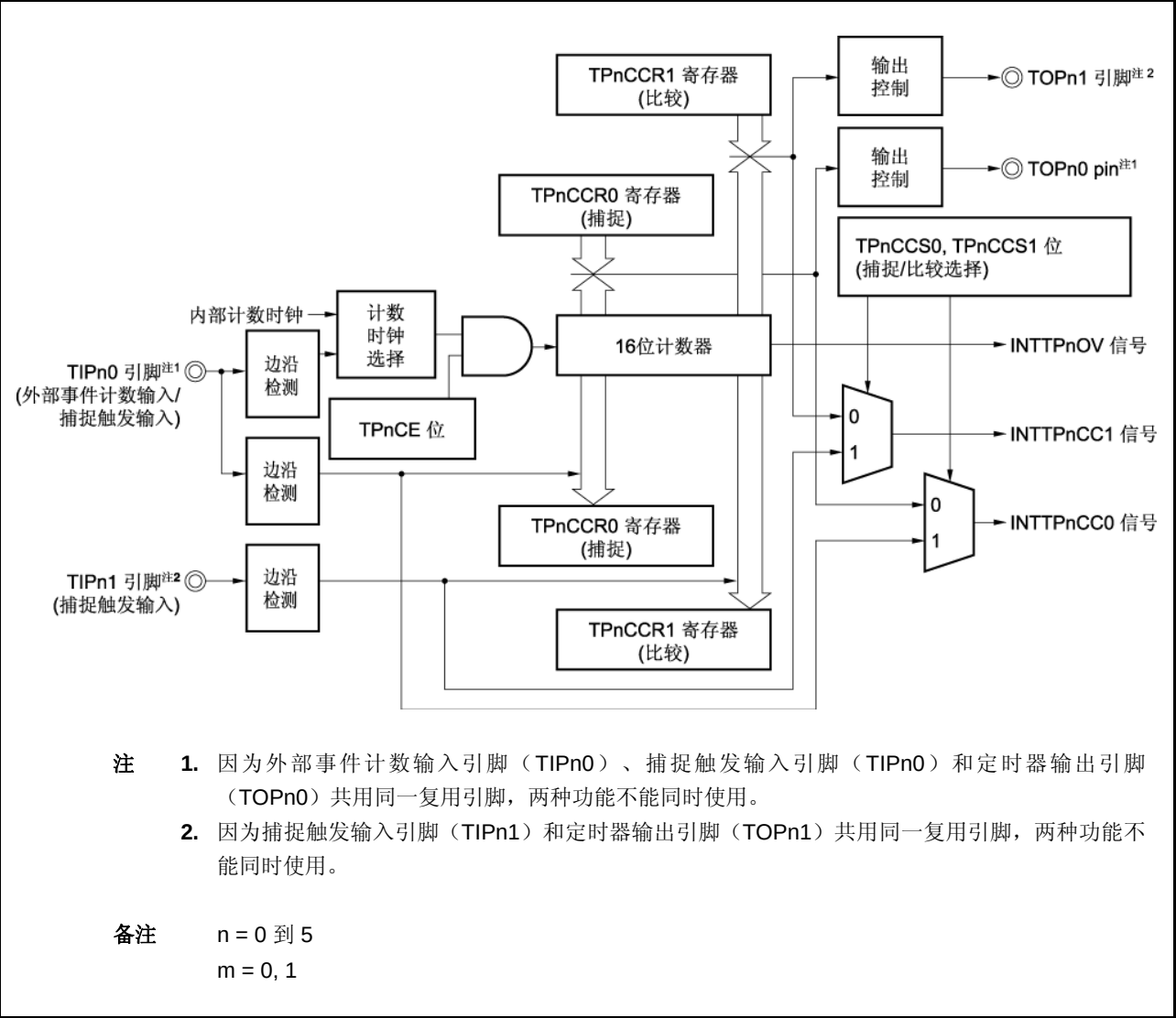
通常情况下, INTTPnCC1 信号会在 16 位计数器计数值与 TPnCCR1 寄存器的值匹配之后的下一个计数时钟同步产生。

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

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

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

图 7-32. 自由运行定时器模式的配置图



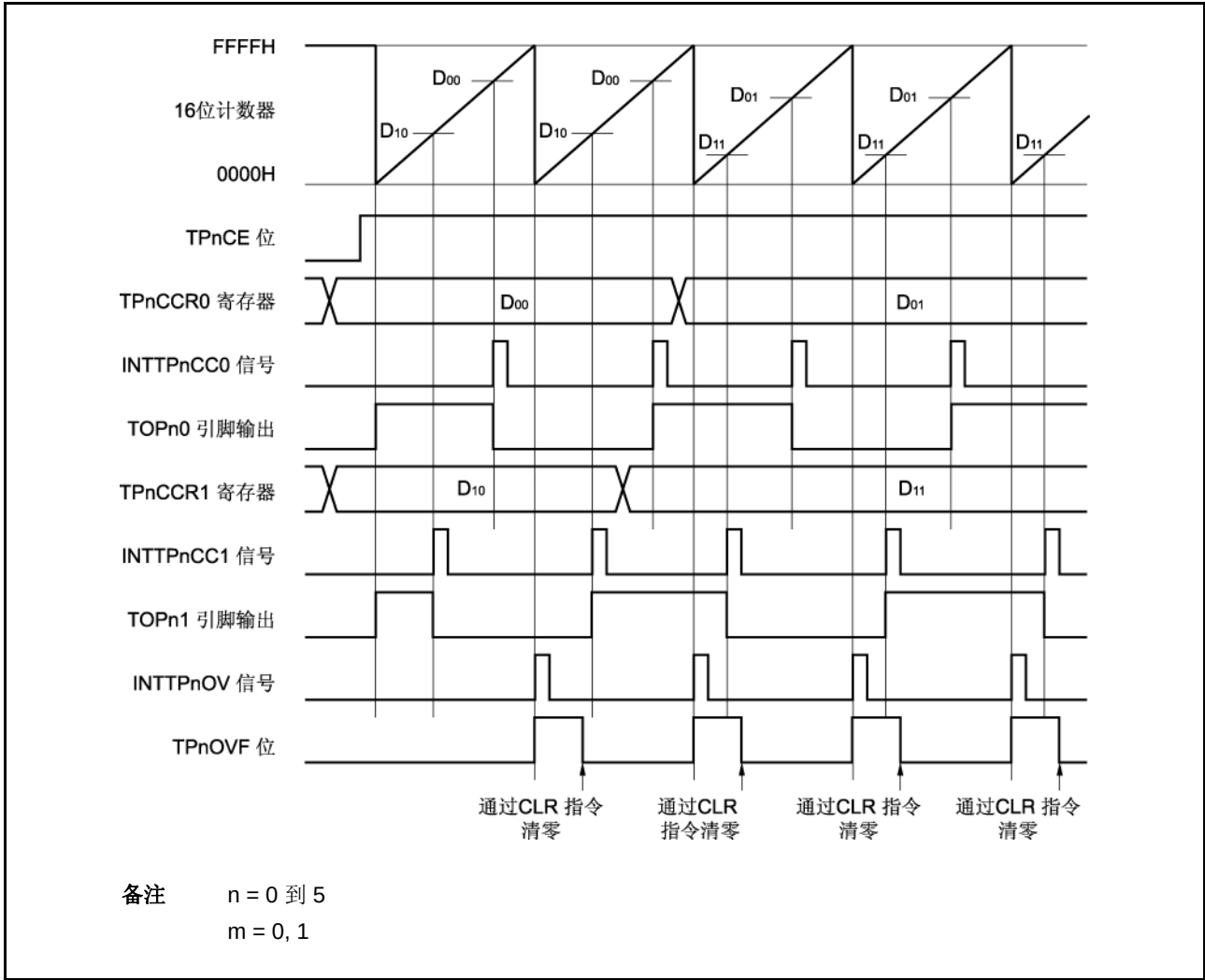
• 比较操作

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 引脚的输出信号将反转。

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。这种情况下，溢出标志位（TPnOPT0.TPnOVF 位）也将被设置为 1。请确认溢出标志置为 1 并通过软件使用 CLR 指令将溢出标志清零。

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

图 7-33. 自由运行定时器模式的基本时序（比较功能）



- (1) 捕捉操作
- TPnCTL0.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。这种情况下, 溢出标志位 (TPnOPT0.TPnOVF 位) 也将被设置为 1。请确认溢出标志置为 1 并通过软件使用 CLR 指令将溢出标志清零。

图 7-34. 自由运行定时器模式的基本时序 (捕捉功能)

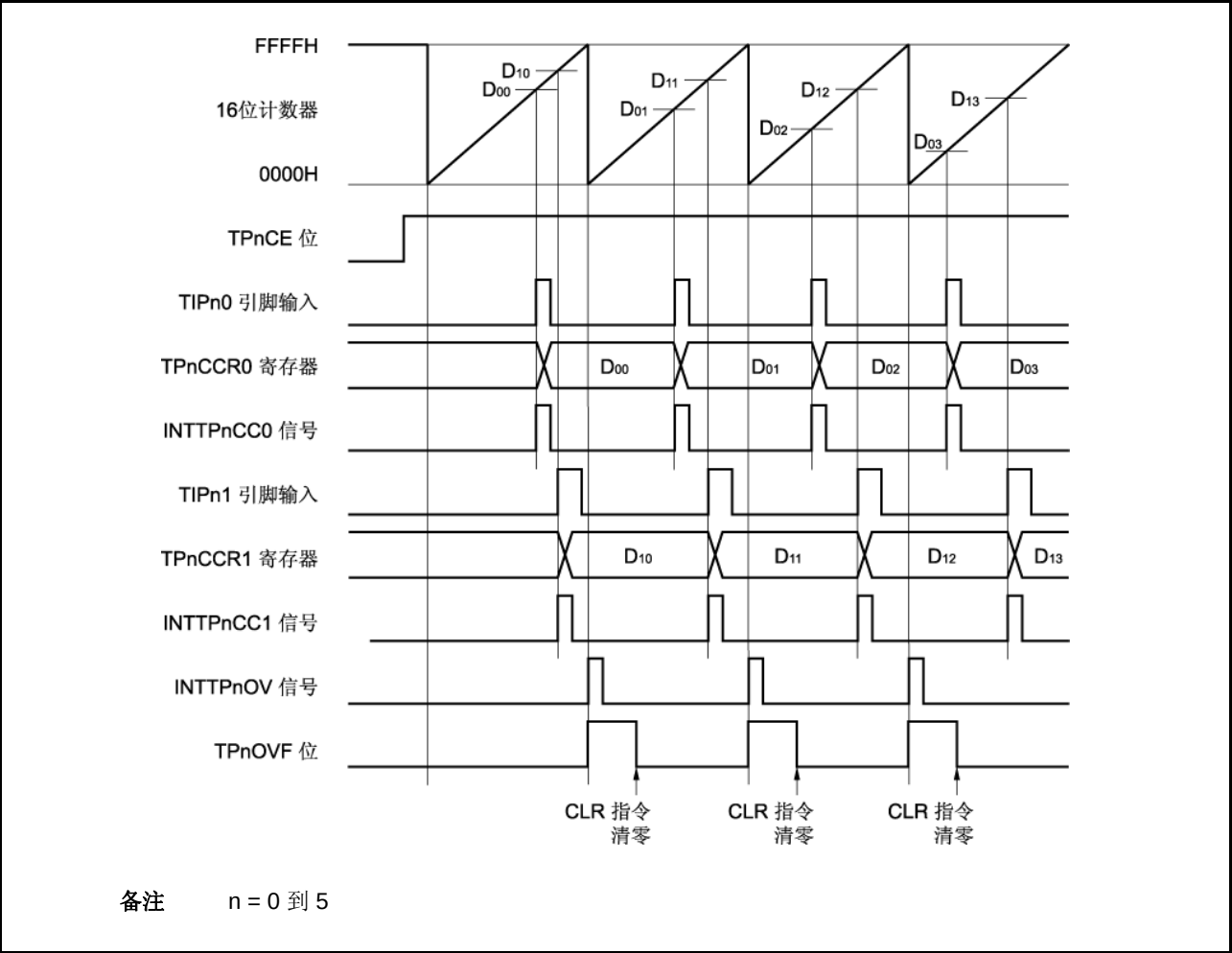
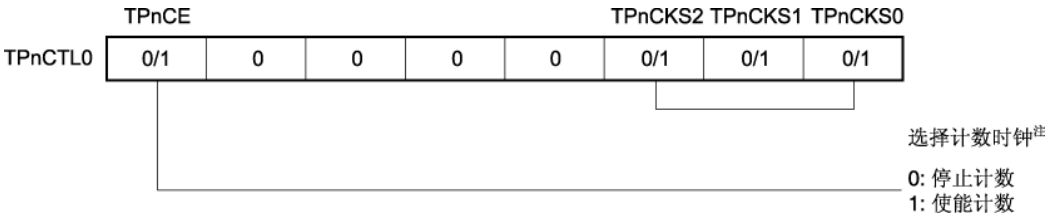


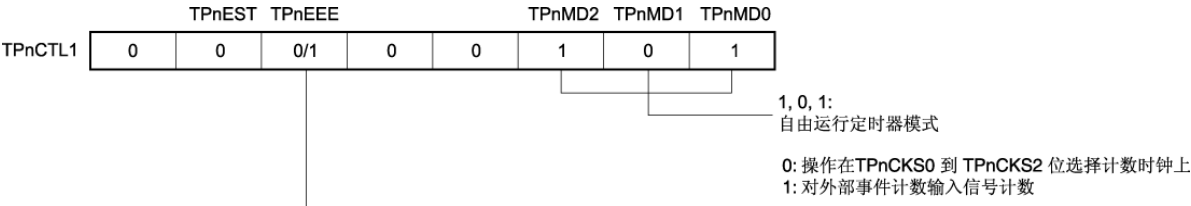
图 7-35. 自由运行定时器模式的寄存器设置 (1/3)

(a) TMPn 控制寄存器 0 (TPnCTL0)

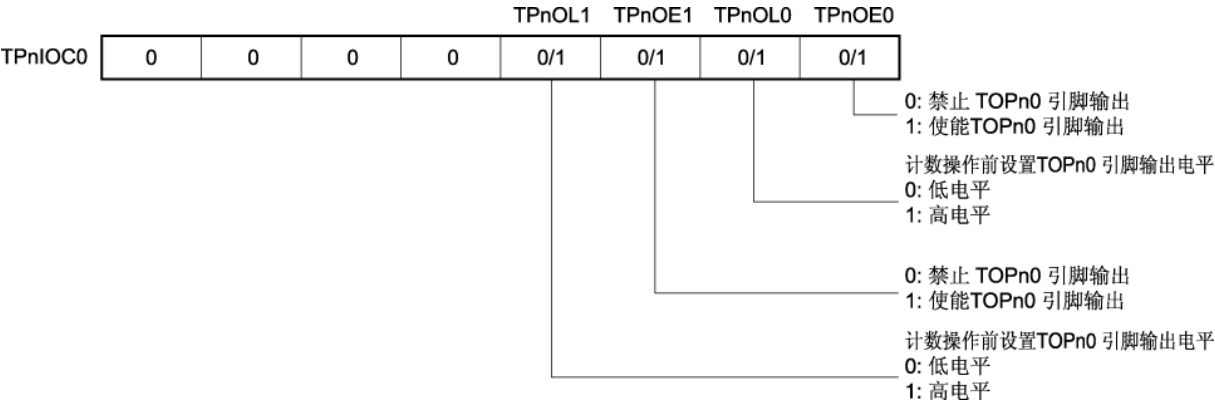


注 当 TPnCTL1.TPnEEE 位 = 1 时, 该设置无效。

(b) TMPn 控制寄存器 1 (TPnCTL1)



(c) TMPn I/O 控制寄存器 0 (TPnIOC0)



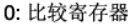


图 7-35. 自由运行定时器模式的寄存器设置 (3/3)

(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 D_m 时，INTTPnCCm 信号将在计数器计数到 $(D_m + 1)$ 时产生，同时 TOPnm 引脚的输出电平反转。

备注 $n = 0$ 到 5
 $m = 0, 1$

(1) 自由运行定时器模式的流程图

(a) 当捕捉/比较寄存器用作比较寄存器时

图 7-36. 自由运行定时器模式的软件处理流程（比较功能）（1/2）

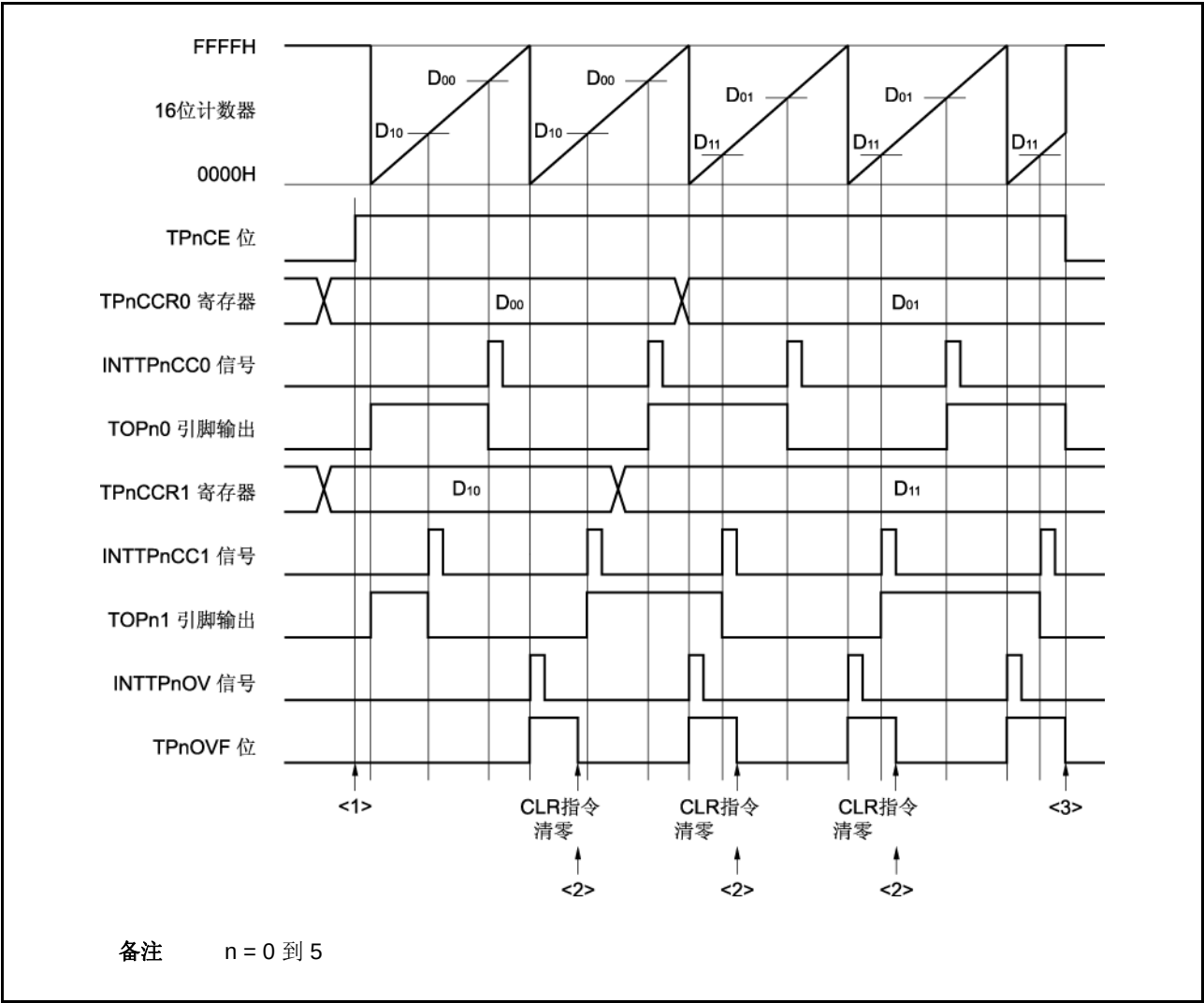
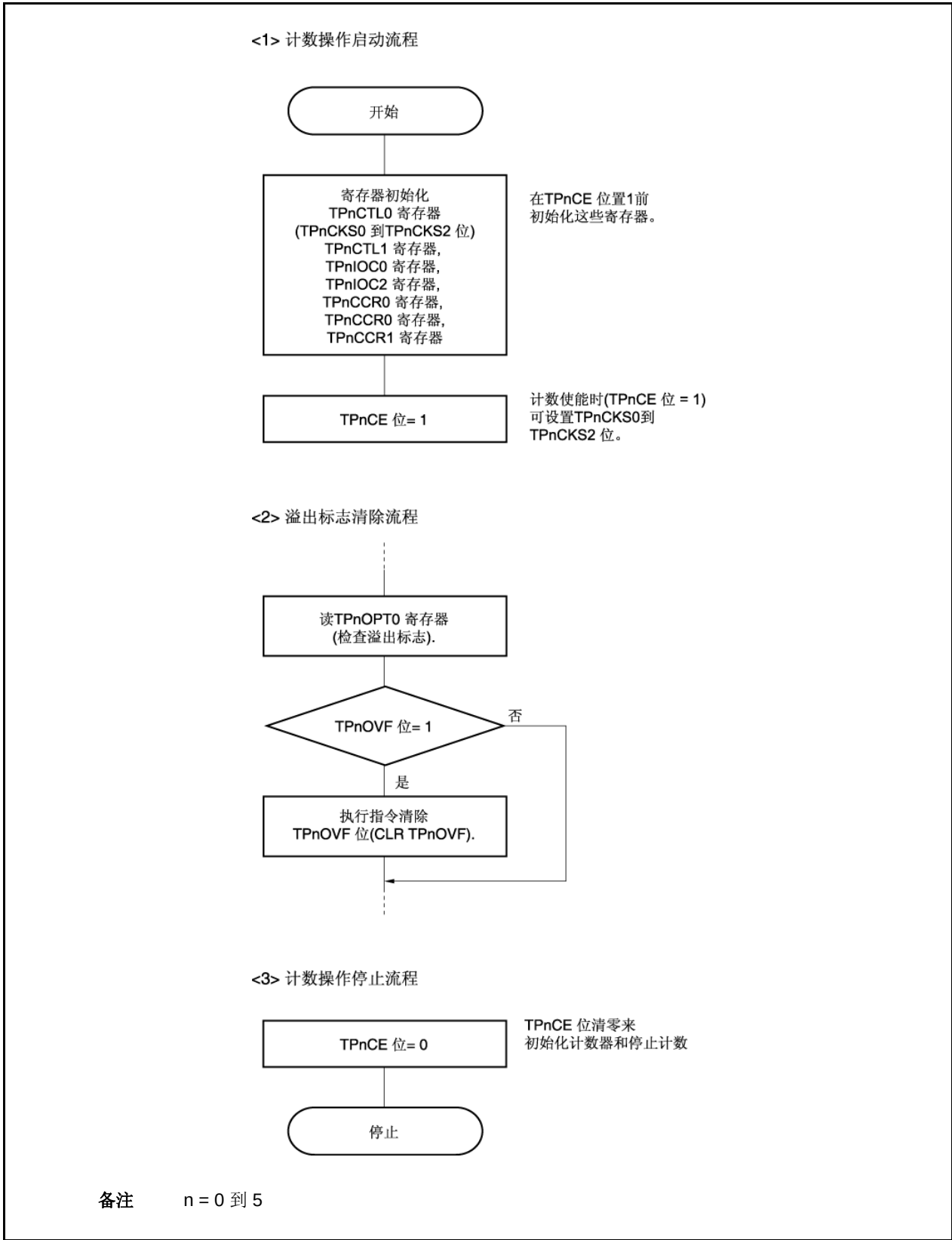


图 7-36. 自由运行定时器模式的软件处理流程（比较功能）（2/2）



(b) 当捕捉/比较寄存器用作捕捉寄存器时

图 7-37. 自由运行定时器模式的软件处理流程（捕捉功能）（1/2）

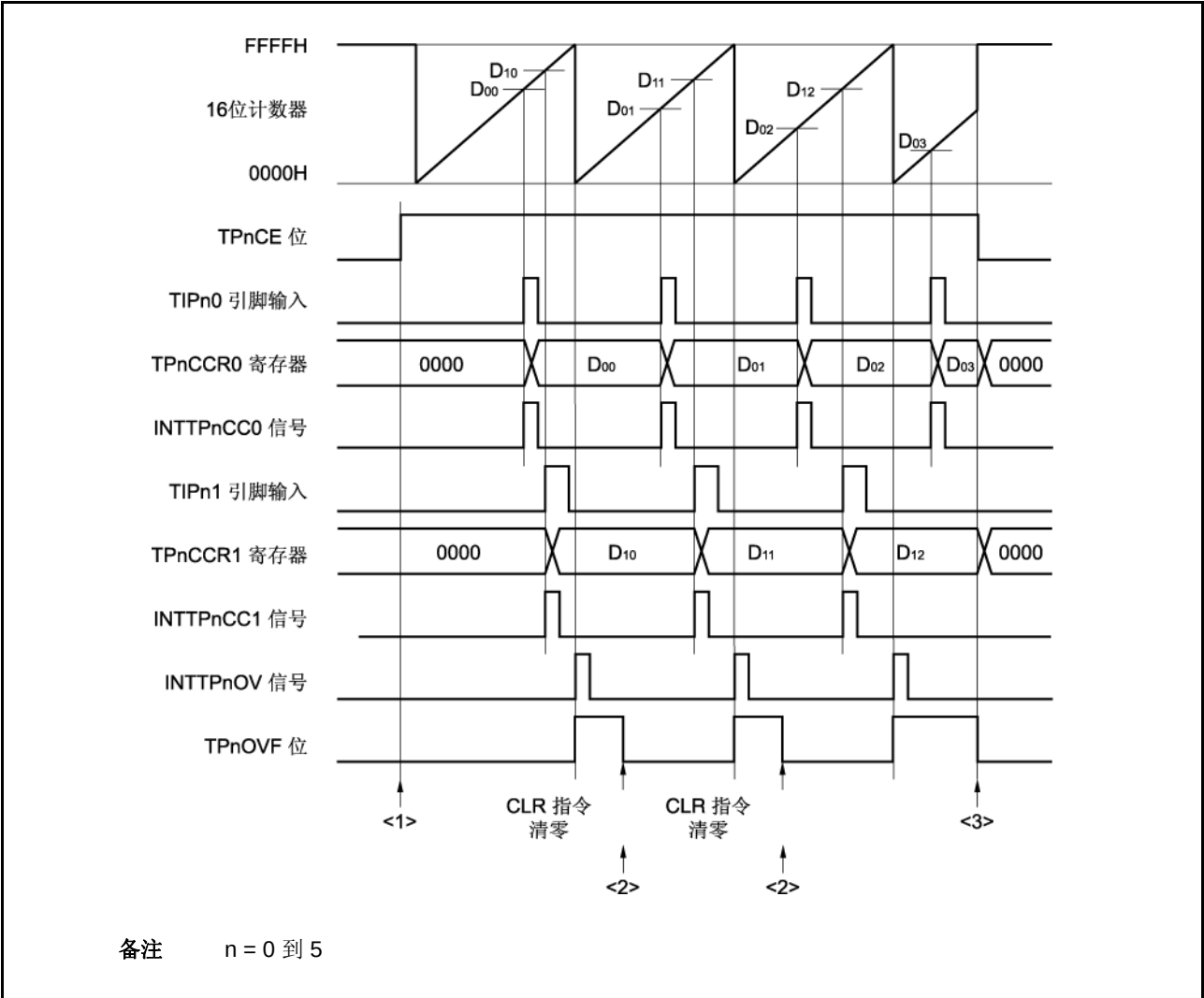
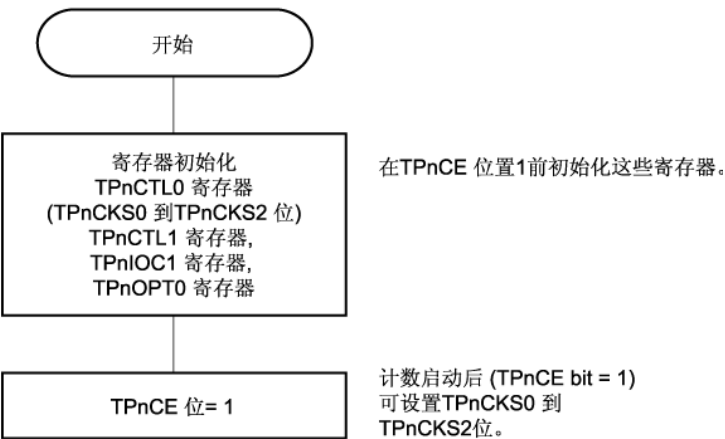
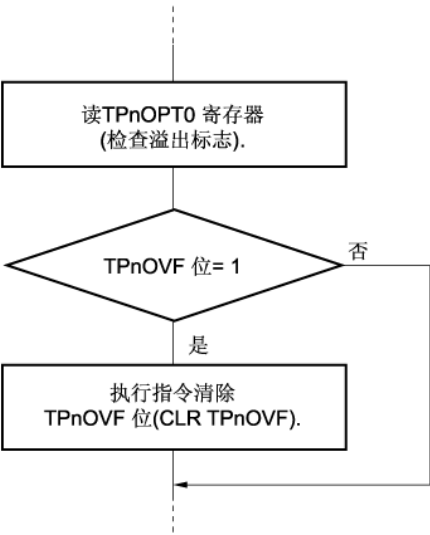


图 7-37. 自由运行定时器模式的软件处理流程（捕捉功能）（2/2）

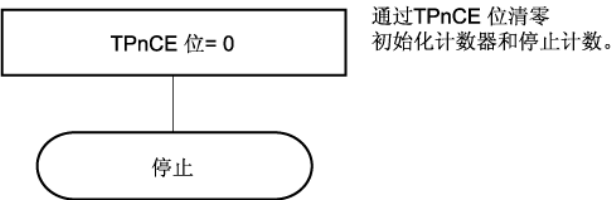
<1> 计数操作启动流程



<2> 溢出标志清除流程



<3>计数操作停止流程

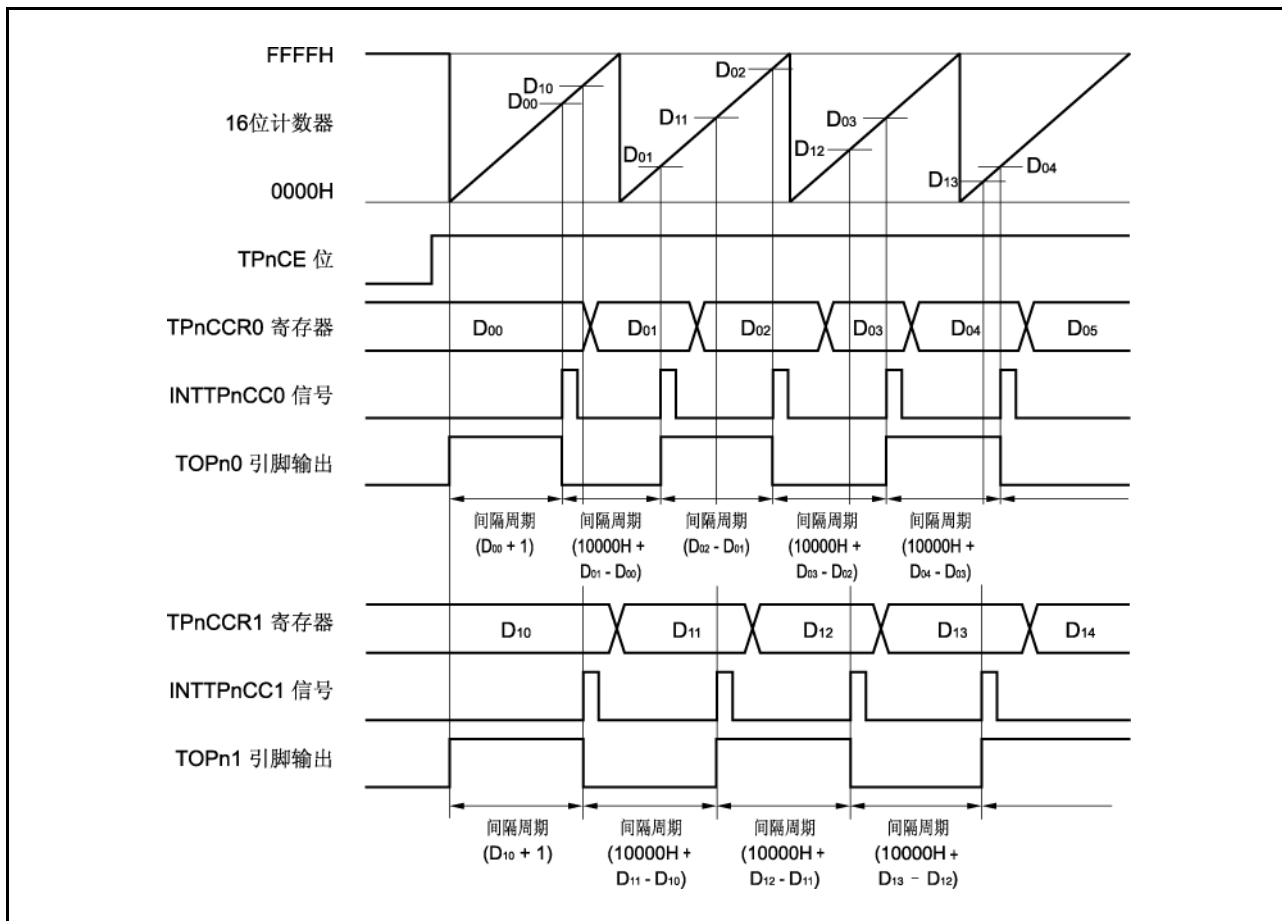


备注 n = 0 到 5

(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 m 个周期。

比较寄存器的最初设置值: $D_m - 1$

第二次设置比较寄存器以后（含第二次）的设置值: 前次设置值 + D_m

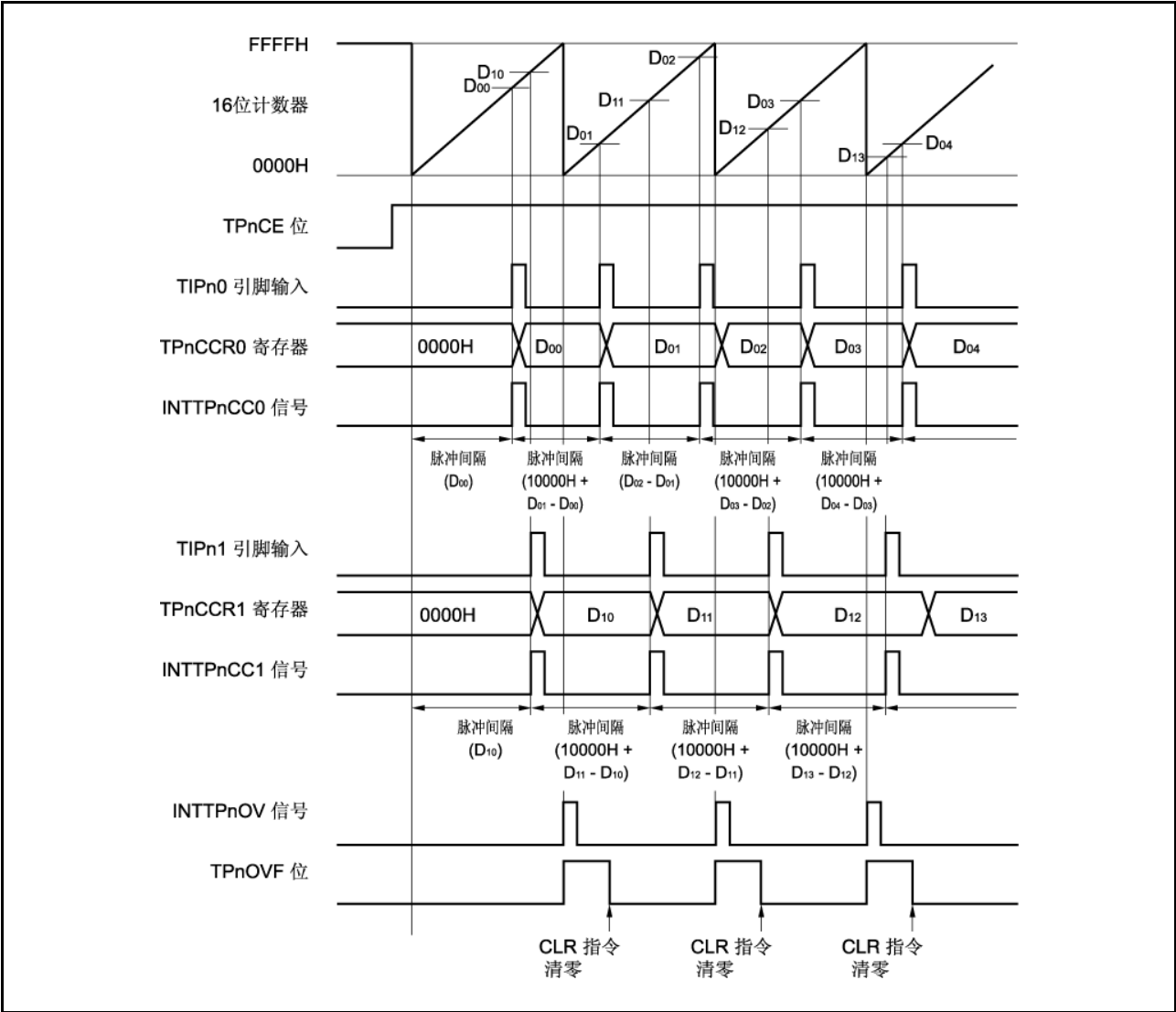
（如果计算结果大于 FFFFH，那么将结果减 10000H 后再设置到寄存器中。

备注 $n = 0$ 到 5

$m = 0, 1$

(b) 使用捕捉寄存器进行脉宽测量

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



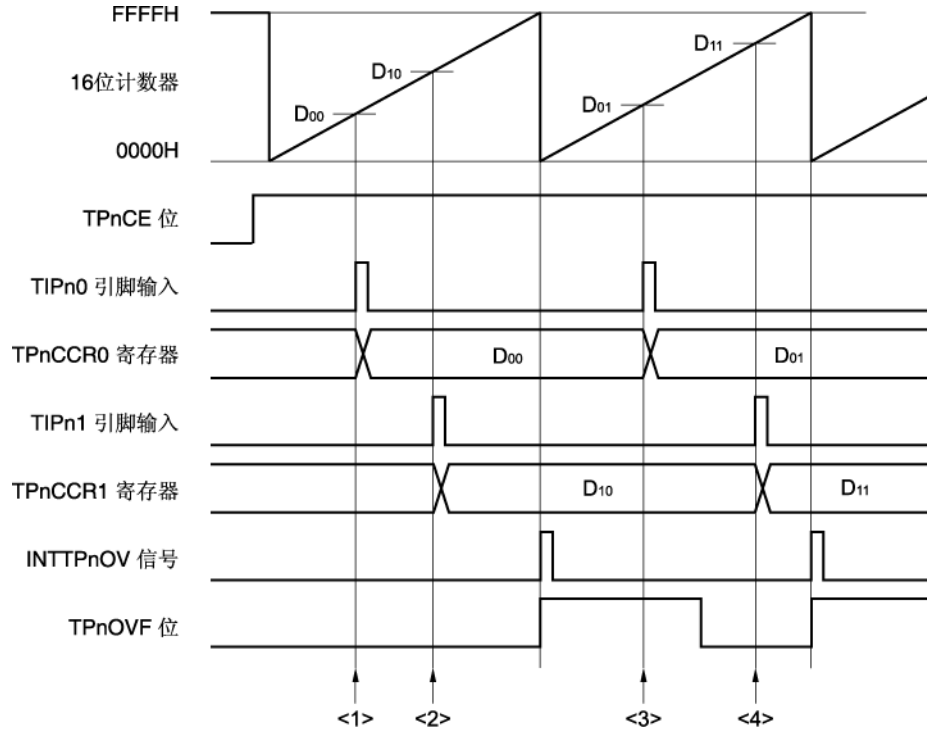
当使用自由运行定时器模式进行脉宽测量操作时, 可以在同一通道对两个不同的脉冲宽度进行测量。通过与 INTTPnCCm 信号同步读取 TPnCCRm 寄存器的值并计算该值与上一次读取值的差, 就可以求得脉冲的宽度。

备注 n = 0 到 5
 m = 0, 1

(c) 使用两个捕捉寄存器产生溢出时的操作

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

使用两个捕捉寄存器时的错误处理举例



使用自由运行定时器模式对两个脉冲宽度进行测量时，可能出现以下的问题。

<1> 读取 TPnCCR0 寄存器（设置为 TIPn0 引脚输入的初始值）。

<2> 读取 TPnCCR1 寄存器（设置为 TIPn1 引脚输入的初始值）。

<3> 读取 TPnCCR0 寄存器。

读取溢出标志位。如果溢出标志为 1，那么将其清零。

由于溢出标志位为 1，脉冲宽度按照 $(10000H + D_{01} - D_{00})$ 来计算。

<4> 读取 TPnCCR1 寄存器。

读取溢出标志位。由于标志位在步骤<3>中被清零，所以此时读取的值为 0。

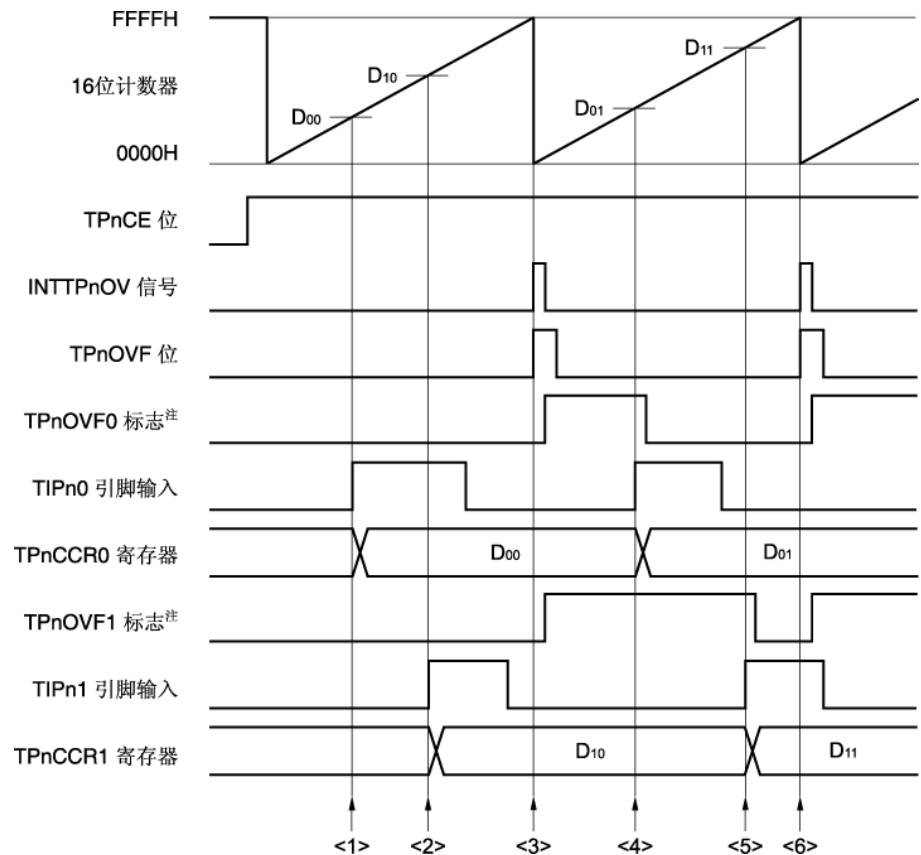
由于溢出标志位为 0，所以脉冲宽度按照 $(D_{11} - D_{10})$ （错误）来计算。

备注 n = 0 到 5

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）

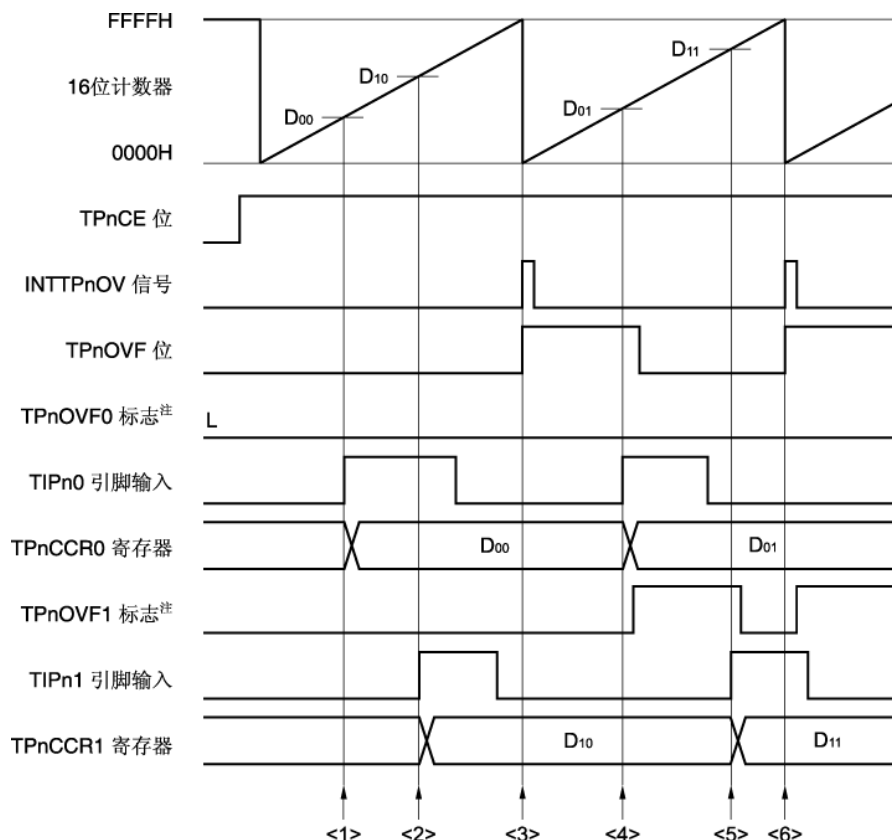


注 这里的 TPnOVF0 和 TPnOVF1 标志是建立于内置 RAM 上的变量，可由软件任意进行设置。

- <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 位。如果 TPnOVF0 标志为 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 位。（步骤<4>中将 TPnOVF0 位清零但 TPnOVF1 位始终保持为 1。
由于 TPnOVF1 标志为 1，脉冲宽度按照 $(10000H + D_{11} - D_{10})$ （正确）来计算。
- <6> 同步骤 <3>

备注 n = 0 到 5

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。

读取溢出标志位。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。

读取溢出标志位。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。

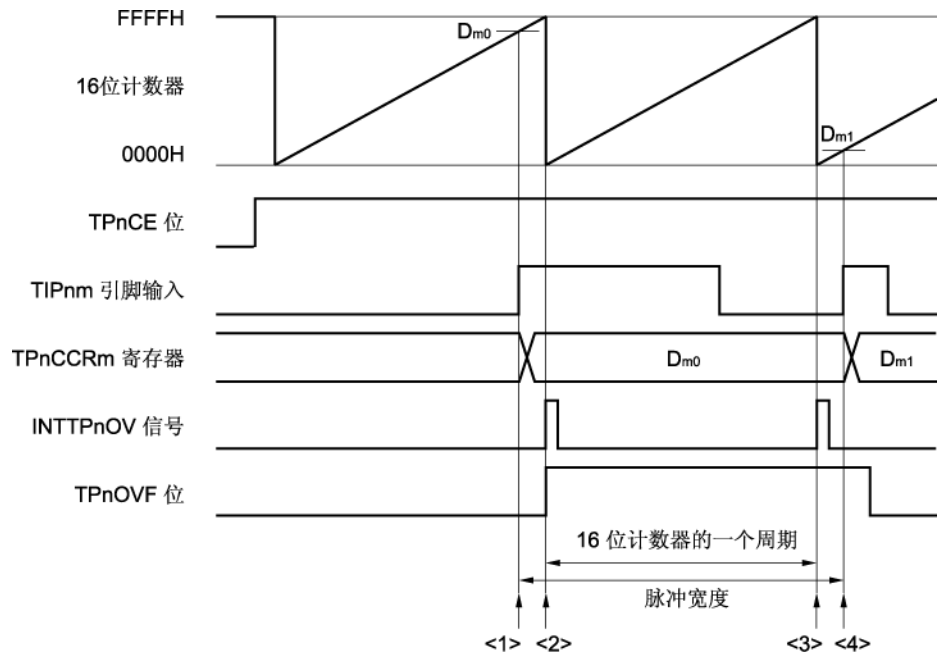
<6> 同步骤 <3>

备注 n = 0 到 5

(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



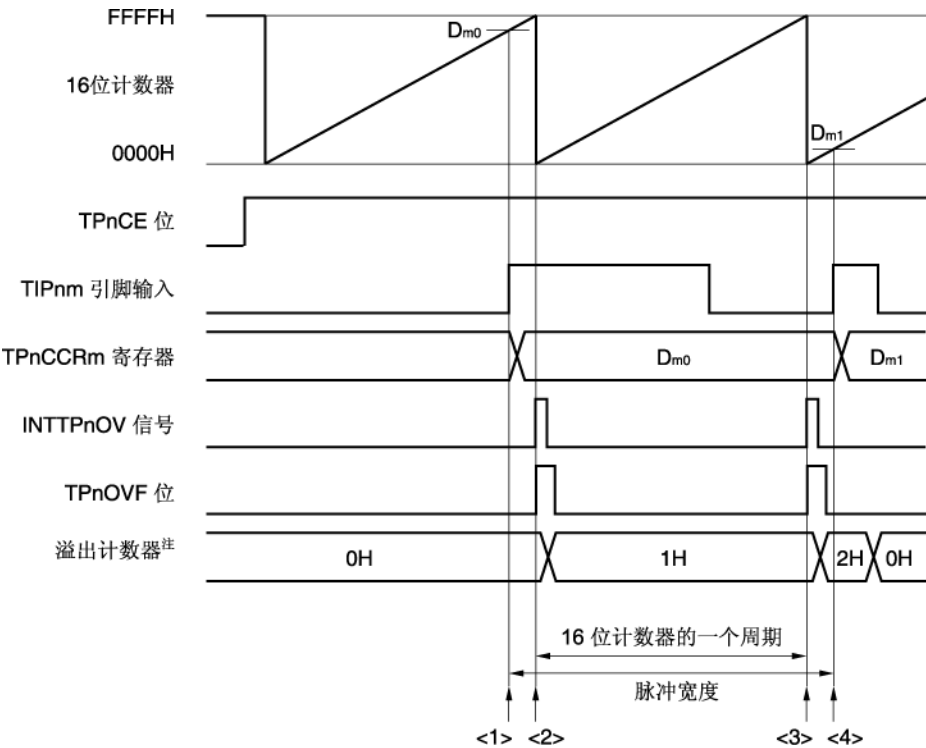
使用自由运行定时器模式对较长的脉冲宽度进行测量时，可能出现以下的问题。

- <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> 产生了第二次溢出。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 $(20000H + D_{m1} - D_{m0})$ 。

备注 n = 0 到 5
 m = 0, 1

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。
 D_{m0} ）。
- 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）。

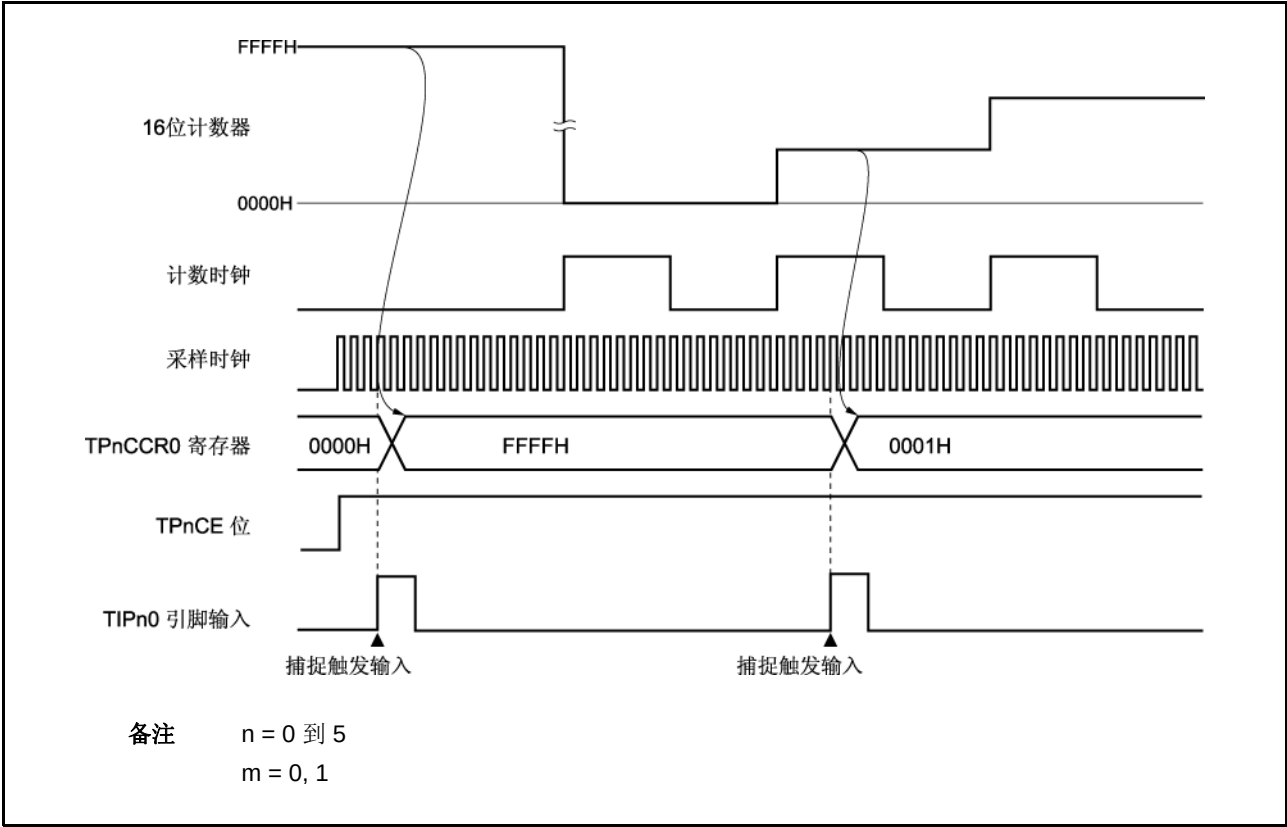
备注 n = 0 到 5
 m = 0, 1

(e) 将溢出标志位清零

通过读 TPnOVF 位后用 CLR 指令清除 TPnOVF 位来清除溢出标志，而通过读 TPnOVF 位后向 TPnOPT0 寄存器写 8 位数据（位 0 为 0）来置位。

<R> (3) 捕捉操作注意

如果使用捕捉操作，慢时钟作为计数时钟，在 TPnCTL0.TPnCE 位设置为 1 后，TPnCCRm 寄存器可能捕捉到 FFFFH，而不是 0000H。



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

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

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

以图 7-39 为例，选择 TIPn0 或 TIPn1 引脚作为捕捉触发输入引脚，用 TPnIOC1 寄存器指定未使用引脚为“无边沿检测”。

图 7-38. 脉宽测量模式的配置图

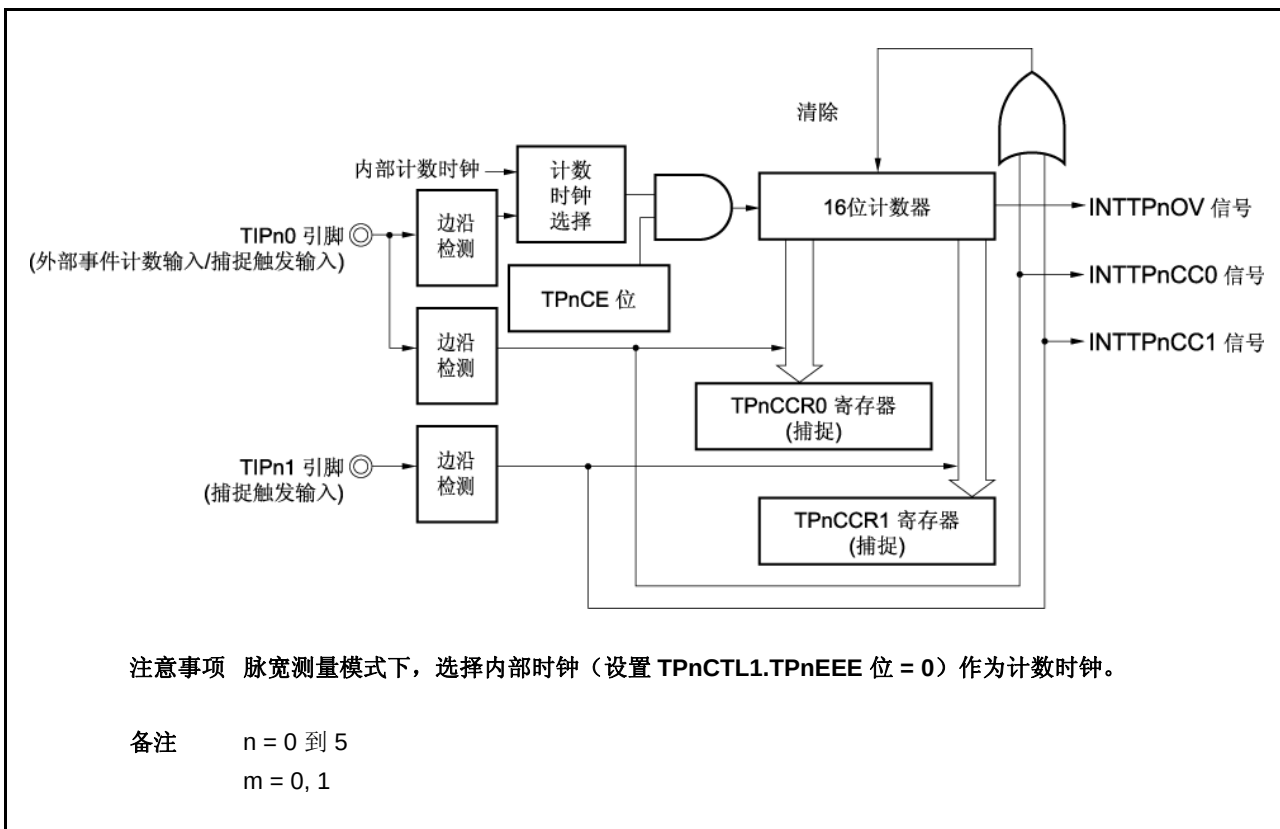
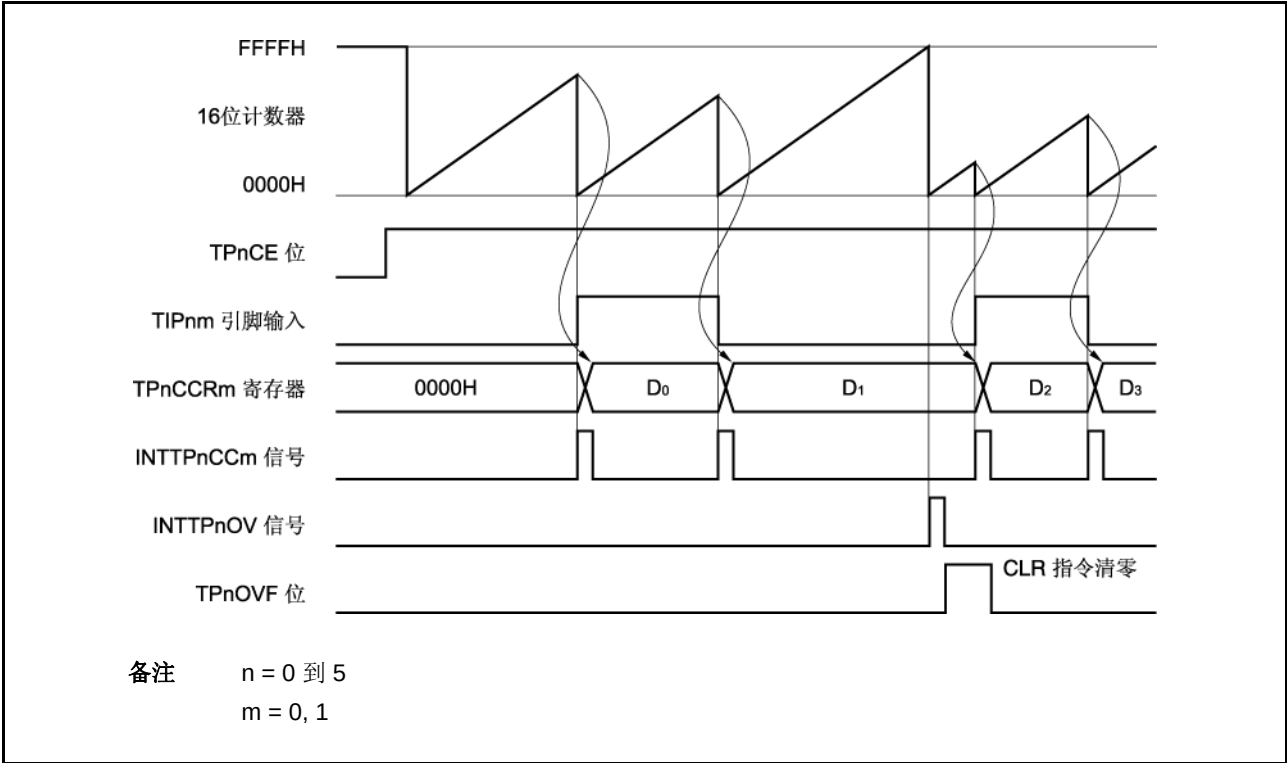


图 7-39. 脉宽测量模式的基本时序



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。

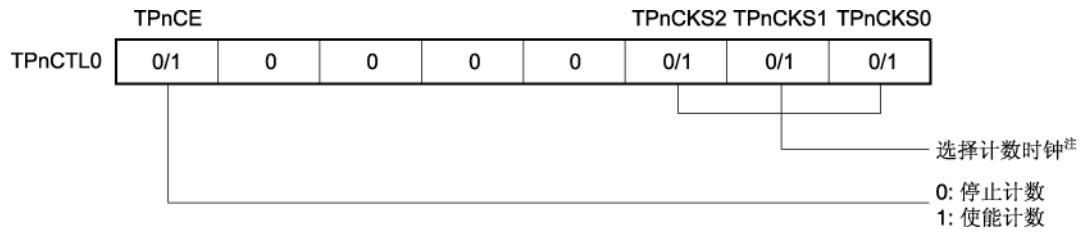
如果溢出标志被设置为 1，那么脉冲宽度可由下式求得。

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

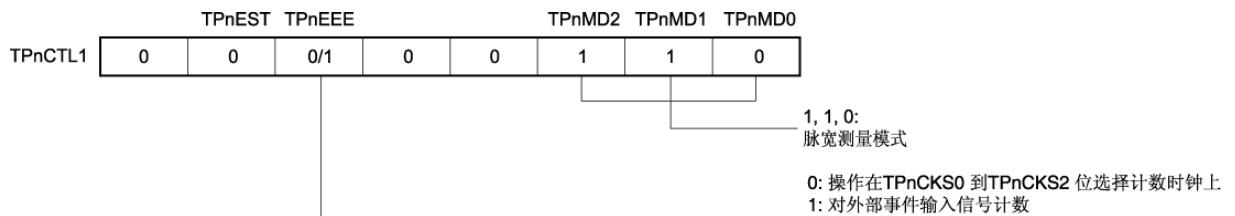
备注 n = 0 到 5
m = 0, 1

图 7-40. 脉宽测量模式的寄存器设置 (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 1 (TPnIOC1)

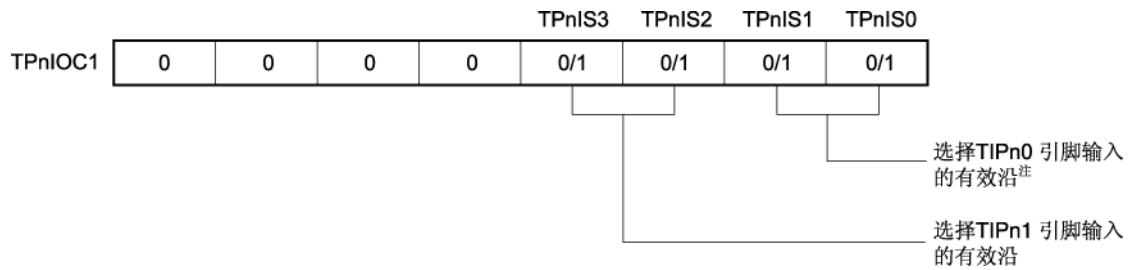
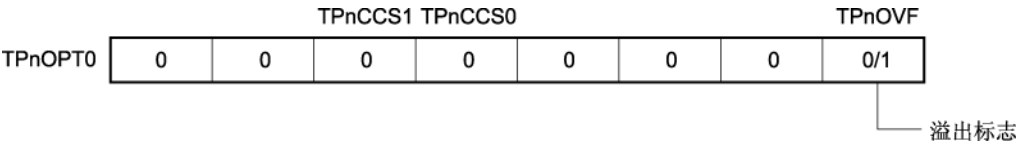


图 7-40. 脉宽测量模式的寄存器设置 (2/2)

(d) TMPn 选项寄存器 0 (TPnOPT0)



(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)

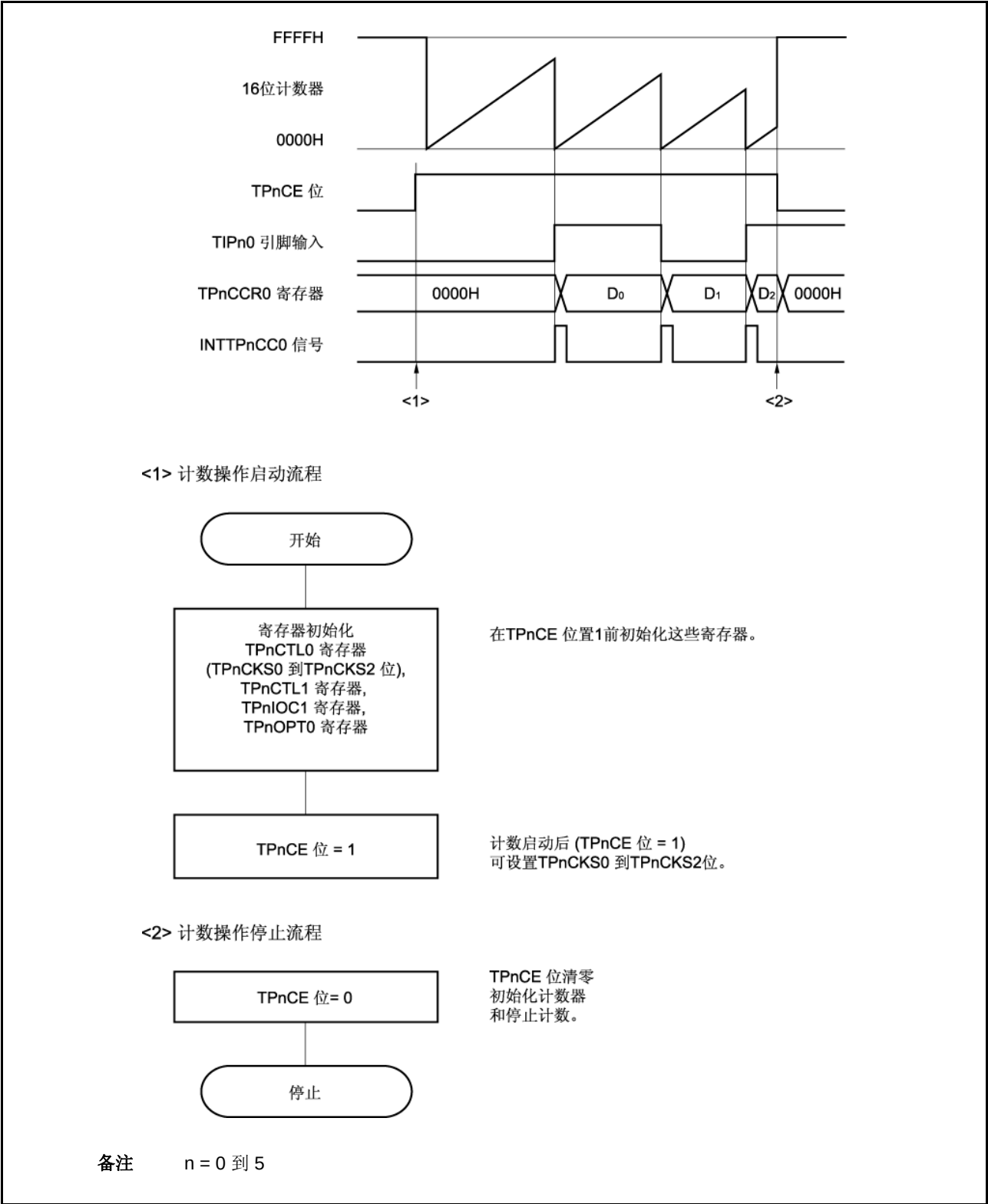
当检测到 TIPnm 引脚的有效沿时，16 位计数器的计数值将被保存在这些寄存器中。

<R>

- 备注**
- 1. 脉宽测量模式下，并未使用 TMPn I/O 控制寄存器 0 (TPnIOC0)，TMPn I/O 控制寄存器 2 (TPnIOC2)。
 - 2. n = 0 到 5
m = 0, 1

(1) 脉宽测量模式的操作流程

图 7-41. 脉宽测量模式的软件处理流程



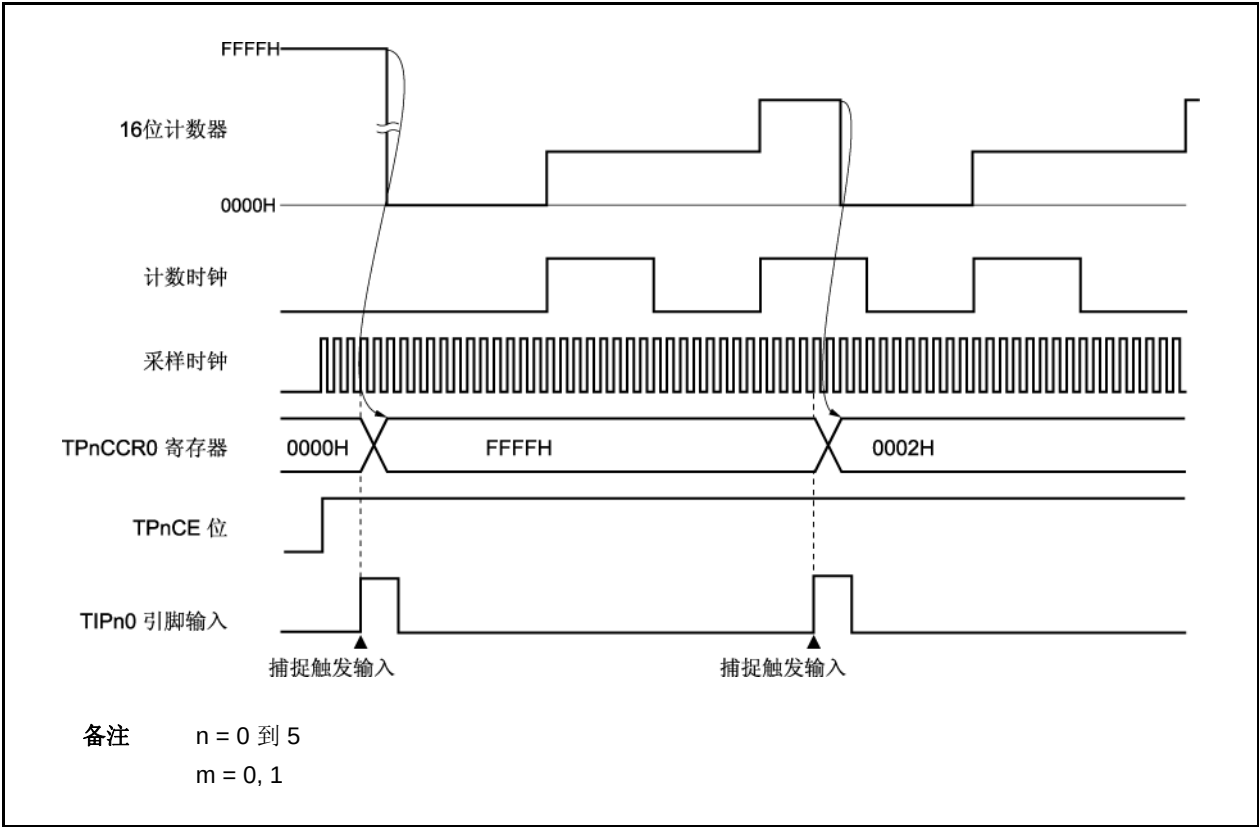
(2) 脉宽测量模式的操作时序

(a) 将溢出标志位清零

通过读 TPnOVF 位后用 CLR 指令清除 TPnOVF 位来清除溢出标志，而通过读 TPnOVF 位后向 TPnOPT0 寄存器写 8 位数据（位 0 为 0）来置位。

<R> (3) 注

如果慢时钟作为计数时钟，在 TPnCTL0.TPnCE 位设置为 1 后，TPnCCRm 寄存器可能捕捉到 FFFFH，而不是 0000H。



7.7 7.6 选择器功能

<R>

在 V850ES/SG2 和 V850ES/SG2-H 中, TIP input/RXDA 的输入和 TIQ input/TSOUT 的信号可以分别用来选择 TMP 和 TMQ 的捕捉触发器输入。

通过使用这个功能, 下列操作成为可能。

- TMQ0 的 TIQ02 输入信号可从端口的定时器复用引脚 (TIQ02 引脚) 和 CAN 控制器的 TSOUT 信号选择。
→ 如果选择了 CAN0 的 TSOUT 信号, 可使用 CAN 控制器的时间标志功能。
- 从端口的定时器复用功能引脚 (TIP10 和 TIP11 引脚) 和 UARTA 的接收复用功能引脚 (RXDA0 和 RXDA1) 可以选择 TMP1 的 TIP10 和 TIP11 输入信号。
→ 当选择了 UART0, UART1 的 RXDA0, RXDA1 信号时, LIN 接收传输率和 UARTA 的波特率误差可被计算。

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 或 TMQ 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/UARTA 或 TMQ/CAN 控制器)。

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 TMP1, TMP3 和 TMQ0 的捕捉触发信号的 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 读/写 地址: FFFFF308H

	7	6	5	<4>	<3>	2	1	<0>
SELCNT0	0	0	0	ISEL4	ISEL3	0	0	ISEL0

ISEL4	TIP11 输入信号选择 (TMP1)
0	TIP11 引脚输入
1	RXDA1 引脚输入

ISEL3	TIP10 输入信号选择 (TMP1)
0	TIP10 引脚输入
1	RXDA0 引脚输入

ISEL0 ^注	TIQ02 输入信号选择 (TMQ0)
0	TIQ02引脚输入
1	CAN0的TSOUT信号

注 仅在含有 CAN 控制器的版本中才可对 ISEL0 位进行操作。

- 注意事项
1. 将 ISEL0, ISEL3, ISEL4 的位置 1, 并设置捕捉输入模式中相应的管脚。
 2. 请务必将第 7 到 5 位, 第 2 和第 1 位清零。

第 8 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。

<R>

V850ES/SG2 和 V850ES/SG2-H 具有 TMQ0。

8.1 概述

以下是 TMQ0 的概况。

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

8.2 功能

TMQ0 具有以下功能。

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

8.3 配置

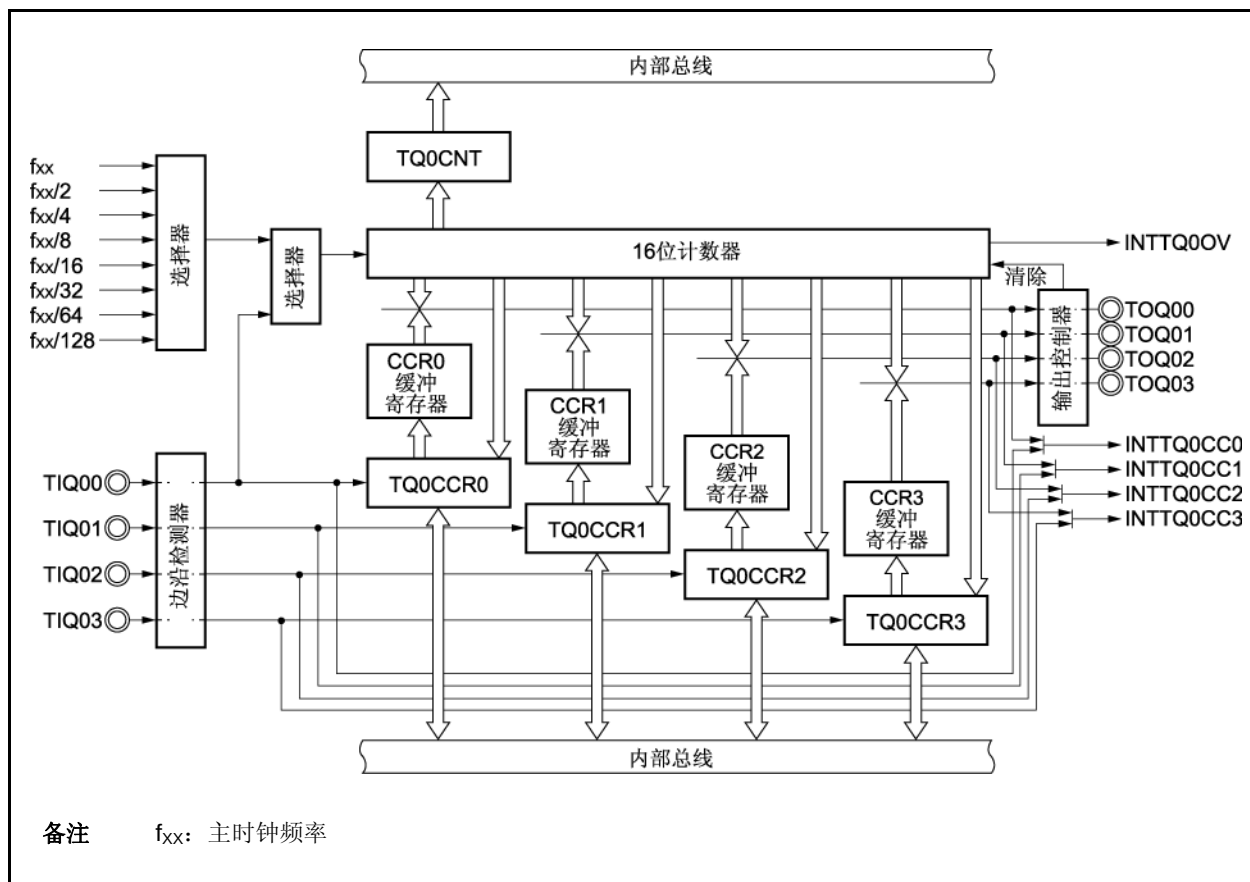
TMQ0 包含以下硬件。

表 8-1. TMQ0 的配置

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

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

图 8-1. TMQ0 的框图



(1) 16 位计数器

该 16 位计数器可以对内部时钟或外部事件进行计数。

可以使用 TQ0CNT 寄存器读取该计数器的计数值。

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

复位置 TQ0CE 位为 0。

(2) CCR0 缓冲寄存器

CCR0 寄存器是一个用于比较 16 位计数器计数值的 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 缓冲寄存器不能被直接读写。

复位后 CCR0 缓冲寄存器和 TQ0CCR0 寄存器都将被清为 0000H。

(3) CCR1 缓冲寄存器

CCR1 寄存器是一个用于比较 16 位计数器计数值的 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 缓冲寄存器不能被直接读写。

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)。

CCR2 缓冲寄存器不能被直接读写。

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 位计数器计数值的 16 位比较寄存器。

当 TQ0CCR3 寄存器用作比较寄存器时, 写到 TQ0CCR3 寄存器的值传输到 CCR3 缓冲寄存器。如果 16 位计数器的计数值与 CCR3 缓冲寄存器的值匹配, 会产生一个比较匹配中断请求信号 (INTTQ0CC3)。

CCR3 缓冲寄存器不能被直接读写。

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。可以将 8 种内部时钟和一个外部事件选择为计数时钟。

8.4 寄存器

用于控制 TMQ0 的寄存器如下所述。

- TMQ0 控制寄存器 0 (TQ0CTL0)
- TMQ0 控制寄存器 1 (TQ0CTL1)
- TMQ0 I/O 控制寄存器 0 (TQ0IOC0)
- TMQ0 I/O 控制寄存器 1 (TQ0IOC1)
- TMQ0 I/O 控制寄存器 2 (TQ0IOC2)
- TMQ0 选项寄存器 0 (TQ0OPT0)
- TMQ0 捕捉 / 比较寄存器 0 (TQ0CCR0)
- TMQ0 捕捉 / 比较寄存器 1 (TQ0CCR1)
- TMQ0 捕捉 / 比较寄存器 2 (TQ0CCR2)
- TMQ0 捕捉 / 比较寄存器 3 (TQ0CCR3)
- TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

备注 使用 TIQ00 至 TIQ03 和 TOQ00 至 TOQ03 引脚功能时，请参阅表 4-15 端口引脚作为复用引脚使用。

(1) TMQ0 控制寄存器 0 (TQ0CTL0)

TQ0CTL0 寄存器为用于控制 TMQ0 操作的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位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		读/写	地址: FFFF540H				
		<7>	6	5	4	3	2 1 0
TQ0CTL0	TQ0CE	0	0	0	0	TQ0CKS2	TQ0CKS1 TQ0CKS0
TQ0CE		TMQ0 操作控制					
0		TMQ0 操作禁止 (TMQ0 异步复位 ^注).					
1		TMQ0 操作使能. TMQ0 操作启动.					
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 位和 16 位计数器同时复位。
另外, 当 16 位计数器复位时, 定时器输出引脚 (TOQ00 到 TOQ03 引脚) 复位到 TQ0IOC0 寄存器设定的状态。

注意事项 1. 在 TQ0CE 位 = 0 时, 设置 TQ0CKS2 至 TQ0CKS0 位为 1。
当将 TQ0CE 位的值从 0 改变为 1 时, 可同时对 TQ0CKS2 至 TQ0CKS0 位进行置位。
2. 请务必将第 3 到 6 位清零。

备注 f_{xx}: 主时钟频率

<R>

(2) TMQ0 控制寄存器 1 (TQ0CTL1)

TQ0CTL1 寄存器是用于控制 TMQ0 操作的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后: 00H 读/写 地址: FFFFF541H

	7	<6>	<5>	4	3	2	1	0
TQ0CTL1	0	TQ0EST	TQ0EEE	0	0	TQ0MD2	TQ0MD1	TQ0MD0

TQ0EST	软件触发控制
0	-
1	为外部触发输入产生有效信号 · 单脉冲输出模式: 向TQ0EST位写1来作为单脉冲输出的触发信号。 · 外部触发脉冲输出模式: 向TQ0EST位写1来作为PWM波形输出的触发信号。
读TQ0EST 位总为 0。	

TQ0EEE	计数时钟选择
0	禁止外部事件计数输入 (TIQ00 引脚)操作。 (TQ0CTL0.TQ0CKS0 到 TQ0CKS2 位选择计数时钟下计数。)
1	使能外部事件计数输入 (TIQ00 引脚)操作。 (对外部事件计数输入信号有效沿计数)
TQ0EEE 位选择对内部计数时钟或外部事件计数输入有效沿计数。	

TQ0MD2	TQ0MD1	TQ0MD0	定时器模式选择
0	0	0	内部定时器模式
0	0	1	外部事件计数输出模式
0	1	0	外部触发脉冲输出模式
0	1	1	单脉冲输出模式
1	0	0	PWM 输出模式
1	0	1	自由运行定时器模式
1	1	0	脉宽测量模式
1	1	1	禁止设置

- 注意事项
1. 对 TQ0EST 位的设置只有在外部触发脉冲输出模式或单脉冲输出模式才有效。在其它模式下, 将忽略对该位的置 1 操作。。
 2. 在外部事件计数模式下, 无论 TQ0EEE 位如何设置, 都将选中外部事件计数输入。
 3. 只有在 TQ0CTL0.TQ0CE 位 = 0 时才可以对 TQ0EEE 以及 TQ0MD2 至 TQ0MD0 位进行设置 (TQ0CE 位 = 1 时, 可以对这些位写入相同的值)。如果在 TQ0CE 位 = 1 时对这些位进行改写, 那么无法保证操作结果。若由于误操作改写这些位, 则要将 TQ0CE 位清零然后再次设置这些寄存器位。
 4. 请务必将第 3、4 和 7 位清零。

(3) TMQ0 I/O 控制寄存器 0 (TQ0IOC0)

TQ0IOC0 寄存器是用于控制定时器输出的 8 位寄存器 (TOQ00 至 TOQ03 引脚)。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

复位后: 00H 读/写 地址: FFFFF542H

	7	<6>	5	<4>	3	<2>	1	<0>
TQ0IOC0	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0

<R>

TQ0OLm	TOQ0m 引脚输出电平设置(m = 0 到 3) ^注
0	TOQ0m 引脚高电平启动
1	TOQ0m 引脚低电平启动

TQ0OEm	TOQ0m 引脚 输出设置(m = 0 到 3)
0	定时器输出禁止 · 当 TQ0OLm 位 = 0: TOQ0m 引脚输出低电平 · 当 TQ0OLm 位 = 1: TOQ0m 引脚输出高电平
1	定时器输出使能(TOQ0m 引脚输出脉冲).

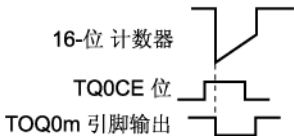
<R>

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。

· 当 TQ0OLm 位 = 0



· 当 TQ0OLm 位 = 1



<R>

- 注意事项
1. 当端口设置为输出 TOQ0m 时重写 TQ0IOC0 寄存器的设置, 引脚输出改变。因此, 通过设置端口为输入模式和使引脚输出状态为高阻态来记录引脚状态的变化。
 2. 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)。若进行了错误的改写, 须先将 TQ0CE 位清零, 然后再次设置这些位。
 3. 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 读/写 地址: FFFF543H

	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) 有效沿设置
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

TQ0IS5	TQ0IS4	捕捉触发输入信号(TIQ02 引脚) 有效沿设置
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

TQ0IS3	TQ0IS2	捕捉触发输入信号(TIQ01 引脚) 有效沿设置
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

TQ0IS1	TQ0IS0	捕捉触发输入信号(TIQ00 引脚) 有效沿设置
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. 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. TQ0IS7 至 TQ0IS0 位只有在自由运行定时器模式 (TQ0OPT.TQ0CCSm 位=1) 和脉宽测量模式下才有效 (m=0 到 3)。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 读/写 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 pin) 有效沿设置
0	0	无边沿检测 (外部触发无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双沿检测

TQ0ETS1	TQ0ETS0	外部触发输入信号 (TIQ00 pin) 有效沿设置
0	0	无边沿检测 (外部触发无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双沿检测

- 注意事项**
1. 在 TQ0CTL0.TQ0CE 位 = 0 时才可对 TQ0EES1, TQ0EES0, TQ0ETS1 和 TQ0ETS0 位进行改写 (TQ0CE 位 = 1 时只可以向这些位写入相同的值)。若进行了错误的改写, 须先将 TQ0CE 位清零, 然后再次设置这些位。
 2. 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 读/写 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	比较寄存器选择
1	捕捉寄存器选择 (设置 TQ0CTL0.TQ0CE 位 = 0清除)
仅当自由运行定时器模式下TQ0CCSm 位 设置有效。	

TQ0OVF	TMQ0 溢出检测标志
Set (1)	发生溢出
Reset (0)	写TQ0OVF 位 0 或 TQ0CTL0.TQ0CE 位 = 0
- 自由运行定时器模式或脉宽测量模式下当16位计数器从FFFFH到0000H 计数值溢出TQ0OVF位置1. - TQ0OVF位置1时产生溢出中断请求信号 (INTTQ0OV)。在除自由运行定时器模式 和脉宽测量模式以外模式下不产生INTTQ0OV 信号。 - 当TQ0OVF 位 = 1时即使读TQ0OVF 位或 TQ0OPT0 寄存器TQ0OVF 位不清零。 - INTTQ0OV信号产生后TQ0OVF 位清零前, 确认TQ0OVF 位置1. - TQ0OVF 位能读和写, 但TQ0OVF 位不能软件置1. 写1对TMQ0操作没有影响。	

- 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 位清零。

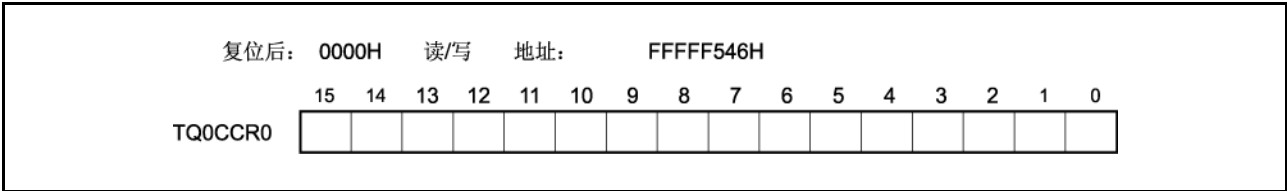
备注 m = 0 至 3

(7) TMQ0 捕捉 / 比较寄存器 0 (TQ0CCR0)

根据选择模式的不同，TQ0CCR0 寄存器可用作捕捉寄存器或比较寄存器。
该寄存器只有在自由运行模式下根据 TQ0OPT0.TQ0CCS0 位设置的不同，可以用作捕捉寄存器或比较寄存器。
在脉宽测量模式下，TQ0CCR0 寄存器只能用作捕捉寄存器。在其它模式下，该寄存器只能用作比较寄存器。
TQ0CCR0 寄存器可以在操作过程中进行读写。
可对该寄存器进行 16 位的读写操作。
复位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。详细内容请参阅 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。

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



(a) 作为比较寄存器时的功能

TQ0CCR0 寄存器可以在 TQ0CTL0.TQ0CE 位 = 1 时被重新写入。
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 引脚输出的电平将反转。
当 TQ0CCR0 寄存器在间隔定时器模式，外部事件计数模式，外部触发脉冲输出模式，单脉冲输出模式或 PWM 输出模式下被用作周期寄存器时，如果 16 位计数器的计数值与 CCR0 缓冲寄存器中的数值相匹配，那么 16 位计数器将被清零（0000H）。
当 TQ0CTL0.TQ0CE 位=0 时不清除比较寄存器。

<R>

(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。
当 TQ0CTL0.TQ0CE 位=0 时不清除捕捉寄存器。

<R>

下表中列出了捕捉 / 比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 8-2. 捕捉 / 比较寄存器在各模式下的功能以及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	无

<R>

<R>

<R>

注 通过写 TQ0CCR1 寄存器触发。

<R>

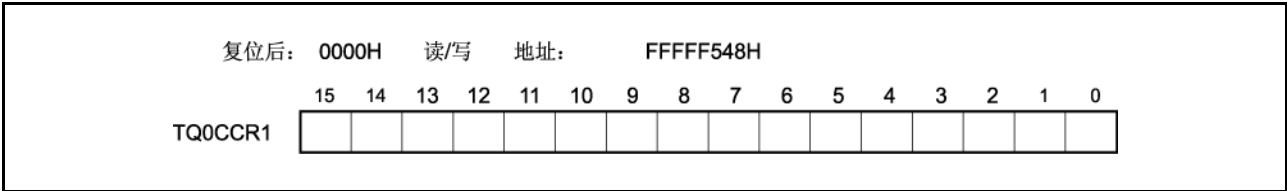
备注 对随时写入和同时写入，参见 8.6（2）随时写入和同时写入。

(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。详细内容请参阅 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。

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



(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。
当 TQ0CTL0.TQ0CE 位=0 时不清除比较寄存器。

(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。
当 TQ0CTL0.TQ0CE 位=0 时不清除捕捉寄存器。

下表中列出了捕捉 / 比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 8-3. 捕捉 / 比较寄存器在各模式下的功能以及对比较寄存器的写入方法

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

注 通过写 TQ0CCR1 寄存器触发。

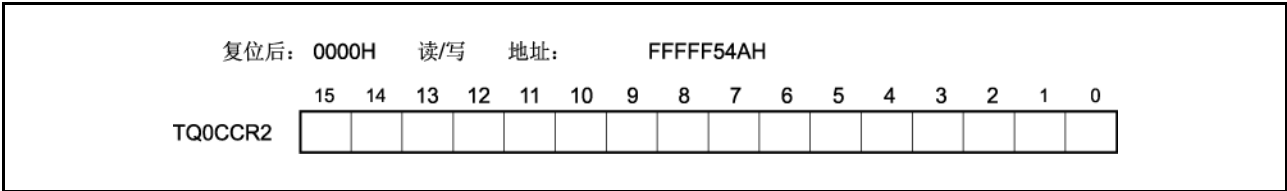
备注 对随时写入和同时写入，参见 8.6（2）随时写入和同时写入。

(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。详细内容请参阅 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。

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



(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。
当 TQ0CTL0.TQ0CE 位=0 时不清除比较寄存器。

(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 TQ0CCR2 寄存器的操作产生冲突，仍可正确读出 TQ0CCR2 寄存器中的值。
当 TQ0CTL0.TQ0CE 位=0 时不清除捕捉寄存器。

下表中列出了捕捉 / 比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 8-4. 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	无

注 通过写 TQ0CCR1 寄存器触发。

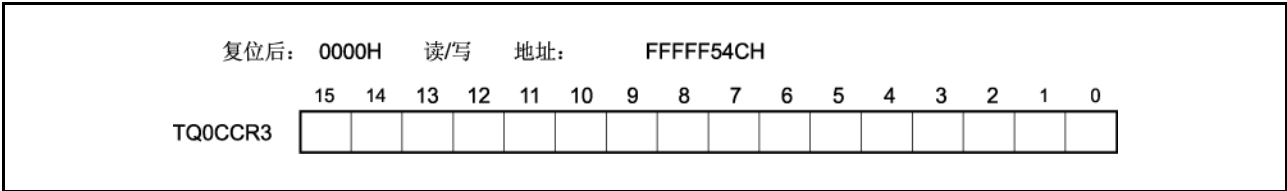
备注 对随时写入和同时写入，参见 8.6（2）随时写入和同时写入。

(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。详细内容请参阅 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。

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



(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。
当 TQ0CTL0.TQ0CE 位=0 时不清除比较寄存器。

<R>

(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。
当 TQ0CTL0.TQ0CE 位=0 时不清除捕捉寄存器。

<R>

下表中列出了捕捉 / 比较寄存器在各模式下的功能以及对比较寄存器的写入方法。

表 8-5. 捕捉 / 比较寄存器在各模式下的功能以及对比较寄存器的写入方法

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

<R>

<R>

<R>

注 通过写 TQ0CCR1 寄存器触发。

<R>

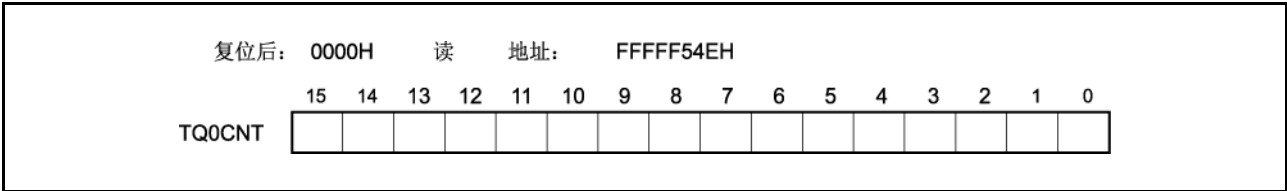
备注 对随时写入和同时写入，参见 8.6 (2) 随时写入和同时写入。

(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。详细内容请参阅 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。

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



8.5 定时器输出操作

TOQ00 和 TOQ01 引脚的工作模式与输出电平的关系如下表所示。

表 8-6. 各模式下的定时器输出控制

工作模式	TOQn00 引脚	TOQn01 引脚	TOQn02 引脚	TOQn03 引脚
间隔定时器模式	方波输出			
外部事件计数模式	无			
外部触发脉冲输出模式	PWM 输出	外部触发脉冲输出		
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）			
脉冲宽度测量模式	无			

表 8-7. TOQ00 至 TOQ03 引脚在定时器输出控制位控制下的真值表

TQ0IOC0.TQ0OLm 位	TQ0IOC0.TQ0OEm 位	TQ0CTL0.TQ0CE 位	TOQ0m 引脚的电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前为低电平，计数过程中为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平，计数过程中为低电平

备注 m = 0 至 3

8.6 操作

TMQ0 可进行以下操作。

操作	TQ0CTL1.TQ0EST 位 (软件触发位)	TIQ00 引脚 (外部触发输入)	捕捉 / 比较寄存器 设 置	比较寄存器写入
间隔定时器模式	无效	无效	仅比较	可随时写入
外部事件计数模式 ^{注 1}	无效	无效	仅比较	可随时写入
外部触发脉冲输出模式 ^{注 2}	有效	有效	仅比较	同时写入
单脉冲输出模式 ^{注 2}	有效	有效	仅比较	可随时写入
PWM 输出模式	无效	无效	仅比较	同时写入
自由运行定时器模式	无效	无效	可选	可随时写入
脉宽测量模式 ^{注 2}	无效	无效	仅捕捉	不可操作

- 注**
1. 使用外部事件计数模式时，请指定未检测到的 TIQ00 引脚捕捉触发输入的有效沿（通过将 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位清零）。
 2. 为使用外部触发脉冲输出模式、单脉冲输出模式或脉宽测量模式，选择内部时钟（设置 TQ0CTL.TQ0EEE 位=0）作为计数时钟。

<R> (1) 计数基本操作

本节说明 16 位计数器的基本操作。对于细节，参见各个模式下操作的说明。

(a) 计数开始操作

所有模式下 TMQ0 的 16 位计数器从默认值 FFFFH 开始计数。
它从 FFFFH 向上到 0000H、0001H、0002H、0003H...计数。

(b) 清除操作

当 16 位计数器的值与比较器的值匹配和捕获时清零。在计数器开始计数后或计数器溢出时从 FFFFH 到 0000H 的计数操作不是清零操作。因此，不会产生 INTTQ0CCm 中断信号（m=0 到 3）。

(c) 溢出操作

在自由运行定时器模式或脉宽测量模式下计数器计数从 FFFFH 到 0000H 时，16 位计数器溢出。如果计数器溢出，TQ0OPT0.TQ0OVF 位置 1，产生中断请求信号（INTTQ0OV）。注意在下面情况下不产生 INTTQ0OV 信号。

- 计数操作刚刚开始后
- 计数值与比较值 FFFFH 匹配被清零
- 在脉宽测量模式下捕捉到 FFFFH，并且从 FFFFH 计数到 0000H

注意事项 溢出中断请求信号（INTTQ0OV）产生后，确认检查溢出标志（TQ0OVF 位）置 1。

(d) 计数期间计数器读操作

可以通过 TQ0CNT 寄存器来读取 TMQ0 的 16 位计数器的计数值。当 TQ0CTL.TQ0CE 位=1 时, 可以通过读取 TQ0CNT 寄存器来读取 16 位计数器的计数值。当 TQ0CE 位=0 时, 16 位计数器为 FFFFH, TQ0CNT 寄存器为 0000H。

(e) 中断操作

TMQ0 产生以下五种中断请求信号。

- INTTQ0CC0 中断: 此信号作为 CCR0 缓冲寄存器的匹配中断请求信号和 TQ0CCR0 寄存器的捕捉中断请求信号。
- INTTQ0CC1 中断: 此信号作为 CCR1 缓冲寄存器的匹配中断请求信号和 TQ0CCR1 寄存器的捕捉中断请求信号。
- INTTQ0CC2 中断: 此信号作为 CCR2 缓冲寄存器的匹配中断请求信号和 TQ0CCR2 寄存器的捕捉中断请求信号。
- INTTQ0CC3 中断: 此信号作为 CCR3 缓冲寄存器的匹配中断请求信号和 TQ0CCR3 寄存器的捕捉中断请求信号。
- INTTQ0OV 中断: 此信号作为溢出中断请求信号。

<R> (2) 随时写入和同时写入

TMQ0 的 TQ0CCR0 到 TQ0CCR3 寄存器在定时器操作时可被重写 (TQ0CTL0.TQ0CE 位=1)，但是 CCR0 到 CCR3 缓冲寄存器的写入方式 (随时写入、同时写入) 因模式而异。

(a) 随时写入

此模式下，定时器操作期间数据随时从 TQ0CCR0 到 TQ0CCR3 的寄存器传输到从 CCR0 到 CCR3 的缓冲寄存器。

图 8-2. 随时写入基本操作流程

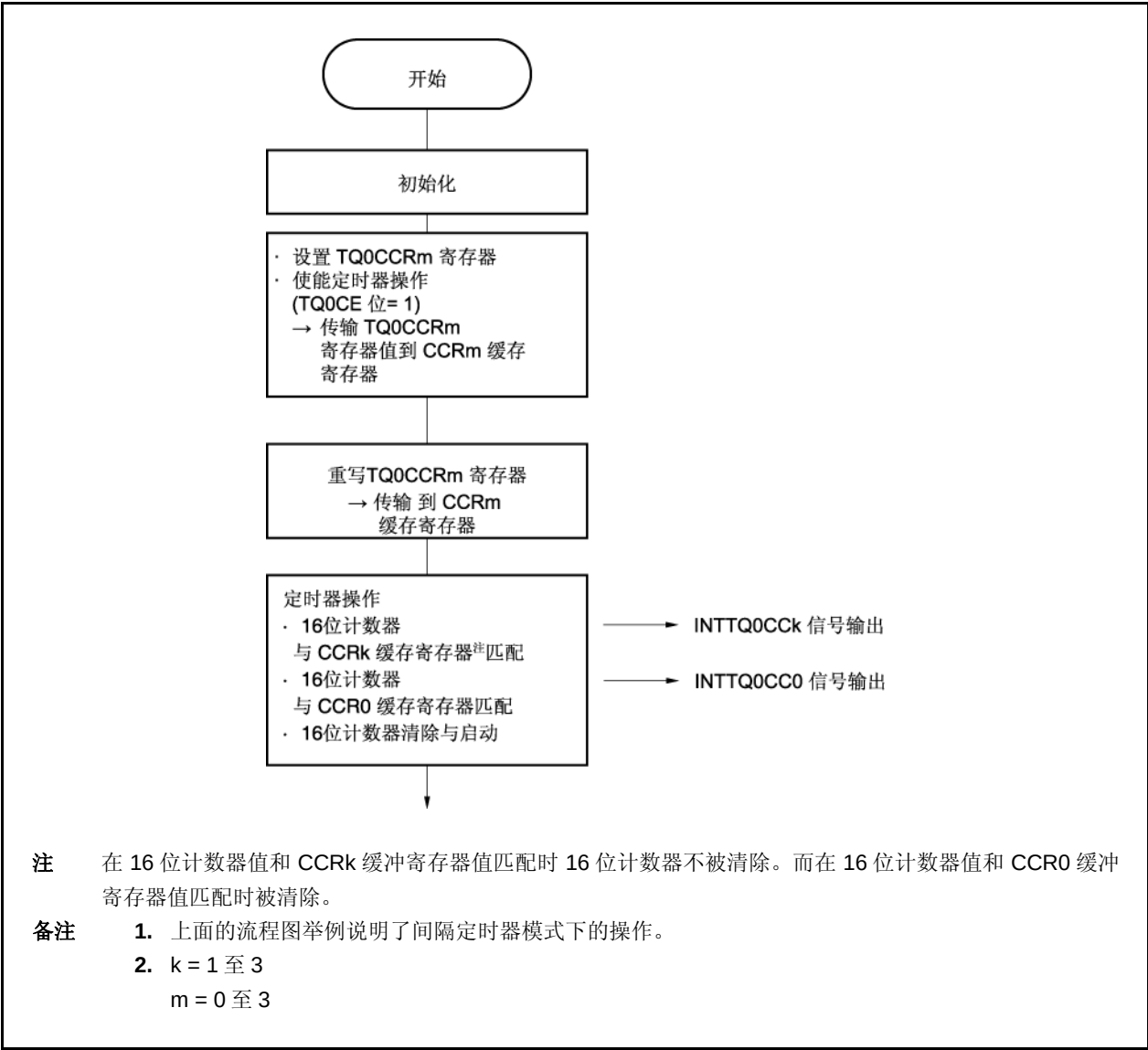
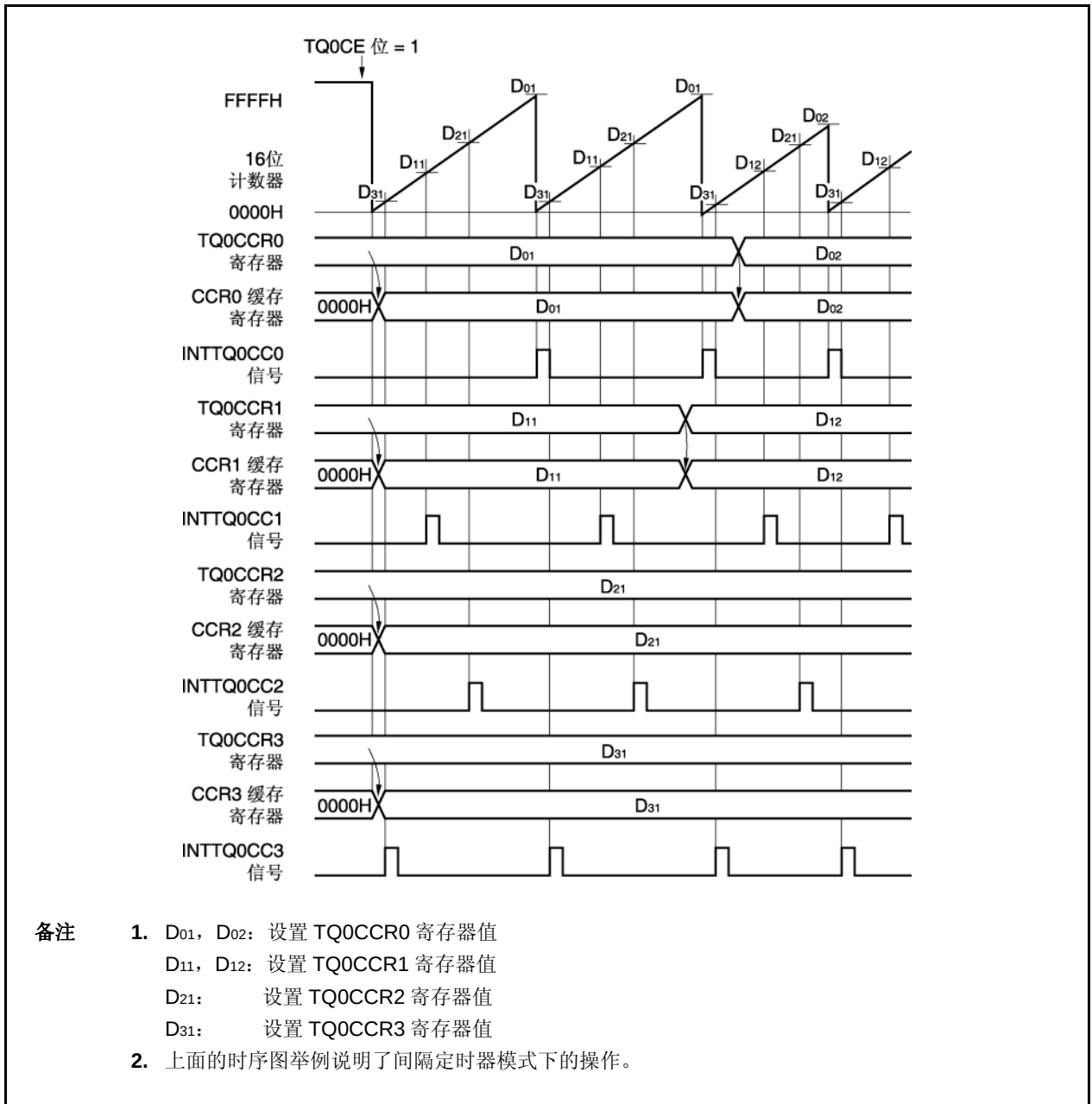


图 8-3. 随时写入时序



(b) 同时写入

此模式下，定时器操作期间数据一次全部从 TQ0CCR0 到 TQ0CCR3 的寄存器传输到从 CCR0 到 CCR3 的缓冲寄存器。在 CCR0 缓冲寄存器值和 16 位计数器值匹配时传输数据。通过写 TQ0CCR1 寄存器使能传输。

使能或禁止下次传输时序被是否写 TQ0CCR1 寄存器控制。

当 TQ0CCR0 到 TQ0CCR3 寄存器的设置值重写为 16 位寄存器比较值（也就是说，这些值传输到 CCR0 到 CCR3 缓冲寄存器），有必要在 16 位计数值和 CCR0 缓冲寄存器值匹配前重写 TQ0CCR0 和 TQ0CCR1 寄存器。在 16 位计数器值和 CCR0 缓冲寄存器值匹配时 TQ0CCR0 到 TQ0CCR3 寄存器的值被传输到 CCR0 到 CCR3 缓冲寄存器。因此，即使仅希望重写 TQ0CCR0、TQ0CCR2 或 TQ0CCR3 寄存器的值，也写相同的值（和 TQ0CCR1 寄存器当前值相同）到 TQ0CCR1 寄存器。

图 8-4. 同时写入基本操作流程

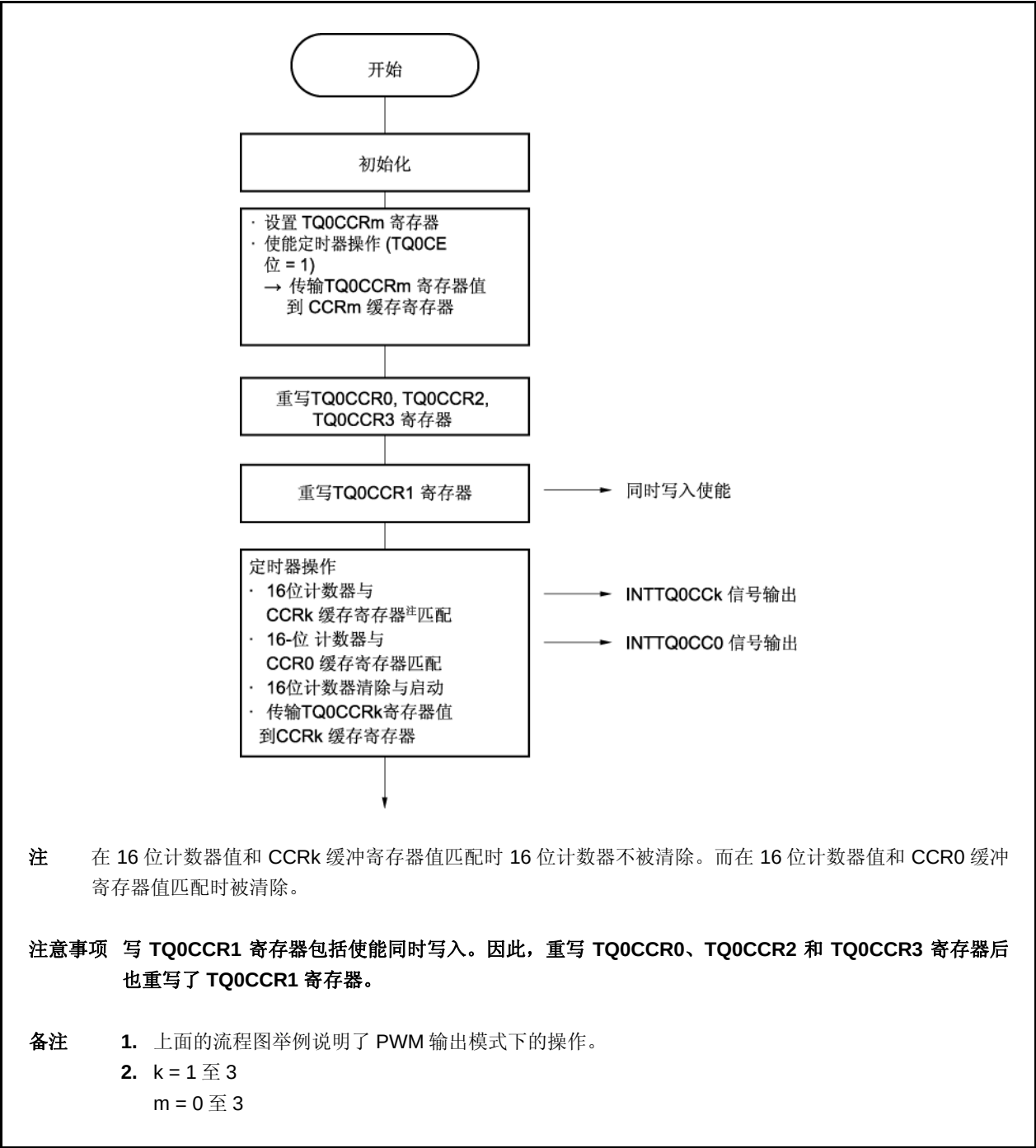
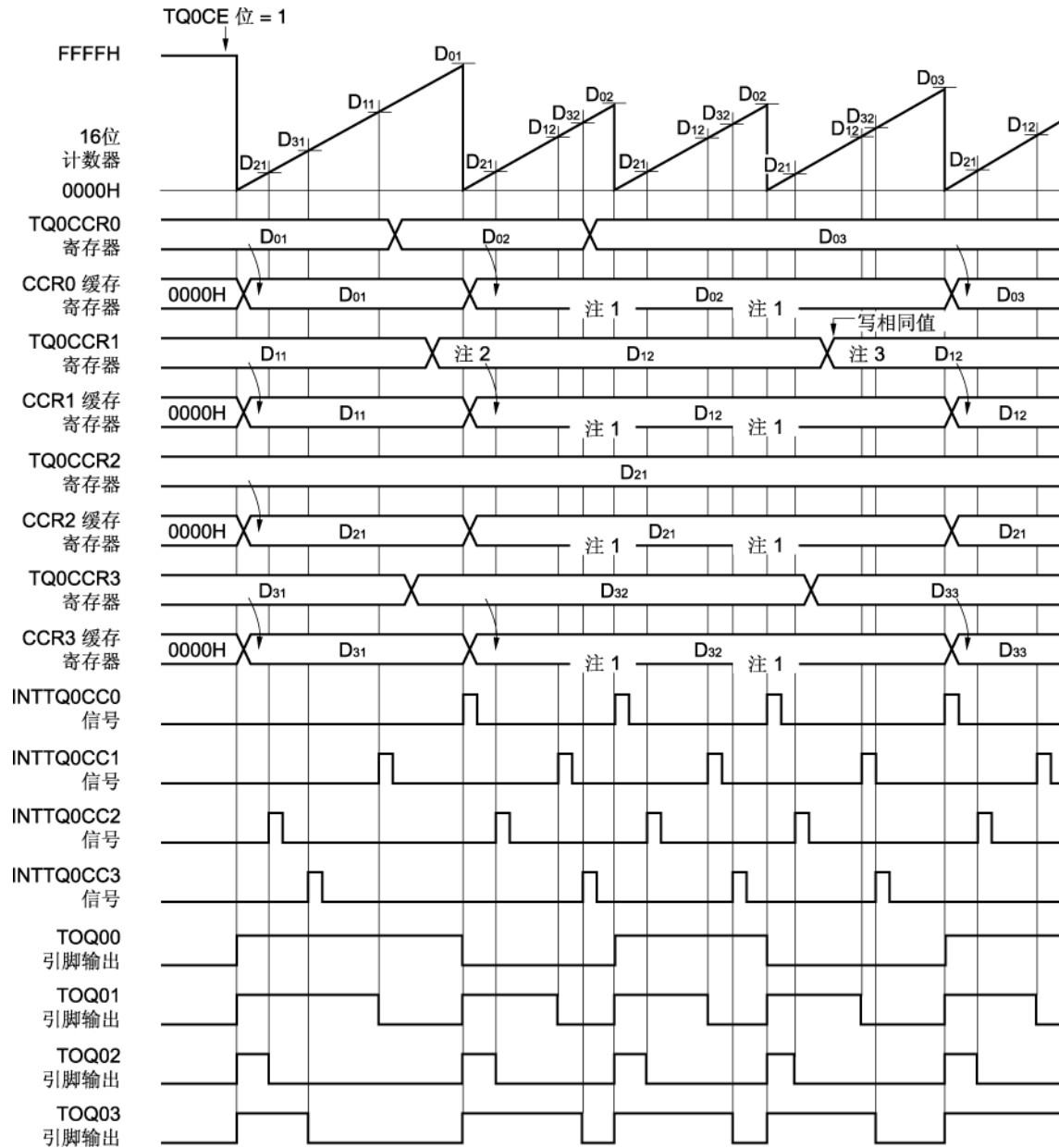


图 8-5. 同时写入时序



- 注
1. 因为没有写 TQ0CCR1 寄存器，没有传输 D02。
 2. 因为写了 TQ0CCR1 寄存器 (D12)，当 16 位定时器的值和 TQ0CCR0 寄存器 (D01) 的值匹配时数据传输到 CCR1 缓冲寄存器。
 3. 因为写了 TQ0CCR1 寄存器 (D12)，当 16 位定时器的值和 TQ0CCR0 寄存器 (D12) 的值匹配时数据传输到 CCR1 缓冲寄存器。

- 备注
1. D01, D02, D03: 设置 TQ0CCR0 寄存器值
D11, D12: 设置 TQ0CCR1 寄存器值
D21: 设置 TQ0CCR2 寄存器值
D31, D32, D33: 设置 TQ0CCR3 寄存器值
 2. 上面的时序图举例说明了 PWM 输出模式下的操作。

<R>

8.6.1 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 TQ0CCR0 寄存器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 倍间隔时间的占空比为 50% 的方波。

间隔定时器模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。然而，当 16 位计数器计数值与 CCR1 到 CCR3 缓冲寄存器的值匹配时，TQ0CCR1 到 TQ0CCR3 寄存器的设置值传输到 CCR1 到 CCR3 缓冲寄存器，产生比较匹配中断请求信号 (INTTQ0CC1 到 INTTQ0CC3)。另外，TOQ01 到 TOQ03 引脚输出占空比为 50% 的方波，当 INTTQ0CC1 到 INTTQ0CC3 信号产生时反转。

即使定时器操作时也能重写 TQ0CCR1 到 TQ0CCR3 寄存器的值。

图 8-6. 间隔定时器的配置图

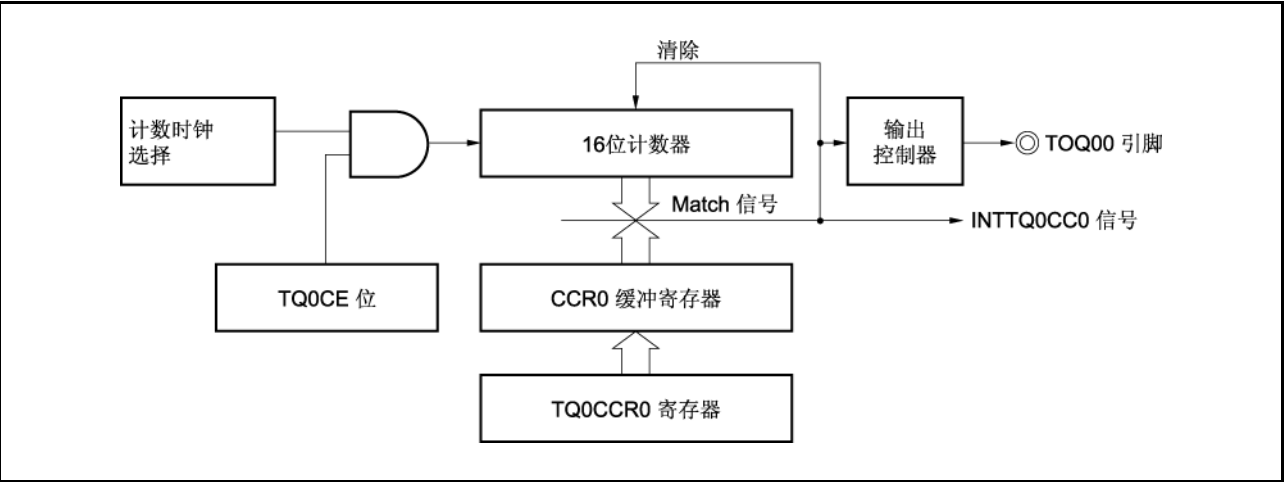
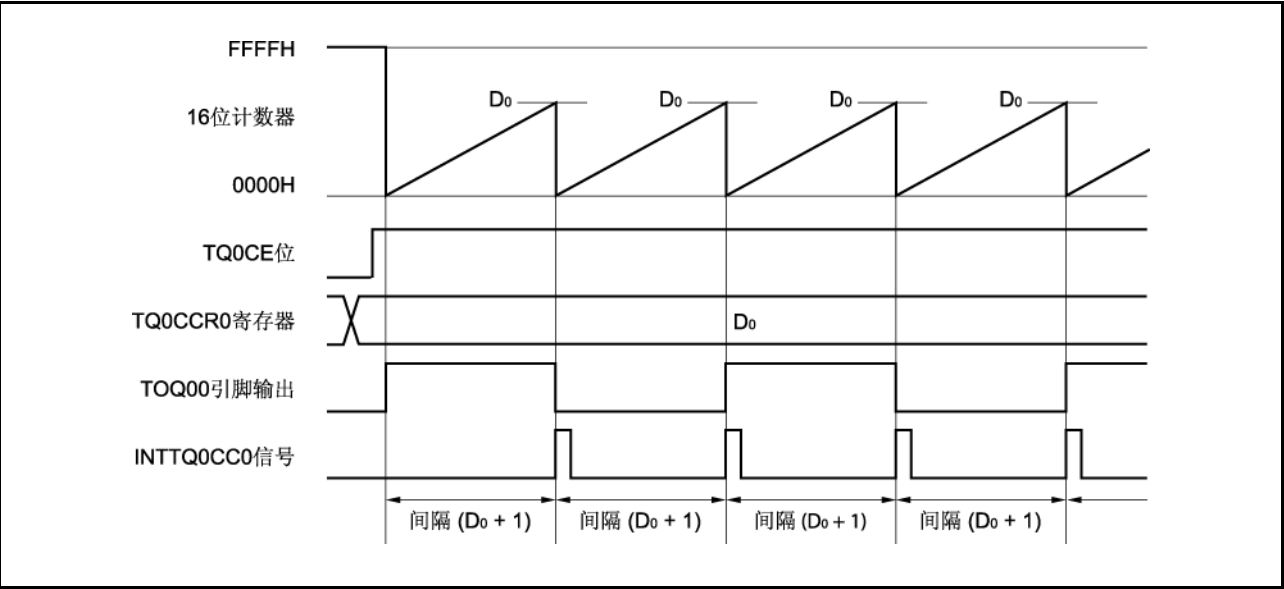


图 8-7. 间隔定时器模式操作的基本时序



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。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 缓冲寄存器中的数值匹配时，16 位计数器清零，TOQ00 引脚输出电平反转并且产生一个比较匹配中断请求信号（INTTQ0CC0）。

间隔时间可由以下算式求得。

间隔时间 = (TPnCCR0 寄存器的设定值 + 1) × 计数时钟周期

图 8-8. 间隔定时器模式操作的寄存器设置 (1/3)

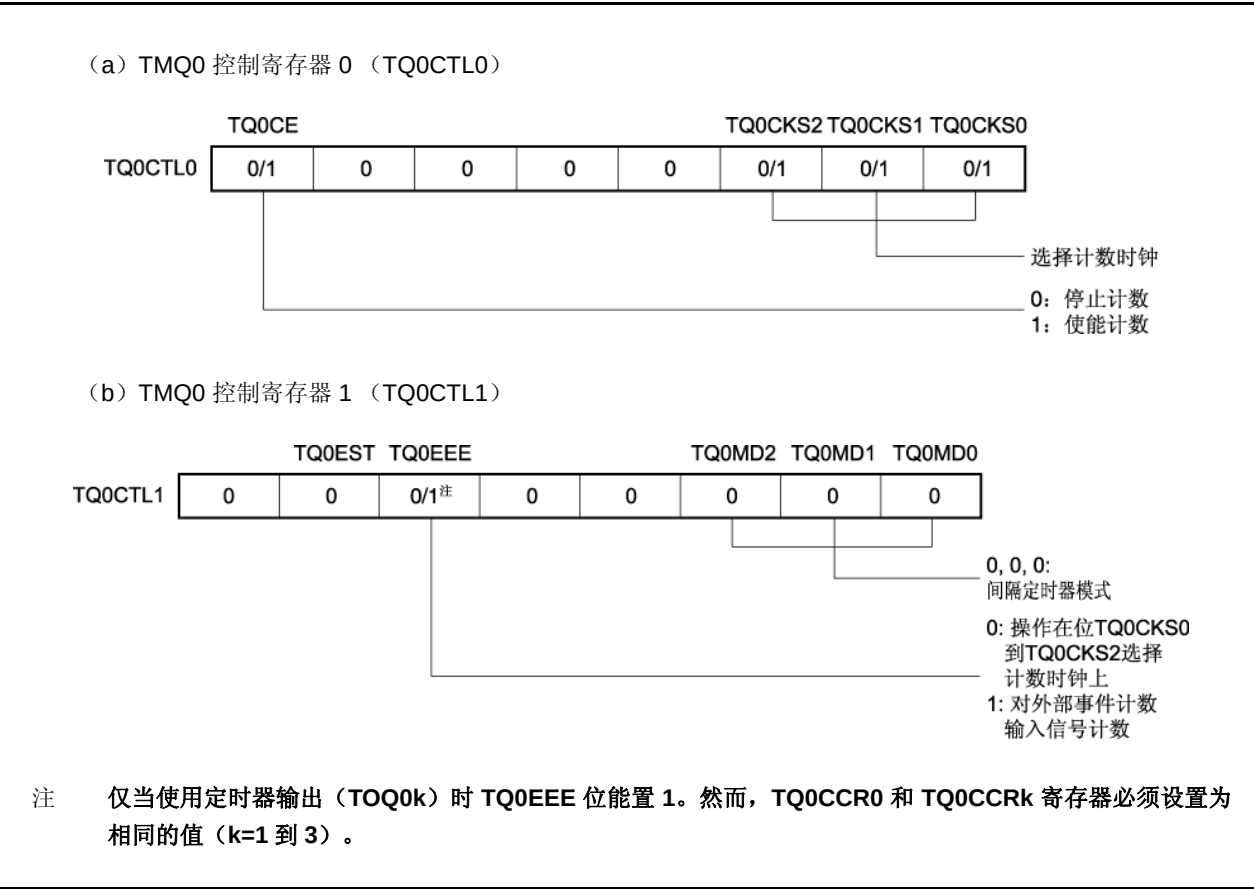
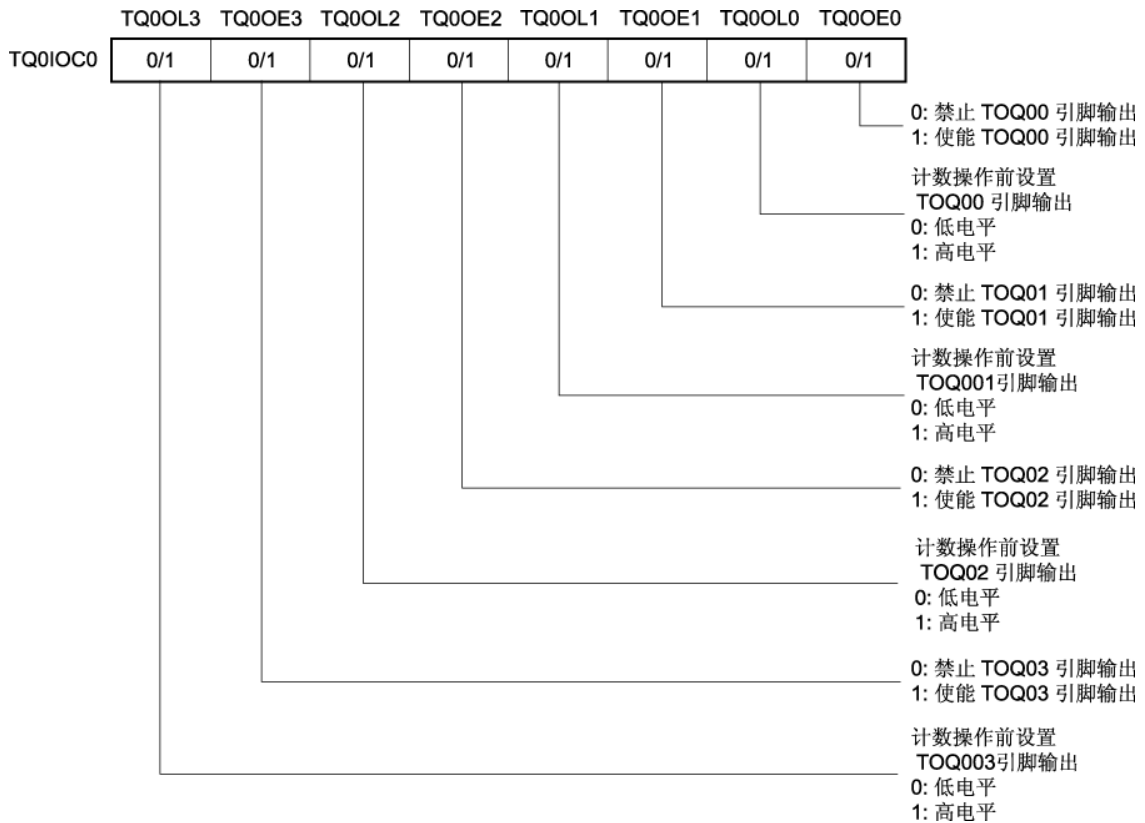
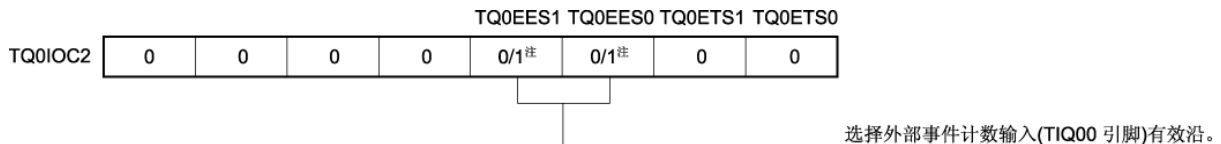


图 8-8. 间隔定时器模式操作的寄存器设置 (2/3)

(c) TMQ0 I/O 控制寄存器 0 (TQ0IOC0)



(d) TMQ0 I/O 控制寄存器 2 (TQ0IOC2)



注 仅当使用定时器输出 (TOQ01 到 TOQ03) 时才能设置 TQ0EES1 和 TQ0EES0 位。而且, 必须将 TQ0CCR0 到 TQ0CCR3 寄存器设置为相同的值。

(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 (TQ0CCR0)

如果将 TQ0CCR0 寄存器设置为 D₀, 那么间隔时间如下。

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

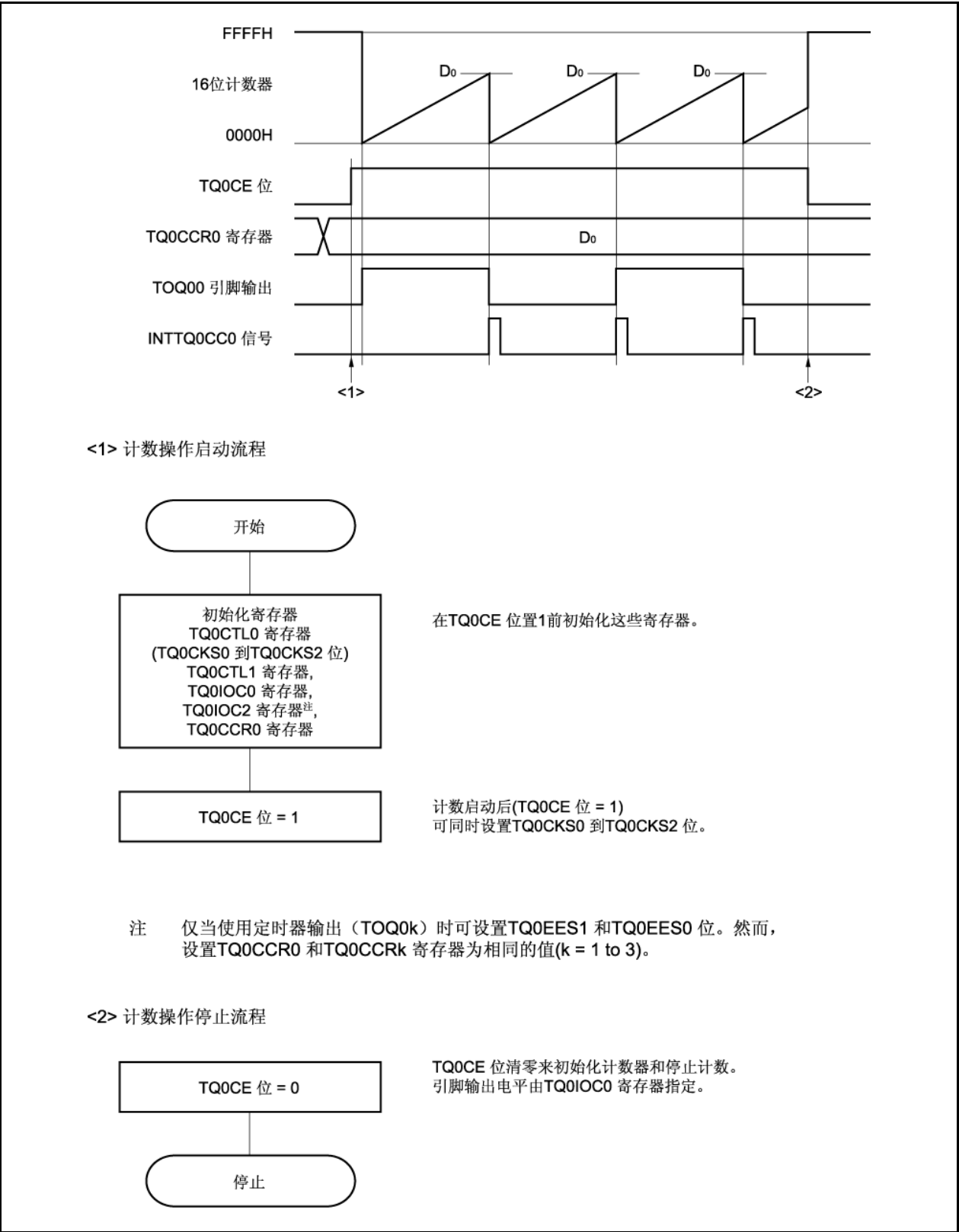
<R>

图 8-8. 间隔定时器模式操作的寄存器设置 (3/3)

<R>	(g) TMQ0 捕捉 / 比较寄存器 1 至 3 (TQ0CCR1 至 TQ0CCR3) 间隔定时器模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。但对 TQ0CCR1 至 TQ0CCR3 寄存器设置的数值也将被传送到 CCR1 至 CCR3 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 至 CCR3 缓冲寄存器中的数值匹配时, TOQ01 到 TOQ03 引脚输出反转, 并产生一个比较匹配中断请求信号 (INTTQ0CC1 至 INTTQ0CCR3)。 当没有使用 TQ0CCR1 到 TQ0CCR3 寄存器时, 推荐设置为 FFFFH。也通过中断屏蔽标志 (TQ0CCIC0.TQ0CCMK0 到 TQ0CCIC3.TQ0CCMK3) 屏蔽这些寄存器。
<R>	备注 间隔定时器输出模式下, 并未使用 TMQ0 I/O 控制寄存器 1 (TQ0IOC1) 和 TMQ0 选项寄存器 0 (TQ0OPT0)。

(1) 间隔定时器模式操作流程

图 8-9. 间隔定时器模式下的软件处理流程



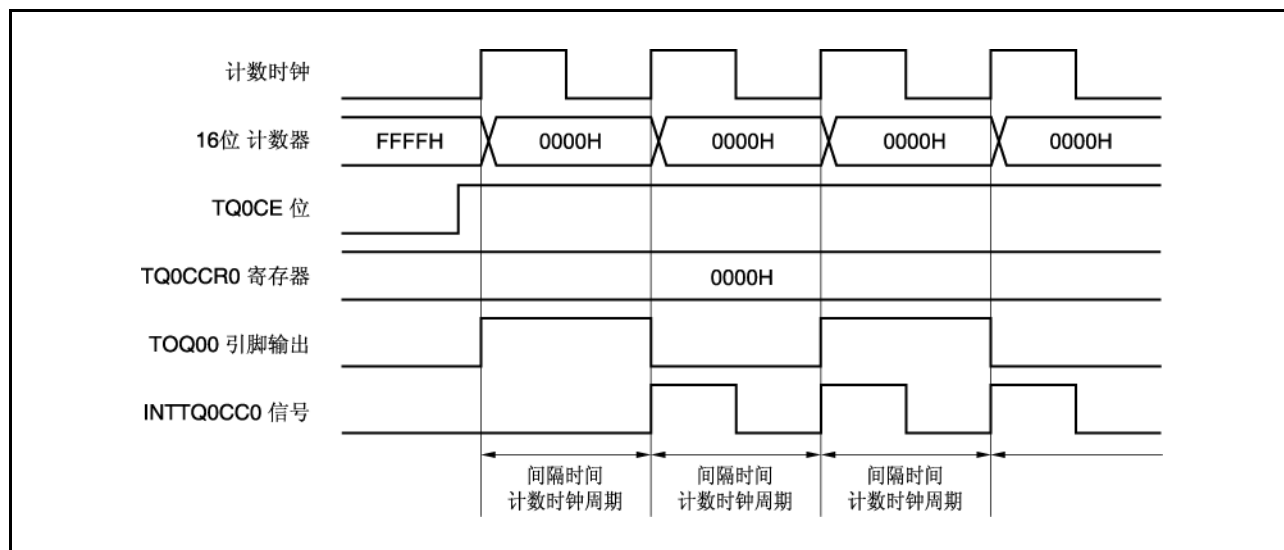
(2) 间隔定时器模式操作时序

<R>

(a) TQ0CCR0 寄存器被设置为 0000H 时的操作

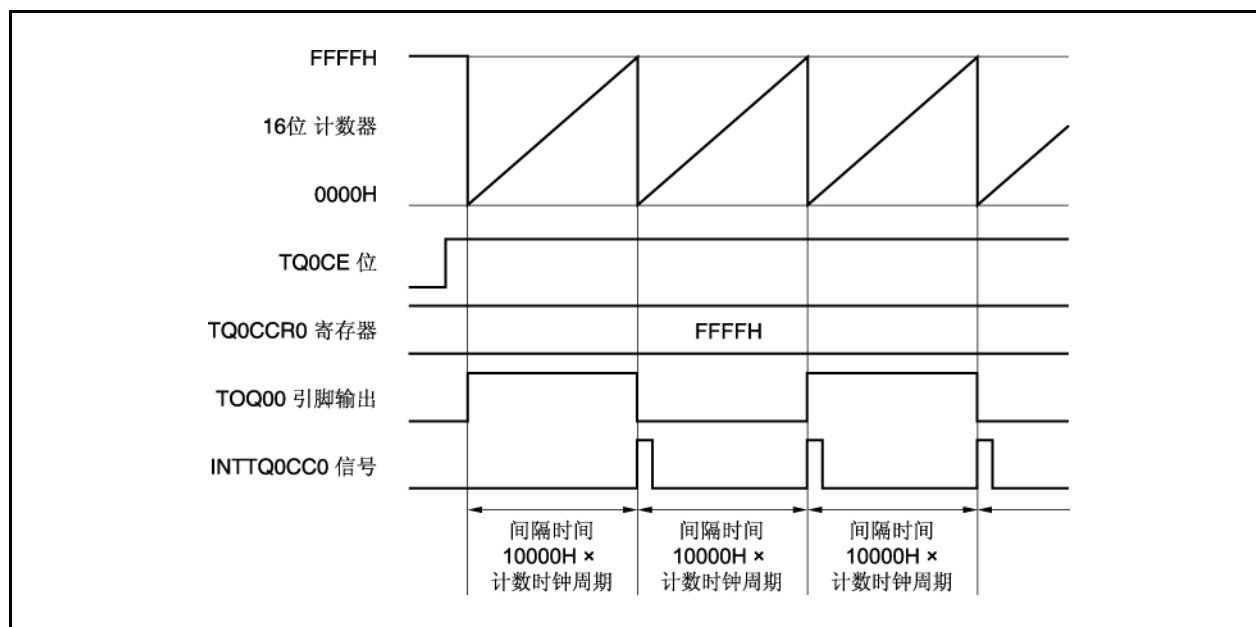
如果 TQ0CCR0 寄存器被设置为 0000H, 那么在每个计数时钟, 都会产生 INTTQ0CC0 信号, 并且 TOQ00 引脚的电平都会反转。

16 位计数器的计数值始终保持为 0000H。



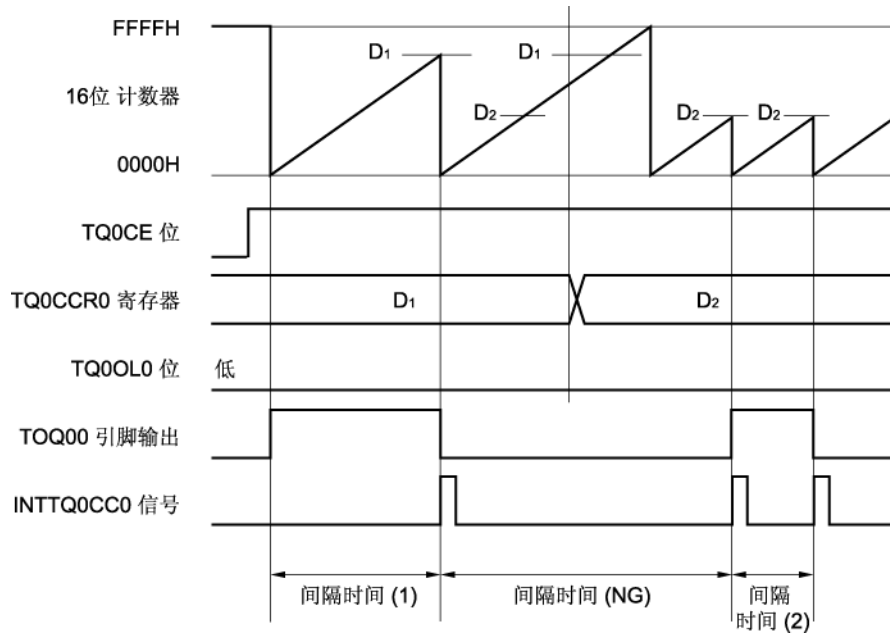
(b) TPnCCR0 寄存器被设置为 FFFFH 时的操作

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

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。如果可能溢出, 停止计数并改变设置值。



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): $(D_2 + 1) \times \text{计数时钟周期}$

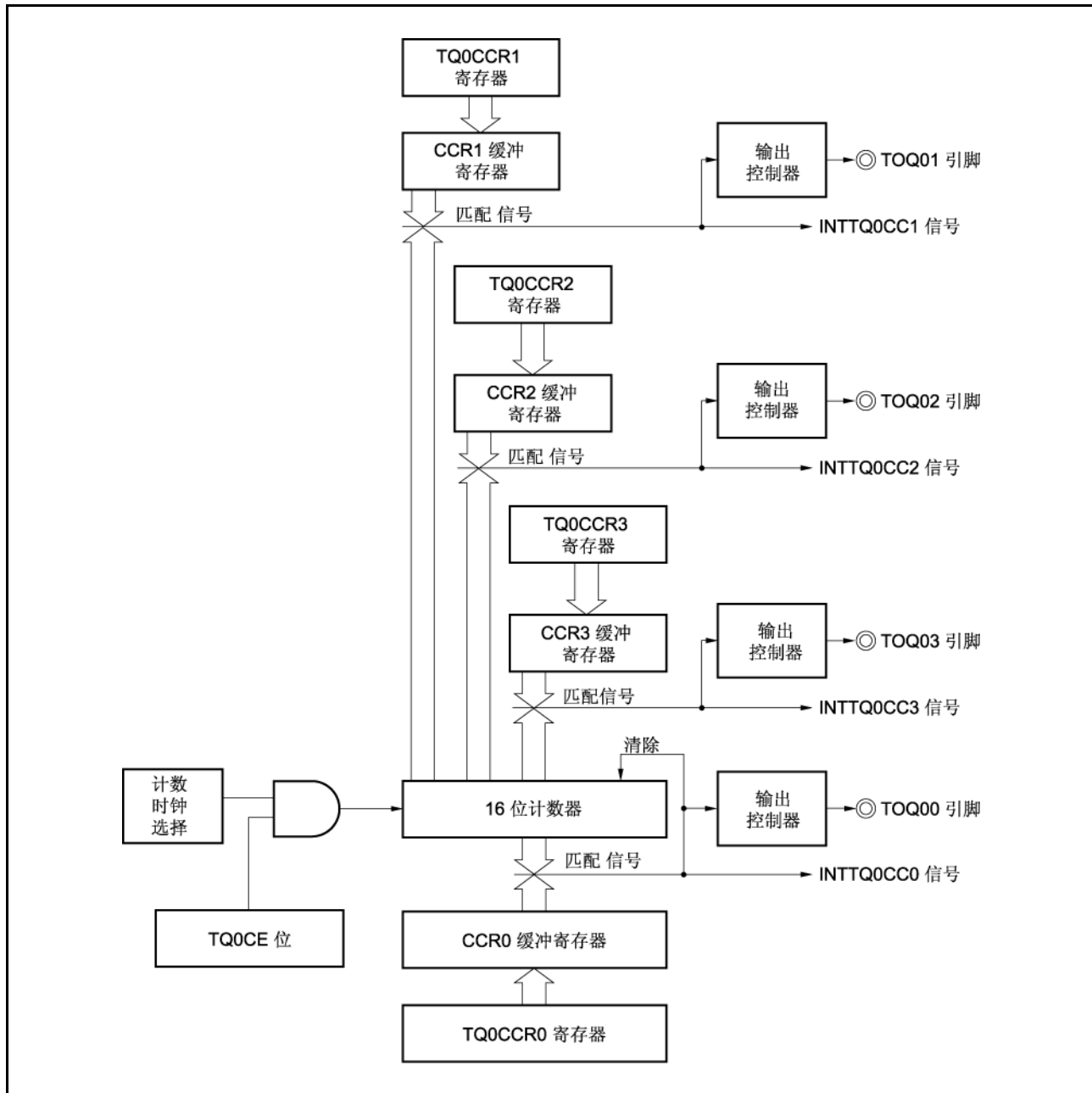
当计数到大于 D_2 小于 D_1 时, 如果 TQ0CCR0 寄存器的值从 D_1 变为 D_2 , 改写 TQ0CCR0 寄存器后, 此值立即传送到 CCR0 缓冲寄存器。因此, 16 位计数器的计数值将与 D_2 进行比较。

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, 当计数值与 D_2 匹配时, 产生 INTTQ0CC0 信号, TQ000 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 “ $(D_2 + 1) \times \text{计数时钟周期}$ ” 的间隔时间内产生 INTTQ0CC0 信号, 而是在 “ $(10000H + D_2 + 1) \times \text{计数时钟周期}$ ” 的间隔时间内产生。

(d) TQ0CCR1 至 TQ0CCR3 寄存器的操作

图 8-10. TQ0CCR1 至 TQ0CCR3 寄存器的构成



<R>

如果 TQ0CCR0 寄存器和 TQ0CCRk 寄存器设置相同的值，INTTQ0CCk 信号将与 INTTQ0CC0 同时产生，TOQ0k 引脚输出反转。这意味着 TOQ0k 引脚输出占空比为 50% 的方波。

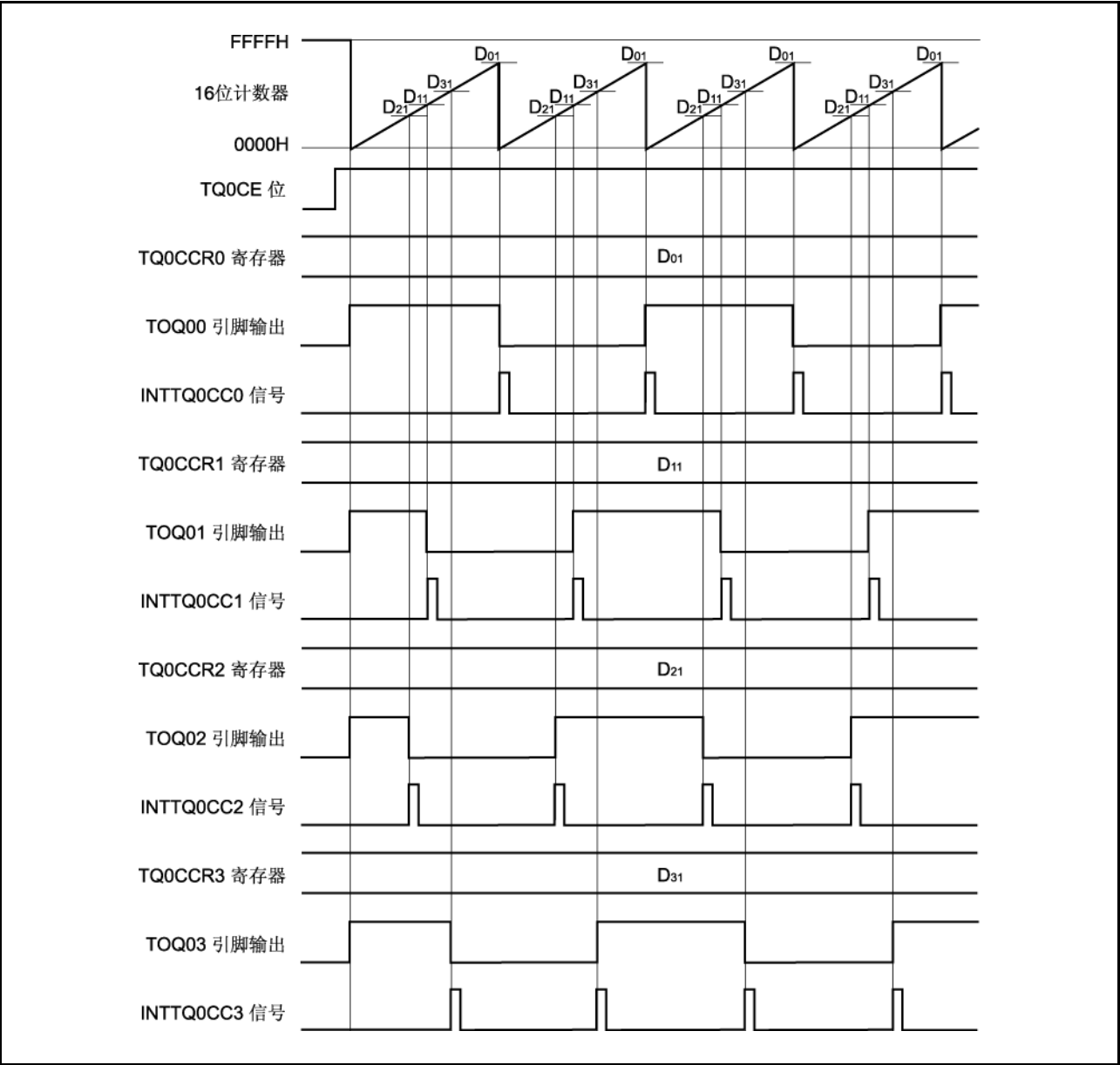
如果 TQ0CCR0 寄存器的值与 TQ0CCRk 寄存器不同，操作如下。

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 引脚在输出一个短脉冲后输出占空比为的方波 50%。

备注 k = 1 至 3

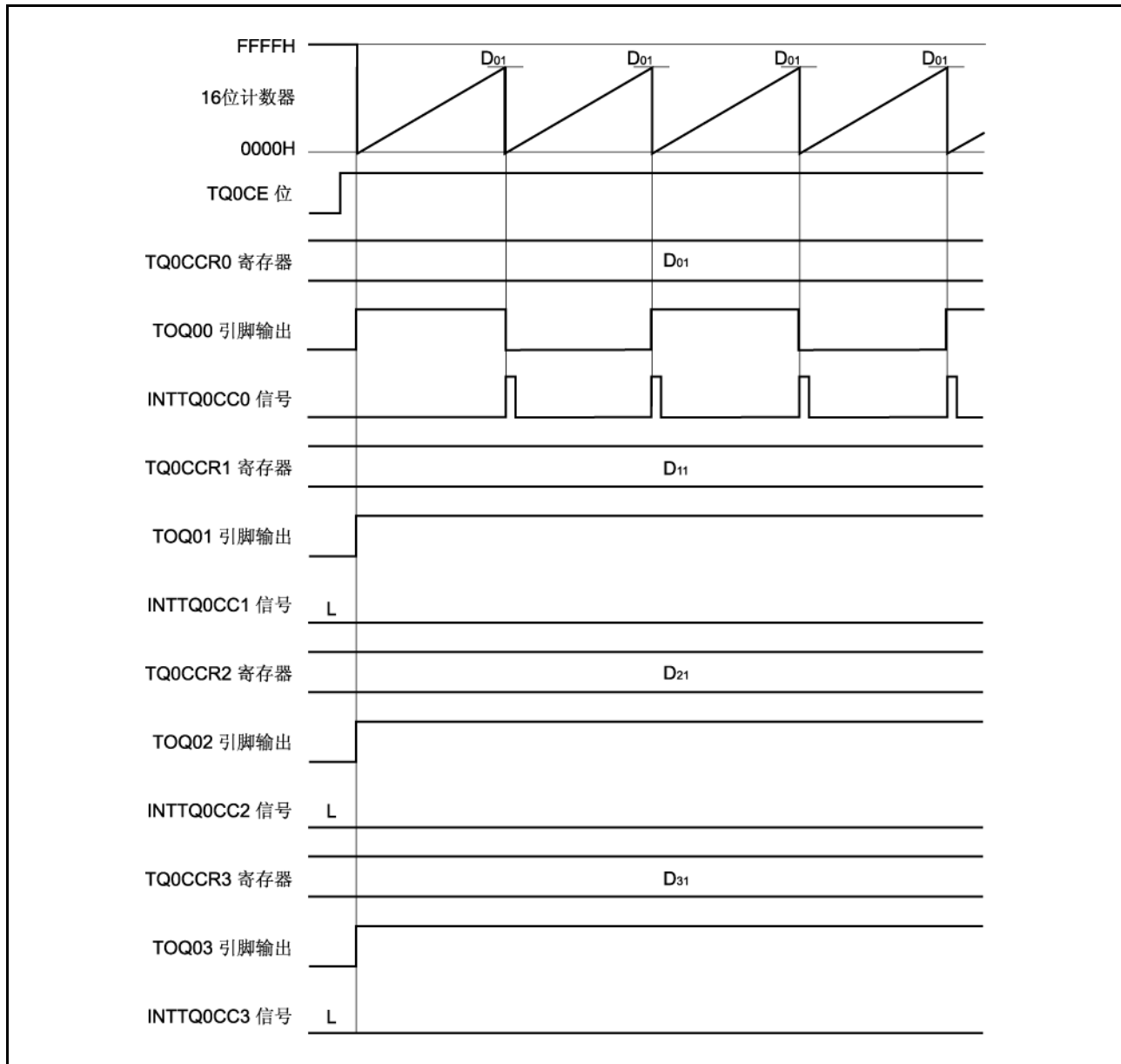
图 8-11. $D_{01} \geq D_{k1}$ 时的时序图



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。当不使用 TQ0CCRk 寄存器时，推荐设置为 FFFFH。

备注 k = 1 至 3

图 8-12. $D_{01} < D_{k1}$ 时的时序图



<R>

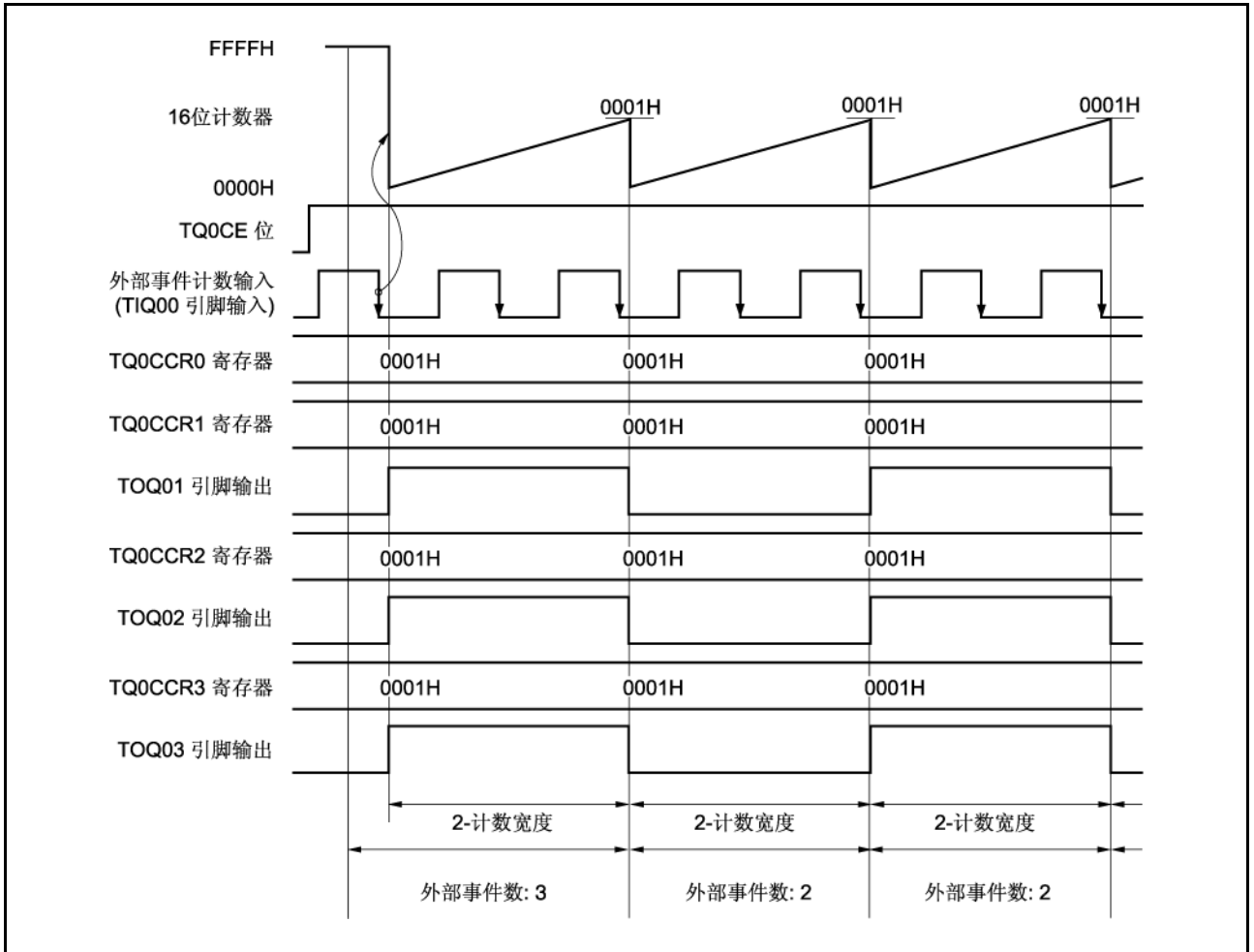
(3) 外部事件计数输入 (TIQ00) 操作

(a) 操作

间隔定时器模式下为在外部事件计数输入 (TIQ00) 的有效沿计数, 由于在 TQ0CE 位从 0 置 1 后 16 位计数器被从 FFFFH 清零, 因此需要外部事件计数输入的有效沿。

当写 001H 到 TQ0CCR0 和 TQ0CCRk 寄存器, TOQ0k 引脚在 16 位计数器计数两次时反转 (k=1 到 3)。

间隔定时器模式下仅当定时器输出 (TOQ0k) 用作外部事件计数输入时 TQ0CTL0.TQ0EEE 可置为 1。



8.6.2 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 (TIQ00) 的有效沿才有效，每次计到 <R> TQ0CCR0 寄存器指定数量的有效沿后都会产生中断请求信号 (INTTQ0CC0)。TOQ00 到 TOQ03 引脚不能使用。间隔定时器模式下使用 TOQ01 和 TOQ03 引脚作为外部事件计数输入时，设置 TQ0CTL1.TQ0EEE 位=1 (参阅 8.6.1 (3) 外部事件计数输入 (TIQ00))。

外部事件计数模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。

<R> 注意事项 外部事件计数模式下，TQ0CCR0 到 TQ0CCR3 寄存器不能清零。

图 8-13. 外部事件计数模式结构

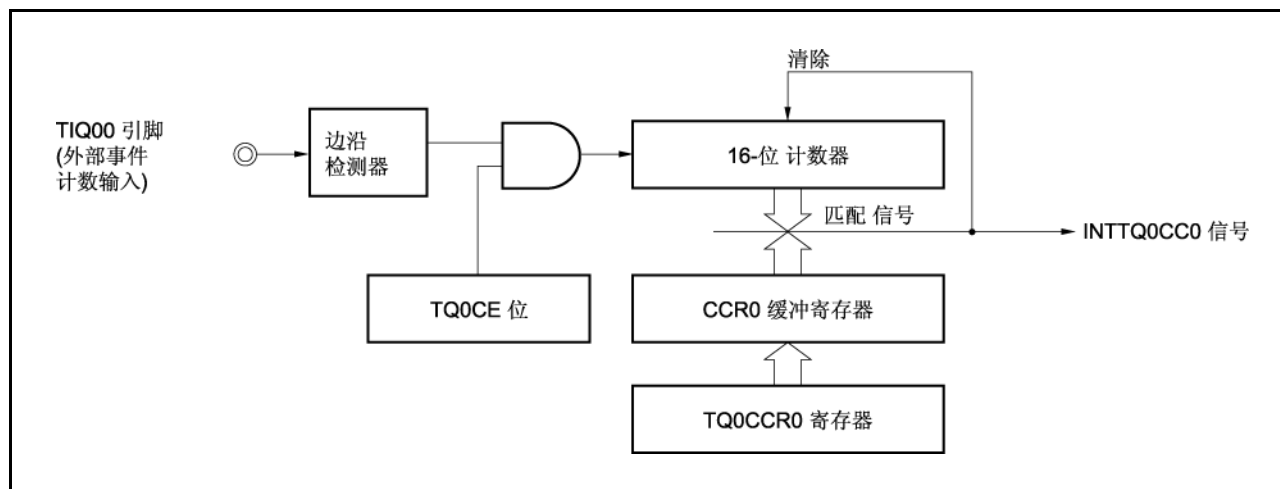
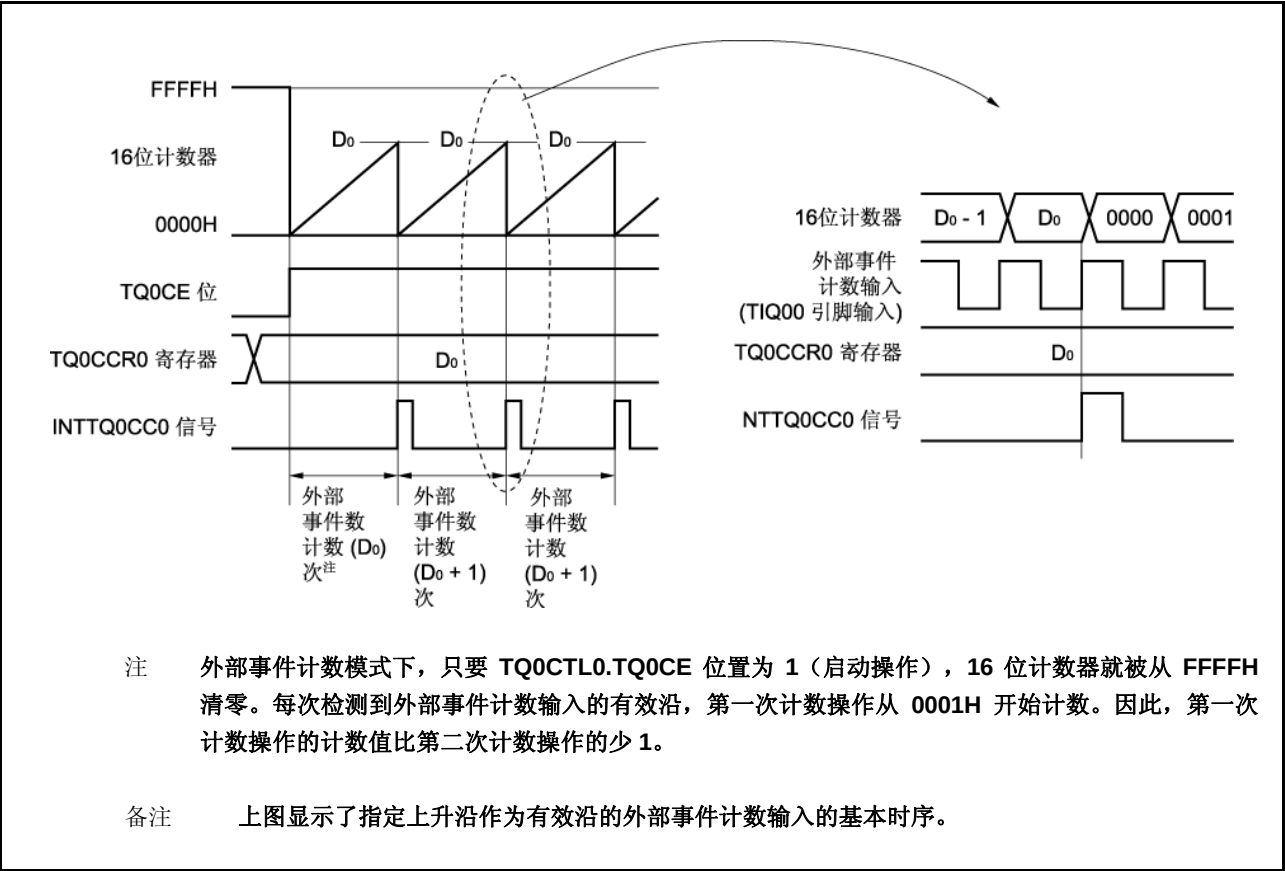


图 8-14. 外部事件计数模式的基本时序



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)。

<R> 当第一次检测到“TQ0CCR0 寄存器的设置值”次数的外部事件计数输入有效沿时，产生 INTTQ0CC0 信号。在这之后，在检测到“TQ0CCR0 寄存器的设置值+1”次数的外部事件计数输入有效沿时，才会产生 INTTQ0CC0 信号。

图 8-15. 外部事件计数模式操作的寄存器设置 (1/2)

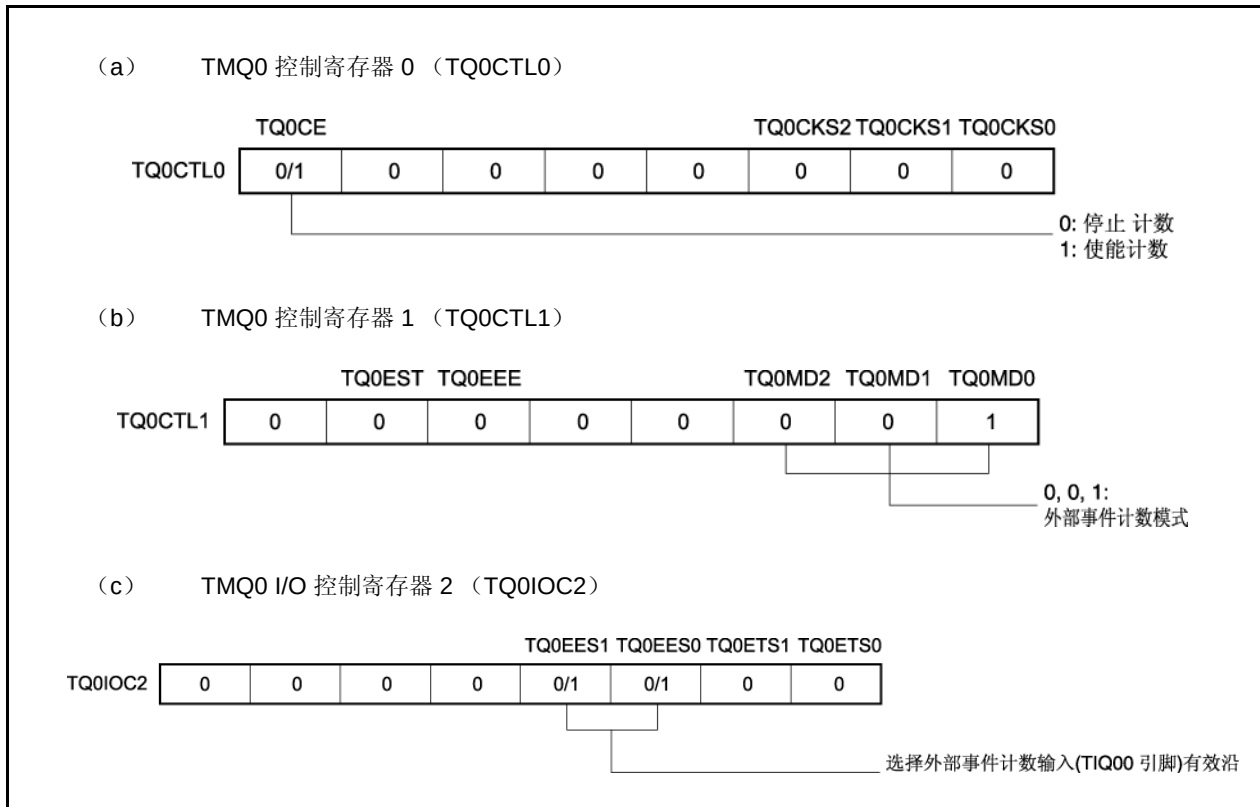


图 8-15. 外部事件计数模式操作的寄存器设置 (2/2)

(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。

<R>

(e) TMQ0 捕捉 / 比较寄存器 0 (TQ0CCR0)

如果 TQ0CCR0 寄存器设置为 D_0 ，当外部事件数达到 (D_0)，计数器清零，产生第一个比较匹配中断请求信号 (INTTQ0CC0)。当外部事件数达到 ($D_0 + 1$) 个，产生第二个比较匹配中断请求信号 (INTTQ0CC0)。

(1)

<R>

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

外部事件计数模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。但对 TQ0CCR1 至 TQ0CCR3 寄存器设置的数值也将被传送到 CCR1 至 CCR3 缓冲寄存器中。当 16 位计数器的计数值与 CCR1 至 CCR3 缓冲寄存器中的数值匹配时，也会产生一个比较匹配中断请求信号 (INTTQ0CC1 至 INTTQ0CC3)。

当没有使用 TQ0CCR1 到 TQ0CCR3 寄存器时，推荐设置为 FFFFH。也通过中断屏蔽标志 (TQ0CCIC1.TQ0CCMK1 到 TQ0CCIC3.TQ0CCMK3) 屏蔽这些寄存器。

注意事项 1. 设置 TQ0IOC0 寄存器为 00H。

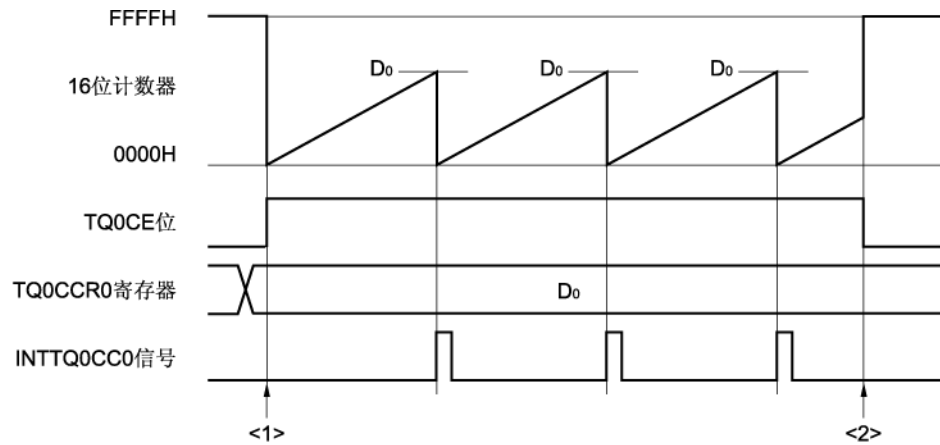
2. 当外部时钟用于计数时钟时，外部时钟只能从 TIQ00 引脚输入。此时要将 TQ0IOC1.TQ0IS1 和 TQ0IS0 位清为 00 (捕捉触发输入 (TIQ00 引脚)：无边沿检测)。

<R>

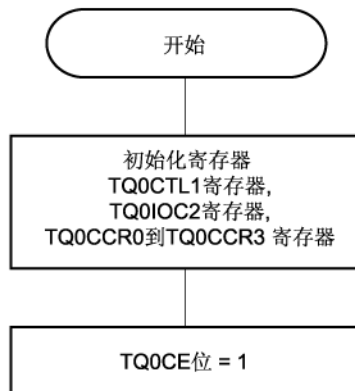
备注 外部事件计数模式下，并未使用 TMQ0 I/O 控制寄存器 1 (TQ0IOC1) 以及 TMQ0 选项寄存器 0 (TQ0OPT0)。

(1) 外部事件计数模式操作流程

图 8-16. 外部事件计数模式的软件处理流程

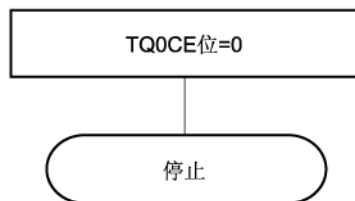


<1> 计数 操作启动 流程



在TQ0CE位置1前初始化这些寄存器。

<2> 计数操作停止流程

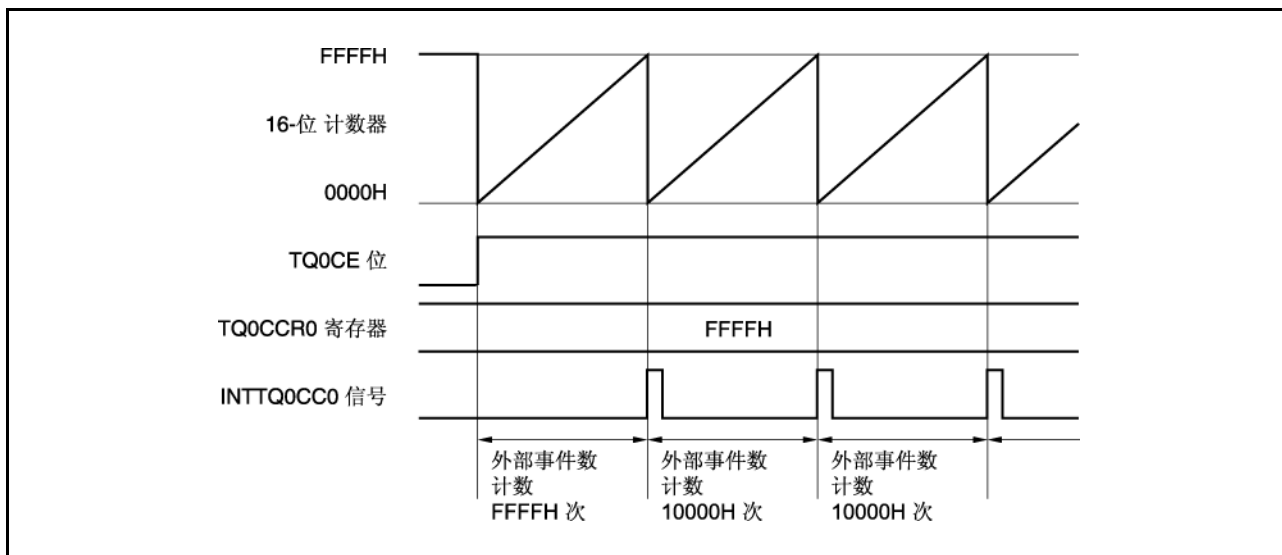
TQ0CE位清零
初始化计数器
和停止计数。

(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 到 TQ0CCR3 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 (TOQ00 到 TOQ03) 功能。如果要将外部事件计数输入 (TIQ00) 作为定时器输出 (TOQ01 to TOQ03) 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.TQ0EEE 位 = 1)。参见 8.6.1 (3) 外部事件计数输入 (TIQ00) 操作。

(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。

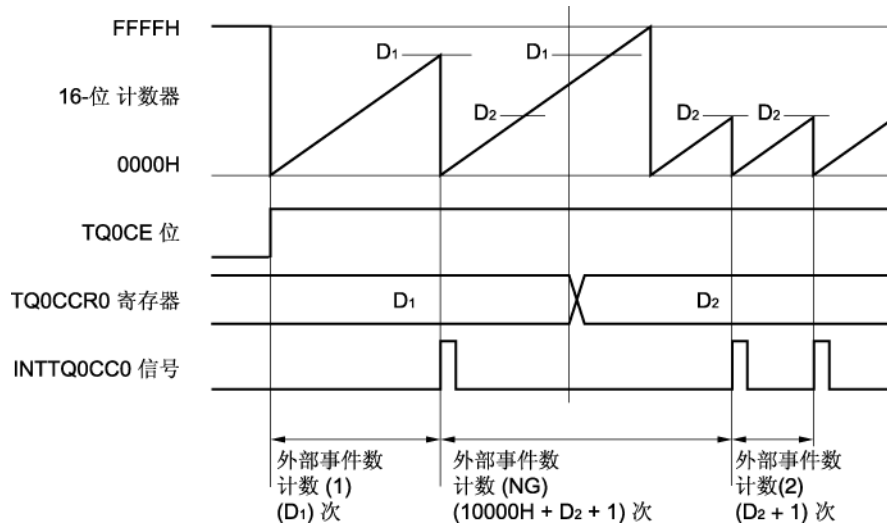


<R>

(b) 改写 TQ0CCR0 寄存器时需要注意的事项

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。如果可能溢出，停止计数并改变设置值。

<R>



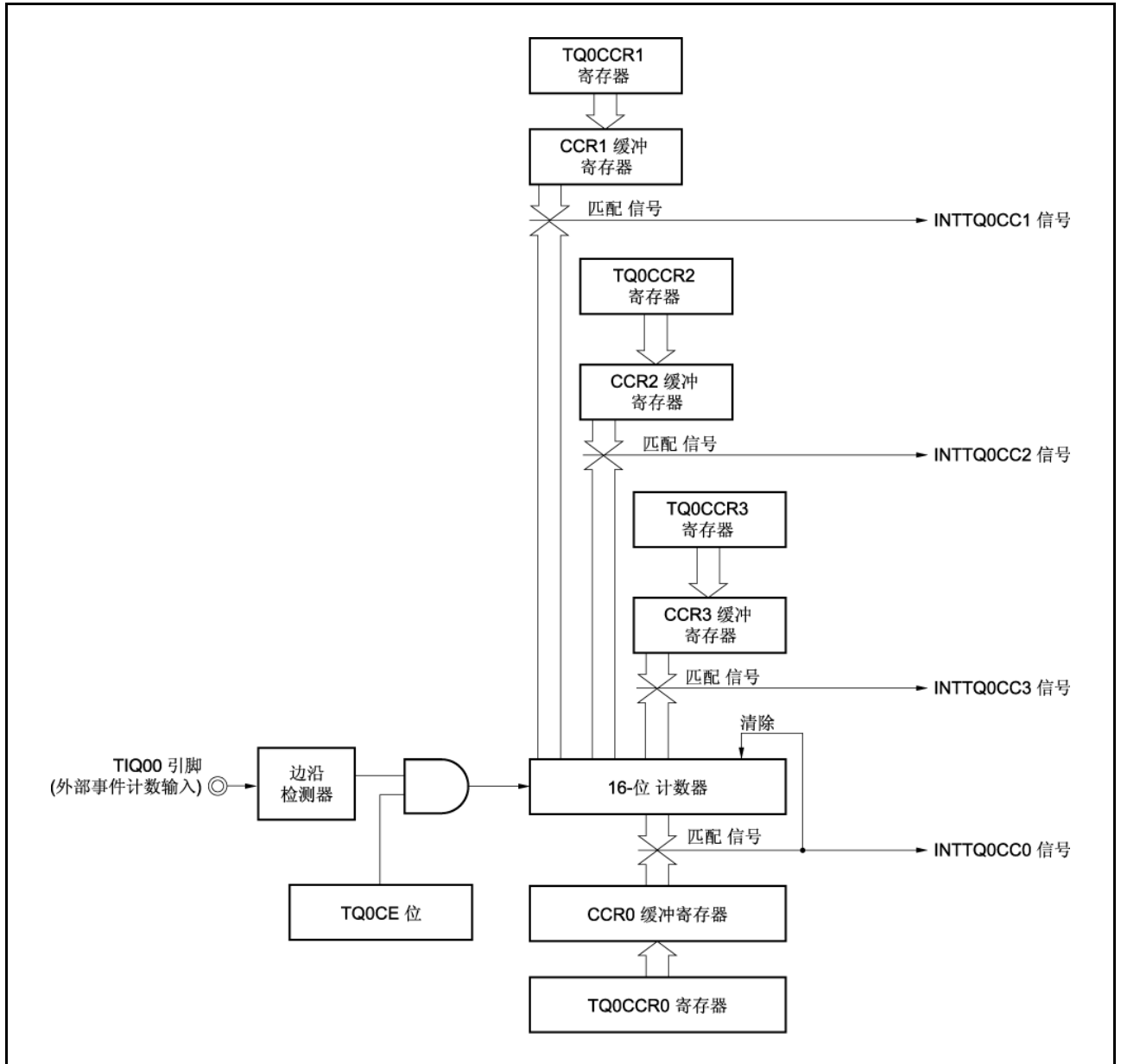
当计数到大于 D2 小于 D1 时如果 TQ0CCR0 寄存器的值从 D1 变为 D2，改写 TQ0CCR0 寄存器后，此值立即传送到 CCR0 缓冲寄存器。因此，16 位计数器的计数值将与 D2 进行比较。

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。

由此可见，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

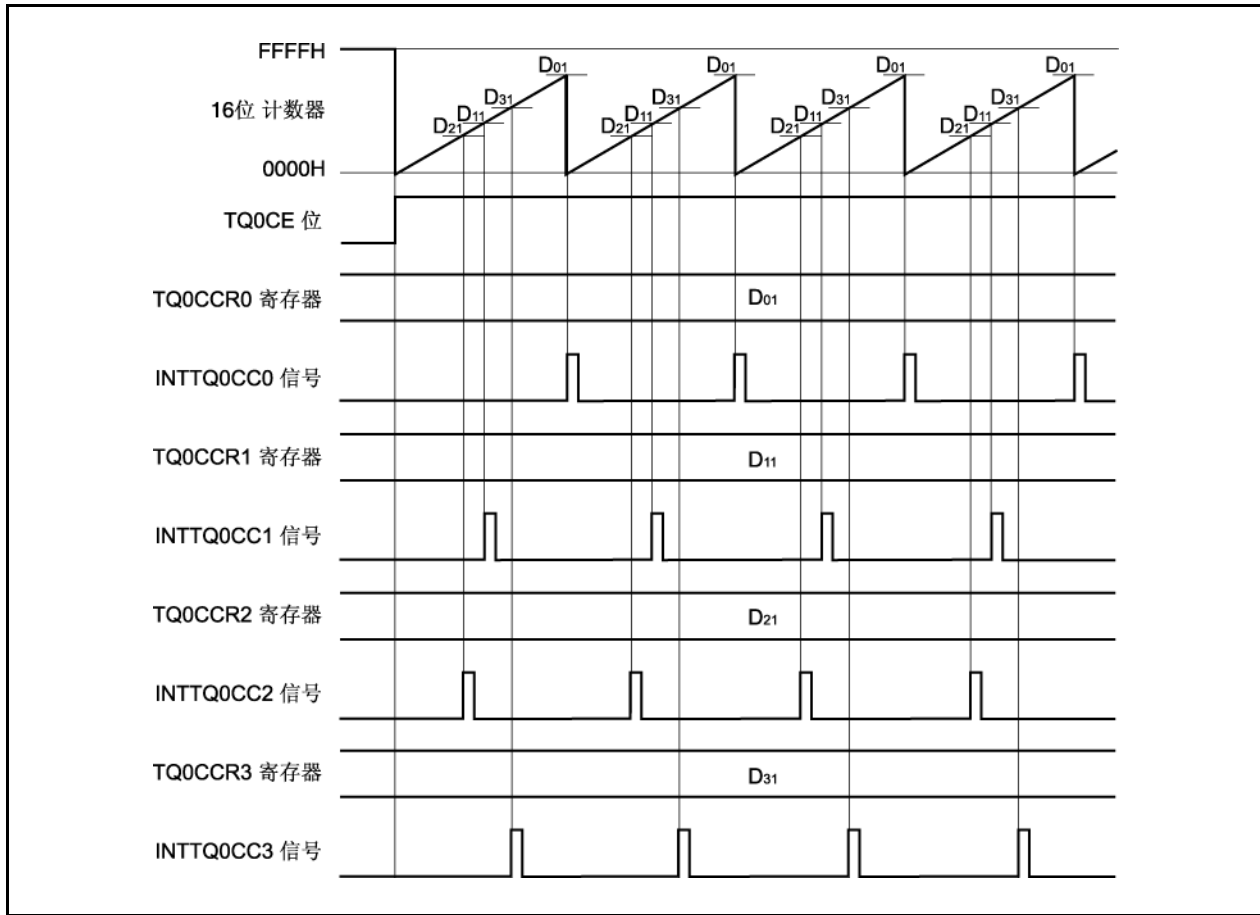
图 8-17. TQ0CCR1 至 TQ0CCR3 寄存器的构成



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 8-18. $D_{01} \geq D_{k1}$ 时的时序图

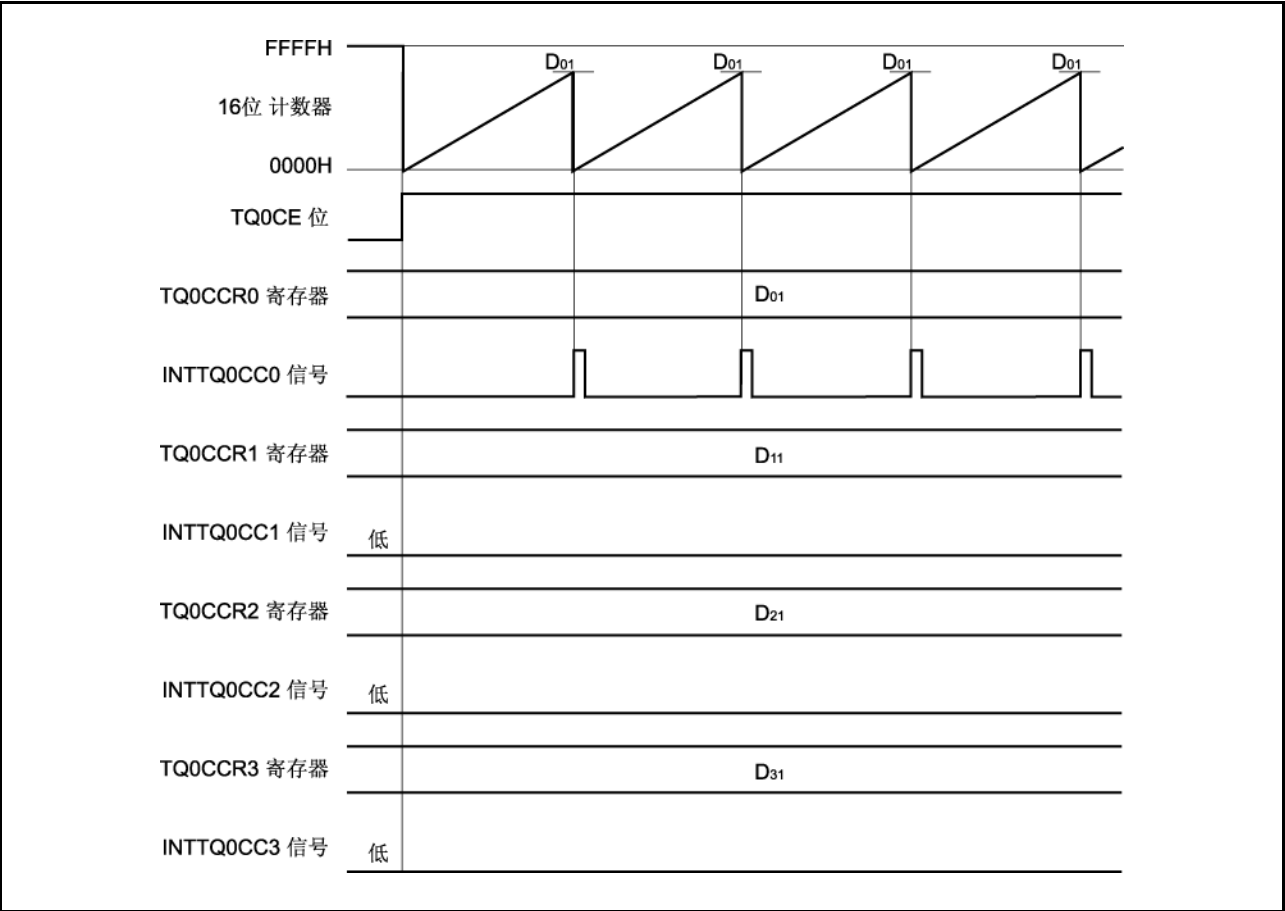


如果 TQ0CCRk 寄存器的值大于 TQ0CCR0 寄存器的值，那么 16 位计数器的计数值不会与 TQ0CCRk 寄存器中的值相匹配。因此，不会产生 INTTQ0CCk 信号。
当不使用 TQ0CCRk 寄存器时，推荐设置为 FFFFH。

<R>

备注 k = 1 至 3

图 8-19. D₀₁ < D_{k1} 时的时序图



8.6.3 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 (TIQ00) 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。

<R> 也可以使用软件触发代替外部触发来开始脉冲的输出。当使用软件触发器时, TOQ00 引脚输出半周期为 (TQ0CCR0 寄存器值+1)、占空比为 50% 的方波。

图 8-20. 外部触发脉冲输出模式的配置

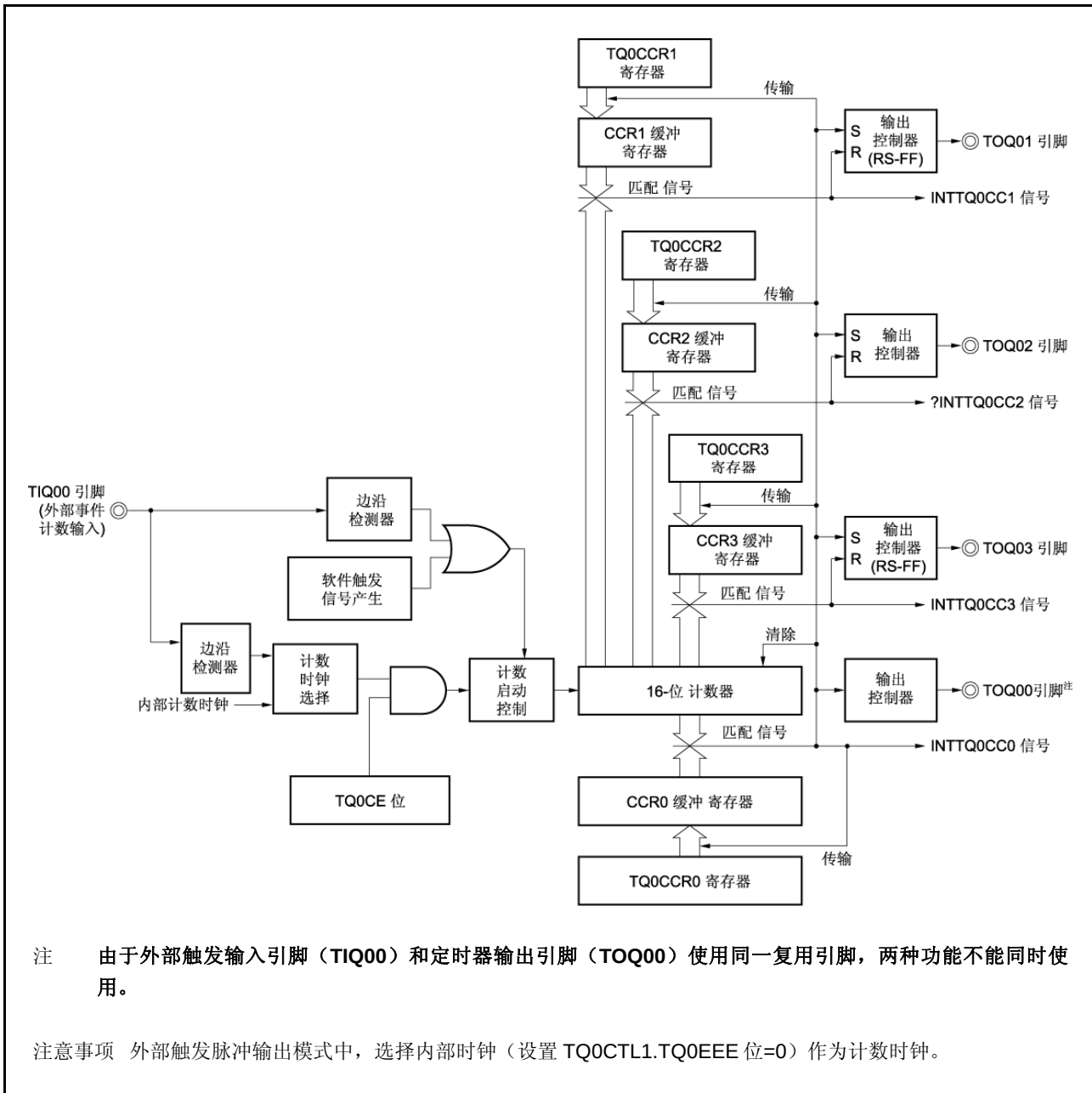
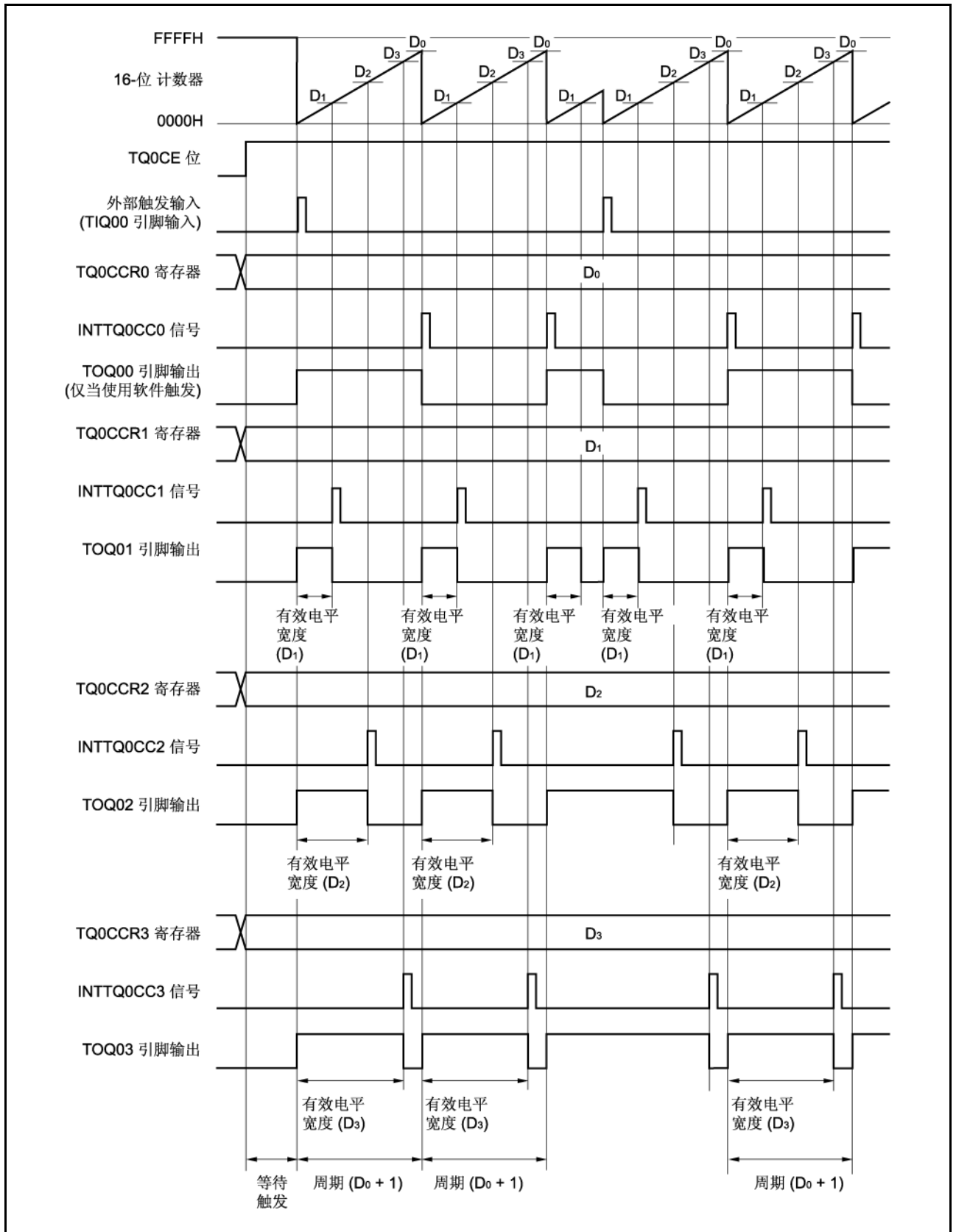


图 8-21. 外部触发脉冲输出模式的寄存器设置



TQ0CE 位被置 1 后, 16 位定时器 / 事件计数器 Q 进入等待触发信号的状态。触发产生后, 16 位计数器从 FFFFH 清零为 0000H, 同时开始计数并从 TOQ0k 引脚输出一个 PWM 波形。如果在计数器计数过程中, 再次检测到触发信号, 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。触发时无论低电平输出周期和高电平输出周期状态, TOQ0k 引脚输出高电平。

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 (TIQ00) 的有效沿和设置软件触发 (TQ0CTL1.TQ0EST 位 = 1) 的方式进行选择。

备注 k = 1 至 3, m = 0 至 3

图 8-22. 外部触发脉冲输出模式的寄存器设置 (1/3)

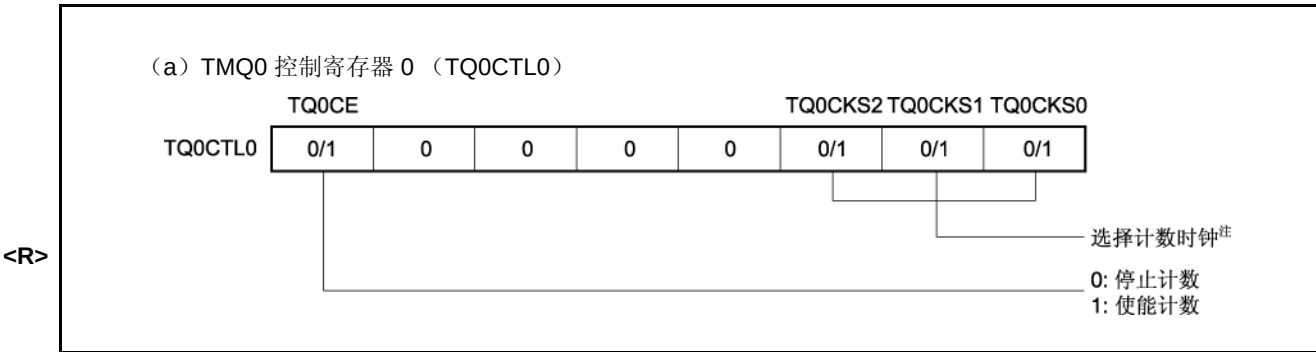


图 8-22. 外部触发脉冲输出模式的寄存器设置 (2/3)

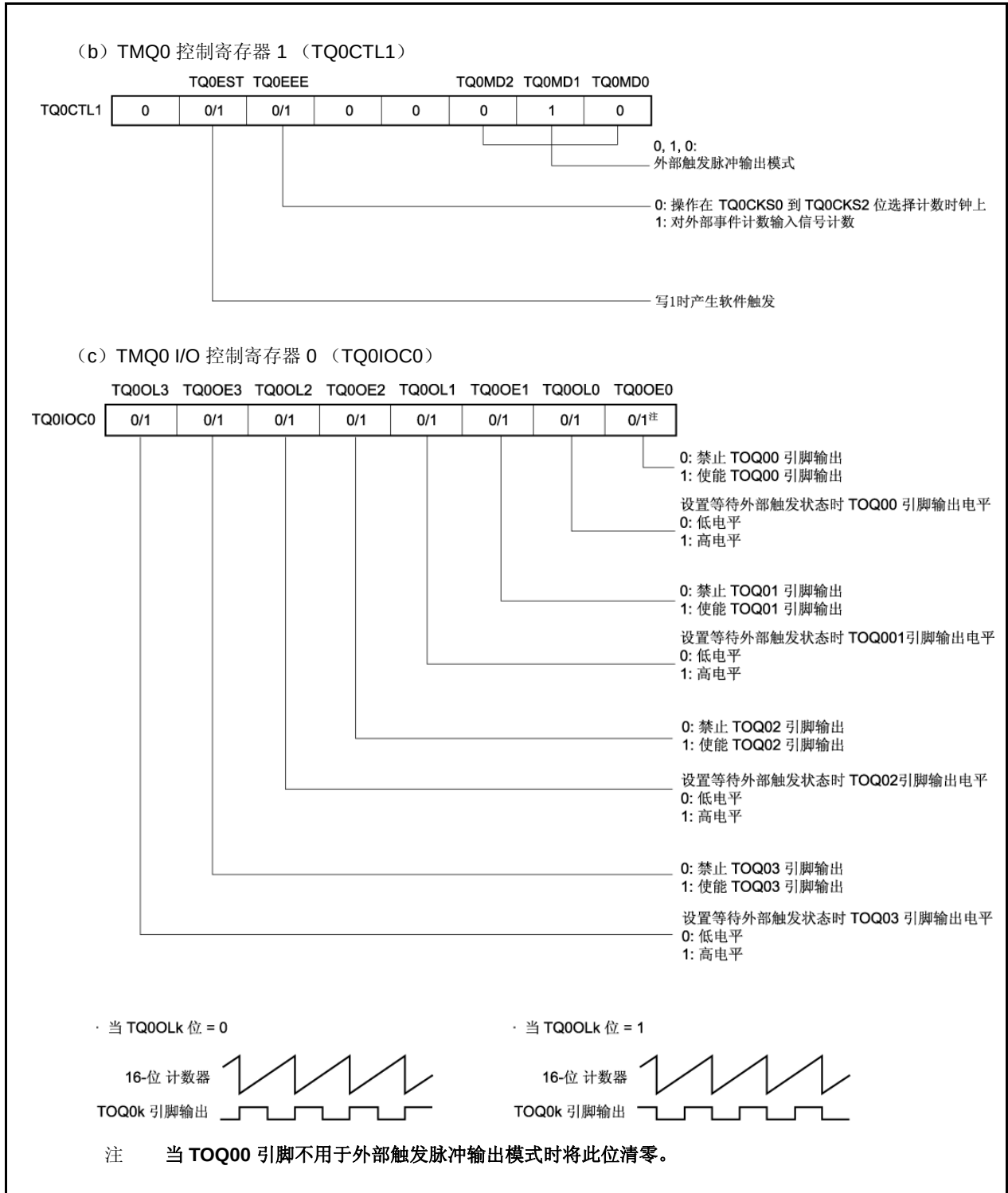
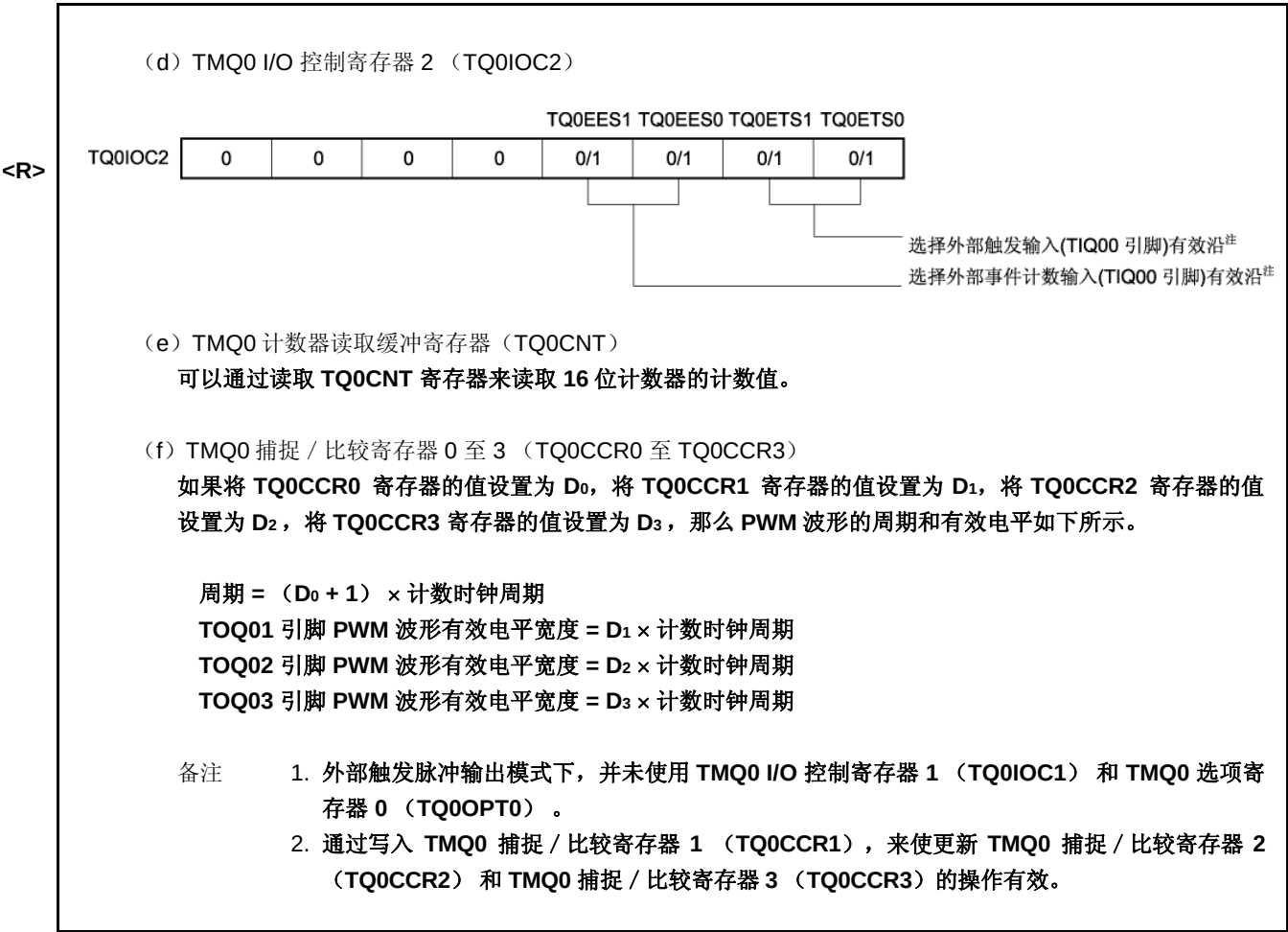


图 8-22. 外部触发脉冲输出模式的寄存器设置 (3/3)



(1) 外部触发脉冲输出模式的操作流程

图 8-23. 外部触发脉冲输出模式的软件处理流程 (1/2)

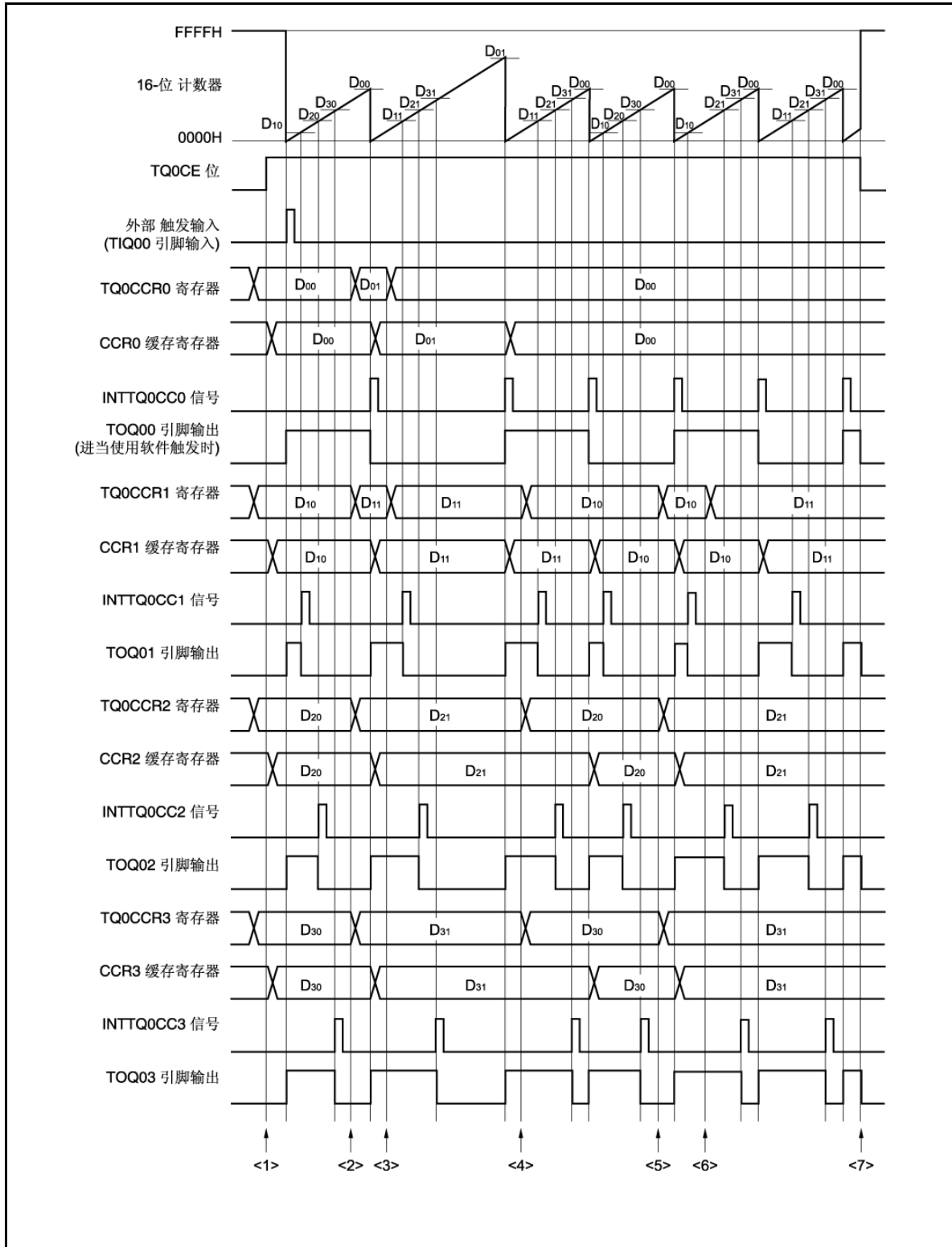
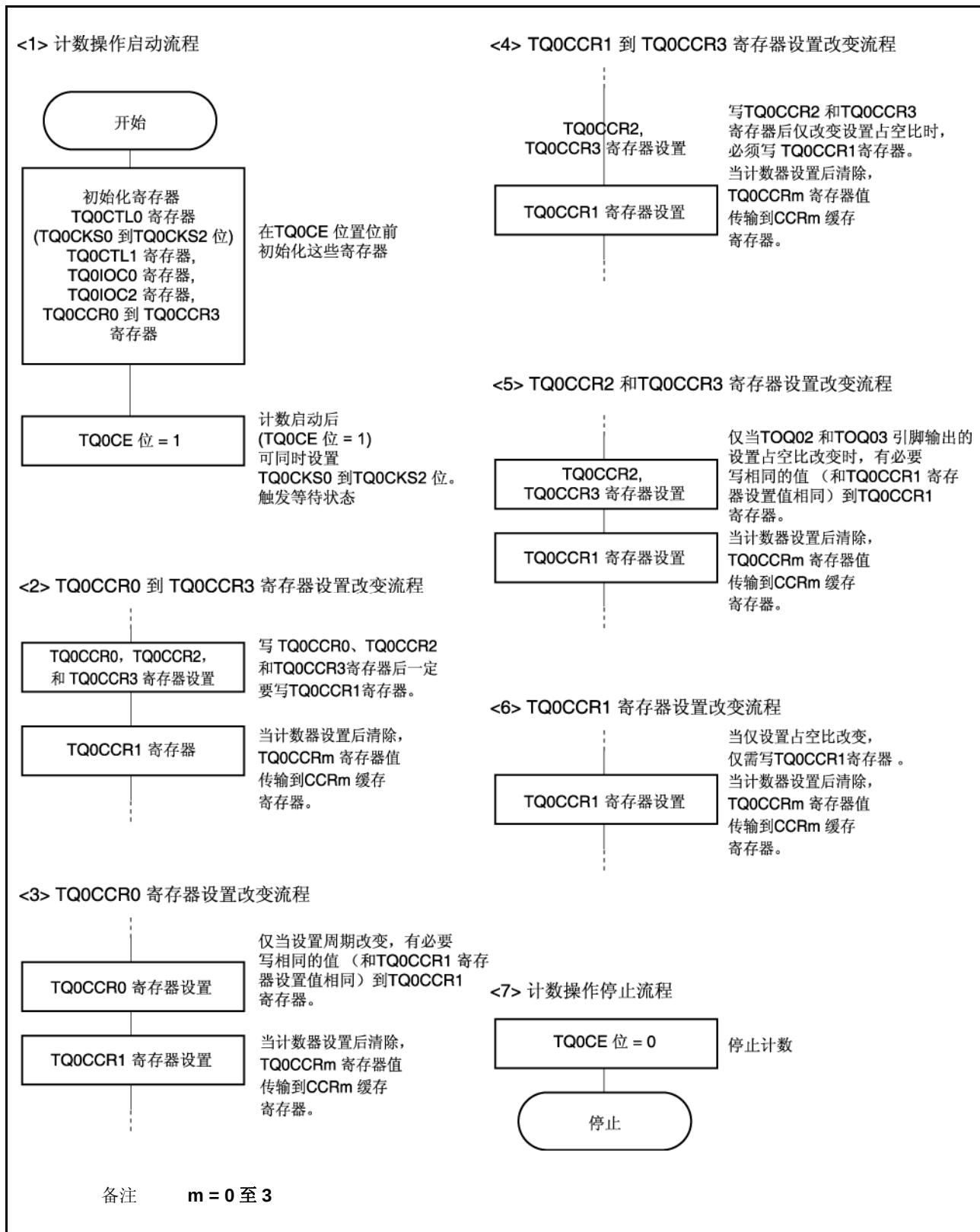


图 8-23. 外部触发脉冲输出模式的软件处理流程 (2/2)

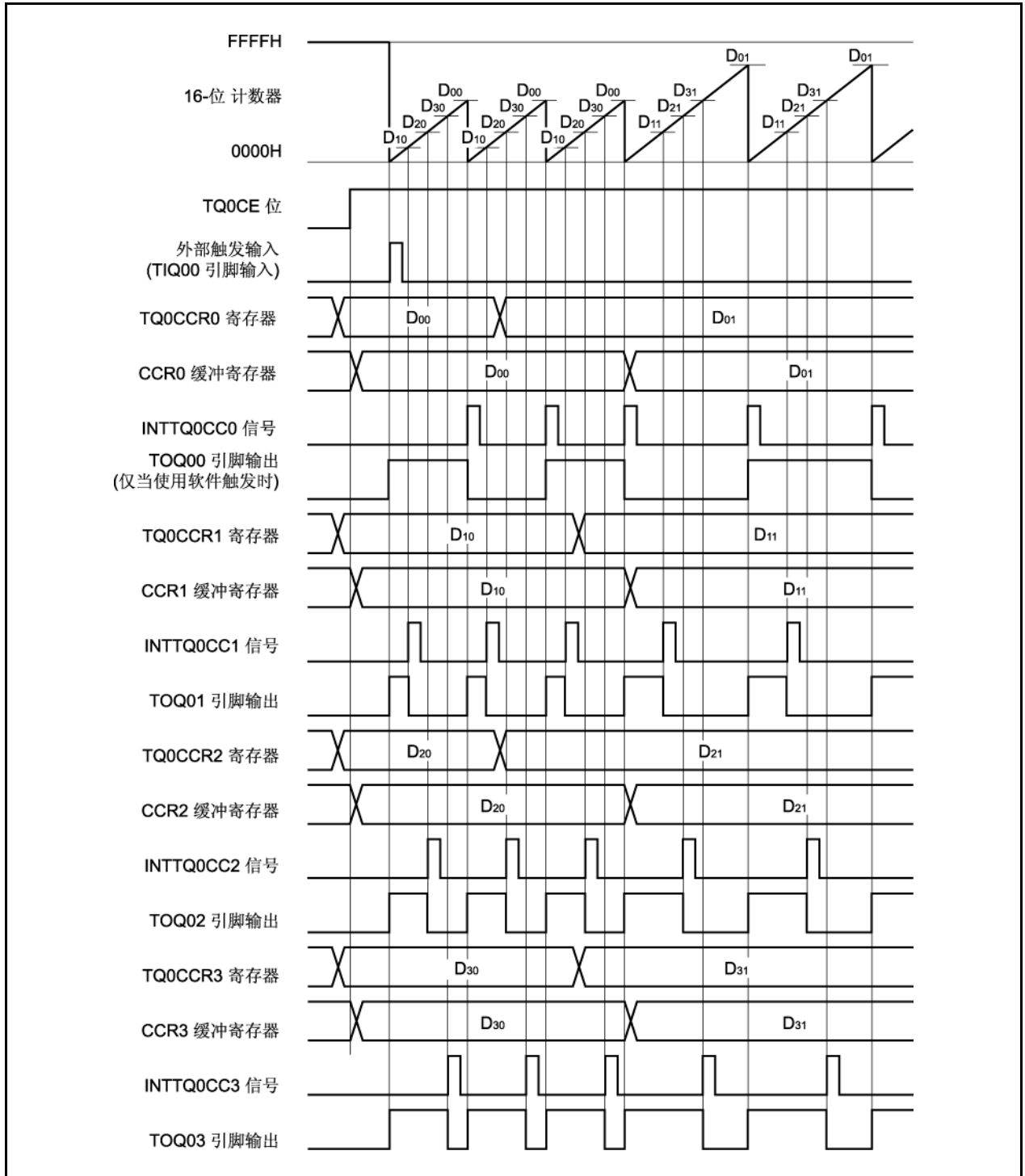


(2) 外部触发脉冲输出模式的操作时序

(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。

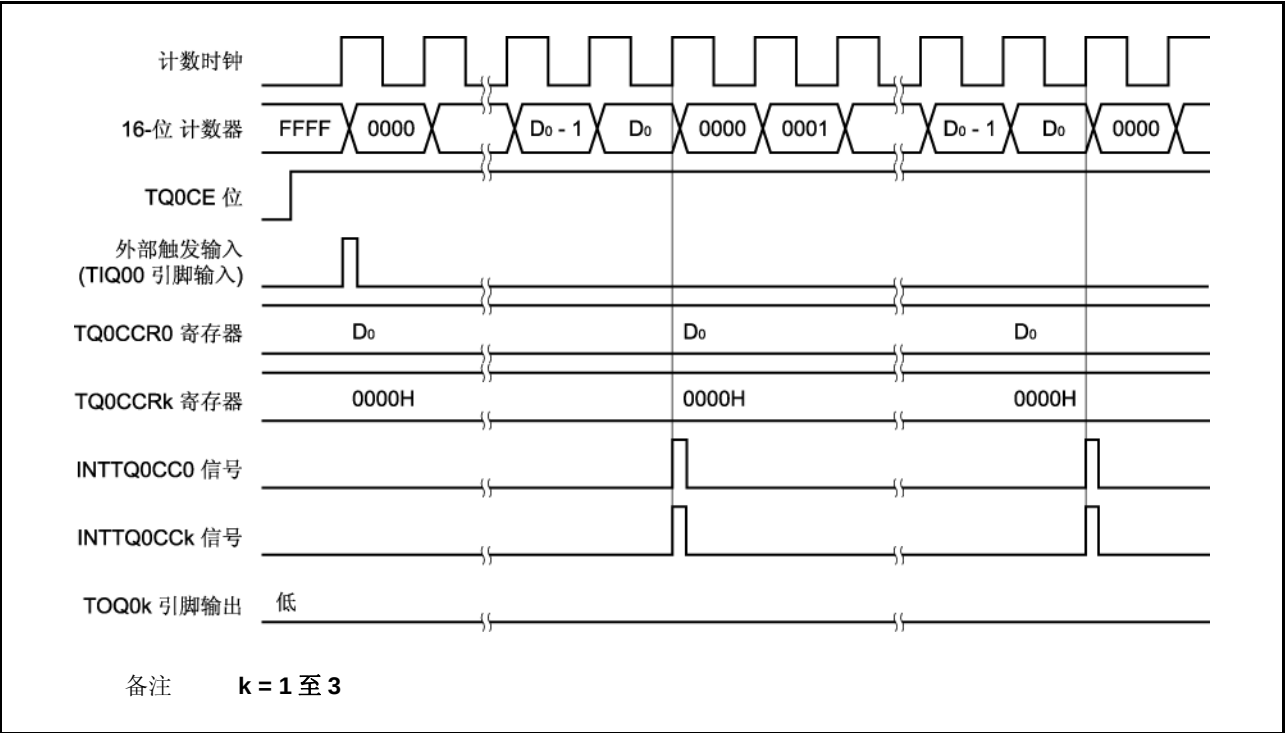
备注 m = 0 至 3

(b) PWM 波形的 0% / 100% 输出

<R>

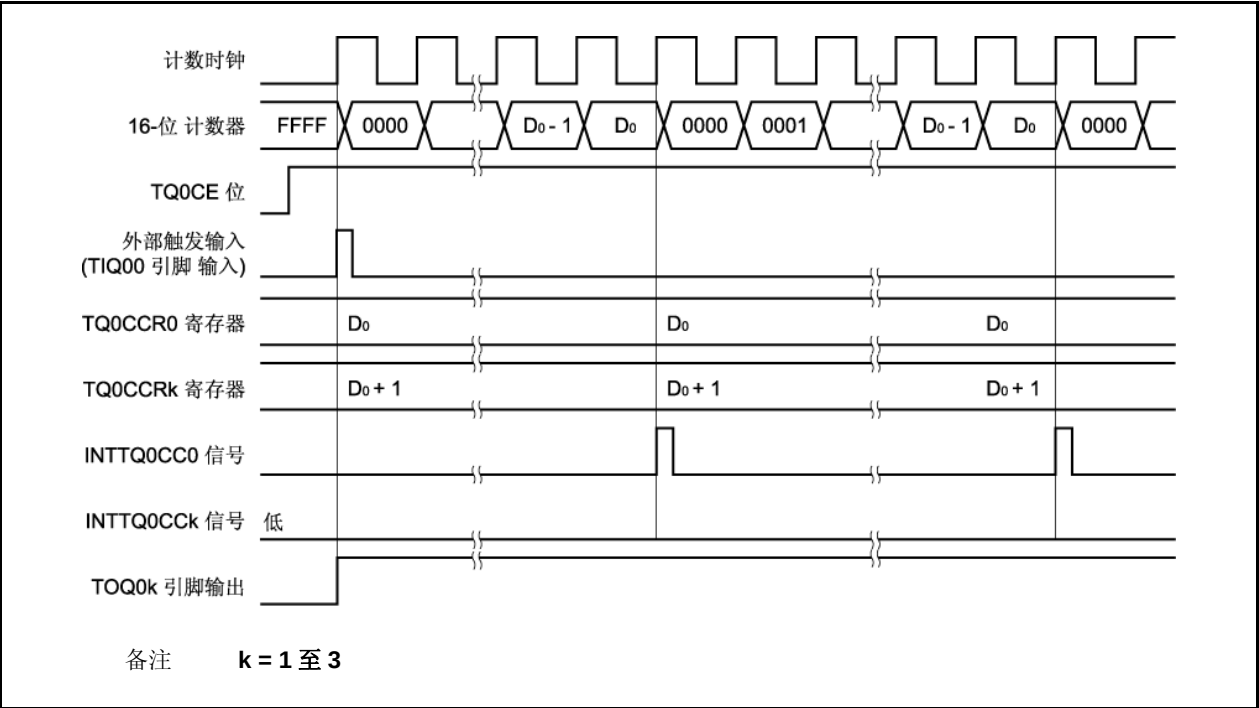
若要输出 0% 的波形，则需将 TQ0CCRk 寄存器设置为 0000H。在 16 位计数器的计数值与 CCR0 缓冲寄存器值匹配后的下一时刻，16 位计数器清零并且产生 INTTQ0CC0 到 INTTQ0CCK 信号。

<R>



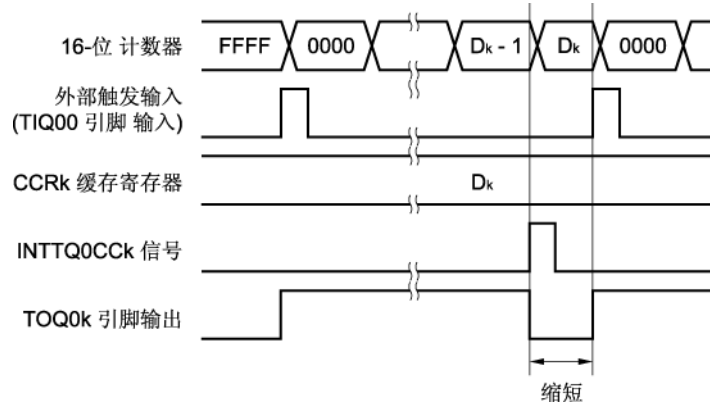
若要输出 100% 波形，则需将 TQ0CCRk 寄存器设置为 (TQ0CCR0 寄存器值 + 1)。TQ0CCR0 寄存器的值为 FFFFH 时，无法实现 100% 输出。

<R>



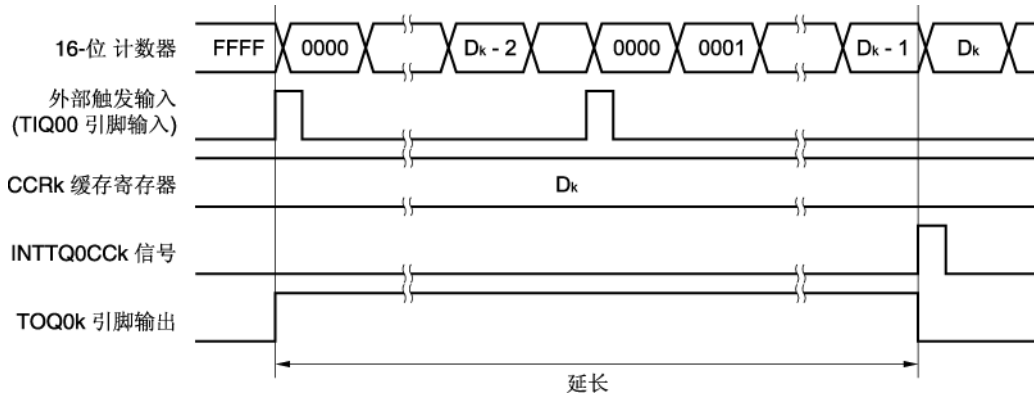
(c) 触发检测和 CCRk 缓冲寄存器匹配之间的冲突

如果紧随在 INTTQ0CCk 中断信号产生后马上检测到触发信号，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

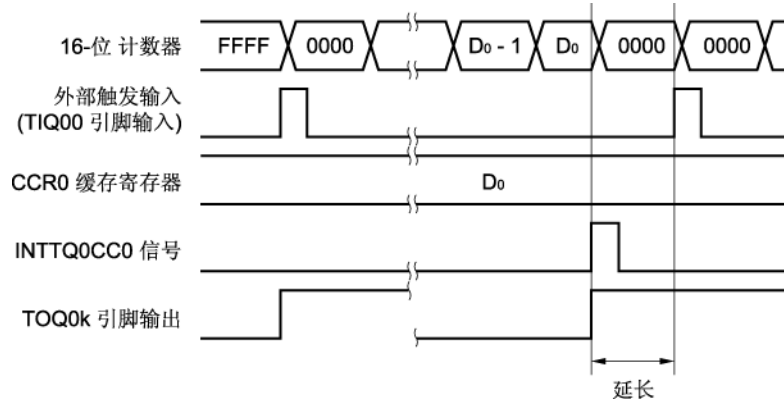
如果在 INTTQ0CCk 信号产生之前检测到触发信号，那么将不会产生 INTTQ0CCk 信号，此时，16 位计数器被清零并重新开始计数。TOQ0k 引脚的输出也会继续保持为有效电平。这种情况下，PWM 波形的有效电平时间就会被延长。



备注 $k = 1$ 至 3

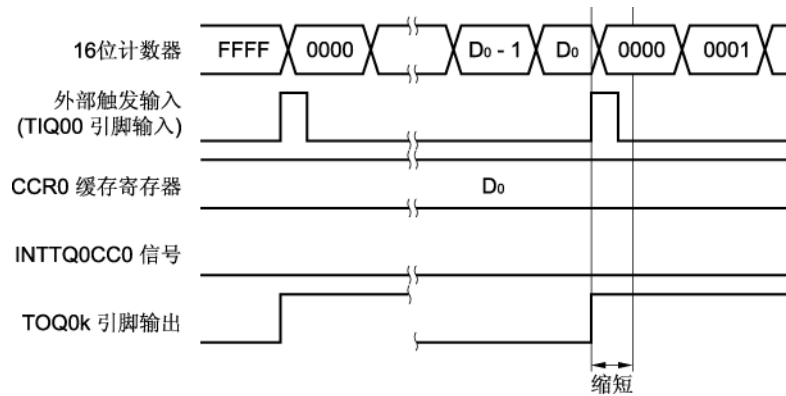
(d) 触发检测和 CCR0 缓冲寄存器匹配之间的冲突

如果在 INTTQ0CC0 中断信号产生后马上检测到触发信号，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 信号产生到检测触发信号的时间。



备注 $k = 1$ 至 3

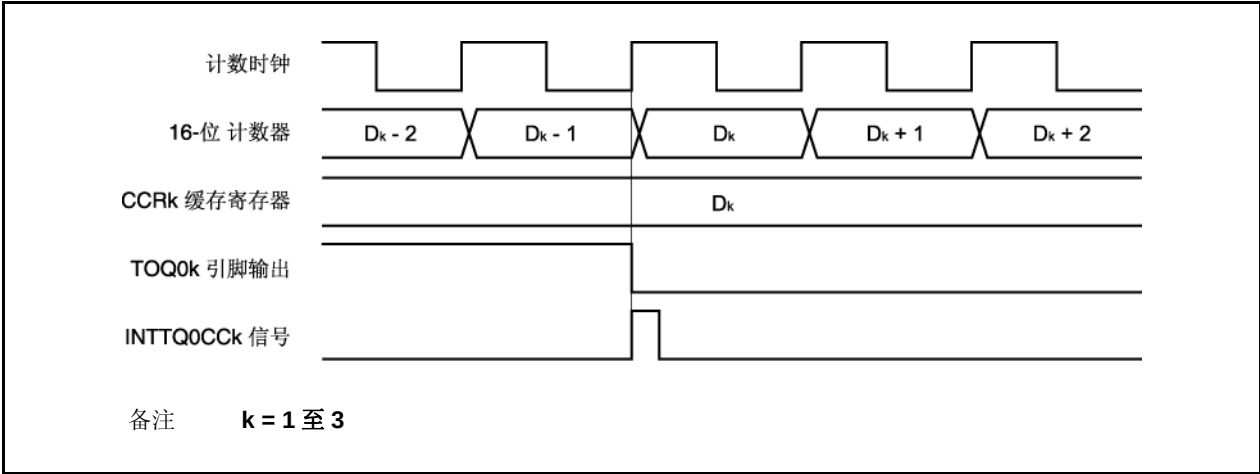
如果在 INTTQ0CC0 信号产生之前检测到触发信号，那么将不会产生 INTTQ0CC0 信号。16 位计数器将被清零并重新开始计数，TOQ0k 引脚输出也会改变为有效电平。这种情况下，PWM 波形的非有效电平时间就会被缩短。



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。

8.6.4 单脉冲输出模式 (TQ0MD2 到 TQ0MD0 位 = 011)

在单脉冲输出模式中，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 (TIQ00) 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。

软件触发器也能代替外部触发器触发输出脉冲，TOQ00 引脚会在 16 位计数器计数时输出有效电平，在计数器停止计数时（等待触发时）输出非有效电平。

图 8-24. 单脉冲输出模式的配置图

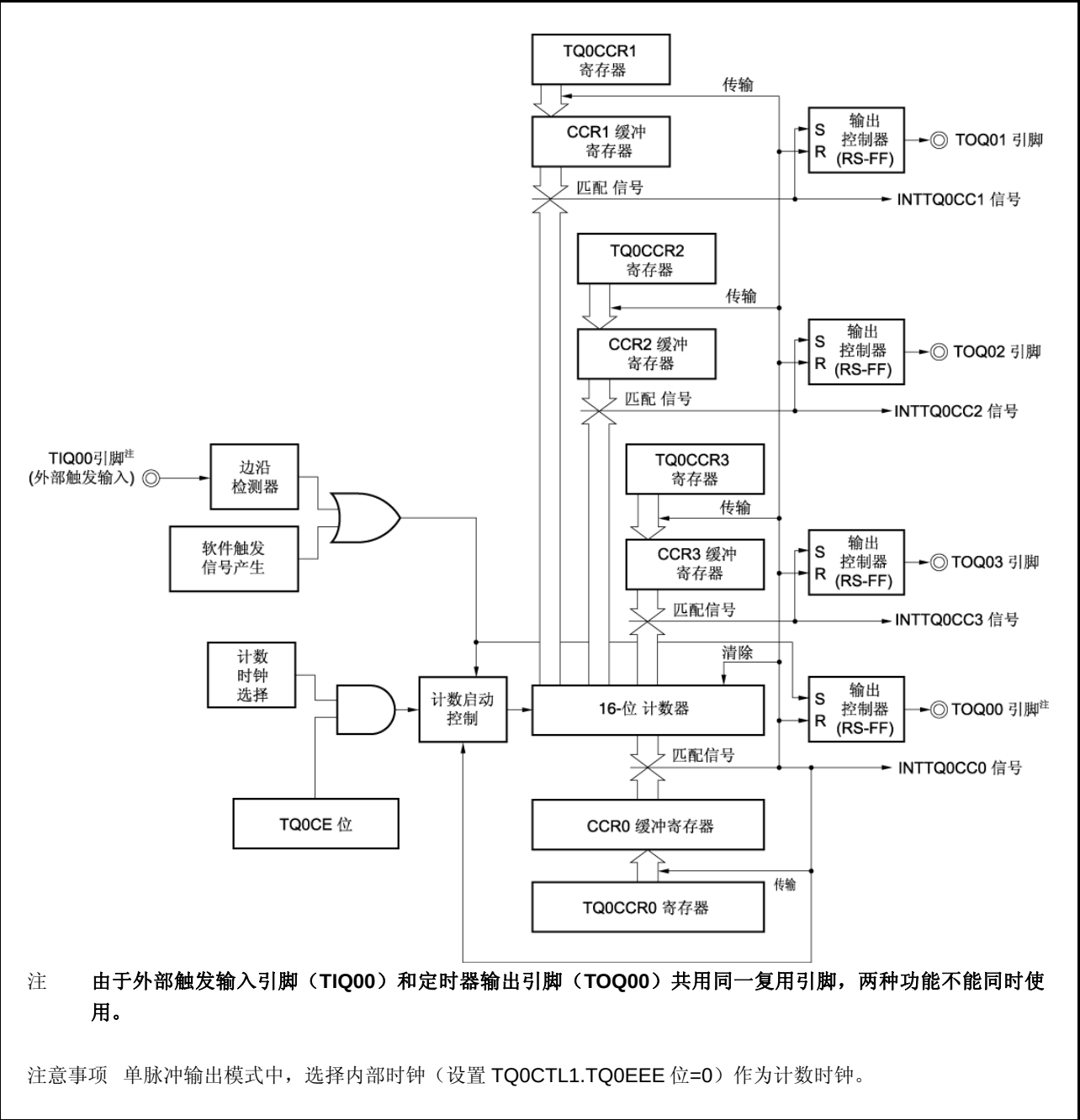
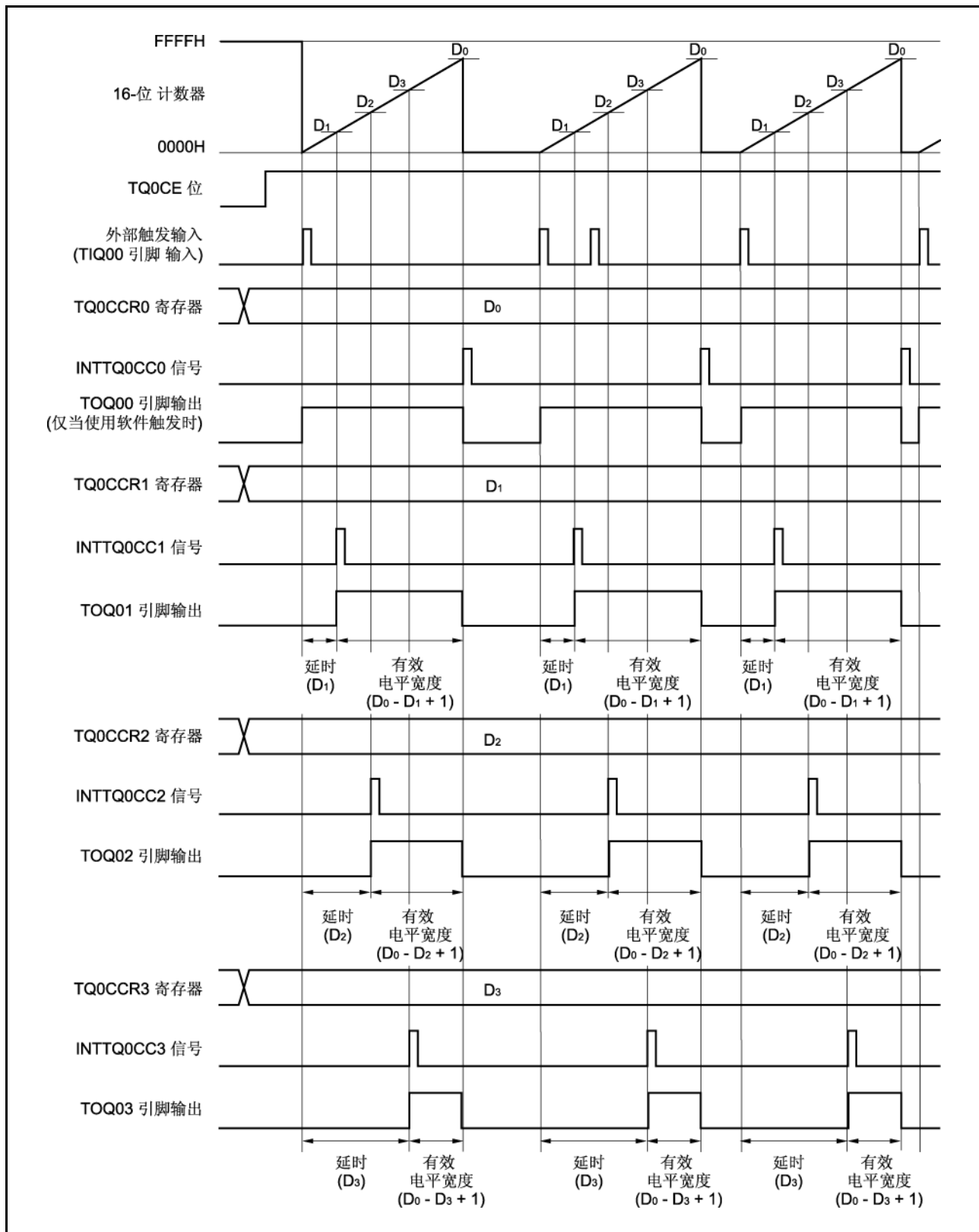


图 8-25. 单脉冲输出模式的基本时序



<R>

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。检测到触发信号后，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 0000H 并停止计数，等待触发信号。当触发信号再次产生，16 位计数器从 0000H 开始计数。如果在单脉冲波形输出过程中再次检测到触发信号，那么将忽略这个触发信号。

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后产生。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 (TIQ00 引脚) 的有效沿和设置软件触发 (TQ0CTL1.TQ0EST 位 = 1) 置 1 的方式进行选择。

备注 k = 1 至 3

图 8-26. 单脉冲输出模式的寄存器设置 (1/3)

<R>

<R>

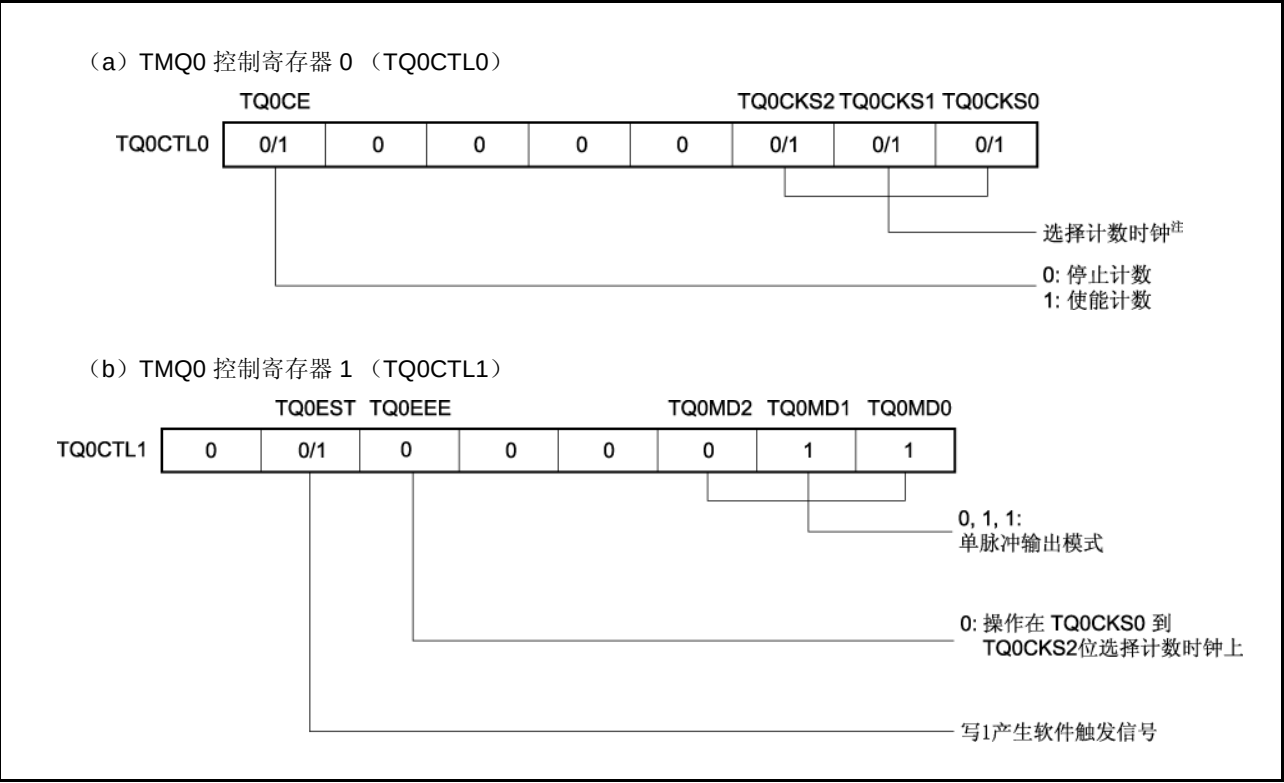
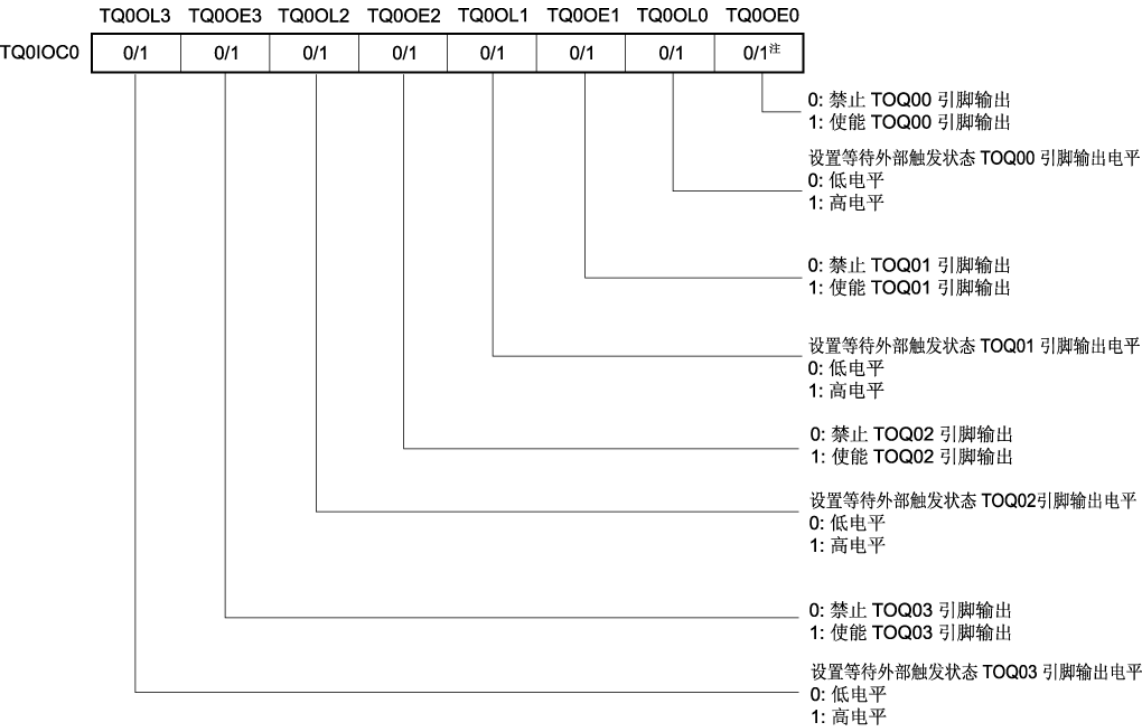


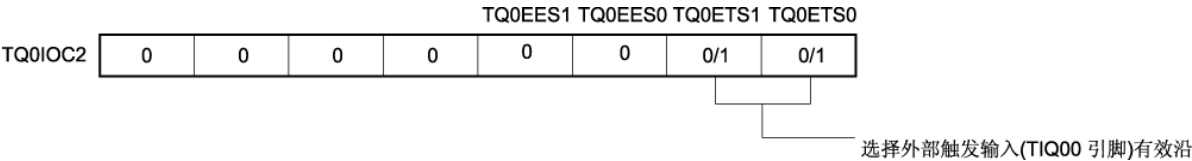
图 8-26. 单脉冲输出模式的寄存器设置 (2/3)

(c) TMQ0 I/O 控制寄存器 0 (TQ0IOC0)



注 当单脉冲输出模式下不使用 TOQ00 引脚时，要将该位清零。

(d) TMQ0 I/O 控制寄存器 2 (TQ0IOC2)



<R>

图 8-26. 单脉冲输出模式的寄存器设置 (3/3)

- (e) TMQ0 计数器读取缓冲寄存器 (TQ0CNT)

可以通过读取 **TQ0CNT** 寄存器来读取 16 位计数器的计数值。
- (f) TMQ0 捕捉 / 比较寄存器 0 至 3 (TQ0CCR0 至 TQ0CCR3)

如果将 **TQ0CCR0** 寄存器的值设置为 D_0 ，将 **TQ0CCRk** 寄存器的值设置为 D_k ，那么单脉冲波形的有效电平宽度和输出延迟时间如下所示。

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

输出延迟时间 = $(D_k) \times \text{计数时钟周期}$
- 注意事项 单脉冲输出模式下，如果 TQ0CCRk 寄存器值大于 TQ0CCR0 寄存器值，不会输出单脉冲波形。
- 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 8-27. 单脉冲输出模式的软件操作流程 (1/2)

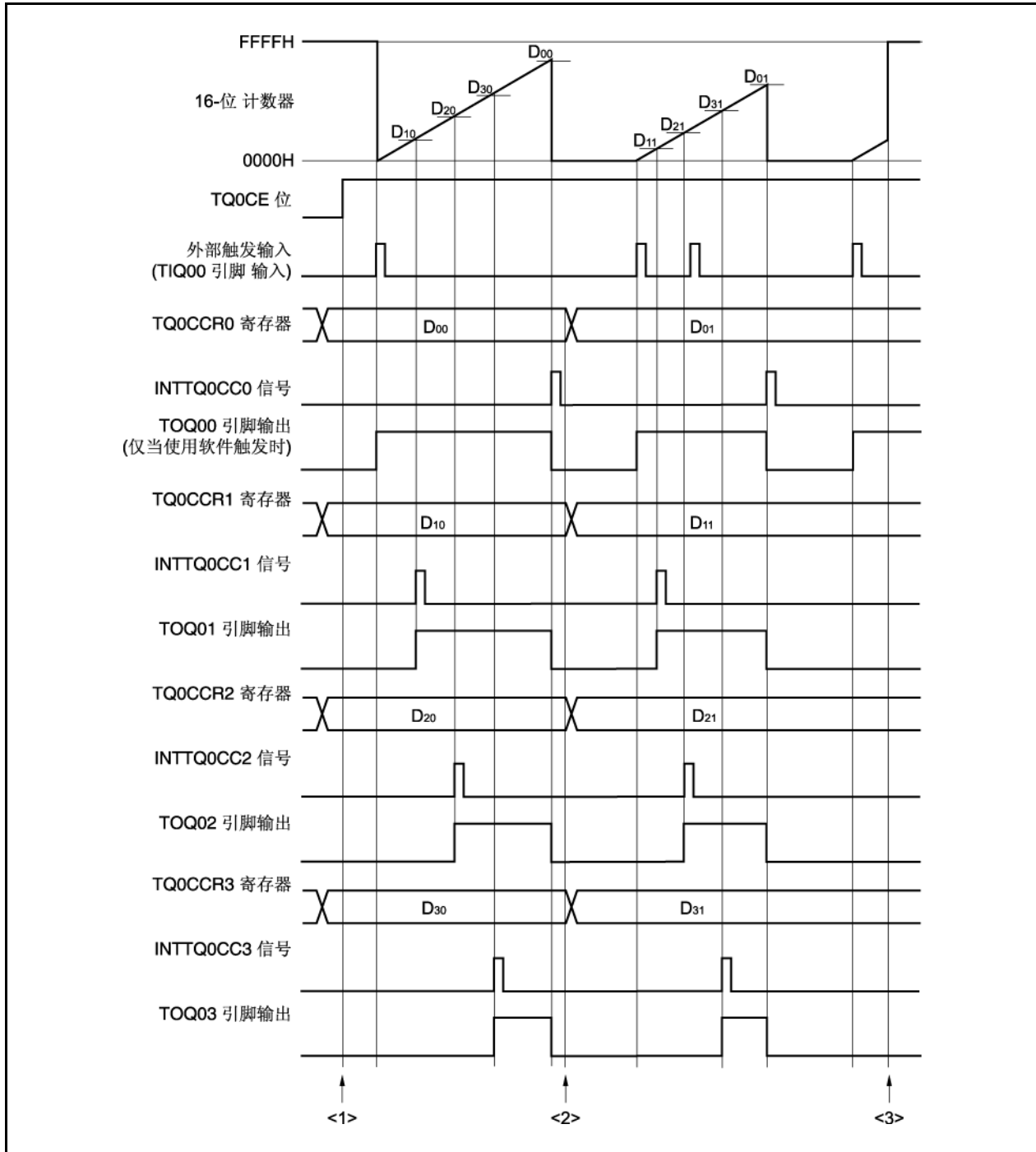
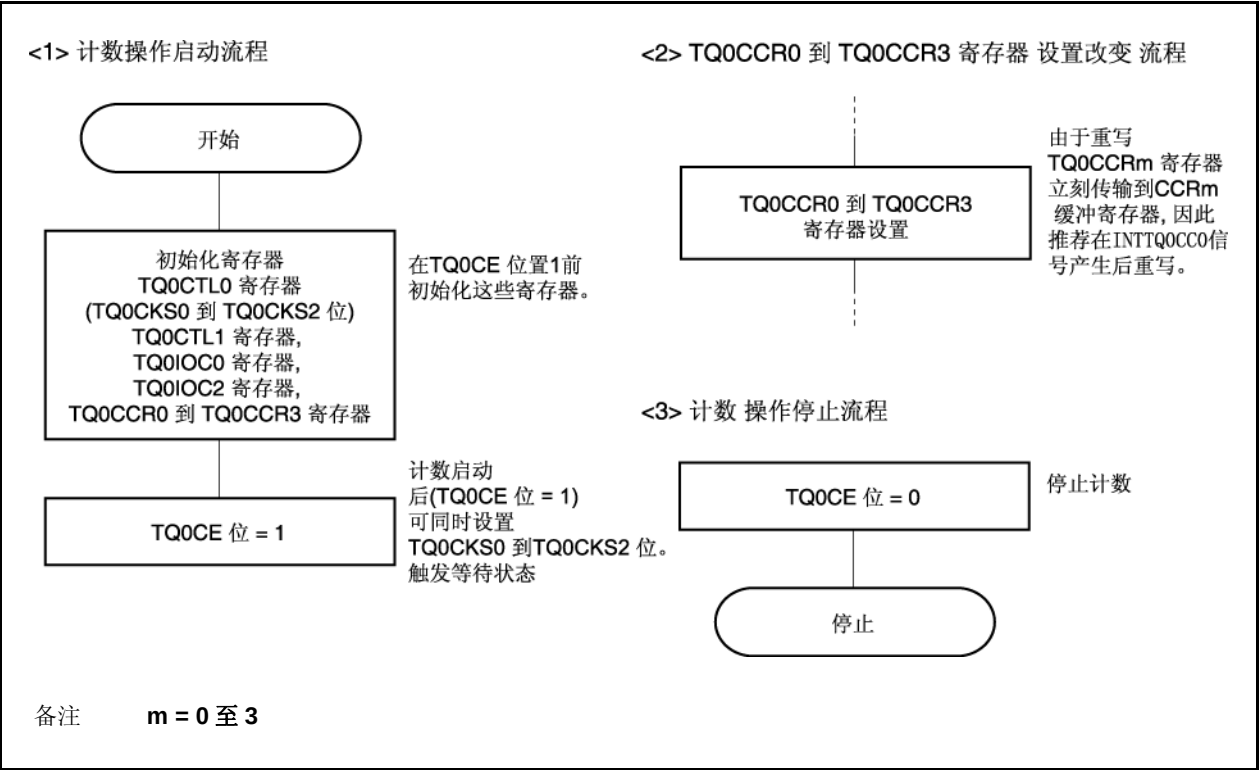


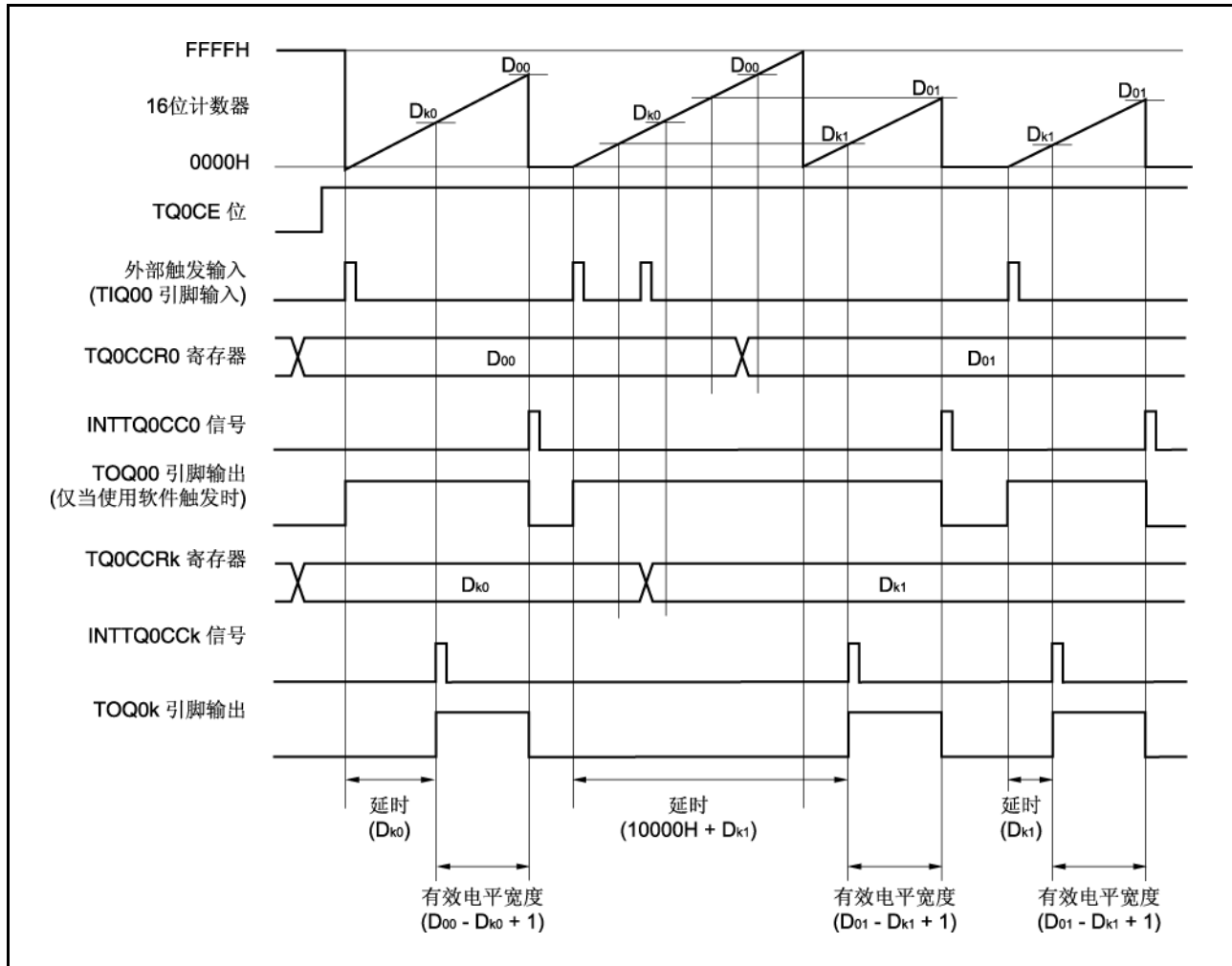
图 8-27. 单脉冲输出模式的软件操作流程（2/2）



(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

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。如果可能溢出，停止计数然后改变设置值。

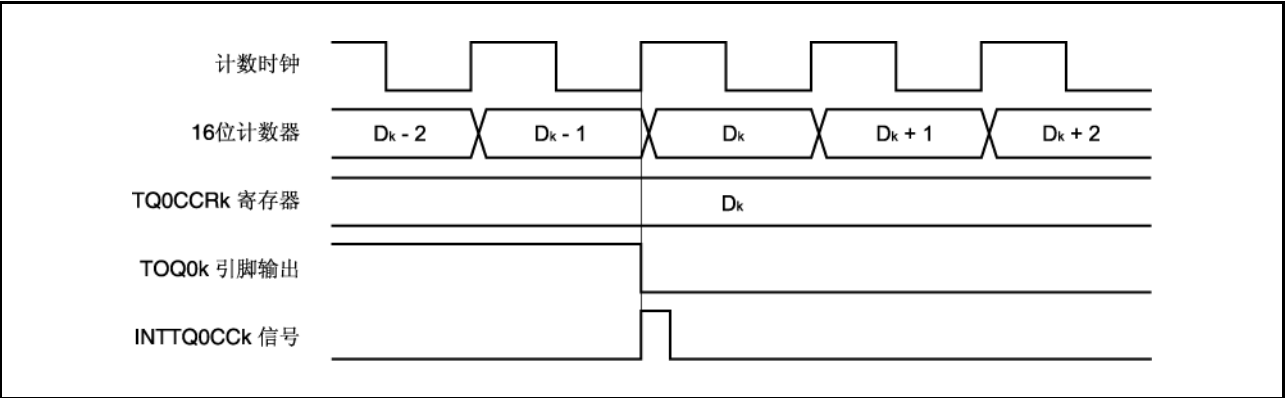


当寄存器 TQ0CCR0 从 D00 改写为 D01，寄存器 TQ0CCRk 从 Dk0 改写为 Dk1 时，这里 $D_{00} > D_{01}$ 和 $D_{k0} > D_{k1}$ ，如果 16 位计数器的计数值大于 D_{k1} 小于 D_{k0} ，并且计数值大于 D_{01} 小于 D_{00} 时，分别重写寄存器 TQ0CCRk 和 TQ0CCR0，那么每个设定值都会在重写寄存器并与计数值比较时马上表现出来。计数器计数到 FFFFH 后将再次从 0000H 开始计数。当计数值与 D_{k1} 相符时，计数器产生 INTTQ0CCk 信号并激活 TOQ0k 引脚。当计数值与 D_{01} 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 $m = 0$ 至 3
 $k = 1$ 至 3

(b) 比较匹配中断请求信号的产生时序 (INTTQ0CCk)

单脉冲输出模式下，INTTQ0CCk 信号的产生时序与其它模式下 INTTQ0CCk 信号的产生时序有所不同。这里，INTTQ0CCk 信号将在 16 位计数器的计数值与 TQ0CCRk 寄存器的值相匹配时产生。



通常情况下，INTTQ0CCk 信号会在 16 位计数器计数值与 TQ0CCRk 寄存器的值匹配之后的下一个计数时钟同步产生。

但在单脉冲输出模式下，INTTQ0CCk 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOQ0k 引脚输出信号变化的时序而改变的。

备注 $k = 1$ 至 3

8.6.5 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 引脚输出半周期为 (TQ0CCR0 寄存器值+1)、占空比为 50%的方波。

<R>

图 8-28. PWM 输出模式的配置图

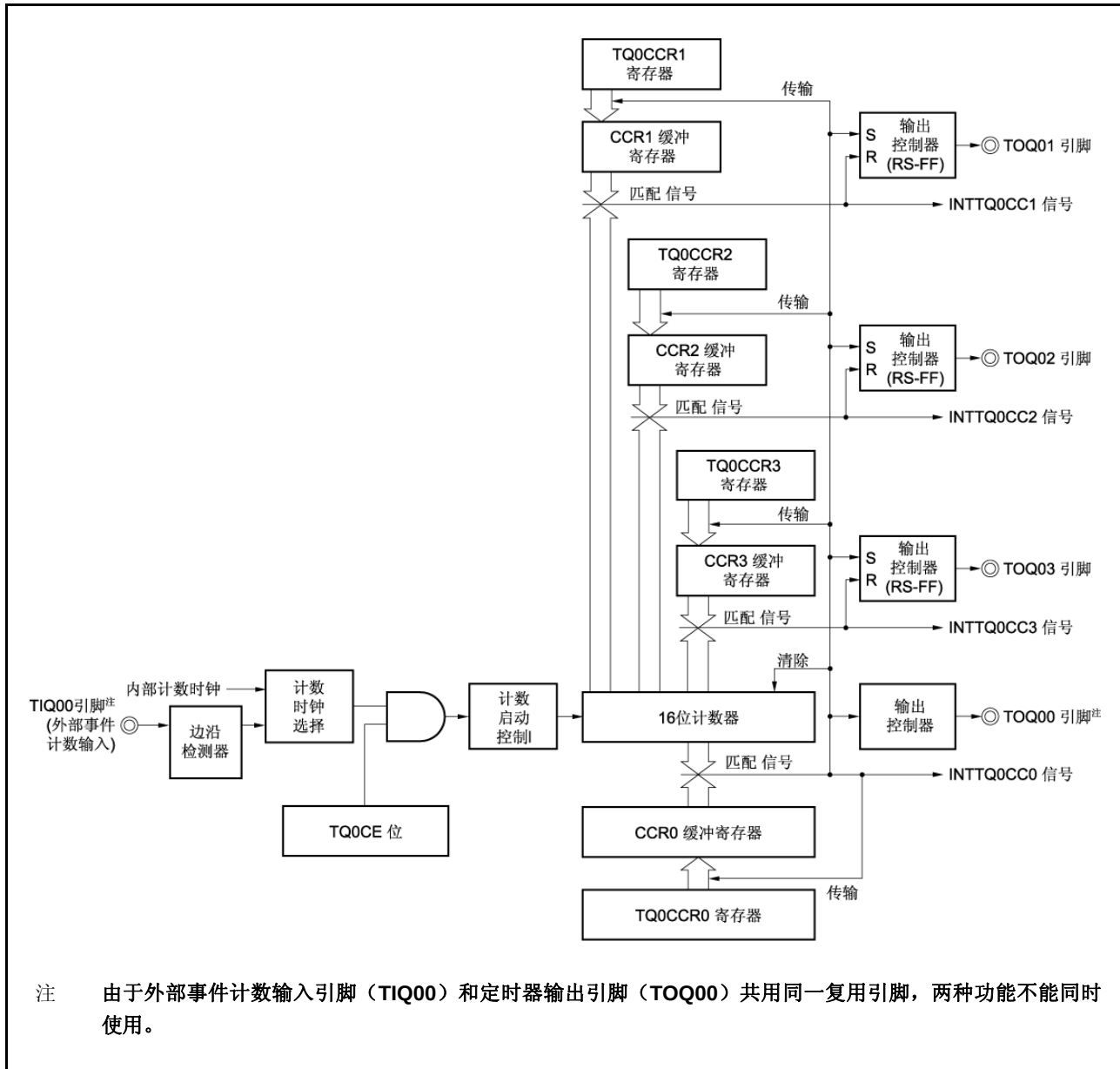
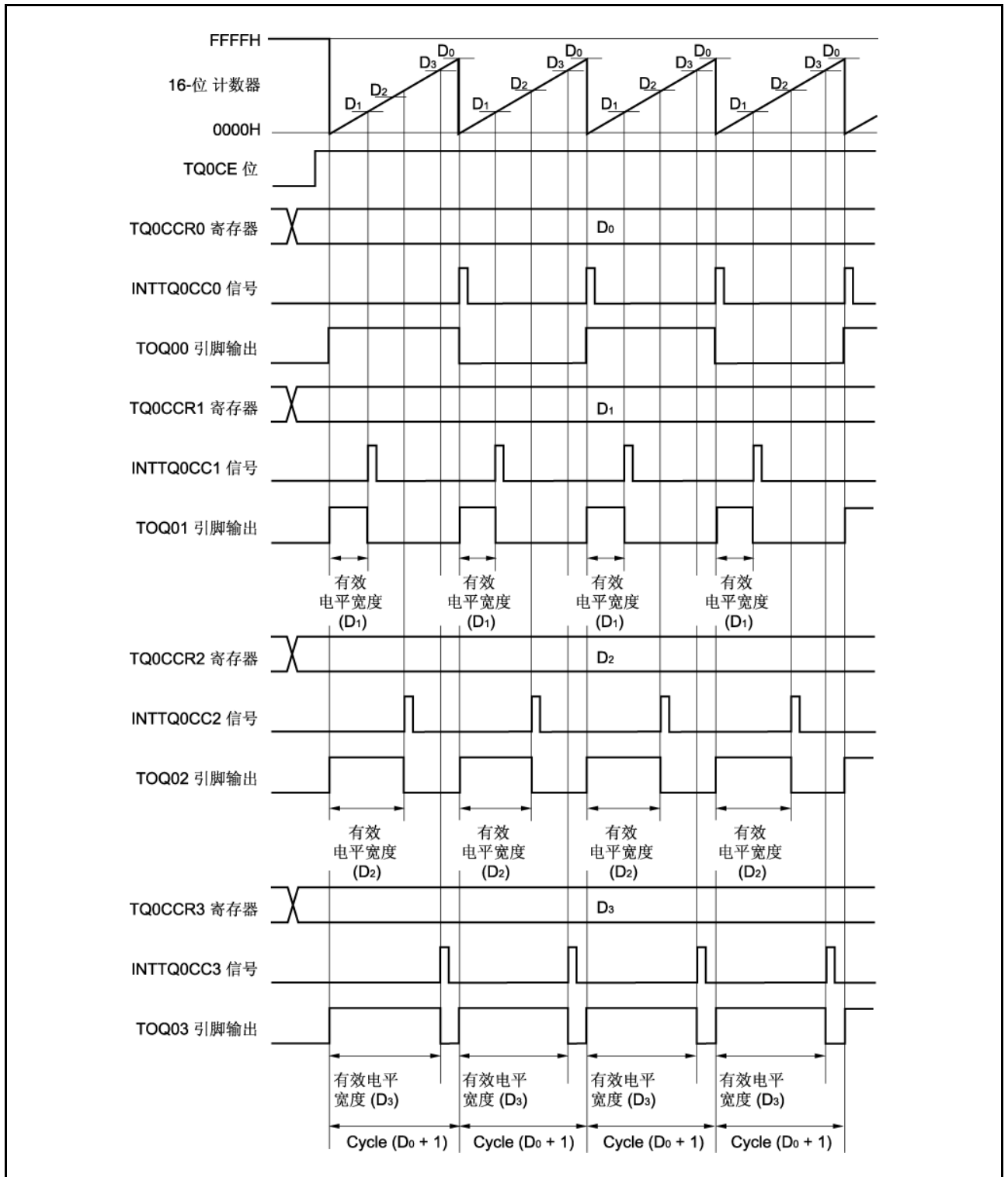


图 8-29. PWM 输出模式的基本时序



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)

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 缓冲寄存器中的值相匹配时产生。

备注 k = 1 至 3, m = 0 至 3

图 8-30. PWM 输出模式的寄存器设置 (1/3)

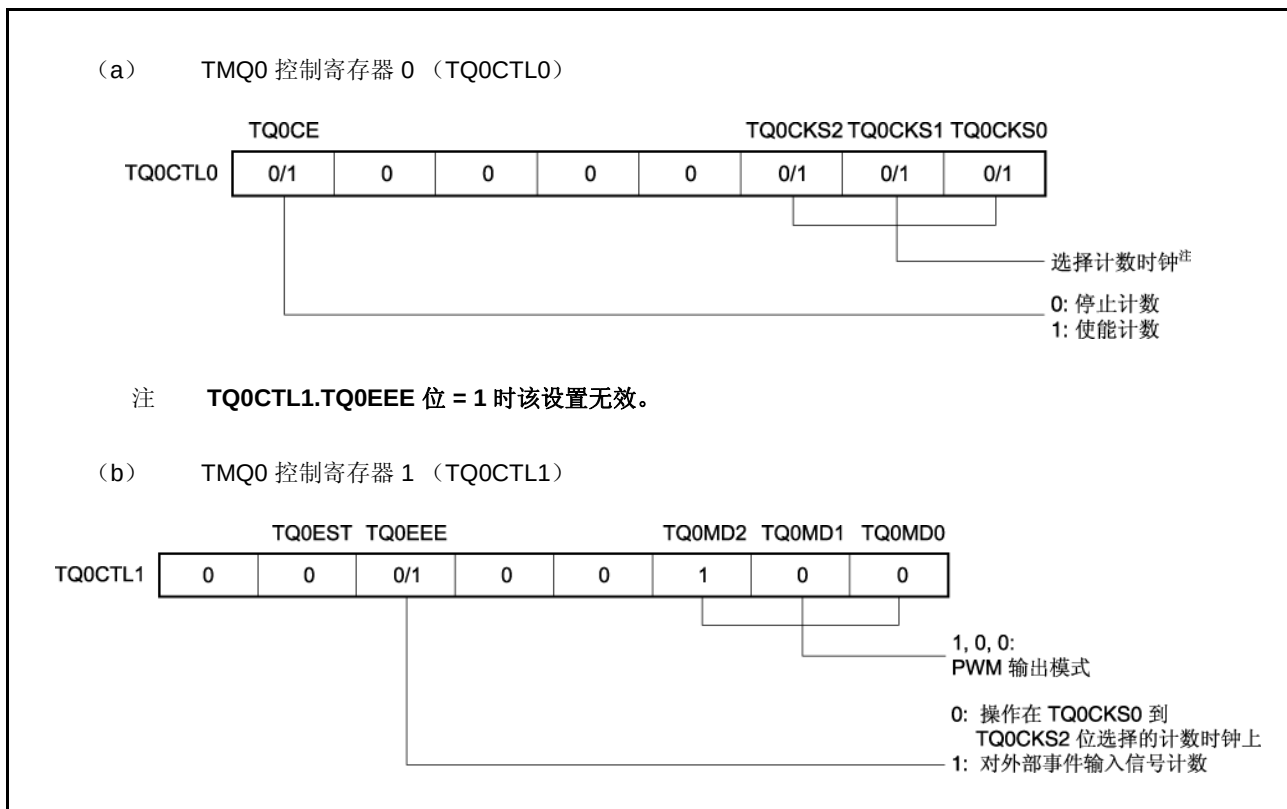


图 8-30. PWM 输出模式的寄存器设置 (2/3)

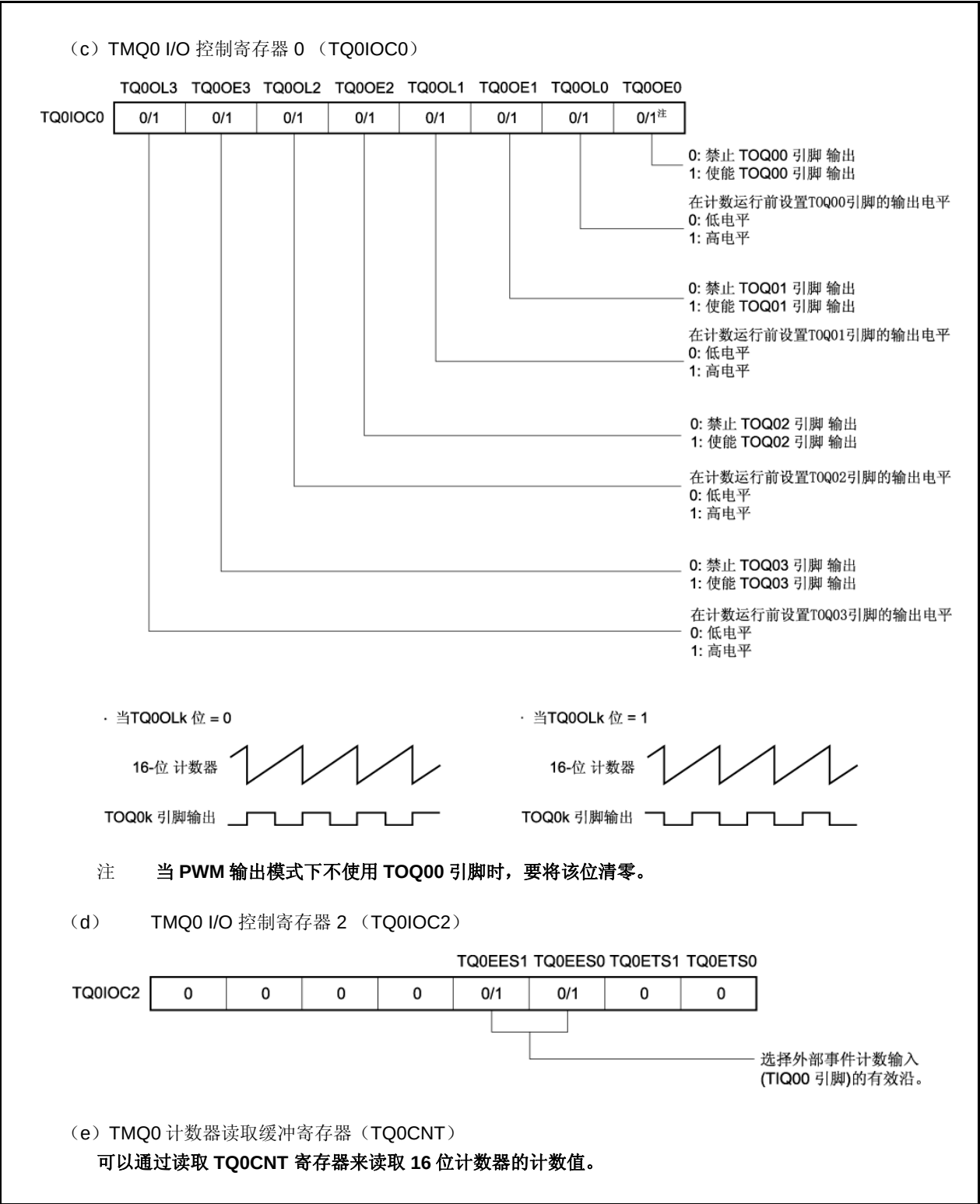


图 8-30. PWM 输出模式的寄存器设置 (3/3)

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

如果将 TQ0CCR0 寄存器的值设置为 D_0 ，将 TQ0CCR k 寄存器的值设置为 D_k ，那么 PWM 波形的周期和有效电平如下所示。

周期 = $(D_0 + 1) \times$ 计数时钟周期

有效电平宽度 = $D_k \times$ 计数时钟周期

- 备注
1. PWM 输出模式下，并未使用 TMQ0 I/O 控制寄存器 1 (TQ0IOC1) 和 TMQ0 选项寄存器 0 (TQ0OPT0)。
 2. 通过写 TMQ0 捕捉 / 比较寄存器 1 (TQ0CCR1) 来使更新 TMQ0 捕捉 / 比较寄存器 2 (TQ0CCR2) 和 TMQ0 捕捉 / 比较寄存器 3 (TQ0CCR3) 有效。

(1) PWM 输出模式的操作流程

图 8-31. PWM 输出模式的软件处理流程 (1/2)

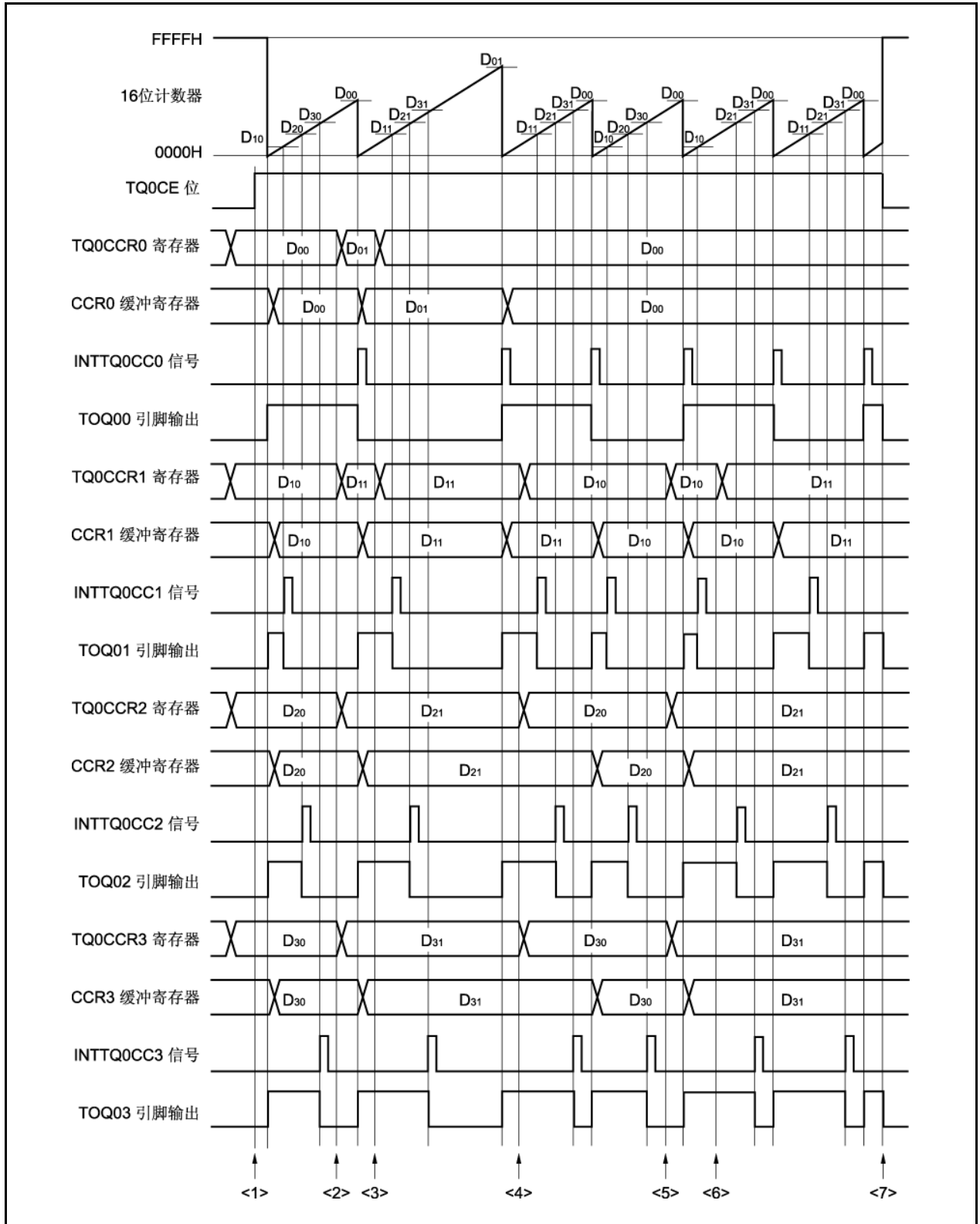
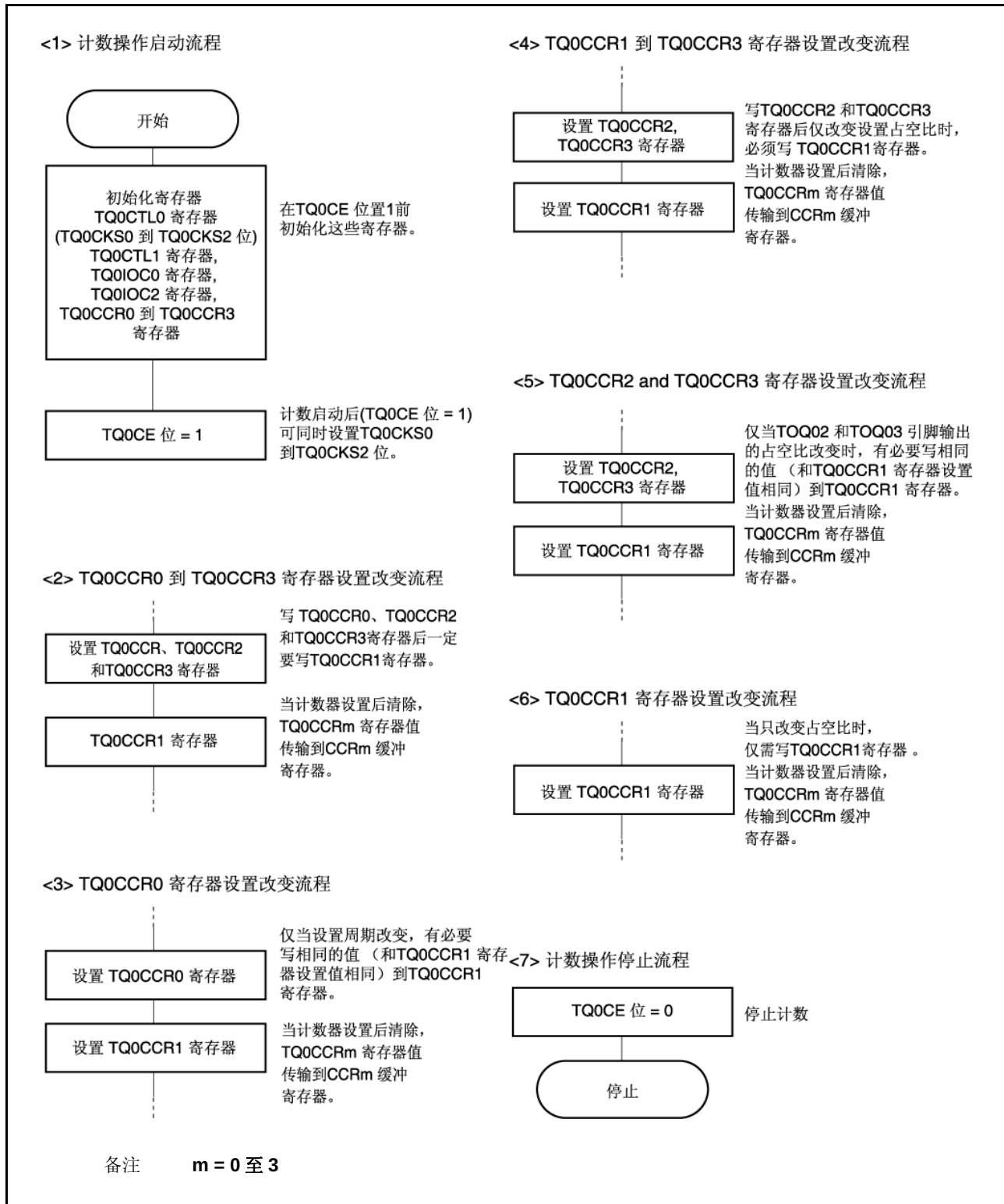


图 8-31. PWM 输出模式的软件处理流程 (2/2)

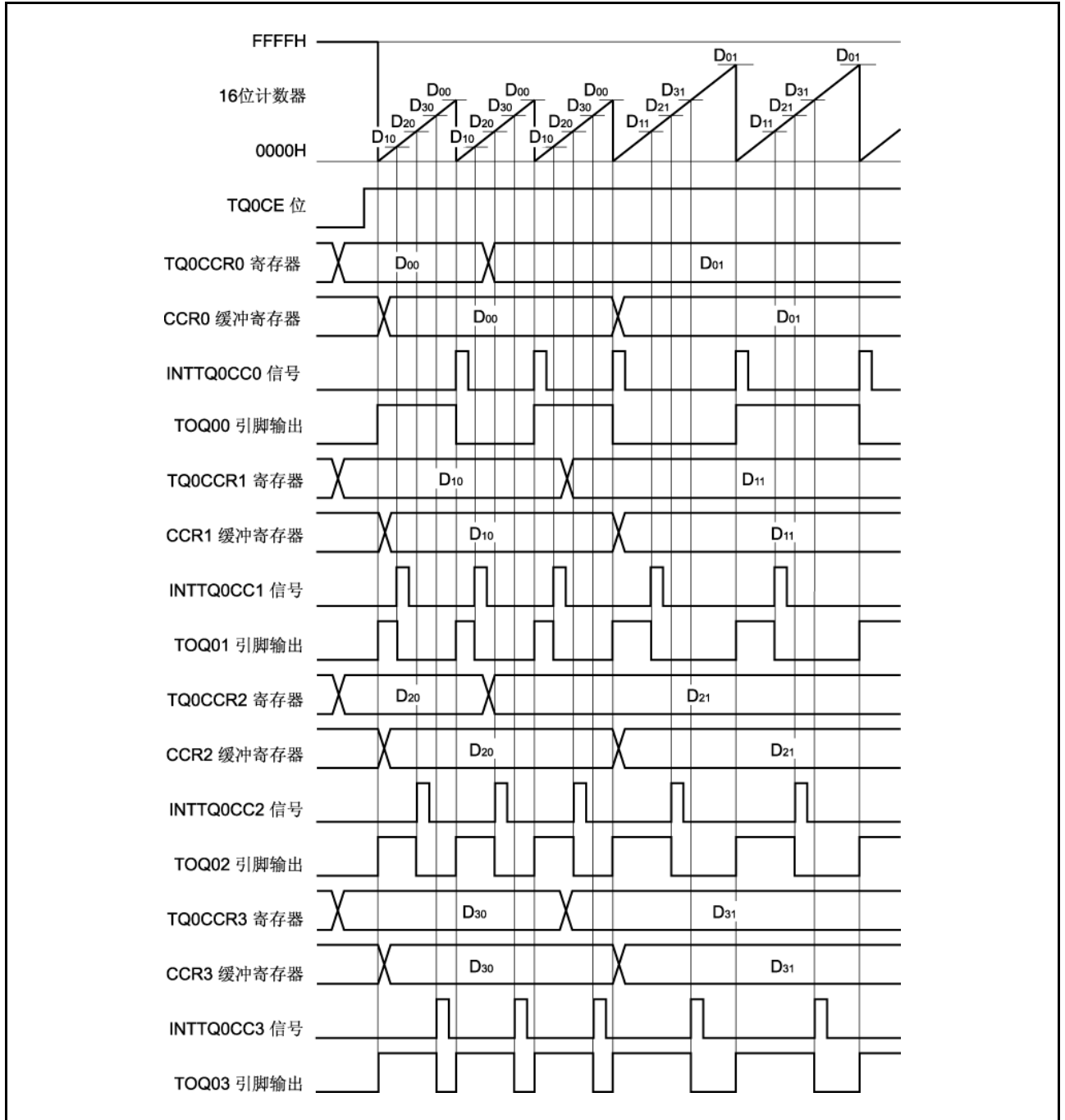


(2) PWM 输出模式的操作时序

(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，在检测到 INTTQ0CC0 信号后再更改 TQ0CCRk 寄存器的值。



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 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 寄存器写入与原设置相同的值。

数据被写入 TQ0CCR1 寄存器之后，16 位计数器被清零时，TQ0CCRM 寄存器中的数值会被同步传送至 CCRm 缓冲寄存器中，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。

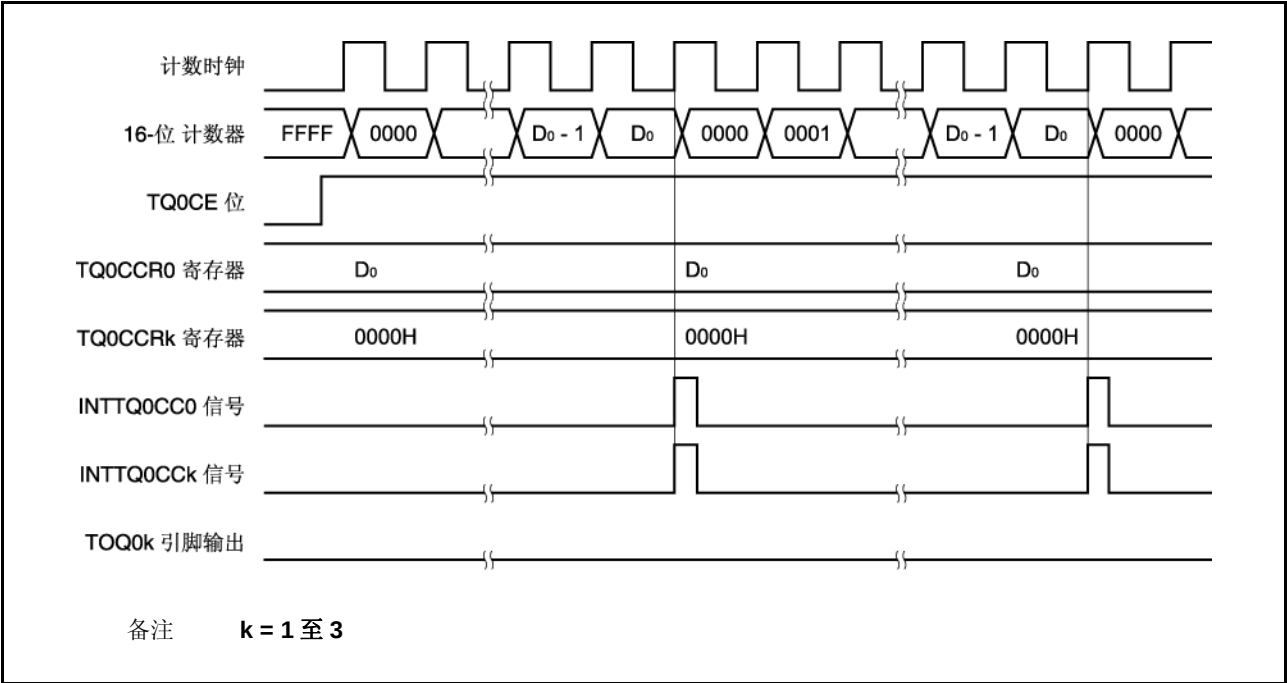
向 TQ0CCR1 寄存器的写入操作完成后，若需再次改写 TQ0CCR0 至 TQ0CCR3 寄存器的值，则要在产生 INTTQ0CC0 信号之后再进行改写。否则，CCRm 缓冲寄存器的值可能无法确定，因为数据从 TQ0CCRM 寄存器传送至 CCRm 缓冲寄存器的时序与 TQ0CCRM 寄存器的写入时序相冲突。

备注 m = 0 至 3

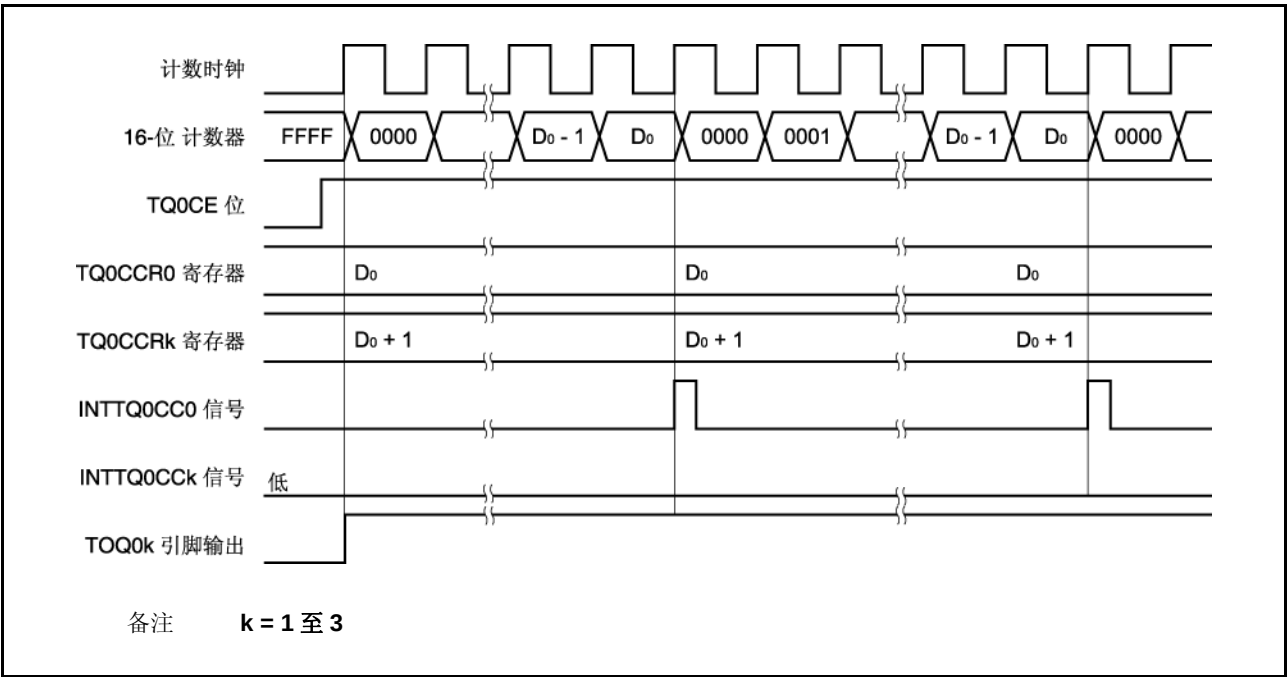
(b) 0%/100% 的 PWM 波形输出

<R>

若要输出 0% 的波形，则需将 TQ0CCRk 寄存器设置为 0000H。在 16 位计数器的计数值与 CCR0 缓冲寄存器值匹配后的下一时刻，16 位计数器清零并且产生 INTTQ0CC0 到 INTTQ0CCK 信号。

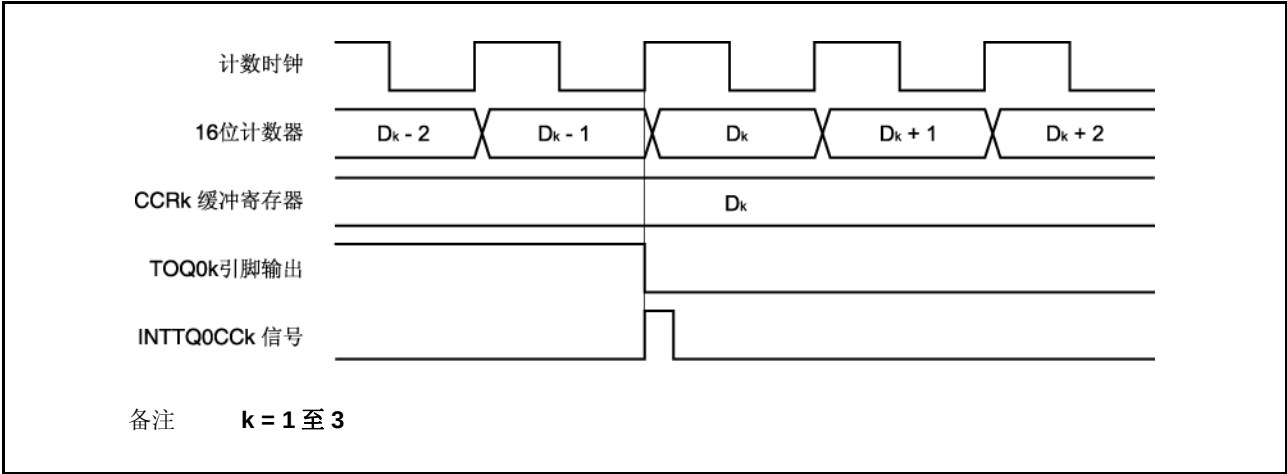


若要输出 100% 波形，则需将 TQ0CCRk 寄存器设置为 (TQ0CCR0 寄存器值 + 1)。TQ0CCR0 寄存器的值为 FFFFH 时，100% 输出是无法实现的。



(c) 比较匹配中断请求信号的产生时序 (INTTQ0CCK)

PWM 输出模式下，INTTQ0CCK 信号的产生时序与其它模式下 INTTQ0CCK 信号的产生时序有所不同，这里，INTTQ0CCK 信号将在 16 位计数器的计数值与 TQ0CCRk 寄存器的值相匹配时产生。



通常情况下，INTTQ0CCK 信号会在 16 位计数器计数值与 TQ0CCRk 寄存器的值匹配之后的下一个计数时钟同步产生。

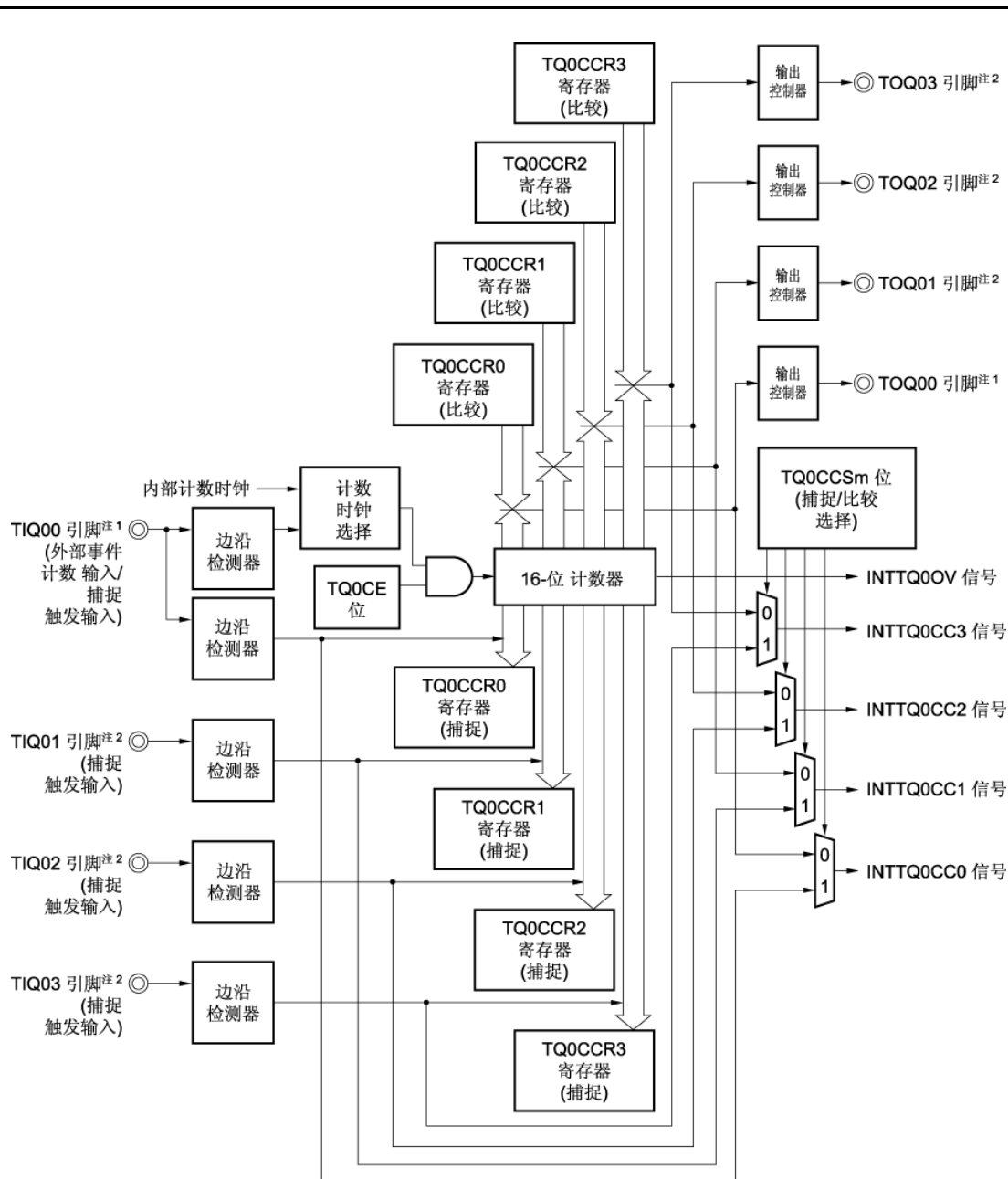
但在 PWM 输出模式下，INTTQ0CCK 信号会提前一个时钟产生。此处，中断信号产生的时序是为了配合 TOQ0k 引脚输出信号变化的时序而改变的。

8.6.6 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.TQ0CCSm 位的设置，TQ0CCRm 寄存器的功能可以在比较寄存器和捕捉寄存器之间进行选择。

备注 m = 0 至 3

图 8-32. 自由运行定时器模式的配置图

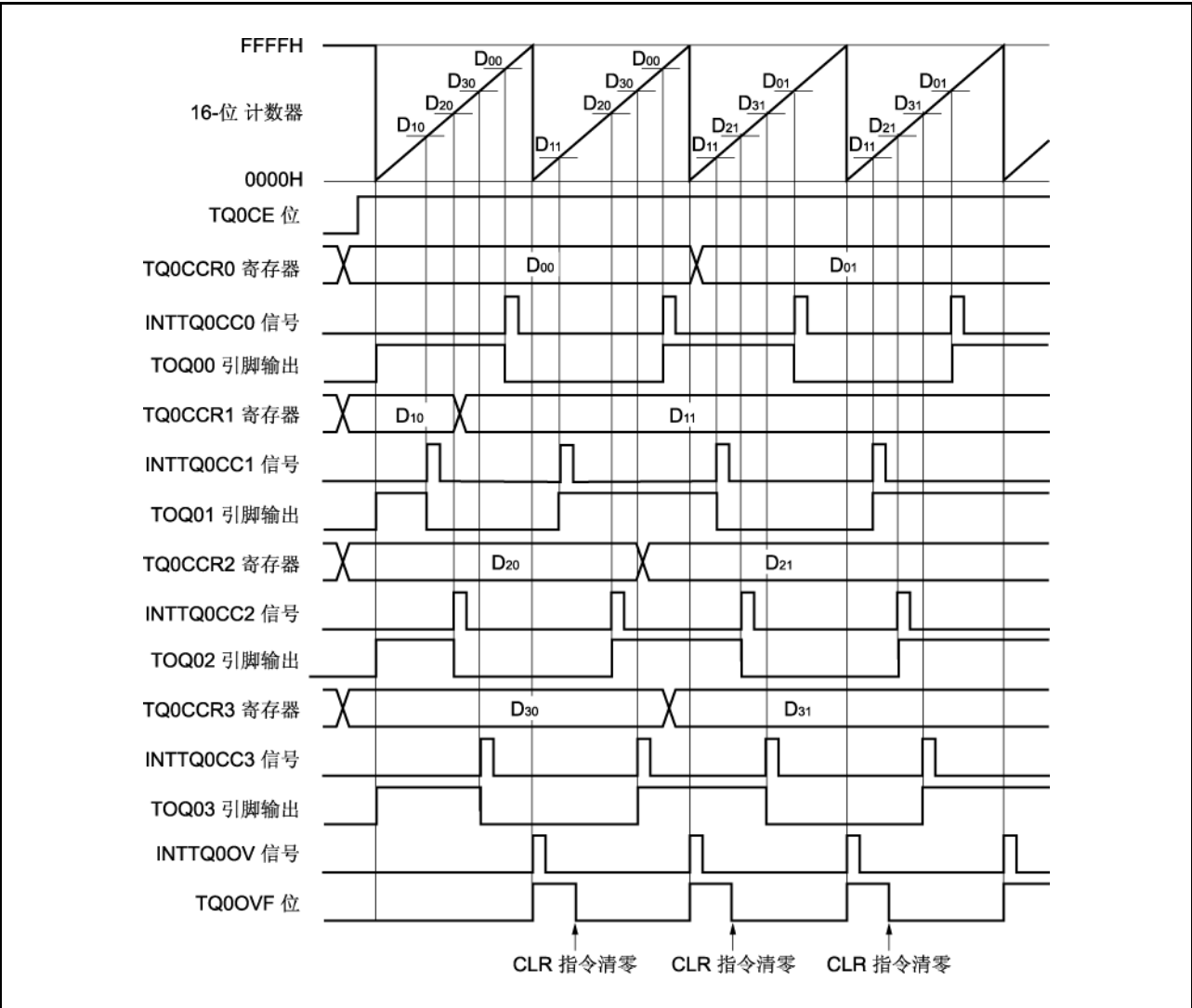


- 注
1. 由于外部事件计数输入引脚（TIQ00）、捕捉触发输入引脚（TIQ00）和定时器输出引脚（TOQ00）共用同一复用引脚，两种功能不能同时使用。
 2. 由于捕捉触发输入引脚（TIQ0k）和定时器输出引脚（TOQ00）共用同一复用引脚，两种功能不能同时使用。

- 比较操作
- 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。这种情况下，溢出标志位 (TQ0OPT0.TQ0OVF 位) 也将被设置为 1。请确认溢出标志置 1 并通过软件使用 CLR 指令将溢出标志清零。
- 可以在计数器计数过程中改写 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。

备注 m = 0 至 3

图 8-33. 自由运行定时器模式的基本时序（比较功能）



- 捕捉操作

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。这种情况下, 溢出标志位 (TQ0OPT0.TQ0OVF 位) 也将被设置为 1。请确认溢出标志置 1 并通过软件使用 CLR 指令将溢出标志清零。

备注 m = 0 至 3

图 8-34. 自由运行定时器模式的基本时序 (捕捉功能)

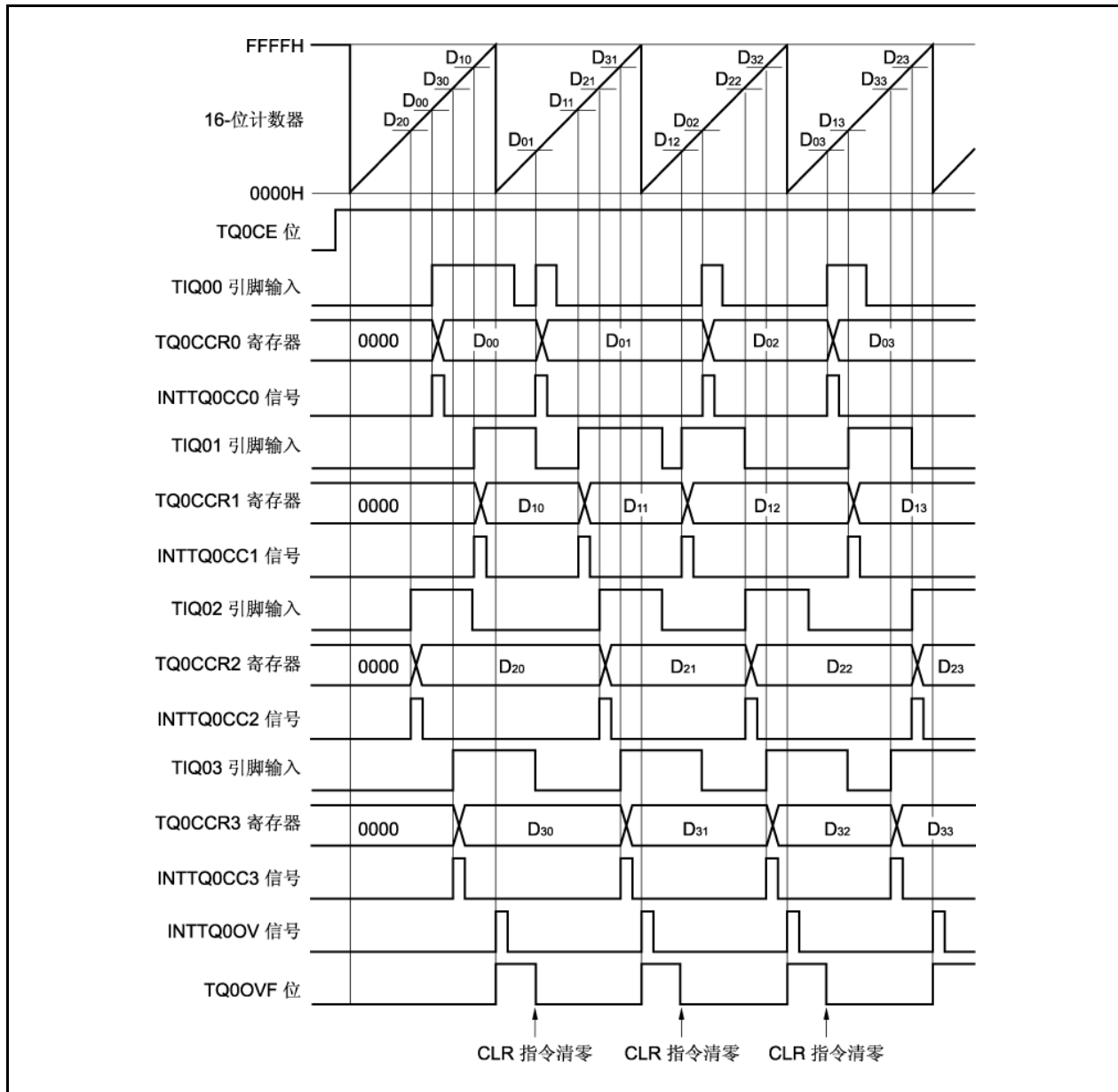


图 8-35. 自由运行定时器模式的寄存器设置 (1/3)

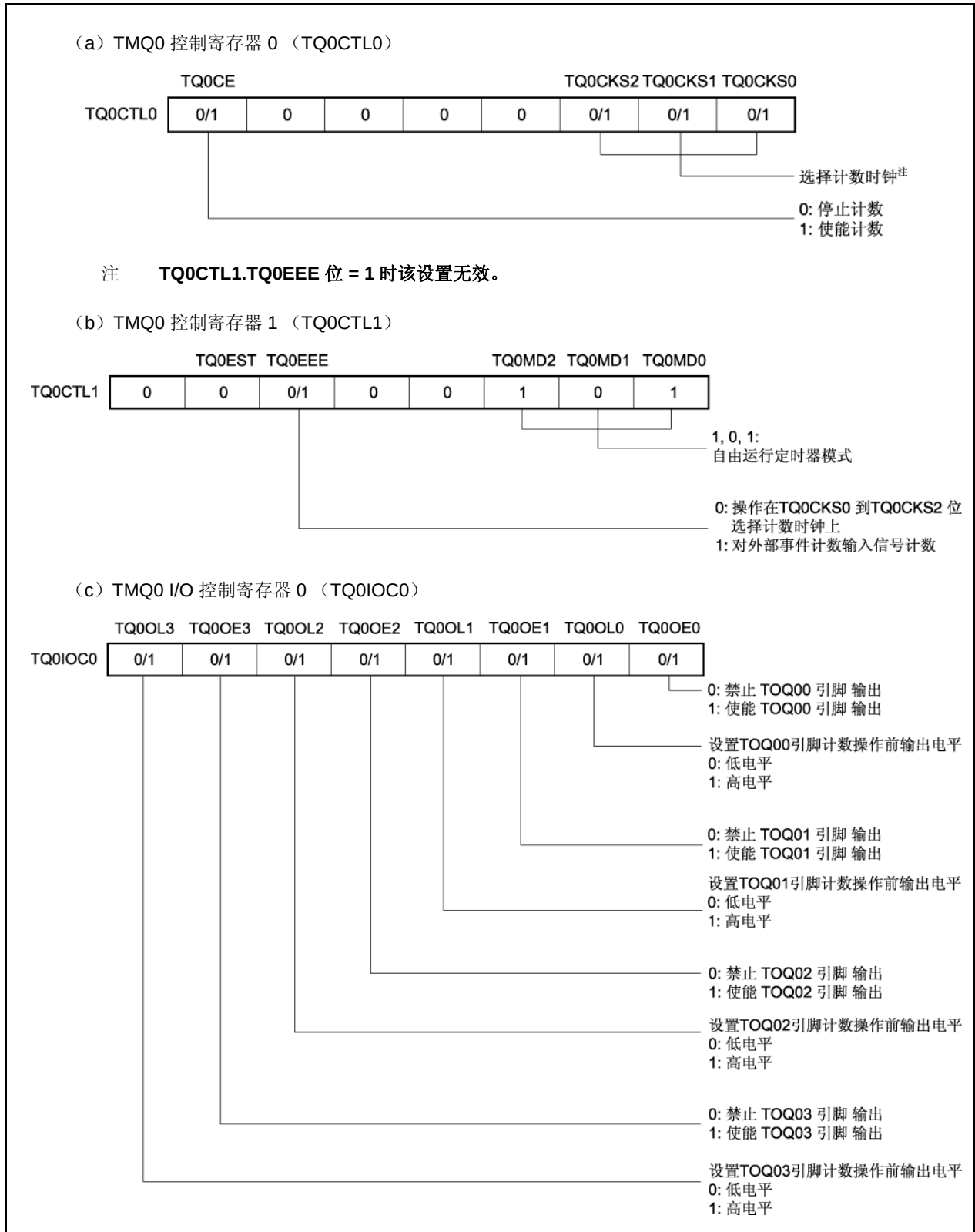
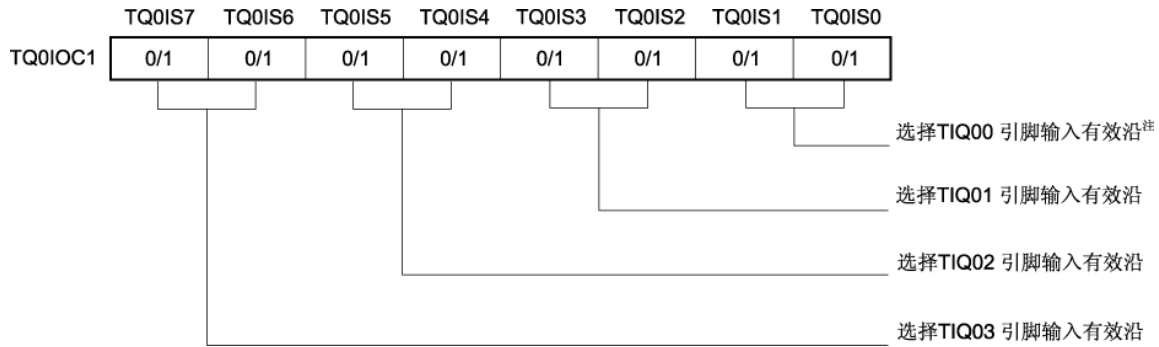


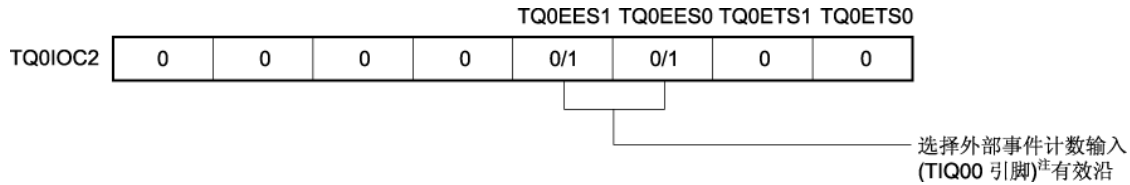
图 8-35. 自由运行定时器模式的寄存器设置 (2/3)

(d) TMQ0 I/O 控制寄存器 1 (TQ0IOC1)



注 设置未使用复用外部输入信号的有效沿选择为“无边沿检测”。

(e) TMQ0 I/O 控制寄存器 2 (TQ0IOC2)



注 设置未使用复用外部输入信号的有效沿选择为“无边沿检测”。

(f) TMQ0 选项寄存器 0 (TQ0OPT0)

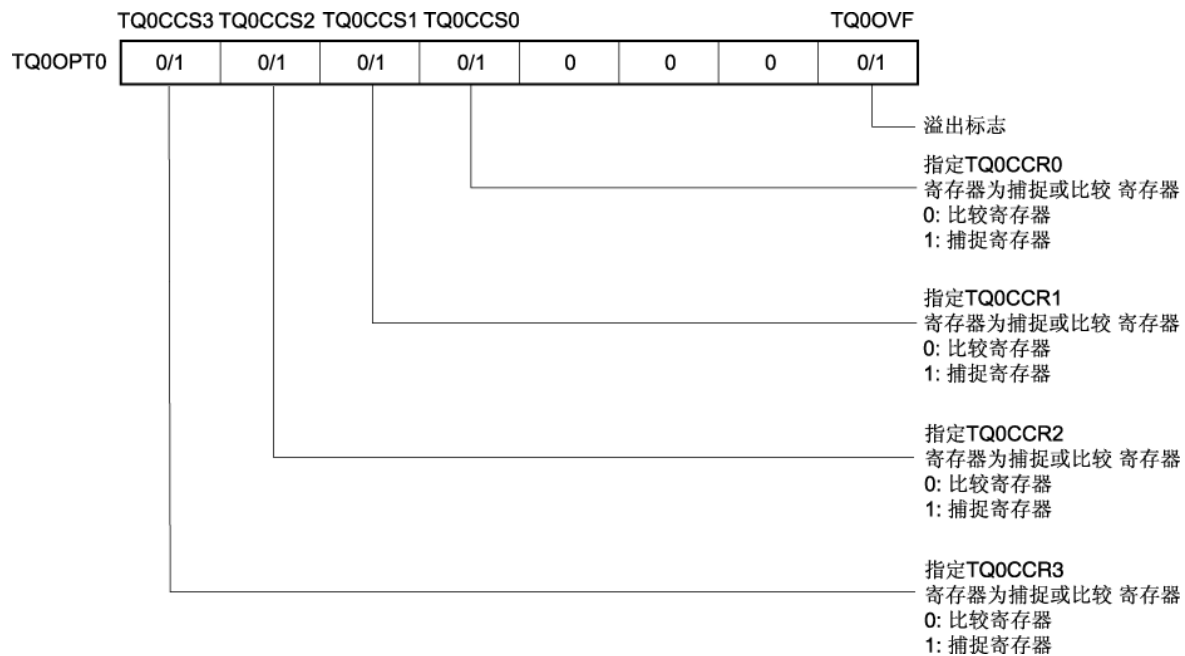


图 8-35. 自由运行定时器模式的寄存器设置 (3/3)

(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系统

(a) 当捕捉 / 比较寄存器用作比较寄存器时

图 8-36. 自由运行定时器模式的软件处理流程（比较功能）（1/2）

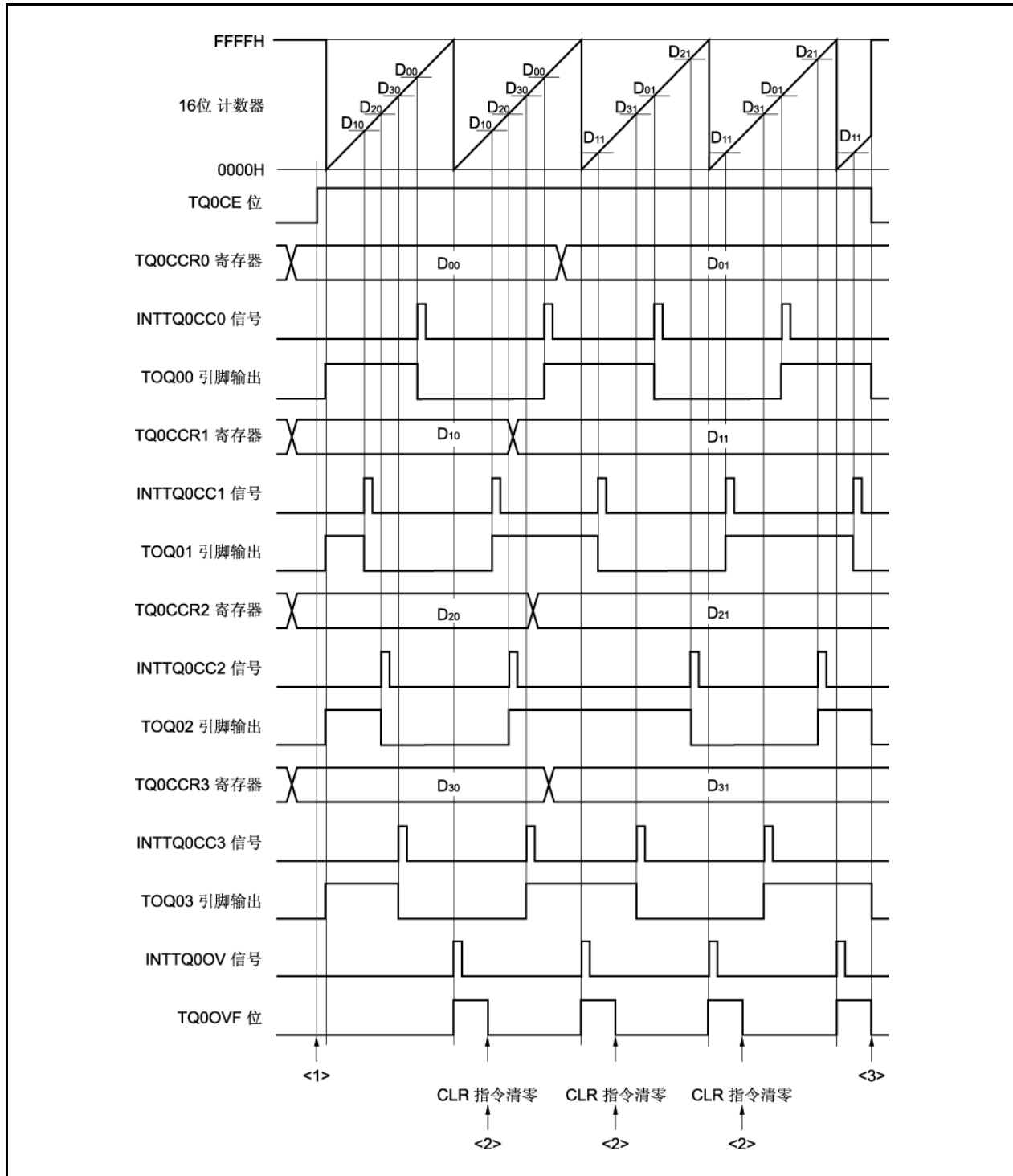
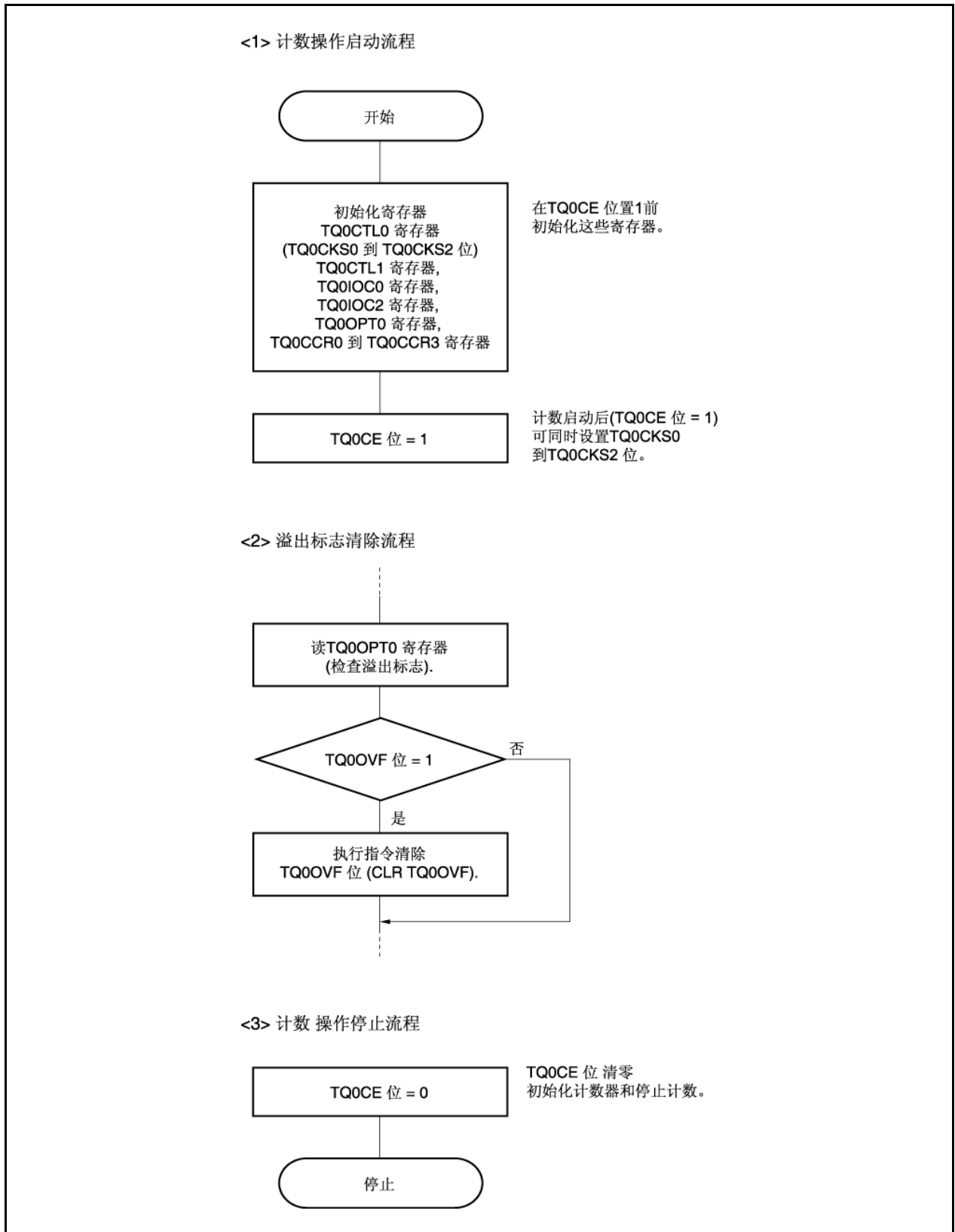


图 8-36. 自由运行定时器模式的软件处理流程（比较功能）（2/2）



(b) 当捕捉 / 比较寄存器用作捕捉寄存器时

图 8-37. 自由运行定时器模式的软件处理流程（捕捉功能）（1/2）

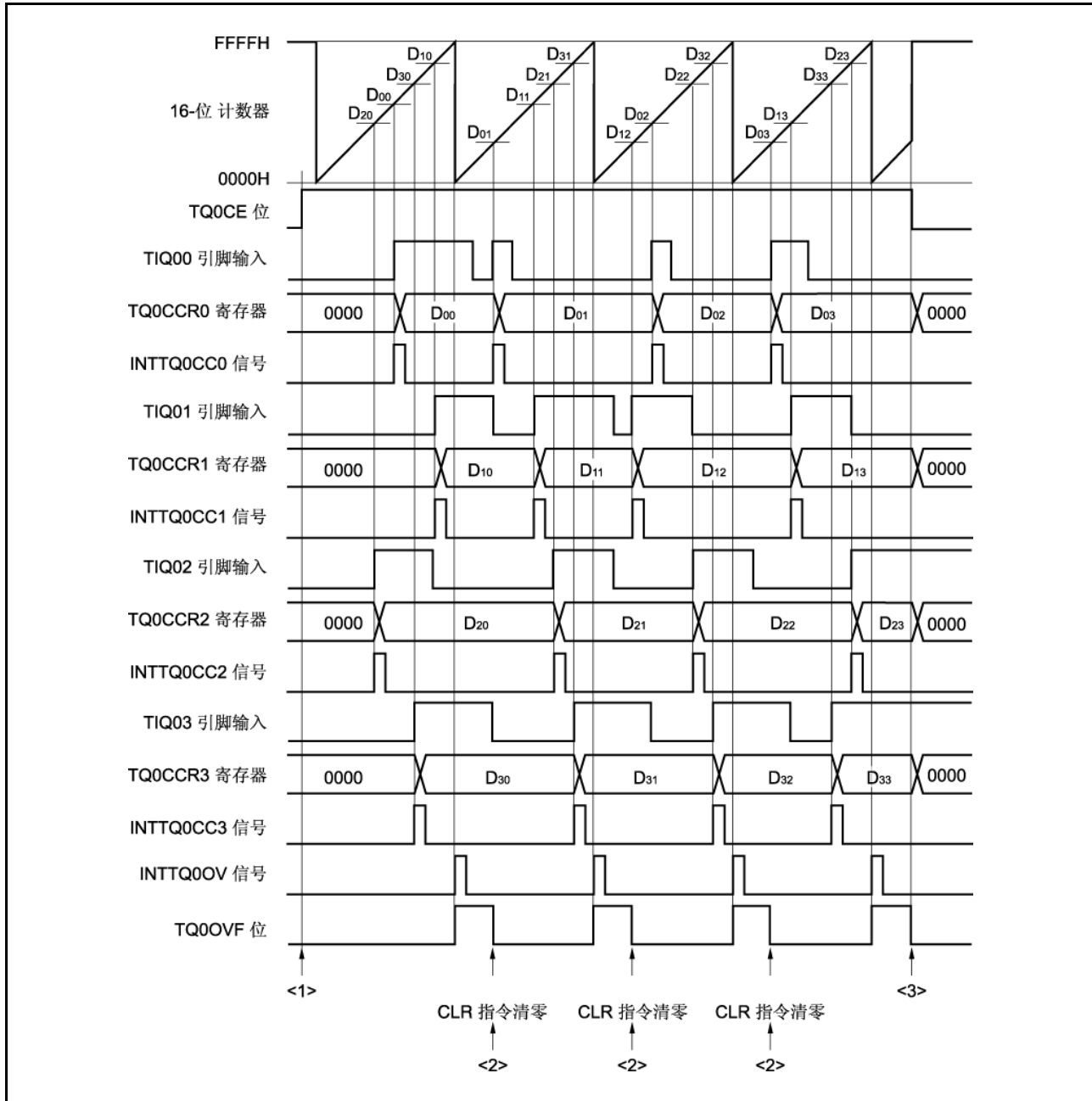
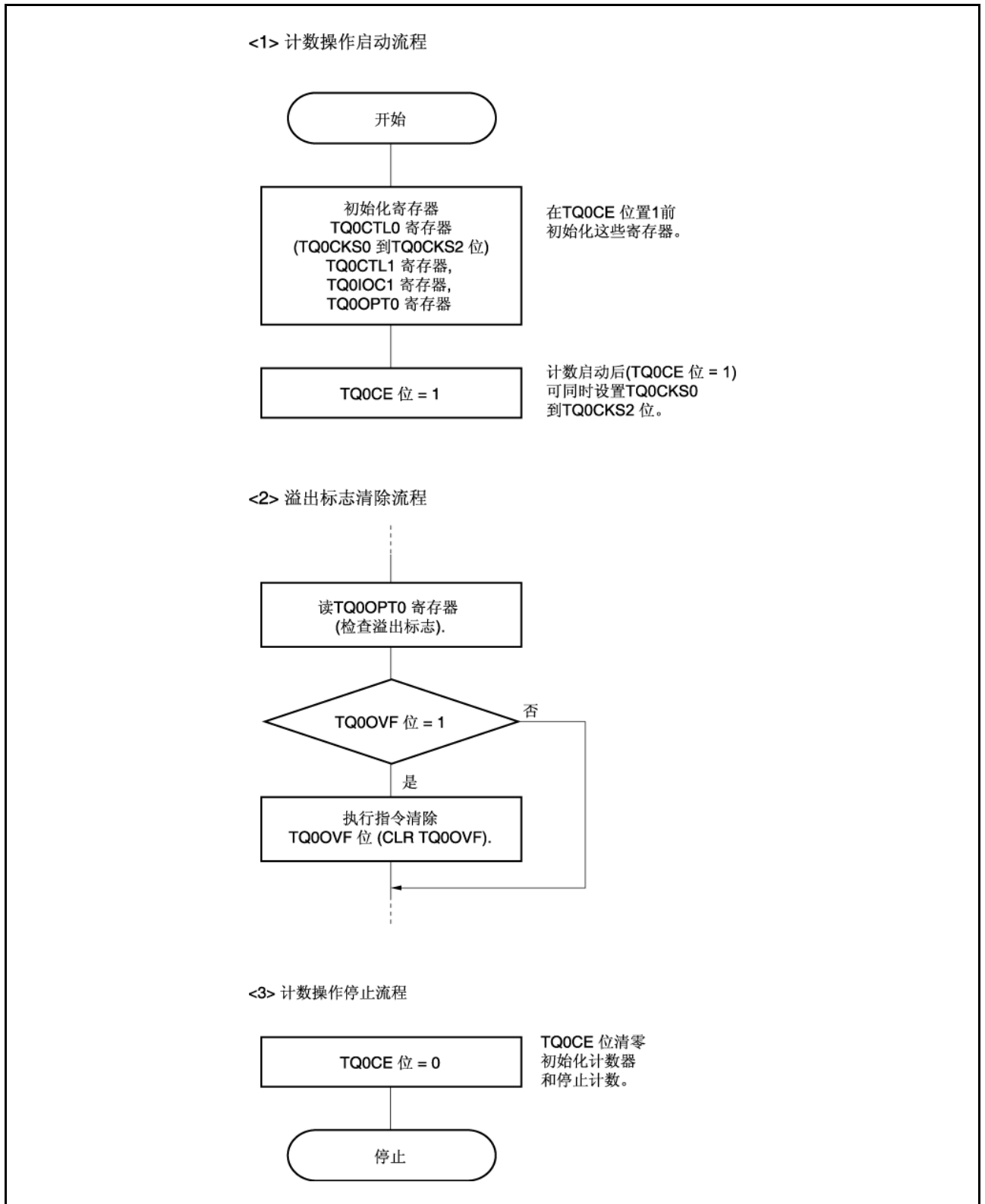


图 8-37. 自由运行定时器模式的软件处理流程（捕捉功能）（2/2）

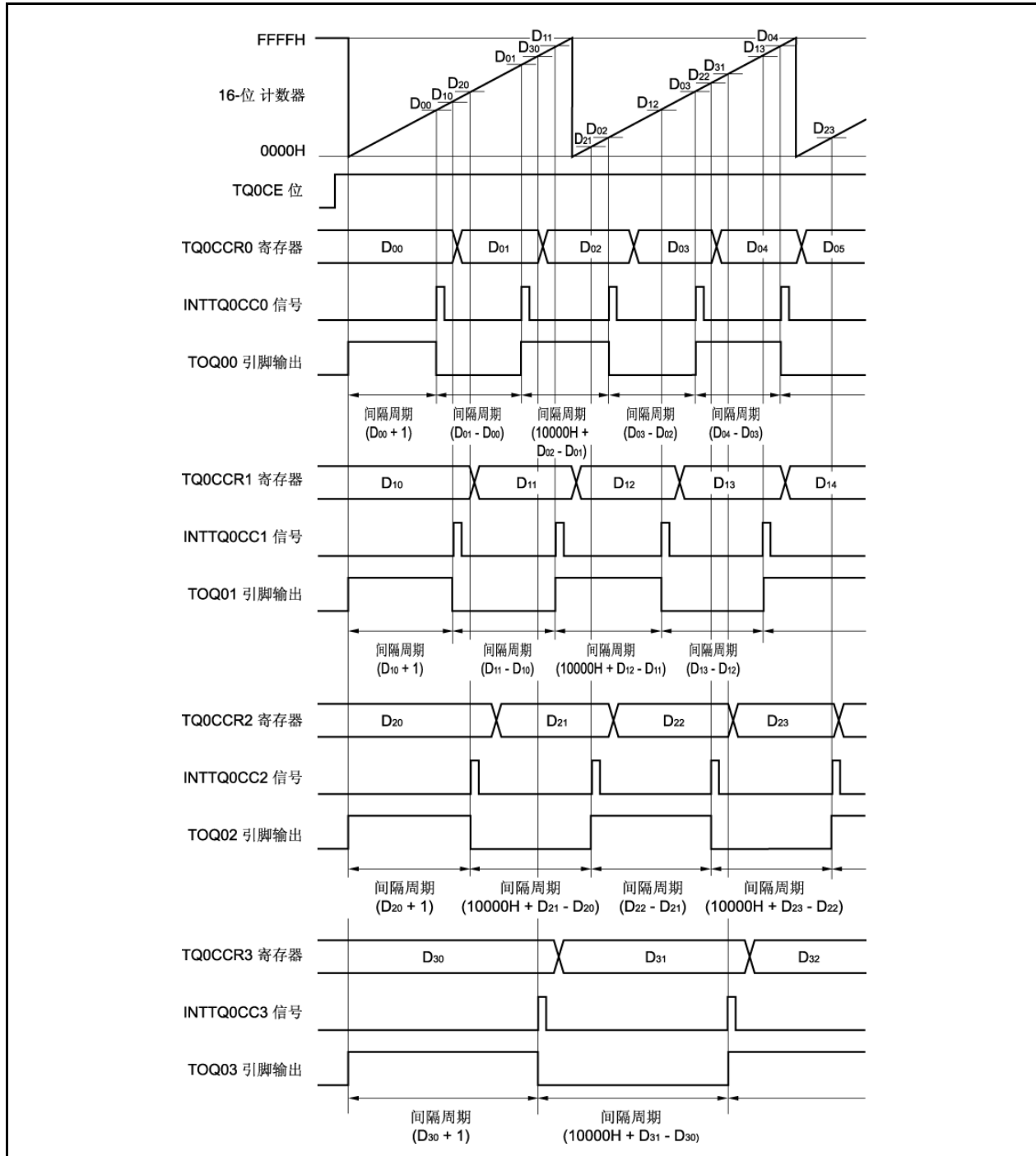


(2) 自由运行定时器模式的操作时序

(a) 比较寄存器的时间间隔操作

当 TQ0CCRm 寄存器被作为比较寄存器, 16 位定时器 / 事件计数器 Q 用作间隔定时器时, 每次检测到 INTTQ0CCm 信号, 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。

备注 m = 0 至 3



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重新设置到 TQ0CCRm 寄存器。

向 TQ0CCRm 寄存器中重新设置的值可由以下算式求出，这里假设间隔时间为“Dm”。

比较寄存器的最初设置值： $D_m - 1$

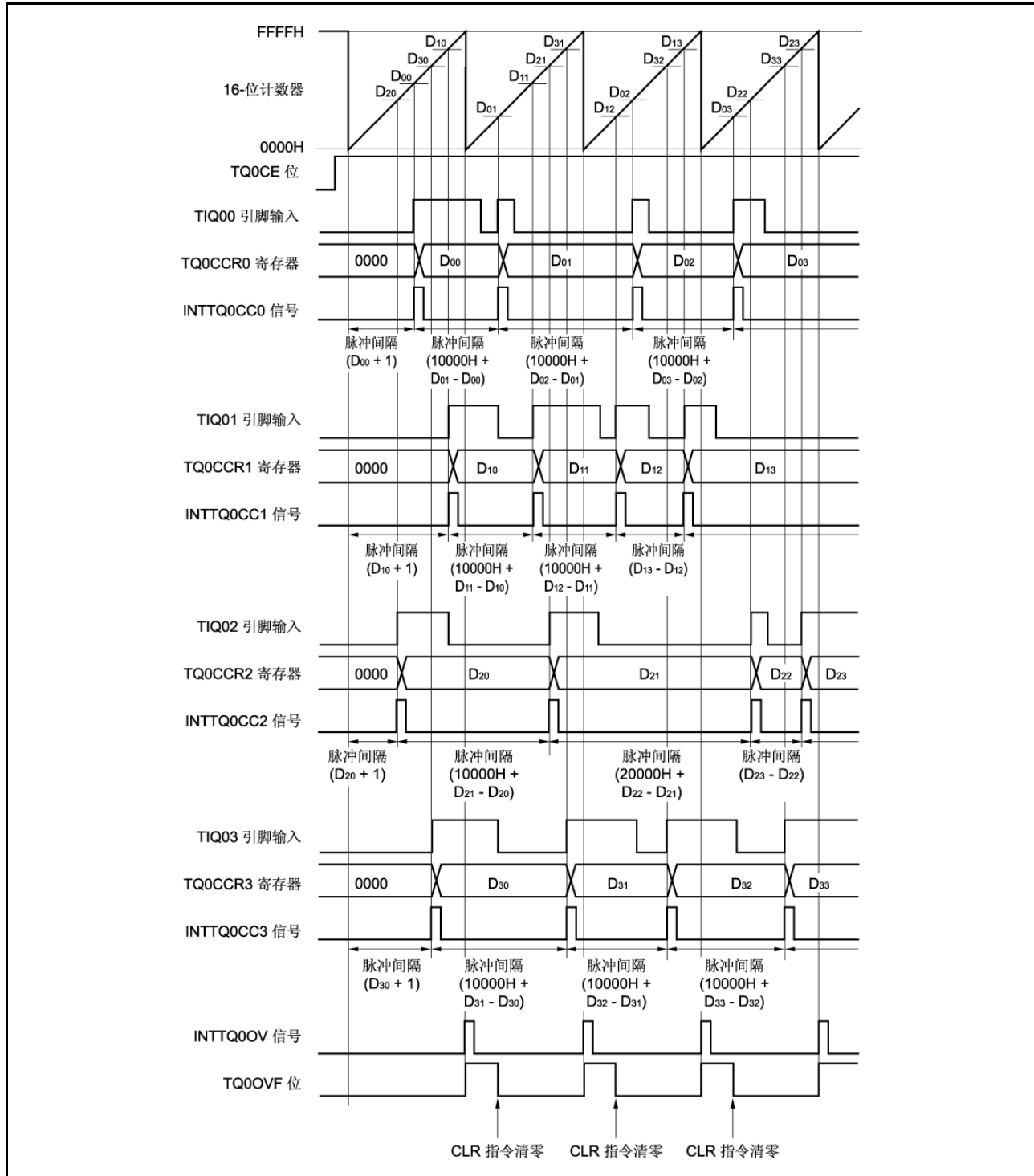
第二次设置比较寄存器以后（含第二次）的设置值：前次设置值 + D_m

（如果计算结果大于 FFFFH，那么将结果减 10000H 后再设置到寄存器中。）

备注 m = 0 至 3

(b) 使用捕捉寄存器进行脉宽测量

当 TQ0CCRm 寄存器用作捕捉寄存器进行脉宽测量时，需要通过软件处理在每次检测到中断信号时对捕捉寄存器进行读取并计算间隔时间。



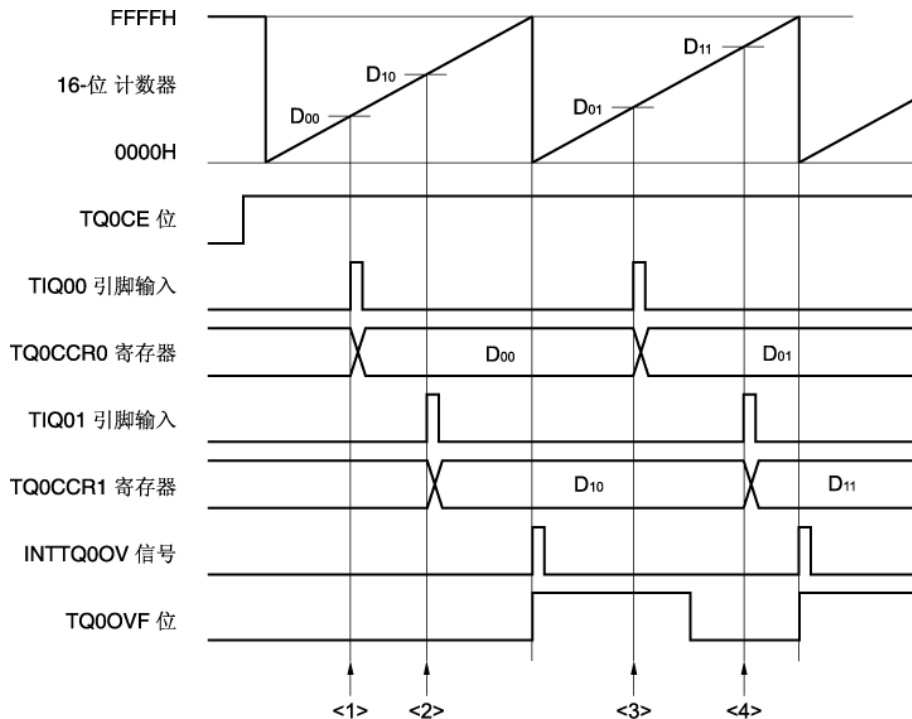
当使用自由运行定时器模式进行脉宽测量操作时，可以在同一通道对四个不同的脉冲宽度进行测量。
通过与 INTTQ0CCm 信号同步读取 TQ0CCRm 寄存器的值并计算该值与上一次读取值的差，就可以求得脉冲的宽度。

备注 m = 0 至 3

(c) 使用两个及以上捕捉寄存器产生溢出时的操作

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

使用两个及以上捕捉寄存器时的错误处理举例



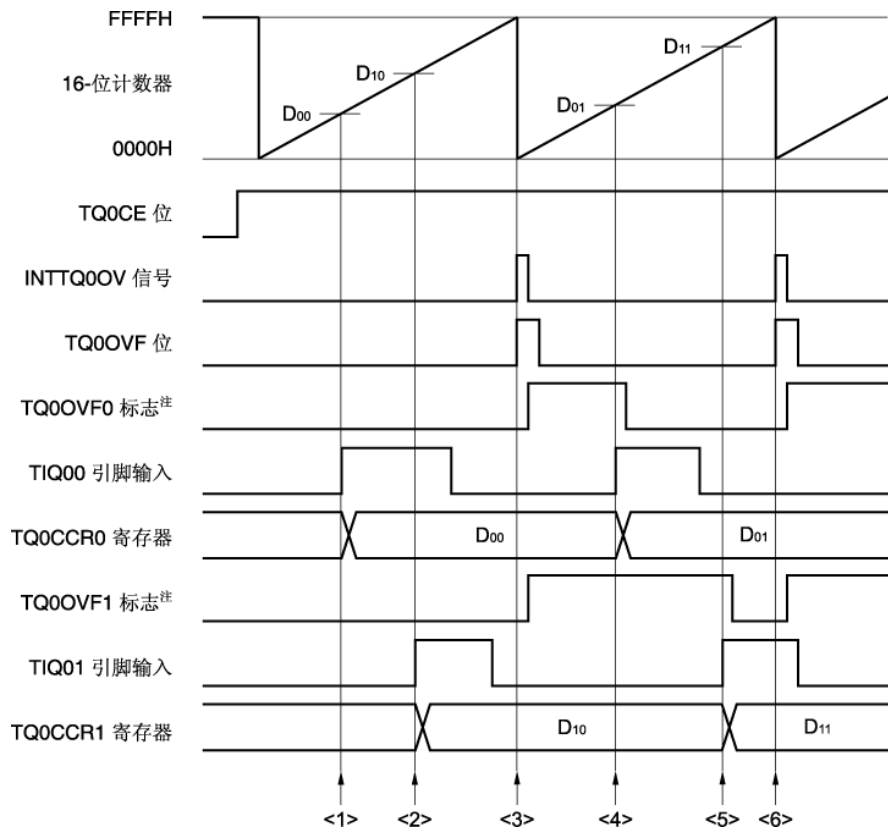
使用自由运行定时器模式对两个脉冲宽度进行测量时，可能出现以下的问题。

- <1> 读取 TQ0CCR0 寄存器（设置为 TIQ00 引脚输入的初始值）。
- <2> 读取 TQ0CCR1 寄存器（设置为 TIQ01 引脚输入的初始值）。
- <3> 读取 TQ0CCR0 寄存器。
读取溢出标志位。如果溢出标志为 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。
- <4> 读取 TQ0CCR1 寄存器。
读取溢出标志位。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。

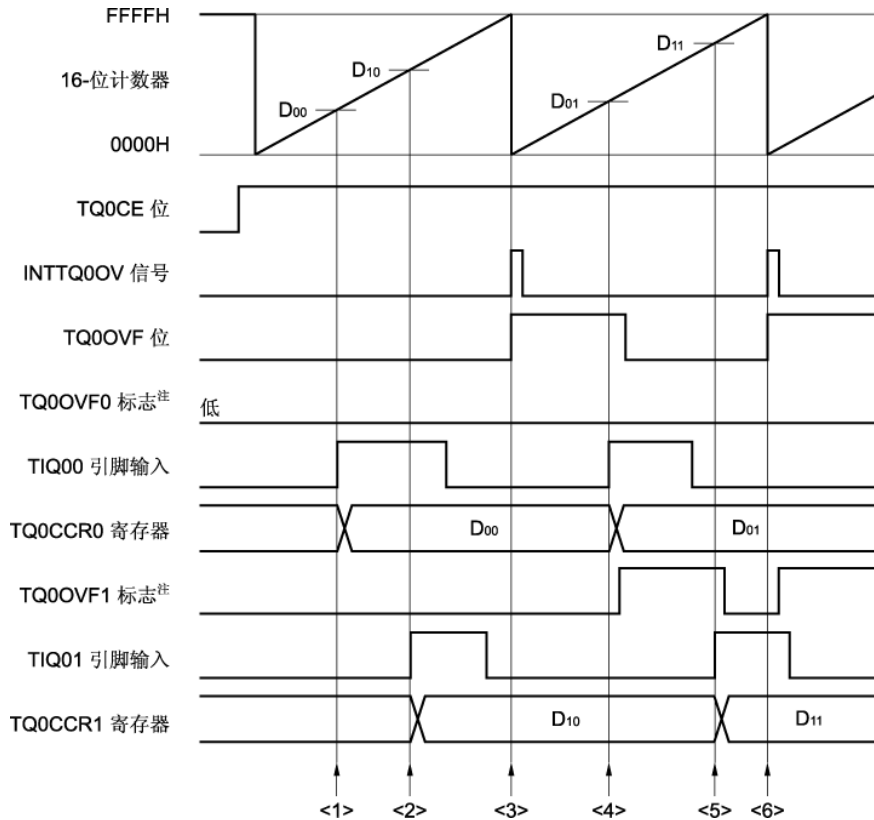
使用两个捕捉寄存器时的处理举例（利用溢出中断）



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 位。如果 TQ0OVF1 标志为 1，将它清零（步骤<4>中将 TQ0OVF0 标志清零但 TQ0OVF1 标志始终保持为 1）。
由于 TQ0OVF1 标志为 1，脉冲宽度按照 $(10000H + D_{11} - D_{10})$ 来计算（正确）。
- <6> 同步<3>

使用两个捕捉寄存器时的处理举例（不利用溢出中断）



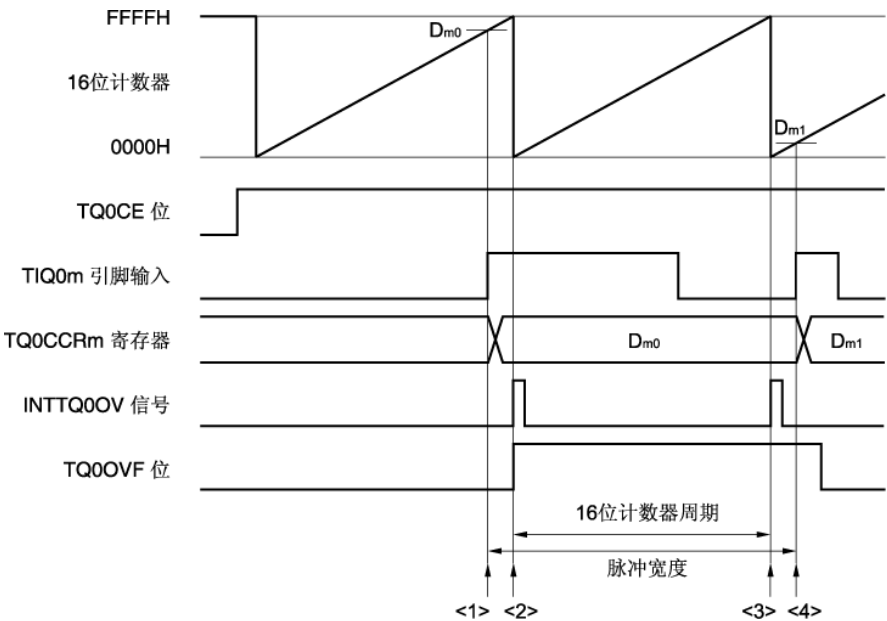
注 TQ0OVF0 和 TQ0OVF1 标志通过软件设置于内部 RAM。

- <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。
读取溢出标志位。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。
读取溢出标志位。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 $(10000H + D_{11} - D_{10})$ 来计算（正确）。
- <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（TIQ0m 引脚输入的默认值设置）。

<2> 产生了溢出。但不进行任何软件处理。

<3> 产生了第二次溢出。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 $(10000H + D_{m1} - D_{m0})$ 来计算（错误）。

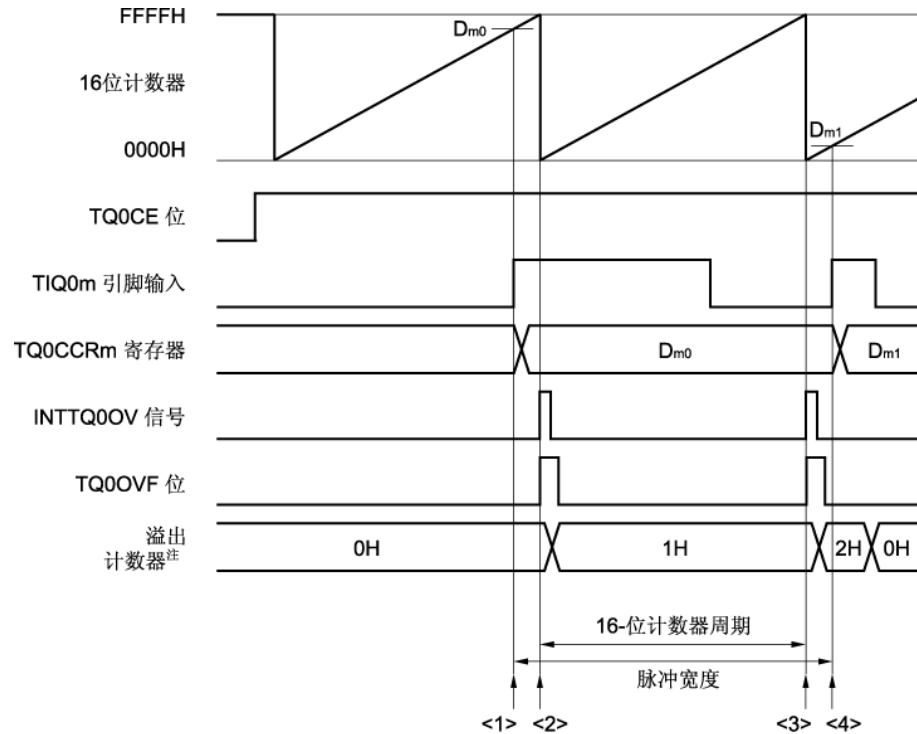
但实际上，由于产生了两次溢出脉冲宽度应该为 $(20000H + D_{m1} - D_{m0})$ 。

备注 m = 0 至 3

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来避免产生错误。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（TIQ0m 引脚输入的默认值设置）。

<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）。

备注 m = 0 至 3

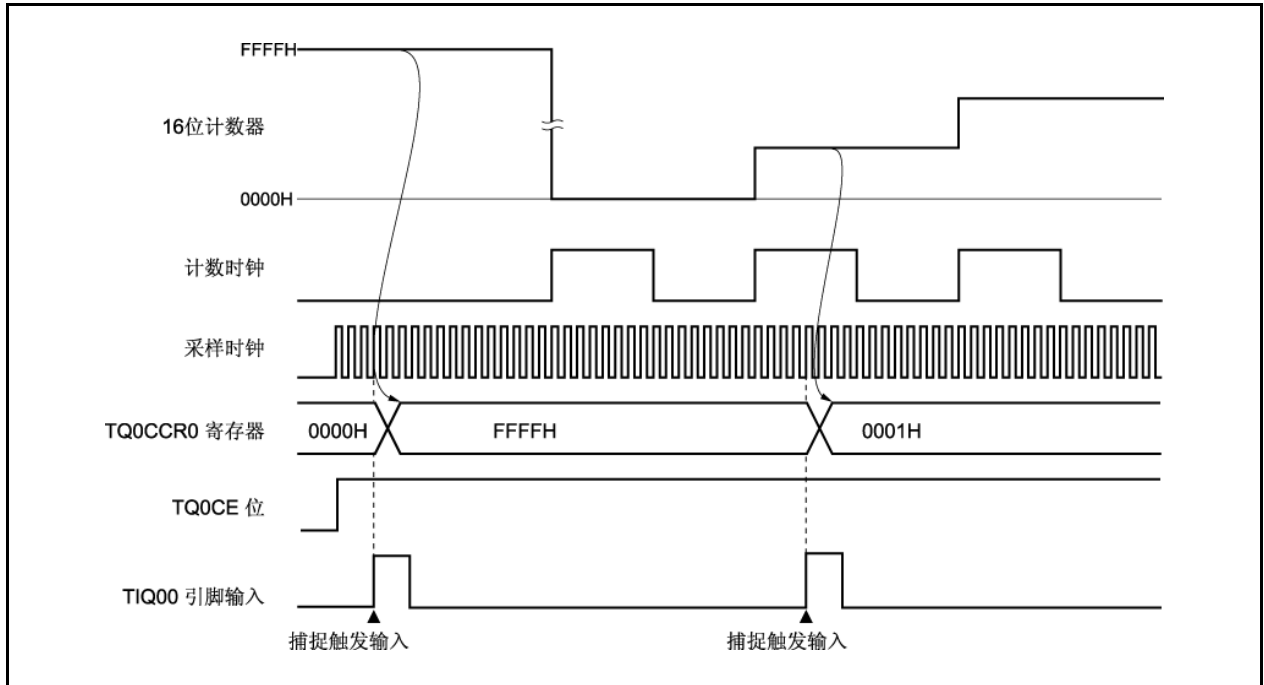
(e) 将溢出标志位清零

读 TQ0OVF 位=1 后，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。

<R>

(3) 捕捉操作时注意

如果使用捕捉操作并选用慢速时钟作为计数时钟，在 TQ0CTL0.TQ0CE 位置 1 后立刻输入捕捉触发信号，TQ0CCRm 寄存器可能捕捉到 FFFFH 而不是 0000H (m=0 到 3)。



8.6.7 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。

图 8-39 中，选择 TIQ00 到 TIQ03 引脚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“非边沿检测”。

备注 m = 0 至 3
 k = 1 至 3

<R>

图 8-38. 脉宽测量模式的配置图

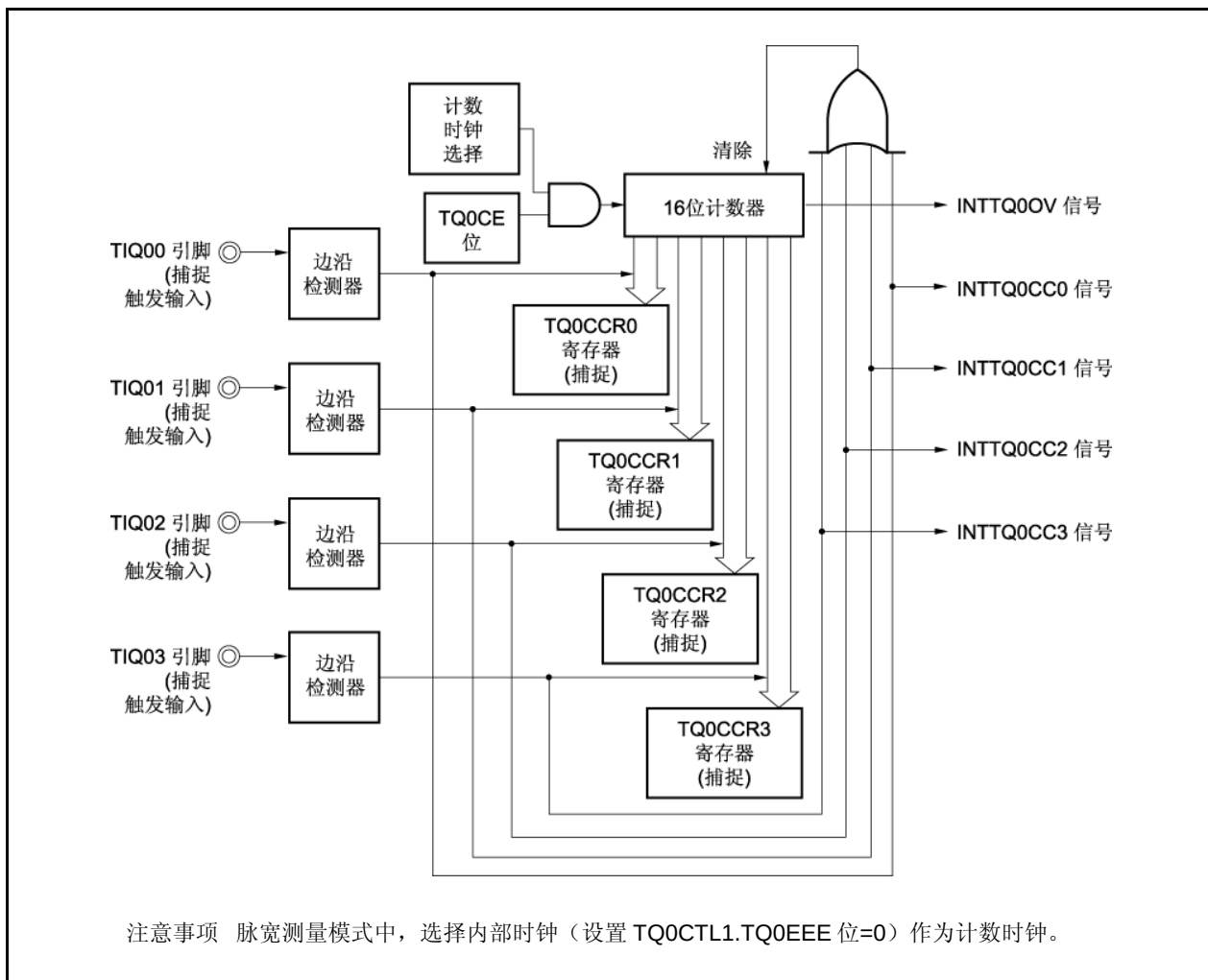
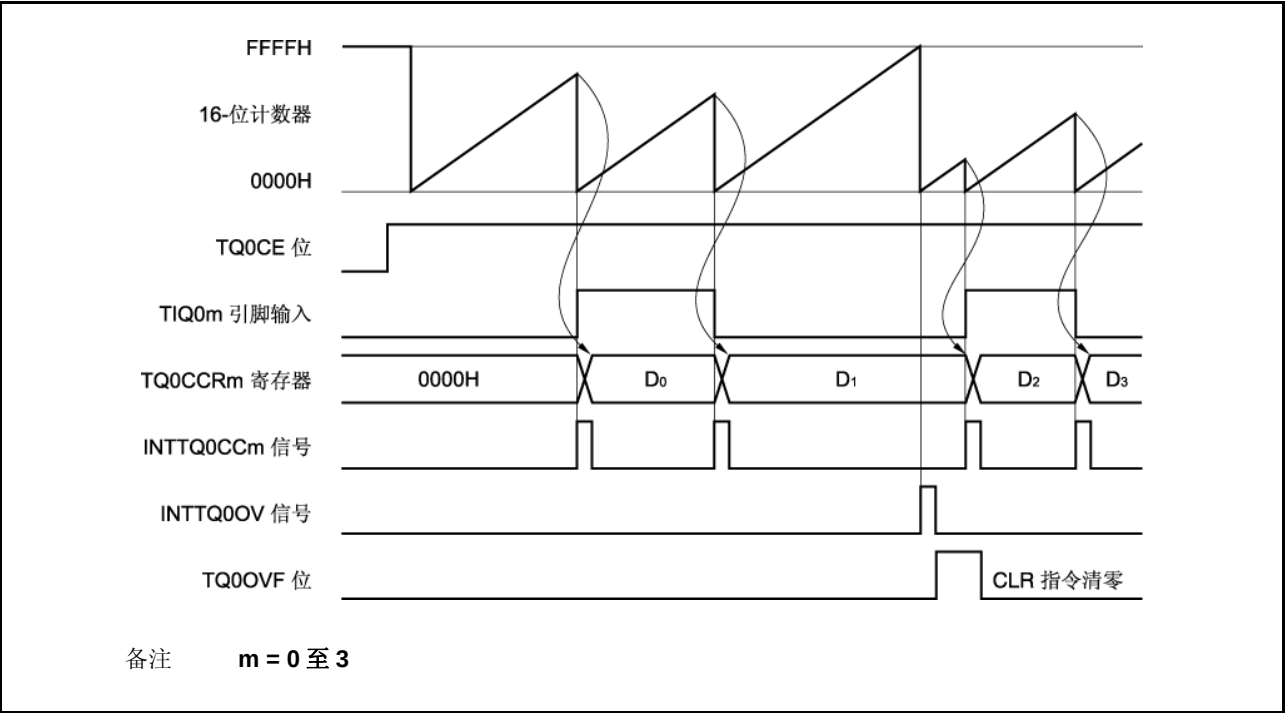


图 8-39. 脉宽测量模式的基本时序



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。

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

如果直到 16 位计数器计数到 FFFFH 仍没有检测到 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。

$$\text{脉冲宽度} = (10000\text{H} \times \text{TQ0OVF 位被设置为 1 的次数} + \text{捕捉计数值}) \times \text{计数时钟周期}$$

备注 m = 0 至 3

图 8-40. 脉宽测量模式的寄存器设置 (1/2)

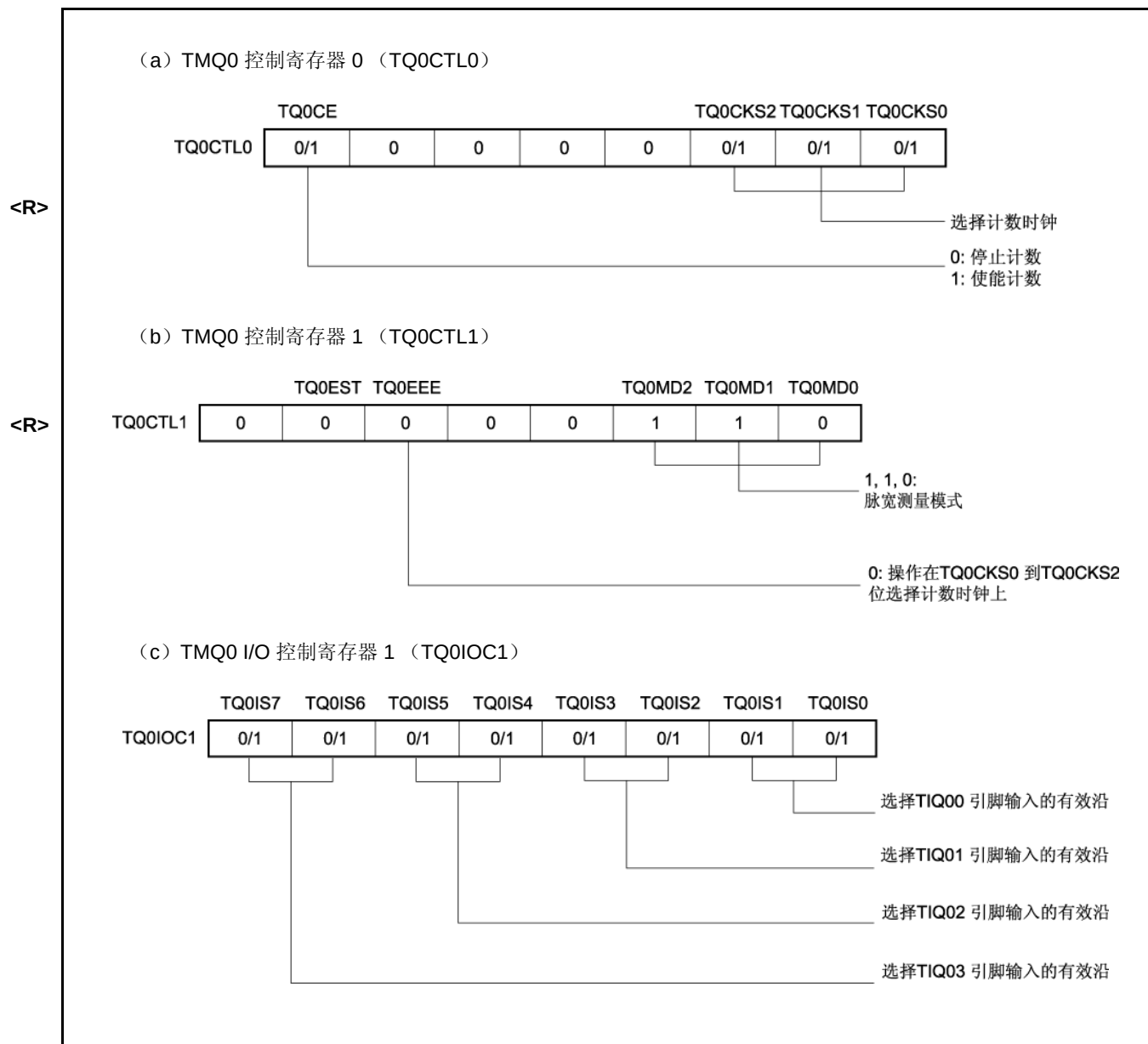
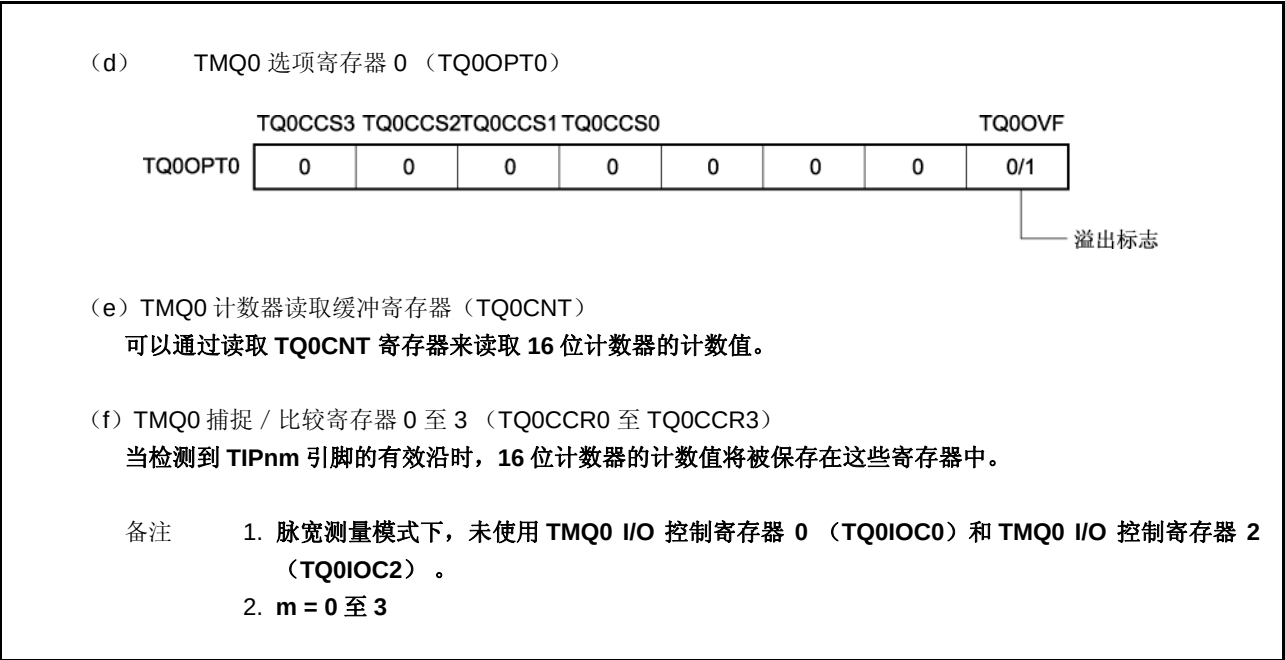
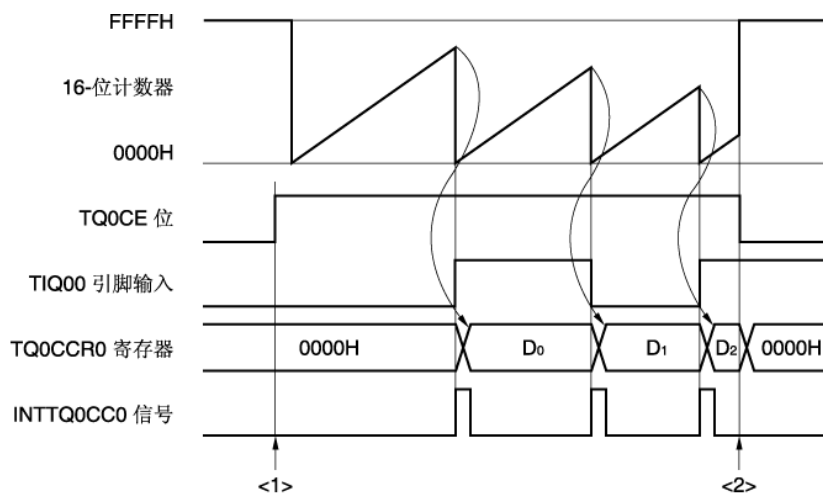


图 8-40. 脉宽测量模式的寄存器设置 (2/2)



(1) 脉宽测量模式的操作流程

图 8-41. 脉宽测量模式的软件处理流程



<1> 计数操作启动流程



在TQ0CE 位置1前
初始化这些寄存器。

计数启动后(TQ0CE 位 = 1)
可同时设置TQ0CKS0
到TQ0CKS2 位。

<2> 计数操作停止流程



TQ0CE 位清零
初始化计数器
和停止计数。

<R>

(2) 脉宽测量模式的操作时序

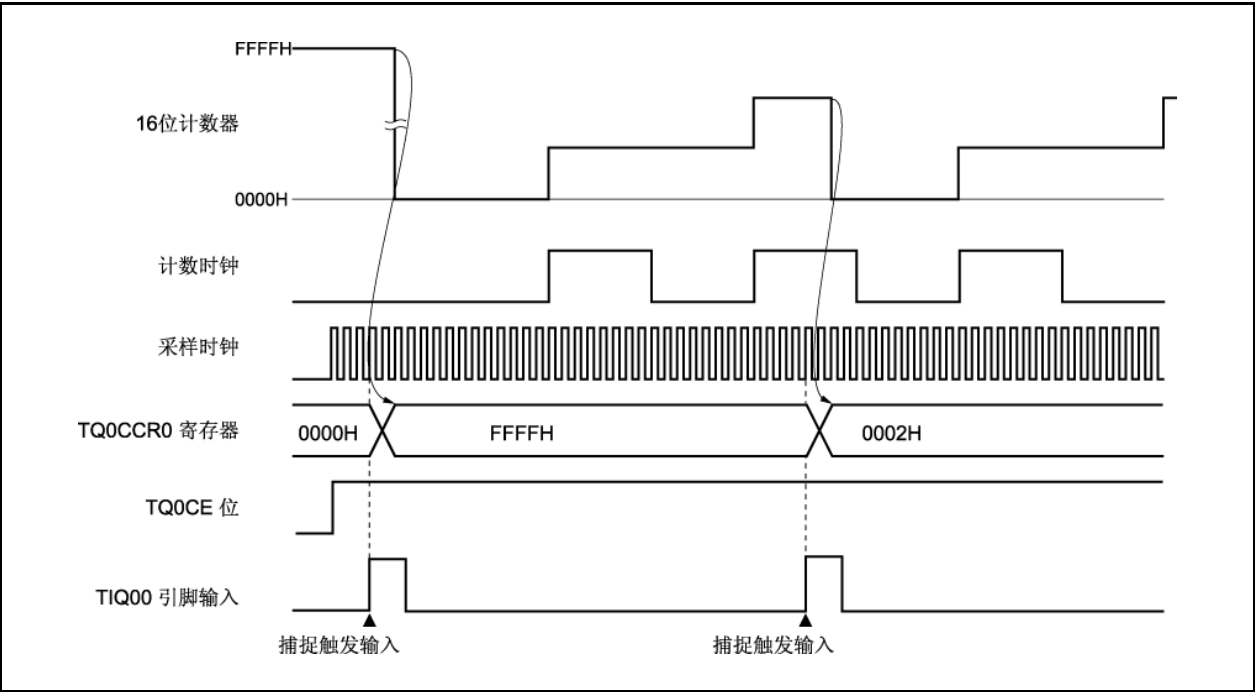
(a) 将溢出标志位清零

读 TQ0OVF 位=1 后，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。

<R>

(3) 注

如果慢速时钟作为计数时钟，在 TQ0CTL0.TQ0CE 位置 1 后立刻输入捕捉触发信号，TQ0CCRm 寄存器可能捕捉到 FFFFH 而不是 0000H（m=0 到 3）。



8.7 选择器功能

有关选择器功能，参看 7.7 选择器功能。

第 9 章 16 位间隔定时器 M (TMM)

定时器 M (TMM) 是一个 16 位间隔定时器。
V850ES/SG2 和 V850ES/SG2-H 都包含 TMM0。

9.1 概述

TMM0 具有以下功能：

- 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。

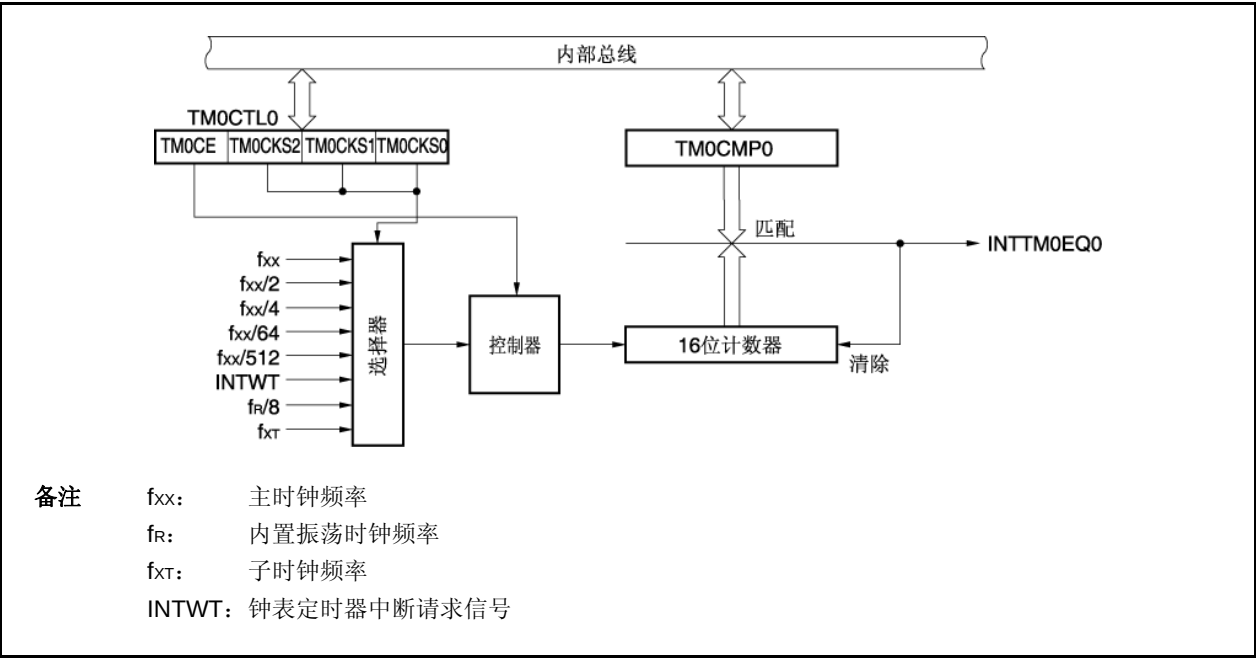
9.2 配置

TMM0 包含以下硬件。

表 9-1. TMM0 的配置

器件	配置
定时器寄存器	16 位计数器
寄存器	TMM0 比较寄存器 0（TM0CMP0）
控制寄存器	TMM0 控制寄存器 0（TM0CTL0）

图 9-1. TMM0 的框图

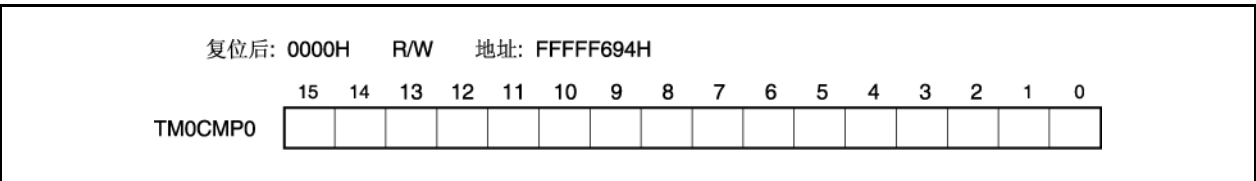


(1) 16 位计数器

该计数器是一个按内部时钟计数的 16 位计数器。
该计数器不可以被读写。

(2) TMM0 比较寄存器 0（TM0CMP0）

TM0CMP0 寄存器是一个 16 位的比较寄存器。
可以对该寄存器进行 16 位的读写操作。
复位输入清除该寄存器为 0000H。
可使用软件向 TM0CMP0 寄存器写入相同的数值。
在 TMM0 工作（TM0CTL0.TM0CE 位 = 1）过程中，禁止重写 TM0CMP0 寄存器。



9.3 寄存器

(1) TMM0 控制寄存器 (TM0CTL0)

TM0CTL0 寄存器是用于控制 TMM0 操作的 8 位寄存器。

可对该寄存器进行 8 位或 1 位的读写操作。

复位输入将把该寄存器设置为 00H。

可使用软件向 TM0CTL0 寄存器写入相同的数值。

复位后: 00H R/W 地址: FFFFF690H

	<7>	6	5	4	3	2	1	0
TM0CTL0	TM0CE	0	0	0	0	TM0CKS2	TM0CKS1	TM0CKS0

TM0CE	指定是否允许内部时钟操作
0	禁止TMM0 操作(16位计数器异步复位)。 停止时钟操作应用。
1	允许TMM0 操作。开始操作时钟的应用。 TMM0 开始工作。 操作开始
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 位清零。

备注 f_{xx}: 主时钟频率
 f_R: 内置振荡时钟频率
 f_{XT}: 子时钟频率

9.4 操作

注意事项 不可以将 TM0CMP0 寄存器设置为 FFFFH。

9.4.1 间隔定时器模式

在间隔定时器模式中，当 TM0CTL0.TM0CE 位被置 1 后，就会按照 TM0CM0P 寄存器设置的间隔时间周期性产生中断请求信号（INTTM0EQ0）。

图 9-2. 间隔定时器的结构图

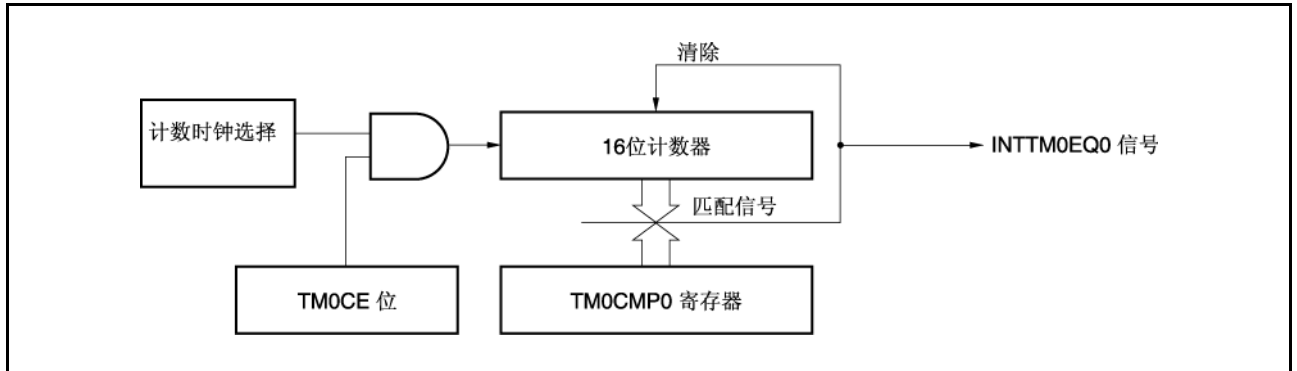
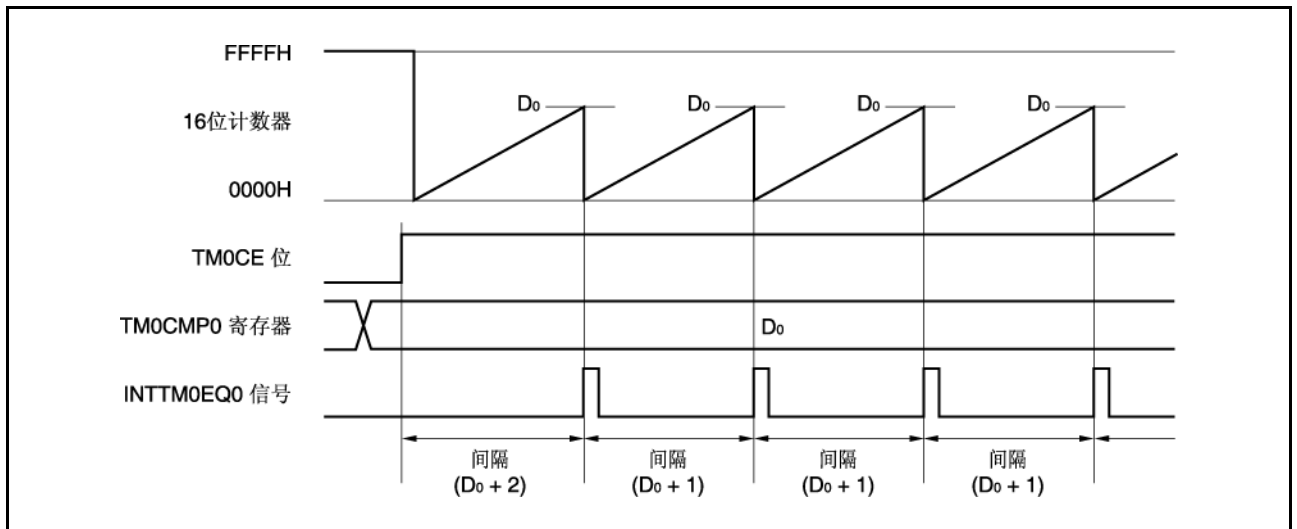


图 9-3. 间隔定时器模式的基本操作时序



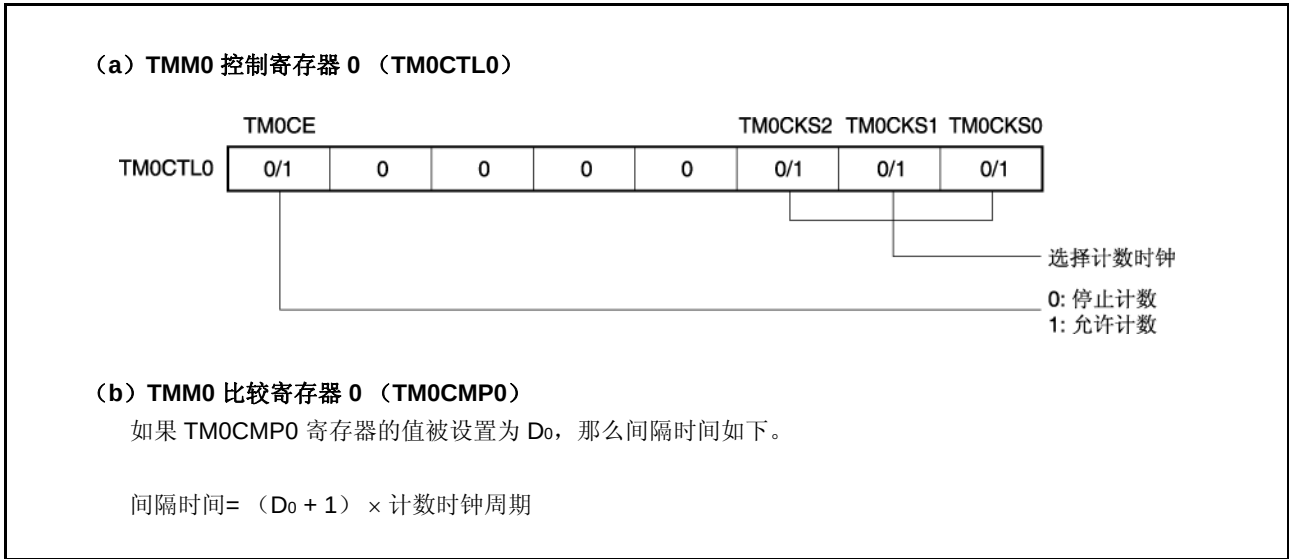
当 TM0CE 位被设置为 1，16 位计数器会与计数时钟同步将计数值从 FFFFH 清零为 0000H，并开始计数。

当 16 位计数器的计数值与 TM0CMP0 寄存器中的值相匹配时，16 位计数器被清零为 0000H 并同时产生一个比较匹配中断请求信号（INTTM0EQ0）。

间隔时间可由下列表达式计算：

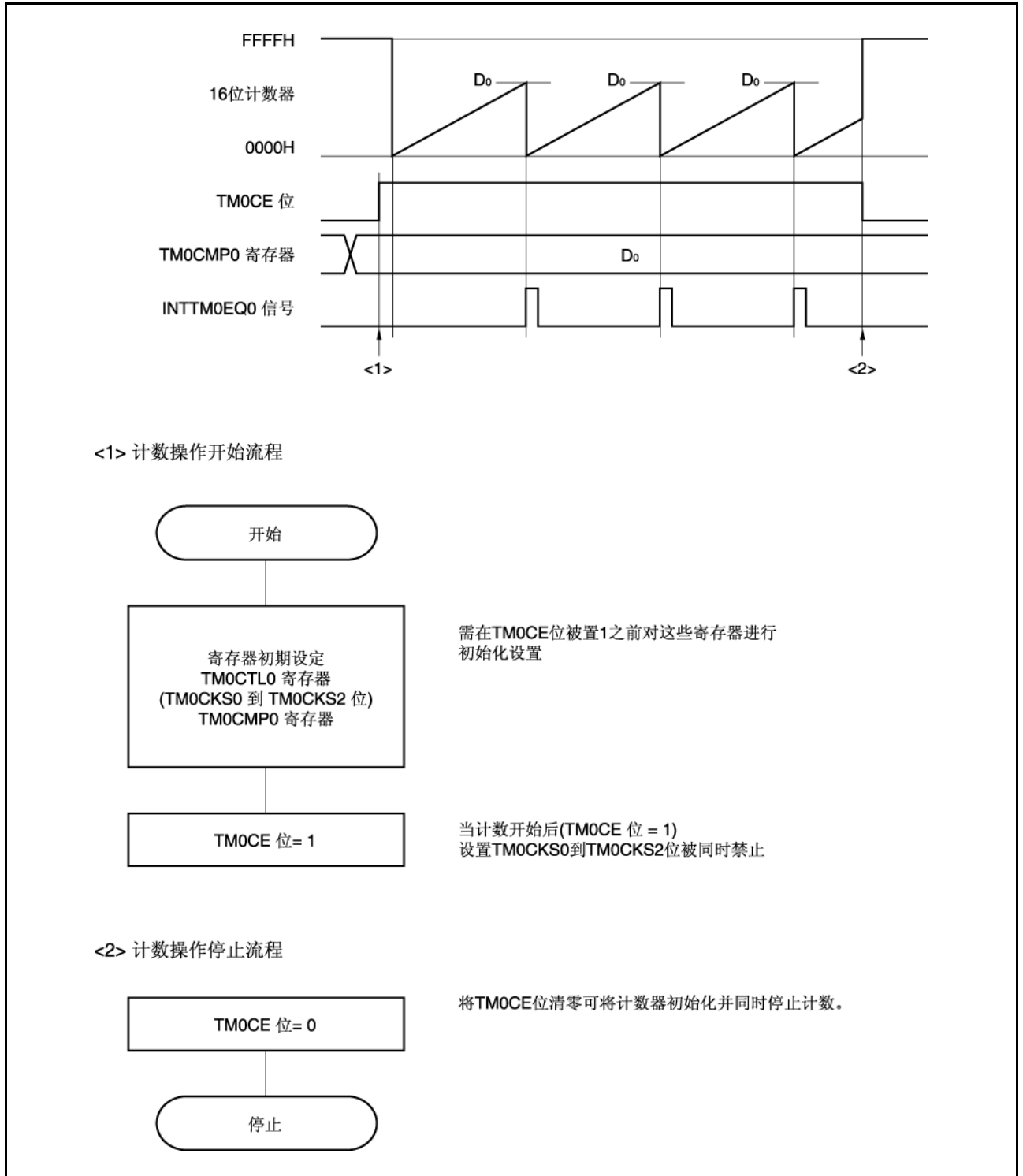
$$\text{间隔时间} = (\text{TM0CMP0 寄存器的设置值} + 1) \times \text{计数时钟周期}$$

图 9-4. 间隔定时器模式寄存器设置



(1) 间隔定时器模式的操作流程

图 9-5. 间隔定时器模式下的软件处理流程



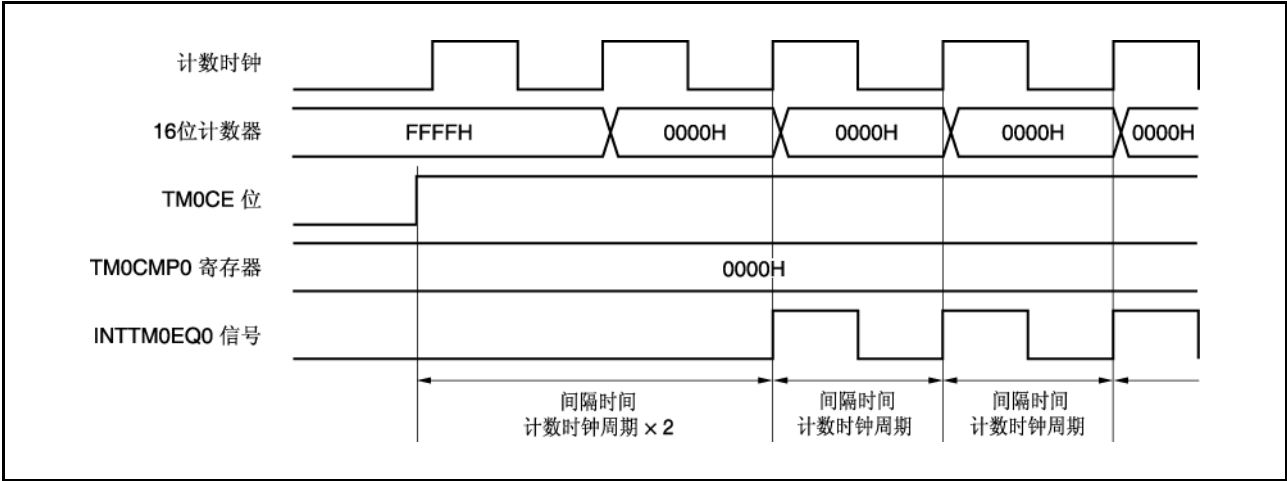
(2) 间隔定时器模式操作时序

注意事项 不可以将 TM0CMP0 寄存器设置为 FFFFH。

<R>

(a) TM0CMP0 寄存器被设置为 0000H 时的操作

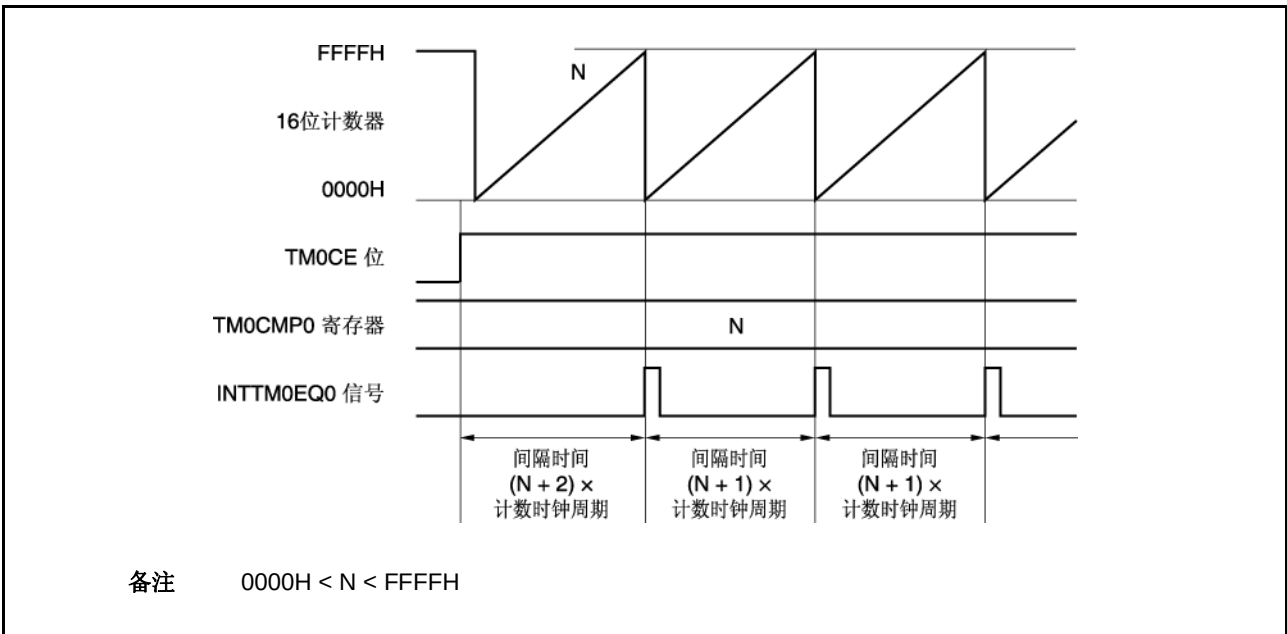
如果 TM0CMP0 寄存器被设置为 0000H，那么每个计数时钟到来时都会产生 INTTM0EQ0 信号。
16 位计数器的值始终保持为 0000H。



(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。

<R>



9.4.2 注意事项

(1) 根据选择的计数时钟的不同，从 TM0CTL0.TM0CE 位被置 1 到 16 位计数器开始计数的时间间隔如下所示。

所选计数时钟	开始计数前的最大等待时间
f_{xx}	$2/f_{xx}$
$f_{xx}/2$	$6/f_{xx}$
$f_{xx}/4$	$24/f_{xx}$
$f_{xx}/64$	$128/f_{xx}$
$f_{xx}/512$	$1024/f_{xx}$
INTWT	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。

<R>

(3) 不可以将 TM0CMP0 寄存器设置为 FFFFH。

第 10 章 钟表定时器功能

10.1 功能

钟表定时器具有以下功能。

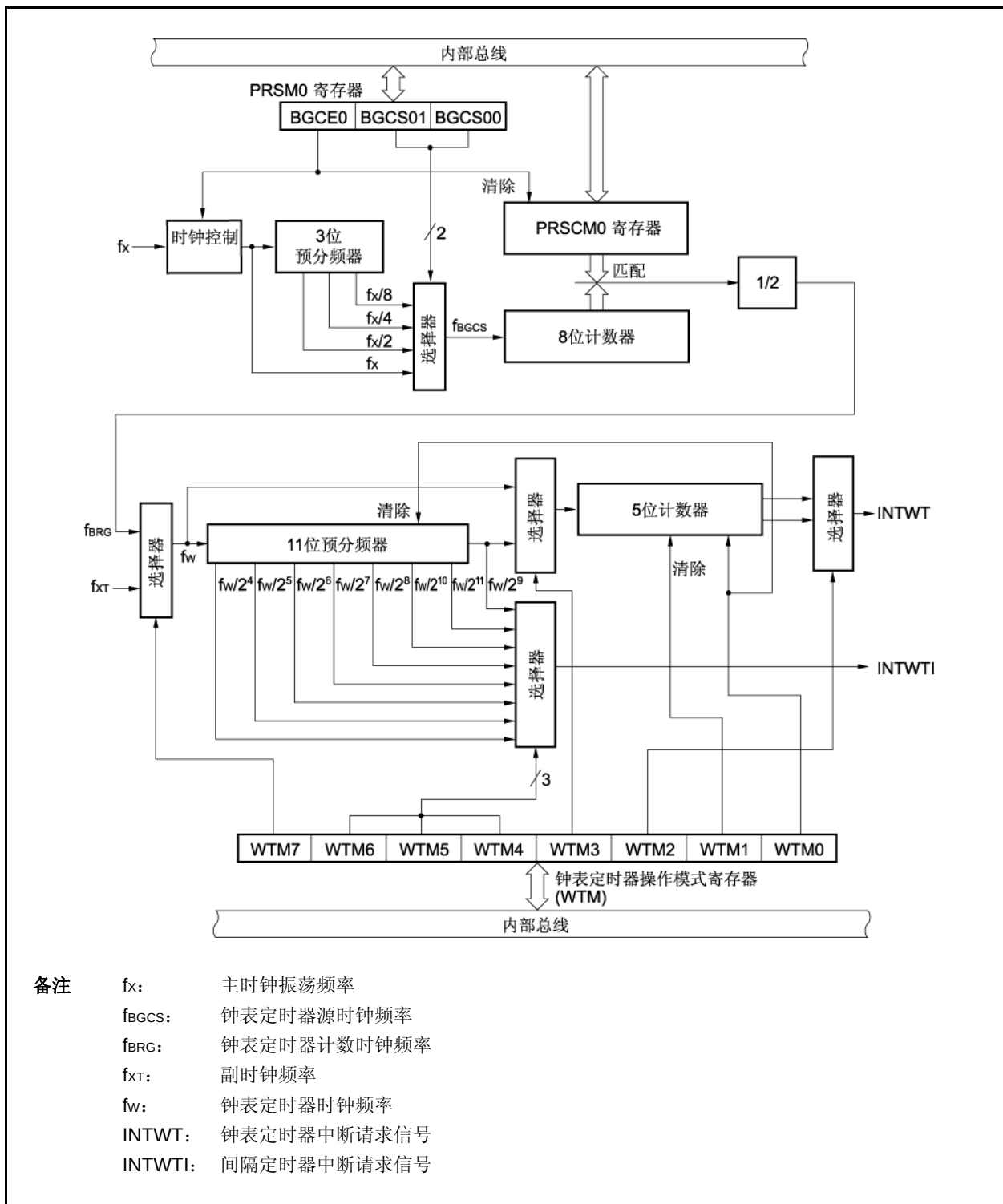
- 钟表定时器：当使用主时钟或副时钟时，以 0.5 或 0.25 秒的间隔产生中断请求信号（INTWT）。
- 间隔定时器：以设定的间隔产生中断请求信号（INTWTI）。

钟表定时器和间隔定时器可以同时使用。

10.2 配置

钟表定时器的结构框图如下。

图 10-1. 钟表定时器的结构框图



(1) 时钟控制

当钟表定时器工作于主时钟时，该模块用于控制操作时钟（ f_x ）的供给和停止。

(2) 3 位预分频器

f_x 被预分频为 $f_x/2$ 、 $f_x/4$ 或 $f_x/8$ 。

(3) 8 位计数器

该 8 位计数器以源时钟（ f_{BGCS} ）为计数时钟进行计数。

(4) 11 位预分频器

f_w 被预分频为 $f_w/2^4 \sim f_w/2^{11}$ 。

(5) 5 位计数器

该计数器以 f_w 或 $f_w/2^9$ 为计数时钟进行计数，以间隔 $2^4/f_w$ 、 $2^5/f_w$ 、 $2^{12}/f_w$ 或 $2^{14}/f_w$ 产生一个钟表定时器中断请求信号。

(6) 选择器

钟表定时器有以下 5 种选择器。

- 选择 f_x 、 $f_x/2$ 、 $f_x/4$ 或 $f_x/8$ 之一作为钟表定时器源时钟
- 选择主时钟（ f_x ）或副时钟（ f_{XT} ）作为钟表定时器时钟
- 选择 f_w 或 $f_w/2^9$ 作为 5 位计数器的计数时钟频率
- 选择 $2^4/f_w$ 、 $2^{13}/f_w$ 、 $2^5/f_w$ ，或 $2^{14}/f_w$ 作为信号 INTWT 所产生时间间隔
- 选择 $2^4/f_w$ 或 $2^{11}/f_w$ 作为间隔定时器中断请求信号（INTWTI）的产生时间间隔

(7) PRSCM 寄存器

该寄存器是 8 位比较寄存器，用于设置间隔时间。

(8) PRSM 寄存器

该寄存器控制提供给钟表定时器的时钟。

(9) WTM 寄存器

该寄存器是 8 位比较寄存器，用于控制钟表定时器/间隔定时器的操作，并设置中断请求信号产生间隔。

10.3 控制寄存器

钟表定时器使用下述寄存器。

- 预分频模式寄存器 0（PRSM0）
- 预分频比较寄存器 0（PRSCM0）
- 钟表定时器操作模式寄存器（WTM）

(1) 预分频模式寄存器 0（PRSM0）

该寄存器控制钟表定时器计数时钟的产生。
可对该寄存器进行 1 位或 8 位的读写操作。
复位信号将该寄存器设置为 00H。

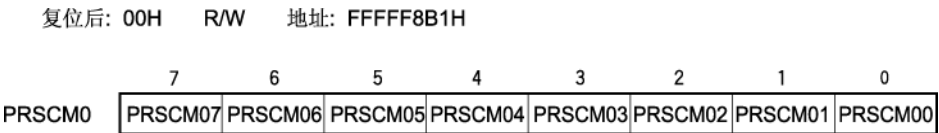
复位后: 00H R/W 地址: FFFFF8B0H

	7	6	5	<4>	3	2	1	0
PRSM0	0	0	0	BGCE0	0	0	BGCS01	BGCS00
BGCE0		主时钟操作使能						
0		禁止						
1		使能						
BGCS01	BGCS00	钟表定时器源时钟(fBGCS)的选择						
			5 MHz		4 MHz			
0	0	fx	200 ns		250 ns			
0	1	fx/2	400 ns		500 ns			
1	0	fx/4	800 ns		1 μs			
1	1	fx/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 fBRG 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 之前，先设置寄存器 PRCM0。
 - 3. 根据主时钟频率，设置寄存器 PRSM0 和 PRSCM0 以获得 32.768 kHz 的 f_{BRG} 频率。

f_{BRG} 的计算如下。

$$f_{BRG} = f_{BGCS}/2N$$

- 备注
- f_{BGCS}: 通过寄存器 PRSM0 设定的钟表定时器源时钟频率
 - N: 寄存器 PRSM0 的设置值= 1 ~ 256
但是，当 PRSM0 的值=00H 时，N = 256。

(3) 钟表定时器操作模式寄存器 (WTM)

该寄存器用于使能 / 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，设置钟表标志的置位时间。

在设置寄存器 WTM 前，先设置寄存器 PRSM0。

可对该寄存器进行 1 位或 8 位的读写操作。

复位信号将该寄存器设置为 00H。

(1/2)

复位后: 00H R/W 地址: FFFF680H

	7	6	5	4	3	2	<1>	<0>
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	WTM6	WTM5	WTM4	预分频器间隔时间的选择
0	0	0	0	$2^4/f_w$ (488 μ s: $f_w = f_{XT}$)
0	0	0	1	$2^5/f_w$ (977 μ s: $f_w = f_{XT}$)
0	0	1	0	$2^6/f_w$ (1.95 ms: $f_w = f_{XT}$)
0	0	1	1	$2^7/f_w$ (3.91 ms: $f_w = f_{XT}$)
0	1	0	0	$2^8/f_w$ (7.81 ms: $f_w = f_{XT}$)
0	1	0	1	$2^9/f_w$ (15.6 ms: $f_w = f_{XT}$)
0	1	1	0	$2^{10}/f_w$ (31.3 ms: $f_w = f_{XT}$)
0	1	1	1	$2^{11}/f_w$ (62.5 ms: $f_w = f_{XT}$)
1	0	0	0	$2^4/f_w$ (488 μ s: $f_w = f_{BRG}$)
1	0	0	1	$2^5/f_w$ (977 μ s: $f_w = f_{BRG}$)
1	0	1	0	$2^6/f_w$ (1.95 ms: $f_w = f_{BRG}$)
1	0	1	1	$2^7/f_w$ (3.91 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	$2^{10}/f_w$ (31.3 ms: $f_w = f_{BRG}$)
1	1	1	1	$2^{11}/f_w$ (62.5 ms: $f_w = f_{BRG}$)

WTM7	WTM3	WTM2	钟表标志的设置时间的选择
0	0	0	$2^{14}/f_w$ (0.5 s: $f_w = f_{XT}$)
0	0	1	$2^{13}/f_w$ (0.25 s: $f_w = f_{XT}$)
0	1	0	$2^5/f_w$ (977 μ s: $f_w = f_{XT}$)
0	1	1	$2^4/f_w$ (488 μ s: $f_w = f_{XT}$)
1	0	0	$2^{14}/f_w$ (0.5 s: $f_w = f_{BRG}$)
1	0	1	$2^{13}/f_w$ (0.25 s: $f_w = f_{BRG}$)
1	1	0	$2^5/f_w$ (977 μ s: $f_w = f_{BRG}$)
1	1	1	$2^4/f_w$ (488 μ s: $f_w = f_{BRG}$)

WTM1	5位计数器操作的控制
0	操作停止后清除
1	开始

WTM0	钟表定时器操作使能
0	停止操作(清除预分频器和5位计数器)
1	使能操作

注意事项 当 WTM0 和 WTM1 均为 0 时才可对 WTM2 ~ WTM7 位进行写入操作。

备注

1. f_w : 钟表定时器时钟频率
2. 括号中的值适用于 $f_w = 32.768$ kHz

10.4 操作

10.4.1 作为钟表定时器的操作

钟表定时器以固定时间间隔产生中断请求信号（INTWT）。钟表定时器使用主时钟或副时钟（32.768 kHz）的 0.5 或 0.25 秒的时间间隔。

当 WTM.WTM1 和 WTM.WTM0 位都置 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 位设置计数时钟，使用寄存器 PRSCM0 设置 8 位比较值，并设置钟表定时器的计数时钟频率（ f_{BRG} ）为 32.768 kHz。

当 PRSM0.BGCE0 位置（1）， f_{BRG} 被用于钟表定时器。

可以使用下面的表达式计算 f_{BRG} 。

$$f_{BRG} = f_x / (2^{m+1} \times N)$$

要设置 f_{BRG} 为 32.768 kHz，进行下面计算，设置 BGCS01 和 BGCS00 位和寄存器 PRSCM0。

- <1> 设置 $N = f_x / 65,536$ ，设置 $m = 0$ 。
- <2> 当 N 的个位的舍入结果为偶数时，在进行舍入之前，请设置 $N = N/2$ ， $m = m + 1$ 。
- <3> 重复步骤 <2> 直到 N 成为奇数或 $m = 3$ 。
- <4> 将舍入后 N 的值（整数部分）设置到 PRSCM0 寄存器，将 m 设置到 BGCS01 和 BGCS00 位中。

示例： 当 $f_x = 4.00$ MHz 时

<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：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 ~ 3

N ：寄存器 PRSCM0 的设置值= 1 ~ 256

但是，当 PRSCM0 的值=00H 时， $N = 256$

f_x ：主时钟振荡频率

10.4.2 作为间隔定时器的操作

钟表定时器可以用作间隔定时器，并按照预设的数值周期性产生中断请求信号（INTWTI）。

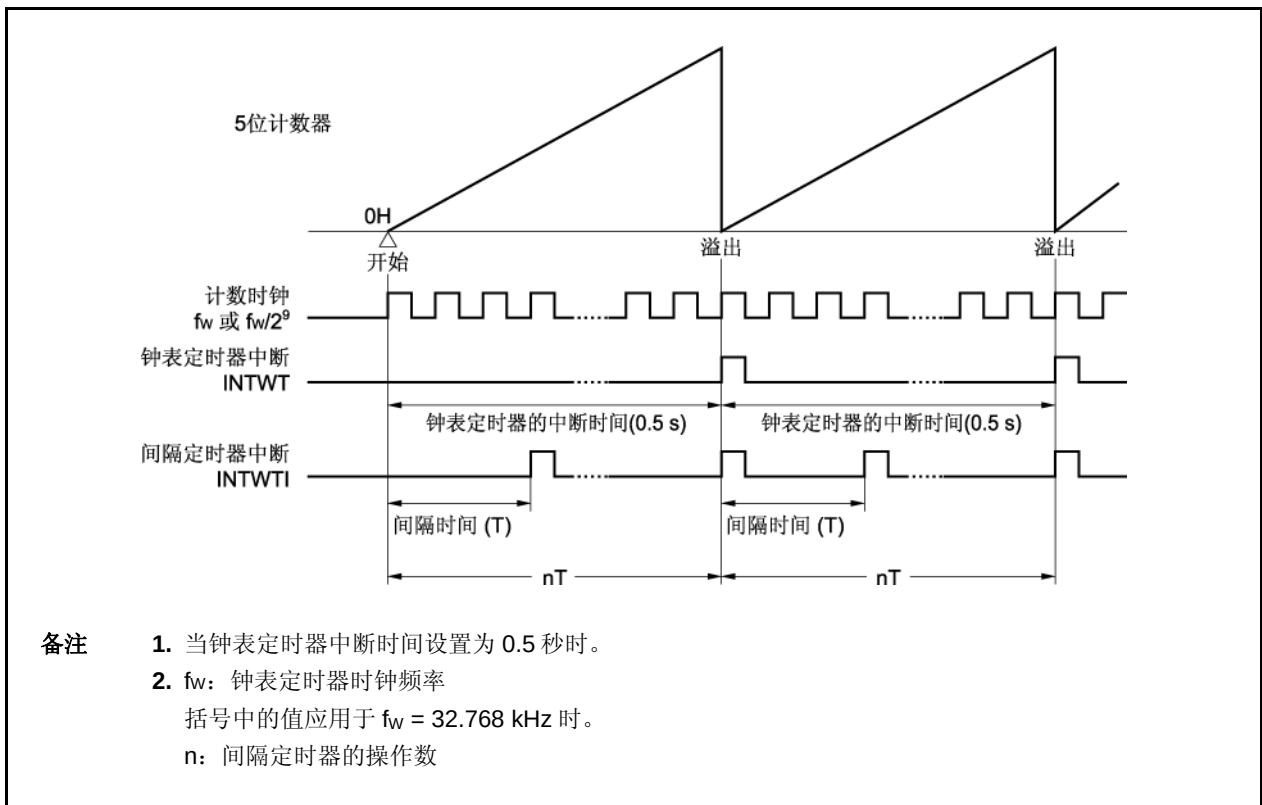
可由寄存器 WTM 的 WTM4 ~ WTM7 位选择间隔时间。

表 10-1. 间隔定时器的间隔时间

WTM7	WTM6	WTM5	WTM4	间隔时间	
0	0	0	0	$2^4 \times 1/f_w$	488 μ s （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	0	0	1	$2^5 \times 1/f_w$	977 μ s （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	0	1	0	$2^6 \times 1/f_w$	1.95 ms （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	0	1	1	$2^7 \times 1/f_w$	3.91 ms （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	1	0	0	$2^8 \times 1/f_w$	7.81 ms （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	1	0	1	$2^9 \times 1/f_w$	15.6 ms （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	1	1	0	$2^{10} \times 1/f_w$	31.3 ms （工作于 $f_w = f_{XT} = 32.768$ kHz）
0	1	1	1	$2^{11} \times 1/f_w$	62.5 ms （工作于 $f_w = f_{XT} = 32.768$ kHz）
1	0	0	0	$2^4 \times 1/f_w$	488 μ s （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	0	0	1	$2^5 \times 1/f_w$	977 μ s （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	0	1	0	$2^6 \times 1/f_w$	1.95 ms （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	0	1	1	$2^7 \times 1/f_w$	3.91 ms （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	1	0	0	$2^8 \times 1/f_w$	7.81 ms （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	1	0	1	$2^9 \times 1/f_w$	15.6 ms （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	1	1	0	$2^{10} \times 1/f_w$	31.3 ms （工作于 $f_w = f_{BRG} = 32.768$ kHz）
1	1	1	1	$2^{11} \times 1/f_w$	62.5 ms （工作于 $f_w = f_{BRG} = 32.768$ kHz）

备注 f_w : 钟表定时器时钟频率

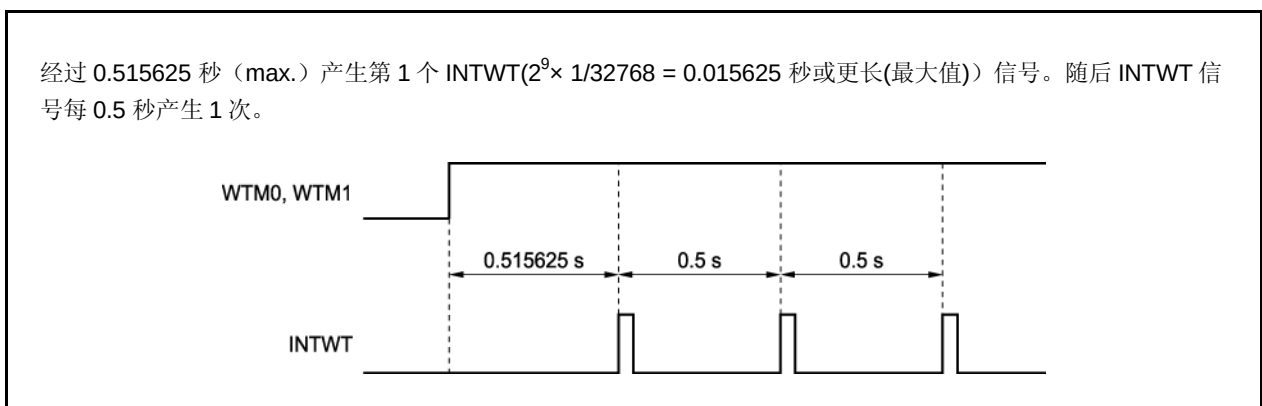
图 10-2. 钟表定时器 / 间隔定时器的操作时序



10.4.3 注意事项

在使能操作（WTM.WTM1 和 WTM.WTM0 = 1）后，第一次钟表定时器中断请求信号（INTWT）产生前需要一段准备时间。

图 10-3. 钟表定时器产生中断请求信号（INTWT）的示例（当中断周期= 0.5 s）



第 11 章 看门狗定时器 2 的功能

11.1 功能

看门狗定时器 2 的功能如下。

- 默认启动看门狗定时器^{注1}
 - 复位模式：看门狗定时器 2 溢出时进行复位操作（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}
- 输入源时钟可从主时钟、内部振荡时钟和副时钟中选择

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 时，在复位前使其停止工作，或清除看门狗定时器 2 并在下一次间隔时间内使其停止工作。

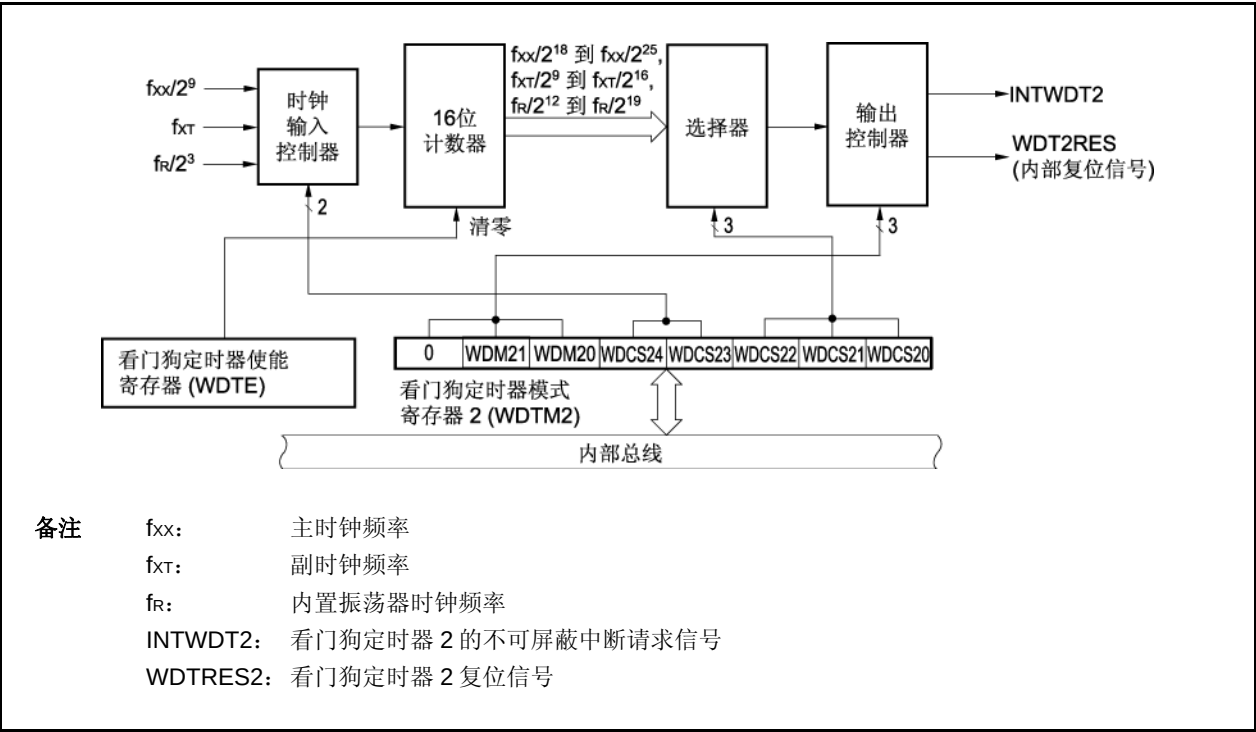
为了确认操作的正确性，即使在不需要改变默认设置（复位模式，间隔时间： $fR/2^{19}$ ）的情况下，也要对寄存器 WDTM2 执行一次写操作。

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 22.2.2 （2）信号 INTWDT2。

11.2 配置

看门狗定时器 2 的框图如下。

图 11-1. 看门狗定时器 2 的框图



看门狗定时器 2 由以下寄存器控制。

表 11-1. 看门狗定时器 2 的配置

项目	配置
控制寄存器	看门狗定时器模式寄存器 2 (WDTM2) 看门狗定时器使能寄存器 (WDTE)

11.3 寄存器

(1) 看门狗定时器模式寄存器 2 (WDTM2)

该寄存器可以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 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。

- 当主时钟振荡停止，CPU 使用副时钟操作时。
- 当 CPU 使用内置振荡时钟进行操作时

复位后: 67H R/W 地址: FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	WDCS24	WDCS23	WDCS22	WDCS21	WDCS20

WDM21	WDM20	看门狗定时器2的操作模式的选择
0	0	停止操作
0	1	不可屏蔽中断请求模式 (INTWDT2 信号的产生)
1	—	复位模式 (WDT2RES 信号的产生)

- 注意事项**
1. WDCS20 到 WDCS24 的详细情况，参见表 11-2 看门狗定时器 2 的时钟选择。
 2. 尽管停止内置振荡器就可以停止看门狗定时器 2，但是，还要对 WDTM2 寄存器清零以确保停止定时器（以避免由于错误的写入操作选中主时钟或副时钟）。
 3. 如果寄存器 WDTM2 在复位后被写入两次，那么产生溢出信号，计数器复位。
 4. 要产生一个溢出信号，对 WDTM2 寄存器写入两次或写入一个“ACH”之外的值到 WDTE 寄存器即可。
然而，当看门狗定时器 2 被设置成停止操作时，即使写入 WDTM2 寄存器两次或写入一个“ACH”之外的值到 WDTE 寄存器也不会产生溢出信号。
 5. 要停止看门狗定时器 2 的操作，设置 RCM.RSTP 为 1（以停止内置振荡器）并向寄存器 WDTM2 写入 00H。如果 RCM.RSTP 不能被置 1，设置 WDCS23 为 1（选中 $2^n/f_{xx}$ ，时钟在 IDLE1，IDLW2，sub-IDLE 和副时钟模式下可停止工作）。

<R>

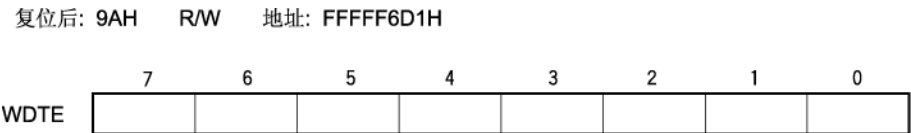
表 11-2. 看门狗定时器 2 的时钟选择

WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	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	$2^{17}/f_R$	1310.7 ms	655.4 ms	327.7 ms	
0	0	1	1	0	$2^{18}/f_R$	2621.4 ms	1310.7 ms	655.4 ms	
0	0	1	1	1	$2^{19}/f_R$	5242.9 ms	2621.47 ms	1310.7 ms	
						$f_{XX} = 32 \text{ MHz}^{\text{註}}$	$f_{XX} = 20 \text{ MHz}$	$f_{XX} = 16 \text{ MHz}$	$f_{XX} = 10 \text{ MHz}$
0	1	0	0	0	$2^{18}/f_{XX}$	8.2 ms	13.1 ms	16.4 ms	26.2 ms
0	1	0	0	1	$2^{19}/f_{XX}$	16.4 ms	26.2 ms	32.8 ms	52.4 ms
0	1	0	1	0	$2^{20}/f_{XX}$	32.8 ms	52.4 ms	65.5 ms	104.9 ms
0	1	0	1	1	$2^{21}/f_{XX}$	65.5 ms	104.9 ms	131.1 ms	209.7 ms
0	1	1	0	0	$2^{22}/f_{XX}$	131.1 ms	209.7 ms	262.1 ms	419.4 ms
0	1	1	0	1	$2^{23}/f_{XX}$	262.1 ms	419.4 ms	524.3 ms	838.9 ms
0	1	1	1	0	$2^{24}/f_{XX}$	524.3 ms	838.9 ms	1048.6 ms	1677.7 ms
0	1	1	1	1	$2^{25}/f_{XX}$	1048.6 ms	1677.7 ms	2097.2 ms	3355.4 ms
						$f_{XT} = 32.768 \text{ kHz}$			
1	×	0	0	0	$2^9/f_{XT}$	15.625 ms			
1	×	0	0	1	$2^{10}/f_{XT}$	31.25 ms			
1	×	0	1	0	$2^{11}/f_{XT}$	62.5 ms			
1	×	0	1	1	$2^{12}/f_{XT}$	125 ms			
1	×	1	0	0	$2^{13}/f_{XT}$	250 ms			
1	×	1	0	1	$2^{14}/f_{XT}$	500 ms			
1	×	1	1	0	$2^{15}/f_{XT}$	1000 ms			
1	×	1	1	1	$2^{16}/f_{XT}$	2000 ms			

注 只适用于 V850ES/SG2-H

(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。



- 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. 要产生一个溢出信号，对 WDTM2 寄存器写入两次或写入一个“ACH”之外的值到 WDTE 寄存器即可。
然而，当看门狗定时器 2 被设置成停止操作时，即使写入 WDTM2 寄存器两次或写入一个“ACH”之外的值到 WDTE 寄存器也不会产生溢出信号。
 - 4. 读取寄存器 WDTE 的值为“9AH”（与写入值“ACH”不同）。

11.4 操作

复位释放后，看门狗定时器 2 自动开始运行。

复位释放后 WDTM2 寄存器只可被写入一次，且必须使用字节访问操作。要使用看门狗定时器 2，使用 8 位操作指令，向 WDTM2 寄存器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 到 WDCS20 位被用于选择看门狗定时器 2 的循环检测时间间隔。

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。

如果在定时间隔时间内没有对寄存器 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的振荡稳定过程中产生 WDT 溢出，将不会产生内部复位，且 CPU 时钟将转换成内部振荡时钟。

如果不使用看门狗定时器 2，向寄存器 WDTM2 写入 00H。

不可屏蔽中断请求模式下的不可屏蔽中断请求服务可参见 22.2.2 (2) 信号 INTWDT2。

第 12 章 实时输出功能 (RTO)

12.1 功能

实时输出功能将预先设置的数据发送到寄存器 RTBL0 和 RTBH0，然后在定时器中断时由硬件通过输出锁存发送这些数据到外部设备。发送这些数据到外部设备的引脚组成了实时输出功能(RTO)端口。

由于 RTO 可以稳定的输出信号，所以适用于控制步进电机。

<R>

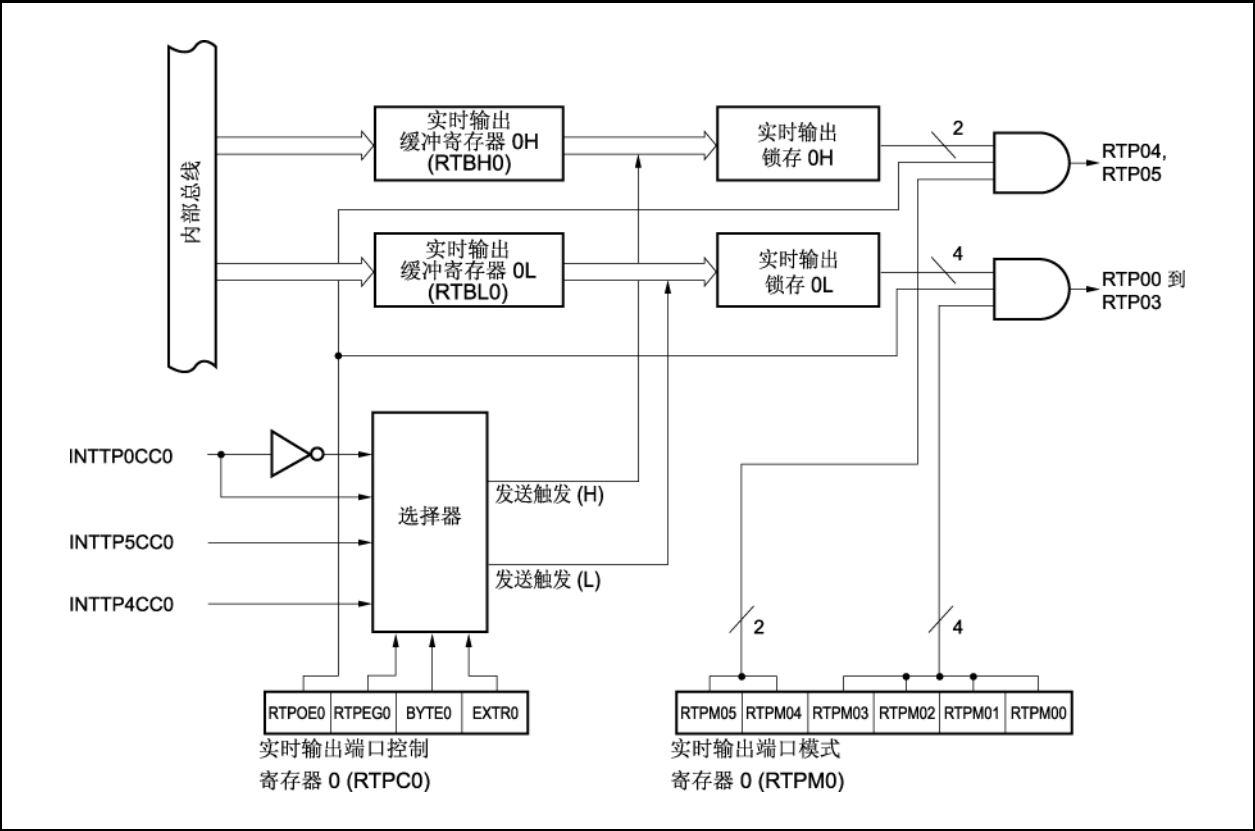
V850ES/SG2 和 V850ES/SG2-H 提供一个 6 位的实时输出端口通道。

可由 1 位存储器操作指令设置实时输出端口工作在普通端口模式或实时输出端口模式。

12.2 配置

RTO 的框图如下

图 12-1. RTO 的框图



RTO 由以下硬件组成。

表 12-1. RTO 的配置

项目	配置
寄存器	实时输出缓冲寄存器 0L, 0H (RTBL0, RTBH0)
控制寄存器	实时输出端口模式寄存器 0 (RTPM0) 实时输出端口模式寄存器 0 (RTPC0)

(1) 实时输出缓冲寄存器 0L, 0H (RTBL0, RTBH0)

寄存器 RTBL0 和 RTBH0 是 4 位寄存器，用于存放预先设置的输出数据。

该寄存器被映射到周边 I/O 寄存器区域的独立地址。

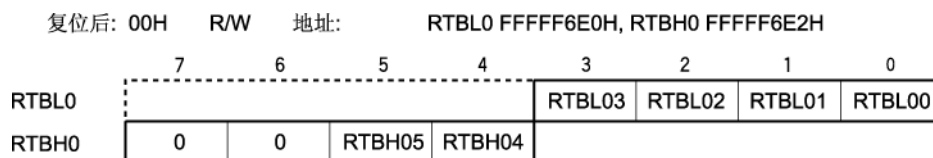
可由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号产生将该寄存器设置为 00H。

如果指定操作模式为 4 位 × 1 通道或 2 位 × 1 通道(RTPC0.BYTE0 位 = 0)，数据可被分别设置到寄存器 RTBL0 和 RTBH0。通过指定这两个寄存器中的任意一个地址，一次读取就可以获得这两个寄存器中的数据。

如果指定了操作模式为 6 位 × 1 通道(BYTE0 = 1)，对这两个寄存器中的任意一个写入，8 位数据可被设置到寄存器 RTBL0 和 RTBH0 中。另外，通过指定这两个寄存器中的任意一个的地址，一次读取就可以获得这两个寄存器中的数据。

表 12-2 表示操作寄存器 RTBL0 和 RTBH0 的过程。



- 注意事项**
1. 当向寄存器 RTBH0 的第 6、7 位进行写操作时，写入的数据恒为 0。
 2. 下述状态下禁止访问寄存器 RTBL0 和 RTBH0。详细情况，参见 3.4.9 (2)访问特定的片上周边 I/O 寄存器。
 - 当 CPU 使用副时钟进行操作，主时钟晶振停止振荡时。
 - 当 CPU 使用内置振荡时钟进行操作时

表 12-2. 寄存器 RTBL0 和 RTBH0 修改过程中的操作

操作模式	被修改的寄存器	读		写 [※]	
		高 4 位	低 4 位	高 4 位	低 4 位
4 位 × 1 通道, 2 位 × 1 通道	RTBL0	RTBH0	RTBL0	无效	RTBL0
	RTBH0	RTBH0	RTBL0	RTBH0	无效
6 位 × 1 通道	RTBL0	RTBH0	RTBL0	RTBH0	RTBL0
	RTBH0	RTBH0	RTBL0	RTBH0	RTBL0

注 设置实时输出端口后，在实时输出触发时，将欲输出的数据设置到寄存器 RTBL0 和 RTBH0。

12.3 寄存器

RTO 由下面两个寄存器控制。

- 实时输出端口模式寄存器 0 (RTPM0)
- 实时输出端口控制寄存器 0 (RTPC0)

(1) 实时输出端口模式寄存器 0 (RTPM0)

该寄存器选择实时输出端口模式或普通端口模式，可由 1 位存储器操作指令完成操作。
可由 1 位或 8 位存储器操作指令设置该寄存器。
复位信号产生将该寄存器设置为 00H H。

复位后: 00H R/W 地址: FFFFF6E4H

	7	6	5	4	3	2	1	0
RTPM0	0	0	RTPM05	RTPM04	RTPM03	RTPM02	RTPM01	RTPM00

RTPM0m	实时输出端口的控制 (m = 0 到5)
0	禁止实时输出
1	允许实时输出

- 注意事项
1. 通过使能实时输出操作(RTPC0.RTPOE0 位 = 1)，RTP00 和 RTP05 中设置为实时输出的位将执行实时输出操作，而那些设置为端口模式的位会输出 0。
 2. 禁止实时输出(RTPOE0 位 = 0)时，无论 RTPM0 寄存器如何设置，实时输出引脚(RTP00 到 RTP05)均输出 0。
 3. 为了将(RTP00 到 RTP05)引脚作为实时输出引脚使用，需要通过 PMC 和 PFC 寄存器将这些引脚的模式设置为实时输出端口模式。

(2) 实时输出端口控制寄存器 0 (RTPC0)

该寄存器用于设置实时输出端口的操作模式和触发模式。
实时输出端口的操作模式和触发模式的关系如表 12-3 所示。
可由 1 位或 8 位存储器操作指令设置该寄存器。
复位信号产生将该寄存器设置为 00H。

复位后: 00H R/W 地址: FFFFF6E5H

RTPC0

<7>	6	5	4	3	2	1	0
RTPOE0	RTPEG0	BYTE0	EXTR0	0	0	0	0

RTPOE0	实时输出操作的选择
0	禁止操作 ^{注 1}
1	允许操作

RTPEG0	INTTP0CC0 信号的有效沿
0	下降沿 ^{注 2}
1	上升沿

BYTE0	实时输出的通道配置指定
0	4 位 × 2 通道, 2 位 × 2 通道
1	6 位 × 2 通道

注

1.

禁止实时输出操作(RTPOE0 位 = 0)时, 实时输出信号的所有位(RTP00 到 RTP05)输出“0”。

2.

输出 INTTP0CC0 信号持续一个时钟周期, 该时钟周期由 TMP0 选择。

注意事项

只有当 RTPOE0 位= 0 时, 设置 RTPEG0, BYTE0 和 EXTR0 位。

表 12-3. 实时输出端口的操作模式和输出触发

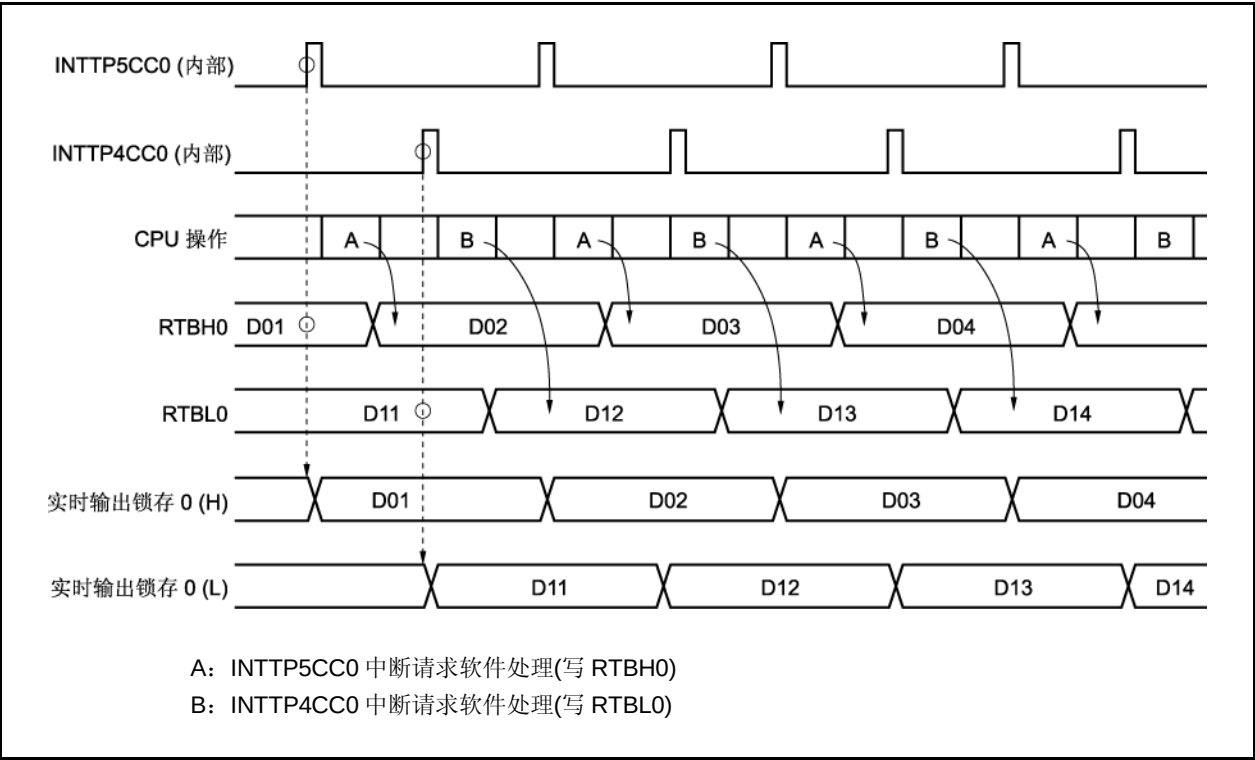
BYTE0	EXTR0	操作模式	RTBH0 (RTP04, RTP05)	RTBL0 (RTP00 到 RTP03)
0	0	4 位 × 1 通道,	INTTP5CC0	INTTP4CC0
	1	2 位 × 1 通道	INTTP4CC0	INTTP0CC0
1	0	6 位 × 1 通道	INTTP4CC0	
	1		INTTP0CC0	

12.4 操作

如果设置 RTPC0.RTPOE0 为 1，就会使能实时输出操作，寄存器 RTBH0 和 RTBL0 的数据将随着选择的发送触发信号(由 RTPC0.EXTR0 和 RTPC0.BYTE0 设置)被同步发送到实时输出锁存器。对于被发送的数据，只有那些由寄存器 RTPM0 设定允许实时输出的位才从 RTP00 到 RTP05 位输出。被寄存器 RTPM0 设定禁止实时输出的位输出为 0。

如果将 RTPOE0 清零来禁止实时输出操作，那么无论寄存器 RTPM0 如何设置，信号 RTP00 到 RTP05 的输出都为 0。

图 12-2. RTO0 操作时序的示例 (当 EXTR0 = 0, BYTE0 = 0)



备注 关于待机模式下的操作，参见第 24 章 待机功能。

12.5 使用方法

- (1) 禁止实时输出。
将 RTPC0.RTPOE0 位清 0。
- (2) 按照下面步骤进行初始化。
 - 设置复用功能引脚端口 5
设置 PFC5.PFC5m 和 PFCE5.PFCE5m 为 1，然后设置 PMC5.PMC5m 为 1(m = 0 到 5)。
 - 可由 1 位存储器操作指令设置实时输出端口模式或普通端口模式。
设置 RTPM0 寄存器。
 - 通道配置：选择触发器和有效沿。
设置 RTPC0.EXTR0, RTPC0.BYTE0 和 RTPC0.RTPEG0 位。
 - 设置寄存器 RTBH0 和 RTBL0^{注1}的初始值。
- (3) 使能实时输出。
设置 RTPOE0 位 = 1。
- (4) 在选中的发送触发^{注2}信号产生时，将下一组输出值设置到寄存器 RTBH0 和 RTBL0。
- (5) 在选中的发送触发相应的中断服务中，将下一组输出值设置到寄存器 RTBH0 和 RTBL0。

- 注
1. 如果当 RTPOE0 = 0 时写寄存器 RTBH0 和 RTBL0，数据被分别发送到实时输出锁存 0H 和 0L。
 2. 即使当 RTPOE0 = 1 时写寄存器 RTBH0 和 RTBL0，数据也不被发送到实时输出锁存 0H 和 0L。

12.6 注意事项

- (1) 通过软件避免下述冲突。
 - 实时输出使能/禁止转换(RTPOE0 位)和选中的实时输出触发之间的冲突。
 - 实时输出使能状态下写寄存器 RTBH0 和 RTBL0，与选中的实时输出触发之间的冲突。
- (2) 在操作初始化前，停止实时输出(RTPOE0 = 0)。
- (3) 一旦实时输出被禁止(RTPOE0 = 0)，确保在再次使能实时输出(RTPOE0 = 0 → 1)前初始化寄存器 RTBH0 和 RTBL0。

第 13 章 A/D 转换器

13.1 概述

A/D 转换器将模拟输入信号转换为数字信号，具有 10 位分辨率，最多可由 12 个通道（ANI0 ~ ANI11）组成。
A/D 转换器有以下特点。

- 10 位分辨率
- 12 通道
- 逐次逼近法
- 操作电压：AVREF0 = 3.0 到 3.6 V
- 模拟输入电压：0 V 到 AVREF0
- 以下为所能提供的操作模式。
 - 连续选择模式
 - 连续扫描模式
 - 单脉冲选择模式
 - 单脉冲扫描模式
- 以下功能为所能提供的触发模式。
 - 软件触发模式
 - 外部触发模式（外部，1）
 - 定时器触发模式
- 电压不足监测功能（转换结果比较功能）

13.2 功能

（1）10 位分辨率 A/D 转换

从 ANI0 到 ANI11 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 转换结束时，都产生一个中断请求信号（INTAD）。

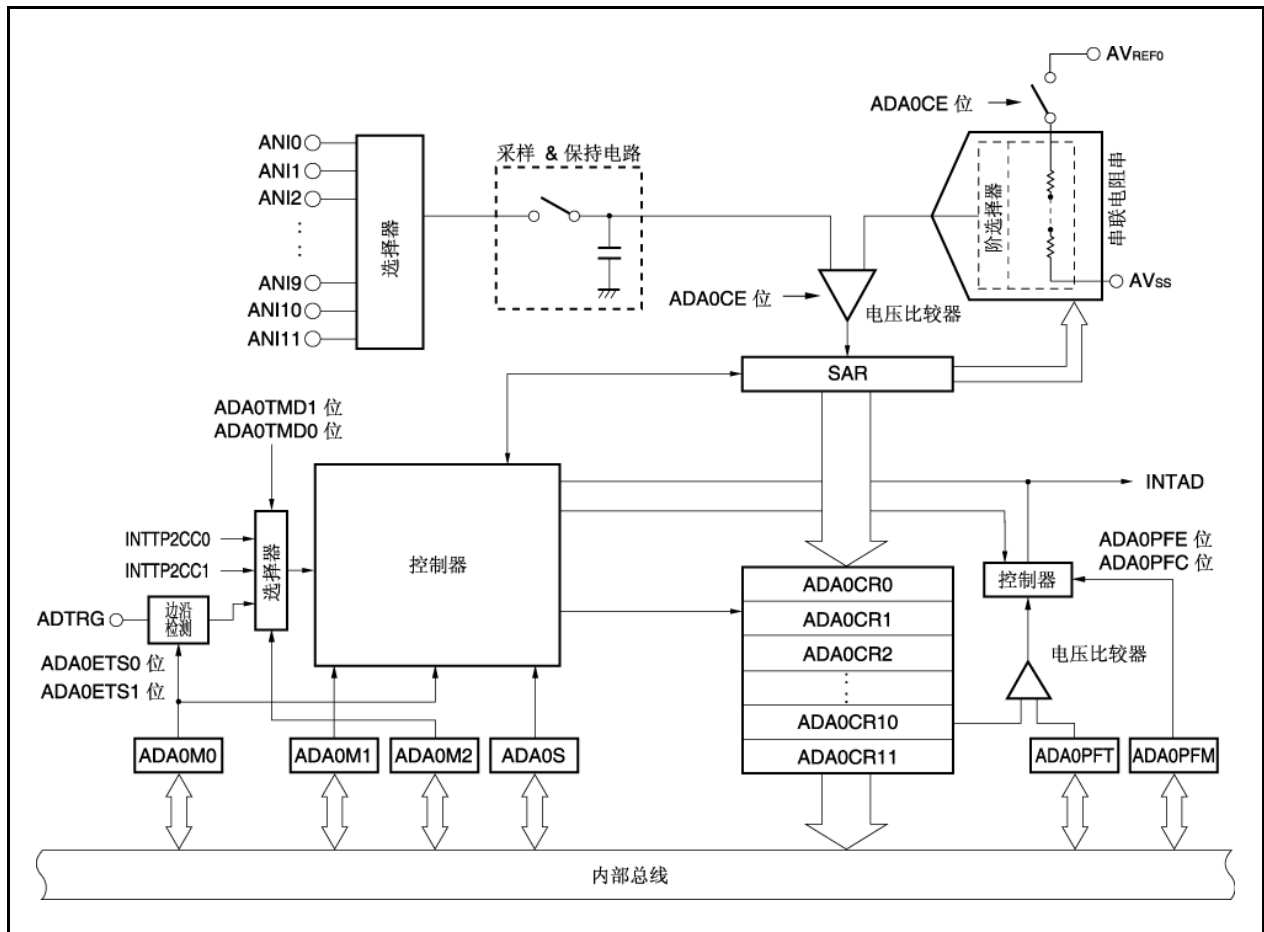
（2）电压不足检测功能

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 ~ 11）满足时，才产生中断信号 INTAD。

13.3 配置

如下所示为 A/D 转换器的框图。

图 13-1. 转换器的框图



A/D 转换器包括以下硬件。

13-1. A/D 转换器的配置

项目	配置
模拟输入	12 通道（ANI0 ~ ANI11 引脚）
寄存器	逐次逼近寄存器（SAR） A/D 转换结果寄存器 0 ~ 11（ADA0CR0 ~ ADA0CR11） A/D 转换结果寄存器 0H ~ 11H（ADCR0H ~ ADCR11H）：只能读取高 8 位
控制寄存器	A/D 转换器模式寄存器 0 ~ 2（ADA0M0 ~ ADA0M2） A/D 转换器通道选择寄存器 0（ADA0S） 电压不足比较模式寄存器（ADA0PFM） 电压不足比较阈值寄存器（ADA0PFT）

(1) 逐次逼近寄存器 (SAR)

SAR 寄存器将模拟输入信号的电压值和分级电阻串分出的分级抽头（比较电压）进行比较，并从最高有效位（MSB）开始保存比较结果。

当比较结果被置以最低有效位（LSB）（也就是说，当 A/D 转换完成），寄存器 SAR 的结果被传送到寄存器 ADA0CRn。

备注 $n = 0 \sim 11$

(2) A/D 转换结果寄存器 n (ADA0CRn)，A/D 转换结果寄存器 nH (ADA0CRnH)

寄存器 ADA0CRn 是 16 位寄存器，用于存储 A/D 转换结果。ADA0CRn 由 12 个寄存器组成。根据输入模拟信号，A/D 转换结果存储于 ADA0CRn 的高 10 位。（低 6 位恒为 0）

(3) A/D 转换器模式寄存器 0 (ADA0M0)

该寄存器设置操作模式并控制 A/D 转换器的转换操作。

(4) A/D 转换器模式寄存器 1 (ADA0M1)

该寄存器设置转换模拟输入信号所需的时间。

(5) A/D 转换器模式寄存器 2 (ADA0M2)

该寄存器用于设置硬件触发模式。

(6) 模拟输入通道选择寄存器 0 (ADA0S)

该寄存器设置输入端口，待转换的模拟电压从该输入端口输入。

(7) 电源电压不足比较模式寄存器 (ADA0PFM)

该寄存器用于设置电压不足监测模式。

(8) 电源电压不足比较限定值寄存器 (ADA0PFT)

寄存器 ADA0PFT 设置一个与 A/D 转换结果寄存器 nH (ADA0CRnH) 比较的阈值。寄存器 ADA0PFT 中所设置的 8 位数据与 A/D 转换结果寄存器 (ADA0CRnH) 比较。

(9) 控制器

当 A/D 转换完成或使用电压不足监测功能时，控制器将 A/D 转换结果 (ADA0CRnH 的值) 与 ADA0PF 的值进行比较，只有当指定的比较条件满足时，才产生中断信号 INTAD。

(10) 采样与保存电路

采样保存电路对每个模拟输入信号进行采样，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

电压比较器对采样保持的电压值和串联电阻串的电压值进行比较。

(12) 串联电阻串

这个串联电阻串连接在 AV_{REF0} 和 AV_{SS} 之间，并产生一个电压用于对模拟输入信号进行比较。

(13) $ANI0 \sim ANI11$ 引脚

它们是 12 个 A/D 转换器通道的模拟输入引脚，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作普通输入端口引脚。

注意事项 确保输入 $ANI0 \sim ANI11$ 的电压不超过额定值。特别是，如果大于或等于 AV_{REF0} 的电压输入某个通道，这个通道的转换结果将不确定，其它通道的转换结果也将受到影响。

(14) 引脚 AV_{REF0}

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，该引脚电压也应与 V_{DD} 引脚保持一致。输入到引脚 $ANI0 \sim ANI11$ 的模拟信号根据引脚 AV_{REF0} 和 AV_{SS} 之间的电压转换为数字信号。

(15) AV_{SS} 引脚

这是 A/D 转换器的信号地引脚。即便没有使用 A/D 转换器，该引脚的电压也应与 V_{SS} 的电压保持一致。

13.4 寄存器

A/D 转换器使用以下寄存器进行控制。

- A/D 转换器模式寄存器 0, 1, 2 (ADA0M0, ADA0M1, ADA0M2)
- A/D 转换器通道选择寄存器 0 (ADA0S)
- 电压不足比较模式寄存器 (ADA0PFM)

还使用下面的寄存器。

- A/D 转换结果寄存器 n (ADA0CRn)
- A/D 转换结果寄存器 nH (ADA0CRnH)
- 电源电压不足比较限定值寄存器 (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。
该寄存器可由 1 位或 8 位存储器操作指令读写。但是，ADA0EF 位是只读的。
复位使该寄存器清零。

注意事项 下述情况时禁止访问寄存器 ADA0M0。详细情况，参见 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。

- 当 CPU 使用副时钟，且主时钟停止振荡
- 当 CPU 使用内部振荡时钟

(1/2)

复位后: 00H R/W 地址: FFFFF200H

	<7>	6	5	4	3	2	1	<0>
ADA0M0	ADA0CE	0	ADA0MD1	ADA0MD0	ADA0ETS1	ADA0ETS0	ADA0TMD	ADA0EF

ADA0CE	A/D 转换控制
0	停止A/D 转换
1	使能A/D 转换

ADA0MD1	ADA0MD0	A/D 转换器操作模式的设置
0	0	连续选择模式
0	1	连续扫描模式
1	0	单次选择模式
1	1	单次扫描模式

ADA0ETS1	ADA0ETS0	外部触发 (ADTRG 管脚) 输入有效沿的设置
0	0	禁止边沿检测
0	1	下降沿检测
1	0	上升沿检测
1	1	上升和下降沿都检测

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 转换进行中

注意事项 1. 对第 0 位的写操作将被忽略。

2. 当 A/D 转换使能 (ADA0CE = 1) 后, 禁止改变 ADA0M1.ADA0FR2 ~ ADA0M1.ADA0FR0 位。

3. 在下列模式中, A/D 转换停止时 (ADA0CE 位 = 0), 将数据写入 ADAM0, ADA0M2, ADA0S, ADA0PFM 和 ADA0PFT 寄存器, 然后允许 A/D 转换操作 (ADA0CE 位 = 1)。

- 普通转换模式
- 高速转换模式的单脉冲选择模式/单脉冲扫描模式

如果在 A/D 转换过程的其他任何模式中 (ADA0EF = 1) 有数据写入寄存器 ADA0M0, ADA0M2, ADA0S, ADA0PFM 和 ADA0PFT, 根据模式的不同操作如下。

- 软件触发模式
A/D 转换停止并从头开始。
- 硬件触发模式
A/D 转换停止并进入触发器待机状态。

4. 要选择外部触发器模式/定时器触发器模式 (ADA0TMD 位 = 1), 须设置高速转换模式 (ADA0M1.ADA0HS1 位 = 1)。A/D 转换操作允许之后 (ADA0CE 位 = 1) 会插入一次稳定时间, 在此稳定期间不要输入触发信号。

5. 当不使用 A/D 转换器时, 设置 ADA0CE 位为 0, 停止操作以减少功耗。

(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。
该寄存器可由 1 位或 8 位存储器操作指令读写。
复位使该位清零。

<R>

复位后: 00H R/W 地址: FFFFF201H

7 6 5 4 3 2 1 0

ADA0HS1	0	0	0	ADA0FR3 ^注	ADA0FR2	ADA0FR1	ADA0FR0
---------	---	---	---	----------------------	---------	---------	---------

ADA0HS1	正常转换模式/高速模式(A/D 转换时间)选择
0	正常转换模式
1	高速转换模式

<R>

注 只适用于 V850ES/SG2-H

<R>

注意事项 1. A/D 转换使能 (ADA0M0.ADA0CE = 1) 后，禁止改变 ADA0M1 的值。

<R>

2. 要选择外部触发器模式/定时器触发器模式 (ADAM0.ADA0TMD 位 = 1)，须设置高速转换模式 (ADA0M1.ADA0HS1 位 = 1)。A/D 转换操作允许之后 (ADA0CE 位 = 1) 会插入一次稳定时间，在此稳定期间不要输入触发信号。

<R>

3. 在 V850ES/SG2 中，确认将第 6 位 ~ 第 3 位清零，并且在 V850ES/SG2-H 中确认将第 6 位 ~ 第 4 位清零。

备注

A/D 转换时间设置示例，见表 13-2 和 13-3。

<R>

表 13-2. 表 13-2. 正常转换模式下转换时间的选择 (ADA0HS1 位 = 0)

ADA0FR3 ~ ADA0FR0 位 注	A/D 转换时间					
	稳定时间 + 转换时间 + 等待时间	$f_{xx} = 32 \text{ MHz}$ 注 2	$f_{xx} = 20 \text{ MHz}$	$f_{xx} = 16 \text{ MHz}$	$f_{xx} = 4 \text{ MHz}$	触发响应 时间
0000	$66/f_{xx}$ ($13/f_{xx} + 26/f_{xx} + 27/f_{xx}$)	禁止设置	禁止设置	禁止设置	$16.50 \mu\text{s}$	$3/f_{xx}$
0001	$131/f_{xx}$ ($26/f_{xx} + 52/f_{xx} + 53/f_{xx}$)	禁止设置	$6.55 \mu\text{s}$	$8.19 \mu\text{s}$	禁止设置	$3/f_{xx}$
0010	$196/f_{xx}$ ($39/f_{xx} + 78/f_{xx} + 79/f_{xx}$)	禁止设置	$9.80 \mu\text{s}$	$12.25 \mu\text{s}$	禁止设置	$3/f_{xx}$
0011	$259/f_{xx}$ ($50/f_{xx} + 104/f_{xx} + 105/f_{xx}$)	$8.09 \mu\text{s}$	$12.95 \mu\text{s}$	$16.19 \mu\text{s}$	禁止设置	$3/f_{xx}$
0100	$311/f_{xx}$ ($50/f_{xx} + 130/f_{xx} + 131/f_{xx}$)	$9.72 \mu\text{s}$	$15.55 \mu\text{s}$	$19.44 \mu\text{s}$	禁止设置	$3/f_{xx}$
0101	$363/f_{xx}$ ($50/f_{xx} + 156/f_{xx} + 157/f_{xx}$)	$11.34 \mu\text{s}$	$18.15 \mu\text{s}$	$22.69 \mu\text{s}$	禁止设置	$3/f_{xx}$
0110	$415/f_{xx}$ ($50/f_{xx} + 182/f_{xx} + 183/f_{xx}$)	$12.97 \mu\text{s}$	$20.75 \mu\text{s}$	禁止设置	禁止设置	$3/f_{xx}$
0111	$467/f_{xx}$ ($50/f_{xx} + 208/f_{xx} + 209/f_{xx}$)	$14.59 \mu\text{s}$	$23.35 \mu\text{s}$	禁止设置	禁止设置	$3/f_{xx}$
1000	$519/f_{xx}$ ($50/f_{xx} + 234/f_{xx} + 235/f_{xx}$)	$16.22 \mu\text{s}$	禁止设置	禁止设置	禁止设置	$3/f_{xx}$
1001	$571/f_{xx}$ ($50/f_{xx} + 260/f_{xx} + 261/f_{xx}$)	$17.84 \mu\text{s}$	禁止设置	禁止设置	禁止设置	$3/f_{xx}$
1010	$623/f_{xx}$ ($50/f_{xx} + 286/f_{xx} + 287/f_{xx}$)	$19.47 \mu\text{s}$	禁止设置	禁止设置	禁止设置	$3/f_{xx}$
1011	$675/f_{xx}$ ($50/f_{xx} + 312/f_{xx} + 313/f_{xx}$)	$21.09 \mu\text{s}$	禁止设置	禁止设置	禁止设置	$3/f_{xx}$
除上面以外	禁止设置					

- 注 1. V850ES/SG2: ADA0FR2 ~ ADA0FR0 位
V850ES/SG2-H: ADA0FR3 ~ ADA0FR0 位
2. 只适用于 V850ES/SG2-H

备注 稳定时间: A/D 转换器建立时间 ($1 \mu\text{s}$ 或更长)

转换时间: 实际 A/D 转换时间 ($2.6 \sim 10.4 \mu\text{s}$)

等待时间: 在下次转换前插入的等待时间

触发响应时间: 如果在稳定时间后产生一个软件触发, 外部触发或定时器触发, 该响应时间将插入到转换时间前。

正常转换模式下, 转换在稳定时间结束后开始。设置 ADA0M0.ADA0CE = 1 后, 稳定时间开始。A/D 转换只在转换时间 ($2.6 \sim 10.4 \mu\text{s}$) 内执行。转换结束后操作停止, 等待时间结束后, 产生中断请求信号 (INTAD)。

由于在等待时间内转换操作已经停止, 所以工作电流降低。

注意事项 设置 $2.6 \mu\text{s} \leq \text{转换时间} \leq 10.4 \mu\text{s}$ 。

<R>

表 13-3. 高速转换模式下转换时间的选择 (ADA0HS1 位 = 1)

ADA0FR3 ~ ADA0FR0 位 注	A/D 转换时间					
	转换时间 (+ 稳定时间)	$f_{XX} = 32 \text{ MHz}^{\#2}$	$f_{XX} = 20 \text{ MHz}$	$f_{XX} = 16 \text{ MHz}$	$f_{XX} = 4 \text{ MHz}$	触发响应时间
0000	$26/f_{XX} (+ 13/f_{XX})$	禁止设置	禁止设置	禁止设置	$6.5 \mu\text{s}$ (+ $3.25 \mu\text{s}$)	$3/f_{XX}$
0001	$52/f_{XX} (+ 26/f_{XX})$	禁止设置	$2.6 \mu\text{s}$ (+ $1.3 \mu\text{s}$)	$3.25 \mu\text{s}$ (+ $1.625 \mu\text{s}$)	禁止设置	$3/f_{XX}$
0010	$78/f_{XX} (+ 39/f_{XX})$	禁止设置	$3.9 \mu\text{s}$ (+ $1.95 \mu\text{s}$)	$4.875 \mu\text{s}$ (+ $2.4375 \mu\text{s}$)	禁止设置	$3/f_{XX}$
0011	$104/f_{XX} (+ 50/f_{XX})$	$3.25 \mu\text{s}$ (+ $1.5625 \mu\text{s}$)	$5.2 \mu\text{s}$ (+ $2.5 \mu\text{s}$)	$6.5 \mu\text{s}$ (+ $3.125 \mu\text{s}$)	禁止设置	$3/f_{XX}$
0100	$130/f_{XX} (+ 50/f_{XX})$	$4.0625 \mu\text{s}$ (+ $1.5625 \mu\text{s}$)	$6.5 \mu\text{s}$ (+ $2.5 \mu\text{s}$)	$8.125 \mu\text{s}$ (+ $3.125 \mu\text{s}$)	禁止设置	$3/f_{XX}$
0101	$156/f_{XX} (+ 50/f_{XX})$	$4.875 \mu\text{s}$ (+ $1.5625 \mu\text{s}$)	$7.8 \mu\text{s}$ (+ $2.5 \mu\text{s}$)	$9.75 \mu\text{s}$ (+ $3.125 \mu\text{s}$)	禁止设置	$3/f_{XX}$
0110	$182/f_{XX} (+ 50/f_{XX})$	$5.6875 \mu\text{s}$ (+ $1.5625 \mu\text{s}$)	$9.1 \mu\text{s}$ (+ $2.5 \mu\text{s}$)	禁止设置	禁止设置	$3/f_{XX}$
0111	$208/f_{XX} (+ 50/f_{XX})$	$6.5 \mu\text{s}$ (+ $1.5625 \mu\text{s}$)	$10.4 \mu\text{s}$ (+ $2.5 \mu\text{s}$)	禁止设置	禁止设置	$3/f_{XX}$
1000	$234/f_{XX} (+ 50/f_{XX})$	$7.3125 \mu\text{s}$ (+ $1.5625 \mu\text{s}$)	禁止设置	禁止设置	禁止设置	$3/f_{XX}$
1001	$260/f_{XX} (+ 50/f_{XX})$	$8.125 \mu\text{s}$ (+ $1.5625 \mu\text{s}$)	禁止设置	禁止设置	禁止设置	$3/f_{XX}$
1010	$286/f_{XX} (+ 50/f_{XX})$	$8.9375 \mu\text{s}$ (+ $1.5625 \mu\text{s}$)	禁止设置	禁止设置	禁止设置	$3/f_{XX}$
1011	$312/f_{XX} (+ 50/f_{XX})$	$9.75 \mu\text{s}$ (+ $1.5625 \mu\text{s}$)	禁止设置	禁止设置	禁止设置	$3/f_{XX}$
除上面以外	禁止设置					

- 注 1. V850ES/SG2: ADA0FR2 到 ADA0FR0 位
V850ES/SG2-H: ADA0FR3 到 ADA0FR0 位
2. 只适用于 V850ES/SG2-H

备注 转换时间: 实际 A/D 转换时间 (2.6 ~ 10.4 μs)
 稳定时间: A/D 转换器建立时间 (1 μs 或更长)
 触发响应时间: 如果在稳定时间后产生一个软件触发, 外部触发或定时器触发, 该响应时间将插入到转换时间前。

高速转换模式下, 转换在稳定时间结束后开始。设置 ADA0M0.ADA0CE = 1 后, 稳定时间开始。A/D 转换只在转换时间 (2.6 ~ 10.4 μs) 内执行。转换结束后操作停止后, 立刻产生中断请求信号 (INTAD)。连续转换模式下, 只在第一次转换前插入稳定时间, 在第二次及以后的转换不插入稳定时间 (A/D 转换器持续运行)。

注意事项 设置 $2.6 \mu\text{s} \leq \text{转换时间} \leq 10.4 \mu\text{s}$ 。

(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。

该寄存器可由 1 位或 8 位存储器操作指令读写。

复位使该寄存器清零。

复位后: 00H		R/W	地址: FFFFF203H					
ADA0M2	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	ADA0TMD1	ADA0TMD0

ADA0TMD1	ADA0TMD0	硬件触发模式的设置
0	0	外部触发模式(当ADTRG 管脚有效沿被检测时)
0	1	定时器触发模式 0 (当INTTP2CC0 中断请求被产生时)
1	0	定时器触发模式 1 (当INTTP2CC1 中断请求被产生时)
1	1	禁止设置

<R>

注意事项 1. 在下列模式中，A/D 转换停止时（ADA0M0.ADA0CE 位 = 0），将数据写入 ADA0M2 寄存器，然后允许 A/D 转换操作（ADA0CE 位 = 1）。

- 普通转换模式
- 高速转换模式的单次选择模式/单次扫描模式

2. 确保第 7 位 ~ 第 2 位清零。

(4) 模拟输入通道指定寄存器 (ADA0S)

该寄存器选择输入模拟电平的引脚。由该引脚输入的信号将被转换为数字信号。

该寄存器可由 1 位或 8 位存储器操作指令读写。

复位使该寄存器清零。

复位后: 00H R/W 地址: FFFFF202H

	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 ~ ANI2
0	0	1	1	ANI3	ANI0 ~ ANI3
0	1	0	0	ANI4	ANI0 ~ ANI4
0	1	0	1	ANI5	ANI0 ~ ANI5
0	1	1	0	ANI6	ANI0 ~ ANI6
0	1	1	1	ANI7	ANI0 ~ ANI7
1	0	0	0	ANI8	ANI0 ~ ANI8
1	0	0	1	ANI9	ANI0 ~ ANI9
1	0	1	0	ANI10	ANI0 ~ ANI10
1	0	1	1	ANI11	ANI0 ~ ANI11
1	1	0	0	禁止设置	禁止设置
1	1	0	1	禁止设置	禁止设置
1	1	1	0	禁止设置	禁止设置
1	1	1	1	禁止设置	禁止设置

- <R> 注意事项 1. 在下列模式中，A/D 转换停止时（ADA0M0.ADA0CE 位 = 0），将数据写入 ADA0S 寄存器，然后允许 A/D 转换操作（ADA0CE 位 = 1）。
- 普通转换模式
 - 高速转换模式的单次选择模式/单次扫描模式
2. 确保第 7 位～第 4 位清零。

(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 位访问 ADA0CRnH 寄存器。10 位转换结果存于寄存器 ADA0CRn 的高 10 位，其低 6 位的值为 0。转换结果的高 8 位存于 ADA0CRnH。

注意事项 下述状态时禁止访问寄存器 ADA0CRn 和 ADA0CRnH。

详细情况，参见 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。

- 当 CPU 使用副时钟，且主时钟停止振荡
- 当 CPU 使用内部振荡时钟

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,
ADA0CR10 FFFFF224H, ADA0CR11 FFFFF226H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADA0CRn (n = 0 到 11)	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

复位后: 未定义 R 地址: ADA0CR0H FFFFF211H, ADA0CR1H FFFFF213H,
ADA0CR2H FFFFF215H, ADA0CR3H FFFFF217H,
ADA0CR4H FFFFF219H, ADA0CR5H FFFFF21BH,
ADA0CR6H FFFFF21DH, ADA0CR7H FFFFF21FH,
ADA0CR8H FFFFF221H, ADA0CR9H FFFFF223H,
ADA0CR10H FFFFF225H, ADA0CR11H FFFFF227H

	7	6	5	4	3	2	1	0
ADA0CRnH (n = 0 到 11)	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

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，正确转换结果可能无法读取。

如下为模拟信号输入引脚（ANI0 到 ANI11）和 A/D 转换结果（ADA0CRn）的关系。

$$ADA0CR = \text{INT} \left(\frac{V_{IN}}{AV_{REF0}} \times 1,024 + 0.5 \right)$$

$$\text{ADA0CR}^{\text{注}} = \text{SAR} \times 64$$

或者，

$$(SAR - 0.5) \times \frac{AV_{REF0}}{1,024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{REF0}}{1,024}$$

INT ()：函数，返回 () 中的整数值

V_{IN}: 模拟输入电压

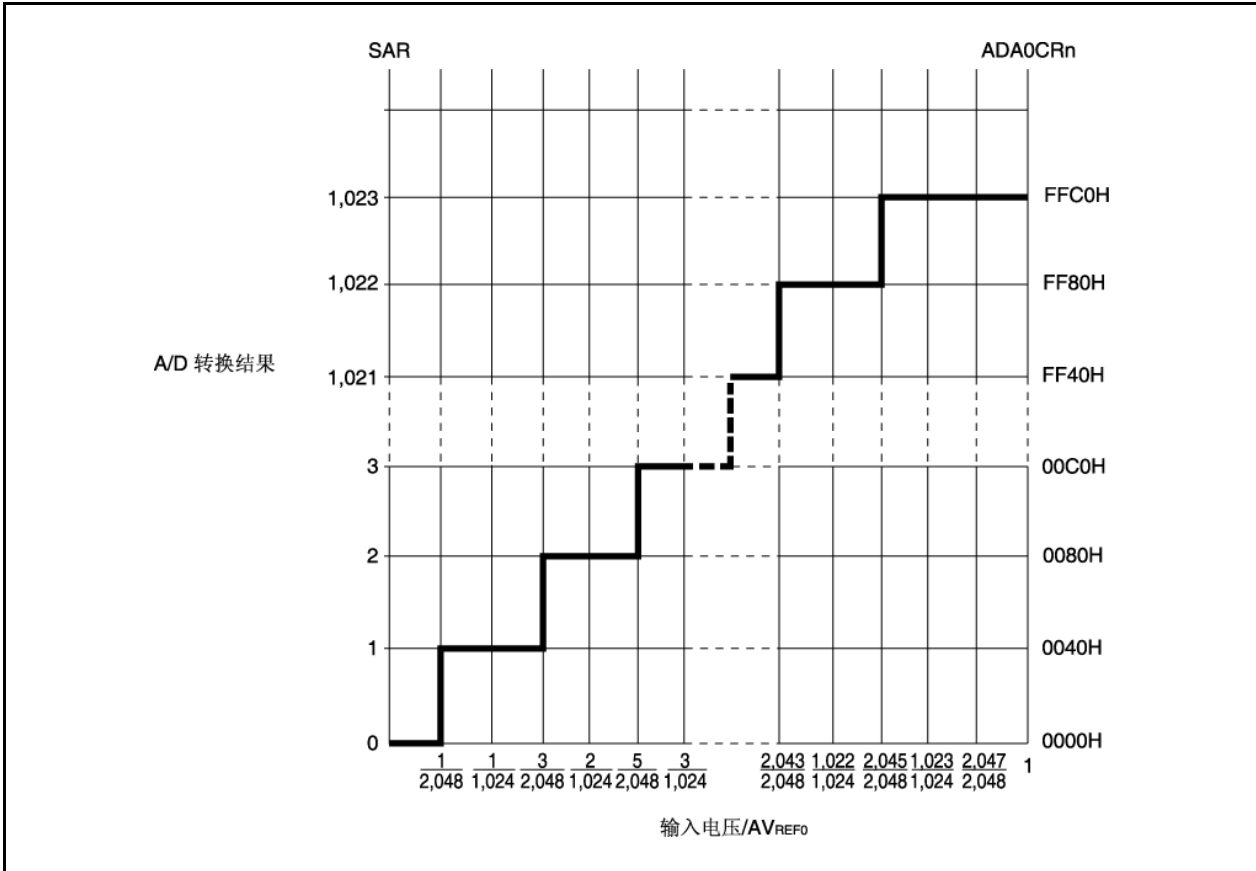
AVREF0: AVREF0 引脚电压

ADA0CR: 寄存器 ADA0CRn 的值

注 ADA0CRn 的低 6 位恒为 0。

如下显示了模拟输入电压和 A/D 转换结果之间的关系。

图 13-2. 模拟输入电压和 A/D 转换结果之间的关系



(6) 电压不足比较模式寄存器 (ADA0PFM)

该寄存器是 8 位寄存器，用于设置电压不足比较模式。

可由 1 位或 8 位存储器操作指令读写该寄存器。

复位使该寄存器清零。

复位后: 00H R/W 地址: FFFFF204H

	<7>	6	5	4	3	2	1	0
ADA0PFM	ADA0PFE	ADA0PFC	0	0	0	0	0	0

ADA0PFE	电压不足比较使能/禁止选项
0	电压不足比较禁止
1	电压不足比较使能

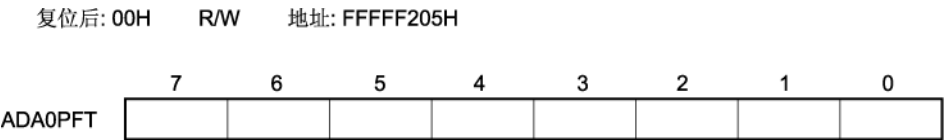
ADA0PFC	电压不足比较模式的选择
0	当 $ADA0CRnH \geq ADA0PFT$ 时，产生一个中断请求信号(INTAD)。
1	当 $ADA0CRnH < ADA0PFT$ 时，产生一个中断请求信号(INTAD)。

- 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**。
 3. 在下列模式中，在 A/D 转换停止时 (**ADA0M0.ADA0CE** 位 = 0)，将数据写入 **ADA0PFM** 寄存器，然后允许 A/D 转换操作 (**ADA0CE** 位 = 1)。
 - 普通转换模式
 - 高速转换模式的单次选择模式/单次扫描模式

<R>

(7) 电源电压不足比较限定值寄存器 (ADA0PFT)

该寄存器用于设置电源电压不足比较模式。
该寄存器可由 1 位或 8 位存储器操作指令读写。
复位使该寄存器清零。



<R> 注意事项 在下列模式中，A/D 转换停止时（ADA0M0.ADA0CE 位 = 0），将数据写入 ADA0PFT 寄存器，然后允许 A/D 转换操作（ADA0CE 位 = 1）。

- 普通转换模式
- 高速转换模式的单次选择模式/单次扫描模式

13.5 操作

13.5.1 基本操作

- <1> 使用寄存器 ADA0M0, ADA0M1, ADA0M2 和 ADA0S 设置操作模式、触发模式、和执行 A/D 转换的时间。当寄存器的 ADA0CE 位被设置, 或在软件触发模式下启动转换, 而在外部/时间触发模式下需要等待 A/D 转换的触发。
- <2> A/D 转换启动后, 采样保持电路对选中的模拟输入通道的输入电压进行采样。
- <3> 当采样保持电路在指定时间对输入通道采样, 电路进入保持状态, 保持输入电压直到 A/D 转换完成。
- <4> 设置逐次逼近寄存器 (SAR) 的第 9 位。分级选择器选中 (1/2) AV_{REF0} 作为串联电阻串的电压分级。
- <5> 串联电阻串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 位保持为 1。如果模拟输入电压低于 (1/2) AV_{REF0} , 则 SAR 寄存器的 MSB 复位为 0。
- <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, 串联电阻串的电压分级选择如下。
 - 第 9 位 = 1: (3/4) AV_{REF0}
 - 第 9 位 = 0: (1/4) AV_{REF0}
 电压阶与模拟输入电压进行比较, 根据结果, 设置第 8 位, 如下所示。
 模拟输入电压 $\geq$ 电压阶: 第 8 位 = 1
 模拟输入电压 $\leq$ 电压阶: 第 8 位 = 0
- <7> 按此方式继续进行比较, 直至 SAR 的第 0 位。
- <8> 全部 10 位比较完成后, 在 SAR 中保留一个有效的数值结果, 然后将结果传送至 A/D 转换结果寄存器 ADA0CRn 中。同时也会产生 A/D 转换结束中断请求 (INTAD)。
- <9> 单次模式下, 转换被停止^注。单次扫描模式下, 扫描一次后转换被停止^注。连续选择模式下, 重复步骤<2> ~ <8>, 直至 ADA0M0.ADA0CE 被清零。连续扫描模式下, 重复步骤<2> ~ <8>以扫描每个通道。

注 外部触发模式/定时器触发模式 0/定时器触发模式 1 下, 进入触发等待状态。

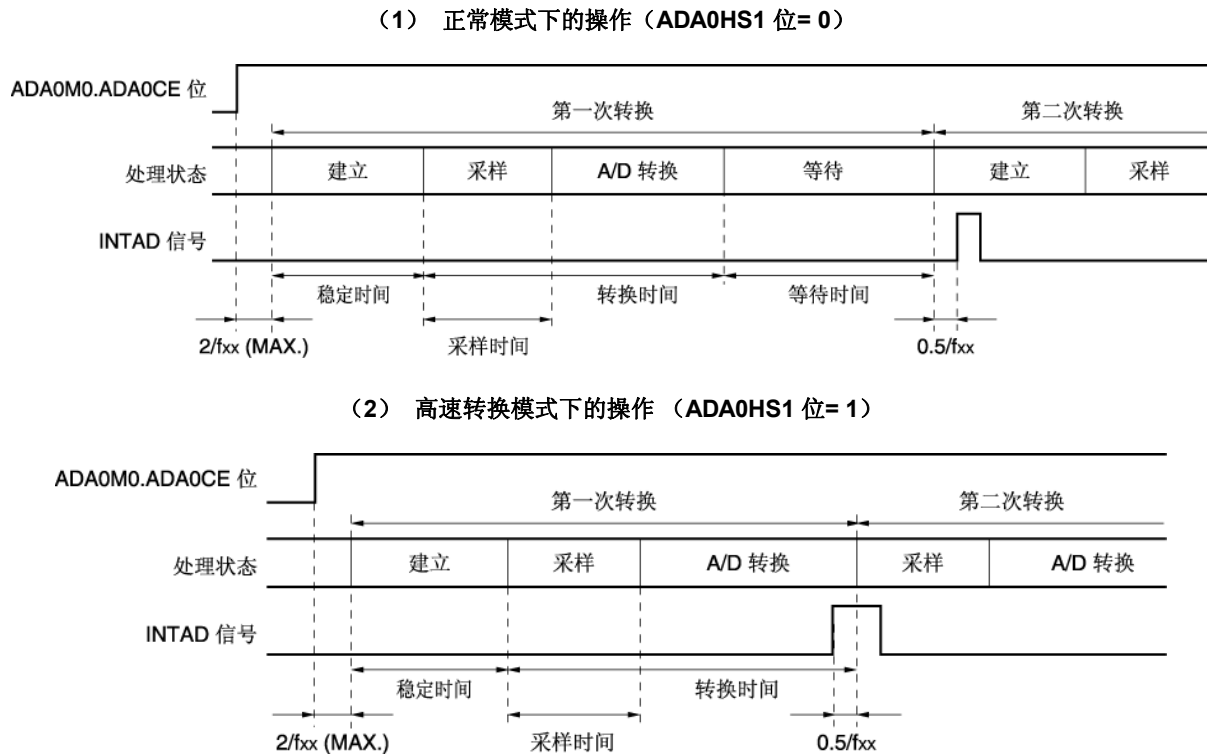
<R>

备注 触发器待机状态是指稳定时间之后的状态。

13.5.2 转换操作时序

<R>

图 13-3. 转换操作时序（连续转换）



ADA0FR3 到 ADA0FR0 位 ^注	稳定时间:	转换时间 (采样时间)	等待时间:	触发响应时间:
0000	13/f _{xx}	26/f _{xx} (4/f _{xx})	27/f _{xx}	3/f _{xx}
0001	26/f _{xx}	52/f _{xx} (8/f _{xx})	53/f _{xx}	3/f _{xx}
0010	39/f _{xx}	78/f _{xx} (12/f _{xx})	79/f _{xx}	3/f _{xx}
0011	50/f _{xx}	104/f _{xx} (16/f _{xx})	105/f _{xx}	3/f _{xx}
0100	50/f _{xx}	130/f _{xx} (20/f _{xx})	131/f _{xx}	3/f _{xx}
0101	50/f _{xx}	156/f _{xx} (24/f _{xx})	157/f _{xx}	3/f _{xx}
0110	50/f _{xx}	182/f _{xx} (28/f _{xx})	183/f _{xx}	3/f _{xx}
0111	50/f _{xx}	208/f _{xx} (32/f _{xx})	209/f _{xx}	3/f _{xx}
1000	50/f _{xx}	234/f _{xx} (36/f _{xx})	235/f _{xx}	3/f _{xx}
1001	50/f _{xx}	260/f _{xx} (40/f _{xx})	261/f _{xx}	3/f _{xx}
1010	50/f _{xx}	286/f _{xx} (44/f _{xx})	287/f _{xx}	3/f _{xx}
1011	50/f _{xx}	312/f _{xx} (48/f _{xx})	313/f _{xx}	3/f _{xx}
除上面以外	禁止设置			

注 V850ES/SG2: ADA0FR2 到 ADA0FR0 位
V850ES/SG2-H: ADA0FR3 到 ADA0FR0 位

备注 上述时序适用于稳定时间内产生触发的情况。如果稳定时间过后再产生触发，将会插入触发响应时间。

13.5.3 触发模式

通过设置触发模式确定启动转换操作的时序。触发模式包括软件触发模式和硬件触发模式。硬件触发模式包括定时器触发模式 0 和 1，和外部触发模式。ADA0M0.ADA0TMD 用于设置 ADA0M0.ADA0TMD。ADA0M2.ADA0TMD1 和 ADA0M2.ADA0TMD0 位用于设置硬件 ADA0M0.ADA0TMD。

(1) 软件触发模式

当 ADA0M0.ADA0CE = 1，由寄存器 ADA0S 设置的模拟输入引脚（ANI0 ~ ANI11）输入的信号被转换。转换完成后，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，除非第一次转换完成后设置 ADA0CE = 0，下一次转换开始。如果操作模式是单次选择/扫描模式，转换执行一次后结束。

当转换开始，ADA0M0.ADA0EF = 1（表明转换正在进行）。

<R>

如果 ADA0M0、ADA0M2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器在转换过程中被写入，转换被终止并且从头重新开始。所以，在正常转换模式和高速转换模式的单次选择模式/单次扫描模式中禁止写入这些寄存器。

(2) 外部触发模式

该模式下，当外部触发信号输入到引脚 ADTRG，开始转换由寄存器 ADA0S 设定的模拟输入引脚（ANI0 ~ ANI11）的信号。ADA0M0.ADA0ETS1 和 ADA0M0.ATA0ETS0 用于设置外部触发的检测边沿（即，上升沿，下降沿，或上升和下降沿）。当 ADA0CE 位设置为 1 时，A/D 转换器等待触发并且在外部触发输入后开始转换。

无论是否设置 ADA0MD1 和 ADA0MD0 的连续选择，连续扫描，单次选择，或单次扫描模式，当转换完成，结果存储于寄存器 ADA0CRn 中。同时产生 INTAD 信号，A/D 转换器等待再次触发。

当转换开始，ADA0EF = 1（表明转换正在进行）。然而，当 A/D 转换器等待触发时，ADA0EF = 0（表明转换停止）。如果在转换过程中输入有效触发，转换中止并重新开始。

<R>

如果 ADA0M0、ADA0M2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器在转换过程中被写入，转换不被终止，并且 A/D 转换器等待再次触发。然而，在单次选择模式/单次扫描模式中禁止写入这些寄存器。

<R>

注意事项 要选择外部触发器模式，必须设置高速转换模式。在 A/D 转换操作允许（ADA0M0.ADA0CE 位 = 1）之后会插入一个稳定时间，在此期间内不能输入触发信号。

<R>

备注 触发器待机状态是指稳定时间之后的状态。

(3) 定时器触发模式

该模式下，当连接到定时器的捕捉/比较寄存器产生比较匹配中断请求信号（INTTP2CC0 或 INTTP2CC1）时，由寄存器 ADA0S 设置的模拟输入引脚（ANI0 ~ ANI11）的信号开始转换。使用 ADA0TMD1 和 ADA0TMD0 选择信号 INTTP2CC0 或 INTTP2CC1，在特定比较匹配中断请求信号的上升沿，启动转换。当 ADA0CE = 1，A/D 转换器等待触发，并在比较匹配中断请求信号输入时开始转换。

无论是否设置 ADA0MD1 和 ADA0MD0 的连续选择，连续扫描，单次选择，或单次扫描模式，当转换完成，结果存储于寄存器 ADA0CRn 中。同时产生 INTAD 信号，A/D 转换器等待再次触发。

当转换开始，ADA0EF = 1（表明转换正在进行）。然而，当 A/D 转换器等待触发时，ADA0EF = 0（表明转换停止）。如果在转换过程中输入有效触发，转换中止并重新开始。

<R> 如果 ADA0M0、ADA0M2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器在转换过程中被写入，转换停止，并且 A/D 转换器等待再次触发。然而，在单次选择模式/单次扫描模式中禁止写入这些寄存器。

<R> **注意事项** 要选择外部触发器模式，必须设置高速转换模式。在 A/D 转换操作允许（ADA0M0.ADA0CE 位 = 1）之后会插入一个稳定时间，在此期间内不能输入触发信号。

<R> **备注** 触发器待机状态是指稳定时间之后的状态。

13.5.4 操作模式

有 4 种操作模式可用来设置 ANI0 到 ANI11 引脚：连续选择模式，连续扫描模式，单脉冲选择模式，单脉冲扫描模式。

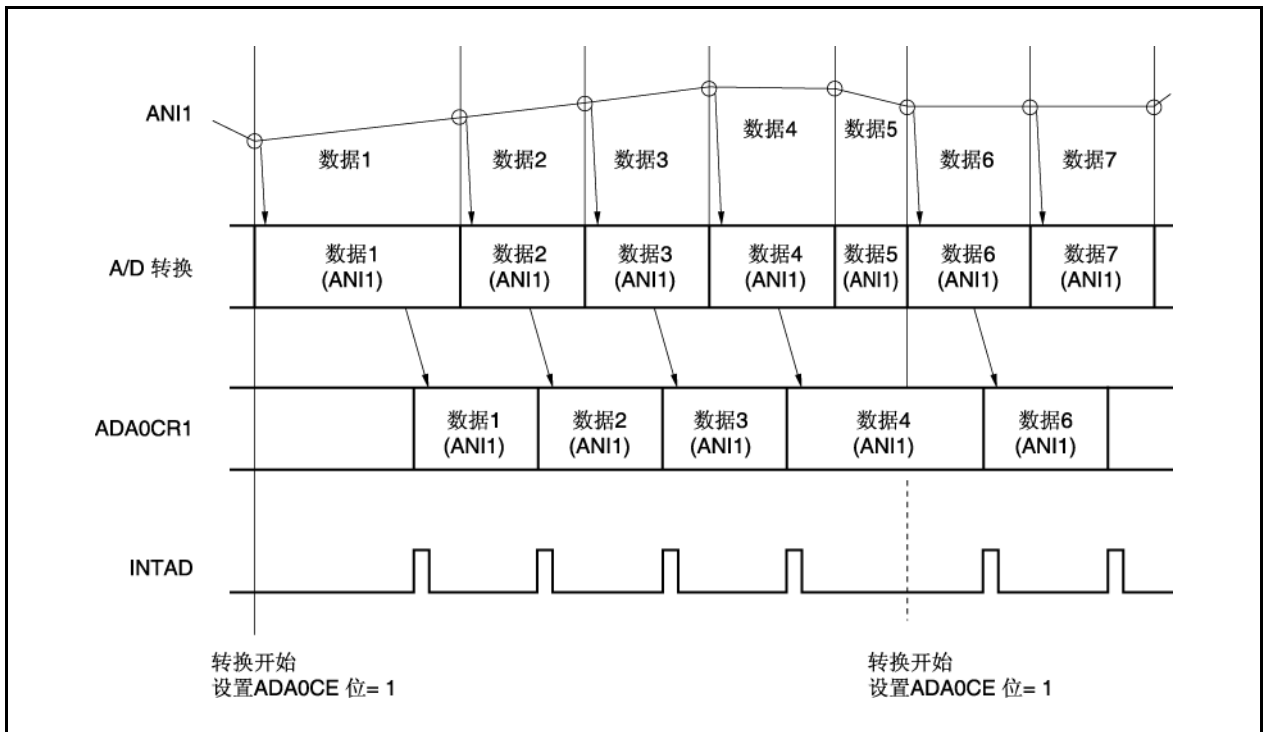
使用 ADA0M0.ADA0MD1 和 ADA0M0.ADA0MD0 设置操作模式。

(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。

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 各自对应。每次 A/D 转换完成，产生 A/D 转换结束中断请求信号 (INTAD)。转换完成后，除非 ADA0M0.ADA0CE = 0 (n = 0 到 11)，否则下一次转换开始。

图 13-4. 连续选择模式的操作时序示例（寄存器 ADA0S = 01H）



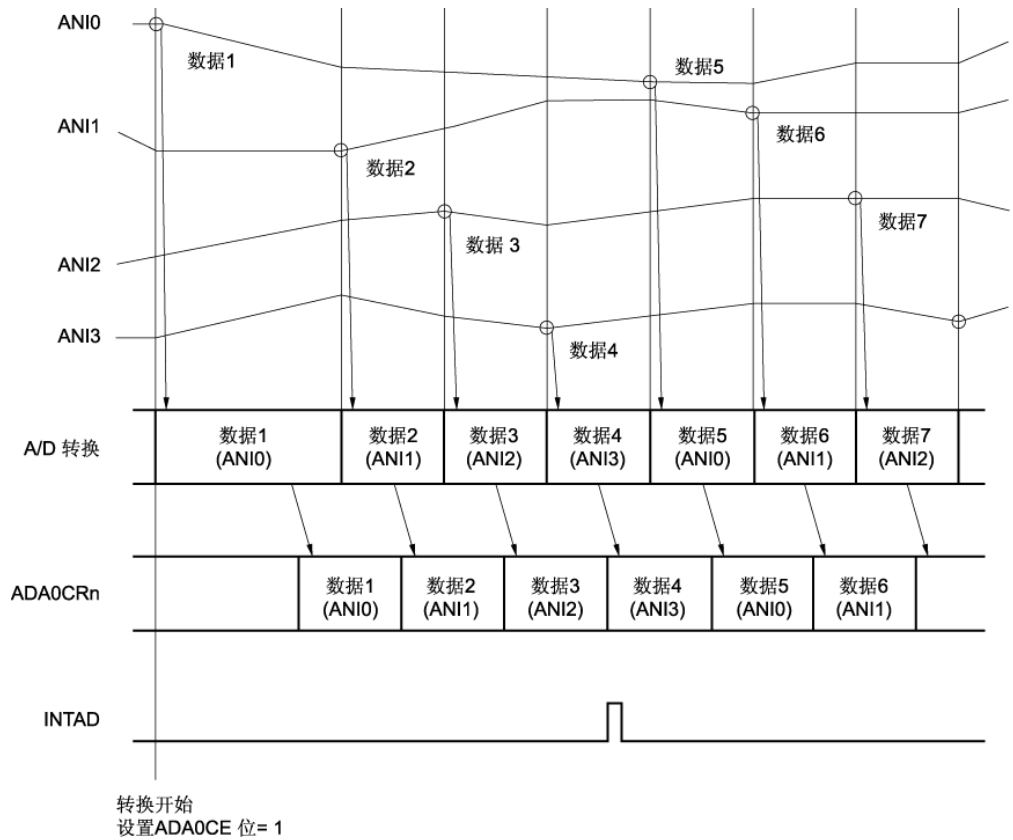
(2) 连续扫描模式

该模式下，模拟输入引脚被依次选择（从引脚 ANI0 到寄存器 ADA0S 所指定的引脚），并将输入的模拟量转换成数字量。

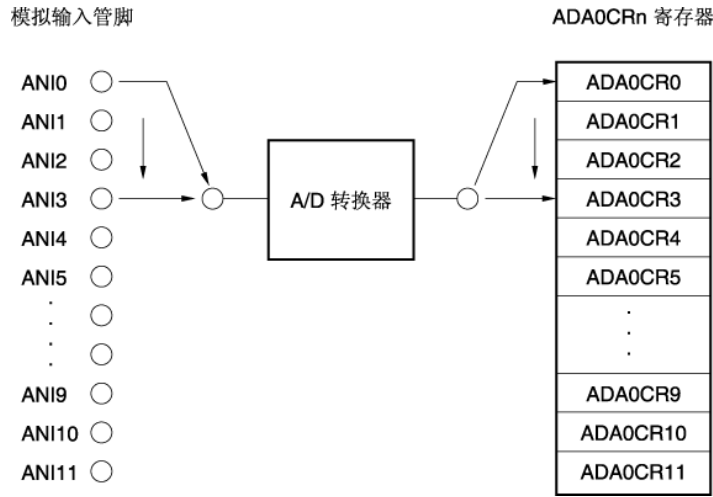
转换结果保存在与相应模拟输入引脚对应的 ADA0CRn 寄存器中。当由 ADA0S 寄存器指定的模拟输入引脚的转换完成之后，会产生 INTAD 信号，然后 A/D 转换从 ANI0 引脚重新开始，除非 ADA0CE 位被清零 (n = 0 到 11)。

图 13-5. 连续扫描模式的操作时序示例（寄存器 ADA0S = 03H）

(a) 时序示例



(b) 框图

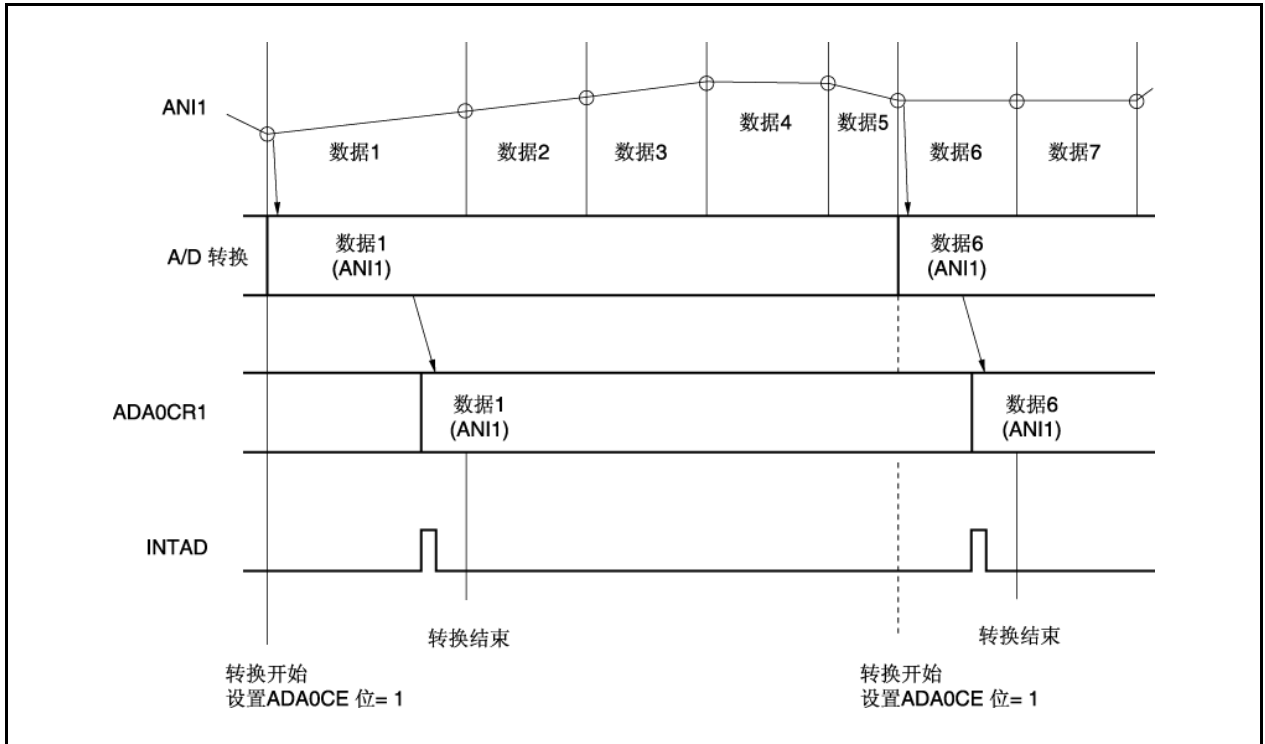


(3) 单脉冲选择模式

该模式下，由寄存器 ADA0S 设置的模拟输入引脚的输入电压转换为数字量，且只转换一次。

相应模拟输入引脚的转换结果存储于寄存器 ADA0CRn 中。该模式下，模拟输入引脚与寄存器 ADA0CRn 各自对应。每次 A/D 转换完成，产生 A/D 转换结束中断请求信号 (INTAD)。A/D 转换 (n = 0 到 11) 完成后，转换操作停止。

图 13-6. 单脉冲选择模式的操作时序示例 (寄存器 ADA0S = 01H)



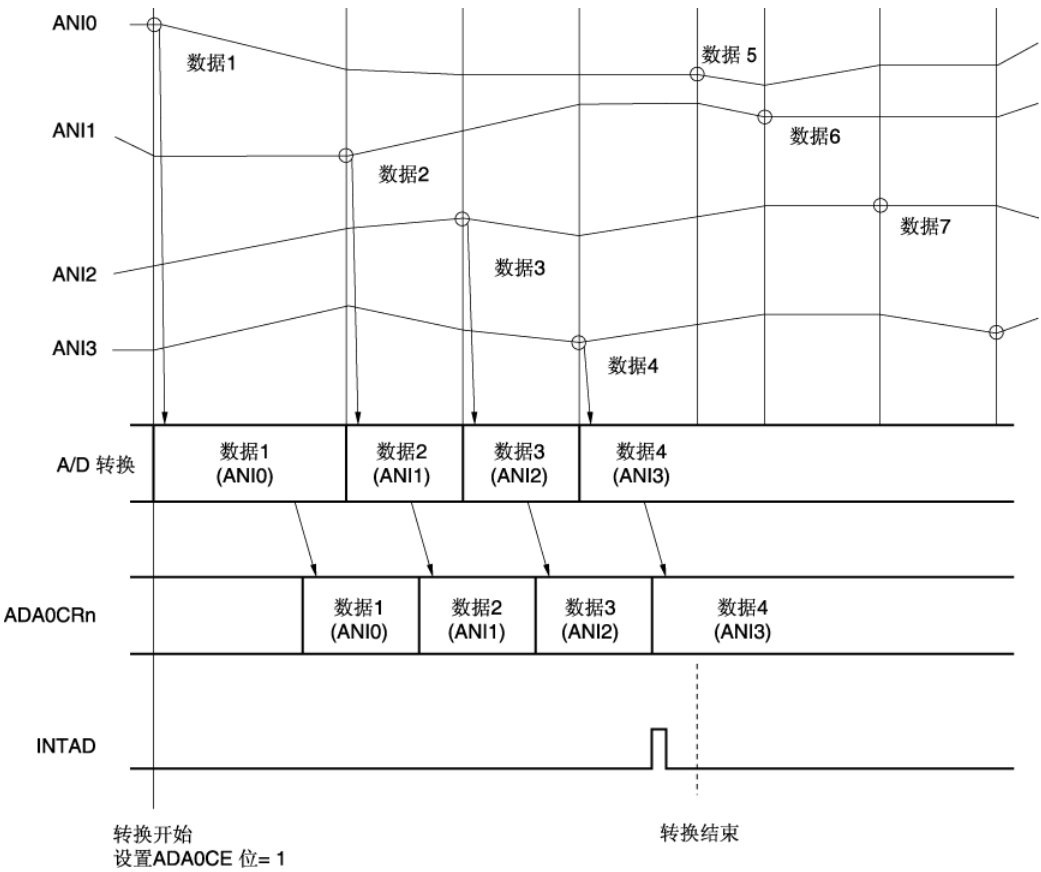
(4) 单脉冲扫描模式

该模式下，模拟输入引脚被依次选择（从引脚 ANI0 到寄存器 ADA0S 所指定的引脚），并将输入的模拟量转换成数字量。

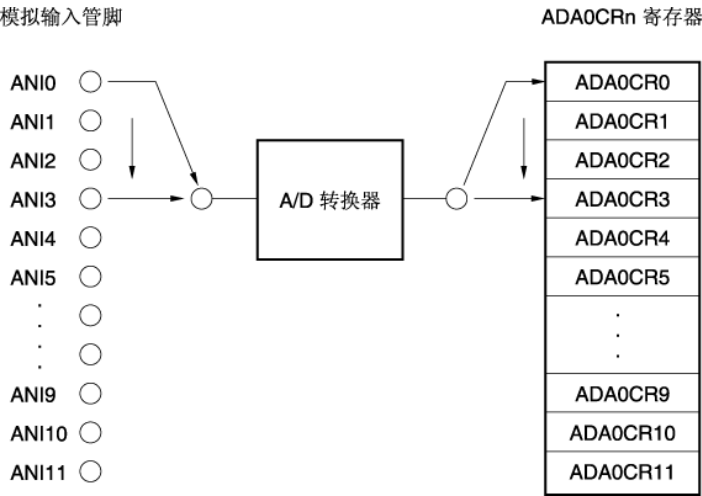
每次转换的结果存储于与模拟输入引脚对应的寄存器 ADA0CRn 中。当寄存器 ADA0S 设置的模拟输入引脚的转换完成时，产生 INTAD 信号。A/D 转换 (n = 0 到 11) 完成后，转换操作停止。

图 13-7. 单脉冲扫描模式的操作时序示例（寄存器 ADA0S = 03H）

(a) 时序示例



(b) 框图



13.5.5 掉电比较模式

可以用寄存器 ADA0PFM 和 ADA0PFT 控制 A/D 转换结束中断请求信号 (INTAD)。

- 当 ADA0PFM.ADA0PFE = 0 时，每当转换完成，则产生信号 INTAD (正常使用 A/D 转换器)。
- 当 ADA0PFE = 1 且 ADA0PFM.ADA0PFC = 0，转换完成后，寄存器 ADA0CRnH 的值与 ADA0PFT 的值比较，只有当 $\text{ADA0CRnH} \geq \text{ADA0PFT}$ 时，才会产生信号 INTAD。
- 当 ADA0PFE = 1 且 ADA0PFC = 1，转换完成后，寄存器 ADA0CRnH 的值与 ADA0PFT 的值比较，只有当 $\text{ADA0CRnH} < \text{ADA0PFT}$ 时，才会产生信号 INTAD。

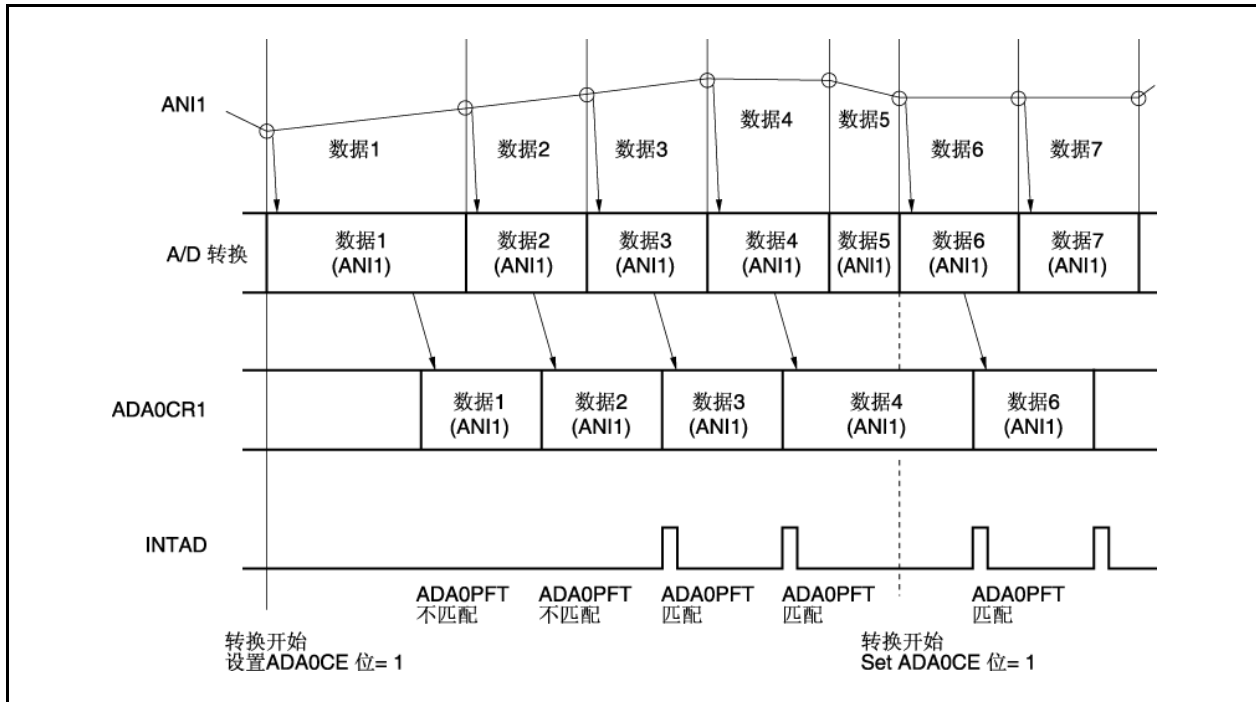
备注 $n = 0 \sim 11$

在掉电比较模式下，ANI0 到 ANI11 引脚可以设置为 4 种模式：连续选择模式，连续扫描模式，单脉冲选择模式，单脉冲扫描模式。

(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电压转换结果与寄存器 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 ADA0M0.ADA0CE = 0 ($n = 0 \sim 11$)，否则开始下一次转换。

图 13-8. 连续选择模式的操作时序示例
(掉电比较模式下: ADA0S = 01H)



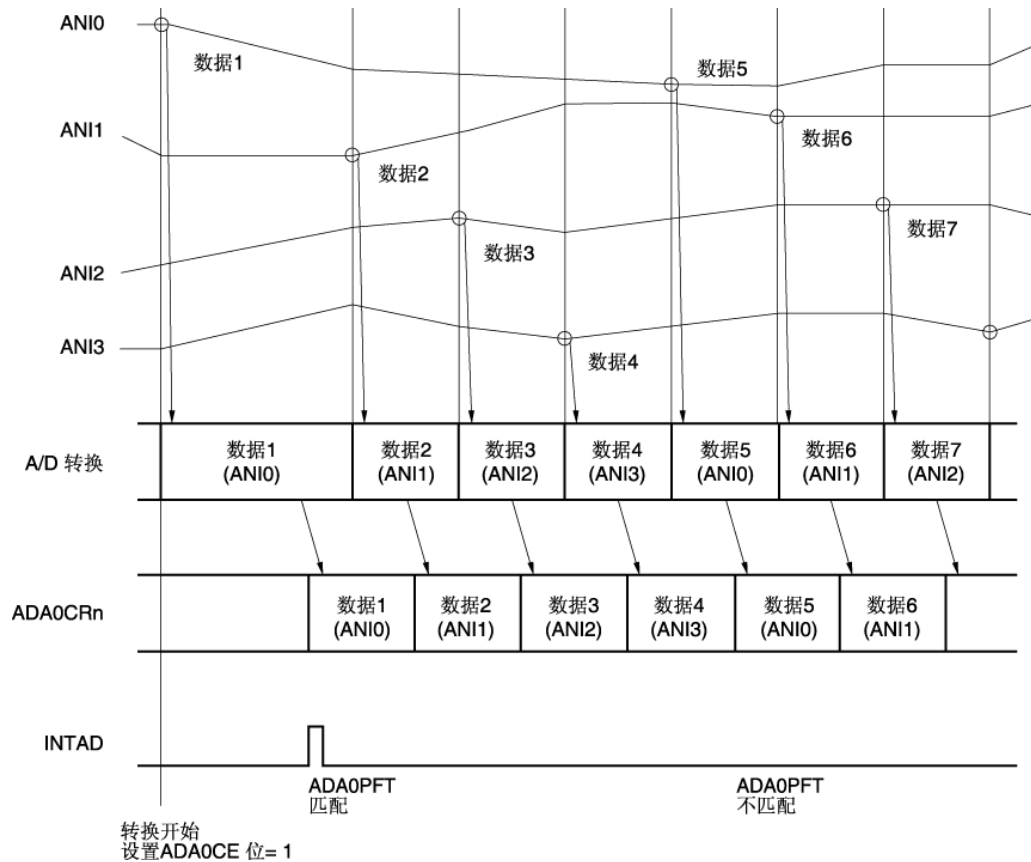
(2) 连续扫描模式

该模式下，模拟输入引脚被依次选择（从引脚 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。

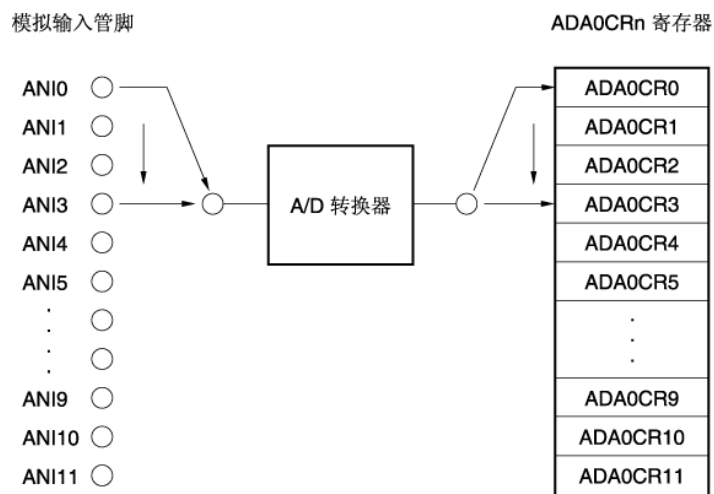
第一次转换的结果存储于寄存器 ADA0CR0 后，模拟输入引脚到寄存器 ADA0S 指定引脚的电压转换结果连续被存储。转换完成后，除非 ADA0M0.ADA0CE = 0 ($n = 0 \sim 11$)，否则开始下一次转换。

图 13-9. 连续扫描模式的操作时序示例
(掉电比较模式下: ADA0S = 03H)

(b) 时序示例



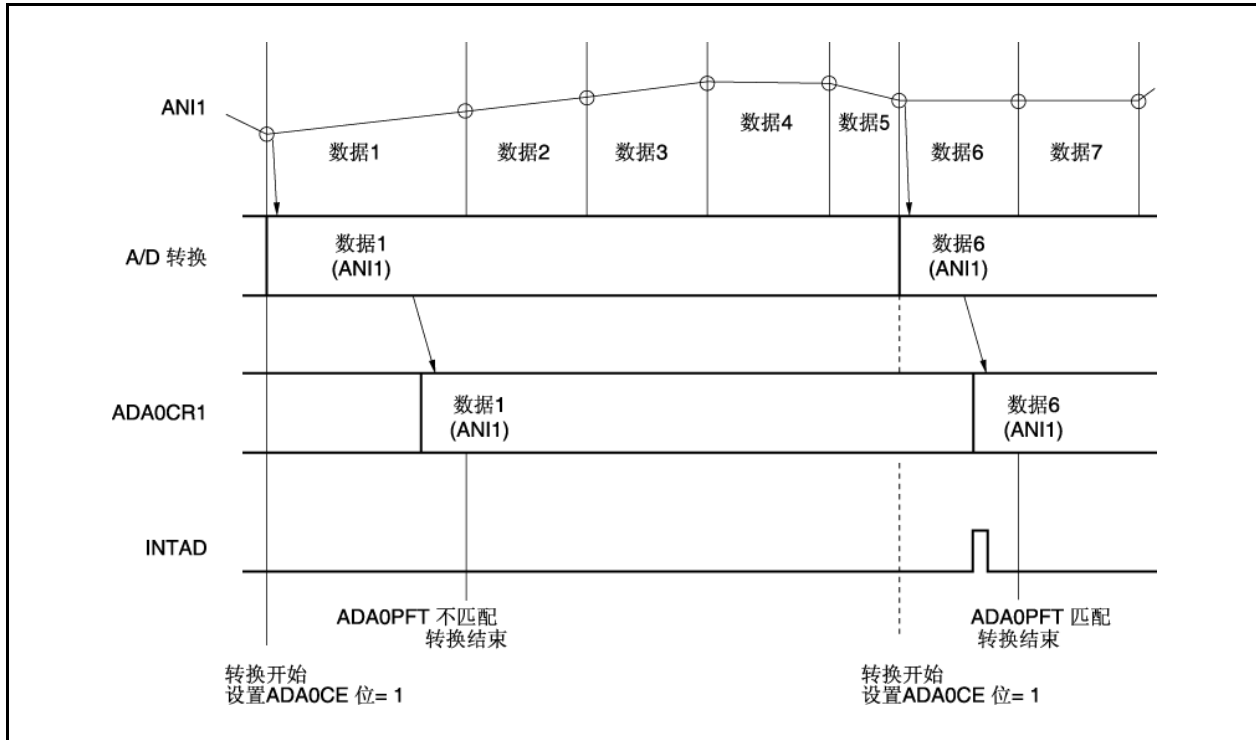
(b) 框图



（3）单脉冲选择模式

该模式下，由寄存器 ADA0S 设置的模拟输入引脚的转换电压结果与寄存器 ADA0PFT 的设置值比较。如果掉电比较的结果与 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。A/D 转换完成后，转换操作停止。

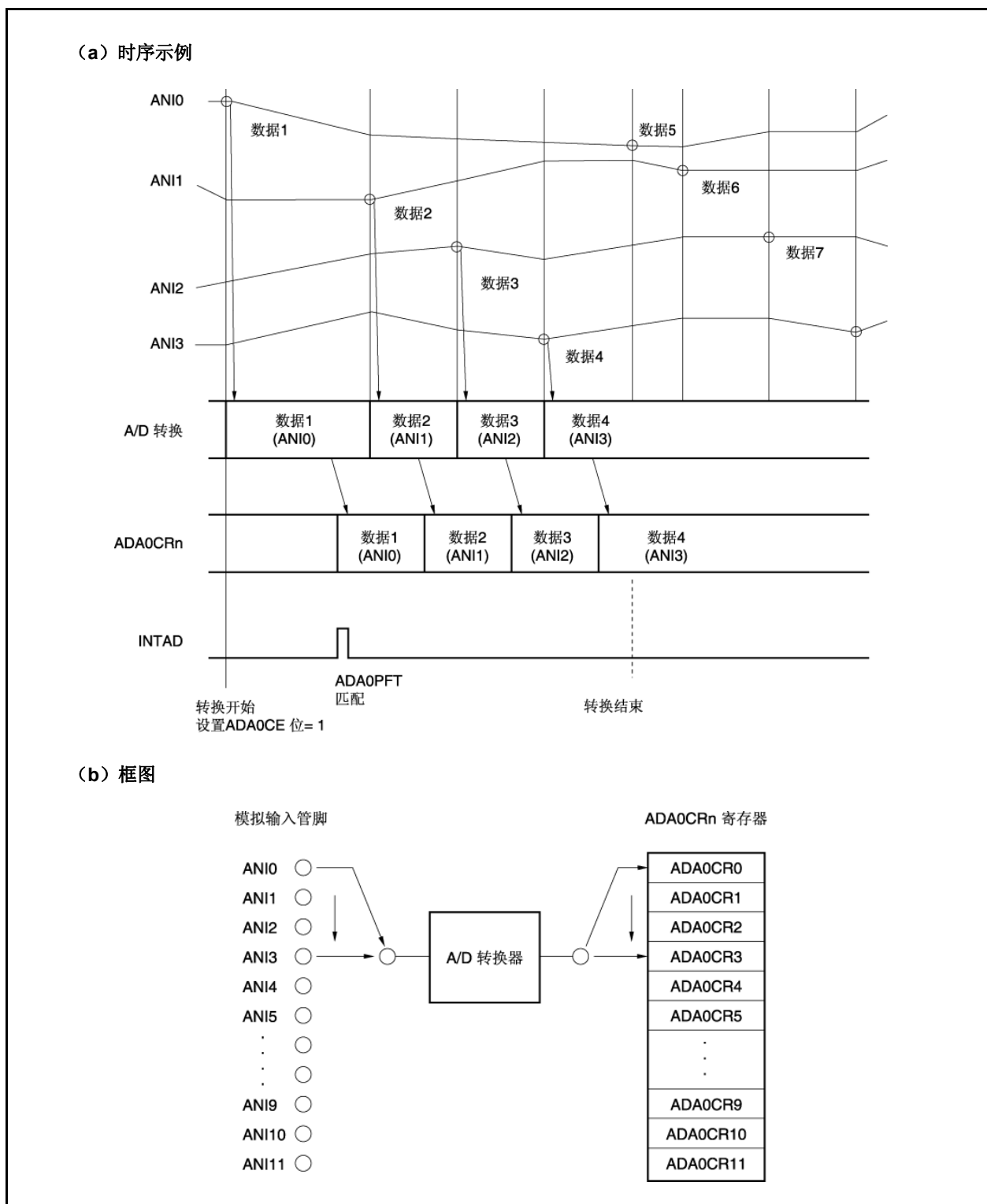
图 13-10. 单脉冲选择模式的操作时序示例
(掉电比较模式下: ADA0S = 01H)



（4）单脉冲扫描模式

该模式下，模拟输入引脚被依次选择（从引脚 ANI0 到寄存器 ADA0S 所设置的指定引脚），模拟输入引脚的电压转换结果被存储，寄存器 ADA0CR0H 通道 0 的设定值与寄存器 ADA0PFT 的值比较。如果掉电比较的结果与 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，寄存器 ADA0S 指定的引脚电压连续转换结果被顺序存储。A/D 转换完成后，转换操作停止。

图 13-11. 单脉冲扫描模式的操作时序示例
(掉电比较模式下: ADA0S = 03H)



13.6 注意事项

(1) 不使用 A/D 转换器

当不使用 A/D 转换器时，设置 $ADA0M0.ADA0CE = 0$ ，可以降低功耗。

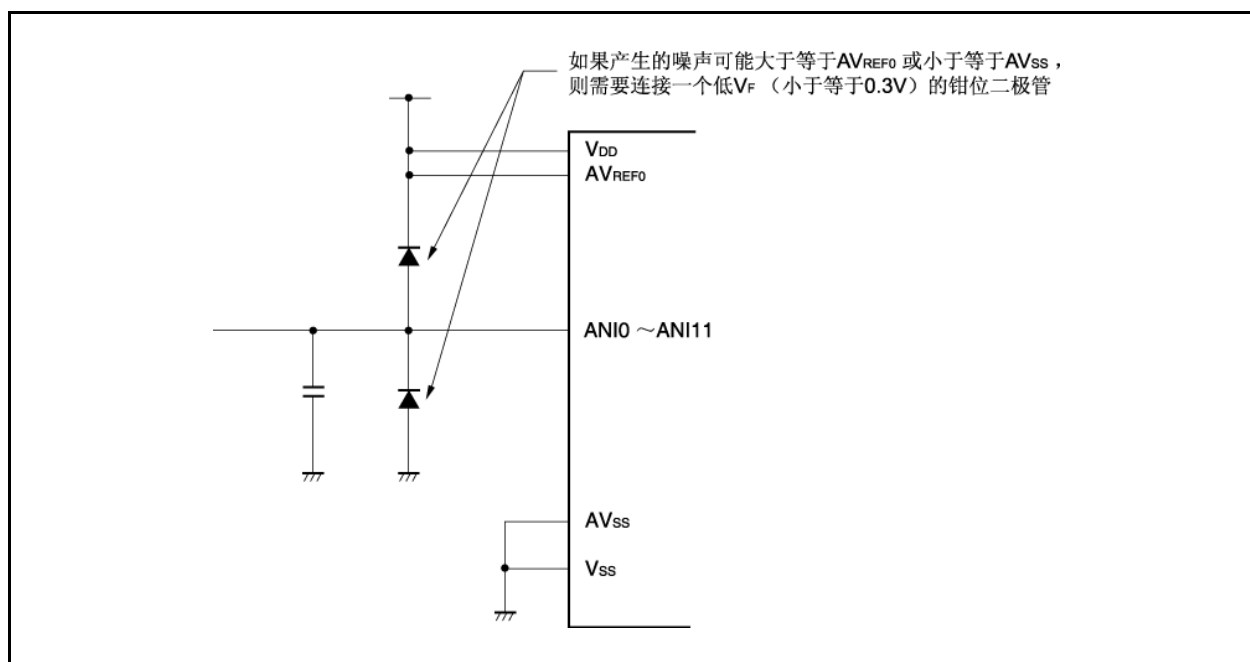
(2) 引脚 ANI0 ~ ANI11 的输入范围

输入到引脚 ANI0 ~ ANI11 的电压值应该不超过指定范围。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，则对应通道的转换值不确定，其它通道的转换值也会受到影响。

(3) 抑制噪声的方法

为了确保 10 位分辨率，引脚 ANI0 ~ ANI11 必须有效抑制噪声。模拟输入源的输出阻抗增加，噪声的影响越大。为了降低噪声，推荐连接一个外部电容，如图 13-12 所示。

图 13-12. 模拟输入引脚的处理



<R>

(4) I/O 复用

模拟输入引脚（**ANI0** ~ **ANI11**）可以功能复用，作为普通端口引脚使用。**AVREF0** 电源引脚既可以为 A/D 转换器提供参考电源，又可以用作端口 7 的 I/O 缓冲器电源。因此，如果在 A/D 转换过程中执行了以下任意处理，可能无法获得期望的 A/D 转换值。

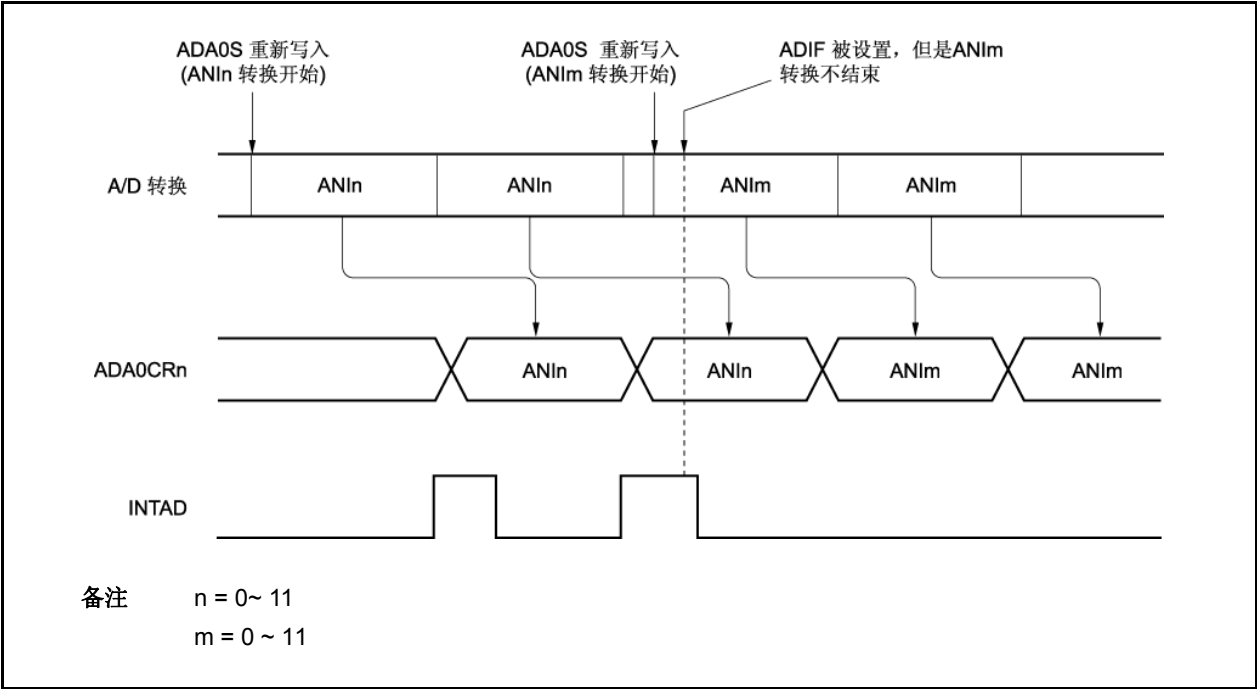
- (a) 如果引入数字脉冲的引脚和输入模拟信号进行 A/D 转换的引脚相邻（例如，在 **ANI3** 转换过程中的 **P72** 和 **P74**）（原因：耦合噪声的影响）
- (b) 如果在执行 A/D 转换过程中读取 **P7H** 或 **P7L** 寄存器到输入端口的指令，或者写入数据到输出端口的指令，导致 **AVREF0** 电源波动（原因：**AVREF0** 电源的影响）
- (c) 如果连接到端口引脚的外部电路的影响，电流在设置为输出模式的端口 7（**P70** 到 **P711**）的引脚中流动，导致 **AVREF0** 电源波动（原因：**AVREF0** 电源的影响）

如果在 A/D 转换过程中，以上任意处理可能会被执行，请确认执行多次 A/D 转换，检查 A/D 转换值并通过软件程序剔除异常值。

(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正好在寄存器 ADA0S 改写之前被置 1。如果寄存器 ADA0S 复写后立即读取标志 ADIF，即使新选择的模拟输入信号的转换还未完成，标志 ADIF 也可能会置 1。当 A/D 转换停止，在再次转换前对标志 ADIF 清零。

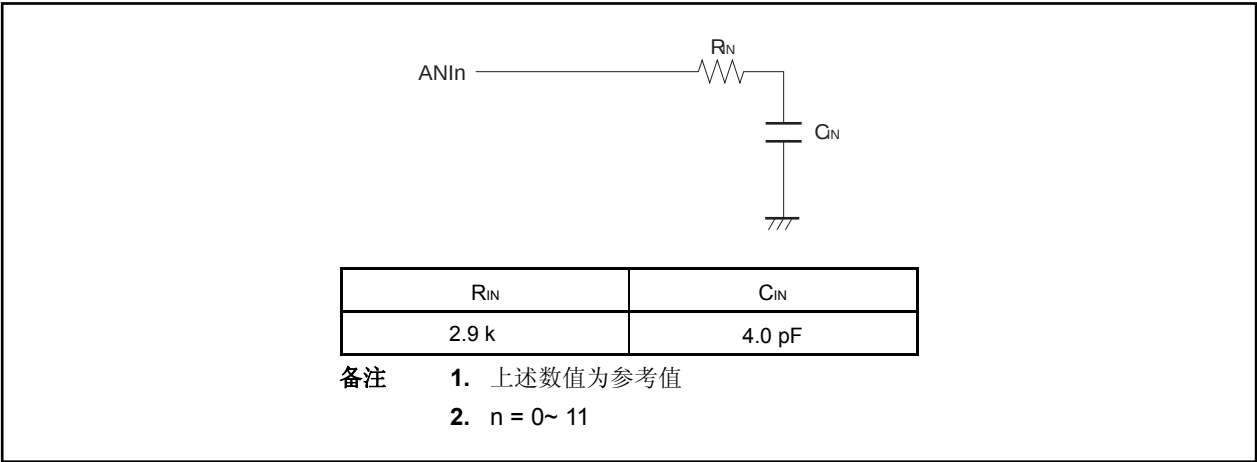
图 13-13. A/D 转换结束中断请求的产生时序



(6) 内部等效电路

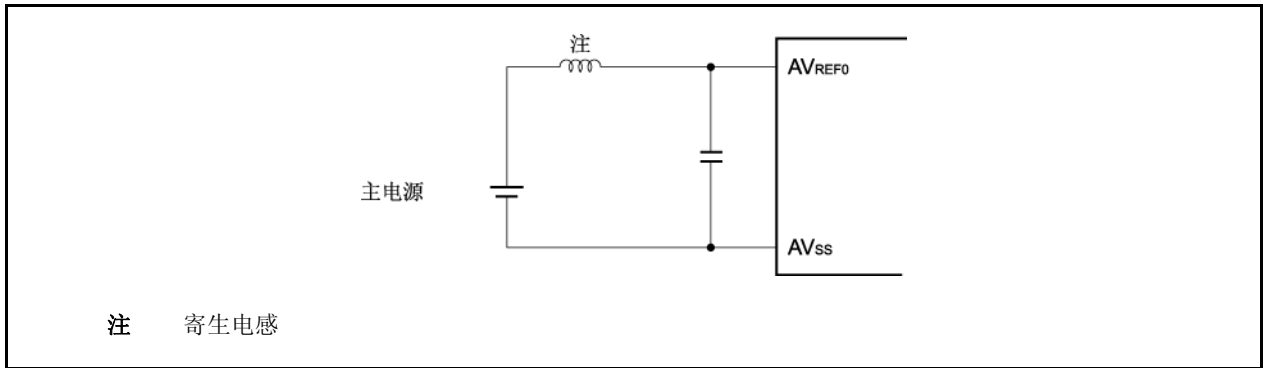
如下展示了模拟输入模块的等效电路。

图 13-14. 引脚 ANIn 的内部等效电路



(7) 引脚 AV_{REF0}

- (a) 引脚 AV_{REF0} 用于 A/D 转换器的供电电压引脚，同样为复用功能端口供电。当使用备用供电电源时，确保向引脚 AV_{REF0} 提供与 V_{DD} 相同的电压，如图 13-15 所示。
- (b) 引脚 AV_{REF0} 也用于 A/D 转换器的参考电压引脚。如果提供给引脚 AV_{REF0} 的供电电源具有高阻抗，或者如果供电电源电流负载能力低，参考电压可能受转换过程中的电流影响（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。为了避免上述情况，推荐在引脚 AV_{REF0} 和 AV_{SS} 之间连接一个电容，以抑制参考电压的波动，如图 13-15 所示。
- (c) 如果提供给引脚 AV_{REF0} 的供电电源具有高直流阻抗（例如，插入一个二极管），使能转换时的电压可能比停止转换时的电压低，因为 A/D 转换电流引起电压失真。

图 13-15. 引脚 AV_{REF0} 处理过程

<R>

(8) 读取寄存器 $ADA0CRn$

当执行寄存器 $ADA0M0 \sim ADA0M2$, $ADA0S$, $ADA0PFM$ 或 $ADA0PFT$ 的写指令时，寄存器的 $ADA0CRn$ 内容可能不确定。转换完成后，读取转换结果之前不要写入寄存器 $ADA0M0 \sim ADA0M2$, $ADA0S$, $ADA0PFM$ 和 $ADA0PFT$ 。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。应在转换完成之后且下一个外部/定时器触发信号收到之前来读取转换结果。在上述时序之外，可能无法读取正确转换结果。

(9) 待机模式

因为 A/D 转换器在 STOP 模式下停止工作，转换结果无效，所以可以降低功耗。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，进入 IDLE1, IDLE2 模式后的 A/D 转换结果无效。进入 IDLE1, IDLE2 模式前的转换结果有效。

<R> (10) 每种模式的限制

(a) 选择外部触发器模式/定时器触发器模式，必须设置高速转换模式。在 A/D 转换操作允许 (ADA0M0.ADA0CE 位 = 1) 之后会插入一个稳定时间，在此稳定期间内不能输入触发信号。

(b) 在下列模式下，在 A/D 转换停止时，将数据写入 A/D 控制寄存器 (ADA0CE 位 = 0)，然后使能 A/D 转换操作 (ADA0CE 位 = 1)。

- 普通转换模式
- 高速转换模式的单脉冲选择模式/单脉冲扫描模式

备注 A/D 控制寄存器: ADAM0, ADA0M2, ADA0S, ADA0PFM 和 ADA0PFT 寄存器。

(12) A/D 转换结果的偏差

A/D 转化的结果会受到供电电压的波动或干扰而变化。要减小偏差，必须采取软件抗干扰措施，例如对 A/D 转换结果进行平滑处理。

(13) A/D 转换结果滞后特性

逐次逼近型 A/D 转换器保持内部采样保持电容上的模拟输入电压，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的转换结果。
- 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的转换结果。

13.7 如何读解A/D转换器特征表

以下介绍 A/D 转换器中的相关术语。

(1) 分辨率

这是可被识别的最小的模拟输入电压，即模拟输入电压对一位数字输出的比率，称为 1 LSB（最低有效位）。对于满量程的 1 LSB 的比率用%FSR（满量程范围）表示。%FSR 是以百分比形式表示的一定范围的可转换模拟输入电压的比率。与分辨率无关，可如下表示。

$$\begin{aligned} 1\%FSR &= (\text{可转换模拟输入电压的最大值} - \text{可转换模拟输入电压的最小值}) / 100 \\ &= (AV_{REF0} - 0) / 100 \\ &= AV_{REF0} / 100 \end{aligned}$$

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。

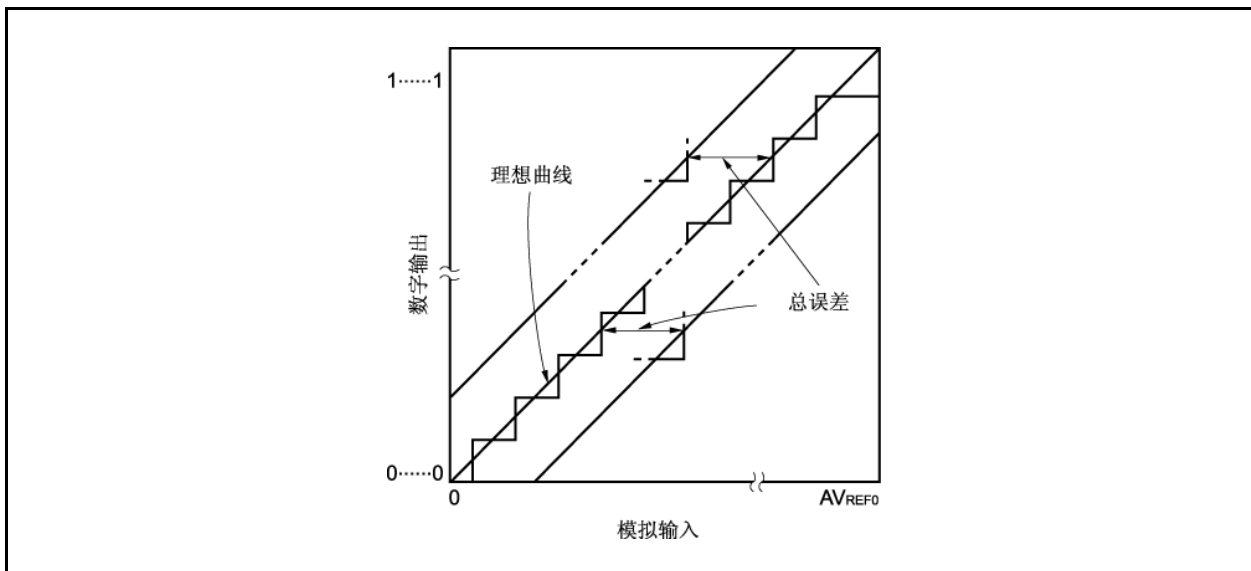
(2) 总误差

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、线性误差和这些误差组合起来表示总误差。

量化误差不属于特征表中总误差的范围。

图 13-16. 总误差

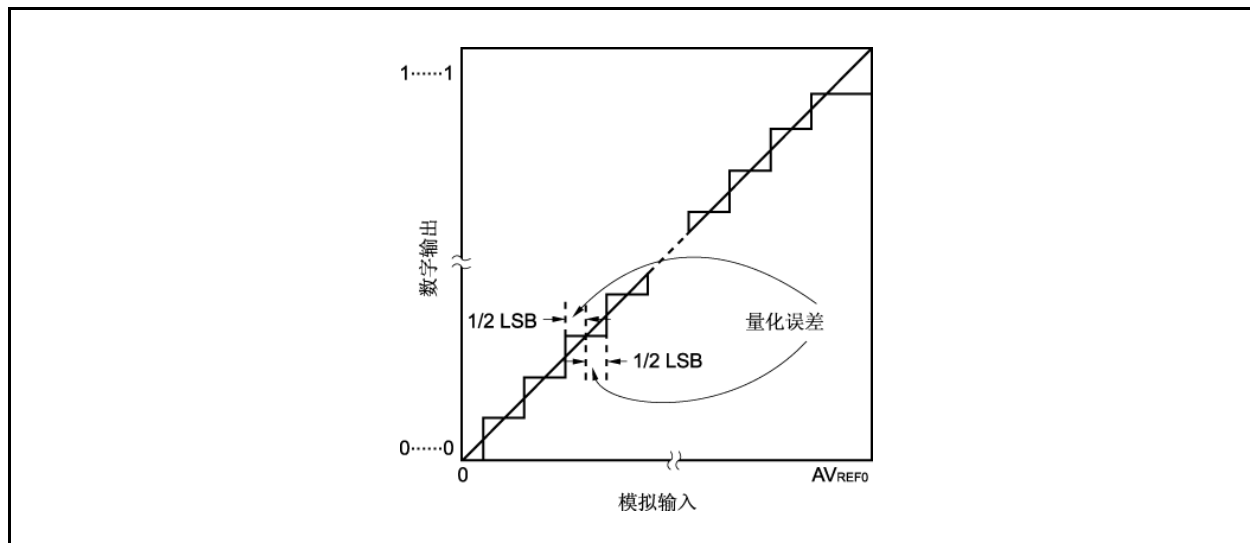


(3) 量化误差

当模拟值转换成数字值时，通常总会产生 $\pm 1/2 \text{ LSB}$ 的误差。因为在 A/D 转换器中，相差 $\pm 1/2 \text{ LSB}$ 范围内的模拟输入电压都被转换成相同的数字代码，因此量化误差不可避免。

量化误差不属于特征表中总误差、零度误差、满度误差、积分线性误差和微分线性误差的范围。

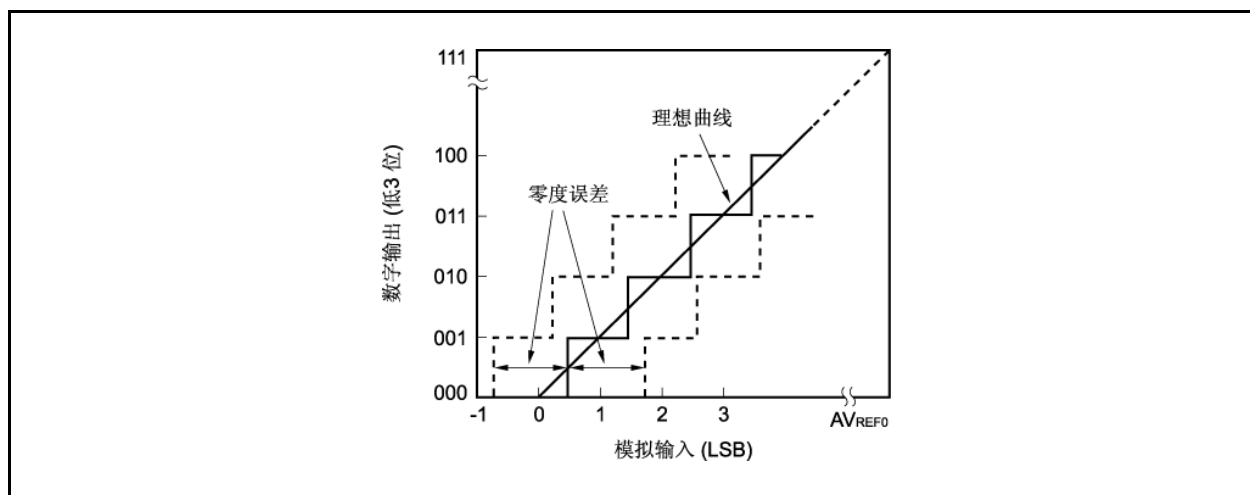
图 13-17. 量化误差



(4) 零度误差

零度误差表示当数字输出范围在 $0 \dots 000 \sim 0 \dots 001$ 之间时模拟输入电压的实际测量值与理论值 ($1/2 \text{ LSB}$) 之间的误差。

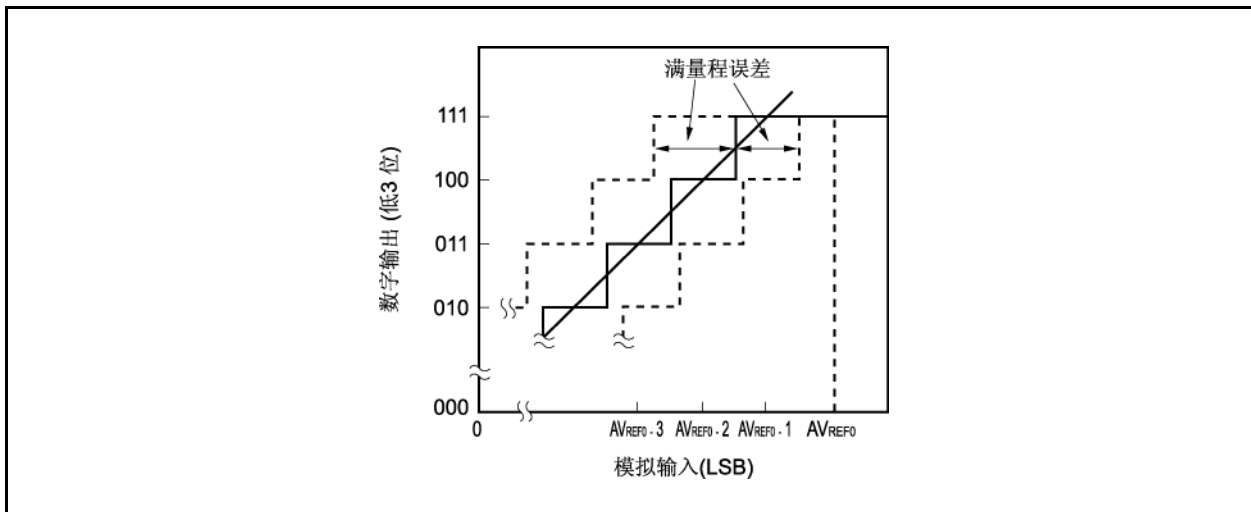
图 13-18. 零度误差



(5) 满量程误差

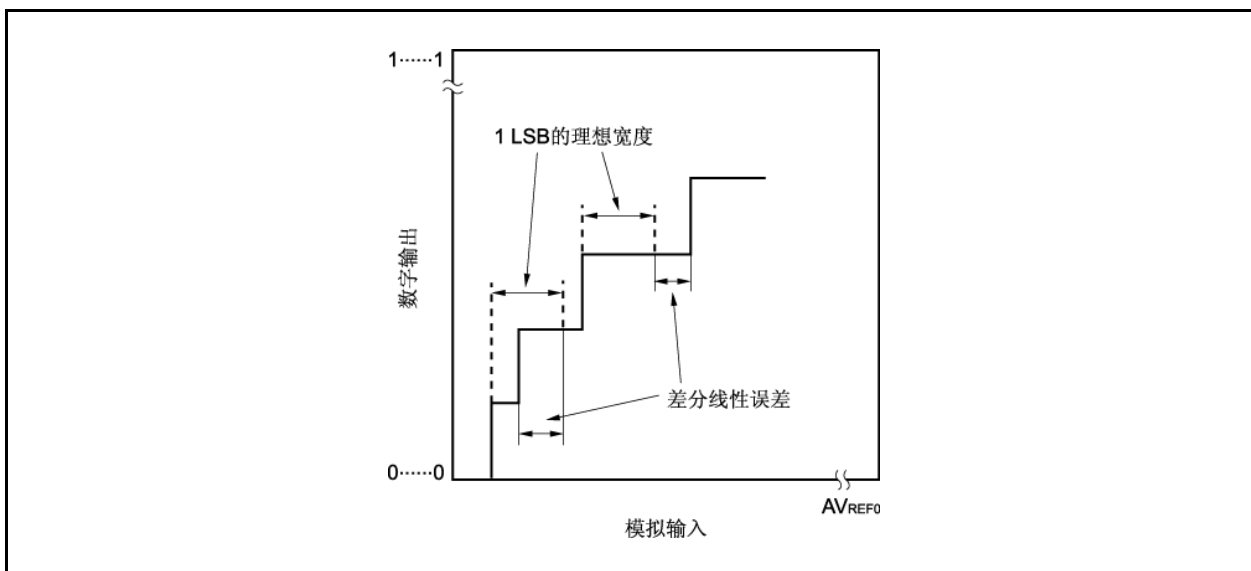
满量程误差表示当数字输出范围在 $1\ldots110 \sim 1\ldots111$ 之间时，模拟输入电压的实际测量值与理论值（满量程 $-3/2\text{LSB}$ ）之间的误差。

图 13-19. 满量程误差

**(6) 微分线性误差**

理论上，输出特定码的宽度为 1LSB 。当输出一个特定码时，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逐位从 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，或使用两个（包括两个以上）通道时，参见 13.7 (2) 总误差。

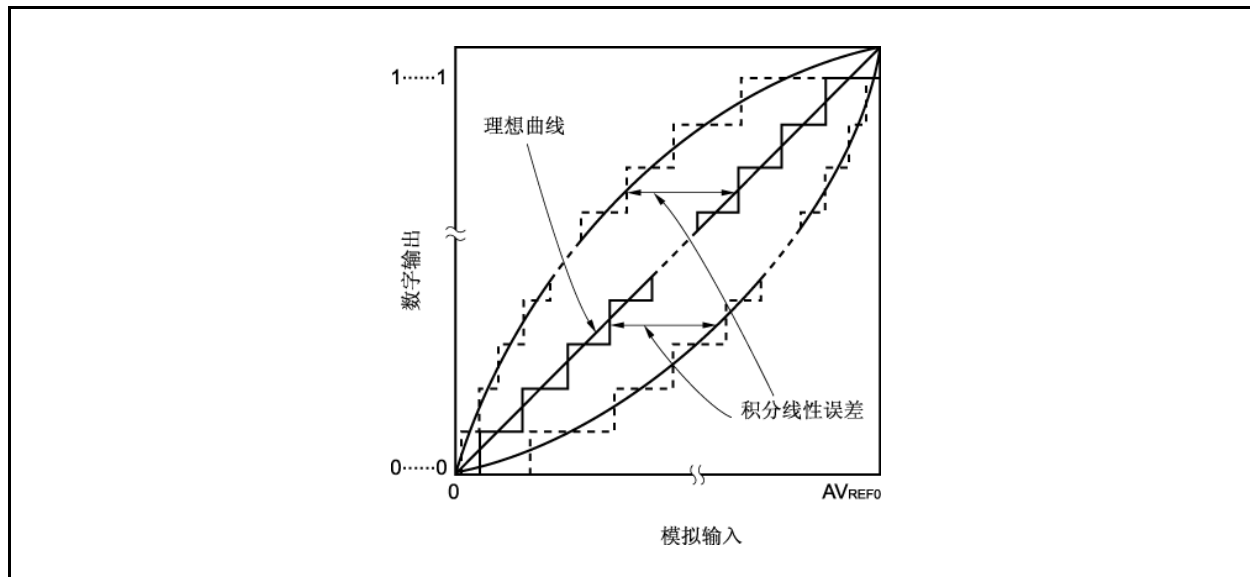
图 13-20. 微分线性误差



(7) 积分线性误差

积分线性误差说明了转换特征偏离理想线性关系的程度。它表示当零度误差和满度误差均为 0 时实际测量值与理想直线之间误差的最大值。

图 13-21. 积分线性误差

**(8) 转换时间**

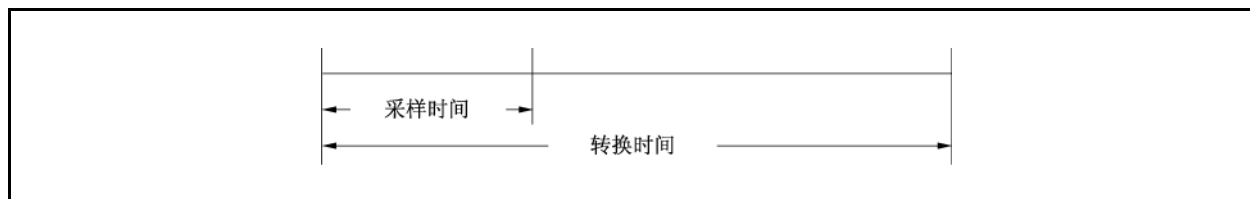
转换时间表示从产生触发到获取数字输出所经历的时间。

特征表中的转换时间包含采样时间。

(9) 采样时间

采样时间表示模拟开关打开，将模拟电压采样装载到采样&保持电路所需的时间。

图 13-22. 采样时间



第 14 章 D/A转换器

14.1 功能

D/A 转换器有以下功能。

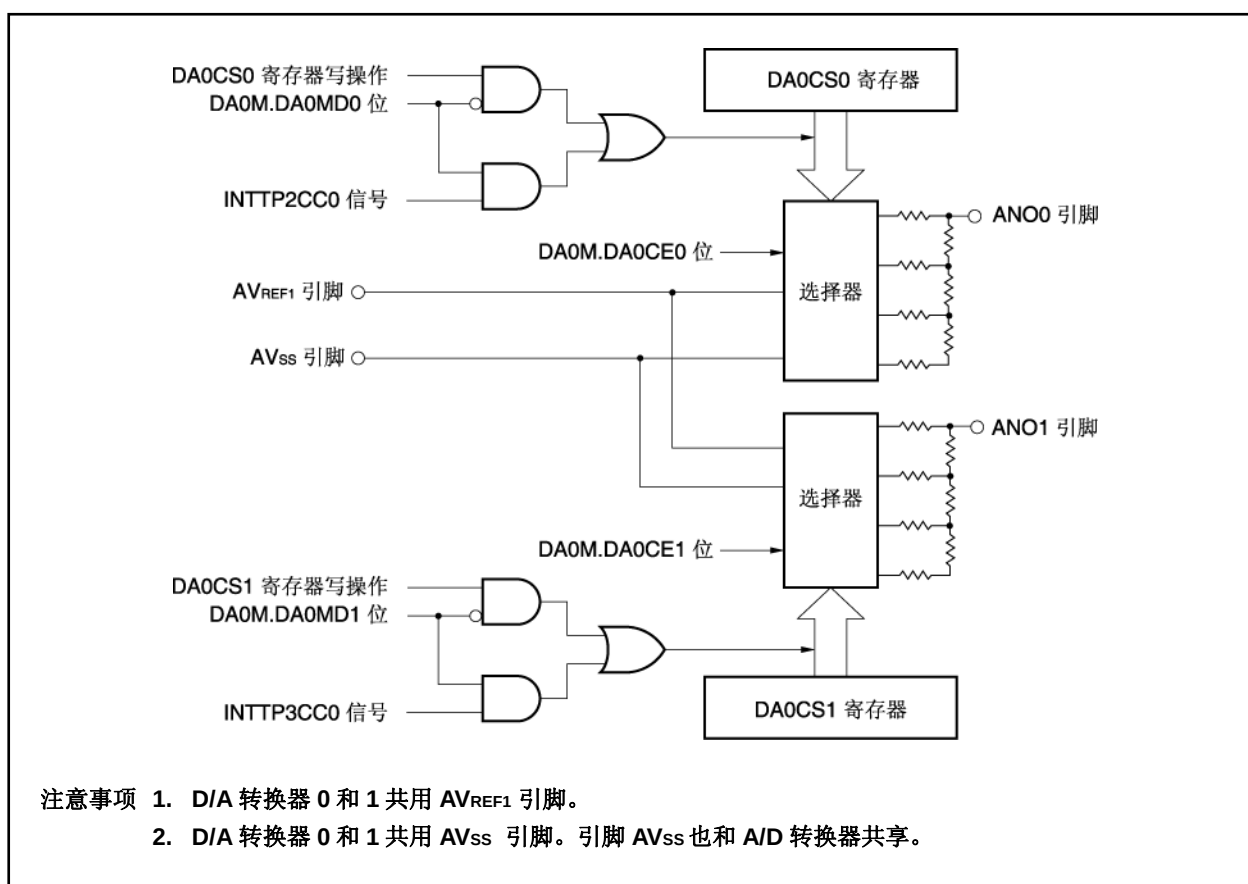
- 8 位分辨率 × 2 通道 (DA0CS0, DA0CS1)
- R-2R 梯形法
- 处理时间: 最长 $3\mu\text{s}$ (当 AV_{REF1} 为 $3.0 \sim 3.6\text{ V}$ 且外部负载为 20 pF)
- 模拟输出电压: $\text{AV}_{\text{REF1}} \times m/256$ ($m = 0$ 到 255 ; 由寄存器 DA0CSn 设置)
- 操作模式: 正常模式、实时输出模式

备注 $n = 0, 1$

14.2 配置

D/A 转换器的配置如下。

图 14-1. D/A 转换器框图



D/A 转换器包括以下硬件。

表 14-1. D/A 转换器的配置

项目	配置
控制寄存器	D/A 转换器模式寄存器 (DA0M) D/A 转换器转换值设置寄存器 0, 1 (DA0CS0, DA0CS1)

14.3 寄存器

控制 D/A 转换器的寄存器如下所述。

- D/A 转换器模式寄存器 (DA0M)
- D/A 转换器转换值设置寄存器 0, 1 (DA0CS0, DA0CS1)

(1) D/A 转换器模式寄存器 (DA0M)

该寄存器控制 D/A 转换器的运行。
可由 1 位或 8 位存储器操作指令读写该寄存器。
复位信号产生将该寄存器清零 (00H)。

复位后: 00H R/W 地址: FFFFF282H

	7	6	<5>	<4>	3	2	1	0
DA0M	0	0	DA0CE1	DA0CE0	0	0	DA0MD1	DA0MD0
DA0CEn	D/A 转换器操作使能/禁止的控制(n = 0, 1)							
0	禁止操作							
1	使能操作							
DA0MDn	D/A 转换器操作模式的选择(n = 0, 1)							
0	正常模式							
1	实时输出模式 ^注							

- 注 实时输出模式 (DA0MDn = 1) 下，输出触发如下所述。
- 当 n = 0: INTTP2CC0 信号 (参见 第 7 章 16 位定时器/事件计数器 P (TMP))
 - 当 n = 1: INTTP3CC0 信号 (参见 第 7 章 16 位定时器/事件计数器 P (TMP))

(2) D/A 转换器值设置寄存器 0, 1 (DA0CS0, DA0CS1)

寄存器 DA0CS0 和 DA0CS1 设置引脚 ANO0 ~ ANO1 的模拟电压输出值。

可由 1 位或 8 位存储器操作指令读写该寄存器。

复位信号产生将该寄存器清零 (00H)。

复位后: 00H R/W 地址: DA0CS0 FFFFF280H, DA0CS1 FFFFF281H

	7	6	5	4	3	2	1	0
DA0CSn	DA0CSn7	DA0CSn6	DA0CSn5	DA0CSn4	DA0CSn3	DA0CSn2	DA0CSn1	DA0CSn0

注意事项 实时输出模式 (DA0MDn = 1) 下, 在信号 INTTP2CC0/INTTP3CC0 产生之前, 设置寄存器 DA0CSn。当信号 INTTP2CC0/INTTP3CC0 产生时, 开始 D/A 转换。

备注 n = 0, 1

14.4 操作

14.4.1 正常模式下的操作

对寄存器 DA0CSn 进行写操作触发 D/A 转换。

设置方法如下所述。

- <1> 设置 DA0M.DA0MDn = 0（正常模式）。
- <2> 设置寄存器 DA0CSn 的模拟电压值，该值将输出到引脚 ANOn。
步骤<1> 和 <2> 完成初始设置
- <3> 设置 DA0M.DA0CEn = 1（D/A 转换使能）。
设置完成后，开始 D/A 转换。
- <4> 对寄存器 DA0CSn 进行写操作，执行下一次 D/A 转换。
上次的 D/A 转换结果会保持到下次 D/A 转换开始。

- 备注**
- 1. 关于引脚复用功能设置，参见表 4-15 端口引脚作为复用功能引脚。
 - 2. n = 0, 1

14.4.2 实时输出模式下的操作

TMP2 和 TMP3 的中断请求信号（INTTP2CC0 和 INTTP3CC0）触发 D/A 转换。

设置方法如下所述。

- <1> 设置 DA0M.DA0MDn = 1（实时输出模式）。
- <2> 设置寄存器 DA0CSn 的模拟电压值，该值将输出到引脚 ANOn。
- <3> 设置 DA0M.DA0CEn = 1（D/A 转换使能）。
步骤<1> 和 <3> 组成初始设置。
- <4> 操作 TMP2 和 TMP3。
- <5> 当产生信号 INTTP2CC0 和 INTTP3CC0 时，D/A 转换开始。
- <6> 之后，每次产生信号 INTTP2CC0 和 INTTP3CC0，寄存器 DA0CSn 的值被输出。

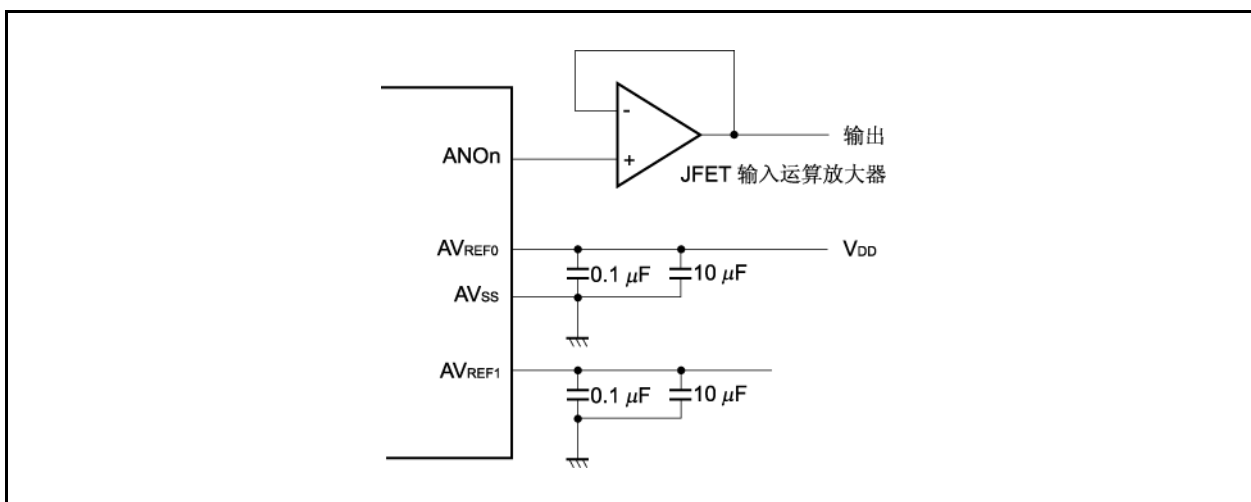
- 备注**
- 1. 引脚 ANOn 和 ANO1 的输出值在步骤<5>之前是不确定的。
 - 2. HALT, IDLE1, IDLE2 和 STOP 模式下的 ANOn 和 ANO1 引脚的输出值，可参见第 24 章 待机功能。
 - 3. 关于引脚复用功能设置，参见表 4-15 端口引脚作为复用功能引脚。

14.4.3 注意事项

当使用 V850ES/SG2 的 D/A 转换器时，注意下述注意事项。

- (1) 实时输出模式下，当触发信号发出后，不要改变寄存器 DA0CSn 的设置值。
- (2) 在改变操作模式前，确保 DA0M.DA0CEn = 0。
- (3) 当引脚 P10/AN00 和 P11/AN01 其中之一被用于 I/O 端口，而另一个用于 D/A 输出引脚时，在 D/A 输出期间不要改变 I/O 端口电平。
- (4) 确保 $AV_{REF0} = V_{DD} = AV_{REF1} = 3.0 \sim 3.6 \text{ V}$ 。如果超出此范围之外，操作不能保证。
- (5) AV_{REF1} 的电压与 AV_{REF0} 的电压一致。
- (6) 引脚 ANOn (n = 0, 1) 不能输出电流，因为 D/A 转换器的输出阻抗很高。当连接一个小于等于 $2 \text{ M}\Omega$ 的电阻时，在电阻和引脚 ANOn 之间插入一个 JFET 输入运算放大器。

图 14-2. 引脚连接示例



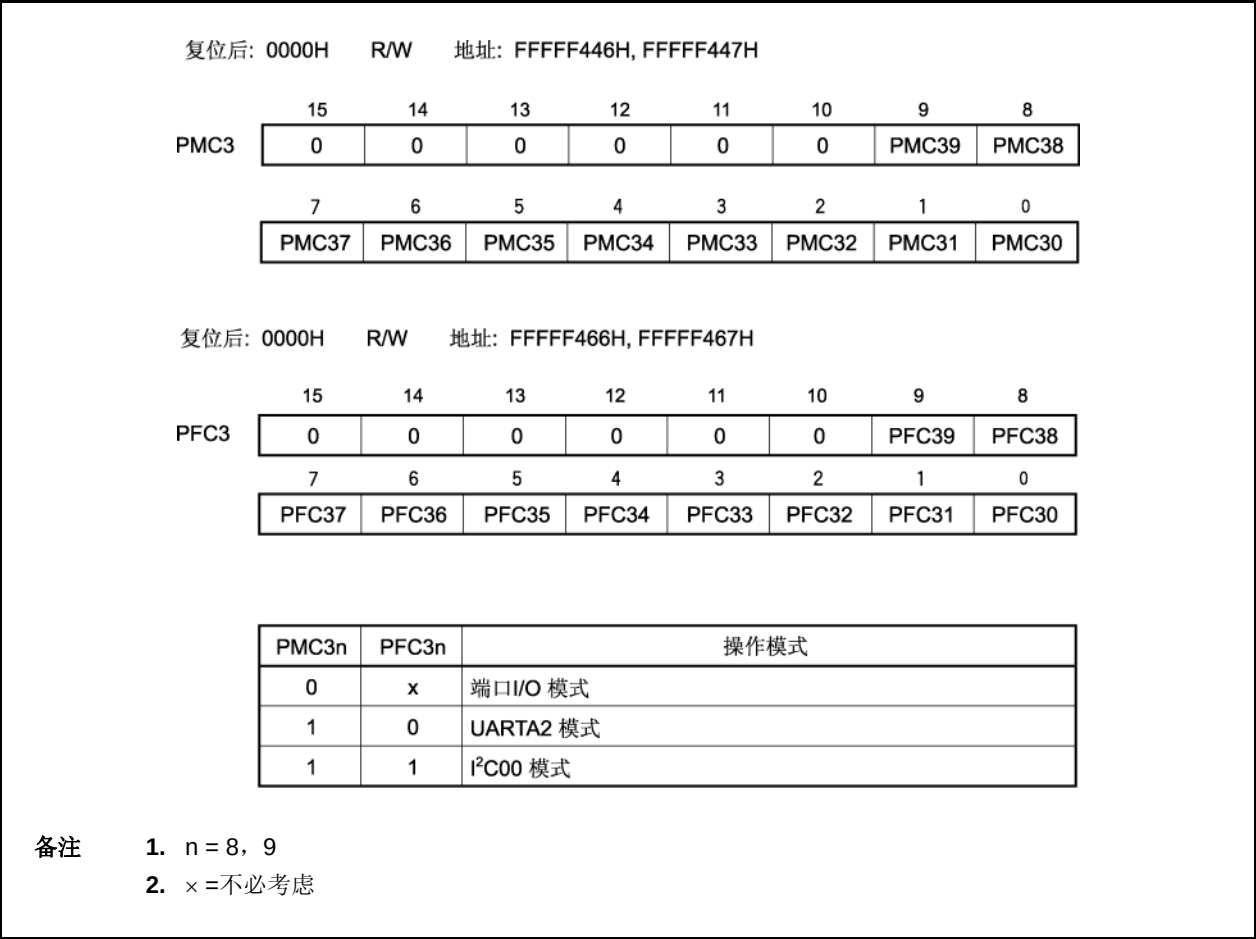
- (7) 由于 STOP 模式下，D/A 转换器停止操作，引脚 ANO0 和 ANO1 引入高阻抗状态，功耗可降低。IDLE1, IDLE2 或副时钟模式下，操作继续。因此，为了降低功耗，应该使 DA0M.DA0CEn = 0。

15.1.2 UARTA2 和 I²C00 模式转换

<R> 在 V850ES/SG2 和 V850ES/SG2-H 的 I²C 总线版本（Y 产品）中，UARTA2 和 I²C00 复用相同引脚，因此不能同时使用。在使用前，先用寄存器 PMC3 和 PFC3 设置 UARTA2。

注意事项 如果在 UARTA2 和 I²C00 的发送/接收操作过程中，进行切换，这两个功能不能保证。确保不被使用的那个处于非使能态。

图 15-2. UARTA2 和 I²C00 的模式转换设置

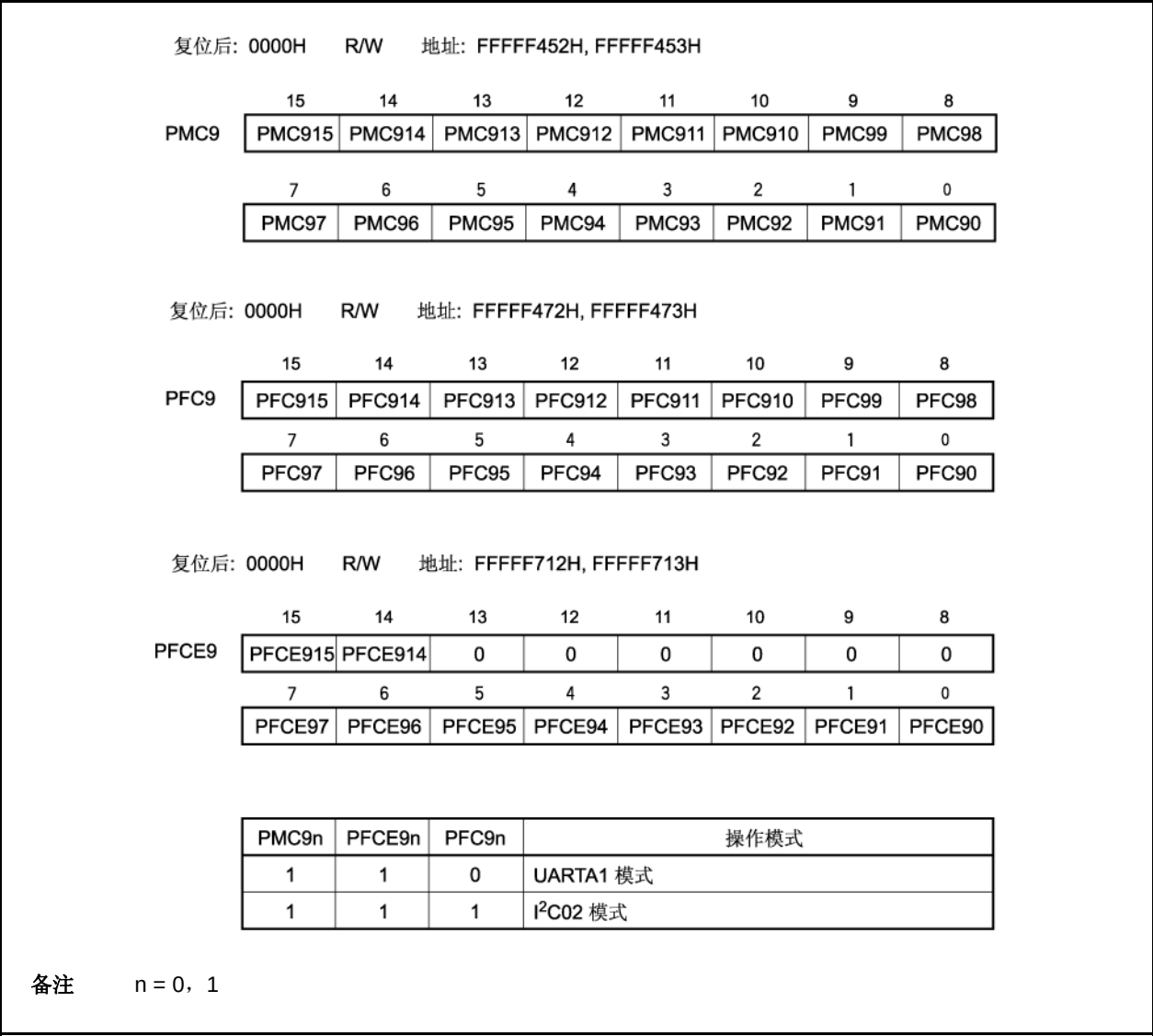


15.1.3 UARTA1 和 I²C02 模式转换

<R> 在 V850ES/SG2 和 V850ES/SG2-H 的 I²C 总线版本（Y 产品）中，UARTA1 和 I²C02 复用相同引脚，因此不能同时使用。在使用前，先用寄存器 PMC9，PFC9 和 PFCE9 设置 UARTA1。

注意事项 如果在 UARTA1 和 I2C02 的发送/接收操作过程中，进行切换，这两个功能不能保证。确保不被使用的那个处于非使能态。

图 15-3. UARTA1 和 I²C02 模式转换设置



15.2 特性

<R>

- 传输速率： 300 bps 到 312.5 kbps
(V850ES/SG2: 使用 20MHz 的内部系统时钟,
V850ES/SG2-H: 使用 32MHz 的内部系统时钟和专用的波特率产生器)
- 全双工通信: 内部 UARTAn 数据接收寄存器 (UAnRX)
内部 UARTAn 数据发送寄存器 (UAnTX)
- 2 引脚设置: TXDAn: 数据发送输出引脚
RXDAn: 数据接收输入引脚
- 接收错误输出功能
 - 校验错误
 - 帧错误
 - 溢出错误
- 中断源: 2
 - 接收完成中断 (INTUAnR): 该中断产生于接收使能状态下, 接收到的数据从接收移位寄存器到 UAnRX 寄存器转换完成后。
 - 发送使能中断 (INTUAnT): 该中断产生于发送使能状态下, 发送数据从 UAnTX 寄存器到发送移位寄存器的转换过程中。
- 通信数据的宽度: 7, 8 位
- 校验功能: 奇校验, 偶校验, 零校验, 无校验
- 传输停止位: 1, 2 位
- 片上专用波特率发生器
- 可选 MSB-/LSB-first 通信
- 发送/接收数据反向输入/输出操作
- LIN (局部互联网) 通信模式下 SBF (同步中断区域) 的发送/接收
 - 同步中断区域发送长度为 13 ~ 20
 - 识别 11 位或更多位的 SBF 接收
 - 提供 SBF 接收标志

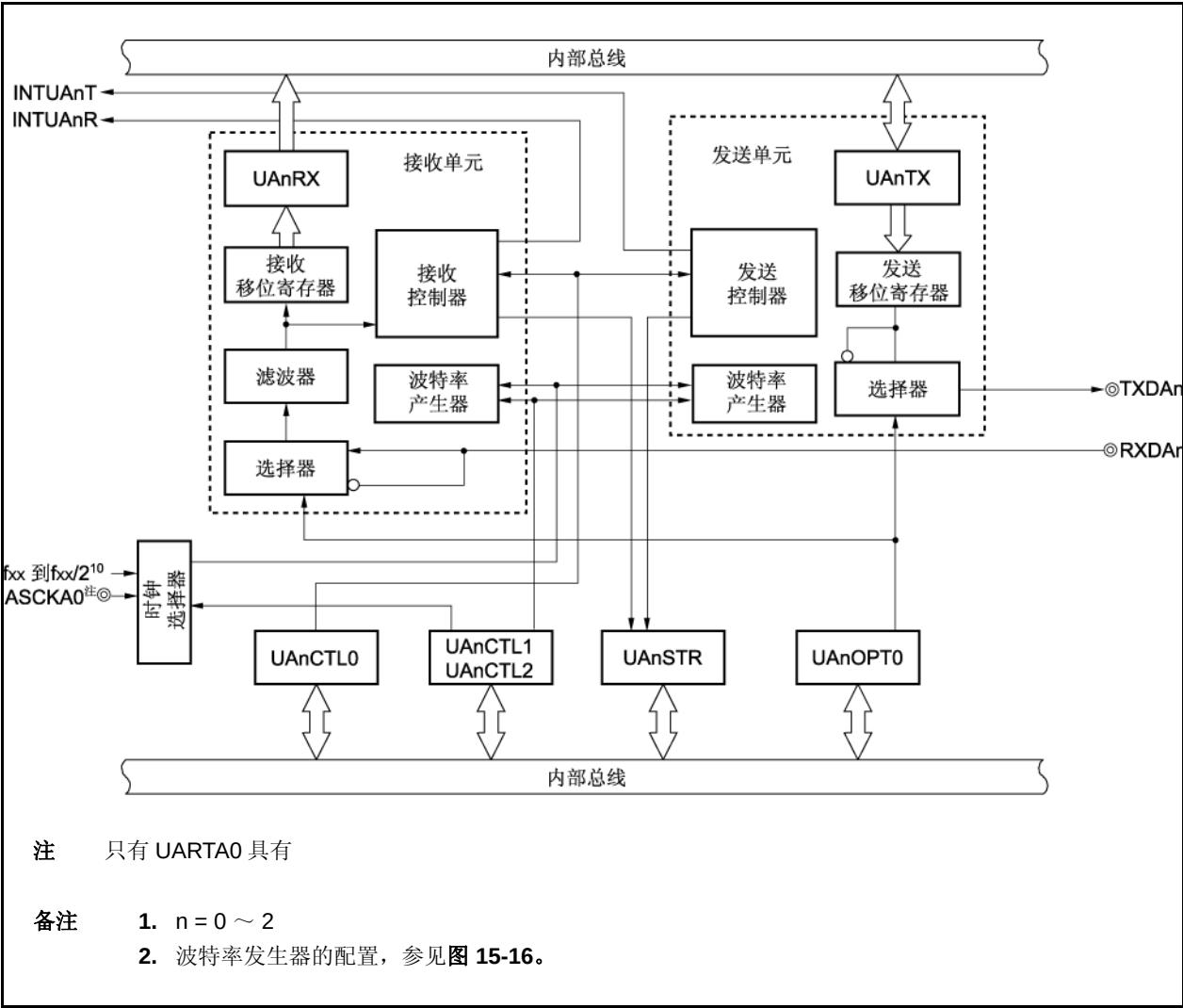
备注 n = 0 ~ 2

15.3 配置

如下为 UARTAn 的框图。

<R>

图 15-4. 异步串行接口 An 的框图



UARTAn 由以下寄存器控制。

表 15-1. UARTAn 的配置

项目	配置
寄存器	UARTAn 控制寄存器 0（UAnCTL0） UARTAn 控制寄存器 1（UAnCTL1） UARTAn 控制寄存器 2（UAnCTL2） UARTAn 选择控制寄存器 0（UAnOPT0） UARTAn 状态寄存器（UAnSTR） UARTAn 接收移位寄存器 UARTAn 数据接收寄存器（UAnRX） UARTAn 发送移位寄存器 UARTAn 数据发送寄存器（UAnTX）

(1) UARTAn 控制寄存器 0 (UAnCTL0)

该寄存器是 8 位寄存器，用于设定 UARTAn 的操作。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

该寄存器是 8 位寄存器，用于选择 UARTAn 的输入时钟。

(3) UARTAn 控制寄存器 2 (UAnCTL2)

该寄存器是 8 位寄存器，用于控制 UARTAn 的波特率。

(4) UARTAn 选择控制寄存器 0 (UAnOPT0)

该寄存器是 8 位寄存器，用于控制 UARTAn 的串行传输。

(5) UARTAn 状态寄存器 (UAnSTR)

该寄存器包含发生接收错误时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 同时产生接收完成中断请求信号 (INTUAnR)。

(8) UARTAn 发送移位寄存器

该寄存器是移位寄存器，用于将寄存器 UAnTX 传出的并行数据转换成串行数据。

当 1 字节数据从寄存器 UAnTX 传送，移位寄存器数据从引脚 TXDAn 输出。

该寄存器不能直接操作。

(9) UARTAn 数据发送寄存器 (UAnTX)

该寄存器是 8 位数据传送缓冲。当要传送的数据写入寄存器 UAnTX，传送开始。当数据可以被写入寄存器 UAnTX（当 1 帧的数据从寄存器 UAnTX 传送到发送移位寄存器 UARTAn），产生发送使能中断请求信号 (INTUAnT)。

15.4 寄存器

(1) UARTAn 控制寄存器 0 (UAnCTL0)

该寄存器是 8 位寄存器，用于控制 UARTAn 的串行发送操作。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位将该寄存器设置为 10H。

(1/2)

复位后: 10H

R/W

地址:

UA0CTL0 FFFFFFFA00H, UA1CTL0 FFFFFFFA10H, UA2CTL0 FFFFFFFA20H

<7>

<6>

<5>

<4>

3

2

1

0

UAnCTL0

UAnPWR

UAnTXE

UAnRXE

UAnDIR

UAnPS1

UAnPS0

UAnCL

UAnSL

(n = 0 到2)

UAnPWR

UARTAn 操作控制

0

禁止UARTAn 操作(UARTAn 异步复位)

1

使能UARTAn 操作

由UAnPWR 位控制UARTAn 的操作。通过清除UAnPWR 位为0, TXDAn 管脚固定为高电平（如果UAnOPT0.UAnTDL 位= 1, 固定为低电平）。

UAnTXE

发送操作使能

0

禁止发送操作

1

使能发送操作

· 要开始发送, 设置UAnPWR 位为1, 然后设置 UAnTXE 位为 1。

· 要初始化发送单元, 清除UAnTXE 位为 0, 等待基准时钟 (fCLK)两个周期, 然后再设置UAnTXE 位为1。否则, 初始化可能不被执行(关于基准时钟, 见 15.7 (1) (a) 基准时钟)。

• 当操作被使能(UAnPWR bit = 1)时, 从UAnTXE = 1开始两个或更多个基准时钟(fCLK)过去后, 发送操作被使能。

• 当UAnPWR 位被清除为0时, 即使UAnTXE 位为1, 通过UAnPWR 位内部电路的状态与UAnTXE 位= 0的状态相同。当UAnPWR 位被再次设置为1时, 发送操作被使能。

<R>

<R>

<R>

UAnRXE	接收操作使能
0	禁止接收操作
1	使能接收操作

• 要开始接收, 设置UAnPWR 位为1, 然后设置 UAnRXE 位为 1。
 • 要初始化接收单元, 清除UAnRXE 位为 0, 等待基准时钟两个周期, 然后再次设置UAnRXE 位为1。否则, 初始化可能不被执行(关于基准时钟, 见 15.7 (1) (a) 基准时钟)。
 • 当操作被使能(UAnPWR bit = 1)时, 从UAnRXE = 1开始两个或更多个基准时钟(f_{CLK})过去后, 接收操作被使能。如果在接收操作被使能前起始位被接收, 起始位被忽略。
 • 当UAnPWR 位被清除为0时, 即使UAnRXE 位为1, 通过UAnPWR 位内部电路的状态与UAnRXE 位= 0的状态相同。当UAnPWR 位被再次设置为1时, 接收操作被使能。

UAnDIR ^注	发送方向选择
0	MSB首先发送
1	LSB首先发送

UAnPS1 ^注	UAnPS0 ^注	发送期间的奇偶选择	接收期间的奇偶选择
0	0	无奇偶输出	无奇偶接收
0	1	0 奇偶输出	0奇偶接收
1	0	奇校验输出	奇校验检查
1	1	偶校验输出	偶校验检查

• 如果 “0奇偶接收”在接收期间被选择, 奇偶检查不被执行。因此 UAnSTR.UAnPE位不被设置。
 • 当发送和接收在LIN格式下被执行时, 清除UAnPS1 和UAnPS0 位为00。

UAnCL ^注	发送/接收一帧数据的数据长度的规范
0	7 位
1	8 位

UAnSL ^注	发送数据的停止位长度的规范
0	1 位
1	2 位

只有接收数据停止位的第一位被检查, 忽略UAnSL 位的值。

注 只有当 UAnPWR 位 = 0 或者 UAnTXE 位 = UAnRXE 位 = 0 时, 这个寄存器可以被重新写入。然而, 可以同时设置任意或所有 UAnPWR、UAnTXE 和 UAnRXE 位为 1。

备注 有关校验的详细内容, 参见 15.6.9 校验形式和操作

(2) UARTAn 控制寄存器 1 (UAnCTL1)

详细情况，参见 15.7 (2) UARTAn 控制寄存器 1 (UAnCTL1)。

(3) UARTAn 控制寄存器 2 (UAnCTL2)

详细情况，参见 15.7 (3) UARTAn 控制寄存器 2 (UAnCTL2)。

(4) UARTAn 选择控制寄存器 0 (UAnOPT0)

该寄存器是 8 位寄存器，用于控制 UARTAn 的串行传输。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号产生将该寄存器设置为 14H。

注意事项 在 SBF 接收期间 (UAnSRF=1), 不要将 UAnSRT 和 UAnSTT 置 1

(1/2)

复位后: 14H R/W 地址: UA0OPT0 FFFFA03H, UA1OPT0 FFFFA13H,
UA2OPT0 FFFFA23H

	<7>	6	5	4	3	2	1	0
UAnOPT0	UAnSRF	UAnSRT	UAnSTT	UAnSLS2	UAnSLS1	UAnSLS0	UAnTDL	UAnRDL

(n = 0 到2)

UAnSRF	SBF 接收标志
0	当UAnCTL0.UAnPWR 位= UAnCTL0.UAnRXE 位= 0 被设置时。 同样在SBF接收正常结束时。
1	SBF 接收期间
- 在LIN通信过程中，SBF (同步中断区)接收被判断。 - 当SBF接收错误发生时，UAnSRF位被保持为1，然后SBF 接收再次启动。	

UAnSRT	SBF 接收触发
0	-
1	SBF 接收触发
- 这是LIN通信期间的SBF接收触发位，当读取时，“0”总是被读出。关于SBF接收，设置UAnSRT 位(为 1)来使能SBF 接收。 - 在设置UAnPWR 位和 UAnRXE位为1后，设置UAnSRT 位。 - 在接收结束中断请求信号(INTUAnR)被产生后的1位期间，设置UAnSRT 位(为1)。(如果在结束操作期间，这一位被设置 (为1)，即使SBF没有被接收，当前数据的接收完成时，UAnSRF位被清除。) - 写入0到UAnSRT 位有效。如果在SBF接收开始前，0被写入UAnSRT 位，SBF不被接收，但是正常UART 接收被执行。 - 如果在SBF接收期间0被写入UAnOPT0 寄存器，已经被接收的数据被作为SBF接收。然而，如果正在接收的数据不是SBF，从下一个接收数据开始，后面的数据作为UART的接收数据。 - 当0被写入UAnSRT 位时，UAnSRF 位被清除。	

<R>

<R>

<R>

UAnSTT	SBF 发送触发
0	-
1	SBF 发送触发
• 这是LIN通信期间的SBF发送触发位，并且当被读取时，“0”总是被读出。 • 在设置UAnPWR 位= UAnTXE 位= 1后，设置UAnSTT 位。 • 写入0到UAnSTT 位有效。如果在1被写入它后并且它被基准时钟采样前，0被写入到这一位，SBF发送不被执行。如果在SBF发送期间，0被写入UAnSTT 位，即使SBF发送被执行，UAnSTR.UAnTSF 位也被清零。	

UAnSLS2	UAnSLS1	UAnSLS0	SBF 发送长度选择
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
当UAnPWR 位= 0或者当UAnTXE 位= 0时，这个寄存器可以被设置。			

UAnTDL	发送数据电平位
0	发送数据的正常输出
1	发送数据的反转输出
• 使用UAnTDL 位，TXDAn 管脚的输出电平可以被反转。 • 当UAnPWR 位= 0或者当UAnTXE 位= 0时，这个寄存器可以被设置。	

UAnRDL	接收数据电平位
0	发送数据的正常输入
1	发送数据的反转输入
• 使用UAnRDL 位，RXDAn 管脚的输入电平可以被反转。 • 当UAnPWR 位= 0或者当UAnRXE 位= 0时，这个寄存器可以被设置。 • 当UAnRDL 位被设置为1 (接收数据的反转输入)时，在接收开始时设置数据接收管脚到UART接收管脚(RXDAn)后，接收必须被使能(UAnCTL0.UAnRXE 位= 1)。当接收被使能后管脚模式更改时，如果管脚电平为高，起始位将被错误检测。	

<R>

(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
寄存器 UAnSTR	• 复位 • UAnCTL0.UAnPWR = 0
UAnTSF 位	• UAnCTL0.UAnTXE = 0
UAnPE，UAnFE，UAnOVE 位	• 写入 0 • UAnCTL0.UAnRXE = 0

<R> 注意事项 确保读取 UAnPE、UAnFE 和 UAnOVE 位的错误标志来检查标志状态，然后通过向它们写入“0”来清除标志。

复位后: 00H R/W 地址: UA0STR FFFFFFFA04H, UA1STR FFFFFFFA14H,
UA2STR FFFFFFFA24H

	<7>	6	5	4	3	<2>	<1>	<0>
UAnSTR (n = 0 到2)	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

UAnTSF	发送状态标志
0	- 当 UAnPWR 位= 0 或者UAnTXE 位= 0 已经被设置时。 - 当发送接收后没有下一个数据从UAnTX 寄存器发送
1	写入UAnTX 寄存器
当执行连续发送时，UAnTSF 位总是1。当初始化发送单元时，执行初始化前检查UAnTSF 位= 0。当UAnTSF 位= 1期间执行初始化时，发送数据不被保证。	

UAnPE	奇偶错误标志
0	- 当 UAnPWR 位= 0 或者UAnRXE 位= 0 已经被设置时。 - 当 0 已经被写入时
1	当接收期间数据的奇偶校验和奇偶位不匹配时。
- UAnPE 位的操作被UAnCTL0.UAnPS1 和UAnCTL0.UAnPS0位的设置控制。 - UAnPE 位可以被读写，但是它只能通过写入0被清除，不能通过写入1来置位。当1被写入这一位时，原值被保持。	

UAnFE	帧错误标志
0	- 当 UAnPWR 位= 0 或者UAnRXE 位= 0 已经被设置时。 - 当 0 已经被写入时
1	当接收期间停止位没有被检测时
- 只有接收数据停止位的第一位被检查，不考虑UAnCTL0.UAnSL位的值。 - UAnFE 位可以被读写，但是它只能通过写入0被清除，不能通过写入1来置位。当1被写入这一位时，原值被保持。	

UAnOVE	溢出错误标志
0	- 当 UAnPWR 位= 0 或者UAnRXE 位= 0 已经被设置时。 - 当 0 已经被写入时
1	当接收数据已经被设置到UAnRX寄存器并且在接收数据被读取前，下一个接收操作完成。
- 当一个溢出错误发生时，数据被忽略，下一个接收数据不被写入UAnRX 寄存器。 - UAnOVE 位可以被读写，但是它只能通过写入0被清除，不能通过写入1来置位。当1被写入这一位时，原值被保持。	

(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。

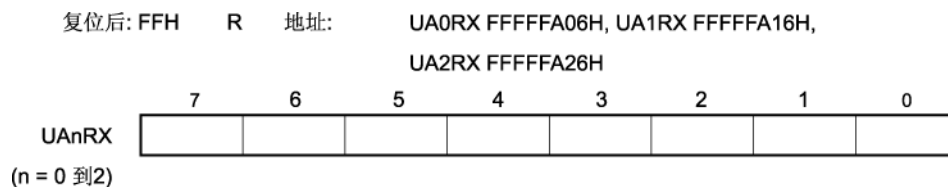
完整接收 1 字节数据后，存储于接收移位寄存器的数据被传送到寄存器 UAnRX。

这时，接收结束中断请求信号（INTUAnR）被产生。LSB-first 接收期间，当数据长度设置为 7 位，接收到的数据置于寄存器 UAnRX 的第 0~6 位，第 7 位恒为 0。MSB-first 接收期间，接收到的数据置于寄存器 UAnRX 的第 1~7 位，第 0 位恒为 0。

当产生溢出错误（UAnSTR.UAnOVE 位 = 1），此时接收到的数据不被传送到寄存器 UAnRX 并丢弃。

该寄存器是只读的，可由 8 位存储器操作指令设置该寄存器。

除了复位输入，通过 UAnCTL0.UAnPWR 置 0，寄存器 UAnRX 可置 FFH。



(7) UARTn 数据发送寄存器 (UAnTX)

该寄存器是 8 位寄存器，用于设置发送数据。

在发送使能状态 (UAnCTL0.UAnTXE 位 = 1) 下, 当发送数据被写入 UAnTX 寄存器时, 发送开始。当 UAnTX 寄存器的数据被传送到发送移位寄存器时, 发送使能中断请求信号 (INTUAnT) 被产生。

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。



15.5 中断请求信号

由 UARTAn 产生下面两个中断请求信号。

- 接收完成中断请求信号 (INTUAnR)
- 发送允许中断请求信号 (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 15-2. 中断及其默认优先级

中断	优先级
接收完成	高
发送使能	低

(1) 接收完成中断请求信号 (INTUAnR)

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。

当接收完成中断请求信号被接收，并且数据被读取，读取寄存器UAnSTR，判断是否产生接收错误。

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。

15.6 操作

15.6.1 数据格式

执行全双工串行数据的接收和发送。

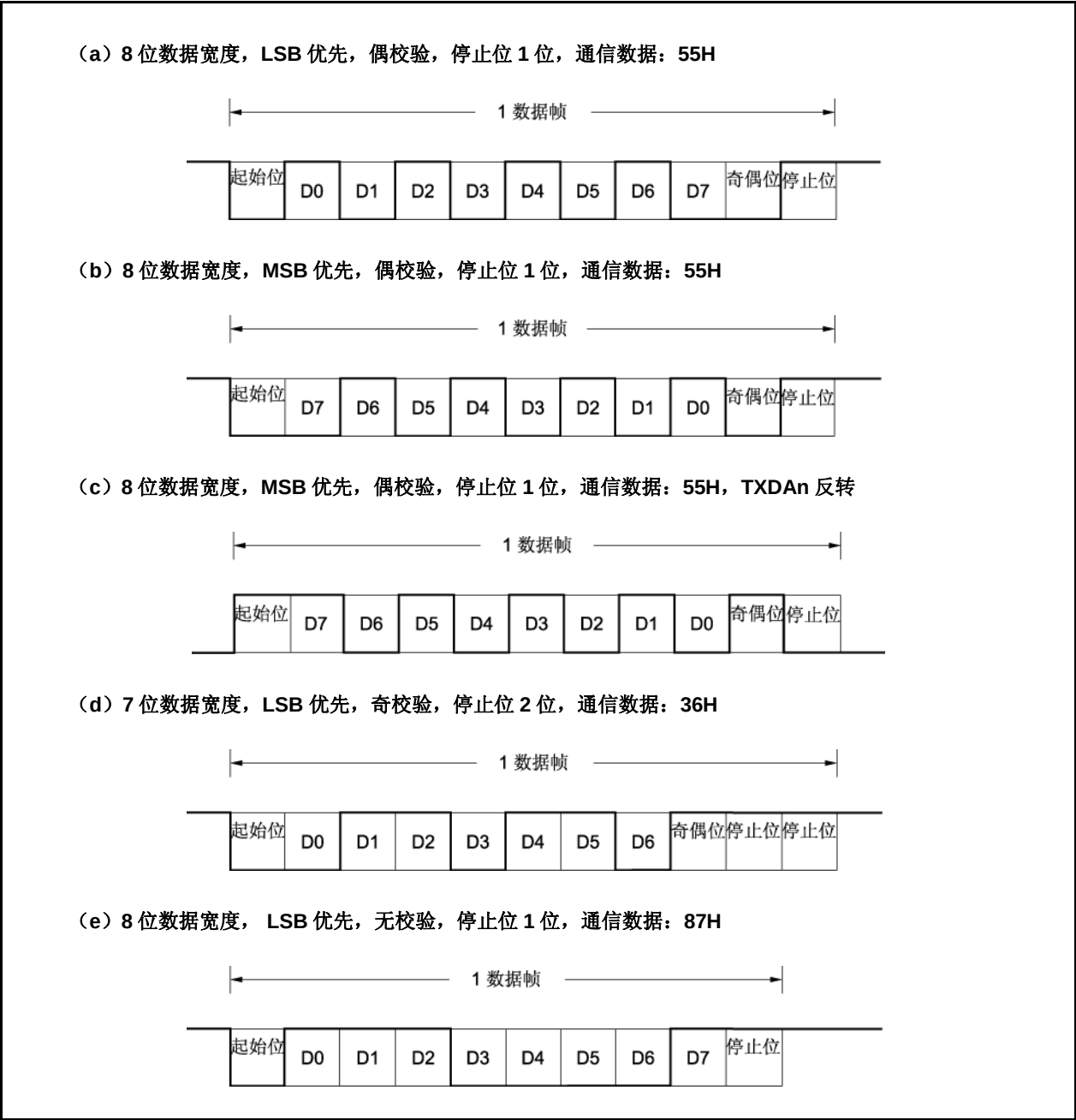
如图 15-5 所示，发送/接收数据的数据帧由起始位，字符位，校验位和停止位组成。

使用寄存器 UAnCTL0 来指定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 15-5. UARTA 发送/接收数据格式



<R>

15.6.2 SBF 发送/接收格式

<R> V850ES/SG2 和 V850ES/SG2-H 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 ~ 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（single-master）通信方式，一个主设备上最多可以连接 15 个从设备。LIN 的从设备用于控制开关、制动器和传感器，这些设备通过 LIN 网与 LIN 主设备相连。通常，LIN 主设备与一个网络如 CAN（控制器局域网 Controller Area Network）相连。此外，LIN 总线采用单线方式，通过符合 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±15%范围内时，可以进行通信。

图 15-6 和 15-7 概括了 LIN 的发送和接收操作。

图 15-6. LIN 发送操作

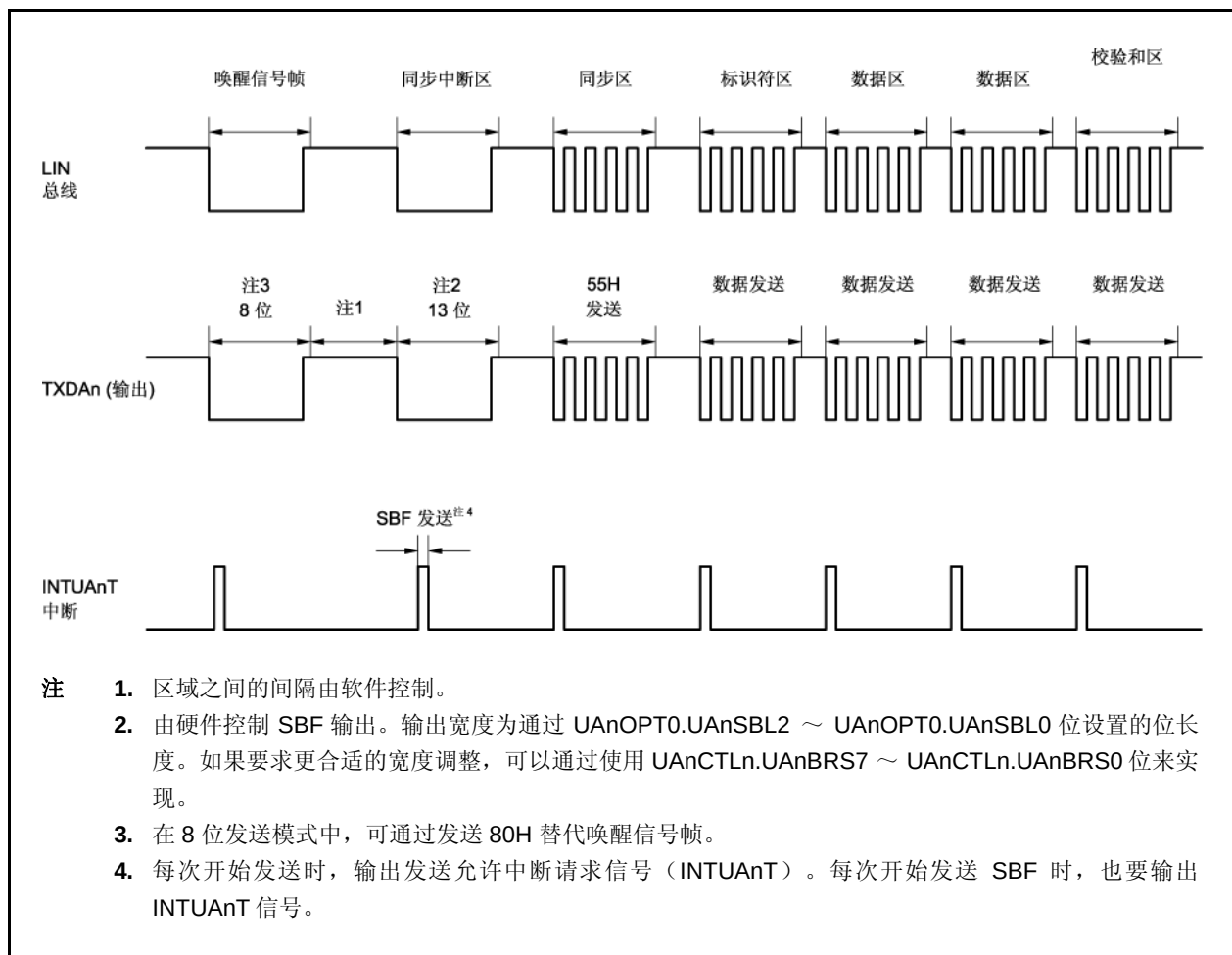
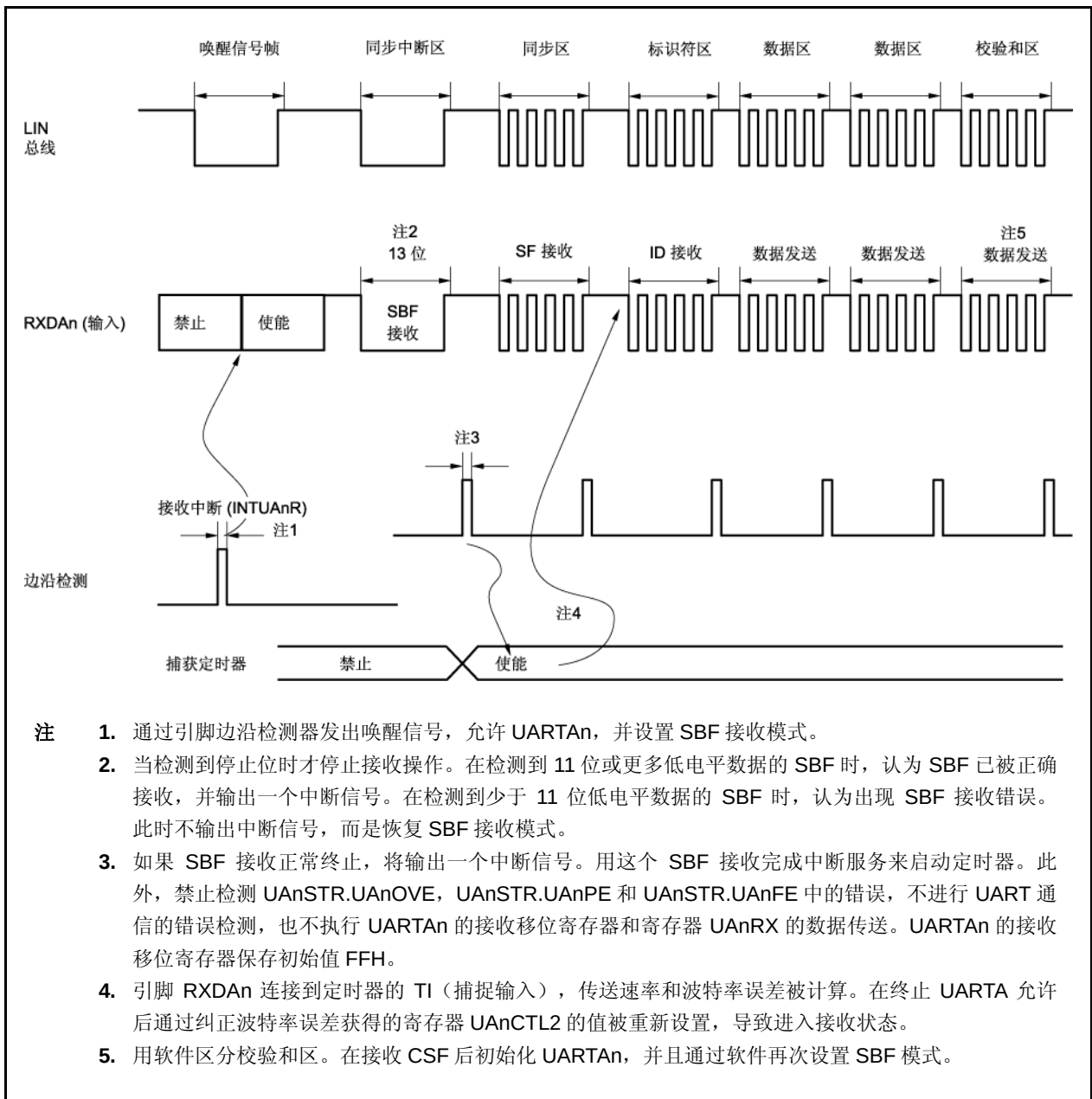


图 15-7. LIN 接收操作



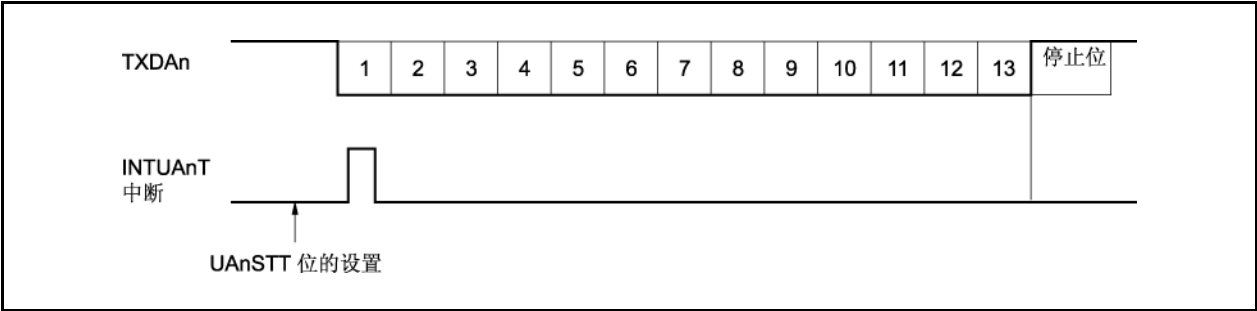
15.6.3 SBF发送

当 UAnCTL0.UAnPWR 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 13 ~ 20 位 (由 UAnOPT0.UAnSLS2 ~ 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。

发送被暂停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。

图 15-8. SBF 发送



15.6.4 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。

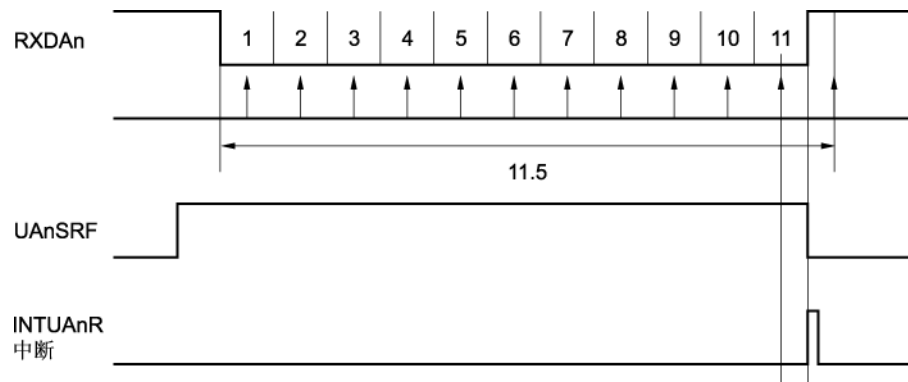
当接收到停止位时，如果 SBF 的宽度大于等于 11 位，判断为正常操作，并输出接收完成中断请求信号 (INTUAnR)。UAnOPT0.UAnSRF 位自动清零，SBF 接收结束。禁止 UAnSTR.UAnOVE，UAnSTR.UAnPE 和 UAnSTR.UAnFE 中的错误检测，不进行 UART 通信的错误检测。此外，不执行 UARTAn 的接收移位寄存器和寄存器 UAnRX 的数据传送，并且保存初始值 FFH。如果 SBF 的宽度小于等于 10 位，接收终止且错误处理中不输出中断，然后返回 SBF 接收模式。UAnSRF 位此时不清除。

<R>

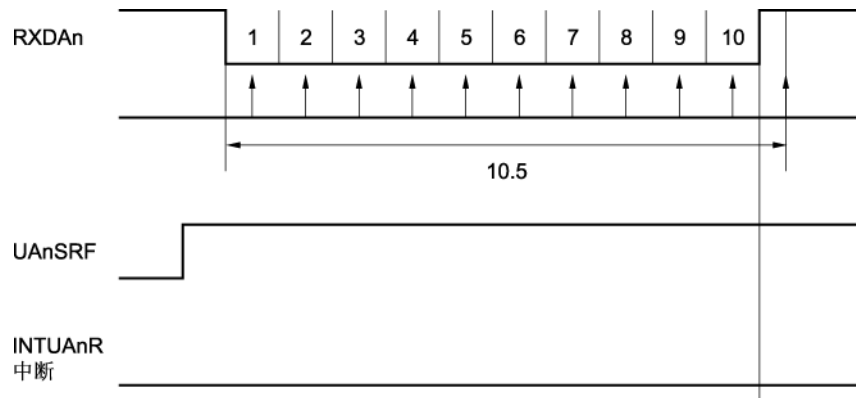
注意事项 LIN 功能不假定数据被接收时 SBF 被发送。因此，如果数据被接收时 SBF 被发送，一个帧错误发生 (UAnSTR.UAnFE 位 = 1)。

图 15-9. SBF 接收

(a) 正常 SBF 接收 (在大于 10.5 位的区域中检测到停止位)



(b) SBF 接收错误 (在小于等于 10.5 位的区域中检测到停止位)



15.6.5 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。

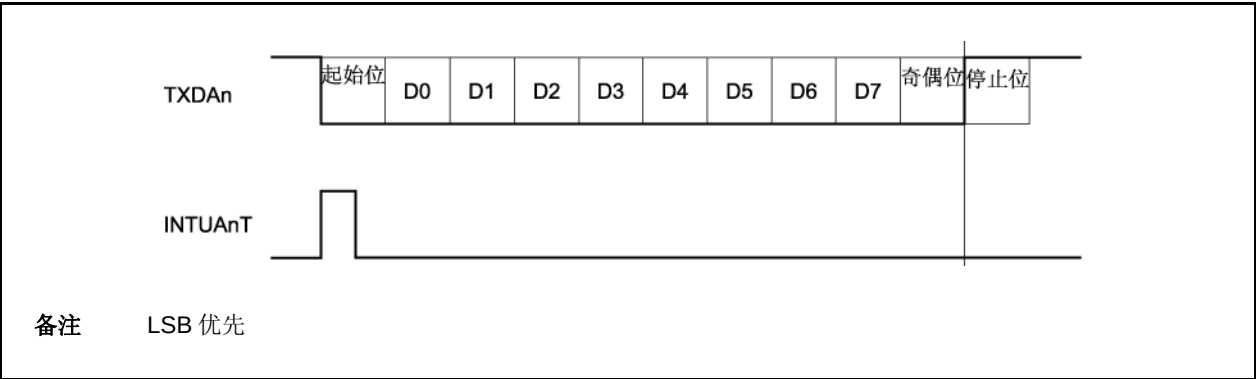
由于在 UARTAn 中没有提供 CTS（发送使能信号）输入引脚，所以使用某个端口检测发送终端的接收状态是否有效。

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 15-10. UART 发送



15.6.6 连续发送

UARTAn 可以在 UARTAn 的发送移位寄存器开始移位操作时，将下一个发送数据写入寄存器 UAnTX。发送移位寄存器 UARTAn 的发送时序可通过发送允许中断请求信号 (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，实现有效的通信率。

注意事项 在连续发送过程中执行发送初始化，确保 UAnSTR.UAnTSF 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。

图 15-11. 连续发送操作流程

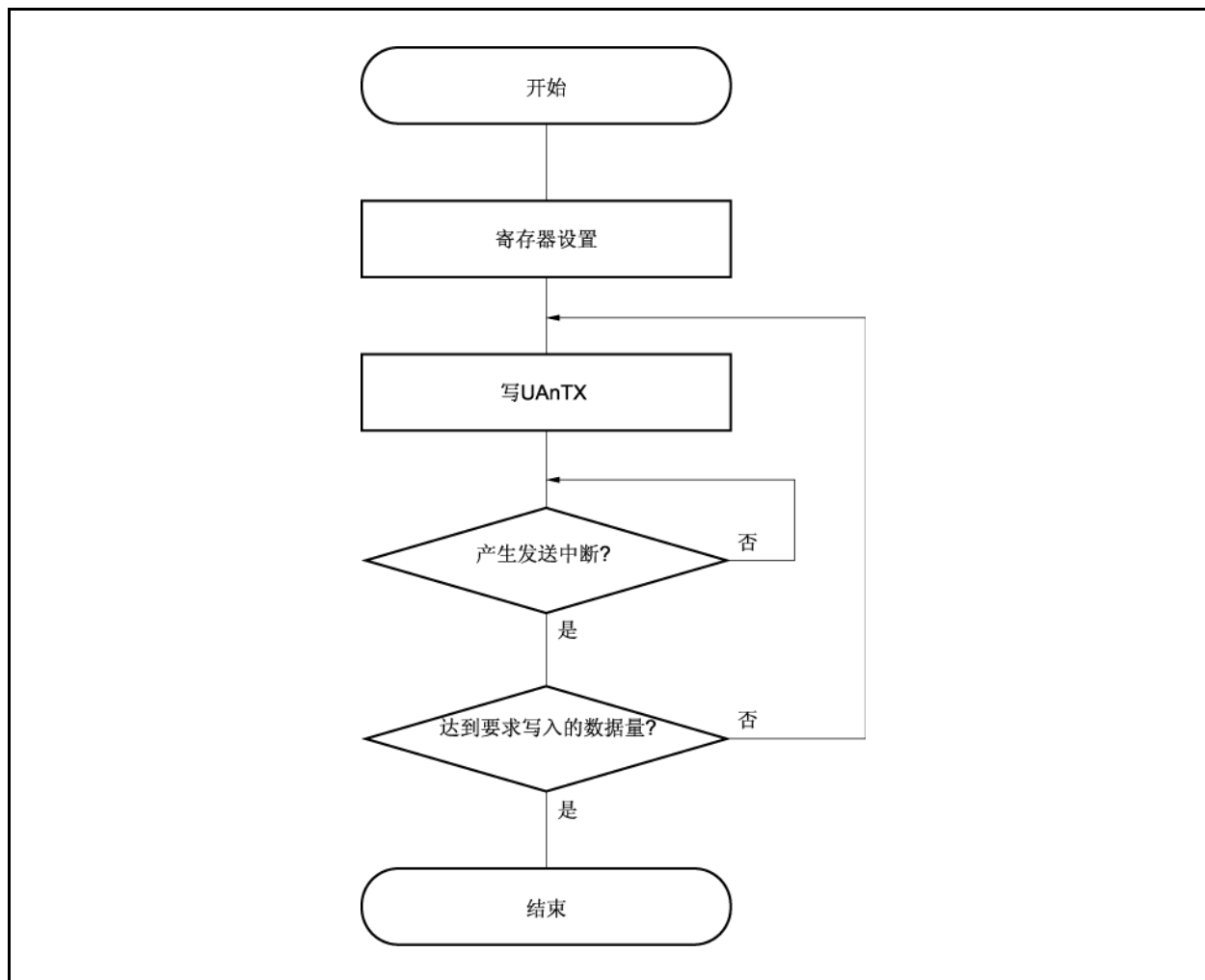
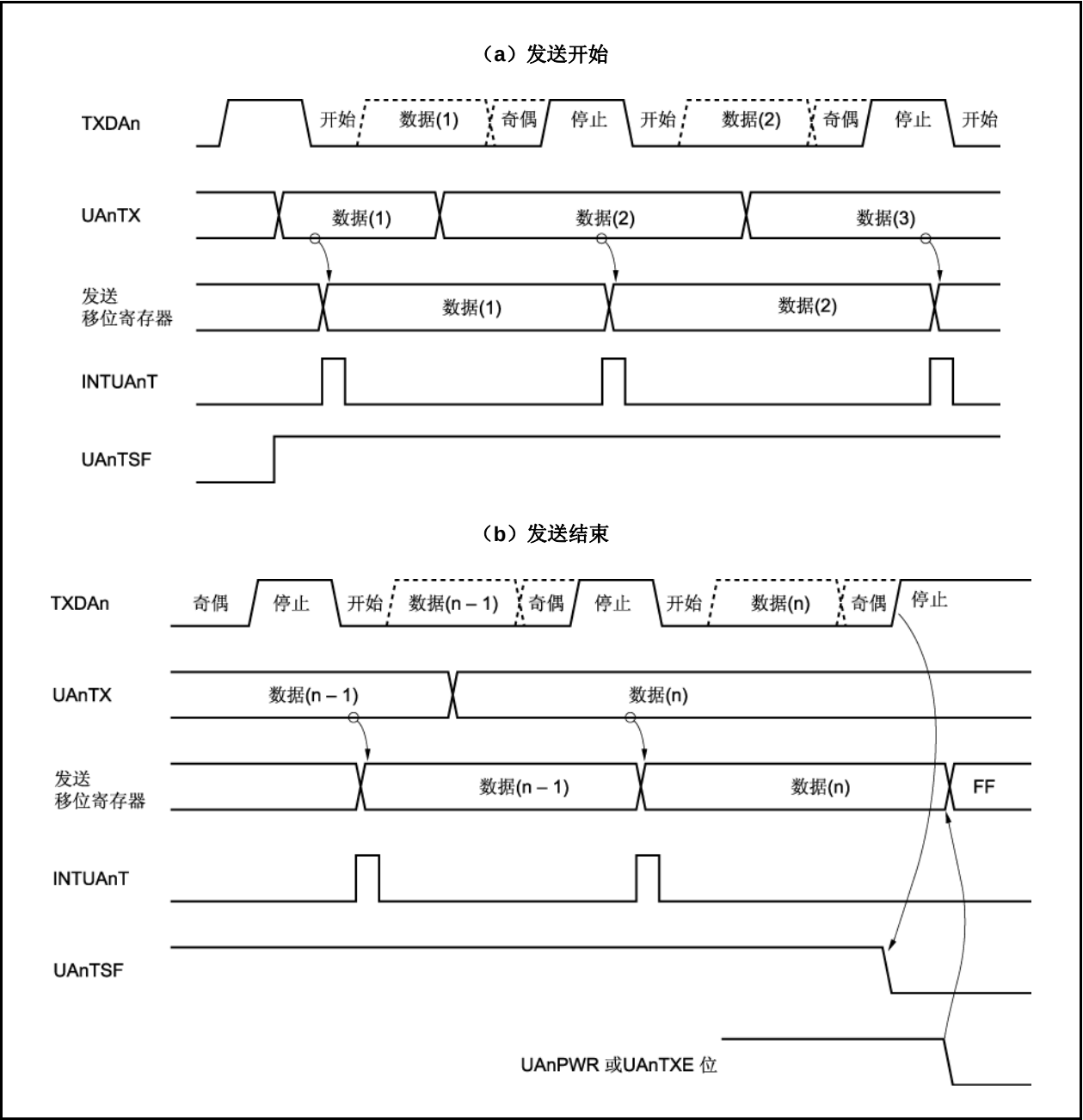


图 15-12. 连续发送操作时序



15.6.7 UART接收

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。

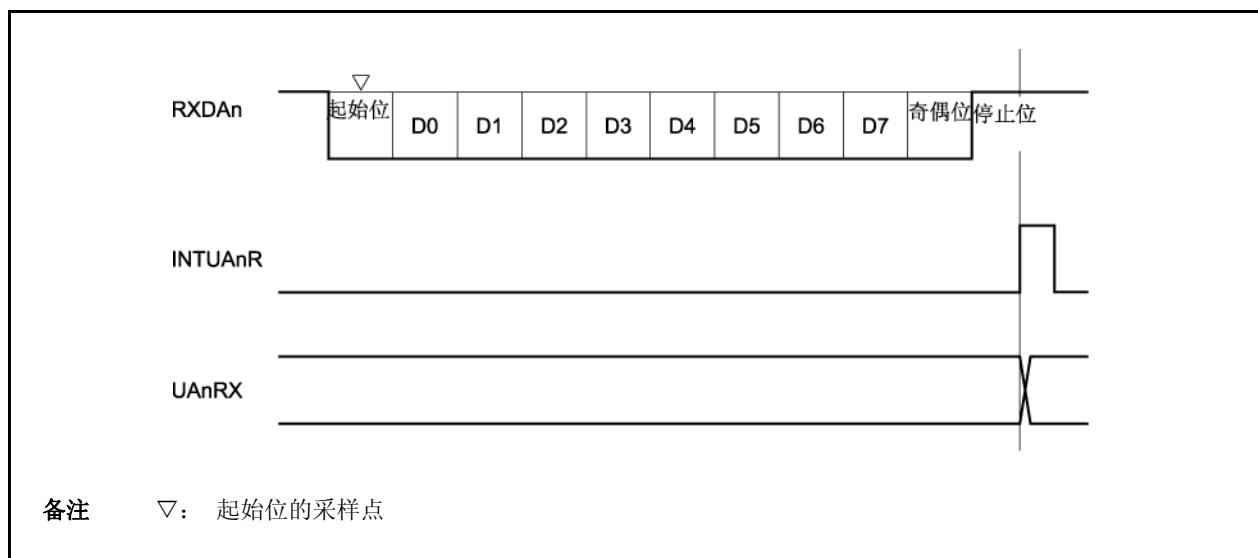
检测起始位使用 2 步检测子程。

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 位 = 1)，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 位 = 1) 或帧错误 (UAnSTR.UAnFE 位 = 1)，继续接收直到接收到第一个停止位，并在接收完全后输出 INTUAnR。

图 15-13. UART 接收



- 注意事项
1. 即使产生接收错误，也要确保读取寄存器 UAnRX。如果寄存器 UAnRX 不被读取，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. 接收操作在假定只有一个停止位的情况下进行。第 2 个停止位被忽略。
 3. 在接收完成，产生接收完成中断请求信号后，读取寄存器 UAnRX，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。

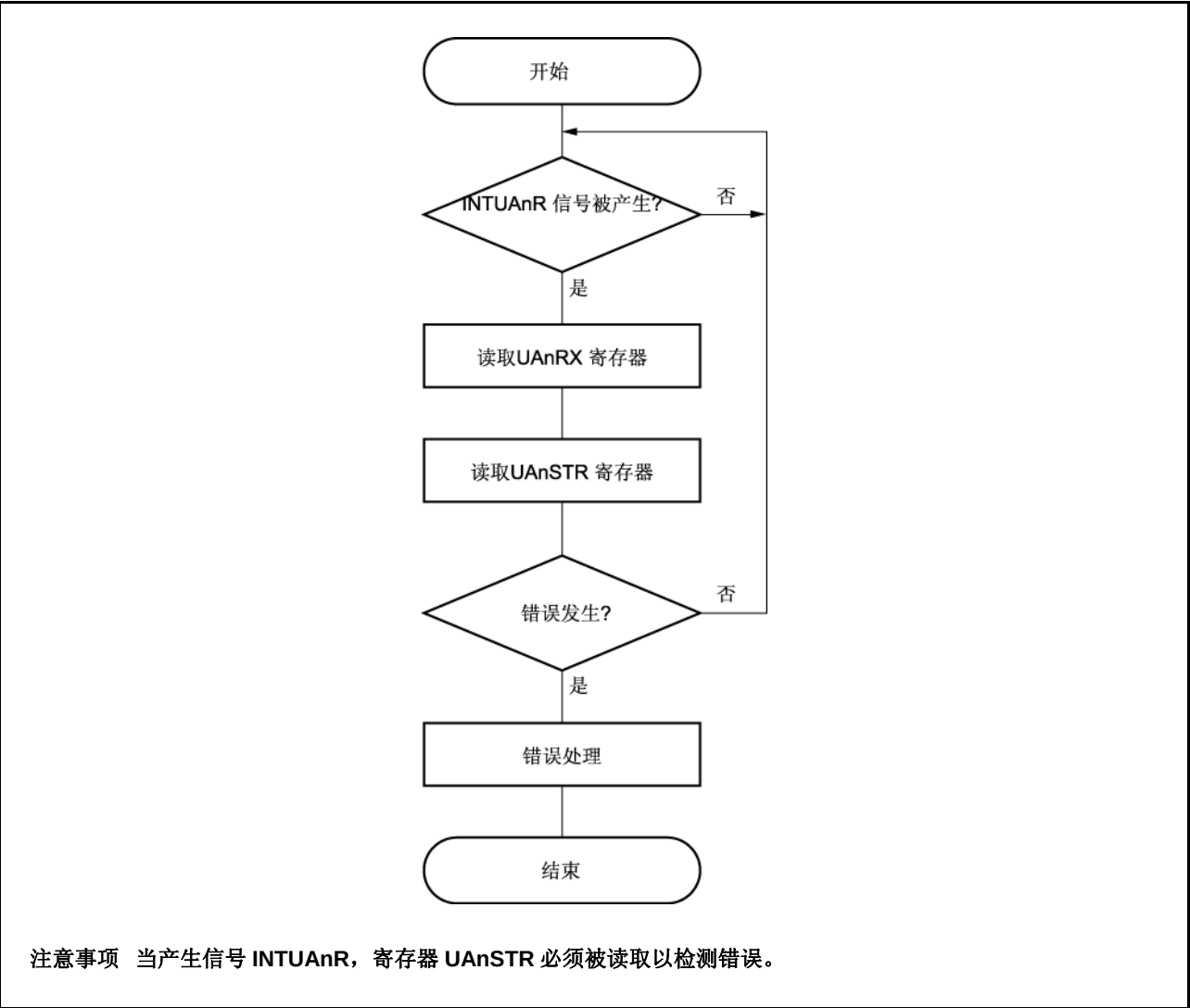
15.6.8 接收错误

接收过程中的错误有三种：校验错误，帧错误和溢出错误。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 UAnRX 寄存器中读取前，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。

15.6.9 校验类型和操作

注意事项 当使用 LIN 功能，固定寄存器 UAnCTL0 的 UAnPS1 和 UAnPS0 位为 00。

校验位用于检测通信数据的位错误。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”：0
- 如果发送数据有偶数个“1”：1

(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。因此无论校验位为何值，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。

15.6.10 接收数据的噪声过滤器

使用由预分频模块输出的基准时钟 (f_{CLK}) 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 15-15）。参见 15.7 (1) (a) 基本时钟 关于基本时钟。

此外，如图 15-14 所示的电路结构，接收操作的内部过程与外部信号状态相比延迟三个时钟才执行。

图 15-14. 噪声过滤器电路

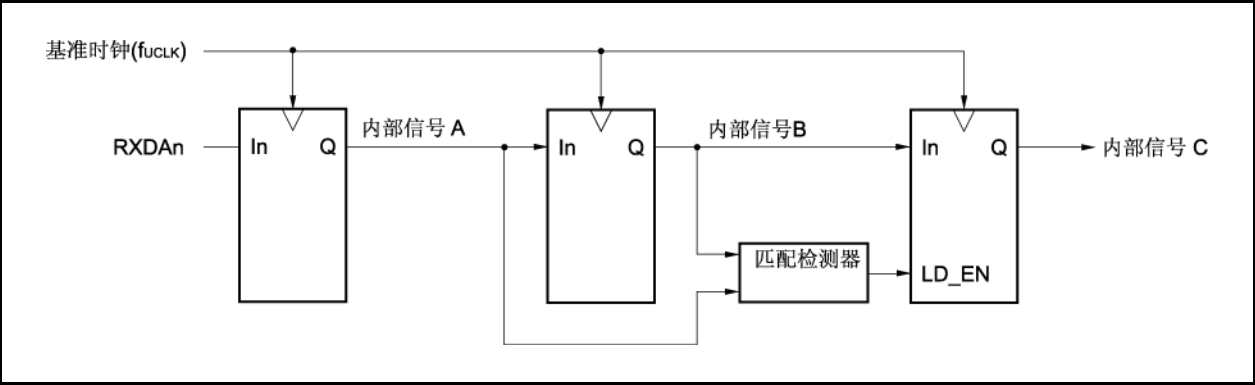
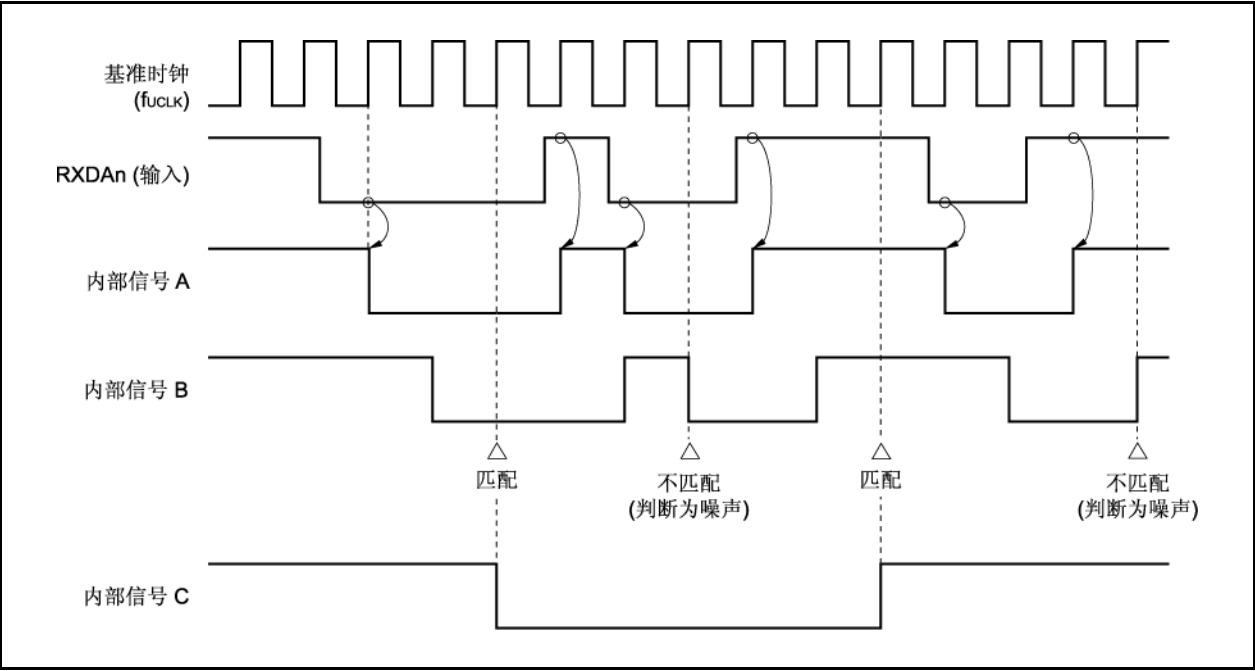


图 15-15. RXDAn 信号判断为噪声的时序



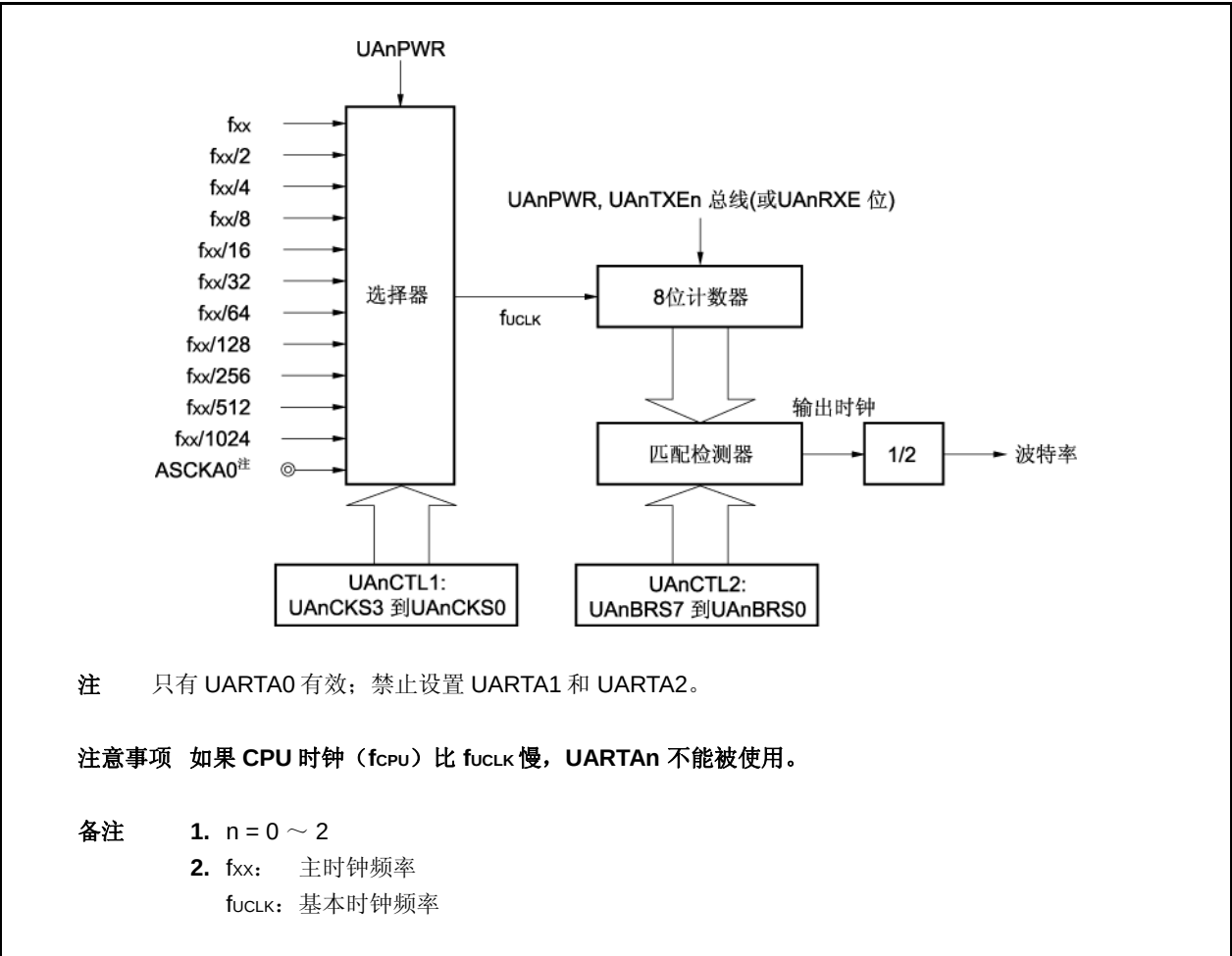
15.7 专用波特率发生器

专用波特率发生器由一个源时钟选择器模块和一个 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 15-16. 波特率发生器的配置



(a) 基本时钟

<R> 当 UAnCTL0.UAnPWR 位为 1 时，通过 UAnCTL1.UAnCKS3 到 UAnCTL1.UAnCKS0 位选择的时钟被提供
<R> 给 8 为计数器。该时钟称为基本时钟 (fUCLK)。当 UAnPWR 位为 0 时，基准时钟 fUCLK 固定为低电平。

(b) 串行时钟发生器

通过设置寄存器 UAnCTL1 和 UAnCTL2 (n = 0 to 2) 产生串行时钟。
通过 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 位选择基本时钟 (fUCLK)。
通过 UAnCTL2.UAnBRS7 ~ UAnCTL2.UAnBRS0 位设置 8 位计数器的分频值。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

该寄存器是 8 位寄存器，用于选择 UARTAn 的基本时钟。

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, UA2CTL1 FFFFFFFA21H

7	6	5	4	3	2	1	0
0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0

(n = 0 到2)

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	基准时钟(f _{CLK}) 选择
0	0	0	0	f _{xx}
0	0	0	1	f _{xx} /2
0	0	1	0	f _{xx} /4
0	0	1	1	f _{xx} /8
0	1	0	0	f _{xx} /16
0	1	0	1	f _{xx} /32
0	1	1	0	f _{xx} /64
0	1	1	1	f _{xx} /128
1	0	0	0	f _{xx} /256
1	0	0	1	f _{xx} /512
1	0	1	0	f _{xx} /1,024
1	0	1	1	外部时钟 ^註 (ASCKA0 管脚)
除上面以外				禁止设置

注 只有 UARTA0 有效；禁止设置 UARTA1 和 UARTA2。

备注 fxx: 主时钟频率

(3) UARTAn 控制寄存器 2 (UAnCTL2)

该寄存器是 8 位寄存器，用于选择 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,
UA2CTL2 FFFFFFFA22H

	7	6	5	4	3	2	1	0
UAnCTL2	UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0

(n = 0 到2)

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 f_{UCLK}: 由 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 位选择的基准时钟频率

(4) 波特率

通过下列公式计算波特率。

$$\text{波特率} = \frac{f_{\text{UCLK}}}{2 \times k} \quad [\text{bps}]$$

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）。

$$\text{波特率} = \frac{f_{\text{xx}}}{2^{m+1} \times k} \quad [\text{bps}]$$

备注 f_{UCLK} = 通过 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 位选择的基本时钟频率
 f_{xx} : 主时钟频率
 m = 由 UAnCTL1.UAnCKS3 ~ UAnCTL1.UAnCKS0 位设置的数值 ($m = 0$ 到 10)
 k = 由 UAnCTL2.UAnBRS7 ~ 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。

$$\begin{aligned} \text{Error (\%)} &= \left(\frac{\text{实际波特率 (有误差的波特率)}}{\text{目标波特率 (正确的波特率)}} - 1 \right) \times 100 [\%] \\ &= \left(\frac{f_{\text{UCLK}}}{2 \times k \times \text{预期波特率}} - 1 \right) \times 100 [\%] \end{aligned}$$

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）。

$$\text{Error (\%)} = \left(\frac{f_{\text{xx}}}{2^{m+1} \times k \times \text{预期波特率}} - 1 \right) \times 100 [\%]$$

注意事项 1. 发送期间必须保持波特率误差在接收端允许的误差范围内。
 2. 接收期间，波特率误差必须满足“（5）接收期间允许的波特率范围”中所描述的范围。

为了设置波特率，进行如下计算并设置寄存器 UAnCTL1 和 UAnCTL2（当使用内部时钟）。

<1> 设置 $k = f_{xx} / (2 \times \text{预期波特率})$ 。设置 $m = 0$ 。

<2> 设置 $k = k/2$ 和 $m = m + 1$ ，其中 $k \geq 256$ 。

<3> 重复步骤 <2> 直到 $k < 256$ 。

<4> 将 k 取整

如果取整后 $k = 256$ ，则再次执行 <2>（ k 会变成 128）。

<5> 将 m 设置到 UAnCTL1 寄存器，并将 k 设置到 UAnCTL2 寄存器。

例如：当 $f_{xx} = 20 \text{ MHz}$ 且预期波特率 = 153,600 bps

<1> $k = 20,000,000 / (2 \times 153,600) = 65.10\dots$ ， $m = 0$

<2>，<3> $k = 65.10\dots < 256$ ， $m = 0$

<4> 设置寄存器 UAnCTL2 的值： $k = 65 = 41\text{H}$ ，设置寄存器 UAnCTL1 的值： $m = 0$

实际波特率 = $20,000,000 / (2 \times 104)$
= 153,846 [bps]

波特率误差 = $\{20,000,000 / (2 \times 65 \times 153,600) - 1\} \times 100$
= 0.160 [%]

波特率设置的典型示例如下。

<R>

表 15-3. 波特率发生器的数据设置 (1/2)

波特率 (bps)	$f_{xx} = 32 \text{ MHz}^{\#}$			$f_{xx} = 20 \text{ MHz}$			$f_{xx} = 18.874 \text{ MHz}$		
	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)
300	08H	D0H	0.16	08H	82H	0.16	07H	F6H	-0.10
600	07H	D0H	0.16	07H	82H	0.16	06H	F6H	-0.10
1,200	06H	D0H	0.16	06H	82H	0.16	05H	F6H	-0.10
2,400	05H	D0H	0.16	05H	82H	0.16	04H	F6H	-0.10
4,800	04H	D0H	0.16	04H	82H	0.16	03H	F6H	-0.10
9,600	03H	D0H	0.16	03H	82H	0.16	02H	F6H	-0.10
19,200	02H	D0H	0.16	02H	82H	0.16	01H	F6H	-0.10
31,250	02H	80H	0.00	01H	A0H	0.00	01H	97H	-0.01
38,400	01H	D0H	0.16	01H	82H	0.16	00H	F6H	-0.10
76,800	00H	D0H	0.16	00H	82H	0.16	00H	7BH	-0.10
153,600	00H	68H	0.16	00H	41H	0.16	00H	3DH	0.72
312,500	00H	33H	0.39	00H	20H	0.00	00H	1EH	0.66

注 值适用于 V850ES/SG2-H

备注 f_{xx} : 主时钟频率

ERR: 波特率误差 (%)

表 15-3. 波特率发生器的数据设置 (2/2)

波特率 (bps)	$f_{xx} = 16 \text{ MHz}$			$f_{xx} = 10 \text{ MHz}$		
	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)
300	07H	D0H	0.16	07H	82H	0.16
600	06H	D0H	0.16	06H	82H	0.16
1,200	05H	D0H	0.16	05H	82H	0.16
2,400	04H	D0H	0.16	04H	82H	0.16
4,800	03H	D0H	0.16	03H	82H	0.16
9,600	02H	D0H	0.16	02H	82H	0.16
19,200	01H	D0H	0.16	01H	82H	0.16
31,250	01H	80H	0.00	00H	A0H	0.00
38,400	00H	D0H	0.16	00H	82H	0.16
76,800	00H	68H	0.16	00H	41H	0.16
153,600	00H	34H	0.16	00H	21H	-1.36
312,500	00H	1AH	-1.54	00H	10H	0.00

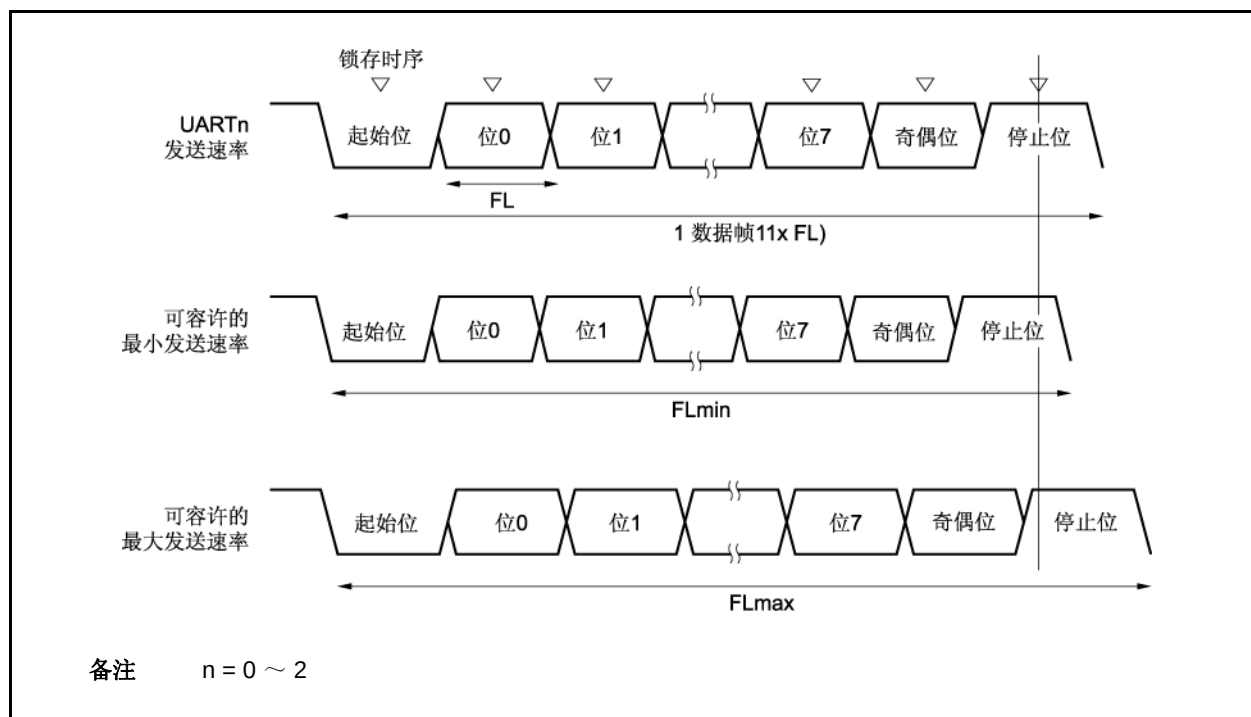
备注 f_{xx} : 主时钟频率
ERR: 波特率误差 (%)

(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 15-17. 接收期间允许的波特率范围



如图 15-17 所示，当检测到起始位后，接收数据的锁存时序由寄存器 UAnCTL2 设置的计数器确定。如果数据的最后一位（停止位）满足该锁存时序，则发送数据可被正确接收。

假定接收数据为 11 位，各项理论值计算如下。

$$FL = (\text{Brate})^{-1}$$

Brate: UARTAn 波特率 (n = 0 to 2)

k: UAnCTL2.UAnBRS7 ~ UAnCTL2.UAnBRS0 位的设置值 (n = 0 到 2)

FL: 1 位数据宽度

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 15-4. 允许的最大/最小波特率误差

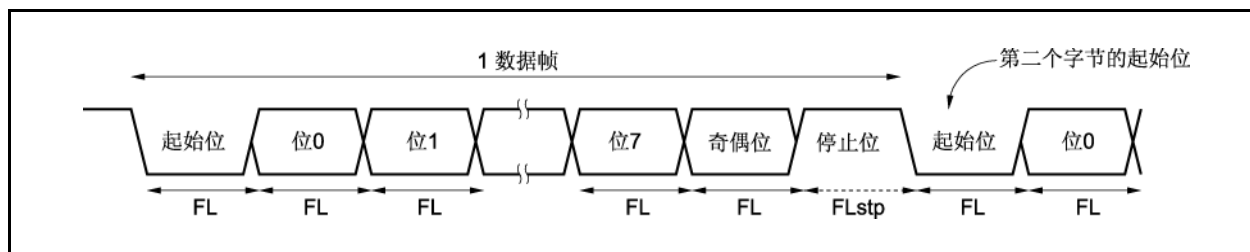
分频比 (k)	允许的最大波特率误差	允许的最小波特率误差
4	+2.32%	-2.43%
8	+3.52%	-3.61%
20	+4.26%	-4.30%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.72%

- 备注**
1. 接收准确度取决于每帧上的位数、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 ~ UAnCTL2.UAnBRS0 位的设置值 (n = 0 到 2)

(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 15-18. 连续发送期间的传送速率



假设 1 位数据宽度为 FL、停止位宽度为 FLstp，基本时钟频率为 f_{CLK} 时，我们可以得到以下的公式。

$$FL_{stp} = FL + 2/f_{uCLK}$$

因此，连续发送期间的发送速率如下：

$$\text{发送速率} = 11 \times FL + (2/f_{uCLK})$$

15.8 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。当使用 DMA 来实现发送数据和接收数据的传送时，即使传送期间出现错误（奇偶校验，溢出，帧错误），也不会进行错误处理。可以在 DMA 传送后读取寄存器 UAnSTR 确认没有错误，或着在通信过程中读取寄存器 UAnSTR 查看是否出错。
- (4) 按照下述次序启动 UARTAn。
 - <1> 设置 UAnCTL0.UAnPWR 为 1。
 - <2> 设置端口。
 - <3> 设置 UAnCTL0.UAnTXE 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。

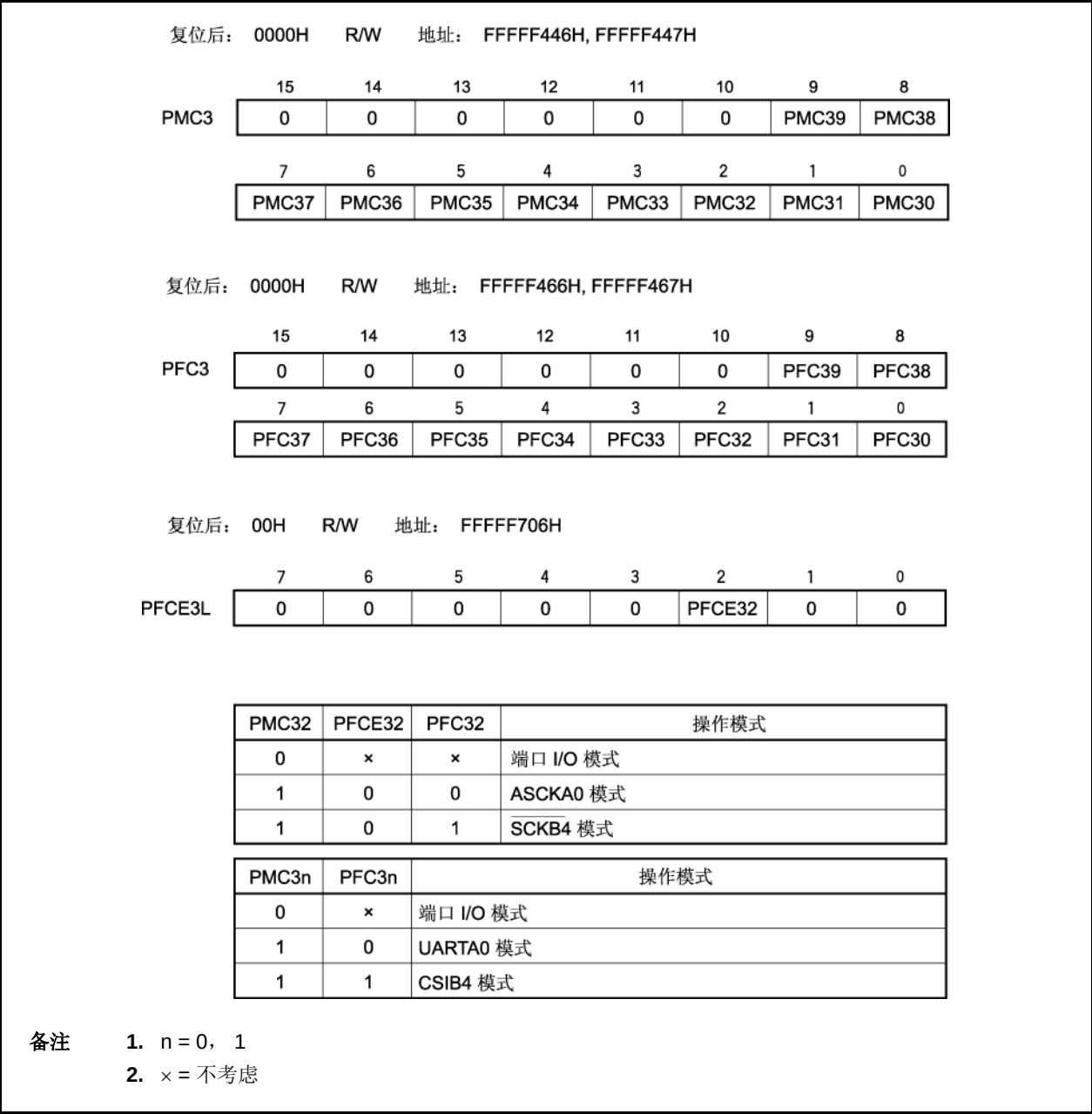
16.1 CSIB和其他接口的模式转换

16.1.1 CSIB4 和 UARTA0 的模式转换

<R> 在 V850ES/SG2 和 V850ES/SG2-H 中，CSIB4 和 UARTA0 是同一引脚的复用功能，因此它们不能同时使用。在使用 CSIB4 前使用 PMC3 和 PFC3 寄存器预先对 CSIB4 进行设置。

注意事项 如果这些功能在传送或接收期间进行切换，那么 CSIB4 和 UARTA0 的发送 / 接收操作将不能被保证。请务必禁止未使用的那个。

图 16-1. CSIB4 和 UARTA0 的模式转换设置



16.1.2 CSIB0 和 I²C01 的模式转换

<R> 在 V850ES/SG2 和 V850ES/SG2-H 的 I²C 总线版本 (Y 产品) 中, CSIB0 和 I²C01 是同一引脚的复用功能, 因此它们不能同时使用。在使用 CSIB0 前使用 PMC4 和 PFC4 寄存器预先对 CSIB0 进行设置。

注意事项 如果这些功能在传送或接收期间进行切换, 那么 CSIB0 和 I²C01 的发送 / 接收操作将不能被保证。请务必禁止未使用的那个。

图 16-2. CSIB0 和 I²C01 的模式转换设置

复位后: 00H R/W 地址: FFFFF448H

PMC4

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

复位后: 00H R/W 地址: FFFFF468H

PFC4

7	6	5	4	3	2	1	0
0	0	0	0	0	0	PFC41	PFC40

PMC4n	PFC4n	操作模式
0	×	端口I/O 模式
1	0	CSIB0 模式
1	1	I ² C01 模式

备注

1. n = 0, 1

2. × = 不考虑

16.2 特点

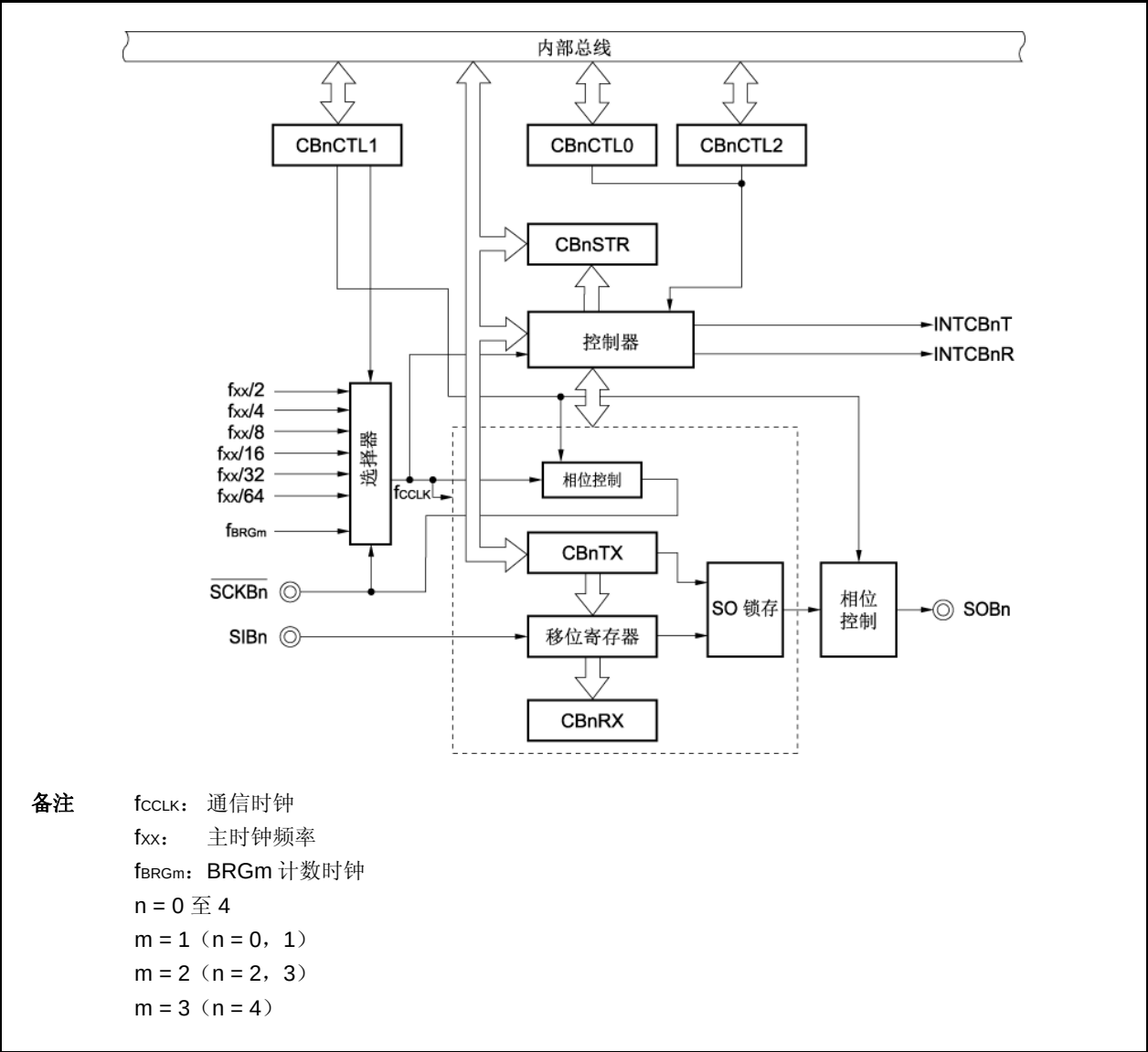
- <R> ○ 传输速率： 最大 8 Mbps。
 (V850ES/SG2: $f_{xx} = 20 \text{ MHz}$,
 V850ES/SG2-H: $f_{xx} = 32 \text{ MHz}$, 使用内部时间)
- 主模式和从模式可选
- 8 位至 16 位发送, 3 线串行接口
- 中断请求信号 (INTCBnT, INTCBnR)
- 串行时钟和数据相位可转换
- 可以 1 位为单位在 8 至 16 位之间选择通信数据长度
- 发送数据 MSB 优先/LSB 优先可切换
- 3 线通信 SOBn: 串行数据输出
 SIBn: 串行数据输入
 $\overline{\text{SCKBn}}$: 串行时钟输出
 可指定发送模式, 接收模式和发送 / 接收模式

备注 n = 0 至 4

16.3 配置

CSIBn 的框图如下。

图 16-3. CSIBn 的框图



CSIBn 包含以下硬件。

表 16-1. CSIBn 的配置

项目	配置
寄存器	CSIBn 数据接收寄存器 (CBnRX) CSIBn 数据发送寄存器 (CBnTX)
控制寄存器	CSIBn 控制寄存器 0 (CBnCTL0) CSIBn 控制寄存器 1 (CBnCTL1) CSIBn 控制寄存器 2 (CBnCTL2) CSIBn 状态寄存器 (CBnSTR)

(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。

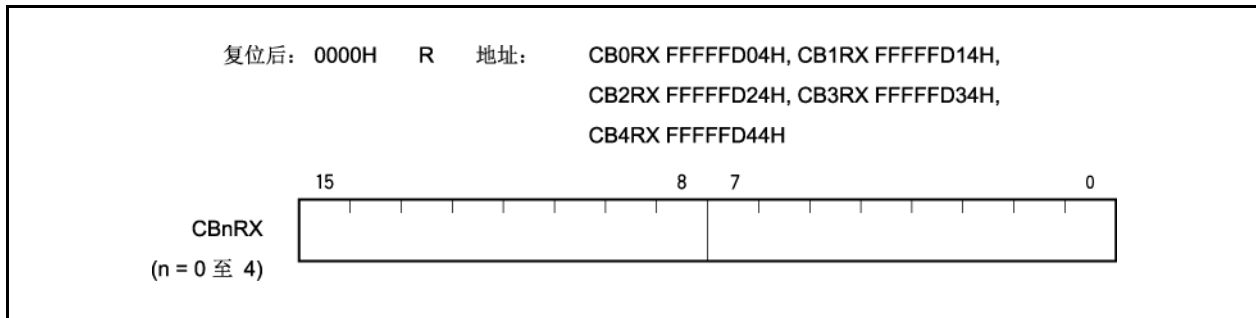
该寄存器是只读的，可由 16 位存储器操作指令设置该寄存器。

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。

复位输入将把该寄存器设置为 0000H。

除了复位信号，通过清除（置 0）寄存器 CBnCTL0 的 CBnPWR 位，寄存器 CBnRX 可被初始化。

**(2) CSIBn 数据发送寄存器 (CBnTX)**

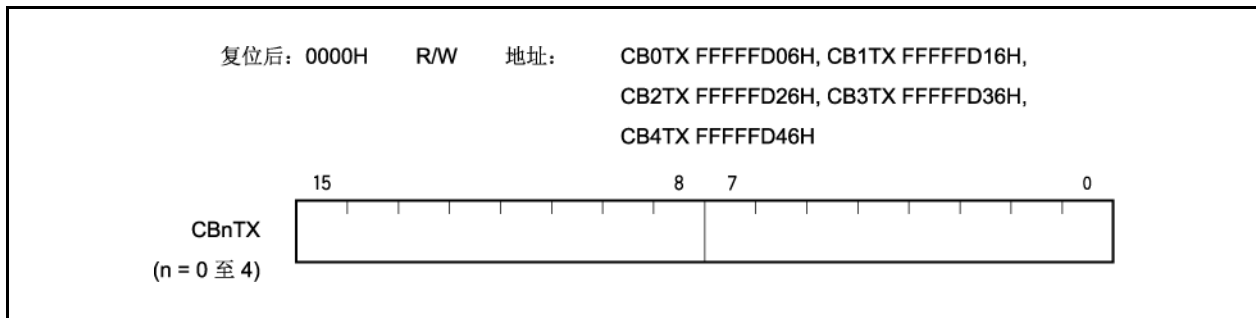
寄存器是 16 位缓冲寄存器，用于写入 CSIBn 以发送数据。

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，该寄存器的低 8 位只能以 8 位为单位进行读取，就如同 CBnTXL 寄存器。

复位输入将把该寄存器设置为 0000H。



备注 通信开始条件如下所示。

发送模式 (CBnTXE 位 = 1, CBnRXE 位 = 0) :

发送 / 接收模式 (CBnTXE 位 = 1, CBnRXE 位 = 1) :

接收模式 (CBnTXE 位 = 0, CBnRXE 位 = 1) :

写入 CBnTX 寄存器

写入 CBnTX 寄存器

从 CBnTX 寄存器中读取

CBnDIR ^注	传送方向模式规范 (MSB/LSB)
0	MSB 优先传送
1	LSB 优先传送

CBnTMS ^注	传送模式规范
0	单个传送模式
1	连续传送模式

[在单个传送模式中]

在通信完成时会产生接收完成中断 (INTCBnR)。

即使允许传送 (CBnTXE 位 = 1)，也不会产生传送允许中断 (INTCBnT)。

如果下个传送数据在通信期间被写入 (CBnSTR.CBnTSF 位 = 1)，它将会被忽略而下个通信也不会开始。同样，如果设置只接收的通信 (CBnTXE 位 = 0，CBnRXE 位 = 1)，那么即使在通信期间读取了接收数据 (CBnSTR.CBnTSF 位 = 1)，下次通信也不会开始。

[在连续传送模式中]

通过在通信期间写入下个传送数据 (CBnSTR.CBnTSF 位 = 1) 可以允许连续传送。

在发生传送允许中断 (INTCBnT) 后允许写入下个传送数据。如果在连续传送模式中只设置了只接收通信 (CBnTXE 位 = 0，CBnRXE 位 = 1)，那么在接收完成中断 (INTCBnR) 后，不管进行何种 CBnRX 寄存器的读取操作都会开始下次接收。

因此，应立即从 CBnRX 寄存器中读取接收数据。如果没有显示该读取操作，那么将会发生溢出错误 (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将 CBnPWR 位设为 1。

CBnSCE	禁止/允许开始传送的规范
0	通信开始触发无效
1	通信开始触发有效
- 在主模式中 该位允许或禁止通信开始触发。 (a) 在单个传送或单个传送/接收模式中，或在连续传送或连续传送/接收模式中。 CBnSCE位的设置对通信操作没有影响。 (b) 在单个接收模式中 在读取最后的接收数据前将CBnSCE位清零，这是因为接收是通过读取接收数据 (CBnRX 寄存器)开始的，这会禁止接收开始^{注1}。 (c) 在连续接收模式中 在完成最后的数据接收前的一个通信时钟内将CBnSCE位清零，用来禁止在接收最后的数据后的接收操作^{注2}。 - 在从模式中 该位允许或禁止通信开始触发。 将CBnSCE位设为1。^{注3}	

- 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。
要在读取最后一个数据后再次开始通信操作，则将 CBnSCE 位设为“1”并对 CBnRX 寄存器进行伪读取。

3. 要开始接收，伪读取是必需的。

<R>

(a) 使用 CBnSCE 位的方法

(i) 在单个接收模式中

- <1> 当通过 INTCBnR 中断服务完成最后的数据接收时，应将 CBnSCE 位清零，然后再读取 CBnRX 寄存器。
- <2> 完成最后的数据接收后，在接收被禁止时应检测 CBnSTR.CBnTSF 位为 0，然后将 CBnPWR 和 CBnRXE 位清零。要继续接收，则应将 CBnSCE 位设为 1 并通过执行 CBnRX 寄存器的伪读取来开始下一次接收操作。

(ii) 在连续接收模式中

- <1> 通过最后的接收的前一次接收的 INTCBnR 中断服务在最后的数据接收期间将 CBnSCE 位清零，然后读取 CBnRX 寄存器。
- <2> 在接收到最后一个的接收的 INTCBnR 信号后，从 CBnRX 寄存器中读取最后的数据。
- <3> 完成最后的数据接收后，在接收被禁止时应检测 CBnSTR.CBnTSF 位为 0，然后将 CBnPWR 和 CBnRXE 位清零。要继续接收，则应将 CBnSCE 位设为 1 并通过执行 CBnRX 寄存器的伪读取来开始下一次接收操作。

注意事项 在连续接收模式中，串行时钟不会被停止直到通过伪读取开始接收后，CBnSCE 位被清零时所进行的接收操作被完成。

(2) CSIBn 控制寄存器 1 (CBnCTL1)

CBnCTL1 是一个 8 位寄存器，用于控制 CSIBn 串行传送操作。
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。
复位输入将把该寄存器置为 00H。

注意事项 CBnCTL1 寄存器只能在 CBnCTL0.CBnPWR 位为 0 时进行改写。

复位后：00H R/W 地址： CB0CTL1 FFFFFFFD01H, CB1CTL1 FFFFFFFD11H,
CB2CTL1 FFFFFFFD21H, CB3CTL1 FFFFFFFD31H,
CB4CTL1 FFFFFFFD41H

	7	6	5	4	3	2	1	0
CBnCTL1	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

(n = 0 至 4)

	CBnCKP	CBnDAP	与SCKBn相关的数据 传送/接收时序的规范
通信 类型 1	0	0	
通信 类型 2	0	1	
通信 类型 3	1	0	
通信 类型 4	1	1	

CBnCKS2	CBnCKS1	CBnCKS0	通信时钟 (f _{CCLK}) ^注	模式
0	0	0	f _{xx} /2	主模式
0	0	1	f _{xx} /4	主模式
0	1	0	f _{xx} /8	主模式
0	1	1	f _{xx} /16	主模式
1	0	0	f _{xx} /32	主模式
1	0	1	f _{xx} /64	主模式
1	1	0	f _{BRGm}	主模式
1	1	1	外部时钟 (SCKBn)	从模式

注 将通信时钟 (f_{CCLK}) 设为 8 MHz 或更低。

备注 当 n = 0, 1, m = 1 时
当 n = 2, 3, m = 2 时
当 n = 4, m = 3 时
关于 f_{BRGm} 的详细信息请参见 16.8 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。

复位输入将把该寄存器置为 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,
CB2CTL2 FFFFFFFD22H, CB3CTL2 FFFFFFFD32H,
CB4CTL2 FFFFFFFD42H

	7	6	5	4	3	2	1	0
CBnCTL2	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0
(n = 0 至 4)								

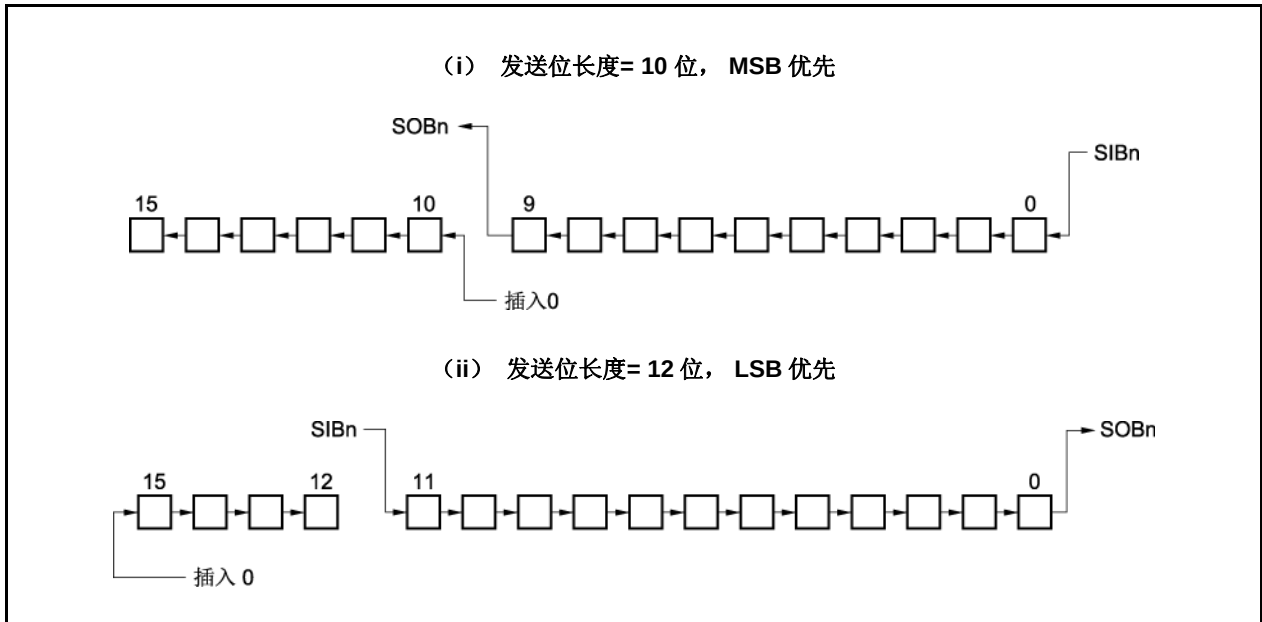
CBnCL3	CBnCL2	CBnCL1	CBnCL0	串行寄存器位长度
0	0	0	0	8 位
0	0	0	1	9 位
0	0	1	0	10 位
0	0	1	1	11 位
0	1	0	0	12 位
0	1	0	1	13 位
0	1	1	0	14 位
0	1	1	1	15 位
1	x	x	x	16 位

- 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. x: 不考虑

(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，但是接收的数据在串行发送后都变为 0。



(4) CSIBn 状态寄存器 (CBnSTR)

CBnSTR 是一个 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。

复位输入将把该寄存器置为 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 FFFFFFFD03H, CB1STR FFFFFFFD13H,
CB2STR FFFFFFFD23H, CB3STR FFFFFFFD33H,
CB4STR FFFFFFFD43H

	<7>	6	5	4	3	2	1	<0>
CBnSTR (n = 0 至 4)	CBnTSF	0	0	0	0	0	0	CBnOVE

CBnTSF	通信状态标志
0	通信停止
1	通信中
在传送期间，在CBnTX寄存器中的数据准备好后对该寄存器进行设置，而在接收期间，则在对CBnRX寄存器进行伪读取时设置该寄存器。传送结束时，该标志在时钟最后的边沿处被清零。	

CBnOVE	溢出错误标志
0	没有溢出
1	溢出
- 完成接收操作后，在没有通过CPU读取CBnRX寄存器值的情况下，在下个接收开始时会发生溢出错误。此时，CBnOVE标志显示了溢出错误发生状态。 - 在单个传送模式中CBnOVE位也是有效的。因此在只使用传送时，需注意以下内容。 - 不检查 CBnOVE 标志。（建议的） - 即使不需要读取接收数据也读取该位。 - 通过对CBnOVE标志写入0来清除它。它不能通过对其写入1来进行设置。	

<R> **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。

例如，如果在通过生成 INTCBnR 信号开始 DMA 时下个数据被写入到 CBnTX 寄存器，那么被写入的数据将不会被发送，这是因为 CBnTSF 位被设为 1。

对这类应用程序使用连续发送模式而不是单独发送模式。

<R> 16.5 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 16-2. 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。
发生溢出错误时也将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，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)

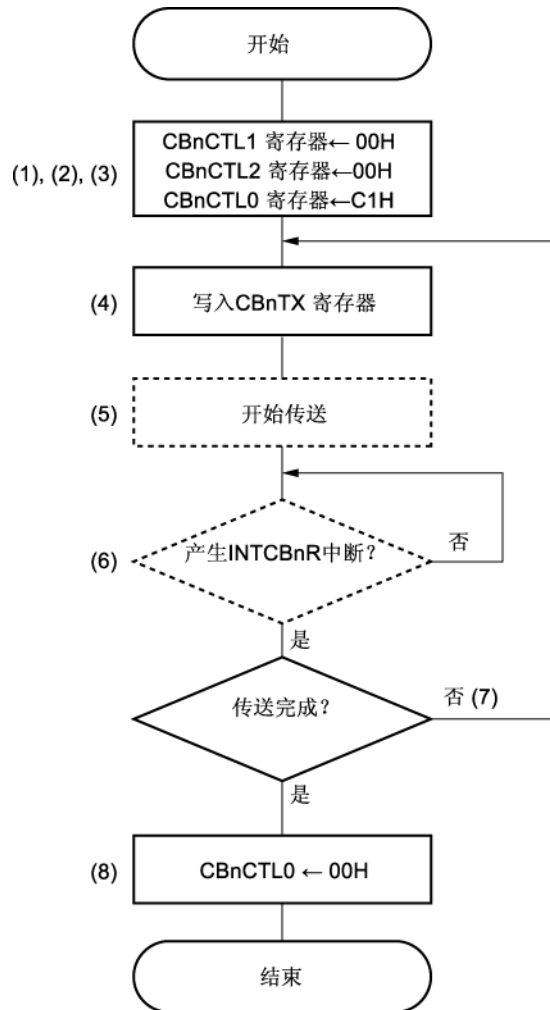
在连续发送或连续发送 / 接收模式中，一旦允许对 CBnTX 进行写操作，发送数据将会立即被转移到 CBnTX 寄存器中且会生成一个发送允许中断请求信号。
在单独发送模式以及单独发送 / 接收模式中，不会生成 INTCBnT 中断。

16.6 操作

<R> 16.6.1 单独发送模式（主模式，发送模式）

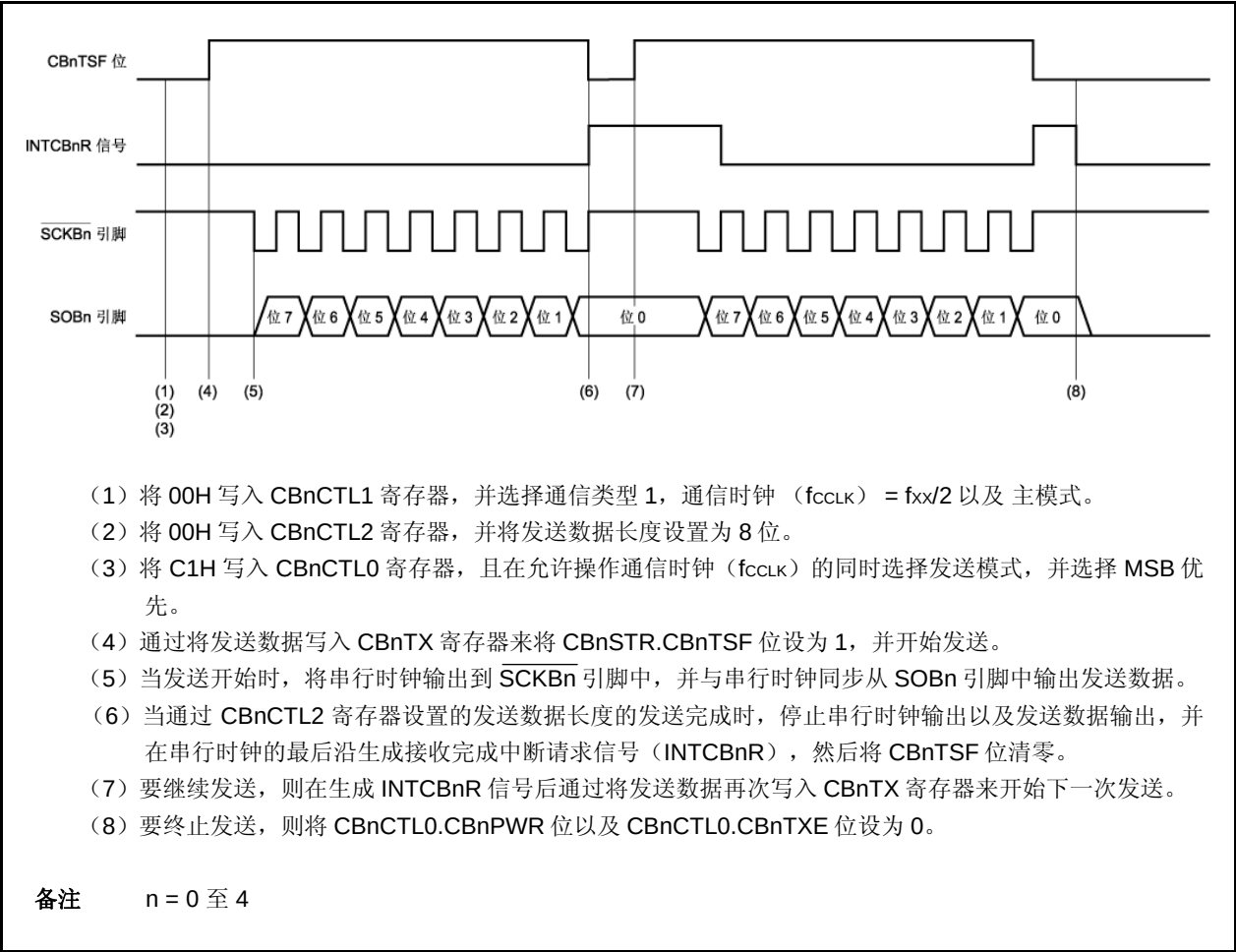
MSB 优先（CBnCTL0.CBnDIR 位 = 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00），通信时钟（f_{CCLK}）= f_{xx}/2（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000），发送数据长度 = 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000）

(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 至 4

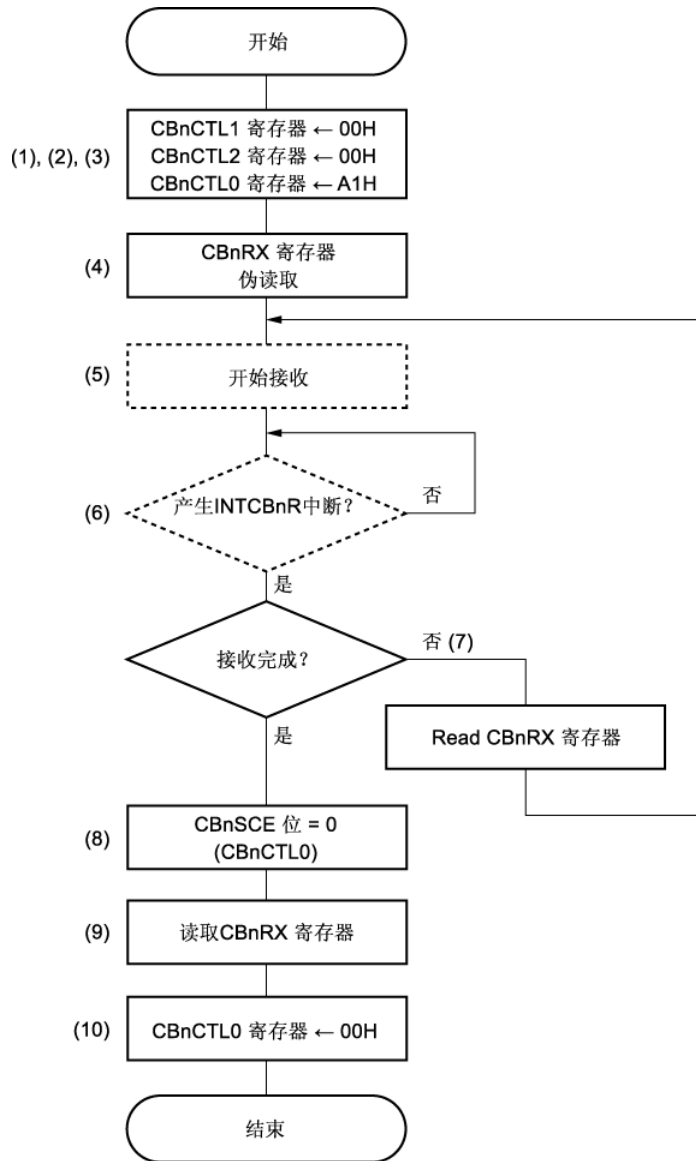
(2) 操作时序



<R> 16.6.2 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 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)

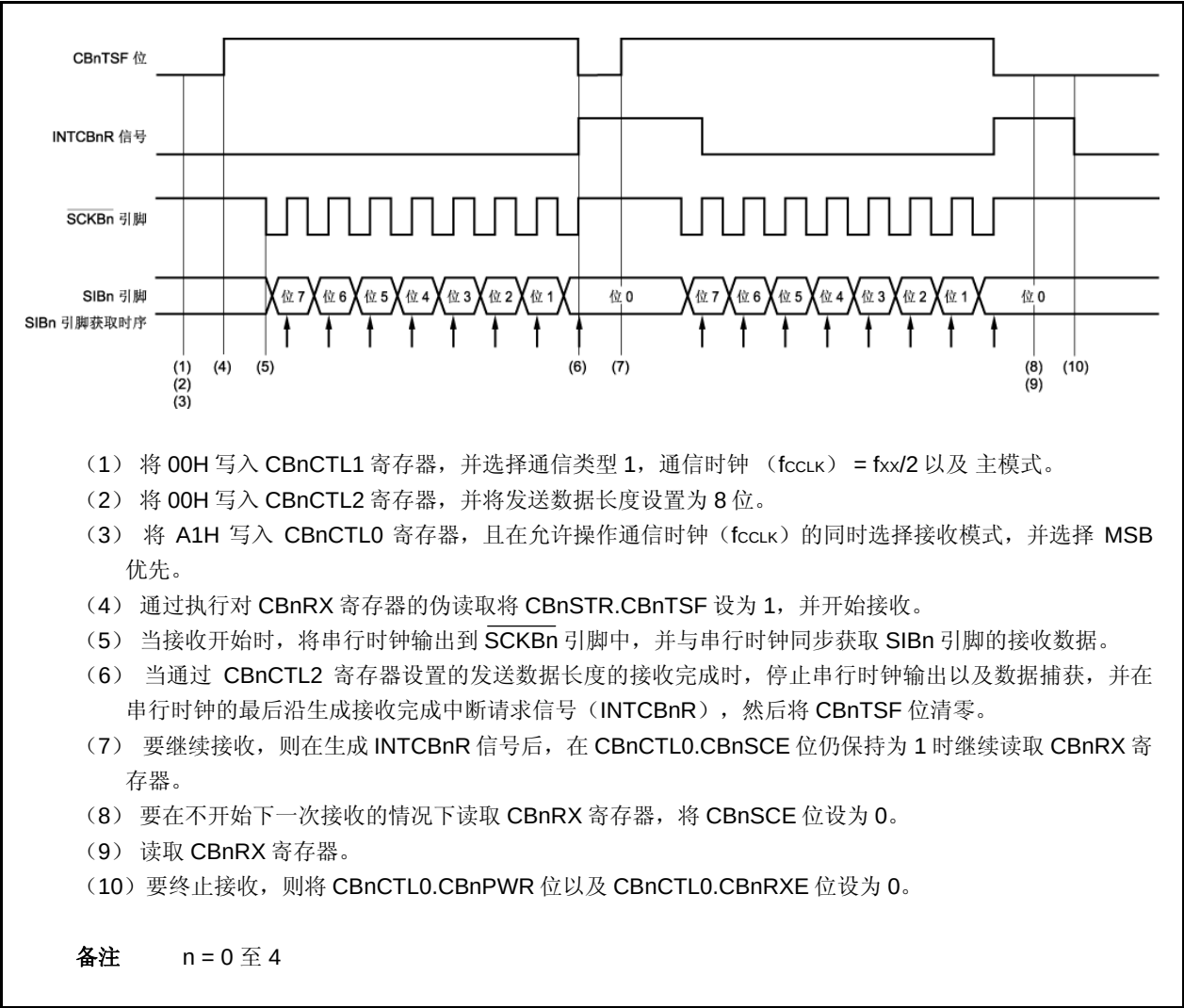
(1) 操作流程



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. $n = 0$ 至 4

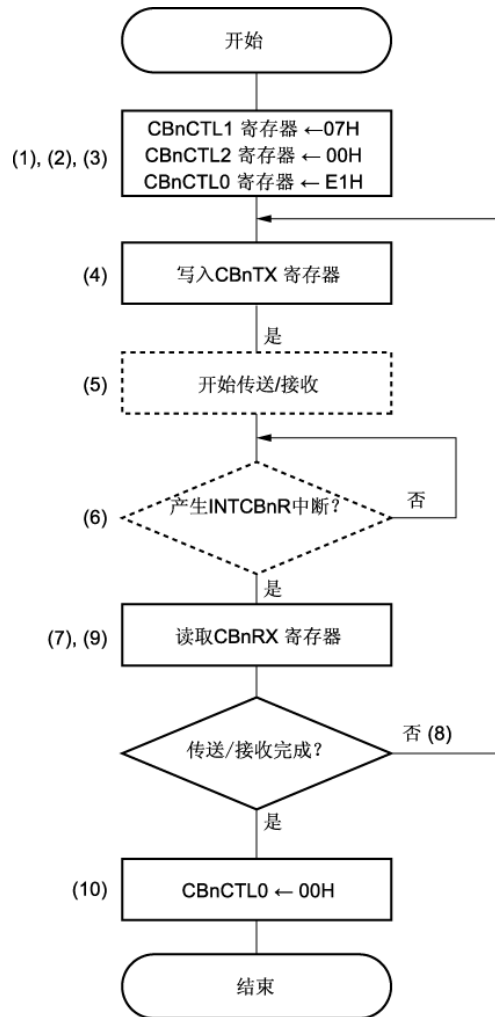
(2) 操作时序



<R> 16.6.3 单独发送模式（主模式，发送 / 接收模式）

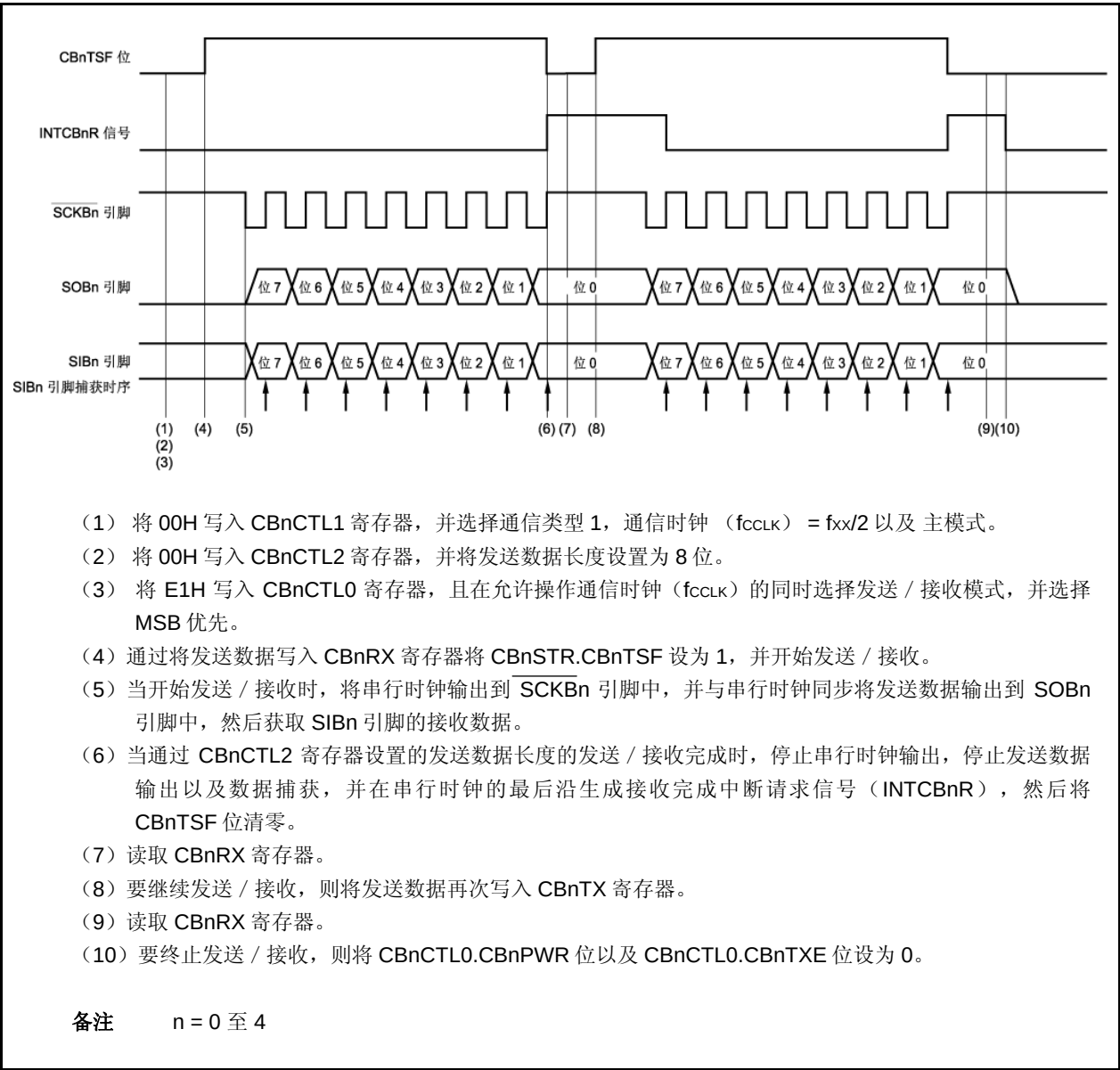
MSB 优先 (CBnCTL0.CBnDIR 位 = 0)，通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通信时钟 (f_{CCLK}) = $f_{xx}/2$ (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000)，发送数据长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

(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. $n = 0$ 至 4

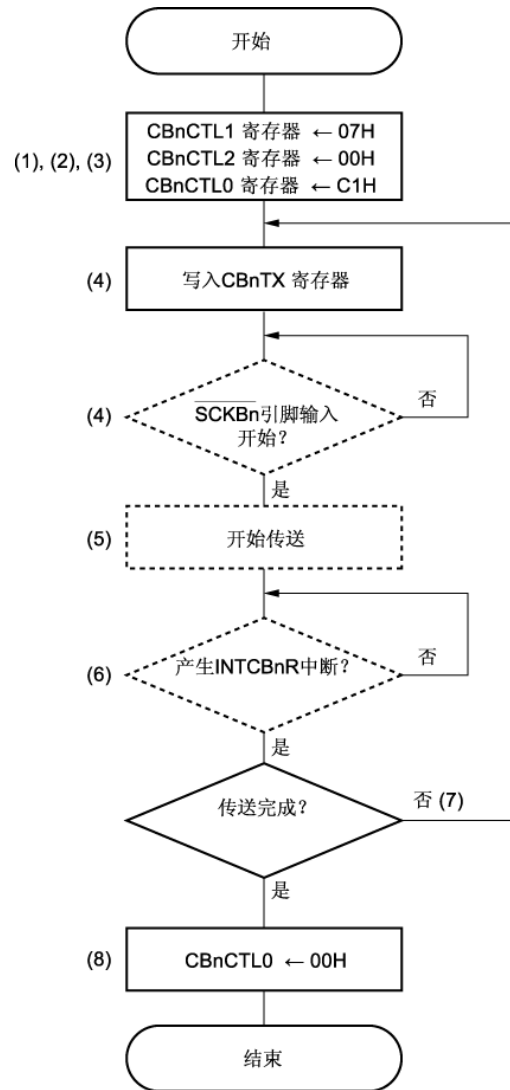
(2) 操作时序



<R> 16.6.4 单独发送模式 (从模式, 发送模式)

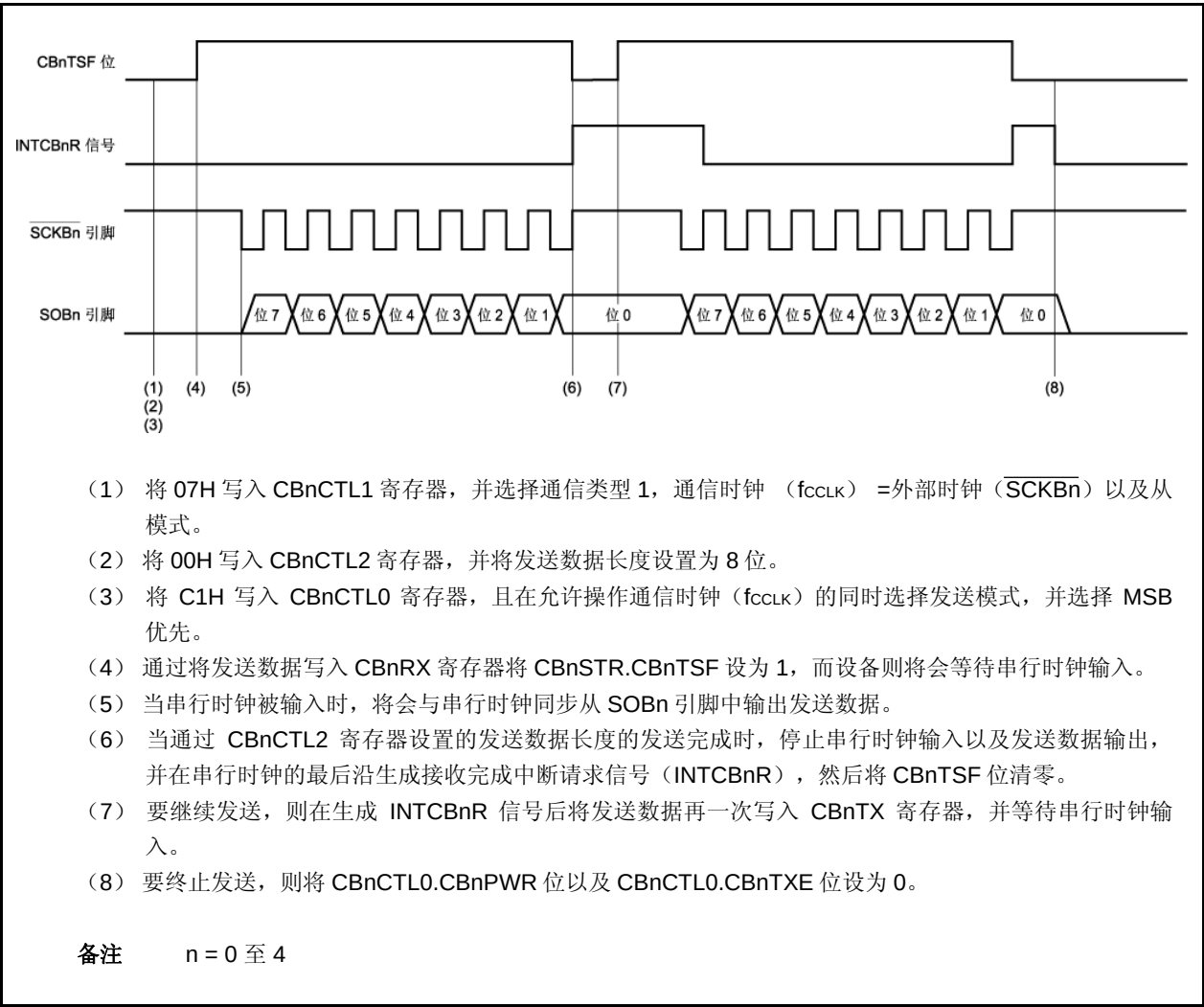
MSB 优先 (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = 外部时钟 ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

(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 至 4

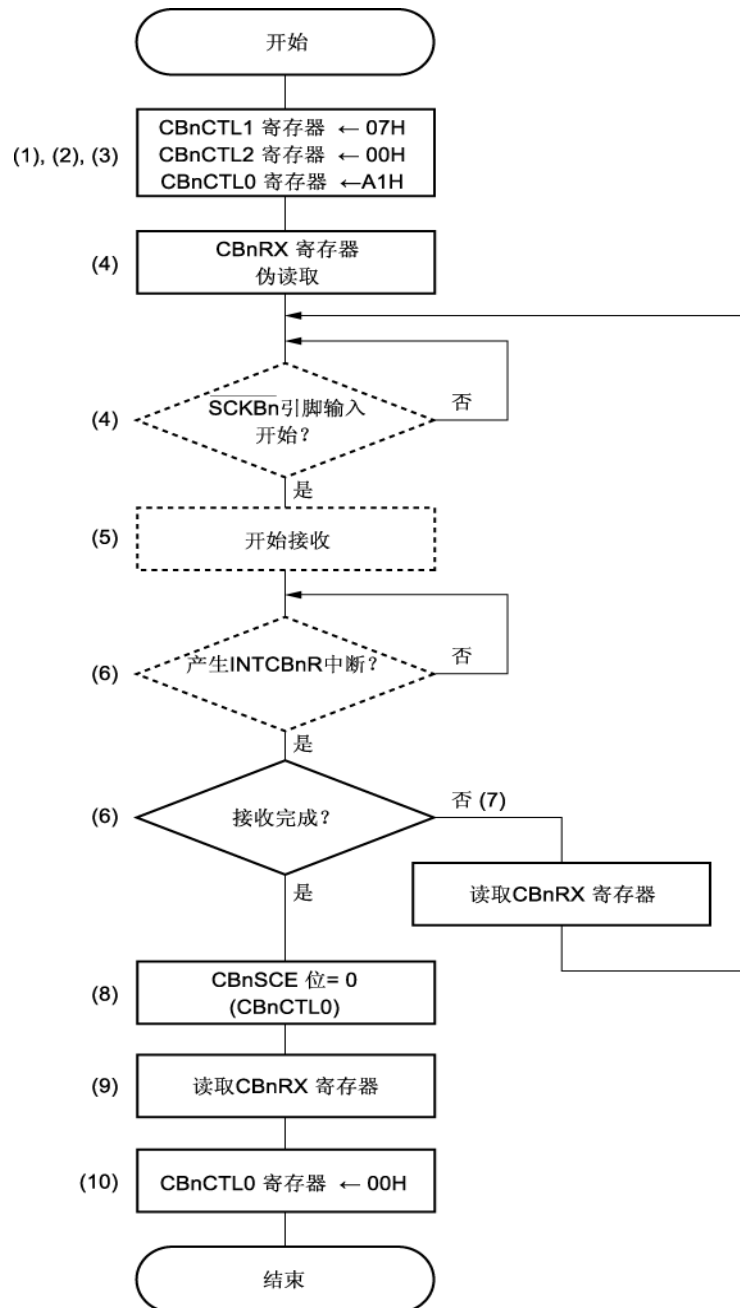
(2) 操作时序



<R> 16.6.5 单独发送模式（从模式，接收模式）

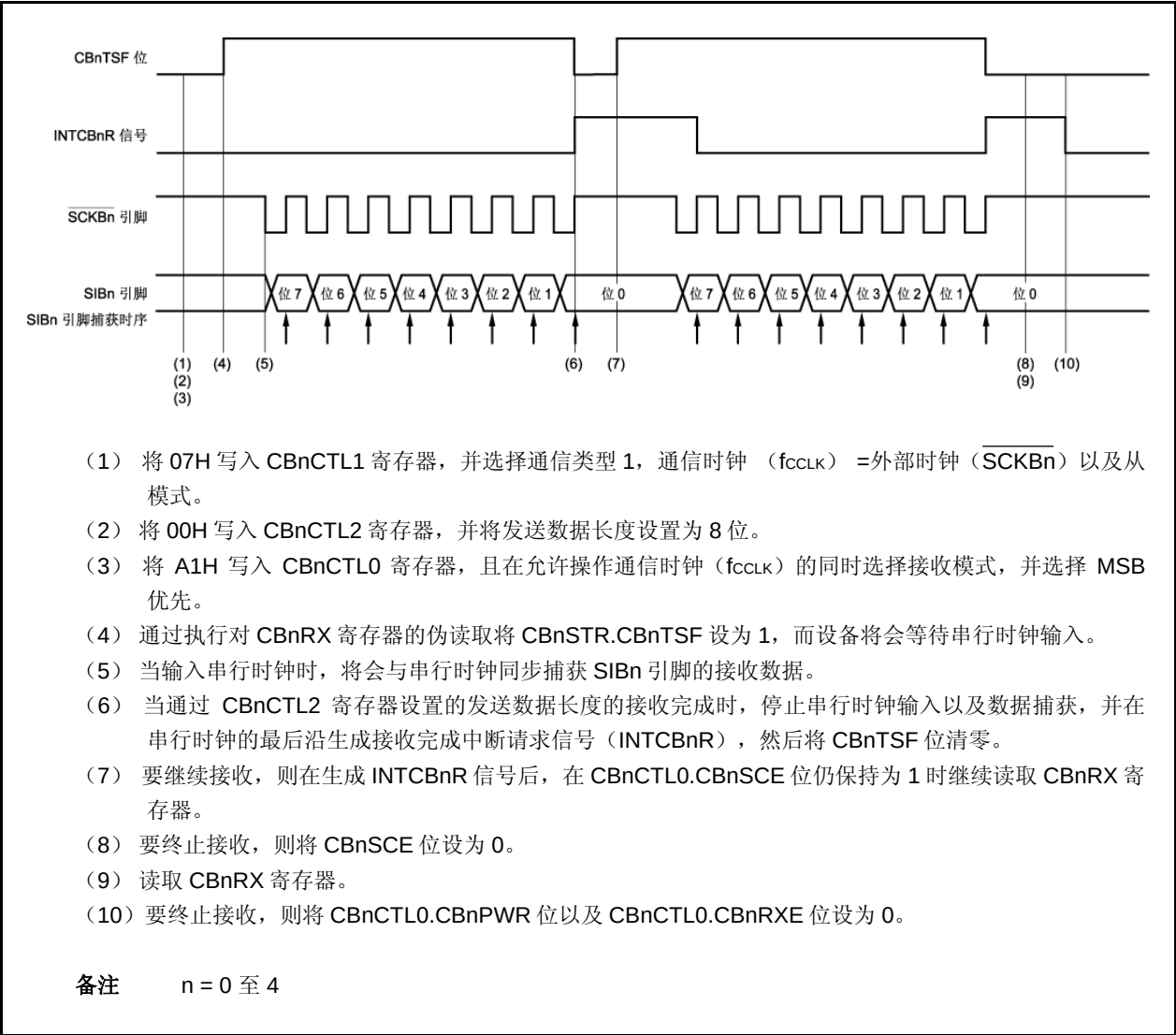
MSB 优先 (CBnCTL0.CBnDIR 位 = 0)，通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00)，通信时钟 (f_{CCLK}) = 外部时钟 ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111)，发送数据长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

(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 至 4

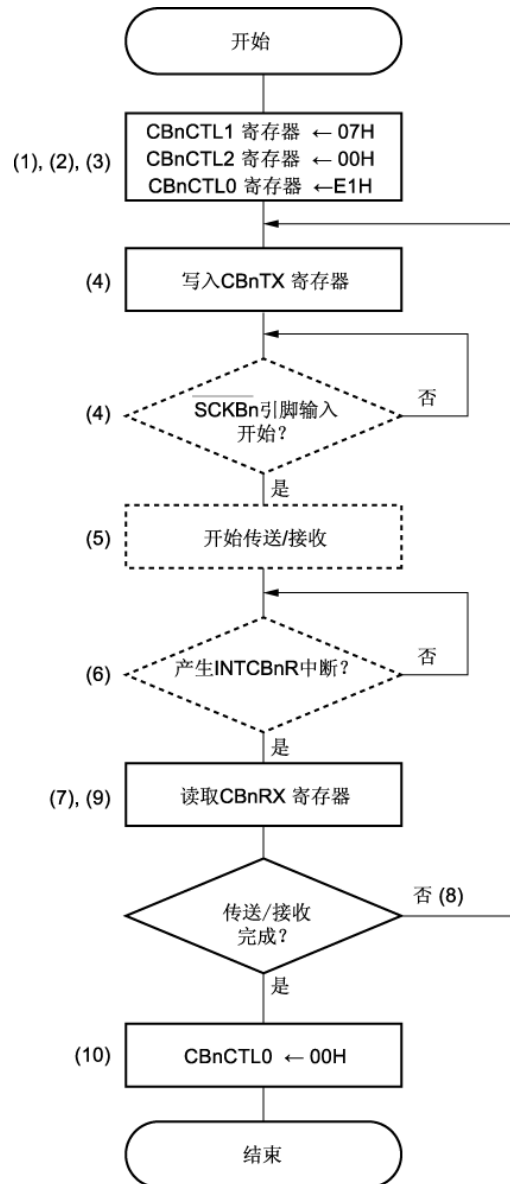
(2) 操作时序



<R> 16.6.6 单独发送模式 (从模式, 发送 / 接收模式)

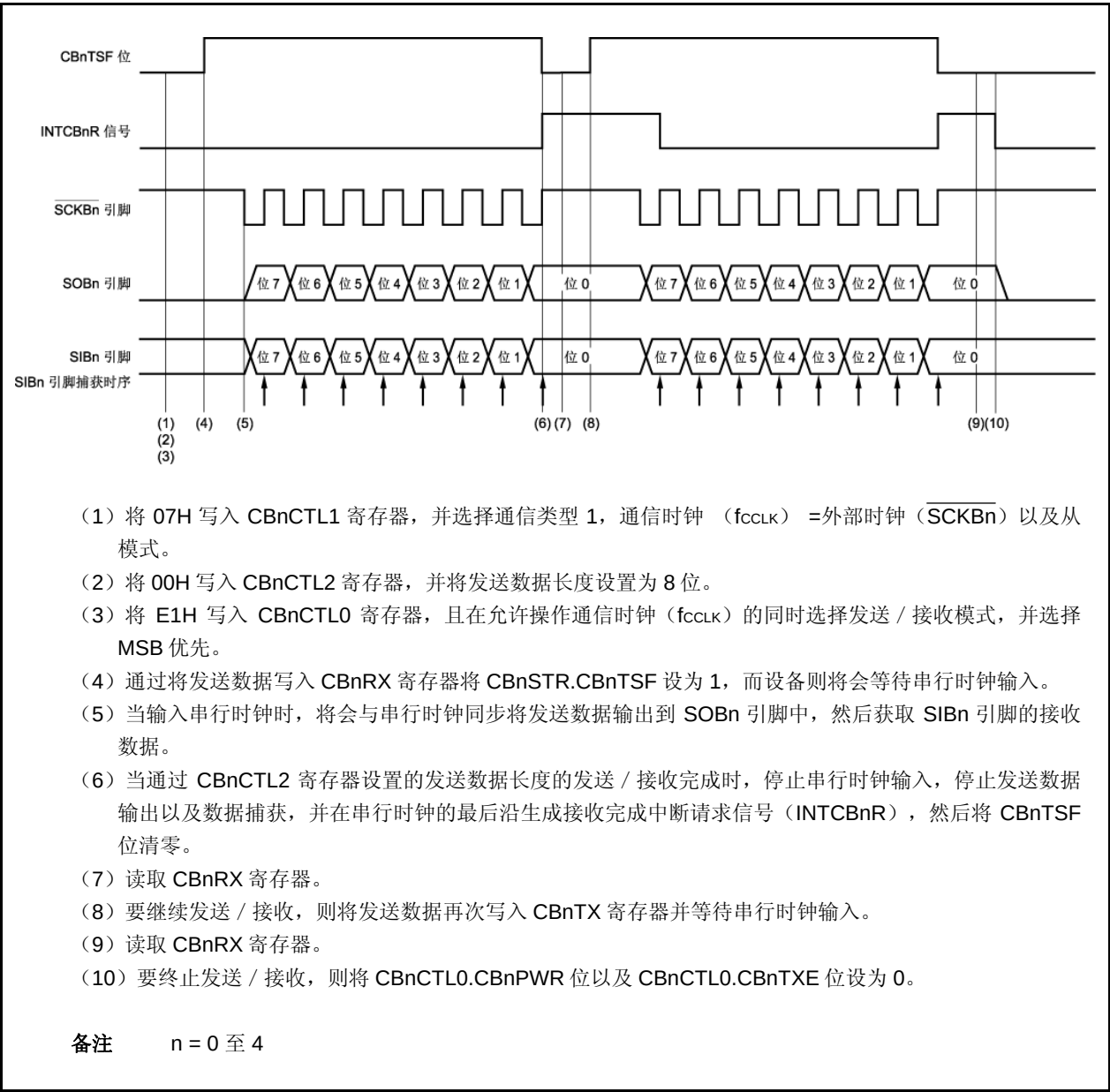
MSB 优先 (CBnCTL0.CBnDIR 位 = 0), 通信类型 1 (CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位 = 00), 通信时钟 (f_{CCLK}) = 外部时钟 ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

(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 至 4

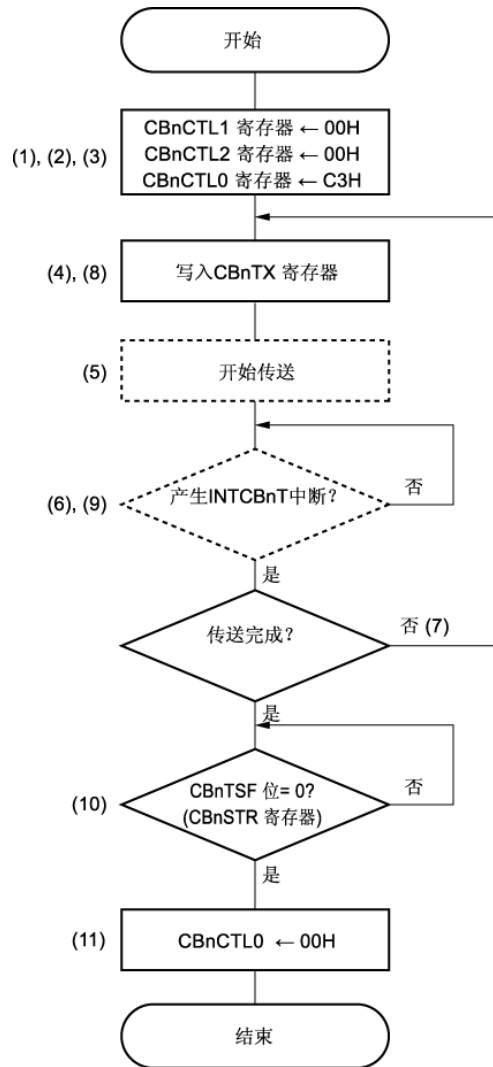
(2) 操作时序



<R> 16.6.7 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_{CLK}) = $f_{\text{xx}}/2$ (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000), 发送数据长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

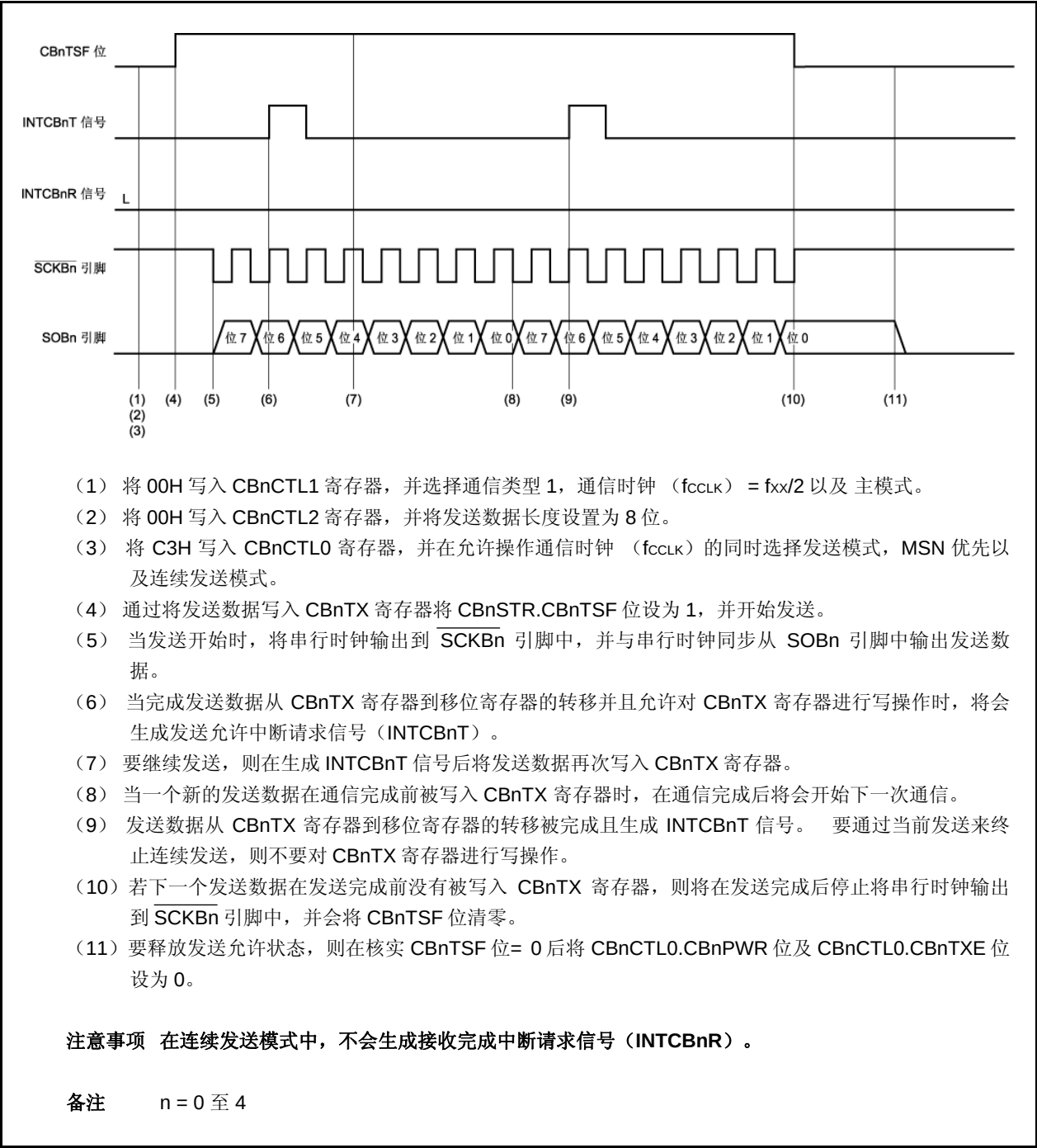
(1) 操作流程



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. $n = 0$ 至 4

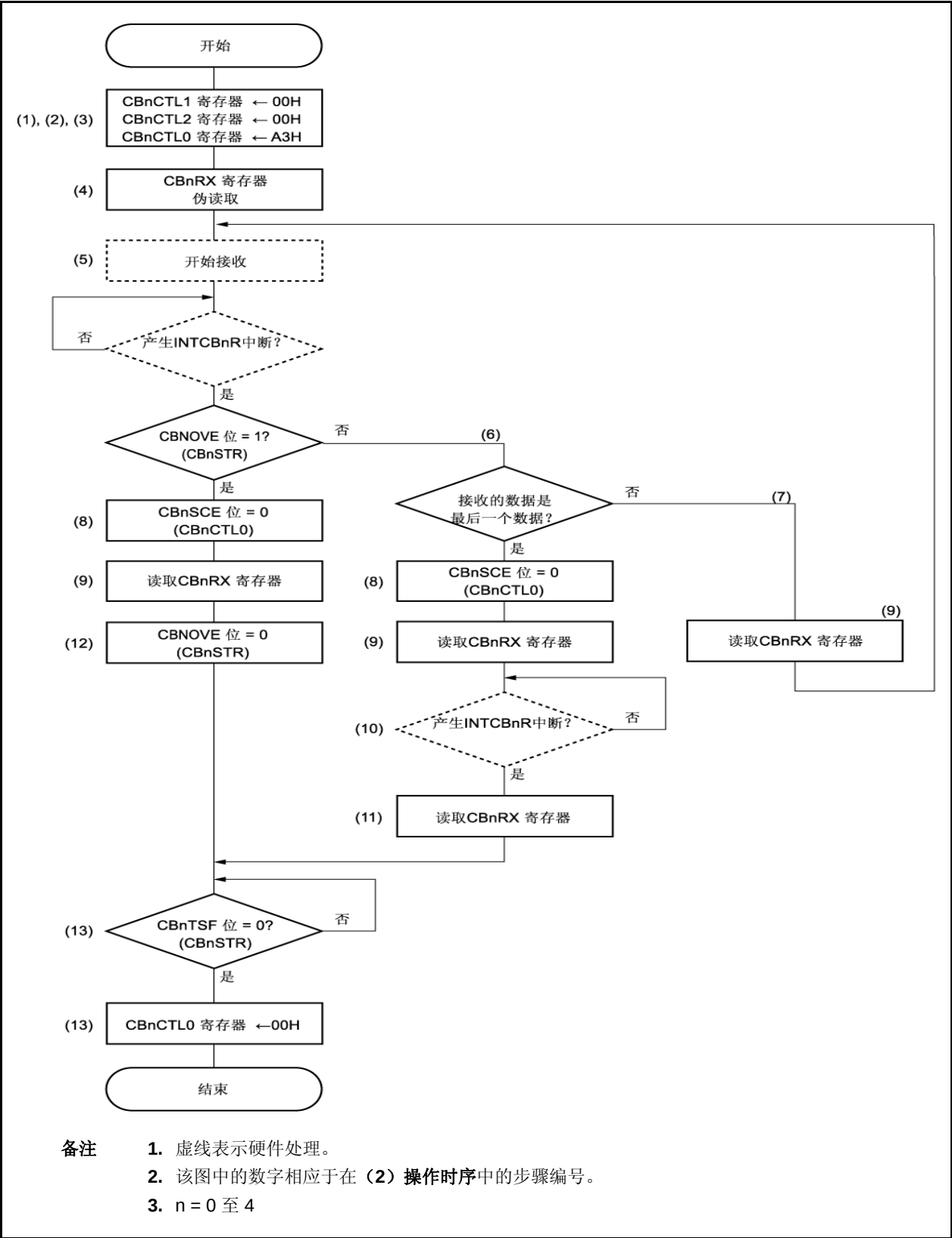
(2) 操作时序



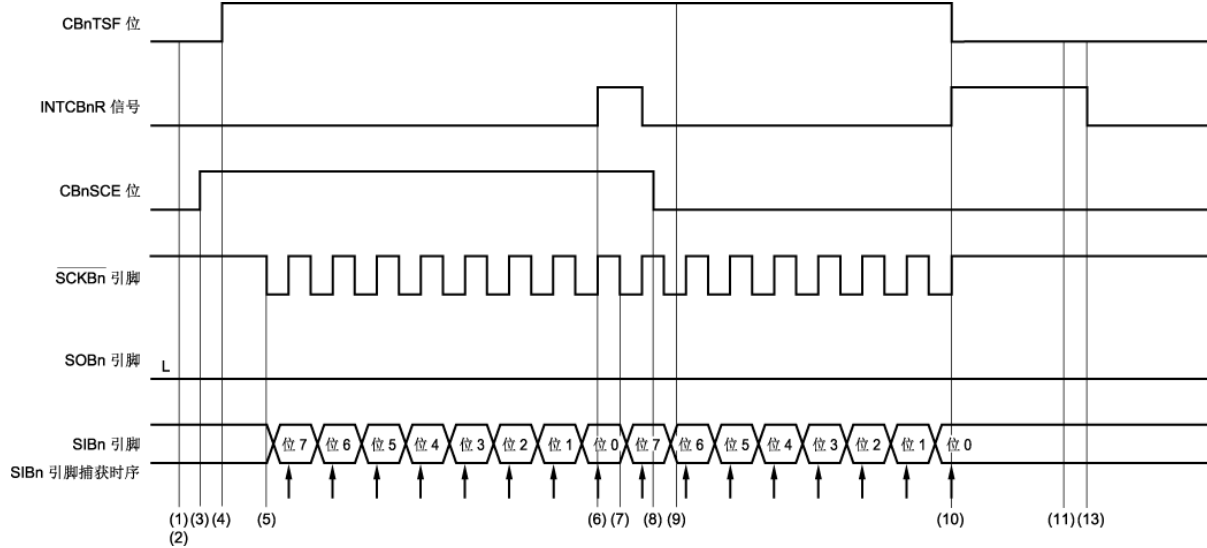
<R> 16.6.8 连续发送模式（主模式，接收模式）

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} ）= $f_{\text{xx}}/2$ （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 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 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 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 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。

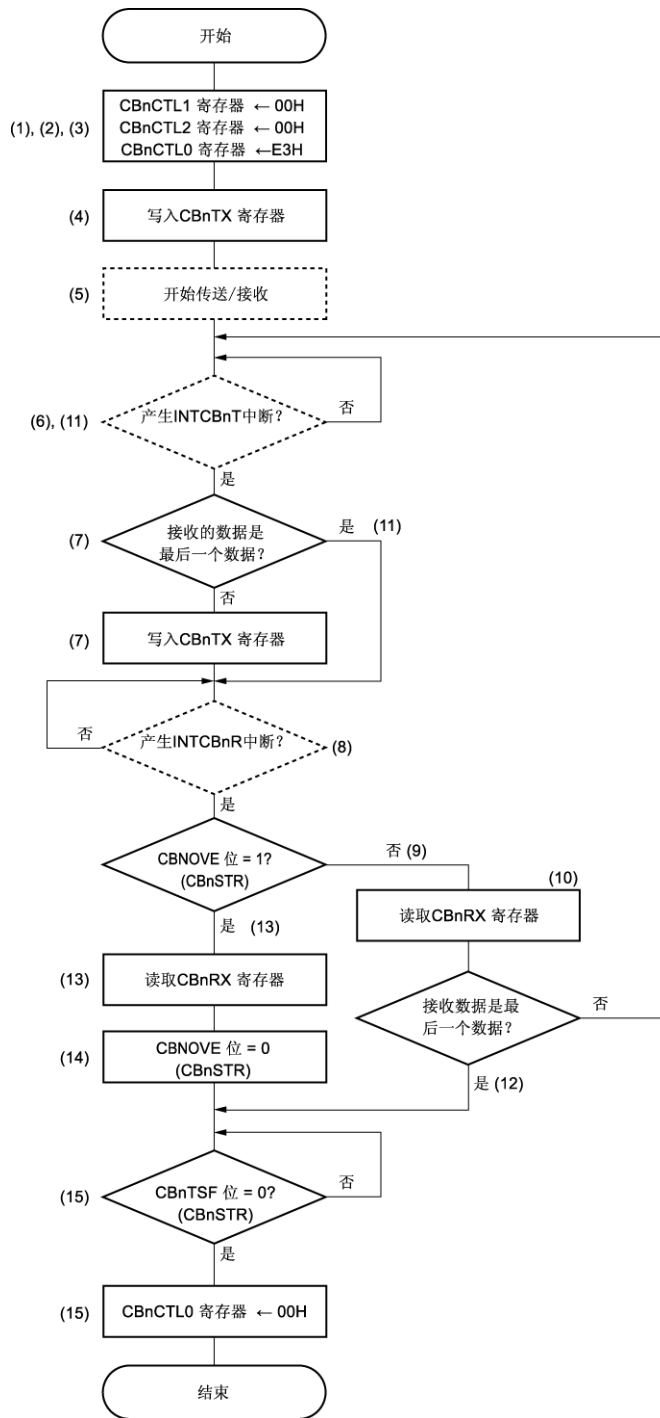
备注 n = 0 至 4

<R>

16.6.9 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_{CLK}) = $f_{\text{xx}}/2$ (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

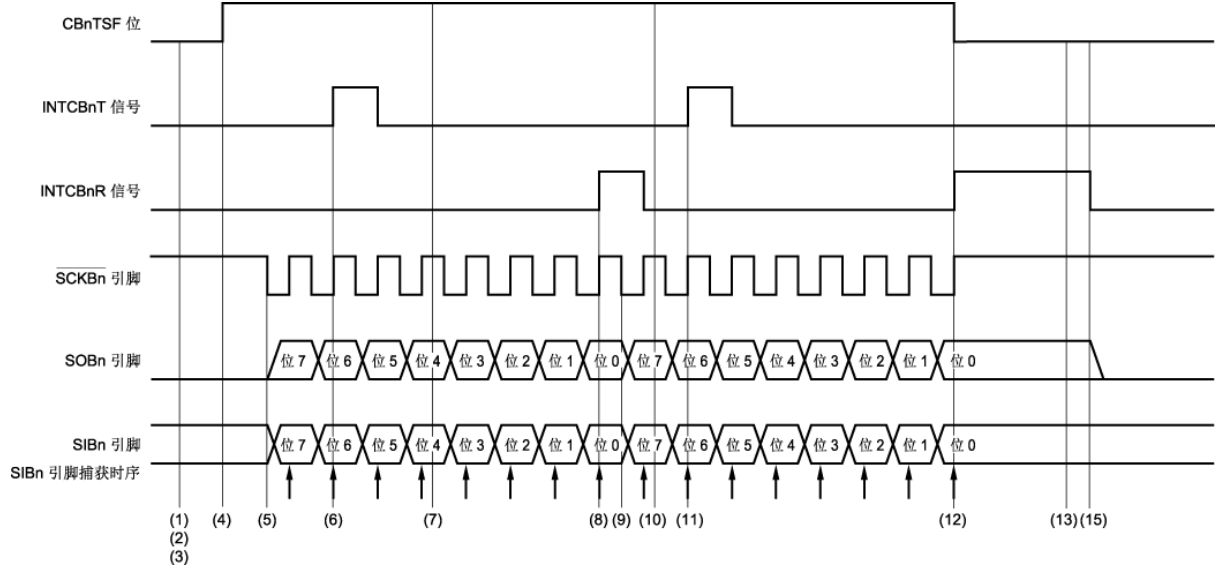


备注

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. $n = 0$ 至 4

(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_{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 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, 并开始发送 / 接收。
- (5) 当开始发送 / 接收时, 将串行时钟输出到 $\overline{SCKBn}$ 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 $n = 0$ 至 4

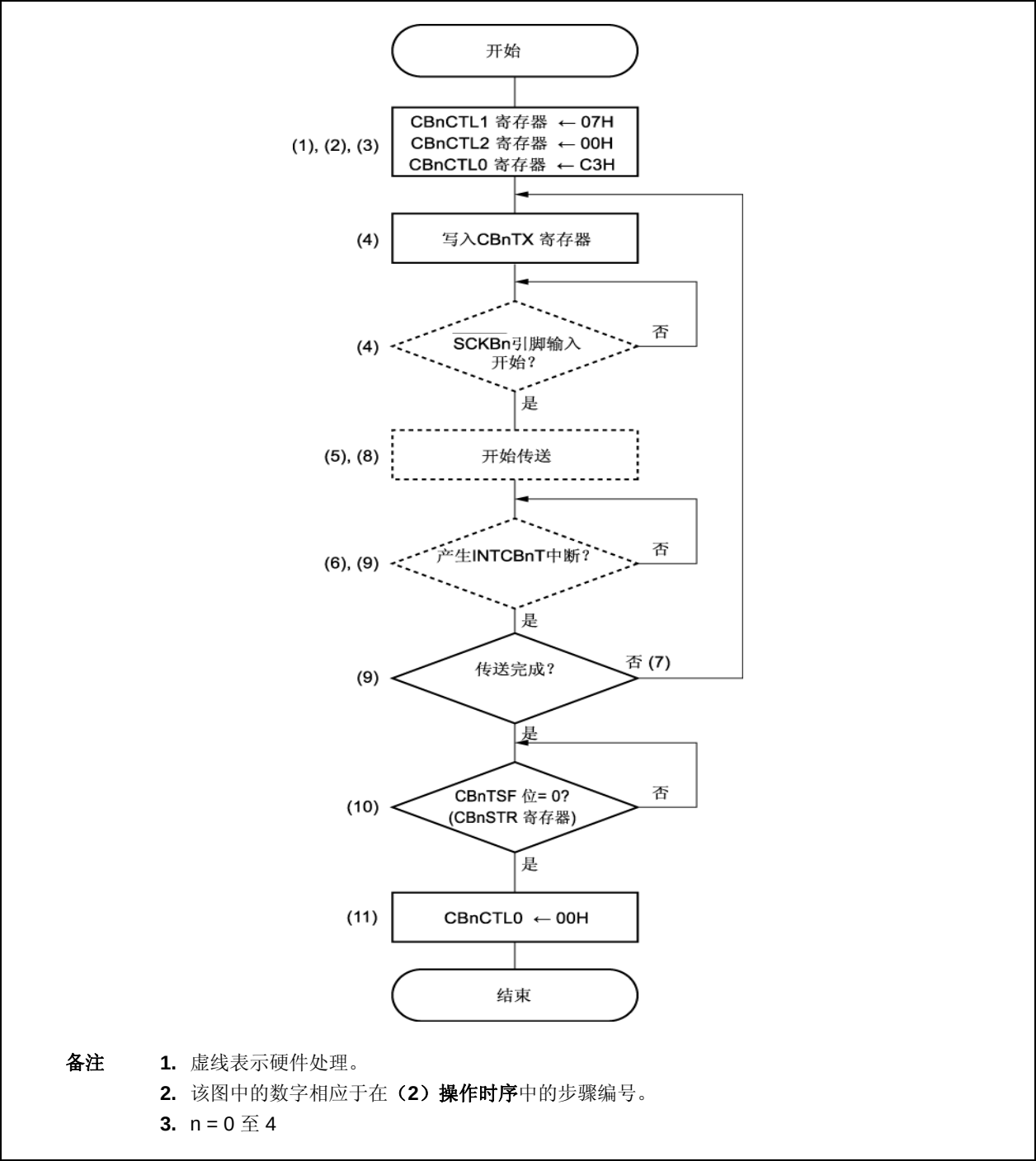
- (11) 发送数据从 CBnTX 寄存器到移位寄存器的转移被完成且生成 INTCBnT 信号。要通过当前发送 / 接收来结束连续发送 / 接收, 则不要对 CBnTX 寄存器进行写操作。
- (12) 若下一个发送数据在发送完成前没有被写入 CBnTX 寄存器, 则将在发送完成后停止将串行时钟输出到 SCKBn 引脚中, 并会将 CBnTSF 位清零。
- (13) 当接收错误中断请求信号 (INTCBnR) 被生成时, 将会读取 CBnRX 寄存器。
- (14) 如果发生溢出错误, 则将 CBnSTR.CBnOVE 位设为 0 并清除错误标志。
- (15) 要释放发送 / 接收允许状态, 则在核实 CBnTSF 位 = 0 后将 CBnCTL0.CBnPWR 位及 CBnCTL0.CBnTXE 位设为 0。

备注 n = 0 至 4

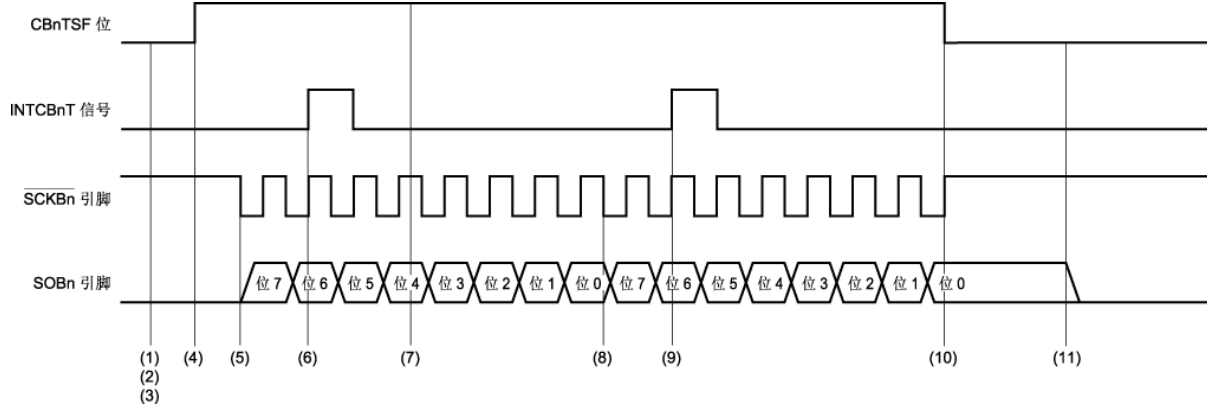
<R> 16.6.10 连续发送模式（从模式，发送模式）

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



(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 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) 当串行时钟被输入时, 将会与串行时钟同步从 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。
- (8) 在 CBnCTL2 寄存器所设置的发送数据长度的传输完成后, 若串行时钟被输入, 则将会开始连续发送。
- (9) 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。要通过当前发送来终止连续发送, 则不要对 CBnTX 寄存器进行写操作。
- (10) 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, 那么 CBnTSF 位将会被清 0 以终止发送。
- (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, 不会生成接收完成中断请求信号 (INTCBnR)。

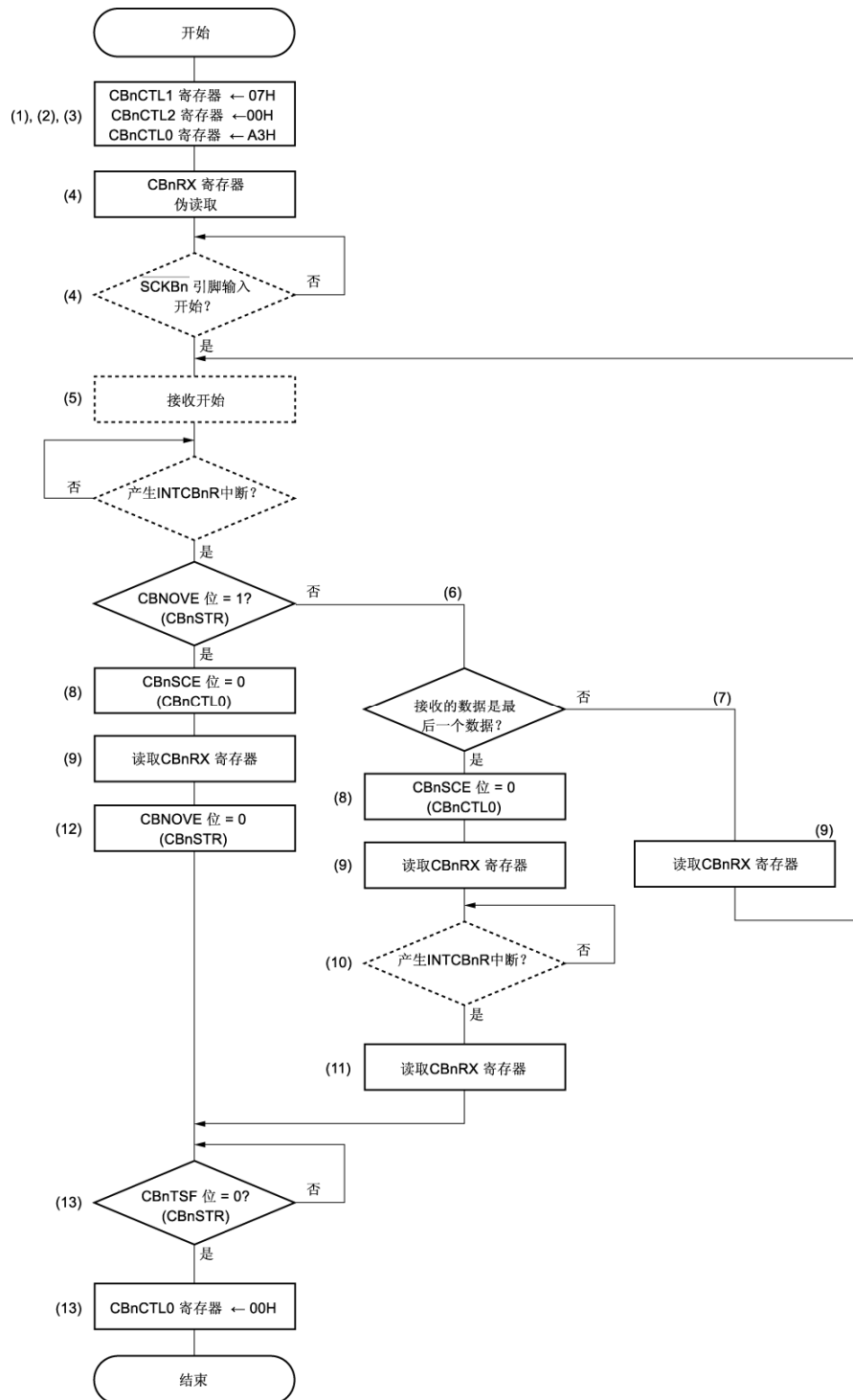
备注 n = 0 至 4

<R>

16.6.11 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_{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)

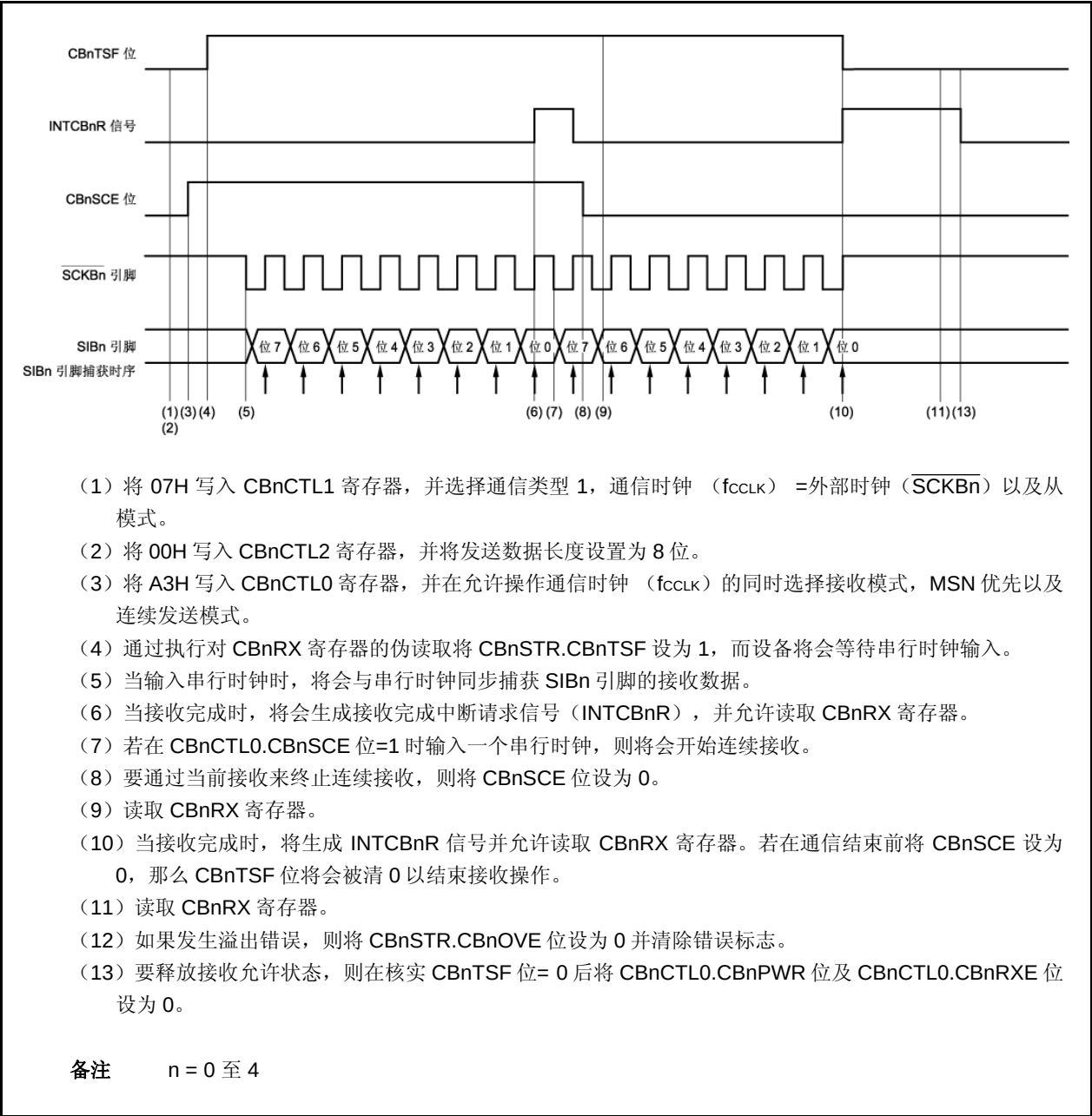
(1) 操作流程



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 至 4

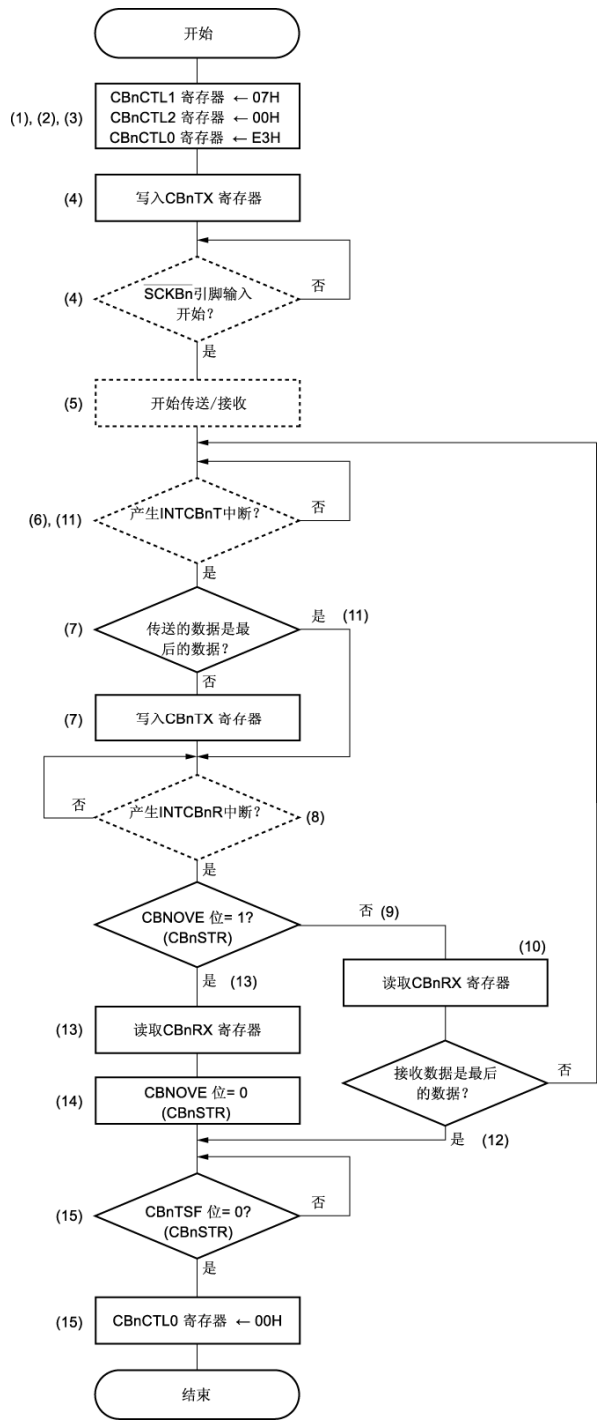
(2) 操作时序



<R> 16.6.12 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 (SCKBn) (CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111)，发送数据长度 = 8 位 (CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位 = 0000)

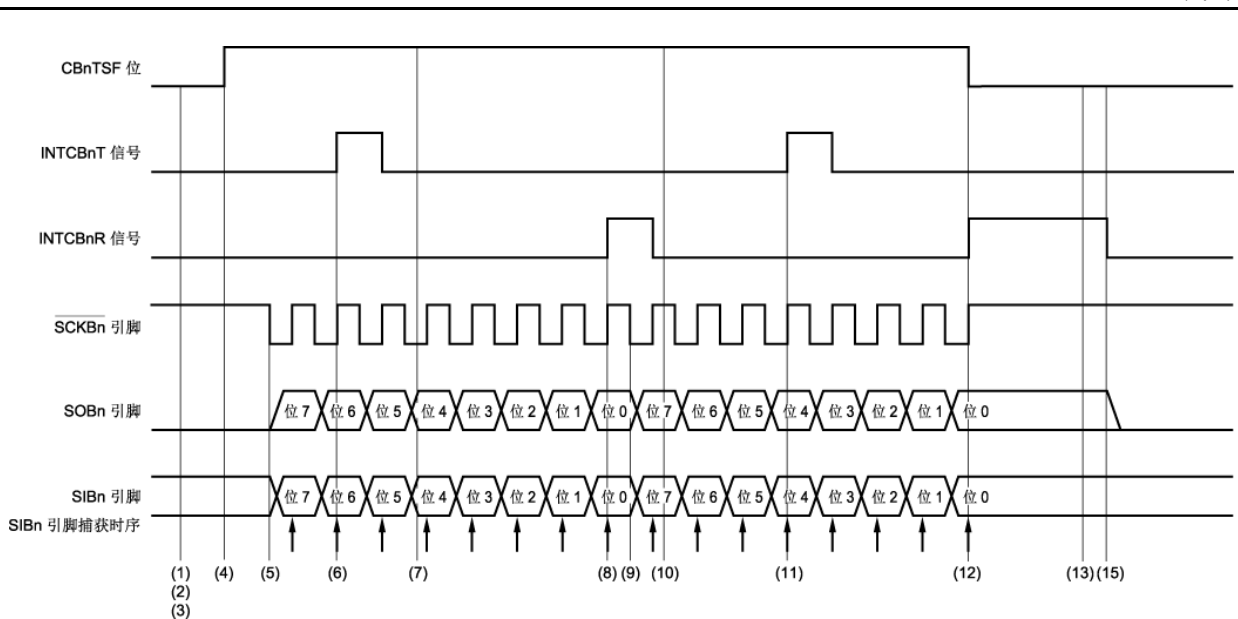
(1) 操作流程



- 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 至 4

(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 至 4

- (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。将 CBnTSF 位清 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。

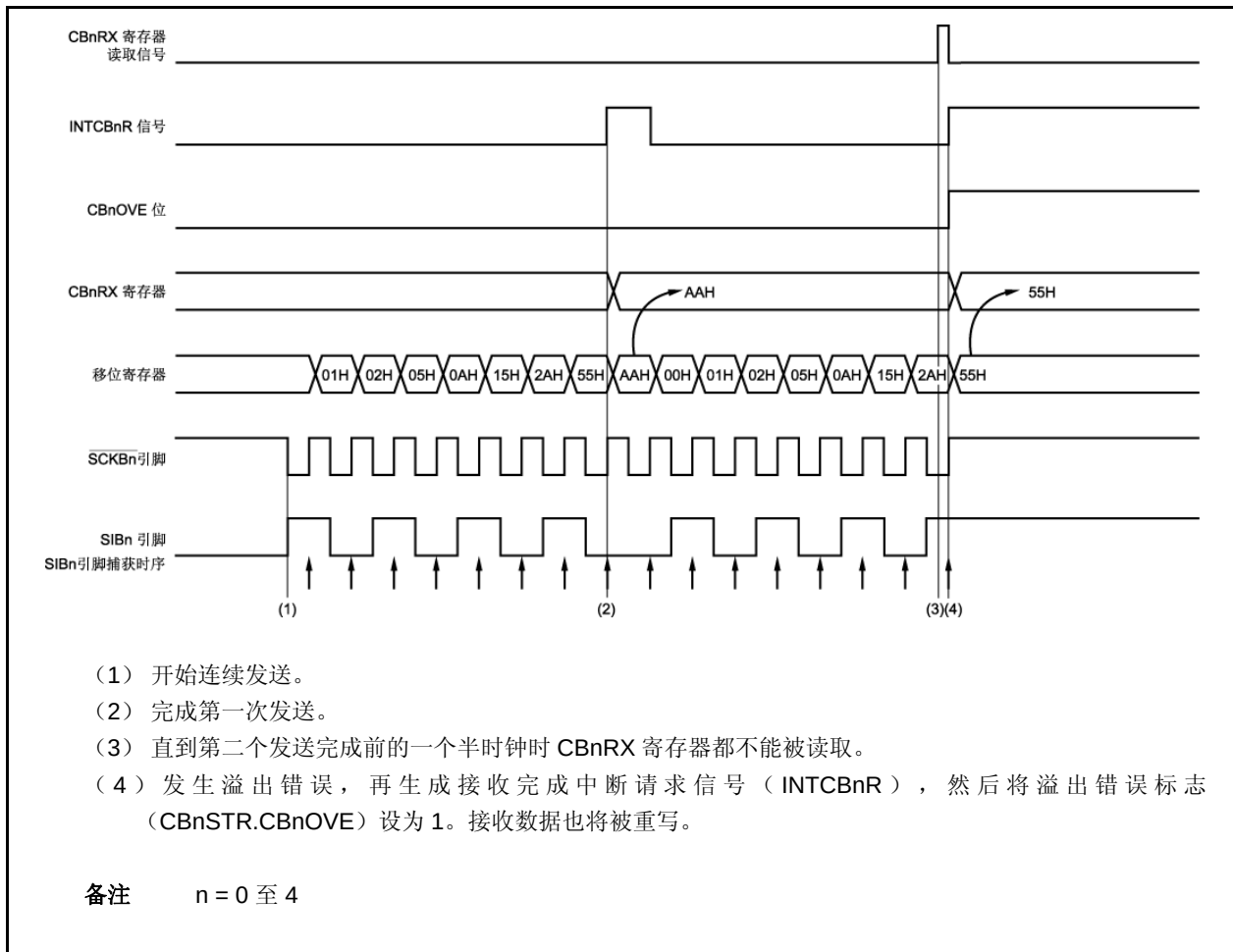
备注 n = 0 至 4

<R> 16.6.13 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。

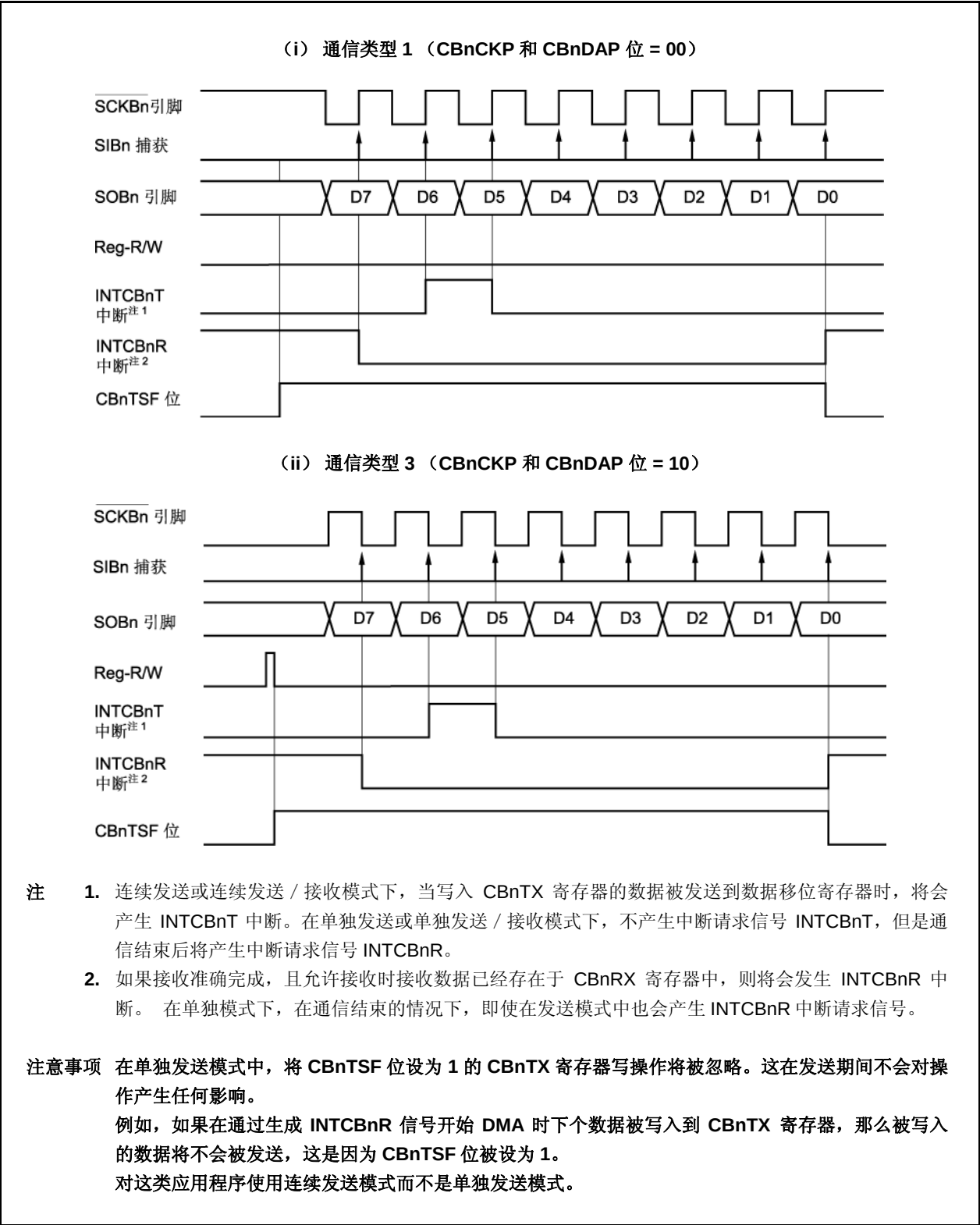
即使已经发生溢出错误，先前的接收数据也会因为 CBnRX 寄存器的更新而丢失。而即使已经发生接收错误，如果 CBnRX 寄存器没有被读取，在下一次接收完成的情况下也将会生成 INTCBnR 信号。

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。

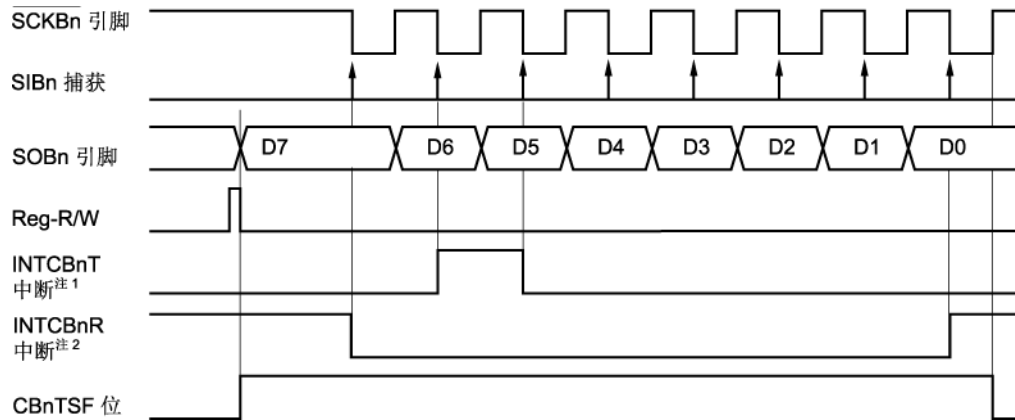
(1) 操作时序

16.6.14 时钟时序

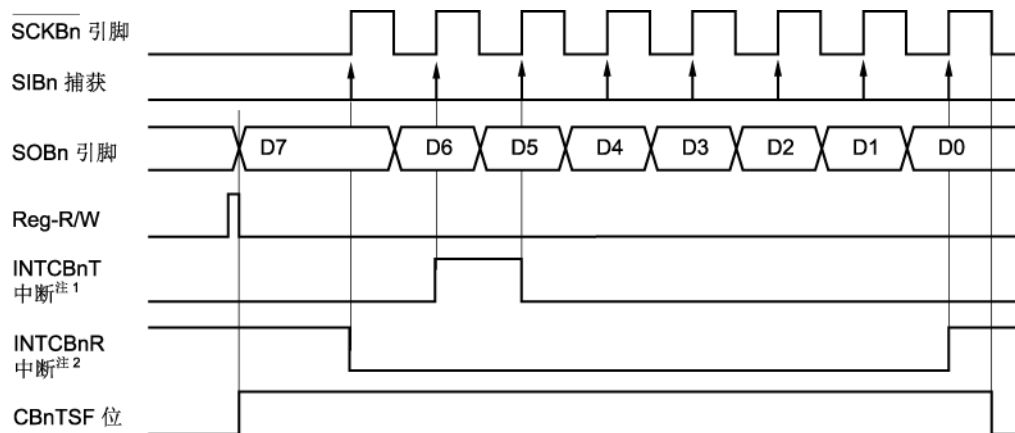
(1/2)



(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 CBnTSF 位设为 1 的 CBnTX 寄存器写操作将被忽略。这在发送期间不会对操作产生任何影响。

例如，如果在通过生成 INTCBnR 信号开始 DMA 时下个数据被写入到 CBnTX 寄存器，那么被写入的数据将不会被发送，这是因为 CBnTSF 位被设为 1。

对这类应用程序使用连续发送模式而不是单独发送模式。

16.7 输出引脚

(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。

CBnCKP	CBnCKS2	CBnCKS1	CBnCKS0	$\overline{\text{SCKBn}}$ 引脚输出
0	1	1	1	高阻抗
	除上述情况			固定于高电平
1	1	1	1	高阻抗
	除上述情况			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 至 4

(2) SOBn 引脚

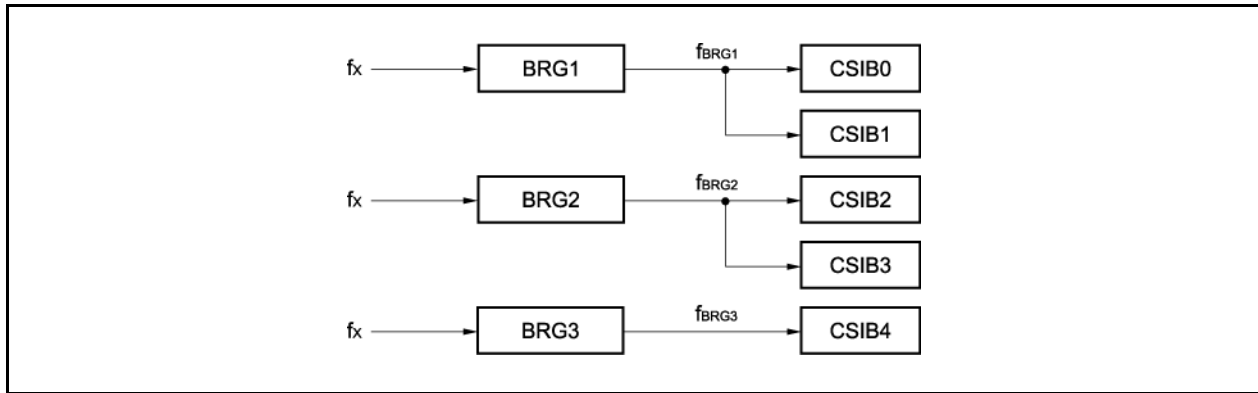
当 CSIBn 的操作被禁止 (CBnPWR 位 = 0) 时, SOBn 引脚输出状态如下所示。

CBnTXE	CBnDAP	CBnDIR	SOBn 引脚输出
0	x	x	固定于低电平
1	0	x	SOBn 锁存值 (低电平)
	1	0	CBnTX0 值 (MSB)
		1	CBnTX0 值 (LSB)

- 备注**
1. 如果 CBnCTL0.CBnTXE, 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. x: 不考虑
 3. n = 0 至 4

16.8 波特率发生器

如下列框图所示，连接了 BRG1 至 BRG3 以及 CSIB0 至 CSIB4 波特率发生器。



(1) BRGm 预分频模式寄存器 (PRSM)

PRSM1 至 PRSM3 寄存器控制产生 CSIB 的波特率信号。

这些寄存器可以以 8-位为单位或以 1-位为单位进行读取操作或写操作。

复位输入将把这些寄存器置为 00H。

复位后: 00H R/W 地址: PRSM1 FFFFF320H, PRSM2 FFFFF324H,
PRSM3 FFFFF328H

PRSMm (m = 1 至 3)	7	6	5	<4>	3	2	1	0
	0	0	0	BGCEm	0	0	BGCSm1	BGCSm0

BGCEm	波特率输出
0	禁止
1	允许

BGCSm1	BGCSm0	输入时钟选择 (f _{BGCSm})	设定值 (k)
0	0	f _{xx}	0
0	1	f _{xx} /2	1
1	0	f _{xx} /4	2
1	1	f _{xx} /8	3

- 注意事项
1. 不要在操作过程中对 PRSMm 寄存器改写。
 2. 在将 BGCEm 位设置为 1 之前设置寄存器 PRSMm。

(2) BRGm 预分频比较寄存器 m (PRSCMm)

PRSCM1 至 PRSCM3 寄存器是 8 位比较寄存器。
这些寄存器可以以 8 位为单位进行读取或写入。
复位输入将把这些寄存器置为 00H。

复位后： 00H R/W 地址： PRSCM1 FFFFF321H, PRSCM2 FFFFF325H,
PRSCM3 FFFFF329H

PRSCMm

7	6	5	4	3	2	1	0
PRSCMm7	PRSCMm6	PRSCMm5	PRSCMm4	PRSCMm3	PRSCMm2	PRSCMm1	PRSCMm0

注意事项 1. 不要在操作过程中对 PRSCMm 寄存器改写。
2. 在将 PRSMm.BGCEm 位设置为 1 之前设置寄存器 PRSCMm。

16.8.1 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。

$$f_{BRGm} = \frac{f_{xx}}{2^{k+1} \times N}$$

<R> 注意事项 将 fBRGm 设为 8 MHz 或更低。

备注 fBRGm: BRGm 计数时钟
fxx: 主时钟频率
k: PRSMm 寄存器设置值= 0 至 3
N: PRSCMm 寄存器设置值= 1 至 256
然而，只有当寄存器 PRSCMm 被设为 00H 时，N = 256。
m = 1 至 3

16.9 注意事项

- (1) 使用 DMA 传送来传送发送数据和接收数据时，即使在串行传送期间发生了溢出错误，也不能执行错误处理。在完成 DMA 传送后通过读取 CBnSTR.CBnOVE 位来核实没有发生溢出错误。
- (2) 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

- (3) 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。建议使用连续传送模式，特别是在使用 DMA 时。

备注 n = 0 至 4

第 17 章 I²C 总线

本章内容仅适用于内置 I²C 总线的版本（Y 产品）

若要使用 I²C 总线功能，则将 P38/SDA00，P40/SDA01，以及 P90/SDA02 引脚用作串行发送 / 接收数据 I/O 引脚，将 P39/SCL00，P41/SCL01，以及 P91/SCL02 引脚用作串行时钟 I/O 引脚，并将它们设为 N-ch 漏极开路输出。

17.1 I²C 总线和其它串行接口的模式切换

17.1.1 UARTA2 和 I²C00 模式的模式切换

<R> 在 V850ES/SG2 和 V850ES/SG2-H 中，UARTA2 和 I²C00 是同一引脚的复用功能，因此它们不能同时使用。使用之前预先用 PMC3 和 PFC3 寄存器设置 I²C00。

注意事项 如果在传送或接收过程中进行功能转换，那么 UARTA2 和 I²C00 的传送 / 接受操作无法确保。请务必禁止未使用的那个。

图 17-1. UARTA2 和 I²C00 的模式切换设置

复位后： 0000H R/W 地址： FFFFF446H, FFFFF447H

PMC3

15	14	13	12	11	10	9	8
0	0	0	0	0	0	PMC39	PMC38

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

复位后： 0000H R/W 地址： FFFFF466H, FFFFF467H

PFC3

15	14	13	12	11	10	9	8
0	0	0	0	0	0	PFC39	PFC38

7	6	5	4	3	2	1	0
PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

PMC3n	PFC3n	操作模式
0	×	端口I/O 模式
1	0	UARTA2 模式
1	1	I ² C00 模式

备注

1. n = 8, 9

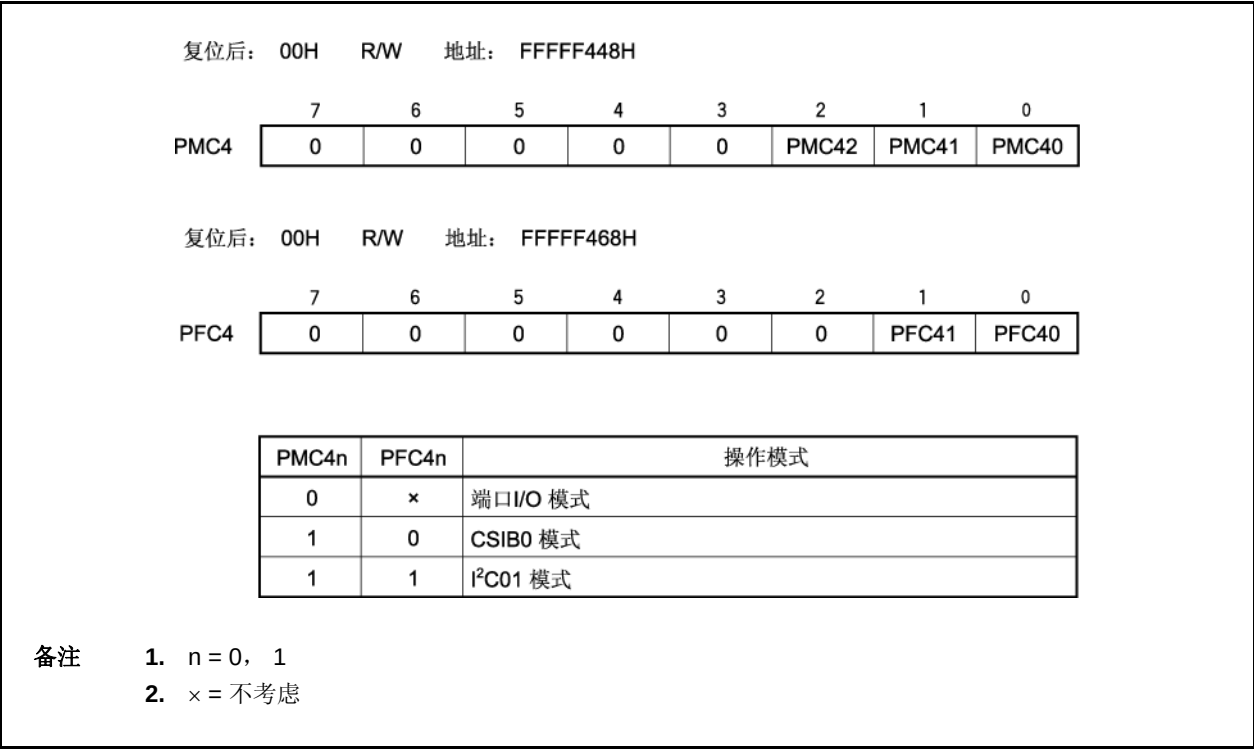
2. × = 不考虑

17.1.2 CSIB0 和 I²C01 的模式转换

<R> 在 V850ES/SG2 和 V850ES/SG2-H 中，CSIB0 和 I²C01 是同一引脚的复用功能，因此它们不能同时使用。使用之前预先用 PMC4 和 PFC4 寄存器设置 I²C01 。

注意事项 如果这些功能在传送或接收期间进行转换，那么 CSIB0 和 I²C01 的发送 / 接收操作将不能被保证。请务必禁止未使用的那个。

图 17-2. CSIB0 和 I²C01 的模式转换设置



17.1.3 UARTA1 和 I²C02 的模式切换

<R> 在 V850ES/SG2 和 V850ES/SG2-H 中，UARTA1 和 I²C02 是同一引脚的复用功能，因此它们不能同时使用。使用之前预先用 PMC9，PFC9，和 PMCE9 寄存器设置 I²C02 。

注意事项 如果在传送或接收过程中进行功能转换，那么 UARTA1 和 I²C02 的传送 / 接受操作无法确保。请务必禁止未使用的那个。

图 17-3. UARTA1 和 I²C02 的模式切换设置

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

PMC9

15	14	13	12	11	10	9	8
PMC915	PMC914	PMC913	PMC912	PMC911	PMC910	PMC99	PMC98
7	6	5	4	3	2	1	0
PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

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

PFC9

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

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

PFCE9

15	14	13	12	11	10	9	8
PFCE915	PFCE914	0	0	0	0	0	0
7	6	5	4	3	2	1	0
PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

PMC9n	PFCE9n	PFC9n	操作模式
1	1	0	UARTA1 模式
1	1	1	I ² C02 模式

备注 n = 0, 1

用户手册 U16541CA5V1UD

595

17.2 特点

I²C00 到 I²C02 有下列两种模式。

- 操作停止模式
- I²C（内部 IC）总线模式（支持多主模式）

（1）操作停止模式

这种模式下，不会进行串行传输，因此能够降低了能耗。

（2）I²C 总线模式（支持多主模式）

该模式用于通过两种线路使用多个设备执行的 8 位数据传送：一个串行时钟引脚（SCL0n）和一个串行数据总线引脚（SDA0n）。

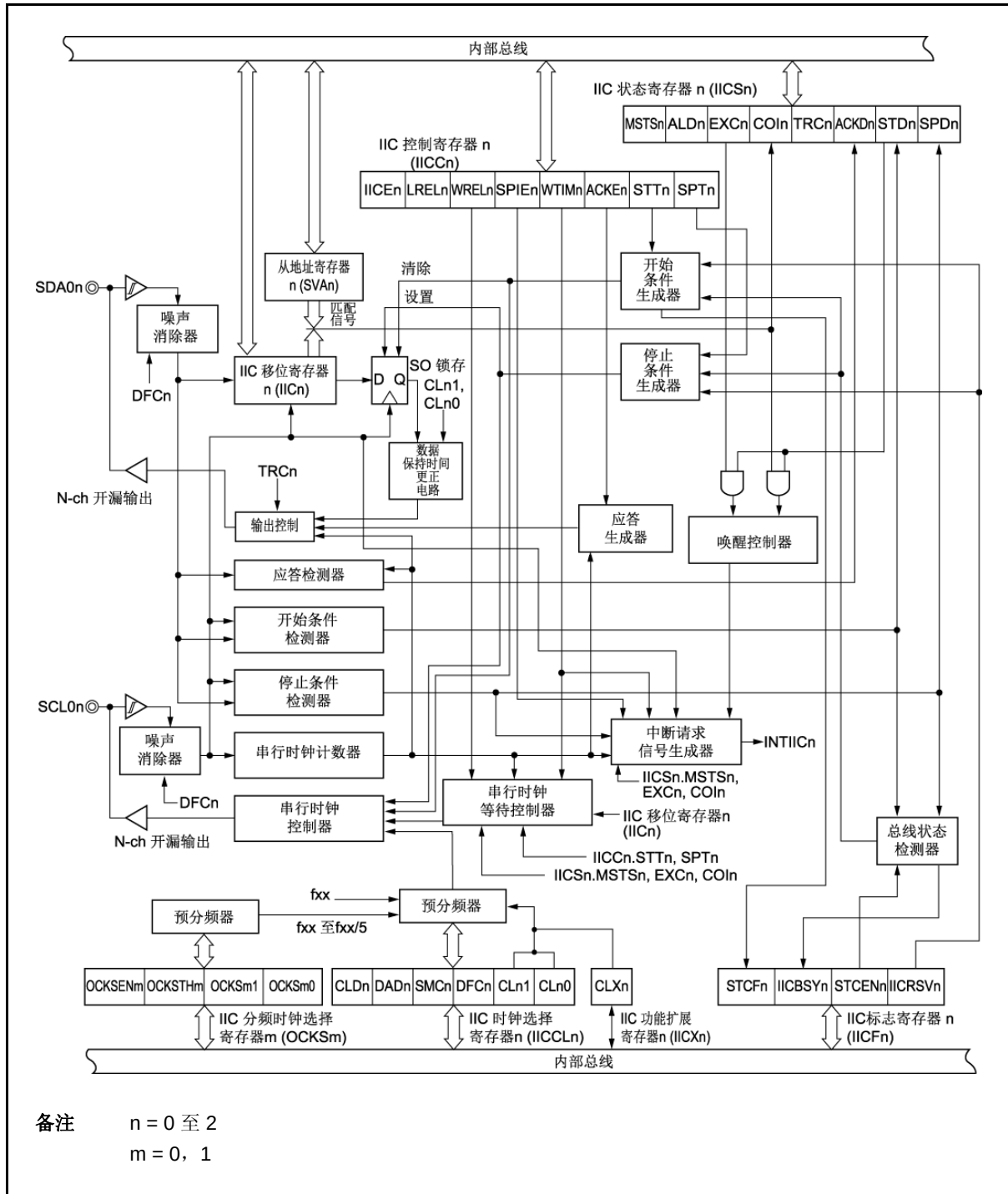
这种模式遵循 I²C 总线格式，并且主设备能通过串行数据总线产生“启动条件”，“地址”，“传送方向”，“数据”以及“停止条件”数据给从设备。从设备通过硬件自动检测接收状态和数据。这个功能可以简化控制 I²C 总线部分的应用程序。

由于 SCL0n 和 SDA0n 引脚用作 N 沟道开漏输出，因此对于串行时钟线和串行数据总线，I²C0n 需要一个上拉电阻。

备注 n = 0 至 2

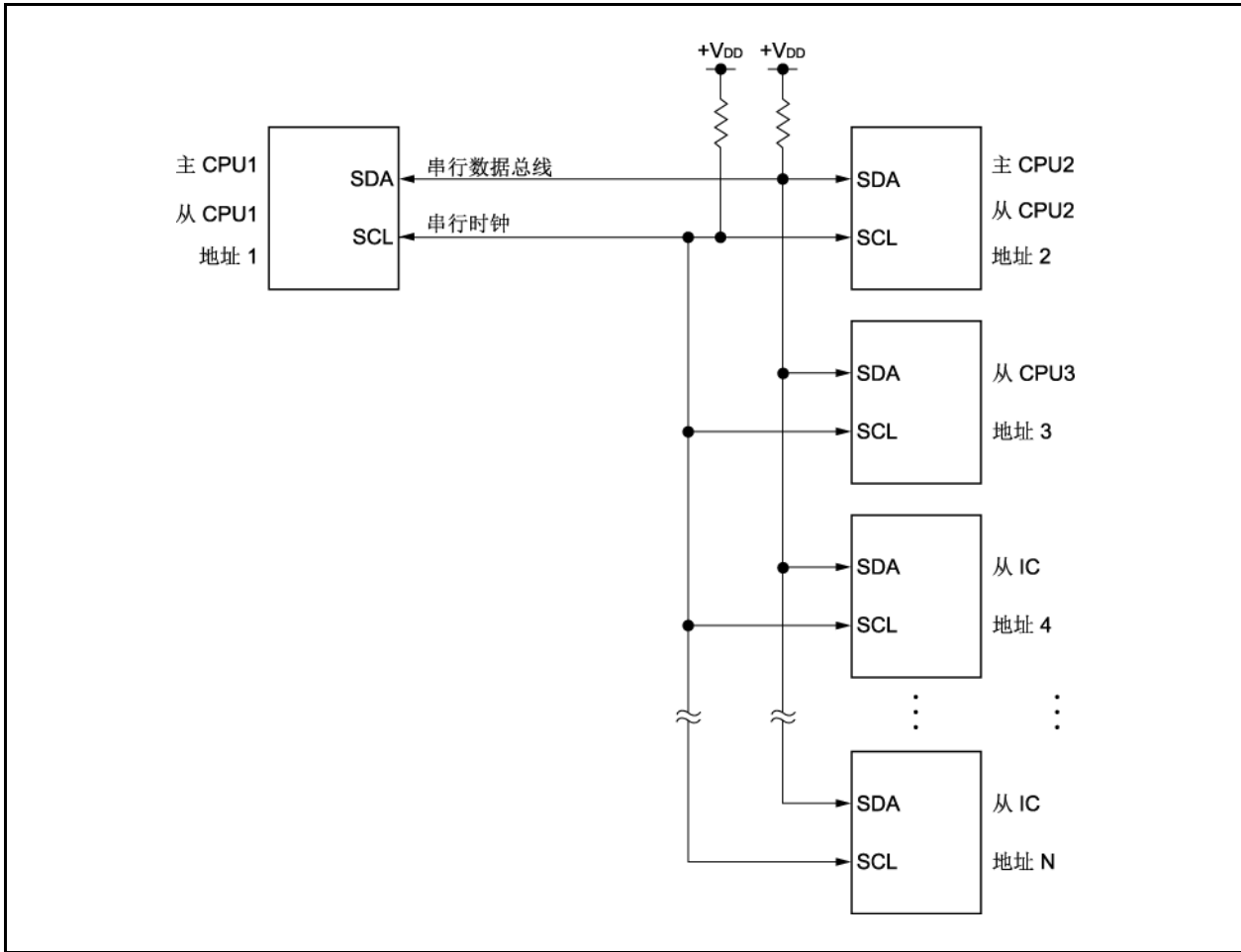
17.3 配置

I²C0n 结构图如下所示。

图 17-4. I²C0n 结构框图

以下显示了一个串行总线配置的实例。

图 17-5. 使用 I²C Bus 的串行总线配置实例



I²C0n 包括下列硬件（n = 0 到 2）。

表 17-1. I²C0n 的配置

项目	配置
寄存器	IIC 移位寄存器 n（IICn） 从地址寄存器 n（SVAn）
控制寄存器	IIC 控制寄存器 n（IICCn） IIC 状态寄存器 n（IICSn） IIC 标志寄存器 n（IICF0n） IIC 时钟选择寄存器 n（IICCLn） IIC 功能扩展寄存器 n（IICXn） IIC 时钟分频选择寄存器 0, 1（OCKS0, OCKS1）

(1) IIC 移位寄存器 n（IICn）

IICn 寄存器把 8 位串行数据转换成 8 位并行数据，反之亦然，并且传送和接收中都能使用（n = 0 到 2）。
对 IICn 寄存器的读操作和写操作用来控制实际的传送和接收操作。
该寄存器可以以 8 位为单位进行读取或写入。
复位输入将该寄存器清除为 00H。

(2) 从地址寄存器 n（SVAn）

SVAn 寄存器从模式时设置本地地址（n = 0 到 2）。
该寄存器可以以 8 位为单位进行读取或写入。
复位输入将该寄存器清除为 00H。

(3) SO 锁存

SO 锁存用来保留 SDA0n 引脚的输出电平（n = 0 到 2）。

(4) 唤醒控制器

由该寄存器接收的地址与设置在 SVAn 中的地址值相匹配或接收到扩展码时，这个电路会产生一个中断请求（INTIICn）（n = 0 到 2）。

(5) 分频器

分频器用于选择使用的采样时钟。

(6) 串行时钟计数器

这个计数器在传送和接收操作期间对输出的串行时钟和输入的串行时钟进行计数，并校验 8 位数据是被传送还是接收。

(7) 中断请求信号发生器

该电路控制中断请求信号 (INTIICn) 产生。

在满足下列任意一个触发器的条件时, 发生 I²C 中断。

- 串行时钟的第 8 个或第 9 个时钟的下降沿 (由 IICn.WTIMn 位设置)
- 由于满足停止条件检测发生的中断 (由 IICn.SPIEn 位设置)

备注 n = 0 至 2

(8) 串行时钟控制器

主模式下, 这个电路通过 SCL0n 引脚从采样时钟产生时钟输出 (n = 0 到 2)。

(9) 串行时钟等待控制器

该电路控制等待时序。

(10) $\overline{\text{ACK}}$ 发生器、停止条件检测器、启动条件检测器和 $\overline{\text{ACK}}$ 检测器。

这些电路用来产生和检测各种状态。

(11) 数据保持时间修正电路

该电路产生相应于 SCL0n 引脚下降沿的数据的保持时间。

(12) 启动条件发生器

当 IICn.STTn 位设置时产生启动条件。

然而, 在通信保留禁止状态下 (IICFn.IICRSVn 位 = 1), 这个请求被忽略, 并且如果总线不释放 (IICFn.IICBSYn 位 = 1), 那么 IICFn.STCFn 位设置为 1。

(13) 停止条件发生器

当 IICn.SPTn 位被设为 (1) 时产生停止条件。

(14) 总线状态检测器

通过检测启动条件和停止条件确定总线是否释放。

然而, 总线状态不能在操作后立即检测到, 因此通过使用 IICFn.STCENn 位将总线状态检测器设置为初始状态。

17.4 寄存器

I²C00 到 I²C02 由下列寄存器控制。

- IIC 控制寄存器 0 到 2 (IICC0 到 IICC2)
- IIC 状态寄存器 0 到 2 (IICS0 到 IICS2)
- IIC 标志寄存器 0 到 2 (IICF0 到 IICF2)
- IIC 时钟选择寄存器 0 到 2 (IICCL0 到 IICCL2)
- IIC 功能扩展寄存器 0 到 2 (IICX0 到 IICX2)
- IIC 分频时钟选择寄存器 0, 1 (OCKS0, OCKS1)

还使用下面的寄存器。

- IIC 移位寄存器 0 到 2 (IIC0 到 IIC2)
- 从地址寄存器 0 到 2 (SVA0 到 SVA2)

备注 复用引脚的设置，请见表 4-15 端口引脚用作复用功能引脚。

(1) IIC 控制寄存器 0 到 2 (IICC0 到 IICC2)

IICC0 ~ IICC2 寄存器可以使能 / 禁止 I²C0n 操作，设置等待时间和其它 I²C 操作 (n = 0 到 2)。

这些寄存器可以以 8-位为单位或以 1-位为单位进行读取操作或写操作。然而，当 IICE0 位为 0 或者处于等待期间，设置 SPIEn, WTIMn 和 ACKEn 位。当设置 IICEn 从“0”变为“1”时，这些位同时也会被设置。

复位输入将把这些寄存器置为 00H。

复位后: 00H R/W 地址: IICC0 FFFFFFFD82H, IICC1 FFFFFFFD92H, IICC2 FFFFFFFDA2H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICn	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

(n = 0 至 2)

IICEn	I ² Cn 允许 / 禁止操作的规定
0	操作停止。 IICSn 寄存器复位 ¹ 。 内部操作停止。
1	操作允许。
当 SCL0n 和 SDA0n 线为高电平时, 确保将这一位设置为 1。	
清除条件 (IICEn 位 = 0)	
- 由指令清除 - 复位后	
设置条件 (IICEn 位 = 1)	
由指令设置	

LRELn ²	退出通信
0	正常操作
1	退出当前通信状态并且设置为待机模式。这个设置在执行后会自动清除。它的使用包括接收到本地不相关扩展码的情况。 SCL0n 和 SDA0n 线设置为高阻态。 IICSn 寄存器的 STTn 和 SPTn 位和 MSTSn, EXCn, COIn, TRCn, ACKDn 和 STDn 位清零。
退出通信后的待机模式保持有效, 直到下一个通信进入条件满足时为止。	
- 检测到停止条件时, 在主模式中重启。 - 启动条件之后, 产生地址匹配或者接收到扩展码。	
清除条件 (LRELn 位 = 0)	
- 执行后自动清除 - 复位后	
设置条件 (LRELn 位 = 1)	
由指令设置	

WRELn ²	等待状态取消控制
0	等待状态不取消
1	等待状态取消。 该设置会在取消等待状态后行动清除。
清除条件 (WRELn 位 = 0)	
- 执行后自动清除 - 复位后	
设置条件 (WRELn 位 = 1)	
由指令设置	

注 1. IICSn 寄存器, IICFn.STCFn 和 IICFn.IICBSYn 位, 以及 IICCLn.CLDn 和 IICCLn.DADn 位被复位。
2. 当 IICEn 位 = 0 时, 标志信号设置无效。

注意事项 当 SCL0n 线是高电平且 SDA0n 线是低电平, 如果 I²Cn 允许操作 (IICEn 位 = 1), 那么立即检测到启动条件。为了避免这种情况, 允许 I²Cn 操作之后, 立即用位操作指令将 LRELn 位置 1。

备注 数据设置后进行读取时, LRELn 和 WRELn 位清零。

SPIEn [※]	当检测到停止条件时允许 / 禁止中断请求产生	
0	禁止	
1	允许	
清除条件（SPIEn 位 = 0）		设置条件（SPIEn 位 = 1）
• 由指令清除 • 复位后		• 由指令设置

WTIMn ^注	等待状态和中断请求产生的控制	
0	在第 8 个时钟下降沿产生中断请求。 主模式：8 个时钟输出后，时钟输出设置为低并设置等待状态。 从模式：8 个时钟输入后，时钟设置为低电平，对主设备设置等待状态。	
1	在第 9 个时钟下降沿产生中断请求。 主模式：9 个时钟输出后，时钟输出设置为低并设置等待状态。 从模式：9 个时钟输入后，时钟设置为低电平，对主设备设置等待状态。	
地址传送过程中，不管此位设置为何值，在第 9 个时钟下降沿产生中断。地址传送完毕，此位设置有效。主模式下，地址传送过程中，在第 9 个时钟下降沿插入等待状态。对于接收本地地址的从设备来说， $\overline{\text{ACK}}$ 产生后，在第 9 个时钟下降沿插入等待状态。然而，当从设备接收到扩展码时，在第 8 个时钟下降沿插入等待状态。		
清除条件（WTIMn 位 = 0）		设置条件（WTIMn 位 = 1）
• 由指令清除 • 复位后		• 由指令设置

ACKEn [※]	响应控制
0	禁止响应。
1	允许响应。在第 9 个时钟期间，SDA0n 线设置为低电平。
对于地址接收来说，ACKEn 位设置无效。这种情况下，当地址匹配时，产生 $\overline{\text{ACK}}$ 。 然而，对扩展码的接收来说，ACKEn 位设置有效。	
清除条件（ACKEn 位 = 0）	设置条件（ACKEn 位 = 1）
• 由指令清除 • 复位后	• 由指令设置

注 当 IICEn 位 = 0 时，标志信号设置无效。

备注 n = 0 至 2

<R>

STTn	启动条件触发
0	不产生启动条件。
1	总线释放时（在 STOP 模式中）： 启动条件产生时（作为主设备开始）。当 SCLn 线为高电平时，SDA0n 线从高电平变为低电平，然后产生启动条件。接着，固定的时间间隔过去后，SCL0 线将变为低电平（等待状态）。 和第三方通信期间： - 假如通信保留功能允许（IICFn.IICRSVn 位 = 0） - 此触发器功能作为启动条件保留标志。当设置为 1 时，释放总线，然后自动产生启动条件。 - 假如通信保留功能禁止（IICRSVn = 1） - IICFn.STCFn 位被设置为 1，且设置（1）到 STTn 位中的信息被清除。触发器不产生启动条件。 等待状态（当主设备时）： 等待状态释放后，重启条件产生。
有关设置时序的注意事项	
对于主接收： 传送期间不能设置为 1。只有当 ACKEn 位设置为 0，并且告知从设备接收终止时，设置为 1 对于主传送： 在 ACK 期间启动条件不能正常产生。第 9 个时钟输出后的等待过程中，设置为 1。 对于从设备： 即使当通信保留功能禁止（IICRSVn 位 = 1）时，也进入通信保留状态。 - 禁止同 SPTn 位一起设为 1。 - 当 STTn 位设置为 1 时，禁止将 STTn 位再次设置为 1，直到设置清零为止。	
清除条件（STTn 位 = 0）	设置条件（STTn 位 = 1）
- 在通信保留禁止状态下，当 STTn 位设置为 1 时 - 由仲裁失败清除 - 在主设备中通过生成启动条件来清除 - 当 LRELn 位 = 1（通信保存） - 当 IICEn 位 = 0（操作停止） - 复位后	由指令设置

备注 1. 如果在数据设置立即读取数据，则 STTn 位为 0。
 2. n = 0 至 2

SPTn	停止条件触发
0	不产生停止条件.
1	产生停止条件（主设备传送终止）。 SDA0n 线进入低电平后，要么将 SCL0n 设置为高电平，要么等待直到 SCL0n 引脚变为高电平。接着，固定的时间间隔过去后，SDAn0 线从低电平变为高电平，停止条件产生。
有关设置时序的注意事项	
对于主接收：	传送期间不能设置为 1。 只有当 ACKEn 位设置为 0，并且从设备接收终止后的等待过程中，才能设置为 1。
对于主传送：	在 ACK 接收期间停止条件不能正常产生。第 9 个时钟输出后的等待过程中，设置为 1。
● 不能和 STTn 位同时设置为 1。 ● SPTn 位只有处于主模式^注时，才设置为 1。 ● 当 WTIMn 位设置为 0 时，假如输出 8 个时钟后的等待过程中。SPTn 位设置为 1，注意在第 9 个时钟的高电平期间，停止条件将产生。 WTIMn 位在输出 8 个时钟后的等待期间应当从 0 变成 1，SPTn 位应当在第 9 个时钟输出后的等待期间设置为 1。 ● 当 SPTn 位设置为 1 时，禁止将 SPTn 位再次设置为 1，直到设置清零为止。	
清除条件（SPTn 位=0）	设置条件（SPTn 位 = 1）
● 由仲裁失败清除 ● 检测到停止条件后自动清除。 ● 当 LRELn 位 = 1（通信保存） ● 当 IICEn 位 = 0（操作停止） ● 复位后	● 由指令设置

注 只在主模式下，设置 SPTn 位为 1。然而，当 IICRSVn 位为 0 时，SPTn 位必须设置为 1，并且切换到操作允许状态后，检测到第一个停止条件之前产生停止条件。详细情况，请看 17.15 注意事项。

注意事项 当 TRCn 位 = 1 时，在第 9 时钟和等待状态取消期间，WRELn 位设置为 1，之后，此位清零，并且 SDA0n 线设置为高阻状态。

备注 1. 如果在数据设置后立即进行读操作，则 SPTn 位设置为 0。
2. n = 0 至 2

(2) IIC 状态寄存器 0 到 2 (IICS0 ~ IICS2)

IICS0 ~ IICS2 寄存器指示 I²C0n 总线的状态 (n = 0 到 2)。

这些寄存器为 8 位或 1 位只读寄存器。然而, 只有当 IICn.STTn 位为 1 或者在等待期间内才能进行读操作。

复位输入将把这些寄存器置为 00H。

注意事项 在下列状态中禁止访问 IICSn 寄存器。详细情况, 参见 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。

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

(1/3)

复位后: 00H

R

地址: IICS0 FFFFFFFD86H, IICS1 FFFFFFFD96H, IICS2 FFFFFFFDA6H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

(n = 0 到 2)

MSTSn	主设备状态
0	从设备状态或者通信等待状态
1	主设备通信状态
清除条件 (MSTSn 位 = 0)	
• 检测到停止条件 • 当 ALDn 位 = 1 (仲裁失败) • 由 LRELn 位 = 1 (通信保存) • 当 IICEn 位从 1 变为 0 (操作停止) • 复位后	
设置条件 (MSTSn 位 = 1)	
• 启动条件产生时	

ALDn	检测仲裁失败
0	此状态表明没有仲裁或者仲裁结果是“成功”。
1	此状态表明仲裁结果是失败。 MSTSn 位清零。
清除条件 (ALDn 位 = 0)	
• IICSn 寄存器读^注后自动清除 • 当 IICEn 位从 1 变为 0 (操作停止) • 复位后	
设置条件 (ALDn 位 = 1)	
• 当仲裁结果是“失败”。	

EXCn	扩展码接收的检测
0	没有接收到扩展码。
1	接收到扩展码。
清除条件 (EXCn 位 = 0)	
• 检测到启动条件 • 检测到停止条件 • 由 LRELn 位 = 1 清除 (通信保存) • 当 IICEn 位从 1 变成 0 (停止操作) • 复位后	
清除条件 (EXCn 位 = 1)	
• 当接收到的地址数据的高四位是“0000”或者“1111” (在第 8 个时钟的上升沿设置)。	

注 当为 IICSn 寄存器中的另一个位执行位操作指令时, ALDn 位也会被清除。

COIn	检测匹配地址	
0	地址不匹配。	
1	地址匹配。	
清除条件 (COIn 位 = 0)		设置条件 (COIn 位 = 1)
• 检测到启动条件 • 检测到停止条件 • 由 LRELn 位 = 1 清除 (通信保存) • 当 IICEn 位从 1 变成 0 (停止操作) • 复位后		• 当接收到的地址与本地地址匹配时 (SVAn 寄存器) (在第 8 个时钟的上升沿设置)。

TRCn	传送 / 接收状态检测	
0	接收状态 (非传送状态)。SDA0n 线设置为高阻状态。	
1	传送状态。SO 锁存器中的数值允许输出到 SDA0n 线 (第 1 个字节的第 9 个时钟下降沿开始有效)。	
清除条件 (TRCn 位 = 0)		设置条件 (TRCn 位 = 1)
• 检测到停止条件 • 由 LRELn 位 = 1 (通信保存) • 当 IICEn 位从 1 变为 0 (操作停止) • 由 IICn.WRELn 位 = 1^注清除 • 当 ALDn 位从 0 变为 1 (仲裁失败) • 复位后		主设备 • 启动条件产生时 • 当“0”输出给第一个字节的 LSB (传送方向指定位) 从设备 • 当“1”由第一个字节的 LSB (传送方向指定位) 输入
主设备		
• 当“1”输出给第一个字节的 LSB (传送方向指定位)		
从设备		
• 检测到开始状态。		
不用于通信时		

ACKDn	ACK 检测	
0	不检测 $\overline{\text{ACK}}$ 。	
1	检测 $\overline{\text{ACK}}$ 。	
清除条件 (ACKDn 位 = 0)		设置条件 (ACKD 位 = 1)
• 检测到停止条件 • 在下一个字节的第一个时钟的上升沿 • 由 LRELn 位 = 1 来清除 (通信保存) • 当 IICEn 位从 1 变为 0 (操作停止) • 复位后		• SCL0n 引脚的第九个时钟的上升沿的 SDA0n 位设置为低电平后

注 当 WRELn 位设置为 1 且在第 9 个时钟通过 TRCn = 1 指令将等待状态取消为 0 时, TRCn 位清零, SDA0n 线为高阻态。

备注 n = 0 至 2

STDn	开始状态检测	
0	不检测开始状态。	
1	检测开始状态。 指示地址传送期间有效。	
清除条件（STDn 位 = 0）		设置条件（STDn 位 = 1）
• 检测到停止条件 • 在地址传送后下一个字节的第一个时钟的上升沿时 • 由 LRELn 位 = 1 来清除（通信保存） • 当 IICEn 位从 1 变为 0（操作停止） • 复位后		• 检测到开始状态。

SPDn	停止条件检测	
0	不检测停止状态。	
1	检测停止状态。主设通信终止，总线释放。	
清除条件（SPDn 位 = 0）		设置条件（SPDn 位 = 1）
• 在此位设置后地址传送字节的第一个时钟的上升沿及检测到启动条件 • 当 IICEn 位从 1 变为 0（操作停止） • 复位后		• 检测到停止条件

备注 n = 0 至 2

(3) IIC 标志寄存器 0 到 2 (IICF0 ~ IICF2)

IICF0 ~ IICF2 寄存器设置 I²C0n 操作模式，并且指示 I²C 总线状态。

这些寄存器可以以 8-位为单位或以 1-位为单位进行读取操作或写操作。然而，STCFn 和 IICBSYn 位为只读。

IICRSVn 允许 / 禁止通信保留功能（请看 **17.14 通信保留**）。

IICBSYn 位的初始值用 STCENn 位设置（请看 **17.15 注意事项**）。

只有当 I²C0n 操作禁止（IICCN.IICEN 位 = 0）时，IICRSVn 和 STCENn 位可写。操作允许后，IICFn 为可读（n = 0 到 2）。

复位输入将把这些寄存器置为 00H。

复位后: 00H R/W[※] 地址: IICF0 FFFFFFFD8AH, IICF1 FFFFFFFD9AH, IICF2 FFFFFFFDAAH

	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

(n = 0 至 2)

STCFn	STTn 位清除
0	启动条件发出
1	启动条件不发出, STTn 位清除
清除条件 (STCFn 位 = 0)	
- 通过 IICn.STTn 位 = 1 清除 - 当 IICn.IICEn 位 = 0 时 - 复位后	
设置条件 (STCFn 位 = 1)	
通信保留禁止 (IICRSVn 位 = 1) 期间, 当启动条件不发出, 并且 STTn 标志清零时。	

IICBSYn	I ² C0n 总线状态
0	总线释放状态 (当 STCENn 位 = 1 时, 默认通信状态)
1	总线通信状态 (当 STCENn 位 = 0 时, 默认通信状态)
清除条件 (IICBSYn 位 = 0)	
- 检测到停止条件 - 当 IICEn 位 = 0 时 - 复位后	
设置条件 (IICBSYn 位 = 1)	
- 检测到启动条件 - 当 STCENn 位 = 0, 通过设置 IICEn 位	

STCENn	初始开始允许触发
0	允许操作 (IICEn 位 = 1) 后, 启动条件不能产生直到检测到停止条件。
1	允许操作 (IICEn 位 = 1) 后, 即使没有检测到停止条件, 启动条件也产生。
清除条件 (STCENn 位 = 0)	
- 检测到启动条件 - 复位后	
设置条件 (STCENn 位 = 1)	
由指令设置	

IICRSVn	通信保留功能禁止位
0	通信保留允许
1	通信保留禁止
清除条件 (IICRSVn 位 = 0)	
- 由指令清除 - 复位后	
设置条件 (IICRSVn 位 = 1)	
由指令设置	

注 位 6 和为 7 为只读位。

- 注意事项**
- 只有当操作停止 (IICEn 位 = 0) 时, 才写 STCENn 位。
 - 当 STCENn 位 = 1 时, 允许 I²Cn 总线操作后, 立即识别总线释放状态 (IICBSYn 位 = 0), 而不管实际总线状态。因此, 为了发出第一个启动条件 (STTn 位 = 1), 有必要确认总线已经释放, 以免干扰其它通信。
 - 只有当操作停止 (IICEn 位 = 0) 时, 才写 IICRSVn 位。

(4) IIC 时钟选择寄存器 0 ~ 2 (IICCL0 ~ IICCL2)

IICCL0 ~ IICCL2 寄存器设置 I²C0n 总线的传送时钟。

这些寄存器可以以 8-位为单位或以 1-位为单位进行读取操作或写操作。然而，CLDn 和 DADn 位只读。

当 IICCN.IICEn 位 = 0 时，设置 IICCLn 寄存器。

SMCn, CLn1 和 CLn0 位和 IICXn.CLXn 位以及 OCKSm 寄存器的 OCKSTHm, OCKSm1 和 OCKSm0 位共同设置。（请参阅 17.4 (6) I²C0n 传送时钟设置方法）（n = 0 到 2, m = 0, 1）。

复位输入将把这些寄存器置为 00H。

复位后: 00H

R/W^{*}

地址: IICCL0 FFFFD84H, IICCL1 FFFFD94H, IICCL2 FFFFD44H

	7	6	<5>	<4>	3	2	1	0
IICCLn	0	0	CLDn	DADn	SMCn	DFCn	CLn1	CLn0

(n = 0 到 2)

CLDn	SCL0n 引脚电平的检测（只有当 IICCN.IICEn 位 = 1 时有效）
0	SCL0n 引脚在低电平时检测。
1	SCL0n 引脚在高电平时检测。
清除条件（CLDn 位 = 0）	
- 当 SCL0n 引脚为低电平 - 当 IICEn 位 = 0（操作停止） - 复位后	
设置条件（CLDn 位 = 1）	
当 SCL0n 引脚为高电平	

DADn	SDA0n 引脚电平的检测（只有当 IICEn 位 = 1 时有效）
0	SDA0n 引脚在低电平时检测。
1	SDA0n 引脚在高电平时检测。
清除条件（DADn 位 = 0）	
- 当 SDA0n 引脚为低电平 - 当 IICEn 位 = 0（操作停止） - 复位后	
设置条件（DAD0n 位 = 1）	
当 SDA0n 引脚为高电平	

SMCn	操作模式切换
0	标准模式中操作。
1	高速模式中操作。

DFCn	数字滤波器操作控制
0	数字滤波器关闭。
1	数字滤波器开启。
数字滤波器只能在高速模式下使用。	
高速模式下，不管 DFCn 位设置为何（开 / 关），传送时钟都不变化。	
数字滤波器用来在高速模式下除掉噪声。	

注 IICCLn 的位 4 和位 5 为只读位。

注意事项 一定要将 IICCLn 的第 7 位和第 6 位清零。

备注 IICCN.IICEn 位 = 0 时，当读 CLDn 和 DADn 位时，读取 0。

(5) IIC 功能扩展寄存器 0 到 2 (IICX0 ~ IICX2)

IICX0 ~ IICS2 寄存器设置 I²C0n 功能扩展（只在高速模式下有效）。
这些寄存器可以以 8-位为单位或以 1-位为单位进行读取操作或写操作。
CLXn 位的设置和 IICCLn 寄存器的 SMCn, CLn1, CLn0 位 以及 OCKSm 寄存器的 OCKSTHm, OCKSm1, OCKSm0 位一起进行（请参阅 17.4 (6) I²C0n 传送时钟设置方法）（m = 0, 1）。
当 IICn.IICEn 位 = 0 时设置 IICXn 寄存器。
复位输入将把这些寄存器置为 00H。

复位后: 00H R/W 地址: IICX0 FFFFFFFD85H, IICX1 FFFFFFFD95H, IICX2 FFFFFFFDA5H

	7	6	5	4	3	2	1	<0>
IICXn	0	0	0	0	0	0	0	CLXn

(n = 0 至 2)

(6) I²C0n 传送时钟设置方法

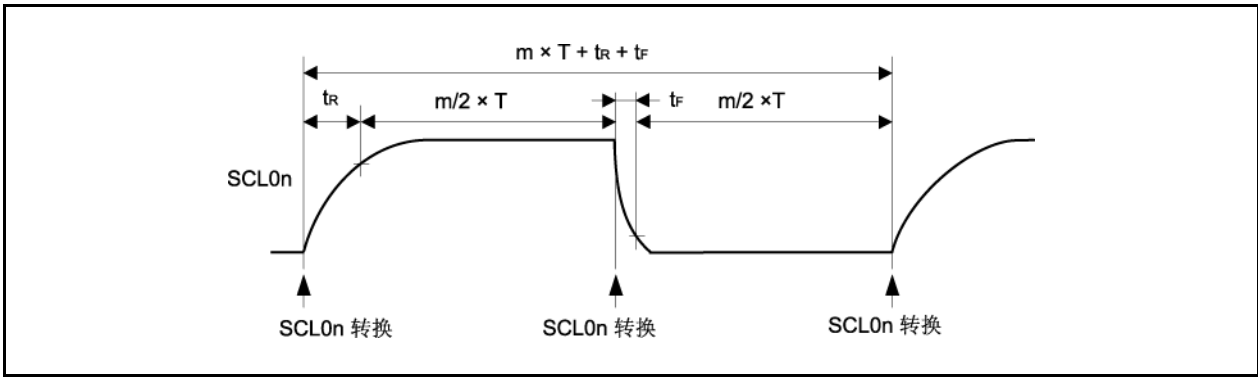
I²C0n 传送时钟频率（f_{SCL}）用下列表达式计算（n = 0 到 2）。

$$f_{SCL} = 1 / (m \times T + t_R + t_F)$$

<R> m = 12, 18, 24, 36, 44, 48, 54, 60, 66, 72, 86, 88, 90, 96, 120, 132, 172, 176, 198, 220, 258, 264, 330, 344, 430（参见 表 17-2 时钟设置）。
T: 1/f_{xx}
t_R: SCL0n 引脚上升时间
t_F: SCL0n 引脚下降时间

例如，当 f_{xx} = 19.2 MHz，m = 198，t_R = 200 ns，和 t_F = 50 ns 时 I²C0n 传送时钟频率（f_{SCL}）用下式计算。

$$f_{SCL} = 1 / (198 \times 52 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 94.7 \text{ kHz}$$



要选择的时钟可以由 IICCLn 寄存器的 SMCn, CLn1 和 CLn0 位，IICXn 寄存器的 CLXn 位以及 OCKSm 寄存器的 OCKSTHm, OCKSm1 和 OCKSm0 位共同设置（n = 0 到 2，m = 0, 1）。

表 17-2. 时钟设置 (1/2)

	IICCL0				选择时钟	传送时钟	可选择的主时钟 频率 (f _{xx}) 范围	操作模式
	第 0 位	第 3 位	第 1 位	第 0 位				
	CLX0	SMC0	CL01	CL00				
<R>	0	0	0	0	f _{xx} (设置 OCKS0 = 18H 时)	f _{xx} /44	2.00 MHz ≤ f _{xx} ≤ 4.19 MHz	标准模式 (SMC0 位 = 0)
					f _{xx} /2 (设置 OCKS0 = 10H 时)	f _{xx} /88	4.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
					f _{xx} /3 (设置 OCKS0 = 11H 时)	f _{xx} /132	6.00 MHz ≤ f _{xx} ≤ 12.57 MHz	
					f _{xx} /4 (设置 OCKS0 = 12H 时)	f _{xx} /176	8.00 MHz ≤ f _{xx} ≤ 16.76 MHz	
					f _{xx} /5 (设置 OCKS0 = 13H 时)	f _{xx} /220	10.00 MHz ≤ f _{xx} ≤ 20.95 MHz ^{注 1}	
<R> <R> <R>	0	0	0	1	f _{xx} (设置 OCKS0 = 18H 时)	f _{xx} /86	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	
					f _{xx} /2 (设置 OCKS0 = 10H 时)	f _{xx} /172	8.38 MHz ≤ f _{xx} ≤ 16.76 MHz	
					f _{xx} /3 (设置 OCKS0 = 11H 时)	f _{xx} /258	12.57 MHz ≤ f _{xx} ≤ 25.14 MHz ^{注 1}	
					f _{xx} /4 (设置 OCKS0 = 12H 时)	f _{xx} /344	16.76 MHz ≤ f _{xx} ≤ 32.00 MHz ^{注 1}	
					f _{xx} /5 (设置 OCKS0 = 13H 时)	f _{xx} /430	20.95 MHz ≤ f _{xx} ≤ 32.00 MHz ^{注 2}	
<R> <R>	0	0	1	0	f _{xx} ^{注 3}	f _{xx} /86	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	
	0	0	1	1	f _{xx} (设置 OCKS0 = 18H 时)	f _{xx} /66	6.40 MHz	
					f _{xx} /2 (设置 OCKS0 = 10H 时)	f _{xx} /132	12.80 MHz	
					f _{xx} /3 (设置 OCKS0 = 11H 时)	f _{xx} /198	19.20 MHz	
					f _{xx} /4 (设置 OCKS0 = 12H 时)	f _{xx} /264	25.60 MHz ^{注 2}	
<R> <R> <R>	0	1	0	×	f _{xx} (设置 OCKS0 = 18H 时)	f _{xx} /24	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	高速模式 (SMC0 bit = 1)
					f _{xx} /2 (设置 OCKS0 = 10H 时)	f _{xx} /48	8.00 MHz ≤ f _{xx} ≤ 16.76 MHz	
					f _{xx} /3 (设置 OCKS0 = 11H 时)	f _{xx} /72	12.00 MHz ≤ f _{xx} ≤ 25.14 MHz ^{注 1}	
					f _{xx} /4 (设置 OCKS0 = 12H 时)	f _{xx} /96	16.00 MHz ≤ f _{xx} ≤ 32.00 MHz ^{注 1}	
					f _{xx} /5 (设置 OCKS0 = 13H 时)	f _{xx} /120	20.00 MHz ≤ f _{xx} ≤ 32.00 MHz ^{注 2}	
<R> <R>	0	1	1	0	f _{xx} ^{注 3}	f _{xx} /24	4.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
	0	1	1	1	f _{xx} (设置 OCKS0 = 18H 时)	f _{xx} /18	6.40 MHz	
					f _{xx} /2 (设置 OCKS0 = 10H 时)	f _{xx} /36	12.80 MHz	
					f _{xx} /3 (设置 OCKS0 = 11H 时)	f _{xx} /54	19.20 MHz	
					f _{xx} /4 (设置 OCKS0 = 12H 时)	f _{xx} /72	25.60 MHz ^{注 2}	
<R>	1	1	0	×	f _{xx} (设置 OCKS0 = 18H 时)	f _{xx} /12	4.00 MHz ≤ f _{xx} ≤ 4.19 MHz	
					f _{xx} /2 (设置 OCKS0 = 10H 时)	f _{xx} /24	8.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
					f _{xx} /3 (设置 OCKS0 = 11H 时)	f _{xx} /36	12.00 MHz ≤ f _{xx} ≤ 12.57 MHz	
					f _{xx} /4 (设置 OCKS0 = 12H 时)	f _{xx} /48	16.00 MHz ≤ f _{xx} ≤ 16.67 MHz	
					f _{xx} /5 (设置 OCKS0 = 13H 时)	f _{xx} /60	20.00 MHz ≤ f _{xx} ≤ 20.95 MHz ^{注 1}	
	1	1	1	0	f _{xx} ^{注 3}	f _{xx} /12	4.00 MHz ≤ f _{xx} ≤ 4.19 MHz	
除上述情况					禁止设置	—	—	—

- 注
1. 只能在 V850ES/SG2-H 中设置 f_{xx} > 20 MHz。
 2. 仅在 V850ES/SG2-H 中可设置。
 3. 由于不管 OCKS0 寄存器的数值如何选择时钟都为 f_{xx}，因此将 OCKS0 寄存器清除为 00H (I²C 分频时钟停止状态)。

备注 ×: 不考虑

表 17-2. 时钟设置 (2/2)

	IICXm		IICCLm		选择时钟	传送时钟	可选择的主时钟 频率 (f _{xx}) 范围	操作模式
	第 0 位	第 3 位	第 1 位	第 0 位				
	CLXm	SMCm	CLm1	CLm0				
<R>	0	0	0	0	f _{xx} (设置 OCKS1 = 18H 时)	f _{xx} /44	2.00 MHz ≤ f _{xx} ≤ 4.19 MHz	标准模式 (SMCm 位 = 0)
					f _{xx} /2 (设置 OCKS1 = 10H 时)	f _{xx} /88	4.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
					f _{xx} /3 (设置 OCKS1 = 11H 时)	f _{xx} /132	6.00 MHz ≤ f _{xx} ≤ 12.57 MHz	
					f _{xx} /4 (设置 OCKS1 = 12H 时)	f _{xx} /176	8.00 MHz ≤ f _{xx} ≤ 16.76 MHz	
					f _{xx} /5 (设置 OCKS1 = 13H 时)	f _{xx} /220	10.00 MHz ≤ f _{xx} ≤ 20.95 MHz ^{#1}	
<R>	0	0	0	1	f _{xx} (设置 OCKS1 = 18H 时)	f _{xx} /86	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	
					f _{xx} /2 (设置 OCKS1 = 10H 时)	f _{xx} /172	8.38 MHz ≤ f _{xx} ≤ 16.76 MHz	
					f _{xx} /3 (设置 OCKS1 = 11H 时)	f _{xx} /258	12.57 MHz ≤ f _{xx} ≤ 25.14 MHz ^{#1}	
					f _{xx} /4 (设置 OCKS1 = 12H 时)	f _{xx} /344	16.76 MHz ≤ f _{xx} ≤ 32.00 MHz ^{#1}	
					f _{xx} /5 (设置 OCKS1 = 13H 时)	f _{xx} /430	20.95 MHz ≤ f _{xx} ≤ 32.00 MHz ^{#2}	
<R>	0	0	1	0	f _{xx} ^{#3}	f _{xx} /86	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	
					f _{xx} (设置 OCKS1 = 18H 时)	f _{xx} /66	6.40 MHz	
					f _{xx} /2 (设置 OCKS1 = 10H 时)	f _{xx} /132	12.80 MHz	
					f _{xx} /3 (设置 OCKS1 = 11H 时)	f _{xx} /198	19.20 MHz	
					f _{xx} /4 (设置 OCKS1 = 12H 时)	f _{xx} /264	25.60 MHz ^{#2}	
<R>	0	1	0	×	f _{xx} (设置 OCKS1 = 18H 时)	f _{xx} /24	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	高速模式 (SMCm 位 = 1)
					f _{xx} /2 (设置 OCKS1 = 10H 时)	f _{xx} /48	8.00 MHz ≤ f _{xx} ≤ 16.76 MHz	
					f _{xx} /3 (设置 OCKS1 = 11H 时)	f _{xx} /72	12.00 MHz ≤ f _{xx} ≤ 25.14 MHz ^{#1}	
					f _{xx} /4 (设置 OCKS1 = 12H 时)	f _{xx} /96	16.00 MHz ≤ f _{xx} ≤ 32.00 MHz ^{#1}	
					f _{xx} /5 (设置 OCKS1 = 13H 时)	f _{xx} /120	20.00 MHz ≤ f _{xx} ≤ 32.00 MHz ^{#2}	
<R>	0	1	1	0	f _{xx} ^{#3}	f _{xx} /24	4.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
					f _{xx} (设置 OCKS1 = 18H 时)	f _{xx} /18	6.40 MHz	
					f _{xx} /2 (设置 OCKS1 = 10H 时)	f _{xx} /36	12.80 MHz	
					f _{xx} /3 (设置 OCKS1 = 11H 时)	f _{xx} /54	19.20 MHz	
					f _{xx} /4 (设置 OCKS1 = 12H 时)	f _{xx} /72	25.60 MHz ^{#2}	
<R>	1	1	0	×	f _{xx} (设置 OCKS1 = 18H 时)	f _{xx} /12	4.00 MHz ≤ f _{xx} ≤ 4.19 MHz	
					f _{xx} /2 (设置 OCKS1 = 10H 时)	f _{xx} /24	8.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
					f _{xx} /3 (设置 OCKS1 = 11H 时)	f _{xx} /36	12.00 MHz ≤ f _{xx} ≤ 12.57 MHz	
					f _{xx} /4 (设置 OCKS1 = 12H 时)	f _{xx} /48	16.00 MHz ≤ f _{xx} ≤ 16.67 MHz	
					f _{xx} /5 (设置 OCKS1 = 13H 时)	f _{xx} /60	20.00 MHz ≤ f _{xx} ≤ 20.95 MHz ^{#1}	
<R>	1	1	1	0	f _{xx} ^{#3}	f _{xx} /12	4.00 MHz ≤ f _{xx} ≤ 4.19 MHz	
					禁止设置	—	—	

<R> 注 1. 只能在 V850ES/SG2-H 中设置 f_{xx} > 20 MHz。

<R> 2. 仅在 V850ES/SG2-H 中可设置。

3. 由于不管 OCKS1 寄存器的数值为何选择时钟都为 f_{xx}，因此将 OCKS1 寄存器清除为 00H (I²C 分频时钟停止状态)。

备注 1. m = 1, 2
2. ×: 不考虑

(7) IIC 分频时钟选择寄存器 0, 1 (OCS0, OCS1)

OCS0 和 OCS1 寄存器控制 I²C_{0n} 分频时钟 (n = 0 到 2)。

这些寄存器通过 OCS0 寄存器控制 I²C₀₀ 分频时钟, 通过 OCS1 寄存器控制 I²C₀₁ 和 I²C₀₂ 分频时钟。

这些寄存器可以以 8 位为单位进行读取或写入。

复位输入将把这些寄存器置为 00H。

复位后: 00H R/W 地址: OCS0 FFFF340H, OCS1 FFFF344H

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

(m = 0, 1)

OCSm	I ² C 分频时钟的操作设置
0	禁止 I ² C 分频时钟操作
1	允许 I ² C 分频时钟操作

OCSm	OCSm	OCSm	I ² C 分频时钟的选择
0	0	0	f _{xx} /2
0	0	1	f _{xx} /3
0	1	0	f _{xx} /4
0	1	1	f _{xx} /5
1	0	0	f _{xx}
其他情况			禁止设置

(8) IIC 移位寄存器 0 到 2 (IIC0 ~ IIC2)

IIC0 ~ IIC2 寄存器用于使串行传送 / 接收 (移位操作) 与串行时钟同步。这些寄存器可以以 8 位为单位进行读写, 但是在数据传送期间不应将数据写入 IIC_n 寄存器。

只在等待期间访问 (读 / 写) IIC_n 寄存器。除了等待期间, 通信状态时禁止访问该寄存器。然而, 对于主设备来说, 只有在传送触发位 (IIC_n.STT_n 位) 置 1 之后 IIC_n 寄存器才能写入。

等待期间通过写入 IIC_n 寄存器来释放等待状态, 并且数据传送开始 (n = 0 到 2)。

复位输入将把这些寄存器置为 00H。

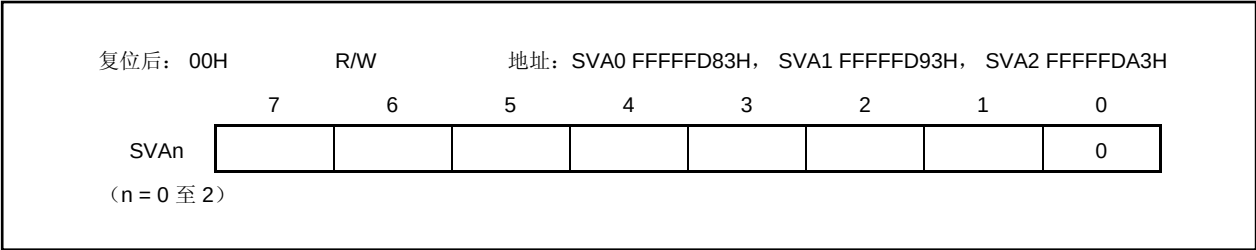
复位后: 00H R/W 地址: IIC0 FFFF80H, IIC1 FFFF80H, IIC2 FFFF80H

	7	6	5	4	3	2	1	0
IICn								

(n = 0 至 2)

(9) 从地址寄存器 0 到 2 (SVA0 ~ SVA2)

SVA0 ~ SVA2 寄存器保存 I²C 总线的从地址。
这些寄存器可以以 8 位为单位进行读写，但是 0 位始终保持为 0。然而，当 IICSn.STDn 位 = 1 (检测启动条件) 时禁止重写寄存器。
复位输入将把这些寄存器置为 00H。



17.5 I²C 总线模式功能

17.5.1 引脚配置

串行时钟引脚（SCL0n）和串行数据总线引脚（SDA0n）配置如下（n = 0 到 2）。

SCL0n 此引脚用于串行时钟输入和输出。

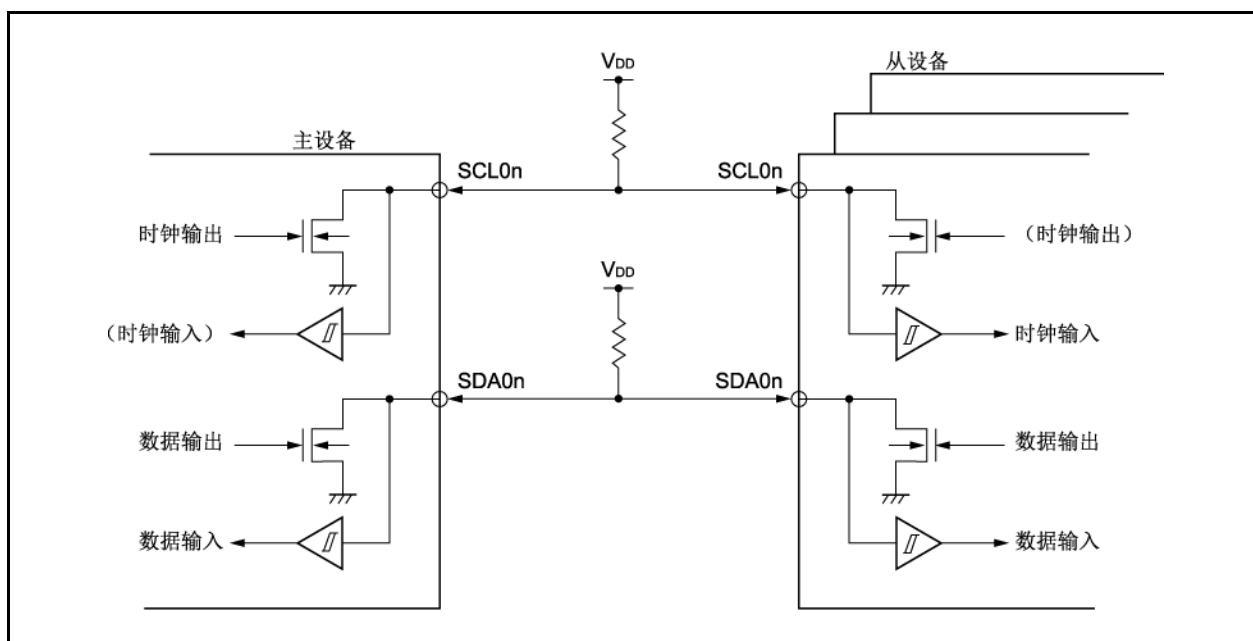
此引脚对主设备和从设备均为 N-ch 开漏输出。输入为施密特输入。

SDA0n 此引脚用于串行数据输入和输出。

此引脚对主设备和从设备均为 N-ch 开漏输出。输入为施密特输入。

由于串行时钟线和串行数据线的输出均为 N-ch 开漏输出，所以需要外部上拉电阻。

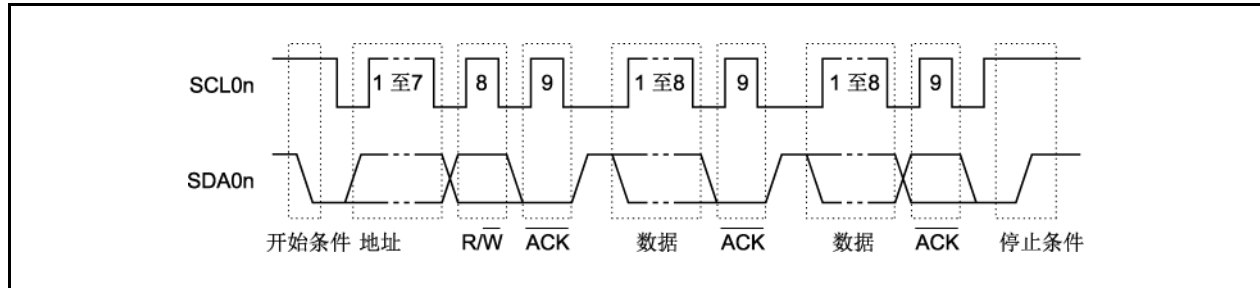
图 17-6. 引脚配置图



17.6 I²C 总线定义和控制方法

下面描述了 I²C 总线串行数据通信格式和 I²C 总线使用的信号。I²C 总线串行数据线上的“启动条件”，“地址”，“传送方向规定”，“数据”和“停止条件”的传送时序分别如下所示。

图 17-7. I²C 总线串行数据传送时序



主设备产生启动条件、从地址和停止条件。

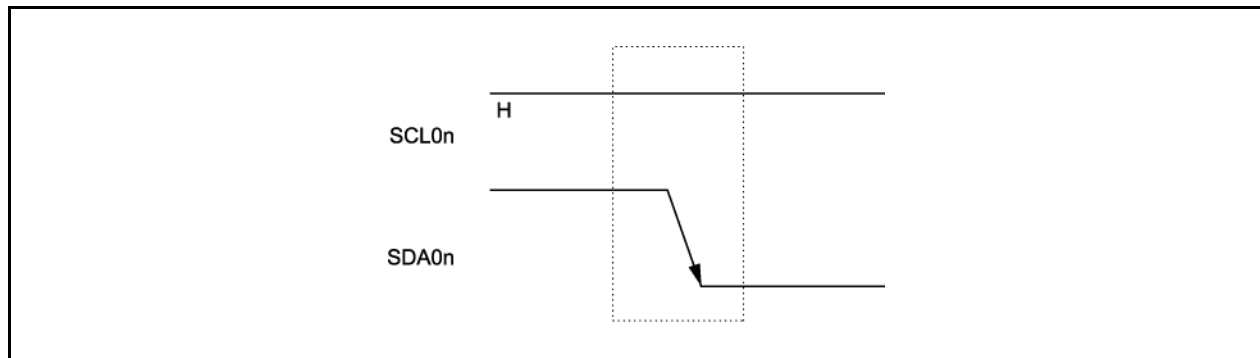
ACK 能通过主设备或者从设备产生。（一般地，它通常由接收 8 位数据的设备产生）。

串行时钟（SCL0n）由主设备连续输出。然而，在从设备中，SCL0n 引脚的低电平时间能够插入等待状态稍作延长。（n = 0 到 2）。

17.6.1 启动条件

当 SCL0n 引脚为高电平，同时 SDA0n 引脚从高电平变成低电平时，此时满足启动条件的要求。开始串行传输时，生成 SCL0 和 SDA0 引脚的启动条件，启动条件是主设备输入到从设备的。从设备能够检测到启动条件（n = 0 到 2）。

图 17-8. 启动条件



检测到停止条件（IICSn.SPDn 位 = 1）后，在 IICCN.STTn 位置 1 时会生成启动条件。检测到启动条件时，IICSn.STDn 位被置 1（n = 0 到 2）。

<R> **注意事项** 在与其它设备进行通信的过程中，当 V850ES/SG2 和 V850ES/SG2-H 的 IICCN.IICEn 位设置为 1 时，是否检测到启动条件要根据通信线的状态来决定。当 SCL0n 和 SDA0n 线为高电平时，请确保设置 IICCN.IICEn 位为 1。

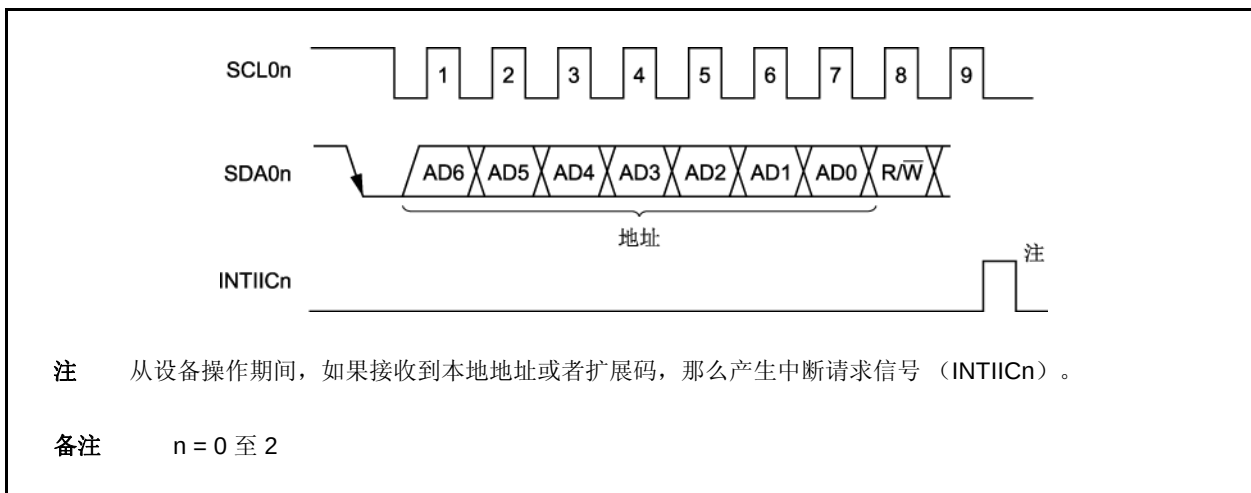
17.6.2 地址

接在启动条件后的 7 位数据被定义为地址。

地址是一个 7 位数据段，它被输出的目的是用来通过总线选择一个连接到主设备上的从设备。因此，每一个通过总线连接的从设备必须具有唯一的地址。

从设备的硬件会检测启动条件并检验收到的 7 位地址数据与存储在 $SVAn$ 寄存器里的地址数据是否匹配。如果地址数据与 $SVAn$ 寄存器里的数值匹配，那么此从设备将被选中，和主设备进行通信直到主设备产生启动条件或停止条件（ $n = 0$ 到 2）。

图 17-9. 地址



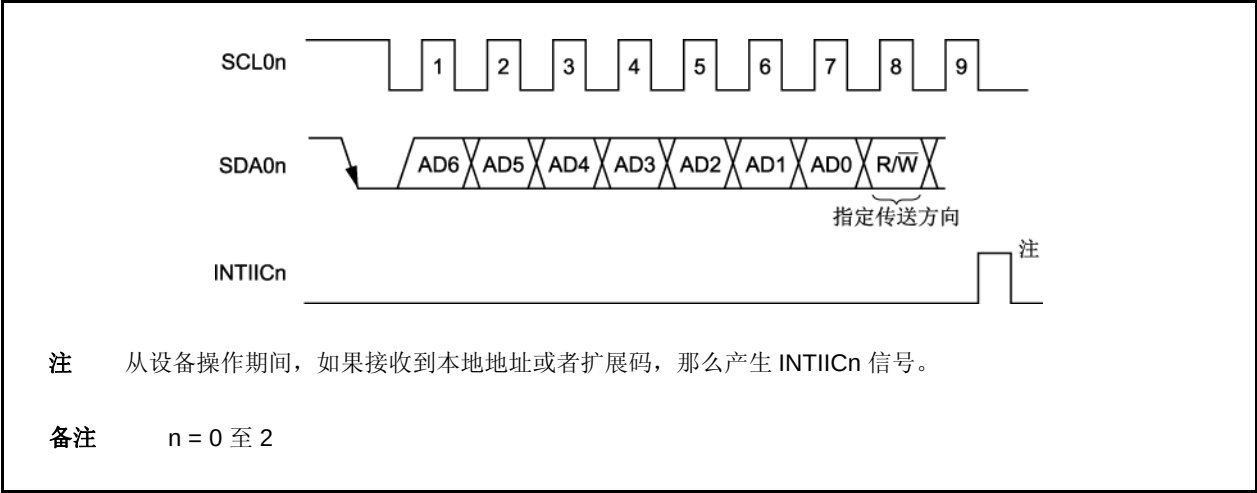
正如以下在 **17.6.3 传送方向规定** 所述，指定了传送方向的从地址和第 8 位一起被写入 IIC 移位寄存器 n （IICn），然后输出。接收到的地址写入 IICn 寄存器（ $n = 0$ 到 2）。

从地址被送入 IICn 寄存器的高 7 位。

17.6.3 指定传输方向

除了 7 位地址数据，主设备发送一位数据用来指定传送方向。当此传送方向位为 0 时，表明主设备正在给从设备传送数据。当此传送方向指定位为 1 时，表明主设备正在接收从从设备来的数据。

图 17-10. 指定传送方向



17.6.4 $\overline{\text{ACK}}$

$\overline{\text{ACK}}$ 用于确认传送和接收设备的串行数据状态。

接收设备每接收到 8 位数据就返回 $\overline{\text{ACK}}$ 。

传送设备通常在传送 8 位数据后，接收 $\overline{\text{ACK}}$ 。当 $\overline{\text{ACK}}$ 从接收设备返回时，接收过程被判断为正常结束并继续处理。 $\overline{\text{ACK}}$ 的检测通过 IICSn.ACKDn 位来确认。

当主设备是接收设备时，接收到最终数据后，它并不返回 $\overline{\text{ACK}}$ 但会产生停止条件。当接收设备是从设备时，不返回 $\overline{\text{ACK}}$ 则主设备会产生停止条件或重启条件，然后停止当前数据传送。返回 $\overline{\text{ACK}}$ 失败可能由下列原因引起。

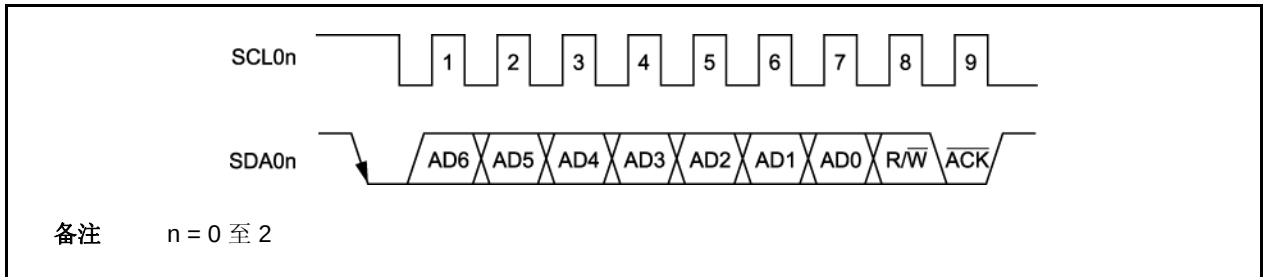
- (a) 接收未能正常进行。
- (b) 接收到终止数据。
- (c) 指定接收地址上，接收设备（从设备）不存在。

第 9 个时钟期间，当接收设备设置 SDA0n 线为低电平时，会产生 $\overline{\text{ACK}}$ （正常接收）。

当 IICSn.ACKEn 位置 1 时，允许 $\overline{\text{ACK}}$ 自动产生。紧跟着 7 位地址数据的第 8 位数据的传送能使得 IICSn.TRCn 位被设置。通常，设置 ACKEn 位进行接收（TRCn 位 = 0）。

当从设备在接收时（当 TRCn 位 = 0），如果从设备不能接收数据，那么将 ACKEn 位清 0，用于通知主设备无法接收更多数据。

类似的，当主设备接收（当 TRCn 位 = 0）并且不需要后续数据时，将 ACKEn 位清零可以禁止 $\overline{\text{ACK}}$ 产生。这就通知从设备（传送设备）数据传送结束（停止传送）。

图 17-11. $\overline{\text{ACK}}$ 

当接收到本地地址时，不管 ACKEn 位为何值， $\overline{\text{ACK}}$ 都会自动产生。如果接收到的地址不是本地地址（NACK），则不会产生 $\overline{\text{ACK}}$ 。

当接收扩展码时，预先设置 ACKEn 位为 1，以产生 $\overline{\text{ACK}}$ 。

在数据接收期间， $\overline{\text{ACK}}$ 的产生方法根据等待时序进行设置，描述如下。

- 选择 8 个时钟等待时（IICSn.WTIMn 位 = 0）：
在等待状态取消前，如果 ACKEn 位设置为 1，那么在 SCL0n 引脚的第 8 个时钟的下降沿会产生 $\overline{\text{ACK}}$ 。
- 选择 9 个时钟等待时（IICSn.WTIMn 位 = 1）：
如果 ACKEn 位预先设置为 1，则会产生 $\overline{\text{ACK}}$ 。

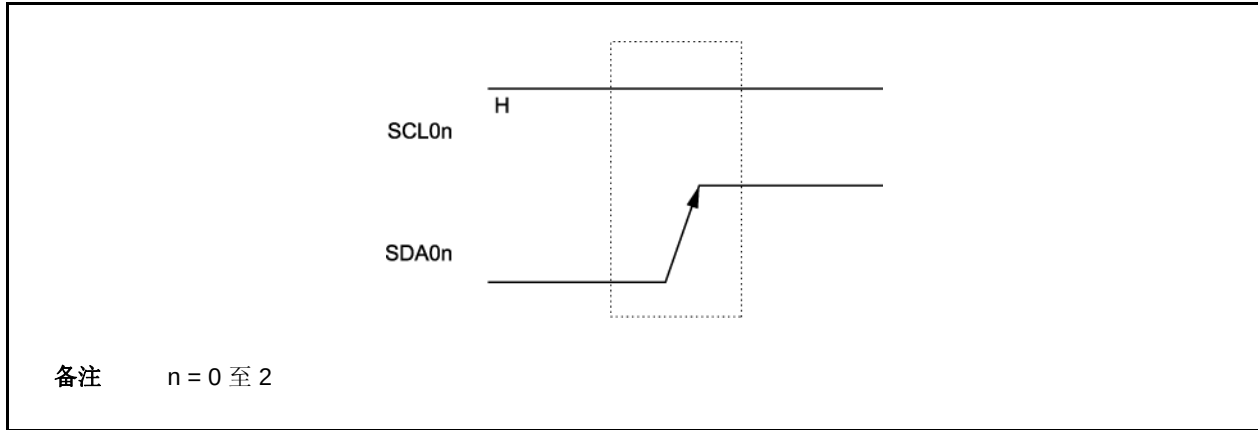
备注 n = 0 至 2

17.6.5 停止条件

当 SCL0n 引脚是高电平时，将 SDA0n 引脚从低电平转变成高电平会产生停止条件。（n = 0 到 2）。

在串行传送完成后，在主设备输出到从设备时会产生一个停止条件。用作从设备时，可以检测到启动条件。

图 17-12. 停止条件



当 IICn.SPTn 位设置为 1 时，产生停止条件。检测到停止条件时，IICSn.SPDn 位设置为 1，并且在 IICn.SPIEn 位设置为 1（n = 0 到 2）时，产生中断请求信号（INTIICn）。

17.6.6 等待状态

等待状态通常用来通知通信双方有设备（主或从设备）正在准备传送或接收数据（也就是说，处于等待状态）。

将 SCL0n 引脚设置为低电平，可以通知通信双方处于等待状态。当主设备和从设备的等待状态都被取消时，就可以开始下一个数据传送（n = 0 到 2）。

图 17-13. 等待状态（1/2）

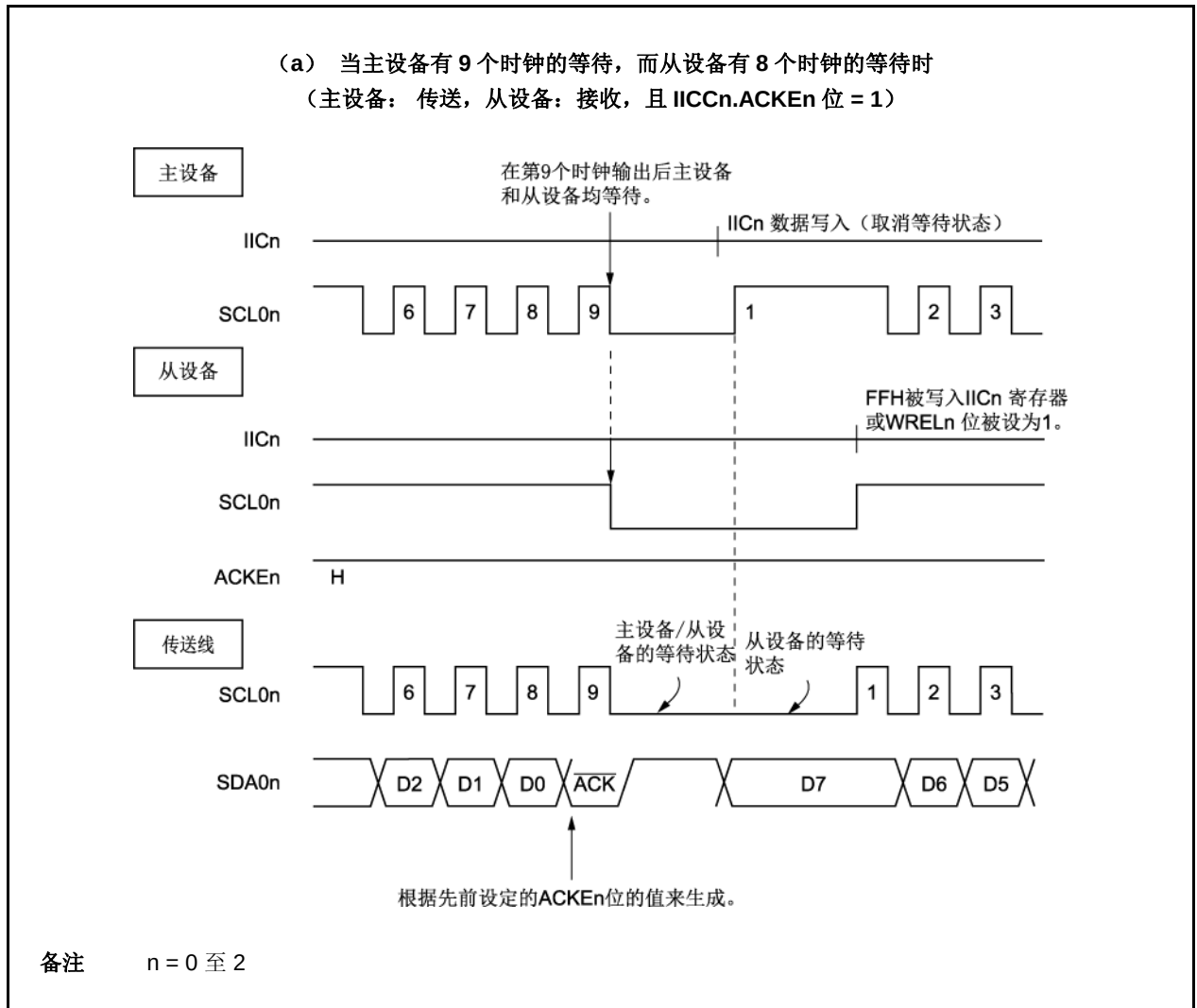
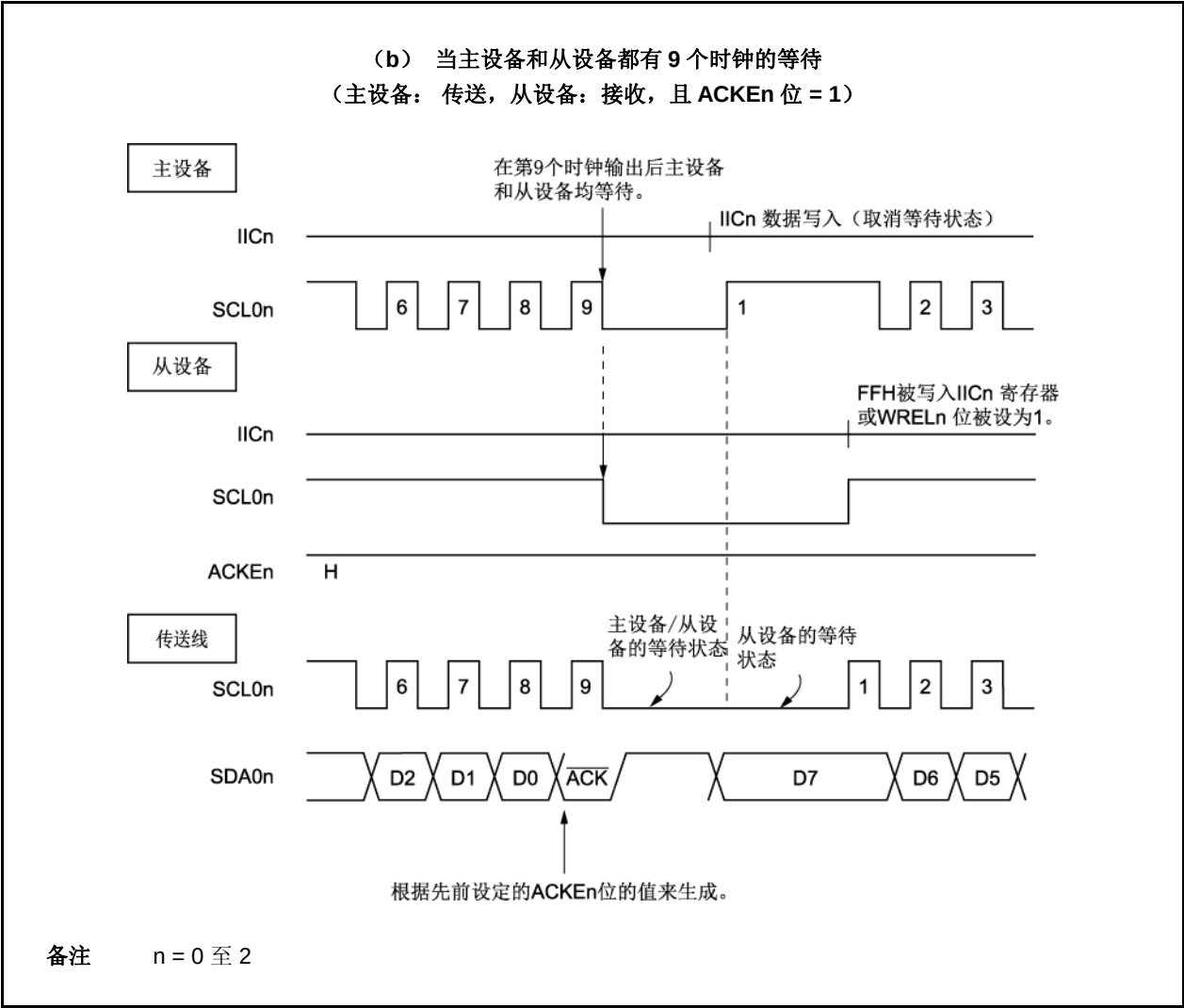


图 17-13. 等待状态（2/2）



<R> 生成启动条件后会自动生成一个等待状态。等待状态也会根据 IICn.WTIMn 位的设置自动生成。

通常，在接收方当 IICn.WRELn 位置 1 或者当 IICn 寄存器写入 FFH 时，等待状态会被取消，传送方则可以写入数据到 IICn 寄存器来取消等待状态。

主设备也可以通过以下方法来取消等待状态。

- 设置 IICn.STTn 位为 1
- 设置 IICn.SPTn 位为 1

17.6.7 等待状态取消方法

在 I²C0n 情况下，等待状态可以通过下列方式取消（n = 0 到 2）。

- 将数据写入到 IICn 寄存器
- 设置 IICn.WRELn 位为 1（等待状态取消）
- 设置 IICn.STTn 位为 1（产生启动条件）[※]
- 设置 IICn.SPTn 位为 1（产生停止条件）[※]

<R>

<R>

<R>

注 仅限主设备

如果进行了任何一个等待状态取消的动作，那么 I²C0n 将取消等待状态，并重新开始通信。

取消等待状态并发送数据（包括地址）时，将数据写入 IICn 寄存器。

取消等待状态后，为了接收数据或者完成数据传送，应设置 WRELn 位为 1。

取消等待状态后，为了产生重启条件，应设置 STTn 位为 1。

取消等待状态后，为了产生停止条件，应设置 SPTn 位为 1。

每个等待状态只能取消一次。

例如，如果通过设置 WRELn 位为 1 来取消等待状态后，数据被写入 IICn 寄存器，那么 SDA_n 线改变时序和 IICn 寄存器写时序之间的冲突可能导致输出给 SDA_n 线的数据不正确。

即使在其它操作的情况下，如果通信中途停止，将 IICn.IICEn 位清零将停止通信，从而允许取消等待状态。

如果 I²C 总线因为噪声等而锁死，将 IICn.LRELn 位置 1 将退出通信操作，从而允许取消等待状态。

17.7 I²C 中断请求信号 (INTIICn)

下面显示在 INTIICn 中断请求信号产生时序中的 IICSn 寄存器的数值，和 INTIICn 信号时序中的 IICSn 寄存器的数值。

- 备注
1.

ST:

启动条件
- AD6 至 AD0:

地址
- R/W:

指定传输方向
- ACK:

应答
- D7 至 D0:

数据
- SP:

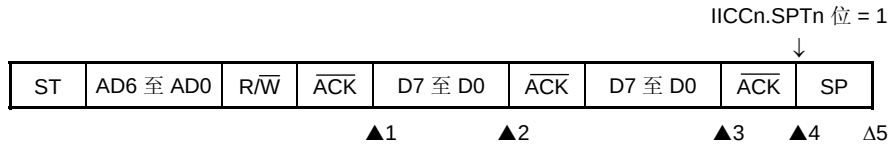
停止条件
2.

n = 0 至 2

17.7.1 主设备操作

(1) 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止 (正常传送 / 接收)

<1> 当 IICn.WTIMn 位 = 0 时



▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 1000X000B

▲3: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1[※])

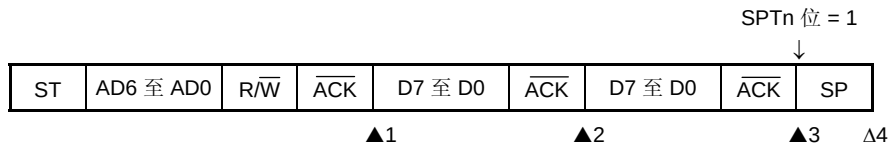
▲4: IICSn 寄存器 = 1000XX00B

Δ5: IICSn 寄存器 = 00000001B

<R> 注 将 WTIMn 位设为 1 并更改生成中断请求信号 (INTIICn) 的时序, 以生成停止条件。

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 1000X100B

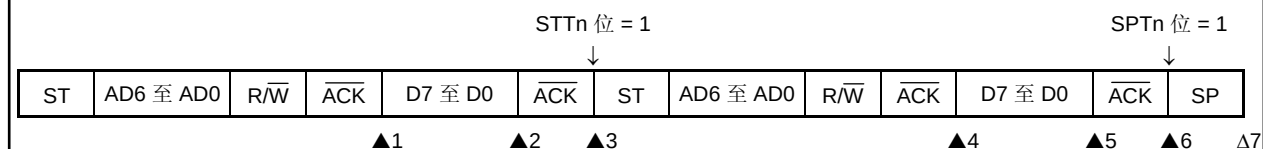
▲3: IICSn 寄存器 = 1000XX00B

Δ4: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

(2) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止 (重启)

<1> 当 WTIMn 位 = 0 时

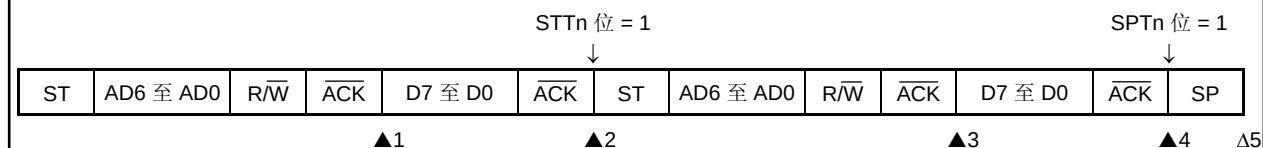


- ▲1: IICSn 寄存器 = 1000X110B
 ▲2: IICSn 寄存器 = 1000X000B (WTIMn bit = 1^{註1})
 ▲3: IICSn 寄存器 = 1000XX00B (WTIMn bit = 0^{註2})
 ▲4: IICSn 寄存器 = 1000X110B
 ▲5: IICSn 寄存器 = 1000X000B (WTIMn bit = 1^{註3})
 ▲6: IICSn 寄存器 = 1000XX00B
 ▲7: IICSn 寄存器 = 00000001B

注	1. 将 WTIMn 位设为 1 并更改生成中断请求信号 (INTIICn) 的时序以生成启动条件。 2. 将 WTIMn 位清零以恢复原始设置。 3. 将 WTIMn 位设为 1 并更改生成中断请求信号 (INTIICn) 的时序以生成停止条件。
---	--

备注	1. ▲: 总是生成 Δ: 仅在 SPIEn 位 = 1 时生成 X: 不考虑 2. n = 0 至 2
----	--

<2> 当 WTIMn 位 = 1 时

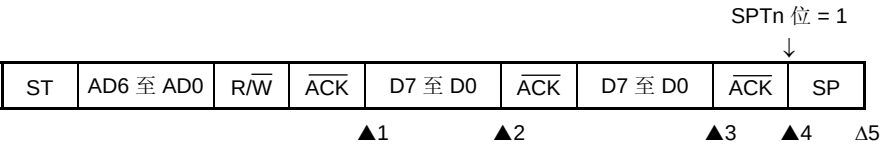


- ▲1: IICSn 寄存器 = 1000X110B
▲2: IICSn 寄存器 = 1000XX00B
▲3: IICSn 寄存器 = 1000X110B
▲4: IICSn 寄存器 = 1000XX00B
△ 5: IICSn 寄存器 = 00000001B

备注	1. ▲: 总是生成 Δ: 仅在 SPIEn 位 = 1 时生成 X: 不考虑 2. n = 0 至 2
----	--

(3) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止 (扩展码传送)

<1> 当 WTIMn 位 = 0 时



- ▲1: IICSn 寄存器 = 1010X110B
▲2: IICSn 寄存器 = 1010X000B
▲3: IICSn 寄存器 = 1010X000B (WTIMn bit = 1[※])
▲4: IICSn 寄存器 = 1010XX00B
Δ 5: IICSn 寄存器 = 00000001B

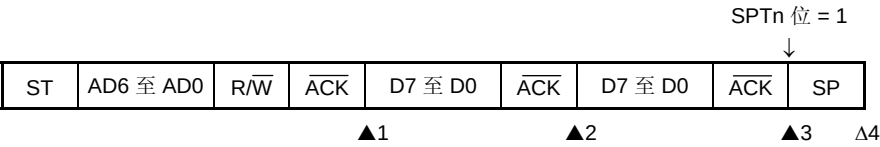
<R>

<R>

注 将 WTIMn 位设为 1 并更改生成中断请求信号 (INTIICn) 的时序以生成停止条件。

- 备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



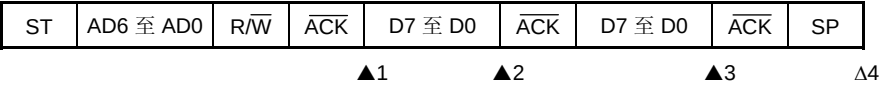
- ▲1: IICSn 寄存器 = 1010X110B
▲2: IICSn 寄存器 = 1010X100B
▲3: IICSn 寄存器 = 1010XX00B
Δ 4: IICSn 寄存器 = 00000001B

- 备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

17.7.2 从设备操作（当接收从地址时（地址匹配））

（1）开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止

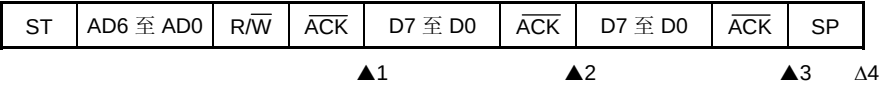
<1> 当 IICn.WTIMn 位 = 0 时



- ▲1: IICSn 寄存器 = 0001X110B
- ▲2: IICSn 寄存器 = 0001X000B
- ▲3: IICSn 寄存器 = 0001X000B
- Δ 4: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 IICn.SPIEn 位 = 1 时产生
 X: 不考虑
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



- ▲1: IICSn 寄存器 = 0001X110B
- ▲2: IICSn 寄存器 = 0001X100B
- ▲3: IICSn 寄存器 = 0001XX00B
- Δ 4: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

ST	AD6 至 AD0	$\overline{R\overline{W}}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	ST	AD6 至 AD0	$\overline{R\overline{W}}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
----	-----------	----------------------------	------------------	---------	------------------	----	-----------	----------------------------	------------------	---------	------------------	----

Δ5

Δ 5: IICSn 寄存器 = 00000001B

<2> 当 WTIMn 位 = 1 时（重启后，地址匹配）

ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	SP
----	-----------	-----	-----	---------	-----	----	-----------	-----	-----	---------	-----	----

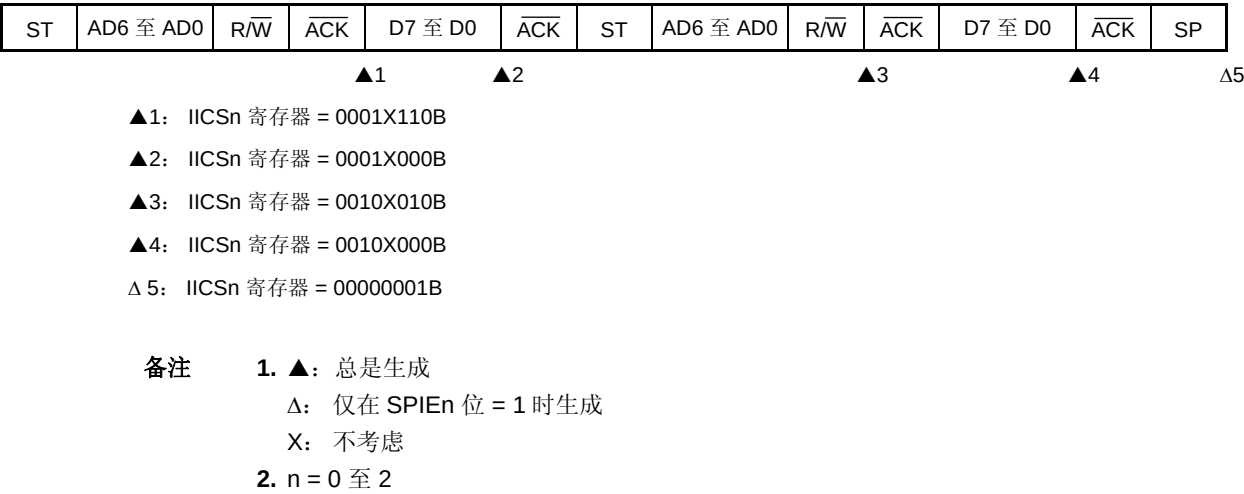
A5

Δ 5: IICSn 寄存器 = 00000001B

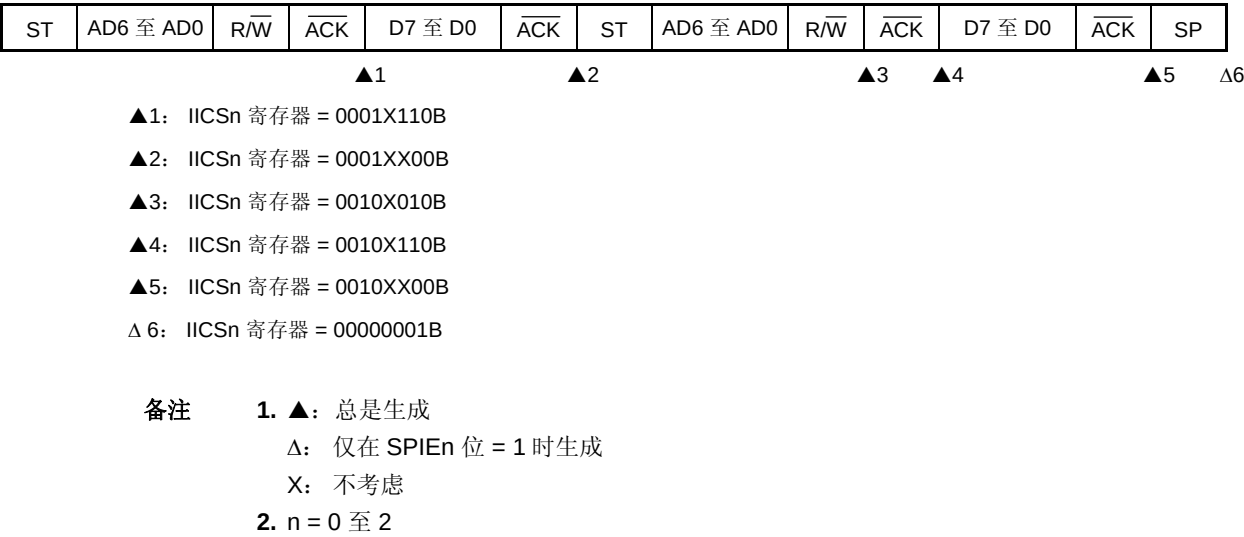
631

(3) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 代码 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0 时（重启后，地址不匹配（扩展码接收））



<2> 当 WTIMn 位 = 1 时（重启后，地址不匹配（扩展码接收））



<1> 当 WTIMn 位 = 0 时（重启后，地址不匹配（= 非扩展码））

ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	SP
			▲1		▲2				▲3			Δ4

▲1: IICSn 寄存器 = 0001X110B
 ▲2: IICSn 寄存器 = 0001X000B
 ▲3: IICSn 寄存器 = 00000110B
 Δ 4: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时（重启后，地址不匹配（= 非扩展码））

ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	SP
			▲1		▲2				▲3			Δ4

▲1: IICSn 寄存器 = 0001X110B
 ▲2: IICSn 寄存器 = 0001XX00B
 ▲3: IICSn 寄存器 = 00000110B
 Δ 4: IICSn 寄存器 = 00000001B

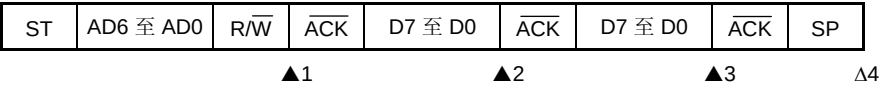
备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

17.7.3 从设备操作（当接收扩展码时）

<R> 设备在接收扩展码时，总是参与通信。

（1）开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止

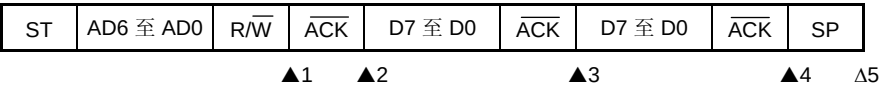
<1> 当 IICn.WTIMn 位 = 0 时



- ▲1: IICSn 寄存器 = 0010X010B
- ▲2: IICSn 寄存器 = 0010X000B
- ▲3: IICSn 寄存器 = 0010X000B
- Δ 4: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 IICn.SPIEn 位 = 1 时产生
 X: 不考虑
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



- ▲1: IICSn 寄存器 = 0010X010B
- ▲2: IICSn 寄存器 = 0010X110B
- ▲3: IICSn 寄存器 = 0010X100B
- ▲4: IICSn 寄存器 = 0010XX00B
- Δ 5: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

(2) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0 时（重启后，地址匹配）



备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

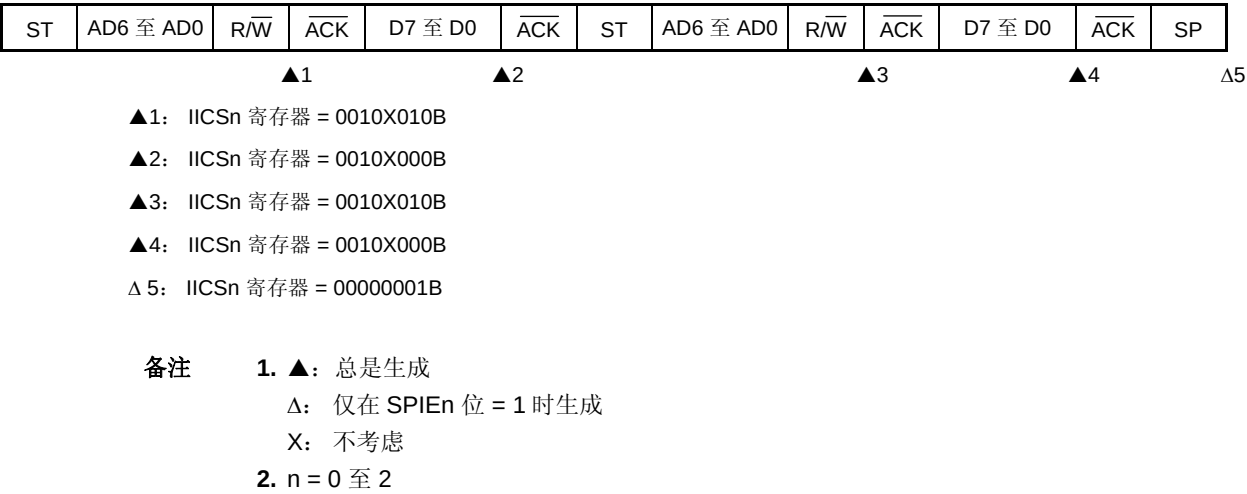
<2> 当 WTIMn 位 = 1 时（重启后，地址匹配）



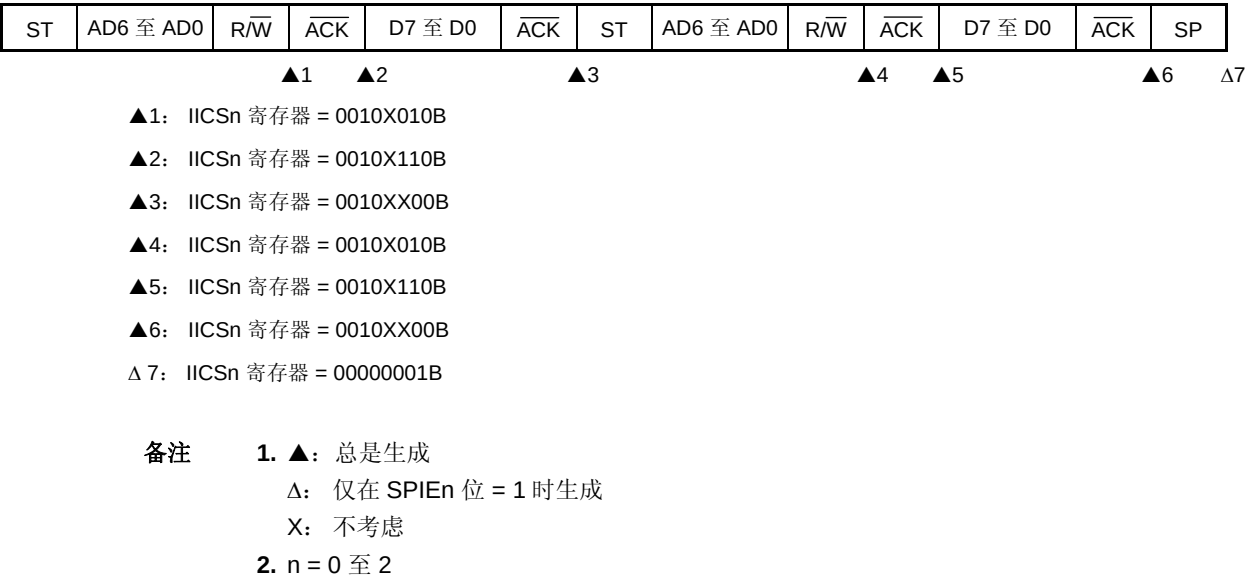
备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

(3) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 代码 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0 时（重启后，扩展码接收）



<2> 当 WTIMn 位 = 1 时（重启后，扩展码接收）





▲2: IICSn 寄存器 = 0010X000B

▲3. IICSn 寄存器 = 00000110B

Λ4: IICSn 寄存器 = 00000001B

Δ: 仅在 SPIEn 位 = 1 时生成

X: 不考慮

2. $n = 0$ 至 2



▲2: IICSn 寄存器 = 0010X110B

▲3. IICSn 寄存器 = 0010XX00B

▲4: IICSn 寄存器 = 00000110B

Λ 5: IICSp 寄存器 = 00000001B

Δ: 仅在 SPIEn 位 = 1 时生成

X: 不考慮

2. $n = 0$ 至 2

17.7.4 无通信时的操作

(1) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止



Δ1

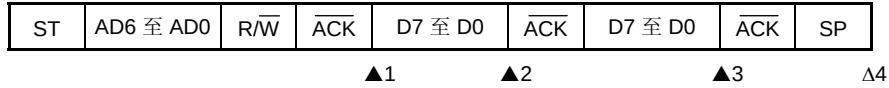
Δ 1: IICSn 寄存器 = 00000001B

- 备注
- 1. Δ: 仅在 SPIEn 位 = 1 时生成
 - 2. n = 0 至 2

17.7.5 仲裁失败操作（仲裁失败后作为从设备操作）

<R> 当设备在多主控系统中用作主设备时，每次生成 INTIICn 中断时都会读取 IICSn.MSTS_n 位以检查仲裁结果。

(1) 在从地址数据传送期间发生仲裁失败时

<1> 当 IICn.WTIM_n 位 = 0 时

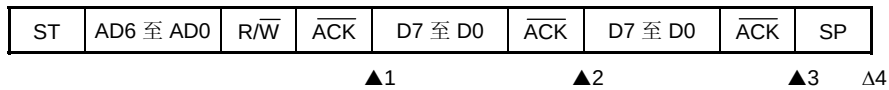
▲1: IICSn 寄存器 = 0101X110B

▲2: IICSn 寄存器 = 0001X000B

▲3: IICSn 寄存器 = 0001X000B

Δ 4: IICSn 寄存器 = 00000001B

- 备注
- ▲: 总是生成
Δ: 仅在 IICn.SPIEn 位 = 1 时产生
X: 不考虑
 - n = 0 至 2

<2> 当 WTIM_n 位 = 1 时

▲1: IICSn 寄存器 = 0101X110B

▲2: IICSn 寄存器 = 0001X100B

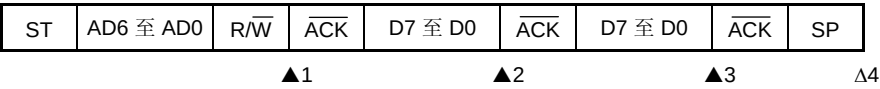
▲3: IICSn 寄存器 = 0001XX00B

Δ 4: IICSn 寄存器 = 00000001B

- 备注
- ▲: 总是生成
Δ: 仅在 SPIEn 位 = 1 时生成
X: 不考虑
 - n = 0 至 2

(2) 在扩展码传送期间发生仲裁失败时

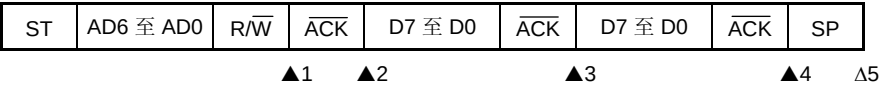
<1> 当 WTIMn 位 = 0 时



- ▲1: IICSn 寄存器 = 0110X010B
- ▲2: IICSn 寄存器 = 0010X000B
- ▲3: IICSn 寄存器 = 0010X000B
- Δ 4: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



- ▲1: IICSn 寄存器 = 0110X010B
- ▲2: IICSn 寄存器 = 0010X110B
- ▲3: IICSn 寄存器 = 0010X100B
- ▲4: IICSn 寄存器 = 0010XX00B
- Δ 5: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

17.7.6 仲裁失败时的操作（仲裁失败后无通信）

<R> 当设备在多主控系统中用作主设备时，每次生成 INTIICn 中断时都会读取 IICSn.MSTSn 位以检查仲裁结果。

(1) 在从地址数据传送期间发生仲裁失败

ST	AD6 至 AD0	R/W	$\overline{\text{ACK}}$	D7 至 D0	$\overline{\text{ACK}}$	D7 至 D0	$\overline{\text{ACK}}$	SP
----	-----------	-----	-------------------------	---------	-------------------------	---------	-------------------------	----

▲1

▲1: IICSn 寄存器 = 01000110B

Δ 2: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是生成

Δ: 仅在 IICn.SPIEn 位 = 1 时产生

2. n = 0 至 2

(2) 在扩展码传送期间发生仲裁失败时

ST	AD6 至 AD0	R/W	$\overline{\text{ACK}}$	D7 至 D0	$\overline{\text{ACK}}$	D7 至 D0	$\overline{\text{ACK}}$	SP
----	-----------	-----	-------------------------	---------	-------------------------	---------	-------------------------	----

▲1

▲1: IICSn 寄存器 = 0110X010B

IICn.LRELn 位由软件置 1

Δ 2: IICSn 寄存器 = 00000001B

备注

1. ▲: 总是生成

Δ: 仅在 SPIEn 位 = 1 时生成

X: 不考虑

2. n = 0 至 2

<1> 当 IICn.WTIMn 位 = 0 时

ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	D7 至 D0	ACK	SP
----	-----------	-----	-----	---------	-----	---------	-----	----

▲1 ▲2 Δ3

▲1: IICSn 寄存器 = 10001110B
 ▲2: IICSn 寄存器 = 01000000B
 Δ 3: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	D7 至 D0	ACK	SP
----	-----------	-----	-----	---------	-----	---------	-----	----

▲1 ▲2 Δ3

▲1: IICSn 寄存器 = 10001110B
 ▲2: IICSn 寄存器 = 01000100B
 Δ 3: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 2. n = 0 至 2

(4) 在数据传送期间由于重启条件致使仲裁失败时

<1> 非扩展码（例如：地址不匹配）

ST	AD6 至 AD0	R/W	ACK	D7 至 Dn	ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	SP
----	-----------	-----	-----	---------	----	-----------	-----	-----	---------	-----	----

▲1

▲2

Δ3

▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 01000110B

Δ 3: IICSn 寄存器 = 00000001B

- 备注
- 1. ▲: 总是生成
Δ: 仅在 SPIEn 位 = 1 时生成
X: 不考虑
 - 2. Dn = D6 至 D0
n = 0 至 2

<2> 扩展码

ST	AD6 至 AD0	R/W	ACK	D7 至 Dn	ST	AD6 至 AD0	R/W	ACK	D7 至 D0	ACK	SP
----	-----------	-----	-----	---------	----	-----------	-----	-----	---------	-----	----

▲1

▲2

Δ3

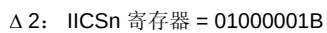
▲1: IICSn 寄存器 = 1000X110B

▲2: IICSn 寄存器 = 0110X010B

IICcn.LRELn 位由软件置 1

Δ 3: IICSn 寄存器 = 00000001B

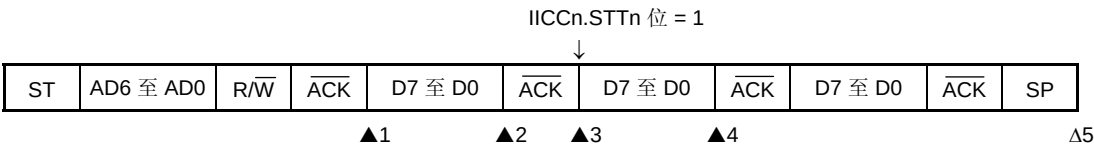
- 备注
- 1. ▲: 总是生成
Δ: 仅在 SPIEn 位 = 1 时生成
X: 不考虑
 - 2. Dn = D6 至 D0
n = 0 至 2



用户手册 U16541CA5V1UD

(6) 当试图产生重启条件时由于 SDA0n 引脚的低电平致使仲裁失败时

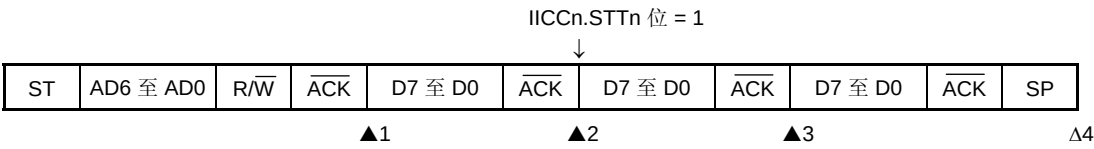
<1> 当 WTIMn 位 = 0 时



- ▲1: IICSn 寄存器 = 1000X110B
▲2: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)
▲3: IICSn 寄存器 = 1000X100B (WTIMn 位 = 0)
▲4: IICSn 寄存器 = 01000000B
Δ 5: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

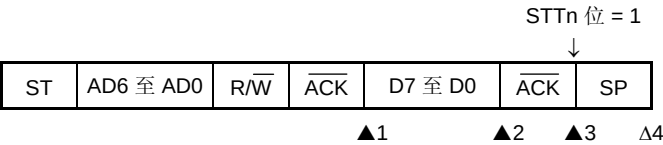


- ▲1: IICSn 寄存器 = 1000X110B
▲2: IICSn 寄存器 = 1000X100B
▲3: IICSn 寄存器 = 01000100B
Δ 4: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

(7) 试图产生重启条件时由于停止条件致使仲裁失败时

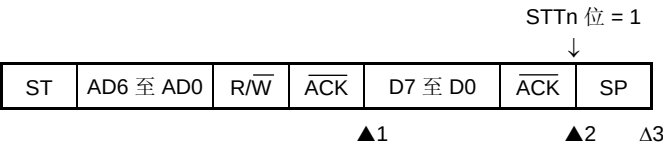
<1> 当 WTIMn 位 = 0 时



- ▲1: IICSn 寄存器 = 1000X110B
▲2: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)
▲3: IICSn 寄存器 = 1000XX00B
Δ4: IICSn 寄存器 = 01000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

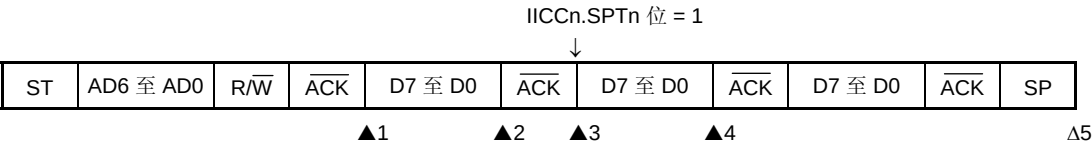


- ▲1: IICSn 寄存器 = 1000X110B
▲2: IICSn 寄存器 = 1000XX00B
Δ3: IICSn 寄存器 = 01000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

(8) 试图产生停止条件时由于 SDA0n 引脚的低电平致使仲裁失败时

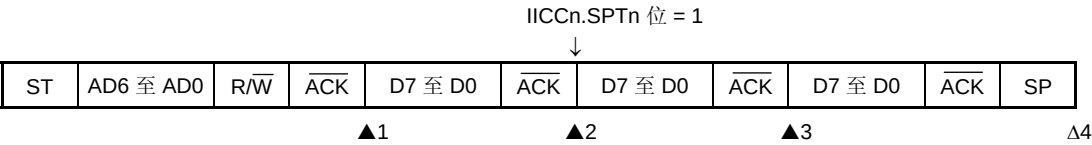
<1> 当 WTIMn 位 = 0 时



- ▲1: IICSn 寄存器 = 1000X110B
▲2: IICSn 寄存器 = 1000X000B (WTIMn 位 = 1)
▲3: IICSn 寄存器 = 1000X100B (WTIMn 位 = 0)
▲4: IICSn 寄存器 = 01000100B
Δ 5: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



- ▲1: IICSn 寄存器 = 1000X110B
▲2: IICSn 寄存器 = 1000X100B
▲3: IICSn 寄存器 = 01000100B
Δ 4: IICSn 寄存器 = 00000001B

备注 1. ▲: 总是生成
 Δ: 仅在 SPIEn 位 = 1 时生成
 X: 不考虑
 2. n = 0 至 2

17.8 中断请求信号（INTIICn）的产生时序和等待控制

由 IICn.WTIMn 位的设置会决定时序，这个时序是 INTIICn 寄存器产生和对应等待控制的依据，如下所示（n = 0 至 2）。

表 17-3. INTIICn 产生时序和等待控制

WTIMn 位	从设备操作期间			主设备操作期间		
	地址	数据接收	数据传送	地址	数据接收	数据传送
0	9 ^{注 1、2}	8 ^{注 2}	8 ^{注 2}	9	8	8
1	9 ^{注 1、2}	9 ^{注 2}	9 ^{注 2}	9	9	9

- 注
1.

只有与 SVAn 寄存器设置的地址匹配时，在第 9 个时钟下降沿才会产生从设备的 INTIICn 信号和等待区间。
这时，不管设置到 IICn.ACKEn 位的值为何，均产生 $\overline{ACK}$ 。对于接收扩展码的从设备来说，INTIICn 信号在第 8 个时钟下降沿产生。
重启后地址如果不匹配，INTIICn 信号会在第 9 个时钟的下降沿产生，但无等待产生。
2.

如果接收到的地址与 SVAn 寄存器的内容不匹配，并且也没有接收到扩展码，那么 INTIICn 信号和等待都不会产生。

- 备注
1.

表中的数字表示串行时钟的时钟信号个数。中断请求和等待控制都和这些时钟信号的下降沿同步。
2.

n = 0 至 2

- (1)

地址传送 / 接收期间

•

从设备操作：决定中断和等待时序时不考虑 WTIMn 位。

•

主设备操作：不管 WTIMn 位是多少，中断和等待时序都会在第 9 个时钟的下降沿处出现。

(2)

数据接收期间

•

主 / 从设备操作：根据 WTIMn 位来决定中断和等待时序。

(3)

数据传送期间

•

主 / 从设备操作：根据 WTIMn 位来决定中断和等待时序。

(4) 等待取消方法

四种等待取消方法如下所示。

- 将数据写入到 IICn 寄存器
- 设置 IICn.WRELn 位为 1（等待取消）
- 设置 IICn.STTn 位为 1（产生启动条件）^注
- 设置 IICn.SPTn 位为 1（产生停止条件）^注

注 仅限主设备

当选择 8 个时钟等待（WTIMn 位 = 0）时， $\overline{ACK}$ 产生与否必须要在等待取消之前决定。

备注 n = 0 至 2

(5) 停止条件检测

当检测到停止条件时，产生 INTIICn 信号。

备注 n = 0 至 2

17.9 地址匹配检测方法

I²C 总线模式中，主设备通过传送对应地址来选择特定的从设备。

地址匹配检测由硬件自动完成。当本地地址已经设置到 SVAn 寄存器中，并且当设置到 SVAn 寄存器的地址与主设备发送的从地址匹配时，或者接收到扩展码时，产生 INTIICn 信号。（n = 0 到 2）。

17.10 错误检测

I²C 总线中，数据传送期间串行数据总线引脚（SDA0n）的状态由传送设备的 IICn 寄存器捕捉，因此先于传送的 IICn 寄存器的数据与传送的 IICn 数据进行比较，从而允许传送错误检测。当比较数据值不匹配时，则判断为发生了传送错误（n = 0 到 2）。

17.11 扩展码

(1) 当接收地址的高 4 位为 0000 或者 1111，扩展码标志 (IICSn.EXCn 位) 设置为扩展码接收，并且第 8 个时钟下降沿产生中断请求 (INTIICn) (n = 0 到 2)。

存储在 SVAn 寄存器里的本地地址不受影响。

(2) 如果 SVAn 寄存器通过 10 位地址传送设置为 11110xx0，并且 11110xx0 从主设备传送，结果如下。注意 INTIICn 信号在第 8 个时钟下降沿产生 (n = 0 到 2)。

- 数据的高 4 位匹配: EXCn 位 = 1
- 7 位数据匹配: IICSn.COIn 位 = 1

(3) 由于中断请求信号产生后的处理会根据扩展码后面的数据有所不同，所以由软件来完成这些处理。

<R> 设备被用作从设备时，即使地址不匹配，它也会在接收扩展码时参与通信。
例如，接收到扩展码后，不希望有从设备操作，因此设置 IICn.LRELn 位为 1，而 CPU 将进入下一个通信等待状态。

表 17-4. 扩展码位定义

从地址	R/W 位	说明
0000 000	0	通用调用地址
0000 000	1	开始字节
0000 001	X	CBUS 地址
0000 010	X	预留给不同总线格式的地址
1111 0xx	X	10 位从地址规定

17.12 仲裁

当几个主设备同时产生启动条件时（在 IICSn.STDn 位设置为 1 之前，IICn.STTn 位已经置 1），主设备之间会进行通信，同时对时钟的个数进行调整直到数据有所不同。这种操作称为仲裁（n = 0 到 2）。

当某个主设备仲裁失败时，通过产生仲裁失败的时序将仲裁失败标志（IICSn.ALDn 位）置 1，SCL0n 和 SDA0n 线均设置为高阻态，从而释放总线（n = 0 到 2）。

仲裁失败的检测基于下一个中断请求信号（INTIICn）的时序（当检测到停止条件等等，第 8 个或者第 9 个时钟）以及 ALDn 位置 1，这些都由软件完成（n = 0 到 2）。

中断请求信号时序的详细情况，请看 17.7 I²C 中断请求信号（INTIICn）。

图 17-14. 仲裁时序实例

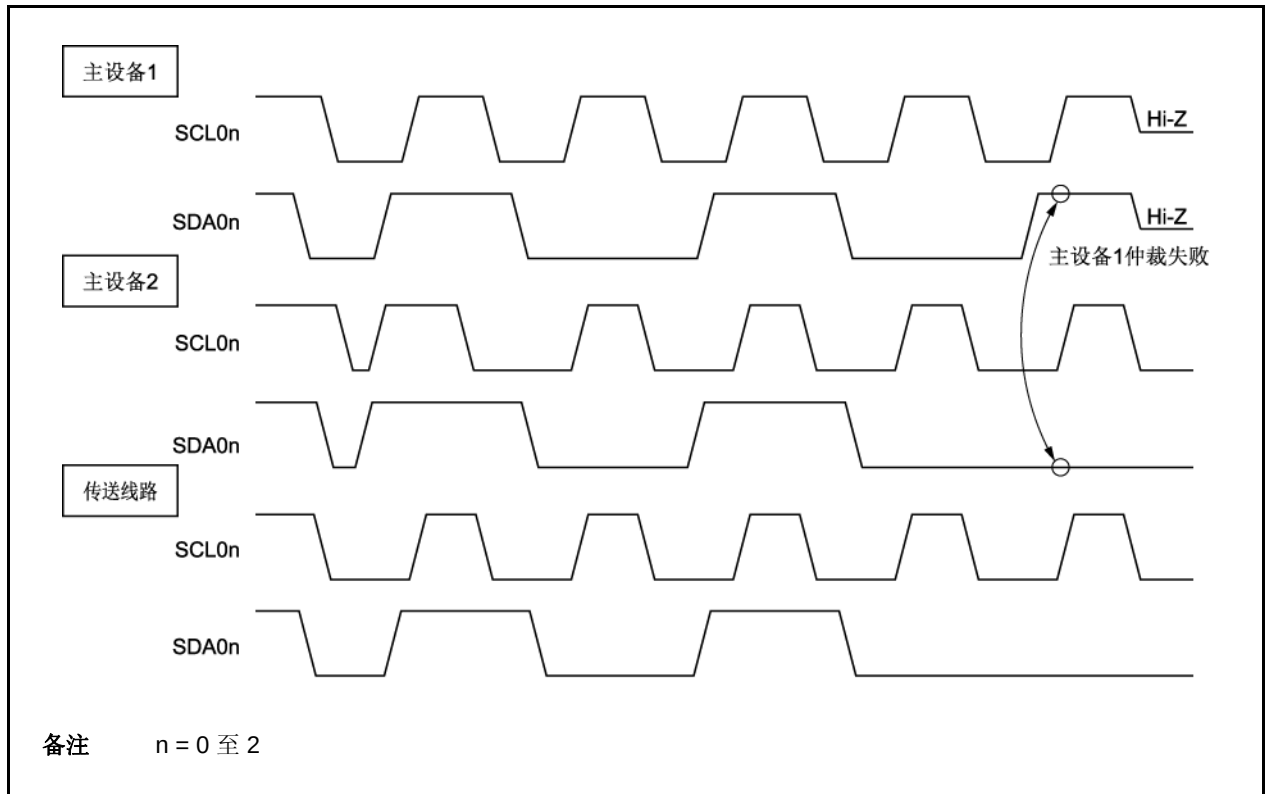


表 17-5. 仲裁和中断请求信号产生时序期间的状态

仲裁期间的状态	中断请求产生时序
发送地址传送	在字节传送后第 8 个或者第 9 个时钟的下降沿处 ^{注 1}
地址传送后读取 / 写入数据	
传送扩展码	
扩展码传送后读取 / 写入数据	
传送数据	
数据接收后 $\overline{ACK}$ 传送期间	
数据传送期间当检测到重启条件时	
数据传送期间当检测到停止条件时	检测到停止条件（当 IICn.SPIEn 位 = 1） ^{注 2}
SDA0n 引脚为低电平时，试图产生重启条件	在字节传送后第 8 个或者第 9 个时钟的下降沿处 ^{注 1}
检测到停止条件时，试图产生重启条件。	检测到停止条件（当 IICn.SPIEn 位 = 1） ^{注 2}
DSA0n 引脚为低电平时，试图产生停止条件	在字节传送后第 8 个或者第 9 个时钟的下降沿处 ^{注 1}
SCL0n 引脚为低电平时，试图产生重启条件	

- 注
1. 当 IICn.WTIMn 位 = 1，在第 9 个时钟的下降沿产生 INTIICn 信号。当 WTIMn 位 = 0，并且接收到扩展码展的从地址时，在第 8 个时钟的下降沿产生 INTIICn 信号（n = 0 到 2）。
 2. 可能发生仲裁产生时，设置 SPIEn 位为 1，用于主设备操作（n = 0 到 2）。

17.13 唤醒功能

I²C 总线的另一个功能就是当接收到本地地址和扩展码时产生中断请求信号（INTIICn）。

这个功能可以提高处理效率，避免在地址不匹配时产生不必要的 INTIICn 信号。

检测到启动条件时，设置唤醒待机模式。可能由于仲裁失败而致使主设备（已经产生启动条件）变为从设备的情况下，地址传送期间唤醒待机模式有效。

然而，当检测到停止条件时，不管唤醒功能如何，都会设置 IICn.SPIEn 位，并且还决定了 INTIICn 信号是允许或被禁止（n = 0 到 2）。

17.14 通信保留

17.14.1 通信保留功能使能时 (IICFn.IICRSVn 位 = 0)

总线空闲期间为了启动主设备通信，可以进行通信保留，来允许在总线释放时传送启动条件。在两种模式下总线未使用。

- 当仲裁结果导致主操作和从操作都无动作。
- 当接收到扩展码，并且从操作禁止时（当 IICCn.LRELn 位设置为 1 时，未返回 $\overline{ACK}$ ，总线释放）（n = 0 到 2）。

当总线不使用时，如果 IICCn.STTn 位设置为 1，总线释放后（检测到停止条件后），自动产生启动条件并设置等待状态。

<R> 在 IICCn.SPIEn 位被设为 1 且已经通过生成中断请求（INTIICn）检测到总线释放（也就是检测到停止条件）后，设备会在地址写入 IICn 寄存器时作为主设备来进行通信。在检测到停止条件前写入 IICn 寄存器的数据无效。

当 STTn 位已经设置为 1 时，操作模式（作为启动条件或者作为通信保留）根据总线状态决定（n = 0 到 2）。

如果总线已经释放产生启动条件
如果总线还没有释放（待机模式）通信保留

为了检测已经为 STTn 位决定了何种操作模式，则设置 STTn 位为 1，接着在经过等待时间后检验 IICSn.MSTS_n 位（n = 0 到 2）。

由软件设置的等待时间如表 17-6 所示。等待时间可以由 IICCLn 寄存器的 SMCn，CLn1 和 CLn0 位以及 IICXn.CLXn 位来进行设置（n = 0 到 2）。

表 17-6. 等待时间

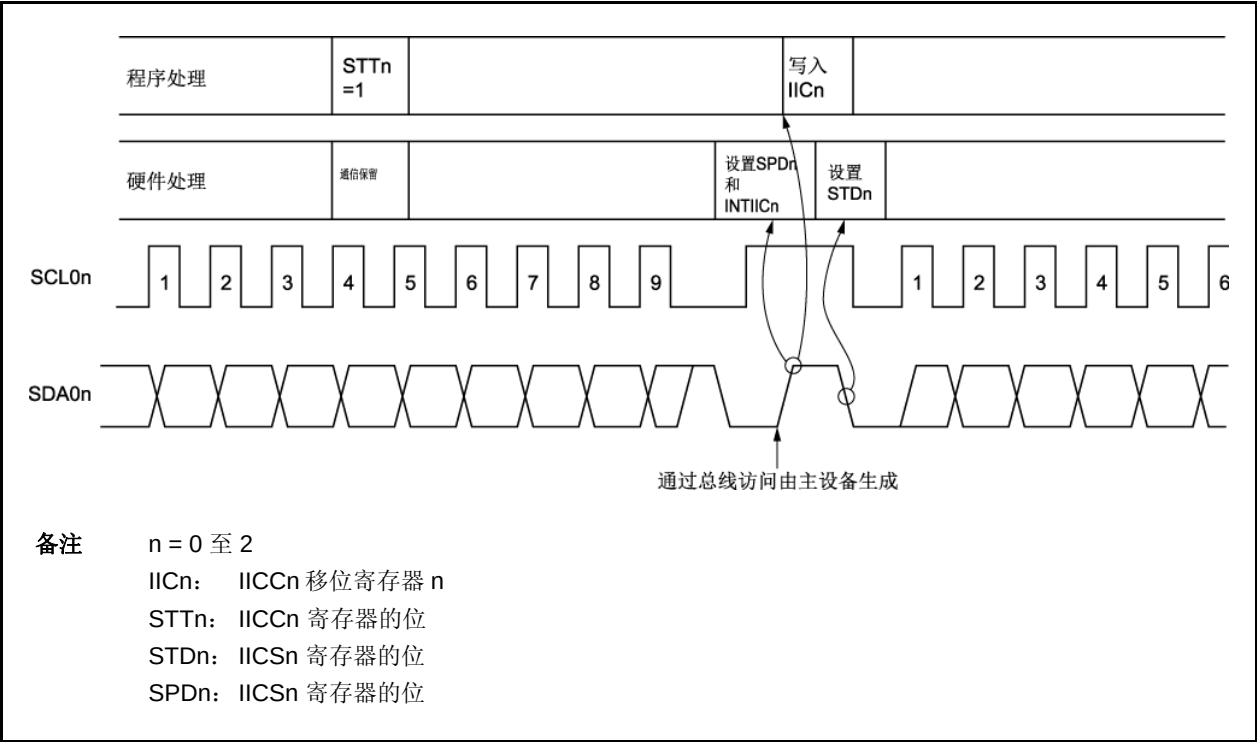
时钟选择	CLXn	SMCn	CLn1	CLn0	等待时间
f _{xx} (设置 OCKSm = 18H 时)	0	0	0	0	26 个时钟
f _{xx} /2 (设置 OCKSm = 10H 时)	0	0	0	0	52 个时钟
f _{xx} /3 (设置 OCKSm = 11H 时)	0	0	0	0	78 个时钟
f _{xx} /4 (设置 OCKSm = 12H 时)	0	0	0	0	104 个时钟
f _{xx} /5 (设置 OCKSm = 13H 时)	0	0	0	0	130 个时钟
f _{xx} (设置 OCKSm = 18H 时)	0	0	0	1	47 个时钟
f _{xx} /2 (设置 OCKSm = 10H 时)	0	0	0	1	94 个时钟
f _{xx} /3 (设置 OCKSm = 11H 时)	0	0	0	1	141 个时钟
f _{xx} /4 (设置 OCKSm = 12H 时)	0	0	0	1	188 个时钟
f _{xx} /5 (设置 OCKSm = 13H 时)	0	0	0	1	235 个时钟 ^注
f _{xx}	0	0	1	0	47 个时钟
f _{xx} (设置 OCKSm = 18H 时)	0	1	0	×	16 个时钟
f _{xx} /2 (设置 OCKSm = 10H 时)	0	1	0	×	32 个时钟
f _{xx} /3 (设置 OCKSm = 11H 时)	0	1	0	×	48 个时钟
f _{xx} /4 (设置 OCKSm = 12H 时)	0	1	0	×	64 个时钟
f _{xx} /5 (设置 OCKSm = 13H 时)	0	1	0	×	80 个时钟 ^注
f _{xx}	0	1	1	0	16 个时钟
f _{xx} (设置 OCKSm = 18H 时)	1	1	0	×	10 个时钟
f _{xx} /2 (设置 OCKSm = 10H 时)	1	1	0	×	20 个时钟
f _{xx} /3 (设置 OCKSm = 11H 时)	1	1	0	×	30 个时钟
f _{xx} /4 (设置 OCKSm = 12H 时)	1	1	0	×	40 个时钟
f _{xx} /5 (设置 OCKSm = 13H 时)	1	1	0	×	50 个时钟 ^注
f _{xx}	1	1	1	0	10 个时钟

<R> 注 仅限 V850ES/SG2-H

- 备注
1. n = 0 至 2
m = 0, 1
 2. × = 不考虑

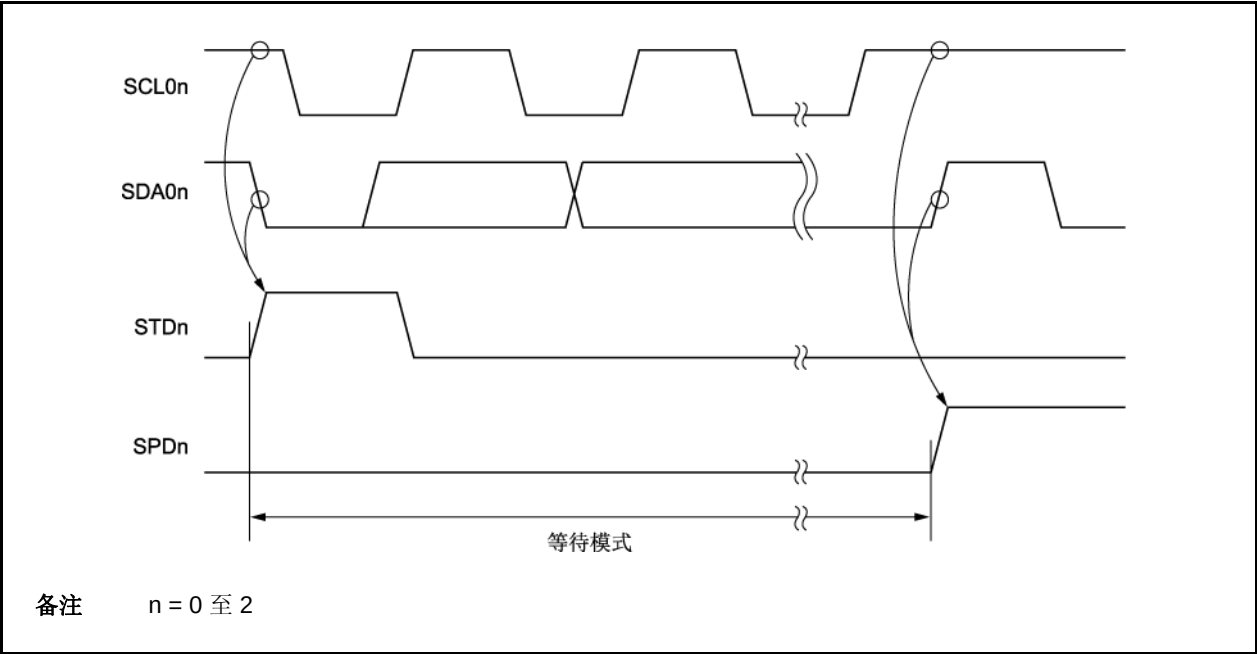
通信保留的时序如下所示。

图 17-15. 通信保留时序



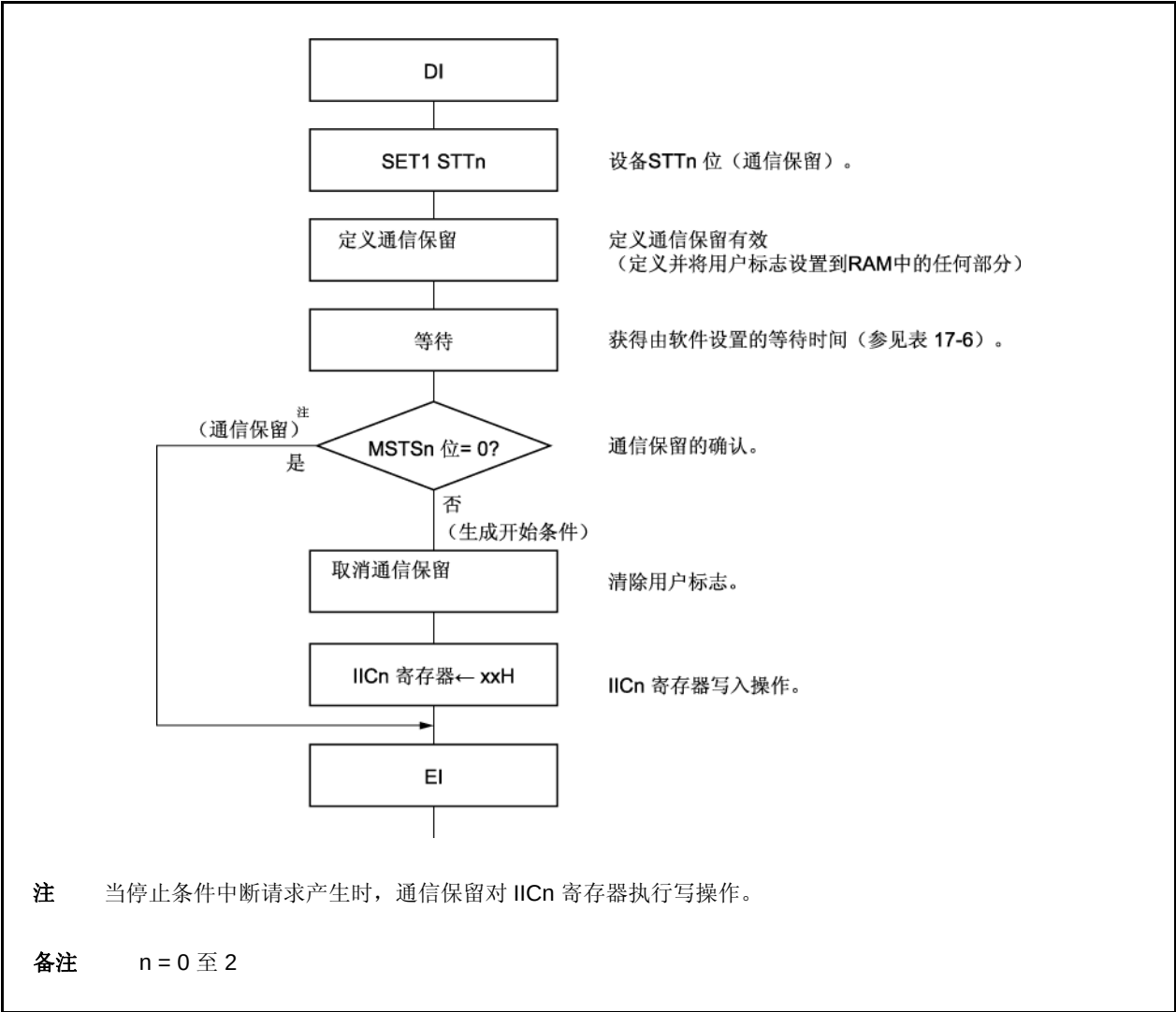
通过下列时序接受通信保留。在 IICSn.STDn 位置 1 后，在检测到停止条件之前通过对 IICn.STTn 位置 1 来进行通信保留设置（n = 0 到 2）。

图 17-16. 接受通信保留的时序



通信保留流程图如下阐释。

图 17-17. 通信保留流程图



17.14.2 通信保留功能禁止时 (IICFn.IICRSVn 位 = 1)

总线通信期间，总线未用于通信时，如果 IICn.STTn 位被设置，则拒绝请求，也不产生启动条件。在两种模式下总线未使用。

- 当仲裁结果导致主操作和从操作都无动作。
- 当接收到扩展码，并且从操作禁止时（当 IICn.LRELn 位设置为 1 时，未返回 $\overline{\text{ACK}}$ ，总线释放）（n = 0 到 2）。

为了确认是否产生开始条或请求是否被拒绝，检验 IICFn.STCFn 标志。表 17-7 显示的是 STTn 设置为 1 后直到设置 STCFn 标志之前所需要的时间。因此，请通过软件确保时间。

<R>

表 17-7. 等待时间

选择时钟	OCKSE Nm	OCKSTH m	OCKSm 1	OCKSm 0	CLn1	CLn0	等待时间
f _{xx} /2（设置 OCKSm = 10H 时）	1	0	0	0	0	X	6 个时钟
f _{xx} /3（设置 OCKSm = 11H 时）	1	0	0	1	0	X	9 个时钟
f _{xx} /4（设置 OCKSm = 12H 时）	1	0	1	0	0	X	12 个时钟
f _{xx} /5（设置 OCKSm = 13H 时）	1	0	1	1	0	X	15 个时钟
f _{xx} （设置 OCKSm = 18H 时）	1	1	0	0	0	X	3 个时钟
f _{xx}	0	0	0	0	1	0	3 个时钟
f _{xx} /2（设置 OCKSm = 10H 时）	1	0	0	0	1	1	6 个时钟
f _{xx} /3（设置 OCKSm = 11H 时）	1	0	0	1	1	1	9 个时钟
f _{xx} /4（设置 OCKSm = 12H 时）	1	0	1	0	1	1	12 个时钟
f _{xx} /5（设置 OCKSm = 13H 时）	1	0	1	1	1	1	15 个时钟
f _{xx} （设置 OCKSm = 18H 时）	1	1	0	0	1	1	3 个时钟

- 备注**
1. x: 不考虑
 2. n = 0 至 2
m = 0, 1
 3. 时钟 = f_{xx}（主时钟频率）

17.15 注意事项

(1) 当 IICFn.STCENn 位 = 0 时

I²C0n 操作允许后，不管实际总线状态为何，立即识别总线通信状态（IICFn.IICBSYn 位 = 1）。在还没有检测到停止条件的状态下，执行主设备通信，在开始主设备通信之前，产生停止条件并释放总线。
依照下列顺序产生停止条件。

<1>设置 IICCLn 寄存器。

<2>设置 IICCN.IICEn 位。

<3>设置 IICCN.SPTn 位。

(2) 当 IICFn.STCENn 位 = 1 时

I²C0n 操作允许后，不管实际的总线状态为何，立即识别到总线释放状态（IICBSYn 位 = 0）。为了产生第一个启动条件（IICCN.STTn 位 = 1）必须要确认总线已经被释放，以免干扰其它通信。

<R> (3) 在与其它设备进行通信的过程中，当 V850ES/SG2 和 V850ES/SG2-H 的 IICCN.IICEn 位设置为 1 时，是否检测到启动条件要根据通信线的状态来决定。当 SCL0n 和 SDA0n 线为高电平时，请确保设置 IICCN.IICEn 位为 1。

(4) 允许操作（IICCN.IICEn 位 = 1）之前，由 IICCLn, IICXn 和 OCKSm 寄存器决定操作时钟频率。如需改变操作时钟频率，对 IICCN.IICEn 位清 0。

(5) IICCN.STTn 和 IICCN.SPTn 位置 1 后，在没有清零的情况下，请勿重置为 1。

(6) 如果通信已经保留，设置 IICCN.SPIEn 位为 1，以便通过检测停止条件产生中断请求。中断请求信号产生后，将通信数据写入 I2Cn 寄存器就可以释放等待状态，然后开始传输。如果中断请求不是通过检测到停止条件而产生的，那么传输会停止在等待状态，因为没有产生中断请求。但是，没有必要对 SPIEn 位置 1 来检测 IICSn.MSTSn 位。

备注 n = 0 至 2

<R>

17.16 通信操作

以下通过流程图显示了三种操作步骤。

(1) 在单个主系统中的主设备操作

以下显示了在单个主系统中将 V850ES/SG2 或 V850ES/SG2-H 用作主设备时的流程图。

该流程图在总体上被分为初始设置和通信处理。在启动时执行初始设置。如果需要使用从设备进行通信，那么应先准备通信，然后执行通信处理。

(2) 在多主控系统中的主设备操作

在 I²C0n 总线多主控系统中，当主线参与通信时，将不能通过 I²C 总线规范来判断总线是被释放还是被使用。此时，当数据和时钟在一段时间（1 帧）内处于高电平时，V850ES/SG2 或 V850ES/SG2-H 将会在总线释放状态下参数通信。

该流程图在总体上被分为初始设置，通信等待，以及通信处理。V850ES/SG2 或 V850ES/SG2-H 仲裁失败且被指定为从设备时的处理在这里将会被忽略，这里只会显示作为主设备时的处理。在启动时执行初始设置以参与通信。然后，等待作为主设备的通信请求或等待作为从设备的指定。实际的通信会在通信处理过程中进行，它支持使用从设备的传送 / 接收以及使用其他主设备进行的仲裁。

(3) 从设备操作

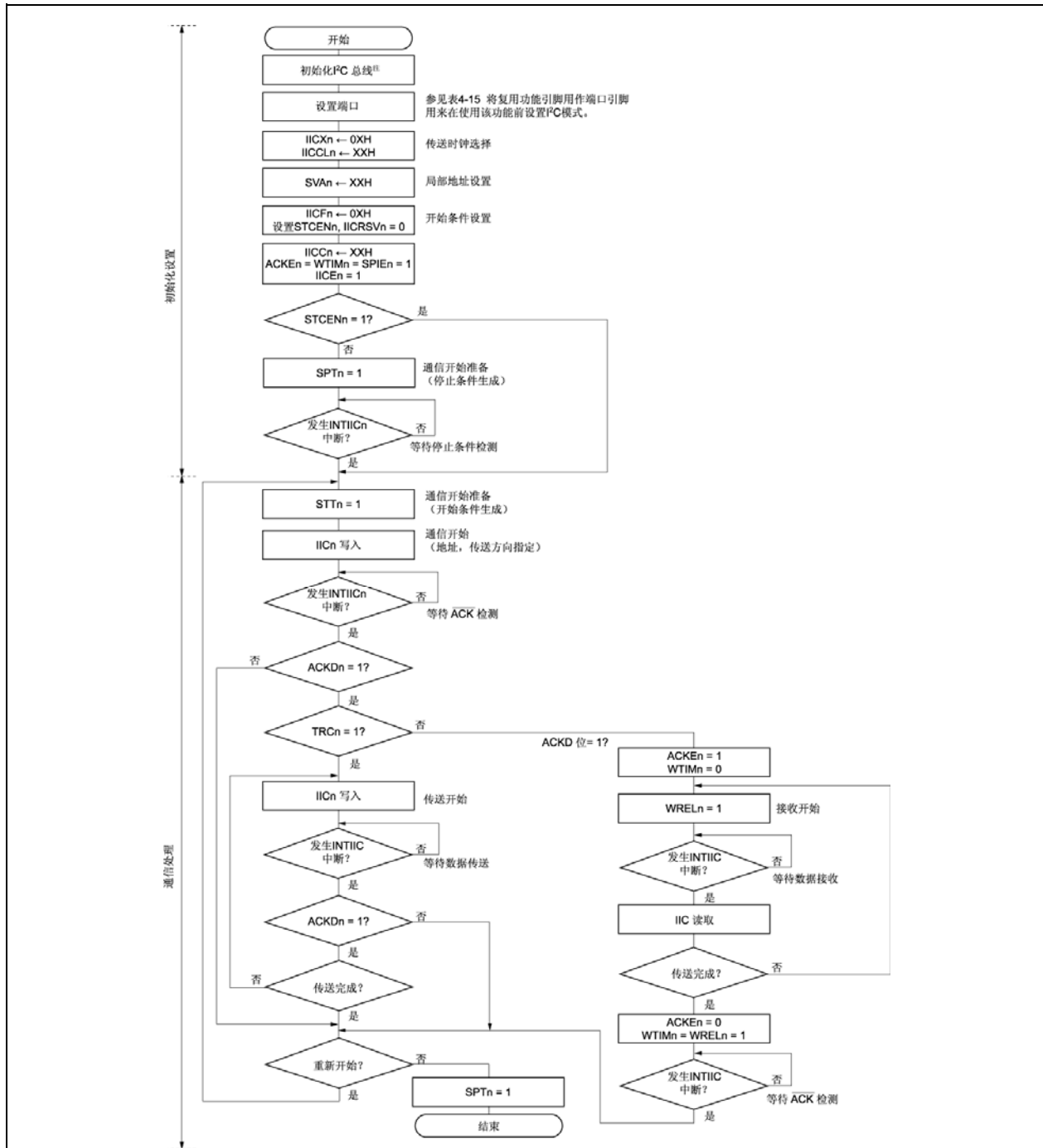
以下显示了 V850ES/SG2 或 V850ES/SG2-H 被用作 I²C0n 总线的从设备时的示例。

使用从设备时，操作会通过中断开始。在启动时执行初始设置，然后等待 INTIICn 中断的发生（通信等待）。INTIICn 中断发生时，会对通信状态进行判断，而判断结果会作为一个标志在整个处理过程中进行传递。通过检查标志，可以完成必要的通信处理。

备注 n = 0 至 2

17.16.1 在单个主系统中的主设备操作

图 17-18. 在单个主系统中的主设备操作



注 按照通信中的产品规范来释放 I²C0n 总线（SCL0n, SDA0n 引脚 = 高电平）。
例如，当 EEPROMTM 输出低电平到 SDA0n 引脚时，则应对输出端口的 SCL0n 引脚以及该输出端口中的输出时钟脉冲进行设置直到 SDA0n 引脚持续为高电平。

备注

1. 传送和接收格式应符合通信中的产品规范。
2. n = 0 至 2

17.16.2 在多主控系统中的主设备操作

图 17-19. 在多主控系统中的主设备操作 (1/3)

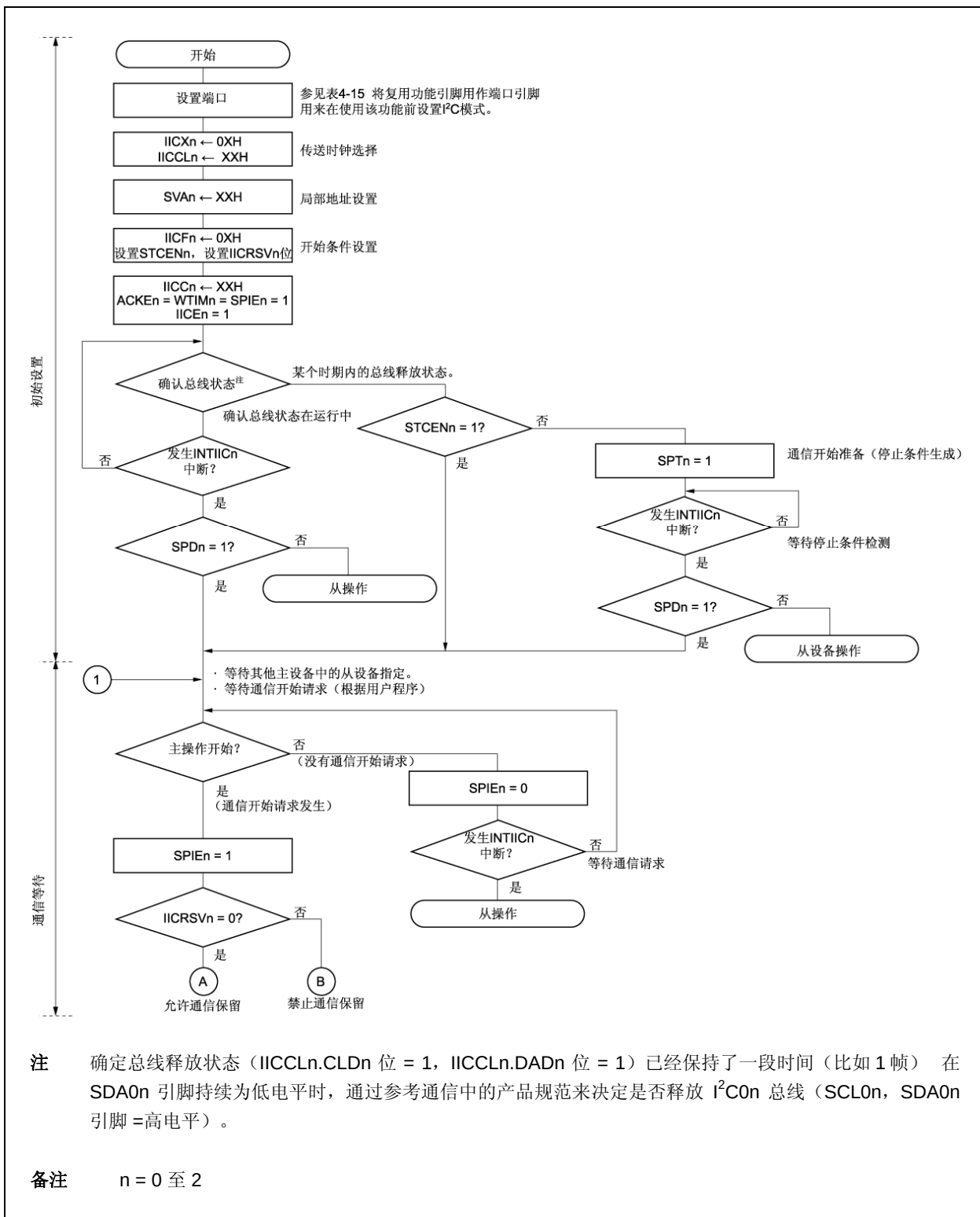


图 17-19. 在多主控系统中的主设备操作 (2/3)

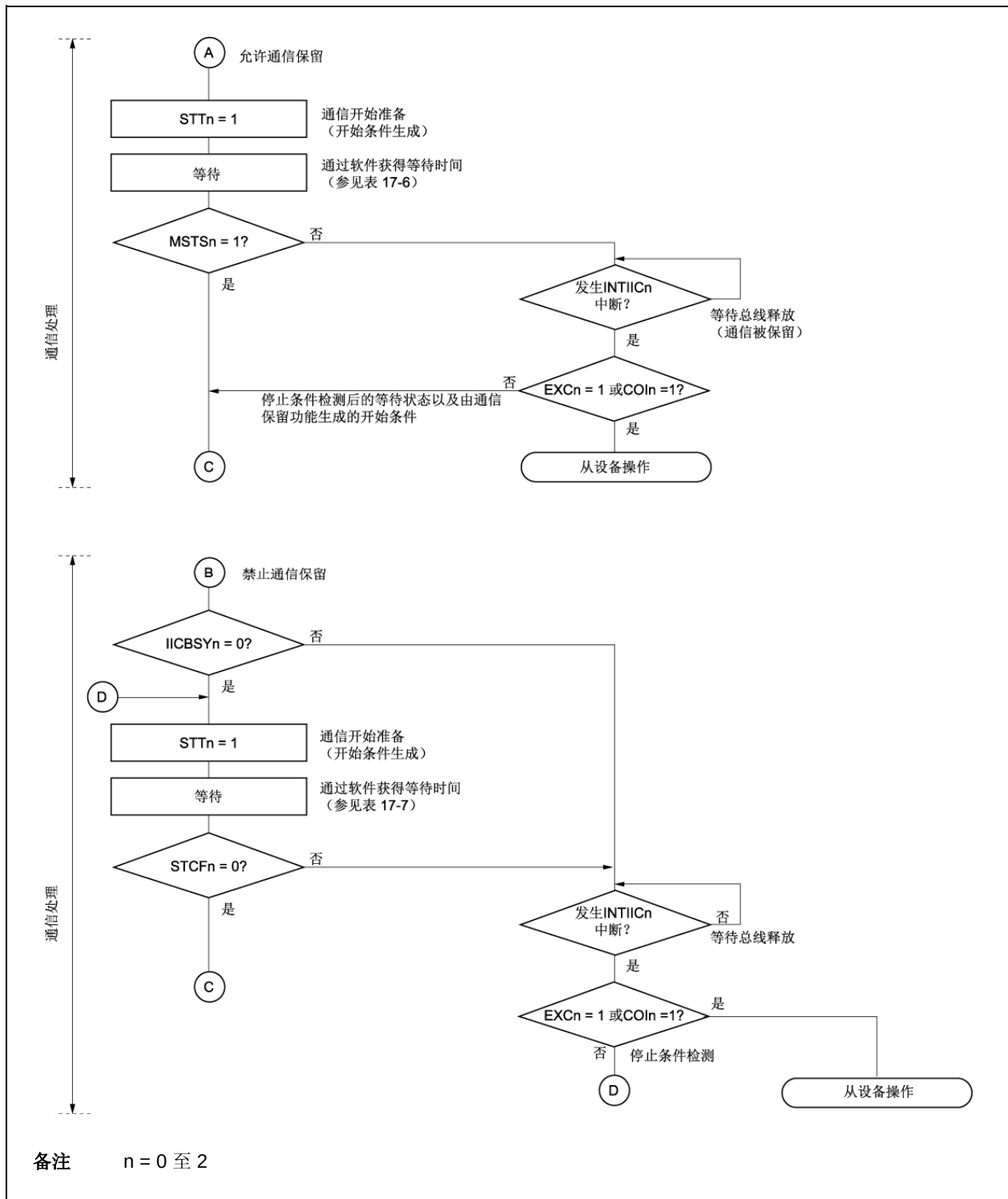
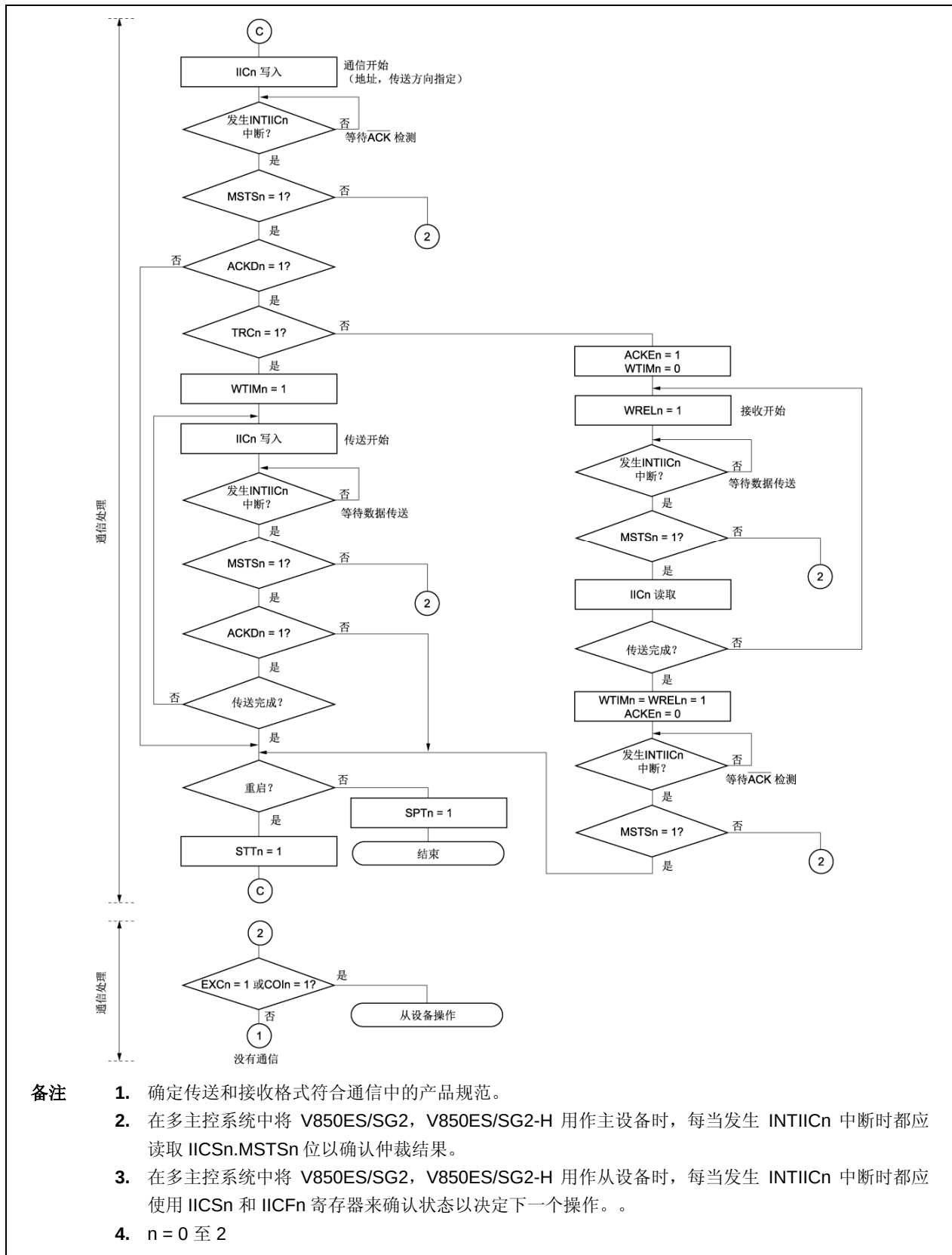


图 17-19. 在多主控系统中的主设备操作 (3/3)



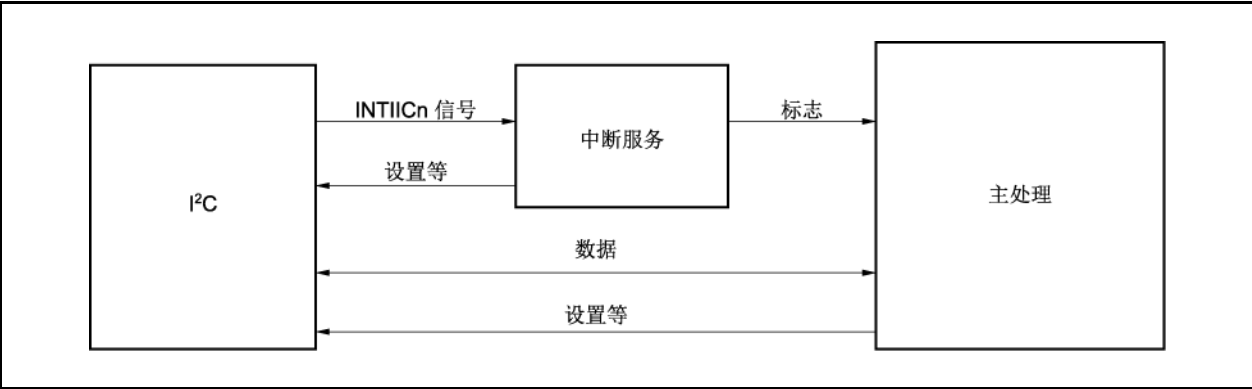
17.16.3 从设备操作

以下显示了从设备操作的处理过程。

从设备的操作基本上都是事件驱动。因此，很有必要由 INTIICn 中断（处理需要对操作状态作重大改变，例如在通信期间检测到停止状态）进行处理。

下面的描述假定数据通信不支持扩展码。同时假定 INTIICn 中断服务程序只完成状态改变处理，以及保证主程序处理期间实际的数据传送。

图 17-20. 从设备操作期间的软件架构



因此，准备下列三个标志，以便使数据传送处理可以通过传送标志给主程序完成，而不是传送 INTIICn 信号。

(1) 通信模式标志

这个标志指示下列通信状态。

清除模式： 不进行数据处理

通信模式： 进行数据通信处理（有效的地址检测，停止条件检测，不检测由主设备来的 $\overline{ACK}$ ，地址不匹配）

(2) 就绪标志

这个标志指示数据通信允许。正常数据传送期间本状态和 INTIICn 中断相同。这个标志在中断服务处理模块中设置，在主程序模块处理期间清除。第一个传送数据的就绪标志不在中断服务处理模块设置，因此第一个数据在没有清除处理的情况下进行发送（地址匹配被认为是请求下一个数据）。

(3) 通信方向标志

这个标志指示通信的方向，并且与 IICSn.TRCn 位的值相同。

下面显示的是从设备操作期间主程序处理模块的操作。

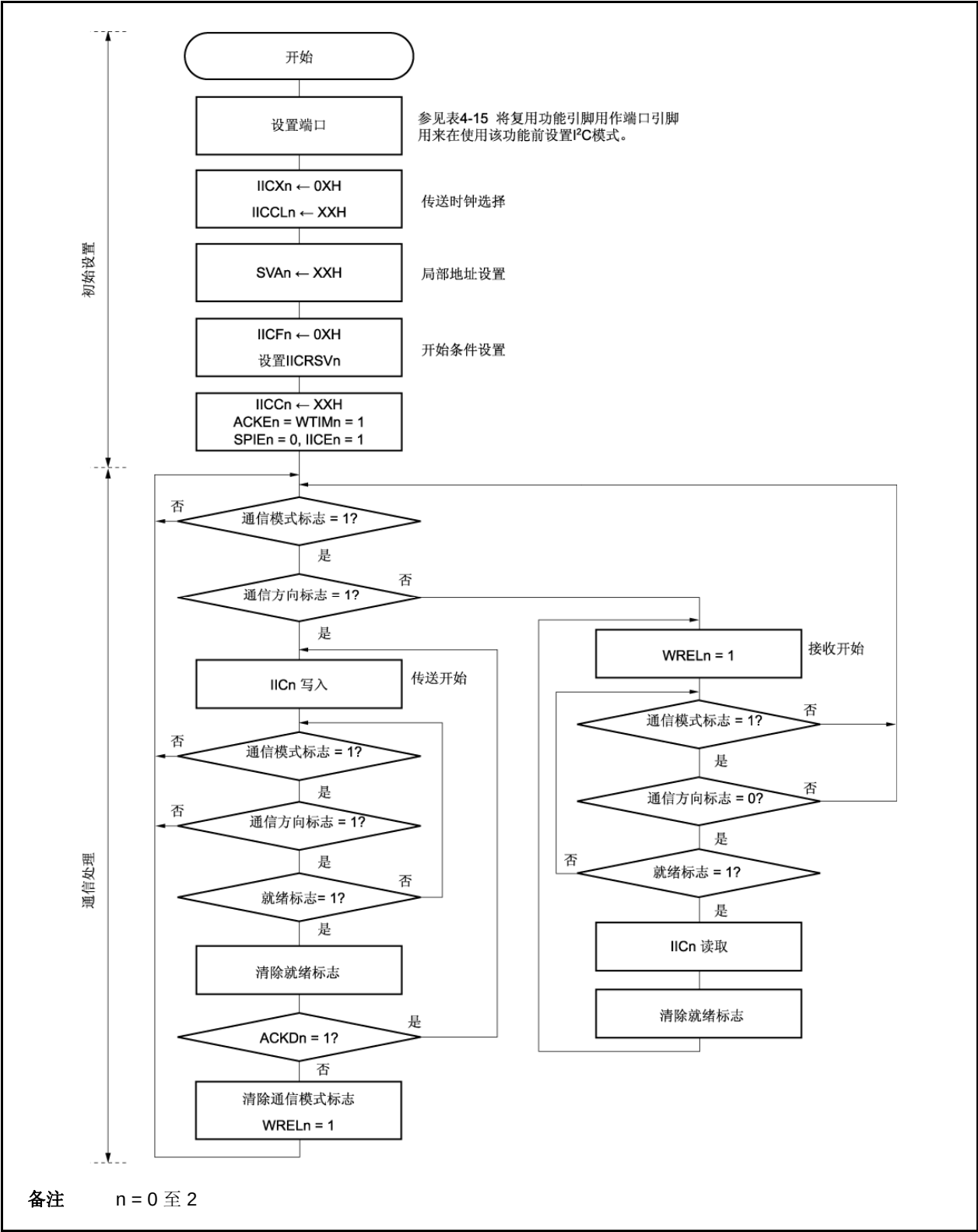
I²COn 开始，并且等待通信允许状态。当通信允许时，使用通信模式标志和就绪标志进行传送（停止条件和启动条件的处理由中断进行，条件由标志来进行确认）。

对于传送来说，重复传送操作，直到主设备停止返回 $\overline{\text{ACK}}$ 。当主设备停止返回 $\overline{\text{ACK}}$ 时，传送完成。

对于接收来说，传送完成后，接收到的数据符合要求的数量，但并不立即为下一个数据返回 $\overline{\text{ACK}}$ 。此后，主设备会产生停止条件或者重启条件。这将会导致退出通信。

<R>

图 17-21. 从设备操作流程图中 (1)



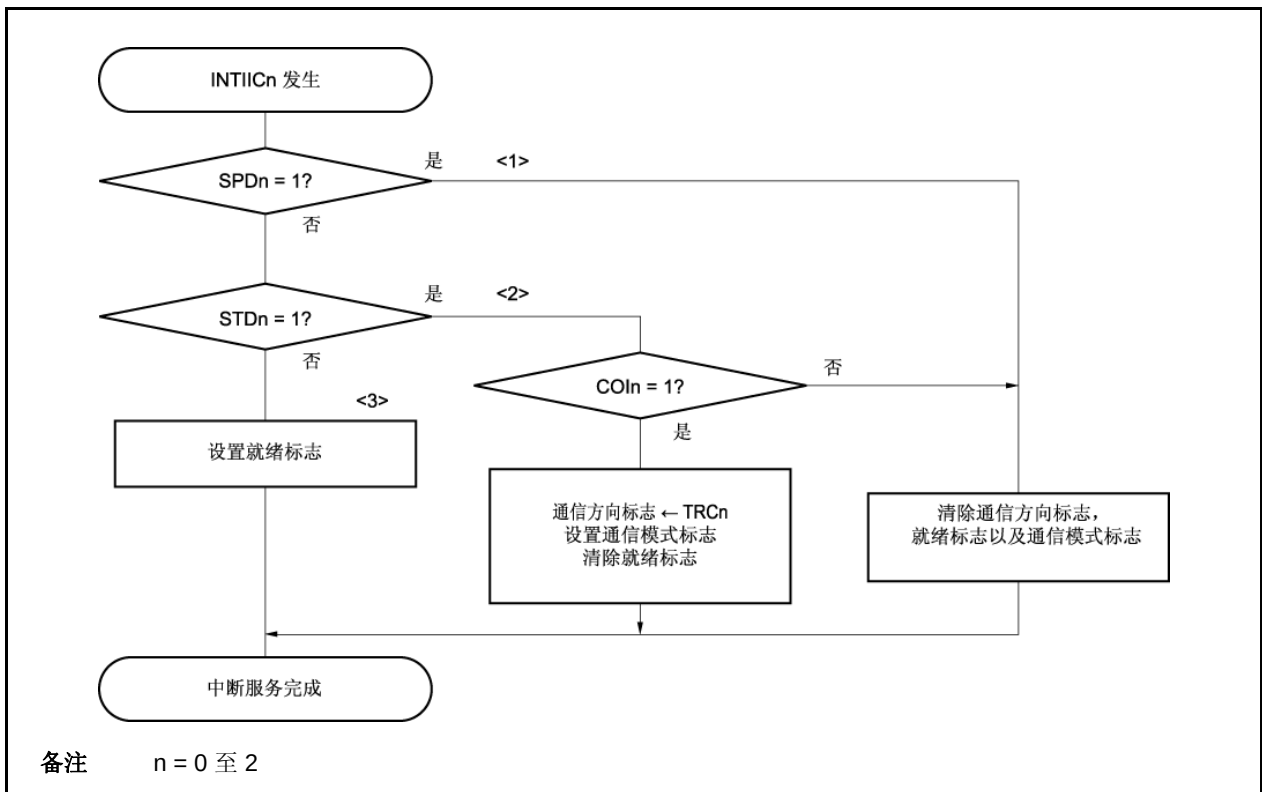
下面显示的从设备处理实例是由 INTIICn 中断完成的（假定这里不使用扩展码）。INTIICn 中断期间，状态被确认后，会执行如下步骤。

- <1> 检测到停止状态时，通信终止。
- <2> 检测到启动条件时，确认地址。如果地址不匹配，通信终止。如果地址匹配，则设定通信模式并释放等待，且操作从中断返回（就绪标志被清除）。
- <3> 对于数据传送 / 接收来说，设置就绪标志时，在 I2C0n 总线处于等待状态期间，操作从中断返回。

备注 上面的<1> 到 <3> 对应于图 17-22 从设备操作流程图中（2）中的<1> 到 <3>。

<R>

图 17-22. 从设备操作流程图中（2）



17.17 数据通信的时序

使用 I²C 总线模式时，主设备通过串行总线生成一个地址用来从设备中选择一个作为它的通信方。

输出从地址后，主设备发送 IICSn.TRCn 位，它规定了数据传送方向，随后开始与从设备进行串行通信。

IICn 寄存器的移位操作和串行时钟引脚（SCL0n）的下降沿同步。传送数据被送到 SO 锁存器，并且通过 SDA0n 引脚（MSB 优先）输出。

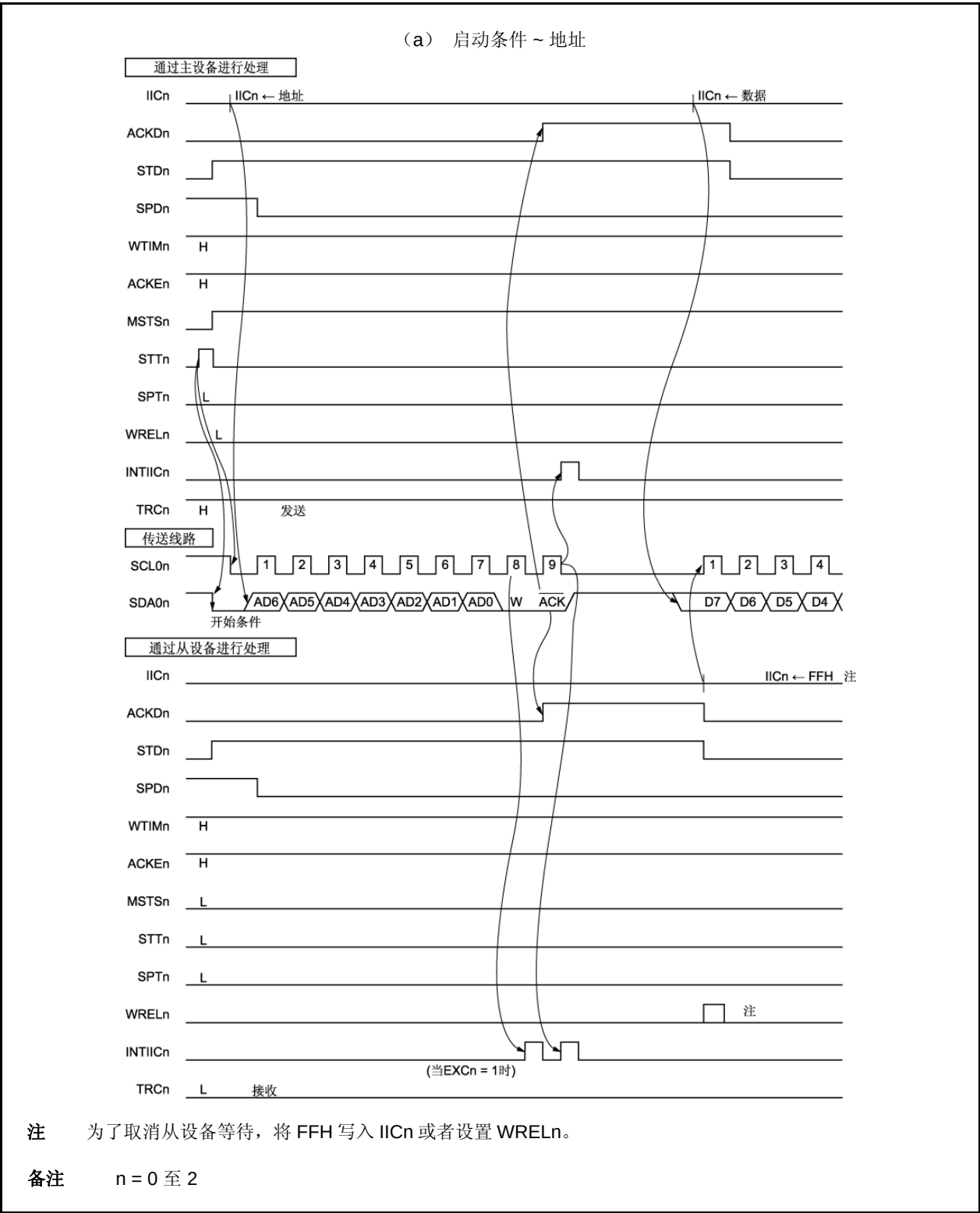
通过 SDA0n 引脚的数据输入在 SCL0n 引脚的上升沿由 IICn 寄存器捕捉。

数据通信的时序如下所示。

备注 n = 0 至 2

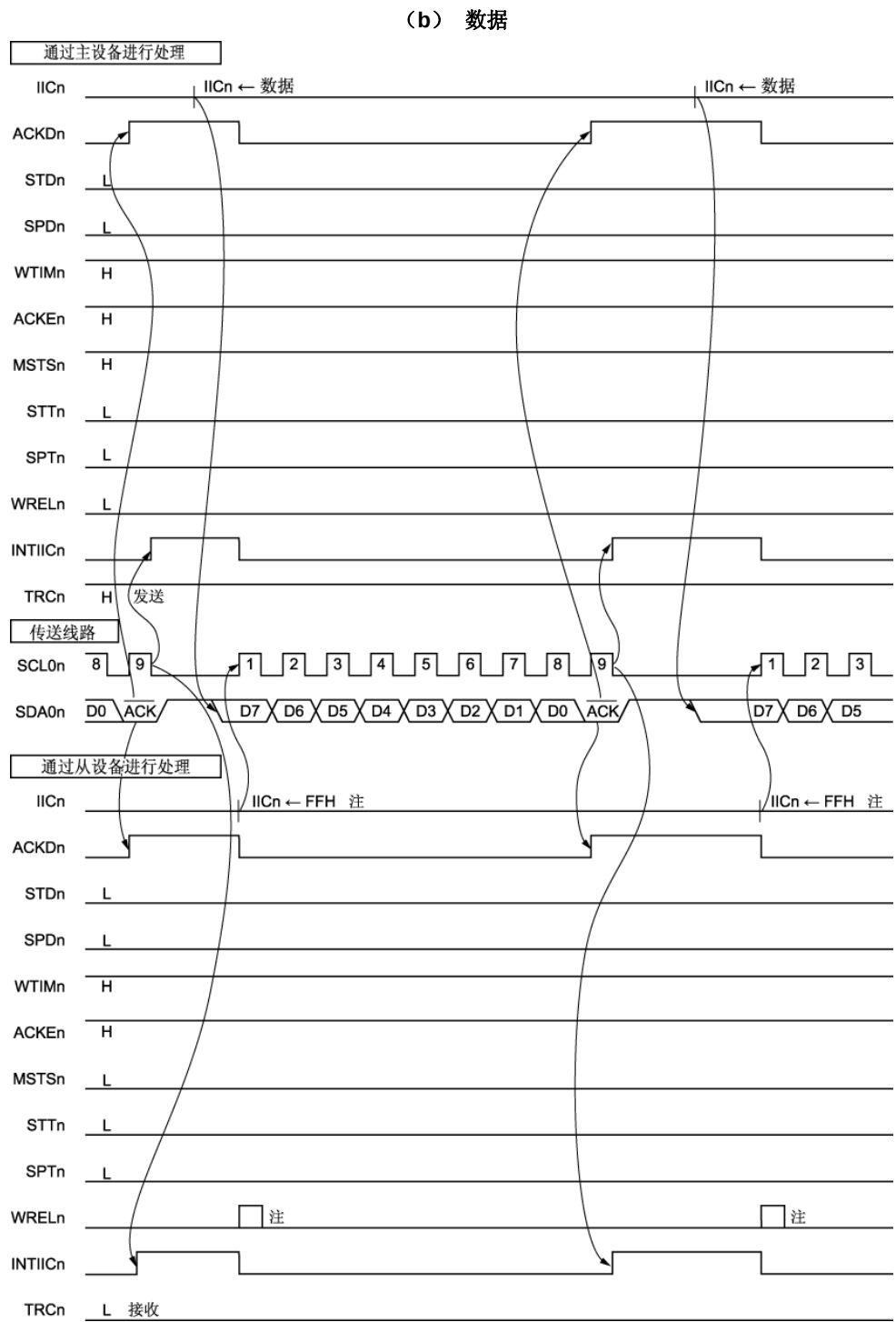
<R>

图 17-23. 从主设备到从设备通信的实例
(当主设备和从设备都选择 9 个时钟等待时) (1/3)



<R>

图 17-23. 从主设备到从设备通信的实例
(当主设备和从设备都选择 9 个时钟等待时) (2/3)

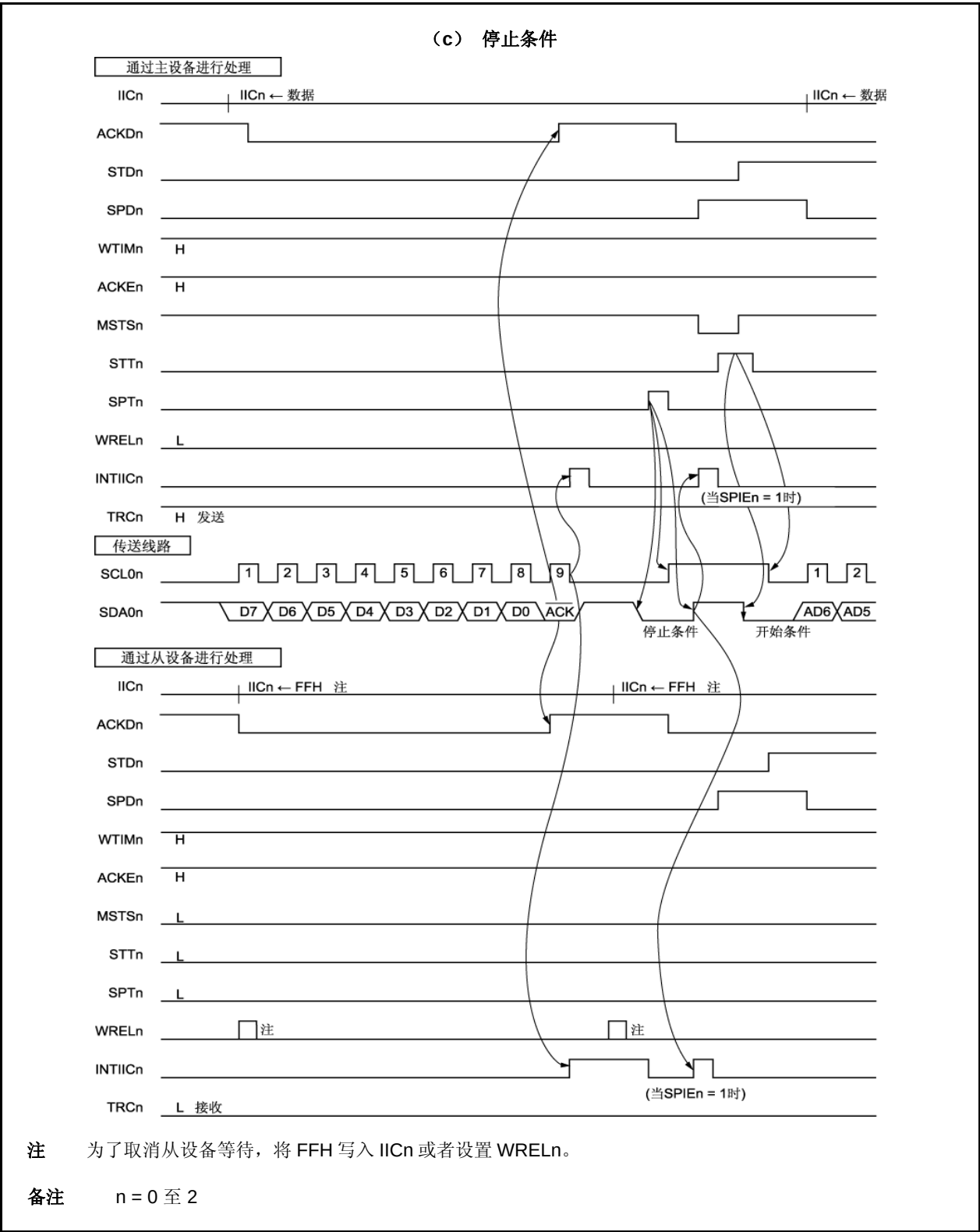


注 为了取消从设备等待，将 FFH 写入 IICn 或者设置 WRELn。

备注 n = 0 至 2

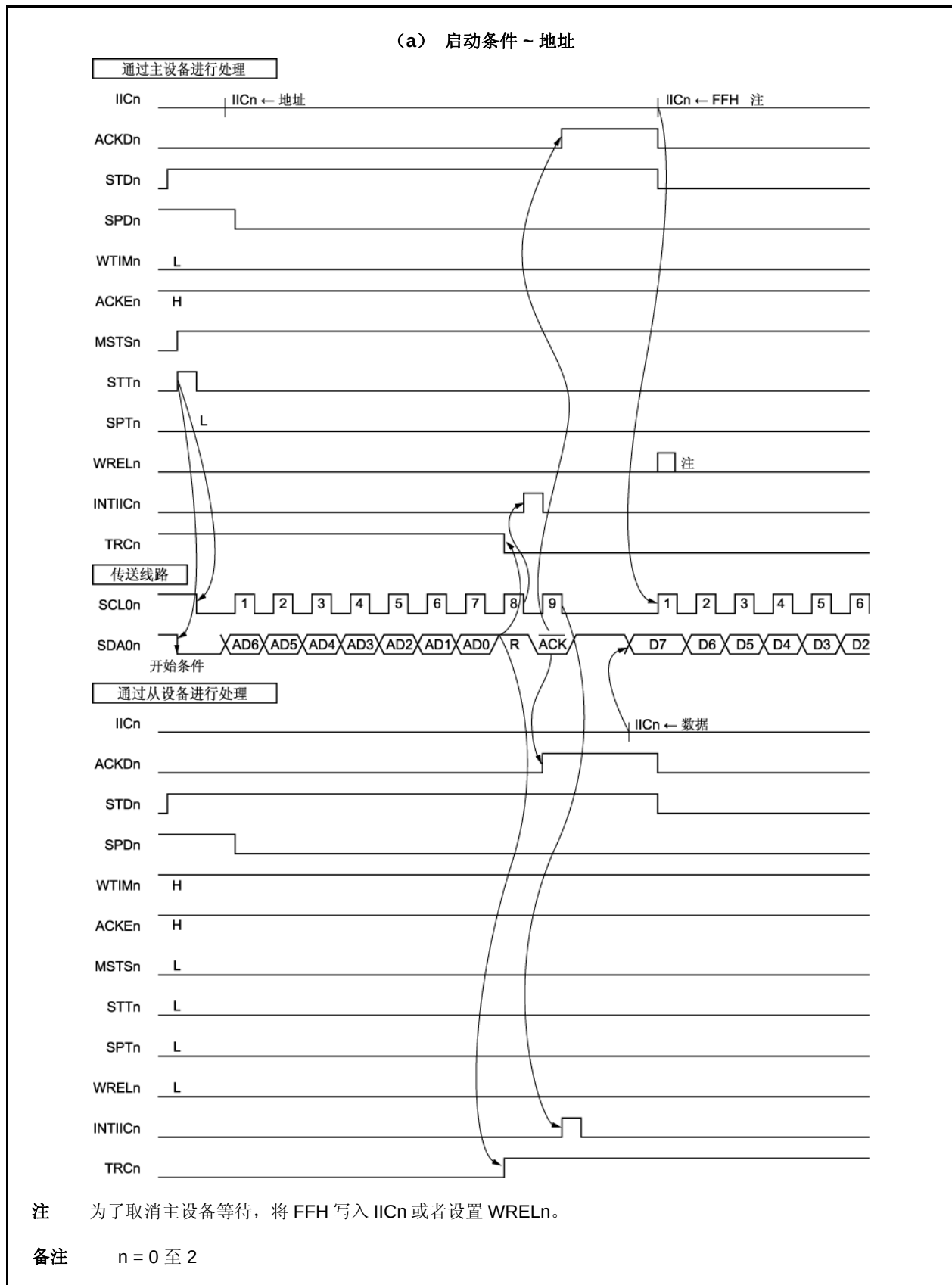
<R>

图 17-23. 从主设备到从设备通信的实例
(当主设备和从设备都选择 9 个时钟等待时) (3/3)



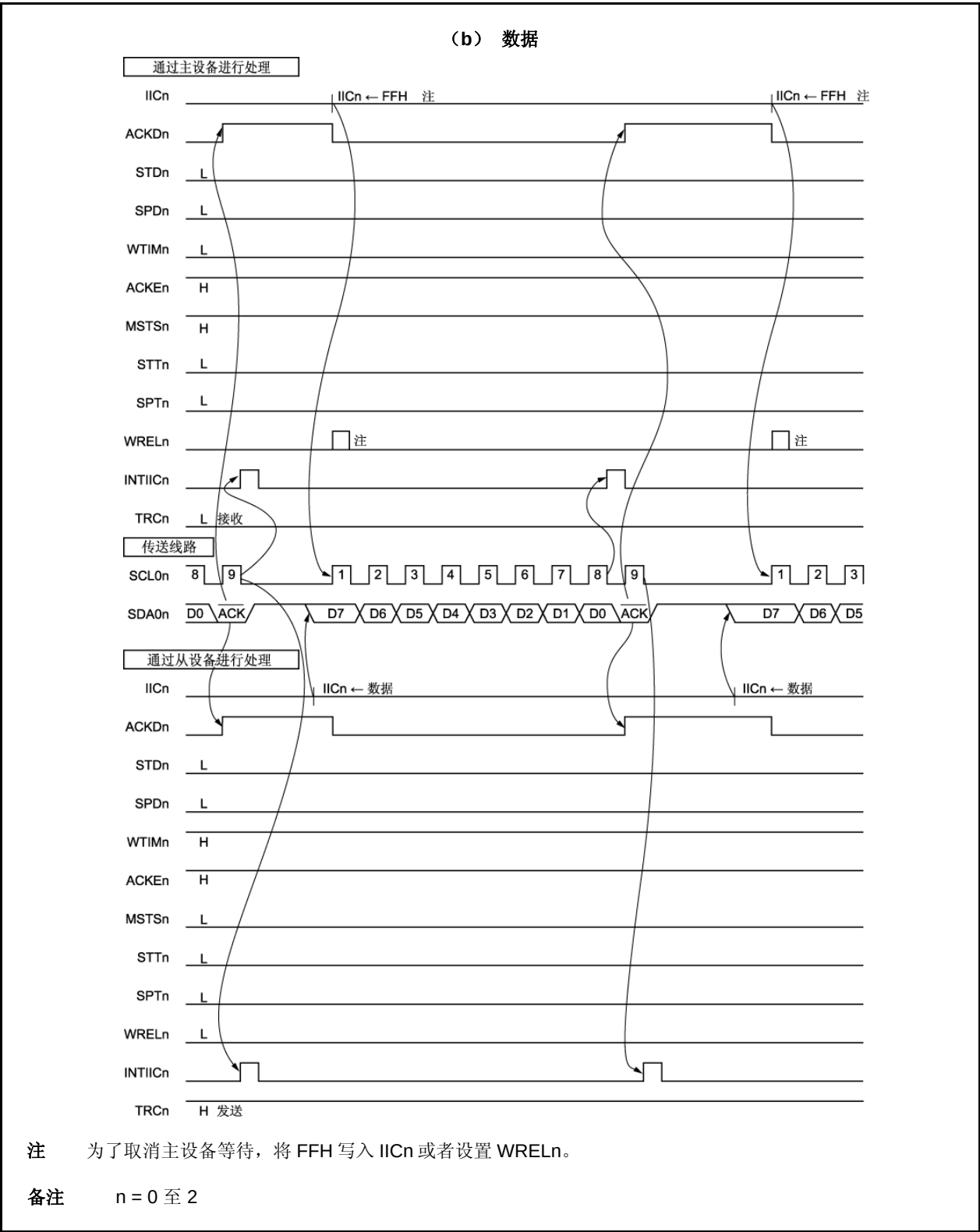
<R>

图 17-24. 从从设备到主设备通信的实例
 (当主设备和从设备都选择 9 个时钟等待时) (1/3)



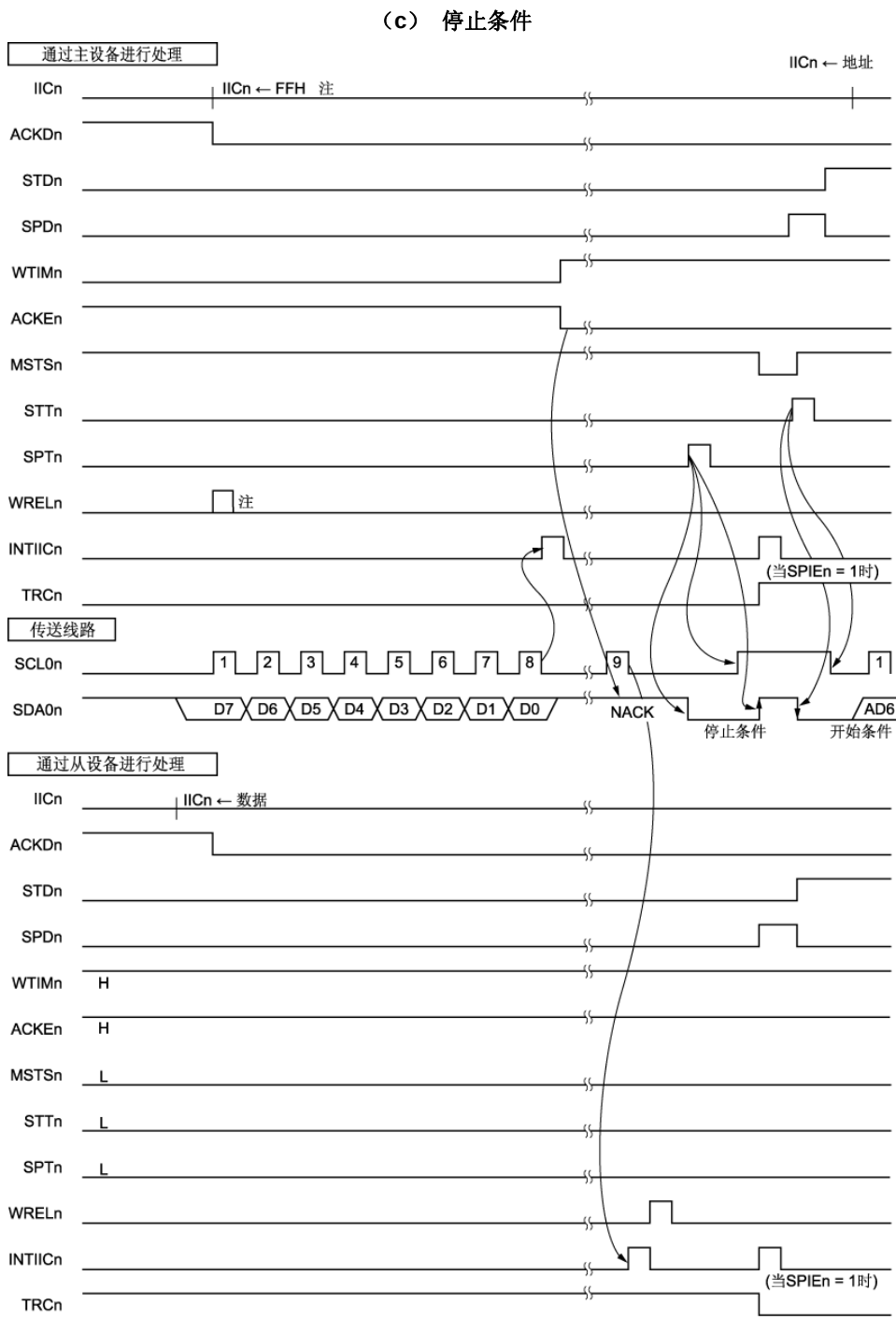
<R>

图 17-24. 从从设备到主设备通信的实例
(当主设备和从设备都选择 9 个时钟等待时) (2/3)



<R>

图 17-24. 从从设备到主设备通信的实例
(当主设备和从设备都选择 9 个时钟等待时) (3/3)



注 为了取消主设备等待，将 FFH 写入 IICn 或者设置 WRELn。

备注 n = 0 至 2

第 18 章 IEBus 控制器

IEBus（内部设备总线）是一种在单元模块之间传输数据的小规模数据传输系统。因为没有提供驱动器和接收器，为了应用 V850ES/SG2 和 V850ES/SG2-H 上的 IEBus 总线，需要接外部 IEBus 驱动器和接收器。

<R> V850ES/SG2 和 V850ES/SG2-H 的内部 IEBus 总线为负逻辑。

<R> V850ES/SG2 和 V850ES/SG2-H 的以下型号包含片上 IEBus 控制器。

<R>

- μ PD703270, 703270Y, 703271, 703271Y, 703272, 703272Y, 703273, 703273Y, 70F3271, 70F3271Y, 70F3273, 70F3273Y, 703272HY, 703273HY, 70F3273HY

18.1 功能

18.1.1 IEBus 总线通信协议

IEBus 总线通信协议如下所示。

(1) 多任务模式

连接到 IEBus 总线上的所有单元模块都能和其它单元传输数据。

(2) 广播通信功能

一个单元与多个单元模块之间的通信按照如下所示的方法进行。

- 组设备单元广播通信：对组设备单元的广播通信。
- 全部设备单元广播通信：对所有单元的广播通信。

(3) 有效传输速率

<R> 模式 1 和模式 2 下的有效传输速率如下（V850ES/SG2 和 V850ES/SG2-H 不支持模式 0 下的有效传输速率）。

- 模式 1：大约 17 kbps
- 模式 2：大约 26 kbps

注意事项 不同的模式不要在同一 IEBus 总线中使用。

(4) 通信模式

数据传输以半双工异步通信模式进行。

(5) 访问控制：CSMA/CD（带冲突监测的载波多路监控）

IEBus 总线的优先级如下所示：

- <1> 广播通信优先于个体通信（一对一的通信）。
- <2> 较低的主设备单元地址优先。

(6) 通信规模

IEBus 总线的通信规模如下。

- 设备单元数目：最多 50 个
- 线长度：最大 150 米（使用双绞线）

注意事项 实际系统中的通信范围随采用的 IEBus 驱动器 / 接收器以及 IEBus 线的特性等的不同而不同。

18.1.2 总线控制权的决定（仲裁）

当连接在 IEBus 总线上的单元控制其它单元时，要进行占用总线的操作。 这种操作叫做仲裁。
当两个或者更多个单元同时开始传输时，仲裁授予其中的一个单元许可权，使其占用总线。
由于仲裁的结果是只有一个单元拥有总线控制权，占用总线的优先级条件确定如下。

注意事项 如果通信异常中断，总线控制权释放。

（1）由通信类型决定的优先级

广播通信（一个单元对多个单元的通信）优先于正常通信（从一个单元到另一个单元的通信）。

（2）由主设备地址决定的优先级

如果通信类型相同，那么较低主设备地址的通信优先。
主设备地址由 12 位组成，000H 优先级最高，FFFH 优先级最低。

18.1.3 通信模式

IEBus 总线有三种通信模式，每一种都有不同的传输速率。 V850ES/SG2 和 V850ES/SG2-H 支持模式 1 和模式 2 通信。通信模式 1 和 2 中，每一个通信帧的传输速率和最大的传输字节数如表 18-1 所示。

表 18-1. 每一种通信模式下的传输速率和传输字节的最大数目

通信模式	最大传输字节（字节 / 帧）	有效传输速率（kbps） ^注
1	32	大约 17
2	128	大约 26

注 当传输字节数目为最大时的有效传输速率。

在开始通信前，为每一个连接到 IEBus 的单元选择通信模式。如果主单元和通信的对方（从单元）的通信模式不同，通信将产生错误。

18.1.4 通信地址

IEBus 总线中每一个单元分配一个 12 位的地址。通信地址包括下列标志码。

- 高 4 位：组号（指示每一个单元属于哪一组的号码）
- 低 8 位：单元号码（指示组内一个单元的号码）

18.1.5 广播通信

通常，传输或者接收在主单元和通信方在单元间一对一地进行。但是，广播通信中，存在两个或多个从单元，主单元执行到这些从单元的传输。因为有两个或者多个从单元存在，从单元返回 **NACK** 作为为应答信号。

通过广播位（查看此位，见 18.1.6（2）广播位），选择是执行广播通信还是选择正常通信。

广播通信分成两种类型：组设备单元广播通信和全部设备单元广播通信。这两种通信类型通过从设备单元地址的数值辨别（查看从设备单元地址，见 18.1.6（4）从设备单元地址域）。

（1）组设备单元广播通信

广播通信在由通信地址的高 4 位标示组号相同的组内单元中进行。

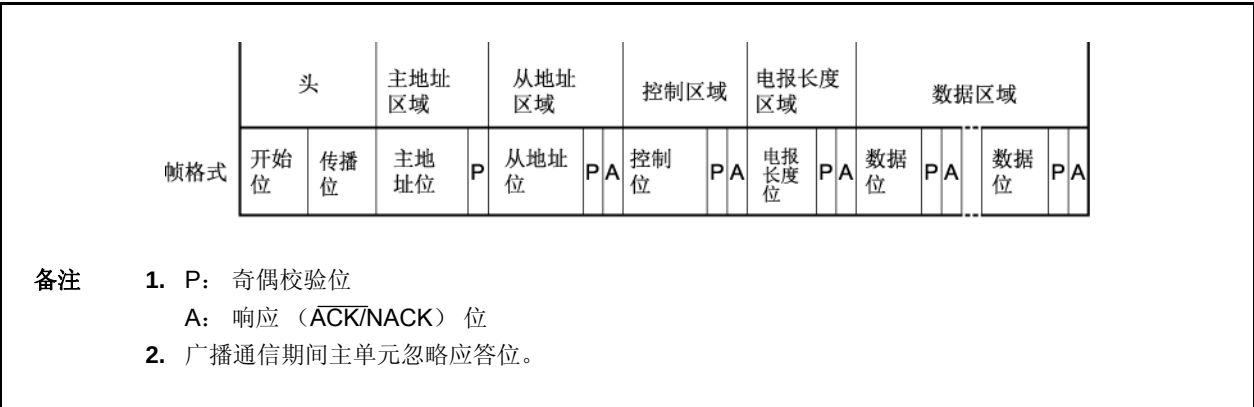
（2）全部设备单元广播通信

广播通信在所有单元中进行，而不管组号是多少。

18.1.6 IEBus 总线的传输格式

图 18-1 显示的是 IEBus 总线的传输信号格式。

图 18-1. IEBus 总线的传输信号格式



（1）开始位

开始位是告知其它单元数据传输开始的信号。将要开始数据传输的单元从 $\overline{\text{IETX}}$ 引脚输出一定长度的高电平信号，然后开始输出广播位。

当一个单元将要输出开始位时，另一个单元已经在输出开始位，那么这一单元不输出开始位，而是等待另一个单元开始位输出完成。当另一单元开始位输出完成时，这一单元和另一已完成开始位输出的单元同步开始输出广播位。

已经开始通信的另一方监测到开始位，就进入接收状态。

(2) 广播位

这一位指示主单元是选择一个从单元（个体通信）还是选择多个从单元（广播通信）作为通信的另一方。当此位为0，表示是广播通信，此位为1则表示是个体通信。广播通信分成两种类型： 组设备单元模式和全部设备单元模式。通信类型由从设备单元地址的数值标示（查看从设备单元地址，见18.1.6 （4） 从设备地址域）。

由于在广播通信模式下，存在两个或多个从单元作为通信对方， NACK 信号在紧跟着主设备单元地址域的每一个区域中返回 作为应答位。

如果两个或多个单元同时开始传输通信帧，广播通信优先于个体通信，并且会赢得仲裁。

如果一个单元作为主单元占用了总线，那么输出设置在广播请求标志（BCR.ALLRQ 位）内的数据。

(3) 主设备地址域

主设备地址域由主单元输出，告知从单元主设备单元地址。

主设备单元地址域的配置如图 18-2所示。

如果两个或更多个单元同时开始传输广播位，主设备单元地址域会做出仲裁判断。

主设备单元地址域每输出一位，都要和总线上的数据相比较。如果比较的结果是主设备单元地址域输出的数据不同于总线上的数据，就认为主单元在仲裁中失败。结果就是主单元停止传输，进入接收状态。

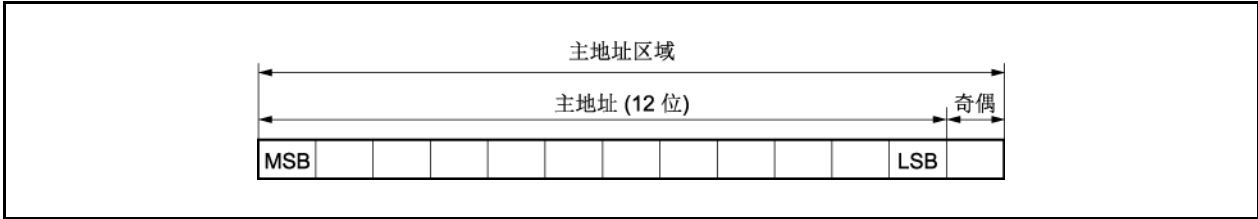
因为 IEBus 总线是线与配置，参与仲裁（仲裁主单元）的单元中，具有最小主设备单元地址的单元会赢得仲裁。

输出12位主设备单元地址后，只有一个主单元处于作为主设备的通信状态。

接下来，这一个主单元输出一个校验位，校验主地址的数据位，并且开始输出从设备单元地址域。

如果一个单元作为主设备占用总线， 输出由 UAR 寄存器设置的地址。

图 18-2. 主设备单元地址域



(4) 从设备地址域

主单元输出与它进行通信的单元的地址。

图 18-3 显示从设备单元地址域的配置。

为了避免错误的从设备单元地址误接收，传输 12 位从设备单元地址后输出校验位。接着，主单元监测从单元来的 $\overline{\text{ACK}}$ 信号，确认总线上存在从单元。当主单元监测到 $\overline{\text{ACK}}$ 信号时，它就开始输出控制域。但是，在广播通信中，主单元并不确认应答位，而是直接开始输出控制域。

如果从设备单元地址匹配，并且主设备单元地址和从设备单元地址的校验位都是偶，那么从单元输出 $\overline{\text{ACK}}$ 信号。如果为奇，那么从单元判定主设备单元地址或从设备单元地址没有正确接收，输出 NACK 信号。这时，主单元处于待机（监控）状态，通信结束。

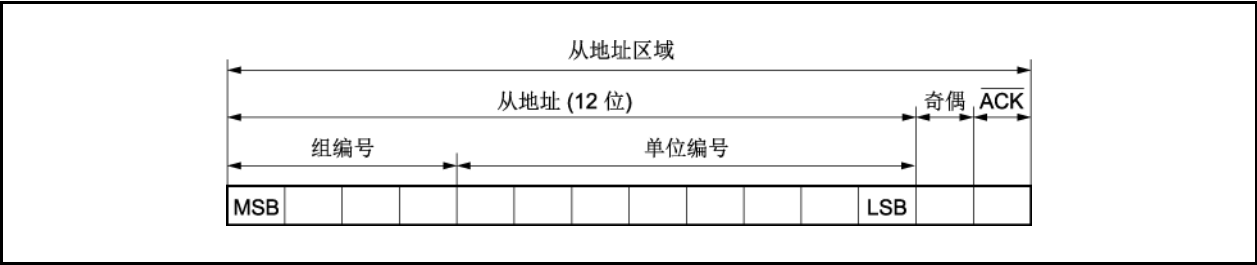
广播通信期间，从设备单元地址用来标示组设备单元广播或者全部设备单元广播，如下：

如果从设备单元地址为 FFFH ：全部设备单元广播通信
从设备单元地址不是 FFFH ：组设备单元广播通信

备注 组设备单元广播通信期间的组号为从设备单元地址的高 4 位。

如果一个单元作为主设备占用总线，输出由 SAR 寄存器设置的地址。

图 18-3. 从设备单元地址域



(5) 控制域

通过控制域，主单元输出它要求从单元进行的操作。

控制域的配置如图 18-4 所示。

如果紧跟控制位的校验位为偶，并且从单元能够执行主单元要求它执行的操作，那么从单元输出 $\overline{\text{ACK}}$ 信号，并且开始输出报文长度域。如果校验位为偶但从单元不能执行主单元要求它执行的功能，或者校验位为奇，从单元输出 NACK 信号，并且返回待机（监控）状态。

主单元监测到 $\overline{\text{ACK}}$ 信号后，开始输出电报区域。

如果主设备监测到 NACK 信号，那么主单元进入待机状态，通信结束。但是，在广播通信期间，主单元并不确认是否得到应答位，就开始输出报文长度域。

控制位的内容如下所示。

表 18-2. 控制位域的内容

位 3 ^{注1}	位 2	位 1	位 0	功能
0	0	0	0	读从设备状态
0	0	0	1	不确定
0	0	1	0	不确定
0	0	1	1	读数据，并锁定 ^{注2}
0	1	0	0	读锁地址（低 8 位） ^{注3}
0	1	0	1	读锁地址（高 4 位） ^{注3}
0	1	1	0	读从设备状态，并解锁 ^{注2}
0	1	1	1	读数据
1	0	0	0	不确定
1	0	0	1	不确定
1	0	1	0	写命令，并锁定 ^{注2}
1	0	1	1	写数据，并锁定 ^{注2}
1	1	0	0	不确定
1	1	0	1	不确定
1	1	1	0	写命令
1	1	1	1	读数据

- 注 1. 报文长度域的报文长度位和数据域的数据传输方向如下，根据位 3（MSB）的数值而变化。
如果位 3 是“1”：从主单元到从单元传输
如果位 3 是“0”：从从单元到主单元传输
2. 由一个控制位规定锁定或者解锁（参见 18.1.7（4） 锁定和解锁）。
3. 锁地址以一个字节（8 位）为单位传输，配置如下：



如果从主单元接收到的控制位不在表 18-3 所示范围内，那么由主单元锁定的单元拒绝响应控制位，并输出 NACK 信号。

表 18-3. 锁定的从单元的控制域

位 3	位 2	位 1	位 0	功能
0	0	0	0	读取从设备状态
0	1	0	0	读取锁状态（低 8 位）
0	0	0	1	读取锁地址（高 4 位）

另外，当响应表 18-4 中所示的控制数据时，没有被主单元锁定的单元拒绝响应，并输出 NACK 信号。

表 18-4. 不锁定的从单元控制域

位 3	位 2	位 1	位 0	功能
0	1	0	0	锁地址读取（低 8 位）
0	1	0	1	锁地址读取（高 4 位）

如果一个单元作为主设备占用总线，那么输出设置在 CDR 寄存器里的数值。

图 18-4. 控制域



表 18-5. 控制域的响应信号输出条件

(a) 如果接收到的控制数据是 AH, BH, EH, 或者 FH

通信类型 (USR.ALLTRANS 位) 个体通信 = 0 广播通信 = 1	通信目标 (USR.SLVRQ 位) 指定从设备 = 1 未指定 = 0	锁定状态 (USR.LOCK 位) 锁定 = 1 不锁定 = 0	主单元标示 (和 PAR 寄存器匹配) 锁请求单元 = 1 其它 = 0	从设备传输允许 (BCR.ENSLVTX 位)	从设备接收允许 (BCR.ENSLVRX 位)	接收的控制数据			
						AH	BH	EH	FH
0	1	0	忽略	忽略	1	○			
		1	1						
其他情况						×			

(b) 如果接收的控制数据是 0H, 3H, 4H, 5H, 6H, 或者 7H

通信类型 (USR.ALLTRANS 位) 个体通信 = 0 广播通信 = 1	通信目标 (USR.SLVRQ 位) 指定从设备= 1 未指定 = 0	锁定状态 (USR.LOCK 位) 锁定 = 1 不锁定 = 0	主单元标示 (和 PAR 寄存器匹 配) 锁请求单元 = 1 其它 = 0	从设备传输允许 (BCR.ENSLVTX 位)	从设备接收允许 (BCR.ENSLVRX 位)	接收的控制数据					
						0H	3H	4H	5H	6H	7H
0	1	0	忽略	0	忽略	○	×	×	×	○	×
				1		○	○	×	×	○	○
		1	0	忽略		○	×	○	○	×	×
			1	0		○	×	○	○	○	×
				1		○	○	○	○	○	○
				其他情况						×	

注意事项 如果接收到的数据非表 18-5 中所示的数据，一律认为是×（NACK 信号返回）。

备注 ○: 返回 ACK 信号。
×: 返回 NACK 信号。

(6) 报文长度域

此区域由传输方输出给接收方，告知传输数据的字节数。

报文长度域的配置如图 18-5 所示。

表 18-6 显示报文长度位与传输数据数量的关系。

图 18-5. 报文长度域

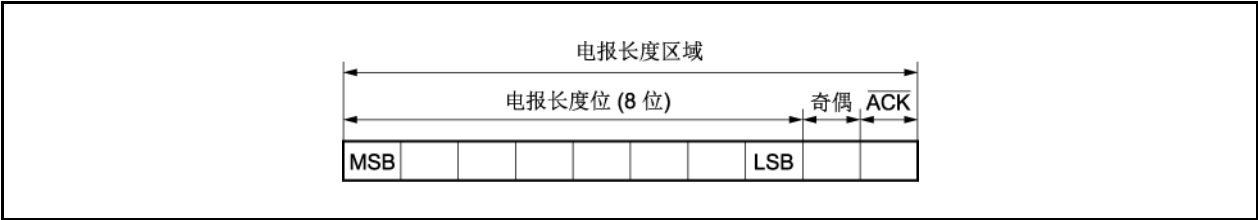


表 18-6. 报文长度位的内容

报文长度位（十六进制）	传输数据字节的数目
01H	1 个字节
02H	2 个字节
FFH	255 个字节
00H	256 个字节

依据主设备传输数据（控制位 3 为 1）或接收数据（控制位 3 为 0）的不同，报文长度域的操作不同。

(a) 当主设备单元传输数据时

主单元输出报文长度位和校验位以及位的同步信号。当从单元监测到校验位为偶，就输出 $\overline{\text{ACK}}$ 信号，并开始输出数据域。但是，广播通信期间，从单元输出 NACK 信号。

如果校验位为奇，从单元判定报文长度位没有正确接收，输出 NACK 信号，并返回到待机（监控）状态。这时，主单元也返回到待机状态，通信结束。

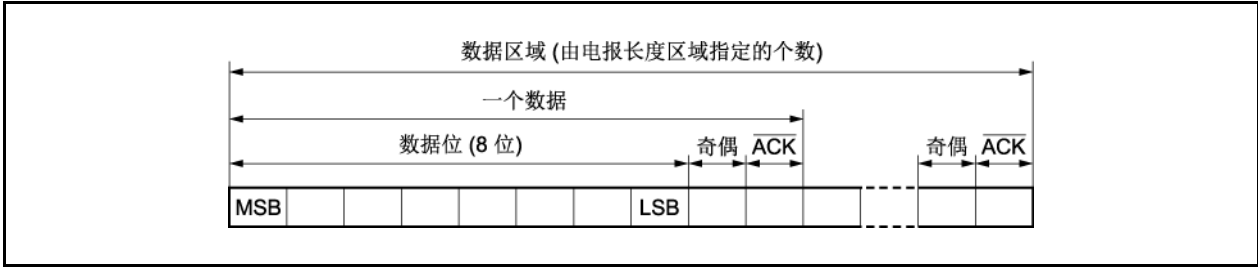
(b) 当主设备单元接收数据时

从单元输出报文长度位和校验位，主设备输出位的同步信号。当主单元监测到校验位为偶，输出 $\overline{\text{ACK}}$ 信号。如果校验位为奇，主单元判定报文长度位没有正确接收，输出 NACK 信号，并返回到待机（监控）状态。这时，从单元也返回到待机状态，通信结束。

(7) 数据域

这是由传输方输出的数据。
通过数据域，主单元传输数据给从单元或者从单元接收数据。
数据域的配置如下所示。

图 18-6. 数据域



紧跟着数据位的是，由相应的主单元或从单元输出的校验位和应答位。
只有当主单元传输数据时，使用广播通信。同时，忽略应答位。
依据主传输或接收数据，操作有所不同。

(a) 主传输数据时

当主单元写数据给从单元时，主单元传输数据位和校验位给从单元。当从单元接收到数据位和校验位时，如果校验位是偶且接收数据不存储在 DR 寄存器里，那么从单元输出 $\overline{\text{ACK}}$ 信号。如果校验位为奇，或者接收的数据存储在 DR 寄存器里，那么从单元拒绝接收数据，输出 NACK 信号。
如果从单元输出 NACK 信号，那么主单元又传输同样的数据。这种操作一直持续到主单元监测到从单元来的 ACK 信号或者数据超出最大传输字节数为止。
当校验位为偶，从设备输出 $\overline{\text{ACK}}$ 信号时，如果有更多的数据，且没有超出最大传输字节数，那么主设备单元接着进行下一个传输。
广播通信期间，从设备单元输出 NACK 信号，主设备单元一次传输 1 个字节的数据。广播通信期间，从设备单元接收到数据位和校验位后，如果校验位为奇或者 DR 寄存器正在接收数据，从设备单元判定接收没有正常进行，停止接收。

(b) 当主接收数据时

当主设备单元从从设备单元读数据时，主设备单元输出对应读取的位数的同步信号。

与从主设备单元来的同步信号相对应，从设备单元输出数据位和校验位。

主设备单元读入由从单元输出的数据位和校验位，并且查验校验位。

如果校验位是奇，或者 DR 寄存器正在接收数据，那么主设备单元拒绝接收数据，并输出 NACK 信号。如果最大传输字节数能在一个数据帧内传输完毕，那么主设备单元重复读相同的数据。

如果校验位是偶，并且 DR 寄存器未在接收数据，那么主设备单元接收数据，并输出 ACK 信号。如果最大传输字节数能在一个数据帧内传输完毕，那么主设备单元读下一个数据。

注意事项 广播通信中，不进行主接收操作，因为不能确定从设备单元，故数据传输不能正确进行。

(8) 奇偶校验位

奇偶校验位用来校验传输数据是否无误。

奇偶校验位附加在主设备单元地址位，从设备单元地址位，控制位，报文长度位和数据位的数据之后。

奇偶校验位为偶校验。如果数据中“1”的个数是奇数，那么奇偶校验位是“1”。如果为偶数，那么为“0”。

(9) 应答位

正常的通信期间（一对一通信），应答位附加在下列位置之后，用来检验数据是否正确接收。

- 从设备单元地址域结尾
- 控制域结尾
- 报文长度域结尾
- 数据域结尾

应答位定义如下。

- 0: 表示接受传输数据（ACK 信号）。
- 1: 表示不接受传输数据（NACK 信号）。

但是，广播通信期间，忽略应答位的内容。

(a) 从设备单元区域的最后一个应答位

从设备单元区域的应答位在下列任何一种情况下为 NACK 信号，传输停止。

- 主设备单元地址位或者从设备单元地址位的奇偶校验位不正确
- 产生时序错误（位格式错误）
- 从设备单元不存在

(b) 控制域的最后一个应答位

控制域的最后一个应答位在下列任何一个情况下为 NACK 信号，传输停止。

- 控制域的奇偶校验位不正确
- 从接收允许标志 (BCR.ENSLVRX 位) 没有置 (1)^注 时，如果控制位 3 是 “1” 的时候
- 从传输允许标志 (BCR.ENSLVTX 位) 没有置 (1) 时，如果控制位指示读数据时 (3H 或者 7H)
- 当设置锁定时，如果不是已经设置锁定请求的设备单元 (控制位为 3H, 6H, 7H, AH, BH, EH 或者 FH)
- 在没有设置锁定的情况下，如果控制位指示读锁地址 (4H, 5H) 时
- 时序错误
- 控制位不确定

注意事项 1. 即使 ENSLVTX 位 = 0 的情况下，当接收到从设备状态请求的控制数据时，一直返回 $\overline{\text{ACK}}$ 。

2. 即使 ENSLVRX 位 = 0，当接收到数据 / 命令写的控制数据时，控制域的应答位返回 NACK 信号。

只有在个体通信的情况下，通过 ENSLVRX 位禁止从设备接收操作 (通信停止)。广播通信的情况下，通信保留，数据中断请求信号 (INTIE1) 或者 IEBus 总线结束中断请求信号 (INTIE2) 产生。

(c) 报文长度域的最后一个应答位

报文长度域的最后一个应答位在下列任何一种情况下为 NACK 信号，传输停止。

- 报文长度位的奇偶校验位不正确时
- 时序错误

(d) 数据域的最后一个应答位

数据域的最后一个应答位在下列任何一种情况下为 NACK 信号，传输停止。

- 数据位的奇偶校验位不正确^注
- 在之前的应答位传输后产生时序错误
- 如果接收数据存储在 DR 寄存器里，不能接收到数据^注

注 这种情况下，当执行个体通信时，如果最大传输字节在一帧能够传输的数值之内，那么传输方再次执行数据域的传输操作。对广播通信来说，传输方不再进行传输操作，接收方产生通信错误，接收停止。

18.1.7 传输数据

(1) 从设备状态

主设备单元可以通过读入从设备状态得知从单元没有返回 $\overline{\text{ACK}}$ 信号的原因。

从设备状态根据从单元已经执行的上一次通信的结果决定。

所有从设备都能提供有关从设备状态的信息。

从设备状态的配置如下所示。

图 18-7. 从设备状态的位配置

MSB				LSB			
位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
位 0 ^{注 1}	含义						
0	传输数据不写入 DR 寄存器						
1	传输数据写入 DR 寄存器						
位 1 ^{注 2}	含义						
0	接收数据不存入 DR 寄存器						
1	接收数据存入 DR 寄存器						
位 2	含义						
0	不锁定设备单元						
1	锁定设备单元						
位 3	含义						
0	恒为 0						
位 4 ^{注 3}	含义						
0	从传输不停止						
1	从传输准备好						
位 5	含义						
0	恒为 0						
位 7	位 6	含义					
0	0	模式 0	表明由单元支持的最高模式 ^{注 4} .				
0	1	模式 1					
1	0	模式 2					
1	1	不使用					

- 注
1. 复位后：位 0 设置为 1。
 2. 接收缓冲器容量为 1 个字节。
 3. V850ES/SG2 作为从设备单元使用时，此位和 BCR. ENSLVTX 位指示的状态相对应。
 4. 位 7 和 6 恒为“10”，因为 V850ES/SG2 和 V850ES/SG2-H 能支持模式 1 和模式 2。

<R>

(2) 锁地址

当读锁地址时（控制位：4H 或者 5H ），发出锁指令的主设备单元的地址（12 位） 配置为 1 个字节单元，显示如下，并读入。

图 18-8. 锁地址的配置



(3) 数据

如果控制位指示读数据（3H 或者 7H ），那么由主设备单元读入从单元的数据缓冲器里的数据。
如果控制位指示写数据（BH 或者 FH ），那么由从单元接收到的数据根据从单元的操作规则进行处理。

(4) 锁定和不锁定

当信息以两个或多个通信帧传输时，使用锁功能。
被锁定的设备单元不接收从锁定它的设备单元之外的单元传输来的数据（不接收广播通信）。
设备单元的锁定和解锁如下。

(a) 锁定时

如果报文长度域接收 / 发送结束后 ($\overline{ACK} = 0$) ，没有成功接收 / 发送到由报文长度帧指定的数据长度通信帧，那么主设备单元将按照由控制位所指定的锁定方法（3H， AH 或者 BH ）将从单元锁定。这时，包括从设备状态的字节中的此位（位 2）设置为 1。

(b) 不锁定时

通过指定了锁定方法（3H， AH 或者 BH ）的控制位或者已经指定了解锁的方法（6H） 的控制位，在传输或者接收了由数据帧中的数据长度位指定的数据长度数据后，从设备由主设备单元解除锁定。这时，指示从设备状态的字节中的和锁定相关的位（位 2）复位为“0”。
广播通信期间不进行锁定和解锁操作。

锁定与解锁的条件如下所示。

表 18-7. 锁定设置的条件

控制数据	广播通信		个体通信	
	通信结尾	帧结尾	通信结尾	帧结尾
3H, 6H ^注			不能锁定	设置锁定
AH, BH	不能锁定	不能锁定	不能锁定	设置锁定
0H, 4H, 5H, EH, FH	不能锁定	不能锁定	不能锁定	不能锁定

注 当数据域的奇偶校验位为奇，并且当从 IEBus 总线单元来的 NACK 信号重复次数达到了正在输出的最大传输字节数，那么产生控制数据 6H （从设备状态读 / 解锁）的帧结尾。

表 18-8. 解锁释放条件（被锁定时）

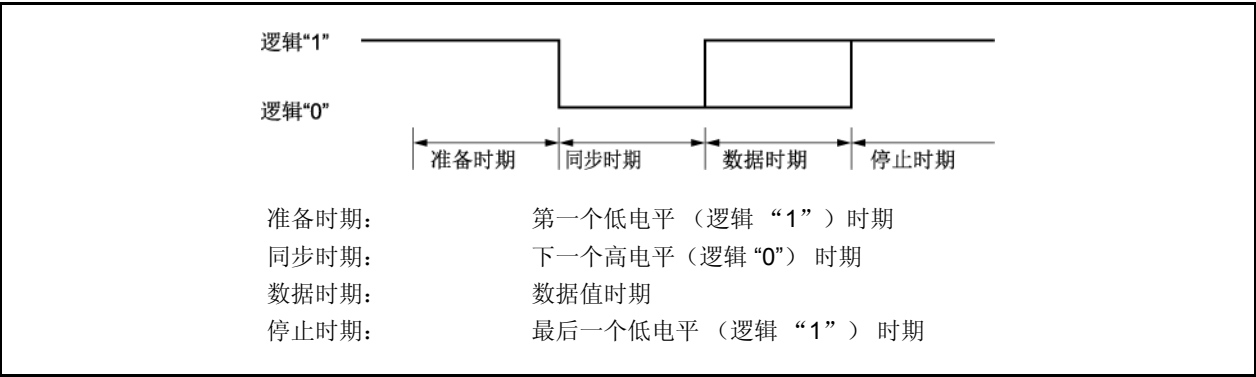
控制数据	从锁请求单元来的广播通信		从锁请求单元来的个体通信	
	通信结尾	帧结尾	通信结尾	帧结尾
3H, 6H ^注			解锁	保持锁定
AH, BH	解锁	解锁	解锁	保持锁定
0H, 4H, 5H, EH, FH	保持锁定	保持锁定	保持锁定	保持锁定

注 当数据域的奇偶校验位为奇，并且当从 IEBus 总线单元来的 NACK 信号重复次数达到了正在输出的最大传输字节数，那么产生控制数据 6H （从设备状态读 / 解锁）的帧结尾。

18.1.8 位格式

构成 IEBus 总线的数据帧的位格式如下所示。

图 18-9. IEBus 总线的位格式

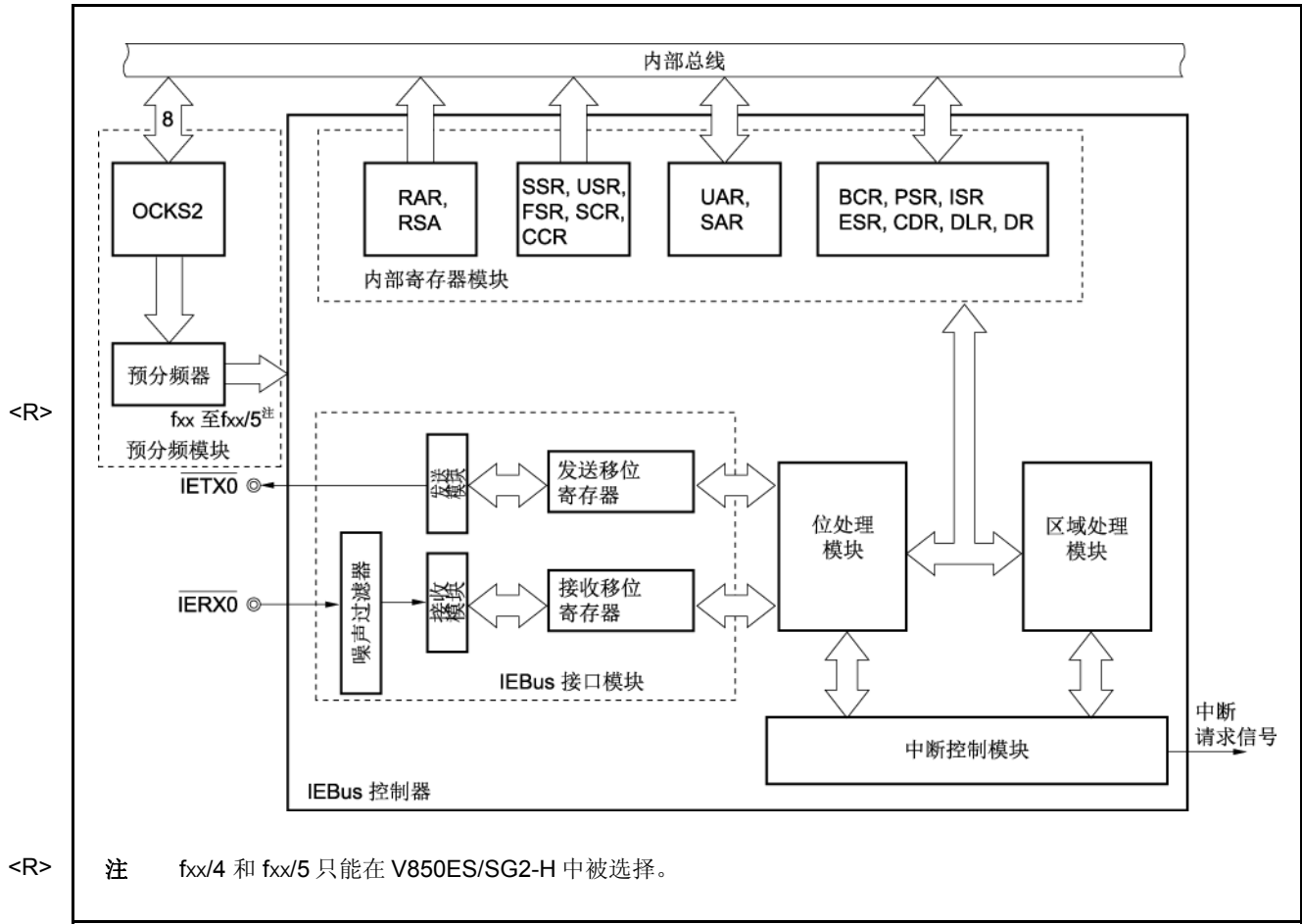


同步期间和数据期间的长度几乎相等。
IEBus 使每一位都同步。全部位的时间以及分配给这一位时间，依据传输位的类型和此单元是主单元还是从单元而不同。
通信期间，主设备单元和从单元监控各个时期（准备时期，同步时期，数据时期和停止时期）是否输出规定的时间，若否，那么主设备单元和从单元都报告时序错误，立即终止通信，然后进入待机状态。

18.2 配置

IEBus 控制器的框图如下所示。

图 18-10. IEBus 控制器框图



(1) 硬件配置和功能

IEBus 主要由以下六个内部模块组成。

- 中断控制模块
- 内部寄存器模块
- 位处理模块
- 域处理模块
- IEBus 总线接口模块
- 分频器模块

(a) 中断控制模块

中断控制模块将中断请求信号从 IEBus 控制器传输给 CPU。

(b) 内部寄存器模块

这些寄存器设置数据给控制寄存器和控制 IEBus 总线的区域（相关内部寄存器，见 **18.3 寄存器**）。

(c) 位处理模块

此模块产生和停止位时序，主要由位序列 ROM，8 位预置定时器 和比较器组成。

(d) 域处理模块

此模块产生通信帧中的每一个区域，主要由区域次序 ROM，4 位降序计数器，和比较器组成。

(e) IEBus 总线接口模块

这是外部驱动器 / 接收器的接口模块，主要由噪声滤波器，移位寄存器和传输 / 接收模块（冲突监测器，奇偶监测器，奇偶校验位发生器以及 $\overline{\text{ACK/NACK}}$ 发生器）组成。

(f) 预分频模块

此模块选择供应给 IEBus 总线控制器的时钟。

18.3 寄存器

控制 IEBus 总线控制器的寄存器如下所示。

表 18-9. IEBus 总线控制器的控制寄存器

地址	功能寄存器名称	符号	R/W	操作位单元			复位后
				1	8	16	
FFFFF348H	IEBus 时钟选择寄存器	OCKS2	R/W		√		00H
FFFFF360H	IEBus 控制寄存器	BCR		√	√		
FFFFF361H	IEBus 节能寄存器	PSR		√	√		
FFFFF362H	IEBus 从设备状态寄存器	SSR	R	√	√		81H
FFFFF363H	IEBus 单元状态寄存器	USR		√	√		00H
FFFFF364H	IEBus 中断状态寄存器	ISR	R/W	√	√		
FFFFF365H	IEBus 错误状态寄存器	ESR		√	√		
FFFFF366H	IEBus 单元地址寄存器	UAR				√	0000H
FFFFF368H	IEBus 从设备单元地址寄存器	SAR				√	
FFFFF36AH	IEBus 通信对方地址寄存器	PAR	R			√	
FFFFF36CH	IEBus 接收从设备单元地址寄存器	RSA				√	
FFFFF36EH	IEBus 控制数据寄存器	CDR	R/W		√		00H
FFFFF36FH	IEBus 报文长度寄存器	DLR			√		01H
FFFFF370H	IEBus 数据寄存器	DR			√		00H
FFFFF371H	IEBus 域状态寄存器	FSR	R		√		
FFFFF372H	IEBus 成功计数寄存器	SCR			√		01H
FFFFF373H	IEBus 通信计数寄存器	CCR			√		20H

(1) IEBus 控制寄存器（BCR）

BCR 寄存器是控制 IEBus 总线控制器操作的 8 位寄存器。

此寄存器能进行 8 位或 1 位读写。

复位输入将此寄存器清 00H。

复位后：00H R/W 地址：FFFFF360H

	<7>	<6>	<5>	<4>	<3>	2	1	0
BCR	ENIEBUS	MSTRQ	ALLRQ	ENSLVTX	ENSLVRX	0	0	0

ENIEBUS	IEBus 单元停止通信允许标志
0	IEBus 单元停止
1	IEBus 单元有效

MSTRQ	主设备单元请求标志
0	IEBus 单元不请求用作主设备
1	IEBus 单元请求用作主设备

ALLRQ	广播请求标志
0	个体通信请求
1	广播通信请求

ENSLVTX	从传输允许标志
0	从设备传输禁止
1	从设备传输允许

ENSLVRX	从接收允许标志
0	子设备接收禁止
1	子设备接收允许

- 注意事项 1. IEBus 作为主设备运行时，写 BCR 寄存器（包括位操作指令）禁止，直到通信或帧的结束，或者直到由于仲裁失败的通信错误产生而停止通信。主请求因此不能有多个。但是，通信强制停止（ENIEBUS 标志 = 0）的情况不在此列。
2. 如果对 BCR 寄存器的位操作与 MSTRQ 位的硬件复位相冲突，那么 BCR 寄存器不能正常操作。这种情况下，推荐采取下列措施。
- 因为硬件复位在从设备单元地址域的应答期间引起，所以一定要遵照下面的（b）主请求标志（MSTRQ）中的注意事项 1。
 - 有关 BCR 寄存器的写操作一定要遵照上面的注意事项。

(a) 通信允许标志 (ENIEBUS) ...位 7

<置位 / 清除条件>

置位: 由软件

清除: 由软件

IEBus 控制器参与通信根据设置 ENIEBUS 位 (1) 的时序而有所不同, 如下。

表 18-10. 设置 ENIEBUS 位的时序和参与通信

设置 ENIEBUS 位的时序	IEBus 控制器如何参与通信
如果 IEBus 总线上不进行通信	从下一帧参与通信或者开始通信
总线通信进行过程中, 如果其它的总线主设备正在传输开始位	如果监测到开始位, 那么从那一帧参与通信。如果监测不到, 从下一帧参与通信。
监测到其它总线的开始位后, 如果 IEBus 总线上通信正在进行	从下一帧参与通信

即使通信正在进行中, 如果 ENIEBUS 位清零 (0), 通信立即停止, 并且除几个例外的情况外, 内部标志和寄存器都复位。不复位的寄存器如下表所示。

当 ENIEBUS 位 = 0 时, 即使另一个单元开始通信, IEBus 控制器也不响应。

表 18-11. ENIEBUS 位清零而不复位的寄存器

由 ENIEBUS 位而不复位的寄存器	备注
UAR	不复位
SAR	不复位
CDR	从 CPU 里写入的数据不复位, 但是通信期间接收到的数据要复位。
DLR	从 CPU 里写入的数据不复位, 但是通信期间接收到的数据要复位。
DR	从 CPU 里写入的数据不复位, 但是通信期间接收到的数据要复位。

注意事项 开始设置 ENIEBUS 位 (1) 之前, 下列寄存器必须依据将要开始的通信模式进行设置。

表 18-12. 每一次通信前必须设置的寄存器

通信模式	必须预先设置的寄存器
主设备传输	UAR, SAR, CDR, DLR, DR (第一个 1 字节数据)
主设备接收	UAR, SAR, CDR
从设备传输	UAR, DLR, DR (第一个 1 字节数据) [※]
从设备接收	UAR

注 当开始从从设备传输时, 一些信息例如: DLR 寄存器设置的数值和要返回的数据 (设置给 DR 寄存器的数值) 等要预先分配。

(b) 主设备单元请求标志 (MSTRQ) ...位 6

<置位 / 清除条件>

置位：由软件设置

清除：当主通信开始，并且主设备单元的开始中断发生之前立即由硬件清零 (0)。

在通信错误发生之前由硬件清零 (0)。

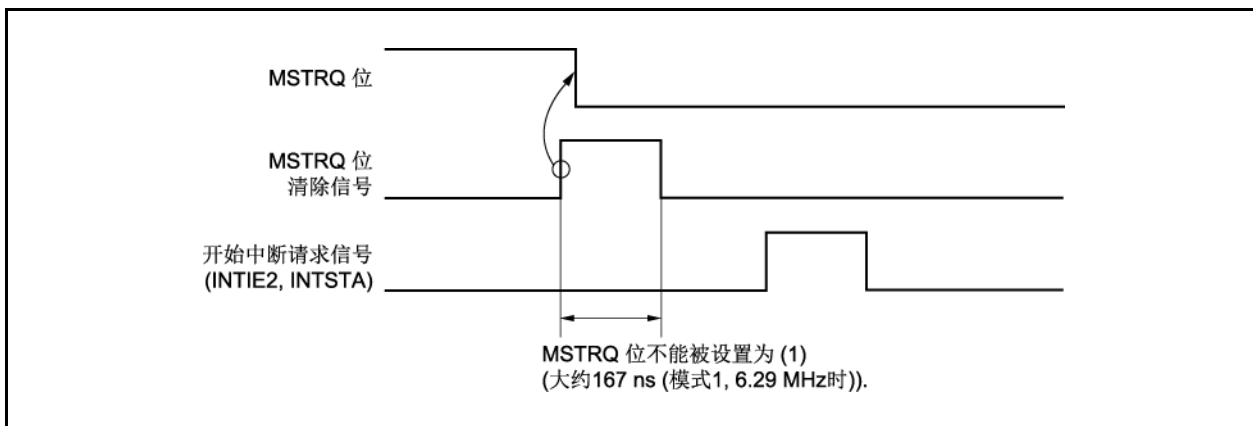
当 ENIEBUS 位清零时。

当 MSTRQ 位设置为 1 时，IEBus 总线控制器作为主设备单元在 IEBus 总线上开始通信。

但是，如果 IEBus 总线上的通信正在进行中（开始位正在传输期间，开始位监测不到，或者监测到开始位之后通信正在进行过程中），那么控制器等待直到当前帧传输结束（保持主请求挂起），帧传输结束后输出开始位，接着开始作为主设备单元开始通信。

注意事项 1. 如果 IEBus 总线控制器在仲裁中失败，那么由软件再次发起请求。
进行该操作时，在除了如下所示的时间段中设置 MSTRQ 位为 1。

图 18-11. MSTRQ 位不能设置的时间段



2. 当主设备单元请求已发出，并获得总线使用权时，不要设置 MSTRQ，ENSLVTX 和 ENSLVRX 位，直到通信结束（也就是，通信结束标志 (ISR.ENDTRNS 位) 或者帧结束标志 (ISR.ENDFRAM 位) 设置为 1）。因为设置这些寄存器就禁止了中断请求信号的产生。但是，如果通信异常中断，就可以设置。

(c) 广播请求标志 (ALLRQ) ...位 5

<置位 / 清除条件>

置位：由软件

清除：由软件

注意事项 当请求进行广播通信时，一般设置 ALLRQ 位为 1，然后设置 MSTRQ 位为 1。

(d) 从传输允许标志 (ENSLVTX) ...位 4

<置位 / 清除条件>

置位: 由软件

清除: 由软件

- 注意事项**
1. 接收到控制域的奇偶校验位之前，一定要设置 ENSLVTX 位。
 2. 当进行主设备单元请求时，设置 MSTRQ 位为 1 之前，清零 ENSLVTX 位。这是为了避免在主设备操作后控制器在仲裁中失败的情况下，和主设备单元请求从设备传输的情况下，DR 寄存器中的数据试图进行主传输。
 3. 当从禁止态返回到允许态时，传输从接下来的一帧开始有效。
 4. 当 ENSLVTX 位为 0 时，如果接收到数据/命令写的控制数据 (3H 或者 7H)，那么控制域中的应答位返回 NACK 信号。
 5. 即使 ENSLVTX 位 = 0 时，当返回从设备状态请求的控制数据时，将产生状态中断请求信号 (INTIE2, INTSTA)，通信继续。

(e) 从接收允许标志 (ENSLVRX) ...位 3

<置位 / 清除条件>

置位: 由软件

清除: 由软件

- 注意事项**
1. 接收到控制域中的奇偶校验位之前，一定要设置 ENSLVRX 位。
 2. 当 CPU 忙于其它处理时，通过清除 ENSLVRX 位 (0)，避免进行从接收。个体通信期间，控制域返回 NACK 信号，通信完成。广播通信期间，因为应答位忽略，通信不能完成。但是，IEBus 总线控制器并不响应广播通信，也不产生中断请求信号。
 3. 当从禁止态返回允许态时，传输从接下来的一帧开始有效。

(2) IEBus 节能寄存器 (PSR)

PSR 寄存器是一个控制 IEBus 总线控制器的内部时钟和通信模式的 8 位寄存器。
此寄存器可以进行 8 位或 1 位读写操作。
复位输入将此寄存器清为 00H。

复位后: 00H R/W 地址: FFFFF361H

	<7>	<6>	5	4	3	2	1	0
PSR	ENCLK	IEMODE	0	0	0	0	0	0

ENCLK	内部时钟操作允许标志
0	停止 IEBus 控制器的内部时钟
1	允许 IEBus 控制器的内部时钟

IEMODE	IEBus 通信模式设置标志
0	设置通信模式 1
1	设置通信模式 2

- 注意事项 1. 通信允许 (BCR.ENIEBUS 位 = 1) 期间, 不要设置 PSR 寄存器。
2. 一定要把位 5 清零 0。

(3) IEBus 从设备状态寄存器 (SSR)

SSR 寄存器是一个指示从单元通信状态的 8 位寄存器。当接收到来自主设备的从设备状态传输请求后，由软件读此寄存器，并将从设备状态写入 DR 寄存器来传输从设备状态。此时，报文长度自动设置为“01H”，因此不用设置 DLR 寄存器（它由硬件预设置）。

位 6 和 位 7 指示此单元支持的最高模式，恒为“10”（模式 2）。

此寄存器只读，8 位或者 1 位均可。

复位操作将此寄存器清为 81H。

复位后: 81H R 地址: FFFFF362H

	7	6	5	<4>	3	<2>	<1>	<0>
SSR	1	0	0	STATSLV	0	STATLOCK	STATRX	STATTX

STATSLV	从设备传输状态标志
0	从设备传输停止
1	从设备传输允许

STATLOCK	锁状态标志
0	解锁状态
1	锁定状态

STATRX	DR 寄存器接收状态
0	接收数据不存储在 DR 寄存器中
1	接收数据存储在 DR 寄存器中

STATTX	DR 寄存器传输状态
0	传输数据不存储在 DR 寄存器中
1	传输数据存储在 DR 寄存器中

- (a) 从设备传输状态标志 (STATSLV) ...位 4
反映从传输允许标志的内容 (BCR.ENSLVTX 位)。
- (b) 锁状态标志 (STATLOCK) ...位 2
反映锁定状态的标志 (USR.LOCK 位)。
- (c) DR 寄存器接收状态 (STATRX) ...位 1
此标志指示 DR 寄存器接收状态。
- (d) DR 寄存器传输状态 (STATTX) ...位 0
此标志指示 DR 寄存器传输状态。

(4) IEBus 单元状态寄存器 (USR)

USR 寄存器是一个指示 IEBus 总线单元状态的 8 位寄存器。

此寄存器只读，8 位或 1 位均可。

复位操作将此寄存器清零 00H。

复位后：00H R 地址：FFFF363H

	7	<6>	<5>	<4>	<3>	2	1	0
USR	0	SLVRQ	ARBIT	ALLTRNS	ACK	LOCK	0	0

SLVRQ	从设备请求标志
0	没有从主单元到从单元的请求
1	有从主单元到从单元的请求

ARBIT	仲裁结果标志
0	仲裁成功
1	仲裁失败

ALLTRNS	广播通信标志
0	个体通信状态
1	广播通信状态

ACK	应答传输标志
0	传输 NACK 信号
1	传输 $\overline{\text{ACK}}$ 信号

LOCK	锁状态标志
0	单元解锁
1	单元锁定

(a) 从设备请求标志 (SLVRQ) ...位 6

此标志指示是否有从主设备来的从设备请求。

<置位 / 清除条件>

置位：当请求此单元用作从设备单元时（如果 **表 18-13 从设备请求条件 (SLVRQ 位设置条件)** 满足），此标志当从设备单元地址域的应答时期开始时，由硬件设置为 1。

清除：当没有请求此单元用作从设备单元时，此标志通过硬件清零（如果**表 18-13 从请求条件 (SLVRQ 位设置条件)** 不满足）。复位时序和设置时序相同。如果此单元在通信正确接收（当 SLVRQ 位 = 1）之后立即被请求用作从设备单元，还有如果通信中从设备单元地址域中产生奇偶错误，那么此标志不清除。

表 18-13. 从设备请求条件 (SLVRQ 位设置条件)

单元状态	接收到的主设备单元地址	通信模式	接收到的从设备单元地址
未锁定	忽略	个体	UAR 寄存器匹配
		广播	组匹配
			FFFH
锁定	锁定的主设备单元匹配	个体	UAR 寄存器匹配
		广播	组匹配
			FFFH

注意事项 当单元锁定的时候，如果锁定的主设备单元之外的单元与其通信，SLVRQ 位不设置，但是返回 ACK 信号给从设备单元地址域。这是因为如果控制数据是从设备状态请求，那么即使锁定的主设备单元之外的单元返回此信号，通信也必须继续。

(b) 仲裁结果标志 (ARBIT) ...位 5

此标志指示仲裁结果。

<置位 / 清除条件>

置位：仲裁期间，当 IEBus 总线单元上输出的数据不能匹配总线线数据时，此标志设置为 1。

清除：此标志由开始位时序清除。

注意事项 1. 仲裁结果标志 (ARBIT 位) 清除时的时序，依据是否输出开始位而不同。

- 如果输出开始位：此标志在输出开始时序时清除。
- 如果不输出开始位：此标志在监测到开始位的时序时清除（大约在输出后 160 μ s（在 6.29 MHz 时的模式 1 时））

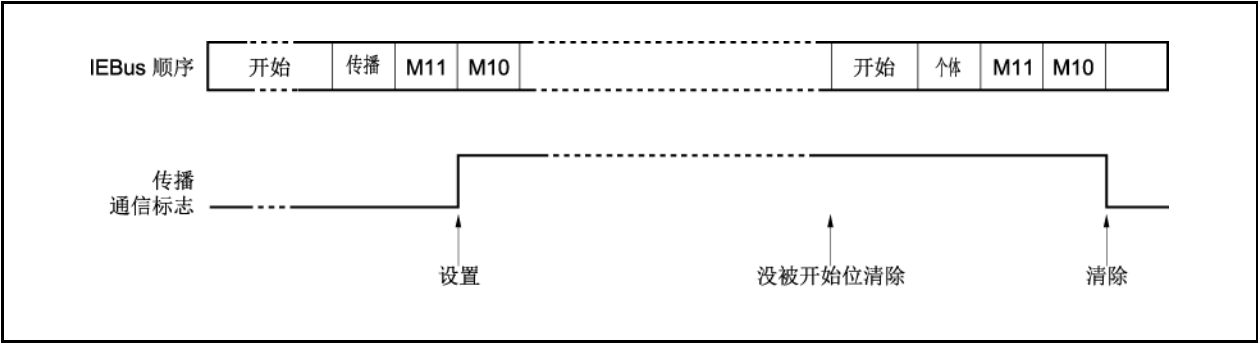
2. 如果其它的单元较早输出开始位，并且此单元在主设备单元请求后并不输出开始位，那么此标志在监测到开始位时清零。

(c) 广播通信标志 (ALLTRNS) ...位 4

此标志指示单元设备是否正在进行广播通信。此标志的内容在每一帧的广播区域更新。
除了由系统复位 (reset) 引起初始化外，它的置位 / 清除条件根据广播区域位的接收数据而变化。
<置位 / 清除条件>
设置：当广播区域接收到“广播”时
清除：当广播区域接收到“个体”，或系统复位输入时。

注意事项 不管 IEBus 总线是否是通信目标，广播通信标志都更新。

图 18-12. 广播通信标志操作的实例



(d) 应答传输标志 (ACK) ...位 3

IEBus 是接收单元时，此标志指示 $\overline{\text{ACK}}$ 信号在应答期间应答位区域是否已传输。此标志的内容在每一帧的应答位期间更新。但是，如果内部电路由于奇偶校验错误等原因而初始化，那么其内容不更新。

(e) 锁状态标志 (LOCK) ...位 2

此标志指示单元是否锁定。
<置位 / 清除条件>
置位：此标志在控制域中收到锁定命令 (3H, 6H, AH, BH) 后，当通信结束标志 (ISR.ENDTRNS 位) 为低，并且帧结束标志 (ISR.ENDFRAM 位) 为高时，设置为 1。
清除：当通信允许标志 (BCR.ENIEBUS 位) 清零时。
在控制域中接收到解锁 (3H, 6H, AH, BH) 后，通信结束标志 (ENDTRNS 位) 设置为 1 时。

注意事项 广播通信中不存在锁定 / 解锁。锁定状态中，不响应从请求锁定的单元之外发出的个体通信。但是，即使通信是从请求锁定的单元之外发出的，只要通信是从设备状态请求，那么也响应。

(5) IEBus 中断状态寄存器 (ISR)

当 IEBus 发出一个中断请求信号时，此寄存器指示中断源。读此寄存器产生一个中断请求信号，之后执行指定的中断处理程序。

此寄存器可以进行 8 位或 1 位读写。

复位操作将此寄存器清零 为 00H。

复位后: 00H	R/W ¹	地址: FFFFF364H						
	7	<6>	<5>	<4>	<3>	<2>	1	0
ISR	0	IEERR	STARTF	STATUSF	ENDTRNS	ENDFRAM	0	0
		IEERR	通信出错标志 (通信期间)					
		0	无通信错误					
		1	有通信错误					
		STARTF	开始中断标志					
		0	开始中断请求信号不产生					
		1	开始中断请求信号产生					
		STATUSF	状态传输标志 (从设备)					
		0	没有从设备状态 / 锁地址 (高 4 位, 低 8 位) 传输请求					
		1	有从设备状态 / 锁地址 (高 4 位, 低 8 位) 传输请求					
		ENDTRNS	通信结束标志					
		0	传输完报文长度域设置的数据字节后通信不结束					
		1	传输完报文长度域设置的数据字节后通信结束					
		ENDFRAM	帧结束标志					
		0	帧 (传输字节的最大数目 ²) 没有结束					
		1	帧 (传输字节的最大数目 ²) 结束					

- 注
- 1. 只有 IEERR 位可以写，并且只能是 0（也就是说，IEERR 位只能清除）。此位即使把 1 写入，它也不置 1。
 - 2. 模式 1: 32 个字节
模式 2: 128 个字节

(a) 通信出错标志 (IEERR) ...位 6

此标志指示的是产生了通信错误。当产生通信错误时，产生 INTIE2 和 INTERR 中断请求信号。

<置位 / 清除条件>

置位：如果时序出错，奇偶校验出错（数据域中的除外），NACK 接收出错（数据域中的除外），空栈读
出错，溢出出错（发生在广播通信接收过程中），或者写出错产生时，此标志设置为 1。

清除：由软件进行

(b) 开始中断标志 (STARTF) ...位 5

指示开始中断。当开始中断发生时，产生 INTIE2 和 INTSTA 中断请求信号。

<置位 / 清除条件>

置位：主设备单元请求之后，在从设备单元地址域中设置此标志为 1。

IEBus 总线作为从单元，从主设备单元发出请求（只假设在锁定的状态下，从请求锁定的单元中发出
的从设备请求的情况下）之后设置此标志为 1

清除：如果状态传输中断，通信结束中断，帧结束中断或者 INTIE1 中断请求信号产生时，此标志清零。

(c) 状态传输标志 (STATUSF) ...位 4

指示的是当控制器用作从设备单元时，主设备单元请求从设备状态和锁地址（高 4 位和低 8 位）传输。

<置位 / 清除条件>

置位：当 IEBus 总线用作从单元时，控制域中收到的自从主设备单元来的是 0H，4H，5H，或者 6H 的情
况下，此标志设置为 1。

清除：如果开始中断，通信结束中断，帧结束中断或者 INTIE1 中断请求信号产生时，此标志清零。

(d) 通信结束标志 (ENDTRANS) ...位 3

指示的是报文长度域中指定的字节数目传输完之后，通信是否结束的标志。当通信出错时，产生 INTIE2 和
INTSTA 中断请求信号。

<置位 / 清除条件>

置位：当 SCR 寄存器的计数值是 00H 时，设置为 1。

清除：当开始中断，状态传输中断，帧结束中断（如果不产生通信结束中断）或者 INTIE1 中断请求信号产
生时，此标志清零。

(e) 帧结束标志 (ENDFRAM) ...位 2

指示的是最大字节数（模式 1： 32 个字节， 模式 2： 128 个字节）传输完之后，通信是否结束的标志。

<置位 / 清除条件>

置位：CCR 寄存器的计数值是 00H 时，此标志设置为 1。

清除：如果开始中断，状态传输中断，通信结束中断（如果不产生帧结束中断），或者 INTIE1 中断请求信号产生时，此标志清零。

注意事项 1. 如果 CCR 和 SCR 寄存器都清零为 00H，ENDTRNS 和 ENDFRAM 位同时设置为 1。

2. 当重传数据导致达到最大传输字节时，如果最后一个数据域是 NACK 信号，那么 ENDFRAM 位和 IEERR（NACK 接收出错）位同时置 1。

(6) IEBus 错误状态寄存器 (ESR)

此寄存器指示的是 IEBus 总线的通信错误中断请求信号的中断源。只要通信出错标志 (ISR.IEERR 位) 设置为 (1)，那么此寄存器的相关位将设置 (1)。如果存在通信错误源，可以通过查看此寄存器的内容确定。

此寄存器可以进行 8 位或 1 位读写。

复位输入将此寄存器清零 00H。

复位后: 00H R/W 地址: FFFFF365H

	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
ESR	TERR	PERR	NERR	UERR	OERR	WERR	0	DEFLAG

TERR	时序出错标志
0	没有发生时序错误
1	发生时序错误

PERR	奇偶错误发生标志
0	没有发生奇偶错误
1	发生奇偶错误

NERR	NACK 接收出错标志
0	NACK 接收没有出错
1	NACK 接收出错

UERR	空栈读出错误标志
0	没有发生空栈读出错误
1	发生空栈读出错误

OERR	溢出出错标志
0	没有发生溢出错误
1	发生溢出错误

WERR	写出错标志
0	没有发生写错误
1	发生写错误

DEFLAG	第三方出错标志
0	单元通信期间出错
1	和单元之外的站点通信时出错

- 注意事项**
1. 每一位只能清除 (0)。即使写入 1，它也不能置 (1)。
 2. 出错时，ESR 寄存器的值更新。如果此时读寄存器，将得到不确定的值。建议在错误中断处理程序中读此寄存器。
 3. 如果通信出错，那么 IEBus 总线控制器返回默认状态，为通信做好准备。如果错误没有纠正就开始通信，那么出错标志将累积错误。在下一通信前纠正错误。

(a) 时序出错标志 (TERR) ...位 7

<置位 / 清除条件>

置位：如果时序出错，此标志设置为 (1)。

清除：由软件设置

如果通信位的高电平 / 低电平宽度不是定义值，那么时序就出错。高电平和低电平的宽度是通过内部定时器设置在位处理模块中，并且进行监控的。如果时序出错，那么 INTERR 和 INTIE2 中断请求信号产生。

(b) 奇偶校验出错标志 (PERR) ...位 6

<置位 / 清除条件>

置位：如果奇偶校验出错，此标志设置为 (1)。

清除：由软件清除。

当控制器用作接收单元时，如果每一个区域中的奇偶校验位与接收到的不匹配时，就产生错误。但是，个体通信期间，如果数据域中的校验位不匹配，那么返回 NACK 信号，并且要求重传。因此，不产生奇偶校验出错。

表 18-14. 如果校验不匹配时的操作

区域	通信模式	如果校验不匹配时的操作
主设备单元地址域	个体 / 广播	发生奇偶校验错误
从设备单元地址域	个体 / 广播	发生奇偶校验错误
控制数据域	个体 / 广播	发生奇偶校验错误
报文长度域	个体 / 广播	发生奇偶校验错误
数据域	个体	通过返回 NACK 信号，要求重传
	广播	发生奇偶校验错误

(c) NACK 接收出错标志 (NERR) ...位 5

<置位 / 清除条件>

置位：如果 NACK 接收出错，那么此标志设置为 (1)。

清除：由软件设置

不管控制器是作为主设备单元还是作为从设备单元，在个体通信的从设备单元地址域，控制数据域或者报文长度域的应答位期间，如果接收到 NACK 信号，那么 NACK 接收错误产生。如果在数据域的应答位期间接收到 NACK 信号，那么因为数据重传，不产生 NACK 接收错误。当达到最大传输字节时，在最后一个数据域的应答位期间接收到 NACK 信号，那么产生错误。

广播通信期间，因为不识别 ACK/NACK 信号，不产生 NACK 接收错误。

NACK 接收错误在第三方通信期间不发生，因为只监测时序 / 奇偶错误。

(d) 空栈读出错标志 (UERR) ... 位 4

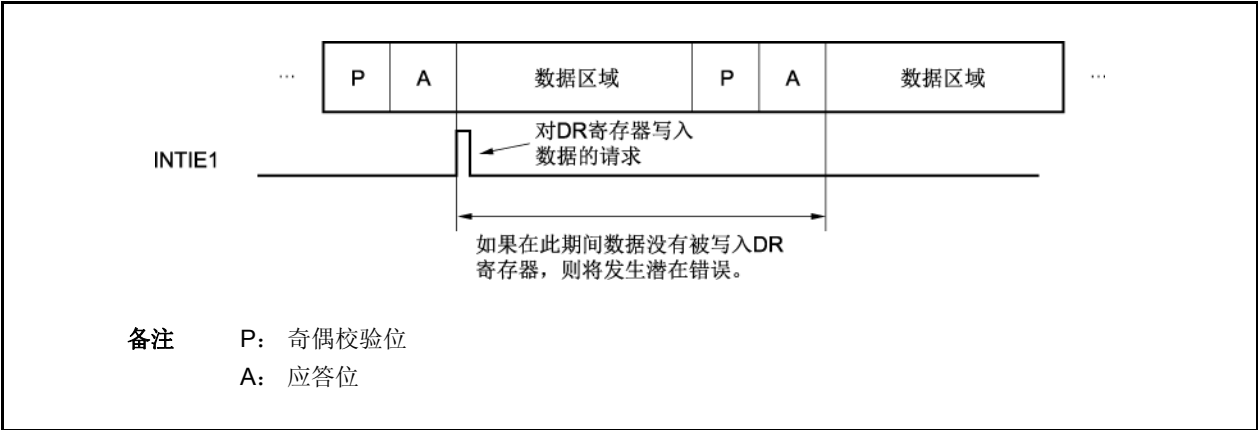
<置位 / 清除条件>

置位：空栈读出错时，此标志设置（1）。

清除：由软件设置。

如果在接收到 $\overline{\text{ACK}}$ 信号之前，下一个数据没有传入 DR 寄存器中，就会出错。个体通信期间和应答位期间，如果接收到 NACK 信号，因为数据重传，所以不产生错误。

图 18-13. 空栈读出错的时序



(e) 溢出出错标志 (OERR) ... 位 3

<置位 / 清除条件>

置位：如果溢出错误发生，此标志设置为 (1)。

清除：由软件设置。

当 IEBus 总线控制器用作接收单元时，如果 1 个字节数据存储在 DR 寄存器中，数据请求中断信号 (INTIE1) 就产生，并且通过 DMA 或者软件方式读取 DR 寄存器。如果这个读取操作延迟，接收到下一个数据，那么就产生溢出错误。

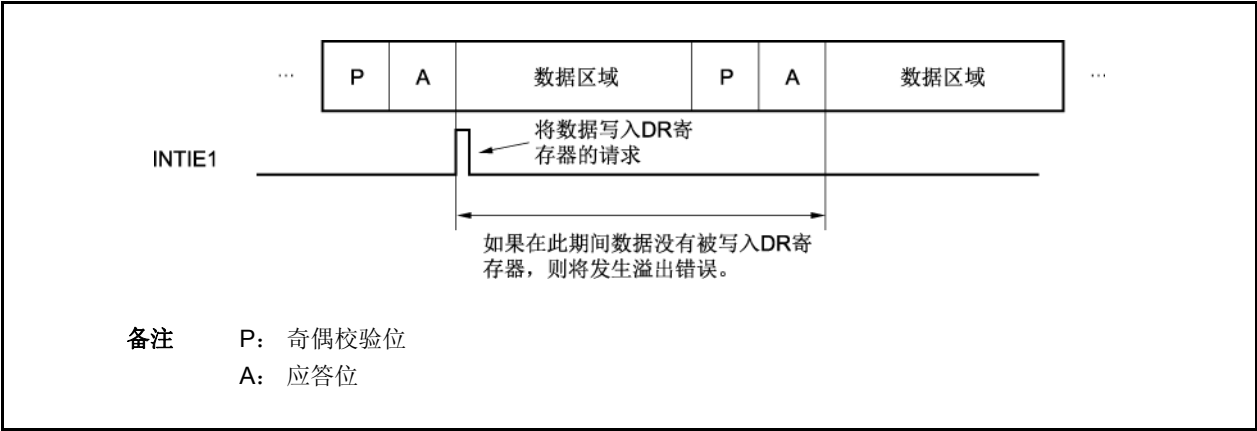
- 注意事项
- 1. 溢出错误发生时，如果没有读取 DR 寄存器，传输的字节数据数目已经达到传输字节的最大数目 (32 个字节)，那么帧结束中断请求信号 (INTSTA 或者 INTIE2) 产生。溢出状态一直维持到 DR 寄存器被读取，甚至在帧已经结束后仍然维持。
 - 2. 只有当读取 DR 寄存器和系统复位时，溢出状态清除。因此，在通信错误中断处理程序中，一定要读取 DR 寄存器。
 - 3. 如果有 2 个或更多个数据，那么在溢出状态中，下一个数据不能传输。因为数据请求中断请求信号 (INTIE1) 不发生，传输数据不能设置，将会发生空栈读出错。因此，在清除溢出状态之后，一定要执行传输操作。

备注

个体通信接收期间，在下一个数据的应答位期间，NACK 信号返回。作为回应，传输单元重传数据。因此，CCR 寄存器值减少，但 SCR 寄存器值不减少。

广播通信接收期间，产生通信错误中断请求信号 (INTIE2)，接收停止。这时，DR 寄存器不更新。INTIE1 信号不产生。SSR 寄存器的 STATRX 位保持为置位状态 (为 1)。当读取 DR 寄存器后，当接收到数据时，溢出状态清除。

图 18-14. 溢出出错时序



(f) 写出错标志 (WERR) ... 位 2

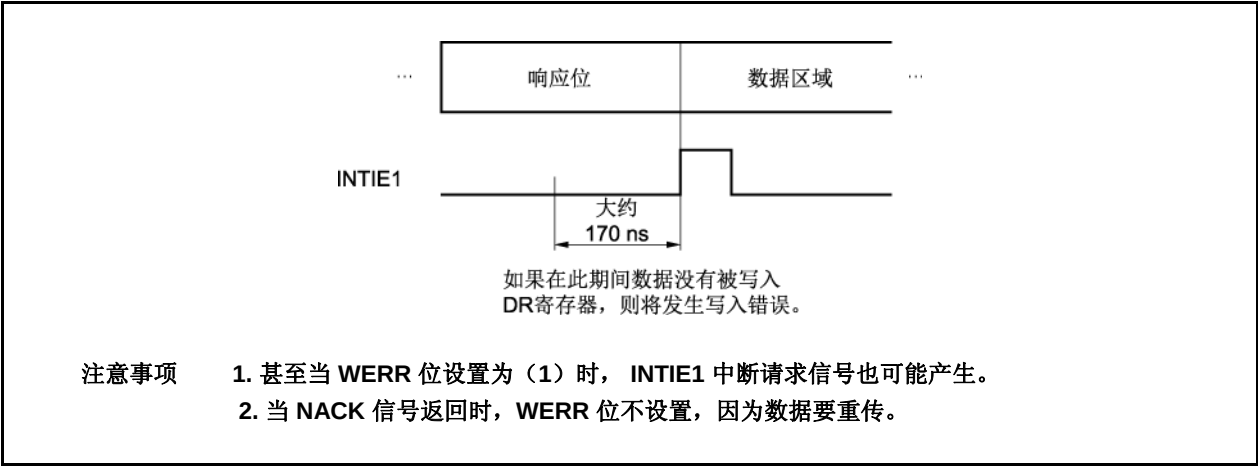
<置位 / 清除条件>

置位：当写错误发生时，此标志设置为 (1)。

清除：由软件

单元传输期间，写入 DR 寄存器中的数据没传输到数据域中，那么写错误发生。写错误发生的时序图如下。

图 18-15. 写错误发生的时序



(g) 第三方出错标志 (DEFLAG) ... 位 0

<置位 / 清除条件>

置位：通信期间，不管单元为何（第三方之间通信期间），如果时序出错或者奇偶校验出错，此标志设置 (1)。

清除：由软件

注意事项 第三方通信开始之前，甚至当从地址域不匹配（例如，当接收到的地址与从地址域中的单元中的不匹配（如果 NERR 位设置为 (1)）时，如果接收到 NACK 信号）时，如果发生错误，DEFLAG 不置为 (1)。

备注 第三方之间通信可能在下列两种情况下产生。

- <1> 如果从地址域中接收到的地址不匹配（个体通信期间：与 UAR 寄存器匹配，广播通信期间：与组或者 FFFH 匹配），如果接收到 ACK 信号后，通信继续就监控通信。
- <2> 广播通信期间，如果单元设备不能响应控制域中接收到的控制数据，如果通信继续，单元设备监控通信。例如，广播通信期间，单元从主设备单元中接收到控制数据 FH，但是此单元的不接收允许标志禁止 (BCR. ENSLVRX 位 = 0)（个体通信期间，NACK 信号返回，并且通信结束）时，这种情况就会发生。

(7) IEBus 总线设备单元地址寄存器 (UAR)

UAR 寄存器设置 IEBus 总线单元的单元地址。此寄存器必须在通信开始之前设置。

在位 11 到位 0 之间设置单元地址 (12 位)。

此寄存器可以进行 16 位读写。

复位操作将此寄存器清 0000H。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	地址	复位后	R/W
UAR	0	0	0	0													FFFFF366H	0000H	R/W

注意事项 通信允许时，不要设置 UAR 寄存器 (BCR.ENIEBUS 位 = 1)。

(8) IEBus 总线从设备地址寄存器 (SAR)

主设备请求期间，此寄存器的数值包含在从地址域中的传送数据数值中。此寄存器必须在通信开始之前设置。

在位 11 到位 0 之间设置从地址 (12 位)。

此寄存器可以进行 16 位读写。

复位操作将此寄存器清 0000H。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	地址	复位后	R/W
SAR	0	0	0	0													FFFFF368H	0000H	R/W

注意事项 通信允许时，不要设置 UAR 寄存器 (BCR.ENIEBUS 位= 1)。

(9) IEBus 总线通信对方地址寄存器 (PAR)

PAR 寄存器存储在主设备单元地址域中接收到的主设备单元地址数值，不管此设备是作为主设备还是作为从设备。

如果从主设备中接收到读锁地址 (低 8 位) 的请求“4H”，那么通过软件读此寄存器的数值，并且将此低 8 位数据写入 DR 寄存器。

如果从主设备中接收到读锁地址 (高 4 位) 的请求“5H”，那么通过软件读此寄存器的数值，并且将位 11 到位 8 的数据写入 DR 寄存器的高 4 位。

在位 11 到位 0 之间设置通信对方地址 (12 位)。

此寄存器只能 16 位读。

复位操作将此寄存器清 0000H。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	地址	复位后	R/W
PAR	0	0	0	0													FFFFF36AH	0000H	R

注意事项 当主设备单元地址域的奇偶校验期间结束时，如果奇偶校验正确，并且没有锁定，那么 PAR 寄存器存储地址数值。如果此时读取 PAR 寄存器，读取数值将不确定。

(10) IEBus 总线接收从设备地址寄存器 (RSA)

RSA 寄存器存储从地址域中接收到的从设备地址数值，不管是作为主设备还是作为从设备。

此寄存器只能 16 位读。

复位输入将此寄存器清 0000H。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	地址	复位后	R/W
RSA	0	0	0	0													FFFFF36CH	0000H	R

注意事项 当从地址域的奇偶校验期间结束时，如果奇偶校验正确，并且没有锁定，那么 RSA 寄存器存储地址数值。如果此时读取 RSA 寄存器，读取数值将不确定。

(11) IEBus 总线控制数据寄存器 (CDR)

CDR 寄存器只能进行 8 位读写。

复位输入将此寄存器清 00H。

备注 CDR 寄存器由写寄存器和读寄存器组成，并且写入 CDR 寄存器的数值不能像这样读。从寄存器读到的数据是由 IEBus 总线通信接收的数据。

(a) 当为主设备单元时

低 4 位数据包含在控制域中传输的数据中。主设备请求期间，开始通信之前，CDR 寄存器必须预先设置。

(b) 当为从设备单元时

控制域中接收到的数据写入到低 4 位。

当状态传输标志 (ISR.STATUSF 位) 设置 (1) 时，发出中断请求信号 (INTIE2)，并且根据 CDR 寄存器低 4 位的数值，由软件完成每一个处理操作。

复位后：00H R/W 地址： FFFF36EH

	7	6	5	4	3	2	1	0
CDR	0	0	0	0	MOD	SELCL2	SELCL1	SELCL0

MOD	SELCL2	SELCL1	SELCL0	功能
0	0	0	0	读从状态
0	0	0	1	不确定
0	0	1	0	不确定
0	0	1	1	读数据并锁定
0	1	0	0	读锁地址（低 8 位）
0	1	0	1	读锁地址（低 4 位）
0	1	1	0	读从状态并解锁
0	1	1	1	读数据
1	0	0	0	不确定
1	0	0	1	不确定
1	0	1	0	写命令并锁定
1	0	1	1	写数据并锁定
1	1	0	0	不确定
1	1	0	1	不确定
1	1	1	0	写命令
1	1	1	1	写数据

- 注意事项
1. 因为从设备单元必须判别接收到的数据是“命令”还是“数据”，通信完成后，要读取 CDR 寄存器的数值。
 2. 如果主设备单元设置为不确定数值，从单元返回 NACK 信号，并且通信异常中断。广播通信期间，主设备单元忽略应答位，通信继续。因此，不要设置为不确定数值。

(c) 从状态返回操作

当 IEBus 总线接收到从主设备状态到从设备状态转移的请求或者锁地址请求（控制数据：0H，6H），控制域中是否返回 ACK/NACK 信号依据于 IEBus 总线单元的状态。

- | | |
|--|--------------------------------|
| (1) 解锁状态中接收到 0H 或者 6H 控制数据 | → $\overline{\text{ACK}}$ 信号返回 |
| (2) 解锁状态中接收到 4H 或者 5H 控制数据 | → NACK 信号返回 |
| (3) 锁定状态中从发送锁定请求的单元中接收到
0H，4H，5H 或者 6H 控制数据 | → $\overline{\text{ACK}}$ 信号返回 |
| (4) 锁定状态中从发送锁定请求的单元之外接收到
0H，4H，或者 5H 控制数据 | → $\overline{\text{ACK}}$ 信号返回 |
| (5) 锁定状态中从发送锁定请求的单元之外接收到
6H 控制数据 | → NACK 信号返回 |

在以上锁定情况中，从设备状态或锁定请求的响应会引起 STATUSF 位置位（1），并产生状态中断（INTIE2， INTSTA）。产生时序在控制域奇偶校验位结束（应答位的开始）时开始。然而，如果返回 NACK 信号，那么在应答位之后产生 NACK 接收错误，通信终止。

图 18-16. 中断请求信号产生时序（对于（1），（3），和（4）来说）

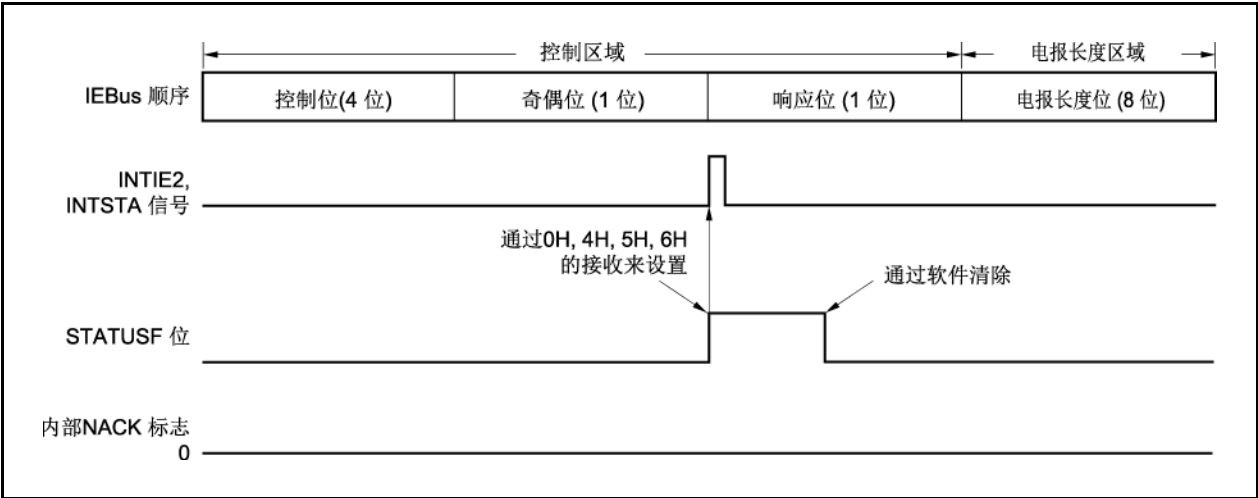
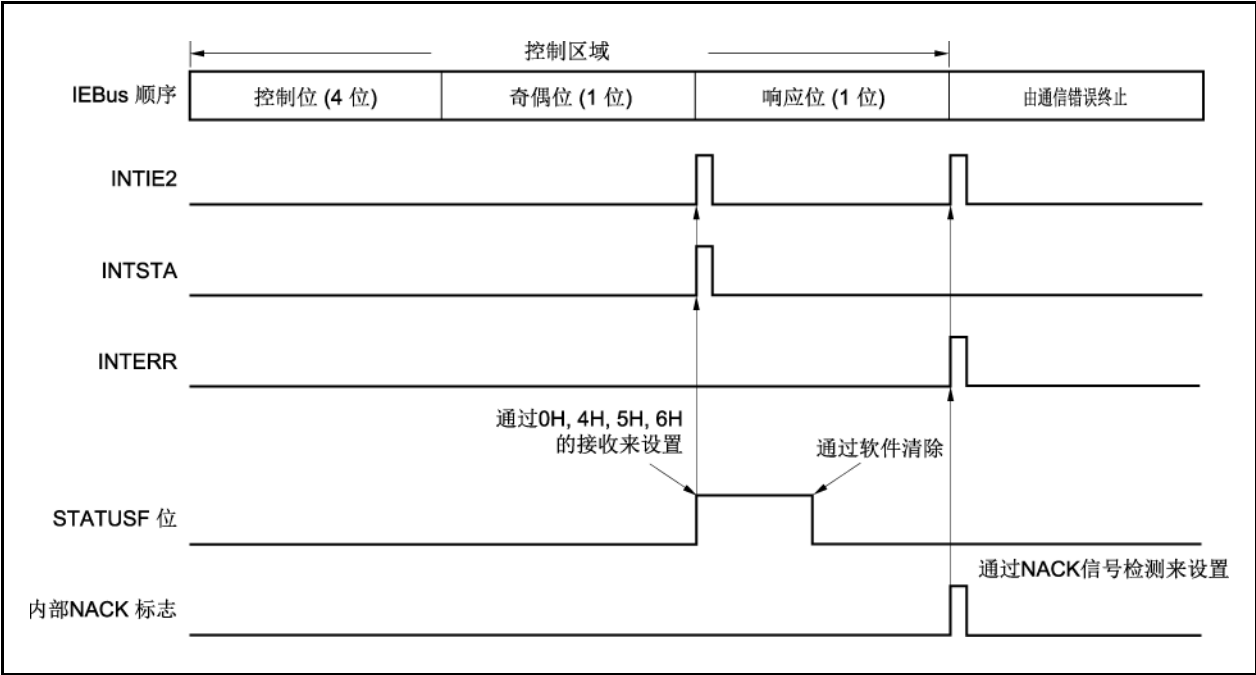


图 18-17. 中断请求信号产生时序（对于（2）和（5）来说）



IEBus 总线处于锁定状态时，因为（4）和（5）中通信是从发送锁定请求的单元之外来的，不产生开始或通信结束中断请求信号（INTIE2， INTSTA），即使 IEBus 总线单元是通信的目标。然而，一旦响应了从状态或者锁地址请求，那么 STATUSF 位置位（1）并产生状态中断请求信号（INTIE2， INTSTA）。注意当 IEBus 总线处于锁定状态时，即使接收到同一个控制数据，INTIE2 和 INTSTA 的中断产生时序依据主设备单元（3）或者另一单元（4）是否正在请求锁定状态而有所不同。

图 18-18. 锁定状态下 INTIE2 和 INTSTA 中断请求信号产生的时序（对于（4）和（5）来说）

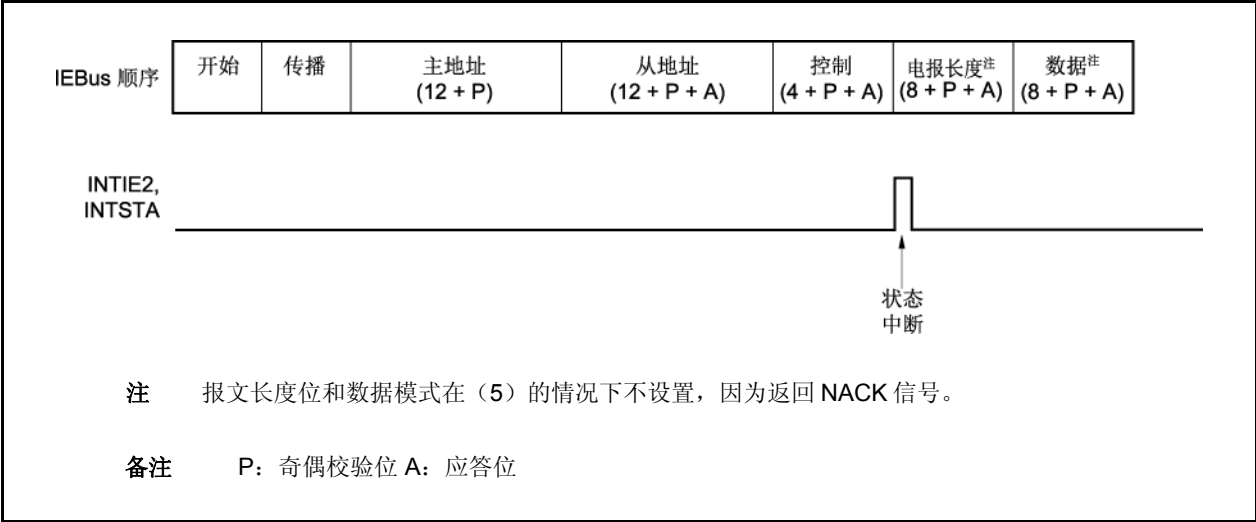
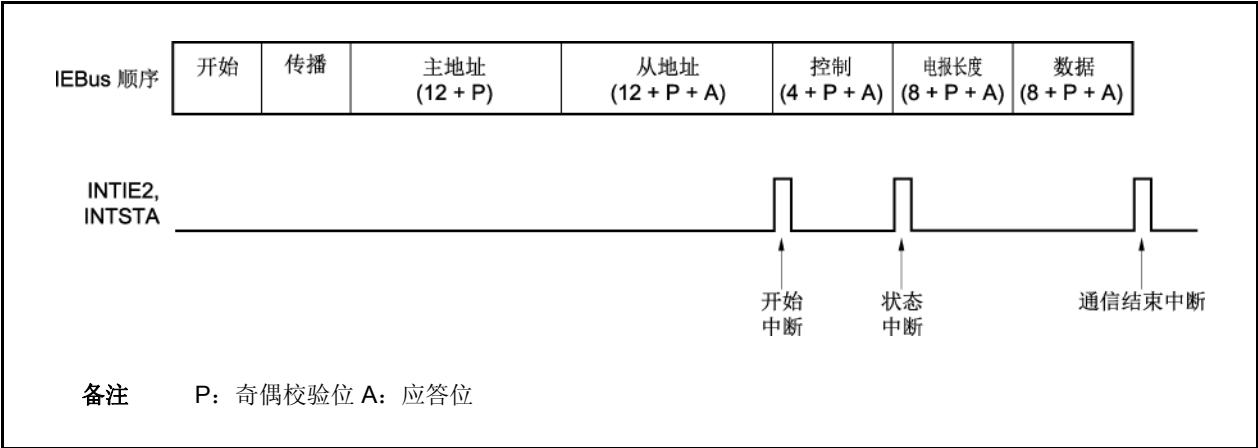


图 18-19. 锁定状态下 INTIE2 和 INTSTA 中断请求信号产生的时序（对于（3）来说）



(12) IEBus 总线报文长度寄存器 (DLR)

DLR 寄存器可以进行 8 位读写。

复位操作将此寄存器设置为 01H。

(a) 当为传输单元... 主设备传输， 从设备传输

此寄存器的数据包括在报文长度域传输的数据中， 指示传输数据的字节数。DLR 寄存器在传输之前必须预先设置。

(b) 当为接收单元... 主设备接收， 从设备接收

从设备传输单元中传输来的报文长度域中的接收数据写入此寄存器。

备注 DLR 寄存器由写寄存器和读寄存器组成。相应地， 写入此寄存器的数据不能像这样读。可以读的数据是 IEBus 总线通信期间接收到的数据。

复位后: 01H R/W 地址: FFFFF36FH

	7	6	5	4	3	2	1	0
DLR								

位								设置数据	通信数据字节的剩余数
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	1	01H	1 个字节
0	0	0	0	0	0	1	0	02H	2 个字节
:	:	:	:	:	:	:	:	:	:
0	0	1	0	0	0	0	0	20H	32 个字节
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	1	1	FFH	255 个字节
0	0	0	0	0	0	0	0	00H	256 个字节

- 注意事项**
- 1. 当主设备单元发出一个请求 (0H, 4H, 5H, 或 6H) 用于传输从设备状态或者锁地址 (高 4 位和低 8 位) 时, 不管 DLR 寄存器的内容为何值, 都传输 01H 作为报文长度。因此, 不需要通过软件设置 DLR 寄存器。
 - 2. 当 IEBus 总线控制器用作接收单元时, 如果报文长度域奇偶校验位的数值正确, 那么 DLR 寄存器存储报文长度。如果此时读取 DLR 寄存器, 读取的数值不确定。

(13) IEBus 总线数据寄存器 (DR)

DR 寄存器在位 7 到 0 之间设置通信数据 (8 位)。
此寄存器可以进行 8 位读写操作。
复位操作将此寄存器清 00H。

备注 DR 寄存器由写寄存器和读寄存器组成。相应地, 写入此寄存器的数据不能像这样读。可以读的数据是 IEBus 总线通信期间接收到的数据。

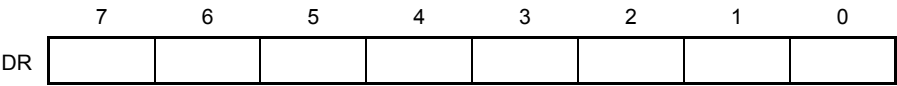
(a) 当作为传输单元时

写入 DR 寄存器中的数据 (1 个字节) 存储在 IEBus 总线接口模块的传输移位寄存器中。然后每一个字节传输时, 从最高位开始输出, 并产生中断请求信号 (INTIE1)。个体传输期间, 如果传输一个字节数据之后接收到 NACK 信号, 那么数据并不从 DR 寄存器传送到传输移位寄存器, 同一数据重传。这时, 并不产生 INTIE1 信号。
当传输移位寄存器存储 DR 寄存器的数值时, 产生 INTIE1 信号。然而, 当最后一个字节和第 32 个字节 (通信帧的最后一个字节) 储存在传输移位寄存器里时, 不产生 INTIE1 信号。

(b) 当作为接收单元时

由 IEBus 总线接口模块的接收移位寄存器接收到的数据的一个字节存储在此寄存器中。
每正确接收一个字节的数, 就产生一个中断请求信号 (INTIE1)。
当发送 / 接收数据传输到和从 DR 寄存器中来时, 使用 DMA 能够降低 CPU 处理负担。

复位后: 00H R/W 地址: FFFFF370H



- 注意事项**
- 1. 当设置传输单元期间, 如果下一个数据没有及时到, 那么产生空栈读, 并且产生通信错误中断请求信号 (INTIE2, INTERR), 停止传输。
 - 2. 个体通信接收期间, IEBus 总线控制器用作接收单元时, 如果在下一个数据读之前, 没有及时读取数据, 那么由数据域的应答位返回 NACK 信号, 要求主设备重传数据。然而, 如果数据已经达到传输字节的最大数目后不读取 DR 寄存器, 那么同时产生帧结束中断请求信号 (INTIE2, INTSTA) 和 NACK 接收错误中断请求信号 (INTIE2, INTERR)。
 - 3. 广播通信接收期间, 当 IEBus 总线控制器用作接收单元时, 如果在接收到下一个数据之前没有及时读取数据, 那么将会产生溢出错误, 并产生通信错误中断请求信号 (INTIE2, INTERR)。
 - 4. 当 IEBus 总线控制器用作接收单元时, 如果数据域的奇偶校验位正确, 那么 DR 寄存器存储接收数据。如果此时读取 DR 寄存器, 读取的数值不确定。

(14) IEBus 总线域状态寄存器 (FSR)

FSR 寄存器当中断请求信号 (INTIE1, INTIE2, INTSTA, 或 INTERR) 产生时, 存储 IEBus 总线控制器的区域状态的状态。

此寄存器只能进行 8 位读。

复位输入将此寄存器清 00H。

- 注意事项
- 1. 第三方之间通信期间, 如果产生中断请求信号, 那么 FSR 寄存器清 00H。然而, 第三方之间通信期间, 因为只有一个中断请求信号产生, 并且它是在错误发生的情况下产生的, 所以错误可以通过读取第三方错误标志 (ESR.DEFLAG 位) 得以标示。
 - 2. 当中断请求信号产生时, FSR 寄存器更新状态信息。然而, 如果此时读取 FSR 寄存器, 读取的数值不确定。
 - 3. 读 FSR 寄存器之前, 如果另一个中断请求信号产生, 那么前面的那个中断产生时的状态信息被新的中断产生时的状态信息更新。
 - 4. FSR 寄存器只用于问题分析; 实际编程不要使用它。

复位后: 00H		R	地址: FFFFF371H					
	7	6	5	4	3	2	1	0
FSR	0	0	0	0	0	0	FSTATE1	FSTATE0

备注 对于 FSTATE1 和 FSTATE0 位, 见表 **18-15 区域状态**。

表 18-15. 区域状态

区域状态	解释		
	主设备 / 从设备	区域	传输 / 接收
从设备发送状态 FSR = 00H	从设备操作	开始域	接收
		主设备单元地址域,	
		从地址域	
		控制数据域	
		报文长度域	
		数据域	
从设备传输状态 FSR = 01H	从设备操作	报文长度域	传输
		数据域	
主设备接收状态 FSR = 02H	主设备操作	报文长度域	接收
		数据域	
主设备传输状态 FSR = 03H	主设备操作	开始区域	传输
		主设备单元地址域	
		从地址域	
		控制数据域	
		报文长度域	
		数据域	

(15) IEBus 总线成功计数寄存器 (SCR)

SCR 寄存器指示剩余的通信字节数目。

从此寄存器中读取计数器的计数值。在此计数器中，由 DLR 寄存器设置的数值是由控制域中的 $\overline{\text{ACK}}$ 信号来逐步递减。当计数值达到“00H”时，通信结束标志 (ISR.ENDTRNS 位) 设置 (1)。

此寄存器 8 位只读。

复位输入将此寄存器清 00H。

复位后: 01H R 地址: FFFFF372H

	7	6	5	4	3	2	1	0
SCR								

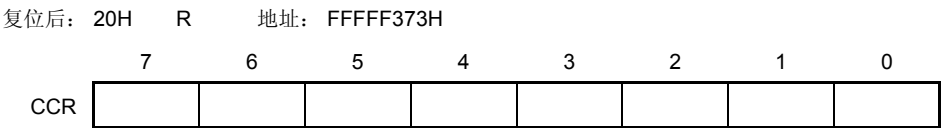
位								设置数值	通信数据字节的剩余数
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	1	01H	1 个字节
0	0	0	0	0	0	1	0	02H	2 个字节
:	:	:	:	:	:	:	:	:	:
0	0	1	0	0	0	0	0	20H	32 个字节
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	1	1	FFH	255 个字节
0	0	0	0	0	0	0	0	00H	0 个字节 (通信结束) 或者 256 个字节 [※]

注 实际计数器由 9 位组成。当读取到的是“00H”时，不能判定剩余的通信数据字节数是 0 (通信结束) 还是 256。因此，使用通信结束标志 (ENDTRNS 位)，或者在通信开始时，当发生第一个中断时，如果读取到“00H”，那么剩余的通信数据字节数目判别为 256。

注意事项 电报区域的奇偶校验位期间结束时和接收到数据域的 $\overline{\text{ACK}}$ 信号时，更新 SCR 寄存器。如果此时读取 寄存器，读取数值不确定。

(16) IEBus 总线通信计数寄存器 (CCR)

CCR 寄存器指示通信字节数目中剩余的字节数目。通信字节数目是由通信模式规定的。
此寄存器指示传送字节的数目。
每一模式中定义的每一帧的最大传输字节数目预先写入此寄存器（模式 1： 32 个字节，模式 2： 128 个字节）。
从此寄存器中读取得到的是计数器的计数值。在此计数器中，不管是 ACK 还是 NACK 信号，在数据域的应答位期间，计数值都递减。但是，SCR 寄存器是在正常通信期间（ACK 信号）减少的，而 CCR 寄存器是在当一个字节已经传送完时，不管是 ACK 还是 NACK 信号都减少。当计数值达到“00H”时， 帧结束标志（ISR.ENDFRAM 位）设置为（1）。
模式 1 中预设的每一帧传输字节的最大数目值是 20H（32 个字节），模式 2 中为 80H（128 个字节）。
此寄存器是只读的，只能以 8 位为单位进行读取。
复位操作将此寄存器设置为 20H。



注意事项 当开始位传输或接收时，传输字节的最大数目预置入 CCR 寄存器，并且当数据域的奇偶校验结束时，此寄存器数值减少。如果此时读取寄存器，得到的数值不确定。

(17) IEBus 总线时钟选择寄存器 (OCKS2)

OCKS2 寄存器选择 IEBus 总线的时钟。可以使用的主时钟频率如下所示。其它的主时钟频率不能使用。
此寄存器可以进行 8 位读写。
复位操作将此寄存器清 00H。

- 6.0 MHz/6.291456 MHz (6.29 MHz)
- 12.0 MHz/12.582912 MHz (12.58 MHz)
- 18.0 MHz/18.874368 MHz (18.87 MHz)
- 24.0 MHz/25.165824 MHz (25.16 MHz)^注
- 30.0 MHz/31.457280 MHz (31.45 MHz)^注

注 仅限 V850ES/SG2-H

复位后: 00H R/W 地址: FFFFF348H

	7	6	5	4	3	2	1	0
OCKS2	0	0	0	OCKSEN2	OCKSTH2	0	OCKS21	OCKS20

OCKSEN2	IEBus 时钟操作规范
0	停止IEBus时钟操作
1	允许IEBus时钟操作

OCKSTH2	OCKS21	OCKS20	IEBus时钟选择
0	0	0	f _{xx} /2 (当 f _{xx} = 12.0 MHz 或 f _{xx} = 12.58 MHz时)
0	0	1	f _{xx} /3 (当 f _{xx} = 18.0 MHz 或 f _{xx} = 18.87 MHz时)
0	1	0	f _{xx} /4 (当 f _{xx} = 24.0 MHz 或 f _{xx} = 25.16 MHz时) ^注
0	1	1	f _{xx} /5 (当 f _{xx} = 30.0 MHz 或 f _{xx} = 31.45 MHz时) ^注
1	0	0	f _{xx} (当 f _{xx} = 6.0 MHz 或 f _{xx} = 6.29 MHz时)
其他情况			禁止设置

注 仅限 V850ES/SG2-H

18.4 IEBus总线控制器的中断操作

18.4.1 中断控制模块

中断请求信号

<1> 通信错误	IEERR	
(i)	时序错误:	TERR
(ii)	奇偶校验错误:	PERR
(iii)	NACK 接收错误:	NERR
(iv)	空栈读错误:	UERR
(v)	溢出错误:	OERR
(vi)	写错误:	WERR
<2> 开始中断	STARTF	
<3> 状态通信	STATUSF	
<4> 通信结束	ENDTRNS	
<5> 帧结束	ENDFRAM	
<6> 传输数据写请求	$\overline{\text{STATTX}}$	
<7> 接收数据读请求	STATRX	

通信错误<1>在以上的 (i) 到 (vi) 错误源中的任何一个产生时，就发生。
这些错误源分配在错误状态寄存器中 (ESR) (见表 18-18 通信错误源处理列表)。
以上的中断信号<1> 到 <5> 分配在 ISR 寄存器中 (见表 18-17 中断源列表)。
中断控制模块的配置如下面图示。

图 18-20. 中断控制模块的配置

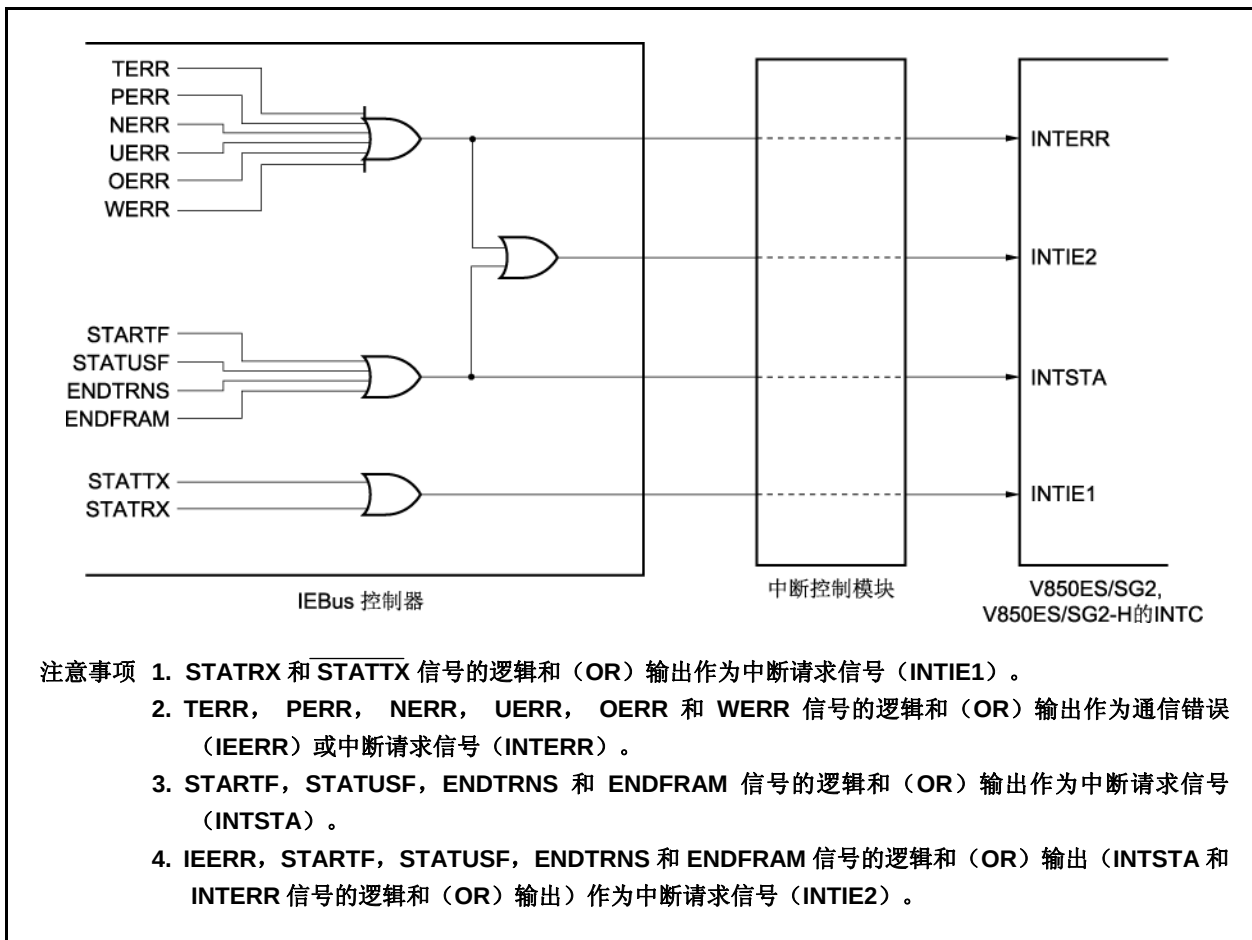


表 18-16. 中断请求信号产生源列表

中断源	符号	中断请求信号			
		INTIE1	INTIE2	INTERR	INTSTA
通信错误中断	IEERR		√	√	
时序错误	TERR				
奇偶校验错误	PERR				
NACK 接收错误	NERR				
空栈读出错误	UERR				
溢出错误	OERR				
写错误	WERR				
开始中断	STARTF		√		√
状态传输	STATUSF		√		√
通信结束	ENDTRNS		√		√
帧结束	ENDFRAM		√		√
传输数据写请求	STATTX	√			
接收数据写请求	STATRX	√			

18.4.2 确定中断的实例

IEBus 总线控制器以下列两种方式处理中断。

- 使用三个中断请求信号：INTIE1，INTERR，和 INTSTA
- 使用两个中断请求信号：INTIE1 和 INTIE2

注意事项 屏蔽不用的中断源，以使其相应中断不发生。

上面所说几种情况中的中断确认方法解释如下。

(1) 当使用 INTIE1，INTERR，和 INTSTA 信号时

图 18-21. 确认 INTIE1 信号中断的实例
(当使用 INTIE1，INTERR，和 INTSTA 信号时)

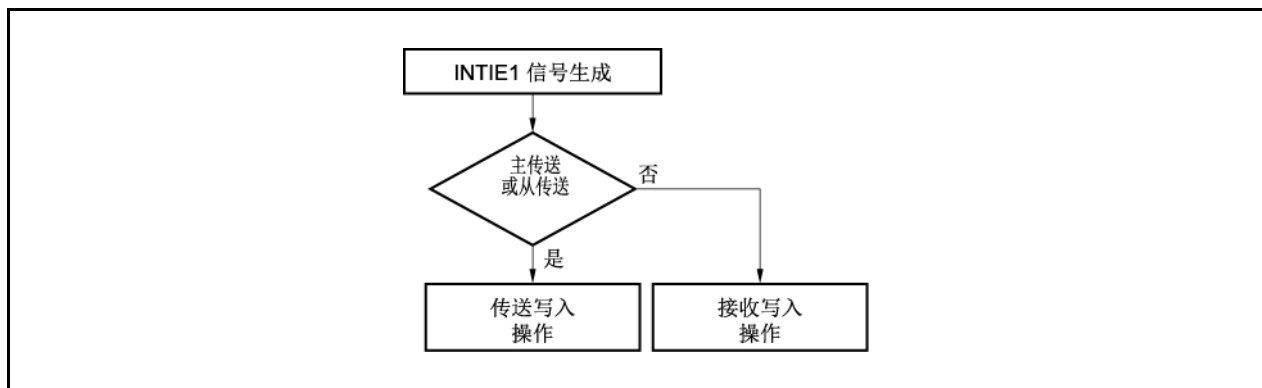


图 18-22. 确认 INTERR 信号中断的实例
(当使用 INTIE1，INTERR，和 INTSTA 信号时)

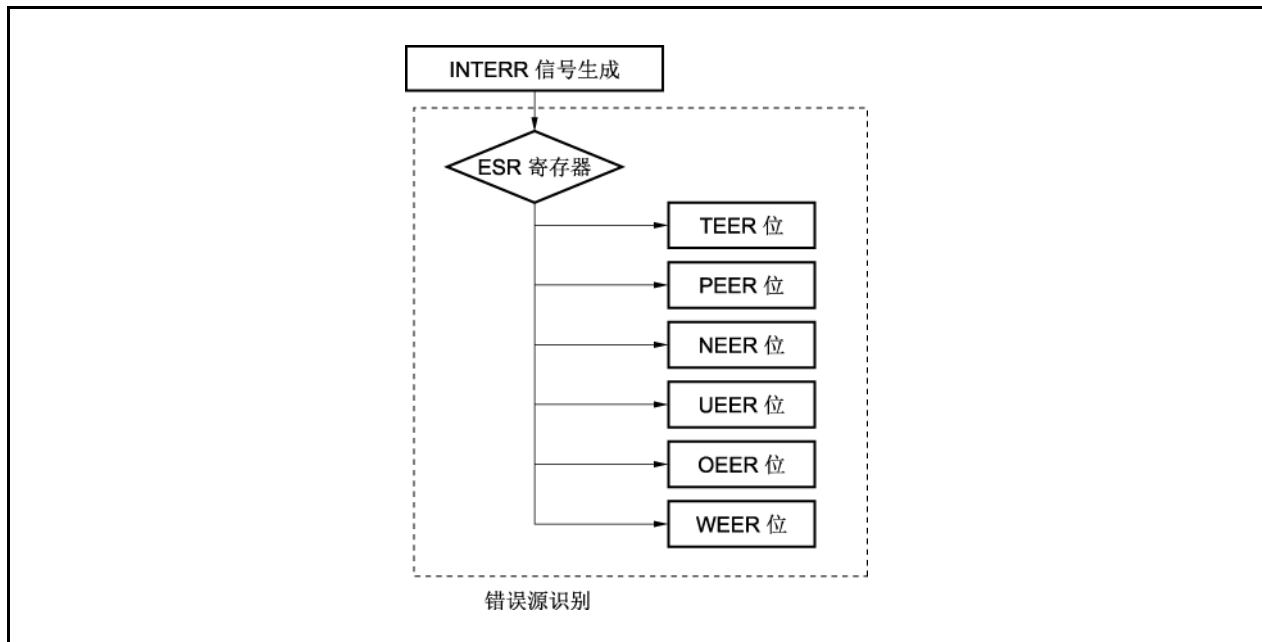
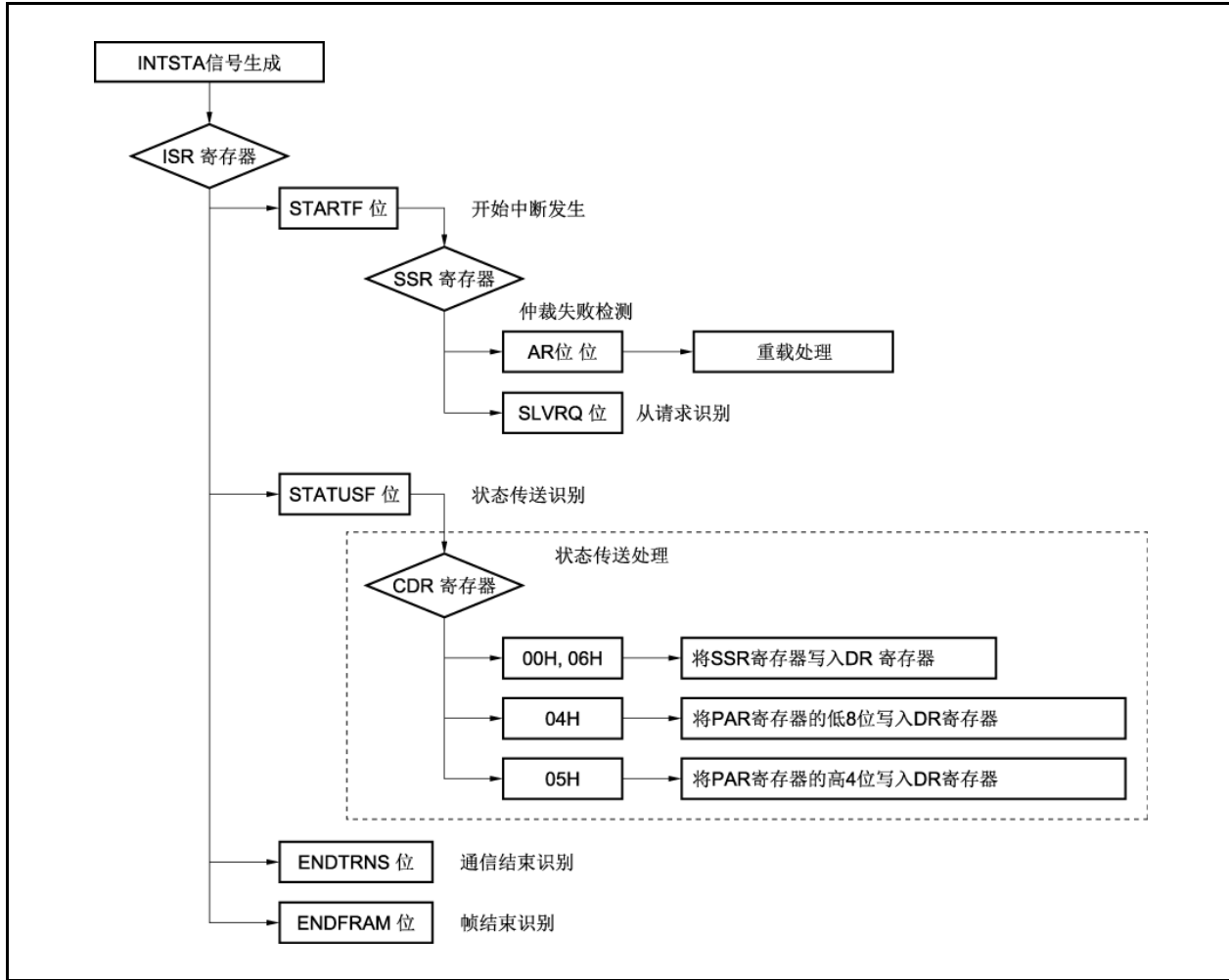


图 18-23. 确认 INTSTA 信号中断的实例
(当使用 INTIE1, INTERR, 和 INTSTA 信号时)



(2) 当使用 INTIE1 和 INTIE2 信号时

图 18-24. 确认 INTIE1 信号中断的实例 (当使用 INTIE1 和 INTIE2 信号时)

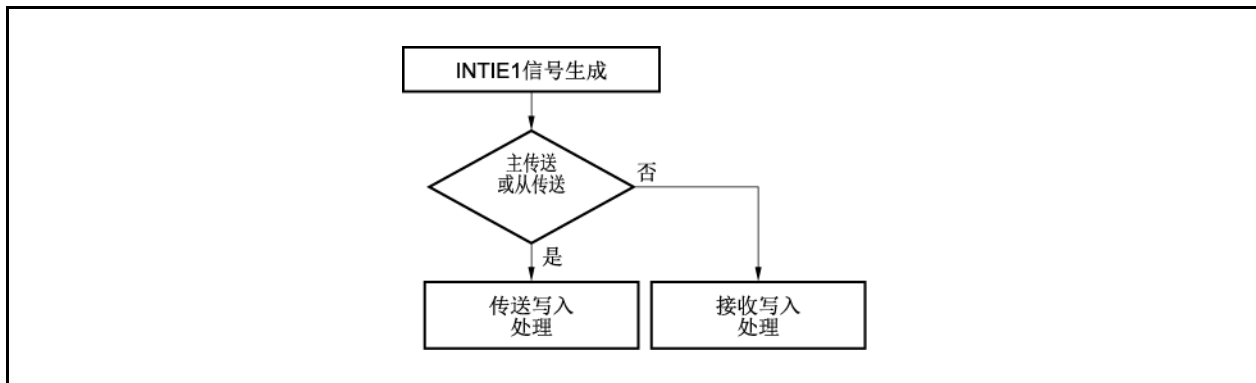
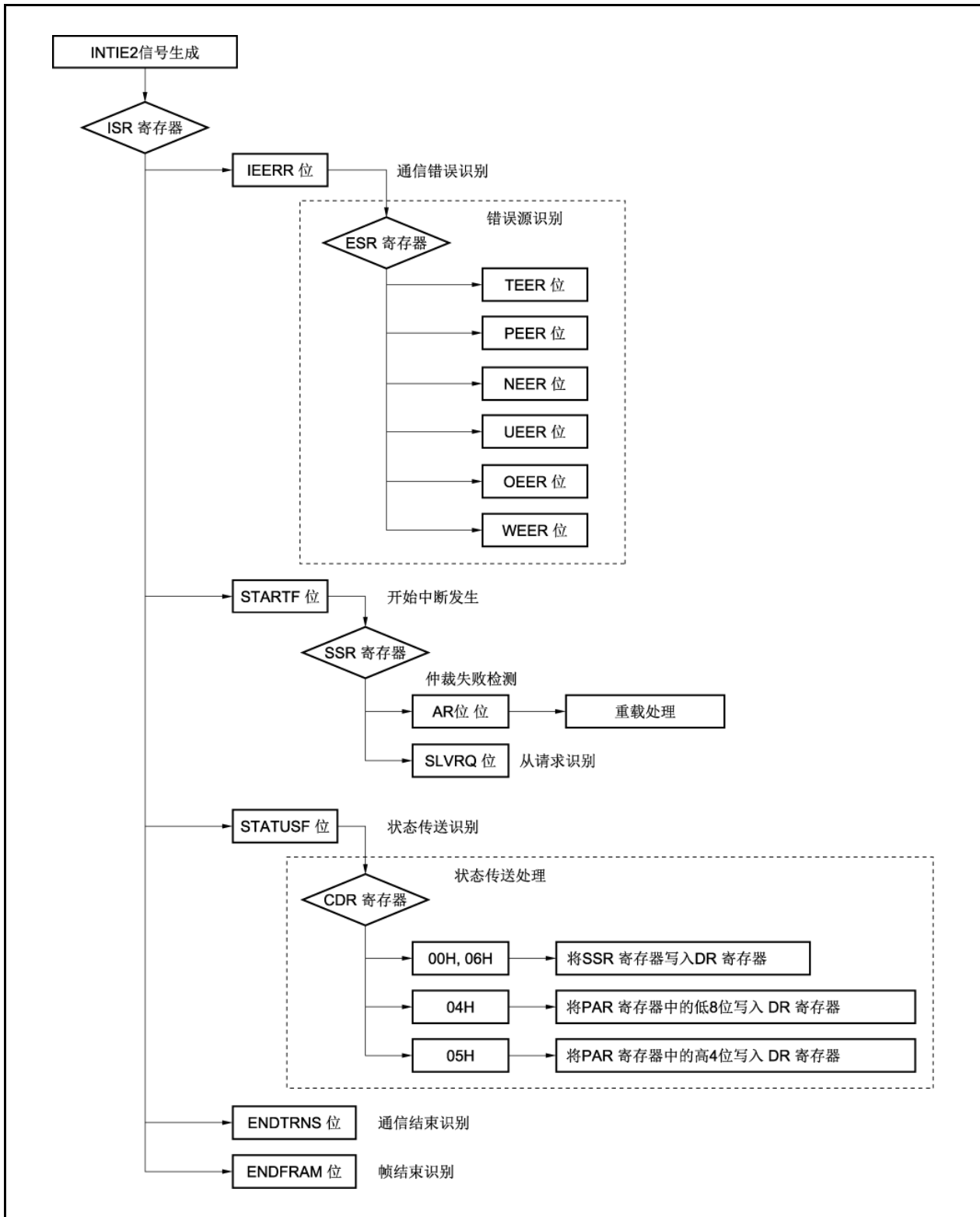


图 18-25. 确认 INTIE2 信号中断的实例（当使用 INTIE1 和 INTIE2 信号时）



18.4.3 中断源列表

<R> V850ES/SG2 和 V850ES/SG2-H 内部 IEBus 总线控制器的中断请求信号可以分成矢量中断和 DMA 传输中断。这些中断请求信号可以通过软件操作规定。

以下列出了中断源。

表 18-17. 中断源列表

中断源		产生条件		中断请求信号产生之后的软件处理	备注
		单元	区域		
通信错误	时序错误	主 / 从	所有区域	不进行通信处理	通信错误是时序错误，奇偶校验错误，NACK 接收错误，空栈读错误，溢出错误和写错误的逻辑和（OR）输出
	奇偶校验错误	接收	其它数据（个体）		
			所有区域（广播）		
	NACK 接收错误	接收（传输）	其它的数据（个体）		
	空栈读出错	传输	数据		
	溢出错误	接收	数据（广播）		
	写错误	传输	数据		
开始中断		主	从 / 地址	判定从设备请求仲裁判定（如果失败，重新处理）进行通信准备处理	主设备请求期间如果仲裁失败中断一直产生
		从	从 / 地址	判定从设备请求进行通信准备处理	只在从设备请求期间产生
状态传输		从	控制	参考状态处理实例，例如从状态	不管从传输允许标志为何，产生中断 如果控制域中返回 NACK，产生中断
通信结束	传输	数据	DMA 传输结束处理	DMA 传输结束处理 接收数据处理	如果 SCR 寄存器清 00H，就设置
	接收	数据			
帧结束	传输	数据	进行重新传输准备处理	进行重新接收准备处理	如果 CCR 寄存器清 00H，就设置
	接收	数据			
传输数据写		传输	数据	传输数据的读取 ^注	将传输数据送入内部移位寄存器后设置 传输最后一个数据时，并不进行
接收数据读		接收	数据	接收到的数据的读取 ^注	正常数据接收后设置

注 如果不执行 DMA 传输或者软件操作。

18.4.4 通信错误源处理列表

下表显示的是通信错误的发生条件（时序错误，NACK 接收错误，溢出错误，空栈读错误，奇偶校验错误和写错误），IEBus 总线控制器进行的错误处理和软件处理的实例。

表 18-18. 通信错误源处理列表（1/2）

		时序错误			
发生条件	接收	接收		传输	
	发生条件	如果位的时序不正确			
	错误源位置	数据域之外的域	数据域	数据域之外的域	数据域
广播通信	硬件处理	•接收停止 •产生 INTIE2 •开始位等待状态 备注 其它单元之间的通信并不结束		•传输停止 •产生 INTIE2 •开始位等待状态	
	软件处理	• 错误处理（例如重传请求）		•错误处理（例如重传请求）	
个体通信	硬件处理	•接收停止 •产生 INTIE2 信号 •返回 NACK 信号 •开始位等待状态		•传输停止 •产生 INTIE2 •开始位等待状态	
	软件处理	•错误处理（例如重传请求）		•错误处理（例如重传请求）	

		NACK 接收错误				
发生条件	单元设备状态	接收		传输		
	发生条件	有 NACK 信号传输		有 NACK 信号传输		
	错误源位置	数据域之外的域	数据域	数据域之外的域	数据域	第 32 个字节数据接收到 NACK 信号
广播通信	硬件处理	—	—	—	—	—
	软件处理	—	—	—	—	—
个体通信	硬件处理	- 接收停止 - 产生 INTIE2 信号 - 开始位等待状态	- 不产生 INTIE2 信号 - 接收到由其它设备单元重传的数据	- 接收停止 - 产生 INTIE2 信号 - 开始位等待状态	- 不产生 INTIE2 信号 - 重传处理	- 产生 INTIE2 信号 - 开始位等待状态
	软件处理	• 错误处理（例如重传请求）	—	• 错误处理（例如重传请求）	—	• 错误处理（例如重传请求）

表 18-18. 通信错误源处理列表 (2/2)

		溢出错误		空栈读错误 / 写错误	
发生条件	单元设备状态	接收		传输	
	发生条件	接收下一个数据之前没有及时读取 DR 寄存器		传输下一个数据之前没有及时写 DR 寄存器	
	错误源位置	数据域之外的域	数据域	数据域之外的域	数据域
广播通信	硬件处理	—	- 接收停止 - 产生 INTIE2 信号 - 开始位等待状态 备注 1. 其它单元之间的通信并不结束 2. 不能接收数据直到清除溢出状态	—	- 传输停止 - 产生 INTIE2 信号 - 开始位等待状态
	软件处理	—	- 读取 DR 寄存器, 清除溢出状态 - 错误处理 (例如重传请求)	—	错误处理 (例如重传请求)
个体通信	硬件处理	—	- 不产生 INTIE2 信号 - 返回 NACK 信号 - 数据从其它单元重传 备注 不能接收数据直到清除溢出状态	—	- 停止传输 - 产生 INTIE2 信号 - 开始位等待状态
	软件处理	—	- 读取 DR 寄存器, 清除溢出状态 - 错误处理 (例如重传请求)	—	错误处理 (例如重传请求)

		奇偶校验错误			
发生条件	设备单元状态	接收		传输	
	产生条件	接收到的数据和接收到的奇偶校验不匹配		—	
	错误源位置	数据域之外的域	数据域	数据域之外的域	数据域
广播通信	硬件处理	- 停止接收 - 产生 INTIE2 信号 - 开始位等待状态 备注 其它单元之间的通信并没有结束		—	—
	软件处理	错误处理 (例如重传请求)		—	—
个体通信	硬件处理	- 停止接收 - 产生 INTIE2 信号 - 开始位等待状态	- 接收不停止 - 不产生 INTIE2 信号 - 返回 NACK 信号 - 接收到其它单元重传的数据	—	—
	软件处理	错误处理 (例如重传请求)	—	—	—

18.5 中断请求信号产生时序和主CPU处理

18.5.1 主传输

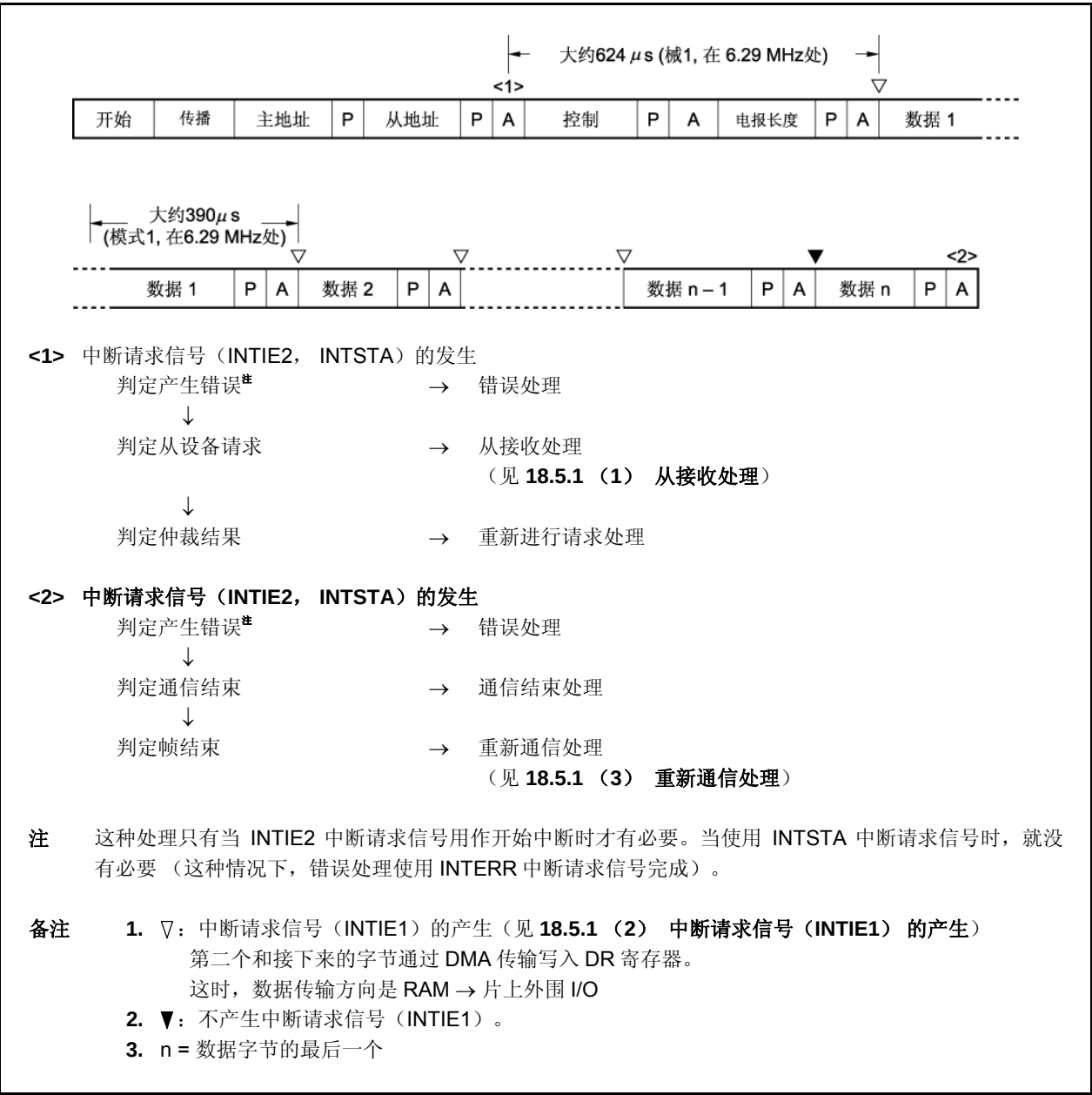
初始准备处理：

设置单元地址，从地址，控制数据，报文长度和传输数据的第一个字节

通信开始处理：

设置 BCR 寄存器（允许通信，主设备请求和从接收）。

图 18-26. 主传输



(1) 从接收处理

如果矢量中断服务处理程序期间，从接收请求得到确认，那么宏处理的数据传输方向必须从 RAM → 片上外围 I/O 到片上外围 I/O → RAM 直到接收到第一个数据。通信模式 1 中，数据传输方向改变处理的最大挂起时间约是 1,040 μs（在 6.29 MHz 时）。

(2) 中断请求信号（INTIE1）的产生

如果数据域中从设备单元接收到 NACK 信号，就不发出中断请求信号（INTIE1）给中断控制器（INTC），同样的数据由硬件重传。

如果写下一个数据期间没有及时写传输数据，由于产生空栈读，就发生通信错误中断请求信号（INTERR），通信中途结束。

(3) 重新通信处理

在图 18-26<2>的矢量中断服务程序中，要判定一帧之内数据是否正确传输。如果没有正确传输，（一帧之内要传输的数据数不能传输），一定要在下一帧重传数据，剩下的数据也必须传输。

18.5.2 主接收

在进行主接收之前，有必要告知将要成为从设备传输的从设备的单元。因此，为了主接收，两个以上的通信帧十分必要。

从单元准备传输数据，设置从传输允许标志（BCR. ENSLVTX 位）为（1），并等待。

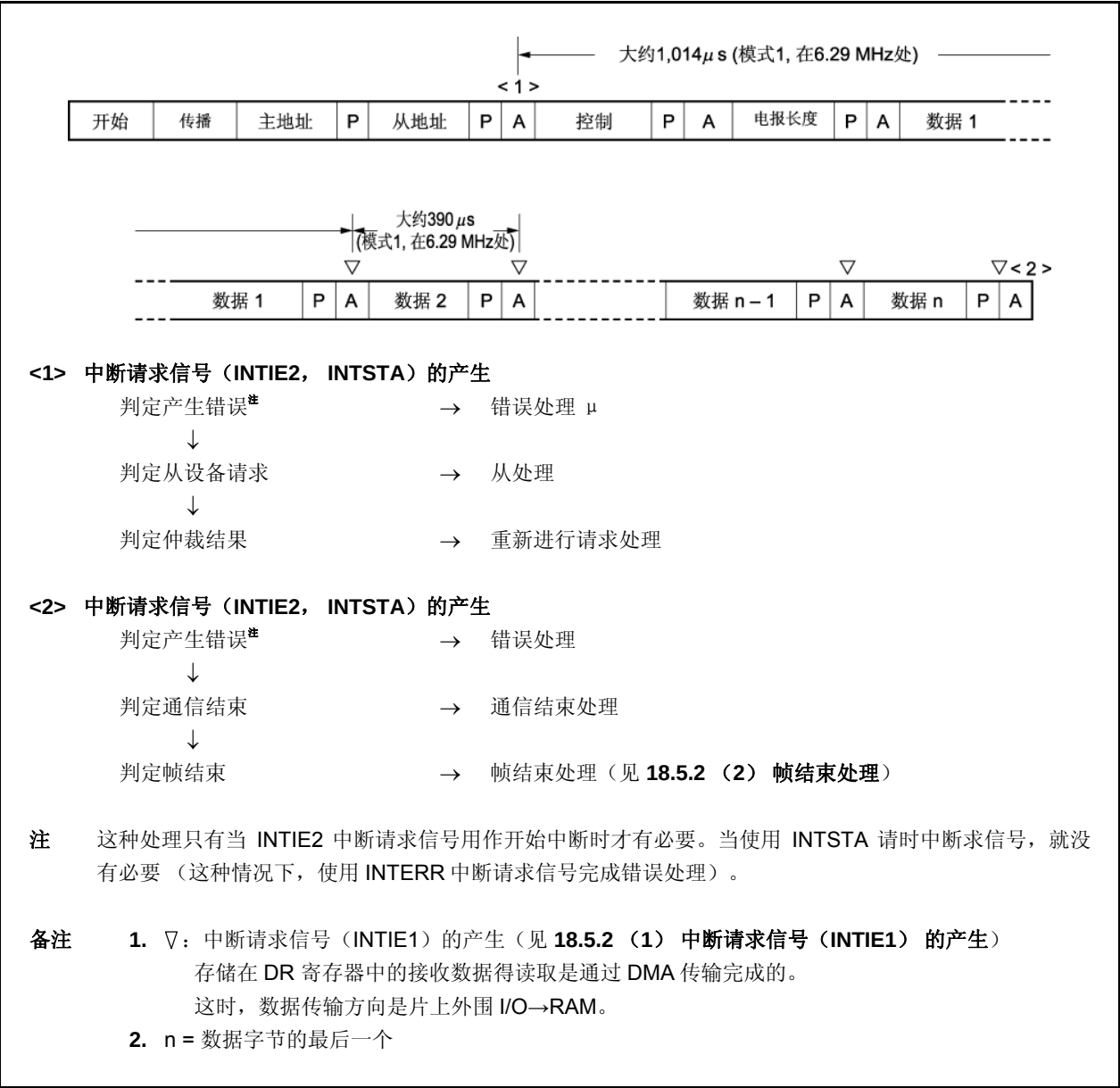
初始准备处理：

设置单元地址，从地址和控制数据。

通信开始处理：

设置 BCR 寄存器（允许通信和主设备请求）。

图 18-27. 主接收



(1) 中断请求信号 (INTIE1) 产生

如果数据域中没有传输（硬件处理）NACK 信号，那么中断请求信号 (INTIE1) 不发给 INTC，同样的数据从设备重传。

如果到接收下一个数据时没有来得及读接收当前数据，硬件自动传输 NACK 信号。

(2) 帧结束处理

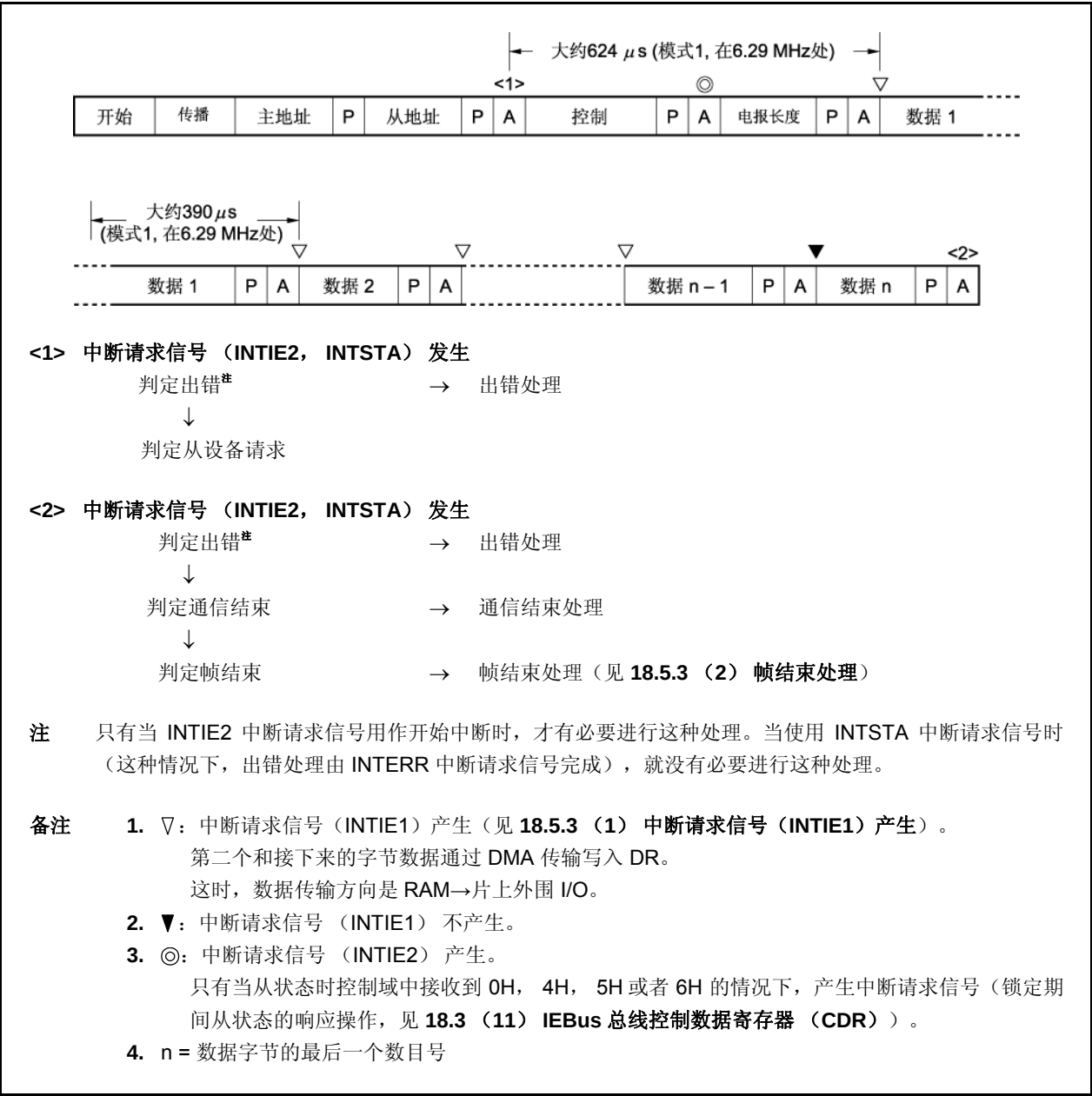
在图 18-27<2>中的矢量中断服务处理中，判定数据是否在一帧之内正确传输。如果数据没有正确接收（如果要在帧中传输的数据数目没有接收），那么在下一个通信帧中，必须请求从设备重传数据。

18.5.3 从传输

初始准备处理：
设置单元地址，报文长度和传输数据的第一个字节。

通信开始处理：
设置 BCR 寄存器（允许通信，从设备传输和从接收）。

图 18-28. 从传输



(1) 中断请求信号 (INTIE1) 产生

如果数据域中没有接收到从主设备单元来的 NACK 信号，那么中断请求信号 (INTIE1) 不发给 INTC，同样的数据从硬件重传。

如果写下一个数据期间传输数据没有及时写入，由于空栈，将产生通信错误中断请求信号 (INTERR)，并且通信异常结束。

(2) 帧结束处理

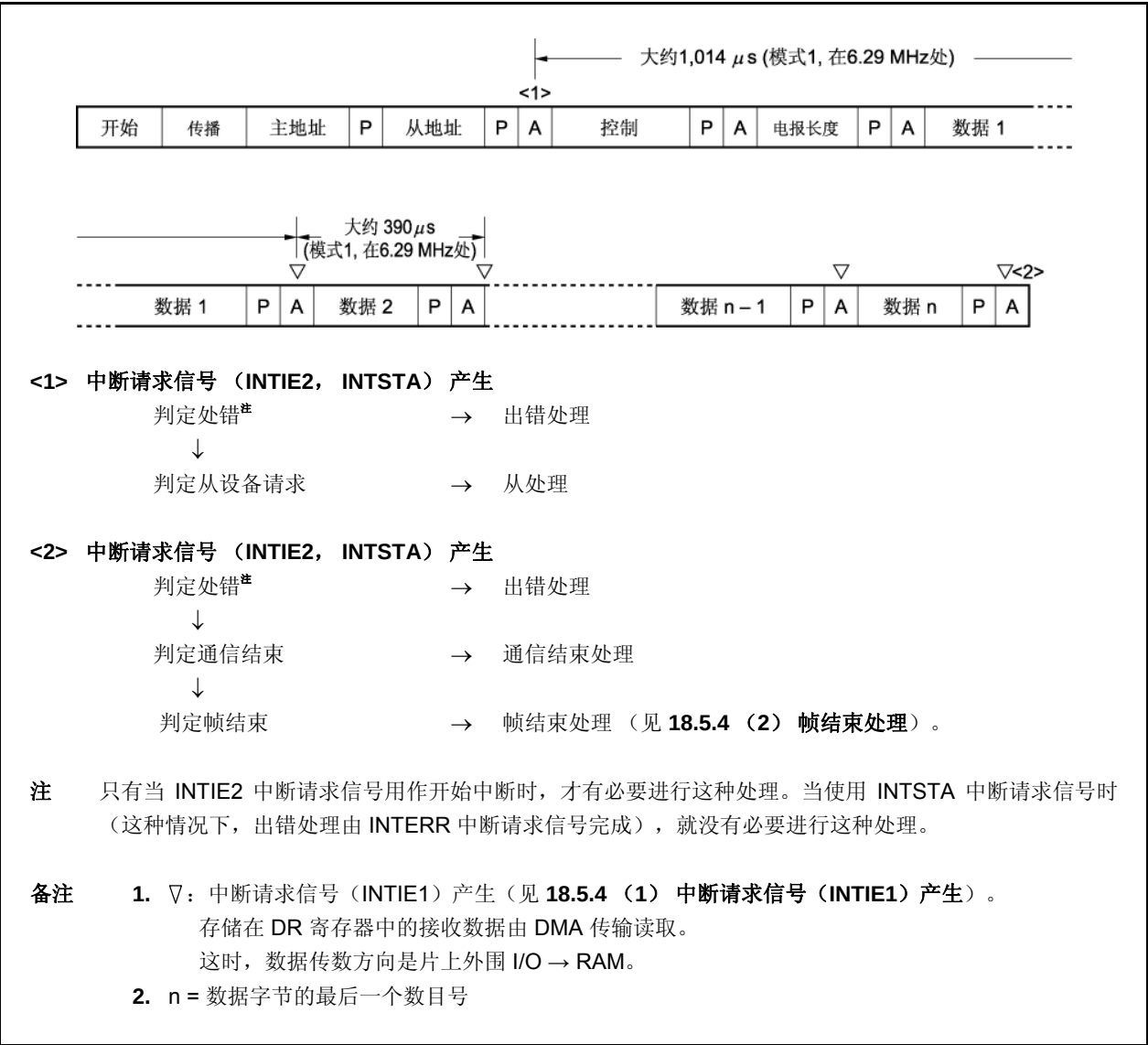
图 18-28 的<2>中的矢量中断服务处理中，判定数据是否在一帧之内正确传输。如果数据没有正确传输（如果要在帧中传输的数据数目没有传输），数据必须在下一帧中重传，剩下的数据也必须传输。

18.5.4 从接收

初始准备处理：
设置单元地址。

通信开始处理：
设置 BCR 寄存器（允许通信，禁止从设备传输，并允许从接收）。

图 18-29. 从接收



(1) 中断请求信号 (INTIE1) 产生

如果数据域中传输 NACK 信号，那么中断请求信号 (INTIE1) 不发给 INTC，同样的数据也不从主设备单元重传。

如果当前接收的数据在接收下一个数据时，还没有读取，那么 NACK 信号自动传输。

(2) 帧结束处理

图 18-29 的<2>中的矢量中断服务处理中，判定数据是否在一帧之内正确接收。

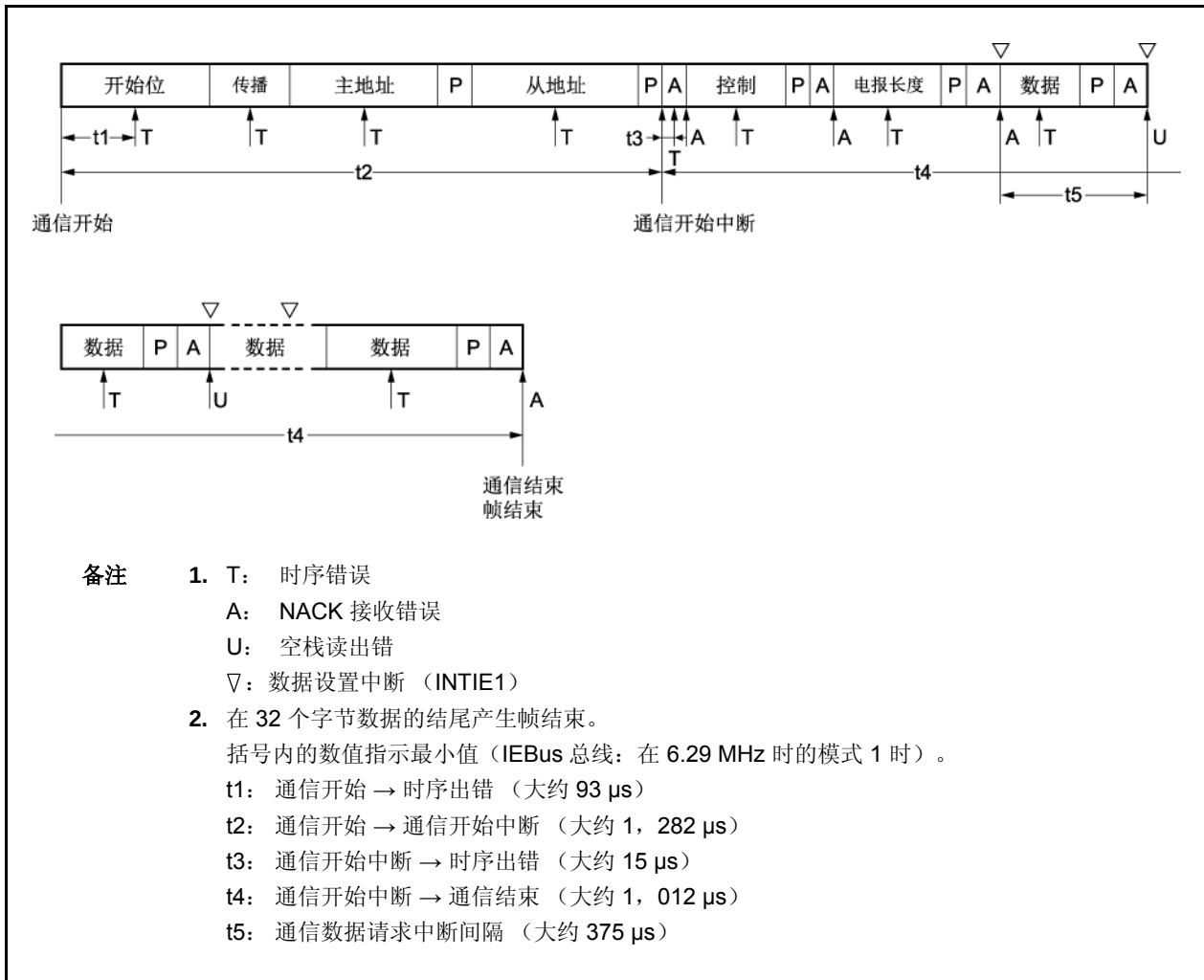
18.5.5 IEBus总线控制的中断请求信号产生间隔

通信的每一点上，每一个控制中断请求必然产生，并执行必要的处理直至下一个中断请求信号产生。因此，IEBus总线控制模块由软件控制时，要考虑此中断请求信号发生最短时间间隔。

下面的中断请求信号在可能发生的区域中用 ↑ 标示。↑ 并不意味着在每一个标注它的地方中断请求信号一定产生。如果中断请求信号出错（时序错误，奇偶校验错误或者 NACK 接收错误）产生，IEBus 总线内部电路就会初始化。结果就是其后的中断请求信号并不在那一通信帧中产生。

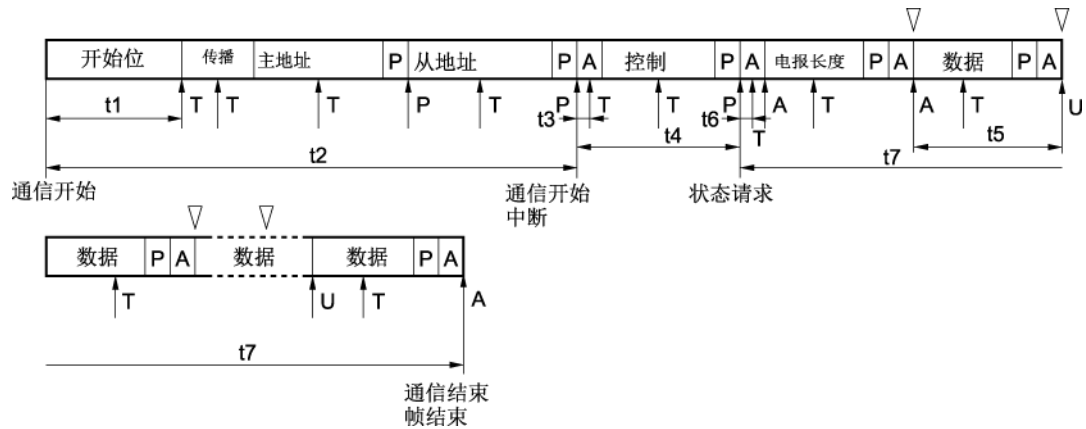
(1) 主传输

图 18-30. 主传输（中断请求信号产生的间隔）



(3) 从传输

图 18-32. 从设备传输（中断请求信号产生的间隔）

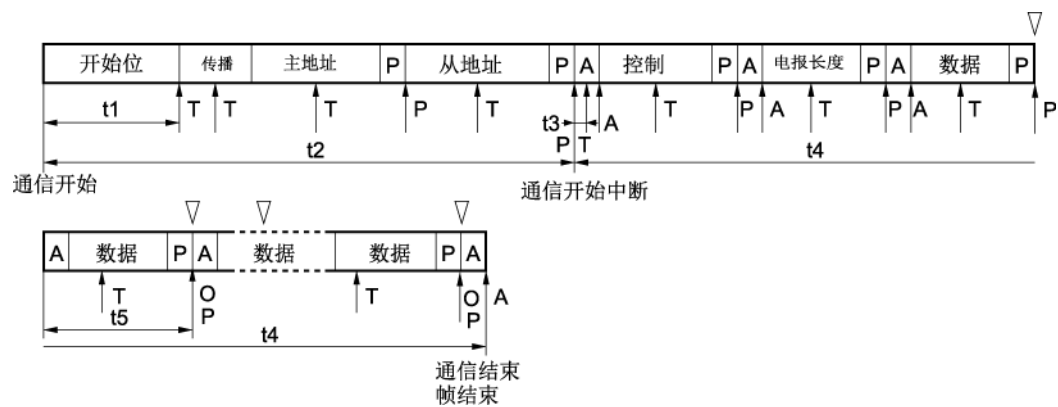


备注

1. T: 时序错误
P: 奇偶校验出错
A: NACK 接收错误
U: 空栈读出错
▽: 数据设置中断 (INTIE1)
2. 在 32 个字节数据的结尾产生帧结束。
括号内的数值指示最小值 (IEBus 总线: 在 6.29 MHz 时的模式 1 时)。
t1: 通信开始 → 时序出错 (大约 196 μs)
t2: 通信开始 → 通信开始中断 (大约 1, 192 μs)
t3: 通信开始中断 → 时序出错 (大约 15 μs)
t4: 通信开始中断 → 通信结束 (大约 225 μs)
t5: 通信数据请求中断间隔 (大约 375 μs)
t6: 状态请求 → 时序出错 (大约 15 μs)
t7: 状态请求 → 通信结束 (大约 787 μs)

(4) 从接收

图 18-33. 从接收（中断请求信号产生的间隔）



- 备注
1. T: 时序错误
P: 奇偶校验出错
A: NACK 接收错误
O: 溢出错误
▽: 数据设置中断 (INTIE1)
 2. 在 32 个字节数据的结尾产生帧结束。
括号内的数值指示最小值 (IEBus 总线: 在 6.29 MHz 时的模式 1 时)。
t1: 通信开始 → 时序出错 (大约 196 μ s)
t2: 通信开始 → 通信开始中断 (大约 1, 192 μ s)
t3: 通信开始中断 → 时序出错 (大约 15 μ s)
t4: 通信开始中断 → 通信结束 (大约 1, 012 μ s)
t5: 接收数据读间隔 (大约 375 μ s)

第 19 章 CAN 控制器

<R> **注意事项** CAN 控制器被布置在可编程周边 I/O 区域。
 在使用 CAN 控制器前，通过使用 BPC 寄存器使能可编程周边 I/O 区域。
 关于细节，参阅 3.4.7 可编程周边 I/O 寄存器。

19.1 概要

<R> V850ES/SG2 和 V850ES/SG2-H 具有符合 ISO11898 规范的 CAN 协议的片上 1 通道 CAN 总线（控制器局域网）控制器。

<R> 具有片上 CAN 总线控制器的 V850ES/SG2 和 V850ES/SG2-H 产品如下。

<R>

- μ PD703280, 703280Y, 703281, 703281Y, 70F3281, 70F3281Y, 703282, 703282Y, 703283, 703283Y, 70F3283, 70F3283Y, 703282HY, 703283HY, 70F3283HY

19.1.1 特性

- 符合 ISO11898 并根据 ISO/DIS16845（CAN 一致性测试）测试
- 允许使用标准帧和扩展帧发送/接收
- 传输速率：最大 1 Mbps（CAN 时钟输入 ≥ 8 MHz）
- 32 个信息缓冲器/通道
- 接收 / 发送历史列表功能
- 自动 block 发送功能
- 多重缓冲器接收 block 功能
- 每一通道有 4 种形式的屏蔽设置

19.1.2 功能概要

表 19-1 表示 CAN 总线控制器的功能概要。

表 19-1. 功能概要

功能	详细消息
协议	CAN 协议 ISO 11898 （标准和扩展帧发送 / 接收）
波特率	最大 1 Mbps （CAN 时钟输入 ≥ 8 MHz）
数据存储	在 CAN RAM 中存储信息
信息数目	32 个信息缓冲器 / 通道 - 每一信息缓冲器可以设定为发送信息缓冲器或接收信息缓冲器
信息接收	- 每一信息缓冲器都可以设置唯一的 ID - 每一通道有 4 种形式的屏蔽设置 - 每次接收到信息并存储到信息缓冲器之后就产生接收完成中断 - 两个或者更多的接收信息缓冲器可以用于 FIFO 接收缓冲器（多重缓冲器接收 block 功能） - 接收历史列表功能
信息发送	- 每一信息缓冲器都可以设置唯一的 ID - 发送完成中断针对每一个信息缓冲器 - 编号从 0 到 7 用作发送信息缓冲器的信息缓冲器可以被用来自动 block 发送。信息发送间隔是可编程的（自动 block 发送功能（此后以“ABT”代表此功能）） - 发送历史列表功能
远程帧协议	通过发送信息缓冲器进行远程帧处理
<R> 时间标记功能	- 当 16 位定时器被结合使用时，时间标记功能可以用于接收信息。 - 可选择时间标记捕捉触发器（在 CAN 总线信息帧中可以检测 SOF 或 EOF）
诊断功能	- 可读取的错误计数器 “Valid protocol operation flag”用于总线连接的验证 - 只接收模式 - 单发模式 - CAN 协议错误类型解码 - 自检测模式
从 bus-off 状态释放	- 可以通过软件从 bus-off 状态强制释放（忽略时序限制）。 - 不能自动地从 bus-off 状态释放（需要软件释放请求）
节能模式	- CAN sleep 模式（可通过 CAN 总线唤醒） - CAN stop 模式（不能通过 CAN 总线唤醒）

19.1.3 配置

CAN 控制器有如下四个功能块。

(1) NPB 接口

这个功能块提供 NPB（NEC 周边 I/O 总线）接口和在 CAN 模块和主机 CPU 之间发送和接收信号的方式。

(2) MCM（存储器控制模块）

此功能块控制对 CAN 协议层和 CAN 模块中 CAN RAM 的访问。

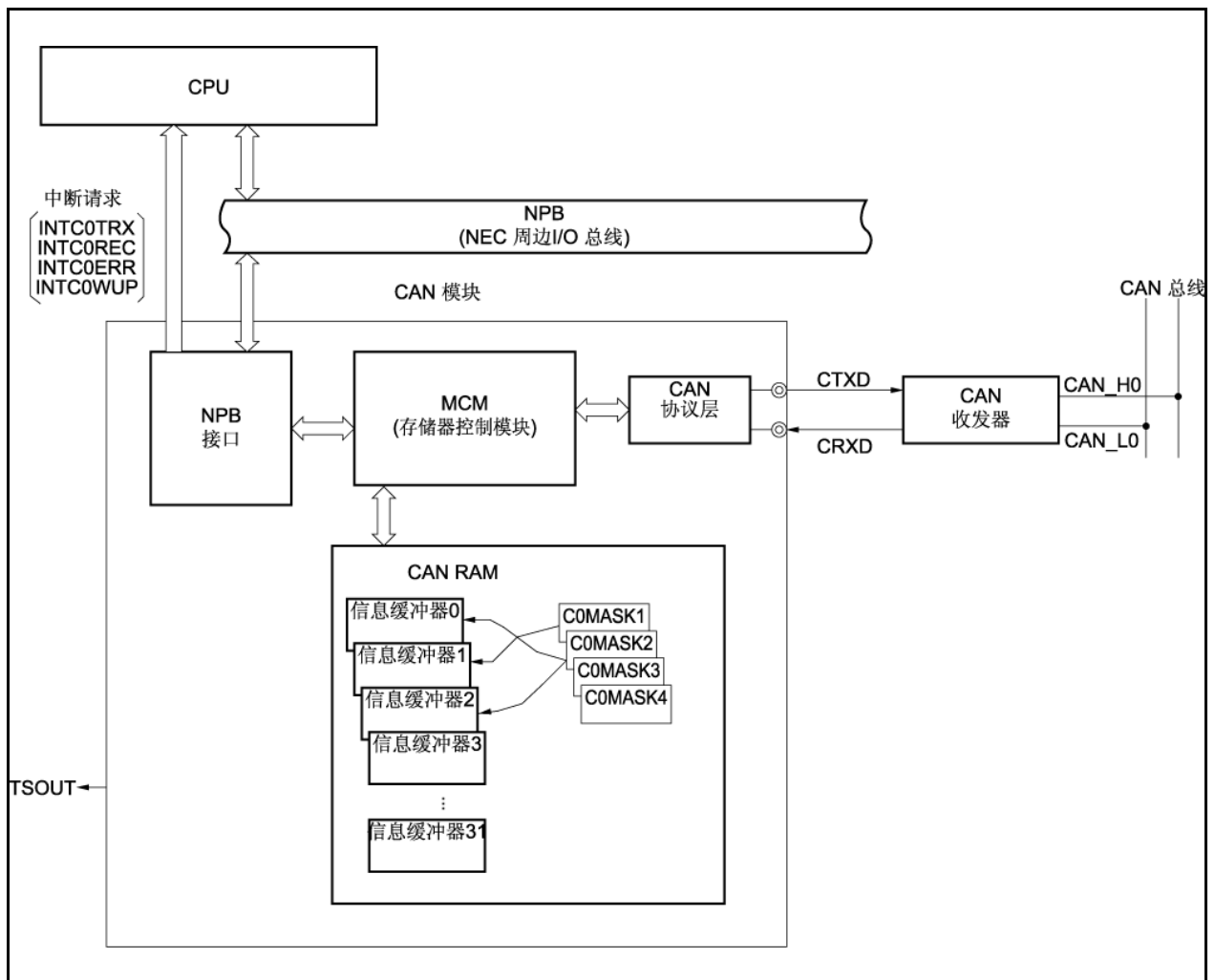
(3) CAN 协议层

此功能块涉及 CAN 协议和它相关设定的操作。

(4) CAN RAM

这是 CAN 存储器功能块，被用于存储信息 ID，信息数据等。

图 19-1. CAN 模块的框图



19.2 CAN协议

CAN（控制器局域网）是一个在汽车应用领域（C 级）中用于实时通信的高速的多路通信的协议。ISO11898 规范对 CAN 做了描述。详细信息，参看 ISO11898 规范。

CAN 规范一般分为两层：物理层和数据链路层。数据链路层包括逻辑链路和媒介访问控制。这些层显示如下。

图 19-2. 层的构成



19.2.1 帧的格式

(1) 标准帧格式

- 标准帧格式使用 11 位的标识域，也就是可以表示 2048 种信息。

(2) 扩展帧格式

- 扩展帧格式使用 29 位（11 位+18 位）的标识域，表示数增加到 2048×2^{18} 种信息。
- 当仲裁域的 SRR 和 IDE 位都被设定为“隐性电平”（CMOS 电平为 1）时，表示使用扩展帧格式。

19.2.2 帧类型

在 CAN 协议中使用以下 4 种帧类型。

表 19-2. 帧类型

帧类型	描述
数据帧	数据帧用于发送数据
远程帧	远程帧用于请求数据帧
错误帧	错误帧用于报告错误检测
过载帧	过载帧用于延迟下一个数据帧或远程帧

(1) 总线状态

总线状态分为显性和隐性两种。

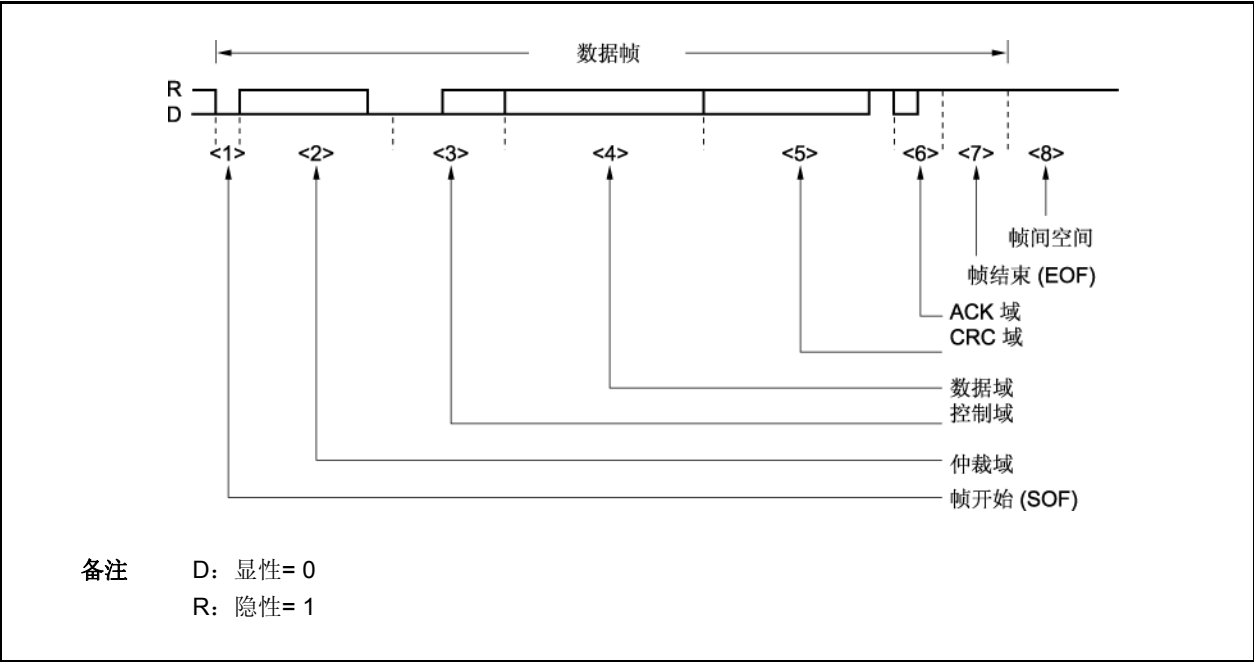
- 显性电平用逻辑 0 来表示。
- 隐性电平用逻辑 1 来表示。
- 显性电平和隐性电平同时传送时，总线电平值为显性。

19.2.3 数据帧和远程帧的格式

(1) 数据帧

数据帧由七个部分组成。

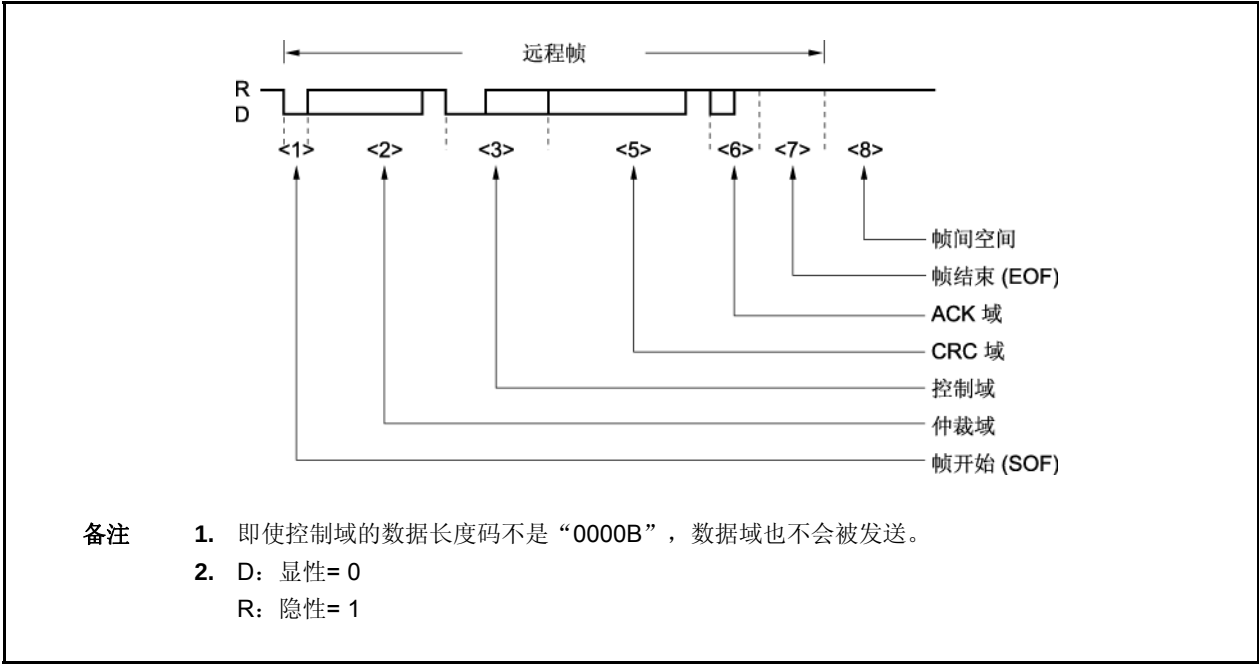
图 19-3. 数据帧



(2) 远程帧

远程帧由六个部分组成。

图 19-4. 远程帧

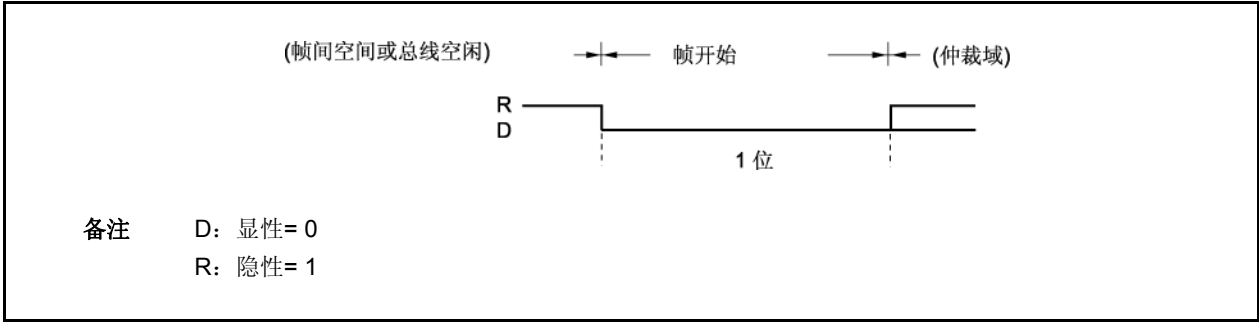


(3) 区域描述

<1> 帧开始 (SOF)

帧开始区域是数据帧或远程帧的起始位。

图 19-5. 帧起始区域 (SOF)



- 如果在总线空闲状态下检测到显性电平，则执行一个硬件同步（当前 TQ 被定义为 SYNC 段）。
- 如果在该硬件同步之后的采样点采样到显性电平，那么认为这个位为 SOF。如果检测到的是隐性电平，则协议层返回到总线空闲状态并且认为前面的显性脉冲只是噪音脉冲。这种情况下不会产生错误帧。

<2> 仲裁域

仲裁域用于设置优先级，数据帧 / 远程帧和帧格式。

图 19-6. 仲裁域（标准格式模式）

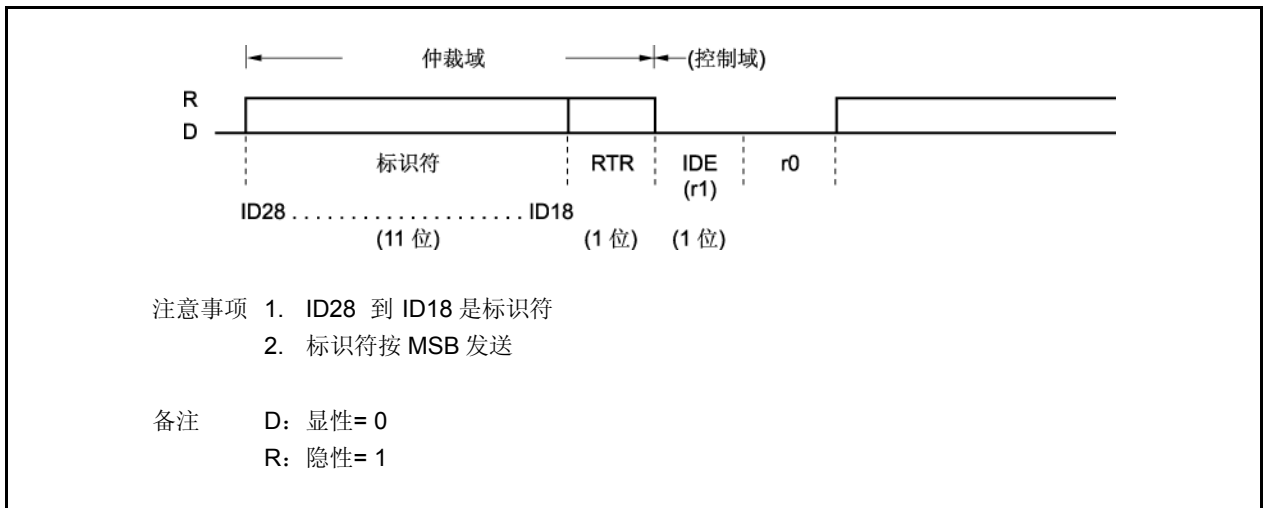


图 19-7. 仲裁域（扩展格式模式）

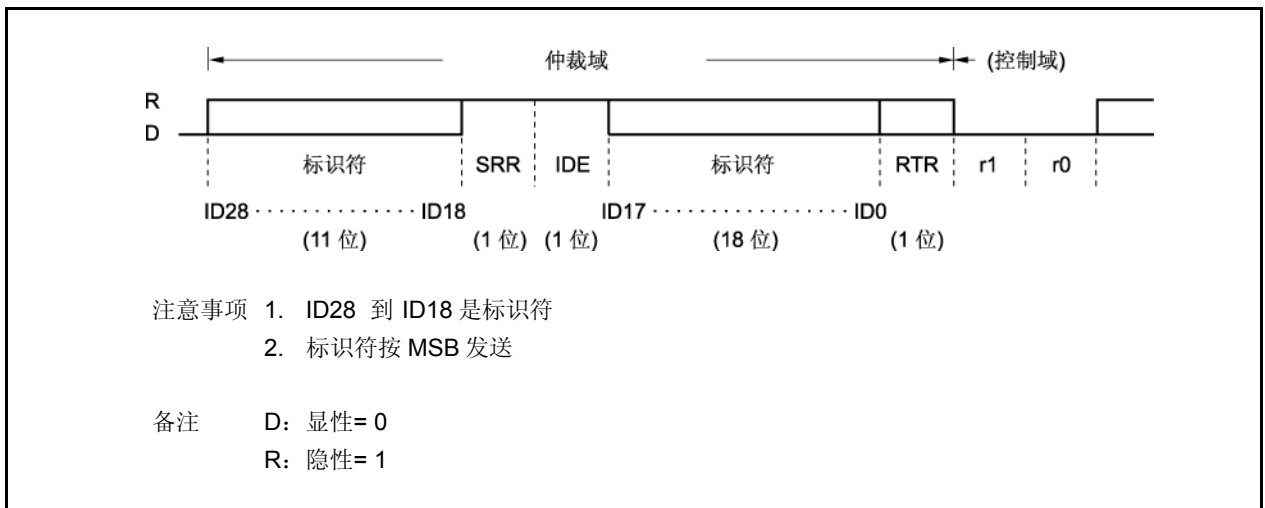


表 19-3. RTR 帧设定

帧类型	RTR 位
数据帧	0 (D)
远程帧	1 (R)

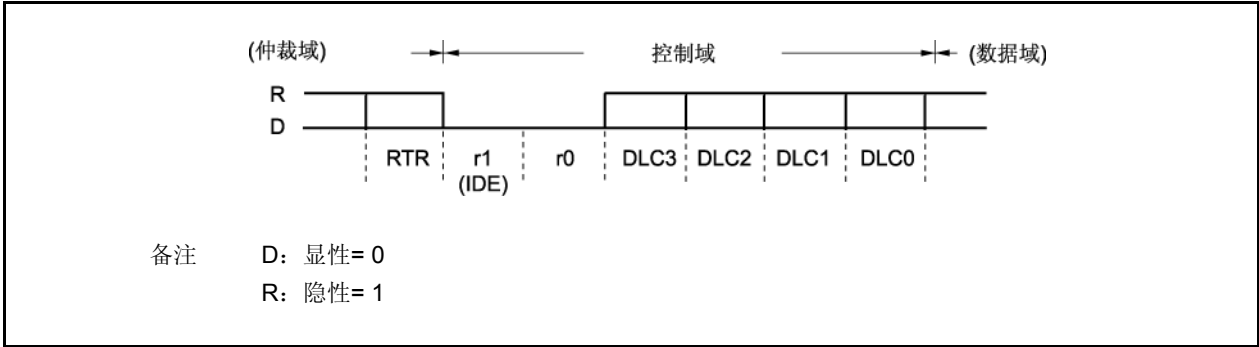
表 19-4. 帧格式设定 (IDE 位) 和标识号码 (ID) 位

帧格式	SRR 位	IDE 位	位数
标准格式模式	无	0 (D)	11 位
扩展格式模式	1 (R)	1 (R)	29 位

<3> 控制域

控制域使用“DLC”来表示数据域中的数据字节数（DLC = 0 ~ 8）。

图 19-8. 控制域



在标准格式帧中，控制域的 IDE 位与 r1 位相同。

表 19-5. 数据长度设定

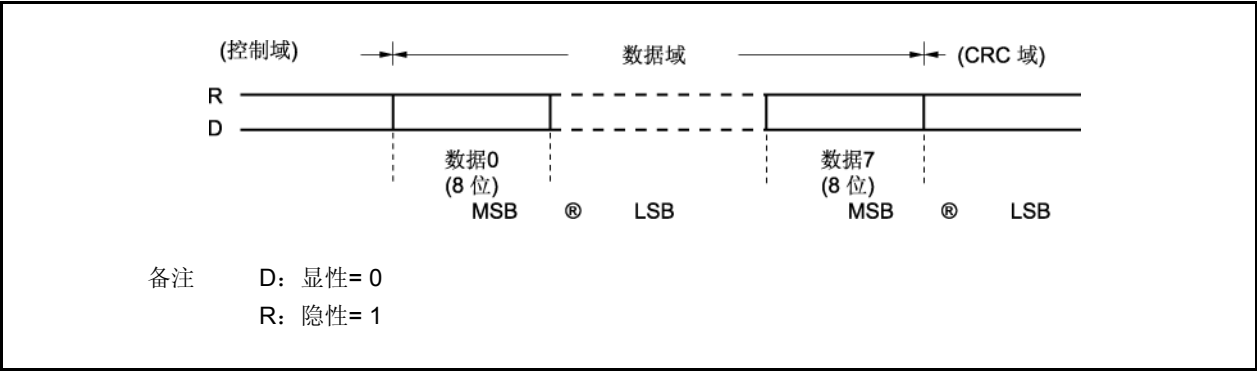
数据长度码				数据字节计数
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0 字节
0	0	0	1	1 字节
0	0	1	0	2 字节
0	0	1	1	3 字节
0	1	0	0	4 字节
0	1	0	1	5 字节
0	1	1	0	6 字节
0	1	1	1	7 字节
1	0	0	0	8 字节
其他				不管 DLC3 ~ DLC0 的值是什么，都是 8 字节

注意事项 在远程帧中，即使数据长度码不是 0000B，也不会有数据域。

<4> 数据域

数据域包含由控制域设置长度的数据（以字节为单位）。数据长度最多可设置为 8 字节。

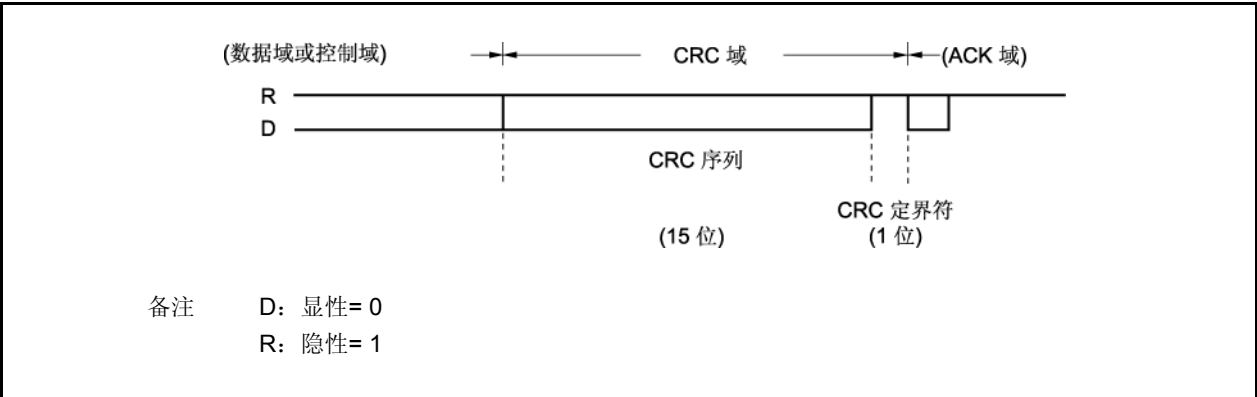
图 19-9. 数据域



<5> CRC 域

CRC 域是一个 16 位区域，用于发送数据的错误校验。

图 19-10. CRC 域

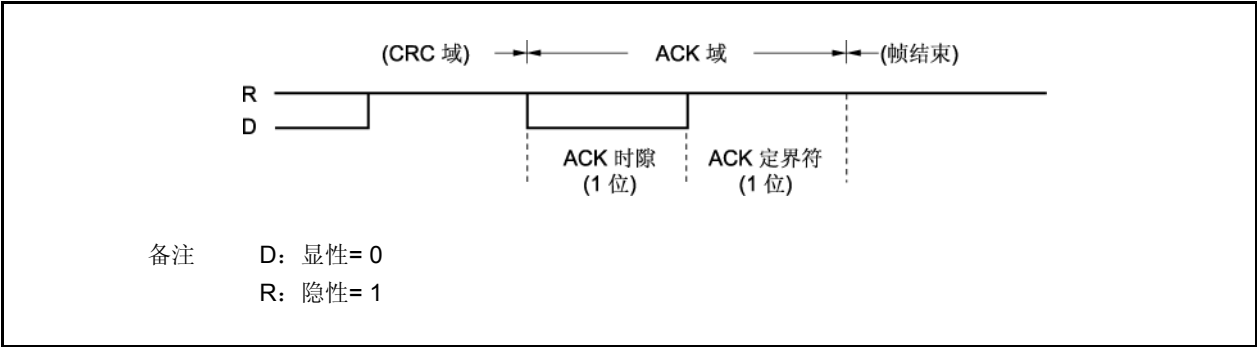


- 此多项式 $P(X)$ 用于产生 15 位 CRC 序列，如下：
$$P(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$$
- 发送节点：按规律从帧开始、仲裁域、控制域和数据域中的数据传输 CRC 序列
- 接收节点：将从接收到的数据中（不含位填充）计算得出的 CRC 序列与 CRC 域中的 CRC 序列相比较。如果两个 CRC 序列不相等，那么节点将发出错误帧。

<6> ACK 域

ACK 域用于对正常接收做出应答。

图 19-11. ACK 域

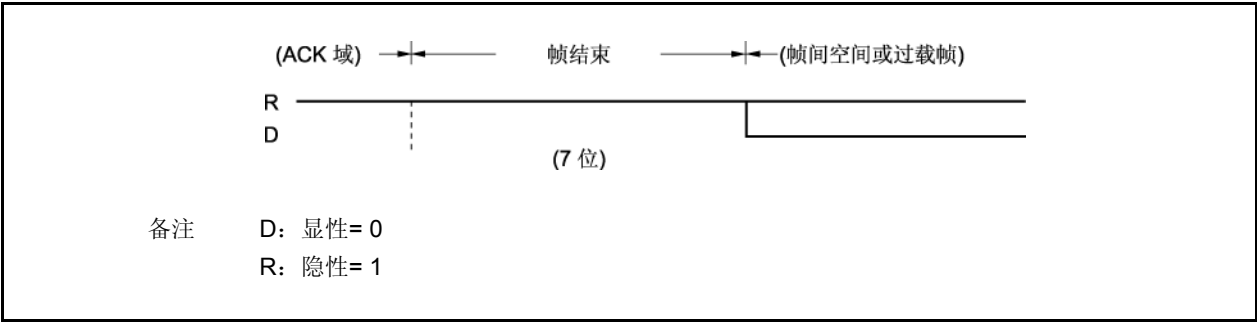


- 如果没有检测到 CRC 错误，接收节点设定 ACK 为显性电平。
- 发送节点输出两个隐性电平位。

<7> 帧结束 (EOF)

帧结束区为数据帧 / 远程帧的结束。

图 19-12. 帧结束 (EOF)



<8> 帧间空间

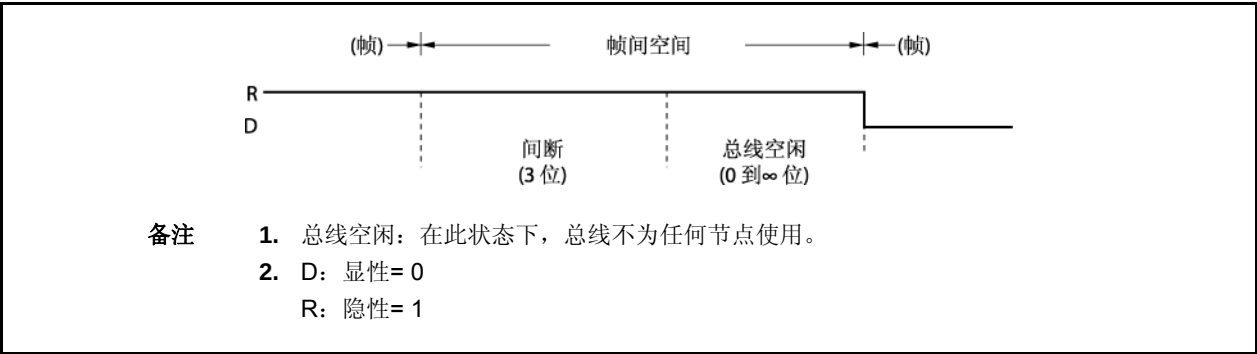
在数据帧、远程帧、错误帧或过载帧之后插入帧间空间来分隔相邻的两帧。

- 根据错误状态，总线状态不同。

(a) 错误有效节点

帧间空间包括 3 位间断区和一个总线空闲区。

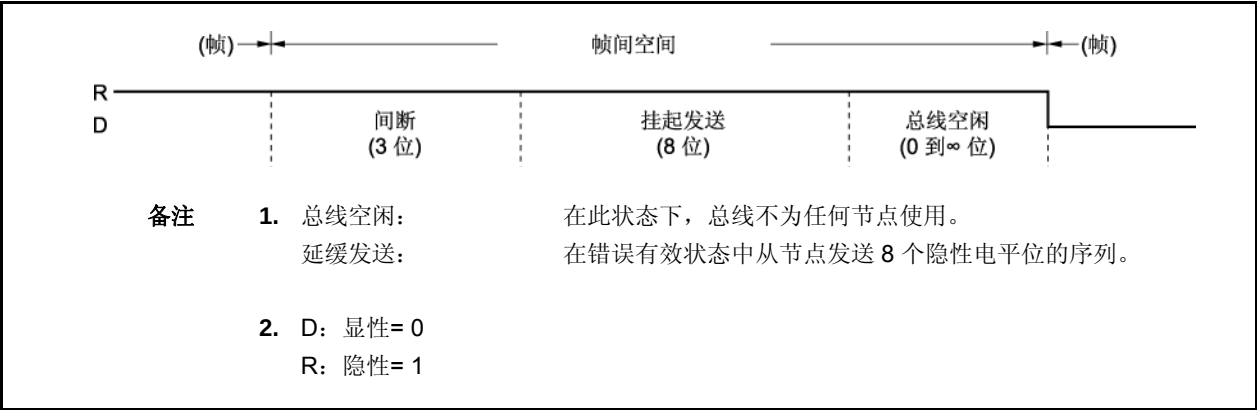
图 19-13. 帧间空间（错误有效节点）



(b) 错误无效节点

帧间空间包含间断区、发送挂起区和总线空闲区。

图 19-14. 帧间空间（错误无效节点）。



通常，间断区是 3 个位。但如果发送点在间断区的第 3 位检测到显性电平，那么它将执行发送。

- 错误状态下的操作

表 19-6. 错误状态下的操作

错误状态	操作
错误有效	在一个 3 位间断后，此状态下的节点可以立即发送。
错误无效	此状态下的节点在帧间空间后，还要再经过 8 个数据位才可以开始发送。

19.2.4 错误帧

检测出错误的节点输出错误帧。

图 19-15. 错误帧

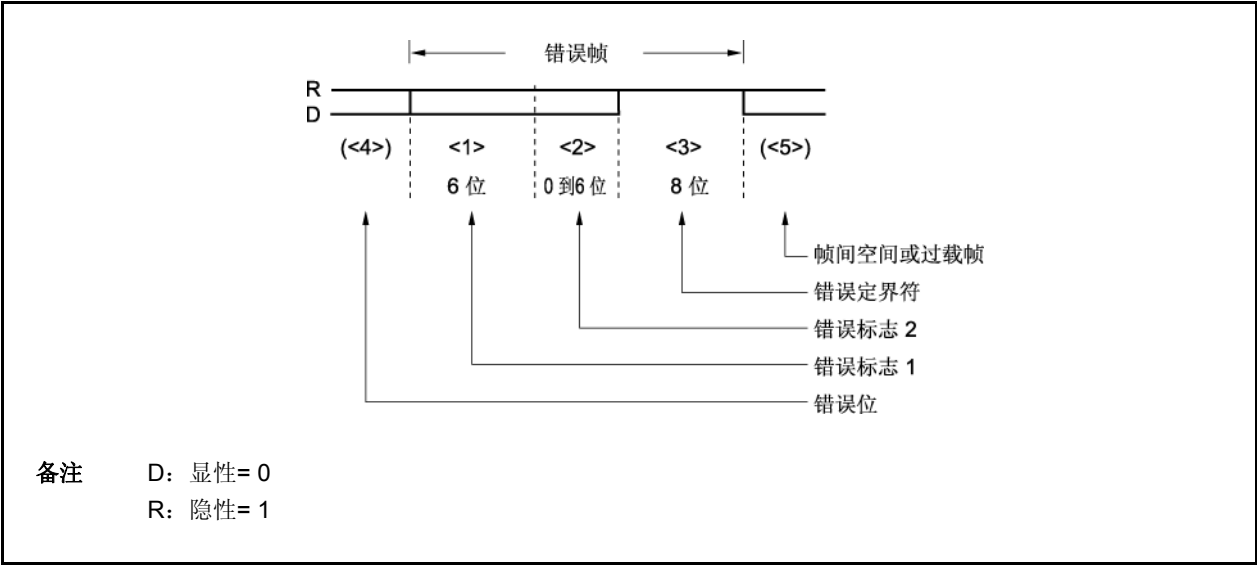


表 19-7. 错误帧区的定义

编号.	名称	位数	定义
<1>	错误标志 1	6	错误有效节点: 连续输出 6 个显性电平位。 错误无效节点: 连续输出 6 个隐性电平位。 一个节点输出隐性错误标志时, 如果其它节点输出一个显性电平, 则直到连续检测到 6 位相同的电平时无效错误标志才会清除。
<2>	错误标志 2	0 到 6	节点接收错误标志 1 检测位填充错误并发出这个错误标志。
<3>	错误分隔符	8	连续输出 8 个隐性电平位。 如果在第 8 位检测到一个显性电平, 则在下一位将发送一个过载帧。
<4>	错误位	-	被检测到错误的位。 紧随错误位的下一位输出错误标志。 在 CRC 错误时, 这一位跟随 ACK 分隔符输出。
<5>	帧间空间 / 过载帧	-	一个帧间空间或过载帧从这里开始。

19.2.5 过载帧

在以下条件下发送过载帧。

- 当接收节点没有完成接收操作^注。
- 间断时期，如果在启始两位检测到显性电平。
- 当显性电平出现在传送帧的最后一位（第 7 位）或错误界定符 / 过载符的最后一位（第 8 位）时。

注 在此 CAN 控制器中，因为高速内部处理，所有的接收帧都能被加载而不会产生过载帧。

图 19-16. 过载帧

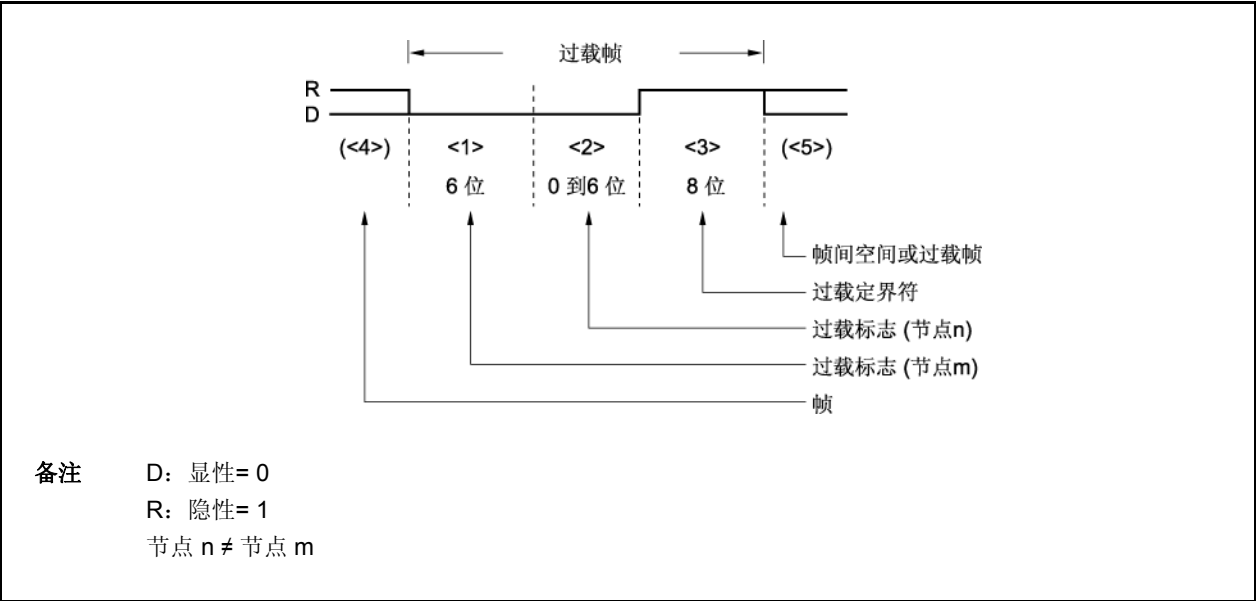


表 19-8. 过载帧区的定义

编号	名称	位数	定义
<1>	过载标志	6	连续输出 6 个显性电平位。
<2>	从其它节点来的过载标志	0 到 6	在帧间空间接收到过载标志的节点输出一个过载标志。
<3>	过载分隔符	8	连续输出 8 个隐性电平位。 如果在第 8 位检测到一个显性电平，则在下一位将发送一个过载帧。
<4>	帧	—	在帧结束、错误分隔符或过载分隔符后输出。
<5>	帧间空间 / 过载帧	—	一个帧间空间或过载帧从这里开始。

19.3 功能

19.3.1 决定总线优先级

(1) 当一个节点开始发送:

- 在总线空闲时，输出数据的节点首先开始发送数据。

(2) 当多于一个节点开始发送时:

- 从仲裁域的起始位连续输出显性电平最长的节点具有更高的总线优先级（如果一个显性电平和隐性电平同时发送，显性电平将占据总线）。
- 发送节点比较它输出的仲裁域和总线上的数据电平。

表 19-9. 总线优先级的决定

电平相同	继续发送
电平不同	从检出电平不一致的下一位数据开始停止发送，并转为接收模式

(3) 数据帧和远程帧的优先级

- 当数据帧和远程帧在总线上时，由于 RTR 位，即仲裁域的最后一位，装载显性电平，所以数据帧具有较高的优先级。

备注 如果扩展格式数据帧和标准格式远程帧在总线上相冲突（如果它们的 ID28 ~ ID18 位都是相同的），标准格式的远程帧取得优先。

19.3.2 位填充

为了避免区间误差，如果相同的电平连续出现 5 位，则通过添加一个相反的电平数据位，实现数据填充来建立同步机制。

表 19-10. 位填充

发送	在发送数据帧或远程帧时，当在帧开始和 ACK 域之间数据有 5 个相同的连续位时，在下一个位之前插入一个数据的反转电平位。
接受	在接收数据帧或远程帧时，当在帧开始和 ACK 域之间数据有 5 个相同的连续位时，删除下一位后连续接收。

19.3.3 多主控制

因为总线优先级（某节点获得发送权限）是由标识符决定的，所以任何节点都可能控制总线。

19.3.4 多点传送

虽然只有一个发送节点，因为可以在不同的节点设置相同的标识符，所以其它的节点可能在同时接收同一数据。

19.3.5 CAN SLEEP 模式 / CAN STOP 模式功能

CAN SLEEP 模式 / CAN STOP 模式功能使 CAN 控制器进入等待模式，以实现低功耗。

总线动作可以将控制器从 CAN SLEEP 模式下唤醒，却不能将其从 CAN STOP 下唤醒（由 CPU 访问来控制 CAN STOP 模式）。

19.3.6 错误控制功能

（1）错误类型

表 19-11. 错误类型

类型	错误描述		检测状态	
	检测方式	检测条件	发送 / 接收	域 / 帧
位错误	比较输出电平和总线电平。	电平不同	发送 / 接收节点	在帧开始到帧结束、错误帧和过载帧发送过程中向总线输出的数据位。
填充错误	验证接收数据的填充位。	连续 6 位的相同输出电平	接收节点	帧开始到 CRC 序列
CRC 错误	比较接收数据产生的 CRC 序列和接收的 CRC 序列。	CRC 的不同	接收节点	CRC 域
格式错误	固定格式的区域 / 帧验证。	违反固定格式的检测	接收节点	CRC 分隔符 ACK 区 帧结束 错误帧 过载帧
ACK 错误	通过发送节点验证 ACK 位。	检测 ACK 的	发送节点	ACK 位

（2）错误帧的输出时序

表 19-12. 错误帧的输出时序

类型	输出时序
位错误，填充错误，格式错误，ACK 错误	在检测到错误的下一位开始输出错误帧。
CEC 错误	在检测到 ACK 分隔符位错误的下一位开始输出错误帧。

（3）错误处理

在错误帧后发送节点重新发送数据帧或远程帧。（但是，在单发模式下不会重新发送这些帧。）

(4) 错误状态

(a) 错误状态的类型

通过 CAN 规范定义如下 3 种错误状态。

- 错误有效
- 错误无效
- Bus-off

错误状态的类型通过 C0ERC.TEC7 ~ C0ERC.TEC0 位（发送错误计数器位）的值和 C0ERC.REC6 ~ C0ERC.REC0 位（接收错误计数器为位）的值来分类，如下表 19-13。

通过 C0INFO 寄存器指示当前的错误状态。

当每一个错误计数器的值变为大于或等于错误警告等级（96）时，C0INFO.TECS0 或 C0INFO.RECS0 位设置为 1。在这种情况下，因为被认为总线有严重的故障，所以必须测试总线状态。错误计数器的值若超过 128 则表明进入错误无效状态，同时会将 TECS1 或 RECS1 置 1。

- 如果发送错误计数器的值大于或者等于 256（实际上，发送错误计数器不能表示大于或等于 256 的数值），达到 bus-off 状态并且设置 C0INFO.BOFF 位为 1。
- 如果启动时总线上只有一个节点是有效的（也就是，当总线仅仅连接到本地站点），即使数据发送了也不返回 ACK。因此，重发错误帧和数据。在错误有效状态，无论如何，发送错误计数器都不增加，不会达到 bus-off 状态。

表 19-13. 错误状态类型

类型	操作	错误计数器的值	C0INFO 寄存器的值	错误状态的操作细节
错误有效	发送	0 到 95	TECS1, TECS0 = 00	在错误检测中，输出一个有效错误标志（6 个连续的显性电平位）。
	接受	0 到 95	RECS1, RECS0 = 00	
	发送	96 到 127	TECS1, TECS0 = 01	
	接受	96 到 127	RECS1, RECS0 = 01	
错误无效	发送	128 到 255	TECS1, TECS0 = 11	- 在错误检测中，输出一个无效错误标志（6 个连续的隐性电平位）。 - 在发送中，在一个间歇后，发送 8 个隐性电平位（延缓发送）。
	接受	128 或更多	RECS1, RECS0 = 11	
Bus-off	发送	256 或更多（不显示） [※]	BOFF = 1, TECS1, TECS0 = 11	- 不能通信。 当接收到帧时，不存储信息，并且执行下面的操作。 <1> TSOUT 锁闭。 <2> REC 增加 / 减少。 <3> REC 增加 / 减少。 - 如果设定了初始模式，在请求发送除了初始模式外的操作模式后，发送 128 次 11 个隐性位，然后错误计数器重置为 0，并可以保存有效错误状态。

注 如果已经设定 BOFF，则发送错误计数器（TEG）的值没有任何意义。当计数器的值在 248 ~ 255 的范围内出现时，发送错误计数器的值以 8 为基数增加，总线进入 bus-off 状态而计数器的计数值并不增加。

(b) 错误计数器

<R>

当发生错误时，错误计数器计数累加，发送和接收正确完成时计数器递减。错误计数器在错误检测后立即向上计数。

表 19-14. 错误计数器

状态	发送错误计数器 (TEC7 ~ TEC0 位)	接收错误计数器 (REC6 ~ REC0 位)
接收节点检测一个错误（除有效错误标志或过载标志中的位错误）	无改变	+1 (REPS 位 = 0)
接收节点在错误标志或者错误帧之后检测到显性电平。	无改变	+8 (REPS 位 = 0)
发送节点发送一个错误标志。 [例外，下例中错误计数器不改变。 <1> 错误无效状态下检测到 ACK 错误，并且当输出被动错误标志时没有检测到显性电平。 <2> 在仲裁域检测到填充错误时，即发送一个隐性电平作为填充位但是检测到一个显性电平。	+8	无改变
当输出有效错误标志或过载标志时，位错误检测（有效错误发送节点）	+8	无改变
当输出有效错误标志或过载标志时，位错误检测（有效错误接收节点）	无改变	+8 (REPS 位 = 0)
当节点从有效错误标志或者过载标志的开始检测到 14 个连续的显性位并在其后检测到 8 个连续显性位时。 当节点在无效错误标志后检测到 8 个连续显性电平时。	+8 (发送)	+8 (接受, REPS 位 = 0)
当发送点完成了没有错误的发送（±0 如果错误计数器 = 0）时	-1	无改变
当接收点完成了没有错误的接收时	无改变	• -1 ($1 \leq -1$ ($1 \leq \text{REC6} \sim \text{REC0} \leq 127$, REPS 位 = 0 • ±0 ($\text{REC6} \sim \text{REC0} = 0$, REPS 位 = 0) • 119 ~ 127 之间的任何设置值 (REPS 位 = 1)

(c) 帧间空间下发生的位错误

产生一个过载帧。

注意事项 发生一个错误时，错误控制是根据错误发生前的发送错误计数器和接收错误计数器的内容进行的。在输出错误标志后，错误计数器的计数值才会递增。

(5) 从 bus-off 状态恢复

当 CAN 模块在 bus-off 状态时，发送引脚（CTXD0）与 CAN 总线分离总是输出隐性电平。

CAN 模块按照以下 bus-off 恢复顺序从 bus-off 状态下恢复。

- <1> 请求进入 CAN 初始模式
- <2> 请求进入 CAN 运行模式
 - (a) 通过正常恢复顺序恢复操作
 - (b) 跳过正常恢复顺序强制恢复操作

(a) 通过正常恢复顺序从 bus-off 状态恢复

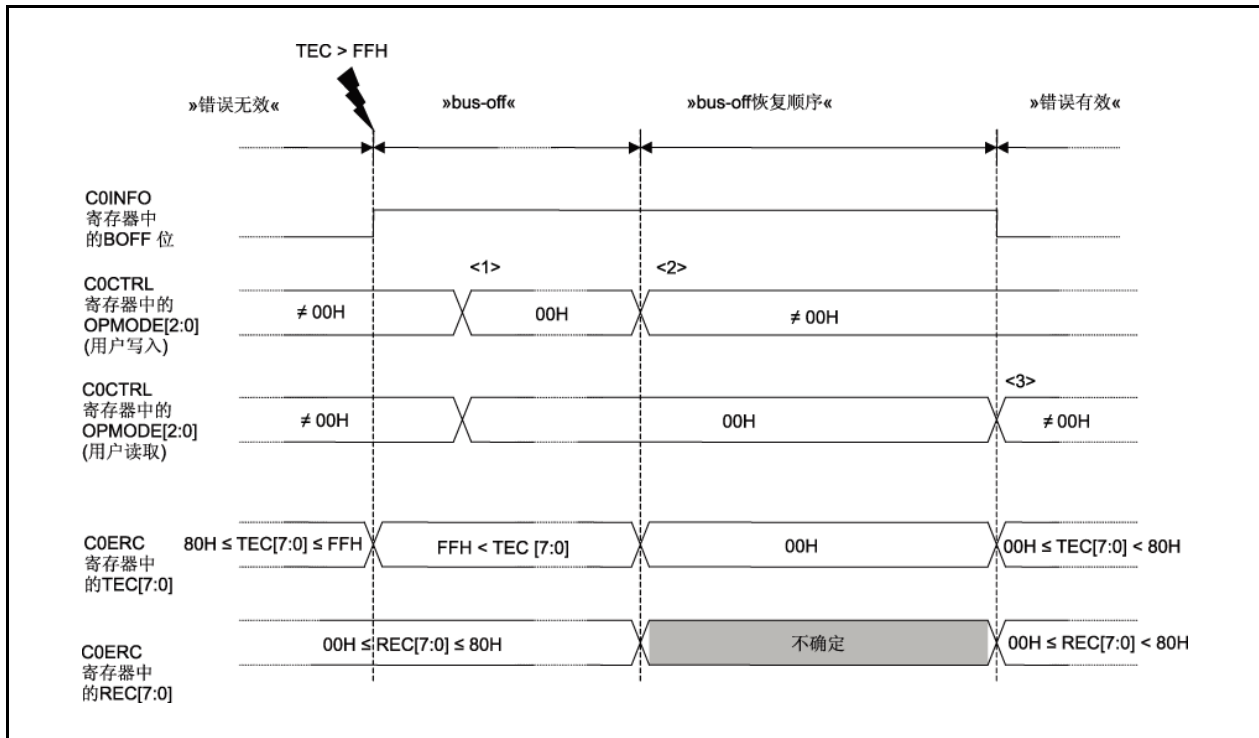
CAN 模块首先发出一个进入初始模式的请求（参看图 19-17 中的时序<1>）。这个请求将会立即被回应，并且 C0CTRL.OPMODE2 到 OPMODE0 位被清零。通过使用应用软件将 C0GMCTRL.GOM 位清零的方法，分析导致 bus-off 状态的故障、重新定义 CAN 模块及缓冲器，或者停止 CAN 模块。

然后，模块请求从初始模式改变为运行模式（请参看图 19-17 的时序<2>）。此时开始一个操作使 CAN 模块从 bus-off 状态恢复。使 CAN 模块从 bus-off 状态恢复的条件是 CAN 协议 ISO11898 定义的，并且需要检测到超过 128 次的 11 个连续隐性电平。这时，挂起改变模式到运行模式的请求，直到满足恢复条件。当满足恢复条件时（参看图 19-17 时序<3>），CAN 模块可以按照请求进入运行模式。直到进入 CAN 模块进入操作模式前，CAN 模块一直处在初始模式。通过读取 OPMODE，可以确认 CAN 模块是否完成转换到任意其它操作模式。在转换到任意其它模式完成之前，可以读到 OPMODE2 到 OPMODE0 位 = 000B。

在 bus-off 时期和 bus-off 恢复顺序中，C0INFO.BOFF 位保持为 1。在 bus-off 恢复顺序中，接收错误计数器（C0ERC.REC0 ~ C0ERC.REC6）记录总线上连续检测到 11 个连续隐性电平的次数。因此，可以通过读取 REC0 ~ REC6 位来验证恢复状态。

- <R> **注意事项**
1. 如果在 bus-off 恢复顺序期间，出现从初始化模式更改为其它任意操作模式的请求，bus-off 恢复顺序从开始启动，并再次开始对总线上 11 个连续隐性位的 128 次计数。
 2. 在 bus-off 恢复顺序中，每检测到连续的 11 次隐性电平，REC0 ~ REC6 位递增（+1）。即使在 bus-off 时期，CAN 模块也可以进入 CAN 休眠模式或者 CAN 停止模式。为了从 bus-off 状态下释放，模块必须进入初始模式一次。如果模块在 CAN 休眠模式或者 CAN 停止模式中，则它不能直接进入初始模式。在这种情况下，即使没有切换到初始模式，在释放 CAN 休眠模式的同时，开始顺序执行 bus-off 恢复顺序。另外，软件清除 C0CTRL.PSMODE1 和 C0CTRL.PSMODE0，在总线上，通过显性边沿检测来唤醒顺序开始 bus-off 恢复顺序（当提供 CAN 时钟时，在检测到显性边沿之后，C0CTRL.PSMODE0 位必须由软件清零。）。

图 19-17. 通过正常恢复处理从 bus-off 状态中恢复



(b) 跳过 bus-off 恢复顺序的强制恢复操作

可以跳过 bus-off 恢复顺序，不管总线状态，强行从 bus-off 状态释放 CAN 模块。过程如下。

首先，CAN 模块请求进入初始模式。在此时的操作和注意的点，请看 19.3.6 (5) (a) 通过正常恢复顺序从 bus-off 状态恢复。

然后，模块请求进入操作模式。同时，C0CTRL.CCERC 位必须设置为 1。

结果，CAN 协议 ISO11898 定义的 bus-off 恢复序列这一项被跳过，并且此模块立即进入操作模式。这样，模块在它连续监测到 11 个隐性电平位后连接到 CAN 总线上。详细信息，请参阅图 19-54 中的处理。

注意事项 此功能不是 CAN 协议 ISO11898 中定义的。当使用此功能时，应完全评估对网络系统的影响。

(6) 在初始化模式中初始化 CAN 模块错误计数器寄存器 (C0ERC)

要调试或评测一个程序，如果必要初始化 C0ERC 和 C0INFO 寄存器，可以在初始模式中通过设置 C0CTRL.CCERC 位来初始化 C0ERC 和 C0INFO，使它们返回默认值。当初始化完成后，CCERC 位自动清零。

注意事项 1. 此功能只在初始化模式中开启。即使在 CAN 运行模式下 CCERC 位设置为 1，C0ERC 和 C0INFO 寄存器也不能被初始化。

2. 可在请求进入 CAN 操作模式的同时，设置 CCERC 位。

19.3.7 波特率控制功

(1) 预分频

CAN 控制器可以预分频，将时钟（f_{CAN}）分频，以提供给 CAN。预分频器产生一个 CAN 协议层基本时钟（f_{rq}）— CAN 模块系统时钟（f_{CANMOD}）从 1 到 256 分频（参看 19.6 (12) CAN0 模块位率预分频寄存器（C0BRP））。

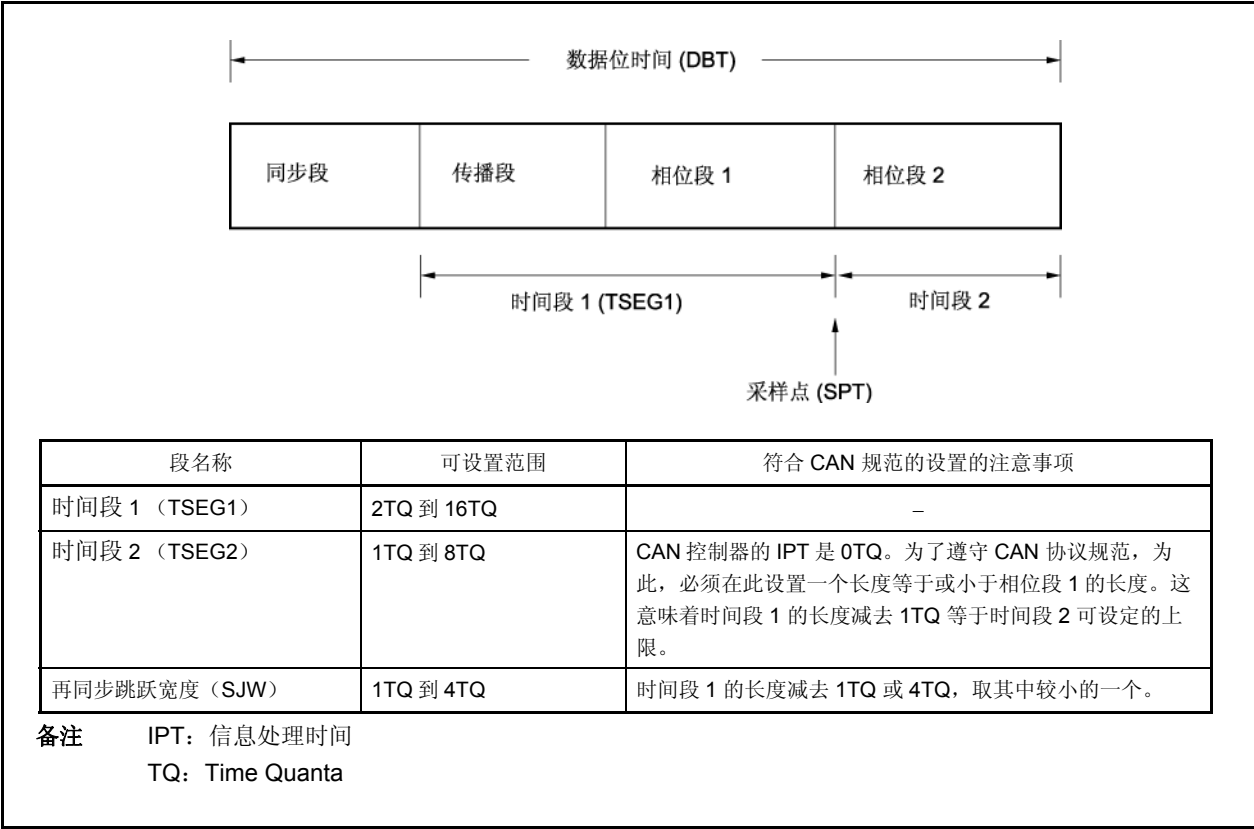
(2) 数据位时间（8 ~ 25 时间量）

如下图 19-18 所示定义单数据位时间。

1 Time Quanta = 1/f_{rq}

CAN 通过用位时间参数，例如时间段 1、时间段 2 和再同步跳跃宽度（SJW），替换来设置数据位时间，如图 19-18 所示。时间段 1 等于 CAN 协议规范定义的传播（prop）段和相位段 1 的总和。时间段 2 等于相位段 2。

图 19-18. 段设置



备注 CAN 协议规范定义了图 19-19 所示的段决定的数据位时间。

图 19-19. CAN 规范定义的数据位时间配置



(3) 同步数据位

- 因为没有同步信号，接收节点需要使用总线的电平转换来建立同步。
- 发送节点在发送节点的位时间下同步发送数据。

(a) 硬件同步

在帧间空间中，当接收节点检测到帧开始时，建立同步。

- 当在总线上检测到下降沿，当前 TQ 即为同步段，下一个段为传播段。如此，不管 SJW 是怎样都建立同步。

图 19-20. 总线空闲期间检测显性电平的硬件同步

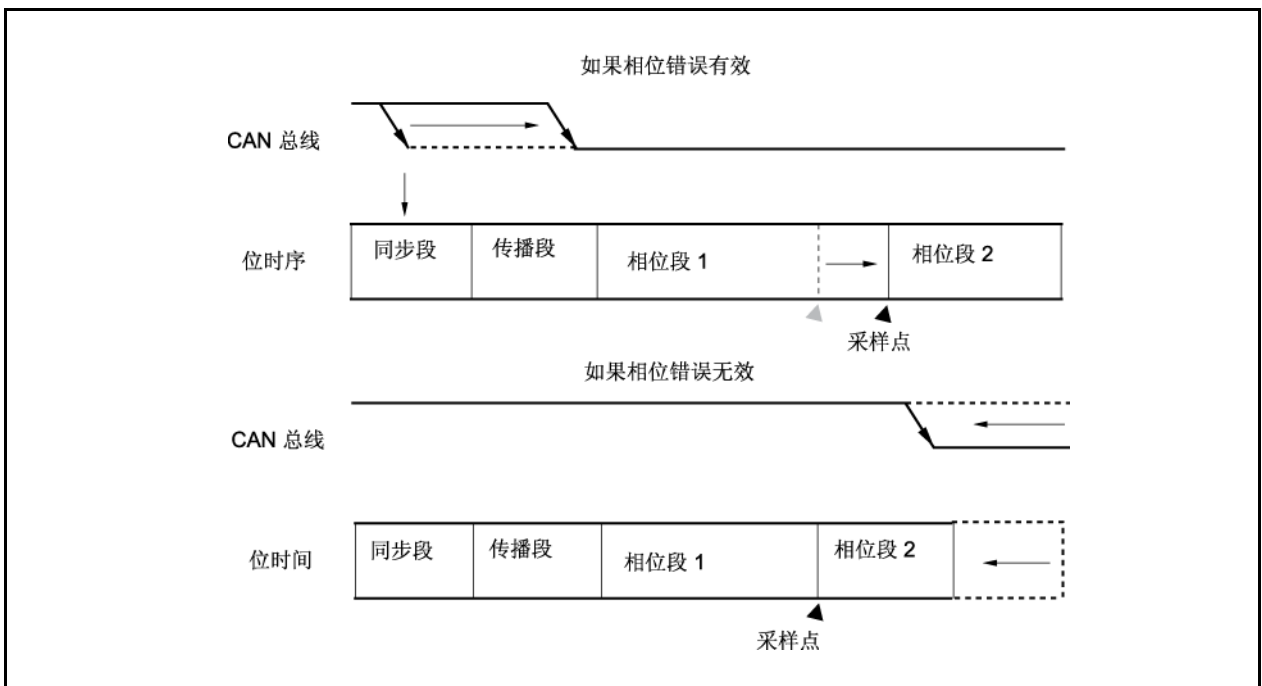


(b) 再同步

在接收中如果检测到电平的改变，则将再次建立同步（仅在采样到前一电平为隐性电平时）。

- 通过被检测的边沿和同步段的相关位置给出边沿的相位错误。
 - <相位错误的符号>
 - 0: 如果边沿在同步段内
 - 正数: 如果是取样点前的边沿（相位错误）
 - 负数: 如果是取样点后的边沿（相位错误）
 - 如果相位错误是正数: 通过指定 SJW, 相位段 1 变长。
 - 如果相位错误是负数: 通过指定 SJW, 相位段 2 变长
- 接收方的采样点会根据发送节点和接收节点间的波特率差异而产生偏移。

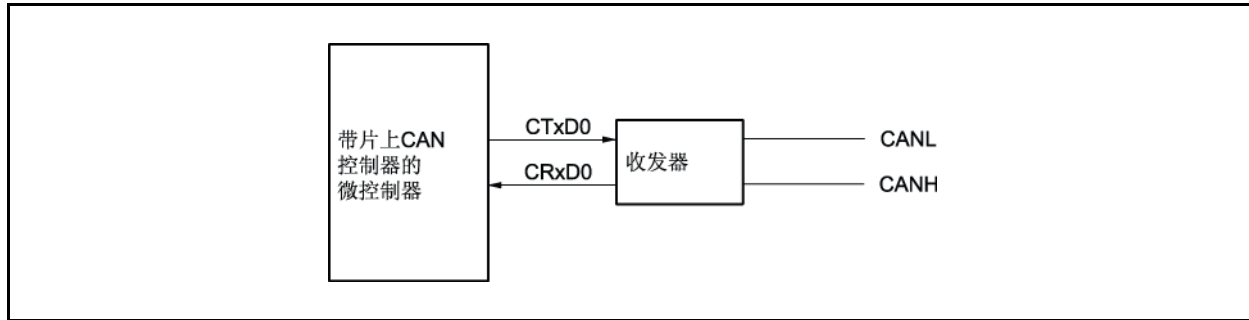
图 19-21. 再同步



19.4 与目标系统连接

带有 CAN 模块的微控制器必须通过使用外收发器连接到 CAN 总线。

图 19-22. 连接到 CAN 总线



19.5 CAN 控制器的内部寄存器

19.5.1 CAN 控制器的配置

表 19-15. CAN 控制器寄存器列表

项目	寄存器名称
CAN 全局寄存器	CAN0 全局控制寄存器 (C0GMCTRL)
	CAN0 全局时钟选择寄存器
	CAN0 全局自动 block 发送控制寄存器 (C0GMABT)
	CAN0 全局自动 block 发送延迟设置寄存器 (C0GMABTD)
CAN 模块寄存器	CAN0 模块屏蔽 1 寄存器 (C0MASK1L, C0MASK1H)
	CAN0 模块屏蔽 2 寄存器 (C0MASK2L, C0MASK2H)
	CAN0 模块屏蔽 3 寄存器 (C0MASK3L, C0MASK3H)
	CAN0 模块屏蔽 4 寄存器 (C0MASK4L, C0MASK4H)
	CAN0 模块控制寄存器 (C0CTRL)
	CAN0 模块最后错误信息寄存器 (C0LEC)
	CAN0 模块信息寄存器 (C0INFO)
	CAN0 模块错误计数器寄存器 (C0ERC)
	CAN0 模块中断开启寄存器 (C0IE)
	CAN0 模块中断状态寄存器 (C0INTS)
	CAN0 模块位率预分频寄存器 (C0BRP)
	CAN0 模块位率寄存器 (C0BTR)
	CAN0 模块最后发送指针寄存器 (C0LIPT)
	CAN0 模块接收历史列表寄存器 (C0RGPT)
	CAN0 模块最后发送指针寄存器 (C0LOPT)
	CAN0 模块发送历史列表寄存器 (C0TGPT)
	CAN0 模块时间标记寄存器 (C0TS)
信息缓冲器寄存器	CAN0 信息数据字节 01 寄存器 m (C0MDATA01m)
	CAN0 信息数据字节 0 寄存器 m (C0MDATA0m)
	CAN0 信息数据字节 1 寄存器 m (C0MDATA1m)
	CAN0 信息数据字节 23 寄存器 m (C0MDATA23m)
	CAN0 信息数据字节 2 寄存器 m (C0MDATA2m)
	CAN0 信息数据字节 3 寄存器 m (C0MDATA3m)
	CAN0 信息数据字节 45 寄存器 m (C0MDATA45m)
	CAN0 信息数据字节 4 寄存器 m (C0MDATA4m)
	CAN0 信息数据字节 5 寄存器 m (C0MDATA5m)
	CAN0 信息数据字节 67 寄存器 m (C0MDATA67m)
	CAN0 信息数据字节 6 寄存器 m (C0MDATA6m)
	CAN0 信息数据字节 7 寄存器 m (C0MDATA7m)
	CAN0 信息数据长度寄存器 m (C0MDLCm)
	CAN0 信息配置寄存器 m (C0MCONFm)
	CAN0 信息 ID 寄存器 m (C0MIDLm, C0MIDHm)
	CAN0 信息控制寄存器 m (C0MCTRLm)

- 备注**
1. CAN 全局寄存器被定义为 C0GM <寄存器功能>。
CAN 模块寄存器被定义为 C0 <寄存器功能>。
信息缓冲器寄存器被定义为 C0M <寄存器功能>。
 2. m = 00 到 31

19.5.2 寄存器访问类型

表 19-16. 寄存器访问类型 (1/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC000H	CAN0 全局控制寄存器	C0GMCTRL	R/W			√	0000H
03FEC002H	CAN0 全局时钟选择寄存器	C0GMCS			√		0FH
03FEC006H	CAN0 全局自动 block 发送寄存器	C0GMABT				√	0000H
03FEC008H	CAN0 全局自动 block 发送延迟寄存器	C0GMABTD			√		00H
03FEC040H	CAN0 模块屏蔽 1 寄存器	C0MASK1L				√	不确定
03FEC042H		C0MASK1H				√	不确定
03FEC044H	CAN0 模块屏蔽 2 寄存器	C0MASK2L				√	不确定
03FEC046H		C0MASK2H				√	不确定
03FEC048H	CAN0 模块屏蔽 3 寄存器	C0MASK3L				√	不确定
03FEC04AH		C0MASK3H				√	不确定
03FEC04CH	CAN0 模块屏蔽 4 寄存器	C0MASK4L				√	不确定
03FEC04EH		C0MASK4H				√	不确定
03FEC050H	CAN0 模块控制寄存器	C0CTRL	R			√	0000H
03FEC052H	CAN0 模块最后错误码寄存器	C0LEC			√		00H
03FEC053H	CAN0 模块信息寄存器	C0INFO	R		√		00H
03FEC054H	CAN0 模块错误计数器寄存器	C0ERC				√	0000H
03FEC056H	CAN0 模块中断开启寄存器	C0IE	R/W			√	0000H
03FEC058H	CAN0 模块中断状态寄存器	C0INTS				√	0000H
03FEC05AH	CAN0 模块位率预分频寄存器	C0BRP			√		FFH
03FEC05CH	CAN0 模块位率寄存器	C0BTR				√	370FH
03FEC05EH	CAN0 模块最后发送指针寄存器	C0LIPT	R		√		不确定
03FEC060H	CAN0 模块接收历史列表寄存器	C0RGPT	R/W			√	xx02H
03FEC062H	CAN0 模块最后发送指针寄存器	C0LOPT	R		√		不确定
03FEC064H	CAN0 模块发送历史列表寄存器	C0TGPT	R/W			√	xx02H
03FEC066H	CAN0 模块时间标记寄存器	C0TS				√	0000H

表 19-16. 寄存器访问类型 (2/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC100H	CAN0 信息数据字节 01 寄存器 00	C0MDATA0100	R/W			√	不确定
03FEC100H	CAN0 信息数据字节 0 寄存器 00	C0MDATA000			√		不确定
03FEC101H	CAN0 信息数据字节 1 寄存器 00	C0MDATA100			√		不确定
03FEC102H	CAN0 信息数据字节 23 寄存器 00	C0MDATA2300				√	不确定
03FEC102H	CAN0 信息数据字节 2 寄存器 00	C0MDATA200			√		不确定
03FEC103H	CAN0 信息数据字节 3 寄存器 00	C0MDATA300			√		不确定
03FEC104H	CAN0 信息数据字节 45 寄存器 00	C0MDATA4500				√	不确定
03FEC104H	CAN0 信息数据字节 4 寄存器 00	C0MDATA400			√		不确定
03FEC105H	CAN0 信息数据字节 5 寄存器 00	C0MDATA500			√		不确定
03FEC106H	CAN0 信息数据字节 67 寄存器 00	C0MDATA6700				√	不确定
03FEC106H	CAN0 信息数据字节 6 寄存器 00	C0MDATA600			√		不确定
03FEC107H	CAN0 信息数据字节 7 寄存器 00	C0MDATA700			√		不确定
03FEC108H	CAN0 信息数据长度寄存器 00	C0MDLC00			√		0000xxxxB
03FEC109H	CAN0 信息配置寄存器 00	C0MCONF00			√		不确定
03FEC10AH	CAN0 信息标识符寄存器 00	C0MIDL00				√	不确定
03FEC10CH		C0MIDH00				√	不确定
03FEC10EH	CAN0 信息控制寄存器 00	C0MCTRL00				√	00x00000 000xx000B
03FEC120H	CAN0 信息数据字节 01 寄存器 01	C0MDATA0101				√	不确定
03FEC120H	CAN0 信息数据字节 0 寄存器 01	C0MDATA001			√		不确定
03FEC121H	CAN0 信息数据字节 1 寄存器 01	C0MDATA101			√		不确定
03FEC122H	CAN0 信息数据字节 23 寄存器 01	C0MDATA2301				√	不确定
03FEC122H	CAN0 信息数据字节 2 寄存器 01	C0MDATA201			√		不确定
03FEC123H	CAN0 信息数据字节 3 寄存器 01	C0MDATA301			√		不确定
03FEC124H	CAN0 信息数据字节 45 寄存器 01	C0MDATA4501				√	不确定
03FEC124H	CAN0 信息数据字节 4 寄存器 01	C0MDATA401			√		不确定
03FEC125H	CAN0 信息数据字节 5 寄存器 01	C0MDATA501			√		不确定
03FEC126H	CAN0 信息数据字节 67 寄存器 01	C0MDATA6701				√	不确定
03FEC126H	CAN0 信息数据字节 6 寄存器 01	C0MDATA601			√		不确定
03FEC127H	CAN0 信息数据字节 7 寄存器 01	C0MDATA701			√		不确定
03FEC128H	CAN0 信息数据长度寄存器 01	C0MDLC01			√		0000xxxxB
03FEC129H	CAN0 信息配置寄存器 01	C0MCONF01			√		不确定
03FEC12AH	CAN0 信息标识符寄存器 01	C0MIDL01				√	不确定
03FEC12CH		C0MIDH01				√	不确定
03FEC12EH	CAN0 信息控制寄存器 01	C0MCTRL01				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (3/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC140H	CAN0 信息数据 字节 01 寄存器 02	C0MDATA0102	R/W			√	不确定
03FEC140H	CAN0 信息数据 字节 0 寄存器 02	C0MDATA002			√		不确定
03FEC141H	CAN0 信息数据 字节 1 寄存器 02	C0MDATA102			√		不确定
03FEC142H	CAN0 信息数据 字节 23 寄存器 02	C0MDATA2302				√	不确定
03FEC142H	CAN0 信息数据 字节 2 寄存器 02	C0MDATA202			√		不确定
03FEC143H	CAN0 信息数据 字节 3 寄存器 02	C0MDATA302			√		不确定
03FEC144H	CAN0 信息数据 字节 45 寄存器 02	C0MDATA4502				√	不确定
03FEC144H	CAN0 信息数据 字节 4 寄存器 02	C0MDATA402			√		不确定
03FEC145H	CAN0 信息数据 字节 5 寄存器 02	C0MDATA502			√		不确定
03FEC146H	CAN0 信息数据 字节 67 寄存器 02	C0MDATA6702				√	不确定
03FEC146H	CAN0 信息数据 字节 6 寄存器 02	C0MDATA602			√		不确定
03FEC147H	CAN0 信息数据 字节 7 寄存器 02	C0MDATA702			√		不确定
03FEC148H	CAN0 信息数据长度寄存器 02	C0MDLC02			√		0000xxxxB
03FEC149H	CAN0 信息配置寄存器 02	C0MCONF02			√		不确定
03FEC14AH	CAN0 信息标识符寄存器 02	C0MIDL02				√	不确定
03FEC14CH		C0MIDH02				√	不确定
03FEC14EH	CAN0 信息控制寄存器 02	C0MCTRL02				√	00x00000 000xx000B
03FEC160H	CAN0 信息数据字节 01 寄存器 03	C0MDATA0103				√	不确定
03FEC160H	CAN0 信息数据字节 0 寄存器 03	C0MDATA003			√		不确定
03FEC161H	CAN0 信息数据字节 1 寄存器 03	C0MDATA103			√		不确定
03FEC162H	CAN0 信息数据字节 23 寄存器 03	C0MDATA2303				√	不确定
03FEC162H	CAN0 信息数据字节 2 寄存器 03	C0MDATA203			√		不确定
03FEC163H	CAN0 信息数据字节 3 寄存器 03	C0MDATA303			√		不确定
03FEC164H	CAN0 信息数据字节 45 寄存器 03	C0MDATA4503				√	不确定
03FEC164H	CAN0 信息数据字节 4 寄存器 03	C0MDATA403			√		不确定
03FEC165H	CAN0 信息数据字节 5 寄存器 03	C0MDATA503			√		不确定
03FEC166H	CAN0 信息数据字节 67 寄存器 03	C0MDATA6703				√	不确定
03FEC166H	CAN0 信息数据字节 6 寄存器 03	C0MDATA603			√		不确定
03FEC167H	CAN0 信息数据字节 7 寄存器 03	C0MDATA703			√		不确定
03FEC168H	CAN0 信息数据长度寄存器 03	C0MDLC03			√		0000xxxxB
03FEC169H	CAN0 信息配置寄存器 03	C0MCONF03			√		不确定
03FEC16AH	CAN0 信息标识符寄存器 03	C0MIDL03				√	不确定
03FEC16CH		C0MIDH03				√	不确定
03FEC16EH	CAN0 信息控制寄存器 03	C0MCTRL03				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (4/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC180H	CAN0 信息数据字节 01 寄存器 04	C0MDATA0104	R/W			√	不确定
03FEC180H	CAN0 信息数据字节 0 寄存器 04	C0MDATA004			√		不确定
03FEC181H	CAN0 信息数据字节 1 寄存器 04	C0MDATA104			√		不确定
03FEC182H	CAN0 信息数据字节 23 寄存器 04	C0MDATA2304				√	不确定
03FEC182H	CAN0 信息数据字节 2 寄存器 04	C0MDATA204			√		不确定
03FEC183H	CAN0 信息数据字节 3 寄存器 04	C0MDATA304			√		不确定
03FEC184H	CAN0 信息数据字节 45 寄存器 04	C0MDATA4504				√	不确定
03FEC184H	CAN0 信息数据字节 4 寄存器 04	C0MDATA404			√		不确定
03FEC185H	CAN0 信息数据字节 5 寄存器 04	C0MDATA504			√		不确定
03FEC186H	CAN0 信息数据字节 67 寄存器 04	C0MDATA6704				√	不确定
03FEC186H	CAN0 信息数据字节 6 寄存器 04	C0MDATA604			√		不确定
03FEC187H	CAN0 信息数据字节 7 寄存器 04	C0MDATA704			√		不确定
03FEC188H	CAN0 信息数据长度寄存器 04	C0MDLC04			√		0000xxxxB
03FEC189H	CAN0 信息配置寄存器 04	C0MCONF04			√		不确定
03FEC18AH	CAN0 信息标识符寄存器 04	C0MIDL04				√	不确定
03FEC18CH		C0MIDH04				√	不确定
03FEC18EH	CAN0 信息控制寄存器 04	C0MCTRL04				√	00x00000 000xx000B
03FEC1A0H	CAN0 信息数据字节 01 寄存器 05	C0MDATA0105				√	不确定
03FEC1A0H	CAN0 信息数据字节 0 寄存器 05	C0MDATA005			√		不确定
03FEC1A1H	CAN0 信息数据字节 1 寄存器 05	C0MDATA105			√		不确定
03FEC1A2H	CAN0 信息数据字节 23 寄存器 05	C0MDATA2305				√	不确定
03FEC1A2H	CAN0 信息数据字节 2 寄存器 05	C0MDATA205			√		不确定
03FEC1A3H	CAN0 信息数据字节 3 寄存器 05	C0MDATA305			√		不确定
03FEC1A4H	CAN0 信息数据字节 45 寄存器 05	C0MDATA4505				√	不确定
03FEC1A4H	CAN0 信息数据字节 4 寄存器 05	C0MDATA405			√		不确定
03FEC1A5H	CAN0 信息数据字节 5 寄存器 05	C0MDATA505			√		不确定
03FEC1A6H	CAN0 信息数据字节 67 寄存器 05	C0MDATA6705				√	不确定
03FEC1A6H	CAN0 信息数据字节 6 寄存器 05	C0MDATA605			√		不确定
03FEC1A7H	CAN0 信息数据字节 7 寄存器 05	C0MDATA705			√		不确定
03FEC1A8H	CAN0 信息数据长度寄存器 05	C0MDLC05			√		0000xxxxB
03FEC1A9H	CAN0 信息配置寄存器 05	C0MCONF05			√		不确定
03FEC1AAH	CAN0 信息标识符寄存器 05	C0MIDL05				√	不确定
03FEC1ACH		C0MIDH05				√	不确定
03FEC1AEH	CAN0 信息控制寄存器 05	C0MCTRL05				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (5/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC1C0H	CAN0 信息数据字节 01 寄存器 06	C0MDATA0106	R/W			√	不确定
03FEC1C0H	CAN0 信息数据字节 0 寄存器 06	C0MDATA006			√		不确定
03FEC1C1H	CAN0 信息数据字节 1 寄存器 06	C0MDATA106			√		不确定
03FEC1C2H	CAN0 信息数据字节 23 寄存器 06	C0MDATA2306				√	不确定
03FEC1C2H	CAN0 信息数据字节 2 寄存器 06	C0MDATA206			√		不确定
03FEC1C3H	CAN0 信息数据字节 3 寄存器 06	C0MDATA306			√		不确定
03FEC1C4H	CAN0 信息数据字节 45 寄存器 06	C0MDATA4506				√	不确定
03FEC1C4H	CAN0 信息数据字节 4 寄存器 06	C0MDATA406			√		不确定
03FEC1C5H	CAN0 信息数据字节 5 寄存器 06	C0MDATA506			√		不确定
03FEC1C6H	CAN0 信息数据字节 67 寄存器 06	C0MDATA6706				√	不确定
03FEC1C6H	CAN0 信息数据字节 6 寄存器 06	C0MDATA606			√		不确定
03FEC1C7H	CAN0 信息数据字节 7 寄存器 06	C0MDATA706			√		不确定
03FEC1C8H	CAN0 信息数据长度寄存器 06	C0MDLC06			√		0000xxxxB
03FEC1C9H	CAN0 信息配置寄存器 06	C0MCONF06			√		不确定
03FEC1CAH	CAN0 信息标识符寄存器 06	C0MIDL06				√	不确定
03FEC1CCH		C0MIDH06				√	不确定
03FEC1CEH	CAN0 信息控制寄存器 06	C0MCTRL06				√	00x00000 000xx000B
03FEC1E0H	CAN0 信息数据字节 01 寄存器 07	C0MDATA0107				√	不确定
03FEC1E0H	CAN0 信息数据字节 0 寄存器 07	C0MDATA007			√		不确定
03FEC1E1H	CAN0 信息数据字节 1 寄存器 07	C0MDATA107			√		不确定
03FEC1E2H	CAN0 信息数据字节 23 寄存器 07	C0MDATA2307				√	不确定
03FEC1E2H	CAN0 信息数据字节 2 寄存器 07	C0MDATA207			√		不确定
03FEC1E3H	CAN0 信息数据字节 3 寄存器 07	C0MDATA307			√		不确定
03FEC1E4H	CAN0 信息数据字节 45 寄存器 07	C0MDATA4507				√	不确定
03FEC1E4H	CAN0 信息数据字节 4 寄存器 07	C0MDATA407			√		不确定
03FEC1E5H	CAN0 信息数据字节 5 寄存器 07	C0MDATA507			√		不确定
03FEC1E6H	CAN0 信息数据字节 67 寄存器 07	C0MDATA6707				√	不确定
03FEC1E6H	CAN0 信息数据字节 6 寄存器 07	C0MDATA607			√		不确定
03FEC1E7H	CAN0 信息数据字节 7 寄存器 07	C0MDATA707			√		不确定
03FEC1E8H	CAN0 信息数据长度寄存器 07	C0MDLC07			√		0000xxxxB
03FEC1E9H	CAN0 信息配置寄存器 07	C0MCONF07			√		不确定
03FEC1EAH	CAN0 信息标识符寄存器 07	C0MIDL07				√	不确定
03FEC1ECH		C0MIDH07				√	不确定
03FEC1EEH	CAN0 信息控制寄存器 07	C0MCTRL07				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (6/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC200H	CAN0 信息数据字节 01 寄存器 08	C0MDATA0108	R/W			√	不确定
03FEC200H	CAN0 信息数据字节 0 寄存器 08	C0MDATA008			√		不确定
03FEC201H	CAN0 信息数据字节 1 寄存器 08	C0MDATA108			√		不确定
03FEC202H	CAN0 信息数据字节 23 寄存器 08	C0MDATA2308				√	不确定
03FEC202H	CAN0 信息数据字节 2 寄存器 08	C0MDATA208			√		不确定
03FEC203H	CAN0 信息数据字节 3 寄存器 08	C0MDATA308			√		不确定
03FEC204H	CAN0 信息数据字节 45 寄存器 08	C0MDATA4508				√	不确定
03FEC204H	CAN0 信息数据字节 4 寄存器 08	C0MDATA408			√		不确定
03FEC205H	CAN0 信息数据字节 5 寄存器 08	C0MDATA508			√		不确定
03FEC206H	CAN0 信息数据字节 67 寄存器 08	C0MDATA6708				√	不确定
03FEC206H	CAN0 信息数据字节 6 寄存器 08	C0MDATA608			√		不确定
03FEC207H	CAN0 信息数据字节 7 寄存器 08	C0MDATA708			√		不确定
03FEC208H	CAN0 信息数据长度寄存器 08	C0MDLC08			√		0000xxxxB
03FEC209H	CAN0 信息配置寄存器 08	C0MCONF08			√		不确定
03FEC20AH	CAN0 信息标识符寄存器 08	C0MIDL08				√	不确定
03FEC20CH		C0MIDH08				√	不确定
03FEC20EH	CAN0 信息控制寄存器 08	C0MCTRL08				√	00x00000 000xx000B
03FEC220H	CAN0 信息数据字节 01 寄存器 09	C0MDATA0109				√	不确定
03FEC220H	CAN0 信息数据字节 0 寄存器 09	C0MDATA009			√		不确定
03FEC221H	CAN0 信息数据字节 1 寄存器 09	C0MDATA109			√		不确定
03FEC222H	CAN0 信息数据字节 23 寄存器 09	C0MDATA2309				√	不确定
03FEC222H	CAN0 信息数据字节 2 寄存器 09	C0MDATA209			√		不确定
03FEC223H	CAN0 信息数据字节 3 寄存器 09	C0MDATA309			√		不确定
03FEC224H	CAN0 信息数据字节 45 寄存器 09	C0MDATA4509				√	不确定
03FEC224H	CAN0 信息数据字节 4 寄存器 09	C0MDATA409			√		不确定
03FEC225H	CAN0 信息数据字节 5 寄存器 09	C0MDATA509			√		不确定
03FEC226H	CAN0 信息数据字节 67 寄存器 09	C0MDATA6709				√	不确定
03FEC226H	CAN0 信息数据字节 6 寄存器 09	C0MDATA609			√		不确定
03FEC227H	CAN0 信息数据字节 7 寄存器 09	C0MDATA709			√		不确定
03FEC228H	CAN0 信息数据长度寄存器 09	C0MDLC09			√		0000xxxxB
03FEC229H	CAN0 信息配置寄存器 09	C0MCONF09			√		不确定
03FEC22AH	CAN0 信息标识符寄存器 09	C0MIDL09				√	不确定
03FEC22CH		C0MIDH09				√	不确定
03FEC22EH	CAN0 信息控制寄存器 09	C0MCTRL09				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (7/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC240H	CAN0 信息数据字节 01 寄存器 10	C0MDATA0110	R/W			√	不确定
03FEC240H	CAN0 信息数据字节 0 寄存器 10	C0MDATA010			√		不确定
03FEC241H	CAN0 信息数据字节 1 寄存器 10	C0MDATA110			√		不确定
03FEC242H	CAN0 信息数据字节 23 寄存器 10	C0MDATA2310				√	不确定
03FEC242H	CAN0 信息数据字节 2 寄存器 10	C0MDATA210			√		不确定
03FEC243H	CAN0 信息数据字节 3 寄存器 10	C0MDATA310			√		不确定
03FEC244H	CAN0 信息数据字节 45 寄存器 10	C0MDATA4510				√	不确定
03FEC244H	CAN0 信息数据字节 4 寄存器 10	C0MDATA410			√		不确定
03FEC245H	CAN0 信息数据字节 5 寄存器 10	C0MDATA510			√		不确定
03FEC246H	CAN0 信息数据字节 67 寄存器 10	C0MDATA6710				√	不确定
03FEC246H	CAN0 信息数据字节 6 寄存器 10	C0MDATA610			√		不确定
03FEC247H	CAN0 信息数据字节 7 寄存器 10	C0MDATA710			√		不确定
03FEC248H	CAN0 信息数据长度寄存器 10	C0MDLC10			√		0000xxxxB
03FEC249H	CAN0 信息配置寄存器 10	C0MCONF10			√		不确定
03FEC24AH	CAN0 信息标识符寄存器 10	C0MIDL10				√	不确定
03FEC24CH		C0MIDH10				√	不确定
03FEC24EH	CAN0 信息控制寄存器 10	C0MCTRL10				√	00x00000 000xx000B
03FEC260H	CAN0 信息数据字节 01 寄存器 11	C0MDATA0111				√	不确定
03FEC260H	CAN0 信息数据字节 0 寄存器 11	C0MDATA011			√		不确定
03FEC261H	CAN0 信息数据字节 1 寄存器 11	C0MDATA111			√		不确定
03FEC262H	CAN0 信息数据字节 23 寄存器 11	C0MDATA2311				√	不确定
03FEC262H	CAN0 信息数据字节 2 寄存器 11	C0MDATA211			√		不确定
03FEC263H	CAN0 信息数据字节 3 寄存器 11	C0MDATA311			√		不确定
03FEC264H	CAN0 信息数据字节 45 寄存器 11	C0MDATA4511				√	不确定
03FEC264H	CAN0 信息数据字节 4 寄存器 11	C0MDATA411			√		不确定
03FEC265H	CAN0 信息数据字节 5 寄存器 11	C0MDATA511			√		不确定
03FEC266H	CAN0 信息数据字节 67 寄存器 11	C0MDATA6711				√	不确定
03FEC266H	CAN0 信息数据字节 6 寄存器 11	C0MDATA611			√		不确定
03FEC267H	CAN0 信息数据字节 7 寄存器 11	C0MDATA711			√		不确定
03FEC268H	CAN0 信息数据长度寄存器 11	C0MDLC11			√		0000xxxxB
03FEC269H	CAN0 信息配置寄存器 11	C0MCONF11			√		不确定
03FEC26AH	CAN0 信息标识符寄存器 11	C0MIDL11				√	不确定
03FEC26CH		C0MIDH11				√	不确定
03FEC26EH	CAN0 信息控制寄存器 11	C0MCTRL11				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (8/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC280H	CAN0 信息数据字节 01 寄存器 12	C0MDATA0112	R/W			√	不确定
03FEC280H	CAN0 信息数据字节 0 寄存器 12	C0MDATA012			√		不确定
03FEC281H	CAN0 信息数据字节 1 寄存器 12	C0MDATA112			√		不确定
03FEC282H	CAN0 信息数据字节 23 寄存器 12	C0MDATA2312				√	不确定
03FEC282H	CAN0 信息数据字节 2 寄存器 12	C0MDATA212			√		不确定
03FEC283H	CAN0 信息数据字节 3 寄存器 12	C0MDATA312			√		不确定
03FEC284H	CAN0 信息数据字节 45 寄存器 12	C0MDATA4512				√	不确定
03FEC284H	CAN0 信息数据字节 4 寄存器 12	C0MDATA412			√		不确定
03FEC285H	CAN0 信息数据字节 5 寄存器 12	C0MDATA512			√		不确定
03FEC286H	CAN0 信息数据字节 67 寄存器 12	C0MDATA6712				√	不确定
03FEC286H	CAN0 信息数据字节 6 寄存器 12	C0MDATA612			√		不确定
03FEC287H	CAN0 信息数据字节 7 寄存器 12	C0MDATA712			√		不确定
03FEC288H	CAN0 信息数据长度寄存器 12	C0MDLC12			√		0000xxxxB
03FEC289H	CAN0 信息配置寄存器 12	C0MCONF12			√		不确定
03FEC28AH	CAN0 信息标识符寄存器 12	C0MIDL12				√	不确定
03FEC28CH		C0MIDH12				√	不确定
03FEC28EH	CAN0 信息控制寄存器 12	C0MCTRL12				√	00x00000 000xx000B
03FEC2A0H	CAN0 信息数据字节 01 寄存器 13	C0MDATA0113				√	不确定
03FEC2A0H	CAN0 信息数据字节 0 寄存器 13	C0MDATA013			√		不确定
03FEC2A1H	CAN0 信息数据字节 1 寄存器 13	C0MDATA113			√		不确定
03FEC2A2H	CAN0 信息数据字节 23 寄存器 13	C0MDATA2313				√	不确定
03FEC2A2H	CAN0 信息数据字节 2 寄存器 13	C0MDATA213			√		不确定
03FEC2A3H	CAN0 信息数据字节 3 寄存器 13	C0MDATA313			√		不确定
03FEC2A4H	CAN0 信息数据字节 45 寄存器 13	C0MDATA4513				√	不确定
03FEC2A4H	CAN0 信息数据字节 4 寄存器 13	C0MDATA413			√		不确定
03FEC2A5H	CAN0 信息数据字节 5 寄存器 13	C0MDATA513			√		不确定
03FEC2A6H	CAN0 信息数据字节 67 寄存器 13	C0MDATA6713				√	不确定
03FEC2A6H	CAN0 信息数据字节 6 寄存器 13	C0MDATA613			√		不确定
03FEC2A7H	CAN0 信息数据字节 7 寄存器 13	C0MDATA713			√		不确定
03FEC2A8H	CAN0 信息数据长度寄存器 13	C0MDLC13			√		0000xxxxB
03FEC2A9H	CAN0 信息配置寄存器 13	C0MCONF13			√		不确定
03FEC2AAH	CAN0 信息标识符寄存器 13	C0MIDL13				√	不确定
03FEC2ACH		C0MIDH13				√	不确定
03FEC2AEH	CAN0 信息控制寄存器 13	C0MCTRL13				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (9/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC2C0H	CAN0 信息数据字节 01 寄存器 14	C0MDATA0114	R/W			√	不确定
03FEC2C0H	CAN0 信息数据字节 0 寄存器 14	C0MDATA014			√		不确定
03FEC2C1H	CAN0 信息数据字节 1 寄存器 14	C0MDATA114			√		不确定
03FEC2C2H	CAN0 信息数据字节 23 寄存器 14	C0MDATA2314				√	不确定
03FEC2C2H	CAN0 信息数据字节 2 寄存器 14	C0MDATA214			√		不确定
03FEC2C3H	CAN0 信息数据字节 3 寄存器 14	C0MDATA314			√		不确定
03FEC2C4H	CAN0 信息数据字节 45 寄存器 14	C0MDATA4514				√	不确定
03FEC2C4H	CAN0 信息数据字节 4 寄存器 14	C0MDATA414			√		不确定
03FEC2C5H	CAN0 信息数据字节 5 寄存器 14	C0MDATA514			√		不确定
03FEC2C6H	CAN0 信息数据字节 67 寄存器 14	C0MDATA6714				√	不确定
03FEC2C6H	CAN0 信息数据字节 6 寄存器 14	C0MDATA614			√		不确定
03FEC2C7H	CAN0 信息数据字节 7 寄存器 14	C0MDATA714			√		不确定
03FEC2C8H	CAN0 信息数据长度寄存器 14	C0MDLC14			√		0000xxxxB
03FEC2C9H	CAN0 信息配置寄存器 14	C0MCONF14			√		不确定
03FEC2CAH	CAN0 信息标识符寄存器 14	C0MIDL14				√	不确定
03FEC2CCH		C0MIDH14				√	不确定
03FEC2CEH	CAN0 信息控制寄存器 14	C0MCTRL14				√	00x00000 000xx000B
03FEC2E0H	CAN0 信息数据字节 01 寄存器 15	C0MDATA0115				√	不确定
03FEC2E0H	CAN0 信息数据字节 0 寄存器 15	C0MDATA015			√		不确定
03FEC2E1H	CAN0 信息数据字节 1 寄存器 15	C0MDATA115			√		不确定
03FEC2E2H	CAN0 信息数据字节 23 寄存器 15	C0MDATA2315				√	不确定
03FEC2E2H	CAN0 信息数据字节 2 寄存器 15	C0MDATA215			√		不确定
03FEC2E3H	CAN0 信息数据字节 3 寄存器 15	C0MDATA315			√		不确定
03FEC2E4H	CAN0 信息数据字节 45 寄存器 15	C0MDATA4515				√	不确定
03FEC2E4H	CAN0 信息数据字节 4 寄存器 15	C0MDATA415			√		不确定
03FEC2E5H	CAN0 信息数据字节 5 寄存器 15	C0MDATA515			√		不确定
03FEC2E6H	CAN0 信息数据字节 67 寄存器 15	C0MDATA6715				√	不确定
03FEC2E6H	CAN0 信息数据字节 6 寄存器 15	C0MDATA615			√		不确定
03FEC2E7H	CAN0 信息数据字节 7 寄存器 15	C0MDATA715			√		不确定
03FEC2E8H	CAN0 信息数据长度寄存器 15	C0MDLC15			√		0000xxxxB
03FEC2E9H	CAN0 信息配置寄存器 15	C0MCONF15			√		不确定
03FEC2EAH	CAN0 信息标识符寄存器 15	C0MIDL15				√	不确定
03FEC2ECH		C0MIDH15				√	不确定
03FEC2EEH	CAN0 信息控制寄存器 15	C0MCTRL15				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (10/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC300H	CAN0 信息数据字节 01 寄存器 16	C0MDATA0116	R/W			√	不确定
03FEC300H	CAN0 信息数据字节 0 寄存器 16	C0MDATA016			√		不确定
03FEC301H	CAN0 信息数据字节 1 寄存器 16	C0MDATA116			√		不确定
03FEC302H	CAN0 信息数据字节 23 寄存器 16	C0MDATA2316				√	不确定
03FEC302H	CAN0 信息数据字节 2 寄存器 16	C0MDATA216			√		不确定
03FEC303H	CAN0 信息数据字节 3 寄存器 16	C0MDATA316			√		不确定
03FEC304H	CAN0 信息数据字节 45 寄存器 16	C0MDATA4516				√	不确定
03FEC304H	CAN0 信息数据字节 4 寄存器 16	C0MDATA416			√		不确定
03FEC305H	CAN0 信息数据字节 5 寄存器 16	C0MDATA516			√		不确定
03FEC306H	CAN0 信息数据字节 67 寄存器 16	C0MDATA6716				√	不确定
03FEC306H	CAN0 信息数据字节 6 寄存器 16	C0MDATA616			√		不确定
03FEC307H	CAN0 信息数据字节 7 寄存器 16	C0MDATA716			√		不确定
03FEC308H	CAN0 信息数据长度寄存器 16	C0MDLC16			√		0000xxxxB
03FEC309H	CAN0 信息配置寄存器 16	C0MCONF16			√		不确定
03FEC30AH	CAN0 信息标识符寄存器 16	C0MIDL16				√	不确定
03FEC30CH		C0MIDH16				√	不确定
03FEC30EH	CAN0 信息控制寄存器 16	C0MCTRL16				√	00x00000 000xx000B
03FEC320H	CAN0 信息数据字节 01 寄存器 17	C0MDATA0117				√	不确定
03FEC320H	CAN0 信息数据字节 0 寄存器 17	C0MDATA017			√		不确定
03FEC321H	CAN0 信息数据字节 1 寄存器 17	C0MDATA117			√		不确定
03FEC322H	CAN0 信息数据字节 23 寄存器 17	C0MDATA2317				√	不确定
03FEC322H	CAN0 信息数据字节 2 寄存器 17	C0MDATA217			√		不确定
03FEC323H	CAN0 信息数据字节 3 寄存器 17	C0MDATA317			√		不确定
03FEC324H	CAN0 信息数据字节 45 寄存器 17	C0MDATA4517				√	不确定
03FEC324H	CAN0 信息数据字节 4 寄存器 17	C0MDATA417			√		不确定
03FEC325H	CAN0 信息数据字节 5 寄存器 17	C0MDATA517			√		不确定
03FEC326H	CAN0 信息数据字节 67 寄存器 17	C0MDATA6717				√	不确定
03FEC326H	CAN0 信息数据字节 6 寄存器 17	C0MDATA617			√		不确定
03FEC327H	CAN0 信息数据字节 7 寄存器 17	C0MDATA717			√		不确定
03FEC328H	CAN0 信息数据长度寄存器 17	C0MDLC17			√		0000xxxxB
03FEC329H	CAN0 信息配置寄存器 17	C0MCONF17			√		不确定
03FEC32AH	CAN0 信息标识符寄存器 17	C0MIDL17				√	不确定
03FEC32CH		C0MIDH17				√	不确定
03FEC32EH	CAN0 信息控制寄存器 17	C0MCTRL17				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (11/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC340H	CAN0 信息数据字节 01 寄存器 18	C0MDATA0118	R/W			√	不确定
03FEC340H	CAN0 信息数据字节 0 寄存器 18	C0MDATA018			√		不确定
03FEC341H	CAN0 信息数据字节 1 寄存器 18	C0MDATA118			√		不确定
03FEC342H	CAN0 信息数据字节 23 寄存器 18	C0MDATA2318				√	不确定
03FEC342H	CAN0 信息数据字节 2 寄存器 18	C0MDATA218			√		不确定
03FEC343H	CAN0 信息数据字节 3 寄存器 18	C0MDATA318			√		不确定
03FEC344H	CAN0 信息数据字节 45 寄存器 18	C0MDATA4518				√	不确定
03FEC344H	CAN0 信息数据字节 4 寄存器 18	C0MDATA418			√		不确定
03FEC345H	CAN0 信息数据字节 5 寄存器 18	C0MDATA518			√		不确定
03FEC346H	CAN0 信息数据字节 67 寄存器 18	C0MDATA6718				√	不确定
03FEC346H	CAN0 信息数据字节 6 寄存器 18	C0MDATA618			√		不确定
03FEC347H	CAN0 信息数据字节 7 寄存器 18	C0MDATA718			√		不确定
03FEC348H	CAN0 信息数据长度寄存器 18	C0MDLC18			√		0000xxxxB
03FEC349H	CAN0 信息配置寄存器 18	C0MCONF18			√		不确定
03FEC34AH	CAN0 信息标识符寄存器 18	C0MIDL18				√	不确定
03FEC34CH		C0MIDH18				√	不确定
03FEC34EH	CAN0 信息控制寄存器 18	C0MCTRL18				√	00x00000 000xx000B
03FEC360H	CAN0 信息数据字节 01 寄存器 19	C0MDATA0119				√	不确定
03FEC360H	CAN0 信息数据字节 0 寄存器 19	C0MDATA019			√		不确定
03FEC361H	CAN0 信息数据字节 1 寄存器 19	C0MDATA119			√		不确定
03FEC362H	CAN0 信息数据字节 23 寄存器 19	C0MDATA2319				√	不确定
03FEC362H	CAN0 信息数据字节 2 寄存器 19	C0MDATA219			√		不确定
03FEC363H	CAN0 信息数据字节 3 寄存器 19	C0MDATA319			√		不确定
03FEC364H	CAN0 信息数据字节 45 寄存器 19	C0MDATA4519				√	不确定
03FEC364H	CAN0 信息数据字节 4 寄存器 19	C0MDATA419			√		不确定
03FEC365H	CAN0 信息数据字节 5 寄存器 19	C0MDATA519			√		不确定
03FEC366H	CAN0 信息数据字节 67 寄存器 19	C0MDATA6719				√	不确定
03FEC366H	CAN0 信息数据字节 6 寄存器 19	C0MDATA619			√		不确定
03FEC367H	CAN0 信息数据字节 7 寄存器 19	C0MDATA719			√		不确定
03FEC368H	CAN0 信息数据长度寄存器 19	C0MDLC19			√		0000xxxxB
03FEC369H	CAN0 信息配置寄存器 19	C0MCONF19			√		不确定
03FEC36AH	CAN0 信息标识符寄存器 19	C0MIDL19				√	不确定
03FEC36CH		C0MIDH19				√	不确定
03FEC36EH	CAN0 信息控制寄存器 19	C0MCTRL19				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (12/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC380H	CAN0 信息数据字节 01 寄存器 20	C0MDATA0120	R/W			√	不确定
03FEC380H	CAN0 信息数据字节 0 寄存器 20	C0MDATA020			√		不确定
03FEC381H	CAN0 信息数据字节 1 寄存器 20	C0MDATA120			√		不确定
03FEC382H	CAN0 信息数据字节 23 寄存器 20	C0MDATA2320				√	不确定
03FEC382H	CAN0 信息数据字节 2 寄存器 20	C0MDATA220			√		不确定
03FEC383H	CAN0 信息数据字节 3 寄存器 20	C0MDATA320			√		不确定
03FEC384H	CAN0 信息数据字节 45 寄存器 20	C0MDATA4520				√	不确定
03FEC384H	CAN0 信息数据字节 4 寄存器 20	C0MDATA420			√		不确定
03FEC385H	CAN0 信息数据字节 5 寄存器 20	C0MDATA520			√		不确定
03FEC386H	CAN0 信息数据字节 67 寄存器 20	C0MDATA6720				√	不确定
03FEC386H	CAN0 信息数据字节 6 寄存器 20	C0MDATA620			√		不确定
03FEC387H	CAN0 信息数据字节 7 寄存器 20	C0MDATA720			√		不确定
03FEC388H	CAN0 信息数据长度寄存器 20	C0MDLC20			√		0000xxxxB
03FEC389H	CAN0 信息配置寄存器 20	C0MCONF20			√		不确定
03FEC38AH	CAN0 信息标识符寄存器 20	C0MIDL20				√	不确定
03FEC38CH		C0MIDH20				√	不确定
03FEC38EH	CAN0 信息控制寄存器 20	C0MCTRL20				√	00x00000 000xx000B
03FEC3A0H	CAN0 信息数据字节 01 寄存器 21	C0MDATA0121				√	不确定
03FEC3A0H	CAN0 信息数据字节 0 寄存器 21	C0MDATA021			√		不确定
03FEC3A1H	CAN0 信息数据字节 1 寄存器 21	C0MDATA121			√		不确定
03FEC3A2H	CAN0 信息数据字节 23 寄存器 21	C0MDATA2321				√	不确定
03FEC3A2H	CAN0 信息数据字节 2 寄存器 21	C0MDATA221			√		不确定
03FEC3A3H	CAN0 信息数据字节 3 寄存器 21	C0MDATA321			√		不确定
03FEC3A4H	CAN0 信息数据字节 45 寄存器 21	C0MDATA4521				√	不确定
03FEC3A4H	CAN0 信息数据字节 4 寄存器 21	C0MDATA421			√		不确定
03FEC3A5H	CAN0 信息数据字节 5 寄存器 21	C0MDATA521			√		不确定
03FEC3A6H	CAN0 信息数据字节 67 寄存器 21	C0MDATA6721				√	不确定
03FEC3A6H	CAN0 信息数据字节 6 寄存器 21	C0MDATA621			√		不确定
03FEC3A7H	CAN0 信息数据字节 7 寄存器 21	C0MDATA721			√		不确定
03FEC3A8H	CAN0 信息数据长度寄存器 21	C0MDLC21			√		0000xxxxB
03FEC3A9H	CAN0 信息配置寄存器 21	C0MCONF21			√		不确定
03FEC3AAH	CAN0 信息标识符寄存器 21	C0MIDL21				√	不确定
03FEC3ACH		C0MIDH21				√	不确定
03FEC3AEH	CAN0 信息控制寄存器 21	C0MCTRL21				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (13/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC3C0H	CAN0 信息数据字节 01 寄存器 22	C0MDATA0122	R/W			√	不确定
03FEC3C0H	CAN0 信息数据字节 0 寄存器 22	C0MDATA022			√		不确定
03FEC3C1H	CAN0 信息数据字节 1 寄存器 22	C0MDATA122			√		不确定
03FEC3C2H	CAN0 信息数据字节 23 寄存器 22	C0MDATA2322				√	不确定
03FEC3C2H	CAN0 信息数据字节 2 寄存器 22	C0MDATA222			√		不确定
03FEC3C3H	CAN0 信息数据字节 3 寄存器 22	C0MDATA322			√		不确定
03FEC3C4H	CAN0 信息数据字节 45 寄存器 22	C0MDATA4522				√	不确定
03FEC3C4H	CAN0 信息数据字节 4 寄存器 22	C0MDATA422			√		不确定
03FEC3C5H	CAN0 信息数据字节 5 寄存器 22	C0MDATA522			√		不确定
03FEC3C6H	CAN0 信息数据字节 67 寄存器 22	C0MDATA6722				√	不确定
03FEC3C6H	CAN0 信息数据字节 6 寄存器 22	C0MDATA622			√		不确定
03FEC3C7H	CAN0 信息数据字节 7 寄存器 22	C0MDATA722			√		不确定
03FEC3C8H	CAN0 信息数据长度寄存器 22	C0MDLC22			√		0000xxxxB
03FEC3C9H	CAN0 信息配置寄存器 22	C0MCONF22			√		不确定
03FEC3CAH	CAN0 信息标识符寄存器 22	C0MIDL22				√	不确定
03FEC3CCH		C0MIDH22				√	不确定
03FEC3CEH	CAN0 信息控制寄存器 22	C0MCTRL22				√	00x00000 000xx000B
03FEC3E0H	CAN0 信息数据字节 01 寄存器 23	C0MDATA0123				√	不确定
03FEC3E0H	CAN0 信息数据字节 0 寄存器 23	C0MDATA023			√		不确定
03FEC3E1H	CAN0 信息数据字节 1 寄存器 23	C0MDATA123			√		不确定
03FEC3E2H	CAN0 信息数据字节 23 寄存器 23	C0MDATA2323				√	不确定
03FEC3E2H	CAN0 信息数据字节 2 寄存器 23	C0MDATA223			√		不确定
03FEC3E3H	CAN0 信息数据字节 3 寄存器 23	C0MDATA323			√		不确定
03FEC3E4H	CAN0 信息数据字节 45 寄存器 23	C0MDATA4523				√	不确定
03FEC3E4H	CAN0 信息数据字节 4 寄存器 23	C0MDATA423			√		不确定
03FEC3E5H	CAN0 信息数据字节 5 寄存器 23	C0MDATA523			√		不确定
03FEC3E6H	CAN0 信息数据字节 67 寄存器 23	C0MDATA6723				√	不确定
03FEC3E6H	CAN0 信息数据字节 6 寄存器 23	C0MDATA623			√		不确定
03FEC3E7H	CAN0 信息数据字节 7 寄存器 23	C0MDATA723			√		不确定
03FEC3E8H	CAN0 信息数据长度寄存器 23	C0MDLC23			√		0000xxxxB
03FEC3E9H	CAN0 信息配置寄存器 23	C0MCONF23			√		不确定
03FEC3EAH	CAN0 信息标识符寄存器 23	C0MIDL23				√	不确定
03FEC3ECH		C0MIDH23				√	不确定
03FEC3EEH	CAN0 信息控制寄存器 23	C0MCTRL23				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (14/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC400H	CAN0 信息数据字节 01 寄存器 24	C0MDATA0124	R/W			√	不确定
03FEC400H	CAN0 信息数据字节 0 寄存器 24	C0MDATA024			√		不确定
03FEC401H	CAN0 信息数据字节 1 寄存器 24	C0MDATA124			√		不确定
03FEC402H	CAN0 信息数据字节 23 寄存器 24	C0MDATA2324				√	不确定
03FEC402H	CAN0 信息数据字节 2 寄存器 24	C0MDATA224			√		不确定
03FEC403H	CAN0 信息数据字节 3 寄存器 24	C0MDATA324			√		不确定
03FEC404H	CAN0 信息数据字节 45 寄存器 24	C0MDATA4524				√	不确定
03FEC404H	CAN0 信息数据字节 4 寄存器 24	C0MDATA424			√		不确定
03FEC405H	CAN0 信息数据字节 5 寄存器 24	C0MDATA524			√		不确定
03FEC406H	CAN0 信息数据字节 67 寄存器 24	C0MDATA6724				√	不确定
03FEC406H	CAN0 信息数据字节 6 寄存器 24	C0MDATA624			√		不确定
03FEC407H	CAN0 信息数据字节 7 寄存器 24	C0MDATA724			√		不确定
03FEC408H	CAN0 信息数据长度寄存器 24	C0MDLC24			√		0000xxxxB
03FEC409H	CAN0 信息配置寄存器 24	C0MCONF24			√		不确定
03FEC40AH	CAN0 信息标识符寄存器 24	C0MIDL24				√	不确定
03FEC40CH		C0MIDH24				√	不确定
03FEC40EH	CAN0 信息控制寄存器 24	C0MCTRL24				√	00x00000 000xx000B
03FEC420H	CAN0 信息数据字节 01 寄存器 25	C0MDATA0125				√	不确定
03FEC420H	CAN0 信息数据字节 0 寄存器 25	C0MDATA025			√		不确定
03FEC421H	CAN0 信息数据字节 1 寄存器 25	C0MDATA125			√		不确定
03FEC422H	CAN0 信息数据字节 23 寄存器 25	C0MDATA2325				√	不确定
03FEC422H	CAN0 信息数据字节 2 寄存器 25	C0MDATA225			√		不确定
03FEC423H	CAN0 信息数据字节 3 寄存器 25	C0MDATA325			√		不确定
03FEC424H	CAN0 信息数据字节 45 寄存器 25	C0MDATA4525				√	不确定
03FEC424H	CAN0 信息数据字节 4 寄存器 25	C0MDATA425			√		不确定
03FEC425H	CAN0 信息数据字节 5 寄存器 25	C0MDATA525			√		不确定
03FEC426H	CAN0 信息数据字节 67 寄存器 25	C0MDATA6725				√	不确定
03FEC426H	CAN0 信息数据字节 6 寄存器 25	C0MDATA625			√		不确定
03FEC427H	CAN0 信息数据字节 7 寄存器 25	C0MDATA725			√		不确定
03FEC428H	CAN0 信息数据长度寄存器 25	C0MDLC25			√		0000xxxxB
03FEC429H	CAN0 信息配置寄存器 25	C0MCONF25			√		不确定
03FEC42AH	CAN0 信息标识符寄存器 25	C0MIDL25				√	不确定
03FEC42CH		C0MIDH25				√	不确定
03FEC42EH	CAN0 信息控制寄存器 25	C0MCTRL25				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (15/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC440H	CAN0 信息数据字节 01 寄存器 26	C0MDATA0126	R/W			√	不确定
03FEC440H	CAN0 信息数据字节 0 寄存器 26	C0MDATA026			√		不确定
03FEC441H	CAN0 信息数据字节 1 寄存器 26	C0MDATA126			√		不确定
03FEC442H	CAN0 信息数据字节 23 寄存器 26	C0MDATA2326				√	不确定
03FEC442H	CAN0 信息数据字节 2 寄存器 26	C0MDATA226			√		不确定
03FEC443H	CAN0 信息数据字节 3 寄存器 26	C0MDATA326			√		不确定
03FEC444H	CAN0 信息数据字节 45 寄存器 26	C0MDATA4526				√	不确定
03FEC444H	CAN0 信息数据字节 4 寄存器 26	C0MDATA426			√		不确定
03FEC445H	CAN0 信息数据字节 5 寄存器 26	C0MDATA526			√		不确定
03FEC446H	CAN0 信息数据字节 67 寄存器 26	C0MDATA6726				√	不确定
03FEC446H	CAN0 信息数据字节 6 寄存器 26	C0MDATA626			√		不确定
03FEC447H	CAN0 信息数据字节 7 寄存器 26	C0MDATA726			√		不确定
03FEC448H	CAN0 信息数据长度寄存器 26	C0MDLC26			√		0000xxxxB
03FEC449H	CAN0 信息配置寄存器 26	C0MCONF26			√		不确定
03FEC44AH	CAN0 信息标识符寄存器 26	C0MIDL26				√	不确定
03FEC44CH		C0MIDH26				√	不确定
03FEC44EH	CAN0 信息控制寄存器 26	C0MCTRL26				√	00x00000 000xx000B
03FEC460H	CAN0 信息数据字节 01 寄存器 27	C0MDATA0127				√	不确定
03FEC460H	CAN0 信息数据字节 0 寄存器 27	C0MDATA027			√		不确定
03FEC461H	CAN0 信息数据字节 1 寄存器 27	C0MDATA127			√		不确定
03FEC462H	CAN0 信息数据字节 23 寄存器 27	C0MDATA2327				√	不确定
03FEC462H	CAN0 信息数据字节 2 寄存器 27	C0MDATA227			√		不确定
03FEC463H	CAN0 信息数据字节 3 寄存器 27	C0MDATA327			√		不确定
03FEC464H	CAN0 信息数据字节 45 寄存器 27	C0MDATA4527				√	不确定
03FEC464H	CAN0 信息数据字节 4 寄存器 27	C0MDATA427			√		不确定
03FEC465H	CAN0 信息数据字节 5 寄存器 27	C0MDATA527			√		不确定
03FEC466H	CAN0 信息数据字节 67 寄存器 27	C0MDATA6727				√	不确定
03FEC466H	CAN0 信息数据字节 6 寄存器 27	C0MDATA627			√		不确定
03FEC467H	CAN0 信息数据字节 7 寄存器 27	C0MDATA727			√		不确定
03FEC468H	CAN0 信息数据长度寄存器 27	C0MDLC27			√		0000xxxxB
03FEC469H	CAN0 信息配置寄存器 27	C0MCONF27			√		不确定
03FEC46AH	CAN0 信息标识符寄存器 27	C0MIDL27				√	不确定
03FEC46CH		C0MIDH27				√	不确定
03FEC46EH	CAN0 信息控制寄存器 27	C0MCTRL27				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (16/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC480H	CAN0 信息数据字节 01 寄存器 28	C0MDATA0128	R/W			√	不确定
03FEC480H	CAN0 信息数据字节 0 寄存器 28	C0MDATA028			√		不确定
03FEC481H	CAN0 信息数据字节 1 寄存器 28	C0MDATA128			√		不确定
03FEC482H	CAN0 信息数据字节 23 寄存器 28	C0MDATA2328				√	不确定
03FEC482H	CAN0 信息数据字节 2 寄存器 28	C0MDATA228			√		不确定
03FEC483H	CAN0 信息数据字节 3 寄存器 28	C0MDATA328			√		不确定
03FEC484H	CAN0 信息数据字节 45 寄存器 28	C0MDATA4528				√	不确定
03FEC484H	CAN0 信息数据字节 4 寄存器 28	C0MDATA428			√		不确定
03FEC485H	CAN0 信息数据字节 5 寄存器 28	C0MDATA528			√		不确定
03FEC486H	CAN0 信息数据字节 67 寄存器 28	C0MDATA6728				√	不确定
03FEC486H	CAN0 信息数据字节 6 寄存器 28	C0MDATA628			√		不确定
03FEC487H	CAN0 信息数据字节 7 寄存器 28	C0MDATA728			√		不确定
03FEC488H	CAN0 信息数据长度寄存器 28	C0MDLC28			√		0000xxxxB
03FEC489H	CAN0 信息配置寄存器 28	C0MCONF28			√		不确定
03FEC48AH	CAN0 信息标识符寄存器 28	C0MIDL28				√	不确定
03FEC48CH		C0MIDH28				√	不确定
03FEC48EH	CAN0 信息控制寄存器 28	C0MCTRL28				√	00x00000 000xx000B
03FEC4A0H	CAN0 信息数据字节 01 寄存器 29	C0MDATA0129				√	不确定
03FEC4A0H	CAN0 信息数据字节 0 寄存器 29	C0MDATA029			√		不确定
03FEC4A1H	CAN0 信息数据字节 1 寄存器 29	C0MDATA129			√		不确定
03FEC4A2H	CAN0 信息数据字节 23 寄存器 29	C0MDATA2329				√	不确定
03FEC4A2H	CAN0 信息数据字节 2 寄存器 29	C0MDATA229			√		不确定
03FEC4A3H	CAN0 信息数据字节 3 寄存器 29	C0MDATA329			√		不确定
03FEC4A4H	CAN0 信息数据字节 45 寄存器 29	C0MDATA4529				√	不确定
03FEC4A4H	CAN0 信息数据字节 4 寄存器 29	C0MDATA429			√		不确定
03FEC4A5H	CAN0 信息数据字节 5 寄存器 29	C0MDATA529			√		不确定
03FEC4A6H	CAN0 信息数据字节 67 寄存器 29	C0MDATA6729				√	不确定
03FEC4A6H	CAN0 信息数据字节 6 寄存器 29	C0MDATA629			√		不确定
03FEC4A7H	CAN0 信息数据字节 7 寄存器 29	C0MDATA729			√		不确定
03FEC4A8H	CAN0 信息数据长度寄存器 29	C0MDLC29			√		0000xxxxB
03FEC4A9H	CAN0 信息配置寄存器 29	C0MCONF29			√		不确定
03FEC4AAH	CAN0 信息标识符寄存器 29	C0MIDL29				√	不确定
03FEC4ACH		C0MIDH29				√	不确定
03FEC4AEH	CAN0 信息控制寄存器 29	C0MCTRL29				√	00x00000 000xx000B

表 19-16. 寄存器访问类型 (17/17)

地址	寄存器名称	符号	R/W	位操作单元			复位后
				1 位	8 位	16 位	
03FEC4C0H	CAN0 信息数据字节 01 寄存器 30	C0MDATA0130	R/W			√	不确定
03FEC4C0H	CAN0 信息数据字节 0 寄存器 30	C0MDATA030			√		不确定
03FEC4C1H	CAN0 信息数据字节 1 寄存器 30	C0MDATA130			√		不确定
03FEC4C2H	CAN0 信息数据字节 23 寄存器 30	C0MDATA2330				√	不确定
03FEC4C2H	CAN0 信息数据字节 2 寄存器 30	C0MDATA230			√		不确定
03FEC4C3H	CAN0 信息数据字节 3 寄存器 30	C0MDATA330			√		不确定
03FEC4C4H	CAN0 信息数据字节 45 寄存器 30	C0MDATA4530				√	不确定
03FEC4C4H	CAN0 信息数据字节 4 寄存器 30	C0MDATA430			√		不确定
03FEC4C5H	CAN0 信息数据字节 5 寄存器 30	C0MDATA530			√		不确定
03FEC4C6H	CAN0 信息数据字节 67 寄存器 30	C0MDATA6730				√	不确定
03FEC4C6H	CAN0 信息数据字节 6 寄存器 30	C0MDATA630			√		不确定
03FEC4C7H	CAN0 信息数据字节 7 寄存器 30	C0MDATA730			√		不确定
03FEC4C8H	CAN0 信息数据长度寄存器 30	C0MDLC30			√		0000xxxxB
03FEC4C9H	CAN0 信息配置寄存器 30	C0MCONF30			√		不确定
03FEC4CAH	CAN0 信息标识符寄存器 30	C0MIDL30				√	不确定
03FEC4CCH		C0MIDH30				√	不确定
03FEC4CEH	CAN0 信息控制寄存器 30	C0MCTRL30				√	00x00000 000xx000B
03FEC4E0H	CAN0 信息数据字节 01 寄存器 31	C0MDATA0131				√	不确定
03FEC4E0H	CAN0 信息数据字节 0 寄存器 31	C0MDATA031			√		不确定
03FEC4E1H	CAN0 信息数据字节 1 寄存器 31	C0MDATA131			√		不确定
03FEC4E2H	CAN0 信息数据字节 23 寄存器 31	C0MDATA2331				√	不确定
03FEC4E2H	CAN0 信息数据字节 2 寄存器 31	C0MDATA231			√		不确定
03FEC4E3H	CAN0 信息数据字节 3 寄存器 31	C0MDATA331			√		不确定
03FEC4E4H	CAN0 信息数据字节 45 寄存器 31	C0MDATA4531				√	不确定
03FEC4E4H	CAN0 信息数据字节 4 寄存器 31	C0MDATA431			√		不确定
03FEC4E5H	CAN0 信息数据字节 5 寄存器 31	C0MDATA531			√		不确定
03FEC4E6H	CAN0 信息数据字节 67 寄存器 31	C0MDATA6731				√	不确定
03FEC4E6H	CAN0 信息数据字节 6 寄存器 31	C0MDATA631			√		不确定
03FEC4E7H	CAN0 信息数据字节 7 寄存器 31	C0MDATA731			√		不确定
03FEC4E8H	CAN0 信息数据长度寄存器 31	C0MDLC31			√		0000xxxxB
03FEC4E9H	CAN0 信息配置寄存器 31	C0MCONF31			√		不确定
03FEC4EAH	CAN0 信息标识符寄存器 31	C0MIDL31				√	不确定
03FEC4ECH		C0MIDH31				√	不确定
03FEC4EEH	CAN0 信息控制寄存器 31	C0MCTRL31				√	00x00000 000xx000B

19.5.3 寄存器位配置

表 19-17. CAN 全局寄存器位配置

地址	符号	位 7/15	位 6/14	位 5/13	位 4/12	位 3/11	位 2/10	位 1/9	位 0/8
03FEC000H	C0GMCTRL (W)	0	0	0	0	0	0	0	清除 GOM
03FEC001H		0	0	0	0	0	0	设置 EFSD	设置 GOM
03FEC000H	C0GMCTRL (R)	0	0	0	0	0	0	EFSD	GOM
03FEC001H		MBON	0	0	0	0	0	0	0
03FEC002H	C0GMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0
03FEC006H	C0GMABT (W)	0	0	0	0	0	0	0	清除 ABTTRG
03FEC007H		0	0	0	0	0	0	设置 ABTCLR	设置 ABTTRG
03FEC006H	C0GMABT (R)	0	0	0	0	0	0	ABTCLR	ABTTRG
03FEC007H		0	0	0	0	0	0	0	0
03FEC008H	C0GMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

表 19-18. CAN 模块寄存器位配置 (1/2)

地址	符号	位 7/15	位 6/14	位 5/13	位 4/12	位 3/11	位 2/10	位 1/9	位 0/8
03FEC040H	C0MASK1L	CMID7 到 CMID0							
03FEC041H		CMID15 到 CMID8							
03FEC042H	C0MASK1H	CMID23 到 CMID16							
03FEC043H		0	0	0	CMID28 到 CMID24				
03FEC044H	C0MASK2L	CMID7 到 CMID0							
03FEC045H		CMID15 到 CMID8							
03FEC046H	C0MASK2H	CMID23 到 CMID16							
03FEC047H		0	0	0	CMID28 到 CMID24				
03FEC048H	C0MASK3L	CMID7 到 CMID0							
03FEC049H		CMID15 到 CMID8							
03FEC04AH	C0MASK3H	CMID23 到 CMID16							
03FEC04BH		0	0	0	CMID28 到 CMID24				
03FEC04CH	C0MASK4L	CMID7 到 CMID0							
03FEC04DH		CMID15 到 CMID8							
03FEC04EH	C0MASK4H	CMID23 到 CMID16							
03FEC04FH		0	0	0	CMID28 到 CMID24				
03FEC050H	C0CTRL (W)	0	清除 AL	清除 VALID	清除 PSMODE1	清除 PSMODE0	清除 OPMODE2	清除 OPMODE1	清除 OPMODE0
03FEC051H		设置 CCERC	设置 AL	0	设置 PSMODE1	设置 PSMODE0	设置 OPMODE2	设置 OPMODE1	设置 OPMODE0
03FEC050H	C0CTRL (R)	CCERC	AL	VALID	PS MODE1	PS MODE0	OP MODE2	OP MODE1	OP MODE0
03FEC051H		0	0	0	0	0	0	RSTAT	TSTAT
03FEC052H	C0LEC (W)	0	0	0	0	0	0	0	0
03FEC052H	C0LEC (R)	0	0	0	0	0	LEC2	LEC1	LEC0
03FEC053H	C0INFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0
03FEC054H	C0ERC	TEC7 到 TEC0							
03FEC055H		REC7 到 REC0							
03FEC056H	C0IE (W)	0	0	清除 CIE5	清除 CIE4	清除 CIE3	清除 CIE2	清除 CIE1	清除 CIE0
03FEC057H		0	0	设置 CIE5	设置 CIE4	设置 CIE3	设置 CIE2	设置 CIE1	设置 CIE0
03FEC056H	C0IE (R)	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
03FEC057H		0	0	0	0	0	0	0	0
03FEC058H	C0INTS (W)	0	0	清除 CINTS5	清除 CINTS4	清除 CINTS3	清除 CINTS2	清除 CINTS1	清除 CINTS0
03FEC059H		0	0	0	0	0	0	0	0
03FEC058H	C0INTS (R)	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0
03FEC059H		0	0	0	0	0	0	0	0

表 19-18. CAN 模块寄存器位配置 (2/2)

地址	符号	位 7/15	位 6/14	位 5/13	位 4/12	位 3/11	位 2/10	位 1/9	位 0/8
03FEC05AH	C0BRP	TQPRS7 到 TQPRS0							
03FEC05CH	C0BTR	0	0	0	0	TSEG13 到 TSEG10			
03FEC05DH		0	0	SJW1, SJW0		0	TSEG22 到 TSEG20		
03FEC05EH	C0LIPT	LIPT7 到 LIPT0							
03FEC060H	C0RGPT (W)	0	0	0	0	0	0	0	清除 ROVF
03FEC061H		0	0	0	0	0	0	0	0
03FEC060H	C0RGPT (R)	0	0	0	0	0	0	RHPM	ROVF
03FEC061H		RGPT7 到 RGPT0							
03FEC062H	C0LOPT	LOPT7 到 LOPT0							
03FEC064H	C0TGPT (W)	0	0	0	0	0	0	0	清除到 VF
03FEC065H		0	0	0	0	0	0	0	0
03FEC064H	C0TGPT (R)	0	0	0	0	0	0	THPM	到 VF
03FEC065H		TGPT7 到 TGPT0							
03FEC066H	C0TS (W)	0	0	0	0	0	清除 TSLOCK	清除 TSSEL	清除 TSEN
03FEC067H		0	0	0	0	0	设置 TSLOCK	设置 TSSEL	设置 TSEN
03FEC066H	C0TS (R)	0	0	0	0	0	TSLOCK	TSSEL	TSEN
03FEC067H		0	0	0	0	0	0	0	0
03FEC068H 到 03FEC0FFH	—	禁止访问（保留，以备将来使用）							

表 19-19. 信息缓冲器寄存器位配置

地址	符号	位 7/15	位 6/14	位 5/13	位 4/12	位 3/11	位 2/10	位 1/9	位 0/8
03FECxx0H	C0MDATA01m	信息数据（字节 0）							
03FECxx1H		信息数据（字节 1）							
03FECxx0H	C0MDATA0m	信息数据（字节 0）							
03FECxx1H	C0MDATA1m	信息数据（字节 1）							
03FECxx2H	C0MDATA23m	信息数据（字节 2）							
03FECxx3H		信息数据（字节 3）							
03FECxx2H	C0MDATA2m	信息数据（字节 2）							
03FECxx3H	C0MDATA3m	信息数据（字节 3）							
03FECxx4H	C0MDATA45m	信息数据（字节 4）							
03FECxx5H		信息数据（字节 5）							
03FECxx4H	C0MDATA4m	信息数据（字节 4）							
03FECxx5H	C0MDATA5m	信息数据（字节 5）							
03FECxx6H	C0MDATA67m	信息数据（字节 6）							
03FECxx7H		信息数据（字节 7）							
03FECxx6H	C0MDATA6m	信息数据（字节 6）							
03FECxx7H	C0MDATA7m	信息数据（字节 7）							
03FECxx8H	C0MDLCm	0				MDLC3	MDLC2	MDLC1	MDLC0
03FECxx9H	C0MCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0
03FECxxAH	C0MIDLm	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
03FECxxBH		ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
03FECxxCH	C0MIDHm	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
03FECxxDH		IDE	0	0	ID28	ID27	ID26	ID25	ID24
03FECxxEH	C0MCTRLm (W)	0	0	0	清除 MOW	清除 IE	清除 DN	清除 TRQ	清除 RDY
03FECxxFH		0	0	0	0	设置 IE	0	设置 TRQ	设置 RDY
03FECxxEH	C0MCTRLm (R)	0	0	0	MOW	IE	DN	TRQ	RDY
03FECxxFH		0	0	MUC	0	0	0	0	0
03FECxx0 到 03FECxxFH	—	禁止访问（保留，以备将来使用）							

备注 m = 00 到 31

xx = 10, 12, 14, 16, 18, 1A, 1C, 1E, 20, 22, 24, 26, 28, 2A, 2C, 2E, 30, 32, 34, 36, 38,
3A, 3C, 3E, 40, 42, 44, 46, 48, 4A, 4C, 4E

19.6 寄存器

注意事项 在下列状态下对 CAN 控制寄存器的访问是被禁止的。如果需要详细信息，请参考 3.4.9 (2) 访问特殊的片上外围 I/O 寄存器。

- 当 CPU 在子系统时钟下工作且主时钟振荡器停止运行。
- 当 CPU 在内部振荡时钟下工作。

备注 m = 00 到 31

(1) CAN0 全局控制寄存器 (C0GMCTRL)

C0GMCTRL 寄存器用于控制 CAN 模块的操作。

(1/2)

复位后的值: 0000H R/W 地址: 03FEC000H

(a) 读取

	15	14	13	12	11	10	9	8
C0GMCTRL	MBON	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	EFSD	GOM

(b) 写入

	15	14	13	12	11	10	9	8
C0GMCTRL	0	0	0	0	0	0	设置 EFSD	设置 GOM
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	清除 GOM

(a) 读取

MBON	对于信息缓存寄存器和发送 / 接收历史寄存器的访问允许位
0	禁止对于信息缓存寄存器和发送 / 接收历史列表寄存器的写入操作和读取操作。
1	允许对于信息缓存寄存器和发送 / 接收历史列表寄存器的写入操作和读取操作。

- 注意事项**
1. 当 MBON 位被清零后，禁止通过软件对信息缓存 (C0MDATA0m, C0MDATA1m, C0MDATA01m, C0MDATA2m, C0MDATA3m, C0MDATA23m, C0MDATA4m, C0MDATA5m, C0MDATA45m, C0MDATA6m, C0MDATA7m, C0MDATA67m, C0MDLCm, C0MCONFm, C0MIDLm, C0MIDHm 和 C0MCTRLm) 以及发送历史和接收历史的相关寄存器 (C0LOPT, C0TGPT, C0LIPT 和 C0RGPT) 进行访问。
 2. 这是一个只读位。即使在 MBON 为 0 时将它置 1，它的值并不会因此而改变依然为 0，并且依然禁止对信息缓存寄存器或是发送历史和接收历史的相关寄存器的访问。

备注 当进入 CAN 休眠模式 / CAN 停止模式，或者 GOM 位被清 0 后，MBON 位也将被清 0。当从 CAN 休眠模式 / CAN 停止模式中释放，或者 GOM 位被置 1 时，MBON 位会被设为 1。

EFSD	强制关闭的允许位
0	禁止通过设定 GOM 位为 0 来执行强制关闭。
1	允许通过设定 GOM 位为 0 来执行强制关闭。

注意事项 为了请求强制关闭，必须在 EFSD 位被置 1 后立即将 GOM 位清 0。如果在这之前对其它任何寄存器进行访问（包括对 C0GMCTRL 寄存器进行读取），都会使 EFSD 位被强制清 0，导致强制关闭的请求无效。

GOM	全局操作模式位
0	禁止对 CAN 模块的操作。
1	允许对 CAN 模块的操作。

注意事项 GOM 位仅在初始化模式或者 EFSD 位置 1 后被清零。

(b) 写入

设置 EFSD	EFSD 位设定
0	EFSD 位保持不变
1	EFSD 位被置 1

设置 GOM	清零 GOM	GOM 位设定
0	1	GOM 位被清 0
1	0	GOM 位被置 1
其它		GOM 位保持不变

注意事项 一定要分别设置 GOM 位和 EFSD 位。

<R>

(2) CAN0 全局时钟选择寄存器 (C0GMCS)

C0GMCS 寄存器用于选择 CAN 模块的系统时钟。

复位后的值: 0FH R/W 地址: 03FEC002H

	7	6	5	4	3	2	1	0
C0GMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0

CCP3	CCP2	CCP1	CCP1	CAN 模块系统时钟 (f _{CANMOD})
0	0	0	0	f _{CAN} /1
0	0	0	1	f _{CAN} /2
0	0	1	0	f _{CAN} /3
0	0	1	1	f _{CAN} /4
0	1	0	0	f _{CAN} /5
0	1	0	1	f _{CAN} /6
0	1	1	0	f _{CAN} /7
0	1	1	1	f _{CAN} /8
1	0	0	0	f _{CAN} /9
1	0	0	1	f _{CAN} /10
1	0	1	0	f _{CAN} /11
1	0	1	1	f _{CAN} /12
1	1	0	0	f _{CAN} /13
1	1	0	1	f _{CAN} /14
1	1	1	0	f _{CAN} /15
1	1	1	1	f _{CAN} /16 (默认值)

备注 f_{CAN} = 提供给 CAN 的时钟 = f_{xx}

(3) CAN0 全局自动块发送控制寄存器 (C0GMABT)

C0GMABT 寄存器用于控制自动块发送 (ABT) 的操作。

(1/2)

复位后的值: 0000H R/W 地址: 03FEC006H

(a) 读取

	15	14	13	12	11	10	9	8
C0GMABT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	ABTCLR	ABTTRG

(b) 写入

	15	14	13	12	11	10	9	8
C0GMABT	0	0	0	0	0	0	设置 ABTCLR	设置 ABTTRG
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	清除 ABTTRG

注意事项 当把带有 **ABT** 的正常操作模式转换到初始化模式之前，一定要将 **C0GMABT** 寄存器设定为它的初始值 (0000H)。然后，确定 **C0GMABT** 寄存器被初始化为 0000H。

(a) 读取

ABTCLR	自动块发送引擎清 0 状态位
0	对自动块发送引擎的清 0 已经完成
1	对自动块发送引擎的清 0 正在进行

- 备注**
1. 将 ABTTRG 位清 0 时，要将 ABTCLR 位置 1。
当 ABTCLR 和 ABTTRG 位同时为 1 时，无法保证操作的正确性。
 2. ABTCLR 位置 1 后，清除自动块发送引擎。一旦被请求的清除进程完成，ABTCLR 位将自动清 0。

ABTTRG	自动块发送状态位
0	自动块发送停止
1	自动块发送正在执行

- 注意事项**
1. 在初始化模式中不能将 **ABTTRG** 位置 1。否则在 **CAN** 模块进入带有 **ABT** 的正常操作模式后，不能保证操作的正确性。
 2. 不要将 **ABTTRG** 位和 **C0CTRL.TSTAT** 位同时置 1。将 **ABTTRG** 位置 1 前必须确认 **TSTAT** 位为 0。

(b) 写入

设置 ABTCLR	自动块发送引擎清除请求位
0	自动块发送引擎处于空闲状态或正在操作。
1	需要清除自动块发送引擎。在清除自动块发送引擎之后，通过把 ABTTTRG 位置 1 可以使自动块发送从信息缓存 0 起始。

设定 ABTTTRG	清除 ABTTTRG	自动块发送起始位
0	1	需要停止自动块发送
1	0	需要进行自动块发送
其它		ABTTTRG 位保持不变

<R>

注意事项 即使 **ABTTTRG** 位被置位，发送也不会立即执行，依赖于当如信息从另一个节点被接收时或当 **ABT** 信息（信息缓冲器 8 到 31）以外的信息被发送时的情况。

即使 **ABTTTRG** 位被清零，发送也不会被中途终止。如果发送正在执行，在完成前，它会继续（不考虑发送是成功还是失败）。

(4) CAN0 全局自动模块发送控制寄存器 (C0GMABT)

CnGMABTD 寄存器用于设定在带有 ABT 的正常操作模式中，分配给 ABT 的信息缓存中的数据发送的时间间隔。

复位后的值: 00H R/W 地址: 03FEC008H

	7	6	5	4	3	2	1	0
C0GMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

ABTD3	ABTD2	ABTD1	ABTD0	自动块发送期间的数据帧时间间隔 (单位: 数据位时间 (DBT))
0	0	0	0	0 DBT (默认值)
0	0	0	1	2^5 DBT
0	0	1	0	2^6 DBT
0	0	1	1	2^7 DBT
0	1	0	0	2^8 DBT
0	1	0	1	2^9 DBT
0	1	1	0	2^{10} DBT
0	1	1	1	2^{11} DBT
1	0	0	0	2^{12} DBT
其它				禁止设定

- 注意事项
1. 在 ABTTRG 位为 1 期间不能改变 C0GMABTD 寄存器的值。
 2. 在 CAN 总线上 ABT 信息实际发送时序的差异取决于其它节点的发送状态或者如何请求发送除 ABT 以外的信息 (信息缓存 8 到 31)。

(5) CAN0 模块屏蔽控制寄存器 (C0MASKaL, C0MASKaH) (a 为 1, 2, 3 或 4)

C0MASKaL 和 C0MASKaH 寄存器用于扩展可接收信息数量，该功能通过标记一部分的信息标识 (ID) 并废弃已经带有标志的部分。

(1/2)

- CAN0 模块屏蔽 1 寄存器 (C0MASK1L, C0MASK1H)

复位后的值: 不确定 R/W 地址: C0MASK1L 03FEC040H, C0MASK1H 03FEC042H

	15	14	13	12	11	10	9	8
C0MASK1L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
C0MASK1H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN0 模块屏蔽 2 寄存器 (C0MASK2L, C0MASK2H)

复位后的值: 不确定 R/W 地址: C0MASK2L 03FEC044H, C0MASK2H 03FEC046H

	15	14	13	12	11	10	9	8
C0MASK2L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
C0MASK2H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN0 模块屏蔽 3 寄存器（C0MASK3L, C0MASK3H）

复位后的值：不确定 R/W 地址： C0MASK3L 03FEC048H, C0MASK3H 03FEC04AH

C0MASK3L	15	14	13	12	11	10	9	8
	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
C0MASK3H	15	14	13	12	11	10	9	8
	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN0 模块屏蔽 4 寄存器（C0MASK4L, C0MASK4H）

复位后的值：不确定 R/W 地址： C0MASK4L 03FEC04CH, C0MASK4H 03FEC04EH

C0MASK4L	15	14	13	12	11	10	9	8
	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
C0MASK4H	15	14	13	12	11	10	9	8
	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

CMID28 到 CMID0	ID 位的屏蔽方式设定
0	比较 CMID28 到 CMID0 位设定的信息缓存的 ID 位和接收到帧的 ID 位。
1	不比较 CMID28 到 CMID0 位设定的信息缓存的 ID 位和接收到帧的 ID 位。（被屏蔽）。

备注 屏蔽永远是通过 29 位的 ID 长度定义的。如果要屏蔽的信息是标准 ID，则忽略 CMID17 到 CMID0 位。因此，只屏蔽接收 ID 的 CMID28 到 CMID18 位。相同的屏蔽可以用于标准和扩展的 ID。

(6) CAN0 模块控制寄存器 (C0CTRL)

C0CTRL 寄存器用于控制 CAN 模块的操作模式。

(1/4)

复位后的值: 0000H R/W 地址: 03FEC050H

(a) 读取

	15	14	13	12	11	10	9	8
C0CTRL	0	0	0	0	0	0	RSTAT	TSTAT
	7	6	5	4	3	2	1	0
	CCERC	AL	VALID	PSMODE	PSMODE	OPMODE	OPMODE	OPMODE
				1	0	2	1	0

(b) 写入

	15	14	13	12	11	10	9	8
C0CTRL	设置 CCERC	设置 AL	0	设置 PSMODE	设置 PSMODE	设置 OPMODE	设置 OPMODE	设置 OPMODE
				1	0	2	1	0
	7	6	5	4	3	2	1	0
	0	清除 AL	清除 VALID	清除 PSMODE	清除 PSMODE	清除 OPMODE	清除 OPMODE	清除 OPMODE
				1	0	2	1	0

(a) 读取

RSTAT	接收状态位
0	接收停止
1	正在进行接收.

备注 当满足以下条件（时序）时，RSTAT 位置 1。

- 检测到接收帧的 SOF 位
- 在发送帧时出现仲裁失败

当满足以下条件（时刻）时，RSTAT 位清 0。

- 在帧间空间的第二位检测到隐性电平
- 在帧间空间的第一位转换到初始化模式

TSTAT	发送状态位
0	发送停止.
1	正在进行发送

备注 当满足以下条件（时刻）时，TSTAT 位置 1。

- 检测到接收帧的 SOF 位

当满足以下条件（时刻）时，TSTAT 位清 0。

- 转换到 bus-off 状态
- 在发送帧时出现仲裁失败
- 在帧间空间的第二位检测到隐性电平
- 在帧间空间的第一位转换到初始化模式

<R>

CCERC	错误计数器清除位
0	在初始化模式中，不清除 C0ERC 和 C0INFO 寄存器
1	在初始化模式中，清除 C0ERC 和 C0INFO 寄存器

- 备注**
1. CCERC 位用于在重新初始化或者从 bus-off 状态强制恢复时清除 C0ERC 和 C0INFO 寄存器。仅能在初始化模式中将该位置 1。
 2. 当清除 C0ERC 和 C0INFO 寄存器时，自动清零 CCERC 位。
 3. 发出从初始化模式到操作模式的请求的同时可以将 CCERC 位置 1。
 4. 如果进入自测试模式的 INIT 模式后立即将 CCERC 位置 1，接收到的数据很可能是无效的。

AL	仲裁失败后的设置操作位
0	在单发模式下如果仲裁失败，不执行重传。
1	在单发模式下如果仲裁失败，执行重传。

- 备注** AL 位仅在单发模式下可用

VALID	正确接收信息帧检测位
0	自从 VALID 位上一次清 0 后没有接收到正确的信息帧。
1	自从 VALID 位上一次清 0 后已经接收到正确的信息帧。

- 备注**
1. 正确接收信息帧的检测不依赖于接收信息缓存（数据帧）或发送信息缓存（远程帧）中存储的存在或不存在。
 2. 在将初始化模式转换为操作模式之前，必须将 VALID 位清 0。
 3. 当只有两个 CAN 节点连接到 CAN 总线，并且一个在正常模式下发送信息而另一个处于接收模式时，在发送节点进入错误无效状态之前不能将 VALID 位置 1。
 4. 如果要将 VALID 位清 0，先将 Clear VALID 位置 1 然后确认 VALID 清 0。如果没有清零，则必须重复上述过程。

PSMODE1	PSMODE0	节能模式
0	0	未选择节能模式
0	1	CAN 休眠模式
1	0	禁止设定
1	1	CAN 停止模式

注意事项 1. 无论是切换到 CAN 停止模式还是释放 CAN 停止模式必须通过 CAN 休眠模式过渡。否则将忽略任何直接对 CAN 停止模式的请求。

2. 在释放节能模式后，必须在再次访问信息缓存之前检查 C0GMCTRL.MBON 标志。

3. 一个转换到 CAN 休眠模式的请求在它被软件取消之前或在 CAN 总线进入总线空闲状态之前被挂起。通过读取 PSMODE0 和 PSMODE1 位，软件可以检查转换到 CAN 休眠模式。

OPMODE2	OPMODE1	OPMODE0	操作模式
0	0	0	未选择操作模式（CAN 模块处于初始化模式）。
0	0	1	正常操作模式
0	1	0	带有自动块发送功能的正常操作模式（带有 ABT 的正常操作模式）
0	1	1	只接收模式
1	0	0	单发模式
1	0	1	自测试模式
其它			禁止设定

注意事项 更改为初始化模式或节能模式可能需要时间。因此，一定要通过执行前读取寄存器值来检查模式是否被成功更改。

备注 在 CAN 休眠模式和 CAN 停止模式中，OPMODE0 到 OPMODE2 位是只读的。

(b) 写入

设定 CCERC	CCERC 位的设定
1	CCERC 位置 1
其它	CCERC 位保持不变

设定 AL	清除 AL	AL 位的设定
0	1	AL 位清 0
1	0	AL 位置 1
其它		AL 位保持不变

清除 VALID	VALID 位的设定
0	VALID 位保持不变
1	VALID 位清 0

设置 PSMODE0	清除 PSMODE0	PSMODE0 位的设定
0	1	PSMODE0 位清 0
1	0	PSMODE 位置 1
其它		PSMODE0 位保持不变

设置 PSMODE1	清除 PSMODE1	PSMODE1 位的设定
0	1	PSMODE1 位清 0
1	0	PSMODE1 位置 1
其它		PSMODE1 位保持不变

设置 OPMODE0	清除 OPMODE0	OPMODE0 位的设定
0	1	OPMODE0 位清 0
1	0	OPMODE0 位置 1
其它		OPMODE0 位保持不变

设置 OPMODE1	清除 OPMODE1	OPMODE1 位的设定
0	1	OPMODE1 位清 0。
1	0	OPMODE1 位置 1
其它		OPMODE1 位保持不变

设置 OPMODE2	清除 OPMODE2	OPMODE2 位的设定
0	1	OPMODE2 位清 0
1	0	OPMODE2 位置 1
其它		OPMODE2 位保持不变

(7) CAN0 模块最终错误信息寄存器 (C0LEC)

C0LEC 寄存器提供 CAN 协议的错误信息。

复位后的值: 00H R/W 地址: 03FEC052H

	7	6	5	4	3	2	1	0
C0LEC	0	0	0	0	0	LEC2	LEC1	LEC0

- 备注**
1. 当 CAN 模块从操作模式转换到初始化模式后，C0LEC 寄存器的内容不会被清除。
 2. 如果试图通过软件对 C0LEC 寄存器写入除 00H 外的其它值时，该访问将被忽略。

LEC2	LEC1	LEC0	最终的 CAN 协议错误信息
0	0	0	无错误
0	0	1	填充错误
0	1	0	格式错误
0	1	1	ACK 错误
1	0	0	位错误位错误 (CAN 模块试图发送一个隐性电平位作为发送信息的一部分 (除去仲裁字段)，但是 CAN 总线的值是显性电平位。
1	0	1	位错误位错误 (CAN 模块试图发送一个显性电平位作为发送信息的一部分、ACK 位、错误帧或是过载帧，但是 CAN 总线的值是隐性电平位。
1	1	0	CRC 错误
1	1	1	未定义

(8) CAN0 模块信息寄存器 (C0INFO)

C0INFO 寄存器指出了 CAN 模块的状态。

复位后的值: 00H R 地址: 03FEC053H

	7	6	5	4	3	2	1	0
C0INFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0

BOFF	bus-off 状态位
0	非总线关闭状态 (发送错误计数器 ≤ 255) . (发送错误计数器的值小于 256.)
1	总线关闭状态 (发送错误计数器 >255) . (发送错误计数器的值大于或是等于 256)

TECS1	TECS0	发送错误计数器状态位
0	0	发送错误计数器的值小于警戒水平 (< 96) .
0	1	发送错误计数器的值在警戒范围之内 (96 到 127) .
1	0	未定义
1	1	发送错误计数器的值在错误无效状态或是 bus-off 状态范围内 (≥ 128) .

RECS1	RECS0	接收错误计数器状态位
0	0	接收错误计数器的值小于警戒水平 (< 96)
0	1	接收发送错误计数器的值在警戒范围之内 (96 到 127) .
1	0	未定义
1	1	接收错误计数器的值在错误无效状态 范围内 (≥ 128) .

(9) CAN0 模块错误计数器寄存器 (C0ERC)

C0ERC 寄存器指示了发送 / 接收错误计数器的计数值。

复位后的值: 0000H R 地址: 03FEC054H

	15	14	13	12	11	10	9	8
C0ERC	REPS	REC6	REC5	REC4	REC3	REC2	REC1	REC0
	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0

REPS	接收错误被动状态位
0	接收错误计数器的值不处于错误无效范围 (< 128)
1	接收错误计数器的值处于错误有效范围 (≥ 128)

REC6 到 REC0	接收错误计数器位
0 到 127	指示接收错误的次数。这些位反映了接收错误计数器的状态。这些错误的数量是由 CAN 协议定义的。

备注 在接收错误无效状态中，接收错误计数器的 REC6 到 REC0 位是无效的。(C0INFO.RECS1, C0INFO.RECS0 位为 11B)。

TEC7 到 TEC0	发送错误计数器位
0 到 255	指示发送错误的次数。这些位反映了发送错误计数器的状态。这些错误的数量是由 CAN 协议定义的。

备注 在 bus-off 状态中，发送错误计数器的 TEC7 到 TEC0 位是无效的。(CnINFO.BOFF 位为 1)。

(10) CAN0 模块中断允许寄存器 (C0IE)

C0IE 寄存器用于允许或是禁止 CAN 模块的中断。

(1/2)

复位后的值: 0000H R/W 地址: 03FEC056H

(a) 读取

	15	14	13	12	11	10	9	8
C0IE	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0

(b) 写入

	15	14	13	12	11	10	9	8
C0IE	0	0	设置	设置	设置	设置	设置	设置
			CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
	7	6	5	4	3	2	1	0
	0	0	清除	清除	清除	清除	清除	清除
			CIE5	CIE4	CIE3	CIE2	CIE1	CIE0

(a) 读取

CIE5 到 CIE0	CAN 模块中断使能位
0	禁止向相应的中断状态寄存器 CINTSx 输出中断。
1	允许向相应的中断状态寄存器 CINTSx 输出中断。

(b) 写入

设置 CIE5	清除 CIE5	CIE5 位的设定
0	1	CIE5 位清 0
1	0	CIE5 位置 1
其它		CIE5 位保持不变

设置 CIE4	清除 CIE4	CIE4 位的设定
0	1	CIE4 位清 0
1	0	CIE4 位置 1
其它		CIE4 位保持不变

设置 CIE3	清除 CIE3	CIE3 位的设定
0	1	CIE3 位清 0
1	0	CIE3 位置 1
其它		CIE3 位保持不变

设置 CIE2	清除 CIE2	CIE2 位的设定
0	1	CIE2 位清 0
1	0	CIE2 位置 1
其它		CIE2 位保持不变

设置 CIE1	清除 CIE1	CIE1 位的设定
0	1	CIE1 位清 0
1	0	CIE1 位置 1
其它		CIE1 位保持不变

设置 CIE0	清除 CIE0	CIE0 位的设定
0	1	CIE0 位清 0
1	0	CIE0 位置 1
其它		CIE0 位保持不变

(11) CAN0 模块中断状态寄存器 (C0INTS)

Cn0INTS 寄存器指示了 CAN 模块的中断状态。

复位后的值: 0000H R/W 地址: 03FEC058H

(a) 读取

	15	14	13	12	11	10	9	8
C0INTS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0

(b) 写入

	15	14	13	12	11	10	9	8
C0INTS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	清除 CINTS5	清除 CINTS4	清除 CINTS3	清除 CINTS2	清除 CINTS1	清除 CINTS0

(a) 读取

CINTS5 到 CINTS0	CAN 中断状态位
0	没有相关的中断源事件发生。
1	有相关的中断源事件发生。

中断状态位	相关的中断源事件
CINTS5	唤醒 CAN 休眠模式注的中断 ^注
CINTS4	仲裁失败中断
CINTS3	CAN 协议错误中断
CINTS2	CAN 错误状态中断
CINTS1	信息缓存 m 完成有效信息帧接收的中断
CINTS0	信息缓存 m 完成发送信息帧的中断

注 CINTS5 位仅当通过 CAN 总线操作使得 CAN 模块从休眠状态中唤醒时被置 1。CINTS5 位在通过软件释放 CAN 休眠模式时不会被置 1。

(b) 写入

清除 CINTS5 到 CINTS0	CINTS5 到 CINTS0 位的设定
0	CINTS5 到 CINTS0 位保持不变
1	CINTS5 到 CINTS0 位被清 0

<R> **注意事项** 这个寄存器的状态位不会被自动清除。在中断服务中，如果必须检查每个状态，通过软件清零。

(12) CAN0 模块位速率预设寄存器 (C0BRP)

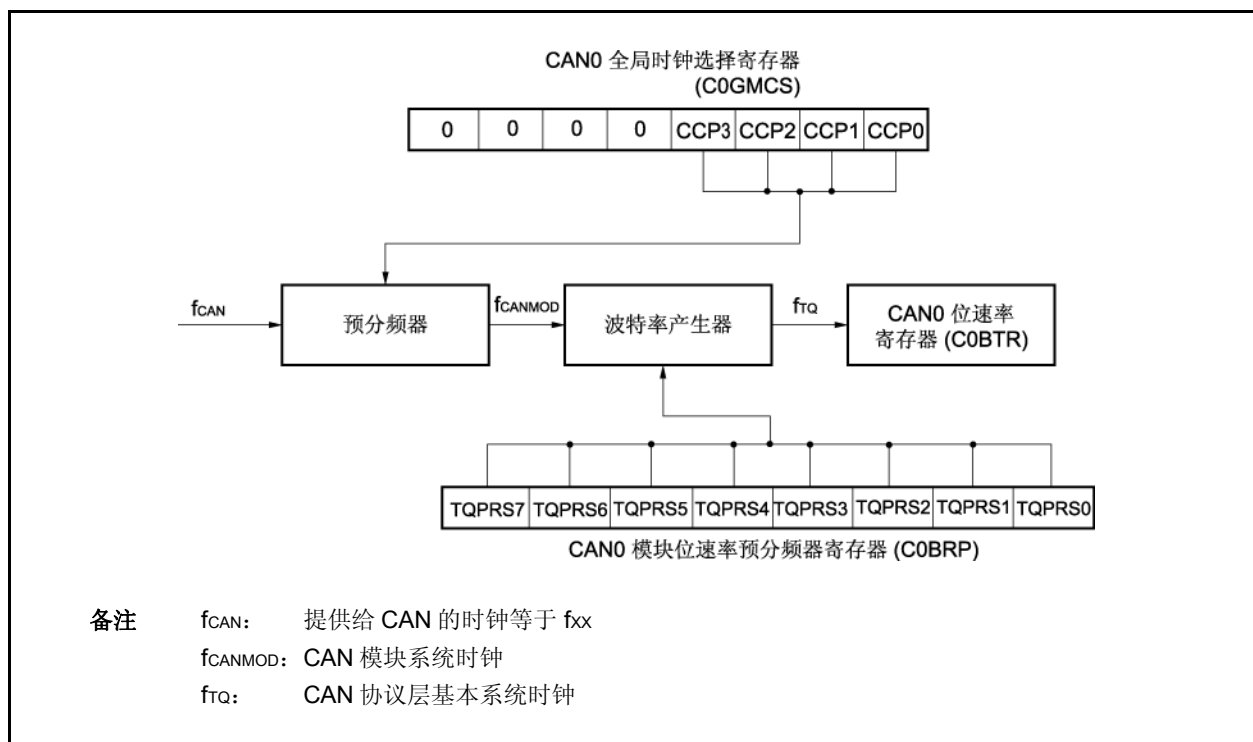
C0BRP 寄存器用于选择 CAN 协议层的基本时钟 (f_{TQ})。通信的波特率在 C0BTR 寄存器中设定。

复位后的值: FFH R/W 地址: 03FEC05AH

	7	6	5	4	3	2	1	0
C0BRP	TQPRS7	TQPRS6	TQPRS5	TQPRS4	TQPRS3	TQPRS2	TQPRS1	TQPRS0

TQPRS7 到 TQPRS0	CAN 协议层基本系统时钟 (f _{TQ})
0	f _{CANMOD} /1
1	f _{CANMOD} /2
n	f _{CANMOD} / (n+1)
:	:
255	f _{CANMOD} /256 (默认值)

图 19-23. CAN 模块时钟

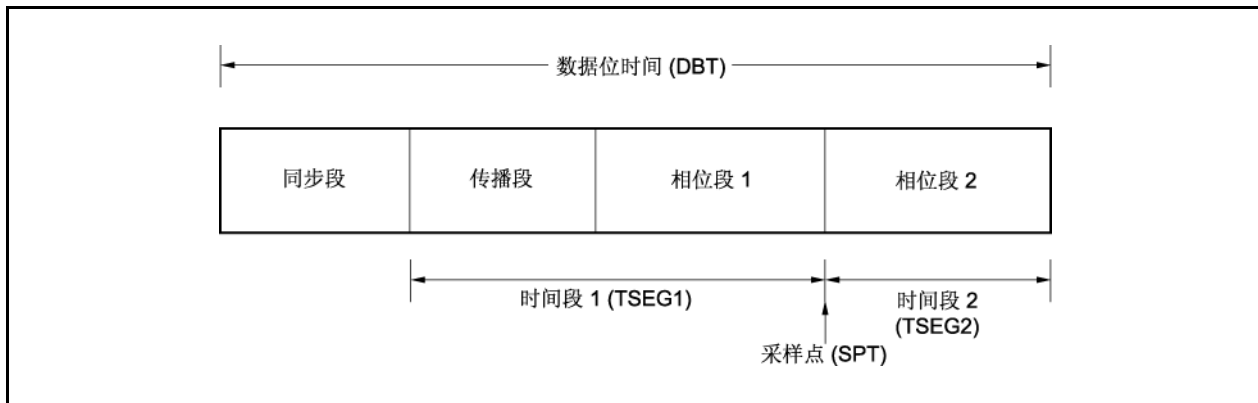


注意事项 C0BRP 寄存器在初始化模式下只能执行写入访问。

(13) CAN0 模块位速率寄存器 (C0BTR)

C0BTR 寄存器用于控制通信波特率数据位时间。

图 19-24. 数据位时间



复位后的值: 370FH R/W 地址: 03FEC05CH

	15	14	13	12	11	10	9	8
C0BTR	0	0	SJW1	SJW0	0	TSEG22	TSEG21	TSEG20
	7	6	5	4	3	2	1	0
	0	0	0	0	TSEG13	TSEG12	TSEG11	TSEG10

SJW1	SJW0	同步跳转宽度
0	0	1TQ
0	1	2TQ
1	0	3TQ
1	1	4TQ (默认值)

TSEG22	TSEG21	TSEG20	时间段 2 的长度
0	0	0	1TQ
0	0	1	2TQ
0	1	0	3TQ
0	1	1	4TQ
1	0	0	5TQ
1	0	1	6TQ
1	1	0	7TQ
1	1	1	8TQ (默认值)

TSEG13	TSEG12	TSEG11	TSEG10	时间段 1 的长度
0	0	0	0	禁止设定
0	0	0	1	2TQ [※]
0	0	1	0	3TQ [※]
0	0	1	1	4TQ
0	1	0	0	5TQ
0	1	0	1	6TQ
0	1	1	0	7TQ
0	1	1	1	8TQ
1	0	0	0	9TQ
1	0	0	1	10TQ
1	0	1	0	11TQ
1	0	1	1	12TQ
1	1	0	0	13TQ
1	1	0	1	14TQ
1	1	1	0	15TQ
1	1	1	1	16TQ (默认值)

注 在 C0BRP 寄存器的值为 00H 时, 不允许设置。

备注 TQ = 1/ftQ (ftQ: CAN 协议层基本系统时钟)

(14) CAN0 模块最终接收指针寄存器 (C0LIPT)

C0LIPT 寄存器表示最后用于存储数据帧或是远程帧的信息缓存编号。

复位后的值：不确定 R 地址： 03FEC05EH

	7	6	5	4	3	2	1	0
C0LIPT	LIPT7	LIPT6	LIPT5	LIPT4	LIPT3	LIPT2	LIPT1	LIPT0

LIPT7 到 LIPT0	最终接收指针寄存器 (C0LIPT)
0 到 31	当对 C0LIPT 进行读取操作时，读出的是接收历史列表的最终进入指针 (LIPT) 中编址的内容。这些表示最后使用的用于存储数据帧或是远程帧信息的缓存编号。

备注 如果信息缓存中从未存储过数据帧或是远程帧，则从 C0LIPT 寄存器中读出的值是不确定的。如果 CAN 模块从初始化模式转换到操作模式后 C0RGPT.RHPM 位被置 1，那么从 C0LIPT 寄存器中读出的值是不确定的。

(15) CAN0 模块接收历史列表寄存器 (C0RGPT)

C0RGPT 用于读取接收历史列表。

复位后的值: xx02H R/W 地址: 03FEC060H

(a)

读取

	15	14	13	12	11	10	9	8
C0RGPT	RGPT7	RGPT6	RGPT5	RGPT4	RGPT3	RGPT2	RGPT1	RGPT0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	RHPM	ROVF

(b)

写入

	15	14	13	12	11	10	9	8
C0RGPT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	清除 ROVF

(a) 读取

RGPT7 到 RGPT0	接收历史列表读取指针
0 到 31	当对 C0RGPT 寄存器进行读取操作时, 读出的是被接收历史列表中接收历史列表获取指针 (RGPT) 编址的元素内容。这些内容指出了已经存储了数据帧和远程帧的信息缓存的编号。

RHPM ^{※1}	接收历史列表指针匹配
0	接收历史列表至少还有一个信息缓存编号没有被读取。
1	接收历史列表的信息缓存编号已经全部被读取。

ROVF ^{※2}	接收历史列表溢出位
0	保存所有未被读取的信息缓存寄存器的地址。所有的用于接收和保存新数据帧和远程帧的信息缓存编号会被记录在接收历史列表中 (接收历史列表中含有空元素)。
1	在 CPU 最后一次访问 RHL 后至少已保存 23 个条目 (例如 读取 C0RGPT)。前 22 个条目按顺序存储, 然而最后一个条目在很多情况下会被最新发送的信息覆盖, 因为当 TOVF 位置 1 时所有的缓存编号保存在 LOPT-1 的位置。所以接收不能按照它们被接收的顺序完全恢复。

- 注
1. 当 RHPM 位置 1 时, 从 RGPT0 到 RGPT7 位读取的值是无效的。
 2. 当 ROVF 位被置位时, 如果所有接收历史被 C0RGPT 寄存器读取, 即使存储了新接收的数据, RHPM 位也不被清零而是保持置位。

(b) 写入

清除 ROVF	ROVF 位的设定
0	ROVF 位保持不变
1	ROVF 位清 0

(16) CAN0 模块最终输出指针寄存器 (C0LOPT)

C0LOPT 寄存器指出了最后用于发送数据帧或远程帧的信息缓存的编号。

复位后的值：不确定 R 地址： 03FEC062H

	7	6	5	4	3	2	1	0
C0LOPT	LOPT7	LOPT6	LOPT5	LOPT4	LOPT3	LOPT2	LOPT1	LOPT0

LOPT7 到 LOPT0	发送历史列表的最终输出指针 (LOPT)
0 到 31	当对 C0LOPT 寄存器进行读操作时，读取的内容是被接收历史列表的最终输出指针 (LOPT) 编址的元素。这些内容指出了最后发送数据帧或远程帧的信息缓存的编号地址。

备注 当信息缓存从来没有发送过数据帧和远程帧时，从 C0LOPT 寄存器中读取的值是不确定的。如果 CAN 模块从初始化模式转换到操作模式后 C0TGPT.THPM 位被置 1，那么从 C0LOPT 寄存器中读出的值是不确定的。

(17) CAN0 模块发送历史列表寄存器 (C0TGPT)

C0TGPT 寄存器用于读取发送历史列表。

复位后的值: xx02H R/W 地址: 03FEC064H

(a) 读取

	15	14	13	12	11	10	9	8
C0TGPT	TGPT7	TGPT6	TGPT5	TGPT4	TGPT3	TGPT2	TGPT1	TGPT0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	THPM	TOVF

(b) 写入

	15	14	13	12	11	10	9	8
C0TGPT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	清除 TOVF

(a) 读取

TGPT7 到 TGPT0	传送历史列表读取指针
0 到 31	当对 C0TGPT 寄存器进行读取时，读出的内容是传送历史列表的读取指针 (TGPT) 记录的传送缓冲寄存器的地址。这些内容指出了最后发送数据帧或远程帧的信息缓存的编号地址。

THPM ^{注1}	传送历史指针匹配
0	传送历史列表中至少有一个信息缓存寄存器未被读取。
1	传送历史列表中的所有信息缓存寄存器都已经被读取。

TOVF ^{注2}	传送历史列表溢出位
0	保存所有未被读取的信息缓存寄存器的地址。传送历史列表记录所有发送过新的数据帧和远程帧的信息缓存寄存器的地址 (传送历史列表拥有一个空元素)。
1	当 CPU 最后一次对 THL 进行访问之后可以保存 7 个条目 (例如读取 C0TGPT)。前 6 个条目按照顺序存储, 然而最后一个条目如果没有及时读取, 在这种情况下会被最新传送的信息覆盖, 因为当 TOVF 位被置 1 时所有新传送的缓存寄存器的地址保存在 LOPT-1 的位置。所以接收的结果并不能完全反映接收的顺序。

- 注
1. 当 THPM 位置 1 时, 从 TGPT0 到 TGPT7 位读出的值是无效的。
 2. 当 TOVF 位被置位时, 如果所有发送历史被 C0TGPT 寄存器读取, 即使完成了新的数据发送, THPM 位也不被清零而是被保持置位。

备注 在带有 ABT 的正常操作模式中, 从信息缓存 0 到 7 进行的发送操作不会被记录到传送历史列表中。

(b) 写入

清除 TOVF	TOVF 位的设定
0	TOVF 位保持不变
1	TOVF 位清 0

(18) CAN0 模块时间标记寄存器 (C0TS)

C0TS 寄存器用于控制时间标记功能。

(1/2)

复位后的值: 0000H R/W 地址: 03FEC066H

(a) 读取

	15	14	13	12	11	10	9	8
C0TS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	TSLOCK	TSSEL	TSEN

(b) 写入

	15	14	13	12	11	10	9	8
C0TS	0	0	0	0	0	设置 TSLOCK	设置 TSSEL	设置 TSEN
	7	6	5	4	3	2	1	0
	0	0	0	0	0	清除 TSLOCK	清除 TSSEL	清除 TSEN

备注 当 CAN 模块处于带有 ABT 的正常操作模式时, 不能使用时间标记的锁定功能。

(a) 读取

TSLOCK	时间标记功能允许位
0	禁止时间标记锁定功能 当选定的时间标记捕获事件发生时, 锁定 TSOUT 信号。
1	使能时间标记锁定功能 当选定的时间标记捕获事件发生时, 锁定 TSOUT 信号。 但是, 当数据帧正确的接收到信息缓存 0 时, 锁定 TSOUT 输出信号。

注 TSEN 位会自动清 0。

TSSEL	时间标记捕获事件选择位
0	SOF 为时间标记捕获所用事件。
1	EOF 的最后一位为时间标记捕获所用事件。

TSEN	TSOUT 的操作设定位
0	禁止 TSOUT 固定波形输出操作
1	允许 TSOUT 固定波形输出操作。

备注 TSOUT 信号从 CAN 控制器输出到一个定时器。关于细节, 参阅 第 7 章 16 位定时器 / 事件计数器 (TMP) 。

(b) 写入

设定 TSLOCK	清除 TSLOCK	TSLOCK 位的设定
0	1	TSLOCK 位清 0
1	0	TSLOCK 位置 1
其它		TSLOCK 位保持不变

设定 TSSEL	清除 TSSEL	TSSEL 位的设定
0	1	TSSEL 位清 0
1	0	TSSEL 位置 1
其它		TSSEL 位保持不变

设定 TSEN	清除 TSEN	TSEN 位的设定
0	1	TSEN 位清 0
1	0	TSEN 位置 1
其它		TSEN 位保持不变

(19) CAN0 信息数据字节寄存器 (C0MDATAxm, C0MDATAym) (x = 0 到 7, y = 01, 23, 45, 67)

C0MDATAxm 寄存器被用来存储发送 / 接收信息的数据, 并且可以以 8 位为单位被访问。

C0MDATAxm 寄存器可以通过 C0MDATAym 寄存器以 16 位为单位被访问。

(1/2)

复位后的值: 不确定 R/W 地址: 见表 19-16.

C0MDATA01m	15	14	13	12	11	10	9	8
	MDATA01	MDATA01	MDATA01	MDATA01	MDATA01	MDATA01	MDATA01	MDATA01
	15	14	13	12	11	10	9	8
	7	6	5	4	3	2	1	0
	MDATA01	MDATA01	MDATA01	MDATA01	MDATA01	MDATA01	MDATA01	MDATA01
	7	6	5	4	3	2	1	0
C0MDATA0m	7	6	5	4	3	2	1	0
	MDATA0	MDATA0	MDATA0	MDATA0	MDATA0	MDATA0	MDATA0	MDATA0
	7	6	5	4	3	2	1	0
C0MDATA1m	7	6	5	4	3	2	1	0
	MDATA1	MDATA1	MDATA1	MDATA1	MDATA1	MDATA1	MDATA1	MDATA1
	7	6	5	4	3	2	1	0
C0MDATA23m	15	14	13	12	11	10	9	8
	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23
	15	14	13	12	11	10	9	8
	7	6	5	4	3	2	1	0
	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23
	7	6	5	4	3	2	1	0
C0MDATA2m	7	6	5	4	3	2	1	0
	MDATA2	MDATA2	MDATA2	MDATA2	MDATA2	MDATA2	MDATA2	MDATA2
	7	6	5	4	3	2	1	0
C0MDATA3m	7	6	5	4	3	2	1	0
	MDATA3	MDATA3	MDATA3	MDATA3	MDATA3	MDATA3	MDATA3	MDATA3
	7	6	5	4	3	2	1	0

(2/2)

C0MDATA45m	15	14	13	12	11	10	9	8
	MDATA45 15	MDATA45 14	MDATA45 13	MDATA45 12	MDATA45 11	MDATA45 10	MDATA45 9	MDATA45 8
	7	6	5	4	3	2	1	0
	MDATA45 7	MDATA45 6	MDATA45 5	MDATA45 4	MDATA45 3	MDATA45 2	MDATA45 1	MDATA45 0
C0MDATA4m	7	6	5	4	3	2	1	0
	MDATA4 7	MDATA4 6	MDATA4 5	MDATA4 4	MDATA4 3	MDATA4 2	MDATA4 1	MDATA4 0
C0MDATA5m	7	6	5	4	3	2	1	0
	MDATA5 7	MDATA5 6	MDATA5 5	MDATA5 4	MDATA5 3	MDATA5 2	MDATA5 1	MDATA5 0
C0MDATA67m	15	14	13	12	11	10	9	8
	MDATA67 15	MDATA67 14	MDATA67 13	MDATA67 12	MDATA67 11	MDATA67 10	MDATA67 9	MDATA67 8
	7	6	5	4	3	2	1	0
	MDATA67 7	MDATA67 6	MDATA67 5	MDATA67 4	MDATA67 3	MDATA67 2	MDATA67 1	MDATA67 0
C0MDATA6m	7	6	5	4	3	2	1	0
	MDATA6 7	MDATA6 6	MDATA6 5	MDATA6 4	MDATA6 3	MDATA6 2	MDATA6 1	MDATA6 0
C0MDATA7m	7	6	5	4	3	2	1	0
	MDATA7 7	MDATA7 6	MDATA7 5	MDATA7 4	MDATA7 3	MDATA7 2	MDATA7 1	MDATA7 0

(20) CAN0 信息数据长度寄存器 m (C0MDLCm)

C0MDLCm 寄存器用于设置信息缓存的数据字段的字节数。

复位后的值: 0000xxxxB R/W 地址: 见表 19-16.

	7	6	5	4	3	2	1	0
C0MDLCm	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0

MDLC3	MDLC2	MDLC1	MDLC0	发送 / 接收信息的数据长度
0	0	0	0	0 字节
0	0	0	1	1 字节
0	0	1	0	2 字节
0	0	1	1	3 字节
0	1	0	0	4 字节
0	1	0	1	5 字节
0	1	1	0	6 字节
0	1	1	1	7 字节
1	0	0	0	8 字节
1	0	0	1	禁止设定 (如果在发送过程中设置这些位, 当发送数据帧时会忽略 DLC 的设定值直接发送 8 字节数据。但是, 实际传输到 CAN 总线的是在该寄存器中设定的 DLC 的数值。) ^注
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

注 发送到 CAN 总线的数据和 DLC 实际值如下所示。

发送帧的类型	发送数据的长度	发送的 DLC
数据帧	DLC 指定的字节数 (然而, 如果 $DLC \geq 8$, 8 字节)	MDLC3 到 MDLC0 位
远程帧	0 字节	

注意事项 1. 一定要将位 7 到 4 设定为 0000B。

2. 接收数据被存储在由相应的 DLC 字节数设定的 C0MDATAxm 寄存器中 (上限为 8)。没有存储数据的 C0MDATAxm 寄存器是不确定的。

(21) CAN0 信息配置寄存器 m (C0MCONFm)

C0MCONFm 寄存器用于指定信息缓存的类型和设定屏蔽。

(1/2)

复位后的值：不确定 R/W 地址：见表 19-16.

	7	6	5	4	3	2	1	0
C0MCONFm	OVS	RTR	MT2	MT1	MT0	0	0	MA0

OVS	重写控制位
0	已经接收数据帧的信息缓存 ^注 不会被新接收到的数据帧改写。新的数据帧将被丢弃。
1	已经接收数据帧的信息缓存会被新接收到的数据帧改写。

注 “已经接收数据帧的信息缓存”是 C0MCTRLm.DN 位已经被置 1 的接收信息缓存

备注 接收和存储远程帧时，忽略 OVS 和 DN 位的设置。满足其它条件的远程帧（ID 匹配，RTR 位为 0，C0MCTRLm.TRQ 位为 0）永远被接收和存储在相应的信息缓存中（发生中断，DN 标志置位，C0MDLCm.MDLC0 到 C0MDLCm.MDLC3 位更新，和记录接收历史列表）。

RTR	远程帧请求位 ^注
0	发送数据帧。
1	发送远程帧

注 RTR 位指定了由定义好的发送信息缓存发送的信息帧的类型。即使收到一个有效的远程帧，该发送信息缓存的 RTR 位依然被清 0。即使从 CAN 总线上收到一个 ID 匹配的远程帧，且发送信息缓存的 RTR 位置 1 以便发送远程帧，那么也不会接收和存储该远程帧（产生中断，设置 DN 标志，MDLC0 到 MDLC3 位更新，记录到接收信息历史列表）。

MT2	MT1	MT0	信息缓存类型设定位
0	0	0	发送信息缓存
0	0	1	接收信息缓存（没有设定屏蔽）
0	1	0	接收信息缓存（设定屏蔽 1）
0	1	1	接收信息缓存（设定屏蔽 2）
1	0	0	接收信息缓存（设定屏蔽 3）
1	0	1	接收信息缓存（设定屏蔽 4）
其它			禁止设定

MA0	信息缓存分配位
0	不使用信息缓存.
1	使用信息缓存.

注意事项 一定要将位 2 和位 1 清 0。

(22) CAN0 信息 ID 寄存器 m (C0MIDLm, C0MIDHm)

C0MIDLm 和 C0MIDHm 寄存器用于设定标识符 (ID)。

复位后的值: 不确定 R/W 地址: 见表 19-16.

	15	14	13	12	11	10	9	8
C0MIDLm	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
	7	6	5	4	3	2	1	0
	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

	15	14	13	12	11	10	9	8
C0MIDHm	IDE	0	0	ID28	ID27	ID26	ID25	ID24
	7	6	5	4	3	2	1	0
	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

IDE	格式模式指定位
0	标准格式模式 (ID28 到 ID18: 11 位) [※]
1	扩展格式模式 (ID28 到 ID0: 29 位)

注 不使用 ID17 到 ID0 位。

ID28 到 ID0	信息 ID
ID28 到 ID18	11 位标准 ID 值 (当 IDE = 0 时)
ID28 到 ID0	29 位扩展 ID 值 (当 IDE = 1 时)

注意事项 1 一定要将 C0MIDHm 寄存器的位 14 和位 13 清 0。

2. 一定要按照这个寄存器位的位置来安排要被登记的 ID 值。对于标准 ID, 将 ID 值的 ID28 移位到 ID18。

<R>

(23) CAN0 信息控制寄存器 m (C0MCTRLm)

C0MCTRLm 寄存器用于控制信息缓存的操作。

(1/3)

复位后的值: R/W 地址: 见表 19-16.

00x000000

000xx000B

(a) 读取

	15	14	13	12	11	10	9	8
C0MCTRLm	0	0	MUC	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	MOW	IE	DN	TRQ	RDY

(b) 写入

	15	14	13	12	11	10	9	8
C0MCTRLm	0	0	0	0	设置 IE	0	设置 TRQ	设置 RDY
	7	6	5	4	3	2	1	0
	0	0	0	清除 MOW	清除 IE	清除 DN	清除 TRQ	清除 RDY

(a) 读取

MUC ^注	信息缓存数据更新指示位
0	CAN 模块未更新信息缓存（接收和存储）。
1	CAN 模块已经更新信息缓存（接收和存储）。

注 在第一次执行接收和存储之前，MUC 位的值是不确定的。

MOW	信息缓存覆盖状态位
0	信息缓存没有被新接收到数据帧覆盖。
1	信息缓存已经被新接收到数据帧覆盖。

备注 即使接收到远程帧、将其存储到发送信息缓存且将 DN 置 1，MOW 位也不会置 1。

IE	信息缓存中断请求允许位
0	接收信息缓存：禁止在有效信息接收完成后产生中断。 发送信息缓存：禁止在正常信息发送完成后产生中断。
1	接收信息缓存：允许在有效信息接收完成后产生中断。 发送信息缓存：允许在正常信息发送完成后产生中断。

DN	信息缓存数据更新位
0	在信息缓存中未存储数据帧和远程帧。
1	在信息缓存中存储数据帧和远程帧。

TRQ	信息缓存发送请求位
0	在信息缓存中没有未处理或是正在发送的信息帧发送请求。
1	信息缓存挂起未处理的信息帧或是正在发送信息帧。

注意事项 不能同时将 TRQ 位和 RDY 位置 1。在 TRQ 置 1 前一定要先将 RDY 位置 1。

RDY	信息缓存就绪位
0	可以通过软件设置信息缓存。CAN 模块不能写入信息缓存。
1	不能通过软件设置信息缓存（除非是对 RDY，TRQ，DN，和 MOW 位的写入操作）。CAN 模块可以写入信息缓存。

注意事项 1. 在信息发送期间不能将 RDY 位清 0。后续的发送中止程序用于清除 RDY 位以便进行重新定义。

2. 即使执行了清除 RDY 位的处理，如果 RDY 位没有被清零，再次执行清除处理。

3. 在写入数据到信息缓存前，通过再次读取 RDY 位，确认 RDY 位已经被清零。

然而，不需要确认 TRQ 或 RDY 位已经被置位或者 DN 或 MOW 位已经被清零。

(b) 写入

清除 MOW	MOW 位设定
0	MOW 位保持不变
1	MOW 位清 0

设置 IE	清除 IE	IE 位设定
0	1	IE 位清 0
1	0	IE 位置 1
其它		IE 位保持不变

注意事项 一定要分别设置 IE 和 RDY 位。

清除 DN	DN 位设定
1	DN 位清 0
0	DN 位保持不变

注意事项 不要通过软件将 DN 位置 1。一定要将位 10 清 0。

设置 TRQ	清除 TRQ	TRQ 位设定
0	1	TRQ 位清 0
1	0	TRQ 位置 1
其它		TRQ 位保持不变

<R>

注意事项 即使 **TRQ** 位被置位，要看当前的情况，例如当信息从另一个节点被接收时或当信息从信息缓存被发送时时，发送也不会立即执行，。

如果 **TRQ** 位被清除，正在执行的发送不会被终止。发送会继续，直到它完成（不考虑它的执行成功或失败）。

设置 RDY	清除 RDY	RDY 位设定
0	1	RDY 位清 0
1	0	RDY 位置 1
其它		RDY 位保持不变

<R>

注意事项 一定要分别设置 **TRQ** 和 **RDY** 位。

19.7 位设置 / 清除功能

CAN 控制寄存器包含了很多寄存器，可以通过 CPU 或是 CAN 接口将它们当中的所有标志位设定或清除。如果直接对以下寄存器进行写入操作，会产生一个操作错误。不要直接通过位操作、读取 / 修改 / 写入或直接写目标数值来写入任何数据。

- CAN0 全局控制寄存器 (C0GMCTRL)
- CAN0 全局自动模块发送控制寄存器 (C0GMABT)
- CAN0 模块控制寄存器 (C0CTRL)
- CANn 模块中断允许寄存器 (C0IE)
- CAN0 模块中断状态寄存器 (C0INTS)
- CAN0 模块接收历史列表寄存器 (C0RGPT)
- CAN0 模块发送历史列表寄存器 (C0TGPT)
- CAN0 模块时间标记寄存器 (C0TS)
- CAN0 信息控制寄存器 m (C0MCTRLm)

备注 m = 00 到 31

上述寄存器的所有 16 位都可以通过通常方式读取。应用图 19-25 所描述的流程可以设置或是清除这些寄存器的低 8 位。

设置或是清除上述寄存器的低 8 位需要联合高 8 位共同执行（参见图 19-26 中指定的在设置/清除操作后的位状态）。图 19-25 显示了在相应的寄存器中涉及到设定 / 清除 / 无变化操作的设置位或清除位的值。

图 19-25. 位设置 / 清除操作举例

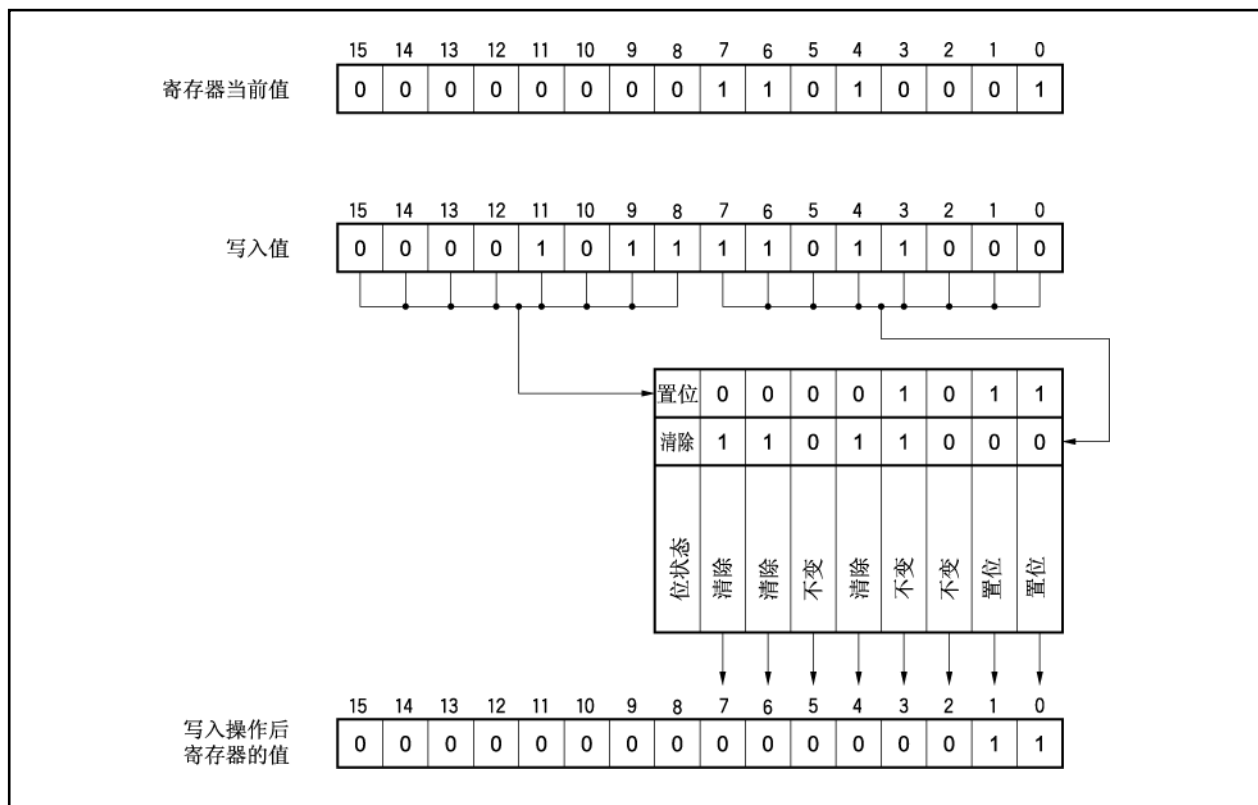


图 19-26. 进行位设置 / 清除操作后的位状态

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
置位7	置位6	置位5	置位4	置位3	置位2	置位1	置位0	清除7	清除6	清除5	清除4	清除3	清除2	清除1	清除0

置位n	清除n	位置位/清除操作后位n的状态
0	0	不变0
0	1	1
1	0	不变
1	1	

备注 n = 0 到 7

19.8 CAN控制器初始化

19.8.1 CAN控制器初始化

在使能 CAN 模块的操作之前，CAN 模块系统时钟需要通过软件设定 C0GMCS.CCP0 到 C0GMCS.CCP3 位来设置。在使能 CAN 模块操作之后不要再改变 CAN 模块系统时钟的设定。

通过设置 C0GMCTRL.GOM 位来使能 CAN 模块。

CAN 模块初始化的程序见 19.16 CAN 控制器的操作。

19.8.2 信息缓存的初始化

在 CAN 模块被激活之后，信息缓存的内容是不确定的值。在 CAN 模块从初始化模式转换到其它操作模式之前，必须对所有的，即使是那些在应用中不使用的信息缓存进行最小初始化。

- 清除 the C0MCTRLm.RDY, C00MCTRLm.TRQ 和 C0MCTRLm.DN 位为 0。
- 清除 the C0MCONFm.MA0 位为 0。

备注 m = 00 到 31

19.8.3 信息缓存的重定义

对信息缓存的重定义意味着在发送或是接收信息的期间更改信息缓存的 ID 和控制信息，而不影响其它的发送或是接收操作。

(1) 在初始化模式中执行信息缓存的重定义

将 CAN 模块置于初始化模式一次，然后在初始化模式中改变 ID 和信息缓冲器控制信息。完成以上工作之后，将 CAN 模块设定为操作模式。

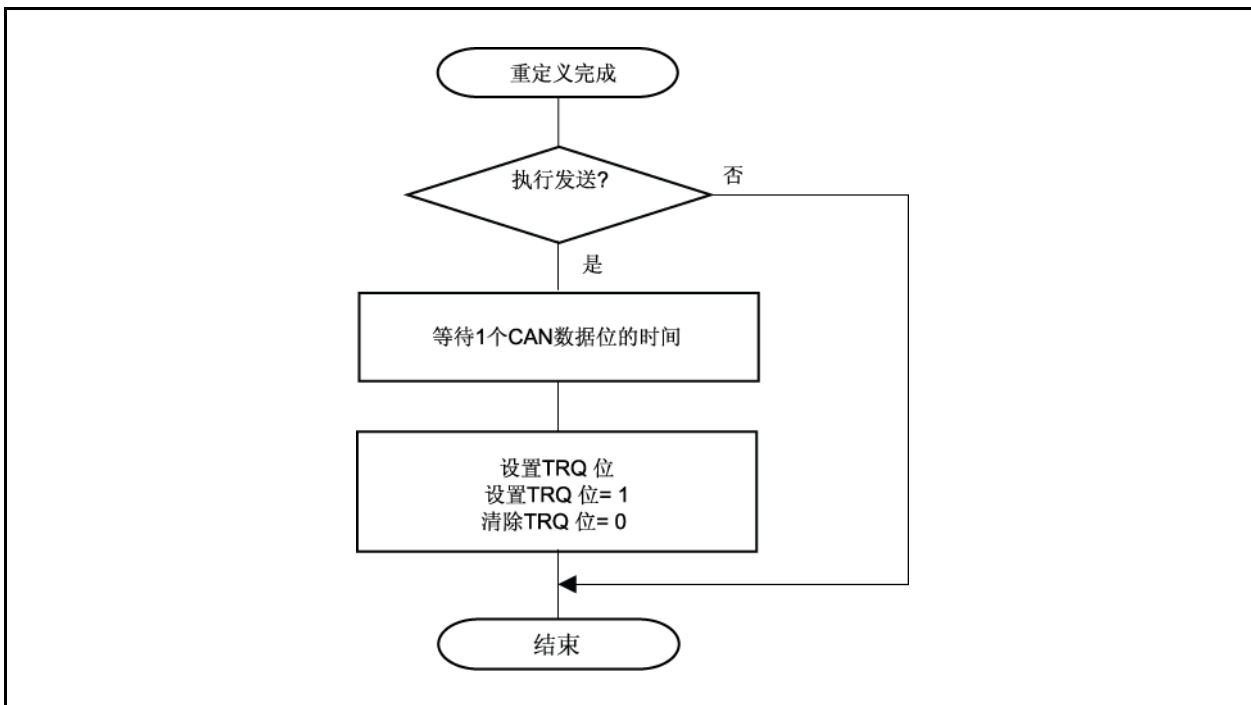
(2) 在接收过程中执行信息缓存的重定义

执行过程在图 19-39 显示。

(3) 在发送过程中执行信息缓存的重定义

在对设定了发送请求的发送信息缓存的内容进行重新写入时，会执行发送中断处理（见 19.10.4 (1) 在带有自动块发送 ABT 的正常操作模式以外的发送中止流程，19.10.4 (2) 在带有自动块发送 ABT 的正常操作模式中除去 ABT 传输的发送中止流程）。确认发送完成或是失败，然后重新定义信息缓存。完成这些后，按下面描述的过程设定发送请求。但是当对一个在运行中被重定义而没有中断发送的信息缓存设置发送请求时，不需要 1 位等待时间。

图 19-27. 在重定义后对发送信息缓存设定发送请求 (TRQ)



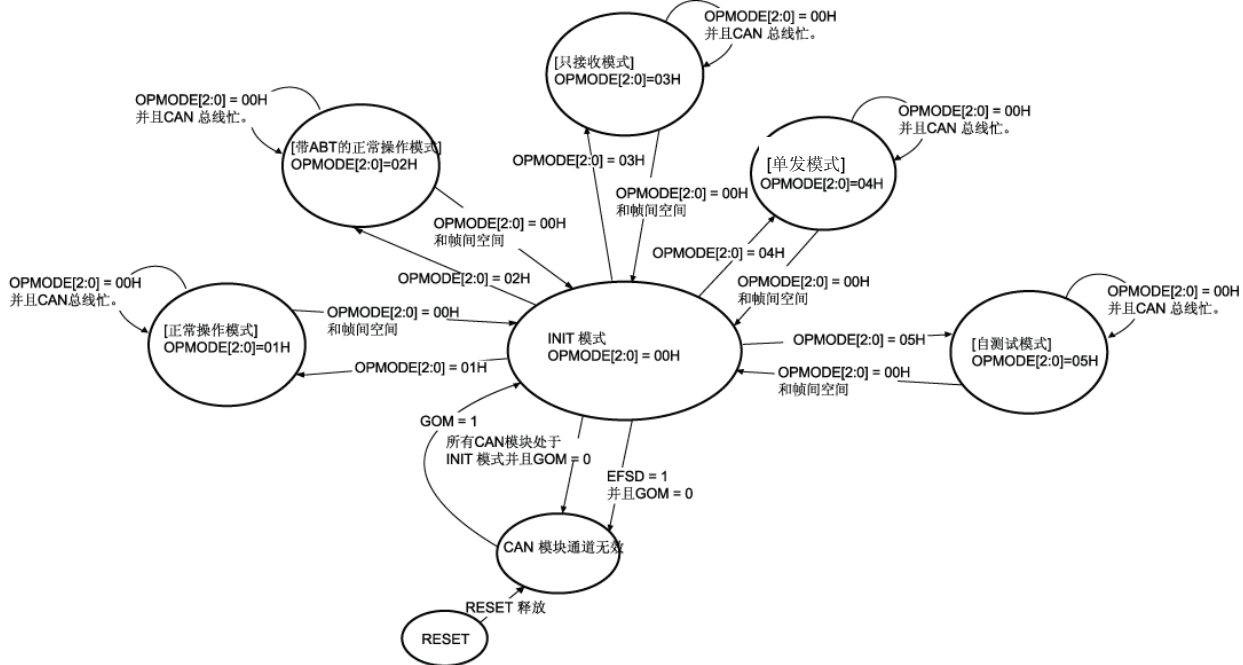
- 注意事项**
1. 当接收到信息时，接收过滤器对比 ID 和每个接收信息缓冲器的屏蔽设置。如果没有按照图 19-39 所表示的处理，信息缓存的内容在被重定义后可能会与接收的结果冲突（这是由于接收过滤器的原因）。如果发生这样的情况，首先要检查接收到的 ID 和 IDE 并且在下一次重定义后保存在信息缓存之中的是那些在信息缓存被重定义后被保存的。如果在重定义后没有保存 ID 和 IDE，将再一次的重定义信息缓存。
 2. 当发送信息时，将会检测发送优先级以及设定了发送请求的发送信息缓存的 ID，IDE，和 RTR 位的设置。选择具有最高优先级的发送信息缓存作为发送对象。如果没有遵守图 19-39 所表示的进程，在重定义之后 ID 中没有最高优先级的信息可能会被发送。

19.8.4 从初始化模式转换到操作模式

CAN 模块可以被切换到下列操作模式。

- 正常操作模式
- 带有 ABT 的正常操作模式
- 只接收模式
- 单发模式
- 自测试模式

图 19-28. 操作模式转换图



从初始化模式到操作模式的转换是通过 C0CTRL.OPMODE2 到 C0CTRL.OPMODE0 位控制的。

从一个模式转换到另外一个模式需要用初始化模式进行过渡。不要试图从一个操作模式直接切换到另一个操作模式；否则无法保证操作成功。

当 CAN 总线未处于帧间空间时，从操作模式切换到初始化模式的请求将会被挂起（例如：继续接收或是发送帧），并且 CAN 模块将会在帧间空间的第一位时进入初始化模式（OPMODE2 到 OPMODE0 位的值变为 000B）。当发出切换到初始化模式的请求时，读取 OPMODE2 到 OPMODE0 位直到它们的值变为 000B 来确认模块已经进入了初始化模式（见 图 19-37）。

19.8.5 CAN模块的复位误差计数器C0ERC

当重新初始化或者从总线关闭状态强制恢复时必须复位 C0ERC 和 C0INFO 寄存器，并且在初始化模式中将 C0CTRL.CCERC 位置 1。将该位设置为 1 后，C0ERC 和 C0INFO 寄存器将变为它们的默认值。

19.9 信息接收

19.9.1 信息接收

在所有操作模式下，在所有信息缓存区域中搜索满足以下条件的缓存，以便存储新接收的信息。

- 用作信息缓存
(COMCONFm.MA0 位置 1.)
- 设定一个接收信息缓存
(将 COMCONFm.MT2 到 COMCONFm.MT0 位设定为 001B, 010B, 011B, 100B, 或 101B.)
- 准备接收
(COMCTRLm.RDY 位置 1.)

备注 m = 00 到 31

当 CAN 模块的两个或更多的信息缓存接收到信息时，信息将按照下面解释的优先级保存。信息永远会被保存在拥有最高优先级的信息缓存中而不是拥有最低优先级的信息缓存。例如，一个非屏蔽的接收信息缓存和一个连接到屏蔽 1 的接收信息缓存拥有相同的 ID，接收到的信息不会被保存在那些未接收到信息的连接到屏蔽 1 的信息缓存中，即使非屏蔽接收信息缓存中已经接收到信息。换言之，如果存储信息的条件被设定到带有不同优先级的信息缓存中，拥有最高优先级的信息缓存将会存储信息；拥有较低优先级的信息缓存将不会存储信息。这条规则同样适用于拥有最高优先级的信息缓存不能接收和存储数据的情况（例如，当 DN 位=1 表示信息已经接收，但是由于 OWS 位=0 而禁止重新写入）。在这种情况下，实际上信息并没有接收并保存带有最高优先级的候选信息缓存中，但是也不存储在具有较低优先级的信息缓存中。

优先级	设置相同 ID 后的保存条件	
1（高）	非屏蔽信息缓存	DN 位 = 0
		DN 位 = 1 和 OWS 位 = 1
2	连接到 mask 1 信息缓存	DN 位 = 0
		DN 位 = 1 和 OWS 位 = 1
3	连接到 mask 2 信息缓存	DN 位 = 0
		DN 位 = 1 和 OWS 位 = 1
4	连接到 mask 3 信息缓存	DN 位 = 0
		DN 位 = 1 和 OWS 位 = 1
5（低）	连接到 mask 4 信息缓存	DN 位 = 0
		DN 位 = 1 和 OWS 位 = 1

<R> 19.9.2 读取接收数据

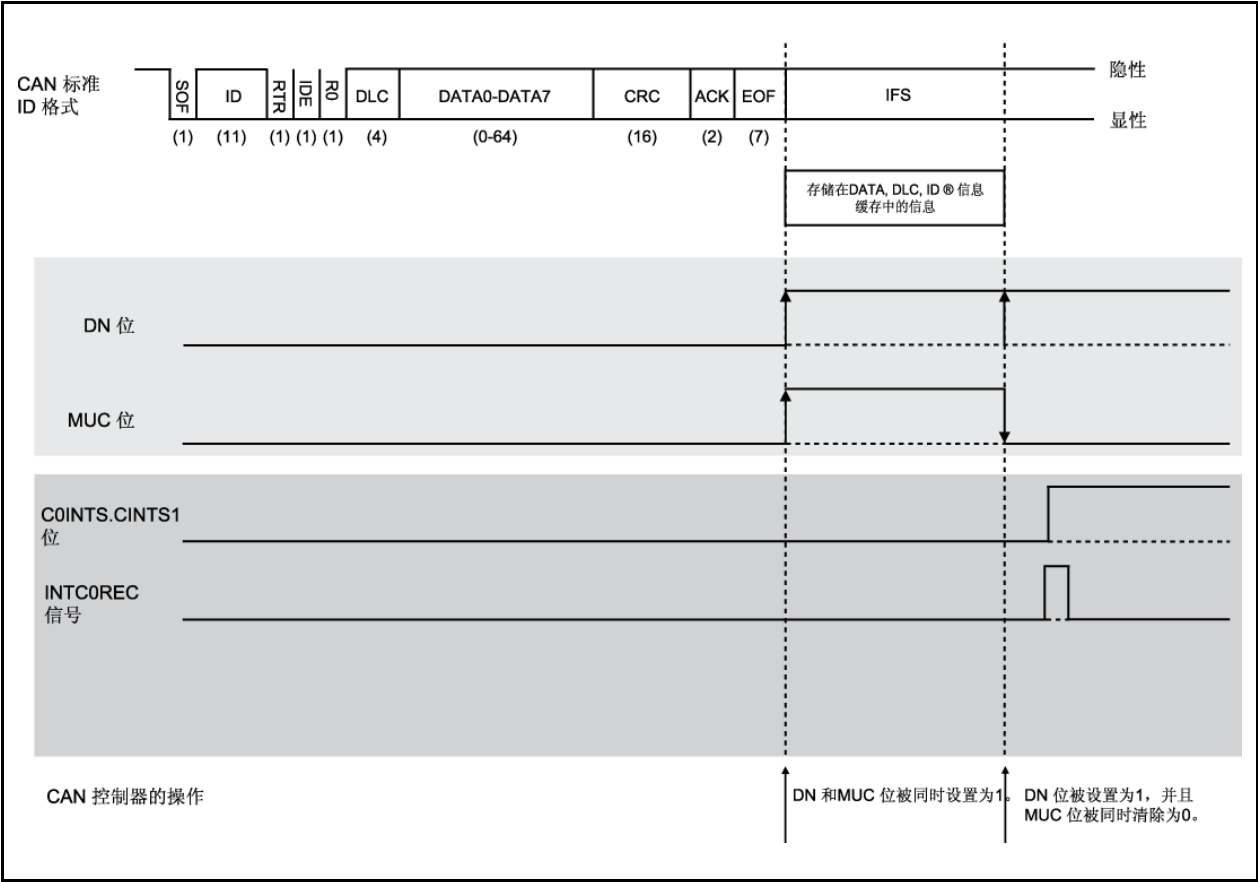
如果需要通过软件从 CAN 信息缓存中连续读取数据，按照图 19-49 和 19-50 中显示的推荐过程进行。

当接收一个信息时，在存储数据到信息缓存的处理开始时和在存储处理的结束时，CAN 模块两次设置 C0MCTRLm.DN 位。在这个存储过程中，信息缓存的 C0MCTRLm.MUC 位被置位（参阅图 19-29）。

在数据被完全存储前，接收历史列表被写入。在这个数据存储期间（MUC 位 = 1）中，禁止 CPU 重写要存储数据的信息缓存的 C0MCTRLm.RDY 位。这个数据存储处理的完成可能由于 CPU 对任意信息缓存的访问而延迟。

备注 m = 0 到 31

图 19-29. DN 和 MUC 位设置周期（以标准 ID 格式）



19.9.3 接收历史列表功能

接收历史列表（RHL）功能在接收历史列表中记录那些接收和存储过数据帧和远程帧的接收信息缓存的编号。RHL 包含了多达 23 个信息的存储元素，相应的 C0LIPT 寄存器的最后输入指针（LIPT）和 C0RGPT 寄存器的接收历史列表获取指针（RGPT）。

当 CAN 模块从初始化模式转换到任何一种操作模式的瞬间，RHL 是不确定的。

C0LIPT 寄存器保存 RHL 元素的内容，这些是通过 LIPT 指针的值减去 1 表示的。因此通过读取 C0LIPT 寄存器，可以检测那些第一次接收和存储了数据帧和远程帧的信息缓存的编号。LIPT 指针用作写入指针来表示那些被记录的 RHL 信息缓存编号。一旦接收和存储了数据帧或是远程帧，RHL 元素会记录相应的信息缓存的编号并通过 LIPT 指针表示出来。如果记录工作完成，LIPT 指针的数值自动增加。在这种方式下，可以按顺序记录下接收并存储了帧的信息缓存的编号。

RGPT 指针作为一个读取指针用来从 RHL 中读取信息缓存的编号地址。该指针指出了没有被 CPU 读取的第一个 RHL 元素。通过软件读取 C0RGPT 寄存器，可以得到接收和存储了数据帧或是远程帧的信息缓存编号。每当从 C0RGPT 寄存器中读取信息缓存编号的时候，RGPT 指针的值会自动增加。

如果 RGPT 指针的值和 LIPT 指针的值相匹配，C0RGPT.RHPM 位（接收历史列表指针匹配）置 1。这表明在 RHL 中已经没有未被读取的信息缓存编号。如果记录了一个新的信息缓存编号，LIPT 指针的值会自动增加，因为它的值与 RGPT 指针不再匹配，RHPM 位被清 0。换言之，未读取的信息缓存编号仍然记录在 RHL 中。

如果 LIPT 指针的值自动增加并且与 RGPT 指针的值减 1 后匹配，将 C0RGPT.ROVF 位（接收历史列表溢出）置 1。这表明了 RHL 中未被读取的信息缓存编号已经超过了上限。此时如果另外的信息接收和存储事件发生，最后一条信息缓存的编号会被接收并存储了新信息的信息缓存的编号所覆盖。在 ROVF 位被置 1 后，RHL 中记录的信息缓存编号将不能完全正确地反映接收的顺序。但是信息不会因此而丢失，并且在 DN 位的帮助下，可以通过 CPU 的查找在信息缓存存储器中准确定位。

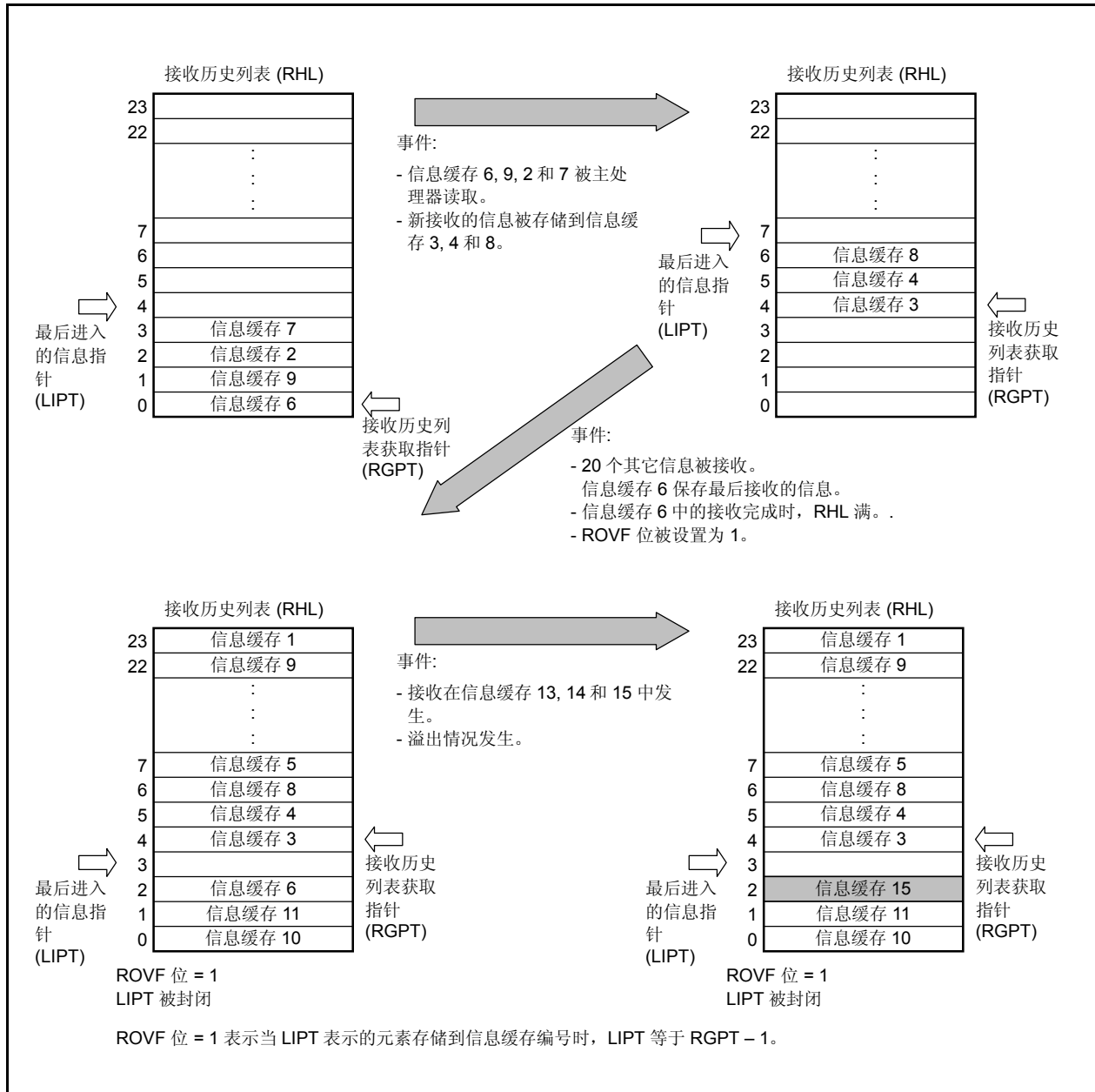
<R>

注意事项 即使接收历史列表溢出（C0RGPT.ROVF 位 = 1），在没有更多历史未读并且 C0RGPT.RHPM 位被置位前，都可以读取接收历史。然而，ROVF 位被保持置位（= 溢出发生），直到通过软件清零。在这种状态下，即使一个新的接收历史被存储并且写入到列表，RHPM 位也不被清零，除非 ROVF 位被清零。因此，如果 ROVF 位 = 1 和 RHPM 位 = 1 并且接收历史列表溢出，即使新的历史被接收和存储，RHPM 位也表明没有更多历史未读。

备注 m = 00 到 31

只要 RHL 中的包含了 23 个或者更少的条目，则事件会被按顺序保存。如果在主处理器未读取的情况下发生了更多的接收事件，则事件发生的顺序将被打乱。

图 19-30. 接收历史列表



19.9.4 屏蔽功能

对于一些用来接收的信息缓存，可以在四个全局接收屏蔽中选择一个来应用。

通过比较信息标识符，屏蔽一些位来加载结果，所以，一个缓冲器可以接收不同标识符。

通过使用屏蔽功能，可以对比从 CAN 总线上接收到信息的标识符与信息缓冲器的预先设置。而不必考虑被屏蔽的 ID 是否设定为 0 或是 1，从而把接收到的信息保存在预先定义好的信息缓存中。

当屏蔽功能生效时，一个在接收信息中屏蔽定义为 1 的标识符不会与信息缓存中相应的标识符位做比较。

但是，被屏蔽选项定义为 0 的位都会执行比较。

例如，让我们假定所有带有标准信息格式 ID 位 ID27 到 ID25 为 0、位 ID24 和 ID22 为 1 的信息都会被存储在信息缓存 14。执行的过程如下所示。

<1> 存放在信息缓存中的标识符

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

备注 x 位忽略

<2> 在信息缓存 14 中配置的标识符（举例说明）
（使用 COMIDL14 和 COMIDH14 寄存器）

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x
ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
x	x	x	x	x	x	x	x	x	x	x
ID6	ID5	ID4	ID3	ID2	ID1	ID0				
x	x	x	x	x	x	x				

信息缓存 14 ID 设置为（初始化）ID27 到 ID25 位清 0 且 ID24 和 ID22 位置 1。

备注 x 位忽略

备注 信息缓存 14 被设定为标准格式的标识符连接到屏蔽 1（COMCONF14.MT2 到 COMCONF14.MT0 位设定为 010B）。

<3> CAN 模块 0 的屏蔽设置（屏蔽 0）（举例）
（使用 CAN0 模块屏蔽 1 寄存器 L 和 H（C0MASKL1 和 C0MASKH1））

CMID28	CMID27	CMID26	CMID25	CMID24	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18
1	0	0	0	0	1	0	1	1	1	1
CMID17	CMID16	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	CMID7
1	1	1	1	1	1	1	1	1	1	1
CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0				
1	1	1	1	1	1	1				

1: 不比较（被屏蔽）
0: 比较

CMID27 到 CMID24 和 CMID22 位清 0，且 CMID28、CMID23 和 CMID21 到 CMID0 位置 1。

19.9.5 复合缓存接收块功能

复合缓存接收块 (MBRB) 功能用于在没有 CPU 干预的情况下在两个或者更多的信息缓存中存储数据块，该功能是通过把同一种类型的两个或者更多的信息缓存设定为相同的 ID 来实现的。这些信息缓存可以分配在信息缓存存储器中的任意区域中，并且它们不需要被分配为互相邻近。

例如，把 10 个信息缓存（信息缓存 10 到 19）设定为相同的信息缓存类型和相同的 ID。如果接收到第一个信息的 ID 与信息缓存的 ID 相匹配，它将会被保存到信息缓存 10。在这点上，信息缓存 10 的 DN 位被置位，从而禁止覆盖该信息缓存。

当接收到下一个具有匹配 ID 的信息时，使用信息缓存 11 将其接收并保存。每当接收到一个带有匹配 ID 的信息时，按顺序（升序）将其存储到信息缓存 12, 13,。即使接收到包含多个信息数据块，该信息依然可以在不覆盖以前接收到的带有匹配 ID 数据的情况下，正常的接收并保存。

可以通过检测每一个信息缓存的 COMCTRLm.IE 位来判定一个数据块是否已经被接收和保存。例如，一个数据块包含了 K 个信息，初始化 K 个信息缓存被用于接收该数据块。信息缓存 0 到 (K-2) 中的 IE 位被清 0（禁止中断），且信息缓存 K-1 的 IE 位被置 1（允许中断）。在这样的情况下，当信息缓存 K-1 完成对信息的接收和储存后会产生一个接收完成中断，指出 MBRB 已经达到存储上限。或者，通过清除信息缓存 0 到 (K-3) 的 IE 位且设定信息缓存 K-2 的 IE 位的方式，会发出 MBRB 即将溢出的警告。

在每一个 MBRB 接收缓存中存储接收数据的基本条件与在单一信息缓存中存储数据的条件相同。

- 注意事项**
1. MBRB 功能可以配置在任何一类相同类型的信息缓存。因此，即使另一个 MBRB 中的一个信息缓存的 ID 匹配但是信息类型不同，而且其中没有存放信息，接收到的信息也不会存储在这里而是被直丢弃。
 2. MBRB 不具备环形的缓存结构。因此，当信息被存储在通过 MBRB 配置拥有最高编号的信息缓存中之后，新接收到的信息不会被保存在拥有最低编号的信息缓存之中。
 3. MBRB 操作基于接收和存储条件；MBRB 操作没有专门的设置，比如允许位。通过对两个或是更多的信息缓存设定相同的信息缓存类型和 ID，自动配置 MBRB。
 4. 在 MBRB 中，“匹配 ID”意味着“在屏蔽后匹配 ID”。即使设定给每个信息缓存的 ID 是不同的，如果被屏蔽寄存器屏蔽的 ID 匹配，系统就会认为是匹配 ID，并将拥有该 ID 的信息缓存作为信息的存储目标。
 5. 在每一个 MBRB 当中优先级依照 19.9.1 信息接收所示的优先级。

备注 m = 00 到 31

19.9.6 接收远程帧

在所有的操作模式中，如果接收到远程帧，会在所有的信息缓存中寻找符合以下条件的信息缓存用于存储远程帧。

- 用作信息缓存
(COMCONFm.MA0 位置 1.)
- 设定为发送信息缓存
(COMCONFm.MT2 到 COMCONFm.MT0 位置 000B)
- 准备接收
(COMCTRLm.RDY 位置 1.)
- 设定为发送信息
(COMCONFm.RTR 位清 0)
- 未设定发送请求。
(COMCTRLm.TRQ 位清 0)

按照以上的方式来接收远程帧，如果接收到的远程帧的 ID 与满足上述条件的信息缓存的 ID 相匹配，将执行下列操作。

- C0MDLCm.DLC3 到 C0MDLCm.DLC0 位保存接收到的 DLC 的值。
- 在数据区的 C0MDATA0m 到 C0MDATA7m 寄存器不会被更新（保存接收之前的数据）。
- C0MCTRLm.DN 位置 1。
- C0INTS.CINTS1 位置 1（如果接收并存储了帧的数据缓存的 C0MCTRLm.IE 位置 1）。
- 输出接收完成中断（INTC0REC）（如果接收并存储了帧的数据缓存的 IE 位置 1 并且如果 C0IE.CIE1 位置 1）。
- 接收历史列表记录信息缓存的编号。

注意事项 当为接收和保存远程帧寻找信息缓存时，不会影响由信息缓存的 COMCONFm.OWS 位和 DN 位控制的覆盖操作。在任何情况下，OWS 位的设置被忽略，并且 DN 位被设置为 1。

如果多于一个发送信息缓存拥有相同的 ID，并且与接收到的远程帧的 ID 相匹配，这个远程帧将会保存在带有最低编号的发送信息缓存之中。

备注 m = 00 到 31

19.10 信息发送

19.10.1 信息发送

在所有的操作模式中，如果满足下列条件的信息缓存的 C0MCTRLm.TRQ 位置 1，系统会搜索用于发送信息的信息缓存。

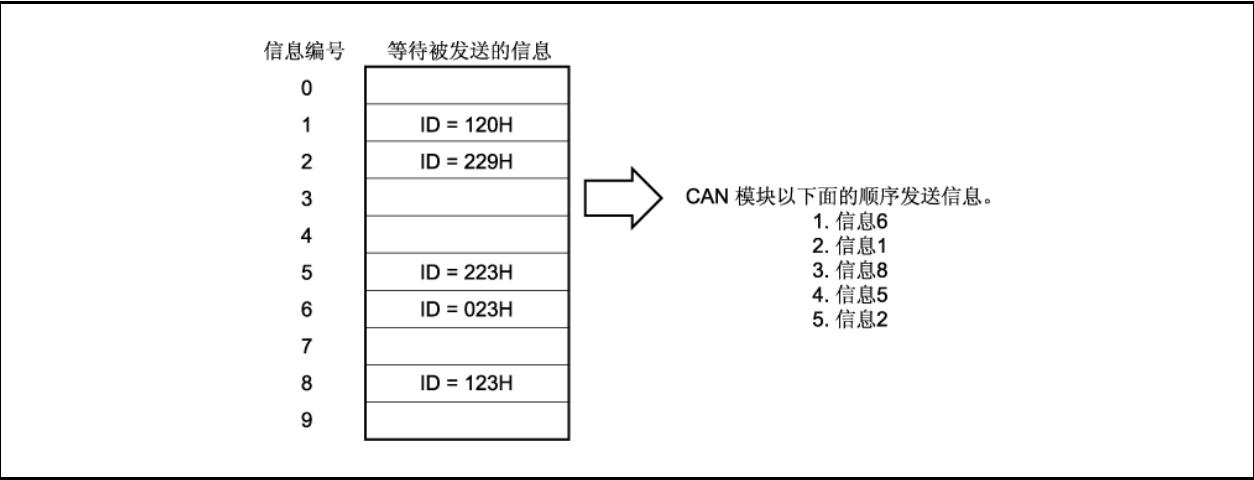
- 用作信息缓存
(C0MCONFm.MA0 位置 1.)
- 设定为发送信息缓存
(C0MCONFm.MT2 到 C0MCONFm.MT0 位设置为 000B.)
- 准备发送
(C0MCTRLm.RDY 位置 1.)

备注 m = 00 到 31

CAN 系统是多主通信系统。在这样的系统中，信息发送的优先级是由信息的标识符决定的 (IDs)。当多个信息等待发送时用软件促进发送处理，CAN 总线模块使用硬件来检测拥有最高优先级的信息的 ID 并且自动识别该信息。这样就不再需要基于软件的优先级控制。

通过标识符 (ID) 来控制发送优先级。

图 19-31. 信息处理举例



在搜索发送信息之后，系统将发送发出等待请求（预先将信息缓存的 TRQ 位置 1）的发送信息缓存中带有最高优先级的发送信息。

如果设置了一个新的发送请求，系统将比较这个带有新的发送请求的发送信息缓存和发出等待请求的发送信息缓存。如果新的发送请求拥有较高的优先级，它将被发送，除非一个拥有较低优先级的信息发送进程已经开始执行。要解决这个优先级的反转，软件可以请求中止发送低优先级的信息。最高优先级的判定依据以下规则。

优先级	条件	描述
1（高）	ID 的前 11 位的值 [ID28 到 ID18]:	首先发送 ID 前 11 位值最小的信息帧。如果标准帧的 11 位 ID 的值小于等于 29 位扩展 ID 的前 11 位，则 11 位标准 ID 比带有 29 位扩展 ID 的信息帧具有更高的优先级。
2	帧类型	带有 11 位标准 ID 的数据帧（COMCONFm.RTR 位清 0）比带有标准 ID 的远程帧以及带有扩展 ID 的信息帧拥有更高的优先级。
3	ID 类型	带有标准 ID 的信息帧（C0MIDHm.IDE 位清 0）比带有扩展 ID 的信息帧具有更高的优先级。
4	ID 的低 18 位的值 [ID17 到 ID0]:	如果一个或多个等待发送进程的信息帧的扩展 ID 的前 11 位相等，且帧类型相同（RTR 位的值相等），首先发送扩展 ID 的低 18 位值最小的的信息帧。
5（低）	信息缓存号	如果两个或以上的信息缓存中有等待发送的信息帧且它们的 ID 相同，那么拥有最低编号的信息缓存中的信息帧首先被发送。

备注 1. 如果在带有 ABT 的正常操作模式中自动块发送请求位 C0GMABT.ABTTRG 置 1，那么仅当 ABT 信息缓存组中存在一个信息缓存时 TRQ 位置 1。

如果通过设置 ABTTRG 位触发 ABT 模式，在 ABT 区域中一个 TRQ 位置 1（缓存 0 到 7）。除了这个 TRQ 位之外，应用进程可以为其它不属于 ABT 区域的发送信息缓存请求发送（置 TRQ 位为 1）。在这种情况下一个内部仲裁进程（发送查询）会评估所有的 TRQ 位置 1 的发送信息缓存并且选择带有较高优先级信息缓存以便下一次发送。如果两个以上的标识符都拥有最高优先级（例如同样的标识符），则具有较低信息缓存编号中的信息会被首先发送。

在成功的发送信息帧的同时，将执行下列操作。

- 相应的发送信息缓存的 TRQ 位清 0。
- C0INTS 寄存器中的发送完成状态位 CINTS0 置 1（如果相应的发送信息缓存的中断允许位（IE）置 1）。
- 输出中断请求信号 INTC0TRX（如果 C0IE.CIE0 位置 1 且相应的发送信息缓存的中断允许位（IE）置 1）。

<R> 2. 在更改发送信息缓存的内容前，这个缓存的 RDY 标志必须被清除。因为 RDY 标志可能在内部处理被更改时暂时被锁定，需要在更改缓存内容后检查 RDY 标志的状态。

19.10.2 发送历史列表功能

发送历史列表（THL）功能在发送历史列表中记录那些用于接收或存储了数据帧或是远程帧的发送信息缓存的编号。THL 包含了多达 7 个信息的等价存储元素，相应的 C0LOPT 寄存器的最后的输出指针（LOPT）和 C0RGPT 寄存器的接收历史列表获取指针（TGPT）。

当 CAN 模块从初始化模式转换到任何一种操作模式的瞬间，THL 是不确定的。

C0LOPT 寄存器保存 THL 元素的内容，这些是通过 LOPT 指针的值减去 1 表示的。因此通过读取 C0LOPT 寄存器，可以检测那些发送了数据帧或远程帧的信息缓存的编号。LOPT 指针用作写入指针来表示那些被记录的 THL 信息缓存编号。一旦发送数据帧或是远程帧，THL 元素会记录相应的信息缓存的编号并通过 LOPT 指针表示出来。如果记录工作完成，LOPT 指针的数值自动增加。在这种方式下，可以按顺序记录下接收并存储了帧的信息缓存的编号。

TGPT 指针作为一个读取指针用来从 THL 中读取信息缓存的编号。该指针指出了没有被 CPU 读取的第一个 THL 元素。通过软件读取 C0TGPT 寄存器，可以得到发送了数据帧或是远程帧的信息缓存编号。每当从 C0TGPT 寄存器中读取信息缓存编号的时候，TGPT 指针的值会自动增加。

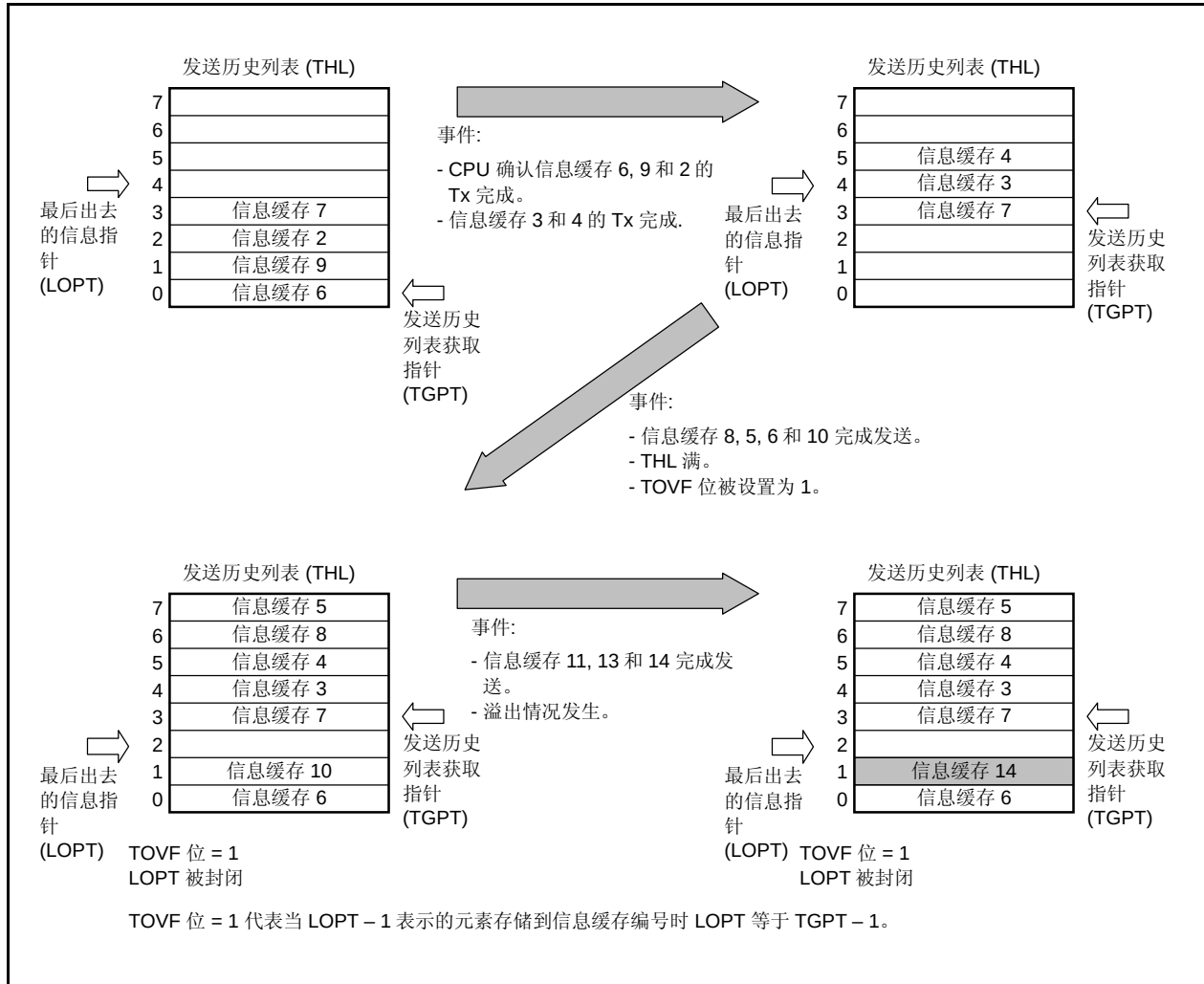
如果 TGPT 指针的值和 LOPT 指针的值相匹配，C0TGPT.THPM 位（发送历史列表指针匹配）置 1。这表明在 THL 中已经没有未被读取的信息缓存编号。如果记录了一个新的信息缓存编号，LOPT 指针的值会自动增加，因为它的值与 TGPT 指针不再匹配，THPM 位被清 0。换言之，未读取的信息缓存编号仍然记录在 THL 中。

如果 LOPT 指针的值自动增加并且与 TGPT 指针的值减 1 后匹配，将 C0TGPT 寄存器中的 TOVF 位（接收历史列表溢出）置 1。这表明了 THL 中未被读取的信息缓存编号已经超过了上限。此时如果发生另外的信息接收和存储事件，最后一条信息缓存的编号会被接收并存储了新信息的信息缓存的编号所覆盖。在 TOVF 位被置 1 后，THL 中记录的信息缓存编号将不能正确地反映接收的顺序。通过 CPU 搜索所有发送缓存可以找到发送信息，除非 CPU 没有预先在其中的一个缓存中覆盖发送目标。在不覆盖 THL 的情况下，总共可以产生 6 个发送完成。

<R>

注意事项 即使发送历史列表溢出（C0TGPT.TOVF 位 = 1），在没有更多历史未读并且 C0TGPT.THPM 位被置位前，可以读取发送历史。然而，直到通过软件清零前 TOVF 位被保持置位（= 溢出发生）。在这种状态下，即使一个新的发送历史被存储并且写入到列表，THPM 位也不被清零，除非 TOVF 位被清零。因此，如果 TOVF 位 = 1 且 THPM 位 = 1 并且接收历史列表溢出，即使新的历史被接收和存储，RHPM 位也表明没有更多历史未读。

图 19-32. 发送历史列表



19.10.3 自动块发送（ABT）

自动块发送（ABT）功能用于在没有 CPU 干预的情况下发送两个或是更多的数据帧。分配给 ABT 功能的最大发送信息缓存的数目是 8 个（信息缓存的编号为 0 到 7）。

通过把 C0CTRL.OPMODE2 到 C0CTRL.OPMODE0 位设定为 010B，可以选择“带有自动块发送功能的正常操作模式”（以下称 ABT 模式）。

为了发出 ABT 发送请求，首先要通过软件定义信息缓存。在所有用于 ABT 的信息缓存中设置 C0MCONFm.MA0 位，通过将 C0MCONFm.MA2 到 C0MCONFm.MA0 位设置为 000B 定义所有缓存为发送信息缓存。确定为每个用于 ABT 的信息缓存设置 ID，即使该 ID 正在被所有的信息缓存应用。为了使用两个或是更多的 ID，通过使用 C0MIDLm 和 C0MIDHm 寄存器为每一个信息缓存设定 ID。在发出 ABT 功能的发送请求之前设定 C0MDLCm 和 C0MDATA0m 到 C0MDATA7m 寄存器。

当 ABT 信息缓存初始化完成后，C0MCTRLm.RDY 位需要置 1。在 ABT 模式中，C0MCTRLm.TRQ 位不需要通过软件来操作。

当 ABT 信息缓存中的数据准备完毕之后，C0GMABT.ABTTRG 位置 1。自动块发送随后开始。当 ABT 开始后，第一个信息缓存（信息缓存 0）中的 TRQ 位自动置 1。在信息缓存 0 中的数据发送完成后，下一个信息缓存（信息缓存 1）中的 TRQ 将自动置 1。通过这种方式，发送将成功的进行。

在成功发送正在执行的过程中，发送请求（TRQ 位）自动置位的间隔中程序会插入一个延迟时间。这个延迟时间是通过 C0GMABTD 寄存器定义的。这个延迟时间单元是 DBT（数据位时间）。DBT 取决于 C0BRP 和 C0BTR 寄存器的设定。

在 ABT 过程中，将不会在 ABT 发送信息缓存中查询 发送 ID 的优先级，而是按顺序发送信息缓存 0 到 7。当信息缓存 7 中的数据帧发送完成后，ABTTRG 位自动清 0 且中止 ABT 操作。

在 ABT 过程中如果 ABT 信息缓存中的 RDY 位清 0，不会发送该缓存中的数据，ABT 操作停止，并且 ABTTRG 位清 0。在这之后，当通过软件将 RDY 和 ABTTRG 位置 1 后，发送可以在 ABT 停止的信息缓存位置恢复。为了避免上述操作，当 ABT 模式停止后可以通过将 C0GMABT.ABTCLR 位置 1 来复位 ABT 引擎并将 ABTTRG 位清 0。在这种情况下，如果 ABTCLR 位清 0 且 ABTTRG 位被置 1 后，发送将从信息缓存 0 开始。

可以通过一个中断来检测 ABT 的所有信息缓存中的数据帧是否已经发送完成。为了实现该操作，除去最后一个信息缓存的所有信息缓存的 C0MCTRLm.IE 位必须都清 0。

如果除去用于 ABT 功能的其它信息缓存（信息缓存 8 到 31）被设置为发送信息缓存，用于发送信息的优先级是通过 ABT 信息缓存的发送 ID 中的优先级决定的，ABT 信息缓存的发送当前处于保持状态并且 ABT 功能正在使用其它的信息缓存的发送 ID。

从 ABT 信息缓存发送的数据帧不会记录在发送历史列表（THL）中。

- 注意事项
1. 如果从信息缓存 0 恢复带有 ABT 的正常操作模式，当 ABTTRG 位清 0 时，将 ABTCLR 位置 1。如果当 ABTTRG 位被置 1 时也将 ABTCLR 位被置 1，后续的操作不会得到保证。
 2. 如果通过将 ABTCLR 位被置 1 的方式清除自动块发送引擎，在清除请求进程完成之后 ABTCLR 位会立刻自动清 0。
 3. 不要在初始化模式中设置 ABTTRG 位。否则在系统从初始化模式转换到 ABT 模式后，无法保证操作成功。
 4. 在带有 ABT 的正常操作模式下不要通过软件将 ABT 信息缓存的 TRQ 位置 1。否则，操作将不能得到保证。
 5. C0GMABTD 寄存器用于为 ABT 模式中依次发送的 ABT 用的各报文按报文号码的顺序设定发送请求的时候，从先前的 ABT 报文的发送完成到下一个 ABT 消息的 TRQ 位的设置期间插入延迟量。信息在 CAN 总线上实际发送的时序改变取决于其它位置的发送状态和除去 ABT 信息的其它信息（信息缓存 8 到 31）设定的发送请求状态。
 6. 如果一个不同于 ABT 信息的其它信息的发送请求，并且没有延迟时间插入到 ABT 发送请求自动设置（C0GMABTD 寄存器= 00H）的时间间隔里，除去 ABT 信息的其它信息可能会被发送，而忽略关于 ABT 信息的优先级。
 7. 当 ABTTRG 位为 1 的时候不要将 RDY 位清 0。
 8. 在带有 ABT 的正常操作模式中如果从其它的节点接受到信息，可能在一个数据帧的时间过后发送该信息，即使 C0GMABTD 寄存器 = 00H。

备注 m = 00 到 31

19.10.4 发送中止进程

备注 m = 00 到 31

（1）在带有自动块发送 ABT 的正常操作模式以外的发送中止流程

用户可以将 C0MCTRLm.TRQ 清 0 用于中止发送请求。如果中止成功，TRQ 位立即会被清 0。发送是否被成功的中止可以通过使用 C0TGPT 寄存器的 C0CTRL.TSTAT 位进行检测，这一位指示了 CAN 总线的发送状态（详细情况参考图 19-46）。

（2）在带有自动块发送（ABT）的正常操作模式中除去 ABT 发送的发送中止进程

用户可以将 C0MCTRLm.TRQ 清 0 用于中止发送请求。在检测 C0GMABT 寄存器的 ABTTRG 位为 0 后，将 C0MCTRLm.TRQ 位清 0。如果中止成功，TRQ 位立即会清 0。发送是否被成功的中止可以通过使用 C0TGPT 寄存器的 C0CTRL.TSTAT 位进行检测，这一位指示了 CAN 总线的发送状态（详细情况参考图 19-47）。

(3) 在带有自动块发送 (ABT) 的正常操作模式中的发送中止

为了中止已经进行的 ABT，需要将 C0GMABT.ABTTRG 位清 0。在这种情况下，如果 ABT 信息正在被发送，则在这个过程完成之前（无论是否成功），ABTTRG 位保持为 1，并且在发送完成之后立即被清 0。从而中止了 ABT。

如果最后的发送（在 ABT 之前）是成功的，带有 ABT 的正常操作模式将离开该模式，并将内部 ABT 指针指向下一个要发送的信息缓存。

在一个不正确的发送过程中，内部 ABT 指针的位置取决于在最后的发送信息缓存中 TRQ 位的状态。当请求清除 ABTTRG 位时，TRQ 位置 1，内部 ABT 指针指向最后的发送信息缓存（详细情况参考图 19-48）。当请求清除 ABTTRG 位时，TRQ 位清 0，内部 ABT 指针以 1 为单位增加并且在 ABT 区域中指向下一个信息缓存（详细情况参考图 19-48）。

注意事项 确定通过将 ABTTRG 位清 0 的方式中止 ABT。如果通过将 RDY 清 0 的方式请求中止发送，操作的结果不会得到保证。

当带有 ABT 的正常操作模式在 ABT 中止并将 ABTTRG 位置 1 后恢复，下一个将要被发送的 ABT 信息缓存可以通过下面的表格确定。

ABT 信息缓存的 TRQ 位状态	在成功发送后中止	在不正常的发送后中止
设置 (1)	在 ABT 区域注中的下一个信息缓存	在 ABT 区域中的同一个信息缓存
清除 (0)	在 ABT 区域注中的下一个信息缓存	在 ABT 区域注中的下一个信息缓存

注 上述的恢复操作仅可以在 ABT 区域中存在准备好的 ABT 信息缓存时才能执行。例如，当信息缓存 7 的 ABT 产生中止请求时，如果信息缓存 7 的发送已经完成，即使 ABTTRG 位被清零，也认为是 ABT 完成而不是中止。如果在 ABT 区域中的下一个信息缓存中的 C0MCTRLm.RDY 位被清 0，内部 ABT 指针保持，但是不会执行恢复操作即使 ABTTRG 位置 1，并且 ABT 立即结束。

备注 m = 00 到 31

19.10.5 远程帧发送

远程帧只能在发送信息缓存中发送。通过 C0MCONFm.RTR 设置是否发送数据帧和远程帧。将 RTR 位设置为 1 允许远程帧发送。

备注 m = 00 到 31

19.11 节能模式

19.11.1 CAN 休眠模式

CAN 休眠模式用来设定 CAN 控制器进入待机模式来降低功耗。CAN 模块可以从所有操作模式进入 CAN 休眠模式。释放 CAN 休眠模式之后能够正确的返回原来的工作模式。

在 CAN 休眠模式中，CAN 模块不发送任何信息，即使发出或是保持了发送请求。

(1) 进入 CAN 休眠模式

CPU 通过将 C0CTRL.PSMODE1 和 C0CTRL.PSMODE0 设为 01B 发出一个 CAN 休眠模式转换请求。

只有在下列条件下这个转化请求才会被响应。

(i) CAN 模块正处于下列操作模式之一

- 正常操作模式
- 带有 ABT 的正常操作模式
- 只接收模式
- 单发模式
- 自测试模式
- 在上述所有模式中的 CAN 停止模式

(ii) CAN 总线处于空闲状态（帧间空间的第 4 位是隐性的）^注

注 如果 CAN 总线固定为显性的，转换到 CAN 休眠模式的请求被保持。休眠模式的请求与总线的状态无关。

(iii) 没有转换请求保存

如果上面所述的条件都没有出现，CAN 模块将按下面的方式操作。

- 如果在初始化模式下请求进入 CAN 休眠模式，则该请求被忽略且 CAN 模块依然处于初始化模式。
- 如果 CAN 总线不处于空闲状态（例如，CAN 总线处于发送或是接收状态），当在其它操作模式中请求 CAN 休眠模式时，不可能直接转为 CAN 休眠模式。在这种条件下，保持 CAN 休眠模式的发送请求直到 CAN 总线进入空闲状态（帧间空间的第 4 位是隐性的）。在成功转为 CAN 休眠状态请求的时间里，PSMODE1 位和 PSMODE0 位仍然是 00B。当模块进入 CAN 休眠模式后，PSMODE1 位和 PSMODE0 位被设定为 01B。
- 如果 CAN 模块正处于一个操作模式时，一个转换到初始化状态的请求和一个转换到 CAN 休眠模式的请求同时产生，首先允许初始化模式的请求。在预先的时序，CAN 模块进入初始化模式。这时，不会保留 CAN 休眠模式的请求，它将被丢弃。
- 即使不同时请求初始化模式和休眠模式（例如当第一个请求发出时第二个请求并没有产生），初始化请求的优先级高于 CAN 休眠模式的请求。当请求初始化模式时，CAN 休眠模式的请求被取消。当一个被保存的初始化请求出现时，后续的 CAN 休眠模式请求在提交的时候就会被取消。

(2) CAN 休眠模式中的状态

在进入 CAN 休眠模式后，CAN 模块会处于下列状态之一。

- 内部操作时钟停止，系统功耗最低。
- CAN 接收引脚（CRXD0）检测下降沿的功能依然有效用于从 CAN 总线激活 CAN 模块。
- 为了从 CPU 激活 CAN 模块，可以向 PSMODE1 位和 PSMODE0 位写入数据，但是却不能向其它的 CAN0 寄存器或是位写入数据。
- CAN0 模块寄存器可以读取除 C0LIPT、C0RGPT、C0LOPT 和 C0TGPT 寄存器的寄存器。
- 不能对 CAN0 信息缓存寄存器进行读取或是写入。
- C0GMCTRL.MBON 位被清零。
- 不会响应转换到初始化模式的请求，它将被忽略。

<R>

(3) 释放 CAN 休眠模式

通过以下事件释放 CAN 休眠模式。

- 当 CPU 向 PSMODE1 位和 PSMODE0 位写入 00B
- CAN 接收引脚（CRXD0）检测到下降沿（例如 CAN 总线电平从隐性转变成显性）

注意事项 1. 即使一个下降沿属于接收信息的 SDF，该信息也不会被接收和储存。在 CAN 休眠模式中如果 CPU 关闭 CAN 的时钟后，CAN 休眠模式也不会被释放并且 PSMODE1 和 PSMODE0 位仍然是 01B，除非重新提供 CAN 时钟。除此之外，后来的信息也不会被接收。

<R>

2. 当 CAN 时钟被提供时，如果在 CAN 接收引脚（CRXD0）上检测到下降沿，必须通过软件清除 PSMODE0 位（关于细节，参阅图 19-53 中的处理）。

在释放休眠模式之后，CAN 模块将返回之前发出 CAN 休眠模式请求的操作模式中，并且 PSMODE1 位和 PSMODE0 位被复位为 00B。如果通过改变 CAN 总线状态来释放 CAN 休眠模式，C0INTS.CINTS5 位置 1，忽略 C0IE.CIE 位的设置。当 CAN 模块从 CAN 休眠模式中释放后，当在 CAN 总线上检测到连续 11 个隐性电平位时，它将再次加入到 CAN 总线中。在释放休眠模式后并且通过程序再次访问信息缓存前，确认 C0GMCTRL.MBON 位 = 1。

当 CAN 模块处于 CAN 休眠模式时，接收到转换到初始化模式的请求会被忽略；在进入初始化模式之前，CPU 必须首先通过软件释放休眠模式。

<R>

注意事项 当 CAN 休眠模式被 CAN 总线的一个事件释放时，即使 CAN 总线的事件在更改为休眠模式后立即发生，也会产生一个唤醒中断。注意，中断可以在任何时间发生。

19.11.2 CAN 停止模式

停止模式用于将 CAN 控制器置于待机模式来降低功耗。CAN 模块只能从 CAN 休眠模式进入 CAN 停止模式。释放 CAN 停止模式后，CAN 模块将进入 CAN 休眠模式。

只能通过对 C0CTRL.PSMODE1 位和 C0CTRL.PSMODE0 位写入 01B 而不能通过改变 CAN 总线的状态来释放 CAN 停止模式（转换到 CAN 休眠模式）。即使发出或保存转换请求时，信息也不能被发送。

（1）进入 CAN 停止模式

向 PSMODE1 位和 PSMODE0 位写入 11B 时将会发出一个 CAN 停止模式转换请求。

只有当 CAN 模块处于 CAN 休眠模式时，才响应 CAN 停止请求。在其它模式中，忽略该请求。

注意事项 为了设定 CAN 模块进入 CAN 停止模式，模块必须处于 CAN 休眠模式。检测 PSMODE1 位和 PSMODE0 位的值为 01B 确认模块处于 CAN 休眠状态，然后才能请求进入 CAN 停止模式。当该进程正在进行中时，如果 CAN 接收引脚（CRXD0）发生总线状态改变，自动释放 CAN 休眠模式。在这种情况下，CAN 停止模式改变请求不能被响应。（但是，在 CAN 时钟加入时，在 CAN 接收引脚（CRXD0）的总线电平改变之后，PSMODE0 必须用软件清零。）

（2）CAN 停止模式中的状态

进入 CAN 停止模式后，CAN 模块处于下列状态之一。

- 内部操作时钟停止，系统功耗最低。
- 为了从 CPU 激活 CAN 模块，可以向 PSMODE1 位和 PSMODE0 位写入数据，但是却不能向其它的 CAN0 寄存器或是位写入数据。
- 可以读取除 C0LIPT、C0RGPT、C0LOPT 和 C0TGPT 寄存器外的 CAN0 模块寄存器。
- 不能对 CAN0 信息缓存寄存器进行读取或是写入。
- C0GMCTRL.MBON 位被清零。
- 不会响应转换到初始化模式的请求，它将被忽略。

（3）释放 CAN 停止模式

只能通过向 PSMODE1 位和 PSMODE0 位写入 01B 释放 CAN 停止模式。在这之后，CAN 模块将进入 CAN 休眠模式。

当 CAN 模块处于 CAN 停止模式时请求初始化模式，将忽略该请求；在进入初始化模式之前，CPU 必须首先依次释放 CAN 停止模式和 CAN 休眠模式。从 CAN 停止模式不经过 CAN 休眠模式直接转换到其它操作模式是不可能的，这样的请求将被忽略。

<R>

19.11.3 使用节能模式举例

在一些应用系统中，有必要将 CPU 置于节能模式以降低功耗。通过联合使用指定给 CAN 模块的节能模式以及指定给 CPU 的节能模式，可以通过 CAN 总线将 CPU 从节能模式中激活。

这里给出一个使用节能模式的范例。

首先，将 CAN 模块置于 CAN 休眠模式（将 PSMODE1 位和 PSMODE0 位置为 01B）。接下来将 CPU 置于节能模式。在这种状态下，如果从 CRXD0 信号中检测到隐性电平到显性电平的转换，CAN 模块中的 CINTS5 位置 1。如果 C0CTRL.CIE5 位置 1，则会产生一个唤醒中断（INTC0WUP）。CAN 模块将自动从 CAN 休眠模式（PSMODE1 位和 PSMODE0 位为 00B）中返回并进入正常操作模式。（PSMODE1, PSMODE0 位 = 00B）并回到正常运行模式（但是，在 CAN 时钟加入时，在 CAN 接收引脚（CRXD0）的总线电平改变之后，PSMODE0 必须用软件清零。）。而 CPU 因为响应了中断 INTC0WUP，也将释放其节能模式而返回正常工作模式。

为了更进一步降低 CPU 的功耗，应该关闭内部时钟（包含 CAN 模块的内部时钟）。在这种情况下，在 CAN 模块进入 CAN 休眠模式之后，应该关闭提供给 CAN 模块的操作时钟。然后 CPU 也进入自己的节能模式（提供给 CPU 的时钟也被关闭）。在这种状态下如果从 CRXD0 信号中检测到隐性电平到显性电平的转换，即使在系统没有提供时钟的情况下 CAN 模块将 CINTS5 位置 1 并产生唤醒中断（INTC0WUP）。但是其它功能不允许这样操作，因为提供给 CAN 模块的系统时钟被关闭，且模块正处于 CAN 休眠模式中。CPU 由于响应了 INTC0WUP 中断，也将释放它自身的节能模式，并且在晶振的稳定时间过后恢复它的内部时钟（包含提供给 CAN 模块的时钟），然后开始执行指令。当时钟恢复后，CAN 模块会立即释放 CAN 休眠模式并返回正常操作模式（PSMODE1 位和 PSMODE0 位为 00B）。

19.12 中断功能

CAN 模块提供 6 种不同的中断源。

中断事件的产生会被存储在中断状态寄存器中。6 个中断源会产生 4 个分离的中断请求信号。当一个符合两个或以上的中断源的中断信号产生时，可以通过使用中断状态寄存器识别中断源。在中断事件发生后，必须通过软件将相应的中断标志清 0。

表 19-20. CAN 模块中断源列表

编号.	中断状态位		中断允许位		中断请求信号	中断源描述
	名称	寄存器	名称	寄存器		
1	CINTS0 ^{注1}	C0INTS	CIE0 ^{注1}	C0IE	INTC0TRX	信息帧从信息缓存 m 中成功发送
2	CINTS1 ^{注1}	C0INTS	CIE1 ^{注1}	C0IE	INTC0REC	信息缓存 m 接收到有效的信息帧
3	CINTS2	C0INTS	CIE2	C0IE	INTC0ERR	CAN 模块错误状态中断 ^{注2}
4	CINTS3	C0INTS	CIE3	C0IE		CAN 模块协议错误中断 ^{注3}
5	CINTS4	C0INTS	CIE4	C0IE		CAN 模块仲裁丢失中断
6	CINTS5	C0INTS	CIE5	C0IE	INTC0WUP	CAN 模块从 CAN 休眠状态 ^{注4} 中激活

- 注
1. 将相应信息缓存的 C0MCTRL.IE 位（信息缓存中断允许位）置 1 说明该信息缓存参与中断产生的进程
 2. 当发送 / 接收错误计数器处于警告级别、错误无效或是 bus-off 状态时，该中断产生。
 3. 当填充错误、格式错误、ACK 错误、位错误或 CRC 错误发生时，该中断产生。
 4. 当 CAN 接收引脚检测到下降沿（CAN 总线从发生从隐性到显性的转变），CAN 模块从 CAN 休眠模式中返回时，该中断产生。

备注 m = 00 到 31

19.13 诊断功能和特殊的操作模式

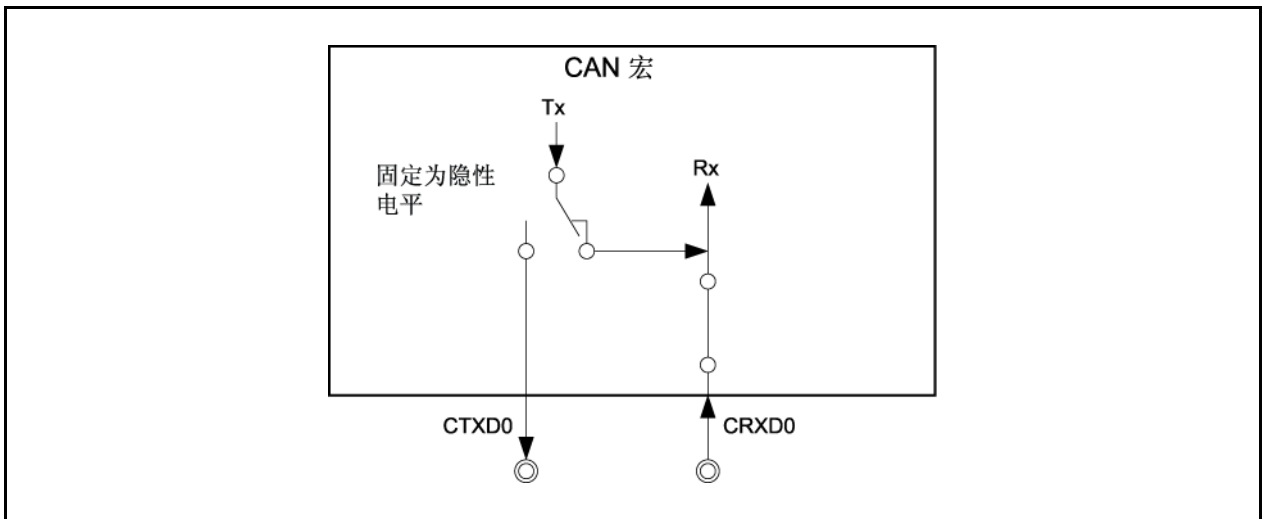
CAN 模块提供只接收模式、单发模式和自测试模式以支持 CAN 总线的诊断功能或是特殊的 CAN 通信模式操作。

19.13.1 只接收模式

只接收模式用于监控接收信息而不引起 CAN 总线的任何冲突，也可以用于 CAN 总线的分析节点。

例如，该模式可以用于自动探测波特率。CAN 模块的波特率不断改变直到检测到“正确接收”，因此模块中的波特率是匹配的（“正确接收”意味着在 CAN 协议层接收到了信息帧，而不会发生错误并且连接在 CAN 总线的节点之间带有相应的 ACK）。一个正确的接收不需要在接受信息缓存（数据帧）或是发送信息缓存（远程帧）中存储信息帧。正确的接收事件通过设定 C0CTRL.VALID 位为 1 来指示。

图 19-33. CAN 模块在接收模式中的终端连接



在只接收模式中，没有信息帧可以从 CAN 模块发送到 CAN 总线上。由被定义为发送信息缓存的信息缓存发出的发送请求被挂起。

在接收模式中，CAN 模块的 CAN 发送引脚（CTXD0）固定为隐性电平。因此，即使在接收信息帧的过程中检测到 CAN 总线错误，CAN 模块也不会发送有效错误标志到 CAN 总线。由于 CAN 模块不会进行发送操作，发送错误计数器的 C0ERC.TEC7 到 C0ERC.TECn 位不会被更新。因此，在接收模式中 CAN 模块不会进入总线关闭状态。

此外，在这种模式下正确的接收到信息帧后，ACK 也不会返回总线。在内部，本地节点认为已经发送了 ACK。过载帧不会被发送到 CAN 总线。

注意事项 如果只有两个 CAN 节点连接到 CAN 总线并且其中一个处于接收模式，CAN 总线上不会出现 ACK。由于缺少 ACK 信息，发送节点会发出一个有效错误标志，并且重复发送信息帧。在 16 次发送相同的信息之后（假定错误计数器的初始值为 0 且没有其它错误发生），发送节点进入错误无效状态。当信息帧第 17 次被发送时，发送节点产生一个有效错误标志。处于接收模式接收节点检测到第一个正确的信息帧，此时将 VALID 位置 1。

19.13.2 单发模式

在单发模式中，在 CAN 协议中定义的自动重发送功能被关闭。（依照 CAN 协议，由于仲裁丢失或是错误而中止的信息帧发送的重传操作不会受软件控制。）所有单发模式其它的操作行为都与正常操作模式相同。单发模式的特性不能在带有 ABT 正常操作模式中使用。

单发模式依照 C0CTRL.AL 位的设定可以禁止重发被中止的信息帧。当 AL 位清 0 时，由于仲裁丢失和发生错误而导致的重传操作被禁止。如果 AL 位置 1，仅禁止由于发生错误导致的重传操作，由于仲裁丢失导致的重传操作依然被允许。结果，被定义为发送信息缓存的信息缓存中的 C0MCTRLm.TRQ 被下述事件清 0。

- 成功的发送信息帧
- 发送信息帧时仲裁丢失（AL 位为 0）
- 发送信息帧时发生错误

仲裁丢失和发生错误可以通过检测 C0INTS.CINTS4 和 C0INTS.CINTS3 位，通过读取寄存器的 C0LEC.LEC2 到 C0LEC.LEC0 位可以得知错误的类型。

在成功发送信息帧，C0INTS 寄存器中的发送完成中断位 CINTS0 置 1。如果此时 C0IE.CIE0 位置 1，发出中断请求信号。

单发模式可以用于仿真时间触发通信模式（例如 TTCAN 协议等级 1）。

注意事项 AL 仅在单发模式下有效。在其它操作模式中，它不会影响仲裁丢失后引发的重传操作。

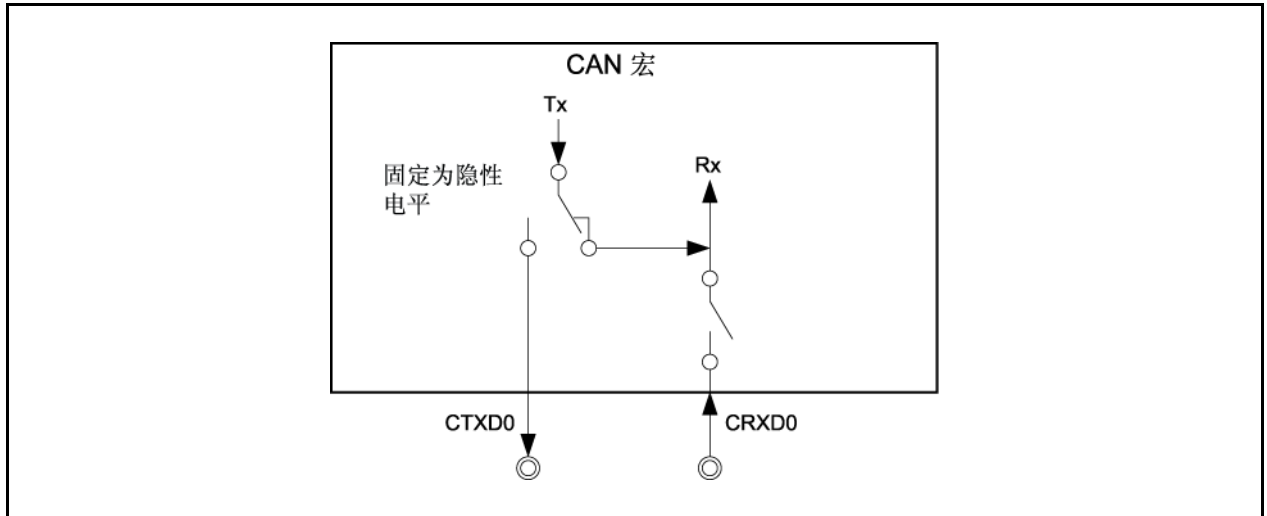
19.13.3 自测试模式

在自测试模式中，信息帧发送和信息帧接收可以在 CAN 节点没有连接到 CAN 总线或是不影响 CAN 总线的条件下进行测试。

在自测试模式中，CAN 模块完全脱离 CAN 总线，但是发送和接收是内部环回的。CAN 发送引脚（CTXD0）被固定在隐性电平。

当 CAN 模块从自测试模式进入 CAN 休眠模式后，如果 CAN 接收引脚（CRXD0）检测到下降沿，和其它操作模式一样，模块将释放 CAN 休眠模式（当提供了 CAN 时钟，在 CAN 接收引脚（CRXD0）检测到总线电平变化时用软件清除 PSMODE0 位）。为了使模块保持 CAN 休眠模式，要将 CAN 接收引脚（CRXD0）设置为端口引脚。

图 19-34. 在自测试模式中 CAN 模块终端连接



<R> 19.13.4 在每种操作模式下的发送 / 接收操作

表 19-21 表示每种模式下的发送 / 接收操作。

表 19-21. 每种操作模式下的发送 / 接收操作概述

操作模式	数据帧 / 远程帧 发送	ACK 发送	错误帧 / 过载帧 发送	重新发送	自动块发送 (ABT)	VALID 位的设置	在信息缓存中存 储数据
初始化模式	—	—	—	—	—	—	—
正常操作模式	√	√	√	√	—	√	√
带有 ABT 的正常操作模式	√	√	√	√	√	√	√
只接收模式	—	—	—	—	—	√	√
单发模式	√	√	√	— ^{注 1}	—	√	√
自测试模式	√ ^{注 2}	√ ^{注 2}	√ ^{注 2}	√ ^{注 2}	—	√ ^{注 2}	√ ^{注 2}

- 注
- 1. 如果仲裁丢失，重新发送可以由 COCTRL.AL 位选择。
 - 2. 每个信号不会输出到外部电路，但是会在 CAN 模块内部产生。

19.14 时间标记功能

CAN 是一个异步串行协议。所有连接到 CAN 总线上的节点使用一个本地的异步时钟。所以节点之间的时钟没有任何关系（例如,时钟可以是异步的甚至拥有不同的频率）。

但是在一些应用中，网络中需要一个通用的节拍（基于全局的节拍）。为了实现这种应用，需要使用时间标记功能。时间标记功能的基本原理是在 CAN 总线上捕获由信号触发的定时器的值。

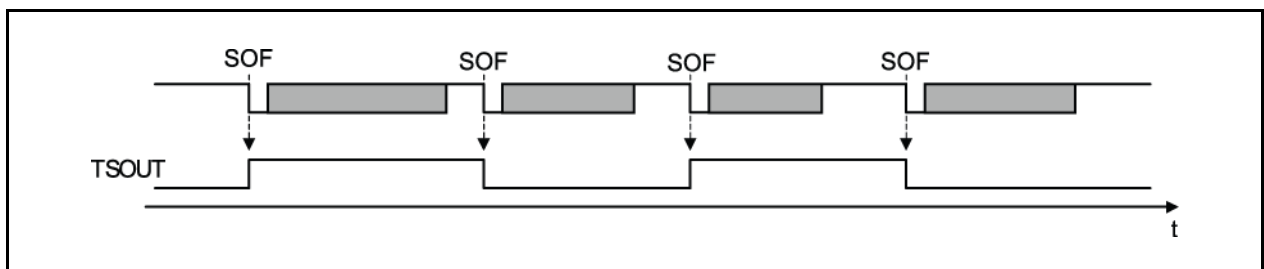
19.14.1 时间标记功能

CAN 控制器支持通过特殊的帧捕获定时器的值。除去 CAN 控制器之外，在单片机系统中还使用一个 16 位捕获定时器单元。这个 16 位捕获定时器单元通过一个触发信号（TSOUT）捕获定时器的值，当 CAN 控制器接收到数据帧时发出触发信号。CPU 可以通过读取捕获值记录捕获时间的时间，例如从 CAN 总线接收到信息的时间标记。TSOUT 信号可以通过以下两个事件源和 C0TS.TSSEL 位的设定来选择。

- SOF 事件（帧的起始）（TSSEL 位 = 0）
- EOF 事件（帧的最后一位）（TSSEL 位 = 1）

TSOUT 信号通过将 C0TS.TSEN 位置 1 来使能。

图 19-35. 捕获信号 TSOUT 的时序图



在数据帧接收期间，TSOUT 信号固定自己的电平直到选定的事件发生（在图 19-34 中，SOF 用于触发事件源）。为了通过 TSOUT 信号捕获定时器的值，捕获定时器单元必须能够检测捕获信号的上升沿和下降沿。

时间标记功能是通过 C0TS.TSLOCK 位控制的。当 TSLOCK 位清 0，TSOUT 信号通过发生选定的事件触发。如果 TSLOCK 位置 1，TSOUT 信号通过发生选定的事件触发，但是在信息帧开始被接收和存储到信息缓存 0 后 TSEN 位自动清 0 时停止触发。这样会通过 TSOUT 信号抑制后续的触发，所以最后触发的时间标记值（最后触发的）被保存作为信息缓存 0 接收到信息的时间。

注意事项 时间标记功能在信息缓存 0 接收到数据帧的时候使用 **TSLOCK** 位停止对 **TSOUT** 信号的触发。因此，信息缓存 0 必须设定为接收信息缓存。由于接收信息缓存不能接收远程帧，不能使用这种方式来解除对 **TSOUT** 信号的触发。当除去信息缓存 0 的其它信息缓存接收到数据帧时，对 **TSOUT** 信号的触发操作不会停止。

由于这样的原因，当 **CAN** 模块处于带有 **ABT** 的正常操作模式时，信息缓存 0 不能接收数据帧，因为信息缓存 0 必须被设定为发送信息缓存。因此在这种操作模式中，不能使用通过 **TSLOCK** 位停止触发 **TSOUT** 信号的功能。

19.15 波特率设定

19.15.1 位速率设定条件

确保该设定在限制值范围之内以保证 CAN 控制器正常工作，具体如下。

	(a) $5TQ \leq SPT$ （采样点） $\leq 17TQ$
<R>	$SPT = TSEG1 + 1TQ$
	(b) $8TQ \leq DBT$ （数据位时间） $\leq 25TQ$
	$DBT = TSEG1 + TSEG2 + 1TQ = TSEG2 + SPT$
	(c) $1TQ \leq SJW$ （同步跳跃宽度） $\leq 4TQ$
	$SJW \leq DBT - SPT$
<R>	(d) $4TQ \leq TSEG1 \leq 16TQ$ [3 ≤ TSEG1[3:0] 的设定值 ≤ 15]
<R>	(e) $1TQ \leq TSEG2 \leq 8TQ$ [0 ≤ TSEG2[2:0] 的设定值 ≤ 7]
备注	$TQ = 1/frq$ （frq: CAN 协议层基本系统时钟）
	TSEG1[3:0]（C0BTR.TSEG13 到 C0BTR.TSEG10 位）
	TSEG2[2:0]（C0BTR.TSEG22 到 C0BTR.TSEG20 位）

表 19-22 显示了满足上述条件的波特率的组合。

表 19-22. 可设定的波特率 (1/3)

有效的波特率设定					COBTR 寄存器的设定值		采样点 (单位 %)
DBT 长度	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG13 到 TSEG10	TSEG22 到 TSEG20	
25	1	8	8	8	1111	111	68.0
24	1	7	8	8	1110	111	66.7
24	1	9	7	7	1111	110	70.8
23	1	6	8	8	1101	111	65.2
23	1	8	7	7	1110	110	69.6
23	1	10	6	6	1111	101	73.9
22	1	5	8	8	1100	111	63.6
22	1	7	7	7	1101	110	68.2
22	1	9	6	6	1110	101	72.7
22	1	11	5	5	1111	100	77.3
21	1	4	8	8	1011	111	61.9
21	1	6	7	7	1100	110	66.7
21	1	8	6	6	1101	101	71.4
21	1	10	5	5	1110	100	76.2
21	1	12	4	4	1111	011	81.0
20	1	3	8	8	1010	111	60.0
20	1	5	7	7	1011	110	65.0
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4

表 19-22. 可设定的波特率 (2/3)

有效的波特率设定					COBTR 寄存器的设定值		采样点 (单位 %)
DBT 长度	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG13 到 TSEG10	TSEG22 到 TSEG20	
17	1	2	7	7	1000	110	58.8
17	1	4	6	6	1001	101	64.7
17	1	6	5	5	1010	100	70.6
17	1	8	4	4	1011	011	76.5
17	1	10	3	3	1100	010	82.4
17	1	12	2	2	1101	001	88.2
17	1	14	1	1	1110	000	94.1
16	1	1	7	7	0111	110	56.3
16	1	3	6	6	1000	101	62.5
16	1	5	5	5	1001	100	68.8
16	1	7	4	4	1010	011	75.0
16	1	9	3	3	1011	010	81.3
16	1	11	2	2	1100	001	87.5
16	1	13	1	1	1101	000	93.8
15	1	2	6	6	0111	101	60.0
15	1	4	5	5	1000	100	66.7
15	1	6	4	4	1001	011	73.3
15	1	8	3	3	1010	010	80.0
15	1	10	2	2	1011	001	86.7
15	1	12	1	1	1100	000	93.3
14	1	1	6	6	0110	101	57.1
14	1	3	5	5	0111	100	64.3
14	1	5	4	4	1000	011	71.4
14	1	7	3	3	1001	010	78.6
14	1	9	2	2	1010	001	85.7
14	1	11	1	1	1011	000	92.9
13	1	2	5	5	0110	100	61.5
13	1	4	4	4	0111	011	69.2
13	1	6	3	3	1000	010	76.9
13	1	8	2	2	1001	001	84.6
13	1	10	1	1	1010	000	92.3
12	1	1	5	5	0101	100	58.3
12	1	3	4	4	0110	011	66.7
12	1	5	3	3	0111	010	75.0
12	1	7	2	2	1000	001	83.3
12	1	9	1	1	1001	000	91.7

表 19-22. 可设定的波特率 (3/3)

有效的波特率设定					C0BTR 寄存器的设定值		采样点 (单位 %)
DBT 长度	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG13 到 TSEG10	TSEG22 到 TSEG20	
11	1	2	4	4	0101	011	63.6
11	1	4	3	3	0110	010	72.7
11	1	6	2	2	0111	001	81.8
11	1	8	1	1	1000	000	90.9
10	1	1	4	4	0100	011	60.0
10	1	3	3	3	0101	010	70.0
10	1	5	2	2	0110	001	80.0
10	1	7	1	1	0111	000	90.0
9	1	2	3	3	0100	010	66.7
9	1	4	2	2	0101	001	77.8
9	1	6	1	1	0110	000	88.9
8	1	1	3	3	0011	010	62.5
8	1	3	2	2	0100	001	75.0
8	1	5	1	1	0101	000	87.5
7 ^注	1	2	2	2	0011	001	71.4
7 ^注	1	4	1	1	0100	000	85.7
6 ^注	1	1	2	2	0010	001	66.7
6 ^注	1	3	1	1	0011	000	83.3
5 ^注	1	2	1	1	0010	000	80.0
4 ^注	1	1	1	1	0001	000	75.0

注 仅当 C0BRP 寄存器的值不为 00H 时，DBT 的设定值不大于 7 是有效的。

注意事项 表 19-22 中的值并不能保证网络系统的操作。彻底检测对网络系统的影响，要考虑振动错误和 CAN 总线以及 CAN 发送器的延迟。

19.15.2 波特率设定的典型实例

表 19-23 和 19-24 显示了波特率设定的典型实例。

表 19-23. 波特率设定的典型实例 ($f_{CANMOD} = 8 \text{ MHz}$) (1/2)

波特率的设定值 (单位: kbps)	C0BRP 寄存器的分频比	C0BRP 寄存器的设定值	有效的波特率设定 (单位: kbps)					C0BTR 寄存器的设定值		采样点 (单位: %)
			DBT 的长度	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG13 到 TSEG10	TSEG22 到 TSEG20	
1000	1	00000000	8	1	1	3	3	0011	010	62.5
1000	1	00000000	8	1	3	2	2	0100	001	75.0
1000	1	00000000	8	1	5	1	1	0101	000	87.5
500	1	00000000	16	1	1	7	7	0111	110	56.3
500	1	00000000	16	1	3	6	6	1000	101	62.5
500	1	00000000	16	1	5	5	5	1001	100	68.8
500	1	00000000	16	1	7	4	4	1010	011	75.0
500	1	00000000	16	1	9	3	3	1011	010	81.3
500	1	00000000	16	1	11	2	2	1100	001	87.5
500	1	00000000	16	1	13	1	1	1101	000	93.8
500	2	00000001	8	1	1	3	3	0011	010	62.5
500	2	00000001	8	1	3	2	2	0100	001	75.0
500	2	00000001	8	1	5	1	1	0101	000	87.5
250	2	00000001	16	1	1	7	7	0111	110	56.3
250	2	00000001	16	1	3	6	6	1000	101	62.5
250	2	00000001	16	1	5	5	5	1001	100	68.8
250	2	00000001	16	1	7	4	4	1010	011	75.0
250	2	00000001	16	1	9	3	3	1011	010	81.3
250	2	00000001	16	1	11	2	2	1100	001	87.5
250	2	00000001	16	1	13	1	1	1101	000	93.8
250	4	00000011	8	1	3	2	2	0100	001	75.0
250	4	00000011	8	1	5	1	1	0101	000	87.5
125	4	00000011	16	1	1	7	7	0111	110	56.3
125	4	00000011	16	1	3	6	6	1000	101	62.5
125	4	00000011	16	1	5	5	5	1001	100	68.8
125	4	00000011	16	1	7	4	4	1010	011	75.0
125	4	00000011	16	1	9	3	3	1011	010	81.3
125	4	00000011	16	1	11	2	2	1100	001	87.5
125	4	00000011	16	1	13	1	1	1101	000	93.8
125	8	00000111	8	1	3	2	2	0100	001	75.0
125	8	00000111	8	1	5	1	1	0101	000	87.5

注意事项 表 19-23 中的值并不能保证网络系统的操作。彻底检测对网络系统的影响, 要考虑振动错误和 CAN 总线以及 CAN 发送器的延迟。

表 19-23. 波特率设定的典型实例 (f_{CANMOD} = 8 MHz) (2/2)

波特率的设定 值 (单位: kbps)	COBRP 寄存 器的分频比	COBRP 寄存器 的设定值	有效的波特率设定 (单位: kbps)					COBTR 寄存器的设定值		采样点 (单位: %)
			DBT 的长度	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG13 到 TSEG10	TSEG22 到 TSEG20	
100	4	00000011	20	1	7	6	6	1100	101	70.0
100	4	00000011	20	1	9	5	5	1101	100	75.0
100	5	00000100	16	1	7	4	4	1010	011	75.0
100	5	00000100	16	1	9	3	3	1011	010	81.3
100	8	00000111	10	1	3	3	3	0101	010	70.0
100	8	00000111	10	1	5	2	2	0110	001	80.0
100	10	00001001	8	1	3	2	2	0100	001	75.0
100	10	00001001	8	1	5	1	1	0101	000	87.5
83.3	4	00000011	24	1	7	8	8	1110	111	66.7
83.3	4	00000011	24	1	9	7	7	1111	110	70.8
83.3	6	00000101	16	1	5	5	5	1001	100	68.8
83.3	6	00000101	16	1	7	4	4	1010	011	75.0
83.3	6	00000101	16	1	9	3	3	1011	010	81.3
83.3	6	00000101	16	1	11	2	2	1100	001	87.5
83.3	8	00000111	12	1	5	3	3	0111	010	75.0
83.3	8	00000111	12	1	7	2	2	1000	001	83.3
83.3	12	00001011	8	1	3	2	2	0100	001	75.0
83.3	12	00001011	8	1	5	1	1	0101	000	87.5
33.3	10	00001001	24	1	7	8	8	1110	111	66.7
33.3	10	00001001	24	1	9	7	7	1111	110	70.8
33.3	12	00001011	20	1	7	6	6	1100	101	70.0
33.3	12	00001011	20	1	9	5	5	1101	100	75.0
33.3	15	00001110	16	1	7	4	4	1010	011	75.0
33.3	15	00001110	16	1	9	3	3	1011	010	81.3
33.3	16	00001111	15	1	6	4	4	1001	011	73.3
33.3	16	00001111	15	1	8	3	3	1010	010	80.0
33.3	20	00010011	12	1	5	3	3	0111	010	75.0
33.3	20	00010011	12	1	7	2	2	1000	001	83.3
33.3	24	00010111	10	1	3	3	3	0101	010	70.0
33.3	24	00010111	10	1	5	2	2	0110	001	80.0
33.3	30	00011101	8	1	3	2	2	0100	001	75.0
33.3	30	00011101	8	1	5	1	1	0101	000	87.5

注意事项 表 19-23 中的值并不能保证网络系统的操作。彻底检测对网络系统的影响, 要考虑振动错误和 CAN 总线以及 CAN 发送器的延迟。

表 19-24. 波特率设定的典型实例 (f_{CANMOD} = 16 MHz) (1/2)

波特率的设定值 (单位: kbps)	C0BRP 寄存器的分频比	C0BRP 寄存器的设定值	有效的波特率设定 (单位: kbps)					C0BTR 寄存器的设定值		采样点 (单位: %)
			DBT 的长度	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG13 到 TSEG10	TSEG22 到 TSEG20	
1000	1	00000000	16	1	1	7	7	0111	110	56.3
1000	1	00000000	16	1	3	6	6	1000	101	62.5
1000	1	00000000	16	1	5	5	5	1001	100	68.8
1000	1	00000000	16	1	7	4	4	1010	011	75.0
1000	1	00000000	16	1	9	3	3	1011	010	81.3
1000	1	00000000	16	1	11	2	2	1100	001	87.5
1000	1	00000000	16	1	13	1	1	1101	000	93.8
1000	2	00000001	8	1	3	2	2	0100	001	75.0
1000	2	00000001	8	1	5	1	1	0101	000	87.5
500	2	00000001	16	1	1	7	7	0111	110	56.3
500	2	00000001	16	1	3	6	6	1000	101	62.5
500	2	00000001	16	1	5	5	5	1001	100	68.8
500	2	00000001	16	1	7	4	4	1010	011	75.0
500	2	00000001	16	1	9	3	3	1011	010	81.3
500	2	00000001	16	1	11	2	2	1100	001	87.5
500	2	00000001	16	1	13	1	1	1101	000	93.8
500	4	00000011	8	1	3	2	2	0100	001	75.0
500	4	00000011	8	1	5	1	1	0101	000	87.5
250	4	00000011	16	1	3	6	6	1000	101	62.5
250	4	00000011	16	1	5	5	5	1001	100	68.8
250	4	00000011	16	1	7	4	4	1010	011	75.0
250	4	00000011	16	1	9	3	3	1011	010	81.3
250	4	00000011	16	1	11	2	2	1100	001	87.5
250	8	00000111	8	1	3	2	2	0100	001	75.0
250	8	00000111	8	1	5	1	1	0101	000	87.5
125	8	00000111	16	1	3	6	6	1000	101	62.5
125	8	00000111	16	1	7	4	4	1010	011	75.0
125	8	00000111	16	1	9	3	3	1011	010	81.3
125	8	00000111	16	1	11	2	2	1100	001	87.5
125	16	00001111	8	1	3	2	2	0100	001	75.0
125	16	00001111	8	1	5	1	1	0101	000	87.5

注意事项 表 19-24 中的值并不能保证网络系统的操作。彻底检测对网络系统的影响, 要考虑振动错误和 CAN 总线以及 CAN 发送器的延迟。

表 19-24. 波特率设定的典型实例 (f_{CANMOD} = 16 MHz) (2/2)

波特率的设定 值 (单位: kbps)	COBRP 寄存 器的分频比	COBRP 寄存 器的设定值	有效的波特率设定 (单位: kbps)					COBTR 寄存器的设定值		采样点 (单位: %)
			DBT 的长度	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG13 到 TSEG10	TSEG22 到 TSEG20	
100	8	00000111	20	1	9	5	5	1101	100	75.0
100	8	00000111	20	1	11	4	4	1110	011	80.0
100	10	00001001	16	1	7	4	4	1010	011	75.0
100	10	00001001	16	1	9	3	3	1011	010	81.3
100	16	00001111	10	1	3	3	3	0101	010	70.0
100	16	00001111	10	1	5	2	2	0110	001	80.0
100	20	00010011	8	1	3	2	2	0100	001	75.0
83.3	8	00000111	24	1	7	8	8	1110	111	66.7
83.3	8	00000111	24	1	9	7	7	1111	110	70.8
83.3	12	00001011	16	1	7	4	4	1010	011	75.0
83.3	12	00001011	16	1	9	3	3	1011	010	81.3
83.3	12	00001011	16	1	11	2	2	1100	001	87.5
83.3	16	00001111	12	1	5	3	3	0111	010	75.0
83.3	16	00001111	12	1	7	2	2	1000	001	83.3
83.3	24	00010111	8	1	3	2	2	0100	001	75.0
83.3	24	00010111	8	1	5	1	1	0101	000	87.5
33.3	30	00011101	24	1	7	8	8	1110	111	66.7
33.3	30	00011101	24	1	9	7	7	1111	110	70.8
33.3	24	00010111	20	1	9	5	5	1101	100	75.0
33.3	24	00010111	20	1	11	4	4	1110	011	80.0
33.3	30	00011101	16	1	7	4	4	1010	011	75.0
33.3	30	00011101	16	1	9	3	3	1011	010	81.3
33.3	32	00011111	15	1	8	3	3	1010	010	80.0
33.3	32	00011111	15	1	10	2	2	1011	001	86.7
33.3	37	00100100	13	1	6	3	3	1000	010	76.9
33.3	37	00100100	13	1	8	2	2	1001	001	84.6
33.3	40	00100111	12	1	5	3	3	0111	010	75.0
33.3	40	00100111	12	1	7	2	2	1000	001	83.3
33.3	48	00101111	10	1	3	3	3	0101	010	70.0
33.3	48	00101111	10	1	5	2	2	0110	001	80.0
33.3	60	00111011	8	1	3	2	2	0100	001	75.0
33.3	60	00111011	8	1	5	1	1	0101	000	87.5

注意事项 表 19-24 中的值并不能保证网络系统的操作。彻底检测对网络系统的影响, 要考虑振动错误和 CAN 总线以及 CAN 发送器的延迟。

19.16 CAN控制器的操作

下面显示的处理过程是操作 CAN 控制器的推荐过程。通过参考这个推荐的处理过程来开发程序。

备注 m = 00 到 31

图 19-36. 初始化

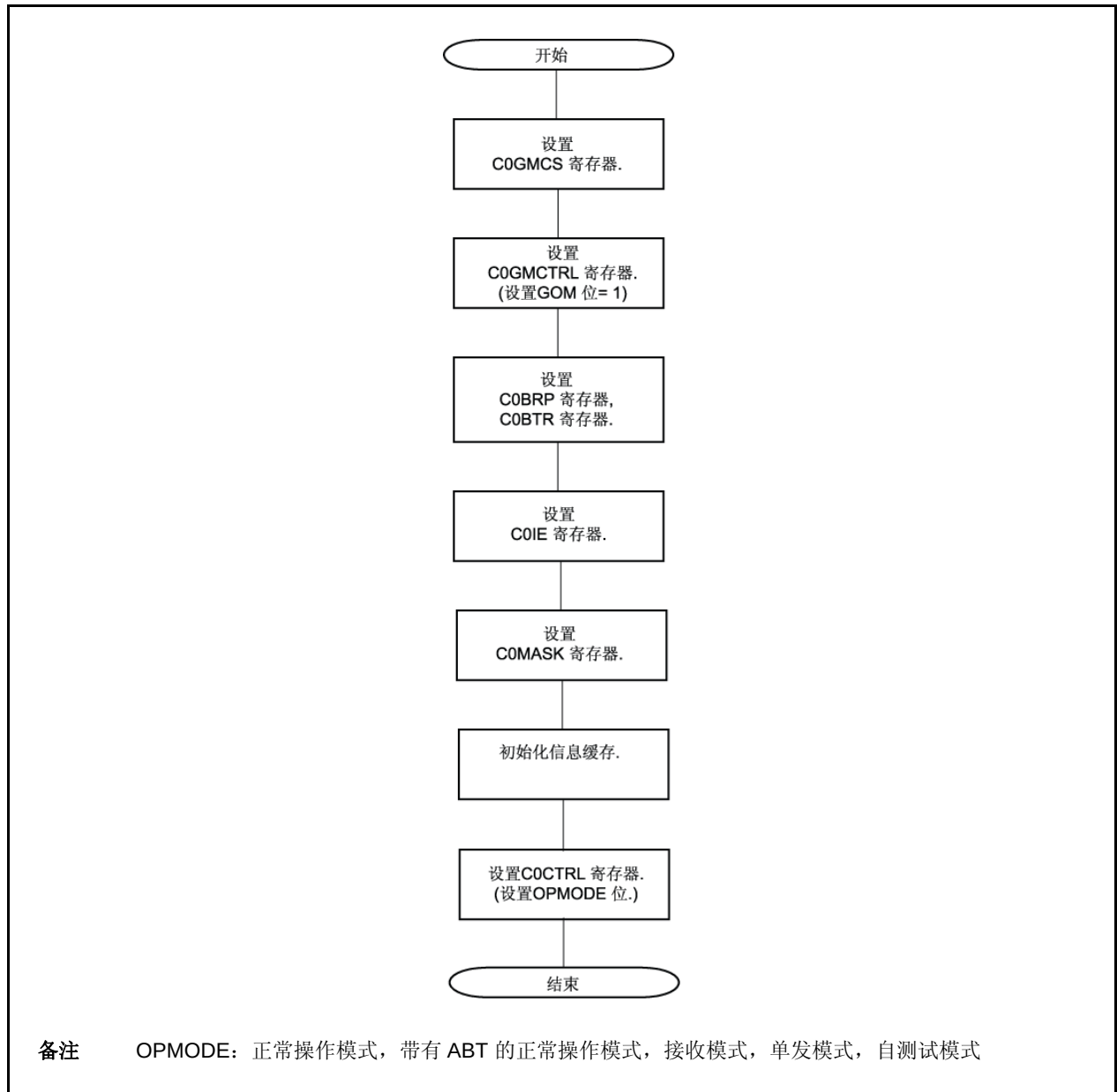
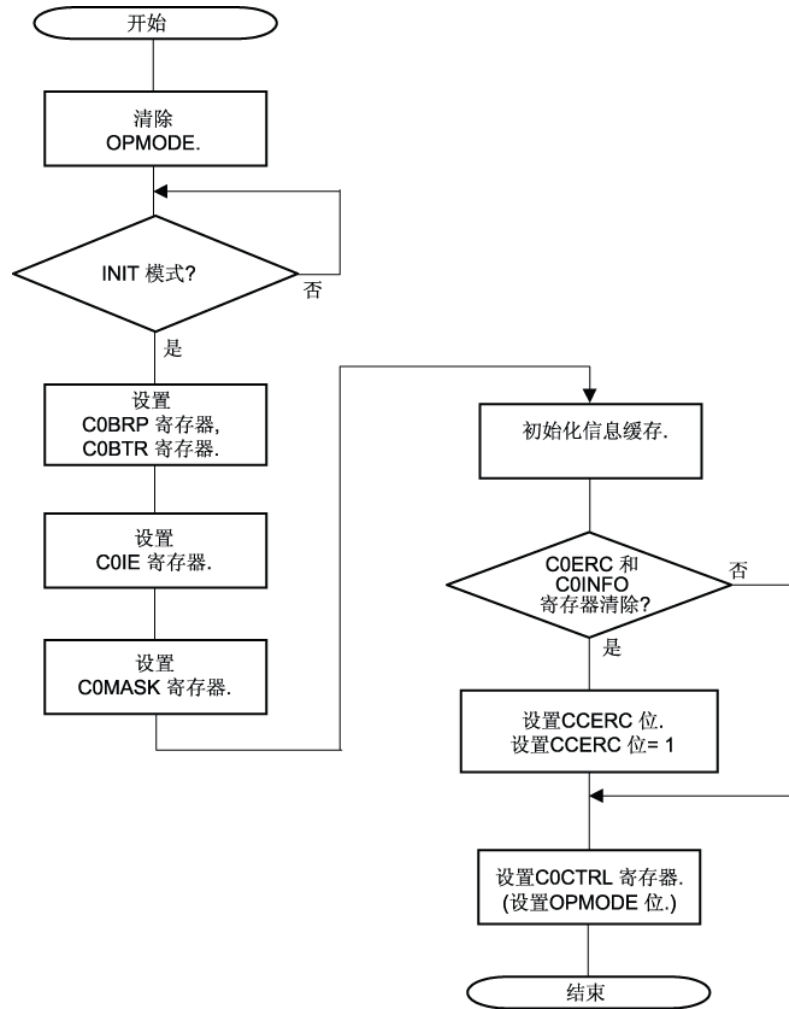


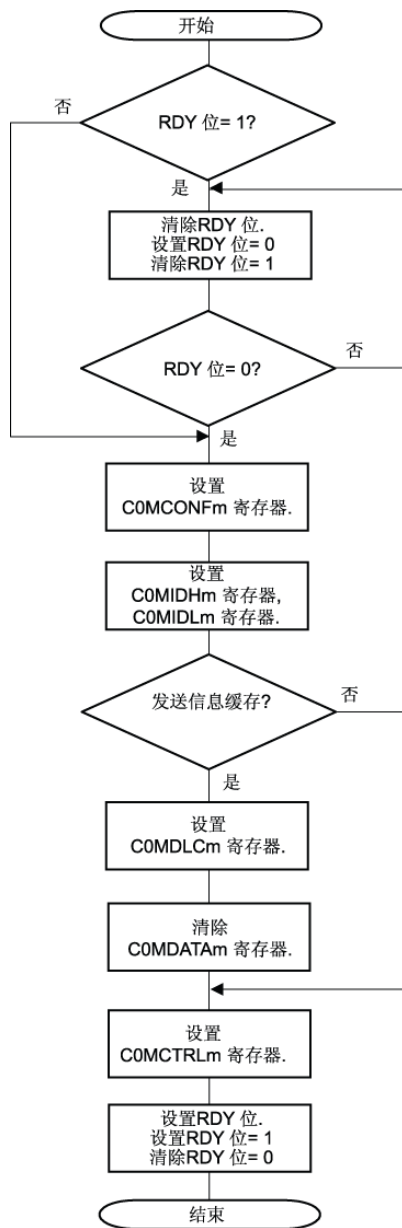
图 19-37. 重新初始化



注意事项 在设定 CAN 模块到初始化模式，应避免立即将该模块再设置为其它模式，不要对 C0CTRL 和 C0GMCTRL 寄存器进行访问（例如设定一个信息缓存）。

备注 OPMODE: 正常操作模式，带有 ABT 的正常操作模式，接收模式，单发模式，自测试模式

图 19-38. 信息缓存初始化



- 注意事项
- 在对信息缓存初始化之前，需要把 RDY 位清 0。
 - 在应用中不使用的信息缓存要按以下方式设定。
 - 将 COMCTRLm.RDY, COMCTRLm.TRQ 和 COMCTRLm.DN 位清 0。
 - 将 COMCONFm.MA0 位清 0。

图 19-40 显示了在发送过程中发送信息缓存的处理过程（COMCONFm 寄存器的 MT2 到 MT0 位为 000B）。

图 19-40. 在发送过程中信息缓存的重定义

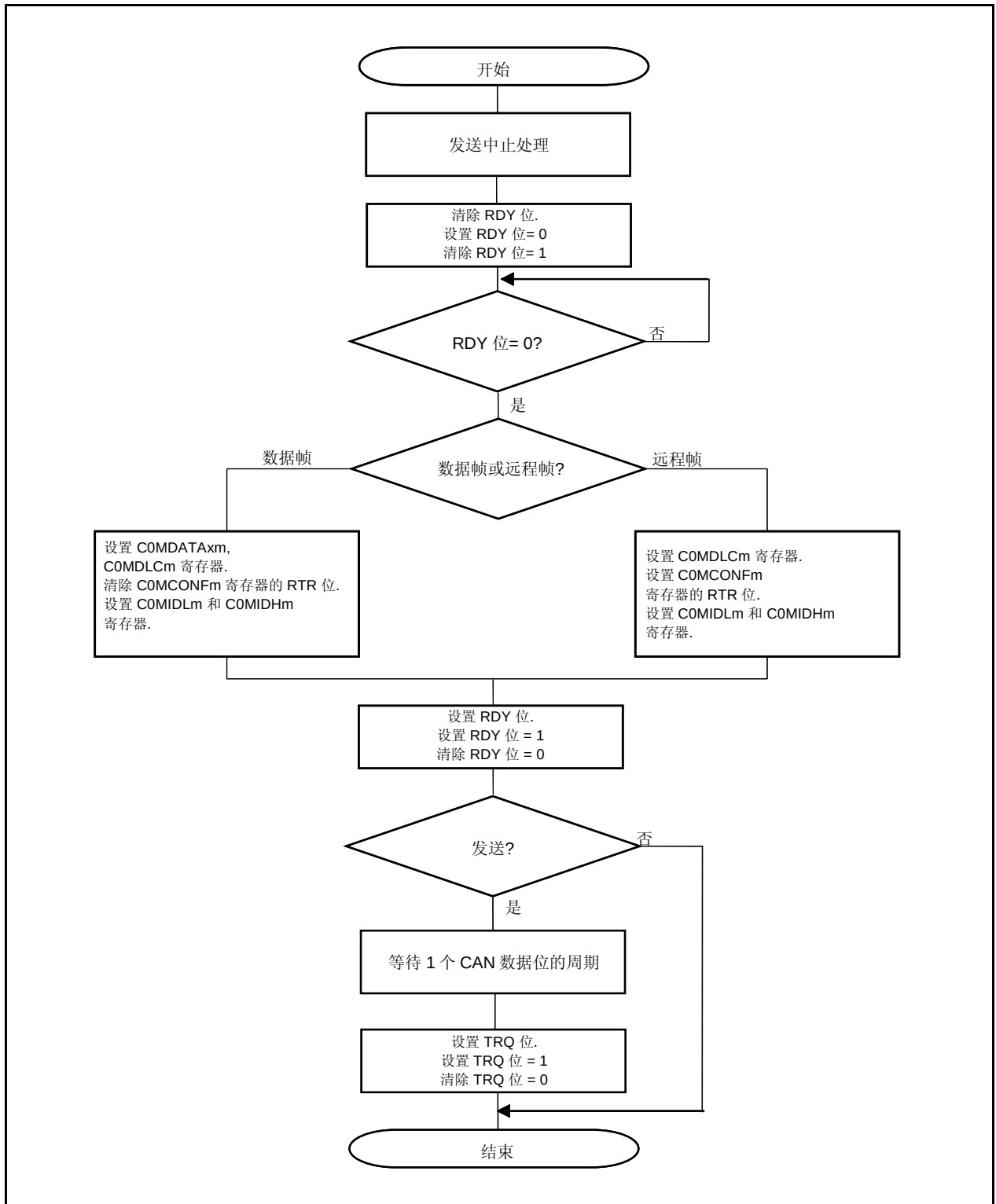
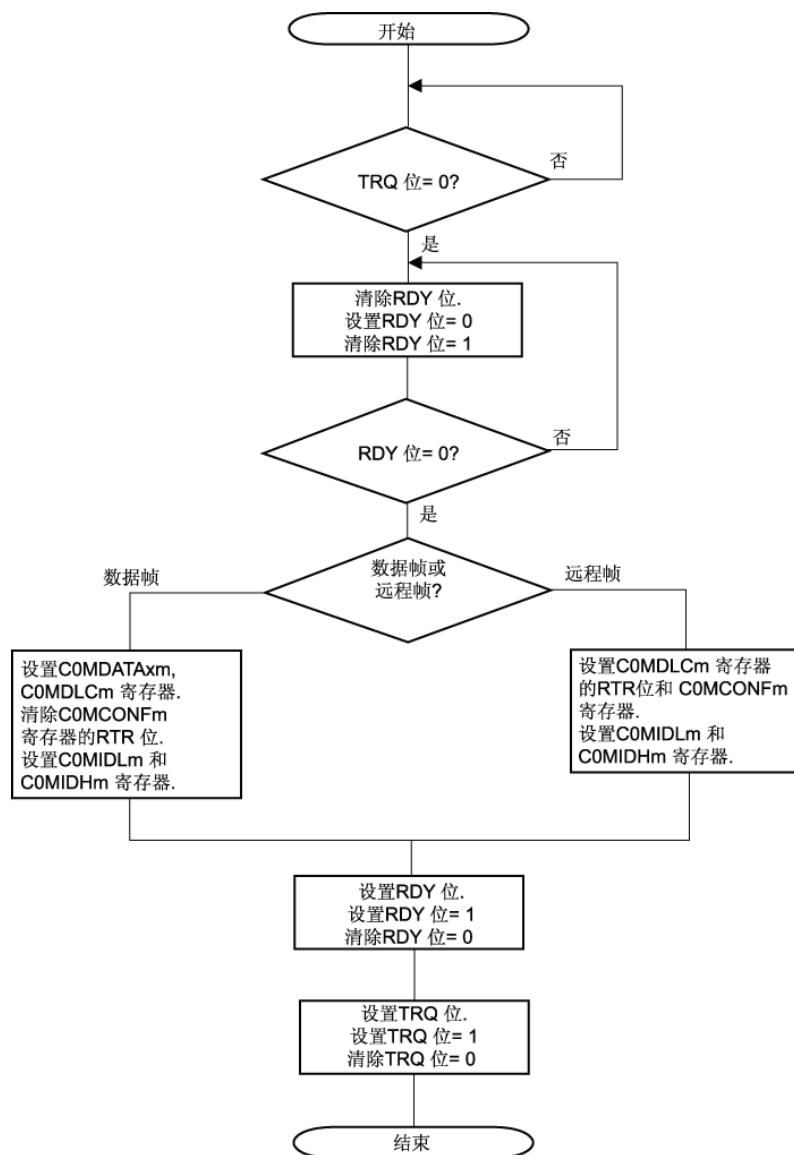


图 19-41 显示了发送信息缓存的处理过程（COMCONFm.MT2 到 COMCONFm.MT0 位为 000B）。

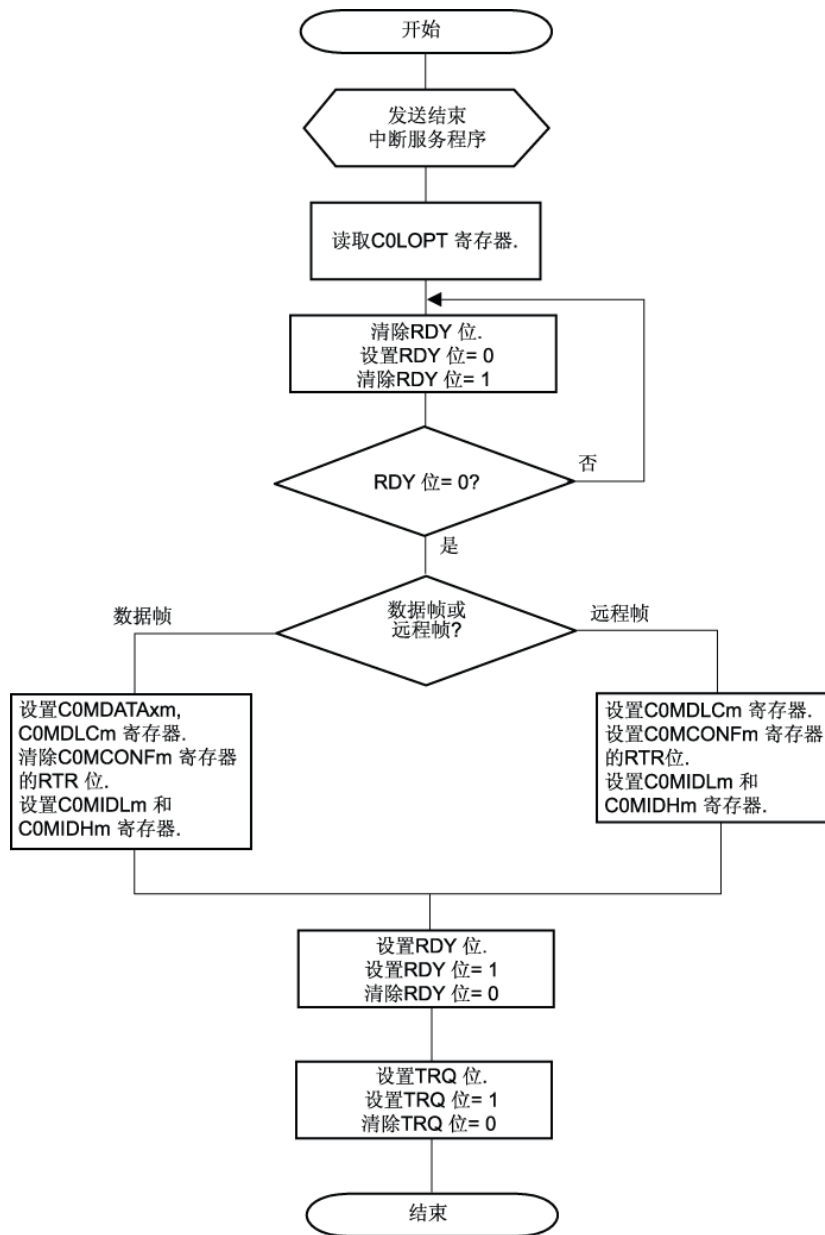
图 19-41. 消息发送过程



注意事项 1. 在 TRQ 位置 1 前，RDY 位应该立即置 1。

2. RDY 位和 TRQ 不能同时置 1。

图 19-43. 通过中断发送（使用 C0LOPT 寄存器）



- 注意事项**
1. 在 RDY 位置 1 后，TRQ 位应该立即置 1。
 2. RDY 位和 TRQ 不能同时置 1。

<R> 备注 在中断程序的开始和结束检查 MBON 位来看信息缓存和发送历史寄存器是否可以被访问，因为可能正在执行已经被挂起的 CAN 休眠模式转换。如果 MBON 位被清零，停止正在执行的处理。在 MBON 位被置位后，重新执行处理。因此，建议在执行发送中断服务程序前，取消 CAN 休眠模式转换请求。

图 19-44. 通过中断发送（使用 C0TGPT 寄存器）

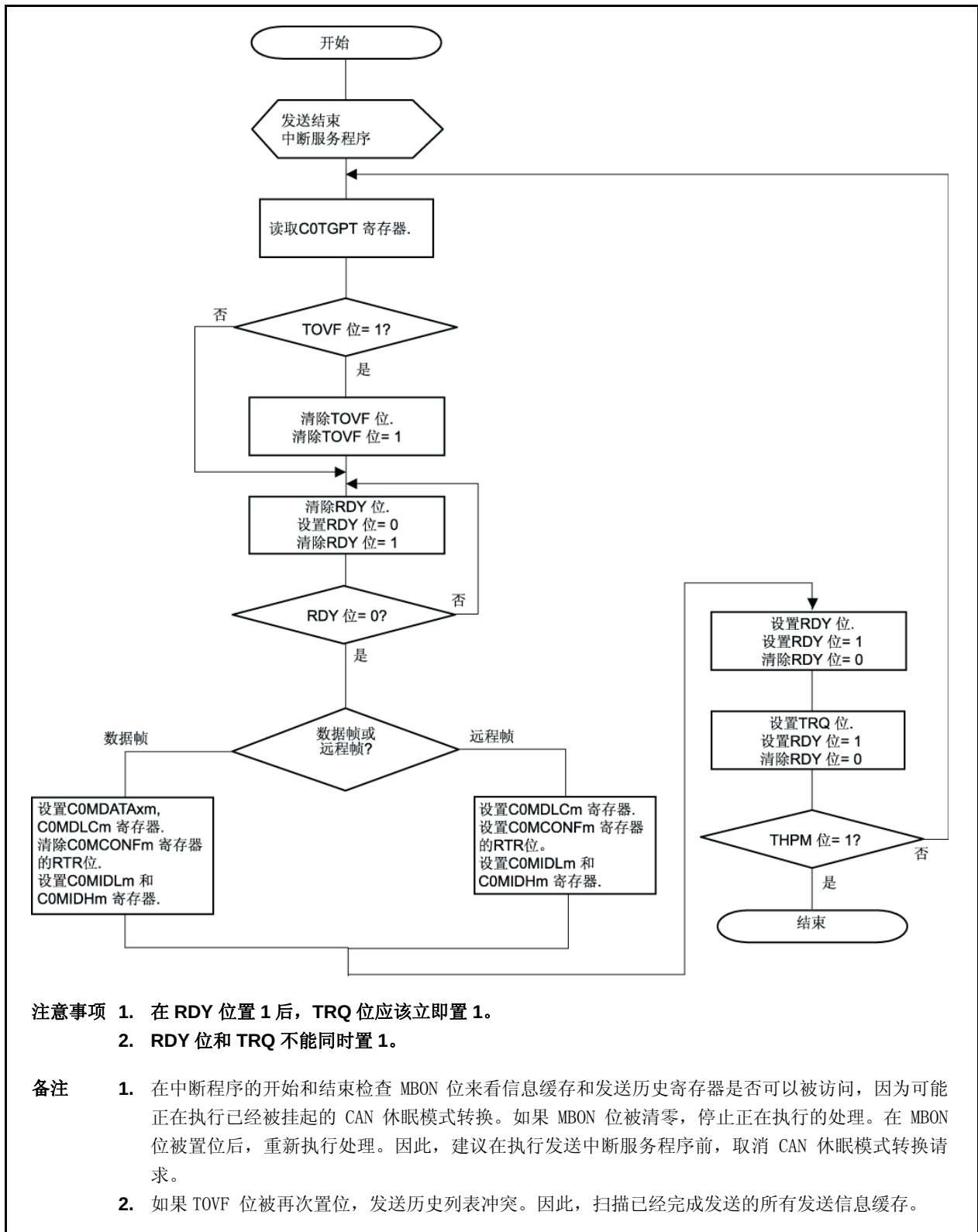
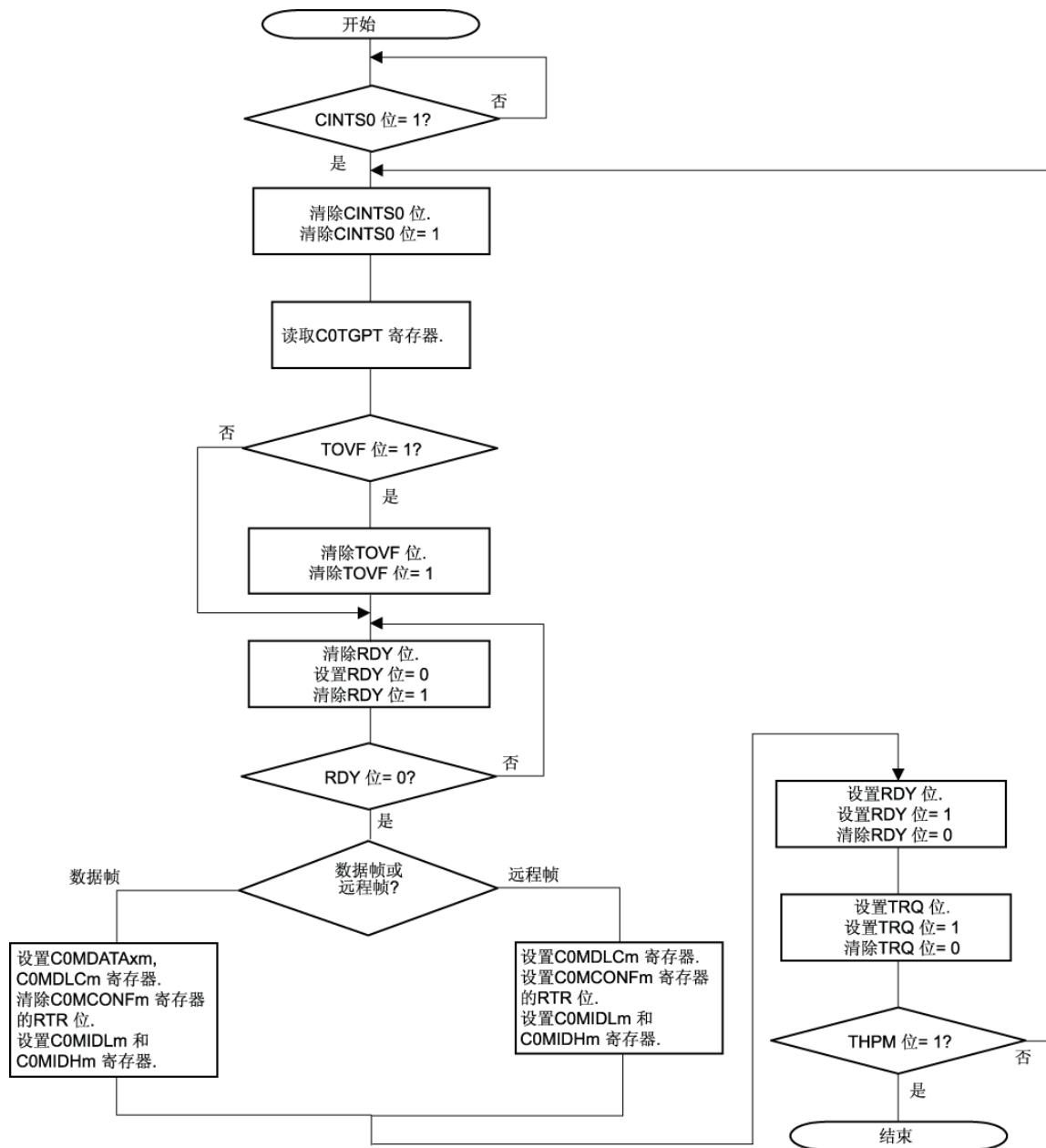


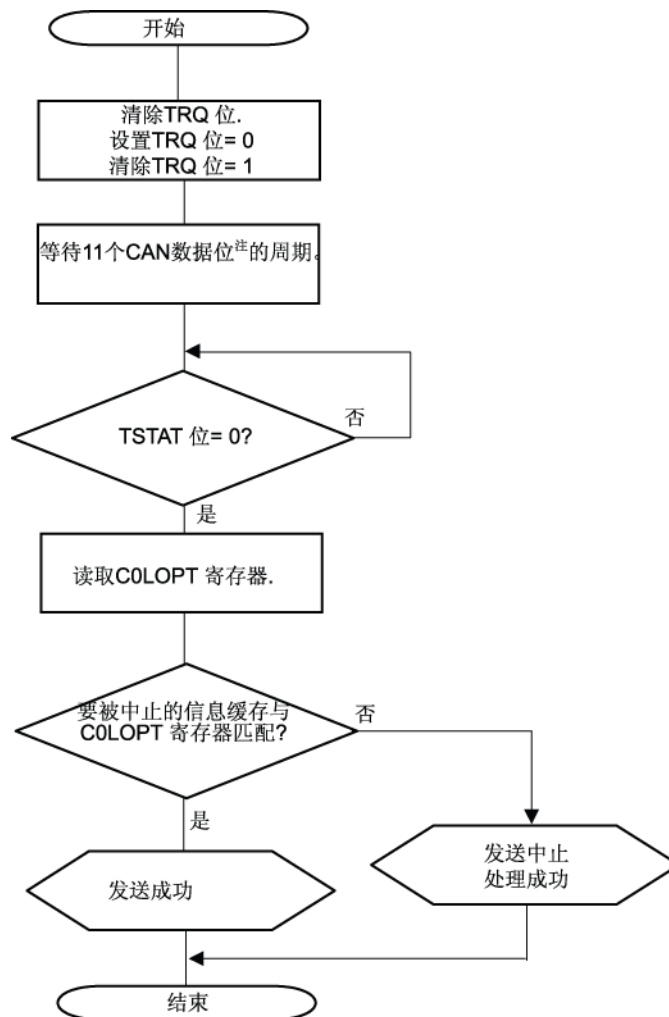
图 19-45. 通过软件轮询发送



- 注意事项**
1. 在 RDY 位置 1 后, TRQ 位应该立即置 1。
 2. RDY 位和 TRQ 不能同时置 1。

- <R> 备注**
1. 在轮询程序的开始和结束检查 MBON 位来看信息缓存和发送历史寄存器是否可以被访问, 因为可能正在执行已经被挂起的 CAN 休眠模式转换。如果 MBON 位被清零, 停止正在执行的处理。在 MBON 位被置位后, 重新执行处理。
 2. 如果 TOVF 位被再次置位, 发送历史列表冲突。因此, 扫描已经完成发送的所有发送信息缓存。

图 19-46. 发送中止处理（带 ABT 的正常操作模式以外）



注 在 3 位帧间空间的和挂起发送的 8 位总共 11 位周期中，协议层可能已经响应发送请求。因此，即使 TRQ 位被清除，发送也可能不被中止而是启动。

- 注意事项**
1. 通过将 TRQ 位而不是 RDY 清 0 来执行发送请求中止进程。
 2. 在发出一个休眠模式发送请求之前，要确认没有遗留的发送请求正在使用该进程。
 3. TSTAT 位可以通过用户的应用或是在发送完成中断后进行周期性的检测。
 4. 正在进行发送请求中止进程时，不要执行新的包含其它信息缓存发送请求。
 5. 如果相同信息缓存的数据被连续发送或者如果只使用一个信息缓存，判断发送执行成功或失败可能冲突。在这种情况下，通过使用 C0TGPT 寄存器的历史信息来作判断。

图 19-47. 除去 ABT 发送的发送中止进程
(带有 ABT 的正常操作模式)

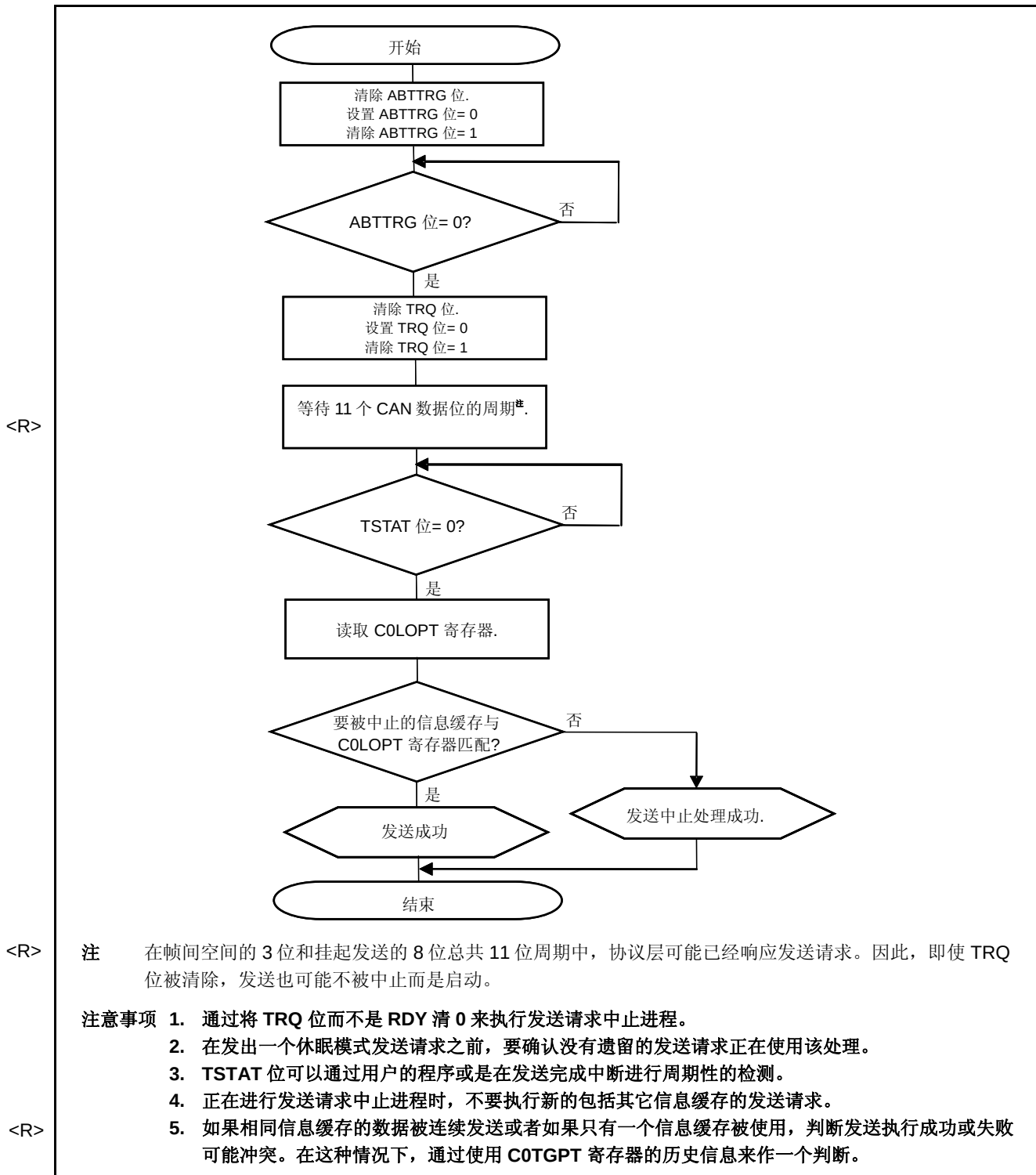


图 19-48 (a) 显示了当 ABT 信息缓存发送中止时，系统没有跳转恢复被中断的发送信息的过程。

图 19-48 (a) . 中止 ABT 发送处理 (带有 ABT 的正常操作模式)

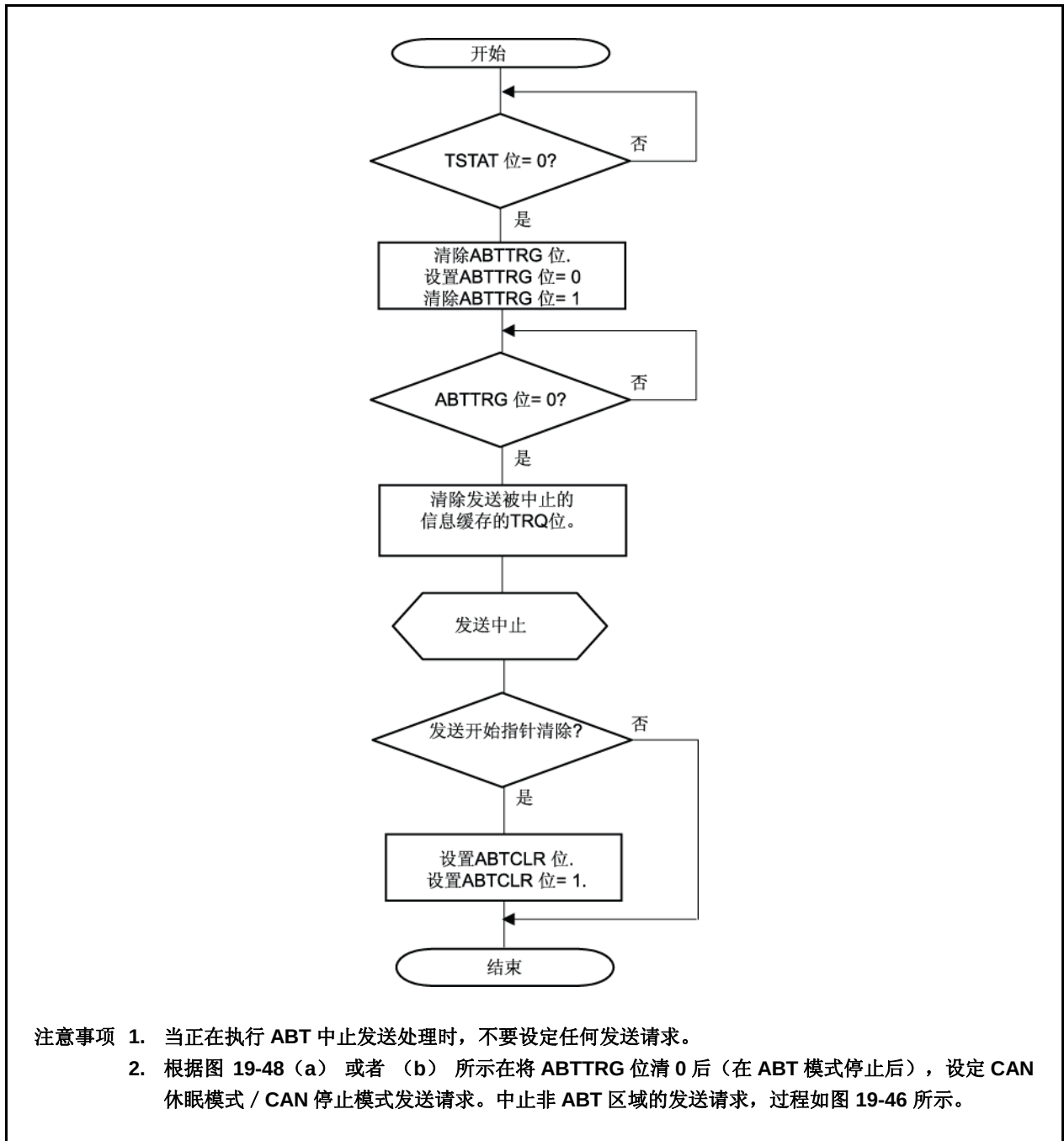
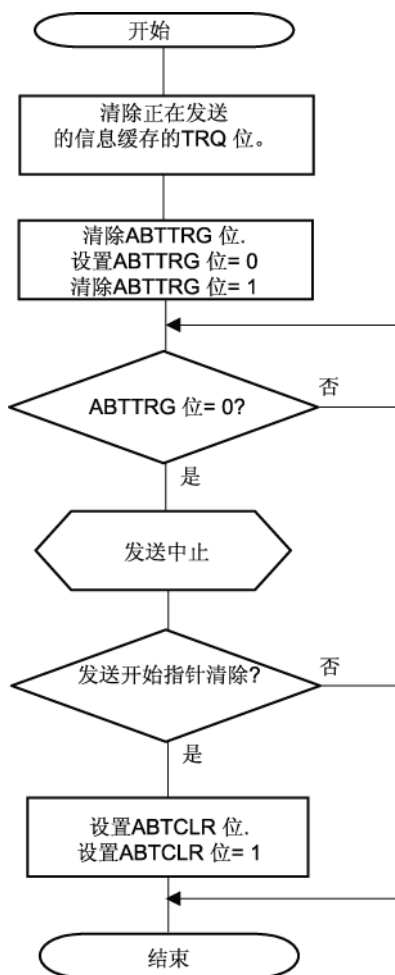


图 19-48 (b) 显示了当 ABT 信息缓存发送被中止时的处理，这里，系统没有跳过被中断的发送信息的恢复过程。

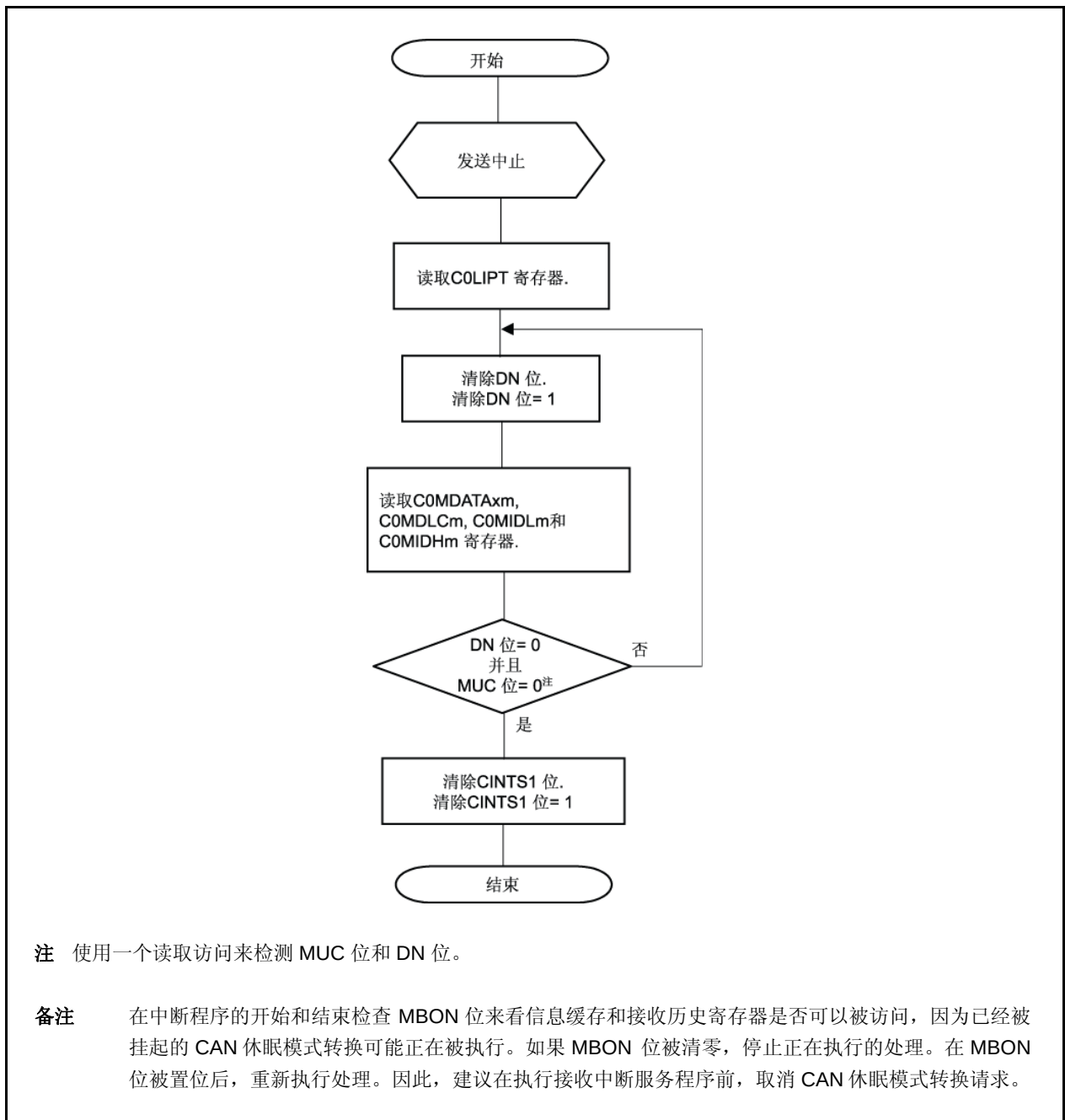
图 19-48 (b) . 中止 ABT 发送请求处理 (带有 ABT 的正常操作模式)



注意事项 1. 当正在执行 ABT 中止发送处理时，不要设定任何发送请求。

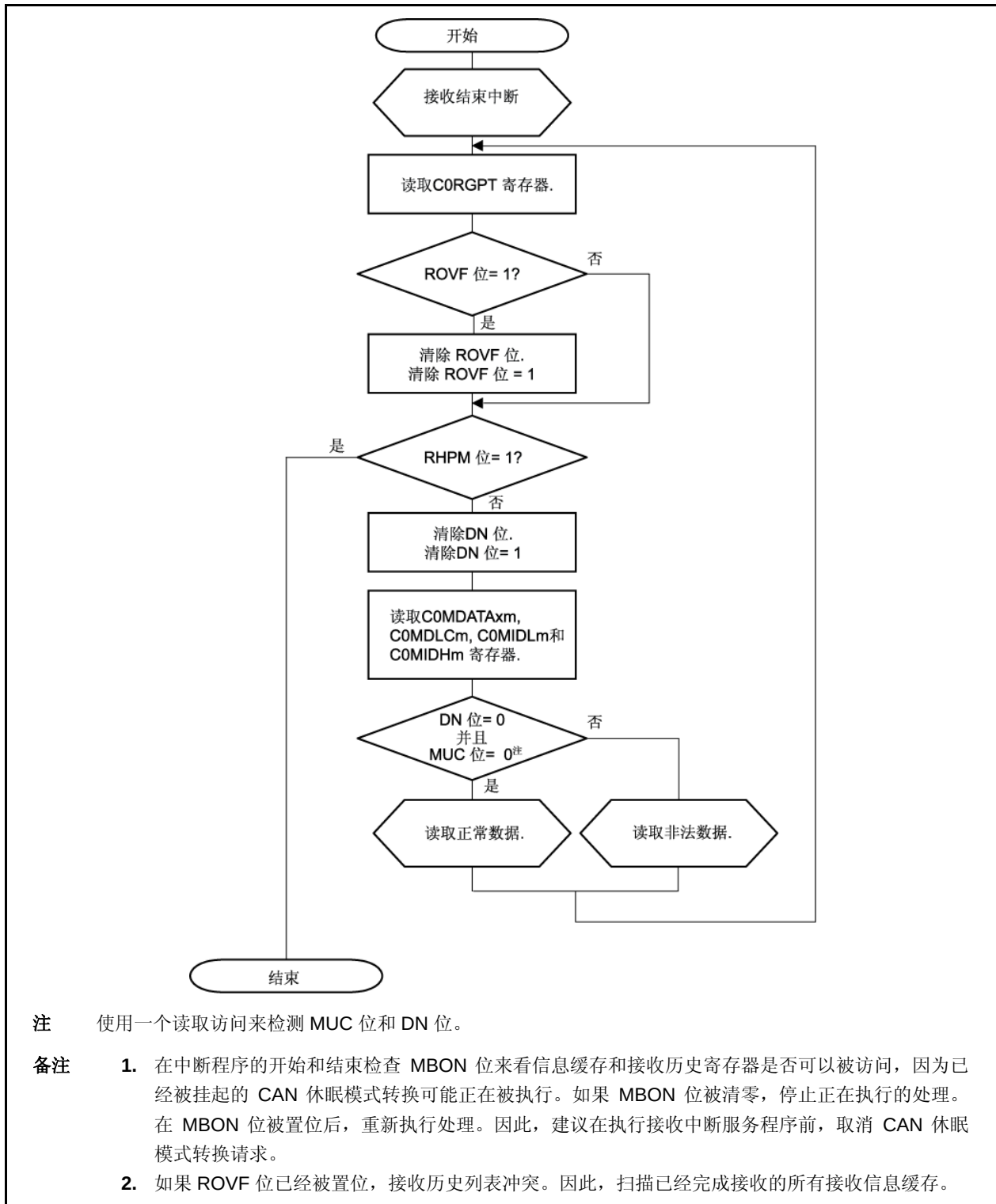
2. 根据图 19-48 (a) 或者 (b) 所示在将 ABTTRG 位清 0 后 (在 ABT 模式停止后)，设定 CAN 休眠模式 / CAN 停止模式发送请求的方法可以参考图 19-47 (a) 或者 (b)。中止非 ABT 区域的发送请求，过程如图 19-46 所示。

图 19-49. 通过中断接收（使用 C0LIPT 寄存器）



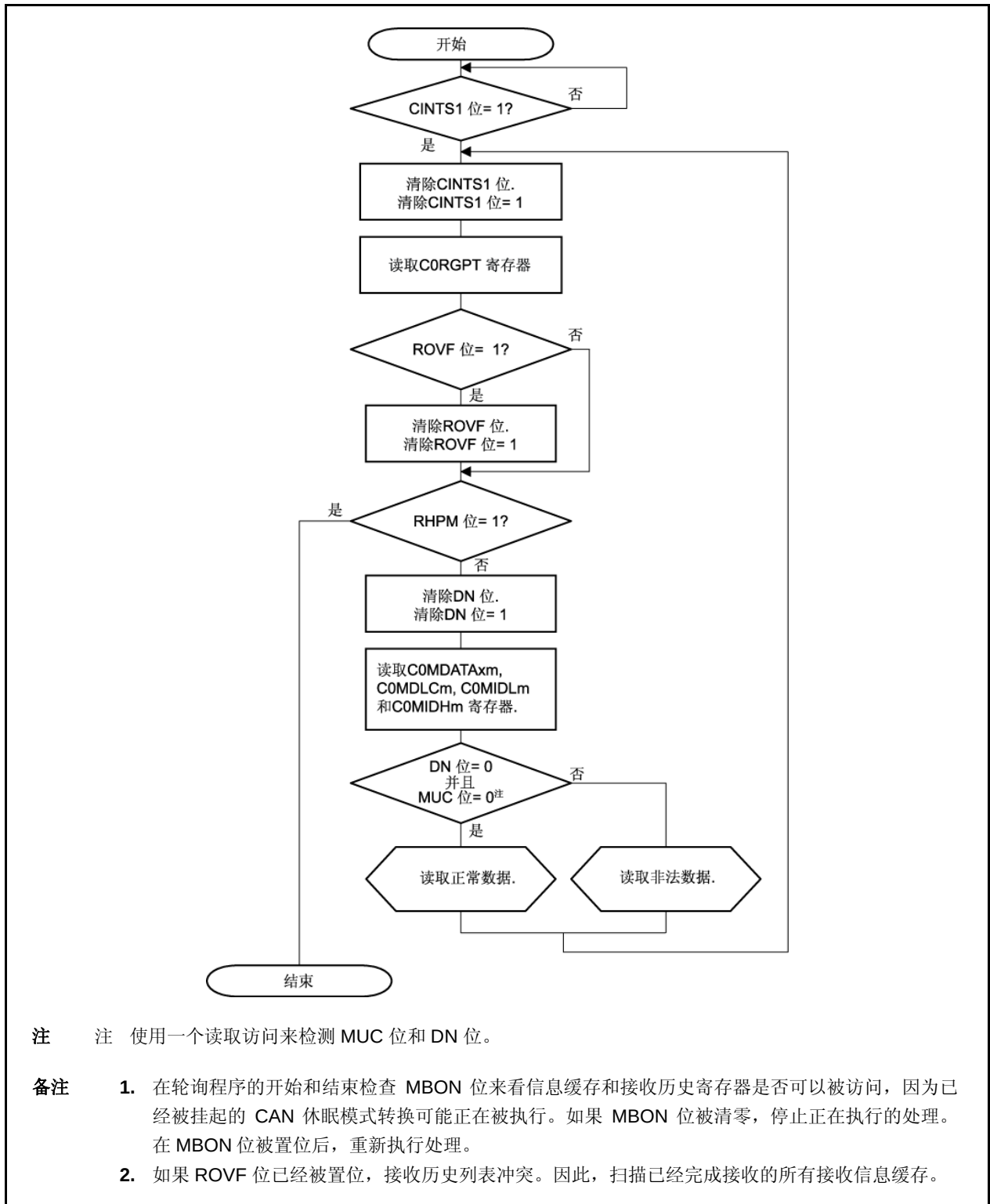
<R>

图 19-50. 通过中断接收（使用 C0RGPT 寄存器）



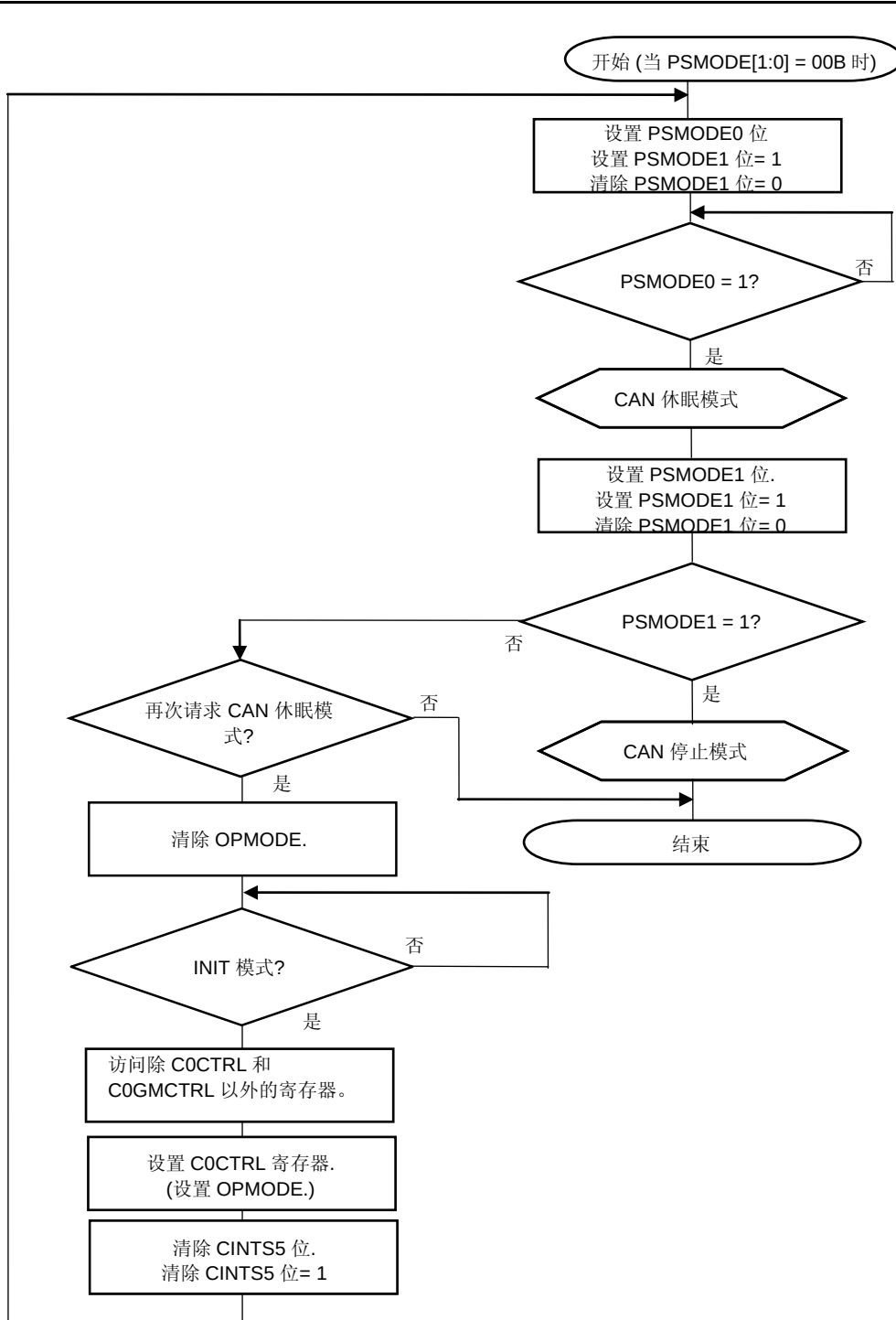
<R>

图 19-51. 通过软件轮询接收



<R>

图 19-52. 设定 CAN 休眠模式 / 停止模式



注意事项 为了在请求进入 CAN 休眠模式之前中止发送，按照图 19-46 和 19-48 给出的方式执行。

<R>

图 19-53. 清除 CAN 休眠 / 停止模式

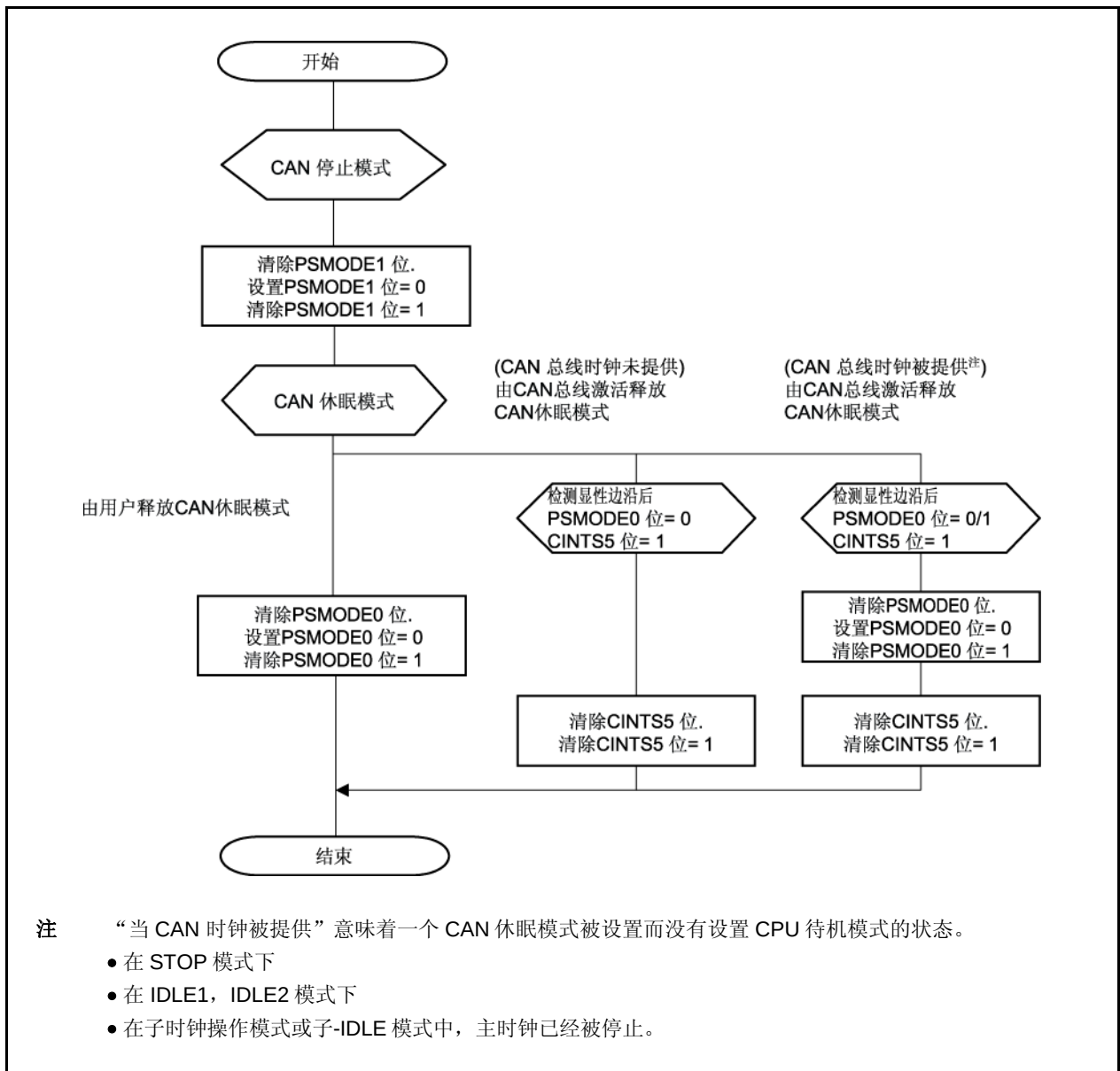
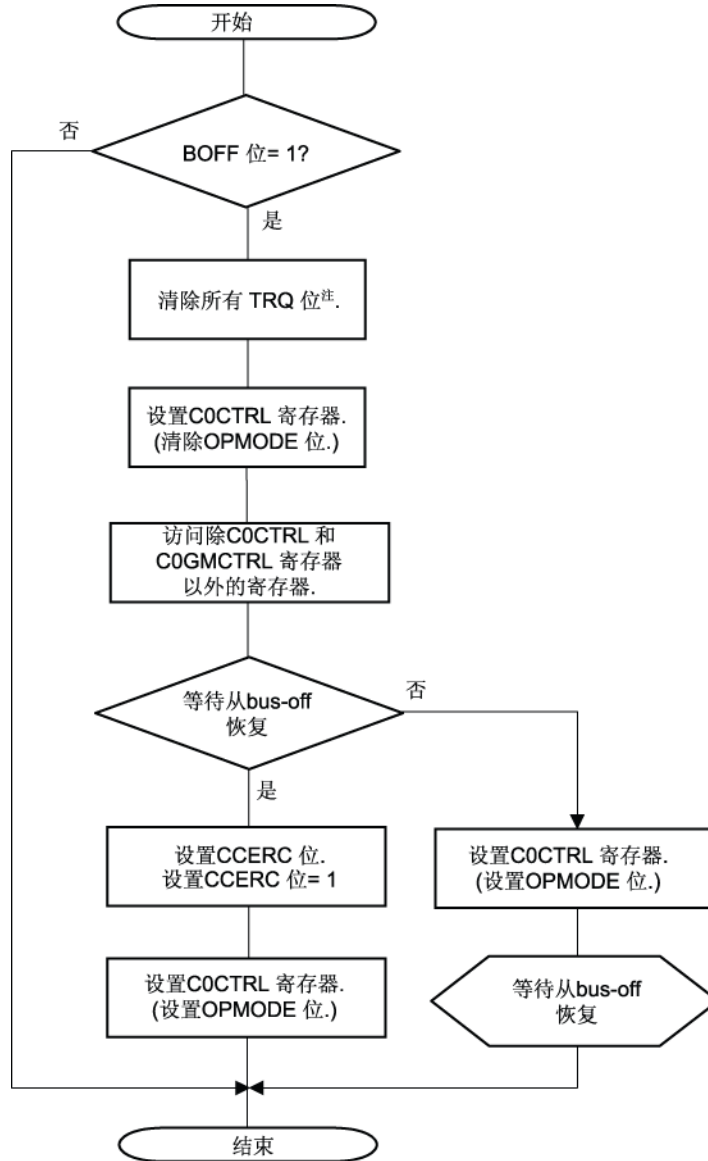


图 19-54. bus-off 恢复（带 ABT 的正常操作模式以外）

<R>



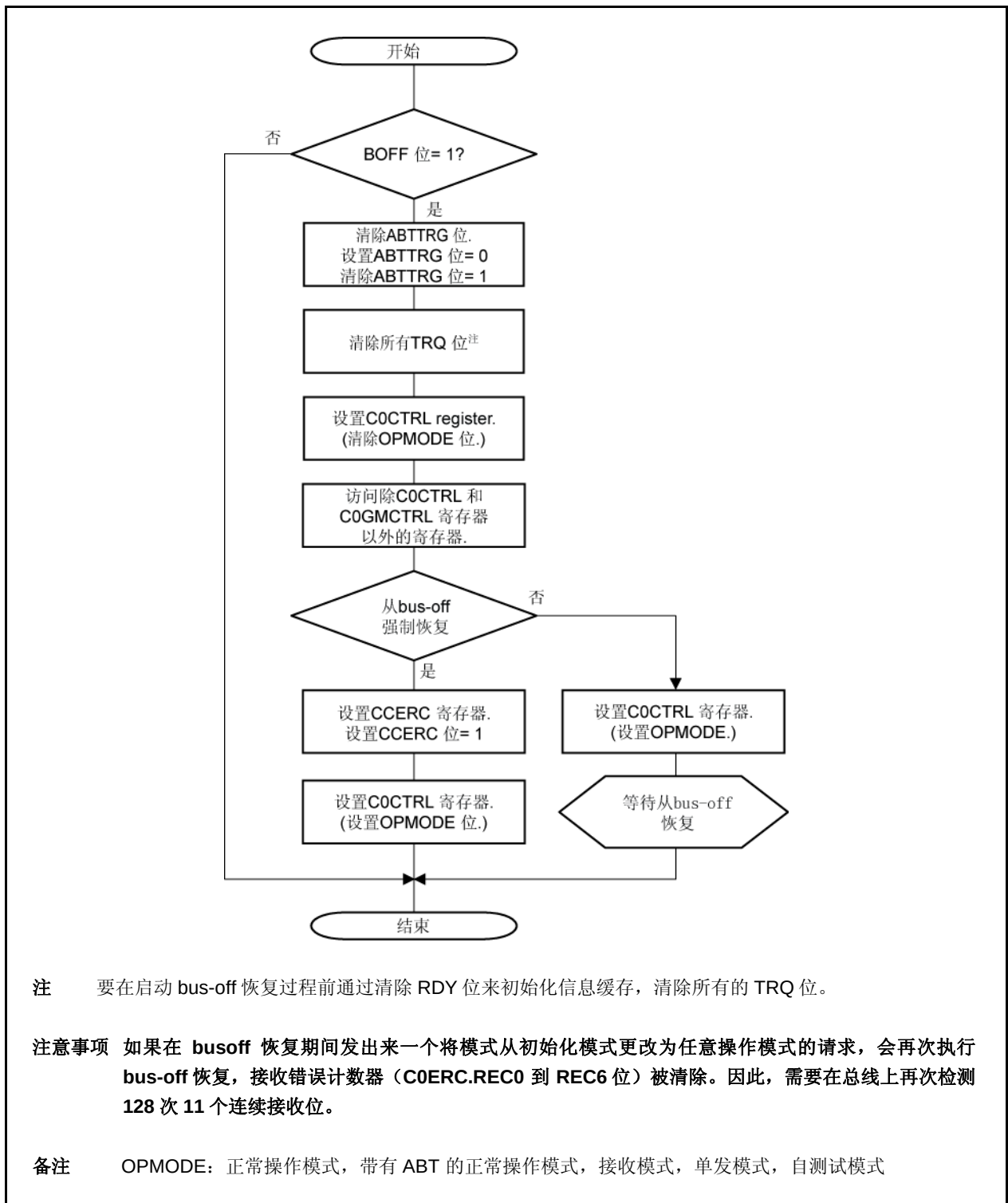
<R> 注 要在启动 bus-off 恢复过程前通过清除 RDY 位来初始化信息缓存，清除所有的 TRQ 位。

<R> 注意事项 如果在 busoff 恢复期间发出来一个将模式从初始化模式更改为任意操作模式的请求，会再次执行 bus-off 恢复，接收错误计数器（C0ERC.REC0 到 REC6 位）被清除。因此，需要在总线上再次检测 128 次的 11 个连续接收位。

备注 OPMODE：正常操作模式，带有 ABT 的正常操作模式，接收模式，单发模式，自测试模式

<R>

图 19-55. bus-off 恢复（带 ABT 的正常操作模式）



<R>

图 19-56. 正常关闭进程

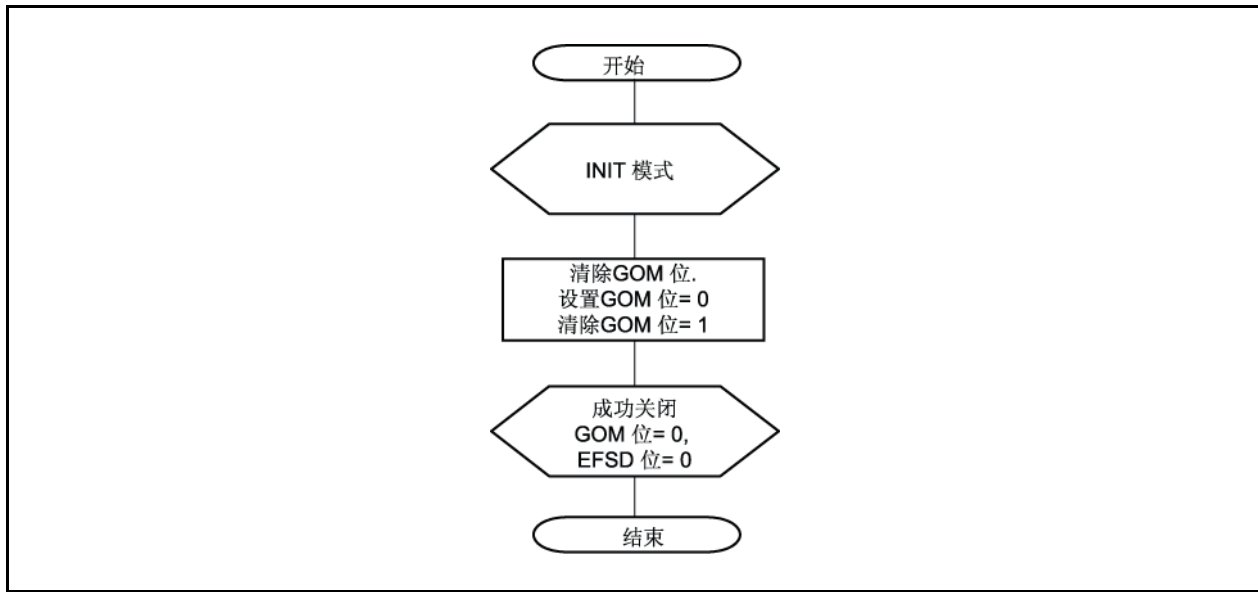


图 19-57. 强制关闭进程

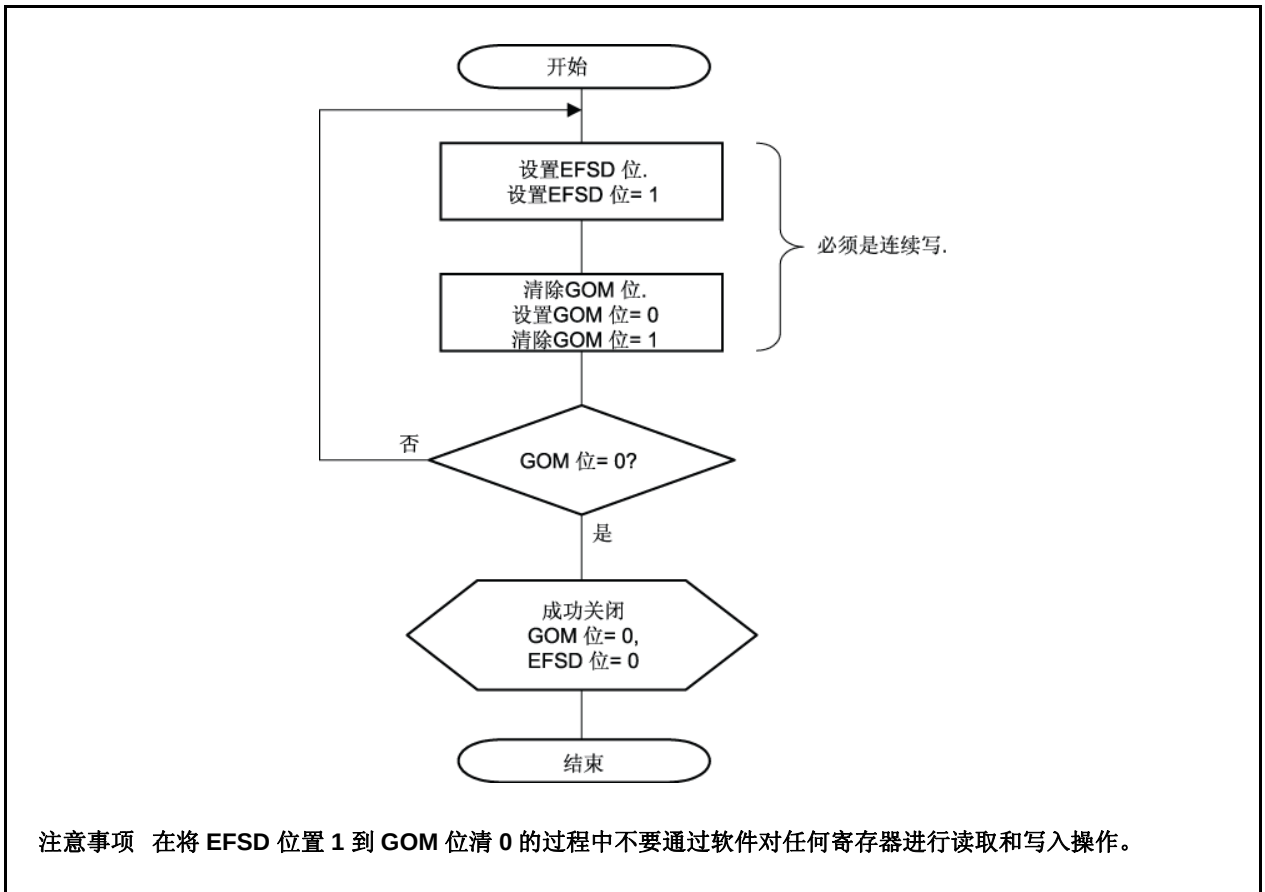
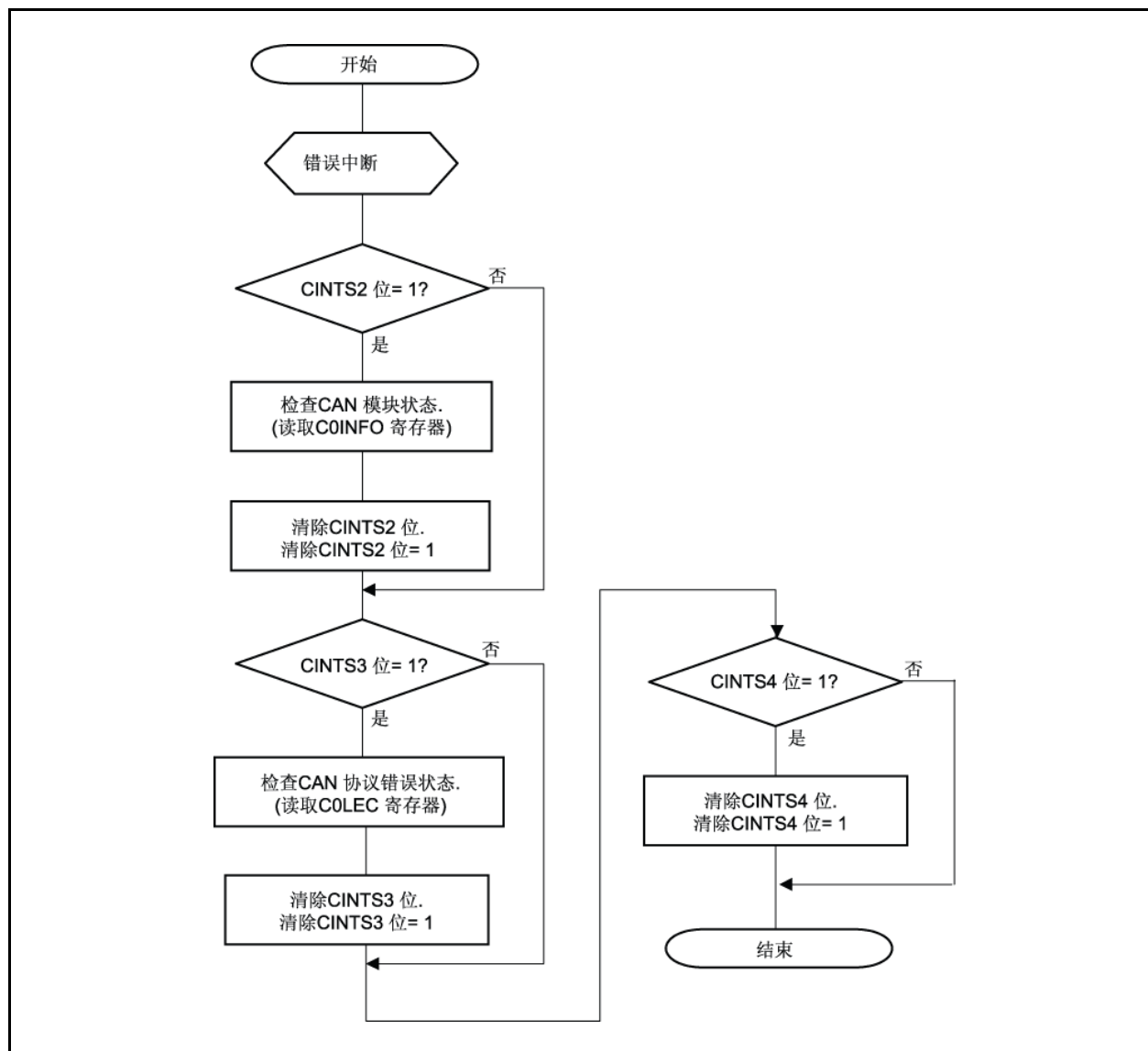
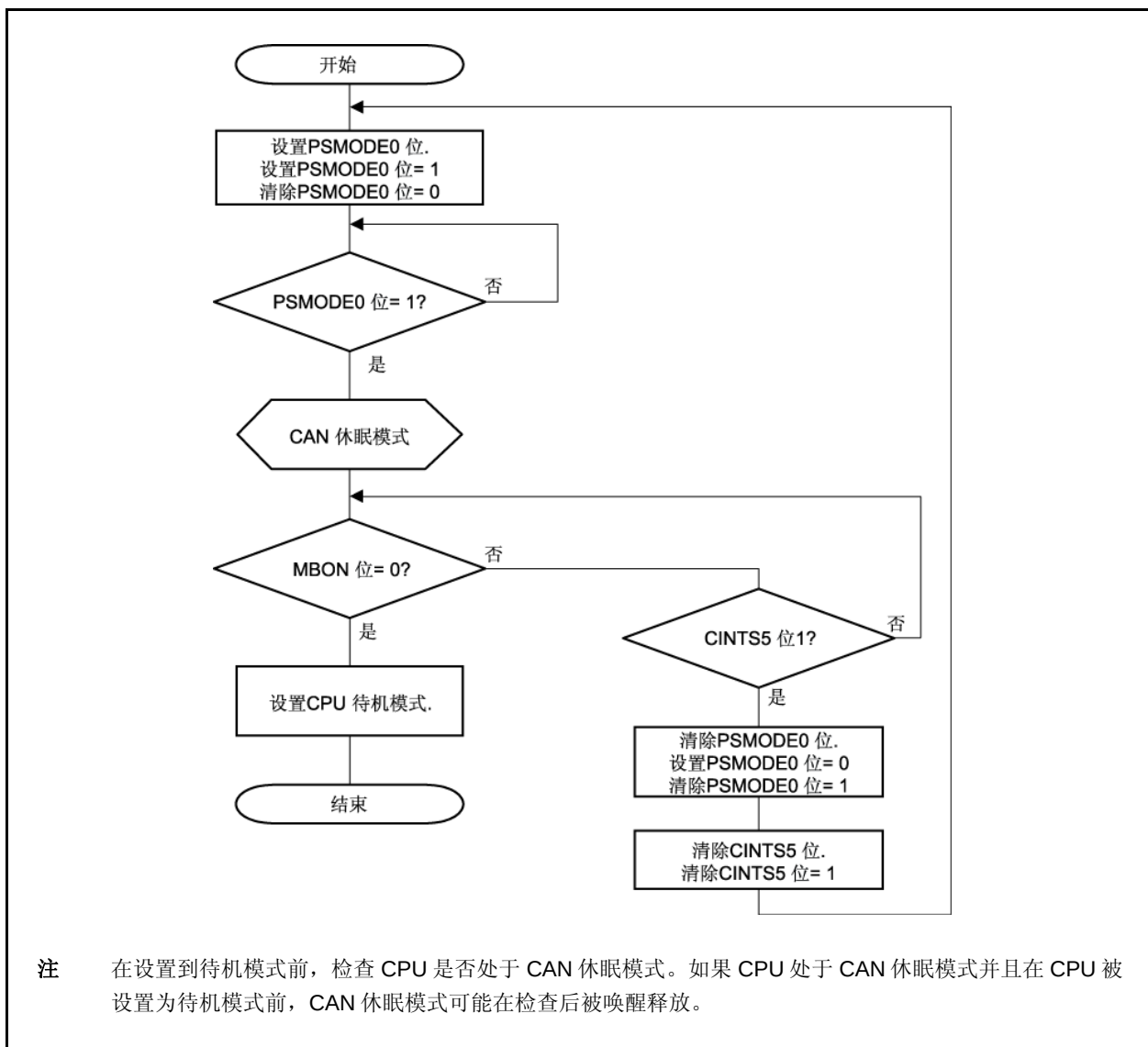


图 19-58. 错误处理



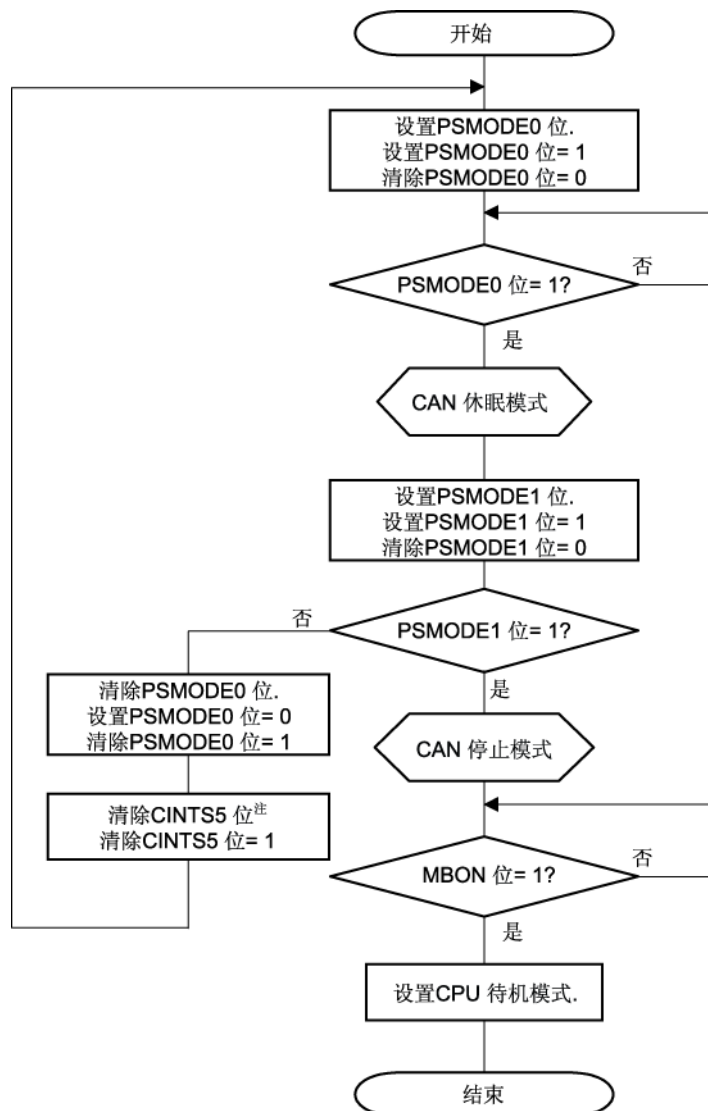
<R>

图 19-59. 设定 CPU 的待机模式（从 CAN 休眠模式）



<R>

图 19-60. 设定 CPU 的待机模式（从 CAN 停止模式）



注 在唤醒中断期间

注意事项 CAN 停止模式仅可以通过向 C0CTRL.PSMODE1 位和 C0CTRL.PSMODE0 位写入 01 的方式释放。
不能通过改变 CAN 总线的方式释放。

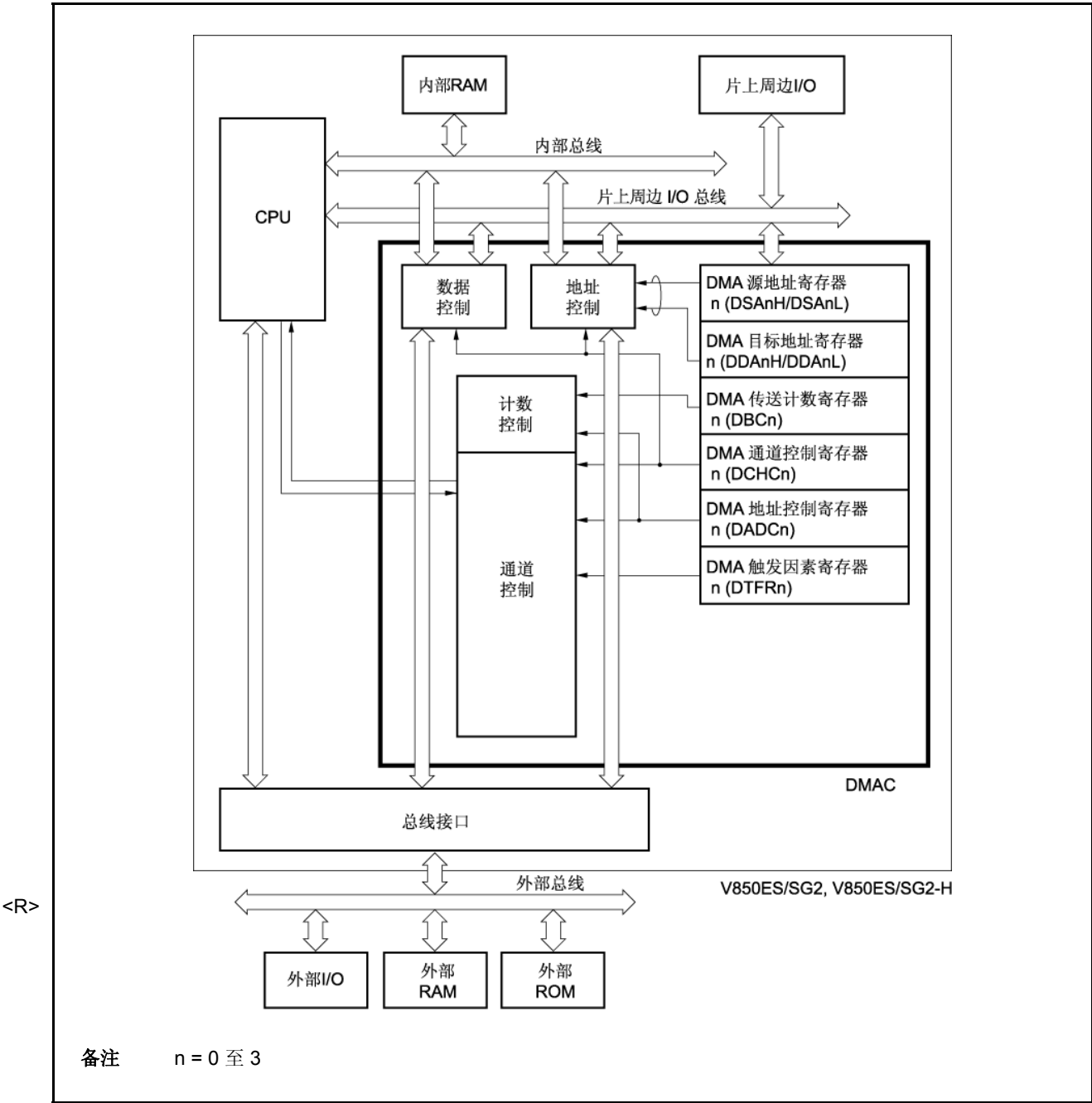
<R> V850ES/SG2 和 V850ES/SG2-H 包含了存储器直接访问（DMA）控制器（DMAC），它用来执行和控制 DMA 传送。

DMAC 控制基于 DMA 请求的存储器和端口、存储器和存储器或者端口和端口之间的数据传送，这种请求通常是由片上周边 I/O（串行接口，定时器/计数器和 A/D 转换器），外部输入引脚的中断，或是软件触发（存储器指的是内部 RAM 或是外部存储器）发起。

20.1 特点

- 4 路独立的 DMA 通道
- 传送单元：8/16 位
- 最大传送计数值：65, 536 (2^{16})
- 传送类型：双周期传送
- 传送模式：单个传送模式
- 传送请求
 - 通过片上周边 I/O（串行接口，定时器 / 计数器和 A/D 转换器）中断或是外部输入引脚的中断而产生的请求。
 - 通过软件触发产生的请求
- 传送目标
 - 内部 RAM ↔ 周边 I/O
 - 周边 I/O ↔ 周边 I/O
 - 内部 RAM ↔ 外部存储器
 - 外部存储器 ↔ 周边 I/O
 - 外部存储器 ↔ 外部存储器

20.2 配置



20.3 寄存器

(1) DMA 源地址寄存器 0 到 3 (DSA0~DSA3)

DSA0~ DSA3 寄存器为 DMA 通道 n (n 的取值范围是 0 到 3) 设定 DMA 的源地址 (每个寄存器 26 位)。

这些寄存器能够被分为 2 个 16 位寄存器, DSA_nH 和 DSA_nL。

这些寄存器可以以 16 位为单位进行读取或写入。

复位后: 未定义 R/W 地址: DSA0H FFFFF082H, DSA1H FFFFF08AH,
DSA2H FFFFF092H, DSA3H FFFFF09AH,
DSA0L FFFFF080H, DSA1L FFFFF088H,
DSA2L FFFFF090H, DSA3L FFFFF098H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSA _n H (n = 0 至 3)	IR	0	0	0	0	0	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSA _n L (n = 0 至 3)	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0

IR	DMA 传送源规范
0	外部存储器或片上周边 I/O
1	内部 RAM

SA25 至 SA16	设置 DMA 传送源的地址 (A25 至 A16) (默认值未定义)。 DMA 传送期间, 下个存放 DMA 传送源地址。 当 DMA 传送完成时, 最先设定的 DMA 地址存放。
-------------	---

SA15 至 SA0	设置 DMA 传送源的地址 (A15 至 A0) (默认值未定义)。 DMA 传送期间, 下个 DMA 传送源地址存放。 当 DMA 传送完成时, 最先设定的 DMA 地址存放。
------------	--

- 注意事项**
- 一定要将 DSA_nH 寄存器的第 14 ~ 10 位清 0。
 - 当 DMA 传送被禁止 (DCHCn.Enn 位为 0) 时, 在下列时序设定 DSA_nH 和 DSA_nL 寄存器。
 - 从复位后到第一个 DMA 传送启动之间的时期
 - 从通过 DCHCn.INITn 位将通道初始化完成后到 DMA 传送启动之间的时期
 - 从完成第一个 DMA 传送 (DCHCn.TCn 位为 1) 到下一个 DMA 传送启动之间的时期
 - 当读取 DSA_n 寄存器中的值时, 两个 16 位寄存器, DSA_nH 和 DSA_nL 的值被读取。如果读取操作与数据更新发生冲突, 可能会读取正在被更新的数据 (见参 20.13 项注意事)。
 - 复位后, 在 DMA 传送启动之前要设定 DSA_nH, DSA_nL, DDA_nH, DDA_nL 和 DBCn 寄存器。否则, 当 DMA 开始发送后的操作将无法得到保证。

(2) DMA 目标地址寄存器 0 到 3 (DDA0 ~ DDA3)

DDA0 到 DDA3 寄存器为 DMA 通道 n (n 的取值范围是 0 到 3) 设定 DMA 的目标地址 (每个寄存器 26 位)。这些寄存器能够被分为两个 16 位寄存器, DDA_nH 和 DDA_nL。
这些寄存器可以以 16 位为单位进行读取或写入。

复位后: 未定义 R/W 地址: DDA0H FFFFF086H, DDA1H FFFFF08EH,
DA2H FFFFF096H, DDA3H FFFFF09EH,
DDA0L FFFFF084H, DDA1L FFFFF08CH,
DDA2L FFFFF094H, DDA3L FFFFF09CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDA _n H (n = 0 至 3)	IR	0	0	0	0	0	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDA _n L (n = 0 至 3)	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

IR	DMA 传送目的规范
0	外部存储器或片上周边 I/O
1	内部 RAM

DA25 至 DA16	设置 DMA 传送目的地址 (A25 至 A16) (默认值未定义)。 DMA 传送期间, 下个 DMA 传送目的地址存放。 当 DMA 传送完成时, 最先设定的 DMA 传送源地址存放。
-------------	---

DA15 至 DA0	设置 DMA 传送目的地址 (A15 至 A0) (默认值未定义)。 DMA 传送期间, 下个 DMA 传送目的地址存放。 当 DMA 传送完成时, 最先设定的 DMA 传送源地址存放。
------------	--

- 注意事项
- 一定要将 DDA_nH 寄存器的第 14 ~ 10 位清 0。
 - 当 DMA 传送被禁止 (DCHC_n.Enn 位为 0) 时, 在下列时序设定 DDA_nH 和 DDA_nL 寄存器。
 - 从复位后到第一个 DMA 传送启动之间的时期
 - 从通过 DCHC_n.INIT_n 位将通道初始化完成后到 DMA 传送启动之间的时期
 - 从完成第一个 DMA 传送 (DCHC_n.TC_n 位为 1) 到下一个 DMA 传送启动之间的时期
 - 当读取 DDA_n 寄存器中的值时, 两个 16 位寄存器, DDA_nH 和 DDA_nL 被读取。如果读取操作与数据更新发生冲突, 可能会读取正在被更新的数据 (见参 20.13 注意事项)。
 - 复位后, 在 DMA 传送启动之前要设定 DSA_nH, DSA_nL, DDA_nH, DDA_nL 和 DBC_n 寄存器。否则, 当 DMA 开始发送后的操作将无法得到保证。

(3) DMA 字节计数器寄存器 0 到 3 (DBC0~DBC3)

DBC0 到 DBC3 寄存器是 16 位寄存器，它用来为 DMA 通道 n (n 的取值范围是 0 到 3) 设定传送的字节数量。在 DMA 发送过程中，这些寄存器保存剩余的发送字节数。这些寄存器在每次传送之后会自动减 1，而不考虑传送的数据单元 (8/16 位)，并且在发生借位时中止发送。这些寄存器可以以 16 位为单位进行读取或写入。

复位后: 未定义	R/W	地址:	DBC0 FFFFF0C0H, DBC1 FFFFF0C2H, DBC2 FFFFF0C4H, DBC3 FFFFF0C6H													
DBCn (n = 0 至 3)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
BC15 至 BC0	字节传送计数设置或在DMA 传送保持字节传送计数															
0000H	字节传送计数 1 或保持字节传送计数															
0001H	字节传送计数 2 或保持字节传送计数															
:	:															
FFFFH	字节传送计数 65,536 (2 ¹⁶) 或保持字节传送计数															
当DMA传送完成时, 最先设定的传送数据的个数被保持。																

- 注意事项 1. 当 DMA 传送被禁止 (DCHCn.Enn 位为 0) 时，在下列时序设定 DDA nH 和 DDA nL 寄存器。
- 从复位后到第一个 DMA 传送启动之间的时期
 - 从通过 DCHCn.INITn 位将通道初始化完成后到 DMA 传送启动之间的时期
 - 从完成第一个 DMA 传送 (DCHCn.TCn 位为 1) 到下一个 DMA 传送启动之间的时期
2. 复位后，在 DMA 传送启动之前要设定 DSA nH, DSA nL, DDA nH, DDA nL 和 DBCn 寄存器。否则，当 DMA 开始发送后的操作将无法得到保证。

(4) DMA 地址控制寄存器 0 到 3 (DADC0 ~ DADC3)

DADC0 到 DADC3 寄存器是 16 位寄存器，它控制着 DMA 通道 n (n 的取值范围是 0 到 3) 的 DMA 传送模式。这些寄存器可以以 16 位为单位进行读取或写入。

复位操作将这些寄存器的值设为 0000H。

复位后: 0000H R/W 地址: DADC0 FFFF0D0H, DADC1 FFFF0D2H,
DADC2 FFFF0D4H, DADC3 FFFF0D6H

DADCn (n = 0 至 3)	15	14	13	12	11	10	9	8
	0	DS0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	SAD1	SAD0	DAD1	DAD0	0	0	0	0

DS0	传送数据尺寸的设置
0	8 位
1	16 位

SAD1	SAD0	传送源地址的计数方向的设置
0	0	增加
0	1	减少
1	0	固定
1	1	禁止设置

DAD1	DAD0	目的地址的计数方向的设置
0	0	增加
0	1	减少
1	0	固定
1	1	禁止设置

- 注意事项**
- 一定要将 DADCn 寄存器的第 15, 13 到 8, 和 3 位清 0。
 - 当 DMA 传送被禁止 (DCHCn.Enn 位为 0) 时, 在下列时序设定 DDA nH 和 DDA nL 寄存器。
 - 从复位后到第一个 DMA 传送启动之间的时期
 - 从通过 DCHCn.INITn 位将通道初始化后到 DMA 传送启动之间的时期
 - 从完成第一个 DMA 传送 (DCHCn.TCn 位为 1) 到下一个 DMA 传送启动之间的时期
 - DS0 位指定传送数据的大小, 但并不能控制总线宽度。因此如果设定了 8 位数据 (DS0 为 0), 那么较低位的数据总线并不经常使用。
 - 如果传送数据设定为 16 位 (DS0 位为 1), 传送不能从奇数地址开始。传送只能从与 0 对齐的低地址 (偶地址) 的首位开始。
 - 如果在片上周边 I/O 寄存器上 (无论作为发送源还是目标) 执行 DMA 操作, 一定要按照寄存器的大小来设定传送数据的宽度。例如, 在一个 8 位寄存器上执行 DMA 传送, 一定要指定 8 位传送。

(5) DMA 通道控制寄存器 0 到 3 (DCHC0 ~ DCHC3)

DCHC0~DCHC3 寄存器是 8 位寄存器，它控制着 DMA 通道 n 的 DMA 发送操作模式。

这些寄存器可以以 8 位为单位或以 1 位为单位进行读取操作或写操作。（但是，第 7 位是只读位、第 1 位和第 2 位是只写位。如果读取第 1 位和第 2 位，结果永远为 0。）

复位输入将把这些寄存器置为 00H。

复位后: 00H R/W 地址: DCHC0 FFFF0E0H, DCHC1 FFFF0E2H,
DCHC2 FFFF0E4H, DCHC3 FFFF0E6H

	<7>	6	5	4	3	<2>	<1>	<0>
DCHCn	TCn ^{注1}	0	0	0	0	INITn ^{注2}	STGn ^{注2}	Enn

(n = 0 至 3)

TCn ^{注1}	状态标志显示了通过DMA通道n的DMA 传送完成与否
0	DMA传送没有完成。
1	DMA传送已经完成。
它在最后一次 DMA传送时被设为 1并在被读取时被清除为0。	

INITn ^{注2}	如果DMA传送禁止(Enn 位= 0)时INITn位被设为1，那么DMA传送状态将被初始化。 在DMA传送完成前（在TCn位被设为1前）重设DMA传送状态（重设 DDAnH, DDAnL, DSAAnH, DSAAnL, DBCn, 以及DADCn 寄存器)时，一定要初始化DMA 通道、 然而，初始化DMA控制器时，一定要按照 20.13 注意事项中所描述的步骤来进行。
---------------------	--

STGn ^{注2}	这是DMA传送的软件开始触发。 如果该位在DMA传送允许状态 (TCn 位= 0, Enn位= 1)中被设为1，那么DMA传送开始。
--------------------	---

Enn	允许或禁止通过DMA通道n的DMA 传送的设置
0	禁止DMA传送
1	允许DMA传送
Enn位被设为1时允许DMA传送。 DMA传送完成时 (终止计数生成时)，该位被自动清除为0。 要中止DMA 传送，则通过软件将Enn 位清除为0。要继续传送，则再次将Enn 位设为1。 然而，在中止或继续DMA传送时，一定要按照 20.13 注意事项中所描述的步骤来进行。	

- 注 1. TCn 位为只读位。
 2. INITn 和 STGn 位为只写位。

- 注意事项 1. 一定要将 DCHCn 寄存器的第 6~3 位清 0。
2. 当 DMA 传送完成时（产生中止计数时），Enn 位清 0，然后 TCn 位置 1。如果在 DCHCn 寄存器的各位更新时读取 DCHCn 寄存器，读取获得的值表明“传送未完成和传送被禁止（TCn 位为 0 和 Enn 为位 0）”。

(6) DMA 触发寄存器 0 到 3 (DTFR0 ~ DTFR3)

DTFR0~DTFR3 寄存器是 8 位寄存器，它通过片上周边 I/O 的中断请求控制着 DMA 传送的启动触发。
这些寄存器设置的中断请求信号被当作 DMA 传送启动源因。
这些寄存器可以以 8 位为单位进行读取或写入。但是，DFn 位只能通过 1 位的方式读取或写入。
复位输入将把这些寄存器置为 00H。

(1/2)

复位后: 00H R/W 地址: DTFR0 FFFFF810H, DTFR1 FFFFF812H,
DTFR2 FFFFF814H, DTFR3 FFFFF816H

DTFRn	<7>	6	5	4	3	2	1	0
	DFn	0	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0

(n = 0 至 3)

DFn ^注	DMA传送请求状态标志
0	没有DMA传送请求
1	DMA传送请求

<R> 注 不要通过软件将 DFn 位设为“1”。因为指定中断，导致在禁止 DMA 传送期间启动 DMA 传送，，则应对该位写入 0 以清除 DMA 传送请求。

注意事项 当禁止 DMA 传送时 (DCHCn.Enn 位的值为 0)，在以下时序设定 IFCn5~IFCn0 位。

- 从复位后到第一个 DMA 传送启动之间的时期
- 从通过 DCHCn.INITn 位将通道初始化完成后到 DMA 传送启动之间的时期
- 从完成第一个 DMA 传送 (DCHCn.TCn 位为 1) 到下一个 DMA 传送启动之间的时期

备注 IFCn5~IFCn0 位的详情，请参见表 20-1 DMA 启动源因。

<R>

注意事项 2. 当更改 DTFRn 寄存器设置时, 必需按以下步骤进行。

[V850ES/SG2]

- 要设置 IFCn5~IFCn0 的值没有被设置到另一通道的 IFCm5~IFCm0 时 (n = 0 到 3, m = 0 到 3, n ≠ m)
 - <1> 停止将要进行改写的通道的 DMA_n 操作 (DCHCn.Enn 位 = 0)。
 - <2> 更改 DTFRn 寄存器的设置。(请确保将 DFn 位设为 0 且更改时需要按照字节操作设置。)
 - <3> 确定 DFn 位为 0 (预先停止中断发生源操作。)
 - <4> 允许 DMA_n 操作 (Enn 位 = 1)。
- 要设置到 IFCn5 ~ IFCn0 的值没有被设置到另一通道的 IFCm5~IFCm0 位时 (n = 0 到 3, m = 0 到 3, n ≠ m)
 - <1> 停止将要进行改写操作的通道的 DMA_n 操作 (DCHCn.Enn 位 = 0)。
 - <2> 停止通道的 DMA_m 操作, 要求该通道中 IFCm5~IFCm0 的值和用来重写 IFCn5 ~ IFCn0 的值相同 (DCHCm.Emm bit = 0)。
 - <3> 更改 DTFRn 寄存器的设置。(请确保将 DFn 位设为 0 且更改时需要按照字节操作设置。)
 - <4> 确定 DFn 位和 DFm 位为 0。(预先停止中断发生源操作。)
 - <5> 允许 DMA_n 操作 (位 Enn 和 Emm = 1)。

[V850ES/SG2-H]

- 要改变 DTFRn 寄存器的设置, 则务必停止通道中的 DMA_m, 停止的通道优先级低于要进行改写的通道中的 DMA_n。
 - <1> 停止将要进行改写操作的通道的 DMA_n 操作 (DCHCn.Enn 位 = 0)。
 - <2> 停止通道的 DMA_m, 停止的通道优先级低于要进行改写的通道的 DMA_n。
 - <3> 更改 DTFRn 寄存器的设置。(请确保将 DFn 位设为 0 且更改时需要按照字节操作设置。)
 - <4> 确定 DFn 位和 DFm 位为 0。(预先停止中断发生源操作。)
 - <5> 允许 DMA_n 操作 (位 Enn 和 Emm = 1)。
- 3. 在待机模式 (IDEL1, IDLE2, STOP, 或 sub- IDLE 模式) 中产生的中断请求不会启动 DMA 传送周期 (DFn 位也不会被设为 1)。
- 4. 如果通过 IFCn5~IFCn0 位选择了 DMA 启动源因, 那么不管 DMA 传送是使能还是禁止, 当被选中的片上周边 I/O 中发生中断时, DFn 位都将设为 1。如果在这种状态中 DMA 为使能, 那么 DMA 传送会立即启动。

备注 IFCn5~IFCn0 位的详情, 请参见表 20-1 DMA 启动源因。

表格 20-1. DMA 启动源因 (1/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	中断源
0	0	0	0	0	0	禁止通过中断产生的 DMA 请求
0	0	0	0	0	1	INTP0
0	0	0	0	1	0	INTP1
0	0	0	0	1	1	INTP2
0	0	0	1	0	0	INTP3
0	0	0	1	0	1	INTP4
0	0	0	1	1	0	INTP5
0	0	0	1	1	1	INTP6
0	0	1	0	0	0	INTP7
0	0	1	0	0	1	INTTQ0OV
0	0	1	0	1	0	INTTQ0CC0
0	0	1	0	1	1	INTTQ0CC1
0	0	1	1	0	0	INTTQ0CC2
0	0	1	1	0	1	INTTQ0CC3
0	0	1	1	1	0	INTTP0OV
0	0	1	1	1	1	INTTP0CC0
0	1	0	0	0	0	INTTP0CC1
0	1	0	0	0	1	INTTP1OV
0	1	0	0	1	0	INTTP1CC0
0	1	0	0	1	1	INTTP1CC1
0	1	0	1	0	0	INTTP2OV
0	1	0	1	0	1	INTTP2CC0
0	1	0	1	1	0	INTTP2CC1
0	1	0	1	1	1	INTTP3CC0
0	1	1	0	0	0	INTTP3CC1
0	1	1	0	0	1	INTTP4CC0
0	1	1	0	1	0	INTTP4CC1
0	1	1	0	1	1	INTTP5CC0
0	1	1	1	0	0	INTTP5CC1
0	1	1	1	0	1	INTTM0EQ0
0	1	1	1	1	0	INTCB0R/INTIIC1 [‡]
0	1	1	1	1	1	INTCB0T
1	0	0	0	0	0	INTCB1R
1	0	0	0	0	1	INTCB1T
1	0	0	0	1	0	INTCB2R
1	0	0	0	1	1	INTCB2T
1	0	0	1	0	0	INTCB3R
1	0	0	1	0	1	INTCB3T
1	0	0	1	1	0	INTUA0R/INTCB4R
1	0	0	1	1	1	INTUA0T/INTCB4T
1	0	1	0	0	0	INTUA1R/INTIIC2 [‡]
1	0	1	0	0	1	INTUA1T
1	0	1	0	1	0	INTUA2R/INTIIC0 [‡]

注 仅限内置 I²C 总线版本 (Y 产品)

备注 n = 0 至 3

表格 20-1. DMA 启动源因（2/2）

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	中断源
1	0	1	0	1	1	INTUA2T
1	0	1	1	0	0	INTAD
1	0	1	1	0	1	INTKR
1	0	1	1	1	0	INTERR [※]
1	0	1	1	1	1	INTSTA [※]
1	1	0	0	0	0	INTIE1 [※]
除上述情况						禁止设置

注 仅限内置 IEBus 控制器版本。

备注 n = 0 至 3

20.4 传送目标

表 20-2 显示了传送目标之间的关系（√：允许传送，×：禁止传送）。

表 20-2. 传送目标之间的关系

		传送目标			
		内置 ROM	片上周边 I/O	内置 RAM	外部存储器
源	片上周边 I/O	×	√	√	√
	内置 RAM	×	√	×	√
	外部存储器	×	√	√	√
	内置 ROM	×	×	×	×

注意事项 在表 20-2 中标记为“×”的传送目标和源之间的合并操作不能被保证。

20.5 传送模式

传送模式即是单个传送。

在单个发送模式中，总线在每一次字节 / 半字传送后被释放。如果存在后续的 DMA 传送请求，传送立即被再次执行一次。这种操作直到发生计数完成才会停止。

当 DMAC 释放总线后，如果有另一个具有更高优先权的 DMA 发送请求发出，那么更高优先权的请求总是优先。

如果在一个传送周期内，正在传输的通道发出一个新的传送请求，同时另外一个通道产生一个较低优先级的传送请求，则在 CPU 释放总线之后，执行具有较低优先级通道的 DMA 传送（在传送周期中同一通道新的传送请求被忽略）。

20.6 传送类型

传送类型支持双周期传送。

在双周期传送中，数据传送在两种周期中执行，一个读取周期和一个写入周期。

在读取周期中，传送源地址被输出，执行从源地址到 DMAC 的读取操作。在写入周期中，传送的目标地址被输出，从 DMAC 到目标地址执行写入操作。

时钟的空闲周期永远插入在读取周期和写入周期之间。如果两个周期的 DMA 传送中的传送源和目标之间的数据总线宽度不一致，操作将按照如下步骤执行。

<16 位数据传送>

<1> 从 32 位总线传送 → 16 位总线

产生一个读取周期（高 16 位处于高阻抗状态）后跟着产生一个写入周期（16 位）。

<2> 从 16-/32-位总线到 8 位总线的传送

产生 16 位读取周期一次，然后产生两次 8 位写入周期。

<3> 从 8 位总线到 16-/32-位总线的传送

8 位读取周期产生两次，随后产生一次 16 位写入周期。

<4> 从 16-位总线到 32 位总线之间的传送

16 位读取周期产生一次，随后产生一次 16 位写入周期。

DMA 传送和片上周边 I/O 寄存器（传送源 / 目标）相关时，一定要指定与寄存器大小相同的传送宽度。例如，对 8 位寄存器执行 DMA 传送，一定要指定字节（8 位）传送。

备注 每种传送目标的总线宽度（传送源 / 目标）如下。

- 片上周边 I/O: 16 位总线宽度
- 内部 RAM: 32 位总线宽度
- 外部存储器: 8 位或者 16 位总线宽度

20.7 DMA 通道优先级

DMA 通道优先级固定如下。

DMA 通道 0 > DMA 通道 1 > DMA 通道 2 > DMA 通道 3

在每一个传送周期检测优先级。

20.8 DMA 传送的相关时间

响应 DMA 请求所需要的时间，和 DMA 传送所需的最小时钟周期如下所示。

单个发送：DMA 响应时间 (<1>) + 传送源存储器访问 (<2>) + 1^{注 1} + 传送目标存储器访问 (<2>)

DMA 周期		最小执行周期
<1> DMA 请求响应时间		4 个时钟 (最小) + 噪声消除时间 ^{注 2}
<2> 存储器访问	外部存储器访问	取决于连接的存储器。
	内部 RAM 访问	2 个时钟 ^{注 3}
	外围 I/O 寄存器访问	3 个时钟 + VSWC 寄存器指定的等待周期数 ^{注 4}

- 注
- 1. 在 DMA 传送的读取周期和写入周期之间总是会插入一个时钟。
 - 2. 如果指定一个外部中断 (INTPn) 作为启动 DMA 传送的触发信号，噪声消除时间将增加 (n 值的范围是 0~7)。
 - 3. DMA 周期至少需要两个时钟。
 - 4. 访问特定的外围 I/O 寄存器需要更多的等待周期 (详见 3.4.9 (2))。

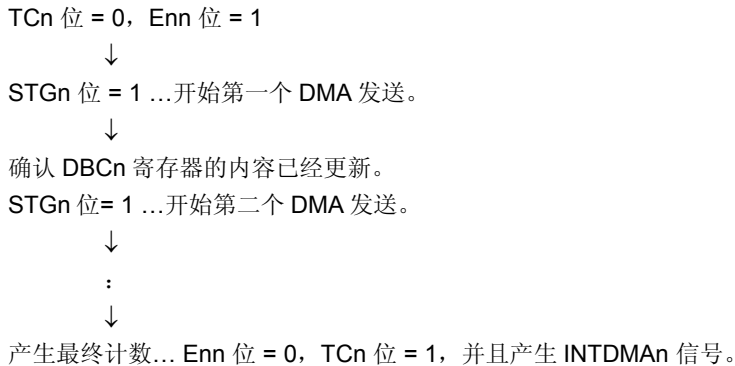
20.9 DMA 传送启动源因

有两种类型的 DMA 传送启动源因，如下所示。

(1) 通过软件发出请求

如果当 DCHCn.TCn 位 = 1 和 Enn 位 = 0 (允许 DMA 传送) 时把 STGn 位置 1，则 DMA 传送启动。

为了随后立即请求下一个 DMA 传送周期，通过 DBCn 寄存器确认先前的 DMA 传送周期已经完成，然后再次将 STGn 位置 1 (n 值围的取范是 0~3)。



(2) 通过片上周边 I/O 发出请求

如果当 DCHCn.TCn 位 = 0 和 Enn 位 = 1 (允许 DMA 传送) 时，由 DTFRn 寄存器设定的片上周边 I/O 产生中断请求，则 DMA 发送开始。

- 注意事项**
1. 不能为一个 DMA 通道使用两种启动源因 (软件触发和硬件触发)。如果为一个 DMA 通道同时生成两种启动源因，那么两种源因中有一种会是无效的。有效的启动源因不能被识别。
 2. 在先前的 DMA 发送请求产生后，或是先前的 DMA 传送周期中产生的新的传送请求将被忽略 (清除)。
 3. 同一个 DMA 通道的传送请求间隔变化取决于在 DMA 发送周期中设定的总线等待，其它通道的启动状态或外部总线的保持请求也对间隔变化有影响。尤其如注意事项 2 中的描述，在 DMA 传送周期之前或是 DMA 传送周期中产生的同一通道新的发送请求将被忽略。因此，相同 DMA 通道的传送请求间隔必须通过系统彻底分离。当使用软件触发时，先前产生的 DMA 传送周期是否完成可以通过 DBCn 寄存器的更新来检测。

20.10 DMA 中止原因

如果发生总线保持，DMA 传送将被中止。

如果在内部存储器 / 片上周边 I/O 与内部存储器 / 片上周边 I/O 之间进行传送时，发生总线保持，则 DMA 传送同样被中止。

当清除总线保持后，DMA 传送恢复。

20.11 DMA 传送结束

当 DMA 传送已经完成了在 DBCn 寄存器中设定的次数，且 DCHCn.Enn 位清 0，TCn 位置 1 时，将会产生一个 DMA 传送结束中断请求信号 (INTDMA_n) (n 的取值范围是 0~3) 到达中断控制器 (INTC)。

<R> V850ES/SG2 和 V850ES/SG2-H 不会将终止计数信号输出到外部设备中。因此，通过 DMA 传送结束中断或是 TCn 位的轮询检测来确认 DMA 传送完成。

20.12 操作时序

图 20-1 到 20-4 显示了 DMA 操作时序。

图 20-1. DMA 的优先级 (1)

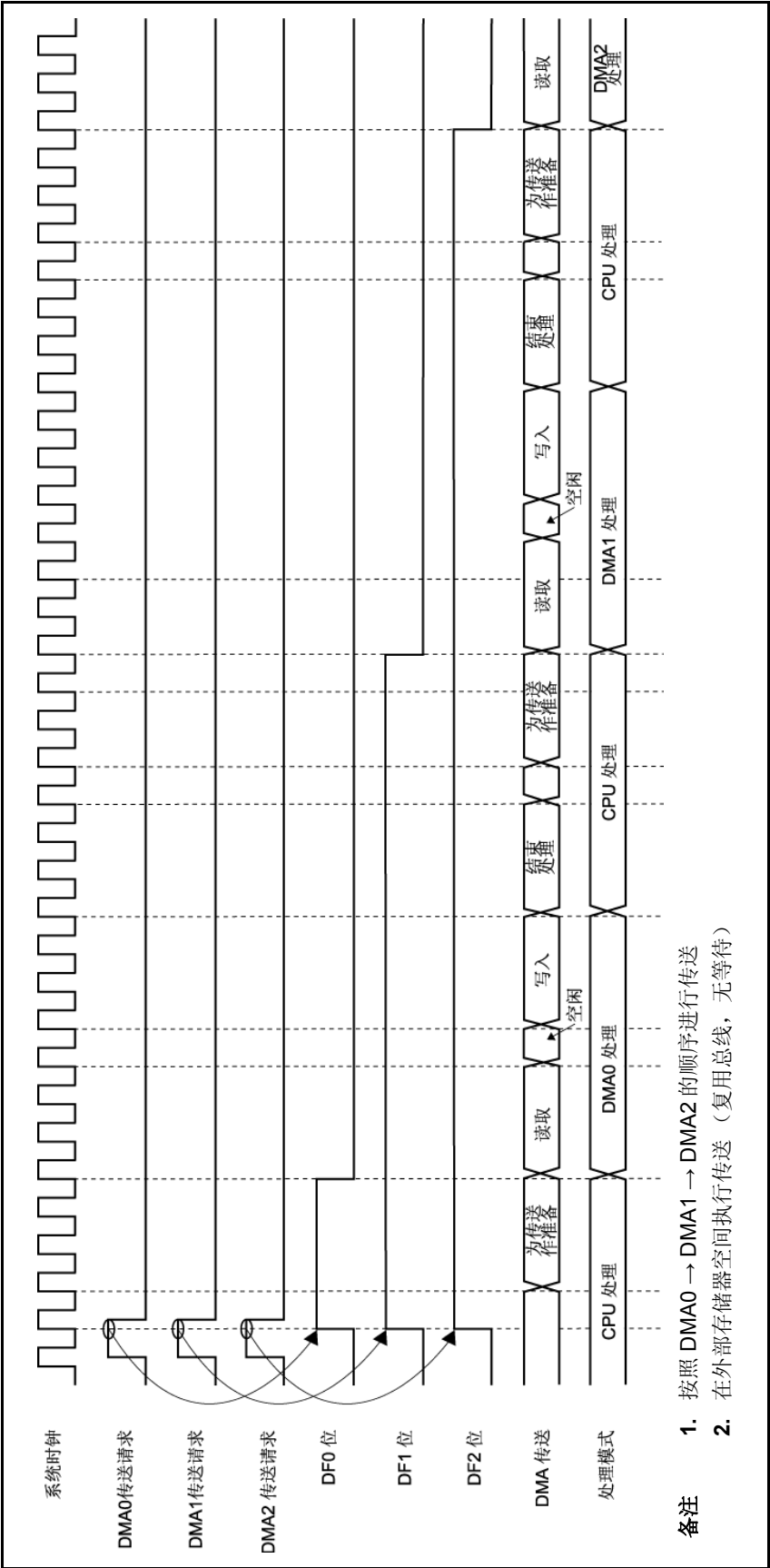


图 20-2. DMA 的优先级 (2)

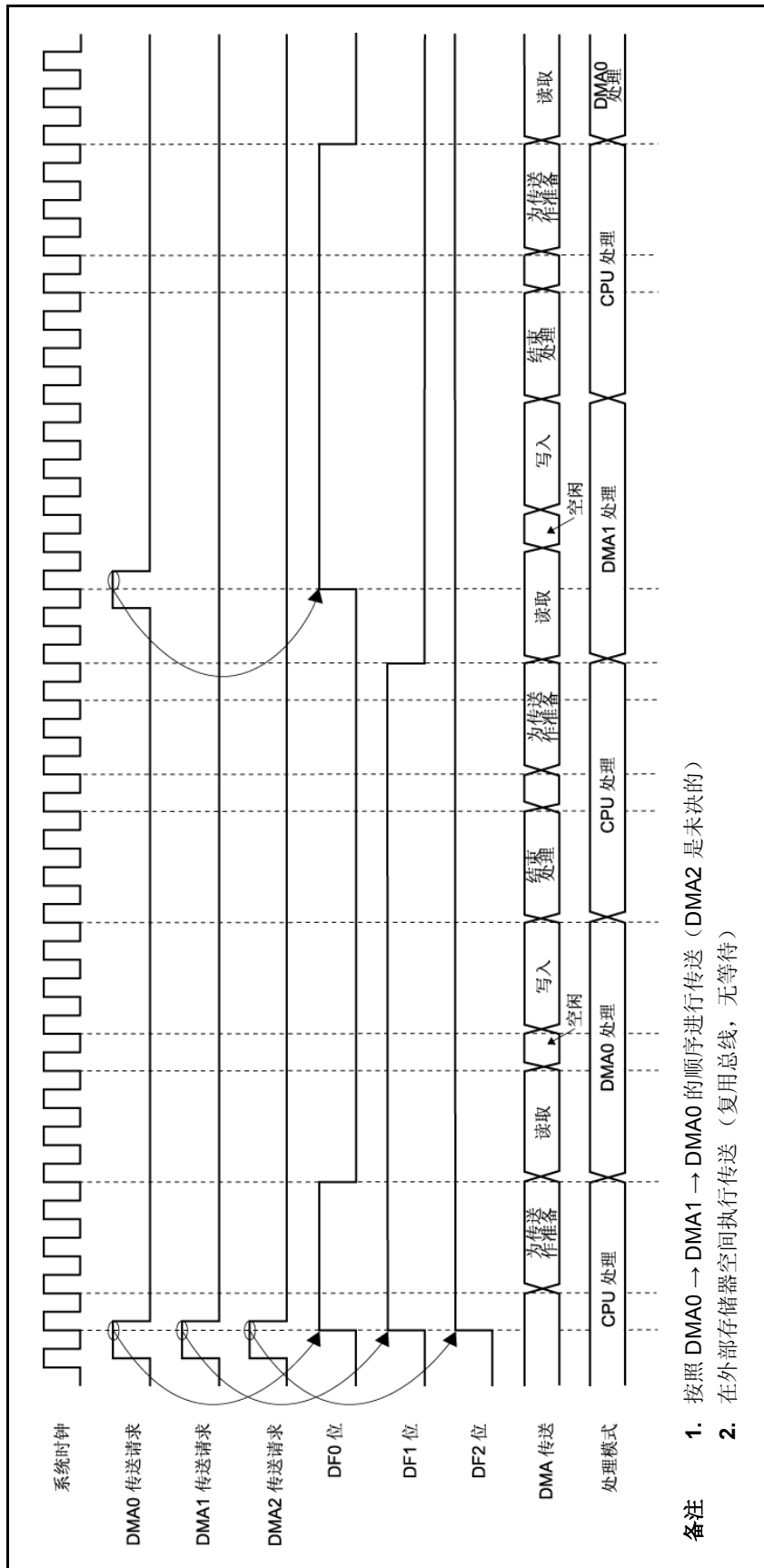
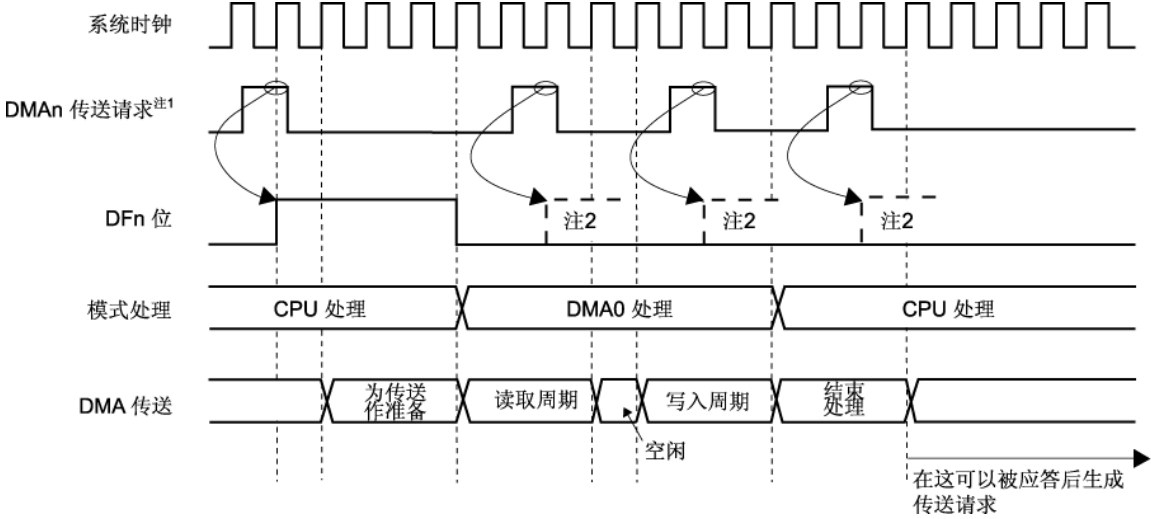


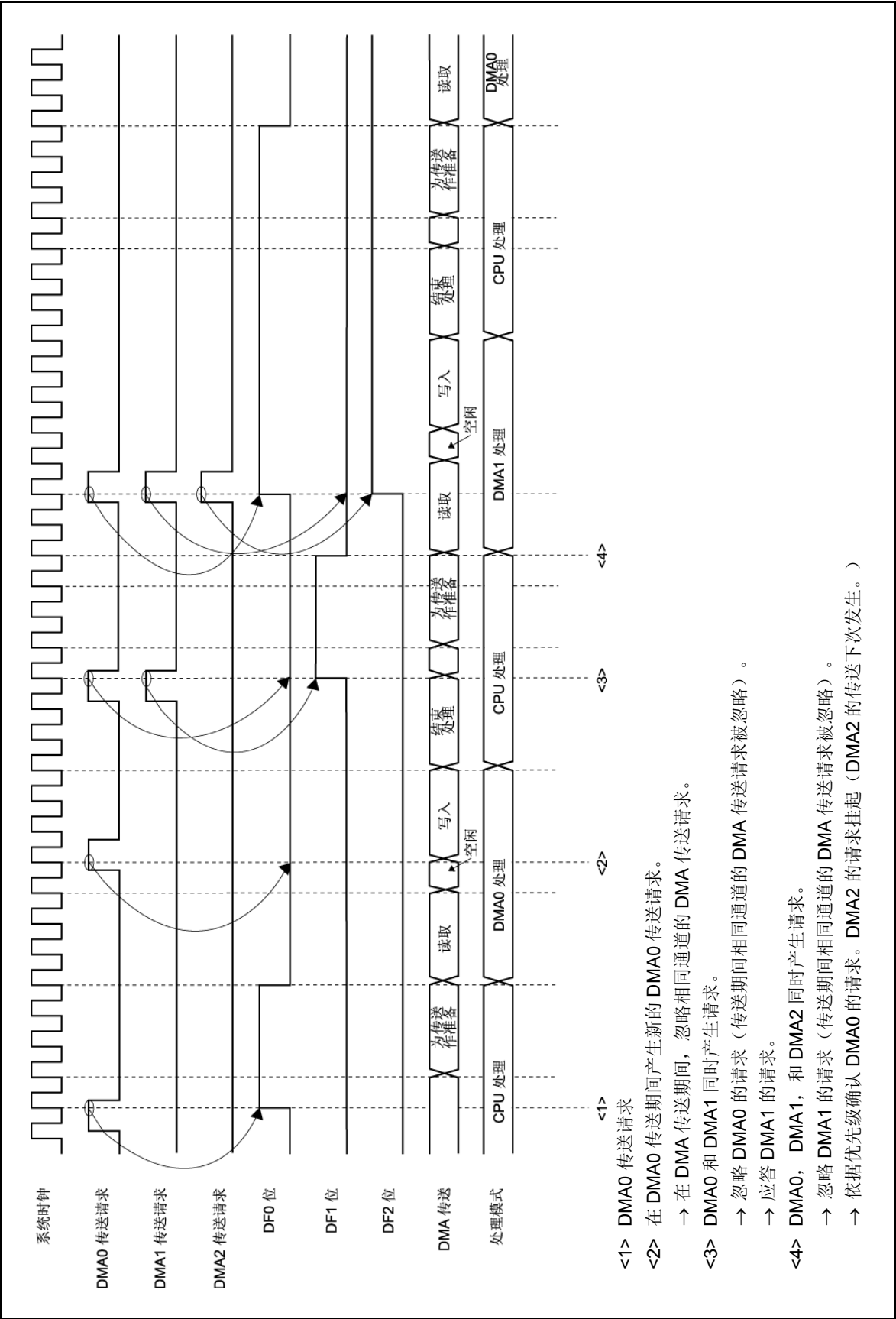
图 20-3. DMA 传送请求被忽略的时期（1）



- 注 1. 由片上周边 I/O 或是软件触发（STG_n 位）产生的中断。
2. 当第一个请求已经发出和处理完成之间，在同一通道产生的新的 DMA 请求将被忽略。

备注 在外部存储器空间执行传送（复用总线，无等待）

图 20-4. DMA 传送请求被忽略的时期 (2)



20.13 注意事项

(1) VSWC 寄存器的注意事项

当使用 DMAC，一定要按照操作频率为 VSWC 寄存器设定适当的值。

当使用 VSWC 寄存器的缺省值 (77H)，或者为 VSWC 寄存器设定的值不适合时，不会执行正确的操作。

((VSWC) 寄存器的详细信息，参见 3.4.9 (1) (a) 系统等待控制寄存器 (VSWC))

(2) 在内部 RAM 中执行 DMA 传送的注意事项

在内部 RAM 中执行下列指令时，不能在内部 RAM 中 (无论是作为源或者目标) 执行 DMA 传送，因为 CPU 随后可能运行不正确。

- 内部 RAM 中的位操作指令 (SET1, CLR1, 或是 NOT1)
- 内部 RAM 中对于未正确对齐地址的数据访问指令

相反的，当在内部 RAM 中 (无论是源或者目标) 执行 DMA 传送时，不要执行上述两种指令。

(3) 读取 DCHCn.TCn 位的注意事项 (n=0 到 3)

当对 TCn 位进行读取操作时它将被清 0，但是如果使用特殊时序读取，则它不会自动清 0。为了正确的将 TCn 位清 0，要按照下列过程处理。

(a) 通过轮询检测 TCn 位等待 DMA 传送完成

确认 TCn 位已经被置 1 (在 TCn 位 = 1 被读取后)，然后另外再读取 TCn 位三次。

(b) 在中断服务程序中读取 TCn 位

读取 TCn 位的操作执行 3 次。

(4) DMA 传送初始化过程 (将 DCHCn.INITn 位置 1)

当要对正在进行 DMA 传送的通道执行初始化时, 即使将 INITn 位置 1, 该通道也不会被初始化。为了正确的对通道进行初始化, 需要在下列两个过程选中一个执行。

(a) 临时中止所有通道的 DMA 传送

要对正在进行 DMA 传送的通道执行初始化, 需要按以下步骤<1>到<7> 执行。

然而, 需要注意的是当执行第<5>步时 TCn 位被清 0。确定其它处理程序不要预期 TCn 位为 1。

<1> 禁止中断 (DI)。

<2> 读取没有被强行中止的其它任何 DMA 通道的 DCHCn.Enn 位, 然后将这个值传送到通用寄存器。

<3> 将正在被使用的 DMA 通道的 Enn 位 (包括被强行中止的通道) 清 0。为了将最后一个 DMA 通道的 Enn 位清 0, 需要执行两次清除指令。如果 DMA 传送目标 (传送源 / 目标) 是内部 RAM, 该指令需要执行三次。

例: 如果使用通道 0, 1 和 2 (如果传送的目标不是内部 RAM), 按照以下顺序执行指令。

- 将 DCHC0.E00 位清 0。
- 将 DCHC1.E11 位清 0。
- 将 DCHC2.E22 位清 0。
- 再次将 DCHC2.E22 位清 0。

<4> 被强行中止的通道的 INITn 位置 1。

<5> 读取没有被强行中止的各个通道的 TCn 位。如果在第<2>步中读取的 TCn 位和 Enn 位的值都为 1 (逻辑与的结果为 1), 将已经保存的 Enn 位清 0。

<6> 在第<5>步操作之后, 将 Enn 位的值写入 DCHCn 寄存器。

<7> 允许中断 (EI)。

注意事项 一定要执行上述的第<5>步, 用于防止对在第<2>和<3>步之间正常完成的 DMA 传送通道的 Enn 位的非法设置。

(b) 重复设定 INITn 位直到传送被正确的强行中止

- <1> 抑制要被强行中止的通道 DMA 请求源发出的请求 (停止片上周边 I/O 的操作)。
- <2> 通过 DTFRn.DFn 位来检测要被强行中止的通道的 DMA 传送请求未处于挂起状态。如果一个 DMA 传送请求处于挂起状态, 需要等待直到该挂起请求执行完成。
- <3> 如果确定要被强行中止的通道的 DMA 请求未处于挂起状态, 将 Enn 位清 0。
- <4> 再次将要被强行中止的通道的 Enn 位清 0。
如果要被强行中止的通道 (传送源 / 目标) 其传送目标是内部 RAM, 再次执行该操作。
- <5> 拷贝要被强行中止的通道的传送初始数量到通用寄存器。
- <6> 将要被强行中止的通道的 INITn 位置 1。
- <7> 读取要被强行中止的通道的 DBCn 寄存器的值, 并与第<5>步中的拷贝值相比较。如果两者不匹配, 重复第<6>步和第<7>步的操作。

- 备注**
- 1. 当在第<7>步中读取了 DBCn 寄存器的值后, 如果强行中止已经正确完成, 那么传送初始数量被读取。否则, 读取到的是传送剩余数量。
 - 2. 如果有某个被强行中止通道之外的 DMA 通道应用程序经常使用 DMA 传送, 那么模式 (b) 可能会需要很长的时间。

(5) 暂停 DMA 传送的步骤 (将 Enn 位清 0)

停止和恢复执行中的 DMA 传送需要以下步骤。

- <1> 抑制 DMA 请求源的传送请求 (停止片上周边 I/O 的操作)。
- <2> 使用 DFn 位来检测 DMA 传送请求未处于挂起状态 (检测 DFn 位是否为 0)。
如果请求挂起, 则等待直到被挂起的请求执行完毕。
- <3> 如果确认没有 DMA 传送请求被挂起, 将 Enn 位清 0 (该操作停止 DMA 传送)。
- <4> 将 Enn 位置 1 以恢复 DMA 发送。
- <5> 恢复被停止的 DMA 请求源的操作 (开始片上周边 I/O 的操作)。

(6) 存储器边界

如果在 DMA 传送期间传送的源地址或者目标地址超出了 DMA 目标的区域 (外部存储器, 内部 RAM, 或是片上周边 I/O), 那么操作将得不到保证。

(7) 传送未正确对齐数据

16 位总线宽度不支持传送未正确对齐数据。

如果一个奇数地址被指定为传送的源或者目标, 地址的最低有效位被强制假定为 0。

<R>

(8) 对于 CPU 的总线仲裁

由于 DMA 控制器对于总线的控制权优先于 CPU，因此发生在 DMA 传送期间的 CPU 访问被挂起直到 DMA 发送周期完成，且将总线释放给 CPU。

然而，CPU 可以访问没有执行 DMA 传送执行中的内部 ROM 和内部 RAM。

- 当在外部存储器和片上周边 I/O 之间执行 DMA 传送时，CPU 可以访问内部 ROM 和内部 RAM。
- 当在片上周边 I/O 和内部 RAM 之间执行 DMA 传送时，CPU 可以访问内部 ROM。

(9) 在 DMA 操作期间，寄存器 / 位必须进行重新写入

下列寄存器的设定需要在未执行 DMA 操作时，在下面的时序进行。

[寄存器]

- DSA_nH, DSA_nL, DDA_nH, DDA_nL, DBC_n, 和 DADC_n 寄存器
- DTFR_n.IFC_n5 ~ DTFR_n.IFC_n0 位

[时序设定]

- 在复位后到第一个 DMA 传送启动之间
- 在通道初始化完成后到 DMA 传送启动之间
- 在完成第一个 DMA 传送 (TC_n 位 = 1) 到下一个 DMA 传送启动之间的时期

(10) 一定要将下列寄存器的位清 0

- DSA_nH 寄存器的第 14~10 位
- DDA_nH 寄存器第 14~10 位
- DADC_n 寄存器第 15, 13~8, 和 3~0 位
- DCHC_n 寄存器第 6~3 位

<R>

(11) DMA 启动源因

在为多个 DMA 通道设置同样的启动触发器时，必须要小心。

如果激活的 DMA 传送通过这些 DMA 通道进行，那么具有较低优先级的 DMA 通道会比具有较高优先级的 DMA 通道更早被应答。

(12) DSA_n 和 DDA_n 寄存器中的读取数值

在 DMA 发送期间，可以在 DSA_n 和 DDA_n 寄存器中读取正在更新的值 (n 值围的取范是 0 到 3)。

例如，如果在读取 DSA_nH 寄存器和 DSA_nL 寄存器时，DMA 传送源地址 (DSA_n 寄存器) 为 0000FFFFH 且计数方向为递增 (DADC_n.SAD1 和 DADC_n.SAD0 位 = 00)，DSA_n 寄存器的值可能有所不同，这将取决于在 DSA_nH 寄存器被读取之后是否被立即执行 DMA 传送。

(a) 如果在读取 DSA_n 时没有发生 DMA 传送

- <1> DSA_nH 寄存器的读取值为: DSA_nH = 0000H
- <2> DSA_nL 寄存器的读取值为: DSA_nL = FFFFH

(b) 如果在读取 DSA_n 时发生了 DMA 传送

- <1> DSA_nH 寄存器的读取值为: DSA_nH = 0000H
- <2> 发生 DMA 传送
- <3> DSA_n 寄存器递增: DSA_n = 00100000H
- <4> DSA_nL 寄存器的读取值为: DSA_nL = 0000H

第 21 章 CRC 功能

21.1 功能

- CRC 操作电路用于检测数据块错误。
- 任何长度的数据块都可以以 8 位为单元通过 CRC-CCITT 生成多项式 ($X^{16} + X^{12} + X^5 + 1$) 产生 16 位的 CRC 码。
- 在寄存器 CRCD 设置初始值之后，每次从 CRCIN 寄存器输入 1 字节数据，CRC 码存储在 CRC 数据寄存器 CRCD 中。

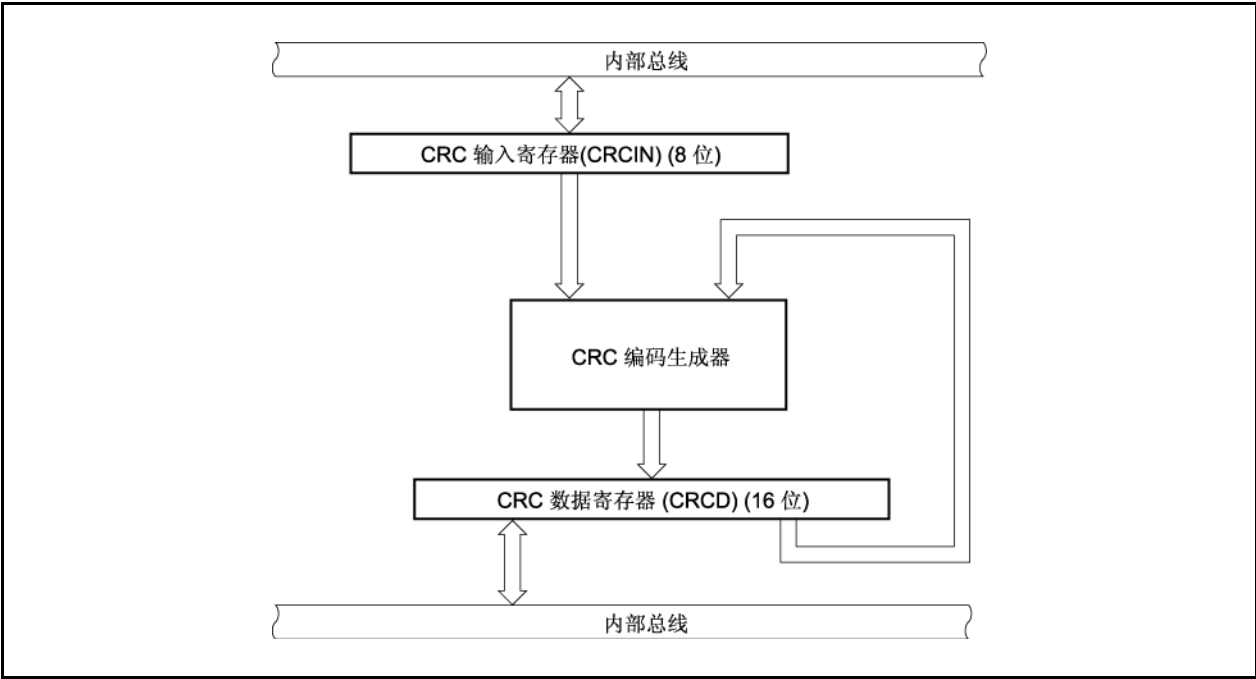
21.2 配置

CRC 功能模块包括以下的硬件。

表 21-1. CRC 配置

项目	配置
控制寄存器	CRC 输入寄存器 (CRCIN) CRC 数据寄存器 (CRCD)

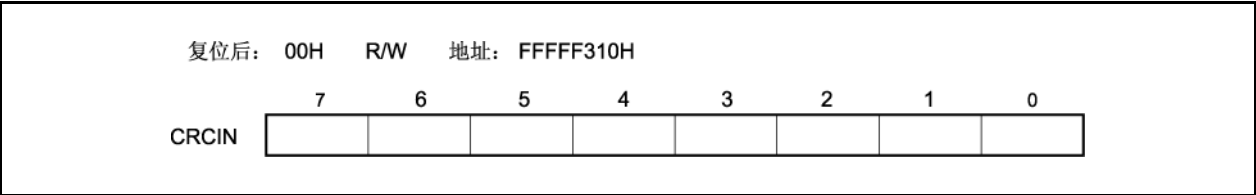
图 21-1. CRC 寄存器的框图



21.3 寄存器

(1) CRC 输入寄存器 (CRCIN)

寄存器 CRCIN 是 8 位寄存器，用于设置数据。
该寄存器可以以 8 位为单位进行读取或写入。
复位输入将把该寄存器置为 00H。

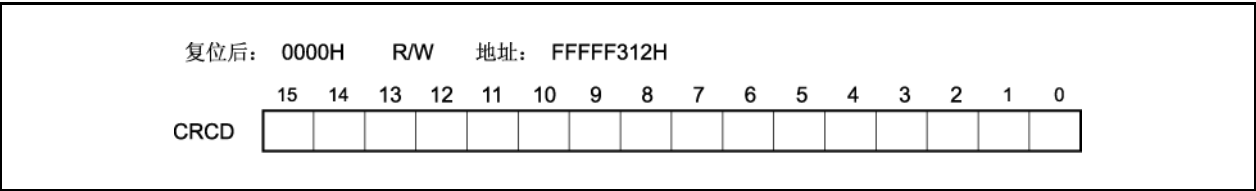


(2) CRC 数据寄存器 (CRCD)

寄存器 CRCD 是 16 位寄存器，用于存储 CRC-CCITT 的操作结果。
该寄存器可以以 16 位为单位进行读取或写入。
复位输入将把该寄存器设置为 0000H。

注意事项 下述情况下禁止访问寄存器 CRCD。详细情况，参见 3.4.9 (2) 访问特定的片上周边 I/O 寄存器。

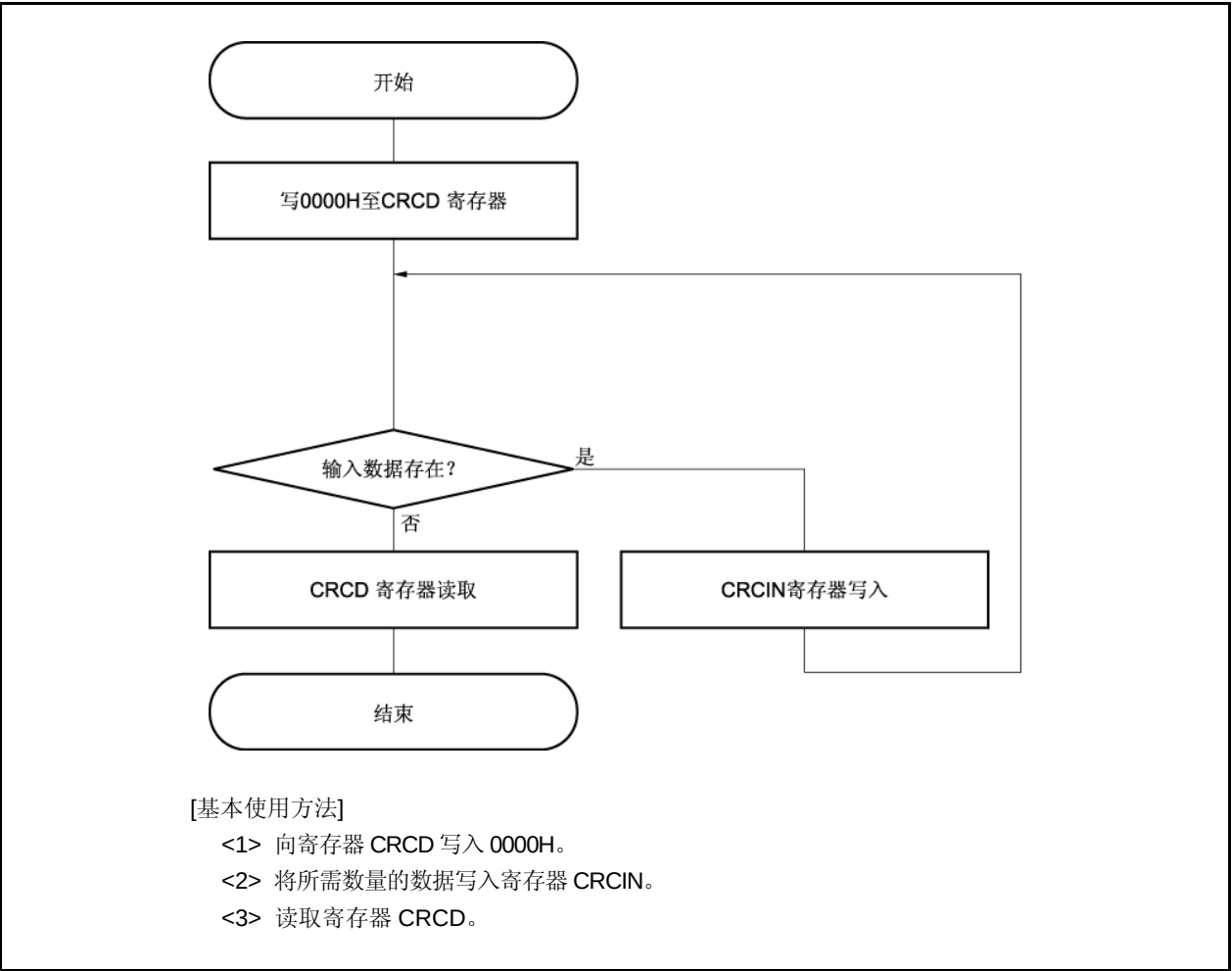
- CPU 运行于副时钟，并且主时钟振荡停止时
- CPU 运行于内部振荡时钟时



21.5 用途

下面描述如何使用 CRC 逻辑电路。

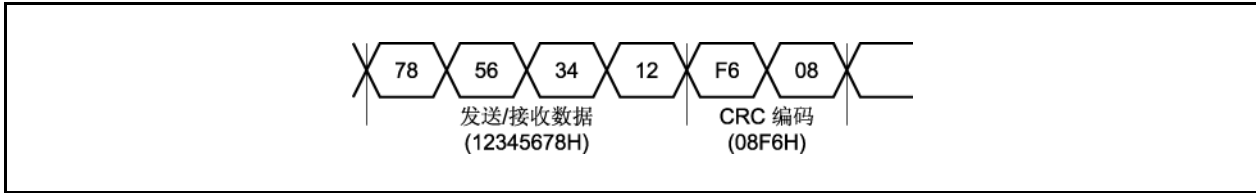
图 21-3. CRC 操作流程图



当发送/接收数据由多个字节组成时，如果 CRC 码与发送/接收数据一起发送/接收，很容易检测到通信中产生的错误。

下面将说明以 LSB 优先方式发送数据 12345678H（0001 0010 0011 0100 0101 0110 0111 1000B）的过程。

图 21-4. CRC 发送示例



发送方设置步骤

- <1> 向寄存器 CRCD 写入初始值 0000H。
- <2> 将第一个要发送的数据的 1 个字节写入发送缓冲寄存器。（同时，也要将相同的数据写入寄存器 CRCIN。）
- <3> 当发送多个字节的数据时，每次将发送数据写入发送缓冲寄存器时，都会将相同的数据写入寄存器 CRCIN。
- <4> 在所有数据都被发送后，把寄存器 CRCD 中的内容（CRC 码）写入发送缓冲寄存器并发送。（由于使用 LSB 优先方式，因此会先发从数据的低字节发送，然后是高字节。）

接收方的设置步骤

- <1> 向寄存器 CRCD 写入初始值 0000H。
- <2> 完成接收第 1 个字节后，把接收数据写入寄存器 CRCIN。
- <3> 如果接收多个字节的数据，那么每完成一次接收，都将会把接收数据写入寄存器 CRCIN。（正常接收的情况下，在所有接收到的数据被写入寄存器 CRCIN 后，接收方寄存器 CRCD 中的内容与发送方的寄存器 CRCD 中的内容相同。）
- <4> 接下来，CRC 码从发送方发出，和其它接收到的数据类似，把该数据写入寄存器 CRCIN。
- <5> 完成所有数据的接收后，包括 CRC 码，如果 CRCD 寄存器的内容是 0000H，那么接收正常。如果 CRCD 寄存器的内容不是 0000H，则会显示通信错误，因此，会将重发请求发送到发送端。

第 22 章 中断 / 异常处理功能

<R> V850ES/SG2 和 V850ES/SG2-H 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 56 至 61 个中断请求。

中断是指不受程序执行影响发生的事件，异常是指依赖于程序执行发生的事件。

<R> V850ES/SG2 和 V850ES/SG2-H 可以处理由片上周边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：获取一个非法操作码）（异常陷阱）来开始。

22.1 特点

- <R>
- 中断
 - 不可屏蔽中断：2 个源
 - 可屏蔽中断：外部：8 个中断源，内部：46/47/50/51 个中断源（参见表 1-1）
 - 8 个可编程中断优先级级别（针对可屏蔽中断）。
 - 依据优先级进行多重中断控制。
 - 针对可屏蔽中断，可以使用屏蔽选项。
 - 噪声消除，边沿检测，和有效沿是外部中断请求信号。
 - 异常
 - 软件异常：32 个源
 - 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 22-1 中。

表 22-1. 中断源列表（1/4）

类型	类别	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	—
不可屏蔽	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	—

注 1. 关于使用 INTWDT2 时的还原，参见 22.2.2（2）INTWDT2 信号。
2. n = 0 至 FH

表 22-1. 中断源列表 (2/4)

类型	类别	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
<R> 可屏蔽	中断	0	INTLVI [※]	低电压检测	POCLVI	0080H	00000080H	下一个 PC	LVIIIC
		1	INTP0	外部中断引脚输入沿检测 (INTP0)	Pin	0090H	00000090H	下一个 PC	PIC0
		2	INTP1	外部中断引脚输入沿检测 (INTP1)	Pin	00A0H	000000A0H	下一个 PC	PIC1
		3	INTP2	外部中断引脚输入沿检测 (INTP2)	Pin	00B0H	000000B0H	下一个 PC	PIC2
		4	INTP3	外部中断引脚输入沿检测 (INTP3)	Pin	00C0H	000000C0H	下一个 PC	PIC3
		5	INTP4	外部中断引脚输入沿检测 (INTP4)	Pin	00D0H	000000D0H	下一个 PC	PIC4
		6	INTP5	外部中断引脚输入沿检测 (INTP5)	Pin	00E0H	000000E0H	下一个 PC	PIC5
		7	INTP6	外部中断引脚输入沿检测 (INTP6)	Pin	00F0H	000000F0H	下一个 PC	PIC6
		8	INTP7	外部中断引脚输入沿检测 (INTP7)	Pin	0100H	00000100H	下一个 PC	PIC7
		9	INTTQ0OV	TMQ0 溢出	TMQ0	0110H	00000110H	下一个 PC	TQ0OVIC
		10	INTTQ0CC0	TMQ0 捕获 0 / 比较 0 匹配	TMQ0	0120H	00000120H	下一个 PC	TQ0CCIC0
		11	INTTQ0CC1	TMQ0 捕获 1 / 比较 1 匹配	TMQ0	0130H	00000130H	下一个 PC	TQ0CCIC1
		12	INTTQ0CC2	TMQ0 捕获 2 / 比较 2 匹配	TMQ0	0140H	00000140H	下一个 PC	TQ0CCIC2
		13	INTTQ0CC3	TMQ0 捕获 3 / 比较 3 匹配	TMQ0	0150H	00000150H	下一个 PC	TQ0CCIC3
		14	INTTP0OV	TMP0 溢出	TMP0	0160H	00000160H	下一个 PC	TP0OVIC
		15	INTTP0CC0	TMP0 捕获 0 / 比较 0 匹配	TMP0	0170H	00000170H	下一个 PC	TP0CCIC0
		16	INTTP0CC1	TMP0 捕获 1 / 比较 1 匹配	TMP0	0180H	00000180H	下一个 PC	TP0CCIC1
		17	INTTP1OV	TMP1 溢出	TMP1	0190H	00000190H	下一个 PC	TP1OVIC
		18	INTTP1CC0	TMP1 捕获 0 / 比较 0 匹配	TMP1	01A0H	000001A0H	下一个 PC	TP1CCIC0
		19	INTTP1CC1	TMP1 捕获 1 / 比较 1 匹配	TMP1	01B0H	000001B0H	下一个 PC	TP1CCIC1
		20	INTTP2OV	TMP2 溢出	TMP2	01C0H	000001C0H	下一个 PC	TP2OVIC
		21	INTTP2CC0	TMP2 捕获 0 / 比较 0 匹配	TMP2	01D0H	000001D0H	下一个 PC	TP2CCIC0
		22	INTTP2CC1	TMP2 捕获 1 / 比较 1 匹配	TMP2	01E0H	000001E0H	下一个 PC	TP2CCIC1
		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

<R> 注 仅限 V850ES/SG2

表 22-1. 中断源列表 (3/4)

类型	类别	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	26	INTTP4OV	TMP4 溢出	TMP4	0220H	00000220H	下一个 PC	TP4OVIC
		27	INTTP4CC0	TMP4 捕获 0 / 比较 0 匹配	TMP4	0230H	00000230H	下一个 PC	TP4CCIC0
		28	INTTP4CC1	TMP4 捕获 1 / 比较 1 匹配	TMP4	0240H	00000240H	下一个 PC	TP4CCIC1
		29	INTTP5OV	TMP5 溢出	TMP5	0250H	00000250H	下一个 PC	TP5OVIC
		30	INTTP5CC0	TMP5 捕获 0 / 比较 0 匹配	TMP5	0260H	00000260H	下一个 PC	TP5CCIC0
		31	INTTP5CC1	TMP5 捕获 1 / 比较 1 匹配	TMP5	0270H	00000270H	下一个 PC	TP5CCIC1
		32	INTTM0EQ0	TMM0 比较匹配	TMM0	0280H	00000280H	下一个 PC	TM0EQIC0
		33	INTCB0R/ INTIIC1 [※]	CSIB0 接收完成 / CSIB0 接收错误 / IIC1 发送完成	CSIB0/ IIC1	0290H	00000290H	下一个 PC	CB0RIC/ IICIC1
		34	INTCB0T	CSIB0 连续传送写入允许	CSIB0	02A0H	000002A0H	下一个 PC	CB0TIC
		35	INTCB1R	CSIB1 接收完成 / CSIB1 接收错误	CSIB1	02B0H	000002B0H	下一个 PC	CB1RIC
		36	INTCB1T	CSIB1 连续传送写入允许	CSIB1	02C0H	000002C0H	下一个 PC	CB1TIC
		37	INTCB2R	CSIB2 接收完成 / CSIB2 接收错误	CSIB2	02D0H	000002D0H	下一个 PC	CB2RIC
		38	INTCB2T	CSIB2 连续传送写入允许	CSIB2	02E0H	000002E0H	下一个 PC	CB2TIC
		39	INTCB3R	CSIB3 接收完成 / CSIB3 接收错误	CSIB3	02F0H	000002F0H	下一个 PC	CB3RIC
		40	INTCB3T	CSIB3 连续传送写入允许	CSIB3	0300H	00000300H	下一个 PC	CB3TIC
		41	INTUA0R/ INTCB4R	UARTA0 接收完成 / CSIB4 接收完成 / CSIB4 接收错误	UARTA0/ CSIB4	0310H	00000310H	下一个 PC	UA0RIC/ CB4RIC
		42	INTUA0T/ INTCB4T	UARTA0 连续发送允许 / CSIB4 连续发送写入允许	UARTA0/ CSIB4	0320H	00000320H	下一个 PC	UA0TIC/ CB4TIC
		43	INTUA1R/ INTIIC2 [※]	UARTA1 接收完成 / UARTA1 接收错误 / IIC2 发送完成	UARTA1/ IIC2	0330H	00000330H	下一个 PC	UA1RIC/ IICIC2
		44	INTUA1T	UARTA1 连续发送允许	UARTA1	0340H	00000340H	下一个 PC	UA1TIC
		45	INTUA2R/ INTIIC0 [※]	UARTA2 接收完成 / IIC0 发送完成	UARTA/ IIC0	0350H	00000350H	下一个 PC	UA2RIC/ IICIC0
		46	INTUA2T	UARTA2 连续发送允许	UARTA2	0360H	00000360H	下一个 PC	UA2TIC

<R>

注 仅限 I²C 总线版本 (Y 产品)

表 22-1. 中断源列表 (4/4)

类型	类别	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	47	INTAD	A/D 转换完成	A/D	0370H	00000370H	下一个 PC	ADIC
		48	INTDMA0	DMA0 发送完成	DMA	0380H	00000380H	下一个 PC	DMAIC0
		49	INTDMA1	DMA1 发送完成	DMA	0390H	00000390H	下一个 PC	DMAIC1
		50	INTDMA2	DMA2 发送完成	DMA	03A0H	000003A0H	下一个 PC	DMAIC2
		51	INTDMA3	DMA3 发送完成	DMA	03B0H	000003B0H	下一个 PC	DMAIC3
		52	INTKR	按键返回中断	KR	03C0H	000003C0H	下一个 PC	KRIC
		53	INTWTI	钟表定时器间隔	WT	03D0H	000003D0H	下一个 PC	WTIC
		54	INTWT	钟表定时器参考时间	WT	03E0H	000003E0H	下一个 PC	WTIC
		55	INTC0ERR ^{※1} / INTERR ^{※2}	AFCAN0 错误 / IEBus 错误	AFCAN0/ IEBus	03F0H	000003F0H	下一个 PC	ERRIC0/ ERRIC
		56	INTC0WUP ^{※1} / INTSTA ^{※2}	AFCAN0 唤醒 / IEBus 状态	AFCAN0/ IEBus	0400H	00000400H	下一个 PC	WUPIC0/ STSAIC
		57	INTC0REC ^{※1} / INTIE1 ^{※2}	AFCAN0 接收 / IEBus 数据中断	AFCAN0/ IEBus	0410H	00000410H	下一个 PC	RECIC0/ IEIC1
		58	INTC0TRX ^{※1} / INTIE2 ^{※2}	AFCAN0 传送 / IEBus 错误 / IEBus 状态	AFCAN0/ IEBus	0420H	00000420H	下一个 PC	TRXIC0/ IEIC2

- 注
1. 仅限 CAN 控制器版本。
 2. 仅限 IEBus 控制器版本。

- 备注
1. 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的优先级顺序。最高值为 0。
不可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, 响应不可屏蔽中断或者可屏蔽中断, 且正在执行下列指令之一时, 还原的 PC 不会成为下一个 PC (如果在中断执行期间响应中断, 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。
 2. 当发生一个非法的操作代码异常时, 非法指令的执行地址通过还原的 PC 值减去 4 得出。

22.2 不可屏蔽中断

不可屏蔽中断请求会被无条件的响应，即使是在禁止中断的条件下（DI）。NMI 不会受到优先级的控制，并且相对于其它的中断请求而言具有绝对的优先权。

在本产品中有列两种不可屏蔽中断请求信号。

- NMI 引脚输入（NMI）
- 看门狗定时器溢出产生的不可屏蔽中断请求信号（INTWDT2）

NMI 引脚的有效边沿可以选择以下四种方式：“上升沿”，“下降沿”，“双边沿”和“不检测边沿”。

当 WDTM2.WDM21 位和 WDTM2.WDM20 位设为“01”时，由看门狗定时器 2 溢出而产生的不可屏蔽中断请求信号（INTWDT2）有效。

如果同时产生两个或是更多的不可屏蔽中断请求，首先响应具有较高优先级的中断，如下所示（具有较低优先级的中断请求信号将被忽略）。

INTWDT2 > NMI

如果在处理 NMI 的同时产生新的 NMI 或 INTWDT2 请求信号，按照下面的方式服务。

（1）如果处理 NMI 的时候产生新的 NMI 请求信号

保持新的 NMI 请求信号，忽略 PSW.NP 位的值。而挂起的 NMI 请求信号会在当前的 NMI 运行完毕后被响应（在 RETI 指令执行后）。

（2）如果处理 NMI 的时候产生 INTWDT2 请求信号

当存在正在进行的 NMI 时，如果 NP 位的值依然是置位的（1），则挂起 INTWDT2 请求信号。在当前的 NMI 运行完毕后响应被挂起的 INTWDT2 请求信号（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 关于不可屏蔽中断请求信号（INTWDT2）执行的不可屏蔽中断服务，参考 22.2.2（2）INTWDT2 信号。

图 22-1. 不可屏蔽中断请求信号响应操作（1/2）

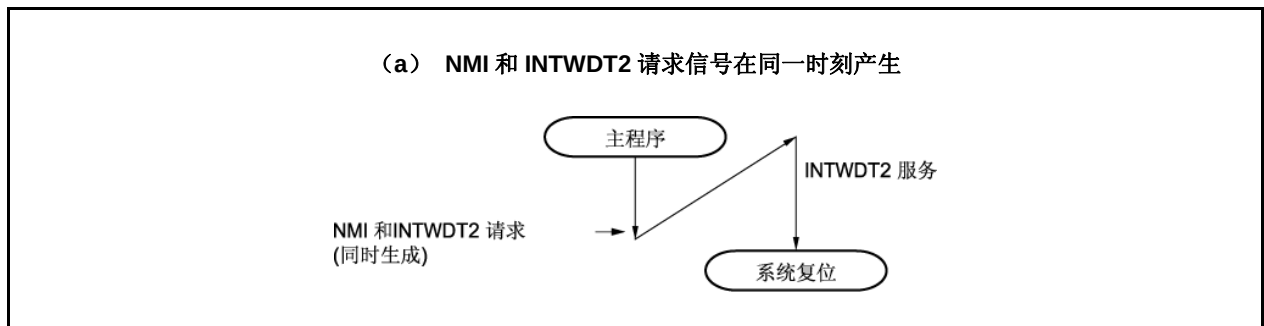
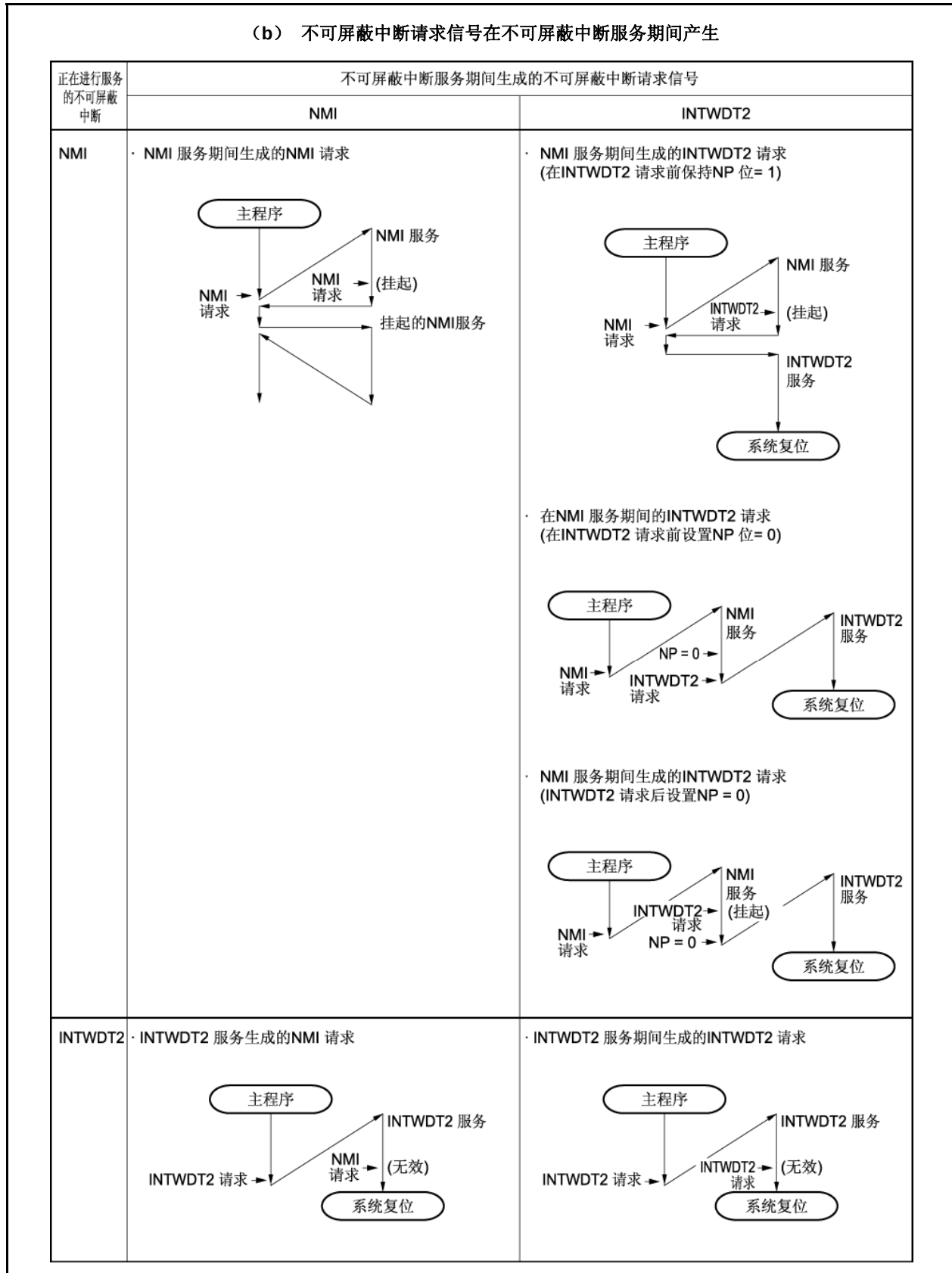


图 22-1. 不可屏蔽中断请求信号响应操作 (2/2)



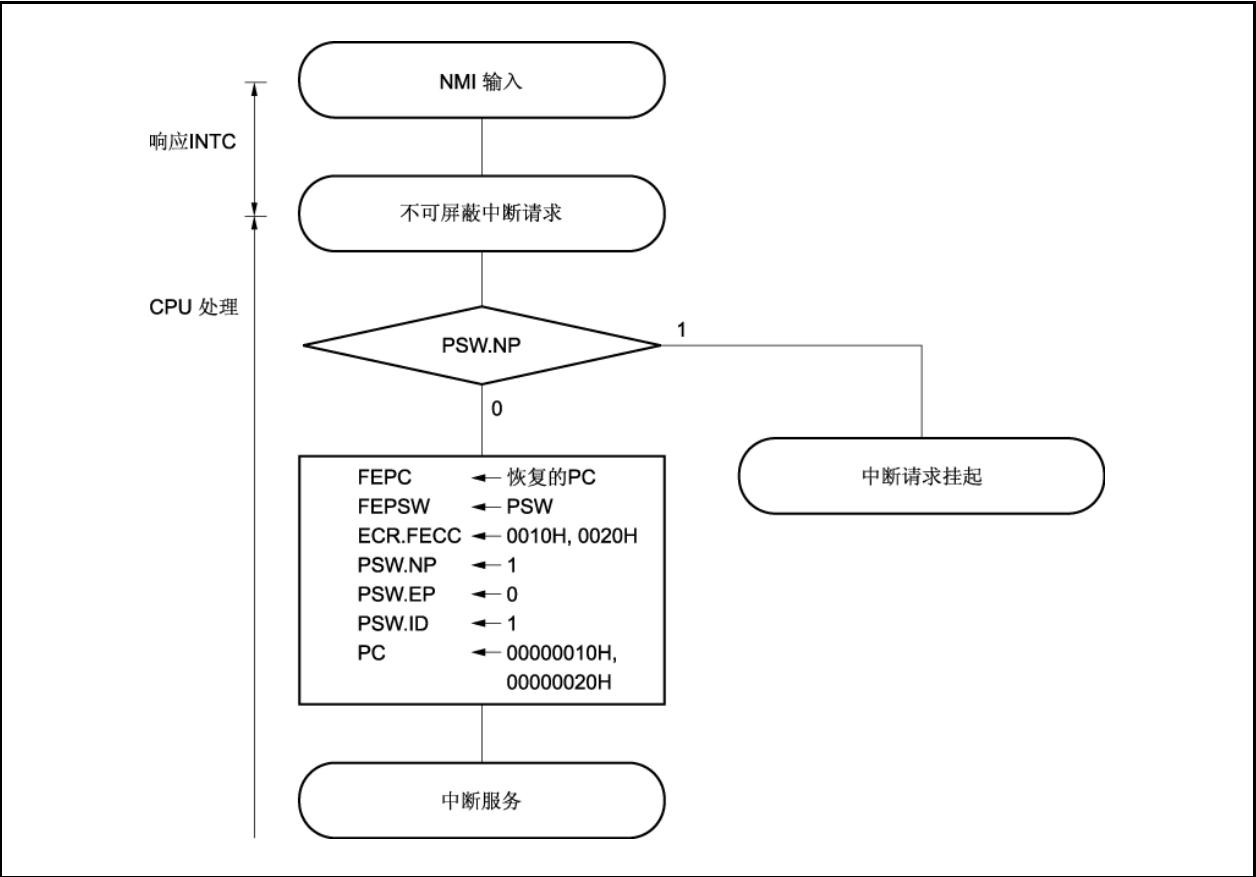
22.2.1 操作

如果产生一个不可屏蔽中断请求信号，CPU 执行下列处理，并且将操作权转移给处理程序。

- <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 中设置相应的不可屏蔽中断的处理器地址 (00000010H, 00000020H)，然后转移操作权。

不可屏蔽中断的服务过程如图 22-2 所示。

图 22-2. 不可屏蔽中断的服务过程



22.2.2 恢复

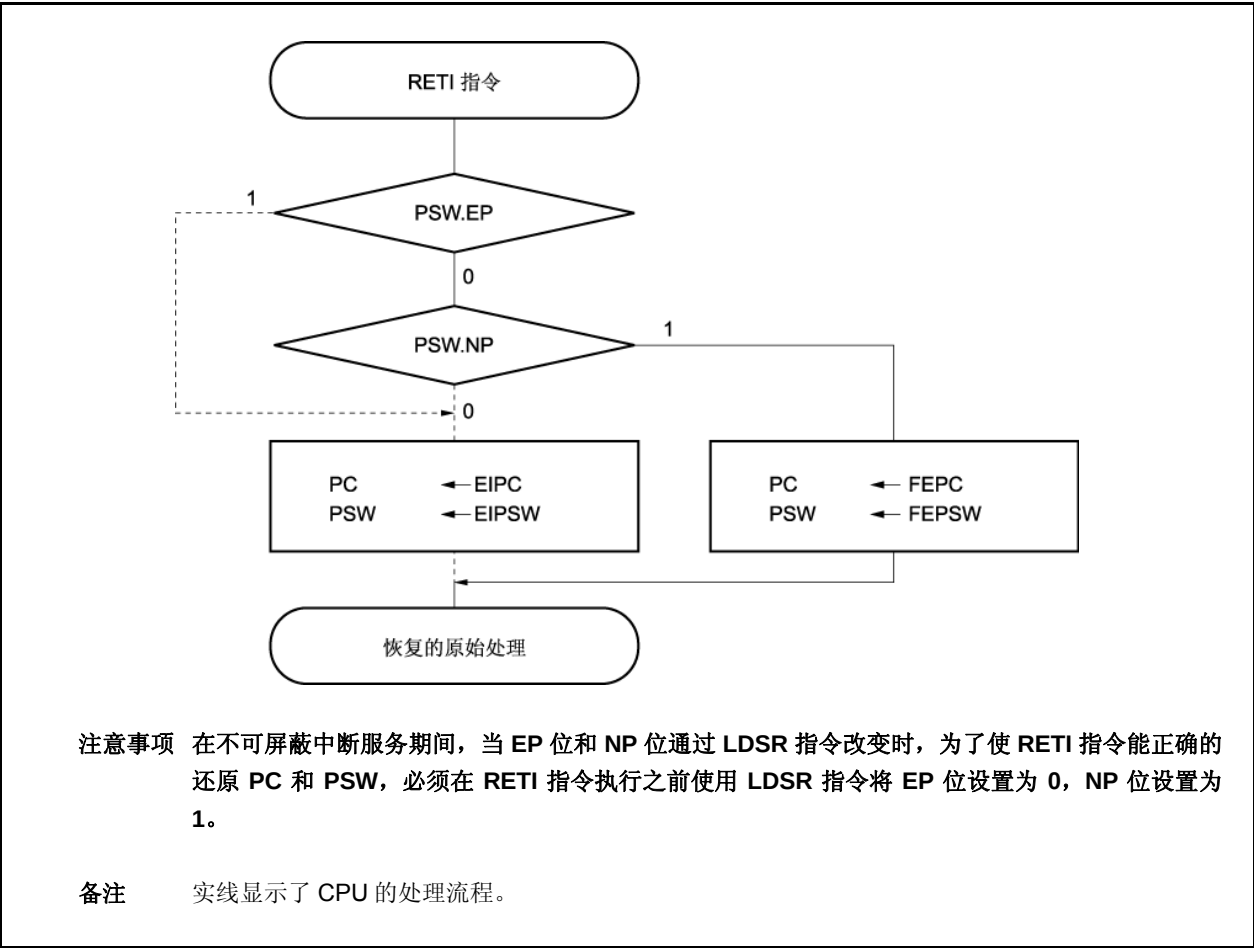
(1) 从 NMI 引脚输入

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> 将控制转移给还原到 PC 和 PSW 的地址中。

图 22-3 举例说明了 RETI 指令是如何处理的。

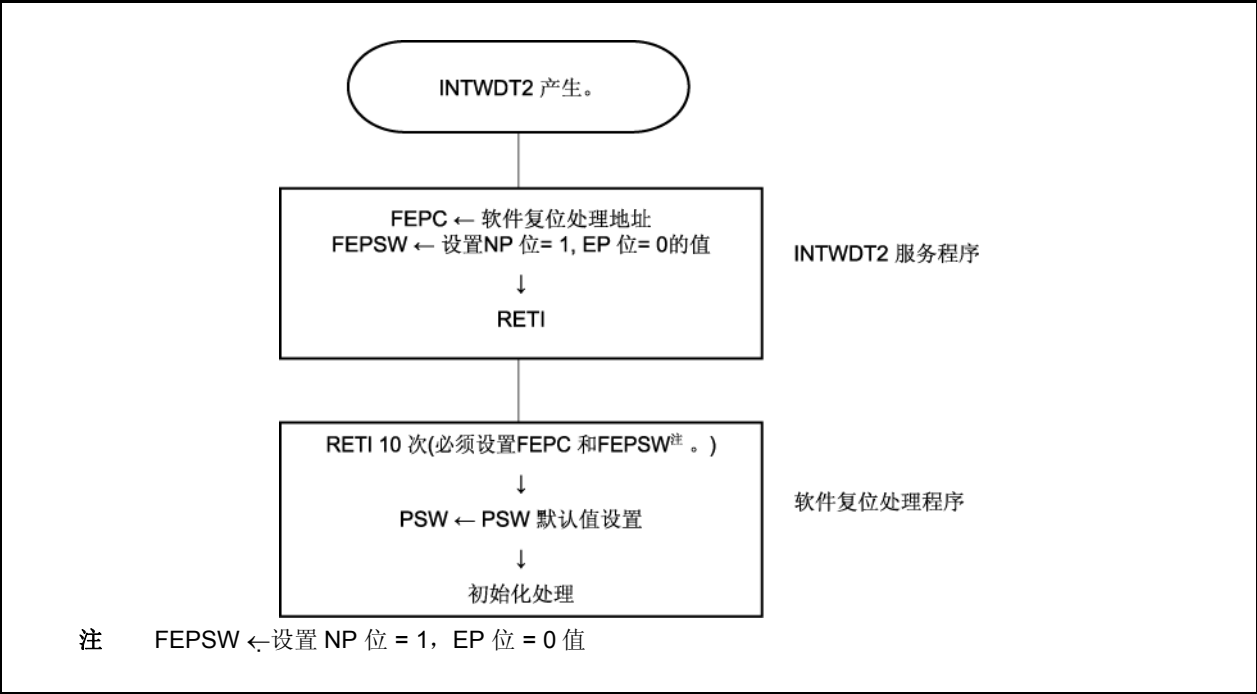
图 22-3. RETI 指令处理过程



(2) 从 INTWDT2 信号

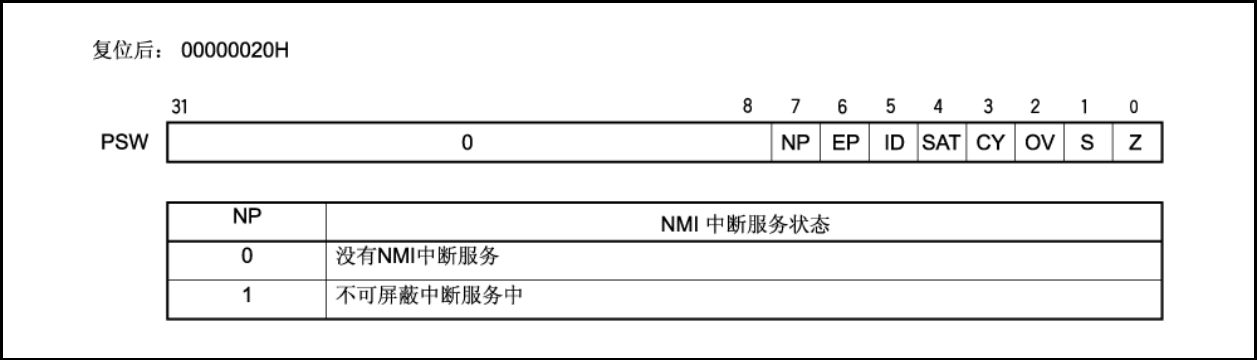
禁止使用 RETI 指令从由不可屏蔽中断请求 (INTWDT2) 请求的不可屏蔽中断服务中恢复。 执行下列软件复位处理。

图 22-4. 软件复位处理



22.2.3 NP标志

NP 标志是一个状态标志，它表明正在执行不可屏蔽中断。
当不可屏蔽中断请求被响应时会设置该标志，并且还可以保证禁止响应多重不可屏蔽中断的情况出现。



22.3 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/SG2, V850ES/SG2-H 拥有 54 至 59 个可屏蔽中断源。

如果同时产生两个或者更多的可屏蔽中断请求信号，它们将依照默认优先级的顺序被响应。除了默认优先级之外，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（可编程优先级控制）。

如果一个中断请求信号被响应，将禁止对其它可屏蔽中断请求信号的响应，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。

22.3.1 操作

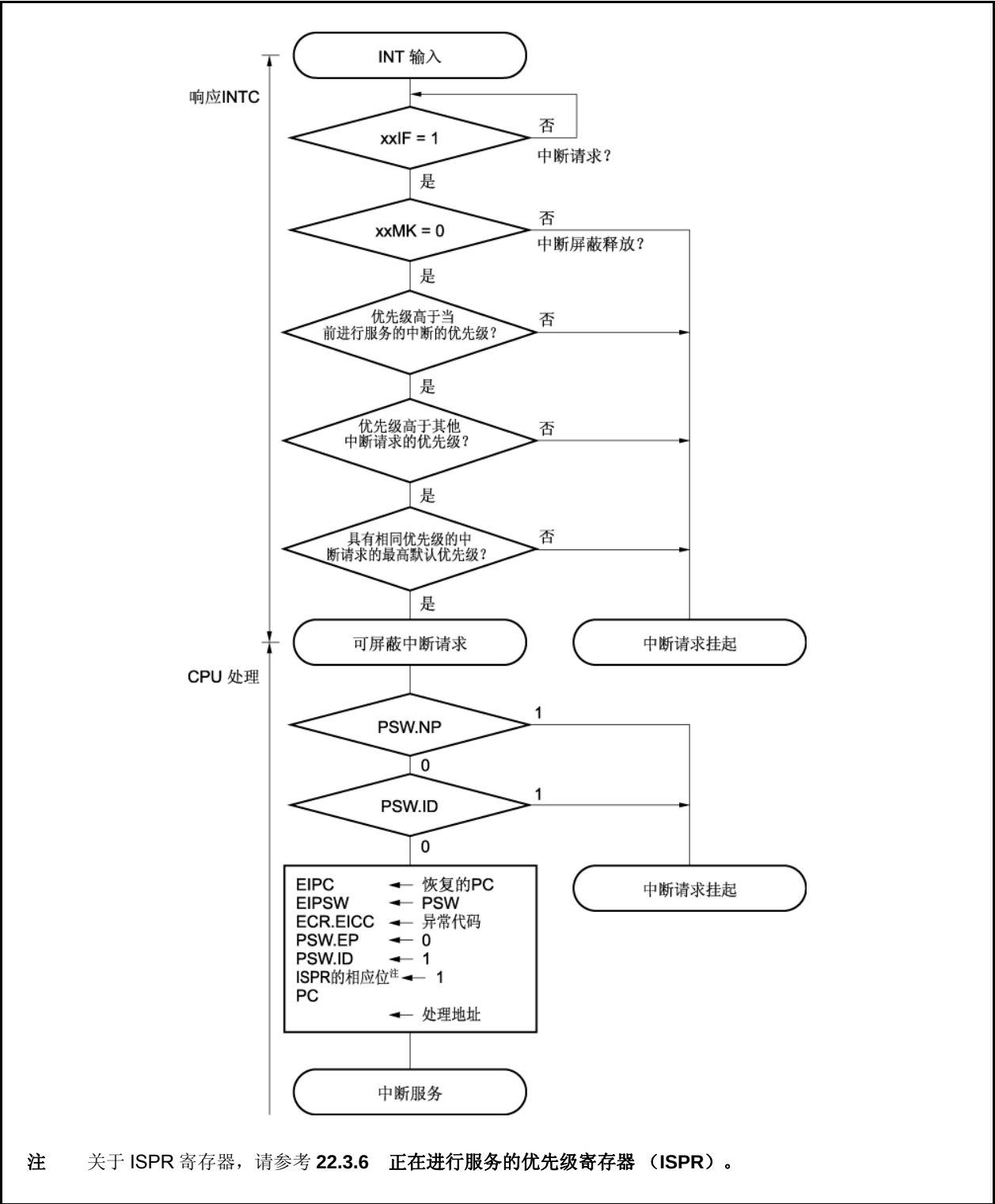
如果产生了可屏蔽中断，CPU 将执行下列处理，并且将操作权转移给处理程序。

- <1> 将恢复的 PC 保存在 EIPC 中。
- <2> 将当前的 PSW 保存在 EIPSW 中。
- <3> 向 ECR 的低半字（EICC）写入一个异常代码。
- <4> 将 PSW.ID 位设为 1 并将 PSW.EP 位清除为 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被保存在 INTC 中。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。

以下说明了可屏蔽中断是如何执行的。

图 22-5. 可屏蔽中断服务



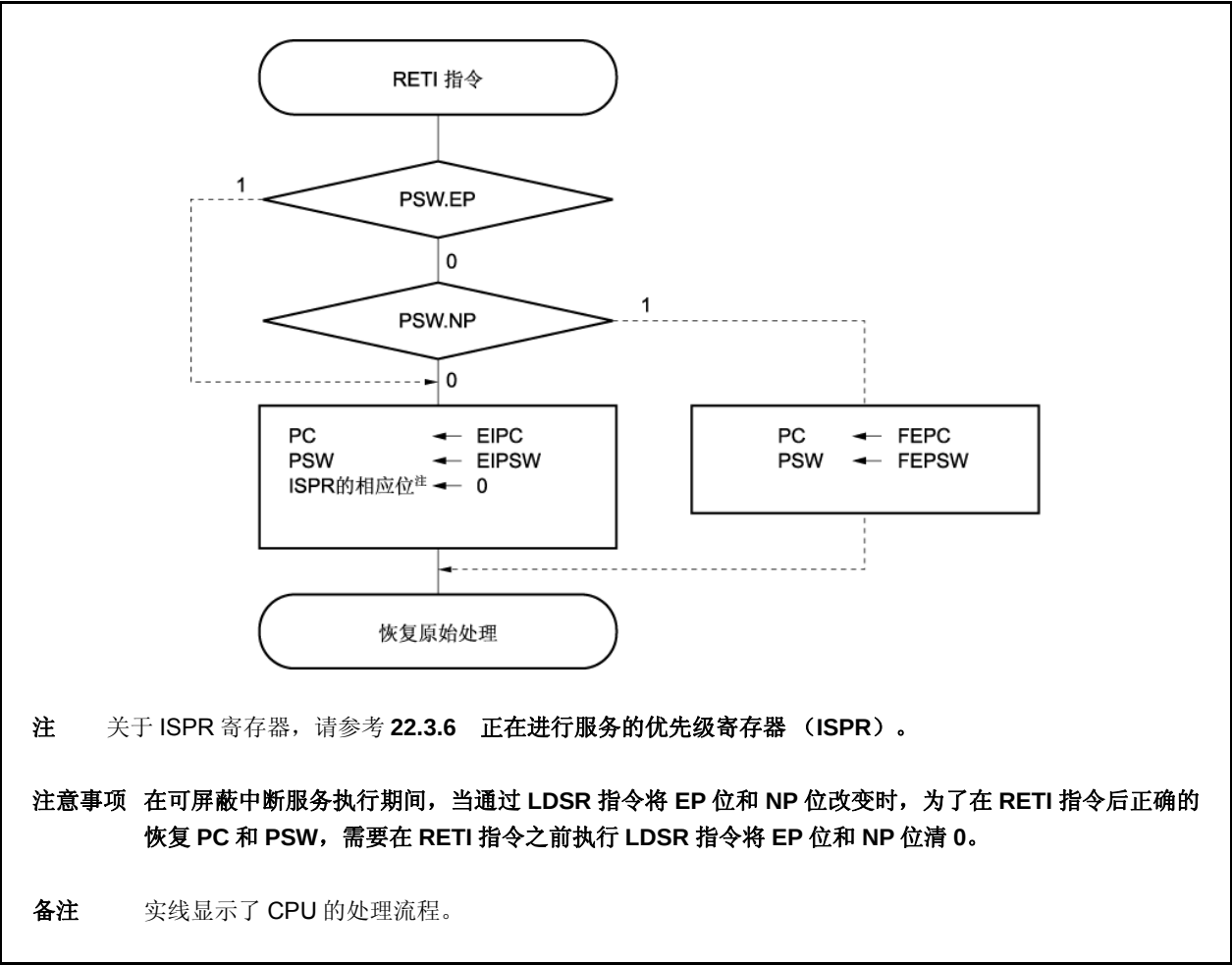
22.3.2 恢复

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> 将控制权转移到恢复 PC 和 PSW 的地址。

图 22-6 举例说明了 RETI 指令的处理过程。

图 22-6. RETI 指令处理过程



22.3.3 可屏蔽中断的优先级

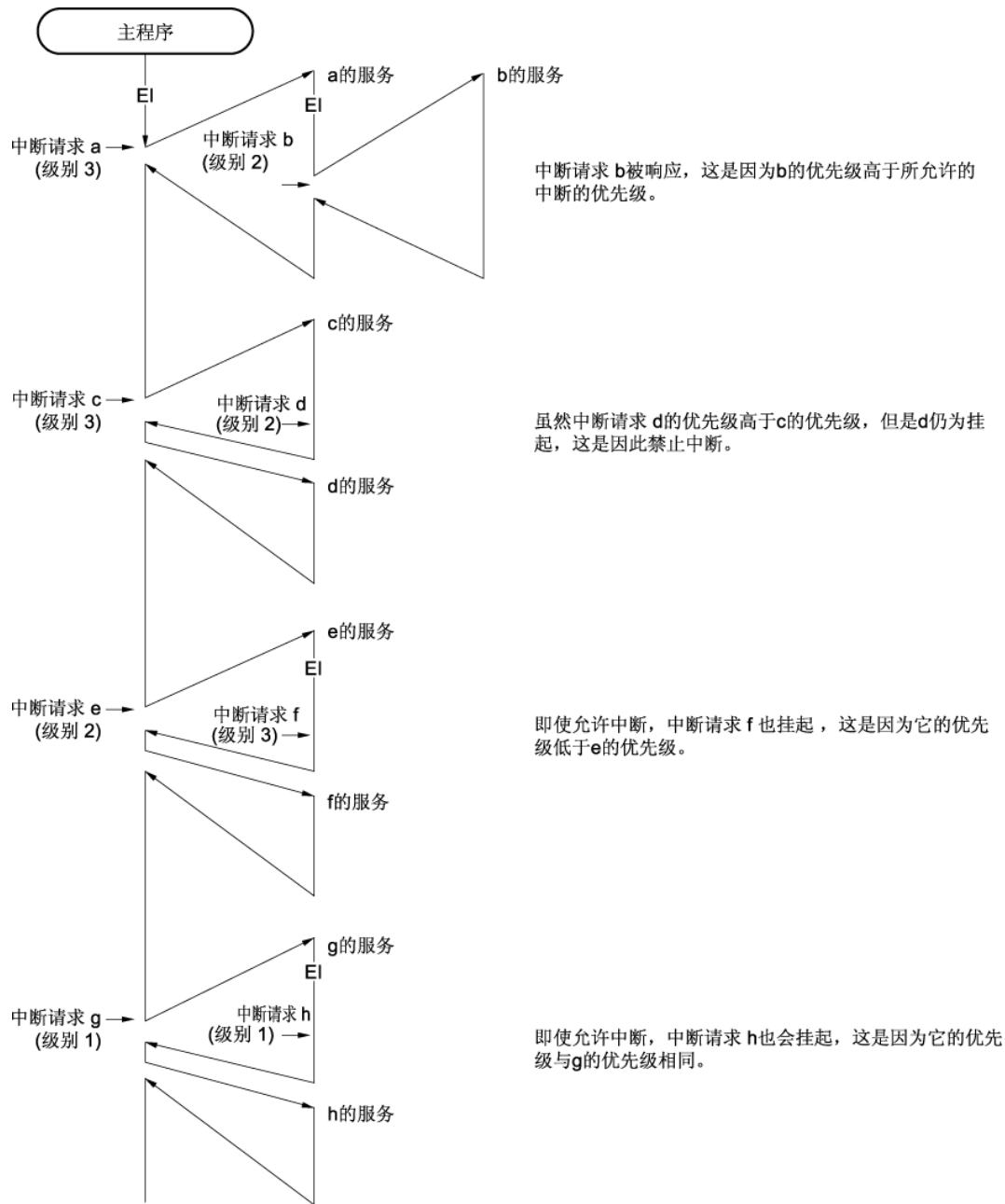
INTC 执行多重中断服务，这是指当一个中断服务正在进行的时候响应另外一个中断。多重中断可以通过优先级别来控制。

一共有两种类型的优先级别控制：基于默认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（默认优先级别）顺序进行中断服务。如需更多的信息，参考表 22-1 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: 表示任意的周边单元的名称（参考表 22-2 中断控制寄存器（xxICn））。
 n: 周边单元编号（参考表 22-2 中断控制寄存器（xxICn））。

图 22-7.中断服务时发生另一个中断请求信号时的处理示例 (1/2)



注意事项 为了执行多重中断服务，在执行 EI 指令之前必须保存 EIPC 寄存器和 EIPSW 寄存器的值。当从多重中断服务中返回，在执行 DI 指令后恢复 EIPC 和 EIPSW 的值。

备注

1. 在图中 a 到 u 表示解释说明而临时假定的中断请求信号。
2. 图中的默认优先级指示了两个中断请求信号的相对优先级。

图 22-7. 中断服务时发生另一个中断请求信号时的处理示例 2(2)

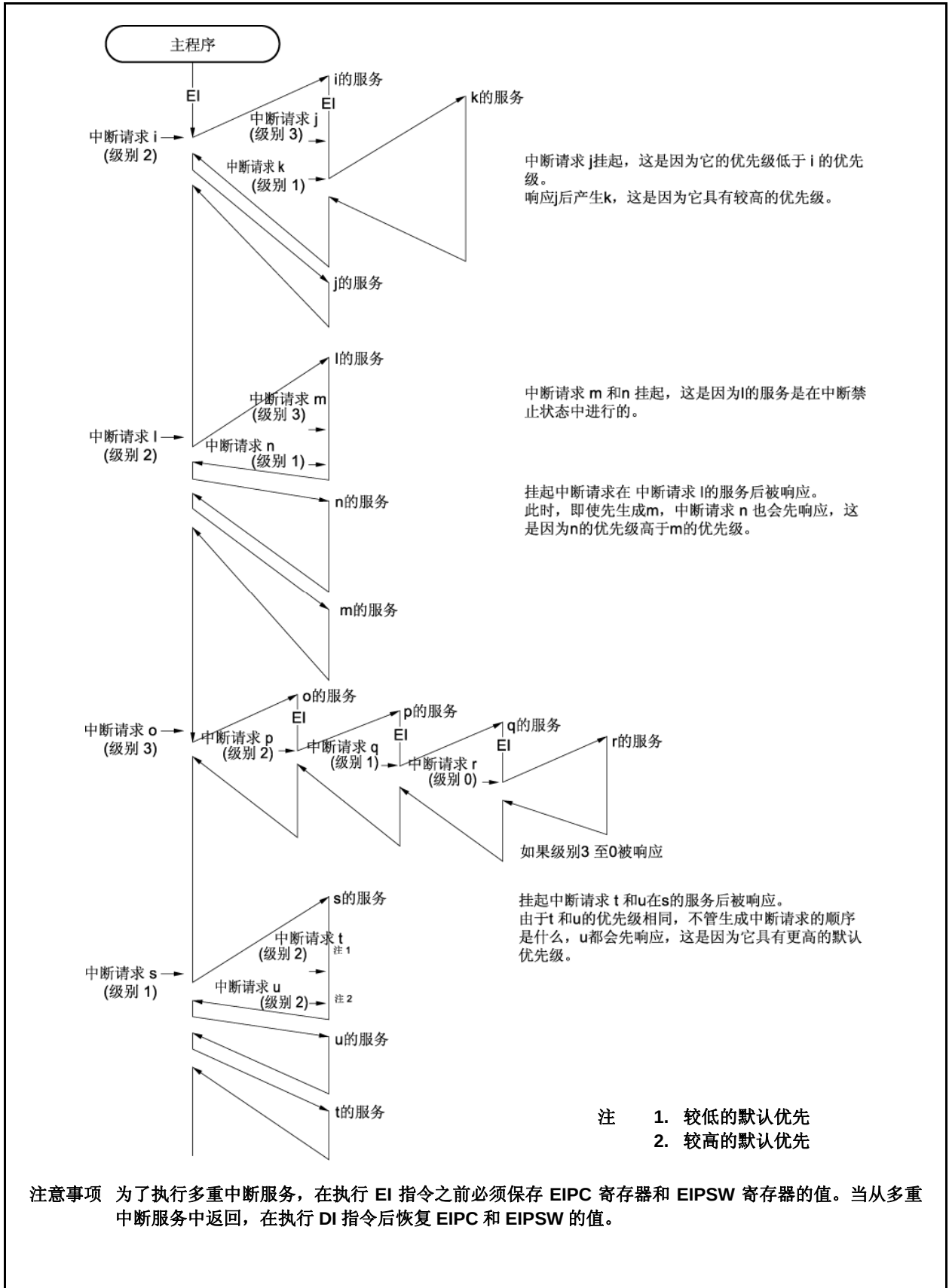
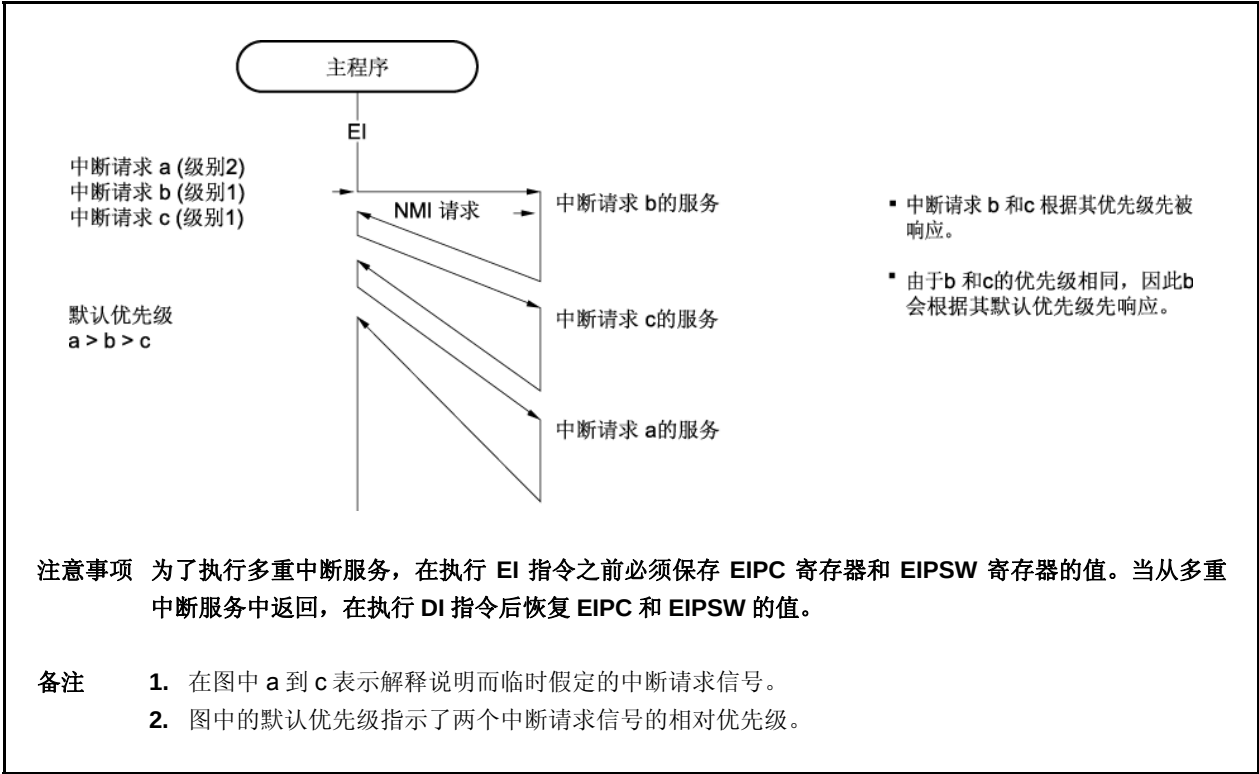


图 22-8. 服务中断请求信号同时发生举例



22.3.4 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。

该寄存器可以以 8 位或 1 位为单位进行读取操作或写操作。

复位后该寄存器的值为 47H。

注意事项 1. 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，响应中断且读取该位冲突时数据可能有误。

<R>

2. 在 V850ES/SG2-H 中，如果中断源的生成与操作已经生成的中断源中 xxMKn 或 xxPRn2 至 xxPRn0 位的位操作指令 (SET1, NOT1, 或 CLR1 (除 TST1)) 发生冲突，那么将不能生成中断请求信号。

这可以通过以下两种方法来避免。

- 当没有用位操作指令操作 xxICn 寄存器时

<1> 通过用位操作指令写入 xxMKn 位来更改 IMRm 寄存器。

<2> 通过用字节访问写入 xxPRn2 至 xxPRn0 位来更改 xxICn 寄存器。

- 当位操作指令被用于 xxICn 寄存器时

在中断允许 (DI) 状态中通过被清零的未使用的 xxICn.xxIFn 位所执行的伪写入 (字节访问) 来执行操作 xxICn 寄存器的位操作指令。

<R>

3. 当在可能产生中断请求的状态下 (包括中断禁止 (DI) 状态) 操作 xxICn.xxMKn 位时，一定要使用位操作指令或通过使用 IMRm.xxMKn 位来进行操作 (m = 0 至 3)。

然而，在 V850ES/SG2-H 中，在使用位操作指令时也应注意以上的注意事项 2。

复位后： 47H R/W 地址： FFFFF110H 至FFFFF184H

	<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	未发出中断请求
1	已发出中断请求

xxMKn	中断屏蔽标志
0	允许中断服务
1	禁止中断服务 (挂起)

xxPRn2	xxPRn1	xxPRn0	中断优先级规范位
0	0	0	指定级别 0 (最高)。
0	0	1	指定级别 1。
0	1	0	指定级别 2。
0	1	1	指定级别 3。
1	0	0	指定级别 4。
1	0	1	指定级别 5。
1	1	0	指定级别 6。
1	1	1	指定级别 7 (最低)。

注 如果响应中断信号，xxIFn 标志将通过硬件自动复位。

备注 xx: 表示任意的周边单元的名称（参考表 22-2 中断控制寄存器（xxICn））。
 n: 周边单元编号（参考表 22-2 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 22-2. 中断控制寄存器 (xxICn) (1/2)

地址	寄存器	位							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	LVIC ^{※1}	LVIF	LVIMF	0	0	0	LVIPR2	LVIPR1	LVIPR0
FFFFF112H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF114H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF116H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF118H	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
FFFFF11AH	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
FFFFF11CH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
FFFFF11EH	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60
FFFFF120H	PIC7	PIF7	PMK7	0	0	0	PPR72	PPR71	PPR70
FFFFF122H	TQ0OVIC	TQ0OVIF	TQ0OVMK	0	0	0	TQ0OVPR2	TQ0OVPR1	TQ0OVPR0
FFFFF124H	TQ0CCIC0	TQ0CCIF0	TQ0CCMK0	0	0	0	TQ0CCPR02	TQ0CCPR01	TQ0CCPR00
FFFFF126H	TQ0CCIC1	TQ0CCIF1	TQ0CCMK1	0	0	0	TQ0CCPR12	TQ0CCPR11	TQ0CCPR10
FFFFF128H	TQ0CCIC2	TQ0CCIF2	TQ0CCMK2	0	0	0	TQ0CCPR22	TQ0CCPR21	TQ0CCPR20
FFFFF12AH	TQ0CCIC3	TQ0CCIF3	TQ0CCMK3	0	0	0	TQ0CCPR32	TQ0CCPR31	TQ0CCPR30
FFFFF12CH	TP0OVIC	TP0OVIF	TP0OVMK	0	0	0	TP0OVPR2	TP0OVPR1	TP0OVPR0
FFFFF12EH	TP0CCIC0	TP0CCIF0	TP0CCMK0	0	0	0	TP0CCPR02	TP0CCPR01	TP0CCPR00
FFFFF130H	TP0CCIC1	TP0CCIF1	TP0CCMK1	0	0	0	TP0CCPR12	TP0CCPR11	TP0CCPR10
FFFFF132H	TP1OVIC	TP1OVIF	TP1OVMK	0	0	0	TP1OVPR2	TP1OVPR1	TP1OVPR0
FFFFF134H	TP1CCIC0	TP1CCIF0	TP1CCMK0	0	0	0	TP1CCPR02	TP1CCPR01	TP1CCPR00
FFFFF136H	TP1CCIC1	TP1CCIF1	TP1CCMK1	0	0	0	TP1CCPR12	TP1CCPR11	TP1CCPR10
FFFFF138H	TP2OVIC	TP2OVIF	TP2OVMK	0	0	0	TP2OVPR2	TP2OVPR1	TP2OVPR0
FFFFF13AH	TP2CCIC0	TP2CCIF0	TP2CCMK0	0	0	0	TP2CCPR02	TP2CCPR01	TP2CCPR00
FFFFF13CH	TP2CCIC1	TP2CCIF1	TP2CCMK1	0	0	0	TP2CCPR12	TP2CCPR11	TP2CCPR10
FFFFF13EH	TP3OVIC	TP3OVIF	TP3OVMK	0	0	0	TP3OVPR2	TP3OVPR1	TP3OVPR0
FFFFF140H	TP3CCIC0	TP3CCIF0	TP3CCMK0	0	0	0	TP3CCPR02	TP3CCPR01	TP3CCPR00
FFFFF142H	TP3CCIC1	TP3CCIF1	TP3CCMK1	0	0	0	TP3CCPR12	TP3CCPR11	TP3CCPR10
FFFFF144H	TP4OVIC	TP4OVIF	TP4OVMK	0	0	0	TP4OVPR2	TP4OVPR1	TP4OVPR0
FFFFF146H	TP4CCIC0	TP4CCIF0	TP4CCMK0	0	0	0	TP4CCPR02	TP4CCPR01	TP4CCPR00
FFFFF148H	TP4CCIC1	TP4CCIF1	TP4CCMK1	0	0	0	TP4CCPR12	TP4CCPR11	TP4CCPR10
FFFFF14AH	TP5OVIC	TP5OVIF	TP5OVMK	0	0	0	TP5OVPR2	TP5OVPR1	TP5OVPR0
FFFFF14CH	TP5CCIC0	TP5CCIF0	TP5CCMK0	0	0	0	TP5CCPR02	TP5CCPR01	TP5CCPR00
FFFFF14EH	TP5CCIC1	TP5CCIF1	TP5CCMK1	0	0	0	TP5CCPR12	TP5CCPR11	TP5CCPR10
FFFFF150H	TM0EQIC0	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR02	TM0EQPR01	TM0EQPR00
FFFFF152H	CB0RIC/ IICIC1 ^{※2}	CB0RIF/ IICIF1	CB0RMK/ IICMK1	0	0	0	CB0RPR2/ IICPR12	CB0RPR1/ IICPR11	CB0RPR0/ IICPR10
FFFFF154H	CB0TIC	CB0TIF	CB0TMK	0	0	0	CB0TPR2	CB0TPR1	CB0TPR0
FFFFF156H	CB1RIC	CB1RIF	CB1RMK	0	0	0	CB1RPR2	CB1RPR1	CB1RPR0
FFFFF158H	CB1TIC	CB1TIF	CB1TMK	0	0	0	CB1TPR2	CB1TPR1	CB1TPR0
FFFFF15AH	CB2RIC	CB2RIF	CB2RMK	0	0	0	CB2RPR2	CB2RPR1	CB2RPR0
FFFFF15CH	CB2TIC	CB2TIF	CB2TMK	0	0	0	CB2TPR2	CB2TPR1	CB2TPR0
FFFFF15EH	CB3RIC	CB3RIF	CB3RMK	0	0	0	CB3RPR2	CB3RPR1	CB3RPR0
FFFFF160H	CB3TIC	CB3TIF	CB3TMK	0	0	0	CB3TPR2	CB3TPR1	CB3TPR0

- 注 1. 仅限 V850ES/SG2
2. 仅限 I²C 总线版本 (Y 产品)

表 22-2. 中断控制寄存器 (xxICn) (2/2)

地址	寄存器	位							
		<7>	<6>	5	4	3	2	1	0
FFFFF162H	UA0RIC/ CB4RIC	UA0RIF/ CB4RIF	UA0RMK/ CB4RMK	0	0	0	UA0RPR2/ CB4RPR2	UA0RPR1/ CB4RPR1	UA0RPR0/ CB4RPR0
FFFFF164H	UA0TIC/ CB4TIC	UA0TIF/ CB4TIF	UA0TMK/ CB4TMK	0	0	0	UA0TPR2/ CB4TPR2	UA0TPR1/ CB4TPR1	UA0TPR0/ CB4TPR0
FFFFF166H	UA1RIC/ IICIC2 ^{※1}	UA1RIF/ IICIF2	UA1RMK/ IICMK2	0	0	0	UA1RPR2/ IICPR22	UA1RPR1/ IICPR21	UA1RPR0/ IICPR20
FFFFF168H	UA1TIC	UA1TIF	UA1TMK	0	0	0	UA1TPR2	UA1TPR1	UA1TPR0
FFFFF16AH	UA2RIC/ IICIC0 ^{※1}	UA2RIF/ IICIF0	UA2RMK/ IICMK0	0	0	0	UA2RPR2/ IICPR02	UA2RPR1/ IICPR01	UA2RPR0/ IICPR00
FFFFF16CH	UA2TIC	UA2TIF	UA2TMK	0	0	0	UA2TPR2	UA2TPR1	UA2TPR0
FFFFF16EH	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF170H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF172H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF174H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF176H	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF178H	KRIC	KRIF	KRMK	0	0	0	KRPR2	KRPR1	KRPR0
FFFFF17AH	WTIC	WTIF	WTMK	0	0	0	WTIPR2	WTIPR1	WTIPR0
FFFFF17CH	WTIC	WTIF	WTMK	0	0	0	WTPR2	WTPR1	WTPR0
FFFFF17EH	ERRIC0 ^{※2/} ERRIC ^{※3}	ERRIF0/ ERRIF	ERRMK0/ ERRMK	0	0	0	ERRPR02/ ERRPR2	ERRPR01/ ERRPR1	ERRPR00/ ERRPR0
FFFFF180H	WUPIC0 ^{※2/} STAIC ^{※3}	WUPIF0/ STAIF	WUPMK0/ STAMK	0	0	0	WUPPR02/ STAPR2	WUPPR01/ STAPR1	WUPPR00/ STAPR0
FFFFF182H	RECIC0 ^{※2/} IEIC1 ^{※3}	RECIF0/ IEIF1	RECMK0/ IEMK1	0	0	0	RECPR02/ IEPR12	RECPR01/ IEPR11	RECPR00/ IEPR10
FFFFF184H	TRXIC0 ^{※2/} IEIC2 ^{※3}	TRXIF0/ IEIF2	TRXMK0/ IEMK2	0	0	0	TRXPR02/ IEPR22	TRXPR01/ IEPR21	TRXPR00/ IEPR20

- 注
1. 仅限 I²C 总线版本 (Y 产品)
 2. 仅限 CAN 控制器版本。
 3. 仅限 IEBus 控制器版本。

22.3.5 中断屏蔽寄存器 0 至 3 (IMR0 至 IMR3)

IMR0 到 IMR3 寄存器为可屏蔽中断设定中断屏蔽状态。IMR0 到 IMR3 寄存器的 xxMKn 位相当于 xxICn.xxMKn 位。

IMRm 寄存器可以以 16 位为单位读取或是写入 (m = 0 至 3)。

如果 IMRm 寄存器的高 8 位用作 IMRmH 寄存器, 低 8 位用作 IMRmL 寄存器, 这些寄存器可以以 8 位或 1 位为单位进行读取或是写入操作 (m = 0 至 3)。

复位后, 这些寄存器的值为 FFFFH。

注意事项 设备文件定义了 xxICn.xxMKn 位为保留字。如果有其它的位使用 xxMKn 的名称进行操作, xxICn 寄存器的内容会被写入, 而不是 IMRm 寄存器 (结果, IMRm 寄存器的内容也会被覆盖)。

复位后: FFFFH		R/W	地址:		IMR3 FFFFF106H, IMR3L FFFFF106H, IMR3H FFFFF107H					
			15	14	13	12	11	10	9	8
IMR3 (IMR3H ^{注 1})			1	1	1	1	1	TRXMK0/ IEMK2	RECMK0/ IEMK1	WUPMK0/ STAMK
			7	6	5	4	3	2	1	0
IMR3L	ERRMK0/ ERRMK	WTMK	WTIMK	KRMK	DMAMK3	DMAMK2	DMAMK1	DMAMK0		

复位后: FFFFH		R/W	地址:		IMR2 FFFFF104H, IMR2L FFFFF104H, IMR2H FFFFF105H					
			15	14	13	12	11	10	9	8
IMR2 (IMR2H ^{注1})	ADMK	UA2TMK	UA2RMK/ IICMK0	UA1TMK	UA1RMK/ IIC2MK	UA0TMK/ CB4TMK	UA0RMK/ CB4RMK	CB3TMK		
	7	6	5	4	3	2	1	0		
IMR2L	CB3RMK	CB2TMK	CB2RMK	CB1TMK	CB1RMK	CB0TMK	CB0RMK/ IICMK1	TM0EQMK0		

复位后: FFFFH		R/W	地址: IMR1 FFFFF102H, IMR1L FFFFF102H, IMR1H FFFFF103H							
			15	14	13	12	11	10	9	8
IMR1 (IMR1H ^{注1})			TP5CCMK1	TP5CCMK0	TP5OVMK	TP4CCMK1	TP4CCMK0	TP4OVMK	TP3CCMK1	TP3CCMK0
			7	6	5	4	3	2	1	0
IMR1L			TP3OVMK	TP2CCMK1	TP2CCMK0	TP2OVMK	TP1CCMK1	TP1CCMK0	TP1OVMK	TP0CCMK1

复位后: FFFFH		R/W	地址:		IMR0 FFFFF100H, IMR0L FFFFF100H, IMR0H FFFFF101H					
			15	14	13	12	11	10	9	8
IMR0 (IMR0H ^{注 1)})			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 ^{注 2)}

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 1. 为了以 8 位或 1 位为单位读取 IMR0 到 IMR3 寄存器的位 8 到 15，将它们设定为 IMR0H 到 IMR3H 寄存器的位 0 到 7。
2. 这些位仅在 V850ES/SG2 中有效。
在 V850ES/SG2-H 中一定要将这些位设为 1。

注意事项 将 IMR3 寄存器的位 11 至 15 设为 1。如果更改了这些位的设置，那么将不能保证操作成功。

备注 xx: 表示任意的周边单元的名称（参考表 22-2 中断控制寄存器（xxICn））。
n: 周边单元编号（参考表 22-2 中断控制寄存器（xxICn））。

22.3.6 正在进行服务优先级寄存器 (ISPR)

ISPR 寄存器保存当前被响应的可屏蔽中断的优先级别。当响应一个中断请求时，中断请求信号的优先级别所对应该该寄存器位被设为 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清除为 0。然而，当操作从不可屏蔽中断服务或是异常处理中返回时，该位不会清除为 0。

该寄存器是只读的，可以以 8 位或是 1 位为单位进行访问。

复位输入将把该寄存器置为 00H。

注意事项 如果在中断允许状态 (EI) 对 ISPR 寄存器进行读取时响应一个中断，可能读取到的是被中断响应设置后的寄存器的值。为了在响应中断之前正确的读取 ISPR 寄存器的值，需在禁止中断 (DI) 时读取寄存器。

复位后: 00H R 地址: FFFFF1FAH

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0
ISPRn	当前响应的中断的优先级							
0	优先级为n的中断请求信号未响应							
1	优先级为n的中断请求信号已响应							

备注 n = 0 至 7 (优先级)

22.3.7 ID标志

这个标志控制可屏蔽中断的操作状态，并且存储有关允许或是禁止中断请求信号的控制信息。中断禁止标志（ID）被分配到 PSW 中。

复位后该标志的值为 00000020H。

复位后：00000020H

	31							8	7	6	5	4	3	2	1	0
PSW	0								NP	EP	ID	SAT	CY	OV	S	Z

ID	可屏蔽中断服务的规范 ^注
0	允许响应可屏蔽中断请求信号
1	禁止响应可屏蔽中断请求信号 (挂起)

注 中断禁止标志（ID）功能

通过 DI 指令时将该位置 1 并能过 EI 指令将该位清 0。当参考 PSW 时，也可以通过 RETI 指令或是 LDSR 指令修改它的值。

不可屏蔽中断请求信号和异常的响应不会考虑这个标志的设定。当一个可屏蔽中断请求信号被响应后，ID 标志位会通过硬件自动置 1。

当 xxICn.xxIFn 位设为 1 时，会响应在禁止响应中断时期（ID 标志 = 1）产生的中断请求信号，且 ID 标志清 0。

22.3.8 看门狗定时器模式寄存器 2（WDTM2）

该寄存器可以以 8 位为单位进行读取或是写入（详情参考第 11 章 看门狗定时器 2 的功能）。

复位信号产生将该寄存器设置为 67H。

复位后：67H R/W 地址：FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	0	0	0	0	0

WDM21	WDM20	看门狗定时器操作模式的选择
0	0	停止操作
0	1	不可屏蔽中断请求模式
1	×	复位模式 (初始值)

22.4 软件异常

当 CPU 执行 TRAP 指令的时候会产生一个软件异常，且总是可以被响应。

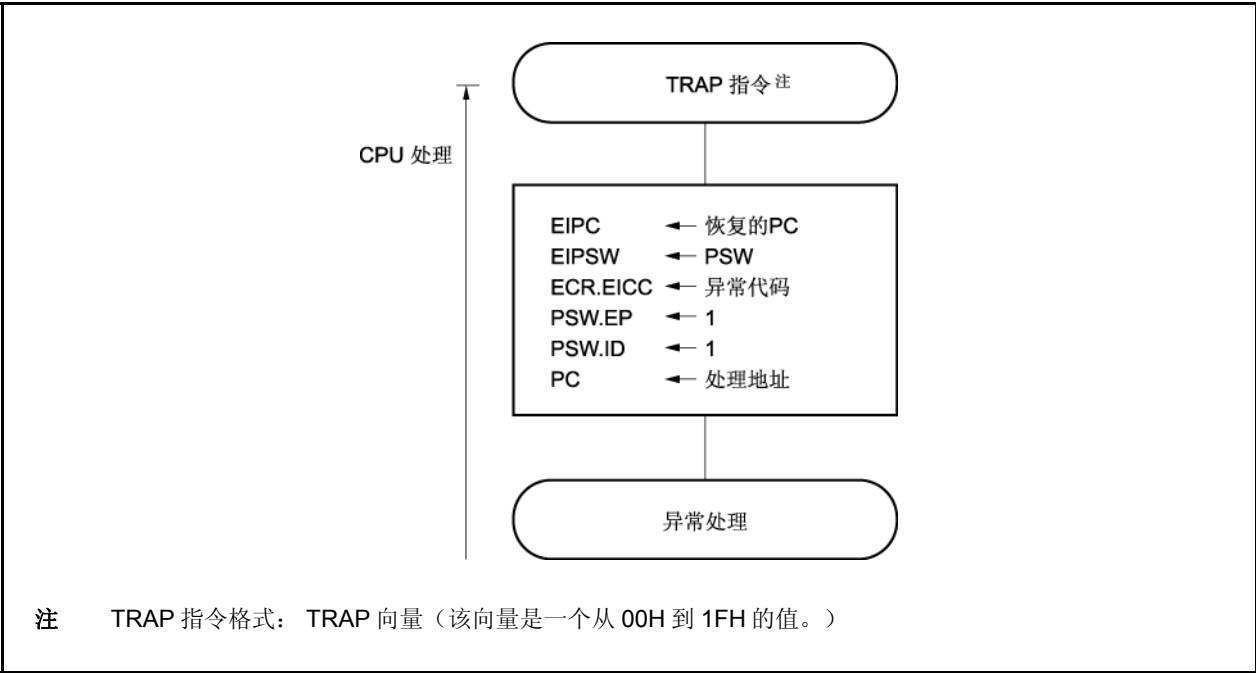
22.4.1 操作

如果产生一个软件异常，CPU 将执行下列处理过程，并且将操作权转移给处理程序。

- <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 中设置软件异常相应的处理地址（00000040H 或 00000050H），并且转移控制权。

图 22-9 举例说明软件异常的处理过程。

图 22-9. 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。

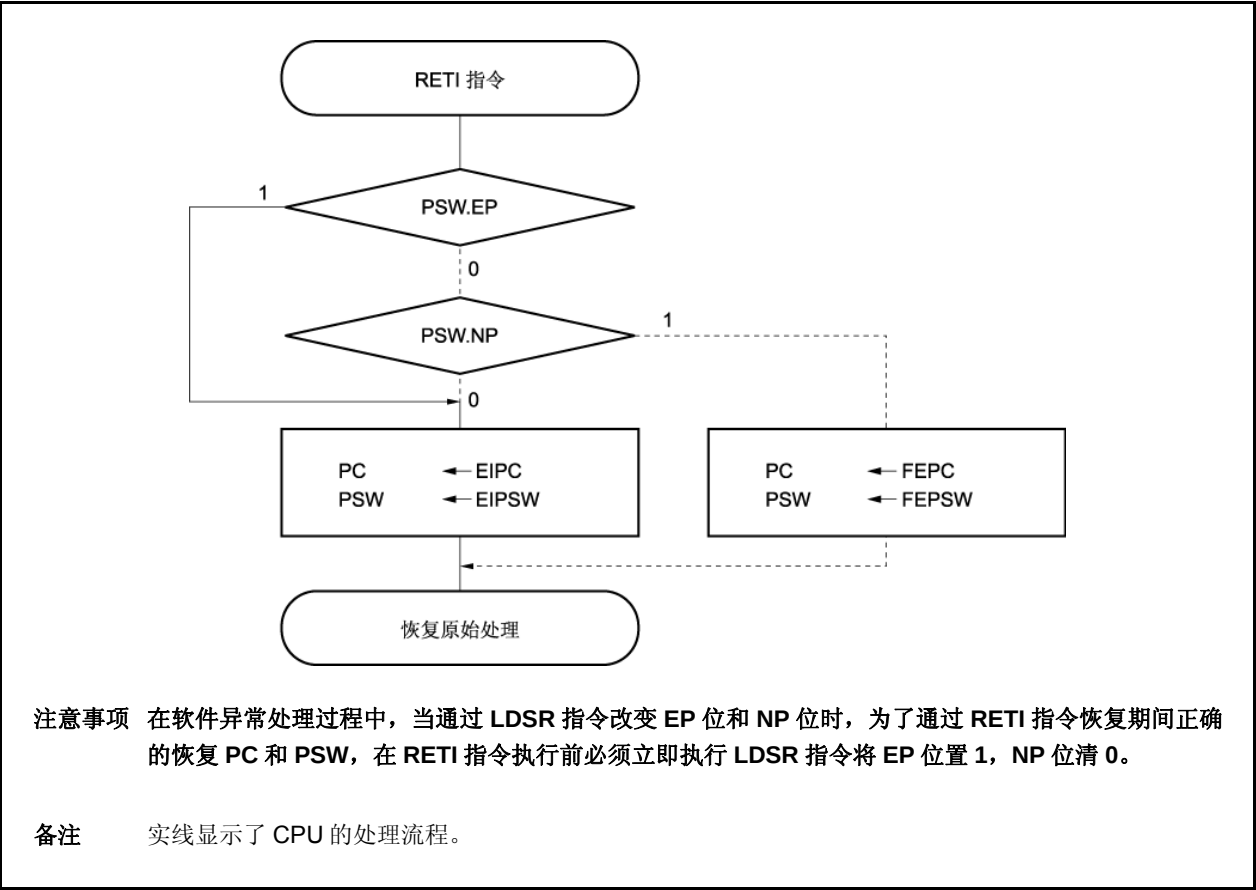
22.4.2 恢复

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的 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的 PC 地址和 PSW。

图 22-10 举例说明了 RETI 指令的处理过程。

图 22-10. RETI 指令处理过程



22.4.3 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

31

8

7

6

5

4

3

2

1

0

0

NP

EP

ID

SAT

CY

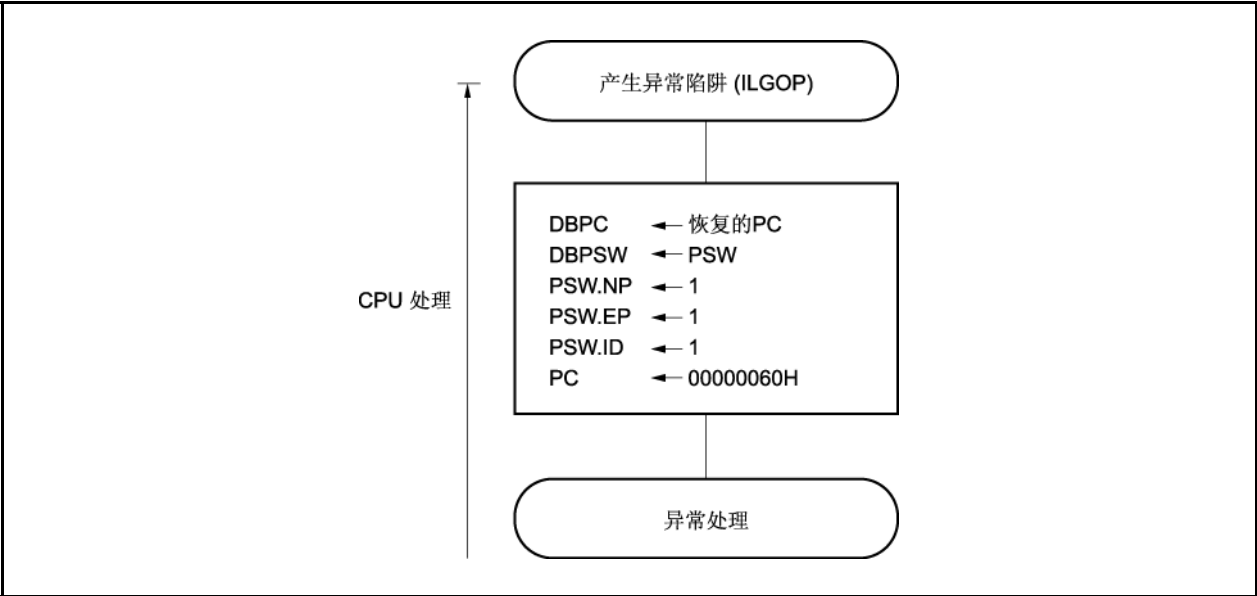
OV

S

Z

EP	异常处理状态
0	没有进行异常处理。
1	正在进行异常处理。

图 22-11. 异常陷阱的处理过程



(2) 恢复

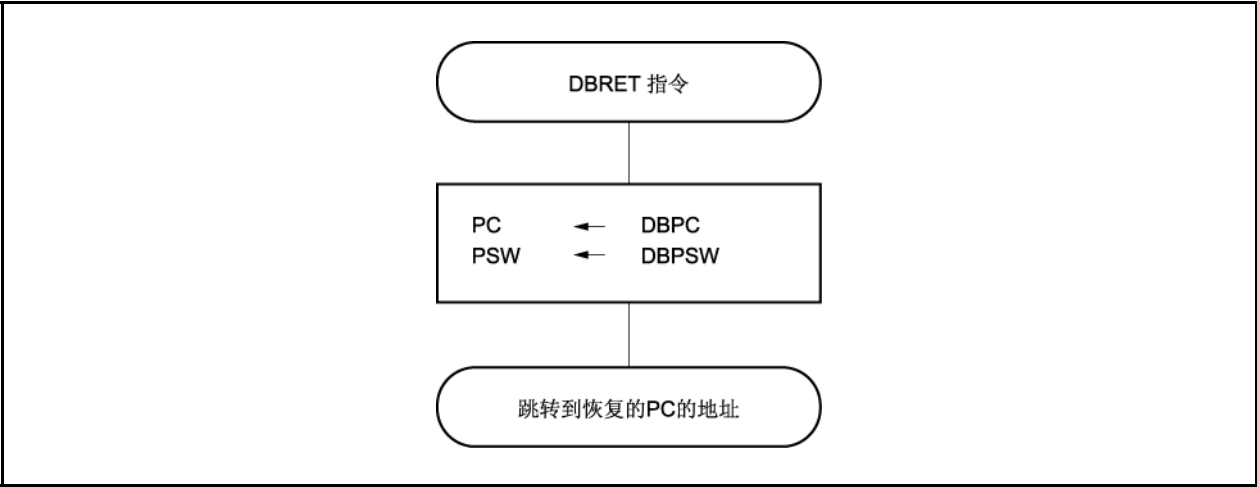
通过执行 DBRET 指令实现从异常陷阱中恢复。通过执行 DBRET 指令，CPU 执行下列处理过程并且控制着恢复 PC 的地址。

- <1> 从 DBPC 和 DBPSW 中加载恢复的 PC 和 PSW。
- <2> 向恢复的 PC 地址和 PSW 移控制权。

<R> 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 22-12 举例说明从异常陷阱中恢复的处理过程。

图 22-12. 从异常陷阱中恢复的处理过程



22.5.2 调试陷阱

调试陷阱是一个异常，它在 DBTRAP 指令执行时产生，且总是会被响应。

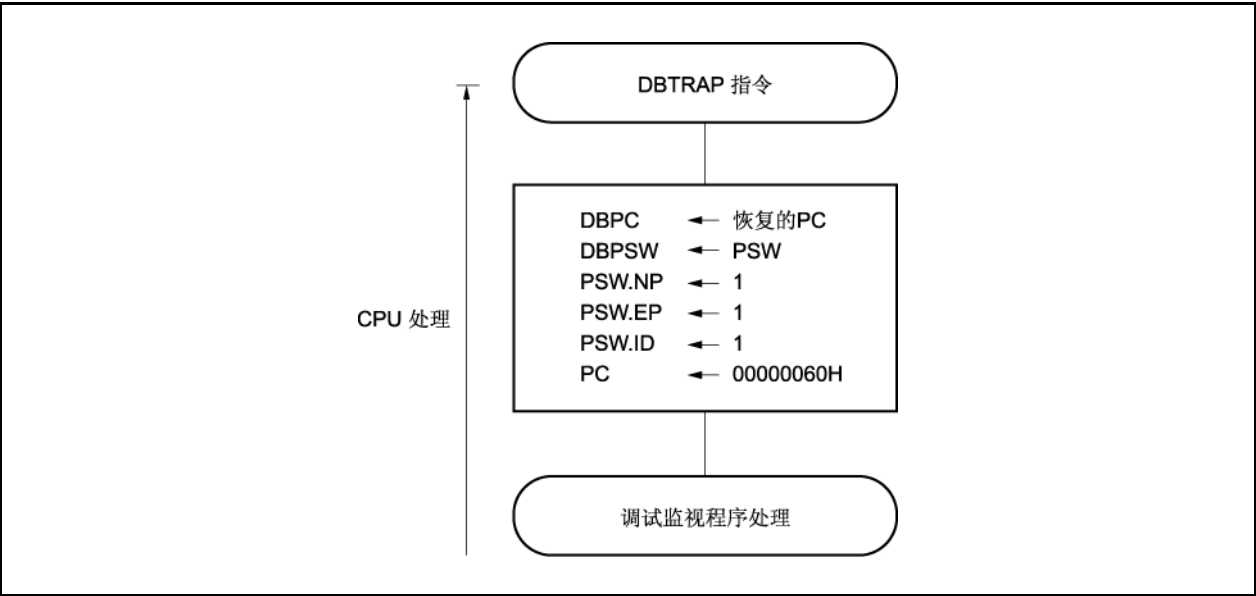
(1) 操作

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 22-13 显示了调试陷阱的处理格式。

图 22-13. 调试陷阱的处理格式



(2) 恢复

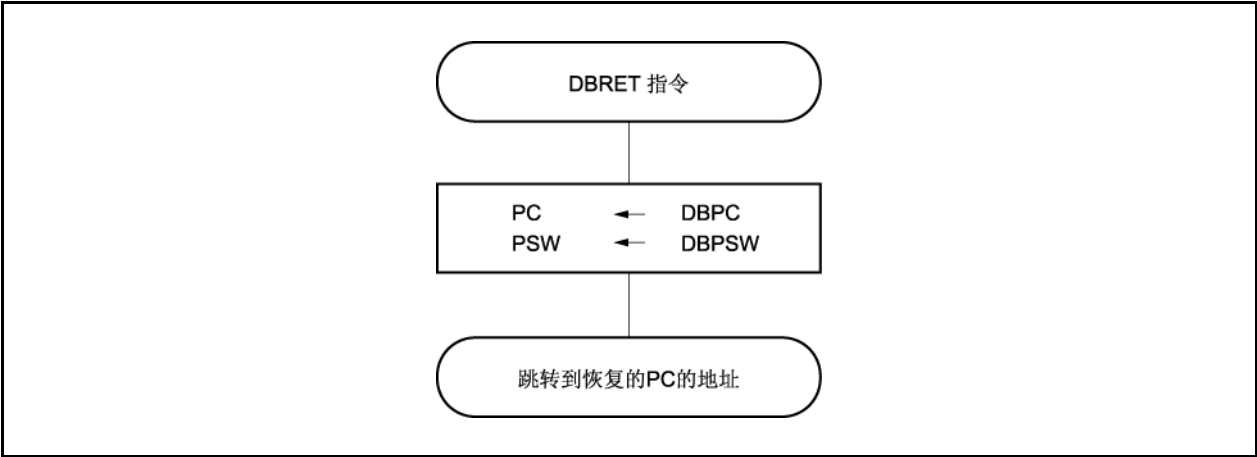
通过执行 DBRET 指令可以从调试陷阱中恢复。
执行 DBRET 指令，CPU 将执行以下步骤，并将控制权转移到恢复 PC 的地址。

- <1> 从 DBPC 和 DBPSW 中加载恢复的 PC 和 PSW。
- <2> 向恢复的 PC 中所获取的地址和 PSW 转移控制权。

注意事项 在执行 DBTRAP 指令后，执行 DBRET 指令前可以访问 DBPC 和 DBPSW。

表 22-14 显示了从调试陷阱中恢复的处理格式。

图 22-14. 从调试陷阱中恢复的处理格式



22.6 外部中断请求输入引脚（NMI 以及INTP0 至INTP7）

22.6.1 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，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。

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。

22.6.2 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
- 无边沿检测

在复位后，不会检测 NMI 引脚的边沿。因此，中断请求信号不会被响应，除非使用 INTF0 和 INTR0 寄存器使能有效边沿（NMI 引脚功能作为一个正常端口引脚）。

(1) 外部中断下降，上升沿指定寄存器 0 (INTF0, INTR0)

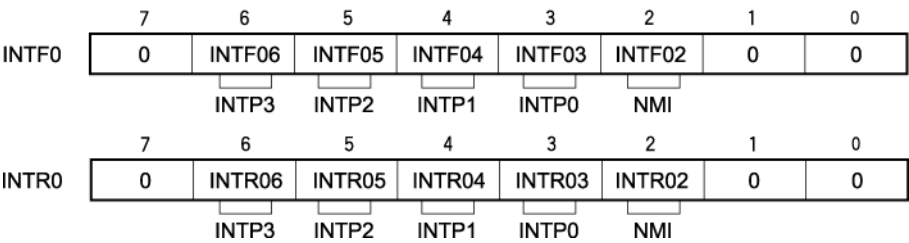
INTF0 和 INTR0 寄存器是 8 位寄存器，它们通过位 2 指定了 NMI 引脚的上升沿和下降沿的检测，通过位 3 到 6 指定了外部中断引脚 (INTP0 至 INTP3) 的上升沿和下降沿的检测。

这些寄存器可以以 8-位为单位或以 1-位为单位进行读取操作或写操作。

复位输入将把这些寄存器置为 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 FFFFFFFC00H, INTR0 FFFFFFFC20H



备注 关于如何指定有效沿，参见表 22-3。

表 22-3. 有效边沿设定

INTF0n	INTR0n	有效边沿指定 (n = 2 至 6)
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	双边沿

注意事项 当这些寄存器没有被用作 NMI 或 INTP0 到 INTP3 引脚时，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) 外部中断下降，上升沿指定寄存器 3 (INTF3, INTR3)

INTF3 和 INTR3 寄存器是 8 位寄存器，它们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-位为单位或以 1-位为单位进行读取操作或写操作。
复位输入将把这些寄存器置为 00H。

- 注意事项
- 1. 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。
 - 2. INTP7 引脚和 RXDA0 引脚是复用功能引脚。将引脚用作 RXDA0 引脚时，禁止对 INTP7 复用功能引脚的边沿检测（将 INTF3.INTF31 和 INTR3.INTR31 位清除为 0）。将引脚用作 INTP7 引脚时，停止 UARTA0 接收（将 UA0CTL0.UA0RXE 位清除为 0）。

复位后: 00H R/W 地址: INTF3 FFFFC06H, INTR3 FFFFC26H

INTF3

7	6	5	4	3	2	1	0
0	0	0	0	0	0	INTF31	0

INTP7

INTR3

7	6	5	4	3	2	1	0
0	0	0	0	0	0	INTR31	0

INTP7

备注 关于如何指定有效沿，参见表 22-4。

表 22-4. 有效边沿设定

INTF31	INTR31	有效边沿设定
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	双边沿

注意事项 当这些寄存器没用被用作 INTP7 引脚时一定要将 INTF31 和 INTR31 位清除为 00。

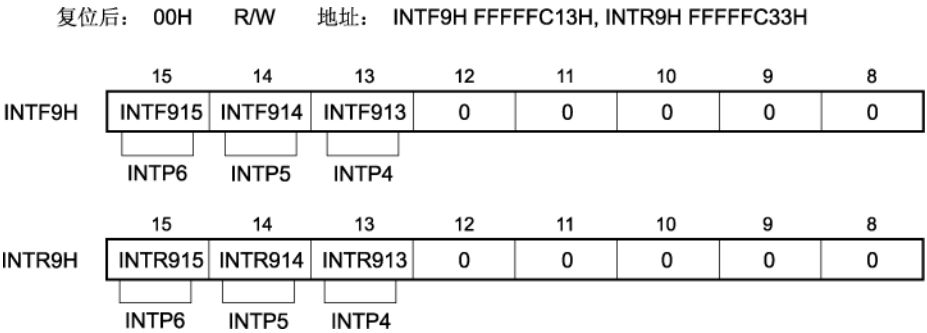
(3) 外部中断下降，上升沿指定寄存器 9H (INTF9H, INTR9H)

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-位为单位或以 1-位为单位进行读取操作或写操作。

复位输入将把这些寄存器置为 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 关于如何指定有效沿，参见表 22-5。

表 22-5. 有效边沿设定

INTF9n	INTR9n	有效边沿指定 (n = 13 至 15)
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	双边沿

注意事项 当这些寄存器没用被用作 INTP4 至 INTP6 引脚时一定要将 INTF9n 和 INTR9n 位清除为 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 fxx/64, fxx/128, fxx/256, fxx/512, fxx/1,024 以及 fxt 中选择。一共执行 3 次采样。

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 fxt 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。

该寄存器可以以 8 位为单位进行读取或写入。

复位输入将把该寄存器置为 00H。

<R>

注意事项 采样时钟被更改后将会需要 3 个采样时钟来初始化数字噪声消除器。因此，如果在改变采样时钟后的 3 个采样时钟内输入 INTP3 有效沿，那么可能会生成中断请求信号。因此，使用中断和 DMA 功能时应注意以下几点。

- 使用中断功能时，在 3 个采样时钟过后，将会在清除中断请求标志 (PIC3.PIF3 位) 后使能中断。
- 使用 DMA 功能 (通过 INTP3 开始) 时，在 3 个采样时钟过后将会使能 DMA。

复位后: 00H R/W 地址: FFFF318H

	7	6	5	4	3	2	1	0
NFC	NFEN	0	0	0	0	NFC2	NFC1	NFC0

NFEN	INTP3引脚噪声消除的设置
0	模拟噪声消除 (60 ns (TYP.))
1	数字噪声消除

NFC2	NFC1	NFC0	数字采样时钟
0	0	0	fxx/64
0	0	1	fxx/128
0	1	0	fxx/256
0	1	1	fxx/512
1	0	0	fxx/1,024
1	0	1	fxt (子时钟)
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。

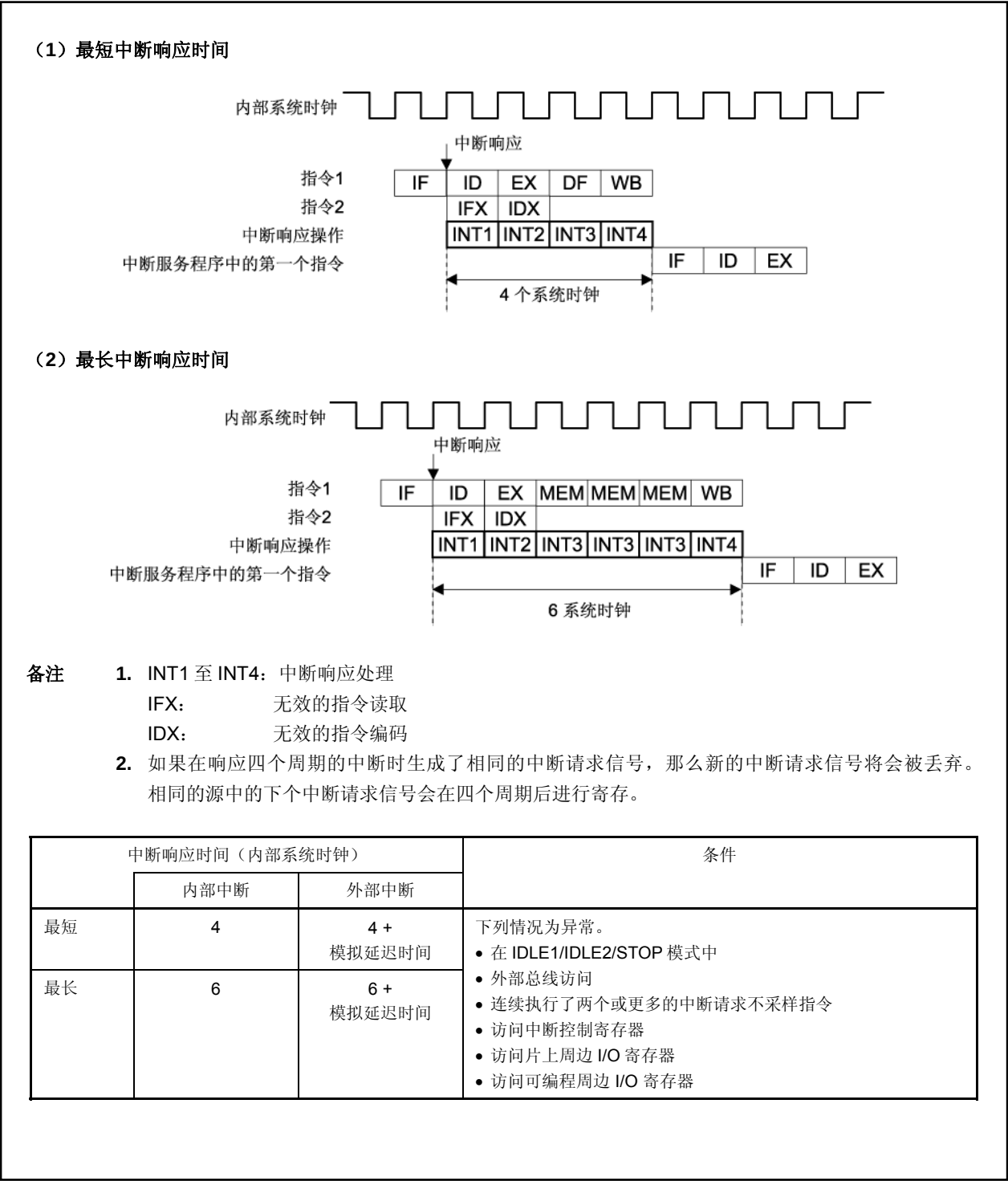
22.7 CPU的中断响应时间

除非出现下列情况，CPU 的中断响应时间最少需要 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
 - 访问外部总线时
 - 成功的执行了中断请求不采样指令时（参考 22.8 CPU 不对中断进行响应的时期。）
 - 当访问中断控制寄存器时
- <R> • 访问片上周边 I/O 寄存器时
- <R> • 访问可编程周边 I/O 寄存器时

<R>

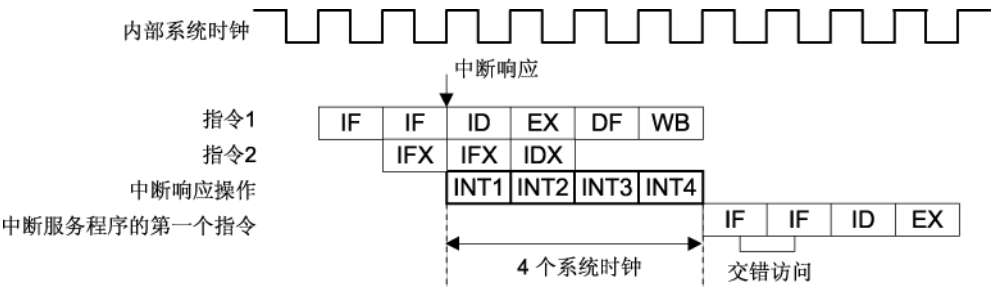
图 22-15. 在 V850ES/SG2 的中断请求信号响应过程中的流水线操作（概要）



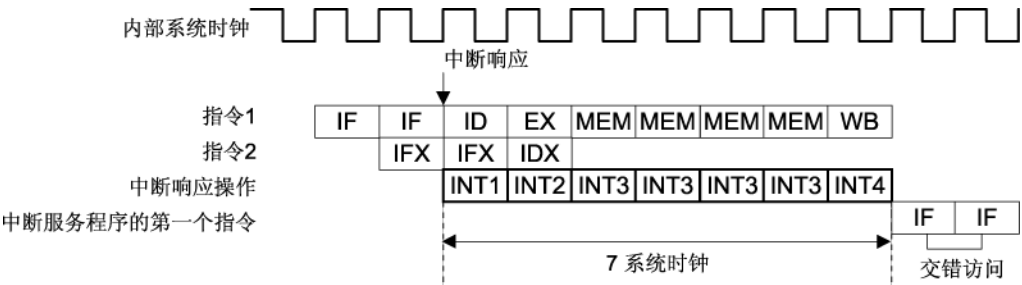
<R>

图 22-16. 在 V850ES/SG2-H 的中断请求信号响应过程中的流水线操作（概要）

(1) 最短中断响应时间



(2) 最长中断响应时间



- 备注
1. INT1 至 INT4: 中断响应处理
IFX: 无效的指令读取
IDX: 无效的指令编码
 2. 如果在响应四个周期的中断时生成了相同的中断请求信号，那么新的中断请求信号将会被丢弃。
相同的源中的下个中断请求信号会在四个周期后进行寄存。

	中断响应时间（内部系统时钟）		条件
	内部中断	外部中断	
最短	4	4 + 模拟延迟时间	下列情况为异常。 • 在 IDLE1/IDLE2/STOP 模式中 • 外部总线访问 • 连续执行了两个或更多的中断请求不采样指令 • 访问中断控制寄存器 • 访问片上周边 I/O 寄存器 • 访问可编程周边 I/O 寄存器
最长	7	7 + 模拟延迟时间	

22.8 CPU不对中断进行响应的时期

当执行指令时，中断会通过 CPU 响应。但是，在中断请求不采样指令及下一指令之间不响应中断（中断被挂起）。

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 3（IMR0 至 IMR3）
 - 节能控制寄存器（PSC）
 - 片上调试模式寄存器（OCDM）

备注 xx: 表示任意的周边单元的名称（参考表 22-2 中断控制寄存器（xxICn））。

 n: 周边单元编号（参考表 22-2 中断控制寄存器（xxICn））。

22.9 注意事项

(1) NMI 引脚

NMI 引脚和 P02 引脚是复用功能引脚，而在复位后则用作普通端口引脚。为了使能 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“无边沿检测”。使用 INTF0 寄存器和 INTR0 寄存器设定 NMI 引脚的有效边沿。

<R>

(2) V850ES/SG2-H 的中断控制寄存器（xxICn）

在 V850ES/SG2-H 中，如果中断源的生成与操作已经生成的中断源中 xxMKn 或 xxPRn2 至 xxPRn0 位的位操作指令（SET1，NOT1，或 CLR1（除 TST1））发生冲突，那么将不能生成中断请求信号。

这可以通过以下两种方法来避免。

- 当位操作指令没有用于 xxICn 寄存器时
 - <1> 用位操作指令写 xxMKn 位来操作 IMRm 寄存器。
 - <2> 通过访问 xxICn 寄存器的字节来改写 xxPRn2 至 xxPRn0 位。
- 当位操作指令被用于 xxICn 寄存器时
 - 在中断允许（DI）状态中通过被清零的未使用的 xxICn.xxIFn 位所执行的伪写入（字节访问）来执行操作 xxICn 寄存器的位操作指令。

<R>

(3) V850ES/SG2 和 V850ES/SG2-H 的中断控制寄存器（xxICn）

在可以产生中断请求的状态（包括中断禁止（DI）状态）下操作 xxICn.xxMKn 位时，一定要使用位操作指令或通过使用 IMRm.xxMKn 位来进行操作（m = 0 至 3）。

然而，在 V850ES/SG2-H 中，在使用位操作指令时也应注意上述（2）。

<R>

(4) 正在进行服务优先级寄存器（ISPR）

如果在中断允许状态（EI）对 ISPR 寄存器进行读取时响应一个中断，可能读到的是 ISPR 寄存器被中断响应置位后的值。为了在响应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。

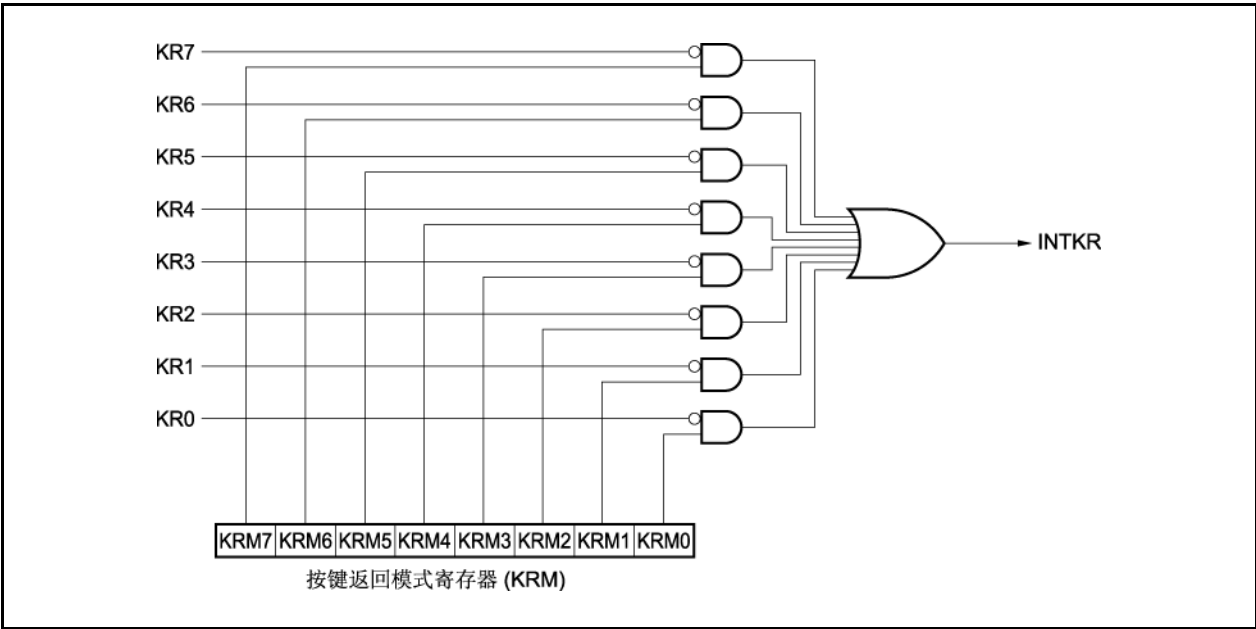
23.1 功能

通过设置 KRM 寄存器，给八个按键输入引脚（KR0 至 KR7）中的对应引脚输入下降沿，可以产生按键中断请求信号（INTKR）。

表 23-1. 按键返回检测引脚的分配

标志	引脚描述
KRM0	位操作控制 KR0 信号
KRM1	位操作控制 KR1 信号
KRM2	位操作控制 KR2 信号
KRM3	位操作控制 KR3 信号
KRM4	位操作控制 KR4 信号
KRM5	位操作控制 KR5 信号
KRM6	位操作控制 KR6 信号
KRM7	位操作控制 KR7 信号

图 23-1. 按键返回框图



23.2 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。

该寄存器可以以字为单位或以 1-位为单位进行读写操作。

复位输入将把该寄存器置为 00H。

复位后: 00H R/W 地址: FFFFF300H

	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	按键返回模式的控制
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 4-15 使用端口引脚作为复用功能引脚。

23.3 注意事项

- (1) 如果给 KR0 至 KR7 中的任何一个输入低电平, 即使给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。要使用 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, 则可能会产生中断请求信号 (INTKR)。为避免这种情况, 在禁止中断 (DI) 或屏蔽中断之后改变 KRM 寄存器, 然后将中断请求标志 (KRIC.KRIF 位) 清零, 再允许中断 (EI) 或清除屏蔽。
- (4) 为了使用按键中断功能, 一定要设置端口引脚为按键返回引脚, 然后通过 KRM 寄存器来允许操作。为了从按键返回引脚切换到端口引脚, 应该先通过 KRM 寄存器禁止操作, 然后再设置端口引脚。

第 24 章 待机功能

24.1 概述

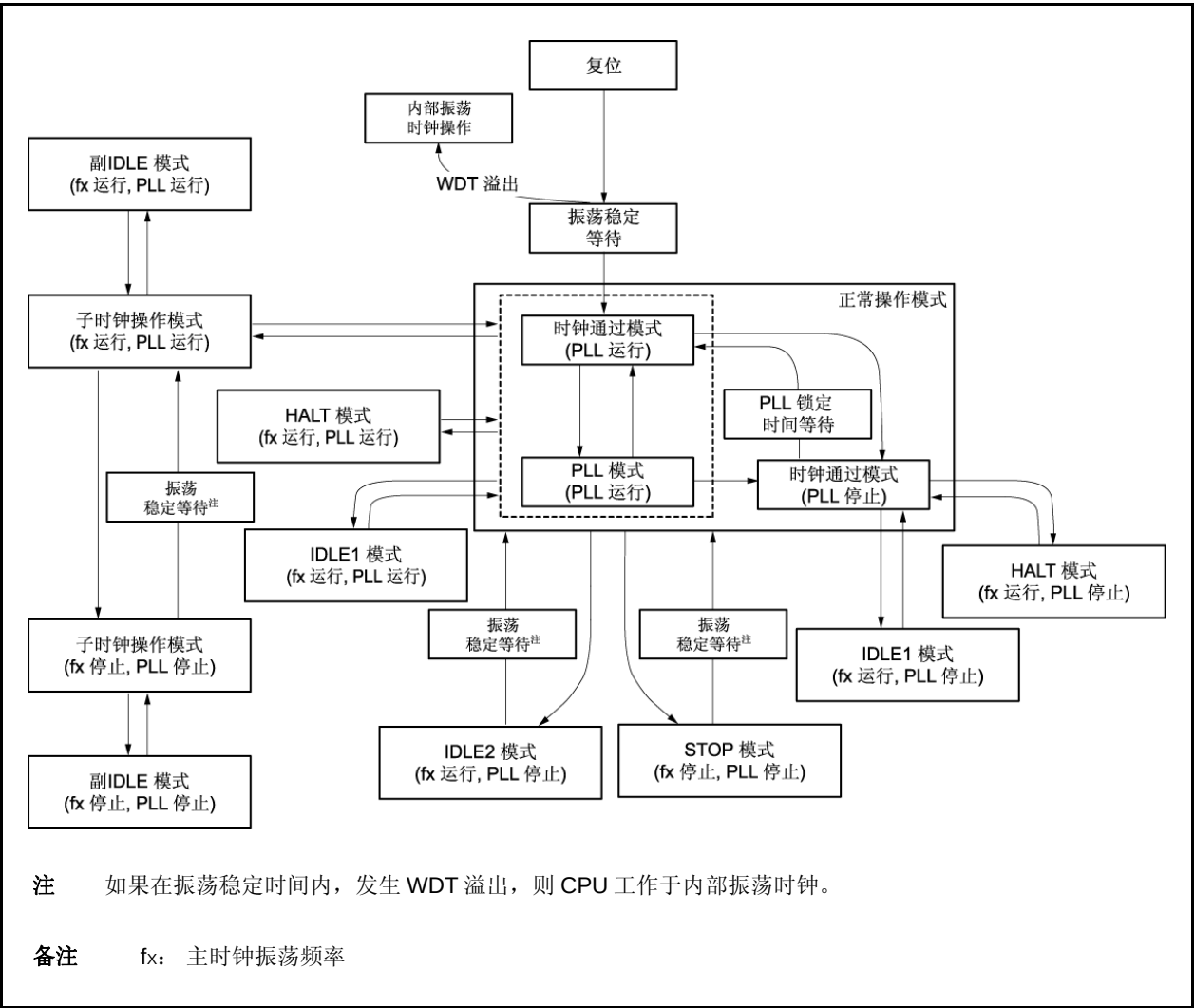
通过组合使用待机模式，并选择合适的应用模式，可以有效的降低系统功耗。可用待机模式列于表 24-1。

表 24-1. 待机模式

模式	功能概述
HALT 模式	该模式中只有 CPU 的操作时钟被停止。
IDLE1 模式	该模式中除了振荡器，PLL [✱] 以及 flash 存储器外，内部电路的其他所有操作都被停止。
IDLE2 模式	该模式中除了振荡器外，所有的内部电路操作都被停止。
STOP 模式	该模式中除了副时钟振荡器外，所有的内部电路操作都被停止。
副时钟操作模式	该模式中的副时钟被用作内部系统时钟
副-IDLE 模式	该模式是在副时钟操作模式中，除了振荡器外，所有的内部电路操作都被停止。

注 **PLL** 保持先前的操作状态。

图 24-1. 状态转换



24.2 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。此寄存器是特殊寄存器，只能使用特殊顺序写入（参见 3.4.8 特殊寄存器）。

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。

复位输入将把该寄存器置为 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	产生INTWDT2信号时待机模式的释放控制
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引脚控制待命模式的释放
0	允许通过NMI引脚输入来释放待机模式
1	禁止通过NMI引脚输入来释放待机模式

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来释放待机模式
1	禁止通过可屏蔽中断请求信号来释放待机模式

STP	待机模式 ^注 设置
0	普通模式
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 HALT 模式被释放时, 对 NMI1M, NMI0M 和 INTM 位的设置无效。

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被挂起, 则将中断请求信号 (NMI1M, NMI0M 或 INTM) 对应的位设为 1, 然后将 STP 位设为 1。

<R>

(2) 节电模式寄存器 (PSMR)

PSMR 寄存器是一个 8 位寄存器，它控制省电模式中的操作状态和时钟操作。
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。
复位输入将把该寄存器置为 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规范
0	0	IDLE1, 副IDLE 模式
0	1	STOP 模式
1	0	IDLE2, 副IDLE 模式
1	1	STOP 模式

- 注意事项
- 1. 确保将位 2 至位 7 清除为“0”。
 - 2. 仅当 PSC.STP 位为 1 时，PSM0 和 PSM1 位有效。

- 备注
- IDLE1: 在此模式中，除了振荡器操作和其它一些电路（flash 存储器和 PLL）外，所有操作都被停止。
在 IDLE1 模式被释放后，可以恢复正常操作模式，而无需确保振荡稳定时间，和 HALT 模式一样。
 - IDLE2: 在此模式中，除了振荡器操作之外，所有操作都被停止。
在 IDLE2 模式被释放后，等待 OSTS 寄存器指定的振荡稳定时间后，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，所有操作都被停止。
在 STOP 模式被释放后，等待 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，所有其它操作都被暂停。在中断请求信号释放 IDLE 模式之后，等待 12 个副时钟周期，然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 寄存器控制等待时间，STOP 模式释放后直到振荡稳定的等待时间，或 IDLE2 模式释放后直到片上 flash 存储器稳定的等待时间。

OSTS 寄存器可以以 8 位为单位进行读取或写入。

复位输入将把该寄存器设置为 06H。

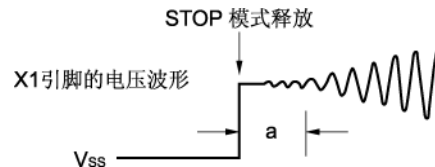
复位后: 06H R/W 地址: FFFFF6C0H

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	振荡稳定时间/设置时间 ^注 的选择	fx	
				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 分别为 STOP 模式释放后要求的振荡稳定时间，和 IDLE2 模式释放后要求的设置时间。

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的这段时间（即下图中“a”所示的部分），不管 STOP 模式是通过复位输入或产生中断请求信号释放。



2. 确保将第 3~7 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 fx: 主时钟振荡频率

24.3 HALT模式

24.3.1 设置和操作状态

当在正常操作模式中执行专用指令（HALT）时，设置 HALT 模式。

在 HALT 模式中，时钟振荡器继续操作。仅停止对 CPU 的时钟供应，提供给其它片上周边设备功能的时钟继续供应。

结果，程序停止执行，并且内部 RAM 保持设置 HALT 模式之前的内容。独立于 CPU 指令处理的片上周边设备功能可继续操作。

表 24-3 显示了 HALT 模式中的操作状态。

通过 HALT 模式和正常操作模式的组合轮换操作，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. 如果在一个未屏蔽中断请求信号被挂起时执行 HALT 指令，则此状态被转移到 HALT 模式中，但因为这个等待响应的中断请求，HALT 模式会立即释放。

24.3.2 释放 HALT 模式

HALT 模式的释放可以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），HALT 模式中来自可操作周边功能中未屏蔽的内部中断请求信号，或复位信号（ $\overline{\text{RESET}}$ 引脚输入，看门狗定时器溢出所生成的复位信号（WDT2RES），低电压检测器（LVI）[※] 生成的复位信号（LVIRE）或时钟监视器（CLM）产生的复位信号（CLMRES））。

在 HALT 模式释放后，恢复正常操作模式。

注 仅限 V850ES/SG2

(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，而和中断请求信号的优先级无关。然而，如果在某个中断服务程序中设置 HALT 模式，则之后发出的中断请求信号按如下所述来响应。

- (a) 如果发出的中断请求信号比当前正在服务的中断请求的优先级更低，则释放 HALT 模式，但是不响应此中断请求信号。此中断请求信号仍被保持。
- (b) 如果发出的中断请求信号比当前正在服务的中断请求的优先级更低高（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 24-2. 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	执行跳转，到服务处理地址。	
可屏蔽中断请求信号	跳转到服务处理地址或执行下一条指令。	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 24-3. 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	
DMA		可操作	
中断控制器		可操作	
ROM 修正		停止操作	
定时器 P (TMP0 至 TMP5)		可操作	
定时器 Q (TMQ0)		可操作	
定时器 M (TMM0)		当 f_{XT} 之外的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) 被选择作为计数时钟时，可操作	可操作
看门狗定时器 2		当 f_{XT} 之外的时钟被选择作为计数时钟时，可操作	可操作
串行接口	CSIB0 至 CSIB4	可操作	
	I ² C00 至 I ² C02	可操作	
	UARTA0 至 UARTA2	可操作	
CAN 控制器		可操作	
IEBus 控制器		可操作	
A/D 转换器		可操作	
D/A 转换器		可操作	
实时输出功能 (RTO)		可操作	
按键中断功能 (KR)		可操作	
CRC 运算电路		可操作 (数据没有被输入到 CRCIN 以停止 CPU 的状态)	
外部总线接口		参见 2.2 引脚状态。	
端口功能		保持设置 HALT 模式之前的状态	
内部数据		CPU 寄存器，状态字，数据和全部其它的内部数据，例如内部 RAM 的内容，被保持为设置 HALT 模式之前的状态。	

24.4 IDLE1 模式

24.4.1 设置和操作状态

在正常操作模式中对 PSMR.PSM1 和 PSMR.PSM0 位清零，将 PSC.STP 位设为 1，可以设置 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 和其它片上周边设备功能的时钟供应。

结果，停止程序执行，并且保留设置 IDLE1 模式之前的内部 RAM 内容。CPU 和其它片上周边设备功能停止操作。然而，可使用副时钟或外部时钟操作的片上周边设备功能继续操作。

表 24-5 显示了在 IDLE1 模式中的操作状态。

因为 IDLE1 模式停止了片上周边设备功能的操作，所以 IDLE1 模式比 HALT 模式降低更多功耗。因为不停止主时钟振荡器，所以当用释放 HALT 模式的方法去释放 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. 在执行存入 PSC 寄存器设置 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挂起，则 IDLE1 模式立即被此挂起的中断请求释放。

24.4.2 释放IDLE1 模式

IDLE1 模式的释放可以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），IDLE1 模式中来自可操作周边功能中未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，看门狗定时器溢出所生成的复位信号（WDT2RES），低电压检测器（LVI）^注生成的复位信号（LVIRE）或时钟监视器（CLM）产生的复位信号（CLMRES））。

在 IDLE1 模式释放后，恢复到正常操作模式。

<R> 注 仅限 V850ES/SG2

（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，而和中断请求信号的优先级无关。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响应。

（a）如果发出的中断请求信号比当前正在服务的中断请求的优先级更低，则释放 IDLE1 模式，但是不响应此中断请求信号。此中断请求信号仍被保持。

（b）如果发出的中断请求信号比当前正在服务的中断请求的优先级更高（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。

注意事项 通过设置 PSC.NMI1M，PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号无效，IDLE1 模式也不会被释放。

表 24-4. 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	执行跳转到服务处理地址。	
可屏蔽中断请求信号	跳转到服务处理地址或执行下一条指令。	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 24-5. IDLE1 模式中的操作状态

IDLE1 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		可操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是允许释放待机模式）	
ROM 修正		停止操作	
定时器 P（TMP0 至 TMP5）		停止操作	
定时器 Q（TMQ0）		停止操作	
定时器 M（TMM0）		当选择 $f_R/8$ 作为计数时钟时，可操作	当选择 $f_R/8$ 或 f_{XT} 作为计数时钟时，可操作
钟表定时器		当 f_x （BRG 分频）被选择作为计数时钟时，可操作	可操作
看门狗定时器 2		当选择 f_R 作为计数时钟时，可操作	当选择 f_R 或 f_{XT} 作为计数时钟时，可操作
串行接口	CSIB0 至 CSIB4	当选择 $\overline{SCKBn}$ 输入时钟作为计数时钟时，可操作（ $n = 0$ 至 4）	
	I ² C00 至 I ² C02	停止操作	
	UARTA0 至 UARTA2	停止操作（但是当选择 ASCKA0 输入时钟时，UARTA0 可操作）	
CAN 控制器		停止操作	
IEBus 控制器		停止操作	
A/D 转换器		保持操作（保持转换结果） [※]	
D/A 转换器		保持操作（输出保持 [※] ）	
实时输出功能（RTO）		停止操作（输出保持）	
按键中断功能（KR）		可操作	
CRC 运算电路		停止操作	
外部总线接口		参见 2.2 引脚状态。	
端口功能		保持设置 IDLE1 模式之前的状态	
内部数据		CPU 寄存器，状态字，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，在转换到 IDLE1 模式之前，停止 A/D 转换器和 D/A 转换器。

24.5 IDLE2 模式

24.5.1 设置和操作状态

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，可以设置 IDLE2 模式。

在 IDLE2 模式中，时钟振荡器继续操作，但停止 CPU，PLL，flash 存储器和其它片上周边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 内容。CPU，PLL 和其它片上周边设备功能停止操作。然而，可使用副时钟或外部时钟操作的片上周边设备功能继续操作。

表 24-7 显示在 IDLE2 模式中的操作状态。

因为 IDLE2 模式停止了片上周边设备功能，PLL 和 flash 存储器的操作，所以 IDLE2 模式比 IDLE1 模式能降低更多功耗。然而，因为 PLL 和 flash 存储器被停止，所以当释放 IDLE2 模式时，PLL 和 flash 存储器要求有一个设置时间。

- 注意事项**
1. 在执行写入 PSC 寄存器来设置 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挂起，则 IDLE2 模式立即被此等待响应的中断请求释放。

24.5.2 释放 IDLE2 模式

IDLE2 模式的释放可以通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号），未屏蔽的外部中断请求信号（INTP0~INTP7 引脚输入），IDLE2 模式下可操作周边功能中未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，看门狗定时器溢出所生成的复位信号（WDT2RES），低电压检测器（LVI）[※]生成的复位信号（LVIRES）或时钟监视器（CLM）产生的复位信号（CLMRES））。

在 IDLE2 模式释放后，恢复到正常操作模式。

<R> 注 仅限 V850ES/SG2

（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，而和中断请求信号的优先级无关。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响应。

- （a）如果发出的中断请求信号比当前正在服务的中断请求的优先级更低，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）如果发出的中断请求信号比当前正在服务的中断请求的优先级更高（包括不可屏蔽中断请求信号），则释放 IDLE2 模式，并且响应此中断请求信号。

注意事项 通过设置 PSC.NMI1M，PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号无效，IDLE2 模式也不会被释放。

表 24-6. 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，执行跳转到服务处理地址。	
可屏蔽中断请求信号	在保证规定的设置时间之后，跳转到服务处理地址或执行下一条指令。	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 24-7. IDLE2 模式中的操作状态

IDLE2 模式设置		操作状态	
		当不使用副时钟时	当使用副时钟时
主时钟振荡器		允许振荡	
副时钟振荡器		—	允许振荡
内部振荡器		允许振荡	
PLL		停止操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是可以释放待机模式）	
ROM 修正		停止操作	
定时器 P（TMP0 至 TMP5）		停止操作	
定时器 Q（TMP0）		停止操作	
定时器 M（TMM0）		当选择 $f_{R/8}$ 作为计数时钟时，可操作	当选择 $f_{R/8}$ 或 f_{XT} 作为计数时钟时，可操作
钟表定时器		当 f_x （BRG 分频）被选择作为计数时钟时，可操作	可操作
看门狗定时器 2		当选择 f_R 作为计数时钟时，可操作	当选择 f_R 或 f_{XT} 作为计数时钟时，可操作
串行接口	CSIB0 至 CSIB4	当选择 $\overline{SCKBn}$ 输入时钟作为计数时钟时，可操作（ $n = 0$ 至 4）	
	I ² C00 至 I ² C02	停止操作	
	UARTA0 至 UARTA2	停止操作（但是当选择 ASCKA0 输入时钟时，UARTA0 可操作）	
CAN 控制器		停止操作	
IEBus 控制器		停止操作	
A/D 转换器		保持操作（保持转换结果） [※]	
D/A 转换器		保持操作（输出保持 [※] ）	
实时输出功能（RTO）		停止操作（输出保持）	
按键中断功能（KR）		可操作	
CRC 运算电路		停止操作	
外部总线接口		参见 2.2 引脚状态。	
端口功能		保持设置 IDLE2 模式之前的状态	
内部数据		CPU 寄存器，状态字，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，在转换到 IDLE2 模式之前，停止 A/D 转换器和 D/A 转换器。

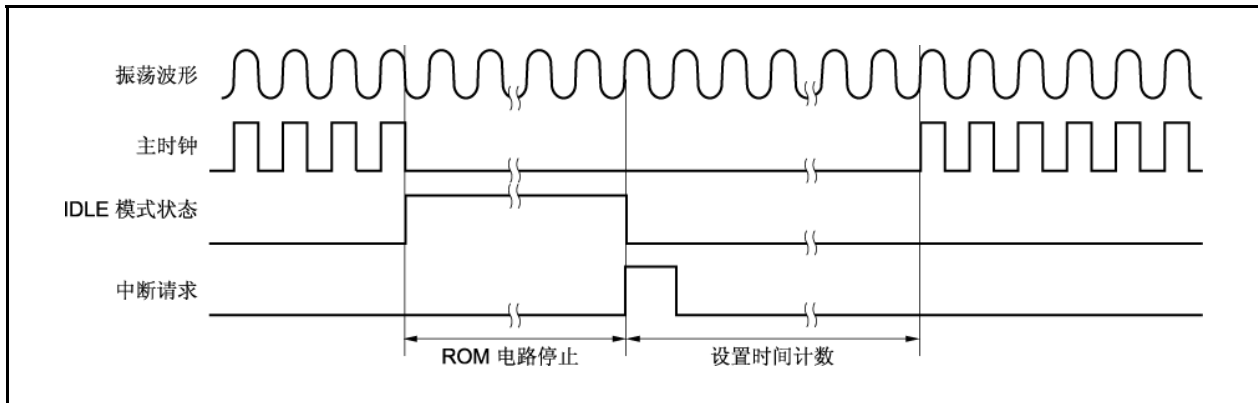
24.5.3 当释放IDLE2 模式时保证设置时间

在释放 IDLE2 模式之后，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，所有模块的操作都被停止。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号释放 IDLE2 模式

通过设置 OSTS 寄存器，保留指定的设置时间。

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（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， $2^{16}/f_x$ 。

24.6 STOP 模式

24.6.1 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。CPU 和片上周边设备功能的时钟停止供应。

结果，停止程序执行，并且保持设置 STOP 模式之前的内部 RAM 内容。使用副时钟振荡器或外部时钟的片上周边设备功能继续操作。

表 24-9 显示在 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 模式能降低更多功耗。如果不使用副时钟振荡器，内部振荡器和外部时钟，则功耗最低可以达到仅有漏电流。

注意事项 1. 在执行写入 PSC 寄存器来设置 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挂起，则 STOP 模式立即被此挂起的中断请求释放。

24.6.2 释放 STOP 模式

STOP 模式的释放可以通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号），未屏蔽的外部中断请求信号（INTP0~INTP7 引脚输入），STOP 模式下可操作外围功能中未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，看门狗定时器溢出生成的复位信号（WDT2RES），低电压检测器（LVI）[※]生成的复位信号（LVIRE））。

在 STOP 模式释放后，在保留的振荡稳定时间之后，恢复正常操作模式。

注 仅限 V850ES/SG2

（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，而和中断请求信号的优先级无关。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响应。

- （a）如果发出的中断请求信号比当前正在服务的中断请求的优先级更低，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）如果发出的中断请求信号比当前正在服务的中断请求的优先级更高（包括不可屏蔽中断请求信号），则释放 STOP 模式，并且响应此中断请求信号。

注意事项 通过设置 PSC.NMI1M，PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号无效，STOP 模式也不会被释放。

表 24-8. 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，跳转到服务处理地址。	
可屏蔽中断请求信号	在保证振荡稳定时间之后，跳转到服务处理地址或执行下一条指令。	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 24-9. STOP 模式中的操作状态

项目		STOP 模式设置		操作状态	
				当不使用副时钟时	当使用副时钟时
主时钟振荡器				停止振荡	
副时钟振荡器				—	允许振荡
内部振荡器				允许振荡	
PLL				停止操作	
CPU				停止操作	
DMA				停止操作	
中断控制器				停止操作（但是可以释放待机模式）	
ROM 修正				停止操作	
定时器 P（TMP0 至 TMP5）				停止操作	
定时器 Q（TMP0）				停止操作	
定时器 M（TMM0）				当选择 $f_{R/8}$ 作为计数时钟时，可操作	当选择 $f_{R/8}$ 或 f_{XT} 作为计数时钟时，可操作
钟表定时器				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	当选择 f_R 或 f_{XT} 作为计数时钟时，可操作
串行接口	CSIB0 至 CSIB4			当选择 $\overline{SCKBn}$ 输入时钟作为计数时钟时，可操作（ $n = 0$ 至 4）	
	I ² C00 至 I ² C02			停止操作	
	UARTA0 至 UARTA2			停止操作（但是当选择 ASCKA0 输入时钟时，UARTA0 可操作）	
CAN 控制器				停止操作	
IEBus 控制器				停止操作	
A/D 转换器				停止操作（转换结果不确定） ^{註 1, 2}	
D/A 转换器				停止操作 ^{註 3, 4} （输出高阻抗）	
实时输出功能（RTO）				停止操作（输出保持）	
按键中断功能（KR）				可操作	
CRC 运算电路				停止操作	
外部总线接口				参见 2.2 引脚状态。	
端口功能				保持设置 STOP 模式之前的状态	
内部数据				CPU 寄存器，状态字，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. 如果在 A/D 转换器操作时设置 STOP 模式，则 A/D 转换器自动停止操作，并且在 STOP 模式释放后再次启动操作。然而，在这种情况下，STOP 模式释放后的 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，功耗也和 A/D 转换器停止时设置 STOP 模式一样。
 3. 如果在 D/A 转换器操作时设置 STOP 模式，则 D/A 转换器自动停止操作且引脚状态将变为高阻抗。在 STOP 模式被释放后，会重新开始 D/A 转换，在设置时间过后，引脚状态会变为设置 STOP 模式之前的输出电平。
 4. 即使在操作 D/A 转换器时设置 STOP 模式，功耗也和 D/A 转换器停止时设置 STOP 模式一样。

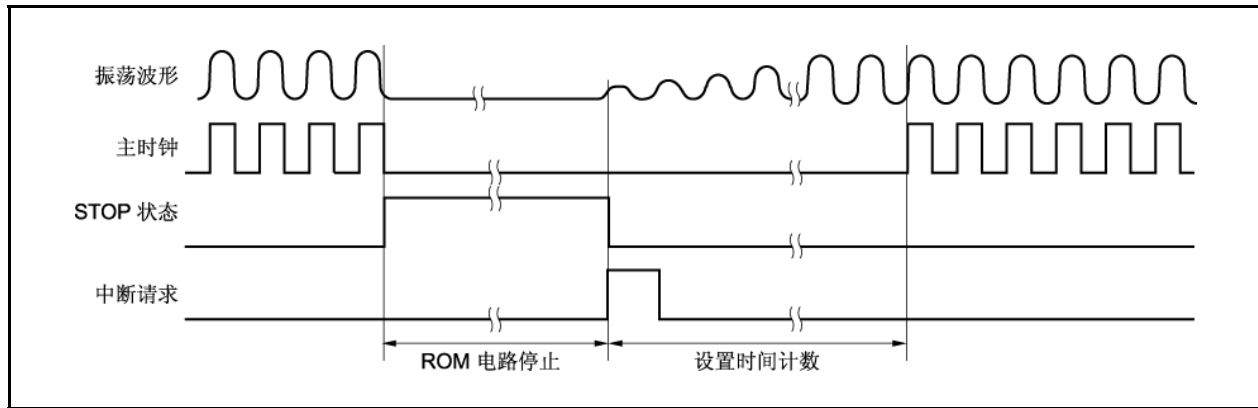
24.6.3 当释放STOP模式时保证振荡稳定时间

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，主时钟振荡器操作会停止。

(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，保留振荡稳定时间。

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， $2^{16}/f_x$ 。

24.7 副时钟操作模式

24.7.1 设置和操作状态

在正常操作模式中设置 PCC.CK3 位为 1，可以设置副时钟操作模式。

当设置副时钟操作模式时，内部系统时钟从主时钟转换为副时钟。通过 PCC.CLS 位检测时钟是否已被切换。

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 24-10 显示副时钟操作模式的状态。

- 注意事项
- 1. 当操作 CK3 位时，不要改变 PCC.CK2 ~ PCC.CK0 位（推荐使用位操作指令来操作）的设置值。对于 PCC 寄存器的详细描述，参见 6.3（1）处理器时钟控制寄存器（PCC）。
 - 2. 如果以下的条件不满足，改变 CK2~CK0 位的设置，以便满足条件并且设置副时钟操作模式。

<R> 内部主时钟（f_{CLK}） > 副时钟（f_{XT} = 32.768 kHz） × 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。

24.7.2 释放副时钟操作模式

副时钟操作模式可通过复位信号释放（RESET 引脚输入，看门狗定时器溢出所生成的复位信号（WDT2RES），低电压检测器（LVI）[※]生成的复位信号（LVIRE_S）或时钟监视器（CLM）产生的复位信号（CLMRES））。

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。

<R> 注 仅限 V850ES/SG2

- 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 寄存器的详细描述，参见 6.3（1）处理器时钟控制寄存器（PCC）。

表 24-10. 副时钟操作模式中的操作状态

副时钟操作模式设置		操作状态		
		当主时钟振荡时	当主时钟停止时	
<R>	项目			
	副时钟振荡器		允许振荡	
	内部振荡器		允许振荡	
	PLL		可操作	停止操作 ^注
	CPU		可操作	
	DMA		可操作	
	中断控制器		可操作	
ROM 修正		可操作		
<R>	定时器 P（TMP0 至 TMP5）		可操作	停止操作 ^注
<R>	定时器 Q（TMP0）		可操作	停止操作 ^注
<R>	定时器 M（TMM0）		可操作	当选择 f _R /8 或 f _{XT} 作为计数时钟时，可操作
	钟表定时器		可操作	当选择 f _{XT} 作为计数时钟时，可操作
	看门狗定时器 2		可操作	当选择 f _R 或 f _{XT} 作为计数时钟时，可操作
	串行接口	CSIB0 至 CSIB4	可操作	当选择 SCKB _n 输入时钟作为计数时钟时，可操作（n = 0 至 4）
		I ² C00 至 I ² C02	可操作	停止操作 ^注
		UARTA0 至 UARTA2	可操作	停止操作（但是当选择 ASCKA0 输入时钟时，UARTA0 可操作）
	<R>	CAN 控制器		可操作
<R>	IEBus 控制器		可操作	停止操作 ^注
<R>	A/D 转换器		可操作	停止操作 ^注
<R>	D/A 转换器		可操作	
	实时输出功能（RTO）		可操作	停止操作 ^注 （输出保持）
	按键中断功能（KR）		可操作	
<R>	CRC 运算电路		可操作	
	外部总线接口		可操作	
	端口功能		可设置	
内部数据		可设置		

<R> 注 要停止主时钟，则一定要停止 PLL（PLLCTL.PLLON 位 = 0）。
运行于主时钟的内部外围功能也要停止。

注意事项 当 CPU 工作在副时钟并且主时钟振荡停止时，无法访问需要产生等待的寄存器。如果产生等待，则其只能通过复位来释放（参见 3.4.9（2））。

24.8 副IDLE模式

24.8.1 设置和操作状态

在副时钟操作模式中设置 PSMR.PSM1 和 PSMR.PSM0 位为 00 划 10，并设置 PSC.STP 位为 1，可以设置副 IDLE 模式。

在此模式中，时钟振荡器继续操作，CPU、flash 存储器和其它片上周边设备功能的时钟供应停止。

结果，停止程序执行，并且保持设置副 IDLE 模式之前的内部 RAM 内容。停止 CPU 和其它片上周边设备功能。然而，使用副时钟或外部时钟操作的片上周边设备功能可以继续操作。

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能降低更多功耗。如果在主时钟停止后设置副 IDLE 模式，当前功耗可以降低至 STOP 模式的等级。

表 24-12 显示副 IDLE 模式中的操作状态。

- 注意事项 1.** 将指令保存到 PSC 寄存器来设置副 IDLE 模式之后，插入五个或更多的 NOP 指令。
- 2.** 如果设置副 IDLE 模式时有一个未被屏蔽的中断请求信号被挂起，则副 IDLE 模式立即被此挂起的中断请求释放。

24.8.2 释放副IDLE模式

副 IDLE 模式的释放可以通过不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号），未屏蔽的外部中断请求信号（INTP0~INTP7 引脚输入），副 IDLE 模式下可操作周边功能中未屏蔽的内部中断请求信号，或复位信号（RESET 引脚输入，看门狗定时器溢出所生成的复位信号（WDT2RES），低电压检测器（LVI）^註生成的复位信号（LVIRESE）或时钟监视器（CLM）产生的复位信号（CLMRES））。PLL 返回到设置副 IDLE 模式之前的操作状态。

当通过中断请求信号释放副 IDLE 模式时，设置为副时钟操作模式。

<R> 注 仅限 V850ES/SG2

（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，而和中断请求信号的优先级无关。

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响应。

- （a）如果发出的中断请求信号比当前正在服务的中断请求的优先级更低，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）如果发出的中断请求信号比当前正在服务的中断请求的优先级更高（包括不可屏蔽中断请求信号），则释放副 IDLE 模式，并且响应此中断请求信号。

- 注意事项**
1. 通过设置 PSC.NMI1M，PSC.NMI0M 和 PSC.INTM 位为 1 来禁止中断请求信号，这样中断请求信号无效，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 模式被释放之间需要等待 12 个副时钟周期（大约 366 μ s）。

表 24-11. 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	执行跳转到服务处理地址。	
可屏蔽中断请求信号	跳转到服务处理地址或执行下一条指令。	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 24-12. 在副 IDLE 模式中的操作状态

副 IDLE 模式设置		操作状态	
		当主时钟振荡时	当主时钟停止时
副时钟振荡器		允许振荡	
内部振荡器		允许振荡	
PLL		可操作	停止操作 ^{註1}
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是可以释放待机模式）	
ROM 修正		停止操作	
定时器 P（TMP0 至 TMP5）		停止操作	
定时器 Q（TMP0）		停止操作	
定时器 M（TMM0）		当选择 $f_{R/8}$ 或 f_{XT} 作为计数时钟时，可操作	
钟表定时器		停止操作	当选择 f_{XT} 作为计数时钟时，可操作
看门狗定时器 2		当选择 f_R 或 f_{XT} 作为计数时钟时，可操作	
串行接口	CSIB0 至 CSIB4	当选择 $\overline{SCKBn}$ 输入时钟作为计数时钟时，可操作（ $n = 0$ 至 4）	
	I ² C00 至 I ² C02	停止操作	
	UARTA0 至 UARTA2	停止操作（但是当选择 ASCKA0 输入时钟时，UARTA0 可操作）	
CAN 控制器		停止操作	
IEBus 控制器		停止操作	
A/D 转换器		保持操作（保持转换结果） ^{註2}	
D/A 转换器		保持操作（输出保持 ^{註2} ）	
实时输出功能（RTO）		停止操作（输出保持）	
按键中断功能（KR）		可操作	
CRC 运算电路		停止操作	
外部总线接口		参见 2.2 引脚状态（与 IDLE1，IDLE2 模式的运行状态相同）	
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，状态字，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，确认停止 PLL（PLLCTL.PLLON 位 = 0）。
 2. 为了实现低功耗，在转换到副 IDLE 模式之前，停止 A/D 和 D/A 转换器。

25.1 概述

可用的复位功能如下。

(1) 复位源

- 通过 $\overline{\text{RESET}}$ 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 (LVI) 供电电压和检测电压 (LVIRE[※]) 的对比而引起的系统复位
- 通过检测到时钟监控器 (CLM) 振荡停止引起的系统复位 (CLMRES)

<R> 注 V850ES/SG2

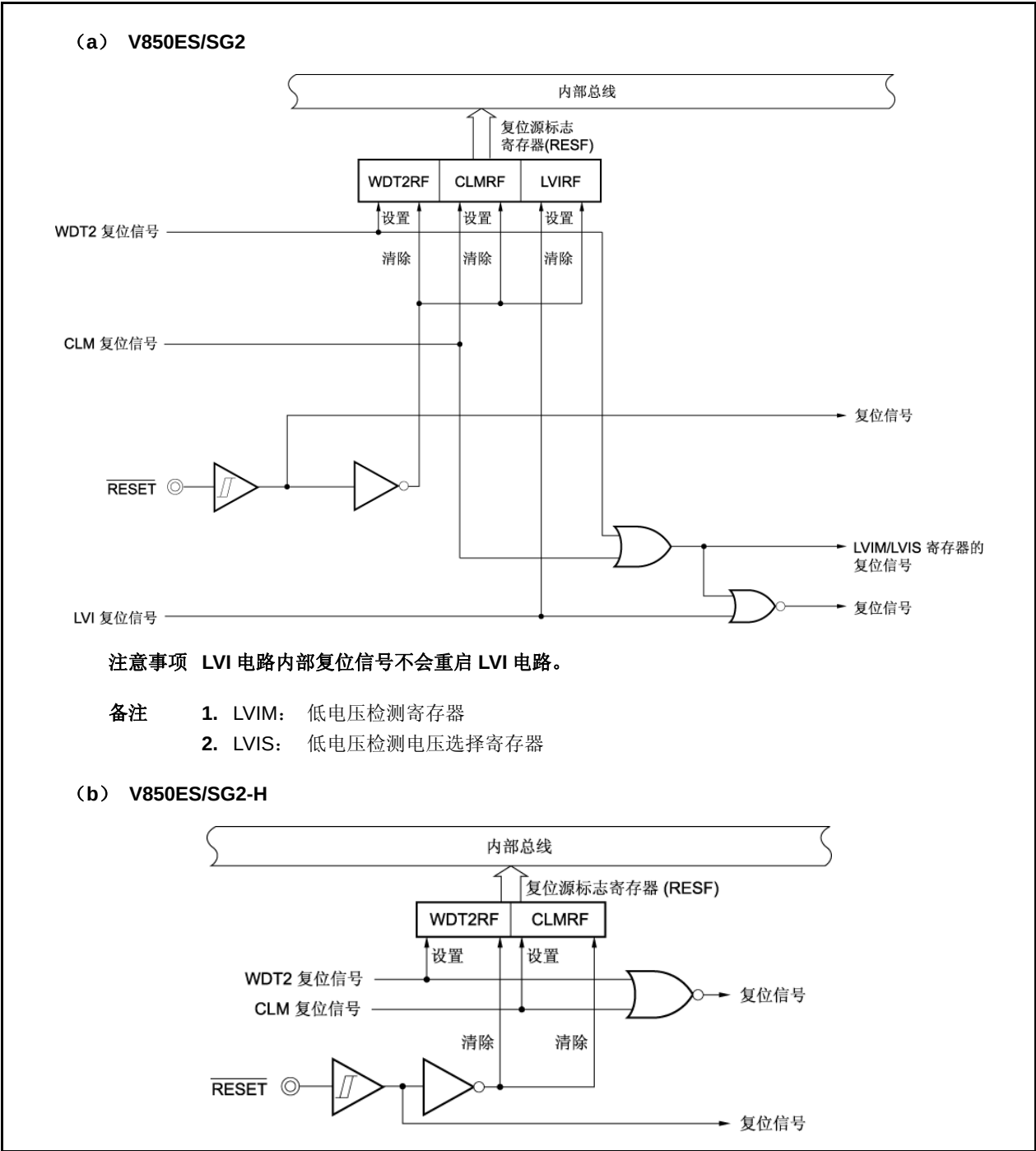
当复位信号释放后，可以通过复位源标志寄存器 (RESF) 来确认复位源。

(2) 紧急操作模式

<R> 如果复位释放后或 STOP 模式释放后，在插入的主时钟振荡稳定时间内 WDT2 溢出，被认为是主时钟振荡异常，并且 CPU 开始在内部振荡时钟下操作。

注意事项 在内部振荡时钟下操作 CPU 时，禁止访问有等待状态的寄存器。关于寄存器产生等待状态的详细信息，请参见 3.4.9 (2) 访问特殊片上外围 I/O 寄存器。

图 25-1. 复位功能的结构框图



25.2 检测复位源的寄存器

<R> V850ES/SG2 具有四种类型的复位源，而 V850ES/SG2-H 则具有三种类型的复位源。当复位释放后，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，只能够按照特殊顺序来写入（参见 3.4.8 特殊寄存器）。

RESF 寄存器表明了产生复位信号的来源。

该寄存器可以以 8 位或 1 位为单位来读取或写入。

RESET 引脚将把该寄存器置为 00H。如果复位源并非来自 RESET 引脚信号，则缺省值会有所不同。

<R>

复位后： 00H ^{注1}		R/W	地址： FFFF888H					
RESF	7	6	5	4	3	2	1	0
	0	0	0	WDT2RF	0	0	CLMRF	LVIRF ^{注2}
WDT2RF		WDT2中的复位信号						
0		未生成						
1		生成						
CLMRF		CLM中的复位信号						
0		未生成						
1		生成						
LVIRF ^{注2}		LVI中的复位信号						
0		未生成						
1		生成						

<R>

- 注 1. 当通过 RESET 引脚执行复位时，RESF 寄存器的值将清为 00H。通过看门狗定时器 2（WDT2），时钟监视器（CLM），或低压检测器（LVI）来执行复位时，该寄存器的对应复位标志（WDT2RF，CLMRF，或 LVIRF 位）将会进行设置；而其它源则会保持不变。
2. LVIRF 仅在 V850ES/SG2 中可用。
在 V850ES/SG2-H 中一定要将第 0 位设为“0”。

<R>

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。

25.3 操作

25.3.1 通过RESET引脚的复位操作

当 RESET 引脚输入一个低电平时，系统将被复位，且所有硬件模块将被初始化。
当 RESET 引脚的电平由低变高时，复位状态将被释放。

表 25-1. 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 (fx)	停止振荡	开始振荡
副时钟振荡器 (fxT)	继续振荡	
内部振荡器	停止振荡	开始振荡
外围时钟 (fxx 至 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 (fCLK)， 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	确保安全振荡稳定时间之后，才开始执行程序
看门狗定时器 2	停止操作（初始化到 0）	将内部振荡时钟作为源时钟，从 0 开始计数
内置 RAM	如果上电复位和复位输入冲突，或 CPU 访问和复位输入冲突（数据被破坏），则不确定。否则，保留复位输入后的即时数值 ^{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）。	
其它片上外围功能	停止操作	安全振荡稳定时间之后可以开始操作

注 1. V850ES/SG2 和 V850ES/SG2-H 中的固件会在内部系统复位操作释放后需要使用一部分内部 RAM，这是因为它支持引导交换功能。因此，在上电复位时一些 RAM 区域中的内容将不会被保留。详细信息请参见 25.3.5 复位释放后的操作。

2. 电源打开时，以下引脚可能会瞬时输出一个未定义的电平。
- P10/ANO0 引脚
 - P11/ANO1 引脚
 - P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 引脚（DDO 引脚仅在 flash 存储器版本中提供）

注意事项 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 的位被清除前，一个高电平在复位释放后被输入到 P05/DRST 引脚中，那么 V850ES/SG2 可能会进入片上调试模式（仅限 flash 存储器版本）。Mask ROM 不支持片上调试模式；而 OCDM 寄存器控制连接到 P05/INTP2 引脚的下拉电阻器。详细信息请参见 第 4 章 端口功能。

图 25-2. RESET 引脚输入引起的复位操作的时序

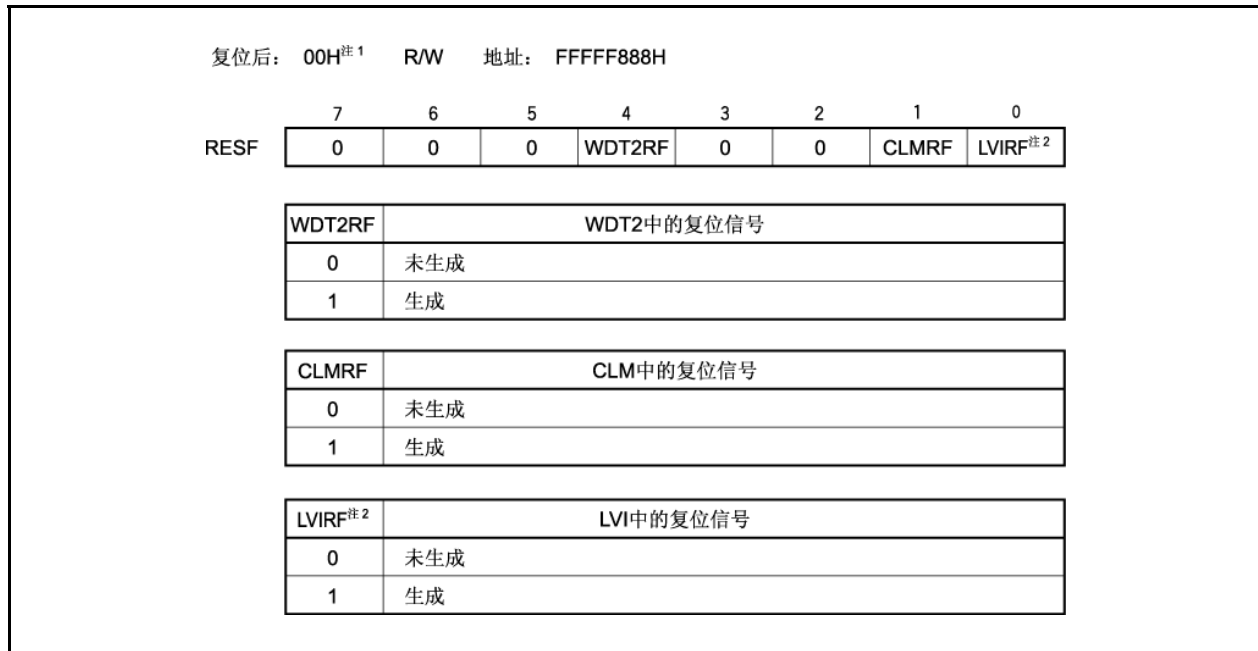
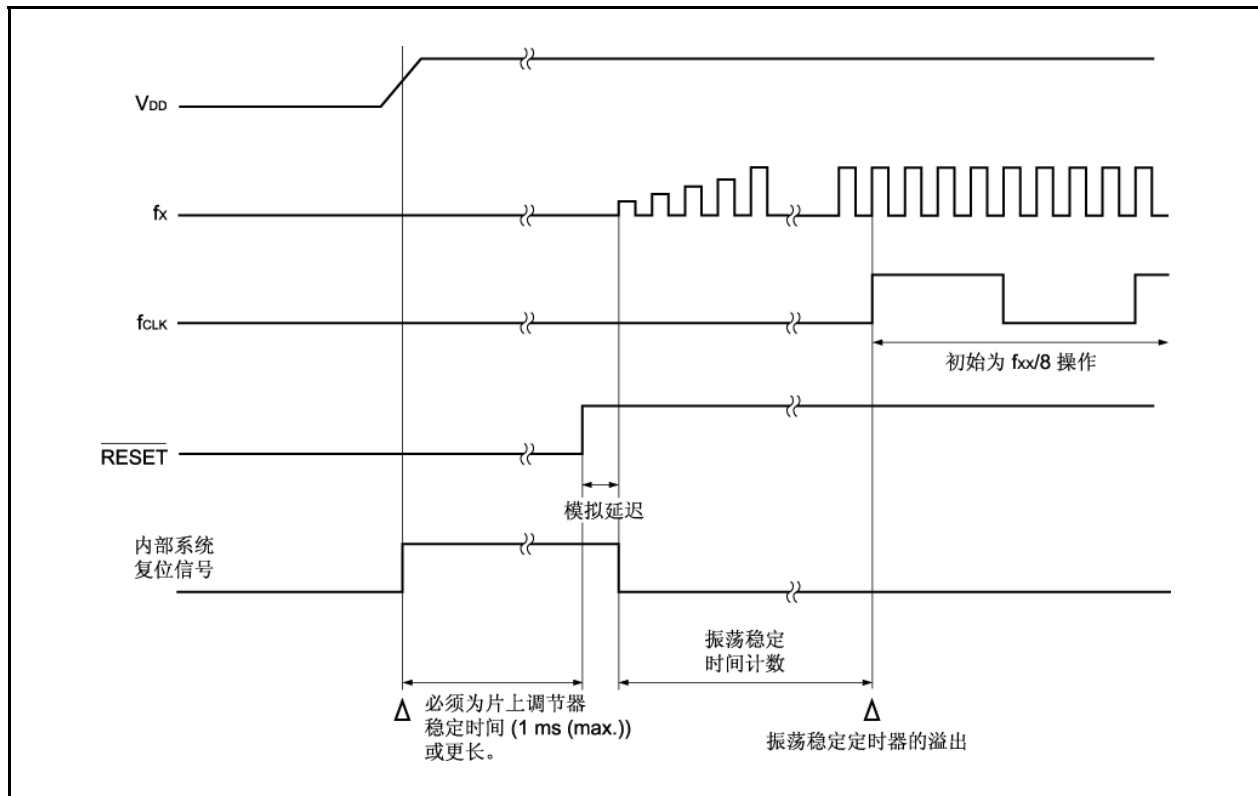


图 25-3. 上电复位操作的时序



25.3.2 通过看门狗定时器 2 执行的复位操作（WDT2RES）

当看门狗定时器 2 由于溢出被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，且对硬件进行初始化，恢复到初始状态。

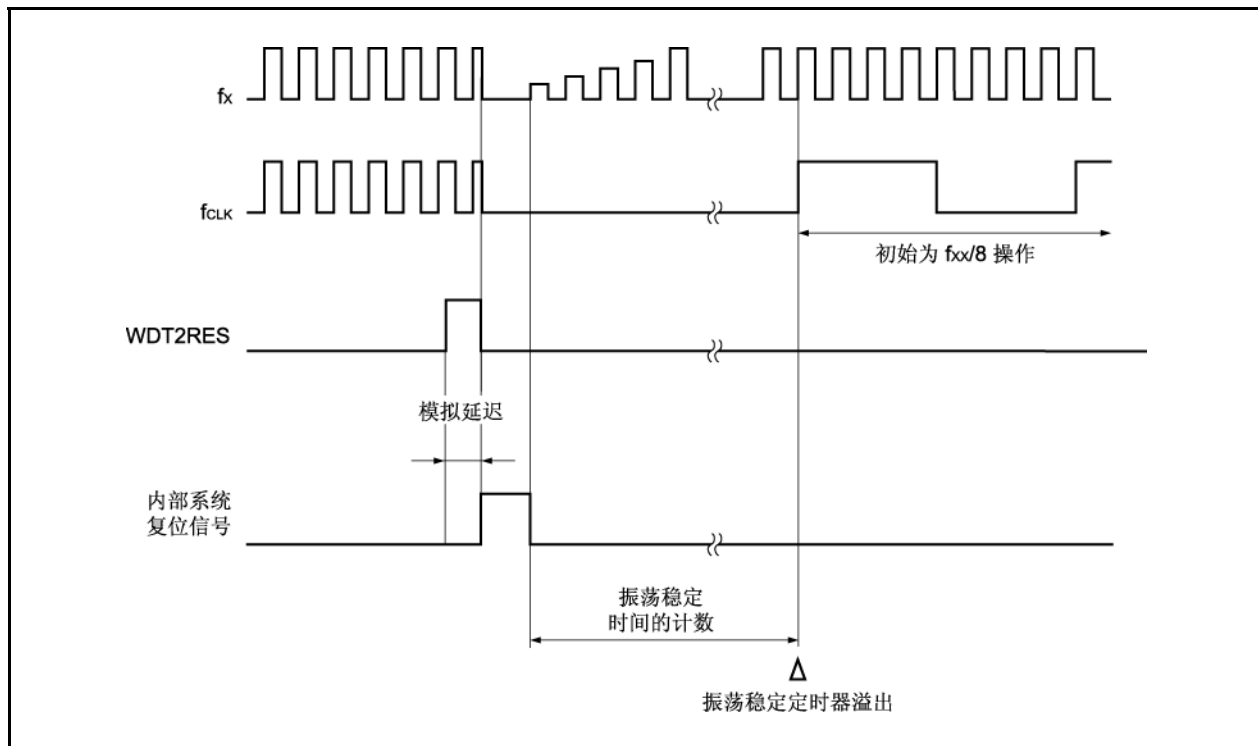
在看门狗定时器 2 溢出后，会进入复位状态并且持续到预定时间（模拟延时），然后复位状态将会自动释放。在复位期间主时钟振荡停止。

表 25-2. 看门狗定时器 2 复位操作时的硬件状态

项目	复位期间	复位后
主时钟振荡器（f _x ）	停止振荡	开始振荡
副时钟振荡器（f _{XT} ）	继续振荡	
内部振荡器	停止振荡	开始振荡
外围时钟（f _{xx} 至 f _{xx} /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（f _{CLK} ）， CPU 时钟（f _{CPU} ）	停止操作	安全振荡稳定时间之后开始振荡（初始化为 f _{xx} /8）
CPU	初始化	确保安全振荡稳定时间之后，才开始执行程序
看门狗定时器 2	停止操作（初始化到 0）	将内部振荡时钟作为源时钟，从 0 开始计数
内置 RAM	如果上电复位和复位输入冲突，或 CPU 访问和复位输入冲突（数据被破坏），则不确定。否则，保留复位输入后的即时数值 [✱] 。	
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	确保安全振荡稳定时间之后，才开始操作

注 V850ES/SG2 和 V850ES/SG2-H 中的固件会在内部系统复位操作被释放后需要使用一部分内部 RAM，这是因为它们支持引导交换功能。因此，在上电复位时一些 RAM 区域中的内容将不会被保留。详细信息请参见 25.3.5 复位释放后的操作。

图 25-4. WDT2RES 信号生成所引起的复位操作的时序



<R>

25.3.3 通过低压检测器执行的复位操作（LVIRES）（仅限V850ES/SG2）

在允许 LVI 操作时，如果供电电压低于通过低电压检测器检测到的电压，那么将会对系统进行复位（LVIM.LVIMD 位被设为 1 时），且对各个硬件进行初始化，恢复到初始状态。

从检测到工作电压下降开始进入复位状态，持续到工作电压高于 LVI 检测电压结束复位状态。

在复位期间主时钟振荡停止。

当 LVIMD 位 = 0 时，如果检测到低电压，那么将会生成一个中断请求信号（INTLVI）。

表 25-3. 通过低压检测器执行复位操作期间的硬件状态

<R>

<R>

<R>

项目	复位期间	复位后
主时钟振荡器（fx）	停止振荡	开始振荡
副时钟振荡器（fxT）	继续振荡	
内部振荡器	停止振荡	开始振荡
外围时钟（fxx 至 fxx/1, 024）	停止操作	确保安全振荡稳定时间之后，才开始操作
内部系统时钟（fclk）， CPU 时钟（fcPU）	停止操作	确保安全振荡稳定时间之后，才开始操作 （初始化为 fxx/8）
CPU	初始化	确保安全振荡稳定时间之后，才开始执行程序
WDT2	停止操作（初始化到 0）	将内部振荡时钟作为源时钟，从 0 开始计数
内置 RAM	如果上电复位和复位输入冲突，或 CPU 访问和复位输入冲突（数据被破坏），则不确定。 否则，保留复位输入后的即时数值 [※] 。	
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。	
LVI	操作继续	
与上述不同的片上外围功能	停止操作	确保安全振荡稳定时间之后，才开始操作

注 V850ES/SG2 中的固件会在内部系统复位操作被释放后需要使用一部分内部 RAM，这是因为它支持引导交换功能。因此，在上电复位时一些 RAM 区域中的内容将不会被保留。详细信息请参见 25.3.5 复位释放后的操作。

备注 关于低电压检测器的复位时序，请参考第 27 章 低电压检测器。

<R> 25.3.4 时钟监控器引起的复位操作（CLMRES）

允许时钟监控器操作时，会通过采样时钟（内部振荡时钟：f_R）来监控主时钟。如果检测到主时钟停止，那么系统将会复位，且对各个硬件进行初始化，恢复到初始状态。

检测到主时钟停止后，CPU 会在某段时间（模拟延迟）内处于复位状态，然后自动从复位状态中释放。在复位状态被释放后，振荡稳定定时器不会执行其计数操作，这是因为主时钟被停止。当默认启动的看门狗定时器 2 发生溢出时，CPU 会以内部振荡时钟（f_R）开始程序的执行。

以下显示了由时钟监视操作的复位信号（CLMRES）引发的复位期间各个硬件单元的状态，以及复位状态释放后各个硬件单元的状态。

关于时钟监视操作产生的复位时序，请参考图 25-5。

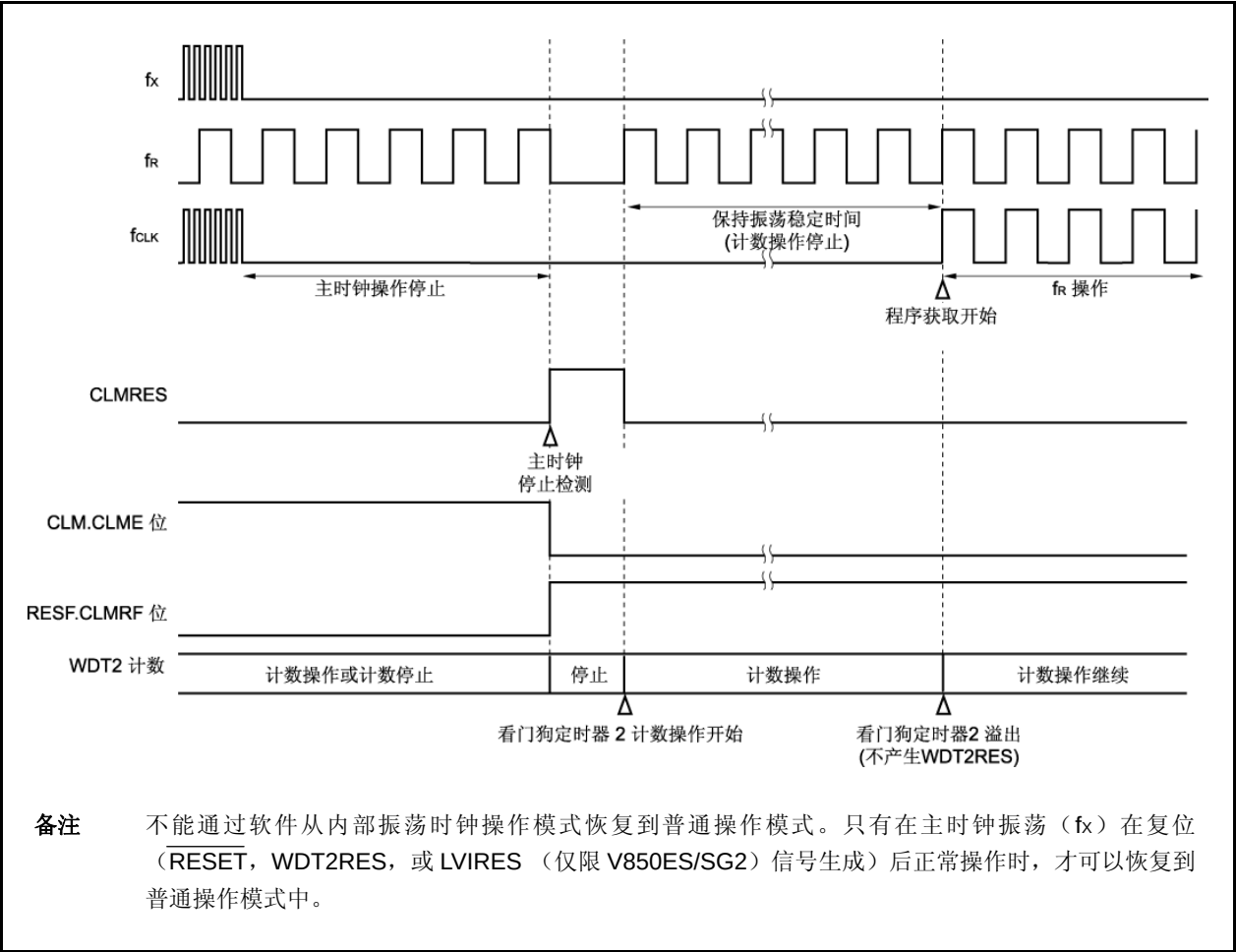
表 25-4. 时钟监视器引发的复位操作期间的硬件状态

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 ^注
副时钟振荡器（f _{XT} ）	继续振荡	
内部振荡器	停止振荡	开始振荡
外围时钟（f _{XX} 至 f _{XX} /1, 024）	停止操作	确保安全振荡稳定时间之后，才开始操作 ^注
内部系统时钟（f _{CLK} ）， CPU 时钟（f _{CPU} ）	停止操作	确保安全振荡稳定时间之后，才开始振荡（初始化为 f _{XX} /8）然而，如果看门狗定时器 2 在 CPU 执行前溢出，则将会以内部振荡时钟（f _R ）启动操作。
CPU	初始化	确保安全振荡稳定时间之后，才开始执行程序。然而，如果看门狗定时器 2 在 CPU 执行前溢出，则将会以内部振荡时钟（f _R ）启动操作。
WDT2	停止操作（初始化到 0）	开始操作。然而，如果在 CPU 执行前只有看门狗定时器 2 溢出，那么将不会生成 WDT2RES。
内置 RAM	未定义	
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	确保安全振荡稳定时间之后，才开始操作 ^注 。

注 在时钟监视器引发的复位操作之后，主时钟开始振荡时。

备注 关于时钟监视器的详细信息，请参考第 26 章 时钟监视器。

图 25-5. 时钟监视器引起的复位时序

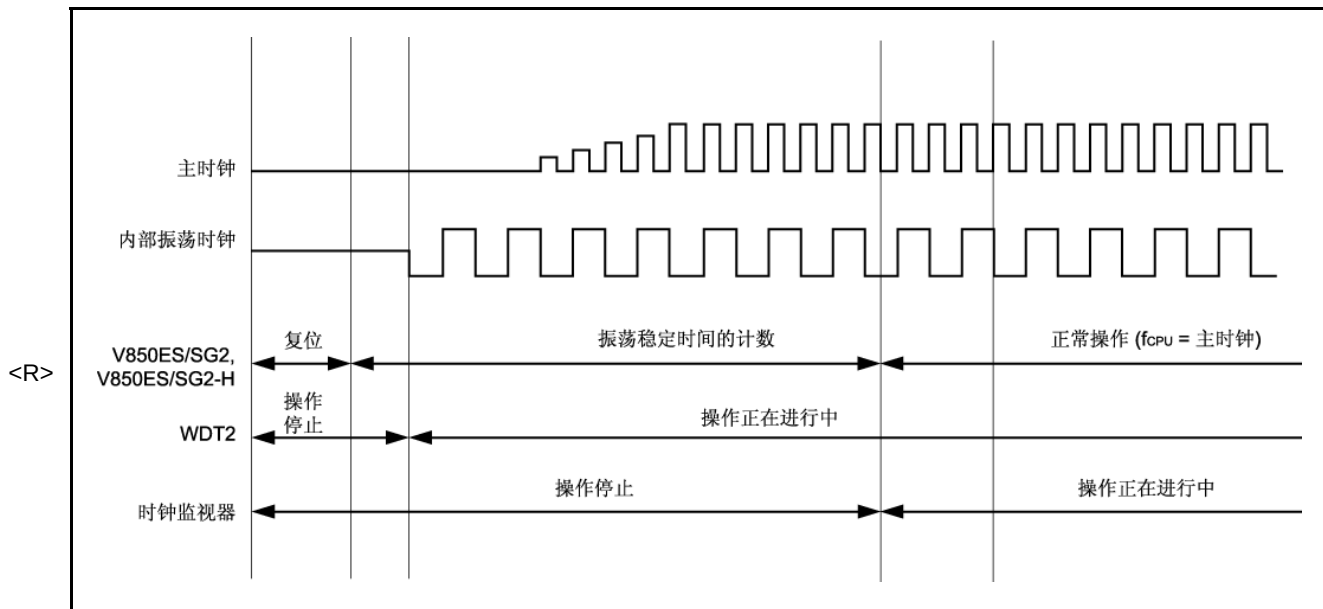


25.3.5 复位释放后的操作

复位释放以后，主时钟开始振荡且确保经过振荡稳定时间（OSTS 寄存器初始值为： $2^{16}/f_x$ ），然后 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。

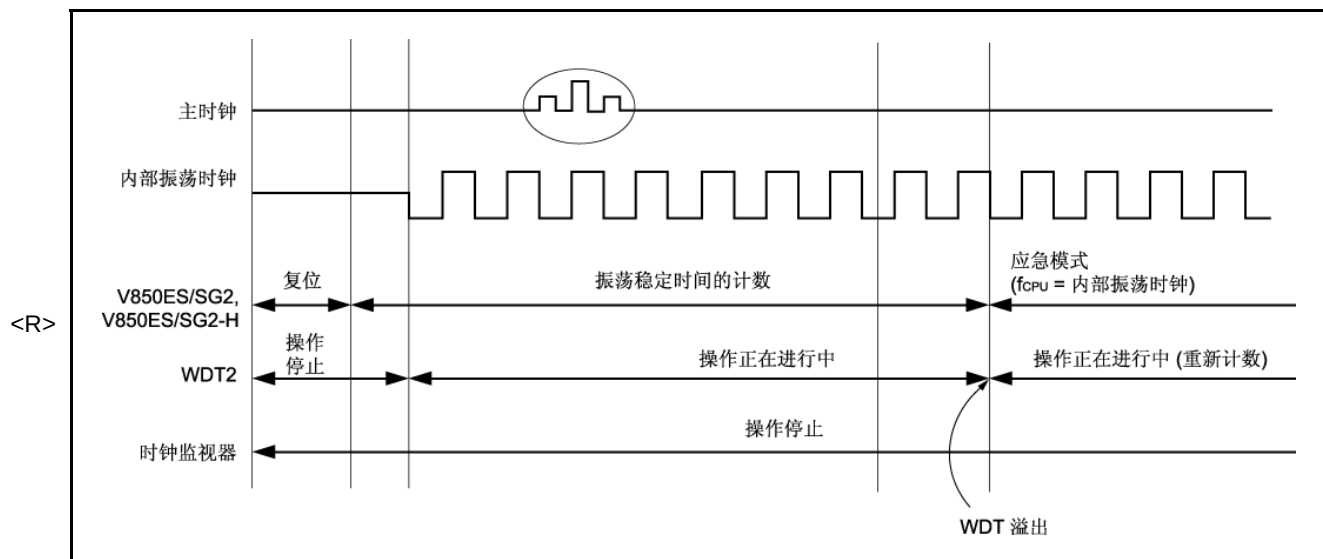
图 25-6. 复位释放后的操作



(1) 紧急操作模式

如果在振荡稳定时间还未完成之前，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。

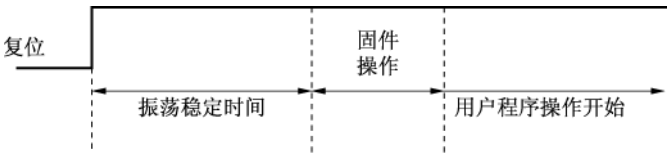
图 25-7. 复位释放后的操作



CPU 操作时钟状态可以通过 CPU 操作时钟状态寄存器（CCLS）来检测。

(2) 固件操作（仅限 flash 存储器版本）

在 flash 存储器版本中，复位释放后，在用户程序开始之前操作的片上固件可以支持引导交换功能。



V850ES/SG2: 固件操作时间: $14,974 \times (1/f_x)$ (秒)

V850ES/SG2-H: 固件操作时间: $11,994 \times (1/f_x)$ (秒)

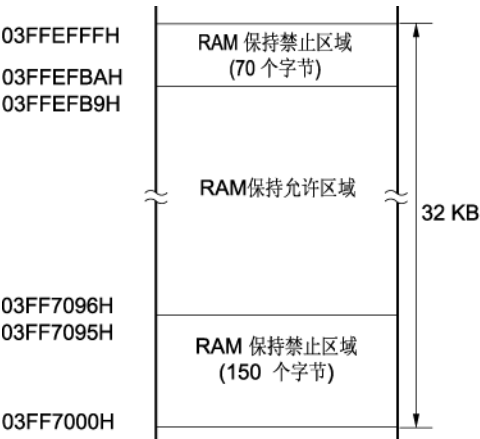
注意事项 要正确启动用户程序操作，将 **FLMD0** 引脚固定为低电平，这是因为在振荡稳定时间过后，复位状态释放，且固件操作也已经完成。

备注 f_x : 主时钟振荡频率 (MHz)

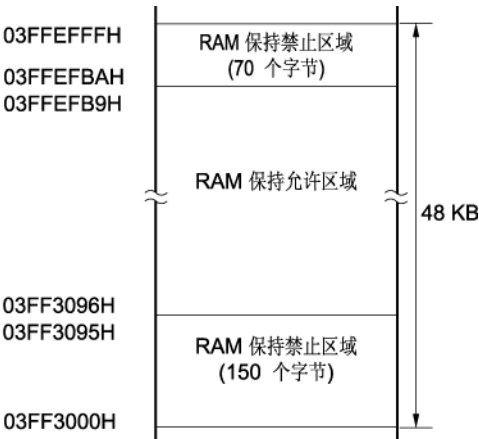
由于固件使用了一部分内部 RAM，因此即使在电源打开的状态下，下列 RAM 区域的内容在复位时不做保留。

- 内置 32 KB RAM 的版本: 03FF7000H 至 03FF7095H, 03FFEFBAH 至 03FFFFFFFFH
- 内置 48 KB RAM 的版本: 03FF3000H 至 03FF3095H, 03FFEFBAH 至 03FFFFFFFFH

(a) 带有 32 KB RAM 的版本

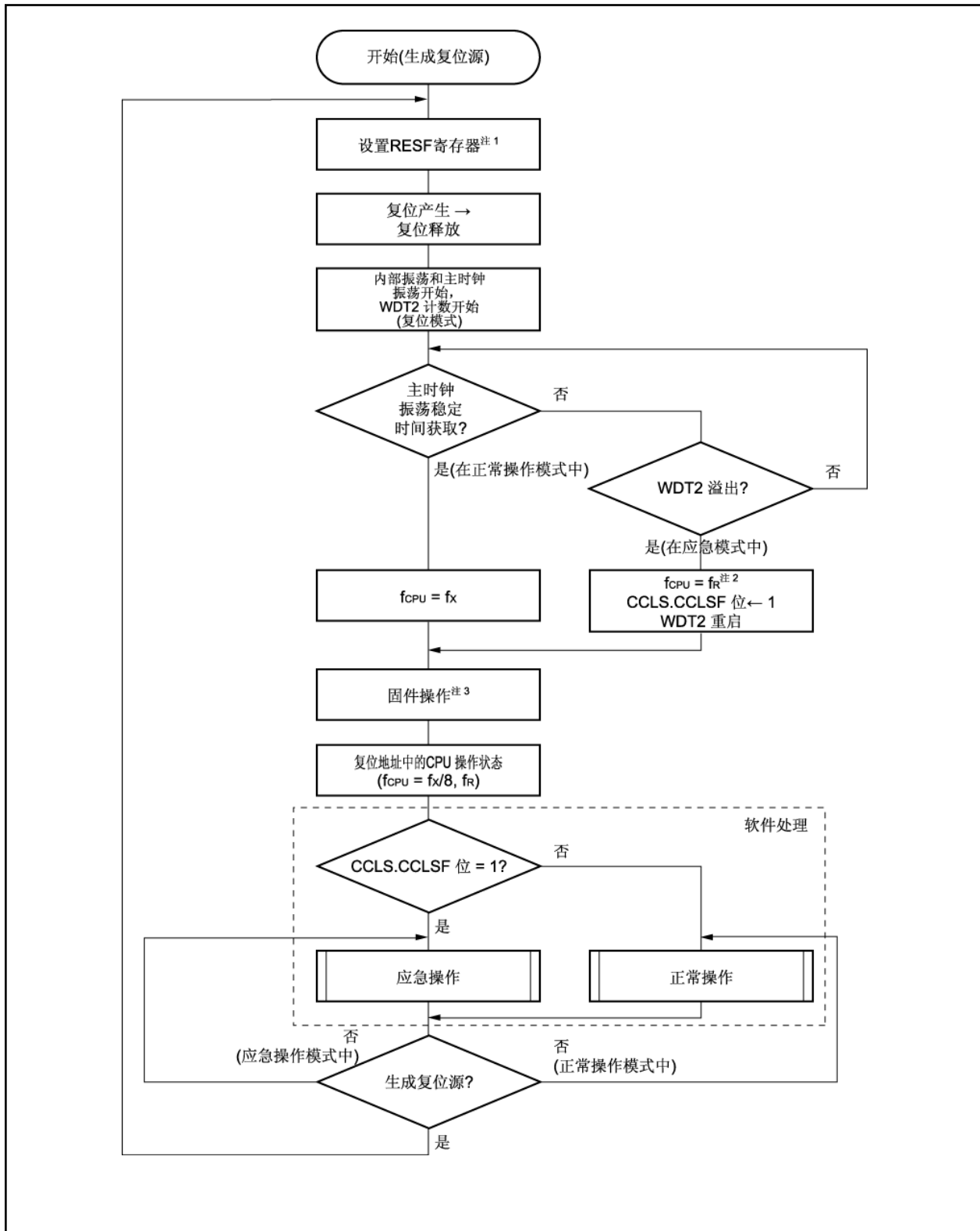


(b) 带有 48 KB RAM 的版本



25.3.6 复位功能操作流程

(1/2)



注 1. 要设置的位会根据复位源来变化。

复位源	WDT2RF 位	CLMRF 位	LVIRF 位 (仅限 V850ES/SG2)
RESET 引脚	0	0	0
WDT2	1	保持复位前的值。	保持复位前的值。
CLM	保持复位前的值。	1	保持复位前的值。
LVI (仅限 V850ES/SG2)	保持复位前的值。	保持复位前的值。	1

2. 不能停止内部振荡器。

3. 仅限内存版本。

25.4 内部RAM数据的有效/无效

(1) 内部 RAM 数据状态寄存器 (RAMS)

RAMS 寄存器是一个特殊寄存器。它只能够按照特殊顺序来写入（参见 3.4.8 特殊寄存器）。

该寄存器是一个用于显示内部 RAM 是否有效的标志寄存器。

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。

以下显示了 RAMF 位的设置/清除条件。

- 设置条件：
 - 低于指定电平电压检测
 - 由指令设置
 - 通过 WDT2 和 CLM 的复位信号生成
 - 访问 RAM 时，复位信号的生成
 - 访问 RAM 时通过 RESET 引脚生成复位信号。
- 清除条件：
 - 按照特定顺序写入 0

复位后: 01H ^註		R/W	地址: FFFFF892H					
	7	6	5	4	3	2	1	<0>
RAMS	0	0	0	0	0	0	0	RAMF

RAMF	内部 RAM 数据的有效/无效
0	有效
1	无效

注 该寄存器在通过 RESET 引脚输入（仅限 RAM 访问），看门狗定时器 2 溢出，或时钟监视器复位后被设为 01H。在通过其它源复位后，此时的寄存器值会被保持。

第 26 章 时钟监视器

26.1 功能

时钟监视器通过内部振荡时钟采样主时钟，并且在主时钟停止振荡的时候产生一个复位请求信号（CLMRES）。一旦通过操作允许标志允许时钟监视器的操作，除了复位之外不能通过其余任何方法来清零。

当通过时钟监视器生成复位（CLMRES）时，RESF.CLMRF 位将被设置。关于 RESF 寄存器的详细资料，请参见

25.2 用于检测复位源的寄存器。

时钟监视器在下列条件下自动停止。

- STOP 模式释放后的振荡稳定时间内
- 当主时钟停止时（从副时钟操作 PCC.MCK 位= 1 开始，到主时钟操作 PCC.CLS 位= 0 结束）
- 当采样时钟停止（内部振荡时钟）时
- CPU 运行于内部振荡时钟时

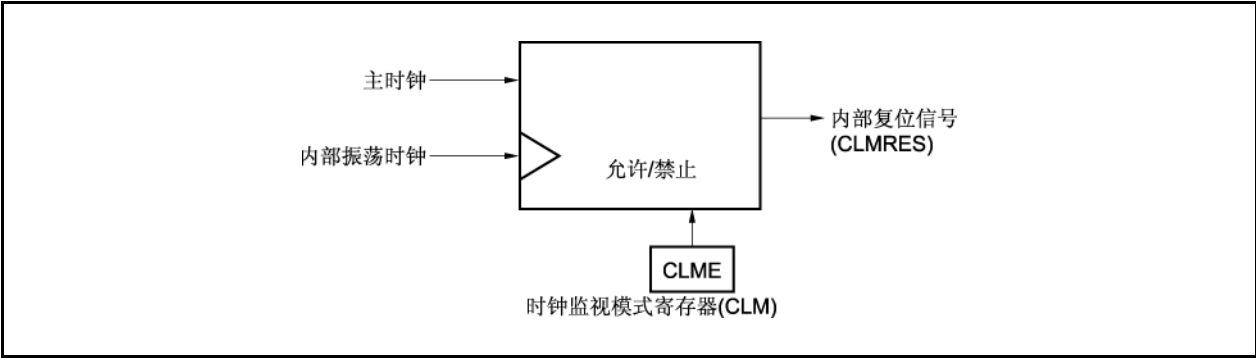
26.2 配置

时钟监视器由以下硬件组成。

表 26-1. 时钟监视器的结构

项目	配置
控制寄存器	时钟监视器模式寄存器（CLM）

图 26-1. 通过时钟监视器复位的时序



26.3 寄存器

时钟监视器通过时钟监视器模式寄存器（CLM）来控制。

（1）时钟监视器模式寄存器（CLM）

CLM 寄存器是一个特殊寄存器。它只能够通过特殊顺序来写入（参见 3.4.8 特殊寄存器）。

该寄存器用来设置操作时钟监视器的模式。

该寄存器可以以 8 位为单位或以 1 位为单位进行读写操作。

复位输入将把该寄存器置为 00H。

复位后：00H R/W 地址：FFFFF870H								
	7	6	5	4	3	2	1	<0>
CLM	0	0	0	0	0	0	0	CLME
CLME		允许或禁止时钟监视器的操作						
0		禁止时钟监视器操作。						
1		允许时钟监视器操作。						

- 注意事项**
- 一旦 CLME 位被设为 1，除了复位之外的其余任何操作都不能将其清 0。
 - 当通过时钟监视器执行复位时，CLME 位将被清为 0 并且 RESF.CLMRF 被设置为 1。

26.4 操作

这个部分说明了时钟监视器的功能。开始和停止条件如下所示。

<开始条件>

当 CLM.CLME 位设置为 1 时，允许操作。

<停止条件>

- STOP 模式被释放后对振荡稳定时间进行计数时。
- 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
- 当采样时钟停止（内部振荡时钟）时
- CPU 运行于内部振荡时钟时

表 26-2. 时钟监视器的操作状态
(当 CLM.CLME 位=1，在内部振荡时钟操作时)

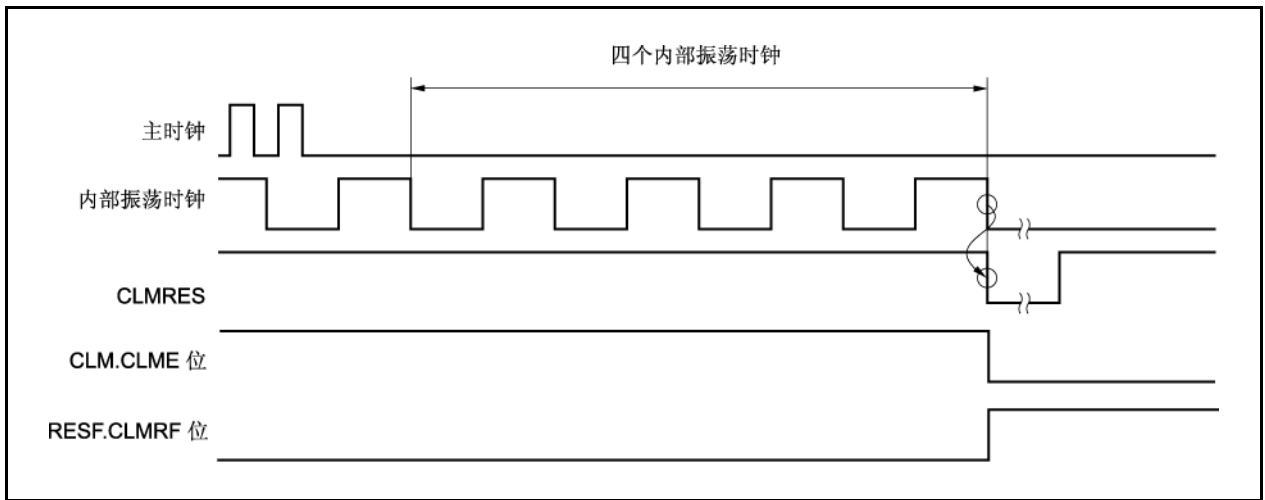
CPU 操作时钟	操作模式	主时钟状态	内部振荡时钟的状态	时钟监视器的状态
主时钟	HALT 模式	振荡	振荡 ^{注 1}	操作 ^{注 2}
	IDLE1, IDLE2 模式	振荡	振荡 ^{注 1}	操作 ^{注 2}
	STOP 模式	停止	振荡 ^{注 1}	停止
副时钟 (PCC 寄存器的 MCK 位 = 0)	副-IDLE 模式	振荡	振荡 ^{注 1}	操作 ^{注 2}
副时钟 (PCC 寄存器的 MCK 位 = 1)	副-IDLE 模式	停止	振荡 ^{注 1}	停止
内部振荡时钟	—	停止	振荡 ^{注 3}	停止
复位期间	—	停止	停止	停止

- 注
1. 内部振荡器可以通过将 RCM.RSTOP 位设为 1 来停止。
 2. 内部振荡器停止的时候，时钟监视器也停止。
 3. 内部振荡器不能通过软件来停止。

(1) 当主时钟振荡器停止时 (CLME 位=1) 的操作

当 CLME 位=1 的时候如果主时钟的振荡停止, 将会产生一个如图 26-2 所示的内部复位信号 (CLMRES)。

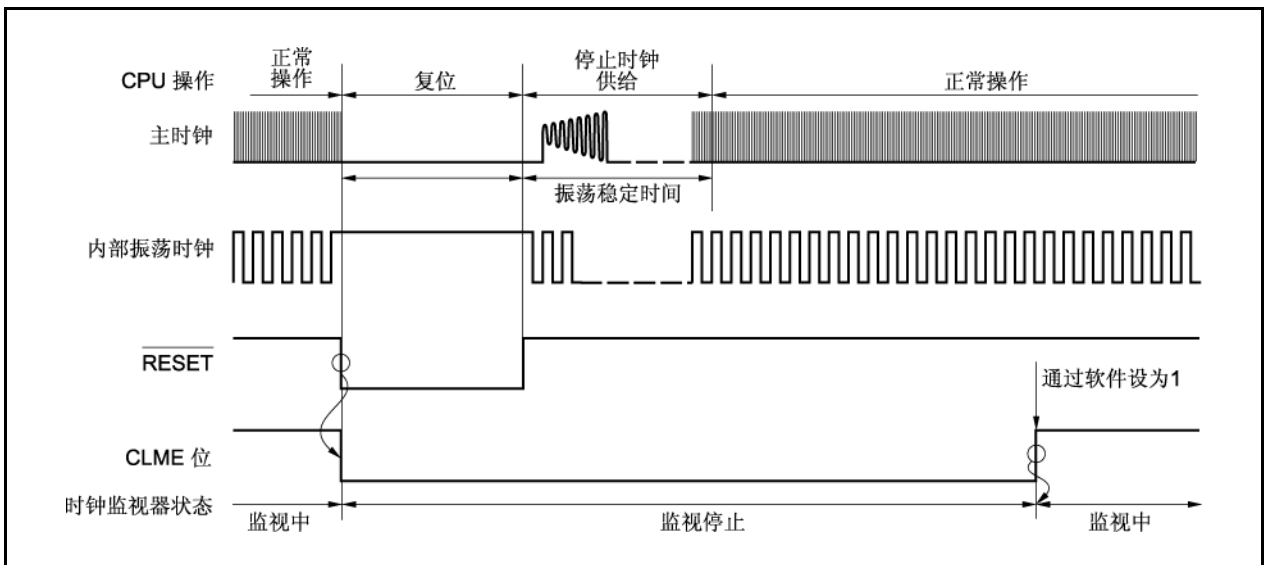
图 26-2. 主时钟振荡停止引起的复位周期

(2) $\overline{\text{RESET}}$ 输入后的时钟监视器状态

$\overline{\text{RESET}}$ 输入将 CLM.CLME 位清除为 0, 并且停止时钟监视器的操作。在主时钟振荡稳定以后, 通过软件把 CLME 位设为 1, 监控操作开始。

图 26-3. $\overline{\text{RESET}}$ 输入后时钟监视器的状态

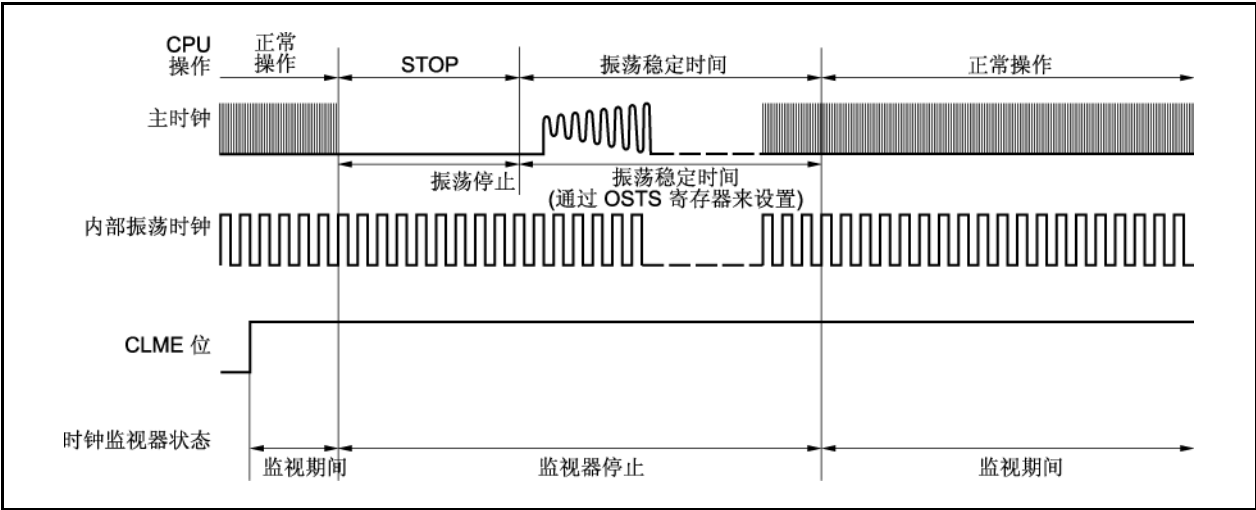
($\overline{\text{RESET}}$ 输入后, 在主时钟振荡稳定时间的最后设置 CLM.CLME 位 = 1)



(3) 在 STOP 模式时被或 STOP 模式被释放后的操作

如果在 CLM.CLME 位=1 时设置 STOP 模式，监控操作将在振荡稳定计数期间停止于 STOP 模式。振荡稳定时间过后监视器将会自动开始操作。

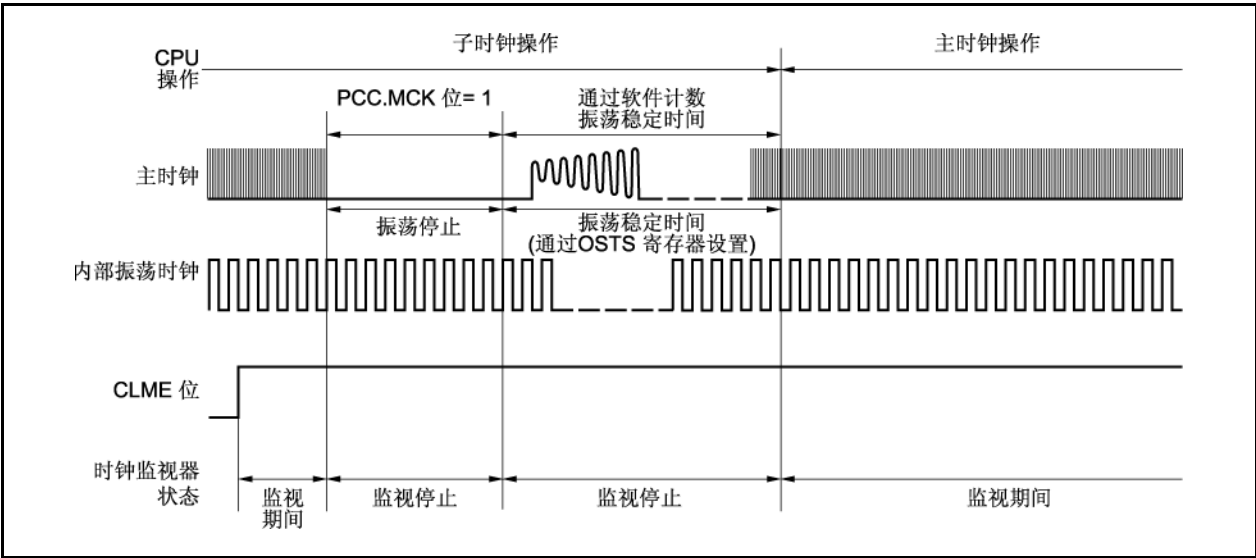
图 26-4. 在 STOP 模式时被或 STOP 模式释放后的操作



(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 PCC.MCK 位置 1 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 26-5. 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.CCLSIF 位 = 1）的操作。

当 CCLSIF 位设为 1 时，监控器的操作不会停止，即使 CLME 位被设为 1。

第 27 章 低电压检测器

低电压检测器 (LVI) 仅为 V850ES/SG2 提供。

<R> RAM 保持电压检测操作与仿真功能提供于 V850ES/SG2 和 V850ES/SG2-H 中。

<R> 27.1 功能

低电压检测器 (LVI) 具有以下功能。

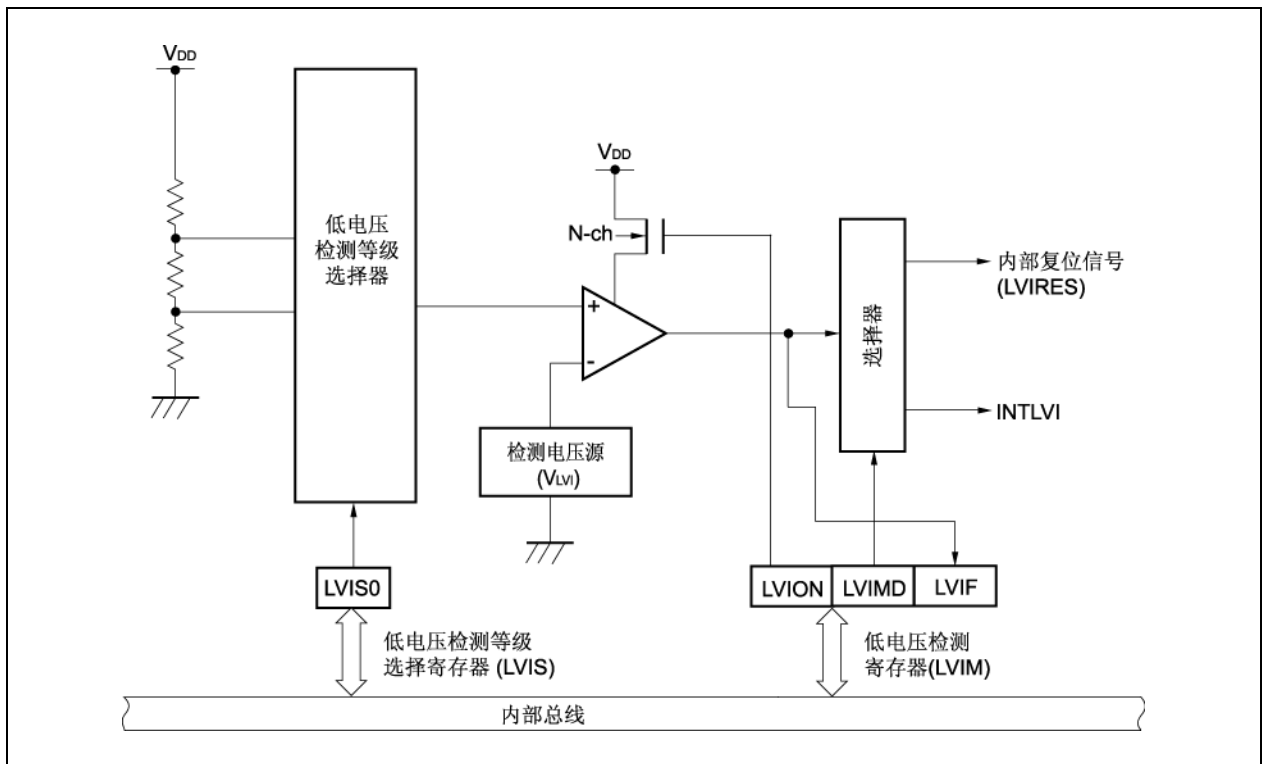
- 将供电电压 (V_{DD}) 与检测电压 (V_{LVI}) 进行比较, 并在 $V_{DD} < V_{LVI}$ 时生成一个内部中断信号或内部复位信号 (LVIRES)。
- 要检测的供电电压的电平可以通过软件来改变 (分为两个步骤)。
- 可以通过软件选择中断或复位信号。

如果低电压检测器用于产生复位信号, 那么当 LVIRES 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 25 章 复位功能。

27.2 配置

图 27-1 显示了低电压检测器的框图。

图 27-1. 低电压检测器的框图



27.3 寄存器

低电压检测器使用以下寄存器进行控制。

- 低电压检测寄存器（LVIM）
- 低电压检测等级选择寄存器（LVIS）

(1) 低电压检测寄存器（LVIM）

LVIM 寄存器是一个特殊的寄存器。它只能够通过特殊顺序来写入（参见 3.4.8 特殊寄存器）。
LVIM 寄存器用于允许或禁止低电压检测，并用于设置低电压检测器的操作模式。
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。但是，LVIF 位是只读的。

复位后：注 1							
R/W 地址： FFFFF890H							
	<7>	6	5	4	3	2	<1> <0>
LVIM	LVION	0	0	0	0	0	LVIMD LVIF
	LVION	允许或禁止低电压检测操作					
	0	禁止操作。					
	1	允许操作。					
	LVIMD	低电压检测操作模式的选择					
	0	当供电电压 < 检测电压时产生中断请求信号（INTLVI）。					
	1	当供电电压 < 检测电压时产生内部复位信号（LVIRES）。					
	LVIF ^{注 2}	低电压检测标志					
	0	供电电压 > 检测电压， 或当禁止操作时					
	1	所连接的电源的供电电压< 检测电压					

- 注 1. 通过低电压检测来复位： 82H
通过其他源复位： 00H
2. 当 LVION 位 = 1 且 LVIMD bit = 0 时，LVIF 标志的值将会被输出为中断请求信号（INTLVI）。

- 注意事项 1. 当 LVION 和 LVIMD 位被设为 1 时，直到 LVI 之外的复位请求产生，低电压检测器都不会停止。
2. 当 LVION 位被设为 1 时，LVI 电路中的比较器开始运行。设置 LVION 位后，在检测电压前通过软件等待 0.2 ms 或更长的时间。
3. 在 V850ES/SG2-H 中，禁止写入 LVIM 寄存器的地址（FFFFF890H）。
4. 确保将第 2 位~ 第 6 位清除为“0”。

<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 位为单位或以 1 位为单位进行读取操作或写操作。

复位后：注		R/W		地址： FFFFF891H				
	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	0	0	LVIS0

LVIS0	检测等级
0	3.0 V ±0.15 V
1	2.85 V ±0.15 V （禁止设置）

注 通过低电压检测来复位： 保持
通过其他源复位： 00H

- 注意事项
1. LVION 和 LVIMD 位设为 1 后，此寄存器不能写入，直到 LVI 之外的复位请求产生。
 2. 在 V850ES/SG2-H 中，禁止写入 LVIS 寄存器的地址（FFFFF891H）。
 3. 确保将第 7 位~ 第 1 位清除为“0”。

<R>

27.4 操作

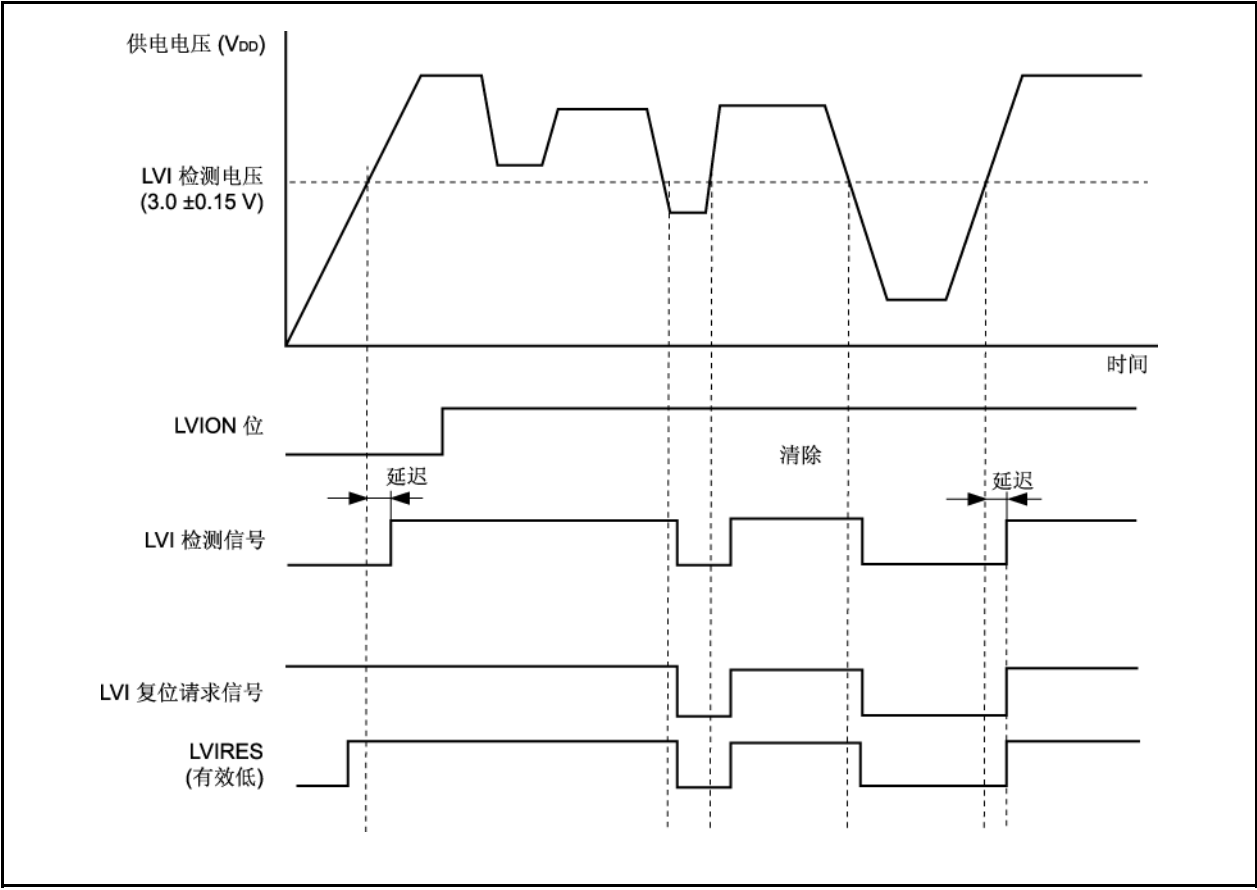
依据 LVIM.VIMD 位的设置，会产生一个中断请求信号（INTLVI）或者一个内部复位信号（LVIRES）。以下显示了指定各个操作的方法，以及时序图。

27.4.1 用作内部复位信号（LVIRES）

- <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（最大值）或更多个的等待周期。
- <5> 通过 LVIM.LVIF 位检验供电电压是否大于检测电压。
- <6> 设置 LVIMD 位为 1（为产生内部复位信号（LVIRES））。

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 27-2. 低电压检测电路的操作时序（LVIMD 位= 1）



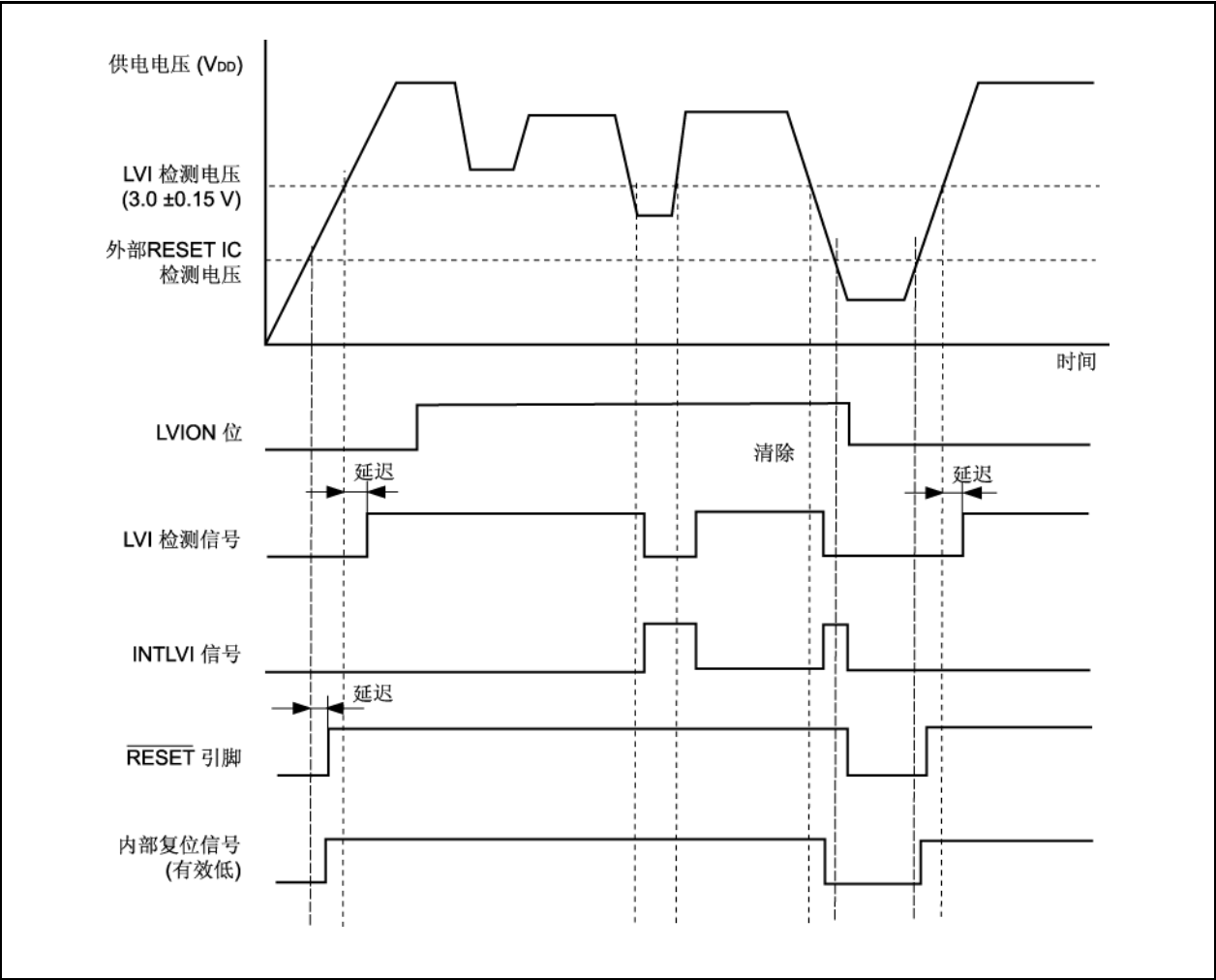
27.4.2 用作中断 (INTLVI)

- <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（最大值）或更多的等待周期。
 - <5> 通过 LVIM.LVIF 位检验供电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。

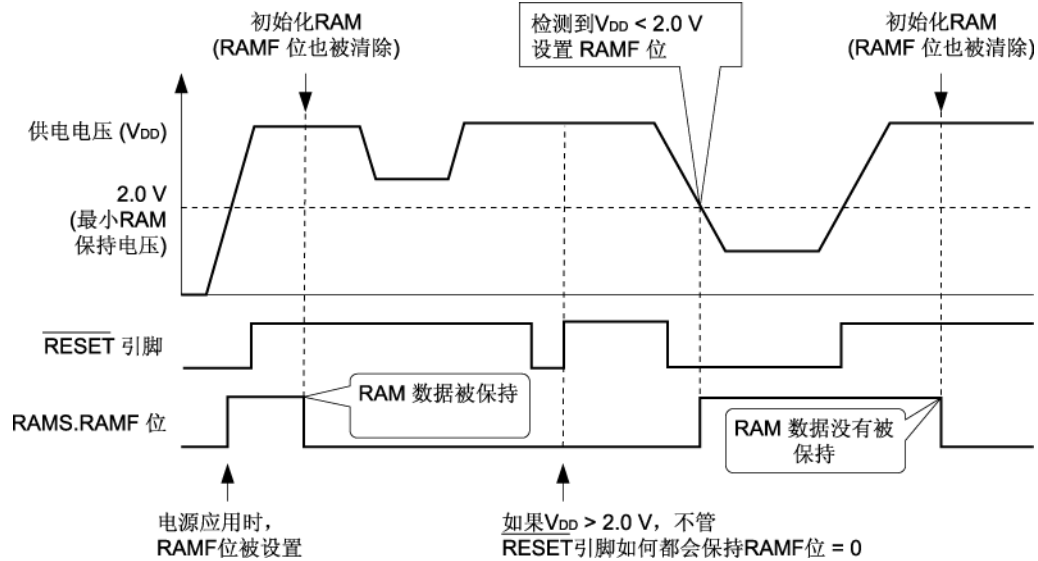
图 27-3. 低电压检测电路的操作时序 (LVIMD 位= 0)



27.5 RAM保持电压检测操作（为V850ES/SG2 和 V850ES/SG2-H提供）

比较供电电压和检测电压。当供电电压低于检测电压时（包括上电），RAMS.RAMF 位设置为 1。

图 27-4. RAM 保持电压检测功能的操作时序



- 备注**
1. 如果供电电压低于最低 RAM 保持电压（2.0 V（TYP.）），那么 RAMF 位将被设为 1。
 2. 不管 RESET 引脚状态如何，RAMF 位都会进行操作。

27.6 仿真功能（为V850ES/SG2 和 V850ES/SG2-H所提供）

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。

该寄存器仅在仿真模式中有效。在普通模式中它是无效的。

(1) 外围仿真寄存器 1（PEMU1）

复位后: 00H

R/W

地址: FFFFF9FEH

	7	6	5	4	3	2	1	0
PEMU1	0	0	0	0	0	EVARAMIN	0	0

EVARAMIN	RAM 保持电压检测信号的假定
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]

当使用在线仿真器时，RAMF 的伪仿真通过重写调试器上的寄存器来实现。

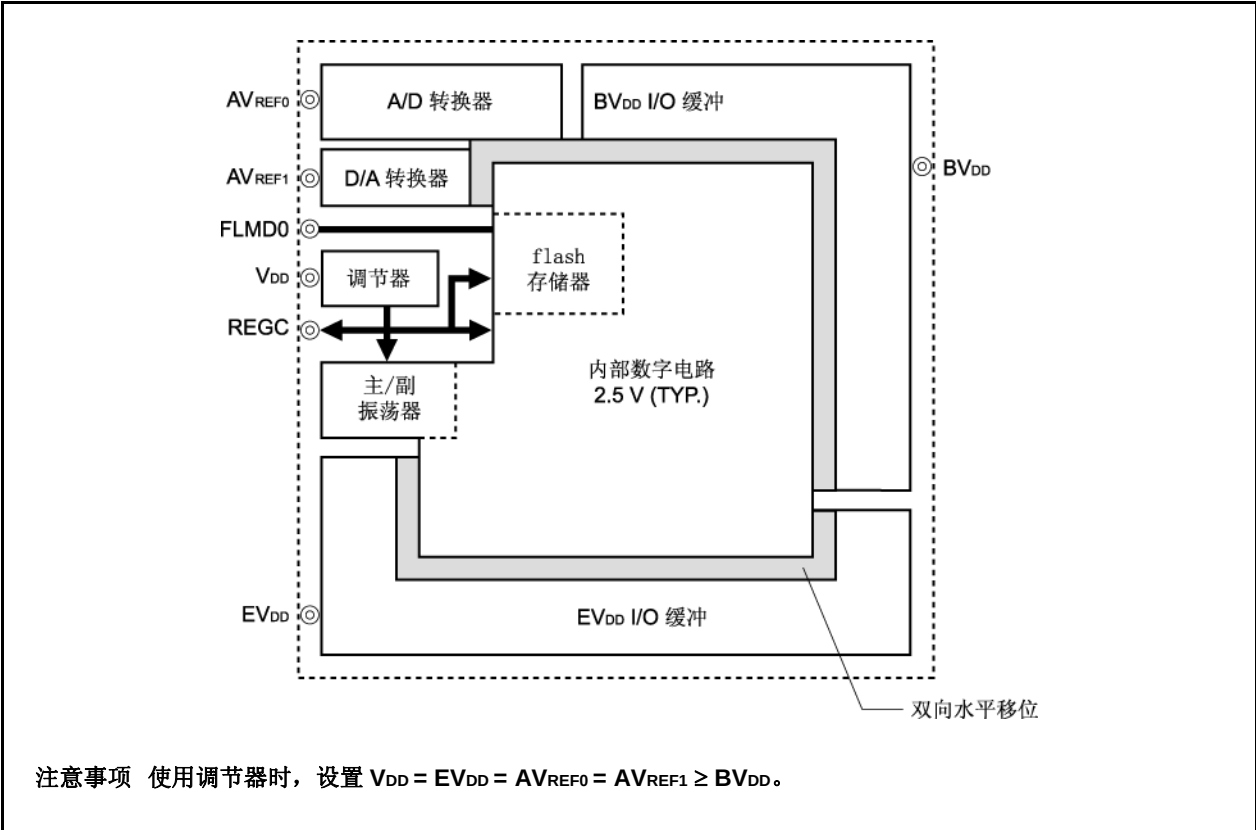
- <1> CPU 中断（CPU 操作停止）。
- <2> 通过寄存器写命令设置 EVARAMIN 位为 1。
设置 EVARAMIN 位为 1，RAMF 位被硬件设置为 1。（内部 RAM 数据无效）。
- <3> 再次通过寄存器写命令将 EVARAMIN 位清零。
除非执行此操作（将 EVARAMIN 位清除为 0），否则 RAMF 位不能通过 CPU 操作指令清 0。
- <4> 运行 CPU 并返回仿真。

第 28 章 调节器

28.1 概要

<R> V850ES/SG2 和 V850ES/SG2-H 中包括一个可以降低能耗减少噪声的调节器。
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，D/A 转换器，和输出缓冲器）。调节器输出电压设置为 2.5 V（典型值）。

<R> 图 28-1. 调节器



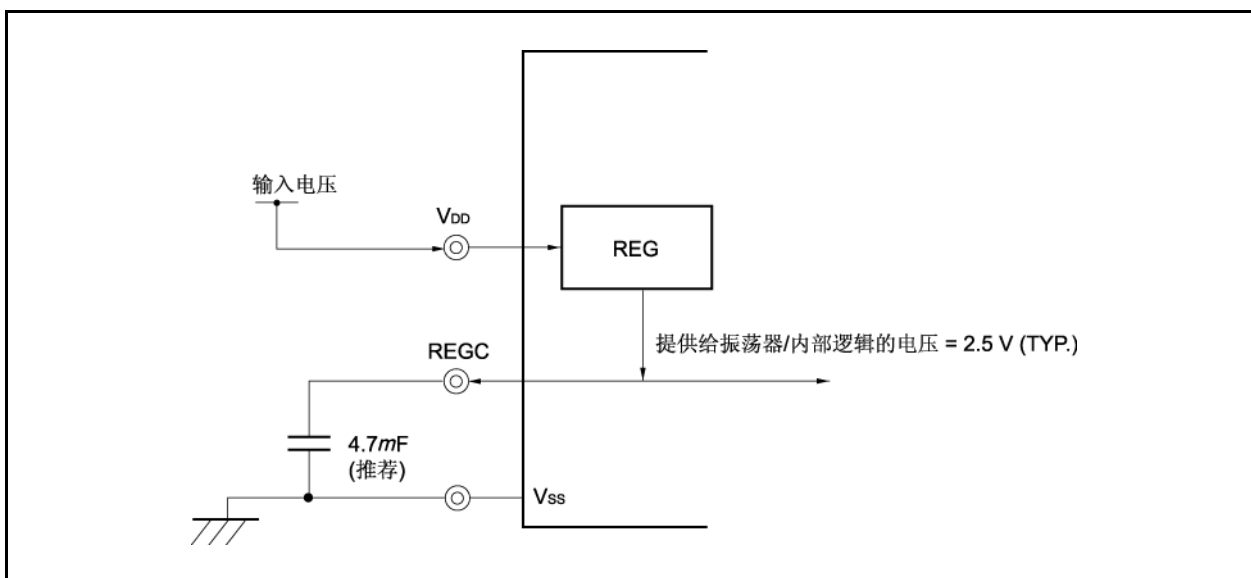
28.2 操作

该产品的调节器可以运行在各种模式下（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 28-2. REGC 引脚连接



第 29 章 ROM 修正功能

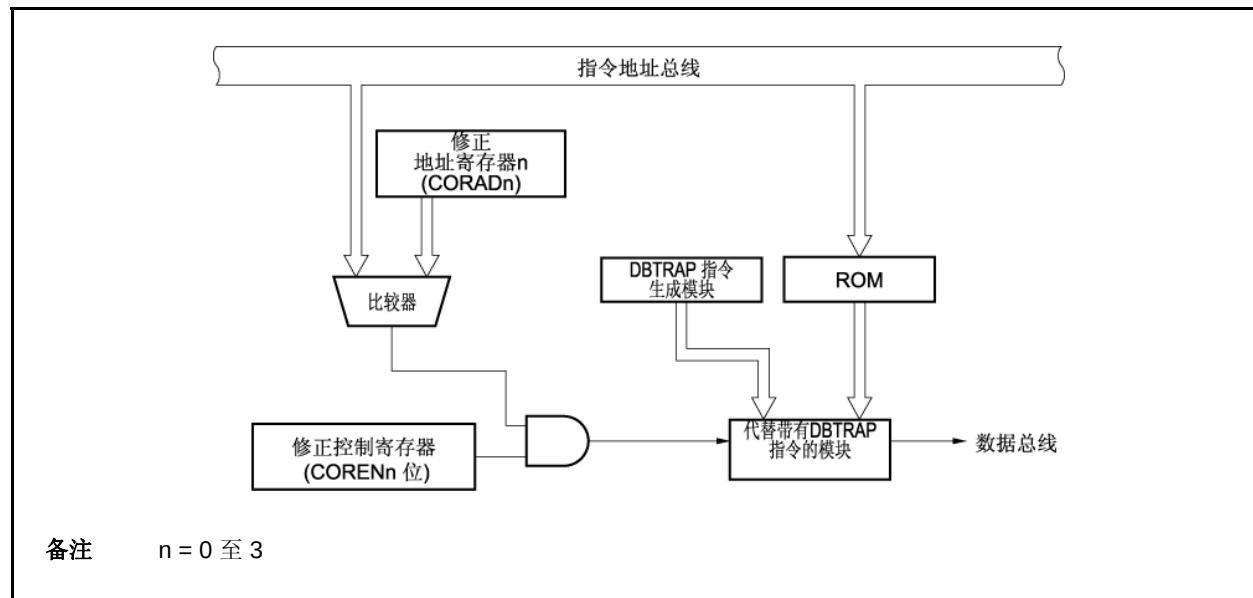
29.1 概述

ROM 修正功能用外部存储器或者内部 RAM 的程序来替代内部 ROM 的部分程序。

通过使用此功能，可以修正内部 ROM 的程序错误。

可以修正的地址多达 4 个。

图 29-1. ROM 修正的框图



29.2 寄存器

(1) 修正地址寄存器 0 到 3 (CORAD0 到 CORAD3)

CORAD0 到 CORAD3 寄存器用于设置 ROM 中待修正程序的首地址。

因为提供 4 个 CORADn 寄存器，所以程序可以在 4 个地方修改 (n = 0 到 3)。

CORADn 寄存器可以以 32 位为单位进行读取或写入。

如果 CORADn 寄存器的高 16 位用作 CORADnH 寄存器，低 16 位用作 CORADnL 寄存器，那么这些寄存器可以进行 16 位读写。

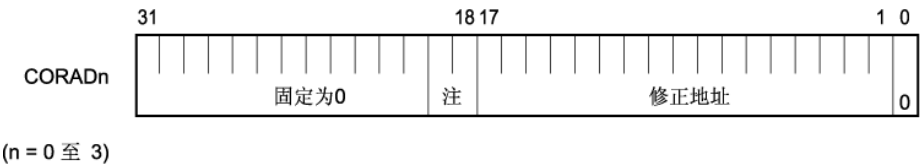
复位输入将该寄存器清除为 00000000H。

因为对各个产品来说，ROM 容量并不相同，在下列范围内设置修正地址。

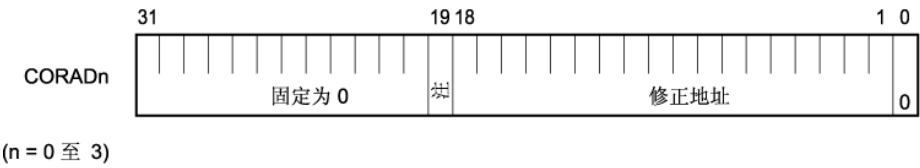
- μ PD703260, 703260Y, 703270, 703270Y, 703280, 703280Y (256 KB) : 0000000H 至 003FFFFH
- μ PD703261, 703261Y, 703271, 703271Y, 703281, 703281Y, 70F3261, 70F3261Y, 70F3271, 70F3271Y, 70F3281, 70F3281Y (384 KB) : 0000000H 至 005FFFFH
- <R> • μ D703262, 703262Y, 703272, 703272Y, 703282, 703282Y, 703262HY, 703272HY, 703282HY (512 KB) : 0000000H 至 007FFFFH
- <R> • μ PD703263, 703263Y, 703273, 703273Y, 703283, 703283Y, 70F3263, 70F3263Y, 70F3273, 70F3273Y, 70F3283, 70F3283Y, 703263HY, 703273HY, 703283HY, 70F3263HY, 70F3273HY, 70F3283HY (640 KB) : 0000000H 至 009FFFFH

复位后: 00000000H R/W 地址: CORAD0 FFFF840H,
CORAD0L FFFF840H, CORAD0H FFFF842H,
CORAD1 FFFF844H,
CORAD1L FFFF844H, CORAD1H FFFF846H,
CORAD2 FFFF848H,
CORAD2L FFFF848H, CORAD2H FFFF84AH,
CORAD3 FFFF84CH,
CORAD3L FFFF84CH, CORAD3H FFFF84EH

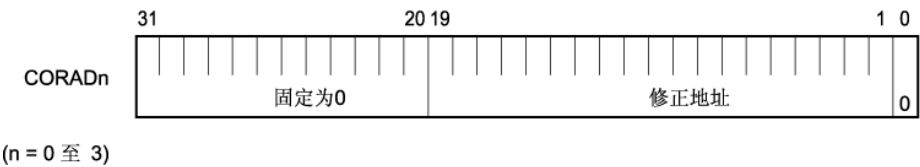
(a) 256 KB



(b) 384 KB, 512 KB



(c) 640 KB



注 清除为 0。

(2) 修正控制寄存器 (CORCN)

CORCN 寄存器可以禁止或使能在设置地址中的修正操作 (n = 0 到 3)。其地址是由 CORADn 寄存器设定的。
每一个通道都可以由此寄存器使能或禁止。
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。
复位输入将把该寄存器置为 00H。

复位后: 00H		R/W	地址: FFFF880H					
CORCN	7	6	5	4	3	2	1	0
	0	0	0	0	COREN3	COREN2	COREN1	COREN0
CORENn		允许/禁止修正操作						
0		禁止						
1		允许						
备注		n = 0 至 3						

表 29-1. CORCN 寄存器位和 CORADn 寄存器之间的对应关系

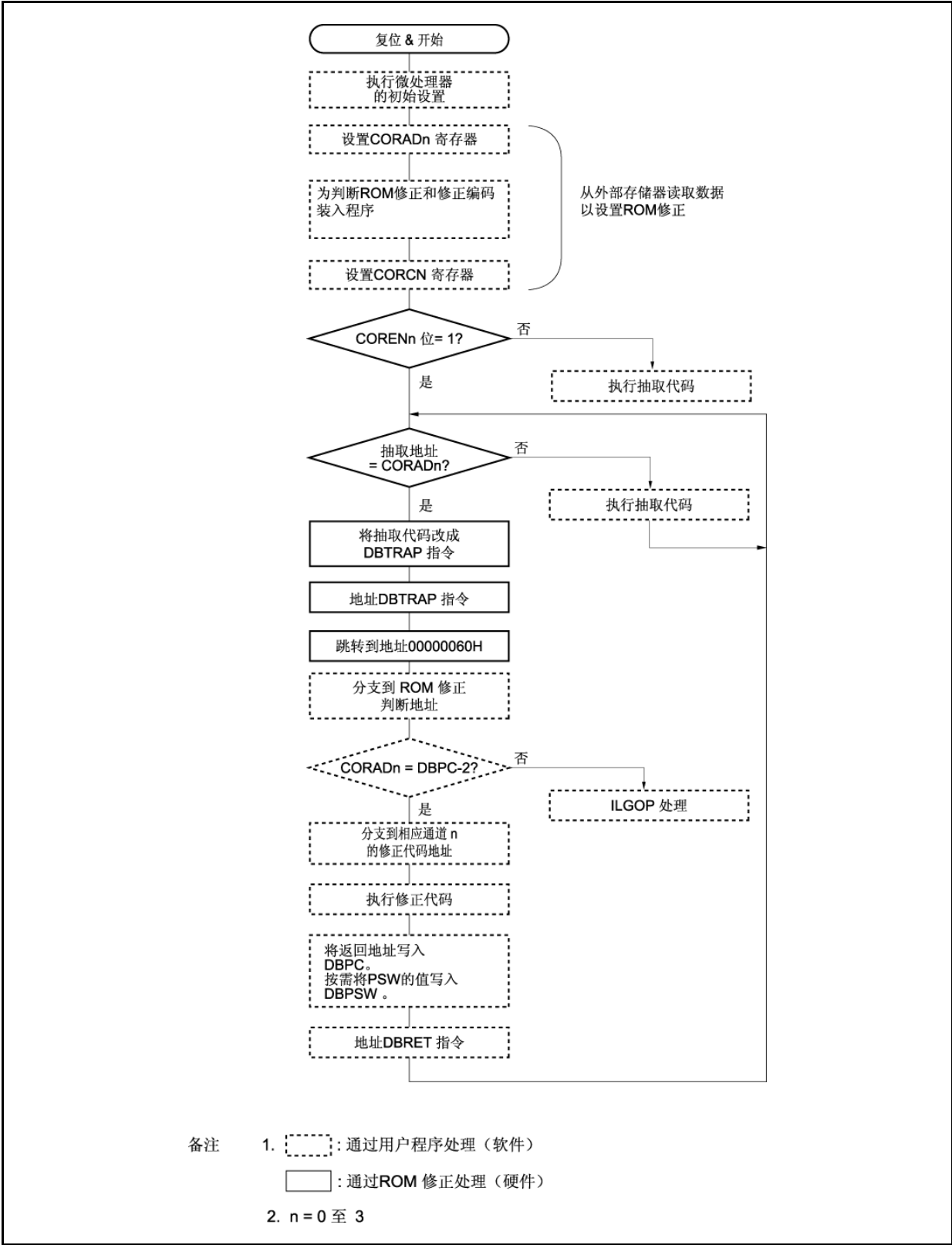
CORCN 寄存器位	相应的 CORADn 寄存器
COREN3	CORAD3
COREN2	CORAD2
COREN1	CORAD1
COREN0	CORAD0

29.3 ROM修正操作和程序流程

- <1> 如果要修正的地址和内部 ROM 的读取地址匹配，则用 DBTRAP 指令取代读取码。
- <2> 当执行 DBTRAP 指令时，执行的分支转向地址 00000060H。
- <3> 分支跳转之后的软件处理引起 ROM 修正的判定结果（读取地址和 ROM 修正操作得到确认），并且执行分支来修正软件。
- <4> 执行完修正软件之后，设置返回地址，由 DBRET 指令启动返回处理。

注意事项 完成 <3> 和 <4>的软件必须在内部 ROM/RAM 中执行。

图 29-2. ROM 修正操作和程序流程



29.4 注意事项

- (1) 在 CORADn 寄存器中设置待修正地址时，根据内部 ROM 的容量将对应高位清 0。
- (2) ROM 修正功能无法用于修正内部 ROM 中的数据。只能用于修正指令代码。如果 ROM 修正用于修正数据，那么此数据将被 DBTRAP 指令代码替换。
- (3) 在写 CORCNn 寄存器结束之前，有关 ROM 代码的 ROM 修正不会执行。
- (4) 执行完 DBTRAP 指令后，PSW.NP、EP、和 DI 位设置为 111，并且不能响应中断/异常。执行完 DBTRAP 指令后，根据要求改变 PSW 寄存器。
- (5) 在 DBTRAP 指令执行期间，DBPC 和 DBPSW 寄存器可以访问。
- (6) 如果 CORCNn 寄存器设置（使能）后立即按照修正地址设置的地址进行操作，那么可能得不到正常的操作（不产生 DBTRAP）。

第 30 章 flash存储器

<R> 以下产品是 V850ES/SG2 和 V850ES/SG2-H 的 flash 存储器版本。

注意事项 在 flash 存储器版本和掩模 ROM 版本之间存在噪声容限和噪声辐射数值差别。当从试验制造到量产的过程中考虑从 flash 存储器版本更改为掩模 ROM 版本时，确认对掩模 ROM 版本的商业样片（CS）（不是过程样片（ES））进行充分评估。

关于重写 flash 存储器的相关电气规范，参阅第 32 章 电气规范。

- <R>
- μ PD70F3261, 70F3261Y, 70F3271, 70F3271Y, 70F3281, 70F3281Y:
384 KB 片上 flash 存储器
 - μ PD70F3263, 70F3263Y, 70F3273, 70F3273Y, 70F3283, 70F3283Y, 70F3263HY,
70F3273HY, 70F3283HY:
640 KB 片上 flash 存储器

flash 存储器版本通常用于以下开发环境和量产应用中。

- <R>
- 在 V850ES/SG2 或 V850ES/SG2-H 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产中依照规范对软件区分。
 - 方便存货管理。
 - 出库后的软件升级

30.1 特点

- 4 字节 /1 个时钟访问（当指令已取得时）
- 容量：640 KB / 384 KB
- 写入电压：独立电源擦除 / 写入
- 重写方法
 - 使用专用 flash 存储器编程器通过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）
- 使用自编程的引导交换功能对全部 flash 存储器区安全重写
- 自编程期间可以应答中断

30.2 存储器配置

<R> V850ES/SG2 和 V850ES/SG2-H 的内部 flash 存储器区域以块（Block）为单位进行划分，如下所示：

- V850ES/SG2
 - 640 KB: 16 块
 - 384 KB: 12 块
- V850ES/SG2-H
 - 640 KB: 8 块

<R>

可以以块为单位进行编程 / 擦除。所有的块也可以一次擦除。

当使用引导交换功能时，分配在块 0 到 1 地址中的物理存储器可由分配在块 2 到 3 的物理存储器来代替。引导交换功能的更多细节，请参见 **30.5 通过自编程写入**。

图 30-1. flash 存储器映射 (1/2)

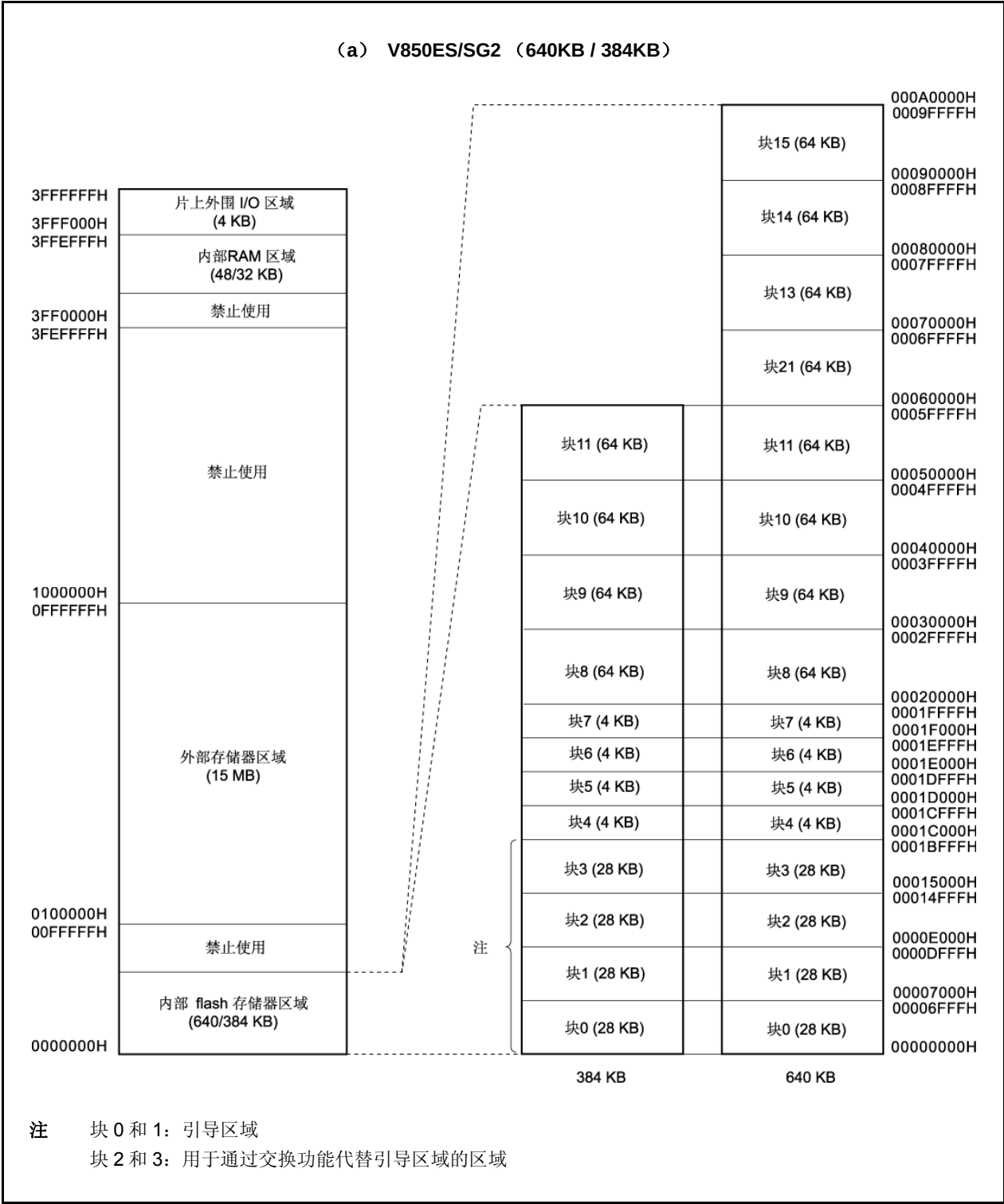
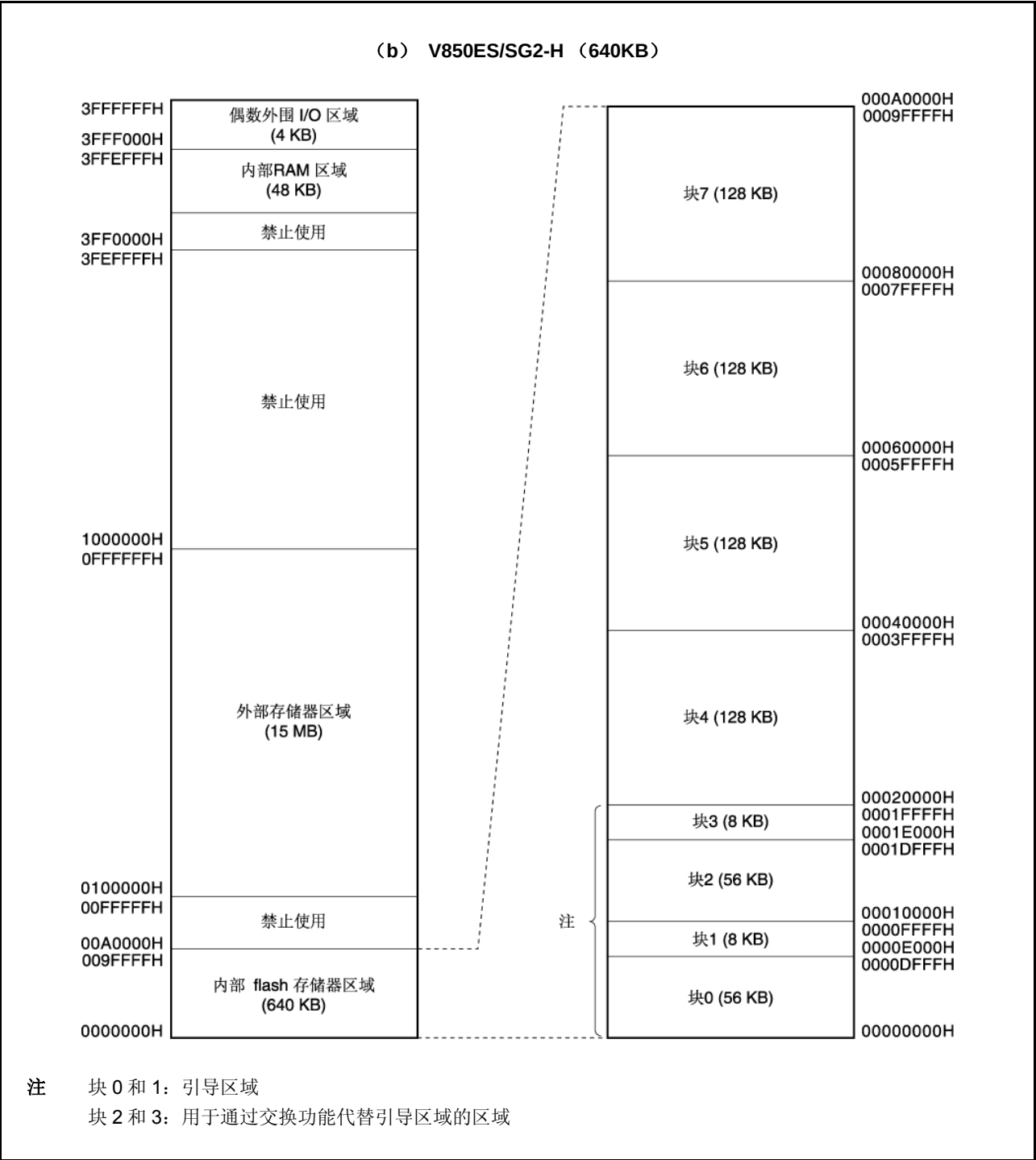


图 30-1. flash 存储器映射 (2/2)

<R>



30.3 功能概述

<R> V850ES/SG2 和 V850ES/SG2-H 的内部 flash 存储器可以使用专用的 flash 编程器写入，不论 V850ES/SG2 或 V850ES/SG2-H 是否已经被安装到目标系统（离线 / 在线编程）。

另外，还支持安全功能，禁止将用户程序写入到内部 flash 存储器，因此程序不能由未经授权的人擅自修改。

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改变的目标系统应用比较理想。也支持引导交换功能，可以对整个 flash 存储器的安全重写。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，例如与外围设备通讯时。

表 30-1. 重写方法

写入方法	功能概述	工作模式
板上编程	设备安装到目标系统后，通过专用 flash 存储器编程器，可以重写 flash 存储器。	flash 存储器编程模式
离线编程	在安装到目标系统之前，可以使用专用 Flash 编程器和专用编程适配版（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户程序进行重写，该用户程序已经用离线/板上编程方法预先写入到 flash 存储器中。（自编程期间，不能获取指令且不能访问内部 flash 存储器区域的数据。因此，写入程序必须预先发送到内部 RAM 或外部存储器。	正常工作模式

备注 FA 系列是 Naito Densei Machida Mfg. Co., Ltd 的产品

表 30-2. 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
块擦除	擦除指定的存储器块中的内容。	√	√
芯片擦除	整个存储器区域中的内容一次擦除。	√	×
写入	写入到指定地址，并且进行校验检查，写入的电平是否安全完成。	√	√
校验 / 校验和	从 flash 存储器中读出的数据与从 flash 编程器发送的数据进行比较。	√	× (可由用户程序读取)
空白检测	检查整个存储器的擦除状态。	√	√
安全设置	可以禁止块擦除命令、芯片擦除命令和编程命令的使用。	√	× 仅支持从使能设置更改为禁止设置。

下表列出了安全功能。块擦除命令禁止功能、片擦除命令禁止功能、编程命令禁止功能以及编程命令禁止功能在出库时都默认为使能，通过在线 / 离线编程重写来进行安全设置。各种安全功能可以同时组合使用。

<R>

表 30-3. 安全功能

功能	功能概述
块擦除命令禁止	所有块禁止执行块擦除命令。可以通过执行一次片擦除命令来初始化设置为禁止块擦除。
整片擦除命令禁止	禁止执行块擦除和整片擦除命令。一旦设置为禁止，由于不能执行整片擦除命令，不能被初始化设置为禁止。
编程命令禁止	所有的块都禁止执行写入和数据擦除命令。可以通过执行整片擦除命令来初始化设置为禁止。
读取命令禁止	不支持（永远禁止）。
引导区域重写禁止	不支持。

<R>

表 30-4. 安全设置

功能	当设置每种安全措施时的擦除、写入、读取操作 (√: 可执行, ×: 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
块擦除命令禁止	块擦除命令: × 芯片擦除命令: √ 编程命令: √ 读取命令: ×	块擦除: (FlashBlockErase) √ 芯片擦除: - flash 存储器字写入 (FlashWordWrite): √ flash 存储器字读取 (FlashWordRead): √	可以通过芯片擦除命令来初始化设置为禁止。	保有当设置从使能更改为禁止时是支持的。
芯片擦除命令禁止	块擦除命令: × 芯片擦除命令: × 编程命令: √ 读取命令: ×	块擦除 (FlashBlockErase): √ 芯片擦除: - 写入 (FlashWordWrite): √ 读取 (FlashWordRead): √	不能被初始化为禁止。	
编程命令禁止	块擦除命令: × 芯片擦除命令: √ 编程命令: × 读取命令: ×	块擦除 (FlashBlockErase): √ 芯片擦除: - 写入 (FlashWordWrite): √ 读取 (FlashWordRead): √	可以通过芯片擦除命令来初始化设置为禁止。	

注 在这种情况下，由于擦除命令无效，因此无法写入与已存入 flash 存储器中数据的不同数据。

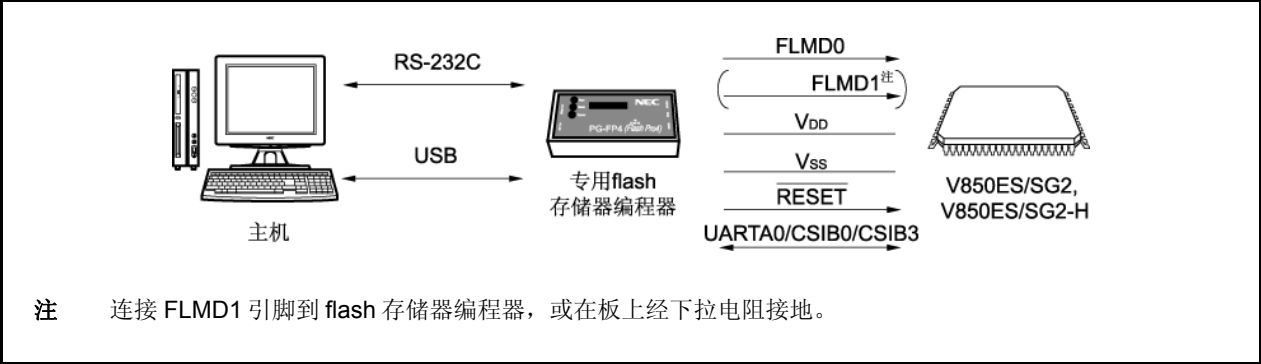
30.4 通过专用Flash编程器重写

<R> 在 V850ES/SG2 或 V850ES/SG2-H 安装到目标系统后（在线编程），可以通过专用 flash 编程器重写 flash 存储器。在设备安装到目标系统前（离线编程），也可以通过专用程序适配器（FA 系列）来重写 flash 存储器。

30.4.1 编程环境

<R> 将程序写入 V850ES/SG2 和 V850ES/SG2-H 的 flash 存储器所需的环境如下所示。

图 30-2. 写程序到 flash 存储器所需要的环境



需要一台主机控制专用 flash 编程器。

在专用 flash 存储器编程器连接 V850ES/SG2 或 V850ES/SG2-H 执行写入、擦除等时，UARTA0，CSIB0 或 CSIB3 作为接口。离线写入需要专用程序适配器（FA 系列）。

备注 FA 系列是 Naito Densei Machida Mfg. Co., Ltd 的产品

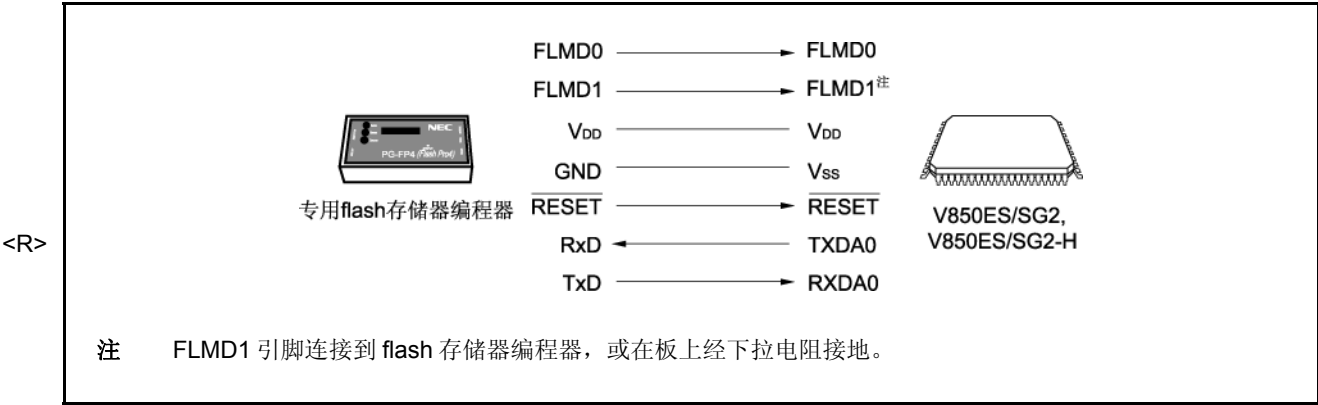
30.4.2 通信模式

<R> V850ES/SG2 或 V850ES/SG2-H 与专用 flash 编程器间的通讯通过 V850ES/SG2, V850ES/SG2-H3 的串行接口 UARTA0, CSIB0 或 CSIB3 完成。

(1) UARTA0

发送速率：9,600 到 153,600 bps

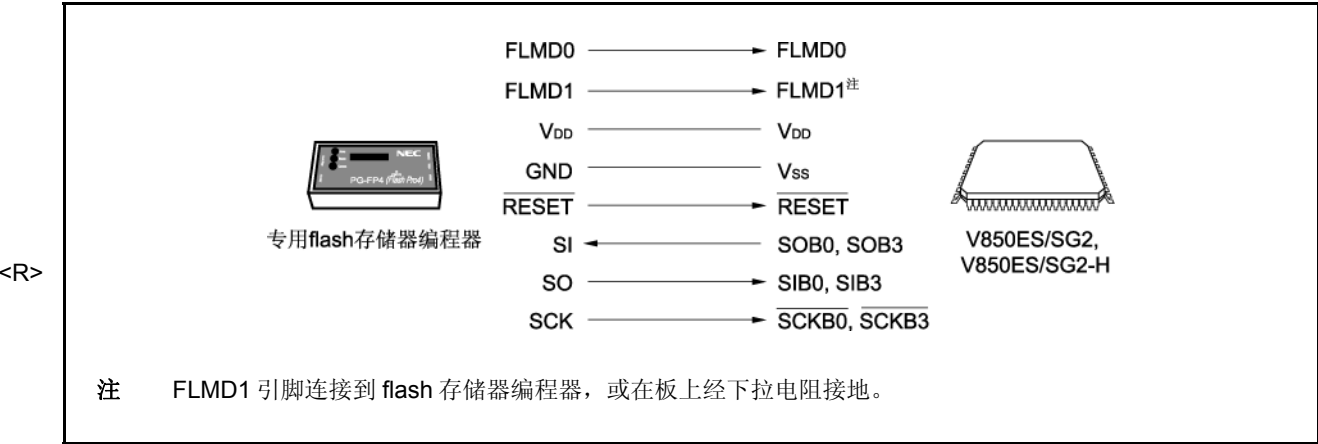
图 30-3. 与专用 flash 存储器编程器通信（UARTA0）



(2) CSIB0, CSIB3

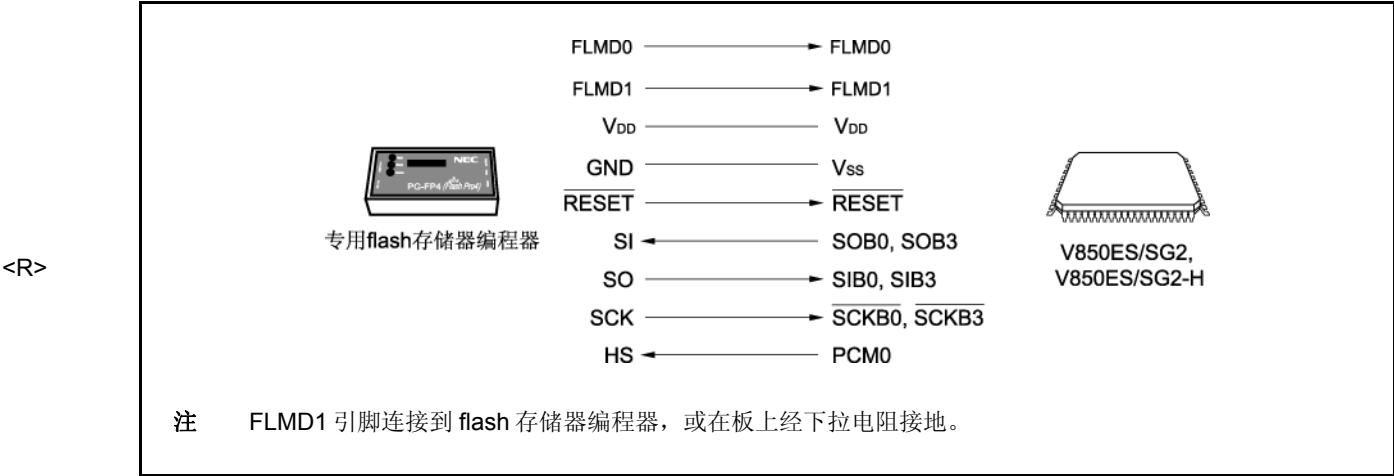
串行时间：2.4 kHz 到 2.5 MHz（MSB first）

图 30-4. 与专用 flash 存储器编程器通信（CSIB0, CSIB3）



(3) CSIB0 + HS, CSIB3 + HS
串行时间：2.4 kHz 到 2.5 MHz（MSB 优先）

图 30-5. 与专用 flash 存储器编程器通信（CSIB0 + HS, CSIB3 + HS）



<R>

<R>

专用 flash 存储器编程器输出传送时钟，V850ES/SG2 和 V850ES/SG2-H 作为从设备操作。

当 PG-FP4 或 PG-FP5 用作专用 flash 存储器编程器时，产生如下信号到 V850ES/SG2 和 V850ES/SG2-H。更多细节，参见 PG-FP4 用户手册（U15260E）和 PG-FP5 用户手册（U18865E）。

<R> 图 30-5. 专用 flash 存储器编程器的信号连接（PG-FP4, PG-FP5）

PG-FP4, PG-FP5			V850ES/SG2, V850ES/SG2-H	连接处理		
信号名	I/O	引脚功能	引脚名	UARTA0	CSIB0, CSIB3	CSIB0 + HS, CSIB3 + HS
FLMD0	输出	写入使能 / 禁止	FLMD0	○	○	○
FLMD1	输出	写入使能 / 禁止	FLMD1	○ ^{注 1}	○ ^{注 1}	○ ^{注 1}
VDD	–	VDD 电压产生 / 电压监测	V _{DD}	○	○	○
GND	–	接地	V _{SS}	○	○	○
CLK	输出	输出到 V850ES/SG2, V850ES/SG2-H 的 时钟	X1, X2	× ^{注 2}	× ^{注 2}	× ^{注 2}
$\overline{\text{RESET}}$	输出	复位信号	$\overline{\text{RESET}}$	○	○	○
SI/RxD	输入	接收信号	SOB0, SOB3/ TXDA0	○	○	○
SO/TxD	输出	传输信号	SIB0, SIB3/ RXDA0	○	○	○
SCK	输出	转移时钟	$\overline{\text{SCKB0}}$, $\overline{\text{SCKB3}}$	×	○	○
HS	输入	给 CSIB0 + HS, CSIB3 + HS 握手信号 通信	PCM0	×	×	○

- 注 1. 这些引脚的连接如图 30-6 和图 30-7 所示，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。用板上加装的振荡器来提供时钟。

备注 ○：必须连接。

 ×：不连接。

<R>

表 30-6. V850ES/SG2, V850ES/SG2-H Flash 写入适配器的连接
(FA-100GF-3BA-A, FA-100GC-8EU-A) (1/2)

<R>

<R>

Flash 存储器编程器 (PG-FP4, PG-FP5) 链接引脚			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	
					GF	GC		GF	GC		GF	GC
SI/RxD	输入	接收信号	SI	P41/SOB0/ SCL01	25	23	P41/SOB0/ SCL01	25	23	P30/TXDA0/ SOB4	27	25
SO/TxD	输出	传输信号	SO	P40/SIB0/ SDA01	24	22	P40/SIB0/ SDA01	24	22	P31/RXDA0 /INTP7/SIB4	28	26
SCK	输出	转移时钟	SCK	P42/ $\overline{\text{SCKB0}}$	26	24	P42/ $\overline{\text{SCKB0}}$	26	24	不需要	–	–
CLK	输出	输出到 V850ES/SG2, V850ES/SG2-H 的时钟	X1	不需要	–	–	不需要	–	–	不需要	–	–
			X2	不需要	–	–	不需要	–	–	不需要	–	–
/RESET	输出	复位信号	/RESET	$\overline{\text{RESET}}$	16	14	$\overline{\text{RESET}}$	16	14	$\overline{\text{RESET}}$	16	14
FLMD0	输出	写入电压	FLMD0	FLMD0	10	8	FLMD0	10	8	FLMD0	10	8
FLMD1	输出	写入电压	FLMD1	PLD5/AD5/ FLMD1	78	76	PLD5/AD5/ FLMD1	78	76	PLD5/AD5/ FLMD1	78	76
HS	输入	CSIO + HS通信 的握手信号	RESERVE/ HS	PCM0/ $\overline{\text{WAIT}}$	63	61	不需要	–	–	不需要	–	–
VDD	–	VDD 电压产生 / 电压监测	VDD	V _{DD}	11	9	V _{DD}	11	9	V _{DD}	11	9
				BV _{DD}	72	70	BV _{DD}	72	70	BV _{DD}	72	70
				EV _{DD}	36	34	EV _{DD}	36	34	EV _{DD}	36	34
				AV _{REF0}	3	1	AV _{REF0}	3	1	AV _{REF0}	3	1
				AV _{REF1}	7	5	AV _{REF1}	7	5	AV _{REF1}	7	5
GND	–	接地	GND	V _{SS}	13	11	V _{SS}	13	11	V _{SS}	13	11
				AV _{SS}	4	2	AV _{SS}	4	2	AV _{SS}	4	2
				BV _{SS}	71	69	BV _{SS}	71	69	BV _{SS}	71	69
				EV _{SS}	35	33	EV _{SS}	35	33	EV _{SS}	35	33

- 注意事项 1. 确保 REGC 引脚经 4.7 μF 电容器接地。
2. 不能由 flash 存储器编程器的 CLK 引脚提供时钟。
用板上加装的振荡器来提供时钟。

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

<R>

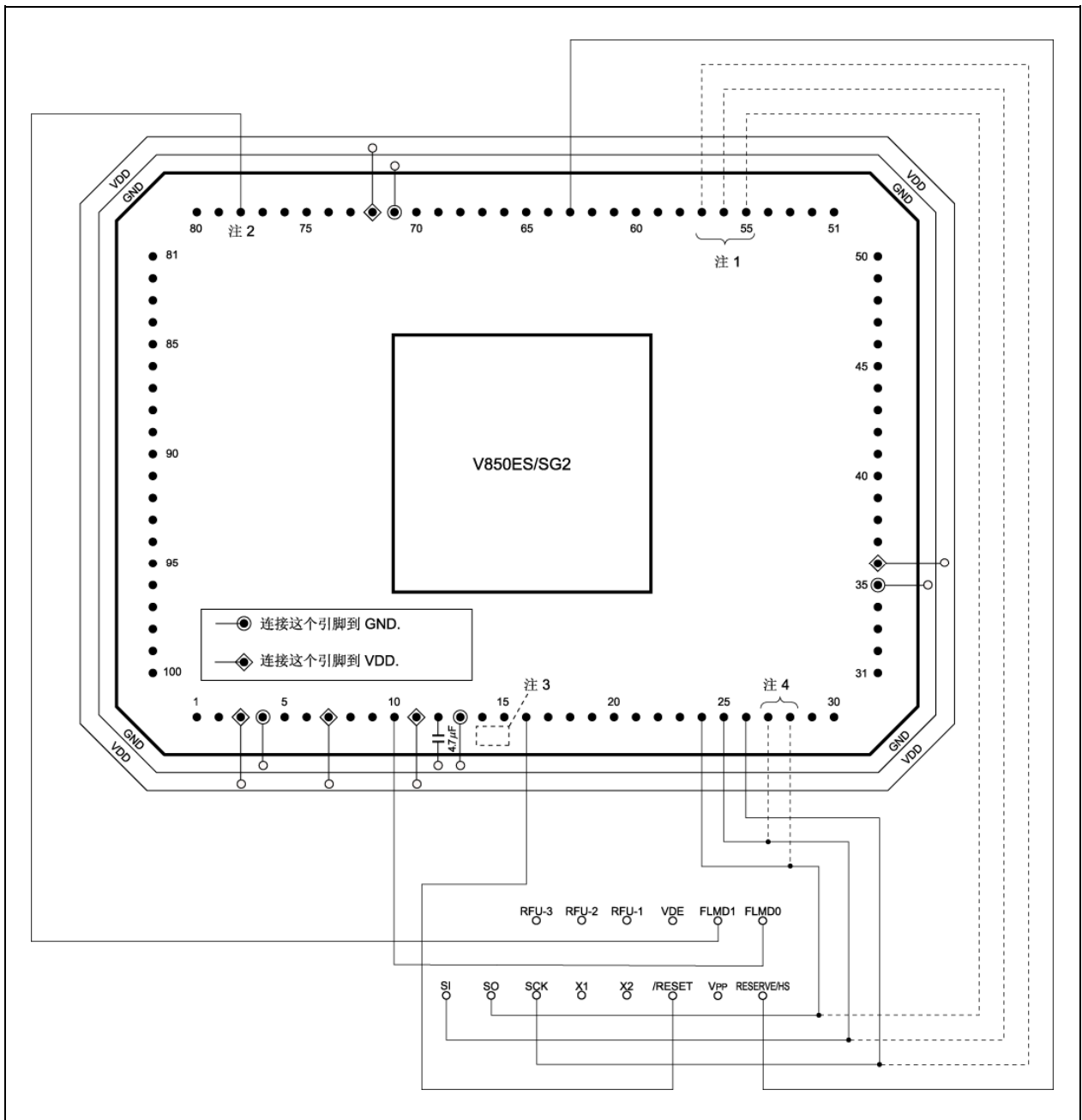
表 30-6. V850ES/SG2, V850ES/SG2-H Flash 写入适配器的连接
(FA-100GF-3BA-A, FA-100GC-8EU-A) (2/2)

<R>	Flash 存储器编程器 (PG-FP4, PG-FP5) 连接引脚			FA板上的引脚 名	当使用CSIB0 + HS时			当使用CSIB3 时		
	信号名	I/O	引脚功能		引脚名	引脚编号		引脚名	引脚编号	
						GF	GC		GF	GC
<R>	SI/RxD	输入	接收信号	SI	P911/A11/SOB3	56	54	P911/A11/SOB3	56	54
	SO/TxD	输出	传输信号	SO	P910/A10/SIB3	55	53	P910/A10/SIB3	55	53
	SCK	输出	转移时钟	SCK	P912/A12/SCKB3	57	55	P912/A12/SCKB3	57	55
	CLK	输出	输出到 V850ES/SG2, V850ES/SG2-H的 时钟	X1	不需要	—	—	不需要	—	—
				X2	不需要	—	—	不需要	—	—
	/RESET	输出	复位信号	/RESET	RESET	16	14	RESET	16	14
	FLMD0	输出	写入电压	FLMD0	FLMD0	10	8	FLMD0	10	8
	FLMD1	输出	写入电压	FLMD1	PLD5/AD5/FLMD1	78	76	PLD5/AD5/FLMD1	78	76
	HS	输入	CSI0 + HS通信的 握手信号	RESERVE/HS	PCM0/WAIT	63	61	不需要	—	—
	VDD	—	VDD 电压产生/电 压监测	VDD	V _{DD}	11	9	V _{DD}	11	9
					BV _{DD}	72	70	BV _{DD}	72	70
					EV _{DD}	36	34	EV _{DD}	36	34
					AV _{REF0}	3	1	AV _{REF0}	3	1
					AV _{REF1}	7	5	AV _{REF1}	7	5
GND	—	接地	GND	V _{SS}	13	11	V _{SS}	13	11	
				AV _{SS}	4	2	AV _{SS}	4	2	
				BV _{SS}	71	69	BV _{SS}	71	69	
				EV _{SS}	35	33	EV _{SS}	35	33	

- 注意事项 1. 确保 REGC 引脚经 4.7 μ F 电容器接地。
2. 不能由 flash 存储器编程器的 CLK 引脚提供时钟。
用板上加装的振荡器来提供时钟。

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

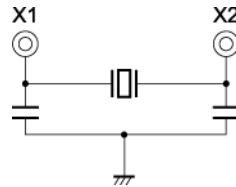
<R> 图 30-6. V850ES/SG2 Flash 写入适配器连接示例 (FA-100GF-3BA-A) (在 CSIB0 + HS 模式下) (1/2)



<R> 图 30-6. V850ES/SG2 Flash 写入适配器连接示例 (FA-100GF-3BA-A) (在 CSIB0 + HS 模式下) (2/2)

- 注
1. 使用 CSIB3 时，对应的引脚。
 2. 按下面所示连接 FLMD1 引脚，或在板上通过下拉电阻将其连接到 GND。
 3. 在 flash 存储器适配器（虚线中所示）上加装一个振荡器来提供时钟。
这是一个振荡器示例。

示例



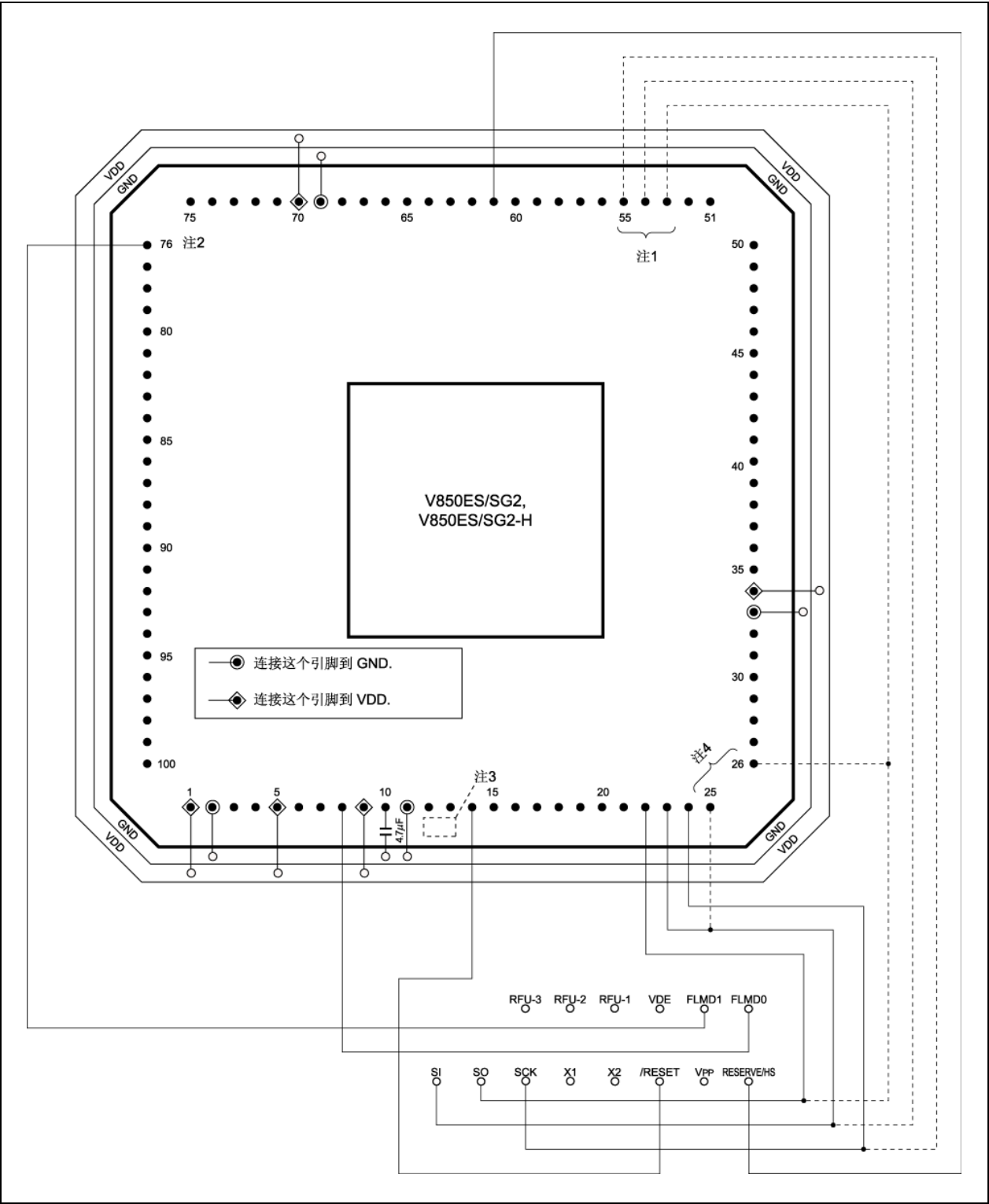
4. 使用 UARTA0 时，对应的引脚。

注意事项 不要输入高电平到 $\overline{\text{DRST}}$ 引脚。

- 备注
1. 没有特意指出的引脚按照未使用引脚的操作处理（参见 2.3 I/O 引脚电路类型，I/O 缓冲供电，和未使用引脚的处理）。
 2. 此适配器为 100 引脚塑封 QFP 封装。

<R> 图 30-7. V850ES/SG2, V850ES/SG2-H Flash 写入适配器连接示例 (FA-100GC-8EU-A) (在 CSIB0 + HS 模式下) (1/2)

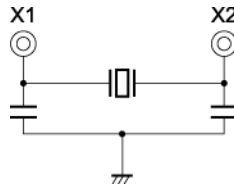
<R>



<R> 图 30-7. V850ES/SG2, V850ES/SG2-H Flash 写入适配器连接示例 (FA-100GC-8EU-A) (在 CSIB0 + HS 模式下) (2/2)

- 注
1. 使用 CSIB3 时, 对应的引脚。
 2. 按下面所示连接 FLMD1 引脚, 或通过下拉电阻将其连接到 GND。
 3. 在 flash 存储器适配器 (虚线中所示) 上加装一个振荡器来提供时钟。这是一个振荡器示例。

示例



4. 使用 UARTA0 时, 对应的引脚。

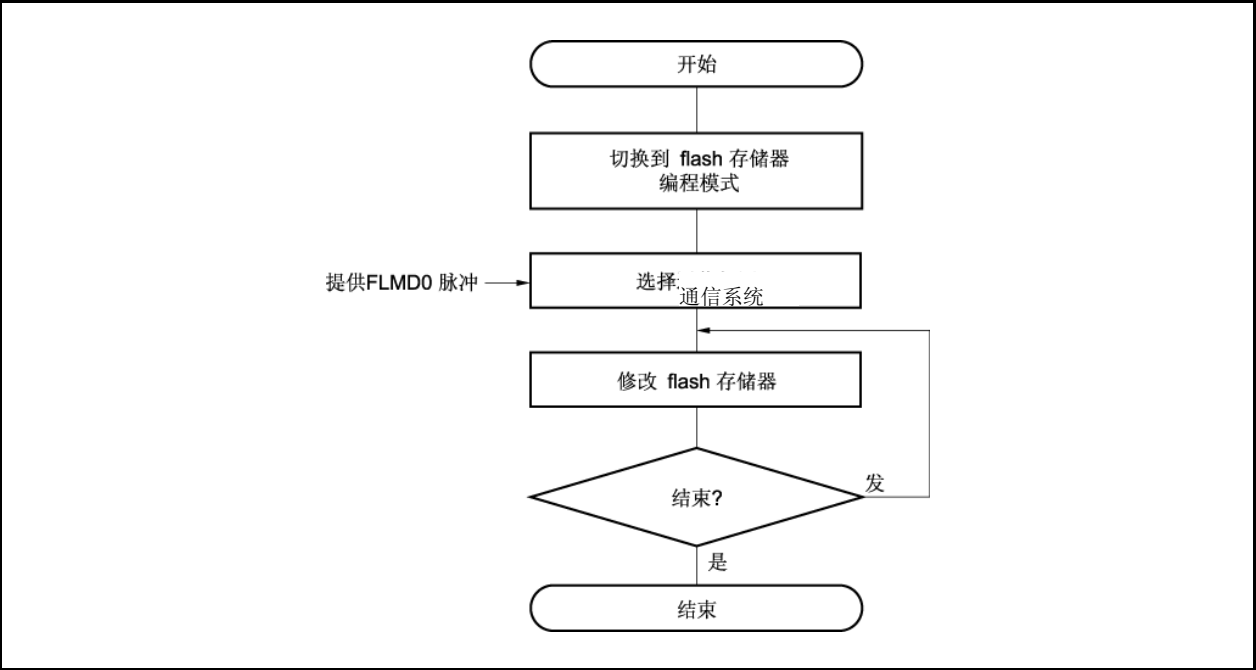
注意事项 不要输入高电平到 $\overline{\text{DRST}}$ 引脚。

- 备注
1. 没有特意指出的引脚按照未使用引脚的操作处理 (参见 2.3 I/O 引脚电路类型, I/O 缓冲供电电平, 和未使用引脚的处理)。
 2. 此适配器为 100 引脚塑封 LQFP 封装。

30.4.3 flash存储器控制

下图显示了 flash 存储器操作过程

图 30-8. flash 存储器操作过程



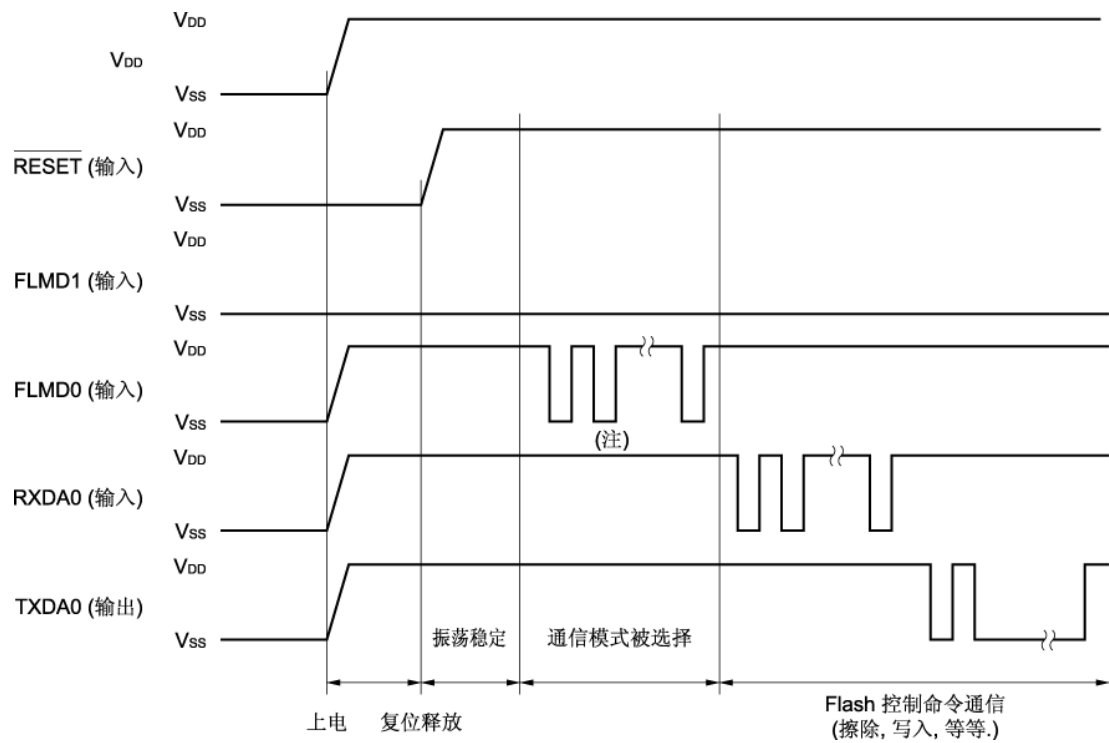
<R>

30.4.4 通讯模式的选择

<R> 在 V850ES/SG2 和 V850ES/SG2-H 中，进入 flash 存储器编程模式后，通过将脉冲（可多达 12 个脉冲）输入到 FLMD0 引脚来选择通信模式。由专用 flash 存储器编程器产生这些 FLMD0 脉冲。

下图显示了脉冲数量与通信模式之间的关系。

图 30-9. 通讯模式的选择



注 如下的时钟个数根据通信模式而不同

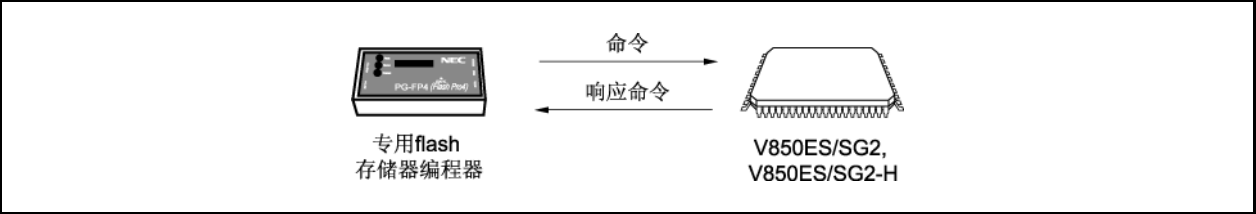
FLMD0 脉冲	通信模式	备注
0	UARTA0	通信速度: 9,600 bps（复位后），LSB 在先
8	CSIB0	V850ES/SG2, V850ES/SG2-H 执行从操作，MSB 优先
9	CSIB3	V850ES/SG2, V850ES/SG2-H 执行从操作，MSB 优先
11	CSIB0 + HS	V850ES/SG2, V850ES/SG2-H 执行从操作，MSB 优先
12	CSIB3 + HS	V850ES/SG2, V850ES/SG2-H 执行从操作，MSB 优先
其他	RFU	设置禁止

注意事项 当选择 UARTA0 时，接收到 FLMD0 脉冲后，根据专用 flash 存储器编程器发出的复位命令计算接收时钟数量。

30.4.5 通信命令

V850ES/SG2 和 V850ES/SG2-H 使用命令方式与专用 Flash 编程器进行通信。由 flash 存储器编程器发往 V850ES/SG2 和 V850ES/SG2-H 的信号称为“命令”，从 V850ES/SG2 和 V850ES/SG2-H 发往专用 flash 存储器编程器的命令称为“响应命令”。

图 30-10. 通信命令



下面所示为在 V850ES/SG2 和 V850ES/SG2-H 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/SG2 和 V850ES/SG2-H 根据这些命令执行处理。

图 30-7. flash 存储器控制命令

类别	命令名称	支持			功能
		CSIB0, CSIB3	CSIB0 + HS, CSIB3 + HS	UARTA0	
空白检测	块空白检测命令	√	√	√	检查存储器指定块的内容是否已经被正确地擦除。
擦除	片擦除命令	√	√	√	擦除整个存储器的内容。
	块擦除命令	√	√	√	擦除存储器指定块的内容。
写入	写入命令	√	√	√	向指定地址范围写入，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	√	√	√	在指定地址范围读取校验和。
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，使能块擦除命令，并且禁止写入命令。

30.4.6 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，板上的连接器也需要可以从正常操作模式切换到 flash 存储器编程模式。

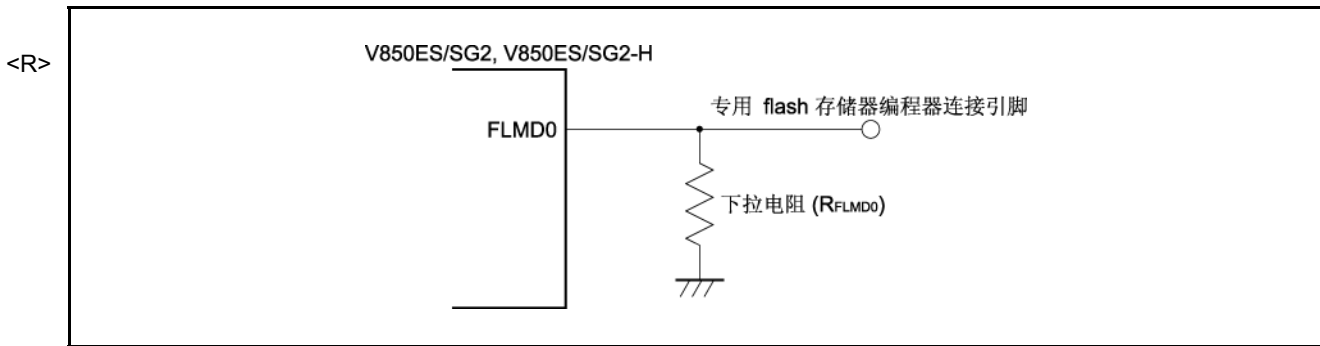
在 flash 存储器编程模式，所有那些和 flash 存储器编程无关的引脚状态都处于复位后的默认状态。因此如果外部设备不能识别复位后的即刻状态，则必须采用如下方式连接引脚。

(1) FLMD0 引脚

在正常工作模式下，输入一个 V_{SS} 电平的电压到 FLMD0 引脚。在 flash 存储器编程模式中，将 V_{DD} 写入电压提供给 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控制端口等供给 FLMD0 引脚。更多细节，参见 30.5.5 (1) FLMD0 引脚。

图 30-11. 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，因此必须输入 0 V 到 FLMD1 引脚。FLMD1 引脚的连接示意图如下所示。

图 30-12. FLMD1 引脚连接示例

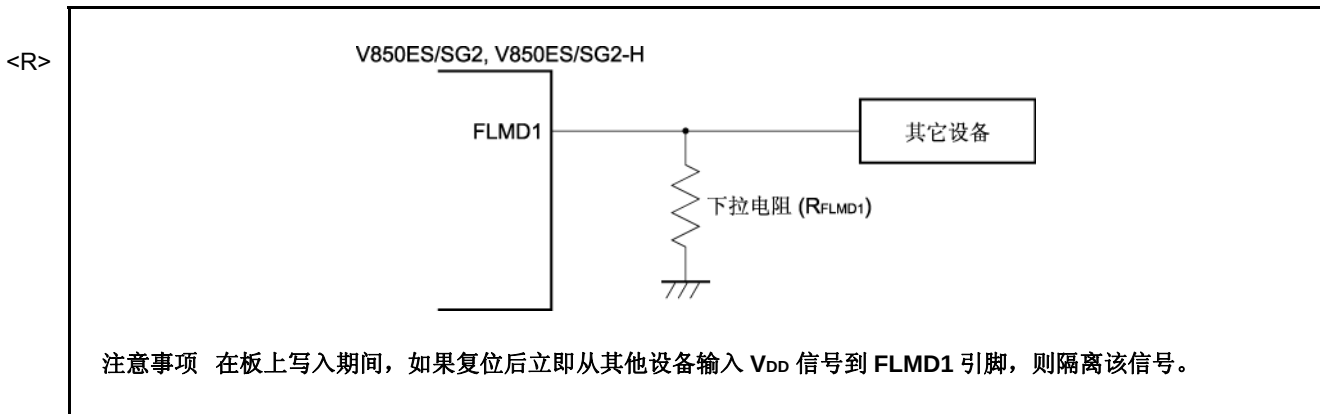


表 30-8. FLMD0 和 FLMD1 引脚的关系和复位释放后的工作模式

FLMD0	FLMD1	工作模式
0	忽略	正常工作模式
V _{DD}	0	flash 存储器编程模式
V _{DD}	V _{DD}	设置禁止

(3) 串行接口引脚

串行接口使用的引脚如下所示。

表 30-9. 串行接口使用的引脚

串行接口	使用引脚
UARTA0	TXDA0, RXDA0
CSIB0	SOB0,SIB0, $\overline{\text{SCKB0}}$
CSIB3	SOB3,SIB3, $\overline{\text{SCKB3}}$
CSIB0 + HS	SOB0,SIB0, $\overline{\text{SCKB0}}$, PCM0
CSIB3 + HS	SOB3,SIB3, $\overline{\text{SCKB3}}$, PCM0

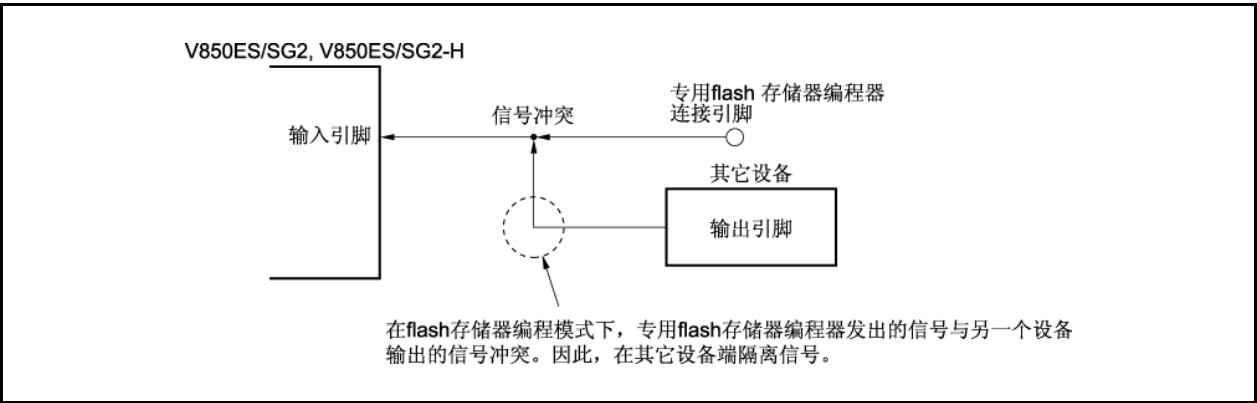
当专用 flash 存储器编程器和串行接口相连时，该串口和板上其它设备连接，必须特别注意以避免信号之间冲突或者其他设备的故障。

(a) 信号冲突

如果专用 Flash 编程器（输出）与串行接口（输入）相连，该串口已连接到另一个设备（输出），这时会产生信号冲突。为了避免这种信号冲突，应隔离与另一个设备的连接，或者设置另一个设备处于高阻抗状态。

图 30-13. 信号冲突（串行接口输入引脚）

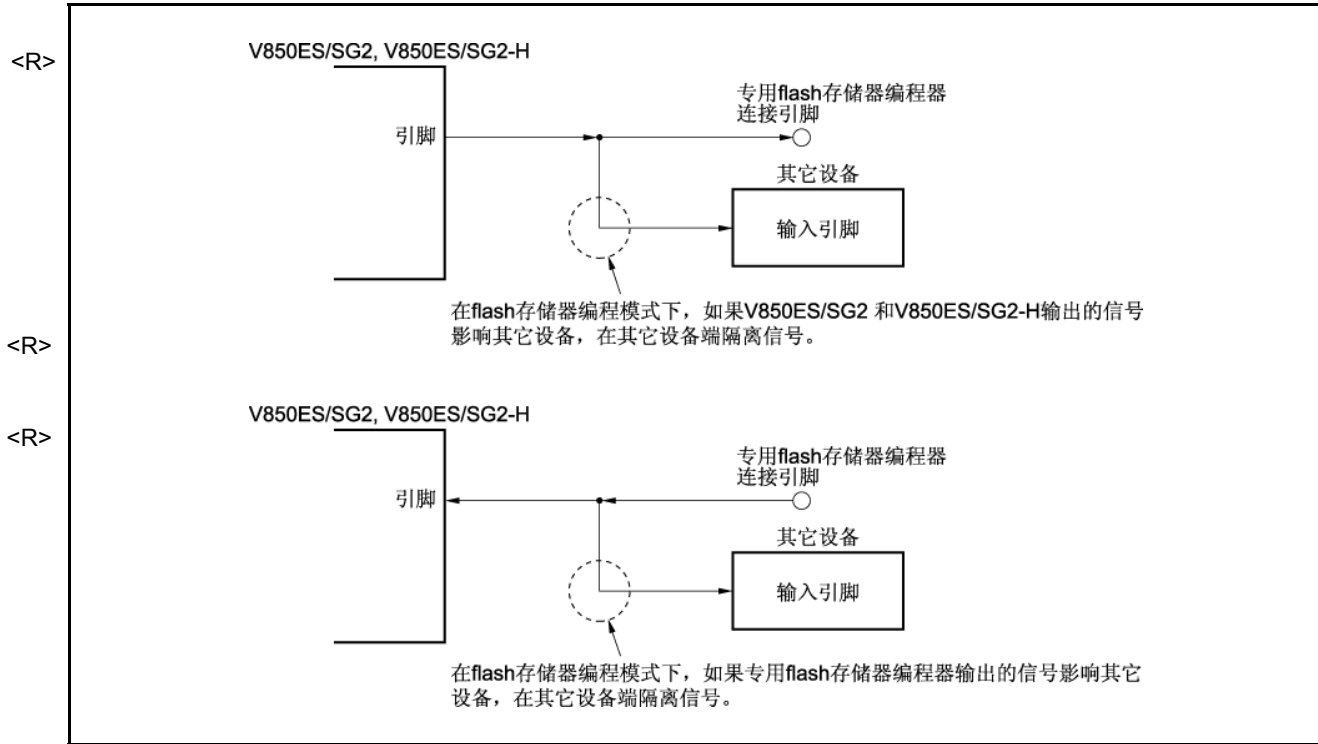
<R>



(b) 其他设备故障

如果专用 flash 存储器编程器（输出或输入）与串行接口（输出或输入）相连，该串口已连接到另一个设备（输入），与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 30-14. 其他设备故障



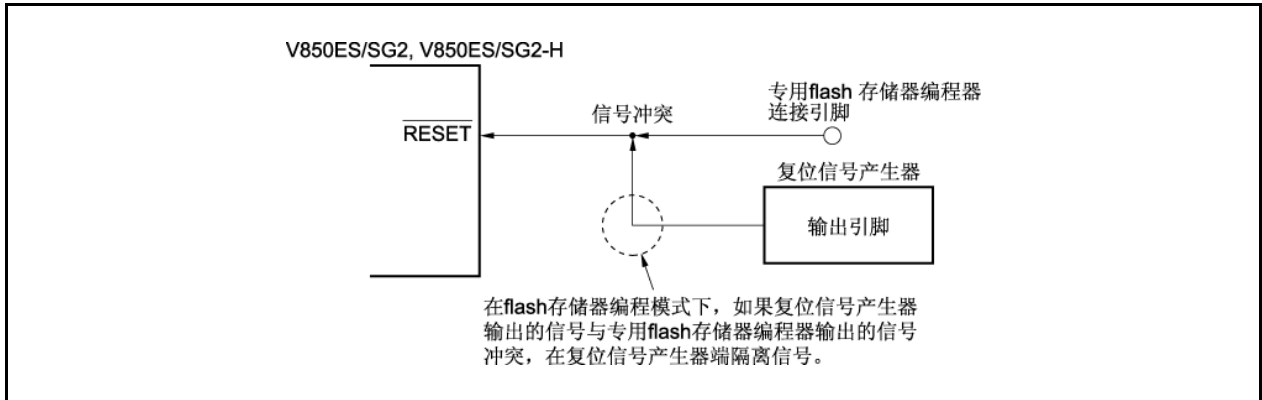
(4) RESET 引脚

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 存储器编程模式下，如果从用户系统输入复位信号，则不能对 flash 存储器进行正确编程。因此除了专复位信号外，不要从 flash 存储器编程器输入其它信号。

图 30-15. 信号冲突 (RESET 引脚)

<R>



(5) 端口引脚 (包括 NMI)

当系统切换到 flash 存储器编程模式时，所有那些和 flash 存储器编程无关的引脚状态都处于复位后的默认状态。因此，如果连接到端口的设备不能识别复位后的端口即刻状态，则引脚需要处理，比如通过一个电阻将端口引脚连接到 V_{DD} 或 V_{SS} 。

(6) 其他信号引脚

连接 X1, X2, XT1, XT2, 并且 REGC 与正常工作模式时相同。

在 flash 存储器编程期间，DRST 引脚输入低电平或开路。不要输入高电平。

(7) 电源

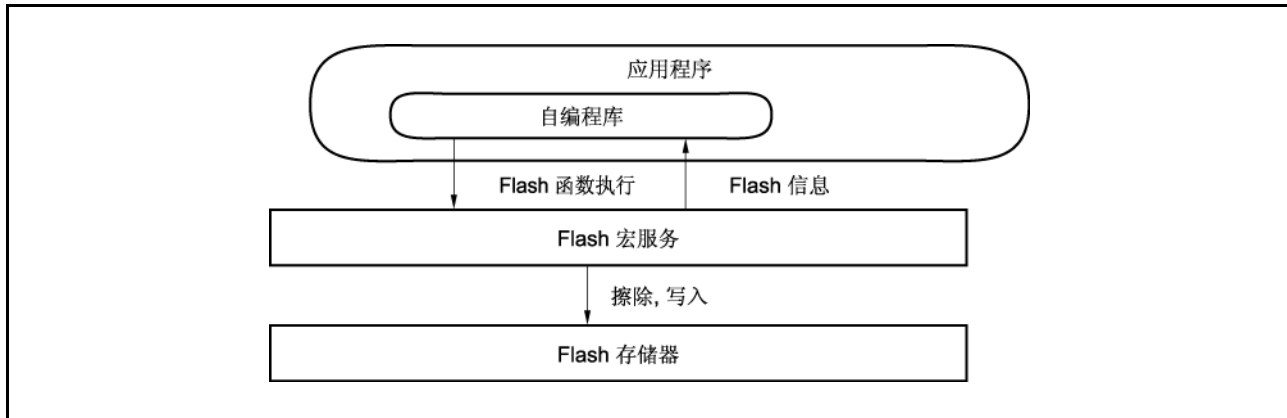
与正常工作模式下电源相同 (V_{DD} , V_{SS} , EV_{DD} , EV_{SS} , BV_{DD} , BV_{SS} , AV_{REF0} , AV_{REF1} , AV_{SS})。

30.5 通过自编程重写

30.5.1 概述

<R> V850ES/SG2 和 V850ES/SG2-H 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帮助用户应用程序实现重写 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。然后，在这个区域内的用户程序可以更新，并且常数也可以重写。

图 30-16. 自编程的概念

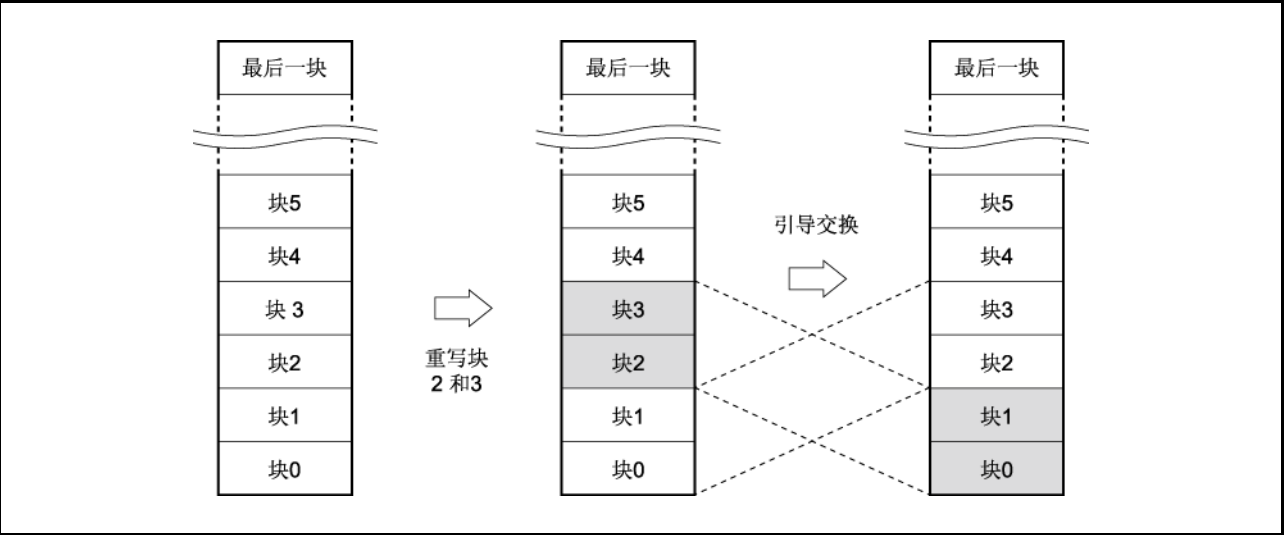


30.5.2 特点

(1) 可靠的自编程（引导交换功能）

<R> V850ES/SG2 和 V850ES/SG2-H 支持引导交换功能，该功能将物理存储器的块 2~3 与块 0~1 进行交换。预先在块 2~3 中写入一个要被重写的启动程序，然后交换物理存储器，因为正确的用户程序总是存在于块 0~1 中，因此即使在重写过程中发生掉电，也可以保证所有区域被安全地重写。

图 30-17. 重写所有存储器区域（引导交换）



<R> (2) 中断支持

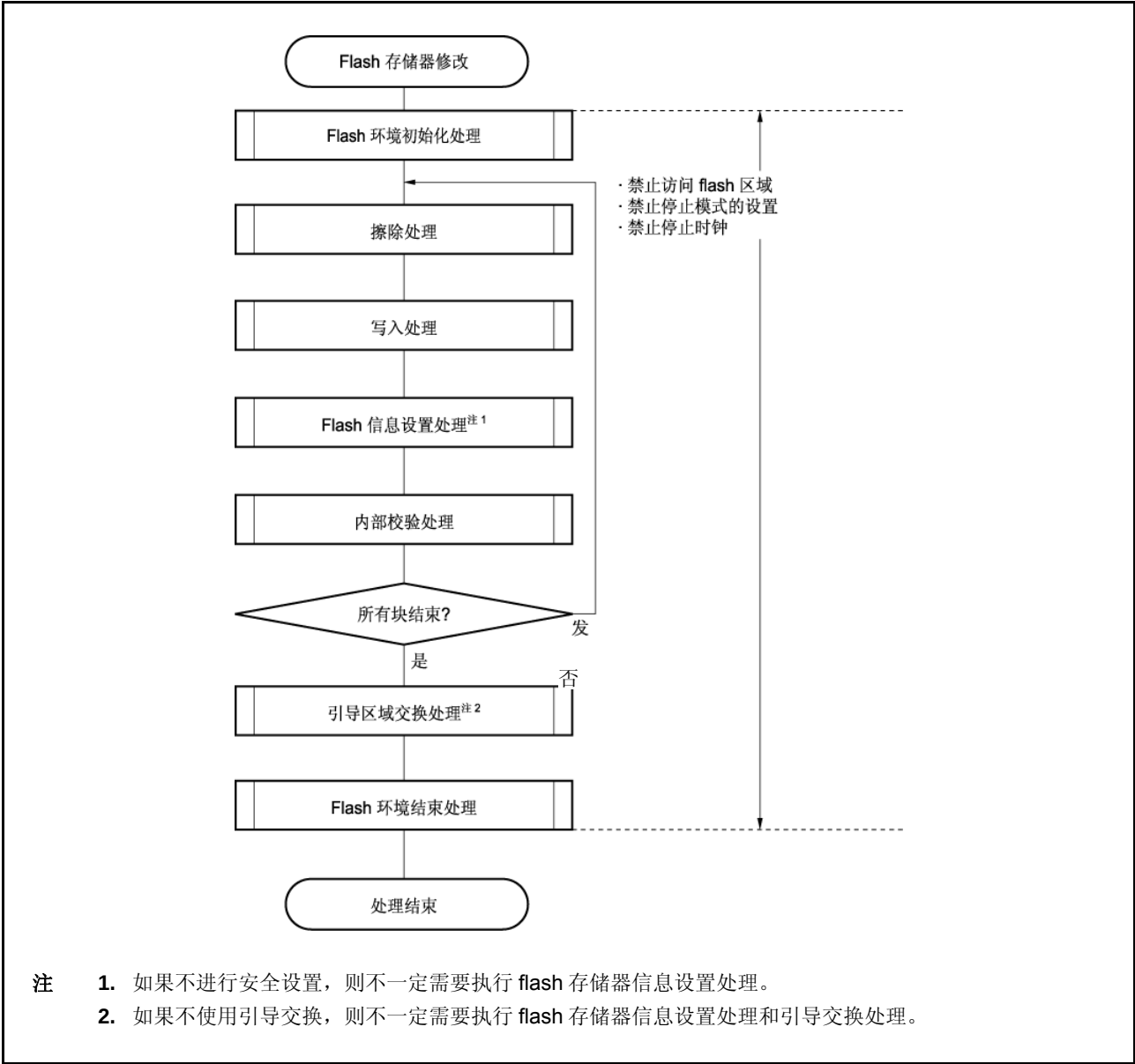
即使在自编程期间执行 flash 功能时，V850ES/SG2 和 V850ES/SG2-H 也可以执行中断服务程序。如果发生可屏蔽中断或不可屏蔽中断，每个中断的服务程序可以被执行。
自编程期间，中断服务程序的两种模式可以使用。

- 简易模式
一个中断句柄可以由 flash 函数 FlashSetUserHandler 注册。可以使用 10 个中断，但是中断注册地越晚，中断句柄的响应越低。，
- 定制模式
中断句柄可以由用户注册，通过在内部 RAM 区域的起始地址处写入中断向量。尽管内部 RAM 的起始地址被占用，但是，根据用户编写的代码，所有需要使用的中断都可以被注册，并且可以控制中断的响应。

30.5.3 标准自编程流程

由 flash 存储器自编程写入 flash 存储器的完整过程如下图所示。

图 30-18. 标准自编程流程



30.5.4 flash存储器功能

表 30-10. flash 存储器功能列表

功能名称	概要	支持
FlashEnv	flash 控制宏的初始化	√
FlashBlockErase	指定的一个块的擦除	√
FlashWordWrite	从指定地址写入	√
FlashBlockIVerify	指定块的内部确认	√
FlashBlockBlankCheck	指定块的空白检查	√
FlashFLMDCheck	FLMD 引脚的检查	√
FlashStatusCheck	对刚指定的操作状态进行检查	√
FlashGetInfo	flash 信息的读取	√
FlashSetInfo	flash 信息的设置	√
FlashBootSwap	引导区域的交换	√
FlashSetUserHandler	用户指定句柄注册函数	√

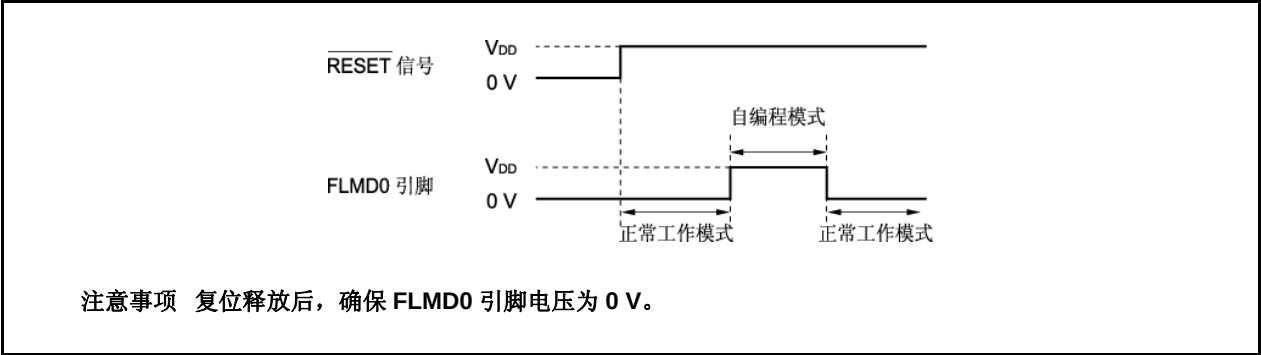
30.5.5 引脚处理

(1) FLMD0 引脚

复位释放后设置工作模式时，以及在自写入时保护 flash 存储器不被写入，需要使用 FLMD0 引脚。因此当复位释放后和正常工作模式时要保持 FLMD0 引脚电压为 0 V。重写存储器前，在自编程模式中需要经由端口控制提供 V_{DD} 电压给 FLMD0 引脚。
自编程完成后，FLMD0 引脚的电压必须返回 0 V。

<R> 注意事项 对应 flash 存储器版本，在复位释放后和用户程序开始前，内部固件支持引导交换功能。
因此，要精确启动用户程序操作，在复位释放后的振荡稳定时间内，确保 FLMD0 引脚接低电平，并且低电平要持续到固件操作完成（V850ES/SG2 的固件操作时间 = 14,974 × (1/f_x) 秒，V850ES/SG2-H 的固件操作时间 = 11,994 × (1/f_x) 秒）。

图 30-19. 模式改变时序



30.5.6 使用的内部资源

下表所列为可用于自编的内部资源。这些内部资源除了自编程也可以自由用于其他用途。

表 30-11. 可用的内部资源

资源名称	描述
RAM 入口区域（内部 RAM / 外部 RAM 的 124 字节）	用于 flash 宏服务的程序和参数位于这个区域中。通过调用库初始化函数，复制入口程序和缺省参数。
堆栈区（用户堆栈 + 300 字节）	用户扩展使用的堆栈，由库使用（内部 RAM 和外部 RAM 均可使用）。
库代码（1,900 字节）	库的程序实体（除了要使用的 flash 存储器块之外，可以用于任何地方）
应用程序	作为用户应用程序执行。 叫做 flash 功能
可屏蔽中断	可以在用户应用执行状态或自编程状态中使用。要在自编程状态中使用这种中断，中断服务程序起始地址必须通过注册函数预先注册。
NMI 中断	可以在用户应用执行状态或自编程状态中使用。要在自编程状态中使用这种中断，中断服务程序开始地址必须通过注册函数预先注册。

第 31 章 片上调试功能

<R> V850ES/SG2 和 V850ES/SG2-H 具有片上调试功能，可以使用 JTAG（联合测试行动小组）接口（ $\overline{\text{DRST}}$ ，DCK，DMS，DDI 和 DDO 引脚）并且可以通过片上调试仿真器（IE-V850E1-CD-NW，QB-V850MINI）来使用。

注意事项 片上调试功能只在 flash 存储器版本中提供，在 Mask ROM 版本中不提供。然而，OCDM 寄存器在掩模 ROM 版本中也存在，并且它控制连接到 P05/INTP2 引脚的下拉电阻，所以即使对掩模 ROM 版本，也需要设置 OCDM 寄存器。

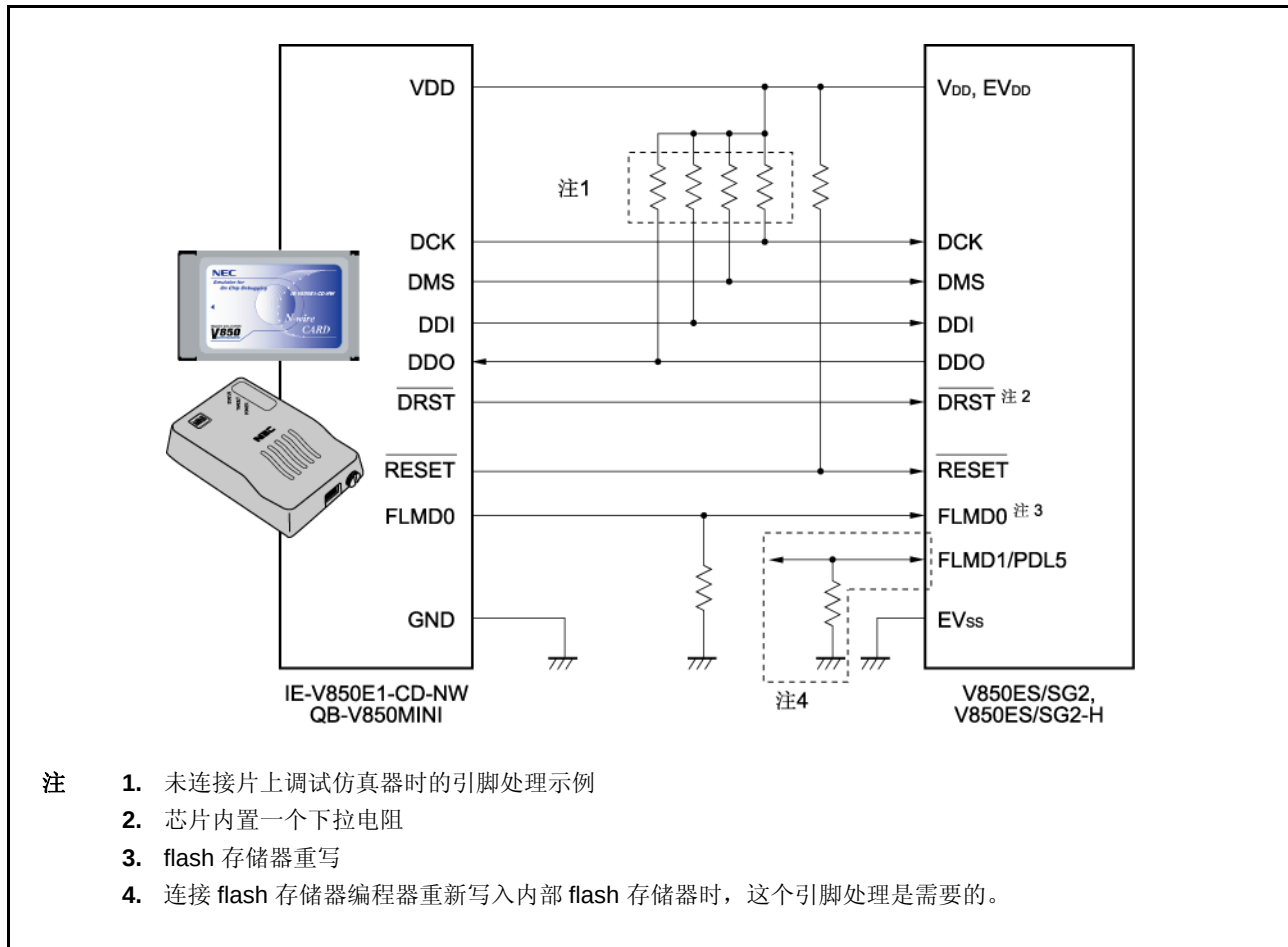
31.1 特性

- 硬件中断功能：2 个
- 软件中断功能：4 个
- 实时 RAM 监视功能：在程序执行期间可读取存储器的内容。
- 动态存储器修改功能（DMM 功能）：在程序执行期间 RAM 内容可以重写。
- 屏蔽功能： $\overline{\text{RESET}}$ ，NMI， $\overline{\text{HLDRQ}}$ ， $\overline{\text{WAIT}}$
- ROM 安全功能：10 字节 ID 代码验证。

注意事项 不支持以下功能

- 跟踪功能
- 事件功能
- 调试中断接口功能（DBINT）

<R> 31.2 连接电路示例



31.3 接口信号

接口信号说明如下。

<R> (1) $\overline{\text{DRST}}$

对于片上调试单元这是一个复位输入信号。它是对调试控制单元进行异步初始化的负逻辑信号。

调试器启动后，当片上调试仿真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进行调试时，通常会产生 CPU 复位。

<R> (2) DCK

这是一个时钟输入信号。它由片上调试仿真器中提供一个 20MHz 或 10MHz 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 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会根据 DMS 信号的电平改变。

(4) DDI

这是一个数据输入信号。在 DCK 上升沿时从片上调试单元采样。

(5) DDO

这是一个数据输出信号。在 DCK 信号下降沿时由片上调试单元输出。

(6) V_{DD}, EV_{DD}

这是一个用于检测目标系统 V_{DD} 的信号。如果目标系统中的 V_{DD} 没有被检测到，那么从片上调试仿真器中输出的信号（ $\overline{\text{DRST}}$, DCK, DMS, DDI, FLMD0 和 $\overline{\text{RESET}}$ ）会进入高阻抗状态。

(7) FLMD0

flash 自编程功能用于通过集成调试器将数据下载到 flash 存储器。在 flash 自编程期间，FLMD0 引脚必须保持为高。另外，对 FLMD0 引脚连接下拉电阻。

FLMD0 引脚也可以用如下两种方式控制。

<1> 由片上调试仿真器控制

将片上调试仿真器中的 FLMD0 信号连接到 FLMD0 引脚。

在正常模式下，片上调试仿真器（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，片上调试仿真器会使 FLMD0 引脚升为高电平。

<2> 由端口控制

连接设备的任意端口到 FLMD0 引脚。

通过用户程序实现 flash 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，执行下载功能前将端口引脚设置为高电平，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（集成调试器）用户操作手册。

(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 位的值被设定无效，将不能执行片上调试功能。因此，reset 功能会受片上调试仿真器的影响，使用 $\overline{\text{RESET}}$ 引脚以使 $\overline{\text{DRST}}$ 引脚有效（初始化）。

31.4 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，并且只能按照特殊顺序才能写入（参见 **3.4.8 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普通端口/外围功能引脚。也可用于断开 P05/INTP2/ $\overline{\text{DRST}}$ 引脚的内部下拉电阻。

仅当 $\overline{\text{DRST}}$ 引脚输入为低电平时，才允许写入 OCDM 寄存器。

该寄存器可用 8 位或 1 位指令读写。

复位后: 01H^注 R/W 地址: FFFFF9FCH

	7	6	5	4	3	2	1	<0>
OCDM	0	0	0	0	0	0	0	OCDM0

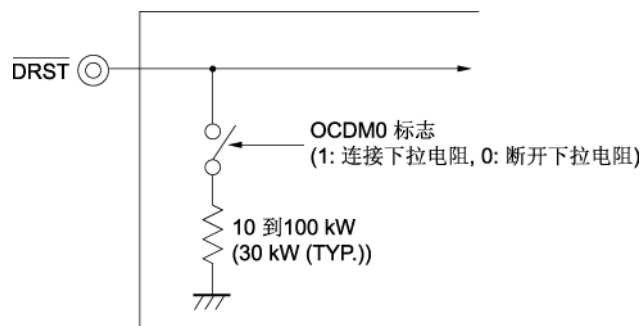
OCDM0	操作模式
0	选择正常操作模式（其中交替作为调试功能的引脚被用作端口/外围功能引脚），并且断开P05/INTP2/DRST 引脚的片上下拉电阻。
1	当 $\overline{\text{DRST}}$ 引脚为低时： 正常操作模式(其中交替作为调试功能的引脚被用作端口/外围功能引脚) 当 $\overline{\text{DRST}}$ 引脚为高时： 片上调试模式(其中交替作为调试功能的引脚被用作片上调试模式引脚)

注 RESET 引脚引发的复位将会设置这个寄存器为 01H。然而在其他原因引发的复位时，OCDM 寄存器的值被保持。其他复位包括看门狗定时器（WDT2RES）引起的复位、低电压检测电路（LVI）引起的复位（LVIRES）（只适用于 V850ES/SG2）或者时钟监视器（CLM）引起的复位（CLMRES）后，

注意事项 1.外部复位后，当 DDI, DDO, DCK, 和 DMS 引脚不用作片上调试引脚而作为端口引脚时，可以进行下面的任何一项操作

- 输入低电压到 P05/INTP2/ $\overline{\text{DRST}}$ 引脚。
- 设置 OCDM0 位。在这种情况下，进行如下操作。
 - <1> OCDM0 位清零。
 - <2> P05/INTP2/ $\overline{\text{DRST}}$ 引脚恒为低电平，直到 <1> 完成。

2. $\overline{\text{DRST}}$ 引脚有内置的下拉电阻。当 OCDM0 标志被清零时，这个电阻被断开。Mask ROM 版本没有片上调试功能，但是它同样具有上述的下拉电阻。因此，同样对于掩模 Mask 版本，内置的下拉电阻必须通过 OCDM0 位清零来断开。



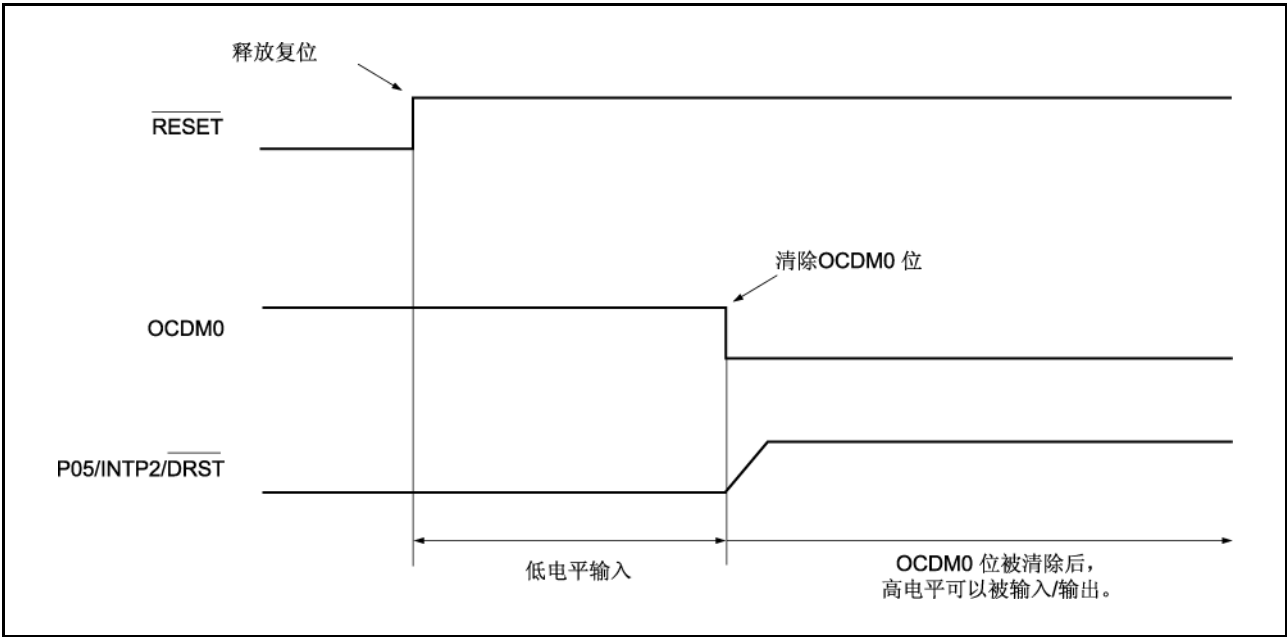
31.5 操作

在满足下表中列出的条件时，片上调试功能无效。
当不使用此功能时，保持 $\overline{\text{DRST}}$ 引脚为低，直到 OCDM.OCDM0 标志被清零。

OCDM0 标志 DRST 引脚	0	1
L	无效	无效
H	无效	有效

备注 L: 低电平输入
 H: 高电平输入

图 31-1. 不使用片上调试功能时的时序



31.6 ROM 安全功能

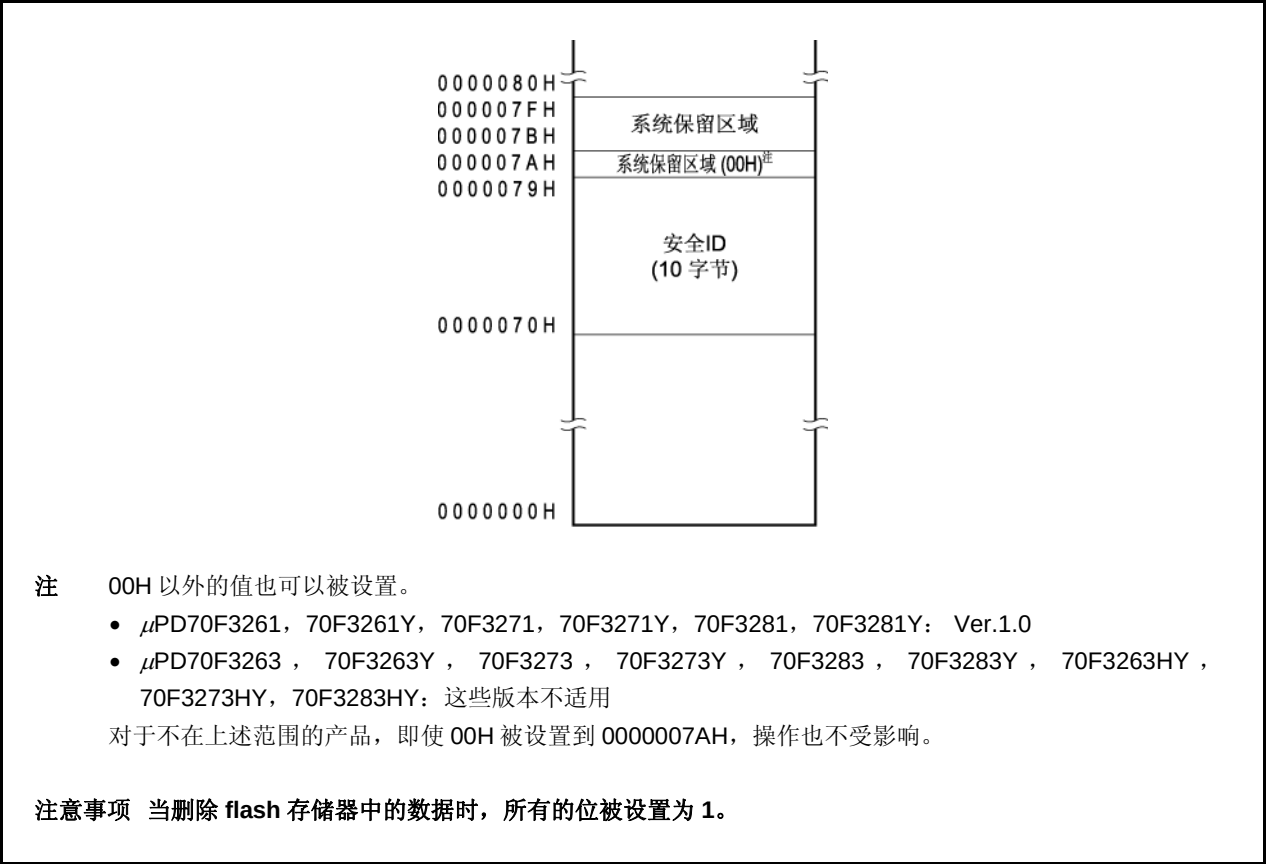
31.6.1 安全ID

V850ES/SG2 和 V850ES/SG2-H 的 flash 存储器版本使用 10 个字节的 ID 码来进行加密，以防在使用通过片上调试仿真器进行片上调试时 flash 存储器的内容被无授权读取。

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 ID 码都匹配，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仿真器。

- 在 0000070H ~ 0000079H 之间设置 10 个字节的 ID 号码。
- 0000079H 的第 7 位是片上调试仿真器的允许标志。（0：禁止，1：使能）
- 当启动片上调试仿真器时，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。



<R>

<R> 31.6.2 设置

示例 当如下的值被设置到地址 0x70 ~ 0x79 时

地址	值
0x70	0x12
0x71	0x34
0x72	0x56
0x73	0x78
0x74	0x9A
0x75	0xBC
0x76	0xDE
0x77	0xF1
0x78	0x23
0x79	0xD4
0x7A	0x00

← 保留代码（参见 3.4.9
(3)）

下面所示为使用 CA850 的程序示例。

[程序示例 1]

接在“LGOP”段（地址 0x60）之后，输入 10 个字节的安全保密密码和 1 个字节的系统保留区域数据（00H）。

```
#-----
# ILGOP handler
#-----
.section "ILGOP" -- Interrupt handler address 0x60
-- Input ILGOP handler code
.org 0x10 -- Skip handler address to 0x70

#-----
# SECURITYID (continue ILGOP handler)
#-----
.word 0x78563412 --0-3 byte code
.word 0xF1DEBC9A --4-7 byte code
.hword 0xD423 --8-9 byte code
.byte 0x00 --Reserve code
```

注意事项 当使用 CA850 Ver. 3.00 或更高的版本时，可指定选项来禁止产生安全保密 ID 号码。

由连接器所产生的安全 ID 附加功能被添加到 CA850 Ver. 3.00 中。结果，在上述程序实例的链接期间产生错误。

错误信息：

F4264: "SECURITY_ID" 段的开始地址（0x00000070）与上一段 "ILGOP" 在地址（0xFFFFFFFF）之前交叠。

[程序示例 2]

使用“SECURITY_ID”部分，输入 10 个字节的安全保密码（地址 0x70）。

```
#-----  
# SECURITY_ID  
#-----  
.section "SECURITY_ID"  
.word      0x78563412 --0-3 byte code  
.word      0xF1DEBC9A --4-7 byte code  
.hword     0xD423     --8-9 byte code
```

注意事项 可以被设置到“SECURITY_ID”段的数据被限制为 10 个字节。由于这个原因，数据不能被设置到安全保密码后的系统保留区域（0x7A）。因此，当使用的设备需要将数据设置到系统保留区域时，使用“程序示例 1”中的方法来设置安全保密码和系统保留区域数据。
关于将数据设置到系统保留区域的设备的细节，参阅 3.4.9（3）系统保留区域。

31.7 注意事项

- (1) RUN（程序执行）期间，如果输入复位信号（从目标系统或从内部复位源来的复位信号），中断功能就可能出现故障。
- (2) 即使复位信号通过屏蔽功能被屏蔽，如果复位信号是从引脚输入的话，I/O 缓冲器（端口引脚）也可能复位。
- (3) 因为内部 flash 存储器中的设置软件断点是通过 ROM 修正功能实现的，目标复位或看门狗定时器 2 产生的内部复位导致断点暂时无效。当硬件中断或强制中断发生时，断点再次有效，但是在那之前，软件中断不会中断。
- (4) 中断期间屏蔽引脚复位，CPU 和 外围 I/O 不复位。用户程序执行的时候，如果 flash 存储器由 DMM（动态存储器修改）重写或者由 RAM 监视功能读取，同时产生引脚复位或内部复位，那么 CPU 和外围 I/O 可能不被正确复位。
- (5) 不能执行 ROM 修正的仿真。

<R> (6) 当满足以下情况 (a) 和 (b) 并且由于中断使仿真器（QB-V850ESSX2，IE-703288-G1-EM1，IE-V850E1-CD-NW，QB-V850MINI）的操作被停止时，看门狗定时器 2 并不停止，并且发生复位或不可屏蔽中断。当复位发生时，调试器挂起。

- (a) 主时钟或副时钟被用作看门狗定时器 2 的源时钟。
- (b) 内部振荡时钟被停止（RCM.RSTOP 位 = 1）。

要避免上述情况，执行下面任意一条。

- 使用仿真器时，源时钟使用内部振荡时钟。
- 使用仿真器时，不要停止内部振荡器。

<R> (7) 当满足以下情况 (a) 和 (b) 并且由于中断使仿真器（QB-V850ESSX2，IE-703288-G1-EM1，IE-V850E1-CD-NW，QB-V850MINI）的操作被停止时，即使外围中断功能被设置为“Break”，TMM 也不停止。

- (a) TMM 源时钟可以选择 INTWT、内部振荡时钟（ $f_R/8$ ）或副时钟。
- (b) 主时钟被停止。

要避免上述情况，执行下面任意一条。

- 使用仿真器时，源时钟使用主时钟（ f_{xx} ， $f_{xx}/2$ ， $f_{xx}/4$ ， $f_{xx}/64$ ， $f_{xx}/512$ ）。
- 使用仿真器时，不停止主时钟振荡。

- (8) 在片上调试模式下，DDO 引脚被强制设置为高电平输出。

第 32 章 电气规范

32.1 绝对最大额定值

(T_A = 25°C) (1/2)

参数	符号	条件	额定值	单位
电源电压	V _{DD}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	−0.5 到 +4.6	V
	BV _{DD}		−0.5 到+4.6	V
	EV _{DD}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	−0.5 到+4.6	V
	AV _{REF0}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	−0.5 到+4.6	V
	AV _{REF1}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	−0.5 到+4.6	V
	V _{SS}	V _{SS} = EV _{SS} = BV _{SS} = AV _{SS}	−0.5 到+0.5	V
	AV _{SS}	V _{SS} = EV _{SS} = BV _{SS} = AV _{SS}	−0.5 到+0.5	V
	BV _{SS}	V _{SS} = EV _{SS} = BV _{SS} = AV _{SS}	−0.5 到+0.5	V
	EV _{SS}	V _{SS} = EV _{SS} = BV _{SS} = AV _{SS}	−0.5 到+0.5	V
输入电压	V _{I1}	$\overline{\text{RESET}}$, FLMD0 ^{註1} , PDH4, PDH5	−0.5 到EV _{DD} + 0.5 ^{註2}	V
	V _{I2}	PCM0 到PCM3, PCT0, PCT1, PCT4, PCT6, PDH0 到PDH3, PDL0 到PDL15	−0.5 到 BV _{DD} + 0.5 ^{註2}	V
	V _{I3}	P10, P11	−0.5 到AV _{REF1} + 0.5 ^{註2}	V
	V _{I4}	X1, X2, XT1, XT2	−0.5 到V _{RO} ^{註3} + 0.5 ^{註2}	V
	V _{I5}	P02 到P06, P30 到P39, P40 到P42, P50 到P55, P90 到P915	−0.5 到 +6.0	V
模拟输入电压	V _{IAN}	P70 到P711	−0.5 到AV _{REF0} + 0.5 ^{註2}	V

- 注
1. 只适用于 Flash 存储器版本
 2. 确保不要超过每个电源电压的绝对最大额定值（最大值）。
 3. 片上稳压器输出电压（2.5 V（典型值））

(T_A = 25°C) (2/2)

参数	符号	条件		额定值	单位
输出电流，低	I _{OL}	P02 到P06, P30 到P39, P40 到P42, P50 到P55, P90 到P915, PDH4, PDH5	每个引脚	4	mA
			所有引脚总和	50	mA
		PCM0 到PCM3, PCT0, PCT1, PCT4, PCT6, PDH0 到PDH3, PDL0 到PDL15	每个引脚	4	mA
			所有引脚总和	50	mA
		P10, P11	每个引脚	4	mA
			所有引脚总和	8	mA
		P70 to P711	每个引脚	4	mA
			所有引脚总和	20	mA
输出电流，高	I _{OH}	P02 到P06, P30 到P39, P40 到P42, P50 到P55, P90 到P915, PDH4, PDH5	每个引脚	−4	mA
			所有引脚总和	−50	mA
		PCM0 到PCM3, PCT0, PCT1, PCT4, PCT6, PDH0 到PDH3, PDL0 到PDL15	每个引脚	−4	mA
			所有引脚总和	−50	mA
		P10, P11	每个引脚	−4	mA
			所有引脚总和	−8	mA
		P70 to P711	每个引脚	−4	mA
			所有引脚总和	−20	mA
工作环境温度	T _A			−40 到 +85	°C
存储温度	T _{stg}	Mask ROM版本		−65 到 +150	°C
		Flash 存储器版本		−40 到 +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，或将输出连接到 V_{DD}、V_{CC}，和 GND。但是，漏极开路引脚或集电极开路引脚，可以直接互相连接。如果输出引脚设置为高阻抗状态并且避免与外部电路的输出时序冲突，可以直接将 IC 产品的输出引脚与外部电路连接起来。
 2. 如果有任何一项参数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 特性所示的额定值和条件代表正常操作的质量保证范围。

备注 除非特别指出，复用功能引脚的特性与端口引脚特性相同。

32.2 电容

<R>

(TA = 25°C, VDD = EVDD = BVDD = AVREF0 = AVREF1 = VSS = EVSS = BVSS = AVSS = 0 V)

参数	符号	条件	最小值	典型值	最大值	单位
I/O 电容	C _{IO}	f _x = 1 MHz 未测量的引脚降为 0 V			10	pF

32.3 工作条件

(1) V850ES/SG2

(TA = -40 到 +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V)

内部系统时钟频率	条件	电源电压				单位
		V _{DD}	EV _{DD}	BV _{DD}	AV _{REF0} , AV _{REF1}	
f _{xx} = 2.5 到 20 MHz	C = 4.7 μF A/D 转换停止 D/A 转换停止	2.85 到 3.6	2.85 到 3.6	2.7 到 3.6	2.85 到 3.6	V
	C = 4.7 μF A/D 转换停止 D/A 转换停止	3.0 到 3.6	3.0 到 3.6	2.7 到 3.6	3.0 到 3.6	V
f _{xt} = 32.768 kHz	C = 4.7 μF A/D 转换停止 D/A 转换停止	2.85 到 3.6	2.85 到 3.6	2.7 到 3.6	2.85 到 3.6	V

<R>

(2) V850ES/SG2-H

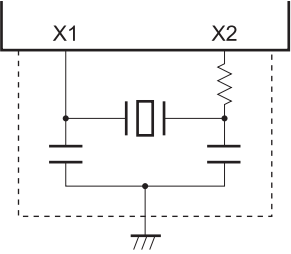
(TA = -40 到 +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V)

内部系统时钟频率	条件	电源电压					单位
		V _{DD}	EV _{DD}	BV _{DD}	AV _{REF0}	AV _{REF1}	
f _{xx} = 2.5 到 32 MHz	稳定电容 C = 4.7 μF 连接到 REGC 引脚, 所有功能工作	3.0 到 3.6	3.0 到 3.6	3.0 到 3.6	3.0 到 3.6	3.0 到 3.6	V
f _{xt} = 32.768 kHz							

32.4 振荡器特性

32.4.1 主时钟振荡器特性

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

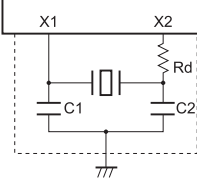
谐振器	电路示例	参数	条件	最小值	典型值	最大值	单位
<R> 陶瓷谐振器 / 晶体谐振器		振荡频率 (f_x) ^{註1}	V850ES/SG2	2.5		10	MHz
			V850ES/SG2-H	2.5		8	MHz
		振荡稳定时间 ^{註2}	复位释放后		$2^{16}/f_x$		s
			STOP 模式释放后	$1^{\text{註4}}$	注 3		ms
			IDLE2 模式释放后	$350^{\text{註4}}$	注 3		μs

- 注 1. 上面所示的振荡频率仅为振荡器特性。因此使用 V850ES/SG2 的内部操作条件不要超过 AC 特性和 DC 特中所示的额定值。
2. 从振荡开始到谐振稳定的时间。
3. 该值根据 OSTs 寄存器的设置而改变。
4. Flash 存储器启动所需要的时间（仅限 flash 存储器版本）。使用 OSTs 寄存器来保留启动时间。

- 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。
 - 不要将电容的接地点设在大电流接地处。
 - 不要从振荡器获取信号。
2. 当主时钟停止且设备运行于副时钟时，则在切换回主时钟之前，要由程序保证振荡稳定时间的等待。

<R>

(i) KYOCERA KINSEKI CORPORATION: 晶体谐振器 ($T_A = -40$ 到 $+85^{\circ}\text{C}$)

生产商 (零件号码)	电路示例	振荡频率 f_x (kHz)	推荐电路常数			振荡电压范围	
			C1 (pF)	C2 (pF)	Rd (k Ω)	最小值 (V)	最大值 (V)
KYOCERA KINSEKI CORPORATION - CX-5FD (电容: 8 pF) - CX-49G (电容: 8 pF) - HC-49/U-S (电容: 8 pF) 关于其他谐振器的类型名称, 请参考谐振器生产厂商。		4,000	8	8	—	2.85	3.6
		5,000	8	8	—	2.85	3.6
		8,000	8	8	—	2.85	3.6
		10,000	8	8	—	2.85	3.6
		3,145.72	8	8	—	2.85	3.6
		4,718.592	8	8	—	2.85	3.6
		6,291.456	8	8	—	2.85	3.6

注意事项 该振荡器常数由振荡器生产商提供, 是在指定环境下评估得出的参考值。

如果在实际应用中需要优化振荡器特性, 请向振荡器生产商申请在实现电路上评价。

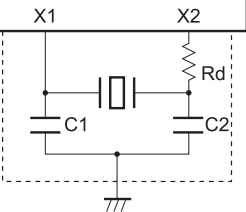
振荡器电压和频率仅为振荡器特性。使用 V850ES/SG2 和 V850ES/SG2-H 时, 内部工作条件不能超出 DC 特性和 AC 特性中规定的额定值范围。

备注 联系方法:

KYOCERA Electronic 器件和设备 <http://global.kyocera.com/prdct/electro/index.html>

振荡器和 IC 的匹配搜索 <http://www3.kyocera.co.jp/electro/app/en/searchTopShow.do>

<R> (ii) Murata Mfg. Co. Ltd.: 陶瓷谐振器 ($T_A = -40$ 到 $+85^\circ\text{C}$)

生产商	电路示例	振荡频率 f_x (MHz)	零件号码	推荐电路常数			振荡电压范围	
				C1 (pF)	C2 (pF)	Rd (k Ω)	最小值 (V)	最大值 (V)
Murata Mfg. Co. Ltd.		4.000	CSTCR4M00G55B-R0	片上 (39)	片上 (39)	0	2.85	3.6
			CSTCR4M00G15C**-R0	片上 (39)	片上 (39)	0	2.85	3.6
		5.000	CSTCR5M00G55B-R0	片上 (39)	片上 (39)	0	2.85	3.6
			CSTCR5M00G15C**-R0	片上 (39)	片上 (39)	0	2.85	3.6
		6.000	CSTCR6M00G55B-R0	片上 (39)	片上 (39)	0	2.85	3.6
			CSTCR6M00G15C**-R0	片上 (39)	片上 (39)	0	2.85	3.6
		8.000	CSTCE8M00G55A-R0	片上 (33)	片上 (33)	0	2.85	3.6
			CSTCE8M00G15C**-R0	片上 (33)	片上 (33)	0	2.85	3.6
		10.000	CSTCE10M0G55A-R0	片上 (33)	片上 (33)	0	2.85	3.6
			CSTCE10M0G15C**-R0	片上 (33)	片上 (33)	0	2.85	3.6

注意事项 该振荡器常数由振荡器生产商提供，是在指定环境下评估得出的参考值。

如果在实际应用中需要优化振荡器特性，请向振荡器生产商申请在实现电路上评价。

振荡器电压和频率仅为振荡器特性。使用 V850ES/SG2 和 V850ES/SG2-H 时，内部工作条件不能超出 DC 特性和 AC 特性中规定的额定值范围。

备注 1. 在产品编号中带“**”的产品的总容限可以调整到 ± 3000 ppm。

2. 联系方式：

Product Engineering Service Section
Piezoelectric Components Department I
Piezoelectric Components Division
Device Business Unit
Murata Mfg. Co., Ltd.
TEL: +81-75-955-6915

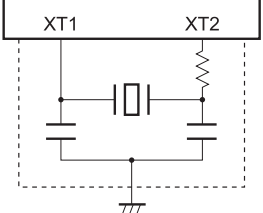
E-mail: piezo@murata.co.jp

IC 产品编号->陶瓷谐振器 搜索

http://search.murata.co.jp/Ceramy/CeMenu_en.do

32.4.2 副时钟振荡器特性

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

谐振器	电路示例	参数	条件	最小值	典型值	最大值	单位
晶体谐振器		振荡频率 (f_{XT}) ^{註1}		32	32.768	35	kHz
		振荡稳定时间 ^{註2}				10	s

注 1. 上面所示的振荡频率仅为振荡器特性。因此使用 V850ES/SG2 和 V850ES/SG2-H 内部工作条件不要超过 AC 特性和 DC 特中所示的额定值。

2. 从 V_{DD} 达到以下振荡电压范围到晶体谐振器稳定所需要的时间。

- V850ES/SG2: 2.85 V (最小值)
- V850ES/SG2-H: 3.0 V (最小值)

<R>

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。
- 不要将电容的接地点设在大电流接地处。
- 不要从振荡器获取信号。

2. 为了降低功耗，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需要注意布线方法。

3. 关于谐振器的选择和振荡器常数，请客户自行评价或向谐振器生产商申请评价。

32.4.3 PLL特性

(1) V850ES/SG2

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
输入频率	f_x	×4模式	2.5		5	MHz
		×8模式	2.5		2.5	MHz
输出频率	f_{xx}	×4模式	10		20	MHz
		×8模式	20		20	MHz
入锁时间	t_{PLL}	在 V_{DD} 达到 2.85 V (最小值) 后			800	μs

<R> (2) V850ES/SG2-H

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
输入频率	f_x	×4模式	2.5		5	MHz
		×8模式	2.5		4	MHz
输出频率	f_{xx}	×4模式	10		20	MHz
		×8模式	20		32	MHz
入锁时间	t_{PLL}	在 V_{DD} 达到 3.0 V (最小值) 后			800	μs

32.4.4 内部振荡器特性

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
输出频率	f_R		100	200	400	kHz

32.5 稳压器特性

(1) V850ES/SG2

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

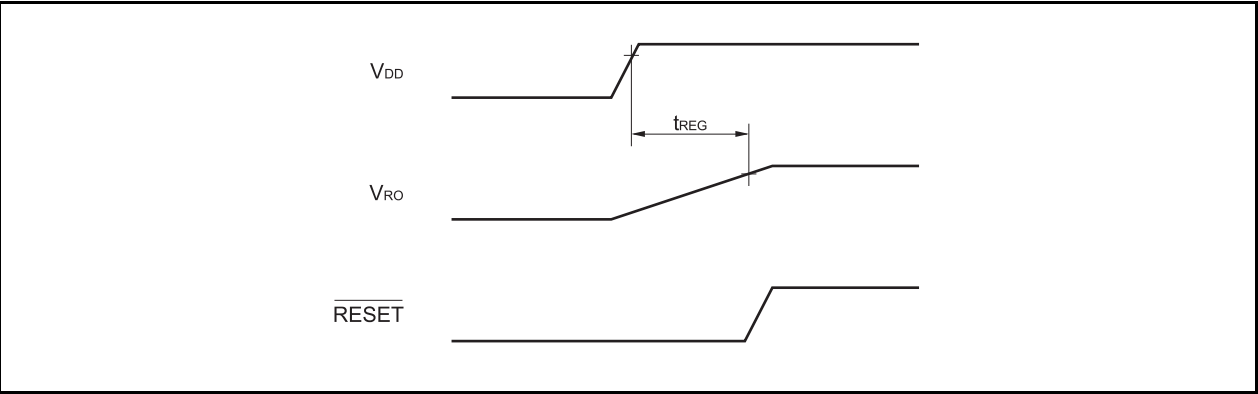
参数	符号	条件	最小值	典型值	最大值	单位
输入电压	V_{DD}	$f_{xx} = 20\text{ MHz}$ (最大值)	2.85		3.6	V
输出电压	V_{RO}			2.5		V
稳压器输出稳定时间	t_{REG}	V_{DD} 达到 2.85 V (最小值) 后, 连接 $C = 4.7\text{ }\mu\text{F}$ 的稳定电容到 REGC 引脚			1	ms

<R>

(2) V850ES/SG2-H

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
输入电压	V_{DD}	$f_{xx} = 32\text{ MHz}$ (最大值)	3.0		3.6	V
输出电压	V_{RO}			2.5		V
稳压器输出稳定时间	t_{REG}	V_{DD} 达到 3.0 V (最小值) 后 连接 $C = 4.7\text{ }\mu\text{F}$ 的稳定电容到 REGC 引脚			1	ms



32.6 DC特性

32.6.1 I/O 电平

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
输入电压, 高	V_{IH1}	PDH4, PDH5	$0.7EV_{DD}$		EV_{DD}	V
	V_{IH2}	RESET, FLMD0 [‡]	$0.8EV_{DD}$		EV_{DD}	V
	V_{IH3}	P02 到 P06, P30 到 P37, P42, P50 到 P55, P92 到 P915	$0.8EV_{DD}$		5.5	V
	V_{IH4}	P38, P39, P40, P41, P90, P91	$0.7EV_{DD}$		5.5	V
	V_{IH5}	PCM0 到 PCM3, PCT0, PCT1, PCT4, PCT6, PDH0 到 PDH3, PDL0 到 PDL15	$0.7BV_{DD}$		BV_{DD}	V
	V_{IH6}	P70 到 P711	$0.7AV_{REF0}$		AV_{REF0}	V
	V_{IH7}	P10, P11	$0.7AV_{REF1}$		AV_{REF1}	V
输入电压, 低	V_{IL1}	PDH4, PDH5	EV_{SS}		$0.3EV_{DD}$	V
	V_{IL2}	RESET, FLMD0 [‡]	EV_{SS}		$0.2EV_{DD}$	V
	V_{IL3}	P02 到 P06, P30 到 P37, P42, P50 到 P55, P92 到 P915	EV_{SS}		$0.2EV_{DD}$	V
	V_{IL4}	P38, P39, P40, P41, P90, P91	EV_{SS}		$0.3EV_{DD}$	V
	V_{IL5}	PCM0 到 PCM3, PCT0, PCT1, PCT4, PCT6, PDH0 到 PDH3, PDL0 到 PDL15	BV_{SS}		$0.3BV_{DD}$	V
	V_{IL6}	P70 到 P711	AV_{SS}		$0.3AV_{REF0}$	V
	V_{IL7}	P10, P11	AV_{SS}		$0.3AV_{REF1}$	V
输入漏电流, 高	I_{LIH}	$V_i = V_{DD} = EV_{DD} = BV_{DD} = AV_{REF0} = AV_{REF1}$			5	μA
输入漏电流, 低	I_{LIL}	$V_i = 0\text{ V}$			-5	μA
输出漏电流, 高	I_{LOH}	$V_o = V_{DD} = EV_{DD} = BV_{DD} = AV_{REF0} = AV_{REF1}$			5	μA
输出漏电流, 低	I_{LOL}	$V_o = 0\text{ V}$			-5	μA

注 仅限 Flash 存储器版本

备注 除非特别指出, 复用功能引脚的特性与端口引脚特性相同。

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件			最小值	典型值	最大值	单位
输出电压, 高	V_{OH1}	P02 到 P06, P30 到 P39, P40 到 P42, P50 到 P55, P90 到 P915, PDH4, PDH5	每个引脚 $I_{OH} = -1.0\text{ mA}$	所有引脚总和 -20 mA	$EV_{DD} - 1.0$		EV_{DD}	V
			每个引脚 $I_{OH} = -100\text{ }\mu\text{A}$	所有引脚总和 -4.2 mA	$EV_{DD} - 0.5$		EV_{DD}	V
	V_{OH2}	PCM0 到 PCM3, PCT0, PCT1, PCT4, PCT6, PDH0 到 PDH3, PDL0 到 PDL15	每个引脚 $I_{OH} = -1.0\text{ mA}$	所有引脚总和 -20 mA	$BV_{DD} - 1.0$		BV_{DD}	V
			每个引脚 $I_{OH} = -100\text{ }\mu\text{A}$	所有引脚总和 -2.8 mA	$BV_{DD} - 0.5$		BV_{DD}	V
	V_{OH3}	P70 到 P711	每个引脚 $I_{OH} = -0.4\text{ mA}$	所有引脚总和 -4.8 mA	$AV_{REF0} - 1.0$		AV_{REF0}	V
			每个引脚 $I_{OH} = -100\text{ }\mu\text{A}$	所有引脚总和 -1.2 mA	$AV_{REF0} - 0.5$		AV_{REF0}	V
	V_{OH4}	P10, P11	每个引脚 $I_{OH} = -0.4\text{ mA}$	所有引脚总和 -0.8 mA	$AV_{REF1} - 1.0$		AV_{REF1}	V
			每个引脚 $I_{OH} = -100\text{ }\mu\text{A}$	所有引脚总和 -0.2 mA	$AV_{REF1} - 0.5$		AV_{REF1}	V
	V_{OL1}	P02 到 P06, P30 到 P37, P42, P50 到 P55, P92 到 P915, PDH4, PDH5	每个引脚 $I_{OL} = 1.0\text{ mA}$	所有引脚总和 20 mA	0		0.4	V
					0		0.4	V
					0		0.4	V
					0		0.4	V
输出电压, 低	V_{OL2}	P38, P39, P40, P41, P90, P91	每个引脚 $I_{OL} = 3.0\text{ mA}$		0		0.4	V
	V_{OL3}	PCM0到 PCM3,PCT0 , PCT1, PCT4, PCT6,PDH0 到 PDH3, PDL0 到 PDL15	每个引脚 $I_{OL} = 1.0\text{ mA}$	所有引脚总和 20 mA	0		0.4	V
	V_{OL4}	P10, P11, P70 到 P711	每个引脚 $I_{OL} = 0.4\text{ mA}$	所有引脚总和 5.6 mA	0		0.4	V
	V_{OL5}	P10, P11, P70 到 P711	每个引脚 $I_{OL} = 0.4\text{ mA}$	所有引脚总和 5.6 mA	0		0.4	V
软件下拉电阻	R_1	P05	$V_I = V_{DD}$		10	30	100	k Ω

备注

1. 除非特别指出, 复用功能引脚的特性与端口引脚特性相同。
2. 当某引脚的 I_{OH} 和 I_{OL} 条件不满足, 但所有引脚的总和值满足时, 仅该引脚不满足 DC 特性。

32.6.2 供电电流

(1) V850ES/SG2

(TA = -40 到 +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V)

参数	符号	条件		最小值	典型值	最大值	单位	
供电电流 ^{注1} (掩模 ROM版本)	IDD1	正常操作	f _{xx} = 20 MHz (f _x = 5 MHz)， 周边功能工作		21	32	mA	
	IDD2	HALT 模式	f _{xx} = 20 MHz (f _x = 5 MHz)， 周边功能工作		15	24	mA	
	IDD3	IDLE1 模式	f _{xx} = 5 MHz (f _x = 5 MHz)，PLL 关		0.3	0.8	mA	
	IDD4	IDLE2 模式	f _{xx} = 5 MHz (f _x = 5 MHz)，PLL 关		0.3	0.8	mA	
	IDD5	副时钟操作模式	f _{XT} = 32.768 kHz， 主时钟，内部振荡器停止		50	100	μA	
	IDD6	Sub-IDLE 模式	f _{XT} = 32.768 kHz， 主时钟，内部振荡器停止		15	70	μA	
	IDD7	STOP 模式	副时钟停止，内部振荡器停止		6	50	μA	
			副时钟工作，内部振荡器停止		10	60	μA	
			副时钟停止，内部振荡器工作		10	60	μA	
供电电流 ^{注1} (flash 存储器版本)	IDD1	正常操作	f _{xx} = 20 MHz (f _x = 5 MHz)， 周边功能工作	注 2		32	48	mA
				注 3		30	45	mA
	IDD2	HALT 模式	f _{xx} = 20 MHz (f _x = 5 MHz)， 周边功能工作	注 2		17	26	mA
				注 3		16	24	mA
	IDD3	IDLE1 模式	f _{xx} = 5 MHz (f _x = 5 MHz)，PLL 关		0.8	1.6	mA	
	IDD4	IDLE2 模式	f _{xx} = 5 MHz (f _x = 5 MHz)，PLL 关		0.3	0.8	mA	
	IDD5	副时钟操作模式	f _{XT} = 32.768 kHz， 主时钟，内部振荡器停止	注 2		300	600	μA
				注 3		200	400	μA
	IDD6	Sub-IDLE 模式	f _{XT} = 32.768 kHz， 主时钟，内部振荡器停止	注 2		18	100	μA
				注 3		18	80	μA
	IDD7	STOP 模式	副时钟停止，内部振荡器停止		6	50	μA	
			副时钟工作，内部振荡器停止		10	60	μA	
			副时钟停止，内部振荡器工作		10	60	μA	
	IDD8	Flash 存储器编程模式	f _{xx} = 20 MHz (f _x = 5 MHz)	注 2		35	54	mA
				注 3		33	51	mA

- 注 1. VDD, EVDD 和 BVDD 电流总和。流过输出缓冲器、A/D 转换器、D/A 转换器、片上下拉电阻的电流不包括在内。
2. 640 KB flash 存储器版本: μPD70F3263, 70F3263Y, 70F3273, 70F3273Y, 70F3283, 70F3283Y
3. 384 KB flash 存储器版本: μPD70F3261, 70F3261Y, 70F3271, 70F3271Y, 70F3281, 70F3281Y

<R>

(2) V850ES/SG2-H

(T_A = -40 到 +85°C, BV_{DD} ≤ V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

参数	符号	条件		最小值	典型值	最大值	单位
供电电流 [✱] (掩模 ROM版本)	I _{DD1}	正常操作	f _{xx} = 32 MHz （f _x = 4 MHz）， 周边功能工作		35	55	mA
	I _{DD2}	HALT 模式	f _{xx} = 32 MHz （f _x = 4 MHz）， 周边功能工作		27	40	mA
	I _{DD3}	IDLE1 模式	f _{xx} = 4 MHz （f _x = 4 MHz）， PLL 关		0.3	0.8	mA
	I _{DD4}	IDLE2 模式	f _{xx} = 4 MHz （f _x = 4 MHz）， PLL 关		0.3	0.8	mA
	I _{DD5}	副时钟操作模式	f _{XT} = 32.768 kHz， 主时钟，内部振荡器停止		50	100	μA
	I _{DD6}	Sub-IDLE 模式	f _{XT} = 32.768 kHz， 主时钟，内部振荡器停止		15	70	μA
	I _{DD7}	STOP 模式	副时钟停止，内部振荡器停止		6	50	μA
			副时钟工作，内部振荡器停止		10	60	μA
			副时钟停止，内部振荡器工作		10	60	μA
供电电流 [✱] (flash 存储器版本)	I _{DD1}	正常操作	f _{xx} = 32 MHz （f _x = 4 MHz）， 周边功能工作		50	70	mA
	I _{DD2}	HALT 模式	f _{xx} = 32 MHz （f _x = 4 MHz）， 周边功能工作		30	40	mA
	I _{DD3}	IDLE1 模式	f _{xx} = 4 MHz （f _x = 4 MHz）， PLL 关		0.8	1.6	mA
	I _{DD4}	IDLE2 模式	f _{xx} = 4 MHz （f _x = 4 MHz）， PLL 关		0.3	0.8	mA
	I _{DD5}	副时钟操作模式	f _{XT} = 32.768 kHz， 主时钟，内部振荡器停止		300	600	μA
	I _{DD6}	Sub-IDLE 模式	f _{XT} = 32.768 kHz， 主时钟，内部振荡器停止		18	100	μA
	I _{DD7}	STOP 模式	副时钟停止，内部振荡器停止		7	50	μA
			副时钟工作，内部振荡器停止		10	60	μA
			副时钟停止，内部振荡器工作		10	60	μA
	I _{DD8}	Flash 存储器编程 模式	f _{xx} = 32 MHz （f _x = 4 MHz）		55	80	mA

注 V_{DD}, EV_{DD} 和 BV_{DD} 电流总和。流过输出缓冲器、A/D 转换器、D/A 转换器、片上下拉电阻的电流不包括在内。

32.7 数据保持特性

(1) 在 STOP 模式下

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

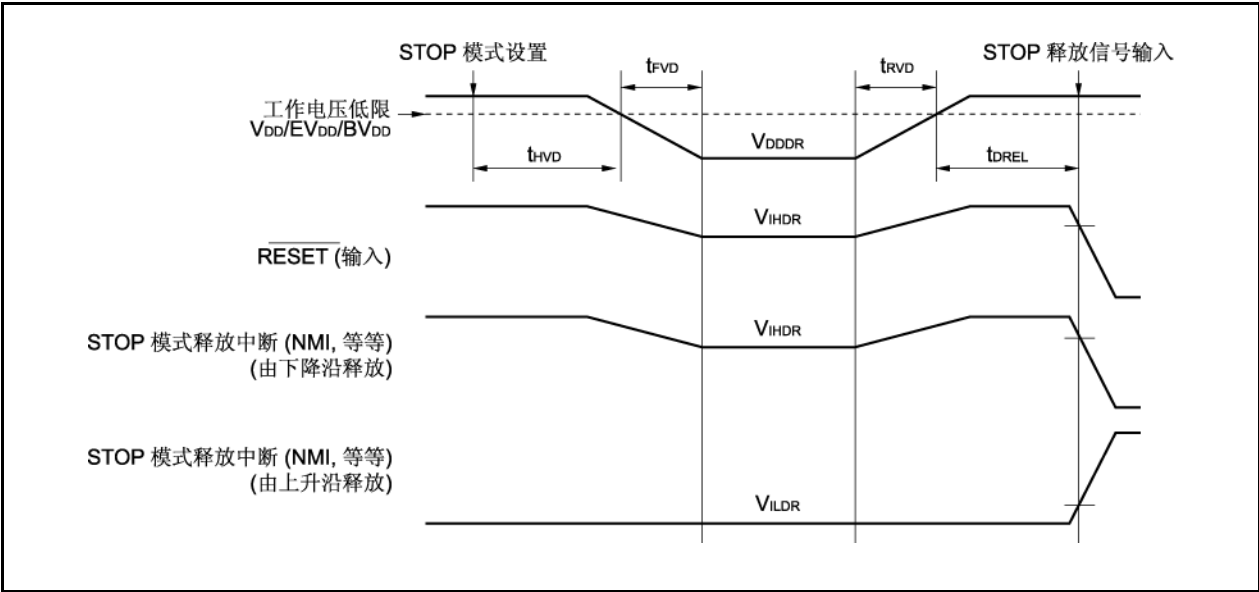
<R>

参数	符号	条件	最小值	典型值	最大值	单位
数据保持电压	V_{DDDR}	STOP 模式 (所有功能停止)	1.9		3.6	V
数据保持电流	I_{DDDR}	STOP 模式 (所有功能停止)		6	50	μA
电源电压上升时间	t_{rVD}		200			μs
电源电压下降时间	t_{fVD}		200			μs
电源电压保持时间	t_{hVD}	STOP 模式设置后	0			ms
STOP 释放信号输入时间	t_{DREL}	注	0			ms
数据保持输入电压, 高	V_{IHDR}	$V_{DD} = EV_{DD} = BV_{DD} = V_{DDDR}$	$0.9V_{DDDR}$		V_{DDDR}	V
数据保持输入电压, 低	V_{ILDR}	$V_{DD} = EV_{DD} = BV_{DD} = V_{DDDR}$	0		$0.1V_{DDDR}$	V

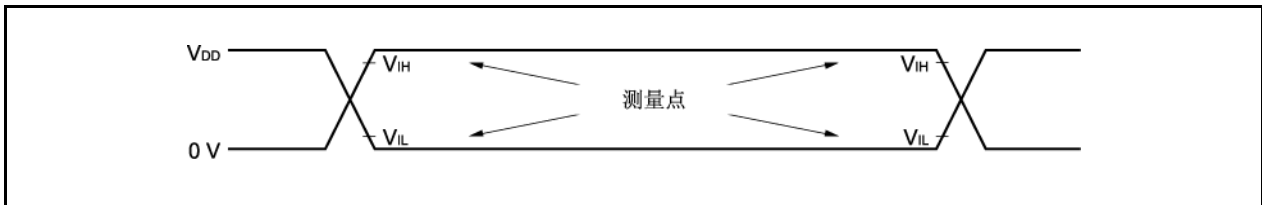
注 V850ES/SG2: 在 V_{DD} 达到 2.85 V (最小值) 后
V850ES/SG2-H: 在 V_{DD} 达到 3.0 V (最小值) 后

<R>

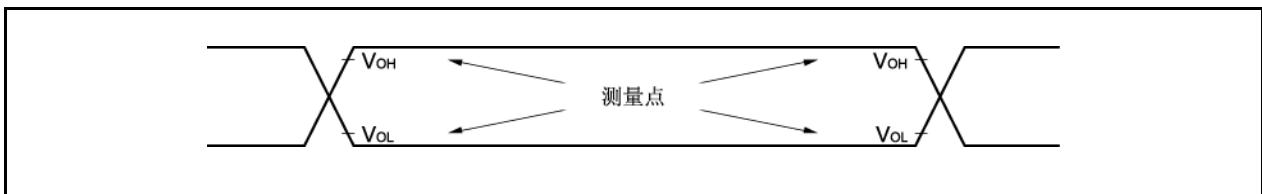
注意事项 转换到 STOP 模式和从 STOP 模式还原必须在额定的工作范围完成。



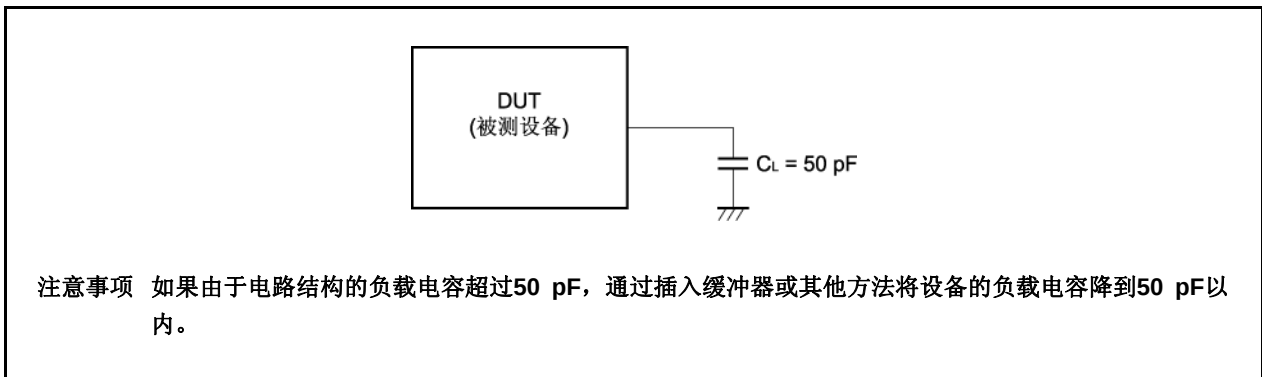
32.8 AC特性

(1) AC 输入测试点 (V_{DD} , AV_{REF0} , AV_{REF1} , EV_{DD} , BV_{DD})

(2) AC 输出测试点



(3) 负载条件



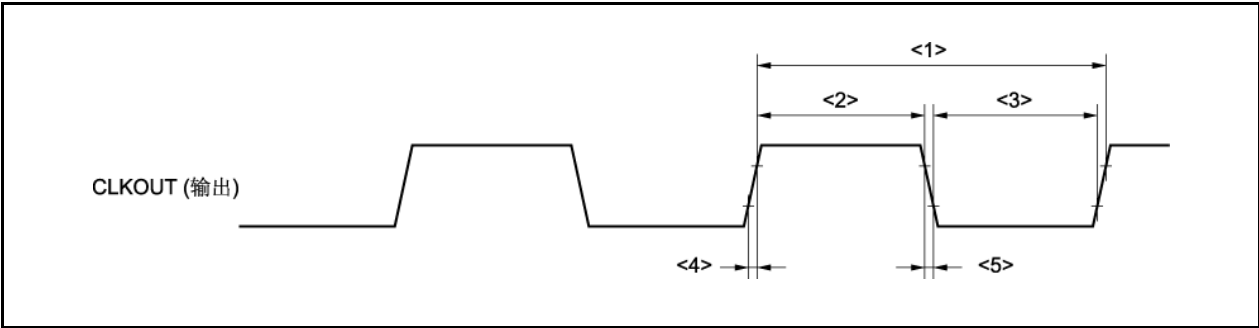
32.8.1 CLKOUT 输出顺序

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$)

<R>

参数	符号	条件	最小值	最大值	单位
输出周期	t_{CYK}	<1> V850ES/SG2	50 ns	31.25 μs	
		V850ES/SG2-H	31.25 ns	31.25 μs	
高电平宽度	t_{WKH}	<2>	$t_{CYK}/2 - 10$		ns
低电平宽度	t_{WKL}	<3>	$t_{CYK}/2 - 10$		ns
上升时间	t_{KR}	<4>		10	ns
下降时间	t_{KF}	<5>		10	ns

时钟时序



32.8.2 总线时序

(1) 在复用总线模式

<R> 注意事项 当 V850ES/SG2-H 运行于 $f_{xx} > 20\text{ MHz}$ 时，务必插入地址保持等待和地址建立等待。

(a) 读 / 写入周期 (CLKOUT 异步)

(TA = -40 到 +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
地址建立时间 (到 ASTB↓)	tSAST	<6>	(0.5 + tASW) T - 20		ns
地址保持时间 (从 ASTB↓)	tHSTA	<7>	(0.5 + tAHW) T - 15		ns
从 RD↓到地址浮动的延迟时间	tFRDA	<8> 注 1		16	ns
		注 2		10	ns
从地址数据输入建立时间	tSAID	<9>		(2 + n + tASW + tAHW) T - 35	ns
<R> 从 RD↓数据输入建立时间	tSRID	<10>		(1 + n) T - 25	ns
从 ASTB↓到 RD, WRm↓的延迟时间	tDSTRDWR	<11>	(0.5 + tAHW) T - 15		ns
数据输入保持时间 (从 RD↑)	tHRDID	<12>	0		ns
<R> 从 RD↑的地址输出时间	tDRDA	<13> 注 1	(1 + i) T - 15		ns
		注 2	(1 + i) T - 10		ns
<R> 从 RD, WRm↑到 ASTB↑的延迟时间	tDRDWRST	<14> 注 1	0.5T - 15		ns
		注 2	0.5T - 10		ns
<R> 从 RD↑到 ASTB↓的延迟时间	tDRDST	<15> 注 1	(1.5 + i + tASW) T - 15		ns
		注 2	(1.5 + i + tASW) T - 10		ns
<R> RD, WRm 低电平宽度	tWRDWRL	<16> 注 1	(1 + n) T - 15		ns
		注 2	(1 + n) T - 10		ns
<R> ASTB 高电平宽度	tWSTH	<17> 注 1	(1 + i + tASW) T - 15		ns
		注 2	(1 + i + tASW) T - 10		ns
数据输出时间从 WRm↓	tDWROD	<18>		15	ns
数据输出建立时间 (到 WRm↑)	tSODWR	<19>	(1 + n) T - 20		ns
数据输出保持时间 (从 WRm↑)	tHWROD	<20>	T - 15		ns
WAIT 建立时间 (到地址)	tSAWT1	<21> n ≥ 1		(1.5 + tASW + tAHW) T - 35	ns
	tSAWT2	<22>		(1.5 + n + tASW + tAHW) T - 35	ns
WAIT 保持时间 (从地址)	tHAWT1	<23> n ≥ 1	(0.5 + n + tASW + tAHW) T		ns
	tHAWT2	<24>	(1.5 + n + tASW + tAHW) T		ns
WAIT 建立时间 (到 ASTB↓)	tSSTWT1	<25> n ≥ 1		(1 + tAHW) T - 25	ns
	tSSTWT2	<26>		(1 + n + tAHW) T - 25	ns
WAIT 保持时间 (从 ASTB↓)	tHSTWT1	<27> n ≥ 1	(n + tAHW) T		ns
	tHSTWT2	<28>	(1 + n + tAHW) T		ns

注 1. V850ES/SG2

<R> 2. V850ES/SG2-H

备注 1. tASW: 地址建立等待时钟的数量

tAHW: 地址保持等待时钟的数量

2. T = 1/fCPU (fCPU: CPU 操作时钟频率)

3. n: 在总线周期插入的等待时钟数量

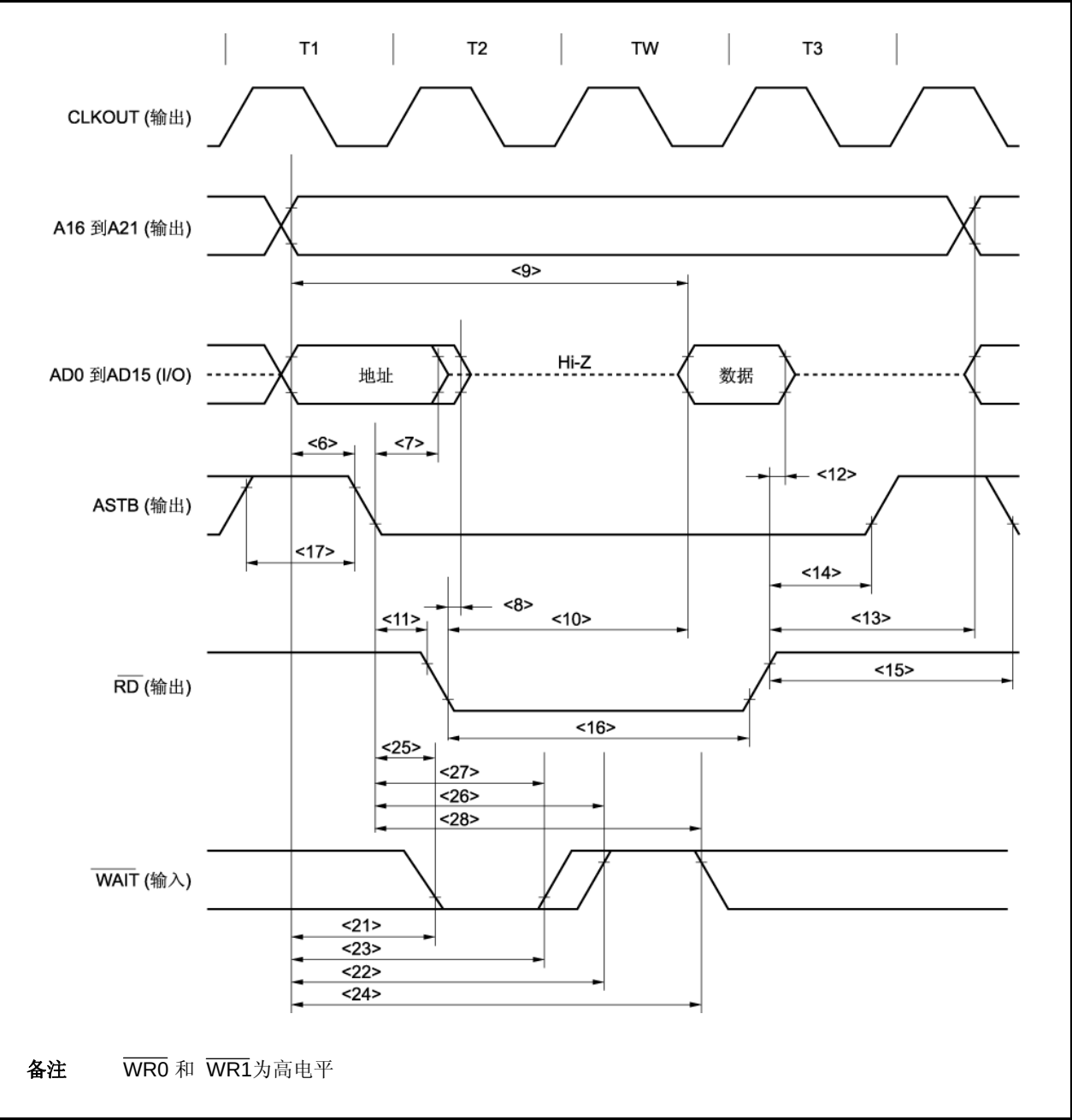
当插入可编程等待时, 采样时序改变。

4. m = 0, 1

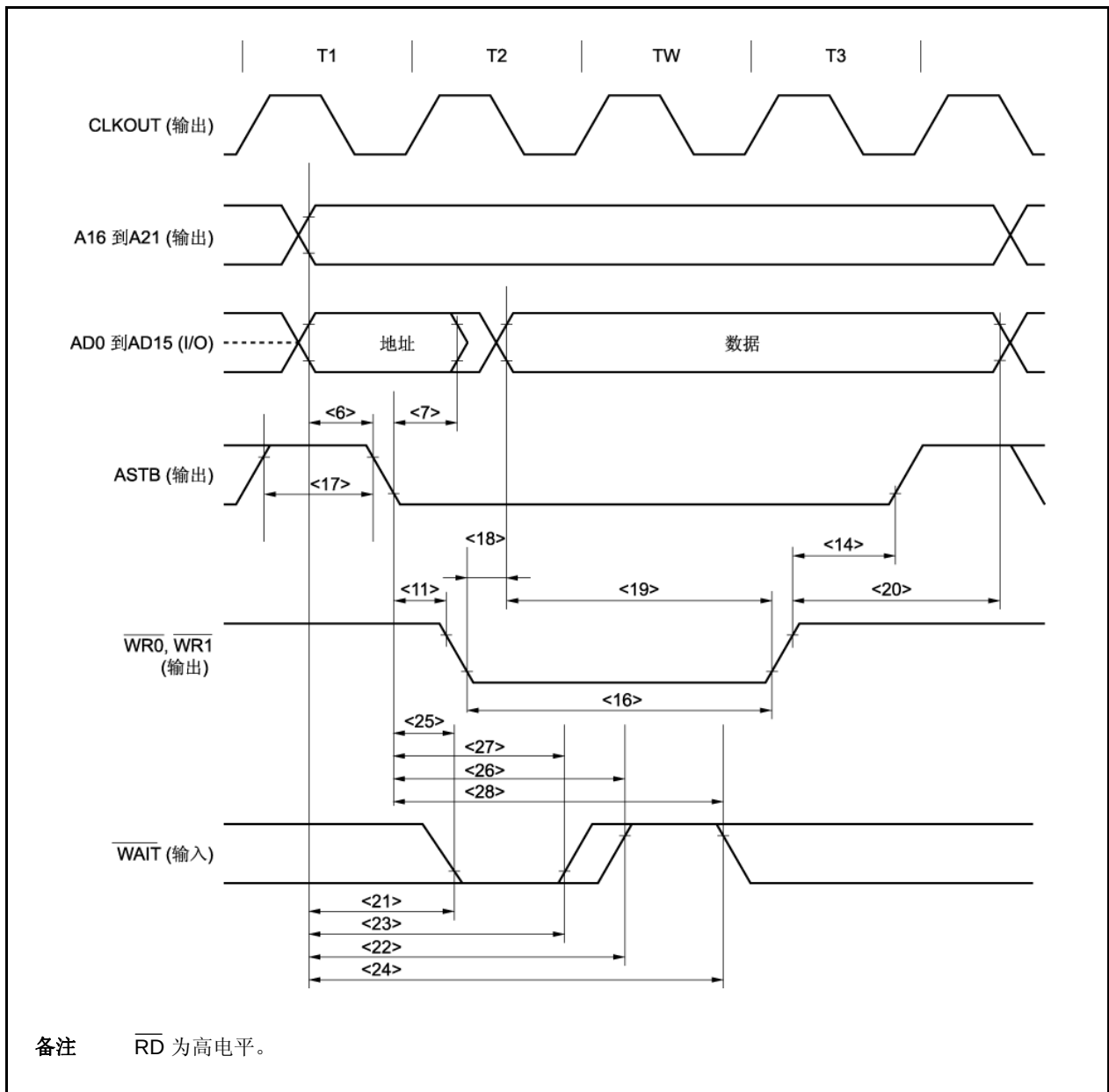
5. i: 在读取周期后插入的闲置状态的序号 (0 或 1)

6. 上面指定的值都是当从 X1 输入的时钟占空比为 1: 1 时的值。

读取周期（CLKOUT 异步）：在复用总线模式下



写入周期（CLKOUT 异步）：在复用总线模式下



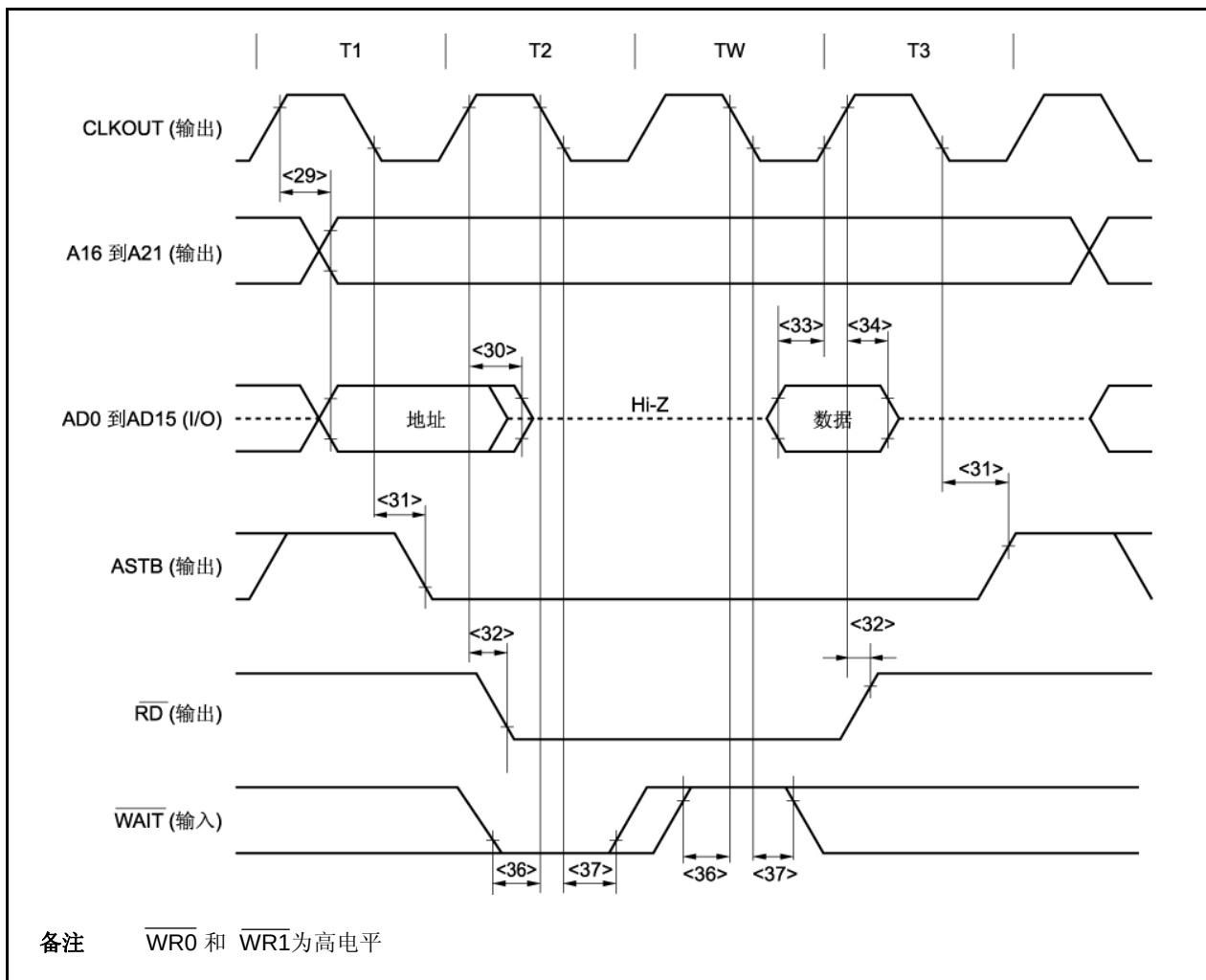
(b) 读 / 写入周期 (CLKOUT 同步)：在复用总线模式

(TA = -40 到 +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V, CL = 50 pF)

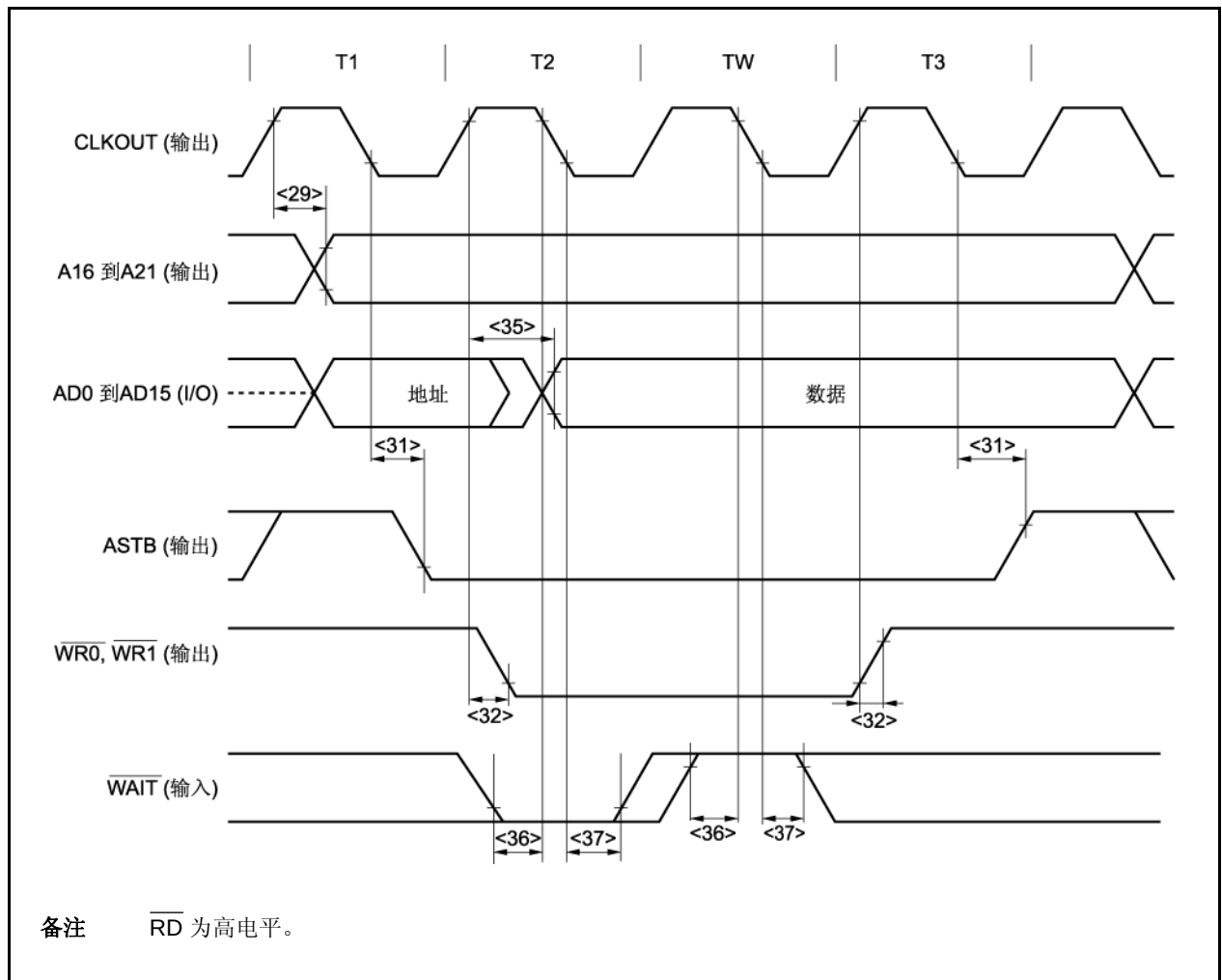
参数	符号	条件	最小值	最大值	单位
从 CLKOUT↑ 到地址的延迟时间	tDKA	<29>	0	25	ns
从 CLKOUT↑ 到地址浮动的延迟时间。	tFKA	<30>	0	19	ns
从 CLKOUT↓ 到 ASTB 的延迟时间	tDKST	<31>	-12	7	ns
从 CLKOUT↑ 到 RD, WRm 的延迟时间	tDKRDWR	<32>	-5	14	ns
数据输入建立时间 (到 CLKOUT↑)	tSIDK	<33>	15		ns
数据输入保持时间 (从 CLKOUT↑)	tHKID	<34>	5		ns
从 CLKOUT↑ 数据输出延迟时间	tDKOD	<35>		19	ns
WAIT 建立时间 (到 CLKOUT↓)	tSWTK	<36>	20		ns
WAIT 保持时间 (从 CLKOUT↓)	tHKWT	<37>	5		ns

- 备注
1. m = 0, 1
 2. 上面指定的值都是当从 X1 输入的时钟占空比为 1: 1 时的值。

读取周期 (CLKOUT 同步)：在复用总线模式下



写入周期（CLKOUT 同步）：在复用总线模式下



(2) 在分离总线模式下

- <R> 注意事项 1. 当 V850ES/SG2-H 运行于 $f_{xx} > 20 \text{ MHz}$ 时, 务必插入地址保持等待和地址建立等待。
- <R> 2. 当 V850ES/SG2-H 运行于 $f_{xx} > 20 \text{ MHz}$ 时, 务必插入至少一个数据等待。

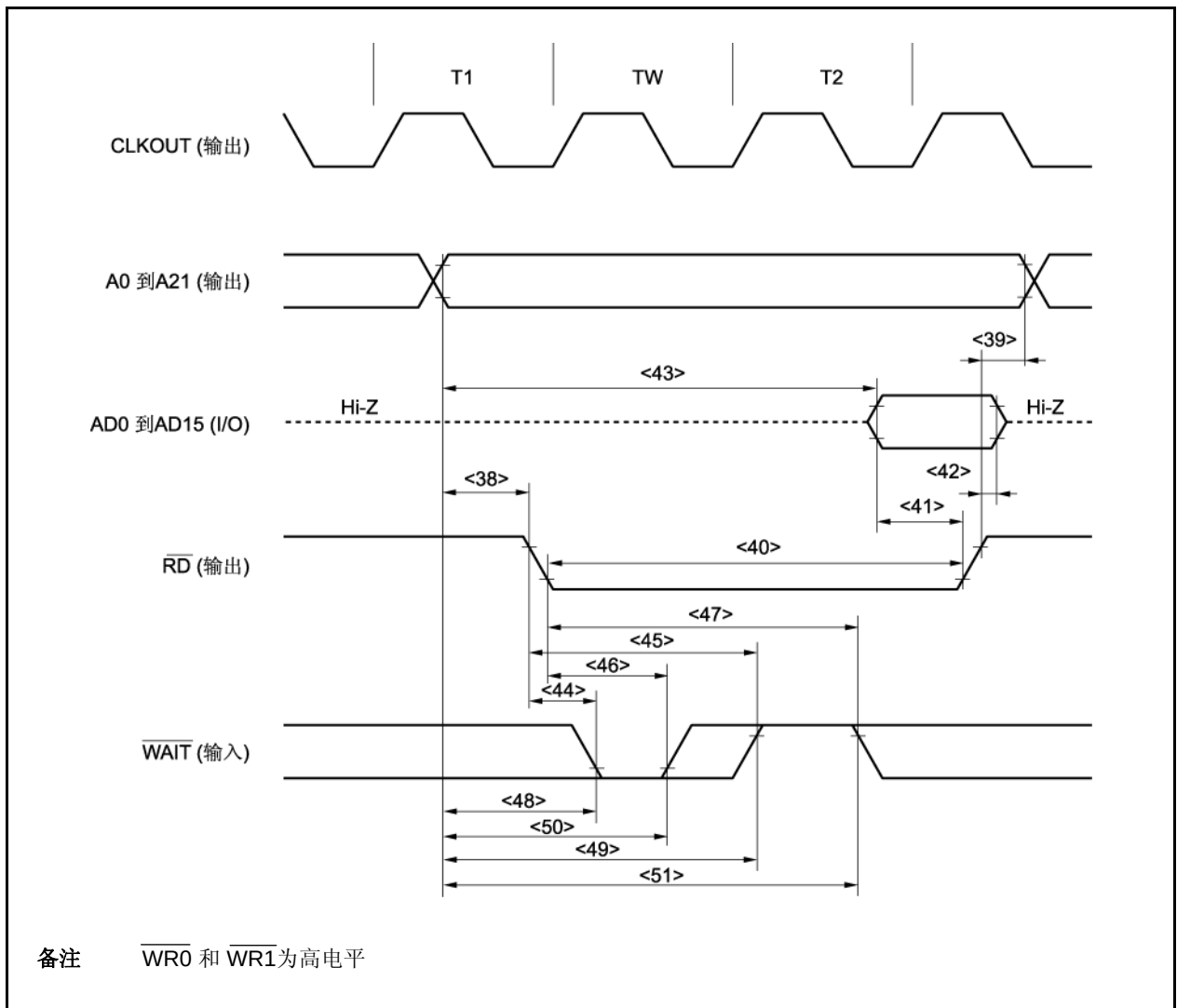
(a) 读取周期 (CLKOUT 异步): 在分离总线模式下

($T_A = -40$ 到 $+85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 \text{ V}$, $C_L = 50 \text{ pF}$)

参数	符号	条件	最小值	最大值	单位
<R> 地址建立时间 (从 $\overline{RD}\downarrow$)	t_{SARD}	<38> V850ES/SG2	$(0.5 + t_{ASW}) T - 23$		ns
		V850ES/SG2-H	$(0.5 + t_{ASW}) T - 25$		ns
地址保持时间 (从 $\overline{RD}\uparrow$)	t_{HARD}	<39>	$iT + 1$		ns
$\overline{RD}$ 低电平宽度	t_{WRDL}	<40>	$(1.5 + n + t_{AHW}) T - 10$		ns
数据建立时间 (到 $\overline{RD}\uparrow$)	t_{SISD}	<41>	23		ns
数据保持时间 (从 $\overline{RD}\uparrow$)	t_{HISD}	<42>	0		ns
数据建立时间 (到地址)	t_{SAID}	<43>		$(2 + n + t_{ASW} + t_{AHW}) T - 40$	ns
$\overline{WAIT}$ 建立时间 (到 $\overline{RD}\downarrow$)	t_{SRDWT1}	<44>		$(0.5 + t_{AHW}) T - 25$	ns
	t_{SRDWT2}	<45>		$(0.5 + n + t_{AHW}) T - 25$	ns
$\overline{WAIT}$ 保持时间 (从 $\overline{RD}\downarrow$)	t_{HRDWT1}	<46>	$(n - 0.5 + t_{AHW}) T$		ns
	t_{HRDWT2}	<47>	$(n + 0.5 + t_{AHW}) T$		ns
$\overline{WAIT}$ 建立时间 (到地址)	t_{SAWT1}	<48>		$(1 + t_{ASW} + t_{AHW}) T - 45$	ns
	t_{SAWT2}	<49>		$(1 + n + t_{ASW} + t_{AHW}) T - 45$	ns
<R> $\overline{WAIT}$ 保持时间 (从地址)	t_{HAWT1}	<50>	$(n + t_{ASW} + t_{AHW}) T$		ns
	t_{HAWT2}	<51>	$(1 + n + t_{ASW} + t_{AHW}) T$		ns

- 备注
1. t_{ASW} : 地址建立等待时钟的数量
 t_{AHW} : 地址保持等待时钟的数量
 2. $T = 1/f_{CPU}$ (f_{CPU} : CPU 作时钟频率)
 3. n : 在总线周期插入的等待时钟数量
当插入可编程等待时, 采样时序改变。
 4. i : 在读取周期后插入的闲置状态的序号 (0 或 1)
 5. 上面指定的值都是当从 X1 输入的时钟占空比为 1: 1 时的值。

读取周期（CLKOUT 异步）：在分离总线模式下



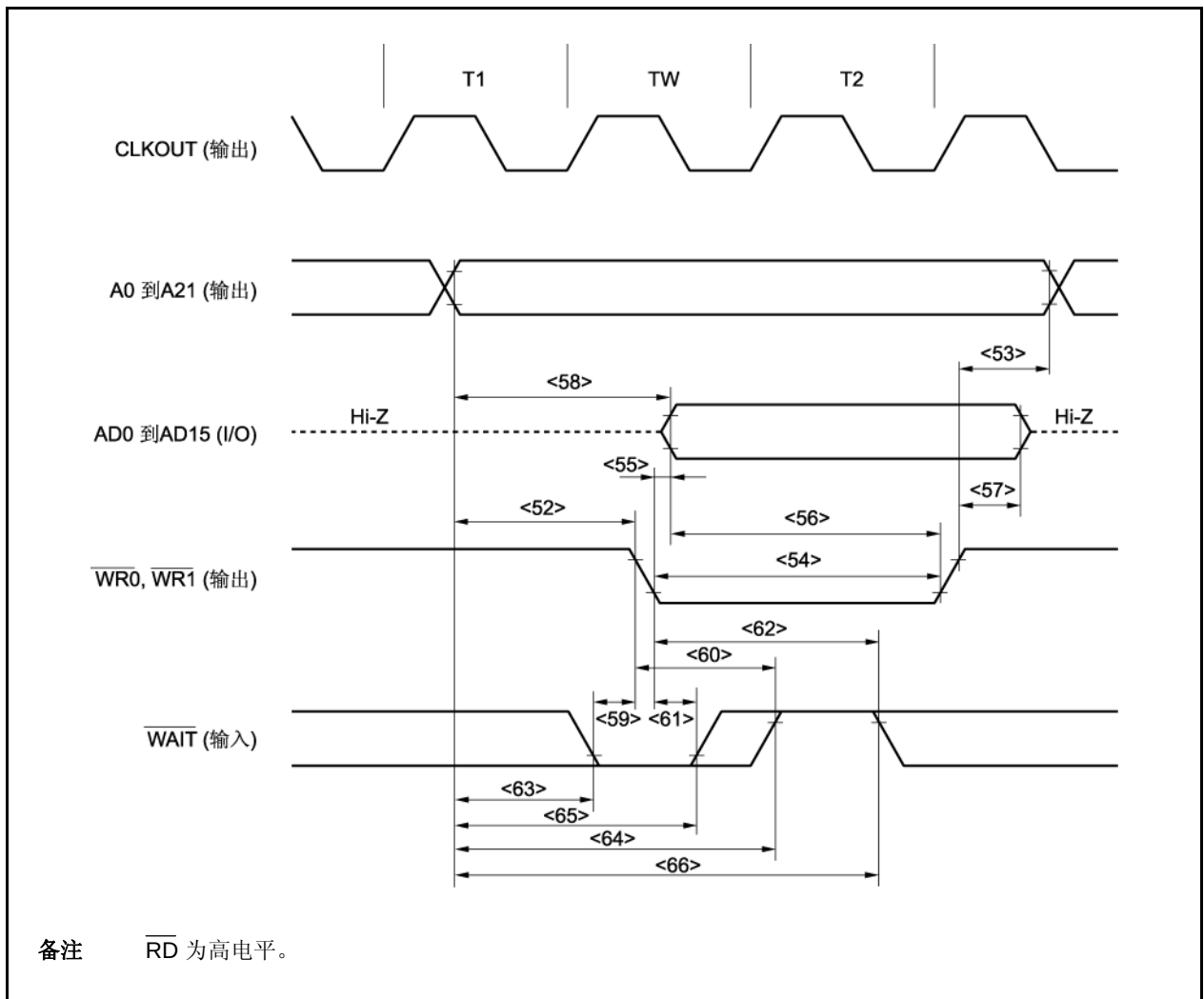
(b) 写入周期 (CLKOUT 异步)：在分离总线模式下

($T_A = -40$ 到 $+85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	最大值	单位
<R> 地址建立时间 (到 $\overline{WRm}\downarrow$)	t_{SAWR}	<52> V850ES/SG2	$(1 + t_{ASW} + t_{AHW}) T - 23$		ns
		V850ES/SG2-H	$(1 + t_{ASW} + t_{AHW}) T - 25$		ns
地址保持时间 (从 $\overline{WRm}\uparrow$)	t_{HAWR}	<53>	$0.5T - 10$		ns
$\overline{WRm}$ 低电平宽度	t_{WWRL}	<54>	$(0.5 + n) T - 10$		ns
数据输出时间从 $\overline{WRm}\downarrow$	t_{DOSDW}	<55>	-5		ns
数据建立时间 (到 $\overline{WRm}\uparrow$)	t_{SOSDW}	<56>	$(0.5 + n) T - 20$		ns
数据保持时间 (从 $\overline{WRm}\uparrow$)	t_{HOSDW}	<57>	$0.5T - 10$		ns
数据建立时间 (到地址)	t_{SAOD}	<58>	$(1 + t_{ASW} + t_{AHW}) T - 25$		ns
$\overline{WAIT}$ 建立时间 (到 $\overline{WRm}\downarrow$)	t_{SWRWT1}	<59>	22		ns
	t_{SWRWT2}	<60>		$nT - 22$	ns
$\overline{WAIT}$ 保持时间 (从 $\overline{WRm}\downarrow$)	t_{HWRWT1}	<61>	0		ns
	t_{HWRWT2}	<62>	nT		ns
$\overline{WAIT}$ 建立时间 (到地址)	t_{SAWT1}	<63>		$(1 + t_{ASW} + t_{AHW}) T - 45$	ns
	t_{SAWT2}	<64>		$(1 + n + t_{ASW} + t_{AHW}) T - 45$	ns
$\overline{WAIT}$ 保持时间 (从地址)	t_{HAWT1}	<65>	$(n + t_{ASW} + t_{AHW}) T$		ns
	t_{HAWT2}	<66>	$(1 + n + t_{ASW} + t_{AHW}) T$		ns

- 备注
1. $m = 0, 1$
 2. t_{ASW} : 地址建立等待时钟的数量
 t_{AHW} : 地址保持等待时钟的数量
 3. $T = 1/f_{CPU}$ (f_{CPU} : CPU 作时钟频率)
 4. n : 在总线周期插入的等待时钟数量
当插入可编程等待时, 采样时序改变.
 5. 上面指定的值都是当从 X1 输入的时钟占空比为 1: 1 时的值。

读取周期（CLKOUT 异步）：在分离总线模式下



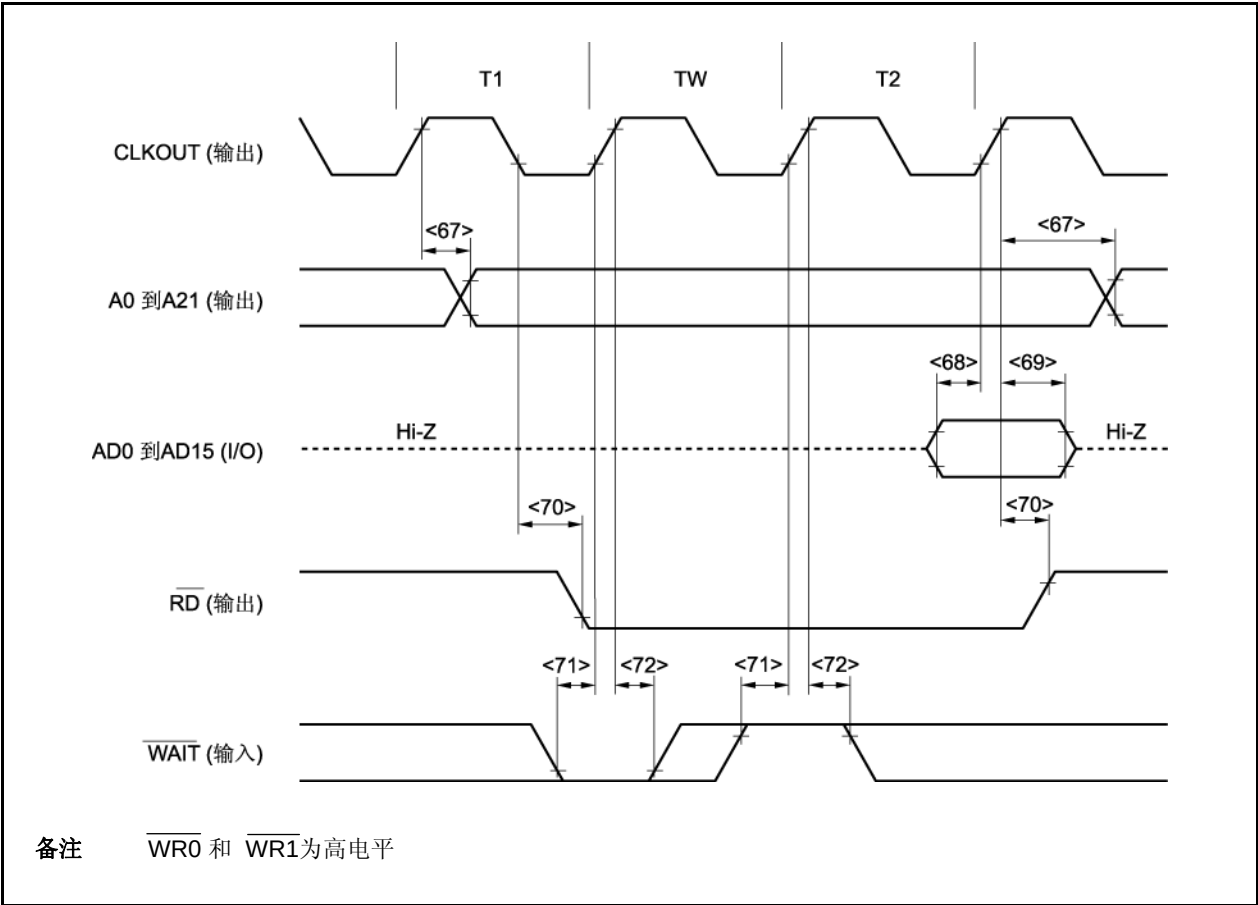
(c) 读取周期（CLKOUT 同步）：在分离总线模式下

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	最大值	单位
<R> 从 CLKOUT↑ 到地址的延迟时间	t_{DKSA}	<67> V850ES/SG2	2	25	ns
		V850ES/SG2-H	2	27	ns
数据输入建立时间（到 CLKOUT↑）	t_{SISDK}	<68>	20		ns
数据输入保持时间（从 CLKOUT↑）	t_{HKISD}	<69>	0		ns
<R> 从 CLKOUT↓↑ 到 $\overline{RD}$ 的延长时间	t_{DKSR}	V850ES/SG2	-2	12	ns
		V850ES/SG2-H	-2	14	ns
$\overline{WAIT}$ 建立时间（到 CLKOUT↑）	t_{SWTK}	<71>	20		ns
$\overline{WAIT}$ 保持时间（从 CLKOUT↑）	t_{HKWT}	<72>	0		ns

备注 上面指定的值都是当从 X1 输入的时钟占空比为 1: 1 时的值。

读取周期（CLKOUT 同步，1 个等待周期）：分离总线模式



(d) 写入周期（CLKOUT 同步）：在分离总线模式下

(TA = -40 到 +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V, CL = 50 pF)

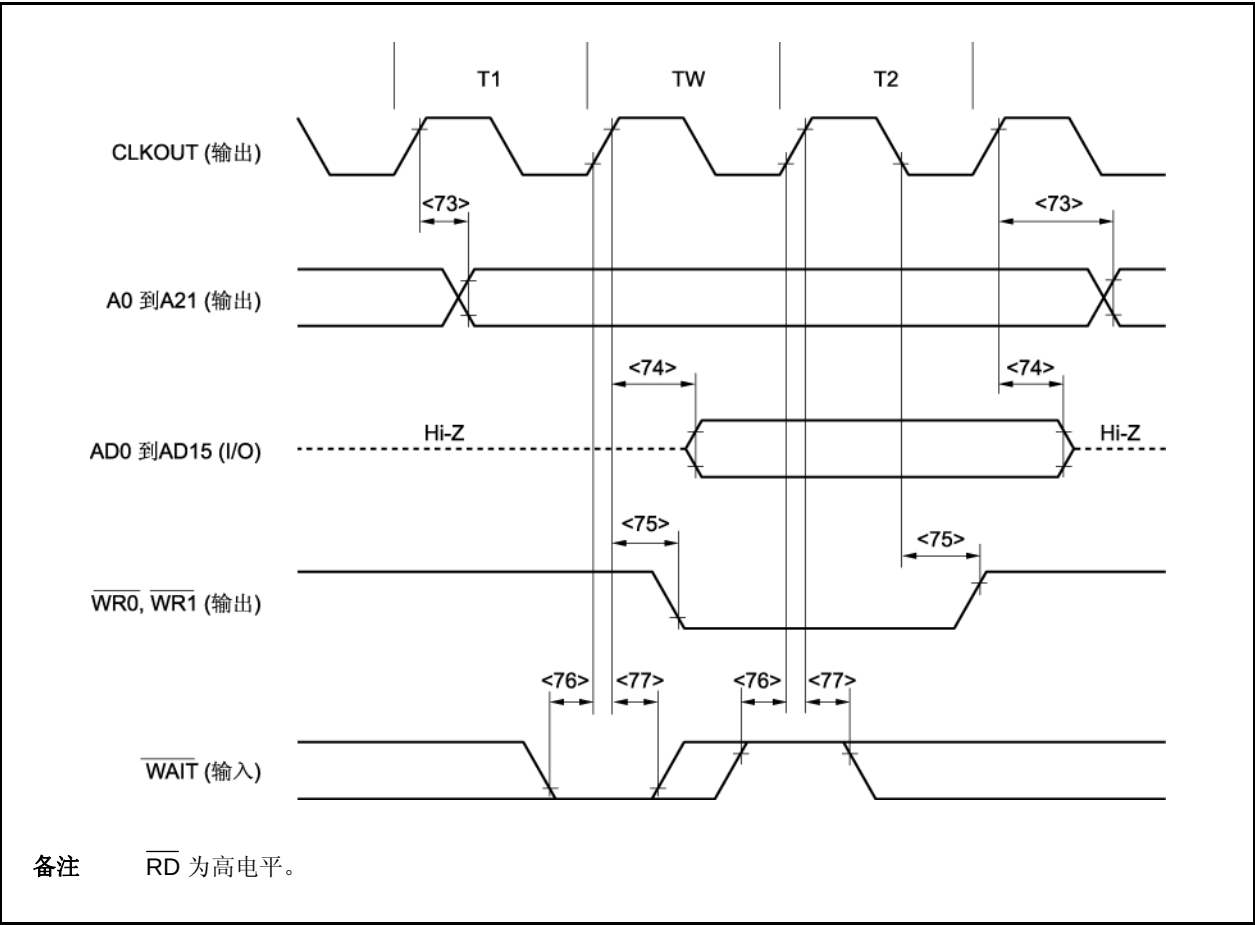
<R>

<R>

参数	符号	条件	最小值	最大值	单位
从 CLKOUT↑ 到地址的延迟时间	tDKSA	<73> V850ES/SG2	2	25	ns
		V850ES/SG2-H	2	27	ns
从 CLKOUT↑ 到数据输出的延迟时间	tDKSD	<74>	2	15	ns
从 CLKOUT↑↓ 到 WRm 的延迟时间	tDKSW	<75> V850ES/SG2	-2	12	ns
		V850ES/SG2-H	-2	14	ns
WAIT 建立时间（到 CLKOUT↑）	tSWTK	<76>	20		ns
WAIT 保持时间（从 CLKOUT↑）	tHKWT	<77>	0		ns

- 备注
- 1. m = 0, 1
 - 2. 上面指定的值都是当从 X1 输入的时钟占空比为 1: 1 时的值。

写入周期（CLKOUT 同步）：在分离总线模式下



(3) 总线保持

(a) CLKOUT 异步

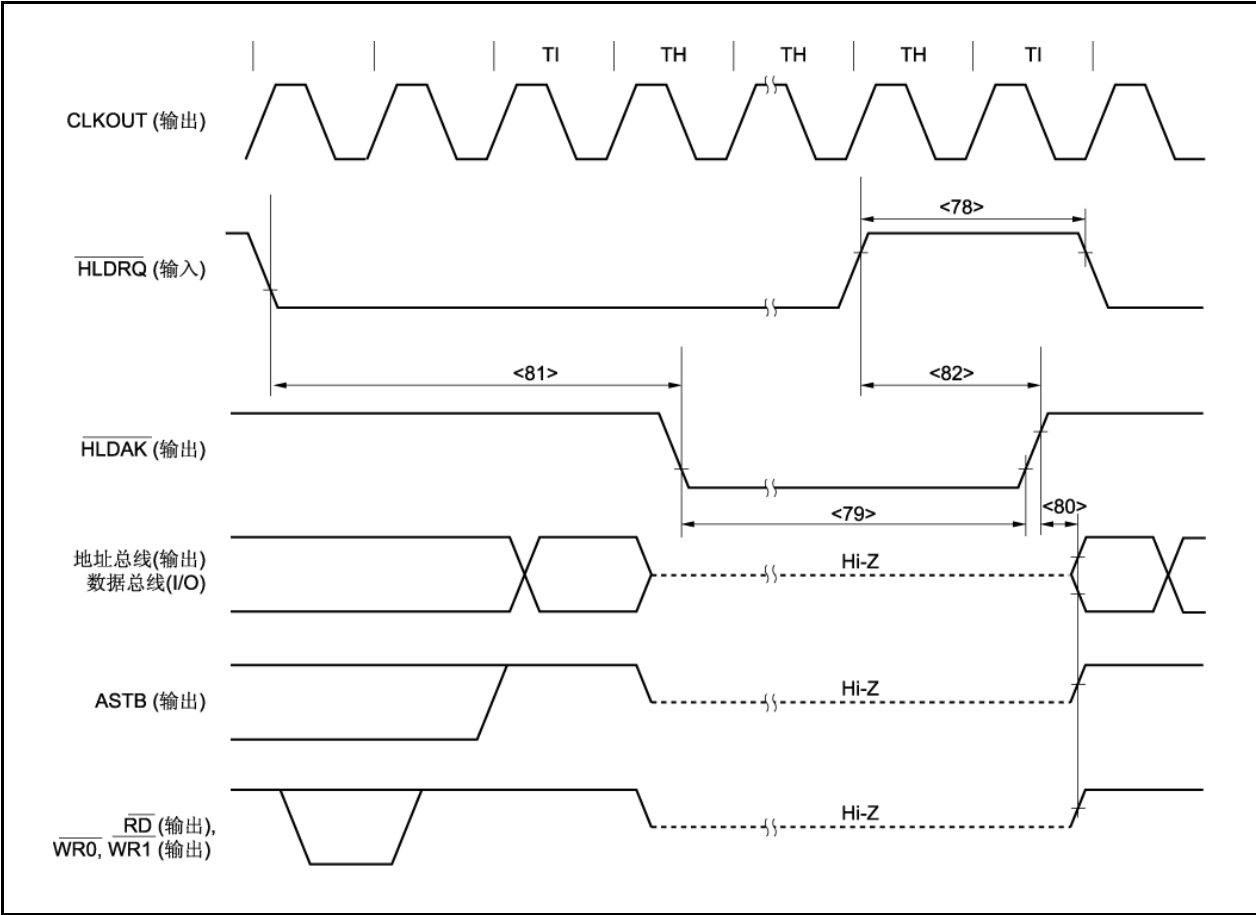
($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

<R>

参数	符号	条件	最小值	最大值	单位
H $\overline{\text{LDRQ}}$ 高电平宽度	t_{WHQH}	<78>	$T + 10$		ns
H $\overline{\text{LDAK}}$ 低电平宽度	t_{WHAL}	<79>	$T - 15$		ns
从 H $\overline{\text{LDAK}}\uparrow$ 到总线输出的延迟时间	t_{DHAC}	<80>	V850ES/SG2	-3	ns
			V850ES/SG2-H	-5	ns
从 H $\overline{\text{LDRQ}}\downarrow$ 到 H $\overline{\text{LDAK}}\downarrow$ 的延迟时间	t_{DHQA1}	<81>		$(2n + 7.5) T + 25$	ns
从 H $\overline{\text{LDRQ}}\uparrow$ 到 H $\overline{\text{LDAK}}\uparrow$ 的延迟时间	t_{DHQA2}	<82>	$0.5T$	$1.5T + 25$	ns

- 备注
- 1. $T = 1/f_{\text{CPU}}$ (f_{CPU} : CPU 作时钟频率)
 - 2. n: 在总线周期插入的等待时钟数量
当插入可编程等待时, 采样时序改变.
 - 3. 上面指定的值都是当从 X1 输入的时钟占空比为 1: 1 时的值。

总线保持 (CLKOUT 异步)



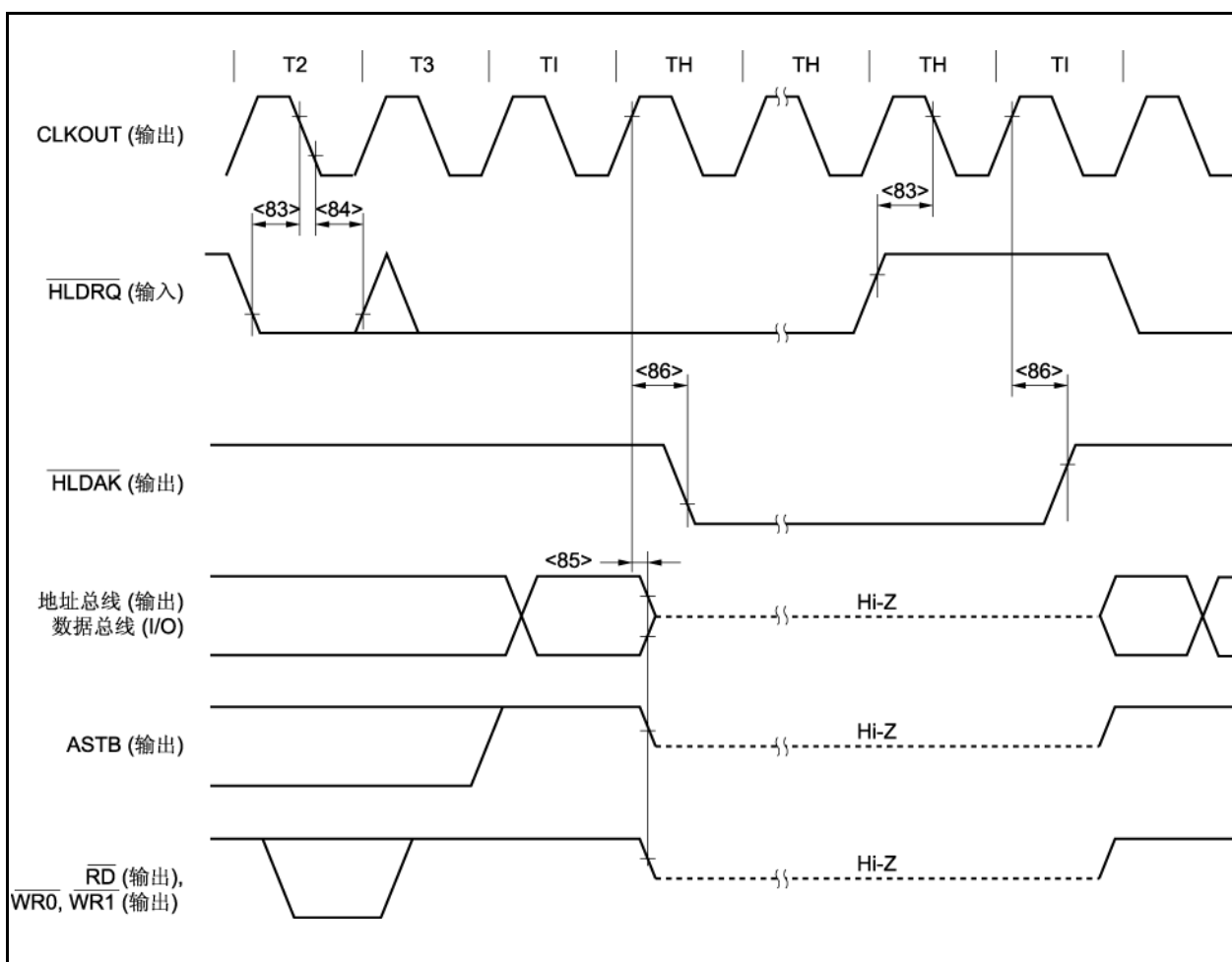
(b) CLKOUT 同步

($T_A = -40$ 到 $+85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	最大值	单位
HLD $\overline{RQ}$ 建立时间 (到 CLKOUT $\downarrow$)	t_{SHQK}	<83>	20		ns
HLD $\overline{RQ}$ 保持时间 (from CLKOUT $\downarrow$)	t_{HKHQ}	<84>	5		ns
从 CLKOUT $\uparrow$ 到总线浮动的延迟时间	t_{DKF}	<85>		19	ns
从 CLKOUT $\uparrow$ 到 HLD $\overline{AK}$ 的延迟时间	t_{DKHA}	<86>		19	ns

备注 上面指定的值都是当从 X1 输入的时钟占空比为 1: 1 时的值。

总线保持 (CLKOUT 同步)



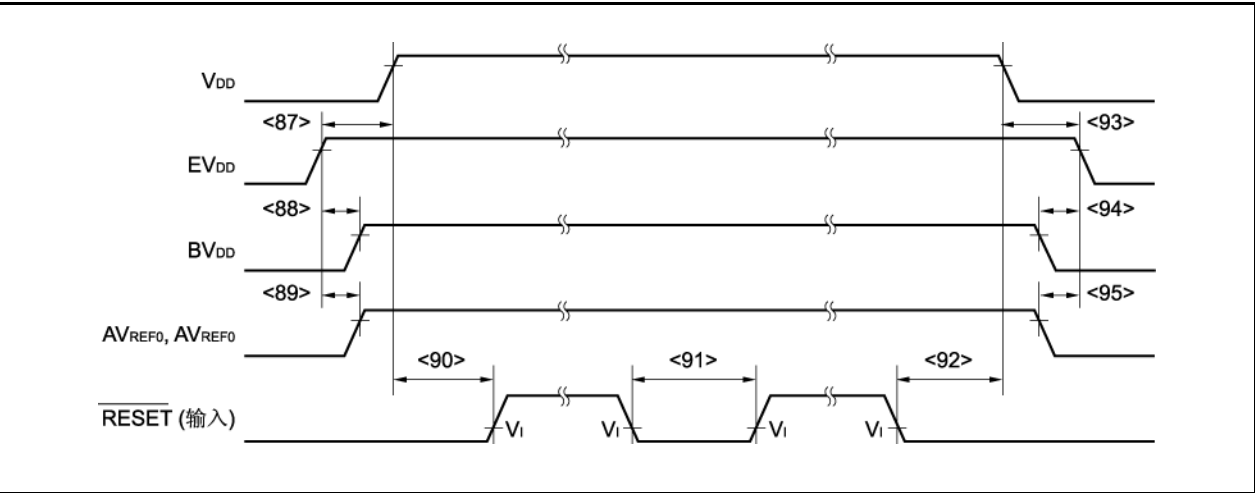
32.9 基本操作

(1) 上电 / 断电 / 复位时序

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号		条件	最小值	最大值	单位
$EV_{DD}\uparrow \rightarrow V_{DD}\uparrow$	t_{REL}	<87>		0		ns
$EV_{DD}\uparrow \rightarrow BV_{DD}\uparrow$	t_{REB}	<88>		0	t_{REL}	ns
$EV_{DD}\uparrow \rightarrow AV_{REF0}, AV_{REF1}\uparrow$	t_{REA}	<89>		0	t_{REL}	ns
$EV_{DD}\uparrow \rightarrow \overline{RESET}\uparrow$	t_{RER}	<90>		$500 + t_{REG}^{*}$		ns
$\overline{RESET}$ 低电平宽度	t_{WRSL}	<91>	模拟噪声消除 (擦除 / 写入期间)	500		ns
			模拟噪声消除	500		ns
$\overline{RESET}\downarrow \rightarrow V_{DD}\downarrow$	t_{FRE}	<92>		500		ns
$V_{DD}\downarrow \rightarrow EV_{DD}\downarrow$	t_{FEL}	<93>		0		ns
$BV_{DD}\downarrow \rightarrow EV_{DD}\downarrow$	t_{FEB}	<94>		0	t_{FEL}	ns
$AV_{REF0}, AV_{REF1}\downarrow \rightarrow EV_{DD}\downarrow$	t_{FEA}	<95>		0	t_{FEL}	ns

注 由片上稳压器特性决定。



(2) 中断, FLMD0 引脚时序

(TA = -40 到 +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
NMI 高电平宽度	tWNIH	模拟噪声消除	500		ns
NMI 低电平宽度	tWNIL	模拟噪声消除	500		ns
INTPn [#] 高电平宽度	tWITH	n = 0 to 7 (模拟噪声消除)	500		ns
		n = 3 (数字噪声消除)	3T _{SMP} + 20		ns
INTPn [#] 低电平宽度	tWITL	n = 0 to 7 (模拟噪声消除)	500		ns
		n = 3 (数字噪声消除)	3T _{SMP} + 20		ns
FLMD0 高电平宽度	tWMDH		500		ns
FLMD0 低电平宽度	tWMDL		500		ns

<R>

<R>

注 $\overline{\text{DRST}}$ 引脚与 INTP2 引脚特性相同。备注 T_{SMP}: 噪声消除采样时钟周期

(3) 按键返回时序

(TA = -40 到 +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
KRn 高电平宽度	tWKRH	模拟噪声消除	500		ns
KRn 低电平宽度	tWKRL	模拟噪声消除	500		ns

备注 n = 0 到 7

(4) 定时器时序

(TA = -40 到 +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
TI 高电平宽度	tTIH	TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, TIP31, TIP40, TIP41, TIP50, TIP51, TIQ00 到 TIQ03	2T + 20		ns
TI 低电平宽度	tTIL		2T + 20		ns

备注 T = 1/f_{xx}

(5) UART 时序

(TA = -40 到 +85°C, BVDD ≤ VDD = EVDD = AVREF0 = AVREF1, VSS = EVSS = BVSS = AVSS = 0 V, CL = 50 pF)

参数	符号	条件	最小值	最大值	单位
发送速率				312.5	kbps
ASCK0 周期时间				10	MHz

(6) CSIB 时序

(a) 主设备模式

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号		条件	最小值	最大值	单位
$\overline{\text{SCKBn}}$ 周期时间	t_{KCY1}	<96>		125		ns
$\overline{\text{SCKBn}}$ 高 / 低电平宽度	t_{KH1} , t_{KL1}	<97>		$t_{\text{KCY1}}/2 - 5$		ns
SIBn 建立时间 (到 $\overline{\text{SCKBn}}\uparrow$)	t_{SIK1}	<98>		30		ns
SIBn 保持时间 (从 $\overline{\text{SCKBn}}\uparrow$)	t_{SSI1}	<99>		30		ns
从 $\overline{\text{SCKBn}}\downarrow$ 到 SOBn 输出的延迟时间	t_{KS01}	<100>			30	ns

备注 $n = 0$ 到 4

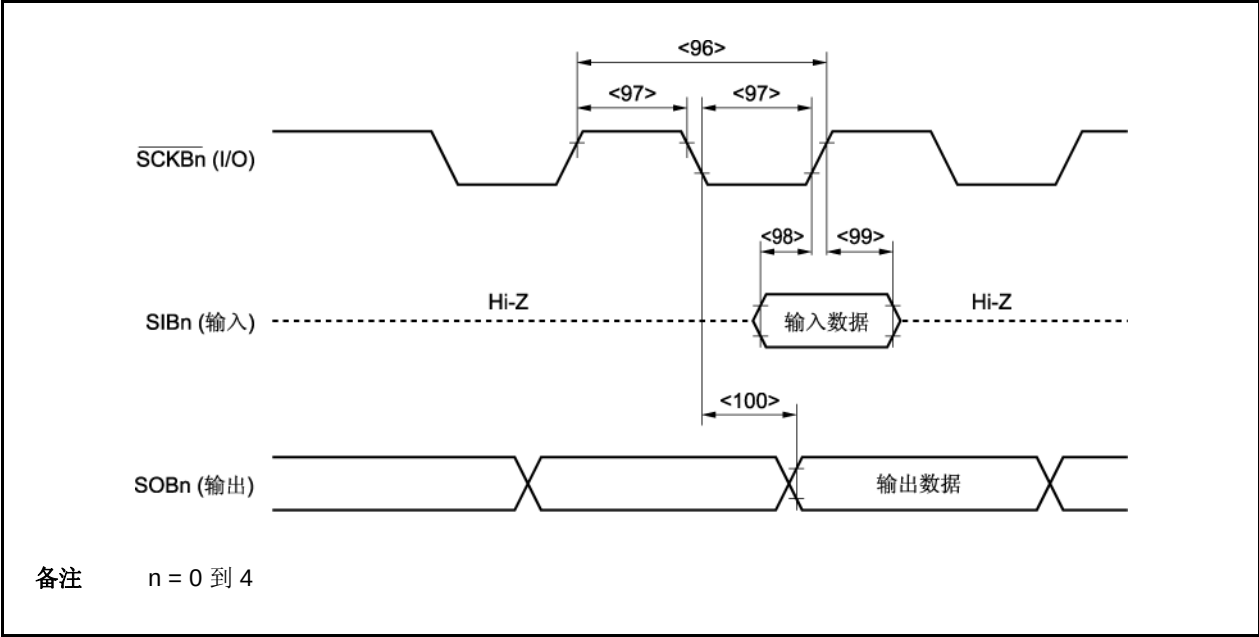
(b) 从设备模式

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号		条件	最小值	最大值	单位
$\overline{\text{SCKBn}}$ 周期时间	t_{KCY2}	<96>		125		ns
$\overline{\text{SCKBn}}$ 高 / 低电平宽度	t_{KH2} , t_{KL2}	<97>		57.5		ns
SIBn 建立时间 (到 $\overline{\text{SCKBn}}\uparrow$)	t_{SIK2}	<98>		30		ns
SIBn 保持时间 (从 $\overline{\text{SCKBn}}\uparrow$)	t_{SSI2}	<99>		30		ns
从 $\overline{\text{SCKBn}}\downarrow$ 到 SOBn 输出的延迟时间	t_{KS02}	<100>	V850ES/SG2		30	ns
			V850ES/SG2-H		35	ns

<R>

备注 $n = 0$ 到 4

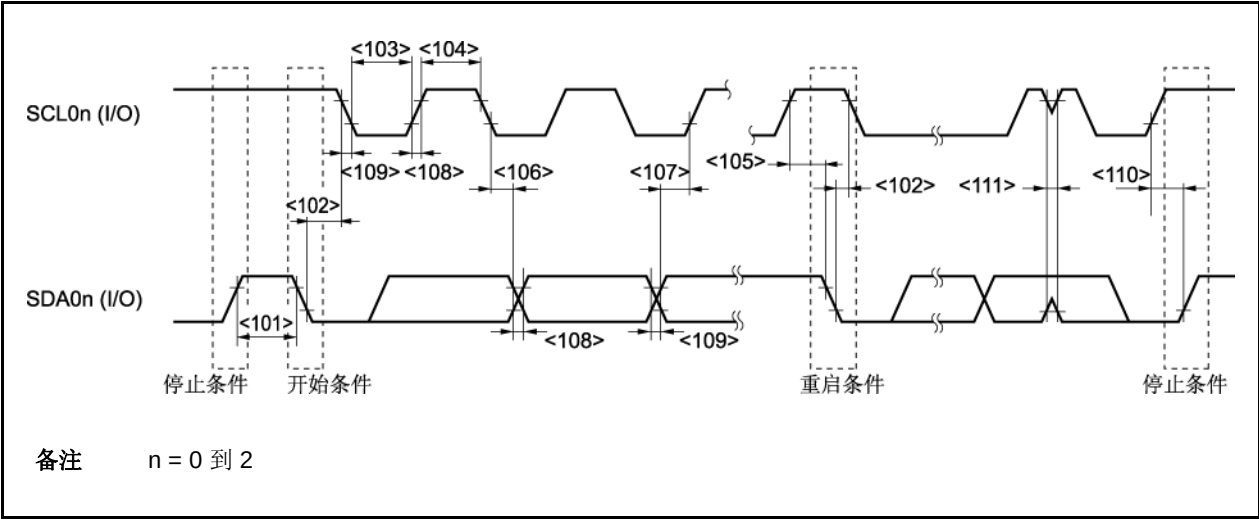


(7) I²C 总线模块（仅限内置 I²C 总线版本（Y 产品））(T_A = -40 到 +85°C, BV_{DD} ≤ V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号		正常模式		高速模式		单位
			最小值	最大值	最小值	最大值	
SCL0n 时钟频率	f _{CLK}		0	100	0	400	kHz
总线自由时间 (从启动到停止的时间)	t _{BUF}	<101>	4.7	—	1.3	—	μs
保持时间 ^{#1}	t _{HD: STA}	<102>	4.0	—	0.6	—	μs
SCL0n 时钟低电平宽度	t _{LOW}	<103>	4.7	—	1.3	—	μs
SCL0n 时钟高电平宽度	t _{HIGH}	<104>	4.0	—	0.6	—	μs
启动 / 重启条件的建立时间	t _{SU: STA}	<105>	4.7	—	0.6	—	μs
数据保持时间	CBUS 兼容主设备	t _{HD: DAT}	<106>	5.0	—	—	μs
	I ² C 模式						
			0 ^{#2}	—	0 ^{#2}	0.9 ^{#3}	μs
数据建立时间	t _{SU: DAT}	<107>	250	—	100 ^{#4}	—	ns
SDA0n 和 SCL0n 信号上升时间	t _R	<108>	—	1000	20 + 0.1Cb ^{#5}	300	ns
SDA0n 和 SCL0n 信号下降时间	t _F	<109>	—	300	20 + 0.1Cb ^{#5}	300	ns
停止条件建立时间	t _{SU: STO}	<110>	4.0	—	0.6	—	μs
输入滤波器尖峰抑制的脉冲宽度	t _{SP}	<111>	—	—	0	50	ns
每根总线的负载电容	C _b		—	400	—	400	pF

- 注
1. 在起始条件中，在保持时间之后产生第一个时钟脉冲。
 2. 系统内 SDA0n（在 SCL0n 信号的 V_{IHmin}）信号需要一个最少 300ns 的保持时间，以使在 SCL0n 的下降沿处在未定义区域。
 3. 如果系统不延长 SCL0n 低电平的信号保持时间（t_{LOW}），仅需要满足最大数据保持时间（t_{HD: DAT}）。
 4. 高速模式 I²C 总线可以用于正常模式 I²C 总线系统。在这种情况下，设定高速模式 I²C 总线会遇到如下条件。
 - 如果系统不延长 SCL0n 低电平的信号保持时间：
t_{SU: DAT} ≥ 250 ns
 - 如果系统延长 SCL0n 低电平的信号保持时间：
在 SCL0n 线释放之前（t_{Rmax} + t_{SU: DAT} = 1,000 + 250 = 1,250 ns：正常模式 I²C 总线规范）发送如下数据位到 SDA0n。
 5. C_b：一根总线的电容总和（单位：pF）

备注 n = 0 到 2



(8) IEBus 控制器（仅限内置 IEBus 控制器的产品）

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

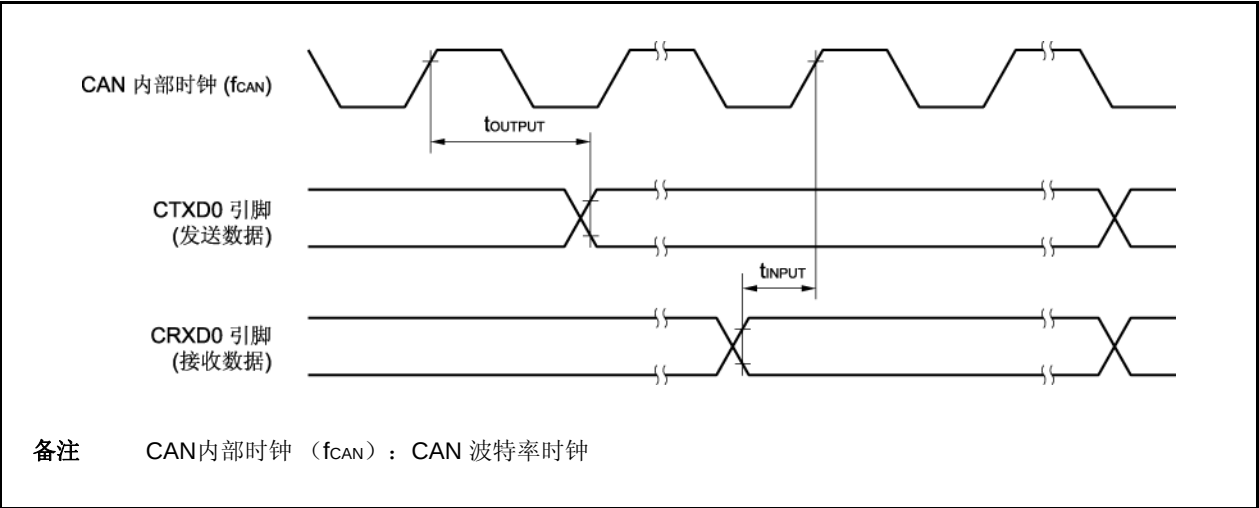
参数	符号	条件	最小值	典型值	最大值	单位
IEBus系统时钟频率	f _s	通信模式: 模式 1, 2	5.91	6.00 [‡]	6.09	MHz
			6.20	6.29 [‡]	6.38	MHz

注 IEBus 系统时钟频率 6.0 MHz 和 6.29 MHz 不能一起使用..

(9) CAN 时序（仅限内置 CAN 控制器的产品）

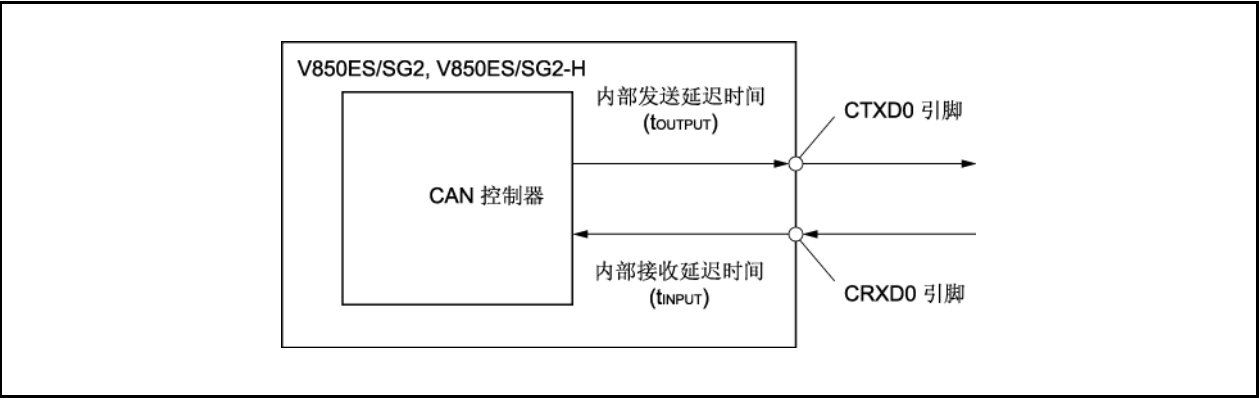
($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	最大值	单位
发送速率				1	Mbps
内部延迟时间	t_{NODE}			100	ns



内部延迟时间 (t_{NODE}) = 内部发送延迟时间 (t_{OUTPUT}) + 内部接受延迟时间 (t_{INPUT})

<R>



(10) A/D 转换器

($T_A = -40$ 到 $+85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $3.0\text{ V} \leq AV_{REF0} \leq 3.6\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	典型值	最大值	单位
分辨率					10	bit
总误差 ^注		$3.0 \leq AV_{REF0} \leq 3.6\text{ V}$			± 0.6	%FSR
转换时间	t_{CONV}		2.6		24	μs
零度误差					± 0.5	%FSR
满度误差					± 0.5	%FSR
非线性误差					± 4.0	LSB
微非线性误差					± 4.0	LSB
模拟输入电压	V_{IAN}		AV_{SS}		AV_{REF0}	V
参考电压	AV_{REF0}		3.0		3.6	V
AV _{REF0} 电流	AI _{REF0}	正常转换模式		3	6.5	mA
		高速转换模式		4	10	mA
		当 A/D 转换未使用时			5	μA

注 不包括量化误差 ($\pm 0.05\%$ FSR)。

注意事项 在 A/D 转换期间不要设置 (读 / 写) 复用端口; 否则转换分辨率可能降低。

备注 LSB: 最小有效位
FSR: 满度范围

(11) D/A 转换器

($T_A = -40$ 到 $+85^\circ\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $3.0\text{ V} \leq AV_{REF1} \leq 3.6\text{ V}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	典型值	最大值	单位
分辨率					8	位
总误差 ^{注1}		$R = 2\text{ M}\Omega$			± 1.2	%FSR
设置时间		$C = 20\text{ pF}$			3	μs
输出电阻	R_o	输出 data 55H		3.5		$\text{k}\Omega$
参考电压	AV_{REF1}		3.0		3.6	V
AV _{REF1} 电流 ^{注2}	AI _{REF1}	D/A 转换操作		1	2.5	mA
		D/A 转换停止			5	μA

注 1. 不包括量化误差 ($\pm 0.5\text{ LSB}$)。
2. D/A 转换器 1 通道的值

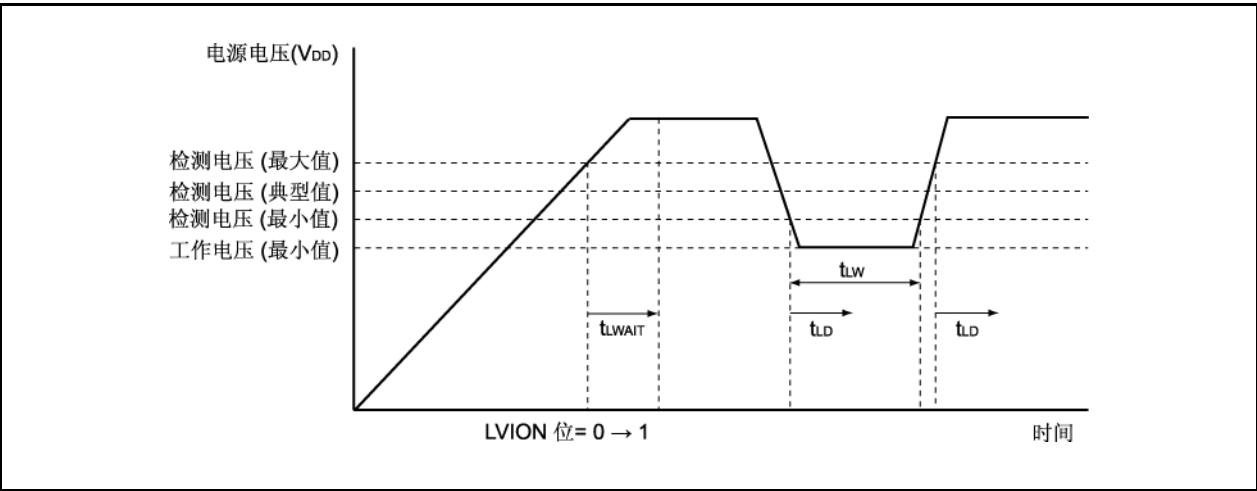
备注 R 为输出引脚负载阻抗, C 为输出引脚负载电容。

<R> (12) LVI 电路规范（只适用于 V850ES/SG2）

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V_{LV10}		2.85	3.0	3.15	V
响应时间 ^注	t_{LD}	在 V_{DD} 达到 V_{LV10}/V_{LV11} （最大值）后，或在 V_{DD} 下降到 V_{LV10}/V_{LV11} （最小值）后		0.2	2.0	ms
最小脉冲宽度	t_{LW}		0.2			ms
参考电压稳定等待时间	t_{LWAIT}	在 V_{DD} 达到 2.85 V（最小值）后		0.1	0.2	ms

注 检测电压和输出一个中断或复位信号所需要的时间。

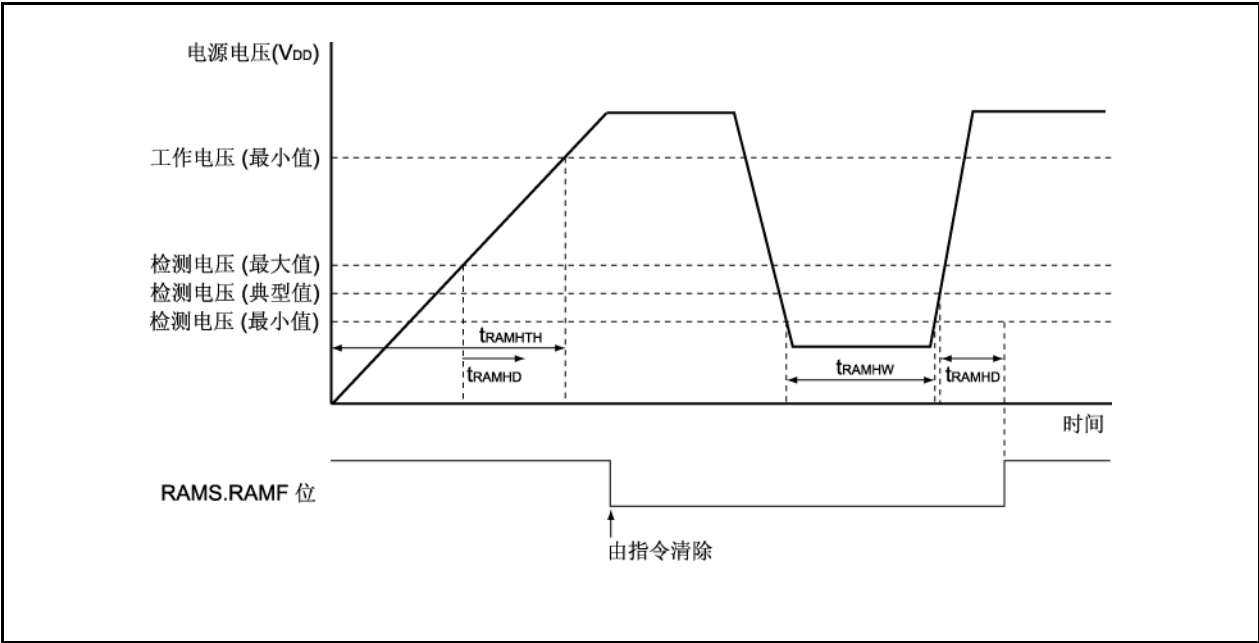


(13) RAM 保存检测

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

参数	符号	条件	最小值	典型值	最大值	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}	$V_{\text{DD}} = 0$ 到 2.85 V (V850ES/SG2) $V_{\text{DD}} = 0$ 到 3.0 V (V850ES/SG2-H)	0.002			ms
响应时间 ^注	t_{RAMHD}	在 V_{DD} 到达 2.1 V		0.2	2.0	ms
最小脉冲宽度	t_{RAMHW}		0.2			ms

注 检测电压和设置 RAMS.RAMF 位所需要的时间。



32.10 Flash存储器编程特性

($T_A = -40$ 到 $+85^{\circ}\text{C}$, $BV_{DD} \leq V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

(1) 基本特性

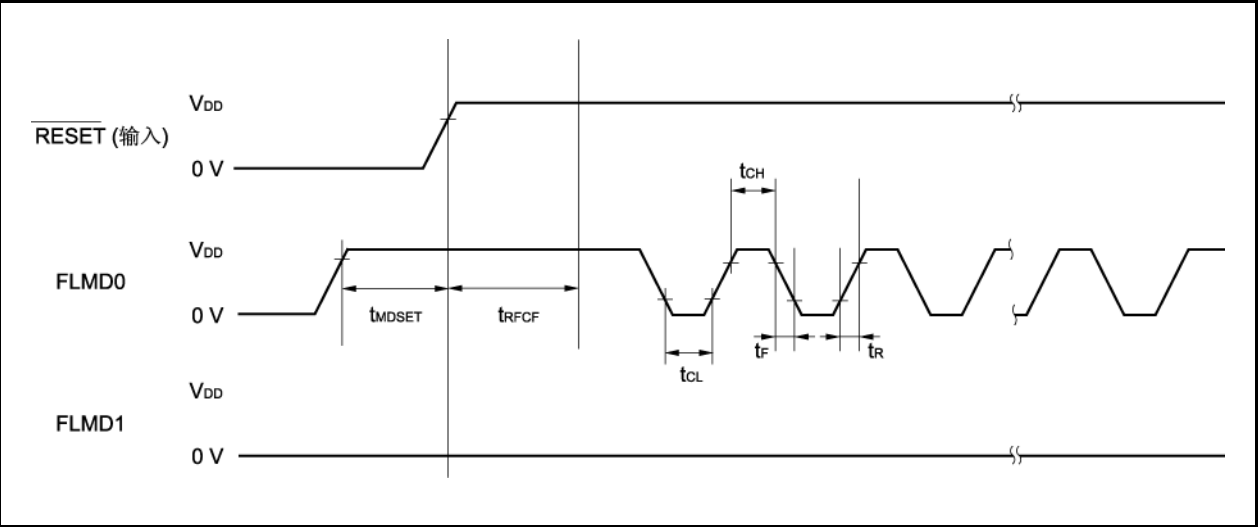
参数	符号	条件	最小值	典型值	最大值	单位
工作频率	f_{CPU}	V850ES/SG2	2.5		20	MHz
		V850ES/SG2-H	2.5		32	MHz
电源电压	V_{DD}	V850ES/SG2	2.85		3.6	V
		V850ES/SG2-H	3.0		3.6	V
重写次数	C_{WRT}				100	times
编程温度	t_{PRG}		-40		+85	$^{\circ}\text{C}$

(2) 串行写入操作特性

参数	符号	条件	最小值	典型值	最大值	单位
FLMD0, FLMD1建立时间	t_{MDSET}		2		3,000	ms
从 $\overline{\text{RESET}}\uparrow$ FLMD 计数开始时间	t_{RFCF}	$f_x = 2.5$ 到 10 MHz (V850ES/SG2) $f_x = 2.5$ 到 8 MHz (V850ES/SG2-H)	$17,855/f_x + \alpha$			s
FLMD0计数器高电平宽度 / 低电平宽度	$t_{\text{CH}}/t_{\text{CL}}$		10	100		μs
FLMD0 计数器上升时间 / 下降时间	$t_{\text{r}}/t_{\text{f}}$				50	ns

备注 α = 振荡稳定时间

Flash 写入模式建立时序



(3) 编程特性

(a) V850ES/SG2

参数	符号	条件		最小值	典型值	最大值	单位
块擦除时间		$f_{xx} = 20 \text{ MHz}$	注 1		304		ms
			注 2		1,405		ms
			注 3		3,057		ms
每 256 字节写入时间		$f_{xx} = 20 \text{ MHz}$			8.1		ms
块内部校验时间		$f_{xx} = 20 \text{ MHz}$	注 1		20		ms
			注 2		141		ms
			注 3		322		ms
块空白检查时间		$f_{xx} = 20 \text{ MHz}$	注 1		9.2		ms
			注 2		64		ms
			注 3		147		ms
Flash 存储器信息设置时间		$f_{xx} = 20 \text{ MHz}$			1.0		ms

- 注
1. 块大小 = 4 KB
 2. 块大小 = 28 KB
 3. 块大小 = 64 KB

注意事项 当对出厂产品进行初次写入时，将“擦除写入”和“只写入”两个过程都按一次重写来计数。

示例 (P: 写入, E: 擦除)

出库产品 $\longrightarrow P \rightarrow E \rightarrow P \rightarrow E \rightarrow P$: 3 次重写

出库产品 $\rightarrow E \rightarrow P \rightarrow E \rightarrow P \rightarrow E \rightarrow P$: 3 次重写

<R>

(b) V850ES/SG2-H

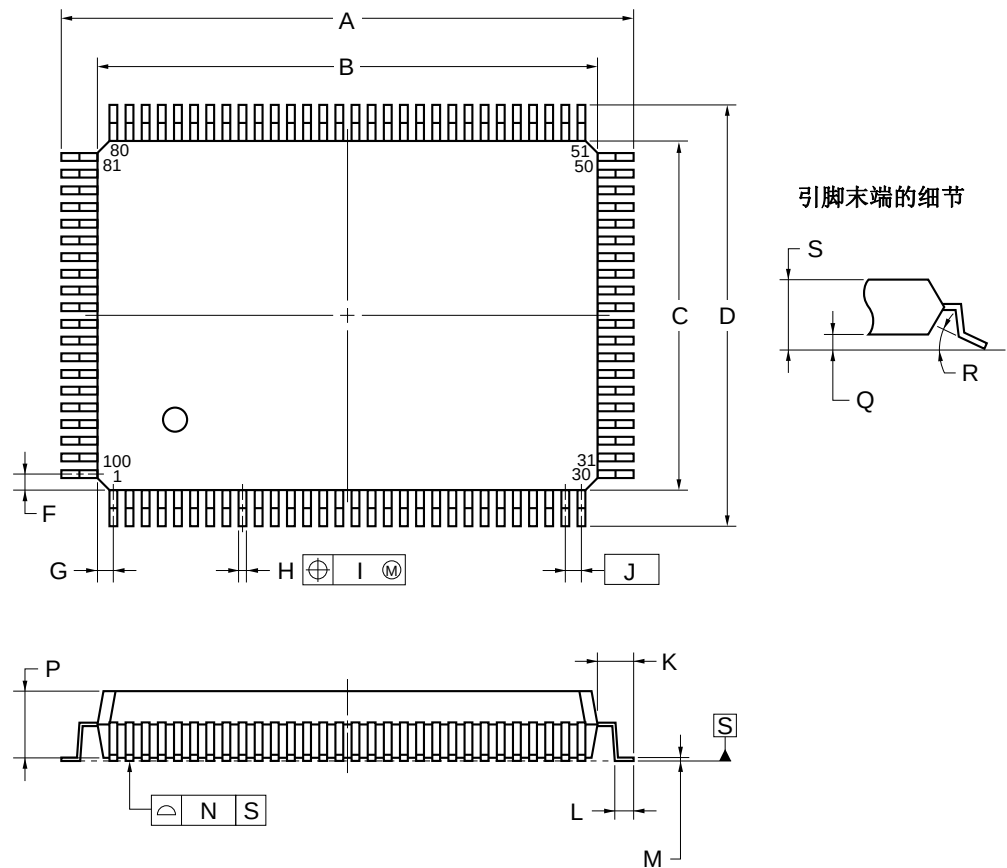
参数	符号	条件		最小值	典型值	最大值	单位
块擦除时间		f _{xx} = 32 MHz	注 1		651.3		ms
			注 2		3,081.9		ms
			注 3		6,727.7		ms
每 256 字节写入时间		f _{xx} = 32 MHz			8.7		ms
块内部校验时间		f _{xx} = 32 MHz	注 1		49.0		ms
			注 2		342.9		ms
			注 3		783.9		ms
块空白检查时间		f _{xx} = 32 MHz	注 1		22.7		ms
			注 2		159.2		ms
			注 3		364.0		ms
Flash 存储器信息设置时间		f _{xx} = 32 MHz			1.1		ms

- 注 1. 块大小 = 8 KB
 2. 块大小 = 56 KB
 3. 块大小 = 128 KB

注意事项 当对出厂产品进行初次写入时，将“擦除写入”和“只写入”两个过程都按一次重写来计数。

示例 (P: 写入, E: 擦除)
出库产品 —————→ P → E → P → E → P: 3 次重写
出库产品 → E → P → E → P → E → P: 3 次重写

100 引脚塑封 QFP（14x20）

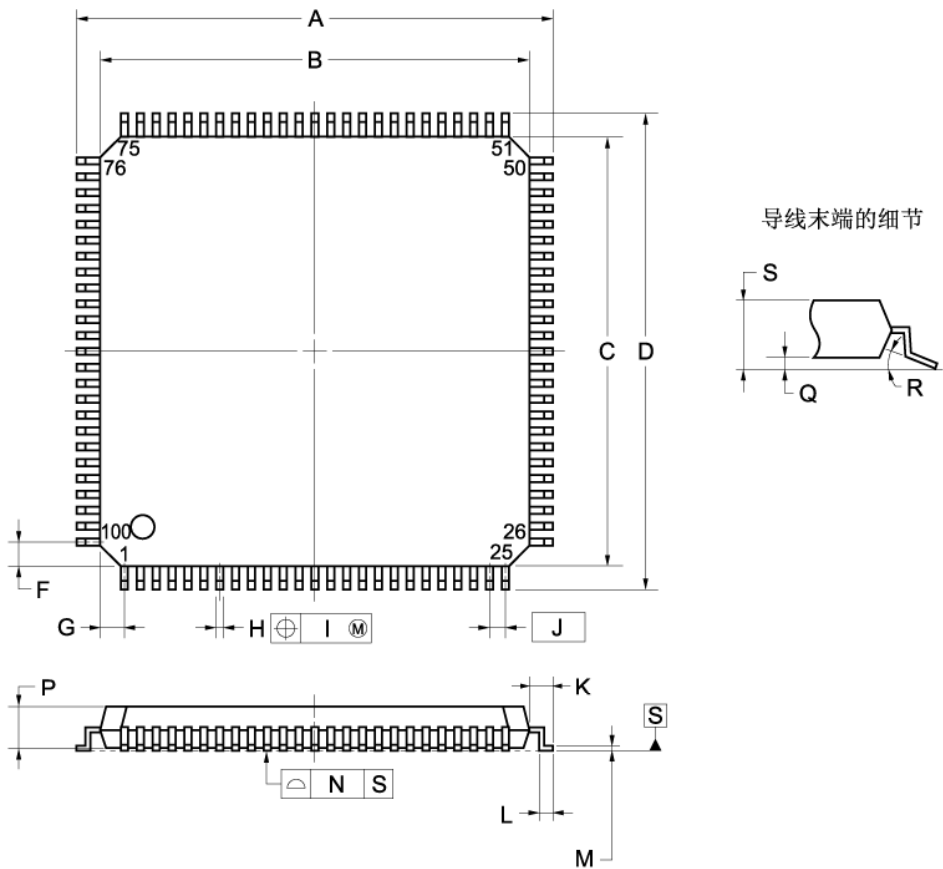


注
在最大用料条件下，每个引脚的中心线位于其理想位置的 0.13mm(TP.)以
内。

项目	毫米
A	23.2±0.2
B	20.0±0.2
C	14.0±0.2
D	17.2±0.2
F	0.825
G	0.575
H	0.32 ^{+0.08} _{-0.07}
I	0.13
J	0.65 (T.P.)
K	1.6±0.2
L	0.8±0.2
M	0.17 ^{+0.06} _{-0.05}
N	0.10
P	2.7±0.1
Q	0.125±0.075
R	3° ^{+7°} _{-3°}
S	3.0 MAX.

S100GF-65-JBT-2

100引脚塑封 LQFP (精密间距) (14x14)



注
在最大材料条件下，每个引脚的中心线位于其实际位置的 0.08 mm 以内。

项目	毫米
A	16.00±0.20
B	14.00±0.20
C	14.00±0.20
D	16.00±0.20
F	1.00
G	1.00
H	0.22 ^{+0.05} _{-0.04}
I	0.08
J	0.50 (T.P.)
K	1.00±0.20
L	0.50±0.20
M	0.17 ^{+0.03} _{-0.07}
N	0.08
P	1.40±0.05
Q	0.10±0.05
R	3° ^{+7°} _{-3°}
S	1.60 最大值

S100GC-50-8EU, 8EA-2

这些产品应该在以下推荐的条件下焊接和装配。
关于技术信息，见以下站点。

半导体设备装配手册 (<http://www.necel.com/pkg/en/mount/index.html>)

表 34-1. 表贴装配类型焊接条件 (1/2)

(1) 100 引脚塑封 LQFP (精密间距) (14 × 14)

μPD703260GC-xxx-8EA, 703260YGC-xxx-8EA, 703261GC-xxx-8EA, 703261YGC-xxx-8EA, 703262GC-xxx-8EA, 703262YGC-xxx-8EA, 703263GC-xxx-8EA, 703263YGC-xxx-8EA, 703270GC-xxx-8EA, 703270YGC-xxx-8EA, 703271GC-xxx-8EA, 703271YGC-xxx-8EA, 703272GC-xxx-8EA, 703272YGC-xxx-8EA, 703273GC-xxx-8EA, 703273YGC-xxx-8EA, 703280GC-xxx-8EA, 703280YGC-xxx-8EA, 703281GC-xxx-8EA, 703281YGC-xxx-8EA, 703282GC-xxx-8EA, 703282YGC-xxx-8EA, 703283GC-xxx-8EA, 703283YGC-xxx-8EA, 70F3261GC-8EA, 70F3261YGC-8EA, 70F3263GC-8EA, 70F3263YGC-8EA, 70F3271GC-8EA, 70F3271YGC-8EA, 70F3273GC-8EA, 70F3273YGC-8EA, 70F3281GC-8EA, 70F3281YGC-8EA, 70F3283GC-8EA, 70F3283YGC-8EA, 703262HYGC-xxx-8EA-A, 703263HYGC-xxx-8EA-A, 703272HYGC-xxx-8EA-A, 703273HYGC-xxx-8EA-A, 703282HYGC-xxx-8EA-A, 703283HYGC-xxx-8EA-A, 70F3263HYGC-8EA-A, 70F3273HYGC-8EA-A, 70F3283HYGC-8EA-A

焊接方法	焊接条件	推荐型号
红外回流	封装最高温度: 260°C, 时间: 最长 60 秒 (在 220°C 或更高), 次数: 3 次或更少, 暴露限制: 7 天 [※] (之后在 125°C 预烘 20 小时 到 72 小时)	IR60-207-3
局部加热	引脚温度: 350°C 最大值, 时间: 最长 3 秒 (每引脚排)	—

注 打开干燥的包装后, 存储温度低于 25°C, 相对湿度 65%条件下, 或减少存储时间。

注意事项 不要一起使用不同的焊接方式 (除局部加热外)。

备注 1. V850ES/SG2 和 V850ES/SG2-H 是无铅产品。
2. 有关不同于以上所推荐的焊接方法和条件, 请与日电电子销售代理联系。

表 34-1. 表贴装配类型焊接条件 (2/2)

(2) 100 引脚塑封 QFP (14 × 20)

μPD703260GF-xxx-JBT, 703260YGF-xxx-JBT, 703261GF-xxx-JBT, 703261YGF-xxx-JBT, 703270GF-xxx-JBT, 703270YGF-xxx-JBT, 703271GF-xxx-JBT, 703271YGF-xxx-JBT, 70F3261GF-JBT, 70F3261YGF-JBT, 70F3271GF-JBT, 70F3271YGF-JBT

焊接方法	焊接条件	推荐型号
红外线回流	封装最高温度: 260°C, 时间: 最长 60 秒 (在 220°C 或更高), 次数: 3 次或更少, 暴露限制: 7 天 ^注 (之后在 125°C 预烘 20 到 72 小时)	IR60-207-3
波峰焊	关于细节, 联系日电电子销售代理。	—
局部加热	引脚温度: 350°C 最大值, 时间: 最长 3 秒 (每引脚排)	—

注 打开干燥的包装后, 存储温度低于 25°C, 相对湿度 65%条件下, 或减少存储时间。

注意事项 不要一起使用不同的焊接方式 (除局部加热外)。

备注

1. V850ES/SG2 是无铅产品。
2. 有关不同于以上所推荐的焊接方法和条件, 请与日电电子销售代理联系。

附录A 开发工具

<R> 在使用 V850ES/SG2 或 V850ES/SG2-H 的系统开发中可应用如下开发工具。
图 A-1 所示为开发工具的配置。

- **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”包括以下几种操作系统：

- Windows 98
- Windows 2000
- Windows Me
- Windows XP
- Windows NT™ Ver. 4.0

图 A-1. 开发工具的组成 (1/4)

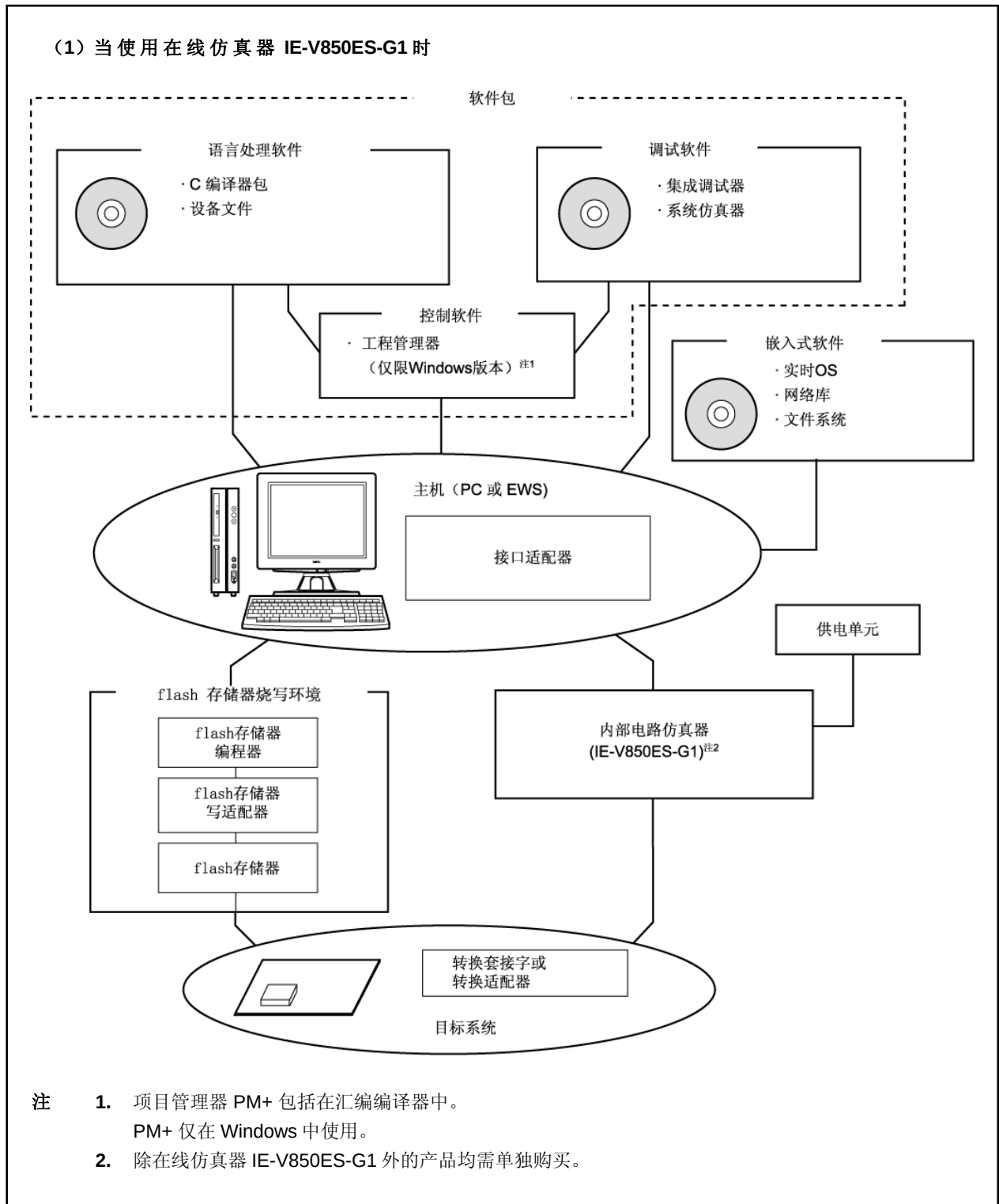


图 A-1. 开发工具的组成 (2/4)

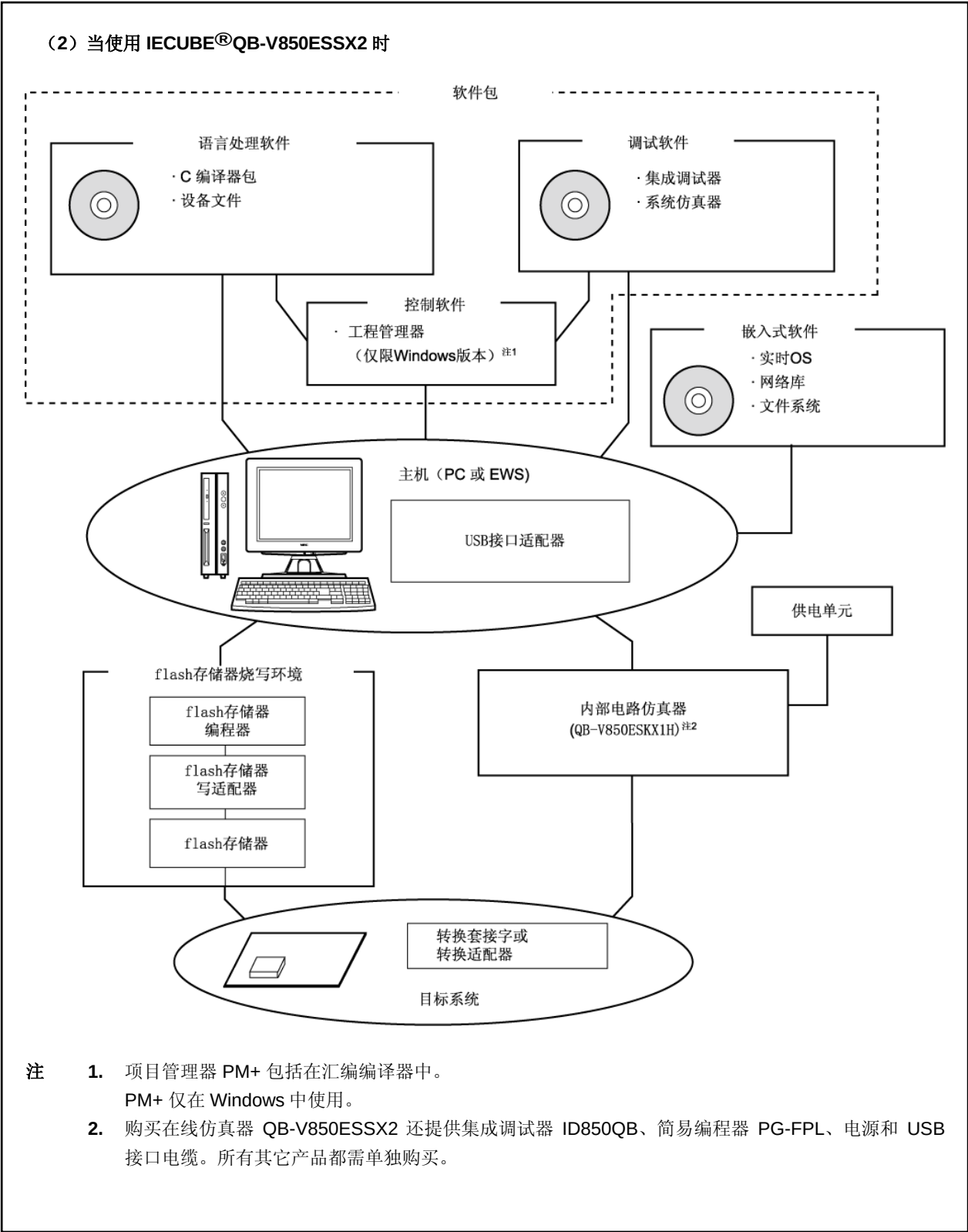


图 A-1. 开发工具的组成 (3/4)

<R>

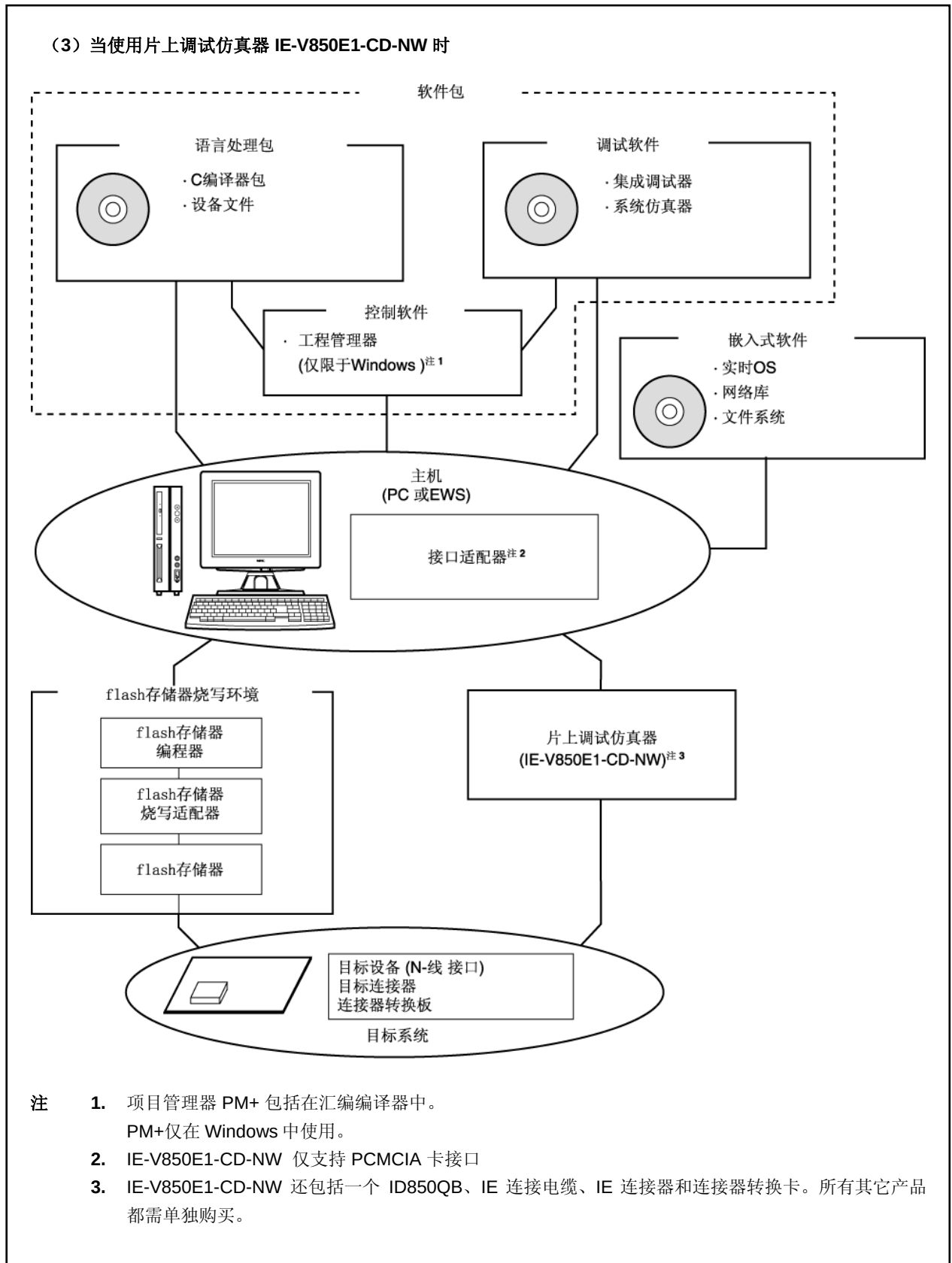
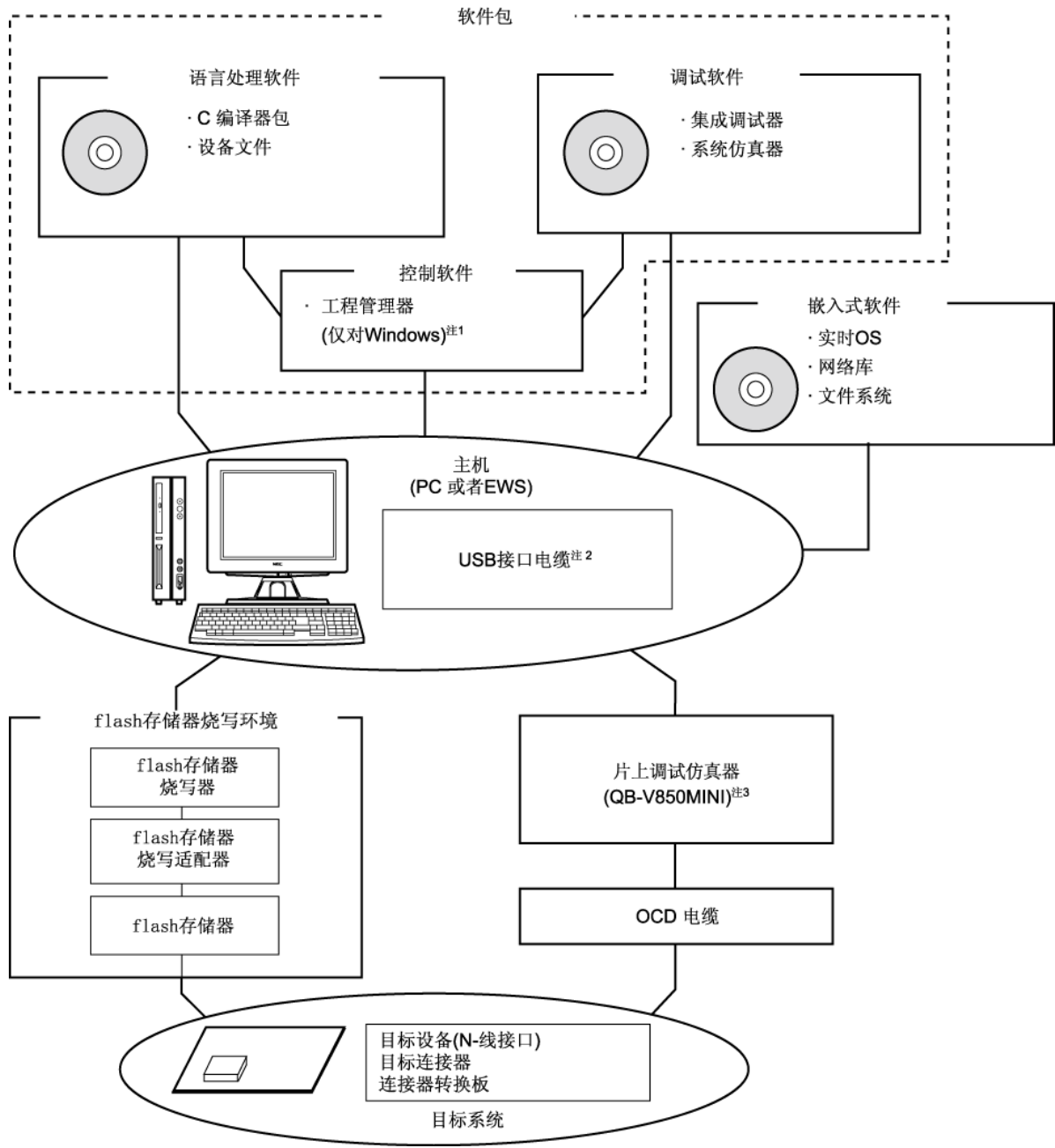


图 A-1. 开发工具的组成 (4/4)

<R>

(4) 当使用 MINICUBE® QB-V850MINI 时



- 注
1. 项目管理器 PM+ 包括在汇编编译器中。
PM+ 仅仅在 Windows 中使用。
 2. QB-V850MINI 支持 USB 接口。
 3. QB-V850MINI 除了一个 ID850QB、USB 接口电缆和 OCD 电缆外，还包括一个 KEL 连接器和 KEL 适配器作为目标连接器以及连接器转换板。所有其它目标连接器都需单独购买。

A.1 软件包

SP850 V850 系列软件包	开发工具（软件）适用于 V850 系列 产品型号：
	产品型号: μ SxxxxSP850

备注 产品型号中的 xxxx 随主机和使用的 OS 而变化。

μ SxxxxSP850

xxxx	主机	OS	存储介质
AB17	PC-9800 系列，	Windows（日文版）	CD-ROM
BB17	IBM PC/AT 兼容机	Windows（英文版）	

A.2 语言处理软件

CA850 c 编译器包	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
DF703283/DF703283H ^注 设备文件	本文件包含该设备特有的信息。 该设备文件应结合工具（CA850 或 ID850）一起使用。 相应的 OS 和主机随使用工具而变化。

<R>

<R>

注 V850ES/SG2: DF703283
V850ES/SG2-H: DF703283H

备注 产品型号中的 xxxx 随主机和使用的 OS 而变化。

μ SxxxxCA703000

xxxx	主机	OS	存储介质
AB17	PC-9800 系列，	Windows（日文版）	CD-ROM
BB17	IBM PC/AT 兼容机	Windows（英文版）	
3K17	SPARCstation™	Solaris（Rel. 2.5.1） 4.1.4）， Solaris™（Rel. 2.5.1）	

A.3 控制软件

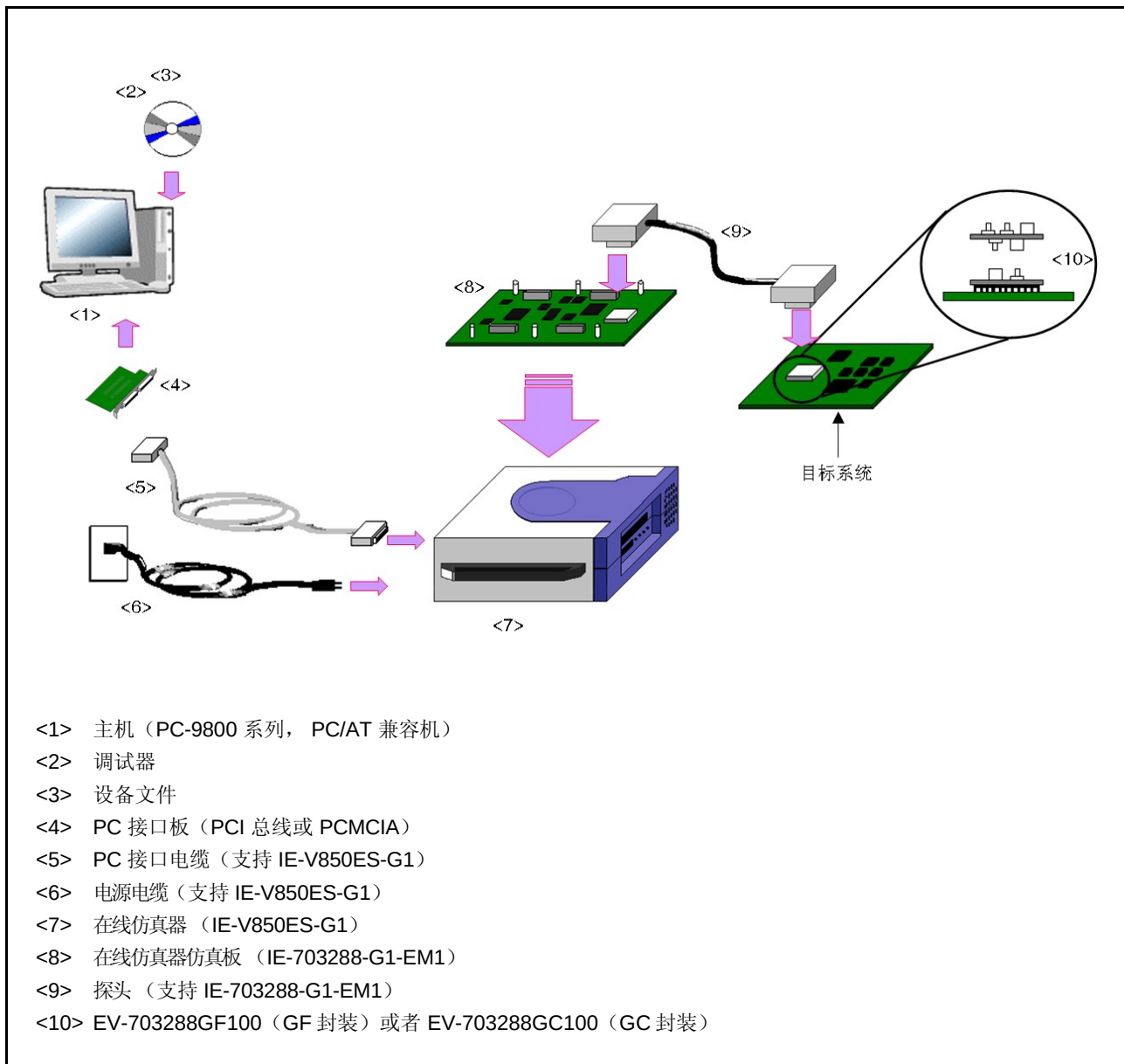
PM+ 项目管理器	这是一个控制软件，可使用户在 Windows 环境下有效地进行软件开发。用户程序开发中进行的任何操作，如启动编辑器、建立和启动调试器，都可以从 PM+ 中执行。 <注意事项> PM+ 包含在 C 编译器软件包 CA850 中。 仅在 Windows 下使用。
--------------	---

A.4 调试工具（硬件）

A.4.1 （1）当使用在线仿真器 IE-V850ES-G1 时

如下所示为连接 IE-703288-G1-EM1 到 IE-V850ES-G1，并通过它连接到主机（PC-9800 系列，PC/AT 兼容机）时的系统配置。

图 A-2. 系统配置（使用 IE-V850ES-G1）



<7> IE-V850ES-G1 在线仿真器	该在线仿真器用于使用 V850 系列产品开发的应用系统的硬件和软件调试。它支持集成调试 ID850。该仿真器应当与电源、仿真头和连接此仿真器与主机所需要的接口适配器结合使用。
<4> IE-70000-CD-IF-A PC 卡接口	当使用笔记本电脑作为主机时需要该 PC 卡和接口电缆（PCMCIA 插口兼容）。
<4> IE-70000-PCI-IF-A 接口适配器	当使用的主机是带有 PCI 总线的计算机时，需要该适配器。
<8> IE-703288-G1-EM1 仿真板	该板仿真设备的外围硬件的操作。应与在线仿真器一起使用。
<9> GXP-CABLE 仿真探头	该探头用于连接在线仿真器和目标系统。与仿真板 IE-703288-G1-EM1 一起提供。
<10> EV-703288GF100 转换适配器	此转换适配器用于连接仿真探头和目标系统板，该目标上可安装 100 引脚塑封 LQFP（GF-JBT 类型）。
<10> EV-703288GC100 转换适配器	此转换适配器用于连接仿真探头和目标系统板，该目标上可安装 100 引脚塑封 LQFP（GC-8EA 类型）。

- 备注**
1. 括号中的数字对应于图 A-2 中的数字。
 2. EV-703288GF100 和 EV-703288GC100 是 Application Corporation（应用公司）的产品。
TEL:+81-42-732-1377 Application Corporation（应用公司）

A.4.2 当使用IECUBE QB-V850ESSX2 时

当 QB-V850ESSX2 连接到主机（PC-9821 系列，PC/AT 兼容机）时的系统配置如下所示。即使没有选择另外的可选产品，也可以连接。

图 A-3. 系统配置（使用 QB-V850ESSX2） 1/2

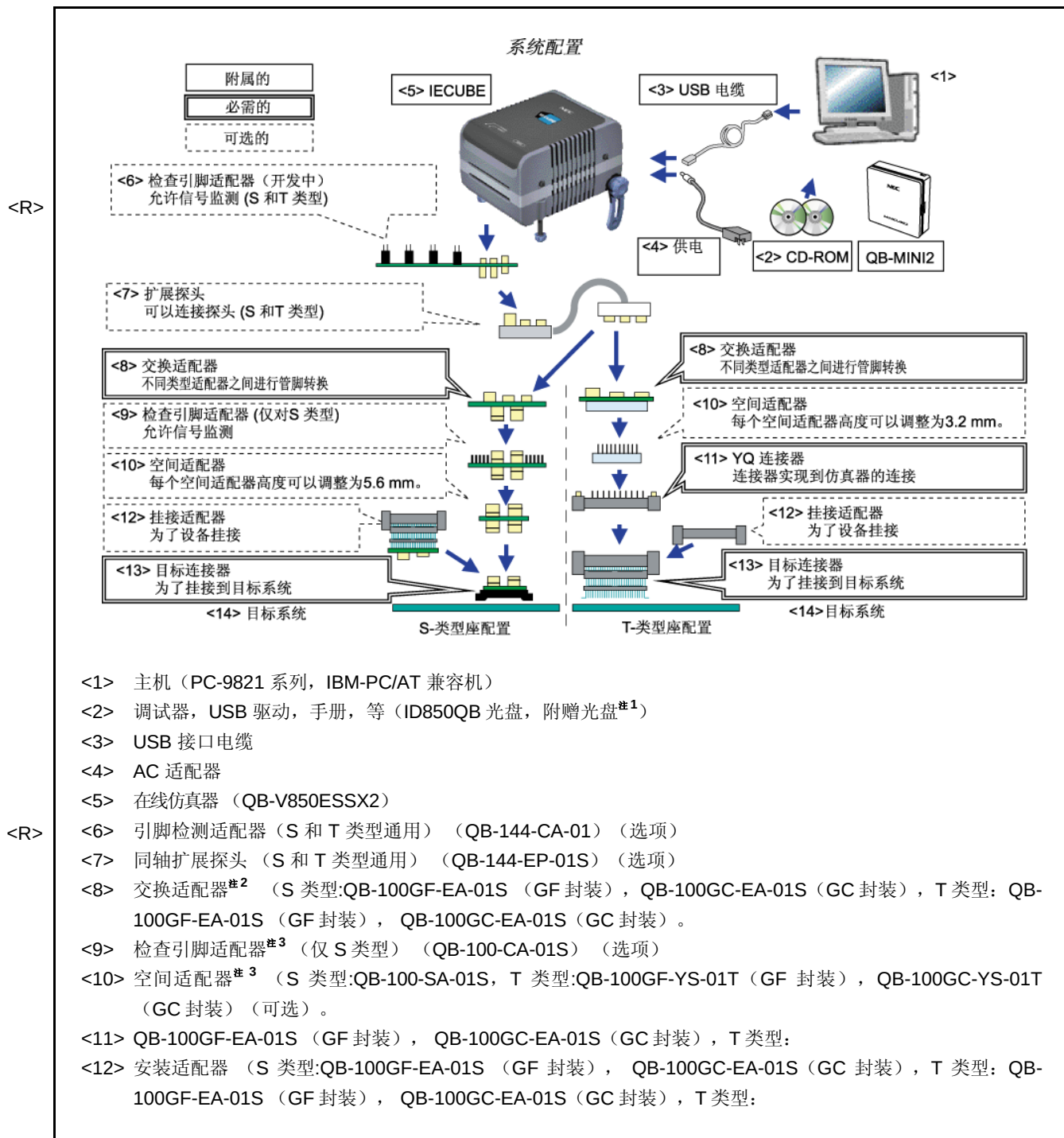


图 A-3. 系统配置 (QB-V850ESSX2 Used) (2/2)

- <13> 目标连接器² (S 类型:QB-100GF-EA-01S (GF 封装), QB-100GC-EA-01S (GC 封装), T 类型: QB-100GF-EA-01S (GF 封装), QB-100GC-EA-01S (GC 封装), T 类型:
- <14> 目标系统

- 注
1. 设备文件可从日电电子网站上获得。
<http://www.necel.com/micro/ods/eng/index.html>
 2. 根据订购的型号提供设备。
 - 订购 QB-V850ESSX2-ZZZ 时
不提供交换适配器和目标连接器。
 - 订购 QB-V850ESSX2-S100GF 时
提供 QB-100GF-EA-01S 和 QB-100GF-TC-01S。
 - 订购 QB-V850ESSX2-S100GC 时
提供 QB-100GC-EA-01S 和 QB-100GC-TC-01S。
 - 订购 QB-V850ESSX2-T100GF 时
提供 QB-100GF-EA-01T、QB-100GF-YQ-01T 和 QB-100GF-NQ-01T。
 - 订购 QB-V850ESSX2-T100GC 时
提供 QB-100GC-EA-01T、QB-100GC-YQ-01T 和 QB-100GC-NQ-01T。
 3. 当同时使用<9>和<10>时, 不必关心<9>和<10>的顺序。

<R>	<5> QB-V850ESSX2 ^注 在线仿真器	该在线仿真器用于使用 V850ES/SG2 或 V850ES/SG2-H 进行产品开发的应用系统的硬件和软件调试。它支持集成调试器 ID850QB。该仿真器应与电源和仿真探头结合使用。通过 USB 将仿真器连接到主机。
	<3> USB 接口电缆	电缆连接主机和 QB-V850ESSX2。
	<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-100GF-EA-01S , QB-100GC-EA-01S, QB-100GF-EA-01T, QB-100GC-EA-01T 交换适配器	用于执行引脚转换的适配器。 • QB-100GF-EA-01S、QB-100GF-EA-01T: 100 引脚塑封 QFP (GF-JBT 类型) • QB-100GC-EA-01S、QB-100GC-EA-01T: 100 引脚塑封 LQFP (GC-8EA 类型)
	<9> QB-100-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器等。
<R>	<10> QB-100-SA-01S, QB-100GF-YS-01T, QB-100GC-YS-01T 空间适配器	用于调整最高点的适配器。 • QB-100-SA-01S、QB-100GF-YS-01T: 100 引脚塑封 QFP (GF-JBT 类型) • QB-100-SA-01S、QB-100GC-YS-01T: 100 引脚塑封 LQFP (GC-8EA 类型)
	<12> QB-100GF-MA-01S, QB-100GC-MA-01S, QB-100GF-HQ-01T, QB-100GC-HQ-01T 挂接适配器	使用套接字挂接 V850ES/SG2 或 V850ES/SG2-H 的适配器。 • QB-100GF-MA-01S、QB-100GF-HQ-01T: 100 引脚塑封 QFP (GF-JBT 类型) • QB-100GC-MA-01S、QB-100GC-HQ-01T: 100 引脚塑封 LQFP (GC-8EA 类型)
	<13> QB-100GF-TC-01S, QB-100GC-TC-01S, QB-100GF-NQ-01T, QB-100GC-NQ-01T 目标连接器	目标系统焊接的连接器。 • QB-100GF-TC-01S、QB-100GF-NQ-01T: 100 引脚塑封 QFP (GF-JBT 类型) • QB-100GC-TC-01S、QB-100GC-NQ-01T: 100 引脚塑封 LQFP (GC-8EA 类型)

<R> 注 QB-V850ESSX2 提供电源单元，USB 接口线缆和 QB-MINI2。同时也提供集成调试器 ID850QB 作为控制软件。

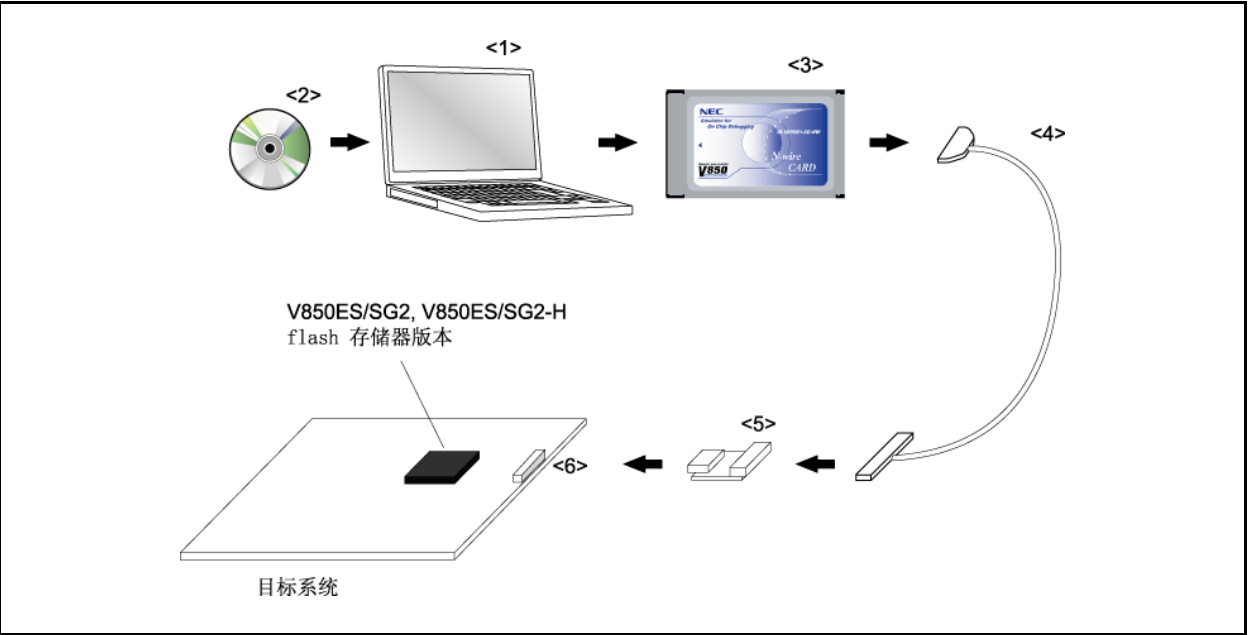
备注 括号中的数字对应于图 A-3 中的数字。

<R>

A.4.3 当使用片上调试仿真器 IE-V850E1-CD-NW 时

下图是 IE-V850E1-CD-NW 连接到主机（PC-9821 系列，PC/AT 兼容机）时的系统结构。

图 A-4. 系统配置（使用 IE-V850E1-CD-NW）



<1> 主机	配备有兼容 PCMCIA2.1/JEIDA V4.2 标准的 PCMCIA 的个人计算机。 当使用不带 PCMCIA 插槽的计算机时，须使用一个 PCI- PCMCIA 转接卡或类似设备。有关转接卡的详细内容，请咨询日电电子销售代表。
<2> CD-ROM ^{注 1}	该 CD-ROM 中包含集成调试器 ID850QB、N-Wire 检查器、设备驱动器、文件等等。 该 CD-ROM 和 IE- V850E1-CD-NW 一起提供。
<3> IE-V850E1-CD-NW 片上调试仿真器	当使用 V850ES/SG2 开发应用系统时，用片上调试仿真器来调试硬件和软件。它支持集成调试器 ID850QB。
<4> IE-V850E1-CD-NW 连接电缆	该连接电缆用来连接 IE-V850E1-CD-NW 和目标系统。随 IE-V850E1-CD-NW 附带提供。电缆长度约为 50 厘米。
<5> 连接器转化卡 KEL 适配器	随 IE-V850E1-CD-NW 附带提供。
<6> IE-V850E1-CD-NW 连接器 KEL 连接器 ^{注 2}	8830E-026-170S （随 IE-V850E1-CD-NW 附带提供）。 8830E-026-170L （单独购买）

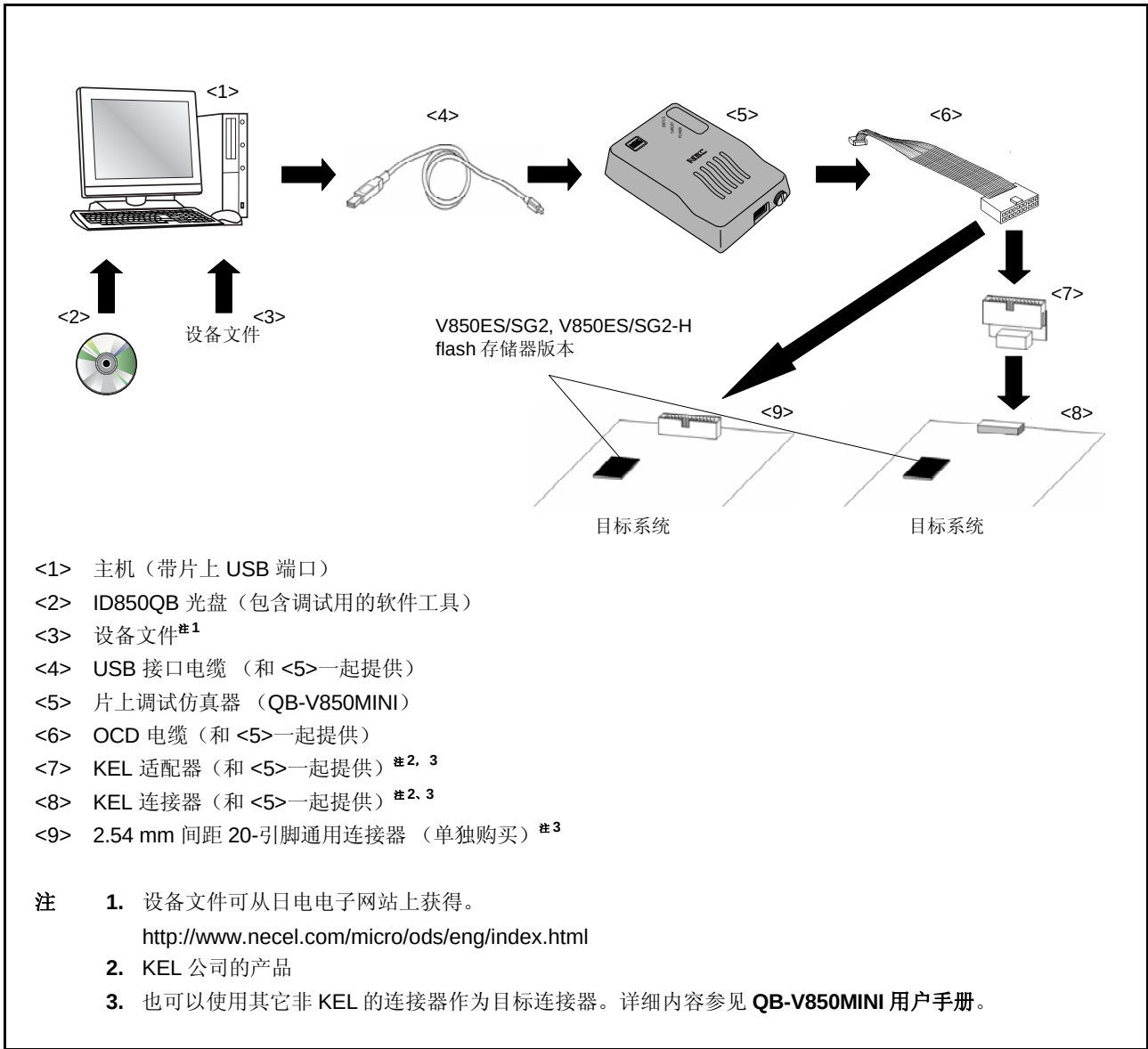
注 1. 设备文件可从日电电子网站上获得。
<http://www.necel.com/micro/ods/eng/index.html>
2. KEL 公司的产品

备注 括号中的数字对应于图 A-4 中的数字。

<R> A.4.4 当使用 MINICUBE QB-V850MINI 时

下图是 QB-V850MINI 连接到主机（PC-9821 系列，PC/AT 兼容机）时的系统结构。

图 A-5. 系统结构（使用 QB-V850MINI）



A.5 调试工具（软件）

<R>	支持 V850ES/Sx2 的 SM+ 系统仿真器	该系统模拟器支持 V850 微控制器。SM+ 是基于 Windows 的软件。在主机上模拟目标系统的操作时，用于调试 C 源程序或汇编程序。使用 SM+ 可以执行应用逻辑测试和脱离硬件开发的性能测试，因此提供了很高的开发效率和软件质量。SM+ 应与设备文件一起使用（独立销售）。
		产品型号: μ SxxxxSM703289-B
<R>	ID850 集成调试器 （支持在线仿真器 IE-V850ES-G1）	该调试器支持 V850 系列微控制器的在线仿真。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显示跟踪结果，反汇编显示，显示存储器追踪结果。该功能需要使用设备文件（独立销售）。
	ID850QB 集成调试器 （支持在线仿真器 QB-V850ESSX2）	产品型号: μ SxxxxID703000, μ SxxxxID703000-GC（ID850）

<R> **备注** 产品型号中的 xxxx 随使用的操作系统而变化。

μ SxxxxSM703289-B
 μ SxxxxID703000
 μ SxxxxID703000-GC

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.6 嵌入式软件

RX850, RX850 Pro 实时操作系统	RX850 和 RX850 Pro 是符合 μ TRON 3.0 规范的实时操作系统。 提供一个可以产生多个信息表格的工具（配置器）。 RX850 Pro 比 RX850 功能更强大。
	产品型号: μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$ (RX850) μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$ (RX850 Pro)
V850mini-NET （临时名称） （网络库）	这是一个符合 RFC 的网络库。 包含精简的 TCP/IP 协议设计，仅需要很小的存储器。 除了标准的 TCP/IP 设定，同时提供对 HTTP 服务器，SMTP 客户端，和 POP 客户端 的支持。
RX-FS850 （文件系统）	这是一个 FAT 文件系统。 它支持 CD-ROM 文件系统功能的文件系统。 此文件系统和实时操作系统 RX850 Pro 配合使用。

注意事项 如需购买 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 $\Delta\Delta\Delta\Delta$ 随使用的操作系统而变化。

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	量产目标	10 万 单元
001M		100 万 单元
010M		1000 万 单元
S01	源程序	量产用的目标源程序

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	SPARC 工作站	Solaris （Rel. 2.5.1）	

A.7flash 存储器写入工具

<R>	Flashpro IV （产品型号：PG-FP4） Flash 编程器	Flash 编程器专用于片上内置 flash 存储器的微控制器。
	Flashpro IV （产品型号：PG-FP5） Flash 存储器编程器	
	FA-100GF-3BA-A Flash 存储器写入适配器	使用连接到 Flashpro IV 的 Flash 存储器写入适配器。 • FA-100GF-3BA-A: 支持 100-引脚 塑封 QFP （GF-JBT 类型）
	FA-100GC-8EU-A Flash 存储器写入适配器	使用连接到 Flashpro IV 的 Flash 存储器写入适配器。 • FA-100GC-8EU-A: 支持 100-引脚塑封 LQFP （GC-8EA 类型）

备注 FA-100GF-3BA-A 与 FA-100GC-8EU-A 是 Naito Densei Machida Mfg. Co., Ltd.的产品。

电话:+81-42-750-4172 Naito Densei Machida Mfg. Co., Ltd.

附录 B V850ES/SG2 和 V850ES/SG2-H 之间的主要差别

V850ES/SG2 和 V850ES/SG2-H 的差别如下所示。详细内容，参见各相关章节。对于 V850ES/SG2 和 V850ES/SG2-H 每部分的数据，参见表 1-1 的注释。

表 B-1. V850ES/SG2 和 V850ES/SG2-H 之间的主要差别 (1/2)

主要差别		V850ES/SG2	V850ES/SG2-H	参见:
介绍	最小指令执行时间	50 ns	31.25 ns	1.2
	封装	100-pin 塑封 QFP (14 × 20) 100-pin 塑封 LQFP (14 × 14)	100-pin 塑封 LQFP (14 × 14)	
CPU 功能	内部 flash 存储器	384/640 KB	640 KB	3.4.4 (1)
	内部 ROM	256/384/512/640 KB	512/640 KB	3.4.4 (2)
	内部 RAM	24/32/40/48 KB	40/48 KB	
	特殊寄存器	PSC, CKC, PCC, CLM, RESF, LVIM, RAMS, OCDM 寄存器	PSC, CKC, PCC, CLM, RESF, RAMS, OCDM 寄存器	3.4.8
	设置 VSWC 的值	00H/01H	00H/01H/11H	3.4.9 (1) (a)
总线控制功能	访问时钟数	内部 ROM (32 位) 取指: 2 操作数据访问内部 ROM (32 位): 3	内部 ROM (32 位) 取指: 3 操作数据访问内部 ROM (32 位): 4	5.5.1
	DWC0 寄存器: 数据等待的注意事项	无	提供	5.6.1 (1)
	AWC 寄存器: 地址保持等待, 地址建立等待的注意事项	无	提供	5.6.4 (1)
主时钟振荡器频率	时钟直通模式	$f_x = 2.5$ 到 10 MHz ($f_{xx} = 2.5$ 到 10 MHz)	$f_x = 2.5$ 到 8 MHz ($f_{xx} = 2.5$ 到 8 MHz)	6.1
	PLL 模式	$f_x = 2.5$ 到 5 MHz ($f_{xx} = 10$ 到 20 MHz)	$f_x = 2.5$ 到 5 MHz ($\times 4: f_{xx} = 10$ 到 20 MHz) $f_x = 2.5$ 到 4 MHz ($\times 8: f_{xx} = 20$ 到 32 MHz)	
A/D 转换器: ADA0M1.ADA0FR3 位		无	已提供	13.4 (2)
IEBus 控制器时钟		f_{xx} , $f_{xx}/2$, $f_{xx}/3$	f_{xx} , $f_{xx}/2$, $f_{xx}/3$, $f_{xx}/4$, $f_{xx}/5$	18.3 (17)
DMA 功能 DTFRn 寄存器		改变设置的过程不同:		20.3 (6)
中断	可屏蔽中断个数	47/51	46/50	22.1
	xxICn 寄存器约束	无	已提供	22.3.4
	中断响应时间 (内部中断 (最大值))	6	7	22.7

表 B-1. V850ES/SG2 和 V850ES/SG2-H 之间的主要差别 (2/2)

主要差别		V850ES/SG2	V850ES/SG2-H	参见:
复位功能:	RESF.LVIRF 位	已提供	无	25.2 (1)
	低电压检测器引起的中断	已提供	无	25.3.3
	固件操作时间 (秒)	$14,974 \times (1/f_x)$	$11,994 \times (1/f_x)$	25.3.5 (2)
低电压检测器		已提供	无	第 27 章
flash 存储器	块结构	内部 flash 存储器 384/640 KB 块 0 到 3: 每块 28 KB 块 4 到 7: 每块 4 KB 块 8 到最后块: 每块 64 KB	内部 flash 存储器 640 KB 块 0, 2: 每块 56 KB 块 1 到 3: 每块 8 KB 块 4 到 7: 每块 128 KB	30.2
	引导区	56 KB	64 KB	
电气特性	操作条件	当使用主时钟 (ADC, DAC 停止), 副时钟 (ADC, DAC 停止) $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.85$ 到 3.6 V $BV_{DD} = 2.7$ 到 3.6 V 当使用主时钟 (ADC, DAC 运行) $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 3.0$ 到 3.6 V $BV_{DD} = 2.7$ 到 3.6 V	当使用主时钟, 副时钟: $V_{DD} = EV_{DD} = BV_{DD} = AV_{REF0} = AV_{REF1} = 3.0$ 到 3.6 V	32.3
	主时钟振荡器特性	振荡器频率(f_x) = 10 MHz (最大值)	振荡器频率(f_x) = 8 MHz (最大值)	32.4.1
	PLL 特性	使用输入频率, $\times 8$ 模式 $f_x = 2.5$ MHz (最大值)	使用输入频率, $\times 8$ 模式 $f_x = 4$ MHz (最大值)	32.4.3
		使用输出频率, $\times 8$ 模式 $f_{xx} = 20$ MHz (最大值)	使用输出频率, $\times 8$ 模式 $f_{xx} = 32$ MHz (最大值)	
	电阻特性 (输入电压 (V_{DD}))	$V_{DD} = 2.85$ 到 3.6 V	$V_{DD} = 3.0$ 到 3.6 V	32.5
	DC 特性 (电源电流)	每个参数值不同。		32.6.2
	CLKOUT 输出时序 (输出周期)	50 ns (最小值)	31.25 ns (最小值)	32.8.1
	总线时序	每个参数值不同。		32.8.2
	CSIB 时序 (从模式)	SCKBn $\downarrow$ 到 $\overline{SOBn}$ 输出延时时间 (t_{KS02}): 30ns(最大值)	SCKBn $\downarrow$ 到 $\overline{SOBn}$ 输出延时时间 (t_{KS02}): 35ns(最大值)	32.9 (6) (b)
	LVI 电路特性	已提供	无	32.9 (12)
	flash 存储器编程特性 (基本特性, 操作频率)	$f_{CPU} = 2.5$ 到 20 MHz	$f_{CPU} = 2.5$ 到 32 MHz	32.10 (1)
	flash 存储器编程特性 (基本特性, 供电电压)	$V_{DD} = 2.85$ 到 3.6 V	$V_{DD} = 3.0$ 到 3.6 V	
	flash 存储器编程特性 (编程特性)	每个参数值不同。		32.10 (3)
要求焊接条件		红外回流 波形焊接	红外回流	第 34 章

附录 C 寄存器索引

(1/12)

符号	名称	单元	页码
ADA0CR0	A/D 转换结果寄存器 0	ADC	473
ADA0CR0H	A/D 转换结果寄存器 0H	ADC	473
ADA0CR1	A/D 转换结果寄存器 1	ADC	473
ADA0CR10	A/D 转换结果寄存器 10	ADC	473
ADA0CR10H	A/D 转换结果寄存器 10H	ADC	473
ADA0CR11	A/D 转换结果寄存器 11	ADC	473
ADA0CR11H	A/D 转换结果寄存器 11H	ADC	473
ADA0CR1H	A/D 转换结果寄存器 1H	ADC	473
ADA0CR2	A/D 转换结果寄存器 2	ADC	473
ADA0CR2H	A/D 转换结果寄存器 2H	ADC	473
ADA0CR3	A/D 转换结果寄存器 3	ADC	473
ADA0CR3H	A/D 转换结果寄存器 3H	ADC	473
ADA0CR4	A/D 转换结果寄存器 4	ADC	473
ADA0CR4H	A/D 转换结果寄存器 4H	ADC	473
ADA0CR5	A/D 转换结果寄存器 5	ADC	473
ADA0CR5H	A/D 转换结果寄存器 5H	ADC	473
ADA0CR6	A/D 转换结果寄存器 6	ADC	473
ADA0CR6H	A/D 转换结果寄存器 6H	ADC	473
ADA0CR7	A/D 转换结果寄存器 7	ADC	473
ADA0CR7H	A/D 转换结果寄存器 7H	ADC	473
ADA0CR8	A/D 转换结果寄存器 8	ADC	473
ADA0CR8H	A/D 转换结果寄存器 8H	ADC	473
ADA0CR9	A/D 转换结果寄存器 9	ADC	473
ADA0CR9H	A/D 转换结果寄存器 9H	ADC	473
ADA0M0	A/D 转换模式寄存器 0	ADC	466
ADA0M1	A/D 转换模式寄存器 1	ADC	468
ADA0M2	A/D 转换模式寄存器 2	ADC	471
ADA0PFM	电压不足比较模式寄存器	ADC	475
ADA0PFT	电压不足比较阈值寄存器	ADC	476
ADA0S	A/D 转换通道指定寄存器	ADC	472
ADIC	中断控制寄存器	INTC	933
AWC	地址等待控制寄存器	BCU	200
BCC	总线周期控制寄存器	BCU	201
BCR	IEBus 控制寄存器	IEBus	693
BPC	外围 I/O 区域选择控制寄存器	BCU	89
BSC	总线宽度配置寄存器	BCU	189
C0BRP	CAN0 模块位速率预分频寄存器	CAN	807
C0BTR	CAN0 比特速率寄存器	CAN	808
C0CTRL	CAN0 模块控制寄存器	CAN	797

(2/12)

符号	名称	单元	页码
C0ERC	CAN0 模块错误计数寄存器	CAN	803
C0GMABT	CAN0 全局自动 block 发送控制寄存器	CAN	792
C0GMABTD	CAN0 全局自动 block 发送控制寄存器	CAN	794
C0GMCS	CAN0 全局时钟选择 寄存器	CAN	791
C0GMCTRL	CAN0 全局控制寄存器	CAN	789
C0IE	CAN0 模块中断允许寄存器	CAN	804
C0INFO	CAN0 模块信息寄存器	CAN	802
C0INTS	CAN0 模块中断状态寄存器	CAN	806
C0LEC	CAN0 模块最终错误信息寄存器	CAN	801
C0LIPT	CAN0 模块最终发送指标寄存器	CAN	810
C0LOPT	CAN0 模块最终发送指标寄存器	CAN	812
C0MASK1H	CAN0 模块屏蔽 1 寄存器 H	CAN	795
C0MASK1L	CAN0 模块屏蔽 1 寄存器 L	CAN	795
C0MASK2H	CAN0 模块屏蔽 2 寄存器 H	CAN	795
C0MASK2L	CAN0 模块屏蔽 2 寄存器 L	CAN	795
C0MASK3H	CAN0 模块屏蔽 3 寄存器 H	CAN	795
C0MASK3L	CAN0 模块屏蔽 3 寄存器 L	CAN	795
C0MASK4H	CAN0 模块屏蔽 4 寄存器 H	CAN	795
C0MASK4L	CAN0 模块屏蔽 4 寄存器 L	CAN	795
C0MCONFm	CAN0 消息配置寄存器 m	CAN	819
C0MCTRLm	CAN0 消息控制寄存器 m	CAN	821
C0MDATA01m	CAN0 消息数据字节 01 寄存器 m	CAN	816
C0MDATA0m	CAN0 消息数据字节 0 寄存器 m	CAN	816
C0MDATA1m	CAN0 消息数据字节 1 寄存器 m	CAN	816
C0MDATA23m	CAN0 消息数据字节 23 寄存器 m	CAN	816
C0MDATA2m	CAN0 消息数据字节 2 寄存器 m	CAN	816
C0MDATA3m	CAN0 消息数据字节 3 寄存器 m	CAN	816
C0MDATA45m	CAN0 消息数据字节 45 寄存器 m	CAN	816
C0MDATA4m	CAN0 消息数据字节 4 寄存器 m	CAN	816
C0MDATA5m	CAN0 消息数据字节 5 寄存器 m	CAN	816
C0MDATA67m	CAN0 消息数据字节 67 寄存器 m	CAN	816
C0MDATA6m	CAN0 消息数据字节 6 寄存器 m	CAN	816
C0MDATA7m	CAN0 消息数据字节 7 寄存器 m	CAN	816
C0MDLCm	CAN0 消息数据长度代码寄存器 m	CAN	818
C0MIDHm	CAN0 消息 ID 寄存器 mH	CAN	820
C0MIDLm	CAN0 消息 ID 寄存器 mL	CAN	820
C0RGPT	CAN0 模块接收历史列表寄存器	CAN	811
C0TGPT	CAN0 模块传输历史列表寄存器	CAN	813
C0TS	CAN0 模块时间标记寄存器	CAN	814
CB0CTL0	CSIB0 控制寄存器 0	CSIB	547
CB0CTL1	CSIB0 控制寄存器 1	CSIB	551
CB0CTL2	CSIB0 控制寄存器 2	CSIB	552
CB0RIC	中断控制寄存器	INTC	933

备注 m = 00 到 31 符号

符号	名称	单元	页码
CB0RX	CSIB0 接收数据寄存器	CSIB	546
CB0RXL	CSIB0 接收数据寄存器 L	CSIB	546
CB0STR	CSIB0 状态寄存器	CSIB	554
CB0TIC	中断控制寄存器	INTC	933
CB0TX	CSIB0 发送寄存器	CSI	546
CB0TXL	CSIB0 发送寄存器 L	CSI	546
CB1CTL0	CSIB1 控制寄存器 0	CSI	547
CB1CTL1	CSIB1 控制寄存器 1	CSI	551
CB1CTL2	CSIB1 控制寄存器 2	CSI	552
CB1RIC	中断控制寄存器	INTC	933
CB1RX	CSIB1 接收数据寄存器	CSI	546
CB1RXL	CSIB1 接收数据寄存器 L	CSI	546
CB1STR	CSIB1 状态寄存器	CSI	554
CB1TIC	中断控制寄存器	INTC	933
CB1TX	CSIB1 发送寄存器	CSI	546
CB1TXL	CSIB1 发送寄存器 L	CSI	546
CB2CTL0	CSIB2 控制寄存器 0	CSI	547
CB2CTL1	CSIB2 控制寄存器 1	CSI	551
CB2CTL2	CSIB2 控制寄存器 2	CSI	552
CB2RIC	中断控制寄存器	INTC	933
CB2RX	CSIB2 接收数据寄存器	CSI	546
CB2RXL	CSIB2 接收数据寄存器 L	CSI	546
CB2STR	CSIB2 状态寄存器	CSI	554
CB2TIC	中断控制寄存器	INTC	933
CB2TX	CSIB2 发送寄存器	CSI	546
CB2TXL	CSIB2 发送寄存器 L	CSI	546
CB3CTL0	CSIB3 控制寄存器 0	CSI	547
CB3CTL1	CSIB3 控制寄存器 1	CSI	551
CB3CTL2	CSIB3 控制寄存器 2	CSI	552
CB3RIC	中断控制寄存器	INTC	933
CB3RX	CSIB3 接收数据寄存器	CSI	546
CB3RXL	CSIB3 接收数据寄存器 L	CSI	546
CB3STR	CSIB3 状态寄存器	CSI	554
CB3TIC	中断控制寄存器	INTC	933
CB3TX	CSIB3 发送寄存器	CSI	546
CB3TXL	CSIB3 发送寄存器 L	CSI	546
CB4CTL0	CSIB4 控制寄存器 0	CSI	547
CB4CTL1	CSIB4 控制寄存器 1	CSI	551
CB4CTL2	CSIB4 控制寄存器 2	CSI	552
CB4RIC	中断控制寄存器	INTC	933
CB4RX	CSIB4 接收数据寄存器	CSI	546
CB4RXL	CSIB4 接收数据寄存器 L	CSI	546
CB4STR	CSIB4 状态寄存器	CSI	554
CB4TIC	中断控制寄存器	INTC	933
CB4TX	CSIB4 发送寄存器	CSI	546

(4/12)

符号	名称	单元	页码
CB4TXL	CSIB4 发送寄存器 L	CSI	546
CCLS	CPU 操作时钟状态寄存器	CG	218
CCR	IEBus 通讯计数寄存器	IEBus	720
CDR	IEBus 控制数据寄存器	IEBus	711
CKC	时钟控制寄存器	CG	221
CLM	时钟监视器模式寄存器	CLM	995
CORAD0	修正地址寄存器 0	ROMC	1009
CORAD0H	修正地址寄存器 0H	ROMC	1009
CORAD0L	修正地址寄存器 0L	ROMC	1009
CORAD1	修正地址寄存器 1	ROMC	1009
CORAD1H	修正地址寄存器 1H	ROMC	1009
CORAD1L	修正地址寄存器 1L	ROMC	1009
CORAD2	修正地址寄存器 2	ROMC	1009
CORAD2H	修正地址寄存器 2H	ROMC	1009
CORAD2L	修正地址寄存器 2L	ROMC	1009
CORAD3	修正地址寄存器 3	ROMC	1009
CORAD3H	修正地址寄存器 3H	ROMC	1009
CORAD3L	修正地址寄存器 3L	ROMC	1009
CORCN	修正控制寄存器	ROMC	1011
CRCD	CRC 数据寄存器	CRC	913
CRCIN	CRC 输入寄存器	CRC	913
CTBP	CALLT 基准指针	CPU	63
CTPC	CALLT 执行状态保存寄存器	CPU	62
CTPSW	CALLT 执行状态保存寄存器	CPU	62
DA0CS0	D/A 转换值设置寄存器 0	DAC	501
DA0CS1	D/A 转换值设置寄存器 1	DAC	501
DA0M	D/A 转换模式寄存器	DAC	500
DADC0	DMA 地址控制寄存器 0	DMAC	894
DADC1	DMA 地址控制寄存器 1	DMAC	894
DADC2	DMA 地址控制寄存器 2	DMAC	894
DADC3	DMA 地址控制寄存器 3	DMAC	894
DBC0	DMA 发送计数寄存器 0	DMAC	893
DBC1	DMA 发送计数寄存器 1	DMAC	893
DBC2	DMA 发送计数寄存器 2	DMAC	893
DBC3	DMA 发送计数寄存器 3	DMAC	893
DBPC	执行/调试陷阱状态保存寄存器	CPU	63
DBPSW	执行/调试陷阱状态保存寄存器	CPU	63
DCHC0	DMA 通道控制寄存器 0	DMAC	895
DCHC1	DMA 通道控制寄存器 1	DMAC	895
DCHC2	DMA 通道控制寄存器 2	DMAC	895
DCHC3	DMA 通道控制寄存器 3	DMAC	895
DDA0H	DMA 目的地址寄存器 0H	DMAC	892
DDA0L	DMA 目的地址寄存器 0L	DMAC	892
DDA1H	DMA 目的地址寄存器 1H	DMAC	892
DDA1L	DMA 目的地址寄存器 1L	DMAC	892

符号	名称	单元	页码
DDA2H	DMA 目的地址寄存器 2H	DMAC	892
DDA2L	DMA 目的地址寄存器 2L	DMAC	892
DDA3H	DMA 目的地址寄存器 3H	DMAC	892
DDA3L	DMA 目的地址寄存器 3L	DMAC	892
DLR	IEBus 电报长度寄存器	IEBus	716
DMAIC0	中断控制寄存器	INTC	933
DMAIC1	中断控制寄存器	INTC	933
DMAIC2	中断控制寄存器	INTC	933
DMAIC3	中断控制寄存器	INTC	933
DR	IEBus 数据寄存器	IEBus	717
DSA0H	DMA 源地址寄存器 0H	DMAC	891
DSA0L	DMA 源地址寄存器 0H	DMAC	891
DMA DSA1H	DMA 源地址寄存器 1H	DMAC	891
DSA1L	DMA 目的地址寄存器 1L	DMAC	891
DMA DSA2H	DMA 源地址寄存器 2H	DMAC	891
DSA2L	DMA 源地址寄存器 2L	DMAC	891
DMA DSA3H	DMA 源地址寄存器 3H	DMAC	891
DSA3L	DMA 源地址寄存器 3L	DMAC	891
DMA DTFR0	DMA 触发源寄存器 0	DMAC	896
DTFR1	DMA 触发源寄存器 1	DMAC	896
DTFR2	DMA 触发源寄存器 2	DMAC	896
DTFR3	DMA 触发源寄存器 3	DMAC	896
DWC0	数据等待控制寄存器 0	BCU	197
ECR	中断源寄存器	CPU	60
EIPC	中断 状态保存寄存器	CPU	59
EIPSW	中断 状态保存寄存器	CPU	59
ERRIC	中断控制寄存器	INTC	933
ERRIC0	中断控制寄存器	INTC	933
ESR	IEBus 错误状态寄存器	IEBus	705
EXIMC	外部总线接口模式控制寄存器	BCU	187
FEPC	NMI 状态保存寄存器	CPU	60
FEPSW	NMI 状态保存寄存器	CPU	60
FSR	IEBus 域状态寄存器	IEBus	718
IEIC1	中断控制寄存器	INTC	933
IEIC2	中断控制寄存器	INTC	933
IIC0	IIC 移位寄存器 0	I ² C	615
IIC1	IIC 移位寄存器 1	I ² C	615
IIC2	IIC 移位寄存器 2	I ² C	615
IICC0	IIC 控制寄存器 0	I ² C	601
IICC1	IIC 控制寄存器 1	I ² C	601
IICC2	IIC 控制寄存器 2	I ² C	601
IICCL0	IIC 时钟选择寄存器 0	I ² C	611
IICCL1	IIC 时钟选择寄存器 1	I ² C	611
IICCL2	IIC 时钟选择寄存器 2	I ² C	611
IICF0	IIC 标志寄存器 0	I ² C	609

(6/12)

符号	名称	单元	页码
IICF1	IIC 标志寄存器 1	I ² C	609
IICF2	IIC 标志寄存器 2	I ² C	609
IICIC0	中断控制寄存器	INTC	933
IICIC1	中断控制寄存器	INTC	933
IICIC2	中断控制寄存器	INTC	933
IICS0	IIC 状态寄存器 0	I ² C	606
IICS1	IIC 状态寄存器 1	I ² C	606
IICS2	IIC 状态寄存器 2	I ² C	606
IICX0	IIC 功能扩展寄存器 0	I ² C	612
IICX1	IIC 功能扩展寄存器 1	I ² C	612
IICX2	IIC 功能扩展寄存器 2	I ² C	612
IMR0	中断掩码寄存器 0	INTC	936
IMR0H	中断掩码寄存器 0H	INTC	936
IMR0L	中断掩码寄存器 0L	INTC	936
IMR1	中断掩码寄存器 1	INTC	936
IMR1H	中断掩码寄存器 1H	INTC	936
IMR1L	中断掩码寄存器 1L	INTC	936
IMR2	中断掩码寄存器 2	INTC	936
IMR2H	中断掩码寄存器 2H	INTC	936
IMR2L	中断掩码寄存器 2L	INTC	936
IMR3	中断掩码寄存器 3	INTC	936
IMR3H	中断掩码寄存器 3H	INTC	936
IMR3L	中断掩码寄存器 3L	INTC	936
INTF0	外部中断下降沿指定寄存器 0	INTC	948
INTF3	外部中断下降沿指定寄存器 3	INTC	949
INTF9H	外部中断下降沿指定寄存器 9H	INTC	950
INTR0	外部中断下降沿指定寄存器 0	INTC	948
INTR3	外部中断下降沿指定寄存器 3	INTC	949
INTR9H	外部中断上升沿指定寄存器 9H	INTC	950
ISPR	当前优先级寄存器	INTC	938
ISR	IEBus 中断 状态寄存器	IEBus	702
KRIC	中断控制寄存器	INTC	933
KRM	按键返回模式寄存器	KR	957
LOCKR	锁定寄存器	CG	222
LVIIC	中断控制寄存器	INTC	933
LVIM	低电压检测寄存器	LVI	1000
LVIS	低电压检测等级选择寄存器	LVI	1001
NFC	噪声消除控制寄存器	INTC	951
OCDM	片上调试模式寄存器	Debug	1046
OCKS0	IIC 时钟选择寄存器 0	I ² C	615
OCKS1	IIC 时钟选择寄存器 1	I ² C	615
OCKS2	IEBus 时钟选择寄存器	IEBus	721
OSTS	振荡稳定时间选择寄存器	备用	962
P0	端口 0 寄存器	端口	104
P1	端口 1 寄存器	端口	107

(7/12)

符号	名称	单元	页码
P3	端口 3 寄存器	端口	109
P3H	端口 3 寄存器 H	端口	109
P3L	端口 3 寄存器 L	端口	109
P4	端口 4 寄存器	端口	115
P5	端口 5 寄存器	端口	118
P7H	端口 7 寄存器 H	端口	122
P7L	端口 7 寄存器 L	端口	122
P9	端口 9 寄存器	端口	125
P9H	端口 9 寄存器 H	端口	125
P9L	端口 9 寄存器 L	端口	125
PAR	IEBus 对方地址寄存器	IEBus	710
PC	程序计数器	CPU	57
PCC	处理器时钟控制寄存器	CG	214
PCM	端口 CM 寄存器	端口	132
PCT	端口 CT 寄存器	端口	134
PDH	端口 DH 寄存器	端口	136
PDL	端口 DL 寄存器	端口	139
PDLH	端口 DL 寄存器 H	端口	139
PDLL	端口 DL 寄存器 L	端口	139
PEMU1	外围扩展寄存器 1	CPU	1005
PF0	端口 0 功能寄存器	端口	106
PF3	端口 3 功能寄存器	端口	114
PF3H	端口 3 功能寄存器 H	端口	114
PF3L	端口 3 功能寄存器 L	端口	114
PF4	端口 4 功能寄存器	端口	117
PF5	端口 5 功能寄存器	端口	121
PF9	端口 9 功能寄存器	端口	131
PF9H	端口 9 功能寄存器 H	端口	131
PF9L	端口 9 功能寄存器 L	端口	131
PFC0	端口 0 功能控制寄存器	端口	106
PFC3	端口 3 功能控制寄存器	端口	112
PFC3H	端口 3 功能控制寄存器 H	端口	112
PFC3L	端口 3 功能控制寄存器 L	端口	112
PFC4	端口 4 功能控制寄存器	端口	116
PFC5	端口 5 功能控制寄存器	端口	120
PFC9	端口 9 功能控制寄存器	端口	128
PFC9H	端口 9 功能控制寄存器 H	端口	128
PFC9L	端口 9 功能控制寄存器 L	端口	128
PFCE3L	端口 3 功能控制扩展寄存器 L	端口	112
PFCE5	端口 5 功能控制扩展寄存器	端口	120
PFCE9	端口 9 功能控制扩展寄存器	端口	128
PFCE9H	端口 9 功能控制扩展寄存器 H	端口	128
端口	端口 9 功能控制扩展寄存器 L	端口	128
PIC0	中断控制寄存器	INTC	933
PIC1	中断控制寄存器	INTC	933

(8/12)

符号	名称	单元	页码
PIC2	中断控制寄存器	INTC	933
PIC3	中断控制寄存器	INTC	933
PIC4	中断控制寄存器	INTC	933
PIC5	中断控制寄存器	INTC	933
PIC6	中断控制寄存器	INTC	933
PIC7	中断控制寄存器	INTC	933
PLLCTL	PLL 控制寄存器	CG	220
PLLS	PLL 锁止时间指定寄存器	CG	223
PM0	端口 0 模式寄存器	端口	105
PM1	端口 1 模式寄存器	端口	107
PM3	端口 3 模式寄存器	端口	109
PM3H	端口 3 模式寄存器 H	端口	109
PM3L	端口 3 模式寄存器 L	端口	109
PM4	端口 4 模式寄存器	端口	115
PM5	端口 5 模式寄存器	端口	119
PM7H	端口 7 模式寄存器 H	端口	123
PM7L	端口 7 模式寄存器 L	端口	123
PM9	端口 9 模式寄存器	端口	125
PM9H	端口 9 模式寄存器 H	端口	125
PM9L	端口 9 模式寄存器 L	端口	125
PMC0	端口 0 模式控制寄存器	端口	105
PMC3	端口 3 模式控制寄存器	端口	110
PMC3H	端口 3 模式控制寄存器 H	端口	110
PMC3L	PMC9L 端口 3 模式控制寄存器 L	端口	110
PMC4	端口 4 模式控制寄存器	端口	116
PMC5	端口 5 模式控制寄存器	端口	119
PMC9	端口 9 模式控制寄存器	端口	126
PMC9H	端口 9 模式控制寄存器 H	端口	126
端口	PMC9L 端口 9 模式控制寄存器 L	端口	126
PMCCM	端口 CM 模式控制寄存器	端口	133
PMCCT	端口 CT 模式控制寄存器	端口	135
PMCDH	端口 DH 模式控制寄存器	端口	137
PMCDL	端口 DL 模式控制寄存器	端口	140
PMCDLH	端口 DL 模式控制寄存器 H	端口	140
PMCDLL	端口 DL 模式控制寄存器 L	端口	140
PMCM	端口 CM 模式寄存器	端口	132
PMCT	端口 CT 模式寄存器	端口	134
PMDH	端口 DH 模式寄存器	端口	136
PMDL	端口 DL 模式寄存器	端口	139
PMDLH	端口 DL 模式寄存器 H	端口	139
PMDLL	端口 DL 模式寄存器 L	端口	139
PRCMD	命令寄存器	CPU	91
PRSCM0	预分频比较寄存器 0	WT	444
PRSCM1	BRG1 预分频比较寄存器	BRG	591
PRSCM2	BRG2 预分频比较寄存器	BRG	591

符号	名称	单元	页码
PRSCM3	BRG3 预分频比较寄存器	BRG	591
PRSM0	预分频 模式寄存器 0	WT	443
PRSM1	BRG1 预分频 模式寄存器	BRG	590
PRSM2	BRG2 预分频 模式寄存器	BRG	590
PRSM3	BRG3 预分频 模式寄存器	BRG	590
PSC	节能控制寄存器	CG	960
PSMR	节能模式寄存器	CG	961
PSR	IEBus 节能寄存器	IEBus	697
PSW	程序状态字	CPU	61
r0 到 r31	通用寄存器 s	CPU	57
RAMS	内部 RAM 数据状态寄存器	CG	993
RCM	内部振荡模式寄存器	CG	218
RECIC0	中断控制寄存器	INTC	933
RESF	复位源标志寄存器	复位	982
RSA	IEBus 接收从地址寄存器	IEBus	711
RTBH0	实时输出缓冲寄存器 0H	RTP	457
RTBL0	实时输出缓冲寄存器 0L	RTP	457
RTPC0	实时输出 端口控制寄存器 0	RTP	459
RTPM0	实时输出 端口模式寄存器 0	RTP	458
SAR	IEBus 从地址寄存器	IEBus	710
SCR	IEBus 正确计数寄存器	IEBus	719
SELCNT0	选择器操作控制寄存器 0	Timer	322
SSR	IEBus 从状态寄存器	IEBus	698
STAIC	中断控制寄存器	INTC	933
SVA0	从地址寄存器 0	I ² C	616
SVA1	从地址寄存器 1	I ² C	616
SVA2	从地址寄存器 2	I ² C	616
SYS	系统状态寄存器 0	CPU	92
TM0CMP0	TMM0 比较寄存器 0	Timer	433
TM0CTL0	TMM0 控制寄存器 0	Timer	434
TM0EQIC0	中断控制寄存器	INTC	933
TP0CCIC0	中断控制寄存器	INTC	933
TP0CCIC1	中断控制寄存器	INTC	933
TP0CCR0	TMP0 捕捉/比较寄存器 0	Timer	235
TP0CCR1	TMP0 捕捉/比较寄存器 1	Timer	237
TP0CNT	TMP0 计数读缓冲寄存器	Timer	239
TP0CTL0	TMP0 控制寄存器 0	Timer	228
TP0CTL1	TMP0 控制寄存器 1	Timer	228
TP0IOC0	TMP0 I/O 控制寄存器 0	Timer	230
TP0IOC1	TMP0 I/O 控制寄存器 1	Timer	232
TP0IOC2	TMP0 I/O 控制寄存器 2	Timer	233
TP0OPT0	TMP0 选项寄存器 0	Timer	234
TP0OVIC	中断控制寄存器	INTC	933
TP1CCIC0	中断控制寄存器	INTC	933
TP1CCIC1	中断控制寄存器	INTC	933

(10/12)

符号	名称	单元	页码
TP1CCR0	TMP1 捕捉/比较寄存器 0	Timer	235
TP1CCR1	TMP1 捕捉/比较寄存器 1	Timer	237
TP1CNT	TMP1 计数读缓冲寄存器	Timer	239
TP1CTL0	TMP1 控制寄存器 0	Timer	228
TP1CTL1	TMP1 控制寄存器 1	Timer	228
TP1IOC0	TMP1 I/O 控制寄存器 0	Timer	230
TP1IOC1	TMP1 I/O 控制寄存器 1	Timer	232
TP1IOC2	TMP1 I/O 控制寄存器 2	Timer	233
TP1OPT0	TMP1 选项寄存器 0	Timer	234
TP1OVIC	中断控制寄存器	INTC	933
TP2CCIC0	中断控制寄存器	INTC	933
TP2CCIC1	中断控制寄存器	INTC	933
TP2CCR0	TMP2 捕捉/比较寄存器 0	Timer	235
TP2CCR1	TMP2 捕捉/比较寄存器 1	Timer	237
TP2CNT	TMP2 计数读缓冲寄存器	Timer	239
TP2CTL0	TMP2 控制寄存器 0	Timer	228
TP2CTL1	TMP2 控制寄存器 1	Timer	228
TP2IOC0	TMP2 I/O 控制寄存器 0	Timer	230
TP2IOC1	TMP2 I/O 控制寄存器 1	Timer	232
TP2IOC2	TMP2 I/O 控制寄存器 2	Timer	233
TP2OPT0	TMP2 选项寄存器 0	Timer	234
TP2OVIC	中断控制寄存器	INTC	933
TP3CCIC0	中断控制寄存器	INTC	933
TP3CCIC1	中断控制寄存器	INTC	933
TP3CCR0	TMP3 捕捉/比较寄存器 0	Timer	235
TP3CCR1	TMP3 捕捉/比较寄存器 1	Timer	237
TP3CNT	TMP3 计数读缓冲寄存器	Timer	239
TP3CTL0	TMP3 控制寄存器 0	Timer	228
TP3CTL1	TMP3 控制寄存器 1	Timer	228
TP3IOC0	TMP3 I/O 控制寄存器 0	Timer	230
TP3IOC1	TMP3 I/O 控制寄存器 1	Timer	232
TP3IOC2	TMP3 I/O 控制寄存器 2	Timer	233
TP3OPT0	TMP3 选项寄存器 0	Timer	234
TP3OVIC	中断控制寄存器	INTC	933
TP4CCIC0	中断控制寄存器	INTC	933
TP4CCIC1	中断控制寄存器	INTC	933
TP4CCR0	TMP4 捕捉/比较寄存器 0	Timer	235
TP4CCR1	TMP4 捕捉/比较寄存器 1	Timer	237
TP4CNT	TMP4 计数读缓冲寄存器	Timer	239
TP4CTL0	TMP4 控制寄存器 0	Timer	228
TP4CTL1	TMP4 控制寄存器 1	Timer	228
TP4IOC0	TMP4 I/O 控制寄存器 0	Timer	230
TP4IOC1	TMP4 I/O 控制寄存器 1	Timer	232
TP4IOC2	TMP4 I/O 控制寄存器 2	Timer	233
TP4OPT0	TMP4 选项寄存器 0	Timer	234

符号	名称	单元	页码
TP4OVIC	中断控制寄存器	INTC	933
TP5CCIC0	中断控制寄存器	INTC	933
TP5CCIC1	中断控制寄存器	INTC	933
TP5CCR0	TMP5 捕捉/比较寄存器 0	Timer	235
TP5CCR1	TMP5 捕捉/比较寄存器 1	Timer	237
TP5CNT	TMP5 计数读缓冲寄存器	Timer	239
TP5CTL0	TMP5 控制寄存器 0	Timer	228
TP5CTL1	TMP5 控制寄存器 1	Timer	228
TP5IOC0	TMP5 I/O 控制寄存器 0	Timer	230
TP5IOC1	TMP5 I/O 控制寄存器 1	Timer	232
TP5IOC2	TMP5 I/O 控制寄存器 2	Timer	233
TP5OPT0	TMP5 选项寄存器 0	Timer	234
TP5OVIC	中断控制寄存器	INTC	933
TQ0CCIC0	中断控制寄存器	INTC	933
TQ0CCIC1	中断控制寄存器	INTC	933
TQ0CCIC2	中断控制寄存器	INTC	933
TQ0CCIC3	中断控制寄存器	INTC	933
TQ0CCR0	TMQ0 捕捉/比较寄存器 0	Timer	334
TQ0CCR1	TMQ0 捕捉/比较寄存器 1	Timer	336
TQ0CCR2	TMQ0 捕捉/比较寄存器 2	Timer	338
TQ0CCR3	TMQ0 捕捉/比较寄存器 3	Timer	340
TQ0CNT	TMQ0 计数读缓冲寄存器	Timer	342
TQ0CTL0	TMQ0 控制寄存器 0	Timer	328
TQ0CTL1	TMQ0 控制寄存器 1	Timer	329
TQ0IOC0	TMQ0 I/O 控制寄存器 0	Timer	330
TQ0IOC1	TMQ0 I/O 控制寄存器 1	Timer	331
TQ0IOC2	TMQ0 I/O 控制寄存器 2	Timer	332
TQ0OPT0	TMQ0 选项寄存器 0	Timer	333
TQ0OVIC	中断控制寄存器	INTC	933
TRXIC0	中断控制寄存器	INTC	933
UA0CTL0	UARTA0 控制寄存器 0	UARTA	510
UA0CTL1	UARTA0 控制寄存器 1	UARTA	512, 534
UA0CTL2	UARTA0 控制寄存器 2	UARTA	512, 535
UA0OPT0	UARTA0 选项控制寄存器 0	UARTA	512
UA0RIC	中断控制寄存器	INTC	933
UA0RX	UARTA0 接收数据寄存器	UARTA	516
UA0STR	UARTA0 状态寄存器	UARTA	514
UA0TIC	中断控制寄存器	INTC	933
UA0TX	UARTA0 发送寄存器	UARTA	516
UA1CTL0	UARTA1 控制寄存器 0	UARTA	510
UA1CTL1	UARTA1 控制寄存器 1	UARTA	512, 534
UA1CTL2	UARTA1 控制寄存器 2	UARTA	512, 535
UA1OPT0	UARTA1 选项控制寄存器 0	UARTA	512
UA1RIC	中断控制寄存器	INTC	933
UA1RX	UARTA1 接收数据寄存器	UARTA	516

(12/12)

符号	名称	单元	页码
UA1STR	UARTA1 状态寄存器	UARTA	514
UA1TIC	中断控制寄存器	INTC	933
UA1TX	UARTA1 发送寄存器	UARTA	516
UA2CTL0	UARTA2 控制寄存器 0	UARTA	510
UA2CTL1	UARTA2 控制寄存器 1	UARTA	512, 534
UA2CTL2	UARTA2 控制寄存器 2	UARTA	512, 535
UA2OPT0	UARTA2 选项控制寄存器 0	UARTA	512
UA2RIC	中断控制寄存器	INTC	933
UA2RX	UARTA2 接收数据寄存器	UARTA	516
UA2STR	UARTA2 状态寄存器	UARTA	514
UA2TIC	中断控制寄存器	INTC	933
UA2TX	UARTA2 发送寄存器	UARTA	516
UAR	IEBus 单元地址寄存器	IEBus	710
USR	IEBus 单元状态寄存器	IEBus	699
VSWC	系统等待控制寄存器	CPU	93
WDTE	看门狗定时器允许寄存器	WDT	454
WDTM2	看门狗定时器模式寄存器 2	WDT	452, 939
WTIC	中断控制寄存器	INTC	933
WTIIC	中断控制寄存器	INTC	933
WTM	钟表定时器操作模式寄存器	WT	445
WUPIC0	中断控制寄存器	INTC	933

附录 D 指令集列表

D.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
向量	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
zero-extend (n)	用 0 来扩展 n 个位直到一个字边界
sign-extend (n)	用符号来扩展 n 个位直到一个字边界。
load-memory (a, b)	从地址 a 读取位 b.
store-memory (a, b, c)	写长度为 c 的数据 b 到地址 a..
load-memory 位 (a, b)	读取地址 a 的 b 位
store-memory 位 (a, b, c)	写入 c 到 地址 a 的 b 位.
saturated (n)	执行标准的 n 处理 (n 是 2 的补码)。 如果 $n \geq 7FFFFFFFH$, 结果为 $7FFFFFFFH$ 。如果 $n \leq 80000000H$, 结果为 $80000000H$ 。
result	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	在执行第一条指令后立即执行另一条指令 (issue) .
r	在执行第一条指令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

D.2 指令集（按字母顺序）

(1/6)

	操作数	操作码	操作	操作	执行时钟			标志				
					i	r	l	CY	OV	S	Z	SAT
	ADD	reg1, reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]	1	1	1	x	x	x	x	
		imm5, reg2	rrrrr010010iiii	GR[reg2]←GR[reg2]+ 符号扩展 (imm5)	1	1	1	x	x	x	x	
	ADDI	imm16, reg1, reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+符号扩展 (imm16)	1	1	1	x	x	x	x	
	AND	reg1, reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]	1	1	1		0	x	x	
	ANDI	imm16, reg1, reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND 零扩展 (imm16)	1	1	1		0	x	x	
<R>	Bcond	disp9	dddd1011ddcccc 注 1	如果条件满足 then PC←PC+符号扩展 (disp9)	2 注2 注22	2 注2 注22	2 注2 注22					
				当条件满足时	1	1	1					
	BSH	reg2, reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) GR[reg2] (31 : 24) GR[reg2] (7 : 0) GR[reg2] (15 : 8)	1	1	1	x	0	x	x	
	BSW	reg2, reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) GR[reg2] (15 : 8) GR [reg2] (23 : 16) GR[reg2] (31 : 24)	1	1	1	x	0	x	x	
<R>	CALLT	imm6	0000001000iiii	CTPC←PC+2 (return PC) CTPSW←PSW adr←CTBP+零扩展 (imm6 逻辑左移 1) PC←CTBP+零扩展 (装载寄存器 (adr, Halfword))	4 注22	4 注22	4 注22					
	CLR1	bit#3, disp16[reg1]	10bbb111110RRRRR dddddddddddddd	adr←GR[reg1]+符号扩展 (disp16) Z flag←Not (装载寄存器 (adr, bit#3)) 存储存储器位 (adr, bit#3, 0)	3 注3	3 注3	3 注3				x	
		reg2, [reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not (Load-memory 位 (adr, reg2)) 存储存储器位 (adr, reg2, 0)	3 注3	3 注3	3 注3				x	
	CMOV	cccc, imm5, reg2, reg3	rrrrr111111iiii wwwww011000cccc0	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	cccc, reg1, reg2, reg3 rrrrr111111RRRR wwwww011001cccc0	如果条件满足 则 GR[reg3]←GR[reg1] 否则 GR[reg3]←GR[reg2]	1	1	1					
	CMP	reg1, reg2	rrrrr001111RRRRR	结果←GR[reg2]-GR[reg1]	1	1	1	x	x	x	x	
		imm5, reg2	rrrrr010011iiii	结果←GR[reg2]-符号扩展 (imm5)	1	1	1	x	x	x	x	
<R>	CTRET		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW	3 注2	3 注2	3 注2	R	R	R	R	R
<R>	DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW	3 注2	3 注2	3 注2	R	R	R	R	R

操作数	操作码	操作	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
<R>	DBTRAP		1111100001000000 DBPC←PC+2 (restored PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3 注2	3 注2	3 注2					
	DI		000001111100000 0000000101100000 PSW.ID←1	1	1	1					
	DISPOSE	imm5, list12	0000011001iiiiL LLLLLLLLLLLL000000 sp←sp+零扩展 (imm5 逻辑左移 2) GR[reg in list12]←装载寄存器 (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]	0000011001iiiiL LLLLLLLLLLLLRRRRR 注 5 sp←sp+零扩展 (imm5 逻辑左移 2) GR[reg in list12]←装载寄存器 (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	rrrrr11111RRRRR wwwww01011000000 GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
	DIVH	reg1, reg2	rrrrr000010RRRRR GR[reg2]←GR[reg2]÷GR[reg1] ^{注 6}	35	35	35		×	×	×	
		reg1, reg2, reg3	rrrrr11111RRRRR wwwww01010000000 GR[reg2]←GR[reg2]÷GR[reg1] ^{注 6} GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
	DIVHU	reg1, reg2, reg3	rrrrr11111RRRRR wwwww01010000010 GR[reg2]←GR[reg2]÷GR[reg1] ^{注 6} GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
	DIVU	reg1, reg2, reg3	rrrrr11111RRRRR wwwww01011000010 GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
	EI		100001111100000 0000000101100000 PSW.ID←0	1	1	1					
	HALT		000001111100000 0000000100100000 Stop	1	1	1					
	HSW	reg2, reg3	rrrrr1111100000 wwwww01101000100 GR[reg3]←GR[reg2] (15 : 0) GR[reg2] (31 : 16)	1	1	1	×	0	×	×	
<R>	JARL	disp22, reg2	rrrrr11110dddddd dddddddddddddd0 注 7 GR[reg2]←PC+4 PC←PC+符号扩展 (disp22)	2 注22	2 注22	2 注22					
<R>	JMP	[reg1]	00000000011RRRRR PC←GR[reg1]	3 注22	3 注22	3 注22					
<R>	JR	disp22	0000011110dddddd dddddddddddddd0 注 7 PC←PC+符号扩展 (disp22)	2 注22	2 注22	2 注22					
	LD.B	disp16[reg1], reg2	rrrrr111000RRRRR ddddddddddddddddd adr←GR[reg1]+符号扩展 (disp16) GR[reg2]←符号扩展 (装载寄存器 (adr, Byte))	1	1	注 11					
	LD.BU	disp16[reg1], reg2	rrrrr11110bRRRRR ddddddddddddddd1 注 8, 10 adr←GR[reg1]+符号扩展 (disp16) GR[reg2]←零扩展 (装载寄存器 (adr, Byte))	1	1	注 11					

操作数	操作码	操作	操作		执行时钟			标志				
					i	r	l	CY	OV	S	Z	SAT
LD.H	disp16[reg1], reg2	rrrrr111001RRRRR ddddddddddddddd0 注 8	adr←GR[reg1]+符号扩展（disp16） GR[reg2]←符号扩展（装载寄存器（adr，Halfword））		1	1	注 11					
LDSR	reg2, regID	rrrrr111111RRRRR 0000000000100000 注 12	SR[regID]←GR[reg2]	不同于 regID = PSW	1	1	1					
				regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1], reg2	rrrrr111111RRRRR ddddddddddddddd1 注 8	adr←GR[reg1]+符号扩展（disp16） GR[reg2]←零扩展（装载寄存器（adr，Halfword））		1	1	注 11					
LD.W	disp16[reg1], reg2	rrrrr111001RRRRR ddddddddddddddd1 注 8	adr←GR[reg1]+符号扩展（disp16） GR[reg2]←装载寄存器（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 iiiiiiiiiiiiiiii iiiiiiiiiiiiiiii	GR[reg1]←imm32		2	2	2					
MOVEA	imm16, reg1, reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+符号扩展（imm16）		1	1	1					
MOVHI	imm16, reg1, reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+（imm16∥0 ¹⁶ ）		1	1	1					
MUL	reg1, reg2, reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3]∥GR[reg2]←GR[reg2]xGR[reg1] 注 14		1	4	5					
	imm9, reg2, reg3	rrrrr111111iiii wwwww01001111100 注 13	GR[reg3]∥GR[reg2]←GR[reg2]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	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] ⁶ ximm16		1	1	2					
MULU	reg1, reg2, reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3]∥GR[reg2]←GR[reg2]xGR[reg1] 注 14		1	4	5					
	imm9, reg2, reg3	rrrrr111111iiii wwwww0100111110 注 13	GR[reg3]∥GR[reg2]←GR[reg2]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 ddddddddddddddd	adr←GR[reg1]+符号扩展（disp16） Z flag←Not（装载存储器位（adr，bit#3）） 存储存储器位（adr，bit#3，Z flag）		3 注 3	3 注 3	3 注 3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not（装载存储器位（adr，reg2）） 存储存储器位（adr，reg2，Z flag）		3 注 3	3 注 3	3 注 3				×	

(4/6)

操作数	操作码	操作	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
OR	reg1, reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16, reg1, reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR 零扩展 (imm16)	1	1	1		0	×	×	
PREPARE	list12, imm5	0000011110iiiiL LLLLLLLLLLLL00001	Store-memory (sp-4, GR[reg in list12], 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}	0000011110iiiiL LLLLLLLLLLLLff011 imm16/imm32 注16	Store-memory (sp-4, GR[reg in list12], 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		000001111100000 0000000101000000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	3 注22	3 注22	3 注22	R	R	R	R	R
SAR	reg1, reg2	rrrrr11111RRRRR 0000000010100000	GR[reg2]←GR[reg2]算数右移 GR[reg1]	1	1	1	×	0	×	×	
	imm5, reg2	rrrrr010101iiii	GR[reg2]←GR[reg2]算数右移 零扩展 (imm5)	1	1	1	×	0	×	×	
SASF	cccc, reg2	rrrrr111110cccc 0000001000000000	如果条件满足 则 GR[reg2]← (GR[reg2] 逻辑左移 1) 或者 00000001H 否则 GR[reg2]← (GR[reg2] 逻辑右移 1) 或者 00000000H	1	1	1					
SATADD	reg1, reg2	rrrrr000110RRRRR	GR[reg2]← 饱和运算 (GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5, reg2	rrrrr010001iiii	GR[reg2]← 饱和运算 (GR[reg2]+符号扩展 (imm5))	1	1	1	×	×	×	×	×
SATSUB	reg1, reg2	rrrrr000101RRRRR	GR[reg2]← 饱和运算 (GR[reg2]-GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16, reg1, reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR[reg2]← 饱和运算 (GR[reg1]-符号扩展 (imm16))	1	1	1	×	×	×	×	×
SATSUBR	reg1, reg2	rrrrr000100RRRRR	GR[reg2]← 饱和运算 (GR[reg1]-GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc, reg2	rrrrr111110cccc 0000000000000000	如果条件满足 则 GR[reg2]←00000001H 否则 GR[reg2]←00000000H 助记符	1	1	1					

<R>

操作数	操作码	操作	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
SET1	bit#3, disp16[reg1]	00bbb11110RRRRR dddddddddddddd	adr←GR[reg1]+符号扩展 (disp16) Z flag←Not (装载寄存器位 (adr, bit#3)) 存储存储位 (adr, bit#3, 1)	3 注3	3 注3	3 注3				×	
	reg2, [reg1]	rrrrr11111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not (装载寄存器位 (adr, reg2)) 存储存储位 (adr, reg2, 1)	3 注3	3 注3	3 注3				×	
SHL	reg1, reg2	rrrrr11111RRRRR 0000000011000000	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	rrrrr11111RRRRR 0000000010000000	GR[reg2]←GR[reg2] 逻辑右移 GR[reg1]	1	1	1	×	0	×	×	
	imm5, reg2	rrrrr010100iiii	GR[reg2]←GR[reg2] 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 (装载寄存器 (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 (装载寄存器 (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 (装载寄存器 (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 (装载寄存器 (adr, Halfword))	1	1	注9					
SLD.W	disp8[ep], reg2	rrrrr1010dddddd 注 21	adr←ep+零扩展 (disp8) GR[reg2]←装载寄存器 (adr, Word)	1	1	注9					
SST.B	reg2, disp7[ep]	rrrrr0111dddddd	adr←ep+零扩展 (disp7) 装载寄存器 (adr, GR[reg2], Byte)	1	1	1					
SST.H	reg2, disp8[ep]	rrrrr1001dddddd 注 19	adr←ep+零扩展 (disp8) 装载寄存器 (adr, GR[reg2], Halfword)	1	1	1					
SST.W	reg2, disp8[ep]	rrrrr1010dddddd 注 21	adr←ep+零扩展 (disp8) 装载寄存器 (adr, GR[reg2], Word)	1	1	1					
ST.B	reg2, disp16[reg1]	rrrrr111010RRRRR dddddddddddddd	adr←GR[reg1]+符号扩展 (disp16) 装载寄存器 (adr, GR[reg2], Byte)	1	1	1					
ST.H	reg2, disp16[reg1]	rrrrr111011RRRRR dddddddddddddd 注 8	adr←GR[reg1]+符号扩展 (disp16) 装载寄存器 (adr, GR[reg2], Halfword)	1	1	1					
ST.W	reg2, disp16[reg1]	rrrrr111011RRRRR dddddddddddddd 注 8	adr←GR[reg1]+符号扩展 (disp16) 装载寄存器 (adr, GR[reg2], Word)	1	1	1					
STSR	regID, reg2	rrrrr11111RRRRR 000000001000000	GR[reg2]←SR[regID] 助记符	1	1	1					

操作数	操作码	操作	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1, reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]-GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1, reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]-GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	0000000010RRRRR	adr←(PC+2)+(GR[reg1]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	向量	000001111111iiii 0000000100000000	EIPC ←PC+4 (恢复的 PC) EIPSW ←PSW ECR.EICC ←中断码 PSW.EP ←1 PSW.ID ←1 PC ←00000040H (当向量是 00H 到 0FH) 00000050H (当向量是 10H 到 1FH)	3 注22	3 注22	3 注22					
TST	reg1, reg2	rrrrr001011RRRRR	result←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]	rrrrr111111RRRRR 0000000011100110	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	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16, reg1, reg2	rrrrr110101RRRRR iiiiiiiiiiiiiiii	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. 如果有重写刚才的 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 加载寄存器的总个数。(按照等待状态数目。如果没有等待状态, n 是 list12 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，为了方便指令描述，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: 除 00000 外

19. ddddddd: disp8 的高 7 位。

20. dddd: disp5 的高 4 位。

21. ddddd: disp8 的高 6 位。

<R>

22. 对于 V850ES/SG2-H, 时钟+1。

附录 E 再版修订记录

E.1 E.1 在此版本中的主要修订

(1/10)

页码	修订说明
所有页面	• 所有的产品都被改为无铅产品。 • 增加了 V850ES/SG2-H
第 26、27 页	修改了 1.2 特征中的描述
第 31、33 页	增加了 1.5 管脚配置（顶层）说明
第 35 页	增加了 1.6.1 内部块图表的注释说明
第 39 页	增加了 2.1（1）端口管脚的注释说明
第 43 页	增加了 2.1（2）无端口管脚的注释说明
第 49 页	修改了表 2-2 各种状态下管脚状态的描述
第 50、51 页	修改了 2.3 管脚 I/O 电路类型，I/O 缓冲器电源提供，无用管脚连接中的描述
第 53 页	增加了图 2-1 管脚 I/O 电路的描述
第 54 页	修改了 2.4（1）上电注意事项中的描述
第 54 页	增加了 2.4（2）FLMD0 管脚注意事项
第 55 页	增加了 3.1 特征的描述
第 58 页	增加了表 3-2 系统寄存器个数的注释
第 63 页	增加了 3.2.2（6）异常/调试捕捉状态保存寄存器（DBPC 和 DBPSW）的描述
第 64 页	修改了 3.3.1 指定运行模式中的描述
第 73 页	修改了 3.4.4（3）片上周边 I/O 区的注意事项 4
第 74 页	修改了 3.4.4（5）外部存储区的注意事项
第 79、86 页	增加了 3.4.6 外围 I/O 寄存器的注释
第 89 页	修改和添加了 3.4.8 特殊寄存器的注释
第 92 页	修改了 3.4.8（3）（a）设置条件（PRERR flag = 1）中的描述
第 93 页	添加了 3.4.9（1）（a）系统等待控制寄存器（VSWC）的描述
第 94 页	修改了 3.4.9（2）访问指定片上周边 I/O 寄存器
第 106 页	增加了 4.3.1（5）端口 0 功能寄存器（PF0）的注意事项
第 107 页	修改了表 4-5 端口 1 功能复用管脚的注意事项
第 107 页	增加了 4.3.2（1）端口 1 寄存器（P1）的注意事项
第 108 页	修改了表 4-6 端口 3 功能复用管脚的注意事项
第 114 页	增加了 4.3.3（7）端口 3 功能寄存器（PF3）的注意事项
第 117 页	增加了 4.3.4（5）端口 4 功能寄存器（PF4）的注意事项
第 118 页	修改了表 4-8 端口 5 功能复用管脚的注意事项
第 121 页	增加了 4.3.5（7）端口 5 功能寄存器（PF5）的注意事项
第 122 页	修改了 4.3.6（1）端口 7 寄存器 H，端口 7 寄存器 L（P7H，P7L）中的描述
第 124 页	修改了表 4-10 端口 9 功能复用管脚的注意事项
第 127 页	修改了 4.3.7（3）端口 9 模式控制寄存器（PMC9）的注意事项
第 128 页	修改了 4.3.7（4）端口 9 功能控制寄存器（PFC9）的注意事项
第 131 页	增加了 4.3.7（7）端口 9 功能寄存器（PF9）的注意事项

(2/10)

页码	修订说明
第 146 页到 149 页、151 页、153 页、156 页、158 到 170 页	修改了图 4-6, 图 4-8 到 图 4-11, 图 4-13 到 图 4-15, 图 4-18, 图 4-20 到 图 4-32
第 175 到 177 页	修改了表 4-15 用端口管脚作为功能复用管脚的注释
第 183 页	增加了 4.6.3 片上调试管脚注意事项中的注意事项
第 183 页	修改了 4.6.5 上电时 P10, P11, 和 P53 管脚注意事项中的描述
第 183 页	修改了 4.6.6 磁滞现象的描述
第 183 页	增加了 4.6.7 分离总线模式注意事项
第 184 页	增加了 5.1 特征的描述
第 188 页	增加了 5.5.1 (2) 对于 V850ES/SG2-H
第 197 页	修改了 5.6.1 (1) 数据等待控制寄存器 0 (DWC0) 中的描述
第 200 页	修改了 5.6.4 (1) 地址等待控制寄存器 (AWC) 中的描述
第 211 页	增加了 6.1 概述的描述
第 213 页	增加了 6.2 (1) 主时钟振荡器的描述
第 218 页	增加了 6.3 (3) CPU 工作时钟状态寄存器 (CCLS) 的注释
第 220 页	增加了 6.5.1 概述的描述
第 220 页	增加了 6.5.2 (1) PLL 控制寄存器 (PLLCTL) 的注意事项
第 221 页	增加了 6.5.2 (2) 时钟控制寄存器 (CKC) 的描述
第 223 页	修改了 6.5.3 (1) (b) PLL 运行模式下设置 IDLE2/STOP 模式
第 228 页	修改了 7.4 (1) TMPn 控制寄存器 0 (TPnCTL0) 的注释
第 229 页	增加了 7.4 (2) TMPn 控制寄存器 1 (TPnCTL1) 中的描述
第 231 页	修改了 7.4 (3) TMPn I/O 控制寄存器 0 (TPnIOC0) 的注释和注意事项
第 232 页	修改了 7.4 (4) TMPn I/O 控制寄存器 1 (TPnIOC1)
第 234 页	增加了 7.4 (6) TMPn 选项寄存器 0 (TPnOPT0) 中的描述
第 236 页	增加了 7.4 (7) (a) 比较寄存器功能的描述
第 236 页	增加了 7.4 (7) (b) 捕捉寄存器功能的描述
第 236 页	增加了 表 7-2 每个模式下捕捉/比较寄存器功能和如何写比较寄存器的注释
第 238 页	增加了 7.4 (8) (a) 比较寄存器功能的描述
第 238 页	增加了 7.4 (8) (b) 捕捉寄存器功能的描述
第 238 页	增加了 表 7-3 每个模式下的捕捉/比较寄存器功能, 以及如何写入比较寄存器的注释
第 242 页	增加了 7.6 (1) 计数器基本操作
第 243 页	增加了 7.6 (2) 任意时间写和批量写
第 248 页	修改了 7.6.1 间隔计时器模式 (TPnMD2 to 到 TPnMD0 bits = 000) 的描述
第 249 到 251 页	修改了图 7-8 间隔计时器模式寄存器设置的描述
第 252 页	修改了图 7-9. 间隔定时器模式下的软件处理流程的描述
第 253 页	修改了 7.6.1 (2) (a) 如果 TPnCCR0 寄存器被设置为 0000H 操作的描述与图
第 257、258 页	增加了 7.6.1 (2) (d) TPnCCR1 寄存器操作的描述
第 259 页	增加了 7.6.1 (3) 外部事件计数器输入 (TIPn0) 操作
第 260 到 263 页	修改了 7.6.2 外部事件计时器模式 (TPnMD2 到 TPnMD0 bits = 000) 的描述
第 261 页	修改了图 7-14 外部事件计数器模式基本时序的描述

(3/10)

页码	修订说明
第 263 页	修改了 图 7-15 外部事件计数模式操作的寄存器设置中描述
第 265 页	修改了 7.6.2 (2) (a) 如果 TPnCCR0 寄存器被设置为 0000H 操作的描述与图
第 266 页	修改了 7.6.2 (2) (b) 重写 TPnCCR0 寄存器注释的描述
第 268 页	增加了 7.6.2 (2) (c) TPnCCR1 寄存器操作的描述
第 269 页	增加了 图 7-20 配置外部触发脉冲输出模式的注释和注意事项
第 270、271 页	修改了 图 7-22 设置外部触发脉冲输出模式的描述
第 277 页	修改了 7.6.3 (2) (b) PWM 波形 0%/100%输出的描述
第 281 页	增加了 图 7-24 配置单触发脉冲输出模式的注释和注意事项
第 282 页	修改了 7.6.4 单触发脉冲输出模式 (TPnMD2 到 TPnMD0 位 = 011) 的描述
第 283、284 页	修改了 图 7-26 设置单触发脉冲输出模式的描述
第 288 页	修改了 图 7-28 配置 PWM 输出模式
第 295 页	修改了 7.6.3 (2) (b) PWM 波形 0%/100%输出的描述
第 297 页	增加了 图 7-32 自由运行计时器模式配置的注释
第 301 页	增加了 图 7-35 自由运行计时器模式配置的注释
第 314 页	增加了 7.6.6 (3) 捕捉模式注释
第 315 页	修改了 图 7-38 配置脉冲宽度测量模式的图与注意事项
第 317、318 页	修改了 图 7-40 设置脉冲宽度测量模式寄存器的描述
第 319 页	删除了 图 7-41 脉冲宽度测量模式软件处理流程的描述
第 320 页	增加了 7.6.7 (3) 注释
第 328 页	修改了 8.4 (1) TMQ0 控制寄存器 0 (TQ0CTL0) 的注释
第 329 页	增加了 8.4 (2) TMQ0 控制寄存器 1 (TQ0CTL1) 中的描述
第 330 页	在 8.4 (3) TMQ0 I/O 控制寄存器中 0 (TQ0IOC0) 中增加了注释与注意事项
第 333 页	增加了 8.4 (6) TMQ0 选项寄存器 0 (TQ0OPT0) 中的描述
第 335 页	增加了 8.4 (7) (a) 比较寄存器功能的描述
第 335 页	增加了 8.4 (7) (b) 捕捉寄存器功能的描述
第 335 页	增加了 表 8-2 每个模式下捕捉/比较寄存器功能和如何写比较寄存器的注释
第 337 页	增加了 8.4 (8) (a) 比较寄存器功能的描述
第 337 页	增加了 8.4 (8) (b) 捕捉寄存器功能的描述
第 337 页	增加了 表 8-3 每个模式下捕捉/比较寄存器功能和如何写比较寄存器的注释
第 339 页	增加了 8.4 (9) (a) 比较寄存器功能的描述
第 339 页	增加了 8.4 (9) (b) 捕捉寄存器功能的描述
第 339 页	增加了 表 8-4 每个模式下捕捉/比较寄存器功能和如何写比较寄存器的注释
第 341 页	增加了 8.4 (10) (a) 比较寄存器功能的描述
第 341 页	增加了 8.4 (10) (b) 捕捉寄存器功能的描述
第 341 页	增加了 表 8-5 每个模式下捕捉/比较寄存器功能和如何写比较寄存器的注释
第 344 页	增加了 8.6 (1) 计数器基本操作
第 346 页	增加了 8.6 (2) 任意时间写和批量写
第 351 页	修改了 8.6.1 间隔计时器模式 (TQ0MD2 到 TQ0MD0 位 = 000) 的描述

页码	修订说明
第 352 到 354 页	修改了图 8-8 间隔计时器模式寄存器设置的描述
第 355 页	修改了图 8-9. 间隔计时器模式下的软件处理流程的描述
第 356 页	修改了 8.6.1 (2) (a) 如果 TQ0CCR0 寄存器被设置为 0000H 操作的描述与图
第 359、360 页	增加了 8.6.1 (2) (d) TQ0CCR1 到 TQ0CCR3 寄存器操作的描述
第 361 页	增加了 8.6.1 (3) 外部事件计数器输入 (TIQ00) 操作
第 362、364 页	修改了 8.6.2 外部事件计时器模式 (TQ0MD2 到 TQ0MD0 bits = 000) 的描述
第 363 页	修改了图 8-14 外部事件计数器模式基本时序的描述
第 365 页	修改了图 8-15 外部事件计数模式操作的寄存器设置中描述
第 367 页	修改了 8.6.2 (2) (a) 如果 TQ0CCR0 寄存器被设置为 0000H 操作的描述与图
第 368 页	修改了 8.6.2 (2) (b) 重写 TQ0CCR0 寄存器注释的描述
第 369 页	增加了 8.6.2 (2) (c) TQ0CCR1 到 TQ0CCR3 寄存器操作的描述
第 372 页	修改了 8.6.3 外部触发脉冲输出模式 (TQ0MD2 到 TQ0MD0 bits = 010) 的描述
第 372 页	增加了图 8-20 配置外部触发脉冲输出模式的注释和注意事项
第 374 到 376 页	修改了图 8-22 设置外部触发脉冲输出模式的描述
第 381 页	修改了 8.6.3 (2) (b) PWM 波形 0%/100%输出的描述
第 385 页	增加了图 8-24 配置单触发脉冲输出模式的注释和注意事项
第 385 页	修改了 8.6.4 单触发脉冲输出模式 (TQ0MD2 到 TQ0MD0 位 = 011) 的描述
第 387 到 389 页	修改了图 8-26 设置单触发脉冲输出模式的描述
第 394 页	修改了图 8-28 配置 PWM 输出模式
第 403 页	修改了 8.6.5 (2) (b) PWM 波形 0%/100%输出的描述
第 406 页	增加了图 8-32 自由运行计时器模式配置的注释
第 410 页	增加了图 8-35 自由运行计时器模式配置的注释
第 425 页	增加了 8.6.6 (3) 捕捉模式注释
第 426 页	修改了图 8-38 配置脉冲宽度测量模式的图与注意事项
第 428、429 页	修改了图 8-40 设置脉冲宽度测量模式寄存器的描述
第 430 页	删除了图 8-41 脉冲宽度测量模式软件处理流程的描述
第 431 页	增加了 8.6.7 (3) 注释
第 435 页	修改了图 9-3 间隔计时器模式操作基本时序
第 438 页	修改了 9.4.1 (2) (a) 如果 TM0CMP0 寄存器被设置为 0000H 操作的描述与图
第 438 页	修改了 9.4.1 (2) (不) 如果 TM0CMP0 寄存器被设置为 N 操作的描述与图
第 439 页	增加了 9.4.2 (3) 不能设置 TM0CMP0 寄存器为 FFFFH
第 452 页	修改了和增加了 11.3 (1) 看门狗定时器模式寄存器 2 (WDTM2) 中的注意事项
第 453 页	修改了表 11-2 看门狗定时器时钟选择的描述
第 454 页	修改了 11.3 (2) 看门狗定时器模式寄存器 (WDTE) 中的注意事项
第 467 页	修改了和增加了 13.4 (1) A/D 转换模式寄存器 0 (ADA0M0) 的注释
第 468 页	修改了和增加了 13.4 (2) A/D 转换模式寄存器 1 (ADA0M0) 的注释
第 469 页	增加了表 13-2 正常转换模式下转换时间的选择 (ADA0HS1 位= 0)
第 470 页	修改了表 13-3 高速转换模式下转换时间的选择 (ADA0HS1 位= 1) 的描述
第 471 页	增加了 13.4 (3) A/D 转换模式寄存器 2 (ADA0M2) 的注意事项

页码	修订说明
第 472 页	增加了 13.4 (4) 模拟输入通道指定寄存器 (ADA0S) 的注意事项
第 475 页	增加了 13.4 (6) 电源失败比较模式寄存器 (ADA0PFM) 的注意事项
第 476 页	增加了 13.4 (7) 电源失败比较门限值寄存器 (ADA0PFTM) 的注意事项
第 477 页	增加了 13.5.1 基本操作的备注
第 478 页	修改了图 13-3 转换操作的时序 (连续转换) 的描述
第 479 页	增加了 13.5.3 (1) 软件触发模式的描述
第 479 页	增加了 13.5.3 (2) 外部触发模式的描述
第 480 页	增加了 13.5.3 (3) 计时器触发模式的描述
第 491 页	修改了 13.6 (4) 多功能 I/O 的描述
第 492 页	修改了图 13-14 ANIn 管脚内部等价电路的描述
第 493 页	修改了 13.6 (8) 读取寄存器 ADA0CRn 中的描述
第 494 页	修改了 13.6 (10) 各个模式约束的描述
第 507 页	增加了 15.2 特征的描述
第 508 页	修改了图 15-4 异步串行接口 An 的模块图的模块
第 511 页	修改了 15.4 (1) UARTAn 控制寄存器 0 (UAnCTL0) 的注释
第 512、513 页	增加了 15.4 (4) UARTAn 选项控制寄存器 0 (UAnOPT0) 中的描述
第 514 页	增加了 15.4 (5) UARTAn 状态寄存器 (P1) 的注意事项
第 516 页	增加了 15.4 (6) UARTAn 接收数据寄存器 (UAnRX) 的描述
第 516 页	增加了 15.4 (7) UARTAn 发送数据寄存器 (UAnTX) 的描述
第 519 页	修改了图 15-5UARTA 发送/接收数据格式的描述
第 523 页	增加了 15.6.4 SBF 接收的注意事项
第 527 页	修改了图 15-12 连续发送操作时序
第 533 页	增加了图 15-16 波特率产生器配置的注意事项
第 533 页	增加了 15.7 (1) (a) 基本时钟的描述
第 537 页	增加了表 15-3. 波特率发生器的数据设置
第 544 页	增加了 16.2 特征的描述
第 547、549 页	在 16.4 (1) CSIBn 控制寄存器 0 (CBnCTL0) 中增加了注释和注意事项
第 550 页	增加了 16.4 (1) (a) 如何使用 CBnSCE 位
第 551 页	增加了 16.4 (2) CSIBn 控制寄存器 1 (CBnCTL1) 中的注释
第 554 页	增加了 16.4 (4) CSIBn 状态寄存器 (CBnSTR) 的注意事项
第 555 页	增加了 16.5 中断请求信号
第 556 页	增加了 16.6.1 单传输模式 (主模式, 传输模式)
第 558 页	修改了 16.6.2 单传输模式 (主模式, 传输模式) 中的描述
第 560 页	修改了 16.6.3 单传输模式 (主模式, 传输/接收模式) 中的描述
第 562 页	增加了 16.6.4 单传输模式 (从模式, 传输模式)
第 564 页	增加了 16.6.5 单传输模式 (从模式, 接收模式)
第 566 页	增加了 16.6.6 单传输模式 (从模式, 传输/接收模式)
第 568 页	增加了 16.6.7 连续传输模式 (主模式, 传输模式)
第 570 页	修改了 16.6.9 连续传输模式 (主模式, 接收模式) 中的描述
第 573 页	修改了 16.6.9 连续传输模式 (主模式, 传输/接收模式) 中的描述

页码	修订说明
第 577 页	增加了 16.6.10 连续传输模式 （从模式，传输模式）
第 579 页	修改了 16.6.11 连续传输模式 （从模式，接收模式）中的描述
第 582 页	修改了 16.6.12 连续传输模式 （从模式，传输/接收模式）中的描述
第 586 页	修改了 16.6.13 接收错误 的描述
第 587、588 页	修改了 16.6.14 时钟时序 的描述
第 591 页	增加了 16.8 1 波特率计算 中的注意事项
第 603 页	删除了 17.4（1）IIC 控制寄存器 0 到 2（IICC0 到 IICC2） 的描述
第 612 页	增加了 17.4（6）I²C0n 传输时钟设置方法 的描述
第 613、614 页	增加了 表 17-2 时钟设置 的描述与注
第 624 页	修改了 17.6.6 等待状态 的描述
第 625 页	增加了 17.6.7 等待状态取消方法 的注
第 627 页	增加了 17.7.1（1）开始～地址～数据～数据～停止（正常传输/接收） 的注
第 628 页	增加了 17.7.1（2）开始～地址～数据～开始～地址～数据～停止（重起） 的注，删除了描述
第 629 页	增加了 17.7.1（3）开始～代码～数据～数据～停止（正常传输/接收） 的注
第 633 页	修改了 17.7.3（4）开始～地址～数据～开始～地址～数据～停止 的描述
第 634 页	增加了 17.7.3 从设备操作（当收到额外代码） 的描述
第 637 页	修改了 17.7.3（4）开始～代码～数据～开始～地址～数据～停止 的描述
第 639 页	增加了 17.7.5 仲裁失败操作（仲裁失败后从设备的操作） 的描述
第 641 页	增加了 17.7.6 仲裁失败时德操作（仲裁失败后无通信） 的描述
第 645 页	修改了 17.7.6（6）仲裁失败时为产生重启条件 SDA0n 拉低电平 的描述
第 647 页	修改了 17.7.6（8）仲裁失败时为产生重启条件 SDA0n 拉低电平 的描述
第 650 页	增加了 17.11 额外代码 的描述
第 653 页	修改了 17.14.1 当通信预定功能使能（IICFn.IICRSVn 位= 0） 的描述
第 654 页	增加了 表 17-6 等待周期 的注
第 655 页	修改了 图 17-15 通信预留时间
第 657 页	修改了 表 17-7 等待周期 的描述
第 659 页	修改了 17.16 通信操作 的描述
第 666 页	修改了 图 17-21 从模式流程图（1）
第 667 页	修改了 图 17-22 从模式流程图（2）
第 669 到 671 页	修改了 图 17-23 主从通信的例子（当主从都选用 9 个等待周期时）
第 672 到 674 页	修改了 图 17-24 从主通信的例子（当主从都选用 9 个等待周期时）
第 690 页	修改了 图 18-10 IEBus 控制器方框图 的描述
第 721 页	增加了 18.3（17）IEBus 时钟选择寄存器（OCSK2） 的描述
第 743 页	增加了 第 19 章 CAN 控制器 的注意事项
第 744 页	删除了 表 19-1 功能概述 的描述

(7/10)

页码	修订说明
第 759 页	修改了 19.3.6 (4) (b) 错误计数器的描述
第 760 页	修改了 19.3.6 (5) (a) 通过恢复顺序从 bus-off 状态恢复 中修改了注意事项
第 762 页	修改了图 19-18 段设置的描述
第 790 页	增加了 19.6 (1) CAN0 全局控制寄存器 (C0GMCTRL) 的注意事项
第 793 页	增加了 19.6 (3) CAN0 全局自动块传输控制寄存器 (C0GMABT) 的注意事项
第 797 到 799 页	修改了 19.6 (6) CAN0 模块控制寄存器 (C0CTRL) 的描述
第 806 页	增加了 19.6 (11) CAN0 模块中断状态寄存器 (C0INTS) 的注意事项
第 811 页	增加了 19.6 (15) CAN0 模块接收历史列表寄存器 (C0RGPT) 的注
第 813 页	增加了 19.6 (17) CAN0 模块发送历史列表寄存器 (C0TGPT) 的注
第 820 页	增加了 19.6 (22) CAN0 消息 ID 寄存器 m (C0MIDLm, C0MIDHm) 的注意事项
第 822、823 页	增加了 19.6 (23) CAN0 消息控制寄存器 m (C0MCTRLm) 的注意事项
第 830 页	增加了 19.9.2 读接收数据
第 831 页	增加了 19.9.3 接收历史列表功能的注意事项
第 838 页	增加了 19.10.1 消息传送的备注
第 839 页	增加了 19.9.3 发送历史列表功能的注意事项
第 845 页	增加了 19.11.1 (2) CAN 睡眠模式状态的描述
第 845 页	增加了 19.11.1 (3) 释放 CAN 休眠模式中的注意事项
第 846 页	在 19.11.2 (1) 进入 CAN 停止模式中修改了注意事项
第 846 页	增加了 19.11.1 (2) CAN 睡眠模式状态的描述
第 847 页	修改了 19.11.3 使用节能模式举例
第 851 页	增加了 19.13.3 自测模式的描述
第 852 页	增加了 19.13.4 每个操作模式传送/接收操作
第 855 页	增加了 19.15.1 比特率设置条件的描述
第 866 页	增加了图 19-39 消息缓冲器重定义的注
第 870 页	增加了图 19-43 通过中断传输 (使用 C0LOPT 寄存器) 的备注
第 871 页	增加了图 19-44 通过中断传输 (使用 C0TGPT 寄存器) 的备注
第 872 页	增加了图 19-45 通过软件表决的备注
第 873 页	增加了图 19-46 传输放弃处理 (除正常模式使用 ABT 之外) 中的注与注意事项
第 874 页	增加了图 19-47 传输放弃处理除去 ABT 传输 (使用 ABT 的正常模式) 中的注与注意事项
第 877 页	增加了图 19-49 通过中断接收 (使用 C0LIPT 寄存器) 的备注
第 878 页	修改了图并增加了图 19-50 通过中断接收 (使用 C0RGPT 寄存器) 的备注
第 879 页	修改了图并增加了图 19-51 通过软件表决的备注
第 880 页	修改了图并删除了图 19-52 设置 CAN 睡眠模式/停止模式的注意事项
第 881 页	修改了图 19-53. 清除 CAN 休眠/停止模式
第 882 页	增加了图 19-54 Bus-off 恢复 (除正常模式使用 ABT 之外) 中的注与注意事项
第 883 页	增加了图 19-54 Bus-off 恢复 (除正常模式使用 ABT 之外)
第 884 页	删除了图 19-56 正常关闭过程中的一部分
第 887 页	修改了图 19-59. 设定 CPU 的待机模式 (从 CAN 休眠模式)

页码	修订说明
第 888 页	修改了图 19-60. 设定 CPU 的备份（从 CAN 停止模式）
第 896、887 页	修改了 20.3（6）DMA 触发因数寄存器 0 到 3（DTFR0 到 DTFR3）中的注并增加了注意事项
第 911 页	修改了 20.13（8）CPU 总线仲裁的描述
第 911 页	在 20.13（11）DMA 起始条件 中修改了描述
第 917 页	修改了第 22 章中断/异常处理功能的描述
第 917 页	增加了 22.1 特征的描述
第 918 页	增加了表 22-1 中断源列表的注
第 926 页	修改了 22.3 可屏蔽中断的描述
第 933 页	增加了 22.3.4 中断控制寄存器（xxICn）的注意事项
第 935 页	增加了表 22-2 中断控制寄存器（xxICn）的注
第 937 页	在 22.3.5 中断屏蔽寄存器 0 到 3（IMR0 到 IMR3）增加了注删除了描述
第 944 页	增加了 22.5.1（2）恢复中的注意事项
第 951 页	修改了 22.6.2（4）噪声消除控制器（NFC）的描述
第 952 页	增加了 22.7 CPU 中断确认时间的描述
第 953 页	修改了图 22-15 V850ES/SG2 确认中断请求信号的流水操作（大纲）的描述
第 954 页	增加了图 22-16 V850ES/SG2 确认中断请求信号的流水操作（大纲）
第 955 页	增加了 22.9（2）V850ES/SG2-H 的中断控制寄存器（xxICn）
第 955 页	增加了 22.9（3）V850ES/SG2 与 V850ES/SG2-H 的中断控制寄存器（xxICn）
第 955 页	增加了 22.9（4）服务优先级寄存器（ISPR）
第 960 页	在 24.2（1）功率节省控制寄存器（PSC）中增加了注意事项
第 963 页	增加了 24.3.2 释放暂停状态的注
第 966 页	增加了 24.4.2 释放空闲 1 模式的注
第 968 页	增加了 24.5.2 释放空闲 2 模式的注
第 969 页	增加了表 24-7 空闲 2 模式下操作状态的描述
第 971 页	增加了 24.6.2 释放停止模式的注
第 973 页	增加了表 24-9 停止模式下操作状态的描述
第 975 页	修改了 24.7.1 设置与运行状态的注意事项
第 975 页	增加了 24.7.2 释放副时钟运行模式的注
第 976 页	修改了表 24-10 副时钟运行模式下运行状态的描述
第 978 页	增加了 24.8.2 释放副空闲模式的注
第 979 页	增加了表 24-12 副空闲模式下运行状态的描述
第 980 页	增加了 25.1（1）复位源的注
第 981 页	修改了 25.1（2）紧急操作模式的描述
第 981 页	增加了图 25-1（b）V850ES/SG2-H
第 982 页	增加了 25.2 检查复位源的寄存器的描述
第 982 页	增加了 25.2（1）复位源标志寄存器（RESF）的注
第 983 页	修改了表 25-1 在复位 RESET 管脚输入情况下的硬件状态的描述
第 985 页	修改了表 25-2 看门狗定时器 2 复位期间硬件状态的描述
第 987 页	修改了 25.3.3 由低电压监测器（LVIREs）（仅限于 V850ES/SG2）引起的复位操作

页码	修订说明
第 987 页	修改了 表 25-3 由低电压检测器引起的复位操作期间的硬件状态 的描述
第 988 页	增加了 25.3.4 由时钟监测器 (CLMRES) 引起的复位操作
第 991 页	增加了 25.3.5 (2) 固件操作 (仅限 flash 存储器版本) 的描述与注意事项
第 999 页	增加了第 27 章低电压检测器的描述
第 999 页	删除了 27.1 特征的描述
第 1000 页	在 27.3 (1) 低电压检测寄存器 (LVIM) 中增加了注意事项
第 1001 页	在 27.3 (2) 低电压检测电平选择寄存器 (LVIS) 中增加了注意事项
第 1006 页	删除了图 28-1 电阻的描述
第 1015 页	增加了 30.2 存储器配置的描述
第 1017 页	增加了图 30-1 (b) V850ES/SG2-H (640 KB)
第 1019 页	修改了表 30-3. 安全功能的描述
第 1020 页	增加了表 30-4. 安全功能
第 1022 页	增加了 30.4.2 通信模式的描述
第 1024 页	增加了表 30-5 专用 flash 存储器编程器 (PG-FP4, PG-FP5) 的信号连接
第 1025、1026 页	修改了表 30-6 V850ES/SG2, V850ES/SG2-H flash 存储器写入适配器 (FA-100GF-3BA-A, FA-100GC-8EU-A) 的连线的描述
第 1027、1028 页	修改了图 30-6 V850ES/SG2 flash 存储器写入适配器 (FA-100GF-3BA-A) (在 CSIB0 + HS 模式) 的连线的标题
第 1029、1030 页	修改了图 30-7 V850ES/SG2, V850ES/SG2-H flash 存储器写入适配器 (FA-100GC-8EU-A) (在 CSIB0 + HS 模式) 的连线的标题
第 1031 页	修改了图 20-8 操作 flash 存储器过程的描述
第 1039 页	修改了 30.5.2 (2) 中断支持的描述
第 1041 页	增加了 30.5.5 (1) FLMD0 管脚的注意事项
第 1043 页	增加了第 31 章 片上调试功能的描述
第 1044 页	增加了 31.2 连接电路示例的描述
第 1044 页	修改了 31.3 (2) DCK 的描述
第 1047 页	增加了 31.4 (1) 片上调试模式寄存器 (OCDM) 的注
第 1052 页	修改了 31.7 (4), (6), (7) 中的描述
第 1055 页	修改了 32.2 电容中的描述
第 1055 页	增加了 32.3 (2) V850ES/SG2-H
第 1057 页	增加了 32.4.1 主时钟振荡器特性的描述
第 1057 页	修改了 32.4.1 (i) KYOCERA KINSEKI 公司: 晶振中的描述
第 1058 页	增加了 32.4.1 (ii) Murata Mfg 有限公司: 陶瓷共振器
第 1059 页	增加了 32.4.2 副时钟振荡器特性的注
第 1060 页	增加了 32.4.3 (2) V850ES/SG2-H
第 1061 页	增加了 32.5 (2) V850ES/SG2-H
第 1063 页	修改了 32.6.1 I/O 电平的描述
第 1065 页	增加了 32.6.2 (2) V850ES/SG2-H
第 1066 页	删除了 32.7 (1) 停止模式中的描述并添加了注
第 1068 页	增加了 32.8.1 CLKOUT 输出时序的描述

(10/10)

页码	修订说明
第 1068 页	增加了 32.8.2 (1) 多总线模式的注意事项
第 1069 页	修改并增加了 32.8.2 (1) (a) 读/写周期 (CLKOUT 异步) 中的描述
第 1074 页	增加了 32.8.2 (2) 分离总线模式的注意事项
第 1074 页	修改并增加了 32.8.2 (2) (a) 读周期 (CLKOUT 异步) 中的描述: 在分离总线模式
第 1076 页	增加了 32.8.2 (2) (b) 写周期 (CLKOUT 异步) 的描述: 在分离总线模式
第 1078 页	增加了 32.8.2 (2) (c) 读周期 (CLKOUT 异步) 的描述: 在分离总线模式
第 1079 页	增加了 32.8.2 (2) (b) 写周期 (CLKOUT 异步) 的描述: 在分离总线模式
第 1080 页	增加了 32.8.2 (3) (a) CLKOUT 异步 的描述
第 1083 页	增加了 32.9 (2) 中断, FLMD0 管脚时序 的注
第 1084 页	增加了 32.9 (6) (b) 从模式 的描述
第 1089 页	修改了 32.9 (12) LVI 电路特殊点 (仅限 V850ES/SG2) 的标题
第 1090 页	增加了 32.9 (13) RAM 保持检测 的描述
第 1091 页	增加了 32.10 (1) 基本特性 的描述
第 1091 页	增加了 32.10 (2) 串行写入操作特性 的描述
第 1093 页	增加了 32.10 (3) (b) V850ES/SG2-H
第 1096 页	修改了 第 34 章 焊接条件 中的描述
第 1101 页	增加了 图 A-1 (3) 当使用片上仿真器 IE-V850E1-CD-NW
第 1102 页	增加了 图 A-1 (4) 当使用 MINICUBE QB-V850MINI
第 1103 页	修改了 A.2 语言处理软件 中的描述
第 1106 页	修改了 图 A-3 系统配置 (使用 QB-V850ESSX2) 中的描述
第 1108 页	修改了 A.4.2 当使用 IECUBE QB-V850ESSX2 中的注
第 1109 页	增加了 A.4.3 当使用片上仿真器 IE-V850E1-CD-NW
第 1110 页	增加了 A.4.4 当使用 MINICUBE QB-V850MINI
第 1111 页	修改了 A.5 调试工具 (软件) 的描述
第 1112 页	修改了 A.7 flash 存储器写入工具 的描述
第 1113 页	增加了 附录 BV850ES/SG2 与 V850ES/SG2-H 之间的主要区别
第 1130、 1131、1133、 1135、1136 页	增加了 D.2 指令设置 (按字母次序) 的注

E.2 先前版本的修订记录

到此版本的修改记录显示如下。“应用于”：指明应用版本的章节。

(1/11)

版本	修订说明	应用于:
第二版	增加了 1.5 管脚配置（顶层）说明	第一章介绍
	修改了 2.1（1）端口管脚	第二章管脚功能
	修改了 2.1（2）无端口管脚	
	修改了表 2-2 各种状态下管脚状态的描述	
	修改了图 2-1 管脚 I/O 电路	
	增加了 3.3.1 特定操作模式	第三章 CPU 功能
	增加了 3.4.5 要求地址空间使用中的注意事项	
	增加了 3.4.6 外围 I/O 寄存器中的注	
	增加了 3.4.9（2）访问指定片上周边 I/O 寄存器	
	增加了 3.4.9（3）使用 flash 存储器版本时的注意事项	第四章端口功能
	修改了 4.3 端口配置	
	增加了 4.4 方框图	
	增加了 4.6 注意事项	
	修改了 5.2.1 当访问 ROM，内部 RAM，或者片上周边 I/O	第五章总线控制功能
	增加了 5.6.4 可编程地址等待功能的注意事项	
	修改了 5.10 总线时序	
	修改了 6.3（1）（a）设置主时钟运行 → 副时钟运行的示例	第六章时钟通用功能
	修改了 6.3（1）（b）设置副时钟运行 → 主时钟运行的示例	
	修改了 6.5.3（1）使用 PLL	
	修改了第七章 16 位时间/事件计数器 P（TMP）	第七章 16 位时间/事件计数器 P（TMP）
	修改了第八章 16 位时间/事件计数器 Q（TMQ）	第八章 16 位时间/事件计数器 Q（TMQ）
	修改了 9.4.1 间隔计时器模式	第九章 16 位间歇计数器 M（TMM）
	增加了 9.4.2 注意事项	
	修改了 10.2 配置	第十章 钟表计时器功能
	修改了 10.4.1 钟表计时器操作	
	修改了 11.3（1）看门狗定时器模式寄存器 2（WDTM2）中的注意事项	第十一章看门狗定时器 2 功能

(2/11)

版本	修订说明	应用于:
第二版	增加了 12.2 (1) 实时输出缓冲寄存器 0L, 0H (RTBL0, RTBH0) 注意事项 2	第十二章实时输出功能 (RTO)
	增加了 12.3 (1) 实时输出模式寄存器 0 (RTPM0) 注意事项 3	
	增加了 13.2 功能	第十三章 A/D 转换器
	修改了 13.3 配置	
	增加了 13.4 (1) A/D 转换模式寄存器 0 (ADA0M0) 的注意事项 4	
	增加了 13.4 (5) A/D 结果寄存器 n, nH (ADA0CRn, ADA0CRnH) 的注意事项 2	
	增加了 13.5.1 <9>	
	增加了 13.6 (8) 保持模式	
	修改了图 14-1 D/A 转换器方框图	第十四章 D/A 转换器
	修改了 14.4.2 实时输出模式操作	
	增加了 14.4.3 (7)	
	修改了 15.4 (1) UARTAn 控制寄存器 0 (UAnCTL0)	第十五章异步串行接口 A (UARTA)
	增加了 15.6.2 SBF 传输/接收格式的备注	
	增加了图 15-15RXDAn 判作噪声的时序	
	在 15.7 (2) UARTAn 控制寄存器 1 (UAnCTL1) 中增加了注意事项	
	在 15.7 (3) UARTAn 控制寄存器 2 (UAnCTL2) 中增加了注意事项	
	增加了 15.8 注意事项	
	增加了 16.3 配置中的备注	第十六章 3 线可变长度串行 I/O (CSIB)
	修改了 16.4 (1) CSIBn 控制寄存器 0 (CBnCTL0)	
	在 16.4 (2) CSIBn 控制寄存器 1 (CBnCTL1) 中增加了注	
	在 16.4 (3) CSIBn 控制寄存器 2 (CBnCTL2) 中增加了注	
	修改了 16.5 操作	
	修改了 16.6 (1) SCKBn 管脚	
	修改了 16.7 操作流程	
	增加了 17.4 (1) IIC 控制寄存器 0 到 2 (IICC0 to IICC2) 中的注	第十七章 I ² C 总线
	增加了 17.4 (2) IIC 控制寄存器 0 到 2 (IICC0 to IICC2) 中的注	
	增加了 17.16.3 从模式	
	修改了 18.3 (17) IEBus 时钟选择寄存器 (OCKS2)	第十八章 IEBus 控制器
	修改了 20.3 控制寄存器	第二十章 DMA 功能 (DMA 控制器)
	修改了 20.4 传输目标	
	修改了 20.5 传输模式	
	修改了 20.6 传输类型	
	修改了 20.7DMA 通道优先级	
	修改了 20.8DMA 传输相关时间	
	修改了 20.9DMA 传输开始因素	
	修改了 20.10DMA 传输放弃因素	

版本	修订说明	应用于:
第二版	修改了 20.11 DMA 传输结束	第二十章 DMA 功能 (DMA 控制器)
	增加了 20.12 操作时序	
	修改了 20.13 注意事项	
	修改了 21.3 (2) CRC 数据寄存器 (CRCD) 的注意事项	第二十一章 CRC 功能
	增加了 表 22-1 中断源列表 的注 1	第二十二章 中断/异常处理功能
	增加了 22.3.5 中断屏蔽寄存器 0 到 3 (IMR0 到 IMR3) 的注和注意事项	
	修改了图 22-14 确认中断请求 (轮廓) 的流水操作	
	增加了 22.9 注意事项	
	增加了 23.1 功能 中的注意事项	第二十三章 关键中断功能
	增加了表 24-5 空闲 1 模式下操作状态的注	第二十四章 保持功能
	增加了表 24-7 空闲 2 模式下操作状态的注	
	修改了 24.7.1 设置与操作状态的注意事项 2	
	修改了表 24-10 副时钟运行模式下运行状态的注意事项	
	增加了 25.2 (1) 复位源标志寄存器 (RESF)	第二十五章 复位功能
	增加了图 26-1 电阻的注意事项	第二十六章 电阻
	增加了 27.2 (1) IIC 地址修正寄存器 0 到 3 (CORAD0 to CORAD3) 中的注	第二十七章 ROM 修正功能
	修改了第二十八章 flash 存储器	第二十八章 flash 存储器
	增加了第 29 章片上调试功能	第 29 章 片上调试功能
	增加了第三十章电气特性 (目标值)	第三十章 电气特性 (目标值)
	增加了 第三十一章封装图	第三十一章 封装图
	增加了附录 A 寄存器序号	附录 A 寄存器序号
	增加了附录 B 指令设置列表	附录 B 指令设置列表
	增加了 附录 C 再版修订记录	附录 C 再版修订记录

(4/11)

版本	修订说明	应用于：
第三版	删除了先前版本提示，增加了新产品的注	所有页面
	修改了 1.4 分类信息	第一章 介绍
	增加了 2.1 (1) 端口管脚的注 1	第二章 管脚功能
	修改了 2.1 (2) 无端口管脚	
	修改了表 2-2 各种状态下管脚状态的描述	
	修改了表 3-2 系统寄存器数量	第三章 CPU 功能
	修改了 3.2.2 (4) 程序状态字 (PSW)	
	增加了 3.2.2 (6) 异常/调试陷阱状态保存寄存器 (DBPC 和 DBPSW) 的描述	
	增加了图 3-1 地址空间镜像的注意事项	
	增加了图 3-2 数据存储映射 (物理地址) 的注 2	
	增加了 3.4.4 (4) 可编程外围 I/O 区域	
	修改了图 3-14 要求存储器映射	
	修改了 3.4.6 外围 I/O 寄存器	
	修改了 3.4.9 (1) 首先设置寄存器	
	修改了 3.4.9 (2) 访问指定片上周边 I/O 寄存器	
	修改了 3.4.9 (3) 使用 flash 存储器版本时的注意事项	
	增加了 3.4.9 (4) sld 指令与中断请求之间冲突的限制	
	修改了表 4-4 端口 0 复用功能管脚	第四章 端口功能
	修改了表 4-8 端口 5 复用功能管脚	
	增加了 4.3.6 (1) (a), (b) 的备注	
	修改了图 4-8. E-3 类型的框图	
	修改了图 4-18. N-3 类型的框图	
	增加了表 4-15 使用端口管脚作为复用管脚的注 3	
	修改了 4.6.1 (1)	
	修改了 4.6.4 P05/INTP2/DRST 管脚注意事项	
	修改了 5.1 特征	第五章 总线控制功能
	增加了图 5-1 数据存储映射 (物理地址) 的注 2: 物理地址	
	增加了 5.6.2 外部等待功能的描述	
	修改了图 6-1 时钟产生器	第六章 时钟通用功能
	增加了 6.3 (2) 内部振荡器模式寄存器 (RCM) 的注意事项	
	增加了 6.3 (3) CPU 运行时钟状态寄存器 (CCLS) 中的注	
	增加了表 6-1 各个时钟运行状态	
	增加了 6.5.2 (2) 时钟控制寄存器 (CKC) 的注 3	
	增加了 6.5.2 (3) 锁定寄存器 (LOCKR) 的描述	
	修改了 6.5.3 (1) 使用 PLL	

版本	修订说明	应用于:
第三版	在 7.4 (5) TMPn I/O 控制寄存器中 2 (TPnIOC2) 中增加了注意事项 3	第七章 16 位时间/事件计数器 P (TMP)
	修改了 7.5.1 (2) (a) 如果 TPnCCR0 寄存器被设置为 0000H 操作	
	修改了图 7-10 外部事件计数模式的基本时序	
	修改了 7.5.2 (2) 外部事件计数模式的基本时序	
	修改了 7.5.3 (2) (b) PWM 波形的 0%/100%输出	
	修改了 7.5.5 (2) (b) PWM 波形的 0%/100%输出	
	修改了 7.5.7 脉冲宽度测量模式 (TPnMD2 到 TPnMD0 位= 110)	
	增加了 7.6 选择器功能	
	增加了 7.7 注意事项	
	在 8.4 (5) TMQ0 I/O 控制寄存器中 2 (TQ0IOC22) 中增加了注意事项 3	第八章 16 位时间/事件计数器 Q (TMQ)
	修改了 8.5.1 (2) (a) 如果 TQ0CCR0 寄存器被设置为 0000H 操作	
	修改了图 8-10 外部事件计数模式的基本时序	
	修改了 8.5.2 (2) 外部事件计数模式的基本时序	
	增加了图 8-1 设置外部触发脉冲输出模式寄存器中的备注 2	
	修改了 8.5.3 (2) (b) PWM 波形的 0%/100%输出	
	增加了图 8-26PWM 输出模式寄存器设置的备注 2	
	修改了 8.5.5 (2) (b) PWM 波形的 0%/100%输出	
	修改了图 8-31 自由运行计时器模式寄存器设置	
	修改了 8.5.7 脉冲宽度测量模式 (TQ0MD2 到 TQ0MD0 位= 110)	
	增加了 8.6 选择器功能	
	增加了 8.7 注意事项	
	修改了 9.3 (1) TMM0 控制寄存器 (TM0CTL0)	第九章 16 位间歇计数器 M (TMM)
	增加了 9.4 功能 中的注意事项	
	修改了 9.4.1 间隔计时器模式	
	修改了 10.3 (1) 预分频器模式寄存器 0 (PRSM0)	第十章 钟表计时器功能
	增加了 10.3 (2) 预分频器比较寄存器 0 (PRSCM0) 中的描述	
	增加了 10.3 (3) 钟表定时器操作模式寄存器 (WTM) 中的描述	
	修改了 11.3 (1) 看门狗定时器模式寄存器 2 (WDTM2) 中的描述	第十一章 看门狗定时器 2 功能
	增加了 11.4 功能 中的描述	
	修改了表 13-2 和表 13-3	第十三章 A/D 转换器
	修改了 13.5.1 基本操作	
	增加了 13.5.2 转换模式时序	
	增加了 13.6 (4) 复用 I/O 的描述	
	增加了 13.6 (6) 内部平衡电路	
	增加了 13.6 (9) 保持模式中的描述	

(6/11)

版本	修订说明	应用于:
第三版	修改了 15.4 (1) UARTAn 控制寄存器 0 (UAnCTL0)	第十五章 异步串行接口 A (UARTA)
	修改了 15.4 (4) UARTAn 选项控制寄存器 0 (UAnCTL0)	
	增加了 15.6.8 接收错误中的描述	
	修改了 15.7 (1) (a) 基本时钟	
	修改了表 15-3. 波特率发生器的数据设置	
	增加了 15.8 功能 中的描述	
	修改了 16.2 特征	第十六章 3 线可变长度串行 I/O (CSIB)
	修改了 16.4 (1) CSIBn 控制寄存器 0 (CBnCTL0)	
	修改了 16.4 (2) CSIBn 控制寄存器 1 (CBnCTL1)	
	增加了 16.4 (4) CSIBn 状态寄存器 (CBnSTR) 中的描述	
	修改了 16.5.7 连续模式 (从模式, 接收模式)	
	修改了 16.5.8 时钟时序	
	修改了 16.6 (1) SCKBn 管脚	
	增加了 16.9 注意事项	
	修改了 17-4 I ² C0n 的框图	第十七章 I ² C 总线
	增加了 7.4 (1) IIC 控制寄存器 0 到 2 (IICC0 to IICC2) 中的描述	
	增加了 17.4 (4) IIC 控制寄存器 0 到 2 (IICC0 to IICC2) 中的备注	
	修改了表 17-2 时钟设置	
	增加了 17.6.1 启动条件的注意事项	
	增加了 17.15 功能 中的描述	
	第十九章 CAN 控制器修订	第十九章 CAN 控制器
	修改了图 20-2 DMA (2) 优先级	第二十章 DMA 功能 (DMA 控制器)
	修改了 22.3.4 中断控制寄存器 (xxICn)	第二十二章 中断/异常处理功能
	修改了 22.3.8 看门狗计时器模式寄存器 2 (WDTM2)	
	修改了 22.8 中断未被 CPU 确认的周期	
	增加了 22.9 功能 中的描述	
	修改了 23.2 (1) 返回值模式寄存器 (KRM)	
	增加了 23.3 注意事项	第二十三章 关键中断功能
	修改了表 24-1 保持模式	第二十四章 保持功能
	修改了图 24-1 状态传输	
	修改了 24.2 (2) 功率保存模式寄存器 (PSMR)	
	修改了 24.6.1 设置和运行状态	
	修改了 24.8.1 设置和运行状态	
	第二十五章 复位功能的修订	第二十五章 复位功能
	增加了第二十六章 时钟监测	第二十六章 时钟监测
	增加了第二十七章 低电压检测器 (LVI)	第二十七章 低电压检测器 (LVI)

(7/11)

版本	修订说明	应用于:
第三版	增加了 29.2 (1) 地址修正寄存器 0 到 3 (CORAD0 to CORAD3)	第二十九章 ROM 修正功能
	增加了 29.4 注意事项	
	修改了第三十八章 flash 存储器	第三十章 flash 存储器
	修改了图 30-1 flash 存储器映射	
	修改了表 30-4 专门 flash 存储器编程器 (PG-FP4) 的信号连接	
	修改了表 30-5 V850ES/SG2 flash 存储器烧写适配器 (FA-100GF-JBT 和 FA-100GC-8EA) 的连线	
	修改了图 30-6 和 30-7	
	修改了图 30-10 使用的内部资源	
	增加了第 31 章片上调试功能的注意事项	第 31 章 片上调试功能
	增加了 31.6.1 安全 ID 的注	
	增加了 31.7 功能 中的描述	
	修订了第三十二章 电气特性	第三十二章 电气特性
	增加了 第 34 章 推荐焊接条件	第 34 章 推荐焊接条件
	增加了附录 A 开发工具	附录 A 开发工具
	增加了 D.2 先前版本修订历史	附录 D 再版修订记录
第四版	删除了如下产品 (开发完成) 开发中的提示	所有页面
	- GF 封装 μPD703260、703260Y、703261、703261Y、703270、703270Y、703271、703271Y - GC 封装 μPD703260、703260Y、703262、703262Y、703263、703263Y、703270、703270Y、703272、703272Y、703273、703273Y、703280、703280Y、703281、703281Y、703282、703282Y、703283、703283Y	
	修改了表 1-2 V850ES/SJ2 产品列表	第一章 介绍
	修改了 2.2 管脚状态	第二章 管脚功能
	增加了 2.4 注意事项	第四章 端口功能
	修改了 3.4.9 (2) 访问指定片上周边 I/O 寄存器和 (3) 系统保留区域	
	增加了 表 4-5 端口 1 复用功能管脚的注意事项	
	在 4.3.2 (2) 端口 1 模式寄存器 (PM1) 中修改了注意事项 1 添加了注意事项 2	
	修改了表 4-6 端口 3 复用功能管脚	
	修改了表 4-7 端口 4 复用功能管脚	
	增加了 表 4-8 端口 5 复用功能管脚的注意事项 1	
	修改了图 4-7. D-3 类型的框图	
	修改了图 4-8. E-3 类型的框图	

版本	修订说明	应用于:
第四版	修改了图 4-13. G-5 类型的框图	第四章 端口功能
	修改了图 4-14. G-6 类型的框图	
	修改了图 4-15. G-12 类型的框图	
	修改了图 4-18. N-2 类型的框图	
	修改了图 4-19. N-3 类型的框图	
	修改了图 4-26. U-10 类型的框图	
	修改了图 4-27. U-11 类型的框图	
	增加了表 4-15 使用端口管脚作为复用管脚的注意事项	
	增加了 4.6.5 上电时 P10, P11, 和 P53 的注意事项	
	修改了 5.1 特征	第五章 总线控制功能
	增加修改了 6.3 (1) 处理器时钟控制寄存器 (PCC) 的描述	第六章 时钟通用功能
	增加了 6.3 (1) (a) 设置主时钟运行 → 副时钟运行的示例中的描述	
	增加了 6.3 (1) (b) 设置副时钟运行 → 主时钟运行的示例中的描述	
	增加了 6.5.2 (4) PLL 锁定时间指定寄存器 (PLLS) 的注意事项 2	
	修改了 7.4 (7) TMPn 捕捉/比较寄存器 0 (TPnCCR0)	第七章 16 位时间/事件计数器 P (TMP)
	修改了 7.4 (8) TMPn 捕捉/比较寄存器 1 (TPnCCR1)	
	修改了 7.4 (9) TMPn 计数器读缓冲寄存器 (TPnCNT)	
	修改了图 7-4 间隔计时器操作模式的寄存器设置	
	修改了图 7-10 外部时间计数模式的基本时序	
	修改了图 7-11 外部事件计数模式操作的寄存器设置	
	增加了 7.5.2 (2) 外部时间计数模式的基本时序的注意事项 2	
	修改了 7.5.2 (2) (c) TPnCCR1 寄存器操作	
	修改了图 7-17 外部触发脉冲输出模式的基本时序	
	增加了 7.5.3 外部触发脉冲输出模式 (TPnMD2 到 TPnMD0 位= 010) 的描述	
	增加了图 7-18 设置外部触发脉冲输出模式寄存器中的备注 2	
	增加了图 7-22 设置单脉冲输出模式寄存器中的备注 2	
	修改了图 7-23 单脉冲输出模式下软件处理流程	
	在图 7-22 设置 PWM 输出模式寄存器中增加了注 2 并做了修改	
	修改了 7.7 (1) 捕获操作	
	修改了 8.4 (7) TMQ0 捕捉/比较寄存器 0 (TQ0CCR0)	第八章 16 位时间/事件计数器 Q (TMQ)
	修改了 8.4 (8) TMQ0 捕捉/比较寄存器 1 (TQ0CCR1)	

版本	修订说明	应用于:
第四版	修改了 8.4 (9) TMQ0 捕捉/比较寄存器 2 (TQ0CCR2)	第八章 16 位时间/事件计数器 Q (TMQ)
	修改了 8.4 (10) TMQ0 捕捉/比较寄存器 3 (TQ0CCR3)	
	修改了 8.4 (11) TMQ0 计数器读缓冲寄存器 (TQ0CNT)	
	修改了图 8-4 间隔计时器操作模式的寄存器设置	
	修改了图 8-10 外部时间计数模式的基本时序	
	修改了图 8-11 外部事件计数模式操作的寄存器设置	
	增加了 8.5.2 (2) 外部事件计数模式的基本时序的注意事项 2	
	修改了 8.5.2 (2) (c) TQ0CCR1 到 TQ0CCR3 寄存器操作	
	修改了图 8-17 外部触发脉冲输出模式的基本时序	
	增加了 8.5.3 外部触发脉冲输出模式 (TPnMD2 到 TPnMD0 位= 010) 的描述	
	增加了图 8-18 设置外部触发脉冲输出模式寄存器中的备注	
	增加了图 8-22 设置单脉冲输出模式寄存器中的备注	
	修改了图 8-23 单脉冲输出模式下软件处理流程	
	在图 8-22 设置 PWM 输出模式寄存器中增加了注并做了修改	
	修改了 8.7 (1) 捕获操作	
	修改了图 10-1 钟表计时器的框图	第十章 钟表计时器功能
	修改了图 11-1 看门狗定时器 2 的框图	第十一章 看门狗定时器 2 功能
	修改了 11.3 (1) 看门狗计时器模式寄存器 2 (WDTM2)	
	增加了 12.2 (1) 实时输出缓冲寄存器 0L, 0H (RTBL0, RTBH0)	第十二章 实时输出功能 (RTO)
	增加了 13.4 (1) A/D 转换模式寄存器 0 (ADA0M0) 的注意事项 3	第十三章 A/D 转换器
	增加了 13.4 (2) A/D 转换模式寄存器 1 (ADA0M1) 的注意事项 1	
	修改了表 13-2 正常转换模式下转换时间选择 (ADA0HS1 位= 0) 和表 13-3 高速转换模式下转换时间选择 (ADA0HS1 位= 1)	
	修改了 13.4 (5) A/D 结果寄存器 n, nH (ADA0CRn, ADA0CRnH)	
	修改了图 13-3 转换操作时间 (连续转换)	
	修改了 13.6 注意事项中 (9) 并增加了 (10) 到 (13)	
	增加了 13.7 (6) (b) 差分线性错误的描述	
	修改了 14.1 功能	第十四章 D/A 转换器
	修改了 14.4.3 注意事项	

(10/11)

版本	修订说明	应用于:
第四版	修改了 15.4 (4) UARTAn 选项控制寄存器 0 (UAnCTL0) 的注意事项	第十五章 异步串行接口 A (UARTA)
	修改了 15.7 (4) 波特率	
	增加修改了 16.4 (1) CSIBn 控制寄存器 0 (CBnCTL0) 的描述	第十六章 3 线可变长度串行 I/O (CSIB)
	修改了 16.5.1 单个传输模式 (主模式, 发送/接收模式)	
	修改了 16.5.2 单个传输模式 (主模式, 接收模式)	
	修改了 16.5.3 连续模式 (主模式, 发送/接收模式)	
	修改了 16.5.4 连续模式 (主模式, 接收模式)	
	修改了 16.5.5 连续接收模式 (错误)	
	修改了 16.5.6 连续模式 (从模式, 发送/接收模式)	
	修改了 16.5.7 连续模式 (从模式, 接收模式)	
	增加了 16.5.8 时钟时序的注意事项	
	修改了 16.6 (1) SCKBn 管脚	
	增加了 16.9 注意事项 (3)	
	修改了图 17-4 I ² C0n 的框图	第十七章 I ² C 总线
	增加了 17.3 注意事项 (13)	
	增加修改了 17.4 (1) IIC 控制寄存器 0 到 2 (IICC0 到 IICC2) 中的描述	
	增加修改了 17.4 (2) IIC 状态寄存器 0 到 2 (IICS0 到 IICS2) 中的描述	
	增加了 17.4 (3) IIC 标志寄存器 0 到 2 (IICF0 到 IICF2) 中的描述	
	增加了 17.4 (4) IIC 时钟选择寄存器 0 到 2 (IICL0 到 IICL2) 中的描述	
	增加了 17.4 (5) IIC 功能扩展寄存器 0 到 2 (IICX0 到 IICX2) 中的描述	
	增加了 17.4 (8) IIC 移位寄存器 0 到 2 (IIC0 到 IIC2) 中的描述	
	增加了 17.4 (9) 从地址寄存器 0 到 2 (SVA0 到 SVA2) 中的描述	
	增加了 17.6.7 等待状态取消方法	
	修改了 17.7.1 (1) 开始~ 地址~ 数据~ 数据~ 停止 (正常传输/接收)	
	修改了 17.7.1 (2) 开始~ 地址~ 数据~ 开始~ 地址~ 数据~ 停止 (重新开始)	
	增加了 17.7.6 (6) 当试图产生一个重启条件 SDA0n 管脚低电平而引起仲裁失败时的<1>	
	增加了 17.7.6 (7) 当试图产生一个重启条件一个停止条件而引起仲裁失败时的<1>	
	增加了 17.7.6 (8) 当试图产生一个重启条件 SDA0n 管脚低电平而引起仲裁失败时的<1>	
	修改了图 17-21 从操作流程 (1)	

版本	修订说明	应用于:
第四版	修改了图 18-32 从传输（中断请求信号发生间隙）和图 18-33 从模式接收（中断请求信号发生间隙）	第十八章 IEBus 控制器
	增加了 19.6 寄存器的注意事项	第十九章 CAN 控制器
	增加了 20.3（1）DMA 源地址寄存器 0 到 3（DSA0 到 DSA3）中的注意事项 4	第二十章 DMA 功能（DMA 控制器）
	增加了 20.3（2）DMA 目的地址寄存器 0 到 3（DDA0 到 DDA3）中的注意事项 4	
	增加了 20.3（3）DMA 字节个数寄存器 0 到 3（DBC0 到 DBC3）中的注意事项 2	
	修改了 21.3（2）CRC 数据寄存器（CRCD）	第二十一章 功能
	增加了图 22-4 软件复位处理 中的注	第二十二章 中断/异常处理功能
	修改了 23.3 注意事项	第二十三章 关键中断功能
	修改了图 24-1 状态传输	第二十四章 保持功能
	修改了 24.4.1 设置与操作状态的注意事项 2	
	修改了 24.5.1 设置与操作状态的注意事项 2	
	修改了 24.6.1 设置与操作状态的注意事项 2	
	增加了表 24-9 停止模式下操作状态	
	修改了 24.6.3 当释放停止模式时安全的振荡器稳定时间	
	修改了 24.8.1 设置与操作状态的注意事项 2	
	修改了表 25-1 复位 RESET 管脚输入的硬件状态的注 2	第二十五章 复位功能
	修改了 25.3.5 复位功能操作流程	
	修改了 27.3（1）低电压检测寄存器（LVIM）	第二十七章 低电压检测器
	增加了 27.3（3）内部 RAM 数据状态寄存器（RAMS）的注	
	修改了 31.6.1 安全 ID	第 31 章片上调试功能
	修改了 31.6.2 设置	
	修改了第三十二章电气特性	第三十二章 电气特性
	修改了 第 34 章 推荐焊接条件	第 34 章 推荐焊接条件
	修改了 A.4.2 当使用 IECUBE QB-V850ESSX2	附录 A 开发工具
	修改了 A.7 flash 存储器写入工具	

详细信息请联系:

中国区

MCU 技术支持热线:

电话: +86-400-700-0606 (普通话)

服务时间: 9:00-12:00, 13:00-17:00 (不含法定节假日)

网址:

<http://www.cn.necel.com/> (中文)

<http://www.necel.com/> (英文)

[北京]

日电电子(中国)有限公司

中国北京市海淀区知春路 27 号

量子芯座 7, 8, 9, 15 层

电话: (+86) 10-8235-1155

传真: (+86) 10-8235-7679

[深圳]

日电电子(中国)有限公司深圳分公司

深圳市福田区益田路卓越时代广场大厦 39 楼

3901, 3902, 3909 室

电话: (+86) 755-8282-9800

传真: (+86) 755-8282-9899

[上海]

日电电子(中国)有限公司上海分公司

中国上海市浦东新区银城中路 200 号

中银大厦 2409-2412 和 2509-2510 室

电话: (+86) 21-5888-5400

传真: (+86) 21-5888-5230

[香港]

香港日电电子有限公司

香港九龙旺角太子道西 193 号新世纪广场

第 2 座 16 楼 1601-1613 室

电话: (+852) 2886-9318

传真: (+852) 2886-9022

2886-9044

上海恩益禧电子国际贸易有限公司

中国上海市浦东新区银城中路 200 号

中银大厦 2511-2512 室

电话: (+86) 21-5888-5400

传真: (+86) 21-5888-5230

[成都]

日电电子(中国)有限公司成都分公司

成都市二环路南三段 15 号天华大厦 7 楼 703 室

电话: (+86)28-8512-5224

传真: (+86)28-8512-5334