User's Manual

32

V850ES/JG3-L（片上 USB 控制器）

用户手册：硬件

RENESAS MCU
V850ES/Jx3-L 微控制器

μPD70F3794
μPD70F3795
μPD70F3796

— 初稿 —

该材料中包含的所有信息，包括产品和产品规格，仅代表文件发布时的产品信息，随时可能被瑞萨电子公司更改，恕不另行通知。请通过各种方式查阅瑞萨电子公司发布的最新信息，包括瑞萨电子公司网站(http://www.renesas.com)。

Renesas Electronics
www.renesas.com

Rev.0.01　2010 年 8 月

- **本文档所登载的内容有效期截止至 2010 年 8 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。否则因本文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瑞萨电子产品。另外，各种瑞萨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”是指瑞萨电子（Renesas Electronics Corporation）及其控股公司。

（2） 本声明中的“本公司产品”是指所有由瑞萨电子株式会社开发或制造的产品或为瑞萨电子株式会社（定义如上）开发或制造的产品。

关于CMOS器件的注意事项

① 输入引脚处的电压波形

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

② 未使用的输入引脚的处理

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

③ ESD防护措施

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

④ 初始化之前的状态

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

⑤ 电源开关顺序

一个器件内部工作和外部接口使用不同电源的情况下，按照规定，应先在接通内部电源之后再接通外部电源。当关闭电源时，按照规定，先关闭外部电源再关闭内部电源。如果电源开关顺序颠倒，可能会导致器件的内部组件过电压，产生异常电流，从而引起内部组件的误操作和性能的劣化。
对每个器件电源的正确开关顺序，必须依据器件的规格说明分别进行判断。

⑥ 电源关闭状态下的输入信号

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

如何使用该手册

读者

本手册适用于那些希望了解 V850ES/JG3-L 单片机的功能，并使用这些产品设计应用系统的用户。

目的

本手册适用于帮助用户了解下述编排组织中所述的 V850ES/JG3-L 的硬件功能。

编排组织

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

硬件	架构
• 引脚功能	• 数据类型
• CPU 功能	• 寄存器组
• 片上外设功能	• 指令格式和指令集
• Flash 存储器编程	• 中断和异常处理
• 电气规范（目标）	• 流水线操作

如何阅读本手册

假定本手册的读者具备电气工程、逻辑电路以及微控制器领域的常识。

要了解 V850ES/JG3-L 单片机的全部功能。
→按照**目录**阅读本使用手册。

要查找已知名称的寄存器的详细情况
→请阅读**附录 C 寄存器索引**

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

要了解指令功能的详细情况
→分别参见可利用的 **V850ES 架构用户手册**。

要了解 V850ES/JG3-L 单片机的电气规范。
→请参阅**第三十三章 电气特性（目标）**

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

惯例

数据有效位：	高位在左，低位在右 低电平有效表示：
有效低电平表示：	$\overline{xxx}$（在引脚或信号名称上有上划线）
存储器映射地址：	高位地址在顶部，低位地址在底部
注：	正文中使用'**注**'标记的脚注
注意事项：	需特别注意的信息
备注：	补充信息
数字表示法：	二进制 ... xxxx 或 xxxxB
	十进制 ... xxxx
	十六进制 ... xxxxH

前缀表示 2 的幂（地址空间、存储容量 ）：

K （千）： $2^{10} = 1,024$

M （兆）： $2^{20} = 1,024^2$

G （千兆）： $2^{30} = 1,024^3$

相关文档

本手册中的相关文档可能包括初稿版本。但是，初稿版本并未特别标注。

V850ES/JG3-L单片机相关文档

文档名称	文档编号
V850ES 体系结构用户手册	U15943E
V850ES/JG3-L（片上 USB 控制器）硬件用户手册	U18953E

开发工具的相关文档

文档名称		文档编号
QB-V850ESJX3L 在线仿真器		准备中
QB-V850MINI，QB-V850MINIL 片上调试仿真器		U17638E
QB-MINI2 具有编程功能的片上调试仿真器		U18371E
CA850 Ver. 3.20 C 编译程序包	操作篇	U18512E
	C 语言篇	U18513E
	汇编语言	U18514E
	连接指南	U18415E
PM+ Ver. 6.30 项目管理器		U18416E
ID850QB Ver. 3.40 集成调试器	操作篇	U18604E
SM850 Ver. 2.50 系统仿真器	操作篇	U16218E
SM850 Ver. 2.00 或更新的系统仿真器	外部组件的用户开放接口规范	U14873E
SM+系统仿真器	操作篇	U18601E
	用户开放界面	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
	任务调试器	U17422E
AZ850 Ver. 3.30 系统性能分析器		U17423E
PG-FP5 Flash 存储器编程器		U18865E

其他文件

文件名称	文件编号
半导体选择指南 – 产品和封装 –	X13769X
半导体设备安装手册	**注**
瑞萨半导体设备质量等级	C11531E
瑞萨半导体设备可靠性/质量控制系统	C10983E
预防静电放电(ESD)损坏半导体设备	C11892E

注 参见 “半导体设备安装手册”网站
(http://www2.renesas.com/pkg/en/mount/index.html).

注意事项 上述列出的文件随时可能被更改，恕不另行通知。设计时，请确保使用各个文件的最新版本。

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

IECUBE是瑞萨电子公司（Renesas Electronics Corporation）在日本和德国的注册商标。
MINICUBE是瑞萨电子公司在日本和德国的注册商标，以及在美国的商标。
EEPROM 是瑞萨电子公司的商标。
Applilet 是瑞萨电子公司在日本、德国、中国、韩国、英国以及美国注册的商标。
Windows 和 Windows NT 是 Microsoft Corporation 在美国和/或其他国家的注册商标或商标。
SuperFlash 是 Silicon Storage Technology, Inc.在其它国家，包括美国和日本，注册的商标。
PC/AT 是 International Business Machines Corporation 的商标。
SPARCstation 是 SPARC International, Inc.的商标。
Solaris 和 SunOS 是 Sun Microsystems, Inc.的商标。
TRON 是实时操作系统内核（The Realtime Operating System Nucleus）的缩写。
ITRON 是工业 TRON 的缩写

[备忘录]

目 录

第一章 导言 1

1.1 概述 1

1.2 特性 3

1.3 应用领域 5

1.4 订购信息 5

1.5 引脚配置（顶视图） 6

1.6 功能框图结构 10

1.6.1 内部框图 10

1.6.2 内部单元 11

第二章 引脚功能 14

2.1 引脚功能列表 14

2.2 引脚状态 24

2.3 引脚 I/O 电路类型、I/O 缓冲器电源、及未使用引脚的连接 25

2.4 注意事项 29

第三章 CPU 功能 30

3.1 特性 30

3.2 CPU 寄存器组 31

3.2.1 程序寄存器组 32

3.2.2 系统寄存器组 33

3.3 操作模式 39

3.4 地址空间 40

3.4.1 CPU 地址空间 40

3.4.2 存储器映射图 41

3.4.3 存储区域 43

3.4.4 数据空间的绕回 47

3.4.5 地址空间的使用建议 47

3.4.6 外围 I/O 寄存器 51

3.4.7 专用寄存器 62

3.4.8 需要先行设置的寄存器 66

3.4.9 注意事项 67

第四章 端口功能 69

4.1 特性 69

4.2 端口基本配置 69

4.3 端口配置 70

4.3.1 端口 0 76

4.3.2 端口 1 80

4.3.3 端口 3 82

4.3.4 端口 4 88

4.3.5 端口 5 90

4.3.6 端口 7 95

4.3.7 端口 9 97

4.3.8 端口 CM 105

4.3.9 端口 CT 107

4.3.10 端口 DH 109
4.3.11 端口 DL 111
4.4 框图 114
4.5 使用复用功能时的端口寄存器设置 146
4.6 注意事项 154
4.6.1 设置端口引脚的注意事项 154
4.6.2 端口 n 寄存器（Pn）位操作指令的注意事项 157
4.6.3 片上调试引脚的注意事项 158
4.6.4 P05/INTP2/DRST 引脚的注意事项 158
4.6.5 电源开启时 P10，P11 和 P53 引脚的注意事项 158
4.6.6 迟滞特性 158

第五章 总线控制功能 159
5.1 特性 159
5.2 总线控制引脚 160
5.2.1 访问片内 ROM 存储器、片内 RAM 存储器或片上外设输入/输出端口时的引脚状态 161
5.2.2 各种工作模式下的引脚状态 161
5.3 存储模块功能 162
5.4 总线访问 163
5.4.1 访问所需的时钟周期数目 163
5.4.2 总线宽度设置功能 164
5.4.3 根据不同总线宽度进行访问 165
5.5 等待功能 172
5.5.1 可编程等待功能 172
5.5.2 外部等待功能 173
5.5.3 可编程等待和外部等待之间的关系 174
5.5.4 可编程地址等待功能 175
5.6 空闲状态插入功能 176
5.7 总线保持功能 177
5.7.1 功能概述 177
5.7.2 总线保持步骤 178
5.7.3 节电模式时的操作 178
5.8 总线优先权 179
5.9 总线时序 180

第六章 时钟发生器 183
6.1 概述 183
6.2 配置 184
6.3 寄存器 187
6.4 操作 193
6.4.1 各时钟的操作 193
6.4.2 时钟输出功能 194
6.4.3 外部时钟信号输入 194
6.5 PLL 功能 194
6.5.1 概述 194
6.5.2 寄存器 195
6.5.3 用法 199
6.6 如何连接谐振器 200

6.6.1 主时钟振荡器 200
6.6.2 副时钟振荡器 200

第七章 16 位定时器/事件计数器 P（TMP） 203
7.1 概述 203
7.2 配置 204
7.2.1 TMPn 使用的引脚 206
7.2.2 中断 207
7.4 寄存器 208
7.4 操作 220
7.4.1 间隔定时器模式（TPnMD2 至 TPnMD0 位＝000） 227
7.4.2 外部事件计数模式（TPnMD2 至 TPnMD0 位＝001） 238
7.4.3 外部触发脉冲输出模式（TPnMD2 至 TPnMD0 位=010） 247
7.4.4 单次触发脉冲输出模式（TPnMD2 至 TPnMD0 位＝011） 259
7.4.5 PWM 输出模式（TPnMD2 至 TPnMD0 位＝100） 267
7.4.6 自由运行定时器模式（TPnMD2 至 TPnMD0 位＝101） 276
7.4.7 脉冲宽度测量模式（TPnMD2 至 TPnMD0 位＝110） 292
7.4.8 定时器输出操作 296
7.5 注意事项 297

第八章 16 位定时器/事件计数器 Q （TMQ） 298
8.1 概述 298
8.2 配置 299
8.2.1 TMQ0 使用的引脚 301
8.2.2 中断 301
8.3 寄存器 302
8.4 操作 317
8.4.1 间隔定时器模式（TQ0MD2 至 TQ0MD0 位＝000） 324
8.5.2 外部事件计数模式（TQ0MD2 至 TQ0MD0 位＝001） 336
8.5.3 外部触发脉冲输出模式（TQ0MD2 至 TQ0MD0 位＝010） 346
8.4.4 单次触发脉冲输出模式（TQ0MD2 至 TQ0MD0 位 ＝ 011） 361
8.4.5 PWM 输出模式（TQ0MD2 至 TQ0MD0 位＝100） 371
8.4.6 自由运行定时器模式（TQ0MD2 至 TQ0MD0 位＝101） 382
8.4.7 脉冲宽度测量模式（TQ0MD2 至 TQ0MD0 位＝110） 402
8.4.8 定时器输出操作 407
8.5 注意事项 408

第九章 16 位间隔定时器 M (TMM) 409
9.1 特征 409
9.2 配置 410
9.3 寄存器 411
9.4 操作 413
9.4.1 间隔计时器模式 413
9.4.2 注意事项 417

第十章 钟表定时器功能 418
10.1 功能 418
10.2 配置 419

10.3 控制寄存器 421
10.4 操作 425
10.4.1 钟表定时器操作 425
10.4.2 作为间隔定时器操作 426
10.5 注意事项 428

第十一章 实时计数器 429
11.1 实时计数器的功能 429
11.2 实时计数器的配置 430
11.2.1 引脚配置 432
11.2.2 中断功能 432
11.3 控制实时计数器的寄存器 433
11.4 操作 448
11.4.1 初始化设置 448
11.4.2 在实时计数器工作期间重写各个计数器 449
11.4.3 在实时计数器工作期间读取各个计数器 450
11.4.4 在实时计数器工作期间改变 INTRTC0 中断设置 451
11.4.5 在实时计数器工作期间改变 INTRTC1 中断设置 452
11.4.6 INTRTC2 中断设置初始化 453
11.4.7 在实时计数器工作期间改变 INTRTC2 中断设置 454
11.4.8 实时计数器初始化 455
11.4.9 实时计数器的钟表误差修正示例 456

第十二章 看门狗定时器 2 460
12.1 功能 460
12.2 配置 461
12.3 寄存器 462
12.4 操作 464

第十三章 实时输出功能(RTO) 465
13.1 功能 465
13.2 配置 466
13.3 寄存器 468
13.4 操作 470
13.5 用法 471
13.6 注意事项 471

第十四章 A/D 转换器 472
14.1 概述 472
14.2 功能 472
14.3 配置 473
14.4 寄存器 476
14.5 操作 487
14.5.1 基本操作 487
14.5.2 转换操作时序 488
14.5.3 触发模式 489
14.5.4 操作模式 491
14.5.5 掉电比较模式 497

14.6 注意事项 504
14.7 A/D 转换器特征表的阅读方法 509

第十五章 D/A 转换器 513
15.1 功能 513
15.2 配置 514
15.3 寄存器 515
15.4 操作 517
15.4.1 正常模式下的操作 517
15.4.2 实时输出模式下的操作 517
15.4.3 注意事项 518

第十六章 异步串行接口 A (UARTA) 519
16.1 特性 519
16.2 配置 520
16.2.1 各个通道的引脚功能 522
16.3 UARTA 和其它串行接口的模式转换 523
16.3.1 UARTA0 和 CSIB4 的模式转换 523
16.3.2 UARTA1 和 I^2C02 的模式转换 524
16.3.3 UARTA2 和 I^2C00 的模式转换 525
16.4 寄存器 526
16.5 中断请求信号 533
16.6 操作 534
16.6.1 数据格式 534
16.6.2 UART 发送 536
16.6.3 连续发送过程 537
16.6.4 UART 接收 539
16.6.5 接收错误 541
16.6.6 奇偶类型和操作 543
16.6.7 LIN 发送/接收格式 544
16.6.8 SBF 发送 546
16.6.9 SBF 接收 547
16.6.10 接收数据噪声过滤器 548
16.7 专用波特率发生器 549
16.8 注意事项 557

第十七章 异步串行接口 C (UARTC) 558
17.1 特性 558
17.2 配置 559
17.2.1 各个通道的引脚功能 561
17.3 UARTC 和其它串行接口的模式转换 562
17.3.1 UARTC0 和 CSIB1 的模式转换 562
17.4 寄存器 563
17.5 中断请求信号 572
17.6 操作 573
17.6.1 数据格式 573
17.6.2 UART 发送 575
17.6.3 连续发送过程 576

17.6.4 UART 接收 578
17.6.5 接收错误 580
17.6.6 奇偶类型和操作 582
17.6.7 LIN 发送/接收格式 583
17.6.8 SBF 发送 585
17.6.9 SBF 接收 586
17.6.10 接收数据噪声过滤器 587
17.7 专用波特率发生器 588
17.8 注意事项 596

第十八章 3-线可变长度串行 I/O（CSIB） 597
18.1 特征 597
18.2 配置 598
18.2.1 各个通道的引脚功能 599
18.3 CSIB 和其它串行接口的模式转换 600
18.3.1 CSIB0 和 I^2C01 的模式转换 600
18.3.2 CSIB4 和 UARTA0 的模式转换 601
18.4 寄存器 602
18.5 中断请求信号 611
18.6 操作 612
18.6.1 单次传输模式（主机模式，发送模式） 612
18.6.2 单次传输模式（主机模式，接收模式） 614
18.6.3 单次传输模式（主机模式，发送/接收模式） 616
18.6.4 单次传输模式（从机模式，发送模式） 618
18.6.5 单次传输模式（从机模式，接收模式） 620
18.6.6 单次传输模式（从机模式，发送/接收模式） 623
18.6.7 连续传输模式（主机模式，发送模式） 625
18.6.8 连续传输模式（主机模式，接收模式） 627
18.6.9 连续传输模式（主机模式，发送/接收模式） 630
18.6.10 连续传输模式（从机模式，发送模式） 634
18.6.11 连续传输模式（从机模式，接收模式） 636
18.6.12 连续传输模式（从机模式，发送/接收模式） 639
18.6.13 接收错误 642
18.6.14 时钟时序 643
18.7 输出引脚 645
18.8 波特率发生器 646
18.8.1 波特率的产生 647
18.9 注意事项 648

第十九章 I^2C 总线 649
19.1 I^2C 总线和其它串行接口的模式转换 649
19.1.1 UARTA2 和 I^2C00 模式转换 649
19.1.2 CSIB0 和 I^2C01 模式转换 649
19.1.3 UARTA1 和 I^2C02 模式转换 651
19.2 特性 652
19.3 配置 653
19.4 寄存器 657
19.5 I^2C 总线模式功能 673

19.5.1 引脚配置 673
19.6 I²C 总线的定义和控制方法 674
19.6.1 开始条件 674
19.6.2 地址 675
19.6.3 传输方向指示 676
19.6.4 ACK 677
19.6.5 停止条件 678
19.6.6 等待状态 679
19.6.7 等待状态取消方法 681
19.7 I²C 中断请求信号（INTIICn） 682
19.7.1 主设备操作 682
19.7.2 从设备操作（当接收从设备地址数据（地址匹配）时） 685
19.7.3 从设备操作（当接收扩展码时） 689
19.7.4 不通信时的操作 693
19.7.5 仲裁失败时的操作（仲裁失败后作为从设备操作） 693
19.7.6 发生仲裁失败时的操作（在仲裁失败后不进行通信） 695
19.8 中断请求信号（INTIICn）的产生时序和等待控制 702
19.9 地址匹配检测方法 704
19.10 错误检测 704
19.11 扩展代码 704
19.12 仲裁 705
19.13 唤醒功能 706
19.14 通信保留 707
19.14.1 使能通信保留功能（IICFn.IICRSVn 位 = 0）时 707
19.14.2 禁止通信保留功能时（IICFn.IICRSVn 位 = 1） 711
19.15 注意事项 712
19.16 通信操作 713
19.16.1 主设备工作于单主系统 714
19.16.2 主设备工作于多主系统 715
19.16.3 从设备操作 718
19.17 数据通信的时序 721

第二十章 USB 功能控制器 USBF 728
20.1 综述 728
20.2 配置 729
20.2.1 框图 729
20.2.2 USB 存储器映射 730
20.3 外部电路配置 731
20.3.1 概述 731
20.3.2 连接配置 732
20.4 注意事项 734
20.5 请求 735
20.5.1 自动请求 735
20.5.2 其它请求 742
20.6 寄存器配置 743
20.6.1 USB 控制寄存器 743
20.6.2 USB 功能控制器寄存器列表 744
20.6.3 EPC 控制寄存器 760

20.6.4 数据保持寄存器 812
20.6.5 EPC 请求数据寄存器 835
20.6.6 桥寄存器 850
20.6.7 DMA 寄存器 854
20.6.8 批量-in 寄存器 858
20.6.9 批量-out 寄存器 859
20.6.10 外设控制寄存器 861
20.7 STALL 握手或无握手 865
20.8 特定状态下寄存器的值 866
20.9 FW 处理 868
20.9.1 初始化处理 870
20.9.2 中断服务 873
20.9.3 USB 主处理 874
20.9.4 挂起/恢复处理 900
20.9.5 电源上电后的处理 903
20.9.6 在 DMA 模式下为批量传输（OUT）接收数据 906
20.9.7 在 DMA 模式下为批量传输（IN）发送数据 911

第二十一章 DMA 功能（DMA 控制器） 916
21.1 特性 916
21.2 配置 917
21.3 寄存器 919
21.4 传输源和目的 927
21.5 传输模式 927
21.6 传输类型 928
21.7 DMA 通道优先级 929
21.8 与 DMA 传输有关的时间 930
21.9 DMA 传输启动因素 931
21.10 DMA 中止因素 932
21.11 DMA 传输结束 932
21.12 操作时序 932
21.13 注意事项 937

第二十二章 中断服务/异常处理功能 942
22.1 特性 942
22.2 不可屏蔽中断 946
22.2.1 操作 948
22.2.2 恢复 949
22.2.3 NP 标志 950
22.3 可屏蔽中断 951
22.3.1 操作 951
22.3.2 恢复 953
22.3.3 可屏蔽中断的优先级 954
22.3.4 中断控制寄存器（xxICn） 958
22.3.5 中断屏蔽寄存器 0 至 3（IMR0 至 IMR3） 960
22.3.6 正在服务中优先级寄存器（ISPR） 962
22.3.7 ID 标志 963
22.3.8 看门狗定时器模式寄存器 2（WDTM2） 963

22.4 软件异常 964
22.4.1 操作 964
22.4.2 恢复 965
22.4.3 EP 标志 966
22.5 异常陷阱 967
22.5.1 非法操作码 967
22.5.2 调试陷阱 969
22.6 多重中断服务控制 971
22.7 外部中断请求输入引脚（NMI 和 INTP0 至 INTP7） 972
22.7.1 噪声消除 972
22.7.2 边沿检测 972
22.8 CPU 中断响应时间 978
22.9 中断不被 CPU 响应期间 979
22.10 注意事项 979
22.10.1 被恢复的 PC 979

第二十三章 按键中断功能 980
23.1 功能 980
23.2 引脚功能 981
23.3 寄存器 981
23.4 注意事项 982

第二十四章 待机功能 983
24.1 概述 983
24.2 寄存器 985
24.3 HALT 模式 990
24.3.1 设置和操作状态 990
24.3.2 解除 HALT 模式 990
24.4 IDLE1 模式 992
24.4.1 设置和操作状态 992
24.4.2 解除 IDLE1 模式 993
24.5 IDLE2 模式 995
24.5.1 设置和操作状态 995
24.5.2 解除 IDLE2 模式 996
24.5.3 解除 IDLE2 模式时确保建立时间 998
24.6 STOP 模式/低电压 STOP 模式 999
24.6.1 设置和操作状态 999
24.6.2 解除 STOP 模式/低电压 STOP 模式 1003
24.6.3 低电压 STOP 模式解除后重新设置 1004
24.6.4 在解除 STOP 模式时确保振荡稳定时间 1005
24.7 副时钟操作模式/低电压副时钟操作模式 1006
24.7.1 设置和操作状态 1006
24.7.2 解除副时钟操作模式 1010
24.7.3 解除低电压副时钟操作模式 1010
24.8 副 IDLE 模式/低电压副 IDLE 模式 1011
24.8.1 设置和操作状态 1011
24.8.2 解除副 IDLE 模式/低电压副 IDLE 模式 1014
24.9 RTC 备份模式 1015

24.9.1 寄存器 1015
24.9.2 RTC 备份模式设置条件 1017
24.9.3 RTC 备份模式设置过程 1018

第二十五章 复位功能 1022
25.1 概述 1022
25.2 配置 1023
25.3 检查复位源的寄存器 1024
25.4 操作 1025
25.4.1 通过 RESET 引脚的复位操作 1025
25.4.2 通过看门狗定时器 2 的复位操作 1028
25.4.3 通过低电压检测器的复位操作 1030
25.4.4 复位结束后的即时操作 1031
25.4.5 复位功能操作流程 1033
25.5 注意事项 1034

第二十六章 时钟监控器 1035
26.1 功能 1035
26.2 配置 1035
26.3 寄存器 1036
26.4 操作 1037

第二十七章 低电压检测器 (LVI) 1040
27.1 功能 1040
27.2 配置 1040
27.3 寄存器 1041
27.4 操作 1043
27.4.1 使用内部复位信号 1043
27.4.2 用作中断 1044

第二十八章 CRC 功能 1045
28.1 功能 1045
28.2 配置 1045
28.3 寄存器 1046
28.4 操作 1047
28.5 使用方法 1048

第二十九章 稳压器 1050
29.1 概述 1050
29.2 操作 1051

第三十章 选项字节 1052
30.1 程序示例 1054

第三十一章 FLASH 存储器 1055
31.1 特性 1055
31.2 存储器配置 1056
31.3 功能概述 1056

31.4 通过专用 flash 存储器编程器进行重写 1060
31.4.1 编程环境 1060
31.4.2 通信模式 1061
31.4.3 接口 1063
31.4.4 Flash 存储器控制 1068
31.4.5 通信模式的选择 1069
31.4.6 通信命令 1070
31.4.7 在板编程时的引脚连接 1071
31.5 通过自编程进行重写 1075
31.5.1 概述 1075
31.5.2 特性 1076
31.5.3 标准自编程流程 1077
31.5.4 Flash 功能 1078
31.5.5 引脚处理 1078
31.5.6 用到的内部资源 1079

第三十二章 片上调试功能 1080
32.1 用 DCU 调试 1082
32.1.1 连接电路示例 1082
32.1.2 接口信号 1083
32.1.3 可屏蔽功能 1084
32.1.4 寄存器 1085
32.1.5 操作 1086
32.1.6 注意事项 1087
32.2 不使用 DCU 调试 1088
32.2.1 电路连接示例 1088
32.2.2 可屏蔽功能 1090
32.2.3 用户资源的保留 1091
32.2.4 注意事项 1098
32.3 ROM 安全功能 1099
32.3.1 安全 ID 1099
32.3.2 设置 1100

第三十三章 电气规范 1101
33.1 最大额定值 1101
33.2 电容 1102
33.3 工作条件 1103
33.4 振荡器特性 1105
33.4.1 主时钟振荡器特性 1105
33.4.2 副时钟振荡器特性 1107
33.4.3 PLL 特性 1108
33.4.4 内部振荡器特性 1108
33.5 稳压器特性 1109
33.6 DC 特性 1110
33.6.1 引脚特性 1110
33.6.2 供电电流特性 1113
33.6.3 数据保持特性（STOP 模式下） 1114
33.7 AC 特性 1115

33.7.1 测量条件 1115
33.7.2 CLKOUT 输出时序 1116
33.7.3 总线时序 1117
33.7.4 电源开启电源/关闭/复位时序 1124
33.8 外设功能特性 1125
33.8.1 中断时序 1125
33.8.2 按键返回时序 1125
33.8.3 定时器时序 1125
33.8.4 UART 时序 1126
33.8.5 CSIB 时序 1126
33.8.6 I^2C 总线模式 1128
33.8.7 A/D 转换器 1129
33.8.8 D/A 转换器 1130
33.8.9 LVI 电路特性 1130
33.8.10 RTC 备份模式特性 1131
33.9 Flash 存储器编程特性 1132

第三十四章 封装图 1134

附录 A 开发工具 1136
A.1 软件包 1138
A.2 语言处理软件 1138
A.3 控制软件 1138
A.4 调试工具（硬件） 1139
A.4.1 使用 IECUBE QB-V850ESJX3L 时 1139
A.4.2 使用 MINICUBE QB-V850MINI 时 1142
A.4.3 使用 MINICUBE2 QB-MINI2 时 1143
A.5 调试工具（软件） 1144
A.6 嵌入式软件 1145
A.7 Flash 存储器写入工具 1146

附录 B 产品之间的主要区别 1147

附录 C 寄存器索引 1150

附录 D 指令集列表 1169
D.1 习惯用法 1169
D.2 指令集 (按字母先后次序) 1172

第一章 导言

V850ES/JG3-L 是瑞萨电子 V850 单片微控制器系列产品中的一员，专门针对实时控制应用的低功耗而设计。

1.1 概述

V850ES/JG3-L 是一款 32 位单片微控制器，它包括 V850ES CPU 内核以及相关外设功能，例如 ROM/RAM、定时器/计数器、串行接口、A/D 转换器、D/A 转换器以及 USB 功能控制器。

除了高速实时响应特性和单时钟基本指令之外，V850ES/JG3-L 还支持乘法指令、饱和运算指令、位操作指令等为数字伺服控制应用而优化的指令，这些指令由硬件乘法器来实现。此外，作为一种实时控制系统，V850ES/还能够在需要 USB 功能控制器的应用中实现卓越的性价比，例如 PC 机周边设备，ECR 周边设备以及工业仪器。

表 1-1. V850ES/JG3-L 产品列表

类属名		V850ES/JG3-L		
部件编号		*u* PD70F3794	*μ*PD70F3795	*u* PD70F3796
内部存储器	Flash 存储器	256 KB	384 KB	512 KB
	RAM	40 KB		
存储空间	逻辑空间	64 MB		
	外部存储区域	15 MB		
外部总线接口		地址总线： 22 地址数据总线： 16 可选择的分离总线/复用总线模式		
通用寄存器		32 位×32 个寄存器		
时钟	主时钟 （振荡频率）	陶瓷/晶体 （PLL 模式下： f_X = 2.5 至 6 MHz（4/8 倍频），时钟直通模式下： f_X = 2.5 至 10 MHz） 外部时钟 （PLL 模式下： f_X = 2.5 至 6 MHz（4/8 倍频），时钟直通模式下： f_X = 2.5 至 6 MHz）		
	副时钟 （振荡频率）	晶体 (f_{XT} = 32.768 kHz)		
	内部振荡器	f_R = 220 kHz (TYP.)		
	最短指令执行时间	50 ns（主时钟（f_{XX}） = 20 MHz：未使用 USB 时） 62.5 ns（主时钟（f_{XX}） = 16 MHz：使用 USB 时）		
I/O 端口		I/O：80（5 V 耐压/N 沟道漏极开路输出可选：28）		
定时器	16 位 TMP	6 通道		
	16 位 TMQ	1 通道		
	16 位 TMM	1 通道		
	钟表定时器	1 通道		
	RTC	1 通道		
	WDT	1 通道		
实时输出端口		4 位 × 1 通道，2 位 × 1 通道，或 6 位 × 1 通道		
10 位 A/D 转换器		12 通道		
8 位 D/A 转换器		2 通道		
串行接口	CSIB	3 通道		
	UARTA/CSIB	1 通道		
	CSIB/I^2C 总线	1 通道		
	UARTA/I^2C 总线	2 通道		
	UARTA	3 通道		
	UARTC	1 通道		
	USB 功能	1 通道		
DMA 控制器		4 通道（传输目标：片上外围 I/O，内部 RAM，外部存储器）		
中断源	外部	9 (9)[注]		
	内部	55		
省电功能		HALT/IDLE1/IDLE2/STOP/副时钟/sub-IDLE/ 低压 STOP/低压副时钟/低压 sub-IDLE 模式/RTC 备份模式		
复位源		RESET 引脚输入，看门狗定时器 2 （WDT2），时钟监视器（CLM），低压检测器（LVI）		
CRC 功能		按 8 位数据单元产生 16 位错误检测码		
片上调试		支持 MINICUBE®、MINICUBE2		
工作电源电压		2.0 V@2.5 MHz，2.2 V@5 MHz，2.7 V@20 MHz，3.0 V 至 3.6 V（USB 操作时）		
运行环境温度		–40 至 +85°C		
封装		100 针 LQFP（14 × 14 mm） 121 针 FBGA（8 × 8 mm）		

注 括号中的数字表示能够解除 STOP 模式的外部中断数量。

1.2 特性

○ 最短指令执行周期：
50ns（工作于 20MHz 的主时钟（fxx）：V_{DD}= 2.7 至 3.6V）
（在 PLL 模式下：×4： 5 MHz）
62.5ns（工作于 16MHz 的主时钟（fxx）：V_{DD}= 3.0 至 3.6V）
（在 PLL 模式下：×8，1/3： 6 MHz）
200ns（工作于 5MHz 的主时钟（fxx）：V_{DD}= 2.2 至 3.6V）
（在时钟直通模式下）
400ns（工作于 2.5MHz 的主时钟（fxx）：V_{DD}= 2.0 至 3.6V）
（在时钟直通模式下）
30.5*us*（工作于 32.768 kHz 的副时钟（f_{XT}）：V_{DD}= 2.0 至 3.6V）

○ 通用寄存器：
32 位 × 32 个寄存器

○ CPU 特性：
带符号乘法（16 × 16 → 32）： 1 至 2 个时钟。
带符号乘法（32 × 32 → 64）： 1 至 5 个时钟。
饱和运算（包括上溢和下溢检测功能）。
通过使用带有 5 级流水线的 32 位 RISC 架构，大部分指令都可以在 1 个时钟周期内完成执行
通过使用哈佛架构，从内部 ROM 取出指令与从内部 RAM 访问数据可以独立执行
通过使用可变指令长度，可以达成高效代码
32 位移位指令： 1 个时钟周期
位操作指令。
长/短两种格式的数据装载/存储指令。

○ 存储空间：
64 MB 线性地址空间（用于程序和数据）
外部扩展： 最多 16 MB（包括用作内部 ROM/RAM 的 1 MB）

- 内部存储器：
  RAM： 40 KB（参见表 **1-1**）
  Flash 存储器： 256/384/512 KB（参见表 **1-1**）
- 外部总线接口：
  分离总线/复用总线输出
  8 位/16 位数据总线功能
  等待功能
  • 可编程等待功能。
  • 外部等待功能
  空闲状态功能
  总线保持功能

○ 中断和异常处理：
不可屏蔽中断： 2 个中断源

	内部			外部		
	可屏蔽	不可屏蔽	总共	可屏蔽	不可屏蔽	总共
*u*PD70F3794	1	54	55	1	8	9
*u*PD70F3795	1	54	55	1	8	9
*u*PD70F3796	1	54	55	1	8	9

软件异常： 32 个中断源
异常陷阱： 2 个中断源

○ I/O 线：
I/O 端口： 80

○ 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
○ 实时计数器：	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 线长度可变串行接口 B（CSIB）
	I^2C 总线接口（I^2C）
	UARTA/CSIB：1 通道
	UARTA/I^2C： 2 通道
	CSIB/I^2C： 1 通道
	CSIB： 3 通道
	UARTA： 3 通道
	UARTC： 1 通道
	USB 功能： 1 通道
○ A/D 转换器：	10 位分辨率： 12 通道
○ D/A 转换器：	8 位分辨率： 2 通道
○ DMA 控制器：	4 通道
○ DCU（调试控制单元）：	JTAG 接口
○ 时钟发生器：	在主时钟或副时钟操作期间
	有 7 个级别 CPU 时钟可选（f_{XX}，$f_{XX}/2$，$f_{XX}/4$，$f_{XX}/8$，$f_{XX}/16$，$f_{XX}/32$，f_{XT}）
	时钟直通模式/PLL 模式可选
○ 内部振荡时钟：	220 kHz（典型值）
○ 省电功能：	HALT/IDLE1/IDLE2/STOP/低压 STOP 模式/副时钟/sub-IDLE/
	低压副时钟/低压 sub-IDLE 模式/RTC 备份模式
○ 封装：	100 针塑封 LQFP（密间距）（14× 14）
	121 针塑封 FBGA（8 × 8）
○ 供电电压：	V_{DD} = 2.0 V 至 3.6 V（2.5 MHz）
	V_{DD} = 2.2 V 至 3.6 V（5 MHz）
	V_{DD} = 2.7 V 至 3.6 V（20 MHz）
	V_{DD} = 3.0 V 至 3.6 V（USB 操作时）

1.3 应用领域

需要 USB 接口的设备，例如 PC 机周边设备，ECR 周边设备（条码扫描器，IC 卡读卡器/写入器，打印器等），工业仪器等。

1.4 订购信息

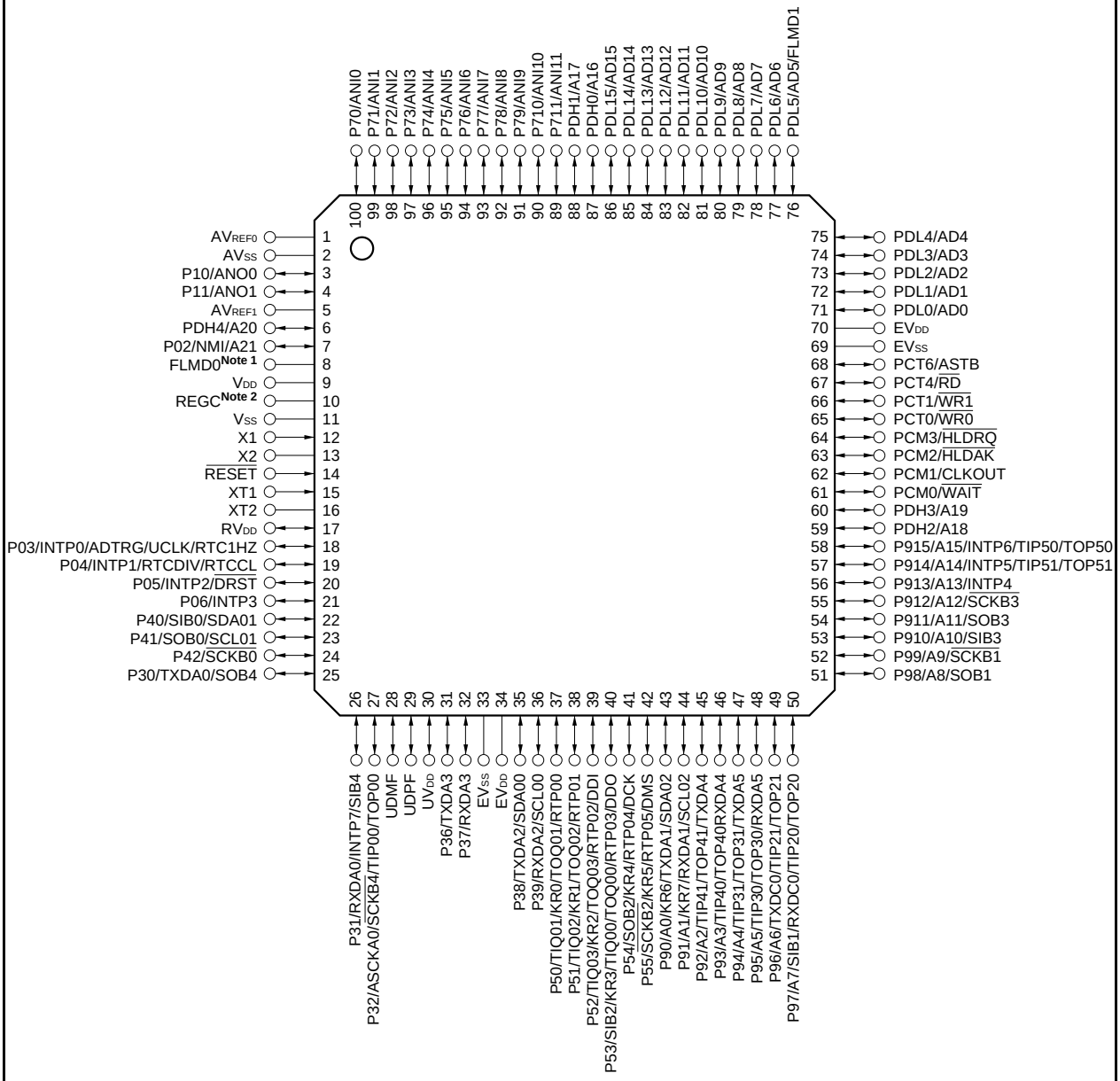
部件编号	封装	内部 Flash 存储器
*u*PD70F3794GC-UEU-AX	100 针塑封 LQFP（密间距）（14× 14）	256 KB
*u*PD70F3795GC-UEU-AX	100 针塑封 LQFP（密间距）（14× 14）	384 KB
*u*PD70F3796GC-UEU-AX	100 针塑封 LQFP（密间距）（14× 14）	512 KB
*u*PD70F3794F1-CAH-A	121 针塑封 FBGA（8 × 8）	256 KB
*u*PD70F3795F1-CAH-A	121 针塑封 FBGA（8 × 8）	384 KB
*u*PD70F3796F1-CAH-A	121 针塑封 FBGA（8 × 8）	512 KB

备注 V850ES/JG3-L 微控制器是无铅产品。

1.5 引脚配置（顶视图）

100 针塑封 LQFP（密间距）（14 × 14）

μPD70F3794GC-UEU-AX　　μPD70F3795GC-UEU-AX　　μPD70F3796GC-UEU-AX

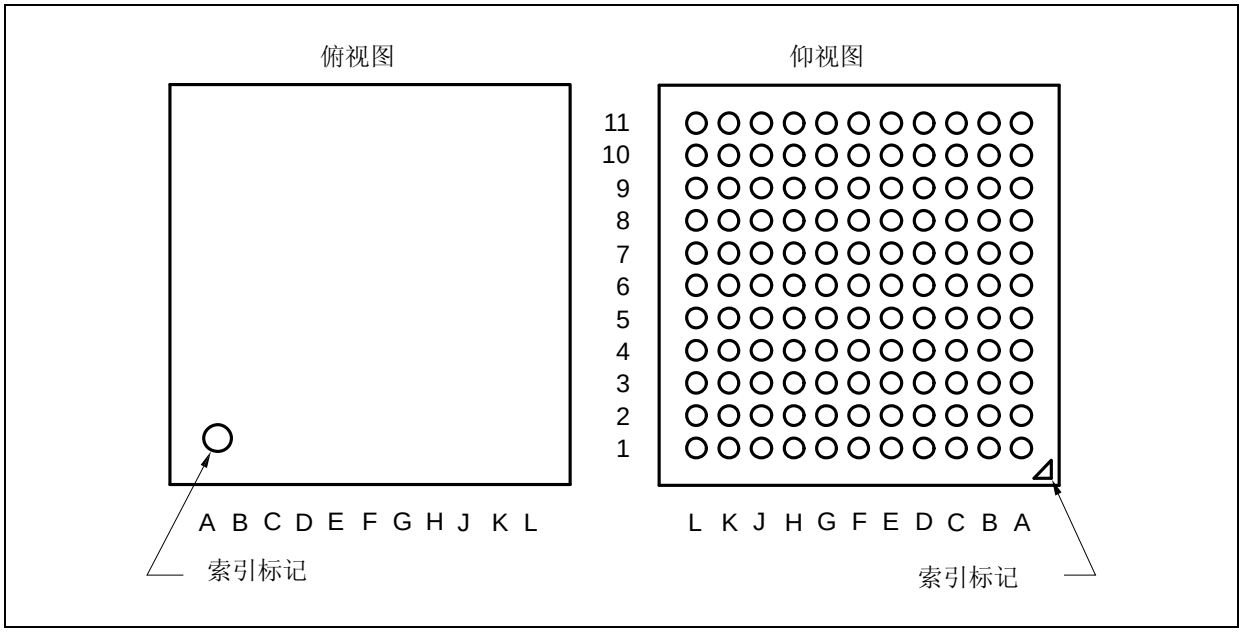


注　**1.** FLMD0 引脚在 flash 编程模式下使用。正常操作模式下，将该引脚连接至 VSS。

2. 通过一个 4.7 μF（推荐值）电容将 REGC 引脚连接至 VSS 。

121 针塑封 FBGA（8 × 8）

*u*PF70F3794F1-CAH-A　*u*PF70F3795F1-CAH-A　*u*PF70F3796F1-CAH-A



(1/2)

引脚编号	引脚名称	引脚编号	引脚名称	引脚编号	引脚名称
A1	AV_{REF0}	C1	AV_{SS}	E1	REGC[注1]
A2	AV_{REF0}	C2	AV_{SS}	E2	REGC[注1]
A3	P70/ANI0	C3	P72/ANI2	E3	P10/ANO0
A4	P74/ANI4	C4	P76/ANI6	E4	P11/ANO1
A5	P78/ANI8	C5	P710/ANI10	E5	EV_{SS}
A6	EV_{SS}	C6	PDH0/A16	E6	EV_{SS}
A7	PDL11/AD11	C7	PDL13/AD13	E7	EV_{SS}
A8	PDL8/AD8	C8	PDL10/AD10	E8	PCT0/$\overline{WR0}$
A9	PDL6/AD6	C9	PDL2/AD2	E9	PCM3/$\overline{HLDRQ}$
A10	PDL5/AD5/FLMD1	C10	PDL1/AD1	E10	PCM2/$\overline{HLDAK}$
A11	EV_{DD}	C11	PDL0/AD0	E11	EV_{SS}
B1	AV_{REF0}	D1	V_{DD}	F1	X1
B2	AV_{REF1}	D2	RV_{DD}	F2	X2
B3	P71/ANI1	D3	P73/ANI3	F3	FLMD0[注2]
B4	P75/ANI5	D4	P77/ANI7	F4	PDH4/A20
B5	P79/ANI9	D5	P711/ANI11	F5	EV_{SS}
B6	PDL15/AD15	D6	PDH1/A17	F6	EV_{SS}
B7	PDL12/AD12	D7	PDL14/AD14	F7	EV_{SS}
B8	PDL9/AD9	D8	PCT6/ASTB	F8	PDH3/A19
B9	PDL7/AD7	D9	PCT4/$\overline{RD}$	F9	PDH2/A18
B10	PDL4/AD4	D10	PCT1/$\overline{WR1}$	F10	PCM1/CLKOUT
B11	PDL3/AD3	D11	EV_{DD}	F11	PCM0/$\overline{WAIT}$

注　**1.** 使用最短的模式连接 E1 和 E2 引脚，并通过一个 4.7 *u*F（推荐值）电容将他们连接至 V_{SS} 。

2. FLMD0 引脚在 flash 编程模式下使用。正常操作模式下，将该引脚连接至 V_{SS}。

(2/2)

引脚编号	引脚名称	引脚编号	引脚名称	引脚编号	引脚名称
G1	V_{SS}	H9	P911/A11/SOB3	K6	UV_{DD}
G2	V_{SS}	H10	P910/A10/SIB3	K7	P51/TIQ02/KR1/TOQ02/RTP01
G3	P03/INTP0/ADTRG/UCLK/ RTC1HZ	H11	P99/A9/$\overline{SCKB1}$	K8	P54/SOB2/KR4/RTP04/DCK
G4	PDH5/NMI/A21	J1	V_{SS}	K9	P92/A2/TIP41/TOP41/TXDA4
G5	EV_{SS}	J2	IC [注]	K10	P95/A5/TIP30/TOP30/RXDA5
G6	EV_{SS}	J3	P05/INTP2/$\overline{DRST}$	K11	P96/A6/TXDC0/TIP21/TOP21
G7	EV_{SS}	J4	P06/INTP3	L1	EV_{SS}
G8	P915/A15/INTP6/TIP50/TOP50	J5	EV_{SS}	L2	P42/$\overline{SCKB0}$
G9	P914/A14/INTP5/TIP51/TOP51	J6	P37/RXDA3	L3	P30/TXDA0/SOB4
G10	P913/A13/INTP4	J7	P52/TIQ03/KR2/TOQ03 /RTP02/DDI	L4	P32/ASCKA0/$\overline{SCKB4}$/TIP00 /TOP00
G11	P912/A12/$\overline{SCKB3}$	J8	P55/$\overline{SCKB2}$/KR5/RTP05/DMS	L5	EV_{SS}
H1	XT1	J9	P93/A3/TIP40/TOP40/RXDA4	L6	EV_{DD}
H2	XT2	J10	P98/A8/SOB1	L7	P50/TIQ01/KR0/TOQ01/RTP00
H3	$\overline{RESET}$	J11	P97/A7/SIB1/RXDC0/TIP20/ TOP20	L8	P53/SIB2/KR3/TIQ00/TOQ00 /RTP03/DDO
H4	P04/INTP1/RTCDIV/RTCCL	K1	P40/SIB0/SDA01	L9	P91/A1/KR7/RXDA1/SCL02
H5	P36/TXDA3	K2	P41/SOB0/SCL01	L10	P94/A4/TIP31/TOP31/TXDA5
H6	P38/TXDA2/SDA00	K3	P31/RXDA0/INTP7/SIB4	L11	EV_{DD}
H7	P39/RXDA2/SCL00	K4	UDMF	–	–
H8	P90/A0/KR6/TXDA1/SDA02	K5	UDPF	–	–

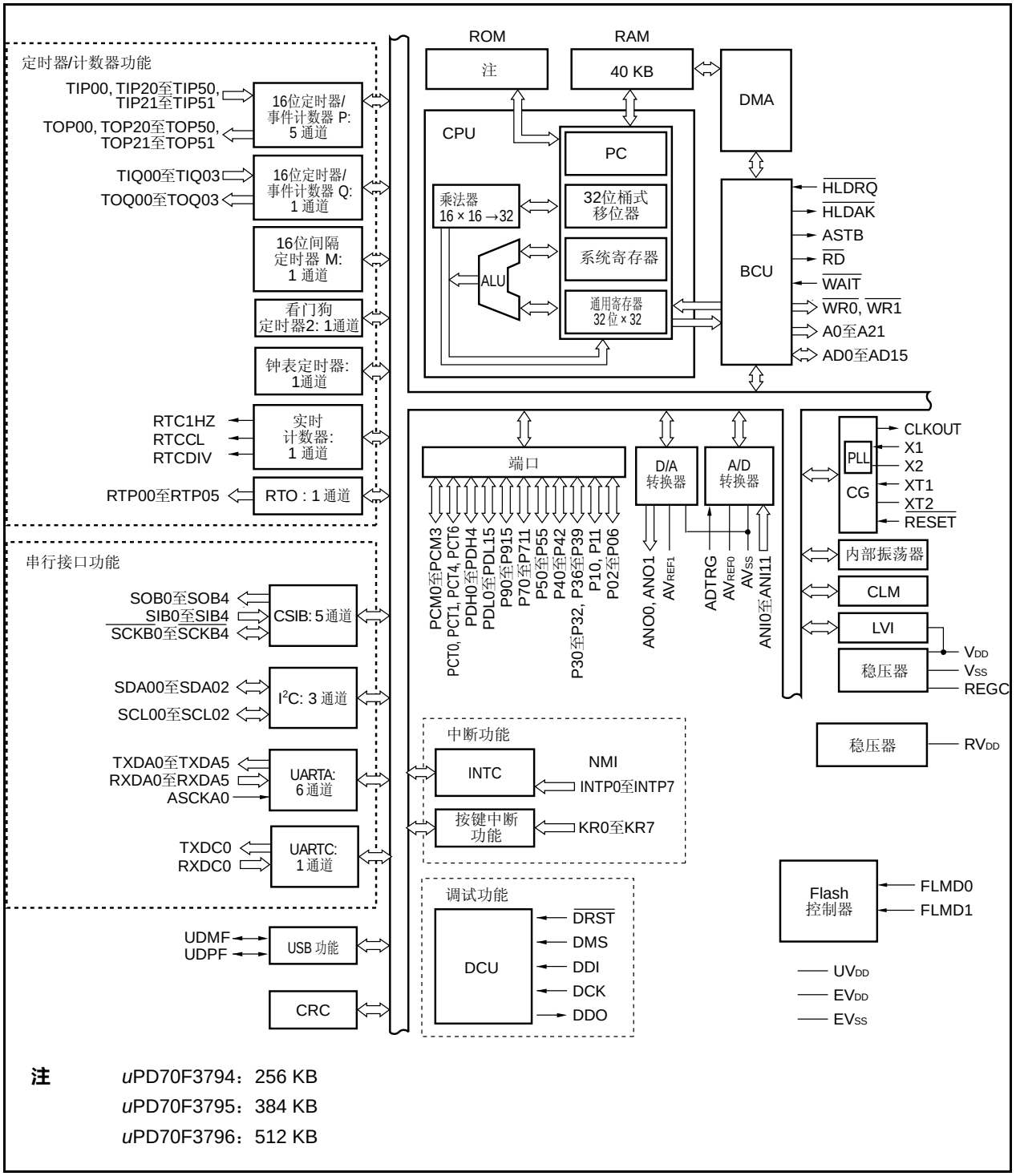
注 请确保开路。

引脚名称

A0 至 A21：	地址总线
AD0 至 AD15：	地址/数据总线
ADTRG：	A/D 触发信号输入
ANI0 至 ANI11：	模拟输入
ANO0，ANO1：	模拟输出
ASCKA0：	异步串行时钟
ASTB：	地址选通
AV_{REF0}， AV_{REF1}：	模拟参考电压
AV_{SS}：	模拟 V_{SS}
CLKOUT：	时钟输出
DCK：	调试时钟
DDI：	调试数据输入
DDO：	调试数据输出
DMS：	调试模式选择
$\overline{DRST}$：	调试复位
EV_{DD}：	外部引脚的电源
EV_{SS}：	外部引脚接地
FLMD0， FLMD1：	Flash 编程模式
$\overline{HLDAK}$：	保持应答
$\overline{HLDRQ}$：	保持请求
IC：	内部已连接
INTP0 至 INTP7：	外部中断输入
KR0 至 KR7：	按键返回
NMI：	不可屏蔽中断请求
P02 至 P06：	端口 0
P10，P11：	端口 1
P30 至 P32：	端口 3
P36 至 P39	
P40 至 P42：	端口 4
P50 至 P55：	端口 5
P70 至 P711：	端口 7
P90 至 P915：	端口 9
PCM0 至 PCM3：	端口 CM
PCT0，PCT1,	
PCT4，PCT6：	端口 CT
PDH0 至 PDH4：	端口 DH
PDL0 至 PDL15：	端口 DL
$\overline{RD}$：	读选通
REGC：	稳压器控制
$\overline{RESET}$：	系统复位
RTC1HZ,	实时计数器时钟输出
RTCCL，RTCDIV	
RTP00 至 RTP05:	实时输出端口
RV_{DD}	RTC 的电源供应
RXDA0 至 RXDA5：	接收数据
RXDC0：	
$\overline{SCKB0}$ 至 $\overline{SCKB4}$：	串行时钟
SCL00 至 SCL02：	串行时钟
SDA00 至 SDA02：	串行数据
SIB0 至 SIB4：	串行输入
SOB0 至 SOB4：	串行输出
TIP00，	定时器输入
TIP20，TIP21，	
TIP30，TIP31，	
TIP40，TIP41，	
TIP50，TIP51,	
TIQ00 至 TIQ03：	
TOP00，	定时器输出
TOP20，TOP21，	
TOP30，TOP31，	
TOP40，TOP41，	
TOP50，TOP51,	
TOQ00 至 TOQ03：	
TXDA0 至 TXDA5:	发送数据
TXDC0:	
UCLK:	USB 时钟
UDMF:	USB 功能数据 I/O (–)
UDPF:	USB 功能数据 I/O (+)
UV_{DD}:	外部 USB 的电源供应
V_{DD}:	电源
V_{SS}:	地
$\overline{WAIT}$:	等待
$\overline{WR0}$:	低字节写选通
$\overline{WR1}$:	高字节写选通
X1，X2:	主时钟晶振
XT1，XT2	副时钟晶振

1.6 功能框图结构

1.6.1 内部框图



注 *u*PD70F3794：256 KB
*u*PD70F3795：384 KB
*u*PD70F3796：512 KB

1.6.2 内部单元

（1）CPU

CPU 使用 5 级流水线控制，单时钟周期内可以完成地址运算、算术逻辑运算、数据传输以及几乎所有其它指令的处理。

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

（2）总线控制单元（BCU）

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

（3）Flash 存储器（ROM）

这是一个 512/384/256 KB flash 存储器，其映射地址为 0000000H 至 007FFFFH/ 0000000H 至 005FFFFH/0000000H 至 003FFFFH。

取指令期间，CPU 可在一个时钟周期内访问 Flash 存储器（ROM）。

（4）RAM

这是一个 40 KB RAM，其映射地址为 3FF5000H 至 3FFEFFFH。CPU 可在一个时钟周期内完成 RAM 数据存取访问。

（5）中断控制器（INTC）

该控制器用来处理片上外设和外部硬件产生的硬件中断请求（NMI、INTP0 至 INTP7）。可以为这些中断请求指定 8 个级别的中断优先级，同时也能执行多重服务控制。

（6）时钟发生器 （CG）

提供主时钟振荡器和副时钟振荡器，并且产生相应的主时钟振荡频率（f_X）和副时钟振荡频率（f_{XT}）。有两种模式：

在时钟直通模式中，fx 直接用作主时钟频率（fxx）。在 PLL 模式下，使用 fx 的 4 倍频。

CPU 工作频率（f_{CPU}）可以有以下选择：f_{XX}，$f_{XX}/2$， $f_{XX}/4$，$f_{XX}/8$，$f_{XX}/16$， $f_{XX}/32$，和 f_{XT}。

（7）内部振荡器

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

（8）定时器/计数器

芯片提供了 6 个通道的 16 位定时器/事件计数器 P（TMP），一个通道的 16 位定时器/事件计数器 Q（TMQ）和一个通道的 16 位间隔定时器 M（TMM）。

（9）钟表定时器

该定时器通过对参考时间周期（0.5 s）的计数来计算时钟（32.768 kHz 的副时钟或来自预分频器的 32.768 kHz f_{BRG} 时钟）。钟表定时器也能够用作基于主时钟的间隔定时器。

（10）实时计数器（钟表）

基于副时钟（32.768 kHz）或主时钟，实时定时器通过对参考时间（一秒）的计数，来计算钟表时钟。该定时器也能够用作基于主时钟的间隔定时器。提供专用于年、月、星期、日、时、分、秒的硬件计数器，最大可以到 99 年。

（11）看门狗定时器 2

片内集成的看门狗定时器用于检测无意中形成的程序死循环、系统异常等。

内部振荡时钟、主时钟、或副时钟都可以被选为源时钟。

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

（12）串行接口

V850ES/JG3-L 包含三种串行接口：异步串行接口 A（UARTA）、3 线长度可变串行接口 B（CSIB）、I^2C 总线接口（I^2C）和 USB 功能控制器（USBF）。

UARTA 时，数据经由 TXDA0 引脚至 TXDA2 引脚和 RXDA0 引脚至 RXDA2 引脚进行传输

CSIB 时，数据经由 SOB0 引脚至 SOB4 引脚、SIB0 引脚至 SIB4 引脚、和 $\overline{\text{SCKB0}}$ 引脚至 $\overline{\text{SCKB4}}$ 引脚进行传输。

I^2C 时，数据经由 SDA00 引脚至 SDA02 引脚和 SCL00 引脚至 SCL02 引脚进行传输。

USBF 时，数据通过 UDMF 和 UDPF 引脚传输。

（13）A/D 转换器

10 位 A/D 转换器包含 12 个模拟输入引脚。使用逐次逼近的方式进行转换。

（14）D/A 转换器

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

（15）DMA 控制器

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

（16）按键中断功能

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

（17）实时输出功能

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

（18）CRC 功能

芯片提供 CRC 操作电路，该电路对 8 位数据的组产生 16 位 CRC（循环冗余检测）码。

（19）DCU（调试控制单元）

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

（20）端口

可使用以下通用寄存器端口功能和控制引脚功能：

表 1-2. 端口功能

端口	I/O	复用功能
P0	5 位 I/O	NMI、外部中断、A/D 转换器触发、调试复位、实时计数器输出
P1	2 位 I/O	D/A 转换器模拟输出
P3	7 位 I/O	外部中断、串行接口、定时器 I/O
P4	3 位 I/O	串行接口
P5	6 位 I/O	定时器 I/O、实时输出、按键中断输入、串行接口、调试 I/O
P7	12 位 I/O	A/D 转换器模拟输入
P9	16 位 I/O	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	5 位 I/O	外部地址总线
PDL	16 位 I/O	外部地址/数据总线

第二章 引脚功能

2.1 引脚功能列表

V850ES/JG3-L 中引脚的功能描述如下。

有 4 种类型的引脚 I/O 缓冲器电源： AV_{REF0}、AV_{REF1}、EV_{DD} 和 UV_{DD} 。这些电源和引脚的关系如下：

表 2-1. 引脚 I/O 缓冲器电源

电源	对应引脚
AV_{REF0}	端口 7
AV_{REF1}	端口 1
EV_{DD}	$\overline{RESET}$，端口 0、3 至 5、9， CM，CT，DH，DL，FLMD0
UV_{DD}	UDMF, UDPF

(1) 端口功能

（1/3）

功能	引脚编号		I/O	描述	复用功能
	GC	F1			
P02	7	G4	I/O	端口0（参见**4.3.1**） 5位I/O端口 可以按位指定为输入/输出。 可以按位指定为N沟道漏极开路输出。 5V 耐压。	NMI/A21
P03	18	G3			INTP0/ADTRG/UCLK/RTC1HZ
P04	19	H4			INTP1/RTCDIV/RTCCL
P05[注]	20	J3			INTP2/$\overline{\text{DRST}}$
P06	21	J4			INTP3
P10	3	E3	I/O	端口1（参见**4.3.2**） 2位I/O端口 可以按位指定为输入/输出。	ANO0
P11	4	E4			ANO1
P30	25	L3	I/O	端口3（参见**4.3.3**） 7位I/O端口 可以按位指定为输入/输出。 可以按位指定为N沟道漏极开路输出。 5V 耐压。	TXDA0/SOB4
P31	26	K3			RXDA0/INTP7/SIB4
P32	27	L4			ASCKA0/$\overline{\text{SCKB4}}$/TIP00/TOP00
P36	31	H5			TXDA3
P37	32	J6			RXDA3
P38	35	H6			TXDA2/SDA00
P39	36	H7			RXDA2/SCL00
P40	22	K1	I/O	端口4（参见**4.3.4**） 3位I/O端口 可以按位指定为输入/输出。 可以按位指定为N沟道漏极开路输出。 5V 耐压。	SIB0/SDA01
P41	23	K2			SOB0/SCL01
P42	24	L2			$\overline{\text{SCKB0}}$
P50	37	L7	I/O	端口5（参见**4.3.5**） 6位I/O端口 可以按位指定为输入/输出。 可以按位指定为N沟道漏极开路输出。 5V 耐压。	TIQ01/KR0/TOQ01/RTP00
P51	38	K7			TIQ02/KR1/TOQ02/RTP01
P52	39	J7			TIQ03/KR2/TOQ03/RTP02/DDI
P53	40	L8			SIB2/KR3/TIQ00/TOQ00/RTP03/DDO
P54	41	K8			SOB2/KR4/RTP04/DCK
P55	42	J8			$\overline{\text{SCKB2}}$/KR5/RTP05/DMS

注 内置有一个下拉电阻。将 OCDM.OCDM0 位清除为 0，可以断开该电阻的连接。

备注 GC：100 针塑封 LQFP（密间距）（14 × 14）
F1：121 针塑封 FBGA（8 × 8）

（2/3）

功能	引脚编号		I/O	描述	复用功能
	GC	F1			
P70	100	A3	I/O	端口7（参见**4.3.6**） 12位I/O端口 可以按位指定为输入/输出。	ANI0
P71	99	B3			ANI1
P72	98	C3			ANI2
P73	97	D3			ANI3
P74	96	A4			ANI4
P75	95	B4			ANI5
P76	94	C4			ANI6
P77	93	D4			ANI7
P78	92	A5			ANI8
P79	91	B5			ANI9
P710	90	C5			ANI10
P711	89	D5			ANI11
P90	43	H8	I/O	端口9（参见**4.3.7**） 16位I/O端口 可以按位指定为输入/输出。 可以按位指定为N沟道漏极开路输出。 5V 耐压。（P90至P96）	A0/KR6/TXDA1/SDA02
P91	44	L9			A1/KR7/RXDA1/SCL02
P92	45	K9			A2/TIP41/TOP41/TXDA4
P93	46	J9			A3/TIP40/TOP40/RXDA4
P94	47	L10			A4/TIP31/TOP31/TXDA5
P95	48	K10			A5/TIP30/TOP30/RXDA5
P96	49	K11			A6/TXDC0/TIP21/TOP21
P97	50	J11			A7/SIB1/RXDC0 /TIP20/TOP20
P98	51	J10			A8/SOB1
P99	52	H11			A9/$\overline{\text{SCKB1}}$
P910	53	H10			A10/SIB3
P911	54	H9			A11/SOB3
P912	55	G11			A12/$\overline{\text{SCKB3}}$
P913	56	G10			A13/INTP4
P914	57	G9			A14/INTP5/TIP51/TOP51
P915	58	G8			A15/INTP6/TIP50/TOP50

备注 GC：100 针塑封 LQFP（密间距）（14 × 14）
F1：121 针塑封 FBGA（8 × 8）

（3/3）

功能	引脚编号		I/O	描述	复用功能
	GC	F1			
PCM0	61	F11	I/O	端口CM（参见**4.3.8**） 4位I/O端口 可以按位指定为输入/输出。	$\overline{\text{WAIT}}$
PCM1	62	F10			CLKOUT
PCM2	63	E10			$\overline{\text{HLDAK}}$
PCM3	64	E9			$\overline{\text{HLDRQ}}$
PCT0	65	E8	I/O	端口CT（参见**4.3.9**） 4位I/O端口 可以按位指定为输入/输出。	$\overline{\text{WR0}}$
PCT1	66	D10			$\overline{\text{WR1}}$
PCT4	67	D9			$\overline{\text{RD}}$
PCT6	68	D8			ASTB
PDH0	87	C6	I/O	端口DH（参见**4.3.10**） 5位I/O端口 可以按位指定为输入/输出。	A16
PDH1	88	D6			A17
PDH2	59	F9			A18
PDH3	60	F8			A19
PDH4	6	F4			A20
PDL0	71	C11	I/O	端口DL（参见**4.3.11**） 16位I/O端口 可以按位指定为输入/输出。	AD0
PDL1	72	C10			AD1
PDL2	73	C9			AD2
PDL3	74	B11			AD3
PDL4	75	B10			AD4
PDL5	76	A10			AD5/FLMD1
PDL6	77	A9			AD6
PDL7	78	B9			AD7
PDL8	79	A8			AD8
PDL9	80	B8			AD9
PDL10	81	C8			AD10
PDL11	82	A7			AD11
PDL12	83	B7			AD12
PDL13	84	C7			AD13
PDL14	85	D7			AD14
PDL15	86	B6			AD15

备注 GC：100 针塑封 LQFP（密间距）（14 × 14）
F1：121 针塑封 FBGA（8 × 8）

（2）非端口功能

（1/6）

功能	引脚编号		I/O	描述	复用功能
	GC	F1			
A0	43	H8	输出	外部存储器地址总线 （使用分离总线时） 可以按位指定为 N 沟道漏极开路输出。 5V 耐压。	P90/KR6/TXDA1/SDA02
A1	44	L9			P91/KR7/RXDA1/SCL02
A2	45	K9			P92/TIP41/TOP41/TXDA4
A3	46	J9			P93/TIP40/TOP40/RXDA4
A4	47	L10			P94/TIP31/TOP31/TXDA5
A5	48	K10			P95/TIP30/TOP30/RXDA5
A6	49	K11			P96 /TXDC0 /TIP21/TOP21
A7	50	J11			P97/SIB1/RXDC0 /TIP20/TOP20
A8	51	J10			P98/SOB1
A9	52	H11			P99/$\overline{\text{SCKB}}$1
A10	53	H10			P910/SIB3
A11	54	H9			P911/SOB3
A12	55	G11			P912/$\overline{\text{SCKB}}$3
A13	56	G10			P913/INTP4
A14	57	G9			P914/INTP5/TIP51/TOP51
A15	58	G8			P915/INTP6/TIP50/TOP50
A16	87	C6	输出	外部存储器地址总线	PDH0
A17	88	D6			PDH1
A18	59	F9			PDH2
A19	60	F8			PDH3
A20	6	F4			PDH4
A21	7	G4			P02/NMI
AD0	71	C11	I/O	外部存储器的地址总线/数据总线	PDL0
AD1	72	C10			PDL1
AD2	73	C9			PDL2
AD3	74	B11			PDL3
AD4	75	B10			PDL4
AD5	76	A10			PDL5/FLMD1
AD6	77	A9			PDL6
AD7	78	B9			PDL7
AD8	79	A8			PDL8
AD9	80	B8			PDL9
AD10	81	C8			PDL10
AD11	82	A7			PDL11
AD12	83	B7			PDL12
AD13	84	C7			PDL13
AD14	85	D7			PDL14
AD15	86	B6			PDL15

备注 GC：100 针塑封 LQFP（密间距）（14 × 14）
F1：121 针塑封 FBGA（8 × 8）

（2/6）

功能	引脚编号		I/O	描述	复用功能
	GC	F1			
ADTRG	18	G3	输入	A/D 转换器外部触发输入。5V 耐压。	P03/INTP0/UCLK/RTC1HZ
ANI0	100	A3	输入	A/D 转换器模拟电压输入	P70
ANI1	99	B3			P71
ANI2	98	C3			P72
ANI3	97	D3			P73
ANI4	96	A4			P74
ANI5	95	B4			P75
ANI6	94	C4			P76
ANI7	93	D4			P77
ANI8	92	A5			P78
ANI9	91	B5			P79
ANI10	90	C5			P710
ANI11	89	D5			P711
ANO0	3	E3	输出	D/A 转换器模拟电压输出	P10
ANO1	4	E4			P11
ASCKA0	27	L4	输入	UARTA0 波特率时钟输入。5 V 耐压。	P32/$\overline{\text{SCKB4}}$/TIP00/TOP00
ASTB	68	D8	输出	外部存储器的地址选通信号输出	PCT6
AV_{REF0}	1	A1, A2, B1	–	用于 A/D 转换器的参考电压输入/用于端口 7 的正极电源供应	–
AV_{REF1}	5	B2		用于 D/A 转换器的参考电压输入/用于端口 1 的正极电源供应	–
AV_{SS}	2	C1, C2	–	用于 A/D 转换器和 D/A 转换器的地电位（与 V_{SS} 电位相同）	–
CLKOUT	62	F10	输出	内部系统时钟输出	PCM1
DCK	41	K8	输入	调试时钟输入。5V 耐压。	P54/SOB2/KR4/RTP04
DDI	39	J7	输入	调试数据输入。5V 耐压。	P52/TIQ03/KR2/TOQ03/RTP02
DDO[注1]	40	L8	输出	调试数据输出。 N 沟道漏极开路输出选择。5 V 耐压。	P53/SIB2/KR3/TIQ00/TOQ00/ RTP03
DMS	42	J8	输入	调试模式选择输入。5 V 耐压。	P55/$\overline{\text{SCKB2}}$/KR5/RTP05
$\overline{\text{DRST}}$	20	J3	输入	调试复位输入。5 V 耐压。	P05/INTP2
EV_{DD}	34, 70	注2	–	用于外部的正极电源供应（与 V_{DD} 电位相同）	–
EV_{SS}	33, 69	注3	–	用于外部的地电位（与 V_{SS} 电位相同）	–
FLMD0	8	F3	输入	Flash 存储器编程模式设置引脚	–
FLMD1	76	A10			PDL5/AD5
$\overline{\text{HLDAK}}$	63	E10	输出	总线保持应答输出	PCM2
$\overline{\text{HLDRQ}}$	64	E9	输入	总线保持请求输入	PCM3

注 **1.** 在片上调试时，被强制设置为高电平输出。
2. A11, D11, K6, L6, L11
3. A6, E5 至 E7, E11, F5 至 F7, G5 至 G7, L1, L5

备注 GC：100 针塑封 LQFP（密间距）（14 × 14）
F1：121 针塑封 FBGA（8 × 8）

（3/6）

功能	引脚编号		I/O	描述	复用功能
	GC	F1			
IC	–	J2	–	内部已连接	–
INTP0	18	G3	输入	外部中断请求输入 （可屏蔽，模拟噪声消除）。 可以为 INTP3 引脚选择模拟噪声消除或数字噪声消除。 5 V 耐压。	P03/ADTRG/UCLK/RTC1HZ
INTP1	19	H4			P04/RTCDIV/RTCCL
INTP2	20	J3			P05/$\overline{\text{DRST}}$
INTP3	21	J4			P06
INTP4	56	G10			P913/A13
INTP5	57	G9			P914/A14/TIP51/TOP51
INTP6	58	G8			P915/A15/TIP50/TOP50
INTP7	26	K3			P31/RXDA0/SIB4
KR0[注1]	37	L7	输入	按键中断请求（片上模拟噪声消除器） 5 V 耐压。	P50/TIQ01/TOQ01/RTP00
KR1[注1]	38	K7			P51/TIQ02/TOQ02/RTP01
KR2[注1]	39	J7			P52/TIQ03/TOQ03/RTP02/DDI
KR3[注1]	40	L8			P53/SIB2/TIQ00/TOQ00/ RTP03/DDO
KR4[注1]	41	K8			P54/SOB2/RTP04/DCK
KR5[注1]	42	J8			P55/$\overline{\text{SCKB2}}$/RTP05/DMS
KR6[注1]	43	H8			P90/A0/TXDA1/SDA02
KR7[注1]	44	L9			P91/A1/RXDA1/SCL02
NMI[注2]	17	G4	输入	外部中断输入（不可屏蔽，模拟噪声消除）。 5 V 耐压。	P02/A21
$\overline{\text{RD}}$	67	D9	输出	外部存储器“读”选通信号输出	PCT4
REGC	10	E1,E2	–	稳压器输出稳定电容的连接（4.7 *u*F（推荐值））	–
$\overline{\text{RESET}}$	14	H3	输入	系统复位输入	–
RTC1HZ	18	G3	输出	实时计数器修正时钟（1 Hz）输出	P03/INTP0/ADTRG/UCLK
RTCCL	19	H4	输出	实时计数器时钟（32 kHz 原始振荡）输出	P04/INTP1/RTCDIV
RTCDIV	19	H4	输出	实时计数器时钟（32 kHz 分频）输出	P04/INTP1/RTCCL
RTP00	37	L7	输出	实时输出端口。 可以选择 N 沟道漏极开路输出。 5 V 耐压。	P50/TIQ01/KR0/TOQ01
RTP01	38	K7			P51/TIQ02/KR1/TOQ02
RTP02	39	J7			P52/TIQ03/KR2/TOQ03/DDI
RTP03	40	L8			P53/SIB2/KR3/TIQ00/TOQ00/DDO
RTP04	41	K8			P54/SOB2/KR4/DCK
RTP05	42	J8			P55/$\overline{\text{SCKB2}}$/KR5/DMS

注 **1.** 外部连接上拉电阻。

2. NMI 引脚复用功能是 P02 引脚。在复位之后作为 P02 引脚功能。要使能 NMI 引脚功能，将 PMC0.PMC02 位设置为 1。NMI 引脚的初始设置是“无边沿检测”。通过 INTF0 和 INTR0 寄存器来选择 NMI 引脚有效边沿。

备注 GC：100 针塑封 LQFP（密间距）（14 × 14）

F1：121 针塑封 FBGA（8 × 8）

（4/6）

功能	引脚编号		I/O	描述	复用功能
	GC	F1			
RXDA0	26	K3	输入	串行接收数据输入（UARTA0至UARTA2） 5 V耐压。	P31/INTP7/SIB4
RXDA1	44	L9			P91/A1/KR7/SCL02
RXDA2	36	H7			P39/SCL00
RXDA3	32	J6			P37
RXDA4	46	J9			P93/A3/TIP40/TOP40
RXDA5	48	K10			P95/A5/TIP30/TOP30
RXDC0	50	J11		串行接收数据输入（UARTC0）	P97/A7/SIB1/TIP20/TOP20
RV_{DD}	17	D2	–	RTC的正极电源供给	–
$\overline{SCKB0}$	24	L2	I/O	串行时钟I/O（CSIB0至CSIB4） 可以选择N沟道漏极开路输出。 5V 耐压。	P42
$\overline{SCKB1}$	52	H11			P99/A9
$\overline{SCKB2}$	42	J8			P55/KR5/RTP05/DMS
$\overline{SCKB3}$	55	G11			P912/A12
$\overline{SCKB4}$	27	L4			P32/ASCKA0/TIP00/TOP00
SCL00	36	H7	I/O	串行时钟I/O（I^2C00至I^2C02） 可以选择N沟道漏极开路输出。 5V 耐压。	P39/RXDA2
SCL01	23	K2			P41/SOB0
SCL02	44	L9			P91/A1/KR7/RXDA1
SDA00	35	H6	I/O	串行发送/接收数据 I/O（I^2C00至I^2C02） 可以选择N沟道漏极开路输出。 5V 耐压。	P38/TXDA2
SDA01	22	K1			P40/SIB0
SDA02	43	H8			P90/A0/KR6/TXDA1
SIB0	22	K1	输入	串行接收数据输入（CSIB0至CSIB4） 5V 耐压。	P40/SDA01
SIB1	50	J11			P97/A7/RXDC0 /TIP20/TOP20
SIB2	40	L8			P53/KR3/TIQ00/TOQ00/RTP03/DDO
SIB3	53	H10			P910/A10
SIB4	26	K3			P31/RXDA0/INTP7
SOB0	23	K2	输出	串行发送数据输出（CSIB0至CSIB4） 可以选择N沟道漏极开路输出。 5V 耐压。	P41/SCL01
SOB1	51	J10			P98/A8
SOB2	41	K8			P54/KR4/RTP04/DCK
SOB3	54	H9			P911/A11
SOB4	25	L3			P30/TXDA0

备注 GC：100 针塑封 LQFP（密间距）（14 × 14）
F1：121 针塑封 FBGA（8 × 8）

（5/6）

功能	引脚编号		I/O	描述	复用功能
	GC	F1			
TIP00	27	L4	输入	外部事件计数输入/捕获触发输入/外部触发输入（TMP0）. 5V 耐压。	P32/ASCKA0/$\overline{\text{SCKB4}}$/TOP00
TIP20	50	J11		外部事件计数输入/捕获触发输入/外部触发输入（TMP2）. 5V 耐压。	P97/A7/SIB1/TOP20
TIP21	49	K11		捕获触发输入（TMP2）. 5V 耐压。	P96/A6/TOP21
TIP30	48	K10		外部事件计数输入/捕获触发输入/外部触发输入（TMP3）. 5V 耐压。	P95/A5/TOP30
TIP31	47	L10		捕获触发输入（TMP3）. 5V 耐压。	P94/A4/TOP31
TIP40	46	J9		外部事件计数输入/捕获触发输入/外部触发输入（TMP4）. 5V 耐压。	P93/A3/TOP40
TIP41	45	K9		捕获触发输入（TMP4）. 5V 耐压。	P92/A2/TOP41
TIP50	58	G8		外部事件计数输入/捕获触发输入/外部触发输入（TMP5）. 5V 耐压。	P915/A15/INTP6/TOP50
TIP51	57	G9		捕获触发输入（TMP5）. 5V 耐压。	P914/A14/INTP5/TOP51
TIQ00	40	L8	输入	外部事件计数输入/捕获触发输入/外部触发输入（TMQ0）. 5V 耐压。	P53/SIB2/KR3/TOQ00/RTP03 /DDO
TIQ01	37	L7		捕获触发输入（TMQ0）. 5V 耐压。	P50/KR0/TOQ01/RTP00
TIQ02	38	K7			P51/KR1/TOQ02/RTP01
TIQ03	39	J7			P52/KR2/TOQ03/RTP02/DDI
TOP00	27	L4	输出	定时器输出（TMP0） 可以选择N沟道漏极开路输出。5V 耐压。	P32/ASCKA0/$\overline{\text{SCKB4}}$/TIP00
TOP20	50	J11		定时器输出（TMP2） 可以选择N沟道漏极开路输出。5V 耐压。	P97/A7/SIB1/TIP20
TOP21	49	K11			P96/A6/TIP21
TOP30	48	K10		定时器输出（TMP3） 可以选择N沟道漏极开路输出。5V 耐压。	P95/A5/TIP30
TOP31	47	L10			P94/A4/TIP31
TOP40	46	J9		定时器输出（TMP4） 可以选择N沟道漏极开路输出。5V 耐压。	P93/A3/TIP40
TOP41	45	K9			P92/A2/TIP41
TOP50	58	G8		定时器输出（TMP5） 可以选择N沟道漏极开路输出。5V 耐压。	P915/A15/INTP6/TIP50
TOP51	57	G9			P914/A14/INTP5/TIP51

备注 GC：100 针塑封 LQFP（密间距）（14 × 14）
F1：121 针塑封 FBGA（8 × 8）

（6/6）

功能	引脚编号		I/O	描述	复用功能
	GC	F1			
TOQ00	40	L8	输出	定时器输出（TMQ0） 可以选择N沟道漏极开路输出。 5V 耐压。	P53/SIB2/KR3/TIQ00/RTP03/DDO
TOQ01	37	L7			P50/TIQ01/KR0/RTP00
TOQ02	38	K7			P51/TIQ02/KR1/RTP01
TOQ03	39	J7			P52/TIQ03/KR2/RTP02/DDI
TXDA0	25	L3	输出	串行发送数据输出（UARTA0至UARTA5） 可以选择N沟道漏极开路输出。 5V 耐压。	P30/SOB4
TXDA1	43	H8			P90/A0/KR6/SDA02
TXDA2	35	H6			P38/SDA00
TXDA3	31	H5			P36
TXDA4	45	K9			P92/A2/TIP41/TOP41
TXDA5	47	L10			P94/A4/TIP31/TOP31
TXDC0	49	K11		串行发送数据输出（UARTAC） 可以选择N沟道漏极开路输出。 5V 耐压。	P96/A6/TIP21/TOP21
UCLK	18	G3	输入	USB时钟信号输入	P03/INTP0/ADTRG/UCLK/RTC1HZ
UDMF	28	K4	I/O	USB数据 I/O（–）功能	–
UDPF	29	K5	I/O	USB数据 I/O（+）功能	–
UV_{DD}	30	K6	–	USB的电源供给	–
V_{DD}	9	D1	–	内部电路的正极电源引脚	–
V_{SS}	11	G1, G2, J1	–	内部电路地电平	–
$\overline{WAIT}$	61	F11	输入	外部“等待”输入	PCM0
$\overline{WR0}$	65	E8	输出	外部存储器的写选通（低 8 位）	PCT0
$\overline{WR1}$	66	D10		外部存储器的写选通（高 8 位）	PCT1
X1	12	F1	输入	用于连接主时钟振荡电路	–
X2	13	F2	–		–
XT1	15	H1	输入	用于连接副时钟振荡电路	–
XT2	16	H2	–		–

备注 GC：100 针塑封 LQFP（密间距）（14 × 14）
F1：121 针塑封 FBGA（8 × 8）

2.2 引脚状态

各种模式下的引脚工作状态描述如下：

表 2-2. 各种模式下的引脚工作状态

引脚名称	接通电源时[注1]	复位期间（接通电源除外）	HALT 模式[注2]	IDLE1、IDLE2、Sub-IDLE 模式[注2]	STOP 模式[注2]	空闲状态[注3]	总线保持	RTC 备份模式
P05/$\overline{\text{DRST}}$	下拉	下拉[注4]	保持	保持	保持	保持	保持	不确定[注9]
P10/ANO0, P11/ANO1	不确定	Hi-Z	保持	保持	Hi-Z	保持	保持	
P53/DDO		Hi-Z[注5]	保持	保持	保持	保持	保持	
AD0 至 AD15	Hi-Z[注6]	Hi-Z[注6]	**注 7, 8**	Hi-Z	Hi-Z	保持	Hi-Z	
A0 至 A15			不确定[注7]					
A16 至 A21								
$\overline{\text{WAIT}}$			–	–	–	–	–	
CLKOUT			工作中	L	L	工作中	工作中	
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$			H[注7]	H	H	H	Hi-Z	
$\overline{\text{RD}}$								
ASTB								
$\overline{\text{HLDAK}}$			工作中				L	
$\overline{\text{HLDRQ}}$				–	–	–	工作中	
其它端口引脚	Hi-Z	Hi-Z	保持	保持	保持	保持	保持	

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

备注
Hi-Z：高阻
保持：保持上一个外部总线周期的状态。
L： 低电平输出
H： 高电平输出
–： 不对输入进行采样（不响应）

2.3 引脚I/O电路类型、I/O缓冲器电源、及未使用引脚的连接

（1/3）

引脚	复用功能	引脚编号		I/O 电路类型	未使用引脚的推荐连接
		GC	F1		
P02	NMI/A21	7	G4	10-D	输入： 通过电阻单独连接至 EV_{DD} 或 EV_{SS} 输出： 保持开路。
P03	INTP0/ADTRG/UCLK/RTC1HZ	18	G3		
P04	INTP1/RTCDIV/RTCCL	19	H4		
P05	INTP2/$\overline{DRST}$	20	J3	10-N	输入： 通过电阻单独连接至 EV_{SS}。禁止固定为 V_{DD} 电平。 输出： 保持开路。 在 $\overline{RESET}$ 引脚引发复位后，内部下拉。
P06	INTP3	21	J4	10-D	输入： 通过电阻单独连接至 EV_{DD} 或 EV_{SS} 输出： 保持开路。
P10	ANO0	3	E3	12-D	输入： 通过电阻单独连接至 AV_{REF1} 或 AV_{SS} 输出： 保持开路。
P11	ANO1	4	E4		
P30	TXDA0/SOB4	25	L3	10-G	输入： 通过电阻单独连接至 EV_{DD} 或 EV_{SS} 输出： 保持开路。
P31	RXDA0/INTP7/SIB4	26	K3	10-D	
P32	ASCKA0/$\overline{SCKB4}$/TIP00	27	L4		
P36	TXDA3	31	H5		
P37	RXDA3	32	J6		
P38	TXDA2/SDA00	35	H6		
P39	RXDA2/SCL00	36	H7		
P40	SIB0/SDA01	22	K1		
P41	SOB0/SCL01	23	K2		
P42	$\overline{SCKB0}$	24	L2		
P50	TIQ01/KR0/TOQ01/RTP00	37	L7	10-D	
P51	TIQ02/KR1/TOQ02/RTP01	38	K7		
P52	TIQ03/KR2/TOQ03/RTP02/DDI	39	J7		
P53	SIB2/KR3/TIQ00/TOQ00 /RTP03/DDO	40	L8		
P54	SOB2/KR4/RTP04/DCK	41	K8		
P55	$\overline{SCKB2}$/KR5/RTP05/DMS	42	J8		

备注 GC：100 针塑封 LQFP（密间距）（14 × 14）
F1：121 针塑封 FBGA（8 × 8）

（2/3）

引脚	复用功能	引脚编号		I/O 电路类型	未使用引脚的推荐连接
		GC	F1		
P70至P711	ANI0至ANI11	100至89	A3至A5, B3至B5, C3至C5, D3至D5	11-G	输入： 通过电阻单独连接至 AV_{REF0} 或 AV_{SS} 输出： 保持开路。
P90	A0/KR6/TDXA1/SDA02	43	H8	10-D	输入： 通过电阻单独连接至 EV_{DD} 或 EV_{SS} 输出： 保持开路。
P91	A1/KR7/RXDA1/SCL02	44	L9		
P92	A2/TIP41/TOP41/TXDA4	45	K9		
P93	A3/TIP40/TOP40/RXDA4	46	J9		
P94	A4/TIP31/TOP31/TXDA5	47	L10		
P95	A5/TIP30/TOP30/RXDA5	48	K10		
P96	A6/TXDC0/TIP21/TOP21	49	K11		
P97	A7/SIB1/RXDC0/TIP20/TOP20	50	J11		
P98	A8/SOB1	51	J10	10-G	
P99	A9/$\overline{\text{SCKB1}}$	52	H11	10-D	
P910	A10/SIB3	53	H10		
P911	A11/SOB3	54	H9	10-G	
P912	A12/$\overline{\text{SCKB3}}$	55	G11	10-D	
P913	A13/INTP4	56	G10		
P914	A14/INTP5/TIP51/TOP51	57	G9		
P915	A15/INTP6/TIP50/TOP50	58	G8		
PCM0	$\overline{\text{WAIT}}$	61	F11	5	
PCM1	CLKOUT	62	F10		
PCM2	$\overline{\text{HLDAK}}$	63	E10		
PCM3	$\overline{\text{HLDRQ}}$	64	E9		
PCT0, PCT1	$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	65, 66	E8, D10		
PCT4	$\overline{\text{RD}}$	67	D9		
PCT6	ASTB	68	D8		
PDH0 toPDH4	A16至A20	87, 88, 59, 60, 6	C6, D6, F9, F8, F4		
PDL0至PDL4	AD0至AD4	71至75	B10, B11, C9至C11		
PDL5	AD5/FLMD1	76	A10		

备注 GC：100 针塑封 LQFP（密间距）（14 × 14）
F1：121 针塑封 FBGA（8 × 8）

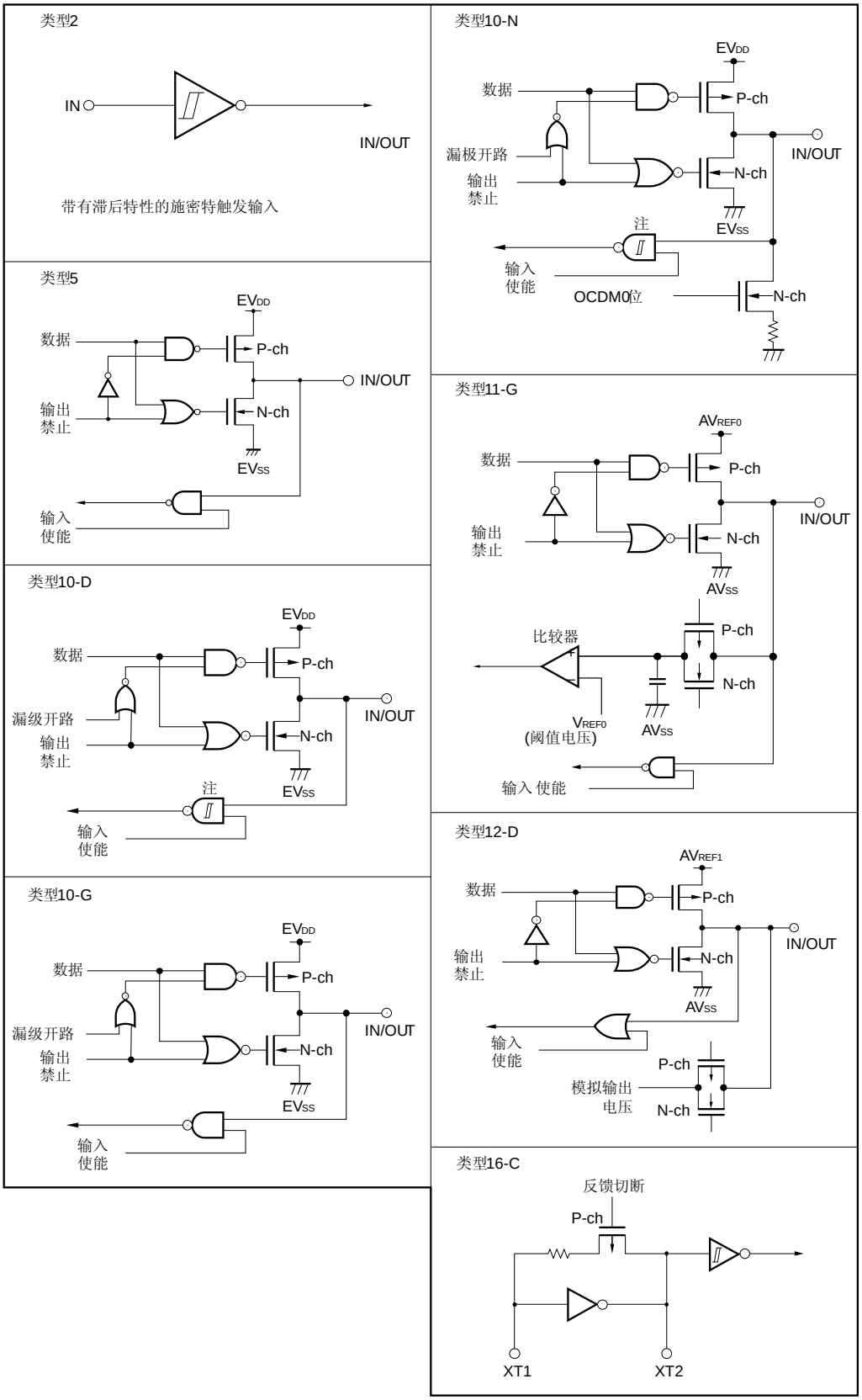
（3/3）

引脚	复用功能	引脚编号		I/O 电路类型	未使用引脚的推荐连接
		GC	F1		
PDL6至PDL15	AD6至AD15	77至86	A7至A9, B6至B9, C7, C8, D7	5	输入： 通过电阻单独连接至 EV_{DD} 或 EV_{SS} 输出： 保持开路。
UDMF	–	28	K4	–	使用50 kΩ 或更大阻值的电阻将该引脚下拉到V_{SS} 电平。
UDPF	–	29	K5	–	使用50 kΩ 或更大阻值的电阻将该引脚下拉到V_{SS} 电平。
UV_{DD}	–	30	K6	–	直接连接至V_{DD}且始终供电。
AV_{REF0}	–	1	A1, A2, B1	–	直接连接至 V_{DD} 且始终供电。
AV_{REF1}	–	5	B2	–	直接连接至 V_{DD} 且始终供电。
AV_{SS}	–	2	C1, C2	–	直接连接至V_{SS}且始终供电。
RV_{DD}	–	17	D2	–	直接连接至 V_{DD} 且始终供电。
EV_{DD}	–	34, 70	A11, D11, K6, L11, L6	–	直接连接至 V_{DD} 且始终供电。
EV_{SS}	–	33, 69	**注**	–	直接连接至V_{SS}且始终供电。
FLMD0	–	8	F3	–	在flash存储器编程模式外的其它模式下，直接连接至V_{SS}。
REGC	–	10	E1, E2	–	连接稳压器输出稳压电容（4.7 *u*F（推荐值））
$\overline{RESET}$	–	14	H3	2	–
V_{DD}	–	9	D1	–	–
V_{SS}	–	11	G1, G2, J1	–	–
X1	–	12	F1	–	–
X2	–	13	F2	–	–
XT1	–	15	H1	16-C	连接到V_{SS}
XT2	–	16	H2	16-C	保持开路

注 A6, E5 至 E7, E11, F5 至 F7, G5 至 G7, L1, L5

备注 GC：100 针塑封 LQFP（密间距）（14 × 14）
F1：121 针塑封 FBGA（8 × 8）

图 2-1 引脚 I/O 电路



注 端口模式下，没有迟滞特性。

2.4 注意事项

电源开启时，即便在复位期间，以下引脚也可能会暂时输出一个不确定电平。

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

第三章 CPU功能

V850ES/JG3-L 单片机的 CPU 是基于 RISC 架构的，且使用 5 级流水线，几乎所有指令的执行都可在一个时钟周期内完成。

3.1 特性

○ 可变指令长度（16 位/32 位）

○ 最短指令执行周期：
50ns （工作于 20MHz 的主时钟（fxx）：V_{DD} = 2.7 至 3.6V，未使用 USB）
62.5ns（工作于 16MHz 的主时钟（fxx）：V_{DD} = 3.0 至 3.6V，使用 USB）
200ns（工作于 5MHz 的主时钟（fxx）：V_{DD} = 2.2 至 3.6V）
400ns（工作于 2.5MHz 的主时钟（fxx）：V_{DD} = 2.0 至 3.6V）
30.5*us*（工作于 32.768 kHz 的副时钟（f_{XT}）：V_{DD} = 2.0 至 3.6V）

○ 存储空间：
程序空间：64MB 线性
数据空间：4GB 线性

○ 通用寄存器： 32 位×32 个寄存器

○ 内部 32 位架构

○ 5 级流水线控制

○ 乘法/除法指令

○ 饱和运算指令

○ 32 位移位指令：1 个时钟

○ 长/短两种格式的装载/存贮指令

○ 4 种类型位操作指令：

- SET1
- CLR1
- NOT1
- TST1

3.2 CPU寄存器组

V850ES/JG3-L 的寄存器可分为两种类型：通用程序寄存器和专用系统寄存器。所有寄存器都是 32 位宽度。详细信息，请参见 **V850ES 结构用户手册**。

（1）程序寄存器组

31 0

r0	(零寄存器)
r1	(汇编程序保留寄存器)
r2	
r3	(堆栈指针 (SP))
r4	(全局指针 (GP))
r5	(文本指针(TP))
r6	
r7	
r8	
r9	
r10	
r11	
r12	
r13	
r14	
r15	
r16	
r17	
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)

（2）系统寄存器组

31 0

EIPC	(中断状态保存寄存器)
EIPSW	(中断状态保存寄存器)

FEPC	(NMI 状态保存寄存器)
FEPSW	(NMI状态保存寄存器)

ECR	(中断源寄存器)

PSW	(程序状态字)

CTPC	(CALLT 执行状态保存寄存器)
CTPSW	(CALLT 执行状态保存寄存器)

DBPC	(异常/调试陷阱状态保存寄存器)
DBPSW	(异常/调试陷阱状态保存寄存器)

CTBP	(CALLT基址指针)

3.2.1 程序寄存器组

程序寄存器组包括若干通用寄存器和一个程序计数器。

（1）通用寄存器（r0 至 r31）

共有 32 个通用寄存器，r0 至 r31。这些寄存器中的任何一个都可用于存储数据变量或地址变量。

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

表 3-1. 程序寄存器

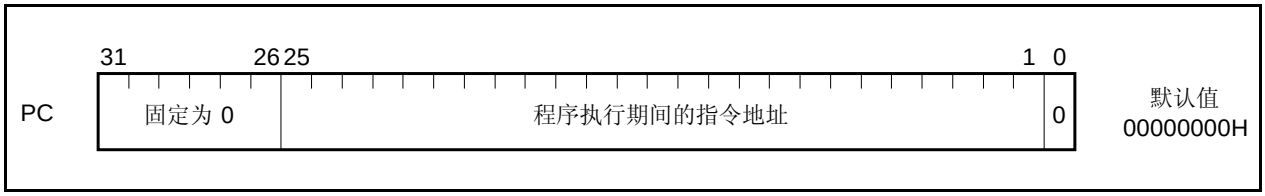
名称	用途	操作
r0	零寄存器	始终保持为零。
r1	汇编程序保留寄存器	用作生成 32 位立即数的工作寄存器
r2	用于地址变量/数据的变量寄存器（如果实时操作系统没有使用寄存器 r2）	
r3	堆栈指针	用于在函数调用时生成堆栈帧
r4	全局指针	用于访问数据区中的全局变量
r5	文本指针	用作指示文本区域起始地址的寄存器（该区域为程序代码存放区域）
r6 至 r29	用于地址/数据变量的寄存器	
r30	元指针	用作访问内存的基址指针
r31	链接指针	编译器调用函数时使用
PC	程序计数器	在程序执行期间，保存指令地址

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

（2）程序计数器（PC）

程序计数器用来保存程序执行期间的指令地址。该寄存器的低 32 位有效。其中第 31 位至第 26 位固定为 0，即使第 25 位向第 26 位有进位，也将忽略不计。

第 0 位固定为 0，这意味着不能执行跳转到奇地址处的操作。



3.2.2 系统寄存器组

系统寄存器控制 CPU 的状态，并保存中断信息。

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

表 3-2. 系统寄存器号

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

注 **1.** 由于只有一组这样中断状态保存寄存器，所以，如果允许多重中断，这些寄存器的内容必须通过编程加以保存。

2. 对这些寄存器的访问，只有在执行 DBTRAP 指令或非法操作码与 DBRET 指令的间隔期间才能进行。

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

备注 √：可以访问

×：禁止访问

（1）中断状态保存寄存器（EIPC 和 EIPSW）

EIPC 和 EIPSW 用于保存产生中断时的状态。

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

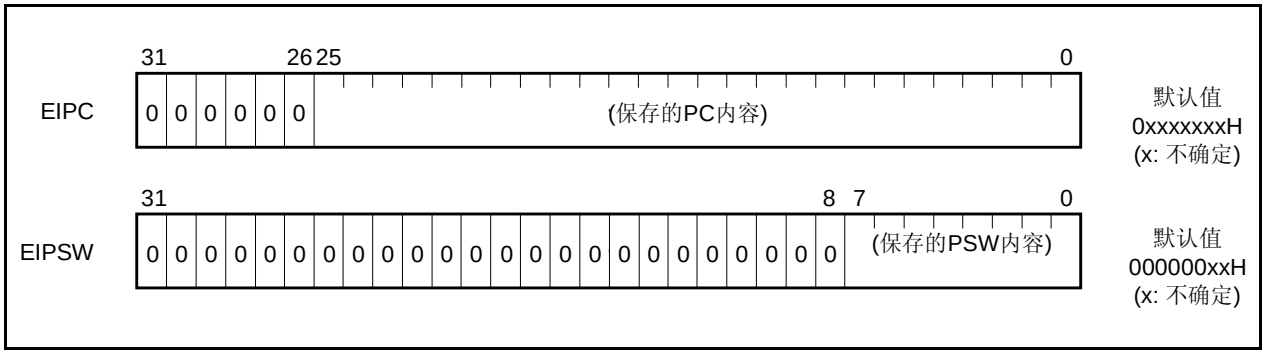
当有软件异常或者是可屏蔽中断产生时，就将当前执行指令的下一条指令地址保存到寄存器 EIPC 中，某些指令除外（可参见 **19.8 CPU 不响应中断期间**）。

将 PSW 当前内容保存到 EIPSW 寄存器中。

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

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

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



（2）NMI 状态保存寄存器（FEPC 和 FEPSW）

FEPC 和 FEPSW 用于保存产生不可屏蔽中断（NMI）时的状态。

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

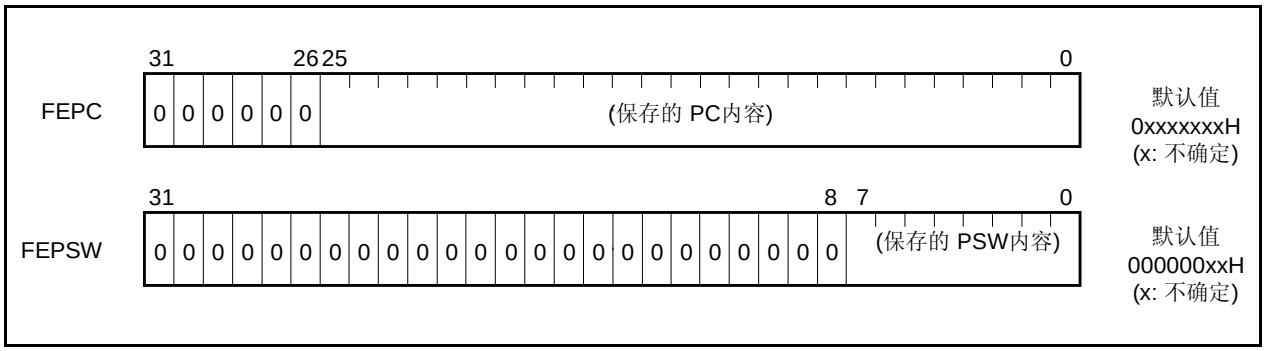
当有 NMI 产生时，就将当前执行指令的下一条指令地址保存到寄存器 FEPC 中，某些指令除外。

将 PSW 当前内容保存到 FEPSW 寄存器中。

由于只有一组 NMI 状态保存寄存器可用，所以，当允许多重中断时（多重中断服务需要使用 NMI 引脚和 INTWDT2 中断请求信号），这些寄存器的内容必须通过编程加以保存。

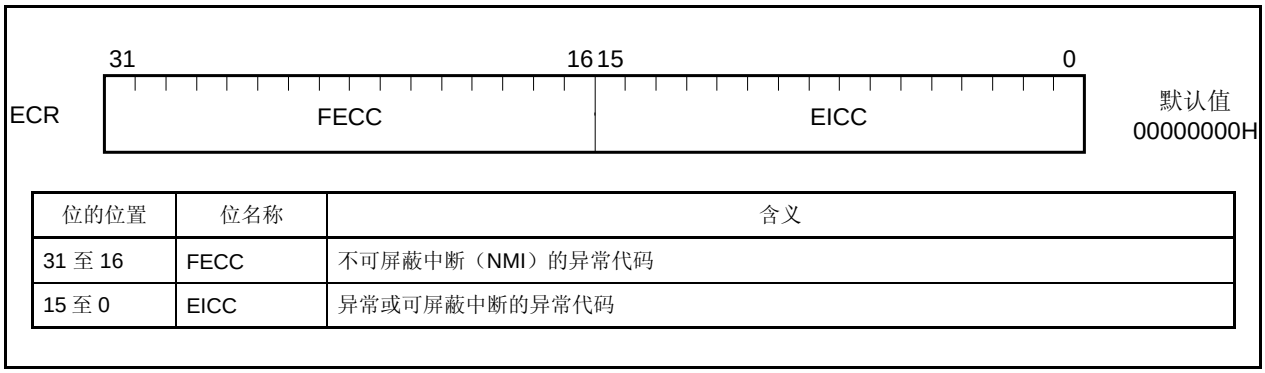
FEPC 的第 31 位至第 26 位以及 FEPSW 的第 31 位至第 8 位作为保留位用于将来功能扩展（这些位始终固定为 0）。

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



（3）中断源寄存器（ECR）

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



位的位置	位名称	含义
31 至 16	FECC	不可屏蔽中断（NMI）的异常代码
15 至 0	EICC	异常或可屏蔽中断的异常代码

（4）程序状态字（PSW）

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

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

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

（1/2）

	31 … 8	7	6	5	4	3	2	1	0	
PSW	RFU	NP	EP	ID	SAT	CY	OV	S	Z	默认值 00000020H

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

备注 请阅读下页的**“注”**。

（2/2）

注 已经执行饱和处理的运算结果由 OV 标志和 S 标志的内容来决定。在执行饱和运算时，只有当 OV 标志位置 1 时，SAT 标志位才置为 1。

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

（5）CALLT 指令执行状态保存寄存器（CTPC 和 CTPSW）

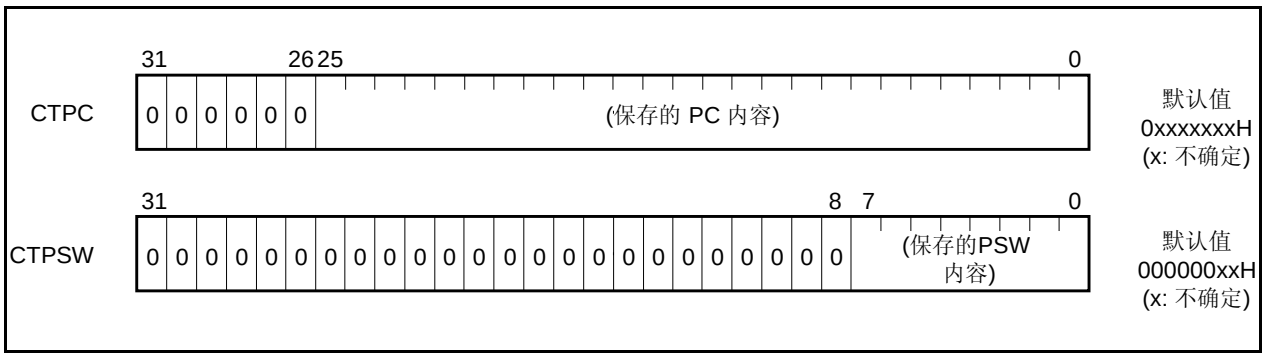
CTPC 和 CTPSW 是 CALLT 指令的执行状态保存寄存器。

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

寄存器 CTPC 中保存的内容是 CALLT 指令之后下一条指令的地址。

将 PSW 当前内容保存到 CTPSW 寄存器中。

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



（6）异常/调试陷阱状态保存寄存器（DBPC 和 DBPSW）

DBPC 和 DBPSW 是异常/调试陷阱状态寄存器。

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

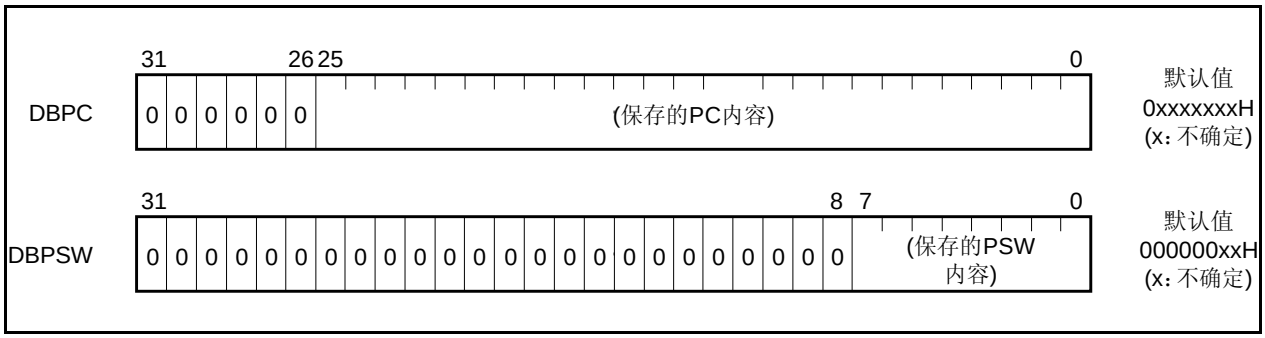
在异常陷阱或者调试陷阱产生时，保存到 DBPC 寄存器中的内容是当前执行指令的下一条指令的地址。

将 PSW 的当前内容保存到 DBPSW 寄存器中。

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

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

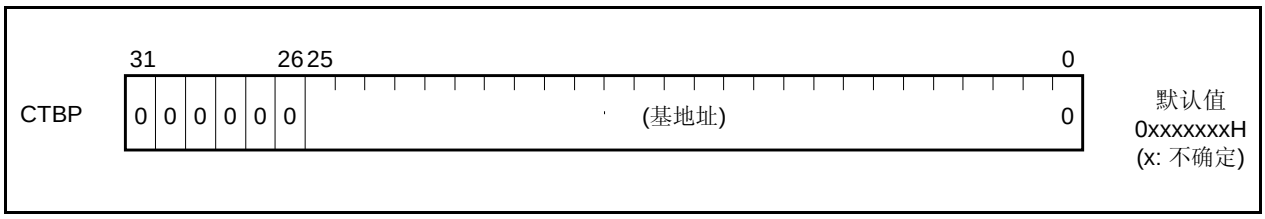
通过执行 DBRET 指令，将寄存器 DBPC 的值恢复到 PC 中，并将寄存器 DBPSW 的值恢复到 PSW 中。



（7）CALLT 指令基址指针（CTBP）

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

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



3.3 操作模式

V850ES/JG3-L 具有以下几种操作模式。

- 正常工作模式
- Flash 存储器编程模式
- 自编程模式
- 片上调试模式

根据 FLMD0 引脚和 FLMD1 引脚的状态，指定操作模式。

要指定正常操作模式，在复位期间向 FLMD0 引脚输入低电平。

如果连接 flash 编程器，在 flash 存储器编程模式下，flash 编程器会将高电平输入 FLMD0 引脚。在自编程模式下，必须由外部电路输入高电平。

在应用系统中请固定这些引脚的规范，在工作期间不要改变这些引脚的设置。

FLMD0	FLMD1	复位后的操作模式
L	×	正常工作模式
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
×： 无需理会

（1）正常工作模式

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

（2）Flash 存储器编程模式

指定了该模式时，可以使用 flash 编程器对内部 flash 进行编程。

（3）自编程模式

通过用户应用程序，可以将 flash 存储器的数据擦除，或者向 flash 存储器写入数据。详情参见**第 30 章 FLASH 存储器**。

（4）片上调试模式

V850ES/JG3-L 提供兼容 JTAG（Joint Test Action Group）通讯规范的片上调试功能。

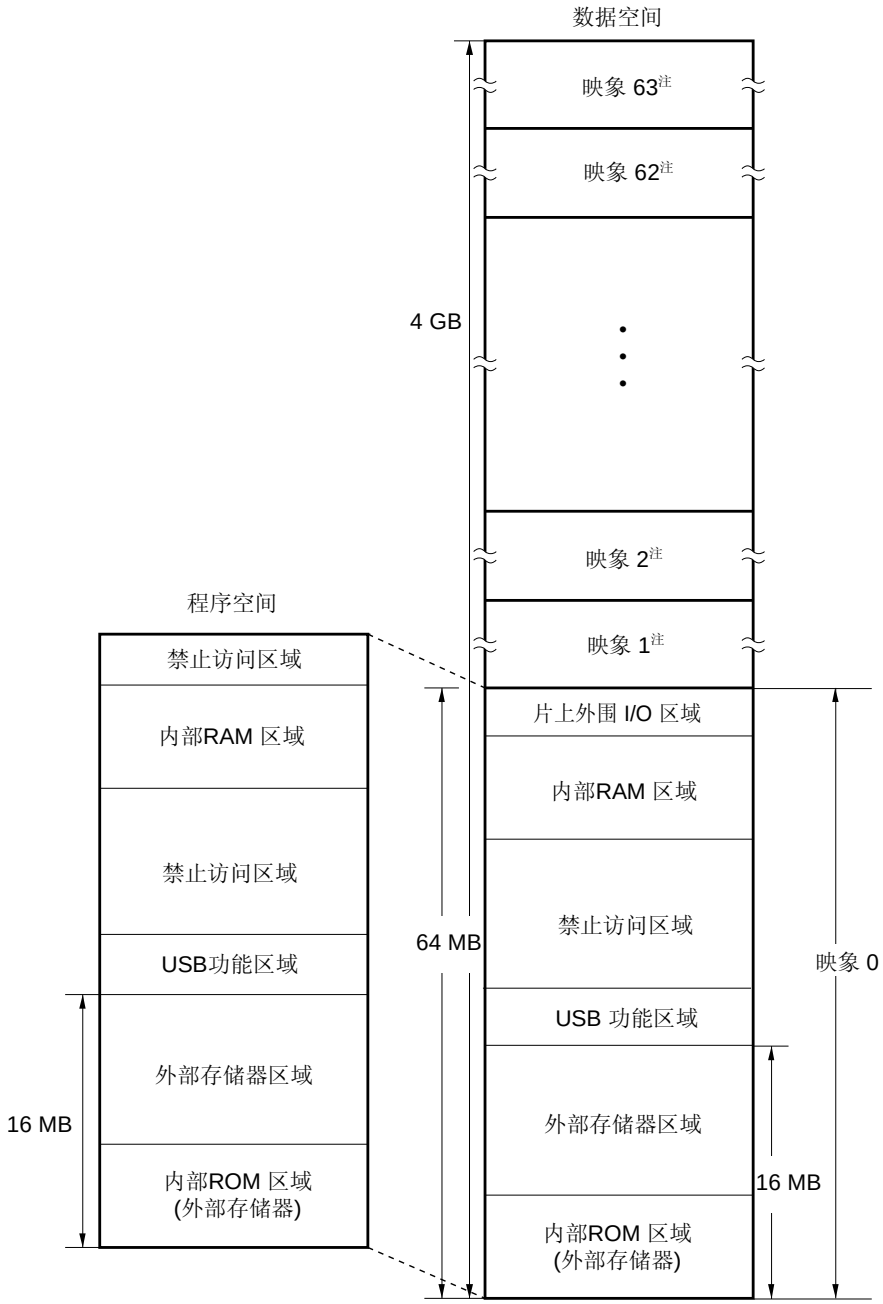
详情参见**第 32 章 片上调试功能**。

3.4 地址空间

3.4.1 CPU地址空间

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

图 3-1. 地址空间映象

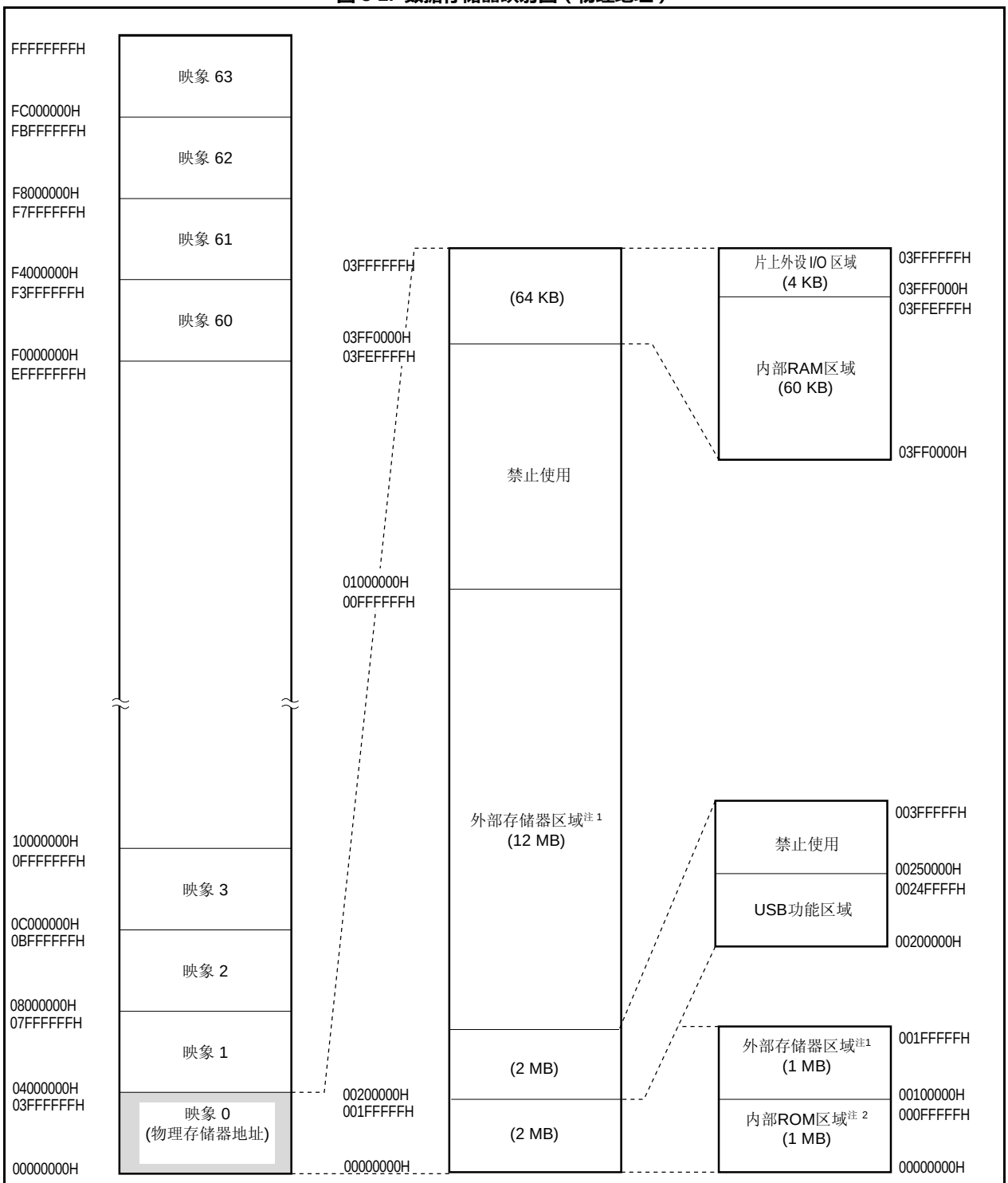


注 映象 1 至 63 是映象 0 结构的重复。

3.4.2 存储器映射图

在 V850ES/JG3-L 内保留了下列区域。

图 3-2. 数据存储器映射图（物理地址）



注 1. V850ES/JG3-L 有 22 个地址引脚，所以外部存储区域表现为重复的 4MB 映象。

2. 对于地址 00000000H 至 000FFFFFH 的取指和读取访问被指向内部 ROM 区域。然而，对这些地址的写入却被指向外部存储区域。

图 3-3. 程序存储器映射图

注 V850ES/JG3-L 有 22 个地址引脚，所以外部存储区域表现为重复的 4MB 映象。

3.4.3 存储区域

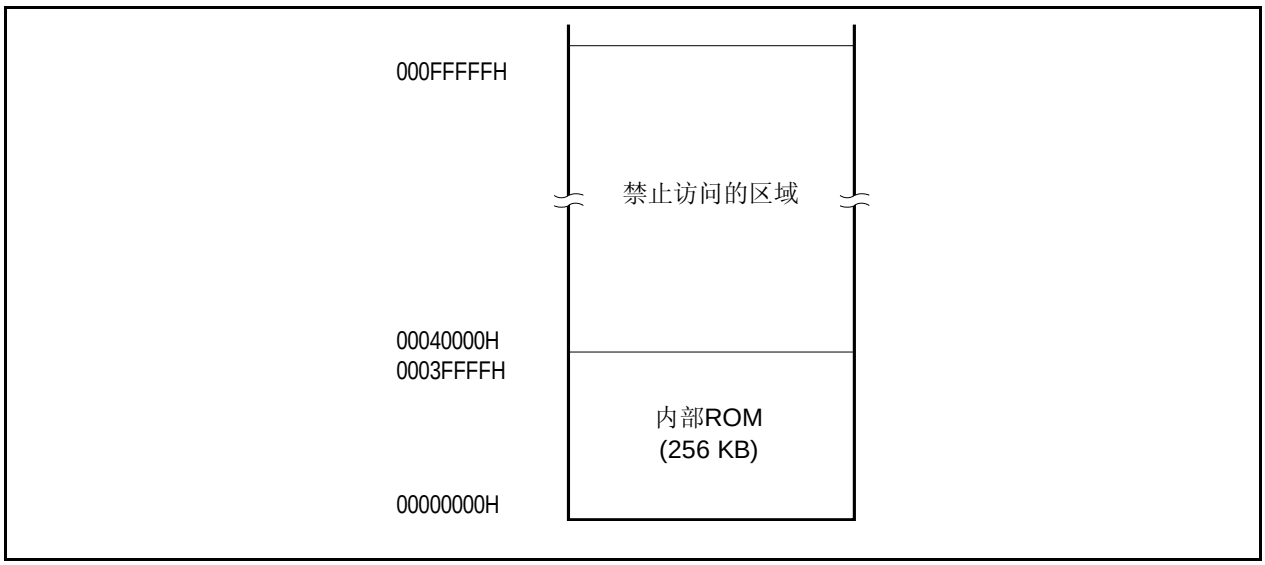
（1）内部 ROM 区域

保留 1 MB 的区域作为内部 ROM 区域。

（a）内部 ROM （256KB）

在μPD70F3794 中，256 KB 被分配在地址 00000000H 至 0003FFFFH。
禁止访问地址 00040000H 至 000FFFFFH。

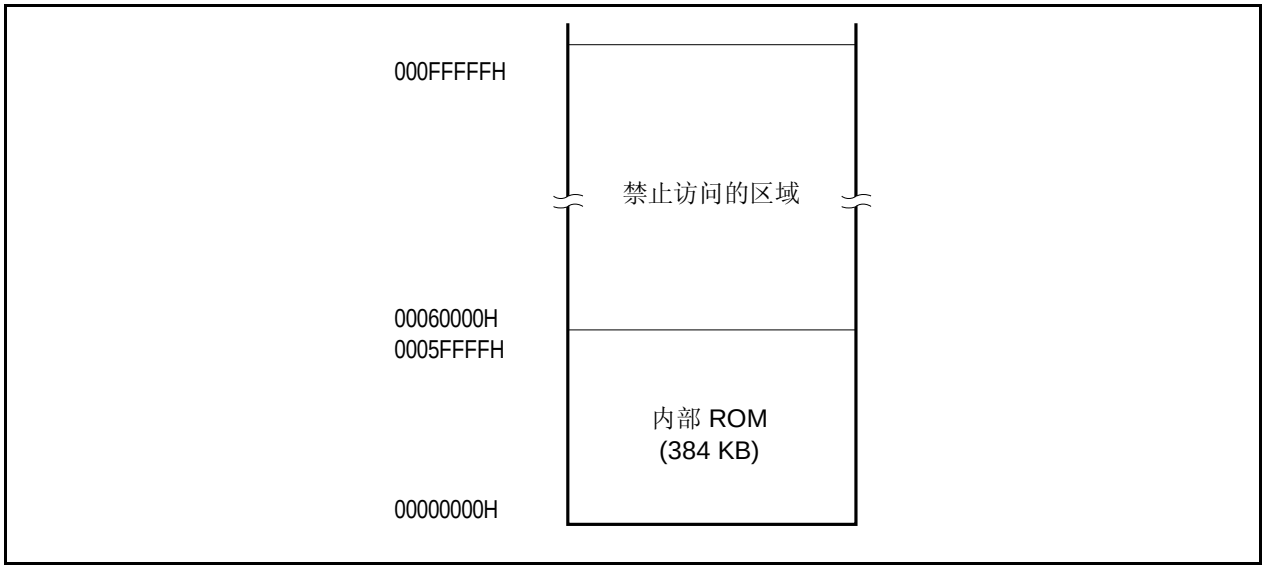
图 3-4. 内部 ROM 区域（256 KB）



（b）内部 ROM （384KB）

在μPD70F3795 中，384 KB 被分配在地址 00000000H 至 0005FFFFH。
禁止访问地址 00060000H 至 000FFFFFH。

图 3-5. 内部 ROM 区域（384 KB）

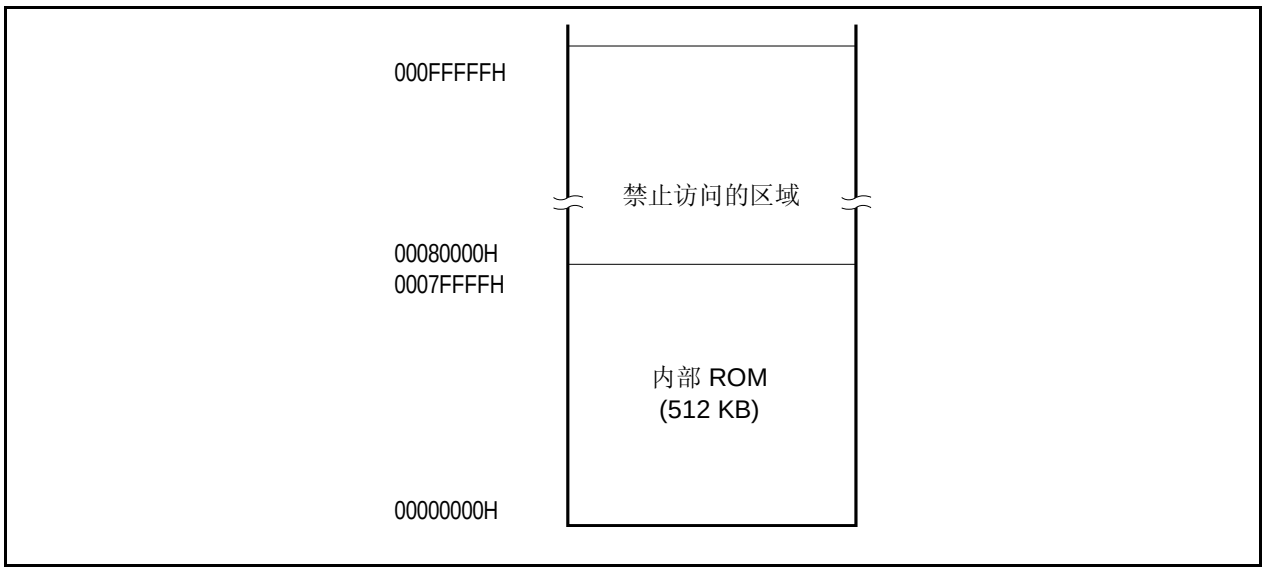


（c）内部 ROM（512 KB）

在μPD70F3796 中，512 KB 分配在地址 00000000H 至 0007FFFFH。

禁止访问地址 00080000H 至 000FFFFFH。

图 3-6. 内部 ROM 区域（512 KB）



（2）内部 RAM 区域

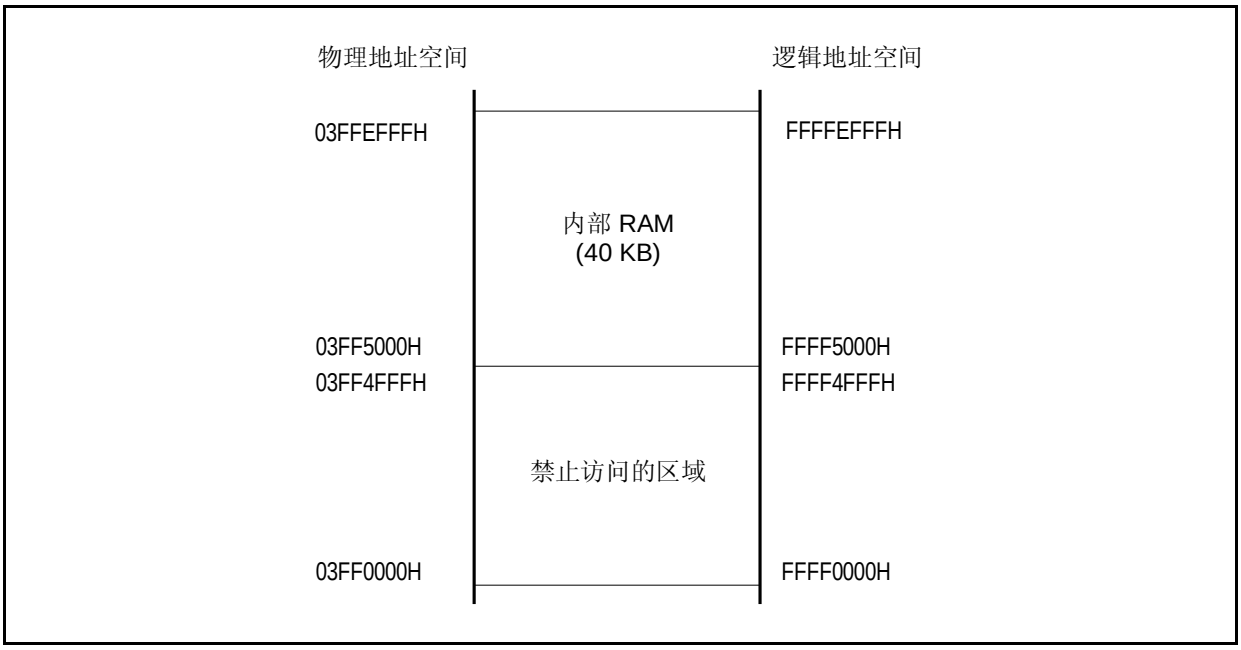
最多有 60 KB 被分配在物理地址 03FF0000H 至 03FFEFFFH 的区域被保留作为内部 RAM 区域。

（a）内部 RAM （40 KB）

在μPD70F3794，70F3795，70F3796 中，40 KB 分配在地址 03FF5000H 至 03FFEFFFH。

禁止访问地址 03FF0000H 至 03FF4FFFH。

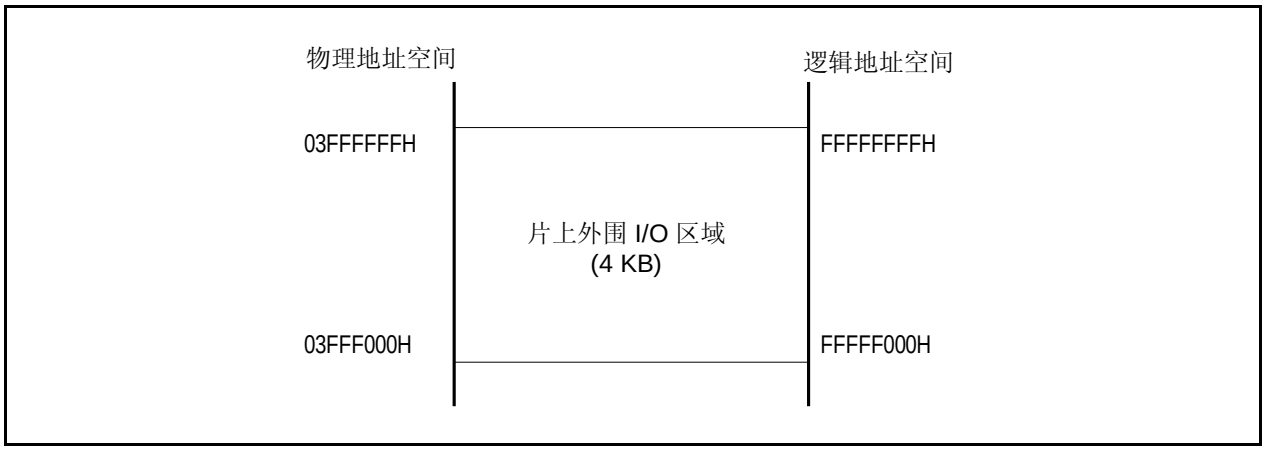
图 3-7. 内部 RAM 区域（40 KB）



（3）片上外围 I/O 区

分配在地址 03FFF000H 至 03FFFFFFH 的 4 KB 存储区域被保留作为片上外围 I/O 区域。

图 3-8. 片上外围 I/O 区域



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

注意事项 1. 以字单元访问外围 I/O 寄存器时，会按照先低位后高位的顺序，对该字区域执行两次半字访问，其中地址的低 2 位被忽略。

2. 如果外围 I/O 寄存器支持字节访问，以半字为单位进行访问时，低 8 位有效。当访问操作是读取操作时，那么该寄存器的高 8 位将不确定。

3. 没有定义为寄存器的地址区域被保留为将来扩展。如果对这些地址进行访问，其操作不确定而且也不保证操作的安全性。

4. 内部 ROM/RAM 区域和片上外围 I/O 区域，都被分配在连续地址。
使用指针或类似的方式，比如，递增或递减地址来访问内部 ROM/RAM 区域时，一定要注意不要越过内部 ROM/RAM 区域而错误访问到片上外围 I/O 区域。

（4）外部存储区

15 MB（00100000H 至 00FFFFFFH）被分配作为外部存储区域。详情参见**第五章 总线控制功能**。

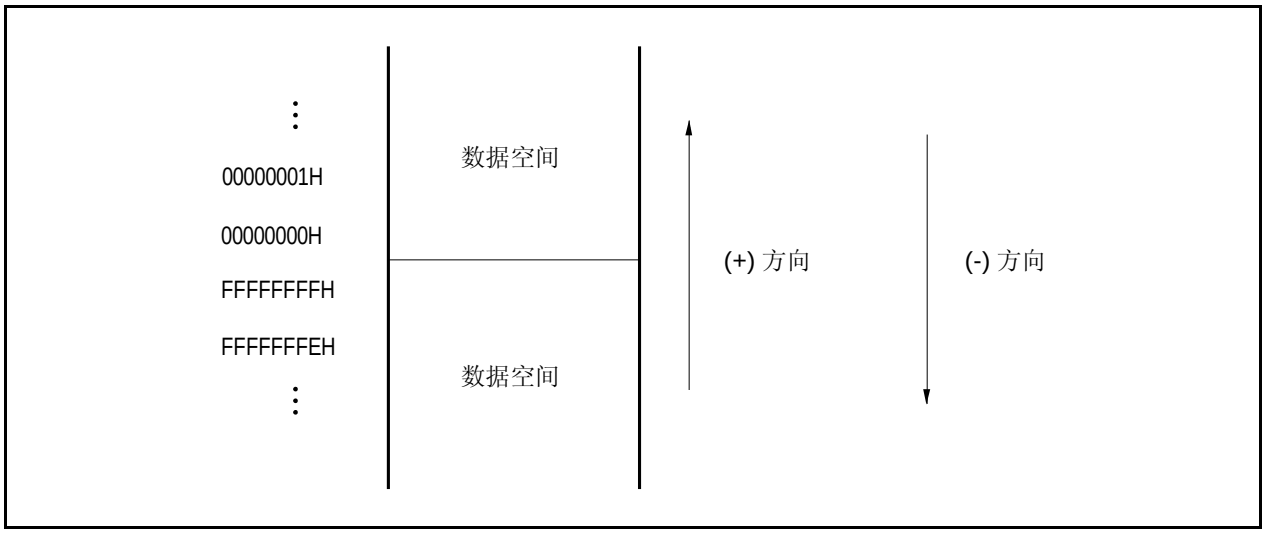
注意事项 V850ES/JG3-L 有 22 个地址引脚（A0 至 A21），所以外部存储区域表现为重复的 4MB 映象。

3.4.4 数据空间的绕回

操作数地址计算的结果超出 32 位时，该结果地址将被忽略。

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

图 3-9. 数据空间的绕回



3.4.5 地址空间的使用建议

V850ES/JG3-L 架构要求访问数据空间的操作数时，必须有一个寄存器用作指针以产生地址。该指针±32 KB 范围的地址可以通过操作数指令直接存取。由于能够用作指针的通用寄存器数量是有限的，当改变指针值计算地址时，为了不影响系统的性能，应该尽可能多的指定通用寄存器来保存变量，这样也可以减少程序所占的空间。

（1）程序空间

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

要使用内部 RAM 作为程序空间，请访问下列地址。

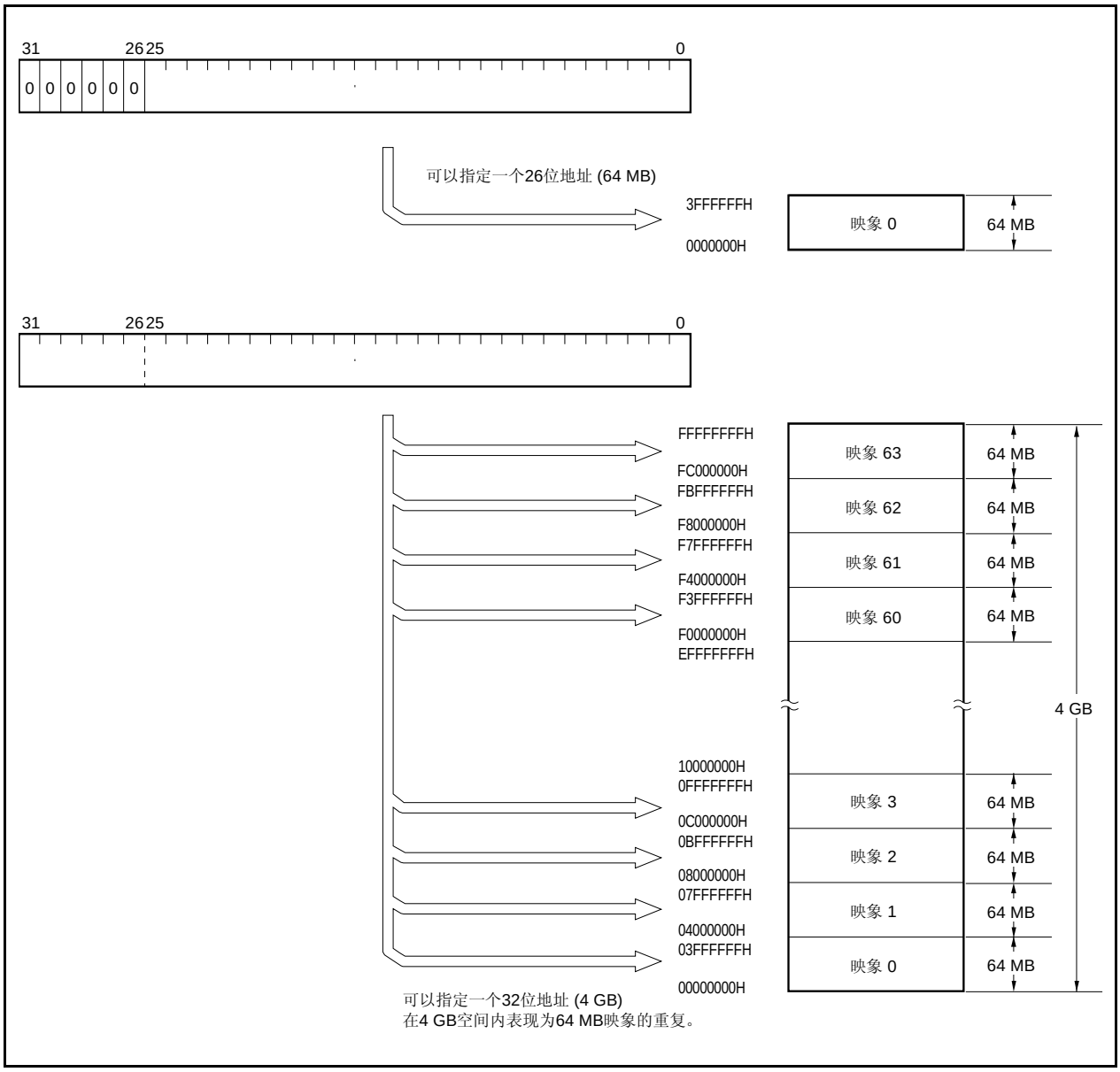
注意事项 如果跳转指令在内部 RAM 区间的地址上限处发生，那么将不会产生跨越片上外围 I/O 区域中的预取指操作（无效取指）。

产品名称	RAM 容量	访问地址
μPD70F3794	40 KB	03FF5000H 至 03FFEFFFH
μPD70F3795		
μPD70F3796		

（2）数据空间

对于 V850ES/JG3-L 来说，4GB（32 位地址）CPU 地址空间内可看作是 64 个 64MB（26 位地址）的地址空间。因此，这些 64 MB 空间的 26 位地址的最高有效位（第 25 位）作为符号扩展到 32 位，并作为地址分配。

图 3-10. 数据空间内的符号扩展



（a）地址绕回应用示例

如果 LD/ST disp16[R]指令使用 R = r0 （零寄存器），对于从 00000000H 开始的±32KB 地址范围空间，就可以通过 disp16 的带符号扩展进行寻址。通过一个指针就可以对所有资源，包括内部硬件，进行寻址。

零寄存器（r0）是由硬件将其值固定为 0 的，实际上不需要专门用作指针的寄存器。

图 3-11. μPD70F3794 内数据空间用法示例

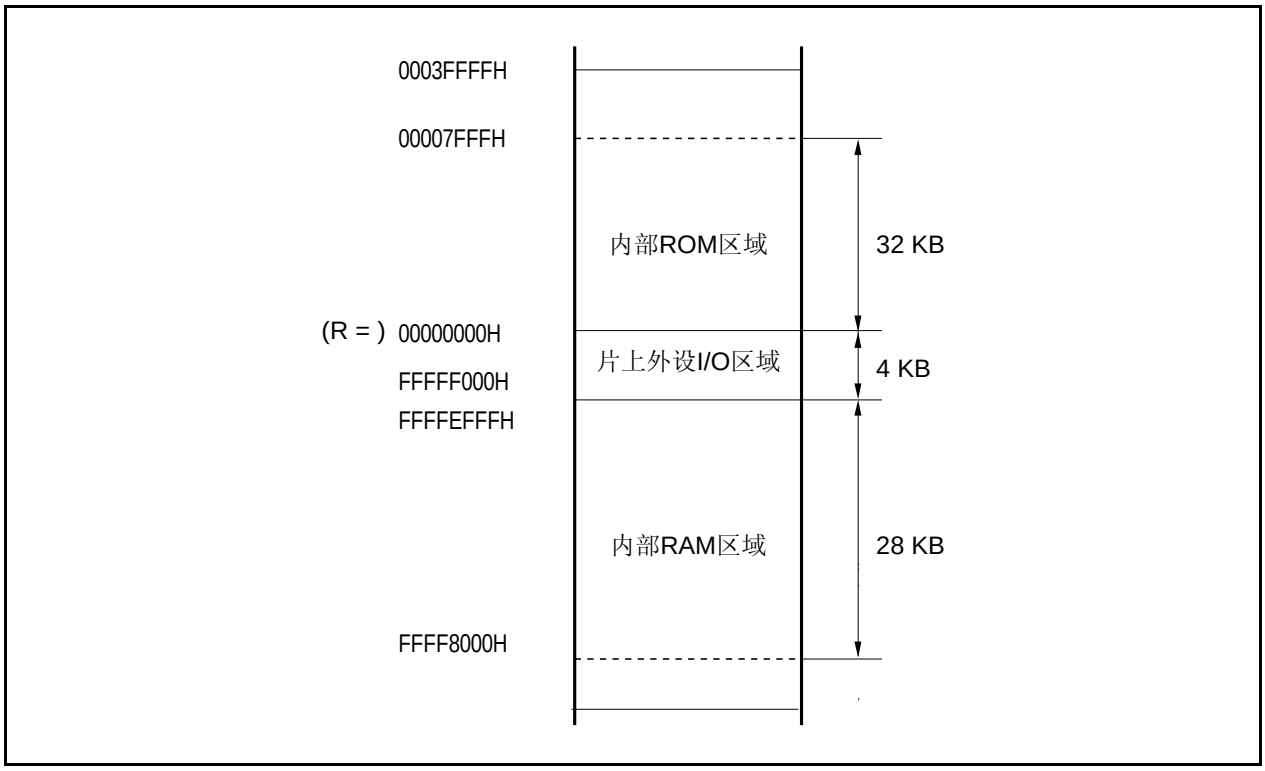
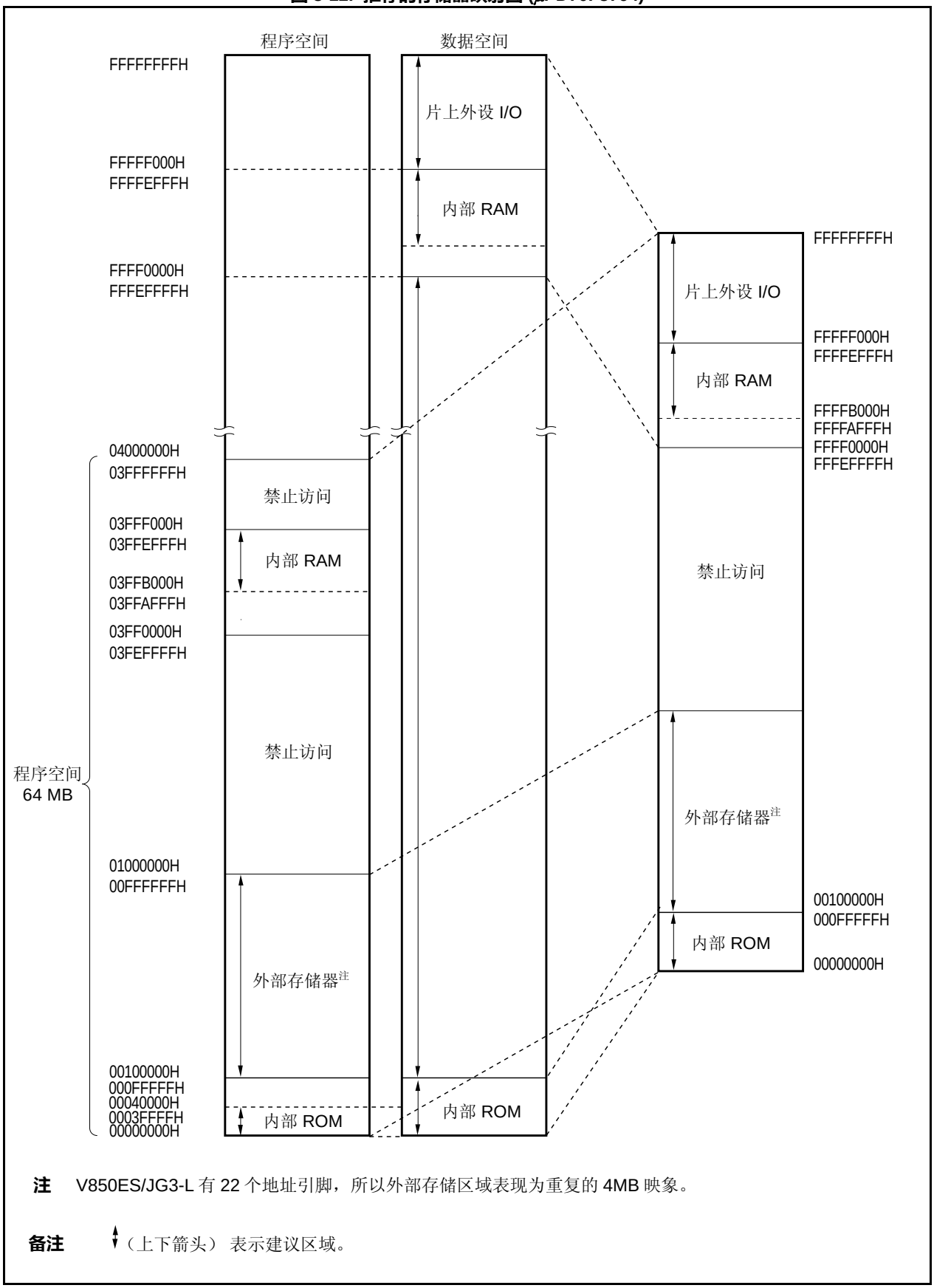


图 3-12. 推荐的存储器映射图 (μPD70F3794)



注 V850ES/JG3-L 有 22 个地址引脚，所以外部存储区域表现为重复的 4MB 映象。

备注 ↕（上下箭头） 表示建议区域。

3.4.6 外围I/O 寄存器

(1/11)

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

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

(2/11)

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF0D2H	DMA 寻址控制寄存器 1	DADC1	R/W			√	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 (INTLVI)	LVIIC		√	√		47H
FFFFF112H	中断控制寄存器 (INTP0)	PIC0		√	√		47H
FFFFF114H	中断控制寄存器 (INTP1)	PIC1		√	√		47H
FFFFF116H	中断控制寄存器 (INTP2)	PIC2		√	√		47H
FFFFF118H	中断控制寄存器 (INTP3)	PIC3		√	√		47H
FFFFF11AH	中断控制寄存器 (INTP4)	PIC4		√	√		47H
FFFFF11CH	中断控制寄存器 (INTP5)	PIC5		√	√		47H
FFFFF11EH	中断控制寄存器 (INTP6)	PIC6		√	√		47H
FFFFF120H	中断控制寄存器 (INTP7)	PIC7		√	√		47H
FFFFF122H	中断控制寄存器 (INTTQ0OV)	TQ0OVIC		√	√		47H
FFFFF124H	中断控制寄存器 (INTTQ0CC0)	TQ0CCIC0		√	√		47H
FFFFF126H	中断控制寄存器 (INTTQ0CC1)	TQ0CCIC1		√	√		47H
FFFFF128H	中断控制寄存器 (INTTQ0CC2)	TQ0CCIC2		√	√		47H
FFFFF12AH	中断控制寄存器 (INTTQ0CC3)	TQ0CCIC3		√	√		47H
FFFFF12CH	中断控制寄存器 (INTTP0OV)	TP0OVIC		√	√		47H
FFFFF12EH	中断控制寄存器 (INTTP0CC0)	TP0CCIC0		√	√		47H
FFFFF130H	中断控制寄存器 (INTTP0CC1)	TP0CCIC1		√	√		47H
FFFFF132H	中断控制寄存器 (INTTP1OV)	TP1OVIC		√	√		47H
FFFFF134H	中断控制寄存器 (INTTP1CC0)	TP1CCIC0		√	√		47H
FFFFF136H	中断控制寄存器 (INTTP1CC1)	TP1CCIC1		√	√		47H
FFFFF138H	中断控制寄存器 (INTTP2OV)	TP2OVIC		√	√		47H
FFFFF13AH	中断控制寄存器 (INTTP2CC0)	TP2CCIC0		√	√		47H
FFFFF13CH	中断控制寄存器 (INTTP2CC1)	TP2CCIC1		√	√		47H
FFFFF13EH	中断控制寄存器 (INTTP3OV)	TP3OVIC		√	√		47H

(3/11)

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF140H	中断控制寄存器 (INTTP3CC0)	TP3CCIC0	R/W	√	√		47H
FFFFF142H	中断控制寄存器 (INTTP3CC1)	TP3CCIC1		√	√		47H
FFFFF144H	中断控制寄存器 (INTTP4OV)	TP4OVIC		√	√		47H
FFFFF146H	中断控制寄存器 (INTTP4CC0)	TP4CCIC0		√	√		47H
FFFFF148H	中断控制寄存器 (INTTP4CC1)	TP4CCIC1		√	√		47H
FFFFF14AH	中断控制寄存器 (INTTP5OV)	TP5OVIC		√	√		47H
FFFFF14CH	中断控制寄存器 (INTTP5CC0)	TP5CCIC0		√	√		47H
FFFFF14EH	中断控制寄存器 (INTTP5CC1)	TP5CCIC1		√	√		47H
FFFFF150H	中断控制寄存器 (INTTM0EQ0)	TM0EQIC0		√	√		47H
FFFFF152H	中断控制寄存器 (INTCB0R/INTIIC1)	CB0RIC/IICIC1		√	√		47H
FFFFF154H	中断控制寄存器 (INTCB0T)	CB0TIC		√	√		47H
FFFFF156H	中断控制寄存器 (INTCB1R)	CB1RIC		√	√		47H
FFFFF158H	中断控制寄存器 (INTCB1T)	CB1TIC		√	√		47H
FFFFF15AH	中断控制寄存器 (INTCB2R)	CB2RIC		√	√		47H
FFFFF15CH	中断控制寄存器 (INTCB2T)	CB2TIC		√	√		47H
FFFFF15EH	中断控制寄存器 (INTCB3R)	CB3RIC		√	√		47H
FFFFF160H	中断控制寄存器 (INTCB3T)	CB3TIC		√	√		47H
FFFFF162H	中断控制寄存器 (INTUA0R/INTCB4R)	UA0RIC/CB4RIC		√	√		47H
FFFFF164H	中断控制寄存器 (INTUA0T/INTCB4T)	UA0TIC/CB4TIC		√	√		47H
FFFFF166H	中断控制寄存器 (INTUA1R/INTIIC2)	UA1RIC/IICIC2		√	√		47H
FFFFF168H	中断控制寄存器 (INTUA1T)	UA1TIC		√	√		47H
FFFFF16AH	中断控制寄存器 (INTUA2R/INTIIC0)	UA2RIC/IICIC0		√	√		47H
FFFFF16CH	中断控制寄存器 (INTUA2T)	UA2TIC		√	√		47H
FFFFF16EH	中断控制寄存器 (INTAD)	ADIC		√	√		47H
FFFFF170H	中断控制寄存器 (INTDMA0)	DMAIC0		√	√		47H
FFFFF172H	中断控制寄存器 (INTDMA1)	DMAIC1		√	√		47H
FFFFF174H	中断控制寄存器 (INTDMA2)	DMAIC2		√	√		47H
FFFFF176H	中断控制寄存器 (INTDMA3)	DMAIC3		√	√		47H
FFFFF178H	中断控制寄存器 (INTKR)	KRIC		√	√		47H
FFFFF17AH	中断控制寄存器 (INTWTI/INTRTC2)	WTIIC/RTC2IC		√	√		47H
FFFFF17CH	中断控制寄存器 (INTWT/INTRTC0)	WTIC/RTC0IC		√	√		47H
FFFFF17EH	中断控制寄存器 (INTRTC1)	RTC1C		√	√		47H
FFFFF180H	中断控制寄存器 (INTUA3R)	UA3RIC		√	√		47H
FFFFF182H	中断控制寄存器 (INTUA3T)	UA3TI		√	√		47H
FFFFF184H	中断控制寄存器 (INTUA4R)	UA4RIC		√	√		47H
FFFFF186H	中断控制寄存器 (INTUA4T)	UA4TIC		√	√		47H
FFFFF188H	中断控制寄存器 (INTUC0R)	UC0RIC		√	√		47H
FFFFF18AH	中断控制寄存器 (INTUC0T)	UC0TIC		√	√		47H
FFFFF1FAH	正在服务的中断优先级寄存器	ISPR	R	√	√		00H
FFFFF1FCH	命令寄存器	PRCMD	W		√		Undefined
FFFFF1FEH	省电控制寄存器	PSC 注	R/W	√	√		00H

注 这是一个专用寄存器。

(4/11)

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF200H	A/D 转换器模式寄存器 0	ADA0M0	R/W	√	√		00H
FFFFF201H	A/D 转换器模式寄存器 1	ADA0M1		√	√		00H
FFFFF202H	A/D 转换器通道指定寄存器	ADA0S		√	√		00H
FFFFF203H	A/D 转换器模式寄存器 2	ADA0M2		√	√		00H
FFFFF204H	掉电比较模式寄存器	ADA0PFM		√	√		00H
FFFFF205H	掉电比较阈值寄存器	ADA0PFT		√	√		00H
FFFFF210H	A/D 转换结果寄存器 0	ADA0CR0	R			√	Undefined
FFFFF211H	A/D 转换结果寄存器 0H	ADA0CR0H			√		Undefined
FFFFF212H	A/D 转换结果寄存器 1	ADA0CR1				√	Undefined
FFFFF213H	A/D 转换结果寄存器 1H	ADA0CR1H			√		Undefined
FFFFF214H	A/D 转换结果寄存器 2	ADA0CR2				√	Undefined
FFFFF215H	A/D 转换结果寄存器 2H	ADA0CR2H			√		Undefined
FFFFF216H	A/D 转换结果寄存器 3	ADA0CR3				√	Undefined
FFFFF217H	A/D 转换结果寄存器 3H	ADA0CR3H			√		Undefined
FFFFF218H	A/D 转换结果寄存器 4	ADA0CR4				√	Undefined
FFFFF219H	A/D 转换结果寄存器 4H	ADA0CR4H			√		Undefined
FFFFF21AH	A/D 转换结果寄存器 5	ADA0CR5				√	Undefined
FFFFF21BH	A/D 转换结果寄存器 5H	ADA0CR5H			√		Undefined
FFFFF21CH	A/D 转换结果寄存器 6	ADA0CR6				√	Undefined
FFFFF21DH	A/D 转换结果寄存器 6H	ADA0CR6H			√		Undefined
FFFFF21EH	A/D 转换结果寄存器 7	ADA0CR7				√	Undefined
FFFFF21FH	A/D 转换结果寄存器 7H	ADA0CR7H			√		Undefined
FFFFF220H	A/D 转换结果寄存器 8	ADA0CR8				√	Undefined
FFFFF221H	A/D 转换结果寄存器 8H	ADA0CR8H			√		Undefined
FFFFF222H	A/D 转换结果寄存器 9	ADA0CR9				√	Undefined
FFFFF223H	A/D 转换结果寄存器 9H	ADA0CR9H			√		Undefined
FFFFF224H	A/D 转换结果寄存器 10	ADA0CR10				√	Undefined
FFFFF225H	A/D 转换结果寄存器 10H	ADA0CR10H			√		Undefined
FFFFF226H	A/D 转换结果寄存器 11	ADA0CR11				√	Undefined
FFFFF227H	A/D 转换结果寄存器 11H	ADA0CR11H			√		Undefined
FFFFF280H	D/A 转换值设置寄存器 0	DA0CS0	R/W		√		00H
FFFFF281H	D/A 转换值设置寄存器 1	DA0CS1			√		00H
FFFFF282H	D/A 转换模式寄存器	DA0M		√	√		00H
FFFFF300H	按键返回模式寄存器	KRM		√	√		00H
FFFFF308H	选择器操作控制寄存器 0	SELCNT0		√	√		00H
FFFFF310H	CRC 输入寄存器	CRCIN			√		00H
FFFFF312H	CRC 数据寄存器	CRCD				√	0000H
FFFFF318H	噪声消除控制寄存器	NFC			√		00H
FFFFF320H	预分频模式寄存器 1	PRSM1		√	√		00H
FFFFF321H	预分频器比较寄存器 1	PRSCM1			√		00H
FFFFF324H	预分频模式寄存器 2	PRSM2		√	√		00H
FFFFF325H	预分频器比较寄存器 2	PRSCM2			√		00H
FFFFF328H	预分频模式寄存器 3	PRSM3		√	√		00H
FFFFF329H	预分频器比较寄存器 3	PRSCM3			√		00H

(5/11)

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF331H	稳压器保护寄存器	REGPR	R/W		√		00H
FFFFF332H	稳压器输出电压级别控制寄存器	REGOVL0			√		00H
FFFFF340H	IIC 分频时钟选择寄存器	OCKS0			√		00H
FFFFF344H	IIC 分频时钟选择寄存器	OCKS1			√		00H
FFFFF348H	时钟选择寄存器	OCKS2			√		00H
FFFFF380H	时钟直通选择寄存器	CKTHSEL		√	√		00H
FFFFF400H	端口 0 寄存器	P0		√	√		00H 注
FFFFF402H	端口 1 寄存器	P1		√	√		00H 注
FFFFF406H	端口 3 寄存器	P3				√	0000H 注
FFFFF406H	端口 3 寄存器 L	P3L		√	√		00H 注
FFFFF407H	端口 3 寄存器 H	P3H		√	√		00H 注
FFFFF408H	端口 4 寄存器	P4		√	√		00H 注
FFFFF40AH	端口 5 寄存器	P5		√	√		00H 注
FFFFF40EH	端口 7 寄存器 L	P7L		√	√		00H 注
FFFFF40FH	端口 7 寄存器 H	P7H		√	√		00H 注
FFFFF412H	端口 9 寄存器	P9				√	0000H 注
FFFFF412H	端口 9 寄存器 L	P9L		√	√		00H 注
FFFFF413H	端口 9 寄存器 H	P9H		√	√		00H 注
FFFFF420H	端口 0 模式寄存器	PM0		√	√		FFH
FFFFF422H	端口 1 模式寄存器	PM1		√	√		FFH
FFFFF426H	端口 3 模式寄存器	PM3				√	FFFFH
FFFFF426H	端口 3 模式寄存器 L	PM3L		√	√		FFH
FFFFF427H	端口 3 模式寄存器 H	PM3H		√	√		FFH
FFFFF428H	端口 4 模式寄存器	PM4		√	√		FFH
FFFFF42AH	端口 5 模式寄存器	PM5		√	√		FFH
FFFFF42EH	端口 7 模式寄存器 L	PM7L		√	√		FFH
FFFFF42FH	端口 7 模式寄存器 H	PM7H		√	√		FFH
FFFFF432H	端口 9 模式寄存器	PM9				√	FFFFH
FFFFF432H	端口 9 模式寄存器 L	PM9L		√	√		FFH
FFFFF433H	端口 9 模式寄存器 H	PM9H		√	√		FFH
FFFFF440H	端口 0 模式控制寄存器	PMC0		√	√		00H
FFFFF446H	端口 3 模式控制寄存器	PMC3				√	0000H
FFFFF446H	端口 3 模式控制寄存器 L	PMC3L		√	√		00H
FFFFF447H	端口 3 模式控制寄存器 H	PMC3H		√	√		00H
FFFFF448H	端口 4 模式控制寄存器	PMC4		√	√		00H
FFFFF44AH	端口 5 模式控制寄存器	PMC5		√	√		00H
FFFFF452H	端口 9 模式控制寄存器	PMC9				√	0000H
FFFFF452H	端口 9 模式控制寄存器 L	PMC9L		√	√		00H
FFFFF453H	端口 9 模式控制寄存器 H	PMC9H		√	√		00H
FFFFF460H	端口 0 功能控制寄存器	PFC0		√	√		00H
FFFFF466H	端口 3 功能控制寄存器	PFC3				√	0000H
FFFFF466H	端口 3 功能控制寄存器 L	PFC3L		√	√		00H
FFFFF467H	端口 3 功能控制寄存器 H	PFC3H		√	√		00H

注 输出锁存是 00H 或 0000H。当这些寄存器作为输入时，读取到的是引脚状态。

(6/11)

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF468H	端口 4 功能控制寄存器	PFC4	R/W	√	√		00H
FFFFF46AH	端口 5 功能控制寄存器	PFC5		√	√		00H
FFFFF472H	端口 9 功能控制寄存器	PFC9				√	0000H
FFFFF472H	端口 9 功能控制寄存器 L	PFC9L		√	√		00H
FFFFF473H	端口 9 功能控制寄存器 H	PFC9H		√	√		00H
FFFFF484H	数据等待控制寄存器 0	DWC0				√	7777H
FFFFF488H	地址等待控制寄存器	AWC				√	FFFFH
FFFFF48AH	总线周期控制寄存器	BCC				√	AAAAH
FFFFF540H	TMQ0 控制寄存器 0	TQ0CTL0		√	√		00H
FFFFF541H	TMQ0 控制寄存器 1	TQ0CTL1		√	√		00H
FFFFF542H	TMQ0 I/O 控制寄存器 0	TQ0IOC0		√	√		00H
FFFFF543H	TMQ0 I/O 控制寄存器 1	TQ0IOC1		√	√		00H
FFFFF544H	TMQ0 I/O 控制寄存器 2	TQ0IOC2		√	√		00H
FFFFF545H	TMQ0 选项寄存器 0	TQ0OPT0		√	√		00H
FFFFF546H	TMQ0 捕获/比较寄存器 0	TQ0CCR0				√	0000H
FFFFF548H	TMQ0 捕获/比较寄存器 1	TQ0CCR1				√	0000H
FFFFF54AH	TMQ0 捕获/比较寄存器 2	TQ0CCR2				√	0000H
FFFFF54CH	TMQ0 捕获/比较寄存器 3	TQ0CCR3				√	0000H
FFFFF54EH	TMQ0 计数器读取缓冲寄存器	TQ0CNT	R			√	0000H
FFFFF590H	TMP0 控制寄存器 0	TP0CTL0	R/W	√	√		00H
FFFFF591H	TMP0 控制寄存器 1	TP0CTL1		√	√		00H
FFFFF592H	TMP0 I/O 控制寄存器 0	TP0IOC0		√	√		00H
FFFFF593H	TMP0 I/O 控制寄存器 1	TP0IOC1		√	√		00H
FFFFF594H	TMP0 I/O 控制寄存器 2	TP0IOC2		√	√		00H
FFFFF595H	TMP0 选项寄存器 0	TP0OPT0		√	√		00H
FFFFF596H	TMP0 捕获/比较寄存器 0	TP0CCR0				√	0000H
FFFFF598H	TMP0 捕获/比较寄存器 1	TP0CCR1				√	0000H
FFFFF59AH	TMP0 计数器读取缓冲寄存器	TP0CNT	R			√	0000H
FFFFF5A0H	TMP1 控制寄存器 0	TP1CTL0	R/W	√	√		00H
FFFFF5A1H	TMP1 控制寄存器 1	TP1CTL1		√	√		00H
FFFFF5A2H	TMP1 I/O 控制寄存器 0	TP1IOC0		√	√		00H
FFFFF5A3H	TMP1 I/O 控制寄存器 1	TP1IOC1		√	√		00H
FFFFF5A4H	TMP1 I/O 控制寄存器 2	TP1IOC2		√	√		00H
FFFFF5A5H	TMP1 选项寄存器 0	TP1OPT0		√	√		00H
FFFFF5A6H	TMP1 捕获/比较寄存器 0	TP1CCR0				√	0000H
FFFFF5A8H	TMP1 捕获/比较寄存器 1	TP1CCR1				√	0000H
FFFFF5AAH	TMP1 计数器读取缓冲寄存器	TP1CNT	R			√	0000H
FFFFF5B0H	TMP2 控制寄存器 0	TP2CTL0	R/W	√	√		00H
FFFFF5B1H	TMP2 控制寄存器 1	TP2CTL1		√	√		00H
FFFFF5B2H	TMP2 I/O 控制寄存器 0	TP2IOC0		√	√		00H
FFFFF5B3H	TMP2 I/O 控制寄存器 1	TP2IOC1		√	√		00H
FFFFF5B4H	TMP2 I/O 控制寄存器 2	TP2IOC2		√	√		00H
FFFFF5B5H	TMP2 选项寄存器 0	TP2OPT0		√	√		00H
FFFFF5B6H	TMP2 捕获/比较寄存器 0	TP2CCR0				√	0000H

(7/11)

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF5B8H	TMP2 捕获/比较寄存器 1	TP2CCR1	R/W			√	0000H
FFFFF5BAH	TMP2 计数器读取缓冲寄存器	TP2CNT	R			√	0000H
FFFFF5C0H	TMP3 控制寄存器 0	TP3CTL0	R/W	√	√		00H
FFFFF5C1H	TMP3 控制寄存器 1	TP3CTL1		√	√		00H
FFFFF5C2H	TMP3 I/O 控制寄存器 0	TP3IOC0		√	√		00H
FFFFF5C3H	TMP3 I/O 控制寄存器 1	TP3IOC1		√	√		00H
FFFFF5C4H	TMP3 I/O 控制寄存器 2	TP3IOC2		√	√		00H
FFFFF5C5H	TMP3 选项寄存器 0	TP3OPT0		√	√		00H
FFFFF5C6H	TMP3 捕获/比较寄存器 0	TP3CCR0				√	0000H
FFFFF5C8H	TMP3 捕获/比较寄存器 1	TP3CCR1				√	0000H
FFFFF5CAH	TMP3 计数器读取缓冲寄存器	TP3CNT	R			√	0000H
FFFFF5D0H	TMP4 控制寄存器 0	TP4CTL0	R/W	√	√		00H
FFFFF5D1H	TMP4 控制寄存器 1	TP4CTL1		√	√		00H
FFFFF5D2H	TMP4 I/O 控制寄存器 0	TP4IOC0		√	√		00H
FFFFF5D3H	TMP4 I/O 控制寄存器 1	TP4IOC1		√	√		00H
FFFFF5D4H	TMP4 I/O 控制寄存器 2	TP4IOC2		√	√		00H
FFFFF5D5H	TMP4 选项寄存器 0	TP4OPT0		√	√		00H
FFFFF5D6H	TMP4 捕获/比较寄存器 0	TP4CCR0				√	0000H
FFFFF5D8H	TMP4 捕获/比较寄存器 1	TP4CCR1				√	0000H
FFFFF5DAH	TMP4 计数器读取缓冲寄存器	TP4CNT	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	TMP5 I/O 控制寄存器 0	TP5IOC0		√	√		00H
FFFFF5E3H	TMP5 I/O 控制寄存器 1	TP5IOC1		√	√		00H
FFFFF5E4H	TMP5 I/O 控制寄存器 2	TP5IOC2		√	√		00H
FFFFF5E5H	TMP5 选项寄存器 0	TP5OPT0		√	√		00H
FFFFF5E6H	TMP5 捕获/比较寄存器 0	TP5CCR0				√	0000H
FFFFF5E8H	TMP5 捕获/比较寄存器 1	TP5CCR1				√	0000H
FFFFF5EAH	TMP5 计数器读取缓冲寄存器	TP5CNT	R			√	0000H
FFFFF680H	钟表定时器操作模式寄存器	WTM	R/W	√	√		00H
FFFFF690H	TMM0 控制寄存器 0	TM0CTL0		√	√		00H
FFFFF694H	TMM0 比较寄存器 0	TM0CMP0				√	0000H
FFFFF6C0H	振荡稳定时间选择寄存器	OSTS			√		06H
FFFFF6C1H	PLL 入锁时间指定寄存器	PLLS			√		03H
FFFFF6D0H	看门狗定时器模式寄存器 2	WDTM2			√		67H
FFFFF6D1H	看门狗定时器使能寄存器	WDTE			√		9AH
FFFFF6E0H	实时输出缓冲寄存器 0L	RTBL0		√	√		00H
FFFFF6E2H	实时输出缓冲寄存器 0H	RTBH0		√	√		00H
FFFFF6E4H	实时输出端口模式寄存器 0	RTPM0		√	√		00H
FFFFF6E5H	实时输出端口控制寄存器 0	RTPC0		√	√		00H
FFFFF700H	端口 0 功能控制扩展寄存器	PFCE0		√	√		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

(8/11)

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFF712H	端口 9 功能控制扩展寄存器	PFCE9	R/W			√	0000H
FFFFF712H	端口 9 功能控制扩展寄存器 L	PFCE9L		√	√		00H
FFFFF713H	端口 9 功能控制扩展寄存器 H	PFCE9H		√	√		00H
FFFFF802H	系统状态寄存器	SYS		√	√		00H
FFFFF80CH	内部振荡模式寄存器	RCM		√	√		00H
FFFFF810H	DMA 触发因素寄存器 0	DTFR0		√	√		00H
FFFFF812H	DMA 触发因素寄存器 1	DTFR1		√	√		00H
FFFFF814H	DMA 触发因素寄存器 2	DTFR2		√	√		00H
FFFFF816H	DMA 触发因素寄存器 3	DTFR3		√	√		00H
FFFFF820H	省电模式寄存器	PSMR		√	√		00H
FFFFF822H	时钟控制寄存器	CKC		√	√		0AH
FFFFF824H	锁定寄存器	LOCKR	R	√	√		00H
FFFFF828H	处理器时钟控制寄存器	PCC 注	R/W	√	√		03H
FFFFF82CH	PLL 控制寄存器	PLLCTL		√	√		01H
FFFFF82EH	CPU 工作时钟状态寄存器	CCLS	R	√	√		00H
FFFFF870H	时钟监视器模式寄存器	CLM	R/W	√	√		00H
FFFFF888H	复位源标志寄存器	RESF		√	√		00H
FFFFF890H	低电压检测寄存器	LVIM		√	√		00H
FFFFF891H	低电压检测电平选择寄存器	LVIS			√		00H
FFFFF8B0H	预分频模式寄存器 0	PRSM0		√	√		00H
FFFFF8B1H	预分频器比较寄存器 0	PRSCM0			√		00H
FFFFF9FCH	片上调试模式寄存器	OCDM 注		√	√		01H
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

注 这是一个专用寄存器。

(9/11)

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFFA30H	UARTA3 控制寄存器 0	UA3CTL0	R/W	√	√		10H
FFFFFA31H	UARTA3 控制寄存器 1	UA3CTL1			√		00H
FFFFFA32H	UARTA3 控制寄存器 2	UA3CTL2			√		FFH
FFFFFA33H	UARTA3 选项控制寄存器 0	UA3OPT0		√	√		14H
FFFFFA34H	UARTA3 状态寄存器	UA3STR		√	√		00H
FFFFFA36H	UARTA3 接收数据寄存器	UA3RX	R		√		FFH
FFFFFA37H	UARTA3 发送数据寄存器	UA3TX	R/W		√		FFH
FFFFFA40H	UARTA4 控制寄存器 0	UA4CTL0		√	√		10H
FFFFFA41H	UARTA4 控制寄存器 1	UA4CTL1			√		00H
FFFFFA42H	UARTA4 控制寄存器 2	UA4CTL2			√		FFH
FFFFFA43H	UARTA4 选项控制寄存器 0	UA4OPT0		√	√		14H
FFFFFA44H	UARTA4 状态寄存器	UA4STR		√	√		00H
FFFFFA46H	UARTA4 接收数据寄存器	UA4RX	R		√		FFH
FFFFFA47H	UARTA4 发送数据寄存器	UA4TX	R/W		√		FFH
FFFFFA50H	UARTA5 控制寄存器 0	UA5CTL0		√	√		10H
FFFFFA51H	UARTA5 控制寄存器 1	UA5CTL1			√		00H
FFFFFA52H	UARTA5 控制寄存器 2	UA5CTL2			√		FFH
FFFFFA53H	UARTA5 选项控制寄存器 0	UA5OPT0		√	√		14H
FFFFFA54H	UARTA5 状态寄存器	UA5STR		√	√		00H
FFFFFA56H	UARTA5 接收数据寄存器	UA5RX	R		√		FFH
FFFFFA57H	UARTA5 发送数据寄存器	UA5TX	R/W		√		FFH
FFFFFAA0H	UARTC0 控制寄存器 0	UC0CTL0		√	√		10H
FFFFFAA1H	UARTC0 控制寄存器 1	UC0CTL1			√		00H
FFFFFAA2H	UARTC0 控制寄存器 2	UC0CTL2			√		FFH
FFFFFAA3H	UARTC0 选项控制寄存器 0	UC0OPT0		√	√		14H
FFFFFAA4H	UARTC0 状态寄存器	UC0STR		√	√		00H
FFFFFAA6H	UARTC0 接收数据寄存器	UC0RX	R			√	01FFH
FFFFFAA6H	UARTC0 接收数据寄存器 L	UC0RXL			√		FFH
FFFFFAA8H	UARTC0 发送数据寄存器	UC0TX	R/W			√	01FFH
FFFFFAA8H	UARTC0 发送数据寄存器 L	UC0TXL			√		FFH
FFFFFAAAH	UARTC0 选项控制寄存器 1	UC0OPT1		√	√		00H
FFFFFAD0H	副计数寄存器	RC1SUBC	R			√	0000H
FFFFFAD2H	秒计数寄存器	RC1SEC	R/W		√		00H
FFFFFAD3H	分钟计数寄存器	RC1MIN			√		00H
FFFFFAD4H	小时计数寄存器	RC1HOUR			√		12H
FFFFFAD5H	星期计数寄存器	RC1WEEK			√		00H
FFFFFAD6H	日计数寄存器	RC1DAY			√		01H
FFFFFAD7H	月数寄存器	RC1MONTH			√		01H
FFFFFAD8H	年计数寄存器	RC1YEAR			√		00H
FFFFFAD9H	时间误差修正寄存器	RC1SUBU			√		00H
FFFFFADAH	闹钟分钟设置寄存器	RC1ALM			√		00H
FFFFFADBH	闹钟时间设置寄存器	RC1ALH			√		12H
FFFFFADCH	闹钟星期设置寄存器	RC1ALW		√	√		00H

(10/11)

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
FFFFFADDH	RTC 控制寄存器 0	RC1CC0	R/W	√	√		00H
FFFFFADEH	RTC 控制寄存器 1	RC1CC1		√	√		00H
FFFFFADFH	RTC 控制寄存器 2	RC1CC2		√	√		00H
FFFFFAE0H	RTC 控制寄存器 3	RC1CC3		√	√		00H
FFFFFB00H	RTC 备份控制寄存器 0	RTCBUMCTL0 注		√	√		00H
FFFFFB03H	副时钟低压操作控制寄存器	SOSCAMCTL 注		√	√		00H
FFFFFC00H	外部中断下降沿指定寄存器 0	INTF0		√	√		00H
FFFFFC06H	外部中断下降沿指定寄存器 3	INTF3		√	√		00H
FFFFFC13H	外部中断下降沿指定寄存器 9H	INTF9H		√	√		00H
FFFFFC20H	外部中断上升沿指定寄存器 0	INTR0		√	√		00H
FFFFFC26H	外部中断上升沿指定寄存器 3	INTR3		√	√		00H
FFFFFC33H	外部中断上升沿指定寄存器 9H	INTR9H		√	√		00H
FFFFFC60H	端口 0 功能寄存器	PF0		√	√		00H
FFFFFC66H	端口 3 功能寄存器	PF3				√	0000H
FFFFFC66H	端口 3 功能寄存器 L	PF3L		√	√		00H
FFFFFC67H	端口 3 功能寄存器 H	PF3H		√	√		00H
FFFFFC68H	端口 4 功能寄存器	PF4		√	√		00H
FFFFFC6AH	端口 5 功能寄存器	PF5		√	√		00H
FFFFFC72H	端口 9 功能寄存器	PF9				√	0000H
FFFFFC72H	端口 9 功能寄存器 L	PF9L		√	√		00H
FFFFFC73H	端口 9 功能寄存器 H	PF9H		√	√		00H
FFFFFD00H	CSIB0 控制寄存器 0	CB0CTL0		√	√		01H
FFFFFD01H	CSIB0 控制寄存器 1	CB0CTL1		√	√		00H
FFFFFD02H	CSIB0 控制寄存器 2	CB0CTL2			√		00H
FFFFFD03H	CSIB0 状态寄存器	CB0STR		√	√		00H
FFFFFD04H	CSIB0 接收数据寄存器	CB0RX	R			√	0000H
FFFFFD04H	CSIB0 接收数据寄存器 L	CB0RXL			√		00H
FFFFFD06H	CSIB0 发送数据寄存器	CB0TX	R/W			√	0000H
FFFFFD06H	CSIB0 发送数据寄存器 L	CB0TXL			√		00H
FFFFFD10H	CSIB1 控制寄存器 0	CB1CTL0		√	√		01H
FFFFFD11H	CSIB1 控制寄存器 1	CB1CTL1		√	√		00H
FFFFFD12H	CSIB1 控制寄存器 2	CB1CTL2			√		00H
FFFFFD13H	CSIB1 状态寄存器	CB1STR		√	√		00H
FFFFFD14H	CSIB1 接收数据寄存器	CB1RX	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		√	√		01H
FFFFFD21H	CSIB2 控制寄存器 1	CB2CTL1		√	√		00H
FFFFFD22H	CSIB2 控制寄存器 2	CB2CTL2			√		00H
FFFFFD23H	CSIB2 状态寄存器	CB2STR		√	√		00H
FFFFFD24H	CSIB2 接收数据寄存器	CB2RX	R			√	0000H
FFFFFD24H	CSIB2 接收数据寄存器 L	CB2RXL			√		00H

注 这是一个专用寄存器。

(11/11)

地址	功能寄存器名称	符号	R/W	操作位数			默认值
				1	8	16	
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		√	√		01H
FFFFFD31H	CSIB3 控制寄存器 1	CB3CTL1		√	√		00H
FFFFFD32H	CSIB3 控制寄存器 2	CB3CTL2			√		00H
FFFFFD33H	CSIB3 状态寄存器	CB3STR		√	√		00H
FFFFFD34H	CSIB3 接收数据寄存器	CB3RX	R			√	0000H
FFFFFD34H	CSIB3 接收数据寄存器 L	CB3RXL			√		00H
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	R/W	√	√		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			√		00H
FFFFFD82H	IIC 控制寄存器 0	IICC0		√	√		00H
FFFFFD83H	从地址寄存器 0	SVA0			√		00H
FFFFFD84H	IIC 时钟选择寄存器 0	IICCL0		√	√		00H
FFFFFD85H	IIC 功能扩展寄存器 0	IICX0		√	√		00H
FFFFFD86H	IIC 状态寄存器 0	IICS0	R	√	√		00H
FFFFFD8AH	IIC 标志寄存器 0	IICF0	R/W	√	√		00H
FFFFFD90H	IIC 移位寄存器 1	IIC1			√		00H
FFFFFD92H	IIC 控制寄存器 1	IICC1		√	√		00H
FFFFFD93H	从地址寄存器 1	SVA1			√		00H
FFFFFD94H	IIC 时钟选择寄存器 1	IICCL1		√	√		00H
FFFFFD95H	IIC 功能扩展寄存器 1	IICX1		√	√		00H
FFFFFD96H	IIC 状态寄存器 1	IICS1	R	√	√		00H
FFFFFD9AH	IIC 标志寄存器 1	IICF1	R/W	√	√		00H
FFFFFDA0H	IIC 移位寄存器 2	IIC2			√		00H
FFFFFDA2H	IIC 控制寄存器 2	IICC2		√	√		00H
FFFFFDA3H	从地址寄存器 2	SVA2			√		00H
FFFFFDA4H	IIC 时钟选择寄存器 2	IICCL2		√	√		00H
FFFFFDA5H	IIC 功能扩展寄存器 2	IICX2		√	√		00H
FFFFFDA6H	IIC 状态寄存器 2	IICS2	R	√	√		00H
FFFFFDAAH	IIC 标志寄存器 2	IICF2	R/W	√	√		00H
FFFFFF40H	USB 时钟控制寄存器	UCKSEL		√	√		00H
FFFFFF41H	USB 功能控制寄存器	UFCKMSK		√	√		03H

3.4.7 专用寄存器

专用寄存器是受到保护的寄存器，可以防止程序循环引起的数据非法写入。V850ES/JG3-L 具有以下 9 个专用寄存器。

- 节电控制寄存器（PSC）
- 时钟控制寄存器（CKC）
- 处理器时钟控制寄存器 （PCC）
- 时钟监控模式寄存器（CLM）
- 复位源标志寄存器（RESF）
- 低电压检测寄存器（LVIM）
- 片上调试模式寄存器 （OCDM）
- RTC 备份控制寄存器 0（RTCBUMCTL0）
- 副时钟低电压操作控制寄存器 0（SOSCAMCTL）

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

（1）设置专用寄存器数据

按照以下流程设置专用寄存器数据。

<1> 禁止 DMA 操作
<2> 需要写入专用寄存器的数据放入通用寄存器中。
<3> 将步骤<2>准备好的数据写入 PRCMD 寄存器中。
<4> 将设置数据写入专用寄存器中（使用如下指令完成）。
- 存储指令（ST/SST 指令）
- 位操作指令 （SET1/CLR1/NOT1 指令）。

（<5> 至 <9> 插入 NOP 指令（5 条））注
<10> 如果必要，则使能 DMA 操作。

注 当切换到 IDLE 模式或 STOP 模式（PSC.STP 位 = 1）时，在执行切换之后必须立即插入 5 条 NOP 指令。

注意事项 在特定序列之后，要恢复到 DMA 操作被禁止之前的 DMA 操作状态，在禁止 DMA 操作之前必须存储 DCHCn 寄存器状态。
在保存了 DCHCn 寄存器状态之后，恢复 DMA 操作之前必须检查 DCHCn.TCn 位，并且必须根据 TCn 位的状态执行下列处理，因为在 DMA 操作被禁止之前可能正好有 DMA 传输完成。

当 TCn 位为 0 时（DMA 传输未完成），在禁止 DMA 操作之前保存的 DCHCn 寄存器内容被重新写入到 DCHCn 寄存器。
当 TCn 位为 1 时（DMA 传输已完成），执行 DMA 传输完成处理。

备注 n = 0 至 3

[实例] 使用 PSC 寄存器（设置待机模式）

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

读取专用寄存器时没有特定序列。

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 **执行存储指令来对命令寄存器进行写操作时，不响应中断。原因是假设上述步骤<3>和<4>是通过连续的存储指令执行。若在步骤<3>和<4>之间插入了其他指令，且该指令又引起了中断响应，则将打乱上述的顺序，导致错误操作。**

备注 虽然写入 PRCMD 寄存器的是伪数据，仍然使用在设置专用寄存器（示例中的步骤<4>）时所用的通用寄存器将数据写入至 PRCMD 寄存器（示例中的步骤<3>）。当通用寄存器用于寻址时，同样使用这种用法。

（2）命令寄存器（PRCMD）

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

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

复位后: 不确定　W　地址: FFFFF1FCH

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

（3）系统状态寄存器（SYS）

该寄存器包含了整个系统的操作状态的状态标志。

该寄存器可进行字节读写或者按位读写。

系统复位后，该寄存器的值被置为 00H。

复位后: 00H　　R/W　　地址: FFFFF802H

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

PRERR	检测保护错误
0	没有发生保护错误
1	发生保护错误

PRERR 标志的操作是按下列条件进行的。

（a）置位条件（PRERR 标志=1）

（i） 当数据写入专用寄存器时，却没有任何内容写入 PRCMD 寄存器，（没有执行 **3.4.7（1） 设置数据至专用寄存器**的步骤 **<3>**，就执行步骤 **<4>**）

（ii） 对 PRCMD 寄存器写入数据后，数据写入了片上周边 I/O 寄存器而没有写入专用寄存器（如果 **3.4.7（1）设置数据至专用寄存器**中的步骤<4>不是对专用寄存器的设置）

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

（b）清零条件（PRERR 标志=0）

（i） 当将 0 写入 PRERR 标志位时。

（ii） 系统被复位时。

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

2. 如果在对 PRCMD 寄存器进行写访问之后，立即将数据写入 PRCMD 寄存器（该寄存器并不是专用寄存器），那么 PRERR 位被设为 1。

3.4.8 需要先行设置的寄存器

当使用 V850ES/JG3-L 单片机时，必需确保首先对下列寄存器进行设置。

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

设置 VSWC、OCDM 和 WDTM2 寄存器后，再按照具体需求对其它寄存器进行设置。
使用外部总线时，在上述寄存器设置完成之后，通过端口相关寄存器将每个引脚设置为复用功能总线控制引脚模式。

（a）系统等待控制寄存器（VSWC）

VSWC 寄存器控制访问片上外围 I/O 寄存器时的总线等待时间。
访问片上外围 I/O 寄存器（没有等待周期）需要三个时钟周期。根据工作频率，V850ES/JG3-L 需要不同的等待周期。按照下表设置不同工作频率下的 VSWC 寄存器值。
该寄存器按字节进行读取或写入。
系统复位后，该寄存器被置为 77H（等待的数量：14）。

复位后：77H　R/W　地址：FFFFF06EH

	7	6	5	4	3	2	1	0
VSWC								

工作频率（f_{CPU}）	VSWC 的设置值	等待周期数量
32 kHz ≤ f_{CLK} ≤ 16.6 MHz	00H	0 （无等待）
16.6 MHz ≤ f_{CLK} ≤ 20 MHz	01H	1

（b）片上调试模式寄存器（OCDM）

详情参见**第 31 章 片上调试功能**。

（c）看门狗定时器模式寄存器 2（WDTM2）

WDTM2 寄存器用以设置看门狗定时器 2 的溢出时间和工作时钟。
复位解除后，在复位模式下看门狗定时器 2 会自动启动。我指定看门狗定时器 2 的操作，在复位解除后写入 WDTM2 寄存器。
详情参见**第 12 章 看门狗定时器 2 的功能。**

3.4.9 注意事项

（1）访问特殊片上外围 I/O 寄存器

本产品有两种类型的内部系统总线。

一种是 CPU 总线，另外一种是用来与低速外围硬件接口的外围总线。

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

当执行实时处理任务时，需要特别注意这种情况。

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

访问条件以及如何计算需要插入的等待状态数量（CPU 时钟数）如下表所示：

表 3-3. 需要等待的寄存器

外围功能	寄存器名称	访问类型	k
16 定时器/事件计数器 P（TMP）（n = 0 至 5）	TPnCNT	读取	1 或 2
	TPnCCR0，TPnCCR1	写入	• 第一次访问：无等待 连续写入： 3 或 4
		读取	1 或 2
16 位定时器/事件计数器 Q（TMQ）	TQ0CNT	读取	1 或 2
	TQ0CCR0 至 TQ0CCR3	写入	• 第一次访问：无等待 • 连续写入： 3 或 4
		读取	1 或 2
看门狗定时器 2 （WDT2）	WDTM2	写入 （ WDT2 工作时）	3
实时输出功能（RTO）	RTBL0， RTBH0	写入 （RTPC0.RTPOE0 位 = 0 时）	1
A/D 转换器	ADA0M0	读取	1 或 2
	ADA0CR0 至 ADA0CR11	读取	1 或 2
	ADA0CR0H 至 ADA0CR11H	读取	1 或 2
I^2C00 至 I^2C02	IICS0 至 IICS2	读取	1
CRC	CRCD	写入	1

访问时必需的时钟数目 = 3 + i + j + （2 + j） × k

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

•当 CPU 运行于副时钟，且主时钟振荡停止时。

•当 CPU 运行于内部振荡时钟时。

备注 i： VSWC 寄存器高 4 位的值（0）

j： VSWC 寄存器低 4 位的值（0 或 1）

（2）sld 指令与中断请求之间的冲突

（a）描述

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

指令 <1>

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

指令<2>

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

<举例>

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

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

（b）对策

<1> 当使用编译器（CA850）时
请使用 CA850 Ver. 2.61 或更高版本，因为它将自动禁止相应指令序列的产生。

<2> 当使用汇编器时
在指令<ii>之后就立即执行 sld 指令，可以采用以下任一种方法来避免上述操作。

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

第四章 端口功能

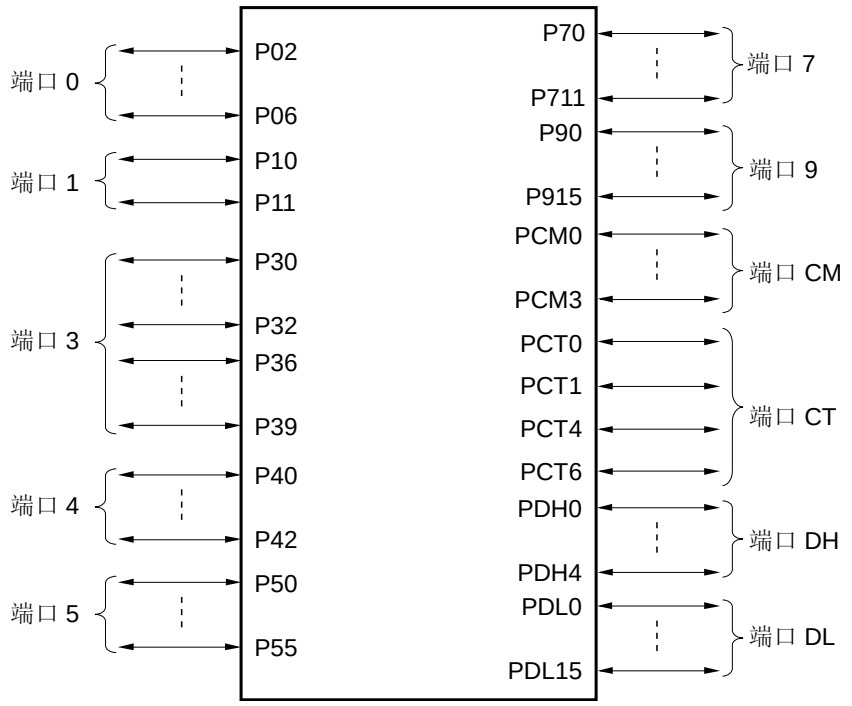
4.1 特性

- I/O 端口： 80
  - 可以选择为 N 沟道漏极开路输出：37（5 V 耐压：28）
- 可以按位指定输入/输出

4.2 端口基本配置

V850ES/JG3-L 总共包括 80 个输入/输出端口引脚，由端口 0，1，3 至 5，7，9，CM，CT，DH 以及 DL 组成。端口配置如下所示：

图 4-1. 端口配置图



注意事项 端口 0，3 至 5 和 9（P90 至 P96）是 5 V 耐压。

表 4-1. 引脚的输入/输出缓冲器电源供应

供电	对应引脚
AV_{REF0}	端口 7
AV_{REF1}	端口 1
EV_{DD}	$\overline{RESET}$，端口 0，3 至 5，9，CM，CT，DH，DL

4.3 端口配置

端口包括下列硬件。

表 4-2. 端口配置

项目	配置
控制寄存器	端口 n 模式寄存器（PMn： n = 0，1，3 至 5，7，9，CD，CM，CT，DH，DL） 端口 n 模式控制寄存器（PMCn： n = 0，3 至 5，9，CM，CT，DH，DL） 端口 n 功能控制寄存器（PFCn： n = 0，3 至 5，9） 端口 n 功能控制扩展寄存器（PFCEn： n = 0，3，5，9） 端口 n 功能寄存器（PFn： n = 0，3 至 5，9）
端口引脚	I/O：80

（1）端口 n 寄存器（Pn）

通过写入或读取 Pn 寄存器，从外部器件输入数据，或输出数据到外部器件。Pn 寄存器由一个保持输出数据的端口锁存器和一个读取引脚状态的电路组成。

Pn 寄存器的每一位对应端口的一个引脚，而且可以按位读出或写入。

复位后： 00H[注]（输出锁存） R/W

	7	6	5	7	3	2	1	0
Pn	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0

Pnm	输出数据的控制 (在输出模式下)
0	输出0
1	输出1

注 对于只能用作输入的端口，该值是不确定的。

根据指定的模式，Pn 寄存器的写入或读取操作不同。

表 4-3. Pn 寄存器的写入/读出

PMCn 寄存器的设置	PMn 寄存器的设置	写入 Pn 寄存器	读取 Pn 寄存器
端口模式 (PMCnm 位 = 0)	输出模式 （PMnm 位 = 0）	数据写入输出锁存器[注]。 输出锁存器的内容从相应引脚输出。	读取得到输出锁存器的值。
	输入模式 （PMnm = 1）	数据写入输出锁存器[注]。 引脚状态不受影响。	读取得到引脚状态。
复用功能模式 (PMCnm 位 = 1)	输出模式 （PMnm 位 = 0）	数据写入输出锁存器[注]。 引脚状态不受影响。 该引脚作为复用功能引脚工作。	读取得到输出锁存器的值。
	输入模式 （PMnm = 1）	数据写入输出锁存器[注]。 引脚状态不受影响。 该引脚作为复用功能引脚工作。	读取得到引脚状态。

注 写入到输出锁存器的值会一直保留，直到有新的数值写入输出锁存器。
复位会导致输出锁存器的值被清除。

（2）端口 n 模式寄存器（PMn）

PMn 寄存器用于指定对应端口引脚的输入或输出模式。

PMn 寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定为输入或输出模式。

复位后: FFH　　R/W

	7	6	5	4	3	2	1	0
PMn	PMn7	PMn6	PMn5	PMn4	PMn3	PMn2	PMn1	PMn0

PMnm	输入/输出模式控制
0	输出模式
1	输入模式

（3）端口 n 模式控制寄存器（PMCn）

如果需要切换端口功能和复用功能，通过该寄存器指定端口模式或复用功能模式。

PMCn 寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定端口的工作模式。

复位后: 00H　　R/W

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

PMCnm	操作模式说明
0	端口模式
1	复用功能模式

（4）端口 n 功能 控制寄存器（PFCn）

如果一个引脚有两个或更多复用功能，PFCn 寄存器用来指定需要使用的引脚复用功能。

PFCn 寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定端口引脚的复用功能。

复位后: 00H　　R/W

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

PFCnm	复用功能说明
0	复用功能 1
1	复用功能 2

（5）端口 n 功能控制扩展寄存器（PFCEn）

如果引脚有三个或更多复用功能，PFCEn 寄存器用来和 PFCn 寄存器组合指定端口引脚的复用功能。

PFCEn 寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定端口引脚的复用功能。

复位后: 00H　　R/W

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

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

PFCEnm	PFCnm	复用功能说明
0	0	复用功能 1
0	1	复用功能 2
1	0	复用功能 3
1	1	复用功能 4

（6）端口 n 功能寄存器（PFn）

PFn 寄存器用于指定端口是正常输出（CMOS 输出）或 N 沟道漏极开路输出。

PFn 寄存器的每一位对应端口 n 的一个引脚，而且可以按位设定。

复位后: 00H　　R/W

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

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

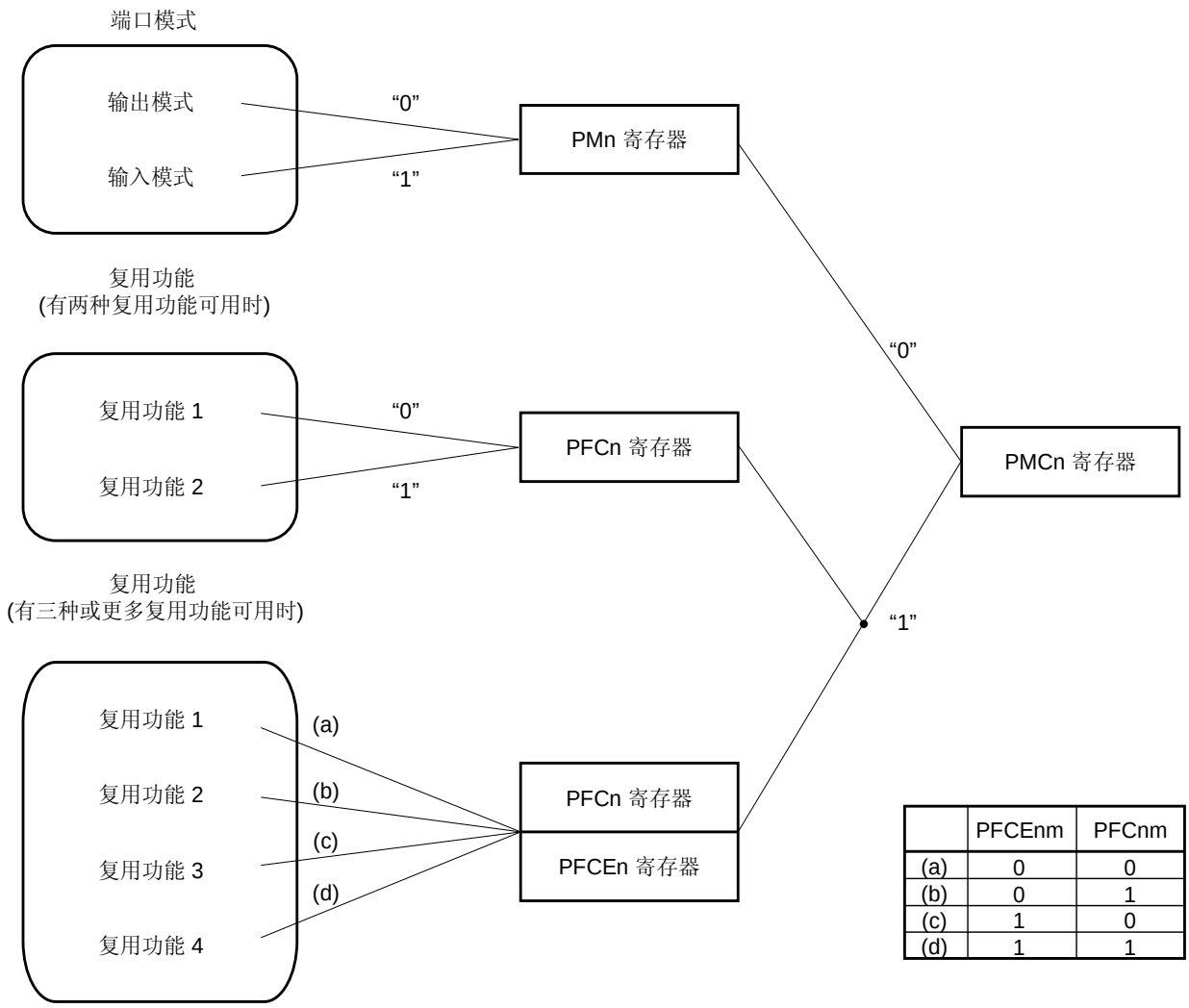
注　不论 PMCn 寄存器的设置如何，只有当 PMn.PMnm 位被设置为 0（输出模式）时，PFnm 位才有效。如果 PMnm 位被设置为 1（输入模式），为 PFn 寄存器指定的值无效。

示例　<1> PFn 寄存器值有效：
PFnm 位 = 1 … 指定为 N 沟道漏极开路输出
PMnm 位 = 0 … 指定为输出模式
PMCnm 位 = 任意值
<2> PFn 寄存器值有效：
PFnm 位 = 1 … 指定为 N 沟道漏极开路输出
PMnm 位 = 1 … 指定为输入模式
PMCnm 位 = 任意值

（7）端口设置

端口设置如下所示：

图 4-2.各寄存器的设置和引脚功能



	PFCEnm	PFCnm
(a)	0	0
(b)	0	1
(c)	1	0
(d)	1	1

备注 按下列步骤设置复用功能：

<1>设置 PFCn 和 PFCEn 寄存器。
<2> 设置 PMCn 寄存器。
<3> 设置 INTRn 或 INTFn 寄存器（用来指定外部中断引脚）。

如果先设置 PMCn 寄存器，然后进行 PFCn 和 PFCEn 寄存器的设置，结果可能不是用户预期的功能。

4.3.1 端口 0

端口 0 有 5 位输入输出端口，可以按位控制输入输出设置。

端口 0 包括以下复用功能引脚：

表 4-4. 端口 0 复用功能引脚

引脚编号		功能名称	复用功能		备注	框图类型
GC	F1		名称	输入/输出		
7	G4	P02	NMI/A21	输入/输出	可以选择作为 N 沟道漏极开路输出	N-2
18	G3	P03	INTP0/ADTRG/UCLK/RTC1HZ	输入/输出		U-17
19	H4	P04	INTP1/RTCDIV/RTCCL	输入/输出		N-2
20	J3	P05	INTP2/$\overline{\text{DRST}}$ 注	输入		AA-1
21	J4	P06	INTP3	输入		L-1

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

注意事项 在复用功能引脚的输入模式下，P02 至 P06 具有迟滞特性，但端口模式没有迟滞特性。

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

（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 (输出模式下) (n = 2 至 6)
0	输出 0
1	输出 1

（2）端口 0 模式寄存器（PM0）

复位后: FFH　R/W　地址: FFFFF420H

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

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

（3）端口 0 模式控制寄存器（PMC0）

复位后：00H　R/W　地址：FFFFF440H

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

PMC06	引脚操作的说明
0	I/O 端口 (p06)
1	INTP3 输入

PMC05	引脚操作的说明
0	I/O 端口 (p05)
1	INTP2 输入

PMC04	引脚操作的说明
0	I/O 端口 (p04)
1	INTP1 输入/RTCDIV 输出/RTCCL 输出

PMC03	引脚操作的说明
0	I/O 端口 (p03)
1	INTP0 输入/ADTRG 输入/UCLK 输入/RTC1HZ 输出

PMC02	引脚操作的说明
0	I/O 端口 (p02)
1	NMI 输入/A21 输出

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

（4）端口 0 功能控制寄存器（PFC0）

复位后：00H　R/W　地址：FFFFF460H

	7	6	5	4	3	2	1	0
PFC0	0	0	0	PFC04	PFC03	PFC02	0	0

备注　关于复用功能规范的详情，参见 **4.3.1 (6) 端口 0 复用功能说明。**

（5）端口 0 功能控制扩展寄存器（PFCE0）

复位后：00H　R/W　地址：FFFFF700H

	7	6	5	4	3	2	1	0
PFCE0	0	0	0	PFCE04	PFCE03	0	0	0

备注　关于复用功能规范的详情，参见 **4.3.1 (6) 端口 0 复用功能说明。**

（6）端口 0 复用功能说明

PFCE04	PFC04	P04 引脚复用功能的说明
0	0	INTP1 输入
0	1	RTCDIV 输出
1	0	RTCCL 输出
1	1	禁止设置

PFCE03	PFC03	P03 引脚复用功能的说明
0	0	INTP0 输入
0	1	ADTRG 输入
1	0	UCLK 输入
1	1	RTC1HZ 输出

PFC02	P02 引脚复用功能的说明
0	NMI 输入
1	A21 输出

（7）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	正常输出或 N沟道漏极开路输出 控制(n = 2 至 6)
0	正常输出 (CMOS 输出)
1	N沟道漏极开路输出

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

4.3.2 端口 1

端口 1 有 2 位输入输出端口，可以按位控制输入输出设置。

端口 1 包括以下复用功能引脚：

表 4-5.端口 1 复用功能引脚

引脚编号		功能名称	复用功能		备注	框图类型
GC	F1		名称	输入/输出		
3	E3	P10	ANO0	输出	–	A-2
4	E4	P11	ANO1	输出	–	A-2

注意事项 电源打开后的复位期间，P10 和 P11 引脚可能会暂时输出不确定电平。

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

（1）端口 1 寄存器（P1）

复位后: 00H (输出锁存) R/W 地址: FFFFF402H

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

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

注意事项 D/A 转换期间，不要读取或写入 P1 寄存器（参见 15.4.3 注意事项）。

（2）端口 1 模式寄存器（PM1）

复位后: FFH　R/W　地址: FFFFF422H

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

PM1n	I/O 模式控制（n=0,1）
0	输出模式
1	输入模式

注意事项 **1.** P1n 用作复用功能（ANOn 引脚输出）时，指定为输入模式（PM1n 位 = 1）。

2. 当使用 P10 和 P11 引脚之一作为 I/O 端口，而另一个用作 D/A 输出引脚。在 D/A 输出期间，如果端口 I/O 电平不改变，则在应用系统中可以这样使用。

4.3.3 端口 3

端口 3 有 7 位输入输出端口，可以按位控制输入输出设置。

端口 3 包括以下复用功能引脚：

表 4-6. 端口 3 复用功能引脚

引脚编号		功能名称	复用功能		备注	框图类型
GC	F1		名称	输入/输出		
25	L3	P30	TXDA0/SOB4	输出	可以选择作为 N 沟道漏极开路输出	G-3
26	K3	P31	RXDA0/INTP7/SIB4	输入		N-3
27	L4	P32	ASCKA0/$\overline{\text{SCKB4}}$/TIP00/TOP00	输入/输出		U-1
31	H5	P36	TXDA3	输出		G-2
32	J6	P37	RXDA3	输入		G-2
35	H6	P38	TXDA2/SDA00	输入/输出		G-12
36	H7	P39	RXDA2/SCL00	输入/输出		G-6

注意事项 在复用功能引脚的输入模式下，P31、P32、P37 至 P39 具有迟滞特性，但端口模式没有迟滞特性。

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

（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	0	0	0	P32	P31	P30

P3n	输出数据控制 (输出模式下) (n = 0至2，6至9)
0	输出 0
1	输出 1

注意事项　请确保将第 15 位至第 10 位，第 5 位至第 3 位设置为“0”。

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

（2）端口 3 模式寄存器（PM3）

复位后：FFFFH　R/W　地址：　PM3　FFFFF426H,
PM3L FFFFF426H, PM3H FFFFF427H

	15	14	13	12	11	10	9	8
PM3 (PM3H)	1	1	1	1	1	1	PM39	PM38
	7	6	5	4	3	2	1	0
(PM3L)	PM37	PM36	1	1	1	PM32	PM31	PM30

PM3n	I/O 模式控制 (n = 0至2，6至9)
0	输出模式
1	输入模式

注意事项　请确保将第 15 位至第 10 位，第 5 位至第 3 位设置为“1”。

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

（3）端口 3 模式控制寄存器（PMC3）

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

	15	14	13	12	11	10	9	8
PMC3 (PMC3H)	0	0	0	0	0	0	PMC39	PMC38

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

PMC39	引脚操作的说明
0	I/O 端口 (P39)
1	RXDA2 输入/SCL00 I/O

PMC38	引脚操作的说明
0	I/O 端口 (P38)
1	TXDA2 输出/SDA00 I/O

PMC37	引脚操作的说明
0	I/O 端口 (P37)
1	RXDA3 输入

PMC36	引脚操作的说明
0	I/O 端口 (P36)
1	TXDA3 输出

PMC32	引脚操作的说明
0	I/O 端口 (P32)
1	ASCKA0 输入/$\overline{\text{SCKB4}}$ I/O/TIP00 输入/TOP00 输出

PMC31	引脚操作的说明
0	I/O 端口 (P31)
1	RXDA0 输入/SIB4 输入/INTP7 输入

PMC30	引脚操作的说明
0	I/O 端口 (P30)
1	TXDA0 输出/SOB4 输出

注意事项　请确保将第 15 位至第 10 位，第 5 位至第 3 位设置为“0”。

备注

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

（4）端口 3 功能控制寄存器（PFC3）

复位后： 0000H R/W 地址： PFC3 FFFFF466H,
PFC3L FFFFF466H, PFC3H FFFFF467H

	15	14	13	12	11	10	9	8
PFC3 (PFC3H)	0	0	0	0	0	0	PFC39	PFC38
	7	6	5	4	3	2	1	0
(PFC3L)	0	0	0	0	0	PFC32	PFC31	PFC30

注意事项 请确保将第 15 位至第 10 位，第 7 位至第 3 位设置为“0”。

备注
1. 关于复用功能规范的详情，参见 **4.3.3（6） 端口 3 复用功能规范。**
2. PFC3 寄存器可以按 16 位宽度进行读写。
   但是当寄存器 PFC3 的高 8 位和低 8 位分别用作 PFC3H 寄存器和 PFC3L 寄存器时，PFC3 可以按字节或按位读出或写入。
3. 若要按字节或按位读取或写入 PFC3 的第 8 位至第 15 位，将这些位作为 PFC3H 寄存器的第 0 位至第 7 位。

（5）端口 3 功能控制扩展寄存器 L（PFCE3L）

复位后: 00H R/W 地址: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3L	0	0	0	0	0	PFCE32	0	0

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

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

（6）端口 3 复用功能规范

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

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

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

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

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

注 INTP7 和 RXDA0 是复用功能。当该引脚用作 RXDA0 引脚时，禁止对 INTP7 复用功能边沿检测。（将 INTF3.INTF31 位和 INTR3.INTR31 位清为 0。）当用作 INTP7 引脚，停止 UARTA0 接收。（UA0CTL0.UA0RXE 位清 0。）

（7）端口 3 功能寄存器（PF3）

复位后： 0000H R/W 地址： PF3 FFFFFC66H,
PF3L FFFFFC66H, PF3H FFFFFC67H

	15	14	13	12	11	10	9	8
PF3 (PF3H)	0	0	0	0	0	0	PF39	PF38

	7	6	5	4	3	2	1	0
(PF3L)	PF37	PF36	0	0	0	PF32	PF31	PF30

PF3n	正常输出 (CMOS 输出) 或 N沟道开漏输出的说明 (n = 0 至 2，6至 9)
0	正常输出 (CMOS 输出)
1	N沟道开漏输出

注意事项 1. 若某个输出引脚被上拉至 EV_{DD} 或更高电压时，确保将 PF3n 位设置为 1。
2. 请确保将第 15 位至第 10 位，第 5 位至第 3 位设置为“0”。

备注 1. PF3 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PF3 的高 8 位和低 8 位分别用作 PF3H 寄存器和 PF3L 寄存器时，PF3 可以按字节或按位读出或写入。
2. 若要按字节或按位读取或写入 PF3 的第 8 位至第 15 位，将这些位作为 PF3H 寄存器的第 0 位至第 7 位。

4.3.4 端口 4

端口 4 有 3 位输入输出端口，可以按位控制输入输出设置。

端口 4 包括以下复用功能引脚：

表 4-7. 端口 4 复用功能引脚

引脚编号		功能名称	复用功能		备注	框图类型
GC	F1		名称	输入/输出		
22	K1	P40	SIB0/SDA01	输入/输出	可以选择作为 N 沟道漏极开路输出	G-6
23	K2	P41	SOB0/SCL01	输入/输出		G-12
24	L2	P42	$\overline{\text{SCKB0}}$	输入/输出		E-3

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

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

（1）端口 4 寄存器（P4）

复位后：00H (输出锁存) R/W 地址：FFFFF408H

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

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

（2）端口 4 模式寄存器（PM4）

复位后：FFH R/W 地址：FFFFF428H

	7	6	5	4	3	2	1	0
PM4	1	1	1	1	1	PM42	PM41	PM40

PM4n	I/O 模式控制(n = 0 至 2)
0	输出模式
1	输入模式

（3）端口 4 模式控制寄存器（PMC4）

复位后: 00H　R/W　地址: FFFFF448H

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

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

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

PMC40	P40 引脚操作模式说明
0	I/O 端口
1	SIB0 输入/SDA01 I/O

（4）端口 4 功能控制寄存器（PFC4）

复位后: 00H　R/W　地址: FFFFF468H

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

PFC41	P41 引脚复用功能说明
0	SOB0 输出
1	SCL01 I/O

PFC40	P40 引脚复用功能说明
0	SIB0 输入
1	SDA01 I/O

（5）端口 4 功能寄存器（PF4）

复位后: 00H　R/W　地址: FFFFFC68H

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

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

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

4.3.5 端口 5

端口 5 有 6 位输入输出端口，可以按位控制输入输出设置。

端口 5 包括以下复用功能引脚：

表 4-8. 端口 5 复用功能引脚

引脚编号		功能名称	复用功能		备注	框图类型
GC	F1		名称	输入/输出		
37	L7	P50	TIQ01/KR0/TOQ01/RTP00	输入/输出	可以选择作为 N 沟道漏极开路输出	U-5
38	K7	P51	TIQ02/KR1/TOQ02/RTP01	输入/输出		U-5
39	J7	P52	TIQ03/KR2/TOQ03/RTP02/DDI 注	输入/输出		U-6
40	L8	P53	SIB2/KR3/TIQ00/TOQ00/RTP03/DDO 注	输入/输出		U-7
41	K8	P54	SOB2/KR4/RTP04/DCK 注	输入/输出		U-8
42	J8	P55	$\overline{\text{SCKB2}}$/KR5/RTP05/DMS 注	输入/输出		U-9

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

注意事项 1. 电源打开后的复位期间，P53 引脚可能暂时输出不确定电平。
2. 在复用功能引脚输入模式下，P50 至 P55 具有迟滞特性，但在端口模式下没有迟滞特性。

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

（1）端口 5 寄存器（P5）

复位后：00H (输出锁存)　R/W　地址：FFFFF40AH

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

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

（2）端口 5 模式寄存器（PM5）

复位后: FFH　R/W　地址: FFFFF42AH

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

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

（3）端口 5 模式控制寄存器（PMC5）

复位后: 00H　R/W　地址: FFFFF44AH

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

PMC55	P55 引脚操作模式说明
0	I/O 端口
1	$\overline{\text{SCKB2}}$ I/O/KR5 输入/RTP05 输出

PMC54	P54 引脚操作模式说明
0	I/O 端口
1	SOB2 输出/KR4 输入/RTP04 输出

PMC53	P53 引脚操作模式说明
0	I/O 端口
1	SIB2 输入/KR3 输入/TIQ00 输入/TOQ00 输出/RTP03 输出

PMC52	P52 引脚操作模式说明
0	I/O 端口
1	TIQ03 输入/KR2 输入/TOQ03 输出/RTP02 输出

PMC51	P51 引脚操作模式说明
0	I/O 端口
1	TIQ02 输入/KR1 输入/TOQ02 输出/RTP01 输出

PMC50	P50 引脚操作模式说明
0	I/O 端口
1	TIQ01 输入/KR0 输入/TOQ01 输出/RTP00 输出

（4）端口 5 功能控制寄存器（PFC5）

复位后：00H　R/W　地址：FFFFF46AH

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

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

（5）端口 5 功能控制扩展寄存器（PFCE5）

复位后：00H　R/W　地址：FFFFF70AH

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

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

（6）端口 5 复用功能规范

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

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

PFCE53	PFC53	P53 引脚复用功能说明
0	0	SIB2 输入
0	1	TIQ00 输入/KR3 注 输入
1	0	TOQ00 输出
1	1	RTP03 输出

PFCE52	PFC52	P52 引脚复用功能说明
0	0	禁止设置
0	1	TIQ03 输入/KR2 注 输入
1	0	TOQ03 输入
1	1	RTP02 输出

PFCE51	PFC51	P51 引脚复用功能说明
0	0	禁止设置
0	1	TIQ02 输入/KR1 注 输入
1	0	TOQ02 输出
1	1	RTP01 输出

PFCE50	PFC50	P50 引脚复用功能说明
0	0	禁止设置
0	1	TIQ01 输入/KR0 注 输入
1	0	TOQ01 输出
1	1	RTP00 输出

注 KRn 和 TIQ0m 是复用功能。当该引脚用作 TIQ0m 引脚时，禁止 KRn 的按键返回检测功能（将 KRM.KRMn 位清除为 0）。同样，当该引脚用作 KRn 引脚时，禁止 TIQ0m 的边沿检测功能（n = 0 至 3，m = 0 至 3）。

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

（7）端口 5 功能寄存器（PF5）

复位后: 00H　R/W　地址: FFFFFC6AH

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

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

注意事项 1. 若某个输出引脚被上拉至 EV_{DD} 或更高电压时，确保将 PF5n 位设置为 1。
2. 请确保将 PF5 寄存器的第 6 位和第 7 位设置为“0”。

4.3.6 端口 7

端口 7 有 12 位输入输出端口，可以按位控制输入输出设置。

端口 7 包括以下复用功能引脚：

表 4-9. 端口 7 复用功能引脚

引脚编号		功能名称	复用功能		备注	框图类型
GC	F1		名称	输入/输出		
100	A3	P70	ANI0	输入	–	A-1
99	B3	P71	ANI1	输入		A-1
98	C3	P72	ANI2	输入		A-1
97	D3	P73	ANI3	输入		A-1
96	A4	P74	ANI4	输入		A-1
95	B4	P77	ANI5	输入		A-1
94	C4	P76	ANI6	输入		A-1
93	D4	P77	ANI7	输入		A-1
92	A5	P78	ANI8	输入		A-1
91	B5	P79	ANI9	输入		A-1
90	C5	P710	ANI10	输入		A-1
89	D5	P711	ANI11	输入		A-1

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

（1）端口 7 寄存器 H，端口 7 寄存器 L（P7H，P7L）

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

	7	6	5	4	3	2	1	0
P7H	0	0	0	0	P711	P710	P79	P78

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

P7n	输出数据控制 (在输出模式下) (n = 0 至 11)
0	输出 0
1	输出 1

注意事项　A/D 转换期间，不要读取或写入 P7H 和 P7L 寄存器（参见 14.6（4）　复用 I/O）。

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

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

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

	7	6	5	4	3	2	1	0
PM7H	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 至 11)
0	输出模式
1	输入模式

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

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

4.3.7 端口 9

端口 9 有 16 位输入输出端口，可以按位控制输入输出设置。

端口 9 包括以下复用功能引脚：

表 4-10.端口 9 复用功能引脚

引脚编号		功能名称	复用功能		备注	框图类型
GC	F1		名称	输入/输出		
43	H8	P90	A0/KR6/TXDA1/SDA02	输入/输出	可以选择作为N沟道漏极开路输出	U-10
44	L9	P91	A1/KR7/RXDA1/SCL02	输入/输出		U-11
45	K9	P92	A2/TIP41/TOP41/TXDA4	输入/输出		U-16
46	J9	P93	A3/TIP40/TOP40 /RXDA4	输入/输出		U-14
47	L10	P94	A4/TIP31/TOP31 /TXDA5	输入/输出		U-16
48	K10	P95	A5/TIP30/TOP30 /RXDA5	输入/输出		U-14
49	K11	P96	A6/TXDC0/TIP21/TOP21	输入/输出		U-16
50	J11	P97	A7/SIB1/RXDC0/TIP20/TOP20	输入/输出		U-14
51	J10	P98	A8/SOB1	输出		G-3
52	H11	P99	A9/$\overline{\text{SCKB1}}$	输入/输出		G-5
53	H10	P910	A10/SIB3	输入/输出		G-2
54	H9	P911	A11/SOB3	输出		G-3
55	G11	P912	A12/$\overline{\text{SCKB3}}$	输入/输出		G-5
56	G10	P913	A13/INTP4	输入/输出		N-2
57	G9	P914	A14/INTP5/TIP51/TOP51	输入/输出		U-15
58	G8	P915	A15/INTP6/TIP50/TOP50	输入/输出		U-15

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

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

（1）端口 9 寄存器（P9）

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

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

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

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

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

（2）端口 9 模式寄存器（PM9）

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

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

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

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

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

（3）端口 9 模式控制寄存器（PMC9）

（1/2）

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

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

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

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

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

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

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

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

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

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

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

备注

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

(2/2)

PMC97	P97引脚操作模式说明
0	I/O 端口
1	A7 输出/SIB1 输入/TIP20 输入/TOP20 输出

PMC96	P96引脚操作模式说明
0	I/O 端口
1	A6 输出/TIP21 输入/TOP21 输出

PMC95	P95引脚操作模式说明
0	I/O 端口
1	A5 输出/TIP30 输入/TOP30 输出

PMC94	P94引脚操作模式说明
0	I/O 端口
1	A4 输出/TIP31 输入/TOP31 输出

PMC93	P93引脚操作模式说明
0	I/O 端口
1	A3 输出/TIP40 输入/TOP40 输出

PMC92	P92引脚操作模式说明
0	I/O 端口
1	A2 输出/TIP41 输入/TOP41 输出

PMC91	P91引脚操作模式说明
0	I/O 端口
1	A1 输出/KR7 输入/RXDA1 输入/SCL02 I/O

PMC90	P90引脚操作模式说明
0	I/O 端口
1	A0 输出/KR6 输入/TXDA1 输出/SDA02 I/O

注意事项 当 A0 至 A15 引脚用作 P90 至 P915 引脚的复用功能时，请确保一次将 PMC9 寄存器的所有 16 位设置为 FFFFH。

（4）端口 9 功能控制寄存器（PFC9）

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

复位后：0000H　R/W　地址：PFC9 FFFFF472H,
PFC9L FFFFF472H, PFC9H FFFFF473H

	15	14	13	12	11	10	9	8
PFC9 (PFC9H)	PFC915	PFC914	PFC913	PFC912	PFC911	PFC910	PFC99	PFC98

	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

备注
1. 关于复用功能规范的详情，参见 **4.3.7（6） 端口 9 复用功能规范。**
2. PFC9 寄存器可以按 16 位宽度进行读写。
   但是当寄存器 PFC9 的高 8 位和低 8 位分别用作 PFC9H 寄存器和 PFC9L 寄存器时，PFC9 可以按字节和按位读出或写入。
3. 若要按字节或按位读取或写入 PFC9 的第 8 位至第 15 位，将这些位作为 PFC9H 寄存器的第 0 位至第 7 位。

（5）端口 9 功能控制扩展寄存器（PFCE9）

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

复位后：0000H　R/W　地址：PFCE9 FFFFF712H,
PFCE9L FFFFF712H, PFCE9H FFFFF713H

	15	14	13	12	11	10	9	8
PFCE9 (PFCE9H)	PFCE915	PFCE914	0	0	0	0	0	0

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

备注
1. 关于复用功能规范的详情，参见 **4.3.7（6） 端口 9 复用功能规范。**
2. PFCE9 寄存器可以按 16 位宽度进行读写。
   但是当寄存器 PFCE9 的高 8 位和低 8 位分别用作 PFCE9H 寄存器和 PFCE9L 寄存器时，PFCE9 可以按字节和按位读出或写入。
3. 若要按字节或按位读取或写入 PFCE9 的第 8 位至第 15 位，将这些位作为 PFCE9H 寄存器的第 0 位至第 7 位。

（6）端口 9 复用功能规范

PFCE915	PFC915	P915 引脚复用功能说明
0	0	A15 输出
0	1	INTP6 输入
1	0	TIP50 输入
1	1	TOP50 输出

PFCE914	PFC914	P914 引脚复用功能说明
0	0	A14 输出
0	1	INTP5 输入
1	0	TIP51 输入
1	1	TOP51 输出

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

PFC912	P912 引脚复用功能说明
0	A12 输出
1	$\overline{\text{SCK}}$B3 I/O

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

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

PFC99	P99 引脚复用功能说明
0	A9 输出
1	$\overline{\text{SCK}}$B1 I/O

PFC98	P98 引脚复用功能说明
0	A8 输出
1	SOB1 输出

PFCE97	PFC97	P97 引脚复用功能说明
0	0	A7 输出
0	1	SIB1 输入/RXDC0 输入[注]
1	0	TIP20 输入
1	1	TOP20 输出

注 SIB1 和 RXDC0 功能不能同时使用。使用 SIB1 引脚时，停止 UARTC0 接收（将 UC0CTL0.UC0RXE 位清除为 0）。而使用 RXDC0 引脚时，停止 CSIB1 接收（将 CB1CTL0.CB1RXE 位清除为 0）。

PFCE96	PFC96	P96 引脚复用功能说明
0	0	A6 输出
0	1	TXDC0 输出
1	0	TIP21 输入
1	1	TOP21 输出

PFCE95	PFC95	P95 引脚复用功能说明
0	0	A5 输出
0	1	TIP30 输入
1	0	TOP30 输出
1	1	RXDA5 输入

PFCE94	PFC94	P94 引脚复用功能说明
0	0	A4 输出
0	1	TIP31 输入
1	0	TOP31 输出
1	1	TXDA5 输出

PFCE93	PFC93	P93 引脚复用功能说明
0	0	A3 输出
0	1	TIP40 输入
1	0	TOP40 输出
1	1	RXDA4 输入

PFCE92	PFC92	P92 引脚复用功能说明
0	0	A2 输出
0	1	TIP41 输入
1	0	TOP41 输出
1	1	TXDA4 输出

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

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

注 RXDA1 和 KR7 功能不能同时使用。当该引脚用作 RXDA1 时，不要使用 KR7 功能。而该引脚用作 KR7 时，不要使用 RXDA1 功能（建议将 PFC91 位设置为 1，并且将 PFCE91 位清除为 0）。

（7）端口 9 功能寄存器（PF9）

复位后: 0000H R/W 地址: PF3 FFFFFC72H,
PF9L FFFFFC72H, PF9H FFFFFC73H

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

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

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

注意事项 **若输出引脚 P90 至 P96 被上拉至 EV_{DD} 或更高电压时，确保将 PF9n 位设置为 1。**
上拉输出引脚 P97 至 P915 至 EV_{DD} 相同电势，即使它们用作 N 沟道漏极开路输出引脚。

备注 **1.** PF9 寄存器可以按 16 位宽度进行读写。
但是当寄存器 PF9 的高 8 位和低 8 位分别用作 PF9H 寄存器和 PF9L 寄存器时，PF9 可以按字节和按位读出或写入。

2. 若要按字节或按位读取或写入 PF9 的第 8 位至第 15 位，将这些位作为 PF9H 寄存器的第 0 位至第 7 位。

4.3.8 端口 CM

端口 CM 有 4 位输入输出端口，可以按位控制输入输出设置。

端口 CM 包括以下复用功能引脚：

表 4-11. 端口 CM 复用功能引脚

引脚编号		功能名称	复用功能		备注	框图类型
GC	F1		名称	输入/输出		
61	F11	PCM0	$\overline{\text{WAIT}}$	输入	–	D-1
62	F10	PCM1	CLKOUT	输出		D-2
63	E10	PCM2	$\overline{\text{HLDAK}}$	输出		D-2
64	E9	PCM3	$\overline{\text{HLDRQ}}$	输入		D-1

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

（1）端口 CM 寄存器（PCM）

复位后：00H (输出锁存)　R/W　地址：FFFFF00CH

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

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

（2）端口 CM 寄存器（PMCM）

复位后：FFH　R/W　地址：FFFFF02CH

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

PMCMn	I/O 模式控制 (n = 0 至 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	PCM3 引脚操作模式说明
0	I/O 端口
1	$\overline{\text{HLDRQ}}$ 输入

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

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

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

4.3.9 端口CT

端口 CT 有 4 位输入输出端口，可以按位控制输入输出设置。

端口 CT 包括以下复用功能引脚：

表 4-12.端口 CT 复用功能引脚

引脚编号		功能名称	复用功能		备注	框图类型
GC	F1		名称	输入/输出		
65	E8	PCT0	$\overline{WR0}$	输出	–	D-2
66	D10	PCT1	$\overline{WR1}$	输出		D-2
67	D9	PCT4	$\overline{RD}$	输出		D-2
68	D8	PCT6	ASTB	输出		D-2

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

（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 (输出模式下) (n = 0, 1, 4, 6)
0	输出 0
1	输出 1

（2）端口 CT 模式寄存器（PMCT）

复位后：FFH　R/W　地址：FFFFF02AH

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

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

（3）端口 CT 模式控制寄存器（PMCCT）

复位后: 00H　R/W　地址: FFFFF04AH

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

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

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

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

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

4.3.10 端口 DH

端口 DH 有 5 位输入输出端口，可以按位控制输入输出设置。

端口 DH 包括以下复用功能引脚：

表 4-13.端口 DH 复用功能引脚

引脚编号		功能名称	复用功能		备注	框图类型
GC	F1		名称	输入/输出		
87	C6	PDH0	A16	输出	–	D-2
88	D6	PDH1	A17	输出		D-2
59	F9	PDH2	A18	输出		D-2
60	F8	PDH3	A19	输出		D-2
6	F4	PDH4	A20	输出		D-2

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

（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	0	PDH4	PDH3	PDH2	PDH1	PDH0

PDHn	输出数据控制 (输出模式下) (n = 0 至 4)
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	1	PMDH4	PMDH3	PMDH2	PMDH1	PMDH0

PMDHn	I/O 模式控制 (n = 0 至 4)
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

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

PMCDHn	PDHn 引脚操作模式说明(n = 0 至 4)
0	I/O 端口
1	Am 输出 (地址总线输出) (m = 16 至 20)

注意事项　逐位指定端口或者复用功能模式时，请确保地址总线输出功能正常。

4.3.11 端口 DL

端口 DL 有 16 位输入输出端口，可以按位控制输入输出设置。

端口 DL 包括以下复用功能引脚：

表 4-14.端口 DL 复用功能引脚

引脚编号		功能名称	复用功能		备注	框图类型
GC	F1		名称	输入/输出		
71	C11	PDL0	AD0	输入/输出	–	D-3
72	C10	PDL1	AD1	输入/输出		D-3
73	C9	PDL2	AD2	输入/输出		D-3
74	B11	PDL3	AD3	输入/输出		D-3
75	B10	PDL4	AD4	输入/输出		D-3
76	A10	PDL5	AD5/FLMD1[注]	输入/输出		D-3
77	A9	PDL6	AD6	输入/输出		D-3
78	B9	PDL7	AD7	输入/输出		D-3
79	A8	PDL8	AD8	输入/输出		D-3
80	B8	PDL9	AD9	输入/输出		D-3
81	C8	PDL10	AD10	输入/输出		D-3
82	A7	PDL11	AD11	输入/输出		D-3
83	B7	PDL12	AD12	输入/输出		D-3
84	C7	PDL13	AD13	输入/输出		D-3
85	D7	PDL14	AD14	输入/输出		D-3
86	B6	PDL15	AD15	输入/输出		D-3

注 该引脚设置为 flash 存储器编程模式后，不需要通过端口控制寄存器进行操作。详情参见**第三十一章 FLASH 存储器**。

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

（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 (输出模式下) (n = 0 至 15)
0	输出 0
1	输出 1

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

（2）端口 DL 模式寄存器（PMDL）

复位后: FFFFH　R/W　地址:　PMDL FFFFF024H,
PMDLL FFFFF024H, PMDLH FFFFF025H

	15	14	13	12	11	10	9	8
PMDL (PMDLH)	PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8

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

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

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

（3）端口 DL 模式控制寄存器（PMCDL）

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

	15	14	13	12	11	10	9	8
PMCDL (PMCDLH)	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8

	7	6	5	4	3	2	1	0
(PMCDLL)	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0

PMCDLn	PDLn 引脚操作模式说明 (n = 0 至 15)
0	I/O 端口
1	ADn I/O (地址/数据总线 I/O)

注意事项　逐位指定端口或 ADn I/O 模式时，请根据已使用的外部存储器来指定。

备注

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

4.4　框图

图 4-3.　A-1 型的功能框图

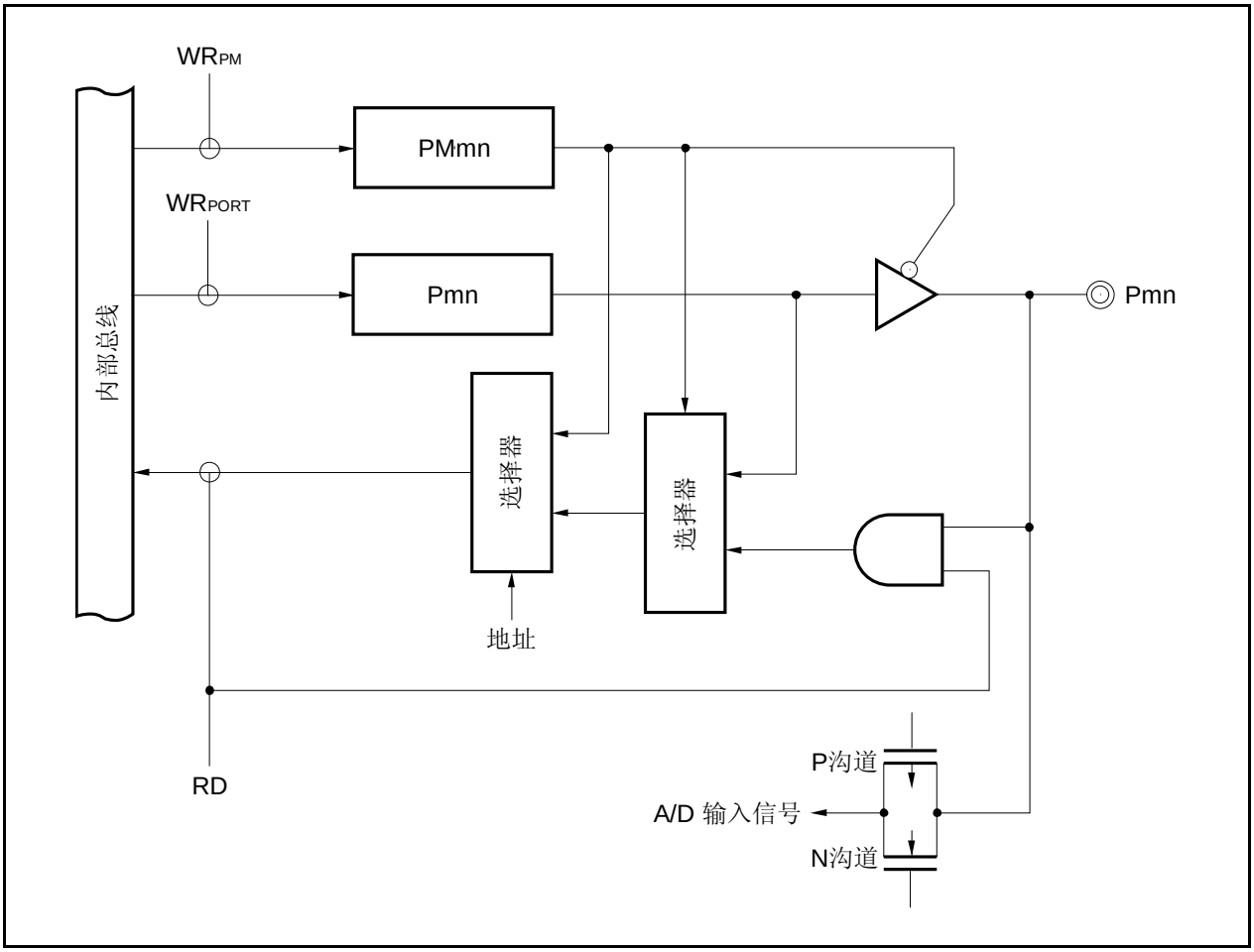


图 4-4.　A-2 型的功能框图

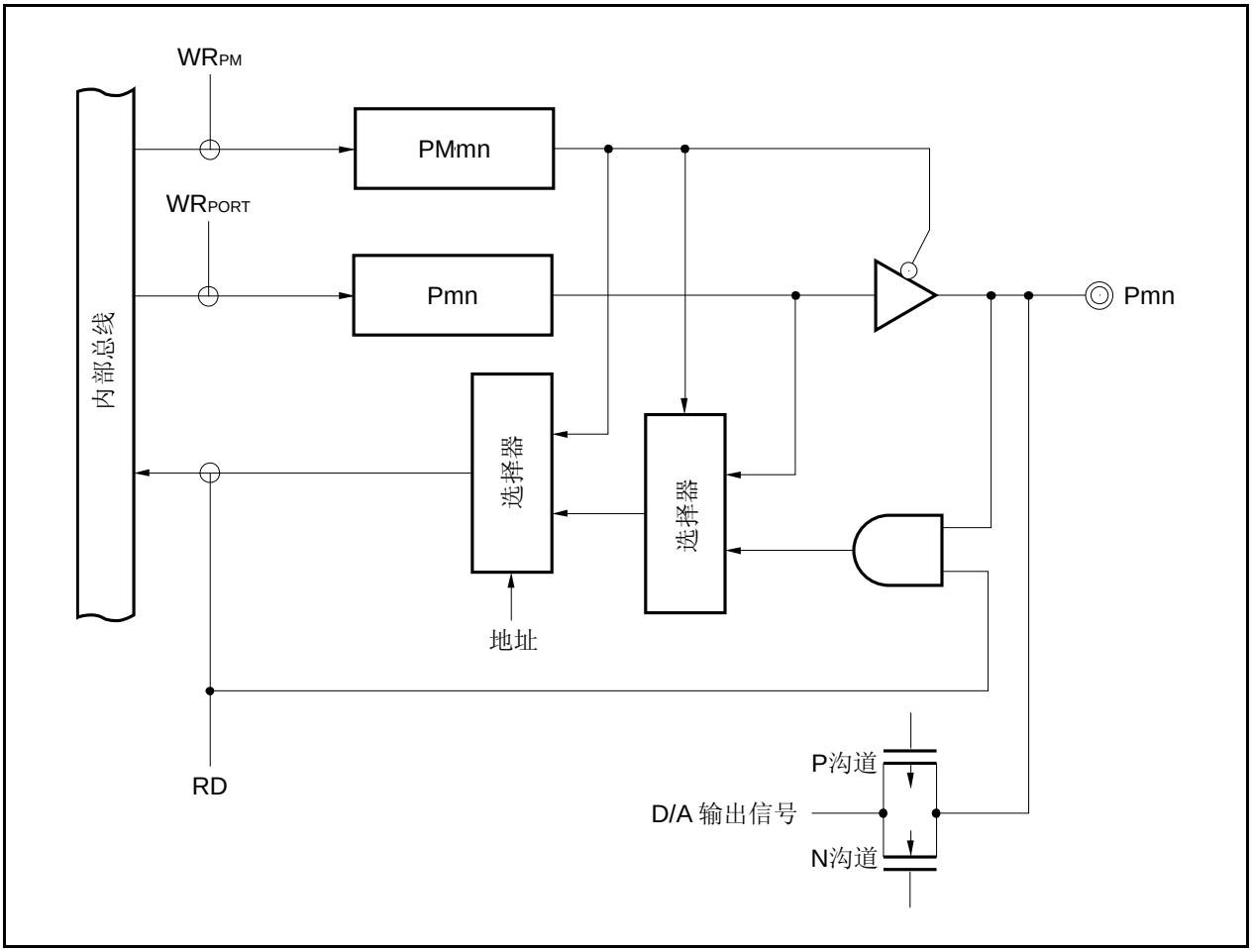


图 4-5.　C-1 型的功能框图

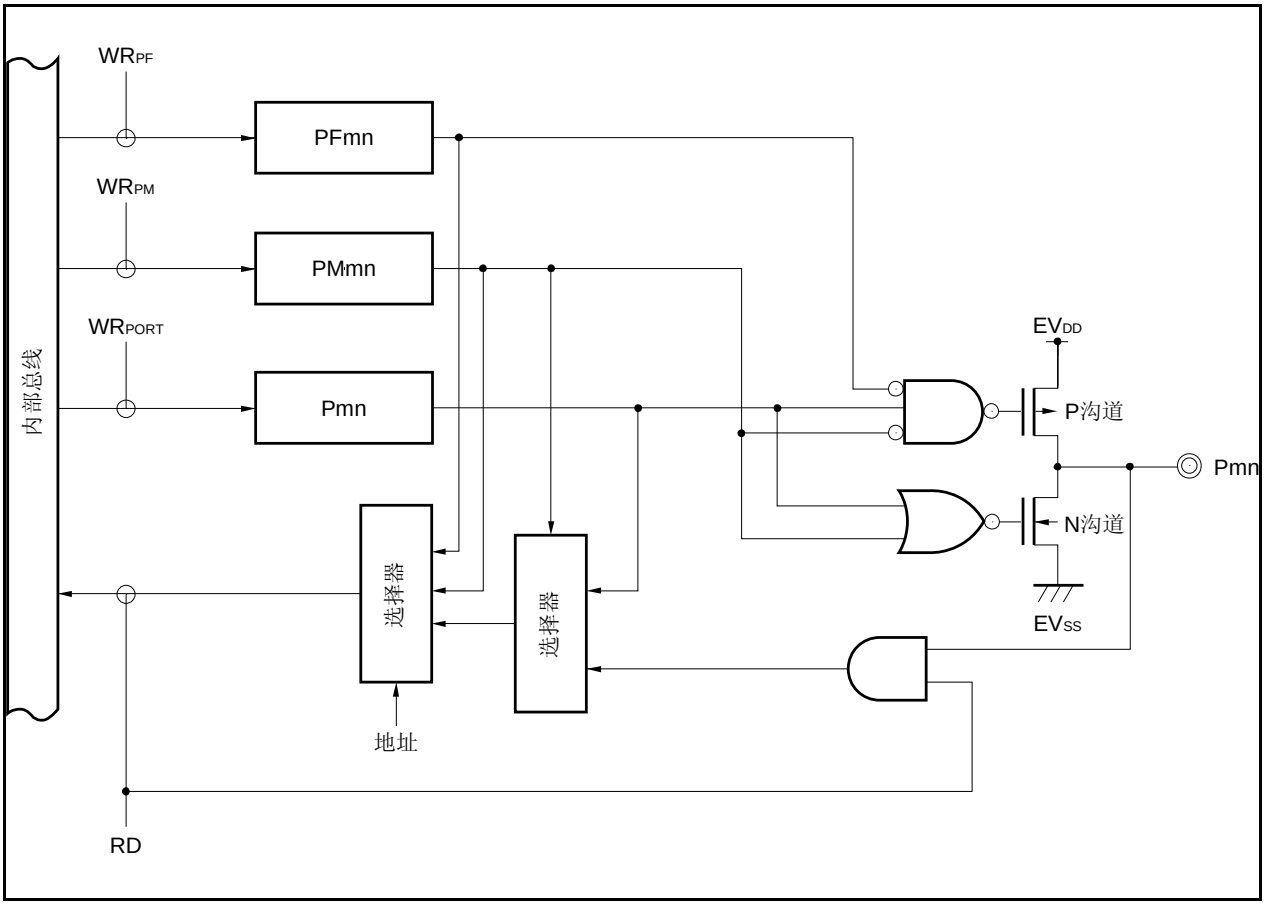


图 4-6.　D-1 型的功能框图

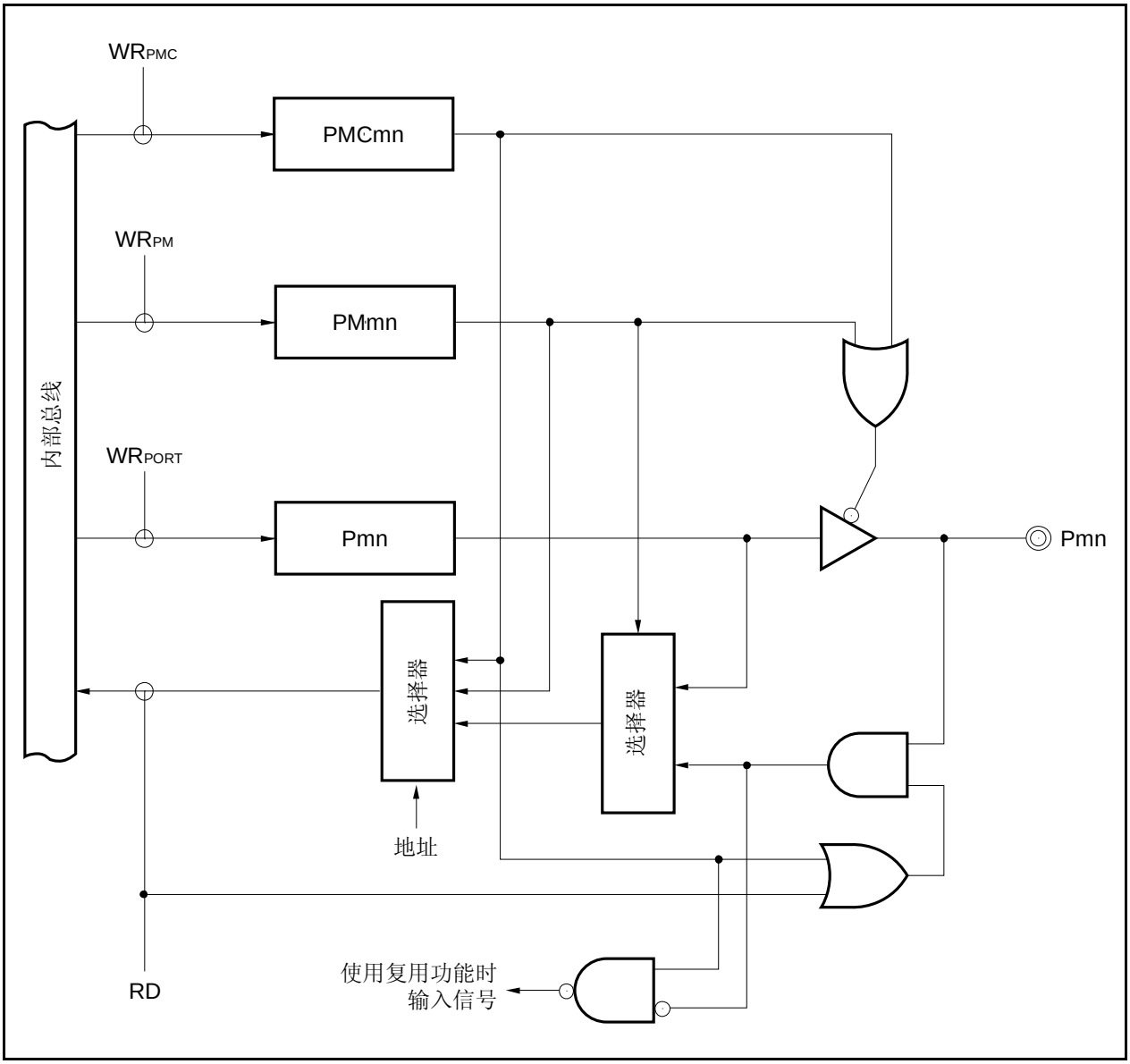


图 4-7.　D-2 型的功能框图

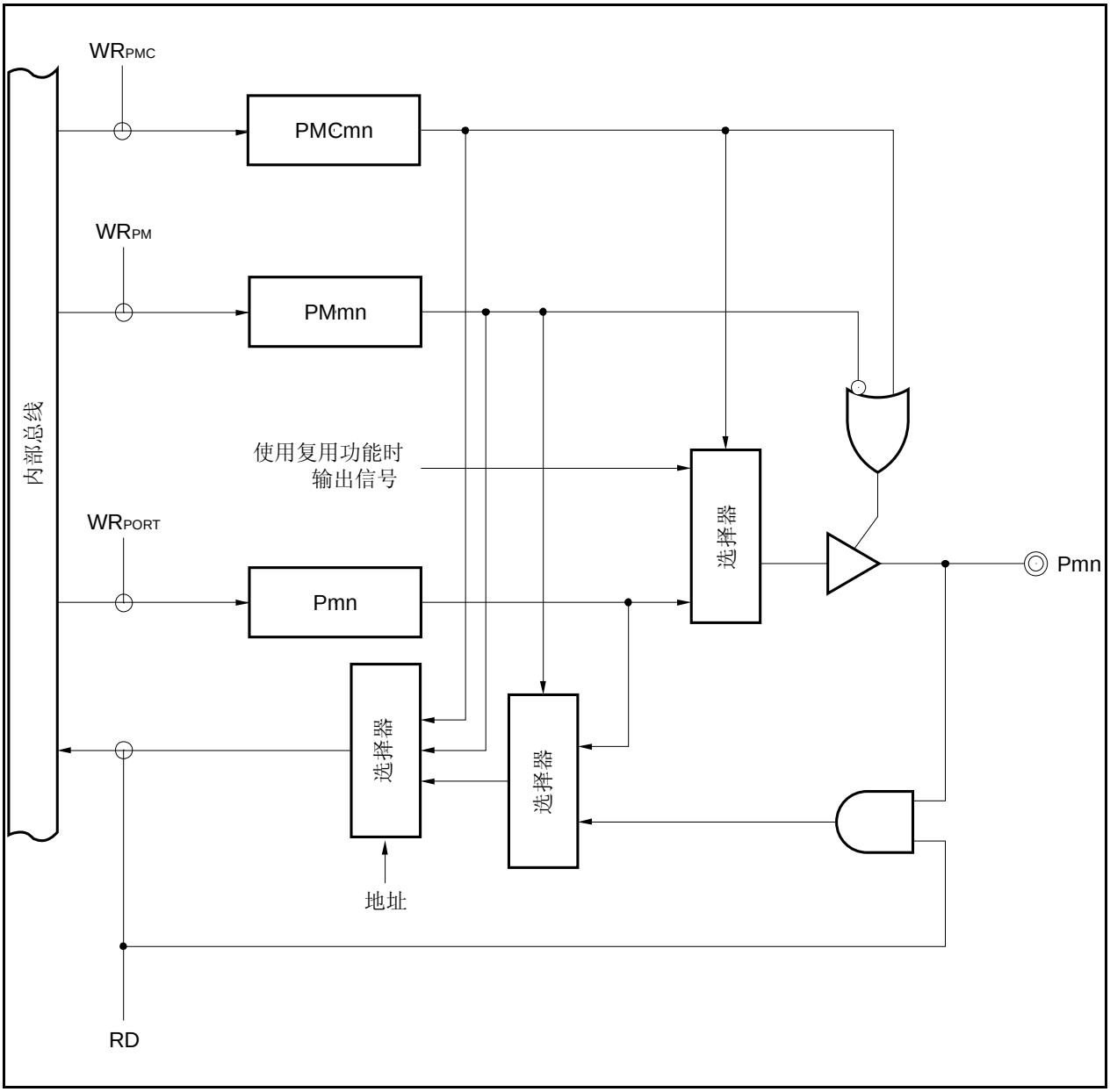


图 4-8.　D-3 型的功能框图

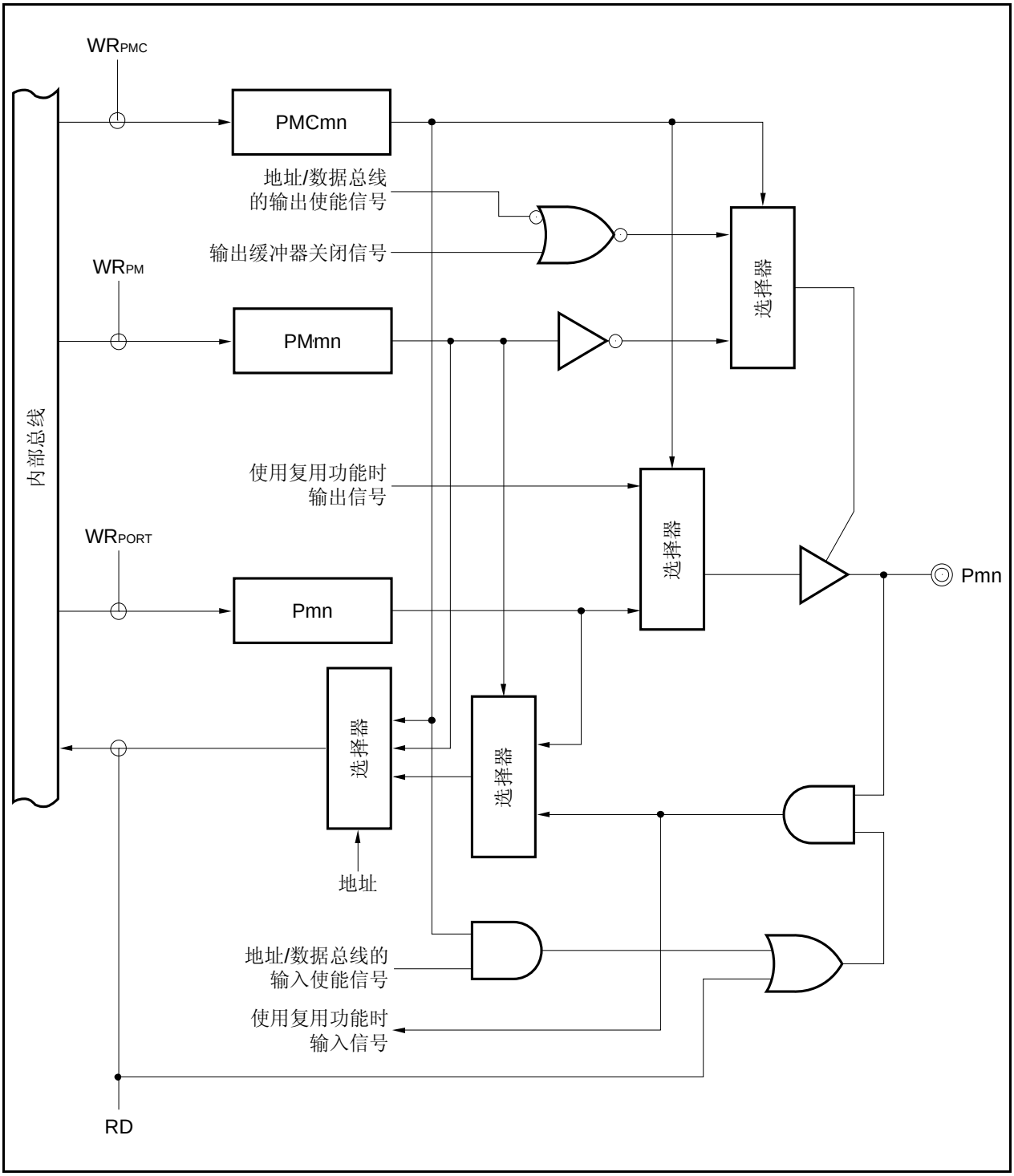
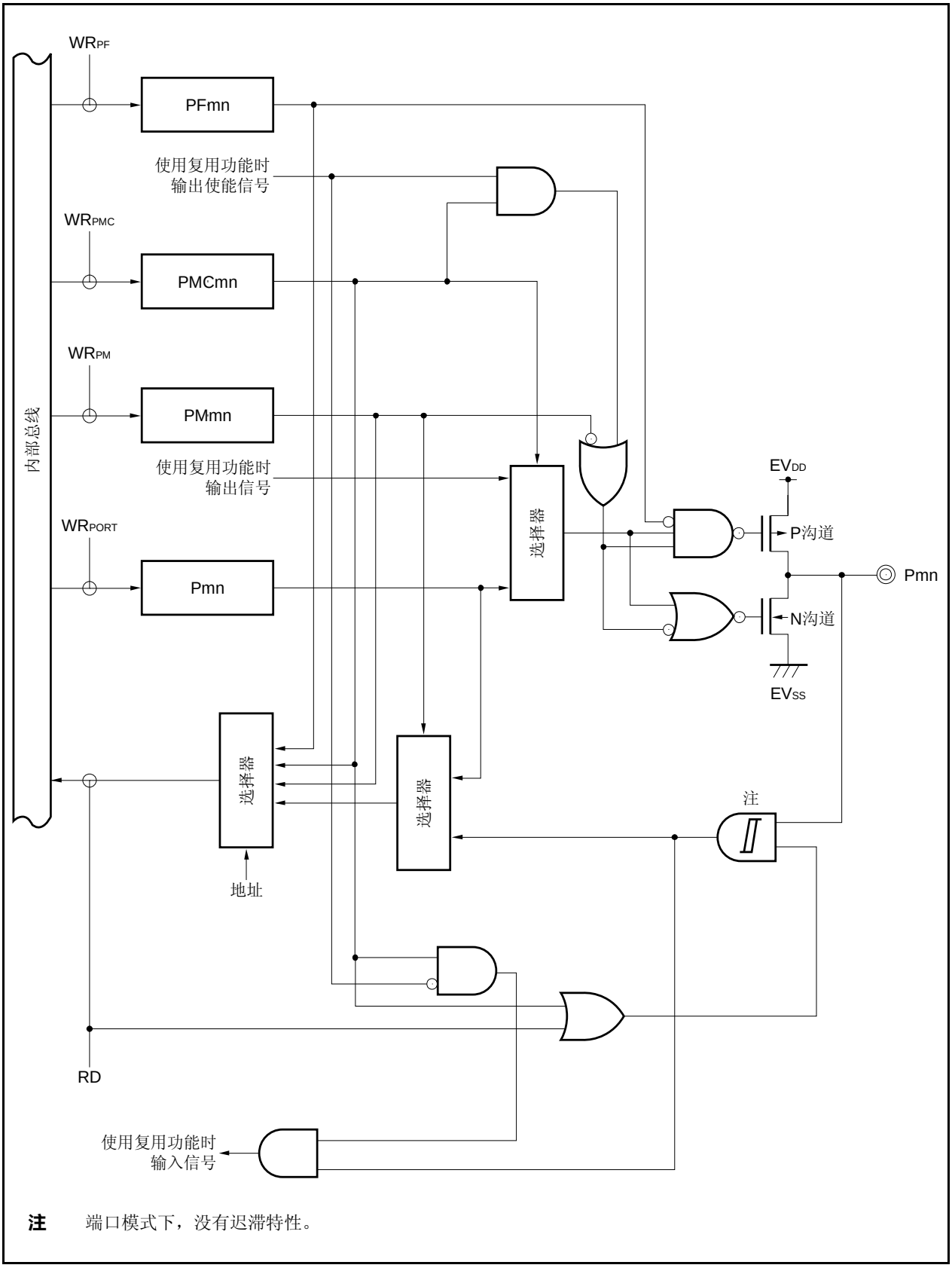
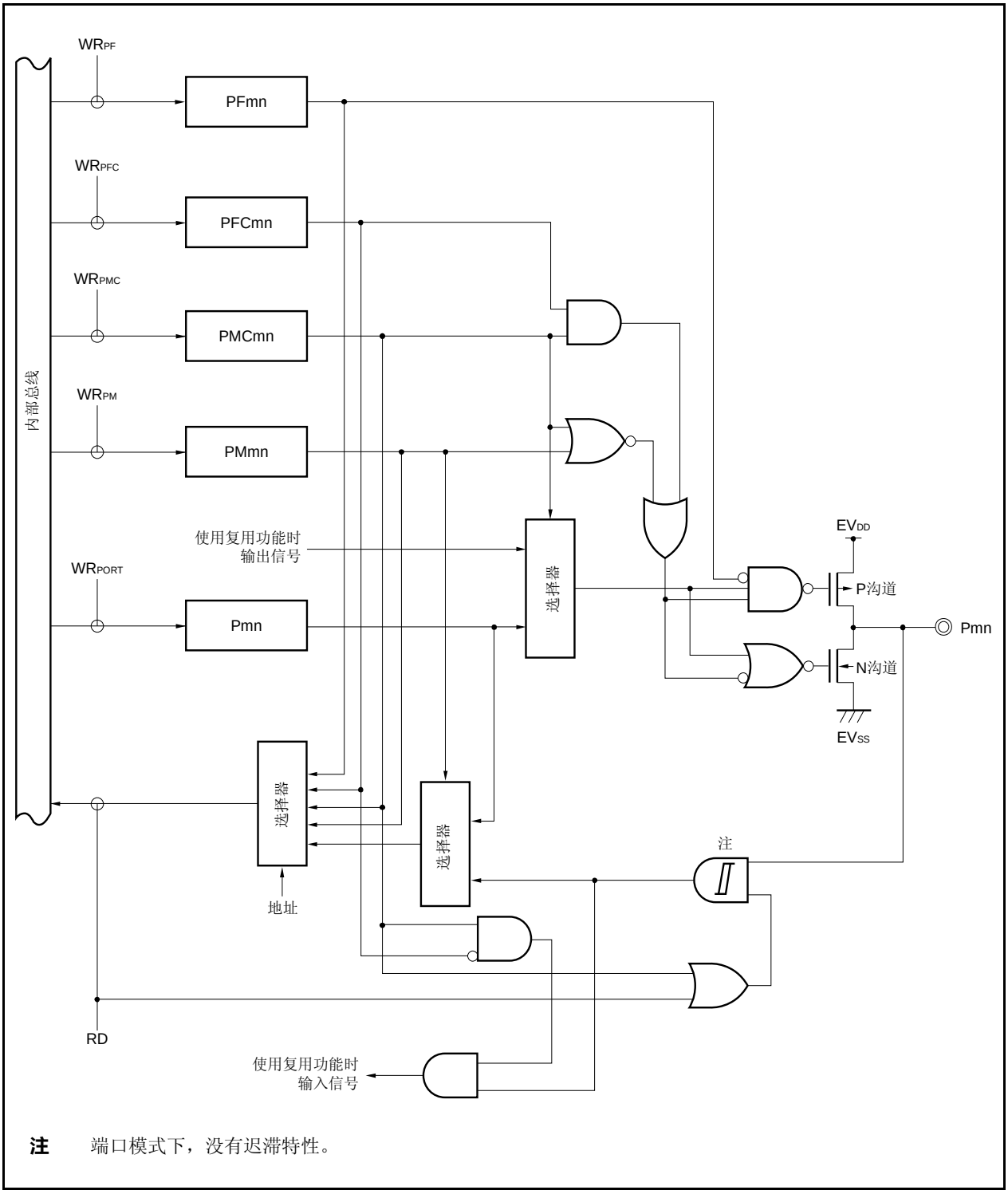


图 4-9.　E-3 型的功能框图



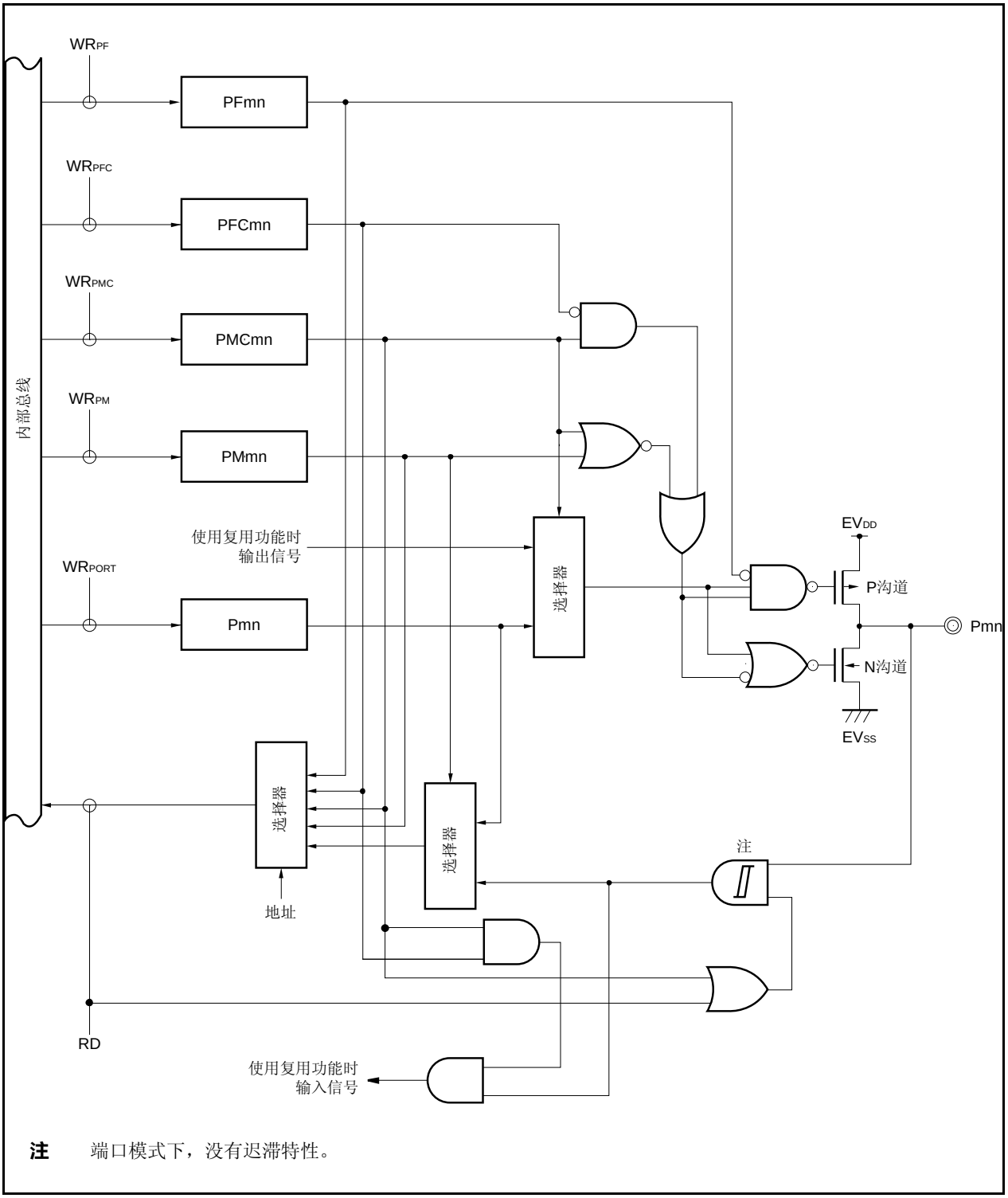
注　端口模式下，没有迟滞特性。

图 4-10.　G-1 型的功能框图



注　端口模式下，没有迟滞特性。

图 4-11.　G-2 型的功能框图



注　端口模式下，没有迟滞特性。

图 4-12.　G-3 型的功能框图

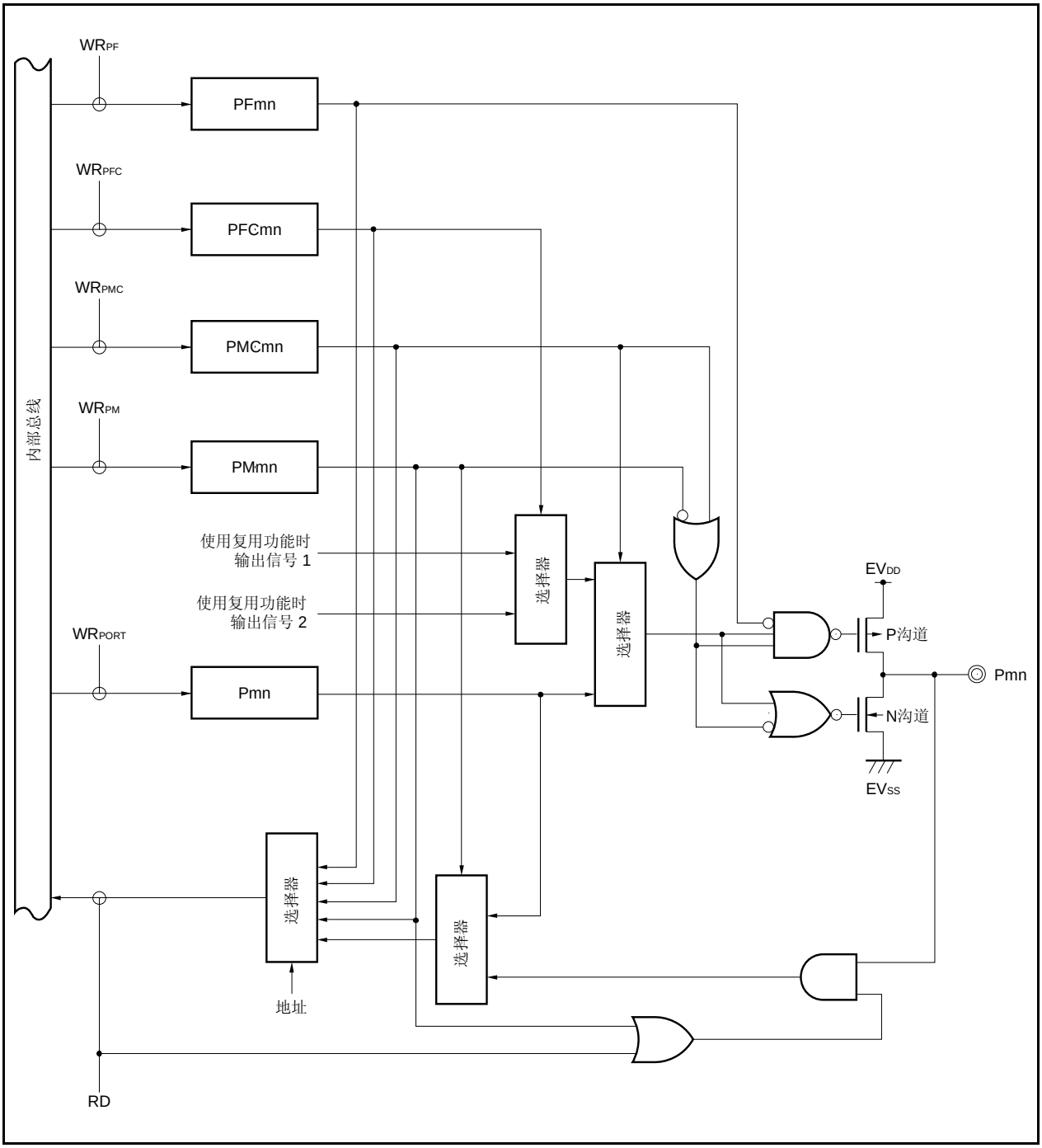
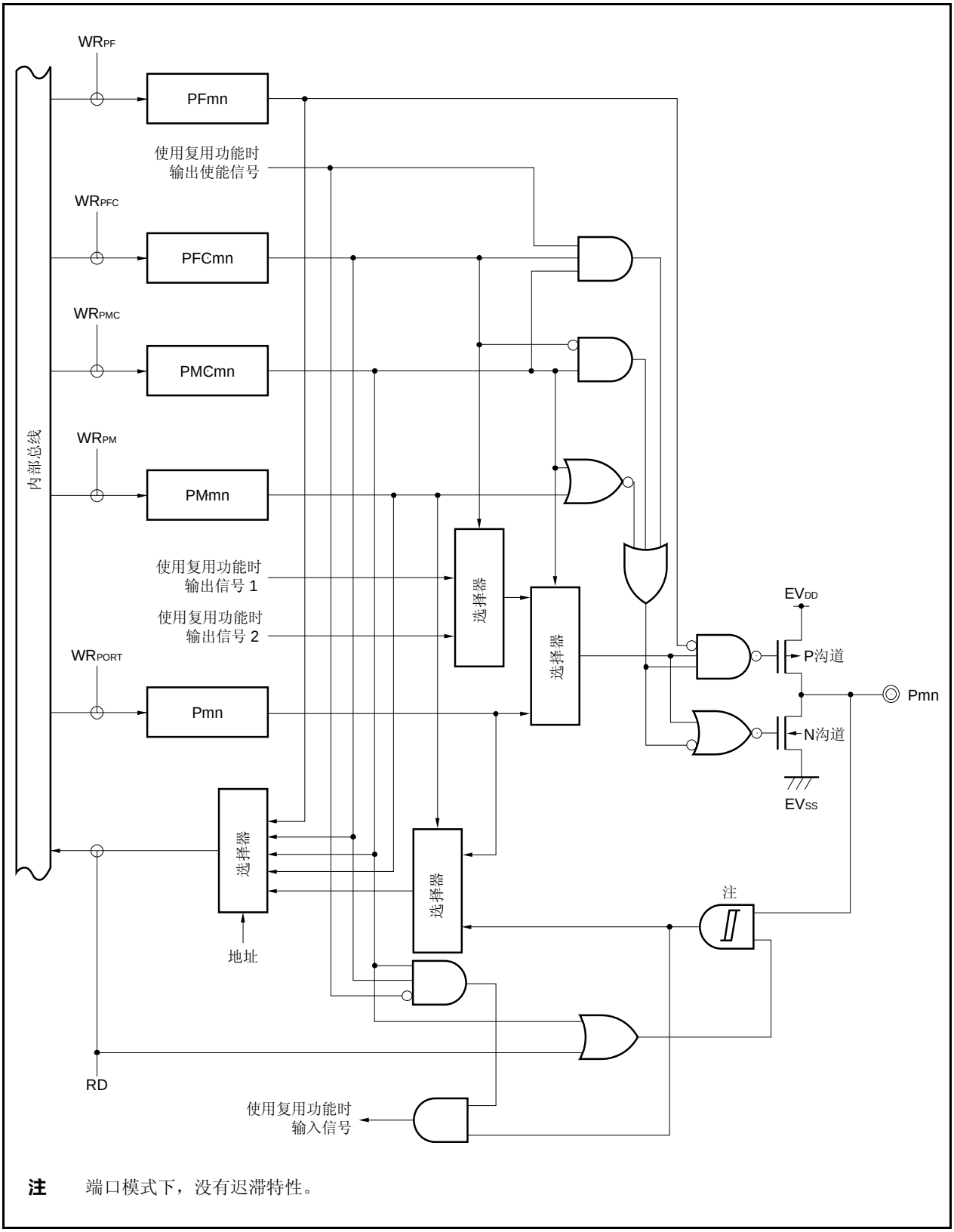
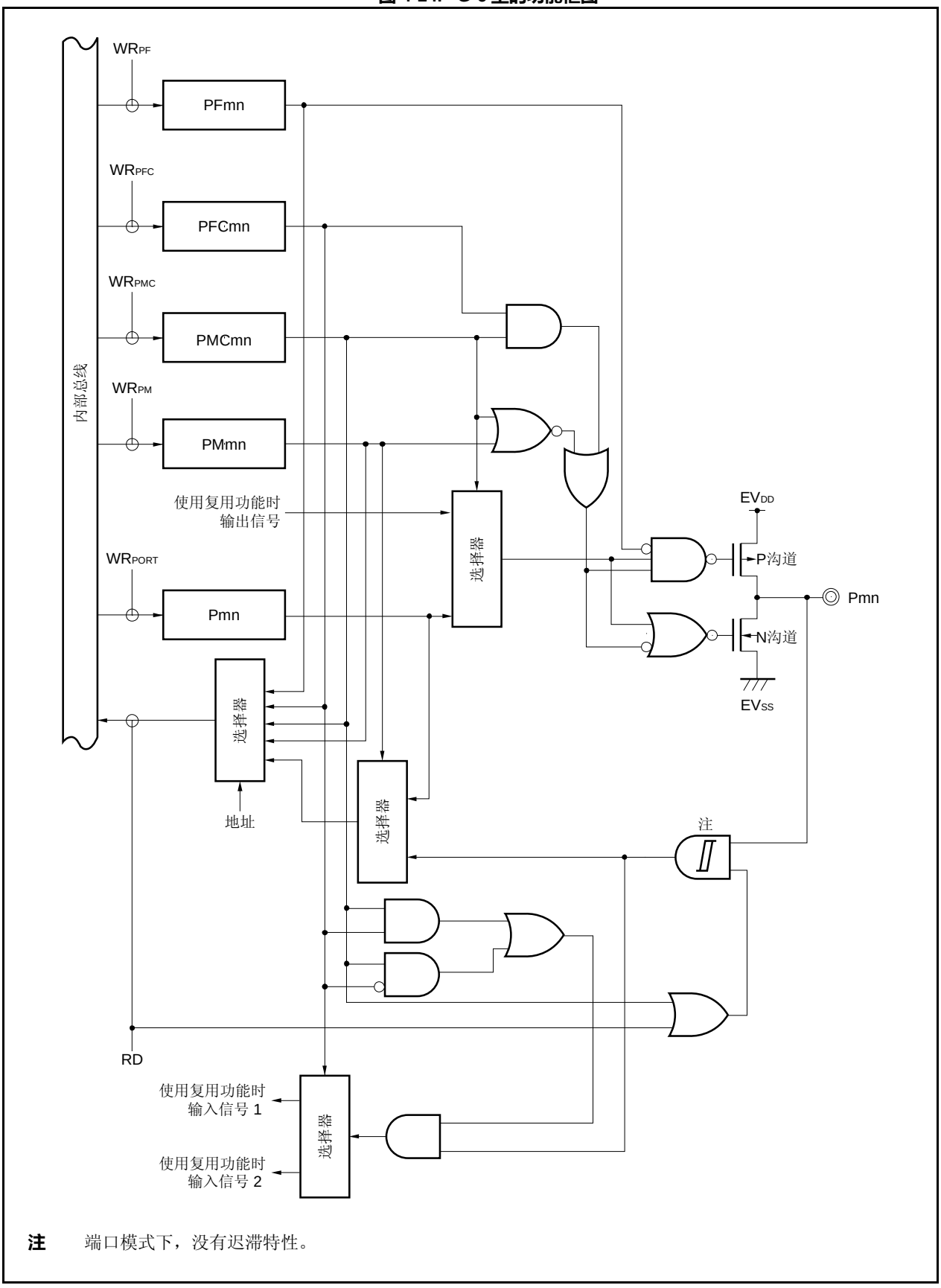


图 4-13.　G-5 型的功能框图



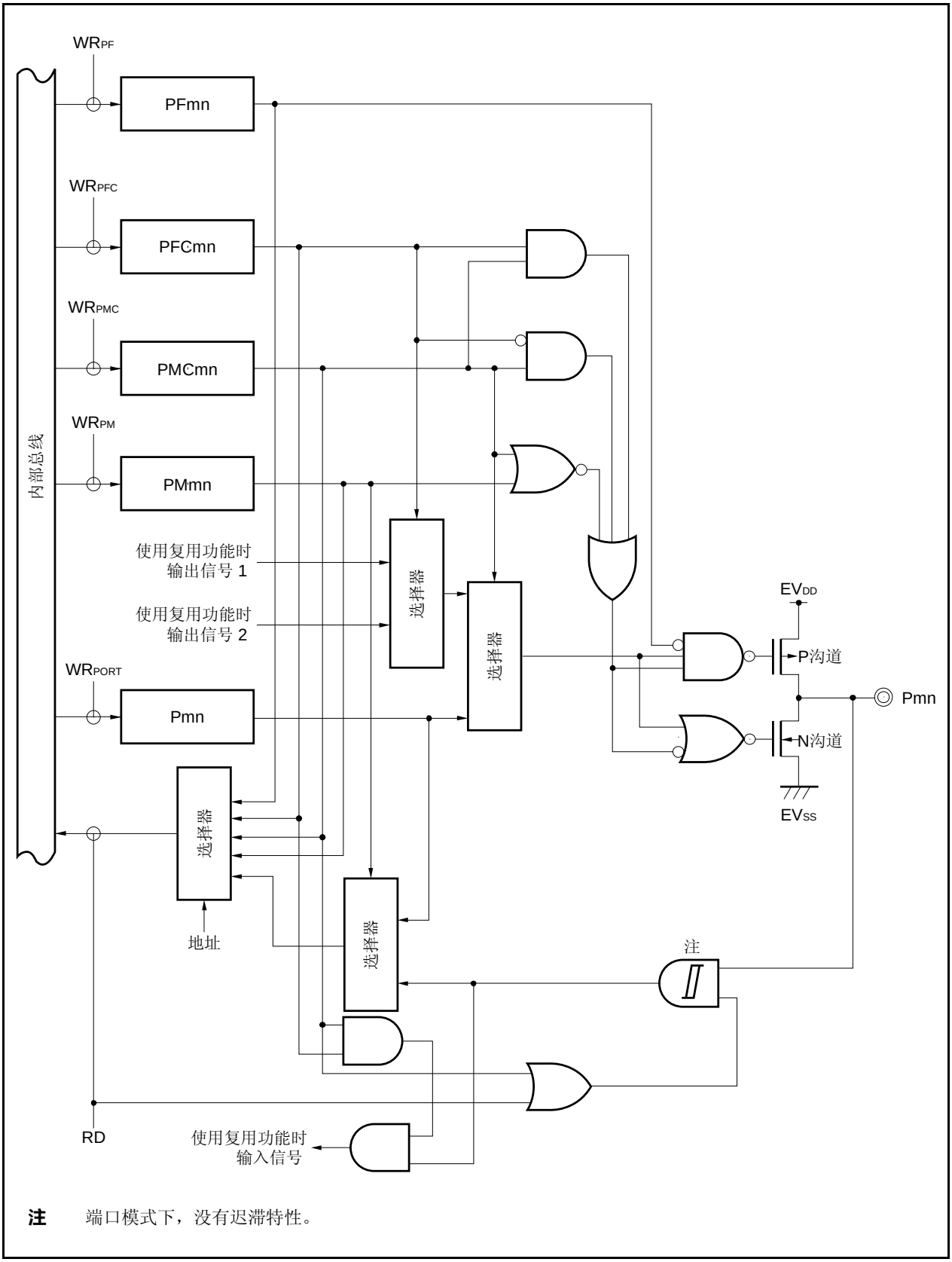
注　端口模式下，没有迟滞特性。

图 4-14.　G-6 型的功能框图



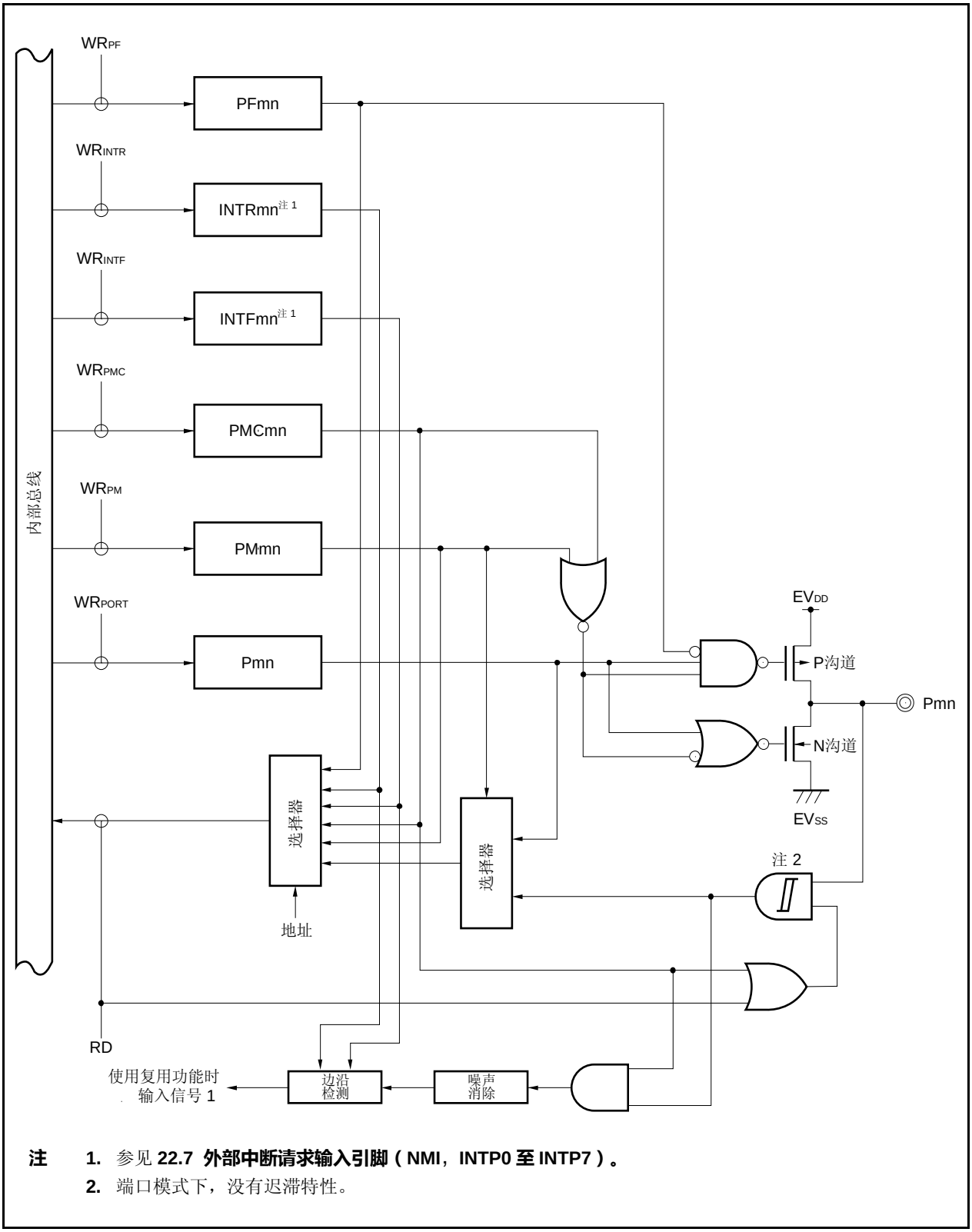
注　端口模式下，没有迟滞特性。

图 4-15.　G-12 型的功能框图



注　端口模式下，没有迟滞特性。

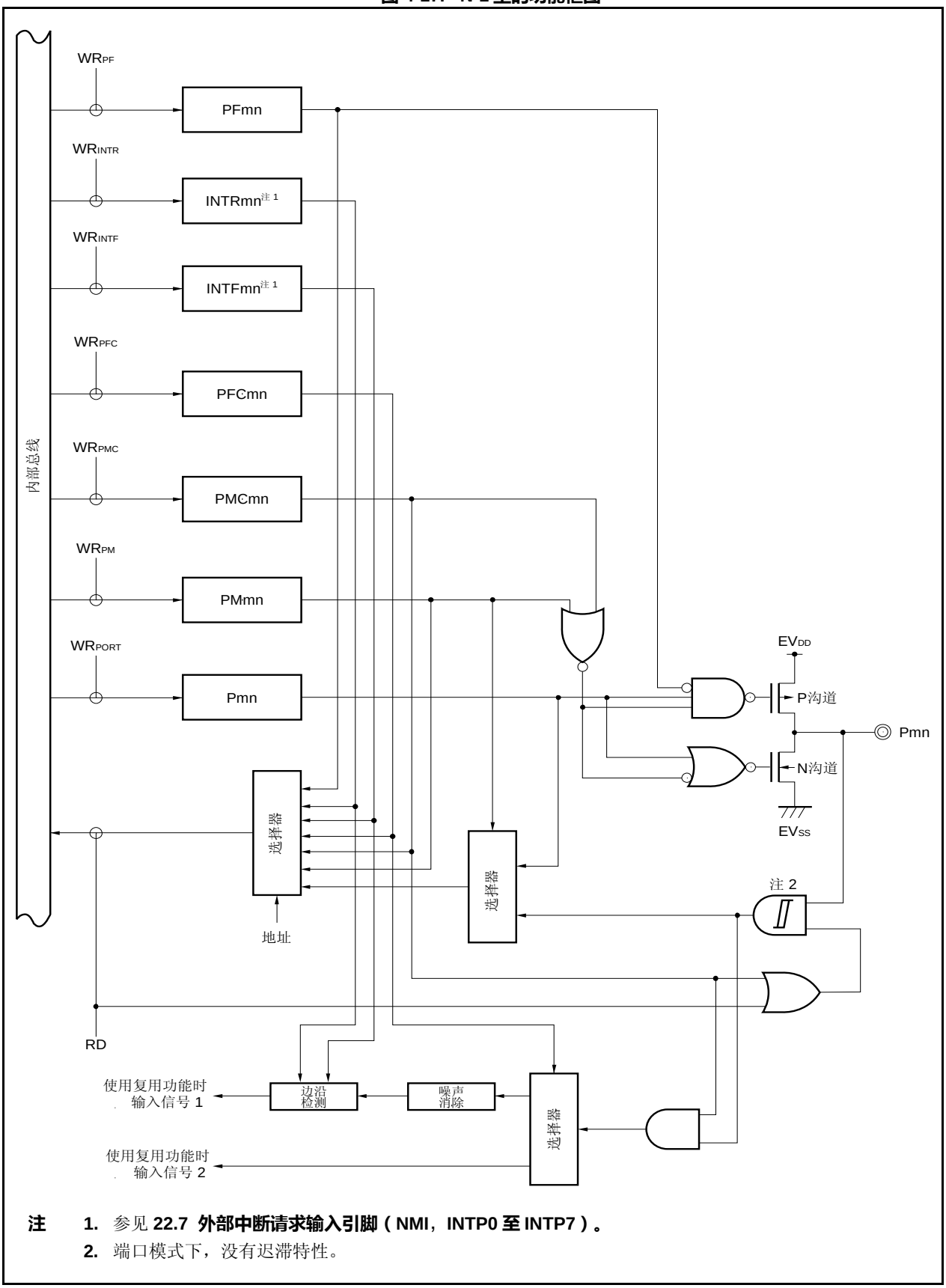
图 4-16.　L-1 型的功能框图



注　**1.**　参见 **22.7　外部中断请求输入引脚（NMI，INTP0 至 INTP7）。**

2.　端口模式下，没有迟滞特性。

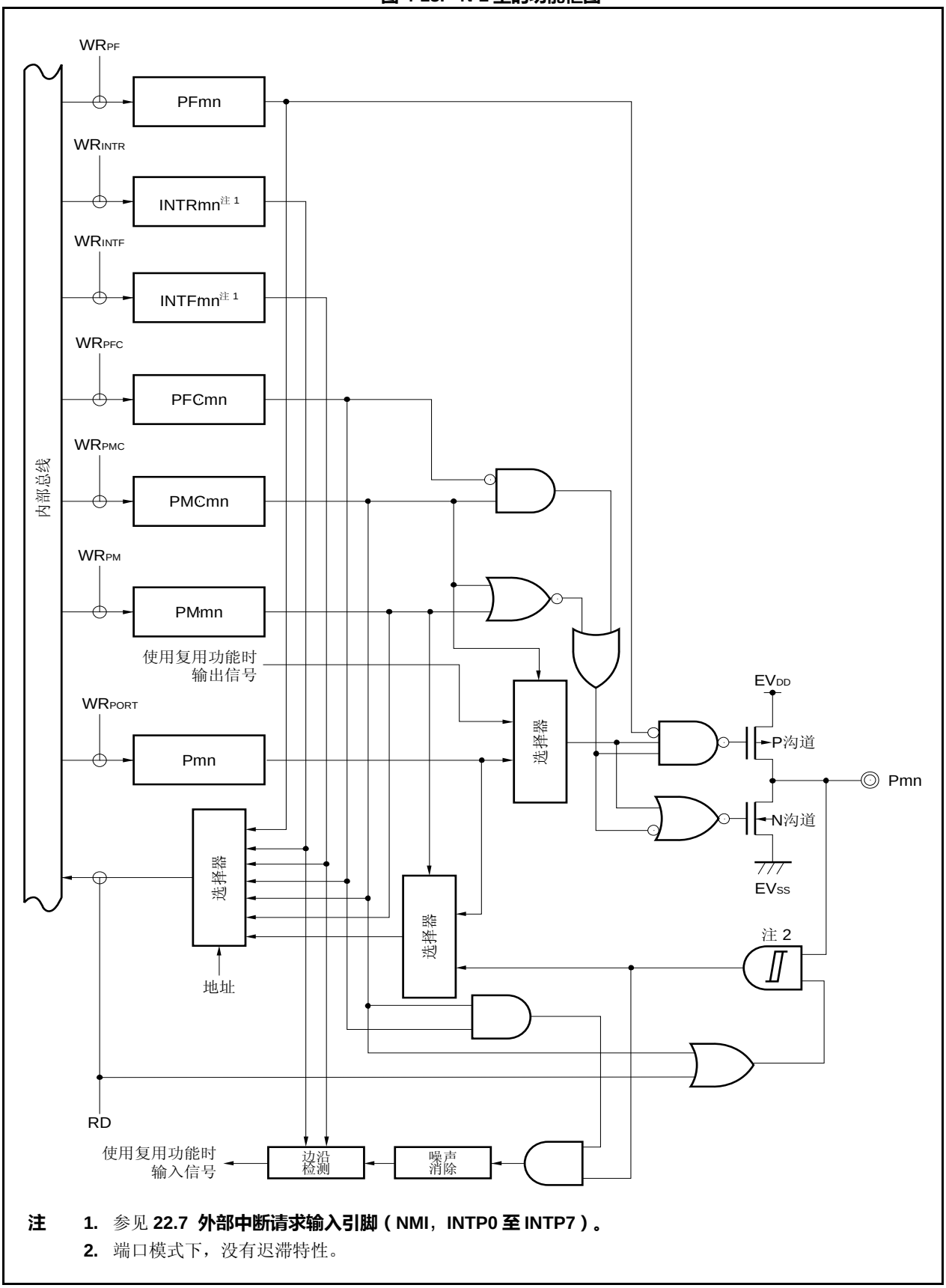
图 4-17.　N-1 型的功能框图



注　**1.**　参见 **22.7 外部中断请求输入引脚（NMI，INTP0 至 INTP7）。**

2.　端口模式下，没有迟滞特性。

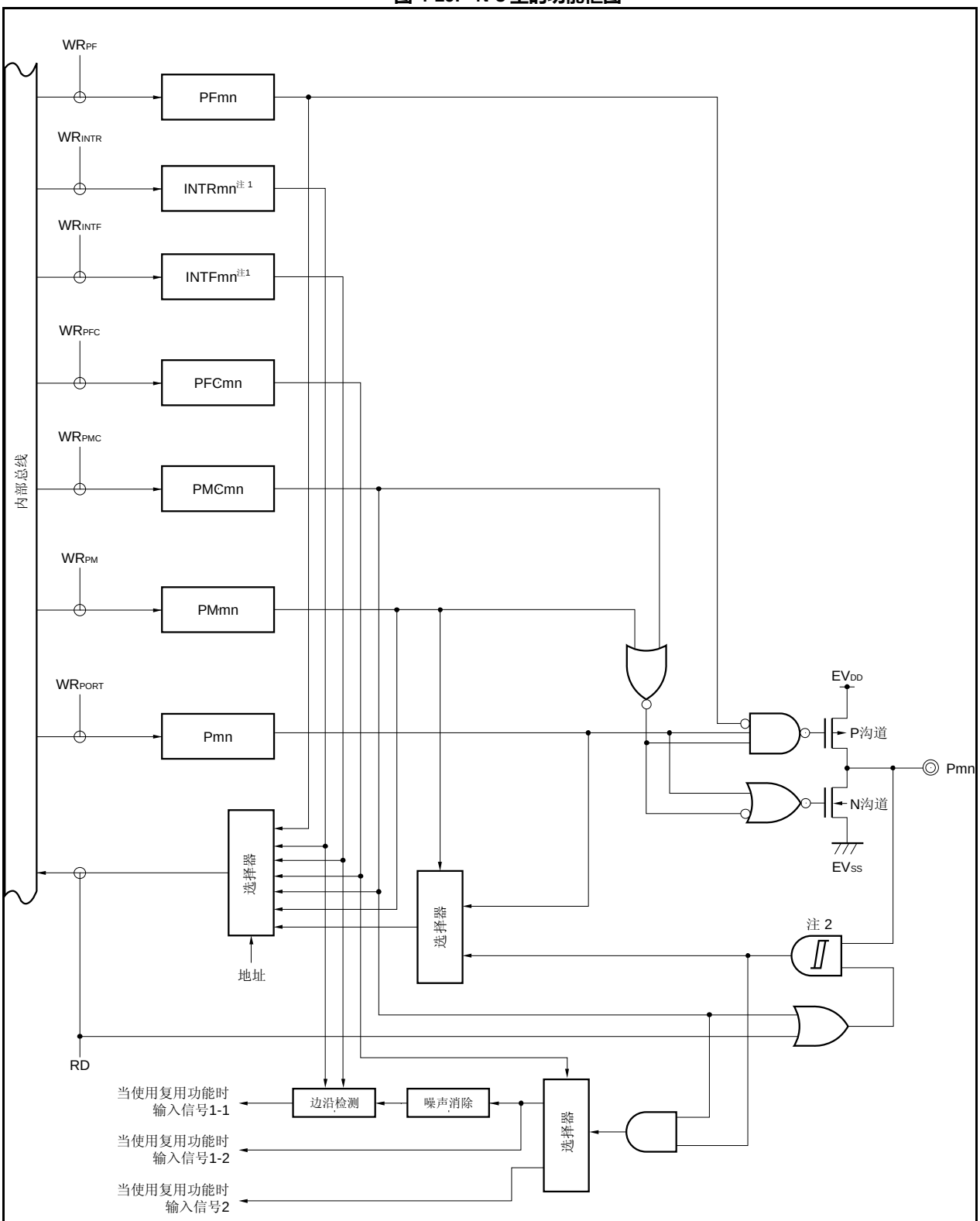
图 4-18.　N-2 型的功能框图



注　**1.**　参见 **22.7　外部中断请求输入引脚（NMI，INTP0 至 INTP7）。**

2.　端口模式下，没有迟滞特性。

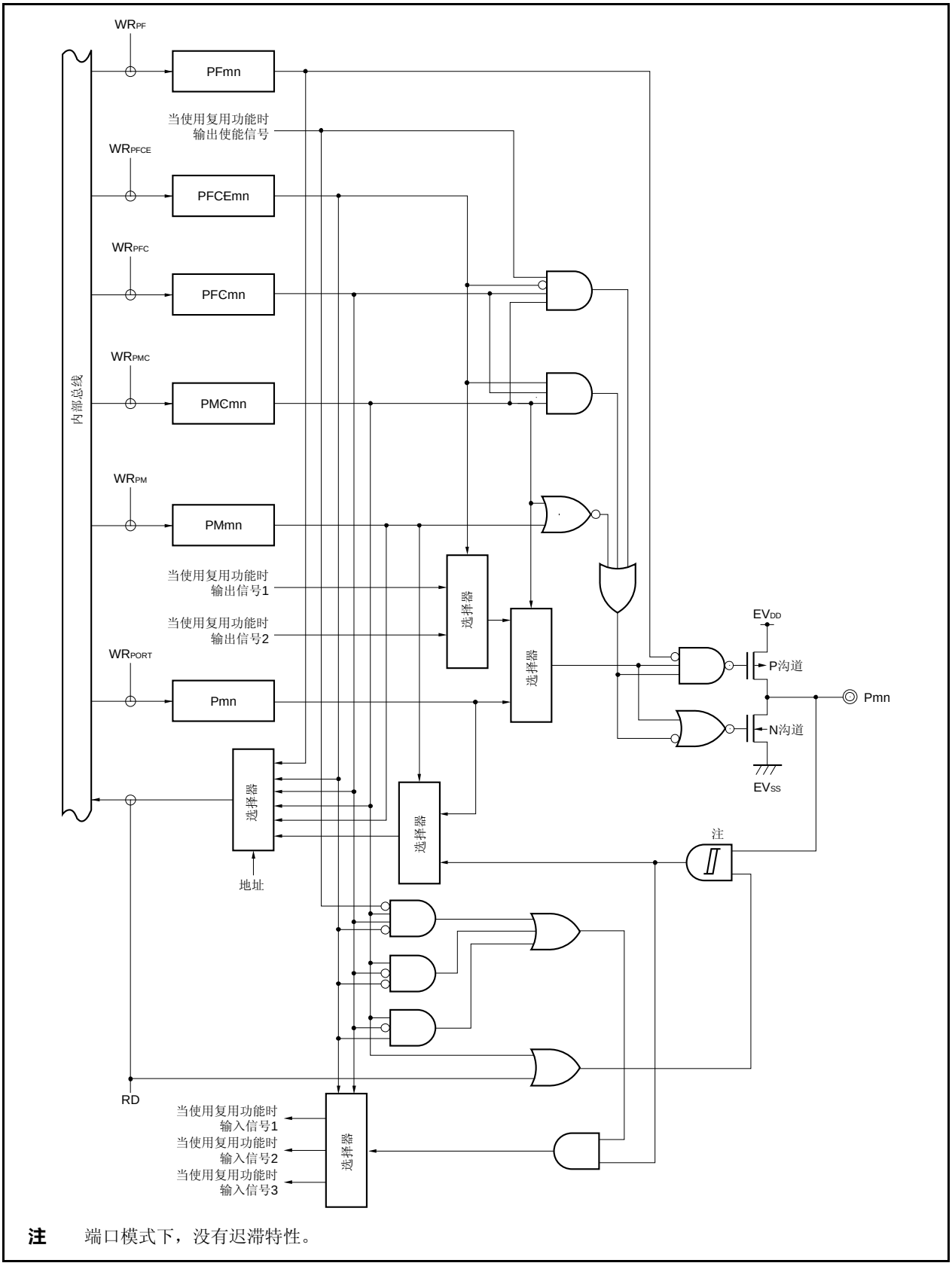
图 4-19.　N-3 型的功能框图



注　**1.**　参见 **22.7　外部中断请求输入引脚（NMI，INTP0 至 INTP7）。**

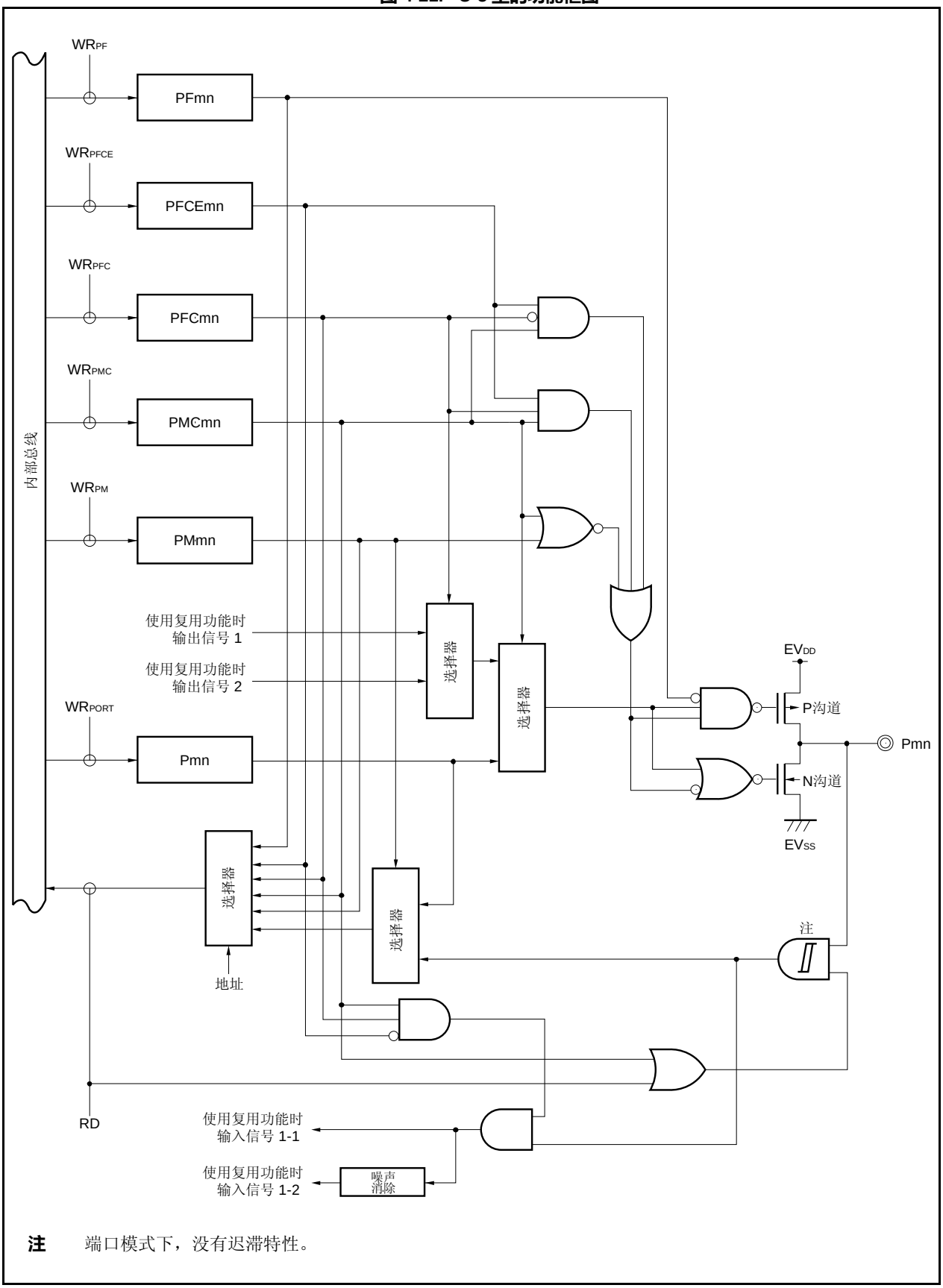
2.　端口模式下，没有迟滞特性。

图 4-20.　U-1 型的功能框图



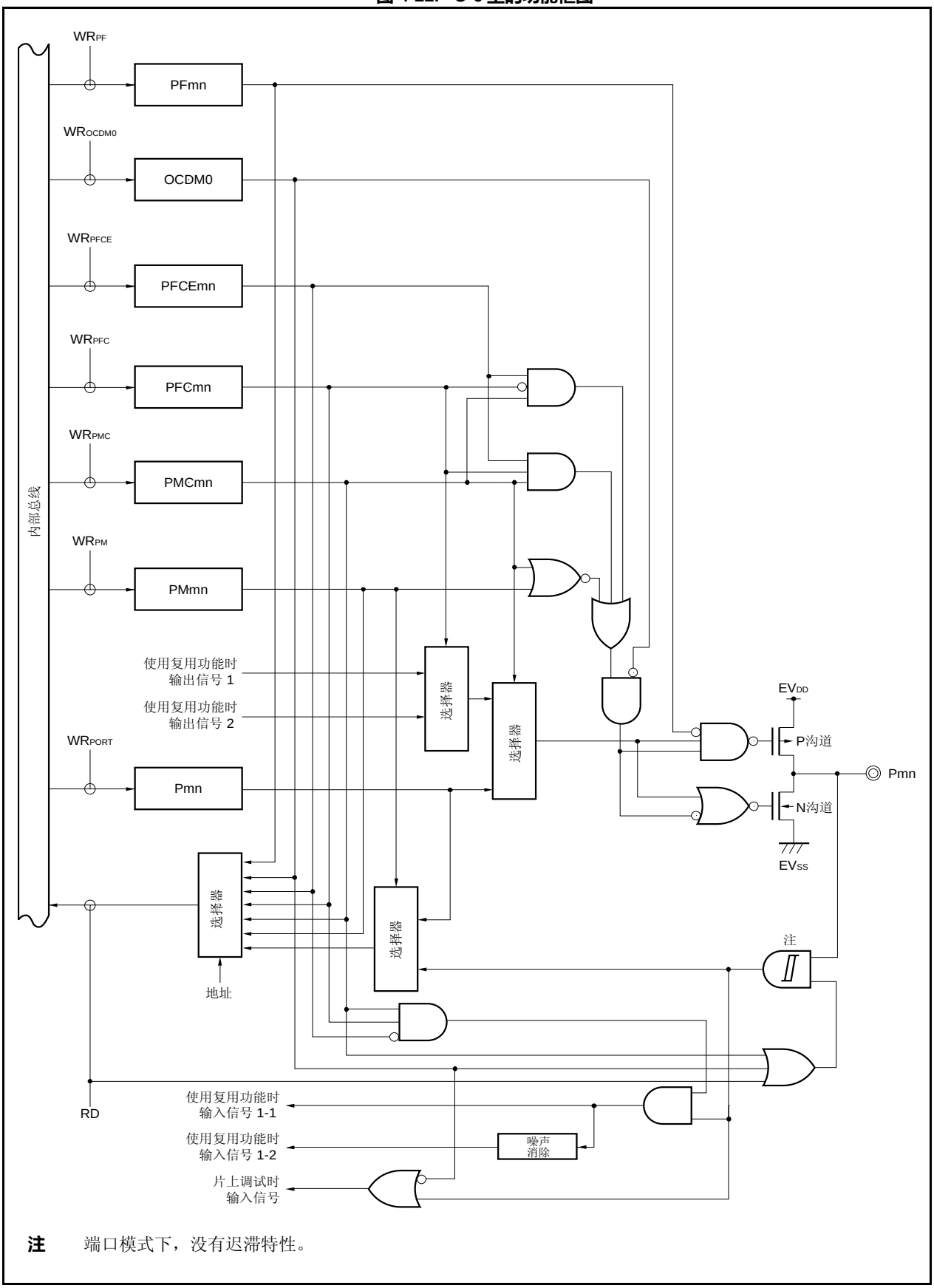
注　端口模式下，没有迟滞特性。

图 4-21.　U-5 型的功能框图



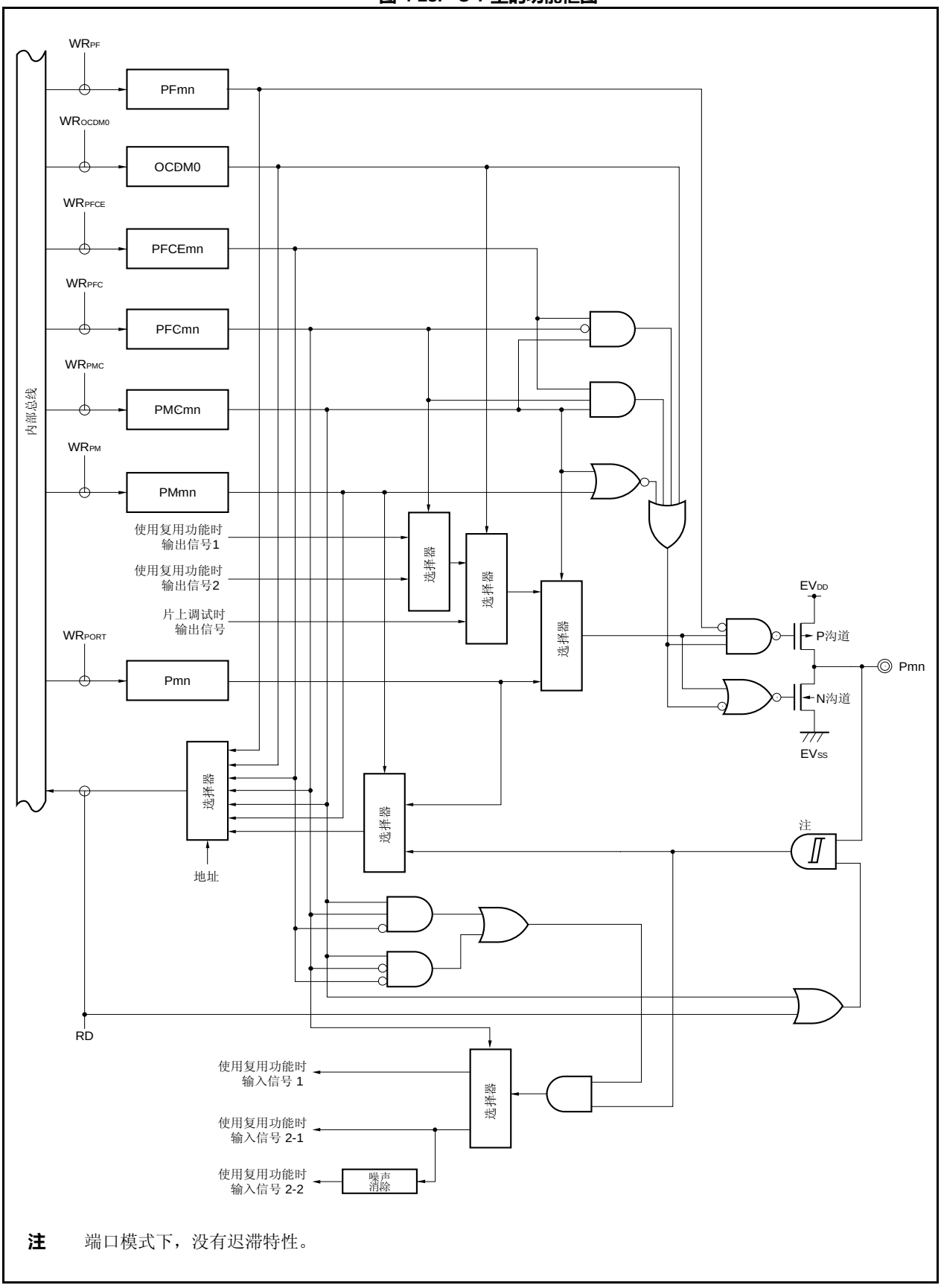
注　端口模式下，没有迟滞特性。

图 4-22.　U-6 型的功能框图



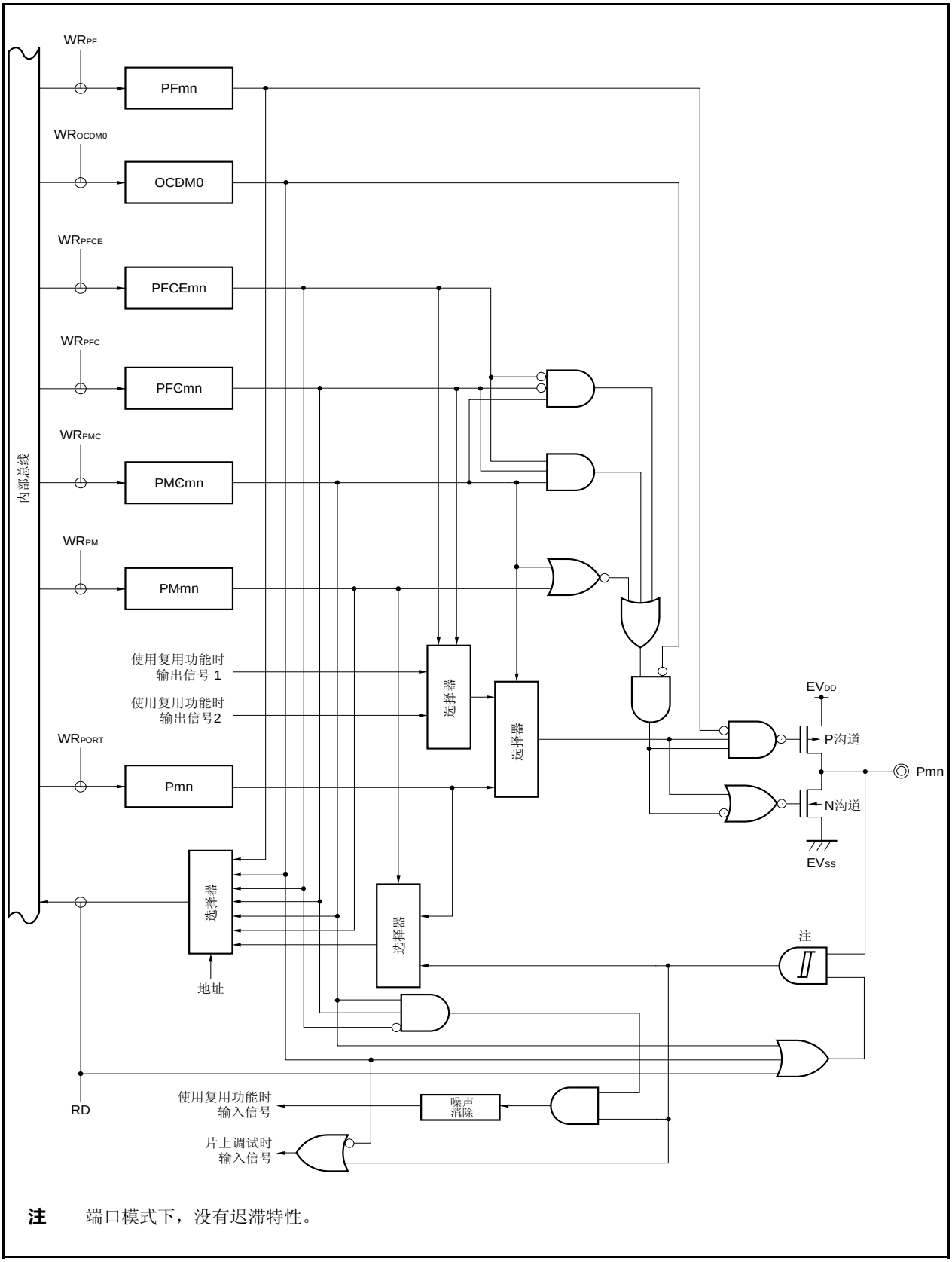
注　端口模式下，没有迟滞特性。

图 4-23.　U-7 型的功能框图



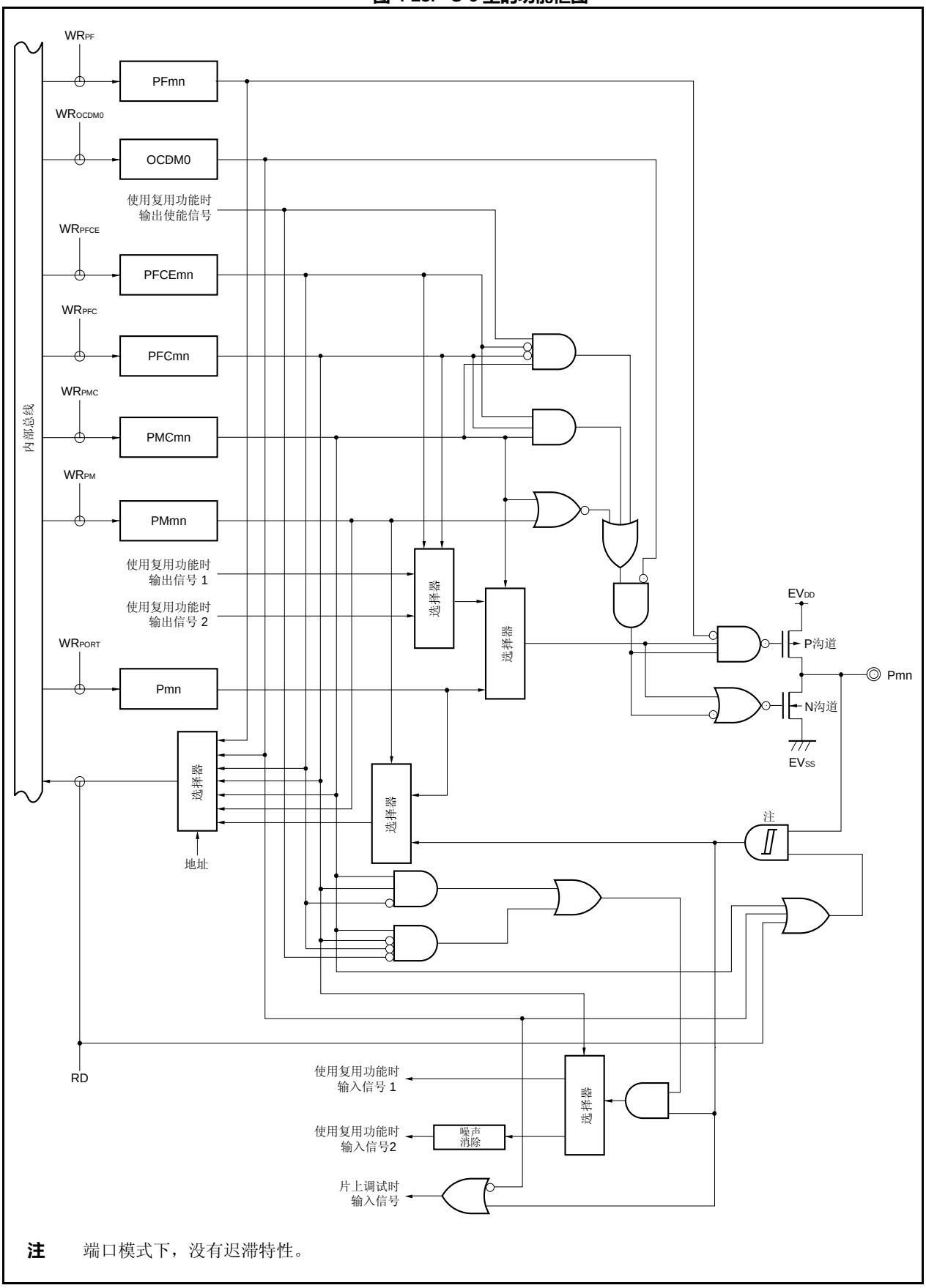
注　端口模式下，没有迟滞特性。

图 4-24.　U-8 型的功能框图



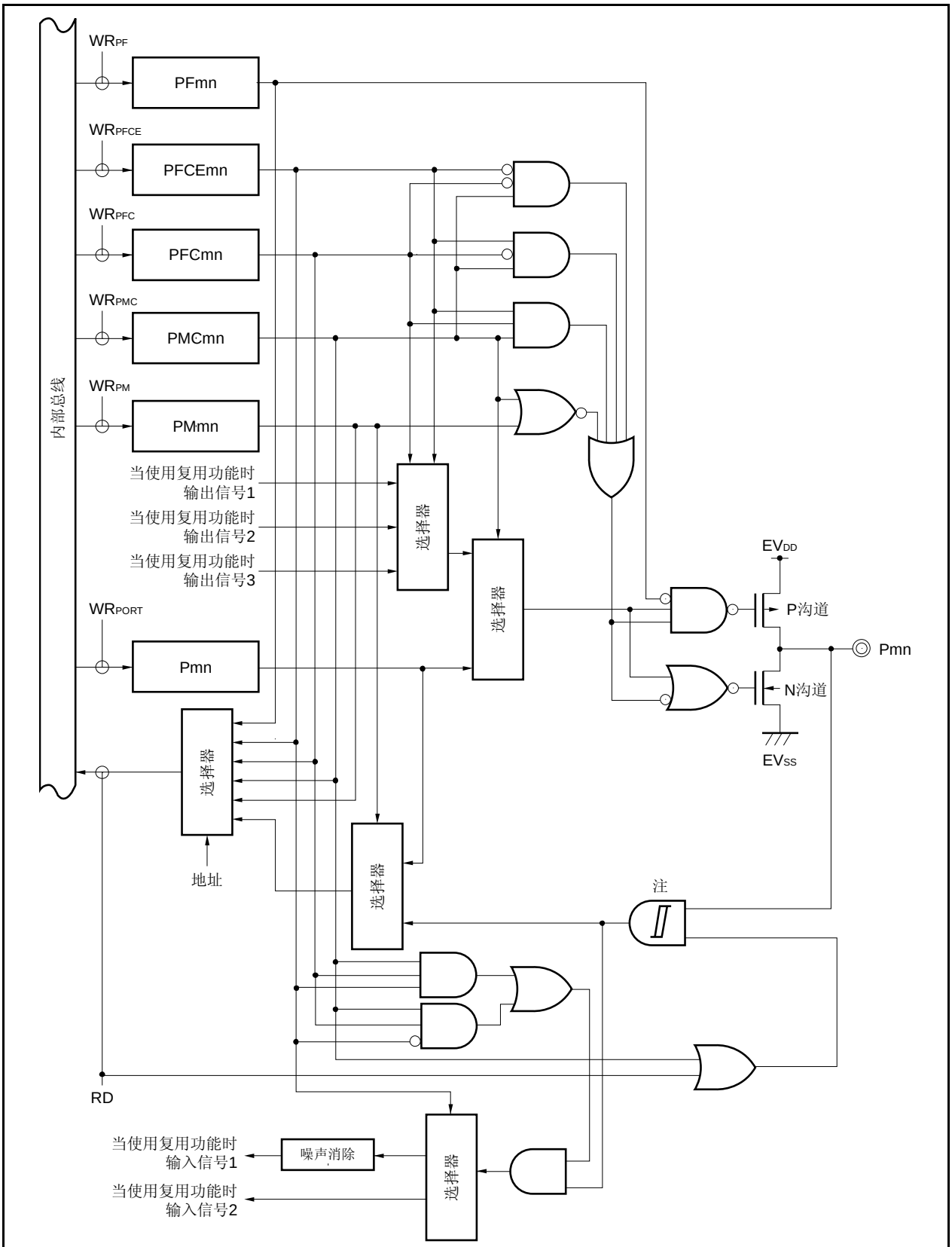
注　端口模式下，没有迟滞特性。

图 4-25.　U-9 型的功能框图



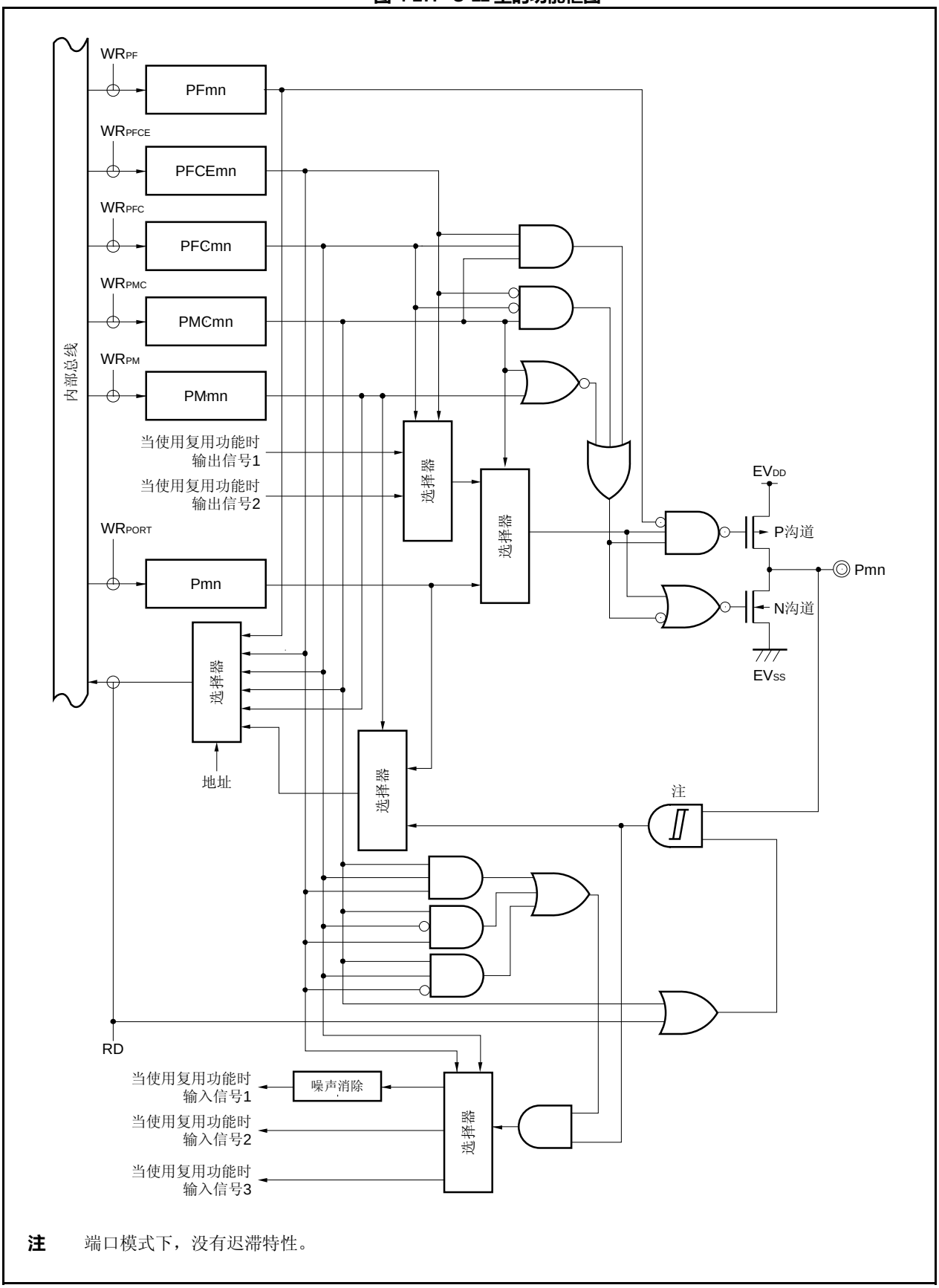
注　端口模式下，没有迟滞特性。

图 4-26.　U-10 型的功能框图



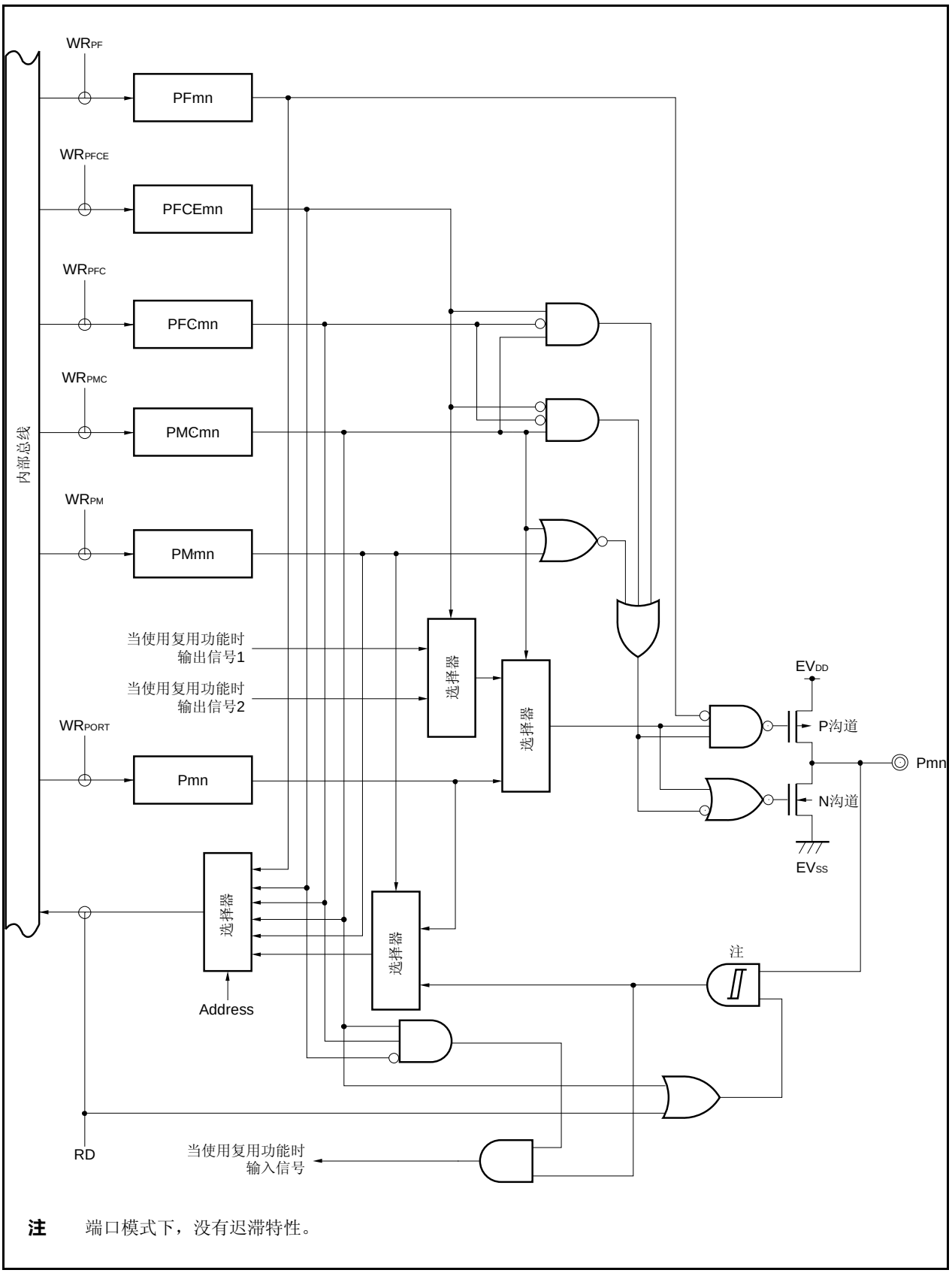
注　端口模式下，没有迟滞特性。

图 4-27. U-11 型的功能框图



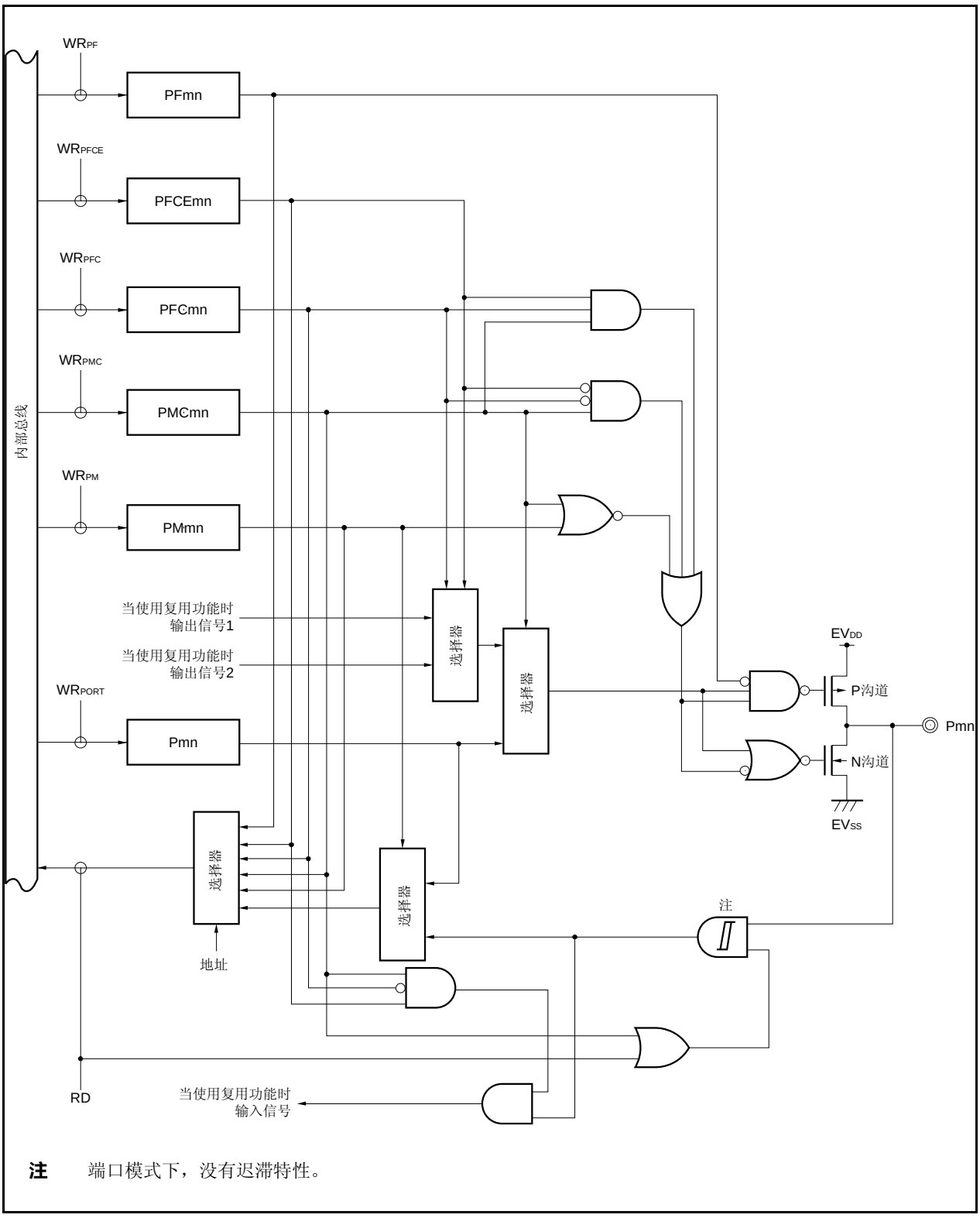
注　端口模式下，没有迟滞特性。

图 4-28.　U-12 型的功能框图



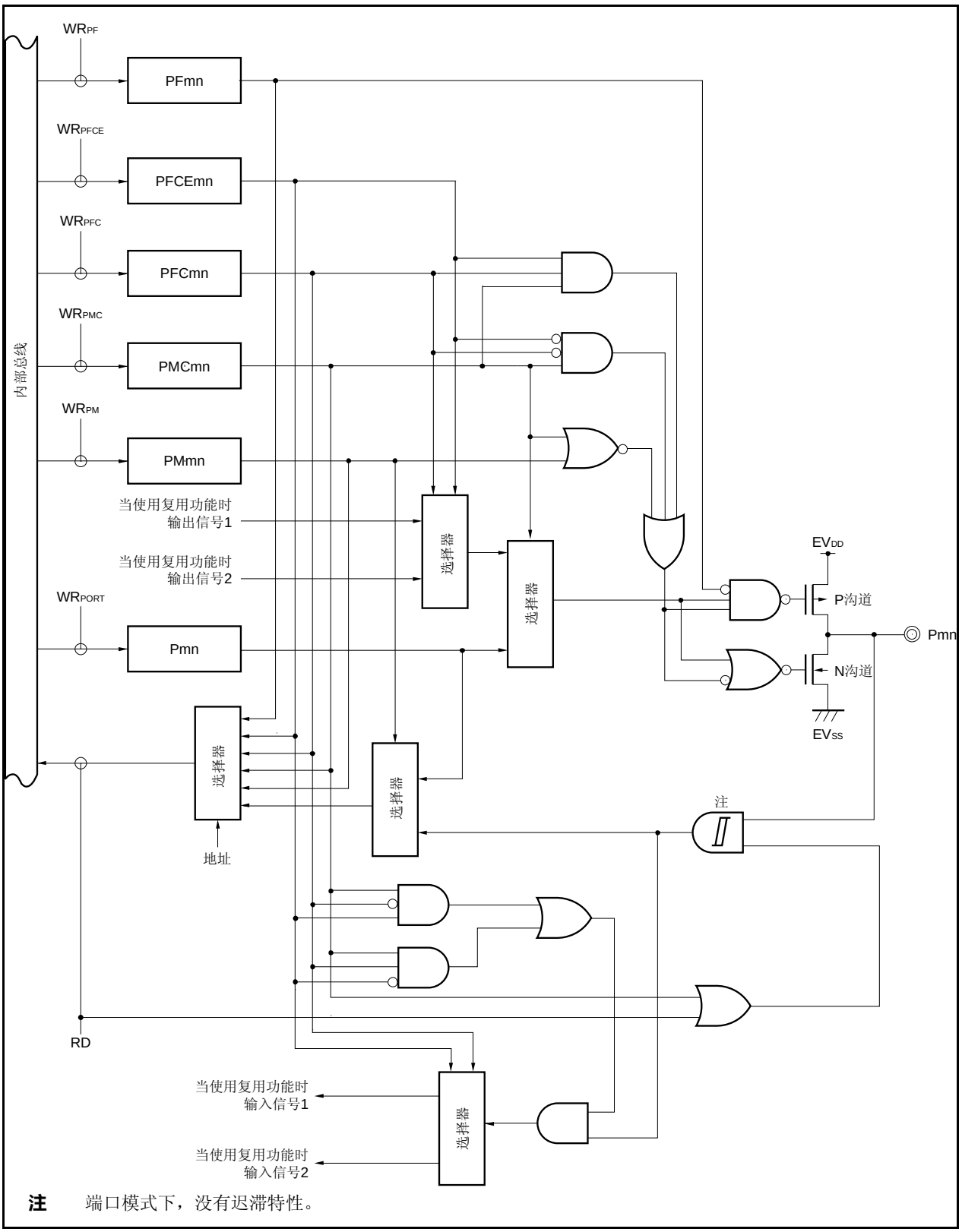
注　端口模式下，没有迟滞特性。

图 4-29.　U-13 型的功能框图



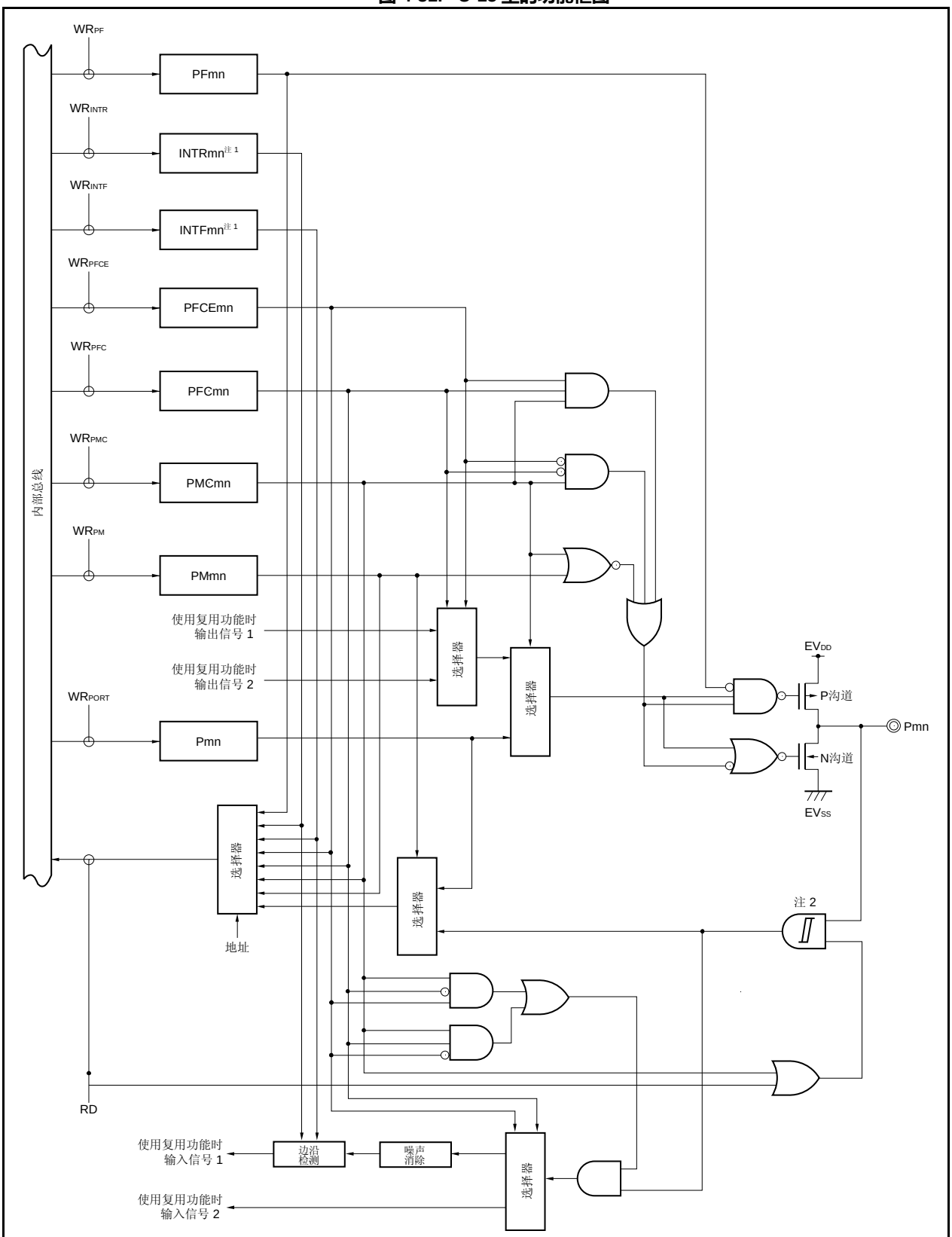
注　端口模式下，没有迟滞特性。

图 4-30.　U-14 型的功能框图



注　端口模式下，没有迟滞特性。

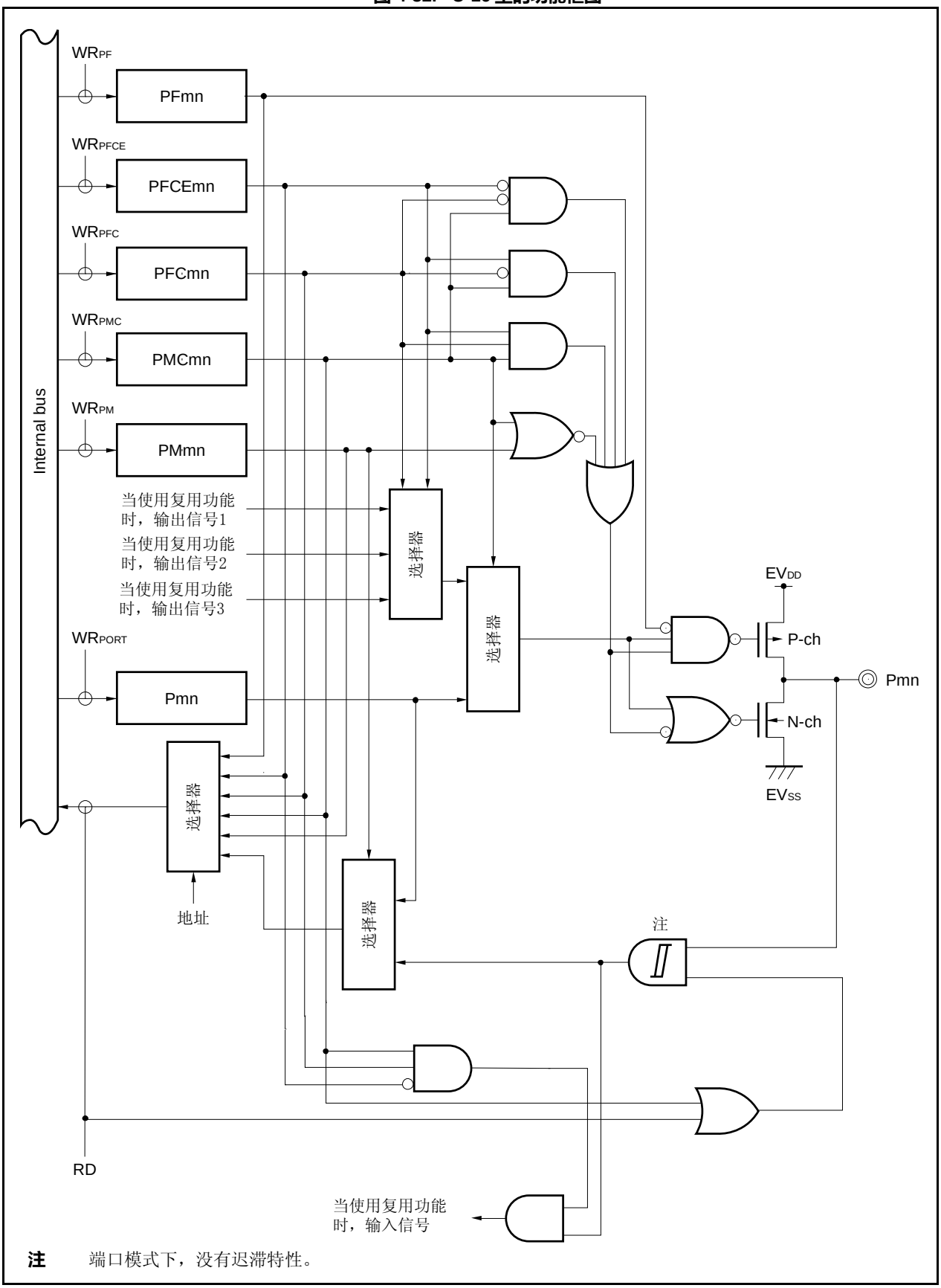
图 4-31.　U-15 型的功能框图



注　**1.**　参见 **22.7　外部中断请求输入引脚（NMI，INTP0 至 INTP7）。**

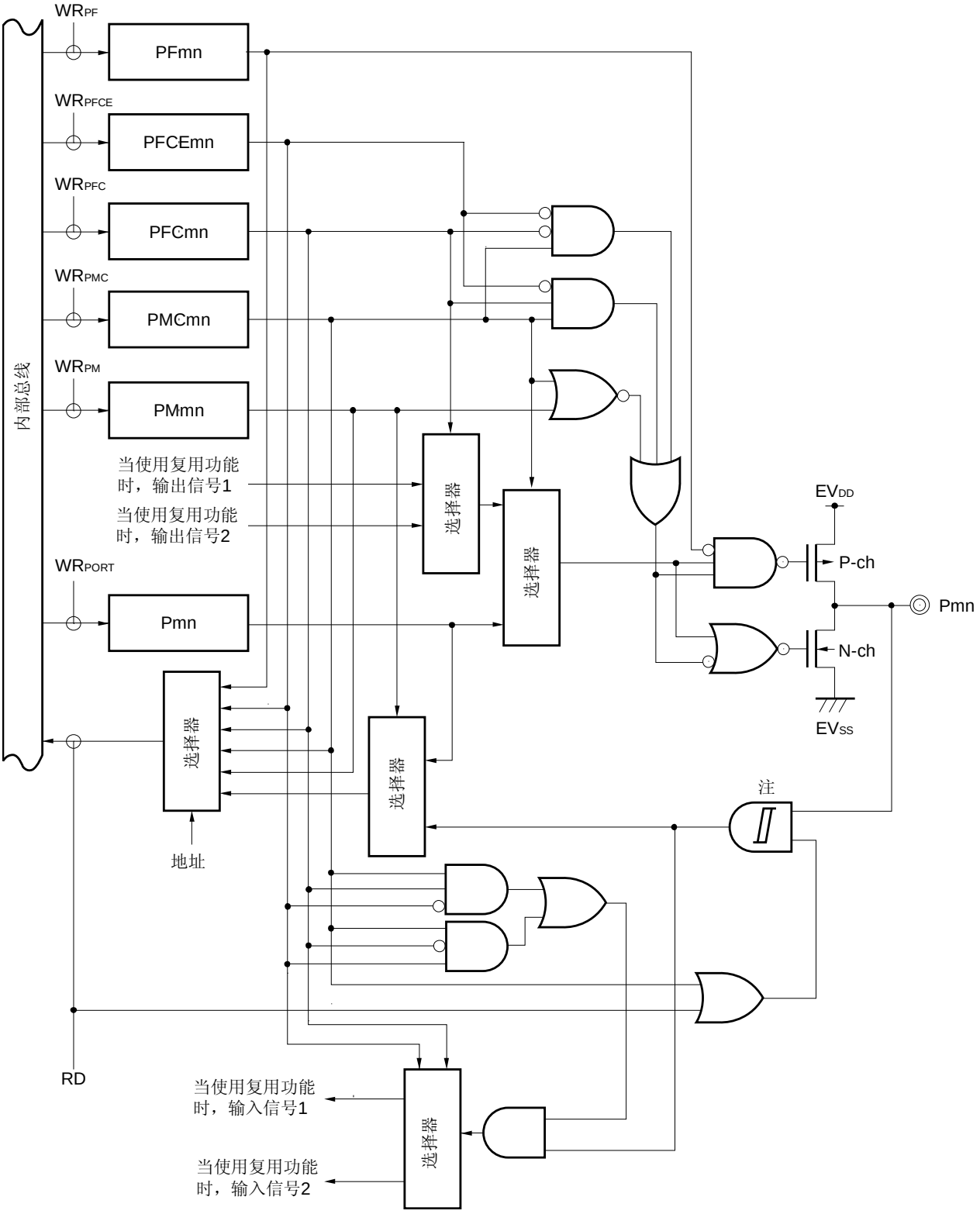
2.　端口模式下，没有迟滞特性。

图 4-32.　U-16 型的功能框图



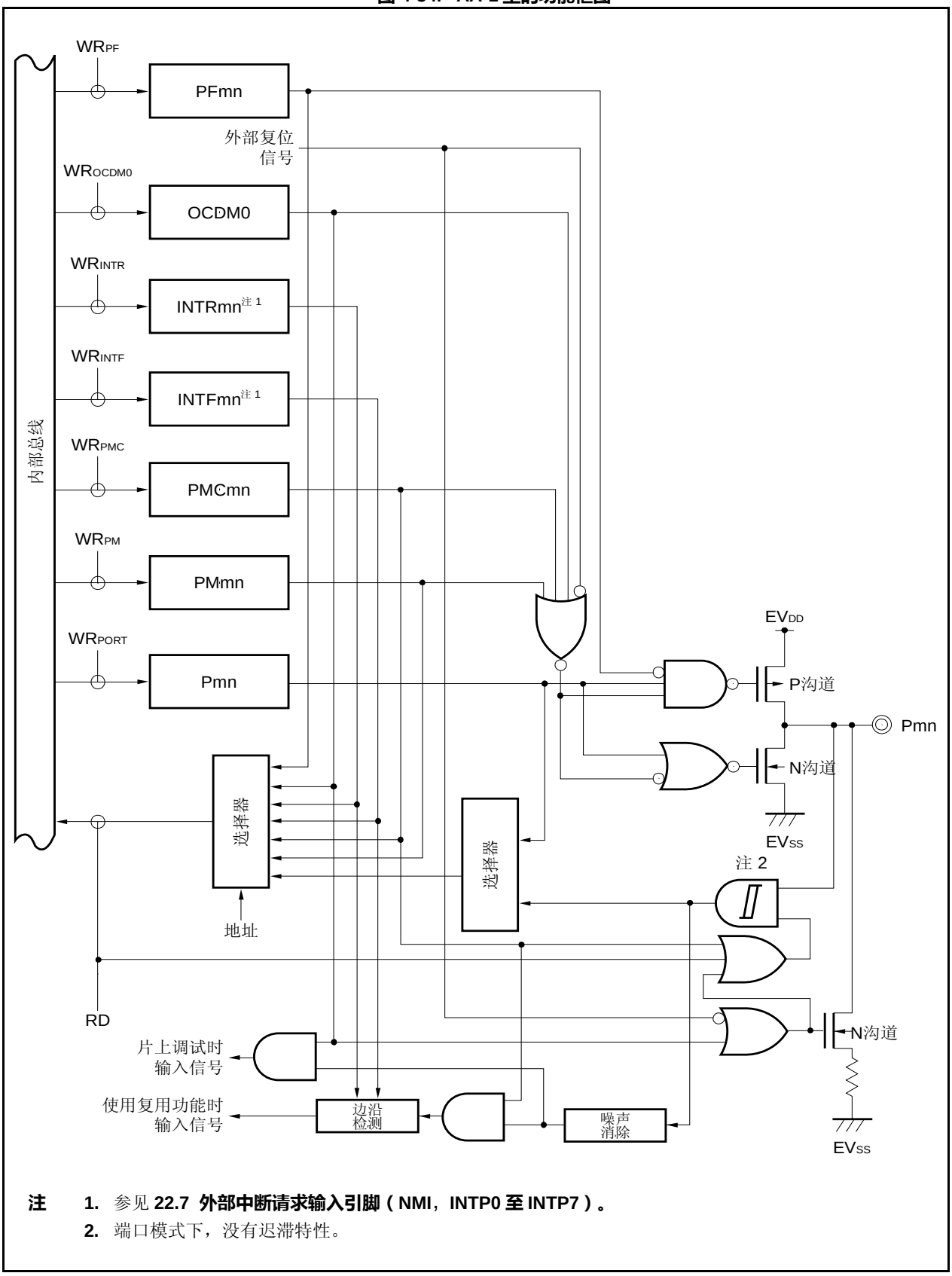
注　端口模式下，没有迟滞特性。

图 4-33.　U-17 型的功能框图



注　　端口模式下，没有迟滞特性。

图 4-34.　AA-1 型的功能框图



注　**1.**　参见 **22.7　外部中断请求输入引脚（NMI，INTP0 至 INTP7）。**

2.　端口模式下，没有迟滞特性。

4.5　使用复用功能时的端口寄存器设置

表 4-15 显示各个端口用作复用功能时的端口寄存器设置。当一个端口引脚用作复用功能引脚时，请参考各引脚的详细描述。

表 4-15. 端口引脚用作复用功能时的设置（1/7）

功能名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P02	NMI	输入	P02 = 无需设置	PM02 = 无需设置	PMC02 = 1	-	PFC02 = 0	
	A21	输入	P02 = 无需设置	PM02 = 无需设置	PMC02 = 1	-	PFC02 = 1	
P03	INTP0	输入	P03 = 无需设置	PM03 = 无需设置	PMC03 = 1	PFCE03 = 0	PFC03 = 0	
	ADTRG	输入	P03 = 无需设置	PM03 = 无需设置	PMC03 = 1	PFCE03 = 0	PFC03 = 1	
	UCLK	输入	P03 = 无需设置	PM03 = 无需设置	PMC03 = 1	PFCE03 = 1	PFC03 = 0	
	RTC1HZ	输入	P03 = 无需设置	PM03 = 无需设置	PMC03 = 1	PFCE03 = 1	PFC03 = 1	
P04	INTP1	输出	P04 = 无需设置	PM04 = 无需设置	PMC04 = 1	PFCE04 = 0	PFC04 = 0	
	RTCDIV	输出	P04 = 无需设置	PM04 = 无需设置	PMC04 = 1	PFCE04 = 0	PFC04 = 1	
	RTCCL	输入	P04 = 无需设置	PM04 = 无需设置	PMC04 = 1	PFCE04 = 1	PFC04 = 0	
P05	$\overline{\text{INTP2}}$	输入	P05 = 无需设置	PM05 = 无需设置	PMC05 = 1	-	-	
	$\overline{\text{DRST}}$	输入	P05 = 无需设置	PM05 = 无需设置	PMC05 = 无需设置	-	-	OCDM0 (OCDM) = 1
P06	INTP3	输出	P06 = 无需设置	PM06 = 无需设置	PMC06 = 1	-	-	
P10	ANO0	输出	P10 = 无需设置	PM10 = 1	-	-	-	
P11	ANO1	输出	P11 = 无需设置	PM11 = 1	-	-	-	
P30	TXDA0	输出	P30 = 无需设置	PM30 = 无需设置	PMC30 = 1	-	PFC30 = 0	
	SOB4	输入	P30 = 无需设置	PM30 = 无需设置	PMC30 = 1	-	PFC30 = 1	
P31	RXDA0	输入	P31 = 无需设置	PM31 = 无需设置	PMC31 = 1	-	注, PFC31 = 0	
	INTP7	输入	P31 = 无需设置	PM31 = 无需设置	PMC31 = 1	-	注, PFC31 = 0	
	SIB4	输入	P31 = 无需设置	PM31 = 无需设置	PMC31 = 1	-	PFC31 = 1	
P32	$\overline{\text{ASCKA0}}$	I/O	P32 = 无需设置	PM32 = 无需设置	PMC32 = 1	-	PFC32 = 0	
	SCKB4	输入	P32 = 无需设置	PM32 = 无需设置	PMC32 = 1	-	PFC32 = 1	
	TIP00	输出	P32 = 无需设置	PM32 = 无需设置	PMC32 = 1	PFCE32 = 0	PFC32 = 0	
	TOP00	输入	P32 = 无需设置	PM32 = 无需设置	PMC32 = 1	PFCE32 = 0	PFC32 = 1	

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

注意事项 当使用 P10/AN00 和 P11/AN01 引脚之一作为 I/O 端口，而另一个用作 D/A 输出引脚。在 D/A 输出期间，如果端口 I/O 电平不改变，则在应用系统中可以这样使用。

表 4-15. 端口引脚用作复用功能时的设置（2/7）

功能名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P36	TXDA3	输出	P36 = 无需设置	PM36 = 无需设置	PMC36 = 1	-	-	
P37	RXDA3	输入	P37 = 无需设置	PM37 = 无需设置	PMC37 = 1	-	-	
P38	TXDA2	输出	P38 = 无需设置	PM38 = 无需设置	PMC38 = 1	-	PFC38 = 0	PF38 (PF3) = 1
	SDA00	I/O	P38 = 无需设置	PM38 = 无需设置	PMC38 = 1	-	PFC38 = 1	
P39	RXDA2	输入	P39 = 无需设置	PM39 = 无需设置	PMC39 = 1	-	PFC39 = 0	PF39 (PF3) = 1
	SCL00	I/O	P39 = 无需设置	PM39 = 无需设置	PMC39 = 1	-	PFC39 = 1	
P40	SIB0	输入	P40 = 无需设置	PM40 = 无需设置	PMC40 = 1	-	PFC40 = 0	PF40 (PF4) = 1
	SDA01	I/O	P40 = 无需设置	PM40 = 无需设置	PMC40 = 1	-	PFC40 = 1	
P41	SOB0	输出	P41 = 无需设置	PM41 = 无需设置	PMC41 = 1	-	PFC41 = 0	PF41 (PF4) = 1
	SCL01	I/O	P41 = 无需设置	PM41 = 无需设置	PMC41 = 1	-	PFC41 = 1	
P42	$\overline{\text{SCKB0}}$	I/O	P42 = 无需设置	PM42 = 无需设置	PMC42 = 1	-	-	KRM0 (KRM) = 0
P50	TIQ01	输入	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 0	PFC50 = 1	TQ0TIG2, TQ0TIG3 (TQ0IOC1) = 0
	KR0	输入	P50 = 无需设置	PM50 = 无需设置	PMC50 = 1	PFCE50 = 0	PFC50 = 1	
	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	

表 4-15. 端口引脚用作复用功能时的设置（3/7）

功能名称	复用功能		Pn 寄存器的 Pnx位	PMn 寄存器的 PMnx位	PMCn 寄存器的 PMCnx位	PFCEn 寄存器的 PFCEnx位	PFCn 寄存器的 PFCnx位	其它位 (寄存器)
	名称	I/O						
P51	TIQ02	输入	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 0	PFC51 = 1	KRM1 (KRM) = 0
	KR1	输入	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 0	PFC51 = 1	TQ0TIG4, TQ0TIG5 (TQ0IOC1) = 0
	TOQ02	输出	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 1	PFC51 = 0	
	RTP01	输出	P51 = 无需设置	PM51 = 无需设置	PMC51 = 1	PFCE51 = 1	PFC51 = 1	
P52	TIQ03	输入	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 0	PFC52 = 1	KRM2 (KRM) = 0
	KR2	输入	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 0	PFC52 = 1	TQ0TIG6, TQ0TIG7 (TQ0I0C1) = 0
	TOQ03	输出	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 1	PFC52 = 0	
	RTP02	输出	P52 = 无需设置	PM52 = 无需设置	PMC52 = 1	PFCE52 = 1	PFC52 = 1	
	DDI	输入	P52 = 无需设置	PM52 = 无需设置	PMC52 = 无需设置	PFCE52 = 无需设置	PFC52 = 无需设置	OCDM0 (OCDM) = 1
P53	SIB2	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 0	PFC53 = 0	
	TIQ00	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 0	PFC53 = 1	KRM3 (KRM) = 0
	KR3	输入	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 0	PFC53 = 1	TQ0TIG0, TQ0TIG1 (TQ0IOC1) = 0, TQ0EES0, TQ0EES1 (TQ0IOC2) = 0, TQ0ETS0, TQ0ETS1 (TQ0IOC2) = 0
	TOQ00	输出	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 1	PFC53 = 0	
	RTP03	输出	P53 = 无需设置	PM53 = 无需设置	PMC53 = 1	PFCE53 = 1	PFC53 = 1	
	DDO	输出	P53 = 无需设置	PM53 = 无需设置	PMC53 = 无需设置	PFCE53 = 无需设置	PFC53 = 无需设置	OCDM0 (OCDM) = 1
P54	SOB2	输出	P54 = 无需设置	PM54 = 无需设置	PMC54 = 1	PFCE54 = 0	PFC54 = 0	
	KR4	输入	P54 = 无需设置	PM54 = 无需设置	PMC54 = 1	PFCE54 = 0	PFC54 = 1	
	RTP04	输出	P54 = 无需设置	PM54 = 无需设置	PMC54 = 1	PFCE54 = 1	PFC54 = 1	
	DCK	输入	P54 = 无需设置	PM54 = 无需设置	PMC54 = 无需设置	PFCE54 = 无需设置	PFC54 = 无需设置	OCDM0 (OCDM) = 1
P55	$\overline{\text{SCKB2}}$	I/O	P55 = 无需设置	PM55 = 无需设置	PMC55 = 1	PFCE55 = 0	PFC55 = 0	
	KR5	输入	P55 = 无需设置	PM55 = 无需设置	PMC55 = 1	PFCE55 = 0	PFC55 = 1	
	RTP05	输出	P55 = 无需设置	PM55 = 无需设置	PMC55 = 1	PFCE55 = 1	PFC55 = 1	
	DMS	输入	P55 = 无需设置	PM55 = 无需设置	PMC55 = 无需设置	PFCE55 = 无需设置	PFC55 = 无需设置	OCDM0 (OCDM) = 1

表 4-15. 端口引脚用作复用功能时的设置（4/7）

功能名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P70	ANI0	输入	P70 = 无需设置	PM70 = 1	-	-	-	
P71	ANI1	输入	P71 = 无需设置	PM71 = 1	-	-	-	
P72	ANI2	输入	P72 = 无需设置	PM72 = 1	-	-	-	
P73	ANI3	输入	P73 = 无需设置	PM73 = 1	-	-	-	
P74	ANI4	输入	P74 = 无需设置	PM74 = 1	-	-	-	
P75	ANI5	输入	P75 = 无需设置	PM75 = 1	-	-	-	
P76	ANI6	输入	P76 = 无需设置	PM76 = 1	-	-	-	
P77	ANI7	输入	P77 = 无需设置	PM77 = 1	-	-	-	
P78	ANI8	输入	P78 = 无需设置	PM78 = 1	-	-	-	
P79	ANI9	输入	P79 = 无需设置	PM79 = 1	-	-	-	
P710	ANI10	输入	P710 = 无需设置	PM710 = 1	-	-	-	
P711	ANI11	输入	P711 = 无需设置	PM711 = 1	-	-	-	
P90	A0	输出	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 0	PFC90 = 0	注 1
	KR6	输入	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 0	PFC90 = 1	
	TXDA1	输出	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 1	PFC90 = 0	
	SDA02	I/O	P90 = 无需设置	PM90 = 无需设置	PMC90 = 1	PFCE90 = 1	PFC90 = 1	PF90 (PF9) = 1
P91	A1	输出	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 0	PFC91 = 0	注 1
	KR7	输入	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 0	PFC91 = 1	
	RXDA1/KR7注2	输入	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 1	PFC91 = 0	
	SCL02	I/O	P91 = 无需设置	PM91 = 无需设置	PMC91 = 1	PFCE91 = 1	PFC91 = 1	PF91 (PF9) = 1

注 1. 当 A0 至 A15 引脚用作复用功能时，请确保一次将 PMC9 寄存器的所有 16 位设置为 FFFFH。

2. RXDA1 和 KR7 功能不能同时使用。引脚用作 RXDA1 时，不要使用 KR7 功能。而引脚用作 KR7 时，不要使用 RXDA1 功能（建议将 PFC91 位设置为 1，并 且将 PFCE91 位设置为 0）。

表 4-15. 端口引脚用作复用功能时的设置（5/7）

功能名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P92	A2	输出	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 0	PFC92 = 0	注1
	TIP41	输入	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 0	PFC92 = 1	
	TOP41	输出	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 1	PFC92 = 0	
	TXDA4	输出	P92 = 无需设置	PM92 = 无需设置	PMC92 = 1	PFCE92 = 1	PFC92 = 1	
P93	A3	输出	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 0	PFC93 = 0	注1
	TIP40	输入	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 0	PFC93 = 1	
	TOP40	输出	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 1	PFC93 = 0	
	RXDA4	输入	P93 = 无需设置	PM93 = 无需设置	PMC93 = 1	PFCE93 = 1	PFC93 = 1	
P94	A4	输出	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 0	PFC94 = 0	注1
	TIP31	输入	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 0	PFC94 = 1	
	TOP31	输出	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 1	PFC94 = 0	
	TXDA5	输出	P94 = 无需设置	PM94 = 无需设置	PMC94 = 1	PFCE94 = 1	PFC94 = 1	
P95	A5	输出	P95 = 无需设置	PM95 = 无需设置	PMC95 = 1	PFCE95 = 0	PFC95 = 0	注1
	TIP30	输入	P95 = 无需设置	PM95 = 无需设置	PMC95 = 1	PFCE95 = 0	PFC95 = 1	
	TOP30	输出	P95 = 无需设置	PM95 = 无需设置	PMC95 = 1	PFCE95 = 1	PFC95 = 0	
	RXDA5	输入	P95 = 无需设置	PM95 = 无需设置	PMC95 = 1	PFCE95 = 1	PFC95 = 1	
P96	A6	输出	P96 = 无需设置	PM96 = 无需设置	PMC96 = 1	PFCE96 = 0	PFC96 = 0	注1
	TXDC0	输入	P96 = 无需设置	PM96 = 无需设置	PMC96 = 1	PFCE96 = 0	PFC96 = 1	
	TIP21	输出	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1
	SIB1[注2]	输入	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 0	PFC97 = 1	
	RXDC0[注2]	输入	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 0	PFC97 = 1	
	$\overline{\text{TIP20}}$	输入	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 1	PFC97 = 0	
	TOP20	输出	P97 = 无需设置	PM97 = 无需设置	PMC97 = 1	PFCE97 = 1	PFC97 = 1	
P98	A8	输出	P98 = 无需设置	PM98 = 无需设置	PMC98 = 1	-	PFC98 = 0	注1
	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1
	$\overline{\text{SCKB1}}$	I/O	P99 = 无需设置	PM99 = 无需设置	PMC99 = 1	-	PFC99 = 1	

注 1. 当 A0 至 A15 引脚用作复用功能时，请确保一次将 PMC9 寄存器的所有 16 位设置为 FFFFH。

2. SIB1 和 RXDC0 功能不能同时使用。使用 SIB1 引脚时，停止 UARTC0 接收（将 UC0CTL0.UC0RXE 位清除为 0）。而使用 RXDC0 引脚时，停止 CSIB1 接收（将 CB1CTL0.CB1RXE 位清除为 0）。

表 4-15. 端口引脚用作复用功能时的设置（6/7）

功能名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
P910	A10	输出	P910 = 无需设置	PM910 = 无需设置	PMC910 = 1	-	PFC910 = 0	注
	SIB3	输入	P910 = 无需设置	PM910 = 无需设置	PMC910 = 1	-	PFC910 = 1	
P911	A11	输出	P911 = 无需设置	PM911 = 无需设置	PMC911 = 1	-	PFC911 = 0	注
	SOB3	输出	P911 = 无需设置	PM911 = 无需设置	PMC911 = 1	-	PFC911 = 1	
P912	A12	输出	P912 = 无需设置	PM912 = 无需设置	PMC912 = 1	-	PFC912 = 0	注
	$\overline{\text{SCKB3}}$	I/O	P912 = 无需设置	PM912 = 无需设置	PMC912 = 1	-	PFC912 = 1	
P913	A13	输出	P913 = 无需设置	PM913 = 无需设置	PMC913 = 1	-	PFC913 = 0	注
	INTP4	输入	P913 = 无需设置	PM913 = 无需设置	PMC913 = 1	-	PFC913 = 1	
P914	A14	输出	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 0	PFC914 = 0	注
	INTP5	输入	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 0	PFC914 = 1	
	TIP51	输入	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 1	PFC914 = 0	
	TOP51	输出	P914 = 无需设置	PM914 = 无需设置	PMC914 = 1	PFCE914 = 1	PFC914 = 1	
P915	A15	输出	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 0	PFC915 = 0	注
	INTP6	输入	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 0	PFC915 = 1	
	TIP50	输入	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 1	PFC915 = 0	
	TOP50	输出	P915 = 无需设置	PM915 = 无需设置	PMC915 = 1	PFCE915 = 1	PFC915 = 1	
PCM0	$\overline{\text{WAIT}}$	输入	PCM0 = 无需设置	PMCM0 = 无需设置	PMCCM0 = 1	-	-	
PCM1	CLKOUT	输出	PCM1 = 无需设置	PMCM1 = 无需设置	PMCCM1 = 1	-	-	
PCM2	$\overline{\text{HLDAK}}$	输出	PCM2 = 无需设置	PMCM2 = 无需设置	PMCCM2 = 1	-	-	
PCM3	$\overline{\text{HLDRQ}}$	输入	PCM3 = 无需设置	PMCM3 = 无需设置	PMCCM3 = 1	-	-	
PCT0	$\overline{\text{WR0}}$	输出	PCT0 = 无需设置	PMCT0 = 无需设置	PMCCT0 = 1	-	-	
PCT1	$\overline{\text{WR1}}$	输出	PCT1 = 无需设置	PMCT1 = 无需设置	PMCCT1 = 1	-	-	
PCT4	$\overline{\text{RD}}$	输出	PCT4 = 无需设置	PMCT4 = 无需设置	PMCCT4 = 1	-	-	
PCT6	ASTB	输出	PCT6 = 无需设置	PMCT6 = 无需设置	PMCCT6 = 1	-	-	

注 当 A0 至 A15 引脚用作复用功能时，请确保一次将 PMC9 寄存器的所有 16 位设置为 FFFFH。

表 4-15. 端口引脚用作复用功能时的设置（7/7）

功能名称	复用功能		Pn 寄存器的 Pnx 位	PMn 寄存器的 PMnx 位	PMCn 寄存器的 PMCnx 位	PFCEn 寄存器的 PFCEnx 位	PFCn 寄存器的 PFCnx 位	其它位 (寄存器)
	名称	I/O						
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	-	-	
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	-	-	
PDL6	FLMD1注	输入	PDL5 = 无需设置	PMDL5 = 无需设置	PMCDL5 = 无需设置	-	-	
	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 该引脚设置为 flash 存储器编程模式后，不需要通过端口控制寄存器进行操作。详情参见**第三十一章 FLASH 存储器**。

4.6　注意事项

4.6.1　设置端口引脚的注意事项

（1）在 V850ES/JG3-L 中，通用端口功能与一些外围 I/O 功能共用一个引脚。通过 PMCn 寄存器的设置，引脚可以在通用端口引脚（端口模式）与外围功能 I/O 引脚（复用功能模式）之间相互切换。设置该寄存器时，请注意下列事项：

（a）从端口模式转换至复用功能模式的注意事项

从端口模式至复用功能模式的切换操作，按下列顺序进行：

<1> 设置 PFn 寄存器[注1]：　N 沟道漏极开路设置
<2> 设置 PFCn 和 PFCEn 寄存器：　复用功能选择
<3> 设置 PMCn 寄存器的相应位为 1：　转换至复用功能模式
<4> 设置 INTRn 和 INTFn 寄存器[注2]：　外部中断设置

请注意，如果先设置 PMCn 寄存器，设置该寄存器时，或者按照 PFn、PFCn 和 PFCEn 寄存器的设置改变引脚状态时，都可能会发生不可预料的操作。
下面有一个具体示例。

注　**1.** 只有 N 沟道漏极开路输出
2. 只选择了外部中断功能时

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

- **Pn 寄存器读取：　读取得到的是端口输出锁存值（当 PMn.PMnm 位 = 0 时），或者读取得到引脚状态（PMn.PMnm 位 = 1）。**
- **Pn 寄存器写入：　写入的是端口输出锁存器。**

[实例]　SCL01 引脚设置举例。

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

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

从 P41 引脚功能转换至 SCL01 引脚功能时，如下的设置顺序可能会导致误操作：

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

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

（b） 复用功能模式（输入）的注意事项

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

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

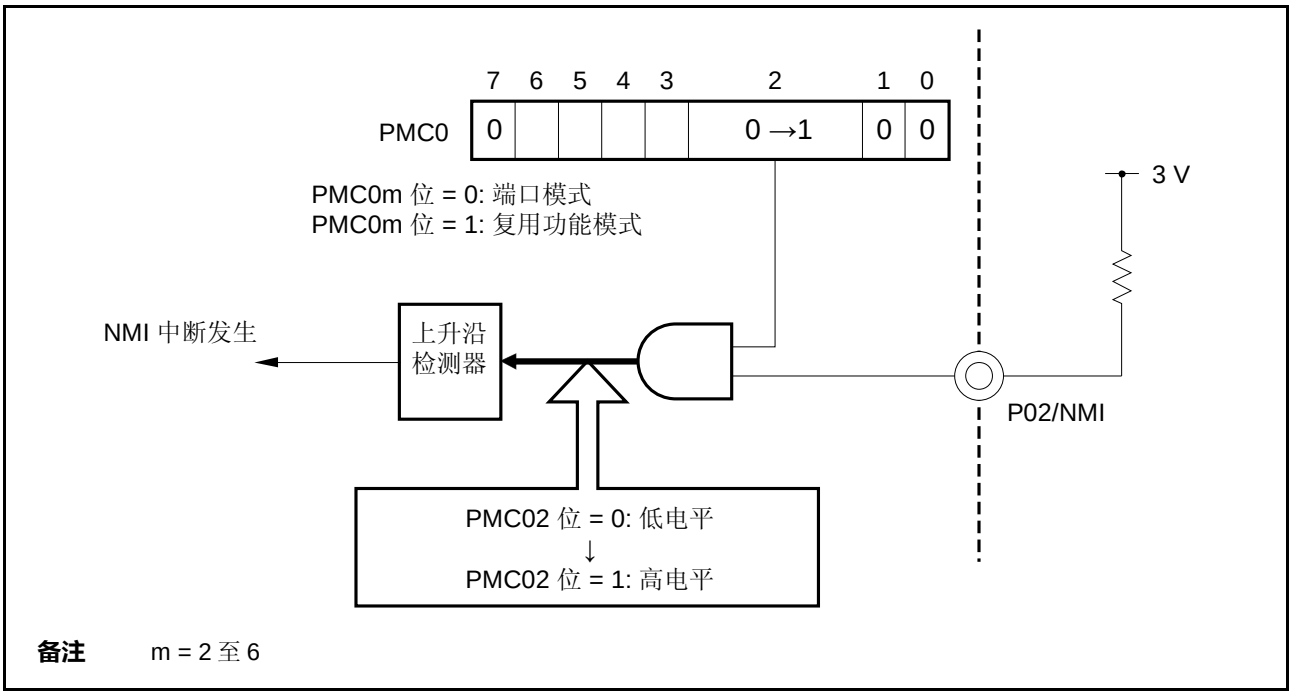
下面的例 1 和例 2 是具体示例。

[例 1]　从通用端口引脚（P02）切换至外部中断引脚（NMI）

当 P02/NMI 引脚上拉如图 4-35 时，且上升沿被指定为 NMI 引脚边沿检测设置时，即使高电平连续输入至 NMI 引脚，当从 P02 引脚切换至 NMI 引脚（PMC02 位 = 0 → 1）时，会被检测认为是上升沿，就像低电平跳变到高电平那样，产生一个 NMI 中断。

为避免该情况，从 P02 引脚切换至 NMI 功能完成后，再设置 NMI 引脚的有效检测沿。

图 4-35. 从 P02 至 NMI 转换举例（不正确）

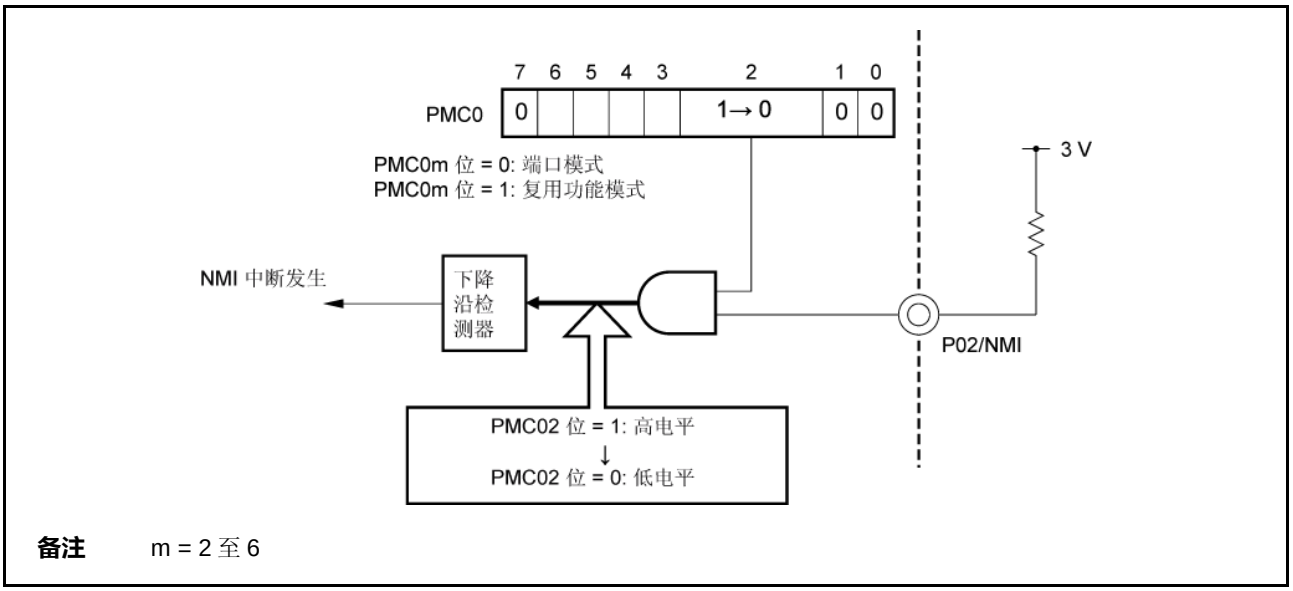


[例 2]　从外部引脚（NMI）切换至通用端口引脚（P02）

当 P02/NMI 引脚上拉如图 4-36 时，且下降沿被指定为 NMI 引脚边沿检测设置时，即使低电平连续输入至 NMI 引脚，当从 NMI 引脚切换至 P02 引脚（PMC02 位 = 1 → 0）时，会被检测认为是下降沿，就像高电平跳变到低电平那样，产生一个 NMI 中断。

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

图 4-36. 从 NMI 至 P02 转换举例（不正确）



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

4.6.2　端口n寄存器（Pn）位操作指令的注意事项

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

所以，当一个端口由输入模式转变为输出模式时，建议重写输出锁存器。

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

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

V850ES/JG3-L 单片机按以下顺序执行位处理指令：

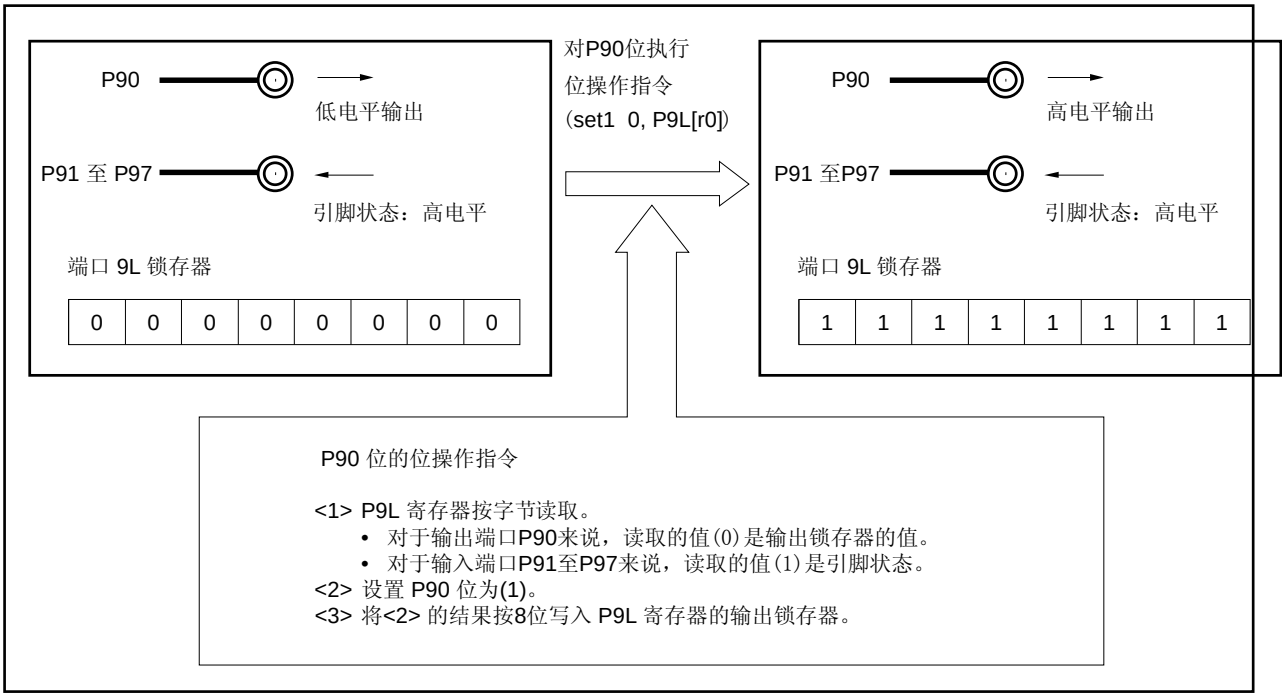
<1>　寄存器 Pn 按 8 位读取。
<2>　对目标位进行操作。
<3>　按 8 位写入寄存器 Pn。

在步骤<1>中，作为输出引脚的 P90 引脚的输出锁存器（0）的值被读取时，也读到了作为输入端口的 P91 至 P97 引脚状态。如果此时 P91 至 P97 引脚的状态为高电平，则读取得到的值为 FEH。

在步骤 <2>时，位操作过程使该值变为 FFH。

在步骤 <3>时，位操作将 FFH 写入输出锁存器。

图 4-37. 位处理指令（P90 引脚）



4.6.3　片上调试引脚的注意事项

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

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

如果不使用片上调试功能，必须进行以下操作。

- 将 OCDM 寄存器（特殊寄存器）的 OCDM0 位清 0 。

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

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

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

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

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

4.6.5　电源开启时P10，P11 和 P53 引脚的注意事项

电源开启后的复位期间，下列引脚可能暂时输出不确定电平。

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

4.6.6　迟滞特性

在端口模式下，以下端口引脚没有迟滞特性。

P02 至 P06
P31 至 P32，P37 至 P39
P40 至 P42
P50 至 P55
P90 至 P97，P99，P910，P912 至 P915

第五章 总线控制功能

外部总线接口功能通过端口 9、CM、CT、DL 和 DH 将外部设备连接到内部 ROM、RAM 或片上 I/O 寄存器之外的区域。这些端口可以控制地址/数据 I/O、读取/写入选通信号、等待、时钟输出、总线保持和地址选通信号。

V850ES/JG3-L 提供外部总线接口，通过该接口可以连接外部存储器设备，诸如 ROM、RAM 和外部 I/O 设备等。

5.1 特性

○ 可以选择最小 3 个总线周期的复用总线输出，和分离总线输出
○ 可以选择 8 位/16 位数据总线（各个存储器块可以分别指定）
○ 等待功能：
- 最高可达 7 个周期的可编程等待功能（各个存储器块可以分别指定）
- 使用 $\overline{\text{WAIT}}$ 引脚的外部等待功能

○ 空闲状态插入功能
- 在读取周期之后插入空闲状态，可以连接低速设置。

○ 总线保持功能
○ 支持未对齐访问
○ 最大可连接 4MB 的物理存储器

5.2 总线控制引脚

在各种总线模式下，可以使用下列信号来控制外部设备。

表 5-1. 总线控制引脚（选择为复用总线时）

总线控制信号	I/O	功能	复用功能	在端口模式/复用功能模式之间切换所需的寄存器
AD0 至 AD15	I/O	地址/数据总线	PDL0 至 PDL15	PMCDL 寄存器
A16 至 A21	输出	地址总线	PDH0 至 PDH4, P02	PMCDH 寄存器, PMC0 寄存器
$\overline{\text{WAIT}}$	输入	外部等待控制	PCM0	PMCCM 寄存器
CLKOUT	输出	内部系统时钟输出	PCM1	PMCCM 寄存器
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	输出	写选通信号	PCT0, PCT1	PMCCT 寄存器
$\overline{\text{RD}}$	输出	读选通信号	PCT4	PMCCT 寄存器
ASTB	输出	地址选通信号	PCT6	PMCCT 寄存器
$\overline{\text{HLDRQ}}$	输入	总线保持控制	PCM3	PMCCM 寄存器
$\overline{\text{HLDAK}}$	输出		PCM2	

表 5-2. 总线控制引脚（选择为分离总线时）

总线控制信号	I/O	功能	复用功能	在端口模式/复用功能模式之间切换所需的寄存器
AD0 至 AD15	I/O	数据总线	PDL0 至 PDL15	PMCDL 寄存器
A0 至 A15	输出	地址总线	P90 至 P915	PMC9 寄存器
A16 至 A21	输出	地址总线	PDH0 至 PDH4, P02	PMCDH 寄存器, PMC0 寄存器
$\overline{\text{WAIT}}$	输入	外部等待控制	PCM0	PMCCM 寄存器
CLKOUT	输出	内部系统时钟输出	PCM1	PMCCM 寄存器
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	输出	写选通信号	PCT0, PCT1	PMCCT 寄存器
$\overline{\text{RD}}$	输出	读选通信号	PCT4	PMCCT 寄存器
$\overline{\text{HLDRQ}}$	输入	总线保持控制	PCM3	PMCCM 寄存器
$\overline{\text{HLDAK}}$	输出		PCM2	

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

当访问片内 ROM 存储器、片内 RAM 存储器或片上外设 I/O 时，引脚状态如下：

表 5-3. 当访问片内 ROM 存储器、片内 RAM 存储器或片上外设 I/O 时的引脚状态

总线控制引脚	复用总线模式 /分离总线模式		
	内部 ROM/RAM	外设 I/O	存储器块 1
地址/数据总线 (AD15 至 AD0)	不确定	不确定	不确定
地址总线 (A21 至 A16)	低电平	不确定	不确定
地址总线 (A15 至 A0)	不确定	不确定	不确定
控制信号	无效	无效	无效

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

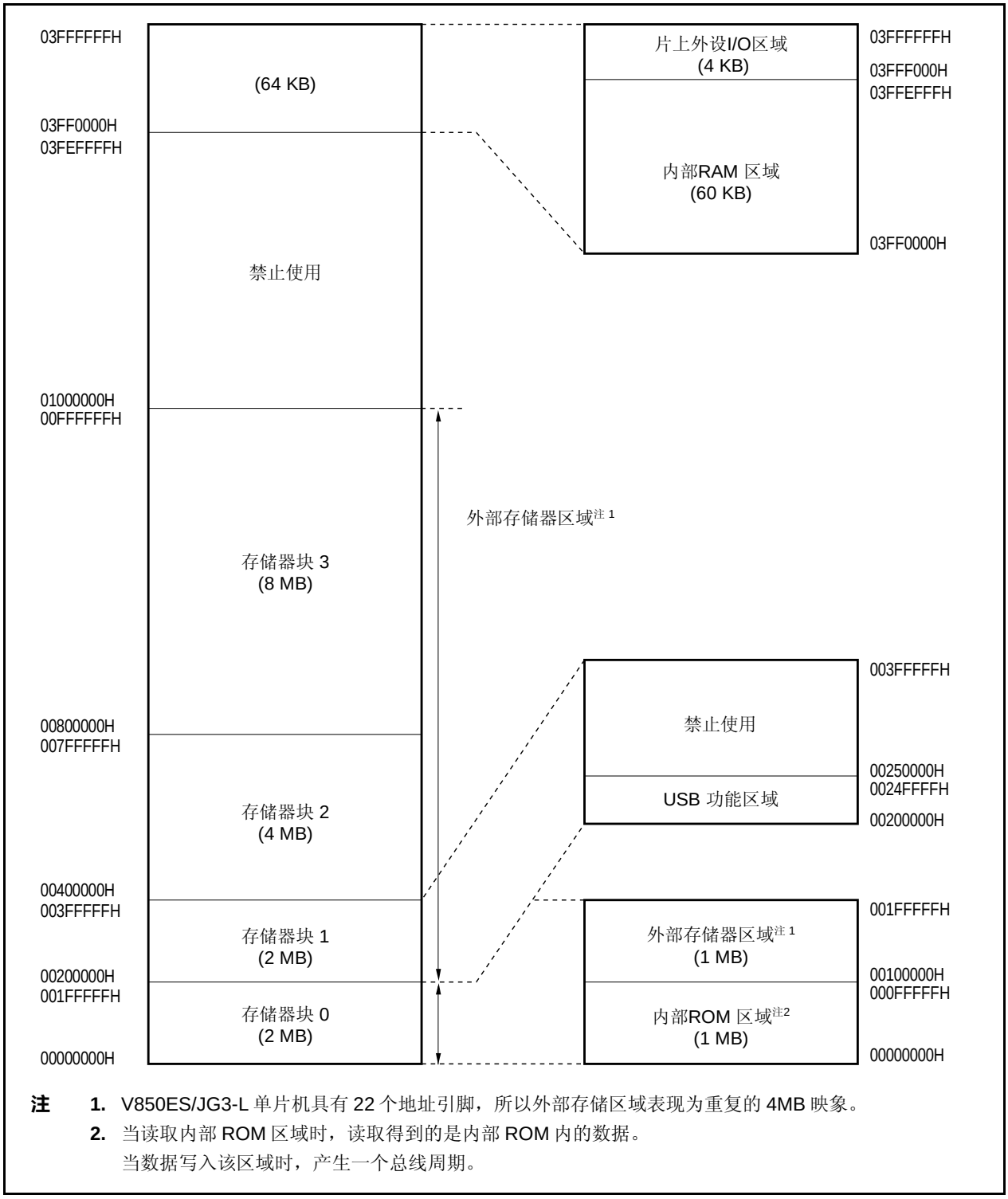
5.2.2 各种工作模式下的引脚状态

关于 V850ES/JG3-L 单片机引脚在各种工作模式下的状态，参见 **2.2 引脚状态**。

5.3 存储模块功能

保留 64 MB 存储器空间的低 16 MB 作为外部存储器扩展，低 16 MB 被划分成 2 MB、2 MB、4 MB 和 8 MB 的存储器块。每个块的总线宽度和可编程等待功能都可以进行独立指定。

图 5-1. 数据存储映射： 物理存储空间



注 **1.** V850ES/JG3-L 单片机具有 22 个地址引脚，所以外部存储区域表现为重复的 4MB 映象。

2. 当读取内部 ROM 区域时，读取得到的是内部 ROM 内的数据。
当数据写入该区域时，产生一个总线周期。

5.4 总线访问

5.4.1 访问所需的时钟周期数目

访问各个存储资源所需的基本时钟周期数目如下表所示：

表 5-4. 访问所需的时钟周期数目

区域（总线宽度） 总线周期类型	内部 ROM （32 位）	内部 RAM （32 位）	外部存储器 （16 位）	片上外设 I/O （16 位）
取指（常规访问）	1	1 注 1	3 + n + i 注 2	–
取指（分支跳转）	2	2 注 1	3+ n + i 注 2	–
操作数数据访问	3	1	3 +n + i 注 2	3 注 3
DMA 传输	–	2	3 +n + i 注 2	3 注 3

注 **1.** 如果与数据访问发生冲突，则时钟数目加 1。
2. .i = 空闲状态
3. 该值会有所不同，由 VSWC 寄存器的设置决定。

备注 单位： 时钟周期/访问

5.4.2 总线宽度设置功能

V850ES/JG3-L 单片机的外部存储器区域由存储器块 0 至 3 选择。

由存储模块 n 选择的每一个外部存储器区域的总线宽度都可以通过 BSC 寄存器来设置。如果指定为 16 位总线宽度，低 8 位用作偶地址，高 8 位用作奇地址。

（1）总线宽度设置寄存器（BSC）

该寄存器控制存储器块空间的总线宽度。

该寄存器可以按 16 位宽度进行读写。

系统复位后，该寄存器的值被设置为 5555H。

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

复位后：5555H R/W 地址： FFFFF066H

	15	14	13	12	11	10	9	8
BSC	0	1	0	1	0	1	0	1
	7	6	5	4	3	2	1	0
	0	BS30	0	BS20	0	1	0	BS00
存储器块 n 信号		存储器块 3		存储器块 2				存储器块 0

BSn0	存储器块 n 空间的数据总线宽度 (n = 0, 2, 3)
0	8 位
1	16 位注

注 如果指定为 16 位总线宽度，通过两个控制引脚（$\overline{WR0}$ 和 $\overline{WR1}$）可以控制按照 8 位单元写入，但是读取只能按照 16 位单元进行，因为读取操作是通过一个控制引脚（$\overline{RD}$）进行控制。但是，在 V850ES/JG3-L 中，无用的数据被忽略，所以可以进行字节访问。

注意事项 请确保将第 14、12、10、8 位和第 2 位设置为“1”，并将第 15、13、11、9、7、5、3 位和第 1 位清除为“0”。

5.4.3 根据不同总线宽度进行访问

V850ES/JG3-L 单片机按照 8 位、16 位或 32 位总线宽度访问片上外设输入/输出端口和外部存储器。总线宽度说明如下：

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

以上各种访问时的操作描述如下。所有数据访问都从低位开始。
V850ES/JG3-L 单片机仅支持“小端”数据格式。

图 5-2. 字数据中的“小端”地址

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

（1）数据空间

V850ES/JG3-L 单片机具有地址不对齐功能。
基于该功能，数据可以放置在任何地址，无论该数据是何种格式（字或半字）。然而，如果字数据或半字数据没有在边界对齐，则产生冗余总线周期，导致总线效率降低。

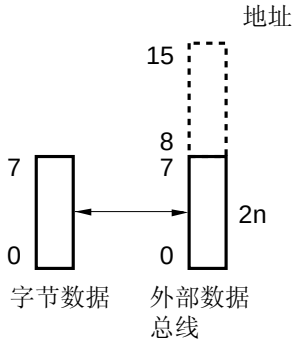
下面是 8 位、16 位和 32 位访问的示例。

（2）字节访问（8 位）

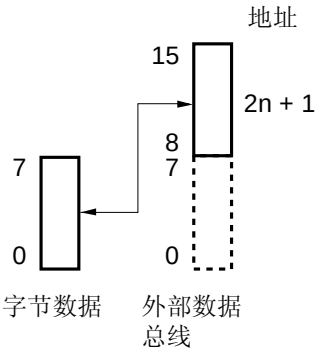
（a） 16 位数据总线宽度

通过 16 位总线发送/接收 8 位数据。于是，如果指定为偶地址，访问的是外部数据总线地址的低位字节。如果指定为奇地址，访问的是外部数据总线地址的高位字节。

<1> 访问偶地址（2n）



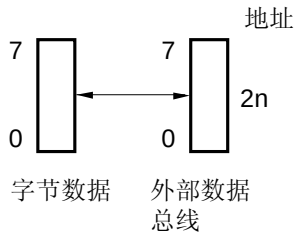
<2> 访问奇地址（2n + 1）



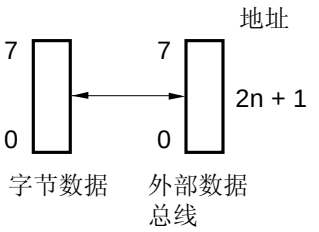
（b） 8 位数据总线宽度

通过 16 位总线发送/接收 8 位数据。于是，可以访问指定外部数据总线的偶地址/奇地址。

<1> 访问偶地址（2n）



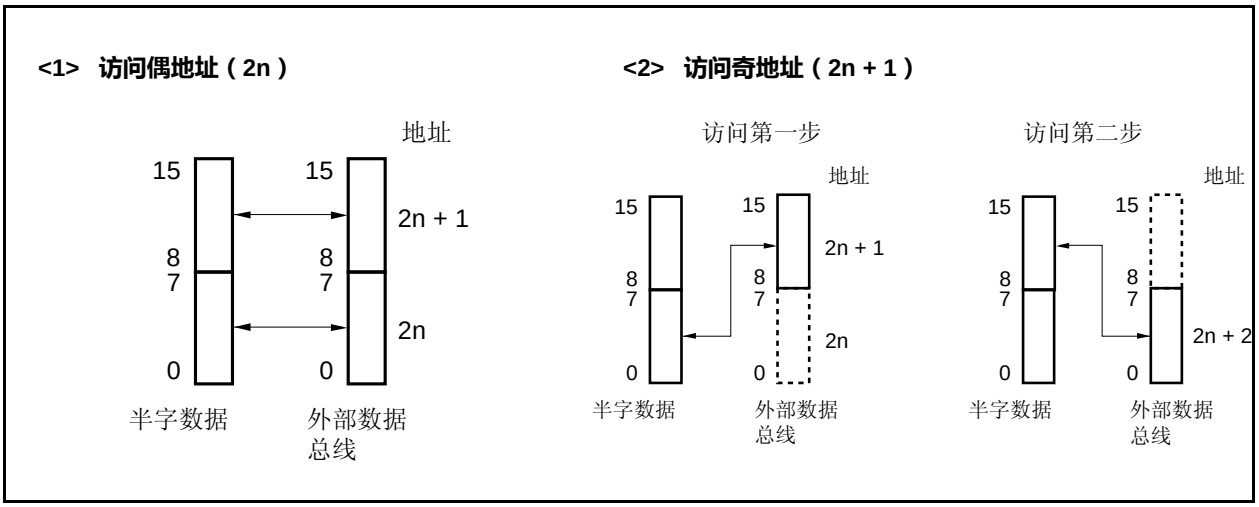
<2> 访问奇地址（2n + 1）



（3）半字访问（16 位）

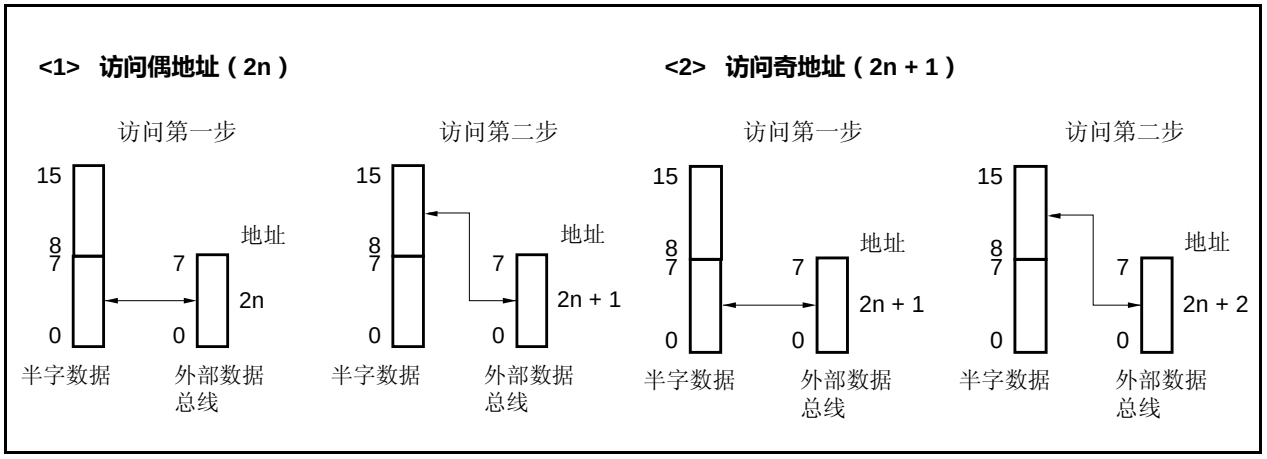
（a） 16 位数据总线宽度

通过 16 位总线发送/接收 16 位数据。于是，如果指定为偶地址，可以同时访问外部数据总线地址的低位和高位字节。如果指定为奇地址，访问的是外部数据总线地址的高位字节。如果指定为奇地址，在第一次访问中通过外部数据总线地址的高位字节来发送/接收数据的低位字节内容，在第二次访问中，通过外部数据总线地址的低位字节来发送/接收数据的高位字节内容。



（b） 8 位数据总线宽度

通过 8 位总线发送/接收 16 位数据。于是，数据需要两次访问来发送/接收。数据的低位/高位字节的发送/接收要通过外部总线地址的低位/高位字节对应进行。

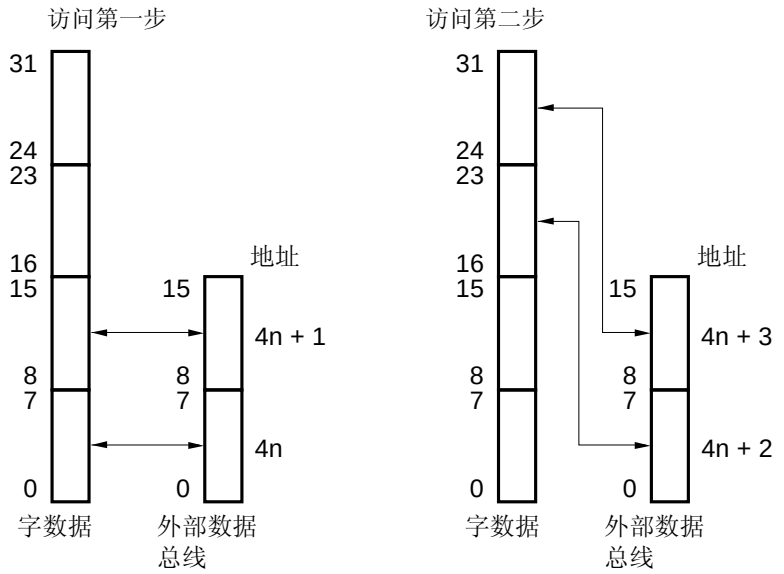


（4）字访问（32 位）

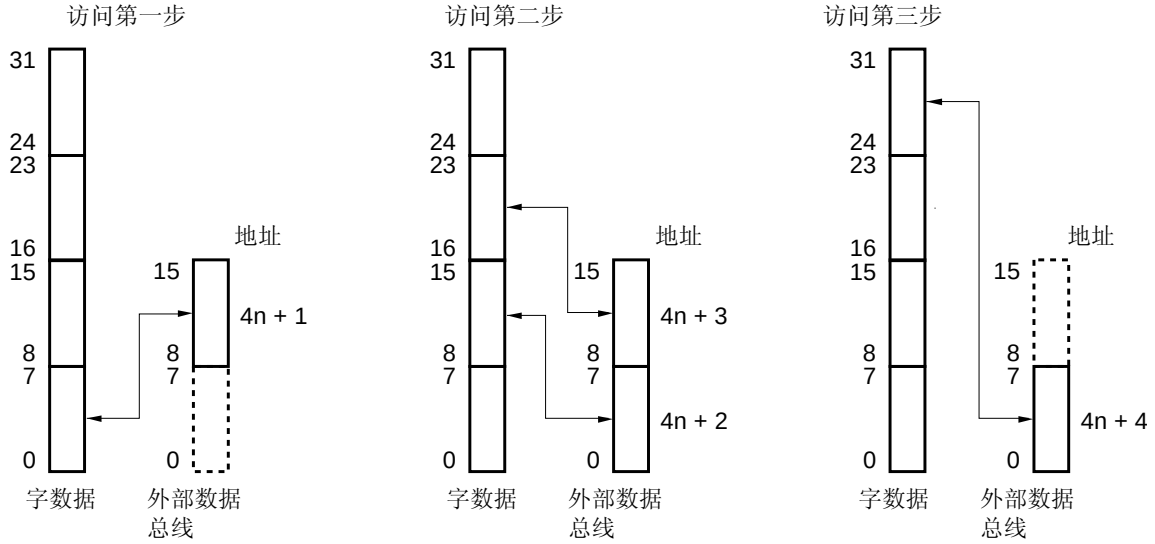
（a） 16 位数据总线宽度（1/2）

通过 16 位总线发送/接收 32 位数据。于是，如果指定为偶地址，可以按照 16 位单元进行两次访问来发送/接收数据。如果指定为奇地址，最低的四分之一字数据通过外部数据总线地址的高位字节来发送/接收（第一次访问），中间的半字数据通过外部数据总线地址的中间字节来发送/接收（第二次访问），最高的四分之一字数据通过外部数据总线地址的低位字节来发送/接收（第三次访问）。

<1> 访问地址（4n）

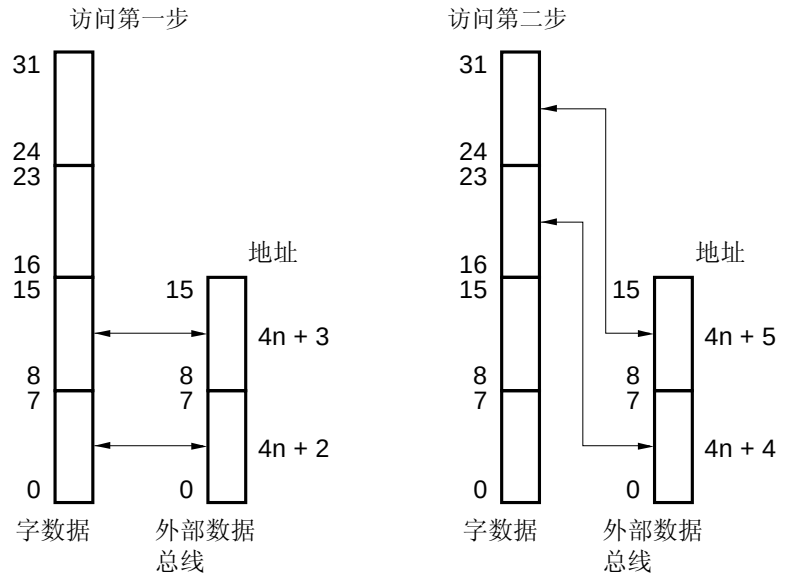


<2> 访问地址（4n + 1）

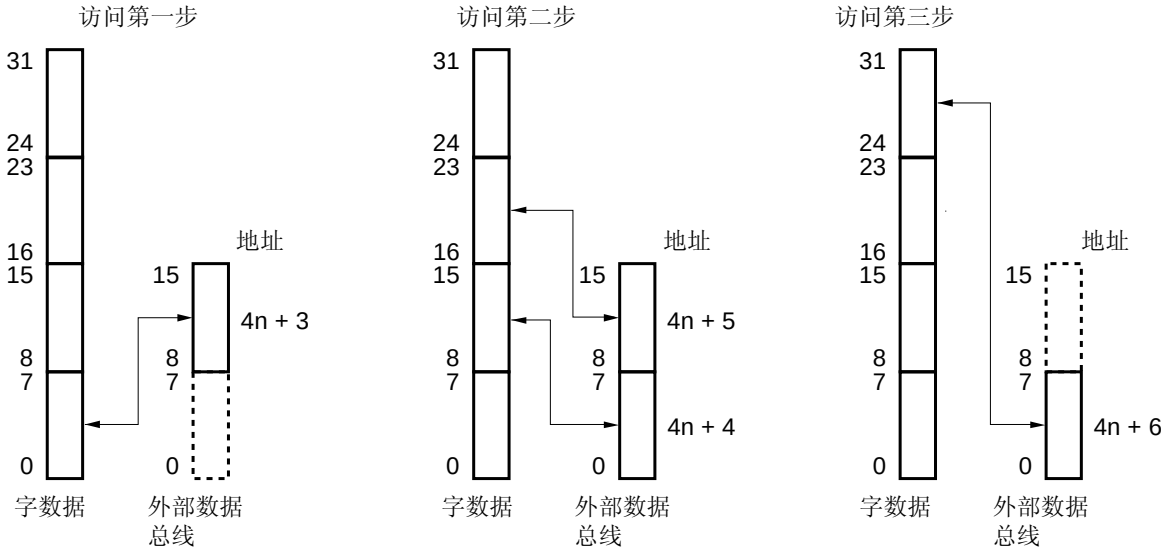


（a） 16 位数据总线宽度（2/2）

<3> 访问地址（4n + 2）



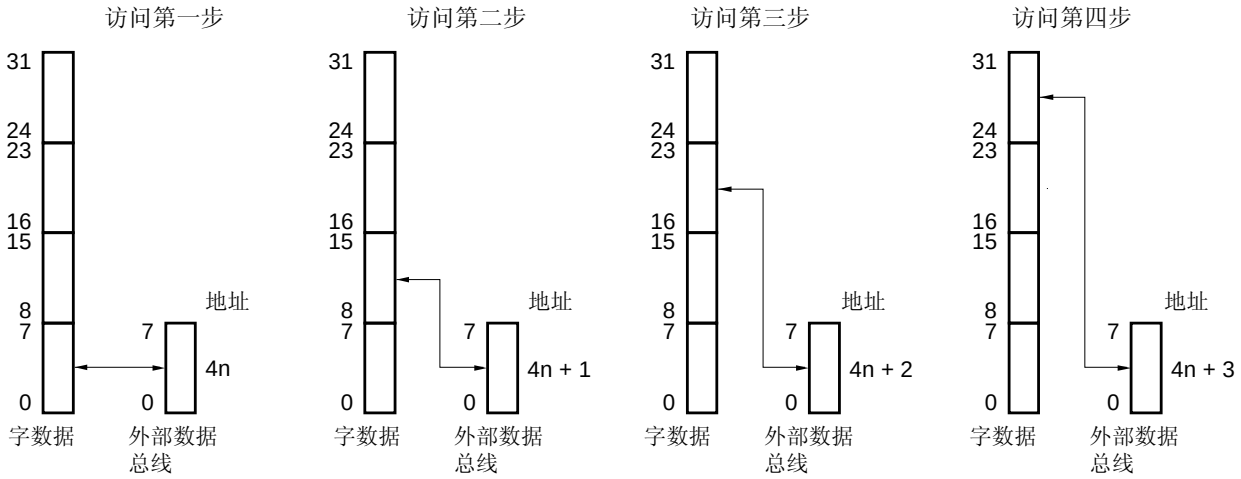
<4> 访问地址（4n + 3）



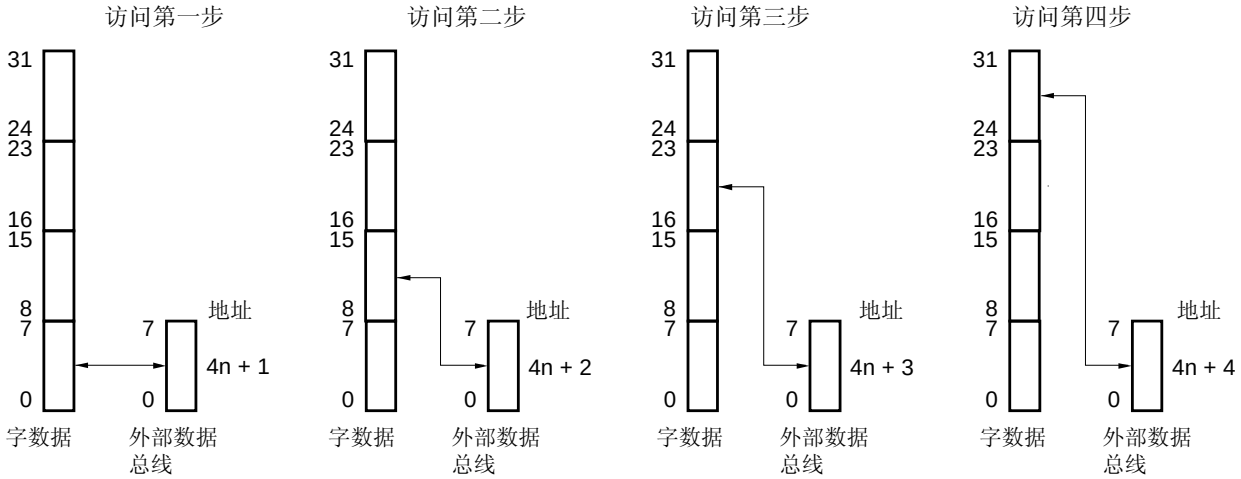
（b） 8 位数据总线宽度（1/2）

通过 8 位总线发送/接收 32 位数据。于是，数据的发送/接收需要四次访问完成。数据通过指定的外部数据总线的偶/奇地址来进行发送/接收。

<1> 访问地址（4n）

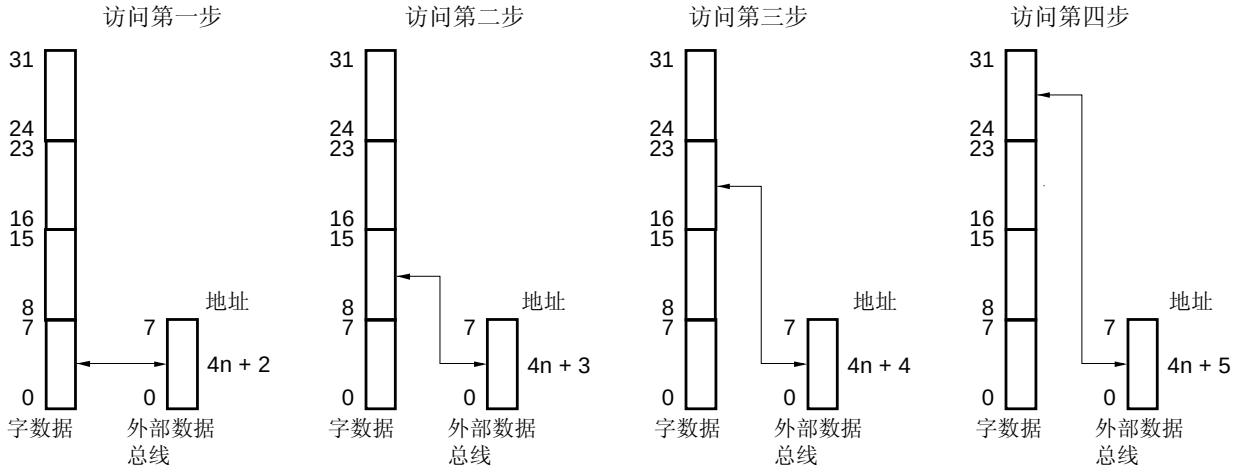


<2> 访问地址（4n + 1）

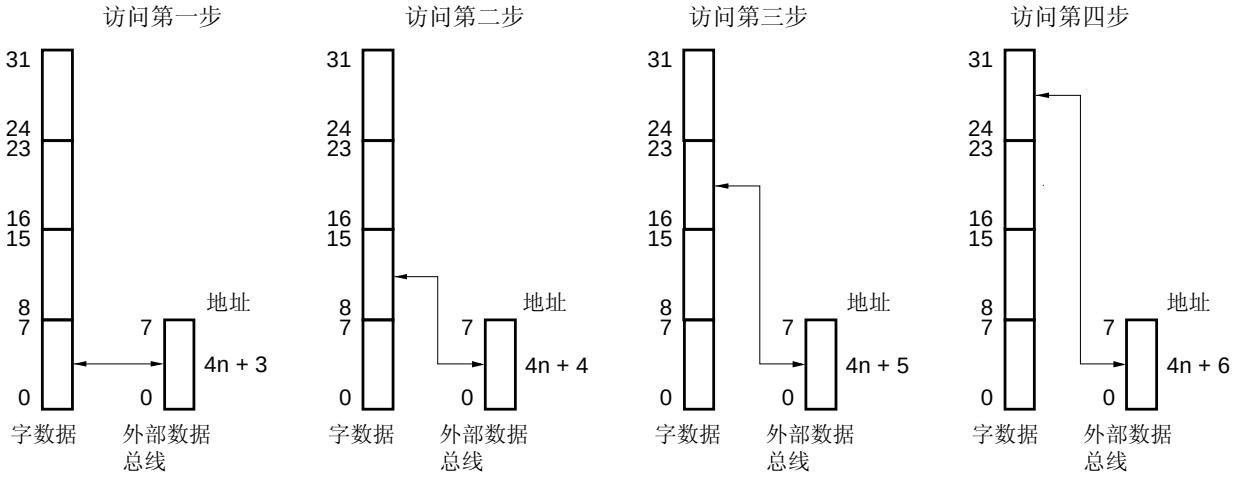


（b） 8 位数据总线宽度（2/2）

<3> 访问地址（4n + 2）



<4> 访问地址（4n + 3）



5.5 等待功能

5.5.1 可编程等待功能

（1）数据等待控制寄存器 0（DWC0）

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

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

DWC0 寄存器可以按 16 位宽度进行读写。

系统复位后，DWC0 寄存器被置为 7777H。

注意事项 1. 对内部 ROM 和内部 RAM 的访问不需要可编程等待，总是可以无等待访问。片上外设输入/输出端口也不需要可编程等待，而是根据各自外设输入/输出端口的控制功能执行等待。

2. 系统复位完成后写入 DWC0 寄存器，此后不要再改变 DWC0 寄存器的值。而且，当改变 DWC0 寄存器的初始值时（设置完成之前），不要访问外部存储器区域。

复位后： 7777H R/W 地址：FFFFF484H

	15	14	13	12	11	10	9	8
DWC0	0	DW32	DW31	DW30	0	DW22	DW21	DW20
存储器块 n 信号		存储器块 3				存储器块 2		

7	6	5	4	3	2	1	0
0	DW12注	DW11注	DW10注	0	DW02	DW01	DW00
存储器块 n 信号					存储器块 0		

DWn2	DWn1	DWn0	在存储器块 n 空间中插入的等待状态数量(n = 0 至 3)
0	0	0	无
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

注 DW12 至 DW10 位用于指定是否为 USB 功能控制器插入访问等待状态。使用 USB 功能控制器时，将这些位设置为 001（指定为插入 1 个等待状态）。

注意事项 请确保将第 15、11、7 位至第 3 位清除为“0”。

5.5.2 外部等待功能

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

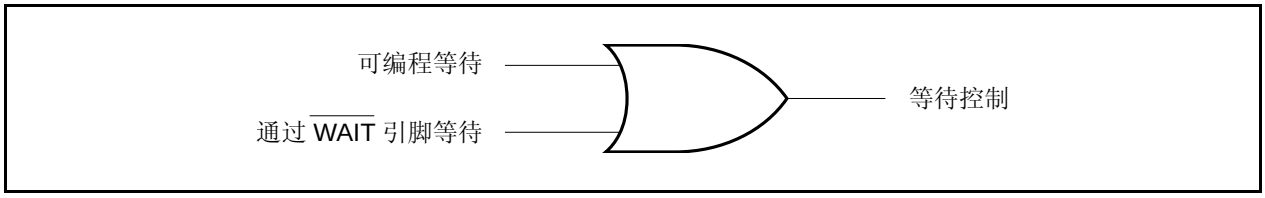
对内部 ROM、内部 RAM 的任何区域以及片上外设输入/输出端口的访问，均不需要外部等待功能，这一点和可编程等待功能一样。

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

将 PMCCM.PMCCM0 位设置为 1 可以使能 $\overline{\text{WAIT}}$ 输入功能（参见 **4.3.8 端口 CM**）。

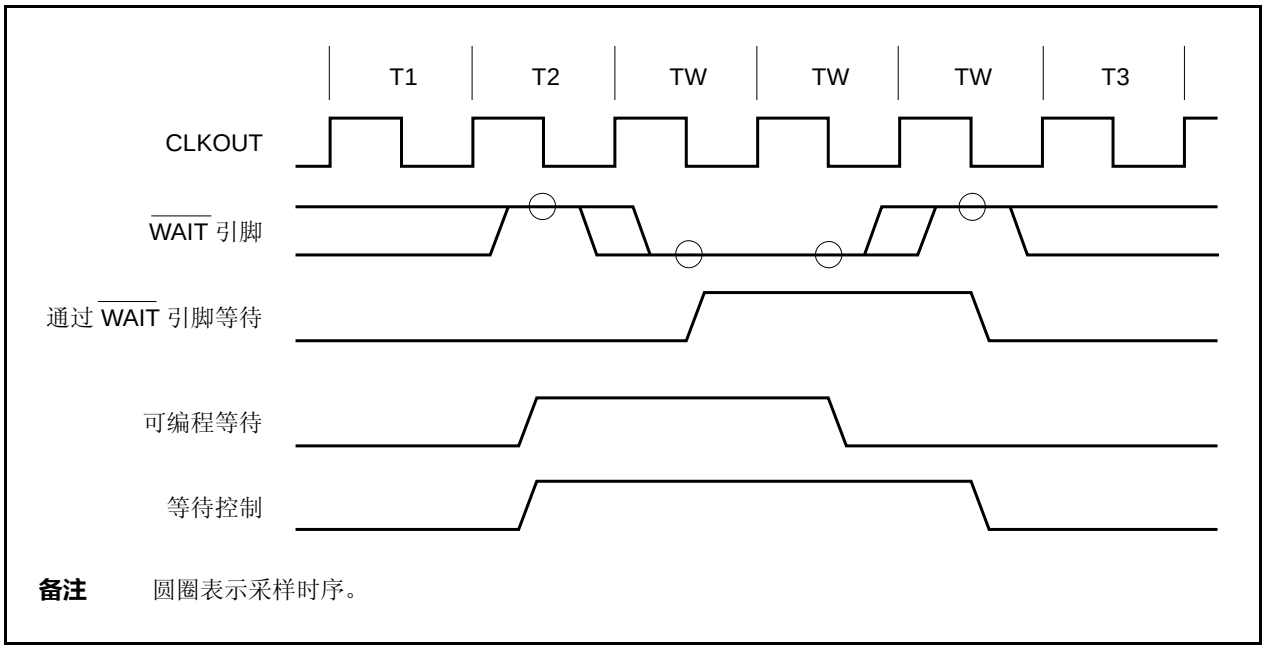
5.5.3 可编程等待和外部等待之间的关系

对可编程等待值设定的等待周期与 $\overline{\text{WAIT}}$ 引脚控制的等待周期执行逻辑或（OR）操作，其结果即是插入的等待周期数目。



例如，假设可编程等待的时序和 $\overline{\text{WAIT}}$ 引脚信号的时序如下图 5-3 所示，则将会在总线周期内插入 3 个等待状态。如果等待插入由 $\overline{\text{WAIT}}$ 引脚控制，等待状态可能不会再期望的时序处插入。这种情况下，通过指定可编程等待值来调整插入时序。

图 5-3. 插入等待的示例



备注 圆圈表示采样时序。

5.5.4 可编程地址等待功能

通过 AWC 寄存器，可以设置插入到各个总线周期中的地址建立等待或地址保持等待。对每个存储模块区（存储模块 0 至 3）都需要设置地址等待插入。

当在总线周期中插入一个地址建立等待时，表现为 T1 状态的高电平时钟被延长了一个时钟周期。当插入一个地址保持等待时，表现为 T1 状态的低电平时钟被延长了一个时钟周期。

（1）地址等待控制寄存器（AWC）

该寄存器可以按 16 位宽度进行读写。

系统复位后，该寄存器被设置为 FFFFH。

注意事项 1. 访问内部 ROM 区、内部 RAM 区和片上外设输入/输出区域时，不需要插入地址建立等待或地址保持等待。

2. 系统复位完成后写入 AWC 寄存器，此后不要再改变该设定值。而且，当改变 AWC 寄存器的初始值时（设置完成之前），不要访问外部存储空间。

复位后： FFFFH R/W 地址：FFFFF488H

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

AHWn	指定地址保持等待的插入 (n = 0 至 3)
0	不插入
1	已插入

ASWn	指定地址建立等待的插入 (n = 0 至 3)
0	不插入
1	已插入

注 AHW1 和 ASW1 位用于指定是否为 USB 功能控制器插入等待状态。使用 USB 功能控制器时，将这些位设置为 00B。

注意事项 请确保将第 15 至 8 位设置为“1”。

5.6 空闲状态插入功能

为了实现与低速设备的连接，对存储模块选择的各个空间执行访问时，在总线周期的 T3 状态后插入一个空闲状态（TI）。在读取访问时插入一个空闲状态，可以保证存储器数据输出的浮动延迟时间（写访问期间，不能插入空闲状态）。

通过 BCC 寄存器，可以设置是否插入空闲状态。

系统复位之后，对所有区域的访问时序都会立刻插入一个空闲状态。

（1）总线周期控制寄存器（BCC）

该寄存器可以按 16 位宽度进行读写。

系统复位后，该寄存器 被置为 AAAAH。

注意事项 1. 内部 ROM 、内部 RAM、片上外设 I/O 端口等区域的访问不支持空闲状态插入。

2. 系统复位完成后写入 BCC 寄存器，此后不要再改变 BCC 寄存器的值。而且，当改变 BCC 寄存器的初始值时（设置完成之前），不要访问外部存储空间。

复位后： AAAAH　R/W　地址： FFFFF48AH

	15	14	13	12	11	10	9	8
BCC	1	0	1	0	1	0	1	0
	7	6	5	4	3	2	1	0
	BC31	0	BC21	0	BC11注	0	BC01	0
存储器块 n 信号	存储器块 3		存储器块 2				存储器块 0	

BCn1	指定空闲状态的插入 (n = 0 至 3)
0	不插入
1	已插入

注 BC11 位用于指定是否为 USB 功能控制器插入等待状态。使用 USB 功能控制器时，将该位设置为 0。

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

5.7 总线保持功能

5.7.1 功能概述

如果 PCM2 和 PCM3 引脚被设为控制模式，则 $\overline{\text{HLDRQ}}$ 和 $\overline{\text{HLDAK}}$ 信号有效。

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

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

总线保持功能允许多处理器系统的配置，其中存在两个或更多个总线主控器。

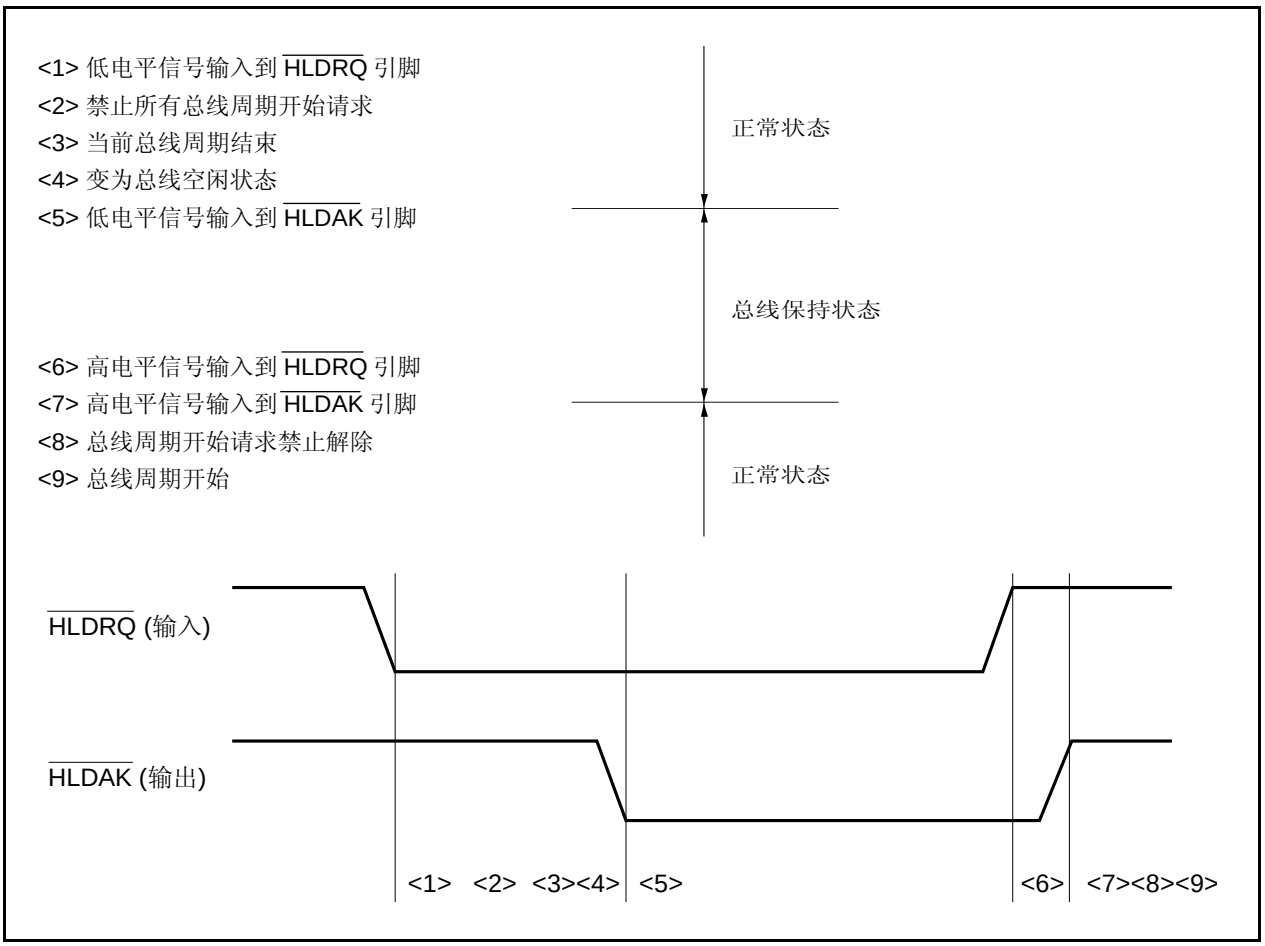
请注意，在由总线宽度功能或位操作指令发起的多重访问循环中，总线保持请求将得不到响应。不响应总线保持请求的时序如下所示。

表 5-6. 不响应总线保持请求的时序

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

5.7.2 总线保持步骤

总线保持状态转变步骤如下所示：



5.7.3 节电模式时的操作

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

HALT 模式下，HLDAK 引脚和 HLDRQ 引脚都变为低电平（即有效状态），会进入总线保持状态。当 HLDRQ 引脚随后变为无效状态后，HLDAK 引脚也变为无效状态，则退出总线保持状态。

5.8 总线优先权

总线保持、取指（分支跳转）、取指（连续）、运算对象数据访问、DMA 传输等操作在外部总线周期中执行。

总线保持具有最高的优先级，接着是 DMA 传输、运算对象数据访问、取指（分支跳转）、取指（连续）。

但是，在读-修改-写访问中，读访问和写访问之间可以插入一个取指令操作。

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

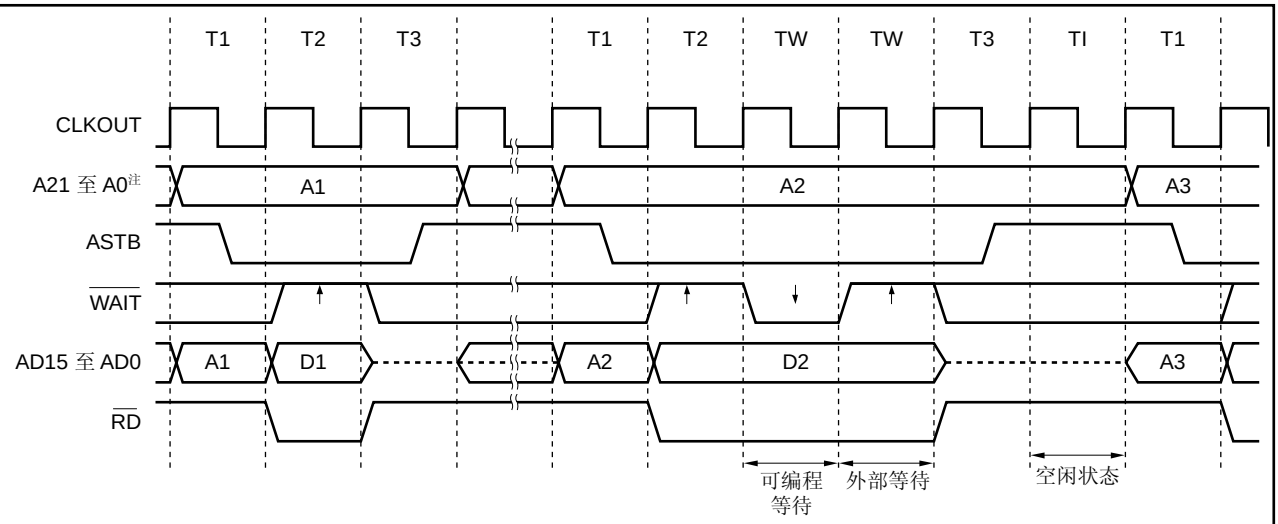
表 5-7. 总线优先级

优先级	外部总线周期	总线主控器
高	总线保持	外部设备
↑	DMA 传输	DMAC
↕	操作数访问	CPU
↓	取指（分支跳转）	CPU
低	取指（连续）	CPU

5.9 总线时序

典型的总线时序图如下所示。

图 5-4. 复用总线/分离总线读时序（总线宽度：16 位，16 位存取）



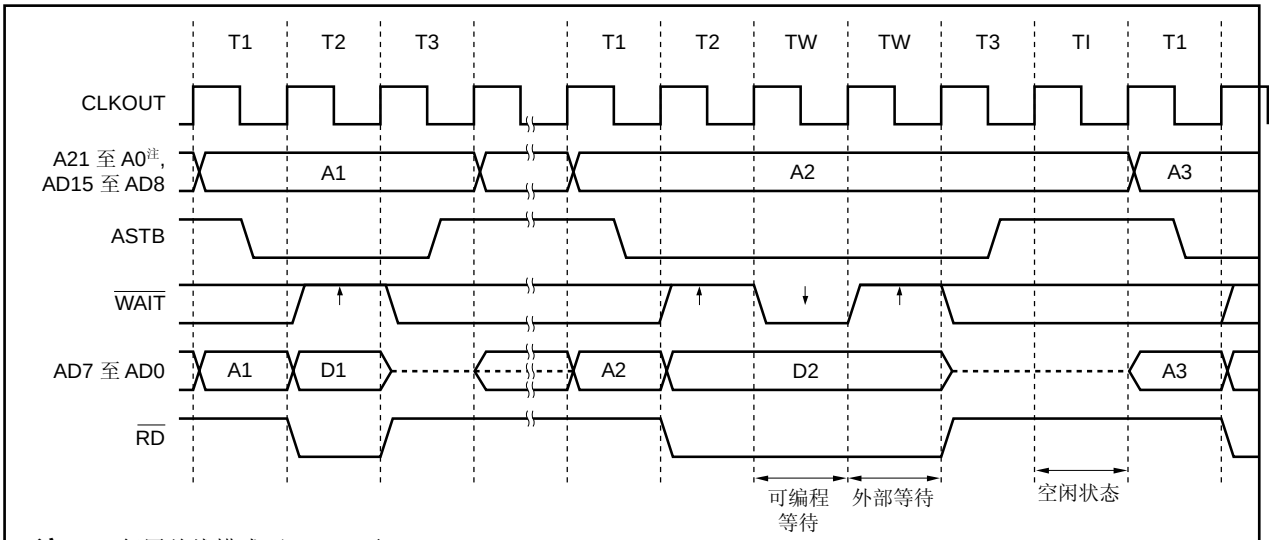
注 复用总线模式下，A21 至 A16。

备注 **1.** 当指定为 16 位访问时，8 位访问的数据有效性如下所示。

8 位访问	奇地址	偶地址
AD15 至 AD8	有效数据	无效数据
AD7 至 AD0	无效数据	有效数据

2. 虚线表示高阻状态。

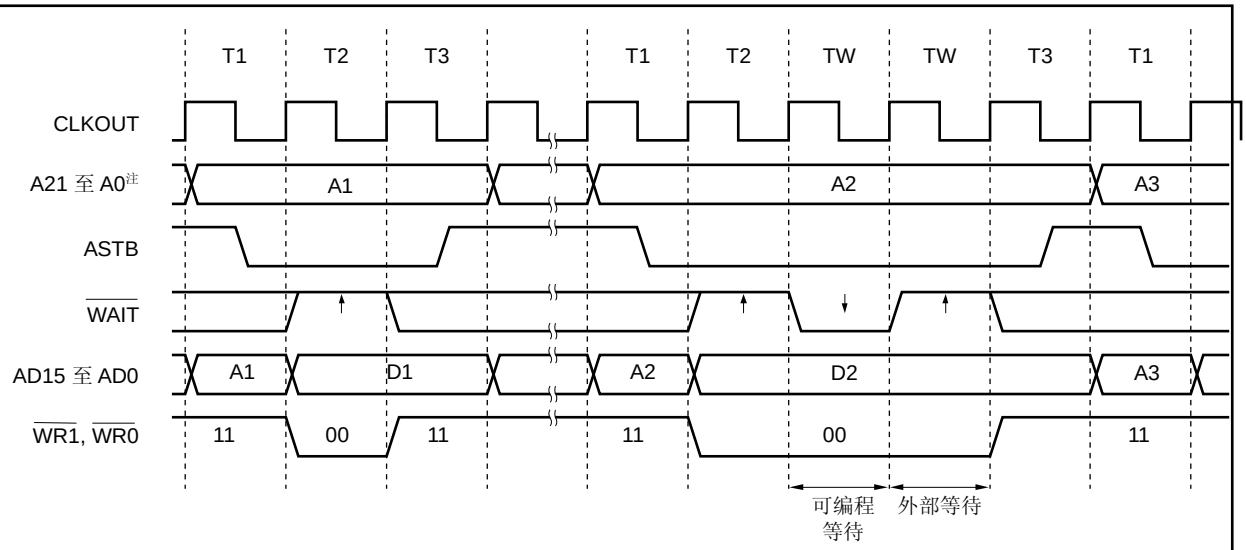
图 5-5. 复用总线/分离总线读取时序（总线宽度：8 位）



注 复用总线模式下，A21 至 A16。

备注 虚线表示高阻状态。

图 5-6. 复用总线/分离总线写入时序（总线宽度：16 位，16 位访问）

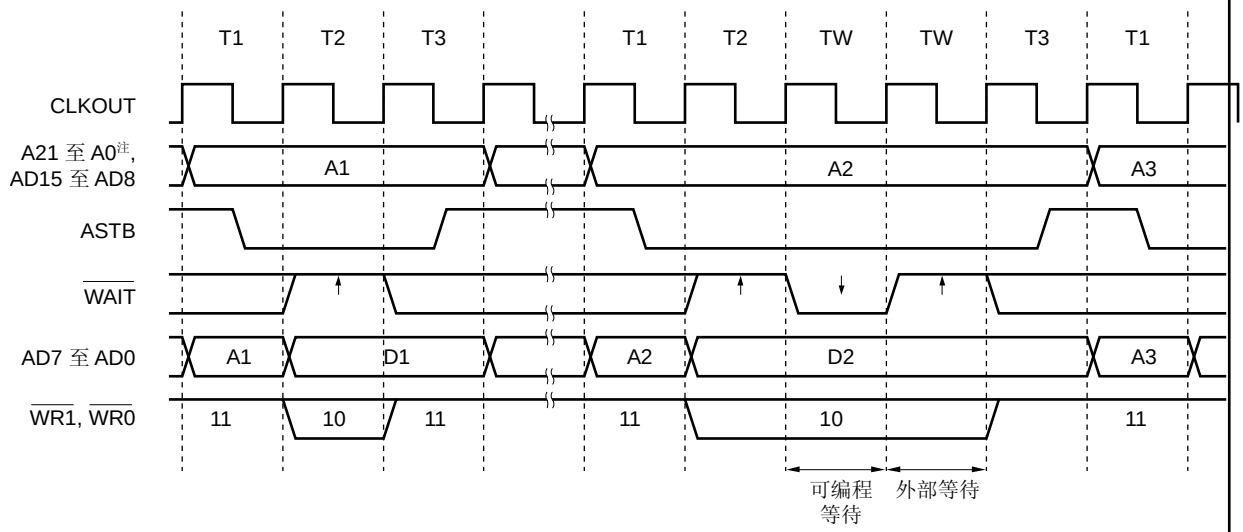


注 复用总线模式下，A21 至 A16。

备注 当指定为 16 位访问时，8 位访问的数据有效性如下所示。

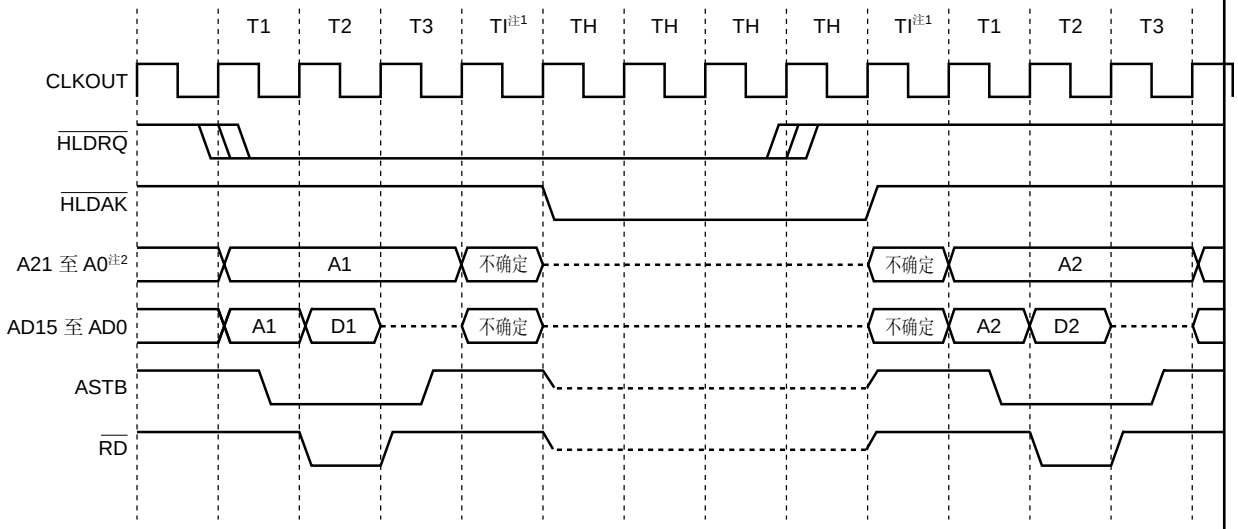
8 位访问	奇地址	偶地址
AD15 至 AD8	有效数据	无效数据
AD7 至 AD0	无效数据	有效数据
$\overline{WR1}$, $\overline{WR0}$	01	10

图 5-7. 复用总线/分离总线写入时序（总线宽度：8 位）



注 复用总线模式下，A21 至 A16。

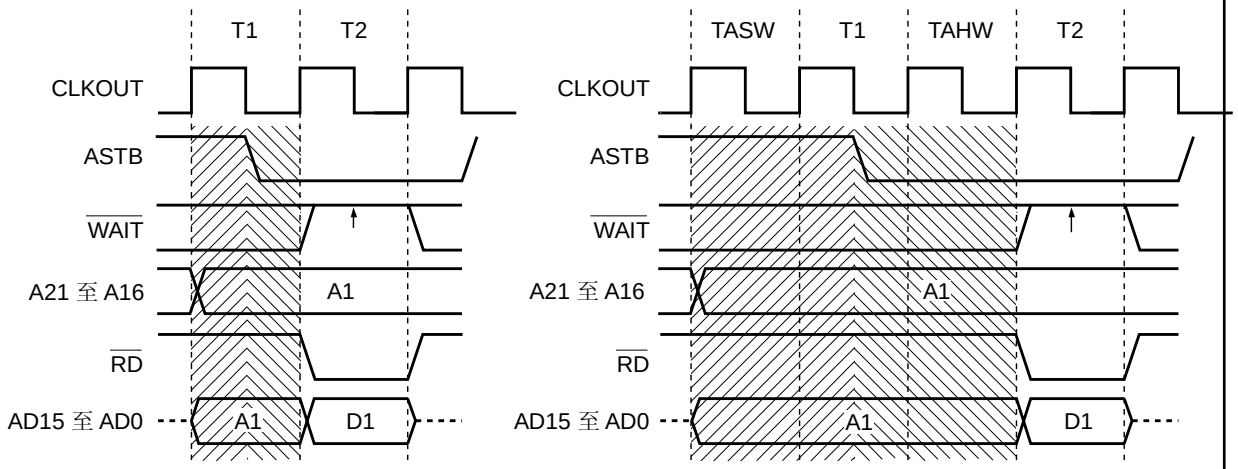
图 5-8. 复用总线/分离总线保持时序（总线宽度：16 位，16 位存取）



注 **1.** 该空闲状态（TI）与 BCC 寄存器的设置无关。
2. 复用总线模式下，A21 至 A16。

备注 **1.** 总线保持模式下的引脚状态参见**表 2-2**。
2. 虚线表示高阻状态。

图 5-9. 地址等待时序（总线宽度：16 位，16 位访问）



备注 **1.** TASW（地址建立等待）：图示 T1 状态的高电平宽度被延长。
2. TAHW（地址保持等待）：图示 T1 状态的低电平宽度被延长。
3. 虚线表示高阻状态。

第六章 时钟发生器

6.1 概述

时钟发生器用于产生输入到 CPU 和各个外设的时钟信号。时钟发生器内置一个 PLL 电路，可以对时钟频率进行 4 倍频或 8 倍频。在时钟信号输入到 CPU 或片上外设之前，也可以对时钟频率进行分频。也可以停止时钟振荡以降低功耗。

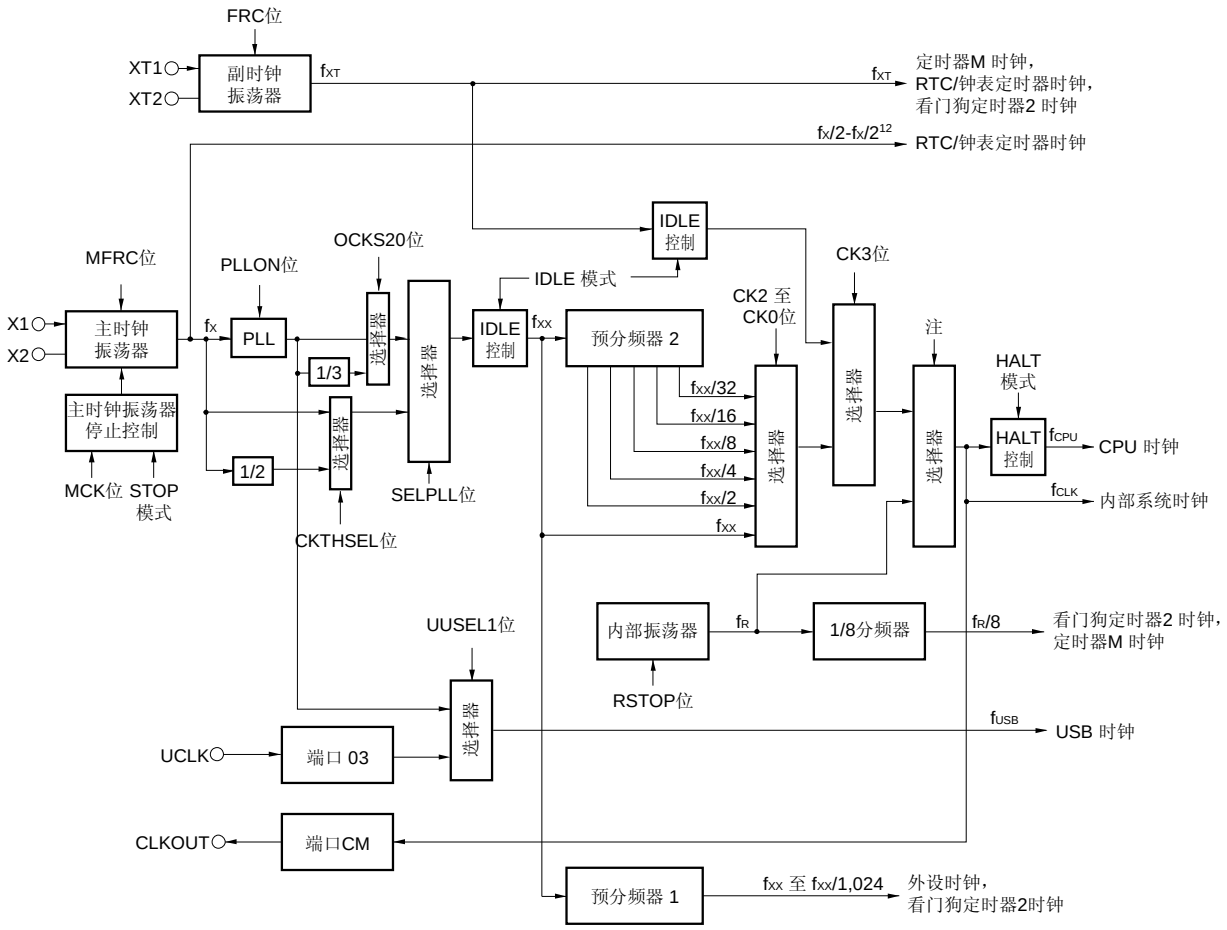
时钟发生器具备以下功能：

- 主时钟振荡器
  - 未使用 USB 时
    - 时钟直通模式： f_X = 2.5 至 10 MHz (f_{XX} = 1.25 至 10 MHz)
    - PLL 模式： f_X = 2.5 至 5 MHz (×4 : f_{XX} = 10 至 20 MHz)
  - 使用 USB（未使用 UCLK）
    - 时钟直通模式： 禁止设置
    - PLL 模式： f_X = 6 MHz (×8, 1/3：f_{XX} = 16 MHz，f_{USB} = 48 MHz)
  - 使用 USB（使用 UCLK）
    - 时钟直通模式： f_X = 2.5 至 10 MHz (f_{XX} = 1.25 至 10 MHz)
    - PLL 模式： f_X = 2.5 至 4 MHz (×4 : f_{XX} = 10 至 16 MHz)
    - f_X = 6 MHz (×8, 1/3：f_{XX} = 16 MHz)
- 副时钟振荡器
  - f_{XT} = 32.768 kHz
- 内部振荡器
  - f_R = 220 kHz（典型值）
- PLL（锁相环）倍频（×4）功能
  - 时钟直通模式/PLL 模式可选（f_X = 2.5 至 5 MHz）
- 内部系统时钟的产生
  - 7 个级别（f_{XX}，$f_{XX}/2$，$f_{XX}/4$，$f_{XX}/8$，$f_{XX}/16$，$f_{XX}/32$，f_{XT}）
- 外设时钟产生功能
- 时钟输出

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

6.2 配置

图 6-1. 时钟发生器



注 振荡稳定等待期间，看门狗定时器 2 溢出时，选择内部振荡时钟。

备注
- f_X： 主时钟振荡频率
- f_{XX}： 主时钟频率
- f_{CLK}： 内部系统时钟频率
- f_{XT}： 副时钟频率
- f_{CPU}： CPU 时钟频率
- f_R： 内部振荡时钟频率
- f_{USB}： USB 时钟频率

（1）主时钟振荡器

主时钟振荡器使用连接至 X1 和 X2 引脚的陶瓷/晶体谐振器，来产生以下频率（f_X）。

(a) 未使用 USB 时

- 时钟直通模式： f_X = 2.5 至 10 MHz (f_{XX} = 1.25 至 10 MHz)
- PLL 模式： f_X = 2.5 至 5 MHz (×4 : f_{XX} = 10 至 20 MHz)

下列频率的外部时钟也可以输入到 X1 引脚。

- 时钟直通模式： f_X = 2.5 至 6 MHz (f_{XX} = 1.25 至 6 MHz)
- PLL 模式： f_X = 2.5 至 5 MHz (×4 : f_{XX} = 10 至 20 MHz)

(b) 使用 USB（未使用 UCLK）

- 时钟直通模式： 禁止设置
- PLL 模式： f_X = 6 MHz (×8, 1/3 : f_{XX} = 16 MHz, f_{USB} = 48 MHz)

下列频率的外部时钟也可以输入到 X1 引脚。

- PLL 模式： f_X = 6 MHz (×8, 1/3 : f_{XX} = 16 MHz, f_{USB} = 48 MHz)

(c) 使用 USB（使用 UCLK）

- 时钟直通模式： f_X = 2.5 至 10 MHz (f_{XX} = 1.25 至 10 MHz)
- PLL 模式： f_X = 2.5 至 4 MHz (×4 : f_{XX} = 10 至 16 MHz)
  f_X = 6 MHz (×8, 1/3 : f_{XX} = 16 MHz)

下列频率的外部时钟也可以输入到 X1 引脚。

- 时钟直通模式： f_X = 2.5 至 6 MHz (f_{XX} = 1.25 至 6 MHz)
- PLL 模式： f_X = 2.5 至 4 MHz (×4 : f_{XX} = 10 至 16 MHz)
  f_X = 6 MHz (×8, 1/3 : f_{XX} = 16 MHz)

注 为了满足 USB 通信数据的 USB 速率，请使用精度为 6 MHz ±2500 ppm (max.)的时钟。

（2）副时钟振荡器

副时钟振荡器振荡于 32.768 kHz（f_{XT}）频率。

即使在 RTC 备份模式下，副时钟处于 RTC 备份区域，导致副时钟会继续振荡。

（3）主时钟振荡器停止控制

该电路产生一个控制信号，可以使主时钟振荡器停止工作。

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

（4）内部振荡器

以 220 kHz（典型值）的频率(f_R)振荡。

（5）预分频器 1

该预分频器产生时钟（f_{XX} 至 f_{XX} /1,024），用来供给下列片上外设功能：TMP0 至 TMP5，TMQ0，TMM0，CSIB0 至 CSIB4，UARTA0 至 UARTA5，UARTC0，I^2C00 至 I^2C02，ADC 以及 WDT2。

（6）预分频器 2

该电路对主时钟（f_{XX}）进行分频。

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

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

（7）PLL

本电路对主时钟振荡器产生的时钟（f_X）进行 4 倍频。

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

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

（8）时钟 I/O 电路

该电路向 CLKOUT 引脚输出内部系统时钟。

端口 CM 的 PMCCM 寄存器的 PMCCM1 位用于控制 PMC1 引脚是作为 I/O 端口或是作为 CLKOUT 输出。

（9）OPS2 控制电路

该电路对经过PLL电路倍频并输出的时钟进行奇数次分频。

当使用 USB 功能控制器时，对 6 MHz 的主时钟振荡频率(fx)进行 8 倍频产生 48 MHz 的 USB 时钟。OPS2 电路对该时钟进行 3 分频，产生的 16 MHz 时钟被用作主时钟，以供 USB 功能控制器之外的外设使用。

6.3 寄存器

（1）处理器时钟控制寄存器（PCC）

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

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被置为 03H。

(1/2)

复位后：03H　　R/W　　地址：FFFFF828H

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

FRC注1, 2	副时钟片上反馈电阻的使用
0	已使用
1	未使用

MCK	主时钟振荡器控制
0	允许振荡
1	停止振荡

MFRC	主时钟片上反馈电阻的使用
0	已使用 (使用陶瓷/晶体谐振器时)
1	未使用 (使用外部时钟时)

CLS注3	CPU时钟 (f_{CPU})的状态
0	主时钟操作
1	副时钟操作

CK3	CK2	CK1	CK0	时钟选择 (f_{CLK}/f_{CPU})
0	0	0	0	f_{XX}
0	0	0	1	$f_{XX}/2$
0	0	1	0	$f_{XX}/4$
0	0	1	1	$f_{XX}/8$ (初始值)
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
1. FRC 位被置位(为 1)时，副时钟停止振荡。
2. 返回到 RTC 备份模式时，指定 RTCBUMCTL0. RBMSET (为 0) 然后置位 FRC 位 (为 1)。
3. CLS 为只读位。

(2/2)

注意事项

1. **当 CLKOUT 正在输出时，不要改变 CPU 时钟（使用 CK3 至 CK0 位）。**
2. **请使用位处理指令来操作 CK3 位。当使用 8 位处理指令时，不要改变 CK2 至 CK0 位的设定值。**
3. **使用外部时钟时，将 MFRC 位设置为“1”，这样就无需内部反馈电阻。**
4. **当系统将主时钟作为 CPU 时钟运行时，即使 MCK 位被置位(1)，主时钟的操作也不会停止。在 CPU 时钟被改变为副时钟之后，它才会停止。**
5. **将 MCK 位从 0 改变为 1 之前，停止那些运行于主时钟的片上外设。**
6. **当主时钟已停止设备运行于副时钟时，清除 MCK 位(0)，并在 CPU 时钟切换到主时钟或运行片上外设功能之前，通过软件确保振荡稳定时间。**

备注 ×：无需理会

（a） 主时钟运行改变为副时钟运行的示例：

<1> CK3 位 ← 1： 建议使用一个位处理指令。不要改变 CK2 至 CK0 位。

<2> 副时钟操作： 读取 CLS 位，检查副时钟是否开始工作。在 CK3 位设置完成后需要以下时间，副时钟才能工作：

最大值：1/f_{XT}（1/副时钟频率）。

<3> MCK 位 ← 1： 只有在主时钟停止时，才可以设定 MCK 位为 1 。

注意事项 1. 当停止主时钟时，停止锁相环。同样使用主时钟的片上外设也停止工作。

2. 如果以下条件不能满足，需要调整 CK2 至 CK0 位的值来满足条件，然后切换到副时钟工作模式。

内部系统时钟（fCLK）> 副时钟（f_{XT}：32.768 kHz）× 4

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

[说明示例]

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

备注 以上描述仅为举例。注意<2>部分，在一个闭合循环中检查 CLS 位。

（b） 副时钟工作改变为主时钟工作的示例：

<1> MCK 位 ← 0： 主时钟振荡器开始工作。

<2> 在程序中插入等待指令，直到确保经过主时钟振荡器稳定时间。

<3> CK3 位 ← 0： 建议使用一个位处理指令。不要改变 CK2 至 CK0 位。

<4> 主时钟工作： 从 CK3 位设置完成后，直到主时钟开始工作，需要以下时间：

最大值： 1/f_{XT}（1/副时钟频率）。

所以，CK3 位清为 0 后，应紧接着插入一条 NOP 指令，或通过读取 CLS 位来确定主时钟是否开始工作。

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

[说明示例]

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

（2）内部振荡模式寄存器（RCM）

RCM 寄存器是 8 位寄存器，用于设置内部振荡器的工作模式。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

复位后：00H　R/W　地址：FFFFF80CH

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

RSTOP	内部振荡器的振荡/停止
0	内部振荡器振荡
1	内部振荡器停止

注意事项 1. CPU 运行于内部振荡时钟时（CCLS.CCLSF 位 = 1），内部振荡器不能被停止。不要将 RSTOP 位设置为 1。

2. 因为发生中断导致 STOP 模式被取消之后，当振荡器信号正在稳定时（也就是说，如果 CCLS.CCLSF 位被置为 1），如果发生看门狗定时器溢出，即使内部振荡器被停止（RSTOP 位为 1），内部振荡器振荡。此时，RSTOP 保持置 1。

3. 通过选项字节设置的 RCM 寄存器设置才有效。详情参见第三十章 选项字节。

注意事项 1. CPU 运行于内部振荡时钟时（CCLS.CCLSF 位 = 1），内部振荡器不能被停止。不要将 RSTOP 位设置为 1。

2. 如果 CCLS.CCLSF 位设置为 1（振荡稳定期间发生看门狗定时器溢出），即使 RSTOP 位设置为 1 时，内部振荡器会继续振荡。同时，RSTOP 位保持为 1。

（3） CPU 工作时钟状态寄存器（CCLS）

CCLS 寄存器表明 CPU 工作时钟的状态。

该寄存器为只读寄存器，可以按字节读取或按位读取。

系统复位后，该寄存器被设为 00H。

复位后：00H[注]　R　地址：FFFFF82EH

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

CCLSF	CPU 工作时钟状态
0	工作于主时钟 (f_X) 或副时钟 (f_{XT})。
1	工作于内部振荡时钟 (f_R)。

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

（4）时钟直通选择寄存器（CKTHSEL）

在时钟直通模式下，CKTHSEL 寄存器用于选择时钟直通频率或时钟直通 2 分频。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

复位后： 00H　　R/W　　地址：FFFFF380H

	7	6	5	4	3	2	1	0
CKTHSEL	0	0	0	0	0	0	0	CKTHSEL0

CKTHSEL0	时钟直通选择寄存器
0	时钟直通频率
1	时钟直通频率2分频

6.4 操作

6.4.1 各时钟的操作

下表显示各个时钟的操作状态：

表 6-1. 各时钟的操作状态

寄存器设置和操作状态 / 目标时钟	PCC 寄存器									
	CLK 位 = 0，MCK 位 = 0					CLS 位 = 1，MCK 位 = 0		CLS 位 = 1，MCK 位 = 1		-注 3
	复位期间	振荡稳定时器计数期间	HALT 模式	IDLE1 模式，IDLE2 模式	STOP 模式	副时钟模式	Sub-IDLE 模式	副时钟模式	Sub-IDLE 模式	RTC 备份模式
主时钟振荡器 (f_X)	×	√	√	√	×	√	√	×	×	-注 4
副时钟振荡器 (f_{XT})	√	√	√	√	√	√	√	√	√	√
CPU 时钟 (f_{CPU})	×	×	×	×	×	√	×	√	×	-注 4
内部系统时钟 (f_{CLK})	×	×	√	×	×	√	×	√	×	-注 4
主时钟 (PLL 模式，f_{XX})	×	√注 1	√	√	×	√	√	×注 2	×注 2	-注 4
外设时钟 (f_{XX} 至 f_{XX}/1,024)	×	×	√	×	×	√	×	×	×	-注 4
WT 时钟 (主)	×	√	√	√	×	√	√	×	×	-注 4
WT 时钟 (副)	√	√	√	√	√	√	√	√	√	-注 4
WDT2 时钟 (内部振荡)	×	√	√	√	√	√	√	√	√	-注 4
WDT2 时钟 (主)	×	×	√	×	×	√	×	×	×	-注 4
WDT2 时钟 (副)	√	√	√	√	√	√	√	√	√	-注 4
RTC 时钟 (主)	×	×	√	√	×	√	√	×	×	-注 4
RTC 时钟 (副)	√	√	√	√	√	√	√	√	√	√

注
1. 入锁时间
2. 请确保将 PLLCTL.PLLON 清除为 0。
3. 因为 V_{DD} 低于保证的操作电压，寄存器的值不确定。
4. 因为 V_{DD} 低于保证的操作电压，操作状态不确定。

备注
√: 运行中
×: 停止
-: 不确定

6.4.2 时钟输出功能

时钟输出功能用于从 CLKOUT 引脚输出内部系统时钟（f_{CLK}）。

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

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

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

6.4.3 外部时钟信号输入

外部时钟信号可以直接输入到振荡器。从 X1 引脚输入时钟，并保持 X2 引脚开路。设置 PCC.MFRC 位为 1 (未使用片上反馈电阻)。但是请注意，即使当输入外部时钟信号时，需要一段时间来稳定振荡器信号。

6.5 PLL功能

6.5.1 概述

V850ES/JG3-L 单片机中，CPU 和片上外设功能的工作时钟可以选择 PLL 功能输出的 4 倍或 8 倍振荡频率，或选择无倍频时钟（时钟直通模式）。

(a) 未使用 USB 时

PLL 模式：	输入时钟 = 2.5 至 5 MHz (f_{XX} = 10 至 20 MHz)
时钟直通模式：	输入时钟 = 2.5 至 10 MHz (f_{XX} = 1.25 至 10 MHz)

(b) 使用 USB（未使用 UCLK）

PLL 模式：	f_X = 6 MHz (f_{XX} = 16 MHz)
时钟直通模式：	禁止设置

(c) 使用 USB（使用 UCLK）

PLL 模式：	输入时钟 = 2.5 至 4 MHz (f_{XX} = 10 至 16 MHz)
	输入时钟 = 6 MHz (f_{XX} = 16 MHz)
时钟直通模式：	输入时钟 = 2.5 至 10 MHz (f_{XX} = 1.25 至 10 MHz)

6.5.2 寄存器

（1）锁相环控制寄存器（PLLCTL）

PLLCTL 寄存器是 8 位寄存器，用于控制 PLL 功能。

寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设置为 01H。

复位后： 01H　　R/W　　地址： FFFFF82CH

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

SELPLL	CPU操作时钟模式的选择
0	时钟直通模式
1	PLL 模式

PLLON	PLL 操作的控制
0	禁止 PLL 操作
1	使能 PLL 操作 (在PLL操作开始之后，频率的稳定需要一段入锁时间)

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

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

（2） 时钟控制寄存器（CKC）

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

CKC 寄存器控制 PLL 模式下的内部系统时钟。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 0AH。

复位后： 0AH R/W 地址：FFFFF822H

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

CKDIV0	PLL模式下的内部系统时钟 (f_{XX})
0	f_{XX} = 4 × f_X (f_X = 2.5 至 5.0 MHz)
1	f_{XX} = 8 × f_X (f_X = 6 MHz)：使用USB时

注意事项 1. f_X = 5.0 至 10.0 MHz，不能使用 PLL 模式。但是，只有当使用 USB 功能控制器时，可以使用 PLL 模式（f_X = 6 MHz）。

2. 请确保在改变 CKC 寄存器设置之前设置为时钟直通模式(PLLCTL.SELPLL = 0)。

3. 由主时钟振荡频率(fx)产生 USB 时钟(f_{USB})时，为 fx 选择 6 MHz，将 CKC 寄存器设置为 0BH，并将 OCK2 寄存器设置为 11H，然后设置 PLL 模式 (PLLCTL.SELPLL = 1)。

（3）时钟选择寄存器（OCKS2）

OCKS2 寄存器选择 PLL 1/3 分频输出时钟。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

复位后： 00H R/W 地址：FFFFF348H

	7	6	5	4	3	2	1	0
OCKS2	0	0	0	OCKSEN2	0	0	0	OCKS20

OCKSEN2	1/3 分频的设置操作使能
0	停止 1/3 分频时钟操作
1	使能 1/3 分频时钟操作

OCKS20	指定PLL输出时钟的分频
0	1/1
1	1/3

注意事项 由主时钟振荡频率(fx)产生 USB 时钟(f_{USB})时，为 fx 选择 6 MHz，将 CKC 寄存器设置为 0BH，并将 OCK2 寄存器设置为 11H，然后设置 PLL 模式 (PLLCTL.SELPLL = 1)。

（4） 锁定寄存器（LOCKR）

上电后或 STOP 模式释放后，PLL 在给定频率上进行锁相，需要等待频率稳定的时间就是入锁时间（频率稳定时间）。直到稳定之前的状态称为入锁状态，稳定后的状态称为锁定状态。

LOCKR 寄存器包括 LOCK 位，该位用于反映 PLL 频率稳定状态。

该寄存器为只读寄存器，按字节读取或按位读取。

系统复位后，该寄存器被设为 00H。

复位后：00H　　R　　地址：　FFFFF824H

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

LOCK	PLL 锁存状态检查
0	锁定状态
1	未锁定状态

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

[设置条件]

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

注　复位将该寄存器设置为 01H，复位解除且振荡稳定时间之后清为 00H。

[清除条件]

- 在复位解除之后振荡稳定时间溢出时（OSTS 寄存器缺省时间（参见**第二十九章 选项字节**））。
- 在 STOP 模式解除后振荡稳定定时器（时间由 OSTS 寄存器设置）溢出时，该 STOP 模式是在 PLL 操作状态下设置的
- 在 PLL 锁存时间定时器（时间由 PLLS 寄存器设置）溢出时，当 PLLCTL.PLLON 位从 0 变到 1 后
- 在 IDLE2 模式释放时插入的建立时间（时间由 OSTS 寄存器设置）完成后，该 IDLE2 模式是在 PLL 操作状态下设置的

（5） PLL 锁定时间规范寄存器（PLLS）

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

该寄存器可以按字节读取或写入。

系统复位后，该寄存器被置为 03H。

复位后：03H　　R/W　　地址：FFFFF6C1H

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

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

注意事项 1. 如此设置，入锁时间最少为 800μs。

2. 入锁期间不要改变 PLLS 寄存器的设置。

6.5.3 用法

（1）当使用锁相环 PLL 时

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

（a） 从时钟直通模式转变至 IDLE2 或 STOP 模式

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

（b） 在 PLL 操作模式中转变至 IDLE 2 或 STOP 模式

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

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

（2）当未使用锁相环 PLL 时

- 复位信号释放后，选择时钟直通模式（SELPLL 位= 0），但是锁相环处于工作状态（PLLON 位= 1），因此必须将其停止（PLLON 位= 0）。
  从 IDLE2 或 STOP 模式恢复所需要的时间如下：

- STOP 模式： 设置 OSTS 寄存器，使得振荡稳定时间最小为 1 ms。
- IDLE2 模式： 设置 OSTS 寄存器，使得建立时间最小为 350 μs。

6.6 如何连接谐振器

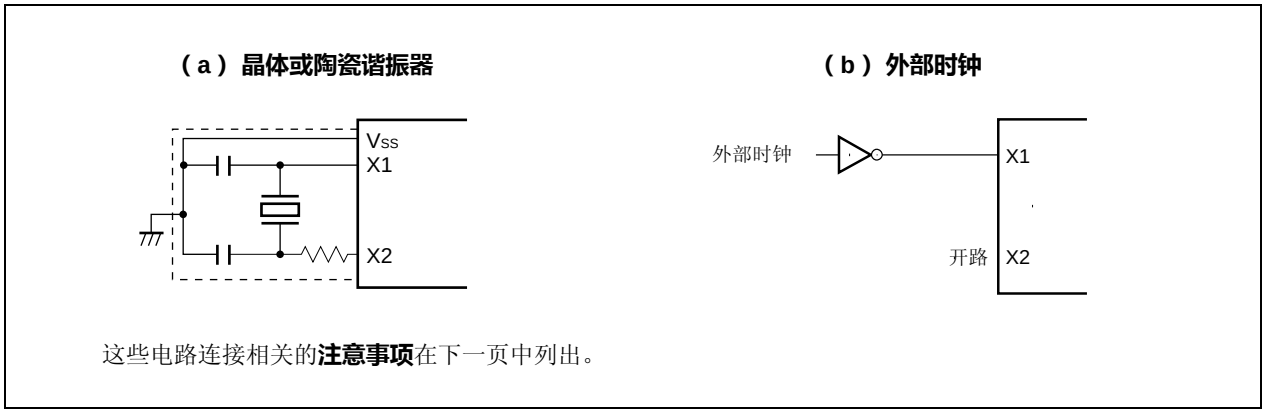
6.6.1 主时钟振荡器

由晶体谐振器或陶瓷谐振器振荡产生的信号连接到 X1 和 X2 引脚，输入到主时钟振荡器。谐振器的频率是 2.5 至 10 MHz。

外部时钟信号也可以输入到主时钟振荡器。

图 6-2 展示了连接到主时钟振荡器的电路示例。

图 6-2. 连接到主时钟振荡器的电路示例

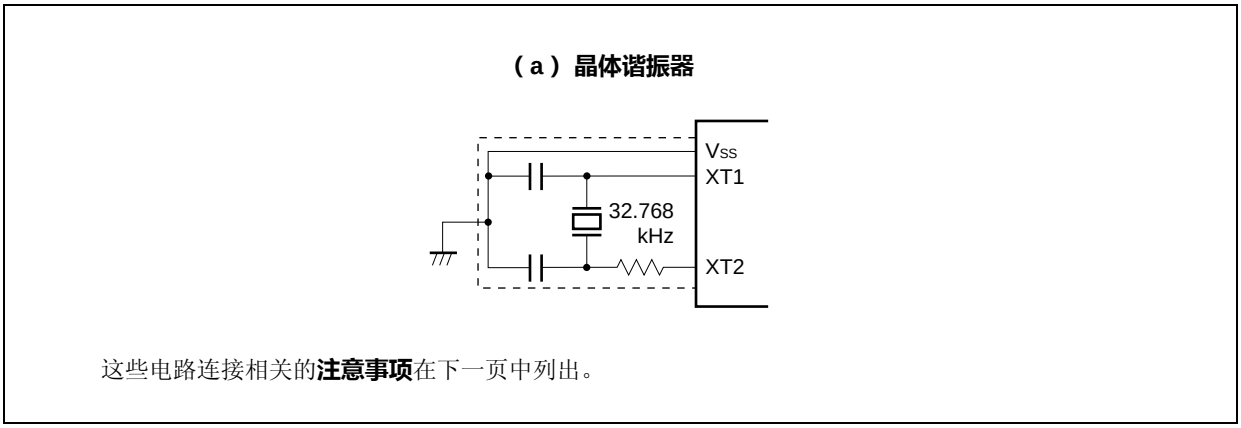


6.6.2 副时钟振荡器

由晶体谐振器或陶瓷谐振器振荡产生的信号连接到 XT1 和 XT2 引脚，输入到副时钟振荡器。谐振器的频率是 32.768 kHz（标准值）。

图 6-3 展示了连接到副时钟振荡器的电路示例。

图 6-3. 连接到副时钟振荡器的电路示例



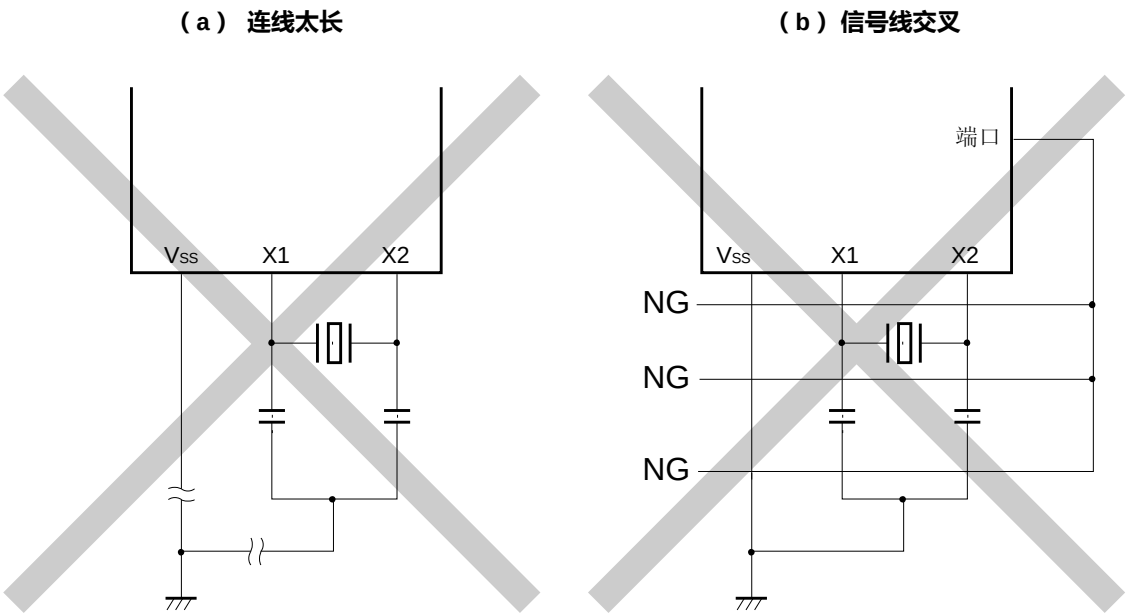
注意事项 1. 当使用 X1 振荡器和 XT1 振荡器时，为了避免线电容引起的不利影响，图 6-2 和图 6-3 中虚线包围区域的连线应该注意下列规则。

- **保证连线尽可能最短。**
- **该连线不要和其它信号线交叉。**
- **布线时不要将该连线靠近电流变化很大的信号线。**
- **总是保持振荡器电容的接地点与 Vss 的电位相同。**
- **不要将电容连接到有大电流通过的接地点**
- **不要从振荡器拾取信号**

副时钟振荡器的设计是通过低增益电路来降低功耗。因此，使用副时钟时需要特别注意布线方式。

图 6-4 显示了不正确的谐振器连接的示例。

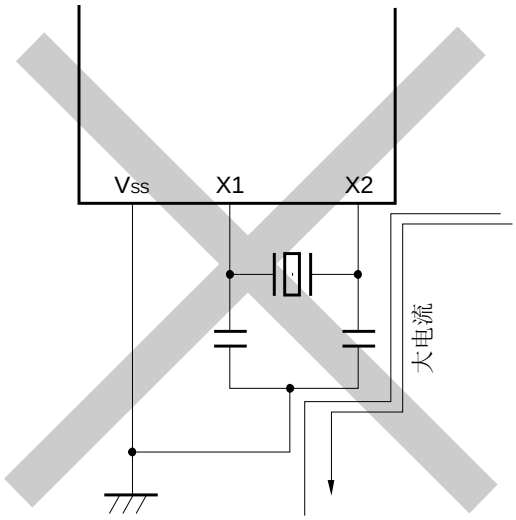
图 6-4. 不正确的谐振器连接示例（1/2）



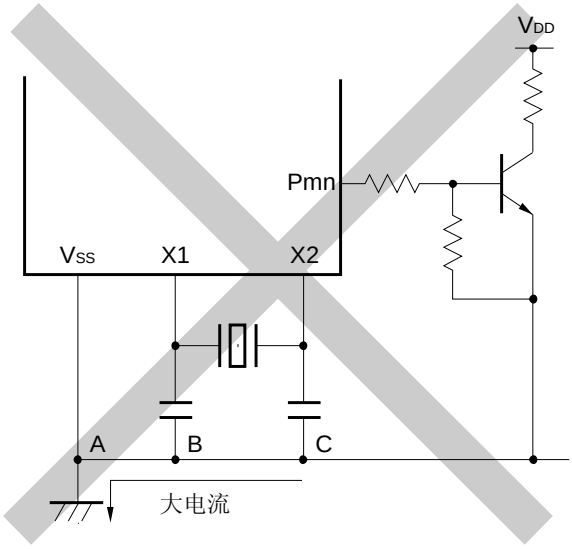
备注 使用副系统时钟时，分别使用 XT1 和 XT2 分别代替 X1 和 X2。另外，在 XT2 侧插入串联电阻。

图 6-4. 不正确的谐振器连接的示例（2/2）

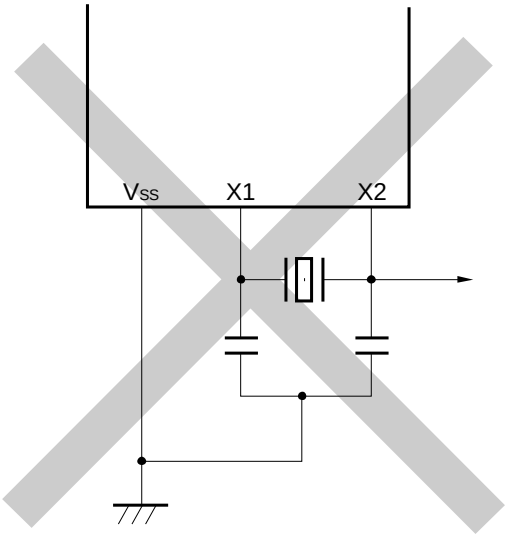
（c） 连线附近有变化很大的电流



(d) 电流流经振荡器的接地线
（点 A、B 和 C 处的电位波动）



（e） 拾取信号



备注 使用副系统时钟时，分别使用 XT1 和 XT2 分别代替 X1 和 X2。注意，必须在 XT2 侧串联插入电阻。

注意事项 当 X2 和 XT1 的连线平行时，X2 的串扰噪声可能会随 XT1 增加，可能会导致误动作。

第七章 16 位定时器/事件计数器 P（TMP）

定时器 P（TMP） 是 16 位定时器/事件计数器。
V850ES/JG3-L 单片机有六个定时器/事件计数器通道：TMP0 至 TMP5（TMPn）。

7.1 概述

TMPn 具有下列特征：

(1) **间隔定时器**
TMPn 按照预置的间隔产生一个中断，可以输出一个方波。

(2) **外部事件计数器**
TMPn 对从外部输入的信号脉冲数量进行计数。

(3) **外部触发脉冲输出**
当指定的外部信号输入时，TMPn 开始计数并输出一个脉冲。

(4) **单次触发脉冲输出**
TMPn 输出一个单次触发脉冲，输出宽度可以自由指定。

(5) **PWM 输出**
TMPn 输出一个固定宽度的脉冲，有效电平宽度可以改变。
即使在定时器工作时，脉冲占空比也可以自由改变。

(6) **自由运行定时器**
TMPn 从 0000H 递增到 FFFFH，然后重置。

(7) **脉冲宽度测量**
TMPn 可以用来测量外部输入信号的脉冲。

备注 n =0 至 5

7.2 配置

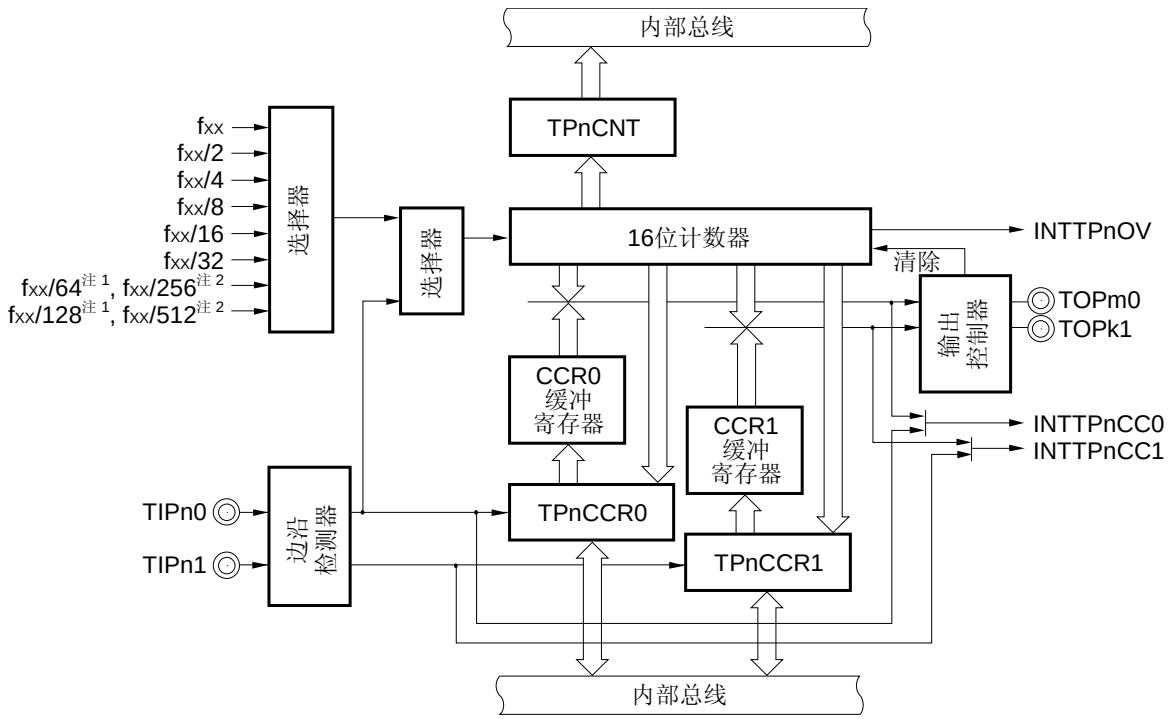
TMPn 包括下列硬件：

表 7-1. 定时器 TMPn 的配置

项目	配置
寄存器	16 位计数器 TMPn 计数器读取缓冲寄存器（TPnCNT） TMPn 捕获/比较寄存器 0，1（TPnCCR0，TPnCCR1） CCR0，CCR1 缓冲寄存器 TMPn 控制寄存器 0，1（TPnCTL0，TPnCTL1） TMPn 输入/输出控制寄存器 0 至 2（TPnIOC0 至 TPnIOC2） TMPn 选项寄存器 0（TPnOPT0）
定时器输入	2（TIPm0，TIPk1 引脚）
定时器输出	2（TOPn0，TOPn1 引脚）

备注 **1.** n =0 至 5
2. m =0, 2 至 5
3. k =2 至 5

图 7-1. 定时器 TMPn 框图



注 **1.** TMP0，TMP2，TMP4
2. TMP1，TMP3，TMP5

备注 fxx：主时钟频率
n =0 至 5
m =0,2 至 5
k =2 至 5

（1）16 位计数器

此 16 位计数器可以计数内部时钟或外部事件。

该计数器的计数值可以通过读取 TPnCNT 寄存器得到。

当 TPnCTL0.TPnCE 位为 0 时，计数器停止，计数器的值为 FFFFH。如果此时读取 TPnCNT 寄存器，则读回的值为 0000H。

系统复位将 TPnCE 位清 0，会停止计数器，并将 16 位计数器的值设置为 FFFFH。

（2）TMPn 计数器读取缓冲寄存器（TPnCNT）

TPnCNT 寄存器是读出缓冲寄存器，可以读取 16 位计数器的计数值。

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

根据不同工作模式，TPnCCR1 寄存器可以用作捕获寄存器或比较寄存器。

（4）CCR0 缓冲寄存器

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

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

CCR0 缓冲寄存器不能直接读写。

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

（5）CCR1 缓冲寄存器

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

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

CCR1 缓冲寄存器不能直接读写。

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

（6）TMPn 控制寄存器 0 和 1（TPnCTL0 和 TPnCTL1）

TPnCTL0 寄存器是 8 位寄存器，用于控制定时器 TMPn 的操作。

（7）TMPn 输入/输出控制寄存器 0 至 2（TPnIOC0 至 TPnIOC2）

这些是 8 位寄存器，用于控制 TMPn 的输入和输出（TOPn0，TOPn1 引脚）。

（8）TMPn 选项寄存器 0（TPnOPT0）

该寄存器是 8 位寄存器，用来控制设置的规范，比如捕获和比较。

（9）边沿检测器

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

（10）输出控制器

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

（11）选择器

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

7.2.1 TMPn使用的引脚

TMPn 使用的输入和输出引脚如表 7-2 所示。当这些引脚用于 TMPn 时，首先设置它们的端口模式。

详情参见**表 4-15 端口引脚用作复用功能时的设置**。

表 7-2. TMPn 使用的引脚

定时器通道	引脚编号		端口	TMP 输入	TMP 输出	复用功能
	GC	F1				
TMP0	27	L4	P32	TIP00 注	TOP00	ASCKA0/$\overline{\text{SCKB4}}$
TMP1	–	–	–	–	–	–
TMP2	50	J11	P97	TIP20 注	TOP20	A7/SIB1/RXDC0
	49	K11	P96	TIP21	TOP21	A6/TXDC0
TMP3	48	K10	P95	TIP30 注	TOP30	A5/RXDA5
	47	L10	P94	TIP31	TOP31	A4/TXDA5
TMP4	46	J9	P93	TIP40 注	TOP40	A3/RXDA4
	45	K9	P92	TIP41	TOP41	A2/TXDA4
TMP5	58	G8	P915	TIP50 注	TOP50	A15/INTP6
	57	G9	P914	TIP51	TOP51	A14/INTP5

注 TIPn0 引脚可以用作捕获触发输入、外部事件输入、及外部触发输入 (n =0, 2 至 5)。

注意事项 除了上述复用功能之外，UART5 的 INTUA5T 中断和 TMP3 的 INTP3CC1 中断、以及 UART5 的 INTUA5R 中断和 TMP3 的 INTP3OV 中断是复用中断信号，因此，它们不能同时使用。

备注 GC：100 针塑封 LQFP (密间距) (14 ×14)
F1：121 针塑封 FBGA (8 × 8)

7.2.2 中断

TMPn 使用下列三种类型的中断信号：

(1) INTTPnCC0

当 16 位计数器的值与 CCR0 缓冲寄存器的值匹配时，或有捕获信号输入到 TIPn0 引脚时，产生该信号。

(2) INTTPnCC1

当 16 位计数器的值与 CCR1 缓冲寄存器的值匹配时，或有捕获信号输入到 TIPn1 引脚时，产生该信号。

(3) INTTPnOV

当 16 位计数器的值递增达到 FFFFH 之后溢出，则产生该信号。

7.4 寄存器

以下寄存器控制 TMPn：

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

备注 **1.** 当使用 TIPn0，TIPn1，TOPn0 和 TOPn1 引脚的功能时，参见**表 4-15 端口引脚用作复用功能时的设置**。
2. n =0，2 至 5

（1）TMPn 控制寄存器 0（TPnCTL0）

TPnCTL0 寄存器是 8 位寄存器，用于控制定时器 TMPn 的操作。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

随时可以用软件将相同的值写入 TPnCTL0 寄存器。

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

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

(n = 0 至 5)

TPnCE	TMPn 操作控制
0	TMPn 操作禁止(TMPn 异步复位[注])。
1	TMPn 操作允许。TMPn 操作开始。

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

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

注意事项　**1. 当 TPnCE 位=0 时，设置 TPnCKS2 至 TPnCKS0 位为“1”。**
当 TPnCE 位的值从 0 变为 1 时，可以同时设定 TPnCKS2 至 TPnCKS0 位。
2. 请确保将第 3 位至第 6 位清为“0”。

备注　f_{XX}：主时钟频率

（2）TMPn 控制寄存器 1（TPnCTL1）

TPnCTL1 寄存器是 8 位寄存器，用于控制定时器 TMPn 的操作。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

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

	7	<6>	<5>	4	3	2	1	0
TPnCTL1	0	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0

(n = 0, 2至 5)

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

TPnEEE	计数时钟选择
0	禁止使用外部事件计数输入操作。 (对 TPnCTL0.TPnCK0 至TPnCK2位所选的计数时钟计数。)
1	允许使用外部事件计数输入操作。 (对外部事件计数输入信号的有效沿计数。)

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

注意事项

1. 只有在外触发脉冲输出模式或单次触发脉冲输出模式下，TPnEST 位有效。在其它模式下，向该位写入 1 的操作都被忽略。

2. 不论 TPnEEE 位的值如何，在外部事件计数模式下，都会选择外部事件输入。

3. 当定时器操作停止时（TPnCTL0.TPnCE =0），设置 TPnEEE 和 TPnMD2 至 TPnMD0 位（TPnCE 位=1 时，可以写入相同的值）。在 TPnCE 位为 1 时，重写操作是无法保证的，若重写发生错误，应把 TPnCE 位清为 0，再重新对这些位进行设定。

4. 请确保将第 3 位，第 4 位和第 7 位清为“0”。

（3）TMPn 输入/输出控制寄存器 0（TPnIOC0）

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

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

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

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

(n = 0 至 5)

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

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

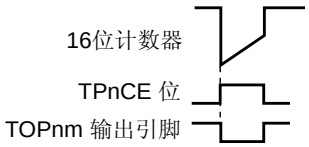
TPnOL0	TOPn0引脚输出电平设置[注]
0	TOPn0引脚开始输出高电平
1	TOPn0引脚开始输出低电平

TPnOE0	TOPn0 引脚输出设置
0	定时器输出禁止 · 当TPnOL0位= 0时：TOPn0引脚输出低电平 · 当TPnOL0位= 1时：TOPn1引脚输出高电平
1	定时器输出允许 (从TOPn0引脚输出方波)。

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

· 当TPnOLm 位 = 0时

· 当TPnOLm 位 = 1时



注意事项　**1. 当 TPnCTL0.TPnCE 位 =0 时，重新写入 TPnOL1，TPnOE1，TPnOL0 和 TPnOE0 位（当 TPnCE 位=1 时，可以写入相同的值）。如果重新写入不能正确执行，将 TPnCE 位清为 0，然后重新设定该位。**

2. 即使在 TPnCE 和 TPnOEm 位均为 0 时对 TPnOLm 位进行操作，TOPnm 引脚的输出电平也会改变（m = 0，1）。

（4）TMPn 输入/输出控制寄存器 1（TPnIOC1）

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

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

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

	7	6	5	4	3	2	1	0
TPnIOC1	0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0

(n = 0, 2 至5)

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

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

注意事项

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

（5）TMPn 输入/输出控制寄存器 2（TPnIOC2）

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

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

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

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

(n = 0, 2 至5)

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

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

注意事项

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

（6）TMPn 选项寄存器 0（TPnOPT0）

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

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H　R/W　地址: TP0OPT0 FFFFF595H, 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, 2 至5)

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

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

TPnOVF	TMPn溢出检测标志
置位 (1)	发生溢出
复位(0)	TPnOVF 位写入0,或 TPnCTL0.TPnCE 位 = 0

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

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

（7）TMPn 捕获/比较寄存器 0（TPnCCR0）

根据不同工作模式，TPnCCR0 寄存器可以用作捕获寄存器或比较寄存器。

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

TPnCCR0 寄存器可以在操作中进行读取和写入。

此寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

注意事项 在以下状态禁止访问 TPnCCR0 寄存器。而且，如果系统处于等待状态，取消等待状态的唯一方式就是执行复位。详细说明，参见 3.4.9（1）访问片上外设输入/输出专用寄存器。

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

复位后：0000H　R/W　地址：TP0CCR0 FFFFF596H, TP1CCR0 FFFFF5A6H,
TP2CCR0 FFFFF5B6H, TP3CCR0 FFFFF5C6H
TP4CCR0 FFFFF5D6H, TP5CCR0 FFFFF5E6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR0																

(n = 0 至 5)

（a）用作比较寄存器时的功能

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

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

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

（b）用作捕获寄存器时的功能

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

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

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

表 7-3. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

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

备注 关于随时写入和批量写入的详情，参见 **7.4 (2) 随时写入和批量写入**。

（8）TMPn 捕获/比较寄存器 1（TPnCCR1）

根据不同工作模式，TPnCCR1 寄存器可以用作捕获寄存器或比较寄存器。

在自由计时器模式下，根据 TPnOPT0.TPnCCS1 位的设定，TPnCCR1 寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TPnCCR1 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以被用作比较寄存器。

TPnCCR1 寄存器可以在操作中进行读取和写入。

此寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。

注意事项 在以下状态禁止访问 TPnCCR1 寄存器。而且，如果系统处于等待状态，取消等待状态的唯一方式就是执行复位。详细说明，参见 3.4.8（2）访问片上外设输入/输出专用寄存器。

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

复位后：0000H　R/W　地址：TP0CCR1 FFFFF598H, TP1CCR1 FFFFF5A8H, TP2CCR1 FFFFF5B8H, TP3CCR1 FFFFF5C8H, TP4CCR1 FFFFF5D8H, TP5CCR1 FFFFF5E8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR1 (n = 0 至 5)																

（a）用作比较寄存器时的功能

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

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

（b）用作捕获寄存器时的功能

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

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

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

表 7-4. 捕获/比较寄存器在各模式下的功能以及如何写入比较寄存器

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

备注 关于随时写入和批量写入的详情，参见 **7.4 (2) 随时写入和批量写入**。

（9）TMPn 计数器读取缓冲寄存器（TPnCNT）

TPnCNT 寄存器是读出缓冲寄存器，可以读取 16 位计数器的计数值。

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

该寄存器为只读寄存器，按 16 位宽度读取。

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

复位后，因为 TPnCE 位被清为 0，TPnCNT 寄存器的值被清为 0000H。

注意事项 在以下状态中，禁止访问 TPnCNT 寄存器。而且，如果系统处于等待状态，取消等待状态的唯一方式就是执行复位。详细说明，参见 3.4.8（2）访问片上外设输入/输出专用寄存器。

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

复位后：0000H　R　地址：TP0CNT FFFFF59AH, TP1CNT FFFFF5AAH, TP2CNT FFFFF5BAH, TP3CNT FFFFF5CAH TP4CNT FFFFF5DAH, TP5CNT FFFFF5EAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCNT																

(n = 0 至 5)

7.4 操作

对 V850ES/JG3-L 而言，各个通道可以用作不同的模式。下表列举了各个通道可以使能的各种模式。

表 7-5. TMPn 操作模式

操作模式	TMP0	TMP1	TMP2	TMP3	TMP4	TMP5
间隔计时器模式	受限可用 注1	受限可用 注2	可用	可用	可用	可用
外部事件计数器模式	受限可用 注1	不可用	可用	可用	可用	可用
外部触发脉冲输出模式	不可用	不可用	可用	可用	可用	可用
单次触发脉冲输出模式	受限可用 注1	不可用	可用	可用	可用	可用
PWM 输出模式	不可用	不可用	可用	可用	可用	可用
自由定时器模式	受限可用 注1	不可用	可用	可用	可用	可用
脉冲宽度测量模式	受限可用 注1	不可用	可用	可用	可用	可用

注 **1.** 因为 TIP01 和 TOP01 不存在，使用这些引脚的模式不可用。
2. 因为 TIP10、TIP11、TOP10 和 TOP11 不存在，使用这些引脚的模式不可用。

TMPn 可以执行下列操作：

表 7-6. TMPn 操作模式

操作模式注1	TPnCTL1.TPnEST 位 (软件触发位)	TIPn0 引脚 (外部触发输入)	捕获/比较寄存器设置	比较寄存器写入	计数时钟
间隔计时器模式	无效	无效	仅作比较寄存器	随时写入	内部/外部
外部事件计数器模式注2	无效	无效	仅作比较寄存器	随时写入	外部
外部触发脉冲输出模式注3	有效	有效	仅作比较寄存器	批量写入	内部
单次触发脉冲输出模式注3	有效	有效	仅作比较寄存器	随时写入	内部
PWM 输出模式	无效	无效	仅作比较寄存器	批量写入	内部/外部
自由定时器模式	无效	无效	可以切换	随时写入	内部/外部
脉冲宽度测量模式注3	无效	无效	仅作比较寄存器	无法写入	内部

注 **1.** 通道 1 只能用于间隔定时器模式。请注意，当使用通道 1 时，使能 INTTP1CC1 中断信号，则它的复用功能 USB 中断 INTUSBF0 不能使用。结果，使用通道 1 时 USB 功能控制器无法使用。
2. 在外部事件计数器模式下，应该指定对 TIPn0 引脚捕获触发输入的有效边沿不作检测（将 TPnIOC1.TPnIS1 和 TPnIOC1.TPnIS0 位清为 0）。
3. 在外部触发脉冲输出模式、单次触发脉冲输出模式和脉冲宽度测量模式下，选择内部时钟作为计数时钟。（将 TPnCTL1.TPnEEE 位清为 0）。

(1) 计数器基本操作

16 位计数器的基本操作描述如下。更多细节，参见各种操作模式的描述。

(a) 开始计数

在所有模式下，TMPn 从 FFFFH 开始计数，然后递增：FFFFH, 0000H, 0001H, 0002H, 0003H….

(b) 清除 TMPn

当 TMPn 的值与比较寄存器的值匹配时，或在有效捕获触发信号输入时 TMPn 的值被捕获，TMPn 被清除为 0000H。

请注意，当 TMPn 开始计数后，它立即从 FFFFH 递增到 0000H 并有一个溢出，这并不意味着 TMPn 已经被清除。于是，这种情况下不会产生 INTTPnCC0 和 INTTPnCC1 中断。

(c) 溢出

在自由运行定时器模式和脉宽测量模式下，当 TMPn 从 FFFFH 递增到 0000H 之后，发生溢出。溢出会将 TPnOPT0.TPnOVF 位设置为 1，并产生一个中断请求信号 (INTTPnOV)。请注意，在下列情况时，不能保证 INTTPnOV：

- TMPn 刚开始计数。
- 用于清除 TMPn 的比较器对比值被指定为 FFFFH。
- 在脉宽测量模式下，当 TMPn 的 FFFFH 值被捕获而导致 TMPn 被清除之后，从 FFFFH 递增到 0000H。

注意事项 在发生 INTTPnOV 溢出中断请求信号之后，请务必确认溢出标志(TPnOVF)被设置为 1。

(d) 正在递增时读取 TMPn

TMPn 正在递增时可以通过 TPnCNT 寄存器来读取。

具体来说，当 TPnCLT0.TpnCE 位为 1 时，TMPn 的值可以通过 TPnCNT 寄存器来读取。但是，请注意，当 TPnCLT0.TpnCE 位为 0 时，TMPn 的值总是 FFFFH 且 TPnCNT 寄存器的值总是 0000H。

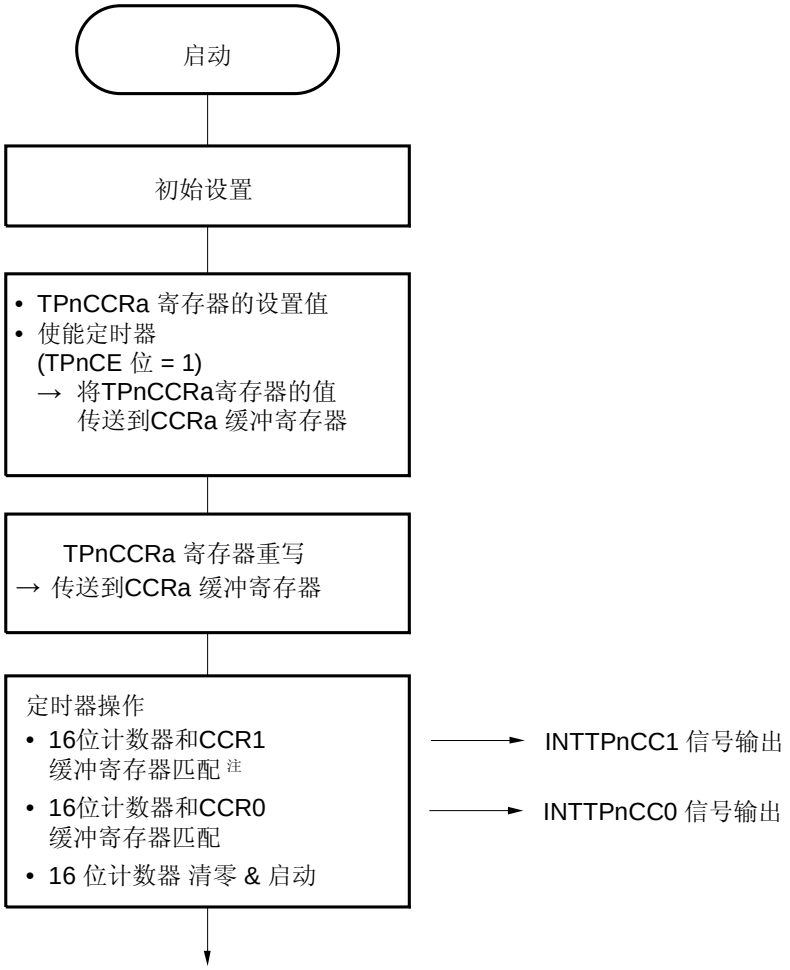
(2) 随时写入和批量写入

在 TMPn 定时器操作期间（也就是 TPnCTL0.TPnCE 位为 1 时），可以写入 TPnCCR0 和 TPnCCR1 寄存器，但是 CCR0 和 CCR1 缓冲寄存器的写入方法取决于不同的模式。两种写入方式就是随时写入和批量写入。

(a) 随时写入

在定时器操作期间的任意时刻，数据可以从 TPnCCR0 和 TPnCCR1 寄存器传送到 CCR0 和 CCR1 缓冲寄存器。

图 7-2. 随时写入的基本操作流程图

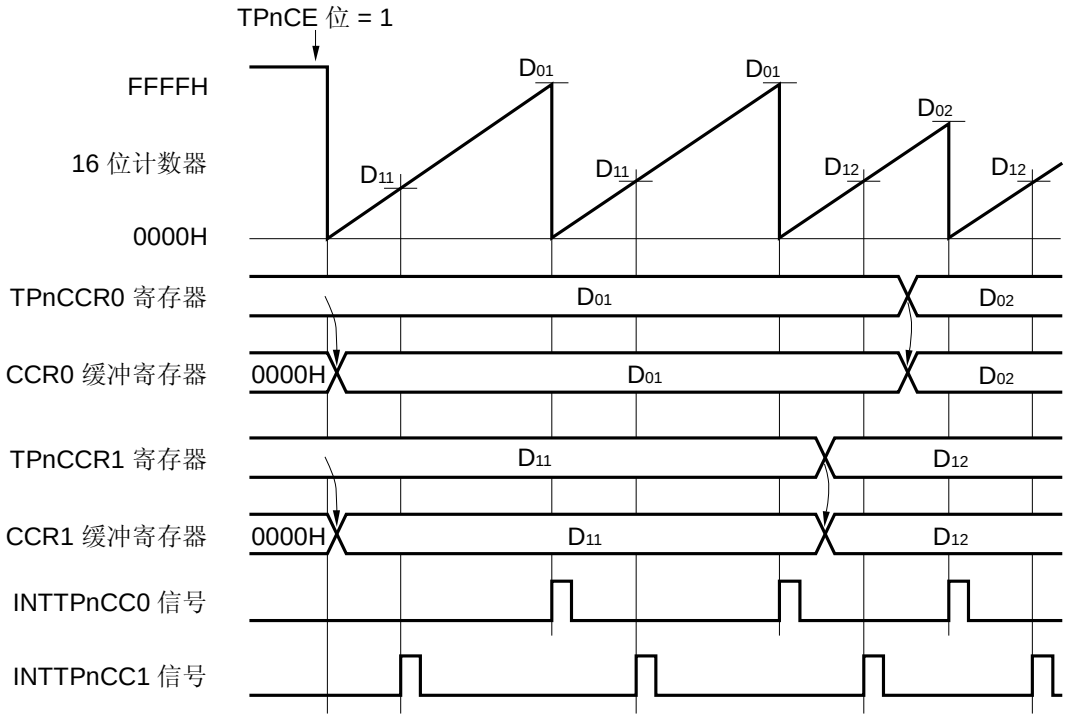


注 只有在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配时，才清除 16 位计数器。与 CCR1 缓冲寄存器的值匹配，不清除。

备注 **1.** 上述流程图适用于 TMPn 用作间隔定时器时。

2. a = 0, 1

图 7-3. 随时写入的时序示例



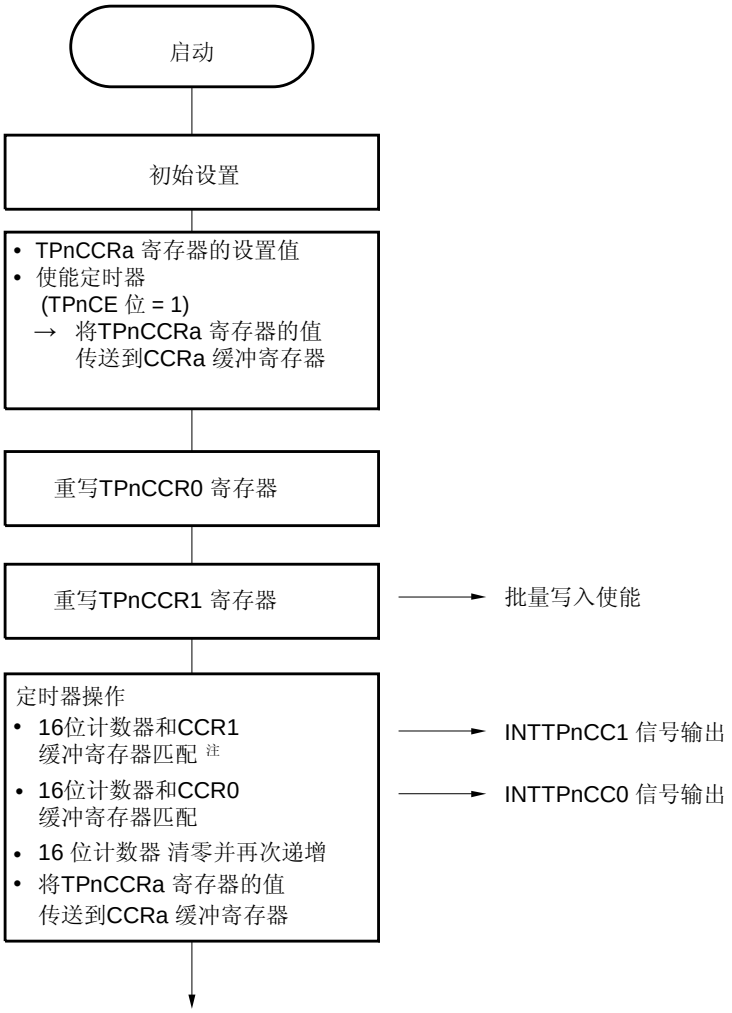
备注 **1.** D_{01}, D_{02}：TPnCCR0 寄存器的设置值
D_{11}, D_{12}：TPnCCR1 寄存器的设置值
2. 上述流程图适用于 TMP 用作间隔定时器时。

(b) 批量写入

该模式下，在定时器操作期间，数据一次性从 TPnCCR0 和 TPnCCR1 寄存器传送到 CCR0 和 CCR1 缓冲寄存器。在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配时，传送该数据。写入 TPnCCR1 寄存器就可以使能传送。是否写入 TPnCCR1 寄存器可以控制使能或禁止下一次传送时序。

为了让重写的 TPnCCR0 和 TPnCCR1 寄存器设置值成为 16 位计数器比较值（换句话说，为了该值被传送到 CCR0 和 CCR1 缓冲寄存器），必须在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配之前写入 TPnCCR1 寄存器，然后重写 TPnCCR0 寄存器。因此，在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配时，TPnCCR0 和 TPnCCR1 寄存器的值被传送到 CCR0 和 CCR1 缓冲寄存器。所以，即使只想重写 TPnCCR0 寄存器的值，也需要将相同的值（和 TPnCCR1 寄存器的当前值相同）写入 TPnCCR1 寄存器。

图 7-4. 批量写入的基本操作流程图



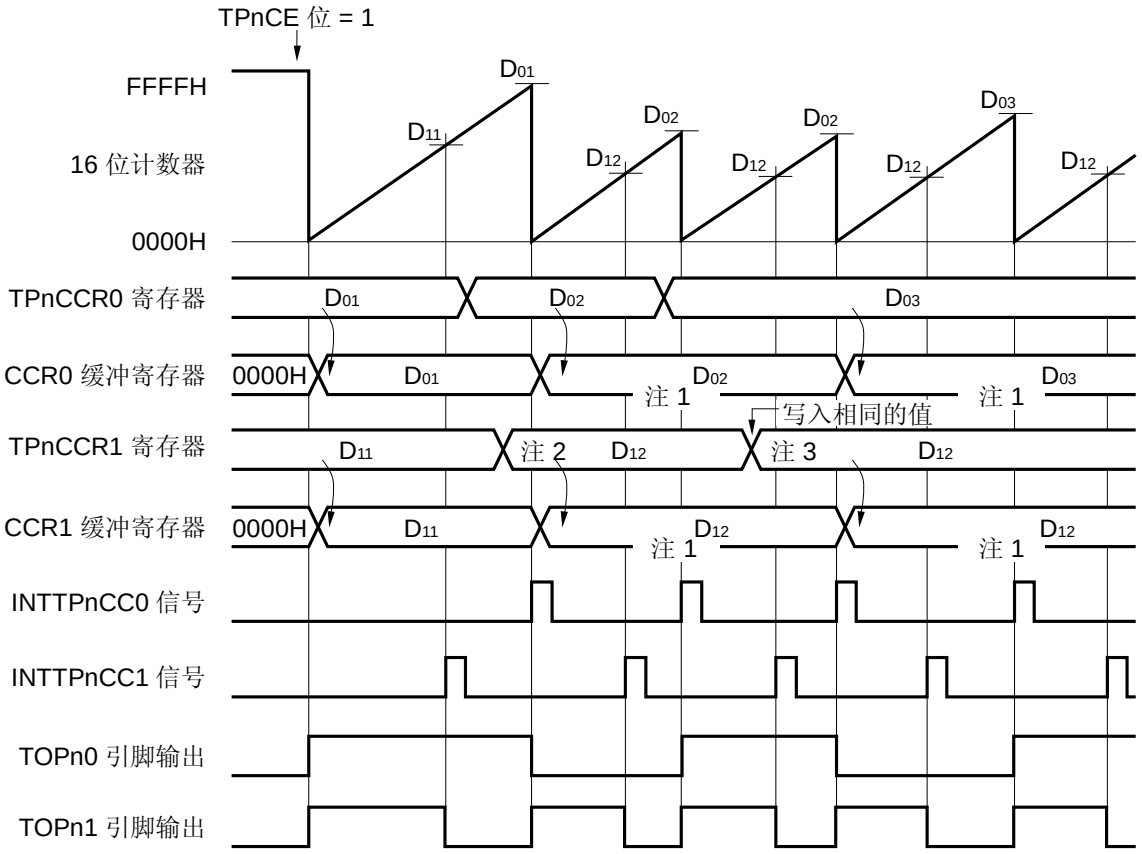
注 只有在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配时，才清除 16 位计数器。与 CCR1 缓冲寄存器的值匹配，不清除。

注意事项 TPnCCR1 寄存器的写入处理包括批量写入的使能。于是，重写 TPnCCR1 寄存器的操作请放在重写 TPnCCR0 寄存器之后。

备注 **1.** 上述流程图适用于 TMPn 用作 PWM 输出模式时。

2. a = 0, 1

图 7-5. 批量写入的时序示例



注 **1.** 因为 TPnCCR1 寄存器没有重写，D_{03} 没有被传送。

2. 因为 TPnCCR1 寄存器已经被重写(D_{12})，在 16 位计数器的值和 TPnCCR0 寄存器的值(D_{01})匹配时，数据被传送到 CCR1 缓冲寄存器 。

3. 因为 TPnCCR1 寄存器已经被重写(D_{12})，在 16 位计数器的值和 TPnCCR0 寄存器的值(D_{02})匹配时，数据被传送到 CCR1 缓冲寄存器 。

备注 **1.** D_{01}, D_{02}, D_{03}：TPnCCR0 寄存器的设置值

D_{11}, D_{12}：TPnCCR1 寄存器的设置值

2. 上述流程图适用于 TMPn 用作 PWM 输出模式时。

7.4.1 间隔定时器模式（TPnMD2 至TPnMD0 位 = 000）

在间隔定时器模式下，如果 TPnCTL0.TPnCE 位被置为 1，则会以指定时间间隔生成中断请求信号（INTTPnCC0）。将 TPnCE 位设置为 1，也可以启动定时器，然后从 TOPn0 引脚输出方波，指定的时间间隔等于方波的半周期。

通常在间隔定时器模式下不使用 TPnCCR1 寄存器。通过设置中断屏蔽标志（TPnCCMK1）可以屏蔽来自该寄存器的中断。

备注 **1.** 关于如何设置 TOPn0 引脚，参见**表 7-2 TMPn 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。
2. 关于如何使能 INTTPnCC0 中断信号，参见**第 22 章 中断服务/异常处理。**

图 7-6. 间隔定时器的配置

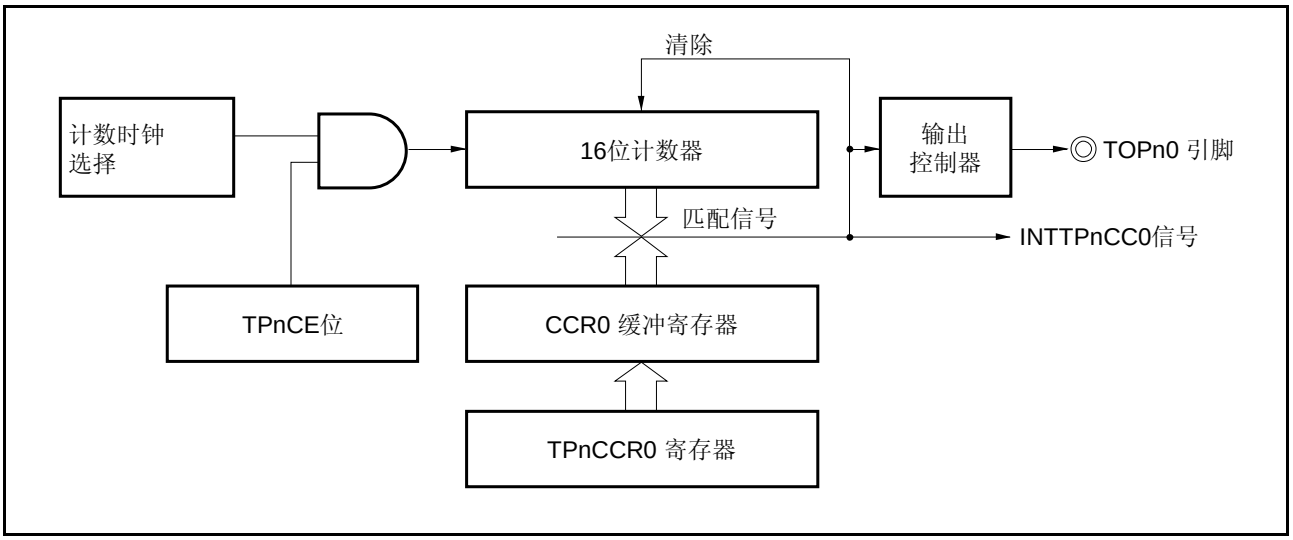
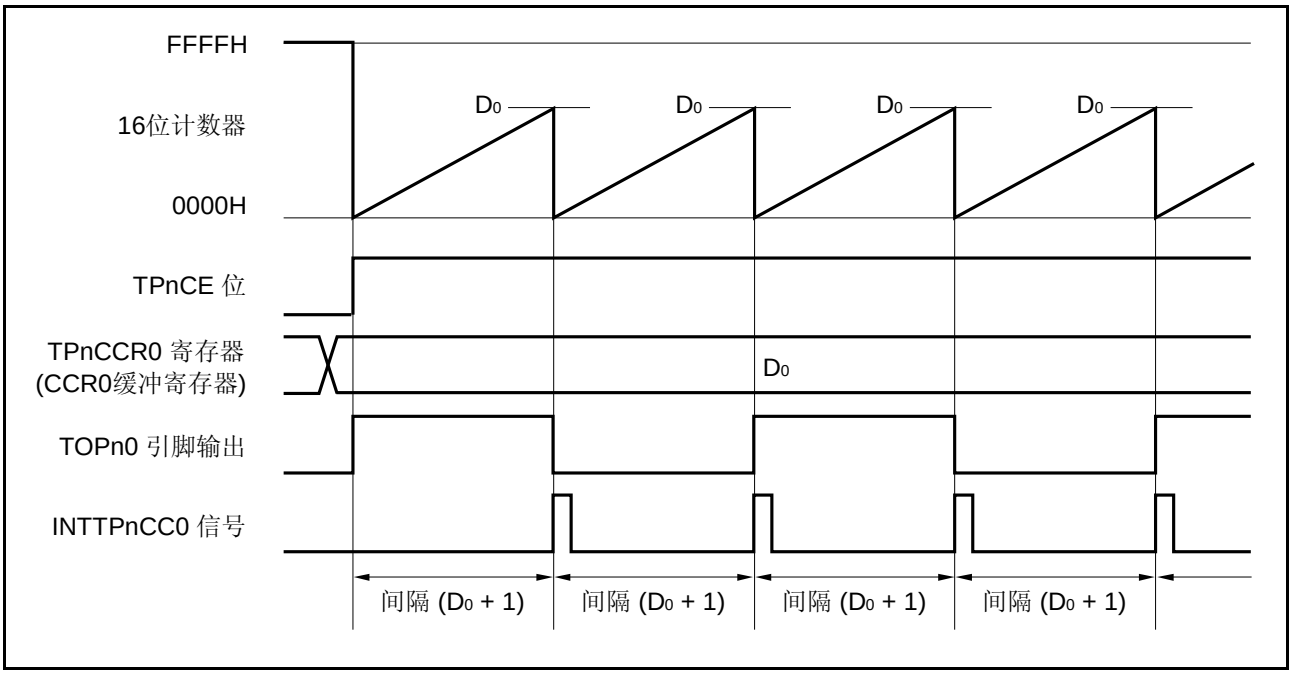


图 7-7. 间隔定时器模式的基本时序



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

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

时间间隔可用下述表达式计算。

间隔时间 =（TPnCCR0 寄存器的设定值＋1）× 计数时钟周期

使用间隔定时器模式时设置寄存器的示例如下。

图 7-8. 间隔定时器模式操作下的寄存器设置（1/2）

（a）TMPn 控制寄存器 0（TPnCTL0）

	TPnCE					TPnCKS2	TPnCKS1	TPnCKS0
TPnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

选择计数时钟

0: 停止计数
1: 允许计数

（b）TMPn 控制寄存器 1（TPnCTL1）

		TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0	0/1注	0	0	0	0	0

0, 0, 0:
间隔定时器模式

0: 计数时钟由TPnCKS0至TPnCKS2位选择。
1: 使用外部事件计数输入信号计数

注 使用定时器输出（TOPn1）时，TPnEEE 位只能被设置为 1。请注意，当 TPnEEE 位设置为 1 时，TPnCCR0 和 TPnCCR1 寄存器必须要设置为相同的值（也就是说，与已经为这些寄存器设置的值相同）。（详情参见 **7.4.1（2）（d）TPnCCR1 寄存器的操作**）。

图 7-8. 间隔定时器模式操作下的寄存器设置（2/2）

（c）TMPn 输入/输出控制寄存器 0（TPnIOC0）

					TPnOL1	TPnOE1	TPnOL0	TPnOE0
TPnIOC0	0	0	0	0	0/1	0/1	0/1	0/1

0: 禁止 TOPn0 引脚输出
1: 允许 TOPn0 引脚输出

禁止TOPn0引脚操作时
输出电平的设置
0: 低电平
1: 高电平

0: 禁止 TOPn1 引脚输出
1: 允许 TOPn1 引脚输出

禁止TOPn1引脚操作时
输出电平的设置
0: 低电平
1: 高电平

（d）TMPn 输入/输出控制寄存器 2（TPnIOC2）

					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0/1注	0/1注	0	0

这些位用于选择
外部事件计数输入
(TIPn0引脚)的有效沿。

注 使用定时器输出（TOPn1）时，TPnEES1 和 TPnEES0 位只能被设置为 1。请注意，TPnCCR0 和 TPnCCR1 寄存器必须要设置为相同的值（也就是说，与已经为这些寄存器设置的值相同）。

（e）TMPn 计数器读取缓冲寄存器（TPnCNT）

通过读取该寄存器，可以读取 16 计数器的当前值。

（f）TMPn 捕获/比较寄存器 0（TPnCCR0）

如果 TPnCCR0 寄存器的值被设置为 D_0，则时间间隔计算如下：

间隔时间＝（D_0＋1）× 计数时钟周期

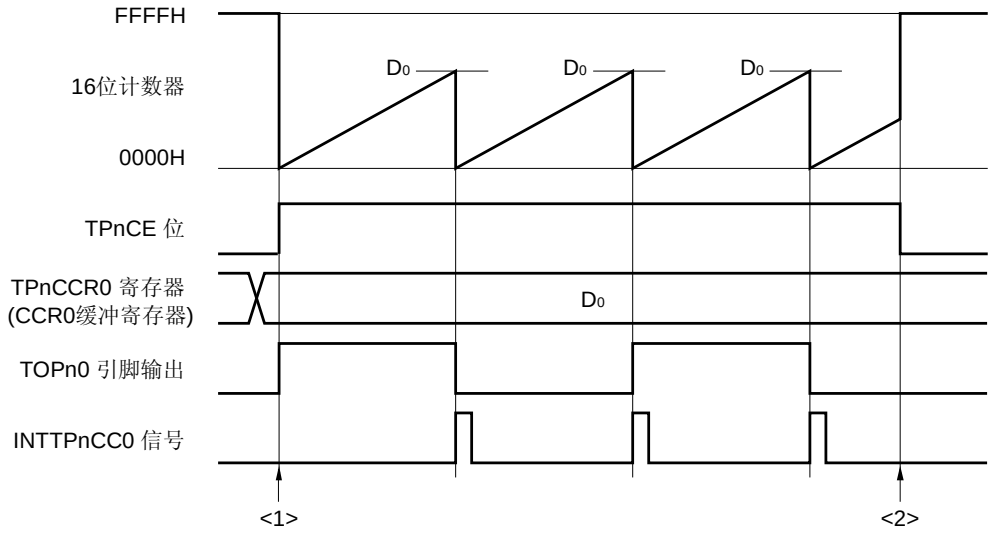
（g）TMPn 捕获/比较寄存器 1（TPnCCR1）

通常在间隔定时器模式下不使用 TPnCCR1 寄存器。但是因为 TPnCCR1 寄存器的设定值被传送给 CCR1 缓冲寄存器，当 16 位计数器的计数值和 CCR1 缓冲寄存器的值匹配时，会产生比较匹配中断请求信号（INTTPnCC1）。设置相应的中断屏蔽标志（TPnCCMK1）来屏蔽来自该寄存器的中断请求。

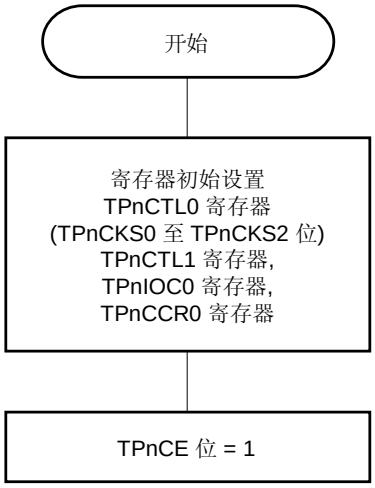
备注 在间隔定时器模式下，不使用 TMPn 输入/输出控制寄存器 1（TPnIOC1）和 TMPn 选项寄存器 0（TPnOPT0）。

（1）间隔定时器模式操作流程

图 7-9. 间隔定时器模式下的时序和操作处理



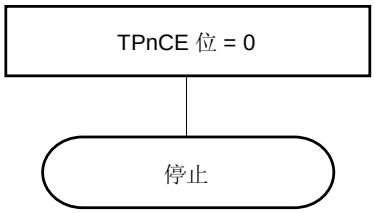
<1> 计数操作开始流程



在TPnCE位设置为1之前，
对这些寄存器执行初始化。

计数开始。
可以设置
TPnCKS0至TPnCKS2位。

<2> 计数操作停止流程



禁止计数 (TPnCE 位 = 0)
计数器被重置，计数停止。

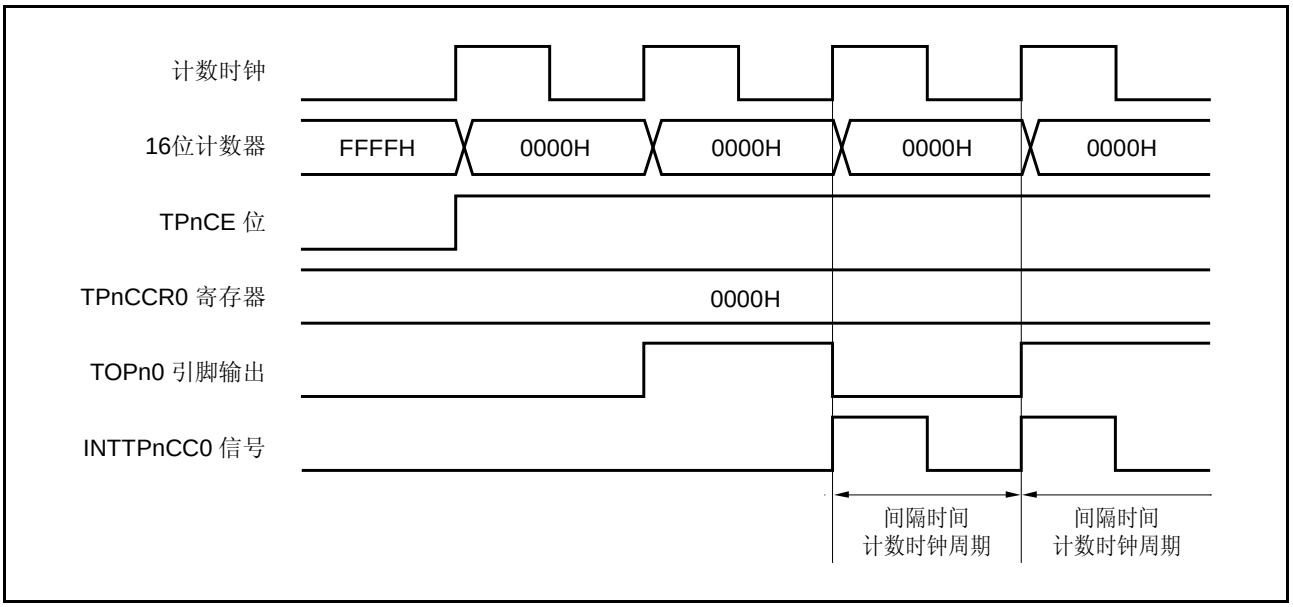
（2）使用间隔定时器模式

（a）当 TPnCCR0 寄存器清为 0000H 时的操作

如果 TPnCCR0 寄存器被清为 0000H，第二个时钟及其后的每个时钟计数时都会产生 INTTPnCC0 信号，而且，TOPn0 引脚的输出反相。

16 位计数器的值总是 0000H。

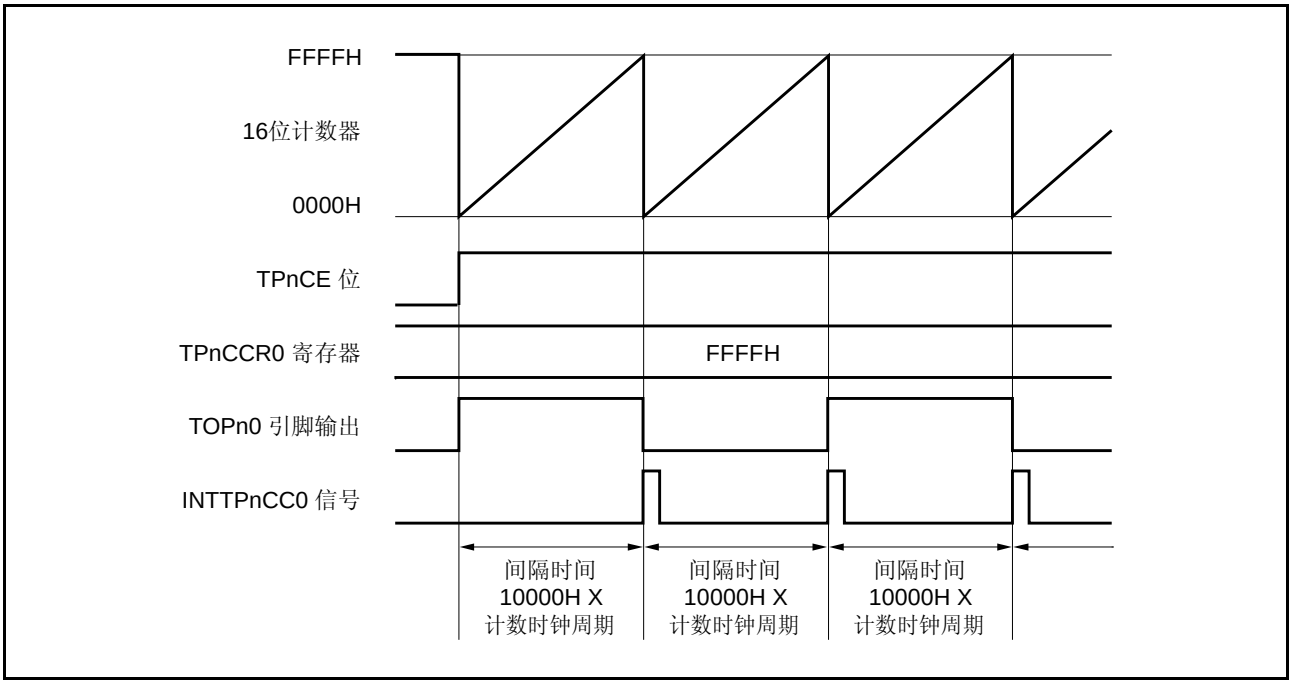
图 7-10. 当 TPnCCR0 寄存器清为 0000H 时间隔定时器的操作



（b）当 TPnCCR0 寄存器被置为 FFFFH 时的操作

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

图 7-11. 当 TPnCCR0 寄存器置为 FFFFH 时间隔定时器的操作

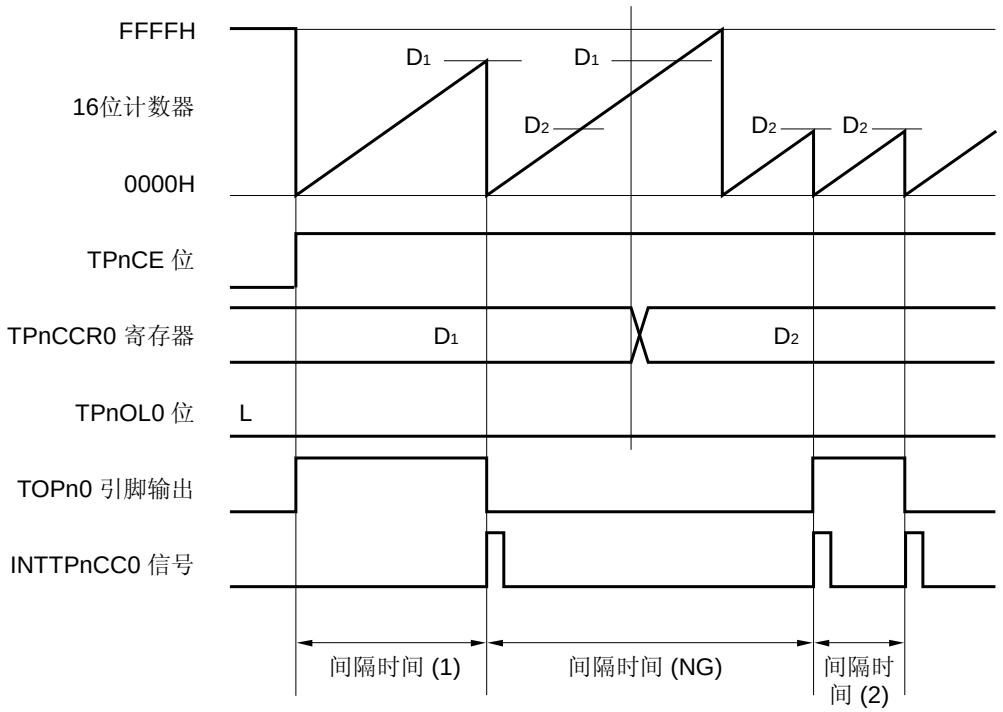


（c）重新写入 TPnCCR0 寄存器的注意事项

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

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

图 7-12. 重写 TPnCCR0 寄存器



备注 间隔时间（1）：（D_1＋1）× 计数时钟周期

间隔时间（NG）：（10000H＋D_2＋1）× 计数时钟周期

间隔时间（2）：（D_2＋1）× 计数时钟周期

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

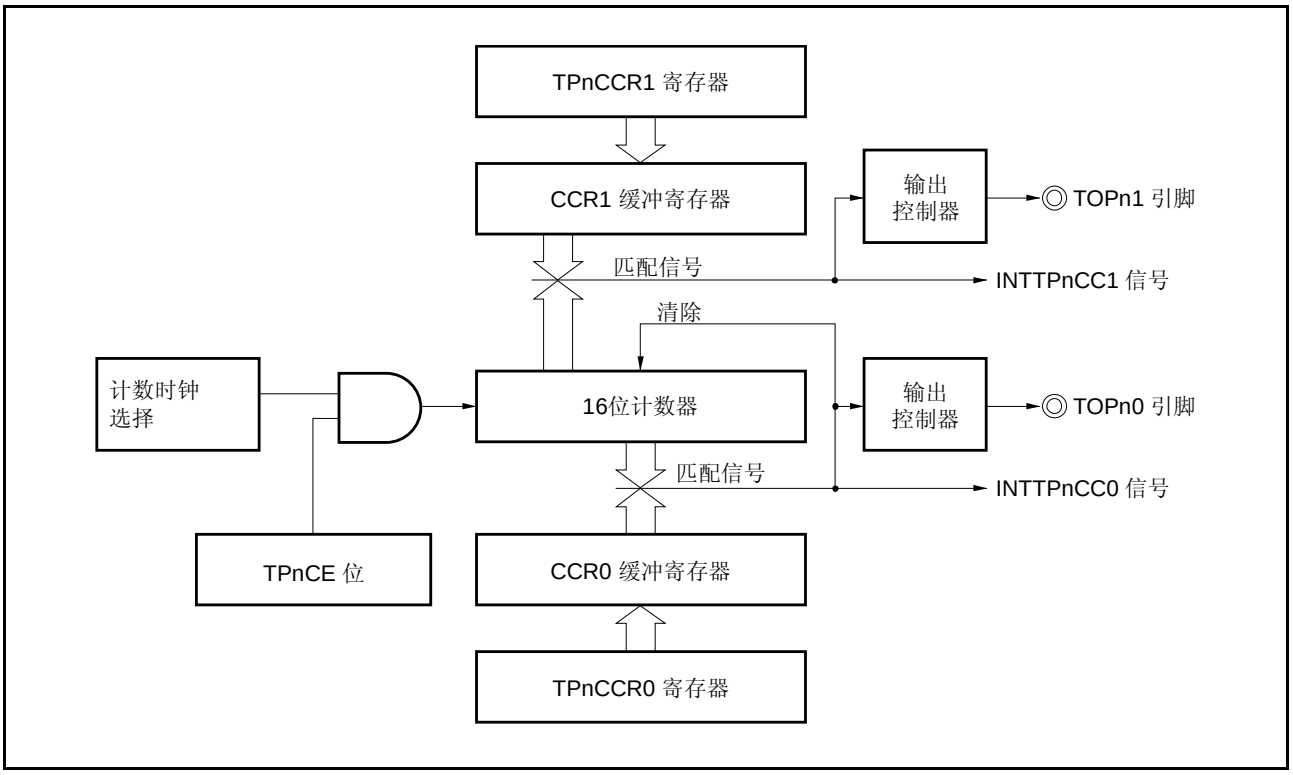
但是，由于计数值已经超过了 D_2，所以 16 位计数器就会向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值和 D_2 相符时才产生 INTTPnCC0 信号，且 TOPn0 引脚的输出反相。

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

（d）TPnCCR1 寄存器的操作

间隔定时器模式下对 TPnCCR1 寄存器的配置如下所示。

图 7-13. TPnCCR1 寄存器的配置

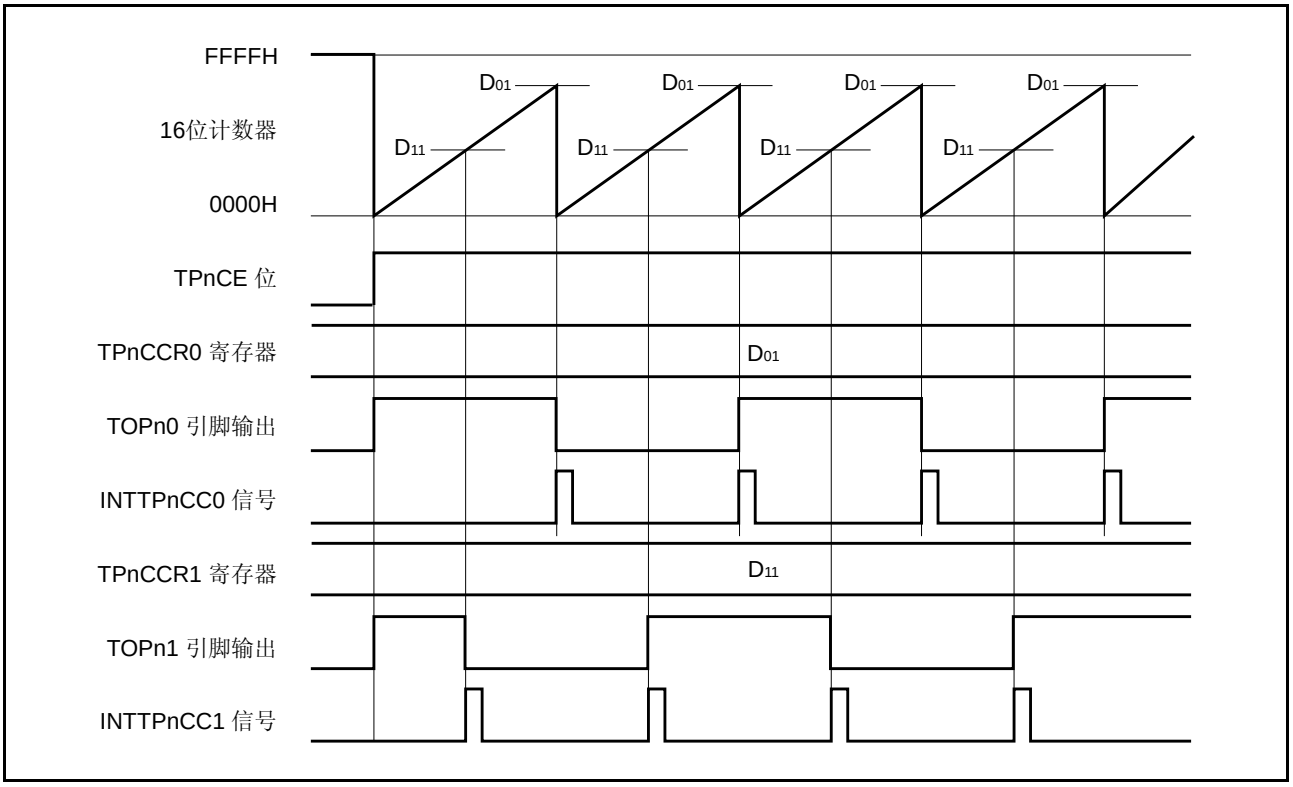


如果 TPnCCR1 寄存器的值小于或等于 TPnCCR0 寄存器的值，则每个计数周期产生一次 INTTPnCC1 中断请求信号。同时，TOPn1 引脚的输出被反转。

TOPn1 引脚输出一个方波，其周期与 TOPn0 引脚的输出周期相同，但是相位不同。

当 TPnCCR1 寄存器的值(D_{11})小于或等于 TPnCCR0 寄存器的值(D_{01})时，操作时序图显示如下。

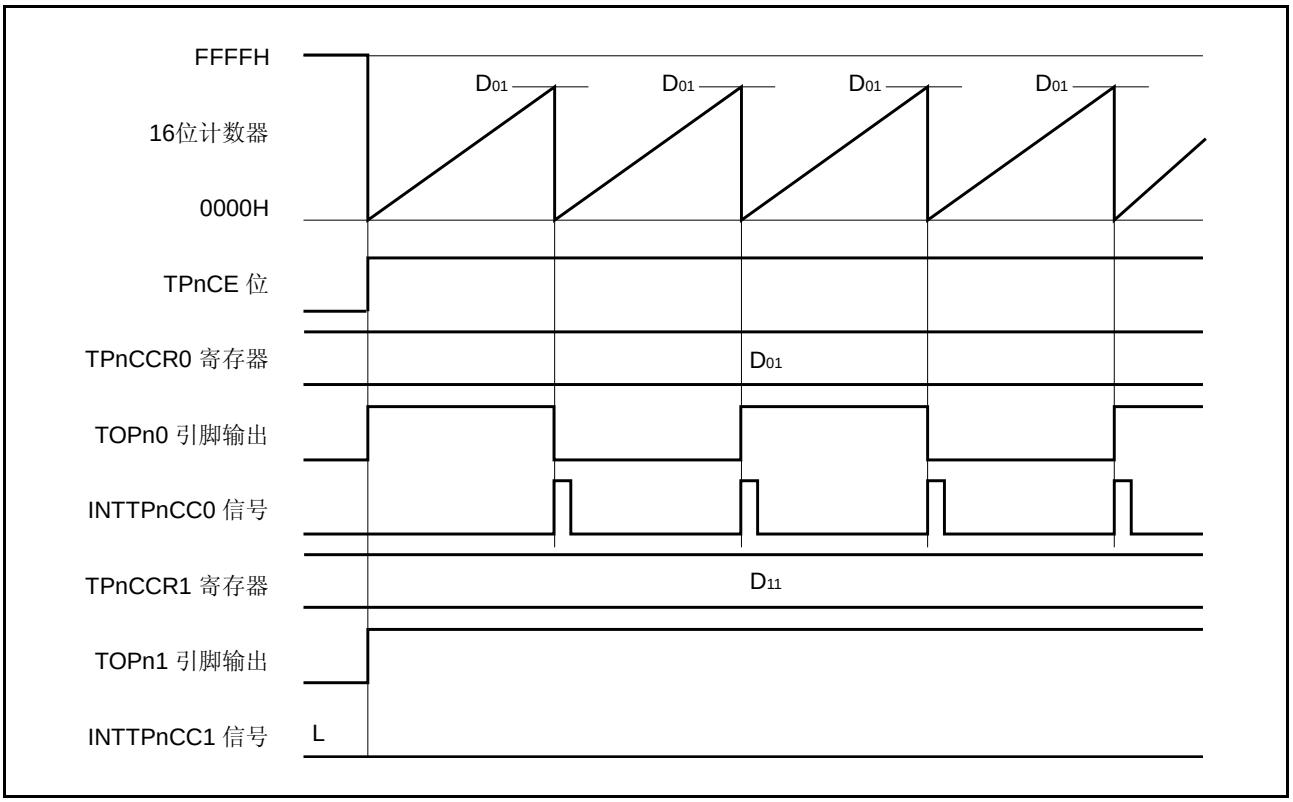
图 7-14. 当 $D_{01} \geq D_{11}$ 时的操作时序



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

当 TPnCCR1 寄存器的值(D_{11})大于 TPnCCR0 寄存器的值(D_{01})时，操作时序图显示如下。

图 7-15. 当 D_{01} < D_{11} 时的操作时序



（3）基于外部事件计数的间隔定时器操作

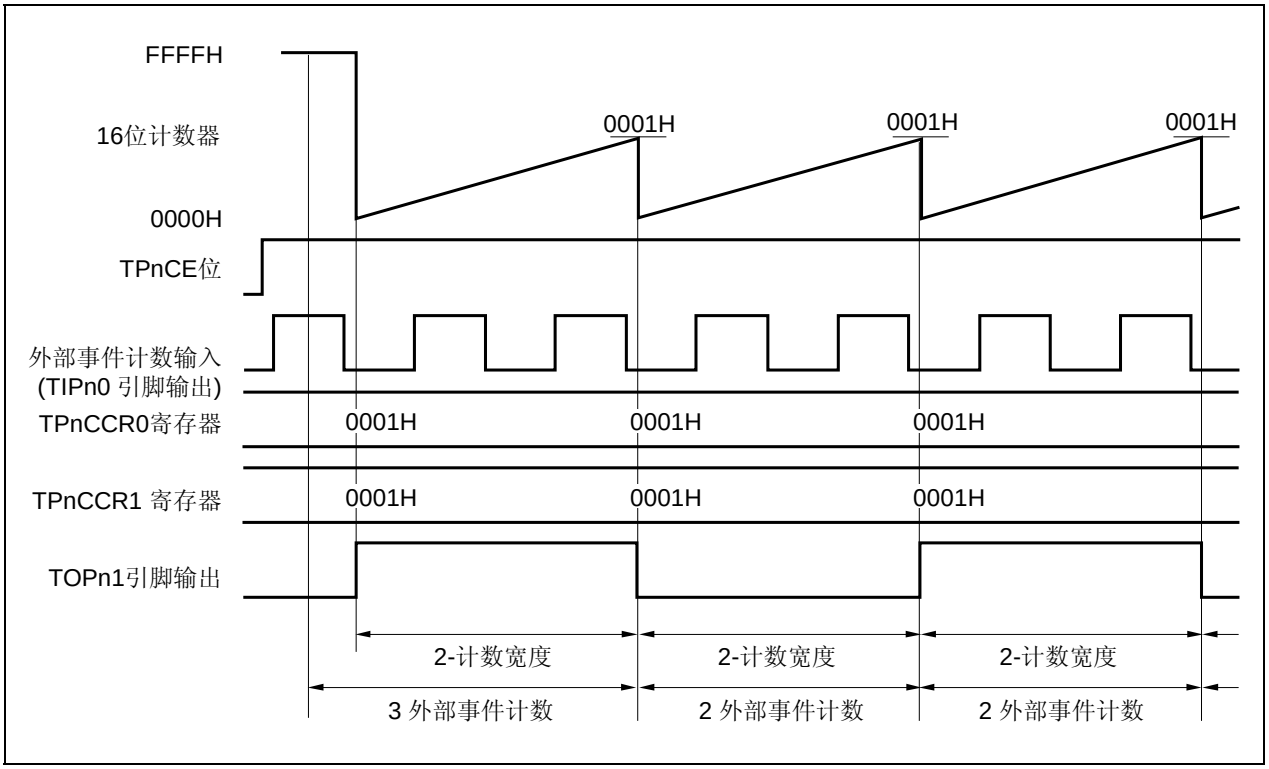
(a) 操作

在间隔定时器模式下，当 16 位计数器基于外部事件计数输入(TIPn0 引脚)的有效沿递增时，16 位计数器从 FFFFH 被清除为 0000H 之后，紧随 TpnCE 位从 0 改变为 1 启动计数器递增之后，必须有外部事件计数有效边沿立即输入。

一旦 TPnCCR0 和 TPnCCR1 寄存器被设置为 0001H (也就是说，与之前设置的值相同)，16 位计数器的每计数两次，TOPn1 引脚输出都会反转。

请注意，当基于外部事件计数使用定时器输出(TOPn1)时，TPnCTL1.TPnEEE 位只能被设置为 1。

图 7-16. 基于外部事件计数输入(TIPn0)的间隔定时器操作



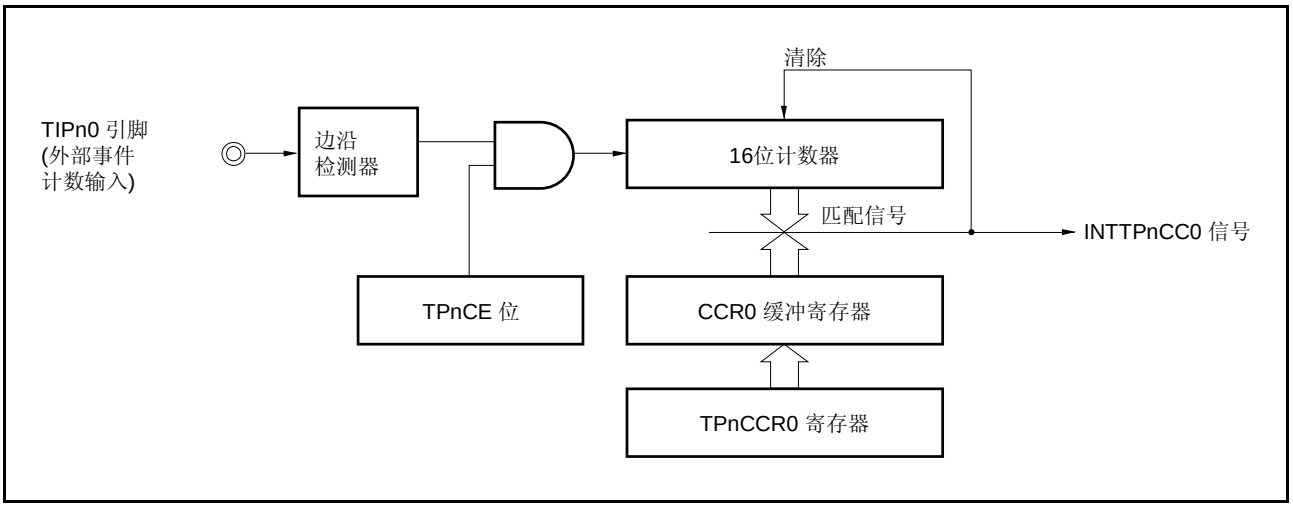
7.4.2 外部事件计数模式（TPnMD2 至TPnMD0 位＝001）

在外部事件计数器模式下，当 TPnCTL0.TPnCE 位被置为 1 时，对外部事件计数输入信号的有效边沿进行计数，每次有效边沿的计数数量达到指定数目时，产生一个中断请求信号（INTTPnCC0）。不能使用定时器输出引脚（TOPn0 和 TOPn1）。要在外部事件计数模式下使用 TOPn1 引脚，需要先在间隔定时器模式下将 TPnCTL1.TPnEEE 位设置为 1（详情参见 **7.4.1 (3) 基于外部事件计数输入的间隔定时器操作**）。

通常在外部事件计数器模式下，不使用 TPnCCR1 寄存器。

备注 **1.** 关于如何设置 TOPn0 引脚，参见**表 7-2 TMPn 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。
2. 关于如何使能 INTTPnCC0 中断信号，参见**第 22 章 中断服务/异常处理。**

图 7-17. 外部事件计数模式下的间隔定时器配置

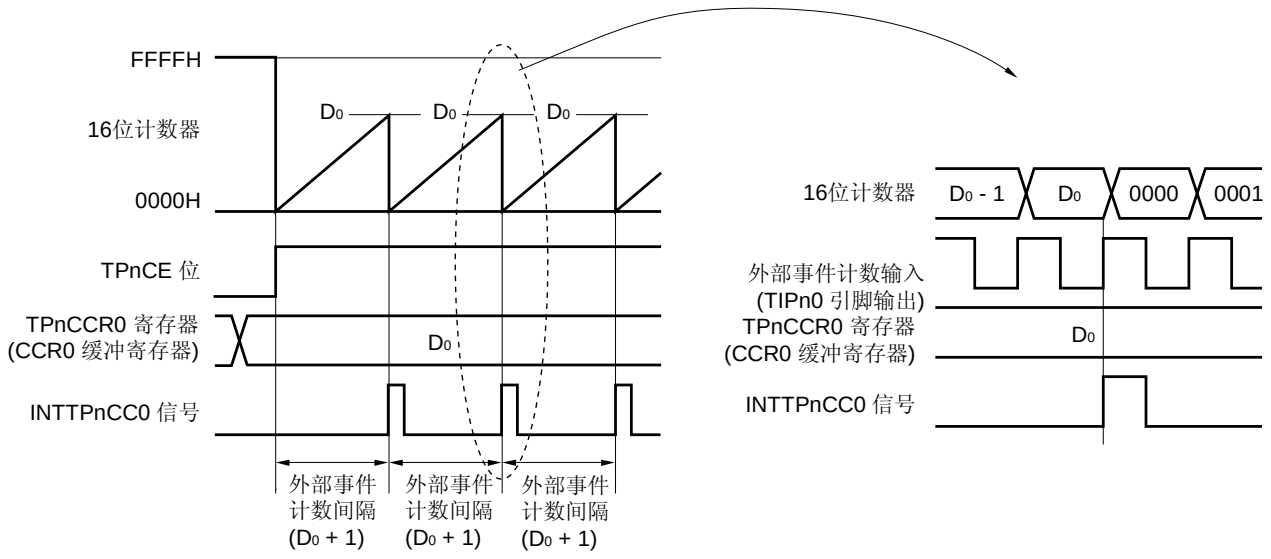


当 TPnCE 位被置为 1 时，16 位计数器的值从 FFFFH 被清为 0000H。计数器对检测到的每一次外部事件计数输入的有效边沿进行计数。此外，TPnCCR0 寄存器的设定值被传送给 CCR0 缓冲寄存器。

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

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

图 7-18. 外部事件计数模式下的基本操作时序



备注 本图显示了当上升沿被指定作为外部事件计数输入有效边沿时的基本时序。

外部事件计数模式下的寄存器操作设置示例如下图所示。

图 7-19. 外部事件计数模式下的寄存器设置（1/2）

（a）TMPn 控制寄存器 0（TPnCTL0）

	TPnCE					TPnCKS2	TPnCKS1	TPnCKS0
TPnCTL0	0/1	0	0	0	0	0	0	0

0: 停止计数
1: 允许计数

（b）TMPn 控制寄存器 1（TPnCTL1）

		TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0	0	0	0	0	0	1

0, 0, 1:
外部事件计数模式

（c）TMPn 输入/输出控制寄存器 0（TPnIOC0）

					TPnOL1	TPnOE1	TPnOL0	TPnOE0
TPnIOC0	0	0	0	0	0	0	0	0

0: 禁止 TOPn0 引脚输出
0: 禁止 TOPn1 引脚输出

（d）TMPn 输入/输出控制寄存器 2（TPnIOC2）

					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0/1	0/1	0	0

选择外部事件计数
输入的有效沿

图 7-11. 外部事件计数模式下的寄存器操作设置（2/2）

（e）TMPn 计数器读取缓冲寄存器（TPnCNT）

可以通过读取 TPnCNT 寄存器，读取 16 位计数器的计数值。

（f）TMPn 捕获/比较寄存器 0（TPnCCR0）

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

（g）TMPn 捕获/比较寄存器 1（TPnCCR1）

通常在外部事件计数器模式中不使用 TPnCCR1 寄存器。但是，因为当 16 位计数器的计数值和 CCR1 缓冲寄存器的值匹配时，TPnCCR1 寄存器的设定传送给 CCR1 缓冲寄存器，并会产生一个比较匹配中断请求信号（INTTPnCC1）。通过设置中断屏蔽标志（TPnCCMK1）可以屏蔽来自该寄存器的中断。

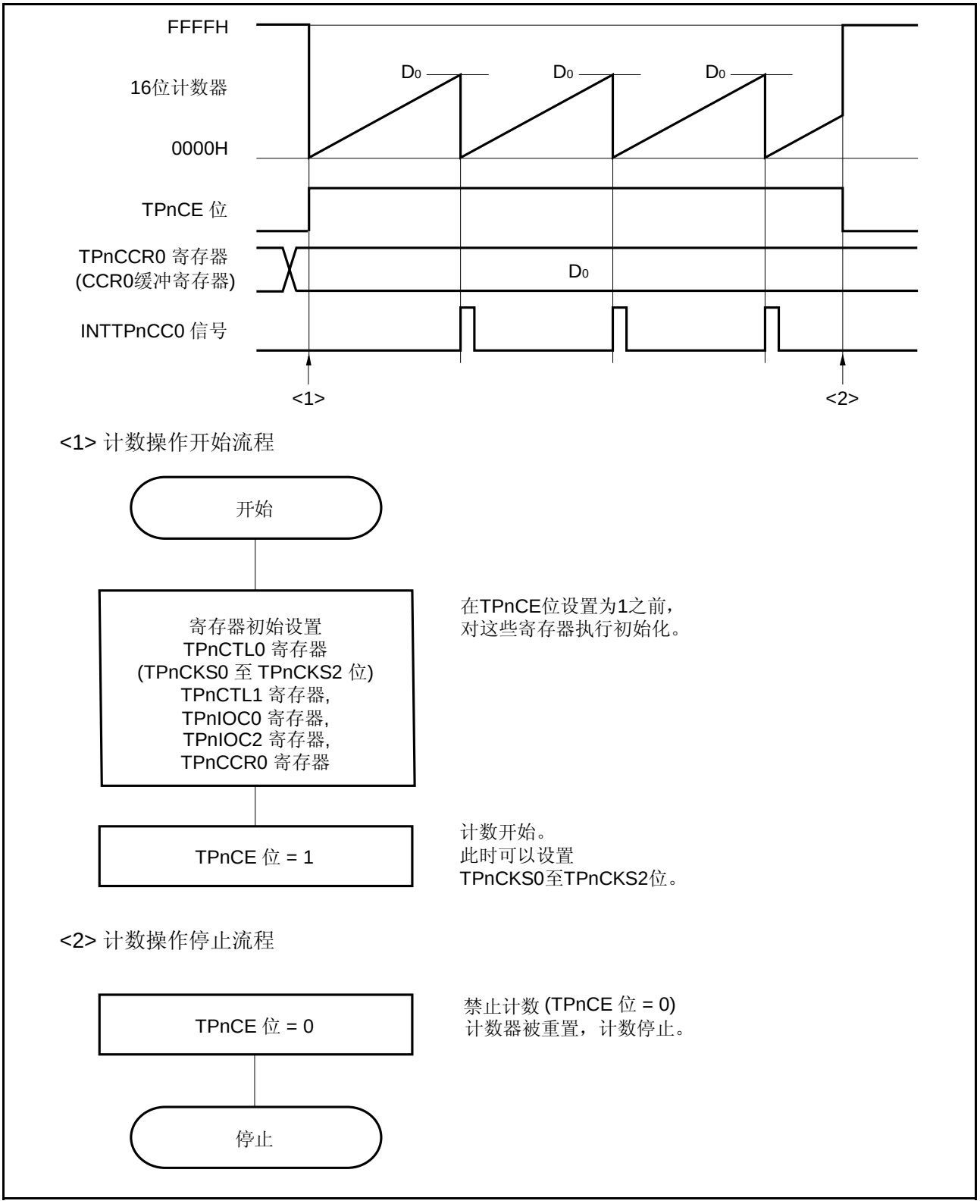
注意事项 1. 外部事件计数模式下，不要将 TPnCCR0 寄存器设置为 0000H。

2. 外部事件计数模式下，不能使用定时器输出。如果基于外部事件计数输入来进行定时器输出，首先应设置为间隔定时器模式，然后指定外部事件计数输入“操作使能”（TPnCTL1.TPnMD2 至 TPnMD0 位为 0，0，0，并将 TPnCTL1.TPnEEE 位设置为 1）。详情参见 7.4.1 (3) 基于外部事件计数输入的间隔定时器操作。

备注 在外部事件计数模式下，不使用 TMPn 输入/输出控制寄存器 1（TPnIOC1） 和 TMPn 选项寄存器 0（TPnOPT0）。

（1）外部事件计数模式的操作流程

图 7-20. 外部事件计数模式下的时序和操作处理

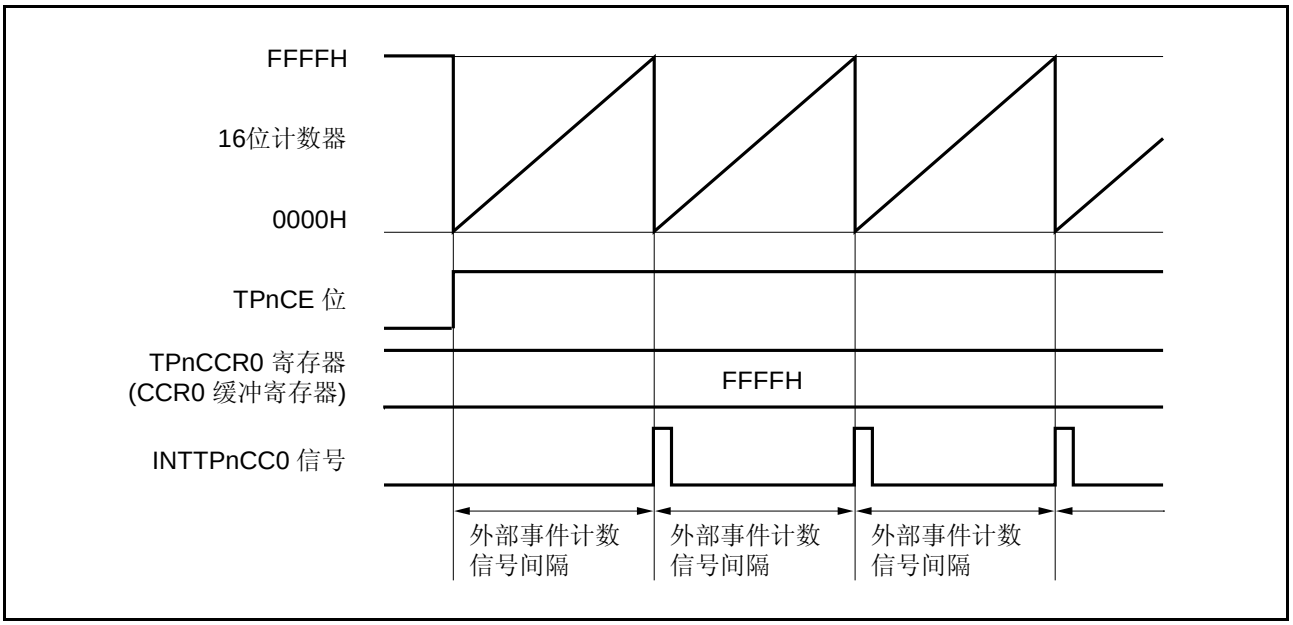


（2）使用外部事件计数模式

（a）当 TPnCCR0 寄存器被设置为 FFFFH 时的操作

如果 TPnCCR0 寄存器被设置为 FFFFH，16 位计数器每次检测到的外部事件输入信号有效边沿时，都递增计数到 FFFFH。在下一个递增计数时，16 位计数器被同步清为 0000H，并且产生 INTTPnCC0 信号。此时，不设置 TPnOPT0.TPnOVF 位。

图 7-21. 当 TPnCCR0 寄存器被设置为 FFFFH 时的操作

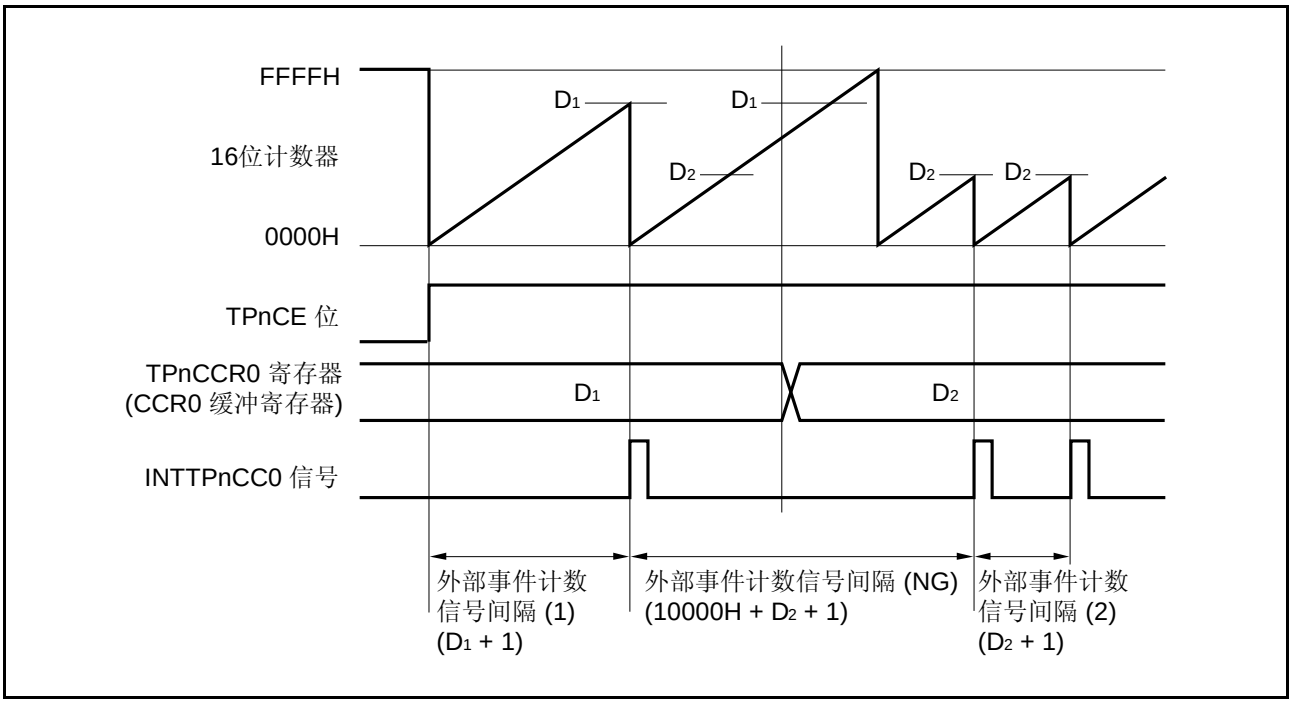


（b）重写 TPnCCR0 寄存器的注意事项

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

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

图 7-22. 重写 TPnCCR0 寄存器



当 16 位计数器计数值大于 D_2 而小于 D_1 时，如果将 TPnCCR0 寄存器的值从 D_1 变为 D_2，则在重写 TPnCCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。因而，与 16 位计数器相比较的值会是 D_2。

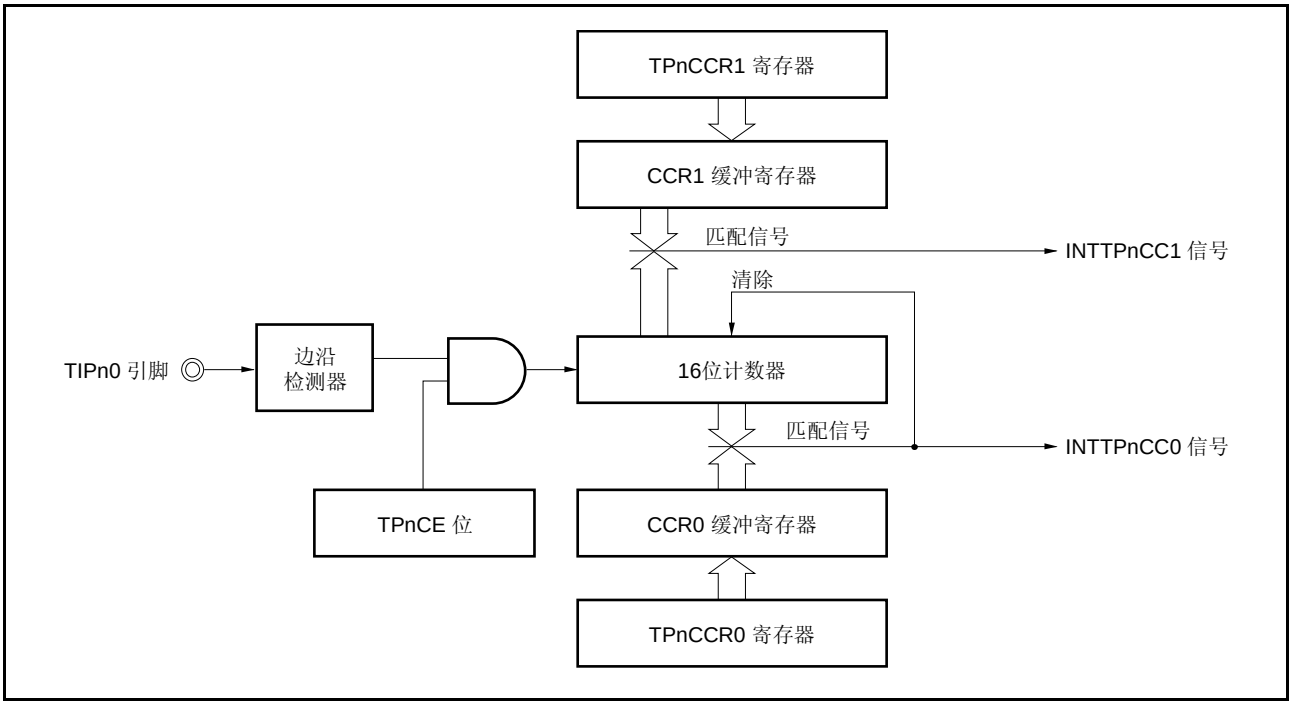
但是，由于计数值已经超过了 D_2，所以 16 位计数器就会向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值与 D_2 匹配时，会生成 INTTPnCC0 信号。

因此，可能不会在原先预计的“（D_1＋1）次”或“（D_2＋1）次”有效边沿生成 INTTPnCC0 中断请求信号，而可能在外部事件计数信号的“（10000H＋D_2＋1）次”有效边沿处生成中断请求信号。

（c）TPnCCR1 寄存器的操作

外部事件计数模式下的 TPnCCR1 寄存器配置如下图所示。

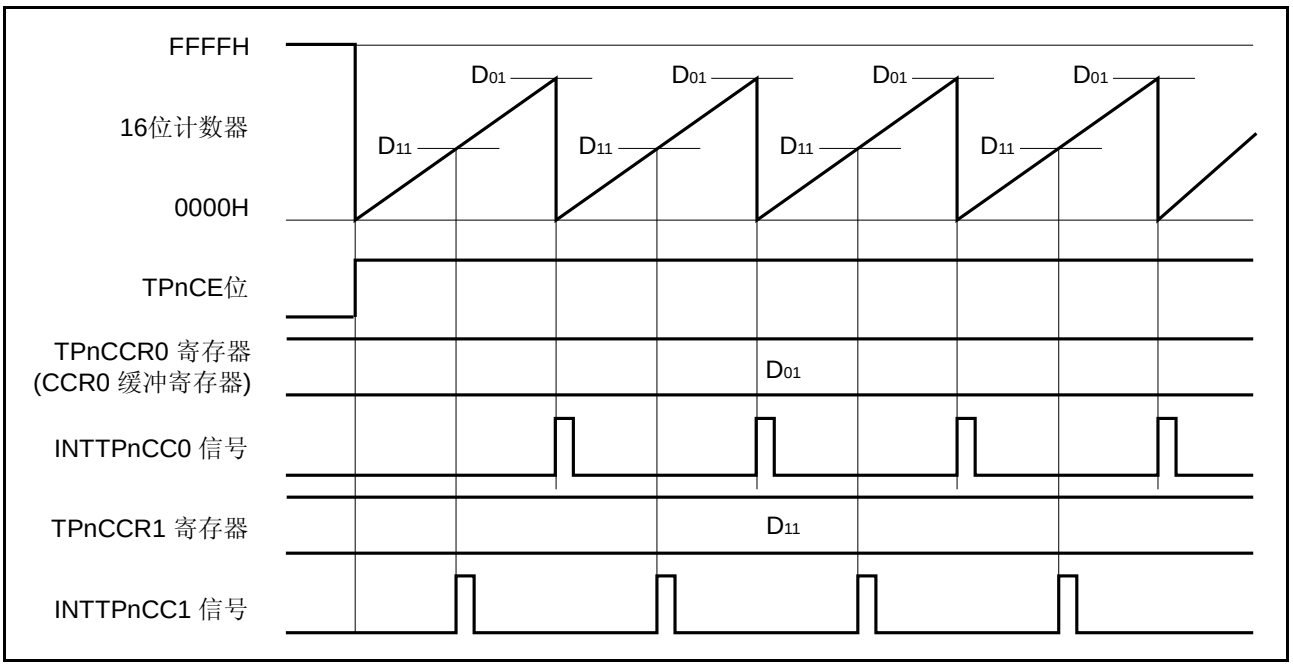
图 7-23. TPnCCR1 寄存器的配置



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

当 TPnCCR1 寄存器的值(D_{11})小于或等于 TPnCCR0 寄存器的值(D_{01})时，操作时序图显示如下。

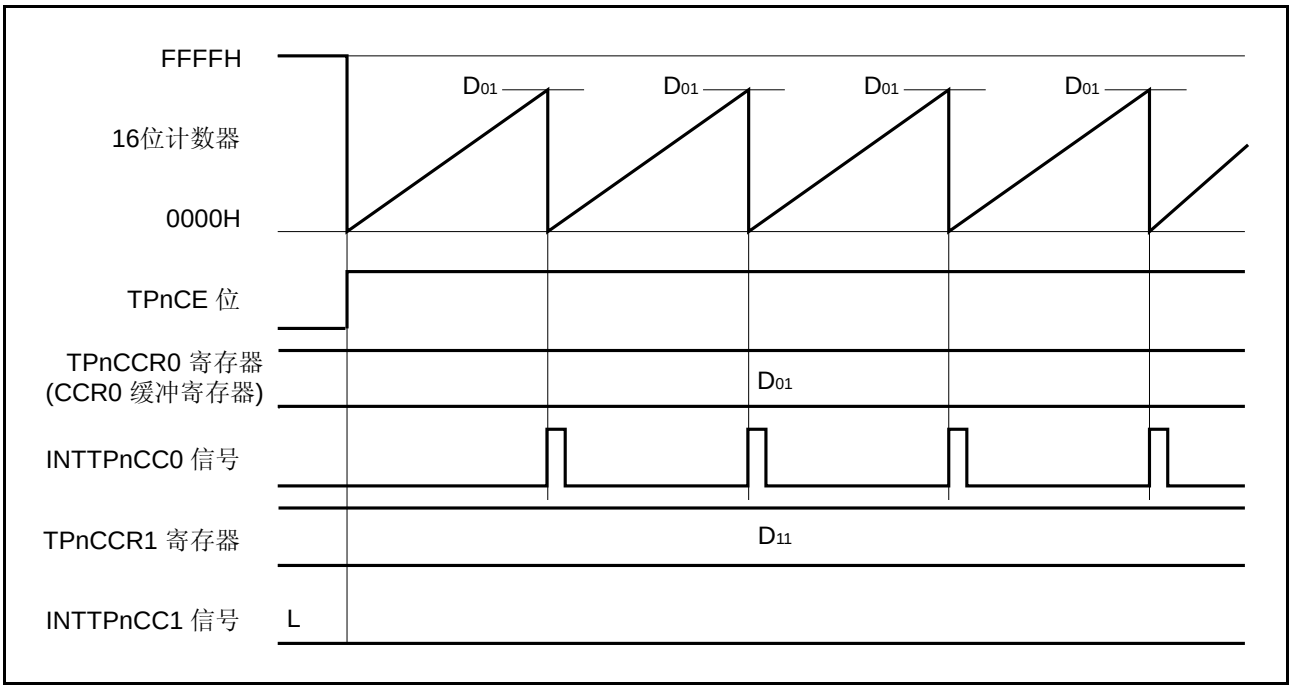
图 7-24. 当 $D_{01} \geq D_{11}$ 时的操作时序



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

当 TPnCCR1 寄存器的值(D_{11})大于 TPnCCR0 寄存器的值(D_{01})时，操作时序图显示如下。

图 7-25. 当 D_{01} < D_{11} 时的操作时序



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

在外部触发脉冲输出模式下，当 TPnCTL0.TPnCE 位被置为 1 时，TMPn 等待触发信号，当检测到外部触发输入信号的有效边沿后，TMPn 开始计数，并且从 TOPn1 引脚输出一个 PWM 波形。

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

备注 **1.** 关于如何设置 TIPn0, TOPn0 和 TOPn1 引脚，参见**表 7-2 TMPn 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。

2. 关于如何使能 INTTPnCC0 和 INTTPnCC0 中断信号，参见**第 22 章 中断服务/异常处理。**

图 7-26. 外部触发脉冲输出模式下 TMP 的配置

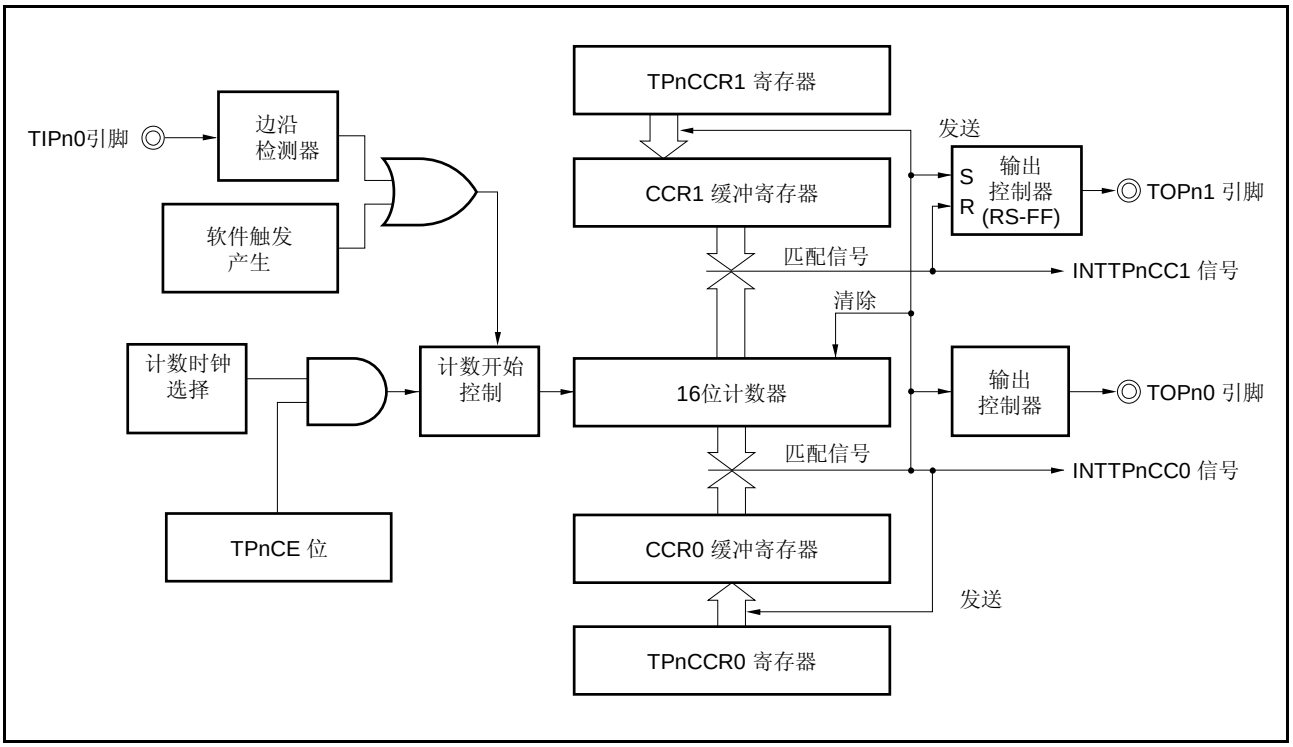
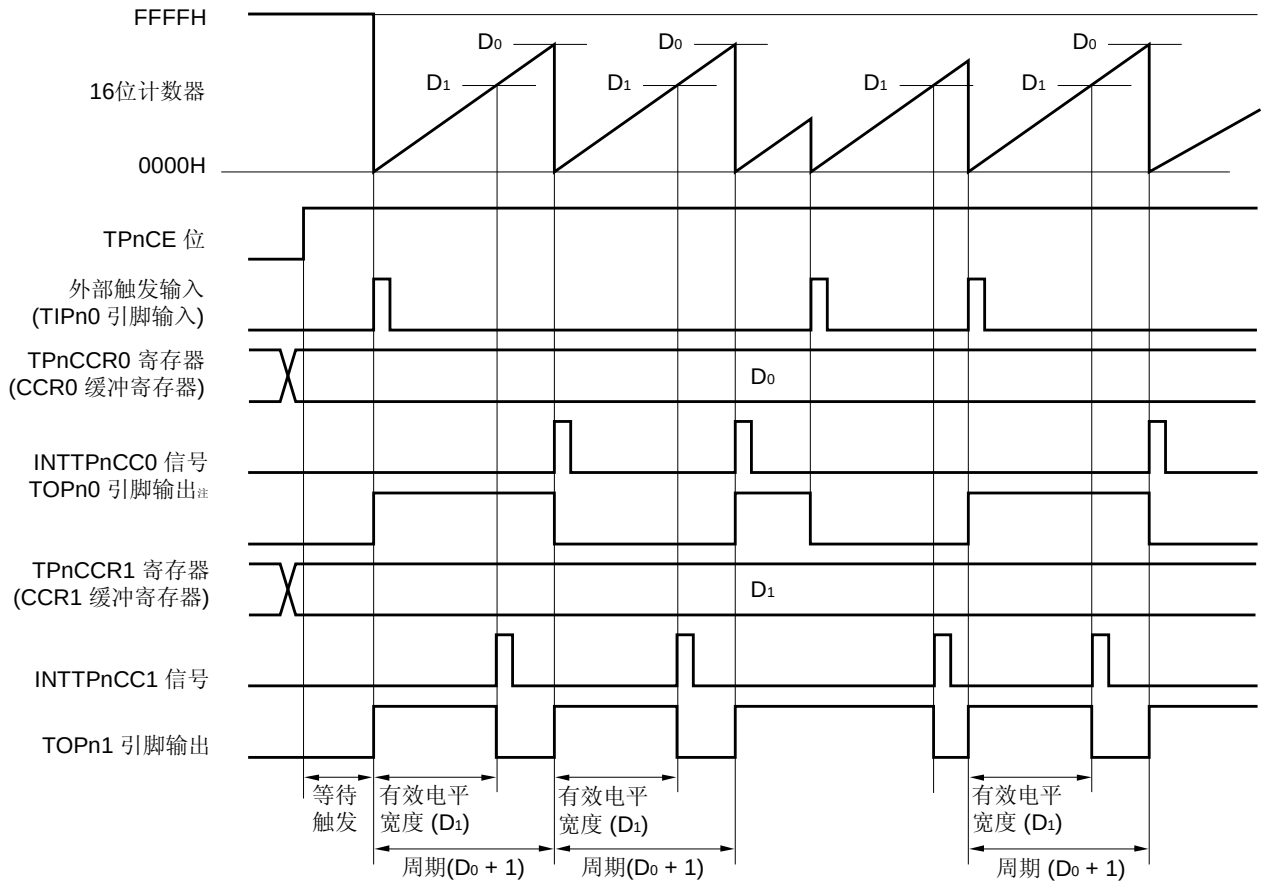


图 7-27. 外部触发脉冲输出模式的基本操作时序



注 TOPn0 引脚的输出也可以用作 TIPn0 引脚的输入。当使用 TOPn0 引脚的输出也可以用作 TIPn0 引脚的输入时，应使用软件触发，而不用外部触发。

当 TPnCE 位设置为 1 时，TMPn 等待触发。当触发产生时，16 位计数器从 FFFFH 清零为 0000H，同时开始递增计数，并从 TOPn1 引脚输出一个 PWM 波形。如果在计数运行中再次产生触发，则计数器清为 0000H 并重新开始计数，且 TOPn0 引脚输出反相。（在发生触发信号时，不管当前状态如何（高/低），TOPn1 引脚都会输出一个高电平。

PWM 波形的有效电平宽度、周期、占空比的计算如下。

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

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

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

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

图 7-28. 外部触发脉冲输出模式的寄存器设置（1/2）

（a）TMPn 控制寄存器 0（TPnCTL0）

	TPnCE					TPnCKS2	TPnCKS1	TPnCKS0
TPnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

选择计数时钟

0: 停止计数
1: 允许计数

（b）TMPn 控制寄存器 1（TPnCTL1）

		TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0/1	0	0	0	0	1	0

0, 1, 0:
外部触发脉冲
输出模式

写入1，产生一个
软件触发

（c）TMPn 输入/输出控制寄存器 0（TPnIOC0）

					TPnOL1	TPnOE1	TPnOL0	TPnOE0
TPnIOC0	0	0	0	0	0/1	0/1	0/1注	0/1注

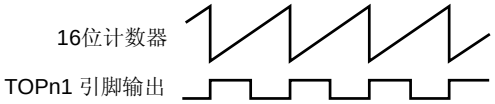
0: 禁止 TOPn0 引脚输出
1: 允许 TOPn0 引脚输出

禁止TOPn0引脚操作时
输出电平的设置
0: 低电平
1: 高电平

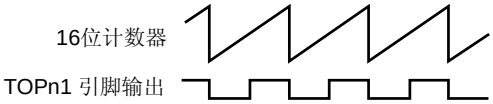
0: 禁止 TOPn1 引脚输出
1: 允许 TOPn1 引脚输出

指定TOPn1引脚输出的
有效电平
0: 高电平有效
1: 低电平有效

· 当 TPnOL1 位 = 0时



· 当 TPnOL1 位 = 1时



注 在外部触发脉冲输出模式下，不使用 TOPn0 引脚时，将该位清除为 0。

图 7-28. 外部触发脉冲输出模式的寄存器设置（2/2）

（d）TMPn 输入/输出控制寄存器 2（TPnIOC2）

					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0	0	0/1	0/1

这些位选择外部触发输入的有效沿

（e）TMPn 计数器读取缓冲寄存器（TPnCNT）

通过读取 TPnCNT 寄存器，可以读出 16 位计数器的值。

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

如果将 TPnCCR0 寄存器的值设置为 D_0，将 TPnCCR1 寄存器的值设置为 D_1，则 PWM 波形的周期和有效电平的计算如下：

PWM 波形周期 =（D_0＋1）× 计数时钟周期
PWM 波形有效电平宽度 = D_1 × 计数时钟周期

备注 在外部触发脉冲输出模式下，不使用 TMPn 输入/输出控制寄存器 1（TPnIOC1） 和 TMPn 选项寄存器 0（TPnOPT0）。

（1）外部触发脉冲输出模式的操作

图 7-29. 外部触发脉冲输出模式的时序和操作处理（1/2）

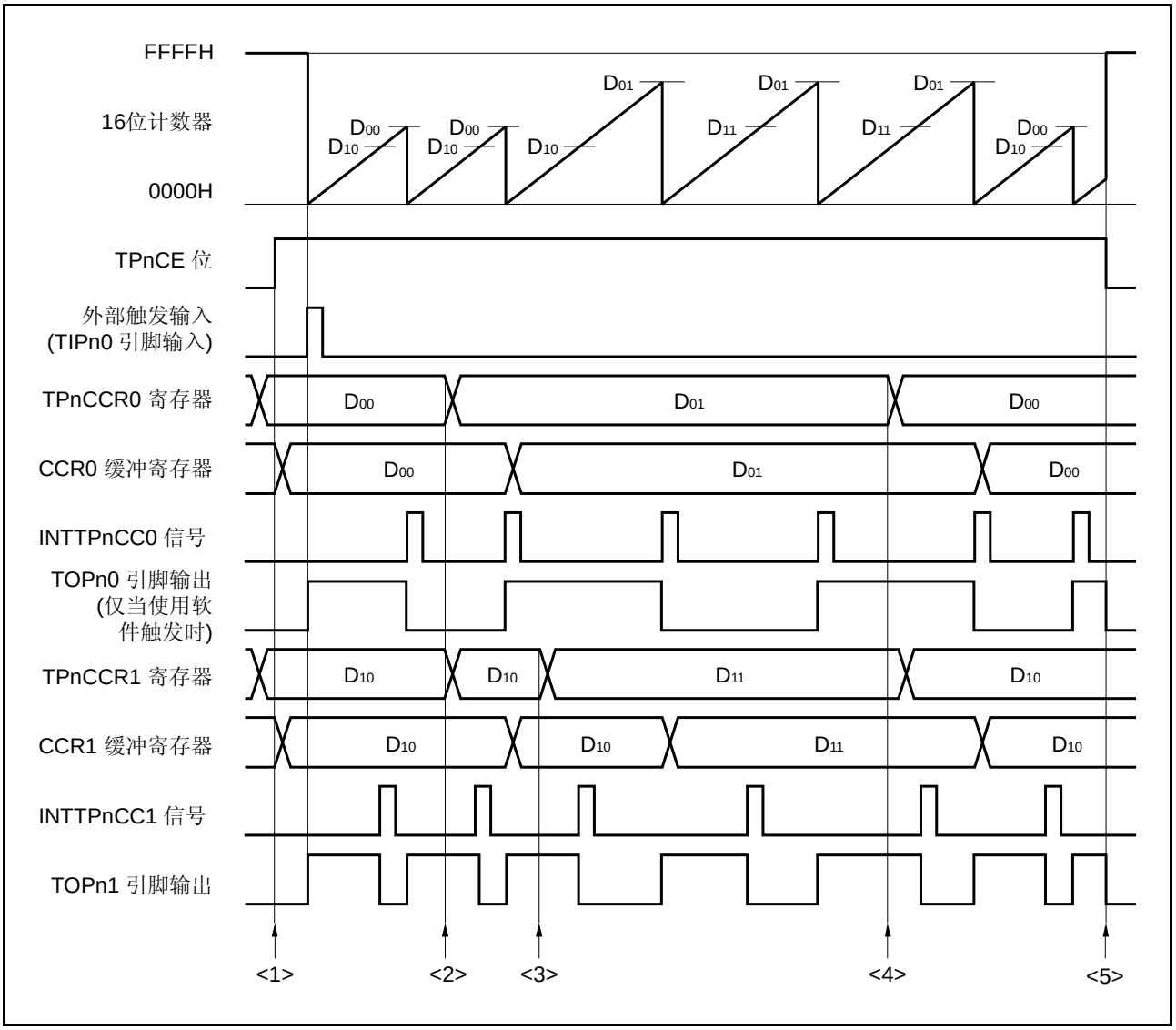
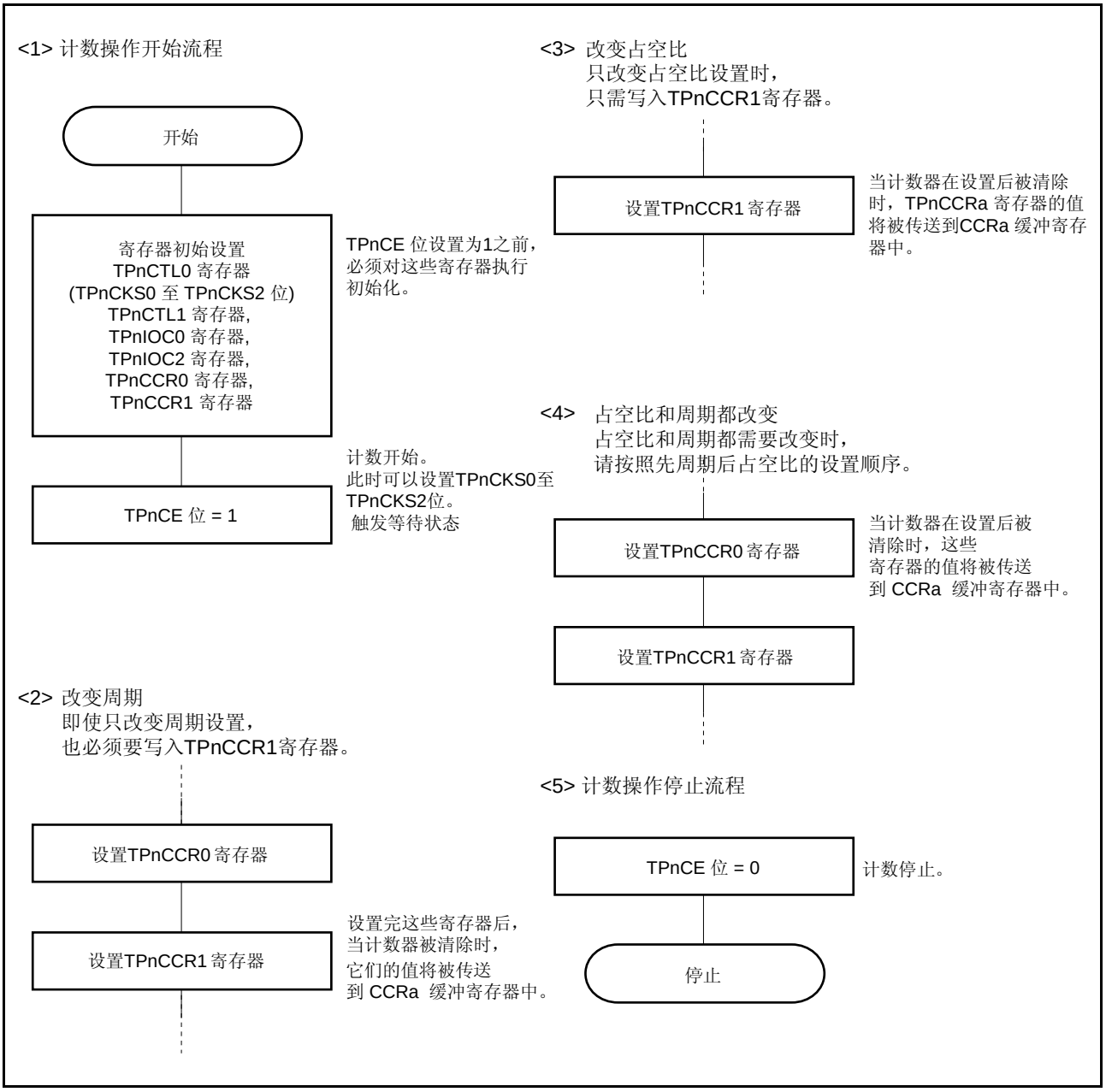


图 7-29. 外部触发脉冲输出模式的时序和操作处理（2/2）



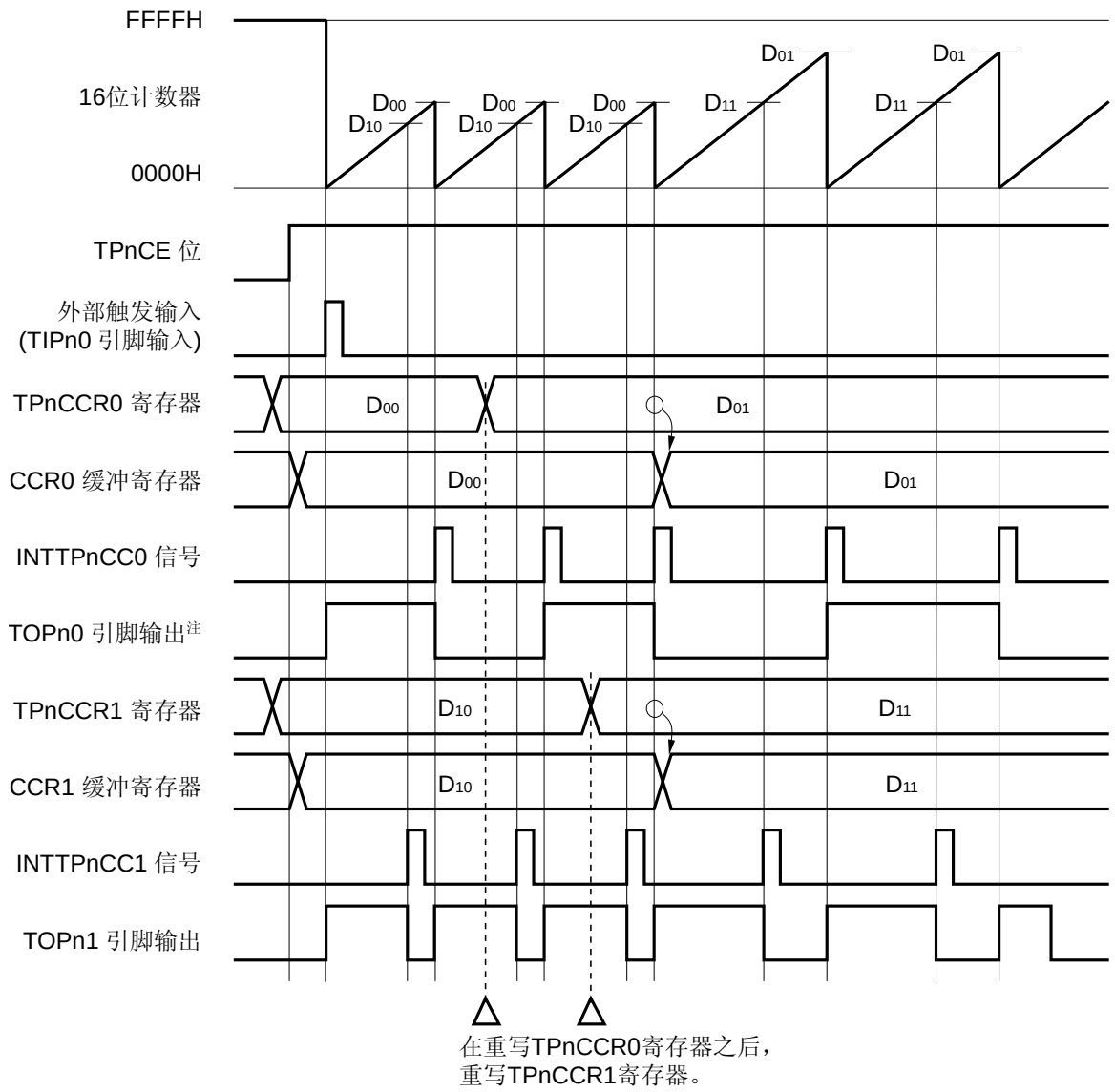
（2）使用外部触发脉冲输出模式

在外部触发脉冲输出模式下，如果改变 PWM 波形的描述如下。

（a）当计数器正在递增时改变 PWM 波形

计数器计数过程中，若要改变 PWM 波形，应在改变波形设置之后重新写入 TPnCCR1 寄存器。在检测到 INTTPnCC0 信号以后，在重写 TPnCCRa 寄存器之前要先写入 TPnCCR1 寄存器。

图 7-30. 计数器计数过程中改变 PWM 波形



注 TOPn0 引脚的输出也可以用作 TIPn0 引脚的输入。当使用 TOPn0 引脚的输出也可以用作 TIPn0 引脚的输入时，应使用软件触发，而不用外部触发。

为了将数据从 TPnCCRa 寄存器传送到 CCRa 缓冲寄存器，必须写入 TPnCCR1 寄存器。
当数据写入 TPnCCR1 寄存器之后，在对 16 位计数器清零的同时，写入到 TPnCCRa 寄存器的值被传送到 CCRa 缓冲寄存器，并作为与 16 位计数器进行比较的值。

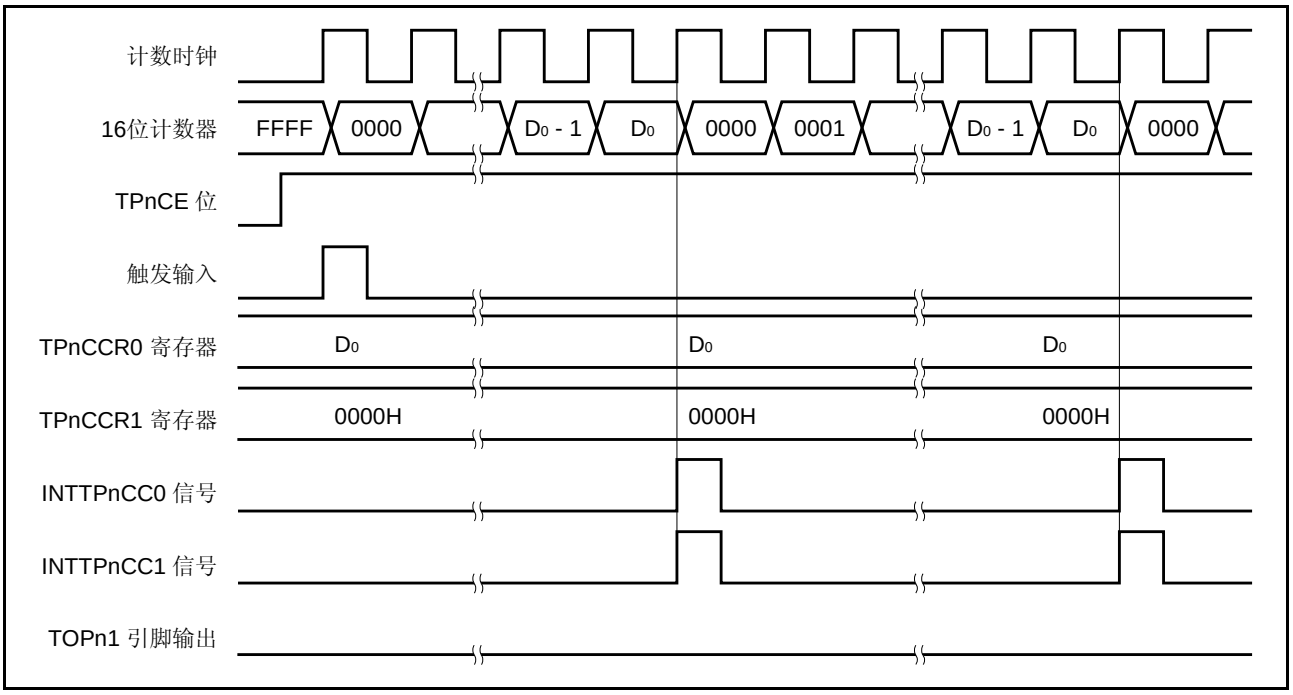
<1> 如果 PWM 波形的周期和有效电平宽度都要改变，应先将周期设定到 TPnCCR0 寄存器，然后将有效电平宽度设定到 TPnCCR1 寄存器。
<2> 如果只改变 PWM 波形的周期，应先把周期设定给 TPnCCR0 寄存器，然后将相同值（原值）写入 TPnCCR1 寄存器（也就是说，与预先为 TPnCCR1 寄存器指定的值相同）。
<3> 只改变 PWM 波形的有效电平宽度（占空比）时，仅需要设置 TPnCCR1 寄存器。

注意事项 在写入 TPnCCR1 寄存器之后，要重写 TPnCCR0 或 TPnCCR1 寄存器，应该在 INTTPnCC0 信号产生之后进行。否则，CCRa 缓冲寄存器的值可能不确定，因为把数据从 TPnCCRa 寄存器传送到 CCRa 缓冲寄存器的时序会与写入 TPnCCRa 寄存器的时序可能会发生冲突。

（b）PWM 波形的 0%和 100% 输出

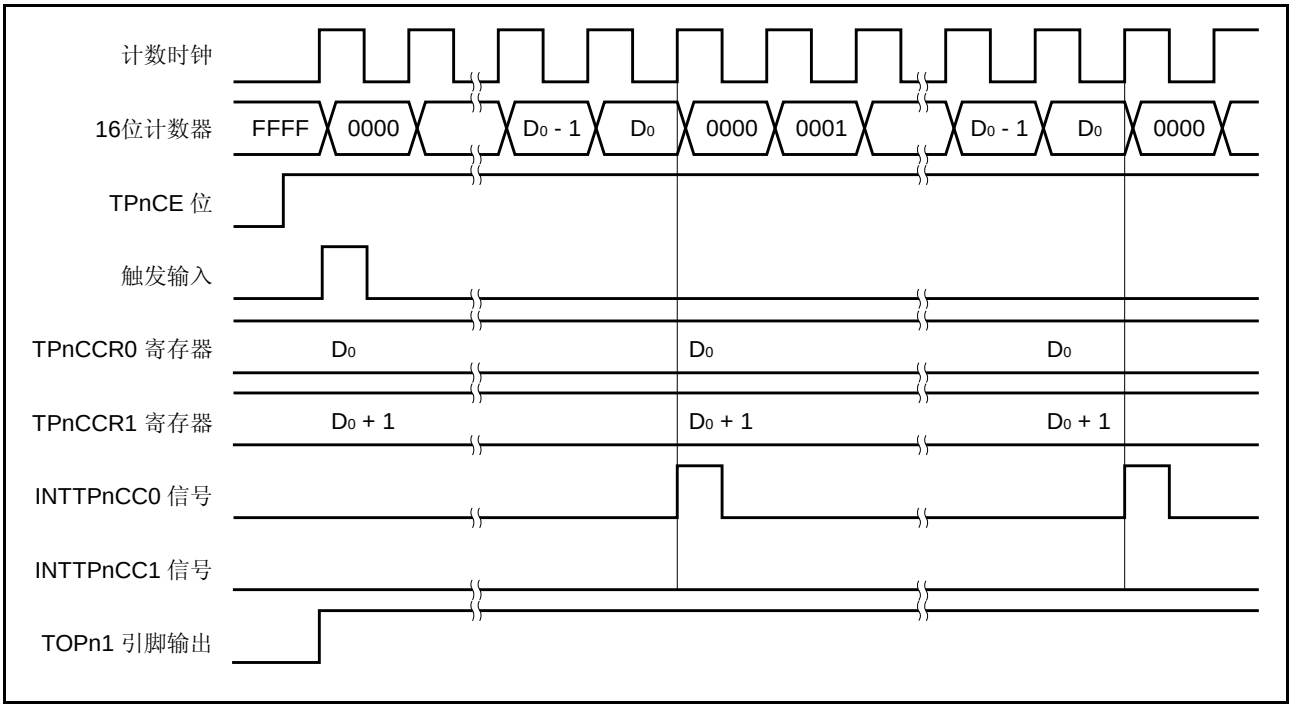
为了输出 0%的波形，需要将 TPnCCR1 寄存器清除为 0000H。如果 TPnCCR0 寄存器的设定值为 FFFFH，则会在每个周期产生 INTTPnCC1 中断请求信号。

图 7-31. 输出 0% PWM 波形



为了输出 100%的波形，需要将 TPnCCR1 寄存器的值设为：（TPnCCR0 寄存器的设定值＋1）。如果 TPnCCR0 寄存器的设定值为 FFFFH，则无法输出 100%波形。

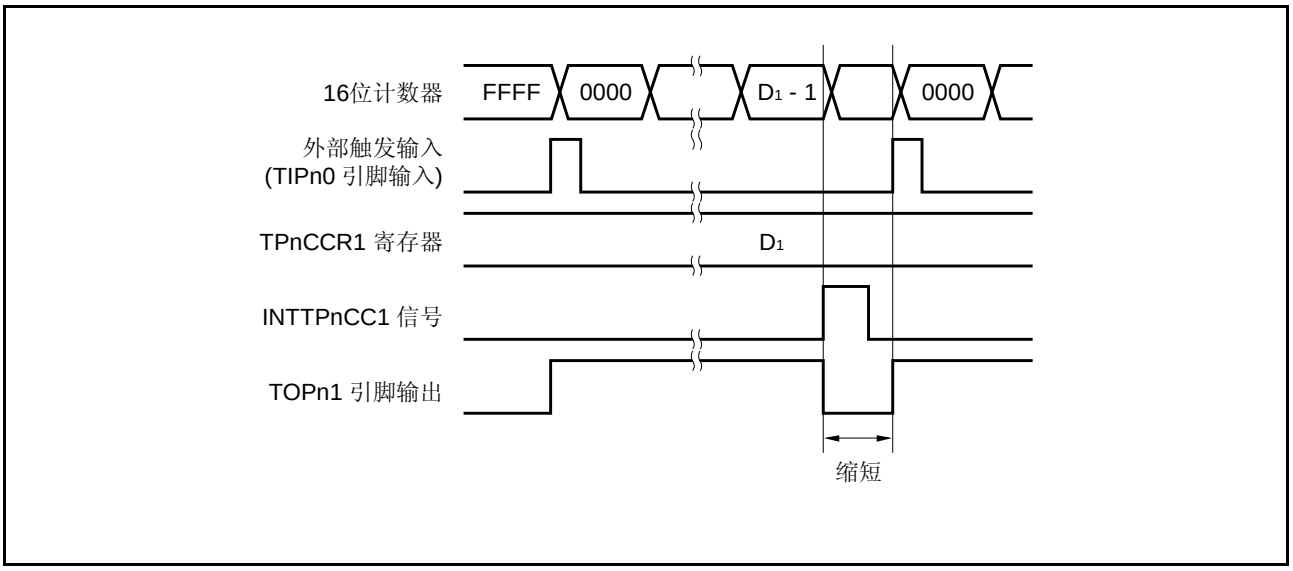
图 7-32. 输出 100% PWM 波形



（c）检测到触发信号紧邻 INTTPnCC1 产生之前或之后

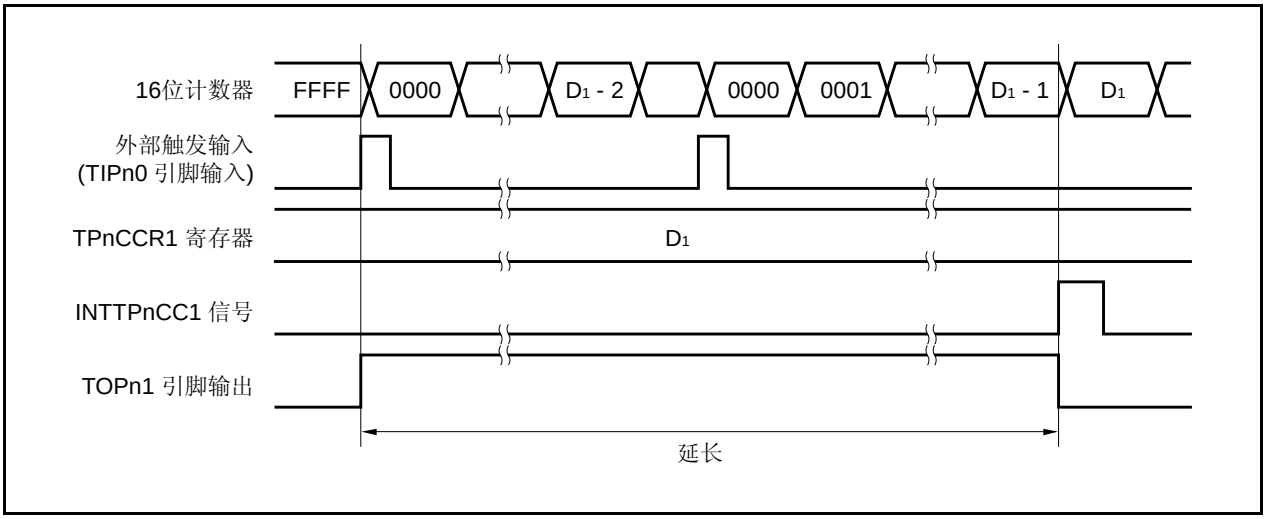
如果在 INTTPnCC1 信号产生后，立刻检测到触发信号，则 16 位计数器立刻清零为 0000H，TOPn1 引脚的输出信号被设置为有效电平，计数器继续递增计数。结果，PWM 波形的无效时段被缩短。

图 7-33. 检测到触发信号紧邻 INTTPnCC1 信号产生之后



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

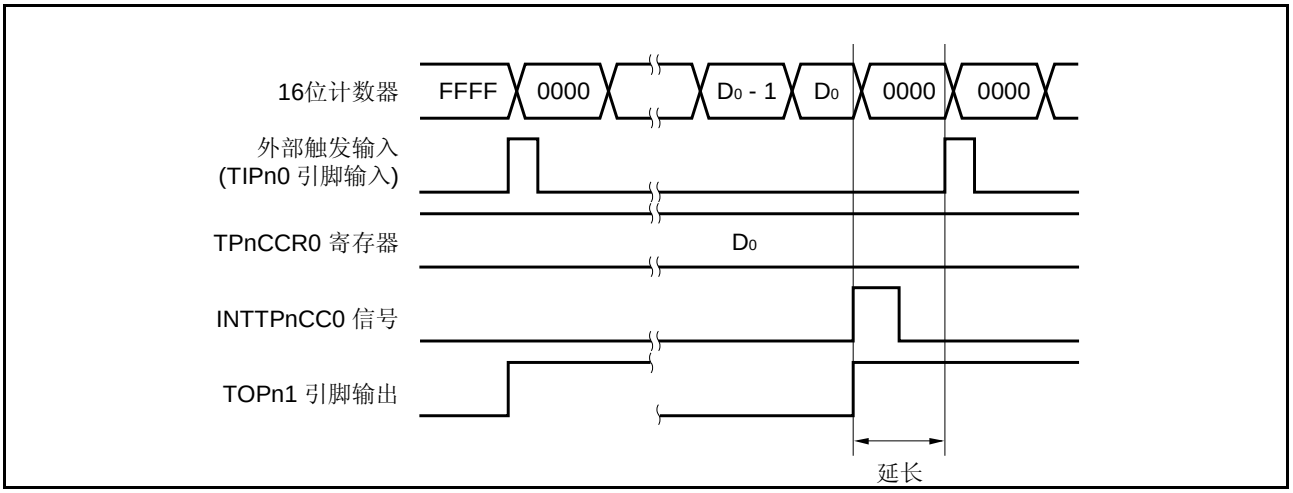
图 7-34. 检测到触发信号紧邻 INTTPnCC1 信号产生之前



（d）检测到触发信号紧邻 INTTPnCC0 产生之前或之后

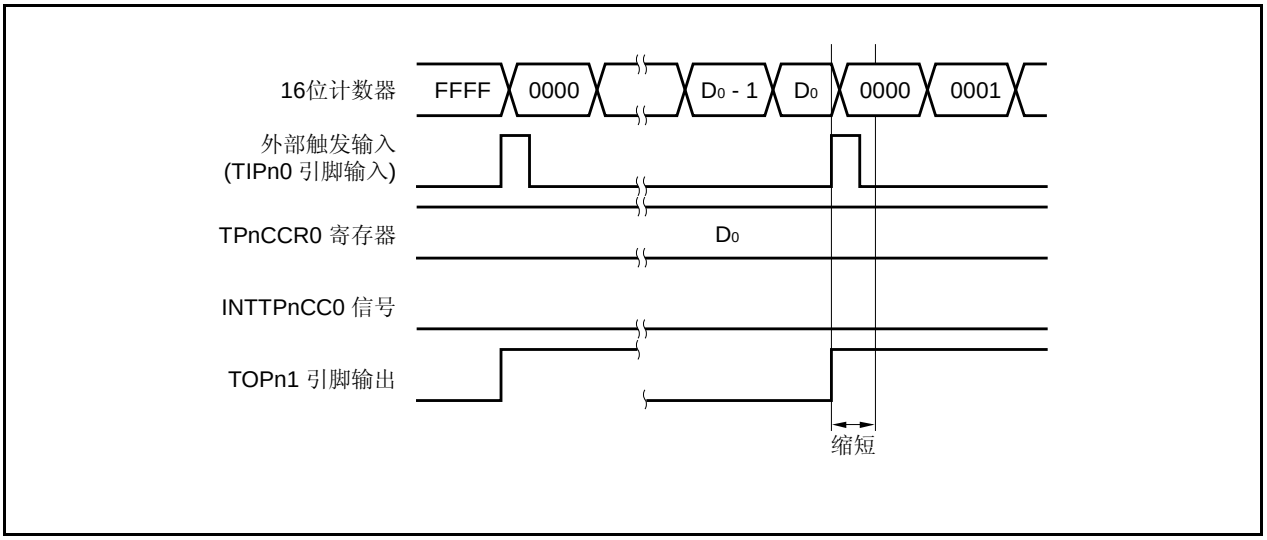
如果在 INTTPnCC0 信号产生后，立刻检测到触发信号，则 16 位计数器立刻清零为 0000H，并继续递增计数。因此，TOPn1 引脚的有效时段被延长，延长长度为 INTTPnCC0 信号产生时刻至检测到触发信号时刻的时间间隔。

图 7-35. 检测到触发信号紧邻 INTTPnCC0 信号产生之后



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

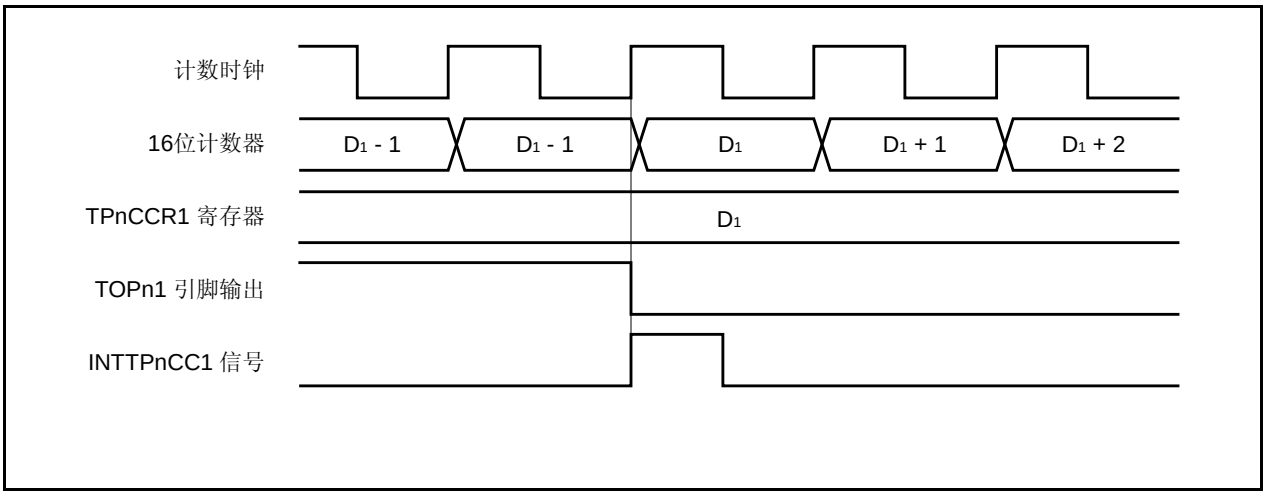
图 7-36. 检测到触发信号紧邻 INTTPnCC0 信号产生之前



（e）比较匹配中断请求信号（INTTPnCC1）的发生时序

在外部触发脉冲输出模式下，当 16 位计数器的计数值与 TPnCCR1 寄存器的值匹配时，产生 INTTPnCC1 中断请求信号。

图 7-37. 比较匹配中断请求信号（INTTPnCC1）的发生时序



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

在单次触发脉冲输出模式下，当 TPnCTL0.TPnCE 位被置为 1 时，TMPn 等待触发。在检测到外部触发输入的有效边沿时，TMPn 开始递增计数，同时从 TOPn1 引脚输出一个单次触发脉冲。

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

备注 **1.** 关于如何设置 TIPn0, TOPn0 和 TOPn1 引脚，参见**表 7-2 TMPn 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。

2. 关于如何使能 INTTPnCC0 和 INTTPnCC0 中断信号，参见**第 22 章 中断服务/异常处理。**

图 7-38. 单次触发脉冲输出模式下的 TMPn 配置

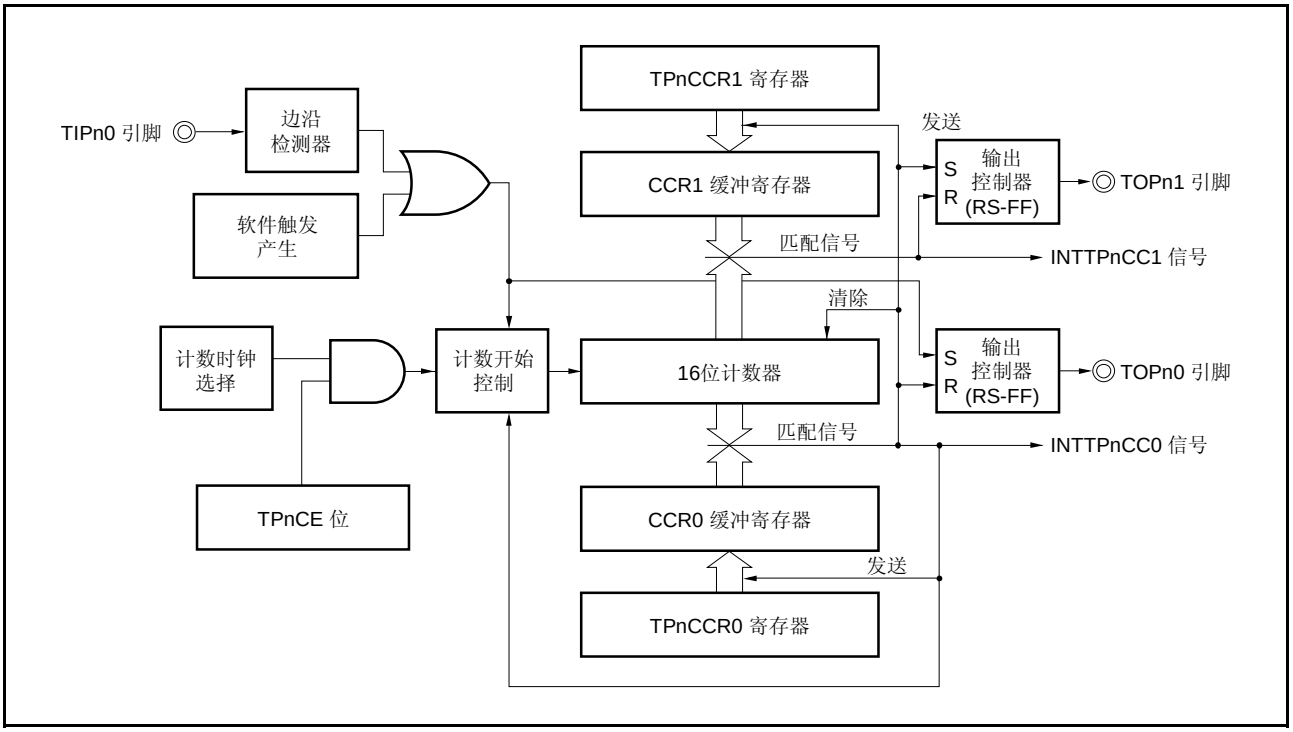
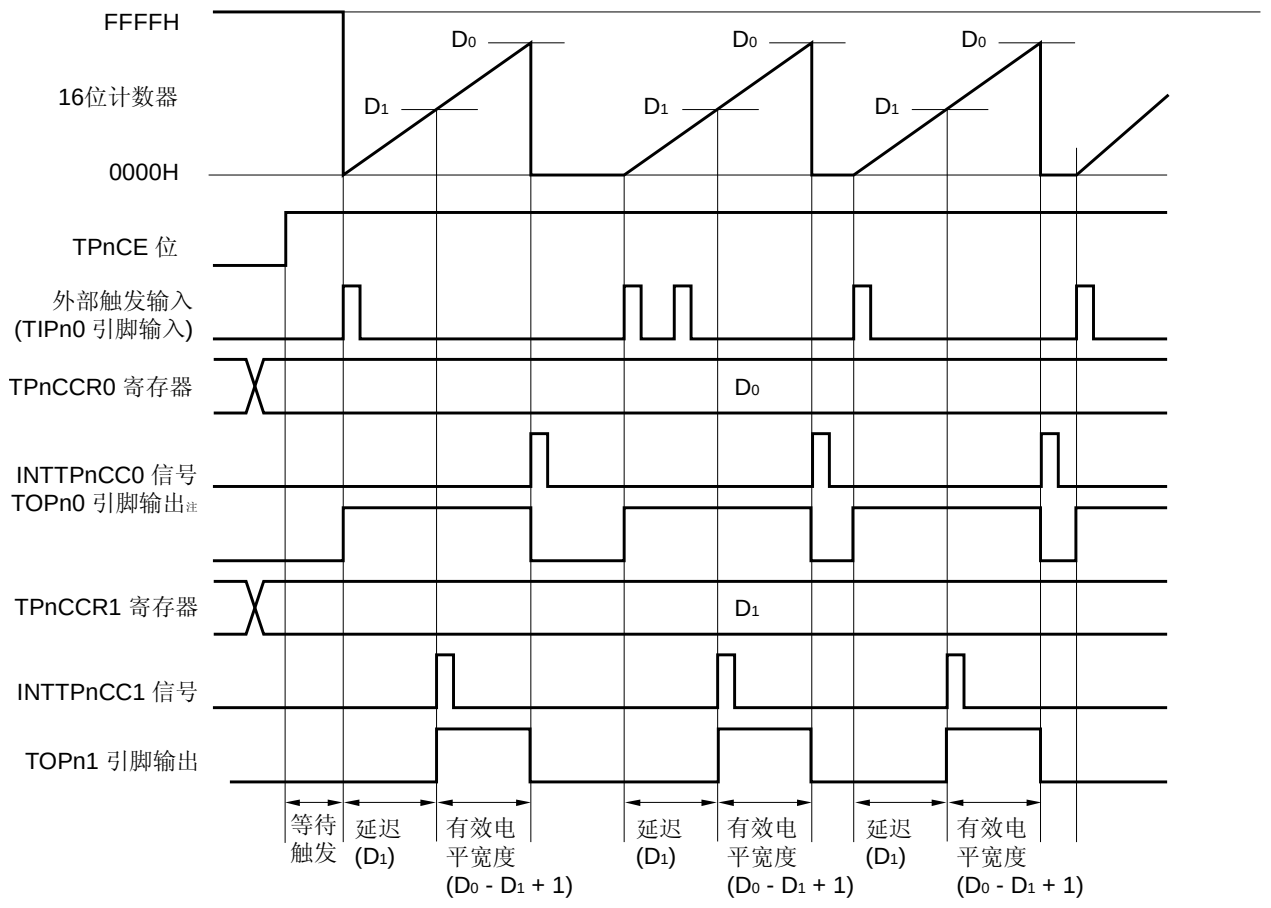


图 7-39. 单次触发脉冲输出模式的基本操作时序



注 TOPn0 引脚的输出也可以用作 TIPn0 引脚的输入。当使用 TOPn0 引脚的输出也可以用作 TIPn0 引脚的输入时，应使用软件触发，而不用外部触发。

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

单次触发脉冲的输出延迟时间和有效电平宽度可计算如下：

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

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

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

图 7-40. 单次触发脉冲输出模式下的寄存器设置（1/2）

（a）TMPn 控制寄存器 0（TPnCTL0）

	TPnCE					TPnCKS2	TPnCKS1	TPnCKS0
TPnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

选择计数时钟

0: 停止计数
1: 允许计数

（b）TMPn 控制寄存器 1（TPnCTL1）

		TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0/1	0	0	0	0	1	1

0, 1, 1:
单次触发脉冲输出模式

写入1，产生
一个软件触发

（c）TMPn 输入/输出控制寄存器 0（TPnIOC0）

					TPnOL1	TPnOE1	TPnOL0	TPnOE0
TPnIOC0	0	0	0	0	0/1	0/1	0/1注	0/1注

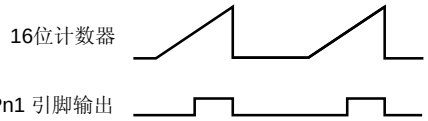
0: 禁止 TOPn0 引脚输出
1: 允许 TOPn0 引脚输出

禁止TOPn0引脚操作时
输出电平的设置
0: 低电平
1: 高电平

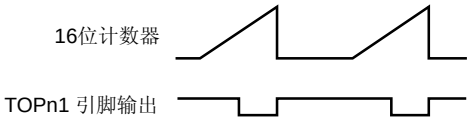
0: 禁止 TOPn1 引脚输出
1: 允许 TOPn1 引脚输出

指定TOPn1引脚输出的有效电平
0: 高电平有效
1: 低电平有效

· 当 TPnOL1 位 = 0时



· 当TPnOL1 位 = 1时



注 在单次触发脉冲输出模式下，当不使用 TOPn0 引脚时，将该位清除为 0。

图 7-40. 单次触发脉冲输出模式下的寄存器设置（2/2）

（d）TMPn 输入/输出控制寄存器 2（TPnIOC2）

					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0	0	0/1	0/1

这些位选择外部触发输入的有效沿

（e）TMPn 计数器读取缓冲寄存器（TPnCNT）

通过读取该寄存器，可以读出 16 位计数器的值。

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

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

单次触发脉冲有效电平宽度＝（$D_0 - D_1 + 1$）× 计数时钟周期

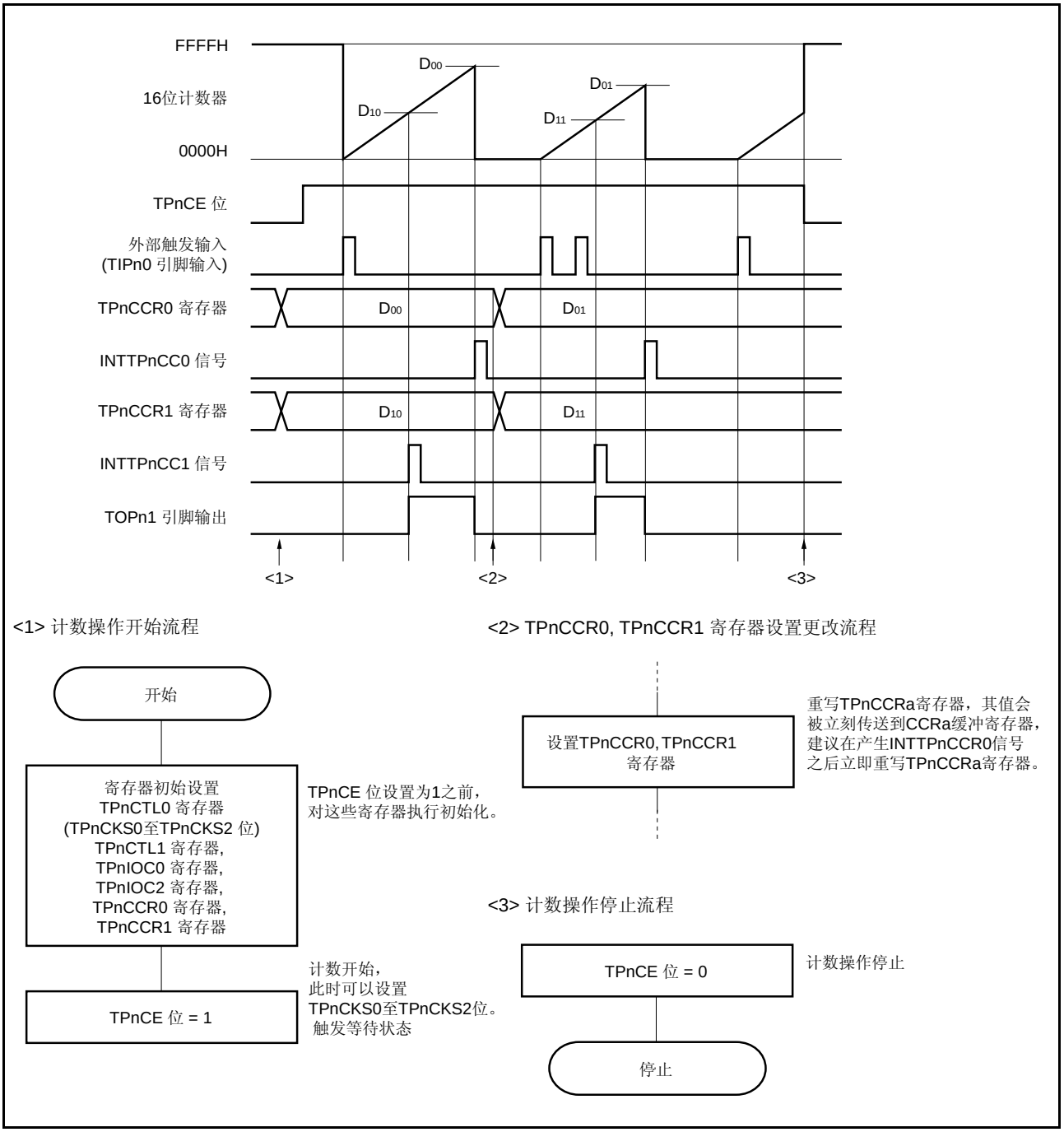
单次触发脉冲输出延迟时间＝D_1 × 计数时钟周期

注意事项 如果 TPnCCR1 寄存器中设置的值大于 TPnCCR0 寄存器的值，那么，即使在单次触发脉冲输出模式下也不会从 TOPn1 引脚输出单次触发脉冲。

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

（1）单次触发脉冲输出模式下的操作

图 7-41. 单次触发脉冲输出模式的时序和操作处理



（2）使用单次触发脉冲输出模式

（a）重写 TPnCCRa 寄存器

要将 TPnCCRa 寄存器的值改为比当前值较小的值，应首先停止计数，然后改变其设定值。
当 TPnCCR0 寄存器的值从 D_{00} 改变为 D_{01} 时，且 TPnCCR1 寄存器的值从 D_{10} 改变为 D_{11} 时，如果在下列条件时该寄存器被重写，则不会输出预期的单次触发脉冲。

条件 1 在重写 TPnCCR0 寄存器时，如果：
$D_{00} > D_{01}$ 或
$D_{00} <$ 16 位计数器值 $< D_{01}$

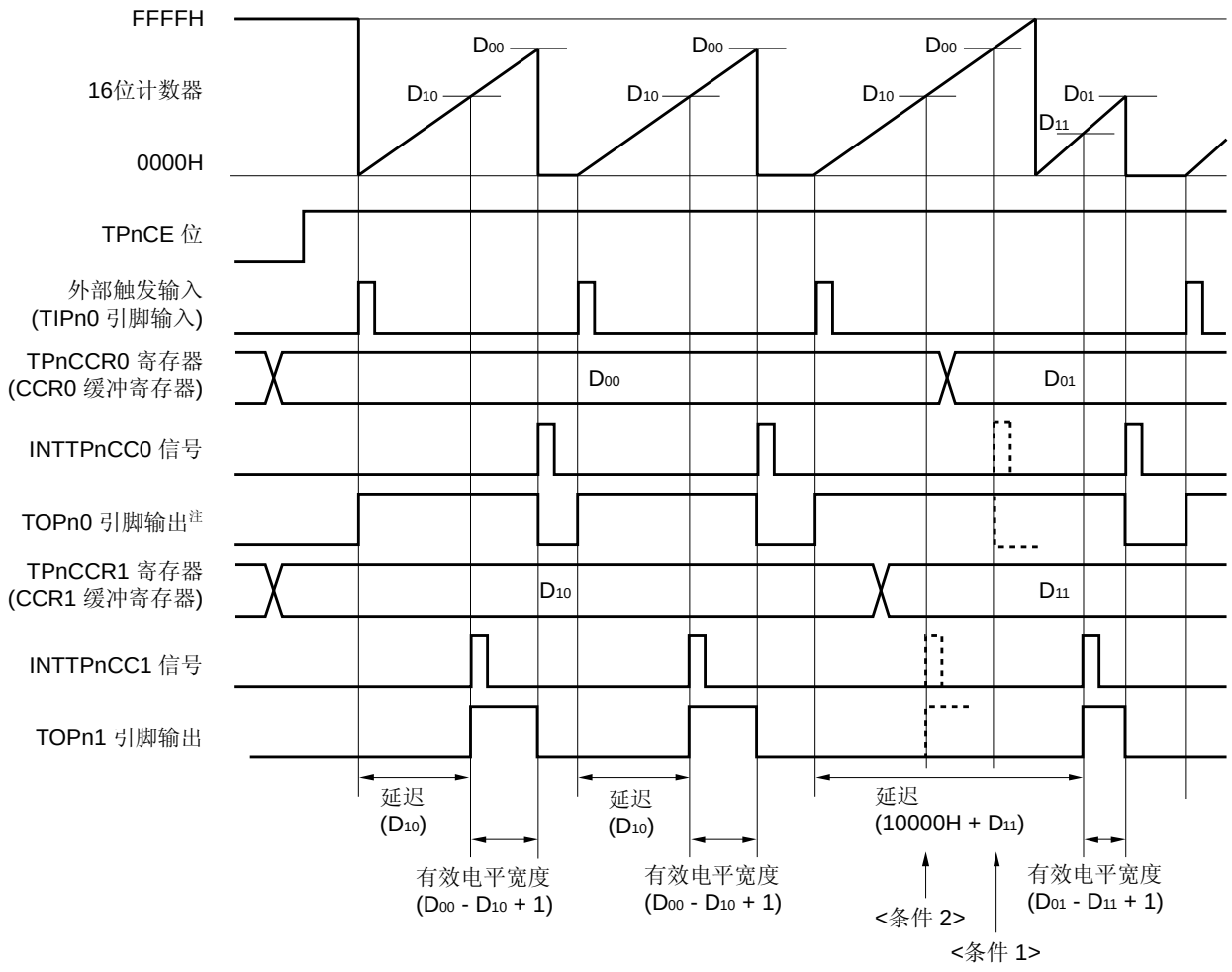
在条件 1 的情况下，16 位计数器不会被清除，且将会在写入新值的周期内溢出。在第一次达到新写入的值(D_{01})的位置，计数器将会被清除。

条件 2 在重写 TPnCCR1 寄存器时，如果：
$D_{10} > D_{11}$ 或
$D_{10} <$ 16 位计数器值 $< D_{11}$

在条件 2 的情况下，在写入新值的周期内 TOPn1 引脚输出不能被反转为有效电平。

在同一个周期内，当条件 1 和条件 2 都满足时的示例，如图 7-42 所示。
16 位计数器向上递增到 FFFFH，溢出，并再次从 0000H 开始递增计数。
当计数值与 D_{11} 匹配时，计数器会产生 INTTPnCC1 信号并使 TOPn1 引脚输出被置为有效电平。接下来，当计数值与 D_{01} 匹配时，计数器产生 INTTPnCC0 信号，使 TOPn1 引脚输出置为无效电平，并停止计数。

图 7-42. 重写 TPnCCRa 寄存器

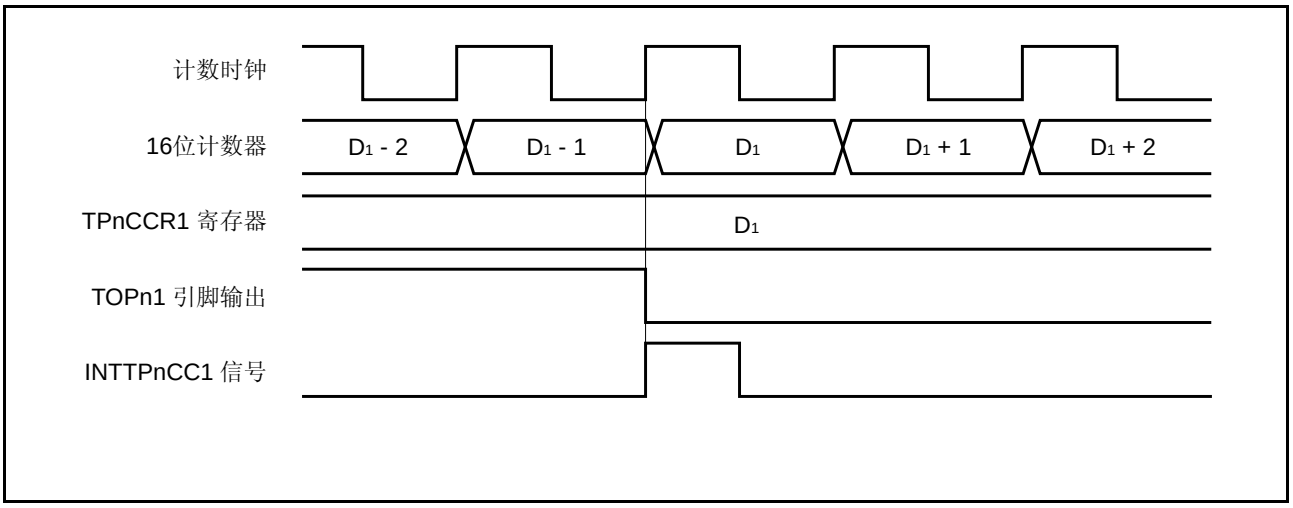


注 TOPn0 引脚的输出也可以用作 TIPn0 引脚的输入。当使用 TOPn0 引脚的输出也可以用作 TIPn0 引脚的输入时，应使用软件触发，而不用外部触发。

（b）比较匹配中断请求信号（INTTPnCC1）的发生时序

在单次触发脉冲输出模式下，当 16 位计数器的计数值与 TPnCCR1 寄存器的值匹配时，产生 INTTPnCC1 中断请求信号。

图 7-43. 比较匹配中断信号（INTTPnCC1）的发生时序



7.4.5 PWM输出模式（TPnMD2 至TPnMD0 位 = 100）

在 PWM 输出模式下，当 TPnCTL0.TPnCE 位被置为 1 时，TMPn 会从 TOPn1 引脚输出 PWM 波形。

由 TOPn0 引脚输出的脉冲，其半周期等于 PWM 的全周期。

备注 **1.** 关于如何设置 TIPn0, TOPn0 和 TOPn1 引脚，参见**表 7-2 TMPn 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。

2. 关于如何使能 INTTPnCC0 和 INTTPnCC0 中断信号，参见**第 22 章 中断服务/异常处理。**

图 7-44. PWM 输出模式下的 TMPn 配置

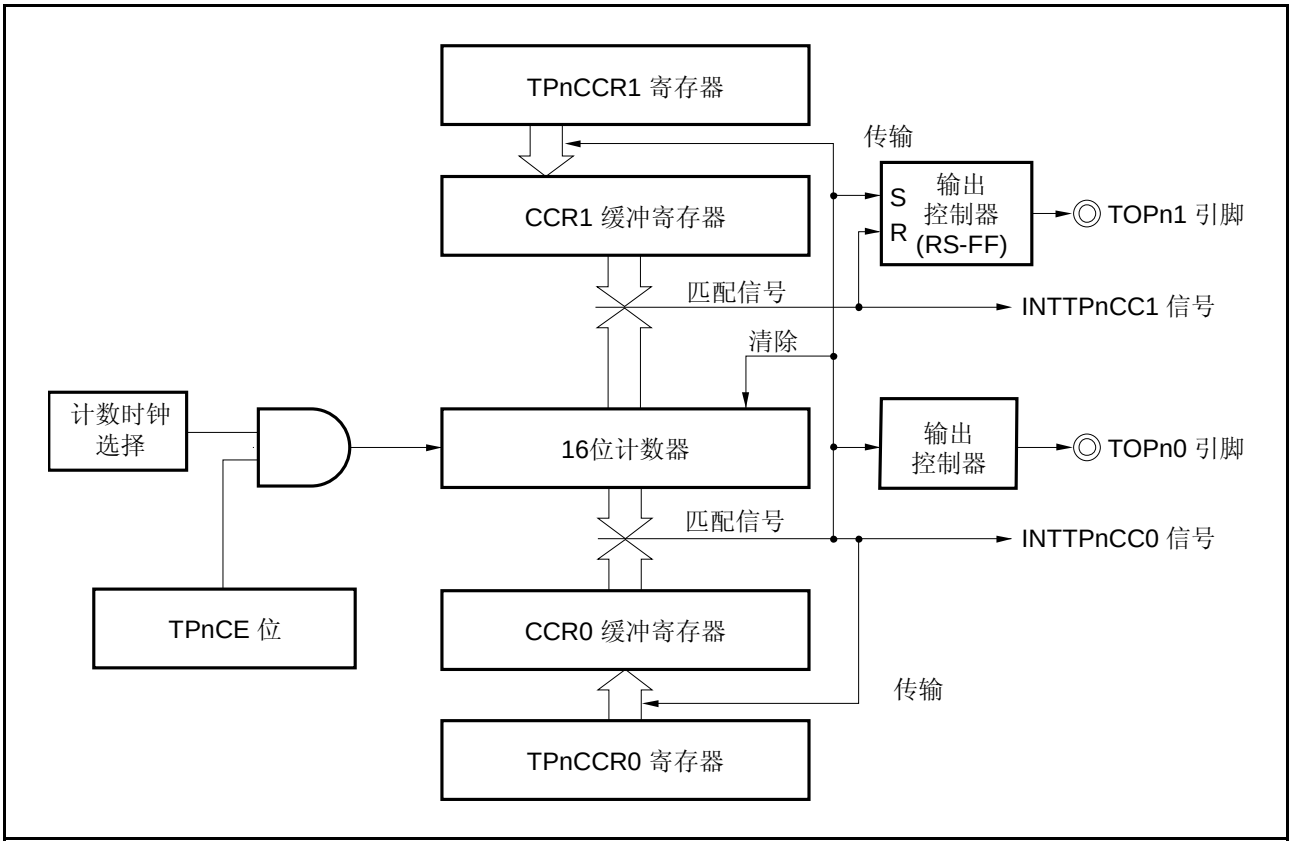
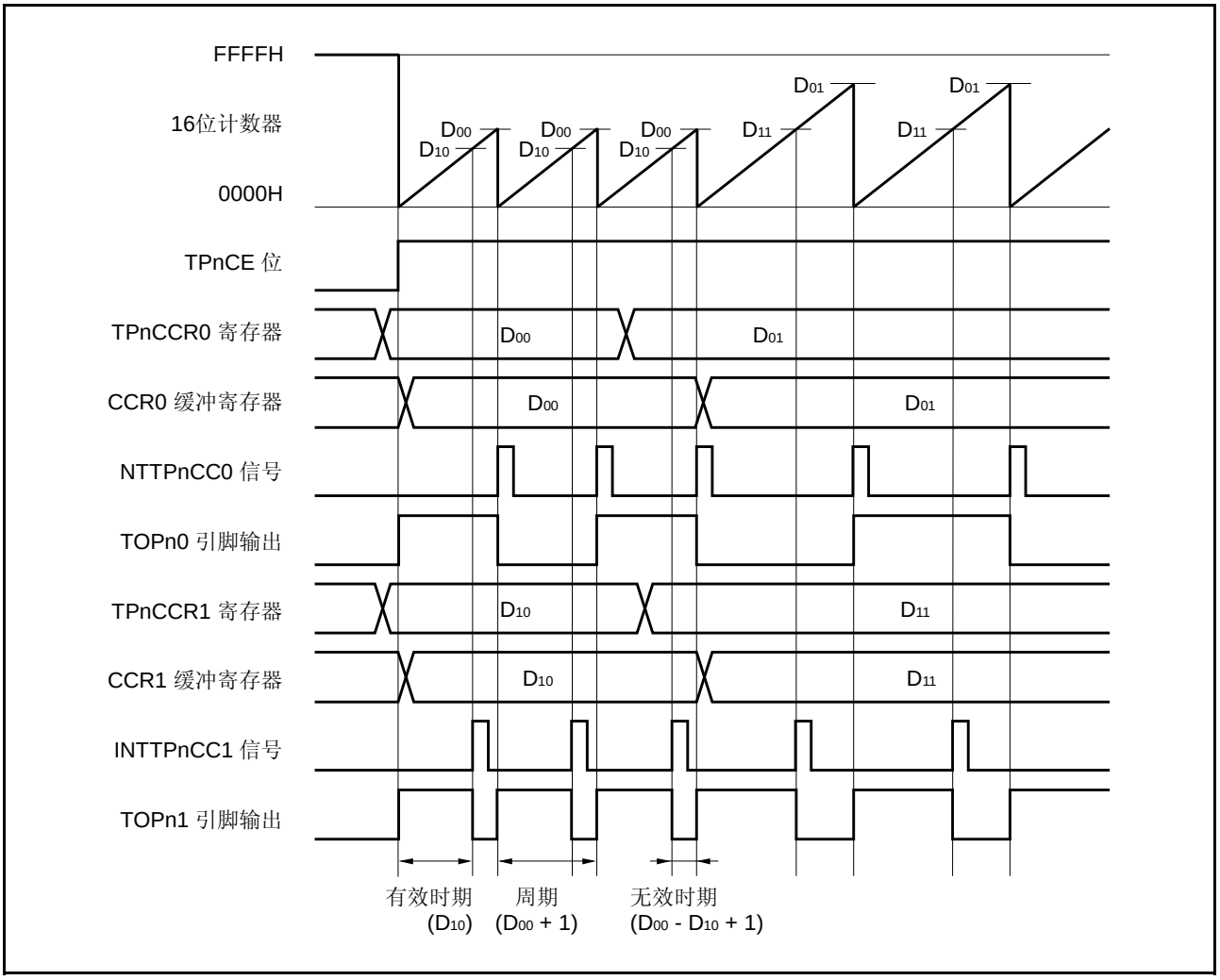


图 7-45. PWM 输出模式的基本操作时序



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

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

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

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

图 7-46. PWM 输出模式时的寄存器设置（1/2）

（a）TMPn 控制寄存器 0（TPnCTL0）

	TPnCE					TPnCKS2	TPnCKS1	TPnCKS0
TPnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

选择计数时钟注 1

0: 停止计数
1: 允许计数

（b）TMPn 控制寄存器 1（TPnCTL1）

		TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0	0/1	0	0	1	0	0

1, 0, 0:
PWM 输出模式

0：计数时钟由TPnCKS0
至TPnCKS2位选择。
1：使用外部事件计数输入信号
计数

（c）TMPn 输入/输出控制寄存器 0（TPnIOC0）

					TPnOL1	TPnOE1	TPnOL0	TPnOE0
TPnIOC0	0	0	0	0	0/1	0/1	0/1注 2	0/1注 2

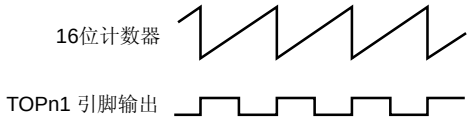
0: 禁止 TOPn0 引脚输出
1: 允许 TOPn0 引脚输出

禁止TOPn0引脚操作时
输出电平的设置
0: 低电平
1: 高电平

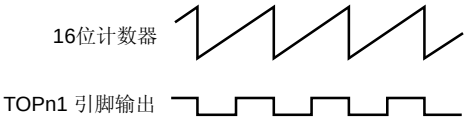
0: 禁止 TOPn1 引脚输出
1: 允许 TOPn1 引脚输出

指定TOPn1引脚输出的有效电平
0: 高电平有效
1: 低电平有效

· 当 TPnOL1 位 = 0时



· 当 TPnOL1 位 = 1时



注 **1.** 当 TPnCTL1.TPnEEE 位为 1 时，这些位的设置无效。
2. 在 PWM 输出模式下，当不使用 TOPn0 引脚时，将该位清除为 0。

图 7-46. PWM 输出模式时的寄存器设置（2/2）

（d）TMPn 输入/输出控制寄存器 2（TPnIOC2）

					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0/1	0/1	0	0

选择外部事件计数输入的有效沿。

（e）TMPn 计数器读取缓冲寄存器（TPnCNT）

通过读取该寄存器，可以读出 16 位计数器的值。

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

如果将 TPnCCR0 寄存器的值设置为 D_0，将 TPnCCR1 寄存器的值设置为 D_1，则 PWM 波形如下：

PWM 波形周期＝（D_0＋1）× 计数时钟周期
PWM 波形有效电平宽度＝D_1 × 计数时钟周期

备注 在 PWM 输出模式下，不使用 TMPn 输入/输出控制寄存器 1（TPnIOC1）和 TMPn 选项寄存器 0（TPnOPT0）。

（1）PWM 输出模式下的操作

图 7-47．PWM 输出模式的时序和操作处理（1/2）

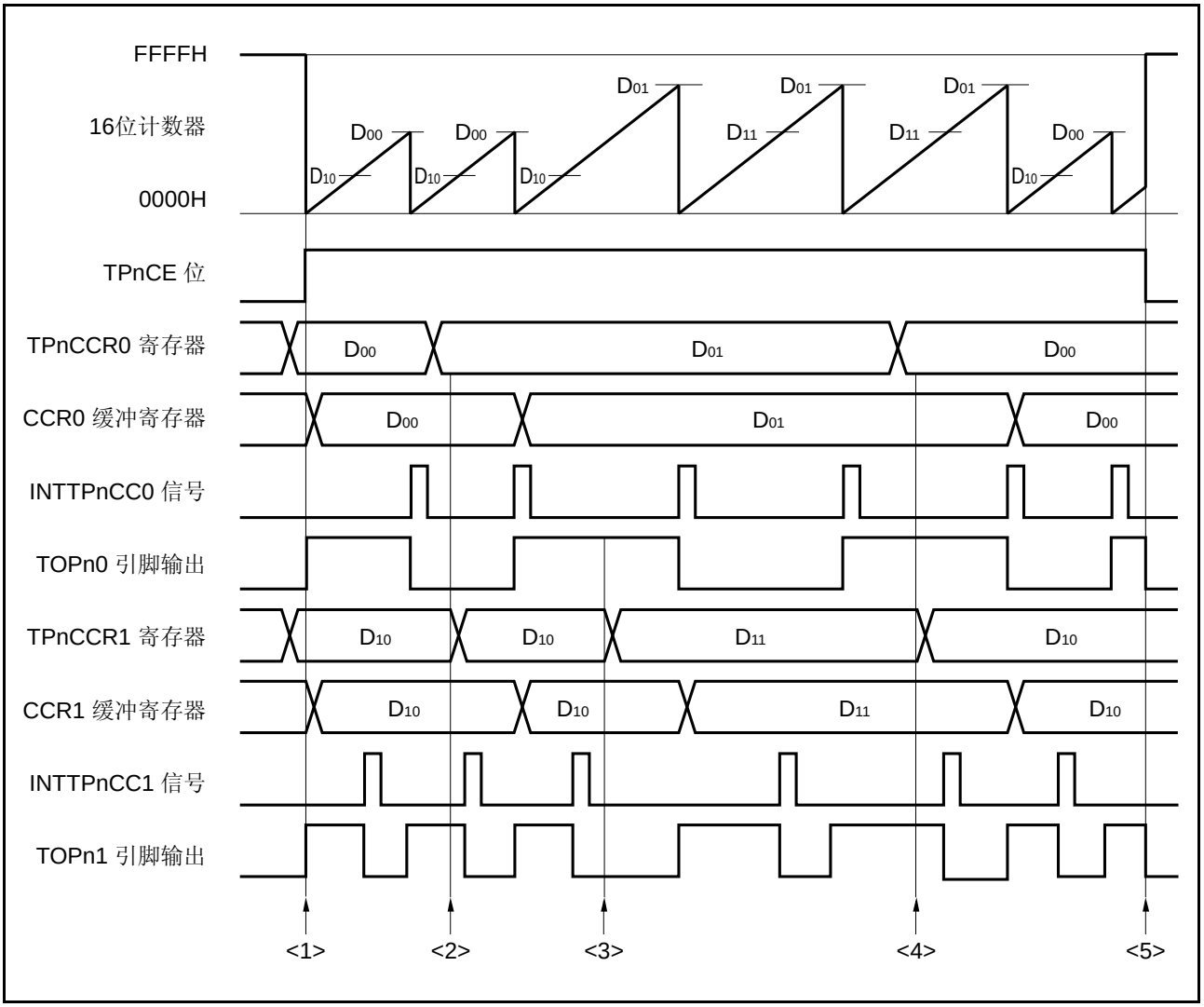
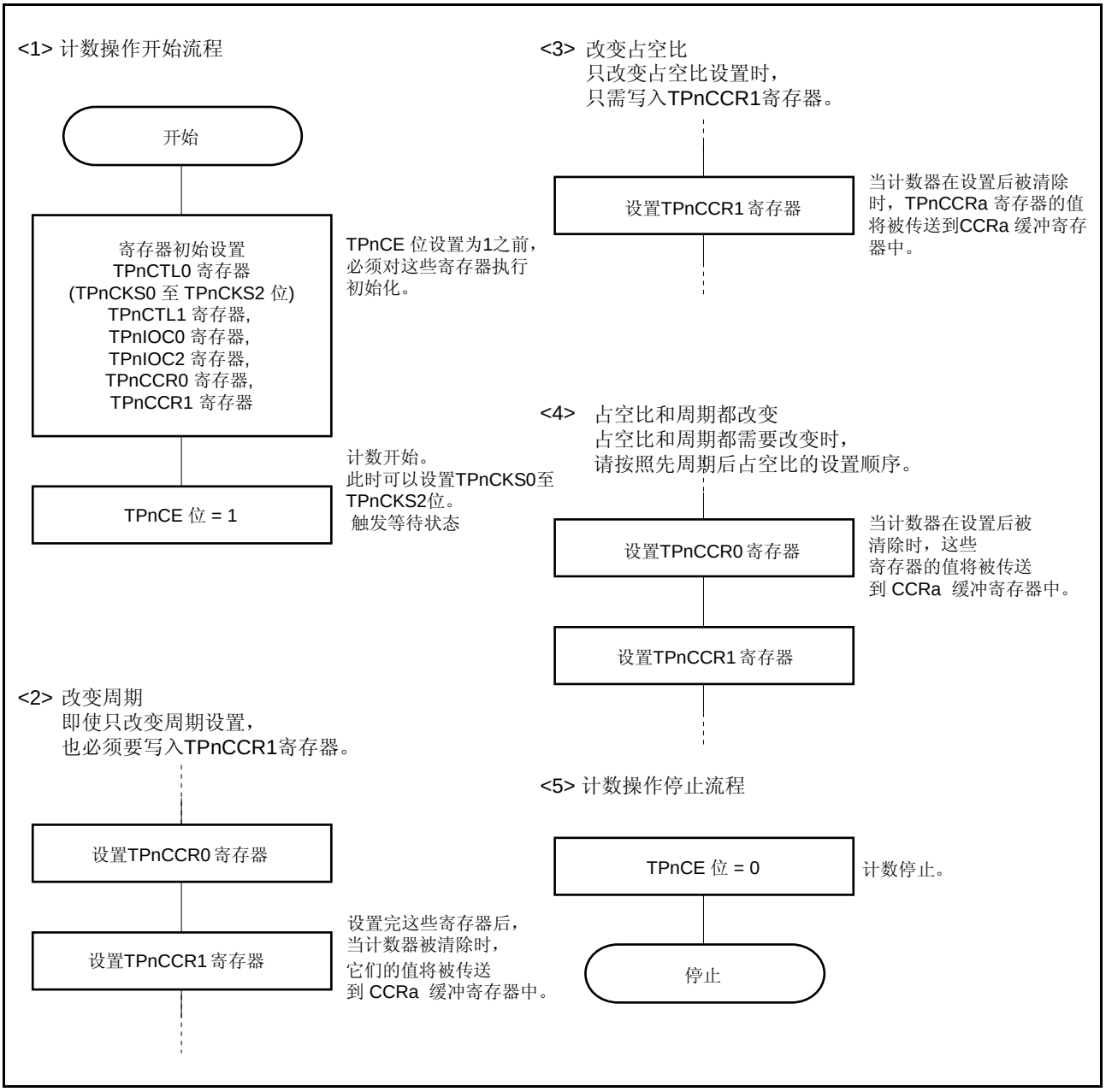


图 7-47. PWM 输出模式的时序和操作处理（2/2）

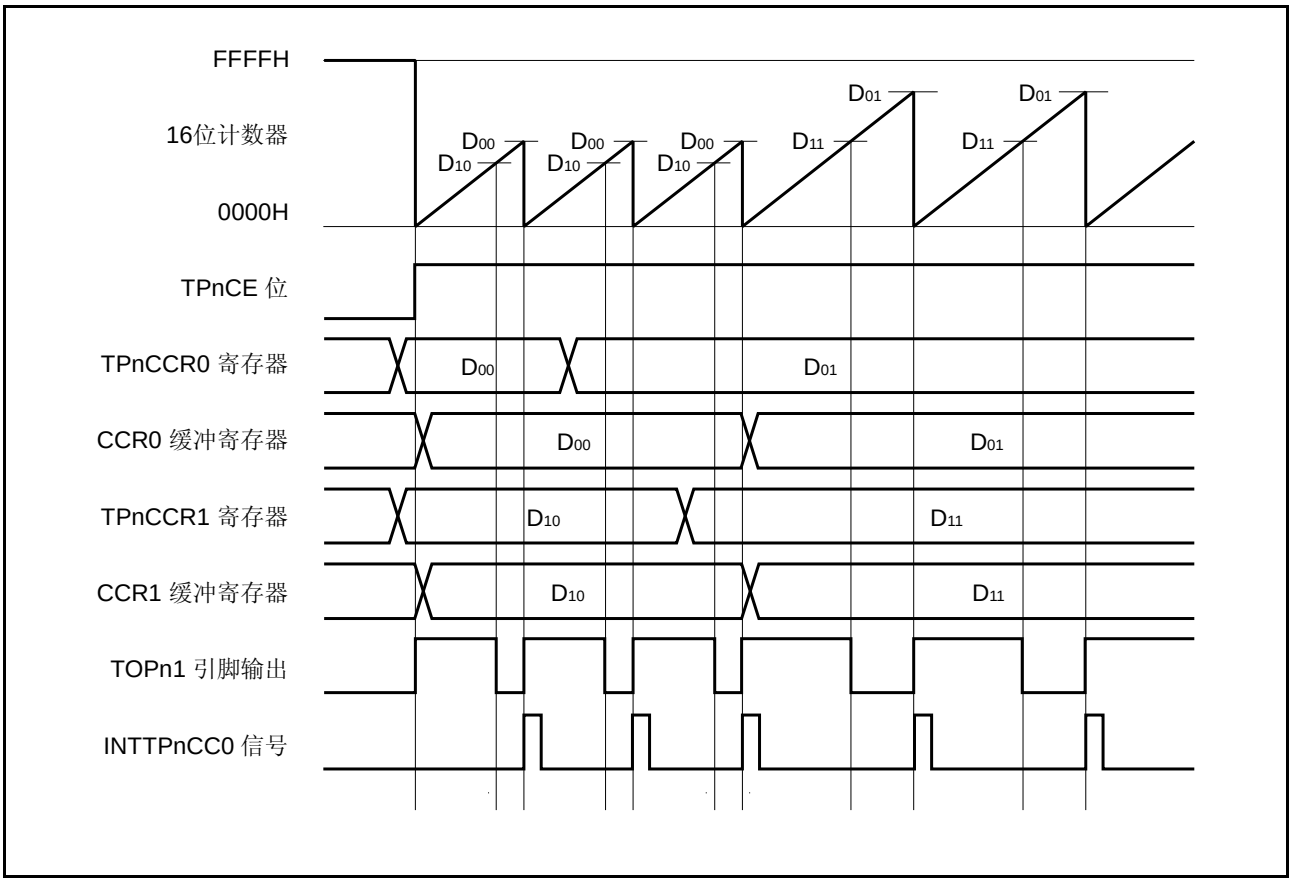


（2）使用 PWM 输出模式

（a）计数器操作过程中改变 PWM 波形

计数器操作过程中，若要改变 PWM 波形，应在改变波形设置之后写入 TPnCCR1 寄存器。当检测到 INTTPnCC1 信号以后，在写入 TPnCCR1 寄存器之后，才能重写 TPnCCRa 寄存器。

图 7-48. 计数器操作过程中改变 PWM 波形



要把数据从 TPnCCRa 寄存器传送到 CCRa 缓冲寄存器，必须对 TPnCCR1 寄存器进行写入操作。
当数据写入 TPnCCR1 寄存器之后，在对 16 位计数器清零的同时，写入到 TPnCCRa 寄存器的值被传送到 CCRa 缓冲寄存器，并作为与 16 位计数器进行比较的值。

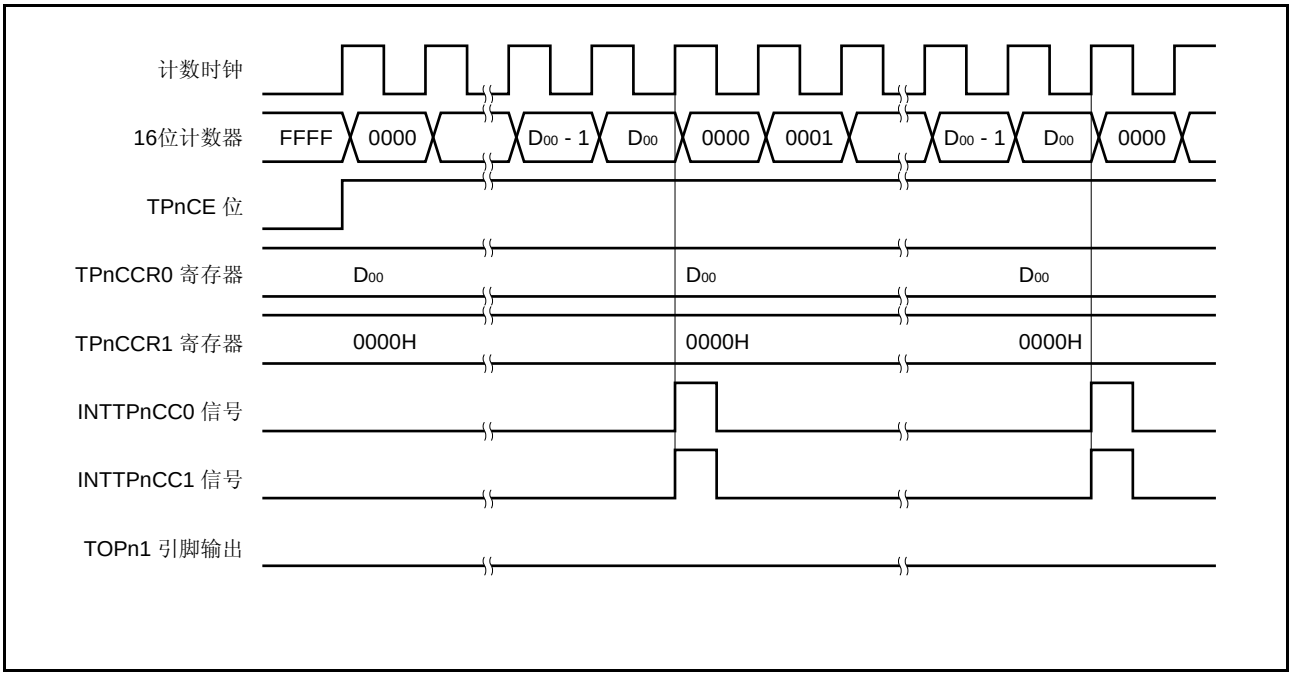
<1> 如果 PWM 波形的周期和有效电平宽度都要改变，应先将周期设定到 TPnCCR0 寄存器，然后将有效电平宽度设定到 TPnCCR1 寄存器。
<2> 如果只改变 PWM 波形的周期，应先把周期设定给 TPnCCR0 寄存器，然后将相同值（原值）写入 TPnCCR1 寄存器（也就是说，与预先为 TPnCCR1 寄存器指定的值相同）。
<3> 只改变 PWM 波形的有效电平宽度（占空比）时，仅需要设置 TPnCCR1 寄存器。

注意事项 在写入 TPnCCR1 寄存器之后，要重写 TPnCCR0 或 TPnCCR1 寄存器，应该在 INTTPnCC0 信号产生之后进行。否则，CCRa 缓冲寄存器的值可能不确定，因为把数据从 TPnCCRa 寄存器传送到 CCRa 缓冲寄存器的时序会与写入 TPnCCRa 寄存器的时序可能会发生冲突。

（b）PWM 波形的 0%和 100% 输出

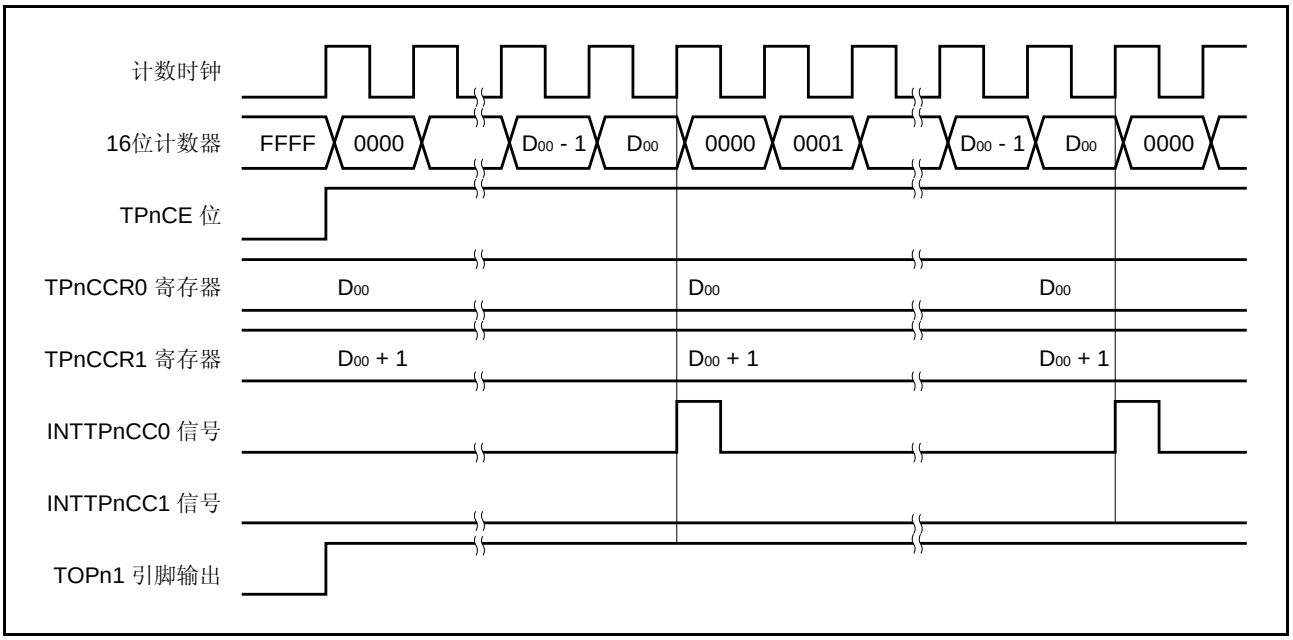
要输出 0%的波形，需要将 TPnCCR1 寄存器清除为 0000H。

图 7-49. 输出 0% PWM 波形



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

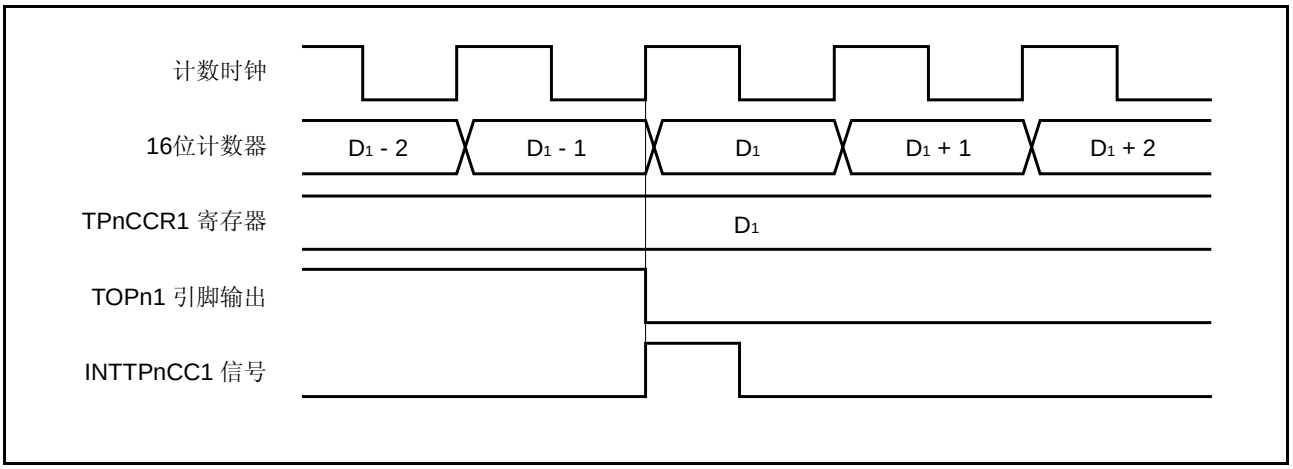
图 7-50. 输出 100% PWM 波形



（c）比较匹配中断请求信号（INTTPnCC1）的发生时序

在 PWM 输出模式下，当 16 位计数器的计数值与 TPnCCR1 寄存器的值匹配时，产生 INTTPnCC1 信号。

图 7-51. 比较匹配中断请求信号（INTTPnCC1）的发生时序



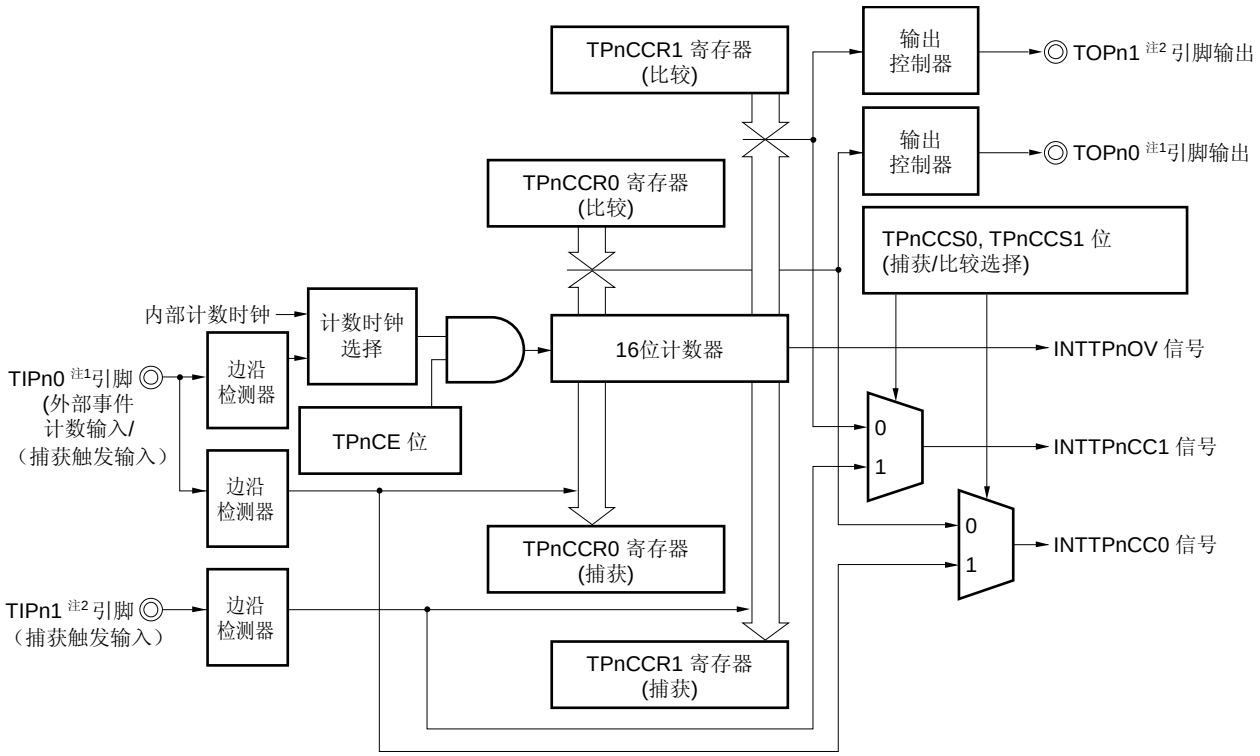
7.4.6 自由运行定时器模式（TPnMD2 至TPnMD0 位 = 101）

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

备注 **1.** 关于如何设置 TIPn0, TIPn1, TOPn0 和 TOPn1 引脚，参见**表 7-2 TMPn 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。

2. 关于如何使能 INTTPnCC0 和 INTTPnCC0 中断信号，参见**第 22 章 中断服务/异常处理。**

图 7-52. 自由运行定时器模式下的 TMPn 配置



注 **1.** 外部事件计数输入/捕获触发输入引脚(TIPn0)可以用作定时器输出引脚(TOPn0)；但是，这些功能每次只能使用一个。

2. 捕获触发输入引脚(TIPn1)也可以用作定时器输出引脚(TOPn1)；但是，这些功能每次只能使用一个。

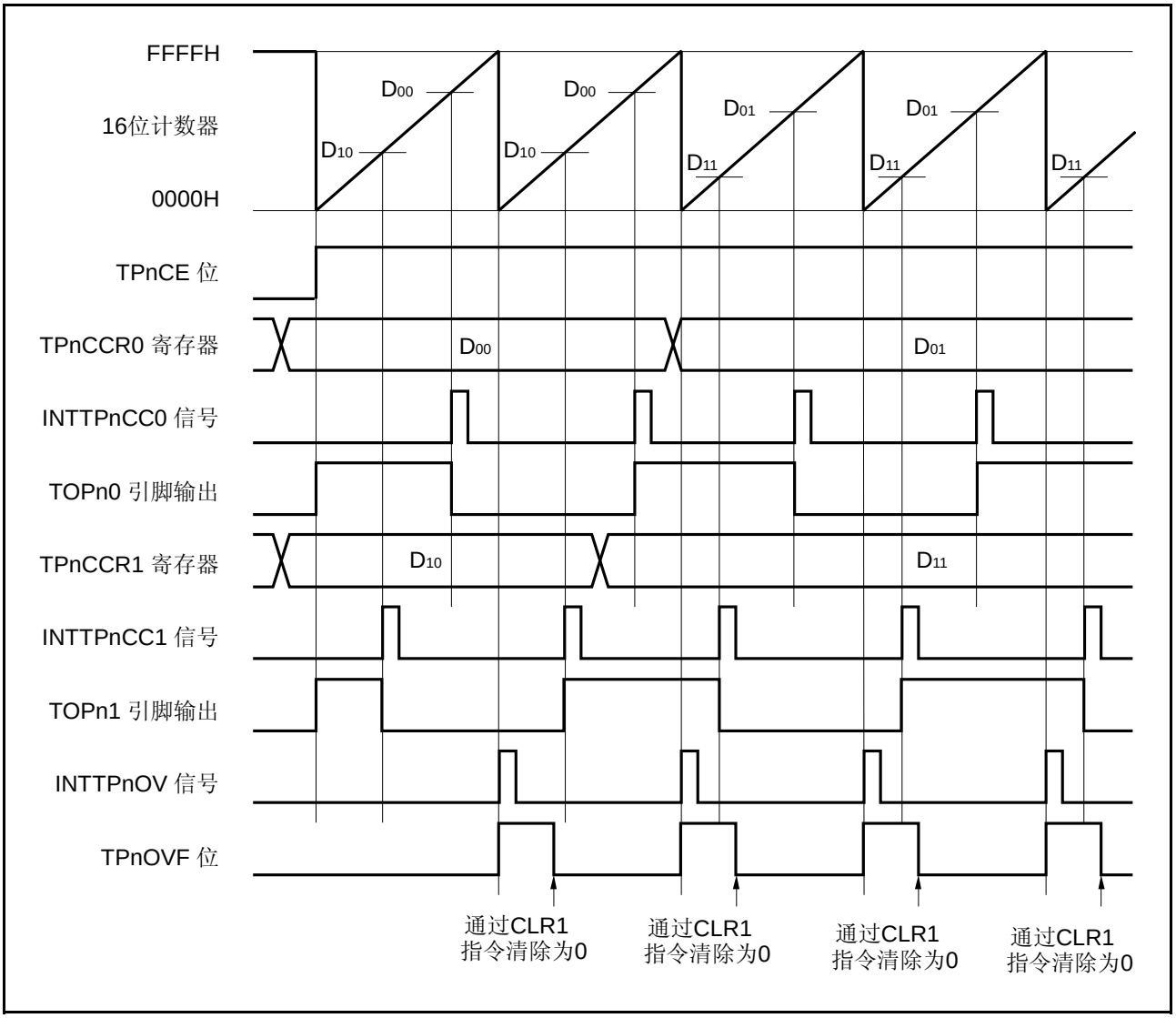
- **比较操作**

当 TPnCE 位被置为 1 时，TMPn 开始计数，TOPn0 和 TOPn1 引脚的输出信号反相。此后当 16 位计数器的计数值与 TPnCCRa 寄存器的值匹配时，产生一个比较匹配中断请求信号（INTTPnCCa），而且 TOPna 引脚的输出信号被反转。

16 位计数器继续对计数时钟同步计数。当它计数到 FFFFH 时，会在下一个计数时钟时产生一个溢出中断请求信号（INTTPnOV），同时计数器清零为 0000H，然后继续进行计数。此时，溢出标志（TPnOPT0.TPnOVF 位）被置为 1，可以通过执行 CLR1 软件指令把溢出标志位清 0。

当计数器工作时，可以重写 TPnCCRa 寄存器。如果进行重写，则新写入的值立即被使用，并与计数值进行比较。

图 7-53. 自由运行定时器模式下的基本操作时序（比较功能）

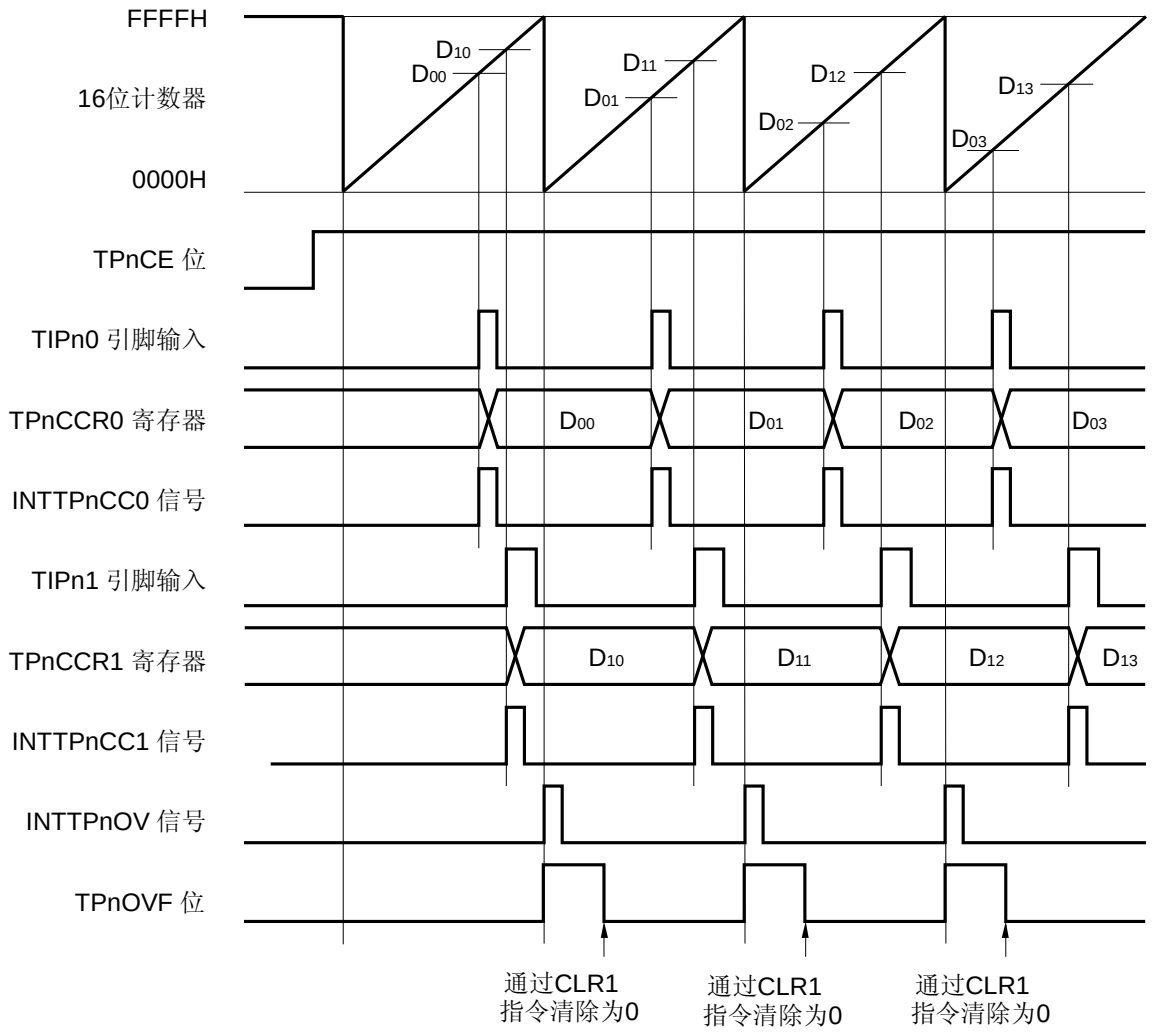


- **捕获操作**

当 TPnCE 位被置为 1 时，16 位计数器开始计数。当检测到有效边沿被输入 TIPna 引脚时，16 位计数器的计数值存储在 TPnCCRa 寄存器中，并产生一个捕获中断请求信号（INTTPnCCa）。

16 位计数器随着计数时钟继续同步计数。当它计数到 FFFFH 时，会在下一个计数时钟时产生一个溢出中断请求信号（INTTPnOV），同时计数器清零为 0000H，然后继续进行计数。此时，溢出标志（TPnOPT0.TPnOVF 位）被置为 1，可以通过执行 CLR1 软件指令将溢出标志位清除为 0。

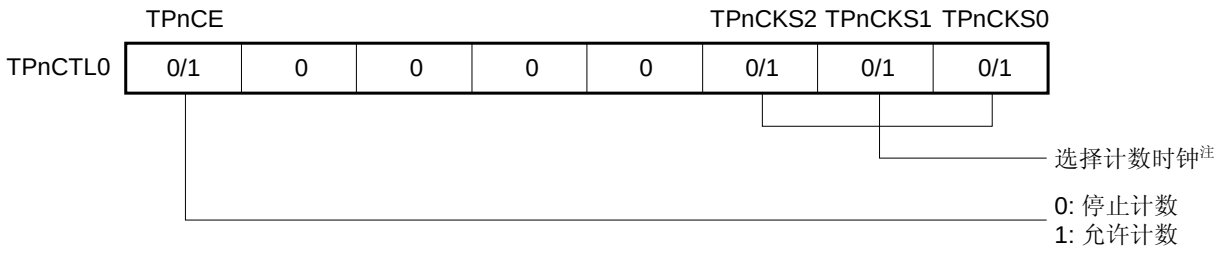
图 7-54. 自由运行定时器模式下的基本操作时序（捕获功能）



备注 有效沿是上升沿。

图 7-55. 自由运行定时器模式下的寄存器设置（1/2）

（a）TMPn 控制寄存器 0（TPnCTL0）

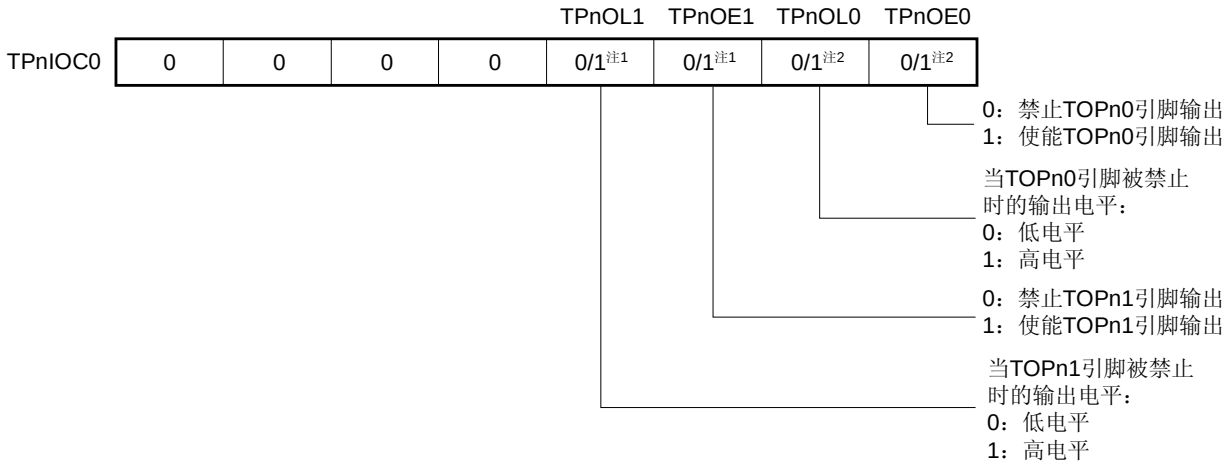


注 当 TPnCTL1.TPnEEE 为 1 时，这些位的设置无效。

（b）TMPn 控制寄存器 1（TPnCTL1）

注 如果 TPnCTL1.TPnEEE 位为 1，不能使用 TIPn0 引脚的捕获功能。

（c）TMPn 输入/输出控制寄存器 0（TPnIOC0）



注 **1.** 当使用 TIPn1 引脚时，不能使用 TOPn1 引脚。
2. 当使用 TIPn0 引脚时，不能使用 TOPn0 引脚。

图 7-55. 自由运行定时器模式下的寄存器设置（2/2）

（d）TMPn 输入/输出控制寄存器 1（TPnIOC1）

					TPnIS3	TPnIS2	TPnIS1	TPnIS0
TPnIOC1	0	0	0	0	0/1	0/1	0/1	0/1

选择TIPn0引脚输入的有效沿

选择TIPn1引脚输入的有效沿

（e）TMPn 输入/输出控制寄存器 2（TPnIOC2）

					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0/1	0/1	0	0

选择外部事件计数输入的有效沿

（f）TMPn 选项寄存器 0（TPnOPT0）

			TPnCCS1	TPnCCS0				TPnOVF
TPnOPT0	0	0	0/1	0/1	0	0	0	0/1

溢出标志

指定 TPnCCR0 寄存器的功能为捕获寄存器或比较寄存器

指定 TPnCCR1 寄存器的功能为捕获寄存器或比较寄存器

（g）TMPn 计数器读取缓冲寄存器（TPnCNT）

通过读取该寄存器，可以读出 16 位计数器的值。

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

根据 TPnOPT0.TPnCCSa 位的设置，这些寄存器可以用作捕获寄存器或比较寄存器。

当用作捕获寄存器时，在检测到有效边沿输入到 TIPna 引脚时，存储 16 位计数器的当前计数值，然后产生 INTTPnCCa 信号。

当用作比较寄存器时，而且 TPnCCRa 寄存器的值被设置为 D_a 时，则当计数器达到（D_a ＋1）时会产生 INTTPnCCa 信号，同时，TOPna 引脚的输出信号被反转。

（1）操作自由运行定时器模式

在自由运行定时器模式下，会发生下列两种操作：

- 捕获操作
- 比较操作

（a）捕获/比较寄存器用作比较寄存器时

图 7-56. 自由运行定时器模式的时序和操作处理（比较功能）（1/2）

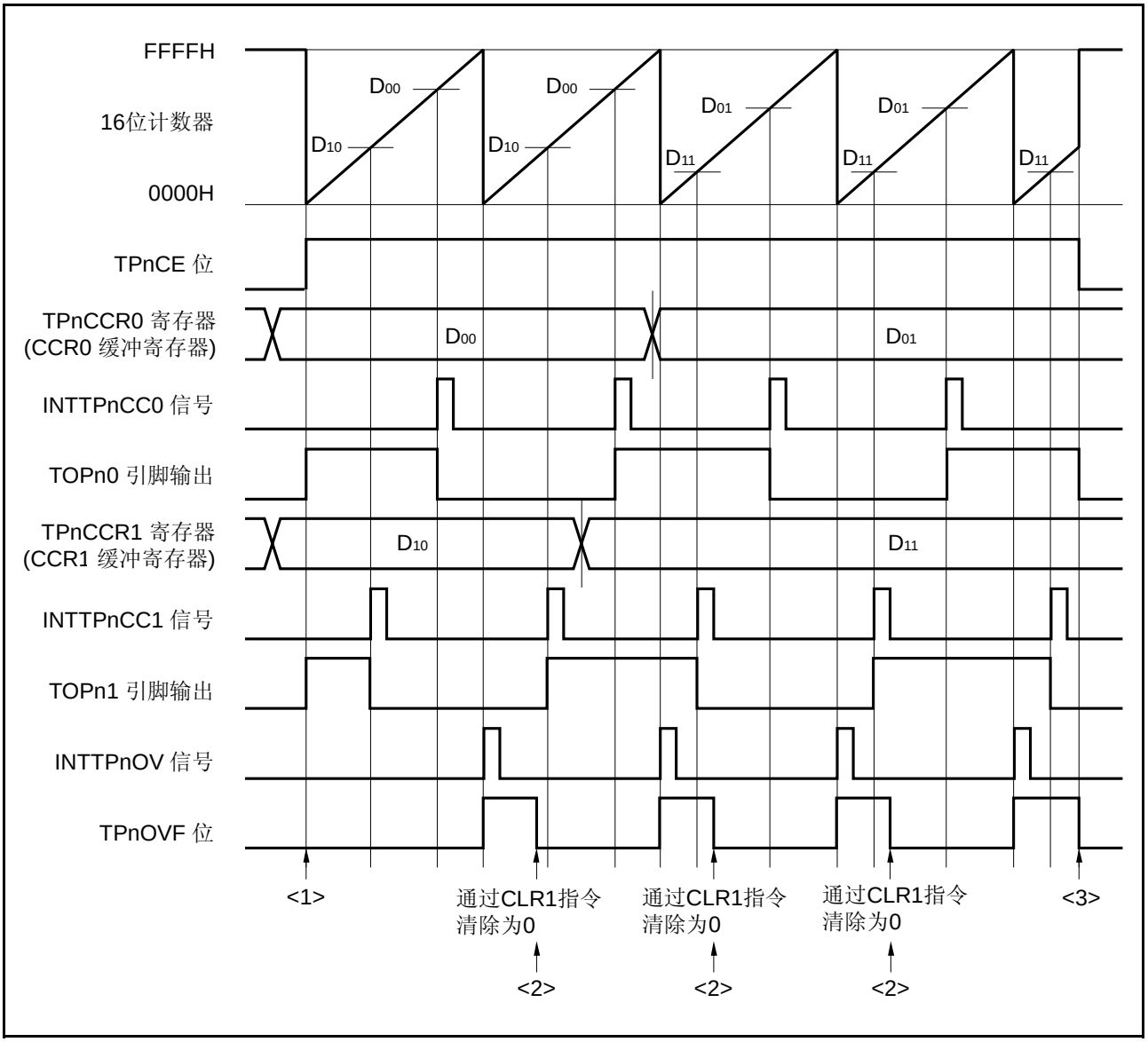
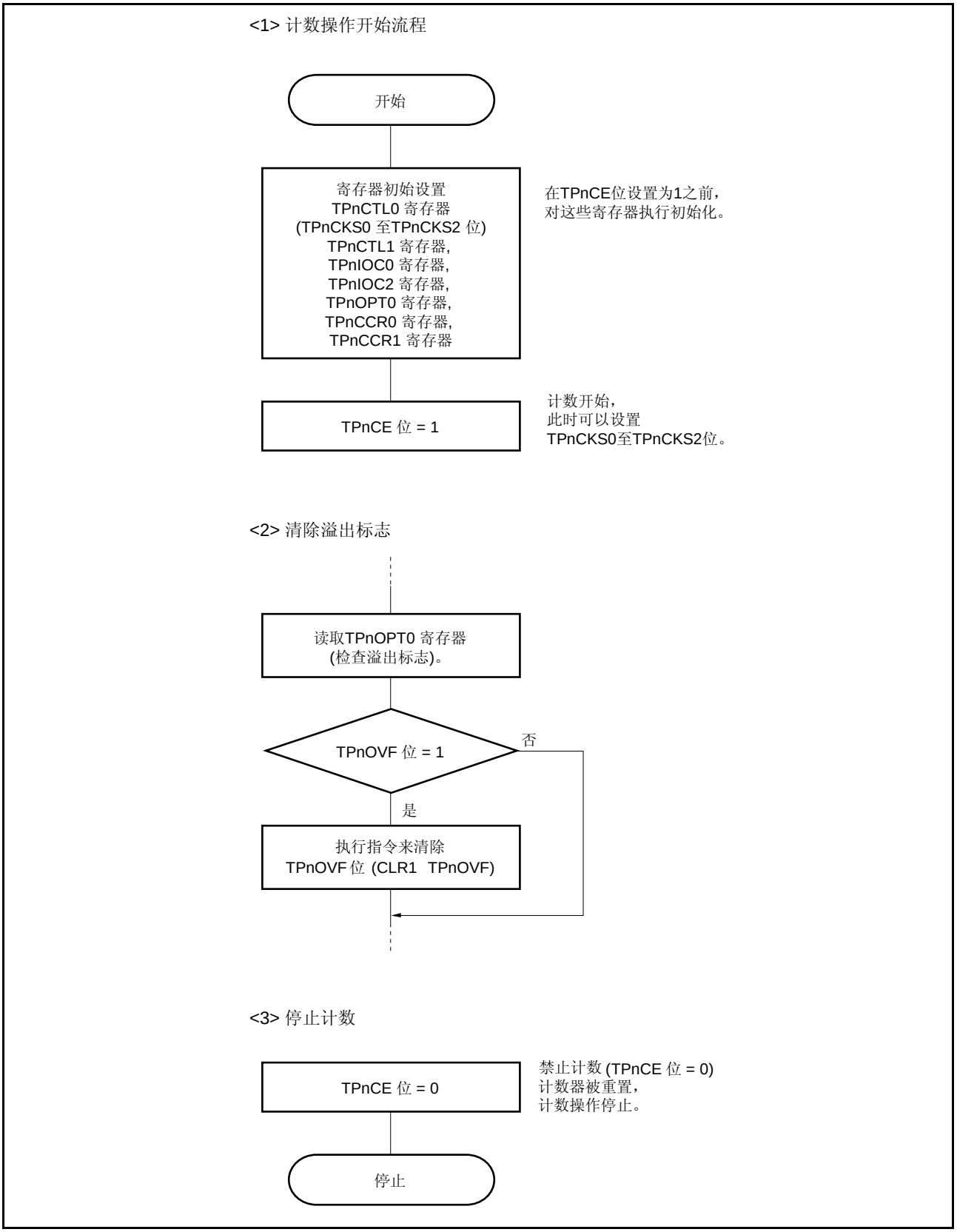


图 7-56. 自由运行定时器模式的时序和操作处理（比较功能）（2/2）



（b）捕获/比较寄存器用作捕获寄存器时

图 7-57. 自由运行定时器模式的时序和操作处理（捕获功能）（1/2）

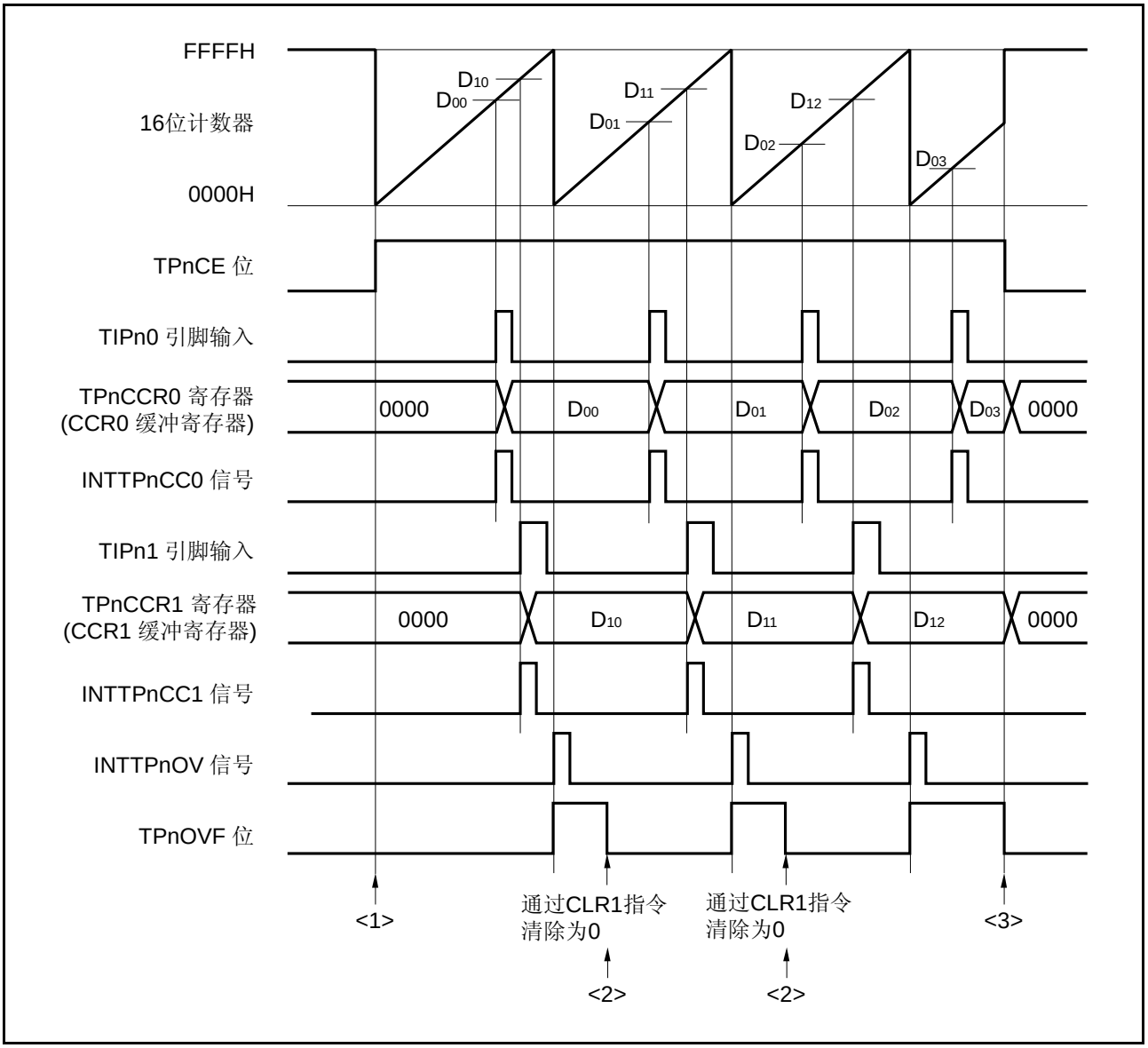
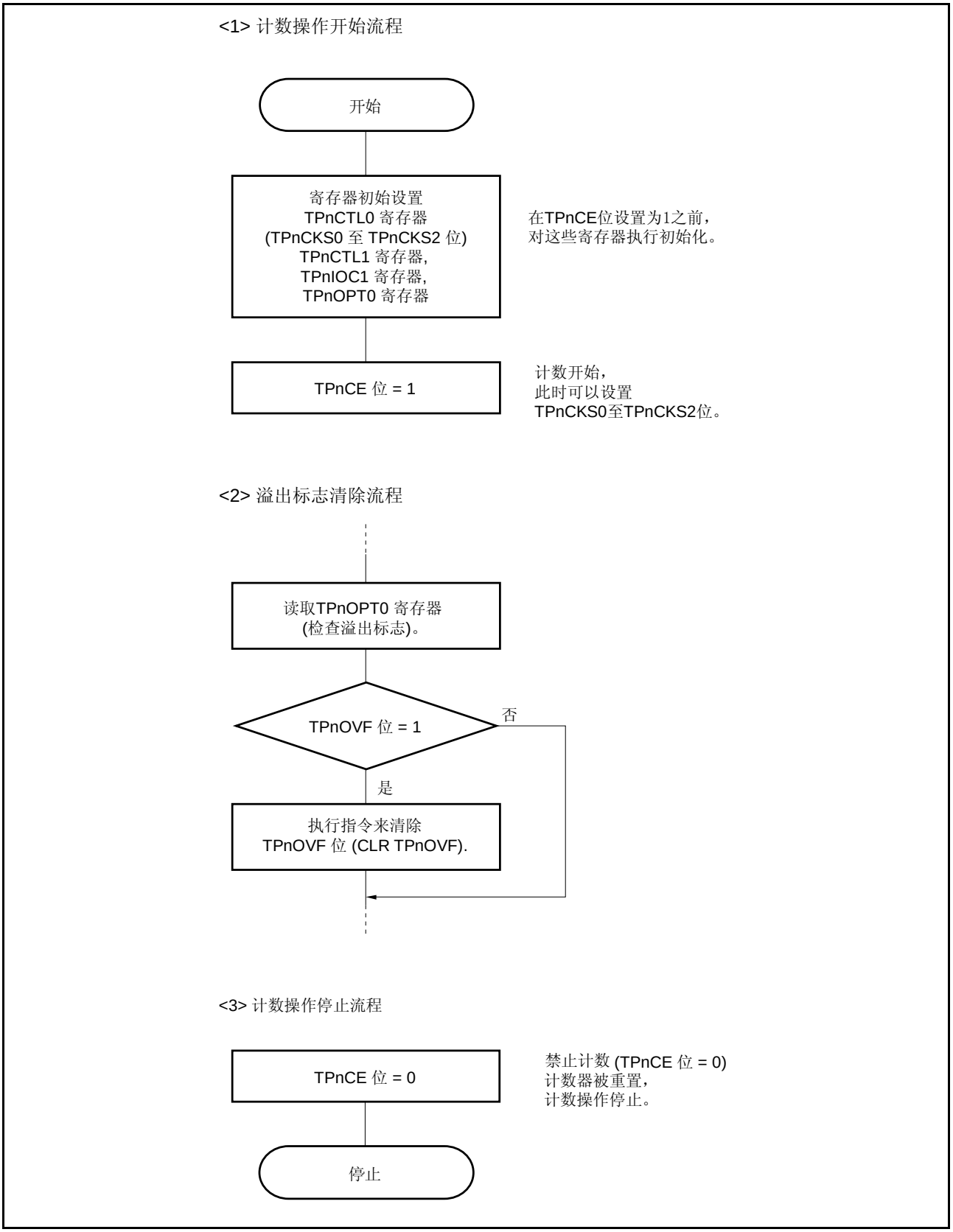


图 7-57. 自由运行定时器模式的时序和操作处理（捕获功能）（2/2）

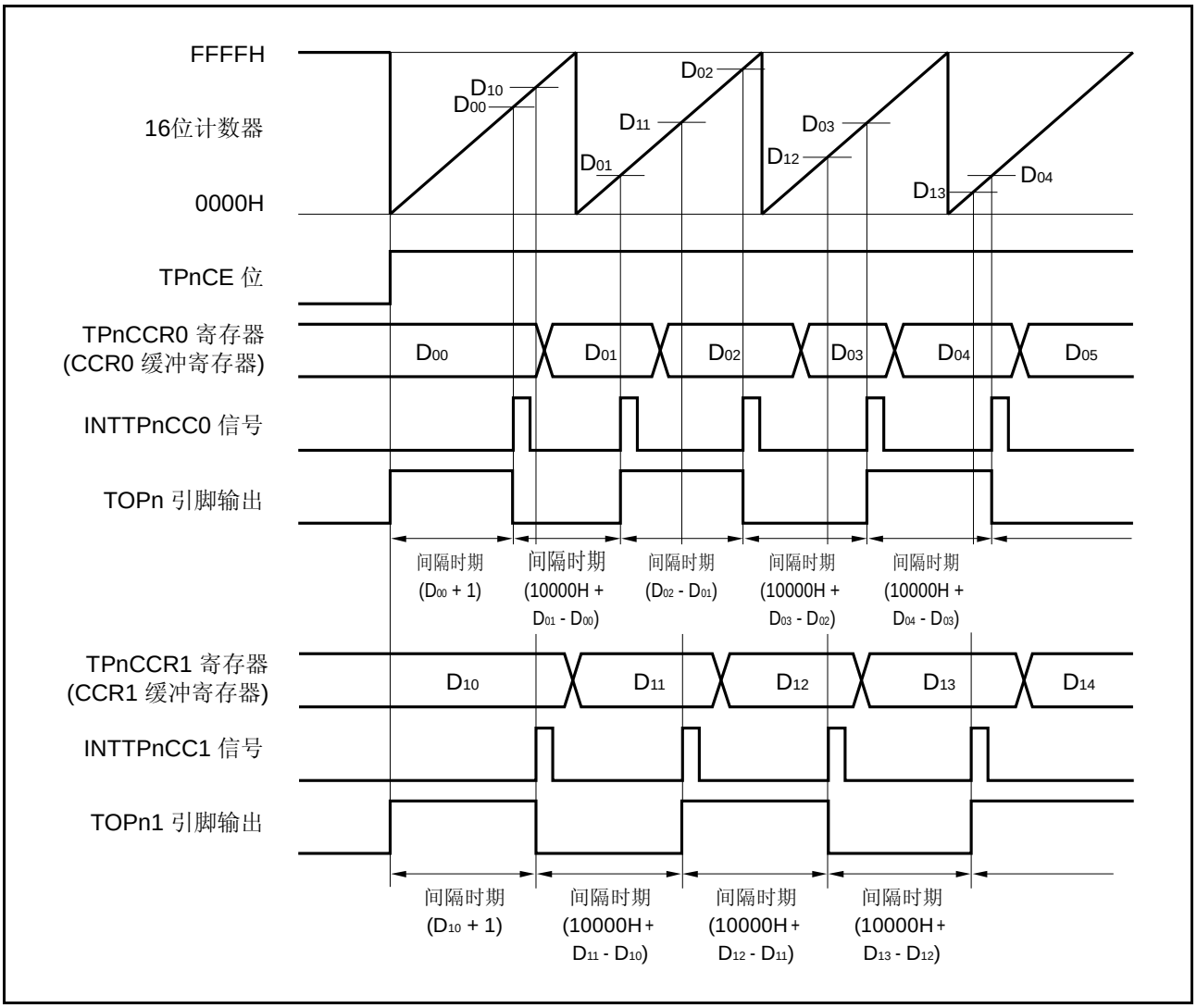


（2）使用自由运行定时器模式

（a）使用 TPnCCRa 作为比较寄存器时的间隔定时操作

当 TMPn 用作间隔定时器，且 TPnCCRa 寄存器用作比较寄存器时，在每次检测到 INTTPnCCa 信号后，都需要通过软件设置一个比较值以产生下一个中断请求信号。

图 7-58. 自由运行定时器模式下的 TMPn 间隔操作



在自由运行定时器模式下进行时间间隔操作时，一个通道可以设置两个时间间隔。

要进行时间间隔操作，当检测到 INTTPnCCa 信号后，必须在中断服务程序中再次设置相应的 TPnCCRa 寄存器的值。

这种情况下要设置的值可由下式计算，其中“D_a”是间隔时间：

比较寄存器默认值： $D_a - 1$

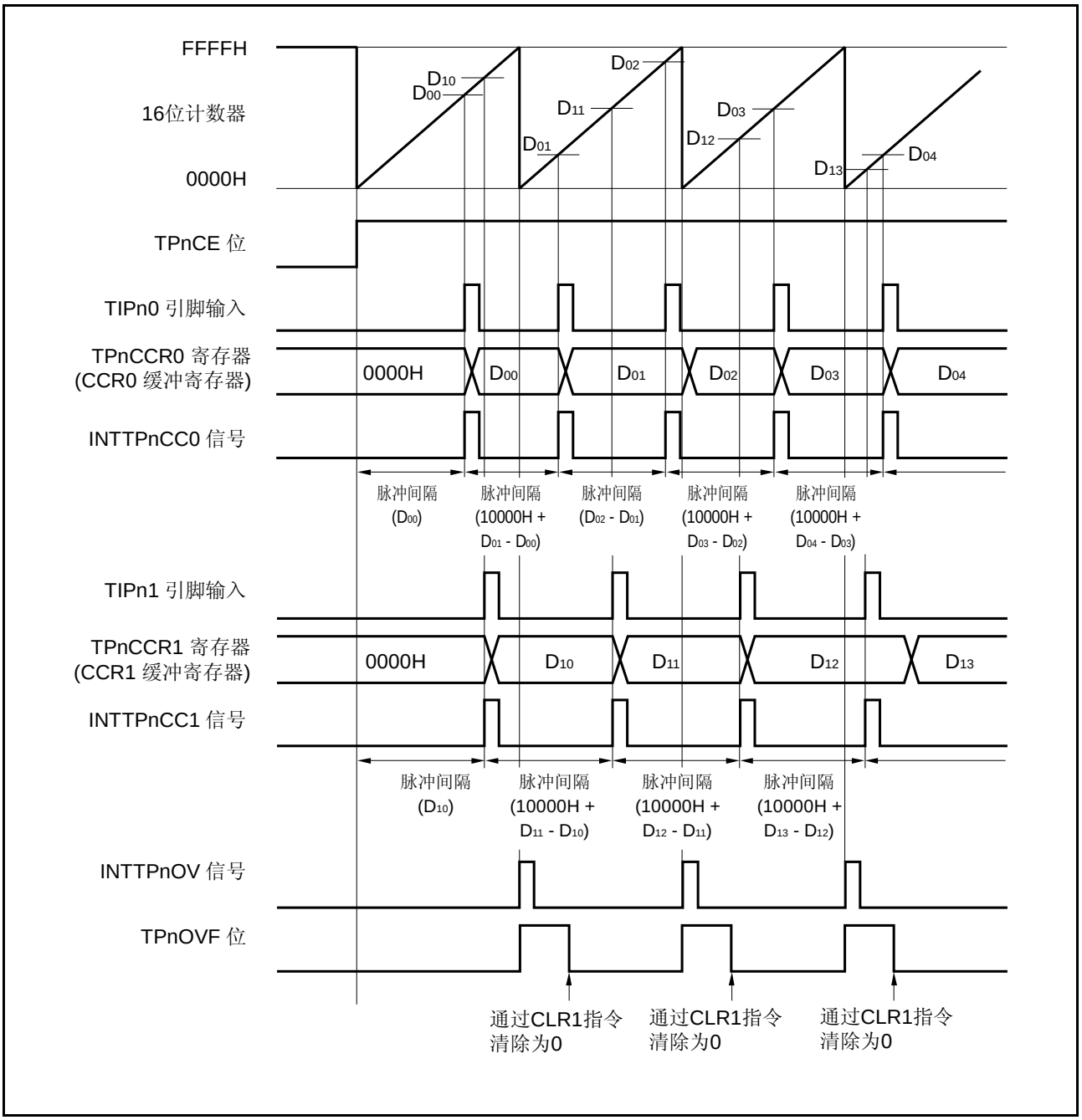
比较寄存器的第二次及其以后的设置值：上次设置值＋D_a

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

（b）TPnCCRa 寄存器用作捕获寄存器进行脉冲宽度测量

在 TPnCCRa 寄存器用作捕获寄存器进行脉宽测量时，每次检测到 INTTPnCCa 信号时，必须要读取捕获寄存器并通过软件计算间隔时间。

图 7-59. 自由运行定时模式下通过 TMPn 进行脉宽测量



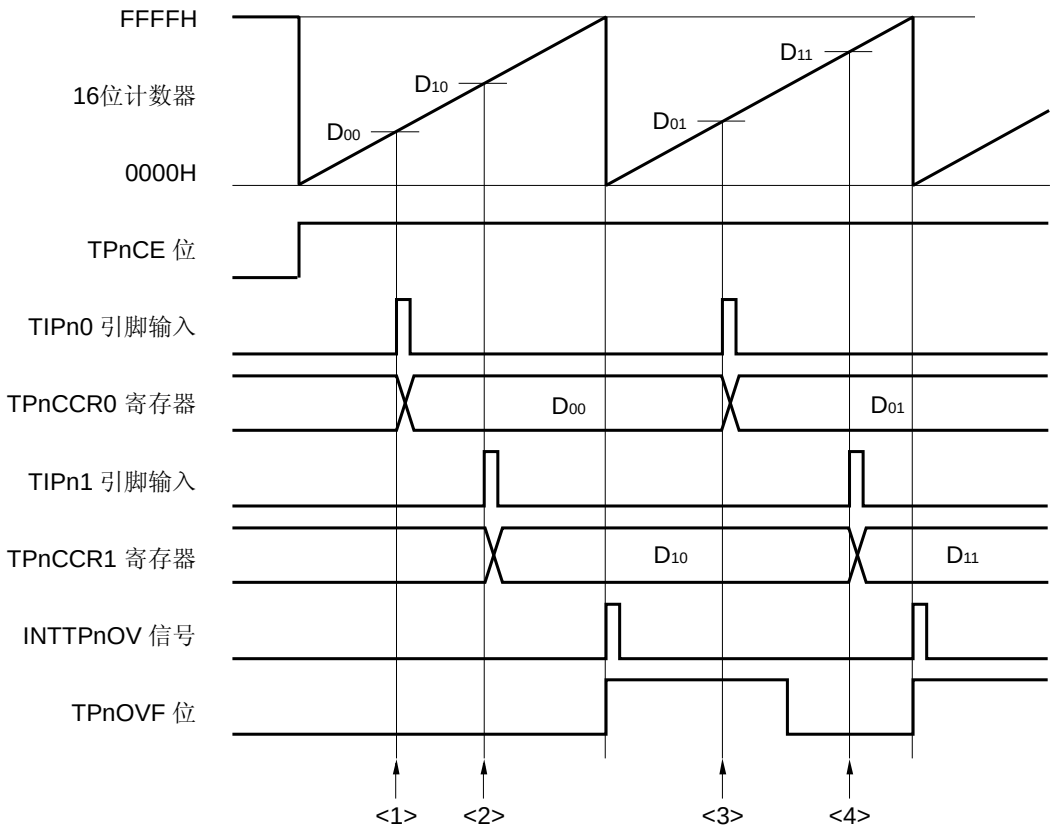
在自由运行定时器模式下执行脉宽测量时，可以用一个通道测量两个脉冲宽度。

要测量脉冲宽度，可以与 INTTPnCCa 信号同步读取 TPnCCRa 寄存器的值，并计算读取值与上次读取值之间的差，得到脉冲宽度。

（c）当使用两个捕获寄存器时对溢出的处理

当使用两个捕获寄存器时必须要格外小心。首先是一个不正确处理的示例，如下所示。

图 7-60. 使用两个捕获寄存器时的不正确处理示例

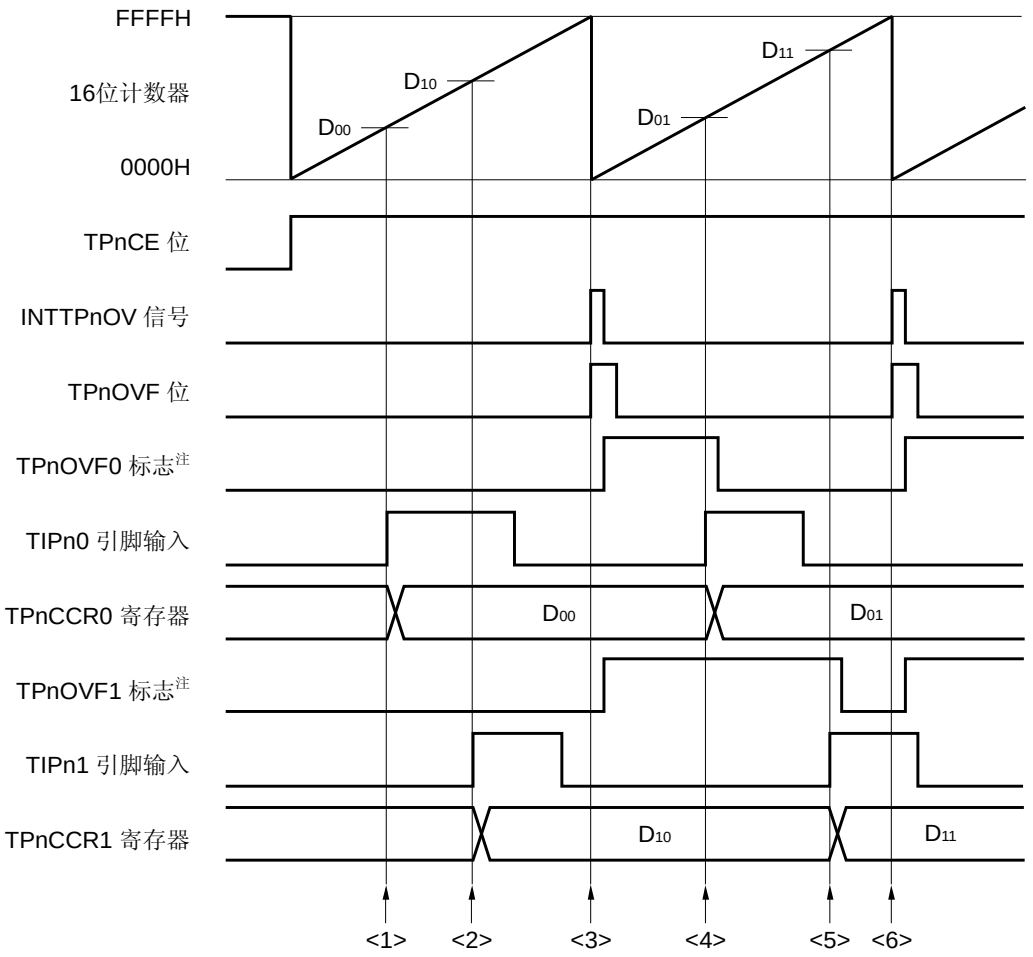


在自由运行定时器模式下测量两个脉冲宽度时，可能出现下列问题。

<1> 读取 TPnCCR0 寄存器（TIPn0 引脚输入的默认值设置）。
<2> 读取 TPnCCR1 寄存器（TIPn1 引脚输入的默认值设置）。
<3> 读取 TPnCCR0 寄存器。
读取 TPnOVF 位，如果 TPnOVF 位为 1，则其被清除为 0。
因为 TPnOVF 位为 1，所以脉冲宽度可以通过（10000H＋D_{01} – D_{00}）计算。
<4> 读取 TPnCCR1 寄存器。
读取 TPnOVF 位。因为此标志在<3>中被清零，所以，读取到 0。
因为 TPnOVF 位为 0，所以脉冲宽度可以用（D_{11} – D_{10}）计算。（不正确）

在使用两个捕获寄存器时，如果溢出标志由一个捕获寄存器清为 0，则另一个捕获寄存器可能得不到正确的脉冲宽度。
该问题可以使用软件解决。关于如何使用软件的示例如下。

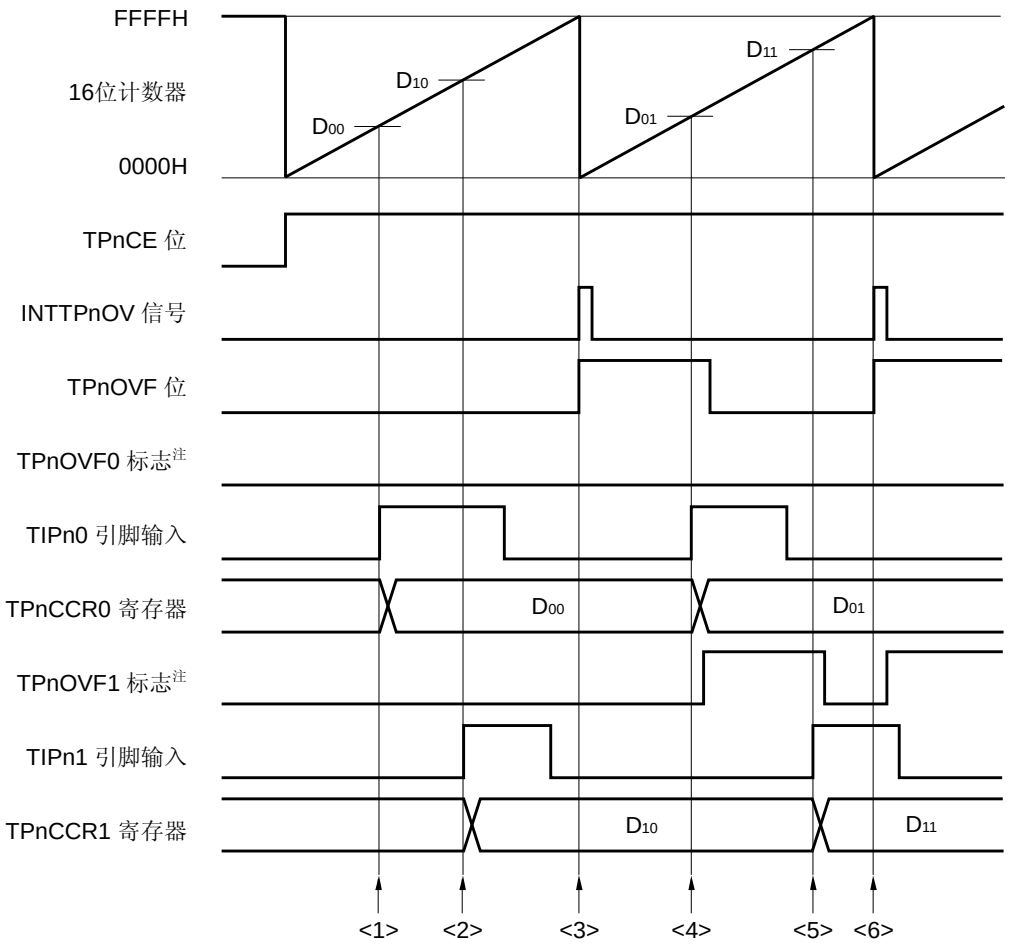
图 7-61. 使用两个捕获寄存器时通过溢出中断解决问题的示例



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

<1> 读取 TPnCCR0 寄存器（设置 TIPn0 引脚输入的默认值）。
<2> 读取 TPnCCR1 寄存器（设置 TIPn1 引脚输入的默认值）。
<3> 发生溢出，在溢出中断服务程序中将 TPnOVF0 和 TPnOVF1 标志设置为 1，且 TPnOVF 位被清除为 0。
<4> 读取 TPnCCR0 寄存器。
读取 TPnOVF0 标志。如果 TPnOVF0 标志为 1，则将其清除为 0。
因为 TPnOVF0 标志为 1，所以脉冲宽度可以通过（10000H＋D_{01} – D_{00}）计算。
<5> 读取 TPnCCR1 寄存器。
读取 TPnOVF1 标志。如果 TPnOVF1 标志为 1，则将其清为 0（TPnOVF0 标志在<4>中已清零，而 TPnOVF1 标志仍然保持为 1）。
因为 TPnOVF1 标志为 1，所以脉冲宽度可以通过（10000H＋D_{11} – D_{10}）计算。（正确）
<6> 与步骤 <3> 相同。

图 7-62. 使用两个捕获寄存器时不通过溢出中断解决问题的示例



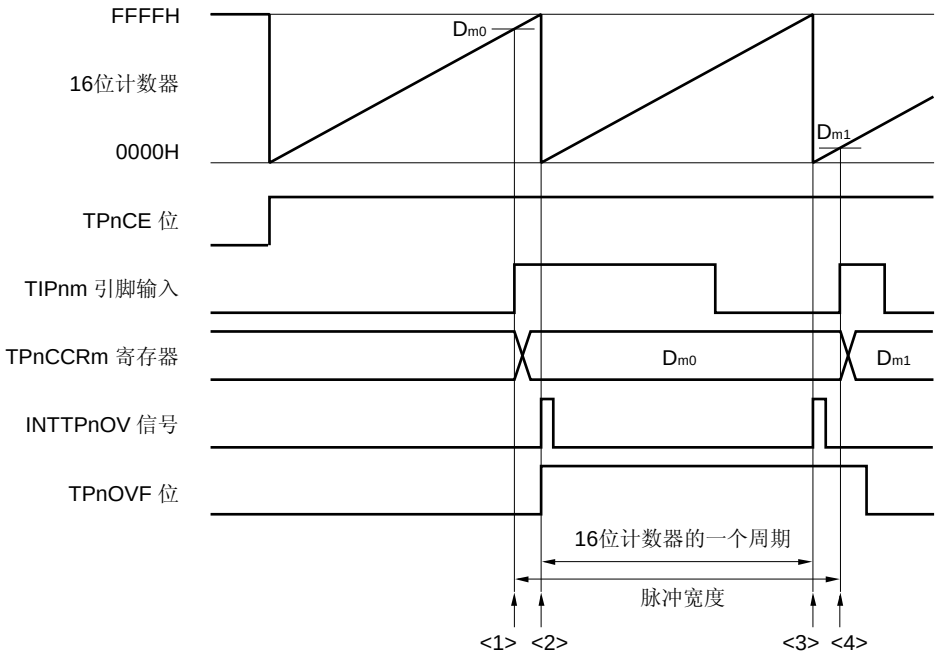
注 由软件对内部 RAM 中的 TPnOVF0 和 TPnOVF1 标志进行设置。

<1> 读取 TPnCCR0 寄存器（设置 TIPn0 引脚输入的默认值）。
<2> 读取 TPnCCR1 寄存器（设置 TIPn1 引脚输入的默认值）。
<3> 发生溢出，不作任何软件处理。
<4> 读取 TPnCCR0 寄存器。
读取 TPnOVF 位。如果 TPnOVF 位为 1，则只把 TPnOVF1 标志置位（1），且 TPnOVF 位被清除为 0。
因为 TPnOVF 位为 1，所以脉冲宽度可以通过（10000H＋D_{01} – D_{00}）计算。
<5> 读取 TPnCCR1 寄存器。
读取 TPnOVF 位。因为 TPnOVF 位在<4>中被清零，所以读取得到的是 0。
读取 TPnOVF1 标志。如果 TPnOVF1 标志为 1，则将其清为 0。
因为 TPnOVF1 标志为 1，所以脉冲宽度可以通过（10000H＋D_{11} – D_{10}）计算。（正确）
<6> 与步骤 <3> 相同。

（d）当捕获触发间隔较长时的溢出处理

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

图 7-63. 捕获触发间隔较长时的不正确处理示例(使用 TIPn0)

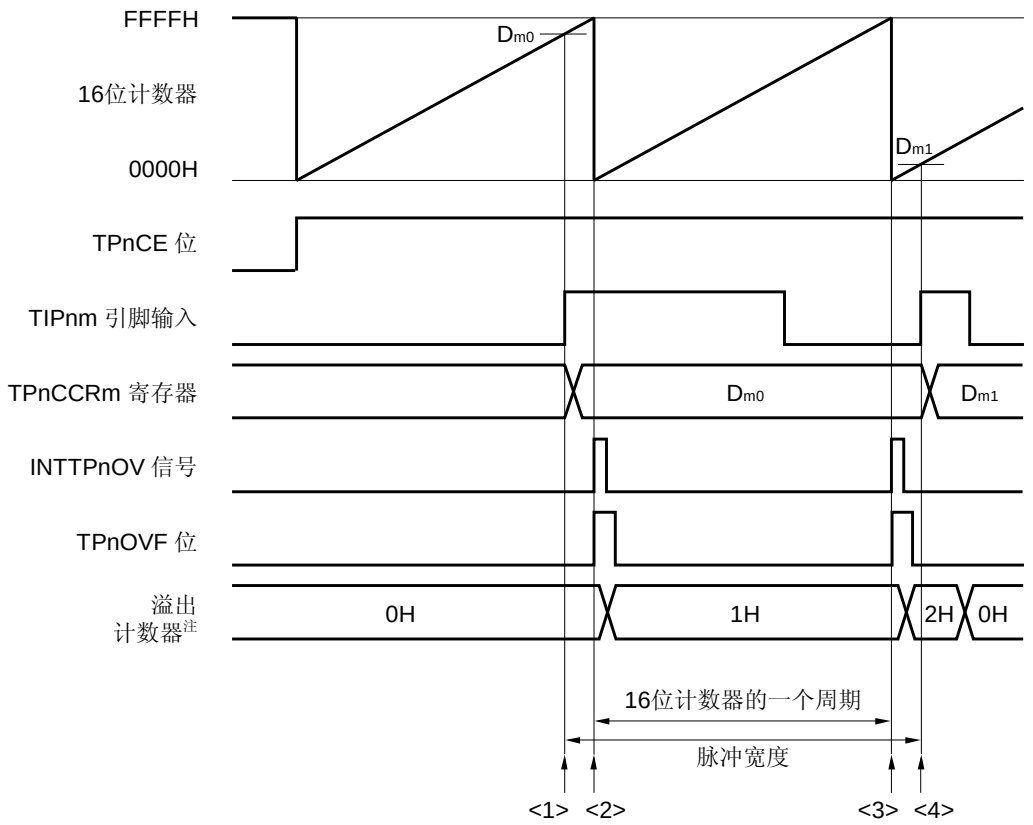


在自由运行定时器模式下测量长脉冲的宽度时，可能出现下列问题。

<1> 读取 TPnCCR0 寄存器（设置 TIPn0 引脚输入的默认值）。
<2> 发生溢出，不作任何软件处理。
<3> 再次发生溢出。不作任何软件处理。
<4> 读取 TPnCCR0 寄存器。
读取 TPnOVF 位。如果 TPnOVF 为 1，将其清除为 0。
因为 TPnOVF 为 1，所以脉冲宽度可以通过（10000H＋D_{a1} – D_{a0}）计算。（不正确）
事实上，脉冲宽度应为（20000H＋D_{a1} – D_{a0}），因为实际已经发生两次溢出。

如果在捕获触发间隔期间，发生两次或更多次的溢出，则无法获得正确的脉冲宽度。
如果捕获触发间隔时间较长时，应该减慢计数时钟以延长 16 位计数器的循环周期，或者使用软件来解决该问题。关于如何使用软件解决该问题，示例如下。

图 7-64. 捕获触发间隔较长时通过软件处理解决问题的示例(使用 TIPn0)



注 内部 RAM 中的溢出计数器由软件设置。

<1> 读取 TPnCCR0 寄存器（设置 TIPn0 引脚输入的默认值）。
<2> 发生溢出，在溢出中断服务程序中增加溢出计数器的值，并将溢出标志 TPnOVF 位清为 0。
<3> 再次发生溢出。在溢出中断服务程序中递增溢出计数器的值，并将溢出标志 TPnOVF 位清为 0。
<4> 读取 TPnCCR0 寄存器。
读取溢出计数器。
→ 当溢出计数器为“N”时，脉冲宽度可用式（N × 10000H＋D_{a1} – D_{a0}）计算。
在本例中，因为溢出实际发生了两次，脉冲宽度计算为（20000H＋D_{a1} – D_{a0}）。
溢出计数器被清除为 0H。

（e）清除溢出标志（TPnOVF）

通过读取 TPnOVF 位可以将溢出标志（TPnOVF）清除为 0。如果它的值为 1，可以通过 CLR1 指令将 TPnOVF 位清为 0，或向 TPnOPT0 寄存器写入 8 位数据（第 0 位是 0）。

7.4.7 脉冲宽度测量模式（TPnMD2 至TPnMD0 位 = 110）

在脉冲宽度测量模式下，当 TPnCTL0.TPnCE 位被置为 1 时，TMPn 开始递增计数。每次检测到有效边沿输入到 TIPna 引脚时，16 位计数器的计数值就存储在 TPnCCRa 寄存器中，且 16 位计数器清零为 0000H。

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

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

备注 **1.** 关于如何设置 TIPn0 和 TIPn1 引脚，参见**表 7-2 TMPn 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。

2. 关于如何使能 INTTPnCC0 和 INTTPnCC0 中断信号，参见**第 22 章 中断服务/异常处理。**

图 7-65. 脉冲宽度测量模式下的 TMPn 配置

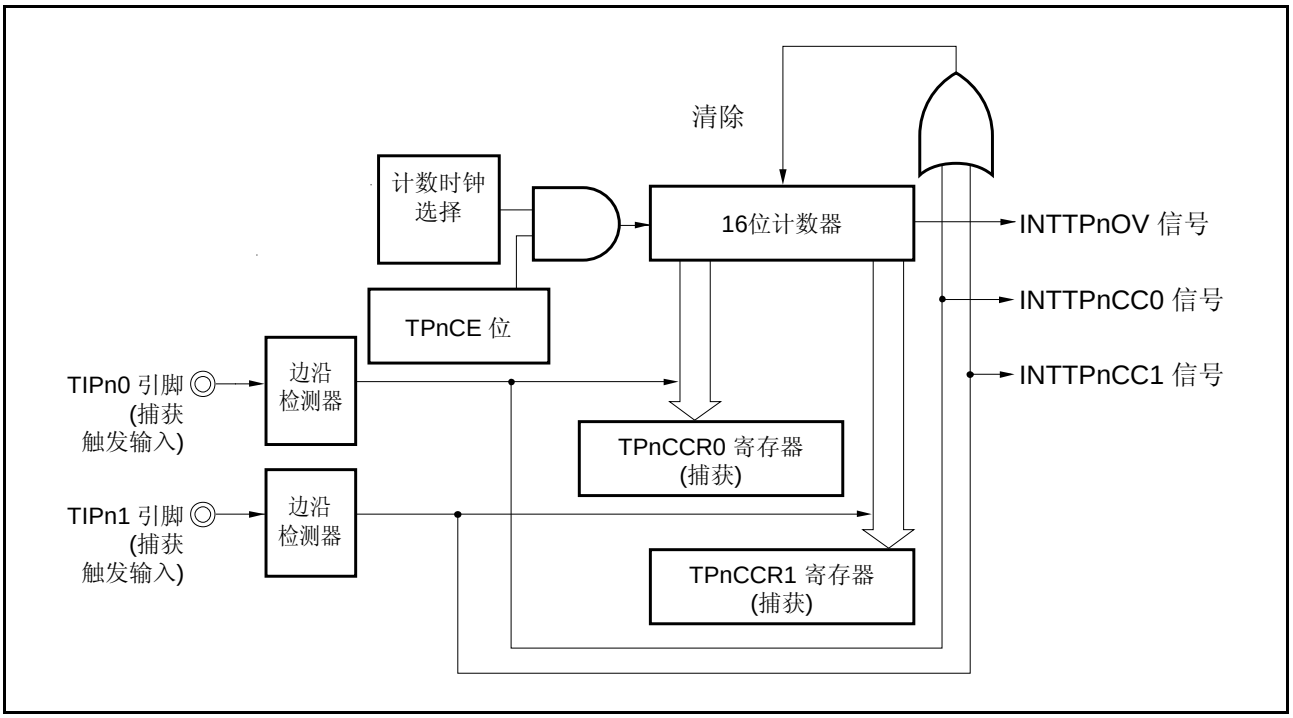
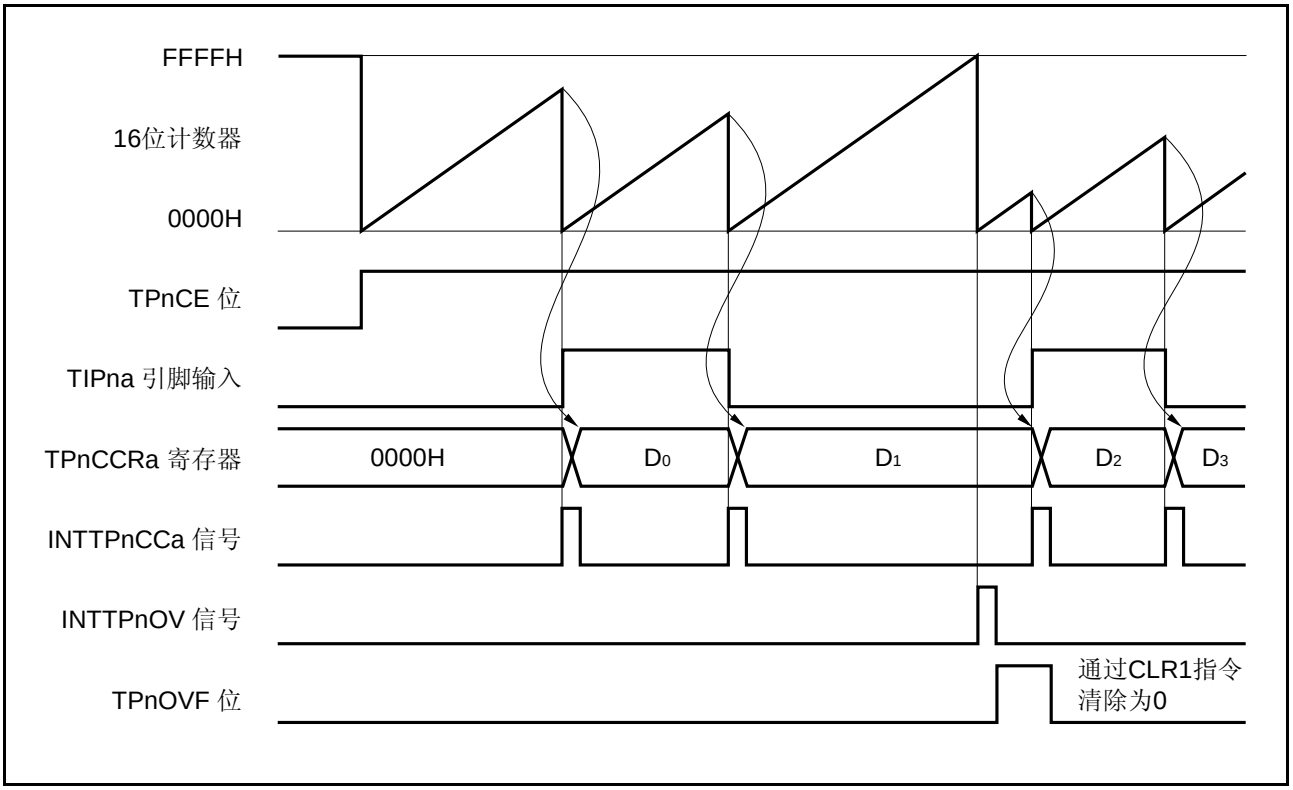


图 7-66. 脉冲宽度测量模式的基本操作时序



当 TPnCE 位被置为 1 时，16 位计数器开始递增计数。然后当检测到信号的有效边沿输入到 TIPna 引脚时，16 位计数器的计数值被存储到 TPnCCRa 寄存器中，16 位计数器清零为 0000H，并产生一个捕获中断请求信号（INTTPnCCa）。

脉冲宽度的计算如下。

脉冲宽度＝捕获值 × 计数时钟周期

当 16 位计数器计数到 FFFFH 时，即使没有有效边沿输入 TIPna 引脚，在下一个计数时钟时，会产生溢出中断请求信号（INTTPnOV），且计数器清零为 0000H，然后计数器继续递增计数。此时，溢出标志（TPnOPT0.TPnOVF 位）仍被置为 1。使用软件执行 CLR1 指令将溢出标志清为 0。

若溢出标志被设置为 1，则脉冲宽度的计算如下。

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

图 7-67. 脉冲宽度测量模式时的寄存器设置

（a）TMPn 控制寄存器 0（TPnCTL0）

	TPnCE					TPnCKS2	TPnCKS1	TPnCKS0
TPnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

选择计数时钟

0: 停止计数
1: 允许计数

（b）TMPn 控制寄存器 1（TPnCTL1）

		TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0	0	0	0	1	1	0

1, 1, 0:
脉宽测量模式

（c）TMPn 输入/输出控制寄存器 1（TPnIOC1）

					TPnIS3	TPnIS2	TPnIS1	TPnIS0
TPnIOC1	0	0	0	0	0/1	0/1	0/1	0/1

选择TIPn0引脚
输入的有效沿

选择TIPn1引脚
输入的有效沿

（d）TMPn 选项寄存器 0（TPnOPT0）

			TPnCCS1	TPnCCS0				TPnOVF
TPnOPT0	0	0	0	0	0	0	0	0/1

溢出标志

（e）TMPn 计数器读取缓冲寄存器（TPnCNT）

通过读取该存器，可以读出 16 位计数器的值。

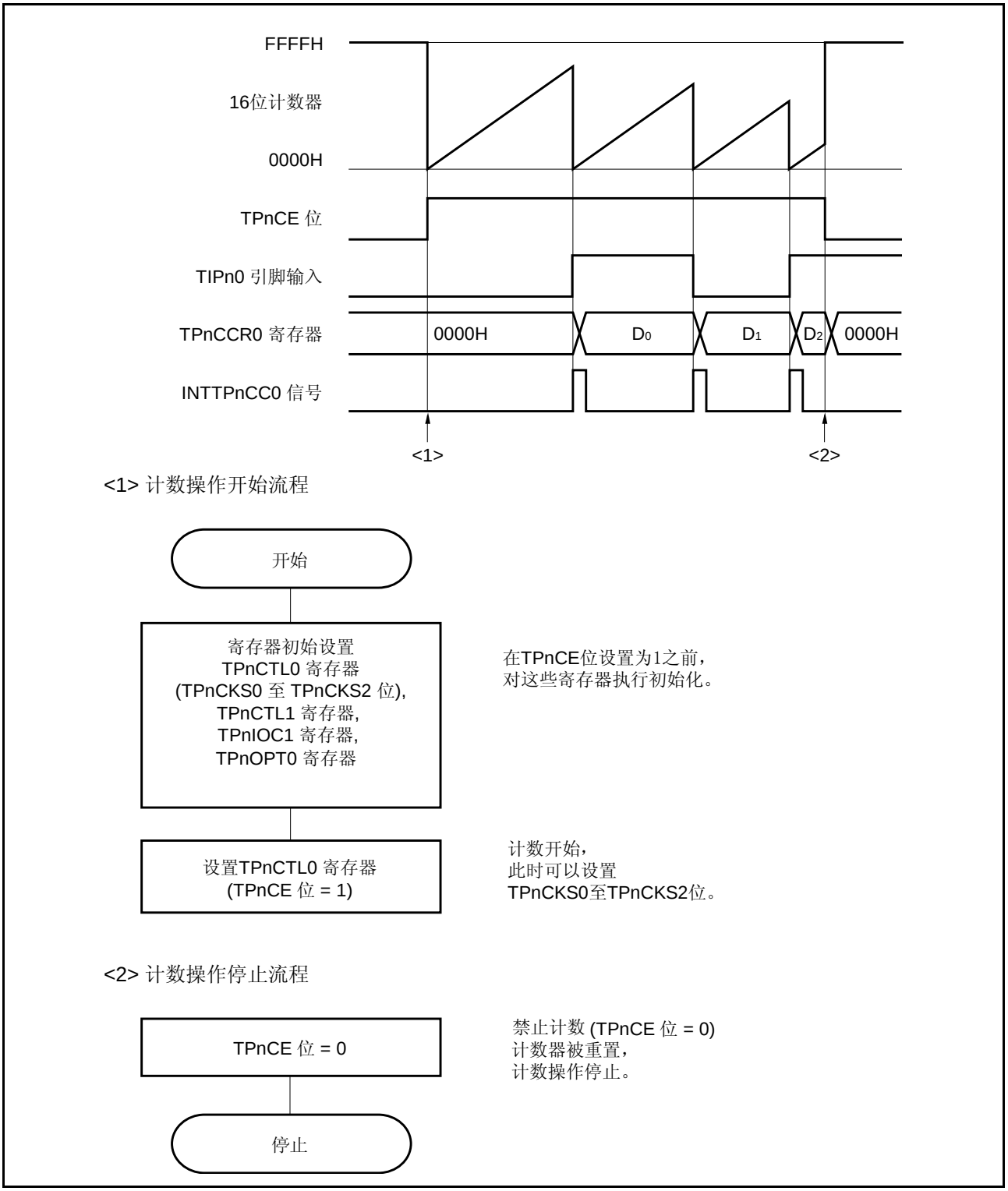
（f）TMPn 捕获/比较寄存器 0 和 1（TPnCCR0 和 TPnCCR1）

当检测到 TIPn0/TIPn1 引脚输入的有效边沿时，这些寄存器保存 16 位计数器的当前计数值。

备注 在脉冲宽度测量模式下，不使用 TMPn 输入/输出控制寄存器 0（TPnIOC0）和 TMPn 输入/输出控制寄存器 2（TPnIOC2）。

（1）脉宽测量模式的操作

图 7-68. 脉宽测量模式时的时序和操作处理



（2）使用脉宽测量模式

（a）清除溢出标志（TPnOVF）

通过读取 TPnOVF 位可以将溢出标志（TPnOVF）清除为 0。如果它的值为 1，可以通过 CLR1 指令将 TPnOVF 位清为 0，或向 TPnOPT0 寄存器写入 8 位数据（第 0 位是 0）。

7.4.8 定时器输出操作

下表显示了 TOPn0 和 TOPn1 引脚的操作和输出电平。

表 7-6. 各种模式下的定时器输出控制

操作模式	TOPn1 引脚	TOPn0 引脚
间隔计时器模式	方波输出	
外部事件计数模式	–	
外部触发脉冲输出模式	外部触发脉冲输出	方波输出
单次触发脉冲输出模式	单次触发脉冲输出	
PWM 输出模式	PWM（脉宽调制信号）输出	
自由定时器模式	方波输出（只有使用比较功能时可用）	
脉冲宽度测量模式	–	

表 7-7. 定时器输出控制位对 TOPn0 和 TOPn1 引脚影响的真值表

TPnIOC0.TPnOLa 位	TPnIOC0.TPnOEa 位	TPnCTL0.TPnCE 位	TOPnm 引脚电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前为低电平，开始计数后为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平，开始计数后为低电平

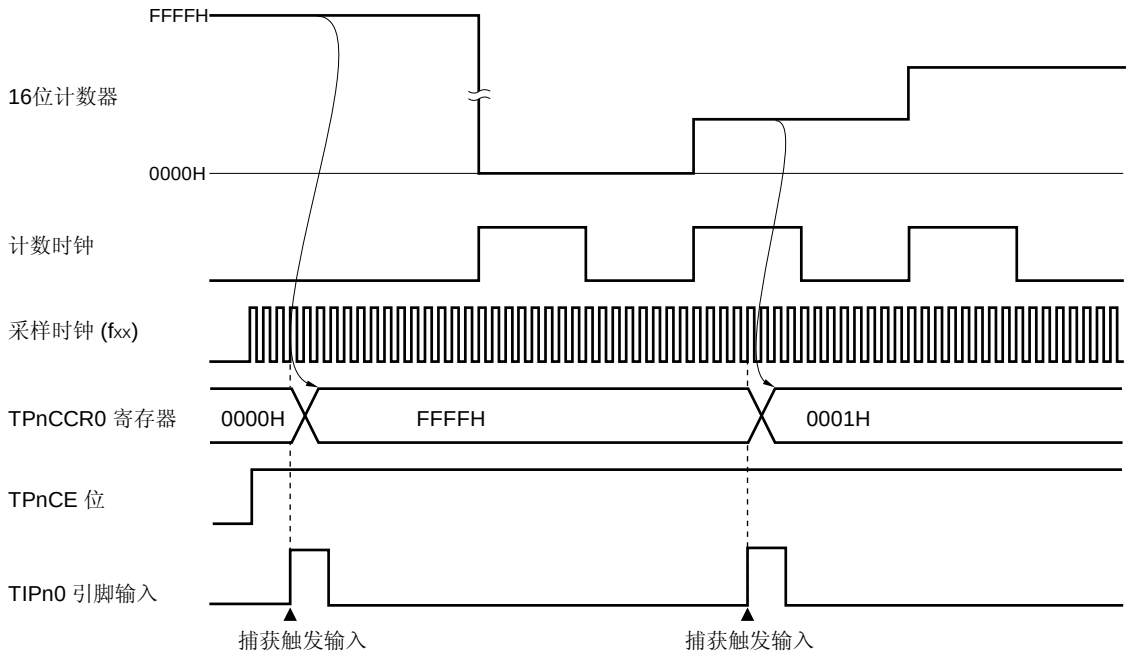
7.5 注意事项

（1）捕获操作

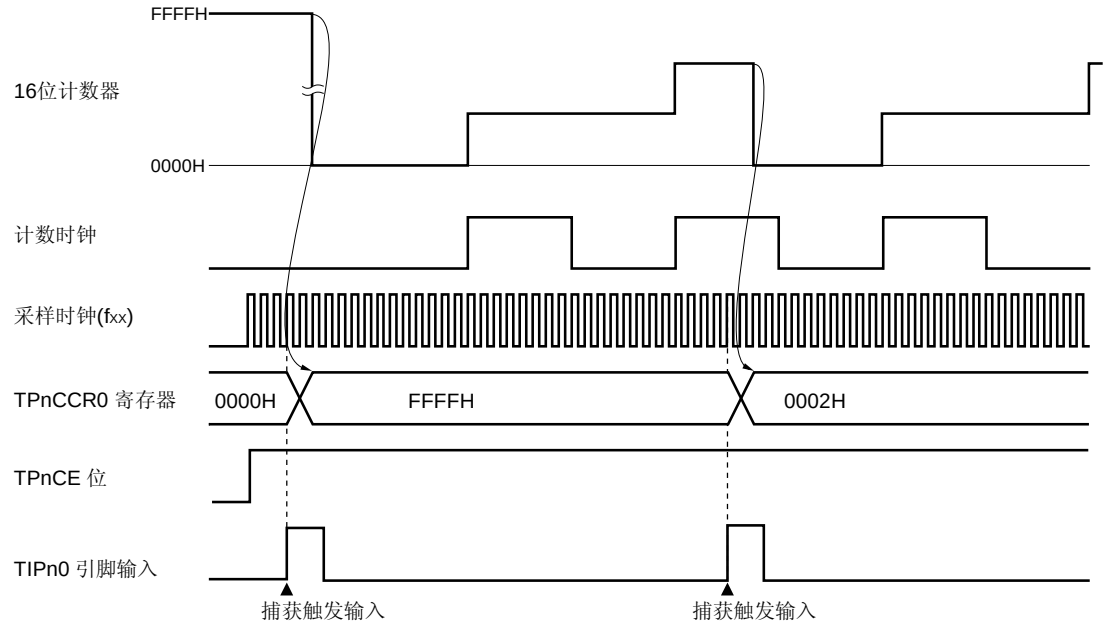
当使用捕获操作，并且选择 fxx/8, fxx/16, fxx/32, fxx/64, fxx/128, fxx/256, 或 fxx/512 作为计数时钟时，如果在 TPnCE 位置 1 之后立即有捕获触发输入，那么捕获到 TPnCCR0 和 TPnCCR1 寄存器中的是可能是 FFFFH 而不是 0000H，或者捕获操作可能无法执行（不产生捕获中断）。

当使用捕获操作且选择外部事件计数输入作为计数时钟时，在没有外部事件计数输入的期间，也有可能发生该问题。

（a）自由定时器模式



（b）脉宽测量模式



第八章 16 位定时器/事件计数器Q（TMQ）

定时器 Q（TMQ）是 16 位定时器/事件计数器。
V850ES/JG3-L 单片机内置一路 TMQ 定时器/计数器，TMQ0。

8.1 概述

TMQ0 具有下列特征：

(1) **间隔定时器**
TMQ0 按照预置的间隔产生一个中断，可以输出一个方波。

(2) **外部事件计数器**
TMQ0 对从外部输入的信号脉冲数量进行计数。

(3) **外部触发脉冲输出**
当指定的外部信号输入时，TMQ0 开始计数并输出一个脉冲。

(4) **单次触发脉冲输出**
TMQ0 输出一个单次触发脉冲，输出宽度可以自由指定。

(5) **PWM 输出**
TMQ0 输出一个固定宽度的脉冲，有效电平宽度可以改变。
即使在定时器工作时，脉冲占空比也可以自由改变。

(6) **自由运行定时器**
TMQ0 从 0000H 递增到 FFFFH，然后重置。

(7) **脉冲宽度测量**
TMQ0 可以用来测量外部输入信号的脉冲。

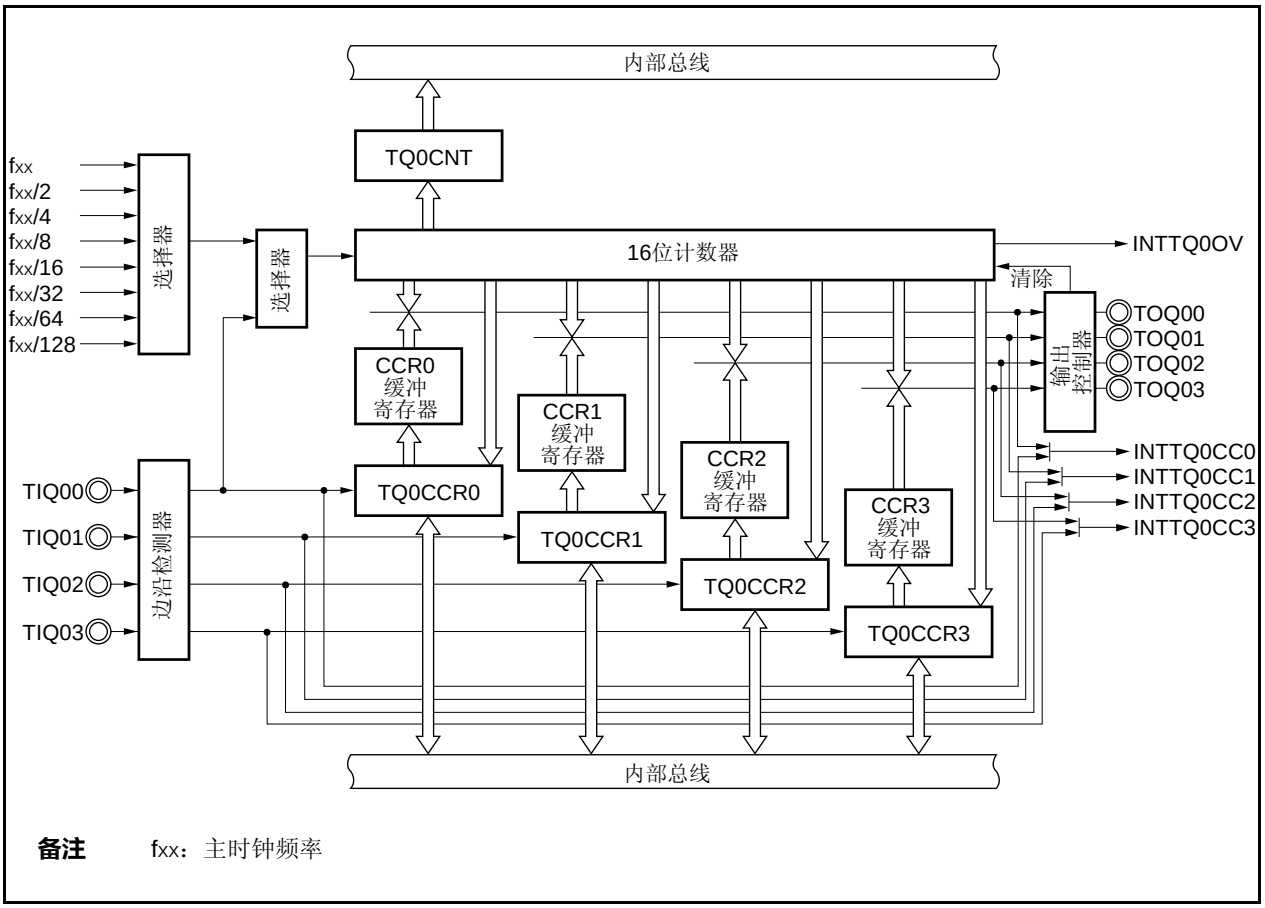
8.2 配置

TMQ0 包括下列硬件：

表 8-1. TMQ0 的配置

项目	配置
寄存器	16 位计数器 TMQ0 捕获/比较寄存器 0 至 3（TQ0CCR0 至 TQ0CCR3） TMQ0 计数器读取缓冲寄存器（TQ0CNT） CCR0 至 CCR3 缓冲寄存器 TMQ0 控制寄存器 0，1（TQ0CTL0，TQ0CTL1） TMQ0 输入/输出控制寄存器 0 至 2（TQ0IOC0 至 TQ0IOC2） TMQ0 选项寄存器 0（TQ0OPT0）
定时器输入	4（TIQ00 至 TIQ03 引脚）
定时器输出	5（TOQ00 至 TOQ03 引脚）

图 8-1. TMQ0 框图



备注 fxx：主时钟频率

（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。

该计数器的计数值可以通过读取 TQ0CNT 寄存器得到。

当 TQ0CTL0.TQ0CE 位为 0 时，计数器停止，计数器的值为 FFFFH。如果此时读取 TQ0CNT 寄存器，则读取得到的值为 0000H 。

系统复位将 TQ0CE 位清 0，停止计数器，并将计数器的值设置为 FFFFH。

（2）CCR0 缓冲寄存器

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

当 TQ0CCR0 寄存器用作比较寄存器时，写入 TQ0CCR0 寄存器的值会传送到 CCR0 缓冲寄存器。如果 16 位计数器的计数值与 CCR0 缓冲寄存器的值相匹配时，就会产生一个比较匹配中断请求信号（INTTQ0CC0）。

CCR0 缓冲寄存器不能直接读写。

CCR0 缓冲寄存器在复位后清零为 0000H，因为 TQ0CCR0 寄存器清零为 0000H。

（3）CCR1 缓冲寄存器

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

当 TQ0CCR1 寄存器用作比较寄存器时，写入 TQ0CCR1 寄存器的值会传送到 CCR1 缓冲寄存器。如果 16 位计数器的计数值和 CCR1 缓冲寄存器的值匹配时，就会产生一个比较匹配中断请求信号（INTTQ0CC1）。

CCR1 缓冲寄存器不能直接读写。

CCR1 缓冲寄存器在复位后清零为 0000H，因为 TQ0CCR1 寄存器清零为 0000H。

（4）CCR2 缓冲寄存器

CCR2 缓冲寄存器是一个 16 位比较寄存器，用于比较 16 位计数器的计数值。

当 TQ0CCR2 寄存器用作比较寄存器时，写入 TQ0CCR2 寄存器的值会传送到 CCR2 缓冲寄存器。如果 16 位计数器的计数值和 CCR2 缓冲寄存器的值匹配时，就会产生一个比较匹配中断请求信号（INTTQ0CC2）。

CCR2 缓冲寄存器不能直接读写。

CCR2 缓冲寄存器在复位后清零为 0000H，因为 TQ0CCR2 寄存器清零为 0000H。

（5）CCR3 缓冲寄存器

CCR3 缓冲寄存器是一个 16 位比较寄存器，用于比较 16 位计数器的计数值。

当 TQ0CCR3 寄存器用作比较寄存器时，写入 TQ0CCR3 寄存器的值会传送到 CCR3 缓冲寄存器。如果 16 位计数器的计数值与 CCR3 缓冲寄存器的值相匹配时，就会产生一个比较匹配中断请求信号（INTTQ0CC3）。

CCR3 缓冲寄存器不能直接读写。

CCR3 缓冲寄存器在复位后清零为 0000H，因为 TQ0CCR3 寄存器清零为 0000H。

（6）边沿检测器

该电路检测输入到 TIQ00 至 TIQ03 引脚信号的有效边沿。通过设置 TQ0IOC1 和 TQ0IOC2 寄存器，可以选择各种有效沿：无边沿、上升沿、下降沿、上升/下降双边沿作为有效沿。

（7）输出控制器

该电路控制 TOQ00 至 TOQ03 引脚的输出。输出控制器由 TQ0IOC0 寄存器控制。

（8）选择器

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

8.2.1 TMQ0 使用的引脚

TMQ0 使用的输入和输出引脚如表 8-2 所示。当这些引脚用于 TMQ0 时，首先设置它们的端口模式。
详情参见**表 4-15 端口引脚用作复用功能时的设置**。

表 8-2. TMQ0 使用的引脚

引脚编号		端口	TMQ0 输入	TMQ0 输出	复用功能
GC	F1				
40	L8	P53	TIQ00[Note]	TOQ00	SIB2/KR3/RTP03/DDO
37	L7	P50	TIQ01	TOQ01	KR0/RTP00
38	K7	P51	TIQ02	TOQ02	KR1/RTP01
39	J7	P52	TIQ03	TOQ03	KR2/RTP02/DDI

8.2.2 中断

TMQ0 使用下列五种类型的中断信号：

(1) INTTQ0CC0
当 16 位计数器的值与 CCR0 缓冲寄存器的值匹配时，或有捕获信号输入到 TIQ00 引脚时，产生该信号。

(2) INTTQ0CC1
当 16 位计数器的值与 CCR1 缓冲寄存器的值匹配时，或有捕获信号输入到 TIQ01 引脚时，产生该信号。

(3) INTTQ0CC2
当 16 位计数器的值与 CCR2 缓冲寄存器的值匹配时，或有捕获信号输入到 TIQ02 引脚时，产生该信号。

(4) INTTQ0CC3
当 16 位计数器的值与 CCR3 缓冲寄存器的值匹配时，或有捕获信号输入到 TIQ03 引脚时，产生该信号。

(5) INTTQ0OV
当 16 位计数器的值递增达到 FFFFH 之后溢出，则产生该信号。

8.3 寄存器

以下寄存器控制 TMQ0：

- TMQ0 控制寄存器 0（TQ0CTL0）
- TMQ0 控制寄存器 1（TQ0CTL1）
- TMQ0 输入/输出控制寄存器 0（TQ0IOC0）
- TMQ0 输入/输出控制寄存器 1（TQ0IOC1）
- TMQ0 输入/输出控制寄存器 2（TQ0IOC2）
- TMQ0 选项寄存器 0（TQ0OPT0）
- TMQ0 捕获/比较寄存器 0（TQ0CCR0）
- TMQ0 捕获/比较寄存器 1（TQ0CCR1）
- TMQ0 捕获/比较寄存器 2（TQ0CCR2）
- TMQ0 捕获/比较寄存器 3（TQ0CCR3）
- TMQ0 计数器读取缓冲寄存器（TQ0CNT）

备注 当使用 TIQ00 至 TIQ03 和 TOQ00 至 TOQ03 引脚的功能时，参见**表 4-15 端口引脚用作可选功能时的设置**。

（1）TMQ0 控制寄存器 0（TQ0CTL0）

TQ0CTL0 寄存器是 8 位寄存器，用于控制定时器 TMQ0 的操作。
该寄存器可以按字节读写或按位读写。
复位输入将该寄存器设为 00H。
随时可以用软件将相同的值写入 TQ0CTL0 寄存器。

复位后： 00H R/W 地址：FFFFF540H

	<7>	6	5	4	3	2	1	0
TQ0CTL0	TQ0CE	0	0	0	0	TQ0CKS2	TQ0CKS1	TQ0CKS0

TQ0CE	TMQ0 操作控制
0	禁止TMQ0 操作。停止工作时钟的供给。 (TMQ0 异步复位[注])
1	使能TMQ0操作。开始工作时钟的供给。 (TMQ0 开始工作)

TQ0CKS2	TQ0CKS1	TQ0CKS0	内部计数时钟选择
0	0	0	f_{XX}
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/8$
1	0	0	$f_{XX}/16$
1	0	1	$f_{XX}/32$
1	1	0	$f_{XX}/64$
1	1	1	$f_{XX}/128$

注 TQ0OPT0.TQ0OVF 位与 16 位计数器被同时复位。
并且，当 16 位计数器被复位时，定时器输出引脚（TOQ00 至 TOQ03 引脚）被重置为 TQ0IOC0 寄存器设置的状态。

注意事项 **1. 当 TQ0CE 位 = 0 时，设置 TQ0CKS2 至 TQ0CKS0 位为“1”。**
当 TQ0CE 位的值从 0 变为 1 的同时，可以设定 TQ0CKS2 至 TQ0CKS0 位。
2. 请确保将第 3 位至第 6 位清除为“0”。

备注 f_{XX}：主时钟频率

（2）TMQ0 控制寄存器 1（TQ0CTL1）

TQ0CTL1 寄存器是 8 位寄存器，用于控制定时器 TMQ0 的操作。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF541H

	7	<6>	<5>	4	3	2	1	0
TQ0CTL1	0	TQ0EST	TQ0EEE	0	0	TQ0MD2	TQ0MD1	TQ0MD0

TQ0EST	软件触发控制
0	-
1	为外部触发输入产生一个有效的信号。 · 在单次脉冲输出模式： 将1写入TQ0EST位作为触发，输出一个单次脉冲。 · 在外部触发脉冲输出模式： 将1写入TQ0EST位作为触发，输出一个PWM脉冲。

TQ0EEE	计数时钟选择
0	禁止使用外部事件计数输入操作。 (对由TQ0CTL0.TQ0CK0至 TQ0CK2位所选的计数时钟进行计数)
1	允许使用外部事件计数输入操作。 (对外部事件计数输入信号的有效沿进行计数)
TQ0EEE位选择使用内部时钟计数还是外部事件计数输入的有效沿进行计数。	

TQ0MD2	TQ0MD1	TQ0MD0	定时器模式选择
0	0	0	内部定时器模式
0	0	1	外部事件计数模式
0	1	0	外部触发脉冲输出模式
0	1	1	单次触发脉冲输出模式
1	0	0	PWM 输出模式
1	0	1	自由运行定时器模式
1	1	0	脉冲宽度测量模式
1	1	1	设置禁止

注意事项

1. 只有在外触发脉冲输出模式或单次触发脉冲输出模式下，TQ0EST 位有效。在其它模式下，向该位写入 1 的操作都被忽略。

2. 不论 TQ0EEE 位的取值如何，在外部事件计数模式下，都会选择外部事件输入。

3. 在停止定时器（将 TQ0CTL0.TQ0CE 位设置为 0）之后，设置 TQ0EEE 和 TQ0MD2 至 TQ0MD0 位（但是，如果 TQ0CE 位为 1 时，可以写入相同的值）。如果在 TQ0CE 位为 1 时重写 TQ0EEE 和 TQ0MD2 至 TQ0MD0 位，则操作无法保证，应把 TQ0CE 位清除为 0，再重新写入这些位。

4. 请确保将第 3 位，第 4 位和第 7 位清除为“0”。

（3）TMQ0 输入/输出控制寄存器 0（TQ0IOC0）

TQ0IOC0 寄存器是 8 位寄存器，用于控制定时器的输出（TOQ00 至 TOQ03 引脚）。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后： 00H R/W 地址:FFFFF542H

	7	<6>	5	<4>	3	<2>	1	<0>
TQ0IOC0	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0

TQ0OLm	TOQ0m 引脚输出电平设置 (m = 0 至 3)注
0	TOQ0m 引脚开始输出高电平
1	TOQ0m 引脚开始输出低电平

TQ0OEm	TOQ0m 引脚输出设置 (m = 0 至 3)
0	禁止定时器输出 • 当 TQ0OLm 位 = 0：从TOQ0m 引脚输出低电平 • 当 TQ0OLm 位 = 1：从TOQ0m 引脚输出高电平
1	使能定时器输出 (从TOQ0m引脚输出一个脉冲)

注 由 TQ0OLm 位指定的定时器输出引脚（TOQ0m）的输出电平如下所示。

· 当TQ0OLm 位 = 0时

· 当TQ0OLm 位 = 1时

注意事项 **1. 当 TQ0CTL0.TQ0CE 位＝0 时，重新写入 TQ0OLm 和 TQ0OEm 位（当 TQ0CE 位＝1 时，可以写入相同值）。如果重新写入不能正确执行，将 TQ0CE 位清除为 0，然后重新设定该位。**

2. 即使在 TQ0CE 和 TQ0OEm 位均为 0 时对 TQ0OLm 位进行操作，TOQ0m 引脚的输出电平也会改变。

备注 m＝0 至 3

（4）TMQ0 输入/输出控制寄存器 1（TQ0IOC1）

TQ0IOC1 寄存器是 8 位寄存器，用于控制捕获触发输入信号（TIQ00 至 TIQ03 引脚）的有效边沿。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF543H

	7	6	5	4	3	2	1	0
TQ0IOC1	TQ0IS7	TQ0IS6	TQ0IS5	TQ0IS4	TQ0IS3	TQ0IS2	TQ0IS1	TQ0IS0

TQ0IS7	TQ0IS6	捕获触发输入信号(TIQ03 引脚) 有效沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

TQ0IS5	TQ0IS4	捕获触发输入信号(TIQ02 引脚) 有效沿检测
0	0	无边沿检测 (捕获操作无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

TQ0IS3	TQ0IS2	捕获触发输入信号(TIQ01 引脚) 有效沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

TQ0IS1	TQ0IS0	捕获触发输入信号(TIQ00 引脚) 有效沿设置
0	0	无边沿检测 (捕获操作无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

注意事项

1. **当 TQ0CTL0.TQ0CE 位 = 0 时，重新写入 TQ0IS7 至 TQ0IS0 位（当 TQ0CE 位 = 1 时，可以写入相同值）。如果重新写入不能正确执行，将 TQ0CE 位清除为 0，然后重新进行位的设定。**
2. **只有在自由运行定时器模式和脉冲宽度测量模式下，TQ0IS7 至 TQ0IS0 位才有效。在所有其它模式下，无法进行捕获操作。**

（5）TMQ0 输入/输出控制寄存器 2（TQ0IOC2）

TQ0IOC2 寄存器是 8 位寄存器，用于控制外部事件计数器输入信号（TIQ00 引脚）和外部触发输入信号（TIQ00 引脚）的有效边沿。

该寄存器可按字节字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF544H

	7	6	5	4	3	2	1	0
TQ0IOC2	0	0	0	0	TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0

TQ0EES1	TQ0EES0	外部事件计数输入信号(TIQ00 引脚) 有效沿设置
0	0	无边沿检测 (外部事件计数无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

TQ0ETS1	TQ0ETS0	外部触发输入信号(TIQ00 引脚) 有效沿设置
0	0	无边沿检测 (外部触发无效)
0	1	上升沿检测
1	0	下降沿检测
1	1	双边沿检测

注意事项

1. **当 TQ0CTL0.TQ0CE 位 = 0 时，重新写入 TQ0EES1，TQ0EES0，TQ0ETS1 和 TQ0ETS0 位（当 TQ0CE 位 = 1，可以写入相同的值）。如果重新写入不能正确执行，将 TQ0CE 位清除为 0，然后重新进行位的设定。**
2. **只有当 TQ0CTL1.TQ0EEE 位 = 1 或设置为外部事件计数模式（TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 = 001）时，TQ0EES1 和 TQ0EES0 位有效。**
3. **只有在外部触发脉冲输出模式（TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 010）或在单次触发脉冲输出模式（TQ0CTL1.TQ0MD2 至 TQ0CTL1.TQ0MD0 位 = 011）时，TQ0ETS1 和 TQ0ETS0 位有效。**

（6）TMQ0 选项寄存器 0（TQ0OPT0）

TQ0OPT0 寄存器为 8 位寄存器，用来设置捕获/比较操作，并用于检测计数器的溢出。

该寄存器可按字节读写或按位读写。

复位输入将该寄存器设为 00H。

复位后: 00H R/W 地址: FFFFF545H

	7	6	5	4	3	2	1	<0>
TQ0OPT0	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0	0	0	0	TQ0OVF

TQ0CCSm	TQ0CCRm寄存器捕获/比较选择
0	比较寄存器选择
1	捕获寄存器选择
TQ0CCSm位的设置仅在自由运行定时器模式下有效。	

TQ0OVF	TMQ0溢出检测
置 (1)	发生溢出
复位 (0)	TQ0OVF位写入0或TQ0CTL0.TQ0CE位 = 0

- 自由运行定时器模式下或脉冲宽度测量模式下，当16位计数器的计数值由FFFFH变为0000H发生溢出时，TQ0OVF位置位。
- 在TQ0OVF 位被置为1的同时，产生中断请求信号(INTTQ0OV) 。只有在自由运行定时器模式和脉冲宽度测量模式下，才会产生INTTQ0OV 信号。
- 即使当TQ0OVF 位 = 1时读取TQ0OVF位或TQ0OPT0 寄存器，TQ0OVF位也不清零。
- TQ0OVF位可读取也可写入，但是TQ0OVF位不能使用软件置1，写入1对TMQ0的操作没有影响。

注意事项 **1. 当 TQ0CTL0.TQ0CE 位= 0 时，重新写入 TQ0CCS3 至 TQ0CCS0 位（当 TQ0CE 位= 1 时，可以写入相同的值）。如果重新写入不能正确执行，将 TQ0CE 位清除为 0 ，然后重新进行位的设定。**

2. 请确保将第 1 位至第 3 位清除为“0”。

备注 m＝0 至 3

（7）TMQ0 捕获/比较寄存器 0（TQ0CCR0）

根据不同工作模式，TQ0CCR0 寄存器可以用作捕获寄存器或比较寄存器 。

在自由计时器模式下，根据 TQ0OPT0.TQ0CCS0 位的设定，该寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TQ0CCR0 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

TQ0CCR0 寄存器可以在运行中进行读取和写入。

该寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清除为 0000H。

注意事项 在以下状态禁止访问 TQ0CCR0 寄存器。而且，如果系统处于等待状态，取消等待状态的唯一方式就是执行复位。详细说明，参见 3.4.9（1）访问片上外设输入/输出专用寄存器。

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

复位后: 0000H R/W 地址: FFFFF546H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CCR0																

（a）用作比较寄存器时的功能

即便在 TQ0CTL0.TQ0CE 位＝1 时，TQ0CCR0 寄存器仍可以重新写入。

TQ0CCR0 寄存器的设定值传送给 CCR0 缓冲寄存器。当 16 位计数器的值与 CCR0 缓冲寄存器的值相匹配时，就会生成一个比较匹配中断请求信号（INTTQ0CC0）。如果此时 TOQ00 引脚输出允许，则 TOQ00 引脚的输出会反转（详情参见各种操作模式的描述）。

在间隔定时器模式、外部事件计数器模式、外部触发脉冲输出模式、单次触发脉冲输出模式或 PWM 输出模式时，当 TQ0CCR0 寄存器作为周期寄存器时，如果 16 位计数器的计数值与 CCR0 缓冲寄存器数值匹配，则计数器清零为（0000H）。

（b）用作捕获寄存器时的功能

在自由运行定时器模式下，当 TQ0CCR0 寄存器用作捕获寄存器时，若检测到捕获触发输入引脚（TIQ00 引脚）的有效边沿，则 16 位计数器的当前计数值存储于 TQ0CCR0 寄存器中。在脉宽测量模式下，若检测到捕获触发输入引脚（TIQ00 引脚）的有效边沿，16 位计数器的当前计数值就储存在 TQ0CCR0 寄存器中，且 16 位计数器清零（0000H）。

即使捕获操作和读取 TQ0CCR0 寄存器的操作发生冲突，还是可以正确读取 TQ0CCR0 寄存器的值。

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

表 8-3. 捕获/比较寄存器在各种模式下的功能以及如何写入比较寄存器

操作模式	捕获/比较寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	随时写入
外部事件计数器	比较寄存器	随时写入
外部触发脉冲输出	比较寄存器	批量写入
单次触发脉冲输出	比较寄存器	随时写入
PWM 输出	比较寄存器	批量写入
自由运行定时器	捕获/比较寄存器	随时写入
脉冲宽度测量	捕获寄存器	–

备注 关于随时写入和批量写入的详情，参见 **8.4 (2) 随时写入和批量写入**。

（8）TMQ0 捕获/比较寄存器 1（TQ0CCR1）

根据不同工作模式，TQ0CCR1 寄存器可以用作捕获寄存器或比较寄存器 。

在自由计时器模式下，根据 TQ0OPT0.TQ0CCS1 位的设定，该寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TQ0CCR1 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

TQ0CCR1 寄存器可以在运行中进行读取和写入。

该寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清除为 0000H。

注意事项 在以下状态禁止访问 TQ0CCR1 寄存器。而且，如果系统处于等待状态，取消等待状态的唯一方式就是执行复位。详细说明，参见 3.4.9（1）访问片上外设输入/输出专用寄存器。

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

复位后: 0000H R/W 地址: FFFFF548H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CCR1																

（a）用作比较寄存器时的功能

即便在 TQ0CTL0.TQ0CE 位= 1 时，TQ0CCR1 寄存器仍可以重新写入。

TQ0CCR1 寄存器的设定值传送给 CCR1 缓冲寄存器。当 16 位计数器的值与 CCR1 缓冲寄存器的值相匹配时，就会生成一个比较匹配中断请求信号（INTTQ0CC1）。如果此时 TOQ01 引脚输出允许，则 TOQ01 引脚的输出会反转（详情参见各种操作模式的描述）。

（b）用作捕获寄存器时的功能

在自由运行定时器模式下，当 TQ0CCR1 寄存器用作捕获寄存器时，若检测到捕获触发输入引脚（TIQ01 引脚）的有效边沿，则 16 位计数器的当前计数值存储于 TQ0CCR1 寄存器中。在脉宽测量模式下，若检测到捕获触发输入引脚（TIQ01 引脚）的有效边沿，16 位计数器的当前计数值就储存在 TQ0CCR1 寄存器中，且 16 位计数器清零（0000H）。

即使捕获操作和读取 TQ0CCR1 寄存器的操作发生冲突，还是可以正确读取 TQ0CCR1 寄存器的值。

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

表 8-4. 捕获/比较寄存器在各种模式下的功能以及如何写入比较寄存器

操作模式	捕获/比较寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	随时写入
外部事件计数器	比较寄存器	随时写入
外部触发脉冲输出	比较寄存器	批量写入
单次触发脉冲输出	比较寄存器	随时写入
PWM 输出	比较寄存器	批量写入
自由运行定时器	捕获/比较寄存器	随时写入
脉冲宽度测量	捕获寄存器	–

备注 关于随时写入和批量写入的详情，参见 **8.4 (2) 随时写入和批量写入**。

（9）TMQ0 捕获/比较寄存器 2（TQ0CCR2）

根据不同工作模式，TQ0CCR2 寄存器可以用作捕获寄存器或比较寄存器 。

在自由计时器模式下，根据 TQ0OPT0.TQ0CCS2 位的设定，该寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TQ0CCR2 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

TQ0CCR2 寄存器可以在运行中进行读取和写入。

该寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清除为 0000H。

注意事项 在以下状态禁止访问 TQ0CCR2 寄存器。而且，如果系统处于等待状态，取消等待状态的唯一方式就是执行复位。详细说明，参见 3.4.9（1）访问片上外设输入/输出专用寄存器。

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

复位后: 0000H R/W 地址: FFFFF54AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CCR2																

（a）用作比较寄存器时的功能

即便在 TQ0CTL0.TQ0CE 位= 1 时，TQ0CCR2 寄存器仍可以重新写入。

TQ0CCR2 寄存器的设定值传送给 CCR2 缓冲寄存器。当 16 位计数器的计数值和 CCR2 缓冲寄存器的值匹配时，也会产生一个比较匹配中断请求信号（INTTQ0CC2）。如果此时 TOQ02 引脚输出允许，则 TOQ02 引脚的输出会反转（详情参见各种操作模式的描述）。

（b）用作捕获寄存器时的功能

在自由运行定时器模式下，当 TQ0CCR2 寄存器用作捕获寄存器时，若检测到捕获触发输入引脚（TIQ02 引脚）的有效边沿，则 16 位计数器的当前计数值存储于 TQ0CCR2 寄存器中。在脉宽测量模式下，若检测到捕获触发输入引脚（TIQ02 引脚）的有效边沿，16 位计数器的当前计数值就储存在 TQ0CCR2 寄存器中，且 16 位计数器清零（0000H）。

即使捕获操作和读取 TQ0CCR2 寄存器的操作发生冲突，还是可以正确读取 TQ0CCR2 寄存器的值。

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

表 8-5. 捕获/比较寄存器在各种模式下的功能以及如何写入比较寄存器

操作模式	捕获/比较寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	随时写入
外部事件计数器	比较寄存器	随时写入
外部触发脉冲输出	比较寄存器	批量写入
单次触发脉冲输出	比较寄存器	随时写入
PWM 输出	比较寄存器	批量写入
自由运行定时器	捕获/比较寄存器	随时写入
脉冲宽度测量	捕获寄存器	–

备注 关于随时写入和批量写入的详情，参见 **8.4 (2) 随时写入和批量写入**。

（10）TMQ0 捕获/比较寄存器 3（TQ0CCR3）

根据不同工作模式，TQ0CCR3 寄存器可以用作捕获寄存器或比较寄存器 。

在自由计时器模式下，根据 TQ0OPT0.TQ0CCS3 位的设定，该寄存器可用作捕获寄存器或比较寄存器。在脉宽测量模式中，TQ0CCR3 寄存器只可以被用作捕获寄存器。而在其它模式时，该寄存器只可以用作比较寄存器。

TQ0CCR3 寄存器可以在运行中进行读取和写入。

该寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清除为 0000H。

注意事项 在以下状态禁止访问 TQ0CCR3 寄存器。而且，如果系统处于等待状态，取消等待状态的唯一方式就是执行复位。详细说明，参见 3.4.9（1）访问片上外设输入/输出专用寄存器。

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

复位后: 0000H R/W 地址: FFFFF54CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CCR3																

（a）用作比较寄存器时的功能

即便在 TQ0CTL0.TQ0CE 位= 1 时，TQ0CCR3 寄存器仍可以重新写入。

TQ0CCR3 寄存器的设定值传送给 CCR3 缓冲寄存器。当 16 位计数器的计数值与 CCR3 缓冲寄存器的值相匹配时，就会生成一个比较匹配中断请求信号（INTTQ0CC3）。如果此时 TOQ03 引脚输出允许，则 TOQ03 引脚的输出会反转（详情参见各种操作模式的描述）。

（b）用作捕获寄存器时的功能

在自由运行定时器模式下，当 TQ0CCR3 寄存器用作捕获寄存器时，若检测到捕获触发输入引脚（TIQ03 引脚）的有效边沿，则 16 位计数器的当前计数值存储于 TQ0CCR3 寄存器中。在脉宽测量模式下，若检测到捕获触发输入引脚（TIQ03 引脚）的有效边沿，16 位计数器的当前计数值就储存在 TQ0CCR3 寄存器中，且 16 位计数器清零（0000H）。

即使捕获操作和读取 TQ0CCR3 寄存器的操作发生冲突，还是可以正确读取 TQ0CCR3 寄存器的值。

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

表 8-6. 捕获/比较寄存器在各种模式下的功能以及如何写入比较寄存器

操作模式	捕获/比较寄存器	如何写入比较寄存器
间隔定时器	比较寄存器	随时写入
外部事件计数器	比较寄存器	随时写入
外部触发脉冲输出	比较寄存器	批量写入
单次触发脉冲输出	比较寄存器	随时写入
PWM 输出	比较寄存器	批量写入
自由运行定时器	捕获/比较寄存器	随时写入
脉冲宽度测量	捕获寄存器	–

备注 关于随时写入和批量写入的详情，参见 **8.4 (2) 随时写入和批量写入**。

（11）TMQ0 计数器读取缓冲寄存器（TQ0CNT）

TQ0CNT 寄存器是读出缓冲寄存器，可以借此读取 16 位计数器的计数值。

如果在 TQ0CTL0.TQ0CE 位＝1 时读取该寄存器，可以读出 16 位定时器的计数值。

该寄存器为只读寄存器，按 16 位宽度读取。

当 TQ0CE 位＝0 时，TQ0CNT 寄存器的值被清除为 0000H。如果此时读取 TQ0CNT 寄存器，则不会读出 16 位计数器的值（FFFFH），读到的值为 0000H 。

在复位后因为 TQ0CE 位被清除为 0，TQ0CNT 寄存器的值被清除为 0000H。

注意事项 在以下状态禁止访问 TQ0CNT 寄存器。而且，如果系统处于等待状态，取消等待状态的唯一方式就是执行复位。详细说明，参见 3.4.9（1）访问片上外设输入/输出专用寄存器。

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

复位后: 0000H R 地址: FFFFF54EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CNT																

8.4 操作

定时器 TMQ0 可以执行如下操作。

表 8-7. TMQ0 操作模式

操作模式	TQ0CTL1.TQ0EST 位（软件触发位）	TIQ00 引脚（外部触发输入）	捕获/比较寄存器设置	比较寄存器写入	计数时钟
间隔计时器模式	无效	无效	仅作比较寄存器	随时写入	内部/外部
外部事件计数器模式注 1	无效	无效	仅作比较寄存器	随时写入	外部
外部触发脉冲输出模式注 2	有效	有效	仅作比较寄存器	批量写入	内部
单次触发脉冲输出模式注 2	有效	有效	仅作比较寄存器	随时写入	内部
PWM 输出模式	无效	无效	仅作比较寄存器	批量写入	内部/外部
自由定时器模式	无效	无效	可以切换	随时写入	内部/外部
脉宽测量模式注 2	无效	无效	仅作捕获寄存器	无法写入	内部

注 **1.** 在用作外部事件计数器模式下，应该指定对 TIQ00 引脚捕获触发输入的有效边沿不作检测（将 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位都清除为 0）。

2. 使用外部触发脉冲输出模式、单次触发脉冲输出模式和脉冲宽度测量模式时，选择内部时钟作为计数时钟。（将 TQ0CTL1.TQ0EEE 位清除为 0）。

(1) 计数器基本操作

16 位计数器的基本操作描述如下。更多细节，参见各种操作模式的描述。

(a) 开始计数

在所有模式下，TMQ0 从 FFFFH 开始计数，然后递增：FFFFH, 0000H, 0001H, 0002H, 0003H….

(b) 清除 TMQ0

当 TMQ0 的值与比较寄存器的值匹配时，或在有效捕获触发信号输入时 TMQ0 的值被捕获，TMQ0 被清除为 0000H。

请注意，当 TMQ0 开始计数后，它立即从 FFFFH 递增到 0000H 并有一个溢出，这并不意味着 TMQ0 已经被清除。于是，这种情况下不会产生 INTTQ0CCm 中断（m = 0 至 3）。

(c) 溢出

在自由运行定时器模式和脉宽测量模式下，当 TMQ0 从 FFFFH 递增到 0000H 之后，发生溢出。溢出会将 TQ0OPT0.TQ0OVF 位设置为 1，并产生一个中断请求信号 (INTTQ0OV)。请注意，在下列情况时，不能保证 INTTQ0OV：

- TMQ0 刚开始计数。
- 用于清除 TMQ0 的比较器对比值被指定为 FFFFH。
- 在脉宽测量模式下，当 TMQ0 的 FFFFH 值被捕获而导致 TMQ0 被清除之后，从 FFFFH 递增到 0000H。

注意事项 在发生 INTTQ0OV 溢出中断请求信号之后，请务必确认溢出标志(TQ0OVF)被设置为 1。

(d) 正在递增时读取 TMQ0

TMQ0 正在递增时可以通过 TQ0CNT 寄存器来读取。

具体来说，当 TQ0CLT0.TQ0CE 位为 1 时，TMQ0 的值可以通过 TQ0CNT 寄存器来读取。但是，请注意，当 TQ0CLT0.TQ0CE 位为 0 时，TMQ0 的值总是 FFFFH 且 TQ0CNT 寄存器的值总是 0000H。

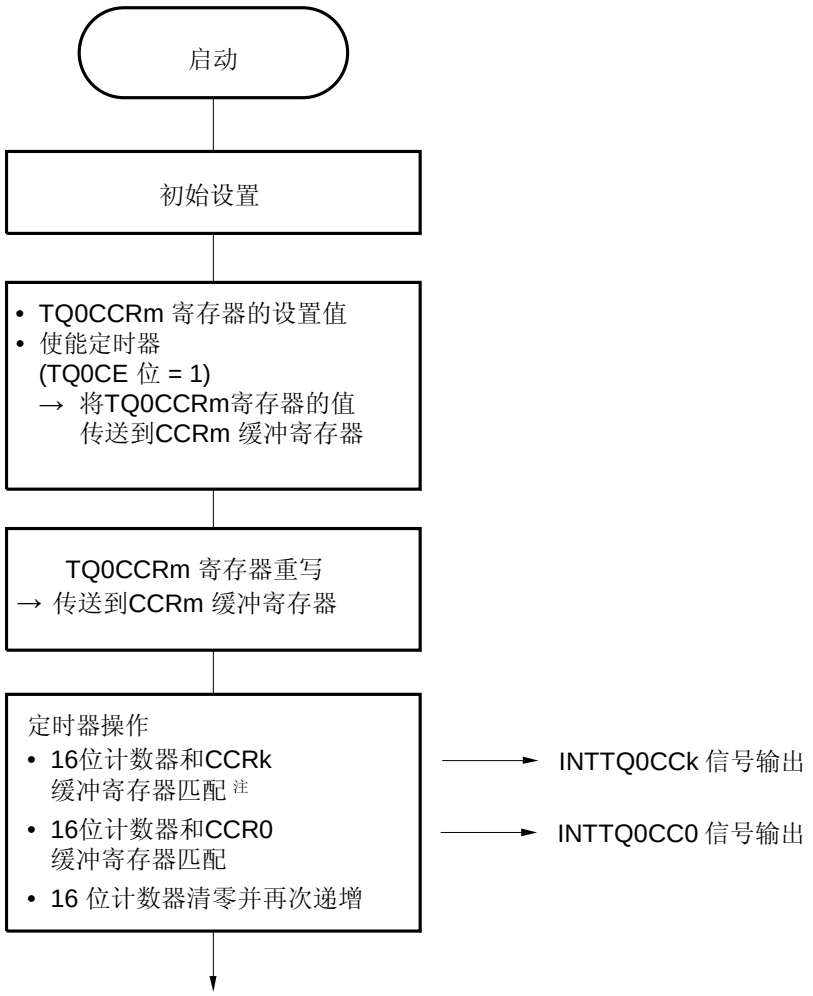
(2) 随时写入和批量写入

在 TMQ0 定时器操作期间（也就是说 TQ0CTL0.TQ0CE 位为 1 时），可以写入 TQ0CCR0 至 TQ0CCR3 寄存器，但是 CCR0 至 CCR3 缓冲寄存器的写入方法取决于不同的模式。两种写入方式就是随时写入和批量写入。

(a) 随时写入

在定时器操作期间的任意时刻，数据可以从 TQ0CCR0 至 TQ0CCR3 寄存器传送到 CCR0 至 CCR3 缓冲寄存器。

图 8-2. 随时写入的基本操作流程图



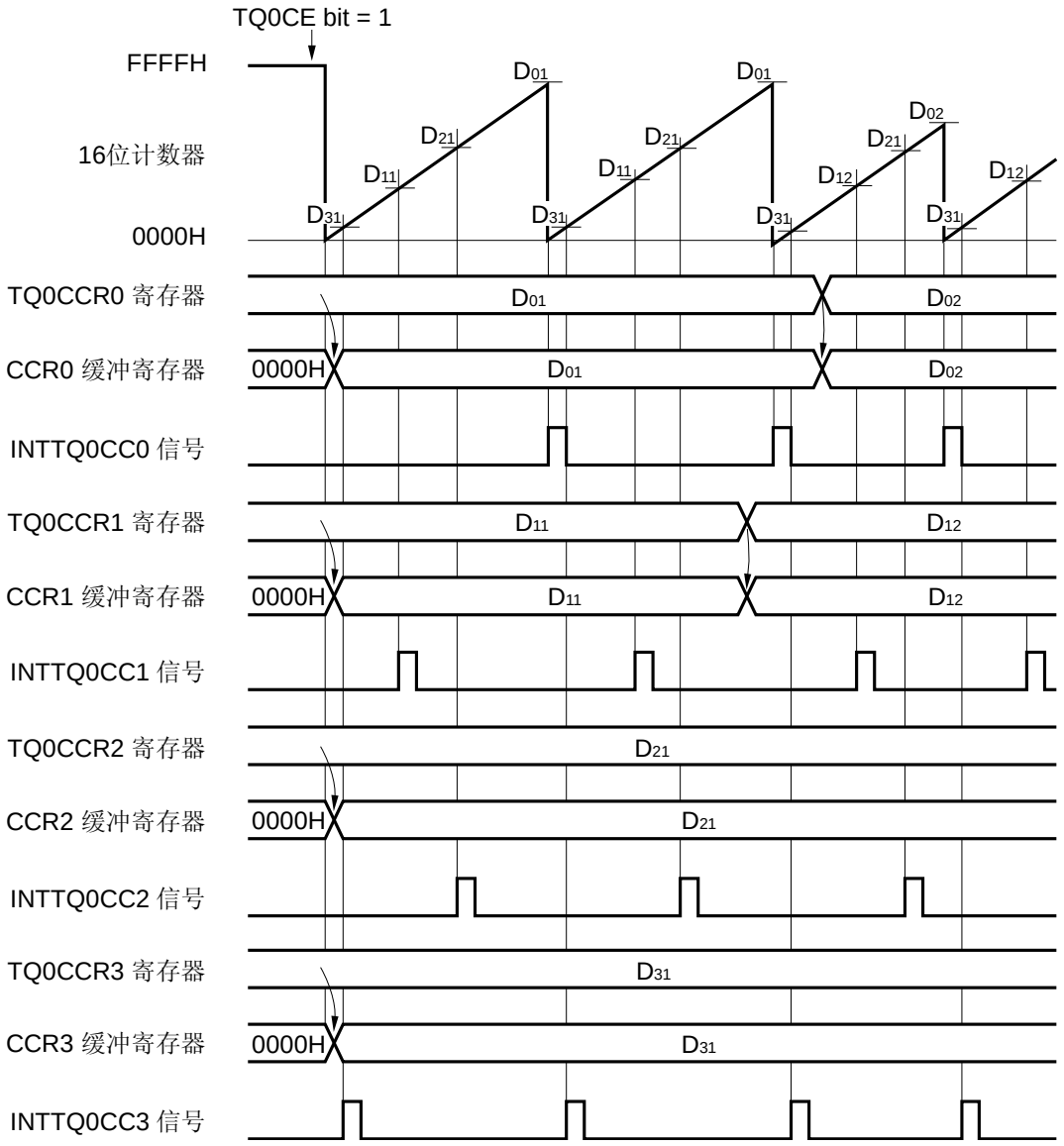
注 只有在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配时，才清除 16 位计数器。与 CCRk 缓冲寄存器的值匹配，不清除。

备注 **1.** 上述流程图适用于 TMQ0 用作间隔定时器时。

2. k = 1 to 3

m = 0 to 3

图 8-3. 随时写入的时序示例



备注 **1.** D_{01}, D_{02}：TQ0CCR0 寄存器的设置值
D_{11}, D_{12}：TQ0CCR1 寄存器的设置值
D_{21}：TQ0CCR2 寄存器的设置值
D_{31}：TQ0CCR3 寄存器的设置值

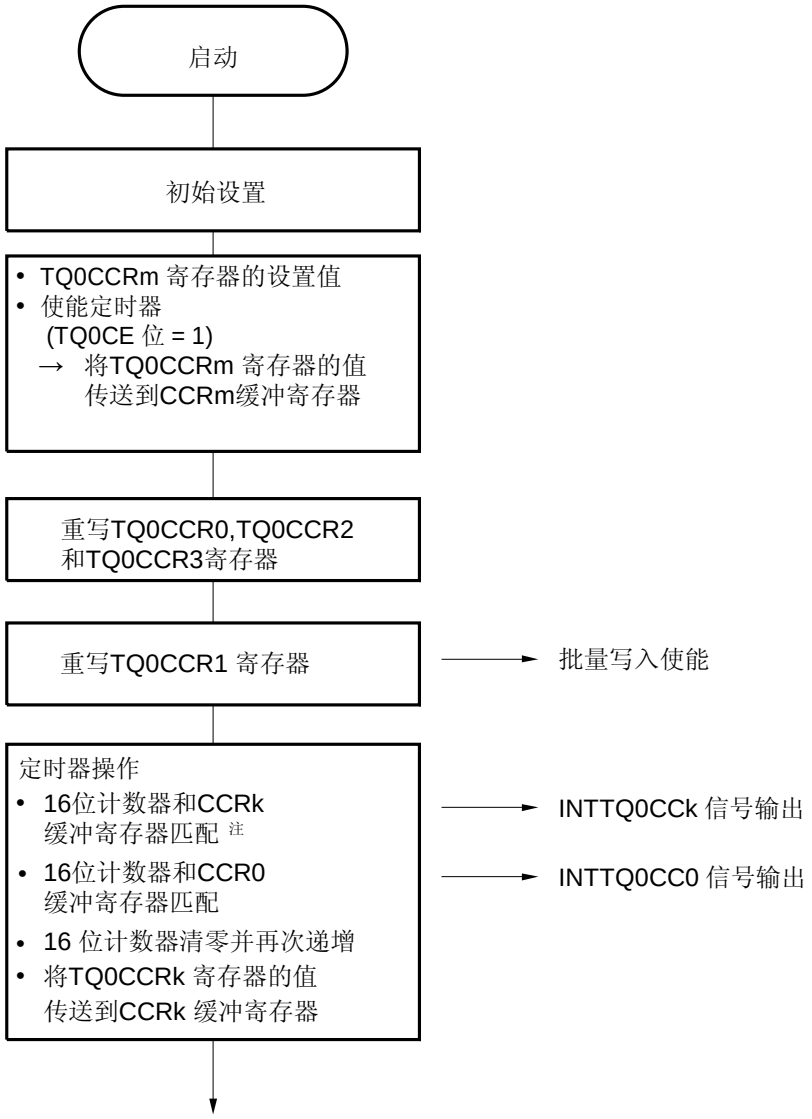
2. 上述流程图适用于 TMQ0 用作间隔定时器时使用。

(b) 批量写入

该模式下，在 TMQ0 定时器操作期间，数据一次性从 TQ0CCR0 至 TQ0CCR3 寄存器传送到 CCR0 至 CCR3 缓冲寄存器。在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配时，传送该数据。写入 TQ0CCR1 寄存器就可以使能传送。是否写入 TQ0CCR1 寄存器可以控制使能或禁止下一次传送时序。

为了让重写的 TQ0CCR0 至 TQ0CCR3 寄存器设置值成为 16 位计数器比较值（换句话说，为了该值被传送到 CCR0 至 CCR3 缓冲寄存器），必须在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配之前写入 TQ0CCR0 寄存器。然后重写 TQ0CCR1 寄存器。因此，在 16 位计数器的值和 CCR0 缓冲寄存器的值匹配时，TQ0CCR0 至 TQ0CCR3 寄存器的值被传送到 CCR0 至 CCR3 缓冲寄存器。所以，即使只想重写 TQ0CCR0 寄存器的值，也需要将相同的值（和 TQ0CCR1 寄存器的当前值相同）写入 TQ0CCR1 寄存器。

图 8-4. 批量写入的基本操作流程图

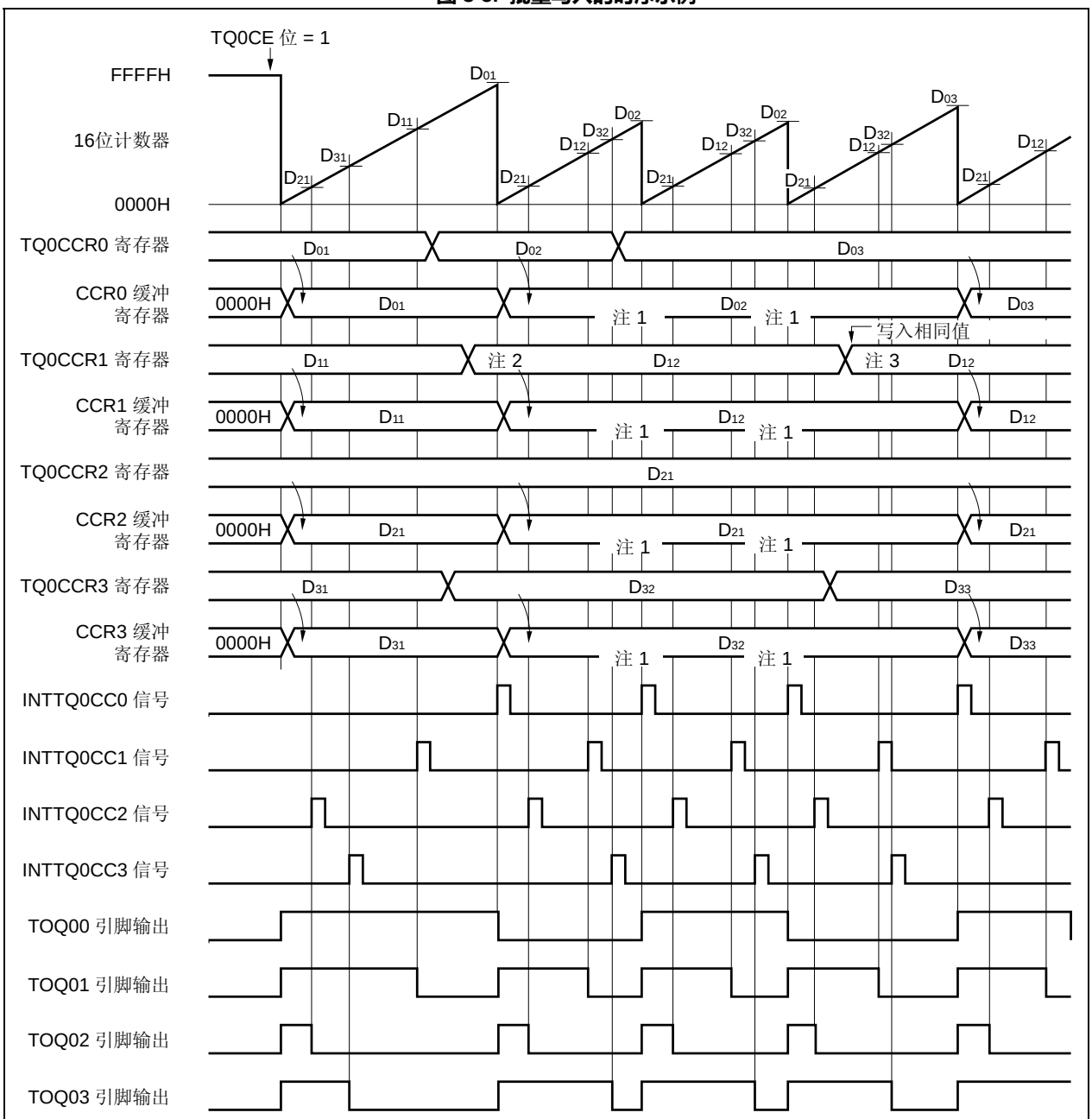


注 只有在 16 位计数器的值和 CCRk 缓冲寄存器的值匹配时，才清除 16 位计数器。与 CCR0 缓冲寄存器的值匹配，不清除。

注意事项 TQ0CCR1 寄存器的写入处理包括批量写入的使能。于是，请在重写 TQ0CCR0, TQ0CCR2 和 TQ0CCR3 寄存器之后，再重写 TQ0CCR1 寄存器。

备注
1. 上述流程图适用于 TMQ0 用作 PWM 输出模式时。
2. **k = 1 to 3**
   m = 0 to 3

图 8-5. 批量写入的时序示例



注 **1.** 因为 TQ0CCR1 寄存器没有重写，D_{02} 没有被传送。

2. 因为 TQ0CCR1 寄存器已经被重写(D_{12})，在 16 位计数器的值和 TQ0CCR0 寄存器的值(D_{01})匹配时，D_{12} 数据被传送到 CCR1 缓冲寄存器 。

3. 因为 TQ0CCR1 寄存器已经被重写(D_{12})，在 16 位计数器的值和 TQ0CCR0 寄存器的值(D_{12})匹配时，D_{12} 数据被传送到 CCR1 缓冲寄存器 。

备注 **1.** D_{01}, D_{02}, D_{03}：TQ0CCR0 寄存器的设置值
D_{11}, D_{12}：TQ0CCR1 寄存器的设置值
D_{21}：TQ0CCR2 寄存器的设置值
D_{31}, D_{32}, D_{33}：TQ0CCR3 寄存器的设置值

2. 上述流程图适用于 TMQ0 用作 PWM 输出模式时使用。

8.4.1 间隔定时器模式（TQ0MD2 至TQ0MD0 位 = 000）

在间隔定时器模式下，将 TQ0CTL0.TQ0CE 位置为 1，则会以指定时间间隔生成中断请求信号（INTTQ0CC0）。将 TQ0CE 位设置为 1，也可以启动定时器，然后从 TOQ00 引脚输出方波，指定的时间间隔等于方波的半周期。

通常在间隔定时器模式下不使用 TQ0CCR1 至 TQ0CCR3 寄存器。通过设置中断屏蔽标志（TQ0CCMK1 至 TQ0CCMK3）可以屏蔽来自该寄存器的中断。

备注 **1.** 关于如何设置 TOQ00 引脚，参见**表 8-2 TMQ0 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。
2. 关于如何使能 INTTQ0CC0 中断信号，参见**第 22 章 中断服务/异常处理。**

图 8-6. 间隔定时器的配置

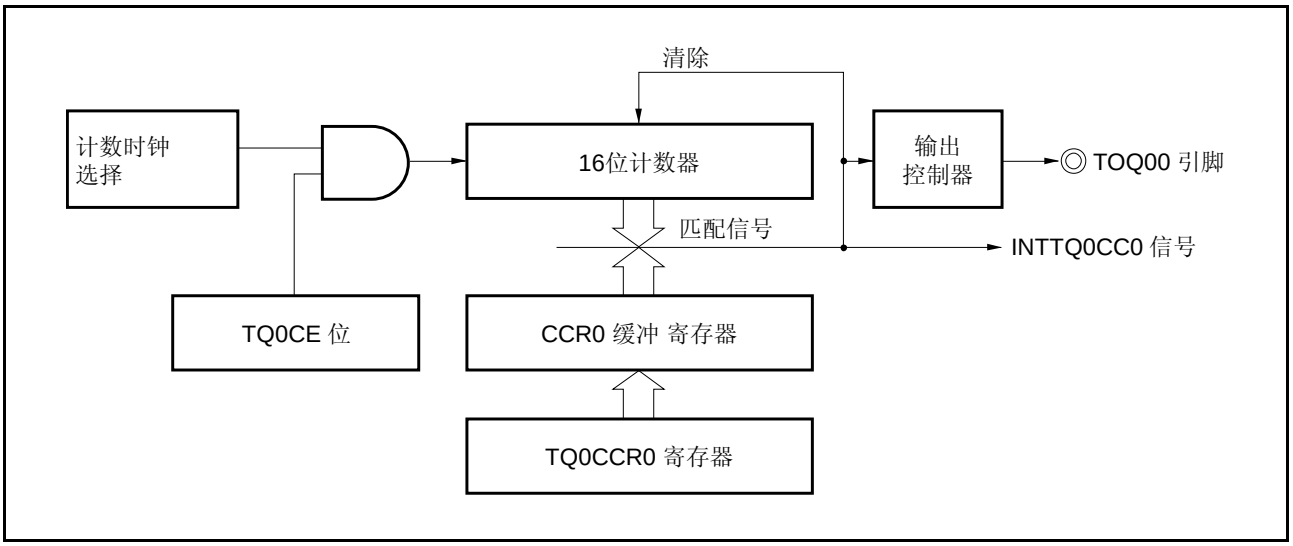
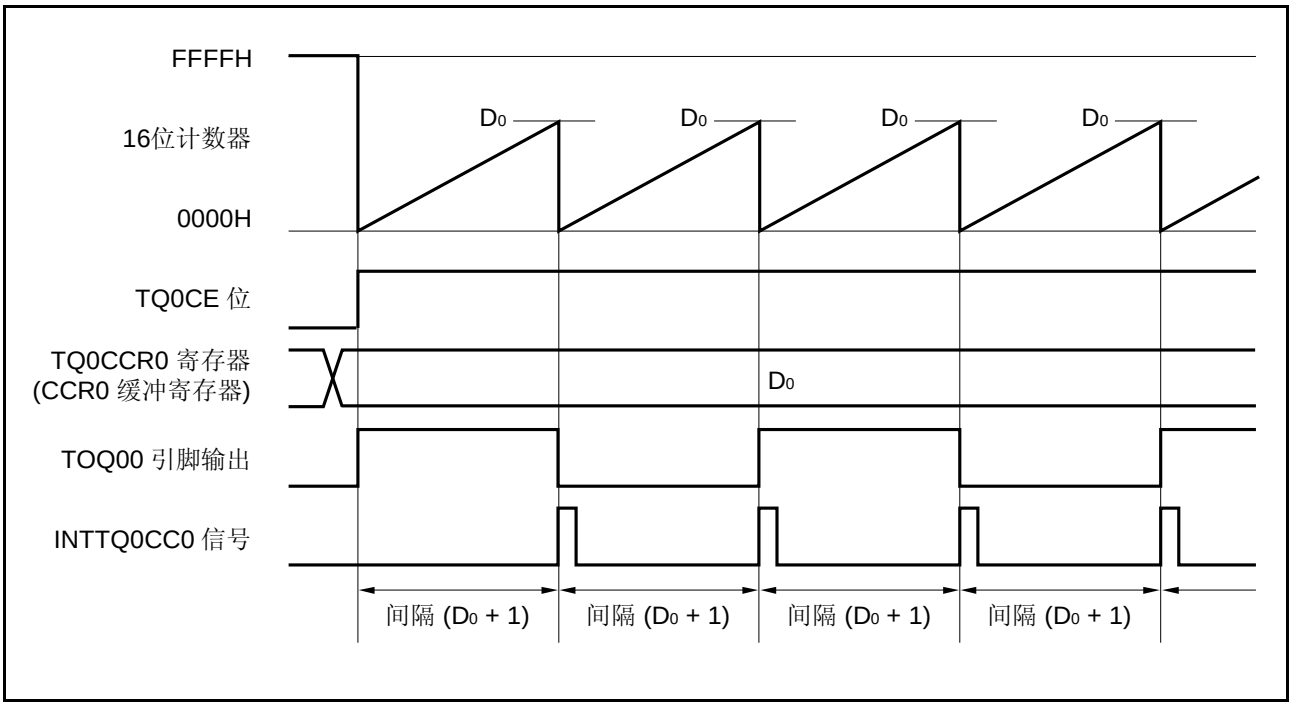


图 8-7. 间隔定时器模式的基本时序



当 TQ0CE 位被置为 1 时，16 位计数器的值由计数时钟同步从 FFFFH 清零为 0000H，计数器开始计数。此时，TOQ00 引脚的输出反相，且 TQ0CCR0 寄存器的设定值传送给 CCR0 缓冲寄存器。

当 16 位计数器的计数值和 CCR0 缓冲寄存器的值相匹配时，则 16 位计数器清除为 0000H，TOQ00 引脚输出反相，并生成一个比较匹配中断请求信号（INTTQ0CC0）。

时间间隔可用下述表达式计算。

时间间隔 =（TQ0CCR0 寄存器的设定值＋1）× 计数时钟周期

使用间隔定时器模式时设置寄存器的示例如下。

图 8-8. 间隔定时器模式操作时的寄存器设置（1/3）

（a）TMQ0 控制寄存器 0（TQ0CTL0）

	TQ0CE					TQ0CKS2	TQ0CKS1	TQ0CKS0
TQ0CTL0	0/1	0	0	0	0	0/1	0/1	0/1

选择计数时钟

0: 停止计数
1: 允许计数

（b）TMQ0 控制寄存器 1（TQ0CTL1）

		TQ0EST	TQ0EEE			TQ0MD2	TQ0MD1	TQ0MD0
TQ0CTL1	0	0	0/1注	0	0	0	0	0

0, 0, 0：
间隔定时器模式

0：计数时钟由TQ0CKS0
至TQ0CKS2位选择。
1：使用外部事件计数输入信号
计数

注 使用定时器输出（TOQ0k）时，TQ0EEE 位只能被设置为 1。请注意，当 TQ0EEE 位设置为 1 时，TQ0CCR0 和 TQ0CCRk 寄存器必须要设置为相同的值（也就是说，与已经为这些寄存器设置的值相同）（详情参见 **8.4.1（2）（d） TQ0CCR1 至 TQ0CCR3 寄存器的操作**）（k＝1 至 3）。

图 8-8. 间隔定时器模式操作时的寄存器设置（2/3）

（c）TMQ0 输入/输出控制寄存器 0（TQ0IOC0）

	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0IOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

- TQ0OE0: 0: 禁止TOQ00引脚输出 1: 允许 TOQ00引脚输出
- TQ0OL0: 禁止TOQ00引脚操作时输出电平设置 0: 低电平 1: 高电平
- TQ0OE1: 0: 禁止TOQ01引脚输出 1: 允许 TOQ01引脚输出
- TQ0OL1: 禁止 TOQ01引脚操作时输出电平设置 0: 低电平 1: 高电平
- TQ0OE2: 0: 禁止TOQ02引脚输出 1: 允许 TOQ02引脚输出
- TQ0OL2: 禁止 TOQ02引脚操作时输出电平设置 0: 低电平 1: 高电平
- TQ0OE3: 0: 禁止TOQ03引脚输出 1: 允许 TOQ03引脚输出
- TQ0OL3: 禁止 TOQ03引脚操作时输出电平设置 0: 低电平 1: 高电平

（d）TMQ0 输入/输出控制寄存器 2（TQ0IOC2）

					TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
TQ0IOC2	0	0	0	0	0/1注	0/1注	0	0

- TQ0EES1, TQ0EES0: 这些位选择外部事件计数输入(TIQ00 引脚)的有效沿

注 使用定时器输出（TOQ01 至 TOQ03）时，TQ0EES1 和 TQ0EES0 位只能被设置为 1。请注意，TQ0CCR0 至 TQ0CCR3 寄存器必须要设置为相同的值（也就是说，与已经为这些寄存器设置的值相同）。

（e）TMQ0 计数器读取缓冲寄存器（TQ0CNT）

通过读取该寄存器，可以读取 16 计数器的当前值。

（f）TMQ0 捕获/比较寄存器 0（TQ0CCR0）

如果 TQ0CCR0 寄存器的值被设置为 D_0，则时间间隔的计算如下：

时间间隔＝（D_0＋1）× 计数时钟周期

图 8-8. 间隔定时器模式操作时的寄存器设置（3/3）

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

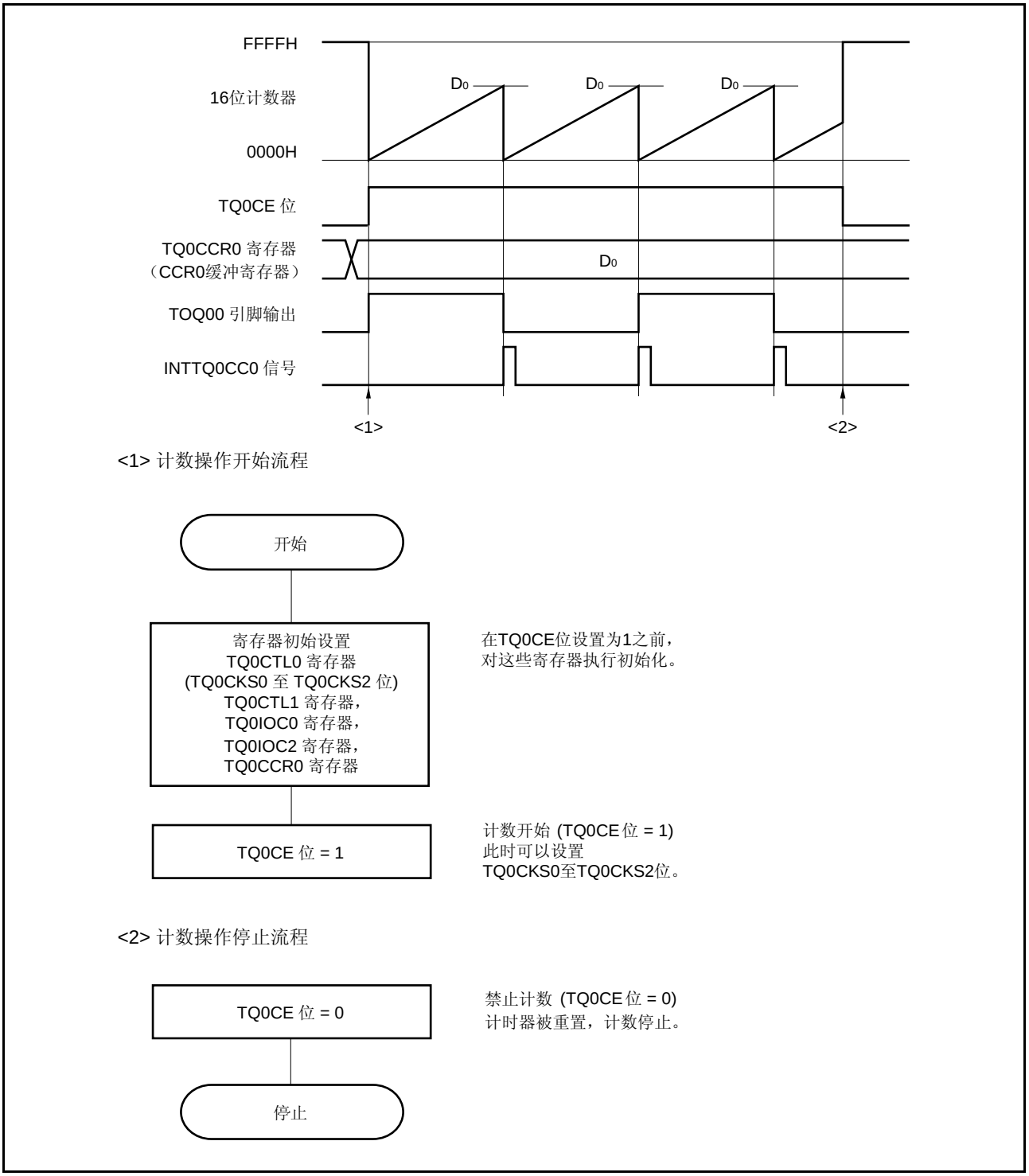
通常在间隔定时器模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。但是，TQ0CCR1 至 TQ0CCR3 寄存器的设定值仍然会被传送给 CCR1 至 CCR3 缓冲寄存器。当 16 位计数器的计数值与 CCR1 至 CCR3 缓冲寄存器的值匹配时，会产生比较匹配中断请求信号（INTTQ0CC1 至 INTTQ0CCR3）。

必须通过设置中断屏蔽标志（TQ0CCMK1 至 TQ0CCMK3）屏蔽来自这些寄存器的中断请求。

备注 在间隔定时器模式下，不使用 TMQ0 输入/输出控制寄存器 1（TQ0IOC1）和 TMQ0 选项寄存器 0（TQ0OPT0）。

（1）间隔定时器模式的操作

图 8-9. 间隔定时器模式下的时序和操作处理



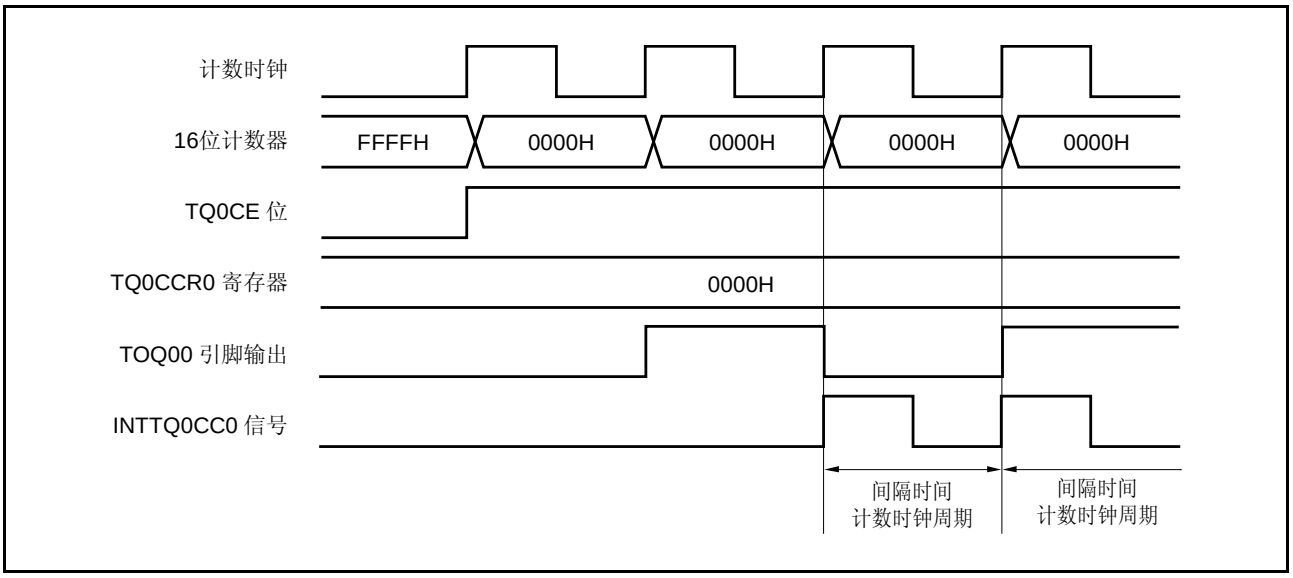
（2）使用间隔定时器模式

（a）TQ0CCR0 寄存器清除为 0000H 时的操作

如果 TQ0CCR0 寄存器被清除为 0000H，第二个时钟及其后的每个时钟计数时都会产生 INTTQ0CC0 信号，而且，TOQ00 引脚的输出反相。

16 位计数器的值总是 0000H。

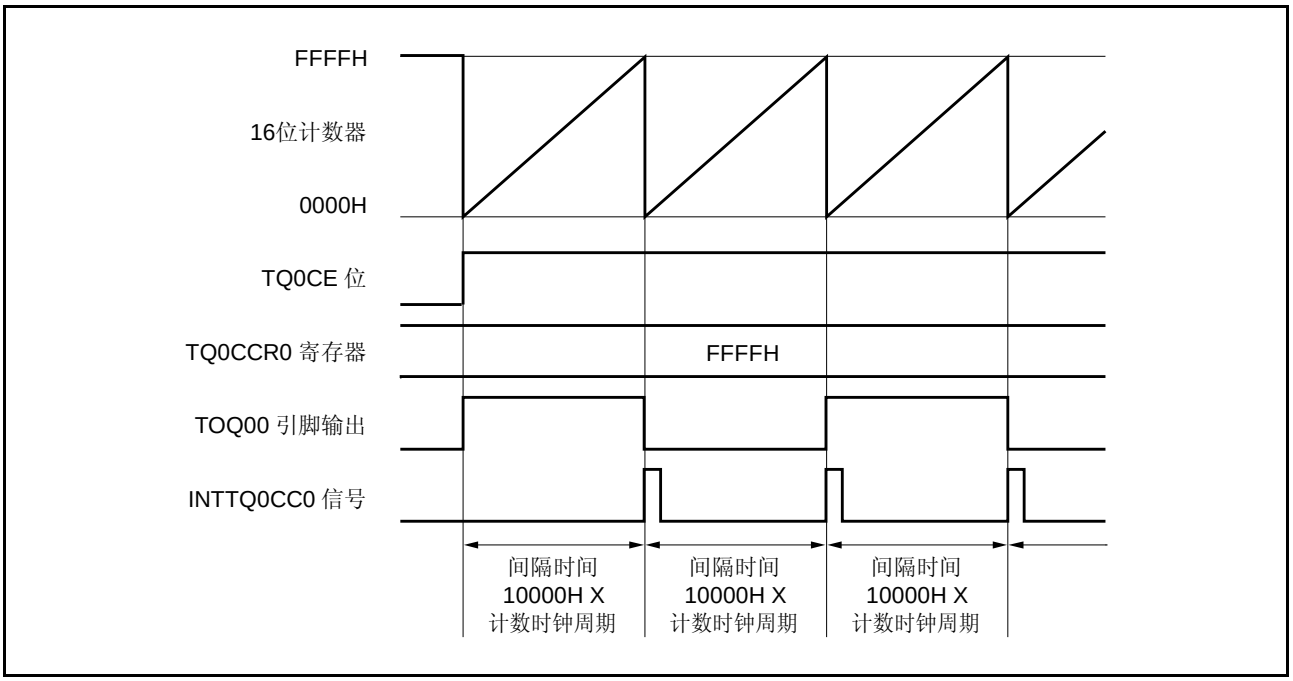
图 8-10. 当 TQ0CCR0 寄存器清为 0000H 时间隔定时器的操作



（b）TQ0CCR0 寄存器被设置为 FFFFH 时的操作

如果 TQ0CCR0 寄存器被置为 FFFFH，那么 16 位计数器可以计数到 FFFFH。该计数器在下一个计数时刻被同步清零。产生 INTTQ0CC0 信号，并且 TOQ00 引脚输出反相。此时，不产生溢出中断请求信号（INTTQ0OV），而且，溢出标志（TQ0OPT0.TQ0OVF 位）也不会置 1。

图 8-11. 当 TQ0CCR0 寄存器置为 FFFFH 时间隔定时器的操作

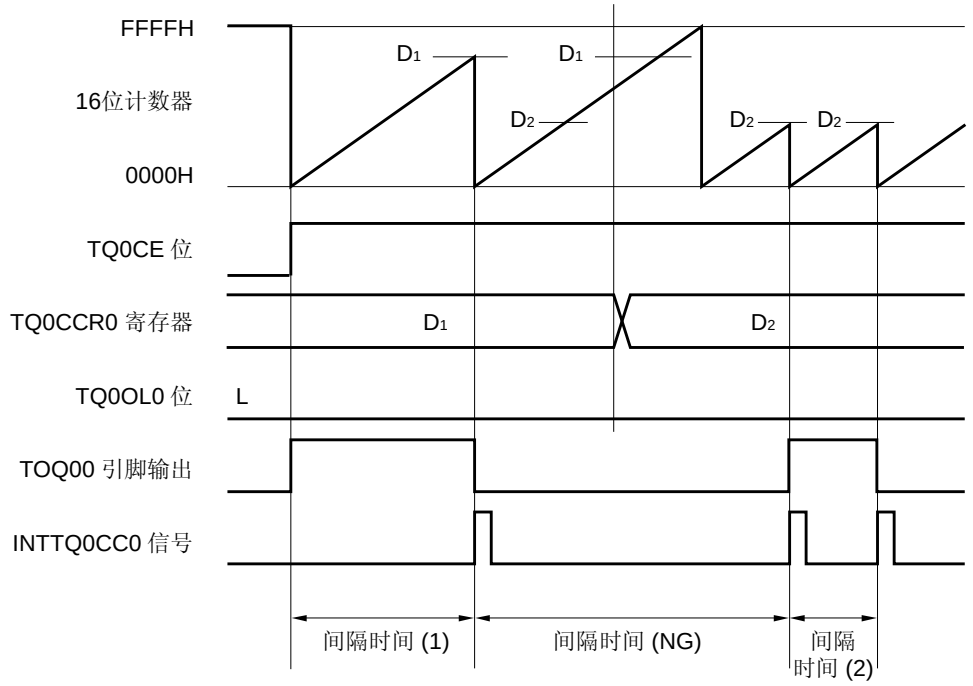


（c）重新写入 TQ0CCR0 寄存器的注意事项

要将 TQ0CCR0 寄存器的值重写为比当前值小的值，应首先停止计数，然后改变其设定值。

如果在计数器工作期间，将寄存器 TQ0CCR0 的值重写为比当前值小的值，16 位计数器可能会溢出。

图 8-12. 重写 TQ0CCR0 寄存器



备注 间隔时间（1）：（D_1＋1）× 计数时钟周期

间隔时间（NG）：（10000H＋D_2＋1）× 计数时钟周期

间隔时间（2）：（D_2＋1）× 计数时钟周期

当 16 位计数器计数值大于 D_2 而小于 D_1 时，如果将 TQ0CCR0 寄存器的值从 D_1 变为 D_2，在重写 TQ0CCR0 寄存器后，TQ0CCR0 寄存器的值会立即传送给 CCR0 缓冲寄存器。结果，用来与 16 位计数器比较的值是 D_2。

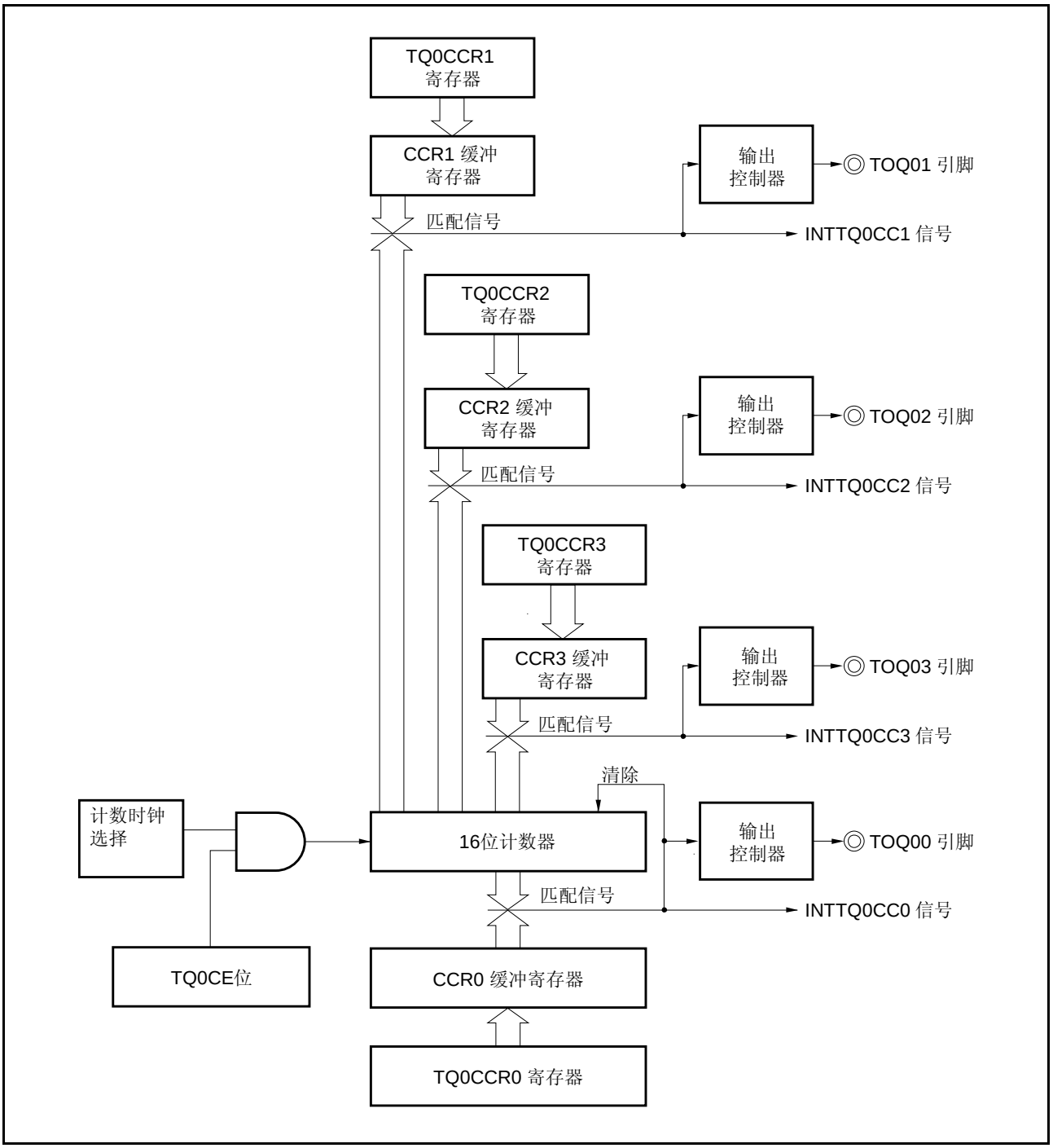
但是，由于计数值已经超过了 D_2，所以 16 位计数器就会向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值和 D_2 相符时才产生 INTTQ0CC0 信号，且 TOQ00 引脚的输出反相。

因此，可能不会在原先预计的间隔时间“（D_1＋1）× 计数时钟周期” 或 “（D_2＋1）× 计数时钟周期”时产生 INTTQ0CC0 信号，而会以“（10000H＋D_2＋1）× 计数时钟周期” 的间隔产生中断请求信号。

（d） TQ0CCR1 至 TQ0CCR3 寄存器的操作

间隔定时器模式下对 TQ0CCR1 至 TQ0CCR3 寄存器的配置如下所示。

图 8-13. TQ0CCR1 至 TQ0CCR3 寄存器的配置



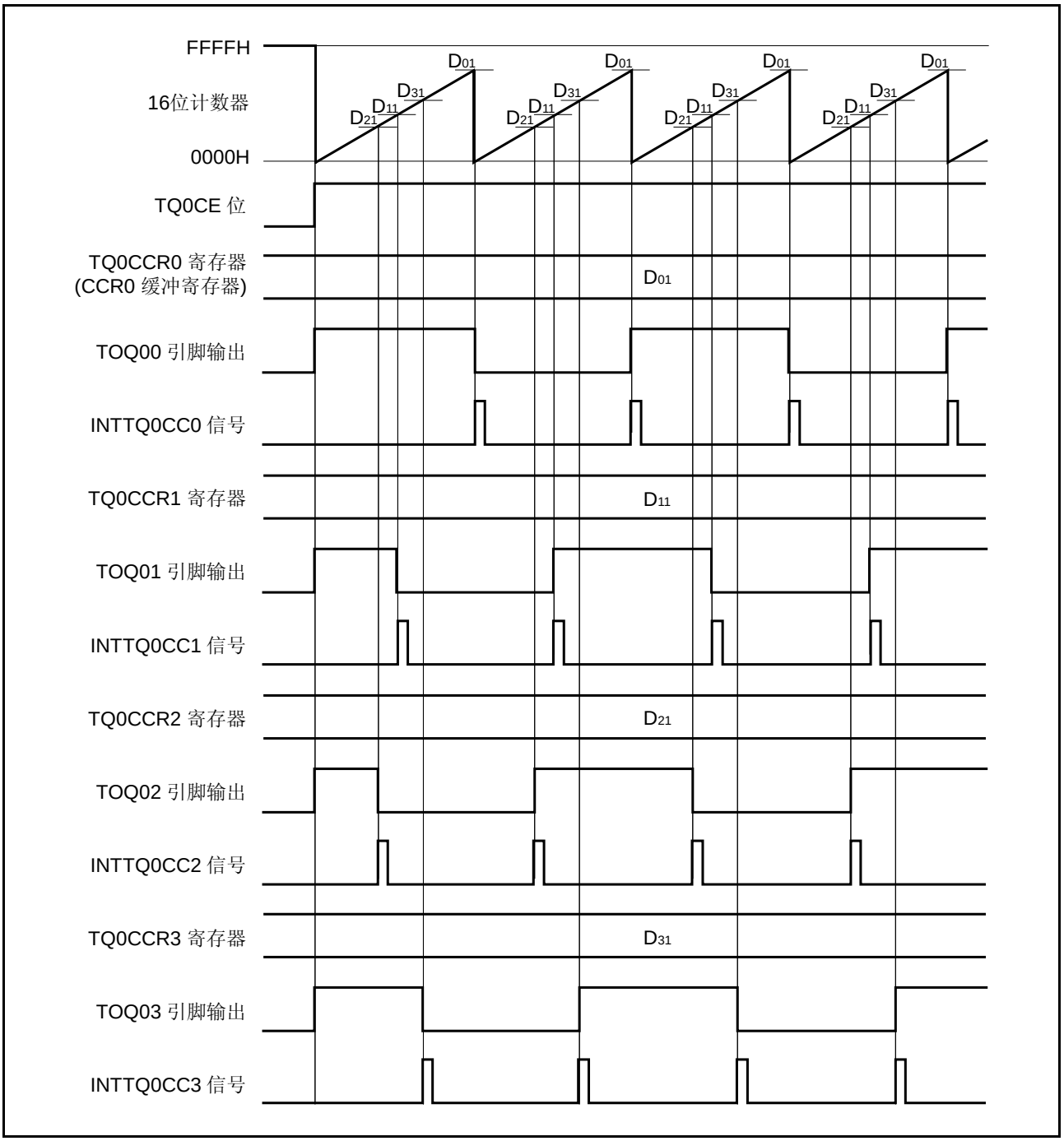
如果 TQ0CCRk 寄存器的值小于或等于 TQ0CCR0 寄存器的值，则每个计数周期产生一次 INTTQ0CCk 中断请求信号。同时，TOQ0k 引脚的输出被反转。

TOQ0k 引脚输出一个方波，其周期与 TOQ00 引脚的输出周期相同，但是相位不同。

当 TQ0CCRk 寄存器的值(D_{k1})小于或等于 TQ0CCR0 寄存器的值(D_{01})时，操作时序图显示如下。

备注 k＝1 至 3

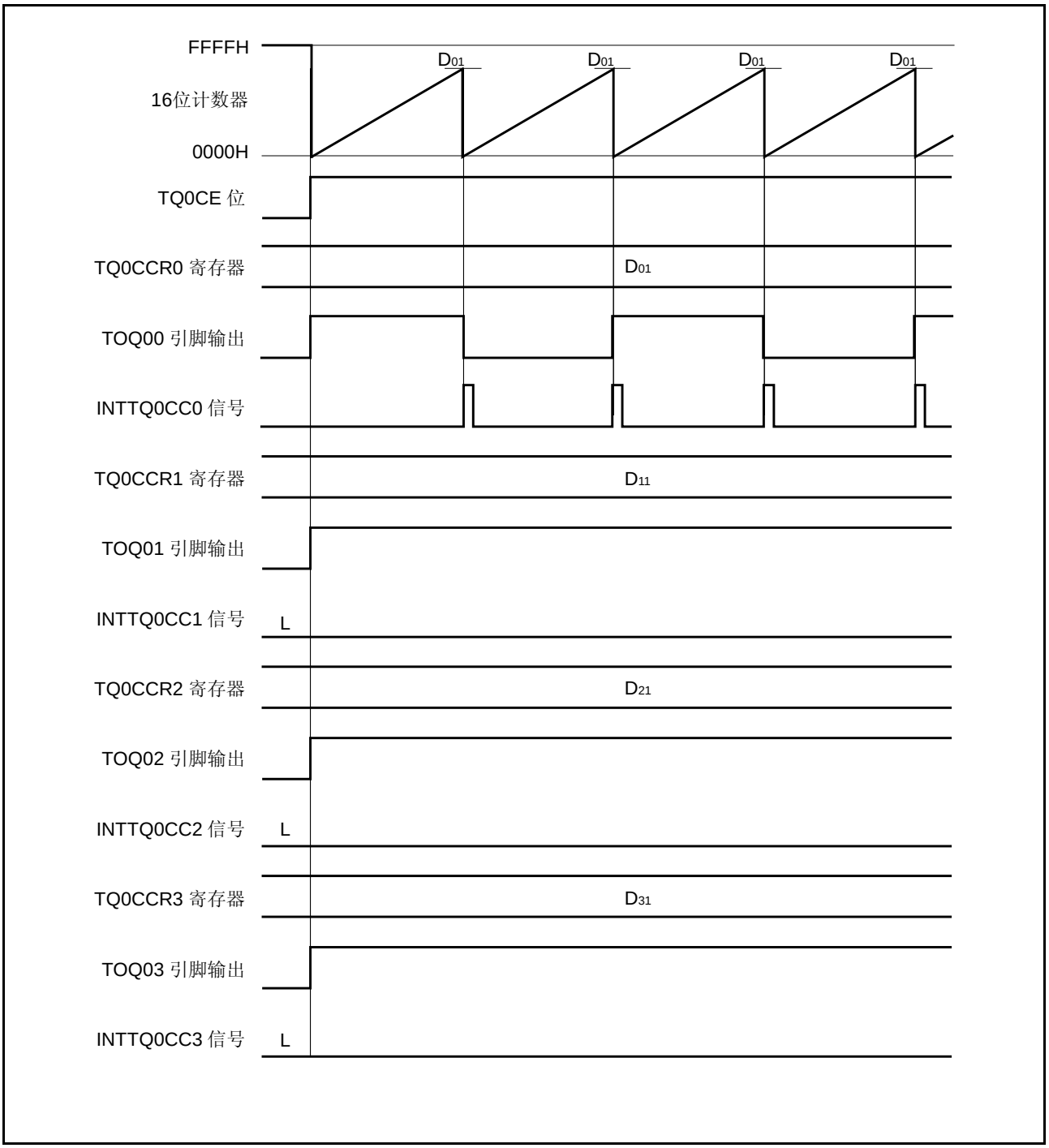
图 8-14. 当 $D_{01} \geq D_{k1}$ 时的时序图



如果 TQ0CCRk 寄存器的设定值大于 TQ0CCR0 寄存器的设定值，则 16 位计数器的计数值不会和 TQ0CCRk 寄存器的值发生匹配。因此也就不会产生 INTTQ0CCk 信号，TOQ0k 引脚的输出也不会改变。
当 TQ0CCRk 寄存器的值(D_{K1})大于 TQ0CCR0 寄存器的值(D_{01})时，操作时序图显示如下。

备注 k＝1 至 3

图 8-15. 当 D_{01} < D_{k1} 时的时序图



（3）基于外部事件计数的间隔定时器操作

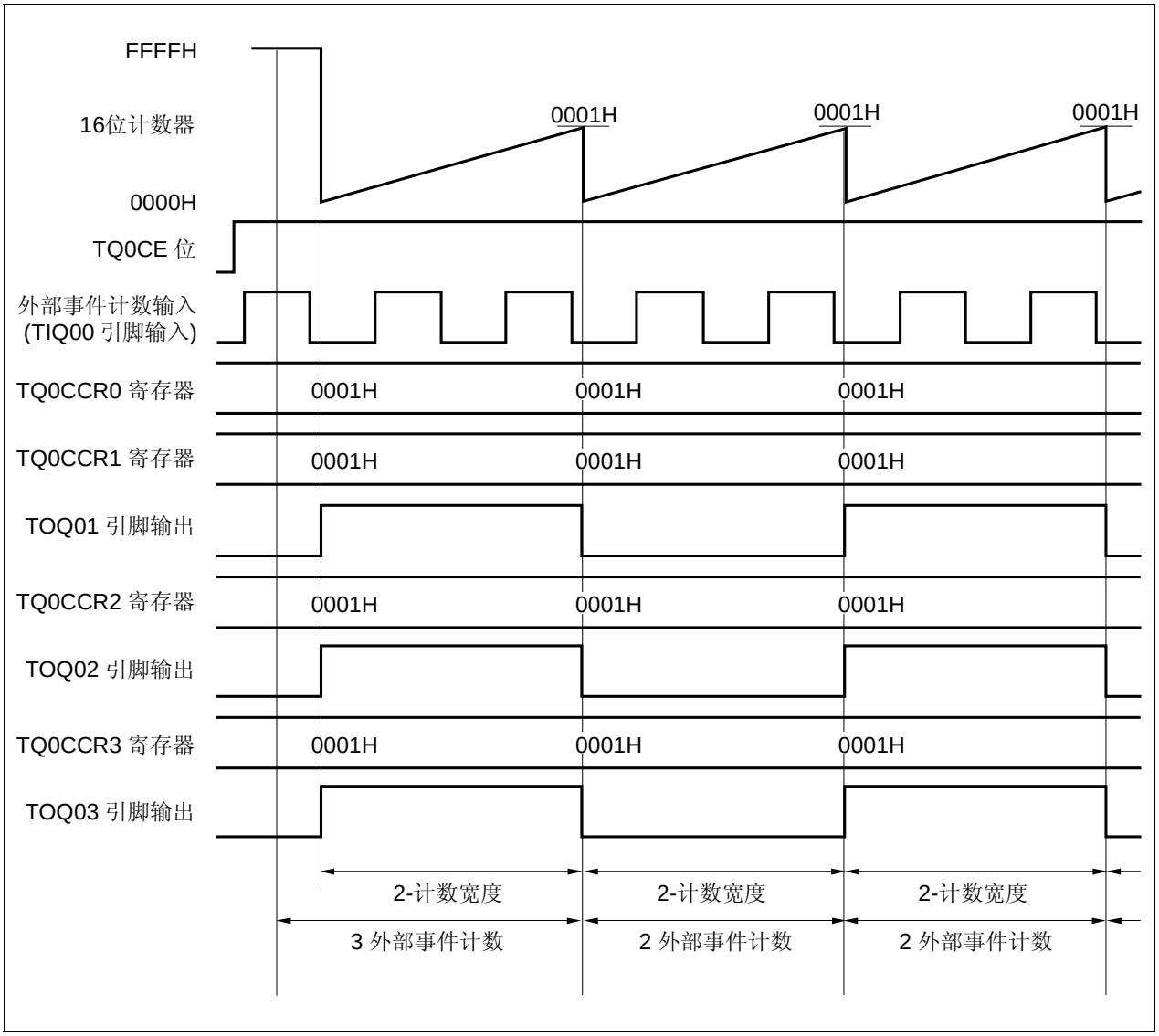
(a) 操作

在间隔定时器模式下，当 16 位计数器基于外部事件计数输入(TIQ00 引脚)的有效沿递增时，16 位计数器从 FFFFH 被清除为 0000H 之后，紧随 TQ0CE 位从 0 改变为 1 启动计数器递增之后，必须有外部事件计数有效边沿立即输入。

一旦 TQ0CCR0 和 TQ0CCRk 寄存器被设置为 0001H (也就是说，与之前设置的值相同)，16 位计数器的每计数两次，TOQ0k 引脚输出都会反转。

请注意，当基于外部事件计数使用定时器输出(TOQ0k)时，TQ0CTL1.TQ0EEE 位只能被设置为 1。

图 8-16. 基于外部事件计数输入的间隔定时器操作



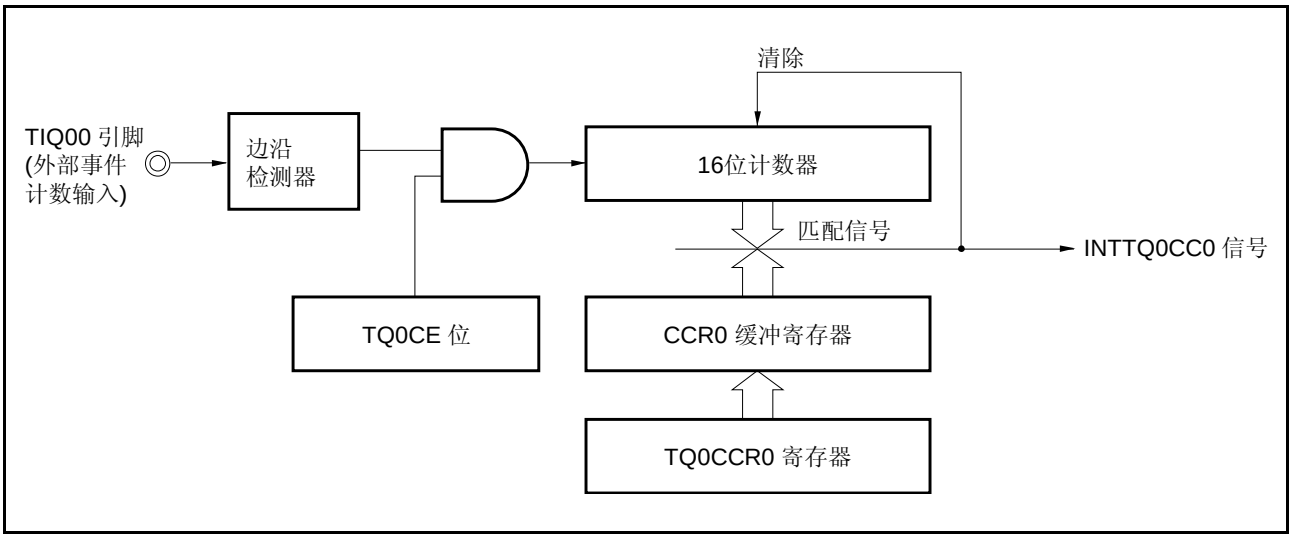
8.4.2 外部事件计数模式（TQ0MD2 至TQ0MD0 位 = 001）

在外部事件计数器模式下，当 TQ0CTL0.TQ0CE 位被置为 1 时，对外部事件计数输入信号的有效边沿进行计数，每次有效边沿的计数数量达到指定数目时，产生一个中断请求信号（INTTQ0CC0）。不能使用定时器输出引脚（TOQ00 至 TOQ03）。要在外部事件计数模式下使用 TOQ01 至 TOQ03 引脚，请确保先在间隔定时器模式下将 TQ0CTL1.TQ0EEE 位设置为 1（详情参见 **8.4.1 (3) 基于外部事件计数输入的间隔定时器操作**）。

通常在外部事件计数器模式下，不使用 TQ0CCR1 至 TQ0CCR3 寄存器。

备注 **1.** 关于如何设置 TOQ00 引脚，参见**表 8-2 TMQ0 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。
2. 关于如何使能 INTTQ0CC0 中断信号，参见**第 22 章 中断服务/异常处理。**

图 8-17. 外部事件计数模式下的间隔定时器配置

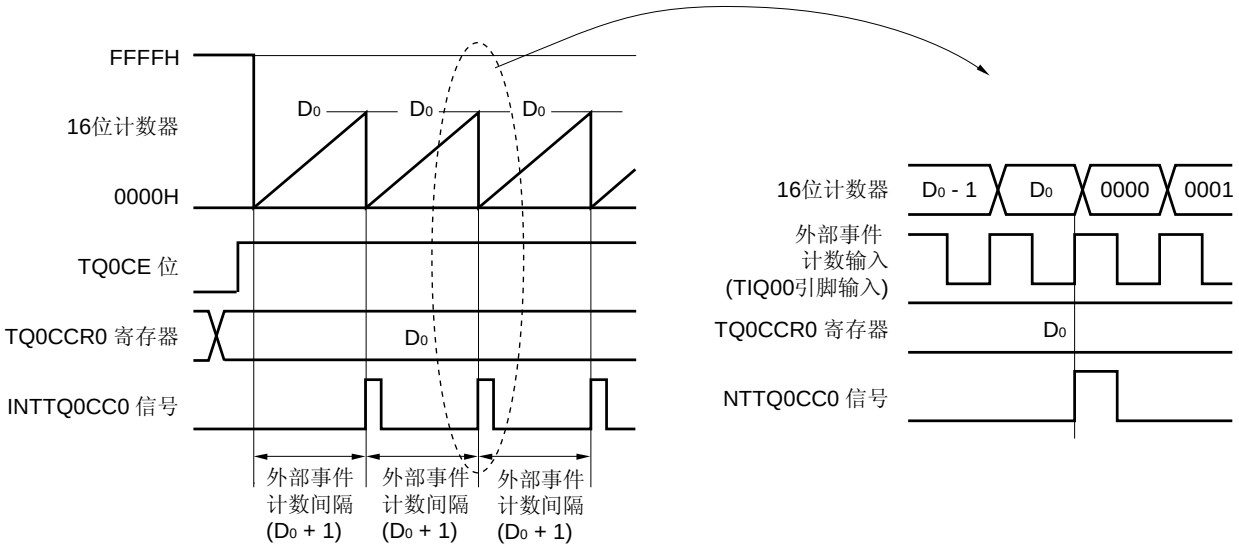


当 TQ0CE 位被置为 1 时，16 位计数器的值从 FFFFH 被清为 0000H。计数器对检测到的每一次外部事件计数输入的有效边沿进行计数。此外，TQ0CCR0 寄存器的设定值被传送给 CCR0 缓冲寄存器。

当 16 位计数器的计数值和 CCR0 缓冲寄存器的数值相匹配时，16 位计数器清零为 0000H，同时生成一个比较匹配中断请求信号（INTTQ0CC0）。

每当外部事件计数输入的有效边沿数量达到（TQ0CCR0 寄存器值＋1）时，就会生成 INTTQ0CC0（比较匹配中断请求）信号。

图 8-18. 外部事件计数模式下的基本操作时序



备注 本图显示了当上升沿被指定作为外部事件计数输入有效边沿时的基本时序。

外部事件计数模式下的寄存器操作设置示例如下图所示。

图 8-19. 外部事件计数模式下的寄存器设置（1/2）

（a）TMQ0 控制寄存器 0（TQ0CTL0）

	TQ0CE					TQ0CKS2	TQ0CKS1	TQ0CKS0
TQ0CTL0	0/1	0	0	0	0	0	0	0

0: 停止计数
1: 允许计数

（b）TMQ0 控制寄存器 1（TQ0CTL1）

		TQ0EST	TQ0EEE			TQ0MD2	TQ0MD1	TQ0MD0
TQ0CTL1	0	0	0	0	0	0	0	1

0, 0, 1:
外部事件计数模式

（c）TMQ0 输入/输出控制寄存器 0（TQ0IOC0）

	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0IOC0	0	0	0	0	0	0	0	0

0: 禁止TOQ00引脚输出
0: 禁止TOQ01引脚输出
0: 禁止TOQ02引脚输出
0: 禁止TOQ03引脚输出

（d）TMQ0 输入/输出控制寄存器 2（TQ0IOC2）

					TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
TQ0IOC2	0	0	0	0	0/1	0/1	0	0

选择外部事件计数
输入的有效沿

（e）TMQ0 计数器读取缓冲寄存器（TQ0CNT）

可以通过读取该寄存器，读取 16 位计数器的计数值。

图 8-19. 外部事件计数模式下的寄存器设置（2/2）

（f）TMQ0 捕获/比较寄存器 0（TQ0CCR0）

如果 TQ0CCR0 寄存器的值被设定为 D_0，当外部事件计数值达到（D_0＋1）时，计数器被清 0，同时产生一个比较匹配中断请求信号（INTTQ0CC0）。

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

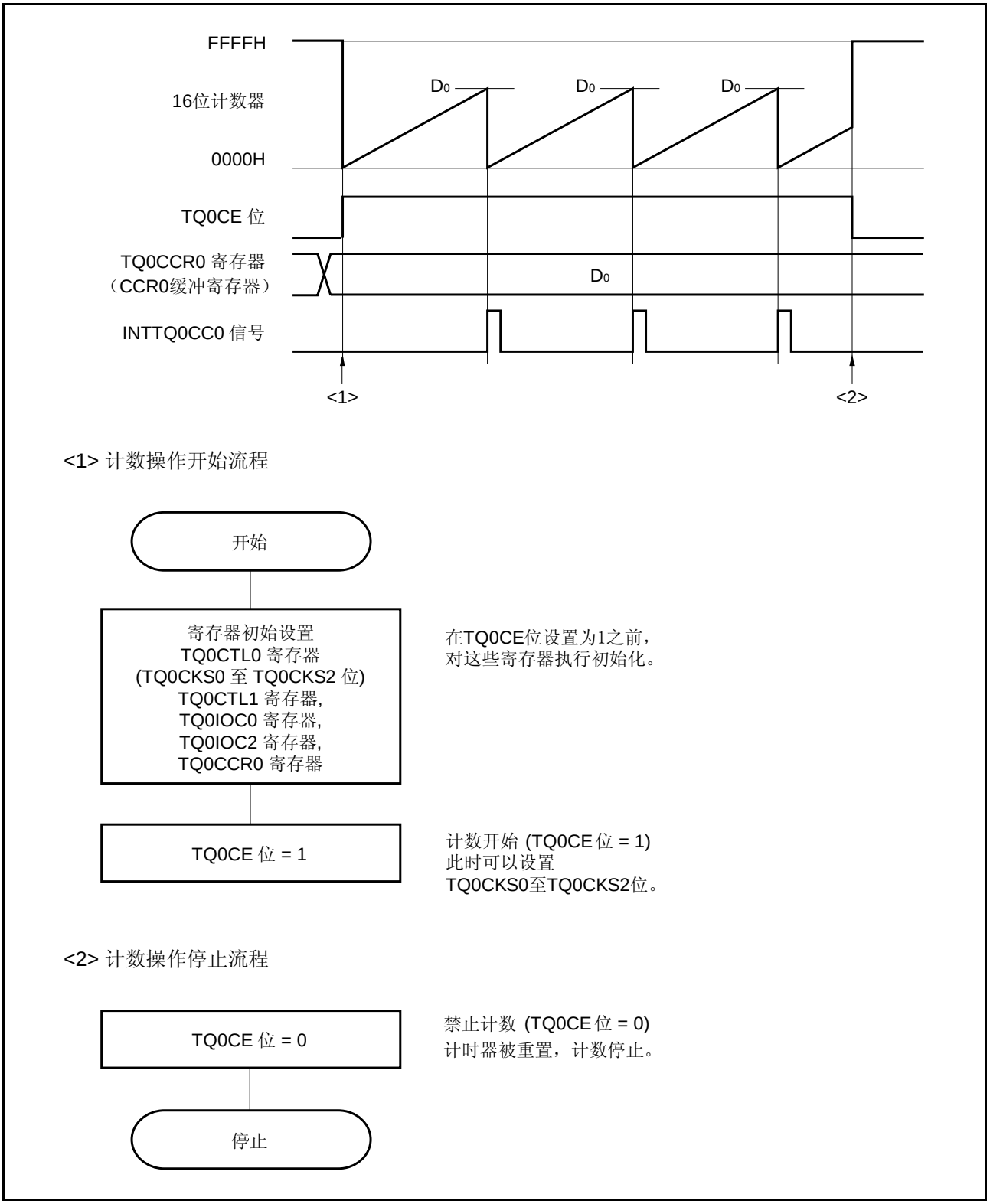
通常在外部事件计数模式中不使用 TQ0CCR1 至 TQ0CCR3 寄存器。但是，当 16 位计数器的计数值和 CCR1 至 CCR3 缓冲寄存器的值匹配时，TQ0CCR1 至 TQ0CCR3 寄存器的设定值仍然被传送给 CCR1 至 CCR3 缓冲寄存器，并产生一个比较匹配中断请求信号（INTTQ0CC1 至 INTTQ0CC3）。通过设置中断屏蔽标志（TQ0CCMK1 至 TQ0CCMK3）屏蔽来自这些寄存器的中断信号。

注意事项
1. **外部事件计数模式下，不要将 TQ0CCR0 寄存器设置为 0000H。**
2. **外部事件计数模式下，不能使用定时器输出。如果基于外部事件计数输入来进行定时器输出，首先应设置为间隔定时器模式，然后指定外部事件计数输入“操作使能”（TQ0CTL1.TQ0MD2 至 TQ0MD0 位为 0, 0, 0，并将 TQ0CTL1.TQ0EEE 位设置为 1）。详情参见 8.4.1 (3) 基于外部事件计数输入的间隔定时器操作。**
3. **当外部时钟用作计数时钟时，外部时钟只能从 TIQ00 引脚输入。此时，将 TQ0IOC1.TQ0IS1 和 TQ0IOC1.TQ0IS0 位清除为 0, 0（捕获触发输入（TIQ00 引脚）：无边沿检测）。**

备注 在外部事件计数模式下，不使用 TMQ0 输入/输出控制寄存器 1（TQ0IOC1）和 TMQ0 选项寄存器 0（TQ0OPT0）。

（1）外部事件计数模式的操作

图 8-20. 外部事件计数模式下的时序和操作处理

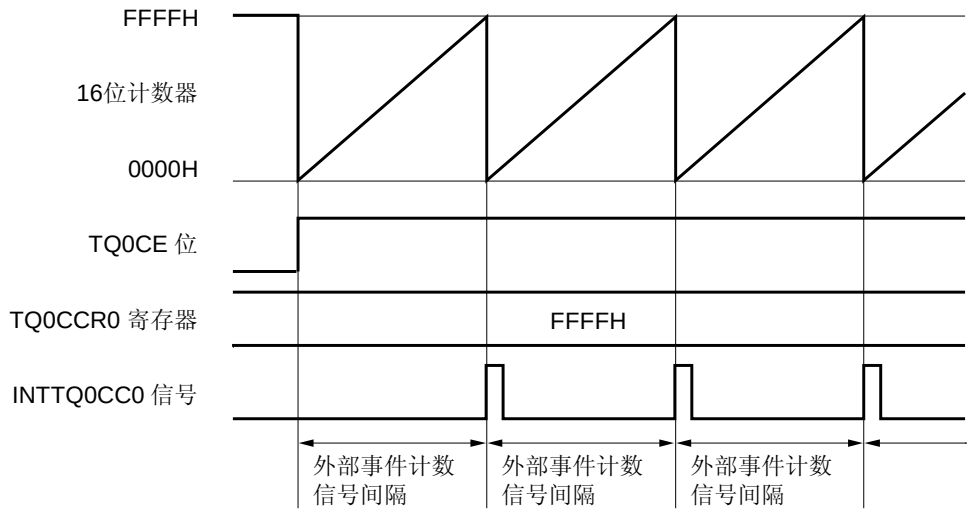


（2）使用外部事件计数模式

（a）当 TQ0CCR0 寄存器被设置为 FFFFH 时的操作

如果 TQ0CCR0 寄存器被设置为 FFFFH，16 位计数器每次检测到的外部事件输入信号有效边沿时，递增计数到 FFFFH。并在下一个递增计数时被同步清为 0000H，并且产生 INTTQ0CC0 信号。此时，TQ0OPT0.TQ0OVF 位不被设置为 1。

图 8-21. 当 TQ0CCR0 寄存器被设置为 FFFFH 时的操作

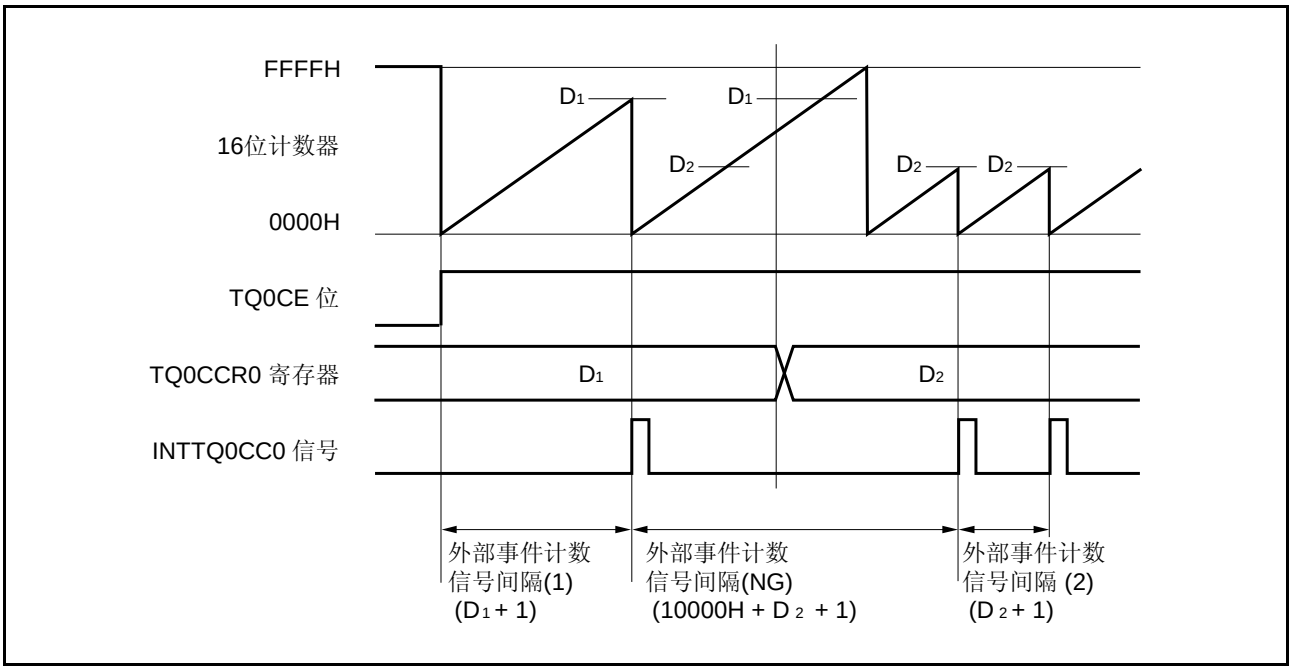


（b）重写 TQ0CCR0 寄存器的注意事项

要将 TQ0CCR0 寄存器的值改为比当前值较小的值，应首先停止计数，然后改变其设定值。

如果在计数器进行计数期间，对寄存器 TQ0CCR0 的值进行重写，则 16 位计数器可能会溢出。

图 8-22. 重写 TQ0CCR0 寄存器



当 16 位计数器计数值大于 D_2 而小于 D_1 时，如果将 TQ0CCR0 寄存器的值从 D_1 变为 D_2，则在重写 TQ0CCR0 寄存器后，计数值会立即传送给 CCR0 缓冲寄存器。因而，与 16 位计数器相比较的值会是 D_2。

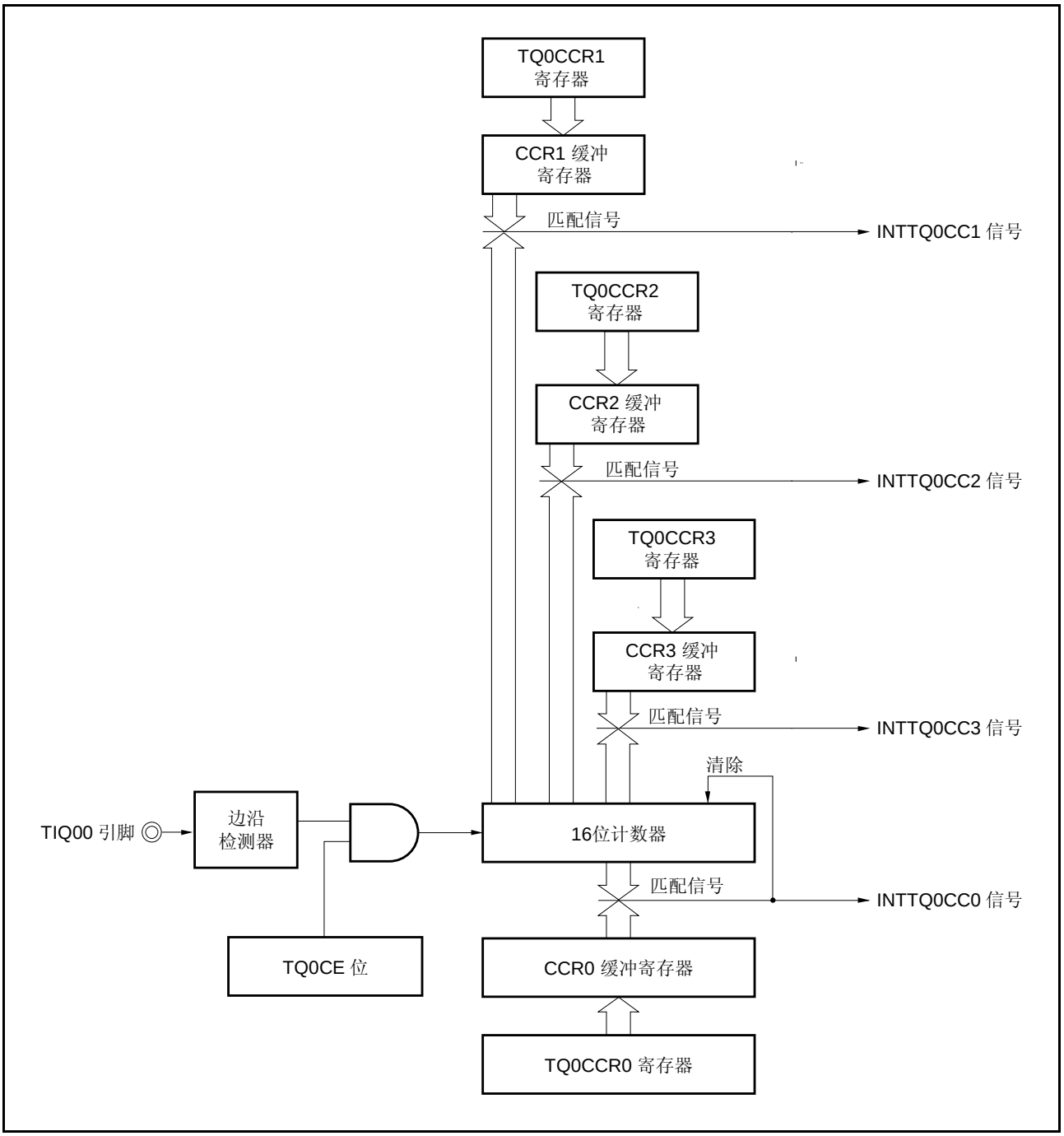
但是，由于计数值已经超过了 D_2，所以 16 位计数器就会向上累积计数到 FFFFH，溢出，然后从 0000H 开始重新计数。当计数值与 D_2 匹配时，会生成 INTTQ0CC0 信号。

因此，可能不会在原先预计的“（D_1＋1）次”或“（D_2＋1）次”有效边沿生成 INTTQ0CC0 中断请求信号，而可能在外部事件计数信号的“（10000H＋D_2＋1）次”有效边沿处生成中断请求信号。

（c）TQ0CCR1 至 TQ0CCR3 寄存器的操作

外部事件计数模式下的 TQ0CCR1 至 TQ0CCR3 寄存器配置如下图所示。

图 8-23. TQ0CCR1 至 TQ0CCR3 寄存器的配置

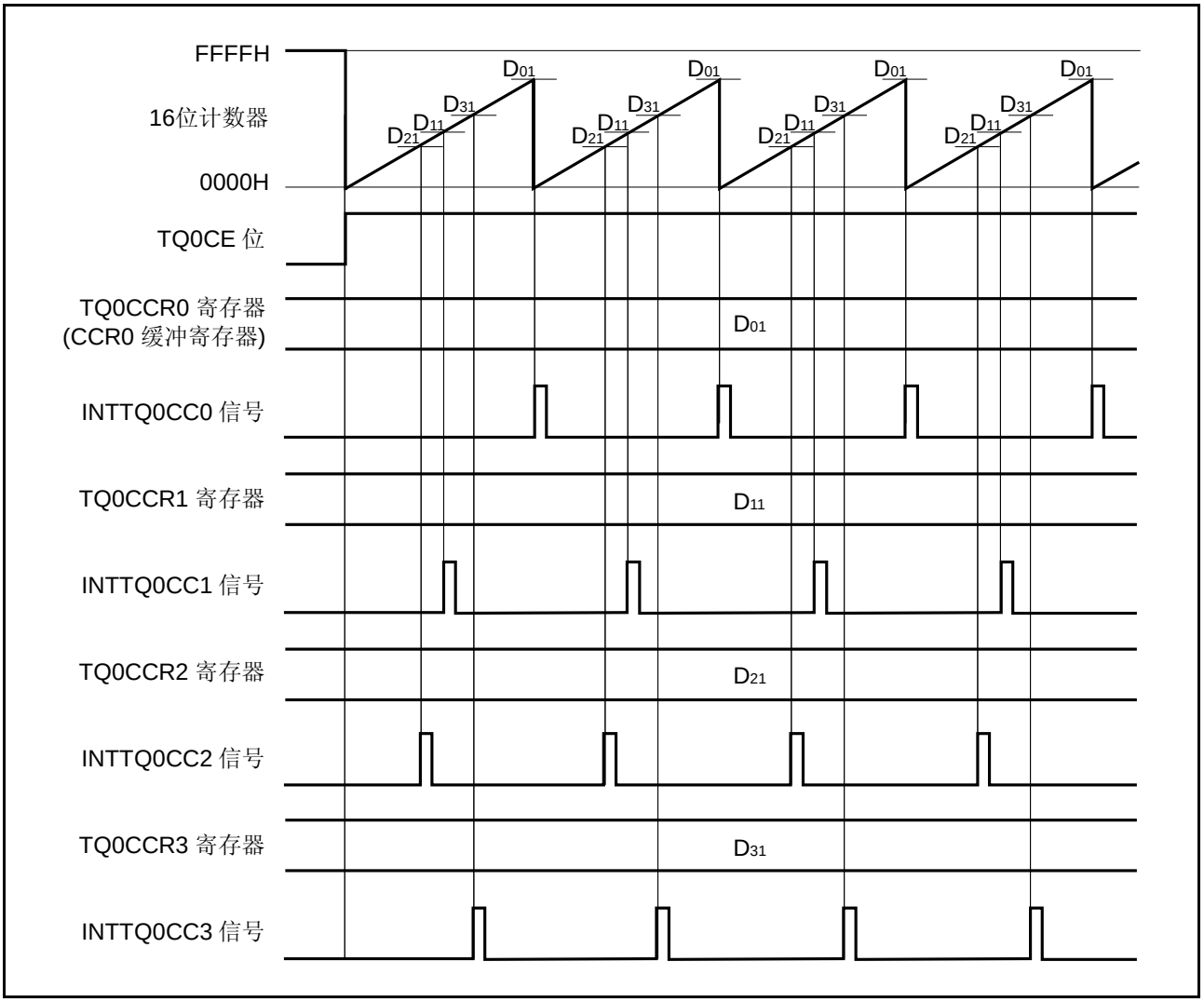


如果 TQ0CCRk 寄存器的设定值小于或等于 TQ0CCR0 寄存器的设定值，则每个计数周期都产生一次 INTTQ0CCk 中断请求信号。

当 TQ0CCRk 寄存器的值(D_{K1})小于或等于 TQ0CCR0 寄存器的值(D_{01})时，操作时序图显示如下。

备注 k＝1 至 3

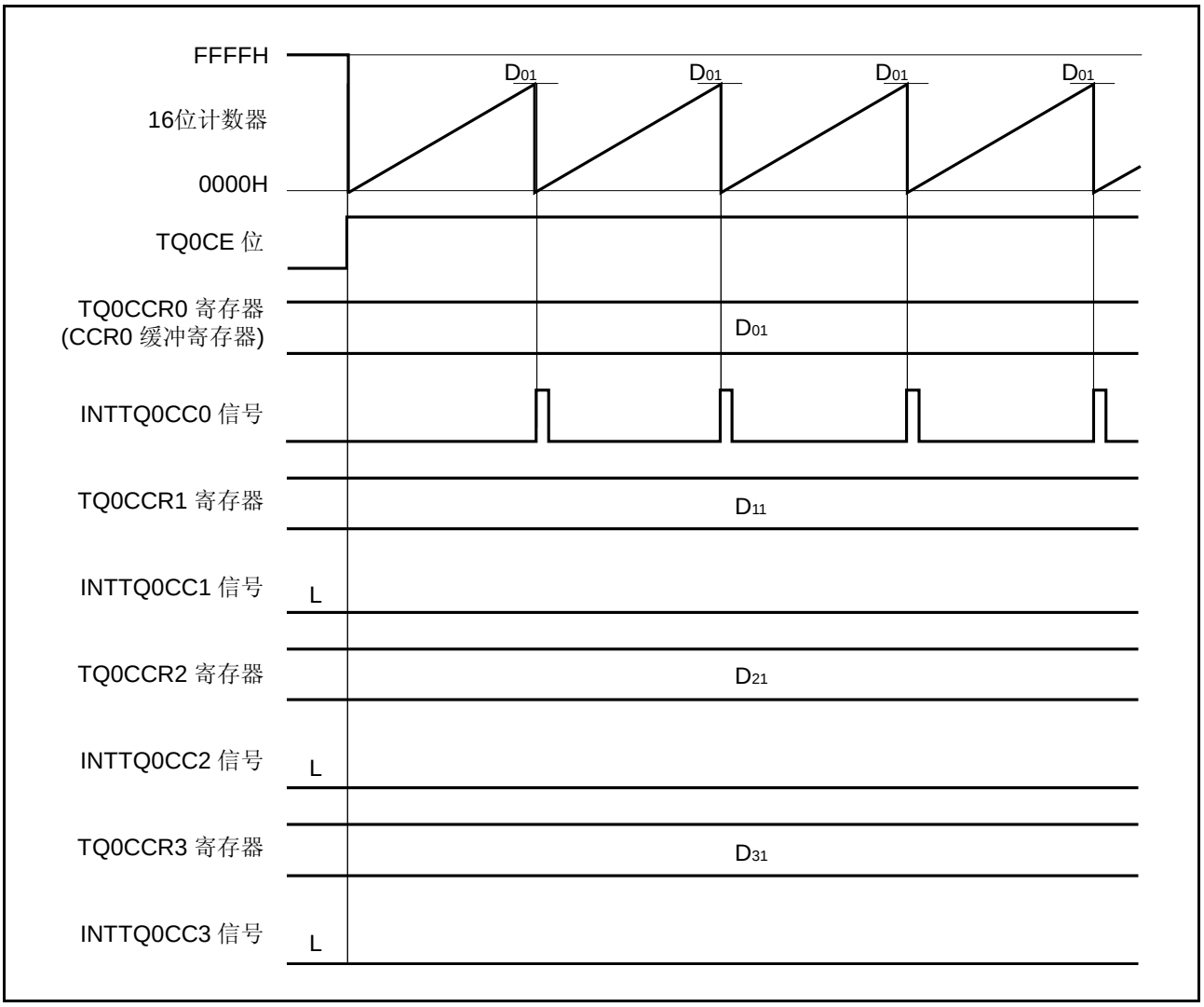
图 8-24. 当 $D_{01} \geq D_{k1}$ 时的时序图



如果 TQ0CCRk 寄存器的值大于 TQ0CCR0 寄存器的值，16 位计数器的计数值不会和 TQ0CCRk 寄存器的值发生匹配，也就不会产生 INTTQ0CCk 中断请求信号。
当 TQ0CCRk 寄存器的值(D_{k1})大于 TQ0CCR0 寄存器的值(D_{01})时，操作时序图显示如下。

备注 k＝1 至 3

图 8-25. 当 D_{01} < D_{k1} 时的时序图



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

在外部触发脉冲输出模式下，当 TQ0CTL0.TQ0CE 位被置为 1 时，TMQ0 等待触发信号，当检测到外部触发输入信号的有效边沿后，TMQ0 开始计数，并且从 TOQ01 至 TOQ03 引脚输出一个 PWM 波形。

不使用外部触发，而是使用软件触发，同样可以得到输出脉冲。当使用软件触发时，TOQ00 引脚输出一个方波，其半周期等于输出 PWM 的一个周期。

备注 **1.** 关于如何设置 TIQ00 和 TOQ00 至 TOQ03 引脚，参见**表 8-2 TMQ0 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。

2. 关于如何使能 INTTQ0CC0 和 INTTQ0CC0 中断信号，参见**第 22 章 中断服务/异常处理。**

图 8-26. 外部触发脉冲输出模式下的 TMQ0 配置

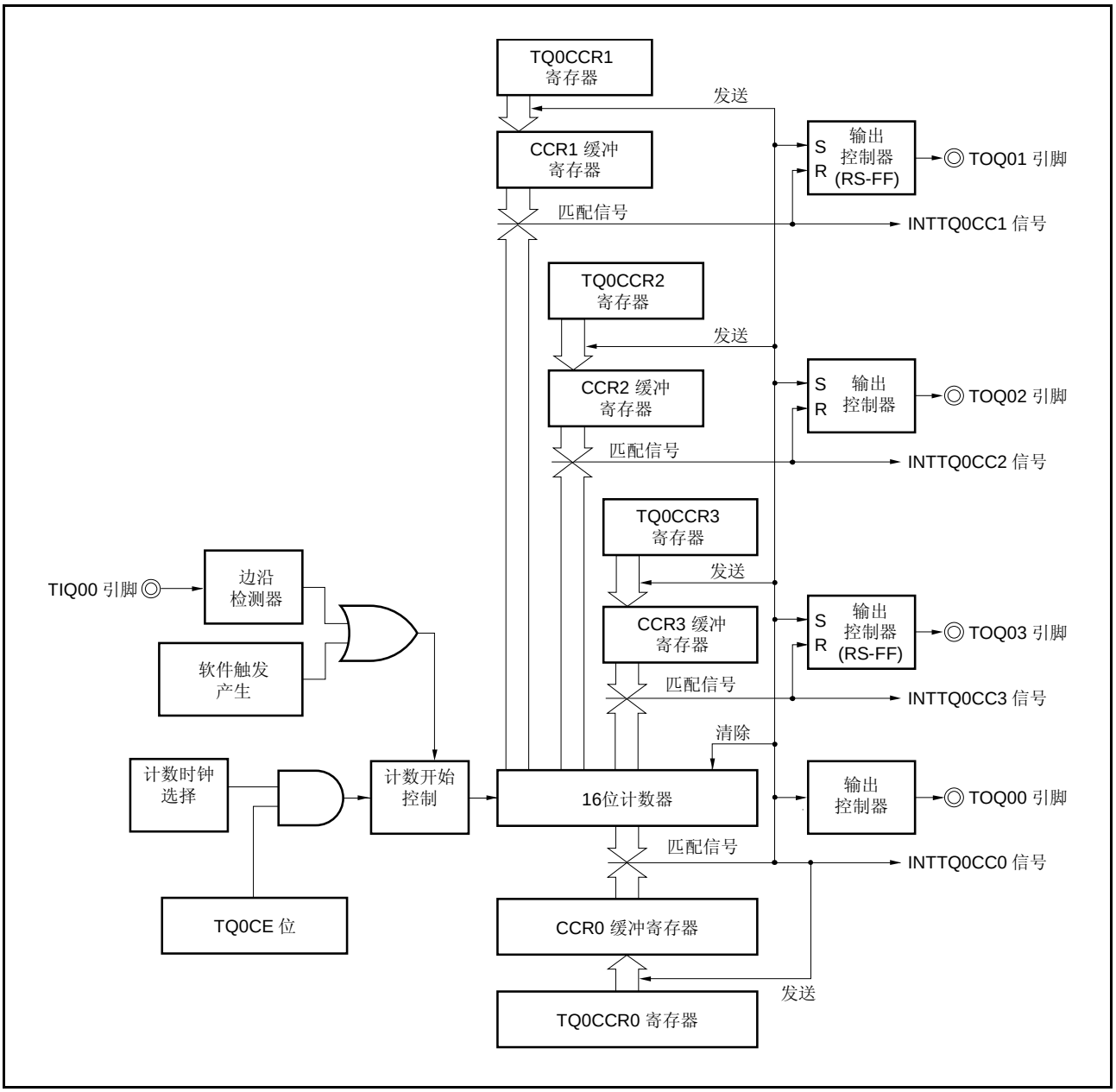
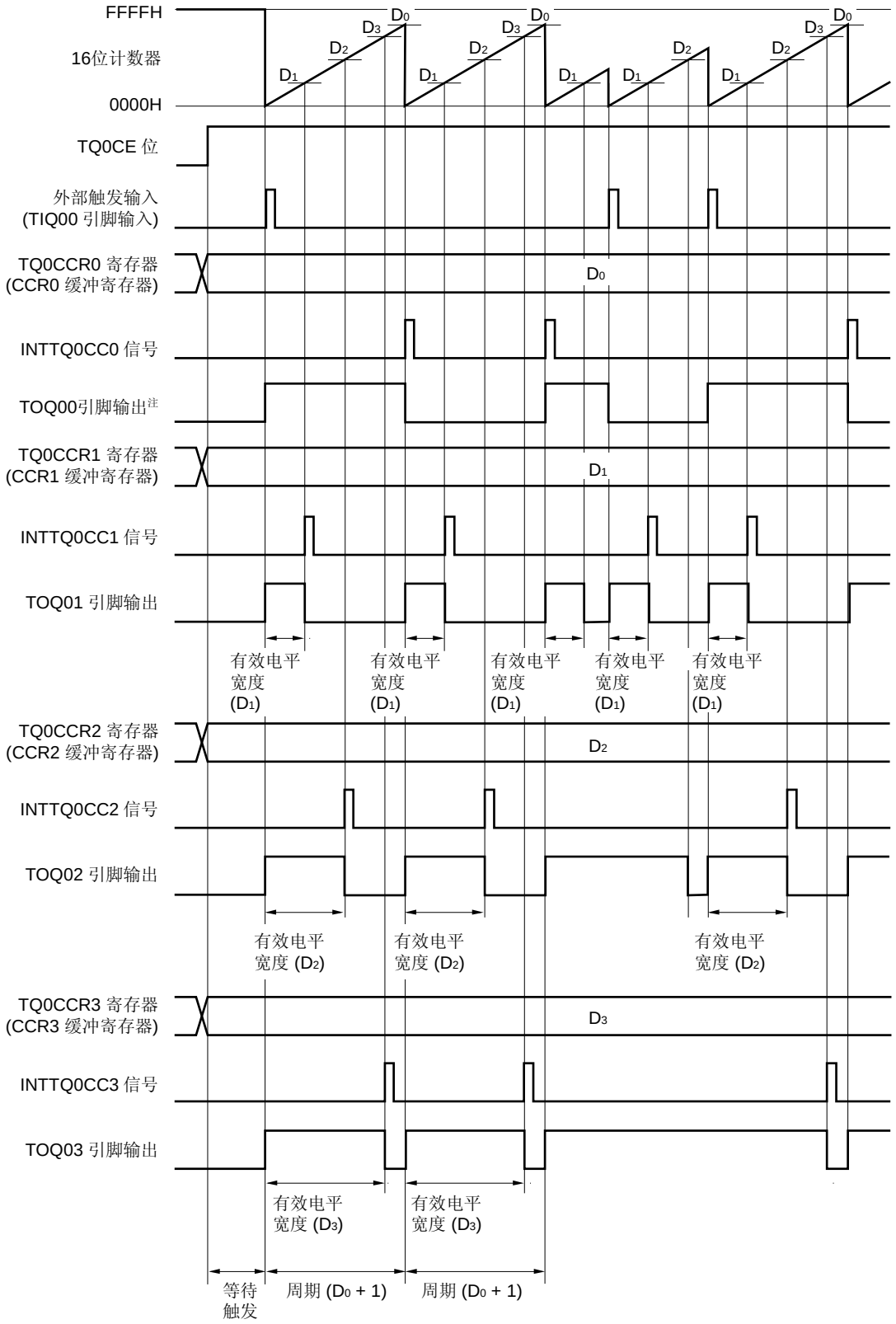


图 8-27. 外部触发脉冲输出模式的基本操作时序



注 TOQ00 引脚的输出也可以用作 TIQ00 引脚的输入。当使用 TOQ00 引脚的输出也可以用作 TIQ00 引脚的输入时，应使用软件触发，而不用外部触发。

当 TQ0CE 位设置为 1 时，TMQ0 等待触发。当触发产生时，16 位计数器从 FFFFH 清零为 0000H，同时开始递增计数，并从 TOQ0k 引脚输出一个 PWM 波形。如果在计数运行中再次产生触发，则计数器清为 0000H 并重新开始计数，且 TOQ00 引脚输出反相。（在发生触发信号时，不管当前状态如何（高/低），TOQ0k 引脚都会输出一个高电平。

PWM 波形的有效电平宽度、周期、占空比的计算如下。

有效电平宽度 =（TQ0CCRk 寄存器的设置值）× 计数时钟周期
周期 =（TQ0CCR0 寄存器的设置值＋1）× 计数时钟周期
占空比系数 =（TQ0CCRk 寄存器的设置值）/（TQ0CCR0 寄存器的设置值＋1）

当 16 位计数器在其计数值与 CCR0 缓冲寄存器的值匹配之后，进行下一次计数时就会同步产生比较匹配中断请求信号 INTTQ0CC0，且 16 位计数器清零为 0000H。当 16 位计数器的计数值与 CCRk 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTQ0CCk。

外部触发输入信号的有效边沿，或将软件触发（TQ0CTL1.TQ0EST 位）设为 1，都可以用作触发信号。

备注 k＝1 至 3

图 8-28. 外部触发脉冲输出模式的寄存器设置（1/3）

（a）TMQ0 控制寄存器 0（TQ0CTL0）

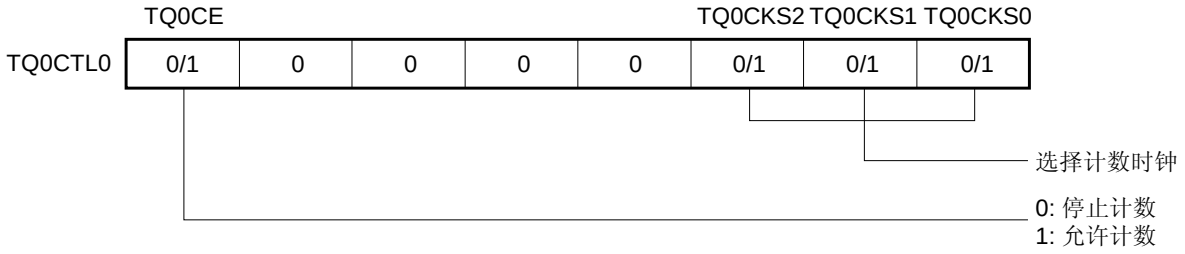


图 8-28. 外部触发脉冲输出模式的寄存器设置（2/3）

（b）TMQ0 控制寄存器 1（TQ0CTL1）

	TQ0EST	TQ0EEE			TQ0MD2	TQ0MD1	TQ0MD0
TQ0CTL1	0	0/1	0	0	0	0	1	0

0, 1, 0:
外部触发
脉冲输出模式

写入1，产生
一个软件触发

（c）TMQ0 输入/输出控制寄存器 0（TQ0IOC0）

	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0IOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1注	0/1注

0: 禁止TOQ00引脚输出
1: 允许 TOQ00引脚输出

禁止TOQ00引脚操作时
输出电平的设置
0: 低电平
1: 高电平

0: 禁止TOQ01引脚输出
1: 允许 TOQ01引脚输出

指定TOQ01引脚输出
的有效电平
0: 高电平有效
1: 低电平有效

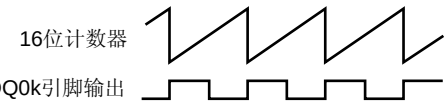
0: 禁止TOQ02引脚输出
1: 允许 TOQ02引脚输出

指定TOQ02引脚输出
的有效电平
0: 高电平有效
1: 低电平有效

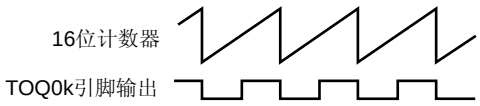
0: 禁止TOQ03引脚输出
1: 允许 TOQ03引脚输出

指定TOQ03引脚输出
的有效电平
0: 高电平有效
1: 低电平有效

· 当TQ0OLk 位 = 0时



· 当TQ0OLk 位 = 1时



注 在外部触发脉冲输出模式下，不使用 TOQ00 引脚时，将该位清除为 0。

图 8-18. 外部触发脉冲输出模式的寄存器设置（3/3）

（d）TMQ0 输入/输出控制寄存器 2（TQ0IOC2）

					TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
TQ0IOC2	0	0	0	0	0	0	0/1	0/1

这些位选择
外部触发输入
的有效沿

（e）TMQ0 计数器读取缓冲寄存器（TQ0CNT）

通过读取该寄存器，可以读出 16 位计数器的值。

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

如果将 TQ0CCR0 寄存器的值设置为 D_0，将 TQ0CCR1 寄存器的值设置为 D_1，将 TQ0CCR2 寄存器的值设置为 D_2，将 TQ0CCR3 寄存器的值设置为 D_3，则 PWM 波形的周期和有效电平的计算如下：

PWM 波形周期＝（D_0＋1）× 计数时钟周期
TOQ01 引脚的 PWM 波形有效电平宽度＝D_1 × 计数时钟周期
TOQ02 引脚的 PWM 波形有效电平宽度＝D_2 × 计数时钟周期
TOQ03 引脚的 PWM 波形有效电平宽度＝D_3 × 计数时钟周期

备注
1. 在外部触发脉冲输出模式下，不使用 TMQ0 输入/输出控制寄存器 1（TQ0IOC1）和 TMQ0 选项寄存器 0（TQ0OPT0）。
2. 通过写入 TMQ0 捕获/比较寄存器 1（TQ0CCR1），可以使能 TMQ0 捕获/比较寄存器（TQ0CCR2）和 TMQ0 捕获/比较寄存器 3（TQ0CCR3）的更新。

（1）外部触发脉冲输出模式的操作

图 8-29. 外部触发脉冲输出模式的时序和操作处理（1/2）

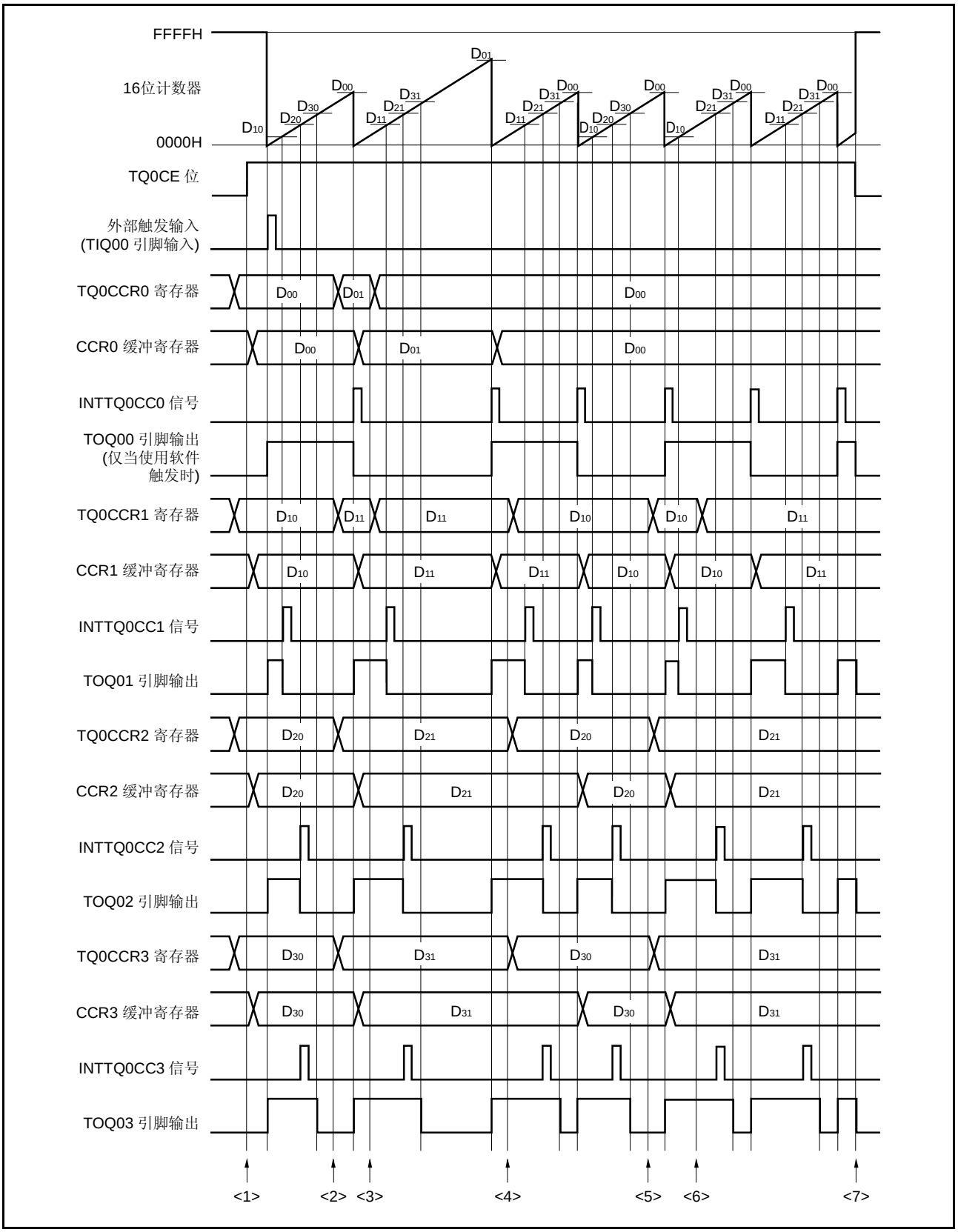
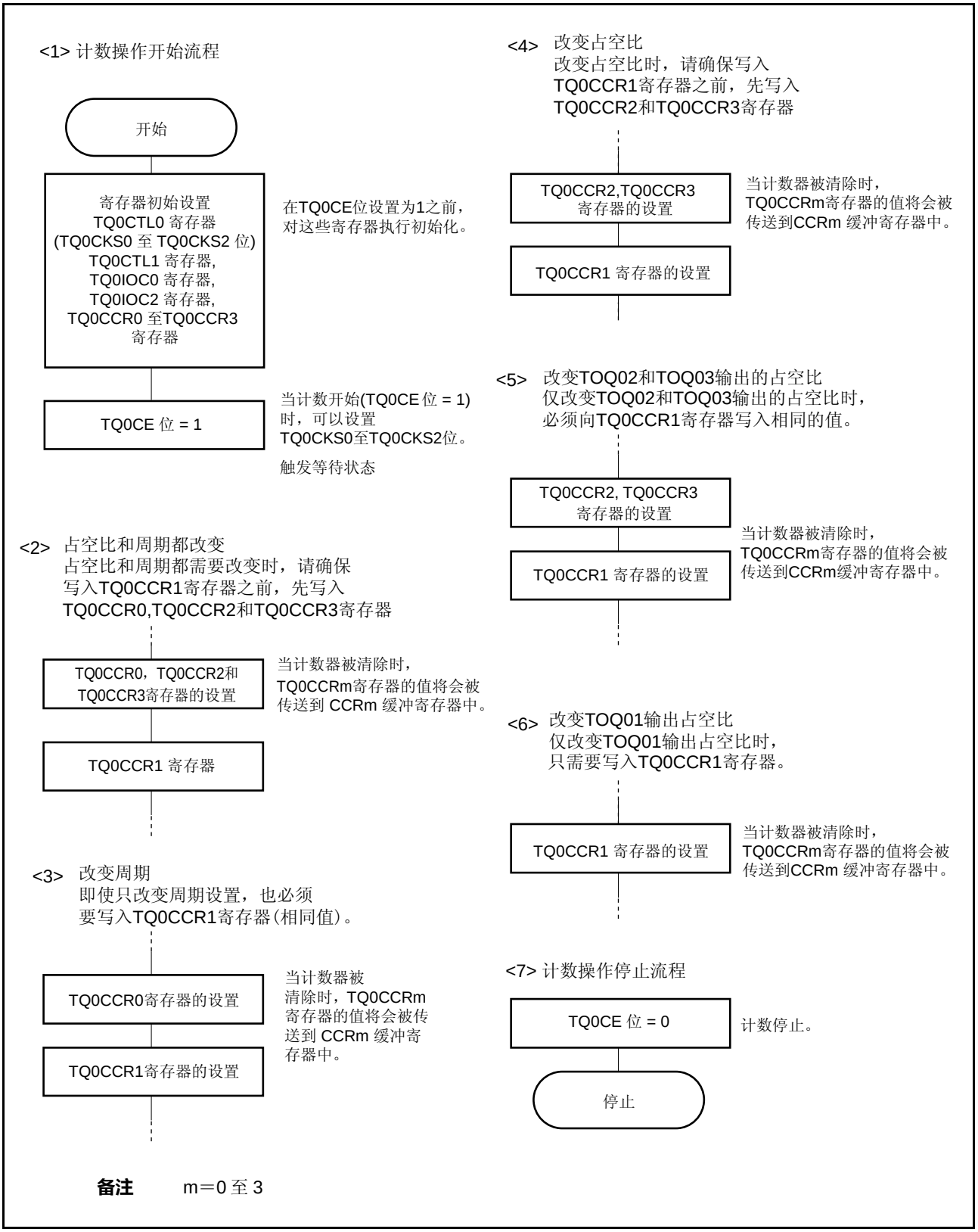


图 8-29. 外部触发脉冲输出模式的时序和操作处理（2/2）



备注 m＝0 至 3

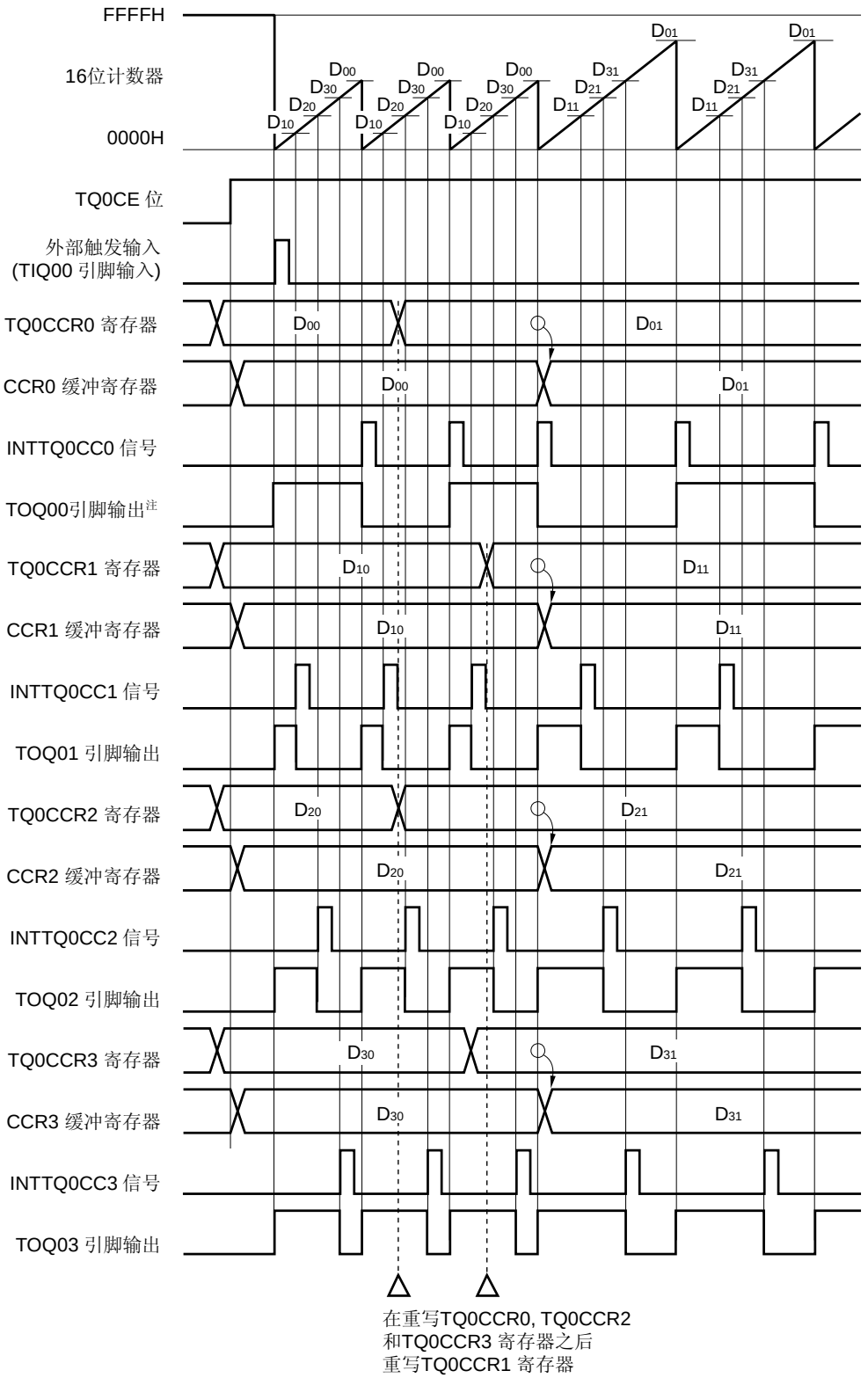
（2）使用外部触发脉冲输出模式

在外部触发脉冲输出模式下，如果改变 PWM 波形的描述如下。

（a）当计数器正在递增时改变 PWM 波形

计数器计数过程中，若要改变 PWM 波形，应在改变波形设置之后重新写入 TQ0CCR1 寄存器。在检测到 INTTQ0CC0 信号以后，在重写 TQ0CCRk 寄存器之前要先写入 TQ0CCR1 寄存器。

图 8-30. 计数器计数过程中改变 PWM 波形



注 TOQ00 引脚的输出也可以用作 TIQ00 引脚的输入。当使用 TOQ00 引脚的输出也可以用作 TIQ00 引脚的输入时，应使用软件触发，而不用外部触发。

为了将数据从 TQ0CCRm 寄存器传送到 CCRm 缓冲寄存器，必须写入 TQ0CCR1 寄存器。
在数据写入 TQ0CCR1 寄存器后，在对 16 位计数器清零的同时，写入到 TQ0CCRm 寄存器的值会被传送到 CCRm 缓冲寄存器，并作为与 16 位计数器进行比较的值。

<1> 如果 PWM 波形的周期和有效电平宽度都要改变，先将周期设定到 TQ0CCR0 寄存器，在设置有效电平宽度到 TQ0CCR1 寄存器之前，先将有效电平宽度设定到 TQ0CCR2 和 TQ0CCR3 寄存器。
<2> 如果只改变 PWM 波形的周期，应先把周期设置到 TQ0CCR0 寄存器，然后再将相同的值写入 TQ0CCR1 寄存器（也就是说，与之前指定 TQ0CCR1 寄存器的值相同）。
<3> 如果只改变 PWM 波形的有效电平宽度（占空比系数），需要先向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平宽度，然后再向 TQ0CCR1 寄存器设置有效电平宽度。
<4> 如果只改变 TOQ01 引脚输出的 PWM 波形的有效电平宽度（占空比系数），仅需要设置 TQ0CCR1 寄存器。
<5> 如果只改变 TOQ02 和 TOQ03 引脚输出的 PWM 波形的有效电平宽度（占空比系数），要先向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平宽度，然后再将相同的值写入 TQ0CCR1 寄存器（也就是说，与之前指定 TQ0CCR1 寄存器的值相同）。

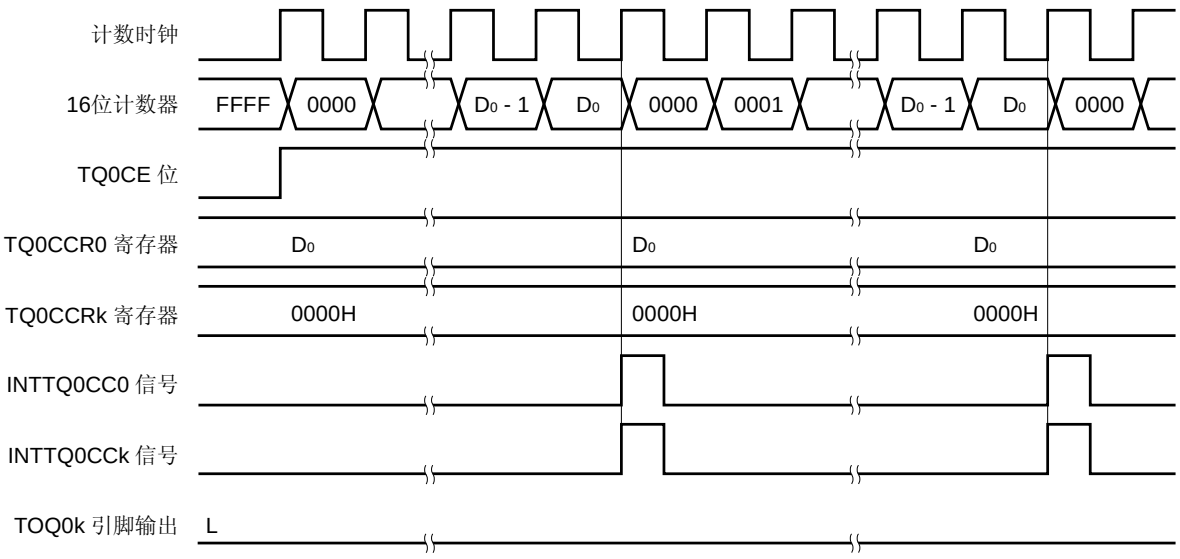
注意事项 在写入 TQ0CCR1 寄存器之后，要重写 TQ0CCR0 至 TQ0CCR3 寄存器，应该在 INTTQ0CC0 信号产生之后进行。否则，CCRm 缓冲寄存器的值可能不确定，因为把数据从 TQ0CCRm 寄存器传送到 CCRm 缓冲寄存器的时序可能会与写入 TQ0CCRm 寄存器的时序发生冲突。

备注 m＝0 至 3

（b）PWM 波形的 0%和 100% 输出

为了输出 0%的波形，需要将 TQ0CCRk 寄存器设置为 0000H。请注意，如果 TQ0CCR0 寄存器的设定值为 FFFFH，则会在每个周期产生 INTTQ0CCk 中断请求信号。

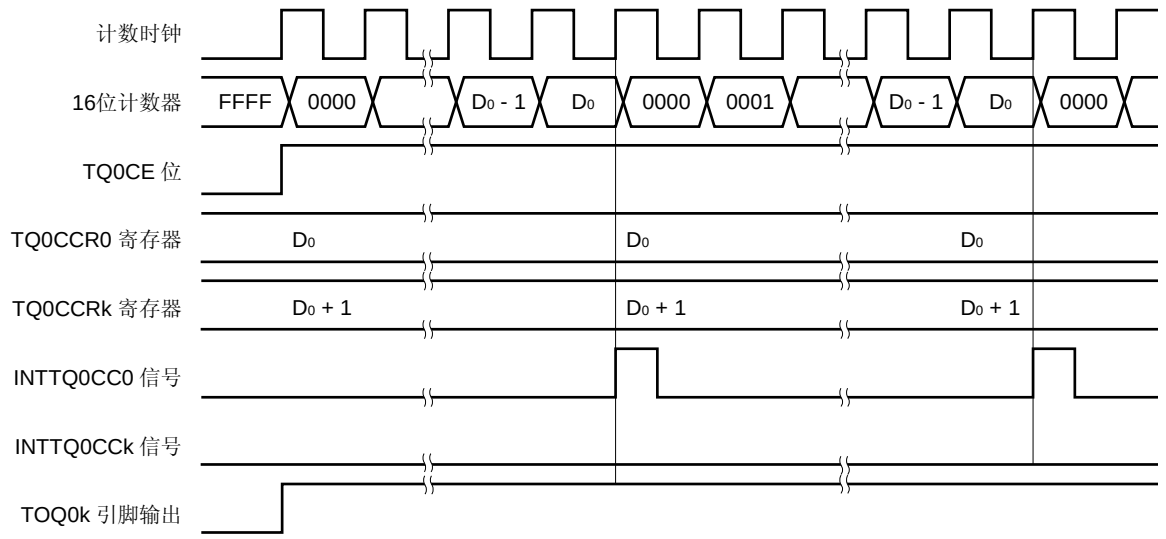
图 8-31. 输出 0% PWM 波形



备注 k＝1 至 3

为了输出 100%的波形，需要将 TQ0CCRk 寄存器的值设为：（TQ0CCR0 寄存器的值＋1）。如果 TQ0CCR0 寄存器的设定值为 FFFFH，则不会产生 100%波形输出。

图 8-32. 输出 100% PWM 波形

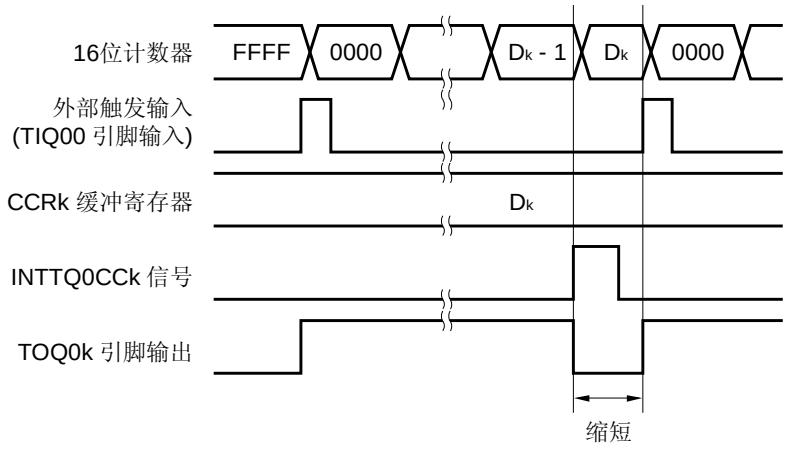


备注 k＝1 至 3

（c）检测到触发信号紧邻 INTTQ0CCk 产生之前或之后

如果在 INTTQ0CCk 信号产生后，立刻检测到触发信号，则 16 位计数器立刻清零为 0000H，TOQ0k 引脚的输出信号被设置为有效电平，计数器继续递增计数。结果，PWM 波形的无效时段被缩短。

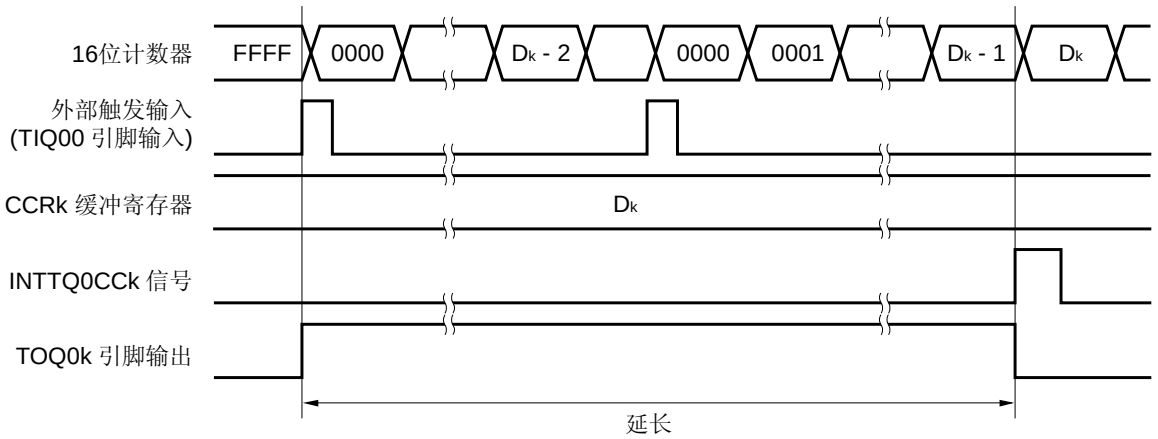
图 8-33. 检测到触发信号紧邻 INTTQ0CCk 信号产生之后



备注 k＝1 至 3

如果紧邻在 INTTQ0CCk 信号产生之前检测到触发信号，则不会产生 INTTQ0CCk 信号，而 16 位计数器清零为 0000H，递增计数继续进行。TOQ0k 引脚的输出信号仍然保持有效。结果，PWM 波形的有效时段被延长。

图 8-34. 检测到触发信号紧邻 INTTQ0CCk 信号产生之前

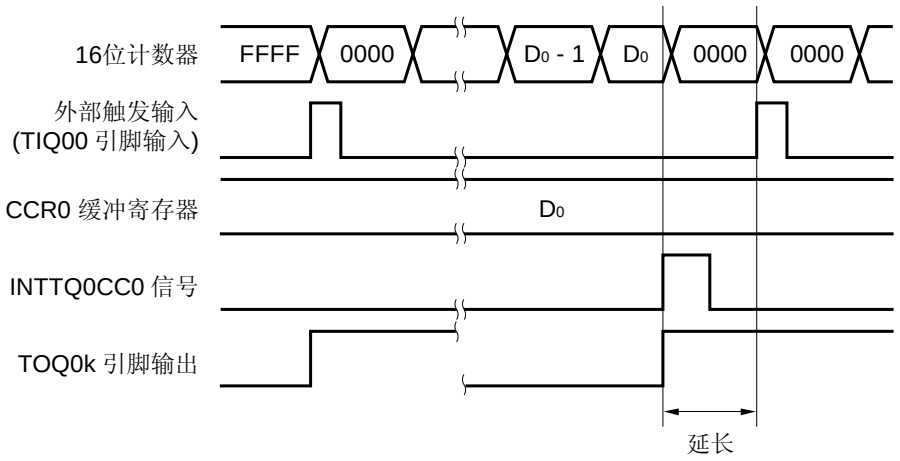


备注 k＝1 至 3

（d）检测到触发信号紧邻 INTTQ0CC0 产生之前或之后

如果在 INTTQ0CC0 信号产生后，立刻检测到触发信号，则 16 位计数器立刻清零为 0000H，并继续递增计数。因此，TOQ0k 引脚的有效时段被延长，延长长度为 INTTQ0CC0 信号产生时刻至检测到触发信号时刻的时间间隔。

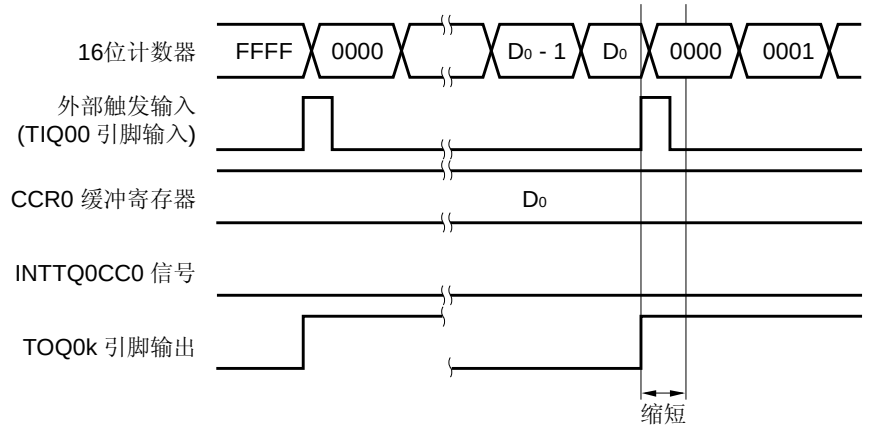
图 8-35. 检测到触发信号紧邻 INTTQ0CC0 信号产生之后



备注 k＝1 至 3

如果紧邻在 INTTQ0CC0 信号产生之前检测到触发信号，则不会产生 INTTQ0CC0 信号。16 位计数器清零为 0000H，TOQ0k 引脚输出被设置为有效电平，计数器继续递增计数。结果，PWM 波形的无效时段被缩短。

图 8-36. 检测到触发信号紧邻 INTTQ0CC0 信号产生之前

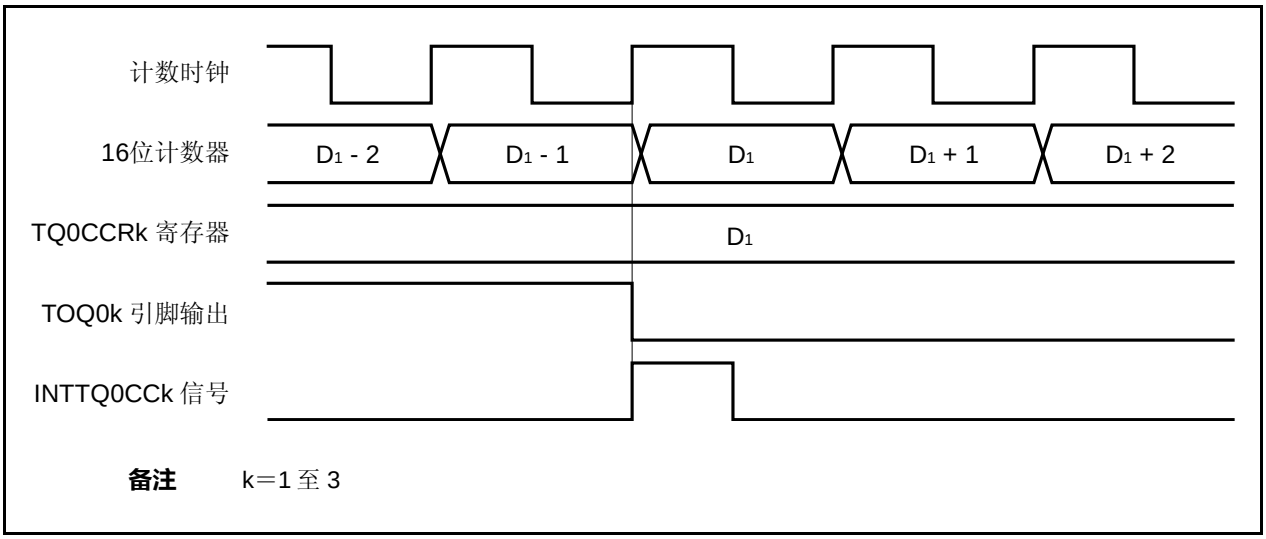


备注 k＝1 至 3

（e）比较匹配中断请求信号（INTTQ0CCk）的发生时序

在外部触发脉冲输出模式下，当 16 位计数器的计数值与 TQ0CCRk 缓冲寄存器的值匹配时，产生 INTTQ0CCk 中断请求信号。

图 8-37. 比较匹配中断请求信号（INTTQ0CCk）的发生时序



8.4.4 单次触发脉冲输出模式（TQ0MD2 至TQ0MD0 位 = 011）

在单次触发脉冲输出模式下，当 TQ0CTL0.TQ0CE 位被置为 1 时，TMQ0 等待触发。在检测到外部触发输入的有效边沿时，TMQ0 开始计数，同时从 TOQ01 至 TOQ03 引脚输出一个单次触发脉冲。

除外部触发外，也可以生成软件触发信号来输出该脉冲。在使用软件触发时，TOQ00 引脚在 16 位计数器计数时输出有效电平，在计数器停止（等待触发）时输出无效电平。

备注 **1.** 关于如何设置 TIQ00 和 TOQ00 至 TOQ03 引脚，参见**表 8-2 TMQ0 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。

2. 关于如何使能 INTTQ0CC0 至 INTTQ0CC3 中断信号，参见**第 22 章 中断服务/异常处理。**

图 8-38. 单次触发脉冲输出模式下的 TMQ0 配置

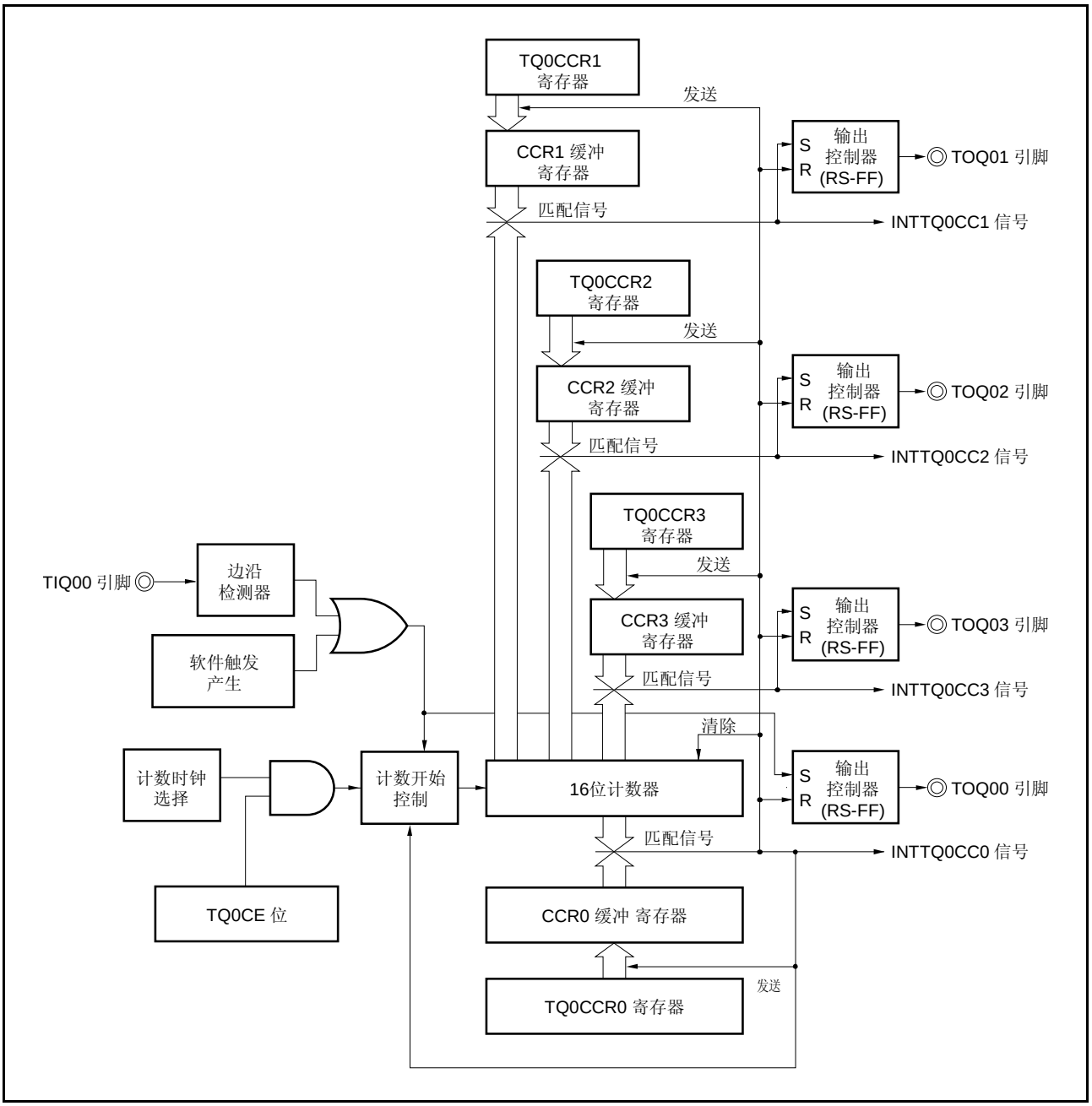
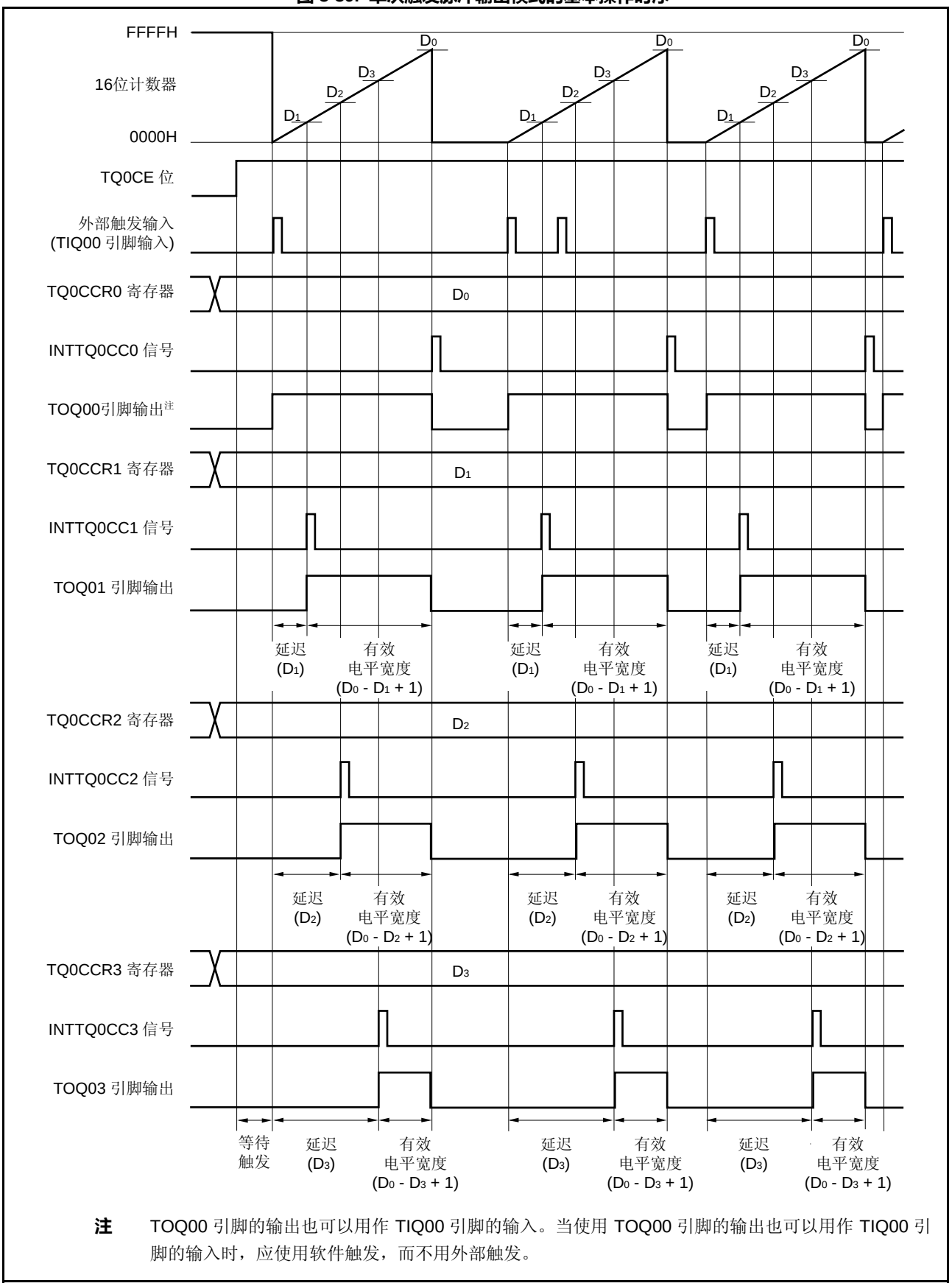


图 8-39. 单次触发脉冲输出模式的基本操作时序



注 TOQ00 引脚的输出也可以用作 TIQ00 引脚的输入。当使用 TOQ00 引脚的输出也可以用作 TIQ00 引脚的输入时，应使用软件触发，而不用外部触发。

当 TQ0CE 位置为 1 时，TMQ0 等待触发。在触发信号发生时，16 位计数器从 FFFFH 清零为 0000H，开始进行计数，并从 TOQ0k 引脚输出一个单次触发脉冲。在输出单次触发脉冲后，16 位计数器被设置为 0000H，计数停止，并重新等待触发。如果在单次触发脉冲输出期间再次产生触发信号，则该触发信号被忽略。

单次触发脉冲的输出延迟时间和有效电平宽度可计算如下：

输出延迟时间＝（TQ0CCRk 寄存器的设定值）× 计数时钟周期
有效电平宽度＝（TQ0CCR0 寄存器的设定值 – TQ0CCRk 寄存器的设定值 ＋1）× 计数时钟周期

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配之后，产生比较匹配中断请求信号 INTTQ0CC0。当 16 位计数器的计数值与 CCRk 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTQ0CCk。

外部触发输入信号的有效边沿，或将软件触发（TQ0CTL1.TQ0EST 位）信号设置为 1，都可被用作触发信号。

备注 k＝1 至 3

图 8-40. 单次触发脉冲输出模式时的寄存器设置（1/3）

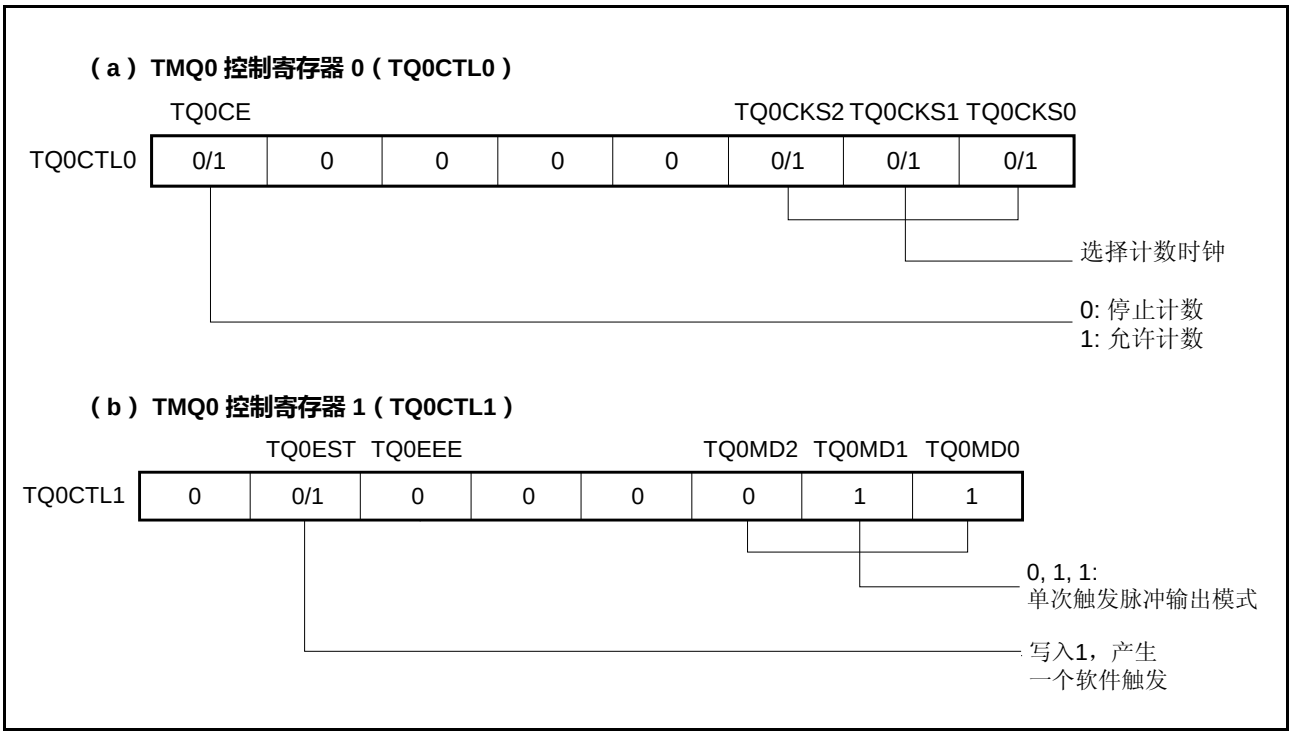


图 8-40. 单次触发脉冲输出模式时的寄存器设置（2/3）

（c）TMQ0 输入/输出控制寄存器 0（TQ0IOC0）

	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0IOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1注	0/1注

0: 禁止TOQ00引脚输出
1: 允许 TOQ00引脚输出

禁止TOQ00引脚的操作时输出电平的设置
0: 低电平
1: 高电平

0: 禁止TOQ01引脚输出
1: 允许 TOQ01引脚输出

指定TOQ01引脚输出的有效电平
0: 高电平有效
1: 低电平有效

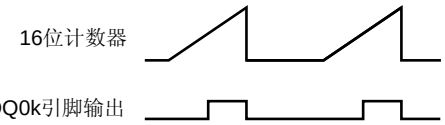
0: 禁止TOQ02引脚输出
1: 允许 TOQ02引脚输出

指定TOQ02引脚输出的有效电平
0: 高电平有效
1: 低电平有效

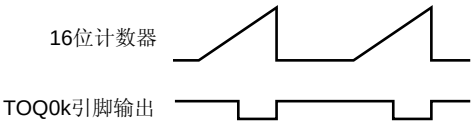
0: 禁止TOQ03引脚输出
1: 允许 TOQ03引脚输出

指定TOQ03引脚输出的有效电平
0: 高电平有效
1: 低电平有效

· 当TQ0OLk 位 = 0时



· 当 TQ0OLk 位 = 1时



（d）TMQ0 输入/输出控制寄存器 2（TQ0IOC2）

					TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
TQ0IOC2	0	0	0	0	0	0	0/1	0/1

这些位选择外部触发输入的有效沿

（e）TMQ0 计数器读取缓冲寄存器（TQ0CNT）

通过读取该寄存器，可以读出 16 位计数器的值。

注 在单次触发脉冲输出模式下，不使用 TOQ00 引脚时，将该位清除为 0。

图 8-40. 单次触发脉冲输出模式时的寄存器设置（3/3）

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

若把 TQ0CCR0 寄存器的值设置为 D_0，TQ0CCRk 寄存器的值设置为 D_k，则单次触发脉冲的有效电平宽度和输出延迟时间的计算如下。

单次触发脉冲有效电平宽度＝（$D_0 - D_k + 1$）× 计数时钟周期
单次触发脉冲输出延迟时间＝D_k × 计数时钟周期

注意事项 在单次触发脉冲输出模式下，如果 TQ0CCRk 寄存器的值大于 TQ0CCR0 寄存器的值，也不会从 TOQ0k 引脚输出单次触发脉冲。

备注
1. 在单次触发脉冲输出模式下，不使用 TMQ0 输入/输出控制寄存器 1（TQ0IOC1）和 TMQ0 选项寄存器 0（TQ0OPT0）。
2. k＝1 至 3

（1）单次触发脉冲输出模式下的操作

图 8-41. 单次触发脉冲输出模式的时序和操作处理（1/2）

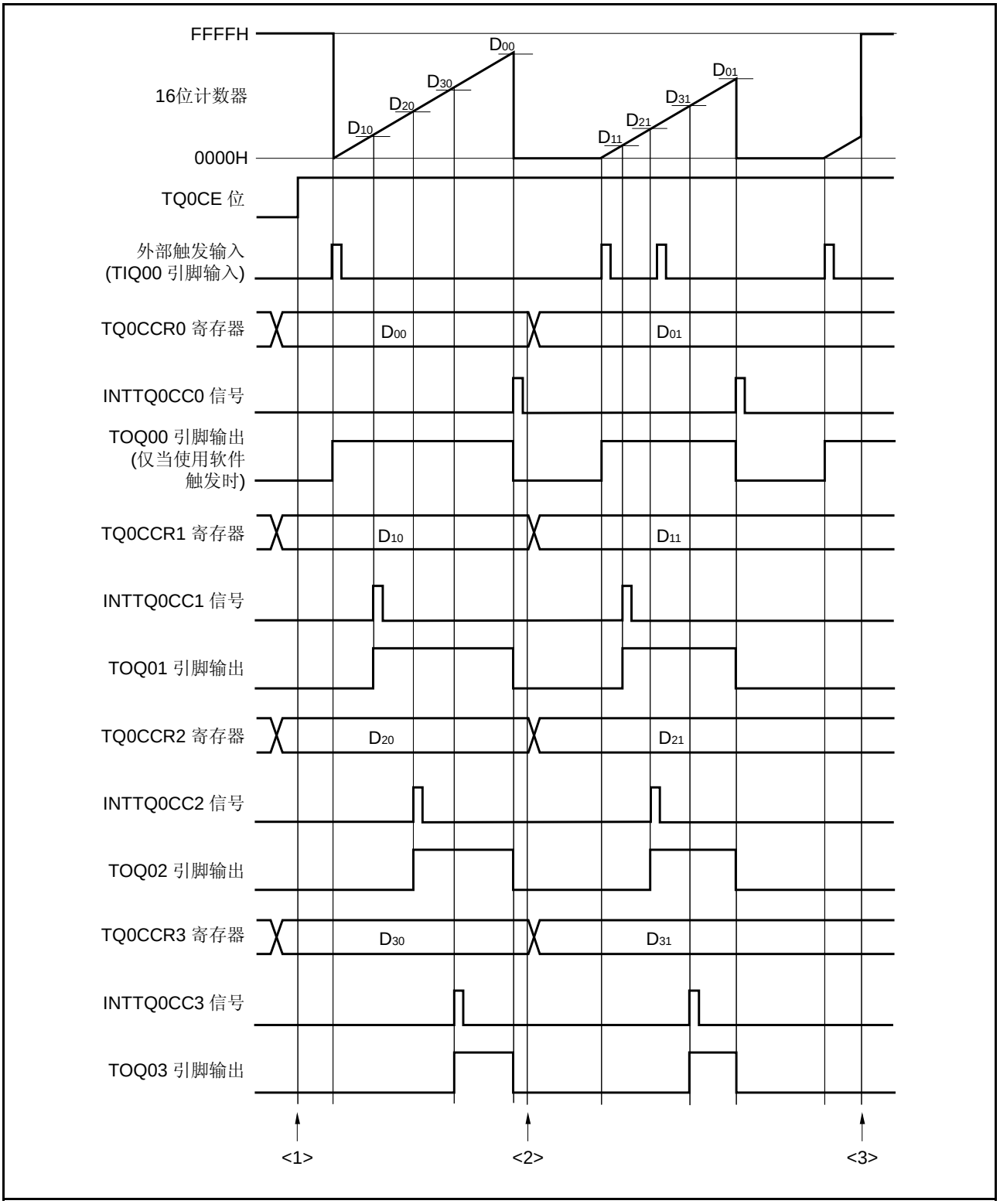
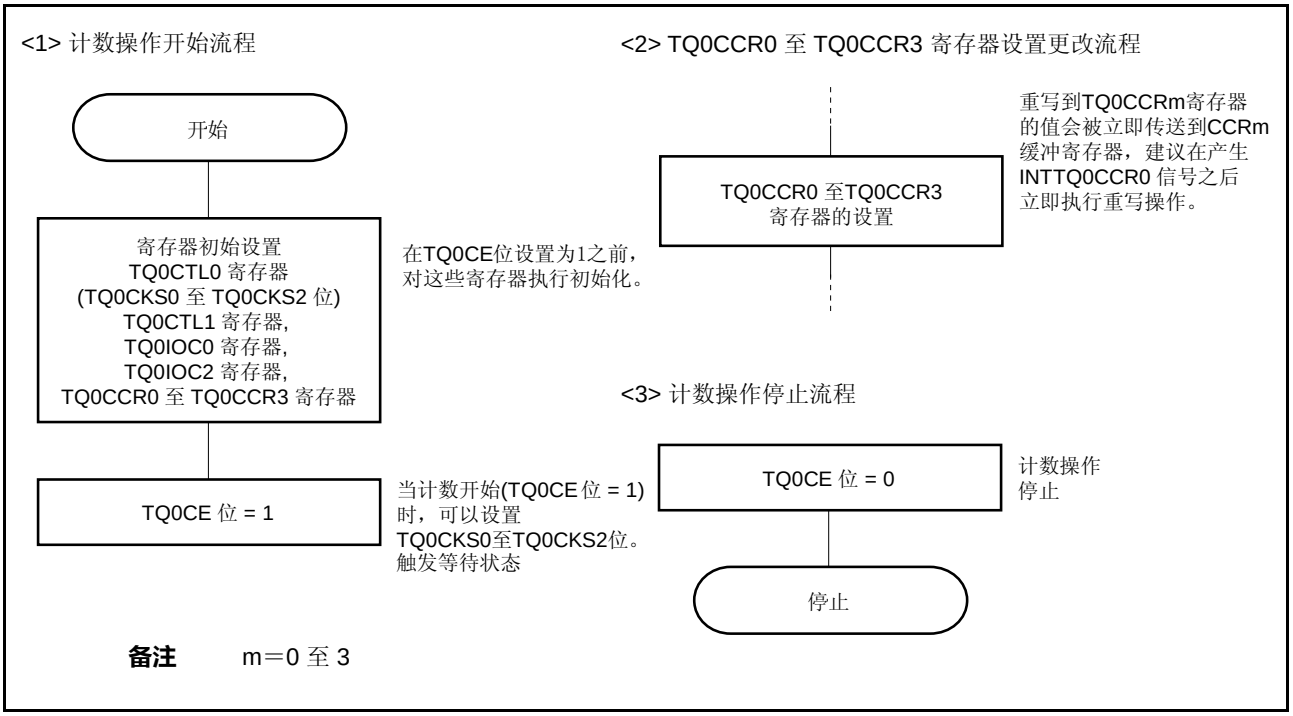


图 8-41. 单次触发脉冲输出模式的时序和操作处理（2/2）



（2）使用单次触发脉冲输出模式

（a）重写 TQ0CCRm 寄存器

要将 TQ0CCRm 寄存器的值改为比当前值较小的值，应首先停止计数，然后改变其设定值。
当 TQ0CCR0 寄存器的值从 D_{00} 改变为 D_{01} 时，且 TQ0CCRk 寄存器的值从 D_{K0} 改变为 D_{K1} 时，如果在下列条件时该寄存器被重写，则不会输出预期的单次触发脉冲。

条件 1 在重写 TQ0CCR0 寄存器时，如果：
$D_{00} > D_{01}$ 或
$D_{00} <$ 16 位计数器值 $< D_{01}$

在条件 1 的情况下，16 位计数器不会被清除，且将会在写入新值的周期内溢出。在第一次达到新写入的值(D_{01})的位置，计数器将会被清除。

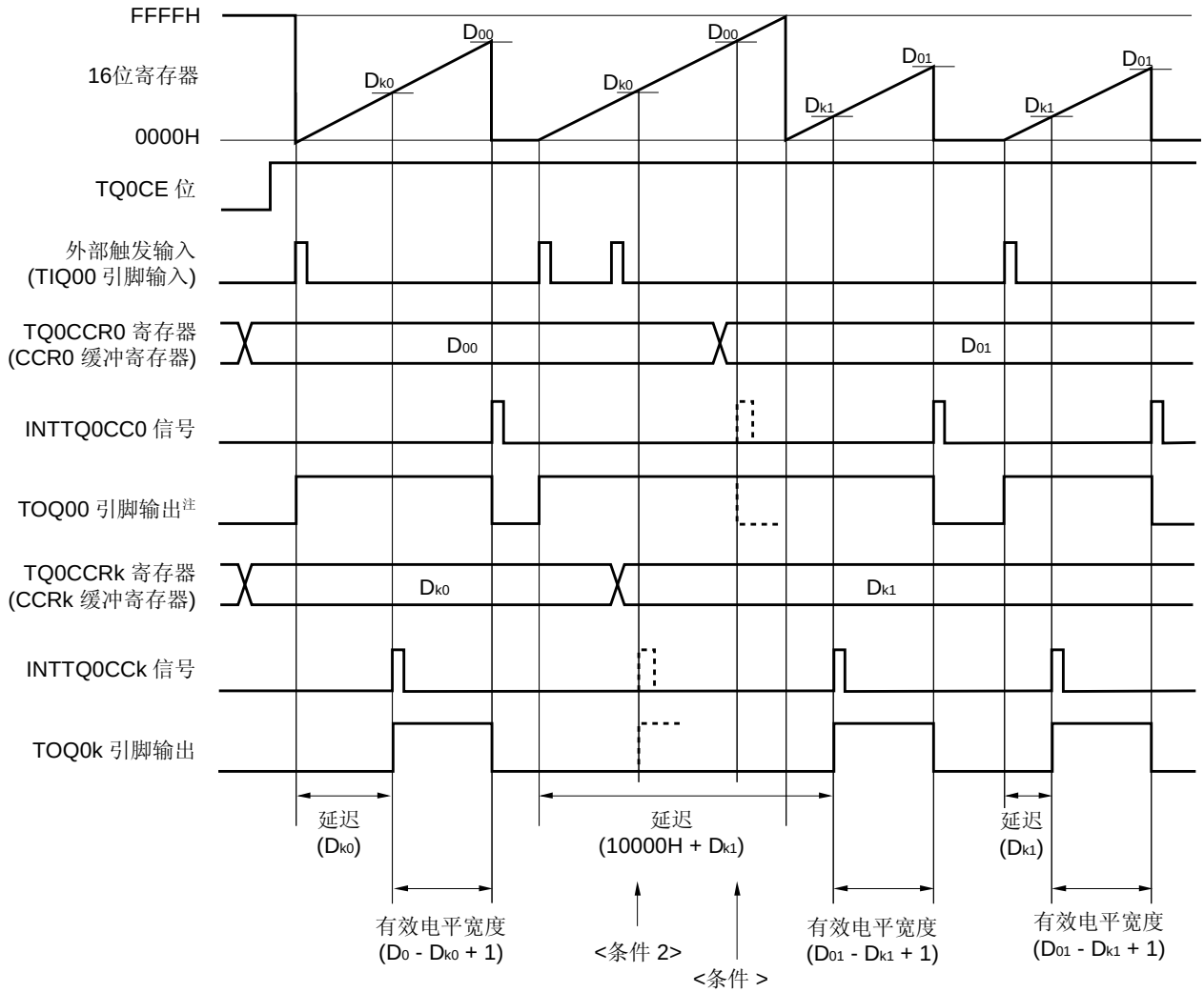
条件 2 在重写 TQ0CCRk 寄存器时，如果：
$D_{K0} > D_{K1}$ 或
$D_{K0} <$ 16 位计数器值 $< D_{K1}$

在条件 2 的情况下，在写入新值的周期内 TOQ0k 引脚输出不能被反转为有效电平。

在同一个周期内，当条件 1 和条件 2 都满足时的示例，如图 8-42 所示。
16 位计数器向上递增到 FFFFH，溢出，并再次从 0000H 开始递增计数。
当计数值与 D_{K1} 匹配时，计数器会产生 $INTTQ0CC_K$ 信号并使 TOQ0k 引脚输出被置为有效电平。接下来，当计数值与 D_{01} 匹配时，计数器产生 INTTQ0CC0 信号，使 TOQ0k 引脚输出置为无效电平，并停止计数。

备注 m = 0 至 3
K = 1 至 3

图 8-42. 重写 TQ0CCRm 寄存器



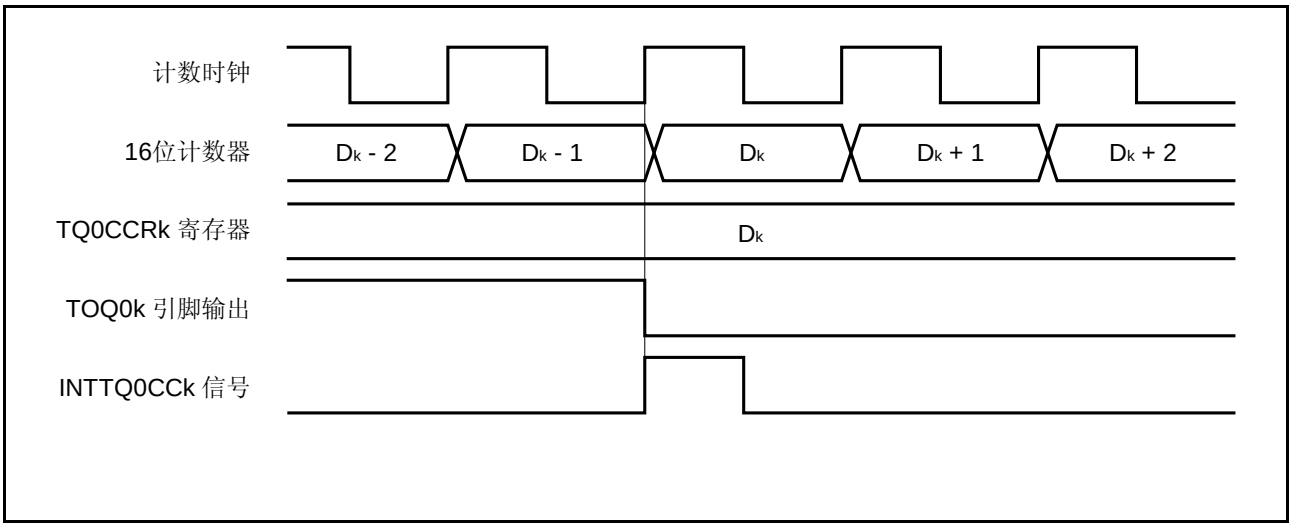
注 TOQ00 引脚的输出也可以用作 TIQ00 引脚的输入。当使用 TOQ00 引脚的输出也可以用作 TIQ00 引脚的输入时，应使用软件触发，而不用外部触发。

备注 m = 0 至 3
k = 1 至 3

（b）比较匹配中断请求信号（INTTQ0CCk）的发生时序

单次触发脉冲输出模式下，当 16 位计数器的计数值与 TQ0CCRk 寄存器的值匹配时，产生 INTTQ0CCk 中断请求信号。

图 8-43. 比较匹配中断信号（INTTQ0CCk）的发生时序



备注 k＝1 至 3

8.4.5 PWM输出模式（TQ0MD2 至TQ0MD0 位 = 100）

在 PWM 输出模式下，当 TQ0CTL0.TQ0CE 位被置为 1 时，PWM 波形从 TOQ01 至 TOQ03 引脚输出。

而且，由 TOQ00 引脚输出的脉冲，其半周期等于 PWM 的全周期。

备注 **1.** 关于如何设置 TIQ00 和 TOQ00 至 TOQ03 引脚，参见**表 8-2 TMQ0 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。

2. 关于如何使能 INTTQ0CC0 至 INTTQ0CC3 中断信号，参见**第 22 章 中断服务/异常处理。**

图 8-44. PWM 输出模式下的 TMQ0 配置

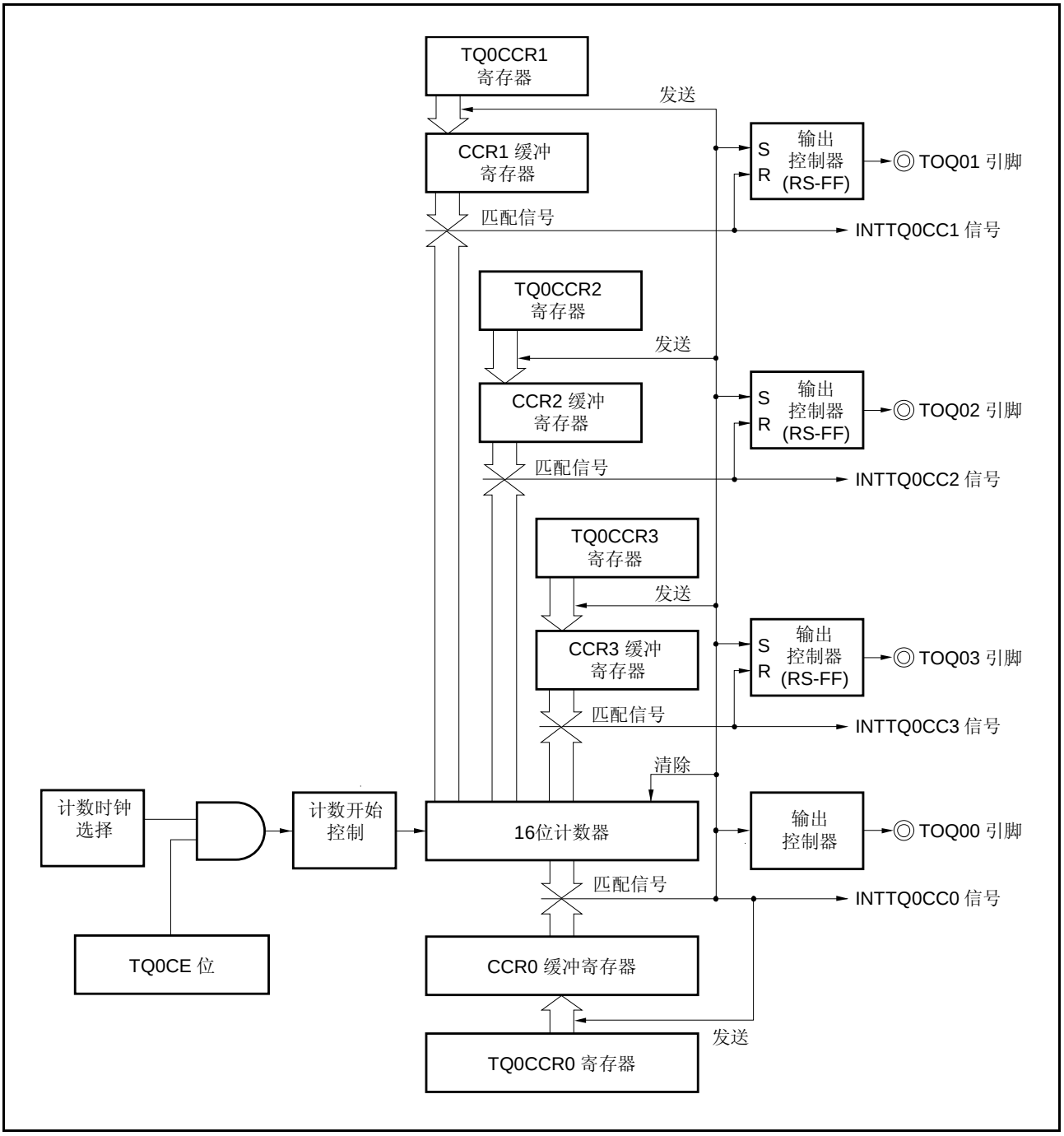
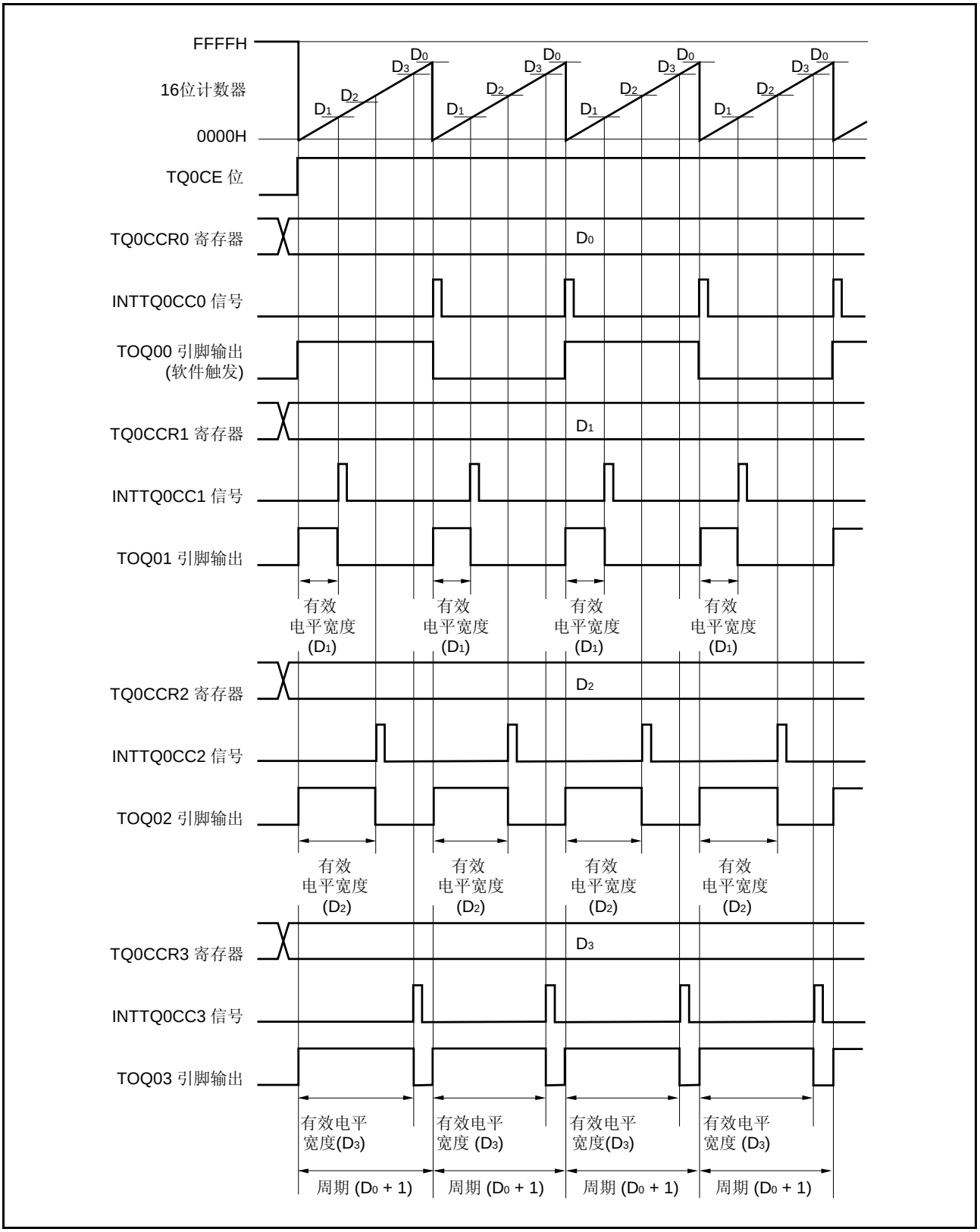


图 8-45. PWM 输出模式的基本操作时序



当 TQ0CE 位置为 1 时，16 位计数器从 FFFFH 清零为 0000H，递增计数开始，同时从 TOQ0k 引脚输出 PWM 波形。PWM 波形的有效电平宽度、周期、占空比系数的计算如下。

有效电平宽度＝（TQ0CCRk 寄存器的设定值）× 计数时钟周期
周期＝（TQ0CCR0 寄存器的设定值 ＋1）× 计数时钟周期
占空比系数＝（TQ0CCRk 寄存器的设定值）/（TQ0CCR0 寄存器的设定值 ＋1）

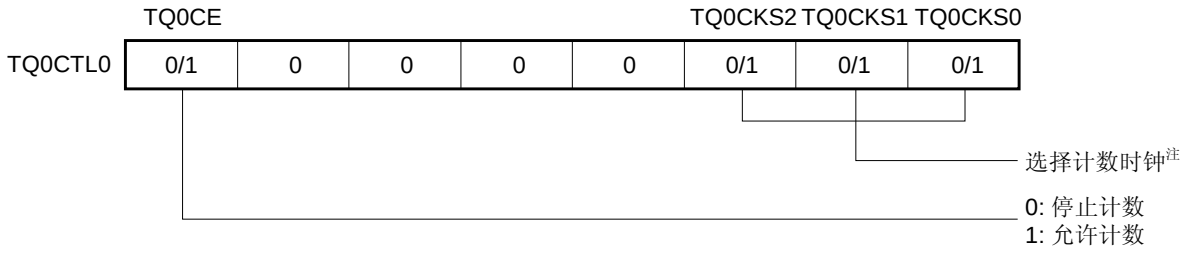
在计数操作时，可通过重写 TQ0CCRm 寄存器来改变 PWM 波形。当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配时，16 位计数器清零为 0000H 时，新写入的值会发生作用。

当 16 位计数器的计数值与 CCR0 缓冲寄存器的值匹配之后，进行下一次计数时，会同步产生比较匹配中断请求信号 INTTQ0CC0，且 16 位计数器清零为 0000H。当 16 位计数器的计数值与 CCRk 缓冲寄存器的值匹配时，会生成比较匹配中断请求信号 INTTQ0CCk。

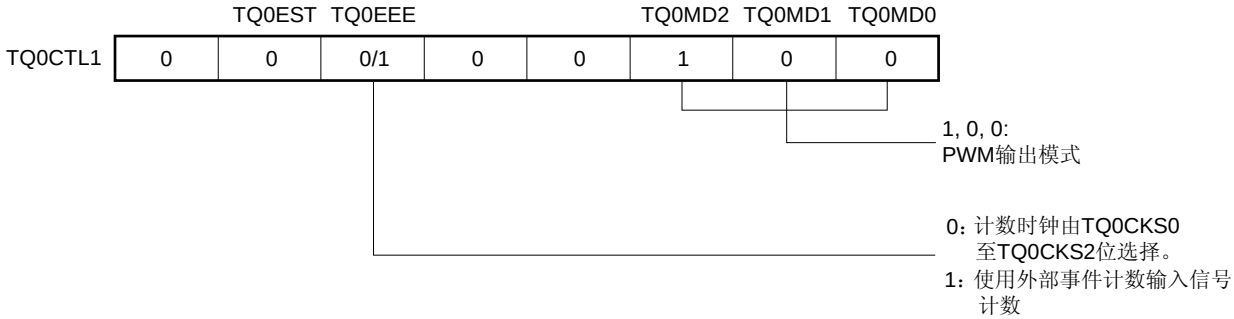
备注 k＝1 至 3
m＝0 至 3

图 8-46. PWM 输出模式时的寄存器设置（1/3）

（a）TMQ0 控制寄存器 0（TQ0CTL0）



（b）TMQ0 控制寄存器 1（TQ0CTL1）



注 当 TQ0CTL1.TQ0EEE 位＝1 时，这些位的设置无效。

图 8-46. PWM 输出模式时的寄存器设置（2/3）

（c）TMQ0 输入/输出控制寄存器 0（TQ0IOC0）

	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0IOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1注	0/1注

0: 禁止TOQ00引脚输出
1: 允许 TOQ00引脚输出

禁止TOQ00引脚操作时
输出电平的设置
0: 低电平
1: 高电平

0: 禁止TOQ01引脚输出
1: 允许 TOQ01引脚输出

指定TOQ01引脚输出
的有效电平
0: 高电平有效
1: 低电平有效

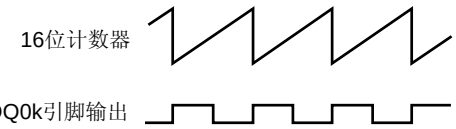
0: 禁止TOQ02引脚输出
1: 允许 TOQ02引脚输出

指定TOQ02引脚输出
的有效电平
0: 高电平有效
1: 低电平有效

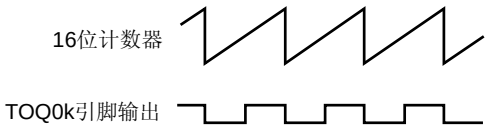
0: 禁止TOQ03引脚输出
1: 允许 TOQ03引脚输出

指定 TOQ03引脚输出
的有效电平
0: 高电平有效
1: 低电平有效

· 当 TQ0OLk 位 = 0时



· 当TQ0OLk 位 = 1时



（d）TMQ0 输入/输出控制寄存器 2（TQ0IOC2）

					TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
TQ0IOC2	0	0	0	0	0/1	0/1	0	0

选择外部事件
计数输入的有
效沿。

（e）TMQ0 计数器读取缓冲寄存器（TQ0CNT）

通过读取该寄存器，可以读出 16 位计数器的值。

注 在 PWM 输出模式下，不使用 TOQ00 引脚时，将该位清除为 0。

图 8-46. PWM 输出模式时的寄存器设置（3/3）

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

如果将 TQ0CCR0 寄存器的值设置为 D_0，将 TQ0CCR1 寄存器的值设置为 D_k，则 PWM 波形的周期和有效电平的计算如下：

PWM 波形周期＝（D_0 ＋1）× 计数时钟周期
PWM 波形有效电平宽度＝D_k × 计数时钟周期

备注
1. 在 PWM 输出模式下，不使用 TMQ0 输入/输出控制寄存器 1（TQ0IOC1）和 TMQ0 选项寄存器 0（TQ0OPT0）。
2. 通过写入 TMQ0 捕获/比较寄存器 1（TQ0CCR1），可以更新 TMQ0 捕获/比较寄存器（TQ0CCR2）和 TMQ0 捕获/比较寄存器 3（TQ0CCR3）。

（1）PWM 输出模式下的操作

图 8-47. PWM 输出模式的时序和操作处理（1/2）

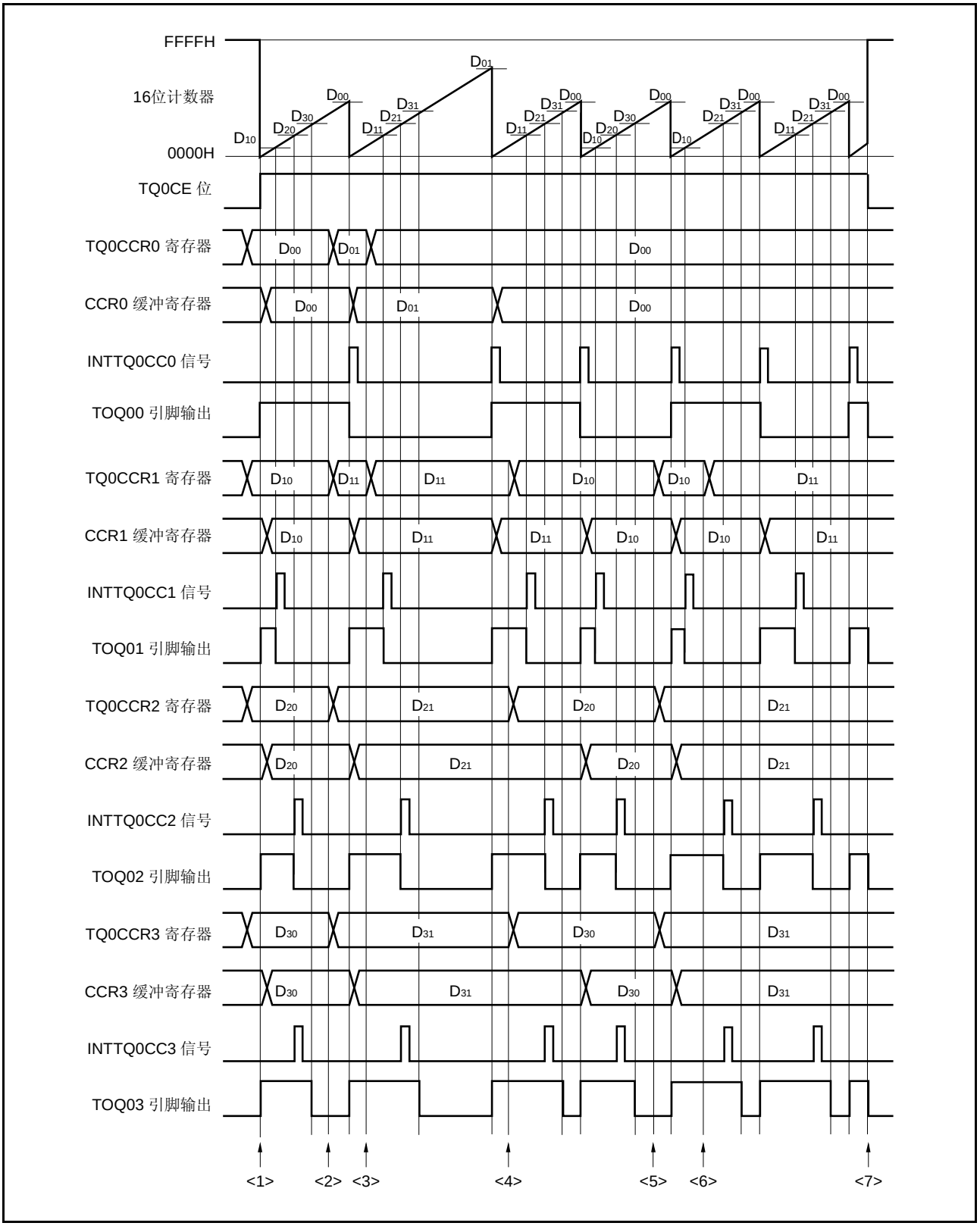
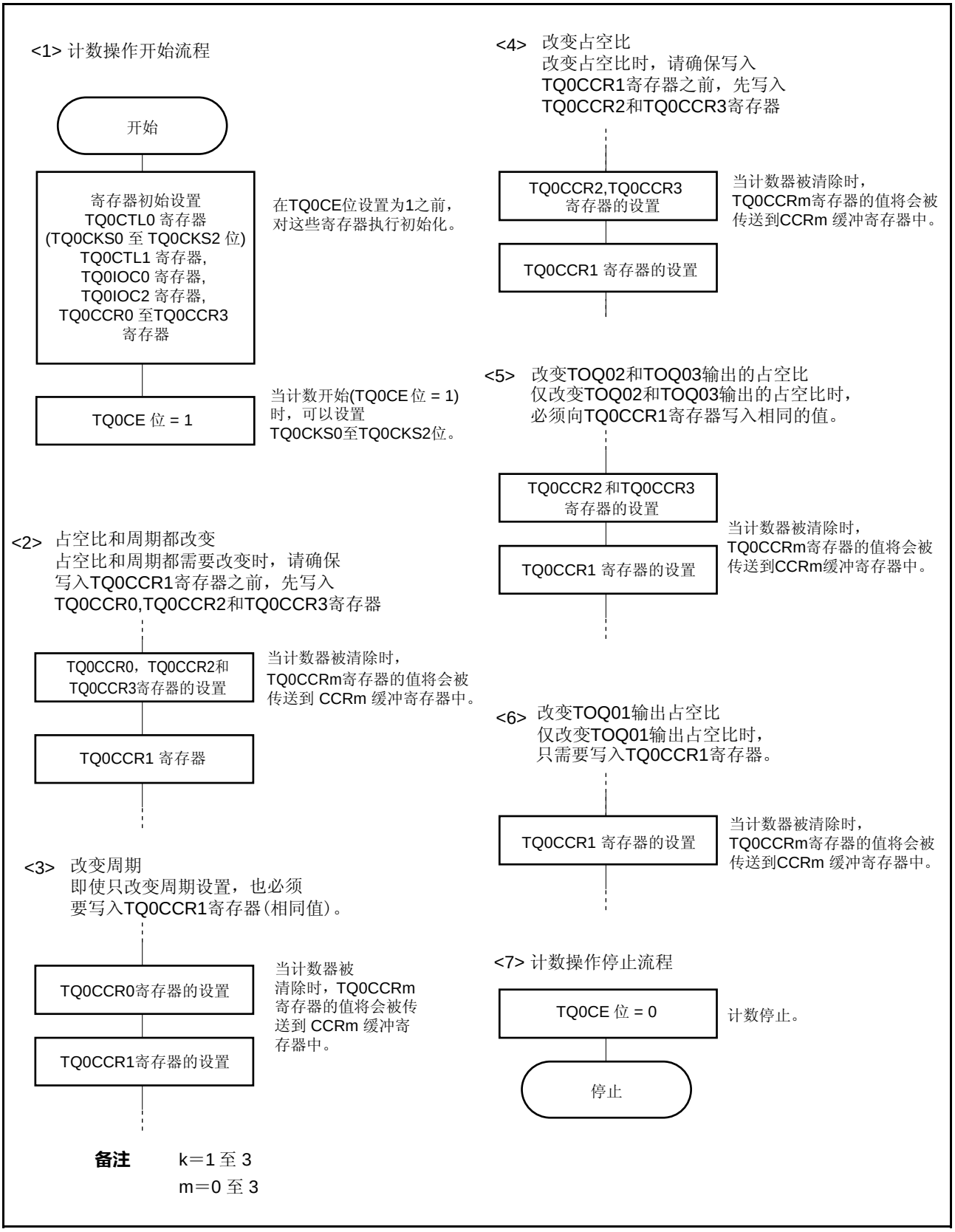


图 8-47. PWM 输出模式时的时序和操作处理（2/2）

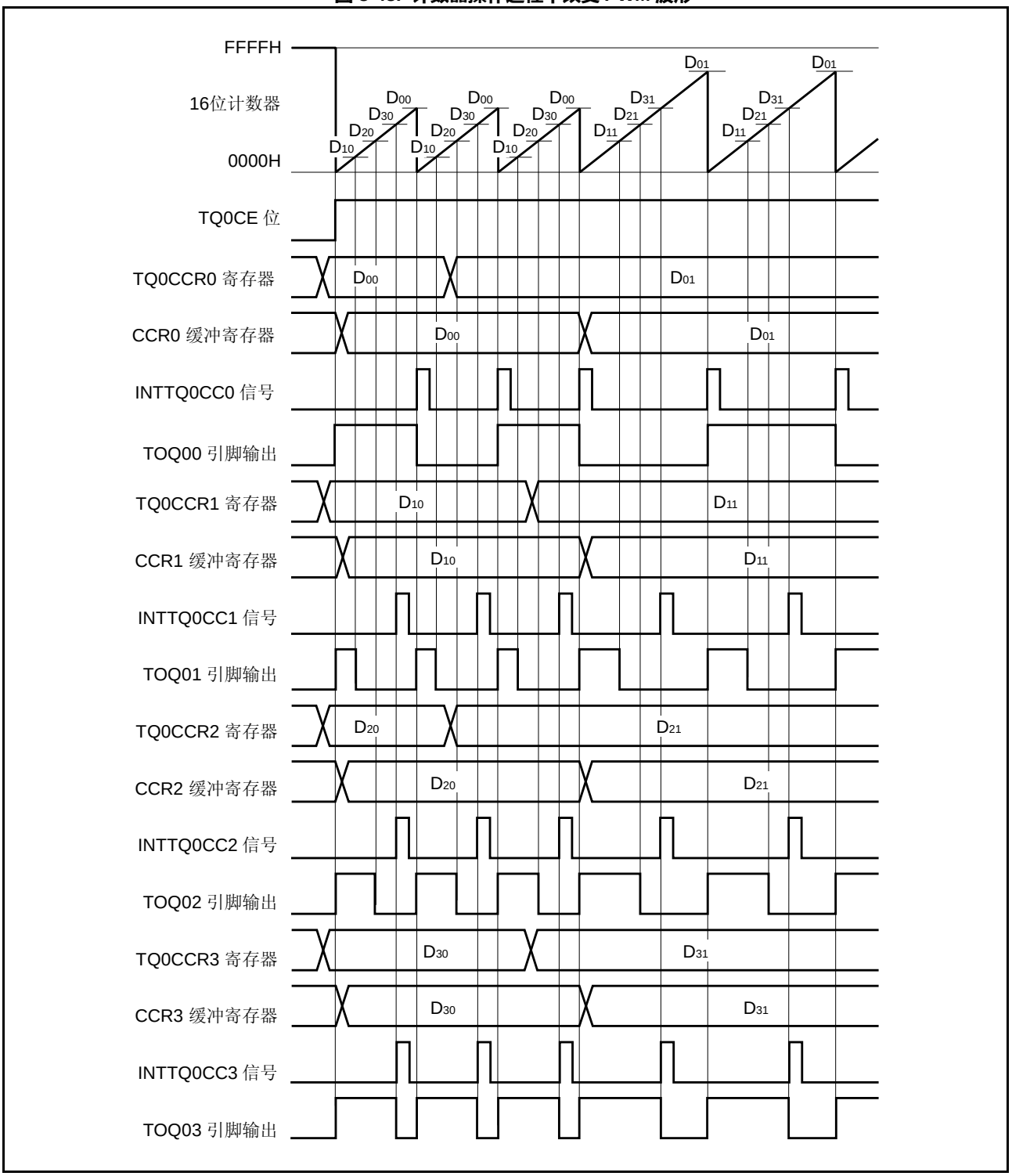


（2）使用 PWM 输出模式

（a）计数器操作过程中改变 PWM 波形

计数器操作过程中，若要改变 PWM 波形，应在改变波形设置之后写入 TQ0CCR1 寄存器。当检测到 INTTQ0CC1 信号以后，在写入 TQ0CCR1 寄存器之后，才能重写 TQ0CCRm 寄存器。

图 8-48. 计数器操作过程中改变 PWM 波形



要把数据从 TQ0CCRm 寄存器传送到 CCRm 缓冲寄存器，必须对 TQ0CCR1 寄存器进行写入操作。
当数据写入 TQ0CCR1 寄存器之后，在对 16 位计数器清零的同时，写入到 TQ0CCRm 寄存器的值被传送到 CCRm 缓冲寄存器，并作为与 16 位计数器进行比较的值。

<1> 如果 PWM 波形的周期和有效电平都要改变，应先将周期设定到 TQ0CCR0 寄存器，再将有效电平宽度设定到 TQ0CCR2 和 TQ0CCR3 寄存器，然后将有效电平设定到 TQ0CCR1 寄存器。
<2> 如果只改变 PWM 波形的周期，应先把周期设定给 TQ0CCR0 寄存器，然后将相同值（原值）写入 TQ0CCR1 寄存器（也就是说，与预先为 TQ0CCR1 寄存器指定的值相同）。
<3> 如果只改变 PWM 波形的有效电平宽度（占空比系数），需要先向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平宽度，然后再向 TQ0CCR1 寄存器设置有效电平宽度。
<4> 如果只改变 TOQ01 引脚输出的 PWM 波形的有效电平宽度（占空比系数），仅需要设置 TQ0CCR1 寄存器。
<5> 如果只改变 TOQ02 和 TOQ03 引脚输出的 PWM 波形的有效电平宽度（占空比系数），应先向 TQ0CCR2 和 TQ0CCR3 寄存器设置有效电平宽度，然后将相同的值写入 TQ0CCR1 寄存器。（也就是说，与预先为 TQ0CCR1 寄存器指定的值相同）。

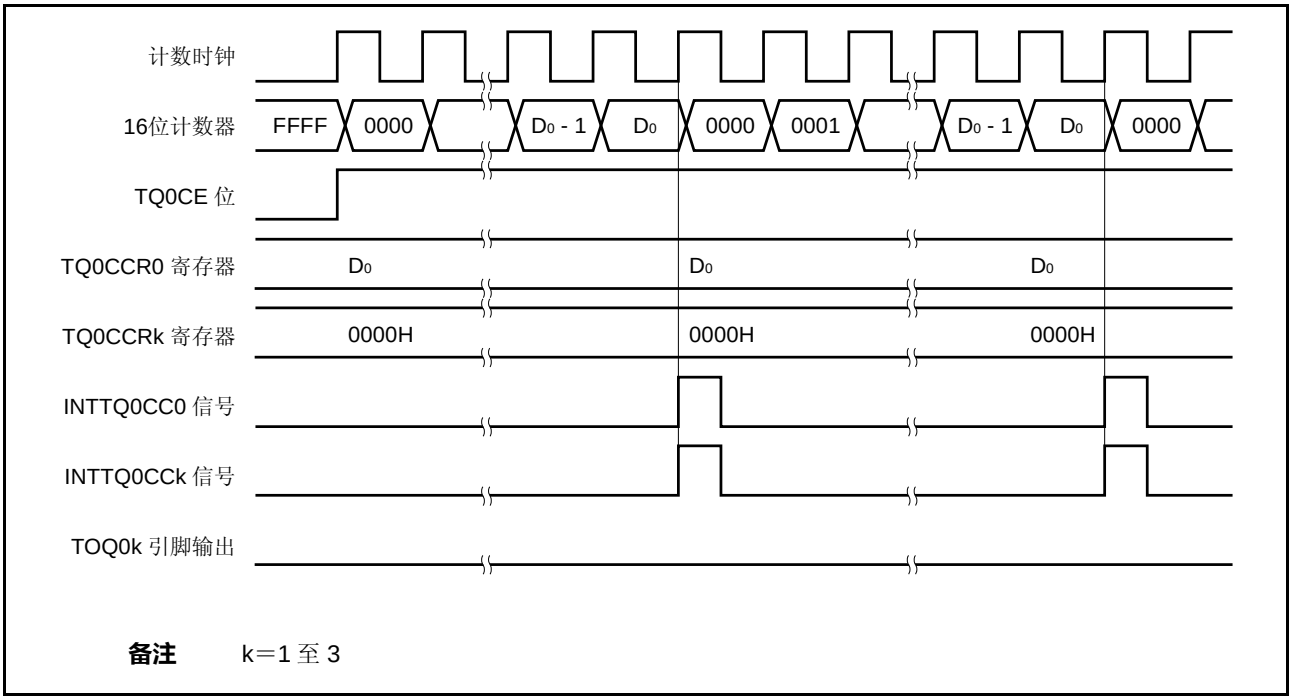
注意事项 在写入 TQ0CCR1 寄存器之后，要重写 TQ0CCR0 至 TQ0CCR3 寄存器，应该在 INTTQ0CC0 信号产生之后进行。否则，CCRm 缓冲寄存器的值可能不确定，因为把数据从 TQ0CCRm 寄存器传送到 CCRm 缓冲寄存器的时序会可能与写入 TQ0CCRm 寄存器的时序发生冲突。

备注 m＝0 至 3

（b）输出 0%或 100%PWM 波形

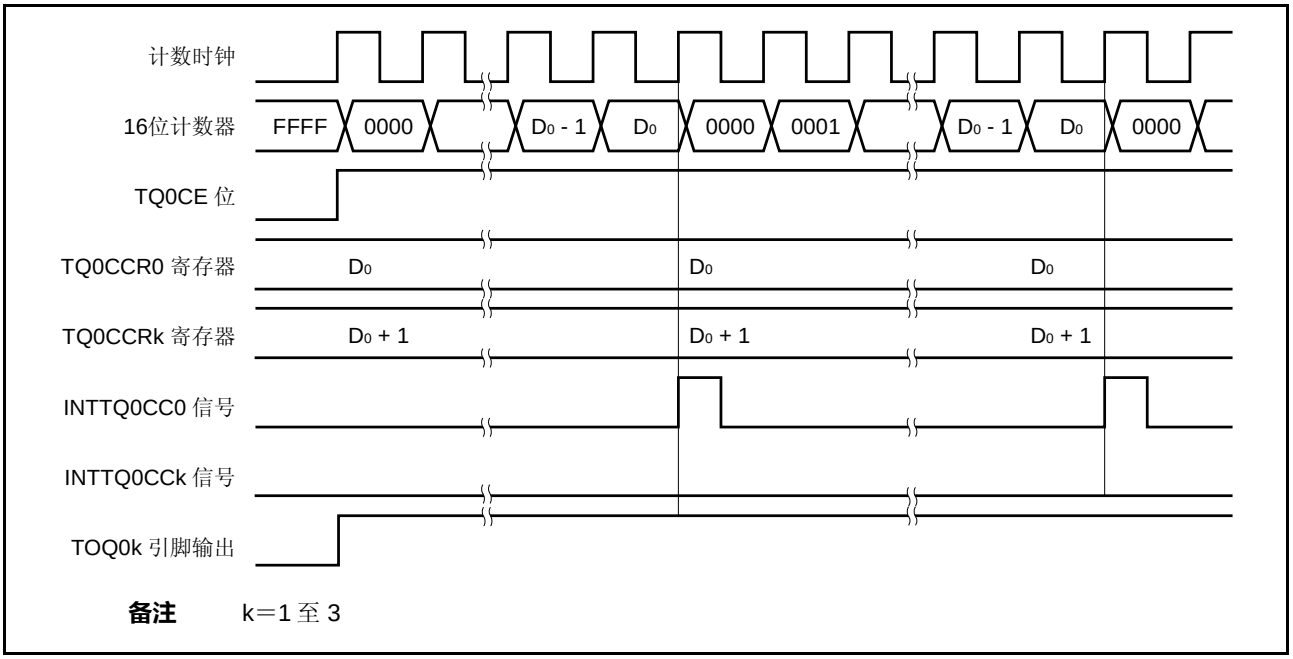
为了输出 0%的波形，需要将 TQ0CCRk 寄存器设置为 0000H。

图 8-49. 输出 0% PWM 波形



为了输出 100%的波形，需要将 TQ0CCRk 寄存器的值设为：（TQ0CCR0 寄存器值 ＋1）。如果 TQ0CCR0 寄存器的设定值为 FFFFH，则不会产生 100%波形输出。

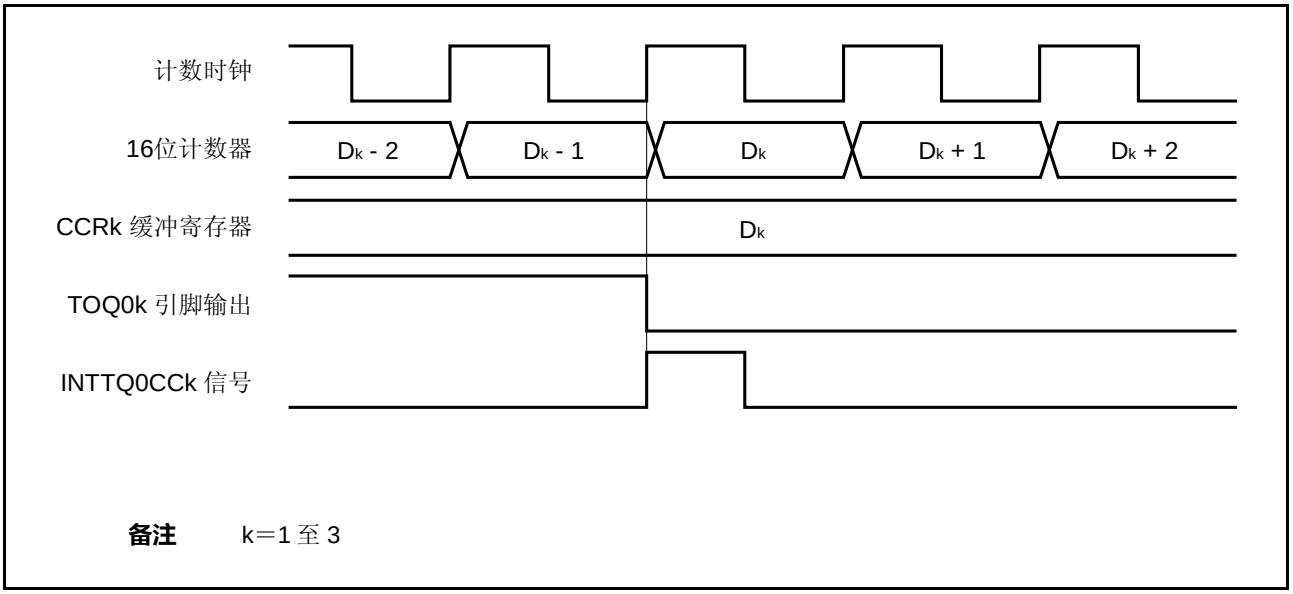
图 8-50. 输出 100% PWM 波形



（c）比较匹配中断请求信号（INTTQ0CCk）的发生时序

在 PWM 输出模式下，当 16 位计数器的计数值与 TQ0CCRk 寄存器的值匹配时，产生 INTTQ0CCk 信号。

图 8-51. 比较匹配中断请求信号（INTTQ0CCk）的发生时序



8.4.6 自由运行定时器模式（TQ0MD2 至TQ0MD0 位 = 101）

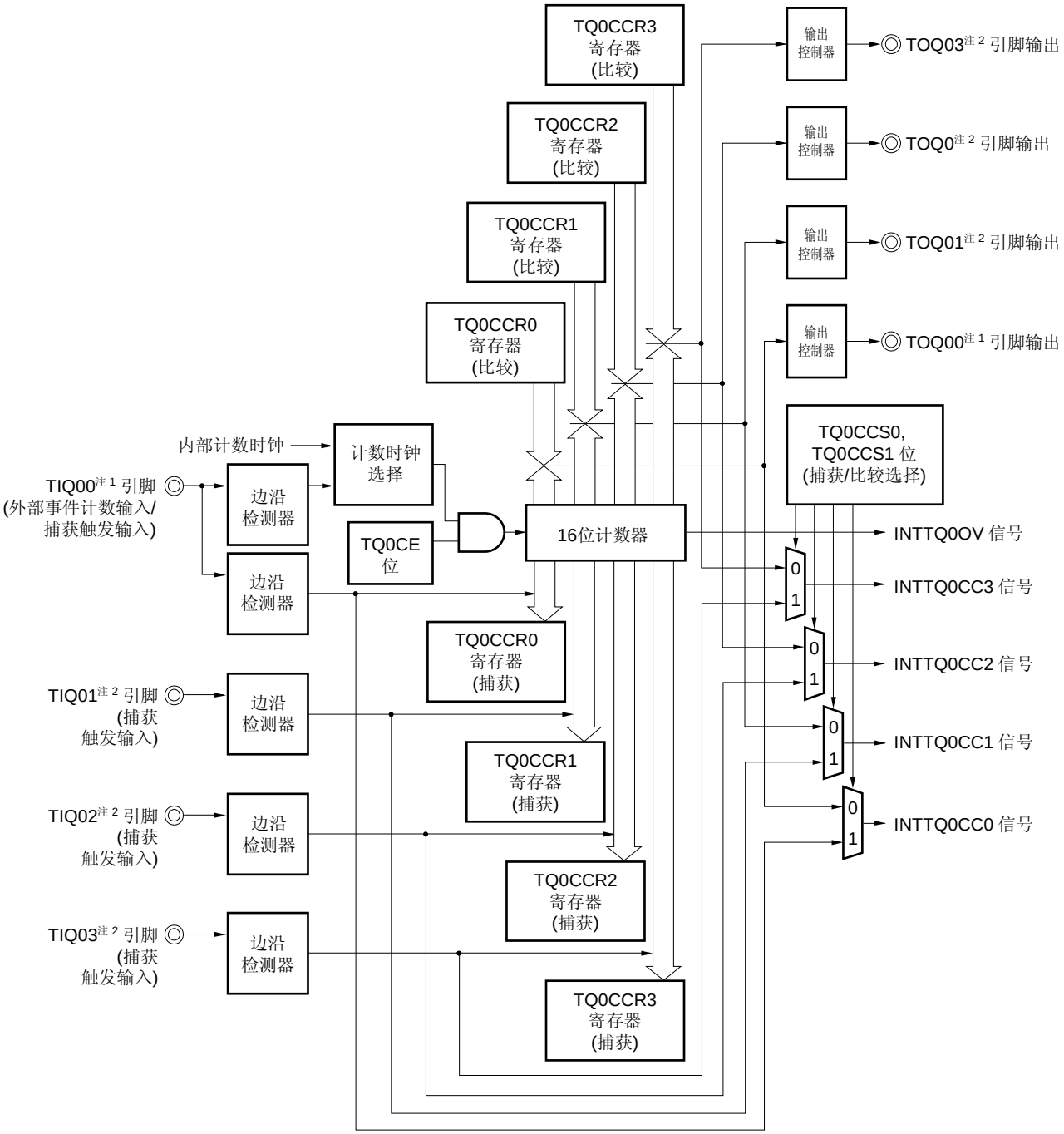
在自由运行定时器模式下，当 TQ0CTL0.TQ0CE 位置为 1 时，TMQ0 开始计数。此时，根据 TQ0OPT0.TQ0CCS0 和 TQ0OPT0.TQ0CCS1 位的设置，TQ0CCRm 寄存器可以用作比较寄存器或捕获寄存器。

备注 **1.** 关于如何设置 TIQ0m 和 TOQ0m 引脚，参见**表 8-2 TMQ0 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。

2. 关于如何使能 INTTQ0CCm 中断信号，参见**第 22 章 中断服务/异常处理。**

3. m = 0 至 3

图 8-52. 自由运行定时器模式下的 TMQ0 配置



注 1. 外部事件计数输入/捕获触发输入引脚(TIQ00)可以用作定时器输出引脚(TOQ00)；但是，这些功能每次只能使用一个。

2. 捕获触发输入引脚(TIQ0k)也可以用作定时器输出引脚(TOQ0k)；但是，这些功能每次只能使用一个 (k = 1 至 3)。

- **比较操作**

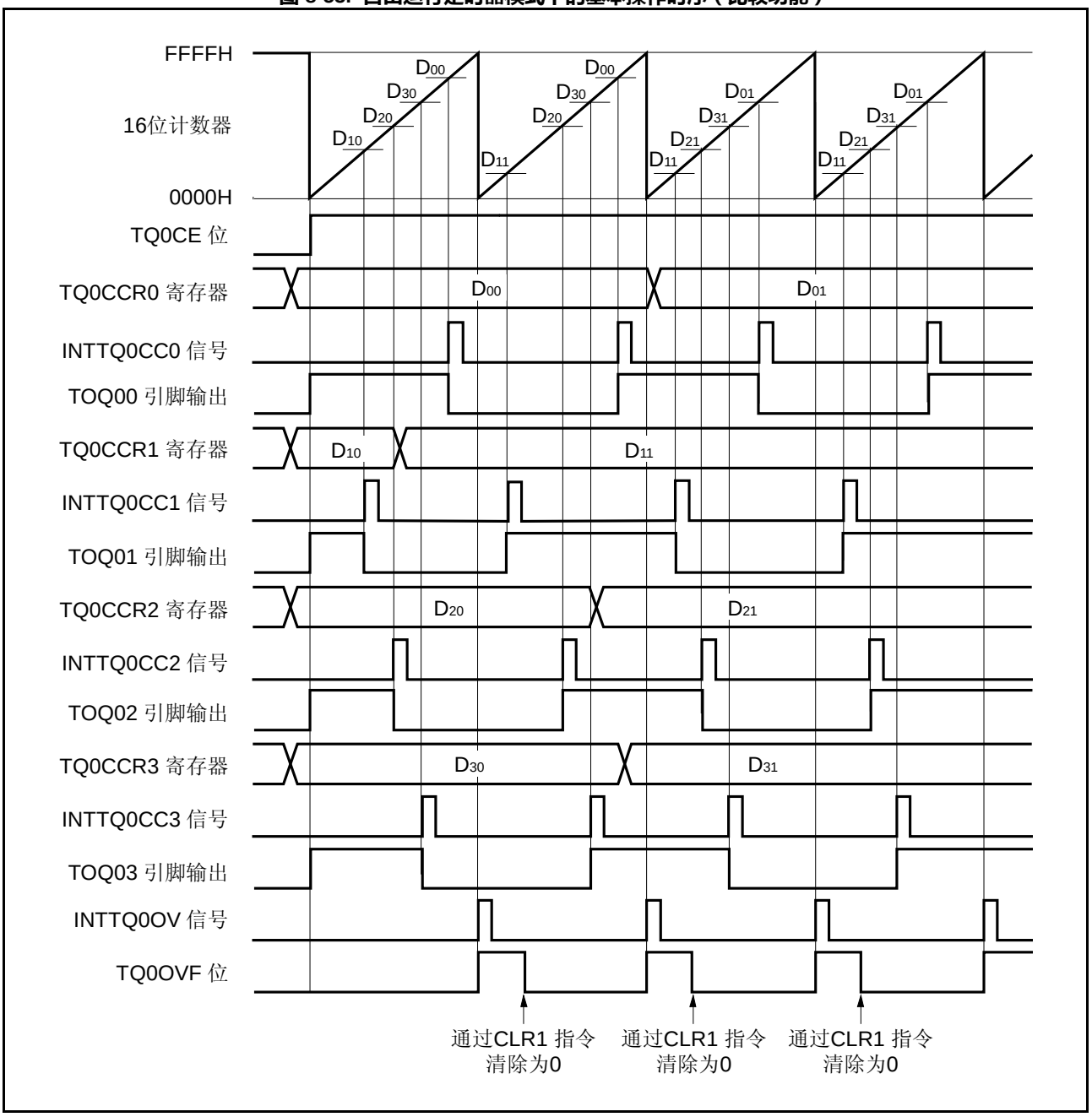
当 TQ0CE 位被置为 1 时，TMQ0 开始计数，TOQ00 至 TOQ03 引脚的输出信号反相。此后当 16 位计数器的计数值与 TQ0CCRm 寄存器的设定值匹配时，产生一个比较匹配中断请求信号（INTTQ0CCm），而且 TOQ0m 引脚的输出信号反相。

16 位计数器继续与计数时钟同步计数。当它计数达到 FFFFH 时，会在下一个计数时钟时产生一个溢出中断请求信号（INTTQ0OV），同时计数器清零为 0000H，然后继续进行计数。此时，溢出标志（TQ0OPT0.TQ0OVF 位）被置为 1，且必须通过执行 CLR1 软件指令把溢出标志位清 0。

当计数器工作时，可以重写 TQ0CCRm 寄存器。如果进行重写，则新写入的值会立刻反映出来，并与计数值进行比较。

备注 m = 0 至 3

图 8-53. 自由运行定时器模式下的基本操作时序（比较功能）



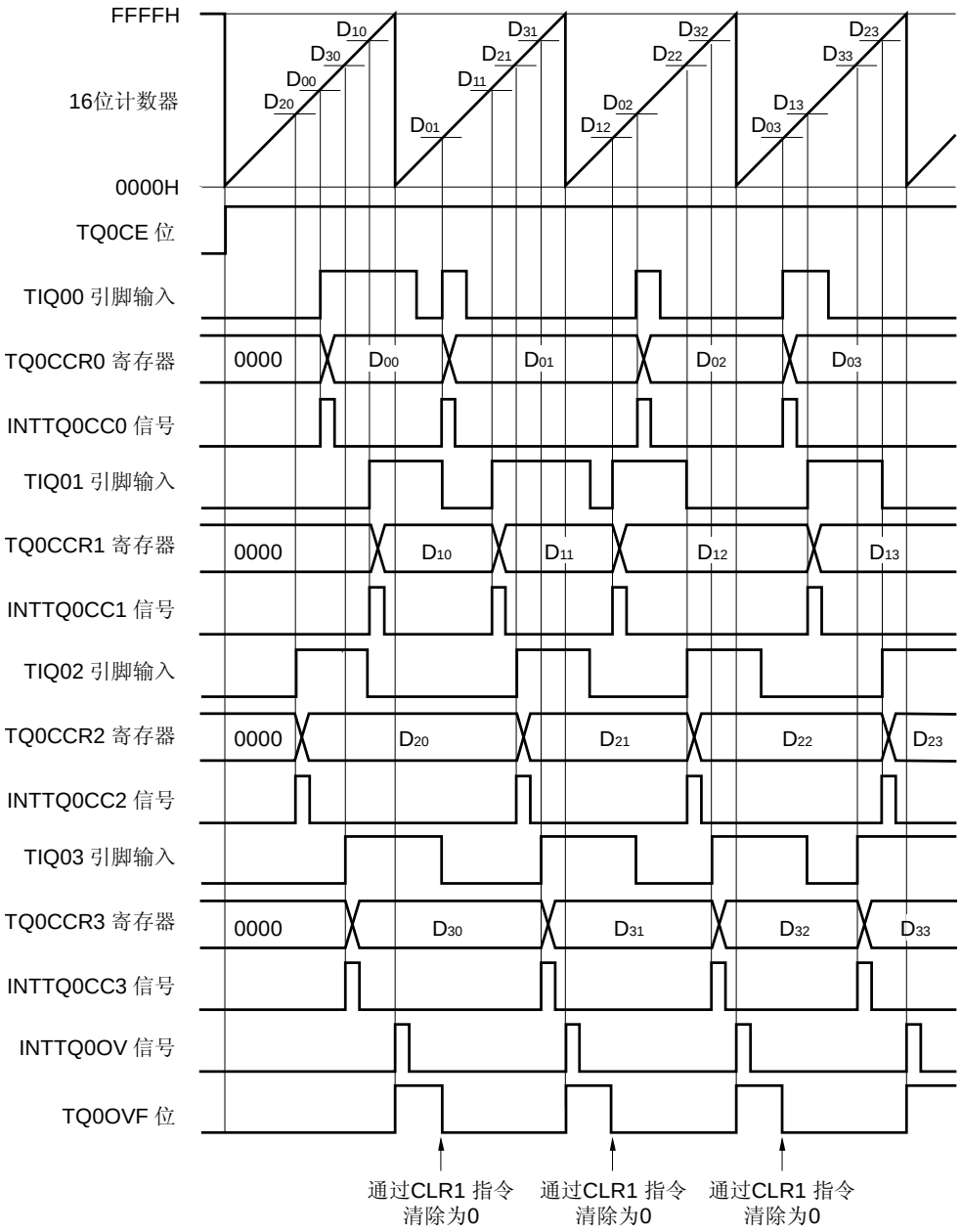
- **捕获操作**

当 TQ0CE 位被置为 1 时，TMQ0 开始计数。当检测到信号的有效边沿输入到 TIQ0m 引脚时，16 位计数器的计数值存储在 TQ0CCRm 寄存器中，并产生一个捕获中断请求信号（INTTQ0CCm）。

16 位计数器继续对计数时钟同步计数。当它计数达到 FFFFH 时，会在下一个计数时钟时产生一个溢出中断请求信号（INTTQ0OV），同时计数器清零为 0000H，然后继续进行计数。此时，溢出标志（TQ0OVF 位）被置为 1，且必须通过执行 CLR1 软件指令把溢出标志位清 0。

备注 m = 0 至 3

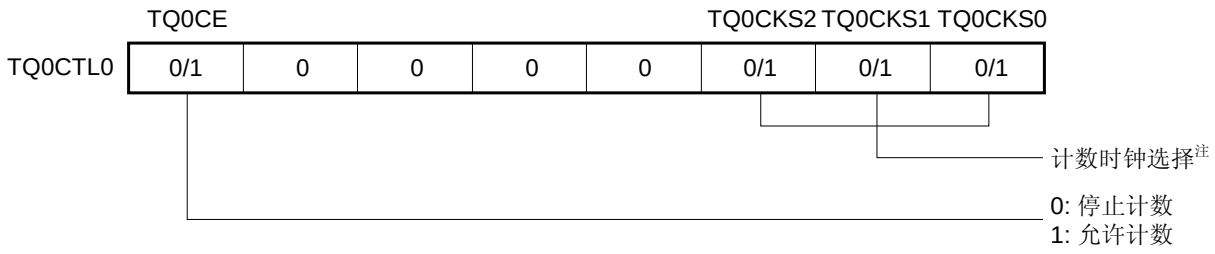
图 8-54. 自由运行定时器模式下的基本操作时序（捕获功能）



备注 有效沿是上升沿。

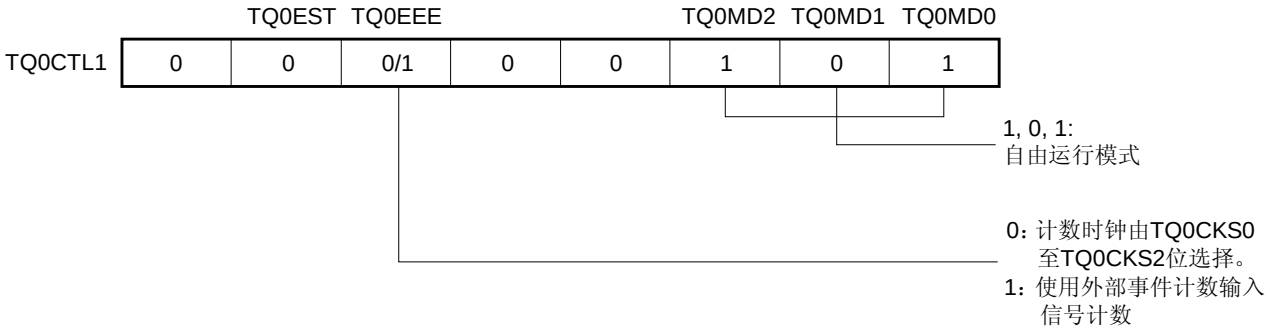
图 8-55. 自由运行定时器模式下的寄存器设置（1/3）

（a）TMQ0 控制寄存器 0（TQ0CTL0）



注 当 TQ0CTL1.TQ0EEE 位＝1 时，这些位的设置无效。

（b）TMQ0 控制寄存器 1（TQ0CTL1）



注 如果 TQ0CTL1.TQ0EEE 位为 1，不能使用 TIQ00 引脚的捕获功能。

图 8-55. 自由运行定时器模式下的寄存器设置（2/3）

（c）TMQ0 输入/输出控制寄存器 0（TQ0IOC0）

	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0IOC0	0/1注	0/1注	0/1注	0/1注	0/1注	0/1注	0/1注	0/1注

- TQ0OE0：0: 禁止 TOQ00 引脚输出 / 1: 使能 TOQ00 引脚输出
- TQ0OL0：当TOQ00引脚被禁止时的输出电平：0：低电平 / 1：高电平
- TQ0OE1：0: 禁止 TOQ01 引脚输出 / 1: 使能 TOQ01 引脚输出
- TQ0OL1：当TOQ01引脚被禁止时的输出电平：0：低电平 / 1：高电平
- TQ0OE2：0: 禁止 TOQ02 引脚输出 / 1: 使能 TOQ02 引脚输出
- TQ0OL2：当TOQ02引脚被禁止时的输出电平：0：低电平 / 1：高电平
- TQ0OE3：0: 禁止 TOQ03 引脚输出 / 1: 使能 TOQ03 引脚输出
- TQ0OL3：当TOQ03引脚被禁止时的输出电平：0：低电平 / 1：高电平

注 当使用 TIQ0m 引脚时，不能使用 TOQ0m 引脚。

备注 m = 0 至 3

（d）TMQ0 输入/输出控制寄存器 1（TQ0IOC1）

	TQ0IS7	TQ0IS6	TQ0IS5	TQ0IS4	TQ0IS3	TQ0IS2	TQ0IS1	TQ0IS0
TQ0IOC1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

- TQ0IS0：选择TIQ00引脚输入的有效沿
- TQ0IS2：选择TIQ01引脚输入的有效沿
- TQ0IS4：选择TIQ02引脚输入的有效沿
- TQ0IS6：选择TIQ03引脚输入的有效沿

图 8-55. 自由运行定时器模式下的寄存器设置（3/3）

（e）TMQ0 输入/输出控制寄存器 2（TQ0IOC2）

					TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
TQ0IOC2	0	0	0	0	0/1	0/1	0	0

选择外部事件计数输入的有效沿

（f）TMQ0 选项寄存器 0（TQ0OPT0）

	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0				TQ0OVF
TQ0OPT0	0/1	0/1	0/1	0/1	0	0	0	0/1

溢出标志

指定 TQ0CCR0寄存器的功能为捕获寄存器或比较寄存器

指定 TQ0CCR1寄存器的功能为捕获寄存器或比较寄存器

指定 TQ0CCR2寄存器的功能为捕获寄存器或比较寄存器

指定 TQ0CCR3寄存器的功能为捕获寄存器或比较寄存器

（g）TMQ0 计数器读取缓冲寄存器（TQ0CNT）

通过读取该寄存器，可以读出 16 位计数器的值。

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

根据 TQ0OPT0.TQ0CCSm 位的设置，这些寄存器可以用作捕获寄存器或比较寄存器。

当用作捕获寄存器时，在检测到信号的有效边沿输入到 TIQ0m 引脚时，存储 16 位计数器的当前计数值，并产生 INTTQ0CCm 中断信号。

当用作比较寄存器时，且 TQ0CCRm 寄存器的值被设置为 D_m 时，则当计数器达到（D_m ＋ 1）时会产生 INTTQ0CCm 中断信号，同时，TOQ0m 引脚的输出信号被反转。

备注 m＝0 至 3

（1）自由运行定时器模式下的操作

在自由运行定时器模式下，会发生下列两种操作：

- 捕获操作
- 比较操作

（a）当捕获/比较寄存器用作比较寄存器时

图 8-56. 自由运行定时器模式的时序和操作处理（比较功能）（1/2）

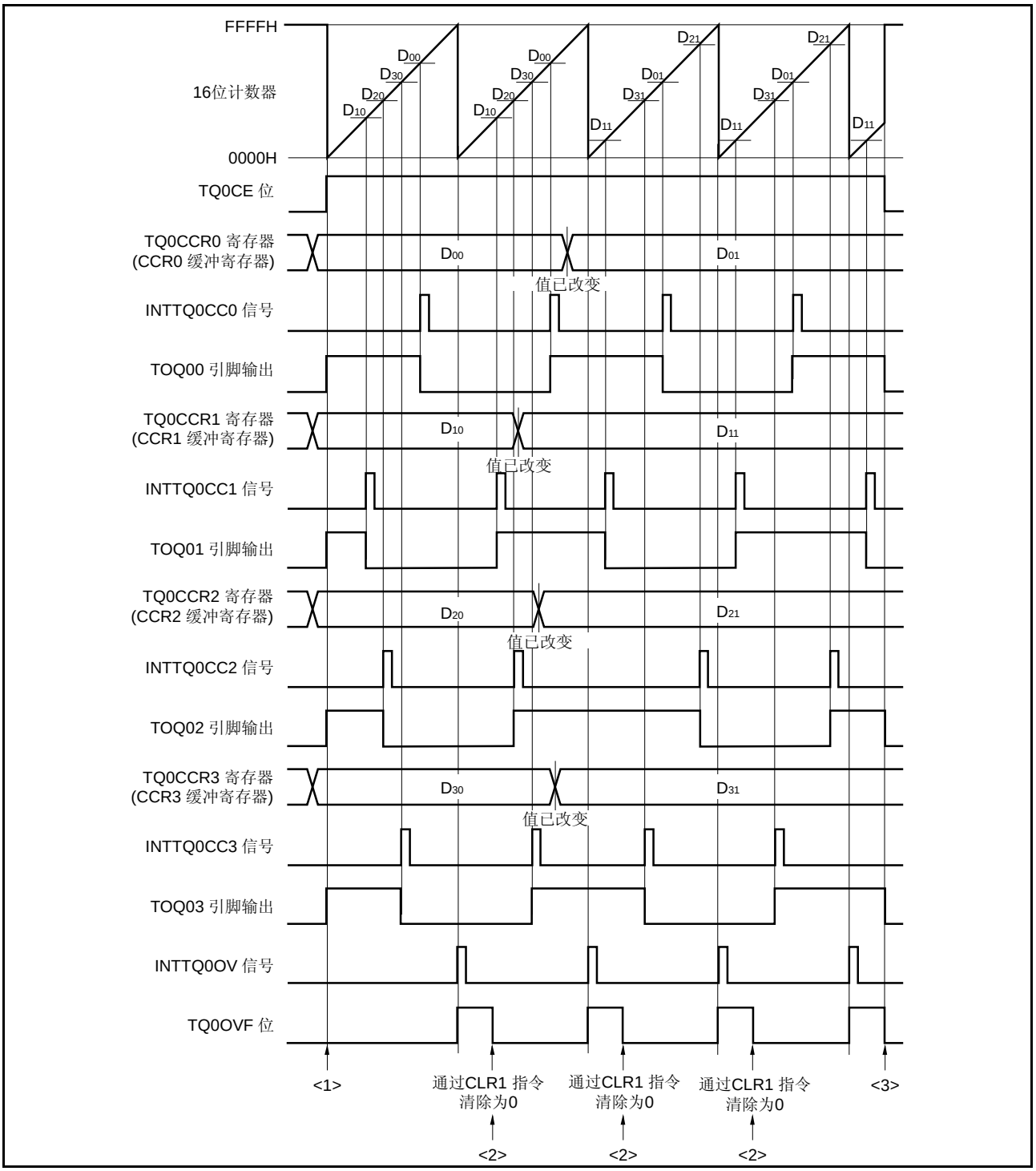
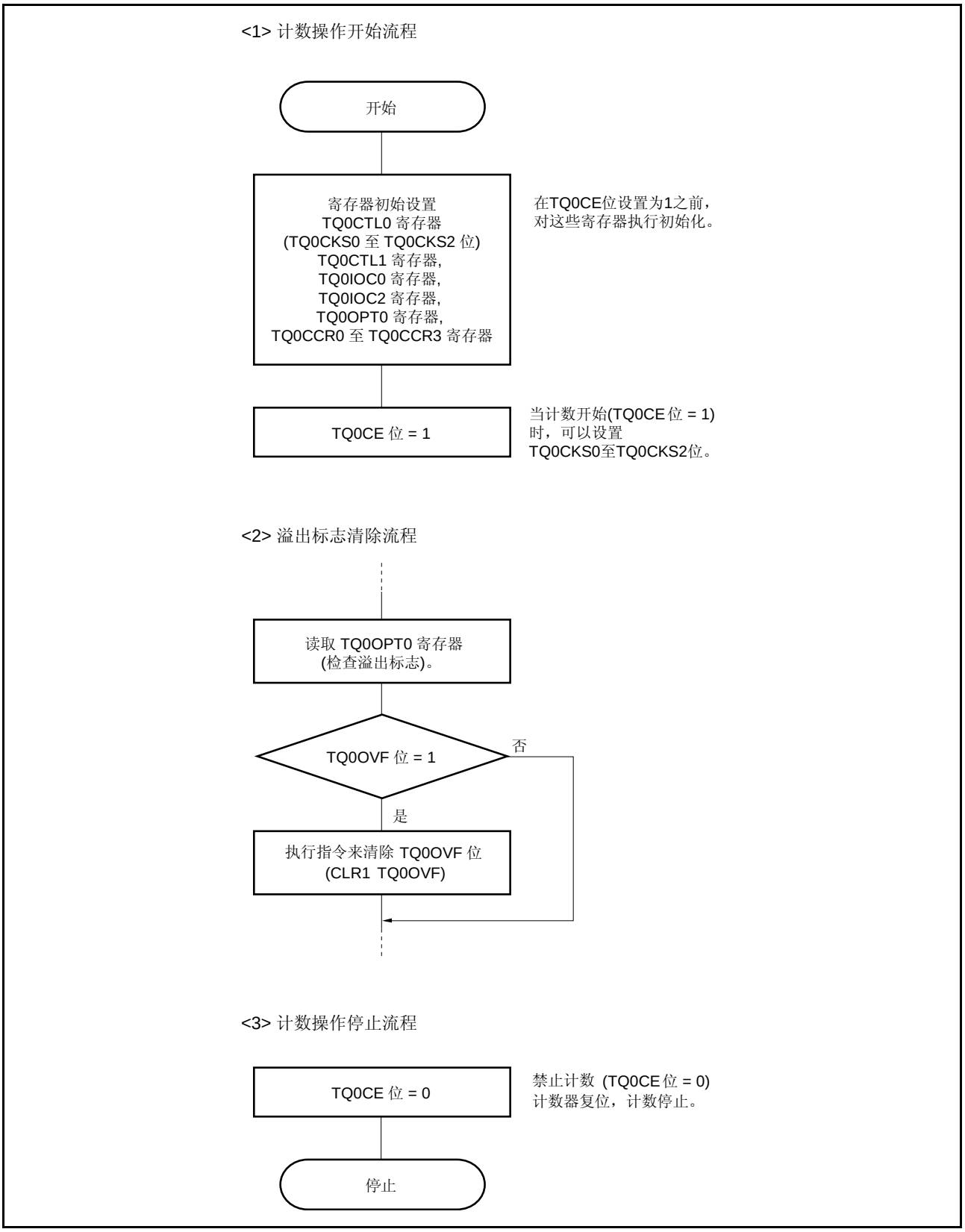


图 8-56. 自由运行定时器模式的时序和操作处理（比较功能）（2/2）



（b）当捕获/比较寄存器用作捕获寄存器时

图 8-57. 自由运行定时器模式的时序和操作处理（捕获功能）（1/2）

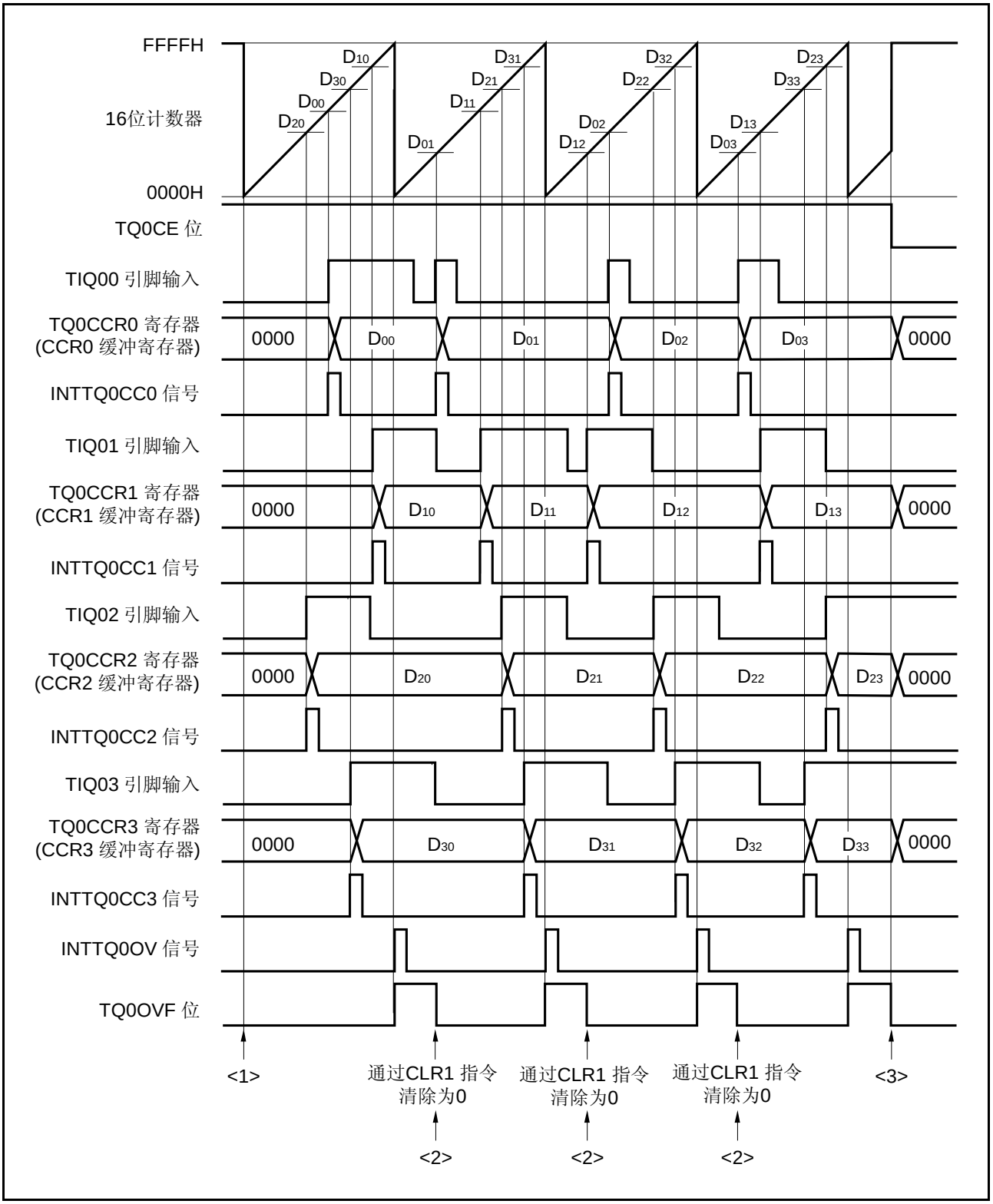
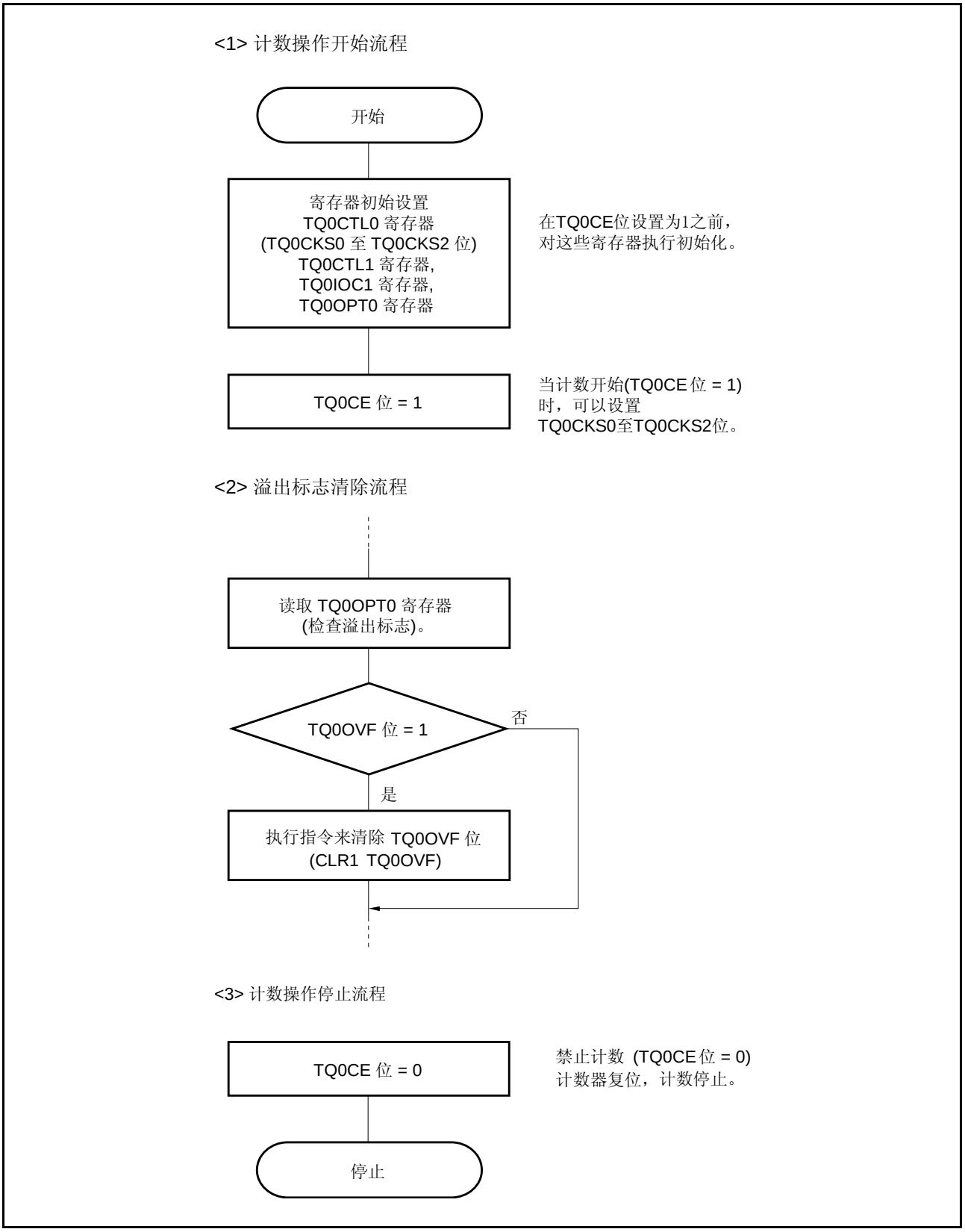


图 8-57. 自由运行定时器模式时的时序和操作处理（捕获功能）（2/2）

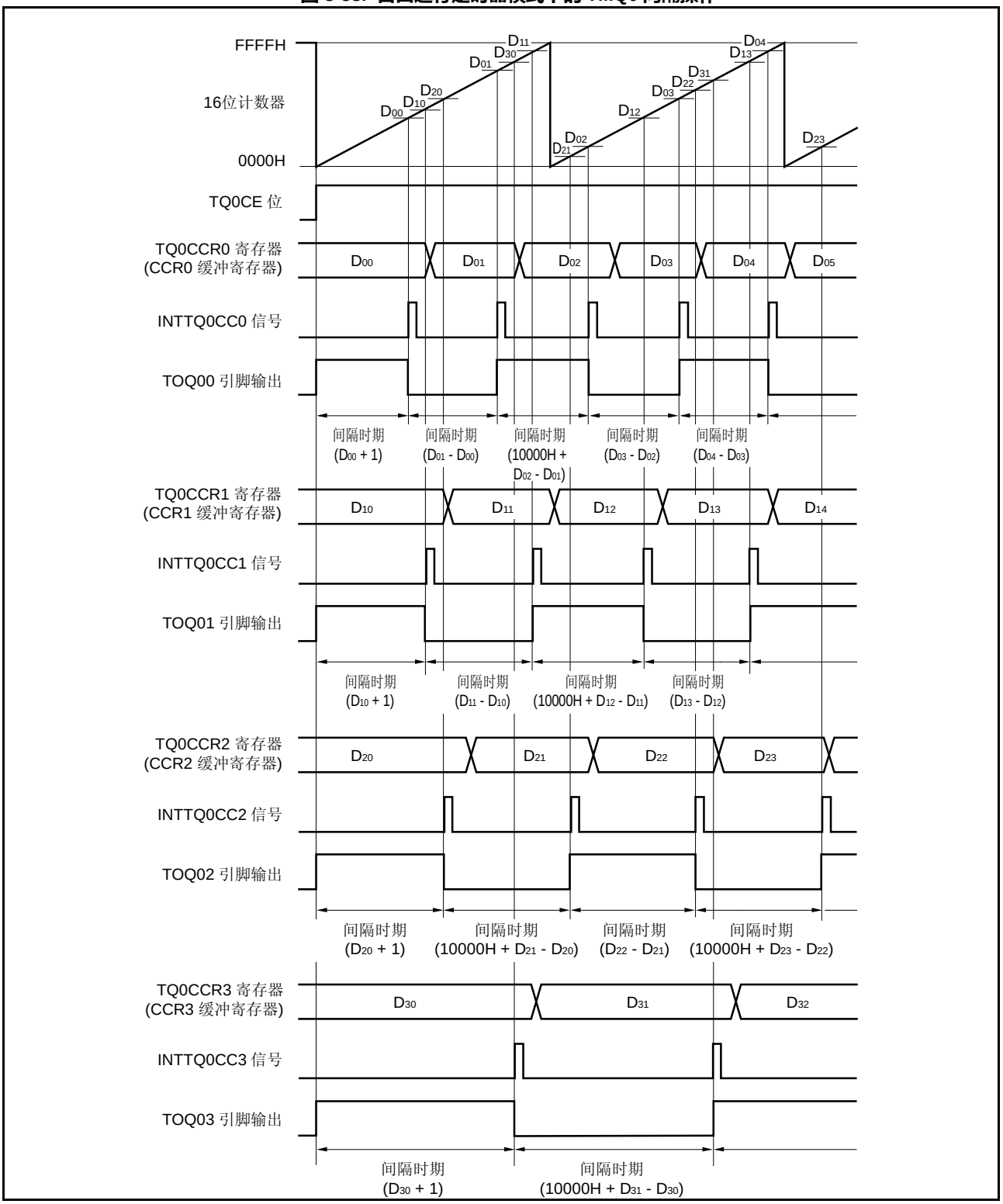


（2）使用自由运行定时器模式

（a）TQ0CCRm 寄存器用作比较寄存器的间隔定时操作

当 TMQ0 用作间隔定时器，且 TQ0CCRm 寄存器用作比较寄存器时，在每次检测到 INTTQ0CCm 信号后，都需要通过软件设置一个比较值以便产生下一个中断请求信号。

图 8-58. 自由运行定时器模式下的 TMQ0 间隔操作



在自由运行定时器模式下进行时间间隔操作时，一个通道可以设置四个时间间隔。
要进行时间间隔操作，当检测到 INTTQ0CCm 信号后，必须在中断服务程序中重置相应的 TQ0CCRm 寄存器的值。
当“D_m”为间隔时间时，这种情况下待重置的值可由下式计算：

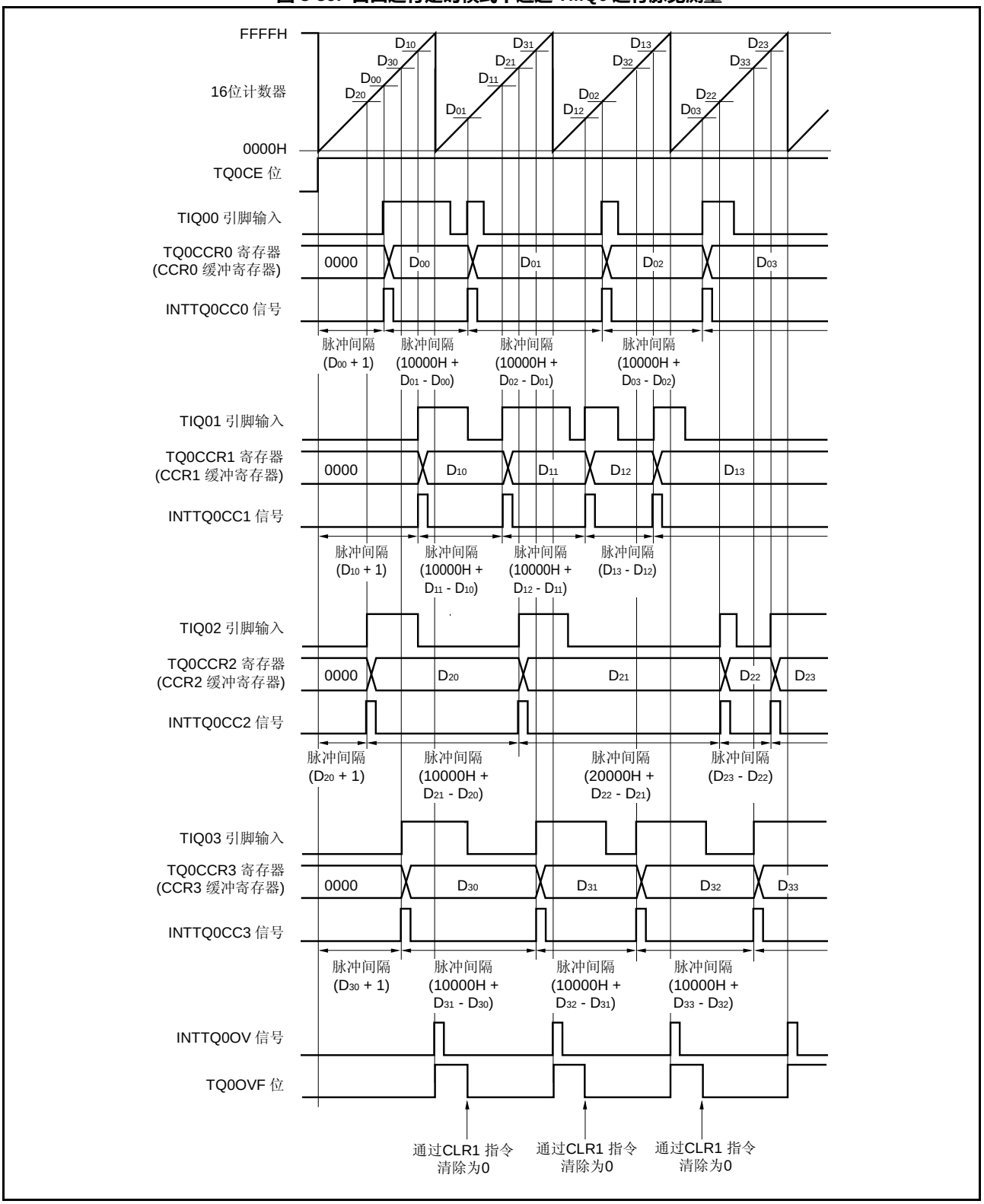
比较寄存器默认值： $D_m - 1$
比较寄存器的第二次及其以后的设置值：上次设置值 ＋D_m
（若计算结果大于 FFFFH，则从结果中减去 10000H，并将此值设置给该寄存器。）

备注 m＝0 至 3

（b）TQ0CCRm 寄存器用作捕获寄存器进行脉宽测量

在 TQ0CCRm 寄存器用作捕获寄存器进行脉宽测量时，每次检测到 INTTQ0CCm 信号时，必须要读取捕获寄存器并通过软件计算间隔时间。

图 8-59. 自由运行定时模式下通过 TMQ0 进行脉宽测量



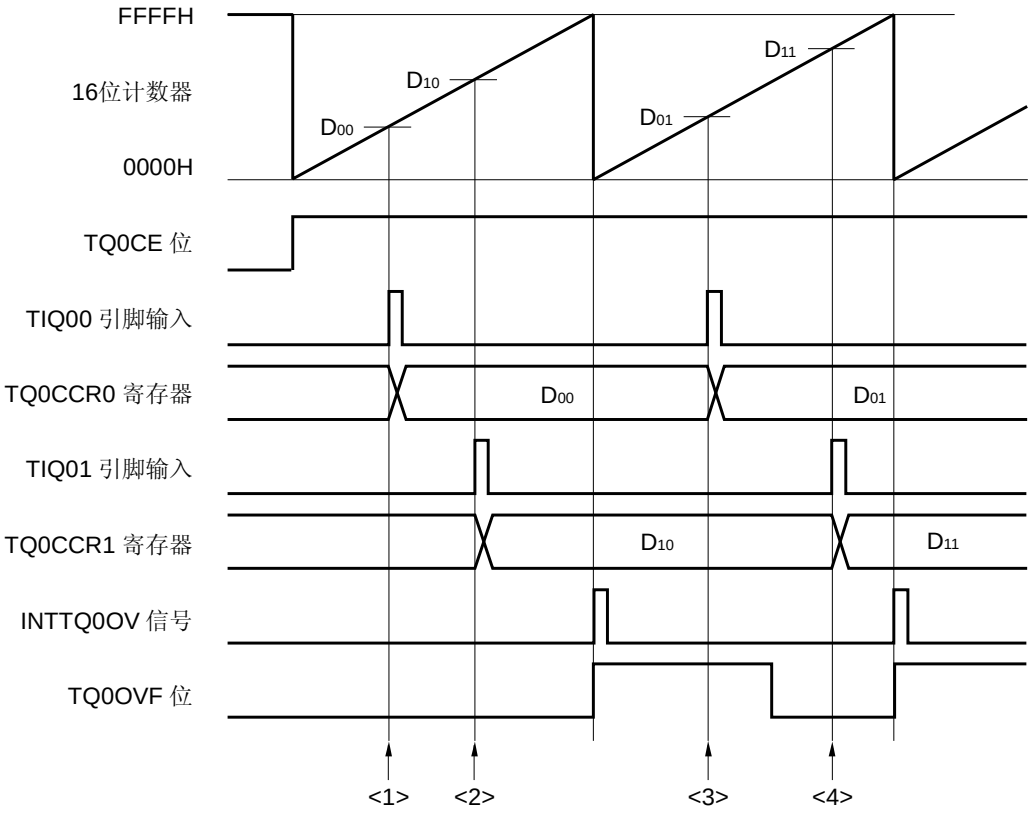
在自由运行定时器模式下执行脉宽测量时，可以用一个通道测量四个脉冲宽度。
要测量脉冲宽度，可以与 INTTQ0CCm 信号同步读取 TQ0CCRm 寄存器的值，并计算读取值与先前读取值之间的差，得到脉冲宽度。

备注 m＝0 至 3

（c）当使用两个或更多捕获寄存器时对溢出的处理

当使用两个或更多捕获寄存器时必须要格外小心。首先是一个不正确处理的示例，如下所示。

图 8-60. 使用两个捕获寄存器时的不正确处理示例

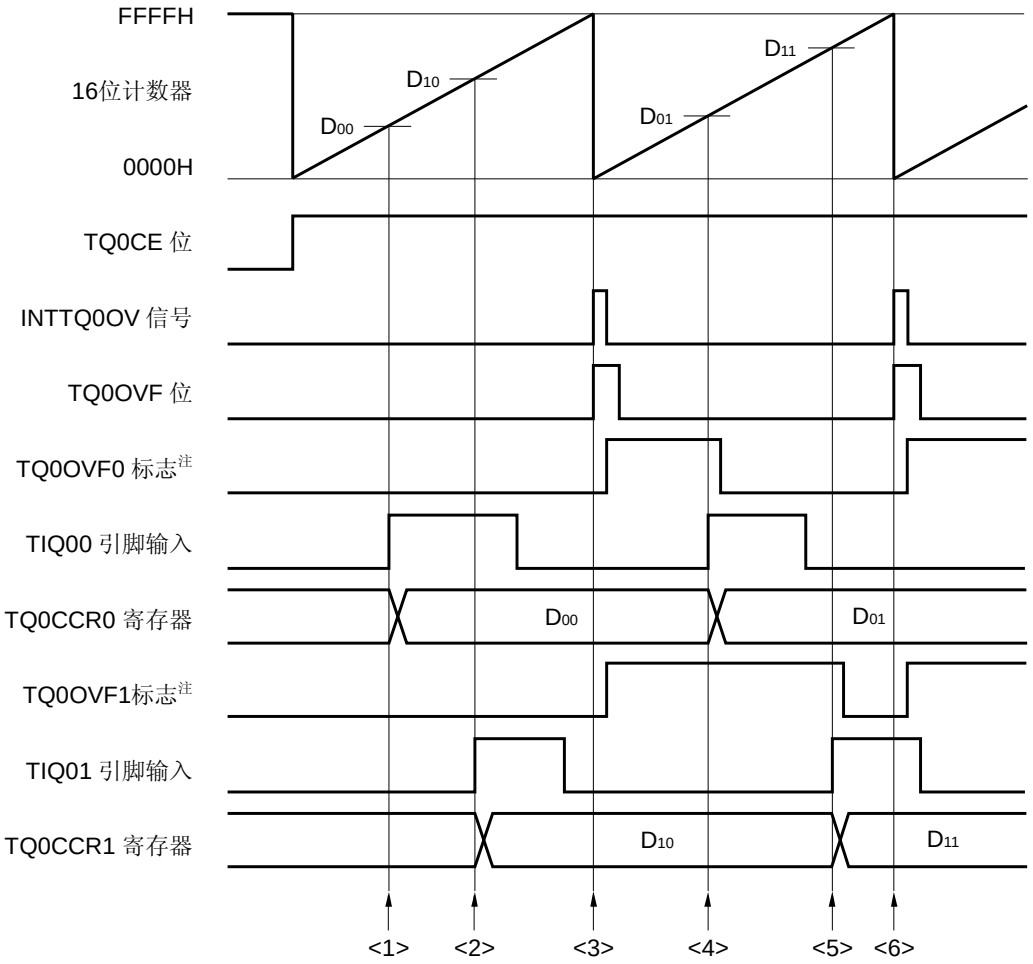


在自由运行定时器模式下测量两个脉冲宽度时，可能出现下列问题。

<1> 读取 TQ0CCR0 寄存器（设置 TIQ00 引脚输入的默认值）。
<2> 读取 TQ0CCR1 寄存器（设置 TIQ01 引脚输入的默认值）。
<3> 读取 TQ0CCR0 寄存器。
读取 TQ0OVF 位，如果 TQ0OVF 位为 1，则其被清除为 0。
因为 TQ0OVF 位为 1，所以脉冲宽度可以通过（10000H＋D_{01} – D_{00}）计算。
<4> 读取 TQ0CCR1 寄存器。
读取 TQ0OVF 位。因为此标志在<3>中被清零，所以，读取到 0。
因为 TQ0OVF 位为 0，所以脉冲宽度可以用（D_{11} – D_{10}）计算。（不正确）

在使用两个或更多捕获寄存器时，如果 TQ0OVF 位由一个捕获寄存器清除为 0，则另一个捕获寄存器可能得不到正确的脉冲宽度。
该问题可以使用软件解决。关于如何使用软件的示例如下。

图 8-61. 使用两个捕获寄存器时通过溢出中断解决问题的示例

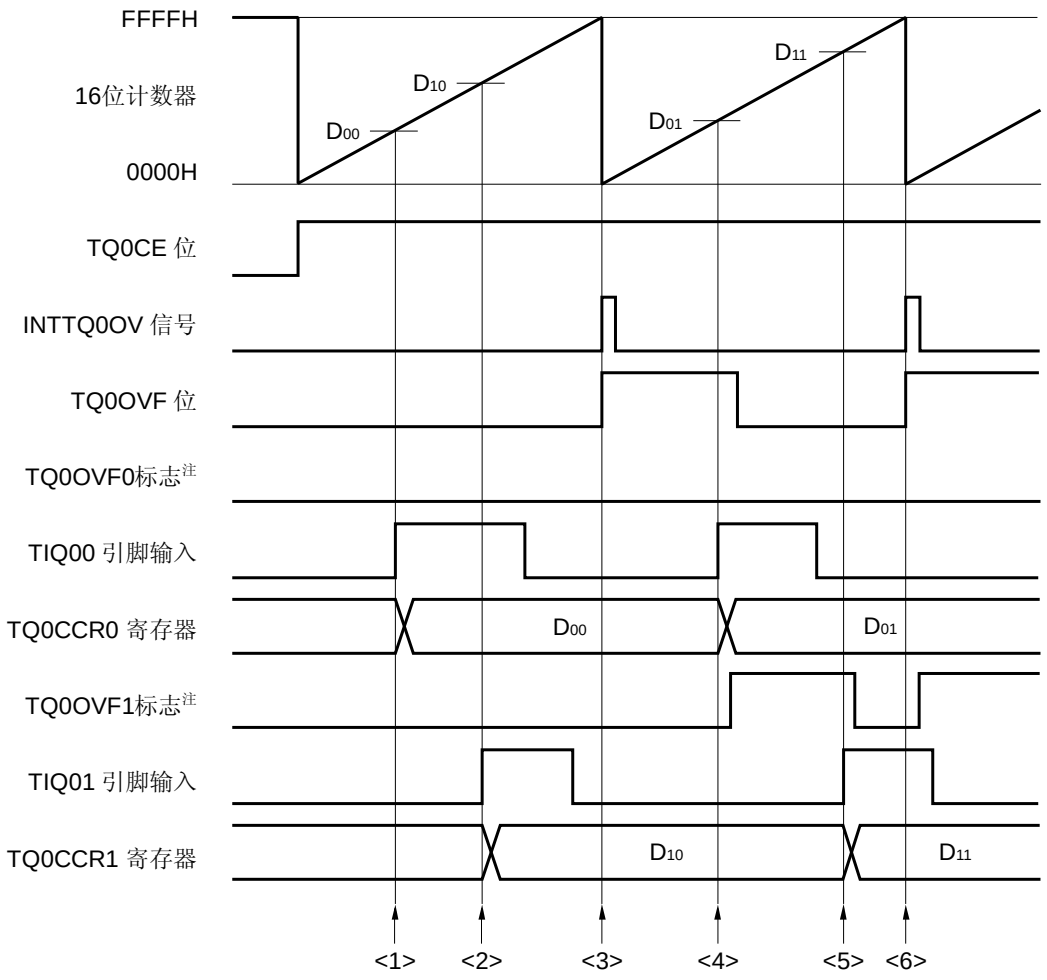


注 由软件对内部 RAM 中的 TQ0OVF0 和 TQ0OVF1 标志进行设置。

<1> 读取 TQ0CCR0 寄存器（设置 TIQ00 引脚输入的默认值）。
<2> 读取 TQ0CCR1 寄存器（设置 TIQ01 引脚输入的默认值）。
<3> 发生溢出，在溢出中断服务程序中将 TQ0OVF0 和 TQ0OVF1 标志设置为 1，并将溢出标志清除为 0。
<4> 读取 TQ0CCR0 寄存器。
读取 TQ0OVF0 标志。如果 TQ0OVF0 标志为 1，则将其清除为 0。
因为 TQ0OVF0 标志为 1，所以脉冲宽度可以通过（10000H＋D_{01} – D_{00}）计算。
<5> 读取 TQ0CCR1 寄存器。
读取 TQ0OVF1 标志。如果 TQ0OVF1 标志为 1，则将其清除为 0（TQ0OVF0 标志在<4>中已清零，而 TQ0OVF1 标志仍然保持为 1）。
因为 TQ0OVF1 标志为 1，所以脉冲宽度可以通过（10000H＋D_{11} – D_{10}）计算。（正确）
<6> 与步骤 <3> 相同。

图 8-62. 使用两个捕获寄存器时不通过溢出中断解决问题的示例

使用两个捕获寄存器时的示例（不使用溢出中断）



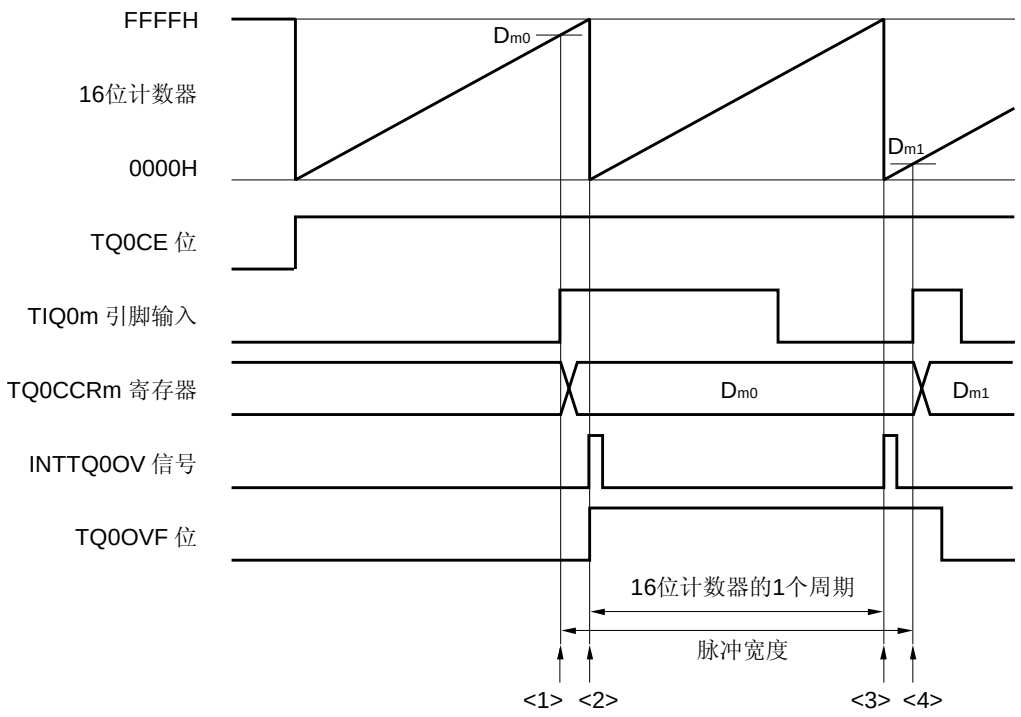
注 由软件对内部 RAM 中的 TQ0OVF0 和 TQ0OVF1 标志进行设置。

<1> 读取 TQ0CCR0 寄存器（设置 TIQ00 引脚输入的默认值）。
<2> 读取 TQ0CCR1 寄存器（设置 TIQ01 引脚输入的默认值）。
<3> 发生溢出，软件不作任何处理。
<4> 读取 TQ0CCR0 寄存器。
读取 TQ0OVF 位。如果 TQ0OVF 位为 1，则只把 TQ0OVF1 标志置位（1），且 TQ0OVF 位被清除为 0。
因为 TQ0OVF 位为 1，所以脉冲宽度可以通过（10000H＋D_{01} – D_{00}）计算。
<5> 读取 TQ0CCR1 寄存器。
读取 TQ0OVF 位。因为 TQ0OVF 位在<4>中被清零，所以读取得到的是 0。
读取 TQ0OVF1 标志。如果 TQ0OVF1 标志为 1，则将其清除为 0。
因为 TQ0OVF1 标志为 1，所以脉冲宽度可以通过（10000H＋D_{11} – D_{10}）计算。（正确）
<6> 与步骤 <3> 相同。

（d）当捕获触发间隔较长时的溢出处理

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

图 8-63. 捕获触发间隔较长时的不正确处理示例(使用 TIQ0m)



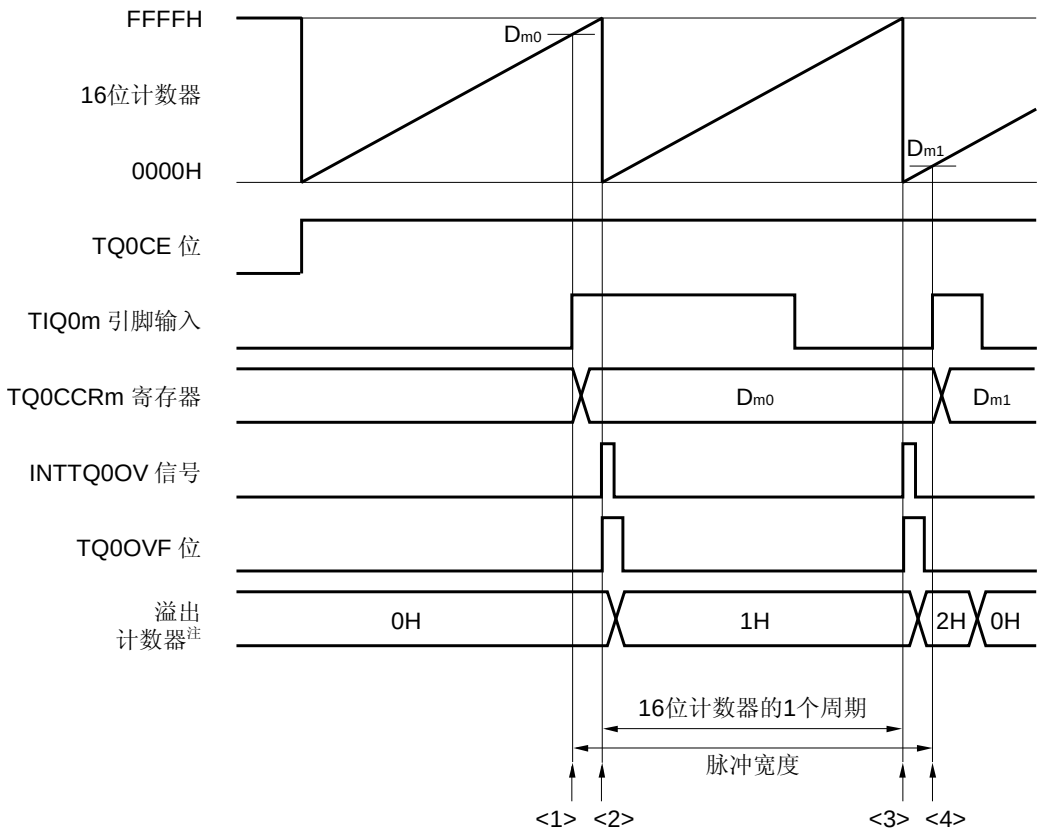
在自由运行定时器模式下测量长脉冲的宽度时，如果脉冲宽度较大，可能出现下列问题。

<1> 读取 TQ0CCRm 寄存器（设置 TIQ0m 引脚输入的默认值）。
<2> 发生溢出，软件不作任何处理。
<3> 再次发生溢出。软件不作任何处理。
<4> 读取 TQ0CCRm 寄存器。
读取 TQ0OVF 位。如果 TQ0OVF 为 1，将其清除为 0。
因为 TQ0OVF 为 1，所以脉冲宽度可以通过（10000H＋D_{m1} – D_{m0}）计算。（不正确）
事实上，脉冲宽度应为（20000H＋D_{m1} – D_{m0}），因为实际已经发生两次溢出。

备注 m = 0 至 3

如果在捕获触发间隔期间，发生两次或更多次的溢出，则无法获得正确的脉冲宽度。
如果捕获触发间隔时间较长时，应该减慢计数时钟以延长 16 位计数器的循环周期，或者使用软件来解决该问题。关于如何使用软件解决该问题，示例如下。

图 8-64. 捕获触发间隔较长时通过软件处理解决问题的示例(使用 TIQ0m)



注 内部 RAM 中的溢出计数器由软件任意设置。

<1> 读取 TQ0CCRm 寄存器（设置 TIQ0m 引脚输入的默认值）。
<2> 发生溢出，在溢出中断服务程序中增加溢出计数器的值，并将 TQ0OVF 位清除为 0。
<3> 再次发生溢出。在溢出中断服务程序中递增溢出计数器的值，并将溢出标志 TQ0OVF 位清为 0。
<4> 读取 TQ0CCRm 寄存器。
读取溢出计数器。
当溢出计数器为“N”时，脉冲宽度可用式（$N \times 10000H + D_{m1} - D_{m0}$）计算。
在本例中，因为溢出实际发生了两次，脉冲宽度计算为（$20000H + D_{m1} - D_{m0}$）。
溢出计数器被清除为 0H。

备注 m = 0 至 3

（e）清除溢出标志（TQ0OVF）

通过读取 TQ0OVF 位可以将溢出标志（TQ0OVF）清除为 0。如果它的值为 1，可以通过 CLR1 指令将 TQ0OVF 位清为 0，或向 TQ0OPT0 寄存器写入 8 位数据（第 0 位是 0）。

8.4.7 脉冲宽度测量模式（TQ0MD2 至TQ0MD0 位 = 110）

在脉冲宽度测量模式下，当 TQ0CTL0.TQ0CE 位被置为 1 时，TMQ0 开始递增计数。每次检测到有效边沿输入到 TIQ0m 引脚时，16 位计数器的计数值就存储在 TQ0CCRm 寄存器中，且 16 位计数器清零为 0000H。

在出现捕获中断请求信号（INTTQ0CCm）之后，可以通过读取 TQ0CCRm 寄存器来测量有效边沿的间隔。

选择 TIQ00 至 TIQ03 引脚之一作为捕获触发信号输入引脚时，使用 TQ0IOC1 寄存器将未使用的引脚指定为“无边沿检测”。

备注 **1.** 关于如何设置 TIQ0m 引脚，参见**表 8-2 TMQ0 使用的引脚** 和**表 4-15 端口引脚用作复用功能时的设置**。
2. 关于如何使能 INTTQ0CCm 中断信号，参见**第 22 章 中断服务/异常处理。**
3. m = 0 至 3
k = 1 至 3

图 8-65. 脉宽测量模式下的 TMQ0 配置

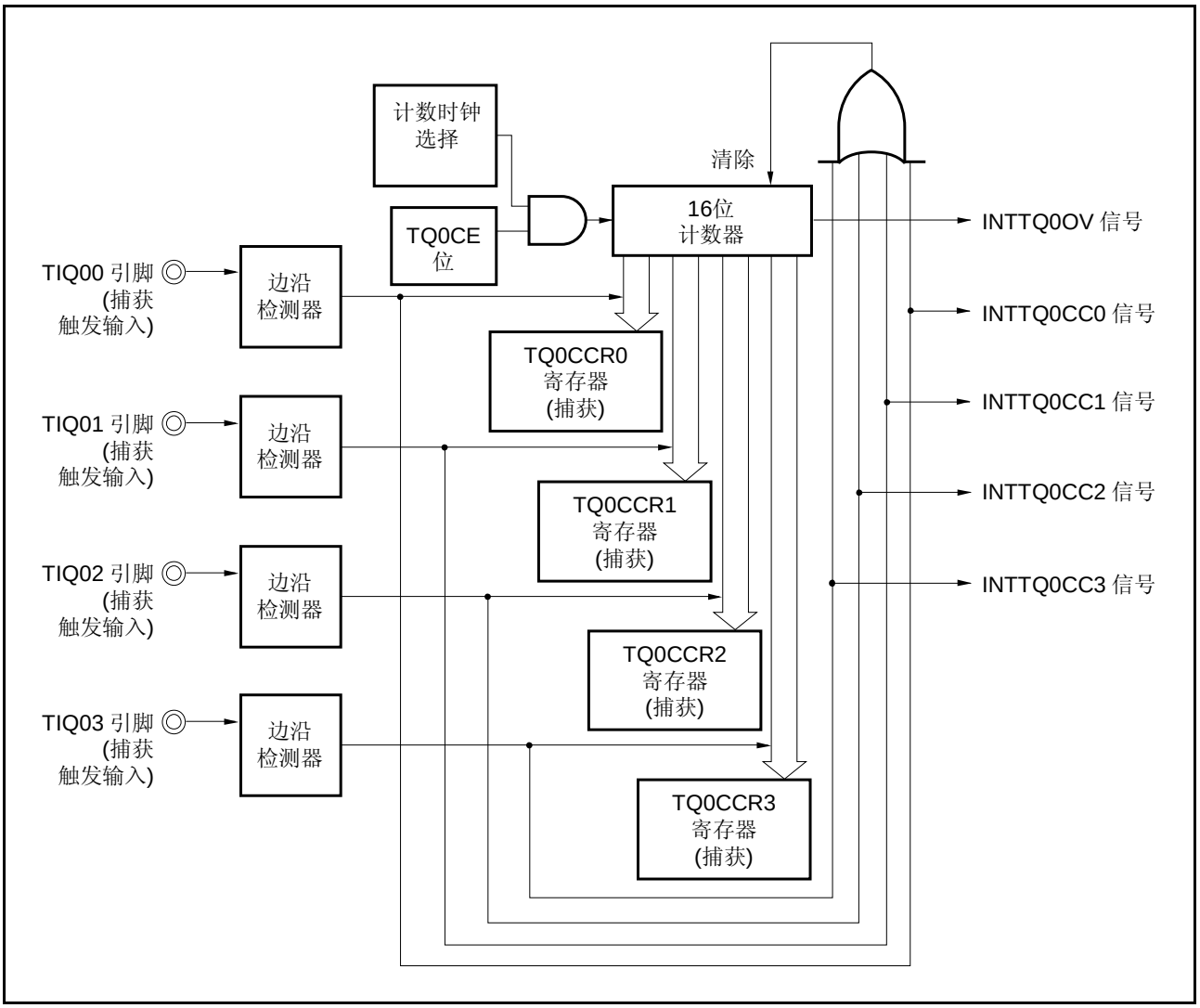
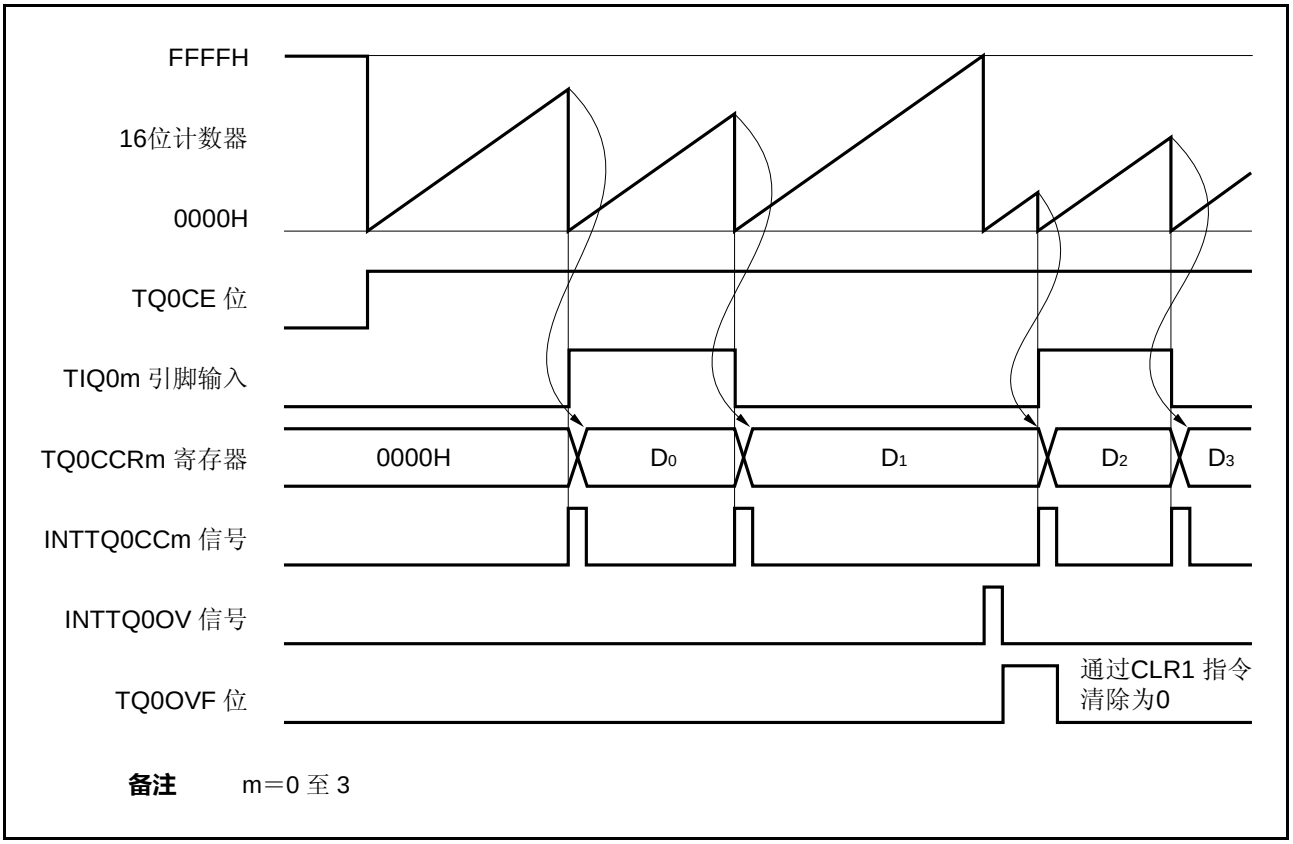


图 8-66. 脉宽测量模式的基本操作时序



当 TQ0CE 位被置为 1 时，16 位计数器开始递增计数。然后当检测到信号的有效边沿输入到 TIQ0m 引脚时，16 位计数器的计数值被存储到 TQ0CCRm 寄存器中，16 位计数器清零为 0000H，并产生一个捕获中断请求信号（INTTQ0CCm）。

脉冲宽度的计算如下。

脉冲宽度＝捕获值 × 计数时钟周期

当 16 位计数器计数到 FFFFH 时，即使没有有效边沿输入 TIQ0m 引脚，在下一个计数时钟时，会产生溢出中断请求信号（INTTQ0OV），且计数器清零为 0000H，然后计数器继续递增计数。此时，溢出标志（TQ0OPT0.TQ0OVF 位）仍被置为 1。使用软件执行 CLR1 指令将溢出标志清为 0。

若溢出标志被设置为 1，则脉冲宽度的计算如下。

脉冲宽度＝（10000H × TQ0OVF 位置位（1）的次数 ＋ 捕获值）× 计数时钟周期

备注 m＝0 至 3

图 8-67. 脉冲宽度测量模式时的寄存器设置（1/2）

（a）TMQ0 控制寄存器 0（TQ0CTL0）

	TQ0CE					TQ0CKS2	TQ0CKS1	TQ0CKS0
TQ0CTL0	0/1	0	0	0	0	0/1	0/1	0/1

选择计数时钟

0: 停止计数
1: 允许计数

（b）TMQ0 控制寄存器 1（TQ0CTL1）

		TQ0EST	TQ0EEE			TQ0MD2	TQ0MD1	TQ0MD0
TQ0CTL1	0	0	0	0	0	1	1	0

1, 1, 0:
脉宽测量模式

（c）TMQ0 输入/输出控制寄存器 1（TQ0IOC1）

	TQ0IS7	TQ0IS6	TQ0IS5	TQ0IS4	TQ0IS3	TQ0IS2	TQ0IS1	TQ0IS0
TQ0IOC1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

选择TIQ00 引脚输入的有效沿

选择TIQ01 引脚输入的有效沿

选择TIQ02 引脚输入的有效沿

选择TIQ03 引脚输入的有效沿

（d）TMQ0 选项寄存器 0（TQ0OPT0）

	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0				TQ0OVF
TQ0OPT0	0	0	0	0	0	0	0	0/1

溢出标志

图 8-67. 脉冲宽度测量模式时的寄存器设置（2/2）

（e）TMQ0 计数器读取缓冲寄存器（TQ0CNT）

通过读取该寄存器，可以读出 16 位计数器的值。

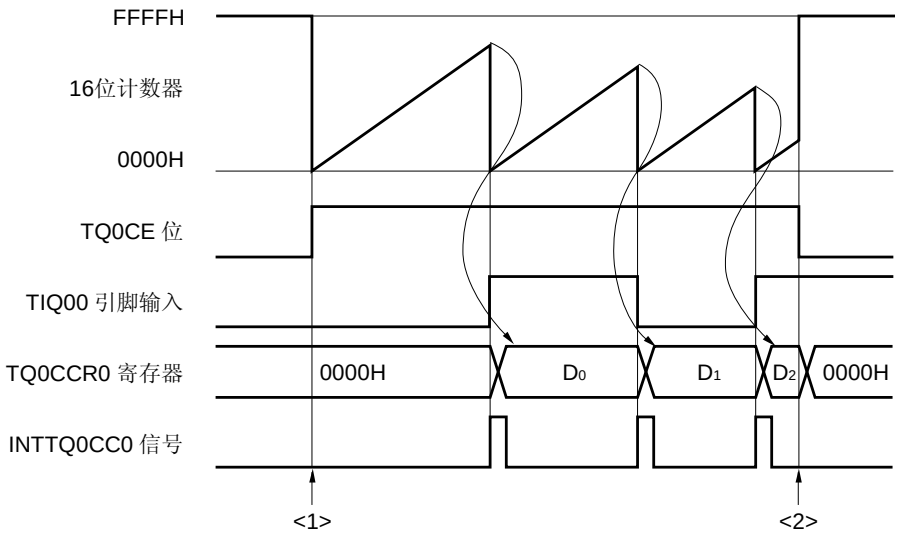
（f）TMQ0 捕获/比较寄存器 0 至 3（TQ0CCR0 至 TQ0CCR3）

当检测到有效边沿输入 TIQ0m 引脚时，这些寄存器保存 16 位计数器的当前计数值。

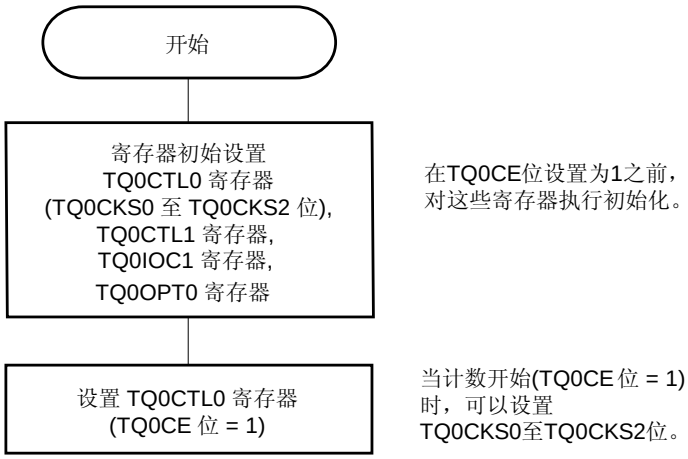
备注 在脉冲宽度测量模式下，不使用 TMQ0 输入/输出控制寄存器 0（TQ0IOC0）和 TMQ0 输入/输出控制寄存器 2（TQ0IOC2）。

（1）脉冲宽度测量模式时的操作

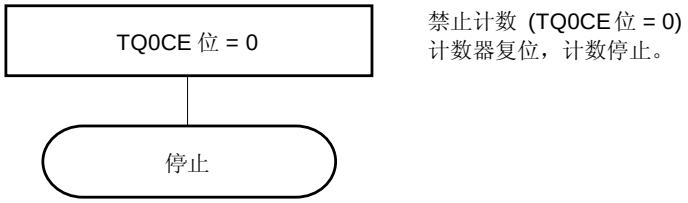
图 8-68. 脉冲宽度测量模式的时序和操作处理



<1> 计数操作开始流程



<2> 计数操作停止流程



（2）使用脉冲宽度测量模式

（a）清除溢出标志（TQ0OVF）

通过读取 TQ0OVF 位可以将溢出标志（TQ0OVF）清除为 0。如果它的值为 1，可以通过 CLR1 指令将 TQ0OVF 位清为 0，或向 TQ0OPT0 寄存器写入 8 位数据（第 0 位是 0）。

8.4.8 定时器输出操作

下表显示了 TOQ00 至 TOQ03 引脚的操作和输出电平。

表 8-8. 各种模式下的定时器输出控制

操作模式	TOQ00 引脚	TOQ01 引脚	TOQ02 引脚	TOQ03 引脚
间隔计时器模式	方波输出			
外部事件计数模式	–			
外部触发脉冲输出模式	方波输出	外部触发脉冲输出	外部触发脉冲输出	外部触发脉冲输出
单次触发脉冲输出模式		单次触发脉冲输出	单次触发脉冲输出	单次触发脉冲输出
PWM 输出模式		PWM（脉宽调制信号）输出	PWM（脉宽调制信号）输出	PWM（脉宽调制信号）输出
自由定时器模式	方波输出（只有使用比较功能时可用）			
脉冲宽度测量模式	–			

表 8-9. 定时器输出控制位对 TOQ00 至 TOQ03 引脚影响的真值表

TQ0IOC0.TQ0OLm 位	TQ0IOC0.TQ0OEm 位	TQ0CTL0.TQ0CE 位	TOQ0m 引脚电平
0	0	×	低电平输出
	1	0	低电平输出
		1	计数前为低电平，开始计数后为高电平
1	0	×	高电平输出
	1	0	高电平输出
		1	计数前为高电平，开始计数后为低电平

备注 m＝0 至 3

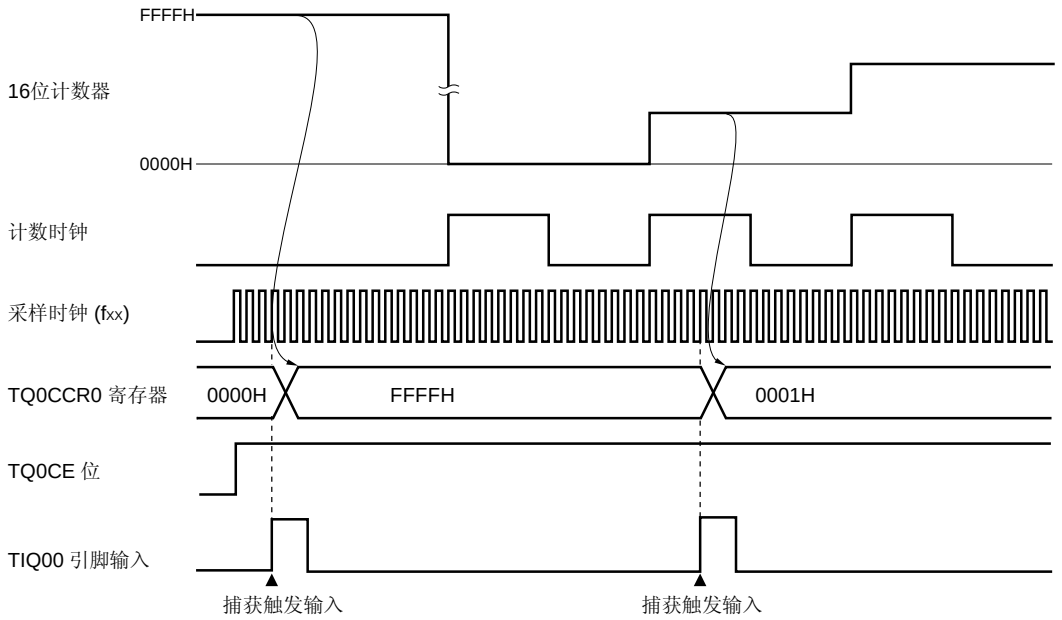
8.5 注意事项

（1）捕获操作

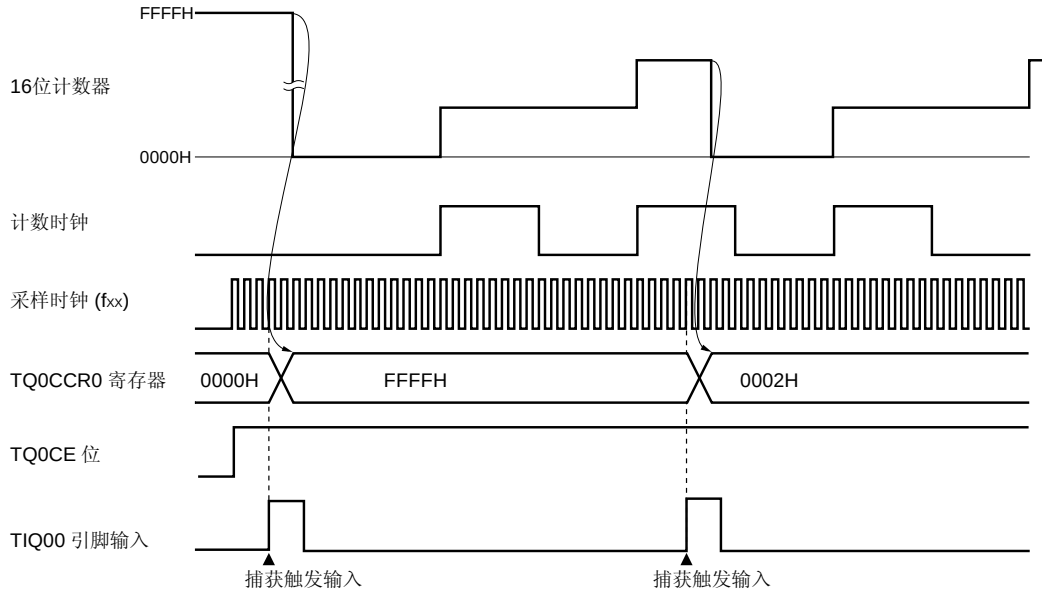
当使用捕获操作时，并且选择 fxx/8, fxx/16, fxx/32, fxx/64, fxx/128 或外部事件计数器（TQ0CLT1.TQ0EEE 位 = 1）作为计数时钟时，如果在 TQ0CE 位置 1 之后立即输入捕获触发，那么捕获到 TQ0CCR0，TQ0CCR1，TQ0CCR2 和 TQ0CCR3 寄存器中的是可能是 FFFFH 而不是 0000H，或者捕获操作可能根本无法执行（不产生捕获中断）。

当使用捕获操作且选择外部事件计数输入作为计数时钟时，在没有外部事件计数输入期间操作结果相同。

（a）自由定时器模式



（b）脉冲宽度测量模式



第九章　16 位间隔定时器M (TMM)

定时器 M（TMM）是 16 位定时器/事件计数器。

V850ES/JG3-L 单片机内置一路 TMM 定时器，TMM0。

9.1　特征

TMQ0 按照预置的间隔产生一个中断，可以输出一个方波。

TMM0 是一个专用的间隔定时器，按照预置的间隔产生一个中断请求，计数时钟可以选择的 8 个时钟源：主时钟 (f_{XX})，主时钟分频 ($f_{XX}/2$, $f_{XX}/4$, $f_{XX}/64$, $f_{XX}/512$)，钟表定时器中断信号(INTWT)，内部时钟 ($f_R/8$)和副时钟(f_{XT})。

TMM0 只支持清零并启动模式，不支持自由运行定时器模式。

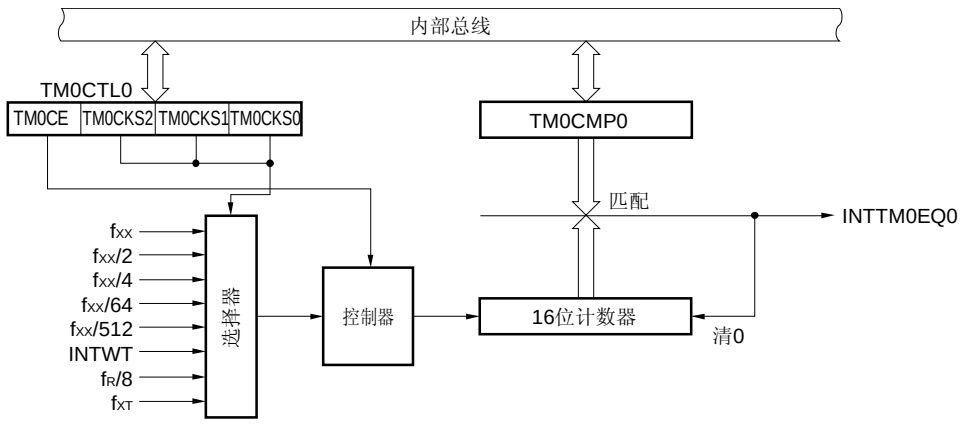
9.2 配置

TMM0 包括下列硬件：

表 9-1. TMM0 的配置

项目	配置
寄存器	16 位计数器 TMM0 比较寄存器 0 （TM0CMP0） TMM0 控制寄存器 0 （TM0CTL0）

图 9-1. TMM0 框图



备注 f_{XX}： 主时钟频率
f_R： 内部时钟频率
f_{XT}： 副时钟频率
INTWT： 钟表定时器中断请求信号

(1) 16 位计数器

该计数器是 16 位计数器，对内部时钟进行计数。
该计数器不可以被读写。

(2) TMM0 比较寄存器 0 （ TM0CMP0 ）

这是一个 16 位的比较寄存器。

(3) TMM0 控制寄存器 0 （ TM0CTL0 ）

这是一个 8 位的寄存器，用于控制 TMM0 的操作。

(4) 选择器

选择器用于选择 16 位计数器的计数时钟。计数时钟可以从 8 个时钟源中选择。

9.3 寄存器

(1) TMM0 控制寄存器（TM0CTL0）

TM0CTL0 寄存器是 8 位寄存器，用于控制定时器 TMM0 的操作。

该寄存器可以按字节读写或按位读写。

复位输入将该寄存器设为 00H。

当 TMM0 运行时，除了 TM0CE 位之外，TM0CTL0 寄存器的位不能被重写为不同的值（这些位只能被重写为先前指定的相同值）。

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	内部时钟操作使能/禁止说明
0	禁止TMM0操作 (16位计数器异步复位)
1	使能TMM0操作
当TM0CE位被清除为0时，TMM0的内部时钟被禁止 (固定为低电平) 且16位计数器被异步复位。	

TM0CKS2	TM0CKS1	TM0CKS0	计数时钟选择
0	0	0	f_{XX}
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/64$
1	0	0	$f_{XX}/512$
1	0	1	INTWT
1	1	0	$f_R/8$
1	1	1	f_{XT}

注意事项 1. 当 TMM0 停止（TM0CE 位=0）时，可以设置 TM0CKS2 至 TM0CKS0 位。
将 TM0CE 的值从 0 改变为 1 的同时，不可以设置 TM0CKS2 到 TM0CKS0 位的值。
2. 请确保将第 3 位至第 6 位清为“0”。

备注 f_{XX}： 主时钟频率
f_R： 内部振荡时钟频率
f_{XT}： 副时钟频率
INTWT： 钟表定时器中断请求信号

(2) TMM0 比较寄存器 0（TM0CMP0）

TM0CMP0 寄存器是一个 16 位的比较寄存器。

该寄存器可以按 16 位宽度进行读取或写入。

复位输入将该寄存器清为 0000H。胆识，如果 TMM0 在停止时被复位，该寄存器被置位 FFFFH。

随时可以用软件将相同的值写入 TM0CMP0 寄存器。

当 TMM0 运行（TM0CTL0.TM0CE 位 = 1）时，无法重写 TM0CMP0 寄存器。

注意事项　不要将 TM0CMP0 寄存器设置为 FFFFH。

复位后: 0000H　R/W　地址: FFFFF694H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TM0CMP0																

9.4 操作

9.4.1 间隔计时器模式

当 TM0CE 位被设置为 1 时，16 位计数器的值由计数时钟同步从 FFFFH 清零为 0000H，计数器开始递增计数。

当 16 位计数器的计数值与 TM0CMP0 寄存器的值相匹配时，则 16 位计数器清除为 0000H，并以指定的间隔生成一个比较匹配中断请求信号（INTTM0EQ0）。

图 9-2. 间隔定时器的配置

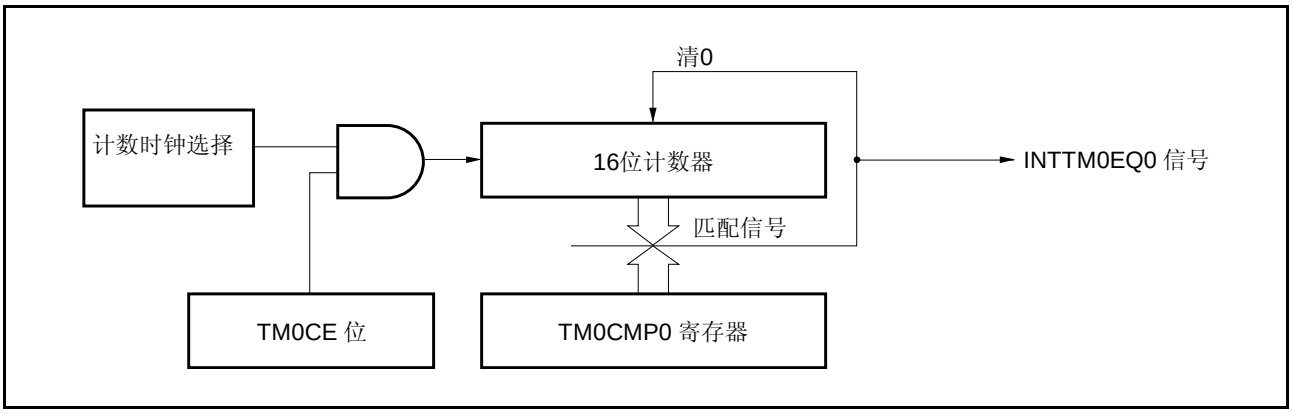
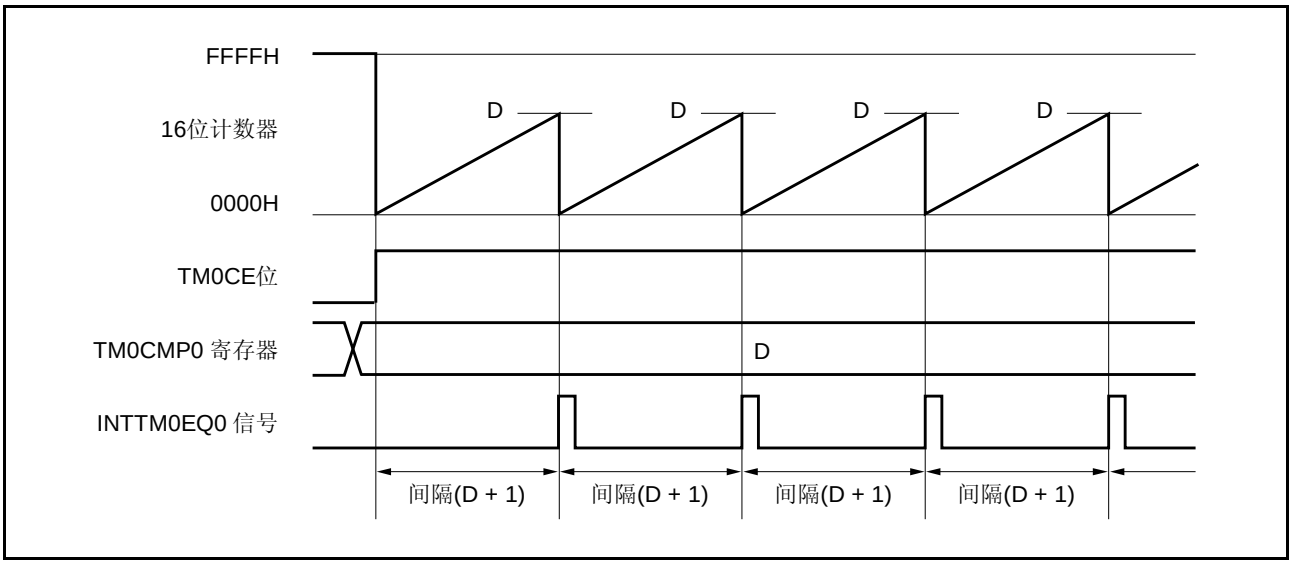


图 9-3. 间隔定时器工作的基本时序



使用间隔定时器模式时设置寄存器的示例如下。

图 9-4. 间隔定时器模式操作时的寄存器设置

(a) TMM0 控制寄存器 0 （TM0CTL0）

	TM0CE					TM0CKS2	TM0CKS1	TM0CKS0
TM0CTL0	0/1	0	0	0	0	0/1	0/1	0/1

选择计数时钟

0: 停止计数
1: 计数使能

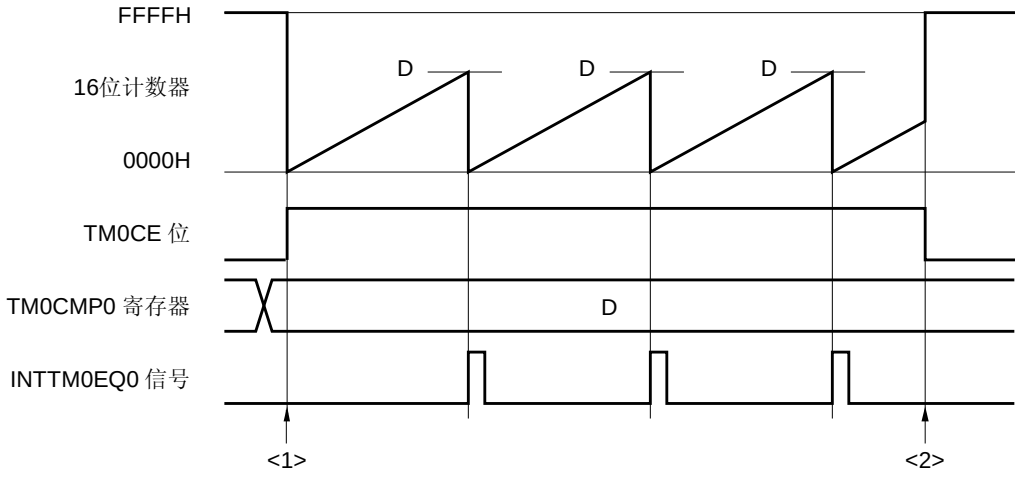
(b) TMM0 比较寄存器 0 （TM0CMP0）

如果 TM0CMP0 寄存器被设为 D，则间隔时间的计算如下：

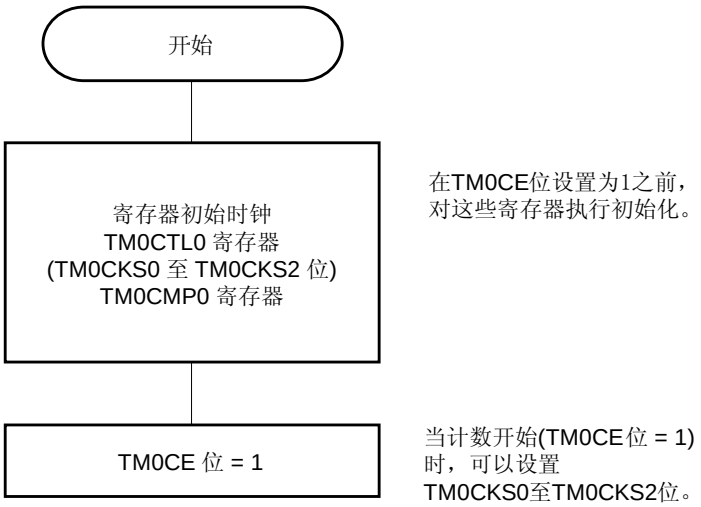
间隔时间＝(D + 1) × 计数时钟周期
(0000H ≤ TM0CMP0 寄存器值 < FFFFH)

(1) 间隔定时器模式的操作

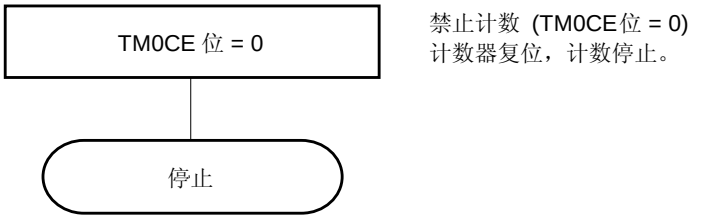
图 9-5. 间隔定时器模式下的时序和操作处理



<1> 计数操作开始流程



<2> 计数操作停止流程

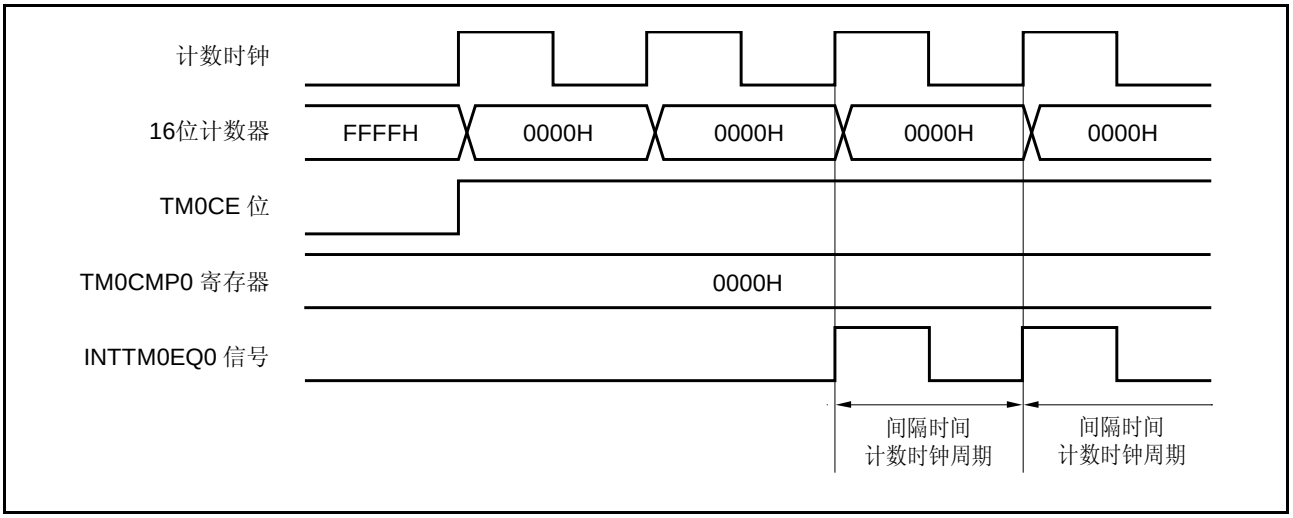


(2) 使用间隔定时器模式

(a) TM0CMP0 寄存器设置为 0000H 时的操作

如果 TM0CMP0 寄存器被清为 0000H，那么每个计数时钟都会产生 INTTM0EQ0 信号。
16 位计数器的值总是 0000H。

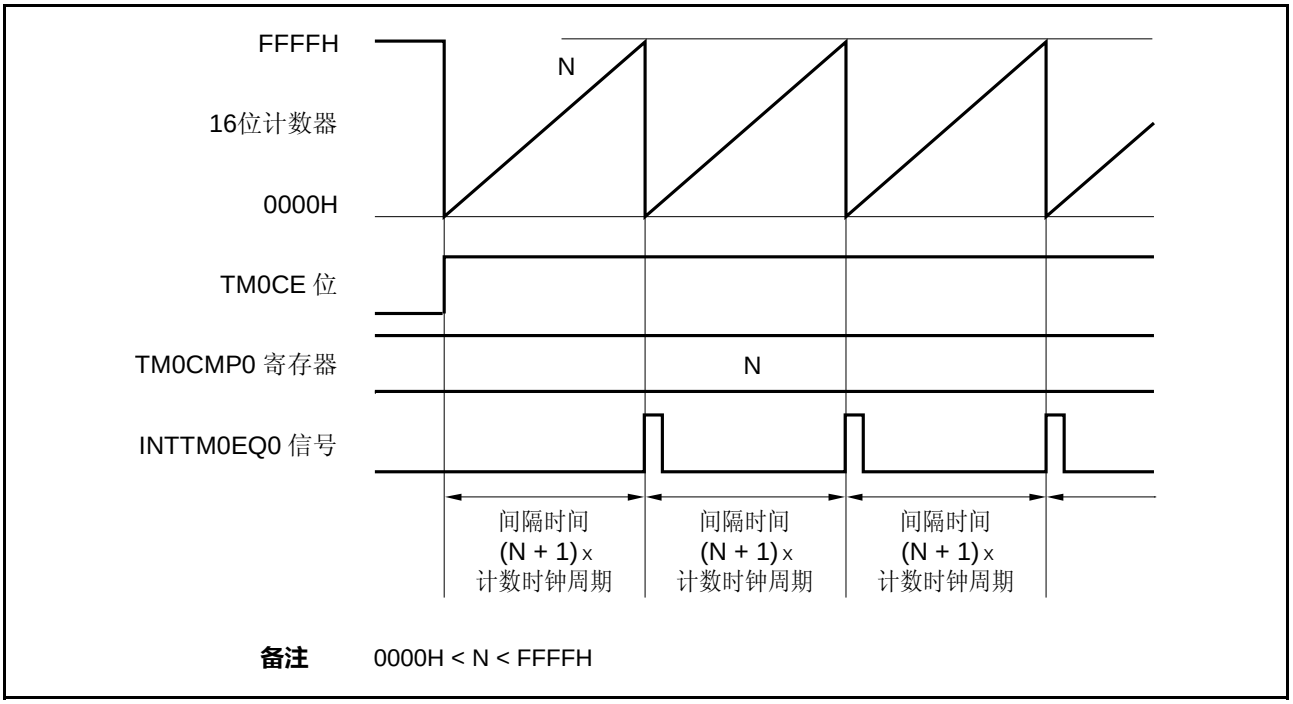
图 9-6. TM0CMP0 寄存器置为 0000H 时的间隔定时器操作



(b) TM0CMP0 寄存器被设置为 N 时的操作

如果 TM0CMP0 寄存器设置为 N，则 16 位计数器最大递增计数至 N，并在下一个计数时刻被同步清为 0000H，并产生 INTTM0EQ0 信号。

图 9-7. TM0CMP0 寄存器置为 0000H，FFFFH 之外时的间隔定时器操作



备注 0000H < N < FFFFH

9.4.2 注意事项

(1) 根据选定的计数时钟，在 TM0CTL0.TM0CE 位置为 1 后，需要 16 位计数器计数以下时间才能启动计数：

选择的计数时钟	计数启动前的最大时间
fXX：	2/f_{XX}
f_{XX}/2	3/f_{XX}
f_{XX}/4	6/f_{XX}
f_{XX}/64	128/f_{XX}
f_{XX}/512	1024/f_{XX}
INTWT	INTWT 信号的第二个上升沿
f_R/8	16/f_R
f_{XT}	2/f_{XT}

(2) TMM0 工作时，禁止重写 TM0CMP0 和 TM0CTL0 寄存器的值。
如果在 TMM0 工作（TM0CE 位 = 1）时对这两个寄存器进行重写，那么操作结果将无法保证。
如果这些寄存器被错误地改写，先将 TM0CTL0.TM0CE 位清除为 0，然后重置这些寄存器。

第十章 钟表定时器功能

10.1 功能

钟表定时器具有以下功能

- 钟表定时器： 使用主时钟或副时钟，以 0.5 或 0.25 秒时间间隔产生中断请求信号(INTWT)。
- 间隔定时器： 按设定的时间间隔产生中断请求信号(INTWTI)。

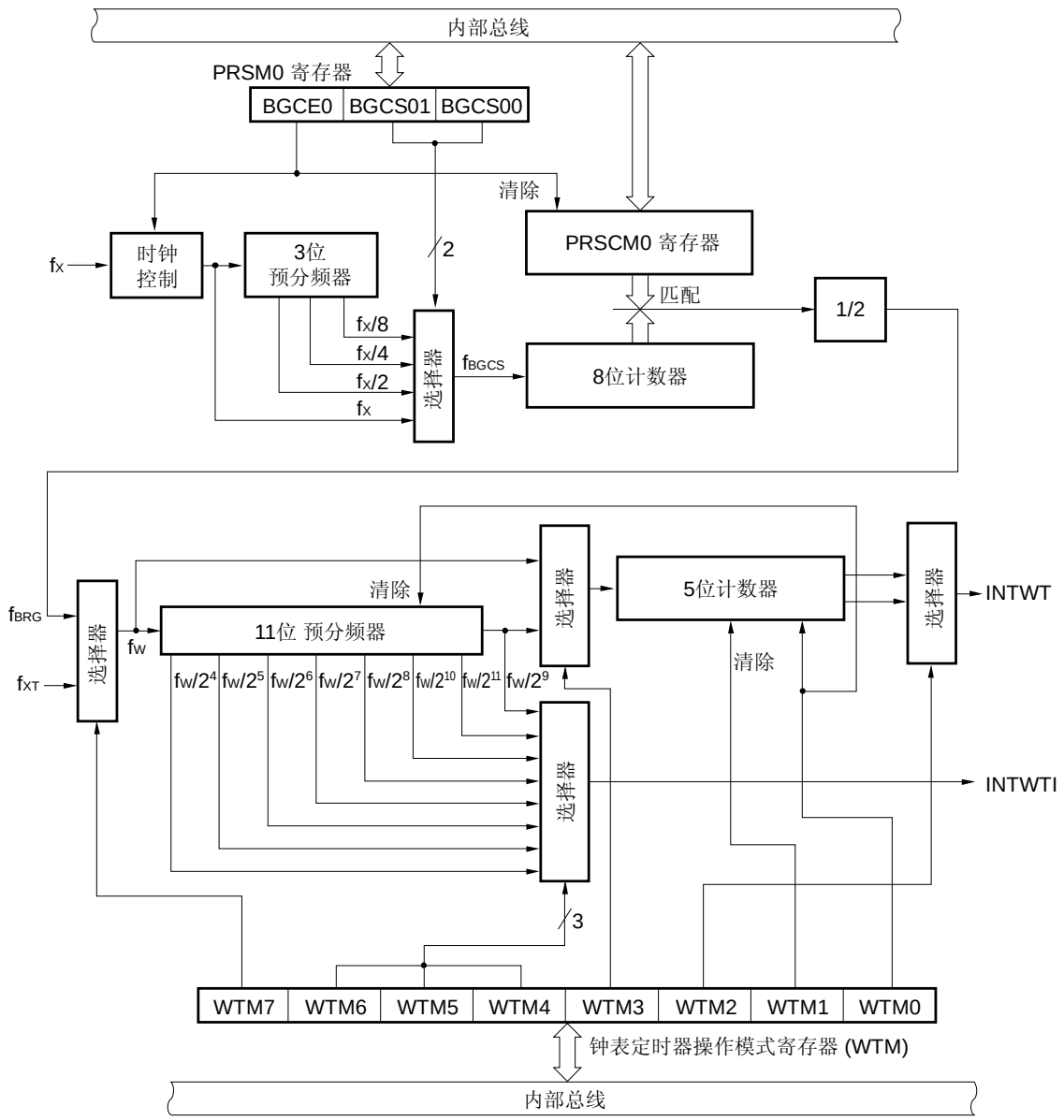
钟表定时器和间隔定时器功能可以同时使用。

注意事项 钟表定时器的 INTWTI 中断与 RTC 的 INTRTC2 中断，钟表定时器的 INTWT 中断与 RTC 的 INTRTC0 中断是服用中断信号，因此不能同时使用。

10.2 配置

钟表定时器的框图如下所示：

图 10-1. 钟表定时器框图



备注 f_X： 主时钟振荡频率

f_{BGCS}： 钟表定时器源时钟频率

f_{BRG}： 钟表定时器计数时钟频率

f_{XT}： 副时钟频率

f_W： 钟表定时器时钟频率

INTWT： 钟表定时器中断请求信号

INTWTI： 间隔定时器中断请求信号

(1) 时钟控制

钟表定时器运行于主时钟时，该模块控制工作时钟（fx）的提供和停止。

(2) 3 位预分频器

该预分频器对 fx 分频，产生 fx/2，fx/4 和 fx/8 频率时钟。

(3) 8 位计数器

该计数器对源时钟(f_{BGCS}) 进行计数。

(4) 11 位预分频器

该预分频器对 fw 进行分频，以产生 $f_w/2^4$ 至 $f_w/2^{11}$ 时钟。

(5) 5 位计数器

该计数器以 fw 或 $f_w/2^9$ 进行计数，并在 $2^4/f_w$, $2^5/f_w$, $2^{12}/f_w$, 或 $2^{14}/f_w$ 时间间隔处产生钟表定时器中断请求信号。

(6) 选择器

钟表定时器有以下 5 个选择器：

- 用于在 fx, fx/2, fx/4 或 fx/8 中选择作为钟表定时器源时钟的选择器。
- 用于选择主时钟（fx）或副时钟（f_{XT}）作为钟表定时器时钟的选择器。
- 用于选择 fw 或 $f_w/2^9$ 作为 5 位计数器计数时钟频率的选择器。
- 用于选择 $2^4/f_w$, $2^{13}/f_w$, $2^5/f_w$ 或 $2^{14}/f_w$ 作为 INTWT 信号产生时间间隔的选择器。
- 用于选择 $2^4/f_w$ 至 $2^{11}/f_w$ 作为间隔定时器中断请求信号（INTWTI）产生时间间隔的选择器。

(7) PRSCM0 寄存器

该寄存器是 8 位比较寄存器，用于设置间隔时间。

(8) PRSM0 寄存器

该寄存器控制提供给钟表定时器的时钟。

(9) WTM 寄存器

该寄存器是 8 位寄存器，用于控制钟表定时器/间隔定时器的工作，并设置中断请求信号产生的时间间隔。

10.3 控制寄存器

钟表定时器提供下列寄存器：

- 预分频模式寄存器 0 (PRSM0)
- 预分频比较寄存器 0 (PRSCM0)
- 钟表定时器操作模式寄存器 (WTM)

(1) 预分频模式寄存器 0 (PRSM0)

PRSM0 寄存器控制钟表定时器计数时钟的发生。

该寄存器可按字节读写或按位读写。

系统复位后，该寄存器被设为 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	主时钟操作允许
0	禁止
1	使能

BGCS01	BGCS00	钟表定时器源时钟 (f_{BGCS})的选择		
			5 MHz	4 MHz
0	0	f_X	200 ns	250 ns
0	1	$f_X/2$	400 ns	500 ns
1	0	$f_X/4$	800 ns	1μs
1	1	$f_X/8$	1.6 μs	2μs

注意事项 1. 钟表定时器工作期间，禁止改变 BGCS00 和 BGCS01 位的值。

2. 在设置 BGCE0 位为 1 之前，先设置 PRSM0 寄存器。

3. 根据主时钟频率设置 PRSM0 和 PRSCM0 寄存器，以得到 32.768 kHz 的 f_{BRG} 频率。

(2) 预分频比较寄存器 0 (PRSCM0)

PRSCM0 寄存器是 8 位比较寄存器。

该寄存器可按字节读取或写入。

系统复位后，该寄存器被设为 00H。

复位后：00H　R/W　地址：FFFFF8B1H

	7	6	5	4	3	2	1	0
PRSCM0	PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00

注意事项 1. 钟表定时器工作期间，禁止重写 PRSCM0 寄存器。

2. 在设置 PRSM0.BGCE0 位为 1 之前，先设置 PRSCM0 寄存器。

3. 根据主时钟频率设置 PRSM0 和 PRSCM0 寄存器，以得到 32.768 kHz 的 f_{BRG} 频率。

关于 f_{BRG} 的计算如下：

$f_{BRG} = f_{BGCS}/2N$

备注 f_{BGCS}：通过 PRSM0 寄存器设置的钟表定时器源时钟

N：　PRSCM0 寄存器的设定值= 1 至 256

但是，当 PRSCM0 寄存器设置为 00H 时，N = 256。

(3) 钟表定时器工作模式寄存器 (WTM)

WTM 寄存器可以使能或禁止钟表定时器的计数时钟和工作，设置预分频器的间隔时间，控制 5 位计数器的操作并且设置钟表标志的设定时间。

设置 WTM 寄存器之前，设置 PRSM0，PRSCM0 寄存器。

该寄存器可按字节读写或按位读写。。

系统复位后，该寄存器被设为 00H。

(1/2)

复位后： 00H R/W 地址： FFFFF680H

	7	6	5	4	3	2	<1>	<0>
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	WTM6	WTM5	WTM4	预分频器间隔时间的选择
0	0	0	0	$2^4/f_W$ (488 us: $f_W = f_{XT}$)
0	0	0	1	$2^5/f_W$ (977 us: $f_W = f_{XT}$)
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 us: $f_W = f_{BRG}$)
1	0	0	1	$2^5/f_W$ (977 us: $f_W = f_{BRG}$)
1	0	1	0	$2^6/f_W$ (1.95 ms: $f_W = f_{BRG}$)
1	0	1	1	$2^7/f_W$ (3.90 ms: $f_W = f_{BRG}$)
1	1	0	0	$2^8/f_W$ (7.81 ms: $f_W = f_{BRG}$)
1	1	0	1	$2^9/f_W$ (15.6 ms: $f_W = f_{BRG}$)
1	1	1	0	$2^{10}/f_W$ (31.2 ms: $f_W = f_{BRG}$)
1	1	1	1	$2^{11}/f_W$ (62.5 ms: $f_W = f_{BRG}$)

(2/2)

WTM7	WTM3	WTM2	钟表标志设置时间的选择
0	0	0	$2^{14}/f_W$ (0.5 s: $f_W = f_{XT}$)
0	0	1	$2^{13}/f_W$ (0.25 s: $f_W = f_{XT}$)
0	1	0	$2^{5}/f_W$ (977μs: $f_W = f_{XT}$)
0	1	1	$2^{4}/f_W$ (488μs: $f_W = f_{XT}$)
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	$2^{5}/f_W$ (977μs: $f_W = f_{BRG}$)
1	1	1	$2^{4}/f_W$ (488μs: $f_W = f_{BRG}$)

WTM1	5位计数器操作的控制
0	操作停止后清零
1	启动

WTM0	钟表定时器操作允许
0	停止操作 (预分频器和5位计数器都清零)
1	允许操作

注意事项 当 WTM0 和 WTM1 位都为 0 时，重写 WTM2 至 WTM7 位。

备注
1. f_W： 钟表定时器时钟频率
2. 当 f_W = 32.768 kHz 时，使用括号内数值。

10.4 操作

10.4.1 钟表定时器操作

钟表定时器运行于副时钟(32.768 kHz)或主时钟，并以固定的精确 0.25 或 0.5 秒产生中断请求信号（INTWT）。

WTM.WTM1 和 WTM.WTM0 位被设置为 11 时，开始计数操作。WTM0 位清除为 0 时，11 位预分频器和 5 位计数器被清零且计数操作停止。

当钟表定时器与间隔定时器同时工作时，通过清除 WTM1 位为 0 然后再清除 5 位计数器，可以对钟表定时器的时间进行调整。此时，钟表定时器可能会产生最大 15.6 ms 的误差 ，但间隔定时器不受影响。

如果主时钟用作钟表定时器的计数时钟，则用 PRSM0.BGCS01 和 BGCS00 位和 PRSCM0 寄存器的 8 位比较值，钟表定时器的计数时钟频率(f_{BRG})可以被设置为 32.768 kHz。

当 PRSM0.BGCE0 位被置为 1 时，f_{BRG} 被提供给钟表定时器。

f_{BRG} 可通过下式进行计算。

$$f_{BRG} = f_X/(2^{m+1} \times N)$$

为了将 f_{BRG} 设置为 32.768 kHz，按以下公式计算设置到 BGCS01 和 BGCS00 位以及 PRSCM0 寄存器的值。

<1> 设置 N = f_X/65,536。设置 m = 0。
<2> 在 N 的第一个十进制数取整，如果为偶数，在取整前将 N 设置为 N/2，m 为 m + 1。
<3> 重复步骤<2> ，直到 N 为奇数 或 m = 3。
<4> 将 N 的第一个十进制数位置取整结果设置到 PRSCM0 寄存器，m 的值设置到 BGCS01 和 BGCS00 位。

例： 当 f_X = 4.00 MHz 时
<1> N = 4,000,000/65,536 = 61.03…, m = 0
<2>, <3>因为 N（取整到第一个十进制位）是奇数，N = 61, m = 0。
<4>设置 PRSCM0 寄存器的值： 3DH (61)，设置 BGCS01 和 BGCS00 位的值： 00

此时，实际的 f_{BRG} 频率如下：

$$f_{BRG} = f_X/(2^{m+1} \times N) = 4{,}000{,}000/(2 \times 61)$$
$$= 32.787 \text{ kHz}$$

备注 m：分频值 (BGCS01 位 和 BGCS00 位设定值) = 0 至 3
N： PRSCM0 寄存器的设定值= 1 至 256
但是，PRSCM0 寄存器设置为 00H 时，N = 256。
f_X： 主时钟振荡频率

10.4.2 作为间隔定时器操作

钟表定时器也可用作间隔定时器，按照事先设定的条件来决定重复产生中断请求信号 (INTWTI)的间隔。

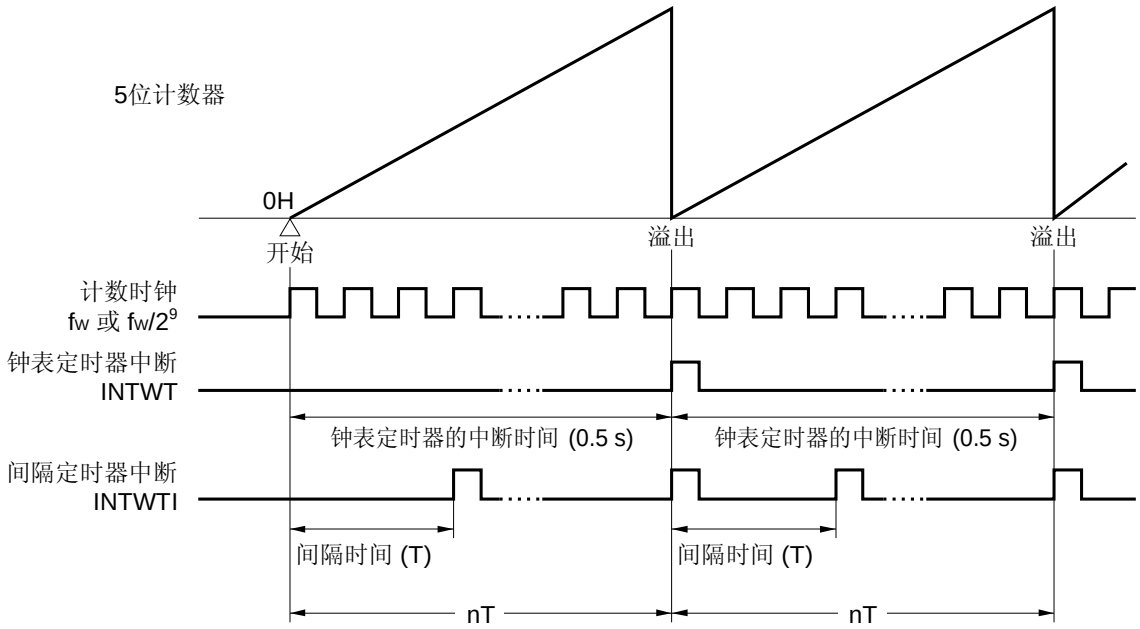
间隔时间由 WTM 寄存器的 WTM4 至 WTM7 位选择。

表 10-1. 间隔定时器的间隔时间

WTM7	WTM6	WTM5	WTM4	间隔时间	
0	0	0	0	$2^4 \times 1/fw$	488 *us*(工作于 $f_W = f_{XT}$ = 32.768 kHz)
0	0	0	1	$2^5 \times 1/fw$	977 *us*(工作于 $f_W = f_{XT}$ = 32.768 kHz)
0	0	1	0	$2^6 \times 1/fw$	1.95 ms (工作于 $f_W = f_{XT}$ = 32.768 kHz)
0	0	1	1	$2^7 \times 1/fw$	3.91 ms (工作于 $f_W = f_{XT}$ = 32.768 kHz)
0	1	0	0	$2^8 \times 1/fw$	7.81 ms (工作于 $f_W = f_{XT}$ = 32.768 kHz)
0	1	0	1	$2^9 \times 1/fw$	15.6 ms (工作于 $f_W = f_{XT}$ = 32.768 kHz)
0	1	1	0	$2^{10} \times 1/fw$	31.3 ms (工作于 $f_W = f_{XT}$ = 32.768 kHz)
0	1	1	1	$2^{11} \times 1/fw$	62.5 ms (工作于 $f_W = f_{XT}$ = 32.768 kHz)
1	0	0	0	$2^4 \times 1/fw$	488 *us* (工作于 $f_W = f_{BRG}$ = 32.768 kHz)
1	0	0	1	$2^5 \times 1/fw$	977 *us* (工作于 $f_W = f_{BRG}$ = 32.768 kHz)
1	0	1	0	$2^6 \times 1/fw$	1.95 ms (工作于 $f_W = f_{BRG}$ = 32.768 kHz)
1	0	1	1	$2^7 \times 1/fw$	3.91 ms (工作于 $f_W = f_{BRG}$ = 32.768 kHz)
1	1	0	0	$2^8 \times 1/fw$	7.81 ms (工作于 $f_W = f_{BRG}$ = 32.768 kHz)
1	1	0	1	$2^9 \times 1/fw$	15.6 ms (工作于 $f_W = f_{BRG}$ = 32.768 kHz)
1	1	1	0	$2^{10} \times 1/fw$	31.3 ms (工作于 $f_W = f_{BRG}$ = 32.768 kHz)
1	1	1	1	$2^{11} \times 1/fw$	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. 钟表定时器的时序和间隔定时器的操作



备注 **1.** 设置钟表定时器中断时间为 0.5 秒时。

2. fw： 钟表定时器时钟频率

当 fw = 32.768 kHz 时，使用括号内的数值（WTM7 = 1, WTM3, WTM2 = 00）。

n： 间隔定时器操作的次数

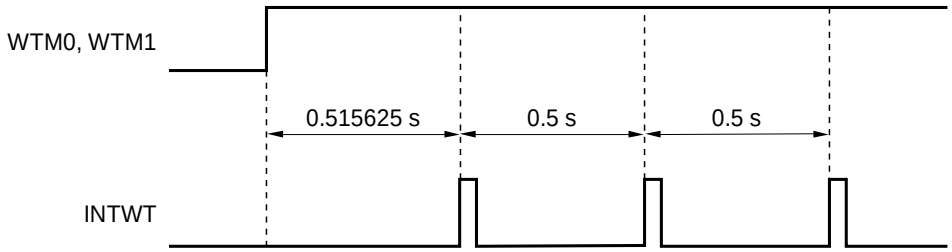
10.5 注意事项

(1) 作为钟表定时器操作

操作使能之后（WTM.WTM1 和 WTM.WTM0 位 = 11），第一个钟表定时器中断请求信号 (INTWT)不能保证在 WTM2 和 WTM3 位指定的精确时间处产生。第二个和随后的 INTWT 信号都在指定时间处产生。

图 10-3. 产生钟表定时器中断请求信号 (INTWT) 的示例(当中断周期 = 0.5 秒时)

第一个 INTWT 信号产生最多需要 0.515625 秒时长（$2^9 \times 1/32768 = 0.015625$ 秒（最大值））。 之后，INTWT 信号每 0.5 秒产生一次。



(2) 钟表定时器和间隔定时器 WT 同时操作

间隔定时器 WT 的间隔时间可以设置为 488 *us* 和 62.5 ms 之间的值，随后无法改变。

不要在钟表定时器工作时停止间隔定时器 WT（将 WTM.WTM0 位清除为 0）。如果 WTM0 位被清除为 0 之后再次设置为 1，钟表定时器的间隔差异可以是 0.5 或 0.25 秒。

第十一章 实时计数器

11.1 实时计数器的功能

实时计数器（RTC）有如下特点。

- 使用内置的年、月、星期、日、小时、分钟和秒计数器，可以计数达 99 年
- 使用 BCD 码[注]显示年、月、星期、日、小时、分钟和秒计数器
- 闹钟中断功能
- 固定周期中断功能（周期：1 个月至 0.5 秒）
- 间隔中断功能（周期：1.95 ms 至 125 ms）
- 1Hz 引脚输出功能
- 32.768kHz 引脚输出功能
- 512HZ 或 16.384kHz 引脚输出功能
- 钟表误差修正功能
- 可以选择运行于副时钟或主时钟

注 1. BCD 码（二进制编码的十进制）用 4 位二进制格式来表示十进制数字的每一位数字。
2. 使用实时定时器的专用波特率发生器将主时钟分频为 32.768kHz。

注意事项 1. 钟表定时器和 RTC 复用中断信号，因此不能同时使用。
2. 如果在进入 RTC 备份模式之后恢复为正常操作模式，RTC 副计数器可能发生最大 1 秒的误差。
3. 在 RTC 备份模式和正常操作模式下，RTC 功能的可用性不同，如下所示。

功能	正常操作模式	RTC 备份模式
年，月，星期，日，时，分，副计数器计数功能	使能	使能
中断功能（闹钟，固定时期，间隔）	使能	禁止
引脚输出功能 (32.768 kHz，16.384 kHz，512 kHz，1Hz)	使能	禁止
钟表错误修正功能	使能	禁止

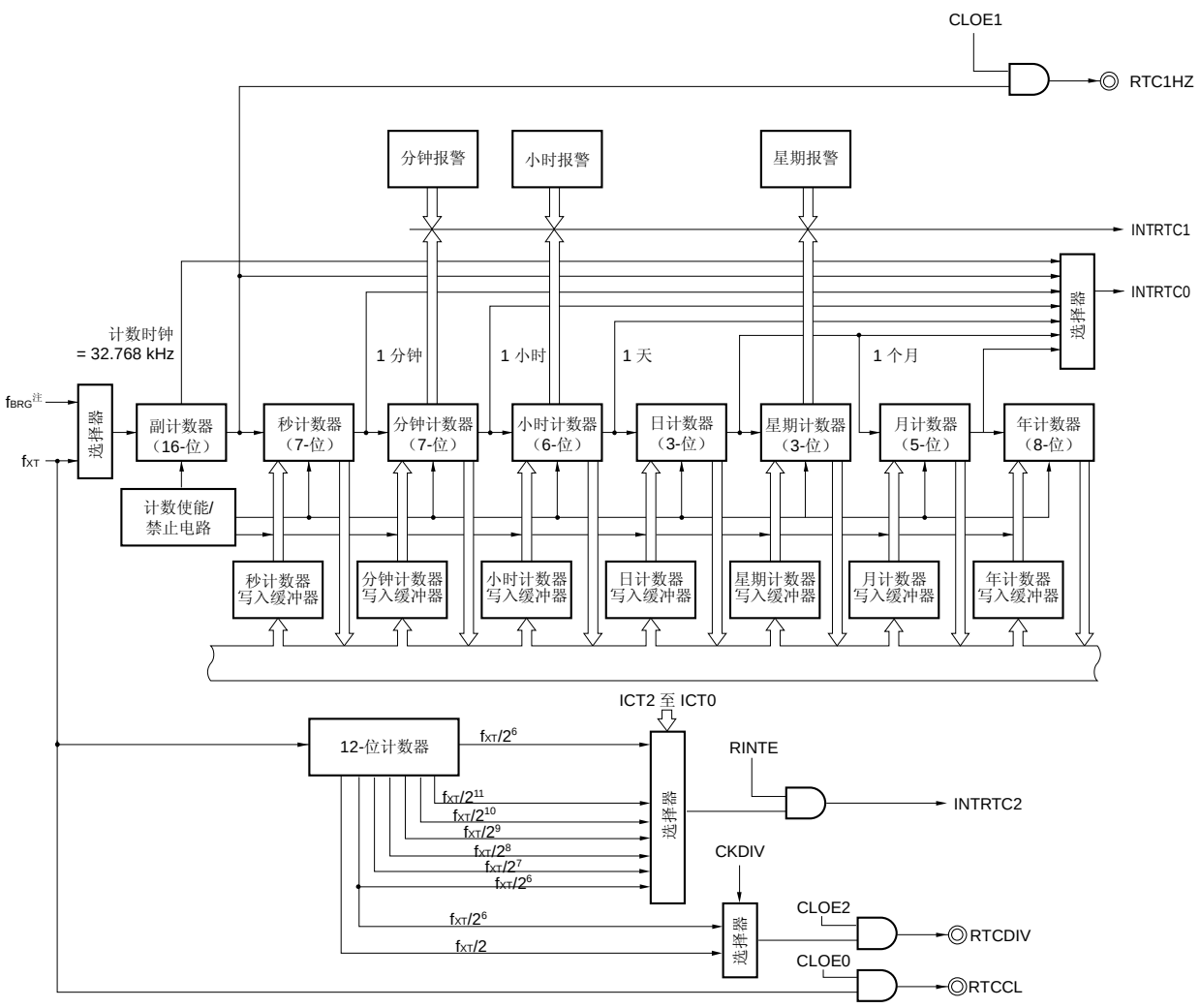
11.2 实时计数器的配置

实时计数器包括下列硬件。

表 11-1. 实时计数器的配置

项目	配置
控制寄存器	实时计数器控制寄存器 0 (RC1CC0) 实时计数器控制寄存器 1 (RC1CC1) 实时计数器控制寄存器 2 (RC1CC2) 实时计数器控制寄存器 3 (RC1CC3) 副计数寄存器 (RC1SUBC) 秒计数寄存器 (RC1SEC) 分钟计数寄存器 (RC1MIN) 小时计数寄存器 (RC1HOUR) 日计数寄存器 (RC1DAY) 星期计数寄存器 (RC1WEEK) 月计数寄存器 (RC1MONTH) 年计数寄存器 (RC1YEAR) 钟表误差修正寄存器 (RC1SUBU) 分钟闹钟寄存器(RC1ALM) 小时闹钟寄存器(RC1ALH) 星期闹钟寄存器(RC1ALW) 预分频模式寄存器 0(PRSM0) 预分频比较寄存器 0(PRSCM0)

图 11-1. 实时计数器的框图



注 f_{BRG} 的详情参见 **11.3 (17) 预分频模式寄存器 0 (PRSM0)** 和 **11.3 (18) 预分频比较寄存器 0 (PRSCM0)。**

备注 f_{BRG}： 实时计数器计数时钟频率

f_{XT}： 副时钟频率

INTRTC0：实时计数器固定周期中断信号

INTRTC1： 实时计数器闹钟匹配中断信号

INTRTC2： 实时计数器间隔中断信号

11.2.1 引脚配置

实时计数器中包括的 RTC 输出可以复用，如表 11-2 所示。当使用每个引脚时必须设置为端口功能（参见表 **4-15 端口引脚用作复用功能引脚**）。

表 11-2. 引脚配置

引脚编号		端口	RTC 输出	其他复用功能
GC	F1			
30	G3	P03	RTC1HZ	INTP0/ADTRG/UCLK
28	H4	P04	RTCDIV	INTP1/ADTRG /RTCCL
28	H4	P04	RTCCL	INTP1/ADTRG /RTCDIV

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

11.2.2 中断功能

RTC 包括下列三种类型的中断信号。

(1) INTRTC0

每 0.5 秒，每秒，每分钟，每小时，每天或者每月产生一个固定周期中断信号。

(2) INTRTC1

闹钟中断信号

(3) INTRTC2

产生周期为 $f_{XT}/2^6$, $f_{XT}/2^7$, $f_{XT}/2^8$, $f_{XT}/2^9$, $f_{XT}/2^{10}$, $f_{XT}/2^{11}$ 或 $f_{XT}/2^{12}$ 的间隔中断信号。

11.3 控制实时计数器的寄存器

以下 18 个寄存器用于控制实时计数器。

(1) 实时计数器控制寄存器 0 (RC1CC0)

RC1CC0 寄存器是 8 位寄存器，用于选择实时计数器的输入时钟。

该寄存器可以按字节或按位进行读取或写入。

RV_{DD} 上电，会使该寄存器被置为 00H。

RV_{DD}上电之后：00H　R/W　地址：FFFFFADDH

符号	7	6	5	4	3	2	1	0
RC1CC0	RC1PWR	RC1CKS注	0	0	0	0	0	0

RC1PWR	实时计数器工作控制
0	停止实时计数器操作。
1	使能实时计数器操作。

RC1CKS	工作时钟选择
0	选择 f_{XT} 作为工作时钟
1	选择 f_{BRG} 作为工作时钟

注在 RTC 备份模式（RTCBUMCTL0.RBMSET = 1）下，请确保将 RC1CKS 位清除为 0。

详情参见 **24.9 RTC 备份模式**。

注意事项 1. 当实时计数器工作时需要停止其操作(RC1PWR = 1 → 0)，请遵循 11.4.8 实时计数器初始化的描述。

2. 只有实时计数器停止(RC1PWR 位 = 0)时，才可以写入 RC1CKS 位。并且，禁止在 RC1PWR 位从 0 被设置为 1 的同事重写 RC1CKS 位。

(2) 实时计数器控制寄存器 0 (RC1CC1)

RC1CC1 寄存器是 8 位寄存器，用于启动或停止实时计数器操作、控制 RTCCL 和 RTC1HZ 引脚，设置 12 小时或 24 小时系统，并设置固定周期中断功能。

该寄存器可以按字节或按位进行读取或写入。

RV_{DD} 上电，会使该寄存器被置为 00H。

RVDD上电之后：00H R/W 地址：FFFFFADEH

符号	7	6	5	4	3	2	1	0
RC1CC1	RTCE	0	CLOE1	CLOE0	AMPM	CT2	CT1	CT0

RTCE	控制每个计数器的工作
0	停止计数器操作。
1	使能计数器操作。

CLOE1	RTC1HZ引脚输出控制
0	禁止 RTC1HZ 引脚输出（1Hz）
1	使能 RTC1HZ 引脚输出（1Hz）

CLOE0	RTCCL引脚输出控制
0	禁止 RTCCL 引脚输出（32.768kHz）
1	使能 RTCCL 引脚输出（32.768kHz）

AMPM	选择12小时/24小时系统
0	12 小时系统(a.m. 和 p.m. 显示)
1	24 小时系统

CT2	CT1	CT0	选择固定周期中断(INTRTC0)
0	0	0	不使用固定周期中断
0	0	1	每 0.5 秒一次(与秒累加同步)
0	1	0	每 1 秒一次(与秒累加同时)
0	1	1	每 1 分钟一次(每分钟的 00 秒)
1	0	0	每 1 小时一次 (每小时的 00 分钟 00 秒)
1	0	1	每 1 日一次 (每日的 00 小时 00 分钟 00 秒)
1	1	×	每 1 月一次 (每月的某天 a.m 00 小时 00 分 00 秒)

注意事项

1. **当 RTCE 位置 1 时，禁止向 RTCE 位写入 0。按照 11.4.8 实时计数器初始化的描述，通过清除 RC1PWR 来对 RTCE 位置 0。**
2. **当 CLOE1 位的设置改变时，RTC1HZ 输出的操作如下：**
   - **从 0 改变为 1 时：在两个时钟(2 × 32.768 kHz)内，RTC1HZ 输出一个 1Hz 脉冲。**
   - **从 1 改变为 0 时：在两个时钟(2 × 32.768 kHz)内，RTC1HZ 输出停止（固定为低电平）。**
3. **关于设置或改变 AMPM 位，参见 11.4.1 初始化设置和 11.4.2 在实时计数器工作期间重写每个计数器。并且，当 AMPM 位被重写时，RC1HOUR 寄存器也被重置。**
4. **关于在实时计数器工作(RC1PWR 位 = 1)时重写 CT2 至 CT0 位，参见 11.4.4 在实时计数器工作期间改变 INTRTC0 中断设置。**

备注 在 RTC 备份模式下，固定周期被屏蔽(RTC0MK 位 = 1) 且 RTCCL 和 RTC1HZ 引脚输出停止。

(3) 实时计数器控制寄存器 2 (RC1CC2)

RC1CC2 寄存器是 8 位寄存器，用于控制闹钟中断功能和计数器等待时间。

该寄存器可以按字节或按位进行读取或写入。

RVDD 上电，会使该寄存器被置为 00H。

RVDD上电之后：00H　R/W　地址：FFFFFADFH

符号	7	6	5	4	3	2	1	0
RC1CC2	WALE	0	0	0	0	0	RWST	RWAIT

WALE	闹钟中断（INTRTC1）操作控制
0	在闹钟匹配时不产生中断
1	在闹钟匹配时产生中断

RWST	实时计数器等待状态
0	计数器操作中
1	从秒到年计数器向上计数停止 （可以读或写计数器的值）
该状态标志显示 RWAIT 位的设置是否有效。 读或写计数器值之前，请确认 RWST 位为 1。	

RWAIT	实时计数器等待控制
0	设置计数器操作
1	停止从秒到年计数器的计数 （可以读或写计数器的值）
该位控制计数器的操作。 当读或写计数器值时，确保该位写入“1”。 如果在 RWAIT = 1 时 RSUBC 寄存器溢出，溢出信息被保留在内部，并且在 RWAIT 位被写入 0 之后的两个时钟内，RC1SEC 寄存器向上计数。 但是，如果 RWAIT 位为 1 时重写秒计数器的值，保留的溢出信息被丢弃。	

注意事项
1. **关于在实时计数器工作(RC1PWR 位 = 1)时重写 WALE 位，参见 11.4.5 在实时计数器工作期间改变 INTRTC1 中断设置。**
2. **当读取或写入每个计数器值时，确认 RWST 位被置 1。**
3. **当每个计数器被写入时，RWST 位不变为 0，即使 RWAIT 位已经被置 0。当每个计数器写入完成后，该位变为 0。**

备注　在 RTC 备份模式下，闹钟中断被屏蔽(RTC1MK 位 = 1)。

(4) 实时计数器控制寄存器 2 (RC1CC3)

RC1CC3 寄存器是 8 位寄存器，用于控制间隔中断功能和 RTCDIV 引脚。

该寄存器可以按字节或按位进行读取或写入。

RV_{DD} 上电，会使该寄存器被置为 00H。

RV_{DD}上电之后：00H　R/W　地址：FFFFFAE0H

符号	7	6	5	4	3	2	1	0
RC1CC3	RINTE	CLOE2	CKDIV	0	0	ICT2	ICT1	ICT0

RINTE	间隔中断（INTRTC2）控制
0	不产生间隔中断
1	产生间隔中断

CLOE2	RTCDIV引脚输出控制
0	禁止 RTCDIV 引脚输出
1	使能 RTCDIV 引脚输出

CKDIV	RTCCL引脚输出控制
0	RTCCL 引脚输出 512HZ（1.95 ms）
1	RTCCL 引脚输出 16.384kHz（0.061 ms）

ICT2	ICT1	ICT0	选择间隔中断(INTRTC2)
0	0	0	$2^6/f_{XT}$ (1.953125 ms)
0	0	1	$2^7/f_{XT}$ (3.90625 ms)
0	1	0	$2^8/f_{XT}$ (7.8125 ms)
0	1	1	$2^9/f_{XT}$ (15.625 ms)
1	0	0	$2^{10}/f_{XT}$ (31.25 ms)
1	0	1	$2^{11}/f_{XT}$ (62.5 ms)
1	1	×	$2^{12}/f_{XT}$ (125 ms)

注意事项 1. 关于在实时计数器工作（RC1PWR 位= 1）时重写 RINTE 位，参见 11.4.7 在实时计数器工作期间改变 INTRTC2 中断设置。

2. 当 CLOE2 位的设置改变时，RTCDIV 输出的操作如下：

- **从 0 改变为 1 时：　在两个时钟(2 × 32.768kHz)内，输出 CKDIV 位设置的脉冲。**
- **从 1 改变为 0 时：　在两个时钟(2 × 32.768kHz)内，RTCDIV 输出停止（固定为低电平）。**

3. 关于在实时计数器工作（RC1PWR 位= 1）时重写 ICT2 至 ICT0 位，参见 11.4.7 在实时计数器工作期间改变 INTRTC2 中断设置。

备注　在 RTC 备份模式下，间隔中断和 RTCDIV 引脚输出被停止。

(5) 副计数寄存器 (RSUBC)

RSUB 寄存器是 16 位寄存器，用于计数实时计数器的 1 秒参考时间。它的值是 0000H 至 7FFFH，用 32.768kHz 的时钟来计数 1 秒钟。

该寄存器是只读的，按照 16 位宽度。

RV_{DD} 上电，会使该寄存器被置为 0000H。

注意事项 1. 当使用 RC1SUBU 寄存器修正，值可能变为 8000H 或更大。

2. 写入秒计数寄存器，该寄存器被清零。

3. 如果在操作期间读取该寄存器，不能保证读出值的有效性，因为读到的是变化值。

RV_{DD} 上电之后：0000H　R　地址：FFFFFAD0H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RC1SUBC																

(6) 秒计数寄存器 (RC1SEC)

RC1SEC 寄存器是 8 位寄存器，它的值为 0 至 59（十进制），表示秒的当前计数值。

当副计数器溢出时，它向上计数。

当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（32.768 kHz）后被写入计数器。以 BCD 码格式设置 00 至 59 的十进制数到该寄存器。如果设置值超出范围，寄存器值在 1 个周期后返回正常值。

该寄存器可以按字节进行读取或写入。

RV_{DD} 上电，会使该寄存器被置为 00H。

注意事项 禁止向 RC1SEC 寄存器设置 00 至 59 之外的值。

备注 当读取或写入 RC1SEC 寄存器时，参见 **11.4.1 初始化设置**，**11.4.2 在实时计数器工作期间重写每个计数器**和 **11.4.3 在实时计数器工作期间读取各个计数器**。

RV_{DD}上电之后：00H　R/W　地址：FFFFFAD2H

	7	6	5	4	3	2	1	0
RC1SEC	0							

(7) 分钟计数寄存器 (RC1MIN **)**

RC1MIN 寄存器是 8 位寄存器，它的值为 0 至 59（十进制），表示分钟的当前计数值。
当秒计数器溢出时，它向上计数。
当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（32.768 kHz）后被写入计数器。以 BCD 码格式设置 00 至 59 的十进制数到该寄存器。
该寄存器可以按字节进行读取或写入。
RV_{DD} 上电，会使该寄存器被置为 00H。

注意事项 禁止向 RC1MIN 寄存器设置 00 至 59 之外的值。

备注 当读取或写入 RC1MIN 寄存器时，参见 **11.4.1 初始化设置 11.4.2 在实时计数器工作期间重写每个计数器**和 **11.4.3 在实时计数器工作期间读取各个计数器**。

RV_{DD}上电之后：00H R/W 地址：FFFFFAD3H

	7	6	5	4	3	2	1	0
RC1MIN	0							

(8) 小时计数寄存器 (RC1HOUR**)**

RC1HOUR 寄存器是 8 位寄存器，它的值为 0 至 23 或 1 至 12（十进制），表示小时的当前计数值。
当分钟计数器溢出时，它向上计数。
当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（32.768 kHz）后被写入计数器。以 BCD 码格式设置 00 至 59 的十进制数到该寄存器。
当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。以 BCD 码设置 0 至 23，01 至 12 或 21 至 32 到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。
该寄存器可以按字节进行读取或写入。
RV_{DD} 上电，会使该寄存器被置为 12H。
但是在 RV_{DD} 上电之后 AMPM 位被置为 1，该寄存器的值清为 00H。

注意事项
1. **当 AMPM = 0 (如果选择 12 小时系统)时，RC1HOUR 寄存器的第 5 位表示 a.m. (0) 或 p.m. (1) 。**
2. **禁止对 RC1HOUR 寄存器设置 01 至 12，21 至 32 (AMPM bit= 0)或 00 至 32(AMPM bit = 1)之外的值。**

备注 当读取或写入 RC1HOUR 寄存器时，参见 **11.4.1 初始化设置，11.4.2 在实时计数器工作期间重写每个计数器**和 **11.4.3 在实时计数器工作期间读取各个计数器**。

RV_{DD}上电之后：12H R/W 地址：FFFFFAD4H

	7	6	5	4	3	2	1	0
RC1HOUR	0	0						

表 11-3 展示了 AMPM 位设置值，RC1HOUR 寄存器值和时间之间的关系。

表 11-3. 时间数字显示

12 小时显示(AMPM 位 = 0)		24 小时显示(AMPM 位 = 1)	
时间	RC1HOUR 寄存器值	时间	RC1HOUR 寄存器值
0:00 a.m.	12 H	0:00	00H
1:00 a.m.	01 H	1:00	01 H
2:00 a.m.	02 H	2:00	02 H
3:00 a.m.	03 H	3:00	03 H
4:00 a.m.	04 H	4:00	04 H
5:00 a.m.	05 H	5:00	05 H
6:00 a.m.	06 H	6:00	06 H
7:00 a.m.	07 H	7:00	07 H
8:00 a.m.	08 H	8:00	08 H
9:00 a.m.	09 H	9:00	09 H
10:00 a.m.	10 H	10:00	10 H
11:00 a.m.	11 H	11:00	11 H
0:00 p.m.	32 H	12:00	12 H
1:00 p.m.	21 H	13:00	13 H
2:00 p.m.	22 H	14:00	14 H
3 :00 p.m.	23 H	15:00	15 H
4:00 p.m.	24 H	16:00	16 H
5:00 p.m.	25 H	17:00	17 H
6:00 p.m.	26 H	18:00	18 H
7:00 p.m.	27 H	19:00	19 H
8:00 p.m.	28 H	20:00	20 H
9:00 p.m.	29 H	21:00	21 H
10:00 p.m.	30 H	22:00	22 H
11:00 p.m.	31 H	23:00	23 H

如果 AMPM 位为 0，RC1HOUR 寄存器值的显示是 12 小时格式，当 AMPM 位为 1 时，显示是 24 小时格式。
在 12 小时格式显示中，a.m.或 p.m. 通过 RCHOUR 的第 5 位来表示：0 表示中午之前(a.m.)，1 表示中午或下午(p.m.)。

(9) 日计数寄存器 (RC1DAY)

RC1DAY 寄存器是 8 位寄存器，它的值为 1 至 31（十进制）表示天的当前计数值。
当小时计数器溢出时，它向上计数。
计数器计数如下。

- 01 至 31 (一月、三月、五月、七月、八月、十月、十二月)
- 01 至 30 (四月、六月、九月、十一月)
- 01 至 29 (闰年的二月)
- 01 至 28 (正常年的二月)

当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（32.768 kHz）后被写入计数器。以 BCD 码格式设置 01 至 31 的十进制数到该寄存器。
该寄存器可以按字节进行读取或写入。
RV_{DD} 上电，会使该寄存器被置为 01H。

注意事项 **禁止向 RC1DAY 寄存器设置 01 至 31 之外的值。设置值不在上述计数范围内，比如“二月 30”也同样被禁止。**

备注 当读取或写入 RC1DAY 寄存器时，参见 **11.4.1 初始化设置**，**11.4.2 在实时计数器工作期间重写每个计数器**和 **11.4.3 在实时计数器工作期间读取各个计数器**。

RV_{DD}上电之后：01H R/W 地址：FFFFFAD6H

	7	6	5	4	3	2	1	0
RC1DAY	0	0						

(10) 星期计数寄存器 (RC1WEEK)

RC1WEEK 寄存器是 8 位寄存器，它的值为 0 至 6（十进制）表示星期的当前计数值。

当日计数器溢出时，它向上计数。

当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（2 × 32.768 kHz）后被写入计数器。以 BCD 码格式设置 00 至 06 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

该寄存器可以按字节进行读取或写入。

RV_{DD} 上电，会使该寄存器被置为 00H。

RV_{DD}上电之后：00H　R/W　地址：FFFFFAD5H

	7	6	5	4	3	2	1	0
RC1WEEK	0	0	0	0	0			

注意事项 1. 禁止对 RC1WEEK 寄存器设置 00 至 06 之外的值。

2. 月计数寄存器和天计数寄存器的对应值不会自动存储到星期寄存器。

在 RV_{DD} 上电解除后，请确保设置如下。

星期几	RC1WEEK
星期日	00H
星期一	01H
星期二	02H
星期三	03H
星期四	04H
星期五	05H
星期六	06H

备注 当读取或写入 RC1WEEK 寄存器时，参见 **11.4.1 初始化设置**，**11.4.2 在实时计数器工作期间重写每个计数器**和 **11.4.3 在实时计数器工作期间读取各个计数器**。

(11) 月计数寄存器 (RC1MONTH)

RC1MONTH 寄存器是 8 位寄存器，它的值为 1 至 12（十进制）表示月的当前计数值。

当日计数器溢出时，它向上计数。

当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（32.768 kHz）后被写入计数器。以 BCD 码格式设置 01 至 12 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

该寄存器可以按字节进行读取或写入。

RV_{DD} 上电，会使该寄存器被置为 01H。

注意事项 禁止向 RC1MONTH 寄存器设置 01 至 12 之外的值。

备注 当读取或写入 RC1MONTH 寄存器时，参见 **11.4.1 初始化设置**，**11.4.2 在实时计数器工作期间重写每个计数器**和 **11.4.3 在实时计数器工作期间读取各个计数器**。

RV_{DD}上电之后：01H R/W 地址：FFFFFAD7H

	7	6	5	4	3	2	1	0
RC1MONTH	0	0	0					

(12) 年计数寄存器 (RC1YEAR)

RC1YEAR 寄存器是 8 位寄存器，它的值为 0 至 99（十进制）表示年的当前计数值。

当月计数器溢出时，它向上计数。

00、04、08…92 和 96 表示闰年。

当数据写入该寄存器时，数据被写入缓冲器，并在 2 个时钟（32.768 kHz）后被写入计数器。以 BCD 码格式设置 00 至 99 的十进制数到该寄存器。

该寄存器可以按字节进行读取或写入。

RV_{DD} 上电，会使该寄存器被置为 00H。

注意事项 禁止向 RC1YEAR 寄存器设置 00 至 99 之外的值。

备注 当读取或写入 RC1YEAR 寄存器时，参见 **11.4.1 初始化设置**，**11.4.2 在实时计数器工作期间重写每个计数器**和 **11.4.3 在实时计数器工作期间读取各个计数器**。

RV_{DD}上电之后：00H R/W 地址：FFFFFAD8H

	7	6	5	4	3	2	1	0
RC1YEAR								

(13) 钟表误差修正寄存器 (RC1SUBCUD)

RC1SUBCUD 寄存器是一个 8 位寄存器，用于高精度修正钟表，当钟表快或慢时，通过改变从副计数器(RSUBC)到秒计数器的溢出值（参考值：7FFFH），来修正钟表。

该寄存器可以按字节或按位进行读取或写入。

RV_{DD} 上电，会使该寄存器被置为 00H。

备注 1. 只有当实时计数器被置为初始值时，可以重写 RC1SUBU 寄存器。请认真参阅 **11.4.1 初始化设置。**

2. 钟表误差修正的详情参见 **11.4.9 实时计数器的钟表误差修正示例**。

RV_{DD}上电之后：00H R/W 地址：FFFFFAD9H

	7	6	5	4	3	2	1	0
RC1SUBU	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	钟表误差修正时序的设置
0	当 RC1SEC（秒计数器）为 00、 20 或 40 秒时，修正钟表误差
1	当 RC1SEC（秒计数器）为 00 秒时，修正钟表误差

F6	钟表误差修正值的设置
0	增加 RC1SUBC 计数值，增加的值由 F5 至 F0 位设置（正向修正） 计算增加值的表达式： {(F5、F4、F3、F2、F1、F0 位的设定值) – 1} × 2
1	减少 RC1SUBC 计数值，减少的值由 F5 至 F0 位设置（负向修正） 计算减少值的表达式： {(/F5、/F4、/F3、/F2、/F1、/F0) + 1} × 2 减小
当 F6 至 F0 位的值是{1/0、0、0、0、0、0、1/0}时，不执行钟表误差修正。	

备注 在 RTC 备份模式下，钟表误差修正停止。

(14) 分钟闹钟寄存器(RC1ALM)

RC1ALM 寄存器是一个 8 位寄存器，用于设置闹钟分钟。

该寄存器可以按字节进行读取或写入。

RV_{DD} 上电，会使该寄存器被置为 00H。

注意事项　　以 BCD 码格式设置 00 至 59 的十进制数到该寄存器。如果设置的值超出范围，闹钟将不作检测。

RV_{DD}上电之后：00H　R/W　地址：FFFFFADAH

	7	6	5	4	3	2	1	0
RC1ALM	0							

(15) 小时闹钟寄存器(RC1ALH)

RC1ALH 寄存器是一个 8 位寄存器，用于设置闹钟小时。

该寄存器可以按字节进行读取或写入。

RV_{DD} 上电，会使该寄存器被置为 12H。

注意事项
1. **以 BCD 码设置 00 至 23，01 至 12 或 21 至 32 到该寄存器。如果设置的值超出范围，闹钟将不检测。**
2. **如果选择了 AMPM 位=0（12 小时格式），RC1ALH 的第 5 位表示 a.m.(0)或 p.m.(1)。**
3. **在 RTC 备份模式下，钟表误差修正停止。**

RV_{DD}上电之后：12H　R/W　地址：FFFFFADBH

	7	6	5	4	3	2	1	0
RC1ALH	0	0						

(16) 星期闹钟寄存器(RC1ALW)

RC1ALW 寄存器是一个 8 位寄存器，用于设置闹钟星期。

该寄存器可以按字节进行读取或写入。

RV_{DD} 上电，会使该寄存器被置为 00H。

注意事项 关于在实时计数器工作(RC1PWR 位 = 1)时重写 RC1ALW 位，参见 11.4.5 在时钟工作期间改变 INTRTC1 中断设置。

RV_{DD}上电之后：00H R/W 地址：FFFFFADCH

	7	6	5	4	3	2	1	0
RC1ALW	0	RC1ALW6	RC1ALW5	RC1ALW4	RC1ALW3	RC1ALW2	RC1ALW1	RC1ALW0

RC1ALWn	闹钟中断星期位（n=0 至 6）
0	如果 RC1WEEK = nH，不产生闹钟中断
1	RC1WEEK 被设置为 nH 时，当 RC1ALM 和 RC1ALH 寄存器指定的时间到达时，产生闹钟中断。

备份 星期几与 RC1WEEK 寄存器之间的关系描述如下。

星期几	RC1WEEK
星期日 Sunday	00H
星期一 Monday	01H
星期二 Tuesday	02H
星期三 Wednesday	03H
星期四 Thursday	04H
星期五 Friday	05H
星期六 Saturday	06H

(a) 闹钟中断设置示例 (RC1ALM, RC1ALH 和 RC1ALW 设置示例)

表 11-4 和 11-5 给出了设置示例，如果设置星期天为 RC1WEEK = 00，设置星期一为 RC1WEEK = 01，设置星期天二为 RC1WEEK = 02, …，设置星期六为 RC1WEEK = 06。

表 11-4. AMPM = 0 的闹钟中断设置示例 (RC1HOUR 寄存器 12 小时显示)

寄存器 / 闹钟设置时间	RC1ALW	RC1ALH	RC1ALM
星期天，7:00 a.m.	01H	07H	00H
星期天/星期一，00:15 p.m.	03H	32H	15H
星期一/星期二/星期五，5:30 p.m.	26H	25H	30H
每天，10:45 p.m.	7FH	30H	45H

表 11-5. AMPM = 1 的闹钟中断设置示例 (RC1HOUR 寄存器 24 小时显示)

寄存器 / 闹钟设置时间	RC1ALW	RC1ALH	RC1ALM
星期天，7:00	01H	07H	00H
星期天/星期一，12:15	03H	12H	15H
星期一/星期二/星期五，17:30	26H	17H	30H
每天，22:45	7FH	22H	45H

(17) 预分频模式寄存器 0 (PRSM0)

PRSM0 寄存器是一个 8 位寄存器，用于控制实时计数器计数时钟的产生(f_{BRG})。

该寄存器可以按字节或按位进行读取或写入。

RV_{DD} 上电，会使该寄存器被置为 00H。

RV_{DD}上电之后：00H　R/W　地址：FFFFF8B0H

符号	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	实时计数器时钟源(f_{BGCS})选择		
			5MHz	4MHz
0	0	f_X	200ns	250ns
0	1	$f_X/2$	400ns	500ns
1	0	$f_X/4$	800ns	1 us
1	1	$f_X/8$	1.6 us	2 us

注意事项 1. 在实时计数器工作期间，不要改变 BGCS00 和 BGCS01 位的值。

2. 置 BGCE0 位为 1 之前，设置 PRSM0 寄存器。

3. 根据获取 32.768 kHz 频率 f_{BRG} 的主时钟来设置 PRSM0 和 PRSCM0 寄存器。

(18) 预分频比较寄存器 0 (PRSCM0)

PRSCM0 寄存器是 8 位比较寄存器。

该寄存器可以按字节进行读取或写入。

RV_{DD} 上电，会使该寄存器被置为 00H。

RV_{DD}上电之后：00H　R/W　地址：FFFFF8B1H

	7	6	5	4	3	2	1	0
PRSCM0	PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00

注意事项 1. 在实时计数器工作期间，不要重写 PRSCM0 寄存器。

2. 将 BGCE0 位设置为 1 之前，先设置 PRSCM0 寄存器。

3. 根据获取 32.768 kHz 频率 f_{BRG} 的主时钟来设置 PRSM0 和 PRSCM0 寄存器。

f_{BRG} 的计算如下。

$$f_{BRG} = f_{BGCS}/2N$$

备注 f_{BGCS}：PRSM0 寄存器设置的钟表定时器时钟源

N：PRSCM0 寄存器的设定值 = 1 至 256

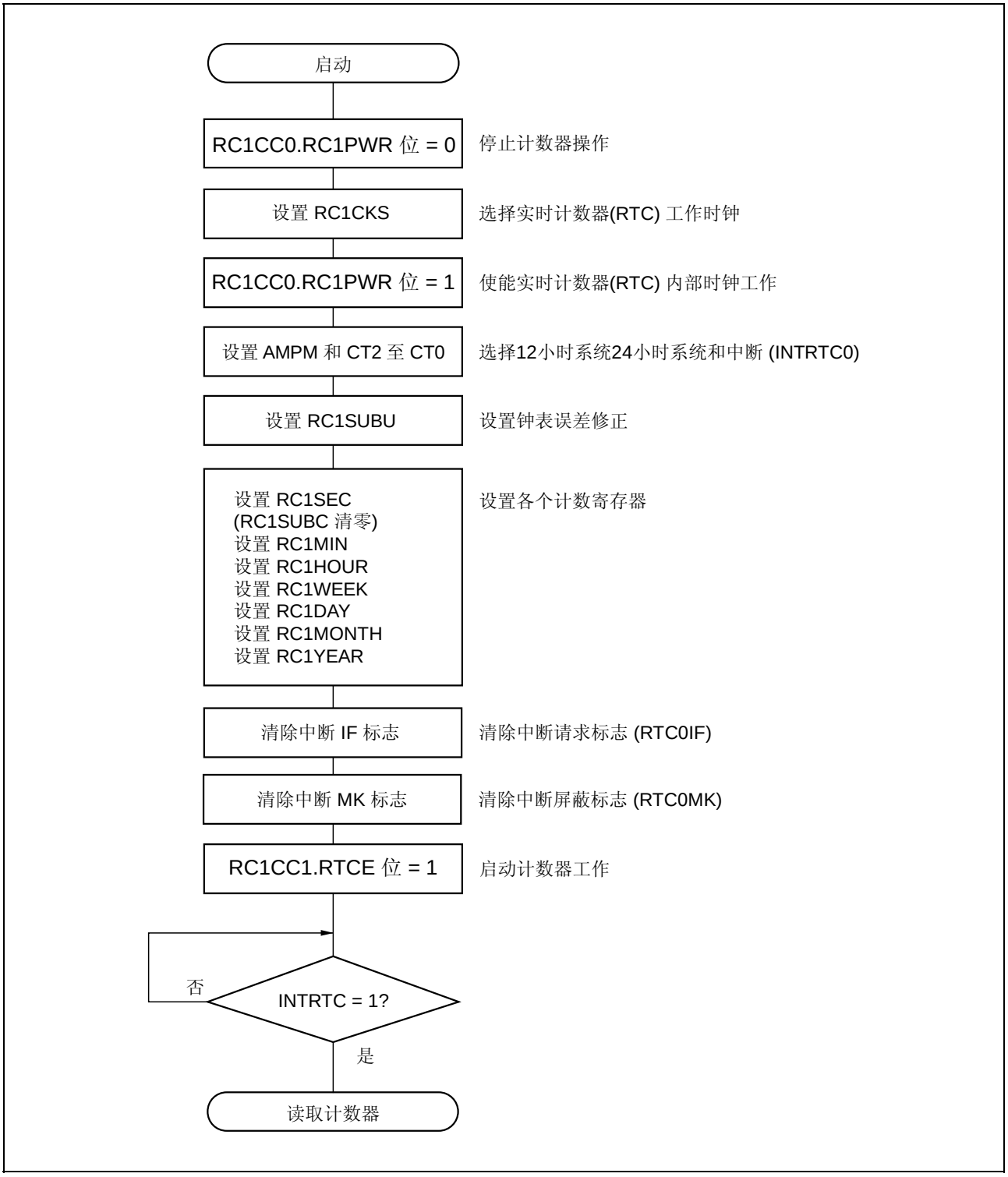
但是，当 PRSCM0 寄存器被设置为 00H 时，N = 256。

11.4 操作

11.4.1 初始化设置

操作钟表功能并执行固定周期中断操作，所需的初始化。

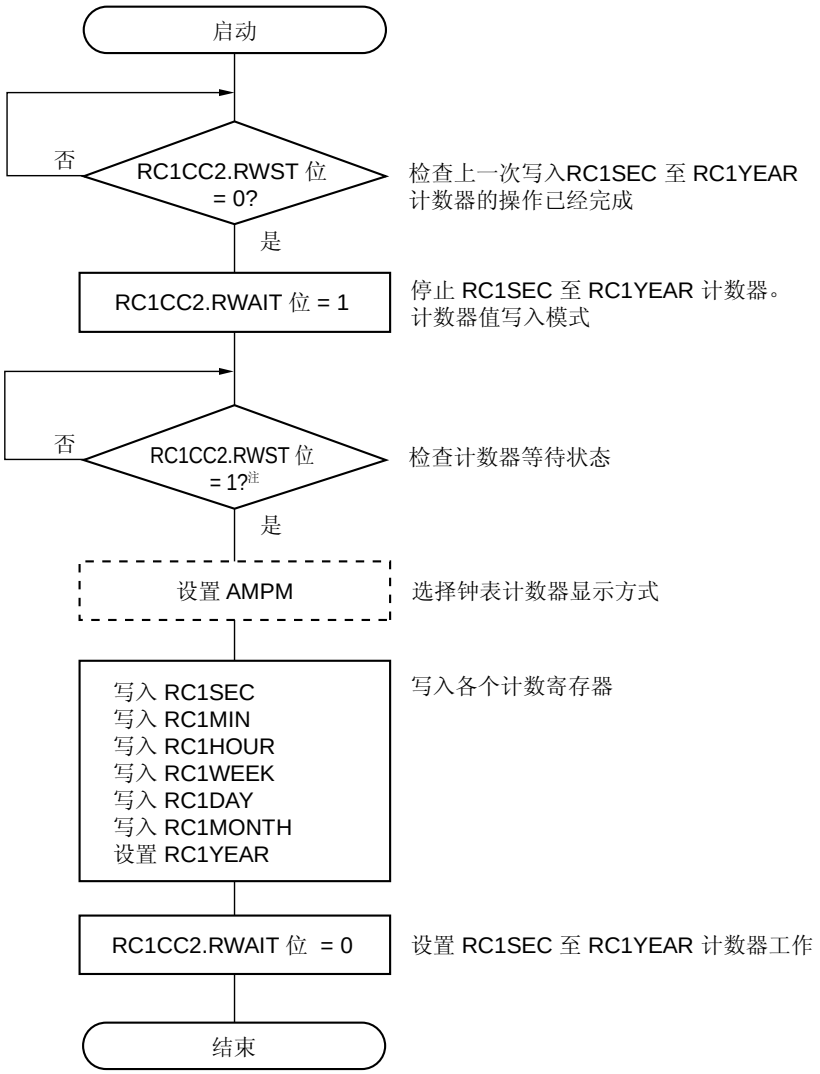
图 11-2. 初始化设置过程



11.4.2 在实时计数器工作期间重写各个计数器

在实时计数器工作期间(RC1PWR = 1)重写各个计数器(RC1SEC, RC1MIN, RC1HOUR, RC1WEEK, RC1DAY, RC1MONTH, RC1YEAR)时，设置如下。

图 11-3. 在时钟工作期间重写各个计数器



注 设置 STOP 模式之前，请确保设置 RWST = 0 。

注意事项 在 1 秒内完成置 RWAIT 为 1 到清 RWAIT 为 0 的系列操作。

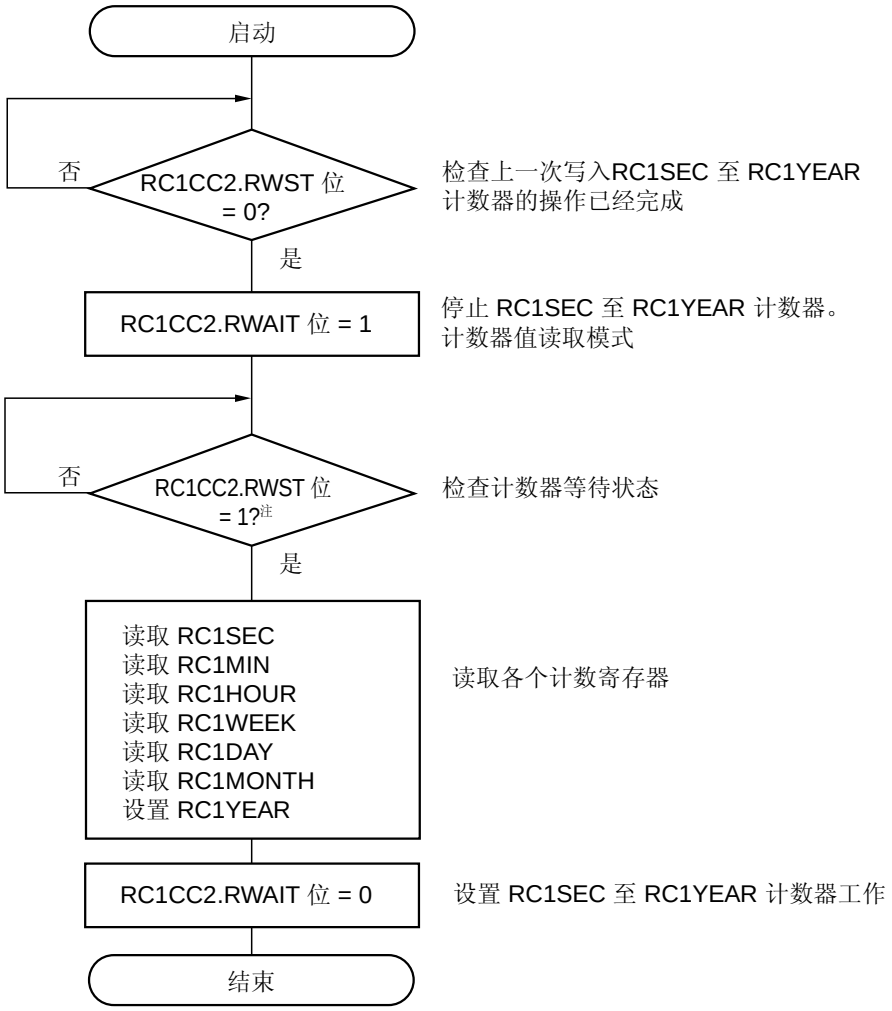
如果设置 RWAIT = 1，RC1SEC 至 RC1YEAR 的操作停止。如果在 RWAIT = 1 时，发生一次从 RC1SUBC 的进位，可以内部保留一个进位。但是有两个或多个进位发生，进位的次数无法保留。

备注 RC1SEC, RC1MIN, RC1HOUR, RC1WEEK, RC1DAY, RC1MONTH, 和 RC1YEAR 可能以任何顺序重写。无需设置所有寄存器，可能仅需读取部分寄存器。

11.4.3 在实时计数器工作期间读取各个计数器

在实时计数器工作期间(RC1PWR = 1, RTCE = 1) 读取各个计数器(RC1SEC, RC1MIN, RC1HOUR, RC1WEEK, RC1DAY, RC1MONTH, RC1YEAR)时，设置如下。

图 11-4. 在时钟工作期间读取各个计数器



注 设置 STOP 模式之前，请确保设置 RWST = 0 。

注意事项 在 1 秒内完成置 RWAIT 为 1 到清 RWAIT 为 0 的系列操作。

如果设置 RWAIT = 1，RC1SEC 至 RC1YEAR 的操作停止。如果在 RWAIT = 1 时，发生一次从 RC1SUBC 的进位，可以内部保留一个进位。但是有两个或多个进位发生，进位的次数无法保留。

备注 RC1SEC, RC1MIN, RC1HOUR, RC1WEEK, RC1DAY, RC1MONTH, 和 RC1YEAR 可能以任何顺序重写。无需设置所有寄存器，可能仅需读取部分寄存器。

11.4.4 在实时计数器工作期间改变INTRTC0 中断设置

如果在实时计数器工作期间(PC1PWR = 1, RTCE =1)，改变INTRTC0中断 (固定周期中断)信号的设置，在输出的INTRTC0中断波形中可能会包括干扰和意外信号。为了屏蔽干扰，在实时计数器工作期间(RC1PWR = 1)要改变INTRTC0中断信号的设置，设置如下。

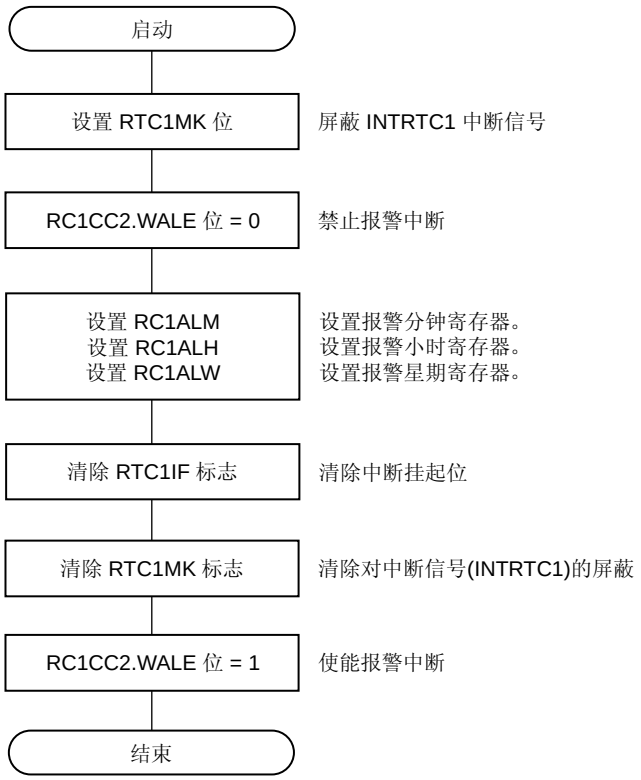
图 11-5. 在时钟工作期间改变 INTRTC0 中断设置

备注 RTC0IF 和 RTC0MK 位的详情参见 **22.3.4 中断控制寄存器 (xxICn)**。

11.4.5 在实时计数器工作期间改变INTRTC1 中断设置

如果在实时计数器工作期间(RC1PWR = 1, RTCE = 1)，改变INTRTC1中断 (闹钟中断)信号的设置，在输出的INTRTC1中断波形中可能会包括干扰和意外信号。为了屏蔽干扰，在实时计数器工作期间(RC1PWR = 1, RTCE = 1) 要改变INTRTC1中断信号的设置，设置如下。

图 11-6. 在时钟工作期间改变 INTRTC1 中断设置

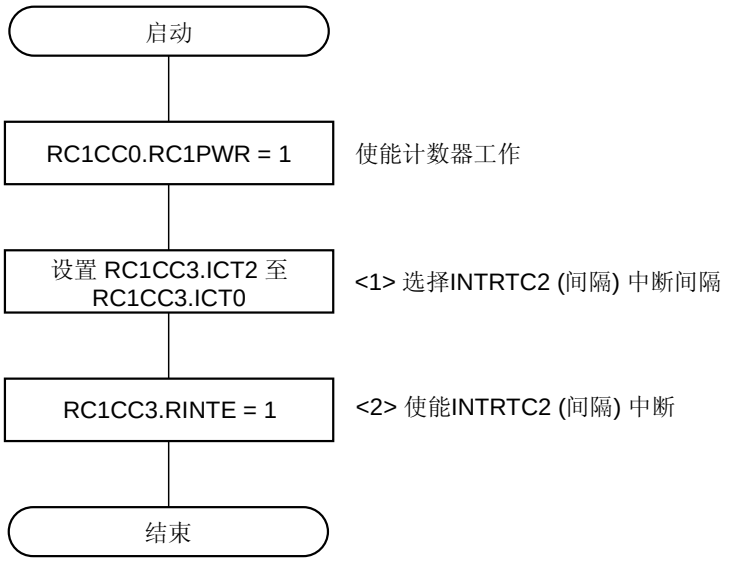


备注 RTC1IF 和 RTC1MK 位的详情参见 **22.3.4 中断控制寄存器 (xxICn)**。

11.4.6 INTRTC2 中断设置初始化

对 INTRTC2 中断(间隔中断)的设置如下。

图 11-7. INTRTC2 中断设置



注意事项 同时设置<1> 和<2>，或者先设置<1>。如果先设置<2>，可能会有意外波形中断发生。

11.4.7 在实时计数器工作期间改变INTRTC2 中断设置

如果在实时计数器的时钟工作期间(PC1PWR = 1, RTCE = 1)，改变INTRTC2中断 (间隔中断)信号的设置，在输出的INTRTC2中断波形中可能会包括干扰和意外信号。为了屏蔽干扰，在实时计数器工作期间(PC1PWR = 1, RTCE = 1) 要改变INTRTC2中断信号的设置，设置如下。

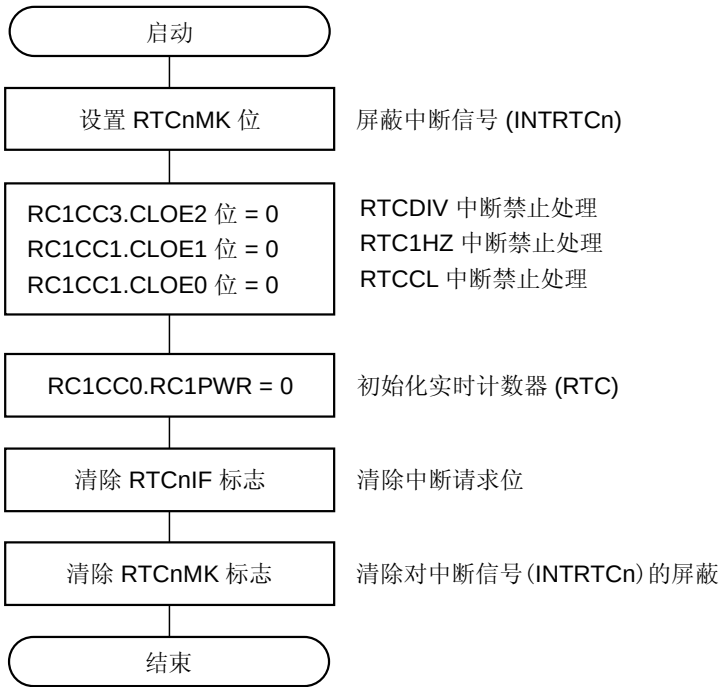
图 11-8. 在时钟工作期间改变 INTRTC2 中断设置

备注 RTC2IF 和 RTC2MK 位的详情参见 **22.3.4 中断控制寄存器 (xxICn)**。

11.4.8 实时计数器初始化

实时计数器初始化过程如下。

图 11-9. 实时计数器初始化



备注 **1.** RTCnIF 和 RTCnMK 位的详情参见 **22.3.4 中断控制寄存器 (xxICn)**。

2. n = 0 至 2

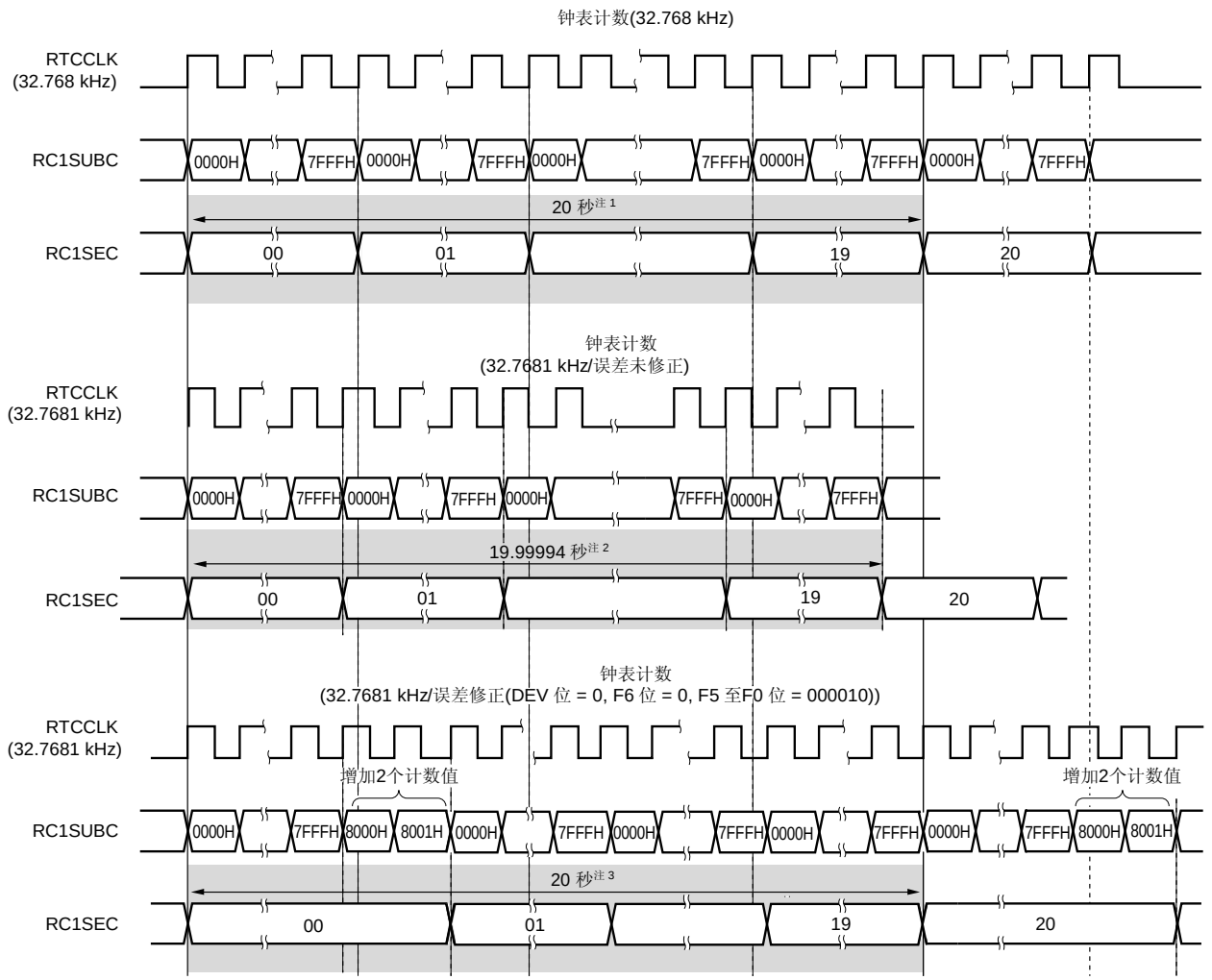
11.4.9 实时计数器的钟表误差修正示例

钟表误差修正功能用于修正连接到 V850ES/JG3-L 微控制器的谐振器的振荡频率偏差。

这里的偏差是指稳定状态偏差，是设计谐振器时频率上的固有偏差。

下面的时序图给出了示例，如果连接到单片机的谐振器设计期望值是 32.768 kHz 但实际为 32.7681kHz，输入时钟会发生错误，并且通过 RC1SUBC 和 RC1SEC 计数操作来修正误差如下所示。

图 11-10. 钟表误差修正示例



注 **1.** RC1SEC 通过对 32.768 kHz 时钟计数 32,768 (0000H 至 7FFFH)次，计数到 1 秒。

2. 当 32.7681 kHz 时钟的 32,768 (0000H 至 7FFFH)次周期输入时，RC1SEC 计数器计数得到的时间计算如下：32,768/3,268.1 ≅ 0.999997 秒。

如果这样的计数持续 20 次，经过的时间计算如下：(32,768/32,768.1) x 20 ≅ 19.99994 秒。这样就导致了 0.00006 秒的误差。

3. 要使用 32.7681 kHz 时间精确计数 20 秒，将 RC1SUBU 的 DEV 和 F6 位清除为 0，并将 F5 至 F0 位设置为 2H (000010B)。结果，每 20 秒会额外增加 2 个周期的计数（当 RC1SEC 计数器计数达到 00, 20 和 40 时），于是在这些点计数的时钟周期数量不是 32,768，而是 32,770 (0000H 至 8001H)，这样就实现了精确的 20 秒时间。

如图 11-10 所示，如果谐振器是正向误差比 32.768 kHz 快，通过增加 RC1SUBC 计数值，钟表可以精确计数。同理，如果谐振器是负向误差比 32.768 kHz 慢，通过减少 RC1SUBC 计数值，钟表也可以精确计数。

RC1SUBC 修正值由 RC1SUBU.F6 至 RC1SUBU.F0 位决定。

F6 位用于决定是增加还是减少 RC1SUBC，F5 至 F0 位决定 RC1SUBC 的值。

(1) 增加 RC1SUBC 计数值

设置 F6 位为 0，RC1SUBC 计数值增加，增加的数值由 F5 至 F0 位设置。

计算增加数值的表达式： (F5 至 F0 位的值– 1) × 2

[增加 RC1SUBC 计数值举例：F6 位 = 0]

如果 F5 至 F0 位设置为 15H (010101B)

(15H – 1) × 2 = 40 (RC1SUBC 计数值增加 40)

RC1SUBC 计数值 = 32,768 + 40 = 32,808

(2) 减少 RC1SUBC 计数值

设置 F6 位为 1，RC1SUBC 计数值减少，减少的数值由 F5 至 F0 位的设定值取反得到。

计算减少数值的表达式：(F5 至 F0 位设定值取反 + 1) × 2

[减少 RC1SUBC 计数值举例：F6 位 = 1]

如果 F5 至 F0 位设置为 15H (010101B)

对 15H (010101B)取反 = 2AH (101010B)

(2AH + 1) × 2 = 86 (RC1SUBC 计数值减少 86)

RC1SUBC 计数值 = 32,768 – 86 = 32,682

(3) DEV 位

DEV 位决定 F6 至 F0 位的设置何时使能。

F6 至 F0 位的设置值在下一个时序体现，但并不是每次都影响 RC1SUBC 计数值。

表 11-6. DEV 位设置

DEV 位的设置	设定值反映到 RC1SUBC 的时序
0	当 RC1SEC 是 00, 20 或 40 秒时
1	当 RC1SEC 是 00 秒时

[F6 至 F0 位的设置值是 0010101B 时的举例]

- 如果 DEV 位是 0
  在 00, 20, 或 40 秒时，RC1SUBC 计数值是 32,808。
  其它时间是 32,768。
- 如果 DEV 位是 1
  在 00, 20, 或 40 秒时，RC1SUBC 计数值是 32,808。
  其它时间是 32,768。

如上所述，为了匹配 RC1SUBC 计数值和谐振器的偏差程序，每 20 秒或 60 秒 RC1SUBC 计数值被修正，而不是每秒都进行修正。

谐振器频率可以被实际修正的范围如下。

- 如果 DEV 位是 0：32.76180000 kHz 至 32.77420000 kHz
- 如果 DEV 位是 1：32.76593333 kHz 至 32.77006667 kHz

DEV 位为 0 时可以修正的频率范围是 DEV 位为 1 时的三倍。

但是，DEV 位为 1 时设定频率的准确度是 DEV 位为 0 时的三倍。

表 11-7 和 11-8 展示了 DEV 和 F6 至 F0 位的设定值，以及可以修正的对应频率。

表 11-7. DEV 位为 0 时可以修正的频率范围

F6	F5 至 F0	RC1SUBC 修正值	连接的时钟频率 （包括稳定状态偏差）
0	000000	无修正	–
0	000001	无修正	–
0	000010	每 20 秒 RC1SUBC 计数值增加 2	32.76810000 kHz
0	000011	每 20 秒 RC1SUBC 计数值增加 4	32.76820000 kHz
0	000100	每 20 秒 RC1SUBC 计数值增加 6	32.76830000 kHz
⋮			
0	111011	每 20 秒 RC1SUBC 计数值增加 120	32.77400000 kHz
0	111110	每 20 秒 RC1SUBC 计数值增加 122	32.77410000 kHz
0	111111	每 20 秒 RC1SUBC 计数值增加 124	32.77420000 kHz (上限)
1	000000	无修正	–
1	000001	无修正	–
1	000010	每 20 秒 RC1SUBC 计数值减少 124	32.76180000 kHz (下限)
1	000011	每 20 秒 RC1SUBC 计数值减少 122	32.76190000 kHz
1	000100	每 20 秒 RC1SUBC 计数值减少 120	32.76200000 kHz
⋮			
1	11011	每 20 秒 RC1SUBC 计数值减少 6	32.76770000 kHz
1	11110	每 20 秒 RC1SUBC 计数值减少 4	32.76780000 kHz
1	11111	每 20 秒 RC1SUBC 计数值减少 2	32.76790000 kHz

表 11-8. DEV 位为 1 时可以修正的频率范围

F6	F5 至 F0	RC1SUBC 修正值	连接的时钟频率 （包括稳定状态偏差）
0	000000	无修正	–
0	000001	无修正	–
0	000010	每 60 秒 RC1SUBC 计数值增加 2	32.76803333 kHz
0	000011	每 60 秒 RC1SUBC 计数值增加 4	32.76806667 kHz
0	000100	每 60 秒 RC1SUBC 计数值增加 6	32.76810000 kHz
⋮			
0	111011	每 60 秒 RC1SUBC 计数值增加 120	32.77000000 kHz
0	111110	每 60 秒 RC1SUBC 计数值增加 122	32.77003333 kHz
0	111111	每 60 秒 RC1SUBC 计数值增加 124	32.77006667 kHz (上限)
1	000000	无修正	–
1	000001	无修正	–
1	000010	每 60 秒 RC1SUBC 计数值减少 124	32.76593333 kHz (下限)
1	000011	每 60 秒 RC1SUBC 计数值减少 122	32.76596667 kHz
1	000100	每 60 秒 RC1SUBC 计数值减少 120	32.76600000 kHz
⋮			
1	11011	每 60 秒 RC1SUBC 计数值减少 6	32.76790000 kHz
1	11110	每 60 秒 RC1SUBC 计数值减少 4	32.76793333 kHz
1	11111	每 60 秒 RC1SUBC 计数值减少 2	32.76796667 kHz

第十二章 看门狗定时器 2

12.1 功能

看门狗定时器 2 是默认启动的看门狗定时器，在复位结束后就立即自动启动。看门狗定时器 2 在复位模式下启动，溢出时间自动设置为内部振荡时钟 = $2^{19}/f_R$。当看门狗定时器 2 溢出时，会产生触发复位的 WDT2RES 信号。

看门狗定时器 2 具有以下特征：

- 它是默认启动的看门狗定时器[注1]
- 当它溢出时，会触发下列操作：
  → 复位模式：当它溢出时，看门狗定时器 2 触发复位（产生 WDT2RES 信号）
  → 不可屏蔽中断请求模式：当它溢出时，看门狗定时器 2 触发 NMI 服务（产生 INTWDT2 信号）[注2]
- 时钟源可以选择主时钟、内部振荡时钟或副时钟

注 1. 在复位模式下，复位释放后看门狗定时器 2 就会自动启动。
当不使用看门狗定时器 2 时，或者在触发系统复位前停止该功能，或者在下一个间隔时间内对看门狗定时器 2 执行一次清除然后再停止其工作。
即使不需要改变默认设置（复位模式，循环检测时间间隔：$f_R/2^{19}$），为了验证操作的正确性，也要对 WDTM2 寄存器执行一次写入操作。

2. 关于不可屏蔽中断请求信号（INTWDT2）的产生而导致的不可屏蔽中断服务程序，详情参见 **22.2.2 （2）自 INTWDT2 的信号**。

12.2 配置

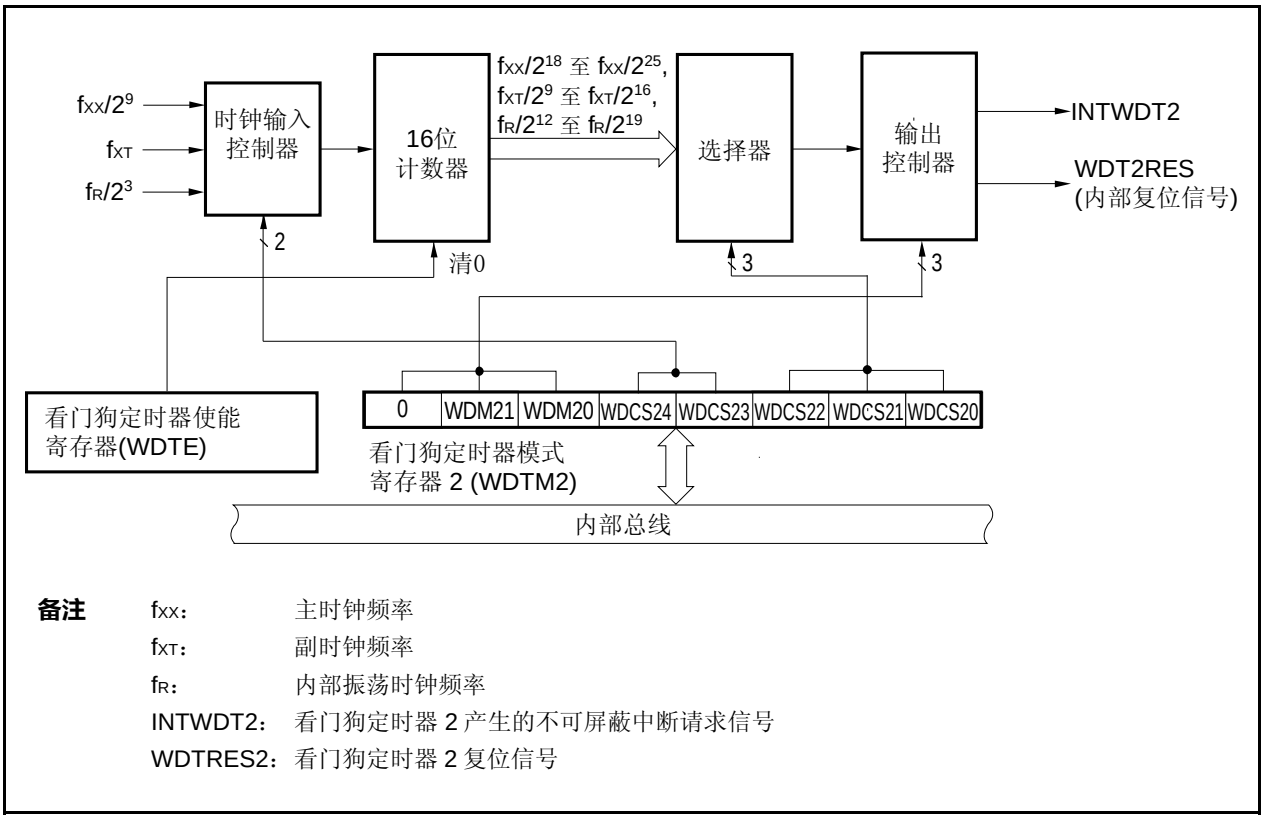
看门狗定时器 2 包括下列硬件：

表 12-1. 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）

看门狗定时器 2 的功能框图如下所示：

图 12-1. 看门狗定时器 2 的功能框图



备注 fXX： 主时钟频率
fXT： 副时钟频率
fR： 内部振荡时钟频率
INTWDT2： 看门狗定时器 2 产生的不可屏蔽中断请求信号
WDTRES2： 看门狗定时器 2 复位信号

12.3 寄存器

（1）看门狗定时器模式寄存器 2（WDTM2）

WDTM2 寄存器用于设置看门狗定时器 2 的溢出时间和工作时钟。

该寄存器可按字节读取或写入。该寄存器读取次数不限，但是仅能在复位解除后写入一次。

复位输入将该寄存器设置为 67H。

注意事项 下列状态下禁止访问 WDTM2 寄存器。而且，如果系统处于等待状态，取消等待状态的唯一方式就是执行复位。详细说明，参见 3.4.9（1）访问片上外设 I/O 专用寄存器。

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

复位后：67H　R/W　地址：FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	WDCS24	WDCS23	WDCS22	WDCS21	WDCS20

WDM21	WDM20	看门狗定时器2操作模式的选择
0	0	停止工作
0	1	不可屏蔽中断请求模式[L] (产生INTWDT2 信号)
1	–	复位模式 (产生WDT2RES 信号)

注 使用选项字节功能时，如果 WDTMD1 位被置位(为 1)，WMD21 和 WDM20 位被固定为 1（用来指定复位模式），且 WDCS24 和 WDCS23 位被固定为 0（指定内部振荡时钟(f_R)作为操作时钟）。详情参见**第三十章 选项字节**。

注意事项

1. **关于 WDCS20 至 WDCS24 位的细节，参见表 12-2 看门狗定时器 2 的循环检测时间间隔。**
2. **复位后，如果对 WDTM2 寄存器执行了两次写入操作，那么将强制产生溢出信号且计数器复位。**
3. **为了有意产生溢出信号，仅需向 WDTM2 寄存器写入两次数据，或者向 WDTE 寄存器写入一次“ACH”之外的数值。**
   但是，当看门狗定时器 2 被设置为“停止运行”时，无论向 WDTM2 寄存器写入两次数据，还是向 WDTE 寄存器写入一次“ACH” 之外的数值，都不会产生溢出信号。
4. **为了停止看门狗定时器 2，将 RCM.RSTOP 位置为 1（停止内部振荡器），并且将 00H 写入 WDTM2 寄存器。如果 RCM.RSTOP 位不能置为 1，就将 WDCS23 位置为 1（$2^n/f_{XX}$ 被选择为时钟，且该时钟可以在 IDLE1、IDLE2、副 IDLE 和副时钟操作模式下被停止）。**

表 12-2. 看门狗定时器 2 的循环检测时间间隔

	WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	选择的时钟	100 kHz (MIN.)	220 kHz (TYP.)	400 kHz (MAX.)
内部振荡时钟	0	0	0	0	0	$2^{12}/f_R$	41.0 ms	18.6 ms	10.2 ms
	0	0	0	0	1	$2^{13}/f_R$	81.9 ms	37.2 ms	20.5 ms
	0	0	0	1	0	$2^{14}/f_R$	163.8 ms	74.5 ms	41.0 ms
	0	0	0	1	1	$2^{15}/f_R$	327.7 ms	148.9 ms	81.9 ms
	0	0	1	0	0	$2^{16}/f_R$	655.4 ms	297.9 ms	163.8 ms
	0	0	1	0	1	$2^{17}/f_R$	1310.7 ms	595.8 ms	327.7 ms
	0	0	1	1	0	$2^{18}/f_R$	2621.4 ms	1191.6 ms	655.4 ms
	0	0	1	1	1	$2^{19}/f_R$	5242.9 ms	2383.1 ms	1310.7 ms
主时钟							f_{XX} = 20 MHz	f_{XX} = 16 MHz	f_{XX} = 10 MHz
	0	1	0	0	0	$2^{18}/f_{XX}$	13.1 ms	16.4 ms	26.2 ms
	0	1	0	0	1	$2^{19}/f_{XX}$	26.2 ms	32.8 ms	52.4 ms
	0	1	0	1	0	$2^{20}/f_{XX}$	52.4 ms	65.5 ms	104.9 ms
	0	1	0	1	1	$2^{21}/f_{XX}$	104.9 ms	131.1 ms	209.7 ms
	0	1	1	0	0	$2^{22}/f_{XX}$	209.7 ms	262.1 ms	419.4 ms
	0	1	1	0	1	$2^{23}/f_{XX}$	419.4 ms	524.3 ms	838.9 ms
	0	1	1	1	0	$2^{24}/f_{XX}$	838.9 ms	1048.6 ms	1677.7 ms
	0	1	1	1	1	$2^{25}/f_{XX}$	1677.7 ms	2097.2 ms	3355.4 ms
副时钟							f_{XT} = 32.768 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		

备注 × = 或者 0 或者 1

（2）看门狗定时器使能寄存器（WDTE）

将“ACH”写入 WDTE 寄存器，可以清除看门狗定时器 2 的计数值，并且重新开始计数。

WDTE 寄存器可以按字节读取或写入（对 WDTE 寄存器执行 1 位存储器操作指令，强制产生一个溢出信号）。

复位输入将该寄存器设为 9AH。

复位后：9AH　R/W　地址：FFFFF6D1H

	7	6	5	4	3	2	1	0
WDTE								

注意事项 1. 当 “ACH”之外的数值被写入 WDTE 寄存器时，强制输出一个溢出信号。

2. 为了有意产生溢出信号，向 WDTE 寄存器写入一次 “ACH” 之外的数据，或者向 WDTM2 寄存器写入两次数据即可。

但是，当看门狗定时器 2 被设置为“停止运行”时，无论向 WDTM2 寄存器写入两次数据，还是向 WDTE 寄存器写入一次“ACH” 之外的数值，都不会产生溢出信号。

3. 寄存器的读出值总是“9AH”（与写入值“ACH”不同）。

12.4 操作

在复位模式下，当复位解除后，看门狗定时器 2 就会立即自动启动。

复位后，仅能使用字节访问立即对 WDTM2 寄存器写入一次。要使用看门狗定时器 2，应使用字节存储处理指令将工作模式以及循环检测时间间隔写入 WDTM2 寄存器。之后，将无法停止看门狗定时器的操作。

WDTM2 的 WDCS24 位至 WDCS20 位用来选择看门狗定时器 2 循环检测间隔时间。

向 WDTE 寄存器写入 ACH 可以清除看门狗定时器 2 的计数值，并且再次开始计数。计数操作开始后，在循环检测时间间隔内将 ACH 写入 WDTE。

如果直到循环检测间隔时间期满，都没有向 WDTE 寄存器写入 ACH，那么根据 WDTM2.WDM21 和 WDTM2.WDM20 位的设定值，将产生一个复位信号（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）时，如果在复位结束之后或待机状态解除后的振荡稳定期间看门狗定时器 2 发生溢出，那么将不发生内部复位，CPU 时钟将切换到内部振荡时钟。

若不使用看门狗定时器 2，则向 WDTM2 寄存器写入 00H。

关于设置为不可屏蔽中断请求模式时的不可屏蔽中断服务的详情，参见 **22.2.2 （2）自 INTWDT2 的信号**。

第十三章 实时输出功能(RTO)

13.1 功能

实时输出功能将预先设置的数据传送至实时输出缓冲寄存器（RTBL0 和 RTBH0）中，然后在产生定时器中断时，通过硬件输出锁存把数据传送到外部设备中。数据传输到外部设备需要使用的引脚构成一个端口，被称为实时输出功能(RTO)。

因为 RTO 能够输出无抖动信号，所以适用于控制步进电机。

在 V850ES/JG3-L 单片机中提供一个 6 位的实时输出端口通道。

实时输出端口可以设置为端口模式或按位设置为实时输出端口模式。

13.2 配置

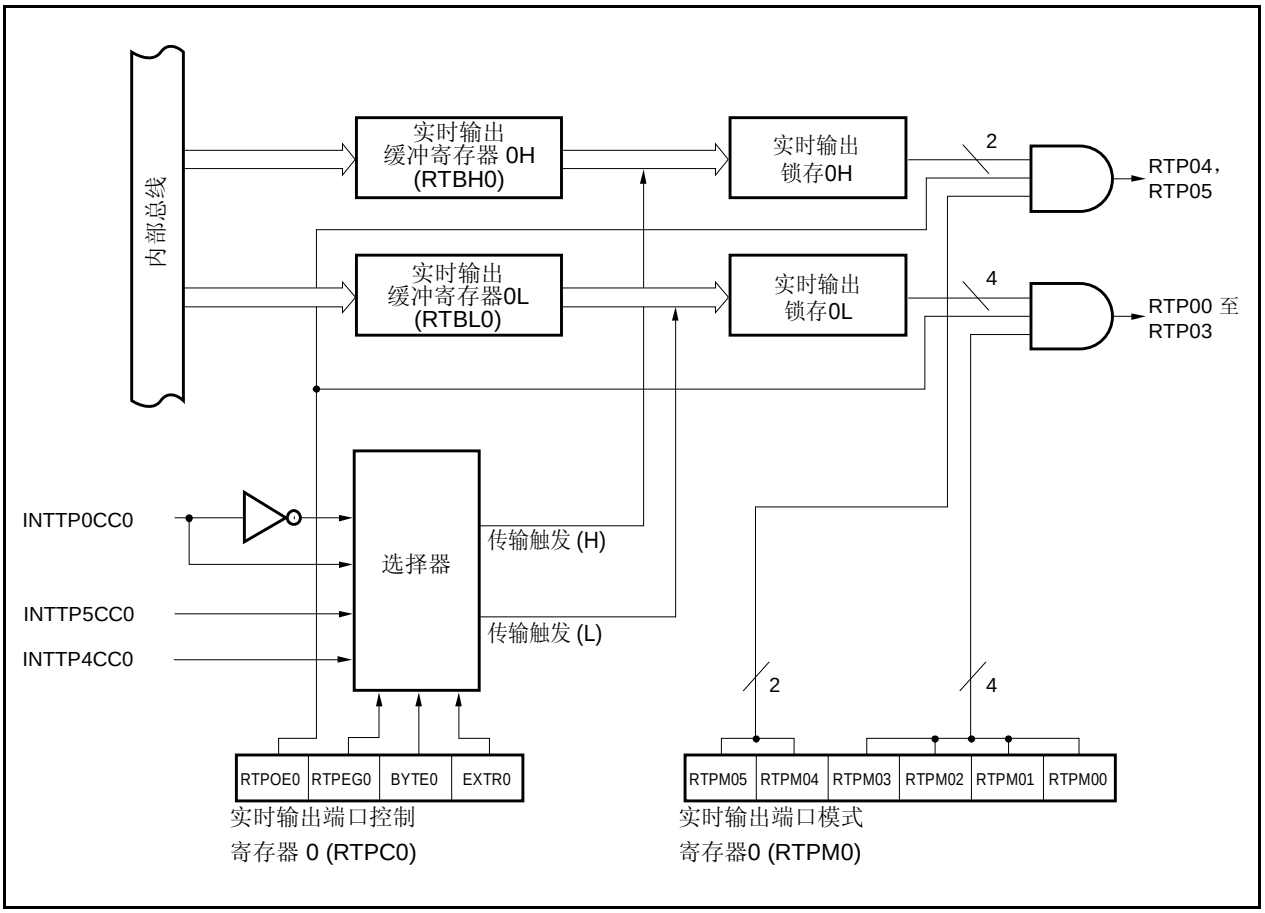
RTO 包括下列硬件：

表 13-1. RTO 的配置

项目	配置
寄存器	实时输出缓冲寄存器 0L，0H (RTBL0，RTBH0) 实时输出锁存器 0L，0H 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)

RTO 的功能框图如下所示：

图 13-1. RTO 的功能框图



(1) 实时输出缓冲寄存器 0L，0H (RTBL0，RTBH0)

RTBL0 和 RTBH0 寄存器是 4 位寄存器，用于保存预设置的输出数据。

这些寄存器被映射到外设 I/O 寄存器区域中的不同地址。

这些寄存器可以按字节或按位进行读写。

复位输入会将这些寄存器设为 00H。

如果指定 4 位× 1 通道 或 2 位× 1 通道的操作模式(RTPC0.BYTE0 位= 0)，数据可以被分别设置到 RTBL0 和 RTBH0 寄存器。通过指定这些寄存器各自的地址，可以立即读出寄存器的数据。

如果指定 6 位× 1 通道的操作模式(BYTE0 位= 1)，向其中任一个寄存器写入数据，都可以把 8 位数据设置到 RTBL0 和 RTBH0 寄存器。此外，通过指定这些寄存器各自的地址，可以立即读出寄存器的数据。

RTBL0 和 RTBH0 寄存器的操作处理过程如表 13-2 所示：

复位后: 00H　R/W　地址:　RTBL0 FFFFF6E0H, RTBH0 FFFFF6E2H

	7	6	5	4	3	2	1	0
RTBL0					RTBL03	RTBL02	RTBL01	RTBL00
RTBH0	0	0	RTBH05	RTBH04				

注意事项 1. 如果向 RTBH0 寄存器第 6 位、第 7 位写入数据，总是写入 0。

2. 在以下状态访问 RTBL0 和 RTBH0 寄存器时，将会发生等待。一旦系统进入等待状态，取消等待状态的唯一方式就是执行复位。详细说明，参见 3.4.9 (1) 访问片上外设 I/O 特定寄存器。

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

表 13-2. 使用 RTBL0 和 RTBH0 寄存器的处理操作

工作模式	处理的寄存器	读取		写注	
		高 4 位	低 4 位	高 4 位	低 4 位
4 位×1 通道, 2 位×1 通道	RTBL0	RTBH0	RTBL0	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　设置为实时输出端口后，必须在实时输出触发信号产生之前，将输出数据设置到 RTBL0 和 RTBH0 寄存器中。

13.3 寄存器

RTO 由以下两个寄存器控制：

- 实时输出端口模式寄存器 0(RTPM0)
- 实时输出端口控制寄存器 0(RTPC0)

注意事项 为了使用实时输出引脚 (RTP00 至 RTP05)，通过 PMC 和 PFC 寄存器将这些引脚设置为实时输出端口引脚。

(1) 实时输出端口模式寄存器 0(RTPM0)

RTPM0 寄存器用于按位选择实时输出端口模式或端口模式。

该寄存器可以进行字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

复位后：00H　R/W　地址：FFFFF6E4H

	7	6	5	4	3	2	1	0
RTPM0	0	0	RTPM05	RTPM04	RTPM03	RTPM02	RTPM01	RTPM00

RTPM0m	实时输出端口的控制 (m = 0 至 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 至 RTP05 引脚中被设置为实时输出的位进行实时输出，而设置为端口模式的位输出 0。**
2. **如果禁止实时输出操作(RTPOE0 位= 0)，不管 RTPM0 寄存器如何设置，实时输出引脚(RTP00 至 RTP05) 都输出 0。**
3. **如果向 RTPM0 寄存器第 6 位、第 7 位写入数据，总是写入 0。**

(2) 实时输出端口控制寄存器 0(RTPC0)

RTPC0 寄存器用于设置操作模式和实时输出端口的输出触发信号。

操作模式和实时输出端口输出触发信号的关系如表 13-3 所示。

该寄存器可以进行字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H　R/W　地址: FFFFF6E5H

	<7>	6	5	4	3	2	1	0
RTPC0	RTPOE0	RTPEG0	BYTE0	EXTR0	0	0	0	0

RTPOE0	实时输出操作的控制
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	实时输出通道配置的说明
0	4位 X 2通道，2位 X 2通道
1	6位 X 2通道

注
1. 禁止实时输出操作（RTPOE0 位 = 0）时，实时输出引脚（RTP00 至 RTP05）的所有位都输出“0”。
2. 在这些设置下，缓冲和锁存之间的数据传输将会延迟一个时钟周期。

注意事项 只有当 RTPOE0 位 = 0 时，才能设置 RTPEG0、 BYTE0 和 EXTR0 位。

表 13-3. 实时输出端口的操作模式和输出触发

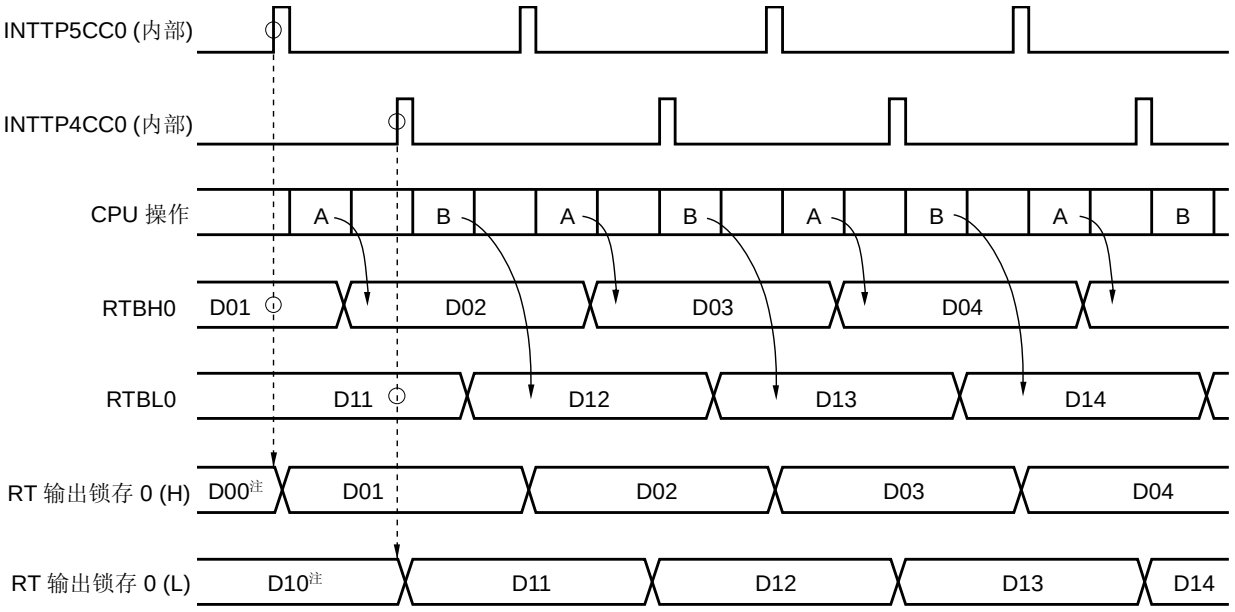
BYTE0	EXTR0	工作模式	RTBH0 (RTP04， RTP05)	RTBL0 (RTP00 至 RTP03)
0	0	4 位×1 通道， 2 位×1 通道	INTTP5CC0	INTTP4CC0
	1		INTTP4CC0	INTTP0CC0
1	0	6 位×1 通道	INTTP4CC0	
	1		INTTP0CC0	

13.4 操作

通过设置 RTPC0.RTPOE0 位为 1，使能实时输出操作，在选定的传送触发信号产生(由 RTPC0.EXTR0 和 RTPC0.BYTE0 位设置)时，RTBH0 和 RTBL0 寄存器的数据被传送到实时输出锁存器中。在已传送的数据中，只有那些在 RTPM0 寄存器中被设置为允许实时输出的位，其数据才会从 RTP00 至 RTP05 位输出。RTPM0 寄存器中设置为禁止实时输出的位输出 0。

通过将 RTPOE0 位清为 0，禁止实时输出操作，则不管 RTPM0 寄存器如何设置，RTP00 至 RTP05 引脚都输出 0。

图 13-2. RTO0 的操作时序示例（当 EXTR0= 0， BYTE0 位= 0）



注 如果在 RTPOE0 位为 0 时写入 RTBH0 和 RTBL0 寄存器，写入的值被分别传输到实时输出锁存 0H 和 0L。

A： 对 INTTP5CC0 中断请求（RTBH0 写入）的软件处理
B： 对 INTTP4CC0 中断请求（RTBL0 写入）的软件处理

备注 关于待机期间的操作，参见**第二十三章 待机功能**。

13.5 用法

(1) 禁止实时输出
将 RTPC0.RTPOE0 位清为 0。

(2) 执行下列初始化操作。
- 设置端口 5 的复用功能引脚
  将 PFC5.PFC5m 和 PFCE5.PFCE5m 位置为 1，然后将 PMC5.PMC5m 位置为 1（m = 0 至 5）。
- 按位指定实时输出端口模式或端口模式。
  设置 RTPM0 寄存器。
- 通道配置： 选择触发信号和有效沿。
  设置 RTPC0.EXTR0、 RTPC0.BYTE0 和 RTPC0.RTPEG0 位。
- 设置 RTBH0 和 RTBL0 寄存器[注1]的初始值。

(3) 实时输出使能
设置 RTPOE0 位 = 1。

(4) 在选择的传送触发信号产生[注2]时，把下一个输出值设置到 RTBH0 和 RTBL0 寄存器。

(5) 通过所选触发信号的中断服务程序，把下一个实时输出值设置到 RTBH0 和 RTBL0 寄存器。

注1. 当 RTPOE0 位为时，如果写入 RTBH0 和 RTBL0 寄存器，则写入的值被分别传送到实时输出锁存 0H 和 0L。

2. 当 RTPOE0 位为 1 时，即使写入 RTBH0 和 RTBL0 寄存器，则数据不被传送到实时输出锁存 0H 和 0L。

注意事项 要对实时输出引脚 (RTP00 至 RTP05)应用以上设置，通过 PMC 和 PFC 寄存器将这些引脚设置为实时输出引脚。

13.6 注意事项

(1) 通过软件避免以下的冲突，比如在选中的实时输出触发的中断服务程序中写入 RTBL0, RTBH0 和 RTPC0 寄存器：

- 实时输出禁止/使能切换(RTPOE0 位)和选择的实时输出输出触发信号之间的冲突。
- 在实时输出状态下对 RTBH0 和 RTBL0 寄存器的写入操作和选择的实时输出触发信号之间的冲突。

(2) 在执行初始化之前，禁止实时输出(RTPOE0 位 = 0)。

(3) 一旦实时输出被禁止（RTPOE0 位 = 0），确保重新使能实时输出（RTPOE0 位 = 0 → 1）之前，对 RTBH0 和 RTBL0 寄存器初始化。

第十四章 A/D转换器

14.1 概述

A/D 转换器可以将模拟输入信号转换为数字值，具有 10 位分辨率，具有 12 路模拟输入信号通道（ANI0 至 ANI11）。A/D 转换器具有下列特征：

- 10 位分辨率。
- 12 通道
- 逐次逼近法
- 工作电压： AV_{REF0} = 2.7 至 3.6 V 。
- 模拟输入电压： 0 V 至 AV_{REF0}
- 操作模式具有下列功能：
  - 连续选择模式
  - 连续扫描模式
  - 单次选择模式
  - 单次扫描模式
- 触发模式具有下列功能：
  - 软件触发模式
  - 外部触发模式（外部，1）
  - 定时器触发模式
- 转换时间
  2.6 至 24 *us*@3.0 V ≤ AV_{REF0} ≤ 3.6 V
  3.9 至 24 *us*@2.7 V ≤ AV_{REF0} < 3.0 V
- 掉电监控功能（转换结果比较功能）

14.2 功能

（1）10 位分辨率的 A/D 转换

A/D 转换以 10 位分辨率反复操作，对 ANI0 至 ANI11 通道中选择的模拟输入通道转换。每次 A/D 转换完成后，都会产生一个中断请求信号（INTAD）。

（2）掉电检测功能

此功能用来检测电池电压的下降。将 A/D 转换结果（ADA0CRnH 寄存器的值）和 ADA0PFT 寄存器值进行比较，仅当 ADA0PFM 寄存器指定的比较条件满足（n = 0 至 11）时，才会产生 INTAD 信号。

14.3 配置

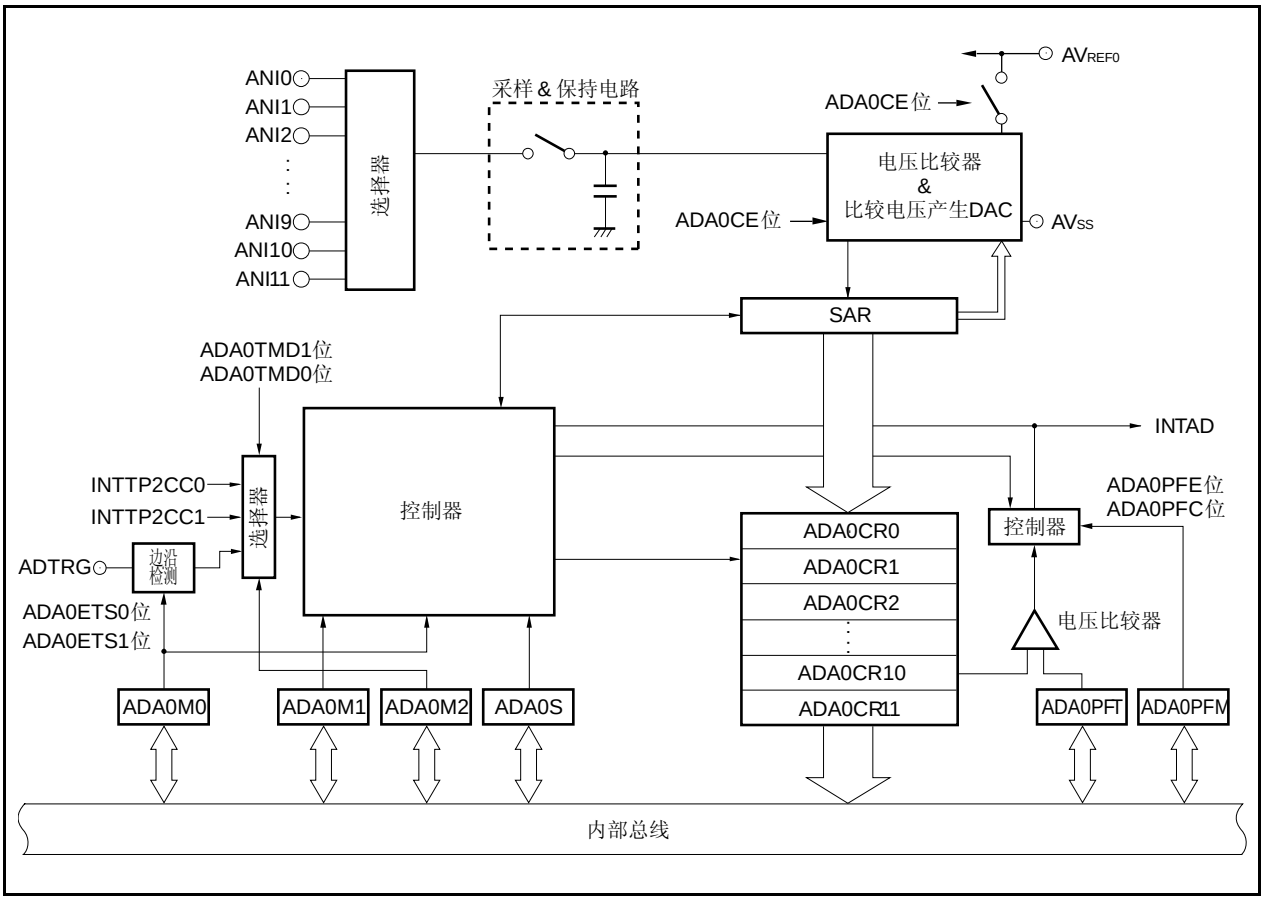
A/D 转换器包括下列硬件：

表 14-1. A/D 转换器的配置

项目	配置
模拟输入	12 个通道（ANI0 引脚至 ANI11 引脚）
寄存器	逐次逼近寄存器（SAR） A/D 转换结果寄存器 0 至 11（ADA0CR0 至 ADA0CR11） A/D 转换结果寄存器 0H 至 11H（ADCR0H 至 ADCR11H）：仅能读取高 8 位 A/D 转换器模式寄存器 0 至 2（ADA0M0 至 ADA0M2） A/D 转换器通道指定寄存器 0（ADA0S） 掉电比较模式寄存器（ADA0PFM） 掉电比较阈值寄存器（ADA0PFT）

A/D 转换器的框图如下所示：

图 14-1. A/D 转换器框图



（1）ANI0 至 ANI11 引脚

这些引脚是 12 路 A/D 转换通道的模拟输入引脚，用于输入将要转换为数字信号的模拟信号。除了 ADA0S 寄存器选定的模拟输入引脚外，其他各个引脚都可以用作输入端口引脚。

注意事项 确保输入 ANI0 引脚至 ANI11 引脚的电压没有超过额定值。如果输入至某通道的电压值为 AV_{REF0} 电压或者更高，那么此通道的转换值不确定，并且其它通道的转换值也会受到影响。

（2）采样&保持电路

采样保持电路对输入电路选择的各个模拟输入信号进行采集，并将采样数据发送到电压比较器。在 A/D 转换期间，该电路保持采样得到的模拟输入信号电压。

（3）比较电压发生 DAC

比较电压发生 DAC 连接在 AV_{REF0} 和 AV_{SS} 之间，并产生用于和模拟输入信号进行比较的电压。

（4）电压比较器

电压比较器将采样保持电路得到的电压值与比较电压产生 DAC 的输出电压进行比较。

（5）渐次逼近寄存器（SAR）

SAR 寄存器将模拟输入信号的电压值与比较电压产生 DAC（比较电压）的输出值进行比较，并且从最高有效位（MSB）开始保存比较结果。

当比较结果已经保存至最低有效位（LSB）（也就是 A/D 转换完成）时，SAR 寄存器的内容将被传送到 ADA0CRn 寄存器。

备注 n = 0 至 11

（6）10 位 AD 转换结果寄存器 n（ADA0CRn）

ADA0CRn 寄存器是一个存储 A/D 转换结果的 16 位寄存器。ADA0ARn 由 12 个寄存器组成，并且 A/D 转换结果存储在对应于模拟输入的 AD0CRn 寄存器高 10 位（低 6 位固定为 0）。

（7）A/D 转换结果寄存器 nH（ADA0CRnH）

ADA0CRnH 寄存器是一个存储 A/D 转换结果的 8 位寄存器。ADA0CRnH 由 12 个寄存器组成，并且 A/D 转换结果存储在对应于模拟输入的 ADA0CRnH 寄存器高 8 位。

（8）A/D 转换器模式寄存器 0（ADA0M0）

该寄存器用于指定操作模式并且控制 A/D 转换器的转换操作。

（9）A/D 转换器模式寄存器 1（ADA0M1）

该寄存器用于设置待转换的模拟输入信号的转换时间。

（10）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。

（11）A/D 转换器通道指定寄存器（ADA0S）

该寄存器用于设置输入端口，该端口用于输入待转换的模拟电压。

（12）掉电比较模式寄存器（ADA0PFM）

该寄存器用于设置掉电监控模式。

（13）掉电比较阈值寄存器（ADA0PFT）

ADA0PFT 寄存器用于设置阈值，和 A/D 转换结果寄存器 nH（ADA0CRnH）的值进行比较。ADA0PFT 寄存器的 8 位数据与 A/D 转换结果寄存器（ADA0CRnH）的高 8 位进行比较。

（14）控制器

当 A/D 转换完成时或使用掉电检测功能时，A/D 控制器将转换结果（ADA0CRnH 寄存器的值）与 ADA0PFT 寄存器的值进行比较，当满足指定的比较条件时，产生 INTAD 信号。

（15）AV_{REF0} 引脚

该引脚用于输入 A/D 转换器的参考电压。即使不使用 A/D 转换器，也要让该引脚的电压与 V_{DD} 引脚的电压相同。根据施加在 AV_{REF0} 引脚和 AV_{SS} 引脚之间的电压，将输入到 ANI0 至 ANI11 引脚的信号转化为数字信号。

（16）AV_{SS} 引脚

该引脚为 A/D 转换器的地引脚。即使不使用 A/D 转换器，也要保证该引脚的电压与 V_{SS} 引脚的电压相同。

14.4 寄存器

A/D 转换器由下列寄存器控制：

- A/D 转换器模式寄存器 0、1、2（ADA0M0、ADA0M1、ADA0M2）
- A/D 转换器通道指定寄存器 0（ADA0S）
- 掉电比较模式寄存器（ADA0PFM）

以下寄存器也可以使用：

- A/D 转换结果寄存器 n（ADA0CRn）
- A/D 转换结果寄存器 nH（ADA0CRnH）
- 掉电比较阈值寄存器（ADA0PFT）

（1）A/D 转换器模式寄存器 0（ADA0M0）

ADA0M0 寄存器是 8 位寄存器，用于指定操作模式并控制转换操作。
该寄存器可进行字节读写或按位读写。但是，ADA0EF 0 是只读位。
系统复位后，该寄存器被设为 00H。

注意事项 下列状态下禁止访问 ADA0M0 寄存器。如果产生等待周期，只能通过复位来清除。详细说明，参见 3.4.9（1）访问片上外设 I/O 专用寄存器。
• 当 CPU 运行于副时钟，且主时钟振荡停止时。
• 当 CPU 运行于内部振荡时钟工作时。

（1/2）

复位后：00H　R/W　地址：FFFFF200H

	<7>	6	5	4	3	2	1	<0>
ADA0M0	ADA0CE	0	ADA0MD1	ADA0MD0	ADA0ETS1	ADA0ETS0	ADA0TMD	ADA0EF

ADA0CE	A/D 转换控制
0	停止 A/D 转换
1	使能 A/D 转换

ADA0MD1	ADA0MD0	A/D转换器操作模式说明
0	0	连续选择模式
0	1	连续扫描模式
1	0	单次选择模式
1	1	单次扫描模式

（2/2）

ADA0ETS1	ADA0ETS0	外部触发(ADTRG 引脚) 输入有效边沿的说明
0	0	无边沿检测
0	1	下降沿检测
1	0	上升沿检测
1	1	双边沿检测

ADA0TMD	触发模式说明
0	软件触发模式
1	外部触发模式/定时器触发模式

ADA0EF	A/D 转换器状态说明
0	A/D 转换停止
1	A/D 转换进行中

注意事项

1. **对第 0 位的写入操作将被忽略。**
2. **A/D 转换使能（ADA0CE 位 = 1）时，禁止改变 ADA0M1.ADA0FR2 位至 ADA0M1.ADA0FR0 位。**
3. **下列模式中，当 A/D 转换停止（ADA0CE 位 = 0）时，向 ADA0M0、ADA0M2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器写入数据，然后再使能 A/D 转换器的操作（ADA0CE 位 = 1）**
   - **正常转换模式下**
   - **高速转换模式下的单次选择模式/单次扫描模式**

   A/D 转换期间（ADA0EF 位= 1），如果在其它模式下写入 ADA0M0、ADA0M2、ADA0S、ADA0PFM 和 ADA0PFT 寄存器，那么根据模式将会执行如下操作。
   - **软件触发模式下：**
     A/D 转换停止，并再次从头开始执行。
   - **硬件触发模式下：**
     A/D 转换停止，且设置为触发等待状态。
4. **为了选择外部触发模式/定时器触发模式（ADA0TMD 位= 1），设置为高速转换模式（ADA0M1.ADA0HS1 位 = 1）。A/D 转换操作使能（ADA0CE 位= 1）后，在随后插入的振荡稳定时间期间，不要输入触发信号。**
5. **当不使用 A/D 转换器时，将 ADA0CE 位设置为 0，停止 A/D 转换，可以降低功耗。**

（2）A/D 转换器模式寄存器 1（ADA0M1）

ADA0M1 是 8 位寄存器，用于指定转换时间。

该寄存器可以按字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H　R/W　地址: FFFFF201H

	7	6	5	4	3	2	1	0
ADA0M1	ADA0HS1	0	0	0	0	ADA0FR2	ADA0FR1	ADA0FR0

ADA0HS1	正常转换模式/高速模式说明(A/D转换时间)
0	正常转换模式
1	高速转换模式

注意事项 1. A/D 转换使能（ADA0M0.ADA0CE 位= 1）时，禁止改变 ADA0M1 寄存器。

2. 当选择外部触发模式/定时器触发模式（ADA0M0.ADA0TMD 位= 1）时，设置高速转换模式（ADA0HS1 位 = 1）。A/D 转换操作使能（ADA0CE 位= 1）后，在随后插入的振荡稳定时间期间，不要输入触发信号。

3. 请确保将第 6 位至第 3 位清为“0”。

备注 关于设置 A/D 转换时间的举例，参见**表 14-2** 和**表 14-3。**

各种时钟下的转换时间示例如下。

表 14-2. 正常转换模式下转换时间的选择（ADA0HS1 位= 0）

ADA0 FR2	ADA0 FR1	ADA0 FR0	A/D 转换时间						
			稳定时间 ＋转换时间＋等待时间	f_{XX} = 20 MHz	f_{XX} = 16 MHz	f_{XX} = 12 MHz	f_{XX} = 10 MHz	f_{XX} = 4 MHz	触发响应时间
0	0	0	66/f_{XX}（13/f_{XX} + 26/f_{XX} + 27/f_{XX}）	禁止设置	禁止设置	禁止设置	6.6 *us* 注	16.5 *us*	3/f_{XX}
0	0	1	131/f_{XX}（26/f_{XX} + 52/f_{XX} + 53/f_{XX}）	6.55 *us* 注	8.19 *us* 注	10.92 *us*	13.1 *us*	禁止设置	3/f_{XX}
0	1	0	196/f_{XX}（39/f_{XX} + 78/f_{XX} + 79/f_{XX}）	9.8 *us*	12.25 *us*	16.33 *us*	19.6 *us*	禁止设置	3/f_{XX}
0	1	1	259/f_{XX}（50/f_{XX} + 104/f_{XX} + 105/f_{XX}）	12.95 *us*	16.19 *us*	21.58 *us*	禁止设置	禁止设置	3/f_{XX}
1	0	0	311/f_{XX}（50/f_{XX} + 130/f_{XX} + 131/f_{XX}）	15.55 *us*	19.44 *us*	禁止设置	禁止设置	禁止设置	3/f_{XX}
1	0	1	363/f_{XX}（50/f_{XX} + 156/f_{XX} + 157/f_{XX}）	18.15 *us*	22.69 *us*	禁止设置	禁止设置	禁止设置	3/f_{XX}
1	1	0	415/f_{XX}（50/f_{XX} + 182/f_{XX} + 183/f_{XX}）	20.75 *us*	禁止设置	禁止设置	禁止设置	禁止设置	3/f_{XX}
1	1	1	467/f_{XX}（50/f_{XX} + 208/f_{XX} + 209/f_{XX}）	23.35 *us*	禁止设置	禁止设置	禁止设置	禁止设置	3/f_{XX}
其它			禁止设置						

注 当 2.7 V ≤ AV_{REF0} < 3.0 V 时，禁止设置。

备注 **1.** 稳定时间： A/D 转换建立时间（1 *us* 或更长）
转换时间： 实际的 A/D 转换时间（2.6 至 10.4 *us*）
等待时间： 下一次转换之前，插入的等待时间
触发响应时间： 如果在稳定时间之后产生软件触发、外部触发或者定时器触发，那么在转换时间之前插入触发响应时间。

2. 操作时序的详情，参见 **14.5.2 转换操作时序**。

在正常转换模式下，在 ADA0M0.ADA0CE 位被置为 1 之后，经过稳定时间，开始转换，并且 A/D 转换仅在指定的转换时间（2.6 至 10.4 us）内执行。转换结束后转换操作停止，经过等待时间，产生 A/D 转换结束中断请求信号（INTAD）。
因为转换操作在等待时间内被停止，所以降低了工作电流。

注意事项 1. 当 3.0 V ≤ AV_{REF0} ≤ 3.6 V 时，设置为 2.6 *us*≤转换时间 ≤ 10.4 *us*
当 2.7 V ≤ AV_{REF0} < 3.0 V 时，设置为 3.9 *us*≤ 转换时间 ≤ 10.4 *us*

2. A/D 转换期间，如果有数据写入 ADA0M0、ADA0M2、ADA0S、ADA0PFM 以及 ADA0PFT 寄存器或者有触发信号输入，则重新执行转换。但是，如果稳定时间结束时序与写入这些寄存器的操作发生冲突，或者稳定时间结束时序与触发信号输入发生冲突，那么将会重新插入 64 个时钟的稳定时间。
如果在插入的稳定时间结束时序时刻再次发生冲突，则仍然重新插入稳定时间。因此，不要将触发输入间隔和控制寄存器写入间隔设置为 64 个时钟或更少。

表 14-3. 高速转换模式下转换时间的选择（ADA0HS1 位= 1）

ADA0 FR2	ADA0 FR1	ADA0 FR0	A/D 转换时间						
			转换时间 (+ 稳定时间)	f_{XX} = 20 MHz	f_{XX} = 16 MHz	f_{XX} = 12 MHz	f_{XX} = 10 MHz	f_{XX} = 4 MHz	触发响应时间
0	0	0	26/f_{XX}(+ 13/f_{XX})	禁止设置	禁止设置	禁止设置	2.6 *us* 注 (+ 1.3 *us*)	6.5 *us* (+ 3.25 *us*)	3/f_{XX}
0	0	1	52/f_{XX}(+ 26/f_{XX})	2.6 *us* 注 (+ 1.3 *us*)	3.25 *us* 注 (+ 1.625 *us*)	4.333 *us* (+ 2.167 *us*)	5.2 *us* (+ 2.6 *us*)	禁止设置	3/f_{XX}
0	1	0	78/f_{XX}(+ 39/f_{XX})	3.9 *us* (+ 1.95 *us*)	4.875 *us* (+ 2.438 *us*)	6.5 *us* (+ 3.25 *us*)	7.8 *us* (+ 3.9 *us*)	禁止设置	3/f_{XX}
0	1	1	104/f_{XX}(+ 50/f_{XX})	5.2 *us* (+ 2.5 *us*)	6.5 *us* (+ 3.125 *us*)	8.667 *us* (+ 4.167 *us*)	10.4 *us* (+ 5 *us*)	禁止设置	3/f_{XX}
1	0	0	130/f_{XX}(+ 50/f_{XX})	6.5 *us* (+ 2.5 *us*)	8.125 *us* (+ 3.125 *us*)	禁止设置	禁止设置	禁止设置	3/f_{XX}
1	0	1	156/f_{XX}(+ 50/f_{XX})	7.8 *us* (+ 2.5 *us*)	9.75 *us* (+ 3.125 *us*)	禁止设置	禁止设置	禁止设置	3/f_{XX}
1	1	0	182/f_{XX}(+ 50/f_{XX})	9.1 *us* (+ 2.5 *us*)	禁止设置	禁止设置	禁止设置	禁止设置	3/f_{XX}
1	1	1	208/f_{XX}(+ 50/f_{XX})	10.4 *us* (+ 2.5 *us*)	禁止设置	禁止设置	禁止设置	禁止设置	3/f_{XX}
其它			禁止设置						

注 当 2.7 V ≤ AV_{REF0} < 3.0 V 时，禁止设置。

备注 **1.** 转换时间： 实际的 A/D 转换时间（2.6 至 10.4 *us*）
稳定时间： A/D 转换建立时间（1 *us* 或更长）
触发响应时间： 如果在稳定时间之后产生软件触发、外部触发或者定时器触发，那么在转换时间之前插入触发响应时间。

2. 操作时序的详情，参见 **14.5.2 转换操作时序**。

在高速转换模式下，在 ADA0M0.ADA0CE 位被置为 1 之后，经过稳定时间，开始转换，并且 A/D 转换仅在指定的转换时间（2.6 至 10.4 us）内执行。转换结束后立即产生 A/D 转换结束中断请求信号（INTAD）。
在连续转换模式下，仅在第一次转换前插入稳定时间，而在第二次转换之后（A/D 转换器继续运行）不会插入。

注 当 2.7 V ≤ AV_{REF0} < 3.0 V 时，禁止设置。

注意事项 1. 当 3.0 V ≤AV_{REF0} ≤ 3.6 V 时，设置为 2.6 *us* ≤转换时间≤ 10.4 *us*
当 2.7 V≤AV_{REF0} < 3.0 V 时，设置为 3.9 *us* ≤ 转换时间 ≤ 10.4 *us*

2. 在高速转换模式下，振荡稳定期间，禁止触发信号的输入，对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 和 ADA0PFT 寄存器的重写也被禁止。

（3）A/D 转换器模式寄存器 2（ADA0M2）

ADA0M2 寄存器用于指定硬件触发模式。

该寄存器可以按字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H　　R/W　　地址: FFFFF203H

	7	6	5	4	3	2	1	0
ADA0M2	0	0	0	0	0	0	ADA0TMD1	ADA0TMD0

ADA0TMD1	ADA0TMD0	硬件触发模式说明
0	0	外部触发模式（检测到ADTRG引脚有效边沿时）
0	1	定时器触发模式 0 (产生INTTP2CC0 中断请求时)
1	0	定时器触发模式 1 (产生INTTP2CC1中断请求时)
1	1	禁止设置

注意事项 1. 下列模式时，当 A/D 转换停止（ADA0M0.ADA0CE 位 = 0）时，向 ADA0M2 寄存器写入数据，然后使能 A/D 转换器的操作（ADA0CE 位 = 1）

- **正常转换模式**
- **高速转换模式下的单次选择模式/单次扫描模式**

2. 请确保将第 7 位至第 2 位清为“0”。

（4）模拟输入通道指定寄存器 0（ADA0S）

ADA0S 寄存器用于指定将模拟电压转换为数字信号的输入引脚。

该寄存器可以按字节或按位进行读写。

系统复位后，该寄存器被设为 00H。

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 至 ANI2
0	0	1	1	ANI3	ANI0 至 ANI3
0	1	0	0	ANI4	ANI0 至 ANI4
0	1	0	1	ANI5	ANI0 至 ANI5
0	1	1	0	ANI6	ANI0 至 ANI6
0	1	1	1	ANI7	ANI0 至 ANI7
1	0	0	0	ANI8	ANI0 至 ANI8
1	0	0	1	ANI9	ANI0 至 ANI9
1	0	1	0	ANI10	ANI0 至 ANI10
1	0	1	1	ANI11	ANI0 至 ANI11
1	1	0	0	禁止设置	禁止设置
1	1	0	1	禁止设置	禁止设置
1	1	1	0	禁止设置	禁止设置
1	1	1	1	禁止设置	禁止设置

注意事项 1. 下列模式中，当 A/D 转换停止（ADA0M0.ADA0CE 位 = 0）时，向 ADA0S 寄存器写入数据，然后使能 A/D 转换器的操作（ADA0CE 位 = 1）。

- **正常转换模式**
- **高速转换模式下的单次选择模式/单次扫描模式**

2. 请确保将第 7 位至第 4 位清为“0”。

（5）A/D 转换结果寄存器 n、nH（ADA0CRn，ADA0CRnH）

ADA0CRn 寄存器和 ADA0CRnH 寄存器用于存储 A/D 转换结果。

这些寄存器是以 8 位或 16 位宽度访问的只读寄存器。但是，指定 ADA0CRn 寄存器为 16 位访问，指定 ADA0CRnH 寄存器为 8 位访问。对于 ADA0CRn 寄存器来说，从高 10 位读取 10 位转换结果，低 6 位的读取值为 0。对于 ADA0CRnH 寄存器来说，读取高 8 位转换结果。

注意事项 下列状态下，禁止访问 ADA0CRn 寄存器和 ADA0CRnH 寄存器。如果产生等待周期，只能通过复位来清除。详细说明，参见 3.4.9（1）访问片上外设 I/O 专用寄存器。

●当 CPU 运行于副时钟，且主时钟振荡停止时。

●当 CPU 运行于内部振荡时钟工作时。

复位后：不确定　　R　　地址：ADA0CR0 FFFFF210H, ADA0CR1 FFFFF212H, ADA0CR2 FFFFF214H, ADA0CR3 FFFFF216H, ADA0CR4 FFFFF218H, ADA0CR5 FFFFF21AH, ADA0CR6 FFFFF21CH, ADA0CR7 FFFFF21EH, ADA0CR8 FFFFF220H, ADA0CR9 FFFFF222H, ADA0CR10 FFFFF224H, ADA0CR11 FFFFF226H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADA0CRn (n = 0 至 11)	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

复位后：不确定　　R　　地址：ADA0CR0H FFFFF211H, ADA0CR1H FFFFF213H, ADA0CR2H FFFFF215H, ADA0CR3H FFFFF217H, ADA0CR4H FFFFF219H, ADA0CR5H FFFFF21BH, ADA0CR6H FFFFF21DH, ADA0CR7H FFFFF21FH, ADA0CR8H FFFFF221H, ADA0CR9H FFFFF223H, ADA0CR10H FFFFF225H, ADA0CR11H FFFFF227H

	7	6	5	4	3	2	1	0
ADA0CRnH (n = 0 至 11)	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

注意事项 对 ADA0M0 寄存器和 ADA0S 寄存器的写入操作可能会导致 ADA0CRn 寄存器的内容不确定。转换完成后，在写入 ADA0M0 寄存器和 ADA0S 寄存器之前读取转换结果。如果没有按上述顺序执行，则读取结果可能会有误。

输入到模拟输入引脚（ANI0 至 ANI11）的模拟电压与转换结果（ADA0CRn 寄存器）之间的关系如下：

$$\mathbf{SAR = INT\left(\frac{V_{IN}}{AV_{REF0}} \times 1{,}024 + 0.5\right)}$$

$$\mathbf{ADA0CR^{注} = SAR \times 64}$$

或，

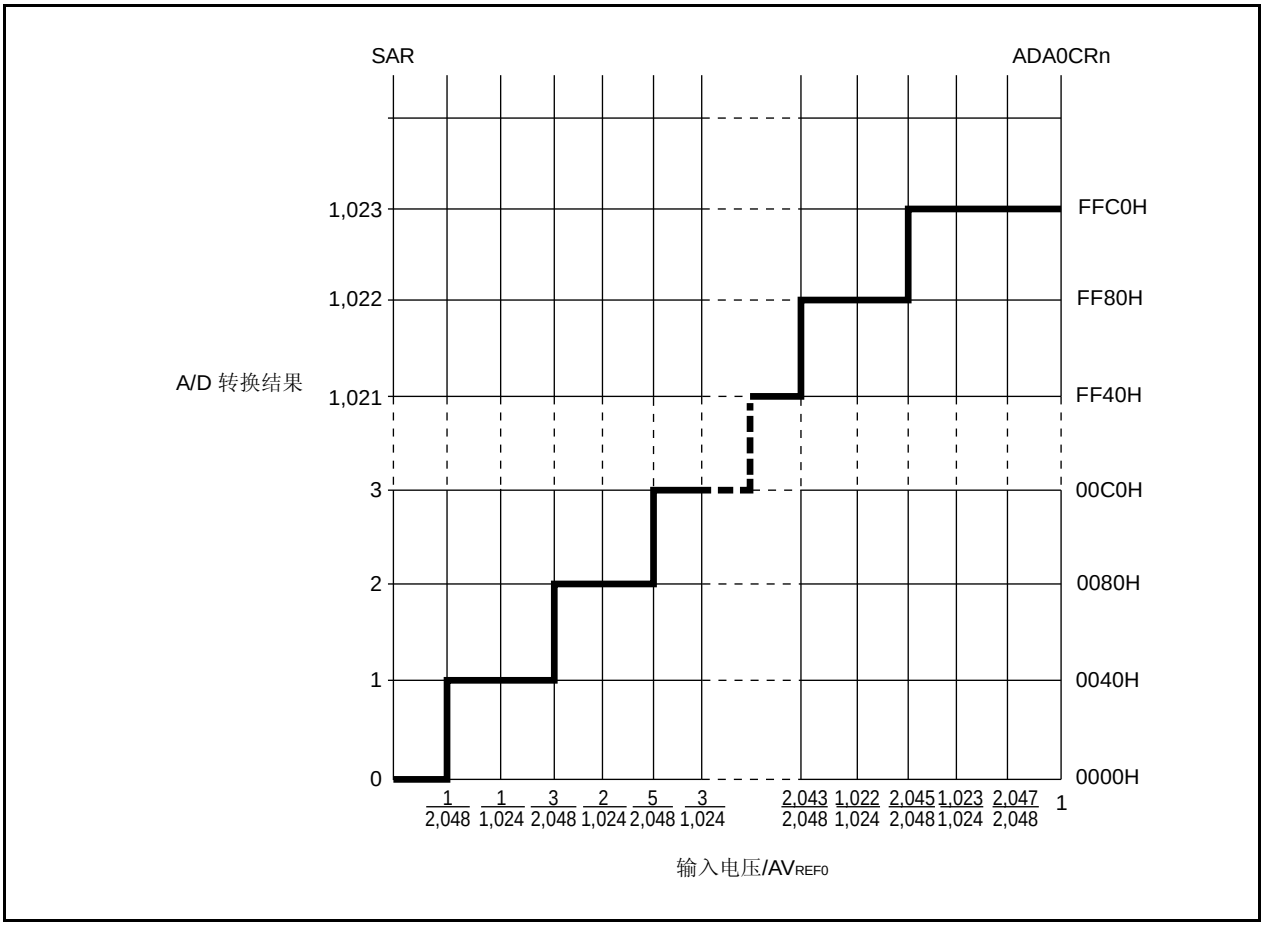
$$\mathbf{(SAR - 0.5) \times \frac{AV_{REF0}}{1{,}024} \le V_{IN} < (SAR + 0.5) \times \frac{AV_{REF0}}{1{,}024}}$$

INT（ ）： 返回括号内数值整数部分的函数
V_{IN}： 模拟输入电压
AV_{REF0}： AV_{REF0} 引脚电压
ADA0CR： ADA0CRn 寄存器的值

注 ADA0CRn 寄存器的低 6 位固定为 0。

下图显示了模拟输入电压和 A/D 转换结果的之间关系。

图 14-2. 模拟输入电压和 A/D 转换结果的关系



（6）掉电比较模式寄存器（ADA0PFM）

ADA0PFM 寄存器是 8 位寄存器，用于设置掉电比较模式。

该寄存器可按字节或按位进行读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H　R/W　地址: FFFFF204H

	<7>	6	5	4	3	2	1	0
ADA0PFM	ADA0PFE	ADA0PFC	0	0	0	0	0	0

ADA0PFE	掉电比较使能/禁止的选择
0	掉电比较禁止
1	掉电比较使能

ADA0PFC	掉电比较模式的选择
0	当ADA0CRnH≥ ADA0PFT时，产生中断请求信号 (INTAD)
1	当ADA0CRnH < ADA0PFT时，产生中断请求信号 (INTAD)

注意事项 1. 在选择模式下，将设置到 ADA0PFT 寄存器的 8 位数据与 ADA0S 寄存器指定通道的转换结果进行比较。如果该结果符合 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CRn 寄存器，并且产生 INTAD 信号。但是，如果不符合条件，将转换结果存储在 ADA0CR0 寄存器，且不产生 INTAD 信号。

2. 在扫描模式下，将设置到 ADA0PFT 寄存器的 8 位数据与通道 0 的转换结果进行比较。如果该结果符合 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CR0 寄存器中，并且产生 INTAD 信号。但是，如果不符合条件，将转换结果存储在 ADA0CR0 寄存器，且不产生 INTAD 信号。无论比较结果如何，扫描操作完成之前，都会继续扫描，并且将转换结果继续存储在 ADA0CRn 寄存器中，直到扫描模式结束。但是，扫描操作完成后不产生 INTAD 信号。

3. 下列模式时，当 A/D 转换停止（ADA0M0.ADA0CE 位 = 0）时，向 ADA0PFM 寄存器写入数据，然后使能 A/D 转换操作（ADA0CE 位 = 1）。

- **正常转换模式**
- **高速转换模式下的单次选择模式/单次扫描模式**

（7）掉电比较阈值寄存器（ADA0PFT）

ADA0PFT 寄存器用于设置掉电比较模式下的比较值。

该寄存器可以按字节或按位进行读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H　　R/W　　地址: FFFFF205H

	7	6	5	4	3	2	1	0
ADA0PFT								

注意事项　下列模式时，当 A/D 转换停止（ADA0M0.ADA0CE 位 = 0）时，向 ADA0PFT 寄存器写入数据，然后使能 A/D 转换操作（ADA0CE 位 = 1）。

- **正常转换模式**
- **高速转换模式下的单次选择模式/单次扫描模式**

14.5 操作

14.5.1 基本操作

<1> 使用 ADA0M0、ADA0M1、ADA0M2 和 ADA0S 寄存器来设置执行 A/D 转换的操作模式、触发模式以及转换时间。当置位 ADA0M0 寄存器的 ADA0CE 位时，在软件触发模式下转换开始，在外部或定时器触发模式下 A/D 转换器等待触发信号。

<2> A/D 转换开始后，采样&保持电路对所选模拟输入通道的输入电压进行采样。

<3> 一旦采样&保持电路对输入通道进行一段指定时间的采样，然后采样&保持电路进入保持状态，并且保持采样得到的输入模拟电压，直到 A/D 转换完成。

<4> 对逐次逼近寄存器（SAR）的第 9 位置位。档位选择器将（1/2）AV_{REF0} 选择作为比较电压产生 DAC。

<5> 电压比较器对比较电压产生 DAC 的电压和模拟输入电压之间的电压差进行比较。如果模拟输入电压高于（1/2）AV_{REF0}，SAR 寄存器的 MSB 保持置位。如果模拟输入电压低于（1/2）AV_{REF0}，MSB 位清零。

<6> 接着，自动将逐次逼近寄存器（SAR）的第 8 位置位，并启动下一次比较。根据先前所设定的位 9 的值，比较电压产生 DAC 的档位电压选择如下：
- 第 9 位 = 1：（3/4） AV_{REF0}
- 第 9 位 = 0：（1/4） AV_{REF0}

该电压和模拟输入电压比较，并且根据比较结果，对第 8 位的操作如下：
模拟输入电压 ≥ 比较电压：第 8 位 = 1
模拟输入电压 ≤ 比较电压：第 8 位 = 0

<7> 此项比较一直持续，直到 SAR 寄存器的第 0 位。

<8> 10 位比较完成时，有效数字结果存储在 SAR 寄存器，然后再传送并存储在 ADA0CRn 寄存器。之后，在下列时序处产生 A/D 转换结束中断请求信号（INTAD）。

- 连续/单次选择模式： 在第一次 A/D 转换完成之后
- 连续/单次扫描模式： 直到 ADA0S 寄存器指定的模拟输入引脚的 A/D 转换顺序执行之后

<9> 单次选择模式下，转换停止[注]。单次扫描模式下，扫描完成之后，立刻停止转换[注]。连续选择模式下，重复步骤<2>至步骤<8>直到 ADA0M0.ADA0CE 位清为 0。连续扫描模式下，每个通道都重复步骤<2>至步骤<8>。

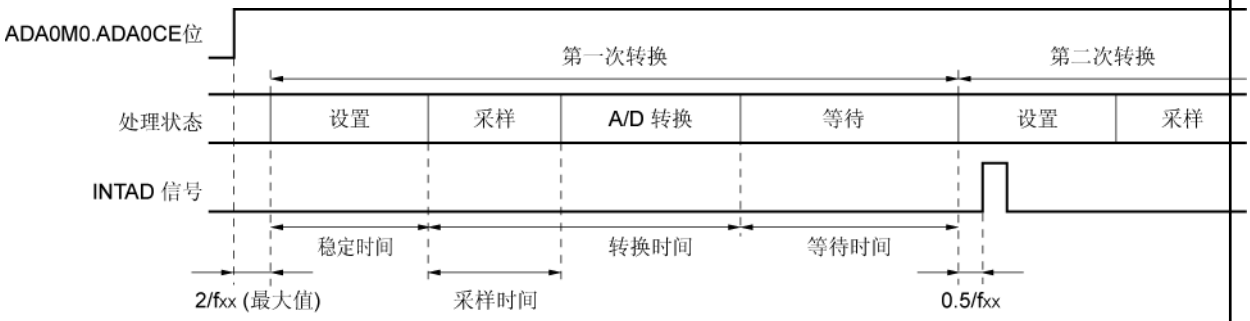
注 在外部触发模式时、定时器触发模式 0 时或定时器触发模式 1 时，进入触发等待状态。

备注 触发等待状态指稳定时间过后的状态。

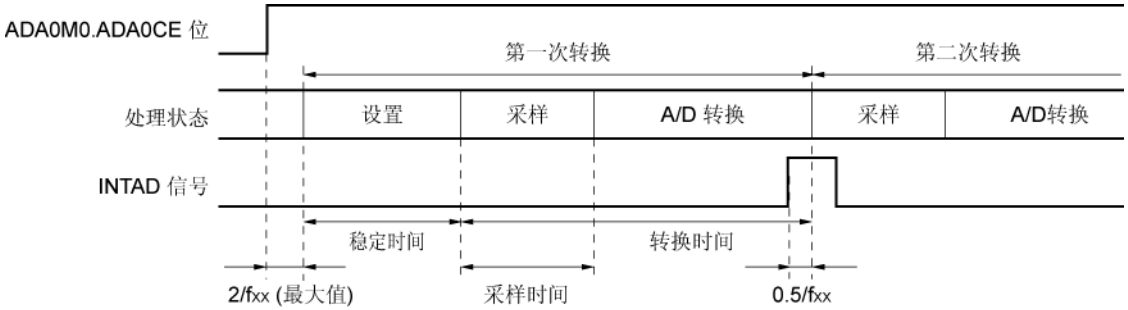
14.5.2 转换操作时序

图 14-3. 转换操作时序（连续转换）

（1） 正常转换模式下的操作（ADA0HS1 位 = 0）



（2） 高速转换模式下的操作（ADA0HS1 位 = 1）



ADA0FR2 位至 ADA0FR0 位	稳定时间	转换时间（采样时间）	等待时间	触发响应时间
000	13/fxx	26/fxx（8/fxx）	27/fxx	3/fxx
001	26/fxx	52/fxx（16/fxx）	53/fxx	3/fxx
010	39/fxx	78/fxx（24/fxx）	79/fxx	3/fxx
011	50/fxx	104/fxx（32/fxx）	105/fxx	3/fxx
100	50/fxx	130/fxx（40/fxx）	131/fxx	3/fxx
101	50/fxx	156/fxx（48/fxx）	157/fxx	3/fxx
110	50/fxx	182/fxx（56/fxx）	183/fxx	3/fxx
111	50/fxx	208/fxx（64/fxx）	209/fxx	3/fxx

备注 以上时序适用于软件触发模式，在外部触发模式/定时器触发模式下，会插入触发响应时间。

14.5.3 触发模式

通过设置触发模式指定转换操作的启动时序。触发模式包括软件触发模式和硬件触发模式。硬件触发模式包含定时器触发模式 0 和 1，以及外部触发模式。ADA0M0.ADA0TMD 位用于设置触发模式。通过 ADA0M2.ADA0TMD1 位和 ADA0M2.ADA0TMD0 位设置硬件触发模式。

表 14-4. 触发模式

ADA0M0 寄存器	ADA0M2 寄存器		触发模式
ADA0TMD 位	ADA0TMD1 位	ADA0TMD0 位	
0	–	–	软件触发模式
1	0	0	外部触发模式 (基于 ADTRG 引脚有效沿检测)
	0	1	定时器触发模式 0 (基于 INTTP2CC0 中断请求的发生)
	1	0	定时器触发模式 1 (基于 INTTP2CC1 中断请求的发生)
	1	1	禁止设置

（1）软件触发模式

当 ADA0M0.ADA0CE 位置 1 时，对 ADA0S 寄存器指定的模拟输入引脚（ANIn 引脚）的信号进行转换。当转换完成时，转换结果存储在 ADA0CRn 寄存器。同时，产生 A/D 转换结束中断信号（INTAD）。

如果通过 ADA0M0.ADA0MD1 位和 ADA0M0.ADA0MD0 位指定操作模式为连续选择/扫描模式，那么第一次转换完成后，将启动下一次转换，直到 ADA0CE 位被清 0。如果操作模式是单次选择/扫描模式，那么转换执行一次后结束。

转换开始时，ADA0M0.ADA0EF 位设为 1（表示转换正在进行）。

如果在转换期间对 ADA0M0、ADA、ADA0S、ADA0PFM 或 ADA0PFT 寄存器执行写入操作，那么此次转换中止，并再次从头开始转换。但是，在正常转换模式和高速转换模式下的单次选择模式/单次扫描模式中，禁止写入这些寄存器（n = 0 至 11）。

（2）外部触发模式

该模式下，当外部触发信号输入（至 ADTRG 引脚）时，对 ADA0S 寄存器指定的模拟输入引脚（ANIn 引脚）的信号进行转换。通过 ADA0M0.ADA0ETS1 位和 ADA0M0.ATA0ETS0 位来指定外部触发器的哪个边沿（也就是上升沿、下降沿或双边沿）被检测。ADA0CE 位设为 1 时，A/D 转换器等待触发，且当外部触发信号输入后，开始转换。

无论通过 ADA0MD1 位和 ADA0MD0 位设置的操作模式是连续选择、连续扫描、单次选择还是单次扫描模式，转换完成后，转换结果都存储在 ADA0CRn 寄存器中。同时，产生 INTAD 信号，并且 A/D 转换器等待下一次触发信号。

转换开始时，ADA0EF 位设为 1（表示转换正在进行）。然而，在 A/D 转换器等待触发期间，ADA0EF 位清为 0。如果在转换操作期间输入有效触发信号，转换操作中止，并再次从头开始。

如果在转换操作期间对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器执行写入操作，那么此次转换停止，并且 A/D 转换器等待再次触发。但是，在单次选择模式/单次扫描模式下，禁止写入这些寄存器（n = 0 至 11）。

注意事项 选择外部触发模式时，设置为高速转换模式。A/D 转换操作使能（ADA0M0.ADA0CE 位= 1）后，在随后插入的振荡稳定时间期间，不要输入触发信号。

备注 触发等待状态指稳定时间过后的状态。

（3）定时器触发模式

该模式下，通过该定时器的捕获/比较寄存器（连接至定时器）的比较匹配中断请求信号（INTTP2CC0 或 INTTP2CC1），开始对 ADA0S 寄存器指定的模拟输入引脚（ANI0 至 ANI11）的信号进行转换。通过 ADA0TMD1 位和 ADA0TMD0 位选择 INTTP2CC0 信号或 INTTP2CC1 信号，并且在指定的比较匹配中断请求信号的上升沿处开始转换。当 ADA0CE 位设为 1 时，A/D 转换器等待触发，当定时器的比较匹配中断请求信号输入后，开始转换。

转换完成时，无论通过 ADA0MD1 位和 ADA0MD0 位设置的操作模式是连续选择、连续扫描、单次选择还是单次扫描模式，转换结果都存储在 ADA0CRn 寄存器中。同时，产生 INTAD 信号，并且 A/D 转换器等待下一次触发。

转换开始时，ADA0EF 位设为 1（表示转换正在进行）。但是，在 A/D 转换器等待触发期间，ADA0EF 位清为 0。如果在转换操作期间输入有效触发，转换操作中止，并再次从头开始。

如果在转换操作期间对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器执行写入操作，那么此次转换停止，并且 A/D 转换器等待再次触发。但是，在单次选择模式/单次扫描模式下，禁止写入这些寄存器。

注意事项 选择定时器触发模式时，设置为高速转换模式。A/D 转换操作使能（ADA0M0.ADA0CE 位= 1）后，在随后插入的振荡稳定时间期间，不要输入触发信号。

备注 触发等待状态指稳定时间过后的状态。

14.5.4 操作模式

有四种操作模式可供选择：连续选择模式，连续扫描模式，单次选择模式以及单次扫描模式。

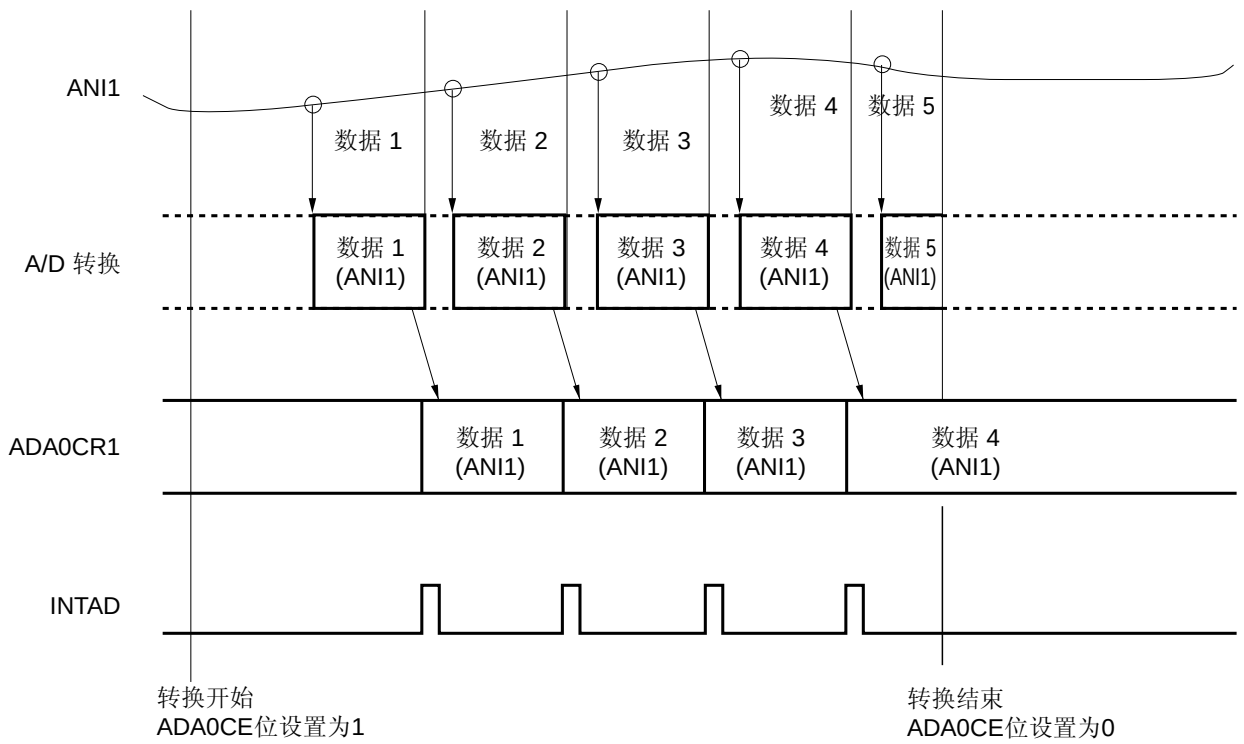
通过 ADA0M0.ADA0MD1 位和 ADA0M0.ADA0MD0 位来选择操作模式。

（1）连续选择模式

该模式下，ADA0S 寄存器选择的一个模拟输入引脚的电压被连续转换为数字值。

转换结果保存在该模拟输入引脚相对应的 ADA0CRn 寄存器中。每次 A/D 转换完成后，都会产生 A/D 转换结束中断请求信号（INTAD）。转换完成后，则开始下一次转换，除非 ADA0M0.ADA0CE 位被清为 0（n = 0 至 11）。

图 14-4. 连续选择模式操作的时序举例（ADA0S 寄存器= 01H）



备注 上述时序适用于软件触发模式(ADA0M0.AD0TMD 位 = 0) 或高速转换模式(ADA0M1.ADA0HS1 位 = 1)。

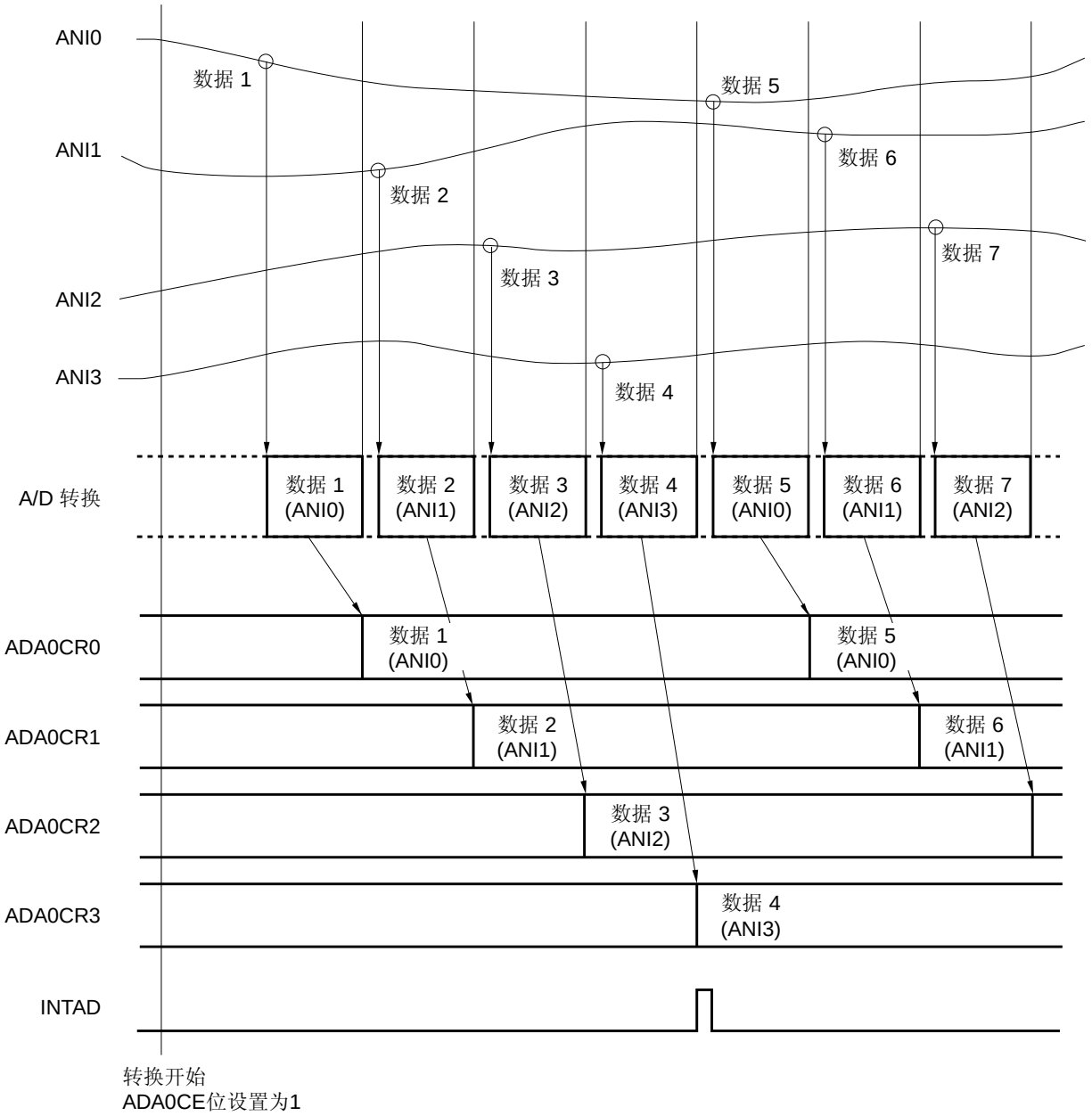
（2）连续扫描模式

该模式下，模拟输入引脚是顺序选择的，从 ANI0 引脚至 ADA0S 寄存器指定的引脚，并且将它们的输入值转换为数字值。

各个通道的转换结果保存在与各个模拟输入引脚相对应的 ADA0CRn 寄存器中。当寄存器指定的模拟输入引脚转换完成时，产生 INTAD 信号，并且又从 ANI0 引脚开始 A/D 扫描转换，除非 ADA0CE 位被清为 0（n = 0 至 11）。

图 14-5. 连续扫描模式操作的时序举例（ADA0S 寄存器= 03H）(1/2)

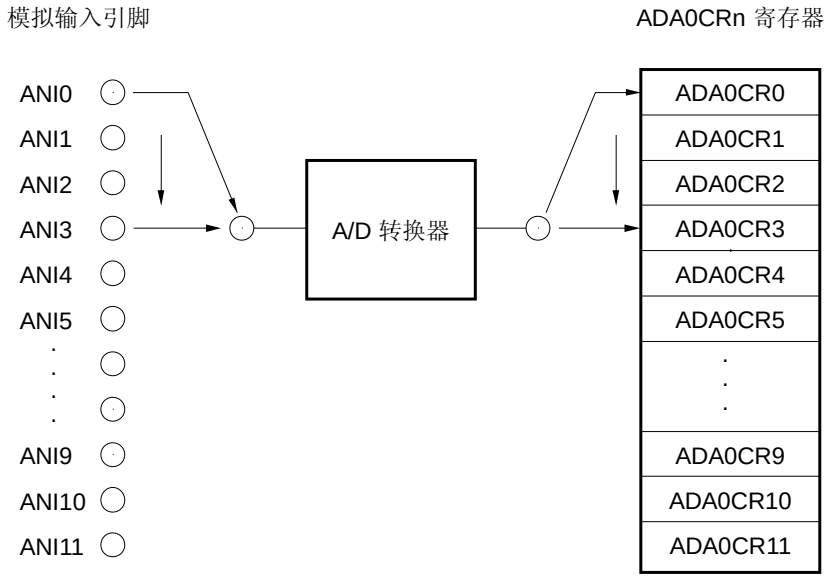
（a）时序举例



备注 上述时序适用于软件触发模式(ADA0M0.AD0TMD 位 = 0) 或高速转换模式(ADA0M1.ADA0HS1 位 = 1)。

图 14-5. 连续扫描模式操作的时序举例（ADA0S 寄存器= 03H）(2/2)

（b）模拟输入引脚与 A/D 转换结果寄存器之间的关系

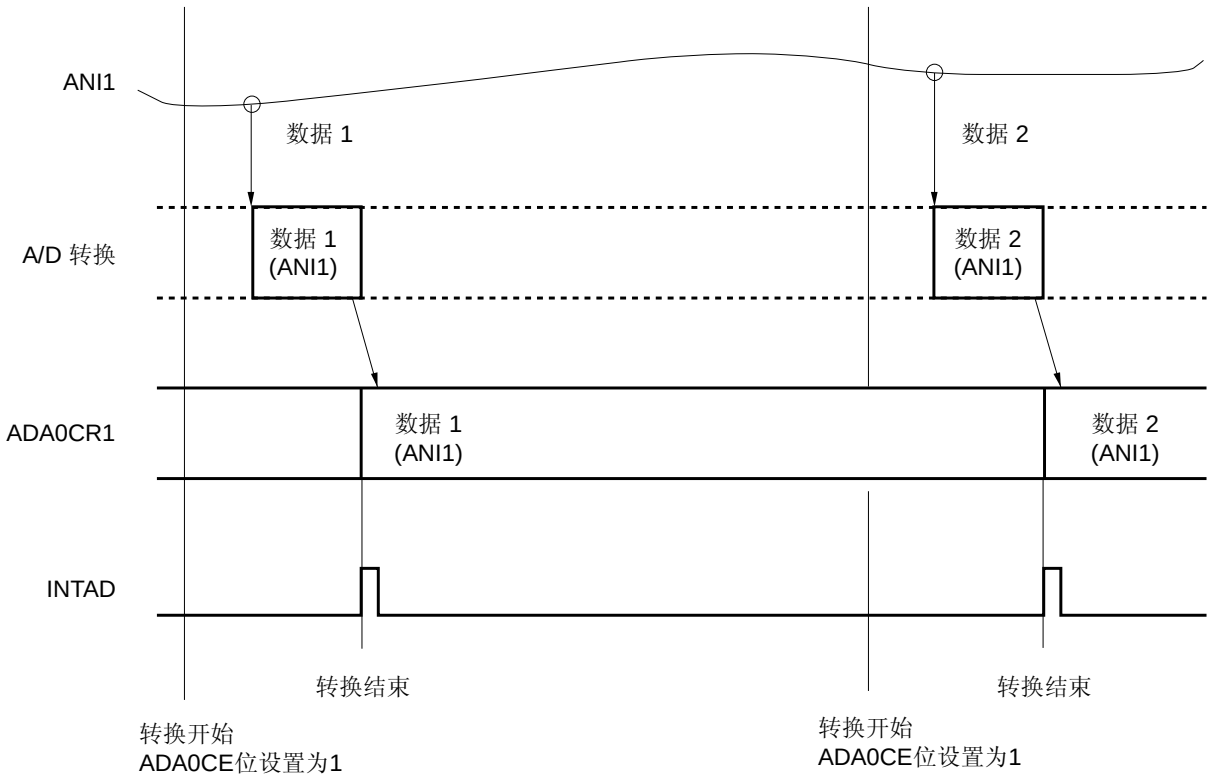


（3）单次选择模式

该模式下，对 ADA0S 寄存器选择的模拟输入引脚的电压转换为数字值的操作仅进行一次。

转换结果保存在该模拟输入引脚相对应的 ADA0CRn 寄存器中。该模式下，模拟输入引脚与 ADA0CRn 寄存器一一对应。A/D 转换完成一次后，产生 INTAD 信号。上述操作完成后，A/D 转换操作停止（n = 0 至 11）。

图 14-6. 单次选择模式操作的时序举例（ADA0S 寄存器= 01H）



备注 上述时序适用于软件触发模式(ADA0M0.AD0TMD 位 = 0) 或高速转换模式(ADA0M1.ADA0HS1 位 = 1)。

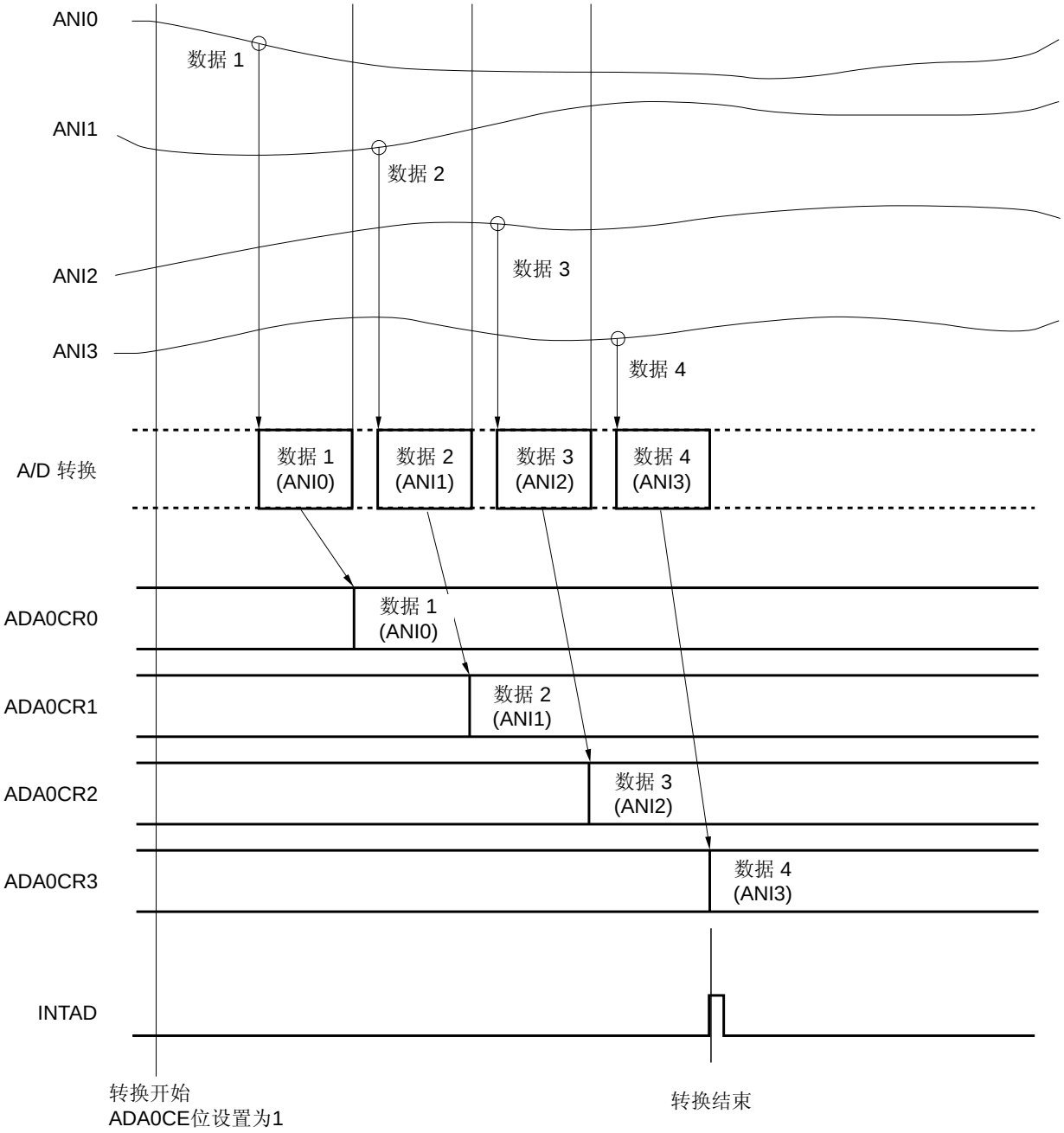
（4）单次扫描模式

该模式下，模拟输入引脚是顺序选择的，从 ANI0 引脚至 ADA0S 寄存器指定的引脚，并且将它们的输入值转换为数字值。

各个转换结果保存在各个模拟输入引脚相对应的 ADA0CRn 寄存器中。当 ADA0S 寄存器指定的模拟输入引脚转换完成时，产生 INTAD 信号。上述操作完成后，A/D 转换停止（n=0 到 11）。

图 14-7. 单次扫描模式操作的时序举例（ADA0S 寄存器= 03H）(1/2)

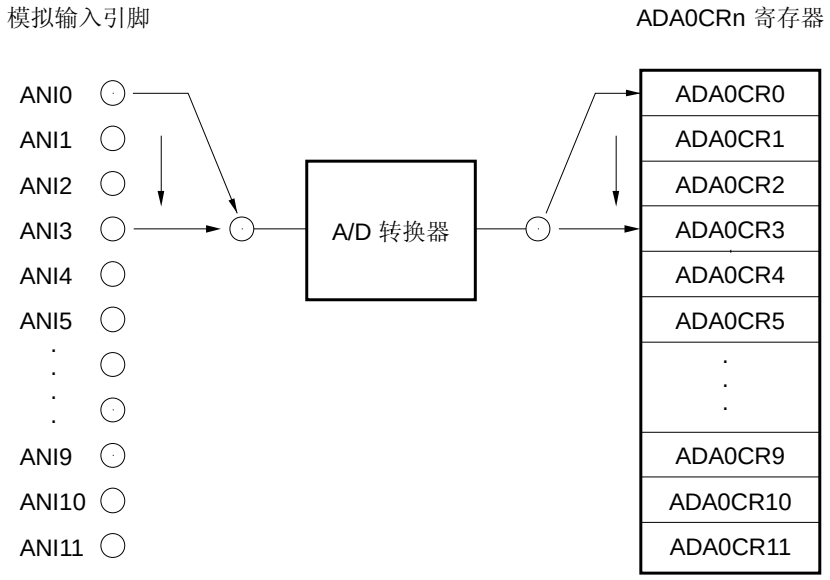
（a）时序举例



备注 上述时序适用于软件触发模式(ADA0M0.AD0TMD 位 = 0) 或高速转换模式(ADA0M1.ADA0HS1 位 = 1)。

图 14-7. 单次扫描模式操作的时序举例（ADA0S 寄存器= 03H）(2/2)

（b）模拟输入引脚与 A/D 转换结果寄存器之间的关系



14.5.5 掉电比较模式

该模式下，无论判断是输入模拟信号电压为指定电压或更高，或低于指定电压。如果符合 ADA0PFC 位指定的条件，将中断请求信号（INTAD）。

- 当 ADA0PFM.ADA0PFE 位= 0 时，每次 A/D 转换完成（正常使用 A/D 转换器）在下列时序产生 INTAD 信号。

  - 连续/单次选择模式：　　在第一次 A/D 转换完成之后
  - 连续/单次扫描模式：　　直到 ADA0S 寄存器指定的模拟输入引脚的 A/D 转换顺序执行之后

- 当 ADA0PFE 位为 1 且 ADA0PFM.ADA0PFC 位为 0 时，ADA0CRnH 寄存器的值与转换完成时 ADA0PFT 寄存器的值进行比较，并且仅当 ADA0CRnH ≥ ADA0PFT 时，产生 INTAD 信号。
- 当 ADA0PFE 位为 1 且 ADA0PFC 位为 1 时，ADA0CRnH 寄存器的值与转换完成时 ADA0PFT 寄存器的值进行比较，并且仅当 ADA0CRnH < ADA0PFT 时，产生 INTAD 信号。

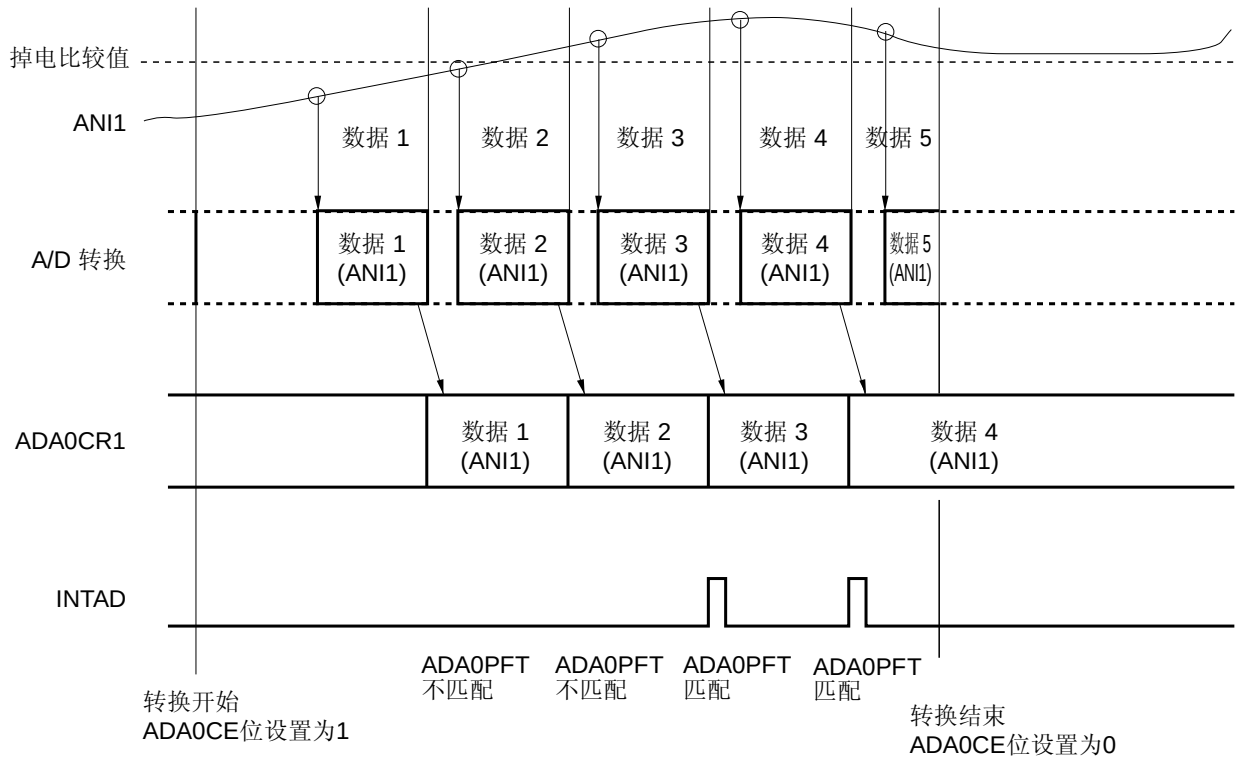
备注　　n = 0 至 11

掉电比较模式中，有四种模式可供选择：连续选择模式，连续扫描模式，单次选择模式以及单次扫描模式。

（1）连续选择模式

该模式下，ADA0S 寄存器指定的模拟输入引脚的电压转换结果与 ADA0PFT 寄存器的设置值相比较。如果掉电比较的结果符合 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CRn 寄存器，并产生 INTAD 信号。如果不符合条件，转换结果仍存储在 ADA0CRn 寄存器中，但不产生 INTAD 信号。第一次转换完成后，则开始下一次转换，除非 ADA0M0.ADA0CE 位被清为 0（n = 0 至 11）。

图 14-8. 连续选择模式操作的时序举例
（进行掉电比较时：ADA0PFM.ADA0PFC 位 = 0，ADA0S 寄存器 = 01H）



备注 上述时序适用于软件触发模式(ADA0M0.AD0TMD 位 = 0) 或高速转换模式(ADA0M1.ADA0HS1 位 = 1)。

（2）连续扫描模式

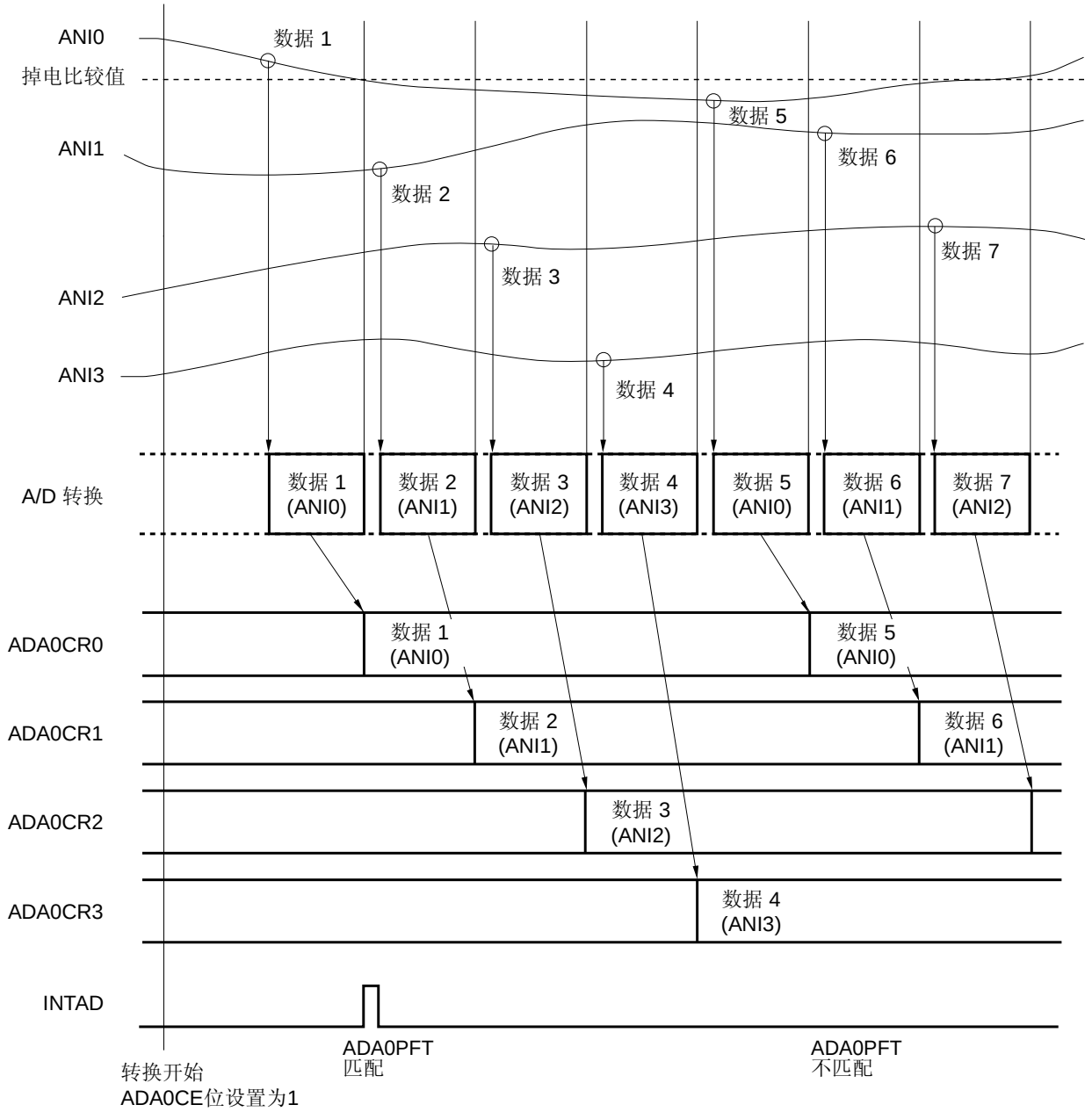
该模式下，对 ANI0 引脚至 ADA0S 寄存器指定引脚的模拟输入转换结果被顺序存储。

首先，比较通道 0 的转换结果。如果掉电比较的结果符合 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CR0 寄存器，并产生 INTAD 信号。如果不符合条件，转换结果仍存储在 ADA0CR0 寄存器中，但不产生 INTAD 信号。

第一次转换结果存储在 ADA0CR0 寄存器之后，顺序对模拟输入引脚上的电压进行转换并存储，至 ADA0S 寄存器指定的模拟输入引脚。转换完成后，又从 ANI0 引脚启动下一轮转换，直到 ADA0CE 位被清为 0。

图 14-9. 连续扫描模式操作的时序举例
（进行掉电比较时：ADA0PFM.ADA0PFC 位 = 0，ADA0S 寄存器 = 03H）(1/2)

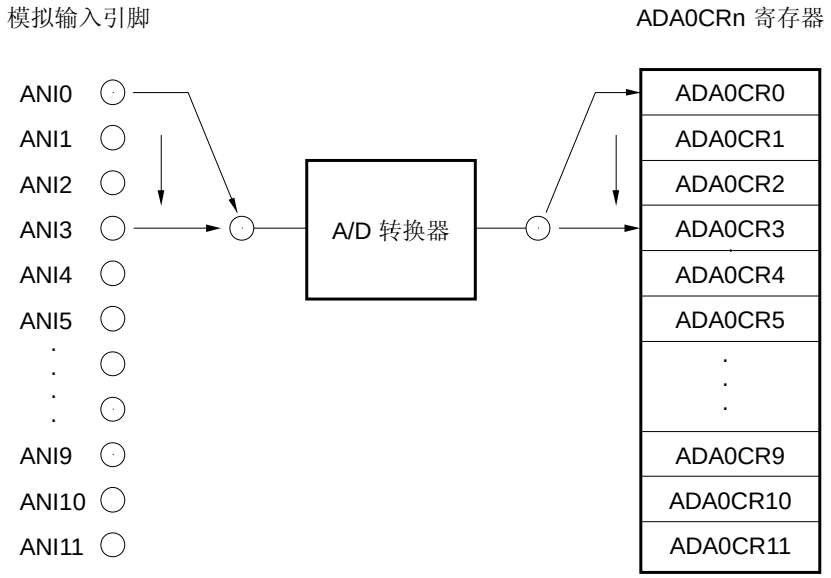
（a）时序举例



备注 上述时序适用于软件触发模式(ADA0M0.AD0TMD 位 = 0) 或高速转换模式(ADA0M1.ADA0HS1 位 = 1)。

图 14-9. 连续扫描模式操作的时序举例
（进行掉电比较时：ADA0PFM.ADA0PFC 位 = 0，ADA0S 寄存器 = 03H）(2/2)

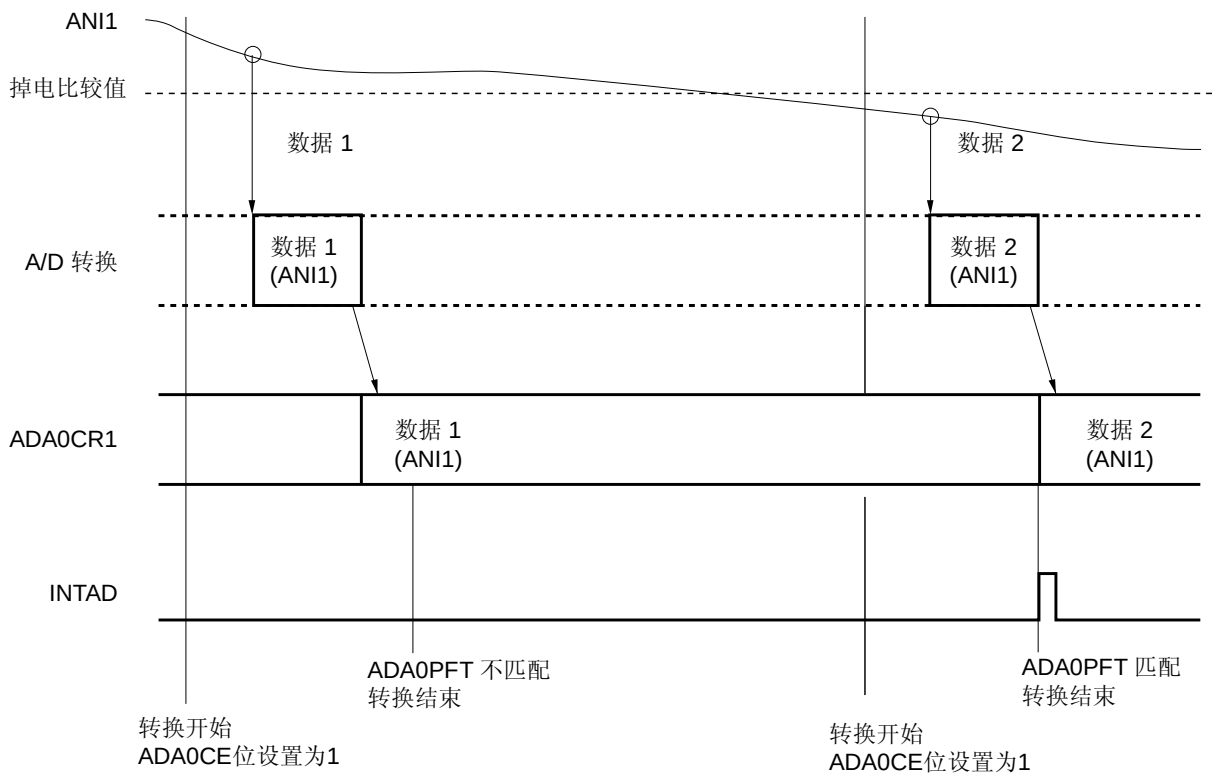
（b）模拟输入引脚与 A/D 转换结果寄存器之间的关系



（3）单次选择模式

该模式下，ADA0S 寄存器指定的模拟输入引脚的电压转换结果与 ADA0PFT 寄存器的设置值相比较。如果掉电比较的结果符合由 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CRn 寄存器，并产生 INTAD 信号。如果不符合条件，转换结果仍存储在 ADA0CRn 寄存器中，但不产生 INTAD 信号。上述操作完成后，停止转换。

图 14-10. 单次选择模式操作的时序举例
（进行掉电比较时：ADA0PFM.ADA0PFC 位 = 0，ADA0S 寄存器 = 01H）



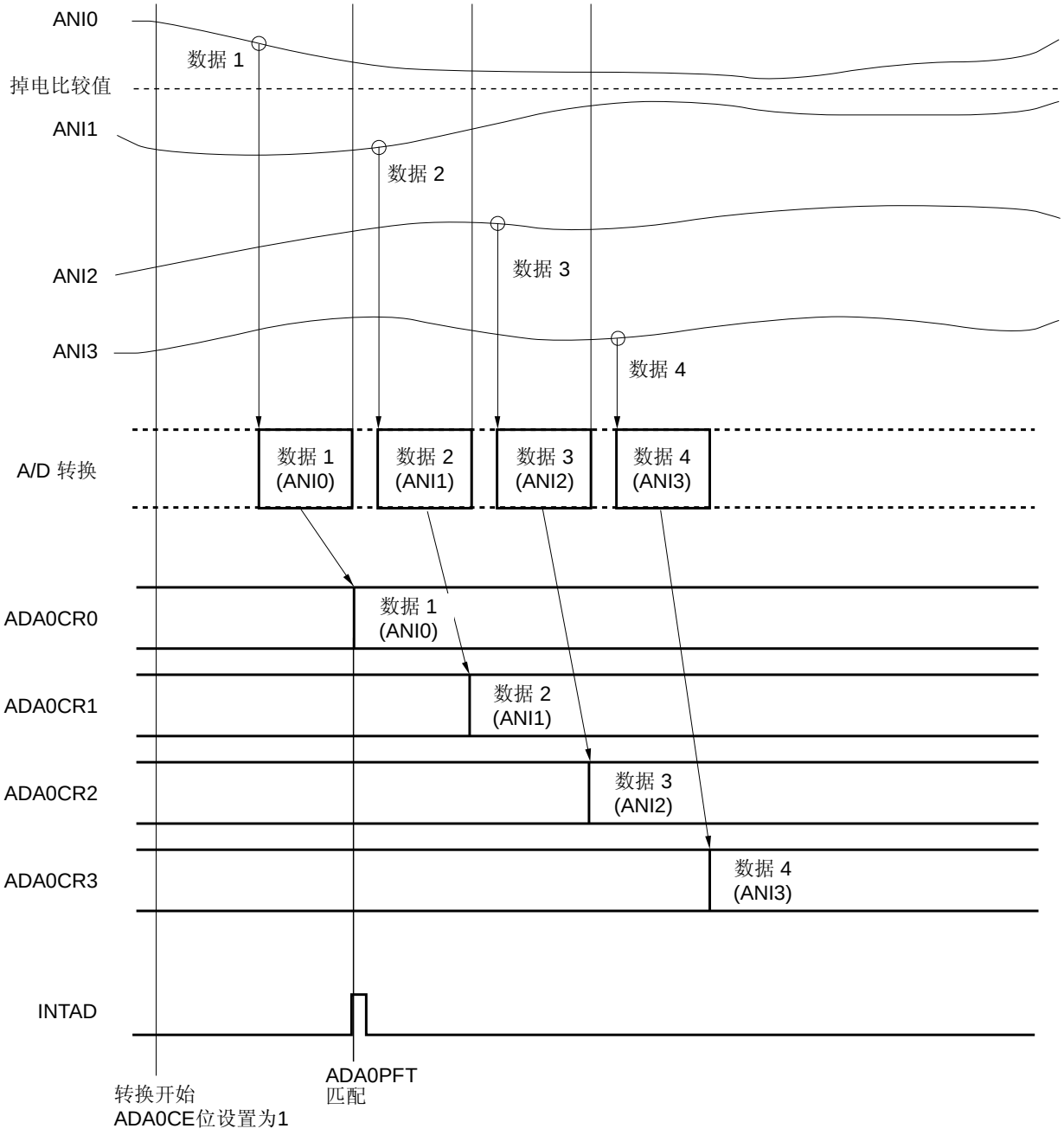
备注 上述时序适用于软件触发模式(ADA0M0.AD0TMD 位 = 0) 或高速转换模式(ADA0M1.ADA0HS1 位 = 1)。

（4）单次扫描模式

该模式下，模拟输入引脚是顺序选择的，从 ANI0 引脚至 ADA0S 寄存器指定的引脚，对它们的模拟输入值进行 A/D 转换，存储电压转换结果，并且将通道 0 的 ADA0CR0H 寄存器设置值与 ADA0PFT 寄存器的设置值作比较。如果掉电比较的结果符合由 ADA0PFC 位指定的条件，那么将转换结果存储在 ADA0CR0 寄存器，并产生 INTAD 信号。如果不符合条件，转换结果仍存储在 ADA0CR0 寄存器中，但不产生 INTAD0 信号。第一次转换结果存储在 ADA0CR0 寄存器后，依次存储 ADA0S 寄存器指定的模拟输入引脚的转换信号结果。上述操作完成后，停止转换。

图 14-11. 单次扫描模式操作的时序举例
（进行掉电比较时：ADA0PFM.ADA0PFC 位 = 0，ADA0S 寄存器 = 03H）(1/2)

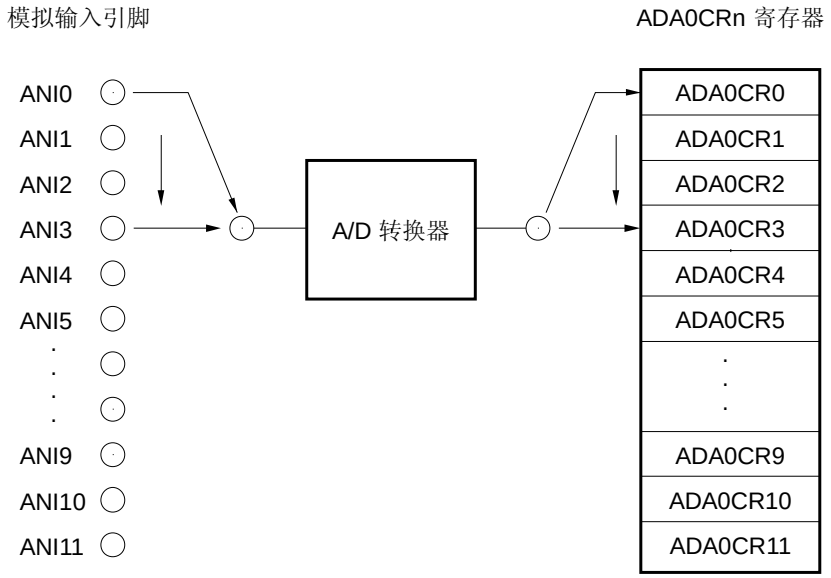
（a）时序举例



备注 上述时序适用于软件触发模式(ADA0M0.AD0TMD 位 = 0) 或高速转换模式(ADA0M1.ADA0HS1 位 = 1)。

图 14-11. 单次扫描模式操作的时序举例
（进行掉电比较时：ADA0PFM.ADA0PFC 位 = 0，ADA0S 寄存器 = 03H）(2/2)

（b）模拟输入引脚与 A/D 转换结果寄存器之间的关系



14.6 注意事项

（1）不使用 A/D 转换器时

不使用 A/D 转换器时，将 ADA0M0.ADA0CE 位清 0，可以降低功耗。

（2）ANI0 至 ANI11 引脚输入范围

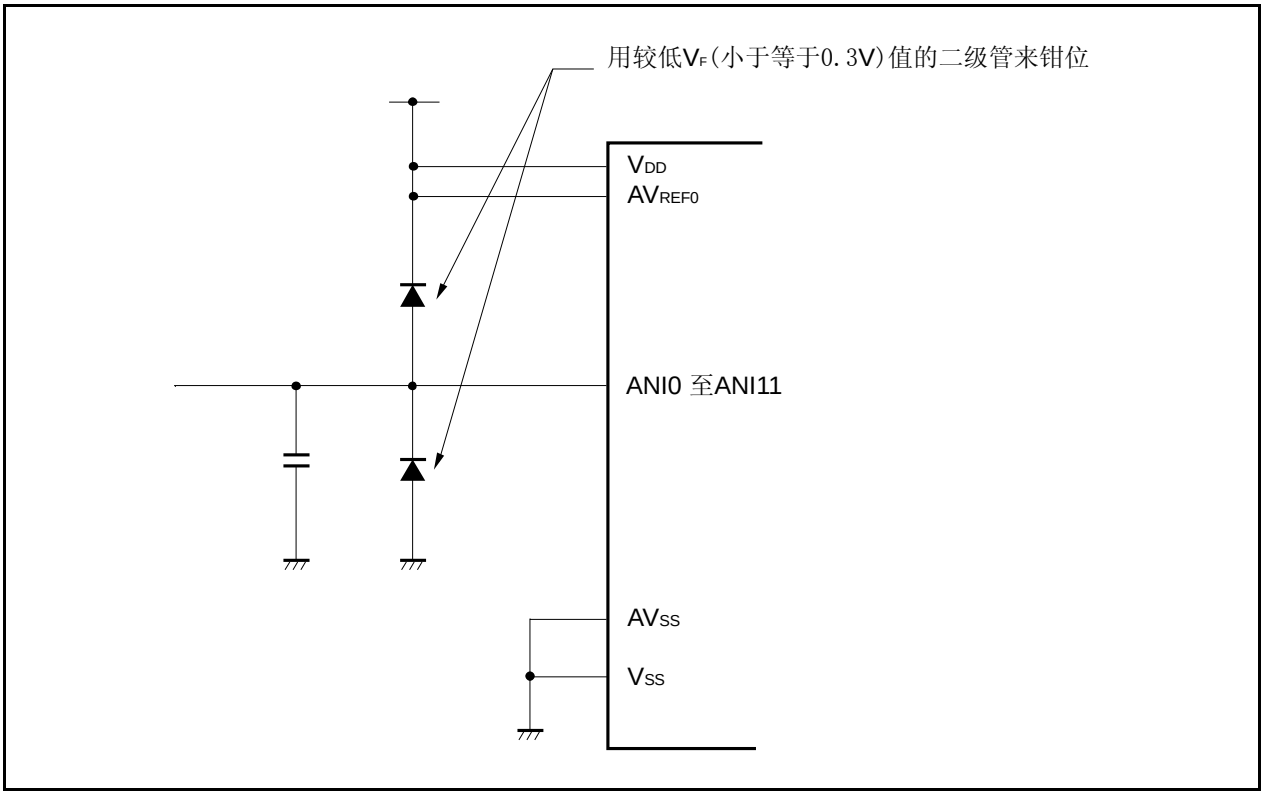
向 ANI0 引脚至 ANI11 引脚输入的电压应该在指定范围内。如果大于等于 AV_{REF0} 或小于等于 AV_{SS}（即使在最大额定值范围内）的电压输入到这些引脚中的任意一个引脚，则此通道的转换值将不确定，并且其它通道的转换值也会受到影响。

（3）应对噪声的措施

为了保证 10 位分辨率，必须有效防止噪声对 ANI0 引脚至 ANI11 引脚的干扰。噪声的影响随着模拟输入源输出阻抗的增大而加强。为了减小噪声，建议连接一个外部电容器，方式如图 14-12 所示。

该电容器的容抗值必须适合输入信号的改变速度。

图 14-12. 模拟输入引脚的处理



（4）复用 I/O

模拟输入引脚（ANI0 至 ANI11）可以复用为端口引脚。当选择 ANI0 引脚至 ANI11 引脚的其中一个来执行 A/D 转换时，在转换期间不要执行读取输入端口或写入输出端口的指令，因为这样可能会降低转换的分辨率。

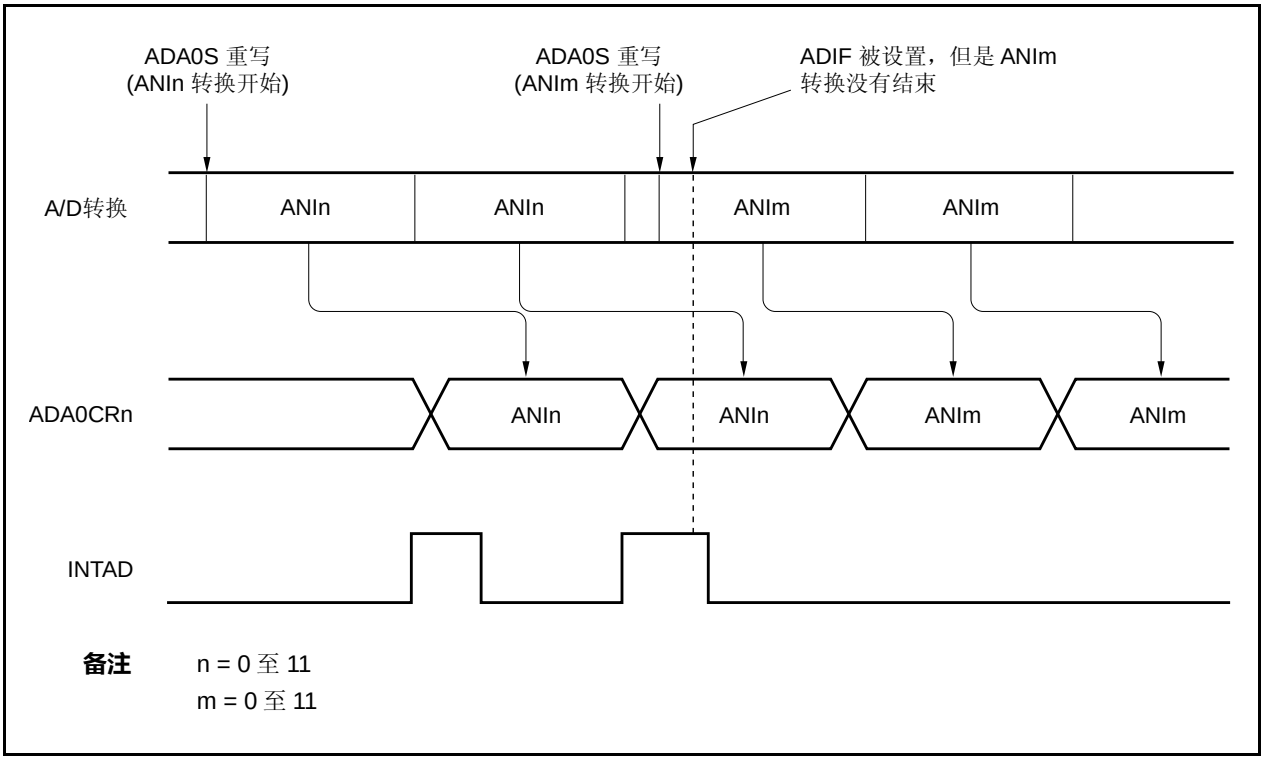
在 A/D 转换期间端口的某些引脚被设置作为输出，由于连接至端口引脚的外部电路干扰使得输出电流波动，输入引脚的转换分辨率也会下降。

如果有数字脉冲施加在与正被转换的输入信号引脚相邻的引脚上，由于耦合噪声的影响，A/D 转换值可能会无法预料。因此，不要在进行 A/D 转换引脚的邻近引脚上施加脉冲。

（5）中断请求标志（ADIF）

即使 ADA0S 寄存器的内容改变，也不会清除(0)中断请求标志（ADIF）。因此，A/D 转换期间如果改变模拟输入通道，那么先前所选模拟输入引脚信号的转换结果可能被存储，且有可能在紧邻 ADA0S 寄存器重写之前，转换结束中断请求标志被置位(1)。如果写入 ADA0S 寄存器之后立即读取 ADIF 标志，那么即使新选择的模拟输入引脚的 A/D 转换还没有完成，也会置位 ADIF 标志(1)。A/D 转换停止时，重新开始转换之前应该清除 ADIF 标志(0)。

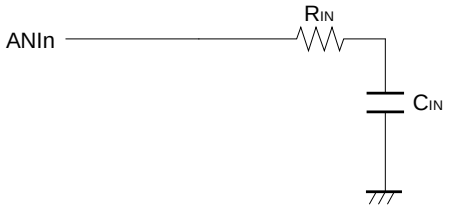
图 14-13. A/D 转换结束中断请求的发生时序



（6）内部等效电路

以下显示了模拟输入模块的等效电路。

图 14-14. ANIn 引脚的内部等效电路



R_{IN}	C_{IN}
14 k	8.0 pF

备注 **1.** 以上数值是参考值。

2. n = 0 至 11

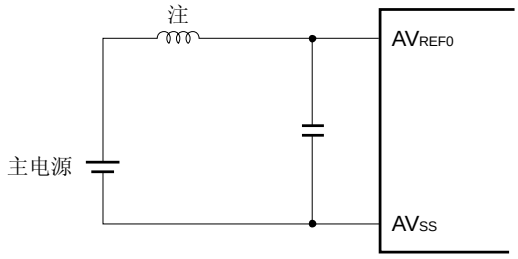
（7）AV_{REF0} 引脚

（a） AV_{REF0} 引脚用作 A/D 转换器的电源，也为复用功能端口提供电源。在使用备用电源的应用系统中，确保提供至 AV_{REF0} 引脚的电压与 V_{DD} 相同，如图 14-15 所示。

（b） AV_{REF0} 引脚用作 A/D 转换器的参考电压引脚。如果 AV_{REF0} 引脚的输入电源具有高阻抗或者电流提供能力较弱，那么转换期间由于电流波动，参考电压可能会波动（尤其紧接在转换操作使能位 ADA0CE 被置为 1 之后）。结果，转换精度可能下降。为了避免这种情况，建议在 AV_{REF0} 引脚和 AV_{SS} 引脚之间连接一个电容器来抑制参考电压的波动，如图 14-15 所示。

（c） 如果连接至 AV_{REF0} 引脚的电源有具有高直流电阻，由于 A/D 转换电流引起电压下降，所以开始转换时的电压可能低于转换停止时的电压。

图 14-15. AV_{REF0} 引脚处理实例



注 寄生电感

（8）读取 ADA0CRn 寄存器

当写入 ADA0M0 至 ADAM2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器时，ADA0CRn 寄存器的内容可能不确定。在转换完成之后且写入 ADA0M0 至 ADAM2、ADA0S、ADA0PFM 或 ADA0PFT 寄存器之前，读取转换结果。当响应外部/定时器触发时，ADA0CRn 寄存器的内容也可能不确定。在转换完成之后且响应下一次外部/定时器触发之前，读取转换结果。不同于上述时序时，可能读取不到正确的转换结果。

（9）待机模式

在 STOP 模式下，由于 A/D 转换器停止工作，因此可以降低功耗，但是转换结果将是无效的。STOP 模式解除后，操作重新开始，但是 STOP 模式解除后的 A/D 转换结果无效。STOP 模式解除后，需要使用 A/D 转换器时，在设置 STOP 模式之前或解除 STOP 模式之前，将 ADA0M0.ADA0CE 位清 0，然后在解除 STOP 模式之后，将 ADA0CE 位设为 1。

在 IDLE1、IDLE2 或副时钟操作模式下，操作继续。在 IDLE1 和 IDLE2 模式下，由于不能够保持模拟输入引脚的电压值，所以 IDLE1 和 IDLE2 模式解除后，A/D 转换结果无效。设置进入 IDLE1 和 IDLE2 模式之前的转换结果是有效的。要降低功耗，将 ADA0M0.ADA0CE 位清除为 0。

（10）高速转换模式下

在高速转换模式下，振荡稳定期间，禁止触发信号的输入，也禁止对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 和 ADA0PFT 寄存器执行重写操作。

（11）A/D 转换时间

A/D 转换时间为稳定时间，转换时间，等待时间和触发响应时间的总和（关于这些时间的细节，参见**表 14-2 正常转换模式（ADA0HS1 位= 0）下转换时间的选择**和**表 14-3 高速转换模式（ADA0HS1 位 = 1）下转换时间的选择。**

在正常转换模式下，A/D 转换期间，如果对 ADA0M0、ADA0M2、ADA0S、ADA0PFM 以及 ADA0PFT 寄存器执行重写操作或者输入触发信号，则重新执行转换。但是，如果稳定时间结束的时序与写入这些寄存器的时序发生冲突，或者稳定时间结束的时序与触发信号输入的时序发生冲突，那么将会插入 64 个时钟的稳定时间。

如果在插入的稳定时间结束时序再次发生时序冲突，则仍然插入稳定时间。因此，不要将触发信号输入间隔和控制寄存器写入间隔设置为 64 个时钟或更少。

（12）A/D 转换结果的变化

因为供应电压的波动或噪声的影响，A/D 转换的结果可能会有变化。为了减小变化，在程序中应用抑制措施，例如取 A/D 转换结果平均值。

（13）A/D 转换结果迟滞特性

逐次比较类型 A/D 转换器在内部采样&保持电容器上保持输入电压，然后进行 A/D 转换。完成后，模拟输入电压仍然保持在内部采样保持电容器上。因此，可能发生以下现象：

- 当相同通道被用于 A/D 转换时，如果当前电压比先前的电压高或低，在转换结果受先前值影响的地方，可能出现滞后特性。因此，即使对相同的电压进行转换，结果也可能不同。
- 当切换模拟输入通道时，在转换结果受先前通道值影响的地方，可能出现滞后特性。这是因为 A/D 转换功能只有一个 A/D 转换器。因此，即使对相同的电压进行转换，结果也可能不同。

因此，要获得更精确的转换结果，对同一个通道连续执行两次 A/D 转换，并丢弃第一次转换结果。

14.7 A/D转换器特征表的阅读方法

本部分描述了 A/D 转换器的相关术语。

（1）分辨率

分辨率为能够识别的最低模拟输入电压，也就是说，每位输出数字所占模拟输入电压的百分比，叫作 1 LSB（最低有效位）。1 LSB 对满量程的比率被表示为%FSR（满标度量程）。%FSR 是百分比表示的可转换模拟输入电压比率，不管分辨率如何，总是可以用下式表示：

1%FSR （可转换的模拟输入电压的最大值 – 可转换的模拟输入电压的最小值）/100
= （AV_{REF0} – 0）/100
= AV_{REF0}/100

当分辨率为 10 位时，1 LSB 表示如下：

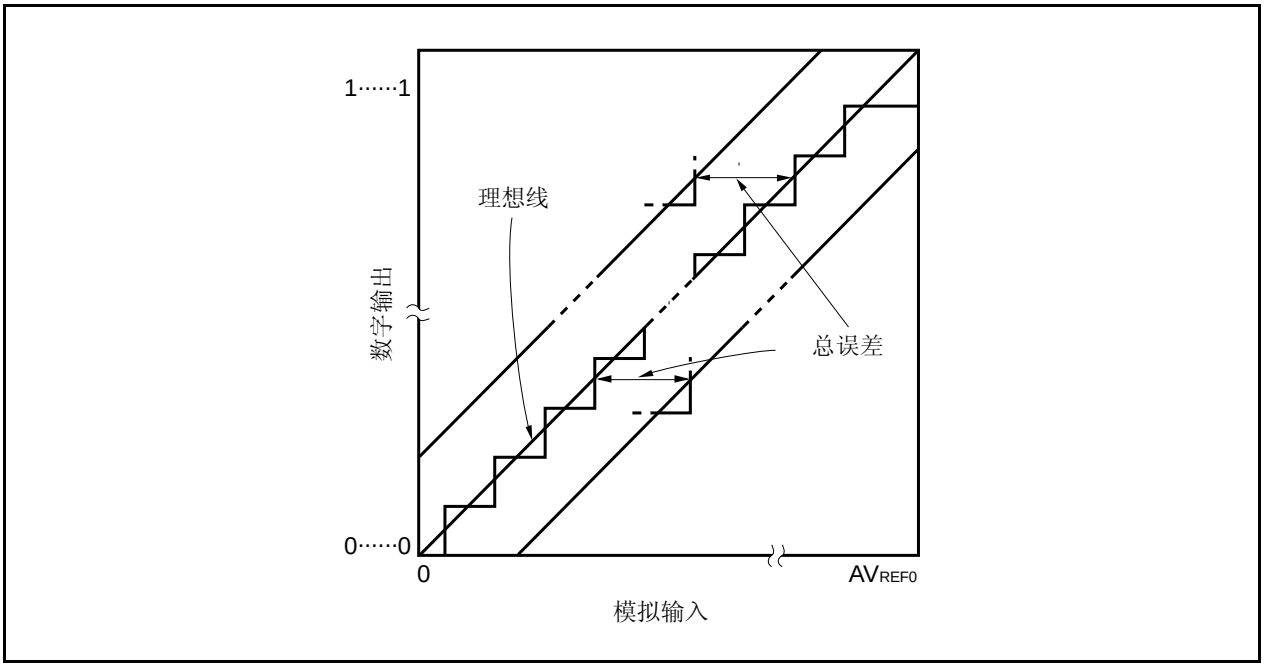
1 LSB = $1/2^{10}$ = 1/1,024
= 0.098%FSR

精确度是由总体误差决定，与分辨率无关。

（2）总体误差

总体误差是实际测量值和理论值之间的最大误差值。
零标度误差、满量程误差、线性误差，这些误差的综合构成总体误差。
在特性表中的总体误差里没有包含量化误差。

图 14-16. 总体误差

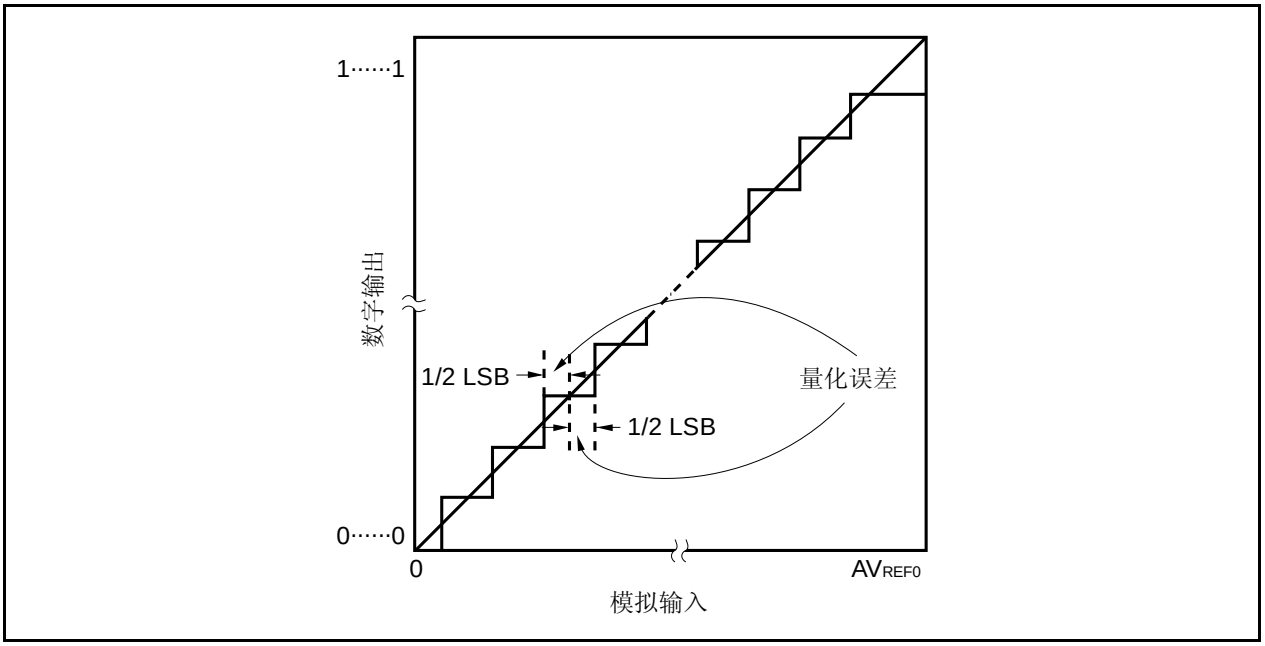


（3）量化误差

当模拟值转换为数字值时，会不可避免的产生 ±1/2 LSB 的误差。因为 A/D 转换器将±1/2 LSB 范围内的模拟输入电压转换成相同的数字值，所以量化误差不能避免。

量化误差不包括在参数表中的总体误差、零标度误差、满量程误差、积分线性误差或微分线性误差之中。

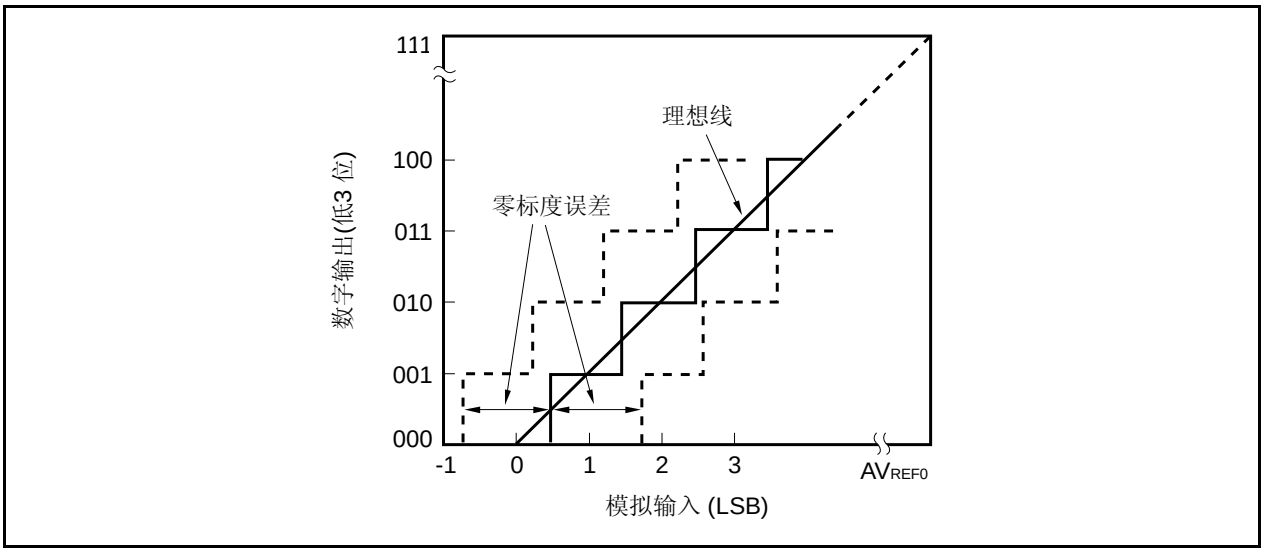
图 14-17. 量化误差



（4）零标度误差

当数字输出从 0……000 改变到 0……001（1/2 LSB）时，零标度误差就是实际测量的模拟输入电压和理论值之间的差别。

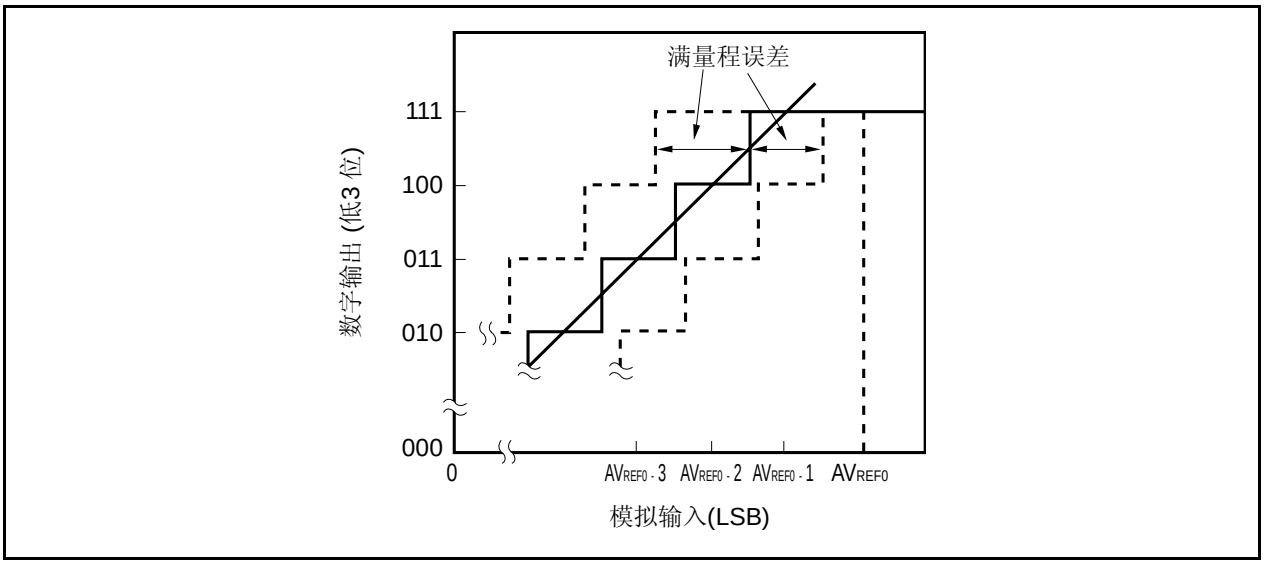
图 14-18. 零标度误差



（5）满量程误差

当数字输出从 1…110 改变到 1…111（满量程 – 3/2 LSB）时，满量程误差就是实际测量的模拟输入电压与理论值之间的差别。

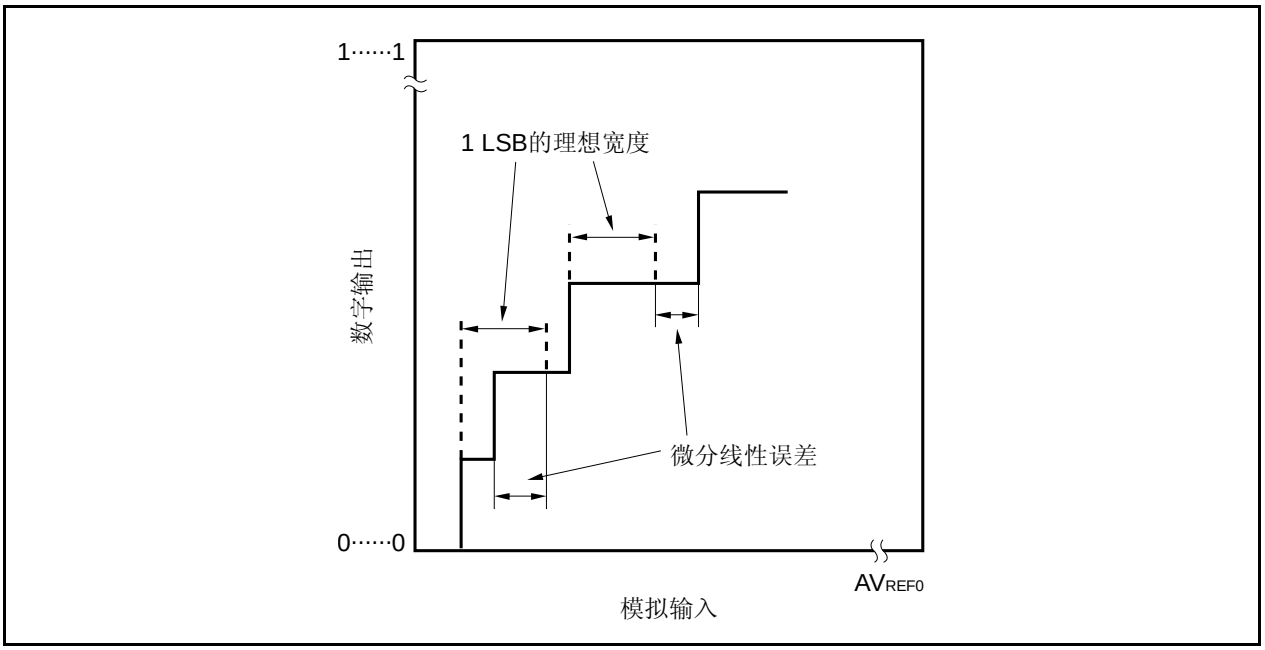
图 14-19. 满量程误差



（6）微分线性误差

理想状态下，输出指定编码的宽度是 1 LSB。输出指定编码时，该误差显示了实际测量值和其理论值之间的差别。输入到相同通道的模拟输入引脚的电压从 AV_{SS} 逐位增加到 AV_{REF0}，这代表了 A/D 转换的基本特性。当输入电压增加或减小时，或使用两个或更多的通道时，参见 **14.7（2）总体误差**。

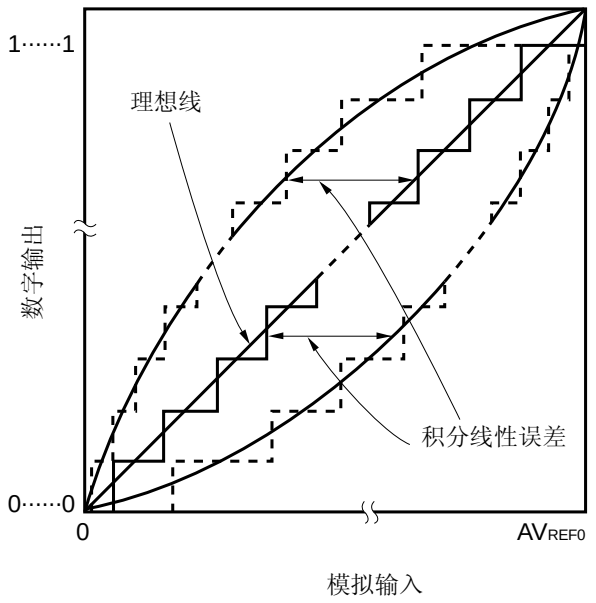
图 14-20. 微分线性误差



（7）积分线性误差

积分线性误差显示转换特性偏离理想线性关系的程度。它表示当零标度误差和满量程误差均为 0 时，实际测量值和其理论值之间的最大误差。

图 14-21. 积分线性误差



（8）转换时间

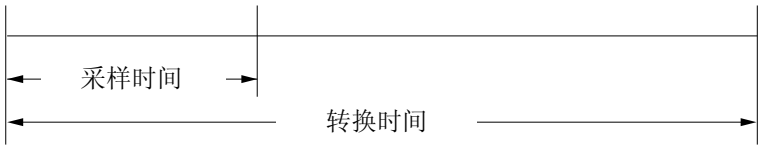
转换时间为触发信号产生后获得数字输出所需的时间。

特性表中，转换时间包含采样时间。

（9）采样时间

采样时间，即模拟开关开启，将模拟电压载入采样&保持电路的时间。

图 14-22. 转换时间与采样时间的关系



第十五章 D/A 转换器

15.1 功能

在 V850ES/JG3-L 中，提供两路 R-2R 阶梯型 D/A 转换通道（DA0CS0 和 DA0CS1）。
D/A 转换器具有下列功能：

- 8 位精度 × 2 通道（DA0CS0，DA0CS1）
- R-2R 阶梯方式
- 转换时间：最大 3 μs（AV_{REF1} 为 2.7 至 3.6 V 且外部阻抗为 20 pF 时）
- 模拟输出电压： $AV_{REF1} \times m/256$ （m = 0 至 255；DA0CSn 寄存器的设定值）
- 操作模式： 正常模式，实时输出模式

备注 n = 0, 1

15.2 配置

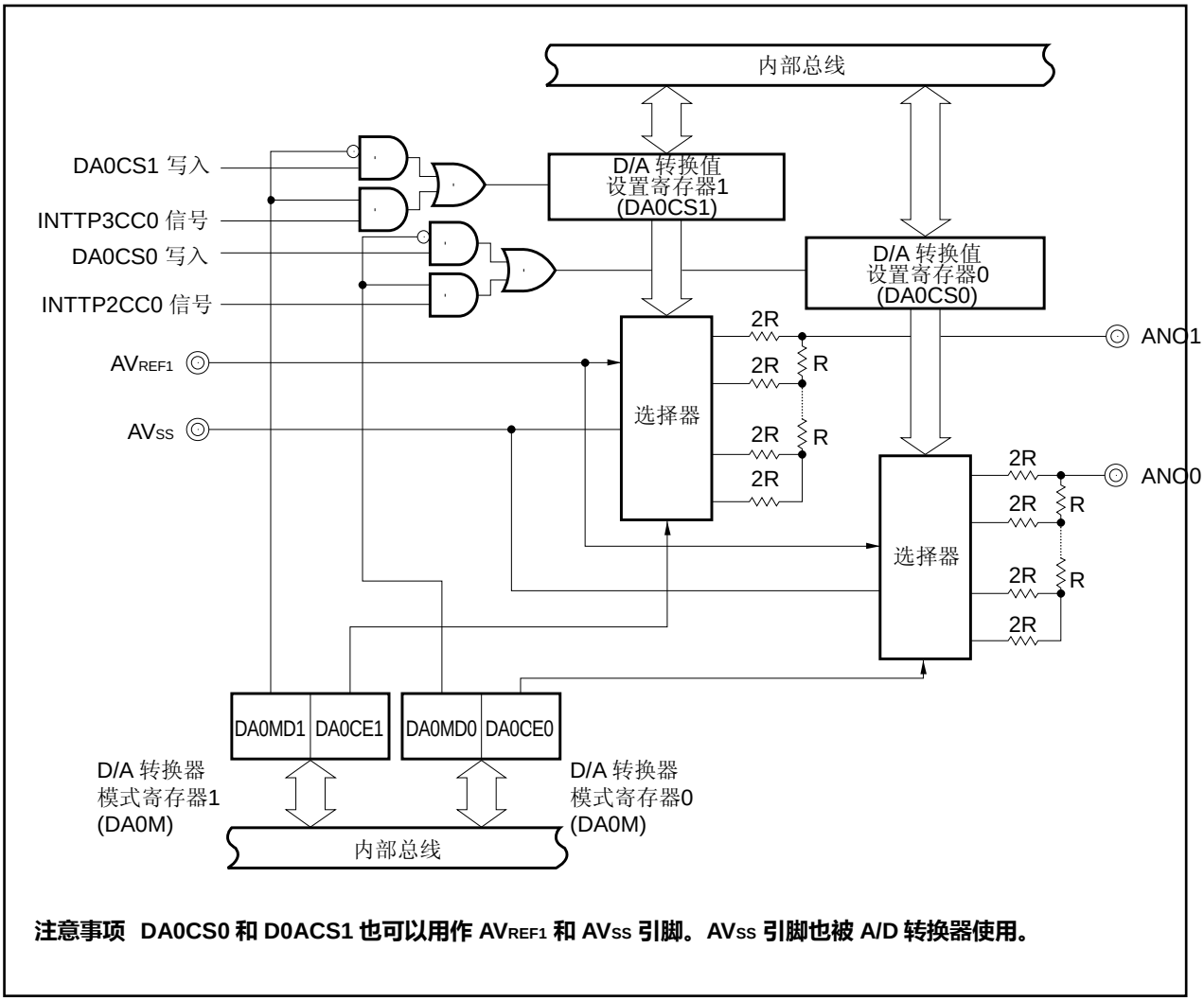
D/A 转换器包括下列硬件：

表 15-1. D/A 转换器的配置

项目	配置
控制寄存器	D/A 转换器模式寄存器 （DA0M） D/A 转换值设置寄存器 0 和 1（DA0CS0，DA0CS1）

D/A 转换器配置如下所示：

图 15-1. D/A 转换器功能框图



注意事项 DA0CS0 和 D0ACS1 也可以用作 AVREF1 和 AVSS 引脚。AVSS 引脚也被 A/D 转换器使用。

15.3 寄存器

控制 D/A 转换器的寄存器如下所示：

- D/A 转换器模式寄存器 （DA0M）
- D/A 转换值设置寄存器 0，1（DA0CS0，DA0CS1）

（1）D/A 转换器模式寄存器 （DA0M）

DA0M 寄存器用于控制 D/A 转换器的操作。
该寄存器可进行字节读写或按位读写。
系统复位后，该寄存器被设为 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/禁止控制（n = 0，1）
0	禁止操作
1	使能操作

DA0MDn	D/A 转换器操作模式的选择（n = 0，1）
0	正常模式
1	实时输出模式[注]

注　实时输出模式 （DA0MD0 位 = 1）的输出触发信号如下：
- 当 n = 0 时：INTTP2CC0 信号（参见**第七章 16 位定时器/事件计数器 P （TMP）**）。
- 当 n = 1 时：INTTP3CC0 信号（参见**第七章 16 位定时器/事件计数器 P （TMP）**）。

（2）D/A 转换值设置寄存器 0，1（DA0CS0，DA0CS1）

DA0CS0 和 DA0CS1 寄存器用于设置输出到 ANO0 和 ANO1 引脚的模拟电压。

这些寄存器可以进行字节读取或写入。

系统复位后，这些寄存器被设为 00H。

复位后: 00H　　R/W　　地址: FFFFF280H

	7	6	5	4	3	2	1	0
DA0CS0	DA0CS07	DA0CS06	DA0CS05	DA0CS04	DA0CS03	DA0CS02	DA0CS01	DA0CS00

注意事项　在实时输出模式（DA0M.DA0MDn 位 = 1）下，在 INTTP2CC0/ INTTP3CC0 信号产生以前，设置 DA0CSn 寄存器。当 INTTP2CC0/ INTTP3CC0 信号产生时，D/A 转换器开始工作。

备注　n = 0, 1

15.4 操作

15.4.1 正常模式下的操作

将写入 DA0CSn 寄存器的操作作为触发信号，执行 D/A 转换操作。
设置方法如下：

<1> 将 PM1n 位设置为 1（输入模式）。
<2> 将 DA0M.DA0MDn 位清除为 0（正常模式）。
<3> 将要输出到 ANOn 引脚的模拟电压值写入 DA0CSn 寄存器。
以上步骤<2> 和步骤<3> 为初始化设置。
<4> 将 DA0M.DA0CEn 位设置为 1（D/A 转换使能）。
执行该设置时，D/A 转换的模拟电压值从 ANOn 引脚输出。
<5> 要改变模拟电压值，写入 DA0CSn 寄存器。
之前的模拟电压值被保持，直到执行下一次写入操作。

备注 **1.** 关于复用功能引脚的设置，参见 **表 4-15 当端口引脚用作复用功能时的设置**。
2. n = 0, 1

15.4.2 实时输出模式下的操作

使用 TMP2 和 TMP3 的中断请求信号（INTTP2CC0 和 INTTP3CC0）作为触发信号，进行 D/A 转换。
设置方法描述如下：

<1> 将 PM1n 位设置为 1（输入模式）。
<2> 将 DA0M.DA0MDn 位设置为 1（实时输出模式）。
<3> 将要输出到 ANOn 引脚的模拟电压值写入 DA0CSn 寄存器 。
<4> 设置 DA0M.DA0CEn 位为 1 （D/A 转换使能）。
以上步骤<2> 至步骤 <4> 为初始化设置。
<5> 操作 TMP2 和 TMP3。
<6> 当产生 INTTP2CC0 或 INTTP3CC0 信号时，D/A 转换的模拟电压值从 ANOn 引脚输出。
在下一次产生 INTTP2CC0 或 INTTP3CC0 信号之前，将待输出的模拟电压值设置到 DA0CSn 寄存器。
<7> 此后，在每次产生 INTTP2CC0 或 INTTP3CC0 信号时，设置到 DA0CSn 寄存器的值会从 ANOn 引脚输出。

备注 **1.** 步骤<6>之前，ANO0 和 ANO1 引脚的输出值不确定。
2. 关于 ANO0 和 ANO1 引脚在 IDLE1，IDLE2，HALT 和 STOP 模式下的输出值， 参见 **第二十四章 待机功能**。
3. n = 0, 1

15.4.3 注意事项

使用 V850ES/JG3-L 的 D/A 转换器时，请遵循以下注意事项：

(1) 将端口引脚设置为输入模式（PM1n 位 = 1）。

(2) 在 D/A 转换期间，不要读取或写入 P1 端口。

（3）当使用 P10/AN00 和 P11/AN01 引脚之一作为 I/O 端口引脚，而另一个用作 D/A 输出引脚时。在 D/A 输出期间，如果端口 I/O 电平不改变，则在应用系统中可以这样使用。

（4）实时输出模式下，通过软件来避免写入 DA0CSn 寄存器的操作与触发信号输出之间的冲突，正在服务选中的触发信号时，如果写入 DA0CSn 寄存器，则会发生冲突。

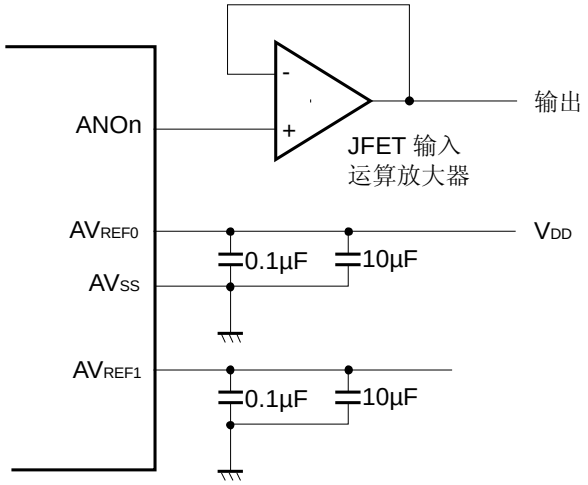
（5）请确保 $AV_{REF1} \leq V_{DD}$ 且 AV_{REF1} = 2.7 至 3.6 V。如果超出该范围，则操作无法保证。

（6）在给 AV_{REF1} 上下电的同时对 AV_{REF0} 供电掉电。

（7）因为 D/A 转换器的输出阻抗高，所以不能从 ANOn 引脚提供电流输出。如果连接 2 MΩ或更小的电阻，需在电阻和 ANOn 引脚之间插入一个 JFET(结型场效应晶体管)输入型运算放大器。

备注 n = 0, 1

图 15-2. 外部引脚连接示例



注意事项 此处展示的图仅作参考使用。请在全面评估其适合性之后再使用。

（8）因为在 STOP 模式下 D/A 转换器停止操作，ANO0 和 ANO1 引脚进入高阻抗状态，因而可以降低功耗。
但是，在 IDLE1，IDLE2 或副时钟工作模式下，D/A 转换操作继续。为了降低功耗，将 DA0M.DA0CEn 位清为 0。

第十六章 异步串行接口A (UARTA)

V850ES/JG3-L 具有 6 路通道 UARTA。

16.1 特性

- 片上专用波特率发生器
- 传输速率： 300 bps 至 625 kbps（使用专用波特率发生器）
- 全双工通信
- 双缓冲配置　　内部 UARTAn 接收数据寄存器 (UAnRX)
  内部 UARTAn 发送数据寄存器 (UAnTX)
- 接收错误检测功能
  - 奇偶错误
  - 帧错误
  - 溢出错误
- 中断源： 2
  - 接收完成中断（INTUAnR）：　在接收使能状态下，串行传输完成后，当接收数据从接收移位寄存器传送到接收数据寄存器时，产生该中断。
  - 发送使能中断（INTUAnT）：　在发送使能状态下，当数据从发送数据寄存器传送到发送移位寄存器时，产生该中断（可以支持连续发送）。
- 字符长度： 7，9 位
- 奇偶校验功能： 奇校验、偶校验、0 校验、无校验
- 发送停止位： 1，2 位
- 可选择 MSB/LSB 先行传输
- 内部数字噪声滤波器
- 支持发送/接收数据的反相输入/输出
- 在 LIN（局域互联网）通信格式下 SBF (同步间隔场)发送/接收
  - SBF 发送时，可选 13 至 20 位
  - SBF 接收时，可以识别 11 位或更多位
  - 提供 SBF 接收标志

备注　　n = 0 至 5

16.2 配置

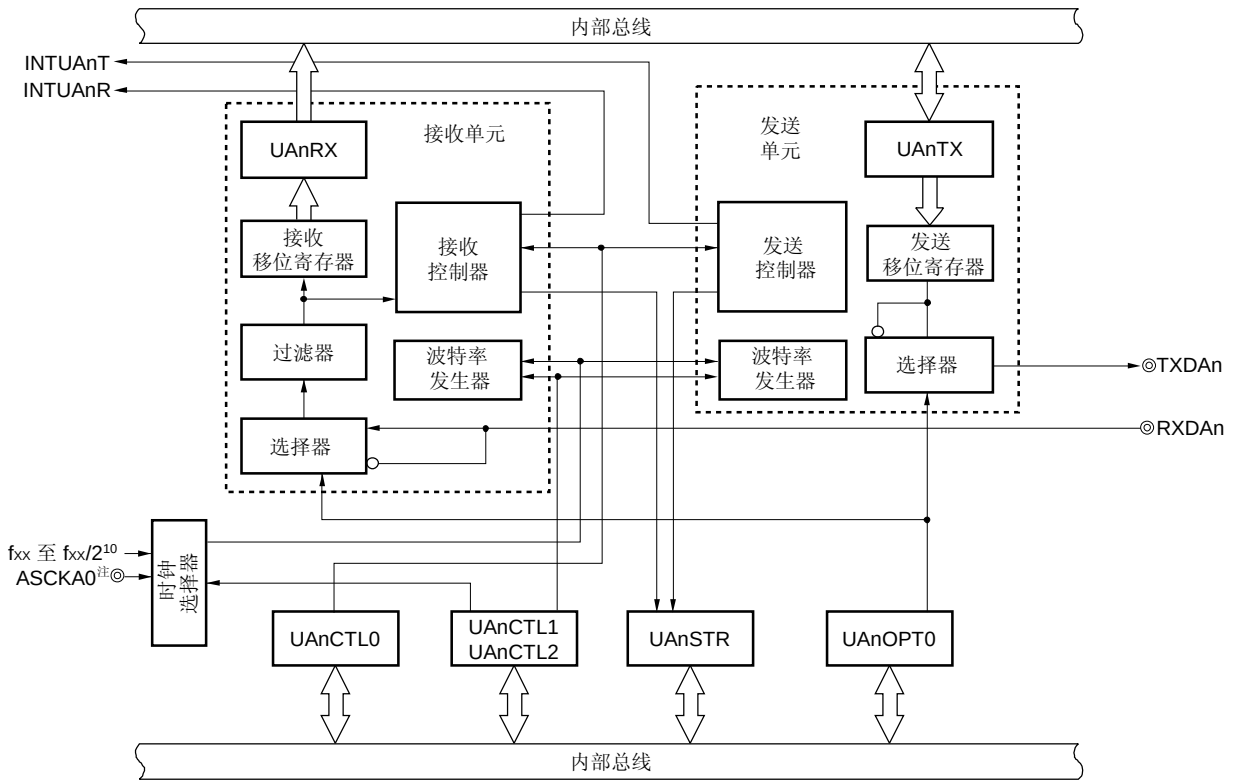
UARTAn 包括下列硬件：

表 16-1. UARTAn 的配置

项目	配置
寄存器	UARTAn 控制寄存器 0 （UAnCTL0） UARTAn 控制寄存器 1 （UAnCTL1） UARTAn 控制寄存器 2 （UAnCTL2） UARTAn 选项控制寄存器 0 （UAnOPT0） UARTAn 状态寄存器 （UAnSTR） UARTAn 接收移位寄存器 UARTAn 接收数据寄存器（UAnRX） UARTAn 发送移位寄存器 UARTAn 发送数据寄存器（UAnTX）

UARTAn 的功能框图如下所示：

图 16-1. UARTAn 的功能框图



注 仅限 UARTA0

备注 **1.** n = 0 至 5

2. 关于波特率发生器的配置，参见**图 16-17**。

(1) UARTAn 控制寄存器 0 (UAnCTL0)

UAnCTL0 寄存器是 8 位寄存器，用于指定 UARTAn 的操作。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

UAnCTL1 寄存器是 8 位寄存器，用于选择 UARTAn 的基准时钟(f_{UCLK})。

(3) UARTAn 控制寄存器 2 (UAnCTL2)

UAnCTL2 寄存器是 8 位寄存器，与 UAnCTL1 寄存器共同使用，为 UARTAn 产生所需的波特率。

(4) UARTAn 选项控制寄存器 0 (UAnOPT0)

UAnOPT0 寄存器是 8 位寄存器，用于控制在 LIN 通信格式下的 SBF 发送/接收，或控制 UARTAn 发送/接收信号的电平。

(5) UARTAn 状态寄存器 (UAnSTR)

UAnSTRn 寄存器由指示接收错误的标志位组成。当发生接收错误时，相应接收错误标志被置位（1）。

(6) UARTAn 接收移位寄存器

本寄存器是移位寄存器，用于将输入至 RXDAn 引脚的串行数据转换为并行数据。接收到 1 帧数据并检测到停止位时，就将接收数据传输至 UAnRX 寄存器。

不能直接操作该寄存器。

(7) UARTAn 接收数据寄存器 (UAnRX)

UAnRX 是 8 位缓冲寄存器，用于保存接收数据。

在接收使能状态下，当 1 帧移入操作完毕时，接收数据同步从 UARTAn 接收移位寄存器传送到 UAnRX 寄存器。

传送到 UAnRX 寄存器也会引发接收完成中断请求信号（INTUAnR）。

(8) UARTAn 发送移位寄存器

该移位寄存器用于将来自 UAnTX 寄存器的并行数据转换为串行数据。

当一帧的数据从 UAnTX 寄存器传送过来时，移位寄存器的数据从 TXDAn 引脚输出。

不能直接操作该寄存器。

(9) UARTAn 发送数据寄存器 (UAnTX)

UAnTX 寄存器是 8 位发送数据缓冲缓存器。当发送的数据被写入 UAnTX 寄存器后，开始发送。当数据能够写入 UAnTX 寄存器时 (当一帧数据从 UAnTX 寄存器传送到 UARTAn 发送移位寄存器时)，产生发送使能中断请求信号 (INTUAnT)。

16.2.1 各个通道的引脚功能

V850ES/JG3-L 中 UARTA 使用的 RXDAn, TXDAn 和 ASCKA0 引脚也可以用作其它功能，如表 16-2 所示。UARTA 要使用这些引脚，相关寄存器的设置描述如**表 4-15 端口引脚用作复用功能时的设置**。

表 16-2. UARTA 使用的引脚

通道	引脚编号		端口	UARTA 接收输入	UARTA 发送输出	UARTA 时钟 I/O 注	其它功能
	GC	F1					
UARTA0	26	K3	P31	RXDA0	–	–	INTP7/SIB4
	25	L3	P30	–	TXDA0	–	SOB4
	27	L4	P32	–	–	ASCKA0 注	$\overline{\text{SCKB4}}$/TIP00/TOP00
UARTA1	44	L9	P91	RXDA1	–	–	A1/KR7/SCL02
	43	H8	P90	–	TXDA1	–	A0/KR6/SDA02
UARTA2	36	H7	P39	RXDA2	–	–	SCL00
	35	H6	P38	–	TXDA2	–	SDA00
UARTA3	32	J6	P37	RXDA3	–	–	–
	31	H5	P36	–	TXDA3	–	–
UARTA4	46	J9	P93	RXDA4	–	–	A3/TIP40/TOP40
	45	K9	P92	–	TXDA4	–	A2/TIP41/TOP41
UARTA5	48	K10	P95	RXDA5	–	–	A5/TIP30/TOP30
	47	L10	P94	–	TXDA5	–	A4/TIP31/TOP31

注 ASCKA0 功能仅供 UARTA0 使用。

注意事项 除了上述复用功能之外，UART5 的 INTUA5T 中断和 TMP3 的 INTP3CC1 中断、以及 UART5 的 INTUA5R 中断和 TMP3 的 INTP3OV 中断是复用中断信号，因此，它们不能同时使用。

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

16.3 UARTA和其它串行接口的模式转换

16.3.1 UARTA0 和CSIB4 的模式转换

在 V850ES/JG3-L 中，UARTA0 和 CSIB4 是共享引脚的复用功能，因而不能同时使用。使用之前，用 PMC3, PFC3, 和 PFCE3 寄存器预先设置 UARTA0。

注意事项 如果在发送或接收期间切换这些功能，则 UARTA0 和 CSIB4 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 16-2. UARTA0 和 CSIB4 模式转换设置

复位后：0000H R/W 地址：FFFFF446H, FFFFF447H

	15	14	13	12	11	10	9	8
PMC3	0	0	0	0	0	0	PMC39	PMC38

7	6	5	4	3	2	1	0
0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

复位后：0000H R/W 地址：FFFFF466H, FFFFF467H

	15	14	13	12	11	10	9	8
PFC3	0	0	0	0	0	0	PFC39	PFC38

7	6	5	4	3	2	1	0
0	0	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

复位后：00H R/W 地址：FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3L	0	0	0	0	0	PFCE32	0	0

PMC32	PFCE32	PFC32	工作模式
0	×	×	Port I/O 模式
1	0	0	ASCKA0 模式
1	0	1	$\overline{\text{SCKB4}}$ 模式

PMC3n	PFC3n	工作模式
0	×	Port I/O 模式
1	0	UARTA0 模式
1	1	CSIB4 模式

备注 **1.** n = 0, 1
2. × = 无需理会

16.3.2 UARTA1 和 I²C02 的模式转换

在 V850ES/JG3-L 中，UARTA1 和 I²C02 是共享引脚的复用功能，因而不能同时使用。使用之前，用 PMC9, PFC9 和 PMCE9 寄存器预先设置 UARTA1。

UARTA1 和 I²C02 的模式转换描述如下。

注意事项 如果在发送或接收期间切换这些功能，则 UARTA1 和 I²C02 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 16-3. UARTA1 和 I²C02 模式转换设置

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

	15	14	13	12	11	10	9	8
PMC9	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

	15	14	13	12	11	10	9	8
PFC9	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

	15	14	13	12	11	10	9	8
PFCE9	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	工作模式
1	1	0	UARTA1 模式
1	1	1	I²C02 模式

备注 n = 0, 1

16.3.3 UARTA2 和I²C00 的模式转换

在 V850ES/JG3-L 中，UARTA2 和 I²C00 是共享引脚的复用功能，因而不能同时使用。使用之前，用 PMC3 和 PFC3 寄存器预先设置 UARTA2。

注意事项 如果在发送或接收期间切换这些功能，则 UARTA2 和 I²C00 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 16-4. UARTA2 和 I²C00 模式转换设置

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

	15	14	13	12	11	10	9	8
PMC3	0	0	0	0	0	0	PMC39	PMC38
	7	6	5	4	3	2	1	0
	0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

复位后：0000H R/W 地址： FFFFF466H, FFFFF467H

	15	14	13	12	11	10	9	8
PFC3	0	0	0	0	0	0	PFC39	PFC38
	7	6	5	4	3	2	1	0
	0	0	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

PMC3n	PFC3n	工作模式
0	×	Port I/O 模式
1	0	UARTA2 模式
1	1	I²C00 模式

备注 **1.** n = 8, 9
2. × = 无须理会

16.4 寄存器

(1) UARTAn 控制寄存器 0 (UAnCTL0)

UAnCTL0 寄存器是 8 位寄存器，用于控制 UARTAn 的串行传输操作。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 10H。

(1/2)

复位后：10H　R/W　地址：　UA0CTL0 FFFFFA00H, UA1CTL0 FFFFFA10H, UA2CTL0 FFFFFA20H, UA3CTL0 FFFFFA30H, UA4CTL0 FFFFFA40H, UA5CTL0 FFFFFA50H

	<7>	<6>	<5>	<4>	3	2	1	0
UAnCTL0	UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL

(n = 0 至 5)

UAnPWR	UARTAn 操作控制
0	禁止 UARTAn 操作 (UARTAn 异步复位)
1	允许 UARTAn 操作

UARTAn 操作由UAnPWR 位控制 。通过清除UAnPWR位为0， TXDAn 引脚输出固定为高电平 (如果UAnOPT0.UAnTDL 位 = 1，则固定为低电平)。

UAnTXE	发送操作使能
0	发送操作禁止
1	发送操作使能

- 若要开始发送，设置 UAnPWR 位为 1 然后设置 UAnTXE 位为 1。
  若要停止发送， 清除UAnTXE 位为0 然后清除 UAnPWR 位为 0。
- 若要初始化发送单元，清除UAnTXE位为0，等待两个周期的基础时钟，然后再次设置 UAnTXE 位为 1 。 否则无法执行初始化 (关于基础时钟，参见 16.7 (1) (a) 基础时钟)。
- 当UARTAn操作使能 (UAnPWR 位 = 1)且UAnTXE位被设置为1时，在经过基础时钟(f_{UCLK})的至少两个周期之后，发送被使能。

UAnRXE	接收操作使能
0	接收操作禁止
1	接收操作使能

- 若要开始接收 , 设置 UAnPWR 位为 1 然后 设置 UAnRXE 位为 1。
  若要停止接收 , 清除 UAnRXE 位为 0 然后清除 UAnPWR 位为 0。
- 若要初始化接收单元， 清除 UAnRXE 位为 0, 等待两个周期的基础时钟，然后再次设置 UAnRXE 位为 1。 否则无法执行初始化 (关于基础时钟，参见 16.7 (1) (a) 基础时钟)。
- 当UARTAn操作使能 (UAnPWR 位 = 1)且UAnRXE位被设置为1时，在经过基础时钟(f_{UCLK})的至少两个周期之后，接收被使能。如果在接收使能之前收到一个起始位，该起始位被忽略。

(2/2)

UAnDIR	传输方向选择
0	MSB先行
1	LSB先行

• 只有 UAnPWR 位 = 0 或 UAnTXE 位 = UAnRXE 位 = 0时，
该寄存器才可以重写。
• 在 LIN 格式下进行发送和接收时， 设置 UAnDIR位为1。

UAnPS1	UAnPS0	发送期间奇偶校验的选择	接收期间奇偶校验的选择
0	0	无奇偶校验输出	无奇偶校验接收
0	1	0 奇偶校验输出	0 奇偶校验接收
1	0	奇校验输出	奇校验检测
1	1	偶校验输出	偶校验检测

• 只有 UAnPWR 位 = 0或 UAnTXE 位 = UAnRXE 位 = 0时，
该寄存器才可以重写。
• 如果接收期间选择“ 0 奇偶校验接收”，则不进行奇偶校验检查 。
因此不设置 UAnSTR.UAnPE 位。
• 在 LIN 格式下进行发送和接收时， 清除 UAnPS1 和UAnPS0 位为 00。

UAnCL	1帧的发送/接收数据字符长度的说明
0	7 位
1	8 位

• 只有 UAnPWR 位 = 0 或 UAnTXE 位 = UAnRXE 位 = 0时，
该寄存器才可以重写。
• 在 LIN 格式下进行发送和接收时，设置 UAnCL位为1。

UAnSL	发送数据的停止位长度的说明
0	1 位
1	2 位

只有 UAnPWR 位 = 0 或 UAnTXE 位 = UAnRXE 位 = 0时，
该寄存器才可以重写。

备注 关于奇偶校验的详情，参见 **16.6.6 奇偶类型和操作。**

(2) UARTAn 控制寄存器 1 (UAnCTL1)

更多详情，参见 **16.7 (2) UARTAn 控制寄存器 1 (UAnCTL1)**。

(3) UARTAn 控制寄存器 2 (UAnCTL2)

更多详情，参见 **16.7 (3) UARTAn 控制寄存器 2 (UAnCTL2)**。

(4) UARTAn 选项控制寄存器 0 (UAnOPT0)

UAnOPT0 寄存器是 8 位寄存器，用于控制在 LIN 通信格式下的 SBF 发送/接收，或控制 UARTAn 发送/接收信号的电平。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 14H。

(1/2)

复位后: 14H　R/W　地址: UA0OPT0 FFFFFA03H, UA1OPT0 FFFFFA13H, UA2OPT0 FFFFFA23H, UA3OPT0 FFFFFA33H, UA4OPT0 FFFFFA43H, UA5OPT0 FFFFFA53H

	<7>	6	5	4	3	2	1	0
UAnOPT0	UAnSRF	UAnSRT	UAnSTT	UAnSLS2	UAnSLS1	UAnSLS0	UAnTDL	UAnRDL

(n = 0 至 5)

UAnSRF	SBF接收标志
0	UAnCTL0.UAnPWR 位 或UAnCTL0.UAnRXE 位 为0 时。 在SBF 接收正常结束时。
1	SBF 接收中

- LIN 通信期间，表示是否接收到 SBF (同步间隔场) 。
- 发生SBF接收错误时，UAnSRF 位保持为1，然后SBF接收重新开始。
- UAnSRF 位是只读位。

UAnSRT	SBF 接收触发
0	-
1	SBF 接收触发

- LIN 通信期间该位是SBF接收触发位，读出值总为“0”。
- 要进行SBF 接收, 需设置 UAnSRT 位 (为 1) 以允许SBF接收。
- 设置 UAnPWR 位 和UAnRXE 位为1 后设置UAnSRT 位。

UAnSTT	SBF 发送 触发
0	-
1	SBF 发送 触发

- LIN 通信期间该位是SBF 发送触发位，读出值总为“0”。
- 将该位设置为1，可以触发SBF发送。
- 设置 UAnPWR 位 和UAnRXE 位为1 后设置UAnSRT 位。

注意事项 SBF 接收（UAnSRF 位 = 1）期间，不要设置 UAnSRT 和 UAnSTT 位（1）。

(2/2)

UAnSLS2	UAnSLS1	UAnSLS0	SBF 发送长度选择
1	0	1	13-位 输出(复位默认值)
1	1	0	14-位 输出
1	1	1	15-位 输出
0	0	0	16-位 输出
0	0	1	17-位 输出
0	1	0	18-位 输出
0	1	1	19-位 输出
1	0	0	20-位 输出
UAnPWR 位 = 0 或 UAnTXE 位 = 0时，可设置该寄存器。			

UAnTDL	发送数据的电平
0	传输数据正常输出
1	传输数据反相输出
• TXDAn 引脚的输出电平可使用UAnTDL 位进行反相。 • UAnPWR 位 = 0 或UAnTXE 位 = 0时，可设置该寄存器。	

UAnRDL	接收数据电平
0	传输数据正常输入
1	传输数据反相输入
• RXDAn 引脚的输入电平可使用UAnRDL 位进行反相。 • UAnPWR 位 = 0 或 UAnRXE 位 = 0时，可设置该寄存器。	

(5) UARTAn 状态寄存器 (UAnSTR)

UAnSTR 寄存器是 8 位寄存器，显示 UARTAn 的传输状态和接收错误的内容。

该寄存器可进行字节读写或按位读写，但是， UAnTSF 位为只读位，而 UAnPE、UAnFE 和 UAnOVE 位既可读也可写。但是，这些位只能写入 0 来清除，而不能通过写 1 来置位（即便写入 1，之前的值保持不变）。

清除 UAnSTR 寄存器的条件如下所示：

表 16-3. 清除 STR 寄存器的条件

寄存器/位	清除条件
UAnSTR 寄存器	• 复位 • UAnCTL0.UAnPWR = 0
UAnTSF 位	• UAnCTL0.UAnTXE = 0
UAnPE 位、UAnFE 位、UAnOVE 位	• 写入 0 • UAnCTL0.UAnRXE = 0

复位后: 00H R/W 地址: UA0STR FFFFFA04H, UA1STR FFFFFA14H, UA2STR FFFFFA24H, UA3STR FFFFFA34H, UA4STR FFFFFA44H, UA5STR FFFFFA54H

	<7>	6	5	4	3	<2>	<1>	<0>
UAnSTR	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

(n = 0 至 5)

UAnTSF	传输状态标志
0	发送移位寄存器没有数据。 • 当UAnPWR位或UAnTXE位已经被设置为0时。 • 在传输完成之后，没有从UAnTX寄存器传来下一个数据。
1	发送移位寄存器有数据。(写入UAnTX寄存器)

进行连续发送时，UAnTSF位总是为1。
初始化发送单元时，在进行初始化前确保UAnTSF位= 0。
如果在UAnTSF位= 1时执行初始化，则发送数据无法保证。

UAnPE	奇偶校验 错误标志
0	• 已经设置UAnPWR位=0 或 UAnRXE=0。 • 已经写入 0 时。
1	接收到的奇偶位与指定的奇偶校验位不匹配。

- 通过设置UAnCTL0.UAnPS1 和 UAnCTL0.UAnPS0 位来控制UAnPE位的操作。
- 一旦UAnPE位被置位(1)，则值被保持，直到该位被清除 (0)。
- UAnPE位可读可写，但是只能写入0进行清除，而不能写入1进行置位，如果向该位写入1，其值保持不变。

UAnFE	帧错误标志
0	• 已经设置UAnPWR位=0 或 UAnRXE=0。 • 已经写入 0 时。
1	接收期间没有检测到停止位。

- 只检测接收数据停止位的第一位，而无需考虑UAnCTL0.UAnSL位的值。
- 一旦UAnFE位被置位(1)，则值被保持，直到该位被清除 (0)。
- UAnFE位可读可写，但是只能写入0进行清除，而不能写入1进行置位，如果向该位写入1，其值保持不变。

UAnOVE	溢出错误标志
0	• 已经设置UAnPWR位=0 或 UAnRXE=0。 • 已经写入 0 时。
1	接收数据已经设置到 UAnRX 寄存器 并且在接收数据被读取之前下一个接收操作已经完成。

- 发生溢出错误时，数据被丢弃，下一个接收数据不会被写入接收缓冲器。
- 一旦UAnOVE位被置位(1)，则值被保持，直到该位被清除 (0)。
- UAnOVE位可读可写，但是只能写入0进行清除，而不能写入1进行置位，如果向该位写入1，其值保持不变。

(6) UARTAn 接收数据寄存器 (UAnRX)

UAnRX 寄存器是 8 位缓冲寄存器，用于存储由接收移位寄存器转换而来的并行数据。

1 帧数据接收完成时，将存储于接收移位寄存器的数据传送到 UAnRX 寄存器。

LSB 接收期间，数据长度指定为 7 位时，接收数据传输至 UAnRX 寄存器的第 6 位至第 0 位，MSB 位总为 0。

MSB 接收期间，将接收数据传输到 UAnRX 的第 7 位至第 1 位，且 LSB 位总为 0。

发生溢出（UAnOVE）错误时，此时的接收数据不会被传送到 UAnRX 寄存器，而是被丢弃。

该寄存器是只读寄存器，按字节读取。

除了复位输入之外，将 UAnCTL0.UAnPWR 位清除为 0，UAnRX 寄存器被置为 FFH。

复位后： FFH R 地址： UA0RX FFFFFA06H, UA1RX FFFFFA16H,
UA2RX FFFFFA26H, UA3RX FFFFFA36H,
UA4RX FFFFFA46H, UA5RX FFFFFA56H

	7	6	5	4	3	2	1	0
UAnRX								

(n = 0 至 5)

(7) UARTAn 发送数据寄存器 (UAnTX)

UAnTX 寄存器是 8 位寄存器，用于设置发送数据。

在发送使能(UAnCTL0.UAnTXE 位 = 1)时向 UAnTX 寄存器写入发送数据，会触发发送。当 UAnTX 寄存器数据向 UARTAn 发送移位寄存器的传输完成时，产生发送使能中断请求信号 (INTUAnT)。

7 位数据按照 LSB 先行方式发送时，数据被传输到 UAnTX 寄存器的第 6 位至第 0 位。数据按照 MSB 先行方式发送时，数据被传输到 UAnTX 寄存器的第 7 位至第 1 位。

该寄存器可进行字节读取或写入。

系统复位后，该寄存器被设为 FFH。

注意事项 在发送使能(UAnPWR 位 = 1 且 UAnTXE 位 = 1)时向 UAnTX 寄存器写入发送数据，会触发发送。如果写入的值与写入前的值相同，则相同的数据被发送两次。要在前一个数据处理期间写入新的发送数据，应等待直到产生发送使能中断请求信号(INTUAnT)。

即使在发送禁止(UAnPWR 位 = 0 或 UAnTXE 位 = 0)时向 UAnTX 寄存器写入发送数据，在发送使能后，发送也不会启动。

复位后： FFH R/W 地址： UA0TX FFFFFA07H, UA1TX FFFFFA17H,
UA2TX FFFFFA27H, UA3TX FFFFFA37H,
UA4TX FFFFFA47H, UA5TX FFFFFA57H

	7	6	5	4	3	2	1	0
UAnTX (n = 0 至 5)								

16.5 中断请求信号

UARTAn 会产生以下两种中断请求信号：

- 接收完成中断请求信号(INTUAnR)
- 发送使能中断请求信号 (INTUAnT)

这两种中断请求信号的默认优先级为接收完成中断请求信号优先，然后为发送使能中断请求信号。

表 16-4. 中断及其默认优先级

中断请求信号	优先级
接收完成	高
发送使能	低

(1) 接收完成中断请求信号(INTUAnR)

接收使能状态下，存储在接收移位寄存器中的数据传输至 UAnRX 寄存器时，产生接收完成中断请求信号。

发生接收错误时，接收完成中断请求信号也会继续输出。因此，响应接收完成中断请求信号并读取数据时，读取 UAnSTR 寄存器并检查接收结果是否错误。

在接收禁止状态下，不产生接收完成中断请求信号。

(2) 发送使能中断请求信号 (INTUAnT)

在发送允许状态下，当发送数据从 UAnTX 寄存器传输至 UARTAn 发送移位寄存器，产生发送使能中断请求信号。

16.6 操作

16.6.1 数据格式

如图 16-5 所示，一个发送/接收数据帧的数据由起始位、字符位、校验位以及停止位组成。

使用 UAnCTL0 寄存器指定一个数据帧内字符位长度、校验段，指定停止位长度，以及 MSB/LSB 先行传输方式。

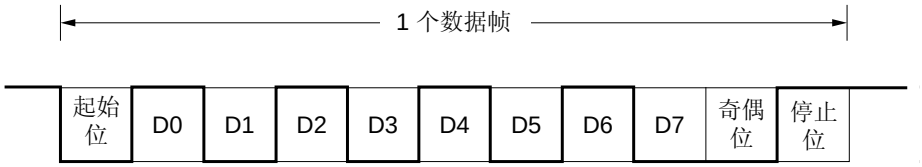
UAnOPT0.UAnTDL 位用于指定要通过 TXDAn 引脚发送的正常输出/反相输出。

UAnOPT0.UAnRDL 位用于指定要通过 RXDAn 引脚接收的正常输出/反相输出。

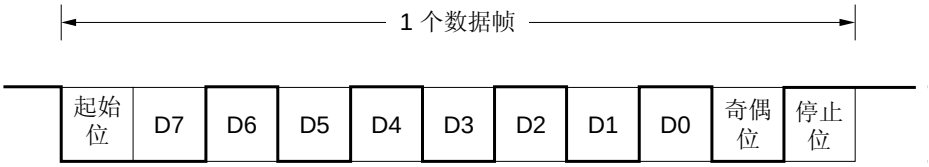
- 起始位.....1 位
- 字符位.....7 位/8 位
- 奇偶校验位 .....奇检验/偶校验/0 校验/无校验
- 停止位.....1 位/2 位
- 输入逻辑.....正常输出/反相输出
- 输出逻辑.....正常输出/反相输出
- 通信方向.....MSB/LSB

图 16-5. UARTA 发送/接收数据格式

(a) 8 位数据长度，LSB，偶校验， 一个停止位，传输数据：55H

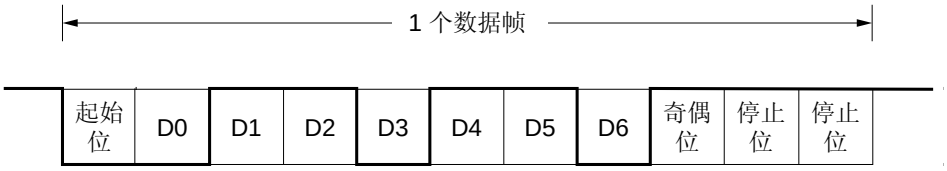


(b) 8 位数据长度，MSB，偶校验，一个停止位，传输数据：55H

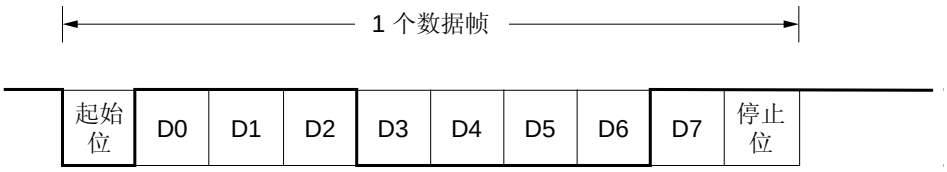


(c) 8 位数据长度，MSB，偶校验， 一个停止位，传输数据：55H，反相发送/接收数据

(d) 7 位数据长度，LSB，奇校验， 2 个停止位，传输数据：36H



(e) 8 位数据长度，LSB，无校验， 一个停止位，传输数据：87H



16.6.2 UART 发送

将 UAnCTL0.UAnPWR 位和 UAnCTL0.UAnTXE 位设置为 1，进入发送使能状态，将发送数据写入 UAnTX 寄存器启动发送。起始位、校验位以及停止位自动添加。

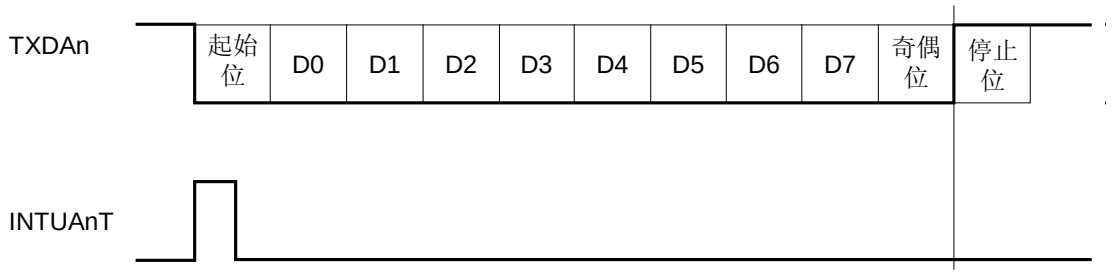
由于 UARTAn 中不提供 CTS (发送允许信号) 输入引脚，所以需要使用一个端口来检查发送目的方已经允许接收。

在发送启动时，UAnTX 寄存器内的数据被传送到 UARTAn 发送移位寄存器 。

UAnTX 寄存器的数据发送至 UARTAn 发送移位寄存器的操作完成时，产生发送使能中断请求信号（INTUAnT），之后 UARTAn 发送移位寄存器的内容由 TXDAn 引脚输出。

INTUAnT 信号产生后，允许向 UAnTX 寄存器写入下一个发送数据。

图 16-6. UART 发送



备注 LSB 先行。

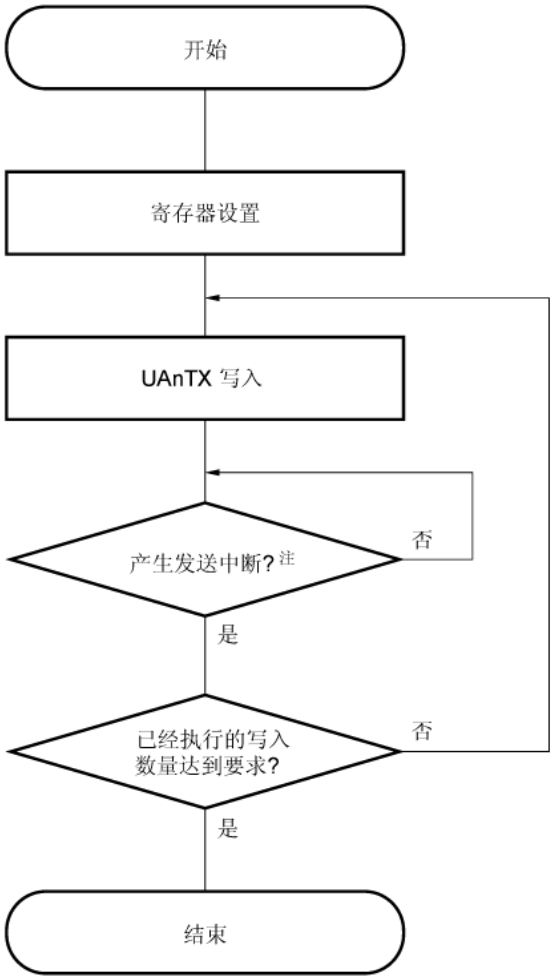
16.6.3 连续发送过程

在发送使能时向 UAnTX 寄存器写入发送数据，会触发发送。UAnTX 寄存器内的数据被传送到 UARTAn 发送移位寄存器，产生发送使能中断请求信号 (INTUAnT)，然后开始移位。在发送使能中断请求信号 (INTUAnT)产生之后，下一个数据可以写入到 UAnTX 寄存器。UARTAn 发送移位寄存器的发送时序可以由发送使能中断请求信号 (INTUAnT)来判断。

在传输过程中将下一个待发送数据写入 UAnTX 寄存器使得传输效率更高。

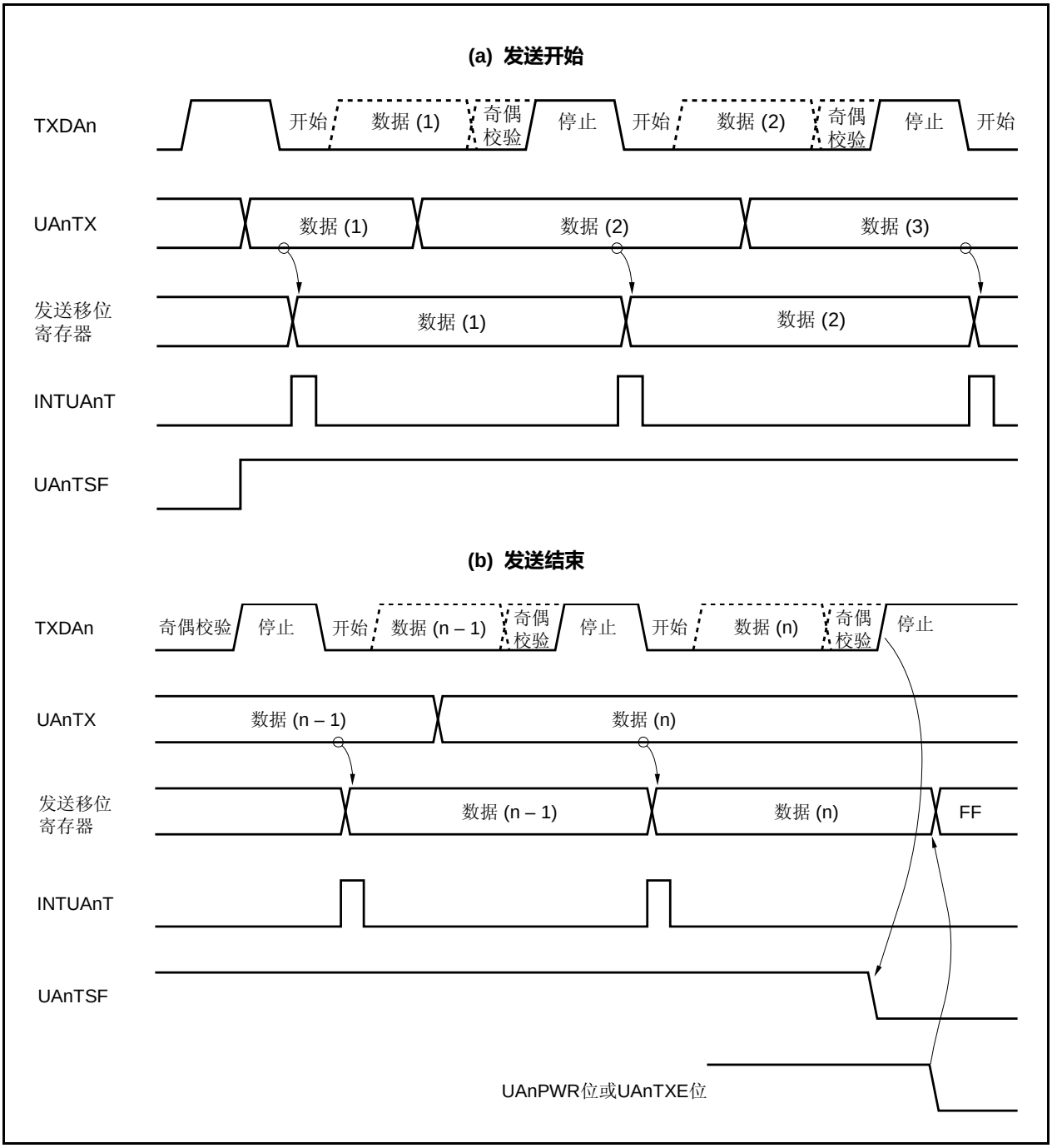
注意事项 连续发送执行期间进行发送初始化时，请确保 UAnSTR.UAnTSF 位为 0，然后再执行初始化。UAnTSF 位为 1 时进行初始化，则发送数据无法保证。

图 16-7. 连续发送处理



注 请确保在发送使能中断请求信号 (INTUAnT)产生之后读取 UAnSTR 寄存器，以检查是否发生发送错误。

图 16-8. 连续发送操作时序



16.6.4 UART 接收

首先，通过执行下列操作使能接收，并监控 RXDAn 输入以检测起始位。

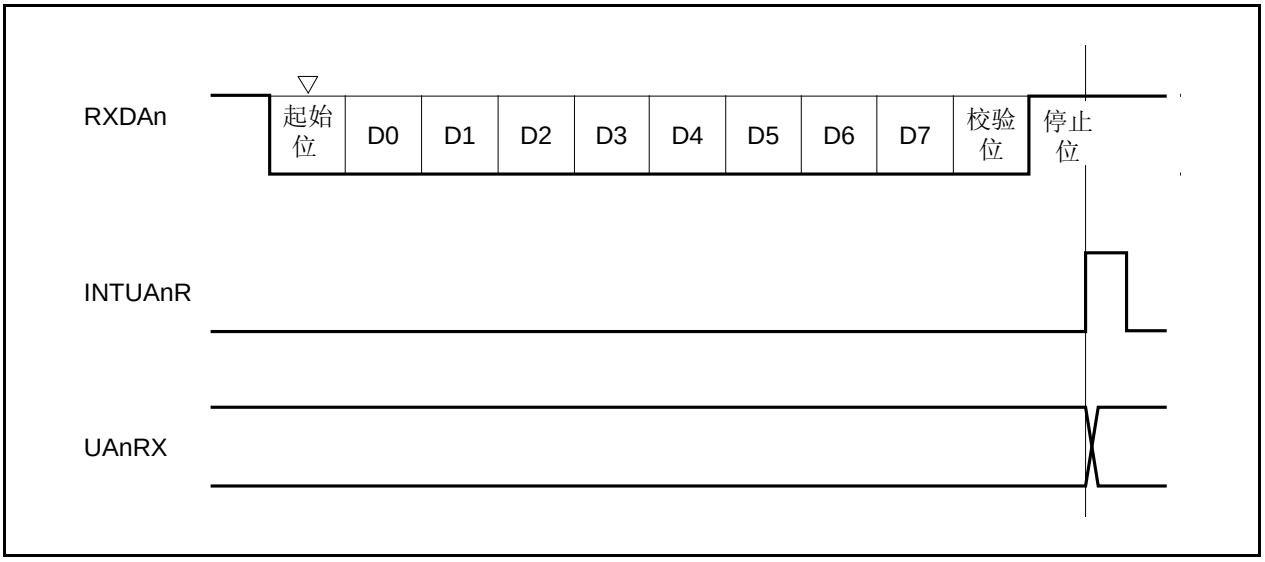
- 使用 UARTA 控制寄存器 1 (UAnCTL1)指定操作时钟。
- 使用 UARTA 控制寄存器 2 (UAnCTL2)指定波特率。
- 使用 UARTA option 控制寄存器 0 (UAnOPT0)指定输出逻辑电平。
- 使用 UARTA 控制寄存器 0 (UAnCTL0)指定通信方向、校验位、数据帧长度和停止位长度。
- 置位供电位和接收使能位 (UAnPWR = 1 和 UAnRXE = 1)。

要改变通信方向、校验位、数据帧长度和停止位长度，清除供电位 (UAnPWR = 0)或事先将发送使能位和接收使能位都清除 (UAnTXE = 0 和 UAnRXE = 0)。

使用操作时钟对 RXDAn 引脚的输入电平采样。如果检测到下降沿，则开始对 RXDAn 的输入数据进行采样。如果在检测到下降沿之后数据后半位是低电平（在图 16-9 中用 ▽ 表示），则认为是起始位。确认起始位之后，开始接收操作，按照设定的波特率，串行数据被顺序存入 UARTAn 接收移位寄存器。当接收到停止位时，会输出接收完成中断请求信号（INTUAnR）时，同时，UARTAn 接收移位寄存器的数据被传送到接收数据寄存器（UanRX）。

但是，如果发生溢出（UAnOVE 位 = 1）错误，此时接收到的数据不写入 UAnRX 寄存器，而是被丢弃。此外，即使发生奇偶错误（UAnPE 位 = 1）或帧错误（UAnFE 位 = 1），接收还会继续进行，接收数据被传送到 UAnRX 寄存器。不论发生哪种接收错误，接收完成之后都会产生 INTUAnR 中断。

图 16-9. UART 接收



注意事项 1. 即使发生接收错误，也必须读取 UAnRX 寄存器。如果不读取 UAnRX，则下一个数据接收期间发生溢出错误，并且接收错误会不断地连续发生。

2. 认为接收操作期间只有一个停止位。忽略第二个停止位。

3. 当接收完成时，在产生接收结束中断请求信号（INTUAnR）之后读取 UAnRX 寄存器，并对 UAnPWR 位或 UAnRXE 位清 0。如果在 INTUAnR 信号产生之前对 UAnPWR 位或 UAnRXE 位清 0，则无法保证对 UAnRX 寄存器值的正确读取。

4. 如果 UARTAn 的接收完成处理(INTUAnR 信号产生)与 UAnPWR 位或 UAnRXE 位设置为 0 发生冲突，则即使 UAnRX 寄存器内没有存储数据，也可能会产生 INTUAnR 信号。
为了完成接收而无需等待 INTUAnR 信号的产生，请确保将 UAnRIC 寄存器的中断请求标志(UAnRIF)清除(0)，在中断控制寄存器（UAnRIC）的中断屏蔽标志(UAnRMK)置位(1)之后，然后设置 UAnPWR 位或 UAnRXE 位为 0。

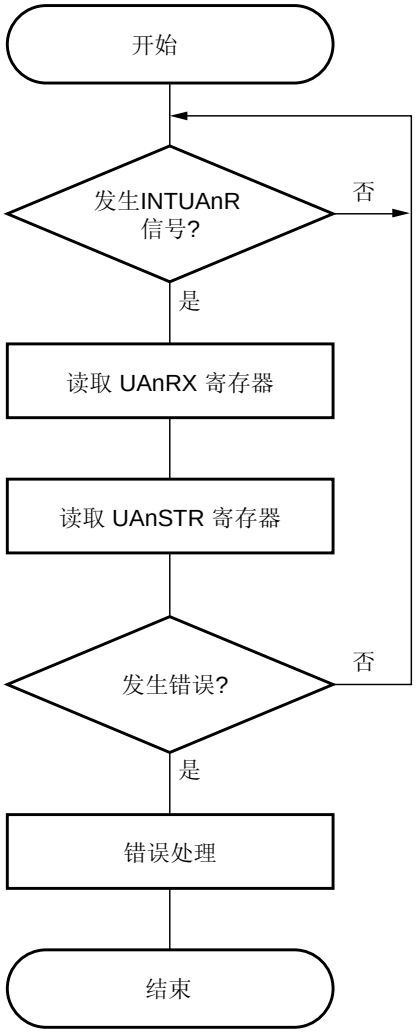
16.6.5 接收错误

接收期间可能发生的错误有 3 种类型：奇偶检验错误、帧错误以及溢出错误。数据接收结果错误标志被设置在 UAnSTR 寄存器中，当错误发生时，输出接收完成中断请求信号 (INTUAnR)。

通过读取 UAnSTR 寄存器的内容，可以确定在接收过程中发生了哪种错误。

读取完成后，写入 0 来清除接收错误标志。

图 16-10. 接收数据读取流程



注意事项 发生 INTUAnR 信号时，必须读取 UAnSTR 寄存器来检查错误。

表 16-5. 导致接收错误的原因

错误标志	接收错误	原因
UAnPE	奇偶错误	接收到的校验位和设定校验位不匹配
UAnFE	帧错误	未检测到停止位
UAnOVE	溢出错误	在数据从接收缓冲读出之前，下一个数据接收完成

发生接收错误时，根据错误类型执行以下处理步骤：

- 奇偶错误
  如果是因为例如接收线上的噪声问题导致接收到错误数据，则丢弃接收到的数据并重新发送。

- 帧错误
  接收端和发送端之间可能发生波特率错误，或者错误的检测到起始位。因为对通信格式来说这是致命错误，确认发送端的操作已经停止，对发送和接收两端各自进行初始化处理，然后重新开始通信。

- 溢出错误
  因为在读出接收数据之前，下一个数据接收完成，1 帧数据会丢弃。如果需要该数据，则需进行重新发送。

注意事项 在接收时，在下一个数据接收完成之前，必须读取 UAnSTR 寄存器，以检查是否发生错误，如果有错误发生，则执行错误处理。

16.6.6 奇偶类型和操作

奇偶位用于检测通信数据中的位错误。通常情况下，发送端和接收端应该使用相同的奇偶校验。
在使用奇校验和偶校验情况下，可以检测到奇数计数的位错误。而在 0 奇偶校验和无校验情况下，检测不到错误。

(a) 偶校验

(i) 在发送期间

发送数据之中位值为“1”的数量，包括奇偶校验位，控制其数量为偶数。奇偶位的值如下：

- 发送数据中位值为“1”的数量为奇数： 1
- 发送数据中位值为“1”的数量为偶数： 0

(ii) 在接收期间

计算接收数据中位值为“1”的数量，包括校验位，如果为奇数，则输出奇偶校验错误。

(b) 奇校验

(i) 在发送期间

与偶校验相对，发送数据中位值为“1”的数量，包括奇偶校验位，控制其数量为奇数。奇偶位的值如下：

- 发送数据中位值为“1”的数量为奇数： 0
- 发送数据中位值为“1”的数量为偶数： 1

(ii) 在接收期间

计算接收数据中位值为“1”的数量，包括校验位，如果为偶数，则输出奇偶校验错误标志。

(c) 0 奇偶校验

在发送期间，奇偶校验位总是“0”，而与发送数据无关。
在接收期间，不执行奇偶校验。因而，无论奇偶位为“0” 或 “1”，均不会产生奇偶错误。

(d) 无奇偶校验

不向发送数据附加奇偶位。
设定无奇偶位进行接收。因为无奇偶位所以没有奇偶错误发生。

注意事项 使用 LIN 功能时，将 UAnCTL0 寄存器的 UAnPS1 位和 UAnPS0 位固定为 00。

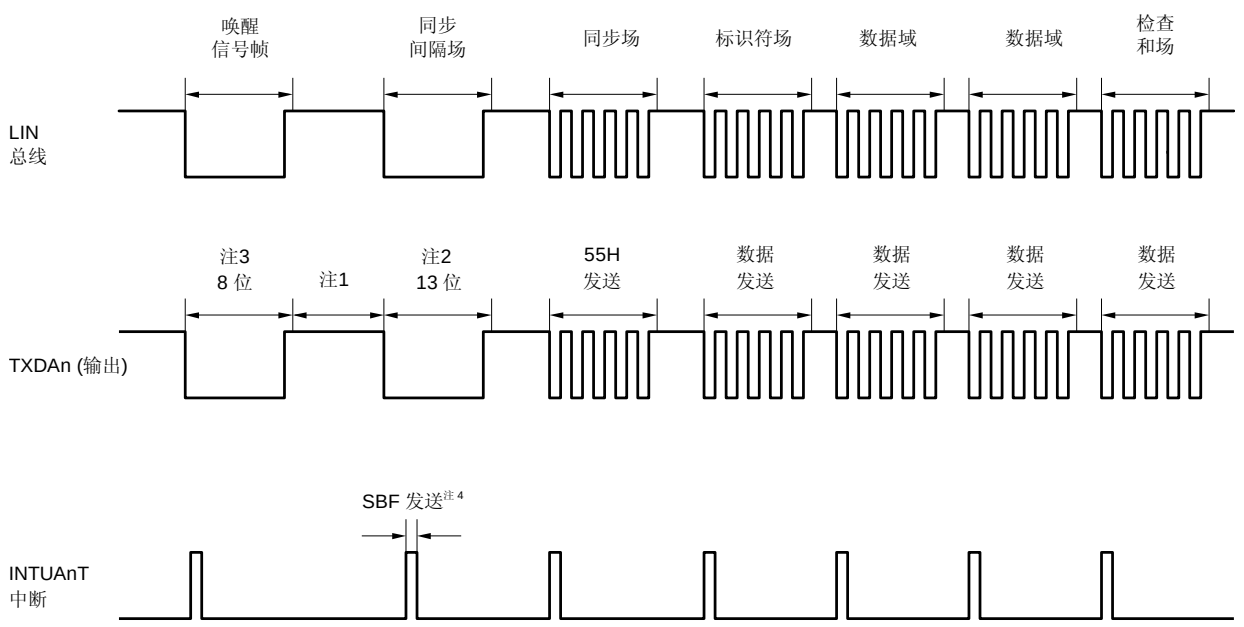
16.6.7 LIN发送/接收格式

V850ES/JG3-L 具有 SBF (同步间隔场)发送 /接收控制功能，可以支持 LIN 功能的使用。

备注 LIN 表示局域内联网，是一种低速（1 至 20 kbps）串行通信协议，有助于降低汽车网络的成本。
LIN 通信为单主机通信，一台主机最多可以连接 15 台从机。
LIN 从机一般是控制开关、传动装置和传感器，且这些装置可通过 LIN 网络连接到 LIN 主机上。
通常，LIN 主机接入诸如 CAN（控制器区域网络）之类的网络。
另外，LIN 总线使用单线方式且通过遵循 ISO9141 规范的收发器连接至各个节点。
LIN 协议中，主机以某一波特率信息发送一个数据帧，从机将其接收并校正波特率错误。因此，从机波特率错误在±15%范围内，通信可以正常进行。

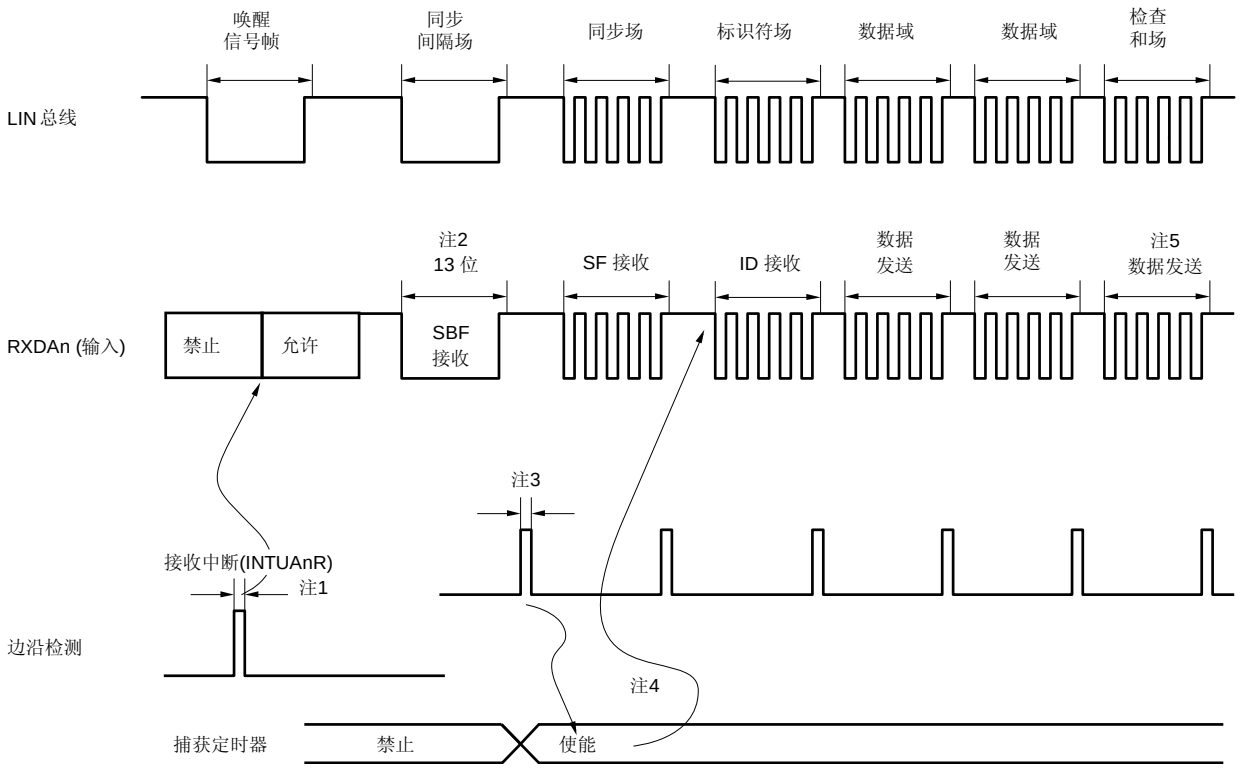
图 16-11 和 16-12 表明 LIN 的发送和接收的处理。

图 16-11. LIN 发送格式



注
1. 各场之间的间隔由软件控制。
2. SBF 输出通过硬件执行。输出宽度即位长度，由 UAnOPT0.UAnSBL2 至 UAnOPT0.UAnSBL0 位设置。如果需要调整输出宽度，可使用 UAnCTLn.UAnBRS7 至 UAnCTLn.UAnBRS0 位进行调整。
3. 8 位模式下，使用 80H 替代唤醒信号帧。
4. 发送使能中断请求信号（INTUAnT）在每次发送开始时输出。INTUAnT 信号同样在每次 SBF 发送开始时输出。

图 16-12. LIN 接收格式



注

1. 引脚边沿检测器检测到唤醒信号，使能 UARTAn，设置 SBF 接收模式。
2. 接收操作会持续到检测到停止位。检测到 11 位或更多位的 SBF 接收，则判断认为正常 SBF 接收结束，同时输出中断信号。检测到的 SBF 接收少于 11 位，则判断为 SBF 接收错误，没有中断信号输出，并且返回 SBF 接收模式。
3. 如果 SBF 接收正常结束，则输出中断请求信号。通过 SBF 接收完成中断来使能定时器。此外，关于 UAnSTR.UAnOVE、UAnSTR.UAnPE 和 UAnSTR.UAnFE 位的错误不作检测，并且不执行 UART 通信错误检测处理，UARTAn 接收移位寄存器的数据不向 UAnRX 寄存器传输。UARTAn 接收移位寄存器保持初始值 FFH。
4. RXDAn 引脚连接至定时器的 T1（捕获输入），计算得出传输速率，波特率误差。通过校正的波特率误差设置 UAnCTL2 寄存器，UARTA 使能变低再次置位，状态变为接收状态。
5. 通过软件可以识别校验和场。在校验和场的接收之后执行 UARTAn 初始化，通过软件重新指定 SBF 接收模式的处理。

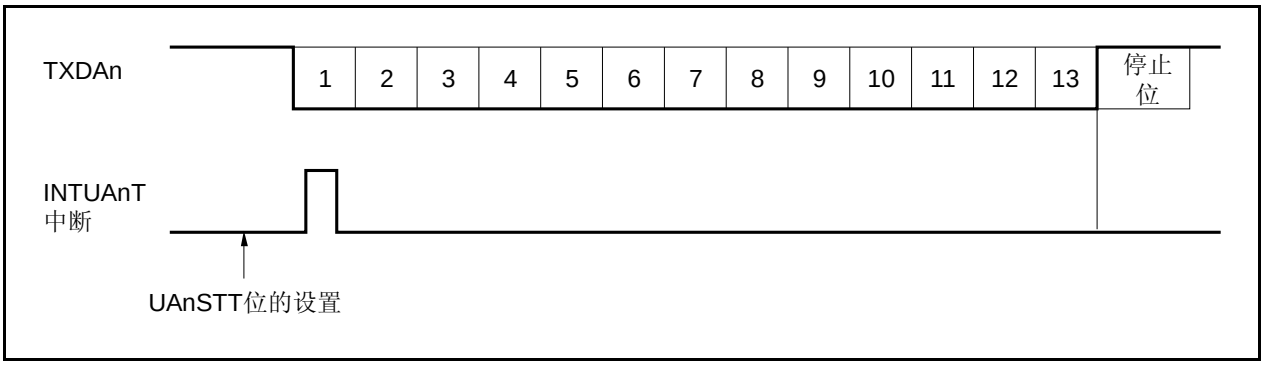
16.6.8 SBF发送

当 UAnCTL0.UAnPWR 位和 UAnCTL0.UAnTXE 位都为 1 时，进入发送使能状态，通过设置 SBF 发送触发信号（UAnOPT0.UAnSTT 位）为 1，开始 SBF 发送。

之后，输出由 UAnOPT0.UAnSLS2 至 UAnOPT0.UAnSLS0 位指定的 13 位至 20 位宽度的低电平信号。SBF 发送开始时，产生发送使能中断请求信号（INTUAnT）。在 SBF 发送结束后，UAnSTT 位自动清零。之后，恢复 UART 发送模式。

暂停发送，直到下一个待发送数据写入 UAnTX 寄存器，或者直到 SBF 发送触发信号（UAnSTT 位）被置位。

图 16-13. SBF 发送示例



16.6.9 SBF接收

设置 UAnCTL0.UAnPWR 位为 1，然后设置 UAnCTL0.UAnRXE 位为 1，进入接收使能状态。

将 SBF 接收触发（UAnOPT0.UAnSTR 位）置 1，来进入 SBF 接收等待状态。

SBF 接收等待状态下，RXDAn 引脚被监控并且开始检测起始位。和 UART 接收等待状态类似。

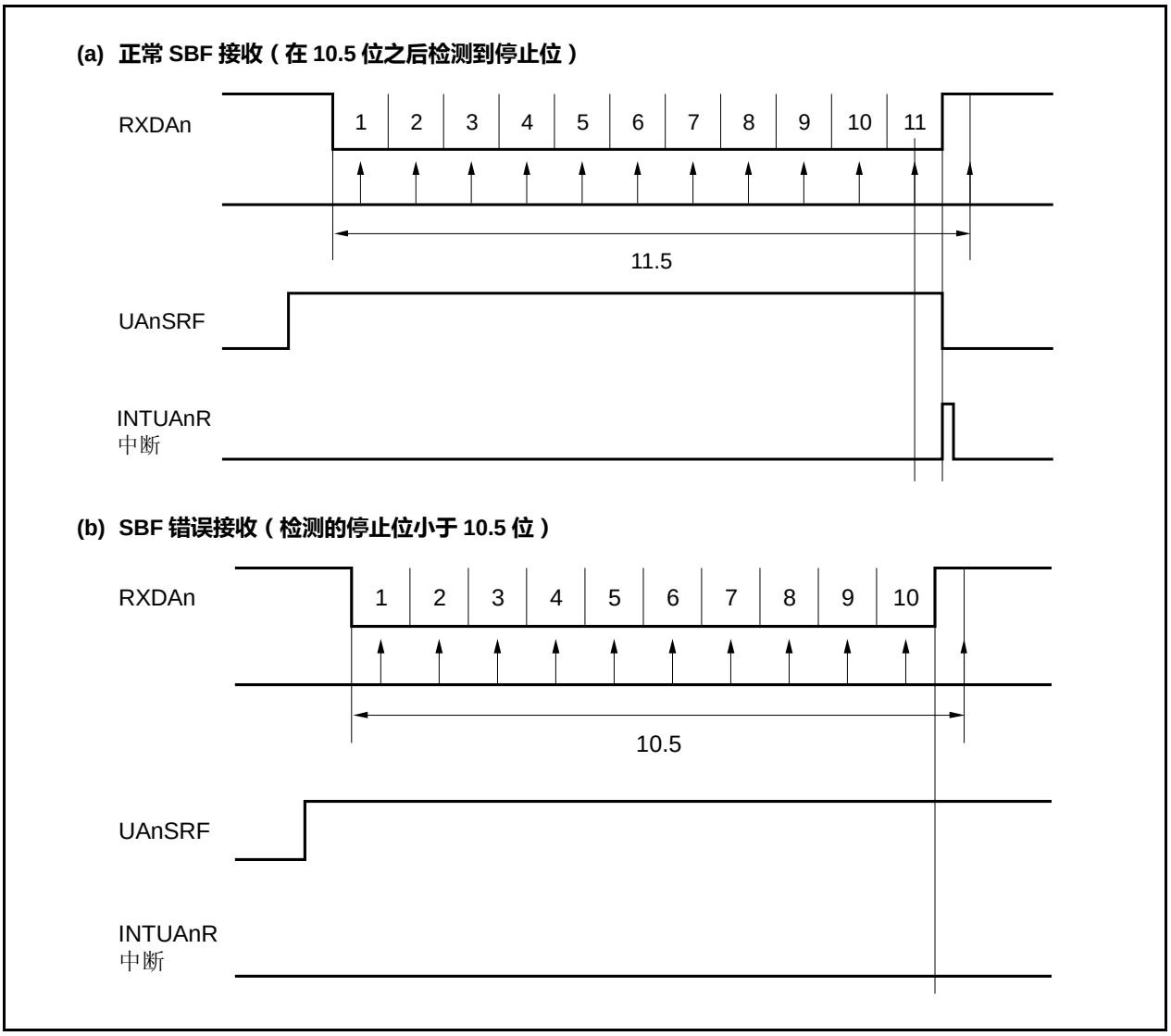
在检测到起始位之后，根据设置的波特率开始接收并且内部计数器开始向上计数。

接收到一个停止位时，如果 SBF 宽度为 11 位或更多位，则判断为正常操作，输出接收完成中断请求信号（INTUAnR）。UAnOPT0.UAnSRF 位自动清除，SBF 接收结束。对 UAnSTR.UAnOVE、UAnSTR.UAnPE 和 UAnSTR.UAnFE 位的错误不作检测，不执行 UART 通信错误检测处理，不执行 UARTAn 接收移位以及向 UAnRX 寄存器的数据传输，UARTAn 接收移位寄存器保持初始值 FFH。如果 SBF 宽度为 10 位或更少位，则接收以错误终止，不产生中断，并返回 SBF 接收模式。此时，UAnSRF 位不被清除。

注意事项 1. 如果数据接收期间 SBF 被发送，会发生一个帧错误。

2. 在 SBF 接收（UAnSRF = 1）期间，不要将 SBF 接收触发位（UAnSRT）和 SBF 发送触发位（UAnSTT）设置为 1。

图 16-14. SBF 接收



16.6.10 接收数据噪声过滤器

该滤波器使用专用波特率发生器提供的基准时钟，通过 RXDAn 引脚对接收到的信号进行采样。

如果读到两次相同的采样值，则匹配检测器输出改变，并且被采样的 RXDAn 信号作为输入数据。所以不超过 2 个时钟宽度的数据被判定为噪声，不会被送交给内部电路(参见 **图 16-16**)。有关基准时钟，参见 16.7 (1) (a) 基准时钟 。

另外，由于电路如图 16-15 所示，所以，接收操作的处理过程与外部信号状态相比，延迟了 3 个时钟周期。

图 16-15. 噪声过滤电路

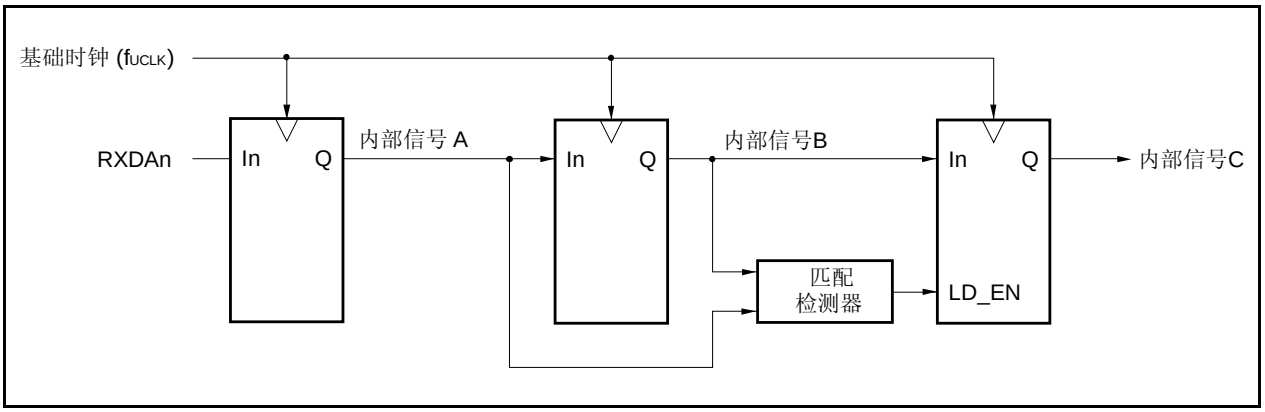
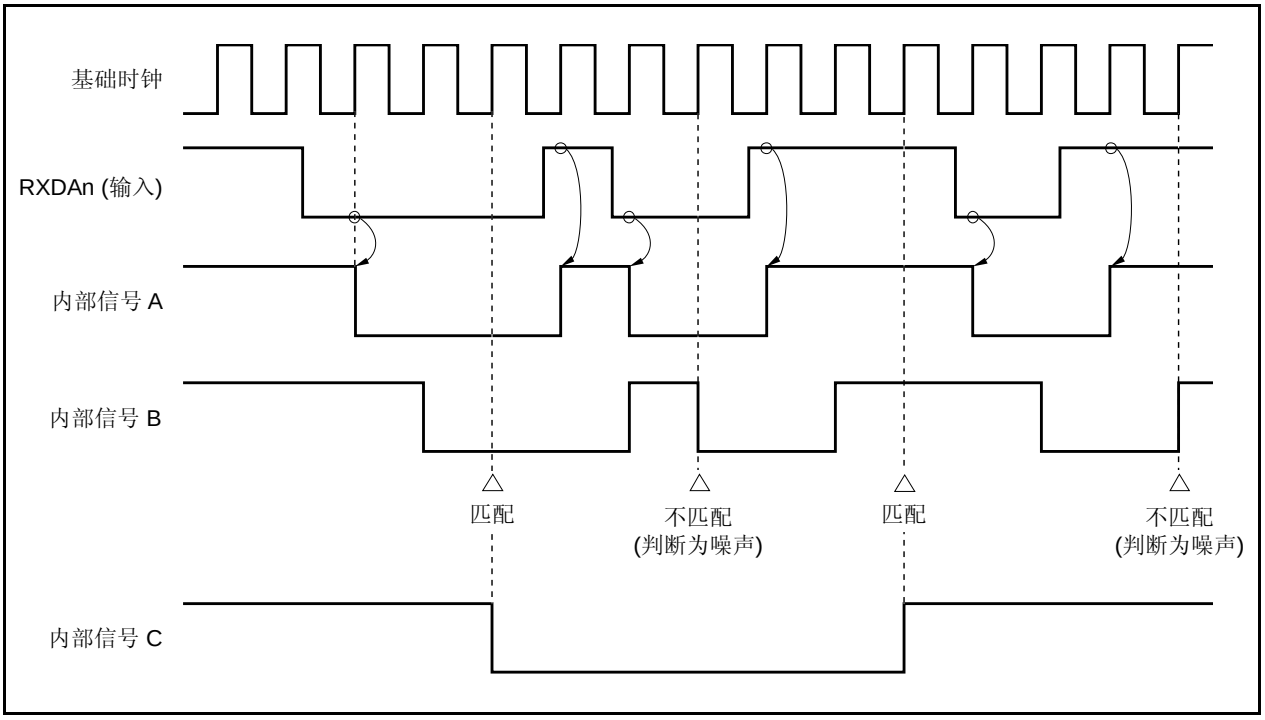


图 16-16. RXDAn 信号判定为噪声的时序



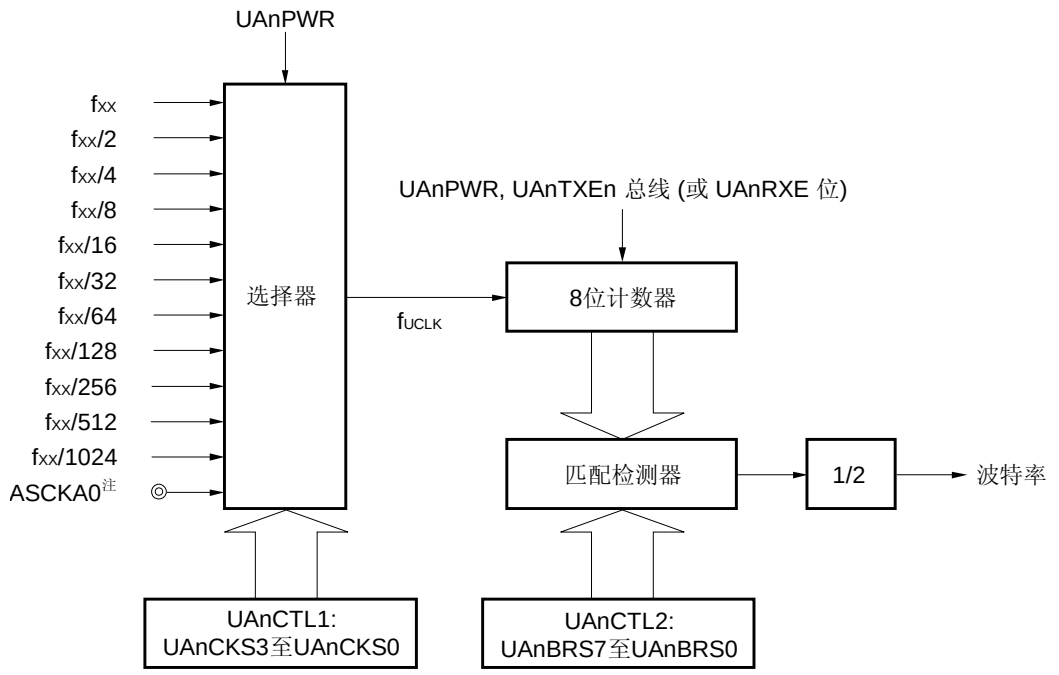
16.7 专用波特率发生器

专用波特率发生器包括一个时钟源选择器和一个 8 位可编程计数器模块，在 UARTAn 发送/接收期间产生串行时钟。关于串行时钟，对各个通道可以选择专用波特率发生器输出。

一个 8 位计数器用于发送，另一个用于接收。

(1) 波特率发生器的配置

图 16-17. 波特率发生器的配置



注 只有 UARTA0 有效；禁止设置 UARTA1 至 UARTA5。

备注 **1.** n = 0 至 5
2. f_{XX}： 主时钟频率
3. f_{UCLK}： 基准时钟频率
4. f_{UCLK}/k： 串行时钟（k：BRGCn 寄存器的值）

(a) 基准时钟

当 UAnCTL0.UAnPWR 位为 1 时，由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择的时钟被提供给 8 位计数器。该时钟称为基准时钟(f_{UCLK})。

(b) 串行时钟的产生

通过设置 UAnCTL1 寄存器和 UAnCTL2（n = 0 至 2）寄存器，来产生一个串行时钟。
基准时钟由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择。
8 位计数器的分频值可以使用 UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位进行设置。
波特率时钟由串行时钟 2 分频产生。

(2) UARTAn 控制寄存器 1 (UAnCTL1)

UAnCTL1 寄存器是 8 位寄存器，用于选择 UARTAn 基准时钟。

该寄存器可进行字节读取或写入。

系统复位后，该寄存器被设为 00H。

注意事项 在改写 UAnCTL1 寄存器之前，将 UAnCTL0.UAnPWR 位清 0。

复位后： 00H R/W 地址： UA0CTL1 FFFFFA01H, UA1CTL1 FFFFFA11H,
UA2CTL1 FFFFFA21H, UA3CTL1 FFFFFA31H,
UA4CTL1 FFFFFA41H, UA5CTL1 FFFFFA51H

	7	6	5	4	3	2	1	0
UAnCTL1	0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0

(n = 0 至 5)

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	基准时钟 (f_{UCLK}) 选择
0	0	0	0	fxx
0	0	0	1	fxx/2
0	0	1	0	fxx/4
0	0	1	1	fxx/8
0	1	0	0	fxx/16
0	1	0	1	fxx/32
0	1	1	0	fxx/64
0	1	1	1	fxx/128
1	0	0	0	fxx/256
1	0	0	1	fxx/512
1	0	1	0	fxx/1,024
1	0	1	1	外部时钟[注] (ASCKA0 引脚)
其它				禁止设置

注 只有 UARTA0 有效；禁止设置 UARTA1 至 UARTA5。

备注 fxx： 主时钟频率

(3) UARTAn 控制寄存器 2 (UAnCTL2)

UAnCTL2 寄存器是 8 位寄存器，用于选择 UARTAn 的波特率(串行传输速度)。

波特率时钟的产生是对串行时钟 2 分频得到，由该寄存器指定。

该寄存器可进行字节读取或写入。

系统复位后，该寄存器被设为 FFH。

注意事项 在改写 UAnCTL2 寄存器之前，或将 UAnCTL0.UAnPWR 位清除为 0，或将 UAnTXE 和 UAnRXE 位清除为 0。

复位后： FFH R/W 地址： UA0CTL2 FFFFFA02H, UA1CTL2 FFFFFA12H,
UA2CTL2 FFFFFA22H, UA3CTL2 FFFFFA32H,
UA4CTL2 FFFFFA42H, UA5CTL2 FFFFFA52H

UAnCTL2 (n = 0 至 5)

7	6	5	4	3	2	1	0
UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0

UAn BRS7	UAn BRS6	UAn BRS5	UAn BRS4	UAn BRS3	UAn BRS2	UAn BRS1	UAn BRS0	默认 (k)	串行 时钟
0	0	0	0	0	0	×	×	×	禁止设置
0	0	0	0	0	1	0	0	4	$f_{UCLK}/4$
0	0	0	0	0	1	0	1	5	$f_{UCLK}/5$
0	0	0	0	0	1	1	0	6	$f_{UCLK}/6$
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	$f_{UCLK}/252$
1	1	1	1	1	1	0	1	253	$f_{UCLK}/253$
1	1	1	1	1	1	1	0	254	$f_{UCLK}/254$
1	1	1	1	1	1	1	1	255	$f_{UCLK}/255$

备注 f_{UCLK}：由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择的时钟频率

(4) 波特率

波特率误差由下列公式计算。

$$波特率 = \frac{f_{UCLK}}{2 \times k} [bps]$$

使用内部时钟时，按以下公式（用 ASCKA0 引脚作为 UARTA0 的时钟时，按上式计算）

$$波特率 = \frac{f_{XX}}{2^{m+1} \times k} [bps]$$

备注 f_{UCLK} =由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位选择的基准时钟频率

f_{XX}： 主时钟频率

m =由 UAnCTL1.UAnCKS3 至 UAnCTL1.UAnCKS0 位设置的值（m = 0 至 10）

k = 由 UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位设置的值（k = 4 至 255）

波特率误差由下列公式计算：

$$误差(\%) = \left(\frac{实际波特率（有错误的波特率）}{目标波特率 （正确的波特率）} - 1 \right) \times 100 [\%]$$

$$= \left(\frac{f_{UCLK}}{2 \times k \times 目标波特率} - 1 \right) \times 100 [\%]$$

使用内部时钟时，按以下公式（用 ASCKA0 引脚输入作为 UARTA0 的时钟，按上式计算波特率误差）

$$误差(\%) = \left(\frac{f_{XX}}{2^{m+1} \times k \times 目标波特率} - 1 \right) \times 100 [\%]$$

注意事项 1. 发送过程中，波特率误差必须在接收端的误差容限之内。

2. 接收时的波特率误差满足一定范围，该范围参见“（5）接收中允许的波特率误差范围”。

要设置波特率，按以下算式计算，设置 UAnCTL1 和 UAnCTL2 寄存器（使用内部时钟时）。

<1> 设置 k 等于 fxx/(2 × 目标波特率)，m 等于 0。
<2> 如果 k 等于 256 或更大（k ≥ 256），将 k 减小一半（k/2），m 加 1（m + 1）。
<3> 重复第<2>步，直到 k 小于 256 (k < 256)。
<4> 对 k 的第一位小数四舍五入，得到最接近的整数。
如果四舍五入后 k 为 256，将 k 减半并对 m 递增 1(m + 1)获得 k = 128。
<5> 将 m 的值设置到 UAnCTL1 寄存器，k 的值设置到 UAnCTL2 寄存器。

示例： 如果 f_{XX} = 20 MHz 而目标波特率 = 153,600 bps
<1> k = 20,000,000/(2 × 153,600) = 65.10…， m = 0
<2>， <3> k = 65.10… < 256， m = 0
<4> 设置 UAnCTL2 寄存器的值： k = 65 = 41H，设置 UAnCTL1 寄存器的值： m = 0

实际波特率= 20,000,000/(2 × 65)
= 153,846 [bps]

波特率误差= {20,000,000/(2 × 65 × 153,600) − 1} × 100
= 0.160 [%]

以下为波特率设置的典型举例：

表 16-6. 波特率发生器设置数据

波特率	f_{XX} = 20 MHz			f_{XX} = 16 MHz			f_{XX} = 10 MHz		
(bps)	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)
300	08H	82H	0.16	07H	D0H	0.16	07H	82H	0.16
600	07H	82H	0.16	06H	D0H	0.16	06H	82H	0.16
1,200	06H	82H	0.16	05H	D0H	0.16	05H	82H	0.16
2,400	05H	82H	0.16	04H	D0H	0.16	04H	82H	0.16
4,800	04H	82H	0.16	03H	D0H	0.16	03H	82H	0.16
9,600	03H	82H	0.16	02H	D0H	0.16	02H	82H	0.16
19,200	02H	82H	0.16	01H	D0H	0.16	01H	82H	0.16
31,250	01H	A0H	0	01H	80H	0	00H	A0H	0
38,400	01H	82H	0.16	00H	D0H	0.16	00H	82H	0.16
76,800	00H	82H	0.16	00H	68H	0.16	00H	41H	0.16
153,600	00H	41H	0.16	00H	34H	0.16	00H	21H	−1.36
312,500	00H	20H	0	00H	1AH	−1.54	00H	10H	0
625,000	00H	10H	0	00H	0DH	−1.54	00H	08H	0

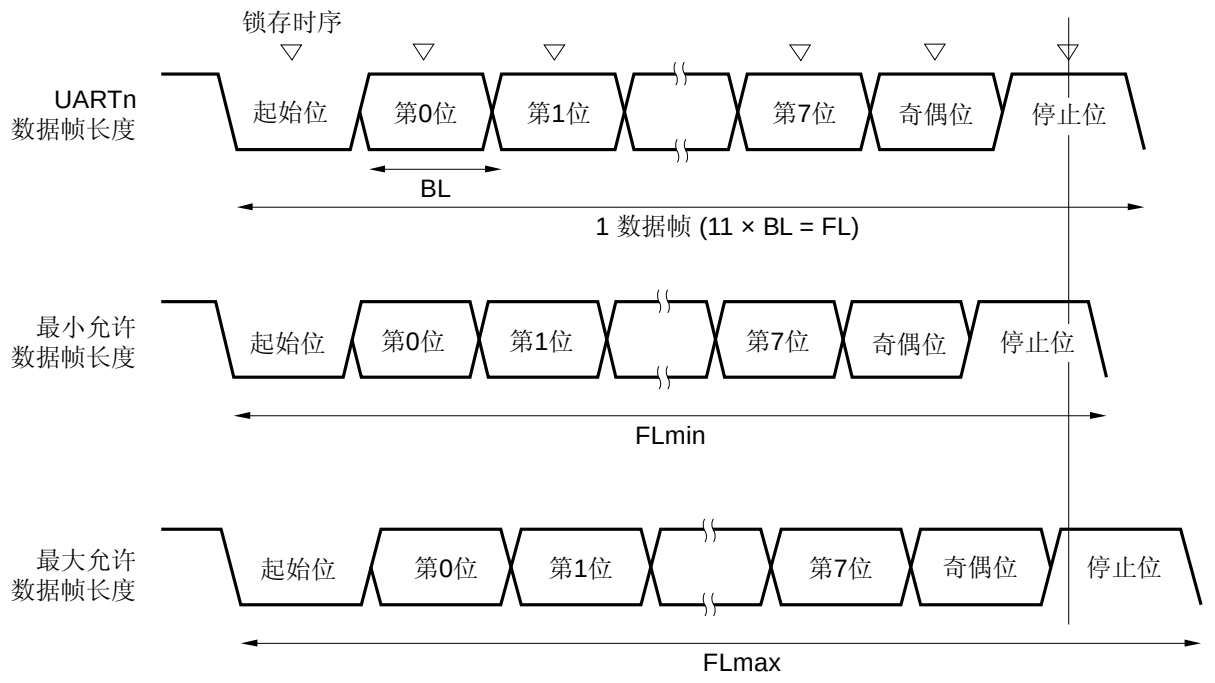
备注 f_{XX}： 主时钟频率
ERR：波特率误差 (%)

(5) 在接收期间允许的波特率范围

接收过程中，目的端允许的波特率误差范围如下所示：

注意事项 下面的波特率误差表示理论值。在实际操作中，信号可能会畸变，或即使误差在可允许范围内，通信却没有正确执行。因此，误差必须最小化。

图 16-18. 接收期间允许的波特率范围



备注 n = 0 至 5

如图 16-18 所示，接收数据的锁存时序取决于检测到起始位后 UAnCTL2 寄存器对计数器的设置值。如果在此锁存时序可以及时接收直到最末位数据（停止位），发送数据就可以被正常接收。

当应用于 11 位接收时，以下为理论结果。

$BL = (\text{波特率})^{-1}$

波特率：UARTAn 波特率(n = 0 至 2)

k： UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位的设定值(n = 0 至 2) 。

BL： 1 位数据长度

FL： 1 个数据帧的长度

锁存时间余量：2 个时钟

最小允许数据帧长度： $FLmin = 11 \times BL - \frac{k-2}{2k} \times BL = \frac{21k+2}{2k} BL$

所以，可以从目的端接收的最大波特率如下所示：

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2}\ Brate$$

同理，可以得到下列公式来计算最大数据帧长度：

$$\frac{10}{11} \times FLmax = 11 \times BL - \frac{k + 2}{2 \times k} \times BL = \frac{21k - 2}{2 \times k}\ BL$$

$$FLmax = \frac{21k - 2}{20\ k}\ BL \times 11$$

所以，可以从目的端接收的最小波特率如下所示：

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2}\ Brate$$

由上述公式计算 UARTAn 以及目的端的允许波特率误差，必须遵从以下表格。

表 16-7. 允许的最大/最小波特率误差（11 位长度）

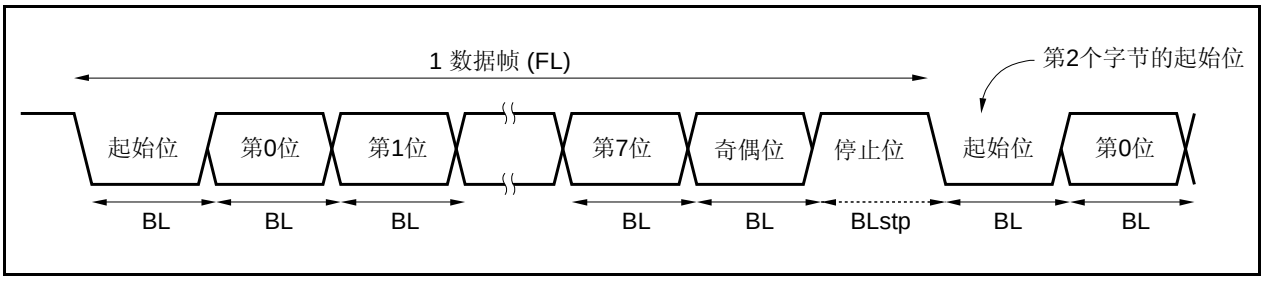
分频比(k)	允许的最大波特率误差	允许的最小波特率误差
4	+2.32%	- 2.43%
8	+3.53%	- 3.61%
20	+4.26%	- 4.31%
50	+4.56%	- 4.58%
100	+4.66%	- 4.67%
255	+4.72%	- 4.73%

备注
1. 接收精度依赖于一帧内的位数量、基准时钟频率(f_{UCLK})和分频比(k)。基准时钟频率(f_{UCLK})越高且分频比(k)越大，则精度越高。
2. k： UAnCTL2.UAnBRS7 位至 UAnCTL2.UAnBRS0 位的设定值(n = 0 至 2)

(6) 连续发送期间的数据帧长度

连续发送期间，从停止位到下一个起始位的传输间隔通常比平常多 2 个基准时钟周期。而时序的初始化由接收端通过检测起始位执行，因此，这对传输结果没有影响。

图 16-19. 连续发送期间的数据帧长度



假定 1 位数据长度为 BL；停止位长度为 FLstp；基准时钟频率为 f_{UCLK}，可以得到如下等式：

$$BLstp = BL + 2/f_{UCLK}$$

所以，连续发送期间的传输速率如下。

$$数据帧长度 = 11 \times BL + (2/f_{UCLK})$$

16.8 注意事项

(1) 当提供给 UARTAn 的时钟停止(比如进入 IDLE1、 IDLE2 或 STOP 模式)时，则操作停止，每一个寄存器都保留时钟供应停止前的瞬间值。TXDAn 引脚输出也保持时钟供给停止前的瞬时值。然而，恢复时钟供给后，不能保证正常操作。因此，在时钟恢复之后，应该将 UAnCTL0.UAnPWR 位、UAnCTL0.UAnRXEn 位和 UAnCTL0.UAnTXEn 位设置为 000，对电路进行初始化。

(2) RXDA1 和 KR7 引脚不能同时使用。使用 RXDA1 引脚时，不要使用 KR7 引脚。若使用 KR7 引脚，就不要使用 RXDA1 引脚（建议将 PFC91 位置为 1，并清除 PFCE91 位为 0）。

(3) UARTAn 中，不会产生由通信错误引起的中断。使用 DMA 传输功能进行发送数据和接收数据的传输时，即使传输期间发生错误（奇偶校验错误，溢出错误，帧错误），也不执行错误处理。或者在 DMA 传输完成后读取 UAnSTR 寄存器检查是否有错误，或者在通信期间读取 UAnSTR 寄存器来检查错误。

(4) INTP7 和 RXDA0 使用相同引脚。当该引脚用作 RXDA0 引脚时，禁止对 INTP7 边沿检测（INTF3.INTF31 位 = 0，INTR3.INTR31 位 = 0）。

(5) 按以下步骤启动 UARTAn。
<1> 将 UAnCTL0.UAnPWR 位置为 1。
<2> 设置该端口。
<3> 将 UAnCTL0.UAnTXE 位和 UAnCTL0.UAnRXE 位设置为 1。

(6) 按以下步骤停止 UARTAn。
<1> 将 UAnCTL0.UAnTXE 位和 UAnCTL0.UAnRXE 位清除为 0。
<2> 设置端口，并设置 UAnCTL0.UAnPWR 位为 0（如果端口设置不改变，则不存在该问题）。

(7) 发送模式下（UAnCTL0.UAnPWR 位 = 1 且 UAnCTL0.UAnTXE 位 = 1），因为写入 UAnTX 寄存器会开始发送，所以不要用软件向该寄存器写入相同的值。若连续发送相同值，则重写相同的值。

(8) 在连续发送中，从停止位到下一个起始位的传输间隔通常比平常多 2 个基准时钟周期。但是，接收端通过检测起始位进行时序的初始化，因此接收结果不受影响。

(9) 如果低电平信号持续输入 RXDAn 引脚，UARTA 无法识别起始位。

第十七章 异步串行接口C (UARTC)

V850ES/JG3-L 具有 1 路通道 UARTC。

17.1 特性

- 片上专用波特率发生器
- 传输速率： 300 bps 至 625 kbps（使用专用波特率发生器）
- 全双工通信
- 双缓冲配置　　内部 UARTC0 接收数据寄存器 (UC0RX)
  内部 UARTC0 发送数据寄存器 (UC0TX)
- 接收错误检测功能
  - 奇偶错误
  - 帧错误
  - 溢出错误
- 中断源： 2
  - 接收完成中断（INTUC0R）：　在接收使能状态下，串行传输完成后，当接收数据从接收移位寄存器传送到接收数据寄存器时，产生该中断。
  - 发送使能中断（INTUC0T）：　在发送使能状态下，当数据从发送数据寄存器传送到发送移位寄存器时，产生该中断（可以支持连续发送）。
- 字符长度： 7 至 9 位
- 奇偶校验功能： 奇校验、偶校验、0 校验、无校验
- 发送停止位： 1，2 位
- 可选择 MSB/LSB 先行传输
- 内部数字噪声滤波器
- 支持发送/接收数据的反相输入/输出
- 在 LIN（局域互联网）通信格式下 SBF (同步间隔场)发送/接收
  - SBF 发送时，可选 13 至 20 位
  - SBF 接收时，可以识别 11 位或更多位
  - 提供 SBF 接收标志

17.2 配置

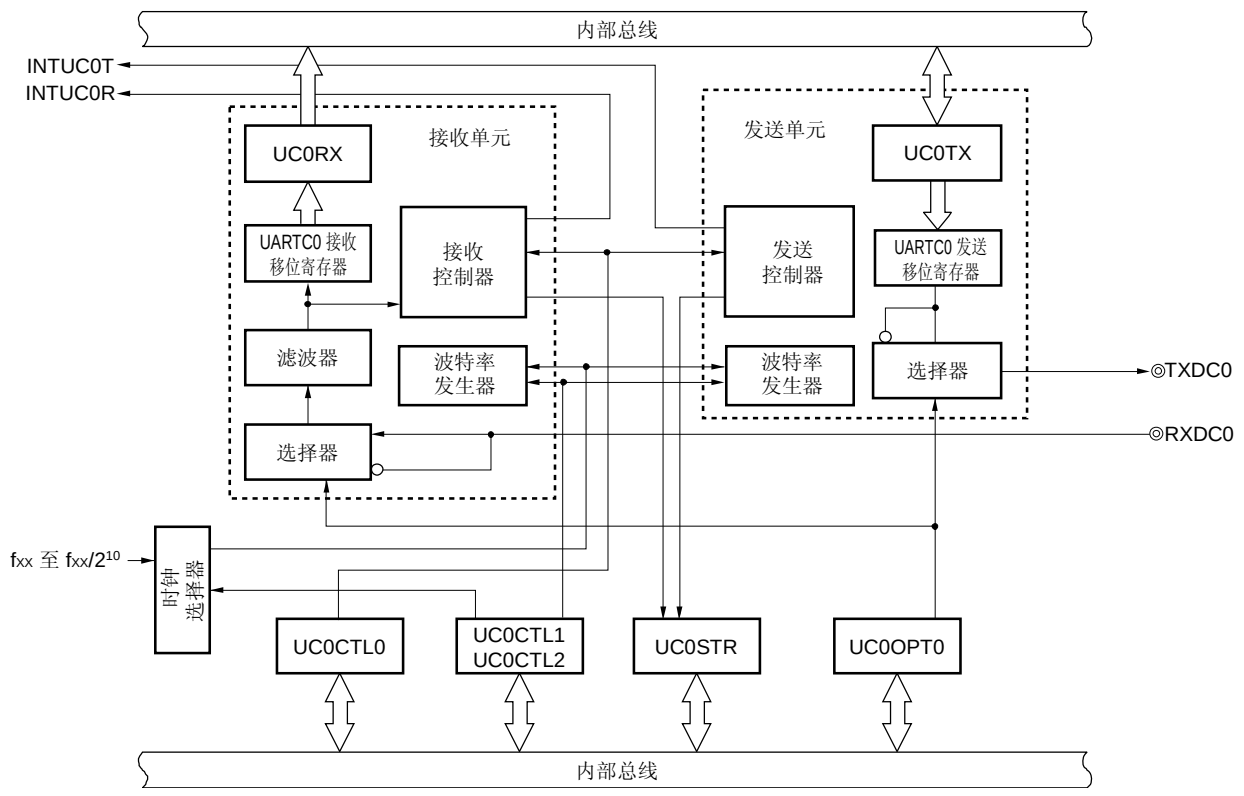
UARTC0 包括下列硬件：

表 17-1. UARTC0 的配置

项目	配置
寄存器	UARTC0 控制寄存器 0 （UC0CTL0） UARTC0 控制寄存器 1 （UC0CTL1） UARTC0 控制寄存器 2 （UC0CTL2） UARTC0 选项控制寄存器 0 （UC0OPT0） UARTC0 选项控制寄存器 1 （UC0OPT1） UARTC0 状态寄存器 （UC0STR） UARTC0 接收移位寄存器 UARTC0 接收数据寄存器（UC0RX） UARTC0 发送移位寄存器 UARTC0 发送数据寄存器（UC0TX）

UARTC0 的功能框图如下所示：

图 17-1. UARTC0 的功能框图



备注 关于波特率发生器的配置，参见**图 17-15**。

(1) UARTC0 控制寄存器 0 (UC0CTL0)

UC0CTL0 寄存器是 8 位寄存器，用于指定 UARTC0 的操作。

(2) UARTC0 控制寄存器 1 (UC0CTL1)

UC0CTL1 寄存器是 8 位寄存器，用于选择 UARTC0 的基准时钟(f_{UCLK})。

(3) UARTC0 控制寄存器 2 (UC0CTL2)

UC0CTL2 寄存器是 8 位寄存器，与 UC0CTL1 寄存器共同使用，为 UARTC0 产生所需的波特率。

(4) UARTC0 选项控制寄存器 0 (UC0OPT0)

UC0OPT0 寄存器是 8 位寄存器，用于控制在 LIN 通信格式下的 SBF 发送/接收，或控制 UARTC0 发送/接收信号的电平。

(5) UARTC0 选项控制寄存器 1 (UC0OPT1)

UC0OPT1 寄存器是 8 位寄存器，用于控制 UARTC0 的 9 位长度串行传输。

(6) UARTC0 状态寄存器 (UC0STR)

UC0STRn 寄存器是 8 位寄存器，用于指示接收错误的内容。当发生接收错误时，相应接收错误标志被置位（1）。

(7) UARTC0 接收移位寄存器

本寄存器是移位寄存器，用于将输入至 RXDC0 引脚的串行数据转换为并行数据。接收到 1 帧数据并检测到停止位时，就将接收数据传输至 UC0RX 寄存器。

不能直接操作该寄存器。

(8) UARTC0 接收数据寄存器 (UC0RX)

UC0RX 是 8 位缓冲寄存器，用于保存接收数据。

在接收使能状态下，当 1 帧移入操作完毕时，接收数据同步从 UARTC0 接收移位寄存器传送到 UC0RX 寄存器。

传送到 UC0RX 寄存器也会引发接收完成中断请求信号（INTUC0R）。

(9) UARTC0 发送移位寄存器

该移位寄存器用于将来自 UC0TX 寄存器的并行数据转换为串行数据。

当一帧的数据从 UC0TX 寄存器传送过来时，移位寄存器的数据从 TXDC0 引脚输出。

不能直接操作该寄存器。

(10) UARTC0 发送数据寄存器 (UC0TX)

UC0TX 寄存器是 8 位发送数据缓冲缓存器。当发送的数据被写入 UC0TX 寄存器后，开始发送。当数据能够写入 UC0TX 寄存器时 (当一帧数据从 UC0TX 寄存器传送到 UARTC0 发送移位寄存器时)，产生发送使能中断请求信号 (INTUC0T)。

17.2.1 各个通道的引脚功能

UARTC 使用的 RXDC0 和 TXDC0 引脚也可以用作其它功能，如表 17-2 所示。UARTC 要使用这些引脚，相关寄存器的设置描述如**表 4-15 端口引脚用作复用功能时的设置**。

表 17-2. UARTC 使用的引脚

通道	引脚编号		端口	UARTC 接收输入	UARTC 发送输出	其它功能
	GC	F1				
UARTC0	50	J11	P97	RXDC0	–	A7/SIB7/TIP20/TOP20
	49	K11	P96	–	TXDC0	A6/TIP21/TOP21

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

17.3 UARTC和其它串行接口的模式转换

17.3.1 UARTC0 和CSIB1 的模式转换

在 V850ES/JG3-L 中，UARTC0 和 CSIB1 是相同引脚的复用功能，因而不能同时使用。使用之前，用 PMC9 和 PFC9 寄存器预先设置 UARTC0。

注意事项 如果在发送或接收期间切换这些功能，则 UARTC0 和 CSIB1 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 17-2. UARTC0 和 CSIB1 模式转换设置

复位后：00H R/W 地址：FFFFF462H

	7	6	5	4	3	2	1	0
PMC9	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

复位后：00H R/W 地址： FFFFF472H

	7	6	5	4	3	2	1	0
PFC9	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

复位后：00H R/W 地址：FFFFF712H

	7	6	5	4	3	2	1	0
PFCE9	PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

PMC97	PFCE97	PFC97	操作模式
0	0	0	端口 I/O 模式
1	0	1	SIB1 (CSIB1)/RXDC0 (UARTC) 注

注 SIB1 和 RXDC0 是复用功能。引脚用作 SIB1 引脚时，禁止 RXDC0 引脚检测的复用功能(将 UC0CTL0.UC0PWR 位清除为 0)。同样，引脚用作 RXDC0 引脚时，禁止 SIB1 引脚的复用功能(将 CB1CTL0.CB1PWR 位清除为 0)。

备注 × = 无需理会

17.4 寄存器

(1) UARTC0 控制寄存器 0 (UC0CTL0)

UC0CTL0 寄存器是 8 位寄存器，用于控制 UARTC0 的串行传输操作。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 10H。

(1/2)

复位后：10H　R/W　地址：　FFFFFAA0H

	<7>	<6>	<5>	<4>	3	2	1	0
UC0CTL0	UC0PWR	UC0TXE	UC0RXE	UC0DIR	UC0PS1	UC0PS0	UC0CL	UC0SL

UC0PWR	UARTC0 操作控制
0	禁止 UARTC0 操作 (UARTC0 异步复位)
1	允许 UARTC0 操作

UARTC0 操作由UC0PWR 位控制 。通过清除UC0PWR位为0， TXDC0 引脚输出固定为高电平 (如果UC0OPT0.UC0TDL 位 = 1，则固定为低电平)。

UC0TXE	发送操作使能
0	发送操作禁止
1	发送操作使能

- 若要开始发送，设置 UC0PWR 位为 1 然后设置 UC0TXE 位为 1。
  若要停止发送， 清除UC0TXE 位为0 然后清除 UC0PWR 位为 0。
- 若要初始化发送单元，清除UC0TXE位为0，等待两个周期的基准时钟，然后再次设置 UC0TXE 位为 1 。 否则无法执行初始化 (关于基准时钟，参见 17.7 (1) (a) 基础时钟)。
- 当UARTC0操作使能 (UC0PWR 位 = 1)且UC0TXE位被设置为1时，在经过基准时钟(f_{UCLK})的至少两个周期之后，发送被使能。

UC0RXE	接收操作使能
0	接收操作禁止
1	接收操作使能

- 若要开始接收 , 设置 UC0PWR 位为 1 然后 设置 UC0RXE 位为 1。
  若要停止接收 , 清除 UC0RXE 位为 0 然后清除 UC0PWR 位为 0。
- 若要初始化接收单元， 清除 UC0RXE 位为 0, 等待两个周期的基准时钟，然后再次设置 UC0RXE 位为 1。 否则无法执行初始化 (关于基准时钟，参见 17.7 (1) (a) 基准时钟)。
- 当UARTC0操作使能 (UC0PWR 位 = 1)且UC0RXE位被设置为1时，在经过基准时钟(f_{UCLK})的至少两个周期之后，接收被使能。如果在接收使能之前收到一个起始位，该起始位被忽略。

(2/2)

UC0DIR	传输方向选择
0	MSB先行
1	LSB先行

• 只有 UC0PWR 位 = 0 或 UC0TXE 位 = UC0RXE 位 = 0时，
才可以重写该寄存器。
• 在 LIN 格式下进行发送和接收时， 设置 UC0DIR位为1。

UC0PS1	UC0PS0	发送期间奇偶校验的选择	接收期间奇偶校验的选择
0	0	无奇偶校验输出	无奇偶校验接收
0	1	0 奇偶校验输出	0 奇偶校验接收
1	0	奇校验输出	奇校验检测
1	1	偶校验输出	偶校验检测

• 只有 UC0PWR 位 = 0或 UC0TXE 位 = UC0RXE 位 = 0时，
才可以重写该寄存器。
• 如果接收期间选择“ 0 奇偶校验接收”，则不进行奇偶校验检查 。
因此不设置 UC0STR.UC0PE 位。
• 在 LIN 格式下进行发送和接收时，将UC0PS1和UC0PS0 清除为00。

UC0CL	1帧的发送/接收数据字符长度的说明
0	7 位
1	8 位

• 只有 UC0PWR 位为0 或 UC0TXE 位为0且UC0RXE位为0时，
才可以重写该寄存器。
• 在 LIN 格式下进行发送和接收时，设置 UC0CL位为1。

UC0SL	发送数据的停止位长度的说明
0	1 位
1	2 位

只有 UC0PWR 位为0 或 UC0TXE 位为0且UC0RXE位为0时，
才可以重写该寄存器。

备注 关于奇偶校验的详情，参见 **17.6.6 奇偶类型和操作。**

(2) UARTC0 控制寄存器 1 (UC0CTL1)

更多详情，参见 **17.7 (2) UARTC0 控制寄存器 1 (UC0CTL1)**。

(3) UARTC0 控制寄存器 2 (UC0CTL2)

更多详情，参见 **17.7 (3) UARTC0 控制寄存器 2 (UC0CTL2)**。

(4) UARTC0 选项控制寄存器 0 (UC0OPT0)

UC0OPT0 寄存器是 8 位寄存器，用于控制在 LIN 通信格式下的 SBF 发送/接收，或控制 UARTC0 发送/接收信号的电平。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 14H。

(1/2)

复位后: 14H　R/W　地址:　FFFFFAA3H

	<7>	6	5	4	3	2	1	0
UC0OPT0	UC0SRF	UC0SRT	UC0STT	UC0SLS2	UC0SLS1	UC0SLS0	UC0TDL	UC0RDL

UC0SRF	SBF接收标志
0	UC0CTL0.UC0PWR 位 或UC0CTL0.UC0RXE位 为0 时。 在SBF 接收正常结束时。
1	SBF 接收中

- LIN 通信期间，表示是否接收到 SBF (同步间隔场) 。
- 发生SBF接收错误时，UC0SRF 位保持为1，然后SBF接收重新开始。
- UC0SRF 位是只读位。

UC0SRT	SBF 接收触发
0	-
1	SBF 接收触发

- LIN 通信期间该位是SBF接收触发位，读出值总为“0”。
- 要进行SBF 接收, 需设置 UC0SRT 位 (为 1) 以允许SBF接收。
- 设置 UC0PWR 位 和UC0RXE位为1 后设置UC0SRT 位。

UC0STT	SBF 发送 触发
0	-
1	SBF 发送 触发

- LIN 通信期间该位是SBF 发送触发位，读出值总为“0”。
- 将该位设置为1，可以触发SBF发送。
- 设置 UC0PWR 位 和UC0RXE位为1 后设置UC0SRT 位。

注意事项 SBF 接收（UC0SRF 位 = 1）期间，不要设置 UC0SRT 和 UC0STT 位（1）。

(2/2)

UC0SLS2	UC0SLS1	UC0SLS0	SBF 发送长度选择
1	0	1	13-位输出（复位默认值）
1	1	0	14-位输出
1	1	1	15-位输出
0	0	0	16-位输出
0	0	1	17-位输出
0	1	0	18-位输出
0	1	1	19-位输出
1	0	0	20-位输出

UC0PWR 位 = 0 或 UC0TXE 位 = 0时，可设置该寄存器。

UC0TDL	发送数据的电平
0	传输数据正常输出
1	传输数据反相输出

- TXDC0 引脚的输出电平可使用UC0TDL 位进行反相。
- UC0PWR 位 = 0 或UC0TXE 位 = 0时，可设置该寄存器。

UC0RDL	接收数据电平
0	传输数据正常输入
1	传输数据反相输入

- RXDC0 引脚的输入电平可使用UC0RDL 位进行反相。
- UC0PWR 位 = 0 或 UC0RXE 位 = 0时，可设置该寄存器。

(5) UARTC0 选项控制寄存器 1 (UC0OPT1)

UC0OPT1 寄存器是 8 位寄存器，用于控制 UARTC0 寄存器的串行传输操作。
该寄存器可进行字节读写或按位读写。
系统复位后，该寄存器被设为 00H。

注意事项 当 UARTC 的操作被禁止时，设置 UC0EBE 位(UC0CTL0.UC0PWR = 0)。

复位后：00H R/W 地址： FFFFFAAAH

	7	6	5	4	3	2	1	0
UC0OPT1	0	0	0	0	0	0	0	UC0EBE

UC0EBE	扩展位使能/禁止
0	禁止操作扩展位。按照UC0CTL0.UC0CL位设置的数据长度执行发送/接收。
1	使能操作扩展位。按照 9 位字符长度执行发送/接收。

- 当UC0EBE 位设置为1，且按照9 位长度发送时，请确保设置如下。如果未执行该设置，则 UC0EBE 位的设置无效。
  - UC0CTL0.UC0PS1, UC0PS0 = 00 (无校验)
  - C0CTL0.UC0CL = 1 (8 位字符长度)
- 如果按照LIN通信格式进行发送或接收，设置UC0EBE 为 0。

下表展示了寄存器设置值和数据格式之间的关系。

表 17-2. 寄存器设置和数据格式之间的关系

寄存器设置					数据格式				
UC0CTL0				UC0OPT1	D0 至 D6	D7	D8	D9	D10
UC0CL	UC0PS1	UC0PS0	UC0SL	UC0EBE					
0	0	0	0	0	数据	停止位	–	–	–
0	除 00 之外				数据	奇偶位	停止位	–	–
1	0	0			数据	数据	停止位	–	–
1	除 00 之外				数据	数据	奇偶位	停止位	–
0	0	0	1	0	数据	停止位	停止位	–	–
0	除 00 之外				数据	奇偶位	停止位	停止位	–
1	0	0			数据	数据	停止位	停止位	–
1	除 00 之外				数据	数据	奇偶位	停止位	停止位
0	0	0	0	1	数据	停止位	–	–	–
0	除 00 之外				数据	奇偶位	停止位	–	–
1	0	0			数据	数据	数据	停止位	–
1	除 00 之外				数据	数据	奇偶位	停止位	–
0	0	0	1	1	数据	停止位	停止位	–	–
0	除 00 之外				数据	奇偶位	停止位	停止位	–
1	0	0			数据	数据	数据	停止位	停止位
1	除 00 之外				数据	数据	奇偶位	停止位	停止位

备注 数据(Data)： 数据位
停止位(Stop)： 停止位
奇偶位(Parity)： 奇偶校验位

(6) UARTC0 状态寄存器 (UC0STR)

UC0STR 寄存器是 8 位寄存器，显示 UARTC0 的传输状态和接收错误的内容。

该寄存器可进行字节读写或按位读写，但是， UC0TSF 位为只读位，而 UC0PE、UC0FE 和 UC0OVE 位既可读也可写。但是，这些位只能写入 0 来清除，而不能通过写 1 来置位(即便写入 1，之前的值保持不变)。

清除 UC0STR 寄存器的条件如下：

表 17-3. 清除 STR 寄存器的条件

寄存器/位	清除条件
UC0STR 寄存器	• 复位 • UC0CTL0.UC0PWR = 0
UC0TSF 位	• UC0CTL0.UC0TXE = 0
UC0PE 位、UC0FE 位、UC0OVE 位	• 写入 0 • UC0CTL0.UC0RXE = 0

复位后: 00H R/W 地址: FFFFFAA4H

	<7>	6	5	4	3	<2>	<1>	<0>
UC0STR	UC0TSF	0	0	0	0	UC0PE	UC0FE	UC0OVE

UC0TSF	传输状态标志
0	发送移位寄存器没有数据。 • 当UC0PWR位或UC0TXE位已经被设置为0时。 • 在传输完成之后，没有从UC0TX寄存器传来下一个数据。
1	发送移位寄存器有数据。(写入UC0TX寄存器)

进行连续发送时，UC0TSF位总是为1。
初始化发送单元时，在进行初始化前确保UC0TSF位= 0。
如果在UC0TSF位= 1时执行初始化，则发送数据无法保证。

UC0PE	奇偶校验 错误标志
0	• 已经设置UC0PWR位=0 或 UC0RXE=0。 • 已经写入 0 时。
1	接收到的奇偶位与指定的奇偶校验位不匹配。

- 通过设置UC0CTL0.UC0PS1 和 UC0CTL0.UC0PS0 位来控制UC0PE位的操作。
- 一旦UC0PE位被置位(1)，则值被保持，直到该位被清除 (0)。
- UC0PE位可读可写，但是只能写入0进行清除，而不能写入1进行置位，如果向该位写入1，其值保持不变。

UC0FE	帧错误标志
0	• 已经设置UC0PWR位=0 或 UC0RXE=0。 • 已经写入 0 时。
1	接收期间没有检测到停止位。

- 只检测接收数据停止位的第一位，而无需考虑UC0CTL0.UC0SL位的值。
- 一旦UC0FE位被置位(1)，则值被保持，直到该位被清除 (0)。
- UC0FE位可读可写，但是只能写入0进行清除，而不能写入1进行置位，如果向该位写入1，其值保持不变。

UC0OVE	溢出错误标志
0	• 已经设置UC0PWR位=0 或 UC0RXE=0。 • 已经写入 0 时。
1	接收数据已经设置到 UC0RX 寄存器 并且在接收数据被读取之前下一个接收操作已经完成。

- 发生溢出错误时，数据被丢弃，下一个接收数据不会被写入接收缓冲器。
- 一旦UC0OVE位被置位(1)，则值被保持，直到该位被清除 (0)。
- UC0OVE位可读可写，但是只能写入0进行清除，而不能写入1进行置位，如果向该位写入1，其值保持不变。

(7) UARTC0 接收数据寄存器 L (UC0RXL)和 UARTC0 接收数据寄存器(UC0RX)

UC0RXL 和 UC0RX 寄存器是 8 位或 9 位缓冲寄存器，用于存储由接收移位寄存器转换而来的并行数据。
1 个字节的数据接收完成时，将存储于接收移位寄存器的数据传送到 UC0RXL 和 UC0RX 寄存器。
LSB 接收期间，数据长度指定为 7 位时，接收数据传输至 UC0RXL 寄存器的第 6 位至第 0 位，MSB 位总为 0。
MSB 接收期间，将接收数据传输到 UC0RXL 的第 7 位至第 1 位，且 LSB 位总为 0。
发生溢出（UC0OVE）错误时，此时的接收数据不会被传送到 UC0RXL 和 UC0RX 寄存器，而是被丢弃。
根据字符长度，访问单元和复位值有所不同。

- 字符长度 7/8 位 (UC0OPT1.UC0EBE = 0)
  该寄存器只读，按照 8 位宽度。
  复位或清除 UC0CTL0.UC0PWR 位 = 0，该寄存器被置为 FFH。
- 字符长度 9 位 (UC0OPT1.UC0EBE = 0)
  该寄存器只读，按照 16 位宽度。
  复位或清除 UC0CTL0.UC0PWR 位 = 0，该寄存器被置为 01FFH。

(a) 字符长度 7/8 位(UC0OPT1.UC0EBE = 0)

复位后： FFH R 地址： UC0RXL FFFFFAA6H

	7	6	5	4	3	2	1	0
UC0RXL								

(b) 字符长度 9 位 (UC0OPT1.UC0EBE = 1)

复位后： 01FFH R 地址：UC0RX FFFFFAA6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UC0RX	0	0	0	0	0	0	0									

(8) UARTC0 发送数据寄存器 L (UC0TXL)，UARTC0 发送数据寄存器 (UC0TX)

UC0TXL 和 UC0TX 寄存器是 8 位或 9 位寄存器，用于设置发送数据。
LSB 发送期间，数据长度指定为 7 位时，发送数据传输至 UC0TX 寄存器的第 6 位至第 0 位。MSB 发送期间，将发送数据传输到 UC0TX 的第 7 位至第 1 位。
根据字符长度，访问单元和复位值有所不同。

- 字符长度 7/8 位 (UC0OPT1.UC0EBE = 0)
  该寄存器按照 8 位宽度进行读取和写入。
  复位后该寄存器被置为 FFH。
- 字符长度 9 位 (UC0OPT1.UC0EBE = 0)
  该寄存器按照 16 位宽度进行读取和写入。
  复位后该寄存器被置为 01FFH。

注意事项 1. 在发送使能状态(UC0PWR 位 = 1 且 UC0TXE 位 = 1)下向 UC0TXL、UC0TX 寄存器写入发送数据，会触发发送启动。如果写入的值与 INTUC0T 中断发生前的值相同，则相同的数据被发送两次。
2. 为连续发送写入数据，要在 INTUC0T 中断发生之后。
如果在 INTUC0T 中断发生之前写入了下一个数据，发送启动处理与 UC0TXL、UC0TX 寄存器写入操作冲突，可能导致意外操作。
3. 如果在发送操作禁止时向 UC0TXL、UC0TX 寄存器写入发送数据，不能作为发送启动触发。于是，即使在发送操作禁止时向 UC0TXL、UC0TX 寄存器写入发送数据，以后发送使能时，发送也不会启动。

(a) 字符长度 7/8 位(UC0OPT1.UC0EBE = 0)

复位后： FFH R/W 地址： UC0TXL FFFFFAA8H

	7	6	5	4	3	2	1	0
UC0TXL								

(b) 字符长度 9 位 (UC0OPT1.UC0EBE = 1)

复位后： 01FFH R 地址： UC0TX FFFFFAA8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UC0TX	0	0	0	0	0	0	0									

17.5 中断请求信号

UARTC0 会产生以下两种中断请求信号：

- 接收完成中断请求信号(INTUC0R)
- 发送使能中断请求信号 (INTUC0T)

这两种中断请求信号的默认优先级为接收完成中断请求信号优先，然后为发送使能中断请求信号。

表 17-4. 中断及其默认优先级

中断请求信号	优先级
接收完成	高
发送使能	低

(1) 接收完成中断请求信号(INTUC0R)

接收使能状态下，存储在接收移位寄存器中的数据传输至 UC0RX 寄存器时，产生接收完成中断请求信号。

发生接收错误时，接收完成中断请求信号也会继续输出。因此，响应接收完成中断请求信号并读取数据时，读取 UC0STR 寄存器并检查接收结果是否错误。

在接收禁止状态下，不产生接收完成中断请求信号。

(2) 发送使能中断请求信号 (INTUC0T)

在发送允许状态下，当发送数据从 UC0TX 寄存器传输至 UARTC0 发送移位寄存器，产生发送使能中断请求信号。

17.6 操作

17.6.1 数据格式

执行全双工串行数据的接收和发送。

如图 17-5 所示，一个发送/接收数据帧包括起始位、字符位、校验位以及停止位组成。

使用 UC0CTL0 寄存器指定一个数据帧内字符位长度、校验段，指定停止位长度，以及 MSB/LSB 先行传输方式。

使用 UC0OPT1 寄存器执行 9 位字符长度说明。

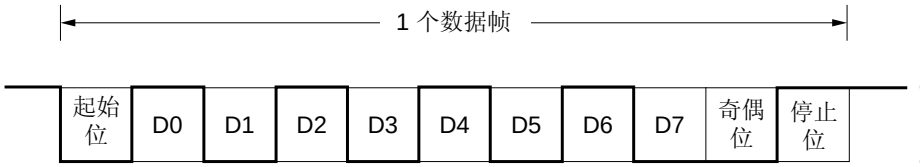
UC0OPT0.UC0TDL 位用于指定要通过 TXDC0 引脚发送的正常输出/反相输出。

UC0OPT0.UC0RDL 位用于指定要通过 RXDC0 引脚接收的正常输出/反相输出。

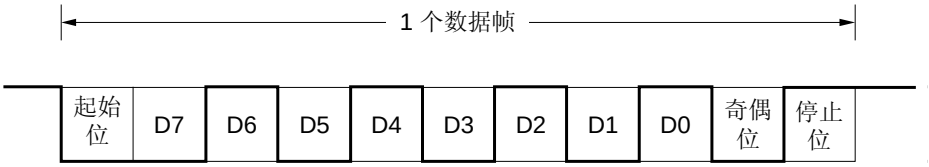
- 起始位.....1 位
- 字符位.....7 位/8 位/9 位
- 奇偶校验位 .....奇检验/偶校验/0 校验/无校验
- 停止位.....1 位/2 位
- 输入逻辑.....正常输出/反相输出
- 输出逻辑.....正常输出/反相输出
- 通信方向.....MSB/LSB 先行

图 17-3. UARTC 发送/接收数据格式

(a) 8 位数据长度，LSB，偶校验， 一个停止位，传输数据：55H

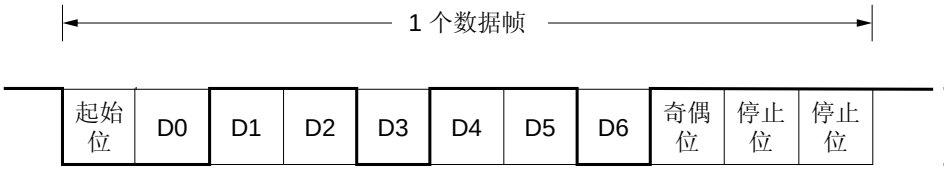


(b) 8 位数据长度，MSB，偶校验， 一个停止位，传输数据：55H

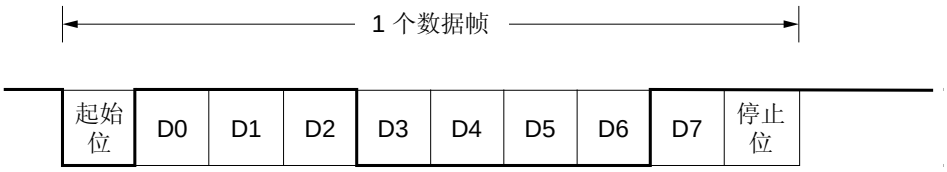


(c) 8 位数据长度，MSB，偶校验， 一个停止位，传输数据：55H，反相发送/接收数据

(d) 7 位数据长度，LSB，奇校验， 2 个停止位，传输数据：36H



(e) 8 位数据长度，LSB，无校验， 一个停止位，传输数据：87H



17.6.2 UART 发送

将 UC0CTL0.UC0PWR 位和 UC0CTL0.UC0TXE 位设置为 1，进入发送使能状态，将发送数据写入 UC0TX 寄存器启动发送。起始位、校验位以及停止位自动添加。

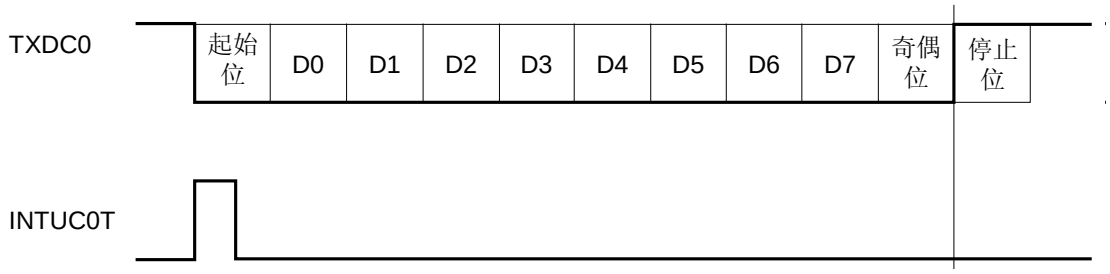
由于 UARTC0 中不提供 CTS (发送允许信号) 输入引脚，所以需要使用一个端口来检查发送目的方已经允许接收。

在发送启动时，UC0TX 寄存器内的数据被传送到 UARTC0 发送移位寄存器 。

UC0TX 寄存器的数据发送至 UARTC0 发送移位寄存器的操作完成时，产生发送使能中断请求信号（INTUC0T），之后 UARTC0 发送移位寄存器的内容由 TXDC0 引脚输出。

INTUC0T 信号产生后，允许向 UC0TX 寄存器写入下一个发送数据。

图 17-4. UART 发送



备注 LSB 先行。

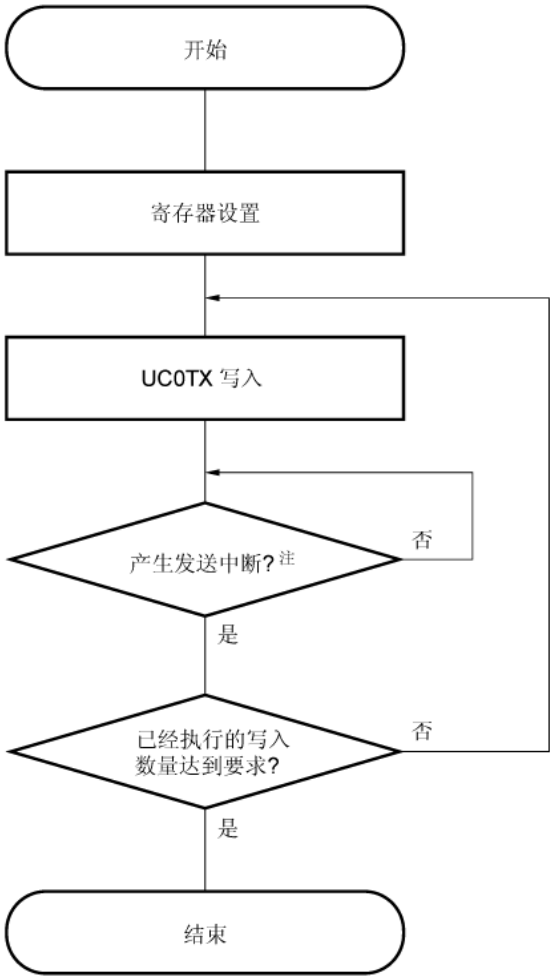
17.6.3 连续发送过程

在发送使能时向 UC0TX 寄存器写入发送数据，会触发发送。UC0TX 寄存器内的数据被传送到 UARTC0 发送移位寄存器，产生发送使能中断请求信号 (INTUC0T)，然后开始移位。在发送使能中断请求信号 (INTUC0T)产生之后，下一个数据可以写入到 UC0TX 寄存器。UARTC0 发送移位寄存器的发送时序可以由发送使能中断请求信号 (INTUC0T)来判断。

在传输过程中将下一个待发送数据写入 UC0TX 寄存器使得传输效率更高。

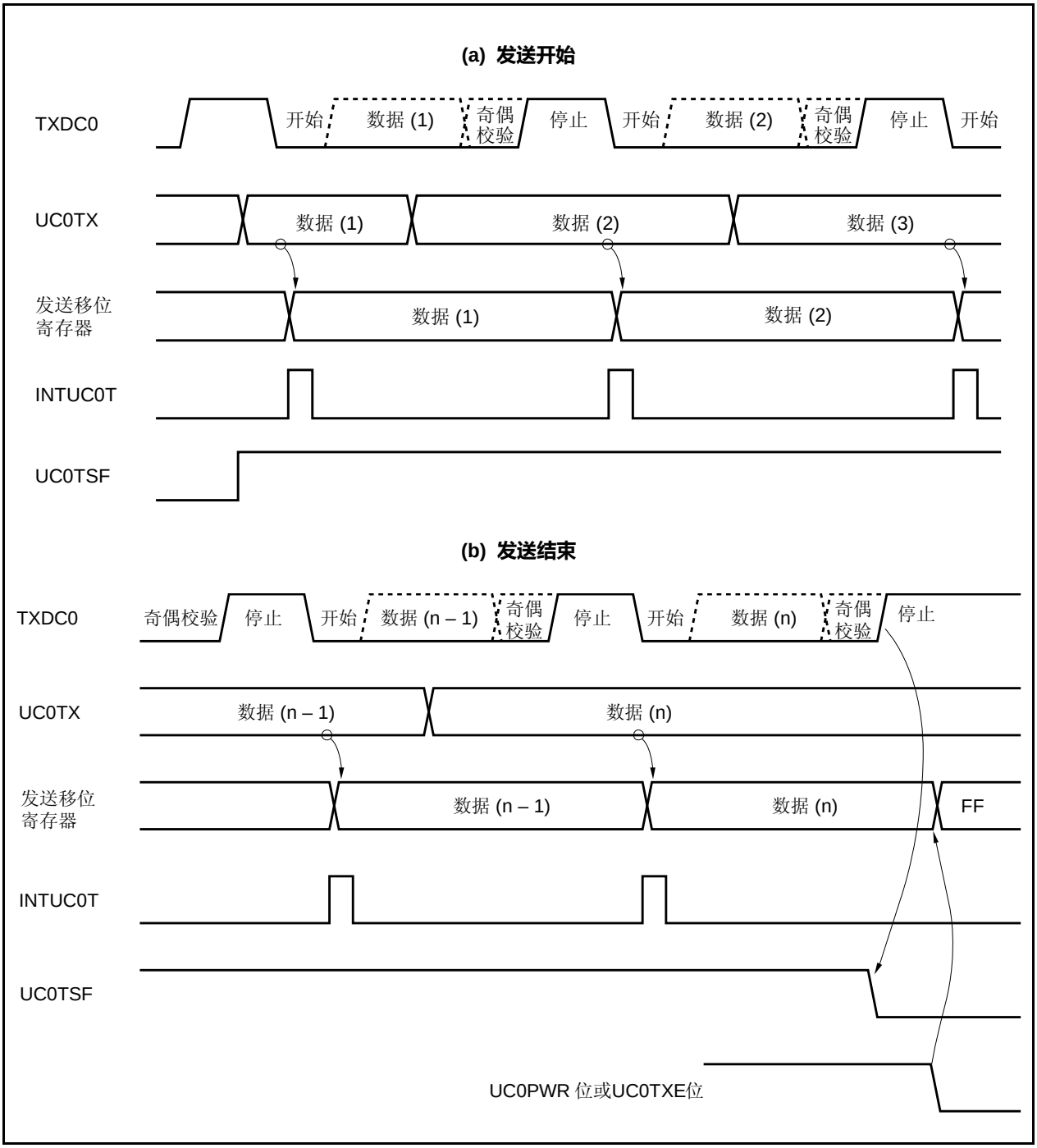
注意事项 连续发送执行期间进行发送初始化时，请确保 UC0STR.UC0TSF 位为 0，然后再执行初始化。UC0TSF 位为 1 时进行初始化，则发送数据无法保证。

图 17-5. 连续发送处理



注 请确保在发送使能中断请求信号 (INTUC0T)产生之后读取 UC0STR 寄存器，以检查是否发生发送错误。

图 17-6. 连续发送操作时序



17.6.4 UART 接收

首先，通过执行下列操作使能接收，并监控 RXDC0 输入以检测起始位。

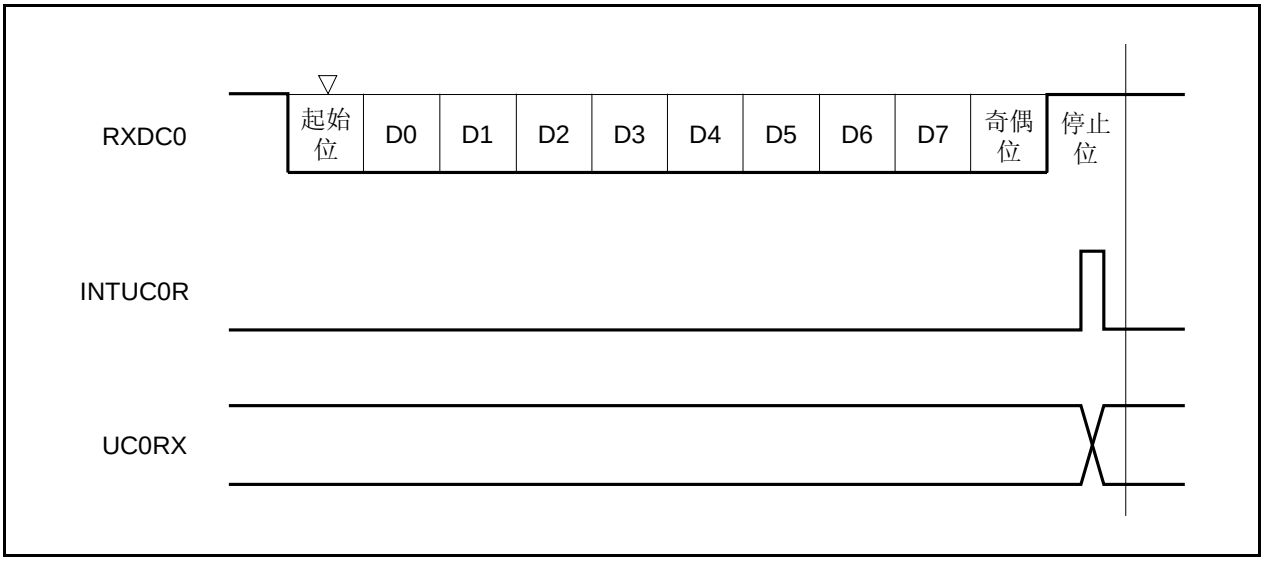
- 使用 UARTC 控制寄存器 1 (UC0CTL1)指定操作时钟。
- 使用 UARTC 控制寄存器 2 (UC0CTL2)指定波特率。
- 使用 UARTC option 控制寄存器 0 (UC0OPT0)指定输出逻辑电平。
- 使用 UARTC 控制寄存器 0 (UC0CTL0)指定通信方向、校验位、数据帧长度和停止位长度。
- 置位供电位和接收使能位 (UC0PWR = 1 和 UC0RXE = 1)。

要改变通信方向、校验位、数据帧长度和停止位长度，清除供电位 (UC0PWR = 0)或事先将发送使能位和接收使能位都清除 (UC0TXE = 0 和 UC0RXE = 0)。

使用操作时钟对 RXDC0 引脚的输入电平采样。如果检测到下降沿，则开始对 RXDC0 的输入数据进行采样。如果在检测到下降沿之后数据后半位是低电平（在图 17-9 中用 ▽ 表示），则认为是起始位。确认起始位之后，开始接收操作，按照设定的波特率，串行数据被顺序存入 UARTC0 接收移位寄存器。当接收到停止位时，会输出接收完成中断请求信号（INTUC0R）时，同时，UARTC0 接收移位寄存器的数据被传送到接收数据寄存器（UC0RX）。

但是，如果发生溢出（UC0OVE 位 = 1）错误，此时接收到的数据不写入 UC0RX 寄存器，而是被丢弃。此外，即使发生奇偶错误（UC0PE 位 = 1）或帧错误（UC0FE 位 = 1），接收还会继续进行，接收数据被传送到 UC0RX 寄存器。不论发生哪种接收错误，接收完成之后都会产生 INTUC0R 中断。

图 17-7. UART 接收



注意事项 1. 即使发生接收错误，也必须读取 UC0RX 寄存器。如果不读取 UC0RX，则下一个数据接收期间发生溢出错误，并且接收错误会不断地连续发生。

2. 认为接收操作期间只有一个停止位。忽略第二个停止位。

3. 当接收完成时，在产生接收结束中断请求信号（INTUC0R）之后读取 UC0RX 寄存器，并对 UC0PWR 位或 UC0RXE 位清 0。如果在 INTUC0R 信号产生之前对 UC0PWR 位或 UC0RXE 位清 0，则无法保证对 UC0RX 寄存器值的正确读取。

4. 如果 UARTC0 的接收完成处理(INTUC0R 信号产生)与 UC0PWR 位或 UC0RXE 位设置为 0 发生冲突，则即使 UC0RX 寄存器内没有存储数据，也可能会产生 INTUC0R 信号。
为了完成接收而无需等待 INTUC0R 信号的产生，请确保将 UC0RIC 寄存器的中断请求标志(UC0RIF)清除(0)，在中断控制寄存器（UC0RIC）的中断屏蔽标志(UC0RMK)置位(1)之后，然后将 UC0PWR 位或 UC0RXE 位设置为 0。

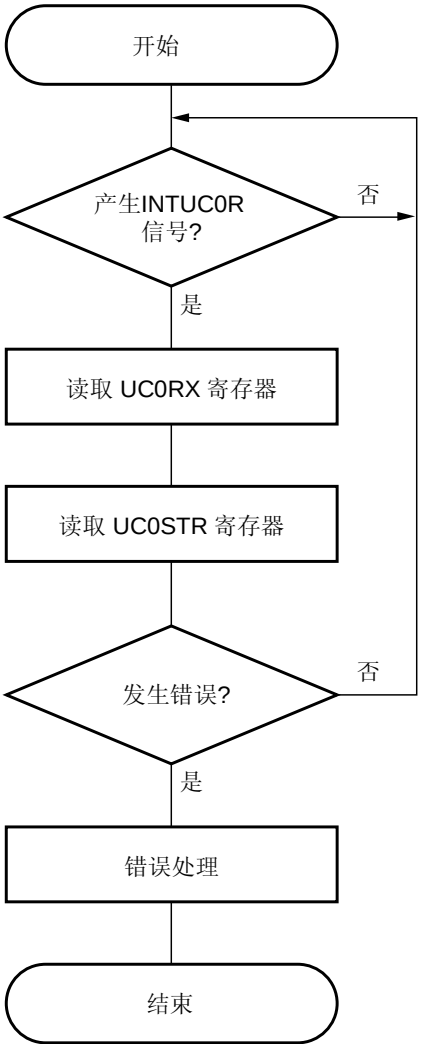
17.6.5 接收错误

接收期间可能发生的错误有 3 种类型：奇偶检验错误、帧错误以及溢出错误。数据接收结果错误标志被设置在 UC0STR 寄存器中，当错误发生时，输出接收完成中断请求信号 (INTUC0R)。

通过读取 UC0STR 寄存器的内容，可以确定在接收过程中发生了哪种错误 。

读取完成后，写入 0 来清除接收错误标志。

图 17-8. 接收数据读取流程



注意事项 发生 INTUC0R 信号时，必须读取 UC0STR 寄存器来检查错误。

表 17-5. 导致接收错误的原因

错误标志	接收错误	原因
UC0PE	奇偶错误	接收到的校验位和设定校验位不匹配
UC0FE	帧错误	未检测到停止位
UC0OVE	溢出错误	在数据从接收缓冲读出之前，下一个数据接收完成

发生接收错误时，根据错误类型执行以下处理步骤：

- 奇偶错误
  如果是因为例如接收线上的噪声问题导致接收到错误数据，则丢弃接收到的数据并重新发送。

- 帧错误
  接收端和发送端之间可能发生波特率错误，或者错误的检测到起始位。因为对通信格式来说这是致命错误，确认发送端的操作已经停止，对发送和接收两端各自进行初始化处理，然后重新开始通信。

- 溢出错误
  因为在读出接收数据之前，下一个数据接收完成，1 帧数据会丢弃。如果需要该数据，则需进行重新发送。

注意事项 在接收时，在下一个数据接收完成之前，必须读取 UC0STR 寄存器，以检查是否发生错误，如果有错误发生，则执行错误处理。

17.6.6 奇偶类型和操作

奇偶位用于检测通信数据中的位错误。通常情况下，发送端和接收端应该使用相同的奇偶校验。
在使用奇校验和偶校验情况下，可以检测到奇数计数的位错误。而在 0 奇偶校验和无校验情况下，检测不到错误。

(a) 偶校验

(i) 在发送期间

发送数据之中位值为“1”的数量，包括奇偶校验位，控制其数量为偶数。奇偶位的值如下：

- 发送数据中位值为“1”的数量为奇数： 1
- 发送数据中位值为“1”的数量为偶数： 0

(ii) 在接收期间

计算接收数据中位值为“1”的数量，包括校验位，如果为奇数，则输出奇偶校验错误。

(b) 奇校验

(i) 在发送期间

与偶校验相对，发送数据中位值为“1”的数量，包括奇偶校验位，控制其数量为奇数。奇偶位的值如下：

- 发送数据中位值为“1”的数量为奇数： 0
- 发送数据中位值为“1”的数量为偶数： 1

(ii) 在接收期间

计算接收数据中位值为“1”的数量，包括校验位，如果为偶数，则输出奇偶校验错误标志。

(c) 0 奇偶校验

在发送期间，奇偶校验位总是“0”，而与发送数据无关。
在接收期间，不执行奇偶校验。因而，无论奇偶位为“0” 或 “1”，均不会产生奇偶错误。

(d) 无奇偶校验

不向发送数据附加奇偶位。
设定无奇偶位进行接收。因为无奇偶位所以没有奇偶错误发生。

注意事项 使用 LIN 功能时，将 UC0CTL0 寄存器的 UC0PS1 位和 UC0PS0 位固定为 00。

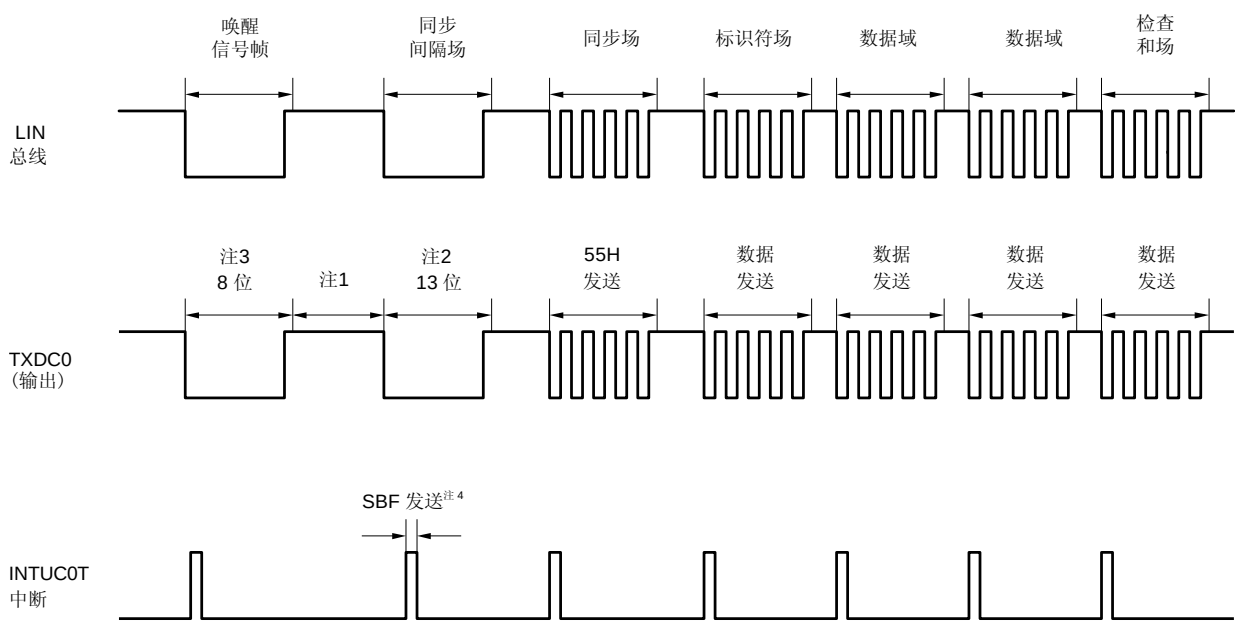
17.6.7 LIN发送/接收格式

V850ES/JG3-L 具有 SBF (同步间隔场)发送 /接收控制功能，可以支持 LIN 功能的使用。

备注 LIN 表示局域内联网，是一种低速（1 至 20 kbps）串行通信协议，有助于降低汽车网络的成本。
LIN 通信为单主机通信，一台主机最多可以连接 15 台从机。
LIN 从机一般是控制开关、传动装置和传感器，且这些装置可通过 LIN 网络连接到 LIN 主机上。
通常，LIN 主机接入诸如 CC0（控制器区域网络）之类的网络。
另外，LIN 总线使用单线方式且通过遵循 ISO9141 规范的收发器连接至各个节点。
LIN 协议中，主机以某一波特率信息发送一个数据帧，从机将其接收并校正波特率错误。因此，从机波特率错误在±15%范围内，通信可以正常进行。

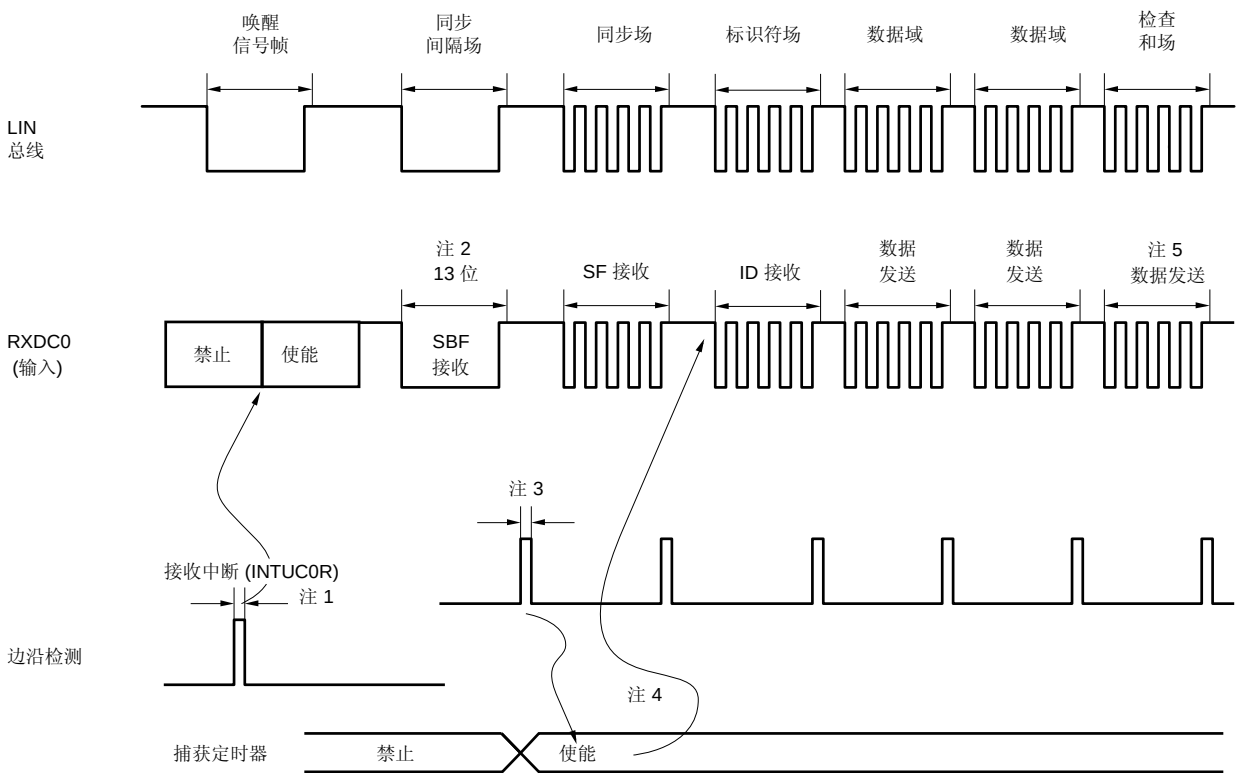
图 17-9 和 17-10 表明 LIN 的发送和接收的处理。

图 17-9. LIN 发送格式



注
1. 各场之间的间隔由软件控制。
2. SBF 输出通过硬件执行。输出宽度即位长度，由 UC0OPT0.UC0SBL2 至 UC0OPT0.UC0SBL0 位设置。如果需要调整输出宽度，可使用 UC0CTLn.UC0BRS7 至 UC0CTLn.UC0BRS0 位进行调整。
3. 8 位模式下，使用 80H 替代唤醒信号帧。
4. 发送使能中断请求信号（INTUC0T）在每次发送开始时输出。INTUC0T 信号同样在每次 SBF 发送开始时输出。

图 17-10. LIN 接收格式



注

1. 引脚边沿检测器检测到唤醒信号，使能 UARTC0，设置 SBF 接收模式。
2. 接收操作会持续到检测到停止位。检测到 11 位或更多位的 SBF 接收，则判断认为正常 SBF 接收结束，同时输出中断信号。检测到的 SBF 接收少于 11 位，则判断为 SBF 接收错误，没有中断信号输出，并且返回 SBF 接收模式。
3. 如果 SBF 接收正常结束，则输出中断请求信号。通过 SBF 接收完成中断来使能定时器。此外，关于 UC0STR.UC0OVE、UC0STR.UC0PE 和 UC0STR.UC0FE 位的错误不作检测，并且不执行 UART 通信错误检测处理，UARTC0 接收移位寄存器的数据不向 UC0RX 寄存器传输。UARTC0 接收移位寄存器保持初始值 FFH。
4. RXDC0 引脚连接至定时器的 T1（捕获输入），计算得出传输速率，波特率误差。通过校正的波特率误差设置 UC0CTL2 寄存器，UARTC 使能变低再次置位，状态变为接收状态。
5. 通过软件可以识别校验和场。在校验和场的接收之后执行 UARTC0 初始化，通过软件重新指定 SBF 接收模式的处理。

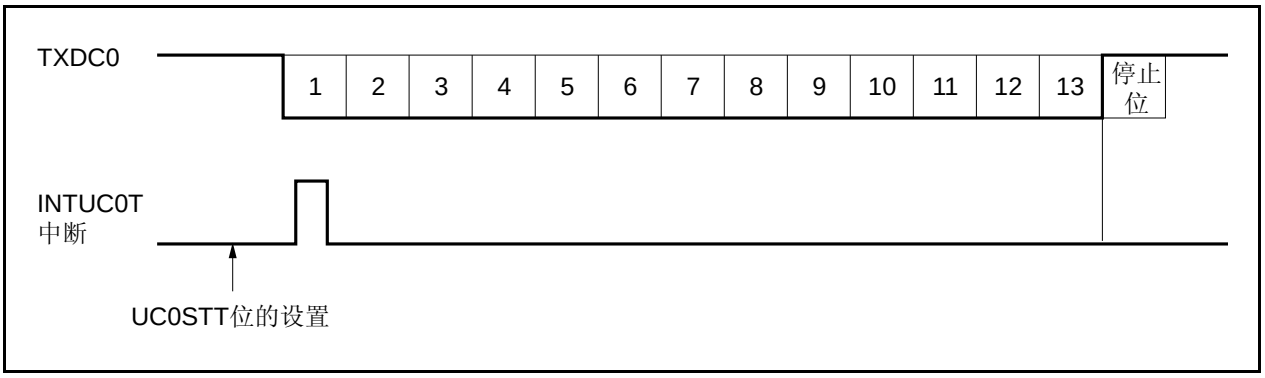
17.6.8 SBF发送

当 UC0CTL0.UC0PWR 位和 UC0CTL0.UC0TXE 位都为 1 时，进入发送使能状态，通过设置 SBF 发送触发信号（UC0OPT0.UC0STT 位）为 1，开始 SBF 发送。

之后，输出由 UC0OPT0.UC0SLS2 至 UC0OPT0.UC0SLS0 位指定的 13 位至 20 位宽度的低电平信号。SBF 发送开始时，产生发送使能中断请求信号（INTUC0T）。在 SBF 发送结束后，UC0STT 位自动清零。之后，恢复 UART 发送模式。

暂停发送，直到下一个待发送数据写入 UC0TX 寄存器，或者直到 SBF 发送触发信号（UC0STT 位）被置位。

图 17-11. SBF 发送示例



17.6.9 SBF接收

设置 UC0CTL0.UC0PWR 位为 1，然后设置 UC0CTL0.UC0RXE 位为 1，进入接收使能状态。

将 SBF 接收触发（UC0OPT0.UC0STR 位）设置为 1，可以设置 SBF 接收等待状态。

SBF 接收等待状态下，RXDC0 引脚被监控并且开始检测起始位。和 UART 接收等待状态类似。

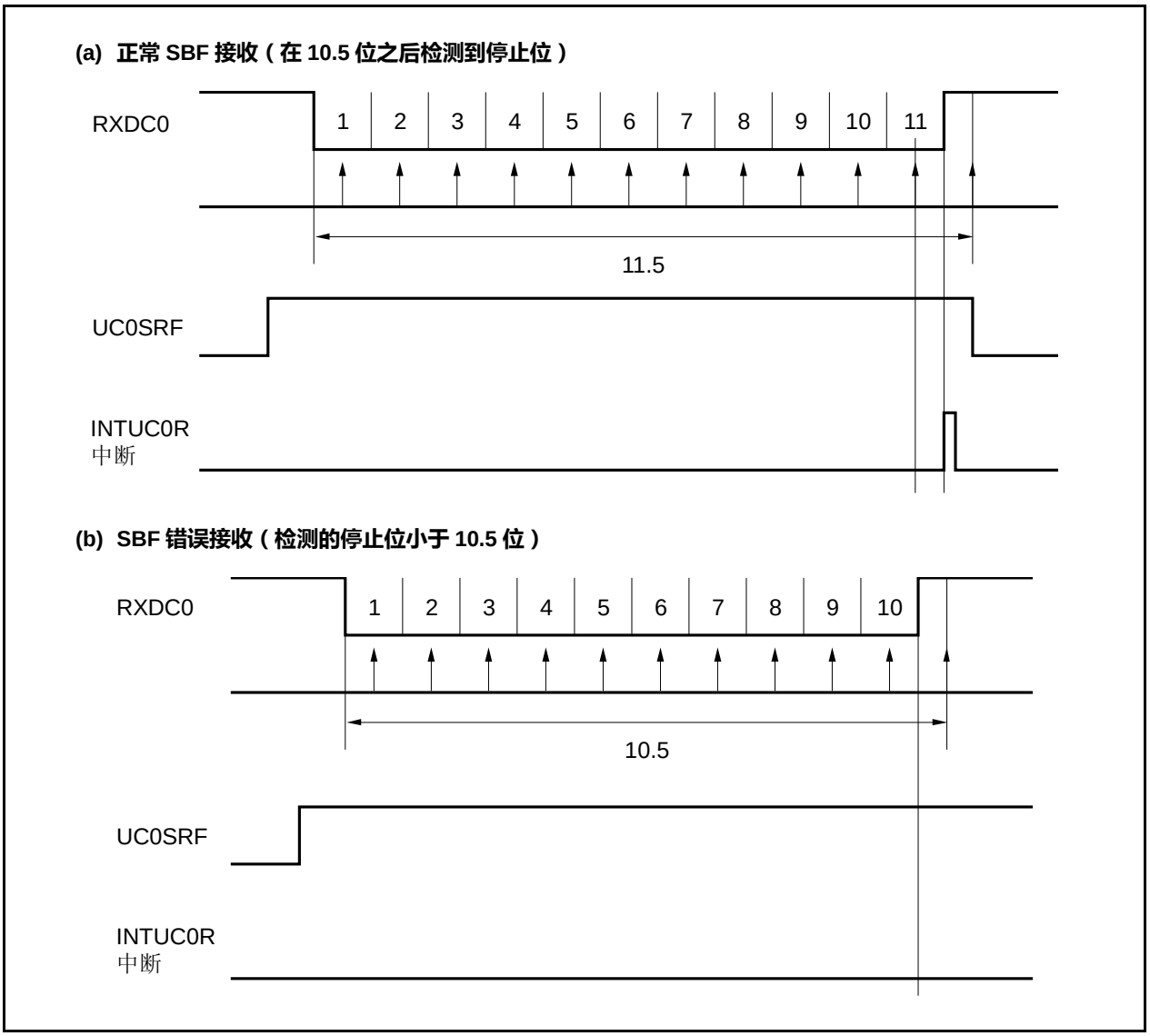
在检测到起始位之后，根据设置的波特率开始接收并且内部计数器开始向上计数。

接收到一个停止位时，如果 SBF 宽度为 11 位或更多位，则判断为正常操作，输出接收完成中断请求信号（INTUC0R）。UC0OPT0.UC0SRF 位自动清除，SBF 接收结束。对 UC0STR.UC0OVE、UC0STR.UC0PE 和 UC0STR.UC0FE 位的错误不作检测，不执行 UART 通信错误检测处理，不执行 UARTC0 接收移位以及向 UC0RX 寄存器的数据传输，UARTC0 接收移位寄存器保持初始值 FFH。如果 SBF 宽度为 10 位或更少位，则接收以错误终止，不产生中断，并返回 SBF 接收模式。此时，UC0SRF 位不被清除。

注意事项 1. 如果数据接收期间 SBF 被发送，会发生一个帧错误。

2. 在 SBF 接收（UC0SRF = 1）期间，不要将 SBF 接收触发位（UC0SRT）和 SBF 发送触发位（UC0STT）设置为 1。

图 17-12. SBF 接收



17.6.10 接收数据噪声过滤器

该滤波器使用专用波特率发生器提供的基准时钟，通过 RXDC0 引脚对接收到的信号进行采样。

如果读到两次相同的采样值，则匹配检测器输出改变，并且被采样的 RXDC0 信号作为输入数据。所以不超过 2 个时钟宽度的数据被判定为噪声，不会被送交给内部电路(参见 **图 17-13**)。有关基准时钟，参见 17.7 (1) (a) 基准时钟 。

另外，由于电路如图 17-15 所示，所以，接收操作的处理过程与外部信号状态相比，延迟了 3 个时钟周期。

图 17-13. 噪声过滤电路

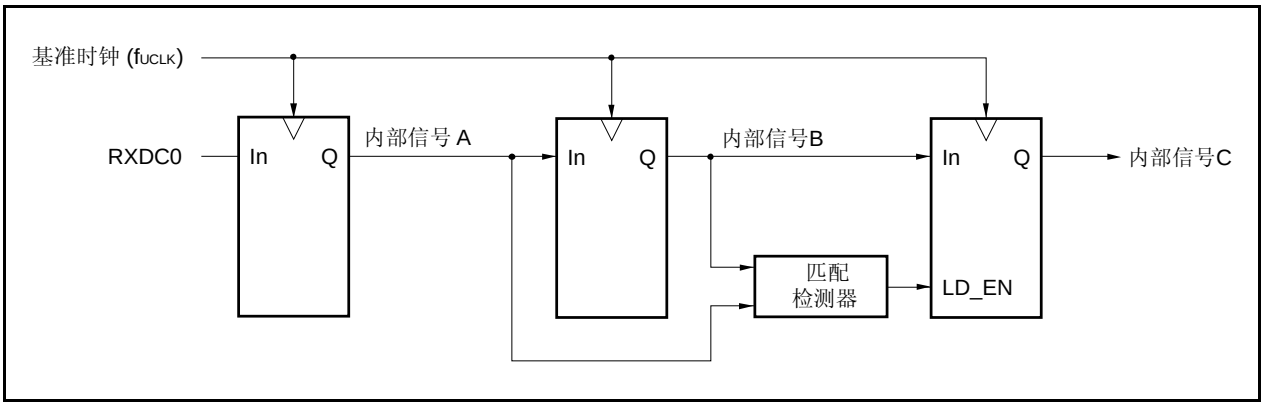
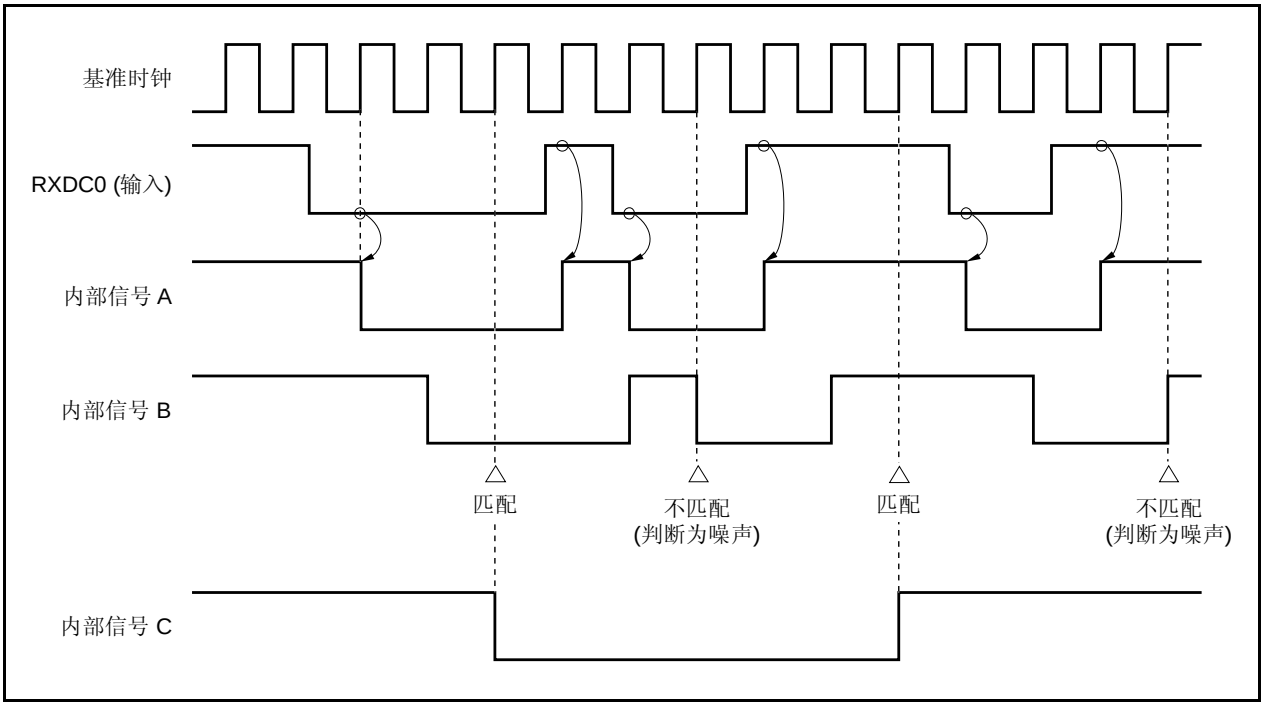


图 17-14. RXDC0 信号判定为噪声的时序



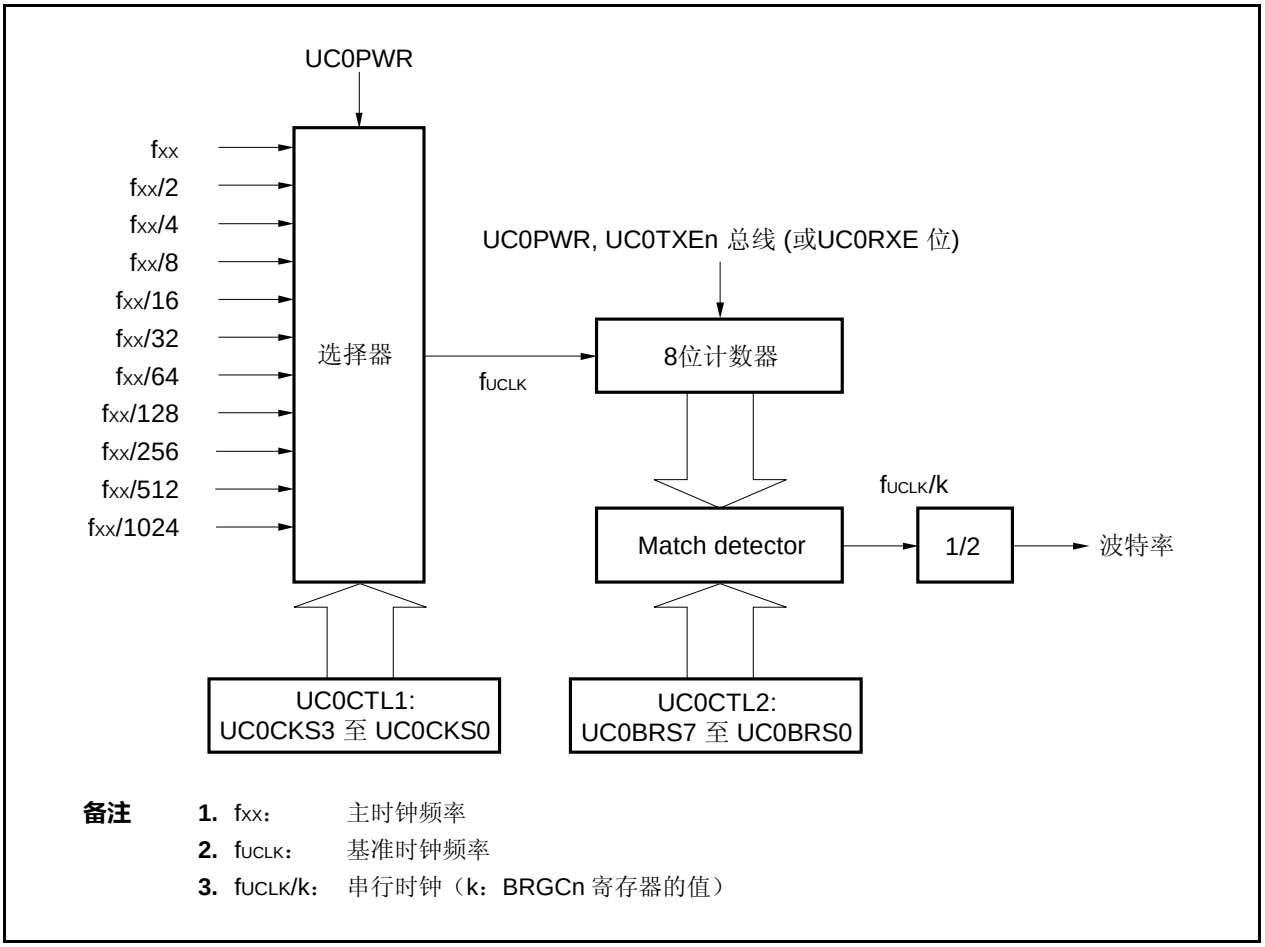
17.7 专用波特率发生器

专用波特率发生器包括一个时钟源选择器和一个 8 位可编程计数器模块，在使用 UARTC0 发送/接收期间产生串行时钟。关于串行时钟，对各个通道可以选择专用波特率发生器输出。

一个 8 位计数器用于发送，另一个用于接收。

(1) 波特率发生器的配置

图 17-15. 波特率发生器的配置



备注 **1.** f_{XX}： 主时钟频率
2. f_{UCLK}： 基准时钟频率
3. f_{UCLK}/k： 串行时钟（k：BRGCn 寄存器的值）

(a) 基准时钟

当 UC0CTL0.UC0PWR 位为 1 时，由 UC0CTL1.UC0CKS3 至 UC0CTL1.UC0CKS0 位选择的时钟被提供给 8 位计数器。该时钟称为基准时钟(f_{UCLK})。

(b) 串行时钟的产生

通过设置 UC0CTL1 寄存器和 UC0CTL2（n = 0 至 2）寄存器，来产生一个串行时钟。
基准时钟由 UC0CTL1.UC0CKS3 至 UC0CTL1.UC0CKS0 位选择。
8 位计数器的分频值可以使用 UC0CTL2.UC0BRS7 至 UC0CTL2.UC0BRS0 位进行设置。
波特率时钟由串行时钟 2 分频产生。

(2) UARTC0 控制寄存器 1 (UC0CTL1)

UC0CTL1 寄存器是 8 位寄存器，用于选择 UARTC0 基准时钟。

该寄存器可进行字节读取或写入。

系统复位后，该寄存器被设为 00H。

注意事项 在改写 UC0CTL1 寄存器之前，将 UC0CTL0.UC0PWR 位清 0。

复位后：00H R/W 地址： FFFFFAA1H

	7	6	5	4	3	2	1	0
UC0CTL1	0	0	0	0	UC0CKS3	UC0CKS2	UC0CKS1	UC0CKS0

UC0CKS3	UC0CKS2	UC0CKS1	UC0CKS0	基准时钟 (f_{UCLK}) 选择
0	0	0	0	fxx
0	0	0	1	fxx/2
0	0	1	0	fxx/4
0	0	1	1	fxx/8
0	1	0	0	fxx/16
0	1	0	1	fxx/32
0	1	1	0	fxx/64
0	1	1	1	fxx/128
1	0	0	0	xx/256
1	0	0	1	fxx/512
1	0	1	0	fxx/1,024
其它				禁止设置

备注 fxx： 主时钟频率

(3) UARTC0 控制寄存器 2 (UC0CTL2)

UC0CTL2 寄存器是 8 位寄存器，用于选择 UARTC0 的波特率(串行传输速度) 。

波特率时钟的产生是对串行时钟 2 分频得到，由该寄存器指定。

该寄存器可进行字节读取或写入。

系统复位后，该寄存器被设为 FFH。

注意事项 在改写 UC0CTL2 寄存器之前，或将 UC0CTL0.UC0PWR 位清除为 0，或将 UC0TXE 和 UC0RXE 位清除为 00。

复位后：FFH R/W 地址： FFFFFAA2H

	7	6	5	4	3	2	1	0
UC0CTL2	UC0BRS7	UC0BRS6	UC0BRS5	UC0BRS4	UC0BRS3	UC0BRS2	UC0BRS1	UC0BRS0

UC0 BRS7	UC0 BRS6	UC0 BRS5	UC0 BRS4	UC0 BRS3	UC0 BRS2	UC0 BRS1	UC0 BRS0	默认 (k)	串行 时钟
0	0	0	0	0	0	×	×	×	禁止设置
0	0	0	0	0	1	0	0	4	$f_{UCLK}/4$
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 UC0CTL1.UC0CKS3 至 UC0CTL1.UC0CKS0 位选择的时钟频率

(4) 波特率

波特率误差由下列公式计算。

$$波特率 = \frac{f_{XX}}{2^{m+1} \times k} \text{ [bps]}$$

备注 f_{UCLK} =由 UC0CTL1.UC0CKS3 至 UC0CTL1.UC0CKS0 位选择的基准时钟频率

f_{XX}： 主时钟频率

m =由 UC0CTL1.UC0CKS3 至 UC0CTL1.UC0CKS0 位设置的值（m = 0 至 10）

k = 由 UC0CTL2.UC0BRS7 至 UC0CTL2.UC0BRS0 位设置的值（k = 4 至 255）

波特率误差由下列公式计算：

$$误差(\%) = \left(\frac{f_{XX}}{2^{m+1} \times k \times 目标波特率} - 1 \right) \times 100 \text{ [\%]}$$

注意事项 1. 发送过程中，波特率误差必须在接收端的误差容限之内。

2. 接收时的波特率误差满足一定范围，该范围参见“（5）接收中允许的波特率误差范围”。

要设置波特率，按以下算式计算，设置 UC0CTL1 和 UC0CTL2 寄存器（使用内部时钟时）。

<1> 设置 k 等于 fxx/(2 × 目标波特率)，m 等于 0。
<2> 如果 k 等于 256 或更大（k ≥ 256），将 k 减小一半（k/2），m 加 1（m + 1）。
<3> 重复第<2>步，直到 k 小于 256 (k < 256)。
<4> 对 k 的第一位小数四舍五入，得到最接近的整数。
如果四舍五入后 k 为 256，将 k 减半并对 m 递增 1(m + 1)获得 k = 128。
<5> 将 m 的值设置到 UC0CTL1 寄存器，k 的值设置到 UC0CTL2 寄存器。

示例： 如果 f_{XX} = 20 MHz 而目标波特率 = 153,600 bps
<1> k = 20,000,000/(2 × 153,600) = 65.10…， m = 0
<2>， <3> k = 65.10… < 256， m = 0
<4> 设置 UC0CTL2 寄存器的值： k = 65 = 41H，设置 UC0CTL1 寄存器的值： m = 0

实际波特率= 20,000,000/(2 × 65)
= 153,846 [bps]

波特率误差= {20,000,000/(2 × 65 × 153,600) − 1} × 100
= 0.160 [%]

以下为波特率设置的典型示例：

表 17-6. 波特率发生器设置数据

波特率	f_{XX} = 20 MHz			f_{XX} = 16 MHz			f_{XX} = 10 MHz		
(bps)	UC0CTL1	UC0CTL2	ERR (%)	UC0CTL1	UC0CTL2	ERR (%)	UC0CTL1	UC0CTL2	ERR (%)
300	08H	82H	0.16	07H	D0H	0.16	07H	82H	0.16
600	07H	82H	0.16	06H	D0H	0.16	06H	82H	0.16
1,200	06H	82H	0.16	05H	D0H	0.16	05H	82H	0.16
2,400	05H	82H	0.16	04H	D0H	0.16	04H	82H	0.16
4,800	04H	82H	0.16	03H	D0H	0.16	03H	82H	0.16
9,600	03H	82H	0.16	02H	D0H	0.16	02H	82H	0.16
19,200	02H	82H	0.16	01H	D0H	0.16	01H	82H	0.16
31,250	01H	A0H	0	01H	80H	0	00H	A0H	0
38,400	01H	82H	0.16	00H	D0H	0.16	00H	82H	0.16
76,800	00H	82H	0.16	00H	68H	0.16	00H	41H	0.16
153,600	00H	41H	0.16	00H	34H	0.16	00H	21H	−1.36
312,500	00H	20H	0	00H	1AH	−1.54	00H	10H	0
625,000	00H	10H	0	00H	0DH	−1.54	00H	08H	0

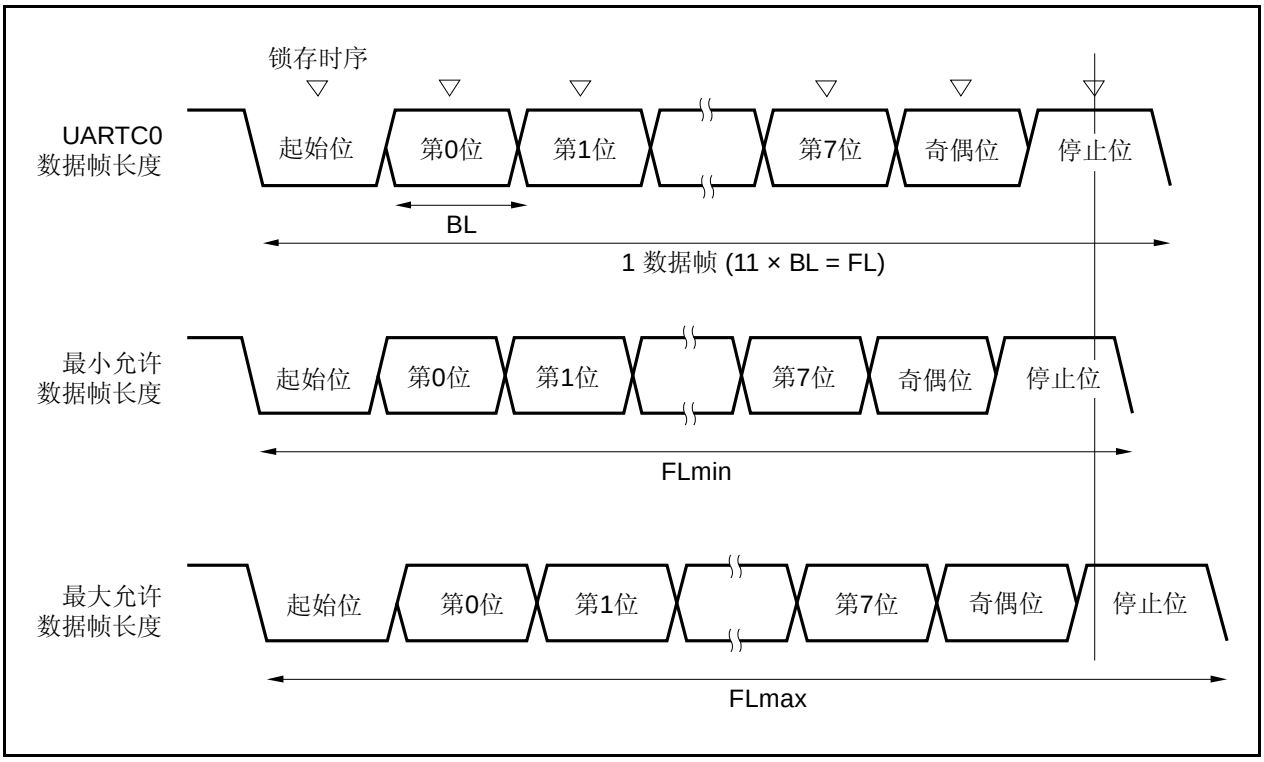
备注 f_{XX}： 主时钟频率
ERR：波特率误差 (%)

(5) 在接收期间允许的波特率范围

接收过程中，目的端允许的波特率误差范围如下所示：

注意事项 下面的波特率误差表示理论值。在实际操作中，信号可能会畸变，或即使误差在可允许范围内，通信却没有正确执行。因此，误差必须最小化。

图 17-16. 接收期间允许的波特率范围



如图 17-16 所示，接收数据的锁存时序取决于检测到起始位后 UC0CTL2 寄存器对计数器的设置值。如果在此锁存时序可以及时接收直到最末位数据（停止位），发送数据就可以被正常接收。

当应用于 11 位接收时，以下为理论结果。

BL = (波特率)$^{-1}$

波特率：UARTAn 波特率(n = 0 至 2)

k： UAnCTL2.UAnBRS7 至 UAnCTL2.UAnBRS0 位的设定值(n = 0 至 2) 。

BL： 1 位数据长度

FL： 1 个数据帧的长度

锁存时间余量：2 个时钟

最小允许数据帧长度： $FLmin = 11 \times BL - \frac{k-2}{2k} \times BL = \frac{21k+2}{2k} BL$

所以，可以从目的端接收的最大波特率如下所示：

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2}\ Brate$$

同理，可以得到下列公式来计算最大允许传输速率：

$$\frac{10}{11} \times FLmax = 11 \times BL - \frac{k + 2}{2 \times k} \times BL = \frac{21k - 2}{2 \times k}\ BL$$

$$FLmax = \frac{21k - 2}{20\ k}\ BL \times 11$$

所以，可以从目的端接收的最小波特率如下所示：

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2}\ Brate$$

由上述公式计算 UARTC0 以及目的端的允许波特率误差，得到的最大和最小波特率值必须遵从以下表格。

表 17-7. 允许的最大/最小波特率误差（11 位长度）

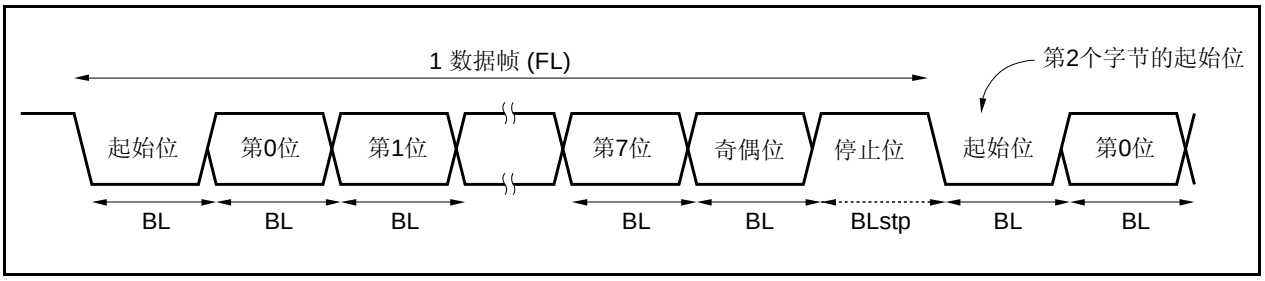
分频比(k)	允许的最大波特率误差	允许的最小波特率误差
4	+2.32%	- 2.43%
8	+3.53%	- 3.61%
20	+4.26%	- 4.31%
50	+4.56%	- 4.58%
100	+4.66%	- 4.67%
255	+4.72%	- 4.73%

备注
1. 接收精度依赖于一帧内的位数量、基准时钟频率(f_{UCLK})和分频比(k)。基准时钟频率(f_{UCLK})越高且分频比(k)越大，则精度越高。
2. k： UC0CTL2.UC0BRS7 位至 UC0CTL2.UC0BRS0 位的设定值(n = 0 至 2)。

(6) 连续发送期间的数据帧长度

连续发送期间，从停止位到下一个起始位的传输间隔通常比平常多 2 个基准时钟周期。而时序的初始化由接收端通过检测起始位执行，因此，这对传输结果没有影响。

图 17-17. 连续发送期间的数据帧长度



假定 1 位数据长度为 BL；停止位长度为 FLstp；基准时钟频率为 f_{UCLK}，可以得到如下等式：

$$BLstp = BL + 2/f_{UCLK}$$

所以，连续发送期间的传输速率如下。

$$数据帧长度 = 11 \times BL + (2/f_{UCLK})$$

17.8 注意事项

(1) 当提供给 UARTC0 的时钟停止(比如进入 IDLE1、IDLE2 或 STOP 模式)时，则操作停止，每一个寄存器都保留时钟供应停止前的瞬间值。TXDC0 引脚输出也保持时钟供给停止前的瞬时值。然而，恢复时钟供给后，不能保证正常操作。所以，在时钟恢复之后，应该将 UC0CTL0.UC0PWR 位、UC0CTL0.UC0RXEn 位和 UC0CTL0.UC0TXEn 位设置为 000，对电路进行初始化。

(2) UARTC0 中，不会产生由通信错误引起的中断。使用 DMA 传输功能进行发送数据和接收数据的传输时，即使传输期间发生错误（奇偶校验错误，溢出错误，帧错误），也不执行错误处理。或者在 DMA 传输完成后读取 UC0STR 寄存器检查是否有错误，或者在通信期间读取 UC0STR 寄存器来检查错误。

(3) 按以下步骤启动 UARTC0。
<1> 将 UC0CTL0.UC0PWR 位置为 1。
<2> 设置此端口。
<3> 将 UC0CTL0.UC0TXE 位和 UC0CTL0.UC0RXE 位置为 1。

(4) 按以下步骤停止 UARTC0。
<1>将 UC0CTL0.UC0TXE 位和 UC0CTL0.UC0RXE 位清为 0。
<2>设置端口，并设置 UC0CTL0.UC0PWR 位为 0（如果端口设置不改变，则不存在该问题）

(5) 发送模式下（UC0CTL0.UC0PWR 位 = 1 且 UC0CTL0.UC0TXE 位 = 1），因为写入 UC0TX 寄存器会开始发送，所以不要用软件向该寄存器写入相同的值。若连续发送相同值，则重写相同的值。

(6) 在连续发送中，从停止位到下一个起始位的传输间隔通常比平常多 2 个基准时钟周期。但是，接收端通过检测起始位进行时序的初始化，因此接收结果不受影响。

(7) 如果低电平信号持续输入 RXDC0 引脚，UARTC 无法识别起始位。

(8) RXDC0 和 SIB1 引脚不能同时使用。该引脚用作 RXDC0 时，停止 CSIB0 接收（将 CB1CTL0.CB1RXE 位清除为 0）。该引脚用作 SIB1 时，停止 UARTC0 接收（将 UC0CTL0.UC0RXE 位清除为 0）。

第十八章 3-线可变长度串行 I/O（CSIB）

18.1 特征

- 3 线串行接口
  - SOBn: 串行数据输出
  - SIBn: 串行数据输入
  - $\overline{\text{SCKBn}}$: 串行时钟 I/O

  可以指定为发送模式，接收模式，发送/接收模式。
- 传输速率：8 Mbps 最快
- 可以选择主机模式和从机模式
- 2 个中断请求信号：
  - 接收完成中断（INTCBnR）：在接收使能时接收数据被传输到 CBnRX 寄存器时，或发生过载错误时，产生该中断。在单次传输模式下，即使只执行发送，在发送完成时，产生该中断。
  - 发送使能中断（INTCBnT）：在连续发送或连续发送/接收模式下，当发送数据从 CBnTX 寄存器传输时，产生该中断，且可以将数据写入到 CBnTX 寄存器。
- 可以通过 $\overline{\text{SCKBn}}$ 指定数据发送/接收的时序
- 传输数据长度可以在 8 位至 16 位之间逐位选择
- 传输数据可以在 MSB 先行和 LSB 先行之间切换
- 对发送和接收都是双缓冲配置
- 过载错误检测

备注 n = 0 至 4

18.2 配置

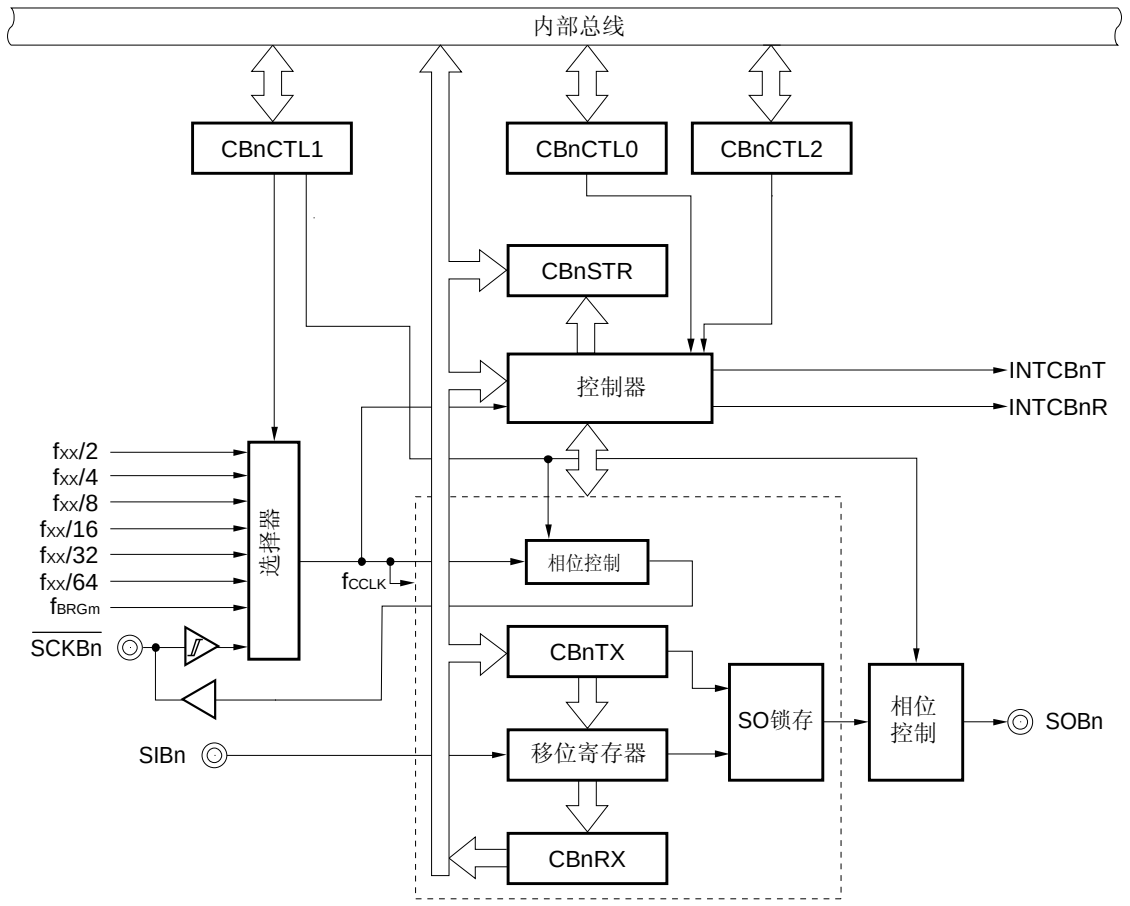
CSIBn 包括下列硬件：

表 18-1. CSIBn 的配置

项目	配置
寄存器	CSIBn 接收数据寄存器（CBnRX） CSIBn 发送数据寄存器（CBnTX） 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）

以下显示 CSIBn 的功能框图。

图 18-1. CSIBn 框图



备注 f_{CCLK}： 通信时钟
f_{XX}： 主时钟频率
f_{BRGm}： BRGm 计数时钟
n = 0 至 4
m = 1（n = 0，1）
m = 2（n = 2，3）
m = 3（n = 4）

18.2.1 各个通道的引脚功能

V850ES/JG3-L 中 CSIB 使用的 SIBn，SOBn 和 SCKBn 引脚也可以用作其它功能，如表 18-2 所示。CSIB 要使用这些引脚，相关寄存器的设置描述如**表 4-15 端口引脚用作复用功能时的设置**。

表 18-2. CSIB 使用的引脚

通道	引脚编号		端口	CSIB 接收输入	CSIB 发送输出	CSIB 时钟 I/O	其它功能
	GC	F1					
CSIB0	22	K1	P40	SIB0	–	–	SDA01
	23	K2	P41	–	SOB0	–	SCL01
	24	L2	P42	–	–	$\overline{\text{SCKB0}}$	–
CSIB1	50	J11	P97	SIB1	–	–	A7/RXDC0/TIP20/TOP20
	51	J10	P98	–	SOB1	–	A8
	52	H11	P99	–	–	$\overline{\text{SCKB1}}$	A9
CSIB2	40	L8	P53	SIB2	–	–	KR3/TIQ00/TOQ00/RTP03/DDO
	41	K8	P54	–	SOB2	–	KR4/RTP04/DCK
	42	J8	P55	–	–	$\overline{\text{SCKB2}}$	KR5/RTP05/DMS
CSIB3	53	H10	P910	SIB3	–	–	A10
	54	H9	P911	–	SOB3	–	A11
	55	G11	P912	–	–	$\overline{\text{SCKB3}}$	A12
CSIB4	26	K4	P31	SIB4	–	–	RXDA0/INTP7
	25	L3	P30	–	SOB4	–	TXDA0
	27	L4	P32	–	–	$\overline{\text{SCKB4}}$	ASCKA0/TIP00/TOP00

备注 GC：100 针塑封 LQFP（密间距）（14 × 14）
F1：121 针塑封 FBGA（8 × 8）

18.3 CSIB和其它串行接口的模式转换

18.3.1 CSIB0 和 I^2C01 的模式转换

在 V850ES/JG3-L 中，CSIB0 和 I^2C01 是共享引脚的复用功能，因而不能同时使用。使用之前，用 PMC4 和 PFC4 寄存器预先设置 CSIB0。

注意事项 如果在发送或接收期间切换这些功能，则 CSIB0 和 I^2C01 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 18-2. CSIB0 和 I^2C01 模式转换设置

复位后: 00H R/W 地址: FFFFF448H

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

复位后: 00H R/W 地址: FFFFF468H

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

PMC4n	PFC4n	操作模式
0	x	端口 I/O 模式
1	0	CSIB0 模式
1	1	I^2C01 模式

备注 **1.** n = 0，1
2. × = 无需理会

18.3.2 CSIB4 和UARTA0 的模式转换

在 V850ES/JG3-L 中，CSIB4 和 UARTA0 是共享引脚的复用功能，因而不能同时使用。使用之前，用 PMC3、PFC3 和 PFCE3L 寄存器预先设置 CSIB4。

注意事项 如果在发送或接收期间切换这些功能，则 CSIB4 和 UARTA0 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 18-3. CSIB4 和 UARTA0 模式转换设置

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

	15	14	13	12	11	10	9	8
PMC3	0	0	0	0	0	0	PMC39	PMC38
	7	6	5	4	3	2	1	0
	0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

复位后： 0000H R/W 地址： FFFFF466H, FFFFF467H

	15	14	13	12	11	10	9	8
PFC3	0	0	0	0	0	0	PFC39	PFC38
	7	6	5	4	3	2	1	0
	0	0	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

复位后： 00H R/W 地址： FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3L	0	0	0	0	0	PFCE32	0	0

PMC32	PFCE32	PFC32	操作模式
0	×	×	端口 I/O 模式
1	0	0	ASCKA0 模式
1	0	1	$\overline{\text{SCKB4}}$ 模式

PMC3n	PFC3n	操作模式
0	×	端口 I/O 模式
1	0	UARTA0 模式
1	1	CSIB4 模式

备注 **1.** n = 0，1
2. × = 无需理会

18.4 寄存器

以下寄存器用来控制 CSIBn。

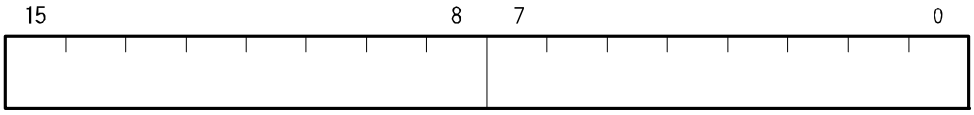
- CSIBn 接收数据寄存器（CBnRX）
- CSIBn 发送数据寄存器（CBnTX）
- CSIBn 控制寄存器 0（CBnCTL0）
- CSIBn 控制寄存器 1（CBnCTL1）
- CSIBn 控制寄存器 2（CBnCTL2）
- CSIBn 状态寄存器（CBnSTR）

（1）CSIBn 接收数据寄存器（CBnRX）

CBnRX 是 16-位缓存寄存器，用于存放接收数据。
该寄存器是只读寄存器，以 16 位宽度读取。
在接收模式下，通过读取 CBnRX 寄存器启动接收操作。
如果传输数据长度为 8 位，该寄存器的低 8 位作为寄存器 CBnRXL，且只能以 8 位宽度进行读取。
系统复位后，该寄存器被设为 0000H。
除复位输入外，清除 CBnCTL0 寄存器的 CBnPWR 位（为 0），也可以对 CBnRX 寄存器进行初始化。

复位后: 0000H　R　地址: CB0RX FFFFFD04H, CB1RX FFFFFD14H, CB2RX FFFFFD24H, CB3RX FFFFFD34H, CB4RX FFFFFD44H

CBnRX
(n = 0 至 4)



（2）CSIBn 发送数据寄存器（CBnTX）

CBnTX 寄存器是 16 位缓冲寄存器，用于写入 CSIBn 的发送数据。

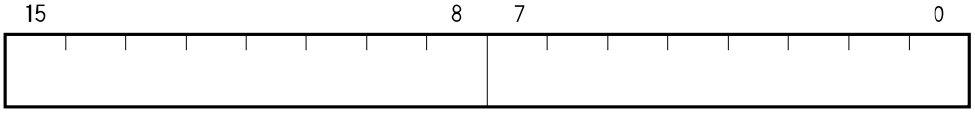
该寄存器可以 16 位宽度进行读取或写入。

当发送使能时，通过向 CBnTX 寄存器写入数据，可以启动发送操作。

如果传输数据长度为 8 位，该寄存器的低 8 位作为寄存器 CBnTXL，且只能以 8 位宽度进行读取。

系统复位后，该寄存器被设为 0000H。

复位后: 0000H　R/W　地址: CB0TX FFFFFD06H, CB1TX FFFFFD16H, CB2TX FFFFFD26H, CB3TX FFFFFD36H, CB4TX FFFFFD46H



CBnTX
(n = 0 至 4)

备注　通信开始条件如下所示。

发送模式（CBnTXE 位 = 1，CBnRXE 位 = 0）：	写入 CBnTX 寄存器
发送/接收模式（CBnTXE 位 = 1，CBnRXE 位 = 1）：	写入 CBnTX 寄存器
接收模式（CBnTXE 位 = 0，CBnRXE 位 = 1）：	从 CBnRX 寄存器读取数据

（3）CSIBn 控制寄存器 0（CBnCTL0）

CBnCTL0 寄存器是 8 位寄存器，用于控制 CSIBn 串行传输操作。
该寄存器可进行字节读写或按位读写。
系统复位后，该寄存器被设为 01H。

（1/3）

复位后: 01H　R/W　地址: CB0CTL0 FFFFFD00H, CB1CTL0 FFFFFD10H, CB2CTL0 FFFFFD20H, CB3CTL0 FFFFFD30H, CB4CTL0 FFFFFD40H

	<7>	<6>	<5>	<4>	3	2	1	<0>
CBnCTL0 (n = 0 至 4)	CBnPWR	CBnTXE注	CBnRXE注	CBnDIR注	0	0	CBnTMS注	CBnSCE

CBnPWR	CSIBn 操作禁止/使能的说明
0	禁止 CSIBn 操作并且清除 CBnSTR 寄存器
1	使能 CSIBn 操作
• CBnPWR 位控制 CSIBn 的操作，并复位内部电路。	

CBnTXE注	发送操作禁止/使能的说明
0	禁止发送操作
1	使能发送操作
• CBnTXE 位为 0时，SOBn输出低电平。	

CBnRXE注	接收操作禁止/使能的说明
0	禁止接收操作
1	使能接收操作
• CBnRXE 位清为0时，不会输出接收完成中断，即使发送来的数据是预定用于禁止接收操作，也不更新接收数据（CBnRX寄存器）。	

注　只有当 CBnPWR 位= 0 时，才可重写这些位。但是，在 CBnPWR 位置位的同时，这些位的值可以从 0 改变为 1。

注意事项　若要强制挂起发送/接收，将 CBnPWR 位清为 0，而不是清除 CBnRXE 和 CBnTXE 位。
此时，时钟输出停止。

（2/3）

CBnDIR注	传输方向模式 (MSB/LSB)的说明
0	MSB先行传输
1	LSB先行传输

CBnTMS注	传输模式说明
0	单次传输模式
1	连续发送模式

[单次传输模式]
通信完成后，发生接收完成中断 (INTCBnR)。
即使允许发送 (CBnTXE 位 = 1)，也不产生发送使能中断(INTCBnT)。
如果通信期间(CBnSTR.CBnTSF 位 = 1)写入下一个发送数据，此数据被忽略，且不开始下一次通信。同样如果设置为只接收通信(CBnTXE位=0，CBnRXE位=1)，即使通信期间读取接收数据(CBnSTR.CBnTSF位=1)，也不开始进行下一次通信。

[连续发送模式]
在通信期间（CBnSTR.CBnTSF位=1）写入下一个发送数据，使能连续发送。产生发送使能中断(INTCBnT)后，允许写入下一个发送数据。如果连续传输模式被设置为只接收通信(CBnTXE 位 = 0,CBnRXE位 = 1)，产生接收完成中断(INTCBnR)后，下一次接收连续开始，而不考虑CBnRX寄存器的读取操作。
因此，从CBnRX寄存器立即读取接收数据。如果此读取操作被延迟，则会发生溢出错误（CBnOVE位=1）。

注 只有当 CBnPWR 位为 0 时，这些位才可重写。但是，在 CBnPWR 位置位的同时，这些位的值可以从 0 改变为 1。

（3/3）

CBnSCE	开始传输禁止/使能的说明
0	通信启动触发无效
1	通信启动触发有效

• 主机模式下
该位允许或禁止通信启动触发。
(a) 单次发送或发送/接收模式或连续发送或连续发送/接收模式下
CBnSCE 位的设置对通信操作没有影响。
(b) 单次接收模式下
因为要通过读取接收数据（CBnRX寄存器）以开始接收，所以在
读取最后一个接收数据之前将 CBnSCE 位清除为 0，来禁止接收开始。注1
(c) 连续接收模式下
最后一个数据接收完成前一个通信时钟将 CBnSCE 位清除为 0，使得
最后一个数据接收注2完成后禁止启动接收。
• 从机模式下
设置CBnSCE 位为 1。

[CBnSCE 位的用法]
• 单次接收模式下
<1>最后一个数据接收完成时通过 INTCBnR 中断服务，在读取CBnRX
寄存器前清除 CBnSCE 位为 0 。
<2>确认 CBnSTR.CBnTSF 位 = 0后，清除 CBnRXE 位为 0 以禁止接收。
若要继续接收，通过伪读取CBnRX 寄存器将CBnSCE 位设置为 1
来启动下一次接收。
• 连续接收模式下
<1>最后一个数据接收期间通过INTCBnR中断服务来清除 CBnSCE 位为 0 。
<2>读取 CBnRX 寄存器。
<3>响应CBnTIR中断后读取CBnRX寄存器来读取最后一个接收数据。
<4>确认 CBnSTR.CBnTSF 位 = 0后，清除 CBnRXE 位为0 以禁止接收。
若要继续接收，通过伪读取CBnRX将CBnSCE 位设置为 1 以等待
下一次接收。

注 **1.** 如果在 CBnSCE 位被置 1 时读取 CBnRX 寄存器，则启动下一次通信操作。
2. 最末数据接收结束之前的一个通信时钟，如果 CBnSCE 位未被清除为 0，则自动开始下一次通信操作。

注意事项 请确保将第 3 位和第 2 位清为“0”。

（4）CSIBn 控制寄存器 1（CBnCTL1）

CBnCTL 是 8 位寄存器，用于指定 CSIBn 串行传输操作模式。

该寄存器可进行字节读写或按位读写。

系统复位后，该寄存器被设为 00H。

注意事项 仅当 CBnCTL0.CBnPWR 位为 0 时，才可以重写 CBnCTL1 寄存器。

复位后：00H R/W 地址: CB0CTL1 FFFFFD01H, CB1CTL1 FFFFFD11H,
CB2CTL1 FFFFFD21H, CB3CTL1 FFFFFD31H,
CB4CTL1 FFFFFD41H

	7	6	5	4	3	2	1	0
CBnCTL1	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

(n = 0 至 4)

	CBnCKP	CBnDAP	关于$\overline{\text{SCKBn}}$的数据发送/接收时序说明
通信类型 1	0	0	$\overline{\text{SCKBn}}$ (I/O) SOBn (输出) D7 D6 D5 D4 D3 D2 D1 D0 SIBn 捕获
通信类型 2	0	1	$\overline{\text{SCKBn}}$ (I/O) SOBn (输出) D7 D6 D5 D4 D3 D2 D1 D0 SIBn 捕获
通信类型 3	1	0	$\overline{\text{SCKBn}}$ (I/O) (输出) D7 D6 D5 D4 D3 D2 D1 D0 SIBn 捕获
通信类型 4	1	1	$\overline{\text{SCKBn}}$ (I/O) (输出) D7 D6 D5 D4 D3 D2 D1 D0 SIBn 捕获

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 ($\overline{\text{SCKBn}}$)	从模式

注 设置通信时钟（f_{CCLK}）为 8 MHz 或更低。

备注 当 n = 0，1，m = 1
当 n = 2，3，m = 2
当 n = 4，m = 3
关于 f_{BRGm} 详情，参见 **18.8 波特率发生器。**

（5）CSIBn 控制寄存器 2（CBnCTL2）

CBnCTL2 是 8 位寄存器，用于指定 CSIBn 串行传输数据长度。

该寄存器可进行字节读取或写入。

系统复位后，该寄存器被设为 00H。

注意事项 仅当 CBnCTL0.CBnPWR 位为 0 或 CBnTXE 和 CBnRXE 位均为 0 时，才可以重写 CBnCTL2 寄存器。

复位后: 00H R/W 地址: CB0CTL2 FFFFFD02H, CB1CTL2 FFFFFD12H, CB2CTL2 FFFFFD22H, CB3CTL2 FFFFFD32H, CB4CTL2 FFFFFD42H

	7	6	5	4	3	2	1	0
CBnCTL2 (n = 0 至 4)	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

CBnCL3	CBnCL2	CBnCL1	CBnCL0	串行寄存器位的长度
0	0	0	0	8 位
0	0	0	1	9 位
0	0	1	0	10 位
0	0	1	1	11 位
0	1	0	0	12 位
0	1	0	1	13 位
0	1	1	0	14 位
0	1	1	1	15 位
1	x	x	x	16 位

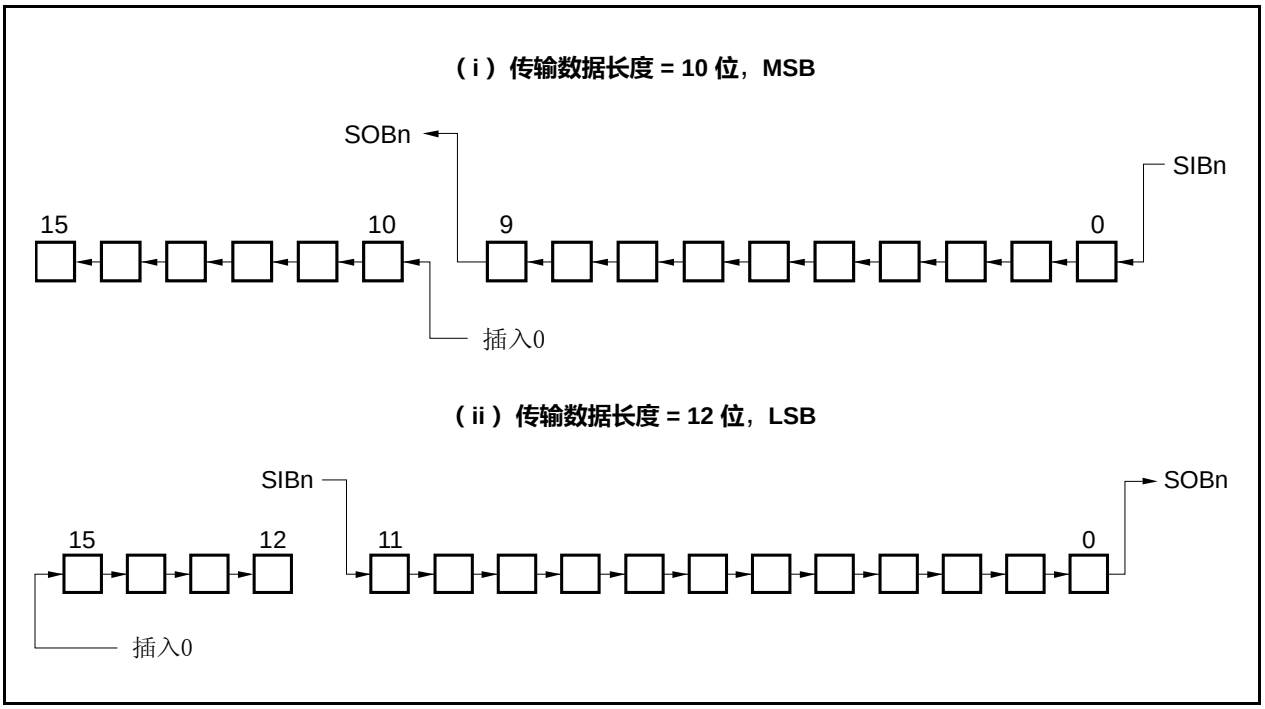
备注
1. 如果传输数据长度不是 8 位或 16 位，数据从 CBnTX 或 CBnRX 寄存器的 LSB 起进行填充。
2. ×：无需理会

（a） 改变传输数据长度

可以使用 CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位，在 8 位和 16 位之间以位为单位设置 CSIBn 传输的数据长度。

当设置的传输数据长度不是 16 位时，数据从 LSB 开始向 CBnTX 或 CBnRX 寄存器设置，而忽略传输起始位是 MSB 还是 LSB。没有使用的高位可以设置为任意数据，但是，在随后的串行传输中接收的对应数据变为 0。

图 18-4. 传输数据长度设置为非 16 位时的操作示例



（6）CSIBn 状态寄存器（CBnSTR）

CBnSTR 是 8 位寄存器，用于显示 CSIBn 的状态。

该寄存器可以按 8 位或 16 位宽度进行读取或写入，但是，CBnTSF 标志只读。

系统复位后，该寄存器被设为 00H。

除复位输入之外，可以通过清除 CBnCTL0.CBnPWR 位（为 0）对 CBnSTR 寄存器进行初始化。

复位后： 00H R/W 地址： CB0STR FFFFFD03H, CB1STR FFFFFD13H, CB2STR FFFFFD23H, CB3STR FFFFFD33H, CB4STR FFFFFD43H

	<7>	6	5	4	3	2	1	<0>
CBnSTR	CBnTSF	0	0	0	0	0	0	CBnOVE

(n = 0 至 4)

CBnTSF	通信状态标志
0	通信停止
1	通信进行中

• 发送期间，CBnTX寄存器中的数据准备好时，设置该寄存器。接收期间，进行CBnRX 寄存器伪读取时，设置该寄存器。
传输结束时，在时钟的最后有效沿将该标志清除为0。

CBnOVE	过载错误标志
0	无过载
1	过载

• 在接收期间或当接收操作完成时，下一个接收已经完成而CPU没有读取接收缓冲器的值，则发生过载错误。
这种情况下，CBnOVE 标志表示过载错误的发生。
• 即使单次传输模式下 CBnOVE 位同样有效。
因此，只用作发送时，需注意以下事项：
  • 不要检查 CBnOVE 标志。
  • 即使不需要接收数据，仍要读取该位。
• 写入0可清除 CBnOVE 标志。不能通过写入1对其进行置位

18.5 中断请求信号

CSIBn 可以产生以下 2 种中断请求信号：

- 接收完成中断请求信号（INTCBnR）
- 发送使能中断请求信号（INTCBnT）

这两种中断请求信号中，默认接收完成中断请求信号有较高优先级，而发送使能中断请求信号优先级较低。

表 18-3. 中断与其默认优先级

中断请求信号	优先级
接收完成	高
发送使能	低

（1）接收完成中断请求信号（INTCBnR）

接收使能状态下，接收数据传送到 CBnRX 寄存器，产生接收完成中断请求信号。

如果发生过载错误，同样也会产生中断请求信号。

响应接收完成中断请求信号并读取数据时，读取 CBnSTR 寄存器并检查接收结果是否错误。

单次传输模式中，即使只进行发送，传输完成后也会产生 INTCBnR 中断请求信号。

（2）发送使能中断请求信号（INTCBnT）

连续发送或连续发送/接收模式中，发送数据从 CBnTX 寄存器移出，并且一旦允许写入 CBnTX 寄存器，就会产生发送使能中断请求信号。

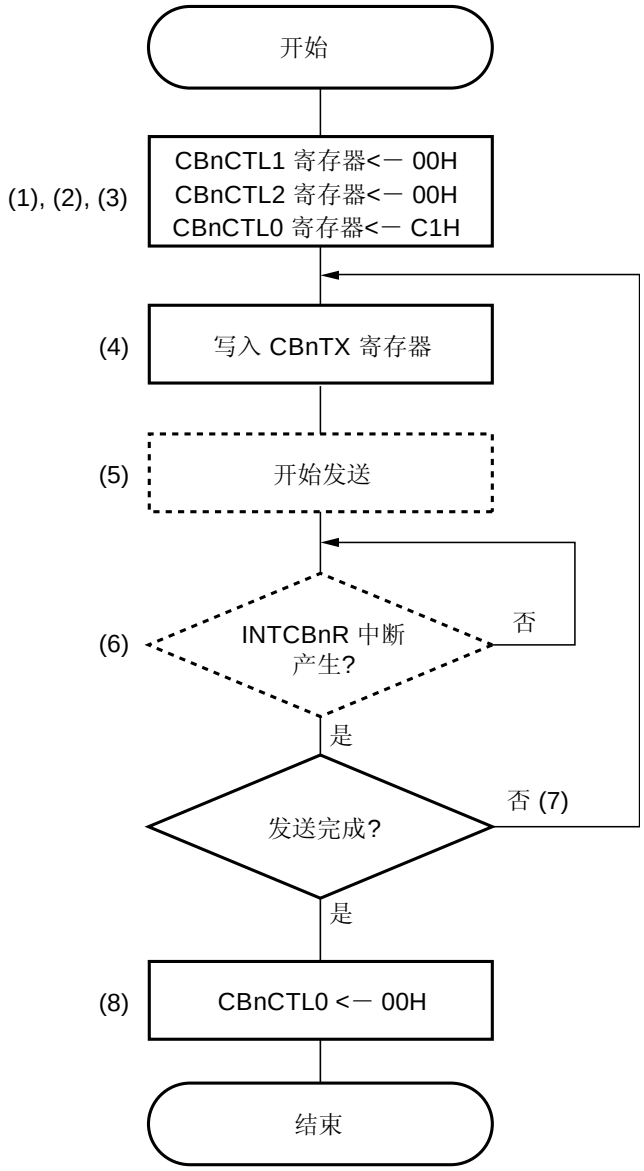
在单次发送和单次发送/接收模式中，不产生 INTCBnT 中断。

18.6 操作

18.6.1 单次传输模式（主机模式，发送模式）

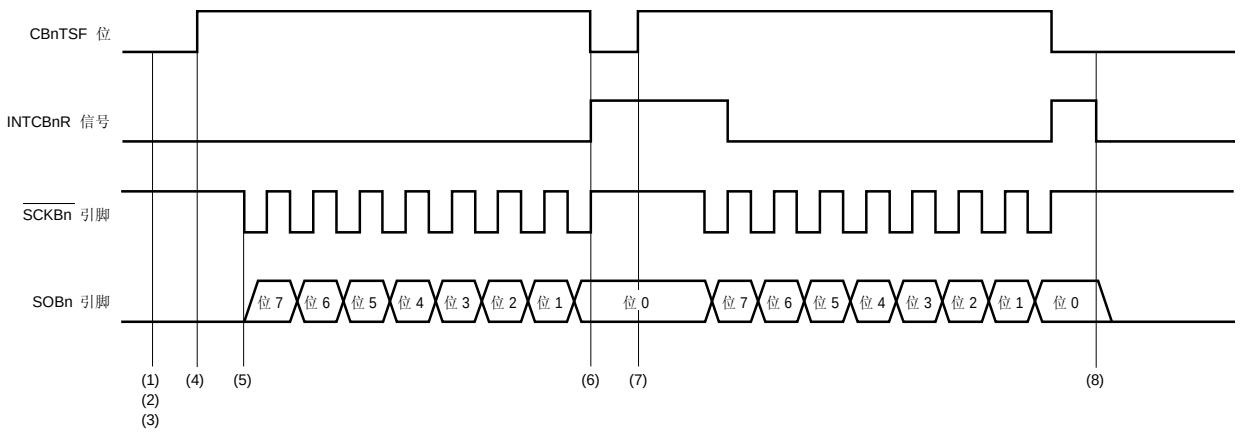
MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= f_{XX}/2（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

图 18-5. 单次传输模式操作（主机模式，发送模式）



备注 **1.** 虚线部分表示硬件处理。
2. 图中的数字对应于**图 18-6** 中的数字。
3. n = 0 至 4

图 18-6. 单次传输模式操作时序（主机模式，发送模式）



（1）向 CBnCTL1 寄存器写入 00H，选择通信类型 1，通信时钟（f_{CCLK}）= f_{XX}/2，且设置为主机模式。

（2）向 CBnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。

（3）向 CBnCTL0 寄存器写入 C1H，并在使能通信时钟（f_{CCLK}）的同时，选择为发送模式和 MSB 先行方式。

（4）将待发送数据写入 CBnTX 寄存器使得 CBnSTR.CBnTSF 位置为 1，启动发送。

（5）当发送开始后，串行时钟输出至 $\overline{\text{SCKBn}}$ 引脚，待发送数据与串行时钟同步，由 SOBn 引脚输出。

（6）当 CBnCTL2 寄存器设定的传输数据长度发送完成时，停止串行时钟和发送数据的输出，在串行时钟周期的最末边沿产生接收完成中断请求信号（INTCBnR），并将 CBnTSF 位清为 0。

（7）若继续进行发送，在 INTCBnR 信号产生之后，从（4）开始重复上述的步骤。

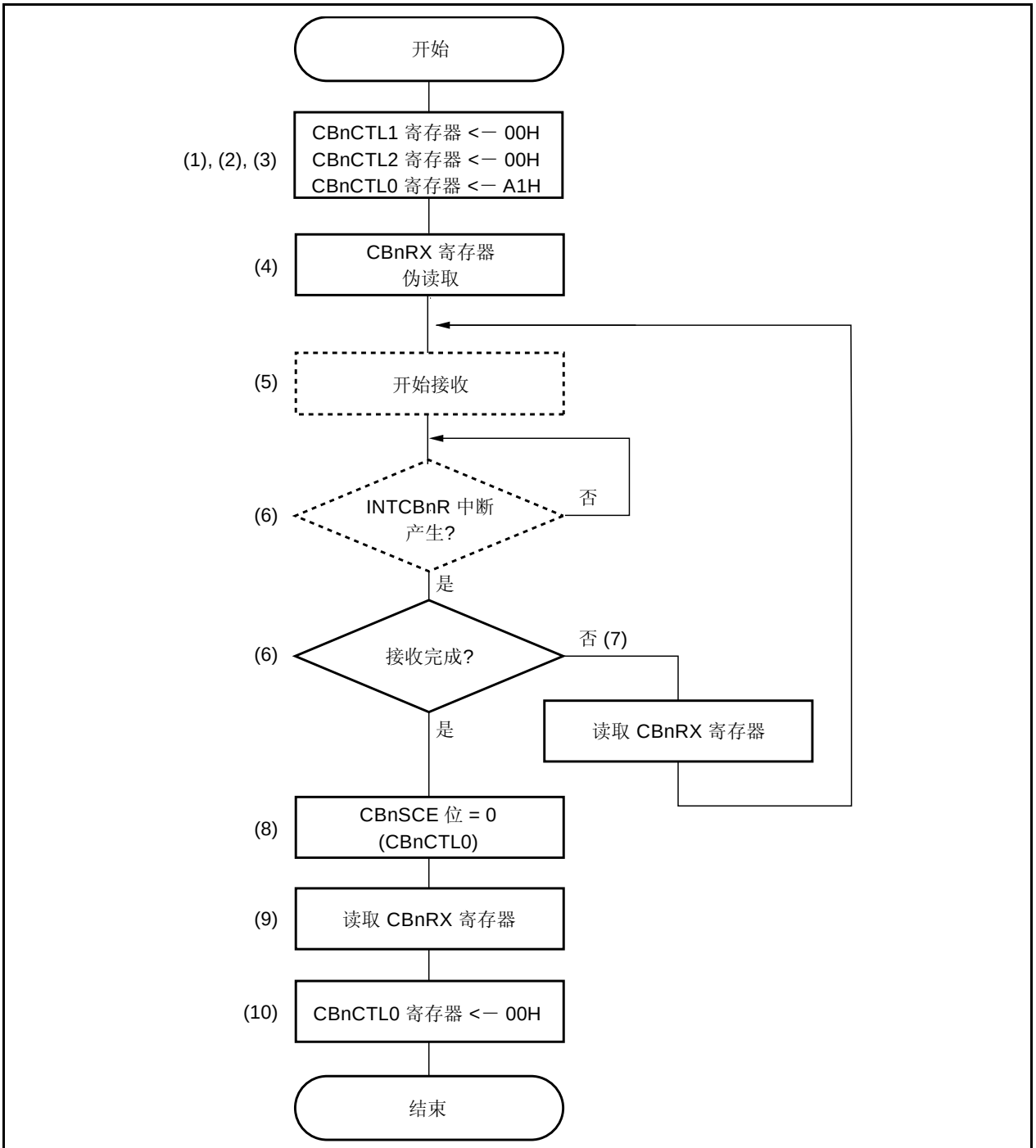
（8）若要结束发送，将 CBnCTL0.CBnPWR 位和 CBnCTL0.CBnTXE 位清除为 0。

备注 n = 0 至 4

18.6.2 单次传输模式（主机模式，接收模式）

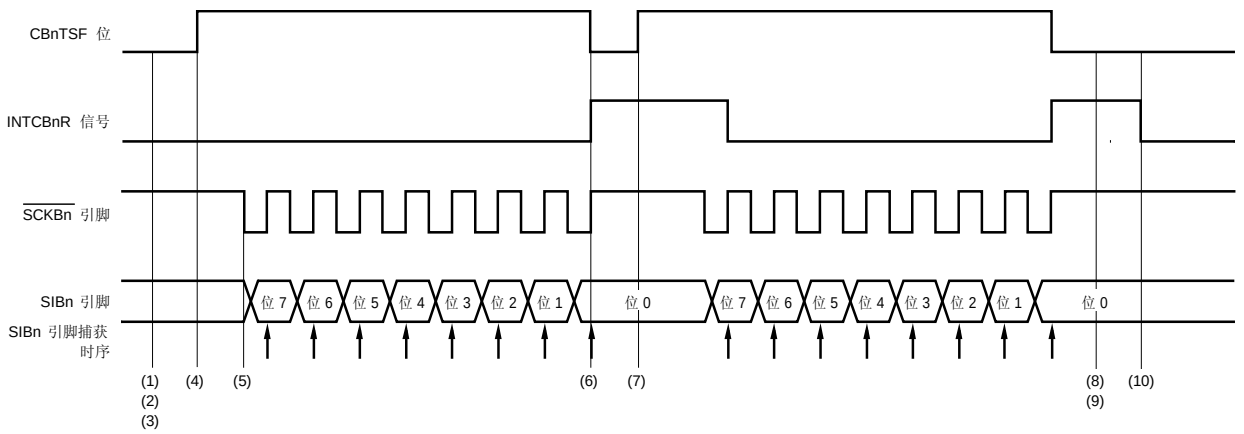
MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= f_{XX} /2（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

图 18-7. 单次传输模式操作（主机模式，接收模式）



备注 1. 虚线部分表示硬件处理。
2. 图中的数字对应于**图 18-8** 中的数字。
3. n = 0 至 4

图 18-8. 单次传输模式操作时序（主机模式，接收模式）



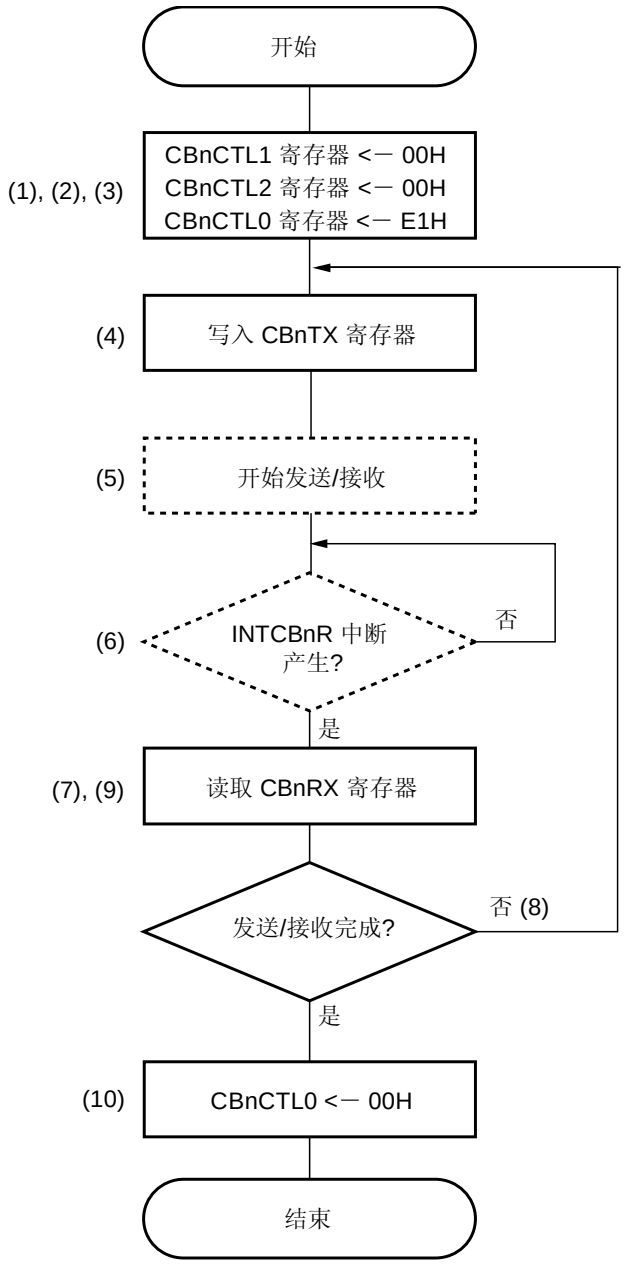
（1）向 CBnCTL1 寄存器写入 00H，选择通信类型 1，通信时钟（f_{CCLK}）= f_{XX}/2，并且设置为主机模式。
（2）向 CBnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。
（3）向 CBnCTL0 寄存器写入 A1H，并在使能通信时钟（f_{CCLK}）的同时，选择为接收模式和 MSB 先行。
（4）通过对 CBnRX 寄存器执行一次伪读取，使得 CBnSTR.CBnTSF 位为 1，并启动接收。
（5）当接收开始后，串行时钟输出至 $\overline{\text{SCKBn}}$ 引脚，并与串行时钟同步捕获 SIBn 引脚的接收数据。
（6）当 CBnCTL2 寄存器设定的传输数据长度接收完成时，停止串行时钟输出和数据捕获，在串行时钟最末边沿产生接收结束中断请求信号（INTCBnR），并将 CBnTSF 位清为 0。
（7）若要连续接收，在 INTCBnR 信号产生之后且 CBnCTL0.CBnSCE 位为 1 时，读取 CBnRX 寄存器。
（8）若要读取 CBnRX 寄存器，而不启动下一次接收，将 CBnSCE 位清除为 0。
（9）读取 CBnRX 寄存器。
（10）若要结束接收，将 CBnCTL0.CBnPWR 位和 CBnCTL0.CBnRXE 位清除为 0。

备注 n = 0 至 4

18.6.3 单次传输模式（主机模式，发送/接收模式）

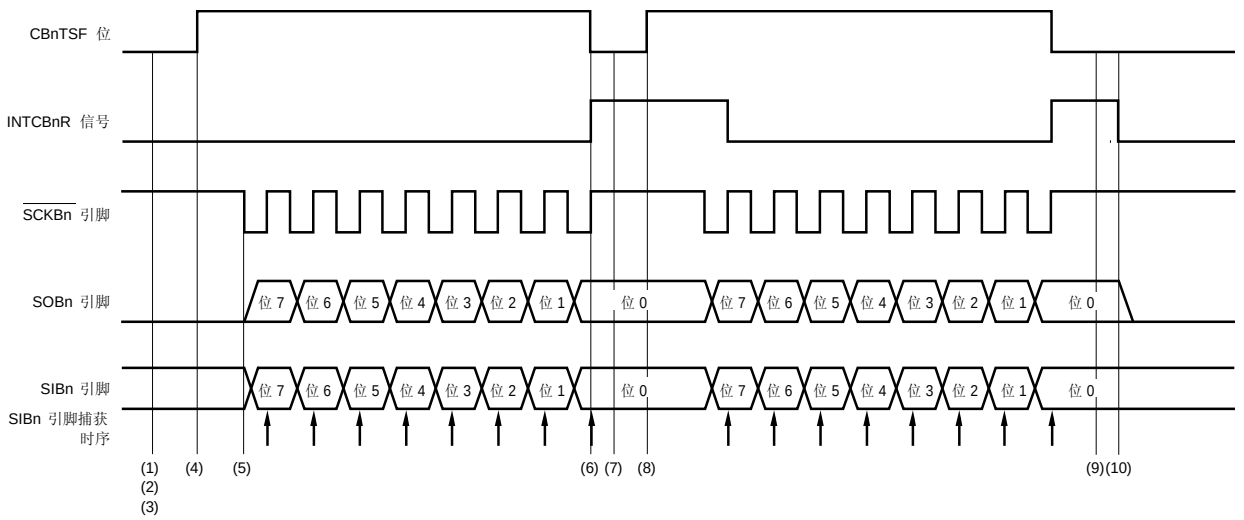
MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= f_{XX} /2（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

图 18-9. 单次传输模式操作（主机模式，发送/接收模式）



备注 **1.** 虚线部分表示硬件处理。
2. 图中的数字对应于**图 18-10** 中的数字。
3. n = 0 至 4

图 18-10. 单次传输模式操作时序（主机模式，发送/接收模式）



（1）向 CBnCTL1 寄存器写入 00H，选择通信类型 1，通信时钟（f_{CCLK}）= f_{XX}/2，并且设置为主机模式。

（2）向 CBnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。

（3）向 CBnCTL0 寄存器写入 E1H，使在使能通信时钟（f_{CCLK}）的同时，选择为发送/接收模式和 MSB 先行。

（4）将发送数据写入 CBnTX 寄存器，使得 CBnSTR.CBnTSF 位置为 1，启动发送/接收。

（5）当发送/接收启动之后，串行时钟输出至 $\overline{\text{SCKBn}}$ 引脚，待发送数据与串行时钟同步，由 SOBn 引脚输出，并且同步捕获 SIBn 引脚的接收数据。

（6）当 CBnCTL2 寄存器设定的传输数据长度发送/接收完成时，停止串行时钟输出、发送数据输出以及数据捕获，在串行时钟最末边沿产生接收结束中断请求信号（INTCBnR），并将 CBnTSF 位清为 0。

（7）读取 CBnRX 寄存器。

（8）若继续发送/接收，从（4）开始重复上述步骤。

（9）读取 CBnRX 寄存器。

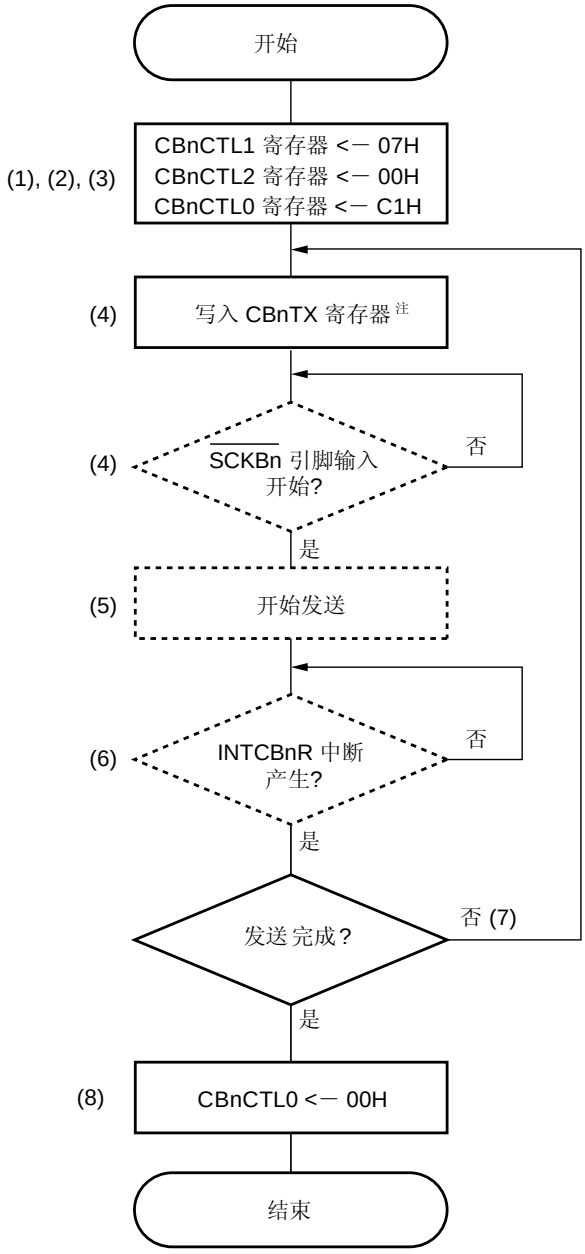
（10）若要结束数据发送/接收，将 CBnCTL0.CBnPWR 位、CBnCTL0.CBnTXE 位和 CBnCTL0.CBnRXE 位清除为 0。

备注 n = 0 至 4

18.6.4 单次传输模式（从机模式，发送模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= 外部时钟（$\overline{SCKBn}$）（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

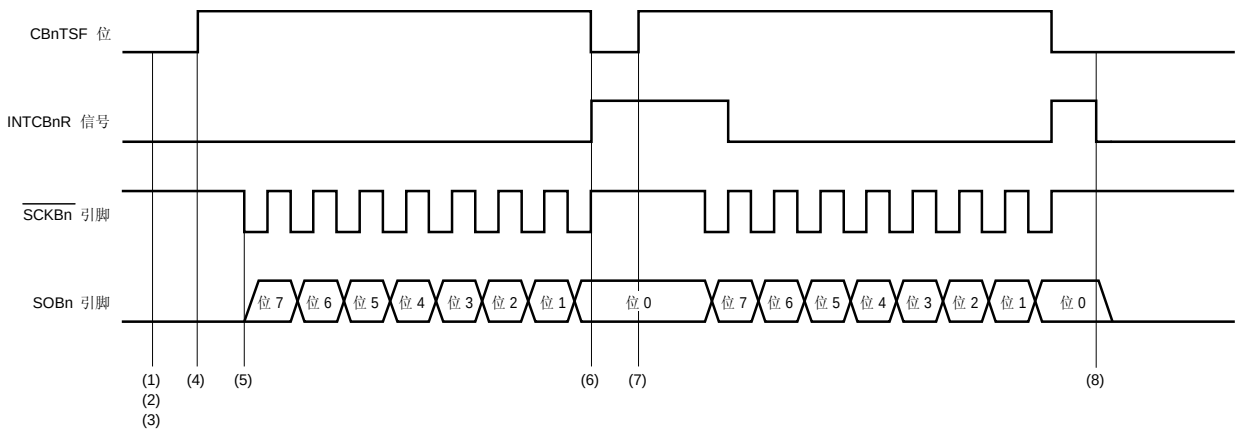
图 18-11. 单次传输模式操作（从机模式，发送模式）



注 在写入 CBnTX 寄存器之前，如果有串行时钟通过 $\overline{SCKBn}$ 引脚输入，数据不能正常发送。这种情况下，对主机和从机端都执行初始化。

备注
1. 虚线部分表示硬件处理。
2. 图中的数字对应于**图 18-12** 中的数字。
3. n = 0 至 4

图 18-12. 单次传输模式操作时序（从机模式，发送模式）



（1）向 CBnCTL1 寄存器写入 07H，选择通信类型 1，通信时钟（f_{CCLK}）= 外部时钟（$\overline{SCKBn}$），并设置为从机模式。

（2）向 CBnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。

（3）向 CBnCTL0 寄存器写入 C1H，并在使能通信时钟（f_{CCLK}）的同时，选择为发送模式和 MSB 先行方式。

（4）将发送数据写入 CBnTX 寄存器，使得 CBnSTR.CBnTSF 位为 1，设备等待串行时钟输入。

（5）当串行时钟输入时，待发送的数据与串行时钟同步，由 SOBn 引脚输出。

（6）当 CBnCTL2 寄存器设定的传输数据长度发送完成时，在串行时钟最末边沿产生接收完成中断请求信号（INTCBnR），停止串行时钟的输入和发送数据的输出，并将 CBnTSF 位清为 0。

（7）若要继续进行发送，在 INTCBnR 信号产生后，从（4）开始重复上述步骤。

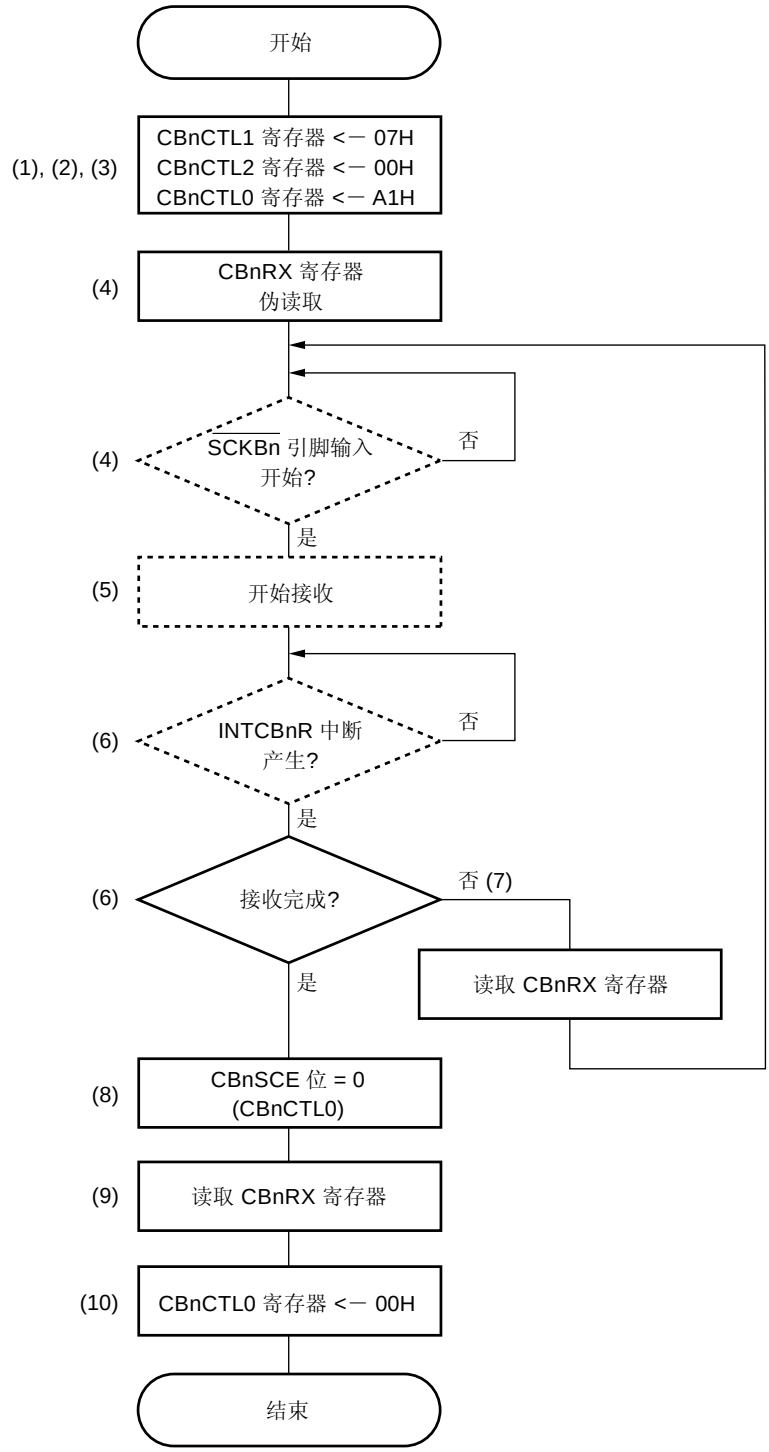
（8）若要结束发送，将 CBnCTL0.CBnPWR 位和 CBnCTL0.CBnTXE 位清除为 0。

备注 n = 0 至 4

18.6.5 单次传输模式（从机模式，接收模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= 外部时钟（$SC\overline{KBn}$）（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

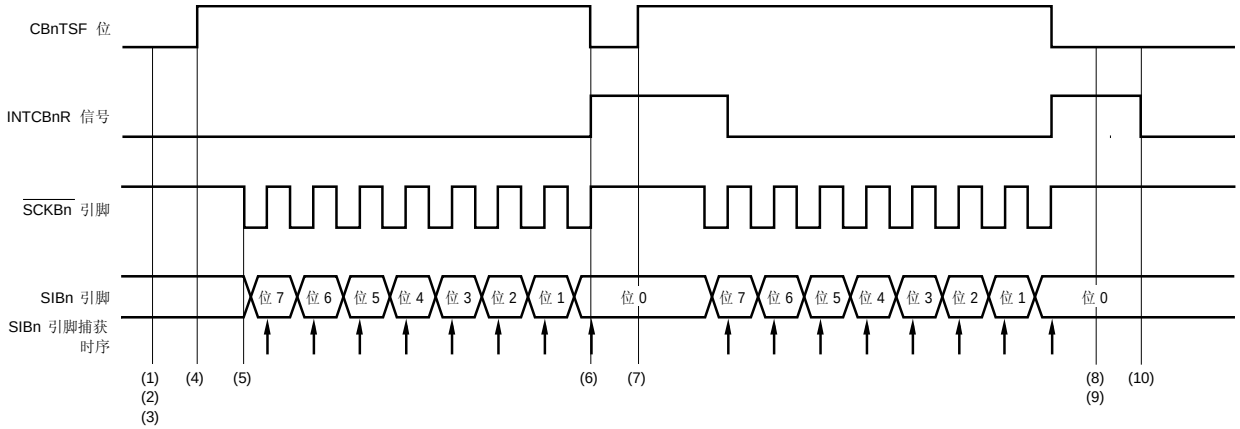
图 18-13. 单次传输模式操作（从机模式，接收模式）



注 在对 CBnRX 寄存器执行伪读取之前，如果如果有串行时钟通过主机的 $\overline{\text{SCKBn}}$ 引脚输入，数据无法正常接收，这种情况下，对主机和从机端都执行初始化。

备注 **1.** 虚线部分表示硬件处理。
2. 图中的数字对应于**图 18-14** 中的数字。
3. n = 0 至 4

图 18-14. 单次传输模式操作时序（从机模式，接收模式）



（1）向 CBnCTL1 寄存器写入 07H，选择通信类型 1，通信时钟（f_{CCLK}）= 外部时钟（$\overline{SCKBn}$），并设置为从机模式。

（2）向 CBnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。

（3）向 CBnCTL0 寄存器写入 A1H，并在使能通信时钟（f_{CCLK}）的同时，选择为接收模式和 MSB 先行方式。

（4）通过对 CBnRX 寄存器进行伪读取使得 CBnSTR.CBnTSF 位置为 1，设备等待串行时钟输入。

（5）当串行时钟输入时，与串行时钟同步捕获 SIBn 引脚输入的接收数据。

（6）当 CBnCTL2 寄存器设定的传输数据长度接收完成时，停止串行时钟输入和数据捕获，在串行时钟最末边沿产生接收结束中断请求信号（INTCBnR），并将 CBnTSF 位清为 0。

（7）若要继续接收，在 INTCBnR 信号产生后，在 CBnCTL0.CBnSCE 位置 1 时读取 CBnRX 寄存器，并等待串行时钟输入。

（8）若要结束接收，将 CBnSCE 位清除为 0。

（9）读取 CBnRX 寄存器。

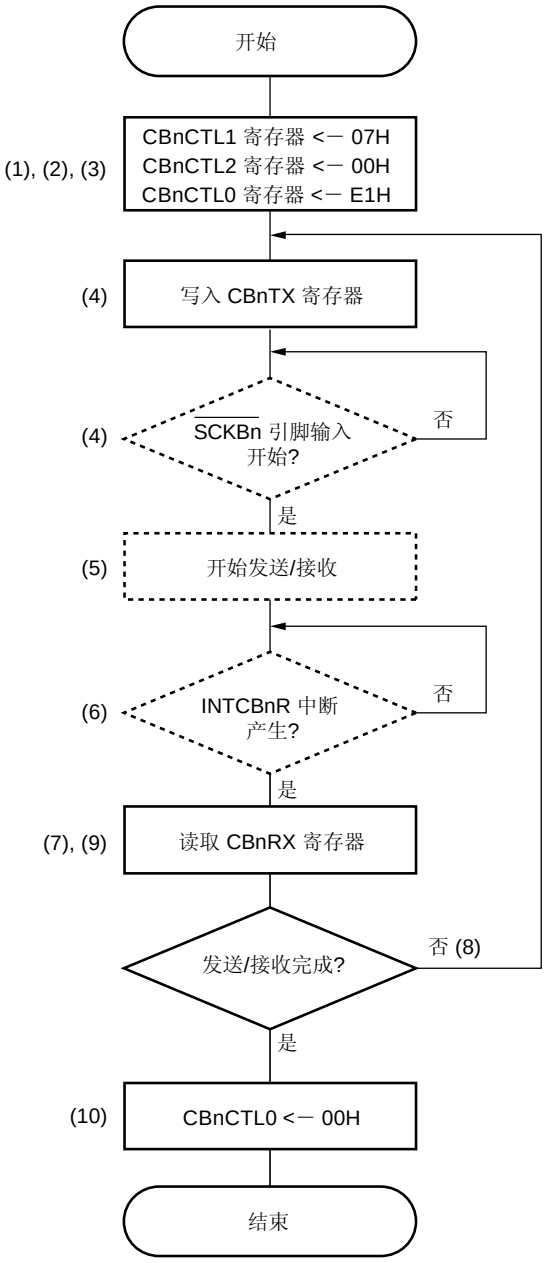
（10）若要结束接收，将 CBnCTL0.CBnPWR 位和 CBnCTL0.CBnRXE 位清除为 0。

备注 n = 0 至 4

18.6.6 单次传输模式（从机模式，发送/接收模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= 外部时钟（$\overline{SCKBn}$）（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

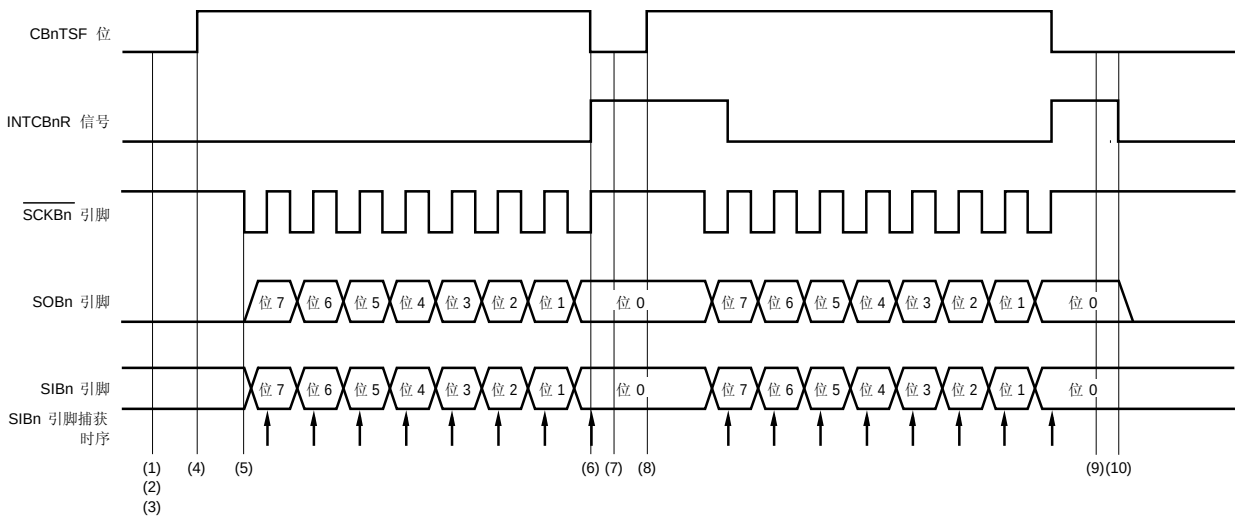
图 18-15. 单次传输模式操作（从机模式，发送/接收模式）



注 在写入 CBnRX 寄存器之前，如果如果有串行时钟通过主机的 $\overline{SCKBn}$ 引脚输入，数据无法正常发送/接收，这种情况下，对主机和从机端都执行初始化。

备注
1. 虚线部分表示硬件处理。
2. 图中的数字对应于**图 18-16** 中的数字。
3. n = 0 至 4

图 18-16. 单次传输模式操作时序（从机模式，发送/接收模式）



（1）向 CBnCTL1 寄存器写入 07H，选择通信类型 1，通信时钟（f_{CCLK}）= 外部时钟（$\overline{SCKBn}$），并设置为从机模式。

（2）向 CBnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。

（3）向 CBnCTL0 寄存器写入 E1H，并在使能通信时钟（f_{CCLK}）的同时，选择为发送/接收模式和 MSB 先行方式。

（4）将待发送数据写入 CBnTX 寄存器，使得 CBnSTR.CBnTSF 位置为 1，设备等待串行时钟的输入。

（5）当串行时钟输入时，待发送数据与串行时钟同步，由 SOBn 引脚输出，并且同步捕获 SIBn 引脚的接收数据。

（6）当 CBnCTL2 寄存器设定的传输数据长度发送/接收完成时，停止串行时钟输入、发送数据输出以及数据捕获，在串行时钟最末边沿产生接收结束中断请求信号（INTCBnR），并将 CBnTSF 位清为 0。

（7）读取 CBnRX 寄存器。

（8）若要继续发送/接收，从（4）开始重复上述步骤。

（9）读取 CBnRX 寄存器。

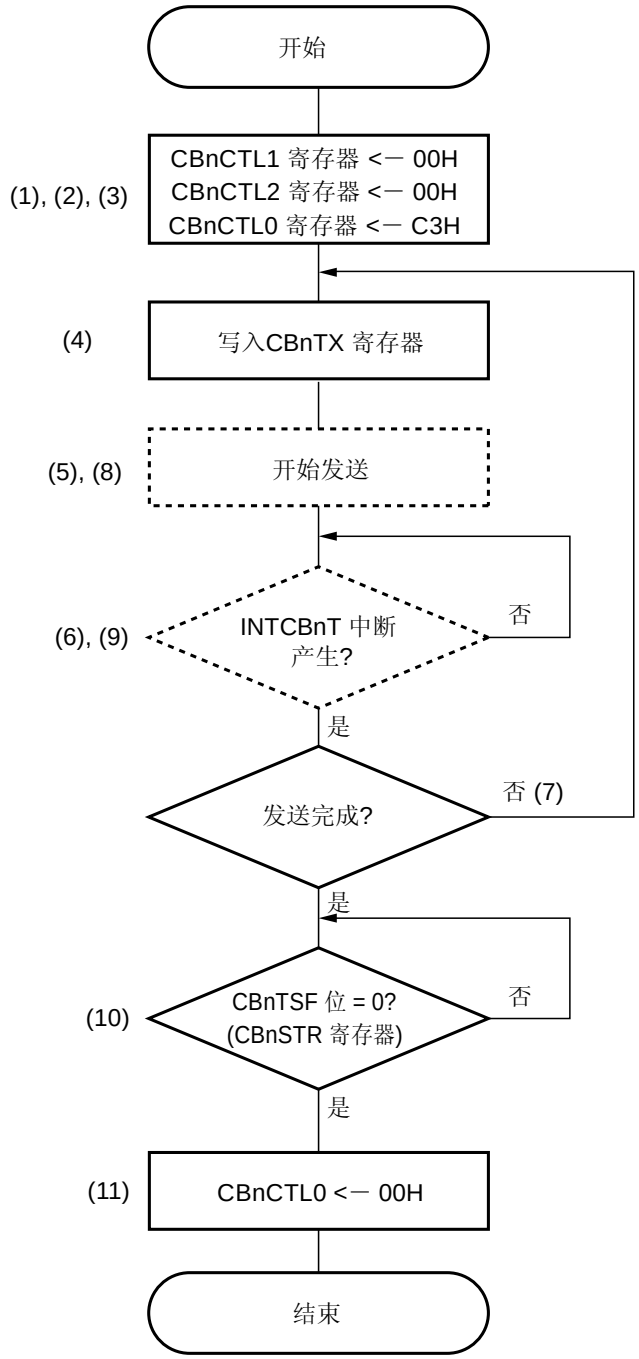
（10）若要结束数据发送/接收，将 CBnCTL0.CBnPWR 位、CBnCTL0.CBnTXE 位和 CBnCTL0.CBnRXE 位清除为 0。

备注 n = 0 至 4

18.6.7 连续传输模式（主机模式，发送模式）

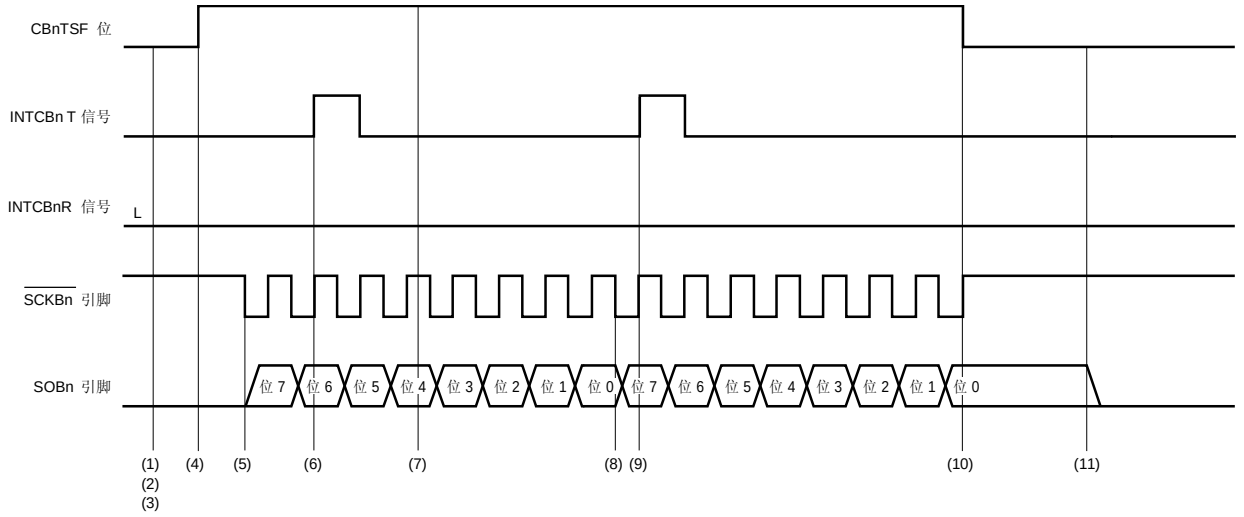
MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= f_{XX} /2（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

图 18-17. 连续传输模式操作（主机模式，发送模式）



备注 **1.** 虚线部分表示硬件处理。
2. 图中的数字对应于**图 18-18** 中的数字。
3. n = 0 至 4

图 18-18. 连续传输模式操作时序（主机模式，发送模式）



（1）向 CBnCTL1 寄存器写入 00H，选择通信类型 1，通信时钟（f_{CCLK}）= f_{XX}/2，并设置为主机模式。
（2）向 CBnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。
（3）向 CBnCTL0 寄存器写入 C3H，在使能通信时钟（f_{CCLK}）操作的同时，选择为发送模式、MSB 先行以及连续传输模式。
（4）将待发送数据写入 CBnTX 寄存器使得 CBnSTR.CBnTSF 位置为 1，启动发送。
（5）当发送开始后，输出串行时钟至 $\overline{\text{SCKBn}}$ 引脚，待发送数据与串行时钟同步，由 SOBn 引脚输出。
（6）当发送数据从 CBnTX 寄存器传输至移位寄存器完成，且允许写入 CBnTX 寄存器时，产生发送使能中断请求信号（INTCBnT）。
（7）若要连续发送，在产生 INTCBnT 信号之后，从（4）开始重复上述步骤。
（8）当通信完成之前，新的发送数据已经写入 CBnTX 寄存器，则在通信完成后开始下一次通信。
（9）从 CBnTX 寄存器传输发送数据至移位寄存器完成，且产生 INTCBnT 信号。若要结束当前的连续发送传输，不要写入 CBnTX 寄存器。
（10）如果传输完成之前下一个发送数据未被写入 CBnTX 寄存器，传输完成后，则等待 CBnTSF 位被清除为 0。
（11）要禁止发送，在确认 CBnTSF 位被置为 0 之后，将 CBnCTL0.CBnPWR 位和 CBnCTL0.CBnTXE 位清除为 0。

注意事项 连续发送模式下，不产生接收完成中断请求信号（INTCBnR）。

备注 n = 0 至 4

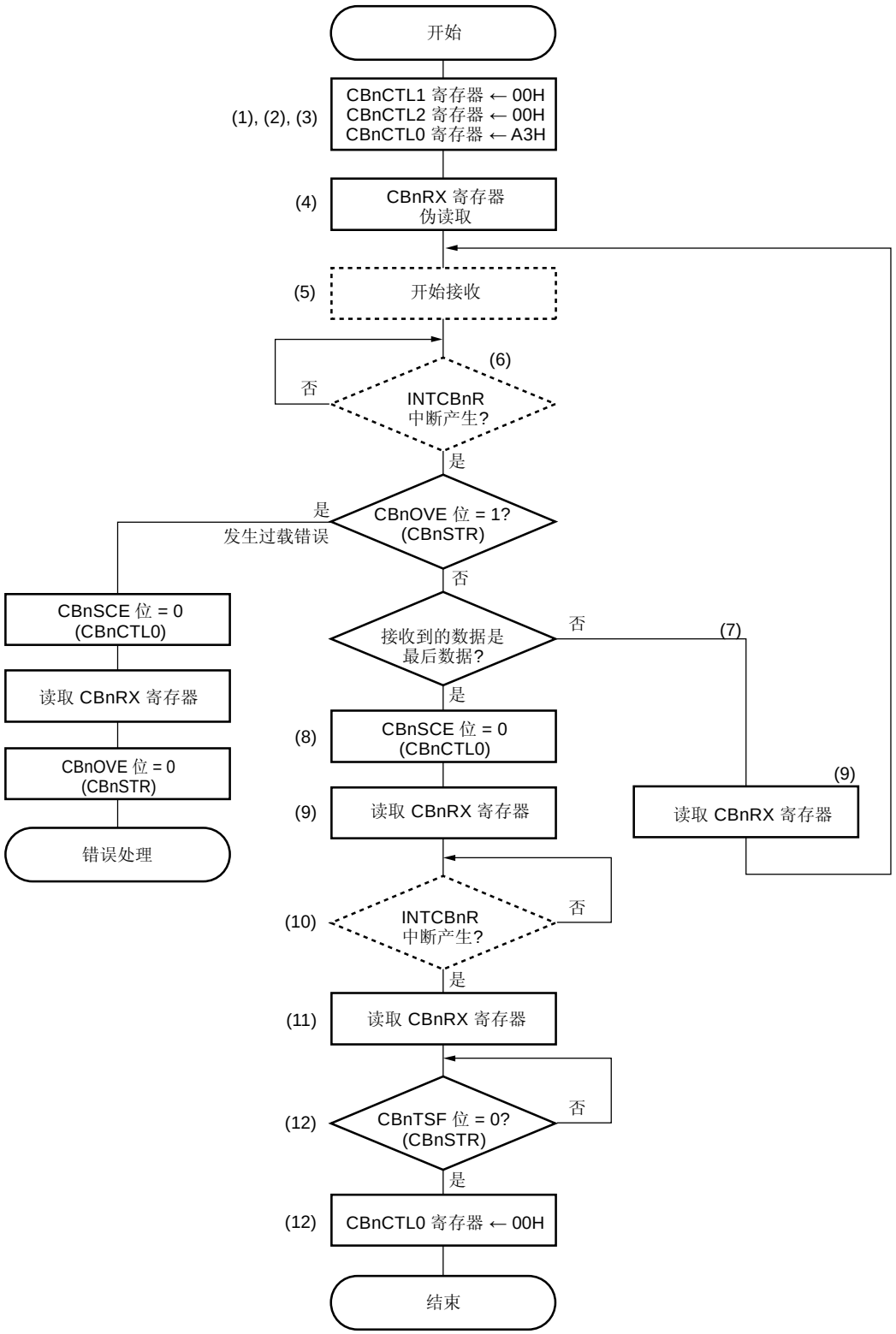
18.6.8 连续传输模式（主机模式，接收模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= f_{XX} /2（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

图 18-19 中的流程图展示了主机模式下接收到指定数量数据项的操作。重复操作，直到所有的指定数据项都已接收到。但是，如果发生过载错误，传输结束。执行必需的错误处理。过载错误的详情参见 **18.6.13 接收错误**。

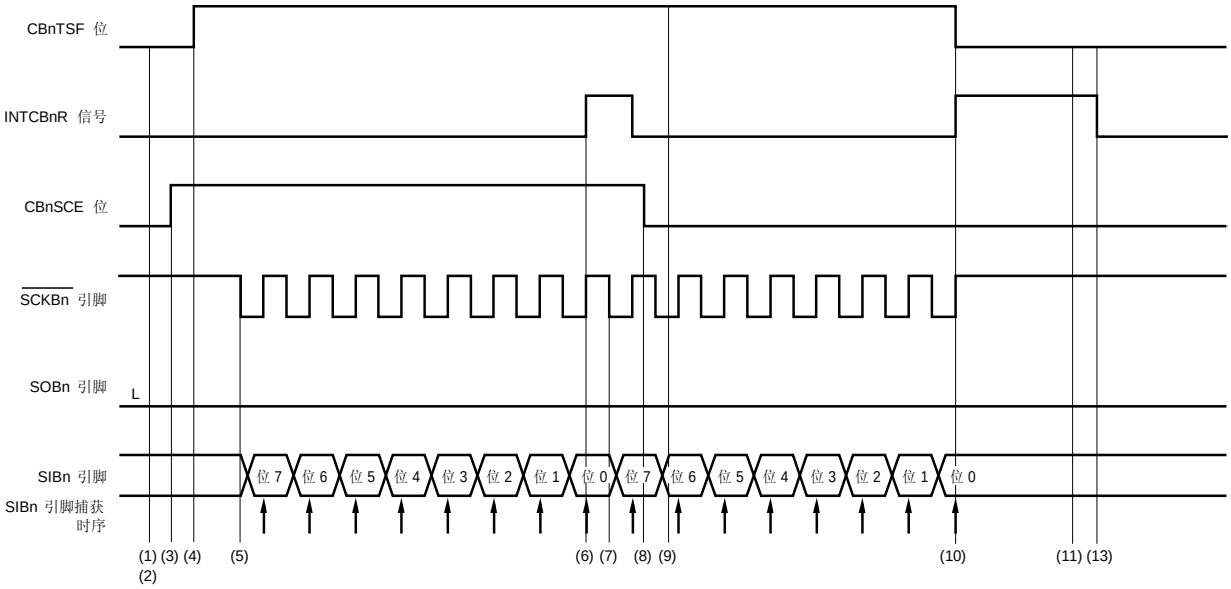
图 18-20 中的操作时序展示了无错误发生时的情况。

图 18-19. 连续传输模式操作（主机模式，接收模式）



备注 **1.** 虚线部分表示硬件处理。
2. 图中的数字对应于**图 18-20** 中的数字。
3. n = 0 至 4

图 18-20. 连续传输模式操作时序（主机模式，接收模式）



（1）向 CBnCTL1 寄存器写入 00H，选择通信类型 1，通信时钟（f_{CCLK}）= f_{XX}/2，并设置为主机模式。
（2）向 CBnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。
（3）向 CBnCTL0 寄存器写入 A3H，在使能通信时钟（f_{CCLK}）操作的同时，选择为接收模式、MSB 先行以及连续传输模式。
（4）对 CBnRX 寄存器执行一次伪读取，使得 CBnSTR.CBnTSF 位置为 1，并启动接收。
（5）当接收开始后，输出串行时钟至 $\overline{\text{SCKBn}}$ 引脚，并与串行时钟同步捕获 SIBn 引脚的接收数据。
（6）当接收完成之后，产生接收完成中断请求信号（INTCBnR），允许从 CBnRX 寄存器读取接收数据。
（7）当通信结束时因为 CBnCTL0.CBnSCE 位为 1，则在本次通信完成后立即开始下一次通信。
（8）若要结束当前连续接收传输，将 CBnSCE 位清除为 0。
（9）读取 CBnRX 寄存器。
（10）当接收完成时，产生 INTCBnR 信号，并且允许从 CBnRX 寄存器读取接收数据。如果在通信结束之前 CBnSCE 位被设置为 0，停止输出串行时钟至 $\overline{\text{SCKBn}}$ 引脚，并将 CBnTSF 清除为 0，从而结束接收操作。
（11）读取 CBnRX 寄存器。
（12）要禁止接收，在确认 CBnTSF 位为 0 之后，将 CBnCTL0.CBnPWR 位和 CBnCTL0.CBnRXE 位清除为 0。

备注 n = 0 至 4

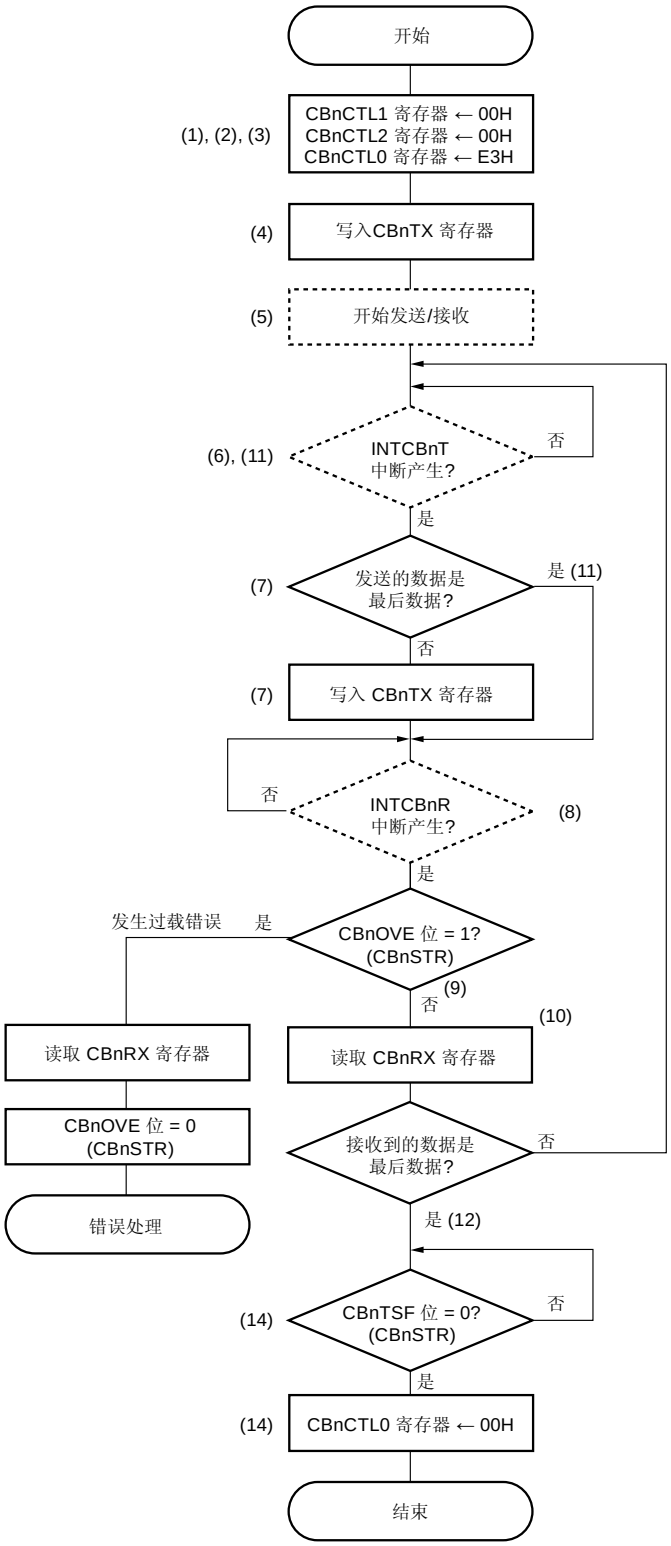
18.6.9 连续传输模式（主机模式，发送/接收模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= f_{XX}/2（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 000），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

图 18-21 中的流程图展示了主机模式下发送/接收到指定数量数据项的发送/接收操作。重复操作，直到所有的指定数据项都已发送/接收。但是，如果发生过载错误，传输结束。执行必需的错误处理。过载错误的详情参见 **18.6.13 接收错误**。

图 18-22 中的操作时序展示了无错误发生时的情况。

图 18-21. 连续传输模式操作（主机模式，发送/接收模式）

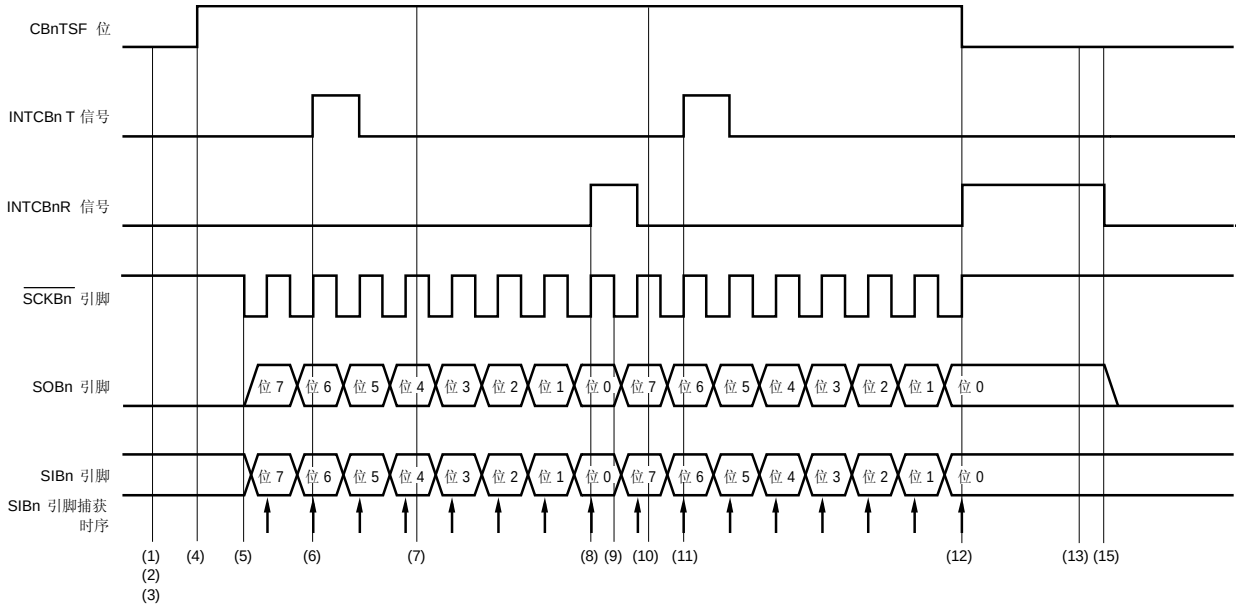


备注 **1.** 虚线部分表示硬件处理。

2. 图中的数字对应于**图 18-22** 中的数字。

3. n = 0 至 4

图 18-22. 连续传输模式操作时序（主机模式，发送/接收模式）（1/2）



（1）向 CBnCTL1 寄存器写入 00H，选择通信类型 1，通信时钟（f_{CCLK}）= f_{XX}/2，并设置为主机模式。

（2）向 CBnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。

（3）向 CBnCTL0 寄存器写入 E3H，并且在使能通信时钟（f_{CCLK}）的同时，选择为发送/接收模式、MSB 先行以及连续传输模式。

（4）将待发送数据写入 CBnTX 寄存器，使得 CBnSTR.CBnTSF 位置为 1，启动发送/接收。

（5）当发送/接收启动之后，输出串行时钟至 $\overline{\text{SCKBn}}$ 引脚，待发送数据与串行时钟同步，由 SOBn 引脚输出，并且同步捕获 SIBn 引脚的接收数据。

（6）当发送数据从 CBnTX 寄存器传输至移位寄存器完成，且允许写入 CBnTX 寄存器时，产生发送使能中断请求信号（INTCBnT）。

（7）若要连续发送/接收，在产生 INTCBnT 信号之后，再次将待发送数据写入 CBnTX 寄存器。

（8）当一次发送/接收完成之后，产生接收完成中断请求信号（INTCBnR）），并且允许读取 CBnRX 寄存器。

（9）当通信完成之前，新的待发送数据已经写入 CBnTX 寄存器，则在本次通信完成后立即开始下一次通信。

（10）读取 CBnRX 寄存器。

备注 n = 0 至 4

图 18-22. 连续传输模式操作时序（主机模式，发送/接收模式）（2/2）

（11）从 CBnTX 寄存器传输发送数据至移位寄存器完成，且产生 INTCBnT 信号。若要结束当前的发送/接收传输，则不写入 CBnTX 寄存器。

（12）如果传输完成之前下一个待发送数据未被写入 CBnTX 寄存器，传输完成后，停止从 $\overline{\text{SCKBn}}$ 引脚输出的串行时钟，并等待 CBnTSF 位被清除为 0。

（13）发生接收错误中断请求信号（INTCBnR）时，读取 CBnRX 寄存器。

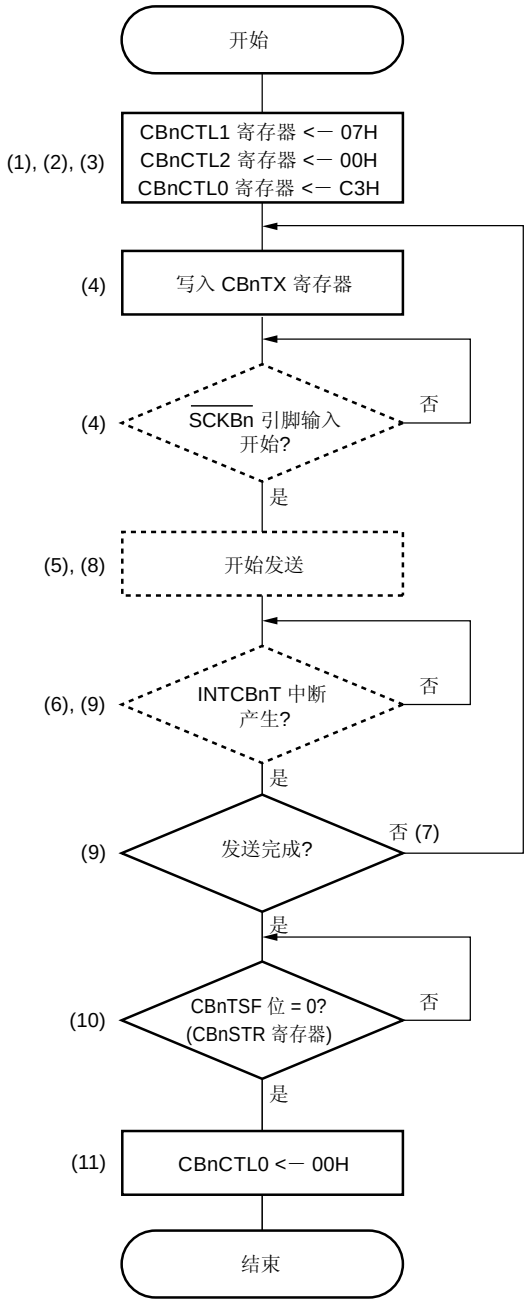
（14）要禁止发送/接收，在确认 CBnTSF 位被设置为 0 之后，将 CBnCTL0.CBnPWR 位、CBnCTL0.CBnTXE 位和 CBnCTL0.CBnRXE 位清除为 0。

备注 n = 0 至 4

18.6.10 连续传输模式（从机模式，发送模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= 外部时钟（$\overline{\text{SCKBn}}$）（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

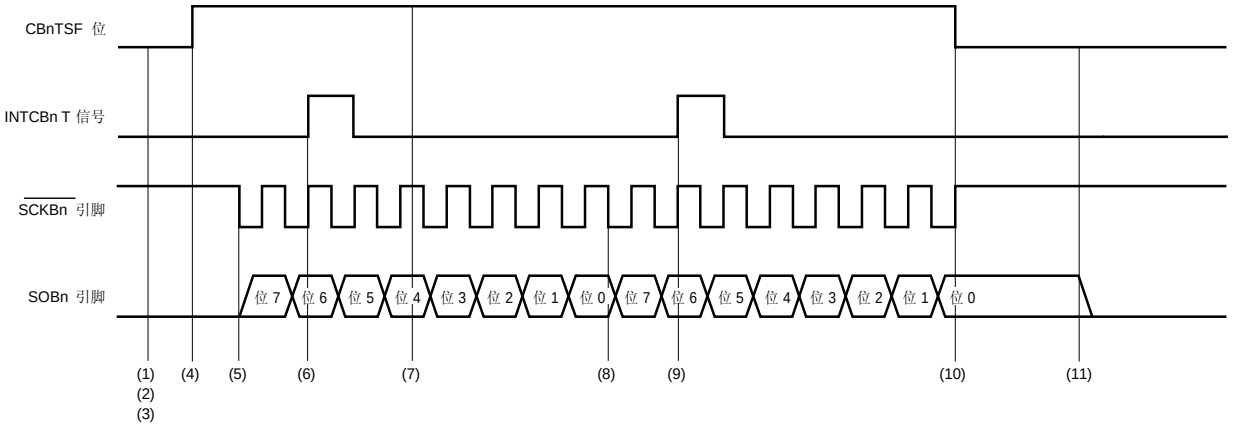
图 18-23. 连续传输模式操作（从机模式，发送模式）



注 在写入 CBnTX 寄存器之前，如果如果有串行时钟通过主机的 $\overline{\text{SCKBn}}$ 引脚输入，数据无法正常发送，这种情况下，对主机和从机端都执行初始化。

备注 **1.** 虚线部分表示硬件处理。
2. 图中的数字对应于**图 18-24** 中的数字。
3. n = 0 至 4

图 18-24. 连续传输模式操作时序（从机模式，发送模式）



（1）向 CBnCTL1 寄存器写入 07H，选择通信类型 1，通信时钟（f_{CCLK}）= 外部时钟（SCK$\overline{\text{Bn}}$），并设置为从机模式。

（2）向 CBnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。

（3）向 CBnCTL0 寄存器写入 C3H，在使能通信时钟（f_{CCLK}）操作的同时，选择为发送模式、MSB 先行以及连续传输模式。

（4）将待发送数据写入 CBnTX 寄存器，使得 CBnSTR.CBnTSF 位为 1，设备等待串行时钟输入。

（5）当串行时钟输入时，待发送数据与串行时钟同步，由 SOBn 引脚输出。

（6）当发送数据从 CBnTX 寄存器至移位寄存器的传输完成，且允许写入 CBnTX 寄存器时，产生发送使能中断请求信号（INTCBnT）。

（7）若要连续发送，在产生 INTCBnT 信号之后，从（4）开始重复上述步骤。

（8）当 CBnCTL2 寄存器设定的传输数据长度发送完成后，有串行时钟输入，则启动连续发送。

（9）当发送数据从 CBnTX 寄存器至移位寄存器的传输完成，并且允许写入 CBnTX 寄存器，产生 INTCBnT 信号。若要结束当前的连续发送传输，不要写入 CBnTX 寄存器。

（10）当 CBnCTL2 寄存器设定的传输数据长度对应的时钟周期数量输入时，而未写入 CBnTX 寄存器，清除 CBnTSF 位为 0，从而结束发送。

（11）要禁止发送，在确认 CBnTSF 位为 0 之后，将 CBnCTL0.CBnPWR 位和 CBnCTL0.CBnTXE 位清除为 0。

注意事项 连续发送模式下，不产生接收完成中断请求信号（INTCBnR）。

备注 n = 0 至 4

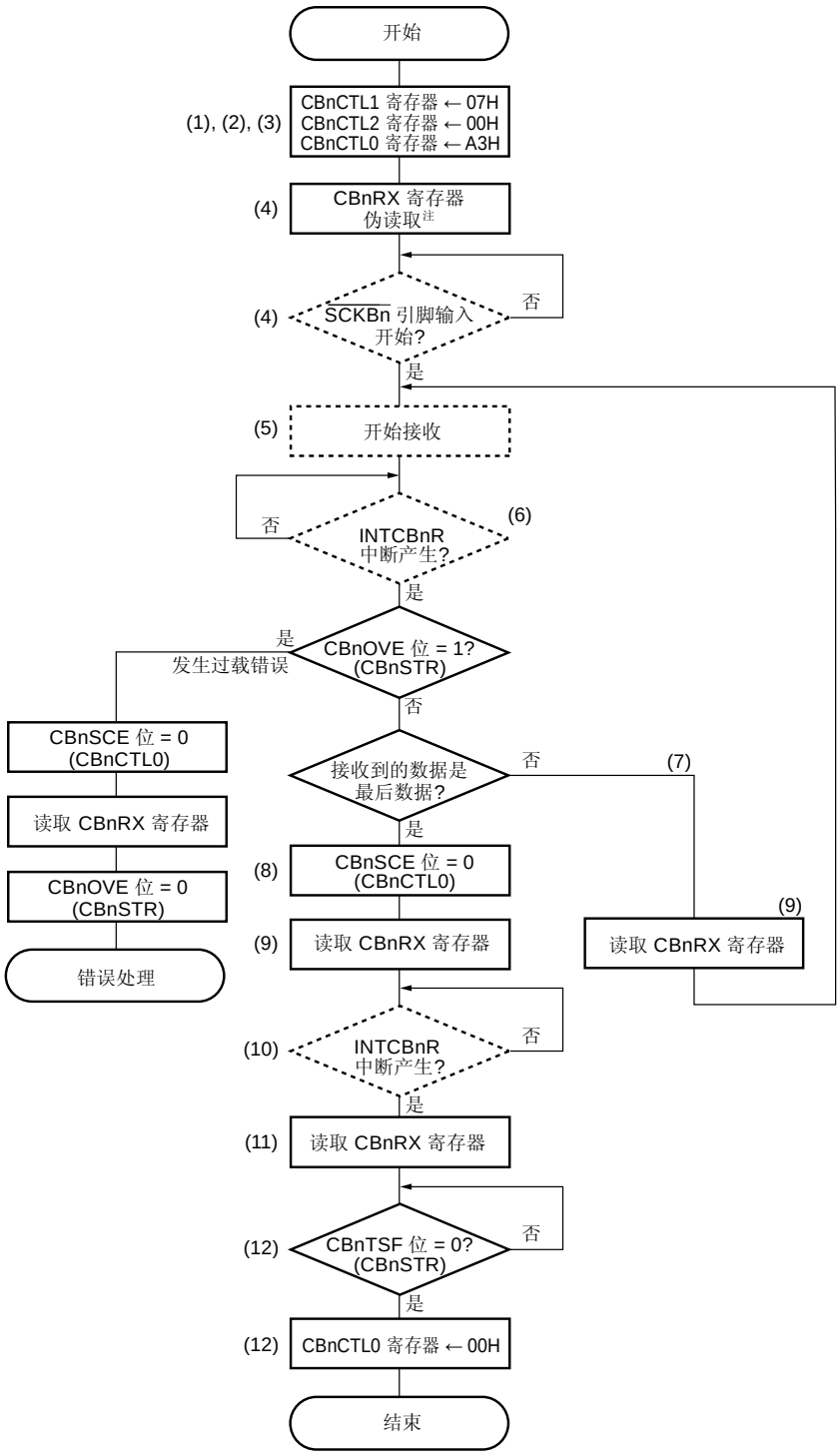
18.6.11 连续传输模式（从机模式，接收模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= 外部时钟（$\overline{SCKBn}$）（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

图 18-25 中的流程图展示了从机模式下接收到指定数量数据项的接收操作。重复操作，直到所有的指定数据项都已接收。但是，如果发生过载错误，传输结束。执行必需的错误处理。过载错误的详情参见 **18.6.13 接收错误**。

图 18-26 中的操作时序展示了无错误发生时的情况。

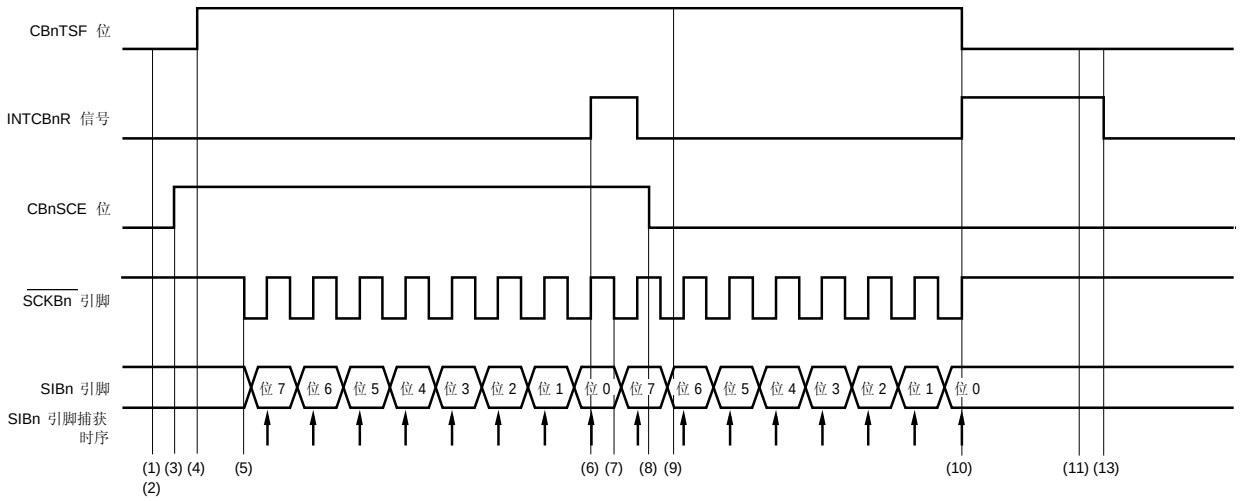
图 18-25. 连续传输模式操作（从机模式，接收模式）



注 在对 CBnRX 寄存器执行伪读取之前，如果如果有串行时钟通过主机的 $\overline{\text{SCKBn}}$ 引脚输入，数据无法正常接收，这种情况下，对主机和从机端都执行初始化。

备注
1. 虚线部分表示硬件处理。
2. 图中的数字和**图 18-26** 中的处理数字相对应。
3. n = 0 至 4

图 18-26. 连续传输模式操作时序（从机模式，接收模式）



（1）向 CBnCTL1 寄存器写入 07H，选择通信类型 1，通信时钟（f_{CCLK}）= 外部时钟（$S\overline{CKBn}$），并设置为从机模式。

（2）向 CBnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。

（3）向 CBnCTL0 寄存器写入 A3H，在使能通信时钟（f_{CCLK}）操作的同时，选择为接收模式、MSB 先行以及连续传输模式。

（4）通过对 CBnRX 寄存器进行伪读取，使得 CBnSTR.CBnTSF 位为 1，设备等待串行时钟输入。

（5）当串行时钟输入时，与串行时钟同步从 SIBn 引脚捕获接收数据。

（6）当接收完成时，产生接收完成中断请求信号（INTCBnR），且允许从 CBnRX 寄存器读取接收数据。

（7）在 CBnCTL0.CBnSCE 位被置 1 状态下输入串行时钟，会启动连续接收。

（8）若要结束当前连续接收传输，将 CBnSCE 位清除为 0。

（9）读取 CBnRX 寄存器。

（10）当接收完成时，产生 INTCBnR 信号，并且允许读取 CBnRX 寄存器。如果在通信完成之前设置 CBnSCE 位为 0，则清除 CBnTSF 位为 0，从而结束操作。

（11）读取 CBnRX 寄存器。

（12）要禁止接收，在确认 CBnTSF 位为 0 之后，将 CBnCTL0.CBnPWR 位和 CBnCTL0.CBnRXE 位清除为 0 。

备注 n = 0 至 4

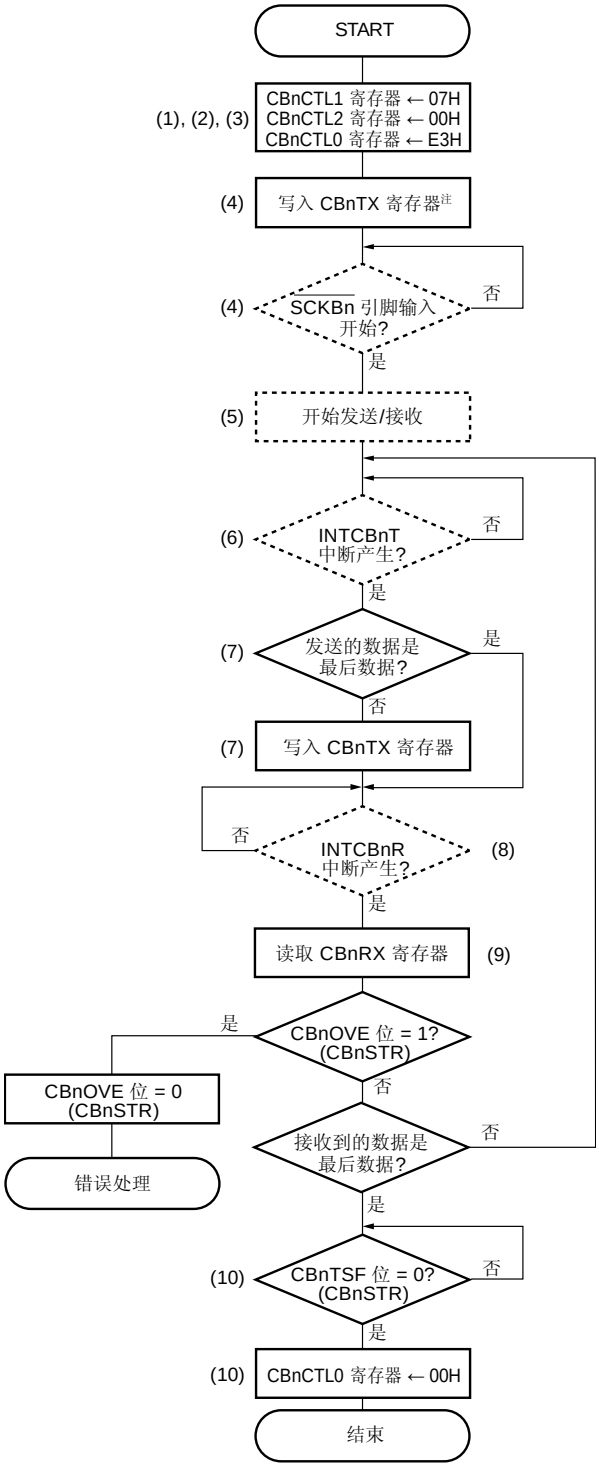
18.6.12 连续传输模式（从机模式，发送/接收模式）

MSB 先行（CBnCTL0.CBnDIR 位= 0），通信类型 1（CBnCTL1.CBnCKP 和 CBnCTL1.CBnDAP 位= 00），通信时钟（f_{CCLK}）= 外部时钟（$SC\overline{K}Bn$）（CBnCTL1.CBnCKS2 至 CBnCTL1.CBnCKS0 位 = 111），传输数据长度= 8 位（CBnCTL2.CBnCL3 至 CBnCTL2.CBnCL0 位= 0000）。

图 18-27 中的流程图展示了从机模式下发送/接收到指定数量数据项的发送/接收操作。重复操作，直到所有的指定数据项都已发送/接收。但是，如果发生过载错误，传输结束。执行必需的错误处理。过载错误的详情参见 **18.6.13 接收错误**。

图 18-28 中的操作时序展示了无错误发生时的情况。

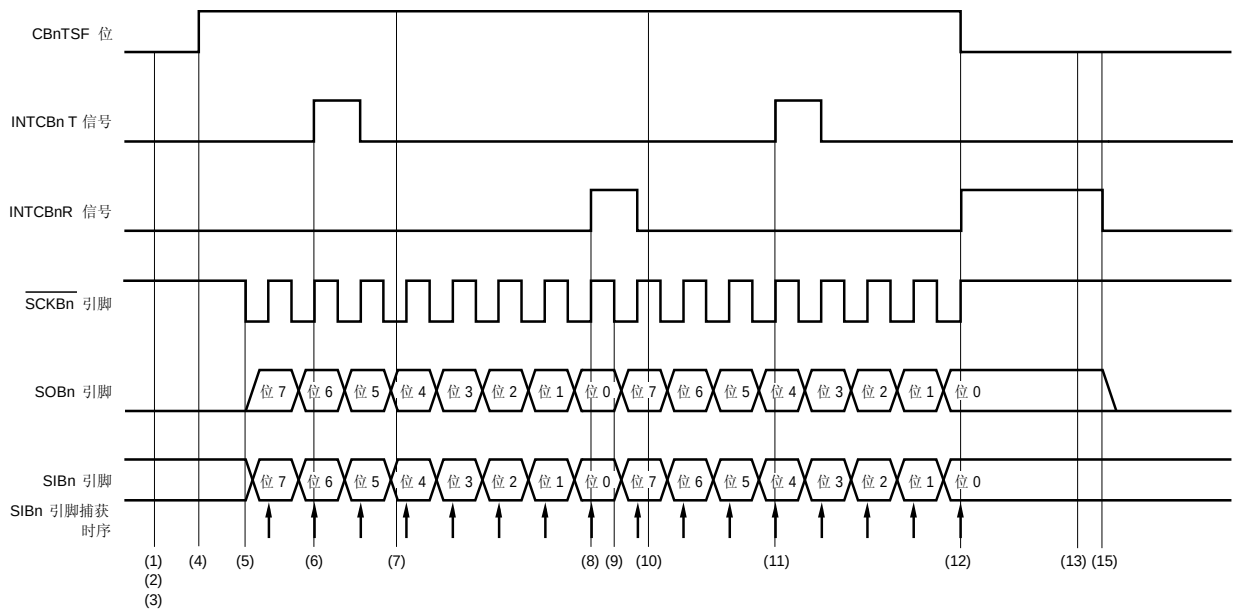
图 18-27. 连续传输模式操作（从机模式，发送/接收模式）



注 在写入 CBnRX 寄存器之前，如果如果有串行时钟通过主机的 $\overline{\text{SCKBn}}$ 引脚输入，数据无法正常发送/接收，这种情况下，对主机和从机端都执行初始化。

备注 **1.** 虚线部分表示硬件处理。
2. 图中的数字对应于**图 18-28** 中的数字。
3. n = 0 至 4

图 18-28. 连续传输模式操作时序（从机模式，发送/接收模式）



（1）向 CBnCTL1 寄存器写入 07H，选择通信类型 1，通信时钟（f_{CCLK}）= 外部时钟（$S\overline{CKBn}$），并设置为从机模式。

（2）向 CBnCTL2 寄存器写入 00H，设定传输数据长度为 8 位。

（3）向 CBnCTL0 寄存器写入 E3H，在使能通信时钟（f_{CCLK}）的同时，选择为发送/接收模式、MSB 先行以及连续传输模式。

（4）将待发送数据写入 CBnTX 寄存器，使得 CBnSTR.CBnTSF 位置为 1，设备等待串行时钟输入。

（5）当串行时钟输入时，与串行时钟同步，输出发送数据至 SOBn 引脚，并且捕获 SIBn 引脚的接收数据。

（6）当发送数据从 CBnTX 寄存器至移位寄存器的传输完成，且允许写入 CBnTX 寄存器时，产生发送使能中断请求信号（INTCBnT）。

（7）若要继续发送，在产生 INTCBnT 信号之后，再次将发送数据写入 CBnTX 寄存器。当发送数据从 CBnTX 寄存器至移位寄存器的传输完成，并且允许写入 CBnTX 寄存器，产生 INTCBnT 信号。若要结束当前的发送/接收传输，则不写入 CBnTX 寄存器。

（8）当 CBnCTL2 寄存器设定的传输数据长度接收完成后，产生接收完成中断请求信号（INTCBnR），并且允许读取 CBnRX 寄存器。如果在步骤（7）中将下一个发送数据写入 CBnTX 寄存器并立即输入串行时钟，会启动新的连续发送/接收。如果下一个数据未写入 CBnTX 寄存器，将 CBnTSF 位清除为 0，从而结束发送/接收。

（9）读取 CBnRX 寄存器。

（10）要禁止发送，在确认 CBnTSF 位为 0 之后，将 CBnCTL0.CBnPWR 位、CBnCTL0.CBnTXE 位和 CBnCTL0.CBnRXE 位清除为 0。

备注 n = 0 至 4

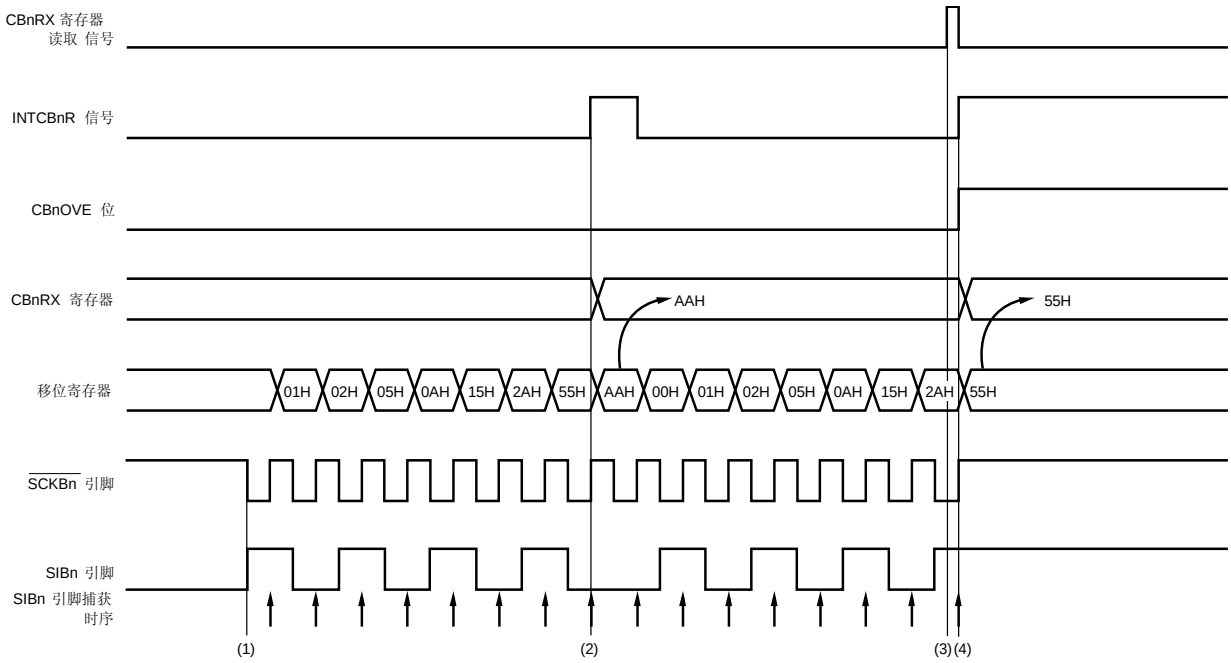
18.6.13 接收错误

连续传输模式下，在允许接收（CBnCTL0.CBnRXE 位= 1）时执行传输，INTCBnR 信号产生后，如果在读取 CBnRX 寄存器之前下一次接收操作就已经完成时，则再次产生接收完成中断请求信号（INTCBnR），并且过载错误标志（CBnSTR.CBnOVE）被置为 1。

即使发生了过载错误，由于 CBnRX 已经被更新，先前接收到的数据丢失。即使发生了接收错误，如果未读取 CBnRX 寄存器，在下次接收完成时，会再次产生 INTCBnR 信号。

在 INTCBnR 信号产生之后，如果在采样下次接收最末位数据之前的半个时钟，对 CBnRX 寄存器的读取尚未完成，则发生过载错误。

图 18-29. 过载错误时序

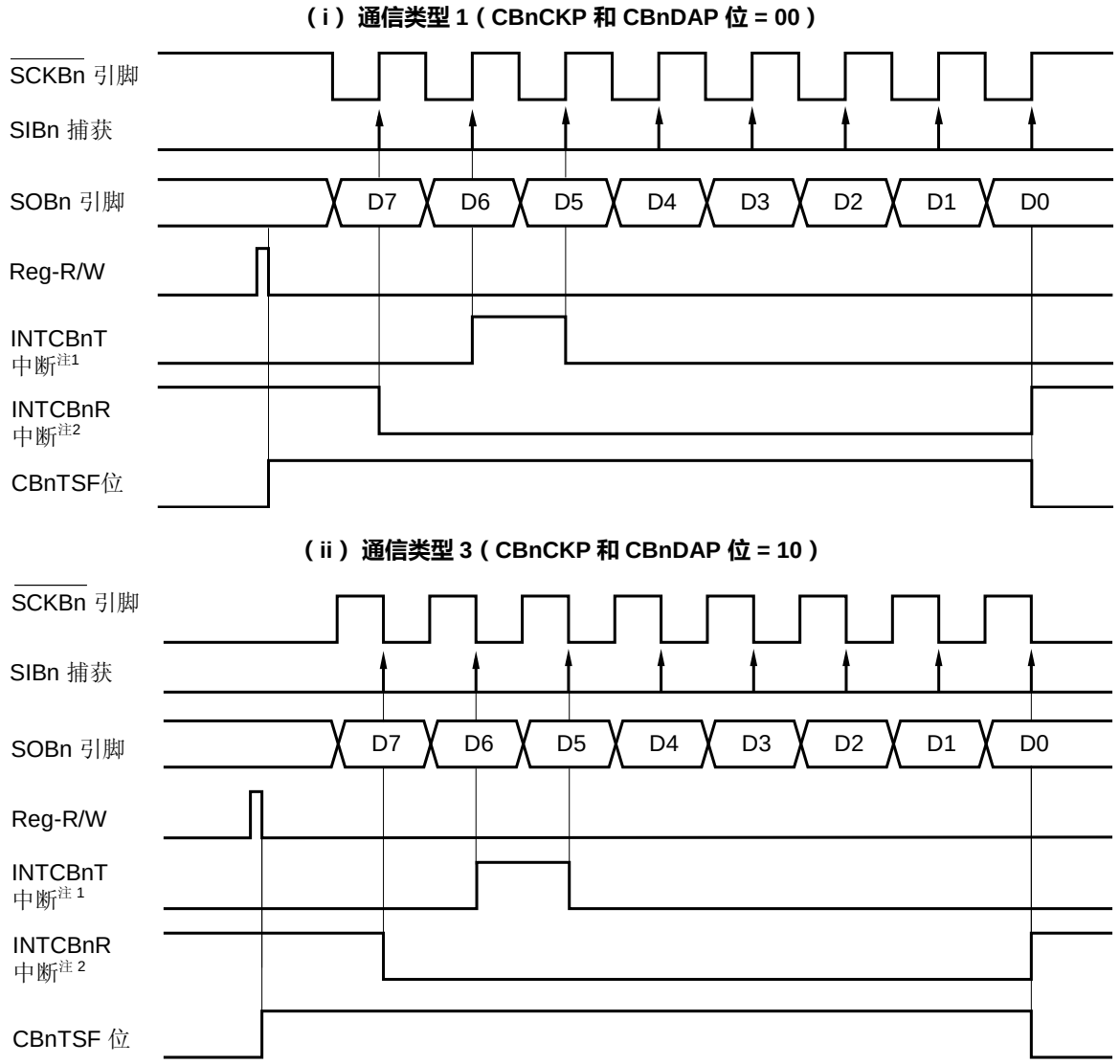


（1）启动连续传输。
（2）完成首次传输。
（3）直到第二次传输完成前的半个时钟周期，都没有读取 CBnRX 寄存器。
（4）发生过载错误，产生接收完成中断请求信号（INTCBnR），而后过载错误标志（CBnSTR.CBnOVE）置为 1。接收数据被覆盖。

备注 n = 0 至 4

18.6.14 时钟时序

图 18-30. 时钟时序（1/2）



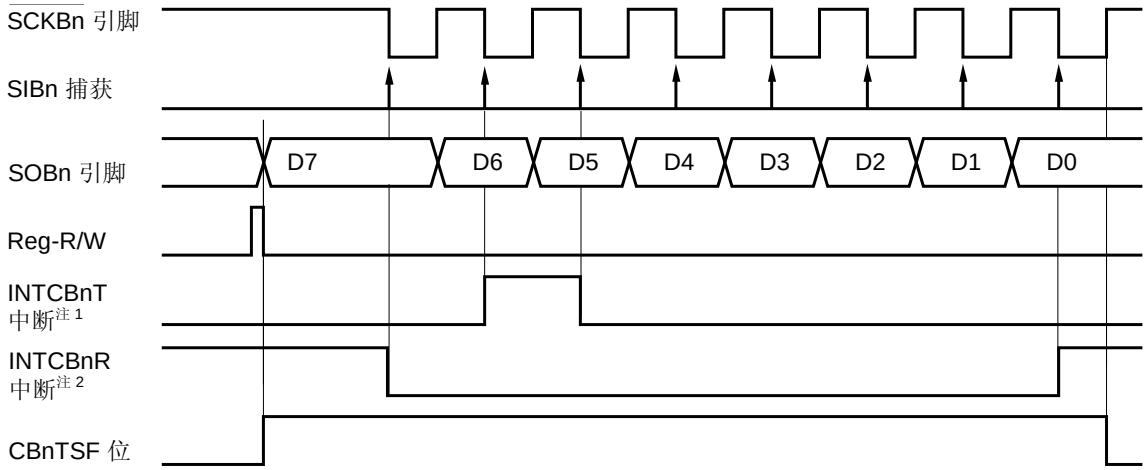
注 **1.** 在连续发送模式或连续发送/接收模式下，当写入 CBnTX 寄存器的数据传输到数据移位寄存器时，INTCBnT 中断置位。在单次发送模式或单次发送/接收模式下，不产生 INTCBnT 中断请求信号，但是在通信结束时产生 INTCBnR 中断请求信号。

2. 如果接收正确结束，且在接收使能的情况下，下一个接收数据在 CBnRX 寄存器内就绪，则产生 INTCBnR 中断。在单次模式下，即使在发送模式下，仍然在通信结束时产生 INTCBnR 中断请求信号。

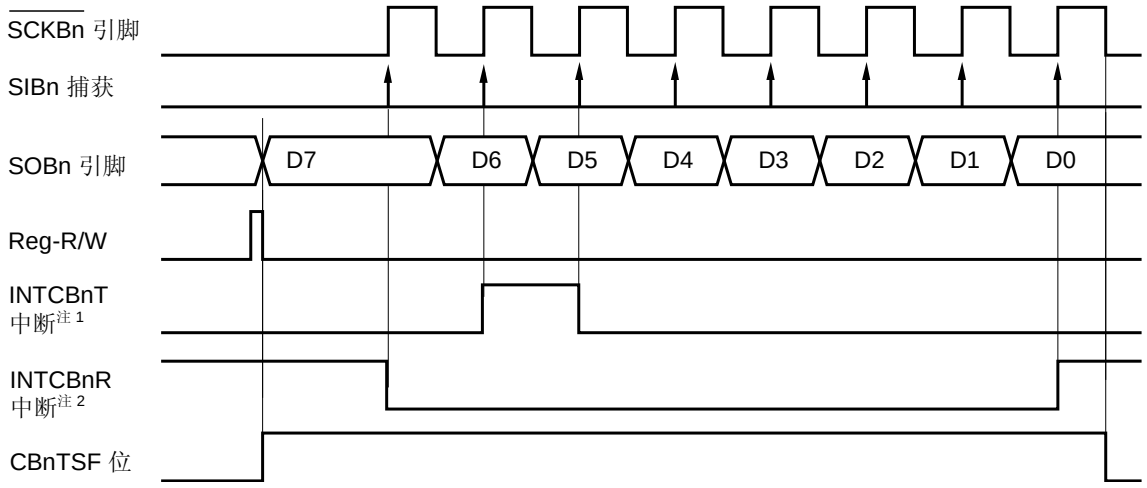
注意事项 在单次传输模式下，CBnTSF 位置 1 时对 CBnTX 寄存器的写入被忽略。这对传输过程的操作没有影响。

图 18-30. 时钟时序（2/2）

（iii）通信类型 2（CBnCKP 和 CBnDAP 位 = 01）



（iv）通信类型 4（CBnCKP 和 CBnDAP 位 = 11）



注 **1.** 在连续发送模式或连续发送/接收模式下，当写入 CBnTX 寄存器的数据被传输到数据移位寄存器时，INTCBnT 中断置位。在单次发送模式或单次发送/接收模式下，不产生 INTCBnT 中断请求信号，但是在通信结束时产生 INTCBnR 中断请求信号。

2. 如果接收正确结束，且在接收使能的情况下，下一个接收数据在 CBnRX 寄存器内就绪，则产生 INTCBnR 中断。在单次模式下，即使在发送模式下，仍然在通信结束时产生 INTCBnR 中断请求信号。

注意事项 **在单次传输模式下，当 CBnTSF 位置为 1 时对 CBnTX 寄存器的写入被忽略。这对传输过程的操作没有影响。**
例如，通过产生的 INTCBnR 信号启动 DMA 时，若下一个数据被写入 CBnTX 寄存器，因为 CBnTSF 位为 1，被写入数据不能传输。
对于这样的应用，应该使用连续传输模式，而不要使用单次传输模式。

18.7 输出引脚

（1）$\overline{\text{SCKBn}}$ 引脚

当 CSIBn 操作禁止（CBnCTL0.CBnPWR 位= 0）时，$\overline{\text{SCKBn}}$ 引脚输出状态如下所示。

表 18-4. CSIBn 禁止时的 $\overline{\text{SCKBn}}$ 引脚输出状态

CBnCKP	CBnCKS2	CBnCKS1	CBnCKS0	$\overline{\text{SCKBn}}$ 引脚输出
0	1	1	1	高阻抗状态
	其它			高电平
1	1	1	1	高阻抗状态
	其它			低电平

备注 **1.** 如果 CBnCTL1.CBnCKP 和 CBnCKS2 至 CBnCKS0 位中有任何位被重写，$\overline{\text{SCKBn}}$ 引脚的输出电平改变。

2. n = 0 至 4

（2）SOBn 引脚

当 CSIBn 操作禁止（CBnPWR 位= 0）时，SOBn 引脚输出状态如下所示。

表 18-5. CSIBn 禁止时的 SOBn 引脚输出状态

CBnTXE	CBnDAP	CBnDIR	SOBn 引脚输出
0	×	×	低电平
1	0	×	低电平
	1	0	CBnTX0 值（MSB）
		1	CBnTX0 值（LSB）

备注 **1.** 如果 CBnCTL0.CBnTXE 位，CBnCTL0.CBnDIR 位以及 CBnCTL1.CBnDAP 位中有任何位被重写，SOBn 引脚输出会改变。

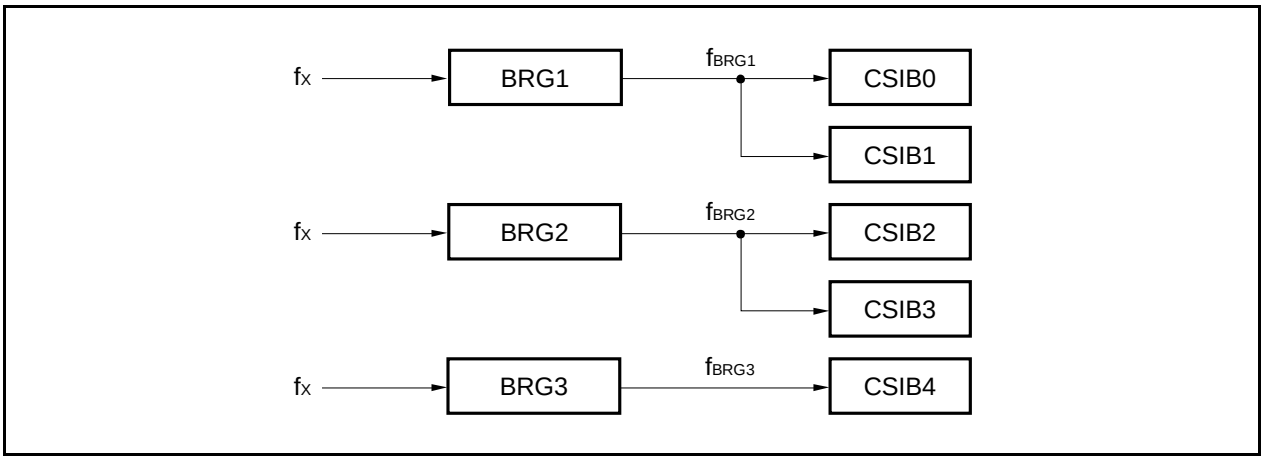
2. ×： 无需理会

3. n = 0 至 4

18.8 波特率发生器

BRG1 至 BRG3 和 CSIB0 至 CSIB4 波特率发生器的连接如下所示：

图 18-31. 波特率发生器的连接



（1）预分频模式寄存器 1 至 3（PRSM1 至 PRSM3）

PRSM1 至 PRSM3 寄存器用于控制 CSIBn 波特率信号的发生。

这些寄存器可以按字节或按位进行读写。

系统复位后，这些寄存器被设置为 00H。

复位后：00H R/W 地址： PRSM1 FFFFF320H, PRSM2 FFFFF324H, PRSM3 FFFFF328H

	7	6	5	<4>	3	2	1	0
PRSMm (m = 1 至 3)	0	0	0	BGCEm	0	0	BGCSm1	BGCSm0

BGCEm	波特率输出
0	禁止
1	使能

BGCSm1	BGCSm0	输入时钟选择 (f_{BGCSm})	设置值 (k)
0	0	f_{XX}	0
0	1	$f_{XX}/2$	1
1	0	$f_{XX}/4$	2
1	1	$f_{XX}/8$	3

注意事项 1. 在操作期间，禁止重写 PRSMm 寄存器。

2. 在设置 BGCEm 位为 1 之前，先设置 BGCSm1 和 BGCSm0 位以及预分频比较寄存器 1 至 3 (PRSCM1 至 PRSCM3)。

3. 请确保将第 7 位至第 5 位，第 3 位和第 2 位清“0”。

（2）预分频比较寄存器 1 至 3（PRSCM1 至 PRSCM3）

PRSCM1 至 PRSCM3 寄存器是 8 位比较寄存器。

这些寄存器可以按字节进行读取或写入。

系统复位后，这些寄存器被设置为 00H。

复位后：00H R/W 地址： PRSCM1 FFFFF321H, PRSCM2 FFFFF325H, PRSCM3 FFFFF329H

	7	6	5	4	3	2	1	0
PRSCMm	PRSCMm7	PRSCMm6	PRSCMm5	PRSCMm4	PRSCMm3	PRSCMm2	PRSCMm1	PRSCMm0

注意事项 1. 操作期间，禁止重写 PRSCMm 寄存器。

2. 在设置 PRSMm.BGCEm 位为 1 之前，先设置 PRSCMm 寄存器。

18.8.1 波特率的产生

通过对主时钟的分频产生发送/接收时钟。从主时钟产生的波特率由以下公式计算获得：

$$f_{BRGm} = \frac{f_{XX}}{2^{k+1} \times N}$$

注意事项 设置 f_{BRGm} 为 8 MHz 或更低。

备注 f_{BRGm}：BRGm 计数时钟

f_{XX}： 主时钟振荡频率

k： PRSMm 寄存器设定值 = 0 至 3

N： PRSCMm 寄存器设定值 = 1 至 256

但是，仅当 PRSCMm 寄存器设置为 00H 时，N = 256。

m = 1 至 3

18.9 注意事项

（1）使用 DMA 传输发送数据和接收数据时，即使串行传输期间发生过载错误，也不执行错误处理。DMA 传输完成后，通过读取 CBnSTR.CBnOVE 位来检查是否发生过载错误。

（2）关于操作期间（CBnCTL0.CBnPWR 位为 1）禁止重写的寄存器，如果操作期间进行了错误的重写，须将 CBnCTL0.CBnPWR 位清 0，然后初始化 CSIBn。

操作期间禁止重写的寄存器列举如下：

- CBnCTL0 寄存器： CBnTXE，CBnRXE，CBnDIR，CBnTMS 位
- CBnCTL1 寄存器： CBnCKP，CBnDAP，CBnCKS2 至 CBnCKS0 位
- CBnCTL2 寄存器： CBnCL3 至 CBnCL0 位

（3）通信类型 2 或 4（CBnCTL1.CBnDAP 位 = 1）中，在发生接收完成中断（INTCBnR）后半个 $\overline{\text{SCKB}}$n 时钟周期时，CBnSTR.CBnTSF 位被清除。

单次传输模式下，通信期间（CBnTSF 位 = 1）写入的下一个待发送数据被忽略，并且不启动下次通信。同样如果设置为只接收通信（CBnCTL0.CBnTXE 位 = 0，CBnCTL0.CBnRXE 位 = 1），即使在通信期间读取接收数据，也不启动下一次通信。

因此，在通信类型 2 或 4 时使用单次传输模式，需特别注意以下情况：

- 若要启动下一次发送，在确认 CBnTSF 位为 0 之后，将待发送数据写入 CBnTX 寄存器。
- 设置为只接收通信（CBnTXE 位 = 0，CBnRXE 位 = 1）时，若要连续执行下一次接收，在确认 CBnTSF 位为 0 之后，读取 CBnRX 寄存器。

或者使用连续传输模式来代替单次传输模式。尤其在用 DMA 传输时，建议使用连续传输模式。

（4） SIB1 和 RXDC0 功能不能同时使用。使用 SIB1 引脚时，停止 UARTC0 接收（将 UC0CTL0.UC0RXE 位清除为 0）。而使用 RXDC0 引脚时，停止 CSIB1 接收（将 CB1CTL0.CB1RXE 位清除为 0）。

备注 n = 0 至 4

第十九章 I²C 总线

要使用 I²C 总线功能，需设置 P38/SDA00，P39/SCL00，P40/SDA01，P41/SCL01，P90/SDA02 和 P91/SCL02 引脚作为串行发送/接收数据输入/输出引脚（SDA00 至 SDA02）和串行时钟输入/输出引脚（SCL00 至 SCL02），并且将它们设置为 N 沟道漏极开路输出。

19.1 I²C总线和其它串行接口的模式转换

19.1.1 UARTA2 和I²C00 模式转换

V850ES/JG3-L 单片机中，UARTA2 和 I²C00 是共享引脚的复用功能，因此不能同时使用。使用前，通过 PMC3 和 PFC3 寄存器预先设置 I²C00。

注意事项 如果发送或接收期间切换这些功能，则 UARTA2 和 I²C00 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 19-1. UARTA2 和 I²C00 模式转换设置

复位后：0000H R/W 地址：FFFFF446H, FFFFF447H

	15	14	13	12	11	10	9	8
PMC3	0	0	0	0	0	0	PMC39	PMC38
	7	6	5	4	3	2	1	0
	0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

复位后：0000H R/W 地址：FFFFF466H, FFFFF467H

	15	14	13	12	11	10	9	8
PFC3	0	0	0	0	0	0	PFC39	PFC38
	7	6	5	4	3	2	1	0
	0	0	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	I²C00 模式

备注 **1.** n = 8，9
2. × = 无需理会

19.1.2 CSIB0 和I²C01 模式转换

V850ES/JG3-L 单片机中，CSIB0 和 I²C01 是共享引脚的复用功能，因此不能同时使用。使用前，通过 PMC4 和 PFC4 寄存器预先设置 I²C01。

注意事项 如果发送或接收期间切换这些功能，则 CSIB0 和 I²C01 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 19-2. CSIB0 和 I²C01 模式转换设置

复位后: 00H R/W 地址: FFFFF448H

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

复位后: 00H R/W 地址: FFFFF468H

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

PMC4n	PFC4n	操作模式
0	x	端口 I/O 模式
1	0	CSIB0 模式
1	1	I²C01 模式

备注 **1.** n = 0，1
2. × = 无需理会

19.1.3 UARTA1 和I²C02 模式转换

V850ES/JG3-L 单片机中，UARTA1 和 I²C02 是共享引脚的复用功能，因此不能同时使用。使用前，通过 PMC9、 PFC9 和 PFCE9 寄存器预先设置 I²C02。

注意事项 如果发送或接收期间切换这些功能，则 UARTA1 和 I²C02 的发送/接收操作得不到保证。确保禁止未使用的功能。

图 19-3. UARTA1 和 I²C02 模式转换设置

复位后：0000H R/W 地址：FFFFF452H, FFFFF453H

	15	14	13	12	11	10	9	8
PMC9	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

	15	14	13	12	11	10	9	8
PFC9	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

	15	14	13	12	11	10	9	8
PFCE9	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	I²C02 模式

备注 n = 0，1

19.2 特性

I^2C00 至 I^2C02 具有下列两种模式：

- 操作停止模式
- I^2C（IC 内部）总线模式（支持多主设备）

（1）操作停止模式

在该模式下，不执行串行传输，因此可以降低功耗。

（2） I^2C 总线模式（支持多主设备）

该模式通过两条信号线在多个设备之间进行 8 位数据传输：一个串行时钟引脚（SCL0n）和一个串行数据总线引脚（SDA0n）。

该模式符合 I^2C 总线规范格式，主设备可以通过其串行数据总线，为从设备产生“开始条件”、“地址”、“传输方向规范”、“传据”和“停止条件”等数据。从设备通过硬件自动检测接收到的状态和数据。这种功能可以使控制 I^2C 总线的应用程序得到简化。

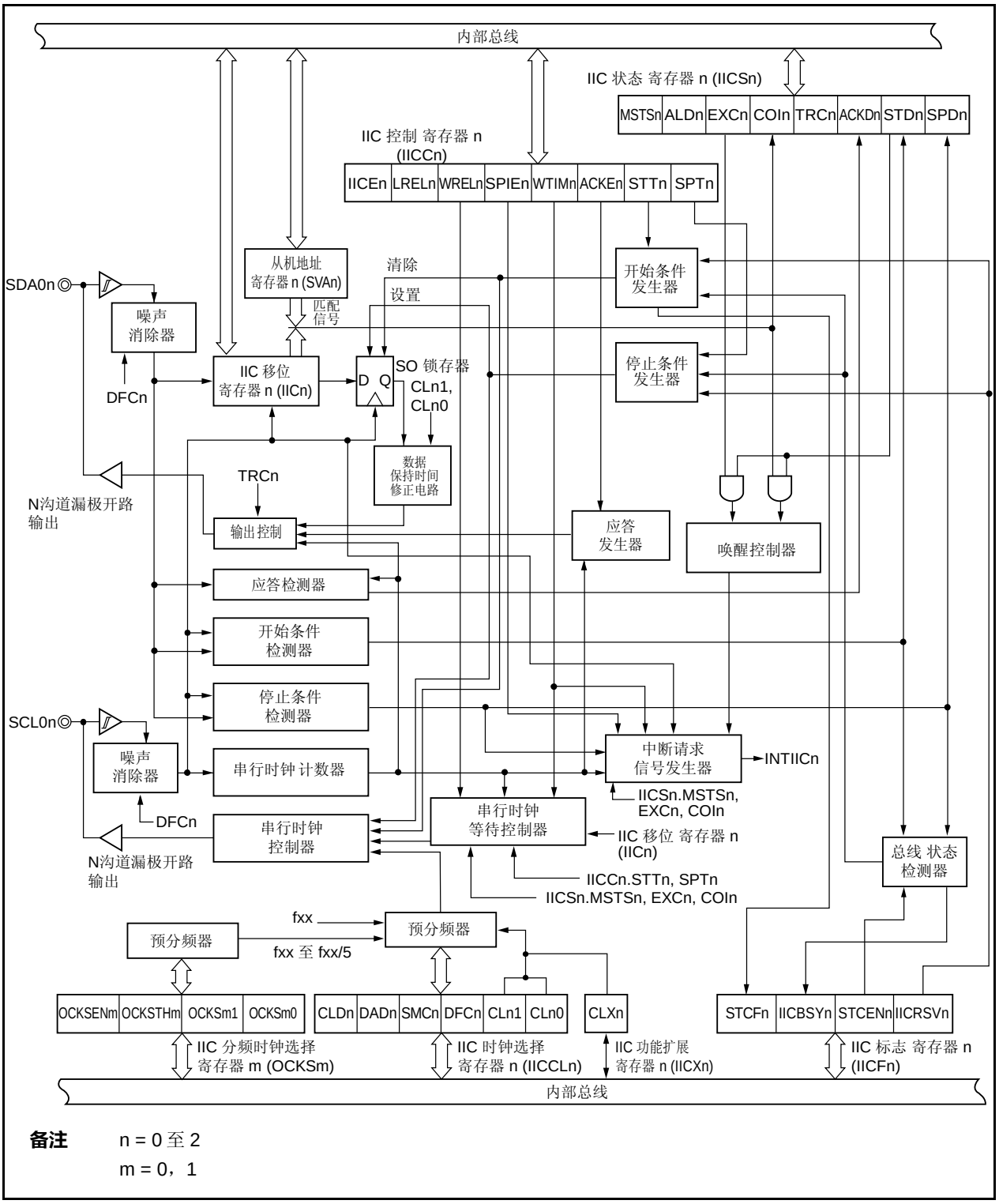
由于 SCL0n 和 SDA0n 引脚被用于 N 沟道漏极开路输出，因此 I^2C0n 的串行时钟线和串行数据线都需要上拉电阻。

备注 n = 0 至 2

19.3 配置

I²C0n 的功能框图如下所示：

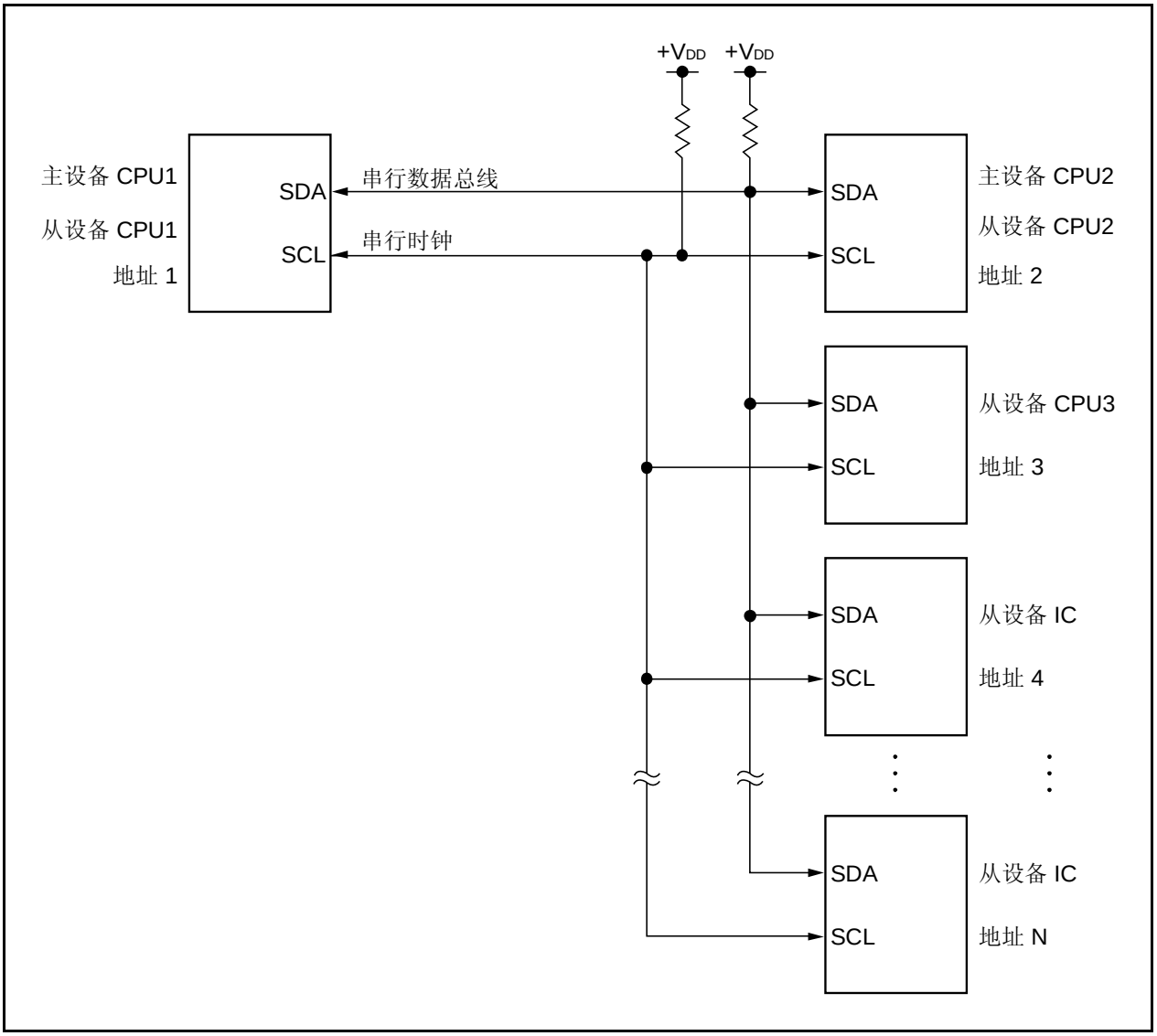
图 19-4. I²C0n 的框图



备注 n = 0 至 2
m = 0，1

以下展示了一个串行总线结构配置示例。

图 19-5. 使用 I²C 总线的串行总线配置示例



I²C0n 包括下列硬件（n = 0 至 2）。

表 19-1. I²C0n 的配置

项目	配置
寄存器	IIC 移位寄存器 n（IICn） 从设备地址寄存器 n（SVAn）
控制寄存器	IIC 控制寄存器 n（IICCn） IIC 状态寄存器 n（IICSn） IIC 标志寄存器 n（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 寄存器将 8 位串行数据转换为 8 位并行数据，也可将 8 位并行数据转换为 8 为串行数据，可用于发送和接收（n = 0 至 2）。

对 IICn 寄存器的写入和读取操作被用来控制实际的发送和接收操作。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被置为 00H。

（2） 从设备地址寄存器 n（SVAn）

在从模式下，SVAn 寄存器用于设定本地地址（n = 0 至 2）。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被置为 00H。

（3） SO 锁存器

SO 锁存器用于保持 SDA0n 引脚的输出电平（n = 0 至 2）。

（4） 唤醒控制器

该寄存器收到的地址和 SVAn 寄存器设定的地址相匹配时，或接收到扩展码时，该电路就会产生一个中断请求信号（INTIICn）（n = 0 至 2）。

（5） 预分频器

用于选择使用的采样时钟。

（6） 串行时钟计数器

在发送/接收操作过程中，串行时钟计数器对输出的串行时钟和输入的串行时钟进行计数，且用于验证发送或接收到的 8 位数据。

（7） 中断请求信号发生器

该电路控制中断请求信号（INTIICn）的产生。

以下两种触发都可以产生一个 I²C 中断。

- 串行时钟的第 8 个或第 9 个（由 IICCn.WTIMn 位设定）时钟的下降沿。
- 因为检测到停止条件（由 IICCn.SPIEn 位设定）而发生中断。

备注 n = 0 至 2

（8） 串行时钟控制器

在主模式下，该电路从采样时钟生成时钟并经由 SCL0n 引脚输出（n = 0 至 2）。

（9） 串行时钟等待控制器

该电路控制等待时序。

（10） $\overline{\text{ACK}}$ 发生器、停止条件检测器、开始条件检测器以及 $\overline{\text{ACK}}$ 检测器

这些电路用来产生和检测各种状态。

（11）数据保持时间修正电路

该电路根据相应的 SCL0n 引脚下降沿，为数据产生保持时间。

（12）开始条件发生器

当 IICCn.STTn 位被置位时，产生一个开始条件。

但是，在通信保留禁止状态（IICFn.IICRSVn 位 = 1）下，如果总线未释放（IICFn.IICBSYn 位 = 1），则请求被忽略，且 IICFn.STCFn 位被置为 1。

（13）停止条件发生器

当 IICCn.SPTn 位被置位时，产生一个停止条件。

（14）总线状态检测器

通过检测开始条件和停止条件，确认总线是否被释放。

但是，总线状态不能在操作后立即被检测到，因此使用 IICFn.STCENn 位将总线状态检测器设置为初始状态。

19.4 寄存器

I²C00 至 I²C02 由下列寄存器进行控制：

- IIC 控制寄存器 0 至 2（IICC0 至 IICC2）
- IIC 状态寄存器 0 至 2（IICS0 至 IICS2）
- IIC 标志寄存器 0 至 2（IICF0 至 IICF2）
- IIC 时钟选择寄存器 0 至 2（IICCL0 至 IICCL2）
- IIC 功能扩展寄存器 0 至 2（IICX0 至 IICX2）
- IIC 时钟分频选择寄存器 0，1（OCKS0，OCKS1）

以下寄存器也可以使用：

- IIC 移位寄存器 0 至 2（IIC0 至 IIC2）
- 从设备地址寄存器 0 至 2（SVA0 至 SVA2）

备注 关于复用功能引脚的设置， 参见 表 4-15 **当端口引脚用作复用功能时的设置**。

（1） IIC 控制寄存器 0 至 2（IICC0 至 IICC2）

IICCn 寄存器允许/停止 I²C0n 操作，设置等待时间并设置其它 I²C 操作（n = 0 至 2）。

这些寄存器可以按字节或按位进行读取和写入。但是，当 IICEn 位为 0 时或在等待期间，对 SPIEn、WTIMn 和 ACKEn 位置位。当 IICEn 位由“0”设置为“1”时，这些位也可以被同时置位。

系统复位后，这些寄存器被置为 00H。

（1/4）

复位后： 00H　　R/W　　地址： IICC0 FFFFFD82H， IICC1 FFFFFD92H， IICC2 FFFFFDA2H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCn	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

（n = 0 至 2）

IICEn	I^2Cn 操作允许/禁止的说明
0	操作停止。IICSn 寄存器复位[注1]。内部操作停止。
1	允许操作。
SCL0n 和 SDA0n 信号线是高电平时，请确保将该位设置为 1。	
清零条件（IICEn 位 = 0）	置位条件（IICEn 位= 1）
• 由指令清除。 • 复位后。	• 由指令设置。

LRELn [注2]	退出通信
0	正常操作
1	从当前通信操作退出，并设置为待机模式。在执行之后该设置被自动清除。当接收到本地无关扩展码的情况下也可以使用。 SCL0n 和 SDA0n 信号线被设为高阻状态。 STTn 和 SPTn 位以及 IICSn 寄存器的 MSTSn， EXCn， COIn， TRCn， ACKDn 和 STDn 位被清零。
从通信中退出之后的待机状态一直保持有效，直到遇到下面的通信入口条件。 • 检测到停止条件后，在主模式下重新开始。 • 在开始条件以后，发生地址匹配或接收到扩展码。	
清零条件（LRELn 位 = 0）	置位条件 （LRELn 位 = 1）
• 执行后自动清除。 • 复位后。	• 由指令设置。

WRELn [注2]	等待状态取消控制
0	等待状态不能取消。
1	等待状态被取消。等待取消以后，该设置被自动清除。
清零条件（WRELn 位 = 0）	置位条件（WRELn 位 = 1）
• 执行后自动清除。 • 复位后。	• 由指令设置。

注
1. IICSn 寄存器、IICFn.STCFn 和 IICFn.IICBSYn 位、IICCLn.CLDn 位和 IICCLn.DADn 位被重置。
2. 当 IICEn 位 = 0 时，该标志信号无效。

注意事项 当 SCL0n 信号线为高电平且 SDA0n 信号线为低电平时，如果 I^2Cn 操作使能（IICEn 位 = 1），则立即检测到开始条件。为了避免这种情况，I^2Cn 操作使能以后，立刻使用位处理指令将 LRELn 位设置为 1。

备注 数据被设置后进行读取时，LRELn 和 WRELn 位为 0。

（2/4）

SPIEn 注	当检测到停止条件时，使能/禁止产生中断请求
0	禁止
1	使能

清零条件（SPIEn 位 = 0）	置位条件（SPIEn 位 = 1）
• 由指令清除。 • 复位后。	• 由指令设置。

WTIMn 注	等待状态和产生中断请求的控制
0	在第 8 个时钟的下降沿产生中断请求信号。 主模式： 在输出 8 个时钟以后，时钟输出被置为低电平，并且设置等待状态。 从模式： 在输入 8 个时钟以后，时钟被设置为低电平，并为主设备设置等待状态。
1	在第 9 个时钟信号的下降沿产生中断请求信号。 主模式： 在输出 9 个时钟以后，时钟输出被设置为低电平，并且设置等待状态。 从模式： 在输入 9 个时钟以后，时钟被设置为低电平，并为主设备设置等待状态。

地址传输期间，不论该位如何设置，在第 9 个时钟下降沿处发生中断。地址传输完成后该位的设置生效。主模式下，在地址传输期间的第 9 个时钟下降沿处，插入一个等待状态。对于接收到本地地址的从设备，产生 ACK 信号之后的第 9 个时钟的下降沿处插入一个等待状态。但是，当从设备收到扩展码后，在第 8 个时钟的下降沿插入等待状态。

清零条件（WTIMn 位 = 0）	置位条件（WTIMn 位 = 1）
• 由指令清除。 • 复位后。	• 由指令设置。

ACKEn 注	应答控制
0	禁止应答。
1	允许应答。在第 9 个时钟期间，SDA0n 信号线设置为低电平。

ACKEn 位的设置对从设备的地址接收是无效的，这种情况下，地址匹配时产生 ACK 信号。

但是，在接收扩展码地址时，ACKEn 位的设置有效。在接收到扩展码的系统中设置 ACKEn 位。

清零条件（ACKEn 位 = 0）	置位条件（ACKEn 位 = 1）
• 由指令清除。 • 复位后。	• 由指令设置。

注 当 IICEn 位 = 0 时，该标志信号无效。

备注 n = 0 至 2

（3/4）

STTn	开始条件触发
0	不产生开始条件。
1	当总线被释放（在 STOP 模式下）： 产生一个开始条件（作为主设备启动）。当 SCLn 信号线为高电平时，SDA0n 信号线由高电平变为低电平，这样就可以产生开始条件。经过一段合理的时间之后，SCL0n 信号线变为低电平。 与第三方通信期间： 如果使能通信保留功能（IICFn.IICRSVn 位 = 0）： • 该触发信号作为开始条件保留标志。当设置为 1 时，释放总线然后自动产生一个开始条件。 如果禁止通信保留功能（IICRSVn = 1）： • IICFn.STCFn 位设置为 1，并且 STTn 位的设置信息（1）被清除，该触发信号不能产生开始条件。 处于等待状态（作为主设备）： 等待状态释放后，产生重新开始条件。

设置时序的有关注意事项：

主设备接收：　传输期间不能设置为 1，只能在 ACKEn 位已经设置 0，且从设备已经得知最末接收时，才可以置为 1 。

主设备发送：　在 ACK 期间，通常不会产生开始条件。在第 9 个时钟输出后的等待期间置为 1。

从设备：　即使禁止通信保留功能（IICRSVn 位 = 1），也可以进入通信保留状态。

• 禁止 SPTn 位的同时设置为 1。
• 当 STTn 位设置为 1 时，禁止再次将 STTn 位置 1，直到设置被清 0 为止。

清零条件（STTn 位 = 0）	置位条件（STTn 位 = 1）
• 在通信保留禁止状态，当 STTn 位设置为 1 时。 • 仲裁失败后清除。 • 主设备产生开始条件后清除。 • 当 LRELn 位 = 1 时（通信保存）。 • 当 IICEn 位 = 0（操作停止）时。 • 复位后	• 由指令设置。

备注　**1.** 如果在数据设置后立即读取 STTn 位，其值为 0。
2. n = 0 至 2

（4/4）

SPTn	停止条件触发
0	不产生停止条件。
1	产生停止条件（主设备传输的终结）。 SDA0 信号线变为低电平之后，将 SCL0n 信号线设置为高电平，或者一直等待直到 SCL0n 引脚变为高电平。然后经过一段合理的时间后，SDA0n 信号线由低电平变为高电平，产生停止条件。

设置时序的有关注意事项

主设备接收： 传输期间不能被设置为 1。
只能在 ACKEn 位已经设置为 0，且从设备已经得知最末接收后的等待期间，才可以被置为 1 。

主设备发送： 在 ACK 接收期间，通常不会产生停止条件。在第 9 个时钟输出后的等待期间置为 1。

- 不能和 STTn 位同时置 1。
- 只有在主模式[注]下， SPTn 位才可以设置为 1。
- 在 WTIMn 位已经设置 0 时，如果在 8 个时钟输出后的等待期间将 SPTn 位设置为 1，注意，将在第 9 个时钟的高电平期间产生停止条件。
  在 8 个时钟输出后的等待期间， WTIMn 位应该从 0 变为 1，且在第 9 个时钟输出后的等待期间应该将 SPTn 位设置为 1。
- 当 SPTn 位设置为 1 时，禁止再次将 SPTn 位置为 1，直到设置被清除为 0。

清零条件（SPTn 位 = 0）	置位条件（SPTn 位 = 1）
• 仲裁失败后清除。 • 检测到停止条件后，自动清除。 • 当 LRELn 位 = 1 时（通信保存）。 • IICEn 位 = 0（操作停止）时。 • 复位后。	• 由指令设置。

注 只有在主设备模式下，才可以将 SPTn 位设置为 1。但是，当 IICRSVn 位为 0 时，改变操作使能状态后检测到第一个停止条件前， SPTn 位必须设置为 1 并产生一个停止条件。详情参见 **19.15 注意事项。**

注意事项 当 TRCn 位 = 1 时，在第 9 个时钟期间 WRELn 位被置 1 且等待状态被取消，之后 TRCn 位被清除为 0 且 SDA0n 信号线被置为高阻状态。

备注 **1.** 如果在数据设置以后立即读取 SPTn 位，其值为 0。
2. n = 0 至 2

（2） IIC 状态寄存器 0 至 2（IICS0 至 IICS2）

IICSn 寄存器表明 I²C0n 的状态（n = 0 至 2）。

这些寄存器只能按字节或按位进行读取。但是，只有当 IICCn.STTn 位为 1 时或在等待期间才能读取 IICSn 寄存器。

系统复位后，这些寄存器被置为 00H。

注意事项 在以下状态时，禁止访问 IICSn 寄存器。详情参见 3.4.9（1）访问专用片上外设 I/O 寄存器。

- **CPU 运行于副时钟且主时钟振荡停止时**
- **CPU 运行于内部振荡时钟时**

（1/3）

复位后： 00H R 地址： IICS0 FFFFFD86H， IICS1 FFFFFD96H， IICS2 FFFFFDA6H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

（n = 0 至 2）

MSTSn	主设备状态
0	从设备状态或通信待机状态
1	主设备通信状态

清零条件（MSTSn 位 = 0）	置位条件（MSTSn 位 = 1）
• 当检测到停止条件时。 • 当 ALDn 位= 1 时（仲裁失败）。 • 由 LRELn 位 = 1 清除（通信保存）。 • 当 IICEn 位 由 1 变为 0（工作停止）时。 • 复位后。	• 当产生开始条件时。

ALDn	仲裁失败检测
0	该状态意味着不存在总线仲裁，或仲裁结果为“成功”。
1	该状态表示仲裁结果为“失败”。MSTSn 位被清除为 0。

清零条件（ALDn 位 = 0）	置位条件（ALDn 位= 1）
• IICSn 寄存器被读取[注]后， 自动清除。 • 当 IICEn 位由 1 变为 0（工作停止）时。 • 复位后。	• 当总线仲裁结果为“失败”。

EXCn	扩展码接收的检测
0	未接收到扩展码。
1	接收到扩展码。

清零条件（EXCn 位 = 0）	置位条件（EXCn 位= 1）
• 当检测到开始条件时。 • 当检测到停止条件时。 • 通过 LRELn 位 = 1 清除（通信保存）。 • 当 IICEn 位由 1 变为 0 时（操作停止）。 • 复位后。	• 当接收地址数据的高 4 位是“0000” 或 “1111”时（在第 8 个时钟的上升沿置位）。

注 对 IICSn 寄存器的其它位进行位操作时，该位也被清除。

（2/3）

COIn	匹配地址检测
0	地址不匹配
1	地址匹配

清零条件（COIn 位 = 0）	置位条件（COIn 位= 1）
• 当检测到开始条件时。 • 当检测到停止条件时。 • 通过 LRELn 位 = 1 清除（通信保存）。 • 当 IICEn 位由 1 变为 0（工作停止）时。 • 复位后。	• 当接收到的地址和本地地址（SVAn 寄存器）匹配时（在第 8 个时钟的上升沿置位）。

TRCn	发送/接收状态检测
0	接收状态（除发送状态外）。SDA0n 信号线被置为高阻态。
1	发送状态。允许 SO 锁存器内的数值从 SDA0n 信号线输出（在第一个字节的第 9 个时钟下降沿开始生效）。

清零条件（TRCn 位 = 0）	置位条件（TRCn 位= 1）
• 当检测到停止条件时。 • 通过 LRELn 位 = 1 清除（通信保存）。 • 当 IICEn 位由 1 变为 0（工作停止）时。 •通过 IICCn.WRELn 位 = 1 注清除。 • 当 ALDn 位从 0 变为 1（仲裁失败）时。 • 复位后。 主设备： • 当第一个字节的 LSB（传输方向指示位）输出为“1”时。 从设备： • 当检测到开始条件时。 当不用于通信时。	主设备： • 当产生开始条件时。 • 当第一个字节的 LSB（传输方向指示位）输出为“0”时。 从设备： • 当第一个字节的 LSB（传输方向指示位）输入为“1”时。

ACKDn	ACK 检测
0	未检测到 $\overline{\text{ACK}}$ 。
1	检测到 $\overline{\text{ACK}}$ 。

清零条件（ACKDn 位= 0）	置位条件（ACKD 位 = 1）
• 当检测到停止条件时。 • 在下一字节第一个时钟的上升沿。 • 通过 LRELn 位 = 1 清除（通信从模式）。 • 当 IICEn 位由 1 变为 0（工作停止）时。 • 复位后。	•在 SCL0n 引脚的第 9 个时钟的上升沿 SDA0n 引脚被设置为低电平后。

注 当 WRELn 位设置为 1，且在第九个时钟通过 TRCn 位 = 1 取消等待状态时，TRCn 被清除为 0，而且 SDA0n 信号线变为高阻态。

备注 n = 0 至 2

（3/3）

STDn	开始条件检测
0	未检测到开始条件。
1	检测到开始条件。这表示在地址传输时期有效时段内。

清零条件（STDn 位 = 0）	置位条件（STDn 位 = 1）
• 当检测到停止条件时。 • 在地址传输后下一个字节第一个时钟的上升沿。 • 通过 LRELn 位 = 1 清除（通信保存）。 • 当 IICEn 位 由 1 变为 0（工作停止）时。 • 复位后。	• 当检测到开始条件时。

SPDn	停止条件检测
0	未检测到停止条件。
1	检测到停止条件。主设备的通信被终结，并且释放总线。

清零条件（SPDn 位 = 0）	置位条件（SPDn 位 = 1）
• 在位置该位之后的地址传输第一个时钟上升沿处，以及检测到开始条件时。 • 当 IICEn 位由 1 变为 0（工作停止）时。 • 复位后。	• 当检测到停止条件时。

备注 n = 0 至 2

（3） IIC 标志寄存器 0 至 2（IICF0 至 IICF2）

IICFn 寄存器设置 I^2C0n 操作模式并指示 I^2C 总线状态。

这些寄存器可以按字节或按位进行读取和写入。但是，STCFn 和 IICBSYn 位只读。

IICRSVn 使能/禁止通信保留功能（参见 **19.14 通信保留）。**

IICBSYn 位的初始值通过 STCENn 位设置（参见 **19.15 注意事项）。**

IICRSVn 和 STCENn 位只能在 I^2C0n 操作禁止（IICCn.IICEn 位 = 0）时写入。操作使能后，IICFn 可被读取（n = 0 至 2）。

系统复位后，这些寄存器被置为 00H。

复位后： 00H　　R/W 注　　地址： IICF0 FFFFFD8AH， IICF1 FFFFFD9AH， IICF2 FFFFFDAAH

	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

（n = 0 至 2）

STCFn	STTn 位清除
0	发出开始条件。
1	不发出开始条件，STTn 位被清除。
清零条件（STCFn 位 = 0）	置位条件（STCFn 位= 1）
• 通过 IICCn.STTn 位 = 1 清除。 • 当 IICCn.IICEn 位 = 0 时。 • 复位后。	• 禁止通信保留（IICRSVn 位 = 1）期间，不发出开始条件且 STTn 标志被清除为 0 时。

IICBSYn	I²C0n 总线状态
0	总线释放状态（STCENn 位 = 1 时的默认通信状态）。
1	总线通信状态（STCENn 位 = 0 时的默认通信状态）。
清零条件（IICBSYn 位 = 0）	置位条件（IICBSYn 位 = 1）
• 当检测到停止条件时。 • IICEn 位 = 0 时。 • 复位后。	• 当检测到开始条件时。 • 当 STCENn 位 = 0 时，通过设置 IICEn 位来置位。

STCENn	初始化开始使能触发
0	不能产生开始条件，直到操作使能后（IICEn 位 = 1）检测到停止条件为止。
1	产生开始条件，即使操作使能后（IICEn 位 = 1）没有检测到停止条件。
清零条件（STCENn 位 = 0）	置位条件（STCENn 位 = 1）
• 当检测到开始条件时。 • 复位后。	• 由指令设置。

IICRSVn	通信保留功能禁止位
0	使能通信保留
1	禁止通信保留
清零条件（IICRSVn 位 = 0）	置位条件（IICRSVn 位 = 1）
• 由指令清除。 • 复位后。	• 由指令设置。

注　第 6 位和第 7 位为只读位。

注意事项
1. **仅当操作停止时（IICEn 位 = 0），才可以写入 STCENn 位。**
2. **当 STCENn 位 = 1 时，紧随在 I²Cn 总线操作使能之后，不管总线的实际状态如何，总是认为处于总线释放状态（IICFn.IICBSYn 位 = 0）。因此，要产生首个开始条件（STTn 位 = 1），必需确认总线已经被释放，这样不会干扰其它通信。**
3. **仅当操作停止时（IICEn 位 = 0），才可以写入 IICRSVn 位。**

（4） IIC 时钟选择寄存器 0 至 2（IICCL0 至 IICCL2）

IICCLn 寄存器为 I²C0n 设置传输时钟。

这些寄存器可以按字节或按位进行读取或写入。但是，CLDn 和 DADn 位是只读位。

当 IICCn.IICEn 位 = 0 时，设置 IICCLn 寄存器。

组合使用 SMCn，CLn1 和 CLn0 位以及 IICXn.CLXn 位与 OCKSm 寄存器的 OCKSTHm，OCKSm1 和 OCKSm0 位进行设置（参见 **19.4（6）I²C0n 传输时钟设置方式）**（n = 0 至 2， m = 0，1）。

系统复位后，这些寄存器被置为 00H。

复位后： 00H　　R/W 注　　地址： IICCL0 FFFFFD84H， IICCL1 FFFFFD94H， IICCL2 FFFFFDA4H

	7	6	<5>	<4>	3	2	1	0
IICCLn（n = 0 至 2）	0	0	CLDn	DADn	SMCn	DFCn	CLn1	CLn0

CLDn	SCL0n 引脚电平的检测（仅当 IICCn.IICEn 位 = 1 时有效）
0	SCL0n 引脚检测为低电平。
1	SCL0n 引脚检测为高电平。

清零条件（CLDn 位 = 0）	置位条件（CLDn 位 = 1）
• 当 SCL0n 引脚为低电平时。 • IICEn 位 = 0（操作停止）时。 • 复位后。	• 当 SCL0n 引脚为高电平时。

DADn	SDA0n 引脚电平的检测（仅当 IICEn 位 = 1 时有效）
0	SDA0n 引脚检测为低电平。
1	SDA0n 引脚检测为高电平。

清零条件（DADn 位 = 0）	置位条件（DAD0n 位 = 1）
• 当 SDA0n 引脚为低电平。 • IICEn 位 = 0（操作停止）时。 • 复位后。	• 当 SDA0n 引脚为高电平时。

SMCn	工作模式切换
0	工作于标准模式。
1	工作于高速模式。

DFCn	数字滤波器操作控制
0	数字滤波器关闭。
1	数字滤波器开启。
数字滤波器只能在高速模式下使用。 在高速模式下，不论 DFCn 位如何设置（开启/关闭），传输时钟都不会改变。 数字滤波器用于消除高速模式下的噪声。	

注　第 4 位和第 5 位是只读位。

注意事项　请确保将第 7 位和第 6 位清除为“0”。

备注　当 IICCn.IICEn 位 = 0 时，读取 CLDn 和 DADn 位，其值为 0。

（5） IIC 功能扩展寄存器 0 至 2（IICX0 至 IICX2）

IICXn 寄存器设置 I²C0n 功能扩展（仅在高速模式下有效）。

这些寄存器可以按字节或按位进行读取或写入。

组合使用 IICCLn 寄存器的 SMCn，CLn1 和 CLn0 位以及 OCKSm 寄存器的 OCKSTHm，OCKSm1 和 OCKSm0 位，对 CLXn 位进行设置（参见 **19.4（6）I²C0n 传输时钟设置方式）**（m = 0，1）。

当 IICCn.IICEn 位 = 0 时，设置 IICXn 寄存器。

系统复位后，这些寄存器被置为 00H。

复位后：00H　R/W　地址：IICX0 FFFFFD85H, IICX1 FFFFFD95H, IICX2 FFFFFDA5H

	7	6	5	4	3	2	1	<0>
IICXn	0	0	0	0	0	0	0	CLXn

(n = 0 至 2)

（6） I²C0n 传输时钟设置方式

使用下列表达式计算 I²C0n 传输时钟频率（f_{SCL}）（n = 0 至 2）。

$$f_{SCL} = 1/（m \times T + t_R + t_F）$$

m = 12，18，24，36，44，48，54，60，66，72，86，88，96，132，172，176，198，220，258，344（参见**表 19-2 时钟设置**）

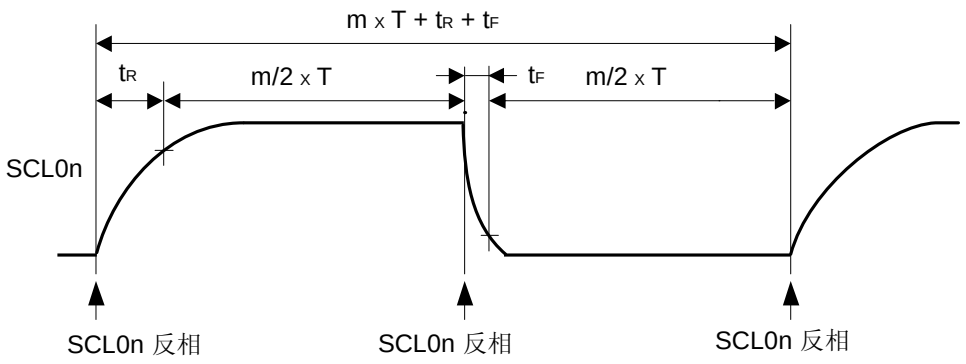
T： 1/f_{XX}

t_R： SCL0n 引脚上升时间

t_F： SCL0n 引脚下降时间

例如，当 f_{XX} = 19.2 MHz， m = 198， t_R = 200 ns 且 t_F = 50 ns 时，I²C0n 传输时钟频率（f_{SCL}）使用下面表达式计算。

$$f_{SCL} = 1/（198 \times 52\ ns + 200\ ns + 50\ ns）\cong 94.7\ kHz$$



组合使用 IICCLn 寄存器的 SMCn，CLn1 和 CLn0 位以及 IICXn 寄存器的 CLXn 位、OCKSm 寄存器的 OCKSTHm，OCKSm1 和 OCKSm0 位来设置选择时钟（n = 0 至 2，m = 0，1）。

表 19-2. 时钟设置（1/2）

IICX0	IICCL0			选择时钟	传输时钟	可设置的主时钟频率（fxx）范围	操作模式
第 0 位	第 3 位	第 1 位	第 0 位				
CLX0	SMC0	CL01	CL00				
0	0	0	0	fxx（当 OCKS0 = 18H 时）	fxx/44	2.50 MHz ≤ fxx ≤ 4.19 MHz	标准模式（SMC0 位 = 0）
				fxx/2（当 OCKS0 = 10H 时）	fxx/88	4.00 MHz ≤ fxx ≤ 8.38 MHz	
				fxx/3（当 OCKS0 = 11H 时）	fxx/132	6.00 MHz ≤ fxx ≤ 12.57 MHz	
				fxx/4（当 OCKS0 = 12H 时）	fxx/176	8.00 MHz ≤ fxx ≤ 16.76 MHz	
				fxx/5（当 OCKS0 = 13H 时）	fxx/220	10.00 MHz ≤ fxx ≤ 20.00 MHz	
0	0	0	1	fxx（当 OCKS0 = 18H 时）	fxx/86	4.19 MHz ≤ fxx ≤ 8.38 MHz	
				fxx/2（当 OCKS0 = 10H 时）	fxx/172	8.38 MHz ≤ fxx ≤ 16.76 MHz	
				fxx/3（当 OCKS0 = 11H 时）	fxx/258	12.57 MHz ≤ fxx ≤ 20.00 MHz	
				fxx/4（当 OCKS0 = 12H 时）	fxx/344	16.76 MHz ≤ fxx ≤ 20.00 MHz	
0	0	1	0	fxx 注	fxx/86	4.19 MHz ≤ fxx ≤ 8.38 MHz	
0	0	1	1	fxx（当 OCKS0 = 18H 时）	fxx/66	fxx = 6.40 MHz	
				fxx/2（当 OCKS0 = 10H 时）	fxx/132	fxx = 12.80 MHz	
				fxx/3（当 OCKS0 = 11H 时）	fxx/198	fxx = 19.20 MHz	
0	1	0	×	fxx（当 OCKS0 = 18H 时）	fxx/24	4.19 MHz ≤ fxx ≤ 8.38 MHz	高速模式（SMC0 位 = 1）
				fxx/2（当 OCKS0 = 10H 时）	fxx/48	8.00 MHz ≤ fxx ≤ 16.76 MHz	
				fxx/3（当 OCKS0 = 11H 时）	fxx/72	12.00 MHz ≤ fxx ≤ 20.00 MHz	
				fxx/4（当 OCKS0 = 12H 时）	fxx/96	16.00 MHz ≤ fxx ≤ 20.00 MHz	
0	1	1	0	fxx 注	fxx/24	4.00 MHz ≤ fxx ≤ 8.38 MHz	
0	1	1	1	fxx（当 OCKS0 = 18H 时）	fxx/18	fxx = 6.40 MHz	
				fxx/2（当 OCKS0 = 10H 时）	fxx/36	fxx = 12.80 MHz	
				fxx/3（当 OCKS0 = 11H 时）	fxx/54	fxx = 19.20 MHz	
1	1	0	×	fxx（当 OCKS0 = 18H 时）	fxx/12	4.00 MHz ≤ fxx ≤ 4.19 MHz	
				fxx/2（当 OCKS0 = 10H 时）	fxx/24	8.00 MHz ≤ fxx ≤ 8.38 MHz	
				fxx/3（当 OCKS0 = 11H 时）	fxx/36	12.00 MHz ≤ fxx ≤ 12.57 MHz	
				fxx/4（当 OCKS0 = 12H 时）	fxx/48	16.00 MHz ≤ fxx ≤ 16.67 MHz	
				fxx/5（当 OCKS0 = 13H 时）	fxx/60	fxx = 20.00 MHz	
1	1	1	0	fxx 注	fxx/12	4.00 MHz ≤ fxx ≤ 4.19 MHz	
其它				禁止设置	–	–	–

注 由于选择时钟是 fxx，与 OCKS0 寄存器的设定值无关，所以将 OCKS0 寄存器清除为 00H（I²C 分频时钟停止状态）。

备注 ×：无需理会

表 19-2. 时钟设置（2/2）

IICXm	IICCLm			选择时钟	传输时钟	可设置的主时钟频率（fxx）范围	操作模式
第 0 位	第 3 位	第 1 位	第 0 位				
CLXm	SMCm	CLm1	CLm0				
0	0	0	0	fxx (当 OCKS1 = 18H 时)	fxx/44	2.50 MHz ≤ fxx ≤ 4.19 MHz	标准模式（SMCm 位 = 0）
				fxx/2 (当 OCKS1 = 10H 时)	fxx/88	4.00 MHz ≤ fxx ≤ 8.38 MHz	
				fxx/3 (当 OCKS1 = 11H 时)	fxx/132	6.00 MHz ≤ fxx ≤ 12.57 MHz	
				fxx/4 (当 OCKS1 = 12H 时)	fxx/176	8.00 MHz ≤ fxx ≤ 16.76 MHz	
				fxx/5 (当 OCKS1 = 13H 时)	fxx/220	10.00 MHz ≤ fxx ≤ 20.00 MHz	
0	0	0	1	fxx (当 OCKS1 = 18H 时)	fxx/86	4.19 MHz ≤ fxx ≤ 8.38 MHz	
				fxx/2 (当 OCKS1 = 10H 时)	fxx/172	8.38 MHz ≤ fxx ≤ 16.76 MHz	
				fxx/3 (当 OCKS1 = 11H 时)	fxx/258	12.57 MHz ≤ fxx ≤ 20.00 MHz	
				fxx/4 (当 OCKS1 = 12H 时)	fxx/344	16.76 MHz ≤ fxx ≤ 20.00 MHz	
0	0	1	0	fxx 注	fxx/86	4.19 MHz ≤ fxx ≤ 8.38 MHz	
0	0	1	1	fxx (当 OCKS1 = 18H 时)	fxx/66	fxx = 6.40 MHz	
				fxx/2 (当 OCKS1 = 10H 时)	fxx/132	fxx = 12.80 MHz	
				fxx/3 (当 OCKS1 = 11H 时)	fxx/198	fxx = 19.20 MHz	
0	1	0	×	fxx (当 OCKS1 = 18H 时)	fxx/24	4.19 MHz ≤ fxx ≤ 8.38 MHz	高速模式（SMCm 位 = 1）
				fxx/2 (当 OCKS1 = 10H 时)	fxx/48	8.00 MHz ≤ fxx ≤ 16.76 MHz	
				fxx/3 (当 OCKS1 = 11H 时)	fxx/72	12.00 MHz ≤ fxx ≤ 20.00 MHz	
				fxx/4 (当 OCKS1 = 12H 时)	fxx/96	16.00 MHz ≤ fxx ≤ 20.00 MHz	
0	1	1	0	fxx 注	fxx/24	4.00 MHz ≤ fxx ≤ 8.38 MHz	
0	1	1	1	fxx (当 OCKS1 = 18H 时)	fxx/18	fxx = 6.40 MHz	
				fxx/2 (当 OCKS1 = 10H 时)	fxx/36	fxx = 12.80 MHz	
				fxx/3 (当 OCKS1 = 11H 时)	fxx/54	fxx = 19.20 MHz	
1	1	0	×	fxx (当 OCKS1 = 18H 时)	fxx/12	4.00 MHz ≤ fxx ≤ 4.19 MHz	
				fxx/2 (当 OCKS1 = 10H 时)	fxx/24	8.00 MHz ≤ fxx ≤ 8.38 MHz	
				fxx/3 (当 OCKS1 = 11H 时)	fxx/36	12.00 MHz ≤ fxx ≤ 12.57 MHz	
				fxx/4 (当 OCKS1 = 12H 时)	fxx/48	16.00 MHz ≤ fxx ≤ 16.67 MHz	
				fxx/5 (当 OCKS1 = 13H 时)	fxx/60	fxx = 20.00 MHz	
1	1	1	0	fxx 注	fxx/12	4.00 MHz ≤ fxx ≤ 4.19 MHz	
其它				禁止设置	–	–	–

注 由于选择时钟是 fxx，与 OCKS1 寄存器设定值无关，所以将 OCKS1 寄存器清除为 00H（I²C 分频时钟停止状态）。

备注 **1.** m = 1，2

2. ×：无需理会

（7）IIC 分频时钟选择寄存器 0，1（OCKS0，OCKS1）

OCKSm 寄存器控制 I^2C0n 分频时钟（n = 0 至 2，m = 0，1）

这些寄存器通过 OCKS0 寄存器控制 I^2C00 分频时钟，通过 OCKS1 寄存器控制 I^2C01 和 I^2C02 分频时钟。

这些寄存器可以按字节进行读取或写入。

系统复位后，这些寄存器被置为 00H。

复位后：00H　R/W　地址：OCKS0 FFFFF340H, OCKS1 FFFFF344H

	7	6	5	4	3	2	1	0
OCKSm (m = 0, 1)	0	0	0	OCKSENm	OCKSTHm	0	OCKSm1	OCKSm0

OCKSENm	I^2C 分频时钟的操作设置
0	禁止 I^2C 分频时钟工作
1	使能 I^2C 分频时钟工作

OCKSTHm	OCKSm1	OCKSm0	I^2C 分频时钟的选择
0	0	0	$f_{XX}/2$
0	0	1	$f_{XX}/3$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/5$
1	0	0	f_{XX}
其它			禁止设置

（8） IIC 移位寄存器 0 至 2（IIC0 至 IIC2）

IICn 移位寄存器和串行时钟同步，用于串行发送/接收（移位操作）。这些移位寄存器可以按字节读取或写入，但是，在数据传输期间，不应该向 IICn 移位寄存器写入数据。

只有在等待期间，可以访问（读/写）IICn 移位寄存器 。在等待期间以外的其它通信状态时，禁止访问该寄存器。但是，对于主设备，只有在传输触发位（IICCn.STTn 位）被置为 1 后， IICn 移位寄存器才可以写入一次。

在等待期间写入 IICn 寄存器，释放一个等待状态，且启动数据传输（n = 0 至 2）。

系统复位后，这些寄存器被置为 00H。

复位后： 00H　R/W　地址： IIC0 FFFFFD80H， IIC1 FFFFFD90H， IIC2 FFFFFDA0H

	7	6	5	4	3	2	1	0
IICn （n = 0 至 2）								

（9） 从设备地址寄存器 0 至 2（SVA0 至 SVA2）

SVAn 寄存器保持 I²C 总线的从地址。

这些移位寄存器可以按字节读取或写入，第 0 位固定为 0。但是当 IICSn.STDn 位 = 1（开始条件检测）时，禁止重写该寄存器。

系统复位后，这些寄存器被置为 00H。

复位后： 00H　　R/W　　地址： SVA0 FFFFFD83H， SVA1 FFFFFD93H， SVA2 FFFFFDA3H

	7	6	5	4	3	2	1	0
SVAn								0

（n = 0 至 2）

19.5 I²C 总线模式功能

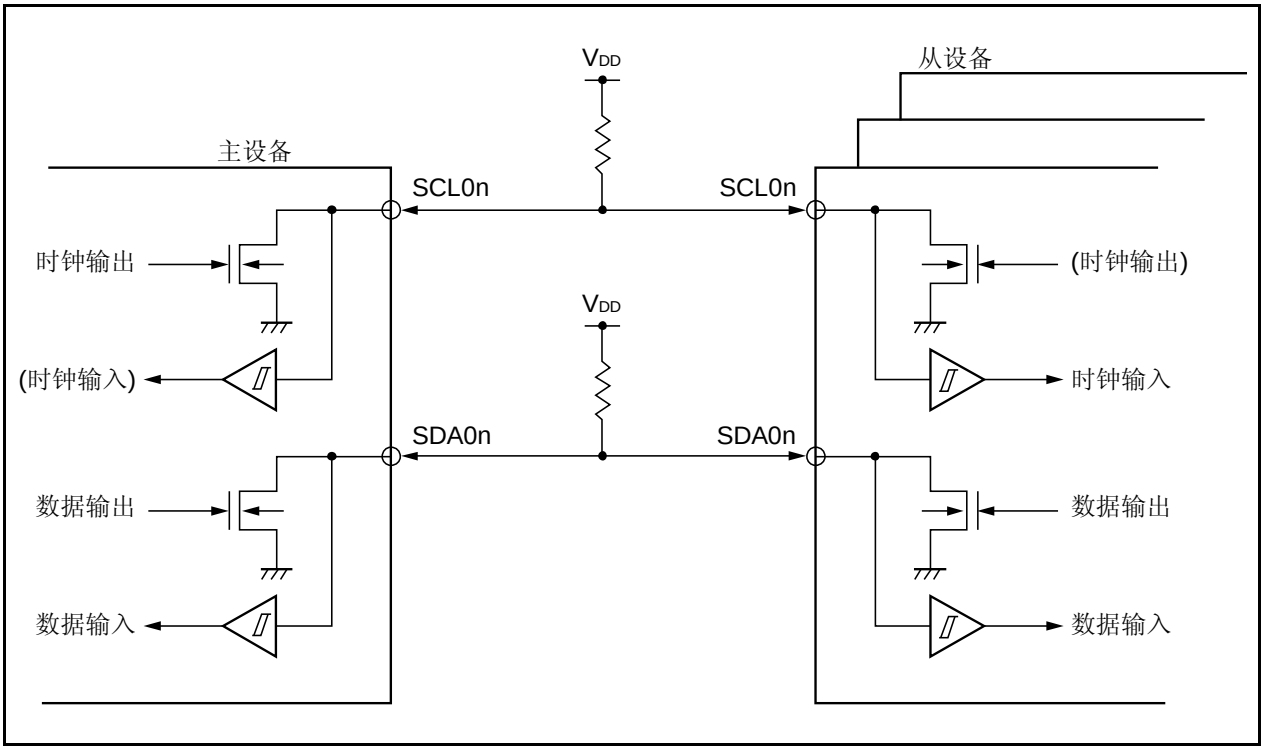
19.5.1 引脚配置

串行时钟引脚（SCL0n）和串行数据总线引脚（SDA0n）配置如下（n = 0 至 2）。

SCL0n .....该引脚用于串行时钟输入和输出。
对于主设备和从设备，该引脚都是 N 沟道漏极开路输出，输入为施密特输入。
SDA0n .....该引脚用于串行数据输入和输出。
对于主设备和从设备，该引脚都是 N 沟道漏极开路输出，输入为施密特输入。

由于串行时钟信号线和串行数据总线输出都是 N 沟道漏极开路输出，所以都需要外接上拉电阻。

图 19-6. 引脚配置图

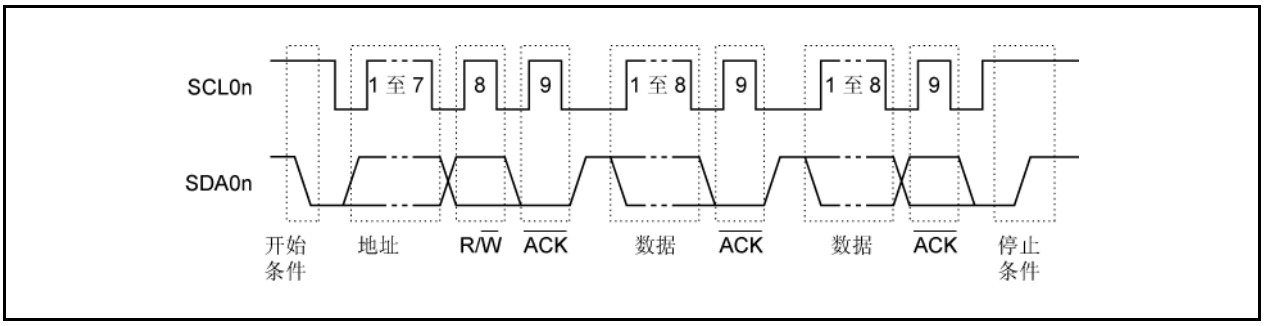


19.6 I²C总线的定义和控制方法

以下章节描述 I²C 总线的串行数据通信格式和 I²C 总线使用的信号。

关于 I²C 总线的串行数据总线产生的“开始条件”、“地址”、“传输方向”、“数据”以及“停止条件”的传输时序，如下所示。

图 19-7. I²C 总线串行数据传输时序



主设备产生开始条件、从设备地址和停止条件。

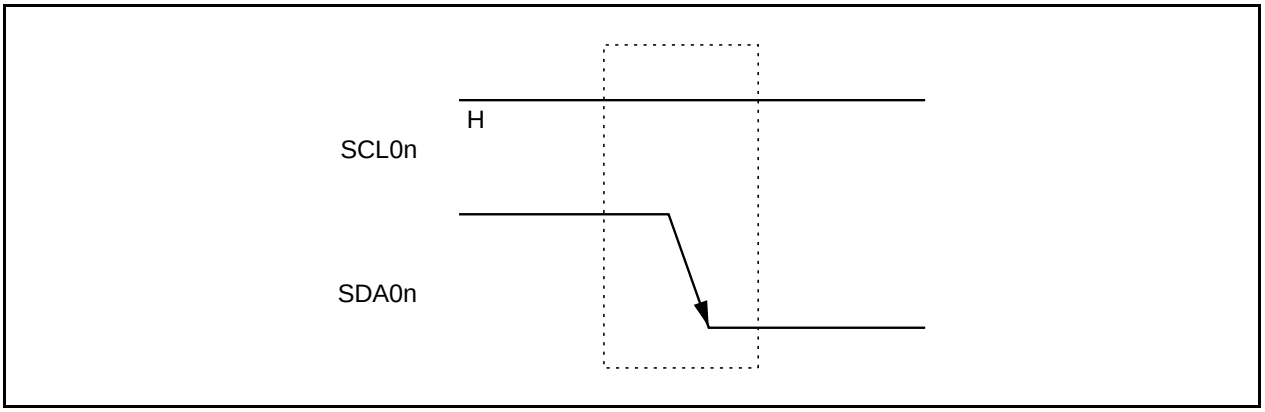
ACK 信号既可由主设备产生，也可由从设备产生。（通常情况下，它由接收 8 位数据的设备产生）。

串行时钟（SCL0n）由主设备连续输出。但是，在从设备中 SCL0n 引脚的低电平时段可以被延长，可以插入一个等待状态（n = 0 至 2）。

19.6.1 开始条件

当 SCL0n 引脚为高电平而且 SDA0n 引脚由高电平变为低电平时，形成一个开始条件。SCL0n 和 SDA0n 引脚的开始条件是启动串行传输时主设备输出至从设备的信号。从设备可以检测开始条件（n = 0 至 2）。

图 19-8. 开始条件



当检测到停止条件（IICSn.SPDn 位 = 1）后，当 IICC0.STT0 位被置位（1）时，输出一个开始条件。当检测到开始条件时，IICSn.STDn 位被置位（1）。（n = 0 至 2）

注意事项 在和其它设备进行通信时，当 V850ES/JG3-L 的 IICCn.IICEn 位被设置为 1 时，根据通信线的状态，可能会检测到开始条件。SCL0n 和 SDA0n 信号线是高电平时，请确保将 IICCn.IICEn 位设置为 1。

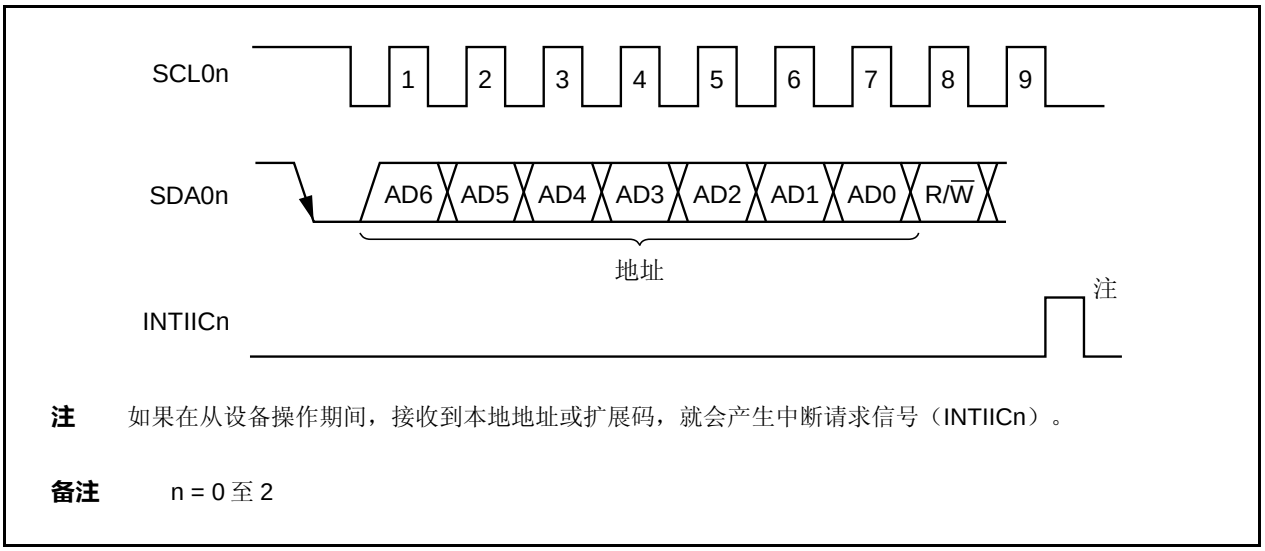
19.6.2 地址

紧随开始条件之后的 7 位数据被定义为地址。

地址是 7 位数据段，输出地址用来选择通过总线连接到主设备的其中一个从设备。所以，每一个接入总线的从设备必须具有一个独一无二的地址。

从设备包含检测开始条件的硬件，并检查 7 位地址数据是否和 SVAn 寄存器内储存的数值匹配。如果地址数据和 SVAn 寄存器内储存的数据值相匹配，这个从设备就被选择，并且和主设备进行通信，直到主设备产生一个开始条件或结束条件。（n = 0 至 2）

图 19-9. 地址



注 如果在从设备操作期间，接收到本地地址或扩展码，就会产生中断请求信号（INTIICn）。

备注 n = 0 至 2

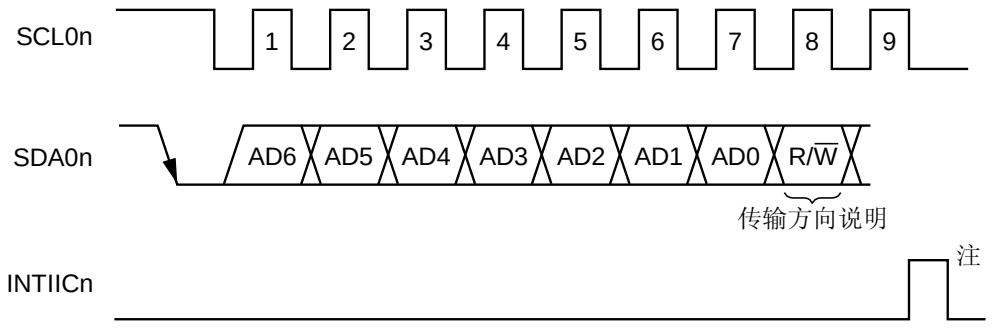
从设备地址和第 8 位一起被写入 IIC 移位寄存器 n（IICn），然后被输出。接收到的地址数据被写入 IICn 寄存器（n = 0 至 2）。第 8 位用来指示传输方向，在 **19.6.3 指定传输方向**中描述。

从设备地址被分配至 IICn 寄存器的高 7 位 。

19.6.3 传输方向指示

除了 7 位地址数据之外，主设备发送 1 位数据指示传输方向。当该传输方向指示位为 0 时，表示主设备向从设备传输数据。当该传输方向指示位为 1 时，表示主设备由从设备接收数据。

图 19-10. 传输方向指示



注 如果在从设备操作期间，接收到本地地址或扩展码，就会产生 INTIICn 信号。

备注 n = 0 至 2

19.6.4 $\overline{\text{ACK}}$

$\overline{\text{ACK}}$ 用来确认发送设备和接收设备的串行数据传输状态。

每接收到的 8 位数据，接收设备返回一次 $\overline{\text{ACK}}$ 信号。

通常情况下，发送设备发送 8 位数据后，接收到一个 $\overline{\text{ACK}}$ 信号。当 $\overline{\text{ACK}}$ 信号由接收端返回后，接收被判定为正常并且继续进行操作。使用 IICSn.ACKDn 位对 ACK $\overline{\text{信号}}$的检测进行确认。

当主设备是接收设备时，收到最后一个数据之后，不返回 $\overline{\text{ACK}}$ 信号且产生停止条件。当从设备为接收设备且不返回 ACK $\overline{\text{信号}}$时，主设备产生一个停止条件或一个重新开始条件，然后停止当前的通信传输。返回 ACK $\overline{\text{失败}}$的情况可由以下因素导致：

（a） 接收没有正常执行。
（b） 接收到最末数据。
（c） 指定地址上的接收设备（从设备）不存在。

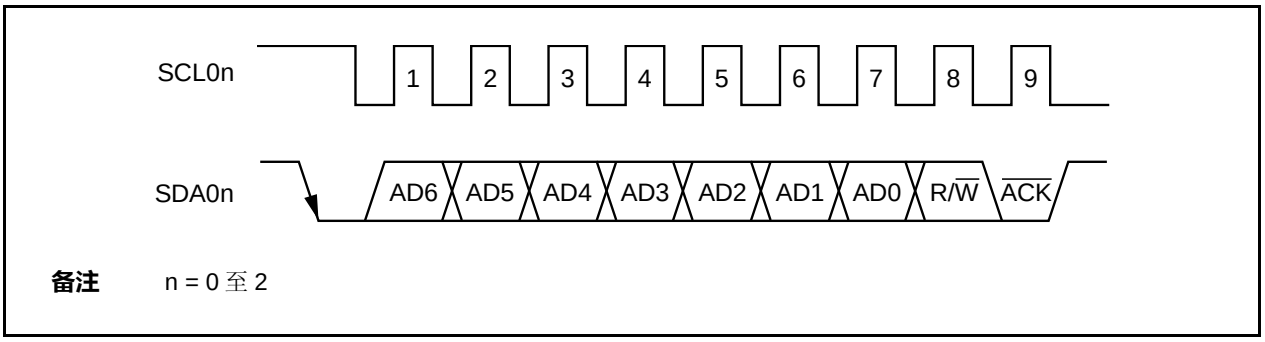
在第 9 个时钟期间，当接收设备将 SDA0n 信号线设置为低电平时，产生 $\overline{\text{ACK}}$ 信号（接收正常）。

当 IICCn.ACKEn 位设置为 1 时，使能自动产生 $\overline{\text{ACK}}$。发送 7 位地址之后的第 8 位会导致 IICSn.TRCn 位被置位。通常情况下，设置 ACKE0 位 为 1 作为接收的应答（TRCn 位 = 0）。

当从设备进行接收（TRCn 位 = 0）时，如果从设备不能接收数据或不再需要接收更多数据，将 ACKEn 位清除为 0 ，提示主设备，不能再接收更多数据。

与上述相似，当主设备进行接收（TRCn 位 = 0）时，并且不再需要后续数据，将 ACKEn 位清除为 0，来防止 $\overline{\text{ACK}}$ 信号的产生。这样可以通知从设备（发送设备）数据发送结束（发送停止）。

图 19-11. $\overline{\text{ACK}}$



接收到本地地址后，不管 ACKEn 位的设置值如何，都会自动产生 $\overline{\text{ACK}}$ 信号。如果收到的地址不是本地地址（NACK），不产生 $\overline{\text{ACK}}$ 信号。

接收到扩展码时，在产生 $\overline{\text{ACK}}$ 信号之前，将 ACKEn 位设置为 1。

数据接收期间 $\overline{\text{ACK}}$ 的产生方式取决于等待时序的设置，描述如下：

- 当选择 8 时钟等待时（IICCn.WTIMn 位 = 0）：
  如果等待状态取消前将 ACKEn 位设置为 1，$\overline{\text{ACK}}$ 信号在 SCL0n 引脚第 8 个时钟下降沿产生。
- 当选择 9 时钟等待时（IICCn.WTIMn 位 = 1）：
  如果 ACKEn 位被预先设置为 1，产生 $\overline{\text{ACK}}$ 信号。

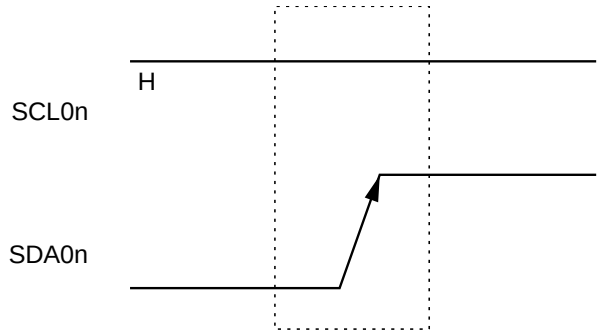
备注 n = 0 至 2

19.6.5 停止条件

当 SCL0n 引脚为高电平时， SDA0n 引脚由低电平变为高电平，即产生一个停止条件（n = 0 至 2）。

当主设备向从设备的串行传输完成之后，产生一个停止条件。被用作从设备时，可以检测开始条件。

图 19-12. 停止条件



备注 n = 0 至 2

当 IICCn.SPTn 位被设置为 1 时，发生一个停止条件。当检测到停止条件时，IICSn.SPDn 位被置为 1，且当 IICCn.SPIEn 位置为 1 时，产生一个中断请求信号（INTIICn）。（n = 0 至 2）

19.6.6 等待状态

等待状态用来通知通信伙伴，设备（主设备或从设备）已经准备就绪可以进行收发数据（即：处于等待状态）。

将 SCL0n 引脚设为低电平，通知通信伙伴本机已经处于等待状态。当主设备和从设备的等待状态都被取消时，下一次传输可以开始。（n = 0 至 2）

图 19-13. 等待状态（1/2）

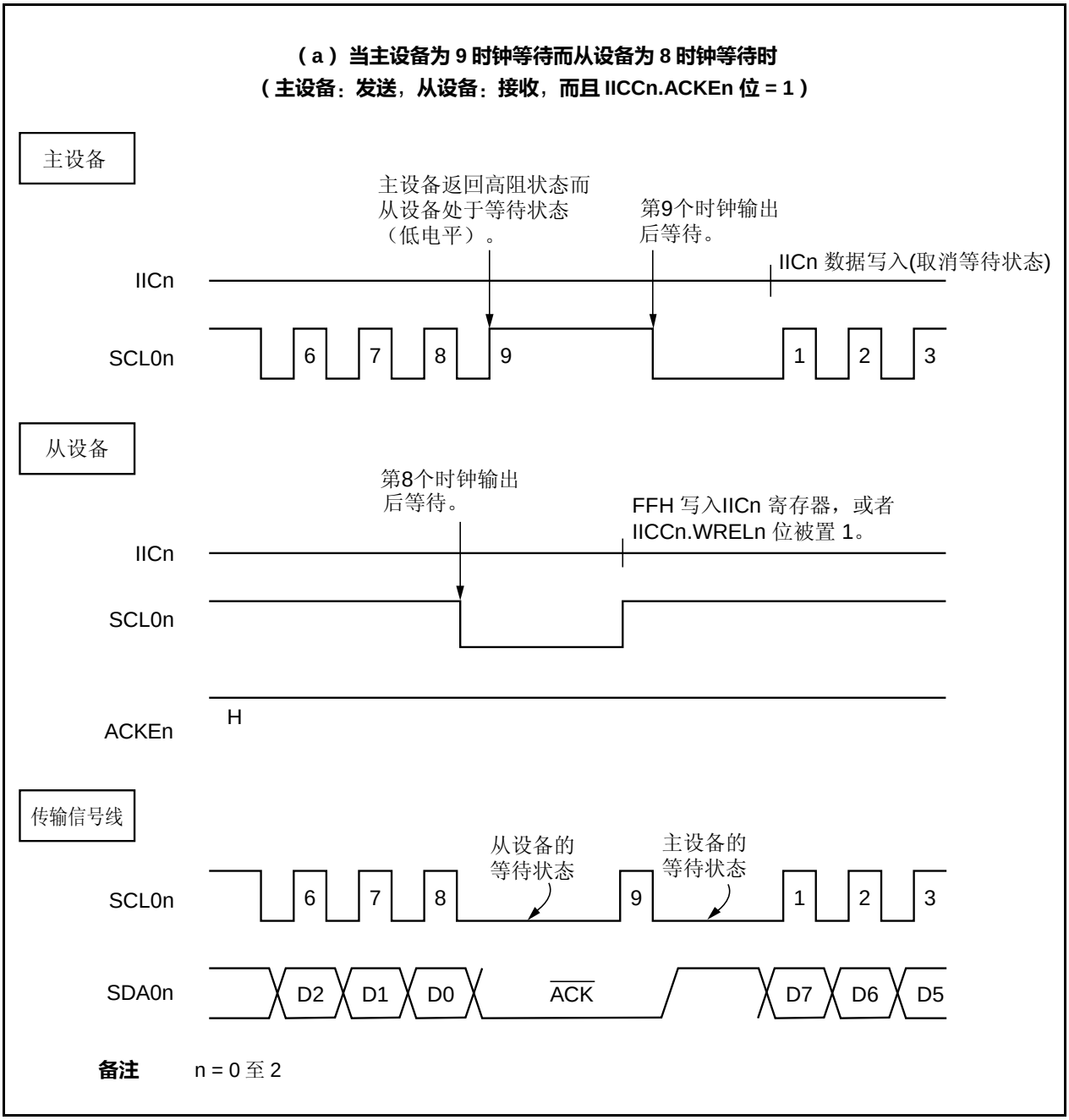
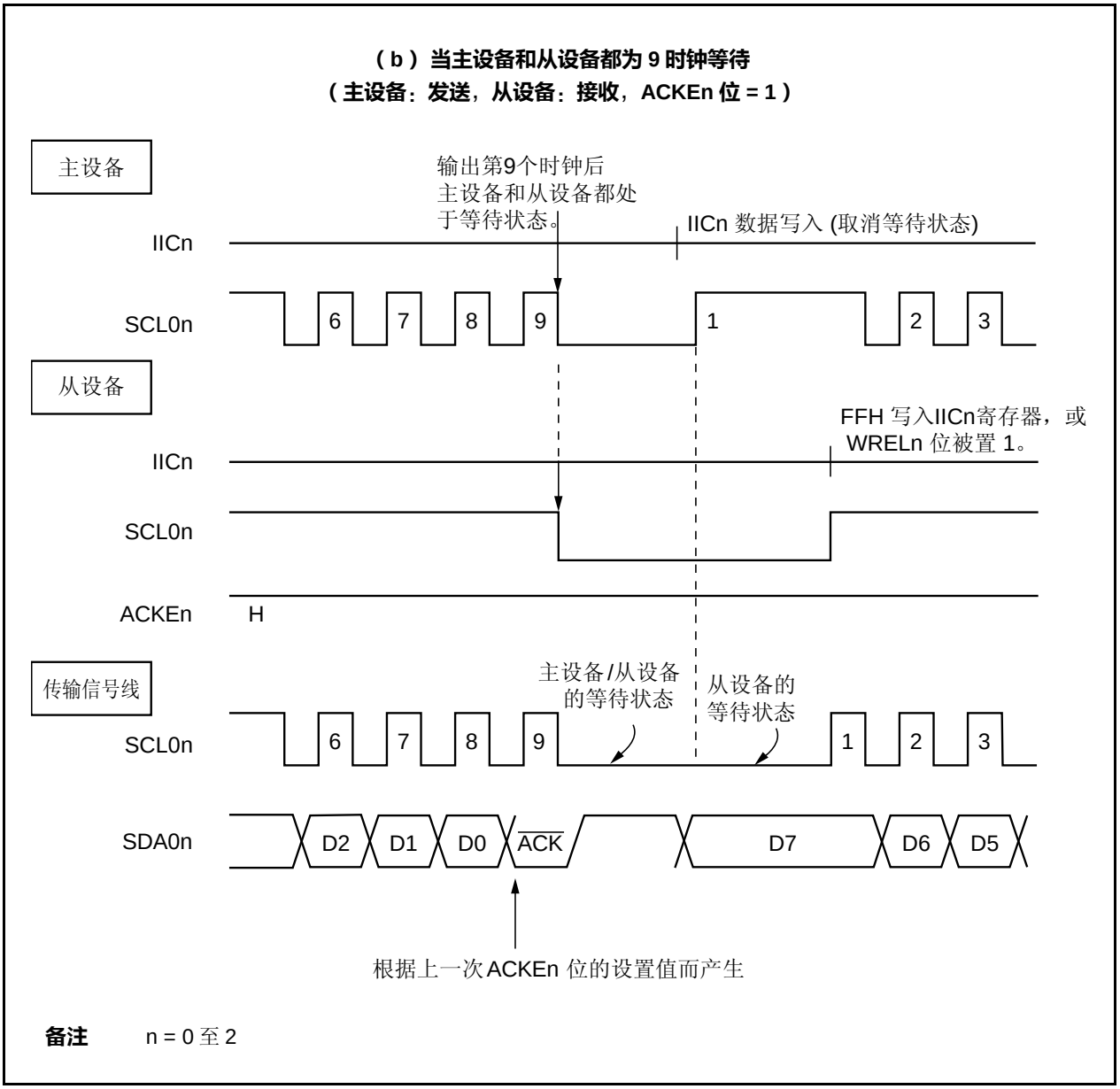


图 19-13. 等待状态（2/2）



备注 n = 0 至 2

根据 IICCn.WTIMn 位的设置，可能会自动产生等待状态。（n = 0 至 2）

通常情况下，当 IICCn.WRELn 位置为 1，或向接收端的 IICn 寄存器写入 FFH 时，取消等待状态，发送端向 IICn 寄存器写入数据，清除等待状态。

主设备还可以通过以下方式取消等待状态。

- 将 IICCn.STTn 位置为 1。
- 将 IICCn.SPTn 位置为 1。

19.6.7 等待状态取消方法

在使用 I²C0n 的情况下，一般使用以下方式取消等待状态（n = 0 至 2）。

- 向 IICn 寄存器写入数据。
- 将 IICCn.WRELn 位置为 1（取消等待状态）。
- 将 IICCn.STTn 位置为 1（产生开始条件）。
- 将 IICCn.SPTn 位置为 1（产生停止条件）。

如果执行任何一种等待取消动作，I²C0n 将取消等待状态，并重新开始通信。

当取消等待状态并发送数据（包括地址）时，向 IICn 寄存器写入数据。

为了在取消等待状态之后接收数据，或完成数据发送，将 WRELn 位置为 1。

为了在取消等待状态之后产生一个重新开始条件，将 STTn 位设置为 1。

为了在取消等待状态之后产生一个停止条件，将 SPTn 位设置为 1。

对每个等待状态只可执行一次取消操作。

例如，如果将 WRELn 位设置为 1，取消等待状态，然后将数据写入 IICn 寄存器，SDA0n 信号线的时序变化和 IICn 寄存器的写入时序可能发生冲突，可能导致输出到 SDA0n 信号线上的数据不正确。

即使在其它操作情况下，如果通信中途停止，将 IICCn.IICEn 位清除为 0 也会停止通信，使能等待状态的取消。

如果 I²C 总线因噪声等原因被死锁，将 IICCn.LRELn 位设置为 1，会停止通信操作，使能等待状态的取消。

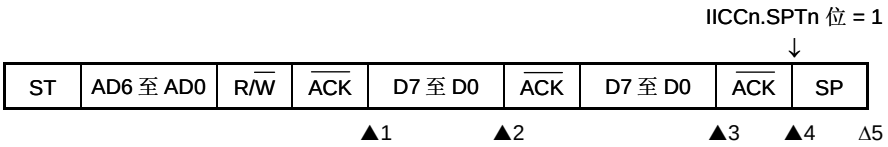
19.7 I²C中断请求信号（INTIICn）

以下展示了 INTIICn 中断请求信号产生时序和 INTIICn 信号时序处的 IICSn 寄存器的值。（n = 0 至 2）

19.7.1 主设备操作

（1） 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止（正常发送/接收）

<1> 当 IICCn.WTIMn 位 = 0 时



▲1： IICSn 寄存器 = 1000X110B

▲2： IICSn 寄存器 = 1000X000B

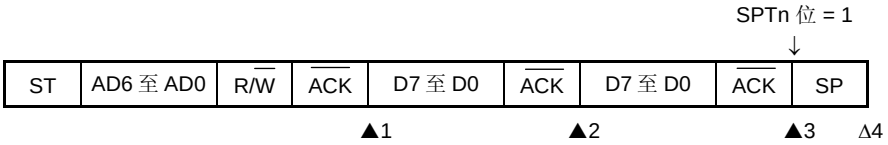
▲3： IICSn 寄存器 = 1000X000B（WTIMn 位 = 1）

▲4： IICSn 寄存器 = 1000XX00B

Δ 5： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1： IICSn 寄存器 = 1000X110B

▲2： IICSn 寄存器 = 1000X100B

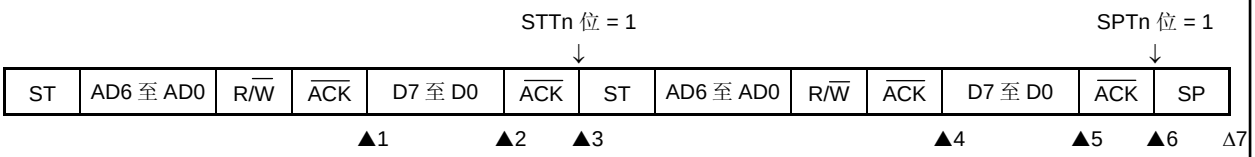
▲3： IICSn 寄存器 = 1000XX00B

Δ 4： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

（2） 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止（重新开始）

<1> 当 WTIMn 位 = 0 时



▲1： IICSn 寄存器 = 1000X110B

▲2： IICSn 寄存器 = 1000X000B（WTIMn 位 = 1）

▲3： IICSn 寄存器 = 1000XX00B（WTIMn 位 = 0）

▲4： IICSn 寄存器 = 1000X110B（WTIMn 位 = 0）

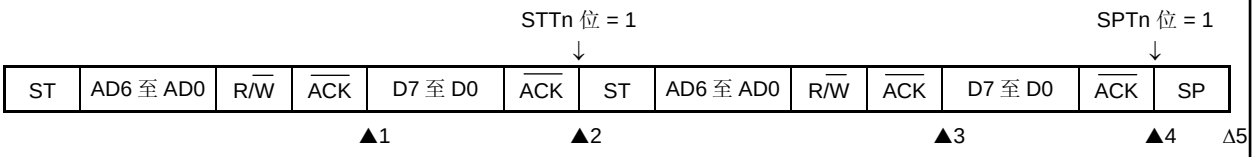
▲5： IICSn 寄存器 = 1000X000B（WTIMn 位 = 1）

▲6： IICSn 寄存器 = 1000XX00B

Δ 7： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1： IICSn 寄存器 = 1000X110B

▲2： IICSn 寄存器 = 1000XX00B

▲3： IICSn 寄存器 = 1000X110B

▲4： IICSn 寄存器 = 1000XX00B

Δ 5： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

（3） 开始 ~ 扩展码 ~ 数据 ~ 数据 ~ 停止（发送扩展码）

<1> 当 WTIMn 位 = 0 时

SPTn 位 = 1 ↓

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1： IICSn 寄存器 = 1010X110B

▲2： IICSn 寄存器 = 1010X000B

▲3： IICSn 寄存器 = 1010X000B（WTIMn 位 = 1）

▲4： IICSn 寄存器 = 1010XX00B

Δ 5： IICSn 寄存器 = 00000001B

备注 **1.** ▲： 总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

SPTn 位 = 1 ↓

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲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.** ▲： 总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

19.7.2 从设备操作（当接收从设备地址数据（地址匹配）时）

（1） 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止

<1> 当 IICCn.WTIMn 位 = 0 时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲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.** ▲：总是产生
Δ： 仅当 IICCn.SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲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.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

（2） 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0（重新开始后，地址匹配）时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
				▲1	▲2					▲3	▲4	Δ5

▲1： IICSn 寄存器 = 0001X110B

▲2： IICSn 寄存器 = 0001X000B

▲3： IICSn 寄存器 = 0001X110B

▲4： IICSn 寄存器 = 0001X000B

Δ 5： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1（重新开始后，地址匹配）时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
				▲1		▲2				▲3	▲4	Δ5

▲1： IICSn 寄存器 = 0001X110B

▲2： IICSn 寄存器 = 0001XX00B

▲3： IICSn 寄存器 = 0001X110B

▲4： IICSn 寄存器 = 0001XX00B

Δ 5： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

（3） 开始 ~ 地址 ~ 数据 ~ 开始 ~ 扩展码 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0（重新开始后，扩展码接收）时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
				▲1	▲2				▲3		▲4	Δ5

▲1： IICSn 寄存器 = 0001X110B

▲2： IICSn 寄存器 = 0001X000B

▲3： IICSn 寄存器 = 0010X010B

▲4： IICSn 寄存器 = 0010X000B

Δ 5： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1（重新开始后，扩展码接收）时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
				▲1		▲2			▲3	▲4	▲5	Δ6

▲1： IICSn 寄存器 = 0001X110B

▲2： IICSn 寄存器 = 0001XX00B

▲3： IICSn 寄存器 = 0010X010B

▲4： IICSn 寄存器 = 0010X110B

▲5： IICSn 寄存器 = 0010XX00B

Δ 6： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

（4） 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0（重新开始后， 地址不匹配（= 无扩展码））时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
				▲1	▲2					▲3		Δ4

▲1： IICSn 寄存器 = 0001X110B

▲2： IICSn 寄存器 = 0001X000B

▲3： IICSn 寄存器 = 00000X10B

Δ 4： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1（重新开始后，地址不匹配（= 无扩展码））时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
				▲1		▲2				▲3		Δ4

▲1： IICSn 寄存器 = 0001X110B

▲2： IICSn 寄存器 = 0001XX00B

▲3： IICSn 寄存器 = 00000X10B

Δ 4： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

19.7.3 从设备操作（当接收扩展码时）

（1） 开始 ~ 扩展码 ~ 数据 ~ 数据 ~ 停止

<1> 当 IICCn.WTIMn 位 = 0 时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1： IICSn 寄存器 = 0010X010B

▲2： IICSn 寄存器 = 0010X000B

▲3： IICSn 寄存器 = 0010X000B

Δ 4： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 IICCn.SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1： IICSn 寄存器 = 0010X010B

▲2： IICSn 寄存器 = 0010X110B

▲3： IICSn 寄存器 = 0010X100B

▲4： IICSn 寄存器 = 0010XX00B

Δ 5： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

（2） 开始 ~ 扩展码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0（重新开始后，地址匹配）时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
			▲1		▲2					▲3	▲4	Δ5

▲1： IICSn 寄存器 = 0010X010B

▲2： IICSn 寄存器 = 0010X000B

▲3： IICSn 寄存器 = 0001X110B

▲4： IICSn 寄存器 = 0001X000B

Δ 5： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1（重新开始后， 地址匹配）时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
			▲1	▲2		▲3				▲4	▲5	Δ6

▲1： IICSn 寄存器 = 0010X010B

▲2： IICSn 寄存器 = 0010X110B

▲3： IICSn 寄存器 = 0010XX00B

▲4： IICSn 寄存器 = 0001X110B

▲5： IICSn 寄存器 = 0001XX00B

Δ 6： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

（3） 开始 ~ 扩展码 ~ 数据 ~ 开始 ~ 扩展码 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0（重新开始后，扩展码接收）时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
			▲1		▲2				▲3		▲4	Δ5

▲1： IICSn 寄存器 = 0010X010B

▲2： IICSn 寄存器 = 0010X000B

▲3： IICSn 寄存器 = 0010X010B

▲4： IICSn 寄存器 = 0010X000B

Δ 5： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1（重新开始后，扩展码接收）时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
			▲1	▲2		▲3			▲4	▲5	▲6	Δ7

▲1： IICSn 寄存器 = 0010X010B

▲2： IICSn 寄存器 = 0010X110B

▲3： IICSn 寄存器 = 0010XX00B

▲4： IICSn 寄存器 = 0010X010B

▲5： IICSn 寄存器 = 0010X110B

▲6： IICSn 寄存器 = 0010XX00B

Δ 7： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

（4） 开始 ~ 扩展码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

<1> 当 WTIMn 位 = 0（重新开始后，地址不匹配（= 无扩展码））时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
			▲1		▲2					▲3		Δ4

▲1： IICSn 寄存器 = 0010X010B

▲2： IICSn 寄存器 = 0010X000B

▲3： IICSn 寄存器 = 00000X10B

Δ 4： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1（重新开始后， 地址不匹配（= 无扩展码））时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
			▲1	▲2		▲3				▲4		Δ5

▲1： IICSn 寄存器 = 0010X010B

▲2： IICSn 寄存器 = 0010X110B

▲3： IICSn 寄存器 = 0010XX00B

▲4： IICSn 寄存器 = 00000X10B

Δ 5： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

19.7.4 不通信时的操作

（1） 开始 ~ 扩展码 ~ 数据 ~ 数据 ~ 停止

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

Δ1

Δ 1： IICSn 寄存器 = 00000001B

备注 **1.** Δ： 仅当 SPIEn 位 = 1 时产生
2. n = 0 至 2

19.7.5 仲裁失败时的操作（仲裁失败后作为从设备操作）

（1） 当在从设备地址数据传输期间发生仲裁失败

<1> 当 IICCn.WTIMn 位 = 0 时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1： IICSn 寄存器 = 0101X110B（例如：中断服务期间读取 IICSn.ALDn 位时）

▲2： IICSn 寄存器 = 0001X000B

▲3： IICSn 寄存器 = 0001X000B

Δ 4： IICSn 寄存器 = 00000001B

备注 **1.** ▲： 总是产生
Δ： 仅当 IICCn.SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1： IICSn 寄存器 = 0101X110B（例如：中断服务期间读取 ALDn 位时）

▲2： IICSn 寄存器 = 0001X100B

▲3： IICSn 寄存器 = 0001XX00B

Δ 4： IICSn 寄存器 = 00000001B

备注 **1.** ▲： 总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

（2） 在扩展码的传输期间发生仲裁失败

<1> 当 WTIMn 位 = 0 时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1： IICSn 寄存器 = 0110X010B（例如：中断服务期间读取 ALDn 位时）

▲2： IICSn 寄存器 = 0010X000B

▲3： IICSn 寄存器 = 0010X000B

Δ 4： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1： IICSn 寄存器 = 0110X010B（例如：中断服务期间读取 ALDn 位时）

▲2： IICSn 寄存器 = 0010X110B

▲3： IICSn 寄存器 = 0010X100B

▲4： IICSn 寄存器 = 0010XX00B

Δ 5： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

19.7.6 发生仲裁失败时的操作（在仲裁失败后不进行通信）

（1） 在从设备地址数据传输期间发生仲裁失败

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1　　Δ2

▲1： IICSn 寄存器 = 01000110B（例如：中断服务期间读取 IICSn.ALDn 位时）

Δ 2： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 IICCn.SPIEn 位 = 1 时产生
2. n = 0 至 2

（2） 在扩展码的传输期间发生仲裁失败

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1　　Δ2

▲1：IICSn 寄存器 = 0110X010B（例如：中断服务期间读取 ALDn 位时）

由软件设置 IICCn.LRELn 位。

Δ 2：IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

（3） 在数据传输期间发生仲裁失败

<1> 当 IICCn.WTIMn 位 = 0 时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1 ▲2 Δ3

▲1： IICSn 寄存器 = 10001110B

▲2： IICSn 寄存器 = 01000000B（例如：中断服务期间读取 ALDn 位时）

Δ 3： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP

▲1 ▲2 Δ3

▲1： IICSn 寄存器 = 10001110B

▲2： IICSn 寄存器 = 01000100B（例如：中断服务期间读取 ALDn 位时）

Δ 3： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
2. n = 0 至 2

（4） 在数据传输期间，因为重新开始条件导致发生仲裁失败

<1> 无扩展码（例如： 地址不匹配）

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 Dn	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
				▲1					▲2		Δ3

▲1： IICSn 寄存器 = 1000X110B

▲2： IICSn 寄存器 = 01000110B（例如：中断服务期间读取 ALDn 位时）

Δ 3： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. Dn = D6 至 D0
n = 0 至 2

<2> 扩展代码

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 Dn	ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
				▲1				▲2			Δ3

▲1： IICSn 寄存器 = 1000X110B

▲2： IICSn 寄存器 = 0110X010B（例如：中断服务期间读取 ALDn 位时）

由软件设置 IICCn.LRELn 位。

Δ 3： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. Dn = D6 至 D0
n = 0 至 2

（5） 在数据传输期间，因为停止条件导致发生仲裁失败

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 Dn	SP

▲1 Δ2

▲1： IICSn 寄存器 = 1000X110B

Δ 2： IICSn 寄存器 = 01000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. Dn = D6 至 D0
n = 0 至 2

（6） 当试图产生重新开始条件时，因为 SDA0n 引脚的低电平导致发生仲裁失败

<1> 当 WTIMn 位 = 0 时

IICCn.STTn 位 = 1 ↓

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
				▲1	▲2	▲3	▲4			Δ5

▲1： IICSn 寄存器 = 1000X110B

▲2： IICSn 寄存器 = 1000X000B（WTIMn 位 = 1）

▲3： IICSn 寄存器 = 1000XX00B（WTIMn 位 = 0）

▲4： IICSn 寄存器 = 01000000B（例如：中断服务期间读取 ALDn 位时）

Δ 5： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

IICCn.STTn 位 = 1 ↓

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
				▲1		▲2		▲3		Δ4

▲1： IICSn 寄存器 = 1000X110B

▲2： IICSn 寄存器 = 1000XX00B

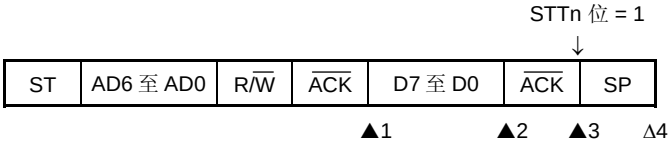
▲3： IICSn 寄存器 = 01000100B（例如：中断服务期间读取 ALDn 位时）

Δ 4： IICSn 寄存器 = 00000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

（7） 当试图产生重新开始条件时，由于停止条件导致发生仲裁失败

<1> 当 WTIMn 位 = 0 时



▲1： IICSn 寄存器 = 1000X110B

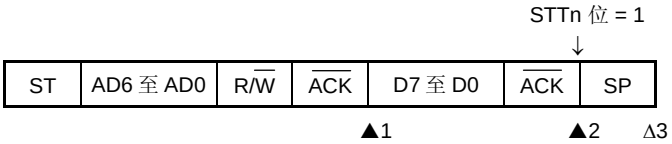
▲2： IICSn 寄存器 = 1000X000B（WTIMn 位 = 1）

▲3： IICSn 寄存器 = 1000XX00B

Δ 4： IICSn 寄存器 = 01000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时



▲1： IICSn 寄存器 = 1000X110B

▲2： IICSn 寄存器 = 1000XX00B

Δ 3： IICSn 寄存器 = 01000001B

备注 **1.** ▲：总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

（8） 当试图产生停止条件时，由于 SDA0n 引脚为低电平导致发生仲裁失败

<1> 当 WTIMn 位 = 0 时

IICCn.SPTn 位 = 1
↓

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
				▲1	▲2	▲3	▲4			Δ5

▲1： IICSn 寄存器 = 1000X110B

▲2： IICSn 寄存器 = 1000X000B（WTIMn 位 = 1）

▲3： IICSn 寄存器 = 1000XX00B（WTIMn 位 = 0）

▲4： IICSn 寄存器 = 01000000B（例如：中断服务期间读取 ALDn 位时）

Δ 5： IICSn 寄存器 = 00000001B

备注 **1.** ▲： 总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

<2> 当 WTIMn 位 = 1 时

IICCn.SPTn 位 = 1
↓

ST	AD6 至 AD0	R/$\overline{W}$	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	D7 至 D0	$\overline{ACK}$	SP
				▲1		▲2		▲3		Δ4

▲1： IICSn 寄存器 = 1000X110B

▲2： IICSn 寄存器 = 1000XX00B

▲3： IICSn 寄存器 = 01000000B（例如：中断服务期间读取 ALDn 位时）

Δ 4： IICSn 寄存器 = 00000001B

备注 **1.** ▲： 总是产生
Δ： 仅当 SPIEn 位 = 1 时产生
X： 无需理会
2. n = 0 至 2

19.8 中断请求信号（INTIICn）的产生时序和等待控制

IICCn.WTIMn 位的设置可以决定 INTIICn 寄存器的产生时序和对应的等待控制，如下所示。（n = 0 至 2）

表 19-3. INTIICn 产生时序和等待控制

WTIMn 位	从设备操作期间			主设备操作期间		
	地址	数据接收	数据发送	地址	数据接收	数据发送
0	9 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.** 仅当接收的地址和 SVAn 寄存器设定的值相匹配时，在第 9 个时钟的下降沿产生从设备的 INTIICn 信号和等待时期。
此刻，不论 IICCn.ACKEn 位如何设置，都产生 $\overline{\text{ACK}}$ 信号，对于已经接收到扩展码的从设备，INTIICn 信号在第 8 个时钟的下降沿发生。
当重新开始后地址不匹配时，INTIICn 信号在第 9 个时钟的下降沿发生，但是，不发生等待。
2. 如果接收地址和 SVAn 寄存器的内容不匹配，而且不能收到扩展码，则既不产生 INTIICn 信号，也不产生等待。

备注 **1.** 表中的数量表示串行时钟信号的时钟数目，中断请求信号和等待控制都和这些时钟信号的下降沿同步。
2. n = 0 至 2

（1） 地址发送/接收期间

- 从设备操作： 根据上述的注 1 和注 2 的条件，决定中断和等待时序，无需考虑 WTIMn 位的设置。
- 主设备操作： 不论 WTIMn 位如何设置，中断请求和等待时序都在第 9 个时钟的下降沿发生。

（2） 在数据接收期间

- 主设备/从设备操作： 中断请求和等待时序由 WTIMn 位的设置决定。

（3） 数据发送期间

- 主设备/从设备操作： 中断请求和等待时序由 WTIMn 位的设置决定。

（4） 等待取消方法

有四种等待取消方法可供使用，如下所示：

- 将 IICCn.WRELn 位置为 1
- 向 IICn 寄存器写入数据
- 设置开始条件（IICCn.STTn 位= 1）注
- 设置停止条件（IICCn.SPTn 位= 1）注

注 仅限主设备。

当已经选择 8 时钟等待时（WTIMn 位 = 0），在取消等待之前必须确定是否已经产生 $\overline{\text{ACK}}$ 信号。

备注 n = 0 至 2

（5） 停止条件检测

当检测到停止条件时，会产生 INTIICn 信号。

备注 n = 0 至 2

19.9 地址匹配检测方法

在 I²C 总线模式下，通过发送从设备的地址，主设备可以选择特定的从设备。

地址匹配的检测由硬件自动执行。当本地地址被置入 SVAn 寄存器以后，且置入 SVAn 寄存器的本地地址和主设备发送的从设备地址匹配时，或接收到扩展码时，产生一个 INTIICn 信号。（n = 0 至 2）

19.10 错误检测

在 I²C 总线模式下，串行数据总线引脚（SDA0n）在数据传送期间的状态被发送设备的 IICn 寄存器捕获，于是，发送之前 IICn 寄存器内的数据就可以和已发送的 IICn 数据进行比较，从而能够检测传输错误。当比较值不匹配时，判断为发生传输错误。（n = 0 至 2）

19.11 扩展代码

（1）当接收地址的高 4 位是 0000 或 1111 时，扩展码标志（IICSn.EXCn 位）被置位，表示扩展码接收，且在第 8 个时钟的下降沿处产生一个中断请求信号（INTIICn）。（n = 0 至 2）
存储在 SVAn 寄存器中的本地地址不受影响。

（2）如果通过 10 位地址传输将 11110xx0 设置到 SVAn 寄存器，且 11110xx0 数据由主设备发出，其结果如下所示。注意， INTIICn 信号在第 8 个时钟的下降沿处产生（n = 0 至 2）。

- 数据匹配的高 4 位： EXCn 位 = 1
- 数据匹配的第 7 位： IICSn.COIn 位 = 1

（3）根据扩展码之后的数据，中断请求信号发生后的处理也各不相同，这些处理由软件执行。
例如，接收到扩展码以后，当不希望作为从设备操作时，将 IICCn.LRELn 位设置为 1，CPU 将进入下一次通信等待状态。

表 19-4. 扩展码位的定义

从设备地址	R/$\overline{W}$ 位	描述
0000 000	0	通用呼叫地址
0000 000	1	开始字节
0000 001	X	CBUS 地址
0000 010	X	为不同总线格式保留的地址
1111 0xx	X	10 位从设备地址规范

19.12 仲裁

当多个主设备同时产生开始条件时（在 IICSn.STDn 位被置 1 之前，IICCn.STTn 位已被置为 1），调整时钟数目，直到数据不同，多个主设备之间的通信才能执行，这种操作被称为仲裁（n = 0 至 2）。

当一个主设备在仲裁中失败时，在仲裁失败发生时序处将仲裁失败标志（IICSn.ALDn 位）设置为 1，且 SCL0n 信号线和 SDA0n 信号线都被设置为高阻状态，总线被释放。（n = 0 至 2）

仲裁失败的检测基于下次中断请求信号（INTIICn）的时序（当检测到停止条件时，第 8 或第 9 个时钟）和通过软件对 ALDn 位置 1 的设置（n = 0 至 2）。

关于中断请求信号时序的详情， 参见 **19.7 I²C 中断请求信号（INTIICn）**。

图 19-14. 仲裁时序示例

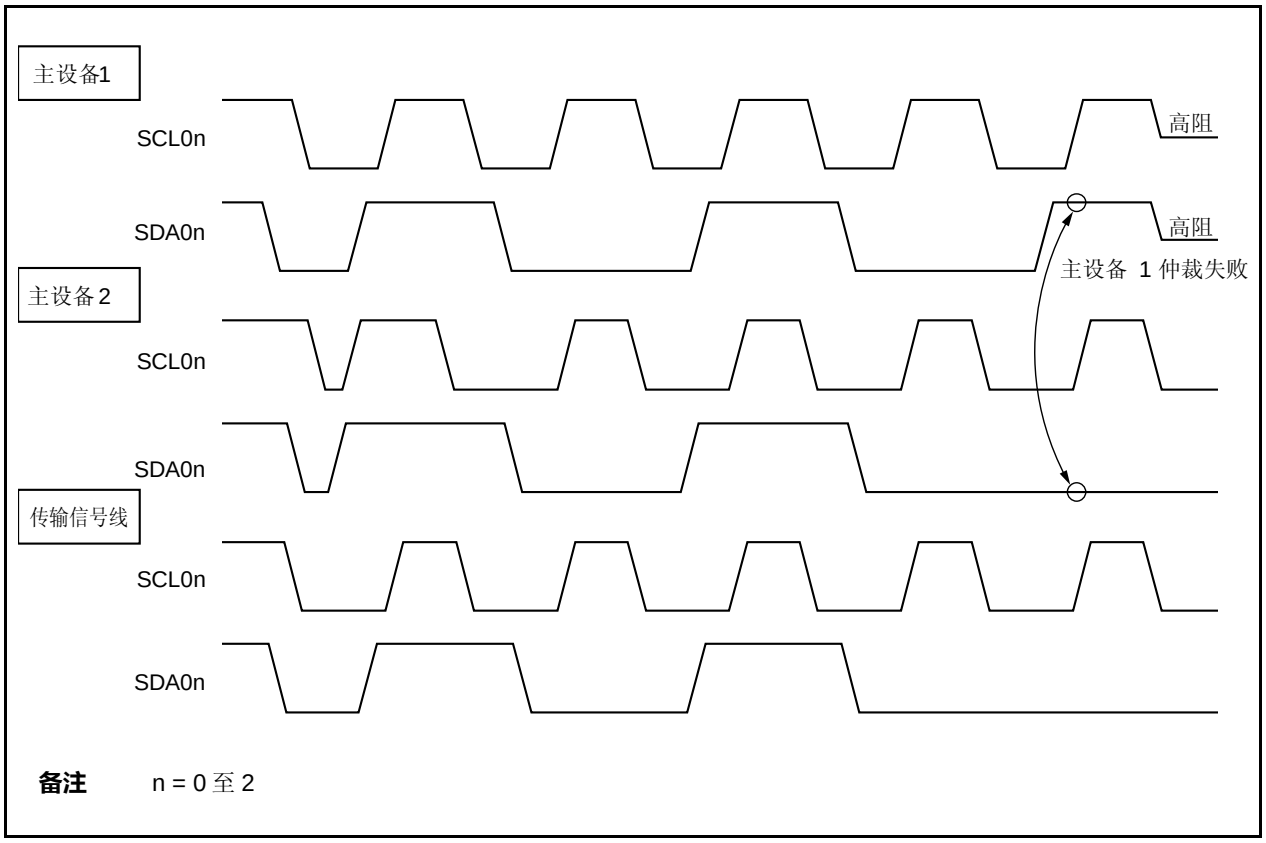


表 19-5. 仲裁期间的状态和中断请求信号发生时序

仲裁期间的状态	中断请求发生时序
发送地址传输	字节传输之后的第 8 个或第 9 个时钟的下降沿[注 1]
地址传输后读出/写入数据	
发送扩展代码	
扩展码传输后读出/写入数据	
发送数据	
数据接收后 $\overline{\text{ACK}}$ 传送期间	
在数据传输期间检测到重新开始条件	
在数据传输期间检测到停止条件	当产生停止条件时（当 IICCn.SPIEn 位 = 1 时）[注 2]
试图产生重新开始条件时 SDA0n 引脚是低电平	字节传输之后的第 8 个或第 9 个时钟的下降沿[注 1]
试图产生重新开始条件时检测到停止条件	当产生停止条件时（当 IICCn.SPIEn 位 = 1 时）[注 2]
试图产生停止条件时 DSA0n 引脚是低电平	字节传输之后的第 8 个或第 9 个时钟的下降沿[注 1]
试图产生重新开始条件时 SCL0n 引脚是低电平	

注 **1.** 当 IICCn.WTIMn 位 = 1 时，INTIICn 信号在第 9 个时钟的下降沿发生。当 WTIMn 位 = 0 且接收到扩展码的从地址时，INTIICn 信号在第 8 个时钟的下降沿发生（n = 0 至 2）。

2. 当有可能发生仲裁时，在主设备操作要设置 SPIEn 位为 1（n = 0 至 2）。

19.13 唤醒功能

已经接收到本地地址和扩展码时，I²C 总线从设备的功能是产生一个中断请求信号（INTIICn）。

当地址数据不匹配时，该功能就会阻止不必要的 INTIICn 信号，从而使处理更有效率。

当检测到开始条件后，待机模式被唤醒。仲裁失败可能将主设备（已经产生开始条件）变为从设备，在这种可能情况下发送地址时，待机模式唤醒有效。

但是，当检测到停止条件后，不论唤醒功能如何设置，IICCn.SPIEn 位总是置位，这可以决定 INTIICn 信号的使能或禁止（n = 0 至 2）。

19.14 通信保留

19.14.1 使能通信保留功能（IICFn.IICRSVn 位 = 0）时

在 V850ES/JG3-L 目前未使用总线时，开始主设备通信，通信保留功能可以在总线释放后使能开始条件的传输。在下列两种模式下，V850ES/JG3-L 不使用总线。

- 当 V850ES/JG3-L 的仲裁结果既不是主设备操作，又不是从设备操作。
- 当接收到扩展码后且从设备操作被禁止时（当 IICCn.LRELn 位被置为 1 时，不返回 $\overline{\text{ACK}}$ 信号且总线释放）（n = 0 至 2）。

如果在 V850ES/JG3-L 未使用总线时，将 IICCn.STTn 位置 1，将自动产生一个开始条件，并且在总线释放后（检测到停止条件后）设置等待状态。

检测到总线释放（检测到停止条件时）时，写入 IICn 寄存器导致开始传送主设备地址。此时，IICCn.SPIEn 位应该被设置为 1（n = 0 至 2）。

当 STTn 位被设置为 1 后，根据总线状态决定工作模式（开始条件或通信保留）（n = 0 至 2）。

如果总线已经被释放 .....产生开始条件
如果总线未被释放（待机模式） .....通信保留

要检测为 STTn 位决定何种工作模式，将 STTn 位设置为 1，待等待时期完毕后，检查 IICSn.MSTSn 位（n = 0 至 2）。

应该由软件设置的等待期间在表 19-6 中列出。

这些等待时期可以通过 IICCLn 寄存器的 SMCn，CLn1 和 CLn0 位以及 IICXn.CLXn 位进行设置（n = 0 至 2）。

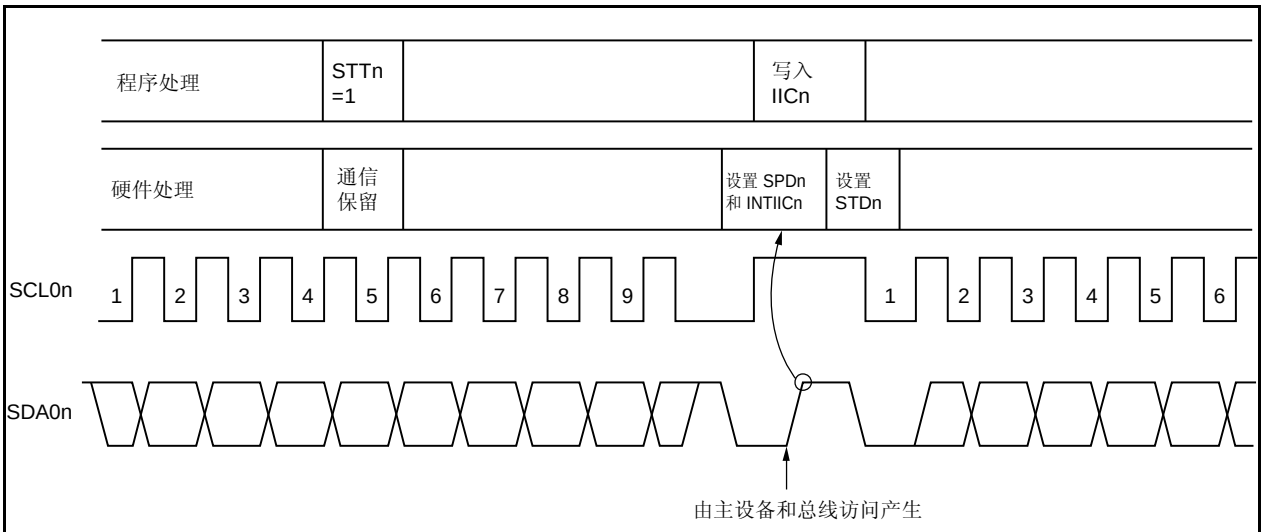
表 19-6. 等待时期

时钟选择	CLXn	SMCn	CLn1	CLn0	等待期
f_{XX} (当 OCKSm = 18H 时)	0	0	0	0	26 个时钟
$f_{XX}/2$ (当 OCKSm = 10H 时)	0	0	0	0	52 个时钟
$f_{XX}/3$ (当 OCKSm = 11H 时)	0	0	0	0	78 个时钟
$f_{XX}/4$ (当 OCKSm = 12H 时)	0	0	0	0	104 个时钟
$f_{XX}/5$ (当 OCKSm = 13H 时)	0	0	0	0	130 个时钟
f_{XX} (当 OCKSm = 18H 时)	0	0	0	1	47 个时钟
$f_{XX}/2$ (当 OCKSm = 10H 时)	0	0	0	1	94 个时钟
$f_{XX}/3$ (当 OCKSm = 11H 时)	0	0	0	1	141 个时钟
$f_{XX}/4$ (当 OCKSm = 12H 时)	0	0	0	1	188 个时钟
f_{XX}	0	0	1	0	47 个时钟
f_{XX} (当 OCKSm = 18H 时)	0	0	1	1	37 个时钟
$f_{XX}/2$ (当 OCKSm = 10H 时)	0	0	1	1	74 个时钟
$f_{XX}/3$ (当 OCKSm = 11H 时)	0	0	1	1	111 个时钟
f_{XX} (当 OCKSm = 18H 时)	0	1	0	×	16 个时钟
$f_{XX}/2$ (当 OCKSm = 10H 时)	0	1	0	×	32 个时钟
$f_{XX}/3$ (当 OCKSm = 11H 时)	0	1	0	×	48 个时钟
$f_{XX}/4$ (当 OCKSm = 12H 时)	0	1	0	×	64 个时钟
f_{XX}	0	1	1	0	16 个时钟
f_{XX} (当 OCKSm = 18H 时)	0	1	1	1	13 个时钟
$f_{XX}/2$ (当 OCKSm = 10H 时)	0	1	1	1	26 个时钟
$f_{XX}/3$ (当 OCKSm = 11H 时)	0	1	1	1	39 个时钟
f_{XX} (当 OCKSm = 18H 时)	1	1	0	×	10 个时钟
$f_{XX}/2$ (当 OCKSm = 10H 时)	1	1	0	×	20 个时钟
$f_{XX}/3$ (当 OCKSm = 11H 时)	1	1	0	×	30 个时钟
$f_{XX}/4$ (当 OCKSm = 12H 时)	1	1	0	×	40 个时钟
$f_{XX}/5$ (当 OCKSm = 13H 时)	1	1	0	×	50 个时钟
f_{XX}	1	1	1	0	10 个时钟

备注 **1.** n = 0 至 2
m = 0，1
2. × = 无需理会

通信保留时序如下所示。

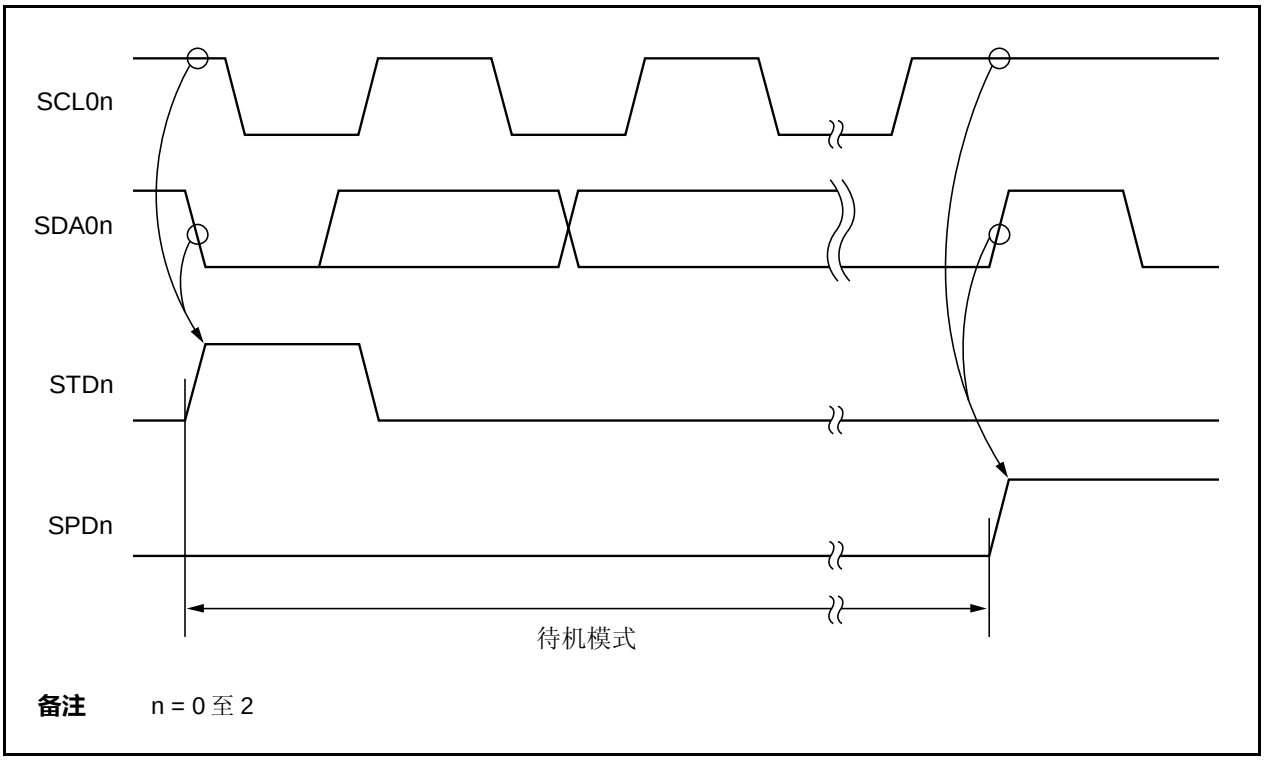
图 19-15. 通信保留时序



备注 n = 0 至 2
STTn： IICCn 寄存器的位
STDn： IICSn 寄存器的位
SPDn： IICSn 寄存器的位

通过以下时序受理通信保留。IICSn.STDn 位设置为 1 后，在检测到停止条件前，设置 IICCn.STTn 位为 1 就可以产生一次通信保留（n = 0 至 2）。

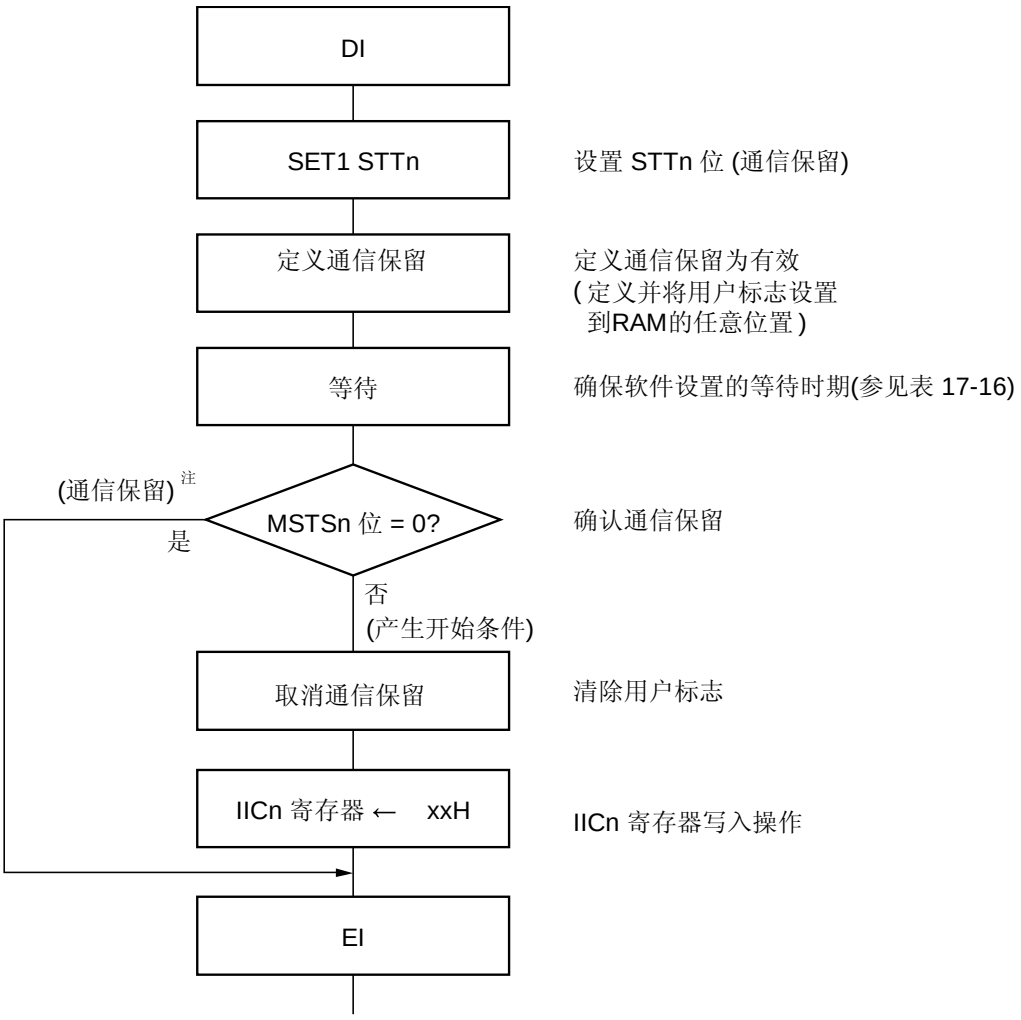
图 19-16. 受理通信保留的时序



备注 n = 0 至 2

通信保留的流程图展示如下。

图 19-17. 通信保留的流程图



注 当通信保留时，发生停止条件中断请求，写入到 IICn 寄存器。

备注 n = 0 至 2

19.14.2 禁止通信保留功能时（IICFn.IICRSVn 位 = 1）

在 V850ES/JG3-L 未使用总线时，当 IICCn.STTn 位被置位时，通信保留请求被拒绝，且不会产生开始条件。在下列两种模式下，V850ES/JG3-L 不使用总线。

- 当 V850ES/JG3-L 的仲裁结果既不是主设备操作，又不是从设备操作。
- 当接收到扩展码后且从设备操作被禁止时（当 IICCn.LRELn 位被置为 1 时，不返回 $\overline{\text{ACK}}$ 信号且总线释放）（n = 0 至 2）。

为了确认是产生了开始条件，还是请求遭到拒绝，检查 IICFn.STCFn 标志。在 STTn 位被设置为 1 后，直到 STCFn 标志被置位，需要的时间在表 19-7 中列出。因而，通过软件来保证这段需要的时间。

表 19-7. 等待时期

OCKSENm	OCKSm1	OCKSm0	CLn1	CLn0	等待时期
1	0	0	0	×	10 个时钟
1	0	1	0	×	15 个时钟
1	1	0	0	×	20 个时钟
1	1	1	0	×	25 个时钟
0	0	0	1	0	5 个时钟

备注 **1.** ×： 无需理会
2. n = 0 至 2
m = 0，1

19.15 注意事项

（1）当 IICFn.STCENn 位 = 0 时

I²C0n 操作使能之后，不管总线的实际状态如何，总是立即认为处于总线通信状态（IICFn.IICBSYn 位 = 1）。要在尚未检测到停止条件的情况下执行主设备通信，先产生一个停止条件，在开始主设备通信之前释放总线。使用以下次序来产生停止条件。

<1> 设置 IICCLn 寄存器。
<2> 设置 IICCn.IICEn 位。
<3> 设置 IICCn.SPTn 位。

（2）当 IICFn.STCENn 位 = 1 时

I²C0n 操作使能之后，不管总线的实际状态如何，总是立即认为处于总线释放状态（IICFn.IICBSYn 位 = 0）。若要产生首个开始条件（IICCn.STTn 位 = 1），必需确认总线已经被释放，以免干扰其它通信。

（3）在和其它设备进行通信时，当 V850ES/JG3-L 的 IICCn.IICEn 位被设置为 1，根据通信线的状态，可能会检测到开始条件。当 SCL0n 和 SDA0n 信号线是高电平时，请确保将 IICCn.IICEn 位设置为 1。

（4）使能操作（IICCn.IICEn 位 = 1）之前，通过 IICCLn ，IICXn 和 OCKSm 寄存器决定工作时钟频率。要改变操作时钟频率，要将 IICCn.IICEn 位清 0 一次。

（5）当 IICCn.STTn 和 IICCn.SPTn 位已经被设置为 1，若不首先将其清 0，则一定不要对其再次置位。

（6）如果传输已经被保留，设置 IICCN.SPIEn 位为 1，这样，检测到停止条件就会产生一个中断请求信号。当产生一个中断请求信号后，向 I²Cn 写入通信数据将会释放等待状态，然后传输开始。如果检测到停止条件后不产生中断请求信号，由于未产生中断请求，传输将暂停在等待状态。但是，没有必要为了软件对 IICSn.MSTSn 位的检测而将 SPIEn 位置为 1 。

备注 n = 0 至 2
m = 0，1

19.16 通信操作

下面通过流程图展示 3 种操作过程。

（1）主设备工作于单主系统

当使用 V850ES/JG3-L 单片机作为单主系统中的主设备时，流程图如下所示。

流程图大体可以分为初始设置和通信处理两部分。在启动时执行初始设置。如果需要和从设备进行通信，进行通信准备后，执行通信处理。

（2）主设备工作于多主系统

在 I^2C0n 总线多主系统中，当总线参与通信之后，不能由 I^2C 总线规范判定总线是释放还是被使用。这里，当数据线和时钟线都在一定的时期（1 帧）内保持为高电平，V850ES/JG3-L 就认为总线已释放并可以进行通信。

操作流程可以大体分为初始设置、通信等待和通信处理。V850ES/JG3-L 在仲裁中失败并被指定为从设备的处理情况这里予以省略，而只是展示作为主设备的处理操作。启动时执行初始设置以便参与通信。然后，作为主设备等待通信请求或作为从设备等待指定。在通信处理中执行实际的通信，且支持与从设备的发送/接收操作和与其它主设备间的仲裁处理。

（3）从设备操作

当 V850ES/JG3-L 用作 I^2C0n 总线的从设备时的示例如下。

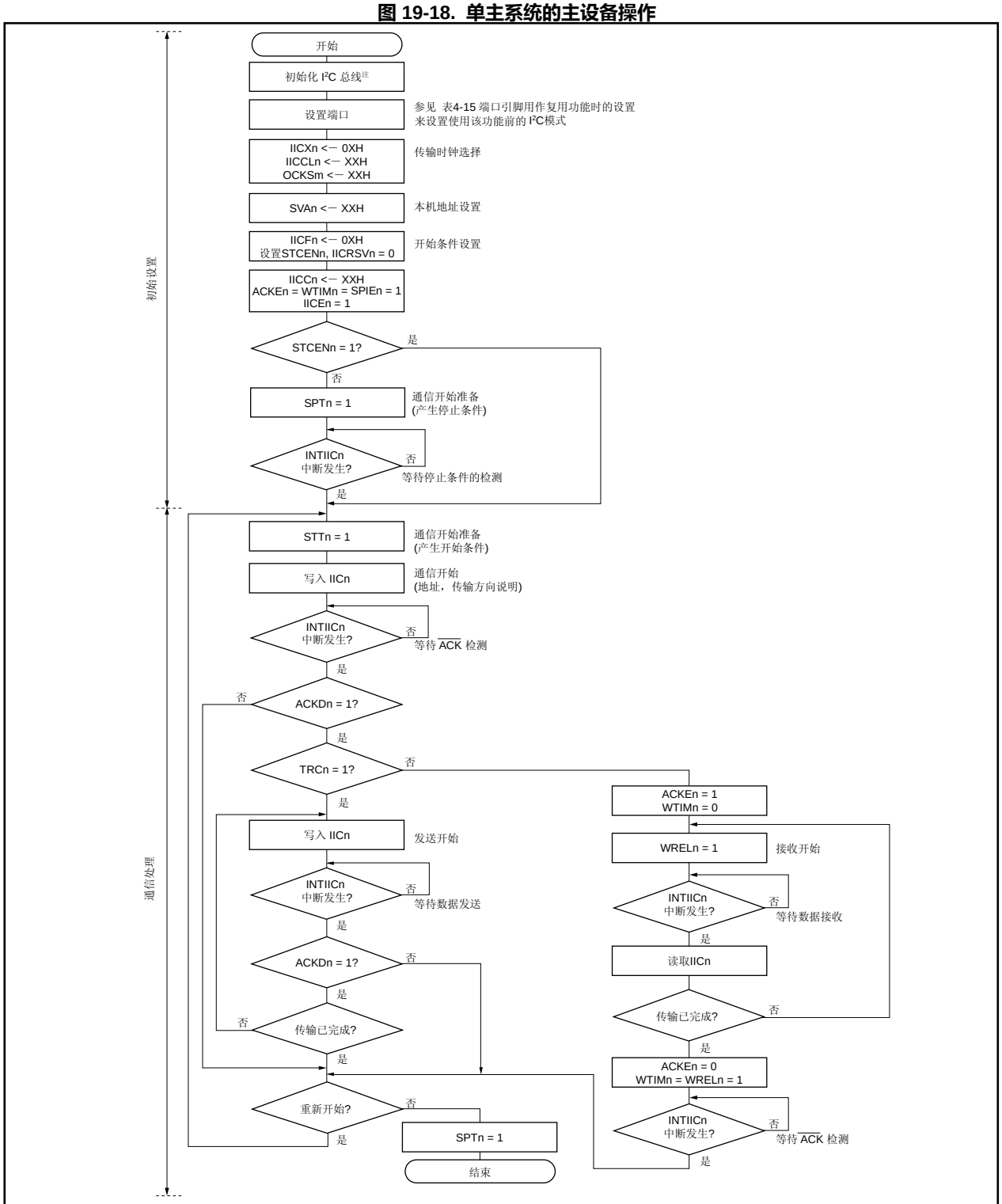
当用作从设备时，由一个中断启动操作。在启动时执行初始设置，然后，等待 INTIICn 中断信号的发生（通信等待）。当 INTIICn 中断发生时，通信状态被判定，并且判定结果作为一个标志传送给主处理程序。

通过检查该标志，可以执行必需的通信处理。

备注 n = 0 至 2

19.16.1 主设备工作于单主系统

图 19-18. 单主系统的主设备操作

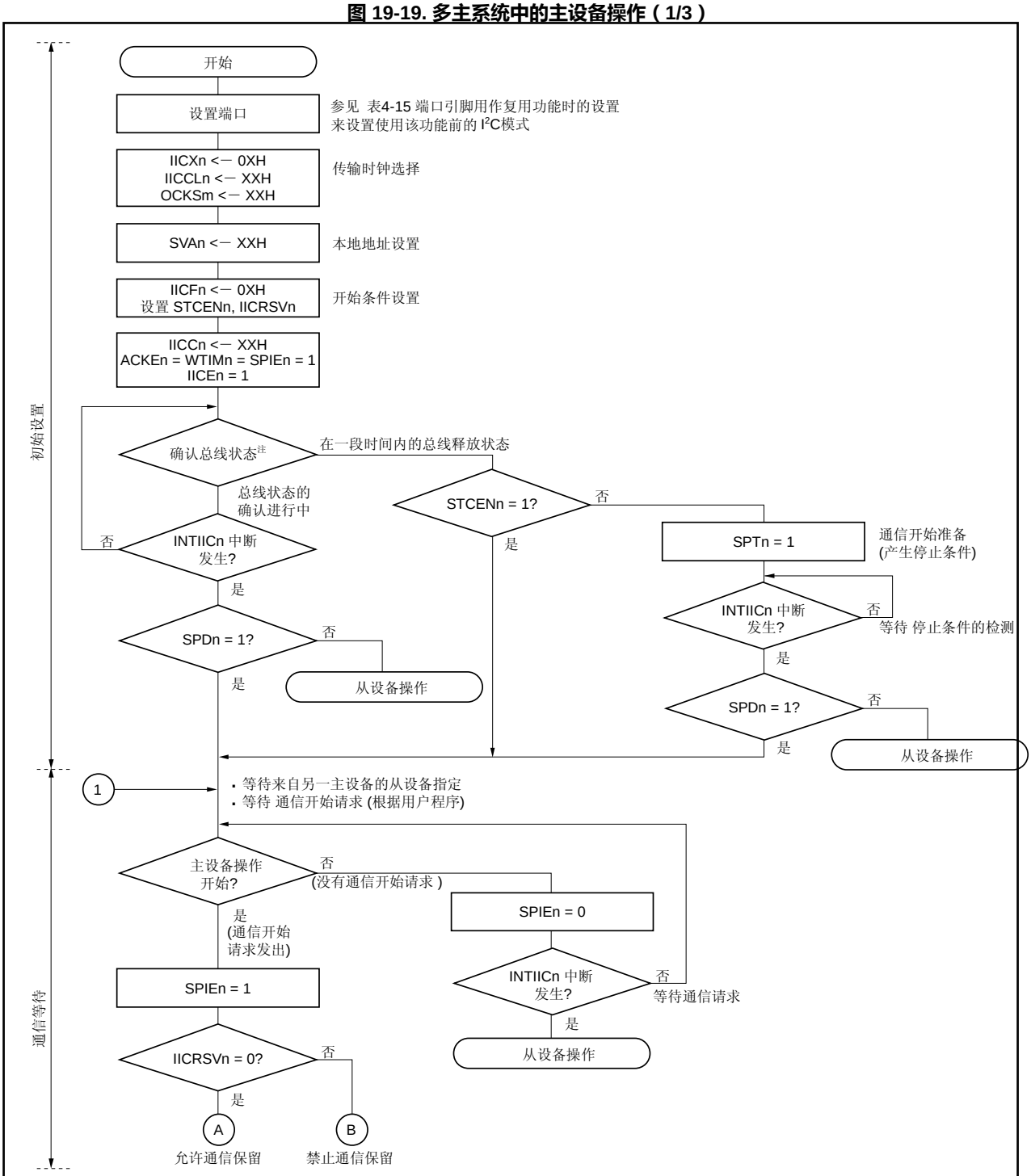


注 释放 I²C0n 总线（SCL0n，SDA0n 引脚 = 高电平），与通信中使用产品的规约一致。
例如，当 EEPROM™ 输出低电平到 SDA0n 引脚时，设置 SCL0n 引脚为输出端口并从中输出时钟脉冲，直到 SDA0n 引脚为稳定的高电平。

备注 **1.** 至于发送格式和接收格式，符合通信中使用产品的规约。
2. n = 0 至 2， m = 0，1

19.16.2 主设备工作于多主系统

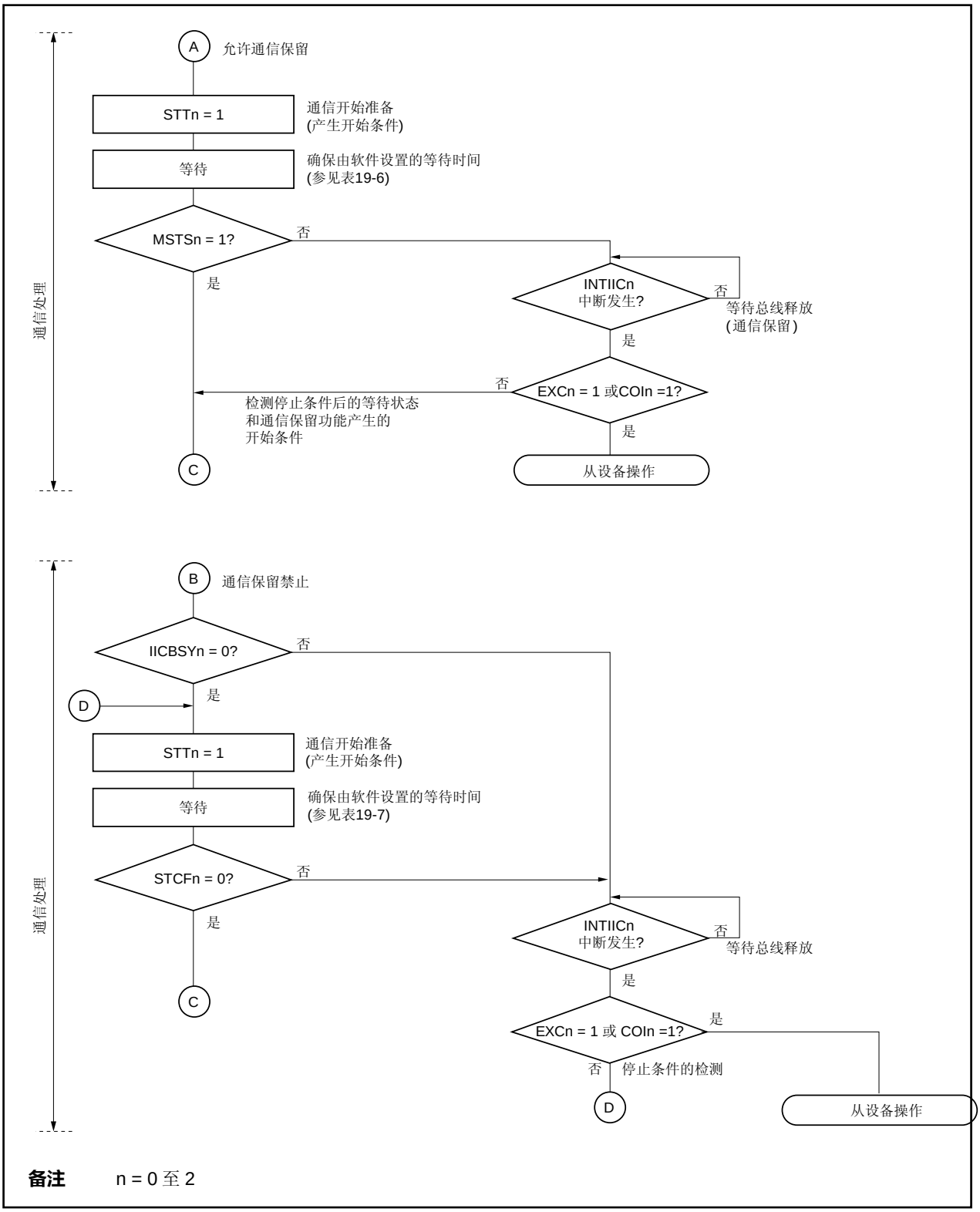
图 19-19. 多主系统中的主设备操作（1/3）



注 确定总线释放状态（IICCLn.CLDn 位 = 1，IICCLn.DADn 位 = 1）已经维持了一段时间（比如：1 帧）。当 SDA0n 引脚稳定为低电平时，参考通信中使用产品的规约，决定是否释放 I2C0n 总线（SCL0n，SDA0n 引脚 = 高电平）。

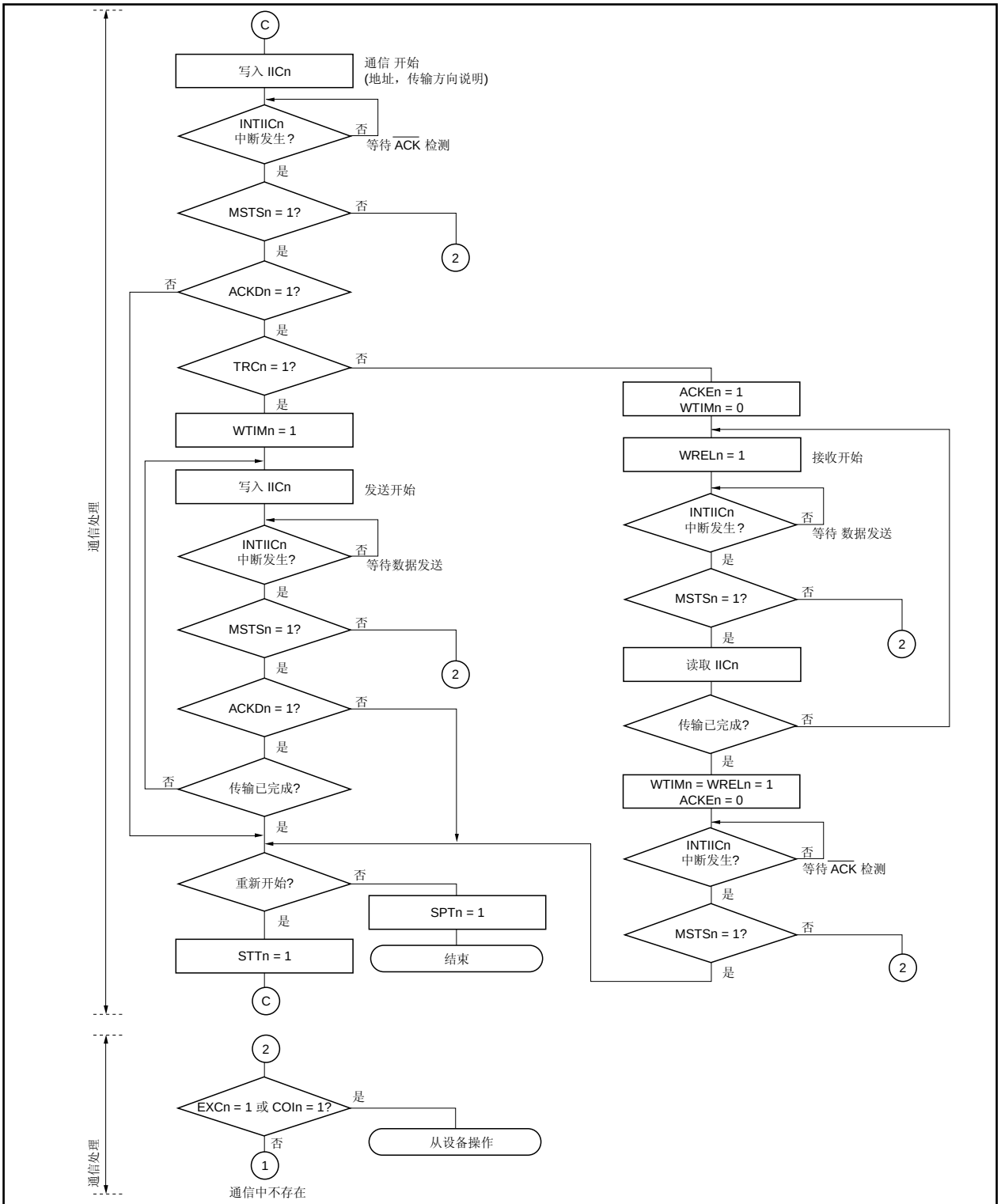
备注 n = 0 至 2， m = 0，1

图 19-19. 多主系统中的主设备操作（2/3）



备注 n = 0 至 2

图 19-19. 多主系统中的主设备操作（3/3）



备注

1. 确认发送和接收格式与参与通信的产品规格一致。
2. 当 V850ES/JG3-L 单片机在多主系统中用作主设备时，每次发生 INTIICn 中断时都要读取 IICSn.MSTSn 位以确定仲裁结果。
3. 当 V850ES/JG3-L 单片机在多主系统中用作从设备时，每次发生 INTIICn 中断时都要使用 IICSn 和 IICFn 寄存器以决定下次处理的状态。
4. n = 0 至 2

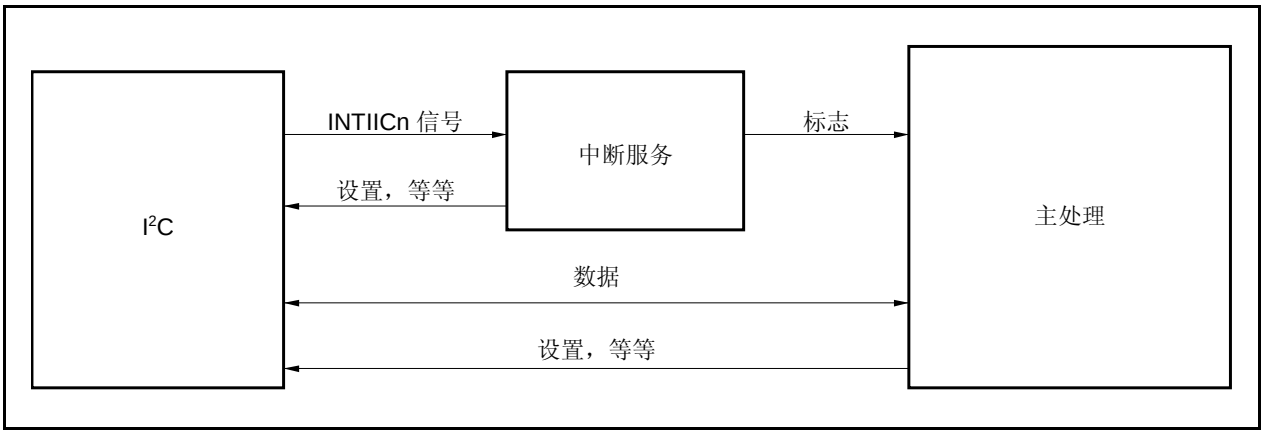
19.16.3 从设备操作

以下展示从设备操作的处理过程。

基本而言，从设备的操作都是由事件驱动的，所以，必需通过 INTIICn 中断进行处理（处理需要操作状态的重大改变，比如，通信过程中检测到停止条件）。

以下描述假定数据通信不支持扩展码，还假设只有在状态改变处理时才执行 INTIICn 中断服务，并且在主程序处理期间执行实际的数据通信。

图 19-20. 从设备操作期间的软件概况



所以，系统准备了以下 3 个标志，以便将其传送给主程序，从而执行数据传输处理，而不使用 INTIICn 信号。

（1）通信模式标志

该标志表示以下通信状态：

清除模式： 未进行数据通信

通信模式： 正在进行数据通信（检测到有效地址，检测到停止条件，未检测到来自主设备的 $\overline{\text{ACK}}$ 以及地址不匹配）

（2）就绪标志

该标志表示使能数据通信。这和正常数据传输期间发生 INTIICn 中断的状态相同。该标志在中断处理模块内被置位，在主程序模块中被清除。第一个数据发送的就绪标志并不在中断处理模块内置位，所以，第一个数据发送后不需要清除处理（地址匹配被视为下一个数据的请求）。

（3）通信方向标志

该标志表示数据通信的方向，和 IICSn.TRCn 位的值相同。

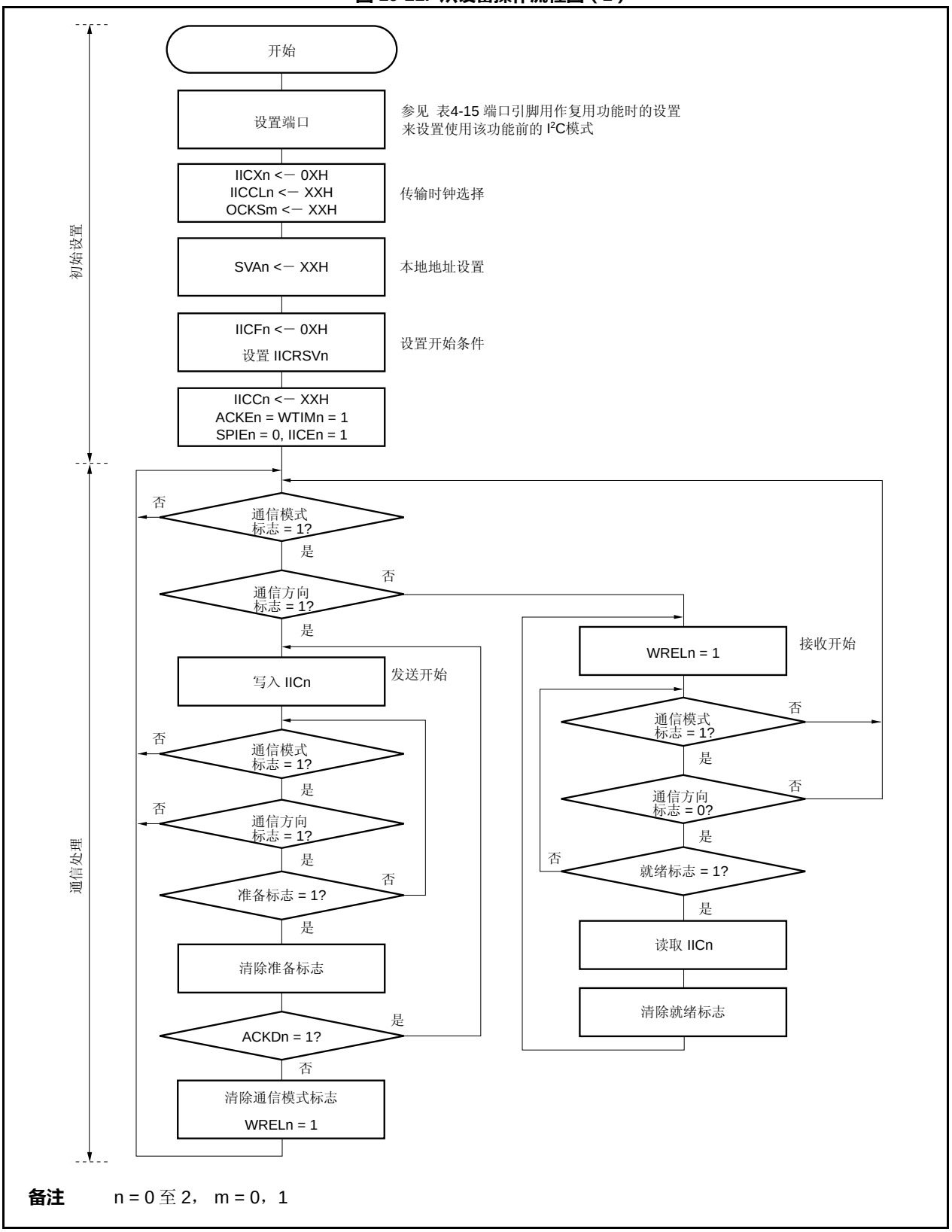
下面展示了从设备操作期间主程序模块的操作。

启动 I²C0n 并等待通信使能状态。当通信被使能后，使用通信模式标志和就绪标志执行传输（停止条件和开始条件的处理由中断处理执行，条件通过标志位确认）。

对于发送操作，重复发送，直到主设备停止返回 $\overline{\text{ACK}}$ 信号，当主设备停止返回 $\overline{\text{ACK}}$ 信号时，传输完毕。

对于接收操作，接收指定数量的数据项，且在传输完成后，不为下一个数据立即返回 $\overline{ACK}$ 信号。此后，主设备产生停止条件或重新开始条件。这将导致退出通信。

图 19-21. 从设备操作流程图（1）



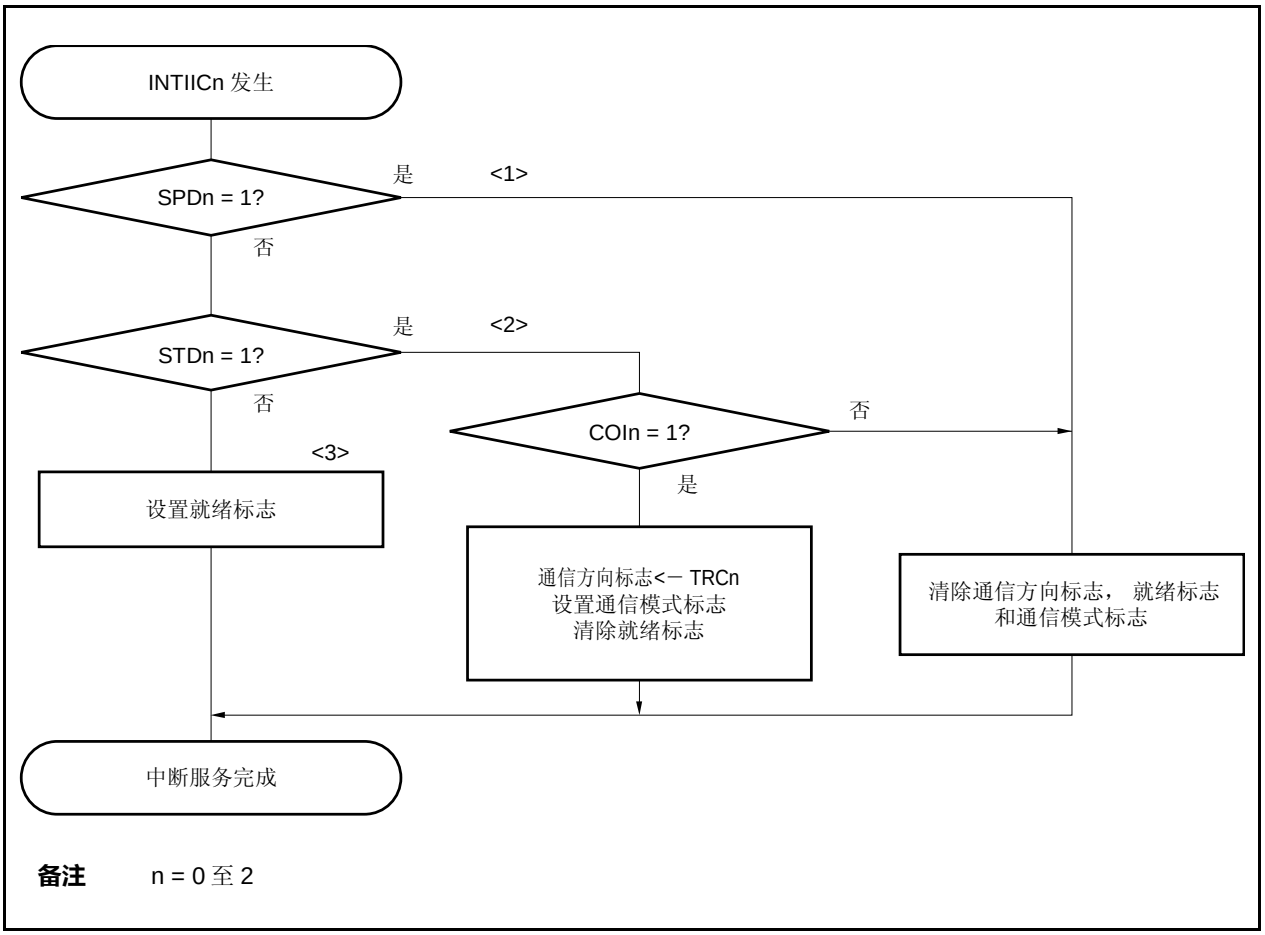
备注 n = 0 至 2， m = 0，1

下面展示了从设备使用 INTIICn 中断进行处理的一个示例（这里假定不使用扩展码）。在 INTIICn 中断期间，状态得以确认并且执行下列步骤。

<1> 当检测到停止条件时，通信终止。
<2> 当检测到开始条件后，进行地址确认。如果地址不匹配，则通信终止。如果地址匹配，设置通信模式并且释放等待状态，从中断中返回操作处理（就绪标志被清除）。
<3> 对于数据的发送/接收过程，当就绪标志置位时，I^2C0n 总线保持为等待状态，从中断中返回操作处理。

备注 上述<1> 至<3> 步骤对应**图 19-22 从设备操作流程（2）**的<1> 至<3>步骤。

图 19-22. 从设备操作流程（2）



19.17 数据通信的时序

当使用 I²C 总线模式时，主设备通过串行总线输出一个地址，在多个从设备中选择其中一个作为自己的通信伙伴。

输出从设备地址以后，主设备发送 IICSn.TRCn 位指定数据传输方向，然后，开始与从设备进行串行通信。

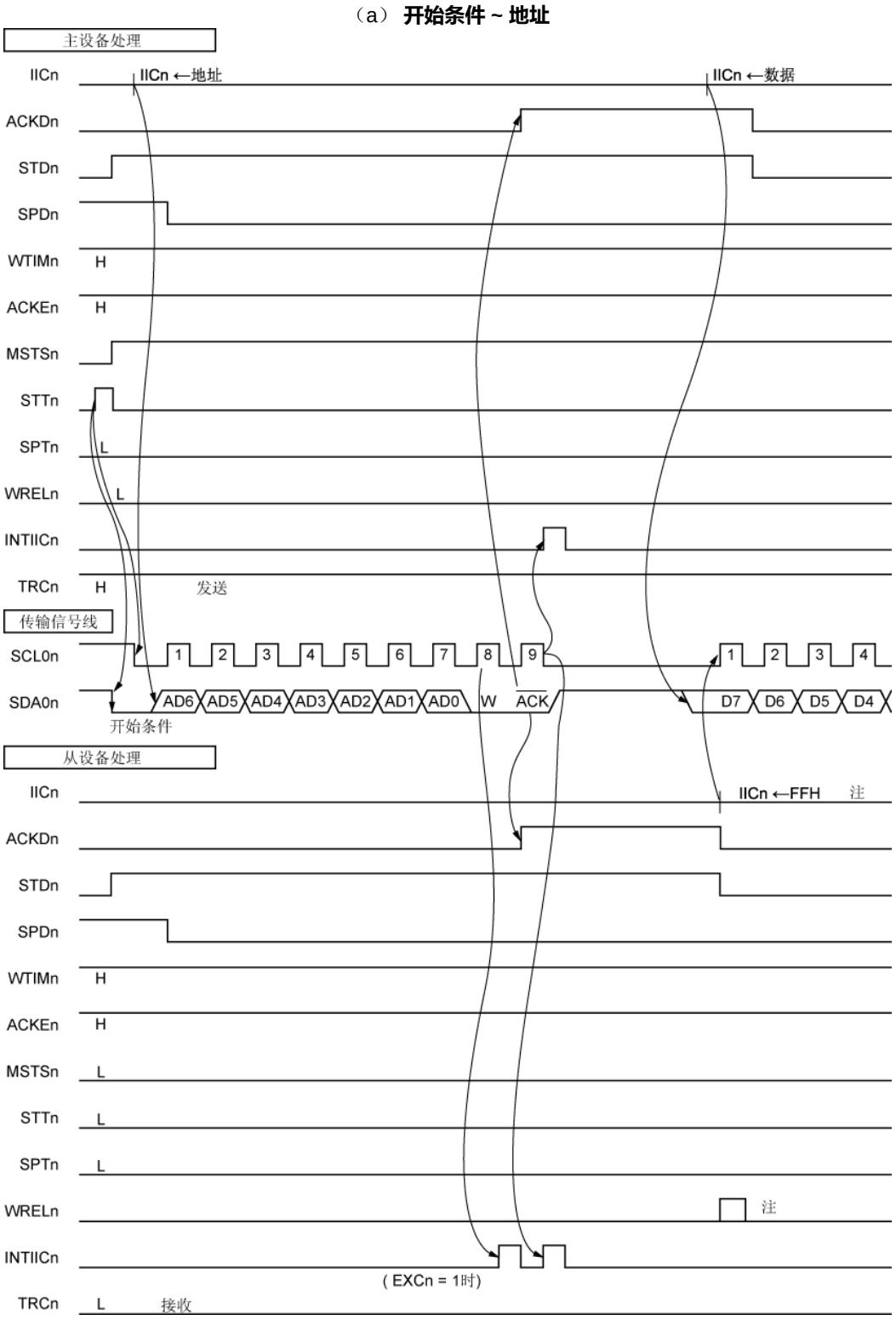
IICn 寄存器的移位操作和串行时钟引脚（SCL0n）信号的下降沿同步。发送数据被传送到 SO 锁存器并经由 SDA0n 引脚输出（MSB 先行）。

由 SDA0n 引脚输入的数据在 SCL0n 引脚信号的上升沿被 IICn 寄存器捕获。

数据通信时序如下所示。

备注 n = 0 至 2

图 19-23. 主设备向从设备传输示例
（当主设备和从设备都选择 9 时钟等待时）（1/3）

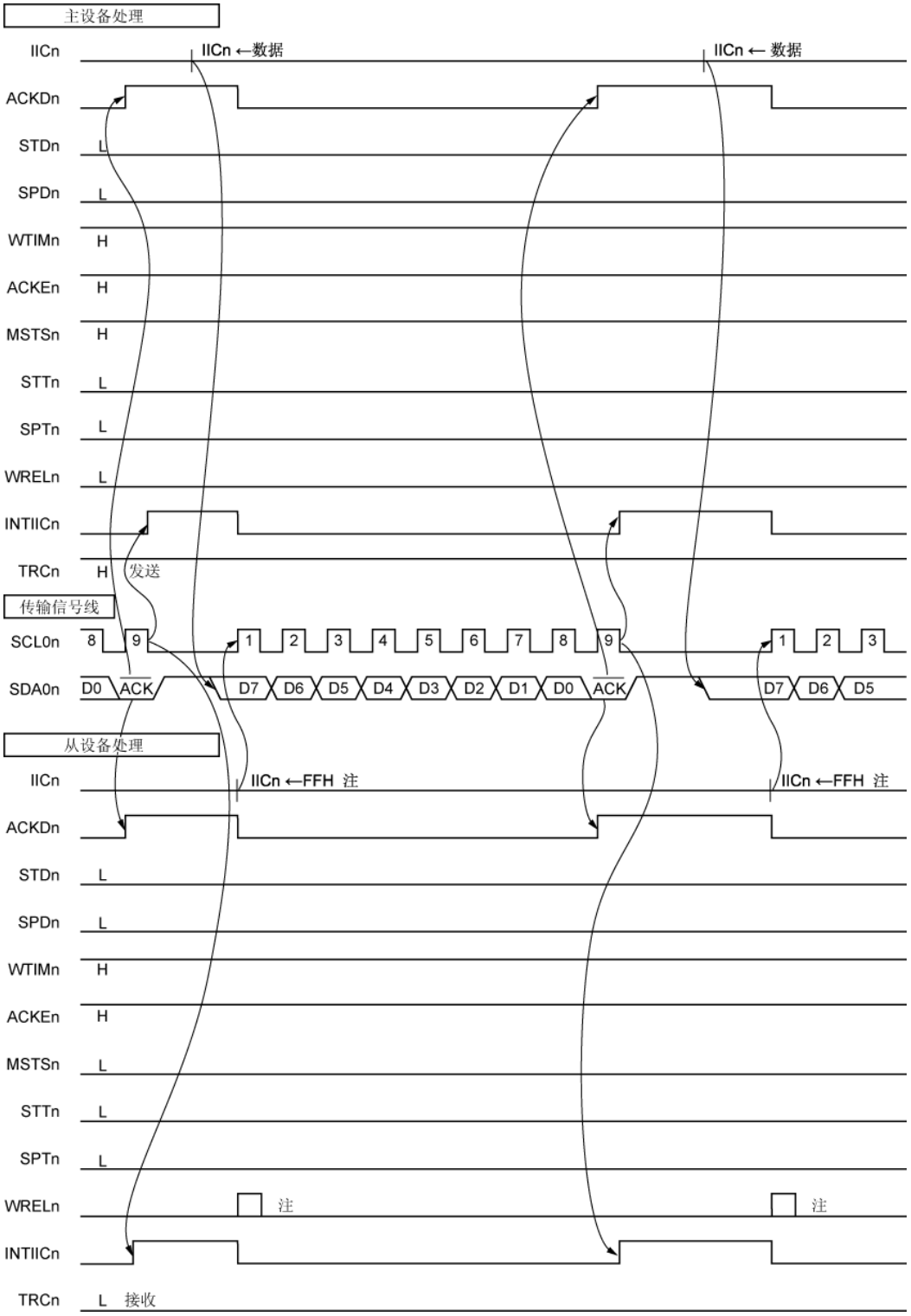


注 要取消从设备等待，向 IICn 写入 FFH，或将 WRELn 置位。

备注 n = 0 至 2

图 19-23. 主设备向从设备传输示例
（当主设备和从设备都选择 9 时钟等待时）（2/3）

（b）数据

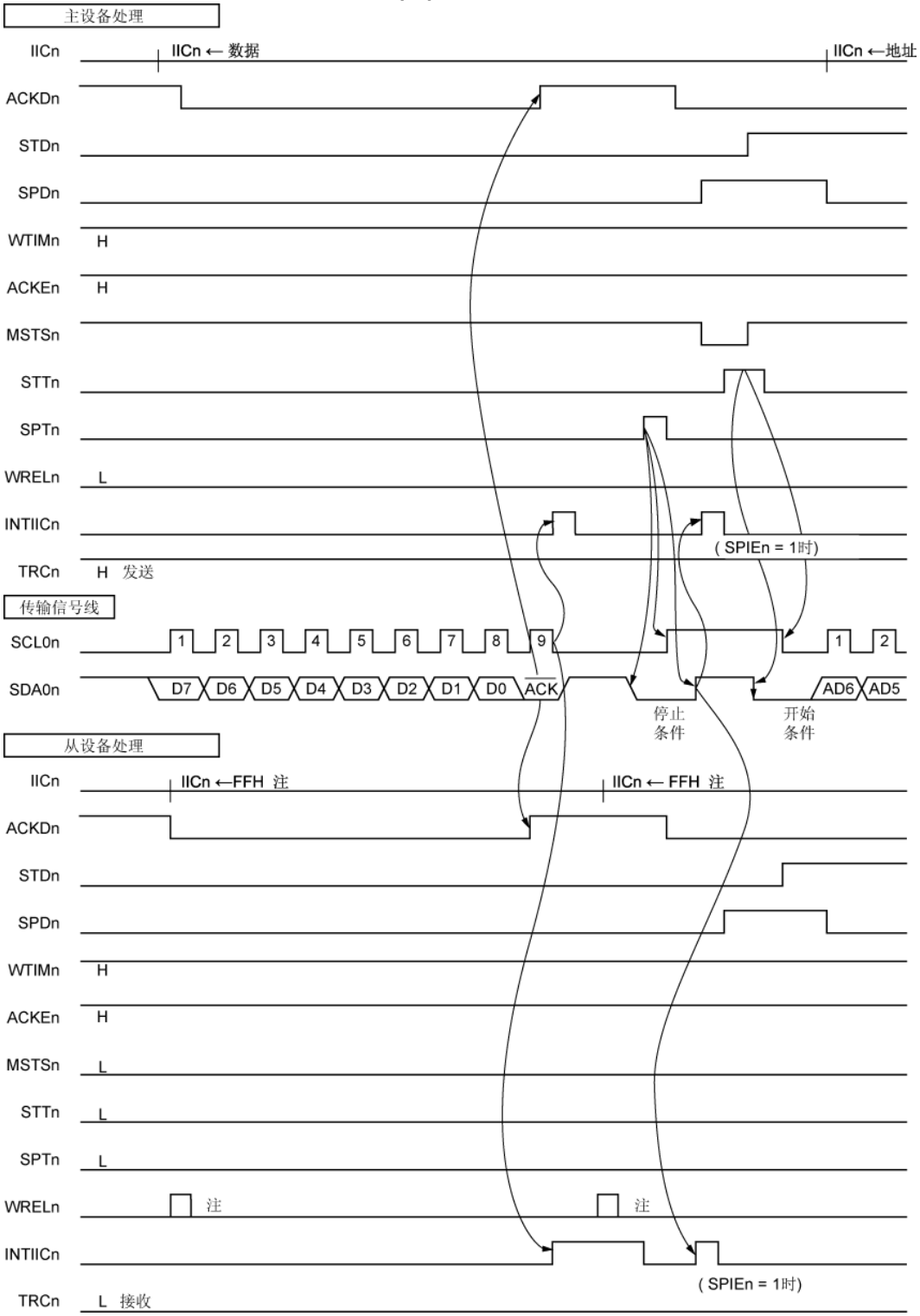


注 要取消从设备等待，向 IICn 写入 FFH，或将 WRELn 置位。

备注 n = 0 至 2

图 19-23. 主设备向从设备传输示例
（当主设备和从设备都选择 9 时钟等待时）（3/3）

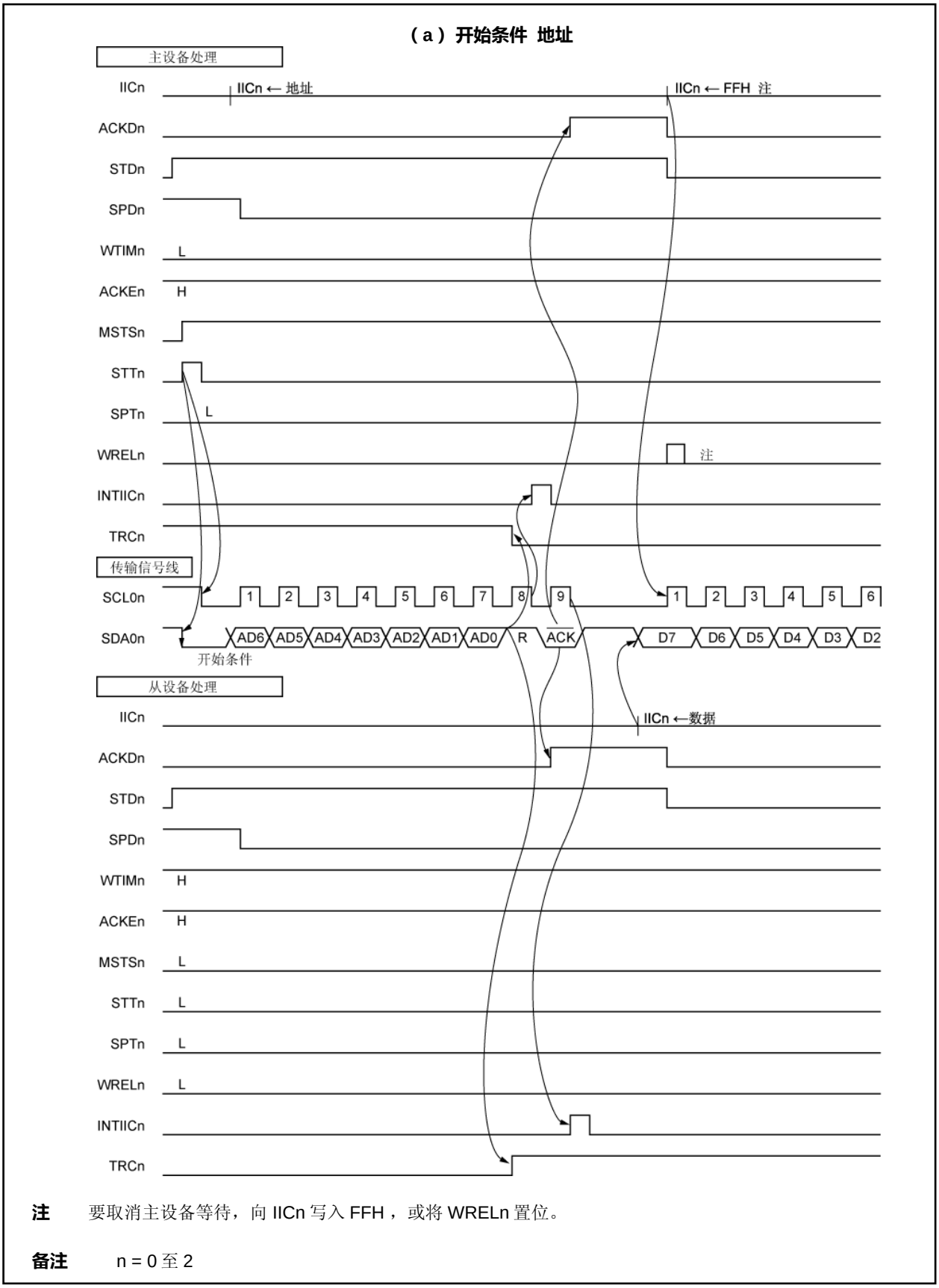
（c）停止条件



注 要取消从设备等待，向 IICn 写入 FFH，或将 WRELn 置位。

备注 n = 0 至 2

图 19-24. 从设备向主设备传输示例
（主设备选择 8 时钟等待，从设备选择 9 时钟等待）（1/3）

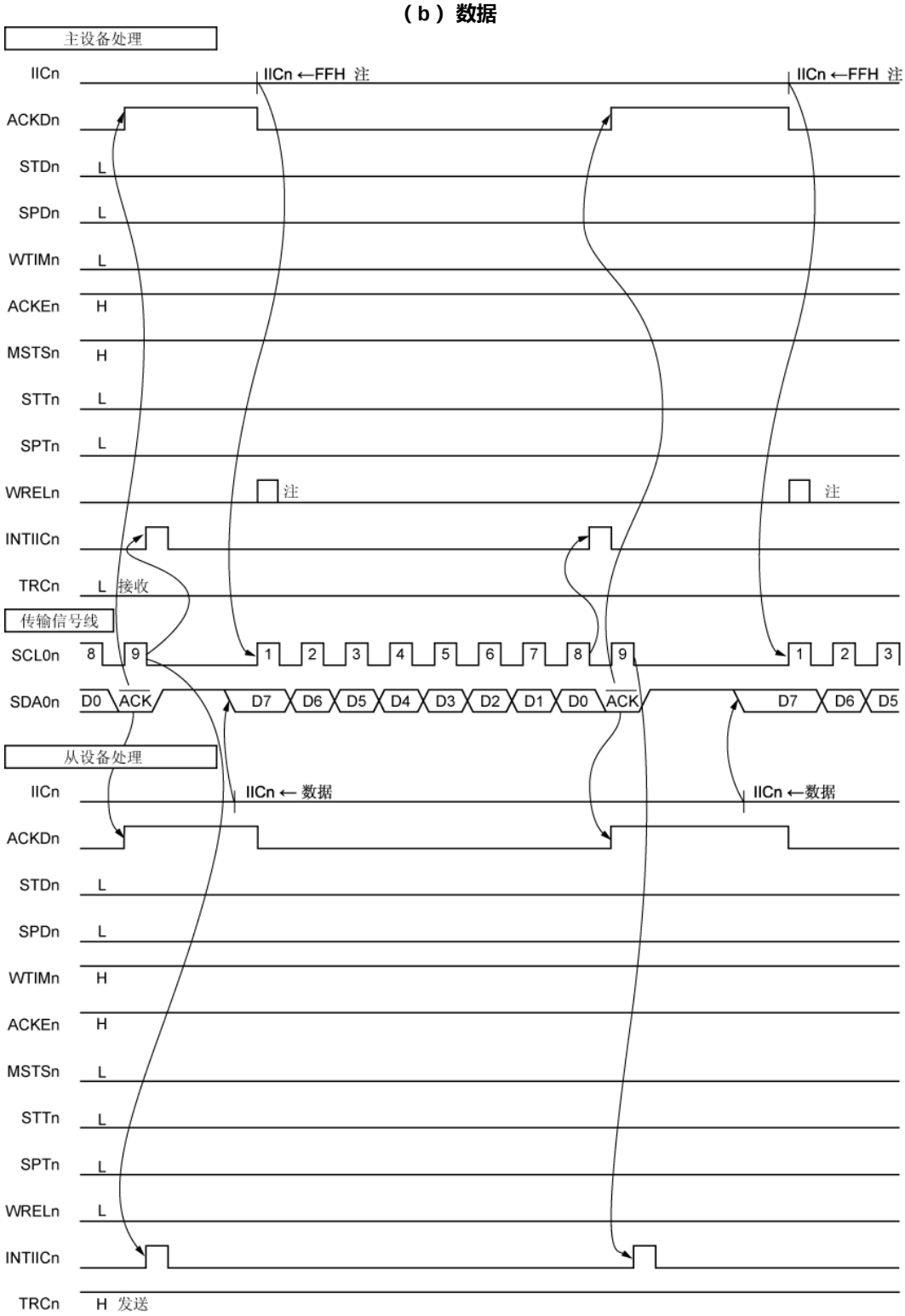


注 要取消主设备等待，向 IICn 写入 FFH ，或将 WRELn 置位。

备注 n = 0 至 2

图 19-24. 从设备向主设备传输示例
（主设备选择 8 时钟等待，从设备选择 9 时钟等待）（2/3）

（b）数据

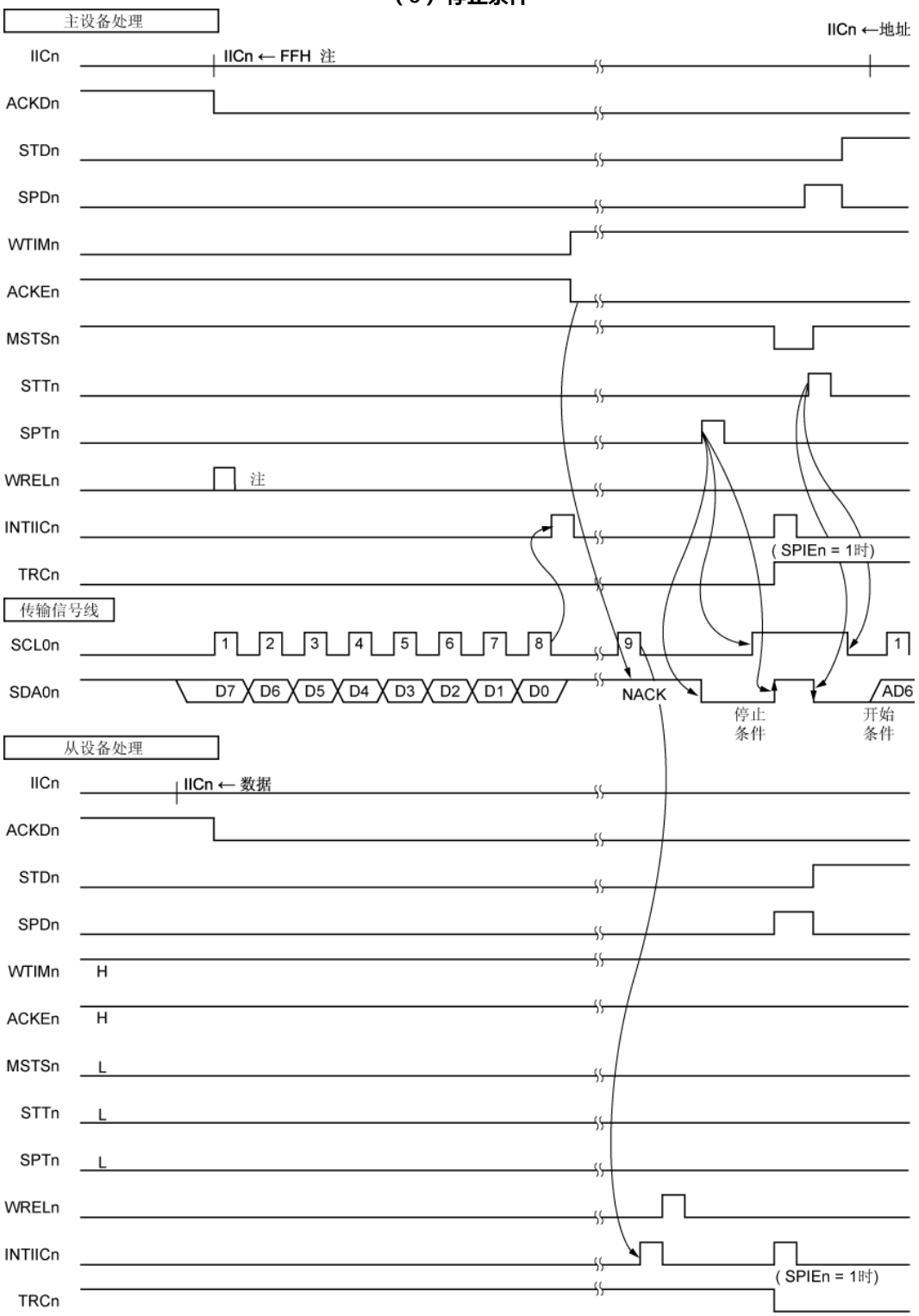


注 要取消主设备等待， 向 IICn 写入 FFH ，或将 WRELn 置位。

备注 n = 0 至 2

图 19-24. 从设备向主设备传输示例
（主设备选择 8 时钟等待，从设备选择 9 时钟等待）（3/3）

（c）停止条件



注 要取消主设备等待， 向 IICn 写入 FFH，或将 WRELn 置位。

备注 n = 0 至 2

第二十章 USB功能控制器USBF

V850ES/JG3-L 都内置有符合通用串行总线规范的 USB 功能控制器(USBF)。USB 功能控制器和外部主机设备之间使用基于令牌的协议，通过轮询方法进行数据通讯。

20.1 综述

- 符合通用串行总线规范
- 支持 12 Mbps（全速） 传输
- 集成有传输端点

端点名称	FIFO 空间大小（字节）	传输类型	备注
端点 0 读	64	控制传输	–
端点 0 写	64	控制传输	–
端点 1	64 × 2	批量 1 传输（IN）	2 缓冲区配置
端点 2	64 × 2	批量 1 传输（OUT）	2 缓冲区配置
端点 3	64 × 2	批量 2 传输（IN）	2 缓冲区配置
端点 4	64 × 2	批量 2 传输（OUT）	2 缓冲区配置
端点 7	8	中断传输	–

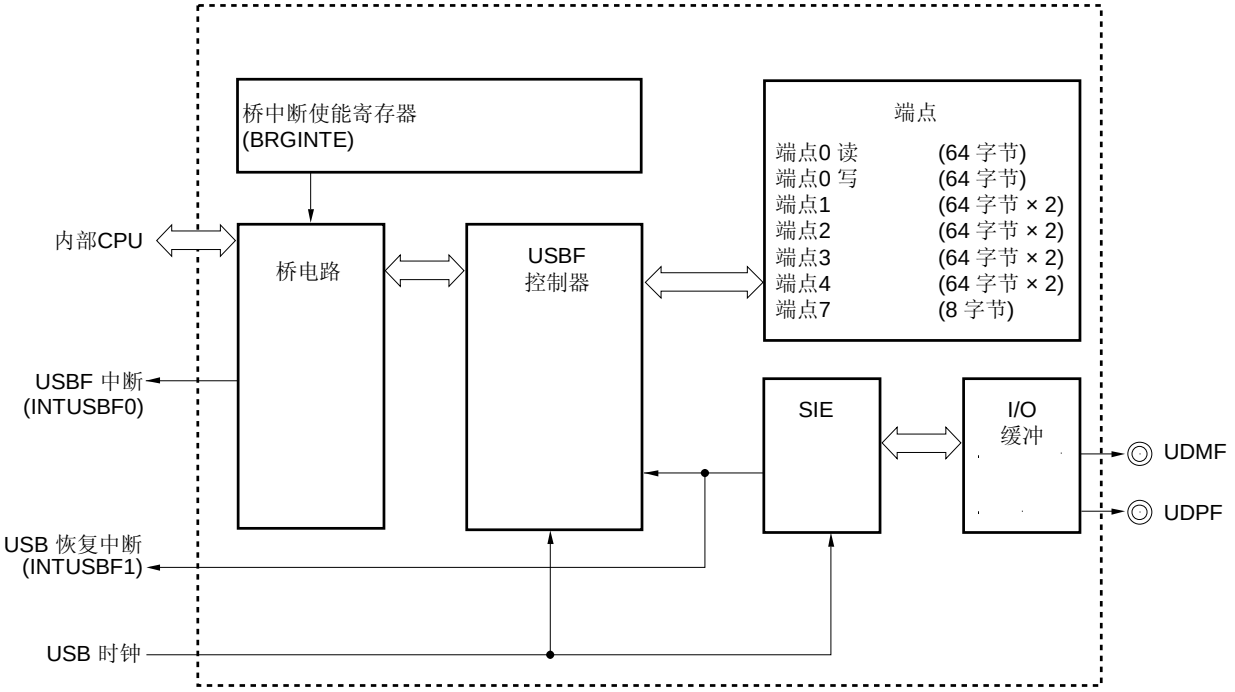
- 批量传输(IN/OUT)可以作为 DMA 传输来执行(2 周期单次传输模式)
- 时钟：可以选择内部时钟(6 MHz 外部时钟 × 内部时钟 8 倍频 = 48 MHz 内部时钟)或外部时钟(外部时钟输入到 UCLK 引脚(f_{USB} = 48 MHz)) 。

注意事项 对于表 20.6.2 USB 功能控制器寄存器列表中列举的寄存器，必须将内部时钟或外部时钟指定为 USB 时钟，且向 USB 功能控制器供给时钟之后，才能访问这些寄存器。

20.2 配置

20.2.1 框图

图 20-1. USB 功能控制器的框图



备注 虚线内：这些功能都包括在 USB 功能控制器中。

20.2.2 USB存储器映射

对于 CPU 来说，USB 功能控制器被分配到微控制器内的 00200000H 至 0024FFFFH 存储器空间。为了方便使用，该存储器空间的划分如下。

表 20-1. CPU 存储器空间的划分

地址	区域	
00200000H 至 00200092H	EPC 控制寄存器区域	
00200100H 至 00200114H	EPC 数据保持寄存器区域	
00200144H 至 002003C4H	EPC 请求数据寄存器区域	
00200400H 至 00200408H	桥寄存器区域	
00200500H 至 0020050EH	DMA 寄存器区域	
00201000H	批量-in 寄存器区域	EP1 (批量-IN1)
00202000H		EP3 (批量-IN2)
00210000H	批量-out 寄存器区域	EP2 (批量-Out1)
00220000H		EP4 (批量-Out2)
00240000H	外设控制寄存器区域	

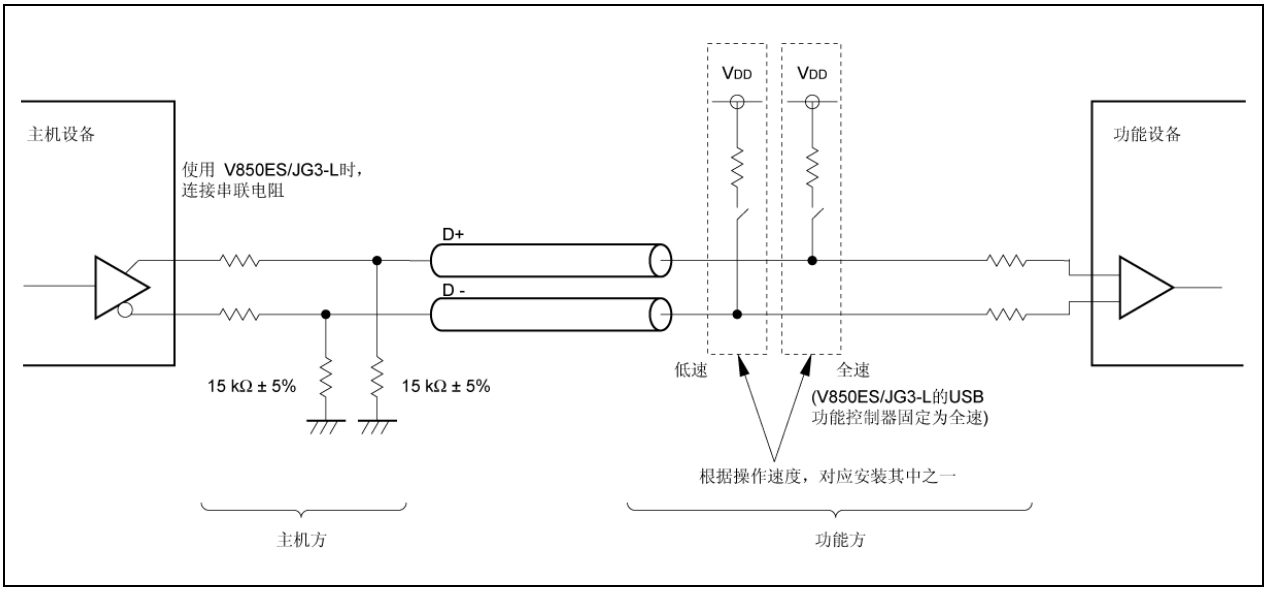
20.3 外部电路配置

20.3.1 概述

在 USB 传输中，当主机控制器和功能控制器互相进行通讯时，USB 信号(D+/D–)必须连接上拉/下拉电阻，以便识别通讯参与方。并且在 V850ES/JG3-L 中，还必须连接串联电阻。

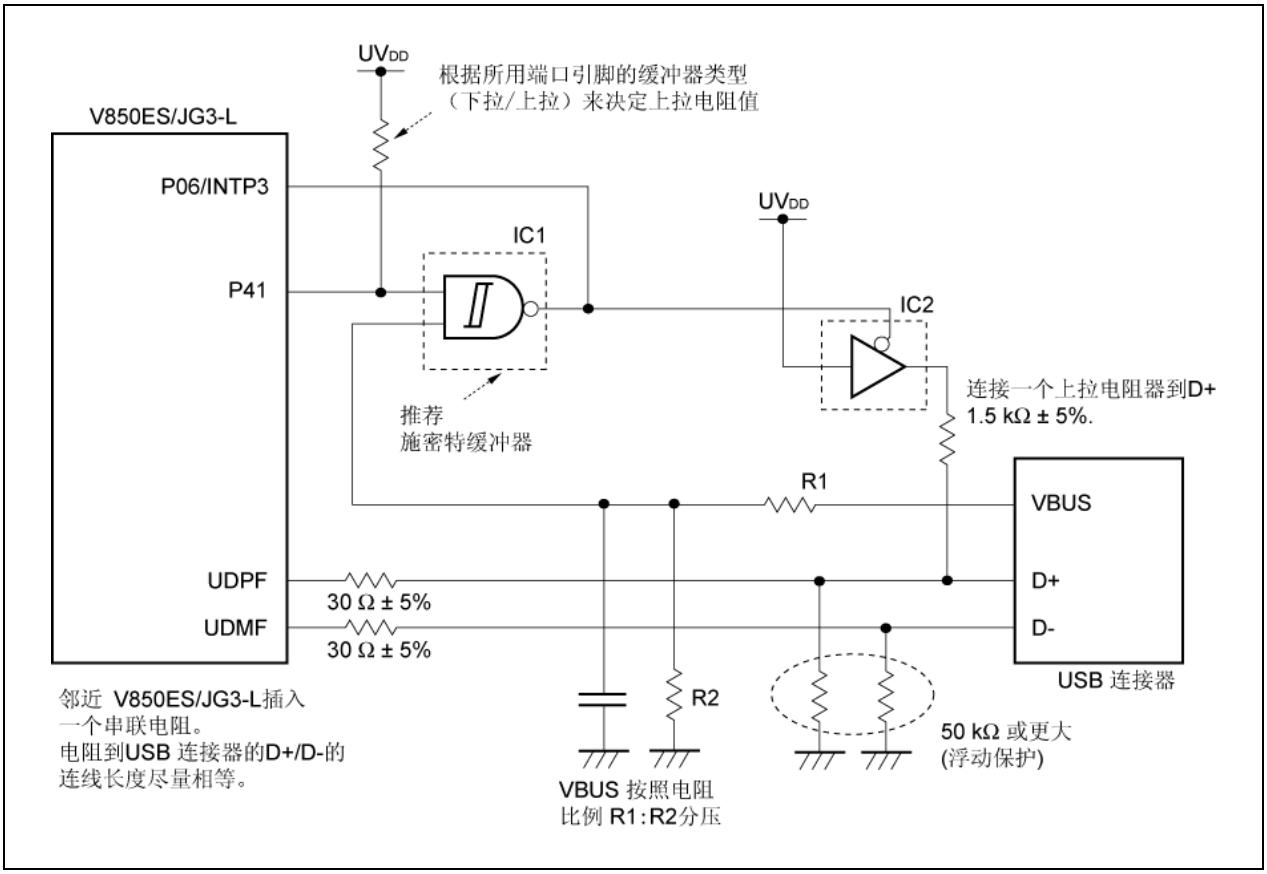
下图展示了 USB 传输线的概要配置。外部配置的详情参见各个章节的描述。

图 20-2. USB 传输线中的上拉，下拉串联电阻概要配置



20.3.2 连接配置

图 20-3. USB 功能控制器连接示例



(1) 连接到 D+/D–的串联电阻

连接 30 Ω ±5%串联电阻到 V850ES/JG3-L 内置 USB 功能控制器的 D+/D– 引脚 (UFDP, UFDM)。如果没有连接串联电阻，阻抗值达不到要求，输出波形可能会受到干扰。

串联电阻的位置要邻近 V850ES/JG3-L，各串联电阻和 USB 连接器之间的连线长度相同，这样使得 D+ 和 D– 的阻抗相等 (推荐使用 90 Ω ±5%的电阻)。

(2) D+的上拉控制

因为 V850ES/JG3-L 的功能控制器被固定位全速(FS)，请确保通过 1.5 kΩ ±5%将 D+ 引脚 (UFDP)上拉到 UVDD。

为了禁止连接报告(D+ 上拉) 到 USB 主机/HUB (比如在高优先级服务或初始化期间)，在系统中通过一个通用端口来控制 D+的上拉电阻。对于图 20-3 中所示的电路，通过一个通用端口和 USB 线 VBUS (与"AND"电路)来控制上拉控制信号和 D+引脚的 VBUS 输入信号。在图 20-3 中，如果通用端口是低电平，D+的上拉被禁止。

对于图 20-3 中使用的 IC2，当系统掉电时，对该 IC 的电压可以维持。

(3) 检测 USB 线连接/断开

USB 功能控制器(USBF)需要一个 VBUS 输入信号来确认 USB 线是连接或断开，因为 USBF 的状态由硬件控制。当 USBF 掉电时，USB 线 VBUS 连接到 USB 主机或 HUB，来自 USB 主机或 HUB 的电压(5 V)被提供作为 VBUS 输入信号。于是，图 20-3 使用的 IC1，当系统掉电时，对该 IC 的电压可以维持。断开图 20-3 中电路的 USB 线，当 VBUS 电压下降时，输入到 INTP3 的信号可能会不稳定。因此图 20-3 中的 IC1，推荐使用施密特缓冲器。

(4) 初始化期间或当 USBF 不使用时的浮动保护

当 USB 功能控制器被初始化或不使用时，为了放置出现浮动状态，使用 50 kΩ或更大的电阻将 D+/D– 引脚下拉。

20.4 注意事项

(1) 时钟准确度

为了操作 USB 功能控制器，USB 时钟必须使用内部时钟 (6 MHz 外部时钟 × 内部时钟 8 倍频 = 48 MHz 内部时钟) 或外部时钟(输入到 UCLK 引脚的外部时钟 (f_{USB} = 48 MHz))。当内部时钟用作 USB 时钟，使用的谐振器准确度 6 MHz ±500 ppm (最大值)。当使用外部时钟，UCLK 引脚使用的时钟准确度 48 MHz ±500 ppm (最大值)。如果 USB 时钟的准确度下降，传输数据可能无法满足 USB 规范要求。

(2) 停止 USB 时钟

当选择主时钟(f_{XX})作为 USB 功能控制器时钟时，必须要停止 USB 功能控制器。请确保停止 USB 功能控制器（将 UFCKMSK 寄存器的第 1 位和第 0 位设置为 1）时，首先停止主时钟(f_{XX})。

如果没有先停止 USB 功能控制器就已停止主时钟(f_{XX})，当主时钟(f_{XX})重启时，可能会因为时钟脉冲上的噪声导致误操作。

同理，从 UCLK 引脚输入的外部时钟信号被选择作为 USB 功能控制器时钟时，必须采取措施，以防止外部电路引起的时钟脉冲上产生的噪声。如果该措施不可行，在必须在停止主时钟(f_{XX})之前先停止 USB 功能控制器。

20.5 请求

USB 标准有请求命令，将来自主机设备的请求报告给功能设备，来执行响应过程。

请求是在控制传输的建立（SETUP）阶段接收到的，大部分可以通过 USB 功能控制器(USBF)的硬件自动处理。

20.5.1 自动请求

（1）译码

下表列出了请求和译码值之间的请求格式以及对应。

表 20-2. 请求格式

偏移量	域名称	
0	bmRequestType	
1	bRequest	
2	wValue	低位端
3		高位端
4	wIndex	低位端
5		高位端
6	wLength	低位端
7		高位端

表 20-3. 请求和译码值之间的对应

偏移量 / 请求	译码值								响应			数据阶段
	bmRequestType	bRequest	wValue		wIndex		wLength		Df	Ad	Cf	
	0	1	3	2	5	4	7	6				
GET_INTERFACE	81H	0AH	00H	00H	00H	0nH	00H	01H	STALL	STALL	ACK NAK	√
GET_CONFIGURATION	80H	08H	00H	00H	00H	00H	00H	01H	ACK NAK	ACK NAK	ACK NAK	√
GET_DESCRIPTOR 设备	80H	06H	01H	00H	00H	00H	XXH	XXH 注1	ACK NAK	ACK NAK	ACK NAK	√
GET_DESCRIPTOR 配置	80H	06H	02H	00H	00H	00H	XXH	XXH 注1	ACK NAK	ACK NAK	ACK NAK	√
GET_STATUS 设备	80H	00H	00H	00H	00H	00H	00H	02H	ACK NAK	ACK NAK	ACK NAK	√
GET_STATUS 端点 0	82H	00H	00H	00H	00H	00H 80H	00H	02H	ACK NAK	ACK NAK	ACK NAK	√
GET_STATUS 端点 X	82H	00H	00H	00H	00H	$$H	00H	02H	STALL	STALL	ACK NAK	√
CLEAR_FEATURE 设备注2	00H	01H	00H	01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
CLEAR_FEATURE 端点 0 注2	02H	01H	00H	00H	00H	00H 80H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
CLEAR_FEATURE 端点 X 注2	02H	01H	00H	00H	00H	$$H	00H	00H	STALL	STALL	ACK NAK	×
SET_FEATURE 设备注3	00H	03H	00H	01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
SET_FEATURE 端点 0 注3	02H	03H	00H	00H	00H	00H 80H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
SET_FEATURE 端点 X 注3	02H	03H	00H	00H	00H	$$H	00H	00H	STALL	STALL	ACK NAK	×
SET_INTERFACE	01H	0BH	00H	0#H	00H	0?H	00H	00H	STALL	STALL	ACK NAK	×
SET_CONFIGURATION 注4	00H	09H	00H	00H 01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
SET_ADDRESS	00H	05H	XXH	XXH	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×

备注 √： 数据阶段
×： 无数据阶段

注 **1.** 如果 wLength 值比准备的值小，wLength 值被返回；如果 wLength 值比准备的值大，准备的值被返回。
2. 在状态阶段接收到 ACK 时，CLEAR_FEATURE 请求会清除 UF0 设备状态寄存器 L（UF0DSTL）和 UF0 EPn 状态寄存器 L（UF0EnSL）（n = 0 到 4，7）。

注 **3.** 在状态阶段中接收到 ACK 时，SET_FEATURE 请求会设置 UF0 设备状态寄存器 L（UF0DSTL）和 UF0 EPn 状态寄存器 L（UF0EnSL）（n = 0 到 4，7）。在 GET_STATUS 端点 0 请求、SET_FEATURE 端点 0 请求以及由 CPUDEC 中断请求产生的请求以外，请求的控制传输数据阶段或状态阶段中，如果在接收到 CLEAR_FEATURE 端点 0 请求之前，UF0E0SL 寄存器的 E0HALT 位被置位，产生 STALL 响应。对于不支持的请求的 STALL 响应，不会将 UF0E0SL 寄存器的 E0HALT 位置 1，并且只要接收到下一个建立（SETUP）令牌，STALL 响应立即被清除。

4. 如果 wValue 不是默认值，产生一个自动 STALL 响应。

注意事项 **1. 通用串行总线规范定义的控制传输的顺序在下列条件下不能满足。在这些条件下，操作无法保证。**

- **如果没有经历 SETUP 阶段，突然接收到一个 IN / OUT 令牌**
- **如果在 SETUP 阶段的数据阶段发送 DATA PID1**
- **如果接收到 128 个地址或更多地址的令牌**
- **如果在 SETUP 阶段发送的请求数据少于 8 字节**

2. 在状态阶段中，即使主机发送 Null 包以外的数据，也会产生 ACK 响应。

3. 如果在 FW 处理的控制传输（读）期间，wLength 值是 00H，对于控制传输（无数据），会自动发送一个 Null 包。FW 请求不会自动发送 Null 包。

备注 **1.** Df：默认状态，Ad：地址状态，Cf：配置状态

2. n = 0 到 4
根据目标的接口编号是否有效，UF0 有效接口编号寄存器（UF0AIFN）的设置决定接口编号为 1 到 4 的请求是否被正确响应。

3. $$：包含传输方向的有效端点号
有效的端点由当前置位的复用设置号码决定（参见 **20.6.3(36) UF0 有效复用设置寄存器（UF0AAS）**，**(38) UF0 端点 1 接口映射寄存器（UF0E1IM）至(42) UF0 端点 7 接口映射寄存器（UF0E7IM）**）。

4. ? 和 #：从主机发送的值（接口编号 0 到 4 的信息）
根据目标的接口编号和复用设置是否有效，UF0 有效接口编号寄存器（UF0AIFN）和 UF0 有效复用设置寄存器（UF0AAS）可以判定对应每个接口编号的复用设置请求是否被正确响应。

（2）处理

默认状态、地址状态和配置状态下的自动请求处理如下所示。

备注 默认状态： 该状态下，操作在默认地址上执行
地址状态： 地址被分配后的状态
配置状态：正确接收到 SET_CONFIGURATION wValue = 1 后的状态

（a）CLEAR_FEATURE() 请求

如果不能清除 CLEAR_FEATURE()请求，如果 FEATURE 不存在，或者如果目标是一个不存在的接口或端点，在状态阶段产生 STALL 响应。如果 wLength 值非 0，也会产生 STALL 响应。

- 默认状态： 只有当目标是对设备或端点 0 的请求时，接收到 CLEAR_FEATURE()请求，才会产生正确响应；否则，在状态阶段产生 STALL 响应。
- 地址状态： 只有当目标是对设备或端点 0 的请求时，接收到 CLEAR_FEATURE()请求，才会产生正确响应；否则，在状态阶段产生 STALL 响应。
- 配置状态： 只有当目标是对一个存在的设备或端点的请求时，接收到 CLEAR_FEATURE()请求，才会产生正确响应；否则，在状态阶段产生 STALL 响应。

当 CLEAR_FEATURE()请求已经被正确处理后，UF0CLR 请求寄存器（UF0CLR）的对应位被置 1，UF0 EPn 状态寄存器 L（UF0EnSL）的 EnHALT 位被清除为 0，并且发出一个中断（n = 0 到 4，7）。如果在对象是端点时接收到 CLEAR_FEATURE()请求，对应端点的双向开关位（控制 DATA0 和 DATA1 之间的切换）总是被重置为 DATA0。

（b）GET_CONFIGURATION() 请求

如果 wValue、wIndex 或 wLength 中的任意一个不在表 20-3 列举的范围内，在数据阶段产生一个 STALL 响应。

- 默认状态： 当接收到 GET_CONFIGURATION()请求时，保存在 UF0 配置寄存器（UF0CNF）中的值被返回。
- 地址状态： 当接收到 GET_CONFIGURATION()请求时，保存在 UF0CNF 寄存器中的值被返回。
- 配置状态： 当接收到 GET_CONFIGURATION()请求时，保存在 UF0CNF 寄存器中的值被返回。

(c) GET_DESCRIPTOR() 请求

如果对象描述符的长度是 wMaxPacketSize 的倍数，返回一个 Null 包表示数据阶段的结束。如果此时描述符的长度小于 wLength 值，返回整个描述符；如果描述符的长度大于 wLength 值，返回 wLength 值长度的描述符。

- 默认状态： 当接收到 GET_DESCRIPTOR()请求时，保存在 UF0 设备描述符寄存器 n（UF0DDn）和 UF0 配置 / 接口 / 端点描述符寄存器 m（UF0CIEm）（n = 0 至 17，m = 0 至 255）中的值被返回。
- 地址状态： 当接收到 GET_DESCRIPTOR()请求时，保存在 UF0DDn 寄存器和 UF0CIEm 寄存器中的值被返回。
- 配置状态： 当接收到 GET_DESCRIPTOR()请求时，保存在 UF0DDn 寄存器和 UF0CIEm 寄存器中的值被返回。

UF0CIEm 寄存器中可以存储最大 256 字节的描述符。要返回的描述符多于 256 字节，需要设置 UF0MODC 寄存器的 CDCGDST 位为 1，并通过 FW 来处理 GET_DESCRIPTOR()请求。
在 UF0 描述符长度寄存器（UF0DSCL）中存储 UF0CIEm 寄存器设置的描述符的总字节数的值 – 1。传输数据由这个数据的值 + 1 和 wLength 控制。

(d) GET_INTERFACE() 请求

如果 wValue 和 wLength 中有一个不在表 20-3 列举的范围内，或者如果 wIndex 不是 UF0 有效接口编号寄存器（UF0AIFN）设置的值，在数据阶段产生一个 STALL 响应。

- 默认状态： 当接收到 GET_INTERFACE()请求时，在数据阶段产生一个 STALL 响应。
- 地址状态： 当接收到 GET_INTERFACE()请求时，在数据阶段产生一个 STALL 响应。
- 配置状态： 当接收到 GET_INTERFACE()请求时，存储在 UF0 接口 n 寄存器（UF0IFn）中对应 wIndex 值的值被返回（n = 0 到 4）。

(e) GET_STATUS() 请求

如果 wValue 、wIndex 或 wLength 中的任意一个不在表 20-3 列举的范围内，在数据阶段产生一个 STALL 响应。如果目标是不存在的接口或端点，在数据阶段产生也会一个 STALL 响应。

- 默认状态： 只有当接收到 GET_STATUS()请求，并且该请求是针对设备或端点 0 时，保存在目标状态寄存器[注] 中的值被返回；否则，在数据阶段产生 STALL 响应。
- 地址状态： 只有当接收到 GET_STATUS()请求，并且该请求是针对设备或端点 0 时，保存在目标状态寄存器[注] 中的值被返回；否则，在数据阶段产生 STALL 响应。
- 配置状态： 只有当接收到 GET_STATUS()请求，并且该请求是针对一个存在的设备或端点时，保存在目标状态寄存器[注] 中的值被返回；否则，在数据阶段产生 STALL 响应。

注 目标状态寄存器如下所示。
- 如果目标是一个设备：UF0 设备状态寄存器 L（UF0DSTL）
- 如果目标是端点 0： UF0 EP0 状态寄存器 L（UF0E0SL）
- 如果目标是端点 n： UF0 EPn 状态寄存器 L（UF0EnSL）（n = 1 到 4，7）

(f) SET_ADDRESS() 请求

如果 wIndex 或 wLength 中有一个不在表 20-3 列举的范围内，在数据阶段产生一个 STALL 响应。如果指定的设备地址大于 127 ，在数据阶段也会产生一个 STALL 响应。

- 默认状态： 当接收到 SET_ADDRESS()请求时，如果指定的地址不是 0，设备进入地址状态，并且将要输入到 SIE 的 USB 地址值更改为一个指定的地址值。如果指定的地址是 0，设备仍然保持为默认状态。
- 地址状态： 当接收到 SET_ADDRESS()请求时，如果指定的地址是 0，设备进入默认状态，并且将要输入到 SIE 的 USB 地址值返回为默认地址。如果指定的地址不是 0，设备仍然保持为地址状态，并且将要输入 SIE 的 USB 地址值更改为一个指定的新地址值。
- 配置状态： 当接收到 SET_ADDRESS()请求时，如果指定的地址是 0，设备仍然处于配置状态，并且将要输入到 SIE 的 USB 地址值返回为默认地址。在这种情况下，这个非 0 的端点仍然有效，并且对端点 0 以外的端点的控制传输（IN）、控制传输（OUT）、批量传输和中断传输也可以响应。如果指定的地址不是 0，设备仍然保持为配置状态，并且将要输入 SIE 的 USB 地址值更改为一个指定的新地址值。

(g) SET_CONFIGURATION() 请求

如果 wValue 、wIndex 或 wLength 中的任意一个不在表 20-3 列举的范围内，在数据阶段产生一个 STALL 响应。

- 默认状态：当接收到 SET_CONFIGURATION()请求时，如果指定的配置值是 1，UF0 模式状态寄存器（UF0MODS）的 CONF 位和 UF0 配置寄存器（UF0CNF）被置为 1。如果指定的配置值是 0，UF0MODS 寄存器的 CONF 位和 UF0CNF 寄存器被清为 0。也就是说，设备跳过地址状态转移到配置状态，在配置状态下响应默认地址。
- 地址状态：当接收到 SET_CONFIGURATION()请求时，如果指定的配置值是 1，UF0MODS 寄存器的 CONF 位和 UF0CNF 寄存器被置为 1，并且设备进入配置状态。如果指定的配置值是 0，设备仍然保持为地址状态。
- 配置状态：当接收到 SET_CONFIGURATION()请求时，如果指定的配置值是 0，UF0MODS 寄存器的 CONF 位和 UF0CNF 寄存器被置为 1，并且设备返回地址状态。如果指定的配置值是 1，设备仍然保持为配置状态。

如果 SET_CONFIGURATION()请求被正确处理，UF0 SET 请求寄存器（UF0SET）的目标位被置为 1，并且发出一个中断。即使指定的配置值与当前的配置值相同，在完成 SET_CONFIGURATION()请求之后，所有的暂停特征（Halt Feature）也被清除。如果 SET_CONFIGURATION()请求被正确处理，所有端点的数据切换总是再次被初始化为 DATA0（被定义为默认状态，复用设置 0，从接收到 SET_CONFIGURATION 请求至接收到 SET_INTERFACE 请求期间被设置位默认状态）。

(h) SET_FEATURE() 请求

如果 SET_FEATURE()请求用于不能被设置或不存在的特征 ，或者如果目标是不存在的接口或端点，在状态阶段产生一个 STALL 响应。如果 wLength 值不是 0，也会产生一个 STALL 响应。

- 默认状态：只有当请求是针对一个设备或端点 0 时，接收到 SET_FEATURE()请求，才会产生正确响应；否则，在状态阶段产生一个 STALL 响应。
- 地址状态：只有当请求是针对一个设备或端点 0 时，接收到 SET_FEATURE()请求，才会产生正确响应；否则，在状态阶段产生一个 STALL 响应。
- 配置状态：只有当请求是针对一个存在的设备或端点时，接收到 SET_FEATURE()请求，才会产生正确响应；否则，在状态阶段产生一个 STALL 响应。

当 SET_FEATURE()请求已经被正确处理时，UF0 SET 请求寄存器（UF0SET）的目标位和 UF0 EPn 状态寄存器 L（UF0EnSL）的 EnHALT 位被置为 1，并且发出一个中断（n = 0 到 4，7）。

（i）SET_INTERFACE() 请求

如果 wLength 不在表 20-3 列举的范围内，如果 wIndex 和 UF0 有效接口编号寄存器（UF0AIFN）的设置值不同，或者如果 wValue 值和 UF0 有效复用设置寄存器（UF0AAS）的设置值不同，在状态阶段产生一个 STALL 响应。

- 默认状态：　当接收到 SET_INTERFACE()请求时，在状态阶段产生一个 STALL 响应。
- 地址状态：　当接收到 SET_INTERFACE()请求时，在状态阶段产生一个 STALL 响应。
- 配置状态：　当接收到 SET_INTERFACE()请求时，在状态阶段发送 Null 包。

当 SET_INTERFACE()请求被正确处理时，发出一个中断。在 SET_INTERFACE()请求被清除后，所有链接连接到目标接口端点的暂停特征（Halt Feature）都被清除。所有与目标接口编号有关的端点的数据切换总是再次被初始化为 DATA0。当前选择的复用设置被正确处理的 SET_INTERFACE()请求更改时，受影响的端点的 FIFO 被完全清除，并且所有相关的中断源也被初始化。

当完成 SET_INTERFACE()请求时，连接到目标接口的所有端点的 FIFO 被清除。同时，暂停特征和 Data PID 被初始化，并且相关的 UF0 INT 状态 n 寄存器（UF0ISn）被清除为 0（n = 0 到 4）。（当 SET_CONFIGURATION 请求被完成时，只有暂停特征和 Data PID 被清除。）

如果在 DMA 传输中，SET_INTERFACE()不支持目标端点，DMA 请求信号马上置 0，SET_INTERFACE()请求已经完成时链接的端点的 FIFO 全部被清除。对 FIFO 清除的结果，通过 DMA 的数据传输没有被正确处理。

20.5.2 其它请求

（1）响应和处理

下表展示了如何响应和处理其它请求。

表 20-4. 其它请求的响应和处理

请求	响应和处理
GET_DESCRIPTOR 字符串	CPUDEC 中断请求的产生
GET_STATUS 接口	自动 STALL 响应
CLEAR_FEATURE 接口	自动 STALL 响应
SET_FEATURE 接口	自动 STALL 响应
所有 SET_DESCRIPTOR	CPUDEC 中断请求的产生
所有其它请求	CPUDEC 中断请求的产生

20.6 寄存器配置

20.6.1 USB控制寄存器

(1) USB 时钟选择寄存器（UCKSEL）

UCKSEL 寄存器用于选择 USB 控制器的工作时钟。

UCKSEL 寄存器可以按字节或按位进行读取或写入。

系统复位后，该寄存器被置为 00H。

复位后：00H R/W 地址：FFFFFF40H

符号	7	6	5	4	3	2	1	0
UCKSEL	0	0	0	0	0	0	UUSEL1	0

UUSEL1	USB控制器的工作时钟的选择
0	从 UCLK 引脚输入的外部时钟(f_{USB} = 48 MHz)
1	PLL 时钟

注意事项　　请确保设置第 7 位至第 2 位，以及第 0 位为“0”。

(2) USB 功能控制寄存器（UFCKMSK）

UFCKMSK 寄存器用于控制 USB 功能控制器操作的使能/禁止。

UFCKMSK 寄存器可以按字节或按位进行读取或写入。

系统复位后，该寄存器被置为 03H。

复位后：03H R/W 地址：FFFFFF41H

符号	7	6	5	4	3	2	1	0
UFCKMSK	0	0	0	0	0	0	UFBUFMSK	UFMSK

UFBUFMSK	UFMSK	USB功能控制器操作的使能/禁止
0	0	操作使能
0	1	操作停止（当 USB 被挂起时设置）
1	1	操作停止
其它		禁止设置

20.6.2 USB功能控制器寄存器列表

(1) EPC 控制寄存器

(1/2)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200000H	UF0 EP0NAK 寄存器	UF0E0N	R/W		√		00H
00200002H	UF0 EP0NAKALL 寄存器	UF0E0NA	R/W		√		00H
00200004H	UF0 EPNAK 寄存器	UF0EN	R/W		√		00H
00200006H	UF0 EPNAK 屏蔽寄存器	UF0ENM	R/W		√		00H
00200008H	UF0 SNDSIE 寄存器	UF0SDS	R/W		√		00H
0020000AH	UF0 CLR 请求寄存器	UF0CLR	R		√		00H
0020000CH	UF0 SET 请求寄存器	UF0SET	R		√		00H
0020000EH	UF0 EP 状态 0 寄存器	UF0EPS0	R		√		00H
00200010H	UF0 EP 状态 1 寄存器	UF0EPS1	R		√		00H
00200012H	UF0 EP 状态 2 寄存器	UF0EPS2	R		√		00H
00200020H	UF0 INT 状态 0 寄存器	UF0IS0	R		√		00H
00200022H	UF0 INT 状态 1 寄存器	UF0IS1	R		√		00H
00200024H	UF0 INT 状态 2 寄存器	UF0IS2	R		√		00H
00200026H	UF0 INT 状态 3 寄存器	UF0IS3	R		√		00H
00200028H	UF0 INT 状态 4 寄存器	UF0IS4	R		√		00H
0020002EH	UF0 INT 屏蔽 0 寄存器	UF0IM0	R/W		√		00H
00200030H	UF0 INT 屏蔽 1 寄存器	UF0IM1	R/W		√		00H
00200032H	UF0 INT 屏蔽 2 寄存器	UF0IM2	R/W		√		00H
00200034H	UF0 INT 屏蔽 3 寄存器	UF0IM3	R/W		√		00H
00200036H	UF0 INT 屏蔽 4 寄存器	UF0IM4	R/W		√		00H
0020003CH	UF0 INT 清除 0 寄存器	UF0IC0	W		√		FFH
0020003EH	UF0 INT 清除 1 寄存器	UF0IC1	W		√		FFH
00200040H	UF0 INT 清除 2 寄存器	UF0IC2	W		√		FFH
00200042H	UF0 INT 清除 3 寄存器	UF0IC3	W		√		FFH
00200044H	UF0 INT 清除 4 寄存器	UF0IC4	W		√		FFH
0020004CH	UF0 INT & DMARQ 寄存器	UF0IDR	R/W		√		00H
0020004EH	UF0 DMA 状态 0 寄存器	UF0DMS0	R		√		00H
00200050H	UF0 DMA 状态 1 寄存器	UF0DMS1	R		√		00H
00200060H	UF0 FIFO 清除 0 寄存器	UF0FIC0	W		√		00H
00200062H	UF0 FIFO 清除 1 寄存器	UF0FIC1	W		√		00H
0020006AH	UF0 数据结束寄存器	UF0DEND	R/W		√		00H
0020006EH	UF0 GPR 寄存器	UF0GPR	W		√		00H
00200074H	UF0 模式控制寄存器	UF0MODC	R/W		√		00H
00200078H	UF0 模式状态寄存器	UF0MODS	R		√		00H

(2/2)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200080H	UF0 有效接口编号寄存器	UF0AIFN	R/W		√		00H
00200082H	UF0 有效复用设置寄存器	UF0AAS	R/W		√		00H
00200084H	UF0 复用设置状态寄存器	UF0ASS	R		√		00H
00200086H	UF0 端点 1 接口映射寄存器	UF0E1IM	R/W		√		00H
00200088H	UF0 端点 2 接口映射寄存器	UF0E2IM	R/W		√		00H
0020008AH	UF0 端点 3 接口映射寄存器	UF0E3IM	R/W		√		00H
0020008CH	UF0 端点 4 接口映射寄存器	UF0E4IM	R/W		√		00H
00200092H	UF0 端点 7 接口映射寄存器	UF0E7IM	R/W		√		00H

(2) EPC 数据保持寄存器

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200100 H	UF0 EP0 读寄存器	UF0E0R	R		√		不确定
00200102H	UF0 EP0 长度寄存器	UF0E0L	R		√		00H
00200104H	UF0 EP0 建立寄存器	UF0E0ST	R		√		00H
00200106H	UF0 EP0 写寄存器	UF0E0W	W		√		不确定
00200108H	UF0 批量-out 1 寄存器	UF0BO1	R		√		不确定
0020010AH	UF0 批量-out 1 长度寄存器	UF0BO1L	R		√		00H
0020010CH	UF0 批量-out 2 寄存器	UF0BO2	R		√		不确定
0020010EH	UF0 批量-out 2 长度寄存器	UF0BO2L	R		√		00H
00200110H	UF0 批量-in 1 寄存器	UF0BI1	W		√		不确定
00200112H	UF0 批量-in 2 寄存器	UF0BI2	W		√		不确定
00200114H	UF0 中断 1 寄存器	UF0INT1	W		√		不确定

(3) EPC 请求数据寄存器

(1/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200144H	UF0 设备状态寄存器 L	UF0DSTL	R/W		√		00H
0020014CH	UF0 EP0 状态寄存器 L	UF0E0SL	R/W		√		00H
00200150H	UF0 EP1 状态寄存器 L	UF0E1SL	R/W		√		00H
00200154H	UF0 EP2 状态寄存器 L	UF0E2SL	R/W		√		00H
00200158H	UF0 EP3 状态寄存器 L	UF0E3SL	R/W		√		00H
0020015CH	UF0 EP4 状态寄存器 L	UF0E4SL	R/W		√		00H
00200168H	UF0 EP7 状态寄存器 L	UF0E7SL	R/W		√		00H
00200180H	UF0 地址寄存器	UF0ADRS	R		√		00H
00200182H	UF0 配置寄存器	UF0CNF	R		√		00H
00200184H	UF0 接口 0 寄存器	UF0IF0	R		√		00H
00200186H	UF0 接口 1 寄存器	UF0IF1	R		√		00H
00200188H	UF0 接口 2 寄存器	UF0IF2	R		√		00H
0020018AH	UF0 接口 3 寄存器	UF0IF3	R		√		00H
0020018CH	UF0 接口 4 寄存器	UF0IF4	R		√		00H
002001A0H	UF0 描述符长度寄存器	UF0DSCL	R/W		√		00H
002001A2H	UF0 设备描述符寄存器 0	UF0DD0	R/W		√		不确定
002001A4H	UF0 设备描述符寄存器 1	UF0DD1	R/W		√		不确定
002001A6H	UF0 设备描述符寄存器 2	UF0DD2	R/W		√		不确定
002001A8H	UF0 设备描述符寄存器 3	UF0DD3	R/W		√		不确定
002001AAH	UF0 设备描述符寄存器 4	UF0DD4	R/W		√		不确定
002001ACH	UF0 设备描述符寄存器 5	UF0DD5	R/W		√		不确定
002001AEH	UF0 设备描述符寄存器 6	UF0DD6	R/W		√		不确定
002001B0H	UF0 设备描述符寄存器 7	UF0DD7	R/W		√		不确定
002001B2H	UF0 设备描述符寄存器 8	UF0DD8	R/W		√		不确定
002001B4H	UF0 设备描述符 寄存器 9	UF0DD9	R/W		√		不确定
002001B6H	UF0 设备描述符寄存器 10	UF0DD10	R/W		√		不确定
002001B8H	UF0 设备描述符寄存器 11	UF0DD11	R/W		√		不确定
002001BAH	UF0 设备描述符寄存器 12	UF0DD12	R/W		√		不确定
002001BCH	UF0 设备描述符寄存器 13	UF0DD13	R/W		√		不确定
002001BEH	UF0 设备描述符寄存器 14	UF0DD14	R/W		√		不确定
002001C0H	UF0 设备描述符寄存器 15	UF0DD15	R/W		√		不确定
002001C2H	UF0 设备描述符寄存器 16	UF0DD16	R/W		√		不确定
002001C4H	UF0 设备描述符寄存器 17	UF0DD17	R/W		√		不确定
002001C6H	UF0 配置/接口/端点描述符寄存器 0	UF0CIE0	R/W		√		不确定
002001C8H	UF0 配置/接口/端点描述符寄存器 1	UF0CIE1	R/W		√		不确定
002001CAH	UF0 配置/接口/端点描述符寄存器 2	UF0CIE2	R/W		√		不确定
002001CCH	UF0 配置/接口/端点描述符寄存器 3	UF0CIE3	R/W		√		不确定

(2/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
002001CEH	UF0 配置/接口/端点描述符寄存器 4	UF0CIE4	R/W		√		不确定
002001D0H	UF0 配置/接口/端点描述符寄存器 5	UF0CIE5	R/W		√		不确定
002001D2H	UF0 配置/接口/端点描述符 寄存器 6	UF0CIE6	R/W		√		不确定
002001D4H	UF0 配置/接口/端点描述符寄存器 7	UF0CIE7	R/W		√		不确定
002001D6H	UF0 配置/接口/端点描述符寄存器 8	UF0CIE8	R/W		√		不确定
002001D8H	UF0 配置/接口/端点描述符寄存器 9	UF0CIE9	R/W		√		不确定
002001DAH	UF0 配置/接口/端点描述符寄存器 10	UF0CIE10	R/W		√		不确定
002001DCH	UF0 配置/接口/端点描述符寄存器 11	UF0CIE11	R/W		√		不确定
002001DEH	UF0 配置/接口/端点描述符寄存器 12	UF0CIE12	R/W		√		不确定
002001E0H	UF0 配置/接口/端点描述符寄存器 13	UF0CIE13	R/W		√		不确定
002001E2H	UF0 配置/接口/端点描述符寄存器 14	UF0CIE14	R/W		√		不确定
002001E4H	UF0 配置/接口/端点描述符寄存器 15	UF0CIE15	R/W		√		不确定
002001E6H	UF0 配置/接口/端点描述符寄存器 16	UF0CIE16	R/W		√		不确定
002001E8H	UF0 配置/接口/端点描述符寄存器 17	UF0CIE17	R/W		√		不确定
002001EAH	UF0 配置/接口/端点描述符寄存器 18	UF0CIE18	R/W		√		不确定
002001ECH	UF0 配置/接口/端点描述符寄存器 19	UF0CIE19	R/W		√		不确定
002001EEH	UF0 配置/接口/端点描述符寄存器 20	UF0CIE20	R/W		√		不确定
002001F0H	UF0 配置/接口/端点描述符寄存器 21	UF0CIE21	R/W		√		不确定
002001F2H	UF0 配置/接口/端点描述符 寄存器 22	UF0CIE22	R/W		√		不确定
002001F4H	UF0 配置/接口/端点描述符 寄存器 23	UF0CIE23	R/W		√		不确定
002001F6H	UF0 配置/接口/端点描述符 寄存器 24	UF0CIE24	R/W		√		不确定
002001F8H	UF0 配置/接口/端点描述符 寄存器 25	UF0CIE25	R/W		√		不确定

(3/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
002001FAH	UF0 配置/接口/端点描述符 寄存器 26	UF0CIE26	R/W		√		不确定
002001FCH	UF0 配置/接口/端点描述符 寄存器 27	UF0CIE27	R/W		√		不确定
002001FEH	UF0 配置/接口/端点描述符 寄存器 28	UF0CIE28	R/W		√		不确定
00200200H	UF0 配置/接口/端点描述符 寄存器 29	UF0CIE29	R/W		√		不确定
00200202H	UF0 配置/接口/端点描述符 寄存器 30	UF0CIE30	R/W		√		不确定
00200204H	UF0 配置/接口/端点描述符 寄存器 31	UF0CIE31	R/W		√		不确定
00200206H	UF0 配置/接口/端点描述符 寄存器 32	UF0CIE32	R/W		√		不确定
00200208H	UF0 配置/接口/端点描述符 寄存器 33	UF0CIE33	R/W		√		不确定
0020020AH	UF0 配置/接口/端点描述符 寄存器 34	UF0CIE34	R/W		√		不确定
0020020CH	UF0 配置/接口/端点描述符 寄存器 35	UF0CIE35	R/W		√		不确定
0020020EH	UF0 配置/接口/端点描述符 寄存器 36	UF0CIE36	R/W		√		不确定
00200210H	UF0 配置/接口/端点描述符 寄存器 37	UF0CIE37	R/W		√		不确定
00200212H	UF0 配置/接口/端点描述符 寄存器 38	UF0CIE38	R/W		√		不确定
00200214H	UF0 配置/接口/端点描述符 寄存器 39	UF0CIE39	R/W		√		不确定
00200216H	UF0 配置/接口/端点描述符 寄存器 40	UF0CIE40	R/W		√		不确定
00200218H	UF0 配置/接口/端点描述符 寄存器 41	UF0CIE41	R/W		√		不确定
0020021AH	UF0 配置/接口/端点描述符 寄存器 42	UF0CIE42	R/W		√		不确定
0020021CH	UF0 配置/接口/端点描述符 寄存器 43	UF0CIE43	R/W		√		不确定
0020021EH	UF0 配置/接口/端点描述符 寄存器 44	UF0CIE44	R/W		√		不确定
00200220H	UF0 配置/接口/端点描述符 寄存器 45	UF0CIE45	R/W		√		不确定
00200222H	UF0 配置/接口/端点描述符 寄存器 46	UF0CIE46	R/W		√		不确定
00200224H	UF0 配置/接口/端点描述符 寄存器 47	UF0CIE47	R/W		√		不确定

(4/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200226H	UF0 配置/接口/端点描述符 寄存器 48	UF0CIE48	R/W		√		不确定
00200228H	UF0 配置/接口/端点描述符 寄存器 49	UF0CIE49	R/W		√		不确定
0020022AH	UF0 配置/接口/端点描述符 寄存器 50	UF0CIE50	R/W		√		不确定
0020022CH	UF0 配置/接口/端点描述符 寄存器 51	UF0CIE51	R/W		√		不确定
0020022EH	UF0 配置/接口/端点描述符 寄存器 52	UF0CIE52	R/W		√		不确定
00200230H	UF0 配置/接口/端点描述符 寄存器 53	UF0CIE53	R/W		√		不确定
00200232H	UF0 配置/接口/端点描述符 寄存器 54	UF0CIE54	R/W		√		不确定
00200234H	UF0 配置/接口/端点描述符 寄存器 55	UF0CIE55	R/W		√		不确定
00200236H	UF0 配置/接口/端点描述符 寄存器 56	UF0CIE56	R/W		√		不确定
00200238H	UF0 配置/接口/端点描述符 寄存器 57	UF0CIE57	R/W		√		不确定
0020023AH	UF0 配置/接口/端点描述符 寄存器 58	UF0CIE58	R/W		√		不确定
0020023CH	UF0 配置/接口/端点描述符 寄存器 59	UF0CIE59	R/W		√		不确定
0020023EH	UF0 配置/接口/端点描述符 寄存器 60	UF0CIE60	R/W		√		不确定
00200240H	UF0 配置/接口/端点描述符 寄存器 61	UF0CIE61	R/W		√		不确定
00200242H	UF0 配置/接口/端点描述符 寄存器 62	UF0CIE62	R/W		√		不确定
00200244H	UF0 配置/接口/端点描述符 寄存器 63	UF0CIE63	R/W		√		不确定
00200246H	UF0 配置/接口/端点描述符 寄存器 64	UF0CIE64	R/W		√		不确定
00200248H	UF0 配置/接口/端点描述符 寄存器 65	UF0CIE65	R/W		√		不确定
0020024AH	UF0 配置/接口/端点描述符 寄存器 66	UF0CIE66	R/W		√		不确定
0020024CH	UF0 配置/接口/端点描述符 寄存器 67	UF0CIE67	R/W		√		不确定
0020024EH	UF0 配置/接口/端点描述符 寄存器 68	UF0CIE68	R/W		√		不确定
00200250H	UF0 配置/接口/端点描述符 寄存器 69	UF0CIE69	R/W		√		不确定

(5/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200252H	UF0 配置/接口/端点描述符 寄存器 70	UF0CIE70	R/W		√		不确定
00200254H	UF0 配置/接口/端点描述符 寄存器 71	UF0CIE71	R/W		√		不确定
00200256H	UF0 配置/接口/端点描述符 寄存器 72	UF0CIE72	R/W		√		不确定
00200258H	UF0 配置/接口/端点描述符 寄存器 73	UF0CIE73	R/W		√		不确定
0020025AH	UF0 配置/接口/端点描述符 寄存器 74	UF0CIE74	R/W		√		不确定
0020025CH	UF0 配置/接口/端点描述符 寄存器 75	UF0CIE75	R/W		√		不确定
0020025EH	UF0 配置/接口/端点描述符 寄存器 76	UF0CIE76	R/W		√		不确定
00200260H	UF0 配置/接口/端点描述符 寄存器 77	UF0CIE77	R/W		√		不确定
00200262H	UF0 配置/接口/端点描述符 寄存器 78	UF0CIE78	R/W		√		不确定
00200264H	UF0 配置/接口/端点描述符 寄存器 79	UF0CIE79	R/W		√		不确定
00200266H	UF0 配置/接口/端点描述符 寄存器 80	UF0CIE80	R/W		√		不确定
00200268H	UF0 配置/接口/端点描述符 寄存器 81	UF0CIE81	R/W		√		不确定
0020026AH	UF0 配置/接口/端点描述符 寄存器 82	UF0CIE82	R/W		√		不确定
0020026CH	UF0 配置/接口/端点描述符 寄存器 83	UF0CIE83	R/W		√		不确定
0020026EH	UF0 配置/接口/端点描述符 寄存器 84	UF0CIE84	R/W		√		不确定
00200270H	UF0 配置/接口/端点描述符 寄存器 85	UF0CIE85	R/W		√		不确定
00200272H	UF0 配置/接口/端点描述符 寄存器 86	UF0CIE86	R/W		√		不确定
00200274H	UF0 配置/接口/端点描述符 寄存器 87	UF0CIE87	R/W		√		不确定
00200276H	UF0 配置/接口/端点描述符 寄存器 88	UF0CIE88	R/W		√		不确定
00200278H	UF0 配置/接口/端点描述符 寄存器 89	UF0CIE89	R/W		√		不确定
0020027AH	UF0 配置/接口/端点描述符 寄存器 90	UF0CIE90	R/W		√		不确定
0020027CH	UF0 配置/接口/端点描述符 寄存器 91	UF0CIE91	R/W		√		不确定

(6/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
0020027EH	UF0 配置/接口/端点描述符 寄存器 92	UF0CIE92	R/W		√		不确定
00200280H	UF0 配置/接口/端点描述符 寄存器 93	UF0CIE93	R/W		√		不确定
00200282H	UF0 配置/接口/端点描述符 寄存器 94	UF0CIE94	R/W		√		不确定
00200284H	UF0 配置/接口/端点描述符 寄存器 95	UF0CIE95	R/W		√		不确定
00200286H	UF0 配置/接口/端点描述符 寄存器 96	UF0CIE96	R/W		√		不确定
00200288H	UF0 配置/接口/端点描述符 寄存器 97	UF0CIE97	R/W		√		不确定
0020028AH	UF0 配置/接口/端点描述符 寄存器 98	UF0CIE98	R/W		√		不确定
0020028CH	UF0 配置/接口/端点描述符 寄存器 99	UF0CIE99	R/W		√		不确定
0020028EH	UF0 配置/接口/端点描述符 寄存器 100	UF0CIE100	R/W		√		不确定
00200290H	UF0 配置/接口/端点描述符 寄存器 101	UF0CIE101	R/W		√		不确定
00200292H	UF0 配置/接口/端点描述符 寄存器 102	UF0CIE102	R/W		√		不确定
00200294H	UF0 配置/接口/端点描述符 寄存器 103	UF0CIE103	R/W		√		不确定
00200296H	UF0 配置/接口/端点描述符 寄存器 104	UF0CIE104	R/W		√		不确定
00200298H	UF0 配置/接口/端点描述符 寄存器 105	UF0CIE105	R/W		√		不确定
0020029AH	UF0 配置/接口/端点描述符 寄存器 106	UF0CIE106	R/W		√		不确定
0020029CH	UF0 配置/接口/端点描述符 寄存器 107	UF0CIE107	R/W		√		不确定
0020029EH	UF0 配置/接口/端点描述符 寄存器 108	UF0CIE108	R/W		√		不确定
002002A0H	UF0 配置/接口/端点描述符 寄存器 109	UF0CIE109	R/W		√		不确定
002002A2H	UF0 配置/接口/端点描述符 寄存器 110	UF0CIE110	R/W		√		不确定
002002A4H	UF0 配置/接口/端点描述符 寄存器 111	UF0CIE111	R/W		√		不确定
002002A6H	UF0 配置/接口/端点描述符 寄存器 112	UF0CIE112	R/W		√		不确定
002002A8H	UF0 配置/接口/端点描述符 寄存器 113	UF0CIE113	R/W		√		不确定

(7/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
002002AAH	UF0 配置/接口/端点描述符 寄存器 114	UF0CIE114	R/W		√		不确定
002002ACH	UF0 配置/接口/端点描述符 寄存器 115	UF0CIE115	R/W		√		不确定
002002AEH	UF0 配置/接口/端点描述符 寄存器 116	UF0CIE116	R/W		√		不确定
002002B0H	UF0 配置/接口/端点描述符 寄存器 117	UF0CIE117	R/W		√		不确定
002002B2H	UF0 配置/接口/端点描述符 寄存器 118	UF0CIE118	R/W		√		不确定
002002B4H	UF0 配置/接口/端点描述符 寄存器 119	UF0CIE119	R/W		√		不确定
002002B6H	UF0 配置/接口/端点描述符 寄存器 120	UF0CIE120	R/W		√		不确定
002002B8H	UF0 配置/接口/端点描述符 寄存器 121	UF0CIE121	R/W		√		不确定
002002BAH	UF0 配置/接口/端点描述符 寄存器 122	UF0CIE122	R/W		√		不确定
002002BCH	UF0 配置/接口/端点描述符 寄存器 123	UF0CIE123	R/W		√		不确定
002002BEH	UF0 配置/接口/端点描述符 寄存器 124	UF0CIE124	R/W		√		不确定
002002C0H	UF0 配置/接口/端点描述符 寄存器 125	UF0CIE125	R/W		√		不确定
002002C2H	UF0 配置/接口/端点描述符 寄存器 126	UF0CIE126	R/W		√		不确定
002002C4H	UF0 配置/接口/端点描述符 寄存器 127	UF0CIE127	R/W		√		不确定
002002C6H	UF0 配置/接口/端点描述符 寄存器 128	UF0CIE128	R/W		√		不确定
002002C8H	UF0 配置/接口/端点描述符 寄存器 129	UF0CIE129	R/W		√		不确定
002002CAH	UF0 配置/接口/端点描述符 寄存器 130	UF0CIE130	R/W		√		不确定
002002CCH	UF0 配置/接口/端点描述符 寄存器 131	UF0CIE131	R/W		√		不确定
002002CEH	UF0 配置/接口/端点描述符 寄存器 132	UF0CIE132	R/W		√		不确定
002002D0H	UF0 配置/接口/端点描述符 寄存器 133	UF0CIE133	R/W		√		不确定
002002D2H	UF0 配置/接口/端点描述符 寄存器 134	UF0CIE134	R/W		√		不确定
002002D4H	UF0 配置/接口/端点描述符 寄存器 135	UF0CIE135	R/W		√		不确定

(8/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
002002D6H	UF0 配置/接口/端点描述符 寄存器 136	UF0CIE136	R/W		√		不确定
002002D8H	UF0 配置/接口/端点描述符 寄存器 137	UF0CIE137	R/W		√		不确定
002002DAH	UF0 配置/接口/端点描述符 寄存器 138	UF0CIE138	R/W		√		不确定
002002DCH	UF0 配置/接口/端点描述符 寄存器 139	UF0CIE139	R/W		√		不确定
002002DEH	UF0 配置/接口/端点描述符 寄存器 140	UF0CIE140	R/W		√		不确定
002002E0H	UF0 配置/接口/端点描述符 寄存器 141	UF0CIE141	R/W		√		不确定
002002E2H	UF0 配置/接口/端点描述符 寄存器 142	UF0CIE142	R/W		√		不确定
002002E4H	UF0 配置/接口/端点描述符 寄存器 143	UF0CIE143	R/W		√		不确定
002002E6H	UF0 配置/接口/端点描述符 寄存器 144	UF0CIE144	R/W		√		不确定
002002E8H	UF0 配置/接口/端点描述符 寄存器 145	UF0CIE145	R/W		√		不确定
002002EAH	UF0 配置/接口/端点描述符 寄存器 146	UF0CIE146	R/W		√		不确定
002002ECH	UF0 配置/接口/端点描述符 寄存器 147	UF0CIE147	R/W		√		不确定
002002EEH	UF0 配置/接口/端点描述符 寄存器 148	UF0CIE148	R/W		√		不确定
002002F0H	UF0 配置/接口/端点描述符 寄存器 149	UF0CIE149	R/W		√		不确定
002002F2H	UF0 配置/接口/端点描述符 寄存器 150	UF0CIE150	R/W		√		不确定
002002F4H	UF0 配置/接口/端点描述符 寄存器 151	UF0CIE151	R/W		√		不确定
002002F6H	UF0 配置/接口/端点描述符 寄存器 152	UF0CIE152	R/W		√		不确定
002002F8H	UF0 配置/接口/端点描述符 寄存器 153	UF0CIE153	R/W		√		不确定
002002FAH	UF0 配置/接口/端点描述符 寄存器 154	UF0CIE154	R/W		√		不确定
002002FCH	UF0 配置/接口/端点描述符 寄存器 155	UF0CIE155	R/W		√		不确定
002002FEH	UF0 配置/接口/端点描述符 寄存器 156	UF0CIE156	R/W		√		不确定
00200300H	UF0 配置/接口/端点描述符 寄存器 157	UF0CIE157	R/W		√		不确定

(9/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200302H	UF0 配置/接口/端点描述符 寄存器 158	UF0CIE158	R/W		√		不确定
00200304H	UF0 配置/接口/端点描述符 寄存器 159	UF0CIE159	R/W		√		不确定
00200306H	UF0 配置/接口/端点描述符 寄存器 160	UF0CIE160	R/W		√		不确定
00200308H	UF0 配置/接口/端点描述符 寄存器 161	UF0CIE161	R/W		√		不确定
0020030AH	UF0 配置/接口/端点描述符 寄存器 162	UF0CIE162	R/W		√		不确定
0020030CH	UF0 配置/接口/端点描述符 寄存器 163	UF0CIE163	R/W		√		不确定
0020030EH	UF0 配置/接口/端点描述符 寄存器 164	UF0CIE164	R/W		√		不确定
00200310H	UF0 配置/接口/端点描述符 寄存器 165	UF0CIE165	R/W		√		不确定
00200312H	UF0 配置/接口/端点描述符 寄存器 166	UF0CIE166	R/W		√		不确定
00200314H	UF0 配置/接口/端点描述符 寄存器 167	UF0CIE167	R/W		√		不确定
00200316H	UF0 配置/接口/端点描述符 寄存器 168	UF0CIE168	R/W		√		不确定
00200318H	UF0 配置/接口/端点描述符 寄存器 169	UF0CIE169	R/W		√		不确定
0020031AH	UF0 配置/接口/端点描述符 寄存器 170	UF0CIE170	R/W		√		不确定
0020031CH	UF0 配置/接口/端点描述符 寄存器 171	UF0CIE171	R/W		√		不确定
0020031EH	UF0 配置/接口/端点描述符 寄存器 172	UF0CIE172	R/W		√		不确定
00200320H	UF0 配置/接口/端点描述符 寄存器 173	UF0CIE173	R/W		√		不确定
00200322H	UF0 配置/接口/端点描述符 寄存器 174	UF0CIE174	R/W		√		不确定
00200324H	UF0 配置/接口/端点描述符 寄存器 175	UF0CIE175	R/W		√		不确定
00200326H	UF0 配置/接口/端点描述符 寄存器 176	UF0CIE176	R/W		√		不确定
00200328H	UF0 配置/接口/端点描述符 寄存器 177	UF0CIE177	R/W		√		不确定
0020032AH	UF0 配置/接口/端点描述符 寄存器 178	UF0CIE178	R/W		√		不确定
0020032CH	UF0 配置/接口/端点描述符 寄存器 179	UF0CIE179	R/W		√		不确定

(10/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
0020032EH	UF0 配置/接口/端点描述符 寄存器 180	UF0CIE180	R/W		√		不确定
00200330H	UF0 配置/接口/端点描述符 寄存器 181	UF0CIE181	R/W		√		不确定
00200332H	UF0 配置/接口/端点描述符 寄存器 182	UF0CIE182	R/W		√		不确定
00200334H	UF0 配置/接口/端点描述符 寄存器 183	UF0CIE183	R/W		√		不确定
00200336H	UF0 配置/接口/端点描述符 寄存器 184	UF0CIE184	R/W		√		不确定
00200338H	UF0 配置/接口/端点描述符 寄存器 185	UF0CIE185	R/W		√		不确定
0020033AH	UF0 配置/接口/端点描述符 寄存器 186	UF0CIE186	R/W		√		不确定
0020033CH	UF0 配置/接口/端点描述符 寄存器 187	UF0CIE187	R/W		√		不确定
0020033EH	UF0 配置/接口/端点描述符 寄存器 188	UF0CIE188	R/W		√		不确定
00200340H	UF0 配置/接口/端点描述符 寄存器 189	UF0CIE189	R/W		√		不确定
00200342H	UF0 配置/接口/端点描述符 寄存器 190	UF0CIE190	R/W		√		不确定
00200344H	UF0 配置/接口/端点描述符 寄存器 191	UF0CIE191	R/W		√		不确定
00200346H	UF0 配置/接口/端点描述符 寄存器 192	UF0CIE192	R/W		√		不确定
00200348H	UF0 配置/接口/端点描述符 寄存器 193	UF0CIE193	R/W		√		不确定
0020034AH	UF0 配置/接口/端点描述符 寄存器 194	UF0CIE194	R/W		√		不确定
0020034CH	UF0 配置/接口/端点描述符 寄存器 195	UF0CIE195	R/W		√		不确定
0020034EH	UF0 配置/接口/端点描述符 寄存器 196	UF0CIE196	R/W		√		不确定
00200350H	UF0 配置/接口/端点描述符 寄存器 197	UF0CIE197	R/W		√		不确定
00200352H	UF0 配置/接口/端点描述符 寄存器 198	UF0CIE198	R/W		√		不确定
00200354H	UF0 配置/接口/端点描述符 寄存器 199	UF0CIE199	R/W		√		不确定
00200356H	UF0 配置/接口/端点描述符 寄存器 200	UF0CIE200	R/W		√		不确定
00200358H	UF0 配置/接口/端点描述符 寄存器 201	UF0CIE201	R/W		√		不确定

(11/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
0020035AH	UF0 配置/接口/端点描述符 寄存器 202	UF0CIE202	R/W		√		不确定
0020035CH	UF0 配置/接口/端点描述符 寄存器 203	UF0CIE203	R/W		√		不确定
0020035EH	UF0 配置/接口/端点描述符 寄存器 204	UF0CIE204	R/W		√		不确定
00200360H	UF0 配置/接口/端点描述符 寄存器 205	UF0CIE205	R/W		√		不确定
00200362H	UF0 配置/接口/端点描述符 寄存器 206	UF0CIE206	R/W		√		不确定
00200364H	UF0 配置/接口/端点描述符 寄存器 207	UF0CIE207	R/W		√		不确定
00200366H	UF0 配置/接口/端点描述符 寄存器 208	UF0CIE208	R/W		√		不确定
00200368H	UF0 配置/接口/端点描述符 寄存器 209	UF0CIE209	R/W		√		不确定
0020036AH	UF0 配置/接口/端点描述符 寄存器 210	UF0CIE210	R/W		√		不确定
0020036CH	UF0 配置/接口/端点描述符 寄存器 211	UF0CIE211	R/W		√		不确定
0020036EH	UF0 配置/接口/端点描述符 寄存器 212	UF0CIE212	R/W		√		不确定
00200370H	UF0 配置/接口/端点描述符 寄存器 213	UF0CIE213	R/W		√		不确定
00200372H	UF0 配置/接口/端点描述符 寄存器 214	UF0CIE214	R/W		√		不确定
00200374H	UF0 配置/接口/端点描述符 寄存器 215	UF0CIE215	R/W		√		不确定
00200376H	UF0 配置/接口/端点描述符 寄存器 216	UF0CIE216	R/W		√		不确定
00200378H	UF0 配置/接口/端点描述符 寄存器 217	UF0CIE217	R/W		√		不确定
0020037AH	UF0 配置/接口/端点描述符 寄存器 218	UF0CIE218	R/W		√		不确定
0020037CH	UF0 配置/接口/端点描述符 寄存器 219	UF0CIE219	R/W		√		不确定
0020037EH	UF0 配置/接口/端点描述符 寄存器 220	UF0CIE220	R/W		√		不确定
00200380H	UF0 配置/接口/端点描述符 寄存器 221	UF0CIE221	R/W		√		不确定
00200382H	UF0 配置/接口/端点描述符 寄存器 222	UF0CIE222	R/W		√		不确定
00200384H	UF0 配置/接口/端点描述符 寄存器 223	UF0CIE223	R/W		√		不确定

(12/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200386H	UF0 配置/接口/端点描述符 寄存器 224	UF0CIE224	R/W		√		不确定
00200388H	UF0 配置/接口/端点描述符 寄存器 225	UF0CIE225	R/W		√		不确定
0020038AH	UF0 配置/接口/端点描述符 寄存器 226	UF0CIE226	R/W		√		不确定
0020038CH	UF0 配置/接口/端点描述符 寄存器 227	UF0CIE227	R/W		√		不确定
0020038EH	UF0 配置/接口/端点描述符 寄存器 228	UF0CIE228	R/W		√		不确定
00200390H	UF0 配置/接口/端点描述符 寄存器 229	UF0CIE229	R/W		√		不确定
00200392H	UF0 配置/接口/端点描述符 寄存器 230	UF0CIE230	R/W		√		不确定
00200394H	UF0 配置/接口/端点描述符 寄存器 231	UF0CIE231	R/W		√		不确定
00200396H	UF0 配置/接口/端点描述符 寄存器 232	UF0CIE232	R/W		√		不确定
00200398H	UF0 配置/接口/端点描述符 寄存器 233	UF0CIE233	R/W		√		不确定
0020039AH	UF0 配置/接口/端点描述符 寄存器 234	UF0CIE234	R/W		√		不确定
0020039CH	UF0 配置/接口/端点描述符 寄存器 235	UF0CIE235	R/W		√		不确定
0020039EH	UF0 配置/接口/端点描述符 寄存器 236	UF0CIE236	R/W		√		不确定
002003A0H	UF0 配置/接口/端点描述符 寄存器 237	UF0CIE237	R/W		√		不确定
002003A2H	UF0 配置/接口/端点描述符 寄存器 238	UF0CIE238	R/W		√		不确定
002003A4H	UF0 配置/接口/端点描述符 寄存器 239	UF0CIE239	R/W		√		不确定
002003A6H	UF0 配置/接口/端点描述符 寄存器 240	UF0CIE240	R/W		√		不确定
002003A8H	UF0 配置/接口/端点描述符 寄存器 241	UF0CIE241	R/W		√		不确定
002003AAH	UF0 配置/接口/端点描述符 寄存器 242	UF0CIE242	R/W		√		不确定
002003ACH	UF0 配置/接口/端点描述符 寄存器 243	UF0CIE243	R/W		√		不确定
002003AEH	UF0 配置/接口/端点描述符 寄存器 244	UF0CIE244	R/W		√		不确定
002003B0H	UF0 配置/接口/端点描述符 寄存器 245	UF0CIE245	R/W		√		不确定

(13/13)

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
002003B2H	UF0 配置/接口/端点描述符 寄存器 246	UF0CIE246	R/W		√		不确定
002003B4H	UF0 配置/接口/端点描述符 寄存器 247	UF0CIE247	R/W		√		不确定
002003B6H	UF0 配置/接口/端点描述符 寄存器 248	UF0CIE248	R/W		√		不确定
002003B8H	UF0 配置/接口/端点描述符 寄存器 249	UF0CIE249	R/W		√		不确定
002003BAH	UF0 配置/接口/端点描述符 寄存器 250	UF0CIE250	R/W		√		不确定
002003BCH	UF0 配置/接口/端点描述符 寄存器 251	UF0CIE251	R/W		√		不确定
002003BEH	UF0 配置/接口/端点描述符 寄存器 252	UF0CIE252	R/W		√		不确定
002003C0H	UF0 配置/接口/端点描述符 寄存器 253	UF0CIE253	R/W		√		不确定
002003C2H	UF0 配置/接口/端点描述符 寄存器 254	UF0CIE254	R/W		√		不确定
002003C4H	UF0 配置/接口/端点描述符 寄存器 255	UF0CIE255	R/W		√		不确定

(4) 桥寄存器

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200400H	桥中断控制寄存器	BRGINTT	R/W			√	0000H
00200402H	桥中断使能寄存器	BRGINTE	R/W			√	0000H
00200404H	EPC 模块控制寄存器	EPCCLT	R/W			√	0000H
00200408H	CPU I/F 总线控制寄存器	CPUBCTL	R/W			√	0000H

(5) DMA 寄存器

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00200500H	EP1 DMA 控制寄存器 1	UF0E1DC1	R/W			√	0000H
00200502H	EP1 DMA 控制寄存器 2	UF0E1DC2	R/W			√	0000H
00200504H	EP2 DMA 控制寄存器 1	UF0E2DC1	R/W			√	0000H
00200506H	EP2 DMA 控制寄存器 2	UF0E2DC2	R/W			√	0000H
00200508H	EP3 DMA 控制寄存器 1	UF0E3DC1	R/W			√	0000H
0020050AH	EP3 DMA 控制寄存器 2	UF0E3DC2	R/W			√	0000H
0020050CH	EP4 DMA 控制寄存器 1	UF0E4DC1	R/W			√	0000H
0020050EH	EP4 DMA 控制寄存器 2	UF0E4DC2	R/W			√	0000H

(6) 批量-in 寄存器

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00201000H	UF0 EP1 批量-in 传输数据寄存器	UF0EP1BI	W			√	0000H
00202000H	UF0 EP3 批量-in 传输数据寄存器	UF0EP3BI	W			√	0000H

(7) 批量-out 寄存器

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00210000H	UF0 EP2 批量-out 传输数据寄存器	UF0EP2BO	R		√	√	0000H
00220000H	UF0 EP4 批量-out 传输数据寄存器	UF0EP4BO	R		√	√	0000H

(8) 外设控制寄存器

地址	功能寄存器名称	符号	R/W	可操作位数			默认值
				1	8	16	
00240000H	USBF DMA 请求使能寄存器	UFDRQEN	R/W		√	√	0000H

20.6.3 EPC控制寄存器

（1）UF0 EP0NAK 寄存器（UF0E0N）

该寄存器用于控制端点 0 的 NAK （自动执行的请求除外）。

该寄存器可以按字节进行读取或写入（但是，第 0 位只能被读取）。

在 UF0FIC0 和 UF0FIC1 寄存器被置位后，需要五个 USB 时钟才能在该寄存器上反映出状态。因此，如果需要正确读取状态，访问 UF0FIC0 和 UF0FIC1 寄存器的写信号与访问 UF0EPS0、UF0EPS1、UF0EPS2、UF0E0N 和 UF0EN 寄存器的读信号要进行分离，至少间隔四个 USB 时钟。

当 NAK 被发送到端点 0 读，端点 2 和端点 4 时，对 EP0NKR 位的写入访问被忽略。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0N	0	0	0	0	0	0	EP0NKR	EP0NKW	00200000H	00H

位的位置	位名称	功能
1	EP0NKR	该位用于控制到端点 0 的 OUT 令牌的 NAK（自动执行的请求除外）。当端点 0 正确接收到数据时，它被硬件自动设置为 1。当 UF0E0R 寄存器的数据被 FW 读取时，它也被硬件清除为 0（计数器值 = 0）。 1：发送 NAK。 0：不发送 NAK （默认值）。 由于某些原因即使 USBF 准备好接收数据，但不应该从 USB 总线接收数据时，通过 FW 置该位为 1。在这种情况下，在该位被 FW 清除为 0 之前 USBF 持续发送 NAK。UF0E0R 寄存器被清除，则该位也立即被清除为 0。
0	EP0NKW	该位表示如何控制到端点 0 的 IN 令牌的 NAK（自动执行的请求除外）。当端点 0 的数据被发送并且主机正确接收到发送的数据时，该位由硬件自动清除为 0。在该位被清除之前，UF0E0W 寄存器的数据被保留。因此，即使在主机不能正确接收数据而产生重发请求情况下，也不需要重新写入该位。要发送一个短包，确保设置 UF0DEND 寄存器的 E0DED 位为 1。当 FIFO 满时，该位被自动设置为 1。UF0DEND 寄存器的 E0DED 位被置为 1，则 EP0NKW 位被同时立即自动设置为 1。 1：不发送 NAK。 0：发送 NAK（默认值）。 在数据阶段不能正确接收 ACK 时，如果控制传输进入状态阶段，只要 UF0E0W 寄存器被清除，则该位立即被清除为 0。当 UF0E0W 由 FW 清除时，该位也被清除为 0。

下面，使用 IN / OUT 令牌的 SETUP 事务的过程解释如下。

（a）使用 IN 令牌时（硬件自动执行的请求除外）

在接收 CPUDEC 中断之后，且在从 UF0E0ST 寄存器读取数据之前，应该使用 FW 来清除 UF0IS1 寄存器的 PROT 位为 0。然后，根据请求来执行处理，并且如果需要由 IN 令牌返回数据，向 UF0E0W 寄存器写入数据。在写入完成之后，确认 UF0IS1 寄存器的 PROT 位是 0，并且设置 UF0DEND 寄存器的 E0DED 位为 1。在 EP0NKW 位被置为 1 之后的第一个 IN 令牌时硬件发送数据。如果 UF0IS1 寄存器的 PROT 位为 1，表明在控制传输完成之前已经再次发生 SETUP 事务。在这种情况下，通过清除 UF0IC1 寄存器的 PROTC 位为 0 来清除 UF0IS1 寄存器的 PROT 位为 0，然后再次从 UF0E0ST 寄存器读取数据。随后接收到的请求可以被读取。

（b）使用 OUT 令牌时（硬件自动执行的请求除外）

在接收 CPUDEC 中断之后，且在从 UF0E0ST 寄存器读取数据之前，应该使用 FW 来清除 UF0IS1 寄存器的 PROT 位。在从 UF0E0R 寄存器读取数据之前，确认 UF0IS1 寄存器的 PROT 位是 0。如果 PROT 是 1，意味着无效数据被保留。通过 FW（EP0NKR 位自动清除为 0）清除 FIFO。如果 UF0IS1 寄存器的 PROT 位为 0，读取 UF0E0L 寄存器的数据长度，并按照设定数量从 UF0E0R 寄存器读取足够的数据。当从 UF0E0R 寄存器读取数据完成时（当 UF0E0R 寄存器的计数器被清除为 0 时），硬件自动清除 EP0NKR 位为 0。

（2）UF0 EP0NAKALL 寄存器（UF0E0NA）

该寄存器用于控制到端点 0 的所有请求的 NAK。对于自动执行的请求，它也同样有效。

该寄存器可以按字节进行读取或写入。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0NA	0	0	0	0	0	0	0	EP0NKA	00200002H	00H

位的位置	位名称	功能
0	EP0NKA	该位控制到端点 0 的 SETUP 事务以外的 NAK （包含自动执行的请求）。该位由 FW 操作。 1：发送 NAK。 0：不发送 NAK （默认值）。 当用于自动执行的请求的数据被更改时，该寄存器用于防止 FW 的写入访问与从 SIE 读取访问之间的冲突。当执行从 SIE 的访问时，推迟从 FW 的写入访问到该位的反映。在从 FW 重新写入请求数据寄存器之前，确认该位被正确置 1。 只有在以下情况，该位置 1 会被反映。 • USBF 刚复位后，并且从未接收过 SETUP 令牌 • 总线复位刚接收后，并且从未接收过 SETUP 令牌 • 检测到 SETUP 令牌的 PID • 阶段被更改为状态阶段 清除该位为 0 会立即被反映，除非正在接收 IN 令牌并且正在发生 NAK 响应。 端点 0 传输期间，上述四种情况下，设置 EP0NKA 位为 1 会被反映，但是当端点 0 没有传输数据时，写入该位后会立即被反映。

（3）UF0 EPNAK 寄存器（UF0EN）

该寄存器用于控制端点 0 以外其它端点的 NAK。

该寄存器可以按字节进行读取或写入（但是，第 5 位，第 4 位，第 1 位和第 0 位只能被读取）。

只有当 UF0ENM 寄存器的 BKO2NKM 位为 1 时，BKO2NK 位才可以被写入。只有当 UF0ENM 寄存器的 BKO1NKM 位为 1 时，BKO1NK 位才可以被写入。

如果 UF0EnIM 寄存器（n = 1 至 4，7）的设置与接口的当前设置不支持各个端点，相关的位无效。

在 UF0FIC0 和 UF0FIC1 寄存器被置位后，需要五个 USB 时钟才能在该寄存器上反映出状态。因此，如果需要正确读取状态，访问 UF0FIC0 和 UF0FIC1 寄存器的写信号与访问 UF0EPS0、UF0EPS1、UF0EPS2、UF0E0N 和 UF0EN 寄存器的读信号要进行分离，至少间隔四个 USB 时钟。

当 NAK 被发送到端点 0 读，端点 2 和端点 4 时，对 BKO1NK 和 BKO2NK 位的写入访问被忽略。

请确保清除第 7 位至第 5 位为“0”。如果这些位被置为 1，操作无法保证。

（1/4）

	7	6	5	4	3	2	1	0	地址	复位后
UF0EN	0	0	0	IT1NK	BKO2NK	BKO1NK	BKI2NK	BKI1NK	00200004H	00H

位的位置	位名称	功能
4	IT1NK	该位用于控制到端点 7 的 NAK（中断 1 传输）。 写入数据到 UF0INT1 寄存器导致该寄存器满，该位被自动置 1 且启动传输。发送一个不会占满 FIFO 的短包，设置 UF0DEND 寄存器的 IT1DEND 位为 1。IT1DEND 位被置为 1，则该位立即被自动置为 1。 1：不发送 NAK。 0：发送 NAK （默认值）。 只要 UF0INT1 寄存器被清除，该位也被清除为 0。

(2/4)

位的位置	位名称	功能
3	BKO2NK	该位用于控制到端点 4 的 NAK（批量 2 传输（OUT））。 1：发送 NAK。 0：不发送 NAK （默认值）。 只有当连接到 UF0BO2 寄存器的 SIE 端的 FIFO（bank 配置的 64 字节 FIFO）不能接收数据时，该位才能被设置为 1。当执行一个切换开关操作时，它被清除为 0。当满足下列条件时，bank 被更改（切换操作）。 • 正确接收的数据存储在连接到 SIE 端的 FIFO 中。 • 连接到 CPU 端的 FIFO 计数器的值为 0（读取完成）。 当接收到 BKO2DT 中断请求时，应该使用 FW 来读取 UF0BO2L 寄存器的数据长度，并按照设定数量从 UF0BO2 寄存器读取足够的数据。如果由于某些原因不从 USB 总线接收数据，即使 USBF 已经准备好接收数据，通过 FW 设置该位为 1。在这种情况下，USBF 持续发送 NAK 直到 FW 清除该位为 0。只要清除 UF0BO2 寄存器，该位也立即被清除为 0。
2	BKO1NK	该位用于控制到端点 2 的 NAK（批量 1 传输（OUT））。 1：发送 NAK。 0：不发送 NAK （默认值）。 只有当连接到 UF0BO1 寄存器的 SIE 端的 FIFO（bank 配置的 64 字节 FIFO）不能接收数据时，该位才能被设置为 1。当执行一个切换开关操作时，它被清除为 0。当满足下列条件时，bank 被更改（切换操作）。 • 正确接收的数据存储在连接到 SIE 端的 FIFO 中。 • 连接到 CPU 端的 FIFO 计数器的值为 0（读取完成）。 当接收到 BKO1DT 中断请求时，应该使用 FW 来读取 UF0BO1L 寄存器的数据长度，并按照设定数量从 UF0BO1 寄存器读取足够的数据。如果由于某些原因不从 USB 总线接收数据，即使 USBF 已经准备好接收数据，通过 FW 设置该位为 1。在这种情况下，USBF 持续发送 NAK 直到 FW 清除该位为 0。只要清除 UF0BO1 寄存器，该位也立即被清除为 0。

注意事项 1. 如果在 PIO 模式下从 UF0BO2 读取数据时，DMA 使能，立即发出一个 DMA 请求。
2. 如果在 DMA 传输模式下读取 CPU 侧 FIFO 中的最后一个数据，DMA 请求信号变为无效。
3. 如果在 DMA 传输模式下接收到 TC 信号，DMA 请求信号变为无效。

(3/4)

位的位置	位名称	功能
1	BKI2NK	该位用于控制到端点 3 的 NAK（批量 2 传输 （IN））。 1：不发送 NAK。 0：发送 NAK （默认值）。 只有当连接到 UF0BI2 寄存器 SIE 端的 FIFO（bank 配置的 64 字节 FIFO）不能接收数据时，该位才能被清除为 0。当执行一个切换开关操作（保留 UF0BI2 寄存器的数据，直到传输已经正确完成）时，它被置为 1。当满足下列条件时，bank 被更改（切换操作）。 • 数据被正确写入到连接在 CPU 总线端的 FIFO 中（写入已经完成，并且 FIFO 为满或者 UF0DEND 寄存器被置位）。 • 连接在 SIE 端的 FIFO 计数器的值为 0。 写入数据到 FIFO 导致 CPU 端的 FIFO 满并执行 FIFO 切换操作时，该位被自动置 1 且启动数据传输。但是，如果在 UF0DEND 寄存器的 BKI2T 位被清除为 0 时，因为 DMA 写入数据到 CPU 端的 FIFO 导致 FIFO 满，不执行切换操作，因为不满足切换操作的条件，直到 UF0DEND 寄存器的 BKI2DED 位被置为 1。要发送一个不会占满 CPU 端 FIFO 的短包，在写入数据完成后设置 BKI2DED 位为 1。当 BKI2DED 位被置为 1 的同时，执行切换开关操作，则该位被自动置为 1。只要清除 UF0BI2 寄存器，该位也立即被清除为 0。

注意事项 1. 如果在 PIO 模式下向 UF0BI2 寄存器写入数据时，DMA 使能，立即发出一个 DMA 请求。

2. 如果在 DMA 传输模式下写入 64 字节数据，DMA 请求信号变为无效。如果然后 BKI2NK 位被置 1，数据和 IN 令牌同步发送。在 FIFO 切换时，只要 DMA 请求没有被屏蔽，DMA 请求信号再次变为有效。如果 BKI2NK 位没有置 1，即使接收到 IN 令牌，数据也不会被发送。这种情况下，设置 UF0DEND 寄存器的 BKI2DED 位为 1。

3. 如果在 DMA 传输模式下接收到 TC 信号，DMA 请求信号变为无效。同时 DMA 请求被屏蔽。如果 BKI2NK 位没有置 1，即使接收到 IN 令牌，数据也不会被发送。当 UF0DEND 寄存器的 BKI2DED 位被 FW 置 1 时，数据和 IN 令牌同步发送。为了再次执行 DMA 传输，取消对 DMA 请求的屏蔽。

(4/4)

位的位置	位名称	功能
0	BKI1NK	该位用于控制到端点 1 的 NAK（批量 1 传输 （IN））。 1：不发送 NAK。 0：发送 NAK （默认值）。 只有当连接到 UF0BI1 寄存器 SIE 端的 FIFO（bank 配置的 64 字节 FIFO）不能接收数据时，该位才能被清除为 0。当执行一个切换开关操作（保留 UF0BI1 寄存器的数据，直到传输已经正确完成）时，它被置为 1。当满足下列条件时，bank 被更改（切换操作）。 • 数据被正确写入到连接在 CPU 总线端的 FIFO 中（写入已经完成，并且 FIFO 为满或者 UF0DEND 寄存器被置位）。 • 连接在 SIE 端的 FIFO 计数器的值为 0。 写入数据到 FIFO 导致 CPU 端的 FIFO 满并执行 FIFO 切换操作时，该位被自动置 1 且启动数据传输。但是，如果在 UF0DEND 寄存器的 BKI1T 位被清除为 0 时，因为 DMA 写入数据到 CPU 端的 FIFO 导致 FIFO 满，不执行切换操作，因为不满足切换操作的条件，直到 UF0DEND 寄存器的 BKI1DED 位被置为 1。要发送一个不会占满 CPU 端 FIFO 的短包，在写入数据完成后设置 BKI1DED 位为 1。当 BKI1DED 位被置为 1 的同时，执行切换开关操作，则该位被自动置为 1。只要清除 UF0BI1 寄存器，该位也立即被清除为 0。

注意事项 1. 如果在 PIO 模式下向 UF0BI1 寄存器写入数据时，DMA 使能，立即发出一个 DMA 请求。

2. 如果在 DMA 传输模式下写入 64 字节数据，DMA 请求信号变为无效。如果然后 BKI1NK 位被置 1，数据和 IN 令牌同步发送。在 FIFO 切换时，只要 DMA 请求没有被屏蔽，DMA 请求信号再次变为有效。如果 BKI1NK 位没有置 1，即使接收到 IN 令牌，数据也不会被发送。这种情况下，设置 UF0DEND 寄存器的 BKI1DED 位为 1。

3. 如果在 DMA 传输模式下接收到 TC 信号，DMA 请求信号变为无效。同时 DMA 请求被屏蔽。如果 BKI1NK 位没有置 1，即使接收到 IN 令牌，数据也不会被发送。当 UF0DEND 寄存器的 BKI1DED 位被 FW 置 1 时，数据和 IN 令牌同步发送。为了再次执行 DMA 传输，取消对 DMA 请求的屏蔽。

（4）**UF0 EPNAK 屏蔽寄存器（UF0ENM）**

该寄存器用于控制对 UF0EN 寄存器写入访问的屏蔽。

该寄存器可以按字节进行读取或写入。

请确保第 7 位至第 4 位、第 1 位和第 0 位清除为“0”。如果该位被设置为 1，操作无法保证。

	7	6	5	4	3	2	1	0	地址	复位后
UF0ENM	0	0	0	0	BKO2NKM	BKO1NKM	0	0	00200006H	00H

位的位置	位名称	功能
3	BKO2NKM	该位用于指定是否屏蔽对 UF0EN 寄存器的第 3 位（BKO2NK）的写入访问。 1：不屏蔽。 0：屏蔽 （默认值）。
2	BKO1NKM	该位用于指定是否屏蔽对 UF0EN 寄存器的第 2 位（BKO1NK）的写入访问。 1：不屏蔽。 0：屏蔽 （默认值）。

（5）UF0 SNDSIE 寄存器（UF0SDS）

该寄存器执行无握手之类的操作。它可以直接操作 SIE 的引脚。

该寄存器可以按字节进行读取或写入。

确保将第 2 位清除为“0”。如果该位被设置为 1，操作无法保证。

	7	6	5	4	3	2	1	0	地址	复位后
UF0SDS	0	0	0	0	SNDSTL	0	0	RSUMIN	00200008H	00H

位的位置	位名称	功能
3	SNDSTL	该位使端点 0 发布一个 STALL 握手。如果系统不支持 CPUDEC 处理的请求，导致一个 STALL 握手响应，设置该位为 1。如果 SET_CONFIGURATION 或 SET_INTERFACE 发送一个不支持的 wValue，硬件设置该位为 1。如果由于一个自动执行的请求溢出，端点 0 发生问题，该位也被设置为 1。但是，UF0E0SL 寄存器的 E0HALT 位不被置为 1。 1： 用 STALL 握手的响应。 0： 不用 STALL 握手的响应（默认值）。 当接收到下一个 SETUP 令牌时，该位被清除为 0，并且总线的握手响应不是 STALL 。要通过 FW 将 SNDSTL 位置为 1，不要向 UF0E0W 寄存器写入数据。根据设置该位的时序，STALL 响应不会及时产生，并且可能在 NAK 响应后的下一个传输中产生。 当该位被置为 1 时，只有当 FW 执行请求正在执行时，对该位的设置才有效。当接收到下一个 SETUP 令牌时，它被自动清除为 0。 **备注** SNDSTL 位只对 FW 执行的请求有效。
0	RSUMIN	该位输出恢复（Resume）信号到 USB 总线上。除非 UF0DSTL 寄存器的 RMWK 位被置为 1，否则写入该位无效。 1： 产生恢复信号 0：不产生恢复信号（默认值） 当该位被设置为 1 时，继续产生恢复信号。在经过指定时间后通过 FW 清除该位为 0。因为信号在内部按照时钟采样，只有在提供 CLK 时，操作才能保证。当系统的 CLK 停止时，必须小心。

（6）UF0 CLR 请求寄存器（UF0CLR）

该寄存器表明接收到 CLEAR_FEATURE 请求的目标。

该寄存器是只读的，按字节进行读取。

只有产生中断请求时，该寄存器才有意义。在状态阶段完成后，每一位被设置为 1，并且当这个寄存器被读取时，自动清除为 0。

如果 UF0EnIM 寄存器（n = 1 至 4，7）的设置与接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0CLR	0	CLREP7	CLREP4	CLREP3	CLREP2	CLREP1	CLREP0	CLRDEV	0020000AH	00H

位的位置	位名称	功能
6 至 1	CLREPn	这些位表明接收到 CLEAR_FEATURE 端点 n 请求并且被自动处理。 1：已自动处理 0：未自动处理（默认值）
0	CLRDEV	该位表明接收到 CLEAR_FEATURE Device 请求并被自动处理。 1：已自动处理 0：未自动处理（默认值）

备注 n = 0 至 4，7

（7）UF0 SET 请求寄存器（UF0SET）

该寄存器表明自动处理的 SET_XXXX （SET_INTERFACE 除外）请求的目标。

该寄存器是只读的，按字节进行读取。

只有产生中断请求时，该寄存器才有意义。在状态阶段完成后，每一位都被设置为 1，并且当这个寄存器被读取时，自动清除为 0。

	7	6	5	4	3	2	1	0	地址	复位后
UF0SET	SETCON	0	0	0	0	SETEP	0	SETDEV	0020000CH	00H

位的位置	位名称	功能
7	SETCON	该位表明接收到 SET_CONFIGURATION 请求并被自动处理。 1：已自动处理 0：未自动处理（默认值）
2	SETEP	该位表明接收到 SET_FEATURE 端点 n 请求 （n = 0 至 4，7）并被自动处理。 1：已自动处理 0：未自动处理（默认值）
0	SETDEV	该位表明接收到 SET_FEATURE Device 并被自动处理。 1：已自动处理 0：未自动处理（默认值）

（8）UF0 EP 状态 0 寄存器（UF0EPS0）

该寄存器表明 USB 总线状态以及寄存器数据的存在或缺席。

该寄存器是只读的，按字节进行读取。

如果 UF0EnIM 寄存器（n = 1 至 4，7）的设置与接口的当前设置不支持各个端点，相关的位无效。

在 UF0FIC0 和 UF0FIC1 寄存器被置位后，需要五个 USB 时钟才能在该寄存器上反映出状态。因此，如果需要正确读取状态，访问 UF0FIC0 和 UF0FIC1 寄存器的写信号与访问 UF0EPS0、UF0EPS1、UF0EPS2、UF0E0N 和 UF0EN 寄存器的读信号要进行分离，至少间隔四个 USB 时钟。

(1/2)

	7	6	5	4	3	2	1	0	地址	复位后
UF0EPS0	0	IT1	BKOUT2	BKOUT1	BKIN2	BKIN1	EP0W	EP0R	0020000EH	00H

位的位置	位名称	功能
6	IT1	这些位表示数据是在 UF0INT1 寄存器 (FIFO)中。通过设置 UF0DEND 寄存器的 IT1DED 位为 1，即使没有数据写入到该寄存器（Null 数据传输），也可以创建数据存于 UF0INT1 寄存器的状态。即使 UF0INT1 寄存器的计数器为 0，只要设置 UF0DEND 寄存器的 IT1DED 位为 1，这些位就被硬件置 1。在正确传输之后，这些位被清除为 0。 1：数据在该寄存器中。 0：没有数据在该寄存器中（默认值）。
5，4	BKOUTn	这些位表明数据是在连接到 CPU 端的 UF0BOn 寄存器（FIFO）中。当 FIFO 配置的 UF0BOn 寄存器被切换时，这些位被硬件自动设置为 1。当读取连接到 CPU 端的 UF0BOn 寄存器（FIFO）完成时（计数器值 = 0），这些位被硬件自动清除为 0。当接收到空数据时（切换 FIFO 也未发生），它不被置为 1。 1：数据在该寄存器中。 0：没有数据在该寄存器中（默认值）。
3，2	BKINn	该位表明数据是在连接到 CPU 端的 UF0BIn 寄存器（FIFO）中。通过设置 UF0DEND 寄存器的 BKInDED 位为 1，即使没有数据写入到该寄存器（Null 数据发送），也可以创建数据存于 UF0BIn 寄存器的状态。当 UF0BIn 寄存器的计数器为 0 时，只要 UF0DEND 寄存器的 BKInDED 位被置为 1，这些位就被硬件置 1。当执行切换操作时，这些位被清除为 0。 1：数据在该寄存器中。 0：没有数据在该寄存器中（默认值）。

备注 n = 1，2

(2/2)

位的位置	位名称	功能
1	EP0W	该位表明数据是在 UF0E0W 寄存器（FIFO）中。通过设置 UF0DEND 寄存器的 E0DED 位为 1，即使没有数据写入到该寄存器（Null 数据发送），也可以创建数据存于 UF0E0W 寄存器的状态。当 UF0E0W 寄存器的计数器为 0 时，只要 UF0DEND 寄存器的 E0DED 位被置为 1，该位就由硬件设置为 1。在正确传输之后，该位被清除为 0。 1：数据在寄存器中。 0：没有数据在寄存器中（默认值）。
0	EP0R	该位表明数据是在 UF0E0R 寄存器（FIFO）中。当读取 UF0E0R 寄存器（FIFO） 完成后，它被硬件自动清除为 0。如果接收到 Null 数据，它被置为 1。 1：数据在寄存器中。 0：没有数据在寄存器中（默认值）。

（9）UF0 EP 状态 1 寄存器（UF0EPS1）

该寄存器表明 USB 总线状态以及寄存器数据的存在或缺席。

该寄存器是只读的，按字节进行读取。

	7	6	5	4	3	2	1	0	地址	复位后
UF0EPS1	RSUM	0	0	0	0	0	0	0	00200010H	00H

位的位置	位名称	功能
7	RSUM	该位表明 USB 总线处于恢复（Resume）状态。只有在产生中断请求时，该位才有意义。 1：挂起状态 0：恢复状态（默认值） 因为信号在内部按照时钟采样，只有提供 CLK 时，操作才能保证。当系统的 CLK 停止时，必须小心。即使 CLK 被停止，SIE 的 INTRSUM 信号也会工作。因此，通过让中断请求标志寄存器 1L （IF1L）的 RSUMIF 位变为有效，或者降低 USBF 的 CLK 的频率，可以支持该位。 当该位被读取时，自动清除为 0。

（10）UF0 EP 状态 2 寄存器（UF0EPS2）

该寄存器表明 USB 总线状态以及寄存器数据的存在或缺席。

该寄存器是只读的，按字节进行读取。

如果 UF0EnIM 寄存器（n = 1 至 4，7）的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0EPS2	0	0	HALT7	HALT4	HALT3	HALT2	HALT1	HALT0	00200012H	00H

位的位置	位名称	功能
5 至 0	HALTn	这些位表明端点 n 当前处于停顿（stall）中。当一个停顿条件满足时，例如发生溢出和接收到未定义的请求，这些位被置为 1。这些位由硬件自动设置为 1。 1：端点处于停顿中。 0：端点没有在停顿中（默认值）。 由于发生溢出或接收到未定义的请求，只要 HALT0 位被置为 1，SNDSTL 位立即被置为 1。如果在这个状态下接收到下一个 SETUP 令牌，SNDSTL 位被清除为 0，因此 HALT0 位也被清除为 0。如果端点 0 因为 SET_FEATURE 端点 0 请求而停顿，该位不会被清除为 0，直到接收到 CLEAR_FEATURE 端点 0 请求或暂停特征（Halt Feature）被 FW 清除。如果接收到 GET_STATUS 端点 0、CLEAR_FEATURE 端点 0 或 SET_FEATURE 端点 0 请求，或者如果因为 CPUDEC 中断请求而接收到要由 FW 处理的请求，HALT0 位被屏蔽并清除为 0，直到接收到下一个 SETUP 令牌。 HALTn 位不被清除为 0，直到端点 n 接收到 CLEAR_FEATURE 端点请求，暂停特征由 SET_INTERFACE 清除，或由端点连接接口的 SET_CONFIGURATION 请求清除，或者暂停特征由 FW 清除。当 SET_INTERFACE 或 SET_CONFIGURATION 请求被正确处理后，即使 wValue 与当前设定值相同，除端点 0 外的所有目标端点的暂停特征都会在请求处理后被清除，并且这些位也会被清除为 0。如果端点 0 因为 STALL 响应而被设置为暂停特征，它不能被清除，因为 STALL 响应是在响应 SET_INTERFACE 和 SET_CONFIGURATION 请求时产生的。

备注 n = 0 至 4，7，8

（11）UF0 INT 状态 0 寄存器（UF0IS0）

该寄存器表明中断源。如果该寄存器的内容被更改，INT0B 信号变为有效。

该寄存器是只读的，按字节进行读取。

如果由 USBF 产生一个中断请求（INTUSBF0），FW 必须读取该寄存器来识别中断源。

向 UF0IC0 寄存器的对应位写入 0 时，该寄存器的各个位被强制清除为 0。

注意事项 在 USBF 中，多个中断源，比如总线复位，恢复，以及 Short，都是内部或操作并作为单个中断请求(INTUSBF0) 发出。于是，在发生多个中断源的情况下，对它们进行或操作并将结果作为 INTUSBF0 中断请求来发出。

比如，如果发生一个总线复位中断源和一个恢复中断源，这两个源进行或操作并发出 INTUSBF0 中断请求。

在这些条件下，如果总线复位中断源被清除为 0(UF0IC0.BUSRSTC = 0)，V850ES/JG3-L 内部 INTUSBF0 中断请求可能保持置 1，因为恢复中断源仍然保持。新的中断请求标志(US0BIC.US0BIF) 可能不被置 1。

在这种情况下，使用 INTUSBF0 中断服务程序执行对各个中断请求的清除处理之后，再次确认 UF0IS0 和 UF0IS1 寄存器的标志状态，如果还有任何中断源标志置 1，执行标志清除（只有适用的位需要清除（不要执行批量清除））。

（1/2）

	7	6	5	4	3	2	1	0	地址	复位后
UF0IS0	BUSRST	RSUSPD	0	SHORT	DMAED	SETRQ	CLRRQ	EPHALT	00200020H	00H

位的位置	位名称	功能
7	BUSRST	该位表明发生总线复位。 1：已发生总线复位（中断请求已产生） 0：非总线复位状态（默认值）
6	RSUSPD	该位表明发生恢复或挂起状态。通过 FW 参考 UF0EPS1 寄存器的第 7 位。 1：已发生恢复或挂起状态（中断请求已产生） 0：未发生恢复或挂起状态（默认值）
4	SHORT	该位表明数据读自 UF0BO1 或 UF0BO2 寄存器的 FIFO，且 USBSPnB 信号 (n = 2，4)变为有效。只有在 DMA 模式下 FIFO 满，它才有效。 1：USBSPnB 信号变为有效（中断请求已产生） 0：USBSPnB 信号未有效（默认值） 使用 UF0DMS1 寄存器来识别操作在哪个端点上执行。即使 UF0DMS1 寄存器被 FW 读取，该位也不会被自动清除为 0。

(2/2)

位的位置	位名称	功能
3	DMAED	该位表明对端点 n(n = 1 至 4，7)的 DMA 结束 (TC) 信号变为有效。 1：对端点 n 的 DMA 结束信号已经输入（中断请求已产生） 0：对端点 n 的 DMA 结束信号没有输入（默认值） 当该位被置 1 时，对端点 n 的 DMA 请求信号变为有效。对端点 n 的 DMA 请求信号没有变为有效，除非 FW 使能 DMA 传输。 使用 UF0DMS0 寄存器来确认在哪个端点上实际执行操作。但是，即使 UF0DMS0 寄存器被 FW 读取，该位也不会被自动清除为 0。
2	SETRQ	该位表明已经接收到将要被自动处理的 SET_XXXX 请求，并且被自动处理（XXXX = CONFIGURATION 或 FEATURE）。 1：已经接收到将要被自动处理的 SET_XXXX 请求（中断请求已产生） 0：没有接收到将要被自动处理的 SET_XXXX 请求（默认值） 状态阶段完成后，该位被置为 1。参考 UF0SET 寄存器来识别请求的目标。即使 UF0SET 寄存器被 FW 读取，该位也不会被自动清除为 0。 当已经接收到 SET_FEATURE 端点请求时，EPHALT 位也被置为 1。
1	CLRRQ	该位表明已经接收到 CLEAR_FEATURE 请求并且被自动处理。 1：已经接收到 CLEAR_FEATURE 请求（中断请求已产生） 0：没有接收到 CLEAR_FEATURE 请求（默认值） 状态阶段完成后，该位被置为 1。参考 UF0CLR 寄存器来识别请求的目标。即使 UF0CLR 寄存器被 FW 读取，该位也不会被自动清除为 0。
0	EPHALT	该位表明一个端点已经停顿。 1：端点已经停顿（中断请求已产生） 0：端点没有停顿（默认值） 当通过设置 FW 停顿一个端点时，该位也被设置为 1。 通过参考 UF0EPS2 寄存器来识别哪一个端点已经停顿。即使接收到 CLEAR_FEATURE 端点、SET_INTERFACE 或 SET_CONFIGURATION 请求时，该位也不会被自动清除为 0。如果在端点 0 溢出的情况下接收到下一个 SETUP 令牌，它也不被自动清除为 0。 **注意事项 即使端点 0 的暂停特征被设置并且产生这个中断请求，在接收到 SET_FEATURE 端点 0、CLEAR_FEATURE 端点 0 或 GET_STATUS 端点 0 请求或 FW 处理的请求，和接收到上述以外的其它 SETUP 令牌之间，UF0EPS2 寄存器的第 0 位被屏蔽并且被清除为 0。**

（12）UF0 INT 状态 1 寄存器（UF0IS1）

该寄存器表明中断源。如果该寄存器的内容被更改，INT0B 信号变为有效。

该寄存器是只读的，按字节进行读取。

如果由 USBF 产生一个中断请求（INTUSBF0），FW 必须读取该寄存器来识别中断源。

向 UF0IC0 寄存器的对应位写入 0 时，该寄存器的各个位被强制清除为 0。然而，当接收到下一个 SETUP 令牌时，UF0IS1 寄存器的 SUCES 和 STG 位被自动清除为 0。

注意事项 在 USBF 中，多个中断源，比如总线复位，恢复，以及 Short，都是内部或操作并作为单个中断请求(INTUSBF0) 发出。于是，在发生多个中断源的情况下，对它们进行或操作并将结果作为 INTUSBF0 中断请求来发出。

比如，如果发生一个总线复位中断源和一个恢复中断源，这两个源进行或操作并发出 INTUSBF0 中断请求。

在这些条件下，如果总线复位中断源被清除为 0 (UF0IC0.BUSRSTC = 0)，V850ES/JG3-L 内部 INTUSBF0 中断请求可能保持置 1，因为恢复中断源仍然保持。新的中断请求标志(US0BIC.US0BIF) 可能不被置 1。

在这种情况下，使用 INTUSBF0 中断服务程序执行对各个中断请求的清除处理之后，再次确认 UF0IS0 和 UF0IS1 寄存器的标志状态，如果还有任何中断源标志置 1，执行标志清除（只有适用的位需要清除（不要执行批量清除））。

（1/2）

	7	6	5	4	3	2	1	0	地址	复位后
UF0IS1	0	E0IN	E0INDT	E0ODT	SUCES	STG	PROT	CPUDEC	00200022H	00H

位的位置	位名称	功能
6	E0IN	该位表明接收到对端点 0 的 IN 令牌，并且硬件已经自动发送 NAK。 1：接收到 IN 令牌并且 NAK 已发送（中断请求已产生）。 0：没有接收到 IN 令牌（默认值）。
5	E0INDT	该位表明数据被正确从 UF0E0W 寄存器发送。 1：从 UF0E0W 寄存器的发送已经完成（中断请求已产生）。 0：从 UF0E0W 寄存器发送尚未结束（默认值）。 紧邻在设置 UF0E0N 寄存器的 EP0NKW 位为 1 后，数据和 IN 令牌同步发送。当主机正确接收到这个数据时，该位由硬件自动设置为 1。即使数据是 Null 包，它也会被设置为 1。 当执行对 UF0E0W 寄存器的第一个写入访问时，该位被硬件自动清除为 0。

（2/2）

位的位置	位名称	功能
4	E0ODT	该位表明数据被正确接收到 UF0E0R 寄存器中。 1：数据是在 UF0E0R 寄存器中 （中断请求已产生）。 0：数据不在 UF0E0R 寄存器中 （默认值）。 当正确接收到数据时，该位被硬件自动设置为 1。同时，UF0EPS0 寄存器的 EP0R 位也被置为 1。如果接收到一个 Null 包，该位不被置 1。当 FW 读取 UF0E0R 寄存器时，它被硬件自动清除为 0，并且 UF0E0L 寄存器的值变为 0。
3	SUCES	该位表明已经接收到一个 FW 处理的请求或硬件处理的请求，并且已经正确完成状态阶段。 1：控制传输已经被正确处理（中断请求已产生）。 0：控制传输尚未被正确处理（默认值）。 状态阶段结束时，该位被置为 1。接收到下一个 SETUP 令牌时，它被硬件自动清除为 0。在控制传输的状态阶段接收到带有 0（空数据）的 Data PID 的数据时，该位也被设置为 1。
2	STG	当控制传输的阶段进入状态阶段时，该位被设置为 1。它对 FW 处理的请求和硬件处理的请求都有效。当控制传输（无数据）的阶段进入状态阶段时，该位也被设置为 1。 1：状态阶段（中断请求已产生） 0：非状态阶段（默认值） 当接收到下一个 SETUP 令牌时，该位被硬件自动清除为 0。 在数据阶段中未能正确接收到 ACK，当控制传输的阶段进入状态阶段，它也会被设置为 1。在这种情况下，如果 FW 正在处理控制传输（读），只要 UF0E0W 寄存器被清除，UF0E0N 寄存器的 EP0NKW 位就立即被清除为 0。
1	PROT	该位表明接收到一个 SETUP 令牌。它对 FW 处理的请求和硬件处理的请求都有效。 1：正确接收到 SETUP 令牌（中断请求已产生）。 0：没有接收到 SETUP 令牌（默认值）。 当数据被正确接收到 UF0E0ST 寄存器中时，该位被置为 1。当执行对 UF0E0ST 寄存器的第一次读取访问时，通过 FW 清除该位为 0。如果它没有被 FW 清除为 0，下一个 SETUP 令牌的接收不能被正确识别。 该位用来准确识别控制传输期间被再次执行的 SETUP 事务。如果在控制传输期间 SETUP 事务被重新执行，并且如果第二个请求由硬件执行，则 CPUDEC 位不被置为 1，但是 PROT 位可以用于对重新执行的识别。
0	CPUDEC	该位表明 UF0E0ST 寄存器有一个请求需要 FW 进行译码。 1：数据是在 UF0E0ST 寄存器中（中断请求已产生）。 0：数据不在 UF0E0ST 寄存器中（默认值）。 当 UF0E0ST 寄存器的所有数据被读取时，该位由硬件自动清除为 0。

（13）UF0 INT 状态 2 寄存器（UF0IS2）

该寄存器表明中断源。如果该寄存器的内容被更改，EPCINT1B 信号变为有效。

该寄存器是只读的，按字节进行读取。

如果由 USBF 产生一个中断请求（INTUSBF0），FW 必须读取该寄存器来识别中断源。

向 UF0IC2 寄存器的对应位写入 0 时，该寄存器的各个位被强制清除为 0。

如果 UF0EnIM（n = 1，3，7）寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IS2	BKI2IN	BKI2DT	BKI1IN	BKI1DT	0	0	0	IT1DT	00200024H	00H

位的位置	位名称	功能
7，5	BKInIN	这些位表明 IN 令牌已经接收到 UF0BIn 寄存器（端点 m）中，并且已经返回 NAK。 1：接收到 IN 令牌，并且 NAK 已发送（中断请求已产生） 0：没有接收到 IN 令牌（默认值）
6，4	BKInDT	这些位表明 UF0BI1 寄存器 （端点 1）的 FIFO 已经被切换。这意味着可以向端点 m 写入数据。 1：FIFO 已经被切换（中断请求已产生） 0：FIFO 尚未被切换（默认值） 紧邻在设置 UF0EN 寄存器的 BKInNK 位为 1 后，写入端点 m 的数据和 IN 令牌同步发送。当 FIFO 被切换后，可以从 CPU 写入数据，该位由硬件自动设置为 1。即使数据是 Null 包，当 FIFO 被切换时，该位也被设置为 1。当执行对 UF0BI1 寄存器的第一个写入访问时，该位由硬件自动清除为 0。
0	IT1DT	这些位表明数据已经从 UF0INT1 寄存器（端点 x）中正确完成接收。 1：传输已经完成（中断请求已产生） 0：传输尚未完成（默认值） 紧邻在设置 UF0EN 寄存器的 ITnNK 位为 1 后，数据和 IN 令牌同步发送。当主机已经正确接收到该数据后，该位由硬件自动设置为 1。当执行对 UF0INT1 寄存器的第一个写入访问时，该位由硬件自动清除为 0。即使数据是 Null 包，该位也被设置为 1。

备注 n = 1，2

当 n = 1 时，m = 1 且 x = 7

当 n = 2 时，m = 3

（14）UF0 INT 状态 3 寄存器（UF0IS3）

该寄存器表明中断源。如果该寄存器的内容被更改，EPCINT1B 信号变为有效。

该寄存器是只读的，按字节进行读取。

如果由 USBF 产生一个中断请求（INTUSBF0），FW 必须读取该寄存器来识别中断源。

向 UF0IC3 寄存器的对应位写入 0 时，该寄存器的各个位被强制清除为 0。

如果 UF0EnIM（n = 2，4）寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

(1/2)

	7	6	5	4	3	2	1	0	地址	复位后
UF0IS3	BKO2FL	BKO2NL	BKO2 NAK	BKO2DT	BKO1FL	BKO1NL	BKO1 NAK	BKO1DT	00200026H	00H

位的位置	位名称	功能
7，3	BKOnFL	这些位表明数据已经被正确接收到 UF0BOn 寄存器（端点 m） 中，并且 CPU 和 SIE 的 FIFO 都保持该数据。 1：接收的数据是在 UF0BOn 寄存器的两个 FIFO 中（中断请求已产生） 0：接收的数据不在 UF0BO1 寄存器的 SIE 端的 FIFO 中（默认值） 如果数据被保持在 CPU 和 SIE 的 FIFO 中，该位由硬件自动设置为 1。当 FIFO 被切换时，该位由硬件自动清除为 0。
6，2	BKOnNL	这些位表明一个 Null 包（长度为 0 的包）被接收到 UF0BOn 寄存器 （端点 m）中。 1：Null 包已被接收（中断请求已产生） 0：Null 包未被接收（默认值） 当 FIFO 为空时，Null 包接收后，这些位立即被设置为 1。如果数据在 CPU 端的 FIFO 中，当完成对这个 FIFO 的读取时，该位被设置为 1。
5，1	BKOnNAK	这些位表明 OUT 令牌已经被接收到 UF0BOn 寄存器（端点 m）中，并且已经返回 NAK。 1：接收到 OUT 令牌并且 NAK 已发送（中断请求已产生）。 0：没有接收到 OUT 令牌（默认值）。

备注 n = 1，2
当 n = 1 时，m = 2
当 n = 2 时，m = 4

(2/2)

位的位置	位名称	功能
4，0	BKOnDT	这些位表明在 UF0BOn 寄存器 （端点 m）中数据被正确接收。 1：接收已经正确完成（中断请求已产生） 0：接收尚未完成（默认值） 当数据被正确接收并且 FIFO 被切换时，这些位被硬件自动设置为 1。同时，UF0EPS0 寄存器的对应位也被置为 1。如果数据是 Null 包，这些位不被置为 1。当由于 UF0BOn 寄存器被 FW 读取，导致 UF0BOnL 寄存器的值变为 0 时，该位被硬件自动清除为 0。 当 CPU 端 FIFO 的所有内容已经被读取时，这些位被自动清除为 0。但是，如果此时数据是在 SIE 端的 FIFO 中，中断请求不会被清除，并且 INTUSBF1 信号不变为无效。如果数据是连续接收的，信号保持有效。

备注 n = 1，2
当 n = 1 时，m = 2
当 n = 2 时，m = 4

（15）UF0 INT 状态 4 寄存器（UF0IS4）

该寄存器表明中断源。如果该寄存器的内容被更改，EPCINT2B 信号变为有效。

该寄存器是只读的，按字节进行读取。

如果由 USBF 产生一个中断请求（INTUSBF0），FW 必须读取该寄存器来识别中断源。

向 UF0IC4 寄存器的对应位写入 0 时，该寄存器的各个位被强制清除为 0。

如果 UF0EnIM（n = 1 至 4，7）寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IS4	0	0	SETINT	0	0	0	0	0	00200028H	00H

位的位置	位名称	功能
5	SETINT	该位表明已经接收到 SET_INTERFACE 请求，并且被自动处理。 1：请求已经被自动处理（中断请求已产生）。 0：请求尚未被自动处理（默认值）。 可以通过读取 UF0ASS 或 UF0IFn 寄存器（n = 0 至 4）来识别该位的当前设置。

（16）UF0 INT 屏蔽 0 寄存器（UF0IM0）

该寄存器用于控制中断源的屏蔽，中断源由 UF0IS0 寄存器表示。

该寄存器可以按字节进行读取或写入。

向该寄存器的对应位写入 1，FW 可以屏蔽来自 USBF 的中断请求（INTUSBF0）的发生。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IM0	BUS RSTM	RSU SPDM	0	SHORTM	DMA EDM	SET RQM	CLR RQM	EP HALTM	0020002EH	00H

位的位置	位名称	功能
7	BUSRSTM	该位屏蔽总线复位中断。 1：屏蔽 0：不屏蔽（默认值）
6	RSUSPDM	该位屏蔽恢复/挂起中断。 1：屏蔽 0：不屏蔽（默认值）
4	SHORTM	该位屏蔽 Short 中断。 1：屏蔽 0：不屏蔽（默认值）
3	DMAEDM	该位屏蔽 DMA_END 中断。 1：屏蔽 0：不屏蔽（默认值）
2	SETRQM	该位屏蔽 SET_RQ 中断。 1：屏蔽 0：不屏蔽（默认值）
1	CLRRQM	该位屏蔽 CLR_RQ 中断。 1：屏蔽 0：不屏蔽（默认值）
0	EPHALTM	该位屏蔽 EP_Halt 中断。 1：屏蔽 0：不屏蔽（默认值）

（17）UF0 INT 屏蔽 1 寄存器（UF0IM1）

该寄存器控制中断源的屏蔽, 中断源由 UF0IS1 寄存器表示。

该寄存器可以按字节进行读取或写入。

向该寄存器的对应位写入 1，FW 可以屏蔽来自 USBF 的中断请求（INTUSB0B）的发生。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IM1	0	E0INM	E0 INDTM	E0 ODTM	SUCESM	STGM	PROTM	CPU DECM	00200030H	00H

位的位置	位名称	功能
6	E0INM	该位屏蔽 EP0IN 中断。 1：屏蔽 0：不屏蔽（默认值）
5	E0INDTM	该位屏蔽 EP0INDT 中断。 1：屏蔽 0：不屏蔽（默认值）
4	E0ODTM	该位屏蔽 EP0OUTDT 中断。 1：屏蔽 0：不屏蔽（默认值）
3	SUCESM	该位屏蔽 Success 中断。 1：屏蔽 0：不屏蔽（默认值）
2	STGM	该位屏蔽 Stg 中断。 1：屏蔽 0：不屏蔽（默认值）
1	PROTM	该位屏蔽 Protect 中断。 1：屏蔽 0：不屏蔽（默认值）
0	CPUDECM	该位屏蔽 CPUDEC 中断。 1：屏蔽 0：不屏蔽（默认值）

（18）UF0 INT 屏蔽 2 寄存器（UF0IM2）

该寄存器控制中断源的屏蔽，中断源由 UF0IS2 寄存器表示。

该寄存器可以按字节进行读取或写入。

向该寄存器的对应位写入 1，FW 可以屏蔽来自 USBF 的中断请求（INTUSBF0）的发生。

如果 UF0EnIM 寄存器（n = 1, 3, 7）的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IM2	BKI2INM	BKI2 DTM	BKI1INM	BKI1 DTM	0	0	0	IT1DTM	00200032H	00H

位的位置	位名称	功能
7，5	BKInINM	该位屏蔽 BKInIN 中断。 1：屏蔽 0：不屏蔽（默认值）
6，4	BKInDTM	该位屏蔽 BLKInDT 中断。 1：屏蔽 0：不屏蔽（默认值）
0	IT1DTM	该位屏蔽 INTnDT 中断。 1：屏蔽 0：不屏蔽（默认值）

备注 n = 1，2

（19）UF0 INT 屏蔽 3 寄存器（UF0IM3）

该寄存器控制中断源的屏蔽，中断源由 UF0IS3 寄存器表示。

该寄存器可以按字节进行读取或写入。

向该寄存器的对应位写入 1，FW 可以屏蔽来自 USBF 的中断请求（INTUSBF0）的发生。

如果 UF0EnIM 寄存器（n = 2，4）的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IM3	BKO2 FLM	BKO2 NLM	BKO2 NAKM	BKO2 DTM	BKO1 FLM	BKO1 NLM	BKO1 NAKM	BKO1 DTM	00200034H	00H

位的位置	位名称	功能
7，3	BKOnFLM	该位屏蔽 BLKOnFL 中断。 1：屏蔽 0：不屏蔽（默认值）
6，2	BKOnNLM	该位屏蔽 BLKOnNL 中断。 1：屏蔽 0：不屏蔽（默认值）
5，1	BKOnNAKM	该位屏蔽 BLKOnNK 中断。 1：屏蔽 0：不屏蔽（默认值）
4，0	BKOnDTM	该位屏蔽 BLKOnDT 中断。 1：屏蔽 0：不屏蔽（默认值）

备注 n = 1，2

（20）UF0 INT 屏蔽 4 寄存器（UF0IM4）

该寄存器控制中断源的屏蔽，中断源由 UF0IS4 寄存器表示。

该寄存器可以按字节进行读取或写入。

向该寄存器的对应位写入 1，FW 可以屏蔽来自 USBF 的中断请求（INTUSBF0）的发生。

如果 UF0EnIM 寄存器（n = 1 至 4，7）的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IM4	0	0	SETINTM	0	0	0	0	0	00200036H	00H

位的位置	位名称	功能
5	SETINTM	该位屏蔽 SET_INT 中断。 1：屏蔽 0：不屏蔽（默认值）

（21）UF0 INT 清除 0 寄存器（UF0IC0）

该寄存器控制中断源的清除，中断源由 UF0IS0 寄存器表示。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 FFH。

向该寄存器的对应位写入 0，FW 可以清除一个中断请求。即使可以由硬件自动清除位 0 的位，在被硬件清除之前也可以由 FW 清除。向该寄存器的某一位写入 0 自动设置该位为 1。写入 1 无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IC0	BUS RSTC	RSU SPDC	1	SHORTC	DMA EDC	SET RQC	CLR RQC	EP HALTC	0020003CH	FFH

位的位置	位名称	功能
7	BUSRSTC	该位清除总线复位中断。 0：清除
6	RSUSPDC	该位清除恢复/挂起中断。 0：清除
4	SHORTC	该位清除 Short 中断。 0：清除
3	DMAEDC	该位清除 DMA_END 中断。 0：清除
2	SETRQC	该位清除 SET_RQ 中断。 0：清除
1	CLRRQC	该位清除 CLR_RQ 中断。 0：清除
0	EPHALTC	该位清除 EP_Halt 中断。 0：清除

（22）UF0 INT 清除 1 寄存器（UF0IC1）

该寄存器控制中断源的清除，中断源由 UF0IS1 寄存器表示。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 FFH。

向该寄存器的对应位写入 0，FW 可以清除一个中断请求。即使可以由硬件自动清除位 0 的位，在被硬件清除之前也可以由 FW 清除。向该寄存器的某一位写入 0 自动设置该位为 1。写入 1 无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IC1	1	E0INC	E0 INDTC	E0ODTC	SUCESC	STGC	PROTC	CPU DECC	0020003EH	FFH

位的位置	位名称	功能
6	E0INC	该位清除 EP0IN 中断。 0：清除
5	E0INDTC	该位清除 EP0INDT 中断。 0：清除
4	E0ODTC	该位清除 EP0OUTDT 中断。 0：清除
3	SUCESC	该位清除 Success 中断。 0：清除
2	STGC	该位清除 Stg 中断。 0：清除
1	PROTC	该位清除 Protect 中断。 0：清除
0	CPUDECC	该位清除 CPUDEC 中断。 0：清除

（23）UF0 INT 清除 2 寄存器（UF0IC2）

该寄存器控制中断源的清除，中断源由 UF0IS2 寄存器表示。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 FFH。

向该寄存器的对应位写入 0，FW 可以清除一个中断请求。即使可以由硬件自动清除位 0 的位，在被硬件清除之前也可以由 FW 清除。向该寄存器的某一位写入 0 自动设置该位为 1。写入 1 无效。

如果 UF0EnIM（n = 1，3，7）寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IC2	BKI2INC	BKI2 DTC	BKI1INC	BKI1 DTC	1	1	1	IT1DTC	00200040H	FFH

位的位置	位名称	功能
7，5	BKInINC	这些位清除 BLKInIN 中断。 0：清除
6，4	BKInDTC	这些位清除 BLKInDT 中断。 0：清除
0	IT1DTC	这些位清除 INTnDT 中断。 0：清除

备注 n = 1，2

（24）UF0 INT 清除 3 寄存器（UF0IC3）

该寄存器控制中断源的清除，中断源由 UF0IS3 寄存器表示。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 FFH。

向该寄存器的对应位写入 0，FW 可以清除一个中断请求。即使可以由硬件自动清除位 0 的位，在被硬件清除之前也可以由 FW 清除。向该寄存器的某一位写入 0 自动设置该位为 1。写入 1 无效。

如果 UF0EnIM（n = 2，4）寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IC3	BKO2 FLC	BKO2 NLC	BKO2 NAKC	BKO2 DTC	BKO1 FLC	BKO1 NLC	BKO1 NAKC	BKO1 DTC	00200042H	FFH

位的位置	位名称	功能
7，3	BKOnFLC	这些位清除 BLKOnFL 中断。 0：清除
6，2	BKOnNLC	这些位清除 BLKOnNL 中断。 0：清除
5，1	BKOnNAKC	这些位清除 BLKOnNK 中断。 0：清除
4，0	BKOnDTC	这些位清除 BLKOnDT 中断。 0：清除

备注 n = 1，2

（25）UF0 INT 清除 4 寄存器（UF0IC4）

该寄存器控制中断源的清除，中断源由 UF0IS4 寄存器表示。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 FFH。

向该寄存器的对应位写入 0，FW 可以清除一个中断请求。即使可以由硬件自动清除位 0 的位，在被硬件清除之前也可以由 FW 清除。向该寄存器的某一位写入 0 自动设置该位为 1。写入 1 无效。

如果 UF0EnIM（n = 1 至 4，7）寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IC4	1	1	SETINTC	1	1	1	1	1	00200044H	FFH

位的位置	位名称	功能
5	SETINTC	该位清除 SET_INT 中断。 0：清除

(26) UF0 INT & DMARQ 寄存器（UF0IDR）

该寄存器选择通过中断请求或启动 DMA 来报告。

该寄存器可以按字节进行读取或写入。

如果数据存在于 UF0BO1 或 UF0BO1 寄存器中，或者如果数据可以写入到 UF0BI1 或 UF0BI2 寄存器，该寄存器选择通过中断请求或启动请求的 DMA 来报告给 FW。如果启动了请求的 DMA，可以根据第 0 位和第 1 位的设置来选择 DMA 传输模式。

如果 UF0EnIM（n = 1 至 4）寄存器的设置和接口的当前设置不支持各个端点，相关的位无效。

请确保将第 3 位和第 2 位清除为“0”。如果它们被设置为 1，操作无法保证。

注意事项 如果在 DMA 传输中 SET_INTERFACE 请求不支持目标端点，DMA 请求信号立即变为无效，并且对应位被硬件自动清除为 0。

(1/2)

	7	6	5	4	3	2	1	0	地址	复位后
UF0IDR	DQBI2MS	DQBI1MS	DQBO2MS	DQBO1MS	0	0	MODE1	MODE0	0020004CH	00H

位的位置	位名称	功能
7，6	DQBInMS	这些位使能 (屏蔽) 一个向 UF0BIn 寄存器写入的 DMA 传输请求（对端点 m 的 DMA 请求信号）。当这些位被置 1，写入数据可以应答时，对端点 m 的 DMA 请求信号变为有效。如果输入对端点 m 的 DMA 结束信号（如果 DMA 控制器发出 TC），这些位被硬件自动清除为 0。要继续 DMA 传输，通过 FW 重置这些位为 1。 1：允许有效对端点 m 的 DMA 请求信号（屏蔽 BKInDT 中断） 0：禁止有效对端点 m 的 DMA 请求信号（默认值）
5，4	DQBOnMS	这些位使能 (屏蔽)一个对 UF0BOn 寄存器读取的 DMA 传输请求（对端点 x 的 DMA 请求信号）。当这些位被置 1，要数据的数据已经在 UF0Bon 寄存器中准备好，对端点 x 的 DMA 请求信号变为有效。如果输入对端点 x 的 DMA 结束信号（如果 DMA 控制器发出 TC），这些位被硬件自动清除为 0。当 USBSPxB 信号变为有效时，它们也被清除为 0。要继续 DMA 传输，通过 FW 重置这些位为 1。 1：允许有效对端点 x 的 DMA 请求信号（屏蔽 BKOnDT 中断） 0：禁止有效对端点 x 的 DMA 请求信号（默认值）

备注 n = 1, 2

当 n = 1 时，m = 1 且 x = 2

当 n = 2 时，m = 3 且 x = 4

(2/2)

<table>
<tr><th>位的位置</th><th>位名称</th><th>功能</th></tr>
<tr><td>1，0</td><td>MODE1，
MODE0</td><td>这些位选择 DMA 传输模式。
<table>
<tr><th>MODE1</th><th>MODE0</th><th>模式</th><th>备注</th></tr>
<tr><td>1</td><td>0</td><td>需求模式</td><td>只要有数据，DMA 请求信号就变为有效。
如果没有数据，它就变为无效。</td></tr>
<tr><td colspan="2">其它</td><td colspan="2">禁止设置</td></tr>
</table></td></tr>
</table>

(27) UF0 DMA 状态 0 寄存器（UF0DMS0）

该寄存器用于表示端点 1 至端点 4 的 DMA 状态。

该寄存器是只读的，按字节进行读取。

如果 UF0EnIM 寄存器（n = 1 至 4）的设置与接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0DMS0	0	0	DQE4	DQE3	DQE2	DQE1	0	0	0020004EH	00H

位的位置	位名称	功能
5	DQE4	该位表明发出一个从端点 4 到存储器的 DMA 读取请求。 1：发出来自端点 4 的 DMA 读取请求 0：没有发出来自端点 4 的 DMA 读取请求（默认值）
4	DQE3	该位表明发出一个从存储器到端点 3 的 DMA 写入请求。 请注意，即使数据是在端点 3 内（在 BKI2DED 位被置 1 后，FIFO 还未满时），DMA 请求信号立即变为有效，且当 UF0IDR 寄存器的 DQBI2MS 位被置 1 时，启动 DMA 传输。 1：发出对端点 3 的 DMA 写入请求 0：没有发出对端点 3 的 DMA 写入请求（默认值）
3	DQE2	该位表明发出一个从端点 2 到存储器的 DMA 读取请求。 1：发出来自端点 2 的 DMA 读取请求 0：没有发出来自端点 2 的 DMA 读取请求（默认值）
2	DQE1	该位表明发出一个从存储器到端点 1 的 DMA 写入请求。 请注意，即使数据是在端点 1 内（在 BKI1DED 位被置 1 后，FIFO 还未满时），DMA 请求信号立即变为有效，且当 UF0IDR 寄存器的 DQBI1MS 位被置 1 时，启动 DMA 传输。 1：发出对端点 1 的 DMA 写入请求 0：没有发出对端点 1 的 DMA 写入请求（默认值）

(28) UF0 DMA 状态 1 寄存器（UF0DMS1）

该寄存器用于表示端点 1 至端点 4 的 DMA 状态。

该寄存器是只读的，按字节进行读取。

如果 UF0EnIM 寄存器（n = 1 至 4）的设置与接口的当前设置不支持各个端点，相关的位无效。

当该寄存器被读取时，每位都被自动清除为 0。即使当该寄存器被读取时，但是 UF0IS0 寄存器的第 4 位和第 3 位不被清除为 0，如果 SET_INTERFACE 请求不再支持目标端点，每位都被硬件自动清除为 0（但是，DMA_END 中断请求和 Short 中断请求不被清除）。

	7	6	5	4	3	2	1	0	地址	复位后
UF0DMS1	DEDE4	DSPE4	DEDE3	DEDE2	DSPE2	DEDE1	0	0	00200050H	00H

位的位置	位名称	功能
7，5，4，2	DEDEn	这些位表明对端点 n 的 DMA 结束 (TC) 信号变为有效，且发出从端点 n 到存储器的 DMA 读取请求时，DMA 停止。 1：对端点 n 的 DMA 结束信号是有效的 0：对端点 n 的 DMA 结束信号是无效的（默认值）
6，3	DSPEm	这些位表明发出了从端点 m 到存储器的 DMA 读取请求，DMA 已经被停止，因为接收到得数据是一个短包，并且没有其它更多数据需要传输。 1：DMASTOP_EPm 信号是有效的 0：DMASTOP_EPm 信号是无效的（默认值）

备注 n = 1 至 4
m = 2，4

（29）UF0 FIFO 清除 0 寄存器（UF0FIC0）

该寄存器清除每个 FIFO。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 00H。

向该寄存器的对应位写入 1，FW 可以清除目标 FIFO。写入 1 的位被自动清除为 0。向该位写入 0 无效。

如果 UF0EnIM 寄存器（n = 1，3，7）的设置与接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0FIC0	BKI2SC	BKI2CC	BKI1SC	BKI1CC	0	ITR1C	EP0WC	EP0RC	00200060H	00H

位的位置	位名称	功能
7，5	BKInSC	这些位只清除 UF0BIn 寄存器的 SIE 端的 FIFO（复位该计数器）。 1：清除 BKInNK 位置为 1，当对端点 m 的 IN 令牌正在被处理时，写入该位无效。 通过清除 FIFO，BKInNK 位被自动清除为 0。当使用这些位时，请确保 CPU 端的 FIFO 为空。
6，4	BKInCC	这些位只清除 UF0BIn 寄存器的 CPU 端的 FIFO（复位该计数器）。 1：清除
2	ITR1C	这些位只清除 UF0INT1 寄存器（复位该计数器）。 1：清除 当 IT1NK 位置为 1，对端点 7 的 IN 令牌正在被处理时，写入该位无效。 通过清除 FIFO，IT1NK 位被自动清除为 0。
1	EP0WC	该位清除 UF0E0W 寄存器（复位该计数器）。 1：清除 EP0NKW 位置为 1，当对端点 0 的 IN 令牌正在被处理时，写入该位无效。 通过清除 FIFO，EP0NKW 位被自动清除为 0。
0	EP0RC	该位清除 UF0E0R 寄存器（复位该计数器）。 1：清除 当 EP0NKR 位被设置为 1（除已经被 FW 设置外）时，通过清除 FIFO，EP0NKR 位被自动清除为 0。

备注 n = 1，2
当 n = 1 时，m = 1
当 n = 2 时，m = 3

（30）UF0 FIFO 清除 1 寄存器（UF0FIC1）

该寄存器清除每个 FIFO。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 00H。

向该寄存器的对应位写入 1，FW 可以清除目标 FIFO。写入 1 的位被自动清除为 0。向该位写入 0 无效。

如果 UF0EnIM 寄存器（n = 2，4）的设置与接口的当前设置不支持各个端点，相关的位无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0FIC1	0	0	0	0	BKO2C	BKO2CC	BKO1C	BKO1CC	00200062H	00H

位的位置	位名称	功能
3，1	BKOnC	这些位清除 UF0BOn 寄存器的 SIE 端和 CPU 端的 FIFO （复位该计数器）。 1：清除 当 BKO1nNK 位被置为 1 时（除已经由 FW 设置外），通过清除 FIFO，BKOnNK 位被自动清除为 0。
2，0	BKOnCC	这些位只清除 UF0BOn 寄存器的 CPU 端的 FIFO （复位该计数器）。 1：清除 当 BKOnNK 位被置为 1 时（除已经由 FW 设置外），通过清除 FIFO，BKOnNK 位被自动清除为 0。

备注 n = 1，2

（31）UF0 数据结束寄存器（UF0DEND）

该寄存器用于报告写入到传输系统的结束。

该寄存器只能写入，可以按字节进行写入（但是，第 7 位和第 6 位可以进行读取和写入）。如果读取该寄存器，读到的值是 00H。

向该寄存器的对应位写入 1，FW 可以清除目标 FIFO。写入 1 的位被自动清除为 0。向该位写入 0 无效。

如果 UF0EnIM 寄存器（n = 1，3，7）的设置与接口的当前设置不支持各个端点，相关的位无效。

(1/2)

	7	6	5	4	3	2	1	0	地址	复位后
UF0DEND	BKI2T	BKI1T	0	0	IT1DEND	BKI2DED	BKI1DED	E0DED	0020006AH	00H

位的位置	位名称	功能
7，6	BKInT	如果因为 DMA，导致 UF0BIn 寄存器的 CPU 端的 FIFO 装满，这些位指定是否自动执行 FIFO 的切换。 1：只要 FIFO 装满，立即自动执行 FIFO 的切换操作 0：即使 FIFO 装满，也不自动执行 FIFO 的切换操作（默认值）
3	IT1DEND	设置该位为 1 来发送 UF0INT1 寄存器的数据。该位被置为 1 时，IT1NK 位被设置为 1 并且执行数据传输。 1：发送一个短包 0：不发送短包（默认值） 如果 UF0FIC0 寄存器的 ITR1C 位被置为 1，然后这些位被置为 1（UF0INT1 寄存器的计数器 = 0 且 UF0EPS0 寄存器的对应位 = 1），发送一个 Null 包（数据长度为 0）。 如果数据存在于 UF0INT1 寄存器中，并且如果该位被设置为 1（UF0INT1 寄存器的计数器 ≠ 0 且 UF0EPS0 寄存器的对应位 = 1），一个短包被发送。 当 FIFO 装满时，这些位被硬件自动控制。

备注 n = 1，2

(2/2)

位的位置	位名称	功能
2，1	BKInDED	向 UF0BIn 寄存器写入发送数据完成时，设置这些位为 1。当这些位被置为 1 时，FIFO 被尽快切换，BKInNK 位被设置为 1，并且数据被传输。 1：发送一个短包 0：不发送短包（默认值） 这些位控制 CPU 端的 FIFO。 如果 UF0FIC0 寄存器的 BKInCC 位被置为 1，然后这些位被置为 1（UF0BIn 寄存器的计数器 = 0），发送一个 Null 包（数据长度为 0）。 如果数据存在于 UF0BIn 寄存器中，且如果这些位被置为 1（UF0BIn 寄存器的计数器≠ 0），如果 FIFO 未满，发送一个短包。 如果因为 DMA，导致 UF0BIn 寄存器的 CPU 端的 FIFO 装满，且 PIO 或 BKInT 位置 1，即使这些位没有被设置为 1，硬件也会启动数据发送。 如果因为 DMA，导致 UF0BIn 寄存器的 CPU 端的 FIFO 装满，且 BKInT 位被清除为 0，请确保设置这些位为 1（参见 **20.6.3 (3) UF0 EPNAK 寄存器（UF0EN）**）。
0	E0DED	设置该位为 1 来发送 UF0E0W 寄存器的数据。当该位被置为 1 时，EP0NKW 位被设置为 1 并且执行数据传输。 1：发送一个短包 0：不发送短包（默认值） 如果 UF0FIC0 寄存器的 EP0WC 位被置为 1，然后该位被置为 1（UF0E0W 寄存器的计数器 = 0 且 UF0EPS0 寄存器的第 1 位 = 1），发送一个 Null 包（数据长度为 0）。 如果数据存在于 UF0E0W 寄存器中，并且如果该位被设置为 1（UF0E0W 寄存器的计数器 ≠ 0 且 UF0EPS0 寄存器的第 1 位 = 1），如果 FIFO 未满，一个短包被发送。

备注 n = 1，2

（32）UF0 GPR 寄存器（UF0GPR）

该寄存器用于控制 USBF 和 USB 接口。

该寄存器只能写入，可以按字节进行写入。如果读取该寄存器，读到的值是 00H。请确保清除第 7 位至第 1 位为“0”。

向该寄存器的第 0 位写入 1，FW 可以复位 USBF。1 写入该位后，该位被自动清除为 0。向该位写入 0 无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0GPR	0	0	0	0	0	0	0	MRST	0020006EH	00H

位的位置	位名称	功能
0	MRST	设置该位为 1 来复位 USBF。 1：复位 实际上， 在该位被 FW 设置为 1 且写信号变无效后的两个 USB 时钟后，USBF 被复位。 当系统时钟工作时由 MRST 位复位 USBF 与通过 $\overline{\text{RESET}}$ 引脚复位（硬件复位）的作用相同 （寄存器值回到默认值）。

（33）UF0 模式控制寄存器（UF0MODC）

该寄存器控制 CPUDEC 处理。

该寄存器可以按字节进行读取或写入。

通过设置该寄存器的每一位，UF0MODS 寄存器的设置可以被更改。只有硬件复位以及 UF0GRP 寄存器的 MRST 位被设置为 1 时，该寄存器的此位才会被自动清除为 0。

即使该寄存器的此位由硬件自动设置为 1，通过 FW 进行的设置会优先作用。

请确保清除第 7 位和第 5 位为“0”。如果这些位被置为 1，操作无法保证。

注意事项 提供该寄存器是用于调试目的。通常情况下，不要设置该寄存器，除非为了检验操作或使用一个特殊模式时。

	7	6	5	4	3	2	1	0	地址	复位后
UF0MODC	0	CDC GDST	0	0	0	0	0	0	00200074H	00H

位的位置	位名称	功能
6	CDCGDST	设置该位为 1，将 GET_DESCRIPTOR 配置请求切换到 CPUDEC 处理。通过设置该位为 1，UF0MODS 寄存器的 CDCGD 位可以被强制置为 1。 1：将 GET_DESCRIPTOR 配置请求强制更改为 CPUDEC 处理（设置 UF0MODS 寄存器的 CDCGD 位为 1）。 0：自动处理 GET_DESCRIPTOR 配置请求 （默认值）。

（34）UF0 模式状态寄存器（UF0MODS）

该寄存器用于表示配置状态。

该寄存器是只读的，可以按字节进行读取。

	7	6	5	4	3	2	1	0	地址	复位后
UF0MODS	0	CDCGD	0	MPACK	DFLT	CONF	0	0	00200078H	00H

位的位置	位名称	功能
6	CDCGD	该位指定对 GET_DESCRIPTOR 配置请求是否执行 CPUDEC 处理。 1：将 GET_DESCRIPTOR 配置请求强制更改为 CPUDEC 处理。 0：自动处理 GET_DESCRIPTOR 配置请求（默认值）。
4	MPACK	该位表明端点 0 的发送包大小。 1：发送一个不是 8 字节的包。 0：发送一个 8 字节的包（默认值）。 在 GET_DESCRIPTOR 设备请求被处理后（状态阶段的正常完成），该位被硬件自动设置为 1。在 USBF 被复位前（它不会被总线复位清除为 0），它不会被清除为 0。 如果该位没有被置为 1，硬件只按照 8 字节单位传输自动执行的请求。因此，即使在 GET_DESCRIPTOR 设备请求完成之前，OUT 令牌发送的 8 个以上字节的数据要由 FW 处理，数据也可以被正确接收。 如果端点 0 的大小是 8 字节，该位被忽略。
3	DFLT	该位表明默认状态（DFLT 位 = 1）。 1：允许响应。 0：禁止响应（总是无响应）（默认值）。 该位由总线复位自动设置为 1。在该位被置为 1 之前，不会响应对所有端点的事务。
2	CONF	该位表明 SET_CONFIGURATION 请求是否已经完成。 1：SET_CONFIGURATION 请求已经完成。 0：SET_CONFIGURATION 请求尚未完成（默认值）。 当通过 SET_CONFIGURATION 请求接收到配置值 = 1 时，该位被设置为 1。 除非该位被置为 1，对端点 0 以外的端点的访问被忽略。 当通过 SET_CONFIGURATION 请求接收到配置值 = 1 时，该位被清除为 0。当检测到总线复位时，该位也被清除为 0。

（35）UF0 有效接口编号寄存器（UF0AIFN）

该寄存器用于设置正确响应 GET/SET_INTERFACE 请求的有效接口编号。因为接口 0 总是有效，可以选择接口 1 到 4。

该寄存器可以按字节进行读取或写入。

	7	6	5	4	3	2	1	0	地址	复位后
UF0AIFN	ADDIF	0	0	0	0	0	IFNO1	IFNO0	00200080H	00H

位的位置	位名称	功能
7	ADDIF	该位允许使用 0 以外的接口编号。 1：支持 IFNO1 和 IFNO0 位指定的接口编号。 0：仅支持接口 0（默认值）。 当该位没有被置为 1 时，该寄存器的第 1 位和第 0 位的设置无效。
1，0	IFNO1， IFNO0	这些位指定可以支持的接口编号范围。

IFNO1	IFNO0	有效接口编号
1	1	0，1，2，3，4
1	0	0，1，2，3
0	1	0，1，2
0	0	0，1

（36）UF0 有效复用设置寄存器（UF0AAS）

该寄存器用于指定接口编号和复用设置之间的一个链接。

该寄存器可以按字节进行读取或写入。

V850ES/JG3-L 的 USBF 可以为一个接口设置 5 级复用设置（复用设置 0、1、2、3 和 4 可以定义）和 2 级 复用设置（复用设置 0 和 1 可以定义）。

	7	6	5	4	3	2	1	0	地址	复位后
UF0AAS	ALT2	IFAL21	IFAL20	ALT2EN	ALT5	IFAL51	IFAL50	ALT5EN	00200082H	00H

位的位置	位名称	功能
7，3	ALTn	这些位指定是否有 n 级复用设置与接口 0 链接。当这些位被置为 1 时，IFALn1 和 IFALn0 位的设置无效。 1：n 级复用设置与接口 0 链接。 0：n 级复用设置与接口 0 没有链接（默认值）。
6，5， 2，1	IFALn1， IFALn0	这些位指定要与 n 级复用设置链接的接口编号。如果接口编号超出 UF0AIFN 寄存器指定的范围，n 级复用设置无效（ALTnEN 位 = 0）。 IFALn1 / IFALn0 / 将被链接的接口编号 1 / 1 / 链接到接口 4 1 / 0 / 链接到接口 3 0 / 1 / 链接到接口 2 0 / 0 / 链接到接口 1 对同一个接口编号，不要同时链接 5 级复用设置和 2 级复用设置。
4，0	ALTnEN	这些位使 n 级复用设置有效。除非这些位被置为 1，否则 ALTn、IFALn1 和 IFALn0 位的设置无效。 1：n 级复用设置有效。 0：n 级复用设置无效（默认值）。

备注 n = 2，5

例如，当 UF0AIFN 寄存器被设置为 82H，且 UF0AAS 寄存器被设置为 15H，接口 0、1、2 和 3 有效。接口 0 和 2 只支持复用设置 0。接口 1 支持复用设置 0 和 1，并且接口 3 支持复用设置 0、1、2、3 和 4。对于这些设置，要求 GET_INTERFACE wIndex = 0/1/2/3、SET_INTERFACE wValue = 0 & wIndex = 0/2、SET_INTERFACE wValue = 0/1 & wIndex = 1 且 SET_INTERFACE wValue = 0/1/2/3/4 & wIndex = 3 被自动响应，并且对于其它 GET/SET_INTERFACE 请求，产生一个 STALL 响应。

（37）UF0 复用设置状态寄存器（UF0ASS）

该寄存器用于表示复用设置的当前状态。

该寄存器是只读的，按字节进行读取。

当 SET_INT 中断请求已经发出时，检查该寄存器。通过 SET_INTERFACE 请求接收到的值反映在 UF0IFn 寄存器（n = 0 到 4）和该寄存器上。

	7	6	5	4	3	2	1	0	地址	复位后
UF0ASS	0	0	0	0	AL5ST3	AL5ST2	AL5ST1	AL2ST	00200084H	00H

位的位置	位名称	功能
3 至 1	AL5ST3 至 AL5ST1	这些位表明 5 级复用设置的当前状态。 AL5ST3 / AL5ST2 / AL5ST1 / 选择的复用设置编号 1 / 0 / 0 / 复用设置 4 0 / 1 / 1 / 复用设置 3 0 / 1 / 0 / 复用设置 2 0 / 0 / 1 / 复用设置 1 0 / 0 / 0 / 复用设置 0
0	AL2ST	该位表明 2 级复用设置 的当前状态（选择的复用设置编号）。 1：复用设置 1 0：复用设置 0

（38）UF0 端点 1 接口映射寄存器（UF0E1IM）

该寄存器用于指定端点 1 对哪个接口和复用设置有效。

该寄存器可以按字节进行读取或写入。

该寄存器的设置和通过 SET_INTERFACE 请求选择的复用设置表明端点 1 当前是否有效，并且硬件决定如何响应 GET_STATUS/CLEAR_FEATURE/SET_FEATURE 端点 1 请求和到端点 1 的 IN 事务，以及相关的位是有效或是无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E1IM	E1EN2	E1EN1	E1EN0	E12AL1	E15AL4	E15AL3	E15AL2	E15AL1	00200086H	00H

<table>
<tr><th>位的位置</th><th>位名称</th><th>功能</th></tr>
<tr><td>7 至 5</td><td>E1EN2 至
E1EN0</td><td>这些位设置一个端点 1 的接口与 2 / 5 级复用设置之间的链接。端点被链接到复用设置 0。
链接到复用设置 0 的端点不能从复用设置 1 至 4 中排除。
<table>
<tr><th>E1EN2</th><th>E1EN1</th><th>E1EN0</th><th>链接状态</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td rowspan="2">没有与接口链接</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>与接口 4 和复用设置 0 链接</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>与接口 3 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>与接口 2 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>与接口 1 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>与接口 0 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>没有与接口链接（默认值）</td></tr>
</table>
当这些位被设置为 110 或 111 时，即使 E12AL1 位被清除为 0，它们也无效。
如果端点被链接，设置 UF0MODS 寄存器的 CONF 位为 1，表明端点 1 有效。</td></tr>
<tr><td>4</td><td>E12AL1</td><td>当 2 级复用设置和链接到接口的复用设置被置为 1 时，该位使端点 1 有效。
1：在设置复用设置 1 时 CONF 位 = 1，使端点有效。
0：即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 （默认值）。
当 E15AL4 至 E15AL1 位为 0000 时，该位有效。</td></tr>
<tr><td>3 至 0</td><td>E15ALn</td><td>当 5 级复用设置和链接到接口的复用设置被设置为 n 时，这些位使端点 1 有效。
1：在设置复用设置 1 时 CONF 位 = 1，使端点有效。
0：即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 （默认值）。</td></tr>
</table>

备注 n = 1 至 4

（39）UF0 端点 2 接口映射寄存器（UF0E2IM）

该寄存器用于指定端点 2 对哪个接口和复用设置有效。

该寄存器可以按字节进行读取或写入。

该寄存器的设置和通过 SET_INTERFACE 请求选择的复用设置表明端点 2 当前是否有效，并且硬件决定如何响应 GET_STATUS/CLEAR_FEATURE/SET_FEATURE 端点 2 请求和到端点 2 的 IN 事务，以及相关的位是有效或是无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E2IM	E2EN2	E2EN1	E2EN0	E22AL1	E25AL4	E25AL3	E25AL2	E25AL1	00200088H	00H

<table>
<tr><th>位的位置</th><th>位名称</th><th>功能</th></tr>
<tr><td>7 至 5</td><td>E2EN2 至
E2EN0</td><td>这些位设置一个端点 2 的接口与 2 / 5 级复用设置之间的链接。端点被链接到复用设置 0。
链接到复用设置 0 的端点不能从复用设置 1 至 4 中排除。
<table>
<tr><th>E2EN2</th><th>E2EN1</th><th>E2EN0</th><th>链接状态</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td rowspan="2">没有与接口链接</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>与接口 4 和复用设置 0 链接</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>与接口 3 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>与接口 2 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>与接口 1 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>与接口 0 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>没有与接口链接（默认值）</td></tr>
</table>
当这些位被设置为 110 或 111 时，即使 E22AL1 位被清除为 0，它们也无效。
如果端点被链接，设置 UF0MODS 寄存器的 CONF 位为 1，表明端点 2 有效。</td></tr>
<tr><td>4</td><td>E22AL1</td><td>当 2 级复用设置和链接到接口的复用设置被置为 1 时，该位使端点 2 有效。
1：在设置复用设置 1 时 CONF 位 = 1，使端点有效。
0：即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 （默认值）。
当 E25AL4 至 E25AL1 位为 0000 时，该位有效。</td></tr>
<tr><td>3 至 0</td><td>E25ALn</td><td>当 5 级复用设置和链接到接口的复用设置被设置为 n 时，这些位使端点 2 有效。
1：在设置复用设置 1 时 CONF 位 = 1，使端点有效。
0：即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 （默认值）。</td></tr>
</table>

备注 n = 1 至 4

（40）UF0 端点 3 接口映射寄存器（UF0E3IM）

该寄存器用于指定端点 3 对哪个接口和复用设置有效。

该寄存器可以按字节进行读取或写入。

该寄存器的设置和通过 SET_INTERFACE 请求选择的复用设置表明端点 3 当前是否有效，并且硬件决定如何响应 GET_STATUS/CLEAR_FEATURE/SET_FEATURE 端点 3 请求和到端点 3 的 IN 事务，以及相关的位是有效或是无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E3IM	E3EN2	E3EN1	E3EN0	E32AL1	E35AL4	E35AL3	E35AL2	E35AL1	0020008AH	00H

位的位置	位名称	功能
7 至 5	E3EN2 至 E3EN0	这些位设置一个端点 3 的接口与 2 / 5 级复用设置之间的链接。端点被链接到复用设置 0。链接到复用设置 0 的端点不能从复用设置 1 至 4 中排除。 E3EN2 / E3EN1 / E3EN0 / 链接状态 1 / 1 / 1 / 没有与接口链接 1 / 1 / 0 1 / 0 / 1 / 与接口 4 和复用设置 0 链接 1 / 0 / 0 / 与接口 3 和复用设置 0 链接 0 / 1 / 1 / 与接口 2 和复用设置 0 链接 0 / 1 / 0 / 与接口 1 和复用设置 0 链接 0 / 0 / 1 / 与接口 0 和复用设置 0 链接 0 / 0 / 0 / 没有与接口链接（默认值） 当这些位被设置为 110 或 111 时，即使 E32AL1 位被清除为 0，它们也无效。 如果端点被链接，设置 UF0MODS 寄存器的 CONF 位为 1，表明端点 3 有效。
4	E32AL1	当 2 级复用设置和链接到接口的复用设置被置为 1 时，该位使端点 3 有效。 1：在设置复用设置 1 时 CONF 位 = 1，使端点有效。 0：即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 （默认值）。 当 E35AL4 至 E35AL1 位为 0000 时，该位有效。
3 至 0	E35ALn	当 5 级复用设置和链接到接口的复用设置被设置为 n 时，这些位使端点 3 有效。 1：在设置复用设置 1 时 CONF 位 = 1，使端点有效。 0：即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 （默认值）。

备注 n = 1 至 4

（41）UF0 端点 4 接口映射寄存器（UF0E4IM）

该寄存器用于指定端点 4 对哪个接口和复用设置有效。

该寄存器可以按字节进行读取或写入。

该寄存器的设置和通过 SET_INTERFACE 请求选择的复用设置表明端点 4 当前是否有效，并且硬件决定如何响应 GET_STATUS/CLEAR_FEATURE/SET_FEATURE 端点 4 请求和到端点 4 的 IN 事务，以及相关的位是有效或是无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E4IM	E4EN2	E4EN1	E4EN0	E42AL1	E45AL4	E45AL3	E45AL2	E45AL1	0020008CH	00H

<table>
<tr><th>位的位置</th><th>位名称</th><th>功能</th></tr>
<tr><td>7 至 5</td><td>E4EN2 至
E4EN0</td><td>这些位设置一个端点 4 的接口与 2 / 5 级复用设置之间的链接。端点被链接到复用设置 0。
链接到复用设置 0 的端点不能从复用设置 1 至 4 中排除。
<table>
<tr><th>E4EN2</th><th>E4EN1</th><th>E4EN0</th><th>链接状态</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td rowspan="2">没有与接口链接</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>与接口 4 和复用设置 0 链接</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>与接口 3 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>与接口 2 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>与接口 1 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>与接口 0 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>没有与接口链接（默认值）</td></tr>
</table>
当这些位被设置为 110 或 111 时，即使 E42AL1 位被清除为 0，它们也无效。
如果端点被链接，设置 UF0MODS 寄存器的 CONF 位为 1，表明端点 4 有效。</td></tr>
<tr><td>4</td><td>E42AL1</td><td>当 2 级复用设置和链接到接口的复用设置被置为 1 时，该位使端点 4 有效。
1：在设置复用设置 1 时 CONF 位 = 1，使端点有效。
0：即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 （默认值）。
当 E45AL4 至 E45AL1 位为 0000 时，该位有效。</td></tr>
<tr><td>3 至 0</td><td>E45ALn</td><td>当 5 级复用设置和链接到接口的复用设置被设置为 n 时，这些位使端点 4 有效。
1：在设置复用设置 1 时 CONF 位 = 1，使端点有效。
0：即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 （默认值）。</td></tr>
</table>

备注 n = 1 至 4

（42）UF0 端点 7 接口映射寄存器（UF0E7IM）

该寄存器用于指定端点 7 对哪个接口和复用设置有效。

该寄存器可以按字节进行读取或写入。

该寄存器的设置和通过 SET_INTERFACE 请求选择的复用设置表明端点 7 当前是否有效，并且硬件决定如何响应 GET_STATUS/CLEAR_FEATURE/SET_FEATURE 端点 7 请求和到端点 7 的 IN 事务，以及相关的位是有效或是无效。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E7IM	E7EN2	E7EN1	E7EN0	E72AL1	E75AL4	E75AL3	E75AL2	E75AL1	0020008CH	00H

<table>
<tr><th>位的位置</th><th>位名称</th><th>功能</th></tr>
<tr><td>7 至 5</td><td>E7EN2 至
E7EN0</td><td>这些位设置一个端点 7 的接口与 2 / 5 级复用设置之间的链接。端点被链接到复用设置 0。
链接到复用设置 0 的端点不能从复用设置 1 至 4 中排除。
<table>
<tr><th>E7EN2</th><th>E7EN1</th><th>E7EN0</th><th>链接状态</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td rowspan="2">没有与接口链接</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>与接口 4 和复用设置 0 链接</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>与接口 3 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>与接口 2 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>与接口 1 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>与接口 0 和复用设置 0 链接</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>没有与接口链接（默认值）</td></tr>
</table>
当这些位被设置为 110 或 111 时，即使 E72AL1 位被清除为 0，它们也无效。
如果端点被链接，设置 UF0MODS 寄存器的 CONF 位为 1，表明端点 7 有效。</td></tr>
<tr><td>4</td><td>E72AL1</td><td>当 2 级复用设置和链接到接口的复用设置被置为 1 时，该位使端点 7 有效。
1：在设置复用设置 1 时 CONF 位 = 1，使端点有效。
0：即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 （默认值）。
当 E75AL4 至 E75AL1 位为 0000 时，该位有效。</td></tr>
<tr><td>3 至 0</td><td>E75ALn</td><td>当 5 级复用设置和链接到接口的复用设置被设置为 n 时，这些位使端点 7 有效。
1：在设置复用设置 1 时 CONF 位 = 1，使端点有效。
0：即使在设置复用设置 1 时 CONF 位 = 1，不使端点有效 （默认值）。</td></tr>
</table>

备注 n = 1 至 4

20.6.4 数据保持寄存器

（1）UF0 EP0 读取寄存器（UF0E0R）

UF0E0R 寄存器是一个 64 字节 FIFO，用于存储到/从端点 0 控制传输的数据阶段中从主机发送的 OUT 数据。

该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当从主机接收到数据时，硬件将数据自动传输到 UF0E0R 寄存器。当数据被正确接收后，UF0IS1 寄存器的 E0ODT 位被置为 1。UF0E0L 保持已接收到数据的数量，并且发出中断请求（INTUSBF0）。当 UF0E0L 寄存器接收数据时，它总是更新已接收到数据的长度。如果最后传输是正确接收，产生中断请求。如果接收异常，UF0E0L 寄存器被清除为 0，并且不产生中断请求。

UF0E0R 寄存器保持的数据必须通过 FW 读取，读取的数量达到 UF0E0L 寄存器数据读取的数量值。使用 UF0EPS0 寄存器的 EP0R 位来检查所有数据已经被读取（当所有数据已经被读取时，EP0R = 0 ）。如果 UF0E0L 寄存器的值是 0，UF0E0N 寄存器的 EP0NKR 位被清除为 0，并且 UF0E0R 寄存器准备好接收。当接收到下一个 SETUP 令牌，UF0E0R 寄存器被清除。

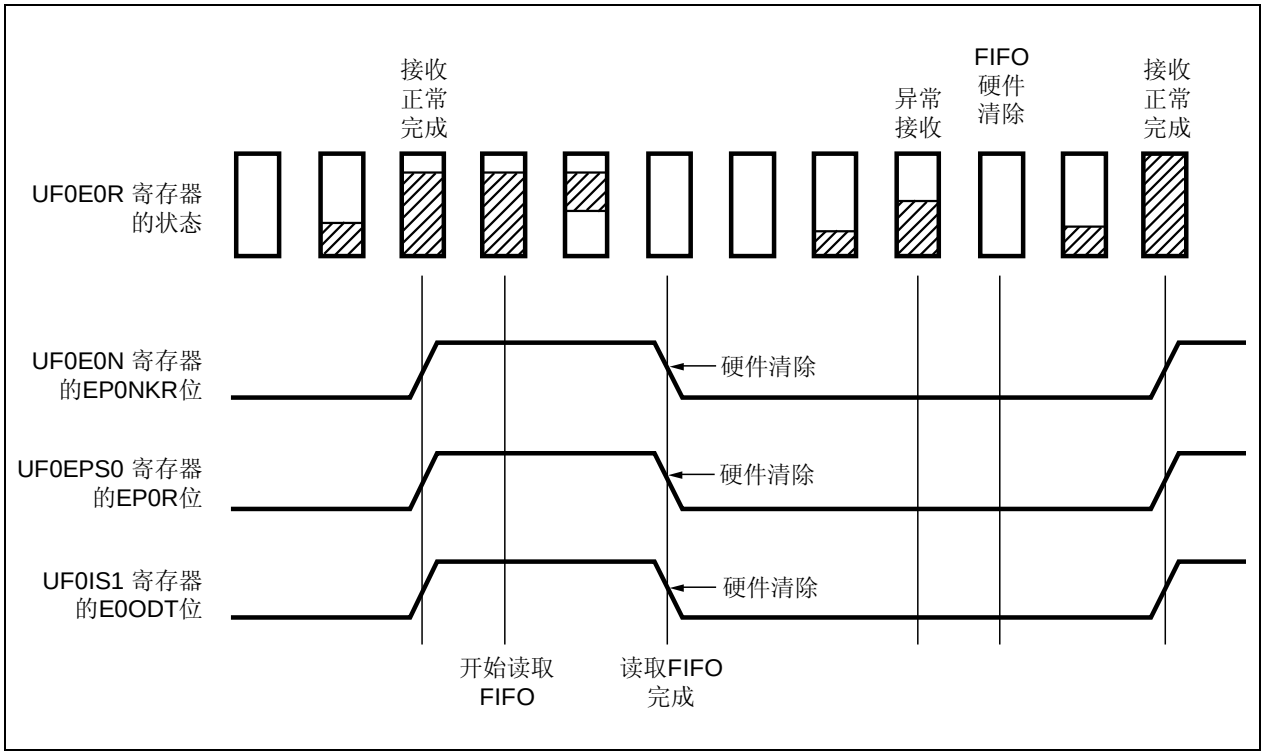
注意事项 读取所有保存的数据。清除 FIFO 来丢弃一些数据。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0R	E0R7	E0R6	E0R5	E0R4	E0R3	E0R2	E0R1	E0R0	00200100H	不确定

位的位置	位名称	功能
7 至 0	E0R7 至 E0R0	这些位存储到/从端点 0 控制传输的数据阶段中从主机发送的 OUT 数据

UF0E0R 寄存器的操作如下所示。

图 20-4. UF0E0R 寄存器的操作



（2）UF0 EP0 长度寄存器（UF0E0L）

UF0E0L 寄存器用于存储保持在 UF0E0R 寄存器中的数据的长度。

该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当接收数据时，UF0E0L 寄存器总是更新已接收到数据的长度。如果最后传输是异常接收，UF0E0L 寄存器被清除为 0，并且不产生中断请求。只有当接收正常，读取 UF0E0L 寄存器的数据长度，并 FW 按照设定数量从 UF0E0R 寄存器读取足够的数据，才会产生中断请求。每次 UF0E0R 寄存器被读取后，UF0E0L 寄存器的值递减。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0L	E0L7	E0L6	E0L5	E0L4	E0L3	E0L2	E0L1	E0L0	00200102H	不确定

位的位置	位名称	功能
7 至 0	E0L7 至 E0L0	这些位存储保持在 UF0E0R 寄存器中的数据的长度。

（3）UF0 EP0 建立寄存器（UF0E0ST）

UF0E0ST 寄存器用于保持从主机发送的 SETUP 数据。

该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当接收到 SETUP 事务时，UF0E0ST 寄存器总是写入数据。当已经正确接收 SETUP 事务后，硬件对 UF0IS1 寄存器的 PROT 位置位。在由 FW 处理请求的情况下，对 UF0IS1 寄存器的 CPUDEC 位置位，然后发出中断请求（INTUSBF0）。在由 FW 处理请求的情况下，确保按照 8 位宽度读取请求。如果不是按照 8 位宽度读取，后续请求不能被正确译码。即使接收到总线复位时，UF0E0ST 寄存器的读取计数器也不会被清除。不管是否接收到总线复位，总是按照 8 位宽度读取这个计数器。

因为 UF0E0ST 寄存器总是允许写入，即使正在读取寄存器数据时接收到一个 SETUP 事务，硬件也会覆盖该寄存器的数据。即使 SETUP 事务未能被正确接收，也不会产生 CPUDEC 中断请求和 Protect 中断请求，但是之前的数据被丢弃。但是，如果接收到的 SETUP 令牌少于 8 字节，接收的 SETUP 令牌被丢弃，并且之前接收的 SETUP 数据被保留。如果当控制传输执行了一次，接收到多次 SETUP 令牌，在以下条件下，确保检查 UF0IS1 寄存器的 PROT 位。如果 PROT 位 = 1，再次读取 UF0E0ST 寄存器，因为 SETUP 事务已经被多次接收。

<1> 如果由 FW 对请求进行译码，且 UF0E0R 寄存器被读取或者 UF0E0W 寄存器被写入

<2> 为该请求准备一个 STALL 响应时，译码结果对其不响应。

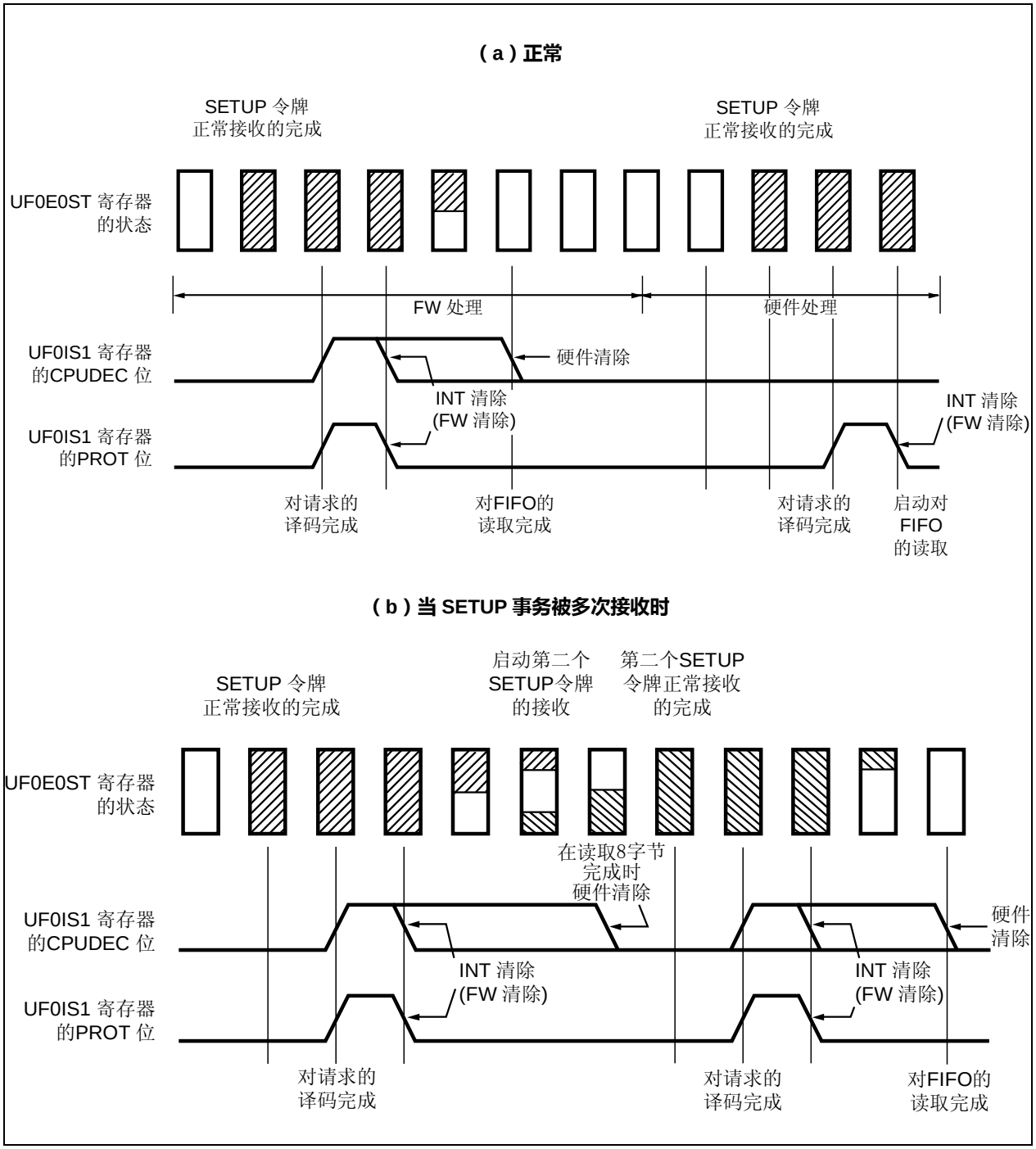
注意事项 确保读取所有保存的数据。UF0E0ST 寄存器总是由 SETUP 事务中的请求来更新。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0ST	E0S7	E0S6	E0S5	E0S4	E0S3	E0S2	E0S1	E0S0	00200104H	00H

位的位置	位名称	功能
7 至 0	E0S7 至 E0S0	这些位保持从主机发送的 SETUP 数据。

UF0E0ST 寄存器的操作如下所示。

图 20-5. UF0E0ST 寄存器的操作



（4）UF0 EP0 写入寄存器（UF0E0W）

UF0E0W 寄存器是一个 64 字节 FIFO，用于保存到端点 0 的数据阶段中发送到主机的 IN 数据（传递到 SIE）。

该寄存器是只能写入，按字节进行写入。如果读取该寄存器，读到的值是 00H。

只有当 UF0E0N 寄存器的 EP0NKW 位被置为 1 时（NAK 没有被发送时），硬件和 IN 令牌同步，发送数据到 USB 总线。当数据已发送且主机正确接收该数据时，UF0E0N 寄存器的 EP0NKW 位由硬件自动清除为 0。当数据已写入 UF0E0W 寄存器且 UF0DEND 寄存器的 E0DED 位被置为 1（UF0EPS0 寄存器的 EP0W 位 = 1 （数据存在））时，发送一个短包。当 UF0E0W 寄存器已清除且 UF0DEND 寄存器的 E0DED 位被置为 1（UF0EPS0 寄存器的 EP0W 位 = 1 （数据存在））时，发送一个空包。

发送没尚未完成期间，接收到下一个 SETUP 令牌时，UF0E0W 寄存器被清除为 0。在数据阶段中 ACK 没有被正确接收时，如果控制传输（读）的阶段更改为状态阶段，UF0E0W 寄存器被自动清除为 0。同时，如果 UF0E0N 寄存器的 EP0NKW 位为 1，它也被清除为 0。

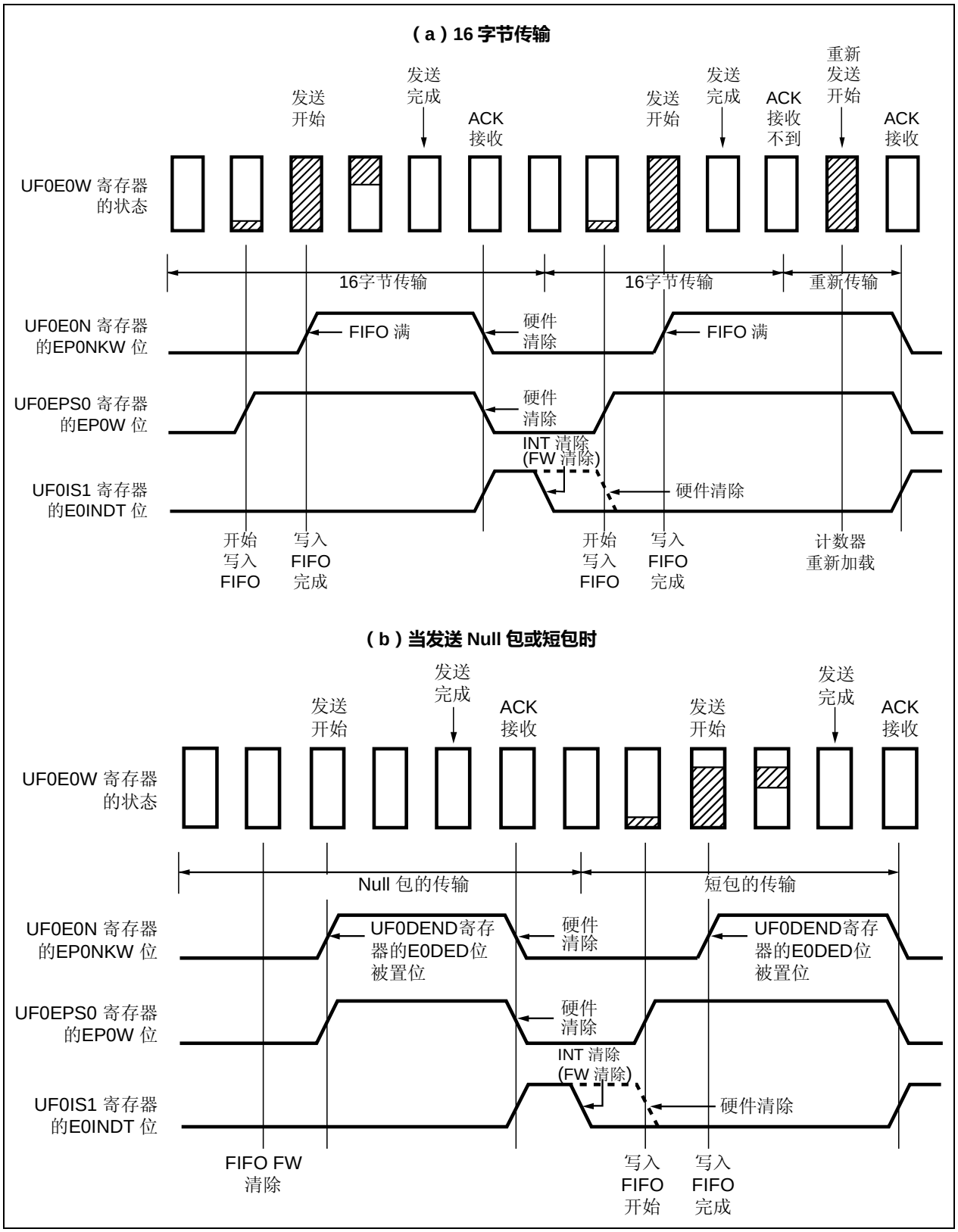
没有数据存于 UF0E0W 寄存器中时，如果被读取，读到的值是 00H。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0W	E0W7	E0W6	E0W5	E0W4	E0W3	E0W2	E0W1	E0W0	00200106H	不确定

位的位置	位名称	功能
7 至 0	E0W7 至 E0W0	这些位保存到端点 0 的数据阶段中发送到主机的 IN 数据。

UF0E0W 寄存器的操作如下所示。

图 20-6. UF0E0W 寄存器的操作



（5）UF0 批量-out 1 寄存器（UF0BO1）

UF0BO1 寄存器是一个 64 字节× 2 的 FIFO，用于保存进出端点 2 的数据。该寄存器由 64 字节 FIFO 的两个 bank 组成，每个 bank 执行切换操作并在 SIE 和 CPU 端反复连接总线。当数据位于 SIE 端的 FIFO 中且 CPU 端的 FIFO 中没有数据（计数器值 = 0）时，切换操作发生。

该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当硬件从主机接收到端点 2 的数据时，自动将数据传送到 UF0BO1 寄存器。当该寄存器正确接收数据时，发生 FIFO 切换操作。因此，UF0IS3 寄存器的 BKO1DT 位被置为 1，接收到的数据的数量被保持在 UF0BO1L 寄存器中，并且向 CPU 发出中断请求或 DMA 请求。可以通过 UF0IDR 寄存器的 DQBO1MS 位来选择是发出中断请求或发出 DMA 请求。

通过 FW 读取 UF0BO1 寄存器中保持的数据，读取的数据的数量由 UF0BO1L 寄存器决定。当正确接收的数据被连接到 SIE 端的 FIFO 保持，且 UF0BO1L 寄存器的值达到 0 时，发生 FIFO 的切换操作，并且 UF0EN 寄存器的 BKO1NK 位被自动清除为 0。如果读取的数据的数量多于 UF0BO1L 寄存器的值，且满足 FIFO 切换条件，则发生 FIFO 的切换操作。因此，下一个包可能被错误读取。注意，如果不满足切换条件，第一个数据需要重复读取。

数据被保持在连接到 CPU 端的 FIFO 时，如果接收到溢出数据，端点 2 停顿，并且 CPU 端的 FIFO 被清除。

没有数据存于 UF0BO1 寄存器中时，如果被读取，读到的值不确定。

注意事项 确保读取所有保存在该寄存器中的数据。

	7	6	5	4	3	2	1	0	地址	复位后
UF0BO1	BKO17	BKO16	BKO15	BKO14	BKO13	BKO12	BKO11	BKO10	00200108H	不确定

位的位置	位名称	功能
7 至 0	BKO17 至 BKO10	这些位保存进出端点 2 的数据。

UF0BO1 寄存器的操作如下所示。

图 20-7. UF0BO1 寄存器的操作（1/2）

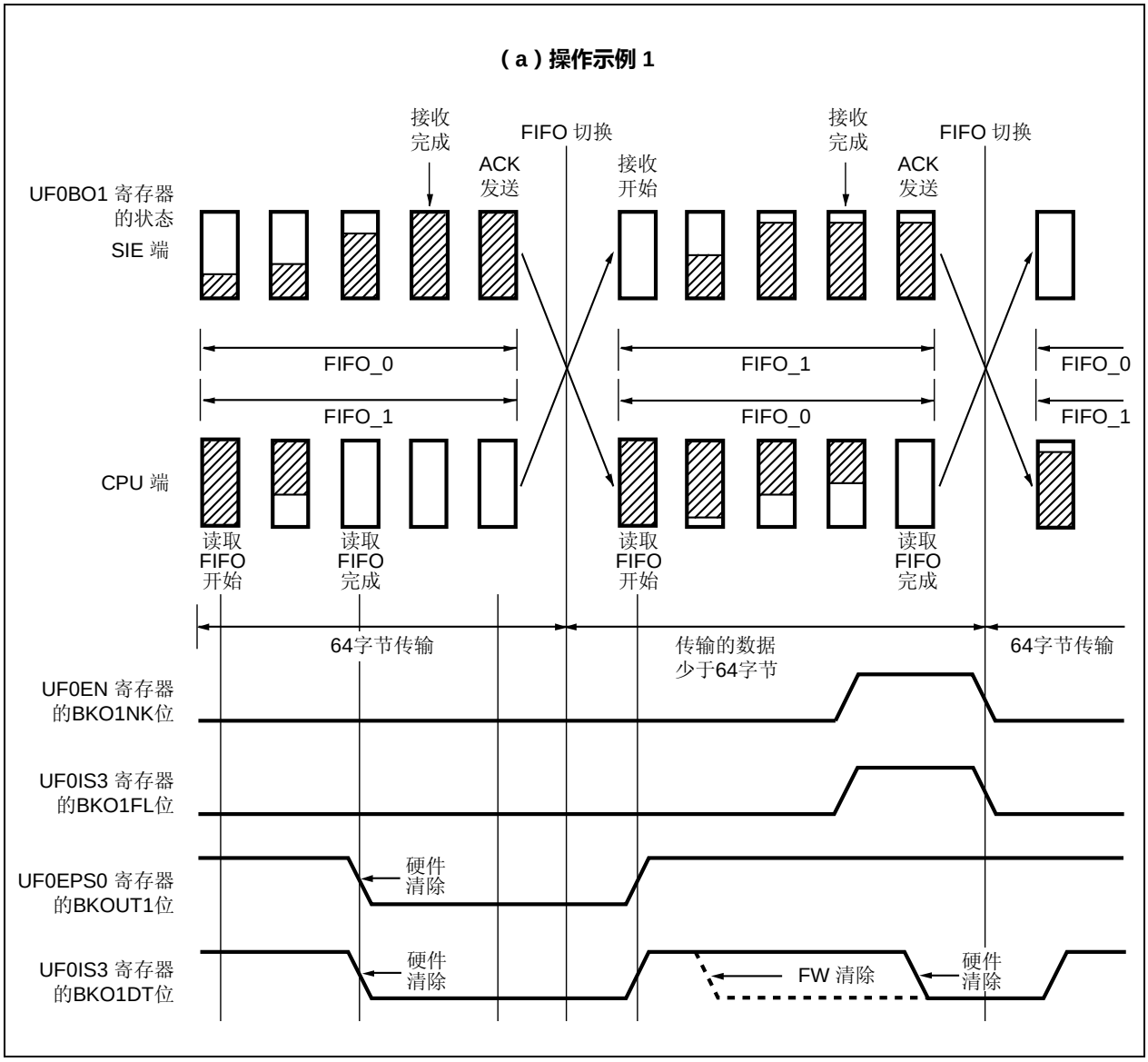
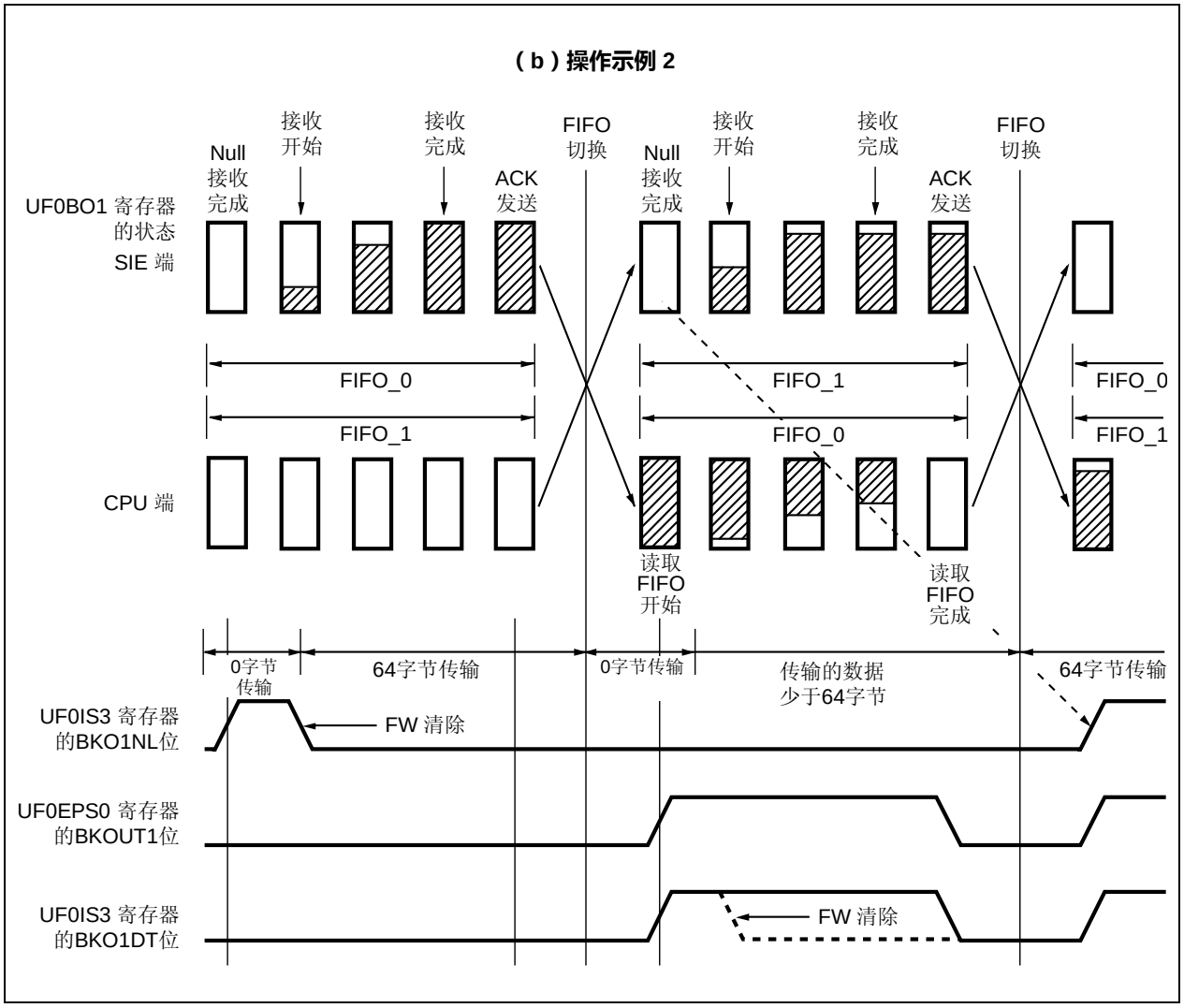


图 20-7. UF0BO1 寄存器的操作（2/2）



（6）UF0 批量-out 1 长度寄存器（UF0BO1L）

UF0BO1L 寄存器用于存储保持在 UF0BO1 寄存器中的数据的长度。

该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当接收数据时，UF0BO1L 寄存器总是更新已接收到的数据的长度。如果最后传输是异常接收，UF0BO1L 寄存器被清除为 0 且不产生中断请求。只有当接收正常，才会产生中断请求，且通过 FW 读取 UF0BO1 寄存器中保持的数据，读取的数据的数量由 UF0BO1L 寄存器决定。每次 UF0BO1 寄存器被读取后，UF0BO1L 寄存器的值递减。

	7	6	5	4	3	2	1	0	地址	复位后
UF0BO1L	BKO1L7	BKO1L6	BKO1L5	BKO1L4	BKO1L3	BKO1L2	BKO1L1	BKO1L0	0020010AH	00H

位的位置	位名称	功能
7 至 0	BKO1L7 至 BKO1L0	这些位存储保持在 UF0BO1 寄存器中的数据的长度。

（7）UF0 批量-out 2 寄存器（UF0BO2）

UF0BO2 寄存器是一个 64 字节× 2 的 FIFO，用于保存进出端点 4 的数据。该寄存器由 64 字节 FIFO 的两个 bank 组成，每个 bank 执行切换操作并在 SIE 和 CPU 端反复连接总线。当数据位于 SIE 端的 FIFO 中且 CPU 端的 FIFO 中没有数据（计数器值 = 0）时，切换操作发生。

该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当硬件从主机接收到端点 4 的数据时，自动将数据传送到 UF0BO2 寄存器。当该寄存器正确接收数据时，发生 FIFO 切换操作。因此，UF0IS3 寄存器的 BKO2DT 位被置为 1，接收到的数据的数量被保持在 UF0BO2L 寄存器中，并且向 CPU 发出中断请求或 DMA 请求。可以通过 UF0IDR 寄存器的 DQBO2MS 位来选择是发出中断请求或发出 DMA 请求。

通过 FW 读取 UF0BO2 寄存器中保持的数据，读取的数据的数量由 UF0BO2L 寄存器决定。当正确接收的数据被连接到 SIE 端的 FIFO 保持，且 UF0BO2L 寄存器的值达到 0 时，发生 FIFO 的切换操作，并且 UF0EN 寄存器的 BKO2NK 位被自动清除为 0。如果读取的数据的数量多于 UF0BO2L 寄存器的值，且满足 FIFO 切换条件，则发生 FIFO 的切换操作。因此，下一个包可能被错误读取。注意，如果不满足切换条件，第一个数据需要重复读取。

数据被保持在连接到 CPU 端的 FIFO 时，如果接收到溢出数据，端点 4 停顿，并且 CPU 端的 FIFO 被清除。

没有数据存于 UF0BO2 寄存器中时，如果被读取，读到的值不确定。

注意事项 确保读取所有保存在该寄存器中的数据。

	7	6	5	4	3	2	1	0	地址	复位后
UF0BO2	BKO27	BKO26	BKO25	BKO24	BKO23	BKO22	BKO21	BKO20	0020010CH	不确定

位的位置	位名称	功能
7 至 0	BKO27 至 BKO20	这些位保存进出端点 4 的数据。

UF0BO2 寄存器的操作如下所示。

图 20-8. UF0BO2 寄存器的操作（1/2）

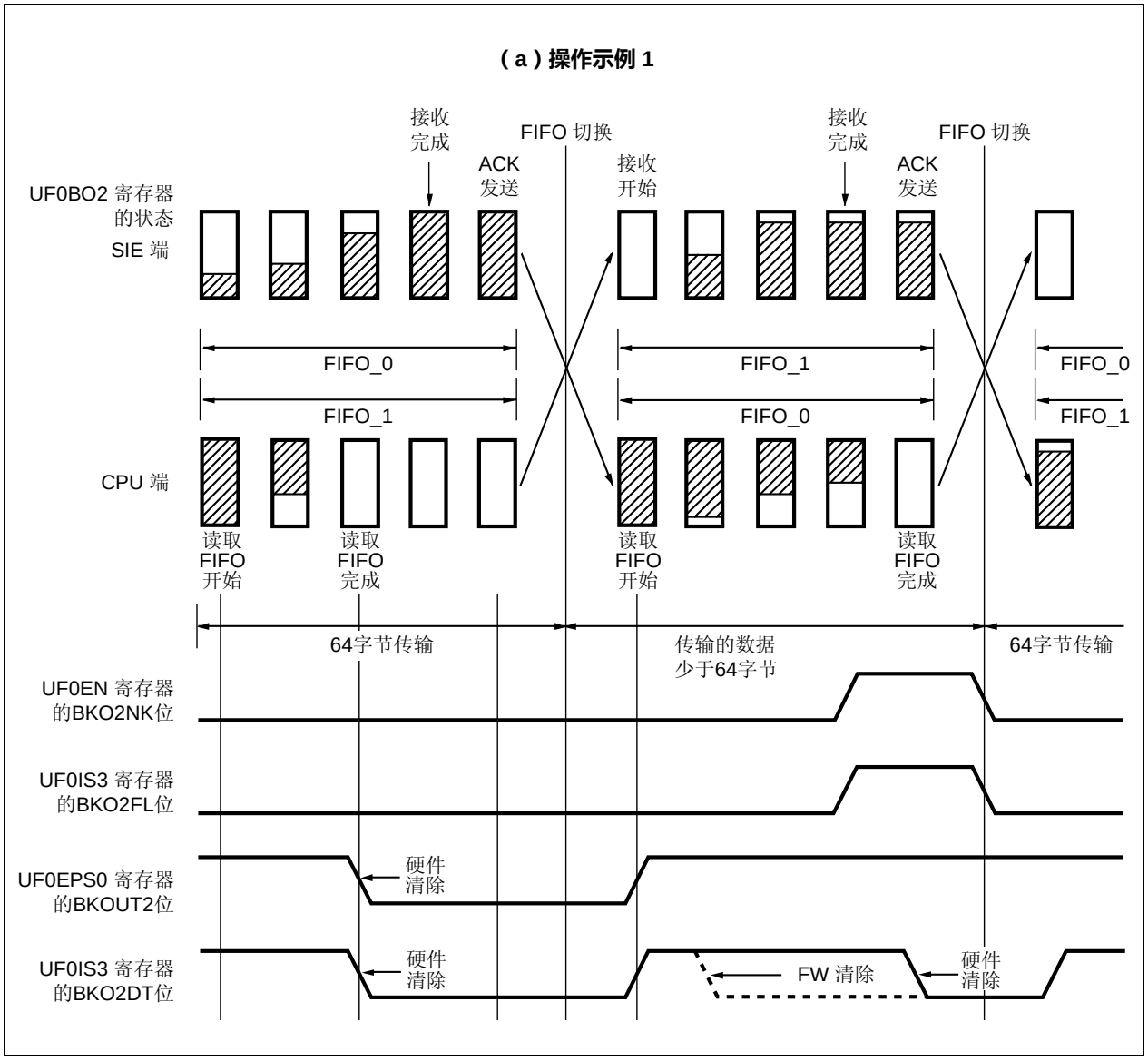
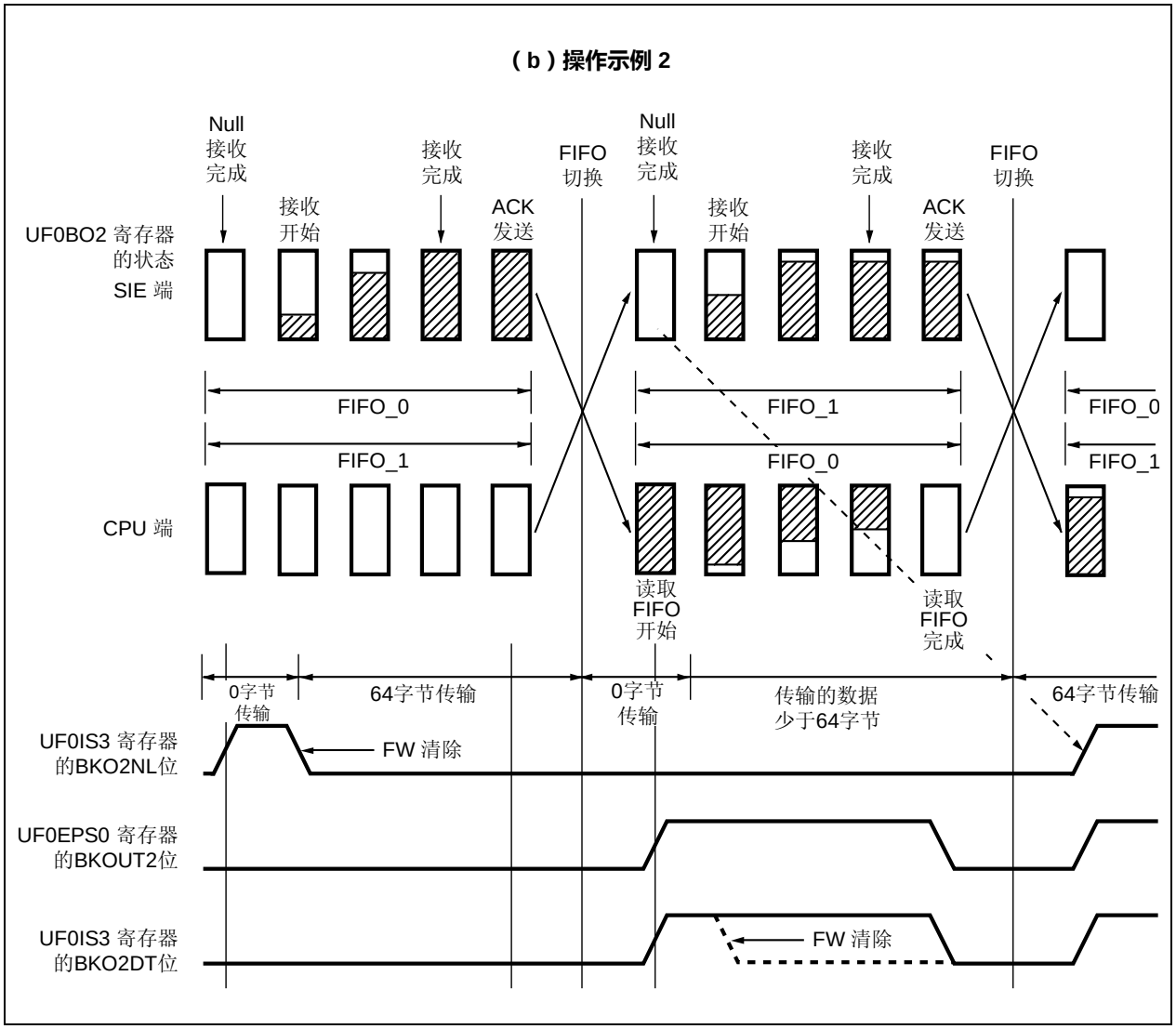


图 20-8. UF0BO2 寄存器的操作（2/2）



（8）UF0 批量-out 2 长度寄存器（UF0BO2L）

UF0BO2L 寄存器用于存储保持在 UF0BO2 寄存器中的数据的长度。

该寄存器是只读的，按字节进行读取。对该寄存器的写入访问被忽略。

当接收数据时，UF0BO2L 寄存器总是更新已接收到的数据的长度。如果最后传输是异常接收，UF0BO2L 寄存器被清除为 0 且不产生中断请求。只有当接收正常，才会产生中断请求，且通过 FW 读取 UF0BO2 寄存器中保持的数据，读取的数据的数量由 UF0BO2L 寄存器决定。每次 UF0BO2 寄存器被读取后，UF0BO2L 寄存器的值递减。

	7	6	5	4	3	2	1	0	地址	复位后
UF0BO2L	BKO2L7	BKO2L6	BKO2L5	BKO2L4	BKO2L3	BKO2L2	BKO2L1	BKO2L0	0020010EH	00H

位的位置	位名称	功能
7 至 0	BKO2L7 至 BKO2L0	这些位存储保持在 UF0BO2 寄存器中的数据的长度。

（9）UF0 批量-in 1 寄存器（UF0BI1）

UF0BI1 寄存器是一个 64 字节× 2 的 FIFO，用于保存进出端点 1 的数据。该寄存器由 64 字节 FIFO 的两个 bank 组成，每个 bank 执行切换操作并在 SIE 和 CPU 端反复连接总线。当 SIE 端的 FIFO 中没有数据（计数器值 = 0）且 CPU 端的 FIFO 被正确写入（FIFO 满或 BKI1DED 位 = 1）时，发生切换操作。

该寄存器是只能写入，按字节进行写入。如果读取该寄存器，读到的值是 00H。

只有当 UF0E0N 寄存器的 BKI1NK 位被置为 1 时（NAK 没有被发送时），硬件和对端点 1 的 IN 令牌同步，发送数据到 USB 总线。被写入或读取的数据地址由硬件管理。因此，只需按顺序将数据写入到 UF0BI1 寄存器，FW 就可以将数据发送到主机。当数据已写入 UF0BI1 寄存器且 UF0DEND 寄存器的 BKI1DED 位被置为 1（UF0EPS0 寄存器的 BKIN1 位 = 1 （数据存在））时，发送一个短包。当 UF0BI1 寄存器已清除且 UF0DEND 寄存器的 BKI1DED 位被置为 1（UF0EPS0 寄存器的 BKIN1 位 = 1 （数据存在））时，发送一个空包。当数据被正确发送时，发生 FIFO 切换操作。UF0IS2 寄存器的 BKI1DT 位被置为 1，并且向 CPU 发送中断请求。可以通过 UF0IDR 寄存器的 DQBI1MS 位来选择是中断请求或 DMA 请求。

	7	6	5	4	3	2	1	0	地址	复位后
UF0BI1	BKI17	BKI16	BKI15	BKI14	BKI13	BKI12	BKI11	BKI10	00200110H	不确定

位的位置	位名称	功能
7 至 0	BKI17 至 BKI10	这些位保存进出端点 1 的数据。

UF0BI1 寄存器的操作如下所示。

图 20-9. UF0BI1 寄存器的操作（1/3）

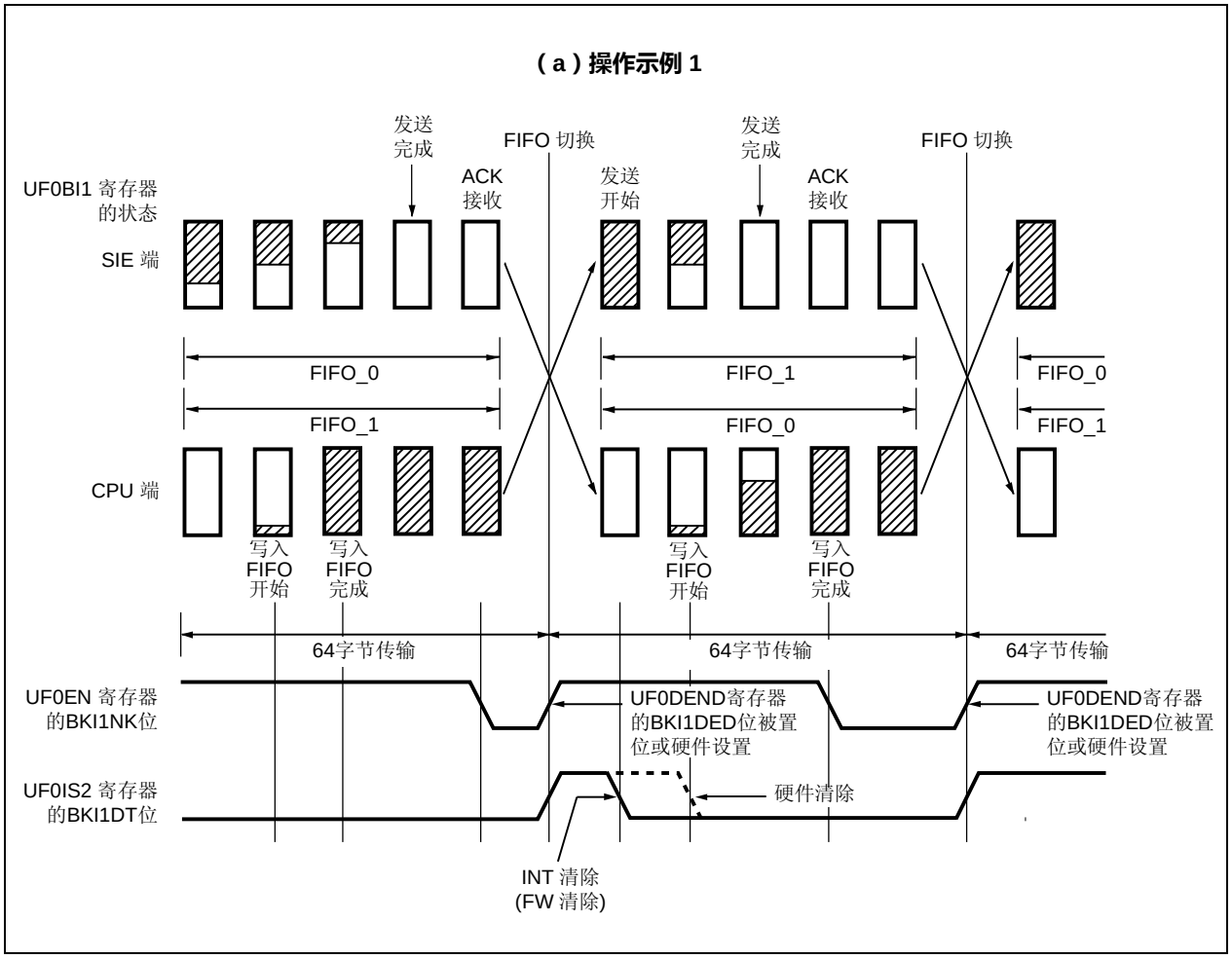


图 20-9. UF0BI1 寄存器的操作（2/3）

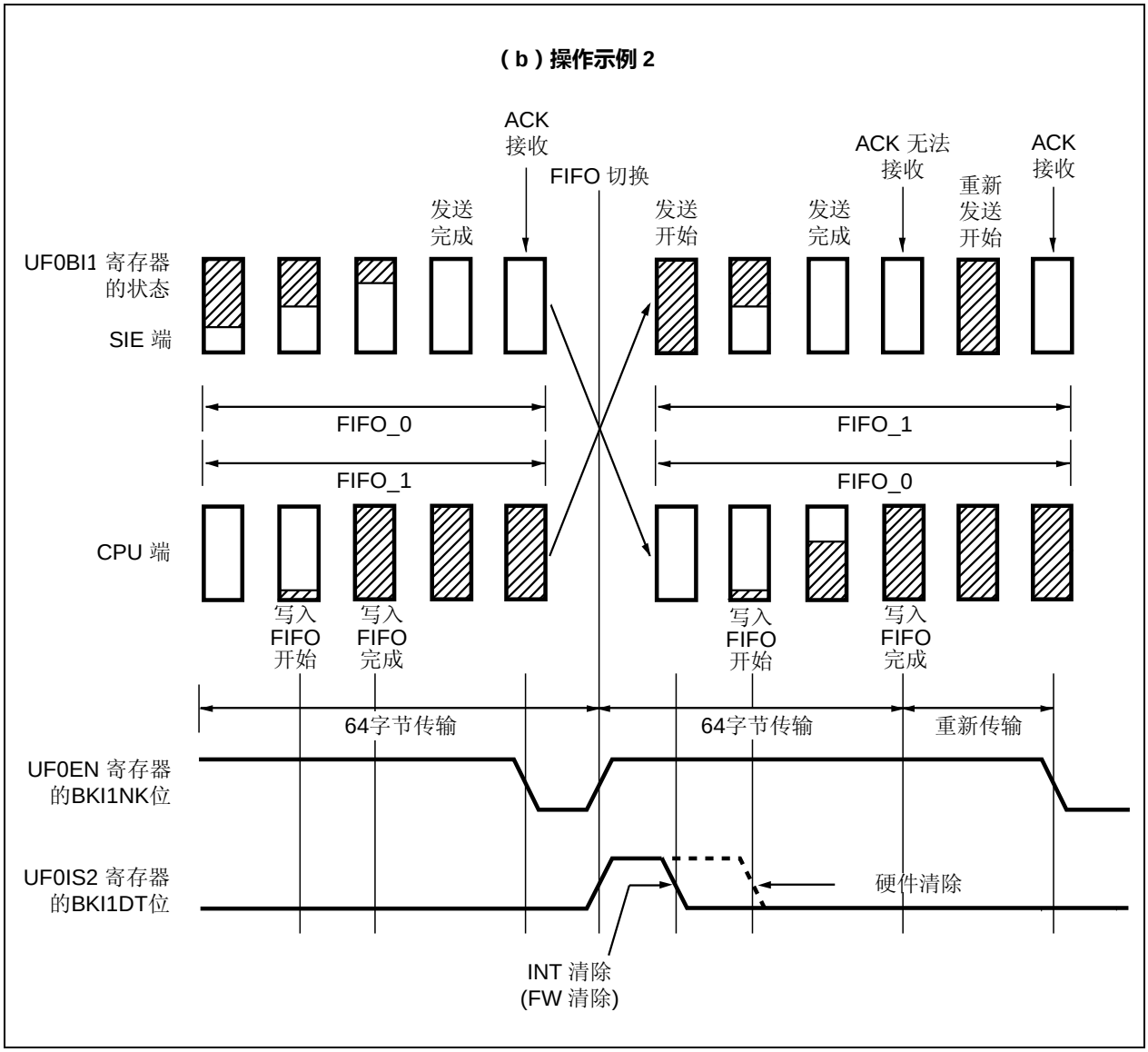
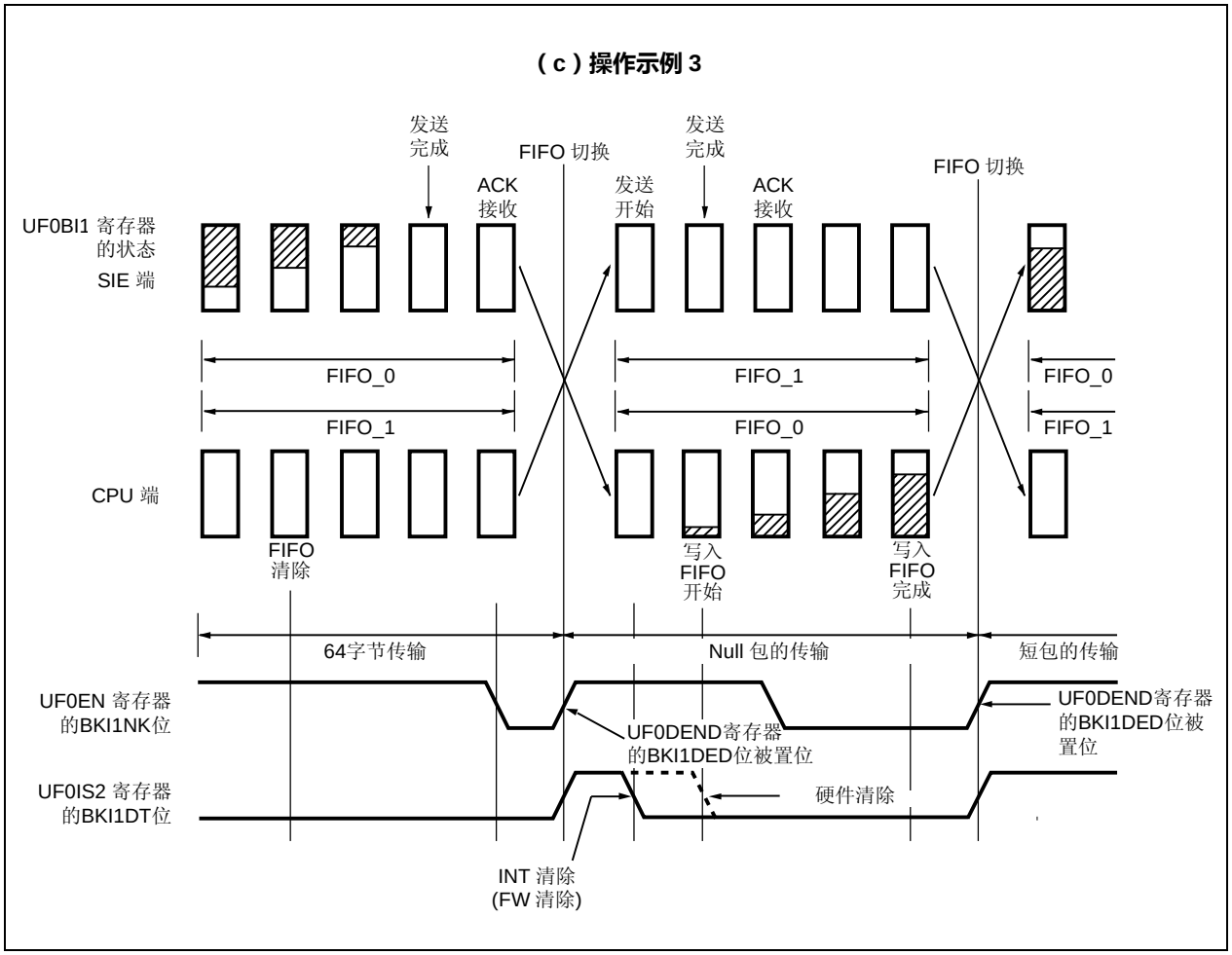


图 20-9. UF0BI1 寄存器的操作（3/3）



（10）UF0 批量-in 2 寄存器（UF0BI2）

UF0BI2 寄存器是一个 64 字节× 2 的 FIFO，用于保存进出端点 3 的数据。该寄存器由 64 字节 FIFO 的两个 bank 组成，每个 bank 执行切换操作并在 SIE 和 CPU 端反复连接总线。当 SIE 端的 FIFO 中没有数据（计数器值 = 0）且 CPU 端的 FIFO 被正确写入（FIFO 满或 BKI3DED 位 = 1）时，发生切换操作。

该寄存器是只能写入，按字节进行写入。如果读取该寄存器，读到的值是 00H。

只有当 UF0E0N 寄存器的 BKI2NK 位被置为 1 时（NAK 没有被发送时），硬件和对端点 3 的 IN 令牌同步，发送数据到 USB 总线。被写入或读取的数据地址由硬件管理。因此，只需按顺序将数据写入到 UF0BI2 寄存器，FW 就可以将数据发送到主机。当数据已写入 UF0BI2 寄存器且 UF0DEND 寄存器的 BKI2DED 位被置为 1（UF0EPS0 寄存器的 BKIN2 位 = 1 （数据存在））时，发送一个短包。当 UF0BI2 寄存器已清除且 UF0DEND 寄存器的 BKI2DED 位被置为 1（UF0EPS0 寄存器的 BKIN2 位 = 1 （数据存在））时，发送一个空包。当数据被正确发送时，发生 FIFO 切换操作。UF0IS2 寄存器的 BKI2DT 位被置为 1，并且向 CPU 发送中断请求。可以通过 UF0IDR 寄存器的 DQBI2MS 位来选择是中断请求或 DMA 请求。

	7	6	5	4	3	2	1	0	地址	复位后
UF0BI2	BKI27	BKI26	BKI25	BKI24	BKI23	BKI22	BKI21	BKI20	00200112H	不确定

位的位置	位名称	功能
7 至 0	BKI27 至 BKI20	这些位保存进出端点 3 的数据。

UF0BI2 寄存器的操作如下所示。

图 20-10. UF0BI2 寄存器的操作（1/3）

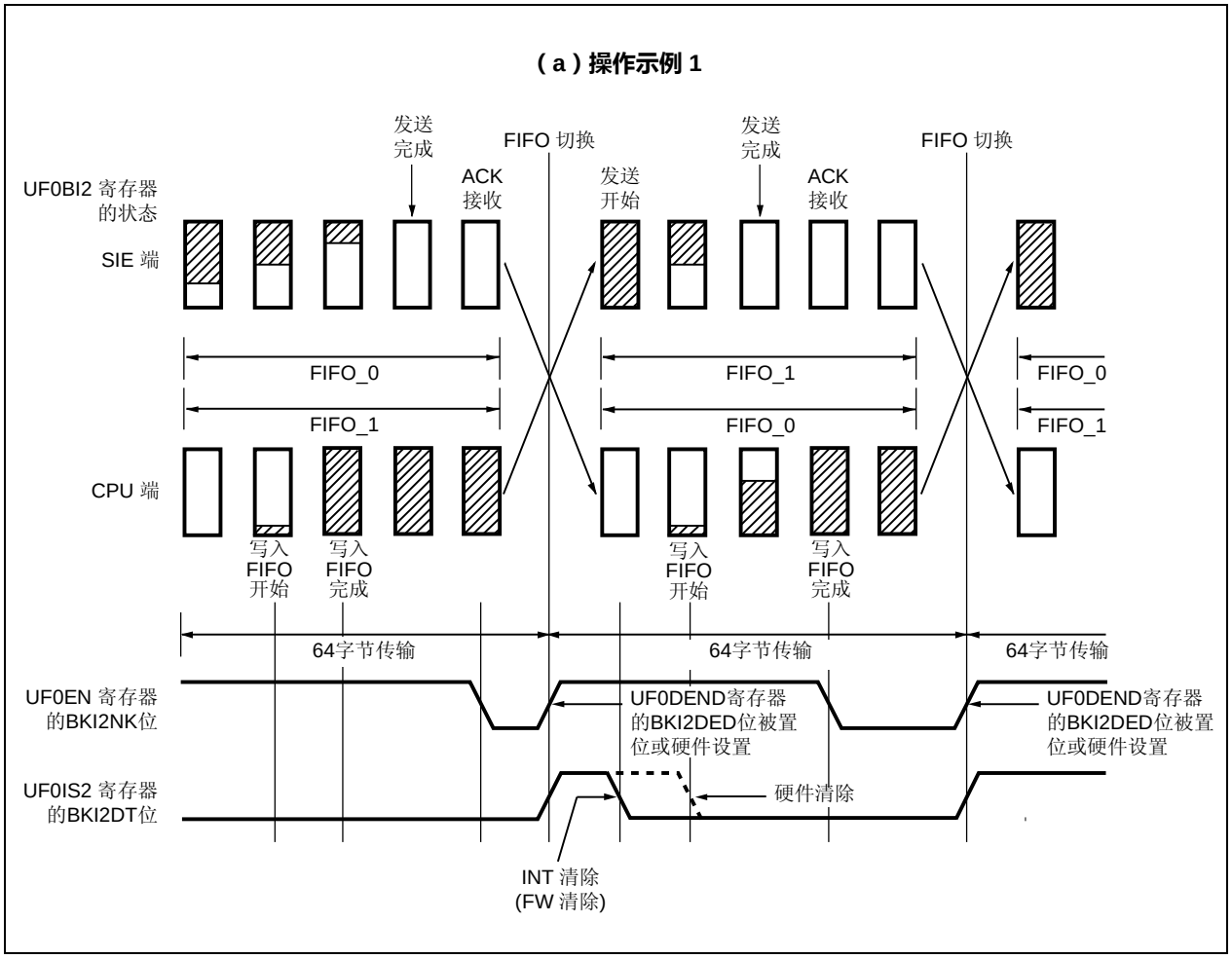


图 20-10. UF0BI2 寄存器的操作（2/3）

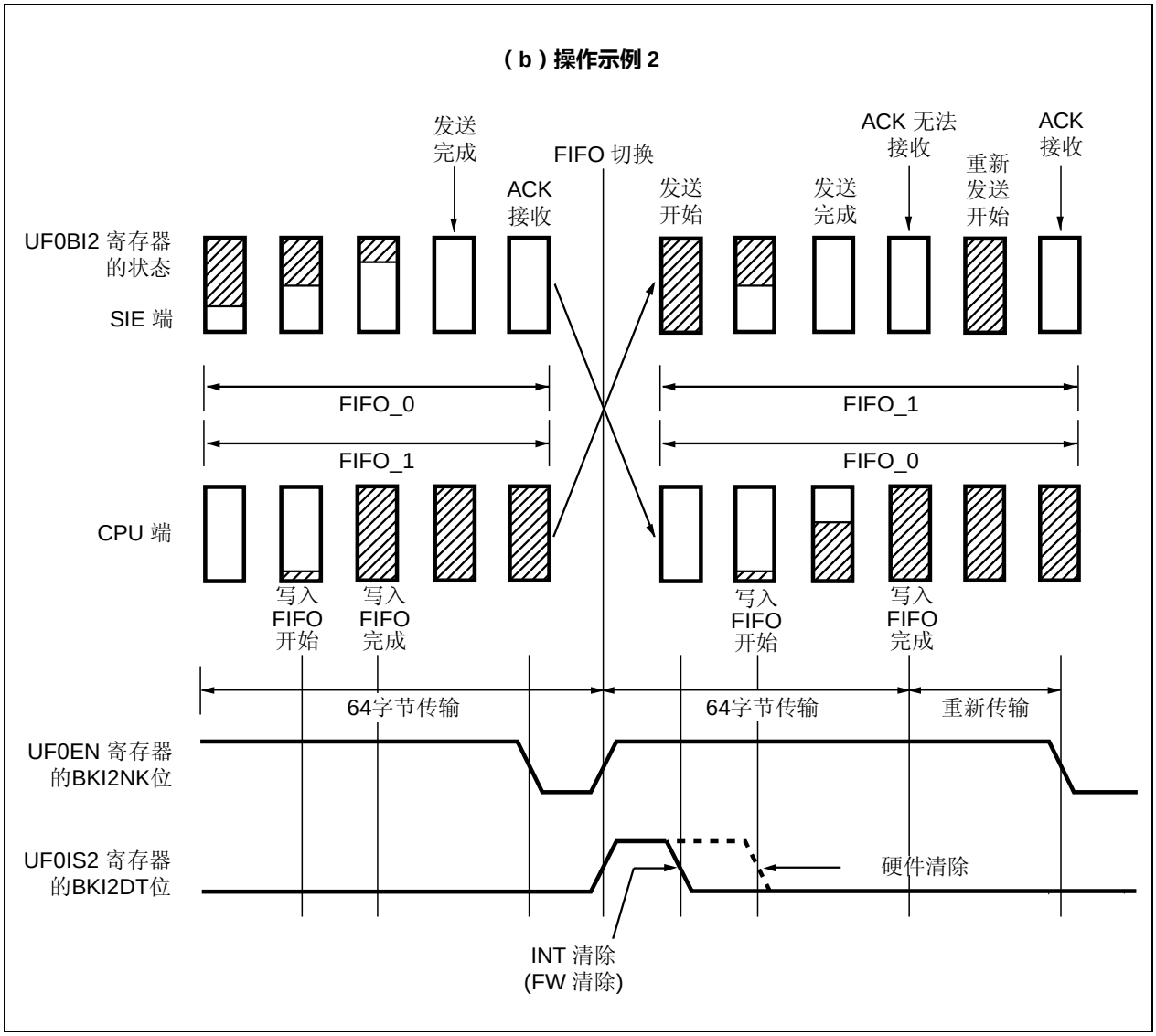
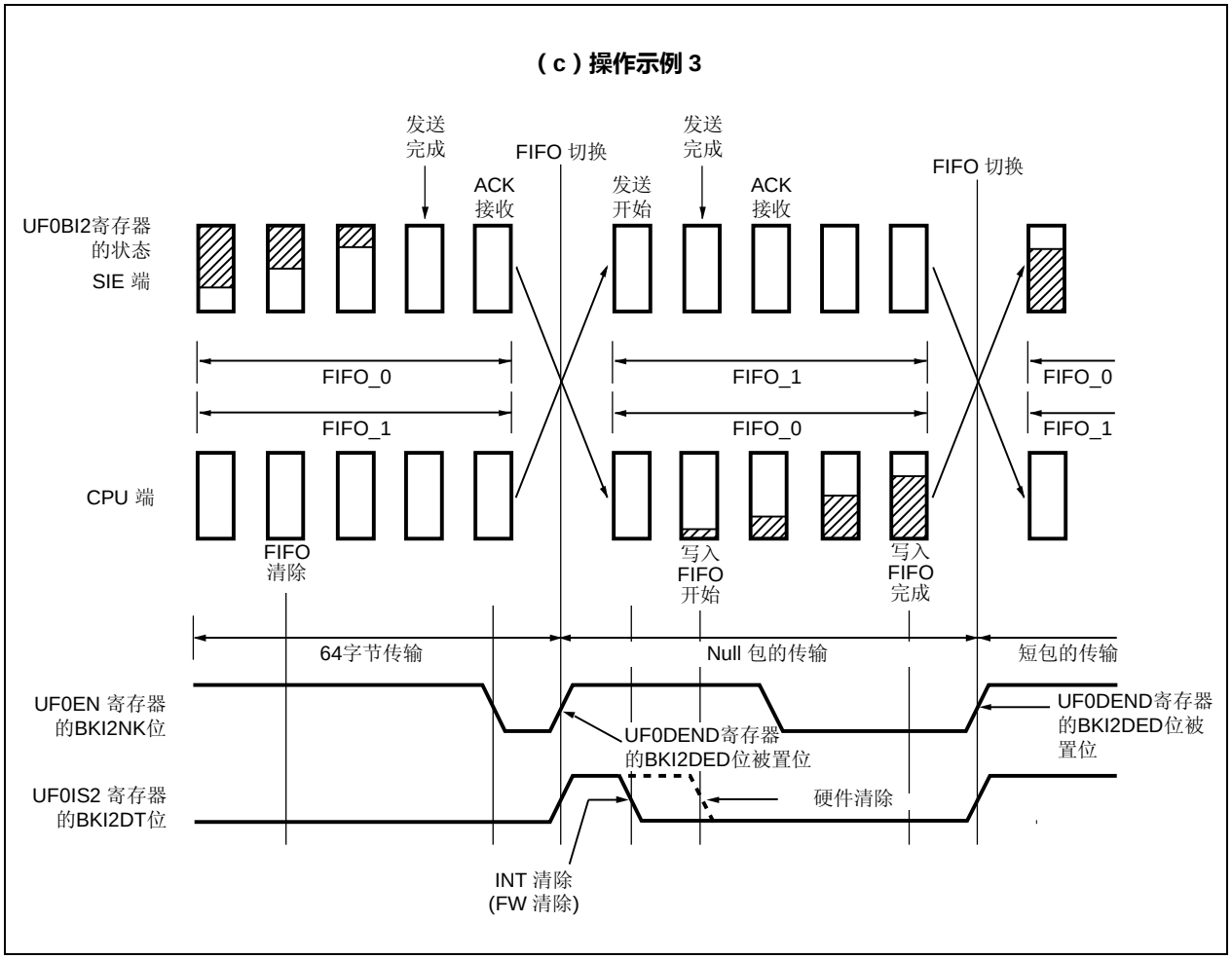


图 20-10. UF0BI2 寄存器的操作（3/3）



(11) UF0 中断 1 寄存器（UF0INT1）

UF0INT1 寄存器是一个 8-字节 FIFO，保存进出端点 7 的数据（传递到 SIE）。

该寄存器是只能写入，按字节进行写入。如果读取该寄存器，读到的值是 00H。

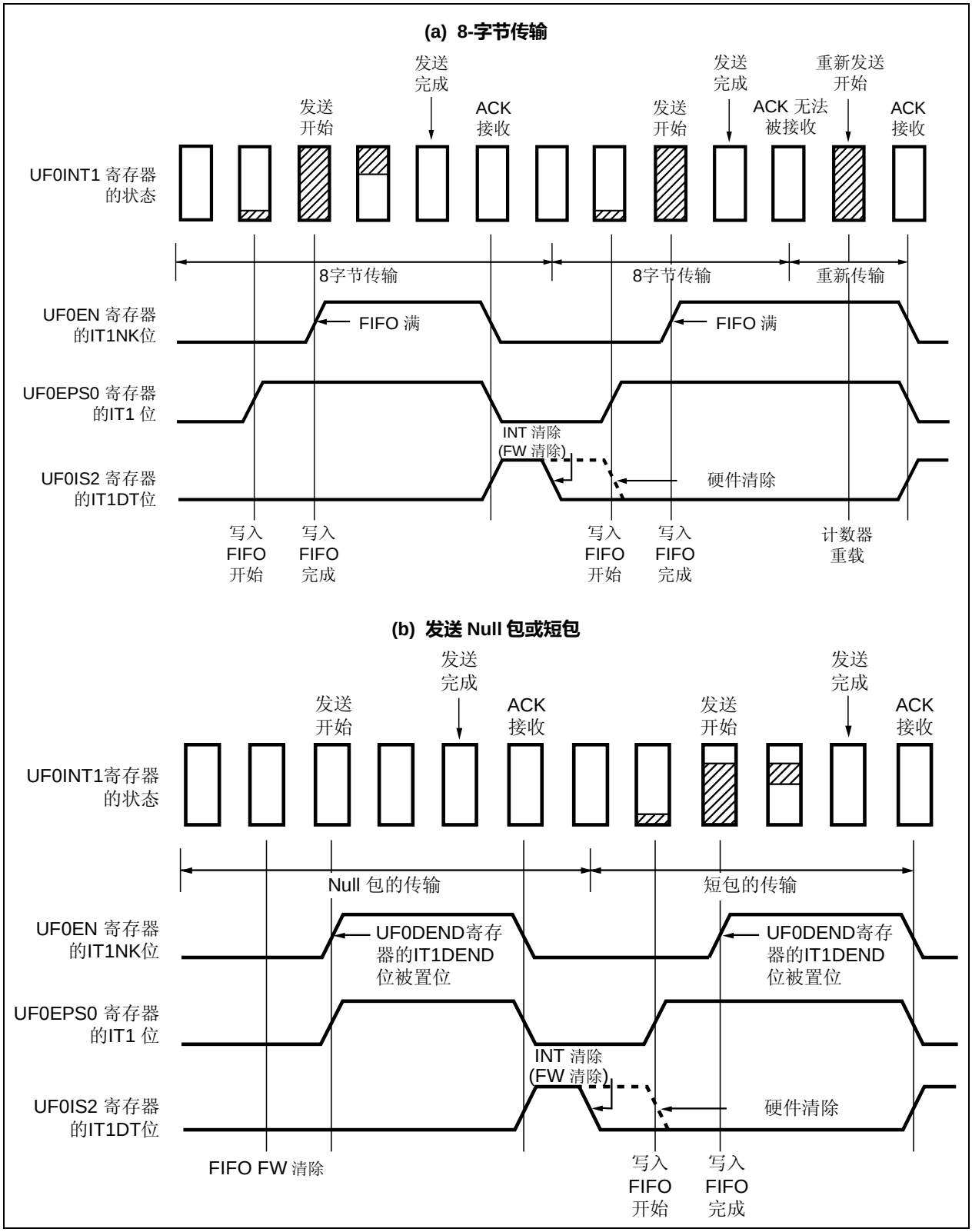
只有当 UF0E0N 寄存器的 IT1NK 位被置为 1 时（NAK 没有被发送时），硬件和对端点 7 的 IN 令牌同步，发送数据到 USB 总线。当数据已发送且主机正确接收该数据时，UF0EN 寄存器的 IT1NK 位由硬件自动清除为 0。当数据已写入 UF0INT1 寄存器且 UF0DEND 寄存器的 IT1DEND 位被置为 1（UF0EPS0 寄存器的 IT1 位 = 1 （数据存在））时，发送一个短包。当 UF0INT1 寄存器已清除且 UF0DEND 寄存器的 IT1DEND 位被置为 1（UF0EPS0 寄存器的 IT1 位 = 1 （数据存在））时，发送一个空包。

	7	6	5	4	3	2	1	0	地址	复位后
UF0INT1	IT17	IT16	IT15	IT14	IT13	IT12	IT11	IT10	00200114H	不确定

位的位置	位名称	功能
7 至 0	IT17 至 IT10	这些位保存进出端点 7 的数据。

UF0INT1 寄存器的操作如下所示。

图 20-11. UF0INT1 寄存器的操作



20.6.5 EPC请求数据寄存器

（1）UF0 设备状态寄存器 L（UF0DSTL）

该寄存器保存响应 GET_STATUS 设备请求的返回值。

该寄存器可以按字节进行读取或写入。

当接收到 GET_STATUS 设备请求时，硬件自动将该寄存器的内容发送给主机。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0DSTL	0	0	0	0	0	0	RMWK	SFPW	00200144H	00H

位的位置	位名称	功能
1	RMWK	该位指定是否使用设备的远程唤醒功能。 1：允许 0：禁止 如果设备支持远程唤醒功能，当接收到 SET_FEATURE 设备请求时，该位由硬件设置为 1，并且当接收到 CLEAR_FEATURE 设备 请求时，由硬件清除为 0。如果设备不支持远程唤醒功能，确保主机不会发出 SET_FEATURE 设备请求。
0	SFPW	该位表明设备是自供电还是总线供电。 1：自供电 0：总线供电

（2）UF0 EP0 状态寄存器 L（UF0E0SL）

该寄存器保存响应 GET_STATUS 端点 0 请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

如果 USBF 发生一个错误，E0HALT 位被 FW 设置为 1。当正在接收对端点 0 的 USB 端访问时，对该寄存器的写入访问被忽略。

如果之前的控制传输是对于 SET_FEATURE 端点 0、CLEAR_FEATURE 端点 0、GET_STATUS 端点 0 请求或由 FW 处理的请求，当 E0HALT 位被 FW 设置为 1 时，它不会立即被反映出来，直到接收到下一个 SETUP 令牌。

当接收到 GET_STATUS 端点 0 请求时，硬件自动将该寄存器的内容发送给主机。如果端点 0 已经停顿，UF0E0W 和 UF0E0R 寄存器被清除，且 UF0E0N 寄存器的 EP0NKW 和 EP0NKR 位被清除为 0。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E0SL	0	0	0	0	0	0	0	E0HALT	0020014CH	00H

位的位置	位名称	功能
0	E0HALT	该位表明端点 0 的状态。 1：已停顿（Stalled） 0：未停顿 当接收到 SET_FEATURE 端点 0 请求时，该位被硬件设置为 1，并且当接收到 CLEAR_FEATURE 端点 0 请求时，该位被硬件清除为 0。DATA PID 被初始化为 DATA0。

（3）UF0 EP1 状态寄存器 L（UF0E1SL）

该寄存器保存响应 GET_STATUS 端点 1 请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

如果端点 1 发生一个错误，E1HALT 位被 FW 设置为 1。当正在接收对端点 1 的 USB 端访问时，对该寄存器的写入访问被忽略。

当接收到 GET_STATUS 端点 1 请求时，硬件自动将该寄存器的内容发送给主机。如果端点 1 已经停顿，UF0BI1 寄存器被清除，且 BKI1NK 位被清除为 0。

当传输到端点 1 而不是控制传输时，被执行时写入该寄存器总是被屏蔽，确保检查该寄存器来查看数据是否被正确写入。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E1SL	0	0	0	0	0	0	0	E1HALT	00200150H	00H

位的位置	位名称	功能
0	E1HALT	该位表明端点 1 的状态。 1：已停顿（Stalled） 0：未停顿 当接收到 SET_FEATURE 端点 1 请求时，该位被硬件设置为 1，并且当正确接收到 CLEAR_FEATURE 端点 1 请求，SET_CONFIGURATION 请求，或正确接收到对端点 1 链接的接口的 SET_INTERFACE 请求时，该位被硬件清除为 0。DATA PID 被初始化为 DATA0。

（4）UF0 EP2 状态寄存器 L（UF0E2SL）

该寄存器保存响应 GET_STATUS 端点 2 请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

如果端点 2 发生一个错误，E2HALT 位被 FW 设置为 1。当正在接收对端点 2 的 USB 端访问时，对该寄存器的写入访问被忽略。

当接收到 GET_STATUS 端点 2 请求时，硬件自动将该寄存器的内容发送给主机。如果端点 2 已经停顿，UF0BO1 寄存器被清除，且 BKO1NK 位被清除为 0。

当传输到端点 2 而不是控制传输时，被执行时写入该寄存器总是被屏蔽，确保检查该寄存器来查看数据是否被正确写入。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E2SL	0	0	0	0	0	0	0	E2HALT	00200154H	00H

位的位置	位名称	功能
0	E2HALT	该位表明端点 2 的状态。 1：已停顿（Stalled） 0：未停顿 当接收到 SET_FEATURE 端点 2 请求时，该位被硬件设置为 1，并且当正确接收到 CLEAR_FEATURE 端点 2 请求，SET_CONFIGURATION 请求，或正确接收到对端点 2 链接的接口的 SET_INTERFACE 请求时，该位被硬件清除为 0。DATA PID 被初始化为 DATA0。

（5）UF0 EP3 状态寄存器 L（UF0E3SL）

该寄存器保存响应 GET_STATUS 端点 3 请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

如果端点 3 发生一个错误，E3HALT 位被 FW 设置为 1。当正在接收对端点 3 的 USB 端访问时，对该寄存器的写入访问被忽略。

当接收到 GET_STATUS 端点 3 请求时，硬件自动将该寄存器的内容发送给主机。如果端点 3 已经停顿，UF0BI2 寄存器被清除，且 BKI2NK 位被清除为 0。

当传输到端点 3 而不是控制传输时，被执行时写入该寄存器总是被屏蔽，确保检查该寄存器来查看数据是否被正确写入。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E3SL	0	0	0	0	0	0	0	E3HALT	00200158H	00H

位的位置	位名称	功能
0	E3HALT	该位表明端点 3 的状态。 1：已停顿（Stalled） 0：未停顿 当接收到 SET_FEATURE 端点 3 请求时，该位被硬件设置为 1，并且当正确接收到 CLEAR_FEATURE 端点 3 请求，SET_CONFIGURATION 请求，或正确接收到对端点 3 链接的接口的 SET_INTERFACE 请求时，该位被硬件清除为 0。DATA PID 被初始化为 DATA0。

（6）UF0 EP4 状态寄存器 L（UF0E4SL）

该寄存器保存响应 GET_STATUS 端点 4 请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

如果端点 4 发生一个错误，E4HALT 位被 FW 设置为 1。当正在接收对端点 4 的 USB 端访问时，对该寄存器的写入访问被忽略。

当接收到 GET_STATUS 端点 4 请求时，硬件自动将该寄存器的内容发送给主机。如果端点 4 已经停顿，UF0BO2 寄存器被清除，且 BKO2NK 位被清除为 0。

当传输到端点 4 而不是控制传输时，被执行时写入该寄存器总是被屏蔽，确保检查该寄存器来查看数据是否被正确写入。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E4SL	0	0	0	0	0	0	0	E4HALT	0020015CH	00H

位的位置	位名称	功能
0	E4HALT	该位表明端点 4 的状态。 1：已停顿（Stalled） 0：未停顿 当接收到 SET_FEATURE 端点 4 请求时，该位被硬件设置为 1，并且当正确接收到 CLEAR_FEATURE 端点 4 请求，SET_CONFIGURATION 请求，或正确接收到对端点 4 链接的接口的 SET_INTERFACE 请求时，该位被硬件清除为 0。DATA PID 被初始化为 DATA0。

（7）UF0 EP7 状态寄存器 L（UF0E7SL）

该寄存器保存响应 GET_STATUS 端点 7 请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

如果端点 7 发生一个错误，E7HALT 位被 FW 设置为 1。当正在接收对端点 7 的 USB 端访问时，对该寄存器的写入访问被忽略。

当接收到 GET_STATUS 端点 7 请求时，硬件自动将该寄存器的内容发送给主机。如果端点 7 已经停顿，UF0INT1 寄存器被清除，且 IT1NK 位被清除为 0。

当传输到端点 7 而不是控制传输时，被执行时写入该寄存器总是被屏蔽，确保检查该寄存器来查看数据是否被正确写入。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0E7SL	0	0	0	0	0	0	0	E7HALT	00200168H	00H

位的位置	位名称	功能
0	E7HALT	该位表明端点 7 的状态。 1：已停顿（Stalled） 0：未停顿 当接收到 SET_FEATURE 端点 7 请求时，该位被硬件设置为 1，并且当正确接收到 CLEAR_FEATURE 端点 7 请求，SET_CONFIGURATION 请求，或正确接收到对端点 7 链接的接口的 SET_INTERFACE 请求时，该位被硬件清除为 0。DATA PID 被初始化为 DATA0。

（8）UF0 地址寄存器（UF0ADRS）

该寄存器保存设备地址。

该寄存器是只读的，按字节进行读取。

由 SET_ADDRESS 请求发送的设备地址被分析，且作为结果的值被自动写入该寄存器。如果 SET_ADDRESS 请求由 FW 处理，在状态阶段中接收到 SUCCESS 信号时，该寄存器的值作为设备地址被反映出来。

注意事项 不要对该寄存器执行写入访问。如果该寄存器被写入，操作无法保证。

	7	6	5	4	3	2	1	0	地址	复位后
UF0ADRS	0	ADRS6	ADRS5	ADRS4	ADRS3	ADRS2	ADRS1	ADRS0	00200180H	00H

位的位置	位名称	功能
6 至 0	ADRS6 至 ADRS0	这些位保持 SIE 的设备地址。

（9）UF0 配置寄存器（UF0CNF）

该寄存器保存响应 GET_CONFIGURATION 请求的返回值。

该寄存器是只读的，按字节进行读取。

当接收到 SET_CONFIGURATION 请求时，wValue 被自动写入该寄存器。

当检测到该寄存器的值从 00H 更改为 00H 以外的其它值时，UF0MODS 寄存器的 CONF 位被设置为 1。如果 SET_CONFIGURATION 请求由 FW 处理，只要数据被写入该寄存器（状态阶段完成之前 CONF 位 = 1），该寄存器的状态立即反映到 UF0MODS 寄存器上。

注意事项 不要对该寄存器执行写入访问。如果该寄存器被写入，操作无法保证。

	7	6	5	4	3	2	1	0	地址	复位后
UF0CNF	0	0	0	0	0	0	CONF1	CONF0	00200182H	00H

位的位置	位名称	功能
1，0	CONF1，CONF0	这些位保存对 GET_CONFIGURATION 请求的响应的返回数据。

（10）UF0 接口 0 寄存器（UF0IF0）

该寄存器保存响应 GET_INTERFACE wIndex = 0 请求的返回值。

该寄存器是只读的，按字节进行读取。

当接收到 SET_INTERFACE 请求时，wValue 被自动写入该寄存器。

如果 SET_INTERFACE 请求由 FW 处理，wIndex 和 wValue 被译码，并且端点的设置被自动更改。这时根据设置，将目标端点的状态位和 DPID 自动清除为 0。FIFO 不被自动清除。

注意事项 不要对该寄存器执行写入访问。如果该寄存器被写入，操作无法保证。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IF0	0	0	0	0	0	IF02	IF01	IF00	00200184H	00H

位的位置	位名称	功能
2 至 0	IF02 至 IF00	这些位保存对 GET_INTERFACE wIndex = 0 请求的响应的返回值。

（11）UF0 接口 1 至 4 寄存器（UF0IF1 至 UF0IF4）

该寄存器保存响应 GET_INTERFACE wIndex = n 请求（n = 1 至 4）的返回值。

该寄存器是只读的，按字节进行读取。

当接收到 SET_INTERFACE 请求时，wValue 被自动写入该寄存器。

根据 UF0AIFN 和 UF0AAS 寄存器的设置，这些寄存器无效。

如果 SET_INTERFACE 请求由 FW 处理，wIndex 和 wValue 被译码，并且端点的设置被自动更改。这时根据设置，将目标端点的状态位和 DPID 自动清除为 0。FIFO 不被自动清除。

注意事项 不要对该寄存器执行写入访问。如果该寄存器被写入，操作无法保证。

	7	6	5	4	3	2	1	0	地址	复位后
UF0IF1	0	0	0	0	0	IF12	IF11	IF10	00200186H	00H

	7	6	5	4	3	2	1	0	地址	复位后
UF0IF2	0	0	0	0	0	IF22	IF21	IF20	00200188H	00H

	7	6	5	4	3	2	1	0	地址	复位后
UF0IF3	0	0	0	0	0	IF32	IF31	IF30	0020018AH	00H

	7	6	5	4	3	2	1	0	地址	复位后
UF0IF4	0	0	0	0	0	IF42	IF41	IF40	0020018CH	00H

位的位置	位名称	功能
2 至 0	IFn2 至 IFn0	这些位保存对 GET_INTERFACE wIndex = n 请求的响应的返回值。

备注 n = 1 至 4

（12）UF0 描述符长度寄存器（UF0DSCL）

该寄存器保存响应 GET_DESCRIPTOR 配置请求的返回值长度。该寄存器的值是 UF0CIEn 寄存器设置的所有描述符的字节数减 1（n = 0 至 255）。响应 GET_DESCRIPTOR 配置请求时返回的总描述符长度根据该寄存器的值决定。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。wLength 的处理被自动控制。如果该寄存器被设置为 00H，它意味着将要返回的描述符长度为 1 字节。如果寄存器被设置为 FFH，返回的描述符长度为 256 字节。当使用的描述符长度超过 256 字节时，设置 UF0MODC 寄存器的 CDCGDST 位为 1，且由 FW 处理 GET_DESCRIPTOR 请求（这时，UF0MODS 寄存器的 CDCGD 位也被置为 1）。

注意事项 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

	7	6	5	4	3	2	1	0	地址	复位后
UF0DSCL	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0	002001A0H	00H

位的位置	位名称	功能
7 至 0	DPL7 至 DPL0	这些位设置响应 GET_DESCRIPTOR 配置请求将要返回的所有描述符的字节数减 1。

（13）UF0 设备描述符寄存器 0 至 17（UF0DD0 至 UF0DD17）

这些寄存器保存响应 GET_DESCRIPTOR 设备请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

注意事项 1. 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

2. 使用 USB 2.0 规范定义的值和最新等级规范作为设置值。

	7	6	5	4	3	2	1	0	地址	复位后
UF0DDn (n=0至17)									参见**表20-5**	不确定

表 20-5. UF0 设备描述符寄存器的映射和数据

符号	地址	域名	内容
UF0DD0	002001A2H	bLength	描述符的大小
UF0DD1	002001A4H	bDescriptorType	设备描述符类型
UF0DD2	002001A6H	bcdUSB	USB 规范的版本号的小数点后面的值
UF0DD3	002001A8H		USB 规范的版本号的小数点前面的值
UF0DD4	002001AAH	bDeviceClass	等级码
UF0DD5	002001ACH	bDeviceSubClass	次等级码
UF0DD6	002001AEH	bDeviceProtocol	协议码
UF0DD7	002001B0H	bMaxPacketSize0	端点 0 的最大包大小
UF0DD8	002001B2H	idVendor	供应商 ID 的低位值
UF0DD9	002001B4H		供应商 ID 的高位值
UF0DD10	002001B6H	idProduct	产品 ID 的低位值
UF0DD11	002001B8H		产品 ID 的高位值
UF0DD12	002001BAH	bcdDevice	设备发布号的低位值
UF0DD13	002001BCH		设备发布号的高位值
UF0DD14	002001BEH	iManufacturer	描述制造商的字符串描述符的索引
UF0DD15	002001C0H	iProduct	描述产品的字符串描述符的索引
UF0DD16	002001C2H	lSerialNumber	描述设备序列号的字符串描述符的索引
UF0DD17	002001C4H	BNumConfigurations	可设置的配置数量

（14）UF0 配置 / 接口 / 端点描述符寄存器 0 至 255（UF0CIE0 至 UF0CIE255）

这些寄存器保存响应 GET_DESCRIPTOR 配置请求的返回值。

该寄存器可以按字节进行读取或写入。但是要注意，只有当 EP0NKA 位被置为 1 时，数据才能写入到该寄存器。

这些寄存器中可以保存最多 256 字节的描述符信息。按照配置、接口和端点的顺序保存各个描述符（参见 **表 20-6**）。如果有两个或更多接口，在接口描述符之后重复保存数据。

表 20-6. UF0CIEn 寄存器的映射

地址	保存的描述符
002001C6H	配置描述符（9 字节）
002001D8H	接口描述符（9 字节）
002001EAH	端点 1 描述符（7 字节）
002001F8H	端点 2 描述符（7 字节）
00200206H	端点 3 描述符（7 字节）
:	:
002002xxH	接口描述符（9 字节）
002002xxH+9	端点 1 描述符（7 字节）
002002xxH+16	端点 2 描述符（7 字节）
002002xxH+23	端点 3 描述符（7 字节）
:	:

可以设置到这些寄存器的有效数据的范围根据 UF0DSCL 寄存器的设置而改变。除了表 20-7 列出的描述符以外，也可以保存特有的等级和供应商描述符。

如果所有的值是固定的，它们可以被存储在 ROM 中。

注意事项 1. 要重新写入该寄存器，为了防止读访问和写访问之间的冲突，在读取该寄存器的内容之前设置 EP0NKA 位为 1，并且在确认该位已经置位后重写该寄存器的内容。

2. 使用 USB 规范 2.0 版本定义的值和最新等级规范作为设置值。

	7	6	5	4	3	2	1	0	地址	复位后
UF0CIEn (n=0至255)									002001C6H 至 002003C4H	不确定

表 20-7. UF0CIEn 寄存器的数据

（a）配置描述符（9 字节）

偏移量	域名	内容
0	bLength	该描述符的大小
1	bDescriptorType	描述符类型
2	wTotalLength	配置、所有接口和所有端点描述符的总字节数的低位
3		配置、所有接口和所有端点描述符的总字节数的高位
4	bNumInterface	接口个数
5	bConfigurationValue	选择该配置的值
6	iConfiguration	描述该配置的字符串描述符的索引
7	bmAttributes	该配置的特征（自供电、无远程唤醒）
8	MaxPower	该配置的最大电源消耗（单位：mA）[注]

注 该值以 2mA 为单位。（例如：50 = 100 mA）

（b）接口描述符（9 字节）

偏移量	域名	内容
0	bLength	该描述符的大小
1	bDescriptorType	描述符类型
2	bInterfaceNumber	该接口的值
3	bAlternateSetting	选择接口的复用设置的值
4	bNumEndpoints	可用端点的数量
5	bInterfaceClass	等级码
6	bInterfaceSubClass	次等级码
7	bInterfaceProtocol	协议码
8	Interface	描述该接口的字符串描述符的索引

（c）端点描述符（7 字节）

偏移量	域名	内容
0	bLength	该描述符的大小
1	bDescriptorType	描述符类型
2	bEndpointAddress	该端点的地址 / 传输方向
3	bmAttributes	传输类型
4	wMaxPaketSize	传输数据的最大数的低位
5		传输数据的最大数的高位
6	bInterval	传输间隔

20.6.6 桥寄存器

(1) 桥中断控制寄存器（BRGINTT）

BRGINTT 寄存器用于控制中断产生状态的 DMA 传输状态，以及从 EPC 到桥电路的各个端点（EP1 至 EP4）。BRGINTT 寄存器可以按 16 位宽度进行读取或写入。

复位后：0000H　R/W　地址：00200400H

	15	14	13	12	11	10	9	8
BRGINTT	0	0	0	0	EP4INT	EP3INT	EP2INT	EP1INT

7	6	5	4	3	2	1	0
0	0	0	0	0	EPCINT2B	EPCINT1B	EPCINT0B

位的位置	位名称	功能
11	EP4INT	在 EP4 中，当 DMA 传输正常结束，或者在 DMA 传输中错误完成，该位被置位。通过写入“1”可以将该位清除位“0”。 0：DMA 传输尚未完成 1：DMA 传输完成
10	EP3INT	在 EP3 中，当 DMA 传输正常结束，或者在 DMA 传输中错误完成，该位被置位。通过写入“1”可以将该位清除位“0”。 0：DMA 传输尚未完成 1：DMA 传输完成
9	EP2INT	在 EP2 中，当 DMA 传输正常结束，或者在 DMA 传输中错误完成，该位被置位。通过写入“1”可以将该位清除位“0”。 0：DMA 传输尚未完成 1：DMA 传输完成
8	EP1INT	在 EP1 中，当 DMA 传输正常结束，或者在 DMA 传输中错误完成，该位被置位。通过写入“1”可以将该位清除位“0”。 0：DMA 传输尚未完成 1：DMA 传输完成
2	EPCINT2B	表示来自 EPC 的中断请求信号 “EPC_INT2B”的状态。 清除来自 EPC 寄存器的请求控制。 0：中断未发出 1：中断已发出
1	EPCINT1B	表示来自 EPC 的中断请求信号 “EPC_INT1B”的状态。 清除来自 EPC 寄存器的请求控制。 0：中断未发出 1：中断已发出
0	EPCINT0B	表示来自 EPC 的中断请求信号 “EPC_INT0B”的状态。 清除来自 EPC 寄存器的请求控制。 0：中断未发出 1：中断已发出

(2) 桥中断使能寄存器（BRGINTE）

BRGINTE 寄存器用于控制使能或禁止桥电路中的中断发生。

BRGINTE 寄存器可以按 16 位宽度进行读取或写入。

复位后：0000H R/W 地址：00200402H

	15	14	13	12	11	10	9	8
BRGINTE	0	0	0	0	EP4INTN	EP3INTN	EP2INTN	EP1INTN

7	6	5	4	3	2	1	0
0	0	0	0	0	EPC INT2BEN	EPC INT1BEN	EPC INT0BEN

位的位置	位名称	功能
11	EP4INTN	当 EP4INT 被置位时，使能或禁止发生中断的设置 0：禁止中断 1：使能中断
10	EP3INTN	当 EP3INT 被置位时，使能或禁止发生中断的设置 0：禁止中断 1：使能中断
9	EP2INTN	当 EP2INT 被置位时，使能或禁止发生中断的设置 0：禁止中断 1：使能中断
8	EP1INTN	当 EP1INT 被置位时，使能或禁止发生中断的设置 0：禁止中断 1：使能中断
2	EPCINT2BEN	当 EPCINT2BEN 位被置位时，使能或禁止发生中断的设置 0：禁止中断 1：使能中断
1	EPCINT1BEN	当 EPCINT1BEN 位被置位时，使能或禁止发生中断的设置 0：禁止中断 1：使能中断
0	EPCINT0BEN	当 EPCINT0BEN 位被置位时，使能或禁止发生中断的设置 0：禁止中断 1：使能中断

(3) EPC 模块控制寄存器（EPCCLT）

EPCCLT 寄存器用于控制到 EPC 模块的复位发生器。

EPCCLT 寄存器可以按 16 位宽度进行读取或写入。

复位后：0000H R/W 地址：00200404H

	15	14	13	12	11	10	9	8
EPCCLT	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	EPCRST

位的位置	位名称	功能
0	EPCRST	设置 EPC 发生的复位。 0：复位解除 1：复位发出

(4) CPU I/F 总线控制寄存器（CPUBCTL）

CPUBCTL 寄存器用于控制桥电路和 CPU 之间的接口。

CPUBCTL 寄存器可以按 16 位宽度进行读取或写入。

复位后：不确定　R/W　地址：00200408H

	15	14	13	12	11	10	9	8
CPUBCTL	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	BULKWAIT	DATAWAIT	NOWAIT

位的位置	位名称	功能
2	BULKWAIT	当访问批量寄存器时，强行插入 1 等待（批量等待）。 0：未强行插入批量等待注（默认值） 1：强行插入批量等待 **注** 在写入访问中该设置无效，批量等待被强行插入。
1	DATAWAIT	在 CPU 总线周期后，强行插入 1 等待（数据等待）。 0：未强行插入数据等待（默认值） 1：强行插入数据等待
0	NOWAIT	设置使能/禁止 CPU 总线周期操作的无等待。 0：禁止无等待注（默认值） 1：使能无等待 **注** 插入 1 等待或更多等待

20.6.7 DMA 寄存器

(1) EPn DMA 控制寄存器 1 （UF0E1DC1 至 UF0E4DC1）

UF0E1DC1 至 UF0E4DC1 寄存器用于控制端点 n (EPn)的 DMA 传输。(n = 1 至 4)

UF0E1DC1 至 UF0E4DC1 寄存器可以按 16 位宽度进行读取或写入。

(1/2)

复位后：0000H R/W 地址：00200500H

	15	14	13	12	11	10	9	8
UF0E1DC1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	EP1BULK2	EP1BULK1	EP1BULK0	EP1STOP	EP1REQ	EP1DMAEN

复位后：0000H R/W 地址：00200504H

	15	14	13	12	11	10	9	8
UF0E2DC1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	EP2BULK2	EP2BULK1	EP2BULK0	EP2STOP	EP2REQ	EP2DMAEN

复位后：0000H R/W 地址：00200508H

	15	14	13	12	11	10	9	8
UF0E3DC1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	EP3BULK2	EP3BULK1	EP3BULK0	EP3STOP	EP3REQ	EP3DMAEN

复位后：0000H R/W 地址：0020050CH

	15	14	13	12	11	10	9	8
UF0E4DC1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	EP4BULK2	EP4BULK1	EP4BULK0	EP4STOP	EP4REQ	EP4DMAEN

(2/2)

位的位置	位名称	功能
5 至 3	EPnBULK2， EPnBULK1， EPnBULK0	表示内部桥的批量传输时状态机“BIN_STATE”的状态 EPnBULK2 EPnBULK1 EPnBULK0 ”BIN_STATE” 状态 0 0 0 BIN_IDLE 0 0 1 BIN_CPU 0 1 0 BIN_EPC 0 1 1 BIN_CMP 1 0 0 BIN_END
2	EPnSTOP	表示来自 EPC 的 DMA 传输结束状态（DMA 传输的结束因素） 0：通过 EPn_TCNT 值为“0”结束 DMA 传输 1：通过对“EPC_DMARQ_EPnB”求反结束 DMA 传输 设置下一个 EP1_DMAEN 为“1”，则自动清除(0)。
1	EPnREQ	表示来自 EPC 的 “EPC_DMARQ_EPnB”信号的状态 0：没有 DMA 请求信号 1：DMA 请求信号
0	EPnDMAEN	设置对来自 EPC 的 DMA 请求的控制 0：屏蔽 DMA 请求 1：使能 DMA 请求 EPn_TCNT 设置的包的数量传输完成时，或完成对 DMARQ_EpnB 求反数量的 DMA 传输时，自动清除(0)。 **注意事项 在强行结束中该设置的值无法保证。**

备注 n = 1 至 4

(2) EPn DMA 控制寄存器 2 (UF0E1DC2 至 UF0E4DC2)

UF0E1DC2 至 UF0E4DC2 寄存器用于控制端点 n (EPn)的 DMA 传输。(n = 1 至 4)

UF0E1DC2 至 UF0E4DC2 寄存器可以按 16 位宽度进行读取或写入。

(1/2)

复位后：0000H R/W 地址：00200502H

	15	14	13	12	11	10	9	8
UF0E1DC2	EP1 TCNT15	EP1 TCNT14	EP1 TCNT13	EP1 TCNT12	EP1 TCNT11	EP1 TCNT10	EP1 TCNT9	EP1 TCNT8

7	6	5	4	3	2	1	0
EP1 TCNT7	EP1 TCNT6	EP1 TCNT5	EP1 TCNT4	EP1 TCNT3	EP1 TCNT2	EP1 TCNT1	EP1 TCNT0

复位后：0000H R/W 地址：00200506H

	15	14	13	12	11	10	9	8
UF0E2DC2	EP2 TCNT15	EP2 TCNT14	EP2 TCNT13	EP2 TCNT12	EP2 TCNT11	EP2 TCNT10	EP2 TCNT9	EP2 TCNT8

7	6	5	4	3	2	1	0
EP2 TCNT7	EP2 TCNT6	EP2 TCNT5	EP2 TCNT4	EP2 TCNT3	EP2 TCNT2	EP2 TCNT1	EP2 TCNT0

复位后：0000H R/W 地址：0020050AH

	15	14	13	12	11	10	9	8
UF0E3DC2	EP3 TCNT15	EP3 TCNT14	EP3 TCNT13	EP3 TCNT12	EP3 TCNT11	EP3 TCNT10	EP3 TCNT9	EP3 TCNT8

7	6	5	4	3	2	1	0
EP3 TCNT7	EP3 TCNT6	EP3 TCNT5	EP3 TCNT4	EP3 TCNT3	EP3 TCNT2	EP3 TCNT1	EP3 TCNT0

复位后：0000H R/W 地址：0020050EH

	15	14	13	12	11	10	9	8
UF0E4DC2	EP4 TCNT15	EP4 TCNT14	EP4 TCNT13	EP4 TCNT12	EP4 TCNT11	EP4 TCNT10	EP4 TCNT9	EP4 TCNT8

7	6	5	4	3	2	1	0
EP4 TCNT7	EP4 TCNT6	EP4 TCNT5	EP4 TCNT4	EP4 TCNT3	EP4 TCNT2	EP4 TCNT1	EP4 TCNT0

(2/2)

位的位置	位名称	功能
15 至 0	EPnTCNT15 至 EPnTCNT0	设置 EPn 中 DMA 传输的字节数量。 在每次传输中，EPn_TCNT 的值递减到“0”，则结束 DMA 传输。 **注意事项 1. 当 EPn_DMAEN = 0 时，设置该寄存器。** **2. 禁止将该寄存器设置为“0” 。** **请确保为 DMA 传输计数寄存器 DBC0 至 DBC3 设置的值就是该寄存器的值 +1。** **3. 该寄存器的设置值可以为桥内的批量传输反映计数器 BIN_TCNT 的值。** **当 BIN_TCNT 的值为“0”，EPn_TCN 也为 “0”。** **4. 当强行中止时，停止为批量传输更新计数器 BIN_TCNT 的值。**

备注 n = 1 至 4

20.6.8 批量-in 寄存器

(1) UF0 EP1 批量-in 传输数据寄存器（UF0EP1BI）

UF0EP1BI 寄存器用于写入 EP1 的批量-in 传输数据。

UF0EP1BI 寄存器可以按字节或按 16 位宽度进行读取或写入。

复位后：0000H R/W 地址：00201000H

	15	14	13	12	11	10	9	8
UF0EP1BI	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
EP1BI7	EP1BI6	EP1BI5	EP1BI4	EP1BI3	EP1BI2	EP1BI1	EP1BI0

位的位置	位名称	功能
7 至 0	EP1BI7 至 EP1BI0	写入 EP1 的批量-in 传输数据。 将数据写入该寄存器，数据就会输出到 EPC 模块。 如果使用该寄存器，将地址 (00201000H) 设置为 DMAC 的 DMA 目的地址寄存器 (DDAn (n = 0 至 3))。而且，将 UFDRQEN 寄存器的 RQnUR1E (n = 0 至 3)位设置为 1，以指定一个 DMA 通道。

(2) UF0 EP3 批量-in 传输数据寄存器（UF0EP3BI）

UF0EP3BI 寄存器用于写入 EP3 的批量-in 传输数据。

UF0EP3BI 寄存器可以按字节或按 16 位宽度进行读取或写入。

复位后：0000H R/W 地址：00202000H

	15	14	13	12	11	10	9	8
UF0EP3BI	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
EP3BI7	EP3BI6	EP3BI5	EP3BI4	EP3BI3	EP3BI2	EP3BI1	EP3BI0

位的位置	位名称	功能
7 至 0	EP3BI7 至 EP3BI0	写入 EP3 的批量-in 传输数据。 将数据写入该寄存器，数据就会输出到 EPC 模块。 如果使用该寄存器，将地址 (00202000H) 设置为 DMAC 的 DMA 目的地址寄存器 (DDAn (n = 0 至 3))。而且，将 UFDRQEN 寄存器的 RQnUR3E (n = 0 至 3)位设置为 1，以指定一个 DMA 通道。

20.6.9 批量-out寄存器

(1) UF0 EP2 批量-out 传输数据寄存器（UF0EP2BO）

UF0EP2BO 寄存器用于读取 EP2 的批量-out 传输数据。

UF0EP2BO 寄存器可以按字节或按 16 位宽度进行读取。

复位后：0000H R 地址：00210000H

	15	14	13	12	11	10	9	8
UF0EP2BO	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
EP2BO7	EP2BO6	EP2BO5	EP2BO4	EP2BO3	EP2BO2	EP2BO1	EP2BO0

位的位置	位名称	功能
7 至 0	EP2BO7 至 EP2BO0	读取 EP2 的批量-out 传输数据。 从该寄存器读取来自 EPC 模块的输入数据。 如果使用该寄存器，将地址 (00210000H) 设置为 DMAC 的 DMA 源地址寄存器 (DSAn (n = 0 至 3))。而且，将 UFDRQEN 寄存器的 RQnUR0E (n = 0 至 3)位设置为 1，以指定一个 DMA 通道。

注意事项 如果执行下列任一操作，存储在该寄存器中的数据被读出，且下一次批量-out 传输数据被设置到该寄存器。

- **在程序执行期间读取 UF0EP2BO 寄存器。**
- **正在使用调试器时，在存储器观察窗口监控 UF0EP2BO 寄存器。**

(2) UF0 EP4 批量-out 传输数据寄存器（UF0EP4BO）

UF0EP4BO 寄存器用于读取 EP4 的批量-out 传输数据。

UF0EP4BO 寄存器可以按字节或按 16 位宽度进行读取。

复位后：0000H R 地址：00220000H

	15	14	13	12	11	10	9	8
UF0EP4BO	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
EP4BO7	EP4BO6	EP4BO5	EP4BO4	EP4BO3	EP4BO2	EP4BO1	EP4BO0

位的位置	位名称	功能
7 至 0	EP4BO7 至 EP4BO0	读取 EP4 的批量-out 传输数据。 从该寄存器读取来自 EPC 模块的输入数据。 如果使用该寄存器，将地址 (00220000H) 设置为 DMAC 的 DMA 源地址寄存器 (DSAn (n = 0 至 3))。而且，将 UFDRQEN 寄存器的 RQnUR2E (n = 0 至 3)位设置为 1，以指定一个 DMA 通道。

注意事项 如果执行下列任一操作，存储在该寄存器中的数据被读出，且下一次批量-out 传输数据被设置到该寄存器。

- **在程序执行期间读取 UF0EP4BO 寄存器。**
- **正在使用调试器时，在存储器观察窗口监控 UF0EP4BO 寄存器。**

20.6.10 外设控制寄存器

(1) USBF DMA 请求使能寄存器（UFDRQEN）

UFDRQEN 寄存器用于指定待使用的 DMA 通道和要进行传输的端点。

UFDRQEN 寄存器可以按 8 位或 16 位宽度进行读取或写入。

(1/2)

复位后：0000H R/W 地址：00240000H

	15	14	13	12	11	10	9	8
UFDRQEN	RQ3UR3E	RQ2UR3E	RQ1UR3E	RQ0UR3E	RQ3UR2E	RQ2UR2E	RQ1UR2E	RQ0UR2E

7	6	5	4	3	2	1	0
RQ3UR1E	RQ2UR1E	RQ1UR1E	RQ0UR1E	RQ3UR0E	RQ2UR0E	RQ1UR0E	RQ0UR0E

位的位置	位名称	功能
15，11，7，3	RQ3UR3E, RQ3UR2E, RQ3UR1E, RQ3UR0E	指定通过 DMA 通道 3 进行传输的端点 n (EPn)。 (n = 1 至 4) RQ3UR3E　RQ3UR2E　RQ3UR1E　RQ3UR0E　通过DMA3传输的EP 1　0　0　0　EP4 0　1　0　0　EP3 0　0　1　0　EP2 0　0　0　1　EP1 其它　DMA3不传输EPn (不使用DMA3)
14，10，6，2	RQ2UR3E, RQ2UR2E, RQ2UR1E, RQ2UR0E	指定通过 DMA 通道 2 进行传输的端点 n (EPn)。 (n = 1 至 4) RQ2UR3E　RQ2UR2E　RQ2UR1E　RQ2UR0E　通过DMA2传输的EP 1　0　0　0　EP4 0　1　0　0　EP3 0　0　1　0　EP2 0　0　0　1　EP1 其它　DMA2不传输EPn (不使用DMA2)

(2/2)

位的位置	位名称	功能
13，9，5，1	RQ1UR3E, RQ1UR2E, RQ1UR1E, RQ1UR0E	指定通过 DMA 通道 1 进行传输的端点 n (EPn)。 (n = 1 至 4) RQ1UR3E　RQ1UR2E　RQ1UR1E　RQ1UR0E　通过DMA1传输的EP 1　0　0　0　EP4 0　1　0　0　EP3 0　0　1　0　EP2 0　0　0　1　EP1 其它　DMA1不传输EPn (不使用DMA1)
12，8，4，0	RQ0UR3E, RQ0UR2E, RQ0UR1E, RQ0UR0E	指定通过 DMA 通道 0 进行传输的端点 n (EPn)。 (n = 1 至 4) RQ0UR3E　RQ0UR2E　RQ0UR1E　RQ0UR0E　通过DMA0传输的EP 1　0　0　0　EP4 0　1　0　0　EP3 0　0　1　0　EP2 0　0　0　1　EP1 其它　DMA0不传输EPn (不使用DMA0)

注意事项 1. 禁止对多个 DMA 通道设置同样的 DMA 传输目标，也禁止对同一个 DMA 通道设置多个 DMA 传输目标。

2. 如果使用该寄存器，设置 DMA 触发因素寄存器(DTFRn (n = 0 至 3))为通过中断(00H)禁止 DMA 请求。

下面的流程图展示了主机断开然后重新连接的程序执行，以及上电时的程序执行情况。

图 20-12. 主机断开然后重新连接的程序流程图

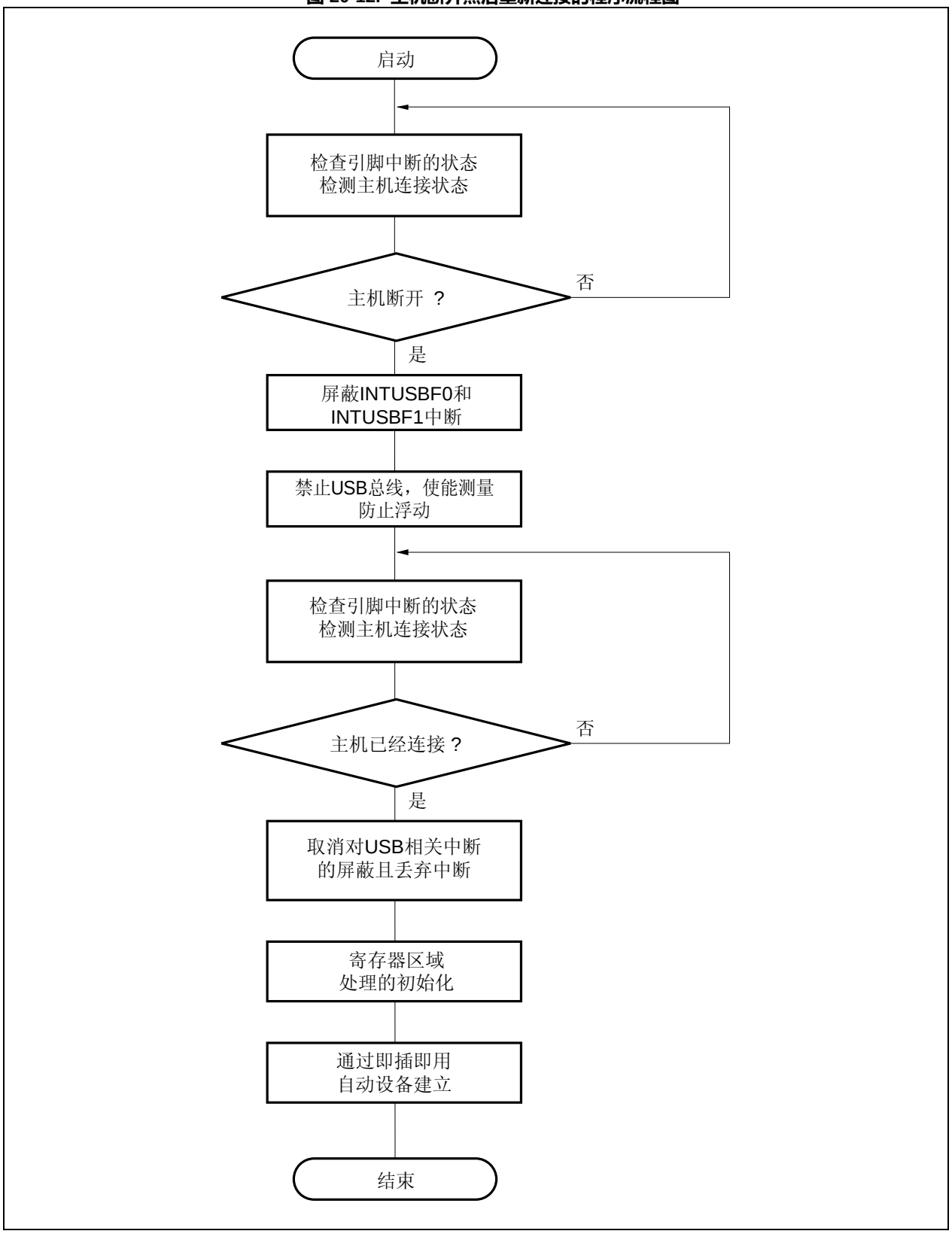
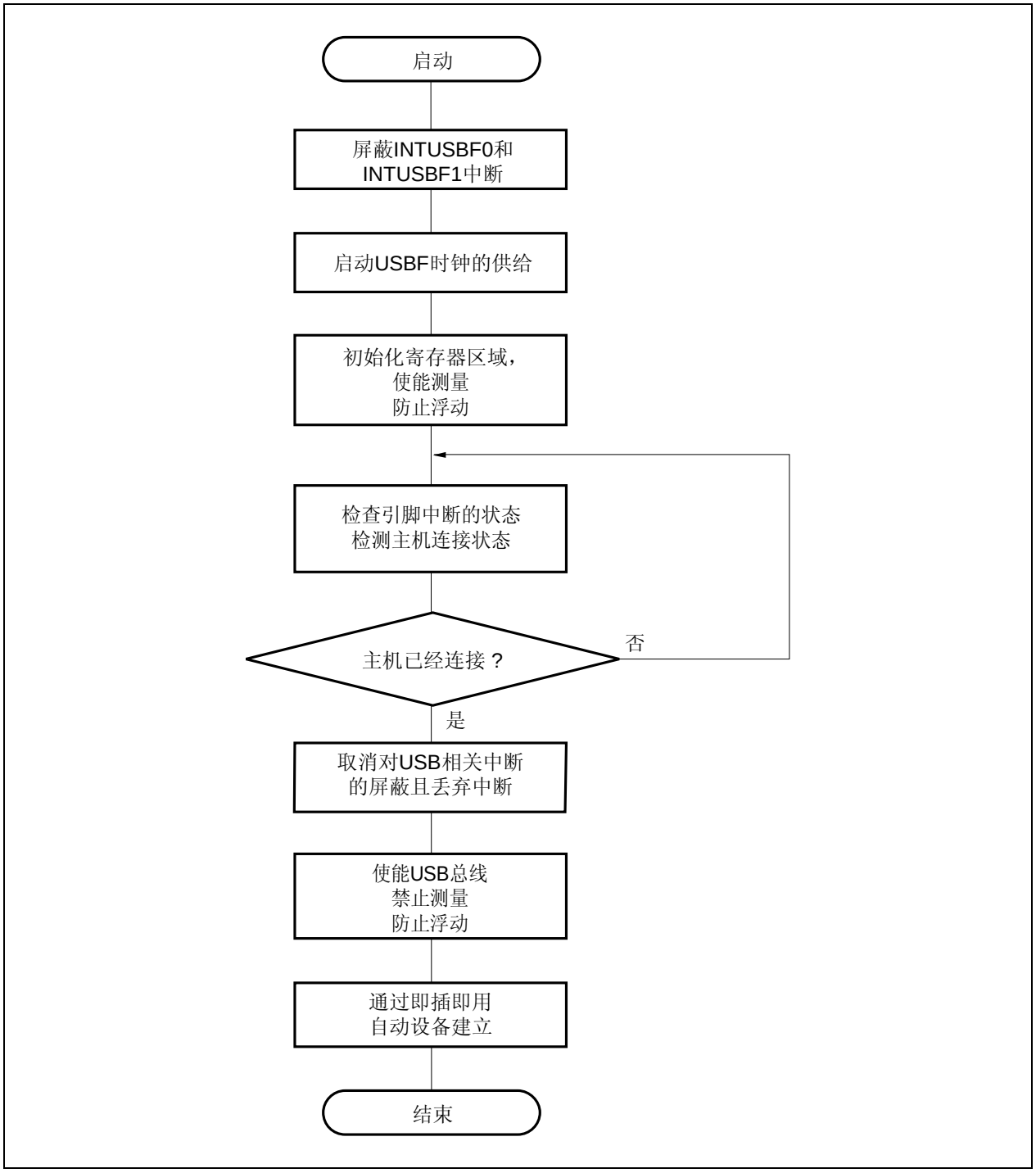


图 20-13. 上电时的程序流程图



20.7 STALL 握手或无握手

对 USBF 错误处理的定义如下。

传输类型	事务	目标包	错误类型	功能响应	处理
控制传输/ 批量传输/ 中断传输	IN/OUT/SETUP	令牌	端点不支持	无响应	无
			端点传输方向不匹配	无响应	无
			CRC 错误	无响应	无
			位填充错误	无响应	无
控制传输/ 批量传输	OUT/SETUP	数据	超时	无响应	无
			PID 检查错误	无响应	无
			不支持的 PID（Data PID 除外）	无响应	无
			CRC 错误	无响应	丢弃接收到的数据
			位填充错误	无响应	丢弃接收到的数据
	OUT	数据	数据 PID 不匹配	ACK	丢弃接收到的数据
控制传输（SETUP 阶段）	SETUP	数据	溢出	无响应	丢弃接收到的数据
控制传输 （数据阶段）	OUT	数据	溢出	无响应[注 1]	设置 UF0SDS 寄存器的 SNDSTL 位为 1，并且丢弃接收到的数据
控制传输 （状态阶段）	OUT	数据	溢出	ACK 或 无响应[注 2]	设置 UF0SDS 寄存器的 SNDSTL 位为 1 并且丢弃接收到的数据
批量传输	OUT	数据	溢出	无响应[注 1]	设置 UF0EnSL 寄存器的 EnHALT 位（n = 0 至 4，7） 为 1
控制传输/ 批量传输/ 中断传输	IN	握手	PID 检查错误	–	保持传输的数据并重新传输数据[注 3]
			不支持的 PID （ACK PID 除外）	–	保持传输的数据并重新传输数据[注 3]
			超时	–	保持传输的数据并重新传输数据[注 3]

注 1. 主机产生一个 STALL 响应来重新传输。
2. 如果传输数据小于最大包长度（MaxPacketSize），产生一个 ACK 响应并且丢弃状态阶段接收到的数据。如果超过最大包长度，不产生响应，UF0SDS 寄存器的 SNDSTL 位被设置为 1，并且丢弃接收到的数据。
3. 如果在控制传输期间，接收到一个 OUT 事务，表明从数据阶段更改到状态阶段，错误不被处理且认为接收已经正确完成。

注意事项 1. 通过当前设置的复用设置号码来判断目标端点是有效还是无效。
2. 关于包含在控制传输中的到/从端点 0 的响应，见 20.5 请求。

20.8 特定状态下寄存器的值

表 20-8. 特定状态下寄存器的值（1/2）

寄存器名	CPU 复位后（$\overline{\text{RESET}}$）	总线复位后
UF0E0N 寄存器	00H	值被保持
UF0E0NA 寄存器	00H	值被保持
UF0EN 寄存器	00H	值被保持
UF0ENM 寄存器	00H	值被保持
UF0SDS 寄存器	00H	值被保持
UF0CLR 寄存器	00H	值被保持
UF0SET 寄存器	00H	值被保持
UF0EPS0 寄存器	00H	值被保持
UF0EPS1 寄存器	00H	值被保持
UF0EPS2 寄存器	00H	值被保持
UF0IS0 寄存器	00H	值被保持
UF0IS1 寄存器	00H	值被保持
UF0IS2 寄存器	00H	值被保持
UF0IS3 寄存器	00H	值被保持
UF0IS4 寄存器	00H	值被保持
UF0IM0 寄存器	00H	值被保持
UF0IM1 寄存器	00H	值被保持
UF0IM2 寄存器	00H	值被保持
UF0IM3 寄存器	00H	值被保持
UF0IM4 寄存器	00H	值被保持
UF0IC0 寄存器	FFH	值被保持
UF0IC1 寄存器	FFH	值被保持
UF0IC2 寄存器	FFH	值被保持
UF0IC3 寄存器	FFH	值被保持
UF0IC4 寄存器	FFH	值被保持
UF0IDR 寄存器	00H	值被保持
UF0DMS0 寄存器	00H	值被保持
UF0DMS1 寄存器	00H	值被保持
UF0FIC0 寄存器	00H	值被保持
UF0FIC1 寄存器	00H	值被保持
UF0DEND 寄存器	00H	值被保持
UF0GPR 寄存器	00H	值被保持
UF0MODC 寄存器	00H	值被保持
UF0MODS 寄存器	00H	第 2 位（CONF）：清除（0）， 其它位： 值被保持
UF0AIFN 寄存器	00H	值被保持
UF0AAS 寄存器	00H	值被保持
UF0ASS 寄存器	00H	00H
UF0E1IM 寄存器	00H	值被保持
UF0E2IM 寄存器	00H	值被保持

表 20-8. 特定状态下寄存器的值（2/2）

寄存器名	CPU 复位后（$\overline{\text{RESET}}$）	总线复位后
UF0E3IM 寄存器	00H	值被保持
UF0E4IM 寄存器	00H	值被保持
UF0E7IM 寄存器	00H	值被保持
UF0E0R 寄存器	不确定[注 1]	值被保持
UF0E0L 寄存器	00H	值被保持
UF0E0ST 寄存器	00H	00H
UF0E0W 寄存器	不确定[注 1]	值被保持
UF0BO1 寄存器	不确定[注 1]	值被保持
UF0BO1L 寄存器	00H	值被保持
UF0BO2 寄存器	不确定[注 1]	值被保持
UF0BO2L 寄存器	00H	值被保持
UF0BI1 寄存器	不确定[注 1]	值被保持
UF0BI2 寄存器	不确定[注 1]	值被保持
UF0INT1 寄存器	不确定	值被保持
UF0DSTL 寄存器	00H	00H
UF0E0SL 寄存器	00H	00H
UF0E1SL 寄存器	00H	00H
UF0E2SL 寄存器	00H	00H
UF0E3SL 寄存器	00H	00H
UF0E4SL 寄存器	00H	00H
UF0E7SL 寄存器	00H	00H
UF0ADRS 寄存器	00H	00H
UF0CNF 寄存器	00H	00H
UF0IF0 寄存器	00H	00H
UF0IF1 寄存器	00H	00H
UF0IF2 寄存器	00H	00H
UF0IF3 寄存器	00H	00H
UF0IF4 寄存器	00H	00H
UF0DSCL 寄存器	00H	值被保持
UF0DDn 寄存器 (n = 0 至 17)	**注 2**	**注 2**
UF0CIEn 寄存器 (n = 0 至 255)	**注 2**	**注 2**

注 **1.** 该寄存器可以由 RESET 信号清除为 0，因为它的写指针、计数器和读取指针在 RESET 信号变为有效时被清除为 0，与被 UF0FICn 寄存器清除相同，因为该寄存器由 FIFO 控制。

2. 该寄存器不能被清除为 0。因为可以通过 FW 写入数据，但是，任意值都可以被写入该寄存器（这样做之前，请确保设置 UF0E0NA 寄存器的 EP0NKA 位为 1）。

20.9 FW 处理

执行以下 FW 处理。

- 枚举处理期间，设备端对 SET_CONFIGURATION、SET_INTERFACE、SET_FEATURE 和 CLEAR_FEATURE 请求的设置处理
- XXXXStandard，XXXXClass 和 XXXXVendor 请求的分析和处理不需要自动处理
- 批量传输的 OUT 令牌后，从接收缓冲器读取数据
- 写入响应批量传输 IN 令牌将要返回的数据
- 写入响应中断传输令牌将要返回的数据

下表列出了 FW 支持的请求。

表 20-9. FW 支持的标准请求

请求	接收端	处理 / 频率	说明
CLEAR_FEATURE	接口	自动 STALL 响应	因为没有功能选择器的值，尽管它为 bmRequestType 保留，被认为该请求没有到达接口。 当接收到该请求时，硬件执行自动 STALL 响应。
SET_FEATURE	接口	自动 STALL 响应	因为没有功能选择器的值，尽管它为 bmRequestType 保留，被认为该请求没有到达接口。 当接收到该请求时，硬件执行自动 STALL 响应。
GET_DESCRIPTOR	字符串	FW	返回字符串描述符。 当该请求由 SETUP 令牌接收时，硬件为 FW 产生 CPUDEC 中断请求。FW 从 CPUDEC 中断请求对请求的内容进行译码，并且将要返回给主机的数据写入到 UF0E0W 寄存器。
SET_DESCRIPTOR	设备	FW	重写设备描述符。 当该请求由 SETUP 令牌接收时，硬件为 FW 产生 CPUDEC 中断请求。FW 从 CPUDEC 中断请求对请求的内容进行译码，并且为下一个控制传输（OUT）将数据写入 UF0DDn 寄存器（n = 0 至 17）。
SET_DESCRIPTOR	配置	FW	重写配置描述符。 当该请求由 SETUP 令牌接收时，硬件为 FW 产生 CPUDEC 中断请求。FW 从 CPUDEC 中断请求对请求的内容进行译码，并且为下一个控制传输（OUT）将数据写入 UF0CIEn 寄存器（n = 0 至 255）。
SET_DESCRIPTOR	字符串	FW	重写字符串描述符。 当该请求由 SETUP 令牌接收时，硬件为 FW 产生 CPUDEC 中断请求。FW 从 CPUDEC 中断请求对请求的内容进行译码，并且为下一个控制传输（OUT）加载数据。
其它	NA	FW	当该请求由 SETUP 令牌接收时，硬件为 FW 产生 CPUDEC 中断请求。FW 从 CPUDEC 中断请求对请求的内容进行译码，并且执行必需的处理。

20.9.1 初始化处理

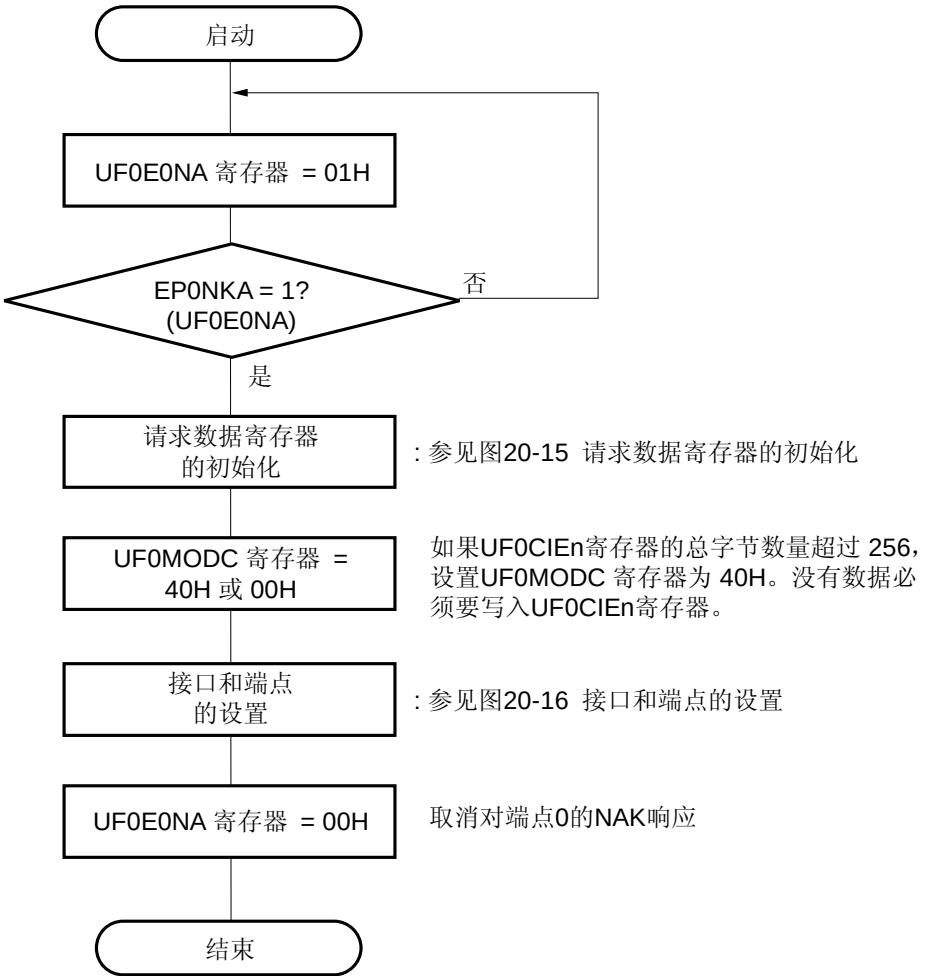
初始化处理按照下面两种方式执行。

- 请求数据寄存器的初始化
- 中断的设置

当请求数据寄存器被初始化时，用于 GET_XXXX 请求的数据被写入，对该请求会自动返回一个值，并且一个端点被分配给一个接口。在中断设置中，不是必须检查的中断源可以使用 UF0IMn 寄存器（n = 0 至 4）来屏蔽。

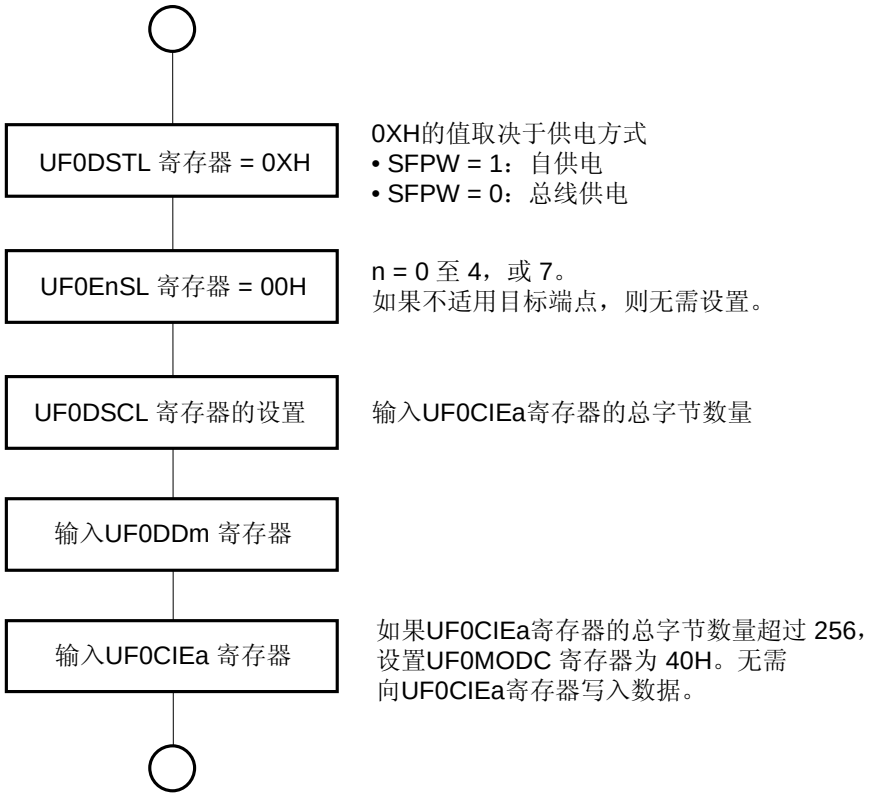
以下流程图展示上述处理。

图 20-14. 请求数据寄存器初始化



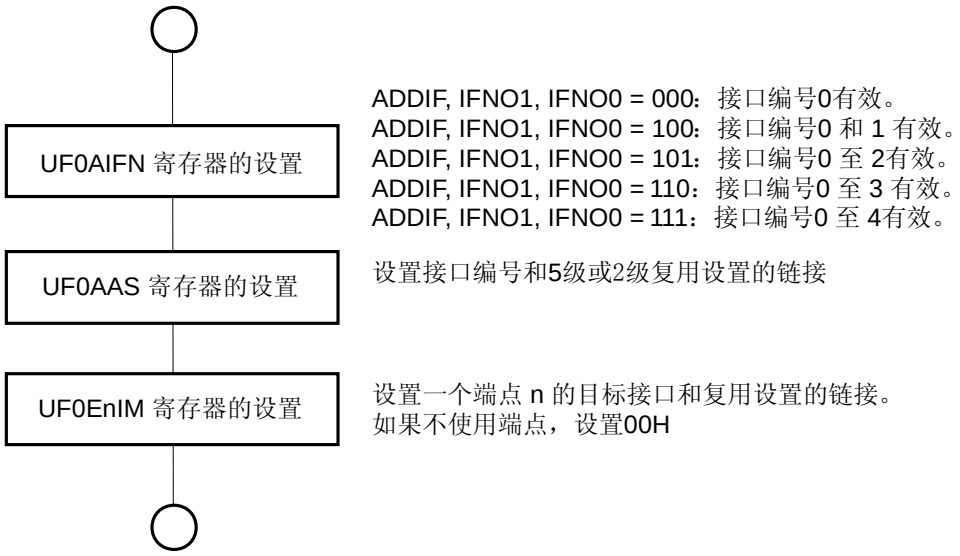
备注 n = 0 至 255

图 20-15. 请求数据寄存器区域的初始化



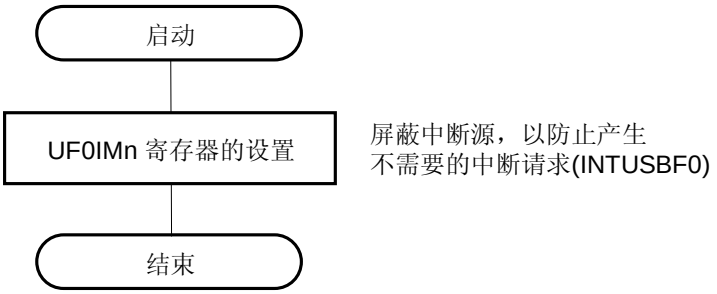
备注 m = 0 至 17
a = 0 至 255

图 20-16. 接口和端点的设置



备注 n = 1 至 4，7

图 20-17. 中断的设置

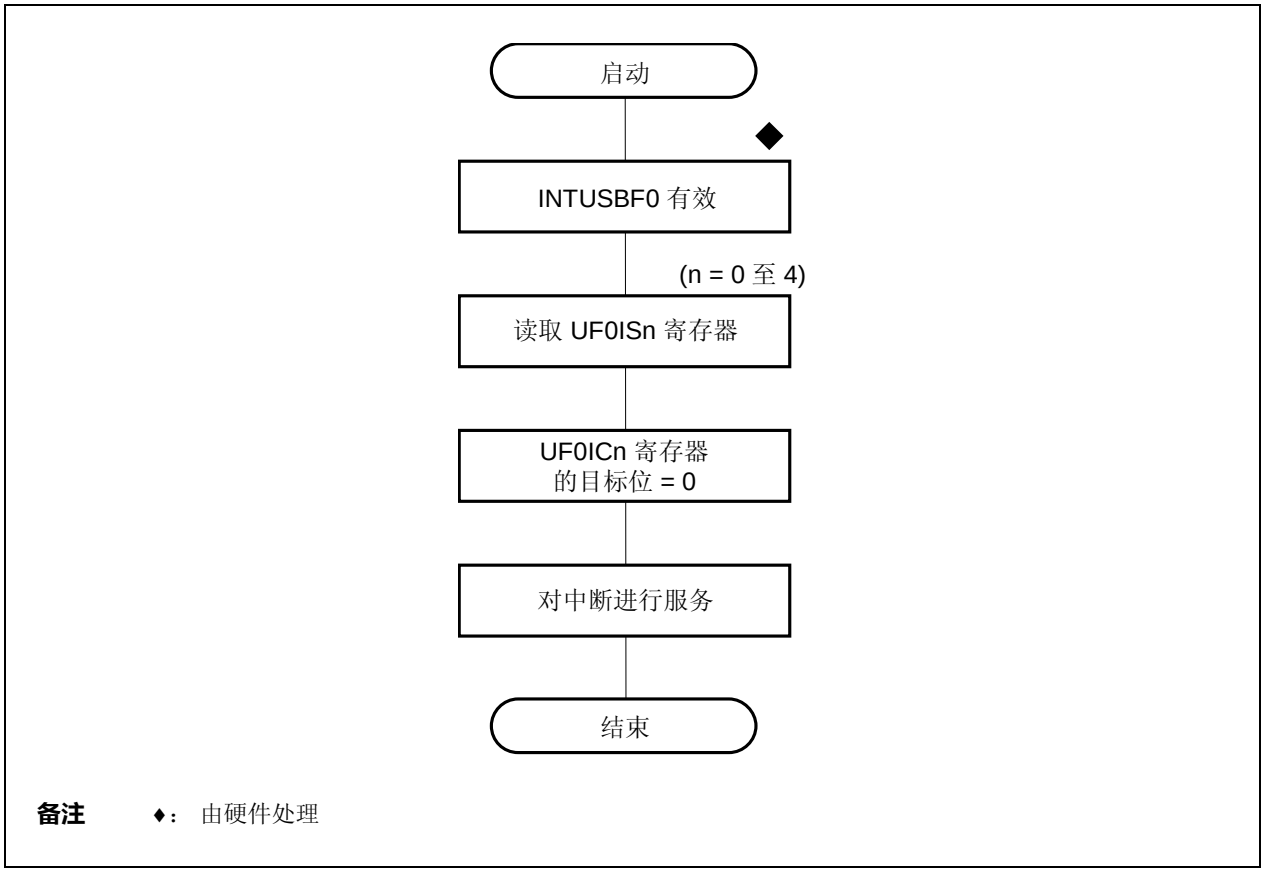


备注 n = 0 至 4

20.9.2 中断服务

以下流程图展示如何服务一个中断。

图 20-18. 中断服务



当给定的条件满足时，UF0ISn 寄存器下列各位由硬件自动清除（n = 0 至 4）。

- UF0IS1 寄存器的 E0INDT、E0ODT、SUCES、STG 和 CPUDEC 位
- UF0IS2 寄存器的 BKI2DT、BKI1DT 和 IT1DT 位
- UF0IS3 寄存器的 BKO2FL、BKO2DT、BKO1FL 和 BKO1DT 位

因为由 UF0ICn 寄存器清除的中断源优先级比硬件设置的中断源优先级低，根据时序，中断源可能未被清除（n = 0 至 4）。

20.9.3 USB 主处理

USB 主处理涉及 USB 事务的处理。将要被处理的事务类型如下。

- 控制传输中全自动处理的请求
- 控制传输中自动处理的请求
  （SET_CONFIGURATION，SET_INTERFACE，SET_FEATURE，CLEAR_FEATURE）
- 控制传输的 CPUDEC 请求
- 批量传输（IN）的处理
- 批量传输（OUT）的处理
- 中断传输（IN）的处理

端点 n 的处理涉及数据传输的写入或读取。下面是对 PIO 的流程图。

（1）控制传输中全自动处理的请求

因为控制传输中全自动处理的请求由硬件执行，不能由 FW 引用。因此，FW 不必为该请求执行任何特殊处理。

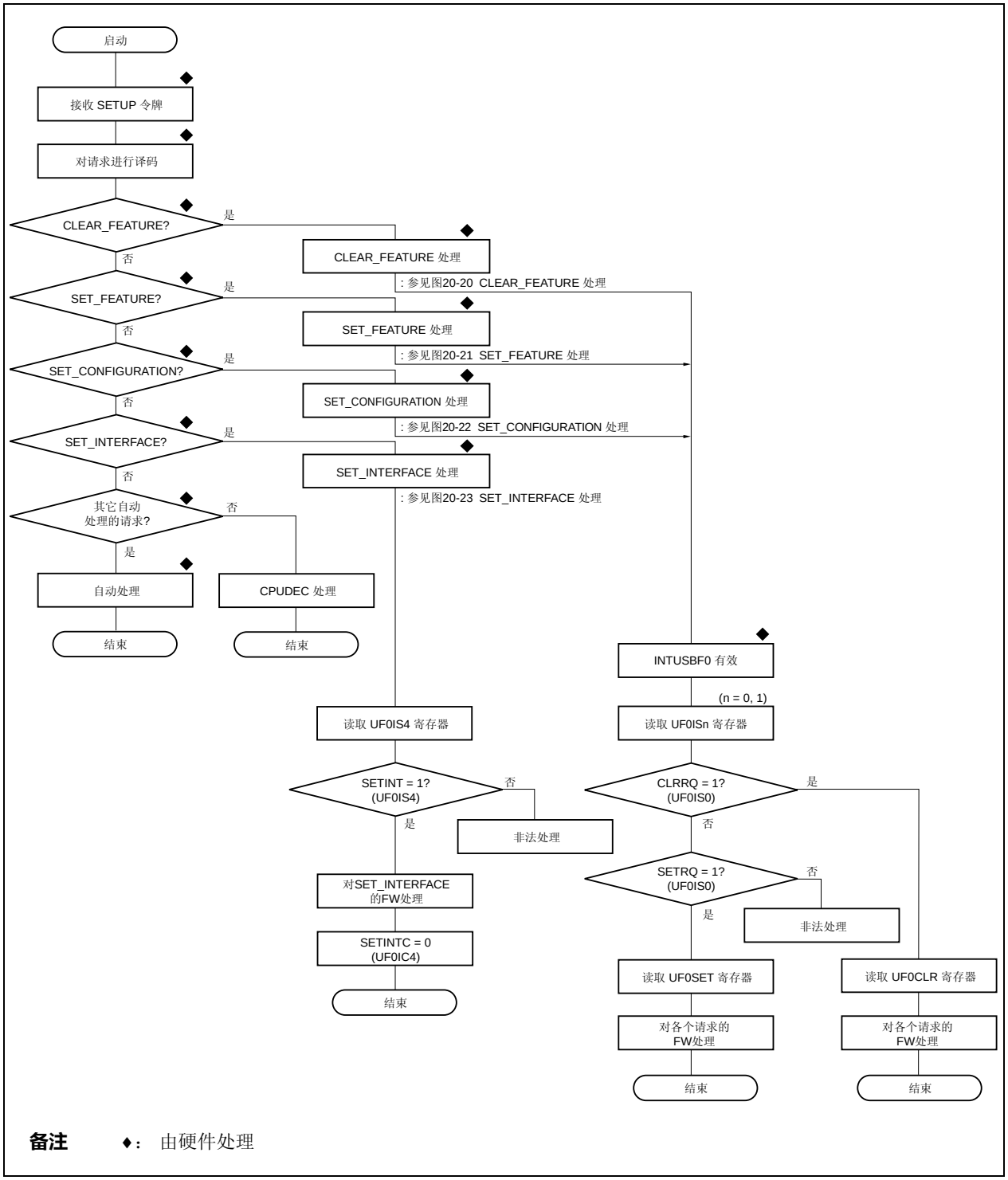
（2）控制传输中自动处理的请求

（SET_CONFIGURATION，SET_INTERFACE，SET_FEATURE，CLEAR_FEATURE）

对于控制传输中自动处理的请求，例如 SET_CONFIGURATION、SET_INTERFACE、SET_FEATURE 和 CLEAR_FEATURE 请求，为其写入一个寄存器的处理由硬件自动执行，但是在设备端发出一个中断请求用来识别。如果不需要执行特殊处理，这个处理可能被忽略。

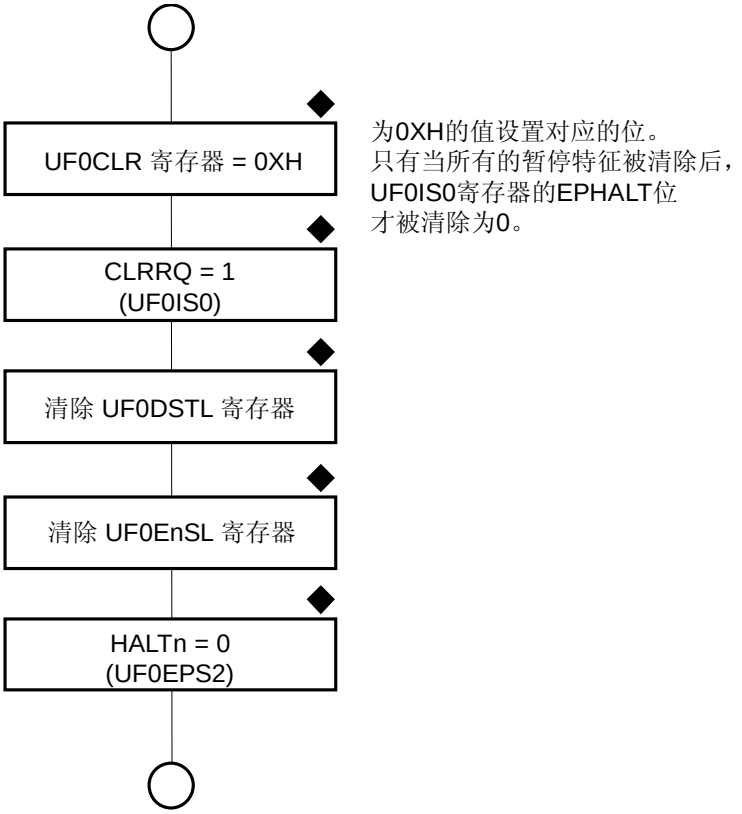
流程图如下所示。

图 20-19. 控制传输中自动处理的请求



备注 ♦: 由硬件处理

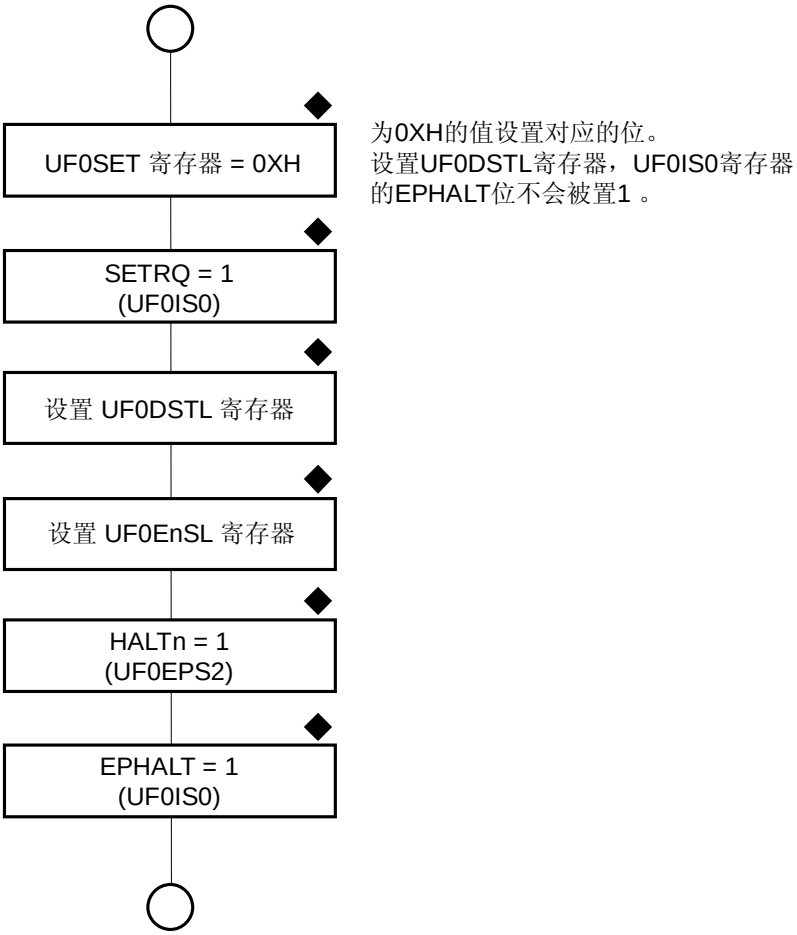
图 20-20. CLEAR_FEATURE 处理



备注 **1.** n = 0 至 4，7

2. ♦： 由硬件处理

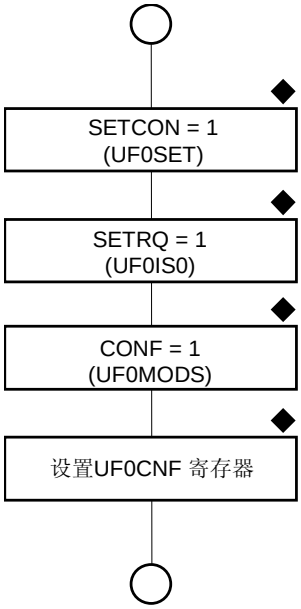
图 20-21. SET_FEATURE 处理



备注 **1.** n = 0 至 4，7

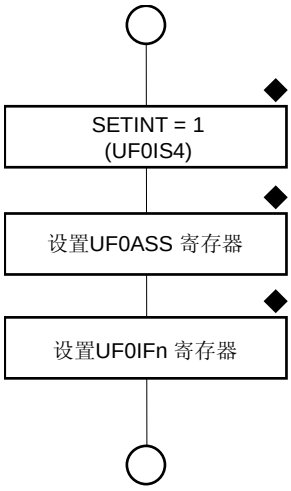
2. ♦： 由硬件处理

图 20-22. SET_CONFIGURATION 处理



备注 ♦： 由硬件处理

图 20-23. SET_INTERFACE 处理



备注 **1.** n = 0 至 4
2. ♦： 由硬件处理

（3）控制传输的 CPUDEC 请求

CPUDEC 请求可以被分为三种处理类型：控制传输（写）、控制传输（读）和控制传输（无数据）。控制传输（写）表示一个在数据阶段使用 OUT 事务的请求（例如 SET_DESCRIPTOR）。控制传输（读）表示一个在数据阶段使用 IN 事务的请求（例如 GET_DESCRIPTOR）。控制传输（无数据）表示一个没有数据阶段的请求（例如 SET_CONFIGURATION）。

流程图如下所示。

图 20-24. 控制传输的 CPUDEC 请求（1/12）

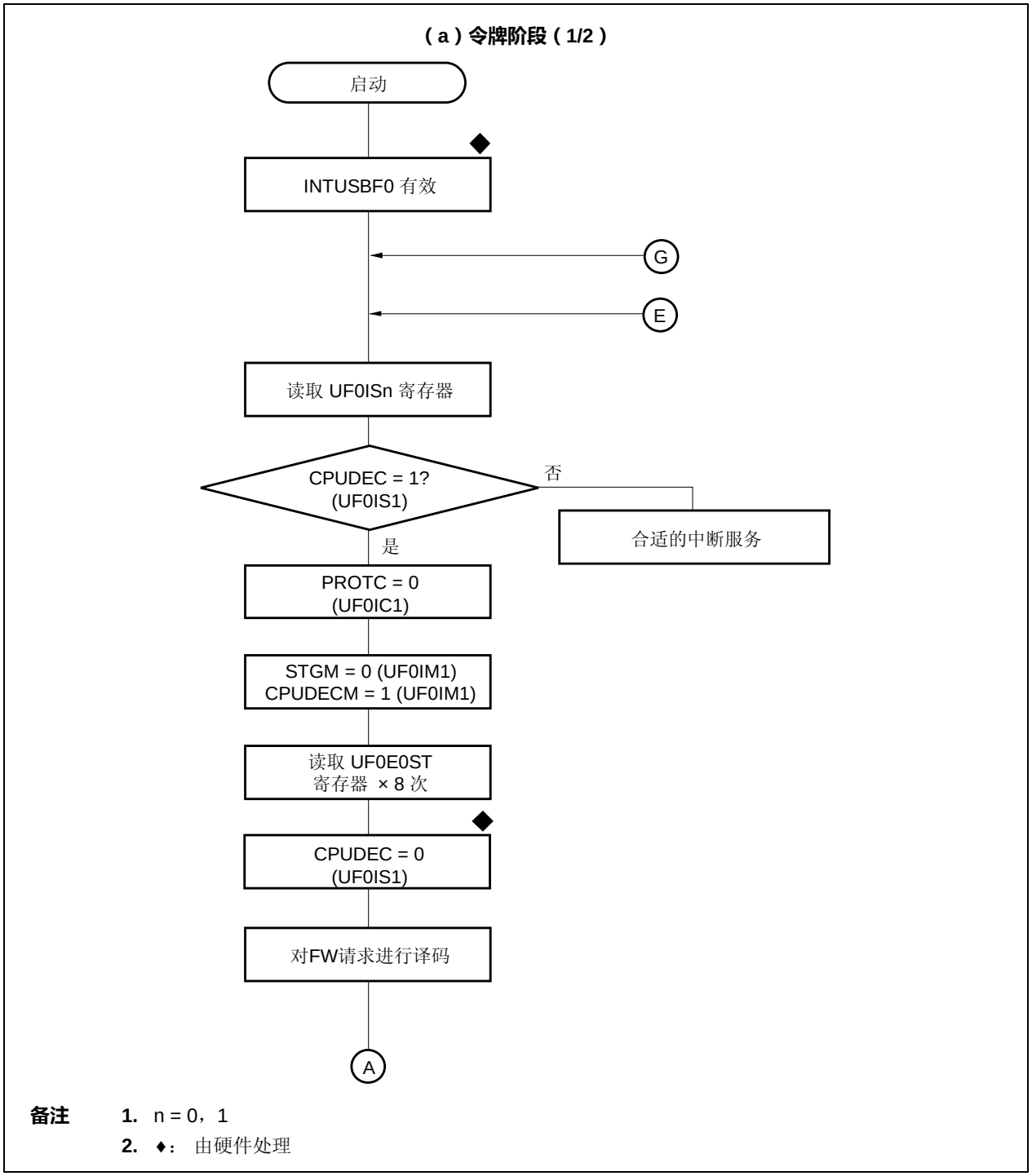
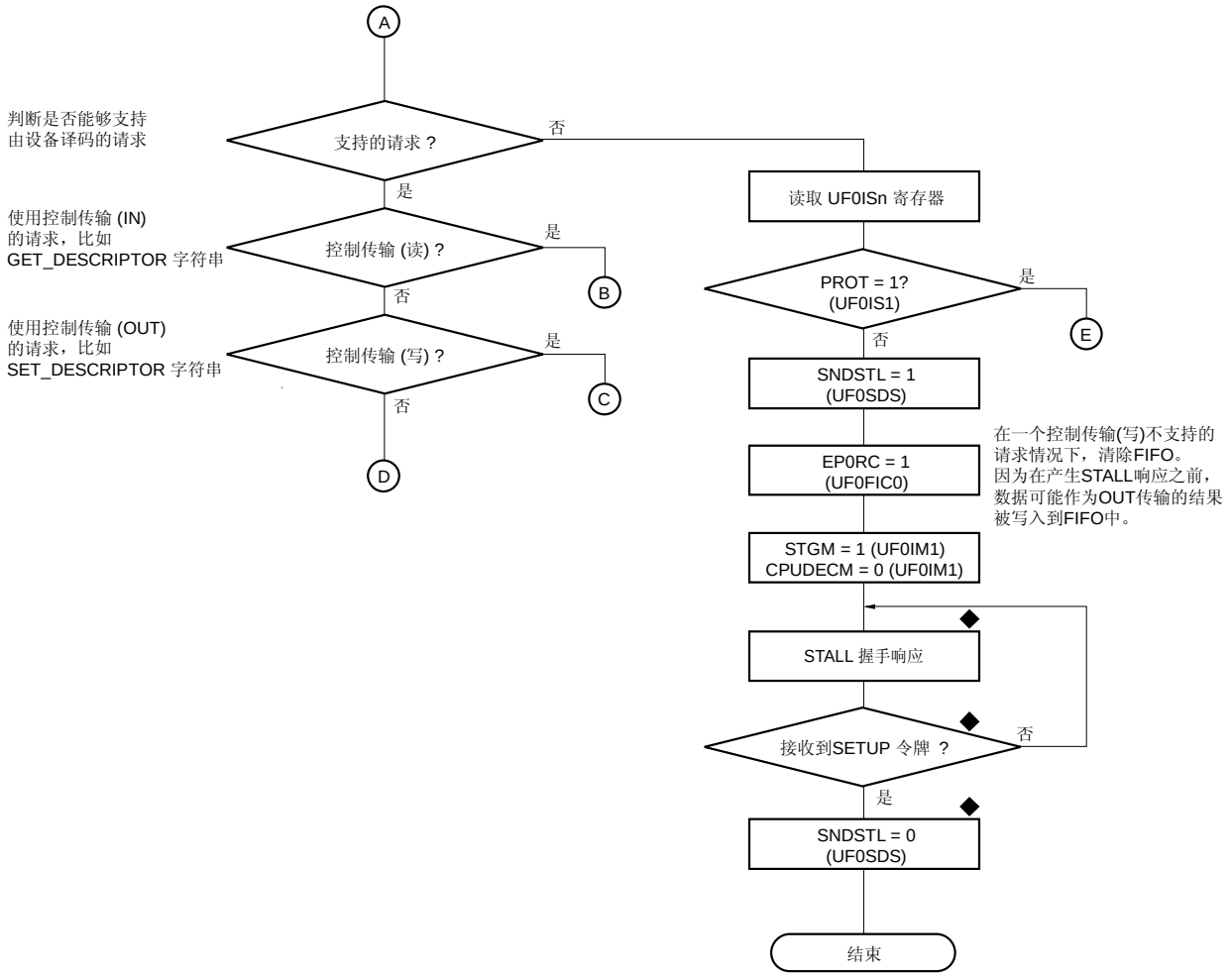


图 20-24. 控制传输的 CPUDEC 请求（2/12）

（a）令牌阶段（2/2）

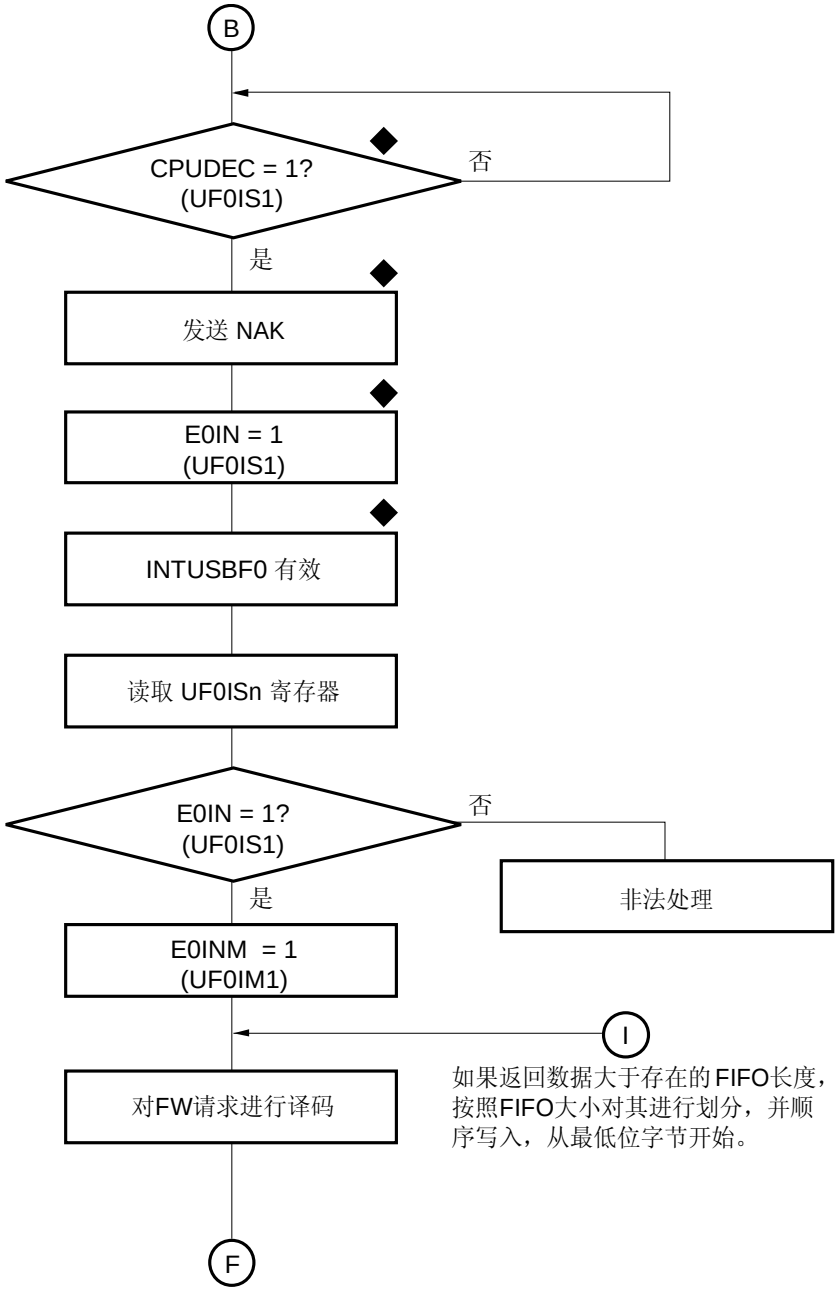


备注 **1.** n = 0，1

2. ♦： 由硬件处理

图 20-24. 控制传输的 CPUDEC 请求（3/12）

（b）控制传输（读）（1/4）

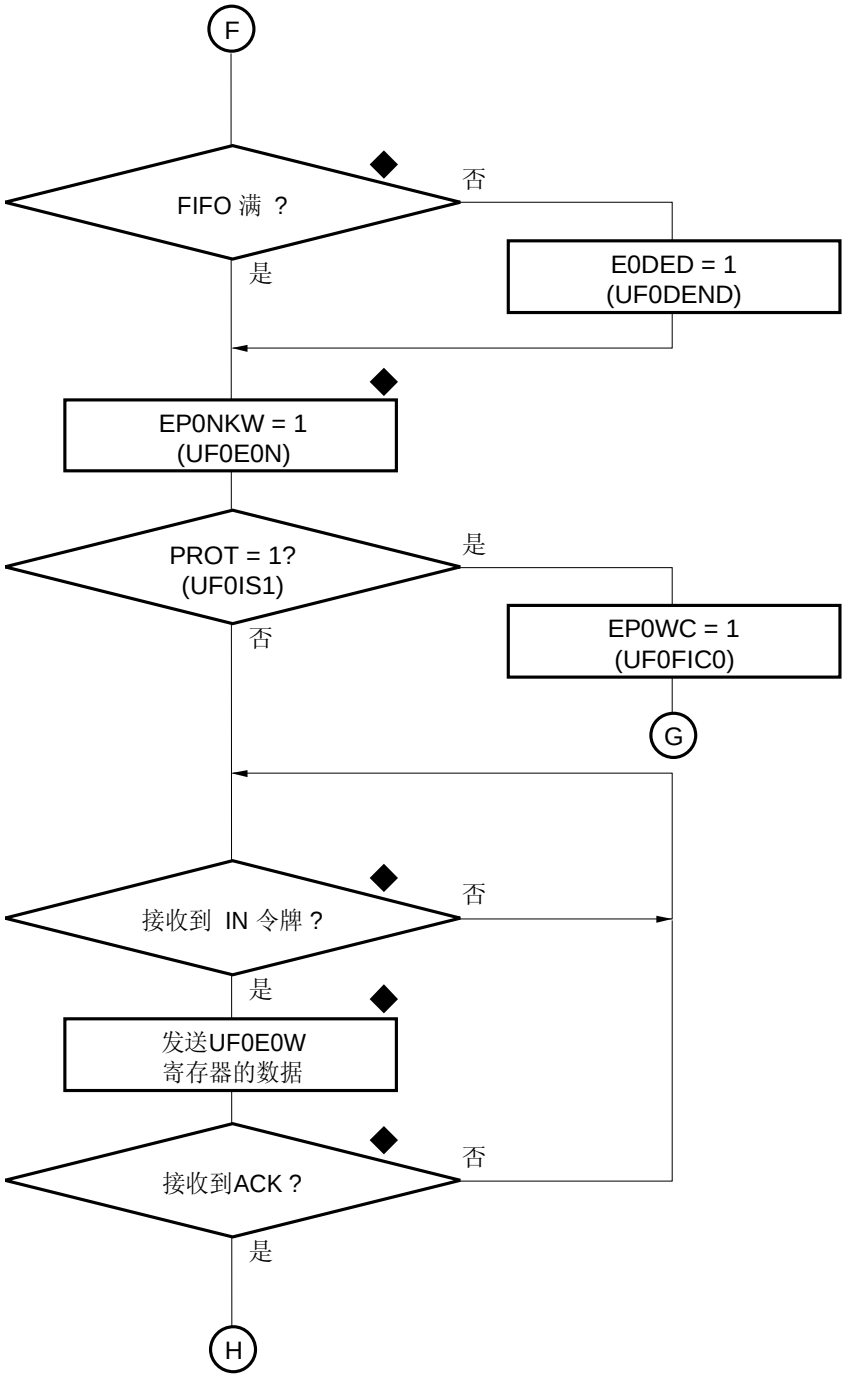


备注 **1.** n = 0，1

2. ♦： 由硬件处理

图 20-24. 控制传输的 CPUDEC 请求（4/12）

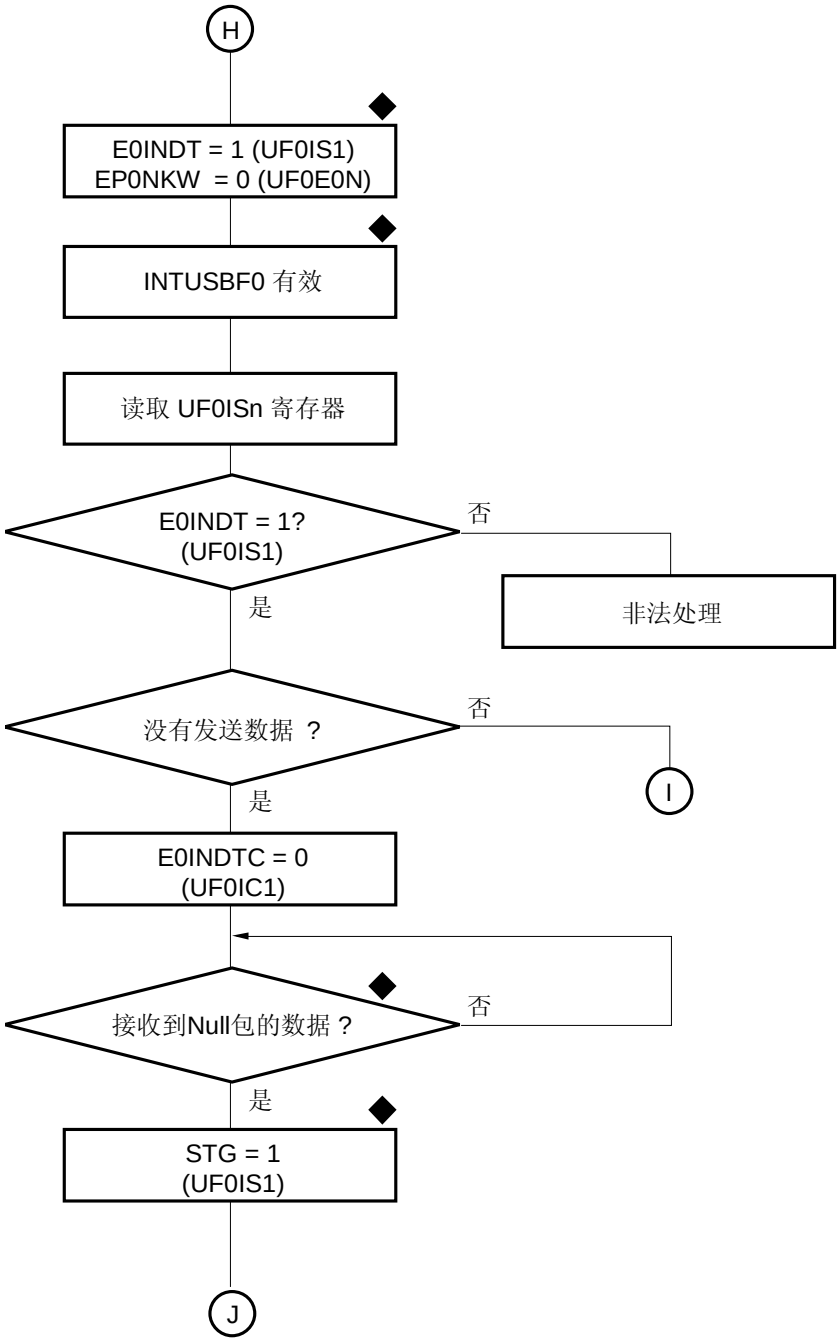
（b）控制传输（读）（2/4）



备注 ♦： 由硬件处理

图 20-24. 控制传输的 CPUDEC 请求（5/12）

（b）控制传输（读取）（3/4）

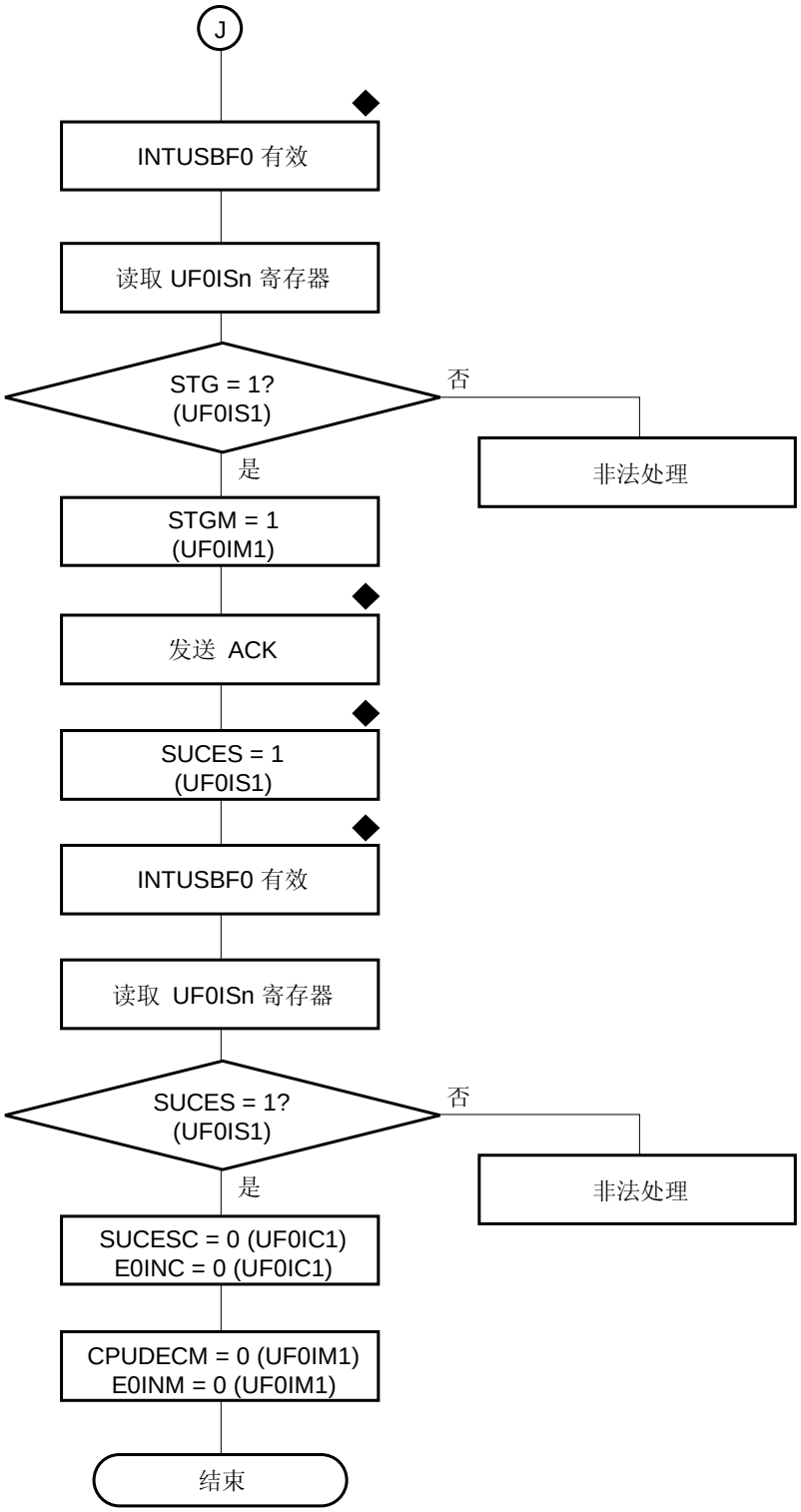


备注 **1.** n = 0，1

2. ♦： 由硬件处理

图 20-24. 控制传输的 CPUDEC 请求（6/12）

（b）控制传输（读取）（4/4）

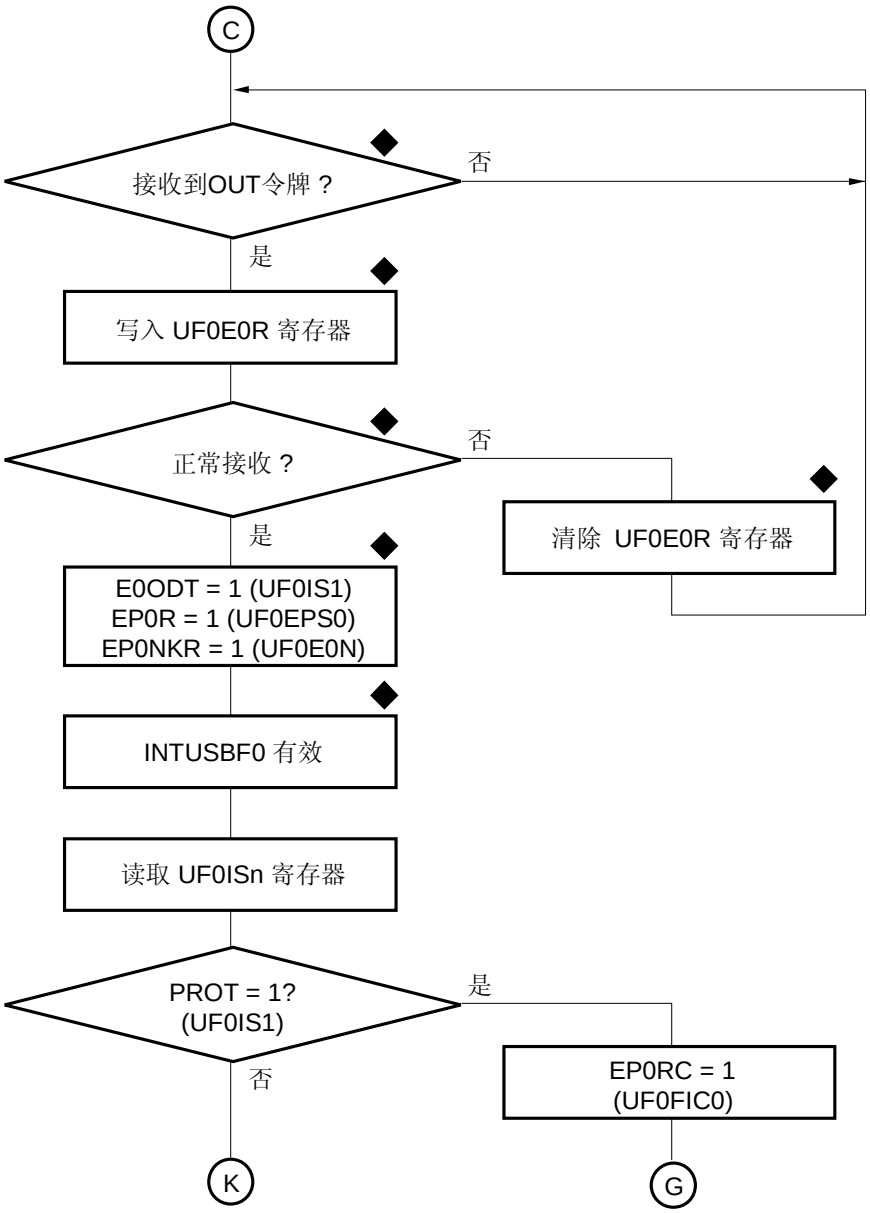


备注 **1.** n = 0，1

2. ♦： 由硬件处理

图 20-24. 控制传输的 CPUDEC 请求（7/12）

（c）控制传输（写）（1/4）

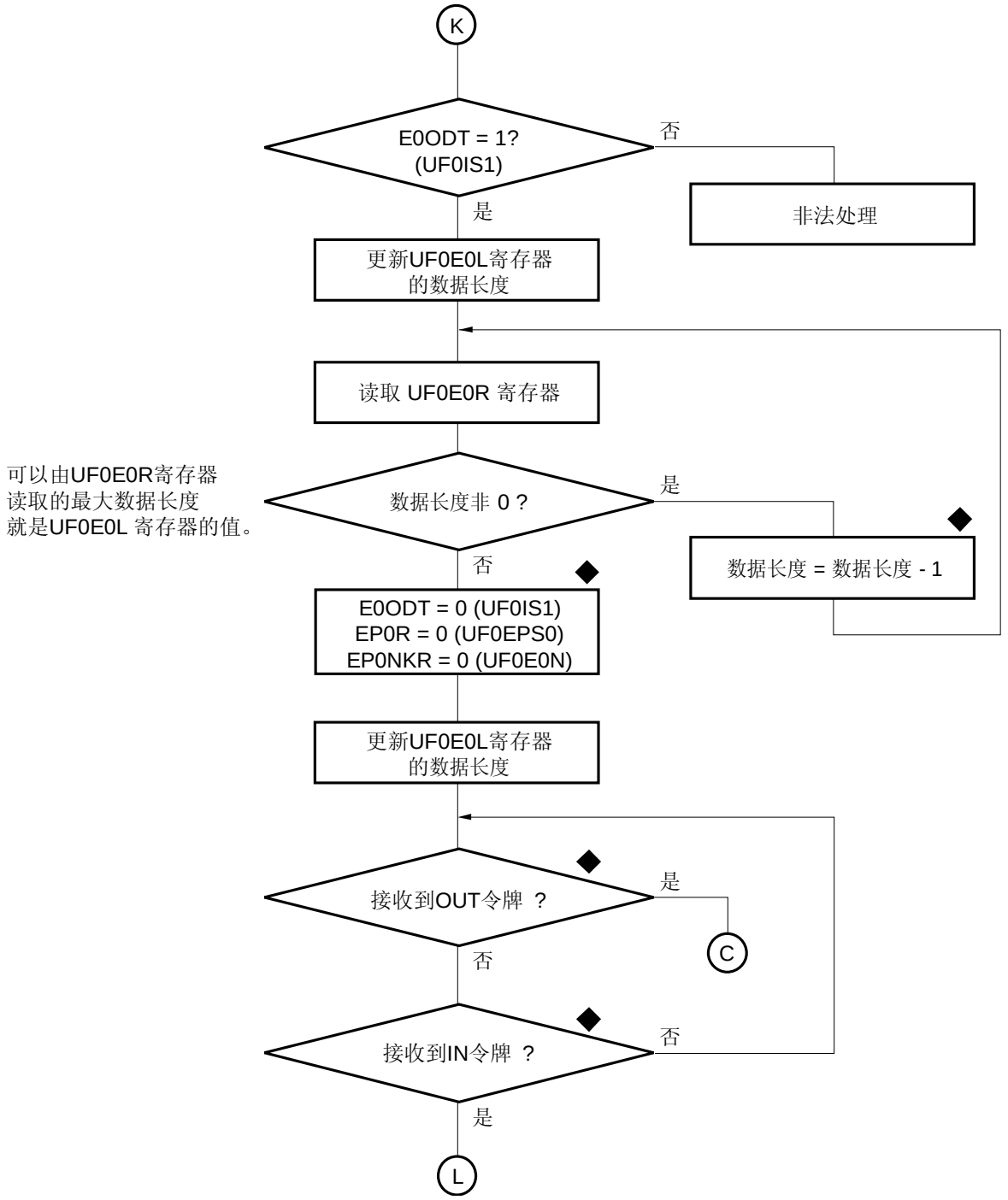


备注 **1.** n = 0，1

2. ♦： 由硬件处理

图 20-24. 控制传输的 CPUDEC 请求（8/12）

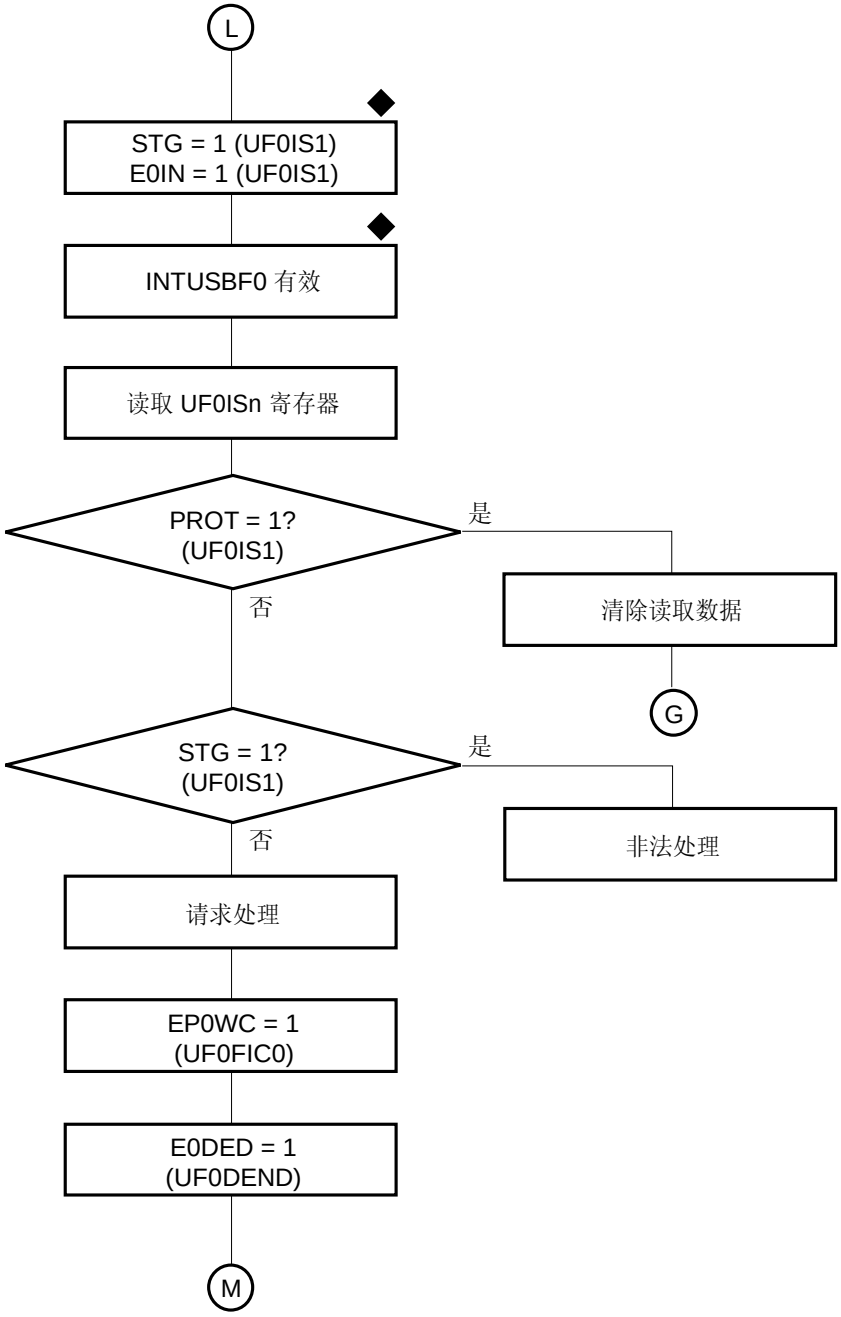
（c）控制传输（写）（2/4）



备注 ♦： 由硬件处理

图 20-24. 控制传输的 CPUDEC 请求（9/12）

（c）控制传输（写入）（3/4）

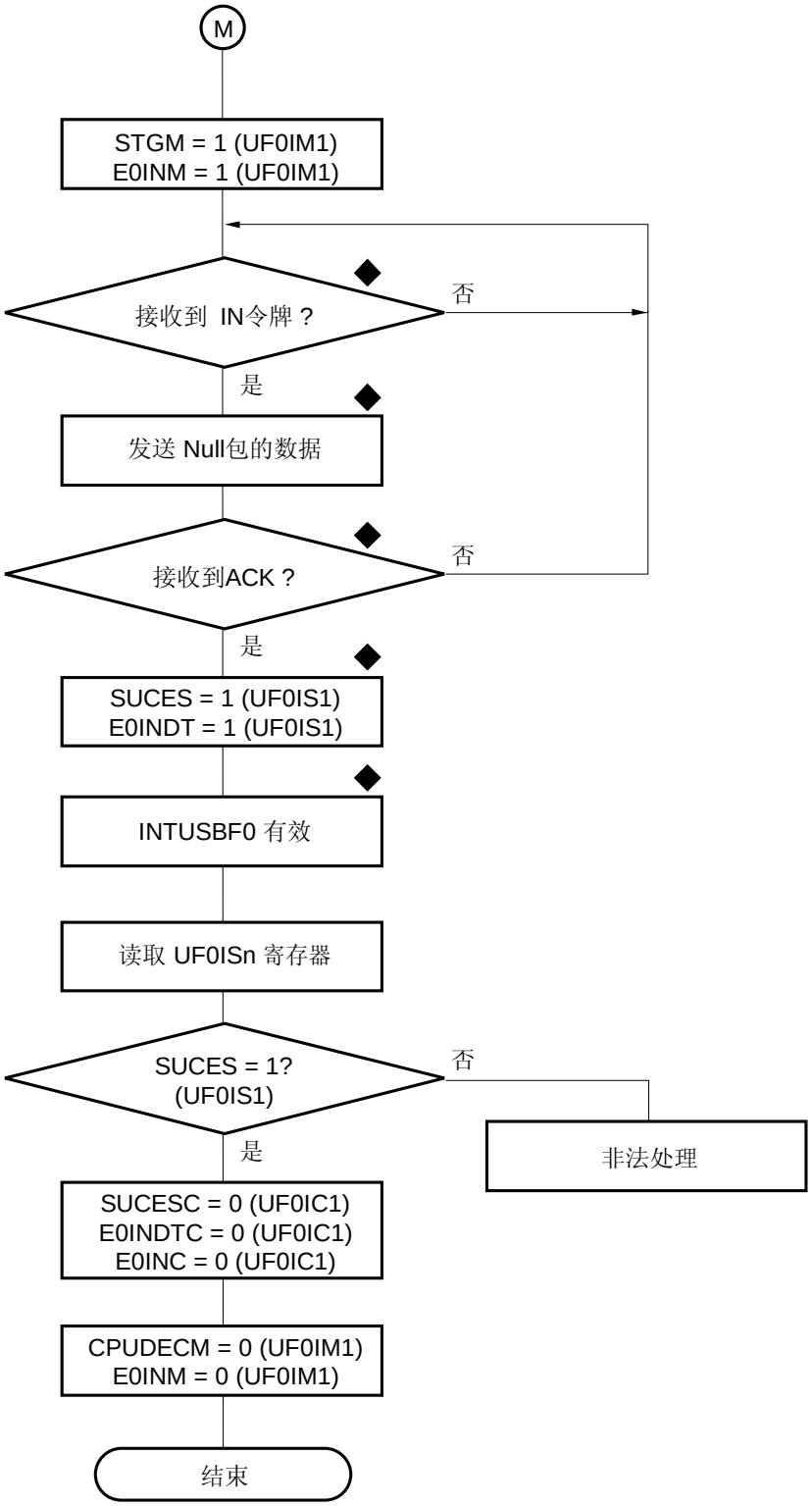


备注 **1.** n = 0，1

2. ♦： 由硬件处理

图 20-24. 控制传输的 CPUDEC 请求（10/12）

（c）控制传输（写入）（4/4）

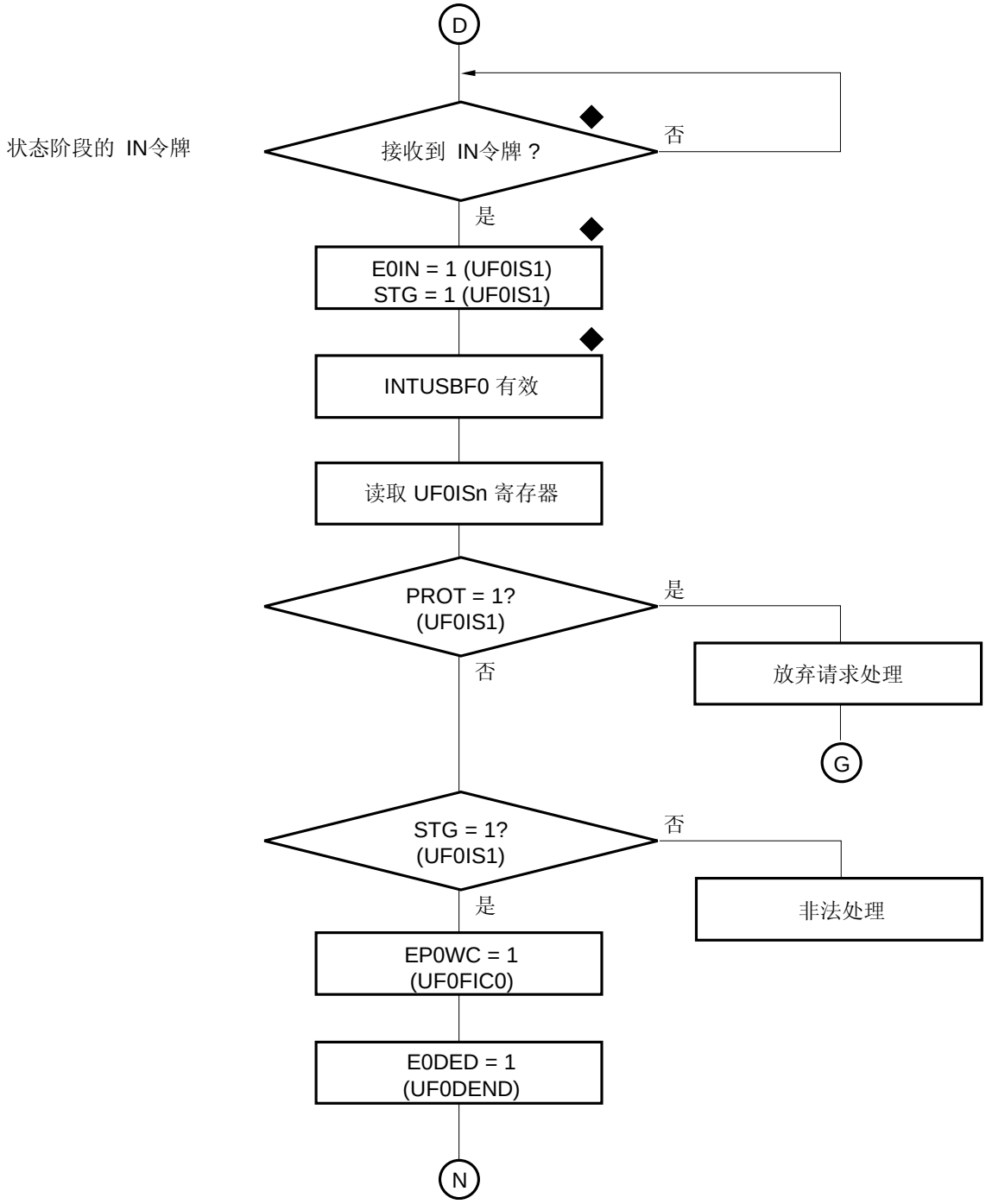


备注 **1.** n = 0，1

2. ♦： 由硬件处理

图 20-24. 控制传输的 CPUDEC 请求（11/12）

（d）控制传输（数据阶段除外）（1/2）

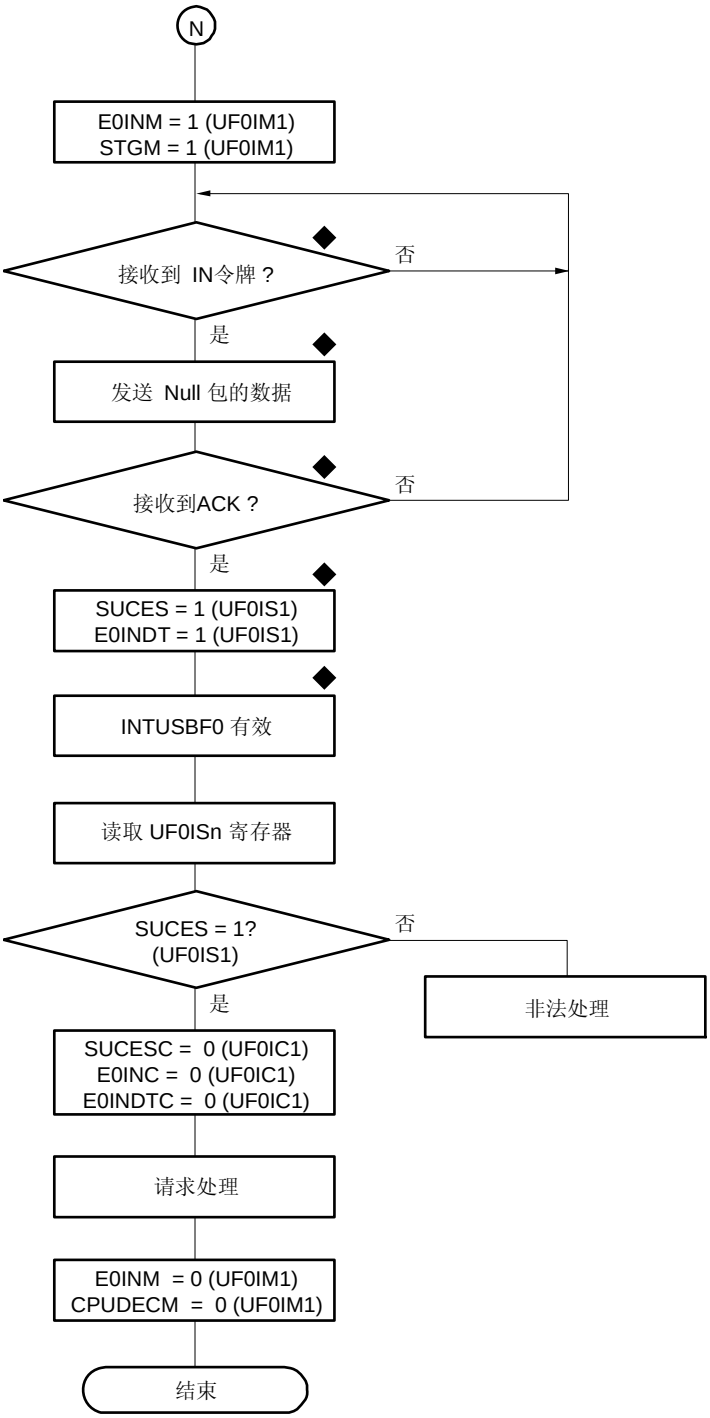


备注 **1.** n = 0，1

2. ♦： 由硬件处理

图 20-24. 控制传输的 CPUDEC 请求（12/12）

（d）控制传输（数据阶段除外）（2/2）



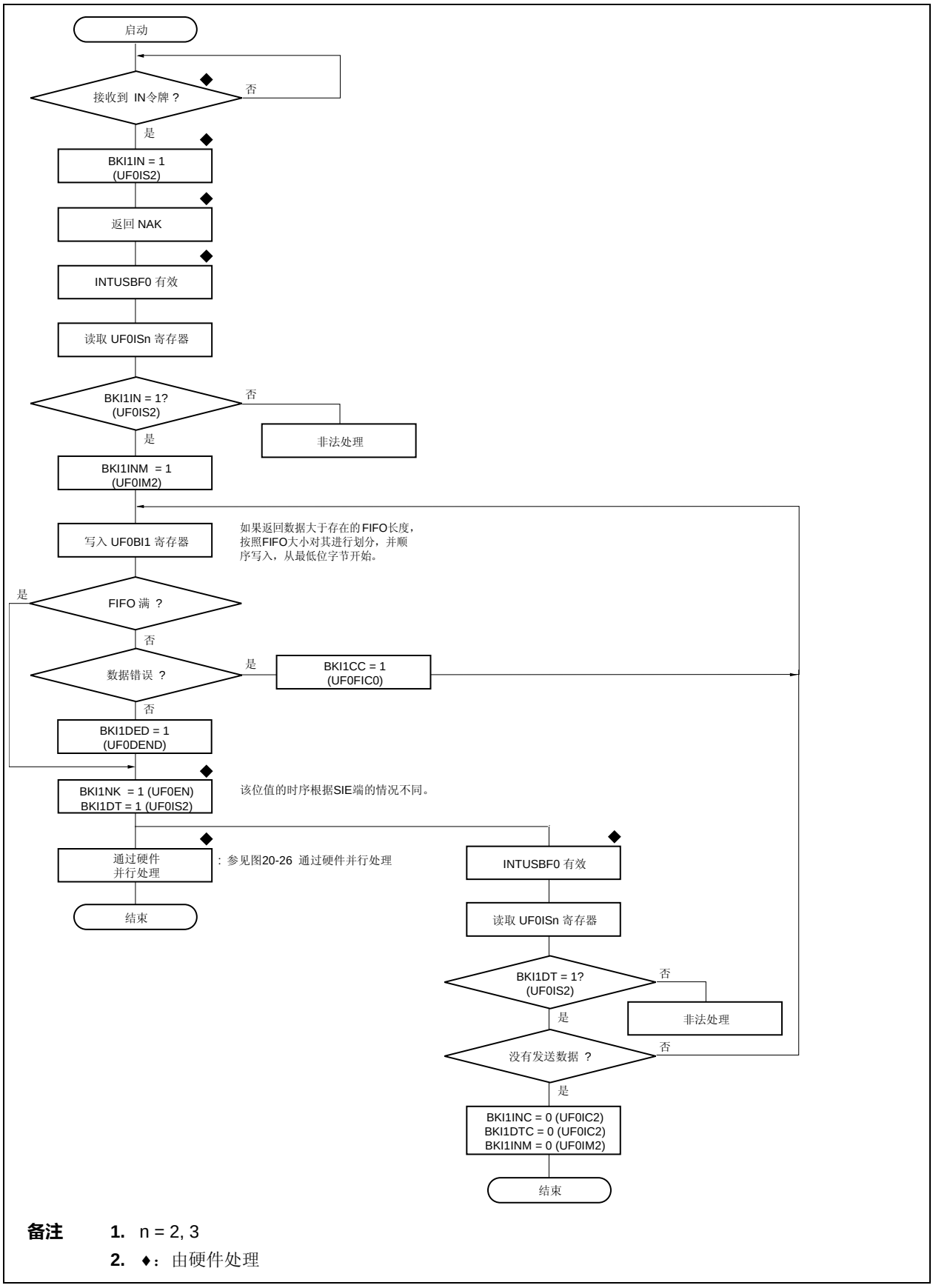
备注 **1.** n = 0，1

2. ♦： 由硬件处理

（4）批量传输（IN）的处理

批量传输（IN）被分配在端点 1 和端点 3。下面的流程图展示了如何控制端点 1。端点 3 可以按照同样的顺序来控制。要使用该流程图作为端点 3 的控制流程，因此，在流程图中读取端点 1 的位名称，与端点 3 的那些位名称类似。

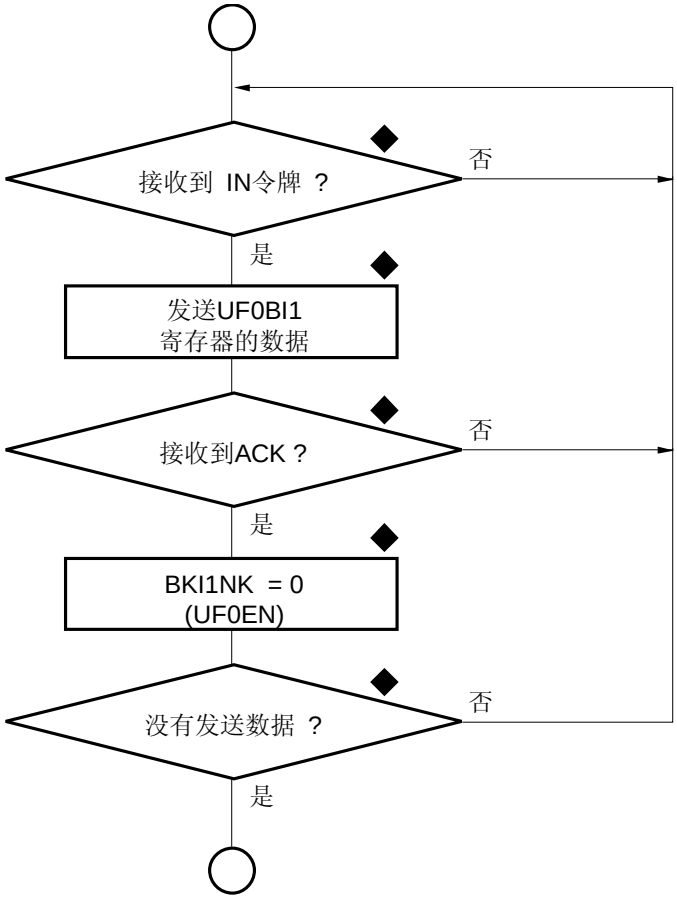
图 20-25. 批量传输（IN）的处理（端点 1）



备注 **1.** n = 2, 3

2. ♦：由硬件处理

图 20-26. 通过硬件并行处理

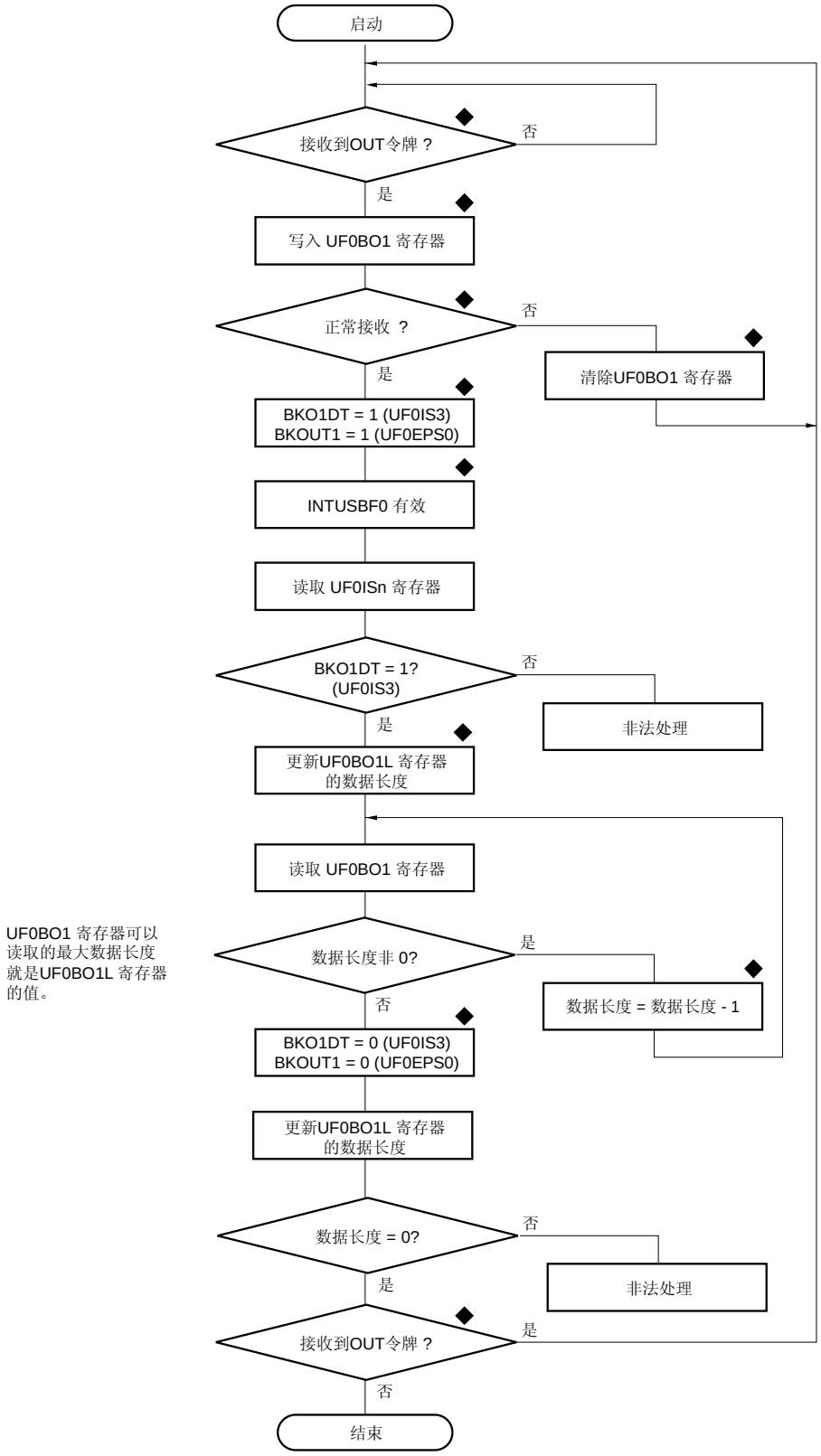


备注 ♦： 由硬件处理

（5）批量传输的处理（OUT）

批量传输（OUT）被分配在端点 2 和端点 4。下面的流程图展示了如何控制端点 2。端点 4 可以按照同样的顺序来控制。要使用该流程图作为端点 4 的控制流程，因此，在流程图中读取端点 2 的位名称，与端点 4 的那些位名称类似。

图 20-27. 批量传输（OUT）的正常处理（端点 2）

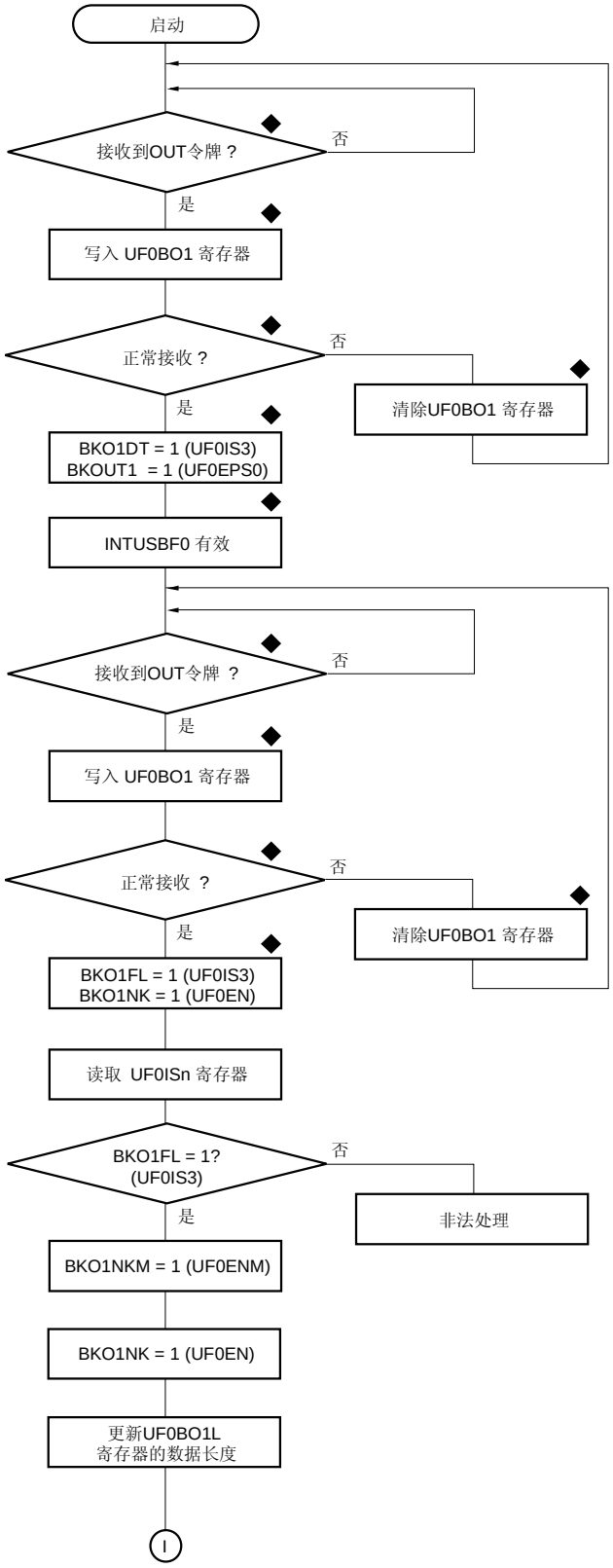


备注 **1.** n = 2, 3

2. ♦：由硬件处理

批量传输（OUT）期间，从主机发送的数据可能多于系统的期望值。V850ES/JG3-L 中用于批量传输（OUT）的端点 2 和端点 4 由两个 64 字节缓冲区组成，于是 NAK 响应被尽可能的抑制并且数据可以从 CPU 端读取，即使随着 USB 总线的传输速率增加，总线端被访问的频率也增加。因此，如果主机发送的数据数量多于系统的期望，在最坏情况下可能自动接收到 128 字节的多余数据。在这种情况下，当系统期望的数据的数量减少为两个包时，将控制流程从端点 2 和端点 4 的正常处理更改为下面表示的流程。下面的流程图展示了如何控制端点 2。端点 4 可以按照同样的顺序来控制。要使用该流程图作为端点 4 的控制流程，因此，在流程图中读取端点 2 的位名称，与端点 4 的那些位名称类似。

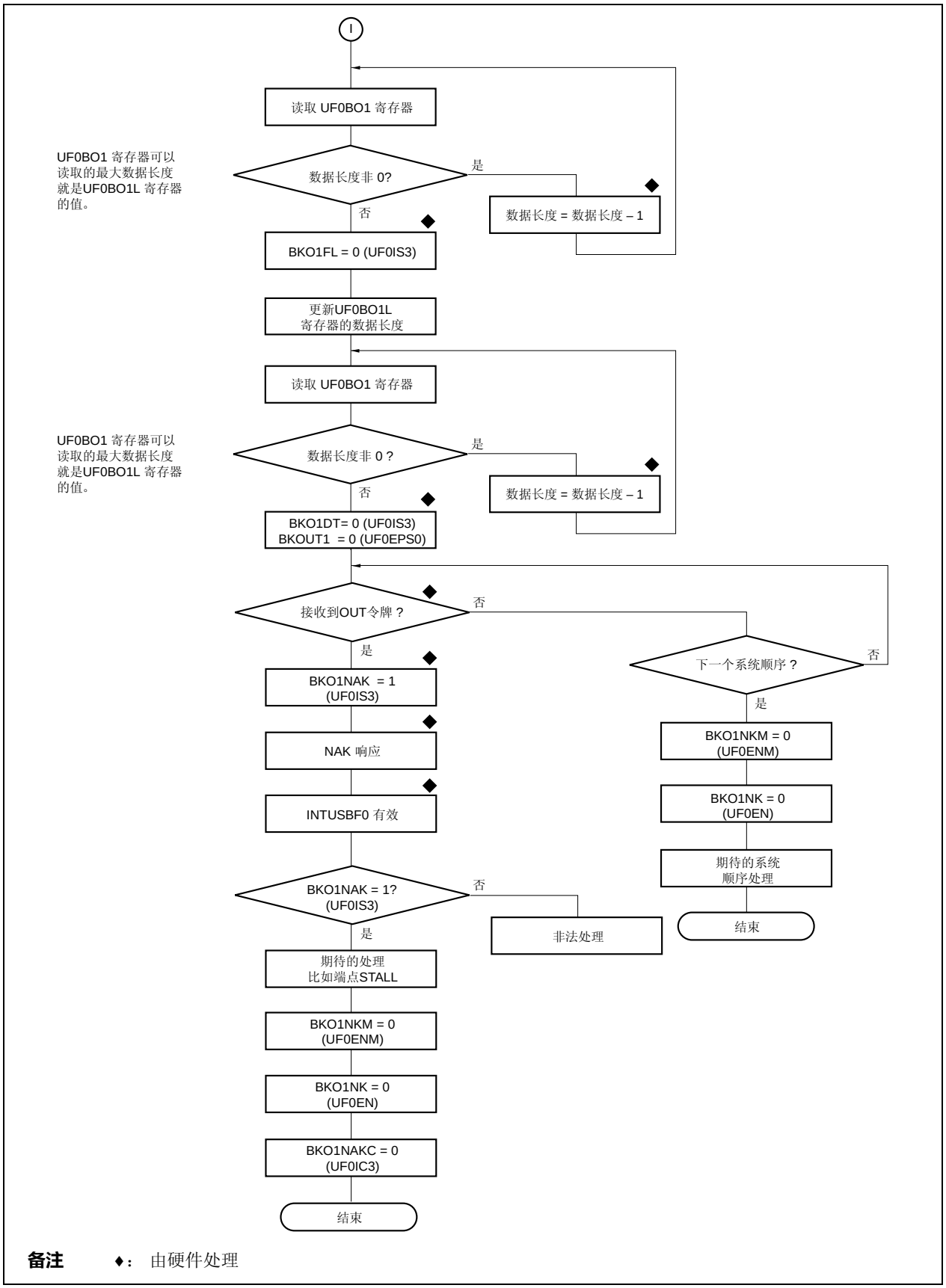
图 20-28. 如果发送的数据数量多于系统期望时的处理（端点 2）（1/2）



备注 **1.** n = 2，3

2. ♦：由硬件处理

图 20-28. 如果发送的数据数量多于系统期望时的处理（端点 2）（2/2）

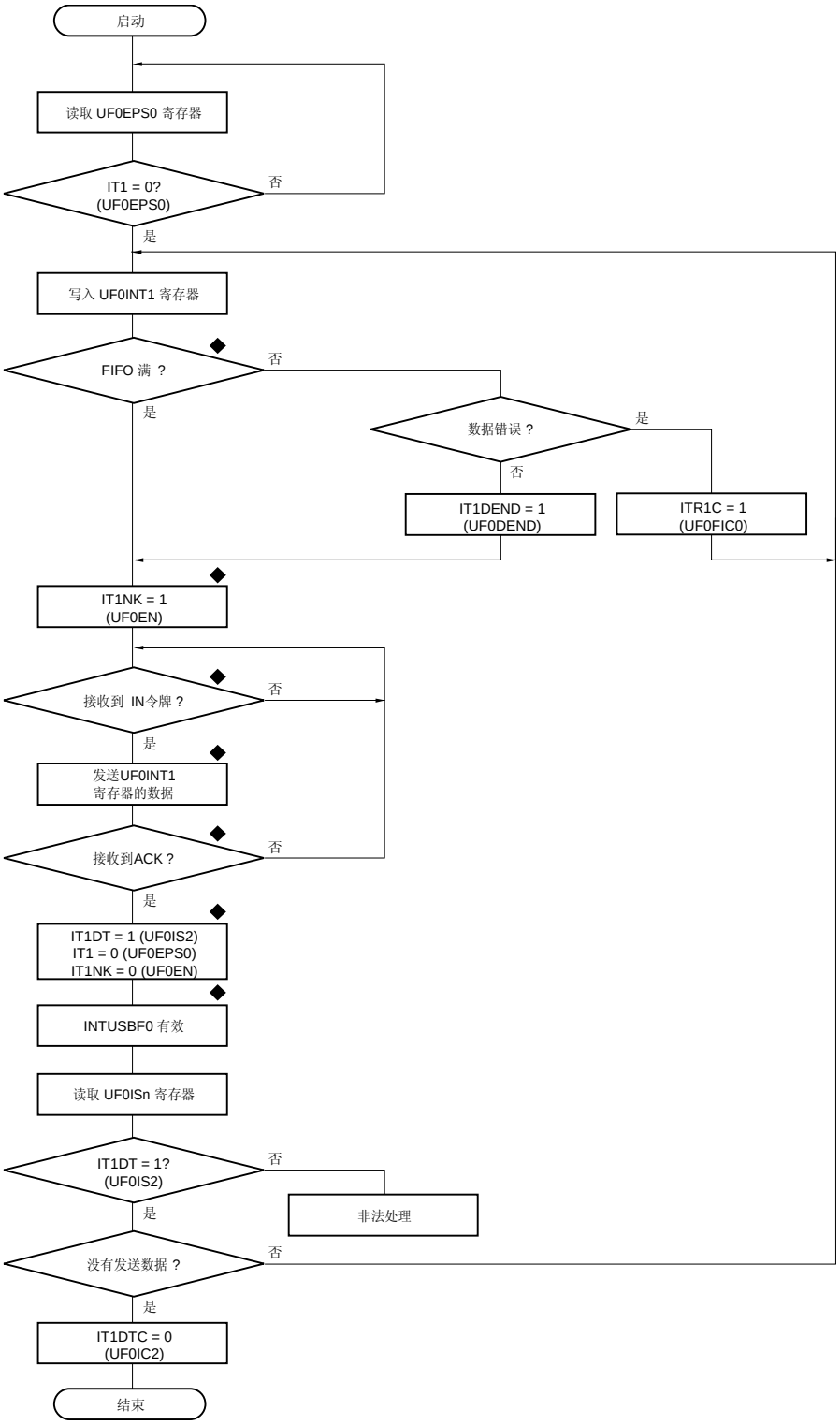


备注 ♦： 由硬件处理

(6) 中断传输(IN)的处理

中断传输(IN)被分配在端点 7。**图 20-29** 展示流程图。

图 20-29. 对中断传输(IN)的处理（端点 7）



备注 **1.** n = 2，3

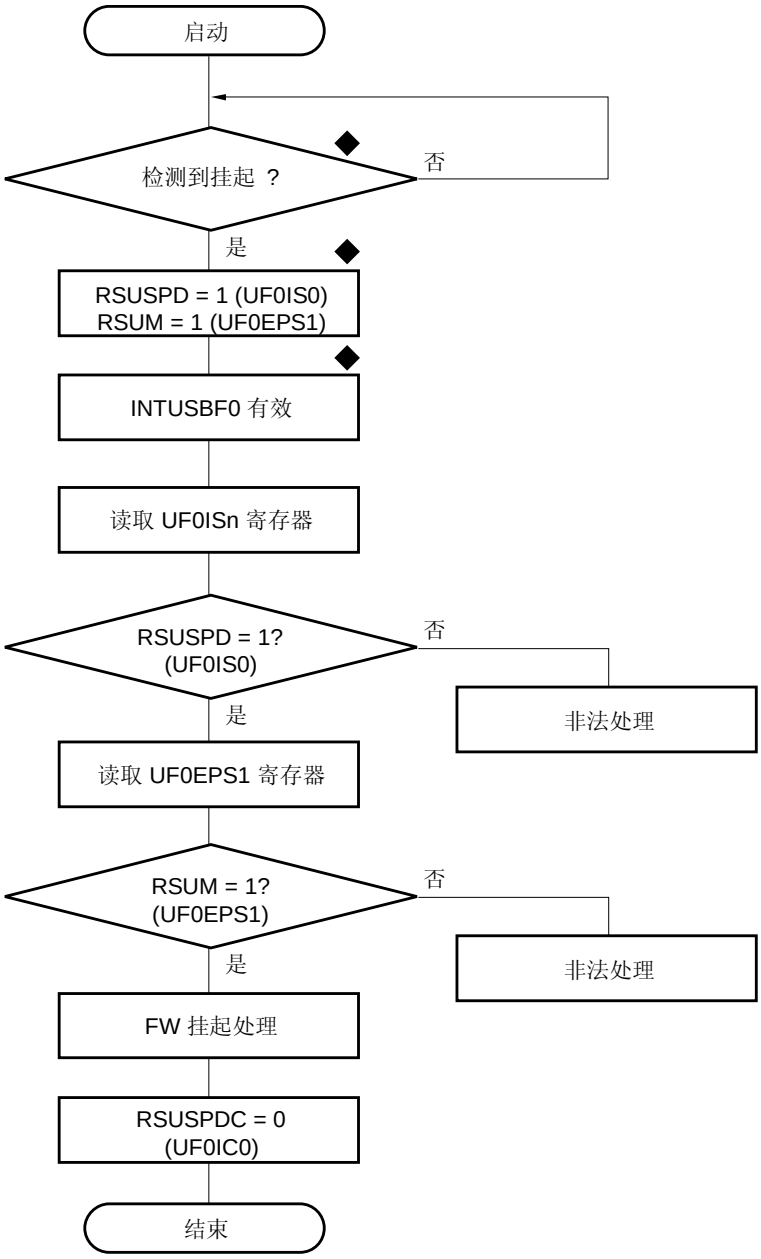
2. ♦：由硬件处理

20.9.4 挂起/恢复处理

挂起/恢复处理如何执行依赖于系统的配置。下面给出一个示例。

图 20-30. 挂起/恢复处理示例（1/3）

（a）挂起处理示例

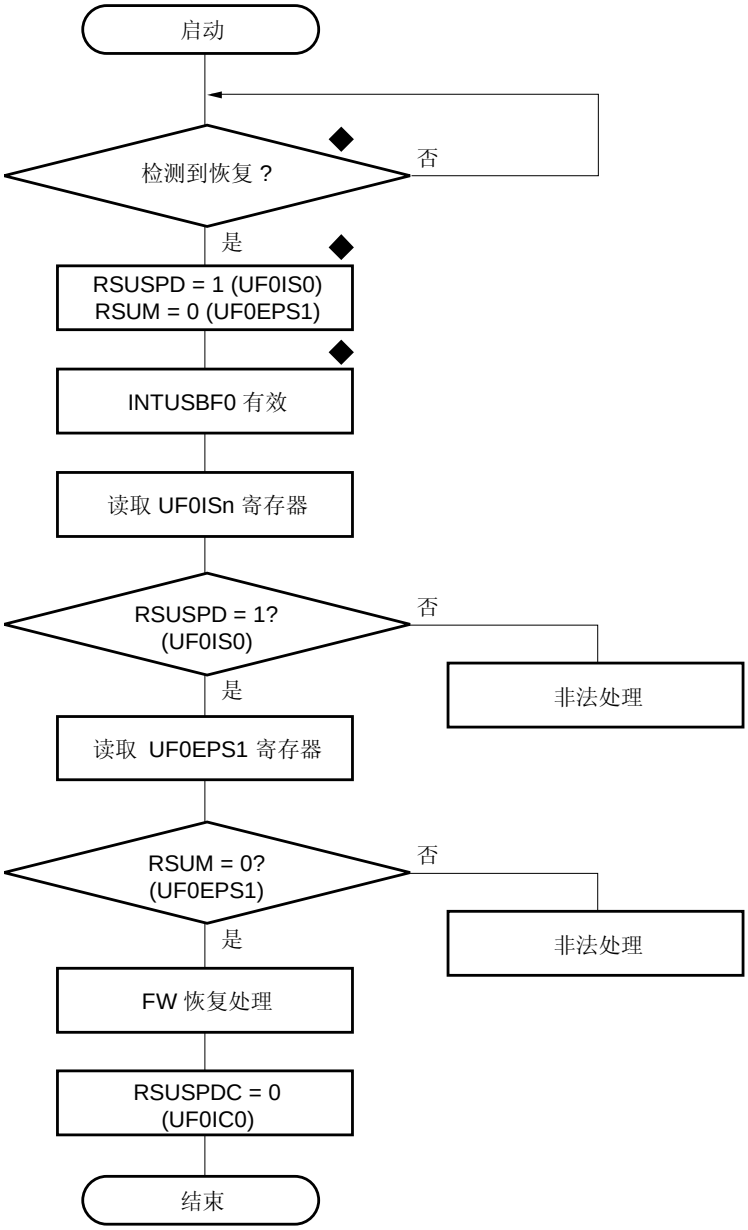


备注 **1.** n = 0，1

2. ♦： 由硬件处理

图 20-30. 挂起/恢复处理示例（2/3）

（b）恢复处理示例

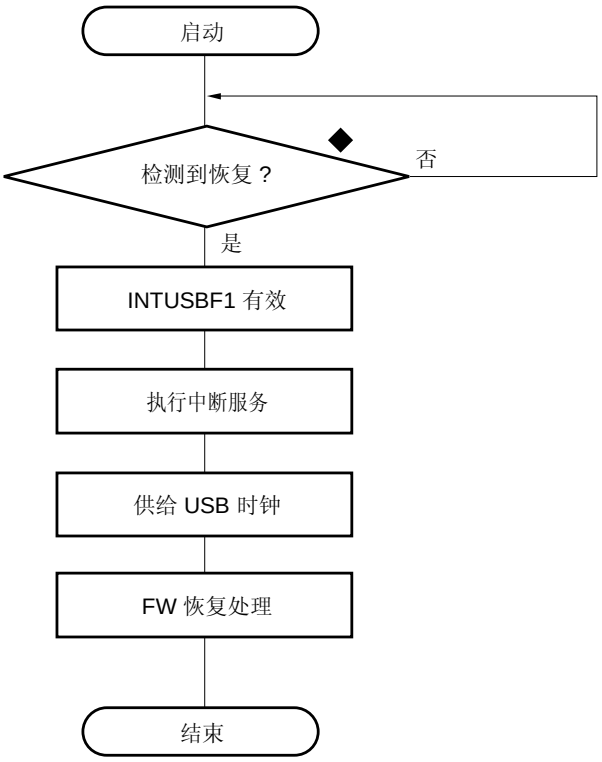


备注 **1.** n = 0，1

2. ♦： 由硬件处理

图 20-30. 挂起/恢复处理示例（3/3）

（c）恢复处理（当供给 USBF 的 USB 时钟停止时）示例



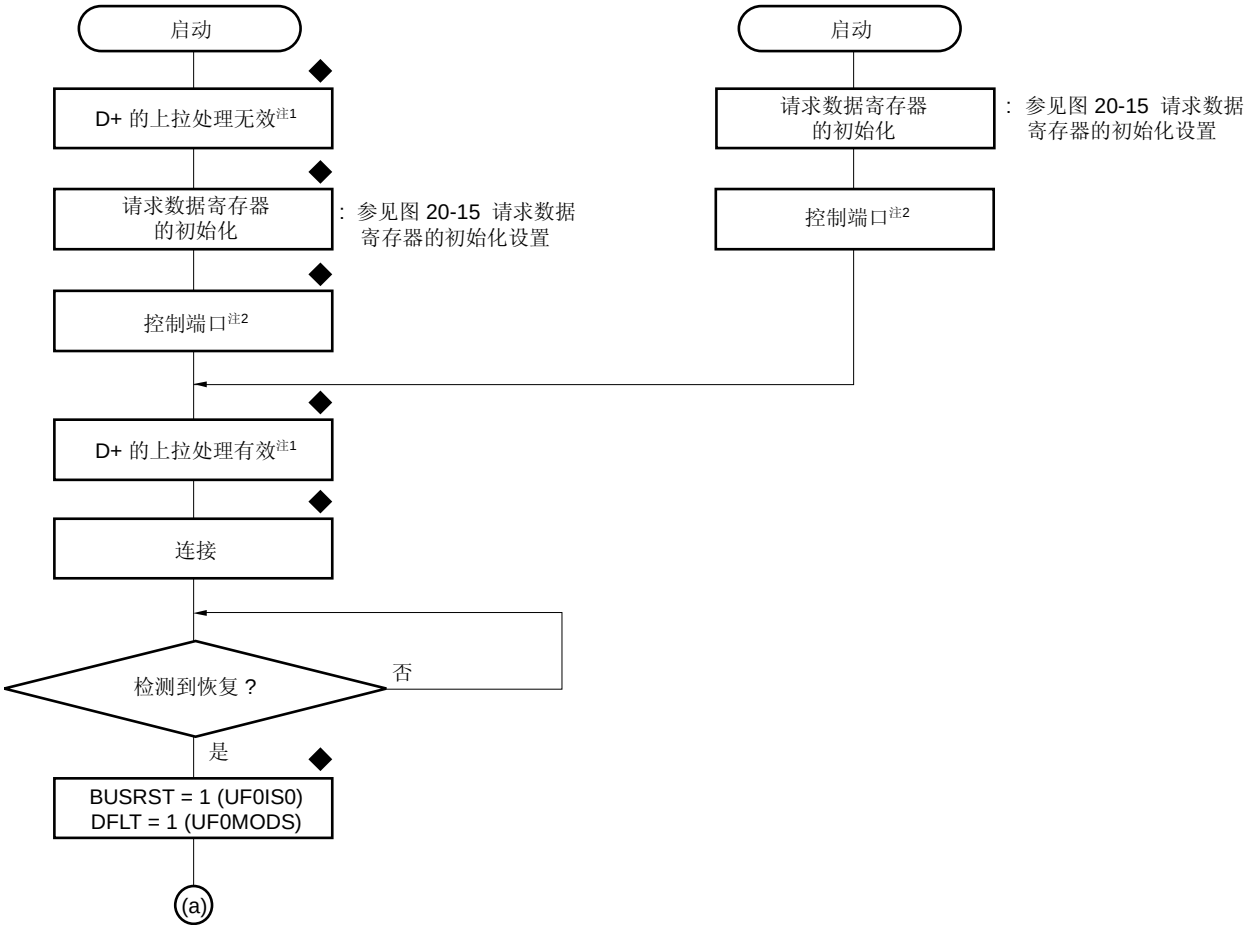
备注 ♦： 由硬件处理

20.9.5 电源上电后的处理

电源上电后要执行的处理依赖于系统的配置。下面给出一个示例。

图 20-31. 电源上电/电源失败后的处理示例（1/3）

（a）电源应用后的处理（1/2）

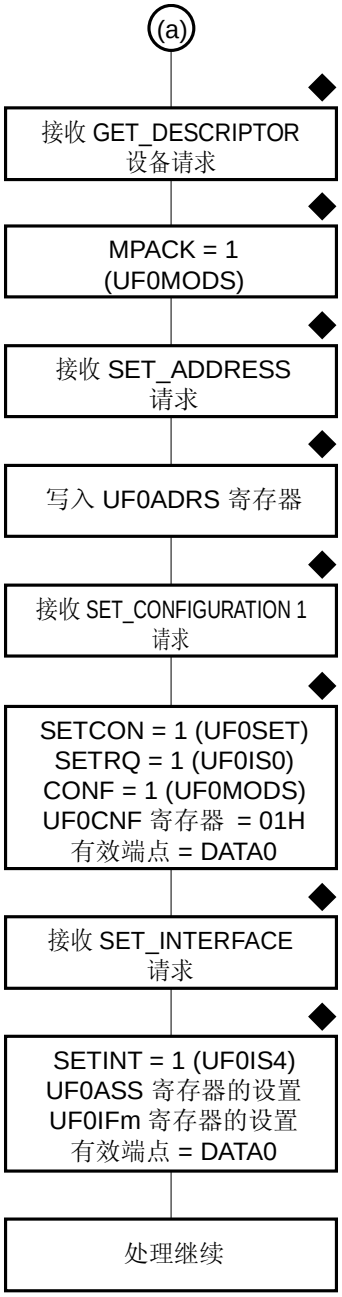


注 1. 使用一个通用端口引脚作为信号来控制 USB 总线的上拉电阻的开关。
2. 在**注 1** 中分配的通用端口引脚的输入模式或控制模式可以选择为默认值。注意电源应用时 D+的上拉处理的有效电平。

备注 ♦： 由硬件处理

图 20-31. 电源上电/电源失败后的处理示例（2/3）

（a）电源上电后的处理（2/2）

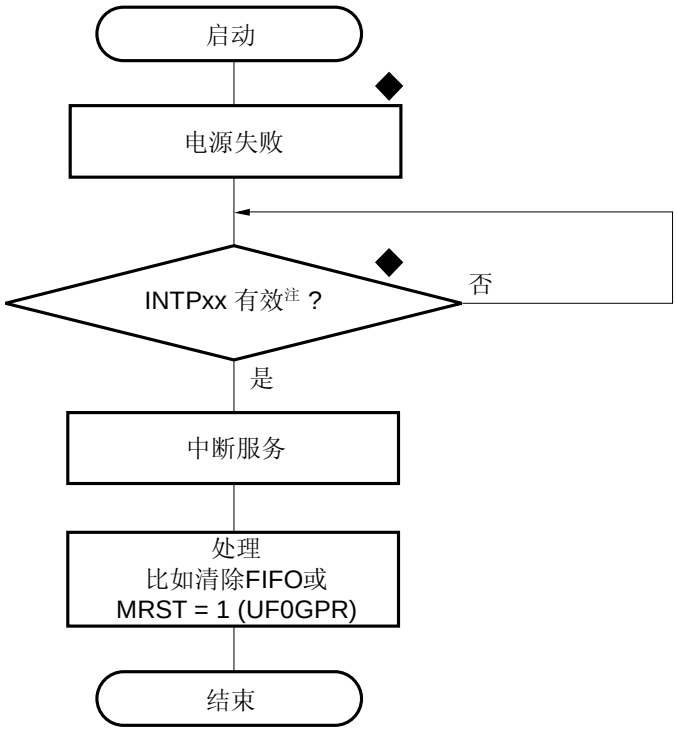


备注 **1.** m = 0 至 4

2. ♦： 由硬件处理

图 20-31. 电源上电/电源失败后的处理示例（3/3）

（b）电源失败的处理



注 INTPxx 表示 V850ES/JG3-L 的一个外部中断引脚（INTP0 至 INTP7）。
为下列应用分配一个外部中断引脚。

- 在自供电模式（UF0DSTL 寄存器的 SFPW 位 = 1）下检测连接器的断开。在这种情况下，监视 USB 连接器的 VDD 线，并且在边沿处将结果输入到外部中断引脚。注意中断输入引脚的噪声消除时间。
- 当设备与 HUB 芯片安装在同一块板上时，由 HUB 芯片检测电源的关闭。

备注 ♦： 由硬件处理

20.9.6 在DMA模式下为批量传输（OUT）接收数据

批量传输 (OUT) 被分配给端点 2 和端点 4。下面的流程图展示了在使用 DMA 时如何控制端点 2。端点 4 可以按照同样的顺序来控制。要使用该流程图作为端点 4 的控制流程，因此，在流程图中读取端点 2 的位名称，正如端点 4 的那些位。下面的控制流程图展示如何通过 CPU 保持读取数据。

如果通过设置 UF0IDR 寄存器的 DQBO1MS 位为 1，接收到批量传输 (OUT)的数据，对端点 2 的 DMA 请求变为有效，而不是产生中断请求(INTUSBF0)。根据 UF0IDR 寄存器的 MODEn(n = 0, 1)位的设置，决定这个对端点 2 的 DMA 请求的操作。如果存储在 UF0BO1 寄存器中的所有数据已经被 DMA 读取，对端点 2 的 DMA 请求变为无效。在这种状态下，如果正确接收到下一个批量传输 (OUT)的数据，对端点 2 的 DMA 请求重新变为有效。如果接收到的批量传输 (OUT)的数据长度等于或小于 FIFO 大小，只要 DMA 读取数据结束，则立即发布一个 Short 中断请求且 INTUSBF0 (EP2_ENDINT)信号变为有效。要使用 DMA 再次读取数据，再次设置 DQBO1MS 位为 1。如果因为对端点 2 的 DMA 结束信号完成 DMA 操作，UF0IDR 寄存器的 DQBO1MS 位被清除为 0，且对端点 2 的 DMA 请求变为无效，同时发出 DMA_END 中断请求。如果此时 UF0BO1 寄存器中还保持有数据，再次设置 UF0IDR 寄存器的 DQBO1MS 位，来重新启动 DMA。但是，批量传输 (OUT)的数据长度总是等于或小于 FIFO 大小。于是，只要 DMA 读取数据，立即发布一个 Short 中断请求且 INTUSBF0 (EP2_ENDINT)信号变为有效，DQBO1MS 位被清除，对端点 2 的 DMA 请求变为无效。

注意事项 1. 只要还有数据需要传输，在需求模式下(UF0IDR 寄存器的 MODE1 位和 MODE0 位 = 10)，对端点 n (n = 2, 4) 的 DMA 请求信号就变为有效。

2. DMA 传输为批量传输(OUT)的数据是短包(63 字节或更少)时，在传输结束后，清除 UF0IC0.SHORTC 和 UF0IS0.SHORT 位。
如果 SHORT 位没有被清除，DMASTOP_EpnB 信号生效，且不执行下一次 DMA 传输操作。

(1) 批量传输 (OUT: EP2, EP4)的初始化设置

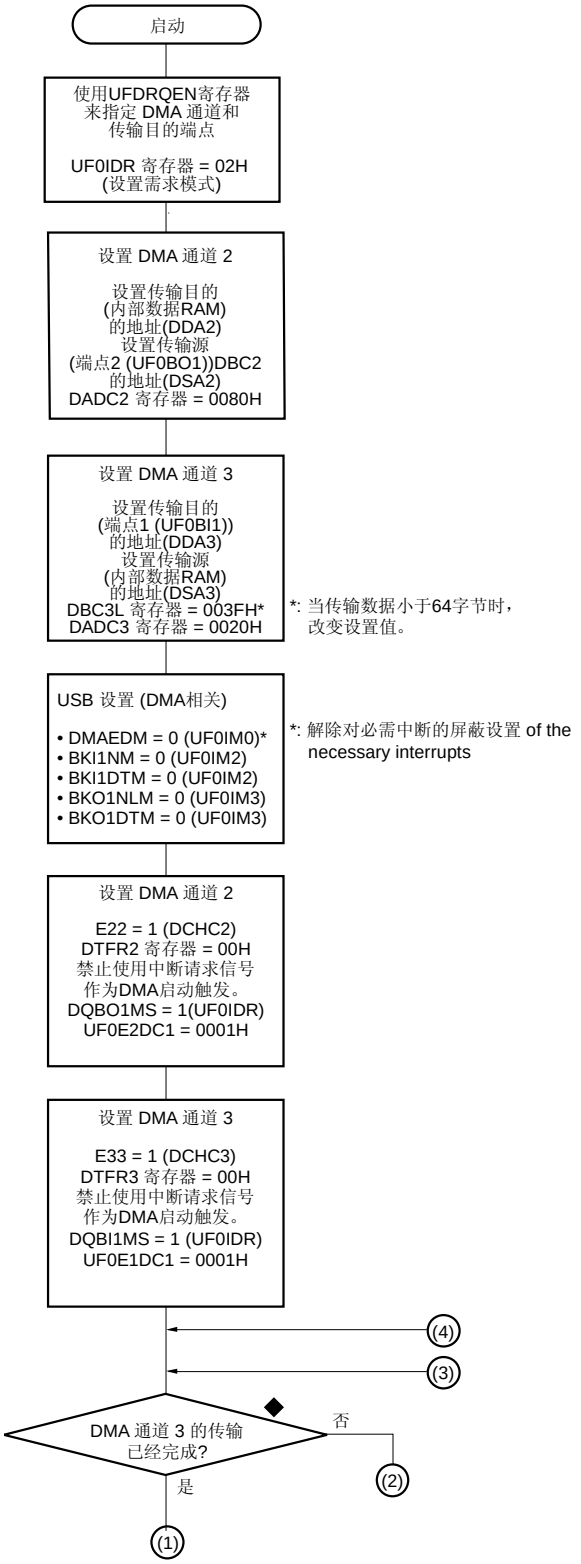
(a) DMAC 的初始化设置

- DSAn 寄存器（n = 0 至 3）被设置为 00210000H (EP2)或 00220000H (EP4)。
- DADCn 寄存器（n = 0 至 3）被设置为 0080H。
  (8 位传输，传输源地址：固定，传输目的地址：递增)
- DTFRn 寄存器（n = 0 至 3）被设置为 0000H。
- UFDRQEN 寄存器根据使用的 DMA 通道来设置。
  (详情参见 20.6.10 (1) USBF DMA 请求使能寄存器 (UFDRQEN)。)

(b) EPC 的初始化设置

- UF0IDR 寄存器被设置为 12H (EP2) 或 22H (EP4) (需求模式)。
- UF0IM0.DMAEDM 位 = 0
- UF0IM3.BKO1NLM 位 = 0 (EP2)
- UF0IM3.BKO1DTM 位 = 0 (EP2)
- UF0IM3.BKO2NLM 位 = 0 (EP4)
- UF0IM3.BKO2DTM 位 = 0 (EP4)

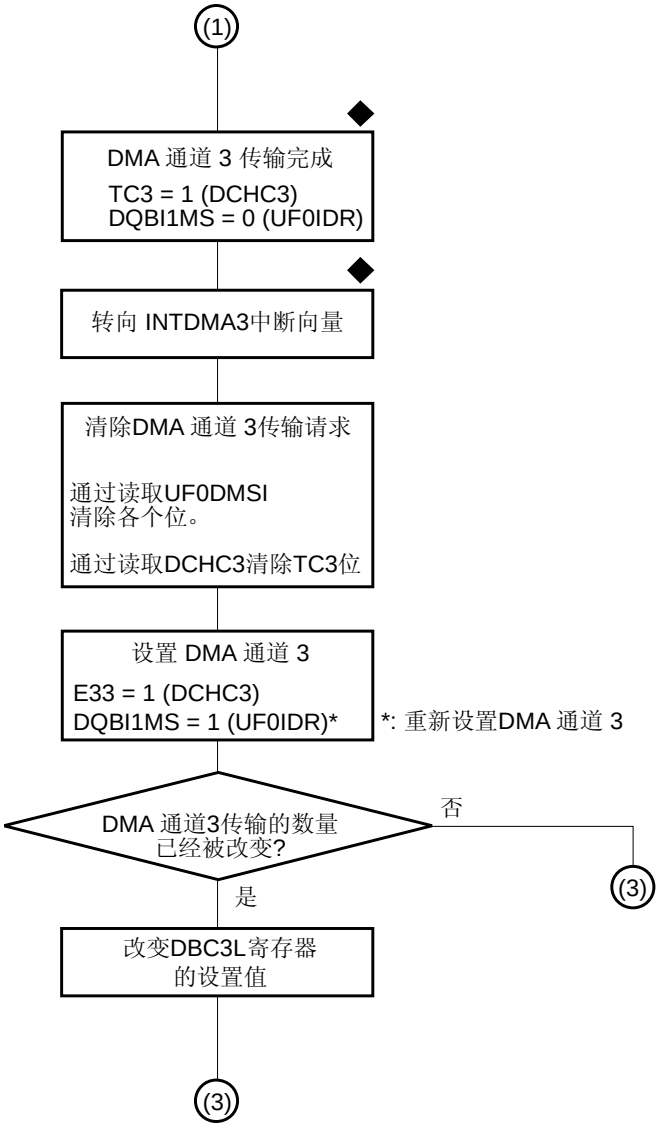
图 20-32. 通过批量传输（OUT）进行 DMA 处理(1/3)



备注 **1.** 上述流程图展示的是通过 DMA 通道 2 从端点 2 到内部数据 RAM 的传输，通过 DMA 通道 3 的传输是从内部数据 RAM 到端点 1。

2. ♦：由硬件处理

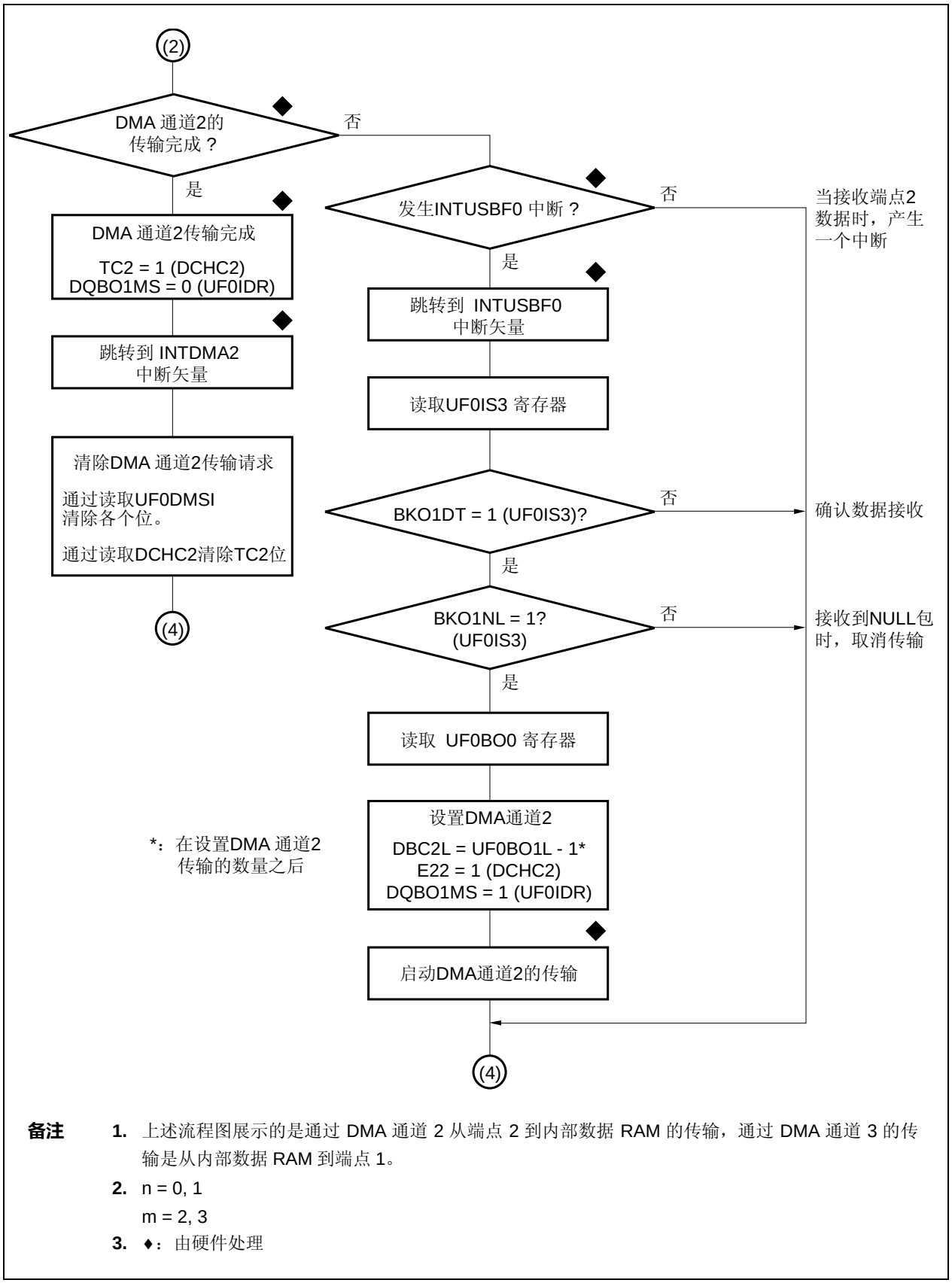
图 20-32. 通过批量传输（OUT）进行 DMA 处理(2/3)



备注

1. 上述流程图展示的是通过 DMA 通道 2 从端点 2 到内部数据 RAM 的传输，通过 DMA 通道 3 的传输是从内部数据 RAM 到端点 1。
2. n = 0, 1
3. ♦：由硬件处理

图 20-32. 通过批量传输（OUT）进行 DMA 处理(3/3)



备注

1. 上述流程图展示的是通过 DMA 通道 2 从端点 2 到内部数据 RAM 的传输，通过 DMA 通道 3 的传输是从内部数据 RAM 到端点 1。
2. n = 0, 1
   m = 2, 3
3. ♦：由硬件处理

20.9.7 在DMA模式下为批量传输（IN）发送数据

批量传输 (IN) 被分配给端点 1 和端点 3。下面的流程图展示了在使用 DMA 时如何控制端点 1。端点 3 可以按照同样的顺序来控制。要使用该流程图作为端点 3 的控制流程，因此，在流程图中读取端点 1 的位名称，正如端点 3 的那些位。

如果通过设置 UF0IDR 寄存器的 DQBI1MS 位为 1，可以写入批量传输 (IN)所需的数据，对端点 1 的 DMA 请求变为有效，而不是产生中断请求(INTUSBF0)。根据 UF0IDR 寄存器的 MODEn(n = 0, 1)位的设置，决定这个对端点 1 的 DMA 请求的操作。如果所有可以写入 UF0BO1 寄存器的数据已经被 DMA 写入，对端点 1 的 DMA 请求变为无效。在这种状态下，进行 FIFO 的切换操作，如果可以写入批量传输 (IN)所需的数据，对端点 1 的 DMA 请求重新变为有效。如果即使因为 DMA 传输导致 FIFO 占满，却没有执行 FIFO 的自动切换操作，除非 UF0DEND 寄存器的 BKI1T 位被置为 1。因此，请确保设置 UF0DEND 寄存器的 BKI1DED 位为 1 来传输数据。如果因为对端点 1 的 DMA 结束信号完成 DMA 操作，UF0IDR 寄存器的 DQBI1MS 位被清除为 0，且对端点 1 的 DMA 请求变为无效，同时发出 DMA_END 中断请求。要在 FIFO 未满时发送一个短包，设置 UF0DEND 寄存器的 BKI1DED 位为 1。

注意事项 **只要还有数据需要传输，在需求模式下(UF0IDR 寄存器的 MODE1 位和 MODE0 位 = 10)，对端点 n (n = 1, 3) 的 DMA 请求信号就变为有效。**

(1) 批量传输 (IN: EP1, EP3)的初始化设置

(a) DMAC 的初始化设置

- DSAn 寄存器（n = 0 至 3）被设置为 00201000H (EP1)或 00202000H (EP3)。
- DADCn 寄存器（n = 0 至 3）被设置为 0020H。
  (8 位传输，传输源地址：递增，传输目的地址：固定)
- DTFRn 寄存器（n = 0 至 3）被设置为 0000H。
- UFDRQEN 寄存器根据使用的 DMA 通道来设置。
  (详情参见 20.6.10 (1) USBF DMA 请求使能寄存器 (UFDRQEN)。)

(b) EPC 的初始化设置

- UF0IDR 寄存器被设置为 42H (EP1) 或 82H (EP3) (需求模式)。
- UF0IM0.DMAEDM 位 = 0
- UF0IM2.BKI1NLM 位 = 0 (EP1)
- UF0IM2.BKI1DTM 位 = 0 (EP1)
- UF0IM2.BKI2NLM 位 = 0 (EP3)
- UF0IM2.BKI2DTM 位 = 0 (EP3)

图 20-33. 通过批量传输（IN）进行 DMA 处理 (1/4)

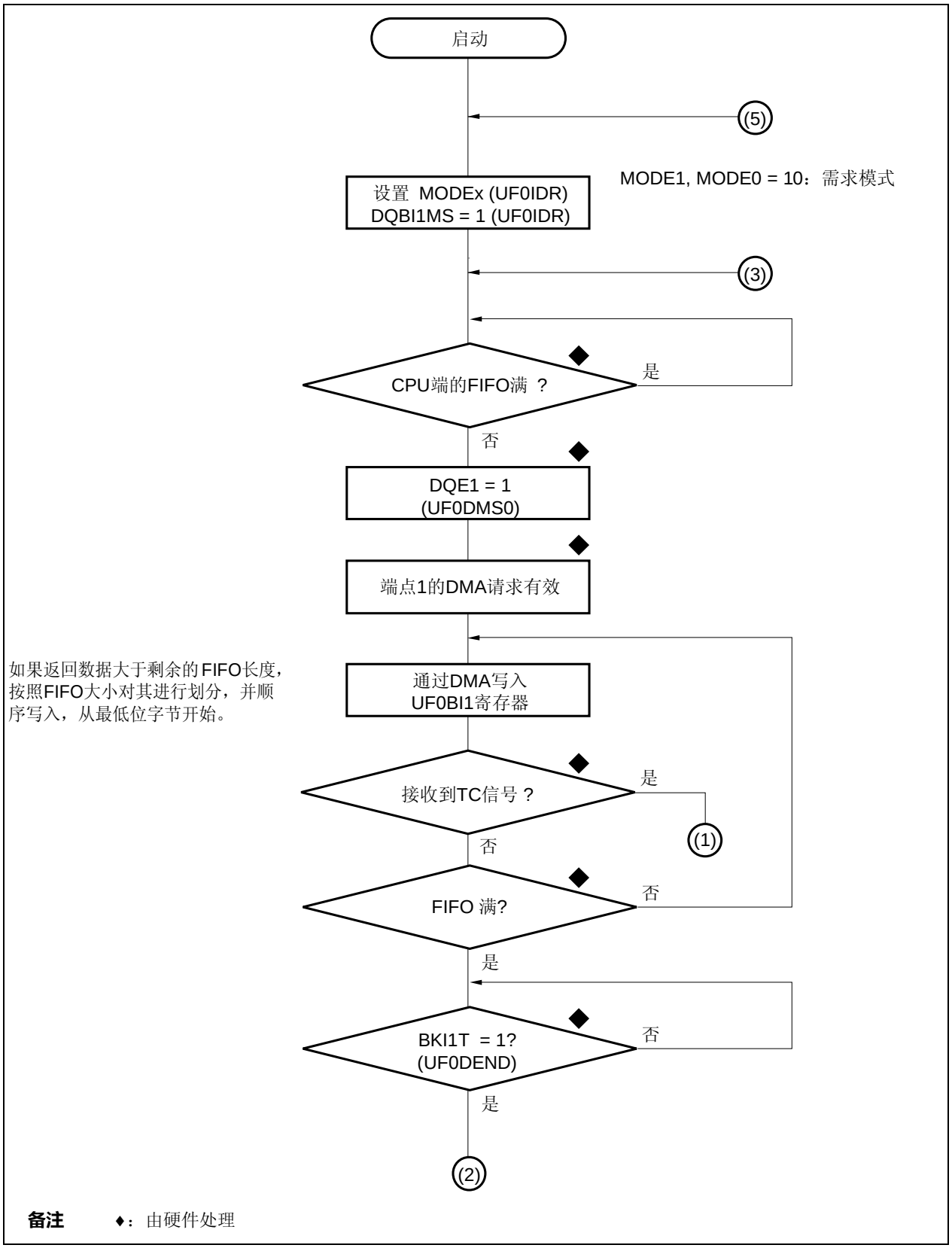
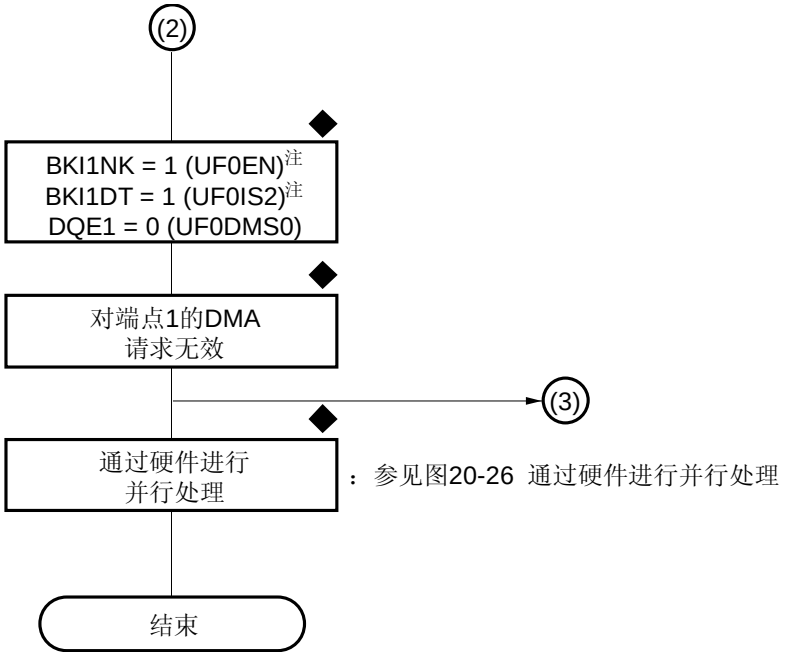


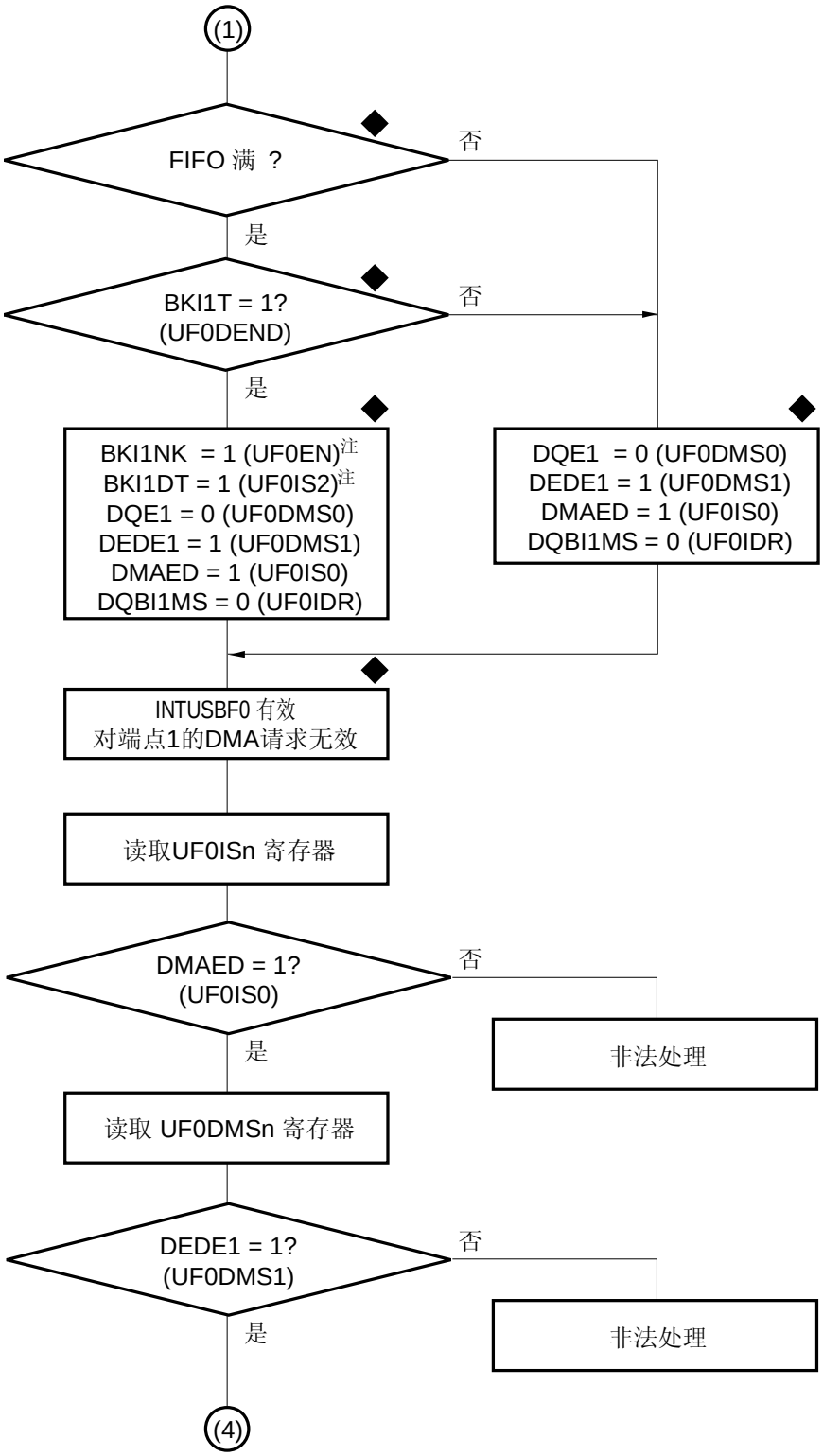
图 20-33. 通过批量传输（IN）进行 DMA 处理(2/4)



注 位值改变的时序取决于 SIE 端的状态。

备注 ♦：由硬件处理

图 20-33. 通过批量传输（IN）进行 DMA 处理(3/4)

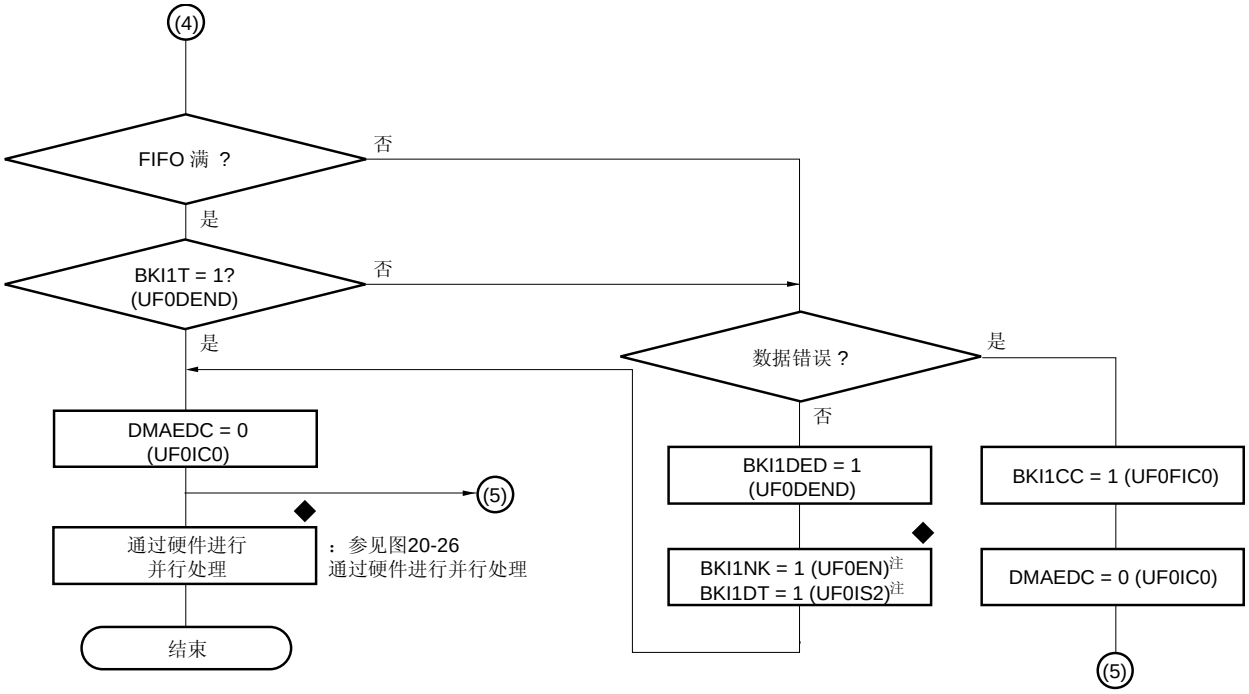


注 位值改变的时序取决于 SIE 端的状态。

备注 **1.** n = 0, 1

2. ♦：由硬件处理

图 20-33. 通过批量传输（IN）进行 DMA 处理(4/4)



注 位值改变的时序取决于 SIE 端的状态。

备注 ♦：由硬件处理

第二十一章 DMA功能（DMA 控制器）

V850ES/JG3-L 单片机包括一个直接存储器访问（DMA）控制器（DMAC），用于执行和控制 DMA 传输。

根据片上外设 I/O（串行接口、定时器/计数器和 A/D 转换器），外部输入引脚的中断或软件触发发出的 DMA 请求，DMAC 控制存储器与 I/O 之间、存储器之间或 I/O 之间的数据传输。（存储器是指内部 RAM 或外部存储器）

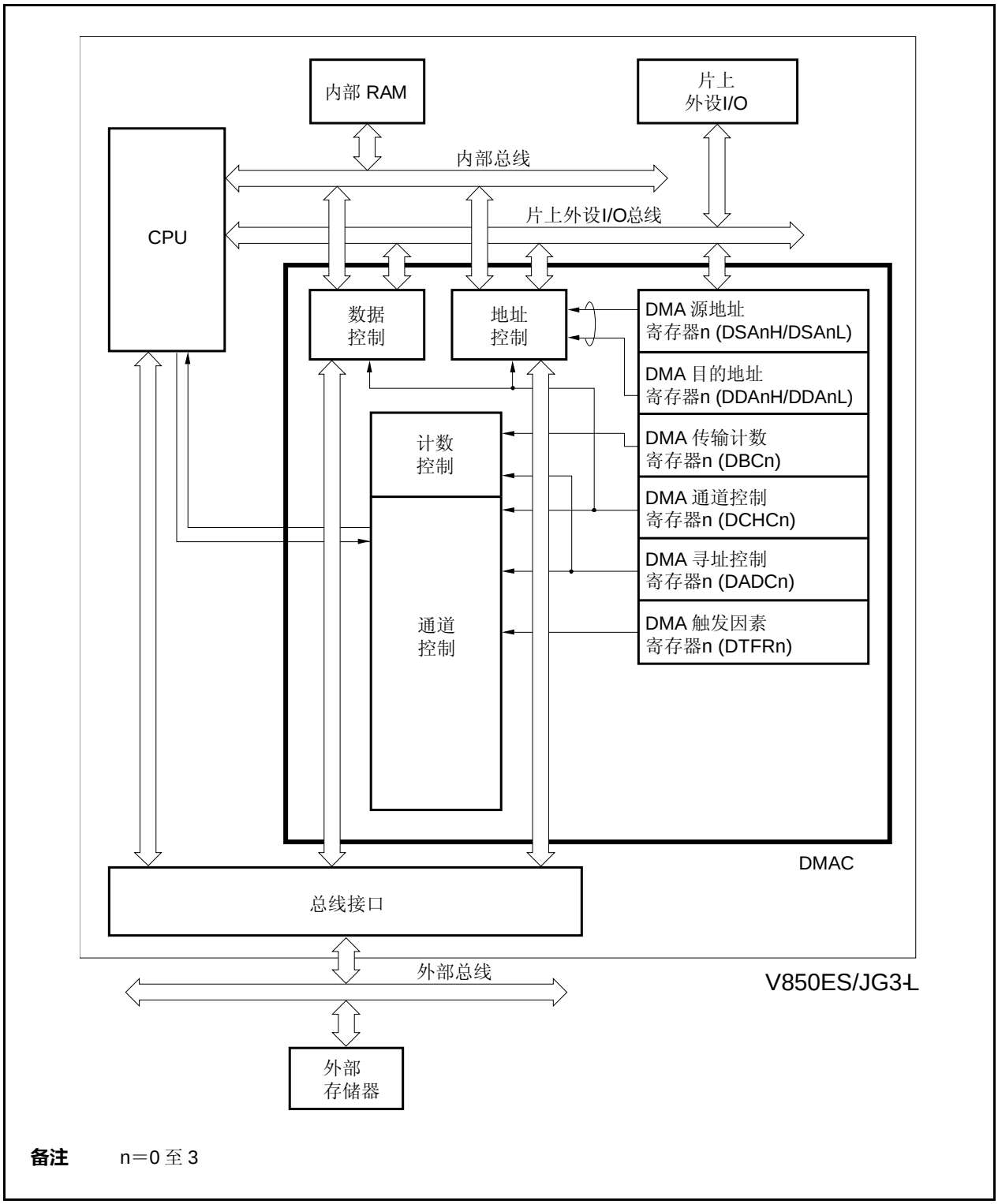
21.1 特性

- 4 路独立 DMA 通道
- 传输单位：8/16 位
- 最大传输计数：65,536（2^{16}）
- 在 DMA 传输期间，使用内部 ROM 执行程序
- 传输类型：双周期传输
  - 不同宽度的总线之间的数据传输
- 传输模式：单次传输模式
- 传输请求：
  - 来自片上外设 I/O （串行接口、定时器/计数器和 A/D 转换器）或来自外部输入引脚中断的请求。
  - 通过软件触发的请求。
- 传输源和传输目的：
  - 内部 RAM ↔ 片上外设 I/O
  - 片上外设 I/O ↔ 片上外设 I/O
  - 内部 RAM ↔外部存储器
  - 外部存储器 ↔ 片上外设 I/O
  - 外部存储器 ↔ 外部存储器

21.2 配置

DMAC 的框图如下所示。

图 21-1. DMAC 的结构框图



备注 n＝0 至 3

DMAC 包括下列硬件。

表 21-1. DMAC 的配置

项目	配置
寄存器	DMA 源地址寄存器 0 至 3（DSA0 至 DSA3） DMA 目的地址寄存器 0 至 3（DDA0 至 DDA3） DMA 传输计数寄存器 0 至 3（DBC0 至 DBC3） DMA 寻址控制寄存器 0 至 3（DADC0 至 DADC3） DMA 通道控制寄存器 0 至 3（DCHC0 至 DCHC3） DMA 触发因素寄存器 0 至 3（DTFR0 至 DTFR3）

21.3 寄存器

（1）DMA 源地址寄存器 0 至 3（DSA0 至 DSA3）

DSA0 至 DSA3 寄存器用于为 DMA 通道 n 设置 DMA 源地址（每个 26 位）（n＝0 至 3）。
这些寄存器被分为两个 16 位寄存器，即 DSAnH 和 DSAnL。
这些寄存器可以按 16 位宽度读取或写入。

复位后: 不确定　R/W　地址:　DSA0H FFFFF082H, DSA1H FFFFF08AH,
DSA2H FFFFF092H, DSA3H FFFFF09AH,
DSA0L FFFFF080H, DSA1L FFFFF088H,
DSA2L FFFFF090H, DSA3L FFFFF098H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSAnH (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
DSAnL (n = 0 至 3)	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0

IR	DMA传送源分类
0	外部存储或片上外设I/O
1	内部 RAM

SA25 至 SA16	设定DMA传送源地址 (A25 至 A16) (缺省值未定义)。 DMA 传送时，保持下一个DMA传送源地址。 当DMA传送完成后，保持最初设定的DMA地址。

SA15 至 SA0	设定DMA传送源地址 (A15 至 A0) (缺省值未定义)。 DMA 传送时，保持下一个DMA传送源地址。 当DMA传送完成后，保持最初设定的DMA地址。

注意事项
1. **请确保将 DSAnH 寄存器的第 14 位至第 10 位清为“0”。**
2. **在下列 DMA 传输禁止（DCHCn.Enn 位 = 0）期间，设置 DSAnH 和 DSAnL 寄存器。**
   - **复位后到第一次 DMA 传输启动期间**
   - **通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间**
   - **DMA 传输完成（DCHCn.TCn 位 = 1）后到下一次 DMA 传输启动期间**
3. **当 DSAn 寄存器的值被读取时，两个 16 位寄存器，即 DSAnH 和 DSAnL 都被读出。如果读取与更新产生冲突，则可能是对正在被更新的值进行读取（请参见 21.13 “注意事项”）。**
4. **未按照 16 位对齐的数据的 DMA 传输，不能支持。**
   如果奇地址被指定作为传输源，地址的最低有效位被强制处理为 0。
5. **系统复位后，在启动 DMA 传输之前，设置 DSAnH，DSAnL，DDAnH，DDAnL 和 DBCn 寄存器。如果没有设置这些寄存器，DMA 传输操作无法保证。**

（2）DMA 目的地址寄存器 0 至 3（DDA0 至 DDA3）

DDA0 至 DDA3 寄存器用于为 DMA 通道 n 设置 DMA 目的地址（每个 26 位）（n＝0 至 3）。

这些寄存器被分为两个 16 位寄存器，即 DDAnH 和 DDAnL。

这些寄存器可以按 16 位宽度读取或写入。

复位后: 不确定　R/W　地址: DDA0H FFFFF086H, DDA1H FFFFF08EH, DDA2H FFFFF096H, DDA3H FFFFF09H
DDA0L FFFFF084H, DDA1L FFFFF08CH, DDA2L FFFFF094H, DDA3L FFFFF09CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDAnH (n = 0 至 3)	IR	0	0	0	0	0	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16
DDAnL (n = 0 至 3)	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

IR	DMA传送目的分类
0	外部存储器或片上外设 I/O
1	内部 RAM

DA25 至 DA16	设定一个DMA传送目的地 (A25 至 A16) (缺省值不确定)。 DMA传送期间，保持下一个DMA传送目的地址。 当DMA传送完成时，保持最初设定的DMA地址。

DA15 至 DA0	设定一个DMA传送目的地 (A15 至 A0) (缺省值不确定)。 DMA传送期间，保持下一个DMA传送目的地址。 当DMA传送完成时，保持最初设定的DMA地址。

注意事项

1. **请确保将 DSAnH 寄存器的第 14 位至第 10 位清为“0”。**
2. **在下列 DMA 传输禁止（DCHCn.Enn 位＝0）期间，设置 DDAnH 和 DDAnL 寄存器。**
   - **复位后到第一次 DMA 传输启动期间。**
   - **通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间。**
   - **DMA 传输完成（DCHCn.TCn 位＝1）后到下一次 DMA 传输启动期间。**
3. **当 DDAn 寄存器的值被读取时，两个 16 位寄存器，即 DDAnH 和 DDAnL 被读出。如果读取与更新产生冲突，则可能是对正在被更新的值进行读取（请参见 21.13 “注意事项”）。**
4. **未按照 16 位对齐的数据的 DMA 传输，不能支持。**
   如果奇地址被指定作为传输源，地址的最低有效位被强制处理为 0。
5. **系统复位后，在开始 DMA 发送之前，设置 DSAnH，DSAnL，DDAnH，DDAnL 和 DBCn 寄存器。如果没有设置这些寄存器，DMA 传输操作无法保证。**

（3）DMA 传输计数寄存器 0 至 3（DBC0 至 DBC3）

DBC0 至 DBC3 寄存器是 16 位寄存器，用于设置 DMA 通道 n 的字节传输计数（n＝0 至 3）。

在 DMA 传输过程中，这些寄存器保存剩余的传输计数。

无论传输数据的单位（8/16 位），每传输一次，这些寄存器就递减 1，当发生借位时，传输终止。

当 DMA 传输完成时，最初指定的传输数量被保持。

这些寄存器可以按 16 位宽度读取或写入。

复位后: 不确定　　R/W　　地址: DBC0 FFFFF0C0H，DBC1 FFFFF0C2H，
DBC2 FFFFF0C4H，DBC3 FFFFF0C6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBCn (n = 0 至 3)	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0

BC15至BC0	DMA传输期间的 传输计数设置或保持传输计数
0000H	传输计数1次，或保持传输计数
0001H	传输计数2次，或保持传输计数
:	:
FFFFH	传输计数 65,536 (2^{16}) 次，或保持传输计数

注意事项 1. 在下列 DMA 传输禁止（DCHCn.Enn 位 = 0）期间，设置 DBCn 寄存器。

- **从复位后到第一次 DMA 传输启动期间**
- **从通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间**
- **从 DMA 传输完成后（DCHCn.TCn 位 = 1）到下一次 DMA 传输启动期间**

2. 系统复位后，在开始 DMA 发送之前，设置 DSAnH，DSAnL，DDAnH，DDAnL 和 DBCn 寄存器。如果没有设置这些寄存器，DMA 传输操作无法保证。

（4）DMA 寻址控制寄存器 0 至 3（DADC0 至 DADC3）

DADC0 至 DADC3 寄存器是 16 位寄存器，用于控制 DMA 通道 n 的 DMA 传输模式（n＝0 至 3）。

这些寄存器可以按 16 位宽度读取或写入。

系统复位后，这些寄存器被设置为 0000H。

复位后: 0000H　　R/W　　地址: DADC0 FFFFF0D0H, DADC1 FFFFF0D2H,
DADC2 FFFFF0D4H, DADC3 FFFFF0D6H

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	发送数据宽度的设置
0	8 位
1	16 位

SAD1	SAD0	发送源地址计数方向的设置
0	0	增加
0	1	减少
1	0	固定
1	1	禁止设置

DAD1	DAD0	目的地址计数方向的设置
0	0	增加
0	1	减少
1	0	固定
1	1	禁止设置

注意事项

1. **请确保将 DADCn 寄存器的第 15 位、第 13 位至第 8 位，以及第 3 位至第 0 位清为“0”。**
2. **在下列 DMA 传输禁止（DCHCn.Enn 位 = 0）期间，设置 DADCn 寄存器。**
   - **从复位后到第一次 DMA 传输启动期间。**
   - **从通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间。**
   - **从 DMA 传输完成后（DCHCn.TCn 位 = 1）到下一次 DMA 传输启动期间。**
3. **DSn0 位指定传输数据的宽度，但是无法控制总线的宽度。关于外部总线宽度的详情，参见 5.5.2 (1) 总线宽度配置寄存器 (BSC)。**
4. **如果传输数据宽度被设置为 16 位（DS0 位 = 1），则不能从奇地址开始传输。总是从低位地址的第一位为 0 的地址启动传输。**
5. **如果 DMA 传输是在一个片上外设 I/O 寄存器（作为传输源或目的地）上执行的，请确保指定的传输宽度与寄存器宽度相同。例如，要在一个 8 位寄存器上执行 DMA 传输，务必指定 8 位宽度传输方式。**

（5）DMA 通道控制寄存器 0 至 3（DCHC0 至 DCHC3）

DCHC0 至 DCHC3 寄存器是 8 位寄存器，用于控制 DMA 通道 n 的 DMA 传输操作模式。

该寄存器可进行字节读写或按位读写。（不过，第 7 位是只读的，而第 1 位和第 2 位是只写的。如果读取第 1 位或第 2 位，则读出的值将始终是 0。）

系统复位后，该寄存器被设为 00H。

复位后: 00H　R/W　地址: DCHC0 FFFFF0E0H, DCHC1 FFFFF0E2H,
DCHC2 FFFFF0E4H, DCHC3 FFFFF0E6H

	<7>	6	5	4	3	<2>	<1>	<0>
DCHCn (n = 0至3)	TCn注1	0	0	0	0	INITn注2	STGn注2	Enn

TCn注1	状态标志指示DMA通道n的DMA传输是否完成
0	DMA传输未完成。
1	DMA传输已完成。
DMA传输的最后设定为1，当读取时清为0。	

INITn注2	DMA传输禁止(Enn位= 0)时，如果INITn位设定为1， 可以初始化DMA传输状态。

STGn注2	这是DMA传输的软件启动触发信号。 如果在DMA传输允许状态(TCn 位 = 0, Enn 位 = 1)时，该位被置为1， DMA传输启动。

Enn	设置是否允许DMA通道n的DMA传输
0	DMA传输禁止
1	DMA传输使能
当Enn位被设置为1时，DMA传输使能。 当DMA传输完成时(产生一个终端计数)，该位自动清为0。 通过软件将Enn位清为0, 可以中止DMA传输。如要恢复DMA传输，则将Enn位再次设置为1。 当中止或恢复DMA传输时，严格遵守 21.13（5）暂时停止DMA传输的步骤中的描述。	

注
1. TCn 位只读。
2. INITn 和 STGn 位只写。

注意事项
1. **请确保将 DCHCn 寄存器的第 6 位至第 3 位清为“0”。**
2. **DMA 传输完成后（当产生末尾计数时），Enn 位被清零，然后 TCn 位被置 1。如果正在更新 DCHCn 寄存器同时发生读取操作，可能会读出一个值表示“传输未完成且传输被禁止”（TCn 位 = 0 且 Enn 位 = 0）。**

（6）DMA 触发因素寄存器 0 至 3（DTFR0 至 DTFR3）

DTFR0 至 DTFR3 寄存器是 8 位寄存器，控制片上外设 I/O 的中断请求信号产生的 DMA 传输启动触发源。
这些寄存器设置的中断请求信号用作 DMA 传输启动因素。
这些寄存器可以按字节读取或写入。但是，DFn 位可以按位进行读写。
系统复位后，该寄存器被设为 00H。

复位后: 00H　R/W　地址: DTFR0 FFFFF810H, DTFR1 FFFFF812H,
DTFR2 FFFFF814H, DTFR3 FFFFF816H

	<7>	6	5	4	3	2	1	0
DTFRn	DFn	0	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0

(n = 0 至 3)

DFn[注]	DMA 传送请求状态标志
0	无DMA 传送请求
1	DMA 传送请求

注　不要通过软件向 DFn 位写入 1。如果在 DMA 传输禁止时发生了一个被指定为启动 DMA 传输的触发因素，则将 0 写入该位，以清除 DMA 传输请求。

注意事项 1. 在 DMA 传输禁止（DCHCn.Enn 位 = 0）期间，设置 IFCn5 至 IFCn0 位。
- **从复位后到第一次 DMA 传输启动期间。**
- **从通道被 DCHCn.INITn 位初始化后到 DMA 传输启动期间。**
- **从 DMA 传输完成后（DCHCn.TCn 位 = 1）到下一次 DMA 传输启动期间。**

2. 改变 DTFRn 寄存器设置时，请确保按照下列步骤进行 (n = 0 至 3，m = 0 至 3，n ≠ m)。
<1> 停止要重写的通道的 DMAn 操作(DCHCn.Enn 位 = 0)。
<2> 改变 DTFRn 寄存器设置（请确保设置 DFn 位 = 0 并用 8 位操作指令改变设置）。
<3> 确认 DFn 位 = 0（事先停止中断产生源的操作）。
<4> 使能 DMAn 操作 (Enn 位 = 1)。

3. 在待机模式（IDEL1、IDLE2、STOP 或 sub-IDLE 模式）中产生的中断请求信号不能启动 DMA 传输周期（DFn 位也不会置 1）。

4. 通过 IFCn5 至 IFCn0 位选定 DMA 启动因素，当选定的片上外设 I/O 产生中断时，不论 DMA 发送操作是允许还是禁止，DFn 位都会置 1。如果这种状态下 DMA 是使能的，则 DMA 传输立即启动。

备注　关于 IFCn5 至 IFCn0 位，请参见**表 21-2 DMA 启动因素。**

表 21-2. DMA 启动因素(1/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	中断源
0	0	0	0	0	0	禁止由中断产生 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/INTUA5T
0	1	1	0	0	1	INTTP4CC0
0	1	1	0	1	0	INTTP4CC1
0	1	1	0	1	1	INTTP5CC0
0	1	1	1	0	0	INTTP5CC1
0	1	1	1	0	1	INTTM0EQ0
0	1	1	1	1	0	INTCB0R/INTIIC1
0	1	1	1	1	1	INTCB0T
1	0	0	0	0	0	INTCB1R
1	0	0	0	0	1	INTCB1T
1	0	0	0	1	0	INTCB2R
1	0	0	0	1	1	INTCB2T
1	0	0	1	0	0	INTCB3R
1	0	0	1	0	1	INTCB3T
1	0	0	1	1	0	INTUA0R/INTCB4R
1	0	0	1	1	1	INTUA0T/INTCB4T
1	0	1	0	0	0	INTUA1R/INTIIC2
1	0	1	0	0	1	INTUA1T
1	0	1	0	1	0	INTUA2R/INTIIC0
1	0	1	0	1	1	INTUA2T
1	0	1	1	0	0	INTAD

备注 n＝0 至 3

表 21-2. DMA 启动因素(2/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	中断源
1	0	1	1	0	1	INTKR
1	0	1	1	1	0	INTRTC1
1	0	1	1	1	1	INTUA3R
1	1	0	0	0	0	INTUA3T
1	1	0	0	0	1	INTUA4R
1	1	0	0	1	0	INTUA4T
1	1	0	0	1	1	INTUA5R
1	1	0	1	0	0	INTUC0R
1	1	0	1	0	1	INTUC0T
其它						禁止设置

备注 n＝0 至 3

21.4 传输源和目的

表 21-3 说明了传输源和目的之间的关系（√：允许传输，×：禁止传输）。

表 21-3 传输源和目的之间的关系

		传输目的地			
		内部 ROM	片上外设 I/O	内部 RAM	外部存储器
传输源	片上外设 I/O	×	√	√	√
	内部 RAM	×	√	×	√
	外部存储器	×	√	√	√
	内部 ROM	×	×	×	×

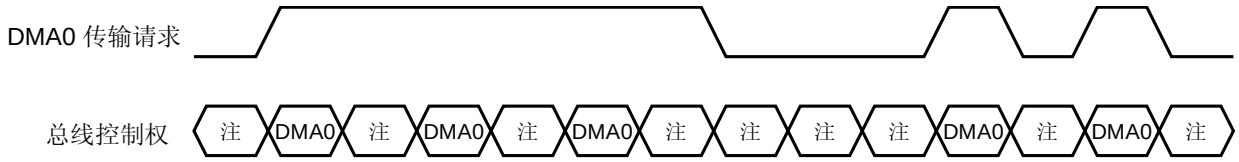
注意事项 在表 21-3 中，用“×”标记的传输目的地和传输源组合，其操作将无法保证。

21.5 传输模式

支持单次传输模式。

在单次传输模式下，在每次字节/半字传输时总线都被释放。如果有后续的 DMA 传输请求，则再执行一次传输。该操作一直进行下去，直到发生一个终止计数。

图 21-2. 单次传输（仅使用一个通道）



当 DMA0 传输请求被响应时，执行 DMA 传输，且总线控制权被释放给 CPU。只要有 DMA0 传输请求，该操作就会重复继续，直到 TC0 位被设置为 1（DMA 传输完成）。

注 CPU 正在使用总线，或总线未被使用。

21.6 传输类型

作为传输模式，可以支持双周期传输。

在双周期传输中，数据传输在两个周期中完成，即读周期和写周期。

在读周期中，输出传输源地址，并执行从源地址向 DMAC 的数据读取操作。在写周期中，输出传输目的地址，并执行从 DMAC 向目的地的数据写入操作。

在读周期和写周期之间总是插入一个时钟的空闲周期。在进行双周期的 DMA 传输时，如果传输源与目的地的数据总线宽度不同，则执行以下操作。

<16 位数据传输 (DADCn.DSn0 位 = 1)>

<1> 从 32 位总线向 16 位总线传输
先产生一个读周期（高 16 位或低 16 位数据），接着产生一个写周期（16 位）。

<2> 从 16/32 位总线向 8 位总线传输
产生一次 16 位读周期，然后产生两次 8 位写周期。

<3> 从 8 位总线向 16/32 位总线传输
先产生两次 8 位读周期，然后产生一次 16 位写周期。

<4> 从 16 位总线向 32 位总线传输
产生一次 16 位读周期，然后产生一次 16 位写周期。

对于在片上外设 I/O 寄存器（传输源或目的地）上执行的 DMA 传输，请确保指定的传输宽度与寄存器宽度相同。例如，要在一个 8 位寄存器上执行 DMA 传输，务必指定字节（8 位宽度）传输方式。

备注 每个传输源和传输目的地的总线宽度如下：

- 片上外设 I/O： 16 位总线宽度
- 内部 RAM： 32 位总线宽度
- 外部存储器： 8 位或 16 位总线宽度

21.7 DMA通道优先级

DMA 通道优先级固定如下所示：

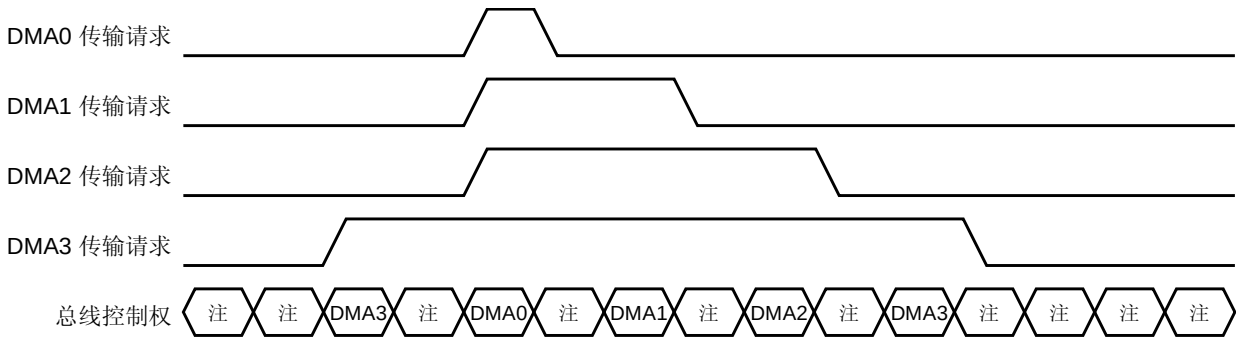
DMA 通道 0 > DMA 通道 1 > DMA 通道 2 > DMA 通道 3

当 DMAC 已释放总线时，如果有另一个具有更高优先级的 DMA 传输请求产生，高优先级总是优先处理。

在一个传输周期内，如果产生两个传输请求，对同一个通道的新的传输请求和一个优先级较低的其他通道传输请求，在总线释放给 CPU 之后，执行较低优先级的通道 DMA 传输（在一个传输周期内对同一个通道的新的传输请求被忽略）。

每个传输周期都要检查这些优先级。

图 21-3. 单次传输（使用多个通道）



如果同时响应多个 DMA 传输请求，从较高优先级的传输开始执行。

注 CPU 正在使用总线，或总线未被使用。

21.8 与DMA传输有关的时间

响应 DMA 请求所需的时间、DMA 传输需要的最少时钟数如下所示：

单次传输： DMA 响应时间（<1>）＋ 传输源存储器访问（<2>）＋1 注1＋传输目的地存储器访问（<2>）

表 21-4. DMA 周期内执行时钟的数量

DMA 周期		执行时钟数量
<1> DMA 请求响应时间		4 时钟（最小值）+ 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注3
	内部 RAM 访问	2 时钟
	外设 I/O 寄存器访问	3 时钟 + VSWC 寄存器设定的等待周期数注4

注
1. DMA 传输时，在读周期和写周期之间总是插入一个时钟。
2. 如果指定一个外部中断（INTPn）作为启动 DMA 传输的触发信号，则增加噪声消除时间（n＝0 至 7）。
3. 使用 8 位总线传输 16 位数据时，必须执行两次传输。
4. 访问专用的外设 I/O 寄存器，可能需要更多等待周期（详细情况，请参见 **3.4.9（1）**）。

21.9 DMA 传输启动因素

DMA 传输启动因素有两种，介绍如下：

（1）软件请求

当 DCHCn.TCn 位为 1 且 DCHCn.Enn 位为 1（允许 DMA 传输）时，如果将 DCHCn.STGn 位设置为 1，则启动 DMA 传输。

若在此之后立刻请求下一个 DMA 传输周期，则应使用 DBCn 寄存器，确认前一个 DMA 传输周期已经完成，并再次将 STGn 位设置为 1（n＝0 至 3）。

TCn 位＝0，Enn 位＝1
↓
STGn 位＝1 …启动第一次 DMA 传输。
↓
确认 DBCn 寄存器的内容已经更新。
STGn 位＝1 …启动第二次 DMA 传输。
↓
：
↓
产生终止计数 … Enn 位＝0，TCn 位＝1 且产生 INTDMAn 信号。

（2）片上外设 I/O 请求

当 DCHCn.TCn 位为 1 且 DCHCn.Enn 位为 1（允许 DMA 传输）时，如果从 DTFRn 寄存器设置的片上外设 I/O 产生一个中断请求，则 DMA 传输启动 (n = 0 至 3)。

注意事项
1. **两个启动因素（软件触发和硬件触发）不能用于一个 DMA 通道。如果对一个 DMA 通道同时产生两个启动因素，则只有其中一个有效。但是，无法确认有效启动因素。**
2. **在前一个 DMA 传输请求产生后且在前一个 DMA 传输完成之前产生的新的 DMA 通道传输请求将被忽略（清除）。**
3. **同一个 DMA 通道的传输请求时间间隔取决于 DMA 传输周期总线等待的设置、其他通道的开始状态或外部总线保持请求也会影响该间隔。特别如注意事项 2 所述，在 DMA 传输周期启动之前或在 DMA 传输期间中，产生的新的同一个通道传输请求将被忽略。因此，同一个 DMA 通道的传输请求，必须由系统确保时间间隔足够长。当使用软件触发时，可以通过更新 DBCn 寄存器来检查之前的 DMA 传输周期完成情况。**

21.10 DMA 中止因素

如果发生总线保持，则 DMA 传输将中止。
如果在内部存储器/片上外设 I/O 和内部存储器//片上外设 I/O 之间传输，同样会发生 DMA 传输中止。
当总线保持被清除，则 DMA 传输将恢复。

21.11 DMA 传输结束

当 DMA 传输已经完成 DBCn 寄存器设置的次数，当 DCHCn.Enn 位被清零、TCn 位被置 1 后，中断控制器（INTC）将会发现一个 DMA 传输结束中断请求信号（INTDMAn）（n＝0 至 3）。

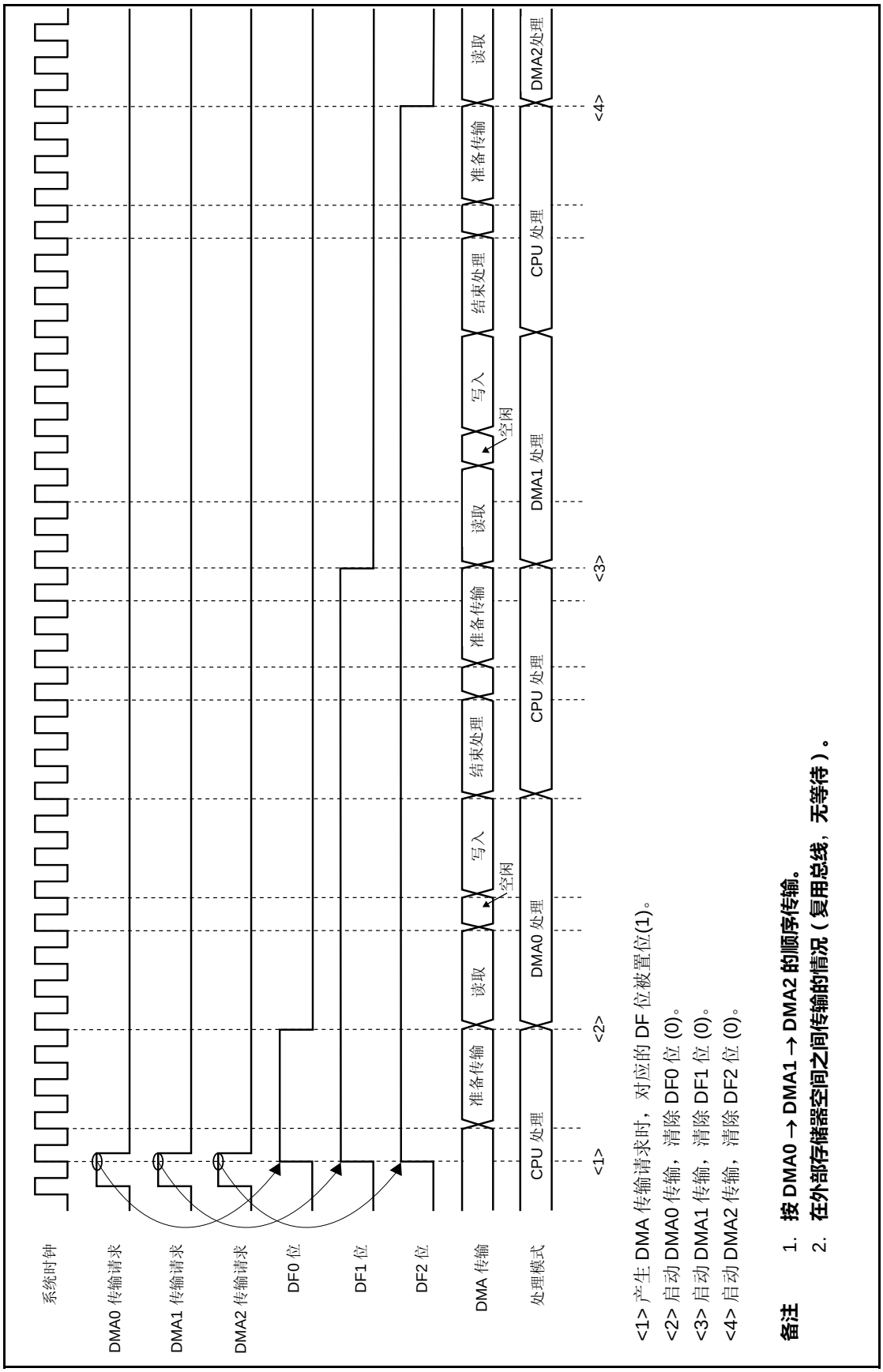
V850ES/JG3-L 单片机不向外部器件输出终止计数信号。因此，通过 DMA 传输结束中断或检测 TCn 位来确认 DMA 传输的完成情况。

21.12 操作时序

DMA 的操作时序如下所示。
下面有四个示例：

- 同时发生多通道请求 DMA 传输 (参见**图 21-4**).
- 在 DMA 传输期间发生一个新的较高优先级 DMA 传输请求 (参见**图 21-5**).
- 对同一个通道的新的 DMA 传输请求被忽略 (单通道) (参见**图 21-6**).
- 对同一个通道的新的 DMA 传输请求被忽略 (多通道) (参见**图 21-7**).

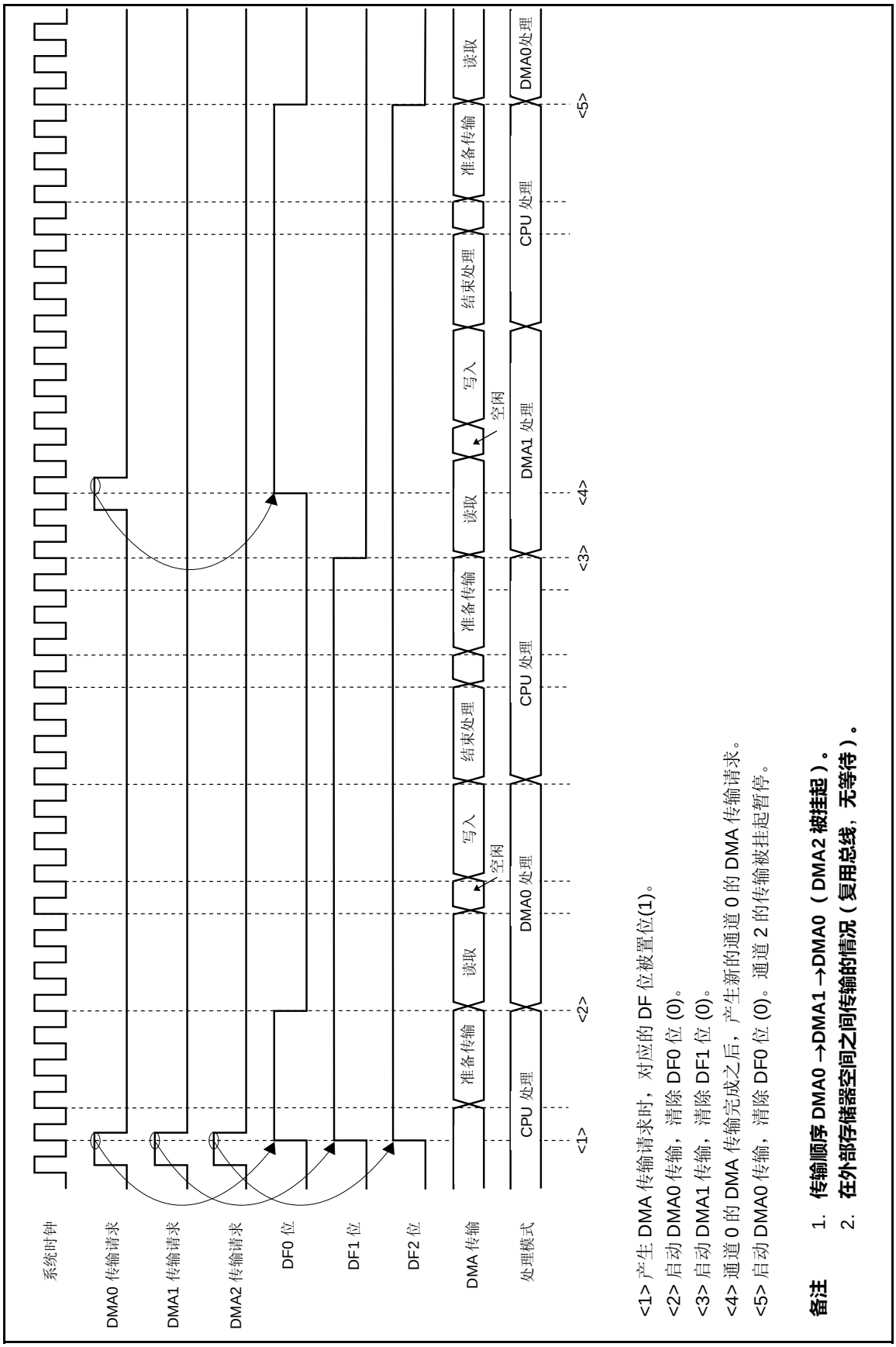
图 21-4. DMA 的优先级 （1）



<1> 产生 DMA 传输请求时，对应的 DF 位被置位(1)。
<2> 启动 DMA0 传输，清除 DF0 位 (0)。
<3> 启动 DMA1 传输，清除 DF1 位 (0)。
<4> 启动 DMA2 传输，清除 DF2 位 (0)。

备注 1. **按 DMA0 → DMA1 → DMA2 的顺序传输。**
2. **在外部存储器空间之间传输的情况（复用总线，无等待）。**

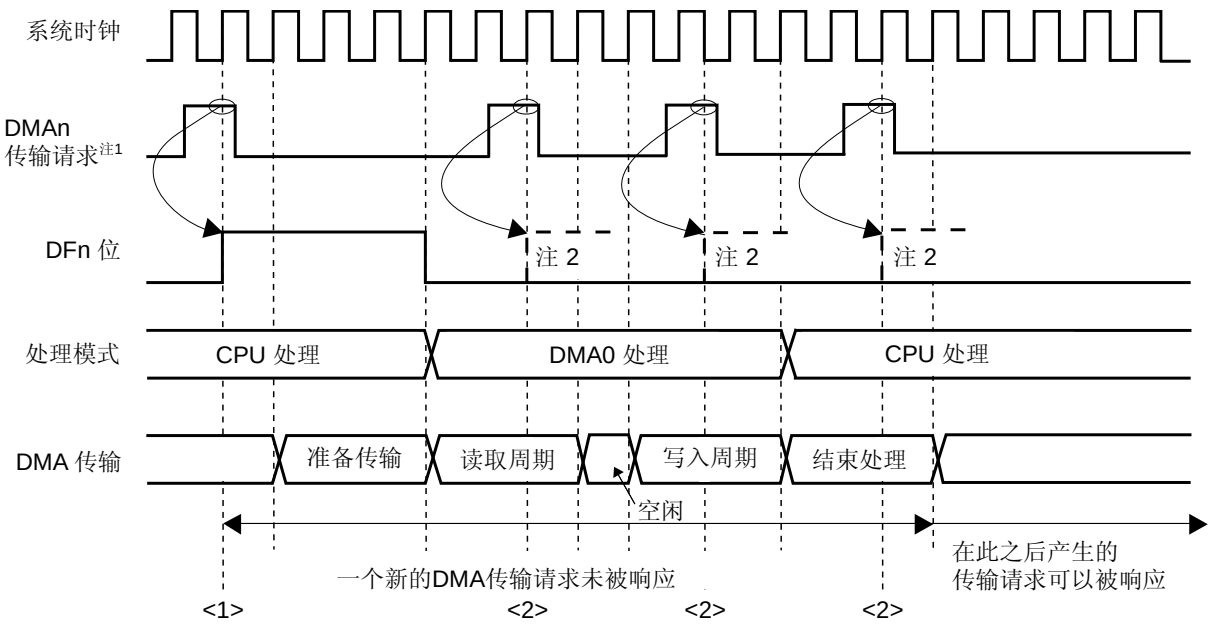
图 21-5. DMA 的优先级 （2）



<1> 产生 DMA 传输请求时，对应的 DF 位被置位(1)。
<2> 启动 DMA0 传输，清除 DF0 位 (0)。
<3> 启动 DMA1 传输，清除 DF1 位 (0)。
<4> 通道 0 的 DMA 传输完成之后，产生新的通道 0 的 DMA 传输请求。
<5> 启动 DMA0 传输，清除 DF0 位 (0)。通道 2 的传输被挂起暂停。

备注 1. **传输顺序 DMA0 →DMA1 →DMA0（DMA2 被挂起）。**
2. **在外部存储器空间之间传输的情况（复用总线，无等待）。**

图 21-6 DMA 传输请求被忽略的周期（1）

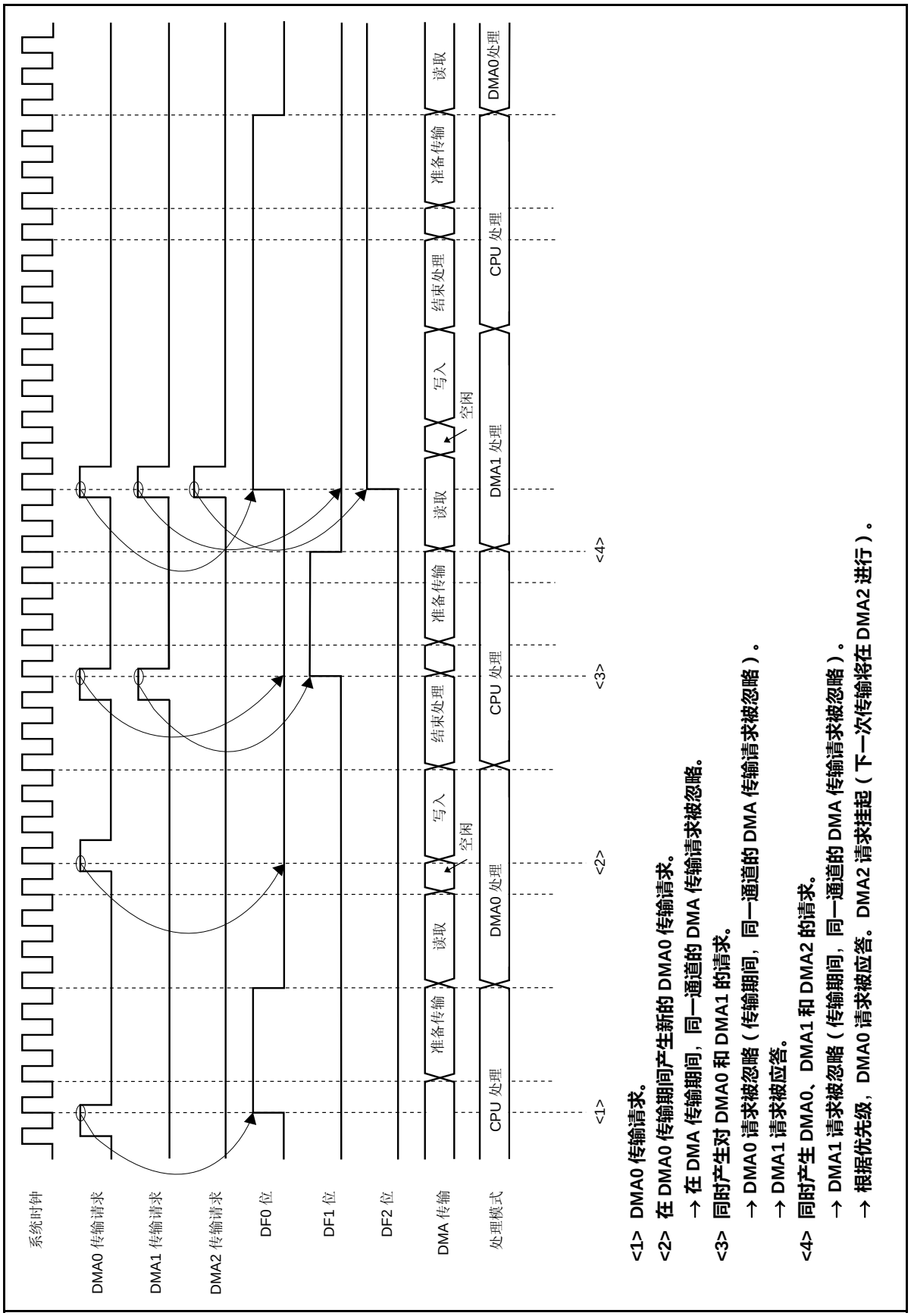


<1> 产生 DMA 传输请求时，对应的 DF 位被置位(1)。
<2> 因为之前的传输尚未完成，新的 DMA 请求被忽略。

注 1. **来自片上外设 I/O 的中断，或软件触发（**DCHCn.**STGn 位）。**
2. **在已产生传输请求和结束处理之间，对同一通道的新的 DMA 请求将被忽略。**

备注 **在外部存储器空间之间传输时（复用总线，无等待）**

图 21-7 DMA 传输请求被忽略的周期（2）



<1> DMA0 传输请求。

<2> 在 DMA0 传输期间产生新的 DMA0 传输请求。

→ 在 DMA 传输期间，同一通道的 DMA 传输请求被忽略。

<3> 同时产生对 DMA0 和 DMA1 的请求。

→ DMA0 请求被忽略（传输期间，同一通道的 DMA 传输请求被忽略）。

→ DMA1 请求被应答。

<4> 同时产生 DMA0、DMA1 和 DMA2 的请求。

→ DMA1 请求被忽略（传输期间，同一通道的 DMA 传输请求被忽略）。

→ 根据优先级，DMA0 请求被应答。DMA2 请求挂起（下一次传输将在 DMA2 进行）。

21.13 注意事项

（1）VSWC 寄存器注意事项

在使用 DMAC 时，确保根据工作频率，为 VSWC 寄存器指定一个合适的值。

如果为 VSWC 寄存器指定一个不合适的值，DMAC 将不能正确操作（关于 VSWC 寄存器的详情，参见 **3.4.8 （1）（a）系统等待控制寄存器（VSWC）**）。

（2）在内部 RAM 中执行 DMA 传输

对未对齐地址执行处于内部 RAM 的数据访问指令时，不要执行与内部 RAM（传输源/目的地）进行数据交互的 DMA 传输，因为 CPU 随后可能会不能正常操作。

同理，当执行与内部 RAM（传输源/目的地）进行数据交互的 DMA 传输时，不要对未对齐地址执行处于内部 RAM 的数据访问指令。

（3）读取 DCHCn.TCn 位（n = 0 至 3）

当读取 TCn 位时， TCn 位被清零。但是在特定时刻读取它也不会清零。要准确清零 TCn 位，请增加以下处理：

（a）通过轮询 TCn 位等待 DMA 传输完成时

确认 TCn 位已经被设置为 1（在读取 TCn 位＝1 后），然后再读 TCn 位三次以上。

（b）在中断服务例程中的读取 TCn 位时

执行 TCn 位的读取操作三次。

（4）DMA 传输初始化方法（将 DCHCn.INITn 位设置为 1）

当执行 DMA 传输的通道要进行初始化时，即使 INITn 位被置为 1，通道可能也无法初始化。要准确地初始化通道，执行以下两种方法之一。

（a）暂时停止所有 DMA 通道的传输

使用以下<1>至<7>步，对执行 DMA 传输的通道进行初始化。

不过，请注意，当执行第<5>步时，TCn 位将被清零。确保其他处理程序不需要 TCn 位为 1。

<1> 禁止中断（DI）。

<2> 除了被强制终止的 DMA 通道之外，读取所有通道的 DCHCn.Enn 位，然后将值传输到通用寄存器。

<3> 清除使用的 DMA 通道（包括被强制终止的通道）的 Enn 位为 0。执行两次清除指令来清除最后一个 DMA 通道的 Enn 位。如果 DMA 传输源或传输目的是内部 RAM，则执行三次清除指令。

举例： 如果使用通道 0，1，2，则按以下顺序执行指令（如果内部 RAM 不是 DMA 传输源或传输目的）。

- 向 DCHC0 写入 00H (将 E00 位清除为 0)。
- 向 DCHC1 写入 00H (将 E11 位清除为 0)。
- 向 DCHC2 写入 00H (将 E22 位清除为 0)。
- 再次向 DCHC2 写入 00H (将 E22 位清除为 0)。

<4> 对强制终止的通道，向该通道对应的 DCHCn 写入 04H（将 INITn 位设置为 1）。

<5> 读取所有未被强制终止的通道的 TCn 位。如果 TCn 位和第<2>步读取的 Enn 位均为 1（逻辑与（AND）结果是 1），将已保存的 Enn 位清零。

<6> 第<5>步的操作完成后，将 Enn 位的值写入 DCHCn 寄存器。

<7> 使能中断（EI）。

注意事项 1. 确保执行上述步骤中的第<5> 步，以防止对某通道 Enn 位的非法设置，该通道的 DMA 传输在第<2>步和第<3>步之间正常完成。

2. 通过位操作指令将 Enn 位清除为 0 (<3>) 并将 INITn 位设置为 1 (<4>)，会清除 TCn 位，所以必须不要使用位操作指令。

（b）重复设置 INITn 位，直到传输被正确地强制终止

<1> 在启动 DMA 之前，将待强制终止通道的初始传输数量拷贝到一个通用寄存器。
<2> 对于待强制终止的通道的 DMA 传输源请求，进行抑制（停止片上外设 I/O 的操作）。
<3> 使用 DTFRn.DFn 位来检查待强制终止的通道的 DMA 传输源请求没有保持挂起。如果一个 DMA 传输请求被保持挂起，等待，直到被挂起的 DMA 传输请求完成。
<4> 已经确认待强制终止的通道的 DMA 传输源请求没有保持挂起，将 Enn 位清除为 0。
<5> 再次将待强制终止的通道的 Enn 位清除为 0。
如果内部 RAM 是待强制终止的通道的传输源或传输目的，再次执行该操作。
<6> 将待强制终止的通道的 INITn 位置为 1。
<7> 读取待强制终止的通道对应的 DBCn 寄存器的值，然后将其与在第<1>步中的拷贝值进行比较。如果两个值不匹配，重复操作第<6>和<7>步。

备注 **1.** 在第<7>步中读取 DBCn 寄存器的值时，如果正确完成强制终止，则读取获得初始传输数目。否则，读取到的值是传输的剩余数目。
2. 注意，如果频繁地使用待强制终止的 DMA 通道之外的通道进行 DMA 传输，则方法（b）会消耗较长时间。

（5）暂时停止 DMA 传输的步骤（清除 Enn 位）

使用以下步骤对操作状态下的 DMA 传输进行停止和恢复：

<1> 对来自 DMA 请求源的传输请求（停止片上外设 I/O 的操作）不作响应。
<2> 使用 DFn 位（检查 DFn 位是否等于 0），确认 DMA 传输请求未处于挂起状态。
如果请求保持挂起状态，等待，直到挂起 DMA 传输的请求执行完成。
<3> 检查 TCn 位以确认 DMA 传输尚未完成（确认 TCn 位为 0）。如果 TCn 位是 1，执行 DMA 传输完成处理。
<4> 如果确认没有 DMA 传输请求处于挂起状态，将 Enn 位清零（该操作会暂停 DMA 传输）。
<5> 将 Enn 位设置为 1，以恢复 DMA 传输。
<6> 对已经停止的 DMA 请求源的操作进行恢复（开始片上外设 I/O 的操作）。

（6）存储边界

在 DMA 传输过程中，如果传输源或目的地（外部存储器、内部 RAM 或片上外设 I/O）的地址超出了 DMA 对象的区域，则操作将无法保证（各个区域地址的详情，参见**图 3-2**）。

（7）传输非对齐的数据

不支持非对齐 16 位数据的 DMA 传输。
如果一个奇地址被指定为传输源或目的地，则地址的最低有效位被强制处理为 0。

（8）CPU 的总线仲裁

因为 DMA 控制器拥有比 CPU 更高优先级的总线权限，在 DMA 传输期间，发生的 CPU 访问将保持挂起状态直到 DMA 传输周期完成，总线被释放给 CPU 为止。

但是，CPU 可以访问未参与 DMA 传输的内部 ROM 和内部 RAM。

- 在外部存储器与片上外设 I/O 之间执行 DMA 传输时，CPU 可以访问内部 ROM 和内部 RAM 。
- 在片上外设 I/O 与内部 RAM 之间执行 DMA 传输时，CPU 可以访问内部 ROM。

（9）DMA 传输时不可重写的寄存器/位

在下列未执行 DMA 传输操作期间，设置下列寄存器（n = 0 至 3）。

[寄存器]

- DSAnH，DSAnL，DDAnH，DDAnL，DBCn 和 DADCn 寄存器。
- DTFRn.IFCn5 至 DTFRn.IFCn0 位。

[设置时序]

- 复位后到第一次 DMA 传输启动期间。
- 通道初始化后到 DMA 传输启动期间。
- DMA 传输完成（TCn 位＝1）后到下一次 DMA 传输启动期间。

（10）确保将以下寄存器的指定位设置为 0（n = 0 至 3）

- DSAnH 寄存器的第 14 位至第 10 位。
- DDAnH 寄存器的第 14 位至第 10 位。
- DADCn 寄存器的第 15 位，第 13 位至第 8 位和第 3 位至第 0 位。
- DCHCn 寄存器的第 6 位至第 3 位。

（11）DMA 启动因素

不要使用相同启动因素开启多个 DMA 通道。如果使用相同的启动因素开启了多个 DMA 通道，那么已经设定的 DMA 通道会再启动，或具有较低优先级的 DMA 通道将在具有较高优先级的 DMA 通道之前被应答，操作无法保证。

（12）读取 DSAn 和 DDAn 寄存器的值

如果在 DMA 传输过程中读取 DSAn 和 DDAn 寄存器，读取得到的可能是更新前或更新后的值。

例如，当 DMA 传输源地址 （DSAn 寄存器）为 0000FFFFH，且计数方向是递增（DADCn.SAD1 和 DADCn.SAD0 位＝00）时，如果先读取 DSAnH 寄存器，然后再读取 DSAnL 寄存器，则根据是否在读取 DSAnH 寄存器之后立即执行 DMA 传输，DSAnL 寄存器的值会有所不同，描述如下。

（a）在读取 DSAn 寄存器时，未发生 DMA 传输

<1> 读取 DSAnH 寄存器的值： DSAnH 寄存器＝0000H

<2> 读取 DSAnL 寄存器的值： DSAnL 寄存器＝FFFFH

（b）在读取 DSAn 寄存器时，发生 DMA 传输

<1> 读取 DSAnH 寄存器的值： DSAnH 寄存器＝0000H

<2> 发生 DMA 传输

<3> DSAn 寄存器递增： DSAn 寄存器＝00100000H

<4> 读取 DSAnL 寄存器的值： DSAnL 寄存器＝0000H

（13）再次设置 DMA 传输

在当前 DMA（TCn 位被设置为 1）期间使用 DDAnH, DDAnL, DSAnH, DSAnL, DBCn 和 DADCn 寄存器重新指定 DMA 设置时，请确保首先对 DMA 通道初始化。DMA 传输的初始化必须使用 **21.13 (4) DMA 传输初始过程**中描述的过程。

第二十二章 中断服务/异常处理功能

V850ES/JG3-L 为中断服务提供专用的中断控制器（INTC），可以处理总共 57 个中断请求。

所谓中断，是独立于程序执行而发生的事件，而异常则是与程序执行相关的事件。

V850ES/JG3-L 能够处理来自于片上外设硬件设备和外部源的中断请求信号。不仅如此，还可以通过 TRAP 指令（软件异常）或产生一个异常事件（非法执行指令）（异常陷阱）来启动异常处理。

22.1 特性

○ 中断

- 不可屏蔽中断： 外部：1 个中断源，内部：1 个中断源
- 可屏蔽中断： 外部：8 个中断源，内部：54 个中断源
- 8 级可编程优先级（可屏蔽中断）
- 依照优先级控制多重中断
- 可以对每一个可屏蔽中断请求进行屏蔽
- 噪声消除、边沿检测以及为外部中断请求信号指定有效边沿

○ 异常

- 软件异常： 32 个中断源
- 异常陷阱： 2 个中断源（非法操作码异常，调试陷阱）

表 22-1 列举了中断/异常发生源。

表 22-1. 中断源列表（1/3）

类型	默认优先级	名称	触发	产生单元	异常码	句柄地址	中断控制寄存器
复位	–	RESET	$\overline{\text{RESET}}$ 引脚输入 由内部源复位输入	RESET	0000H	00000000H	–
不可屏蔽	–	NMI	NMI 引脚有效沿输入	引脚	0010H	00000010H	–
	–	INTWDT2	WDT2 溢出	WDT2	0020H	00000020H	–
软件异常	–	TRAP0n 注2	TRAP 指令	–	004nH 注2	00000040H	–
	–	TRAP1n 注2	TRAP 指令	–	005nH 注2	00000050H	–
异常陷阱	–	ILGOP/ DBG0	非法操作码/ DBTRAP 指令	–	0060H	00000060H	–
可屏蔽	0	INTLVI	低电压检测	POCLVI	0080H	00000080H	LVIIC
	1	INTP0	外部中断引脚输入边沿检测 (INTP0)	Pin	0090H	00000090H	PIC0
	2	INTP1	外部中断引脚输入边沿检测 (INTP1)	Pin	00A0H	000000A0H	PIC1
	3	INTP2	外部中断引脚输入边沿检测 (INTP2)	Pin	00B0H	000000B0H	PIC2
	4	INTP3	外部中断引脚输入边沿检测 (INTP3)	Pin	00C0H	000000C0H	PIC3
	5	INTP4	外部中断引脚输入边沿检测 (INTP4)	Pin	00D0H	000000D0H	PIC4
	6	INTP5	外部中断引脚输入边沿检测 (INTP5)	Pin	00E0H	000000E0H	PIC5
	7	INTP6	外部中断引脚输入边沿检测 (INTP6)	Pin	00F0H	000000F0H	PIC6
	8	INTP7	外部中断引脚输入边沿检测 (INTP7)	Pin	0100H	00000100H	PIC7
	9	INTTQ0OV	TMQ0 溢出	TMQ0	0110H	00000110H	TQ0OVIC
	10	INTTQ0CC0	TMQ0 捕获 0/比较 0 匹配	TMQ0	0120H	00000120H	TQ0CCIC0
	11	INTTQ0CC1	TMQ0 捕获 1/比较 1 匹配	TMQ0	0130H	00000130H	TQ0CCIC1
	12	INTTQ0CC2	TMQ0 捕获 2/比较 2 匹配	TMQ0	0140H	00000140H	TQ0CCIC2
	13	INTTQ0CC3	TMQ0 捕获 3/比较 3 匹配	TMQ0	0150H	00000150H	TQ0CCIC3
	14	INTTP0OV	TMP0 溢出	TMP0	0160H	00000160H	TP0OVIC
	15	INTTP0CC0	TMP0 捕获 0/比较 0 匹配	TMP0	0170H	00000170H	TP0CCIC0
	16	INTTP0CC1	TMP0 捕获 1/比较 1 匹配	TMP0	0180H	00000180H	TP0CCIC1
	17	INTTP1OV/ INTUSBF1	TMP1 溢出/ USBF 恢复中断	TMP1/ USBF	0190H	00000190H	TP1OVIC/ UFIC1
	18	INTTP1CC0	TMP1 捕获 0/比较 0 匹配	TMP1	01A0H	000001A0H	TP1CCIC0
	19	INTTP1CC1/ INTUSBF0	TMP1 捕获 1/比较 1 匹配/ USBF 中断	TMP1/ USBF	01B0H	000001B0H	TP1CCIC1/ UFIC0
	20	INTTP2OV	TMP2 溢出	TMP2	01C0H	000001C0H	TP2OVIC
	21	INTTP2CC0	TMP2 捕获 0/比较 0 匹配	TMP2	01D0H	000001D0H	TP2CCIC0
	22	INTTP2CC1	TMP2 捕获 1/比较 1 匹配	TMP2	01E0H	000001E0H	TP2CCIC1

注 **1.** 产生异常事件的软件可以使用设置在 ECR 寄存器的 EICC 位的异常码来检查。
2. n = 0 至 FH

表 22-1. 中断源列表（2/3）

类型	默认优先级	名称	触发	产生单元	异常码	句柄地址	中断控制寄存器
可屏蔽	23	INTTP3OV /INTUA5R	TMP3 溢出/ INTUA5 接收完成/ UARTA5 接收错误	TMP3/ UARTA5	01F0H	000001F0H	TP3OVIC/ UA5RIC
	24	INTTP3CC0	TMP3 捕获 0/比较 0 匹配	TMP3	0200H	00000200H	TP3/CCIC0
	25	INTTP3CC1 /INTUA5T	TMP3 捕获 1/比较 1 匹配/ INTUA5 连续发送使能	TMP3/ UARTA5	0210H	00000210H	TP3CCIC1/ UA5TIC
	26	INTTP4OV	TMP4 溢出	TMP4	0220H	00000220H	TP4OVIC
	27	INTTP4CC0	TMP4 捕获 0/比较 0 匹配	TMP4	0230H	00000230H	TP4CCIC0
	28	INTTP4CC1	TMP4 捕获 1/比较 1 匹配	TMP4	0240H	00000240H	TP4CCIC1
	29	INTTP5OV	TMP5 溢出	TMP5	0250H	00000250H	TP5OVIC
	30	INTTP5CC0	TMP5 捕获 0/比较 0 匹配	TMP5	0260H	00000260H	TP5CCIC0
	31	INTTP5CC1	TMP5 捕获 1/比较 1 匹配	TMP5	0270H	00000270H	TP5CCIC1
	32	INTTM0EQ0	TMM0 比较匹配	TMM0	0280H	00000280H	TM0EQIC0
	33	INTCB0R/ INTIIC1	CSIB0 接收完成/ CSIB0 接收错误/ IIC1 传输完成	CSIB0/ IIC1	0290H	00000290H	CB0RIC/ IICIC1
	34	INTCB0T	CSIB0 连续发送写入使能	CSIB0	02A0H	000002A0H	CB0TIC
	35	INTCB1R	CSIB1 接收完成/ CSIB1 接收错误	CSIB1	02B0H	000002B0H	CB1RIC
	36	INTCB1T	CSIB1 连续发送写入使能	CSIB1	02C0H	000002C0H	CB1TIC
	37	INTCB2R	CSIB2 接收完成/ CSIB2 接收错误	CSIB2	02D0H	000002D0H	CB2RIC
	38	INTCB2T	CSIB2 连续发送写入使能	CSIB2	02E0H	000002E0H	CB2TIC
	39	INTCB3R	CSIB3 接收完成/ CSIB3 接收错误	CSIB3	02F0H	000002F0H	CB3RIC
	40	INTCB3T	CSIB3 连续发送写入使能	CSIB3	0300H	00000300H	CB3TIC
	41	INTUA0R/ INTCB4R	UARTA0 接收完成/ UARTA0 接收错误/ CSIB4 接收完成/ CSIB4 接收错误	UARTA0/ CSIB4	0310H	00000310H	UA0RIC/ CB4RIC
	42	INTUA0T/ INTCB4T	UARTA0 连续发送使能/CSIB4 连续发送写入使能	UARTA0/ CSIB4	0320H	00000320H	UA0TIC/ CB4TIC
	43	INTUA1R/ INTIIC2	UARTA1 接收完成/ UARTA1 接收错误/ IIC2 传输完成	UARTA1/ IIC2	0330H	00000330H	UA1RIC/ IICIC2
	44	INTUA1T	UARTA1 连续发送使能	UARTA1	0340H	00000340H	UA1TIC
	45	INTUA2R/ INTIIC0	UARTA2 接收完成/ UARTA2 接收错误/ IIC0 传输完成	UARTA2/ IIC0	0350H	00000350H	UA2RIC/ IICIC0

表 22-1. 中断源列表（3/3）

类型	默认优先级	名称	触发	产生单元	异常码	句柄地址	中断控制寄存器
可屏蔽	46	INTUA2T	UARTA2 连续发送使能	UARTA2	0360H	00000360H	UA2TIC
	47	INTAD	A/D 转换完成	A/D	0370H	00000370H	ADIC
	48	INTDMA0	DMA0 传输完成	DMA	0380H	00000380H	DMAIC0
	49	INTDMA1	DMA1 传输完成	DMA	0390H	00000390H	DMAIC1
	50	INTDMA2	DMA2 传输完成	DMA	03A0H	000003A0H	DMAIC2
	51	INTDMA3	DMA3 传输完成	DMA	03B0H	000003B0H	DMAIC3
	52	INTKR	按键返回中断	KR	03C0H	000003C0H	KRIC
	53	INTWTI /INTRTC2	钟表定时器间隔/ RTC 间隔信号	WT/ RTC	03D0H	000003D0H	WTIIC/ RTC2IC
	54	INTWT /INTRTC0	钟表定时器参考时间/ RTC 固定周期信号	WT/ RTC	03E0H	000003E0H	WTIC/ RTC0IC
	55	INTRTC1	RTC 闹钟匹配	RTC	03F0H	000003F0H	RTC1IC
	56	INTUA3R	UARTA3 接收完成/ UARTA3 接收错误	UARTA3	0400H	00000400H	UA3RIC
	57	INTUA3T	UARTA3 连续发送使能	UARTA3	0410H	00000410H	UA3TIC
	58	INTUA4R	UARTA4 接收完成/ UARTA4 接收错误	UARTA4	0420H	00000420H	UA4RIC
	59	INTUA4T	UARTA4 连续发送使能	UARTA4	0430H	00000430H	UA4TIC
	60	INTUC0R	UARTC0 接收完成/ UARTC0 接收错误	UARTC0	0440H	00000440H	UC0RIC
	61	INTUC0T	UARTC0 连续发送使能	UARTC0	0450H	00000450H	UC0TIC

备注 **1.** 默认优先级： 当多个具有相同优先级的可屏蔽中断请求同时发生时，系统指定的优先级次序适用。较小的编号具有较高的优先级，最高优先级是 0。不可屏蔽中断的优先级次序是 INTWDT2 高于 NMI。

2. 当发生非法操作码异常时，非法指令的执行地址计算方法为：（被恢复的 PC－4）。

22.2 不可屏蔽中断

当中断被 CPU 禁止（DI）时，也可以响应不可屏蔽中断请求信号。不可屏蔽中断（NMI）并不受优先级控制，而且，比所有其他中断请求信号更优先。

该产品有以下两种不可屏蔽中断请求信号。

- NMI 引脚输入（NMI）。
- 看门狗定时器溢出产生的不可屏蔽中断请求信号（INTWDT2）。

NMI 引脚的有效沿可以选择为以下 4 种类型：“上升沿”、“下降沿”、“双边沿”和“无边沿检测”。默认选择是“无边沿检测”，请确保指定有效边沿。

当 WDTM2.WDM21 和 WDTM2.WDM20 位被设置为“01”时，看门狗定时器 2（INTWDT2）的溢出会产生不可屏蔽中断请求信号。

当两个或两个以上不可屏蔽中断请求信号同时发生时，较高优先级的中断得到相应（低优先级的中断请求信号被忽略），如下所示：

INTWDT2 > NMI

正在处理不可屏蔽中断时，发生新的 NMI 或 INTWDT2 请求信号，则按以下情况处理。

（1）如果正在处理不可屏蔽中断时，发生新的 NMI 请求信号

不管 PSW.NP 位的值如何，新的 NMI 请求信号都会被挂起。挂起的 NMI 请求信号在当前不可屏蔽中断处理完成后得到响应（在 RETI 指令执行完成之后）。

（2）如果正在处理不可屏蔽中断时，发生 INTWDT2 请求信号

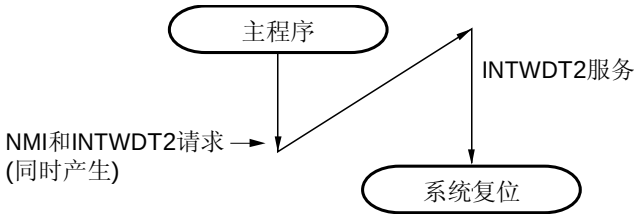
正在处理不可屏蔽中断时，如果 NP 位被置位（1），INTWDT2 请求信号被挂起。挂起的 INTWDT2 请求信号在当前不可屏蔽中断处理完成后得到响应（在 RETI 指令执行完成之后）。

正在处理不可屏蔽中断时，如果 NP 位被清除（0），应答新产生的 INTWDT2 请求信号（当前的不可屏蔽中断处理服务停止）。

注意事项 关于 INTWDT2 信号请求的不可屏蔽中断服务详情，参见 22.2.2（2）来自 INTWDT2 信号。

图 22-1. 不可屏蔽中断请求信号的应答操作（1/2）

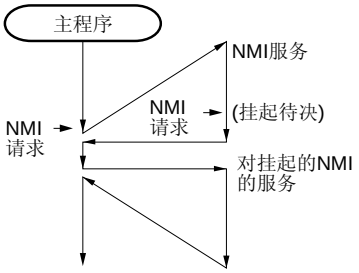
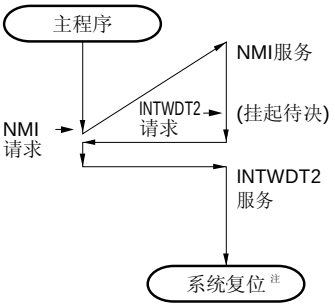
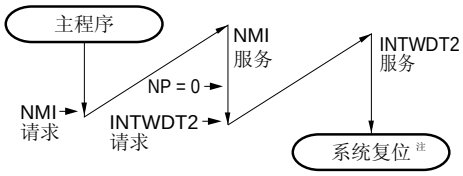
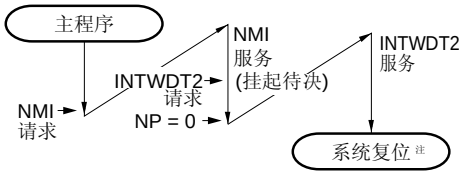
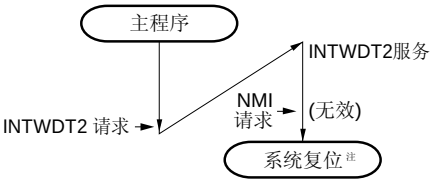
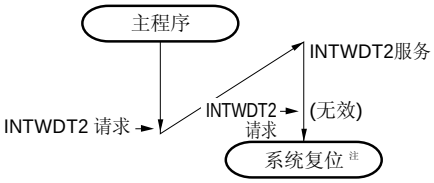
（a） 同时产生 NMI 和 INTWDT2 请求信号



注 执行初始化例程以重新启动处理。

图 22-1. 不可屏蔽中断请求信号的应答操作（2/2）

（b） 如果在不可屏蔽中断服务期间产生不可屏蔽中断请求信号

不可屏蔽中断服务中	不可屏蔽中断服务时，产生新的不可屏蔽中断请求信号	
	NMI	INTWDT2
NMI	· NMI处理时产生新的NMI请求 	· NMI处理时产生新的INTWDT2请求 (INTWDT2 请求前保持NP 位 = 1)  · NMI服务期间产生INTWDT2 请求 (INTWDT2 请求前设置NP 位 = 0)  · NMI服务期间产生INTWDT2 请求 (INTWDT2 请求之后设置NP = 0) 
INTWDT2	· INTWDT2 处理时产生新的NMI请求 	· INTWDT2处理时产生新的INTWDT2请求 

注 执行初始化例程以重新启动处理。

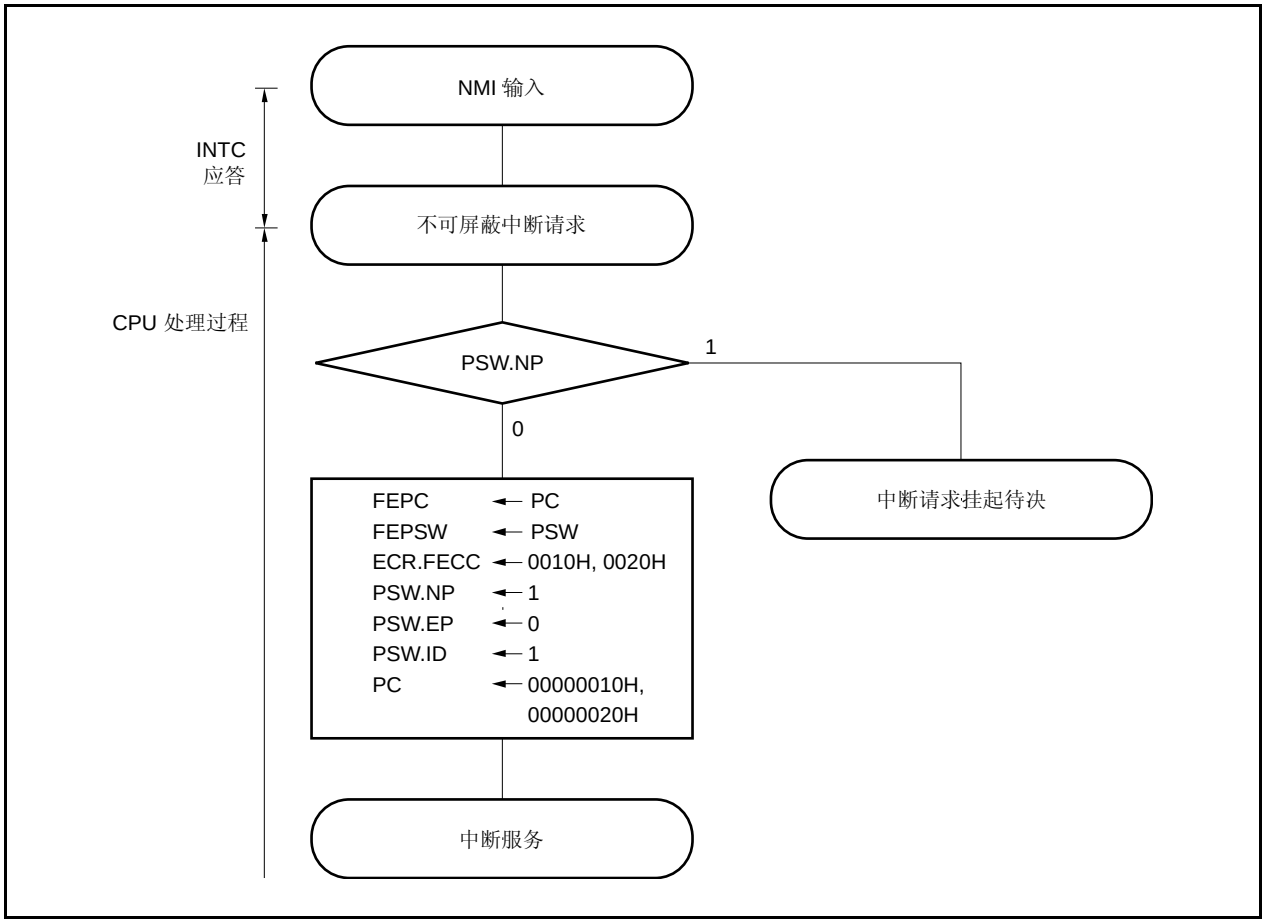
22.2.1 操作

如果产生一个不可屏蔽中断请求信号，则 CPU 执行以下处理过程，并将控制权转移至一个句柄程序。

<1> 将当前 PC 保存至 FEPC。
<2> 将当前 PSW 保存至 FEPSW。
<3> 将异常码（0010H，0020H）写入 ECR 的高半字（FECC）。
<4> 将 PSW.NP 和 PSW.ID 位置为 1，并将 PSW.EP 位清为 0。
<5> 将不可屏蔽中断相应的句柄地址 （00000010H，00000020H）载入 PC，并转移控制。

不可屏蔽中断的服务流程如下所示：

图 22-2. 不可屏蔽中断服务流程



22.2.2 恢复

（1）来自 NMI 引脚的输入

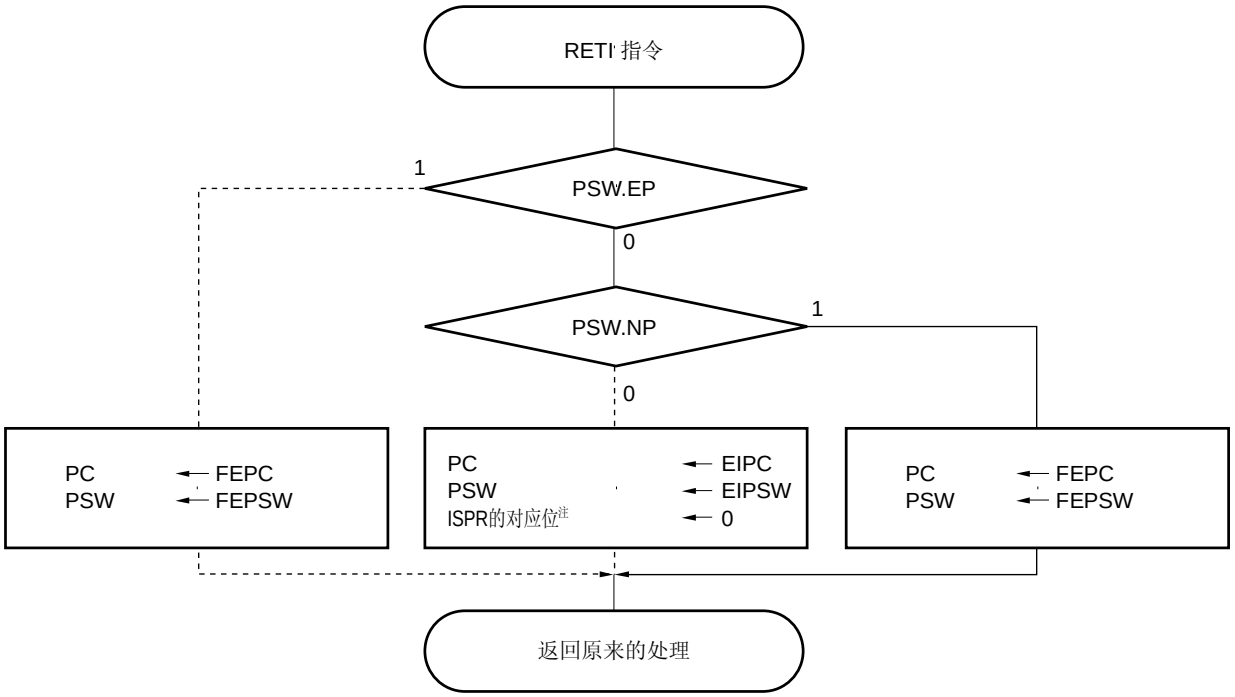
使用 RETI 指令从 NMI 服务中恢复执行。

当执行 RETI 指令时，CPU 执行以下处理过程，并且控制转移到被恢复的 PC 地址。

<1> 因为 PSW.EP 位是 0 且 PSW.NP 位是 1，所以，由 FEPC 和 FEPSW 分别载入相应 PC 和 PSW 的保存值。
<2> 控制转移返回到被恢复的 PC 和 PSW 的地址。

RETI 指令的处理过程如下所示：

图 22-3. RETI 指令处理过程



注 ISPR 寄存器的详情，参见 **22.3.6 正在服务的优先级寄存器 (ISPR)。**

注意事项 在不可屏蔽中断服务期间，如果 EP 位和 NP 位被 LDSR 指令改变时，要在使用 RETI 指令返回时正确恢复 PC 和 PSW，必须在紧邻 RETI 指令之前，必须使用 LDSR 指令将 EP 位设置为 0，并必须将 NP 位置为 1。

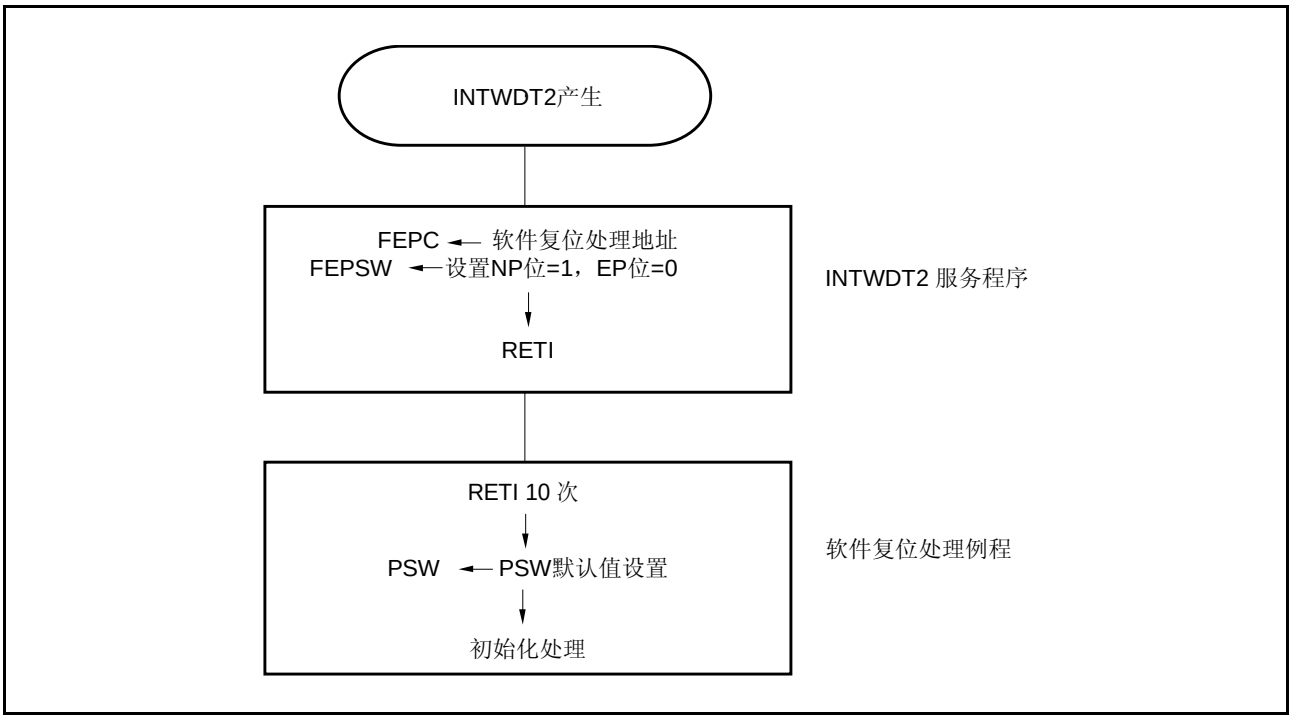
备注 实线显示的是 CPU 的处理流程。

（2）来自 INTWDT2 的信号

对于 INTWDT2 的不可屏蔽中断服务，不能使用 RETI 指令来返回。要想从这种服务中返回，执行下面的软件复位处理。

但是，在软件复位处理中，那些只能在复位结束后立即设置一次的寄存器（比如 WDTM2 寄存器）无法再次设置。要将这些寄存器重置为它们的初始值，需要硬件复位，比如 RESET 引脚低电平输入。

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，用于指示不可屏蔽中断服务正在进行。

当不可屏蔽中断请求信号被响应之后，该标志置位，并且，屏蔽不可屏蔽中断请求，禁止响应多重中断。

	31 … 8	7	6	5	4	3	2	1	0
PSW	0	NP	EP	ID	SAT	CY	OV	S	Z

NP	不可屏蔽中断服务状态
0	没有不可屏蔽中断在服务（初始值）
1	不可屏蔽中断正在处理中

22.3 可屏蔽中断

可屏蔽中断请求信号可以由中断控制寄存器进行屏蔽。V850ES/JG3-L 单片机有 55 个可屏蔽中断源。

当一个中断请求信号已经得到响应时，中断被禁止（DI），且不会响应后面的可屏蔽中断请求信号。

当在中断服务程序中执行 EI 指令，中断被使能（EI），可以使那些比正在处理的中断请求信号（由中断控制寄存器指定）具有更高的优先级的中断得到响应。具有相同优先等级的中断不能嵌套。

多重中断的详情，参见 **22.6 多重中断服务控制。**

22.3.1 操作

如果发生可屏蔽中断请求信号，CPU 执行以下处理过程，并将控制转移至句柄程序。

<1> 将当前 PC 保存至 EIPC。
<2> 将当前 PSW 保存至 EIPSW。
<3> 将异常码写入 ECR 的低半字（EICC）。
<4> 将 PSW.ID 位置 1，并将 PSW.EP 位清 0。
<5> 将对应的句柄地址装载到 PC 指针，并进行控制转移。

由中断控制器（INTC）屏蔽的可屏蔽中断请求信号(xxMK 位 = 1)，和在另外的中断服务处理（PSW.NP 位为 1 或 ID 位为 1）时产生的可屏蔽中断请求信号，都被保持挂起在 INTC 内。被挂起的原因和变通办法描述如下。

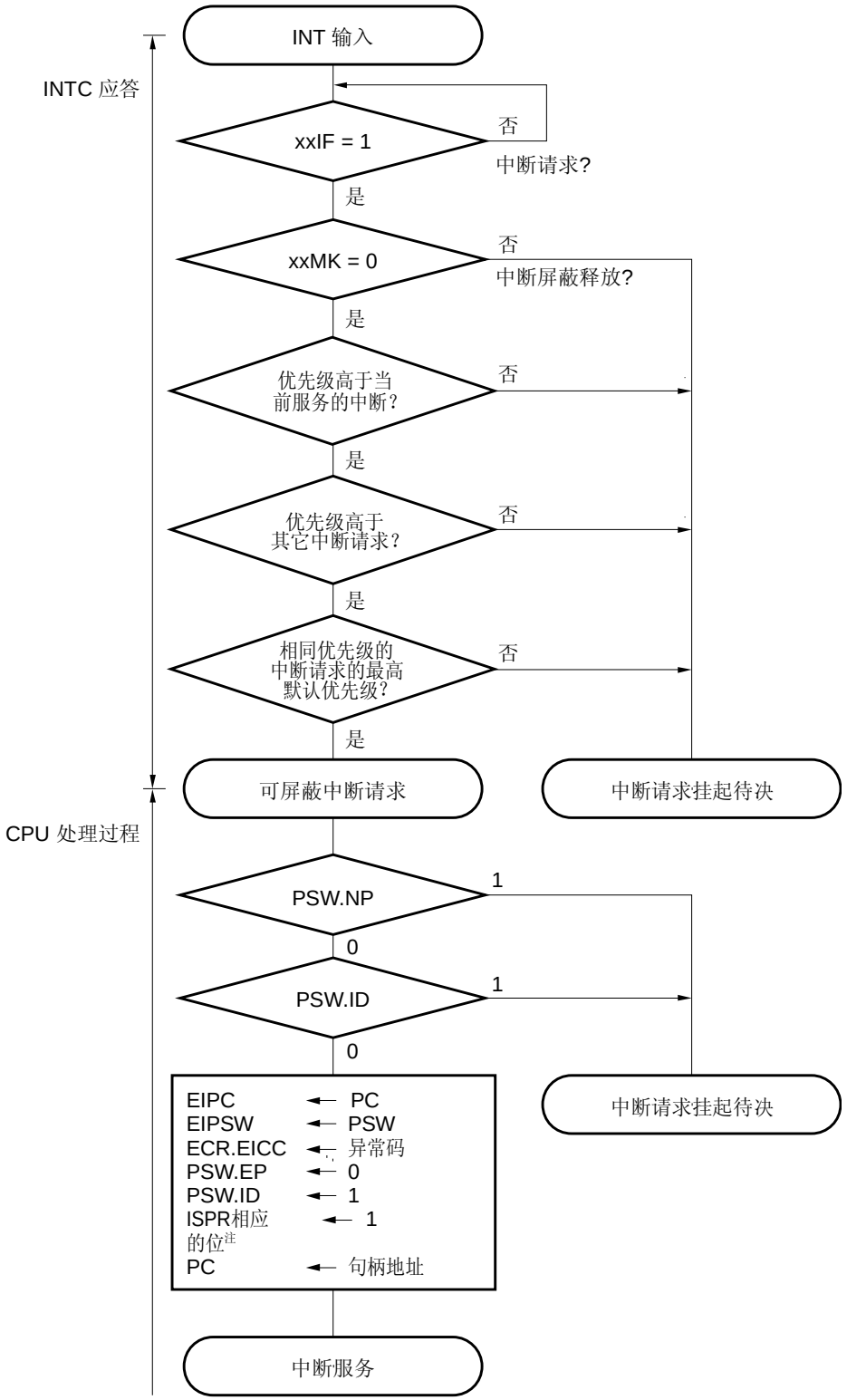
表 22-2. 被挂起的可屏蔽中断

原因	变通办法
xxMK 位 = 1	解除该信号的屏蔽 (将 xxMK 位清除为 0).
另一个具有更高优先级的中断被保持挂起	等待中断服务结束
PSW.NP 位= 1 且 PSW.ID 位 = 1	使用 RETI 和 LDSR 指令设置 NP 位为 1 且 ID 位为 1

备注 xxMK 位的详情，参见 **22.3.4 中断控制寄存器 (xxICn)。**

图 22-5 显示了可屏蔽中断的服务过程。

图 22-5. 可屏蔽中断服务过程



注 ISPR 寄存器的详情，参见 **22.3.6 服务中优先级寄存器（ISPR）。**

22.3.2 恢复

使用 RETI 指令从可屏蔽中断服务中返回执行。

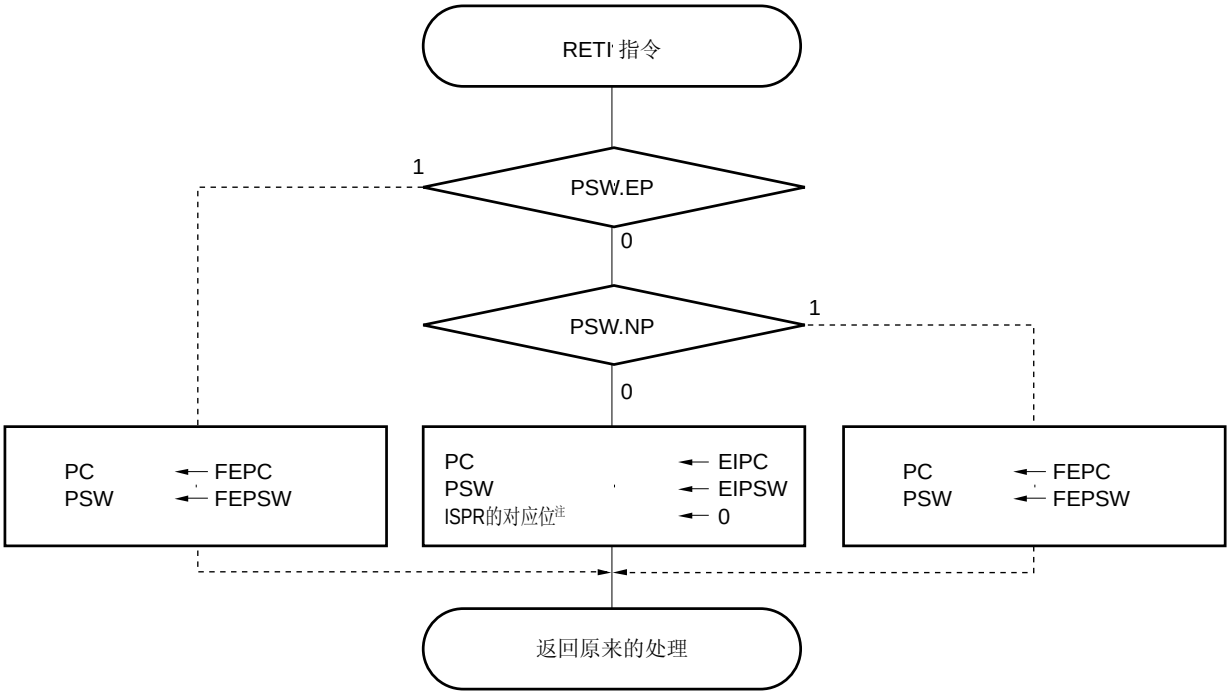
当执行 RETI 指令时，CPU 执行以下处理过程，并且控制转移到被恢复的 PC 地址。

<1> 因为 PSW.EP 位为 0 且 PSW.NP 位为 0，所以由 EIPC 和 EIPSW 分别载入 PC 和 PSW 的保存值。

<2> 控制转移返回到被恢复的 PC 和 PSW 的地址。

RETI 指令的处理过程如下所示：

图 22-6. RETI 指令处理过程



注 ISPR 寄存器的详情，参见 **22.3.6 正在服务的优先级寄存器 (ISPR)。**

注意事项 在不可屏蔽中断服务期间，如果 EP 位和 NP 位被 LDSR 指令改变时，要在使用 RETI 指令返回时正确恢复 PC 和 PSW，必须在紧邻 RETI 指令之前，必须使用 LDSR 指令将 EP 位设置为 0，并必须将 NP 位置为 1。

备注 实线显示的是 CPU 的处理流程。

22.3.3 可屏蔽中断的优先级

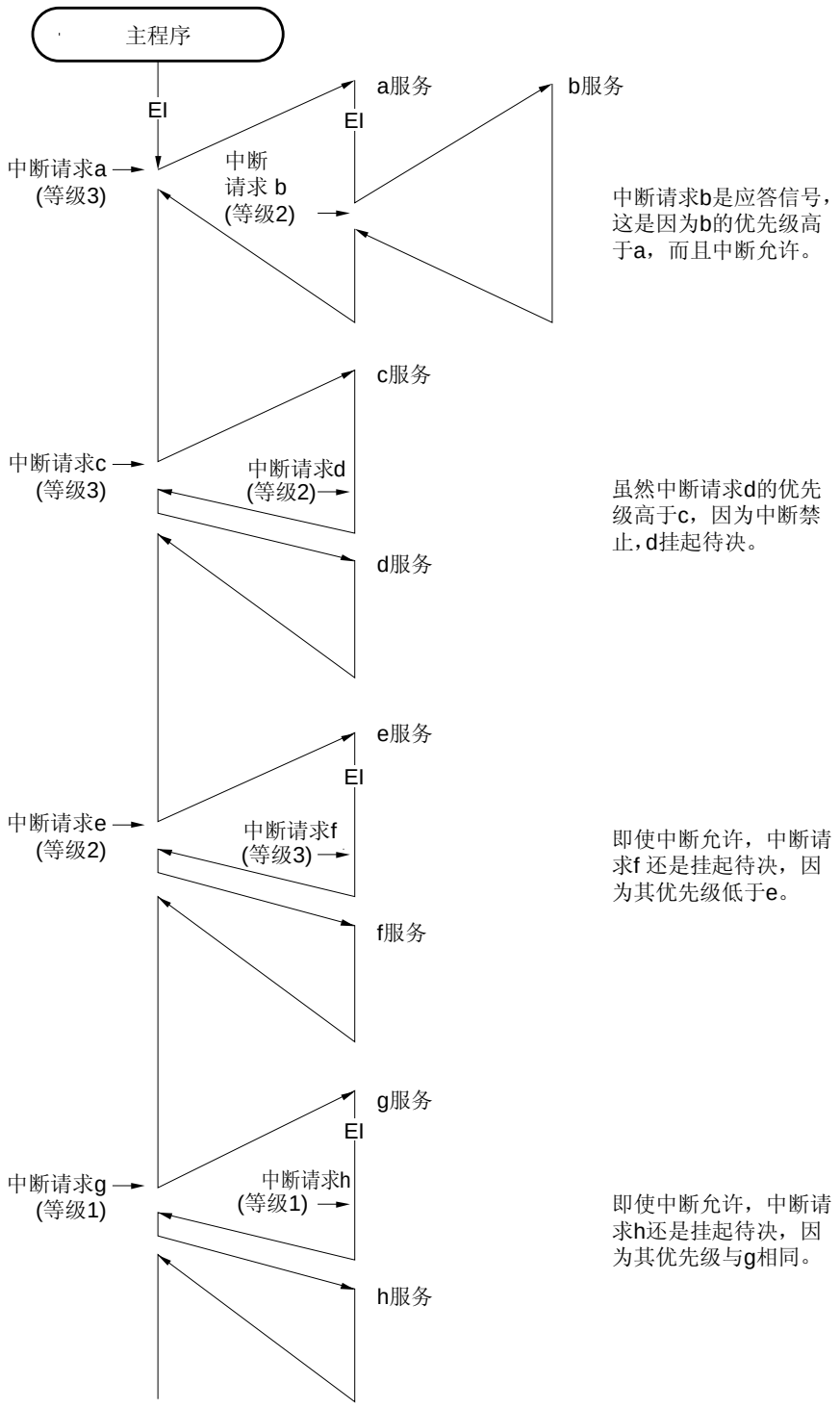
在一个中断正在服务处理的同时 INTC 可以响应另一个中断，并根据优先等级来服务。

有两种类型的优先级控制方式：基于中断控制寄存器（xxICn）的中断优先级指定位（xxPRn）确定的可编程优先级控制，以及基于默认优先级的控制。可编程优先级控制根据 xxPRn 标志将中断请求信号分为 8 个级别。当多个同时发生的中断的具有 xxPRn 位指定的相同优先级时，按照每个中断请求类型预先设定（默认优先级）的优先级次序进行处理。详情参见 **表 22-1 中断源列表**。

关于多重中断的详情，参见 **22.7 多重中断服务控制**。

备注 xx： 每一个外设单元的标志名称（参见 **表 22-3 中断控制寄存器（xxICn）**）。
n： 外设单元编号（参见**表 22-3 中断控制寄存器（xxICn）**）。

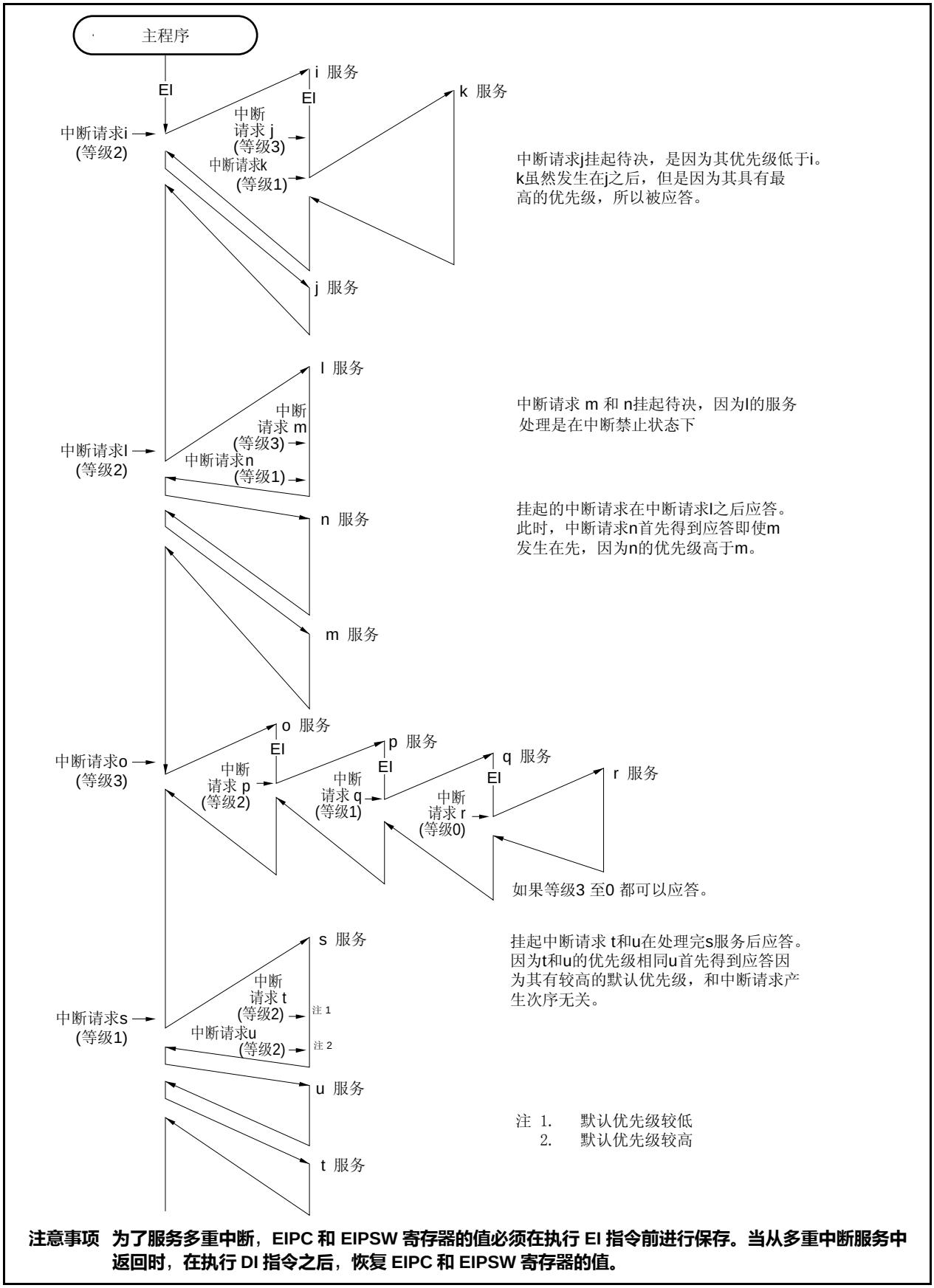
图 22-7. 当一个中断服务时发生另一个中断请求的处理示例（1/2）



注意事项 **为了服务多重中断，EIPC 和 EIPSW 寄存器的值必须在执行 EI 指令前进行保存。当从多重中断服务中返回时，在执行 DI 指令之后，恢复 EIPC 和 EIPSW 寄存器的值。**

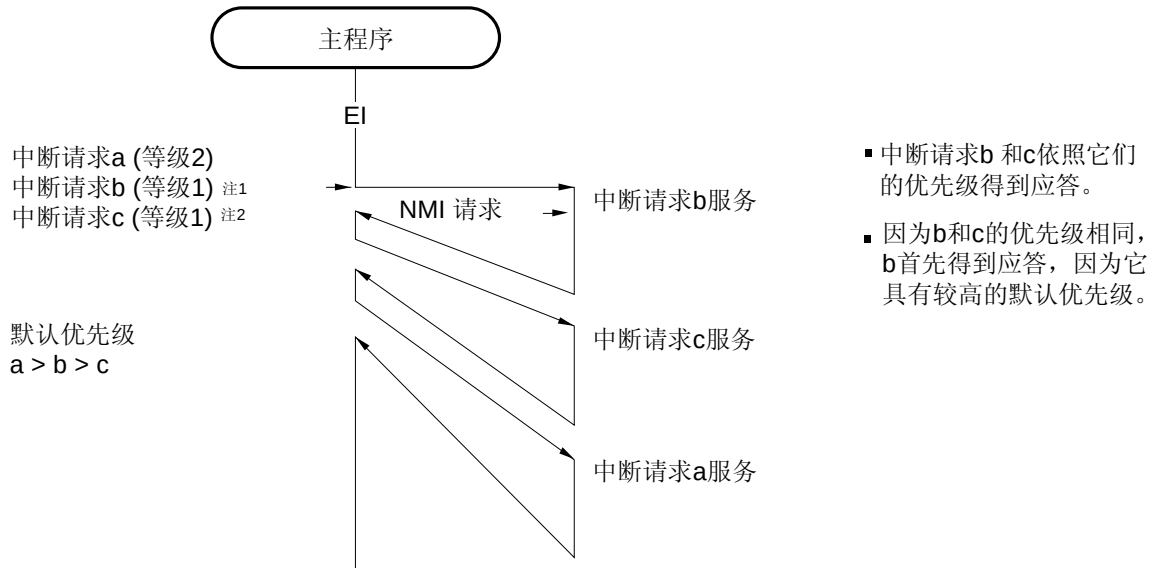
备注
1. 为了解释说明，上图中 a 至 u 是中断请求信号的暂用名。
2. 上图的默认优先级表明两个中断请求信号之间的相对优先级。

图 22-7. 一个中断服务当中而另一个中断请求发生的处理举例（2/2）



注意事项 为了服务多重中断，EIPC 和 EIPSW 寄存器的值必须在执行 EI 指令前进行保存。当从多重中断服务中返回时，在执行 DI 指令之后，恢复 EIPC 和 EIPSW 寄存器的值。

图 22-8. 同时发生的中断请求信号服务举例



注 **1.** 较高默认优先级
2. 较低默认优先级

注意事项 **为了执行多重中断服务，EIPC 和 EIPSW 寄存器的值必须在执行 EI 指令前进行保存。当从多重中断服务中返回时，在执行 DI 指令之后，恢复 EIPC 和 EIPSW 寄存器的值。**

备注 **1.** 为了解释说明，图中的 a 至 c 是中断请求信号的暂用名。
2. 图中的默认优先级表明两个中断请求信号之间的相对优先级。

22.3.4 中断控制寄存器（xxICn）

每一个可屏蔽中断请求信号（可屏蔽中断）都被指定一个 xxICn 寄存器，用来设置每一个可屏蔽中断请求的控制条件。
该寄存器可以按字节或按位进行读写。
系统复位后，寄存器被置为 47H。

注意事项 1. 要屏蔽中断，设置 IMR 寄存器或使用位操作指令。优先级别必须在无中断发生时指定。
2. 读取 xxICn.xxIFn 位之前禁止中断（DI）。如果在中断允许（EI）时读取 xxIFn 位，当响应中断与读取该位的时序发生冲突时，可能读不到正确值。

复位后: 47H R/W 地址: FFFFF110H 至 FFFFF17CH

	<7>	<6>	5	4	3	2	1	0
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	中断请求标志注
0	不产生中断请求
1	产生中断请求

xxMKn	中断屏蔽标志
0	中断服务允许
1	中断服务禁止 (挂起)

xxPRn2	xxPRn1	xxPRn0	中断优先级指定位
0	0	0	指定等级0 (最高)
0	0	1	指定等级1
0	1	0	指定等级2
0	1	1	指定等级3
1	0	0	指定等级4
1	0	1	指定等级5
1	1	0	指定等级6
1	1	1	指定等级7 (最低)

注 如果一个中断请求信号得到响应，xxIFn 标志由硬件自动复位。

备注 xx： **每一个外设单元的标志名称（参见 表 22-3 中断控制寄存器（xxICn））。**
n ： **外设单元编号（参见表 22-3 中断控制寄存器（xxICn））。**

中断控制寄存器的地址及位如下所示：

表 22-3. 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	LVIIC	LVIIF	LVIMK	0	0	0	LVIPR2	LVIPR1	LVIPR0
FFFFF112H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF114H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF116H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF118H	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
FFFFF11AH	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
FFFFF11CH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
FFFFF11EH	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60
FFFFF120H	PIC7	PIF7	PMK7	0	0	0	PPR72	PPR71	PPR70
FFFFF122H	TQ0OVIC	TQ0OVIF	TQ0OVMK	0	0	0	TQ0OVPR2	TQ0OVPR1	TQ0OVPR0
FFFFF124H	TQ0CCIC0	TQ0CCIF0	TQ0CCMK0	0	0	0	TQ0CCPR02	TQ0CCPR01	TQ0CCPR00
FFFFF126H	TQ0CCIC1	TQ0CCIF1	TQ0CCMK1	0	0	0	TQ0CCPR12	TQ0CCPR11	TQ0CCPR10
FFFFF128H	TQ0CCIC2	TQ0CCIF2	TQ0CCMK2	0	0	0	TQ0CCPR22	TQ0CCPR21	TQ0CCPR20
FFFFF12AH	TQ0CCIC3	TQ0CCIF3	TQ0CCMK3	0	0	0	TQ0CCPR32	TQ0CCPR31	TQ0CCPR30
FFFFF12CH	TP0OVIC	TP0OVIF	TP0OVMK	0	0	0	TP0OVPR2	TP0OVPR1	TP0OVPR0
FFFFF12EH	TP0CCIC0	TP0CCIF0	TP0CCMK0	0	0	0	TP0CCPR02	TP0CCPR01	TP0CCPR00
FFFFF130H	TP0CCIC1	TP0CCIF1	TP0CCMK1	0	0	0	TP0CCPR12	TP0CCPR11	TP0CCPR10
FFFFF132H	TP1OVIC/ UFIC1	TP1OVIF/ UFIF1	TP1OVMK/ UFMK1	0	0	0	TP1OVPR2/ UFPPR12	TP1OVPR1/ UFPPR11	TP1OVPR0/ UFPPR10
FFFFF134H	TP1CCIC0	TP1CCIF0	TP1CCMK0	0	0	0	TP1CCPR02	TP1CCPR01	TP1CCPR00
FFFFF136H	TP1CCIC1/ UFIC0	TP1CCIF1/ UFIF0	TP1CCMK1/ UFMK0	0	0	0	TP1CCPR12/ UFPPR02	TP1CCPR11/ UFPPR01	TP1CCPR10/ UFPPR00
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 /UA5RIC	TP3OVIF /UA5RIF	TP3OVMK /UA5RMK	0	0	0	TP3OVPR2 /UA5RPR2	TP3OVPR1 /UA5RPR1	TP3OVPR0 /UA5RPR0
FFFFF140H	TP3CCIC0	TP3CCIF0	TP3CCMK0	0	0	0	TP3CCPR02	TP3CCPR01	TP3CCPR00
FFFFF142H	TP3CCIC1 /UA5TIC	TP3CCIF1 /UA5TIF	TP3CCMK1 /UA5TMK	0	0	0	TP3CCPR12 /UA5TPR2	TP3CCPR11 /UA5TPR1	TP3CCPR10 /UA5TPR0
FFFFF144H	TP4OVIC	TP4OVIF	TP4OVMK	0	0	0	TP4OVPR2	TP4OVPR1	TP4OVPR0
FFFFF146H	TP4CCIC0	TP4CCIF0	TP4CCMK0	0	0	0	TP4CCPR02	TP4CCPR01	TP4CCPR00
FFFFF148H	TP4CCIC1	TP4CCIF1	TP4CCMK1	0	0	0	TP4CCPR12	TP4CCPR11	TP4CCPR10
FFFFF14AH	TP5OVIC	TP5OVIF	TP5OVMK	0	0	0	TP5OVPR2	TP5OVPR1	TP5OVPR0
FFFFF14CH	TP5CCIC0	TP5CCIF0	TP5CCMK0	0	0	0	TP5CCPR02	TP5CCPR01	TP5CCPR00
FFFFF14EH	TP5CCIC1	TP5CCIF1	TP5CCMK1	0	0	0	TP5CCPR12	TP5CCPR11	TP5CCPR10
FFFFF150H	TM0EQIC0	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR02	TM0EQPR01	TM0EQPR00
FFFFF152H	CB0RIC/ IICIC1	CB0RIF/ IICIF1	CB0RMK/ IICMK1	0	0	0	CB0RPR2/ IICPR12	CB0RPR1/ IICPR11	CB0RPR0/ IICPR10
FFFFF154H	CB0TIC	CB0TIF	CB0TMK	0	0	0	CB0TPR2	CB0TPR1	CB0TPR0
FFFFF156H	CB1RIC	CB1RIF	CB1RMK	0	0	0	CB1RPR2	CB1RPR1	CB1RPR0
FFFFF158H	CB1TIC	CB1TIF	CB1TMK	0	0	0	CB1TPR2	CB1TPR1	CB1TPR0
FFFFF15AH	CB2RIC	CB2RIF	CB2RMK	0	0	0	CB2RPR2	CB2RPR1	CB2RPR0
FFFFF15CH	CB2TIC	CB2TIF	CB2TMK	0	0	0	CB2TPR2	CB2TPR1	CB2TPR0
FFFFF15EH	CB3RIC	CB3RIF	CB3RMK	0	0	0	CB3RPR2	CB3RPR1	CB3RPR0

表 22-3. 中断控制寄存器（xxICn）（2/2）

地址	寄存器	位							
		<7>	<6>	5	4	3	2	1	0
FFFFF160H	CB3TIC	CB3TIF	CB3TMK	0	0	0	CB3TPR2	CB3TPR1	CB3TPR0
FFFFF162H	UA0RIC/ CB4RIC	UA0RIF/ CB4RIF	UA0RMK/ CB4RMK	0	0	0	UA0RPR2/ CB4RPR2	UA0RPR1/ CB4RPR1	UA0RPR0/ CB4RPR0
FFFFF164H	UA0TIC/ CB4TIC	UA0TIF/ CB4TIF	UA0TMK/ CB4TMK	0	0	0	UA0TPR2/ CB4TPR2	UA0TPR1/ CB4TPR1	UA0TPR0/ CB4TPR0
FFFFF166H	UA1RIC/ IICIC2	UA1RIF/ IICIF2	UA1RMK/ IICMK2	0	0	0	UA1RPR2/ IICPR22	UA1RPR1/ IICPR21	UA1RPR0/ IICPR20
FFFFF168H	UA1TIC	UA1TIF	UA1TMK	0	0	0	UA1TPR2	UA1TPR1	UA1TPR0
FFFFF16AH	UA2RIC/ IICIC0	UA2RIF/ IICIF0	UA2RMK/ IICMK0	0	0	0	UA2RPR2/ IICPR02	UA2RPR1/ IICPR01	UA2RPR0/ IICPR00
FFFFF16CH	UA2TIC	UA2TIF	UA2TMK	0	0	0	UA2TPR2	UA2TPR1	UA2TPR0
FFFFF16EH	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF170H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF172H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF174H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF176H	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF178H	KRIC	KRIF	KRMK	0	0	0	KRPR2	KRPR1	KRPR0
FFFFF17AH	WTIIC /RTC2IC	WTIIF /RTC2IF	WTIMK /RTC2MK	0	0	0	WTIPR2 /RTC2PPR2	WTIPR1 /RTC2PPR1	WTIPR0 /RTC2PPR0
FFFFF17CH	WTIC /RTC0IC	WTIF /RTC0IF	WTMK /RTC0MK	0	0	0	WTPR2 /RTC0PPR2	WTPR1 /RTC0PPR1	WTPR0 /RTC0PPR0
FFFFF17EH	RTC1IC	RTC1IF	RTC1MK	0	0	0	RTC1PPR2	RTC1PPR1	RTC1PPR0
FFFFF180H	UA3RIC	UA3RIF	UA3RMK	0	0	0	UA3RPR2	UA3RPR1	UA3RPRo
FFFFF182H	UA3TIC	UA3TIF	UA3TMK	0	0	0	UA3TPR2	UA3TPR1	UA3TPR0
FFFFF184H	UA4RIC	UA4RIF	UA4RMK	0	0	0	UA4RPR2	UA4RPR1	UA4RPR0
FFFFF186H	UA4TIC	UA4TIF	UA4TMK	0	0	0	UA4TPR2	UA4TPR1	UA4TPR0
FFFFF188H	UC0RIC	UC0RIF	UC0RMK	0	0	0	UC0RPPR2	UC0RPPR1	UC0RPPR0
FFFFF18AH	UC0TIC	UC0TIF	UC0TMK	0	0	0	UC0TPPR2	UC0TPPR1	UC0TPPR20

22.3.5 中断屏蔽寄存器 0 至 3（IMR0 至IMR3）

IMR0 至 IMR3 寄存器用于设置可屏蔽中断的中断屏蔽状态。IMR0 至 IMR3 寄存器的 xxMKn 位等同于 xxICn.xxMKn 位。

IMRm 寄存器可以按 16 位宽度进行读取或写入（m＝0 至 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，则这些寄存器可以按字节或按位进行读写（m＝0 至 3）。

系统复位后，这些寄存器被设置为 FFFFH。

注意事项 设备文件将 xxICn.xxMKn 位定义为保留字。如果使用 xxMKn 的名称进行位操作，xxICn 寄存器的值而不是 IMRm 寄存器被改写（结果，IMRm 寄存器的值也被改写）。

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	UC0TMK	UC0RMK	UA4TMK	UA4RMK	UA3TMK	UA3RMK
	7	6	5	4	3	2	1	0
IMR3L	RTC1MK	WTMK/ RTC0MK	WTMK/ RTC2MK	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)	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)	TP5CCMK1	TP5CCMK0	TP5OVMK	TP4CCMK1	TP4CCMK0	TP4OVMK	TP3CCMK1/ UA5TMK	TP3CCMK0
	7	6	5	4	3	2	1	0
IMR1L	TP3OVMK/ UA5RMK	TP2CCMK1	TP2CCMK0	TP2OVMK	TP1CCMK1/ UFMK0	TP1CCMK0	TP1OVMK/ UFMK1	TP0CCMK1

复位后： FFFFH R/W 地址： IMR0 FFFFF100H,
IMR0L FFFFF100H, IMR0H FFFFF101H

	15	14	13	12	11	10	9	8
IMR0 (IMR0H注)	TP0CCMK0	TP0OVMK	TQ0CCMK3	TQ0CCMK2	TQ0CCMK1	TQ0CCMK0	TQ0OVMK	PMK7
	7	6	5	4	3	2	1	0
IMR0L	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

xxMKn	中断屏蔽标志的设置
0	使能中断服务
1	禁止中断服务

注 若要按字节或按位读取或写入 IMR0 至 IMR3 寄存器的第 8 位至第 15 位，需指定这些位作为 IMR0H 至 IMR3H 寄存器的第 0 位至第 7 位。

注意事项 将 IMR3 寄存器的第 14 位和第 15 位设置为 1。如果改变这些位的设置，操作将无法保证。

备注 xx： 每一个外设单元的标志名称（参见 **表 22-3 中断控制寄存器（xxICn）**）。
n ： 外设单元编号（参见 **表 22-3 中断控制寄存器（xxICn）**）。

22.3.6 正在服务中优先级寄存器（ISPR）

ISPR 寄存器保存当前响应的可屏蔽中断的优先级。当响应一个中断请求信号时，该寄存器中与中断请求信号优先级对应的位被置为 1，并且在中断服务期间一直保持置位。

当执行 RETI 指令时，对应最高优先级中断请求信号的位被硬件自动清 0。但是，从不可屏蔽中断服务或异常处理中返回时，并不被复位为 0。

该寄存器为只读寄存器，按字节或按位读取。

系统复位后，该寄存器被设为 00H。

注意事项 中断允许（EI）状态下，如果在读取 ISPR 寄存器的同时响应一个中断，读取的 ISPR 寄存器值可能是响应中断后其对应位被置位后的值。为了在响应中断前准确读出 ISPR 寄存器的值，应在中断禁止（DI）的状态下读取。

复位后: 00H　　R　　地址: FFFFF1FAH

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	当前正在应答中断的优先级
0	没有响应优先级为n的中断请求信号
1	正在响应优先级为n的中断请求信号

备注　n＝0 至 7（优先级等级）

22.3.7 ID标志

该标志用于存储有关使能或禁止中断请求信号的信息。中断禁止标志（ID）位于 PSW 内。

系统复位后，该标志被置为 1，PSW 寄存器被置为 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	可屏蔽中断服务说明注
0	可屏蔽中断请求信号应答使能
1	可屏蔽中断请求信号应答禁止（挂起）

注 中断禁止标志（ID）功能

该位由 DI 指令置位（为 1），由 EI 指令清除为 0。其值也可由 RETI 指令或向 PSW 写入数据的 LDSR 指令重写。

不管该标志如何，不可屏蔽中断请求信号和异常事件都可以得到响应。当一个可屏蔽中断请求信号被响应时，ID 标志由硬件自动置 1。

当 xxICn.xxIFn 位置为 1 时，在中断响应禁止期间（ID 标志＝1）产生的中断请求信号可以得到响应，并且 ID 标志被清为 0。

22.3.8 看门狗定时器模式寄存器 2（WDTM2）

该寄存器可以按字节或按位进行读写（详情参见 **第十二章 看门狗定时器 2 功能**）。

系统复位后，该寄存器被设置为 67H。

复位后: 67H R/W 地址: FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	0	0	0	0	0

WDM21	WDM20	看门狗定时器操作模式选择
0	0	停止操作
0	1	不可屏蔽中断请求模式
1	×	复位模式 (初始值)

22.4 软件异常

当 CPU 执行 TRAP 指令时，产生软件异常，并且总是能够得到响应。

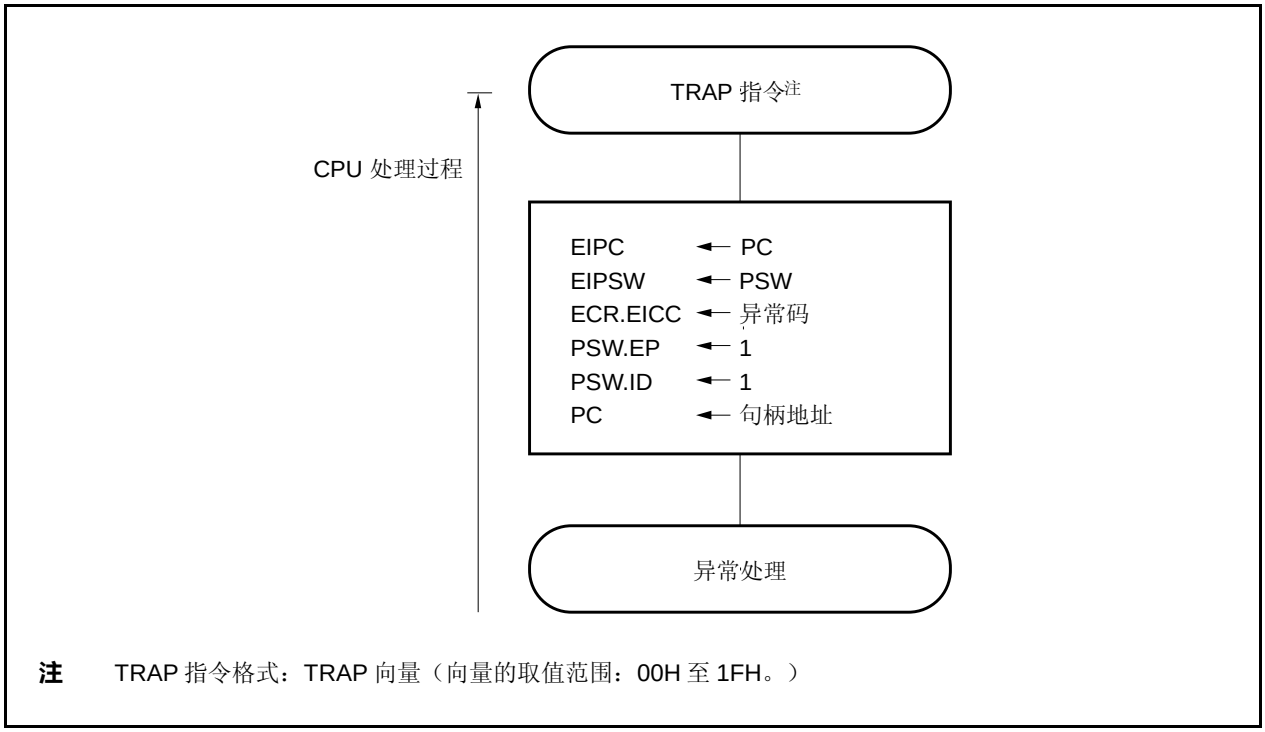
22.4.1 操作

如果发生软件异常，CPU 执行以下处理，并控制转移到句柄程序。

<1> 将当前 PC 保存至 EIPC。
<2> 将当前 PSW 保存至 EIPSW。
<3> 写入异常码到 ECR（中断源）的低 16 位（EICC）。
<4> 将 PSW.EP 和 PSW.ID 位置为 1。
<5> 将软件异常对应的句柄地址（00000040H 或 00000050H）设置到 PC 指针，并进行控制转移。

软件异常的处理过程如下所示：

图 22-9. 软件异常的处理过程



句柄地址由 TRAP 指令的操作数（向量）决定。如果向量为 00H 至 0FH，句柄地址是 00000040H，而如果向量为 10H 至 1FH，句柄地址是 00000050H。

22.4.2 恢复

使用 RETI 指令从软件异常处理中返回。

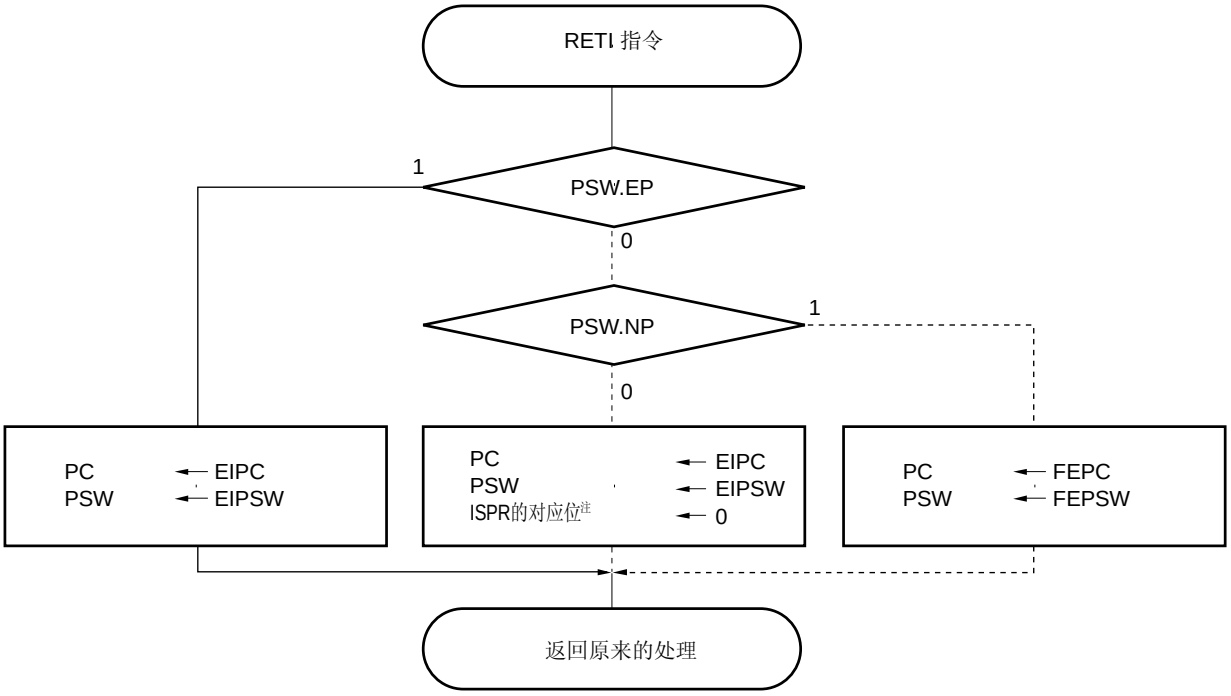
当执行 RETI 指令时，CPU 执行以下处理过程，并且控制转移到被恢复的 PC 地址。

<1> 由于 PSW.EP 位为 1，所以由 EIPC 和 EIPSW 分别载入 PC 和 PSW 的保存值。

<2> 控制转移回已存储的 PC 和 PSW 的地址。

RETI 指令的处理过程如下所示：

图 22-10. RETI 指令处理过程



注 ISPR 寄存器的详情，参见 **22.3.6 正在服务的优先级寄存器 (ISPR)。**

注意事项 在不可屏蔽中断服务期间，如果 EP 位和 NP 位被 LDSR 指令改变时，要在使用 RETI 指令返回时正确恢复 PC 和 PSW，必须在紧邻 RETI 指令之前，必须使用 LDSR 指令将 EP 位设置为 0，并必须将 NP 位置为 1。

备注 实线显示的是 CPU 的处理流程。

22.4.3 EP标志

EP 是一个状态标志，用于指示异常处理正在处理中。当异常发生时该标志被置位。

	31 … 8	7	6	5	4	3	2	1	0
PSW	0	NP	EP	ID	SAT	CY	OV	S	Z

EP	异常处理状态
0	异常处理未进行 (初始值)
1	异常处理进行中

22.5 异常陷阱

异常陷阱是一个中断，当非法执行指令时产生异常陷阱中断请求。在 V850ES/JG3-L 单片机中，非法操作码陷阱（ILGOP：非法操作码陷阱）用作一个异常陷阱。

22.5.1 非法操作码

非法操作码被定义为一条指令，其指令操作码（第 10 位至第 5 位）＝111111B，次操作码（第 26 位至第 23 位）＝0111B 至 1111B，另一个次操作码（第 16 位）＝0B。当执行这样一条指令时，就会产生一个异常陷阱。

15 – 11	10 – 5	4 – 0	31 – 27	26 – 23	22 – 16	
X X X X X	1 1 1 1 1 1	X X X X X	X X X X X	0 1 1 1 至 1 1 1 1	X X X X X X	0

×： 任意

注意事项 必须不要使用非法操作码，因为这些指令将来可能被指定为新用途。

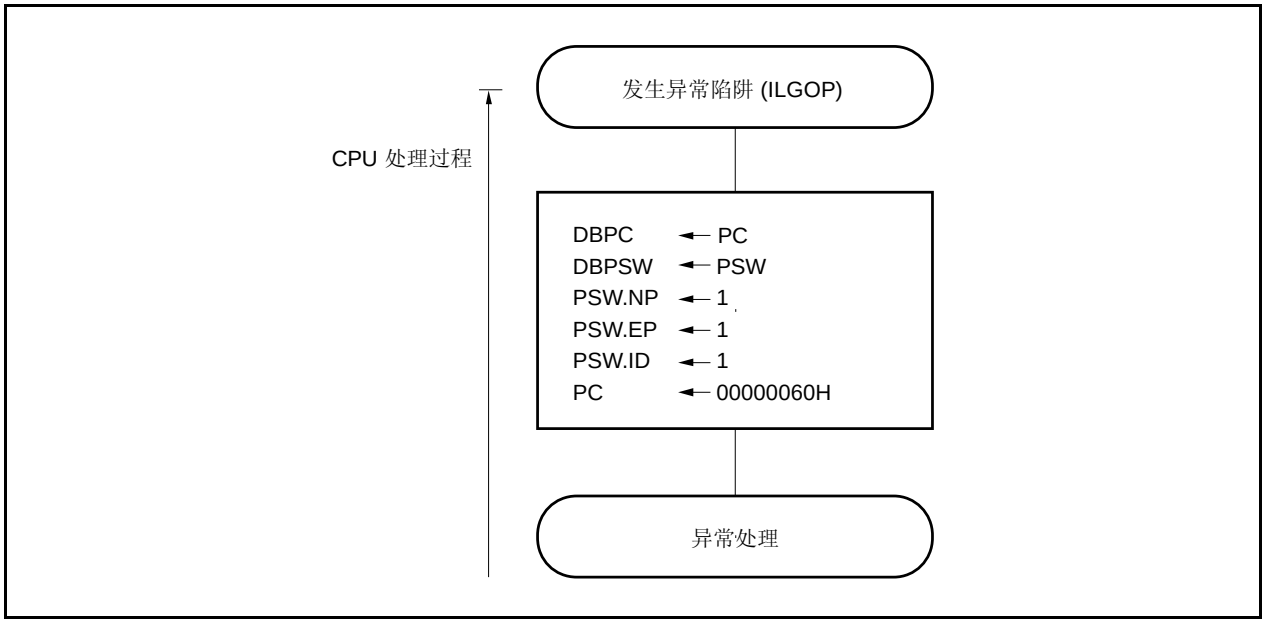
（1）操作

当发生异常陷阱时，CPU 执行以下处理，并将控制转移到句柄程序：

<1> 将当前 PC 保存至 DBPC。
<2> 将当前 PSW 保存至 DBPSW。
<3> 将 PSW.NP，PSW.EP 和 PSW.ID 位置为 1。
<4> 将异常陷阱对应的处理程序地址（00000060H）设置到 PC 指针，并进行控制转移。

异常陷阱的处理过程如下所示：

图 22-11. 异常陷阱处理过程



(2) 恢复

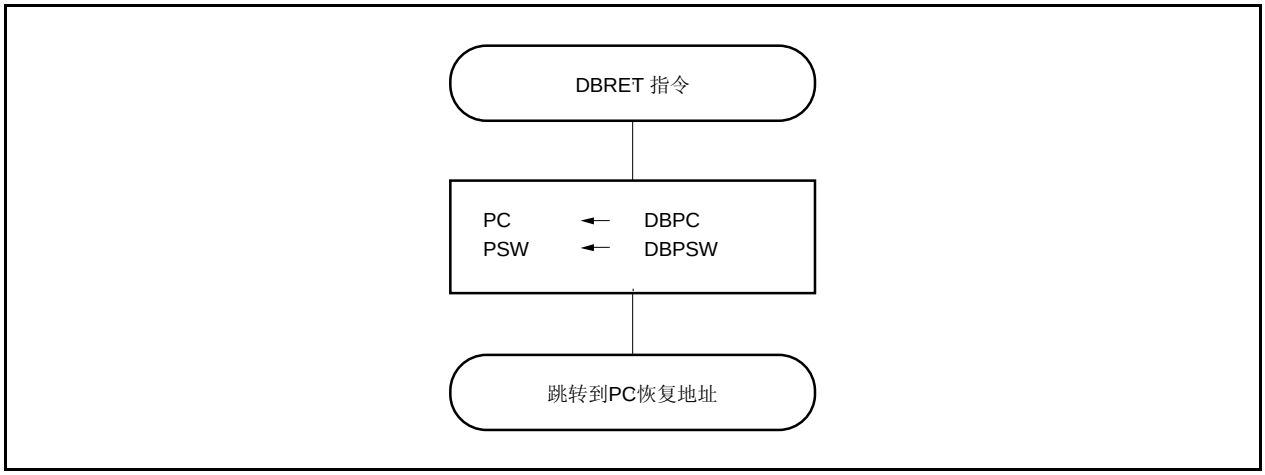
使用 DBRET 指令从异常陷阱中恢复执行。当执行 DBRET 指令时，CPU 执行下列处理并将控制转移至被恢复的 PC 地址。

<1> 从 DBPC 和 DBPSW 中载入 PC 和 PSW 的保存值。
<2> 控制转移回被恢复的 PC 和 PSW 所指定的地址。

注意事项 1. 只有在执行非法操作码和 DBRET 指令之间的间隔内，才可以访问 DBPC 和 DBPSW。
2. 如果执行非法操作码，指定默认设置或停止后续处理。

从异常陷阱中恢复的处理过程如下所示：

图 22-12. 从异常陷阱返回



22.5.2 调试陷阱

调试陷阱是一个异常，在执行 DBTRAP 指令时产生，总是可以得到响应。

（1）操作

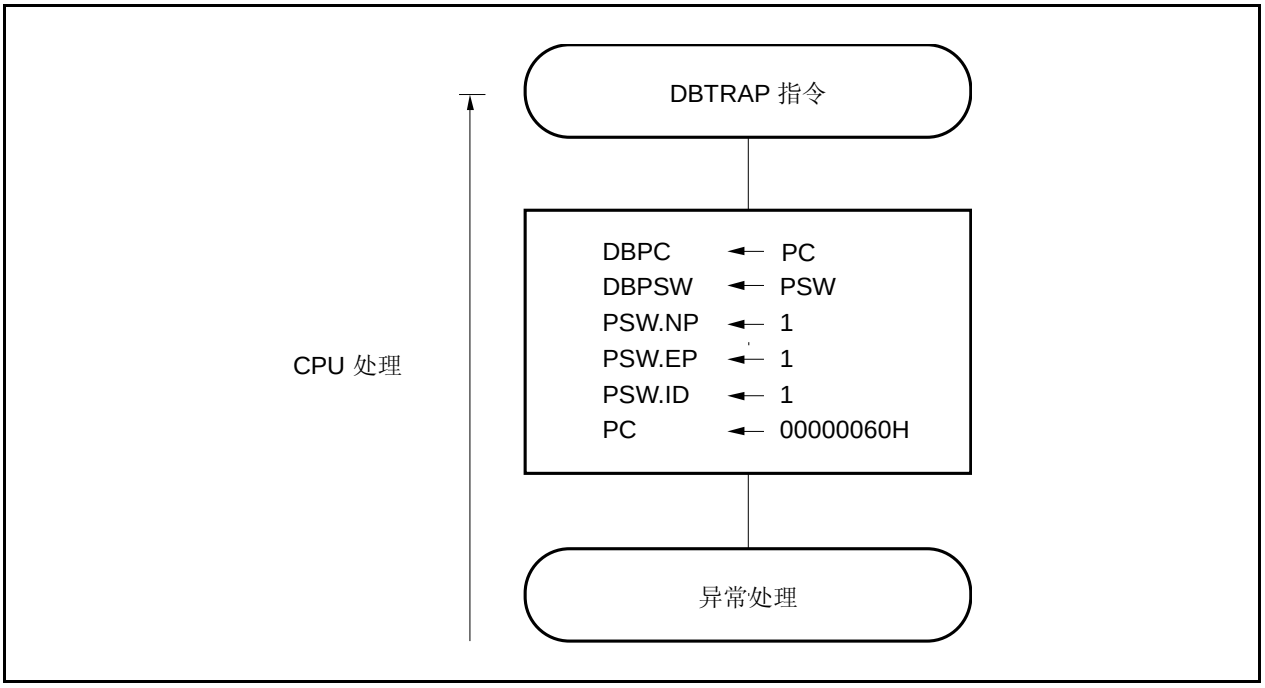
如果发生调试陷阱时，CPU 执行以下处理过程。

<1> 将当前 PC 保存至 DBPC。
<2> 将当前 PSW 保存至 DBPSW。
<3> 将 PSW.NP，PSW.EP 和 PSW.ID 位置为 1。
<4> 将调试陷阱的句柄地址（00000060H）设置到 PC 指针，并进行控制转移。

注意事项 DBTRAP 指令是为调试设计，基本上由调试工具使用。如果有调试工具执行该指令时，用户程序使用该指令，可能会发生误操作。

调试陷阱的处理过程如下所示：

图 22-13. 调试陷阱处理



（2）恢复

使用 DBRET 指令从异常陷阱中恢复执行。

当执行 DBRET 指令时，CPU 执行下列处理并将控制转移至被恢复的 PC 地址。

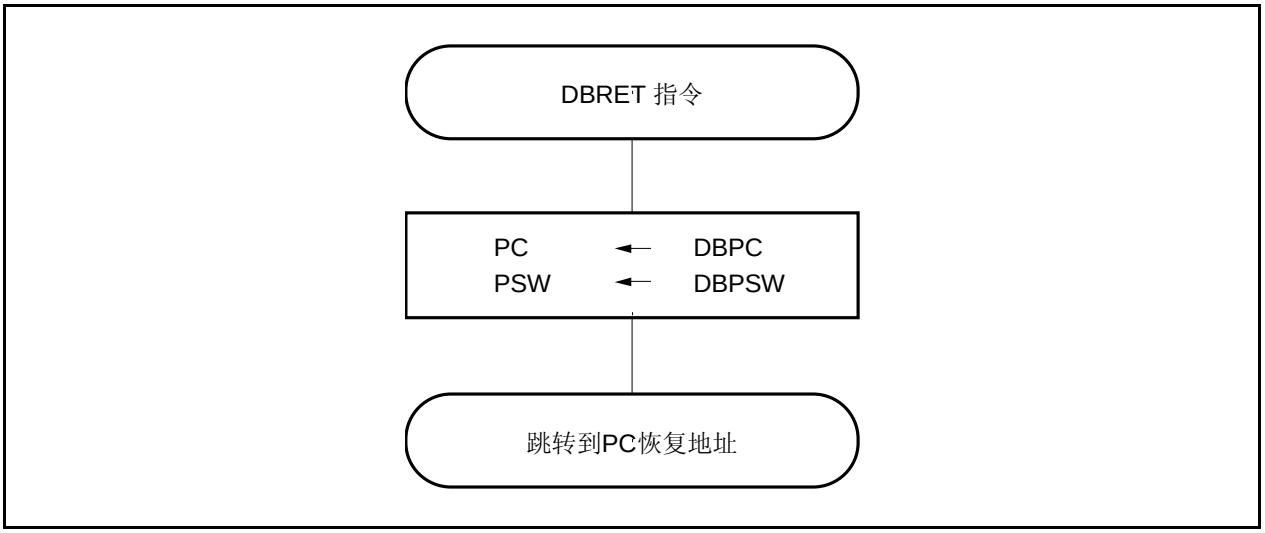
<1> 从 DBPC 和 DBPSW 中载入 PC 和 PSW 的保存值。

<2> 控制转移回被恢复的 PC 和 PSW 所指定的地址。

注意事项　　只有在执行 DBTRAP 指令和 DBRET 指令之间的间隔内，才可以访问 DBPC 和 DBPSW。

从调试陷阱中恢复的处理过程如下所示：

图 22-14. 从调试陷阱中返回



22.6 多重中断服务控制

在多重中断服务控制中，如果有较高优先级的中断请求信号产生，中断的服务会被停止。然后响应较高优先级的中断请求信号，并服务该中断。

在中断正服务时，如果有较低或相等优先级的中断请求信号产生，新产生的中断请求信号被挂起。

在中断使能 (PSW.ID 位 = 0)时，才能执行多重中断服务控制。即使在中断服务程序中，必须在中断使能(ID 位 = 0)状态下执行多重中断服务控制。如果在可屏蔽中断服务程序或软件异常服务程序中发生一个可屏蔽中断或软件异常，必须保存 EIPC 和 EIPSW。

下面是服务多重中断的过程示例。

(1) 在服务程序中响应可屏蔽中断请求信号

可屏蔽中断或异常的服务程序

…
…
- EIPC 保存到存储器或寄存器
- EIPSW 保存到存储器或寄存器
- EI 指令（使能中断响应）

…
…
…
…
- DI 指令（禁止中断响应）
- 保存的值恢复到 EIPSW
- 保存的值恢复到 EIPC
- RETI 指令

←响应可屏蔽中断

22.7 外部中断请求输入引脚（NMI和INTP0 至INTP7）

22.7.1 噪声消除

（1）NMI 引脚的噪声消除

NMI 引脚具有使用模拟延迟（若干 10ns）的内部噪声消除电路。因此，输入 NMI 引脚的信号需要保持输入电平一段特定时间，否则将不会被检测为边沿。因此，只有经过一段特定时间之后，才能检测到边沿。

NMI 引脚可以用来解除 STOP 模式。在 STOP 模式下，使用系统时钟无法消除噪声，因为此时内部系统时钟是停止的。

（2）INTP0 至 INTP7 引脚的噪声消除

INTP0 至 INTP7 引脚具有使用模拟延迟（若干 10ns）的内部噪声消除电路。因此，输入到各个引脚的信号需要保持输入电平一段特定时间，否则将不会被检测为边沿。因此，只有经过一段特定时间之后，才能检测到边沿。

（3）INTP3 引脚的噪声消除

INTP3 引脚具有内部数字/模拟噪声消除电路，使用 NFC.NFEN 位可以选择数字或模拟噪声消除电路（模拟延迟：若干 10ns）。

采样时钟可以使用 NFC.NFC2 至 NFC.NFC0 位选择 f_{XX}/64, f_{XX}/128, f_{XX}/256, f_{XX}/512, f_{XX}/1,024, 或 f_{XT} 。如果采样时钟被设置为 f_{XX}/64, f_{XX}/128, f_{XX}/256, f_{XX}/512, 或 f_{XX}/1,024, 在 IDLE 或 STOP 模式下，采样时钟停止。因此，不能用它来解除待机模式。要解除待机模式，应选择 f_{XT} 作为采样时钟或选择模拟噪声消除电路

22.7.2 边沿检测

NMI 和 INTP0 至 INTP7 各个引脚输入信号的有效边沿可以从以下 4 种类型选择：

- 上升沿
- 下降沿
- 上升和下降双边沿
- 无边沿检测

注意事项 NMI 引脚可以复用作为 P02 引脚。在复位之后作为普通端口引脚功能。要使能 NMI 引脚功能，使用 PMC0 寄存器。NMI 引脚的初始设置是“无边沿检测”。通过 INTF0 和 INTR0 寄存器来选择 NMI 引脚有效边沿。

（1）外部中断下降沿和上升沿指定寄存器 0（INTF0，INTR0）

INTF0 和 INTR0 寄存器是 8 位寄存器，第 2 位用于指定 NMI 引脚的边沿检测，第 3 位至第 6 位用于指定外部中断引脚（INTP0 至 INTP3）的下降沿和上升沿检测。

这些寄存器可以按字节或按位进行读写。

系统复位后，这些寄存器被设置为 00H。

注意事项 当引脚功能由端口功能改变为外部中断功能（复用功能）时，可能会检测到一个边沿。因此，将 INTF0n 和 INTR0n 位清为 00，然后再指定外部中断功能（PMC0.PMC0n 位 = 1）。

当引脚功能由外部中断功能改变为端口功能时，也可能会检测到一个边沿。因此，将 INTF0n 和 INTR0n 位清为 00，然后再指定端口功能（PMC0.PMC0n 位 = 0）。

复位后：00H　R/W　地址：INTF0 FFFFFC00H, INTR0 FFFFFC20H

	7	6	5	4	3	2	1	0
INTF0	0	INTF06	INTF05	INTF04	INTF03	INTF02	0	0
		INTP3	INTP2	INTP1	INTP0	NMI		

	7	6	5	4	3	2	1	0
INTR0	0	INTR06	INTR05	INTR04	INTR03	INTR02	0	0
		INTP3	INTP2	INTP1	INTP0	NMI		

备注 关于如何指定有效边沿，参见 **表 22-4。**

表 22-4. 有效边沿设定

INTF0n	INTR0n	有效边沿设定（n＝2 至 6）
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升和下降双边沿

注意事项 当这些寄存器不用于设置 NMI 或 INTP0 至 INTP3 引脚时，请确保将 INTF0n 和 INTR0n 位清为 00。

备注 n＝2：　NMI 引脚的控制

n＝3 至 6：INTP0 至 INTP3 引脚的控制

（2）外部中断下降沿和上升沿指定寄存器 3（INTF3，INTR3）

INTF3 和 INTR3 寄存器是 8 位寄存器，用于指定外部中断引脚（INTP7）的下降沿和上升沿检测。

这些寄存器可以按字节或按位进行读写。

系统复位后，这些寄存器被设置为 00H。

注意事项 1. 当引脚功能由端口功能改变为外部中断功能（复用功能）时，可能会检测到一个边沿。因此，将 INTF31 和 INTR31 位清为 00，然后再指定外部中断功能（PMC3.PMC31 位 = 1）。
当引脚功能由外部中断功能改变为端口功能时，也可能会检测到一个边沿。所以，将 INTF31 和 INTR31 位清为 00，然后再设置端口功能（PMC3.PMC31 位 = 0）。

2. INTP7 引脚和 RXDA0 引脚为复用功能引脚。当此引脚用作 RXDA0 时，禁止对复用功能引脚 INTP7 的边沿检测（将 INTF3.INTF31 和 INRT3.INTR31 位清为 0）。当此引脚用作 INTP7，停止 UARTA0 接收（将 UA0CTL0.UA0RXE 位清为 0）。

复位后：00H　R/W　地址：INTF3 FFFFFC06H, INTR3 FFFFFC26H

	7	6	5	4	3	2	1	0
INTF3	0	0	0	0	0	0	INTF31	0
							INTP7	

	7	6	5	4	3	2	1	0
INTR3	0	0	0	0	0	0	INTR31	0
							INTP7	

备注　关于如何指定有效边沿，参见 **表 22-5。**

表 22-5. 有效边沿设定

INTF31	INTR31	有效边沿指定
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升和下降双边沿

注意事项 当这些寄存器不用于设置 INTP7 引脚时，请确保将 INTF31 和 INTR31 位清为 00。

（3）外部中断下降沿和上升沿指定寄存器 9H（INTF9H，INTR9H）

INTF9H 和 INTR9H 寄存器是 8 位寄存器，用于指定外部中断引脚（INTP4 至 INTP6）的下降沿和上升沿检测。
这些寄存器可以按字节或按位进行读写。
系统复位后，这些寄存器被设置为 00H。

注意事项 **当引脚功能由端口功能改变为外部中断功能（复用功能）时，可能会检测到一个边沿。因此，将 INTF9n 和 INTR9n 位清为 00，然后再指定外部中断功能（PMC9.PMC9n 位 = 1）。**
当引脚功能由外部中断功能改变为端口功能时，也可能会检测到一个边沿。所以，将 INTF9n 和 INTR9n 位清为 00，然后再设置端口功能（PMC9.PMC9n 位 = 0）。

复位后：00H　R/W　地址：INTF9H FFFFFC13H, INTR9H FFFFFC33H

	15	14	13	12	11	10	9	8
INTF9H	INTF915	INTF914	INTF913	0	0	0	0	0
	INTP6	INTP5	INTP4					

	15	14	13	12	11	10	9	8
INTR9H	INTR915	INTR914	INTR913	0	0	0	0	0
	INTP6	INTP5	INTP4					

备注 关于如何指定有效边沿，参见 **表 22-6。**

表 22-6. 有效边沿设定

INTF9n	INTR9n	有效边沿设定（n＝13 至 15）
0	0	无边沿检测
0	1	上升沿
1	0	下降沿
1	1	上升和下降双边沿

注意事项 **当这些寄存器不用于设置 INTP4 至 INTP6 引脚时，请确保将 INTF9n 和 INTR9n 位清为 00。**

备注 n＝13 至 15：INTP4 至 INTP6 引脚的控制

（4）噪声消除控制寄存器（NFC）

INTP3 引脚具有一个数字噪声消除器。使用 NFC 寄存器来执行噪声消除的设置。

当选择数字噪声消除时，数字采样的采样时钟可从以下范围选择：f_{XX}/64，f_{XX}/128，f_{XX}/256，f_{XX}/512，f_{XX}/1,024 或 f_{XT}。需要执行 3 次采样。

即使选择数字噪声消除，使用 f_{XT} 作为采样时钟，也可以使用 INTP3 中断请求信号来释放 IDLE1，IDLE2 和 STOP 模式。

该寄存器可以按字节进行读取或写入。

系统复位后，该寄存器被设为 00H。

注意事项 在采样时钟改变后，需要 3 个采样时钟周期来初始化数字噪声消除器。因此，如果在采样时钟改变后，有 INTP3 有效沿在这 3 个采样时钟周期内输入，可能会产生一个中断请求信号。因此，当使用中断和 DMA 功能时，注意以下几点。

- **当使用中断功能时，经过 3 个采样时钟周期后，中断请求标志（PIC3.PIF3 位）清零后再使能中断。**
- **当使用 DMA 功能（由 INTP3 启动）时，在 3 个采样时钟周期过后使能 DMA。**

复位后：00H　R/W　地址：FFFFF318H

	7	6	5	4	3	2	1	0
NFC	NFEN	0	0	0	0	NFC2	NFC1	NFC0

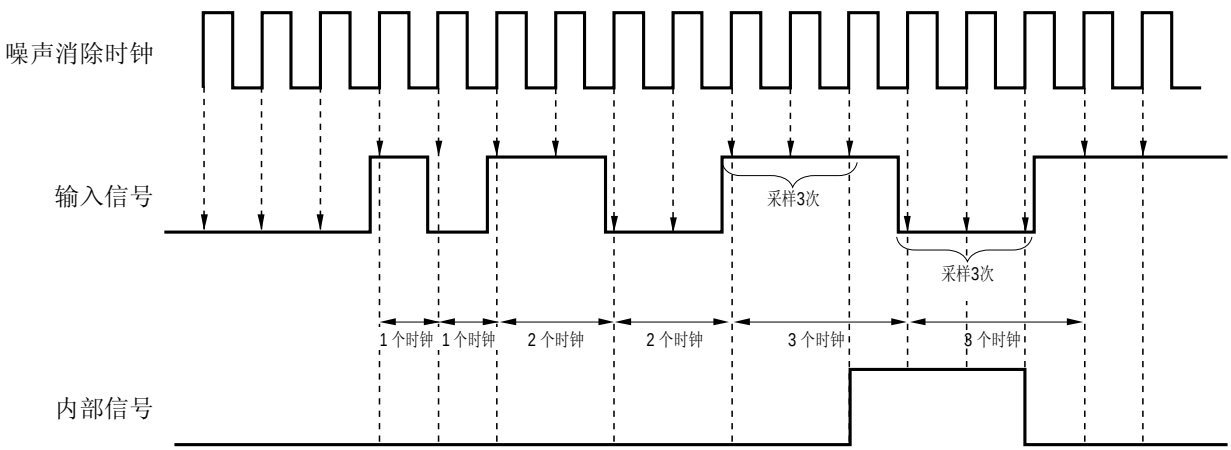
NFEN	INTP3 引脚噪声消除设置
0	模拟噪声消除 (60 ns(典型值))
1	数字噪声消除

NFC2	NFC1	NFC0	数字采样时钟
0	0	0	f_{XX}/64
0	0	1	f_{XX}/128
0	1	0	f_{XX}/256
0	1	1	f_{XX}/512
1	0	0	f_{XX}/1,024
1	0	1	f_{XT} (副时钟)
其它			设置禁止

备注
1. 因为采样执行了 3 次，则可靠消除噪音需要 2 个采样时钟周期。
2. 当噪声宽度小于 2 个采样时钟周期时，如果噪声与采样时钟的输入同步，就会产生一个中断请求信号。

通过定时器 T 输入引脚数字滤波器执行数字噪声消除的时序示例如图 22-15 所示。

图 22-15. 数字噪声消除时序的示例



备注 当 INTP3 输入信号为高电平（或低电平）时，如果噪声消除时钟周期采样了不足两次，会判断认为输入信号是噪声并将其消除。如果噪声消除时钟周期采样了三次以上，信号的边沿被检测认为有效输入。

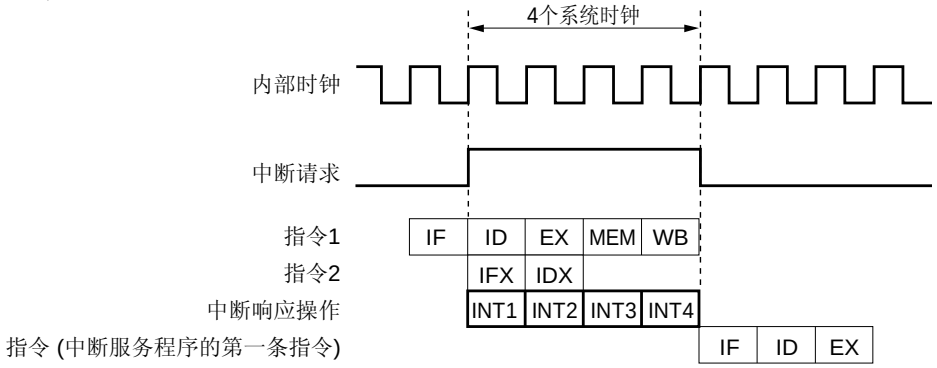
22.8 CPU中断响应时间

除了下列情况外，CPU 的中断响应时间最小为 4 个时钟周期。若要连续输入中断请求信号，至少需要在前一中断之后 5 个时钟周期才能输入下一中断请求信号。

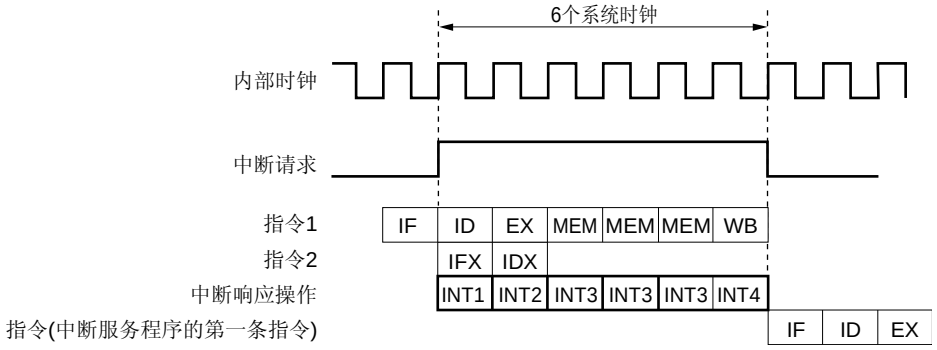
- IDLE1/IDLE2/STOP 模式下
- 当访问外部总线时
- 当中断请求不采样指令被连续执行时（参见 **22.9 中断不被 CPU 响应期间**）。
- 当访问中断控制寄存器时

图 22-16. 中断请求信号响应的流水操作（概述）

（1） 最小中断响应时间



（2） 最大中断响应时间



备注 INT1 至 INT4：中断响应处理
IFX： 无效指令取指
IDX： 无效指令解码

中断响应时间（内部系统时钟周期）			状态
	内部中断	外部中断	
最小	4	4＋ 模拟延迟时间	以下情况除外： • IDLE1/IDLE2/STOP 模式下 • 外部总线访问 • 2 个或更多个中断请求不采样指令连续执行 • 访问外设 I/O 寄存器
最大	6	6＋ 模拟延迟时间	

22.9 中断不被CPU响应期间

当指令正在执行时，CPU 可以响应中断。但是，在中断请求不采样指令和下一条指令之间不会响应中断（中断挂起）。中断请求不采样指令如下所示：

- EI 指令
- DI 指令
- LDSR reg2，0x5 指令（对于 PSW）
- PRCMD 寄存器的存储指令
- 对下列寄存器的存储指令和 SET1，NOT1，CLR1 操作指令
  - 中断相关的寄存器：
    中断控制寄存器（xxICn），中断屏蔽寄存器 0 至 3（IMR0 至 IMR3）
  - 节电控制寄存器（PSC）
  - 片上调试模式寄存器（OCDM）

备注 **1.** xx：每一个外设单元的标志名称（参见 **表 22-3 中断控制寄存器（xxICn）**）。
n ： 外设单元编号（参见 **表 22-3 中断控制寄存器（xxICn）**）。
2. 流水线操作的详情，参见 **V850ES 架构用户手册 (U15943E)**。

22.10注意事项

22.10.1 被恢复的PC

当中断服务开始时，程序计数器（PC）的值被存入 EIPC、FEPC 或 DBPC。在执行以下任意一条指令时，如果程序响应一个不可屏蔽中断或可屏蔽中断，那条指令的执行停止，并在中断服务完成之后恢复执行。

- 载入指令（SLD.B，SLD.BU，SLD.H，SLD.HU，SLD.W）。
- 除法指令（DIV，DIVH，DIVU，DIVHU）。
- PREPARE，DISPOSE 指令（仅当在堆栈指针被更新前产生中断）。

第二十三章 按键中断功能

23.1 功能

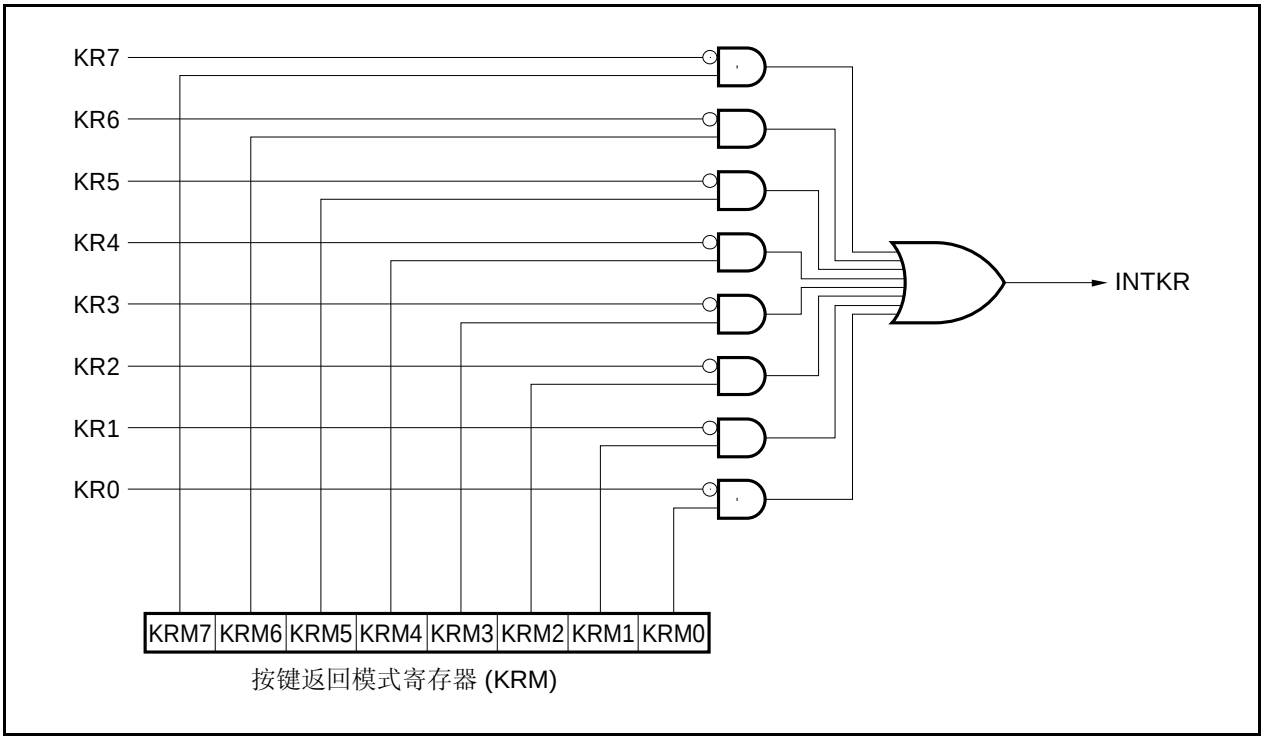
通过设置 KRM 寄存器，向 8 个按键输入引脚（KR0 至 KR7）输入一个下降沿，可以产生一个按键中断请求信号（INTKR）。

注意事项 如果 KR0 至 KR7 引脚中有任意一个为低电平，即使有下降沿输入到另一个引脚也不会产生 INTKR 信号。

表 23-1. 标志分配

标志	引脚说明	复用功能
KRM0	控制 KR0 信号	P50
KRM1	控制 KR1 信号	P51
KRM2	控制 KR2 信号	P52
KRM3	控制 KR3 信号	P53
KRM4	控制 KR4 信号	P54
KRM5	控制 KR5 信号	P55
KRM6	控制 KR6 信号	P90
KRM7	控制 KR7 信号	P91

图 23-1. 按键返回框图



23.2 引脚功能

用作按键中断的按键输入引脚也可以用作**表 23-2** 中的其它功能。要将这些引脚用作按键中断，必须通过相关的寄存器设置来指定（参见**表 4-15 端口引脚用作复用功能时的设置**）。

表 23-2. 引脚功能

引脚编号	端口功能	按键输入功能	其它功能
37	P50	KR0	P50/TIQ01/TOQ01/RTP00
38	P51	KR1	P51/TIQ02/TOQ02/RTP01
39	P52	KR2	P52/TIQ03/TOQ03/RTP02/DDI
40	P53	KR3	P53/SIB2/TIQ00/TOQ00/RTP03/DDO
41	P54	KR4	P54/SOB2/RTP04/DCK
42	P55	KR5	P55/$\overline{\text{SCKB2}}$/RTP05/DMS
61	P90	KR6	P90/A0/TXDA1/SDA02
62	P91	KR7	P91/A1/RXDA1/SCL02

23.3 寄存器

（1）按键返回模式寄存器（KRM）

KRM 寄存器使用 KRM0 至 KRM7 位来控制 KR0 至 KR7 信号。

该寄存器可以按字节或按位读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H　R/W　地址: FFFFF300H

	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	按键返回模式控制
0	禁止检测按键返回信号
1	检测按键返回信号

注意事项 在重写 KRM 寄存器之前，将其清除为 00H。

23.4 注意事项

（1）如果 KR0 至 KR7 中有任意某个引脚输入低电平，即使有下降沿输入其它引脚，也不会产生 INTKR 信号。

（2）RXDA1 和 KR7 引脚不能同时使用。使用 RXDA1 引脚时，不要使用 KR7 引脚。若使用 KR7 引脚，就不要使用 RXDA1 引脚（建议将 PFC91 位设置为 1，并清除 PFCE91 位为 0）。

（3）如果改变 KRM 寄存器，就可能产生一个中断请求信号（INTKR）。为防止此种情况，在禁止中断（DI）或屏蔽中断之后再改变 KRM 寄存器，然后，将中断请求标志（KRIC.KRIF 位）清零后，使能中断（EI）或取消中断的屏蔽。

（4）若要使用按键中断功能，需确保将端口引脚设置为“按键返回引脚”，然后使用 KRM 寄存器使能按键中断功能。若要从按键返回引脚切换至端口引脚，需要使用 KRM 寄存器禁止按键中断功能，然后再设置引脚功能为“端口引脚”。

第二十四章 待机功能

24.1 概述

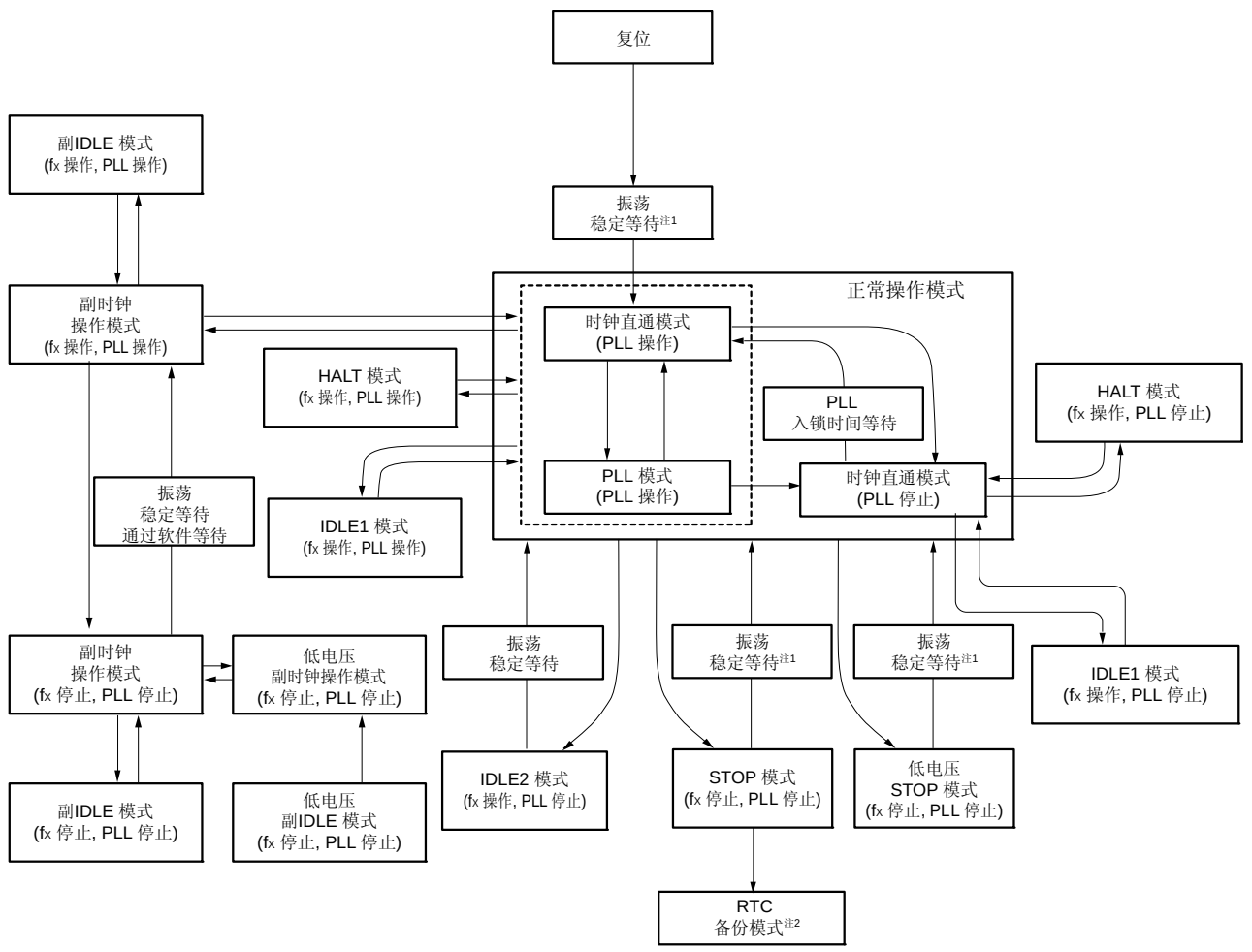
通过结合运用待机模式并为应用程序选择恰当的模式，可以有效减少系统的功耗。表 24-1 列出了可用待机模式：

表 24-1. 待机模式

模式	功能概述
HALT 模式	此模式下，仅停止 CPU 的工作时钟。 为了间歇操作，通过该模式与正常工作模式的组合，可以降低系统的总体电流消耗。
IDLE1 模式	此模式下，除振荡器，PLL [注]和 flash 存储器之外，停止其它所有内部电路的操作。 因为该模式停止了片上外设功能的操作，可以将功耗降低到比 HALT 模式更低的水准。
IDLE2 模式	此模式下，除振荡器外，停止其它所有内部电路的操作。 因为该模式停止了片上外设功能、PLL 和 flash 存储器的操作，可以将功耗降低到比 IDLE1 模式更低的水准。
STOP 模式	此模式下，除副时钟振荡器外，停止其它所有内部电路的操作。 该模式可以将功耗降低到比 IDLE2 模式更低的水准。 有两种模式可用：STOP 模式和低电压 STOP 模式。 因为稳压器的电压被降低，低电压 STOP 模式可以进一步降低功耗。
副时钟工作模式	此模式下，副时钟用作内部系统时钟。 该模式可以将功耗降低到比正常工作模式更低的水准。 有两种模式可用：副时钟操作模式和低电压副时钟操作模式。 因为稳压器的电压被降低，低电压副时钟工作模式可以进一步降低功耗。
副 IDLE 模式	此模式下，除振荡器，PLL 操作[注]和 flash 存储器之外，停止其它所有内部电路的操作，副时钟用作内部系统时钟。 该模式可以将功耗降低到比副时钟工作模式更低的水准。 有两种模式可用：副 IDLE 模式和低电压副 IDLE 模式。 因为稳压器的电压被降低，低电压副 IDLE 模式可以进一步降低功耗。
RTC 备份模式	在该模式下，当 V_{DD} 跌落到低于工作电压时，但副时钟振荡器和 RTC 单独来自于其他内部电路，RTC 基于供给到 RV_{DD} 引脚的备份电压，继续对副时钟计数。该模式下的功耗可以比低电压 STOP 模式更低。 请注意，在 RTC 备份模式下，内部 RAM 的数据和 CPU 寄存器的值无法保持，当系统从该模式恢复时，请确保在重新供给 V_{DD} 之后停止复位信号输入。

注 在 IDLE1 或副 IDLE 模式下，PLL 保持模式转换之前的瞬时操作状态。如果 PLL 操作不是必需的，停止 PLL 以降低功率消耗。在 IDLE2 模式下，模式转换会使 PLL 自动停止。

图 24-1. 状态转换



注 **1.** 在振荡稳定期间，如果发生一个 WDT 溢出，则 CPU 运行于内部振荡时钟。

2. 系统进入 RTC 备份模式，只要满足下列两个条件，RTC 备份区域的接口被分离：

V_{DD} 低于操作电压范围

备份电源供给到 RV_{DD}

备注 fx：主时钟振荡频率

24.2 寄存器

（1）节电控制寄存器（PSC）

PSC 寄存器是 8 位寄存器，用于控制待机功能。该寄存器的 STP 位用于指定待机模式。该寄存器是专用寄存器，只能通过特定的序列组合来写入（参见 **3.4.7 专用寄存器**）。

该寄存器可以按字节或位进行读写。

系统复位后，该寄存器被设置为 00H。

复位后：00H　　R/W　　地址：FFFFF1FEH

	7	<6>	<5>	<4>	3	2	<1>	0
PSC	0	NMI1M	NMI0M	INTM	0	0	STP	0

NMI1M	INTWDT2信号发生时解除待机模式的控制
0	允许通过INTWDT2信号解除待机模式
1	禁止通过INTWDT2信号解除待机模式

NMI0M	通过 NMI 引脚输入控制待机模式解除
0	NMI引脚输入有效时解除待机模式
1	NMI引脚输入无效时解除待机模式

INTM	通过可屏蔽中断请求信号控制待机模式的解除
0	可屏蔽中断请求信号有效时解除待机模式
1	可屏蔽中断请求信号无效时解除待机模式

STP	待机模式注设置
0	正常模式
1	待机模式

注　可以由 STP 位设置的待机模式有 IDLE1，IDLE2，STOP 或副 IDLE 模式。

注意事项
1. **设置某种待机模式（不包括 HALT 模式）之前，使用 PSMR.PSM1 和 PSMR.PSM0 位指定模式，然后再设置 STP 位。**
2. **HALT 模式解除时，NMI1M，NMI0M，和 INTM 位的设置无效。**
3. **如果在 NMI1M，NMI0M，或 INTM 位设为 1 的同时把 STP 位也设为 1，则对 NMI1M，NMI0M，或 INTM 位的设置无效。如果在设置 IDLE1/IDLE2/STOP 模式时有未屏蔽中断请求信号处于挂起状态，则应设置中断请求信号对应的位（NMI1M，NMI0M，或 INTM）为 1，然后再设置 STP 位为 1。**

（2）节电模式寄存器（PSMR）

PSMR 寄存器是 8 位寄存器，用于控制节电模式下的操作状态及时钟操作。

该寄存器可按字节或按位进行读写。

复位输入将该寄存器设置为 00H。

复位后：00H　　R/W　　地址：FFFFF820H

	7	6	5	4	3	2	<1>	<0>
PSMR	0	0	0	0	0	0	PSM1	PSM0

PSM1	PSM0	软件待机模式下的操作说明
0	0	IDLE1，副IDLE 模式
0	1	STOP 模式
1	0	IDLE2，副IDLE 模式
1	1	STOP 模式

注意事项 1. 请确保将第 2 位至第 7 位清为“0”。

2. 仅当 PSC.STP 位为 1 时，PSM0 和 PSM1 位才有效。

备注 IDLE1：此模式下，除振荡器和其它一些电路（flash 存储器和 PLL）之外，停止所有的操作。
在解除 IDLE1 模式后，无需经过振荡稳定保证时间，就可以恢复正常操作模式。这和 HALT 模式一样。

IDLE2：此模式下，除振荡器之外，所有的操作停止。
解除 IDLE2 模式后，在经过 OSTS 寄存器指定的振荡稳定时间后，恢复正常操作模式。

STOP：此模式下，除副时钟振荡器之外，所有的操作停止。
解除 STOP 模式后，在经过 OSTS 寄存器指定的振荡稳定时间后，恢复正常操作模式。

副 IDLE：此模式下，除振荡器之外，暂停所有的操作。在中断请求信号解除 IDLE 模式后，经过 12 个副时钟周期的保证时间后，恢复副时钟操作模式。

（3）振荡稳定时间选择寄存器（OSTS）

在解除 STOP 模式之后至振荡稳定之前需要等待的时间，或在解除 IDLE2 模式之后至内部 flash 存储器稳定之前需要等待的时间，由 OSTS 寄存器控制。

OSTS 寄存器可以按字节进行读写。

复位输入将寄存器设置为 06H。

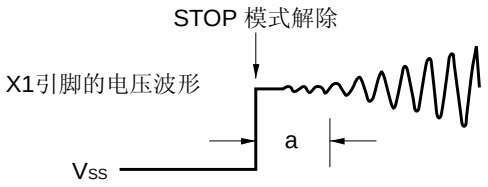
复位后：06H　R/W　地址：FFFFF6C0H

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	振荡稳定时间/建立时间的选择注		
				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　当解除 STOP 模式和 IDLE2 模式时，需要不同的振荡稳定时间和建立时间。

注意事项　1. 不管是因为复位输入还是因为中断请求信号解除了 STOP 模式，在解除 STOP 模式之后的等待时间不包括时钟开始振荡之前（下图的“a”）的那段时间。



2. 请确保将第 3 位至第 7 位清为“0”。

3. 复位解除后的振荡稳定时间会根据选项字节的设置而有所不同。详情参见第三十章选项字节。

备注　fx＝主时钟振荡器频率

（4）稳压器保护寄存器（REGPR）

REGPR 寄存器用于保护稳压器输出电压电平控制寄存器 0（REGOVL0），这样非法数据就不能写入 REGOVL0。除非允许数据（C9H）被写入 REGPR 寄存器，数据都无法写入 REGOVL0 寄存器。只有两种类型的数据可以写入 REGPR 寄存器，C9H（使能数据）和 00H（保护数据）。写入其它任何值都是被禁止的。（如果 C9H 或 00H 之外的值写入 REGPR 寄存器，设置的写入值被禁止写入访问 REGOVL0 寄存器，但是操作将无法保证。）

该寄存器只能按字节进行读写（禁止按位进行访问）。

系统复位后，该寄存器被设为 00H（保护数据状态）。

复位后: 00H　R/W　地址: FFFFF331H

	7	6	5	4	3	2	1	0
REGPR	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0

- 保护数据状态：REGPR＝00H
  在此状态下，可以防止对REGOVL0寄存器的非法写入访问。在保护数据状态，即使企图写入，值也无法写入到REGOVL0寄存器，而且REGOVL0寄存器保持原值。
  除了需要改变REGOVL0寄存器值的时候，请确保将REGPR设置为00H，以避免意外的误操作。

- 允许数据状态：REGPR＝C9H
  在此状态下，允许对REGOVL0寄存器的写入访问。
  - 从正常模式→低电压 STOP 模式的转换
    参见 24.6.1 设置和操作状态。
  - 副时钟操作模式→低电压副时钟操作模式的转换
    参见24.7.1 设置和操作状态。
  - 副时钟操作模式→低电压副IDLE模式的转换。
    参见24.8.1 设置和操作状态。

（5）稳压器输出电压电平控制寄存器 0（REGOVL0）

该寄存器用于选择低电压 STOP 模式，低电压副时钟操作模式或低电压副 IDLE 模式。可以通过降低稳压器的输出电压来减少功率消耗。

该寄存器只能按字节进行读写（禁止按位进行访问）。

系统复位后，该寄存器被设为 00H。

该寄存器必须与稳压器保护寄存器（REGPR）配合写入。

复位后：00H　R/W　地址：FFFFF332H

	7	6	5	4	3	2	1	0
REGOVL0	0	0	0	0	0	0	SUBMD	STPMD

SUBMD	副时钟操作模式/副IDLE模式中稳压器输出模式的选择
0	副时钟操作模式/副IDLE 模式
1	低电压副时钟操作模式/低电压 副IDLE模式

STPMD	STOP模式中稳压器输出模式的选择
0	STOP 模式
1	低电压 STOP 模式

- **REGOVL0 寄存器的写入操作**

仅当C9H写入REGPR寄存器时，允许写入REGOVL0寄存器（参见 **24.2（4） 稳压器保护寄存器（REGPR）**）。

该寄存器只能被设置为00H，01H，和02H。

禁止设置03H。如果设置为03H，操作将无法保证。

- **REGOVL0寄存器的读取操作**

REGOVL0寄存器的默认值为00H。有值被按照正确流程写入该寄存器之后，读取写入值。读取该寄存器的流程没有限制。

注
- 从正常模式→低电压 STOP 模式的转换
  参见 24.6.1 设置和操作状态。
- 副时钟操作模式→低电压副时钟操作模式的转换
  参见 24.7.1 设置和操作状态。
- 副时钟操作模式→低电压副 IDLE 模式的转换。
  参见 24.8.1 设置和操作状态。

注意事项 当设置低电压副时钟模式和低电压副 IDLE 模式时，请确保停止主时钟和 PLL。

24.3 HALT模式

24.3.1 设置和操作状态

在正常操作模式下，执行一条专用指令（HALT），可以设置为 HALT 模式。

在 HALT 模式下，时钟振荡器继续工作。只是停止 CPU 的时钟供给；其它片上外设功能的时钟不受影响。

结果，程序停止执行，内部 RAM 保持设置为 HALT 模式之前的内容。独立于 CPU 指令处理的片上外设功能继续工作。

表 24-3 显示了 HALT 模式下的操作状态。

通过 HALT 模式和正常操作模式的组合操作，可以减少系统的平均电流消耗。

注意事项 1. 在 HALT 指令后，插入五个或更多的 NOP 指令。

2. 如果在执行 HALT 指令时有一个未屏蔽的中断请求信号被挂起，则状态切换至 HALT 模式，但是 HALT 模式立即被该中断请求解除。

24.3.2 解除HALT模式

在 HALT 模式下，有多种方式可以解除 HALT 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽的外部中断请求信号（INTP0 至 INTP7 引脚输入）、来自 HALT 模式下可操作外设的未屏蔽的内部中断请求信号，或复位信号（由 RESET 引脚输入的复位、WDT2RES 信号，低电压检测器（LVI），或时钟监视器（CLM））。

HALT 模式释放后，恢复为正常操作模式。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号解除 HALT 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号，可以解除 HALT 模式，且与中断请求信号的优先级无关。但是，如果在一个中断服务程序中设置 HALT 模式，则之后发生的中断请求信号的服务如下：

表 24-2. 解除 HALT 模式及解除后的操作

解除源	中断响应状态	解除后的状态	解除后的操作
复位	禁止（DI）	–	正常复位操作
	使能（EI）		
不可屏蔽中断请求信号（不包括多重中断）	禁止（DI）	–	当 HALT 模式解除时，中断请求被响应
	使能（EI）		
可屏蔽中断请求信号	禁止（DI）	–	HALT 模式被解除，但是不响应解除源的中断请求。该中断请求继续保持，继续执行切换到 HALT 模式之前的处理。
	使能（EI）	• 具有较高优先级的中断请求信号被服务，高于解除源的优先级	HALT 模式被解除，但是不响应解除源的中断请求。该中断请求继续保持，继续服务切换到 HALT 模式之前的中断服务处理。
		• 具有较低优先级的中断请求信号被服务，低于解除源的优先级	HALT 模式被解除时，响应中断请求。

表 24-3. HALT 模式下的操作状态

HALT 模式的设置 / 项目		运行状态	
		不使用副时钟时	使用副时钟时
LVI		可操作	
主时钟振荡器		振荡	
副时钟振荡器		–	振荡
内部振荡器		振荡允许	
PLL		可操作	
CPU		停止操作	
DMA		可操作	
中断控制器		可操作	
定时器 P（TMP0 至 TMP5）		可操作	
定时器 Q（TMQ0）		可操作	
定时器 M（TMM0）		当选择 f_{XT} 之外的时钟为计数时钟时，可操作	可操作
钟表定时器/RTC		当 f_X（BRG 分频）选择为计数时钟时，可操作	可操作
看门狗定时器 2		当选择 f_{XT} 之外的时钟为计数时钟时，可操作	可操作
串行接口	CSIB0 至 CSIB4	可操作	
	I^2C00 至 I^2C02	可操作	
	UARTA0 至 UARTA5	可操作	
	UARTC0	可操作	
A/D 转换器		可操作	
D/A 转换器		可操作	
实时输出功能（RTO）		可操作	
按键中断功能（KR）		可操作	
CRC 操作电路		可操作（不把数据输入到 CRCIN 寄存器来停止 CPU 的状态）	
外部总线接口		参见 **2.2 引脚状态。**	
端口功能		保持设置为 HALT 模式前的状态。	
CPU 寄存器组		保持为设置 HALT 模式之前的状态	
内部 RAM			
USB 功能		可操作	

24.4 IDLE1 模式

24.4.1 设置和操作状态

在正常操作模式下，通过将 PSMR.PSM1 和 PSMR.PSM0 位清 00，并且将 PSC.STP 设置为 1，可以设置 IDLE1 模式。

在 IDLE1 模式下，时钟振荡器、PLL、以及 flash 存储器继续操作，但停止 CPU 和其它片上外设功能的时钟供给。

结果，程序停止执行，且内部 RAM 保持与设置为 IDLE1 模式前的内容。CPU 及其它片上外设功能停止工作。但是，可以在副时钟或外部时钟下工作的片上外设功能继续工作。

表 24-5 显示了 IDLE1 模式下的操作状态。

与 HALT 模式相比，IDLE1 模式可以减少更多功耗，因为停止了片上外设功能的运行。主时钟振荡器不停止，所以在解除 IDLE1 模式后，无需等待振荡稳定时间就可以恢复到正常操作模式，HALT 模式解除时也是如此。

注意事项 1，为了设置 IDLE1 模式，需要将数据存储到 PSC 寄存器，在存储指令之后，应该插入五个或更多 NOP 指令。

2. 如果在设置 IDLE1 模式时，有一个未屏蔽的中断请求信号被挂起，则 CPU 不会切换到 IDLE1 模式，而是执行下一条指令。

24.4.2 解除IDLE1 模式

在 IDLE1 模式下，有多种方式可以解除 IDLE1 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽外部中断请求信号（INTP0 至 INTP7 引脚输入）、来自 IDLE1 模式下可操作的外设功能的未屏蔽的内部中断请求信号，或复位信号（由 RESET 引脚输入复位、WDT2RES 信号，低电压检测器（LVI），或时钟监视器（CLM））。

IDLE1 模式释放后，恢复为正常操作模式。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号来解除 IDLE1 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号，可以解除 IDLE1 模式，且与中断请求信号的优先级无关。如果在一个中断服务程序中设置 IDLE1 模式，则之后产生的中断请求信号的服务如下。

表 24-4. 中断请求信号解除 IDLE1 模式及解除后的操作

解除源	中断响应状态	解除后的状态	解除后的操作
复位	禁止（DI）	–	正常复位操作
	使能（EI）		
不可屏蔽中断请求信号（不包括多重中断）	禁止（DI）	–	当 IDLE1 模式解除时，响应中断请求。
	使能（EI）		
可屏蔽中断请求信号	禁止（DI）	–	IDLE1 模式被解除，但是不响应解除源的中断请求。该中断请求继续保持，继续执行切换到 IDLE1 模式之前的处理。
	使能（EI）	• 具有较高优先级的中断请求信号被服务，高于解除源的优先级	IDLE1 模式被解除，但是不响应解除源的中断请求。该中断请求继续保持，继续服务切换到 IDLE1 模式之前的中断服务处理。
		• 具有较低优先级的中断请求信号被服务，低于解除源的优先级	IDLE1 模式被解除时，响应中断请求。

注意事项 通过 PSC.NMI2M，PSC.NMI0M 和 PSC.INTM 位设置为 1（禁止中断）的中断请求信号将被禁止，并且 IDLE1 模式不会被解除。

表 24-5. IDLE1 模式下的操作状态

IDLE1 模式的设置 / 项目		运行状态	
		不使用副时钟时	使用副时钟时
LVI		可操作	
主时钟振荡器		振荡	
副时钟振荡器		–	振荡
内部振荡器		振荡允许	
PLL		可操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是待机模式解除允许）	
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 f_R/8 选择为计数时钟时，可操作	当 f_R/8 或 f_{XT} 选择为计数时钟时，可操作
钟表定时器/RTC		当 f_X（BRG 分频）选择为计数时钟时，可操作	可操作
看门狗定时器 2		当 f_R 选择为计数时钟时，可操作	当 f_R 或 f_{XT} 选择为计数时钟时，可操作
串行接口	CSIB0 至 CSIB4	$\overline{\text{SCKBn}}$ 输入时钟被选为计数时钟时，可操作（n＝0 至 4）	
	I^2C00 至 I^2C02	停止操作	
	UARTA0 至 UARTA5	停止操作（但是当选择 ASCKA0 引脚时钟时，UARTA0 可操作）	
	UARTC0	停止操作	
A/D 转换器		保持操作（转换结果保持）[注]	
D/A 转换器		保持操作（输出保持[注]）	
实时输出功能（RTO）		停止操作（输出保持）	
按键中断功能（KR）		可操作	
CRC 操作电路		停止操作	
外部总线接口		参见 **2.2 引脚状态。**	
端口功能		保持设置为 IDLE1 模式前的状态	
CPU 寄存器组		保持设置 IDLE1 模式之前的状态	
内部 RAM			
USB 功能		停止操作	

注 为了降低功率损耗，在切换到 IDLE1 模式之前，停止 A/D 转换器和 D/A 转换器。

24.5 IDLE2 模式

24.5.1 设置和操作状态

在正常操作模式下，通过将 PSMR.PSM1 和 PSMR.PSM0 位设置为 10，并且将 PSC.STP 设置为 1，可以设置为 IDLE2 模式。

在 IDLE2 模式下，时钟振荡器继续操作，但停止对 CPU、PLL、flash 存储器以及其它片上外设功能的时钟供给。

结果，程序停止执行，且内部 RAM 保持与设置为 IDLE2 模式前的内容。CPU、PLL 以及其它片上外设功能停止工作。但是，可以在副时钟或外部时钟下工作的片上外设功能继续工作。

表 24-7 显示了 IDLE2 模式下的操作状态。

与 IDLE1 模式相比，IDLE2 模式可以减少更多功耗，因为停止了片上外设功能、PLL 和 flash 存储器的运行。但是，因为 PLL 和 flash 存储器停止，当解除 IDLE2 模式时，PLL 和 flash 存储器需要一段建立时间。

注意事项 1. 为了设置 IDLE2 模式，需要将数据存储到 PSC 寄存器，在存储指令之后，应该插入五个或更多 NOP 指令。

2. 如果在设置 IDLE2 模式时，有一个未屏蔽的中断请求信号被挂起，则 CPU 不会切换到 IDLE2 模式，而是执行下一条指令。

24.5.2 解除IDLE2 模式

在 IDLE2 模式下，有多种方式可以解除 IDLE2 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽外部中断请求信号（INTP0 至 INTP7 引脚输入）、来自 IDLE2 模式下可操作的外设功能的未屏蔽的内部中断请求信号，或复位信号（由 RESET 引脚输入复位、WDT2RES 信号，低电压检测器（LVI），或时钟监视器（CLM））。PLL 返回到设置为 IDLE2 模式前的操作状态。

IDLE2 模式释放后，恢复为正常操作模式。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号来解除 IDLE2 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号，可以解除 IDLE2 模式，且与中断请求信号的优先级无关。如果在一个中断服务程序中设置 IDLE2 模式，则之后产生的中断请求信号的服务如下：

表 24-6. 中断请求信号解除 IDLE2 模式及解除后的操作

解除源	中断响应状态	解除后的状态	解除后的操作
复位	禁止（DI）	–	正常复位操作
	使能（EI）		
不可屏蔽中断请求信号（不包括多重中断）	禁止（DI）	–	当 IDLE2 模式解除，并在确保经过指定的建立时间后，响应中断请求。
	使能（EI）		
可屏蔽中断请求信号	禁止（DI）	–	IDLE1 模式被解除，但是不响应解除源的中断请求。该中断请求继续保持，在确保经过指定的建立时间后，继续执行切换到 IDLE1 模式之前的处理。
	使能（EI）	• 具有较高优先级的中断请求信号被服务，高于解除源的优先级	IDLE1 模式被解除，但是不响应解除源的中断请求。该中断请求继续保持，在确保经过指定的建立时间后，继续服务切换到 IDLE1 模式之前的中断服务处理。
		• 具有较低优先级的中断请求信号被服务，低于解除源的优先级	IDLE1 模式被解除时，在确保经过指定的建立时间后，响应中断请求。

注意事项 通过 PSC.NMI2M，PSC.NMI0M 和 PSC.INTM 位设置为 1（禁止中断）的中断请求信号将被禁止，并且 IDLE2 模式不会被解除。

表 24-7. IDLE2 模式下的操作状态

IDLE2 模式的设置 / 项目		运行状态	
		不使用副时钟时	使用副时钟时
LVI		可操作	
主时钟振荡器		振荡	
副时钟振荡器		–	振荡
内部振荡器		振荡允许	
PLL		停止操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是待机模式解除允许）	
定时器 P（TMP0 至 TMP5）		停止操作	
定时器 Q（TMQ0）		停止操作	
定时器 M（TMM0）		当 f_R/8 选择为计数时钟时可操作	当 f_R/8 或 f_{XT} 选择为计数时钟时可操作
钟表定时器/RTC		当 f_X（BRG 分频）选择为计数时钟时可操作	可操作
看门狗定时器 2		当 f_R 选择为计数时钟时可操作	当 f_R 或 f_{XT} 选择为计数时钟时可操作
串行接口	CSIB0 至 CSIB4	$\overline{\text{SCKBn}}$ 输入时钟被选为计数时钟时可操作（n＝0 至 4）	
	I^2C00 至 I^2C02	停止操作	
	UARTA0 至 UARTA5	停止操作（但是当选择 ASCKA0 引脚时钟时，UARTA0 可操作）	
	UARTC0	停止操作	
A/D 转换器		保持操作（转换结果保持）注	
D/A 转换器		保持操作（输出保持注）	
实时输出功能（RTO）		停止操作（输出保持）	
按键中断功能（KR）		可操作	
CRC 操作电路		停止操作	
外部总线接口		参见 **2.2 引脚状态。**	
端口功能		保持设置为 IDLE2 模式前的状态	
CPU 寄存器组		保持设置 IDLE2 模式之前的状态	
内部 RAM			
USB 功能		停止操作	

注 为了降低功率损耗，在切换到 IDLE2 模式之前，停止 A/D 转换器和 D/A 转换器。

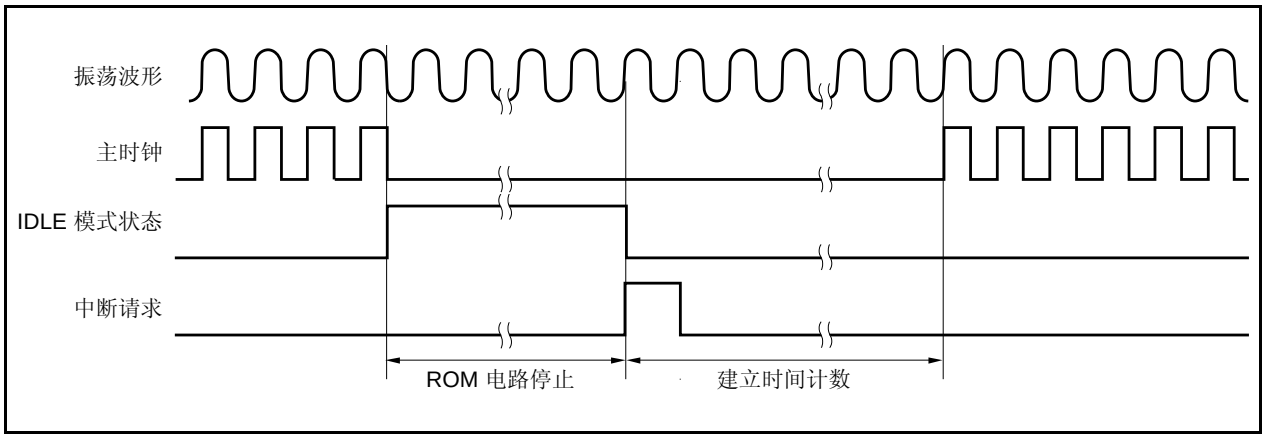
24.5.3 解除IDLE2 模式时确保建立时间

设置 IDLE2 模式，除主时钟振荡器之外的模块都停止操作，所以在解除 IDLE2 模式之后，会为 PLL 或 flash 存储器自动保障建立时间，该时间由 OSTS 寄存器指定。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号来解除 IDLE2 模式

设置 OSTS 寄存器，确保指定的建立时间。

当产生释放源时，依据 OSTS 寄存器的设置，专用内部定时器开始计数。当其溢出时，恢复正常工作模式。



（2）通过复位解除（$\overline{\text{RESET}}$ 引脚输入，WDT2RES 产生）

此操作与正常复位的操作相同。

振荡稳定时间会根据选项字节的设置而有所不同。详情参见 **第三十章 选项字节**。

24.6 STOP模式/低电压STOP模式

24.6.1 设置和操作状态

在正常操作模式下，将 PSMR.PSM1 和 PSMR.PSM0 位设为 01 或 11，并将 PSC.STP 设为 1，可以设置为 STOP 模式。在正常操作模式下，将 REGOVL0 寄存器设置为 01H 后，将 PSMR.PSM1 和 PSMR.PSM0 位设为 01 或 11，并将 PSC.STP 设为 1，可以设置为低电压 STOP 模式。

在 STOP 模式下，副时钟振荡器继续工作，但主时钟振荡器停止。停止对 CPU 和片上外设功能的时钟供给。

结果，程序停止执行，且内部 RAM 保持与设置为 STOP 模式前的内容。停止对 CPU 和片上外设功能的时钟供给，但是副时钟振荡器继续工作。在 STOP 模式下，CSIBn 和 UARTA0 可以使用外部时钟继续工作。但是，在低电压 STOP 模式下，要停止 CSIBn 和 UARTA0（n＝0 至 4）的外部时钟供给，因为这些模块不能继续工作。

表 24-8 显示了在 STOP 模式下的操作状态，而表 24-9 显示了在低电压 STOP 模式下的操作状态。

因为 STOP 模式使主时钟振荡器的运行停止，所以可将功耗降低到低于 IDLE2 模式的水平。若不使用副时钟振荡器、内部振荡器、低电压检测器（LVI）和外部时钟，功耗达到最小化，仅有漏电流通过。

在低电压 STOP 模式下功率消耗进一步减少，因为稳压器的电压更低。

必须按照下列步骤设置低电压 STOP 模式。

（1）从设置“正常模式”→ “低电压 STOP 模式”的过程

在正常操作模式（当主时钟运行时）下按下列步骤设置，而且，还必须设置 OSTS 寄存器。

<1> 停止那些在**表 24-9 低电压 STOP 模式的操作状态**中被指定停止的功能。
尤其要确保停止以下这些功能，因为它们是来自外部源的信号。
- 当选择从 $\overline{\text{SCKBn}}$ 输入时钟至 CSIBn 时，停止 $\overline{\text{SCKBn}}$ 的输入时钟（n＝0 至 4）。
- 当选择从 ASCKA0 输入时钟至 UARTA0 时，停止 ASCKA0 的输入时钟。

<2> 禁止 DMA 操作。

<3>
- 通过 DI 指令禁止可屏蔽中断。
- 禁止 NMI 中断（INTF02＝0，INTR02＝0）。
- 创建一个不会产生 INTWDT2 信号的状态（创建一个状态，该状态下看门狗定时器 2 被清除后不会立即产生 INTWDT2 信号）。

<4> 将 C9H（使能数据）写入 REGPR 寄存器。

<5> 将 01H 写入 REGOVL0 寄存器。
此时，稳压器的输出电压为正常电平。

<6> 将 00H（保护数据）写入 REGPR 寄存器。

<7> 如果必要的话，通过 EI 指令使能可屏蔽中断、NMI 中断或 INTWDT2 中断（恢复上面步骤<2> 和<3>的设置）。

<8> 设置 STOP 模式。
（PSMR.PSM1，PSMR.PSM0 位＝01 或 11
PSC.STP 位＝1）
在 STOP 模式中，稳压器的输出电压下降，将电流消耗减少到非常低的水平。

请确保遵循上面的顺序。
但是，请注意，只要在步骤<6>之后，可在任意时间执行步骤<7>。（步骤<7>中的设置随时可以进行，即使在解除低电压 STOP 模式后。）

注意事项 1. 为了设置 STOP 模式/低电压 STOP 模式，需要将数据存储到 PSC 寄存器，在存储指令之后，应该插入五个或更多 NOP 指令。
2. 如果在设置 STOP 模式/低电压 STOP 模式时，有一个未屏蔽的中断请求信号被挂起，则 CPU 不会切换到 STOP 模式/低电压 STOP 模式，而是执行下一条指令。

表 24-8. STOP 模式的操作状态

项目 \ STOP 模式的设置		运行状态	
		不使用副时钟时	使用副时钟时
LVI		可操作	
主时钟振荡器		停止振荡	
副时钟振荡器		–	振荡
内部振荡器		振荡允许	
PLL		停止操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是待机模式解除允许）	
定时器 P（TMP0 至 TMP5）		停止操作	
定时器 Q（TMQ0）		停止操作	
定时器 M（TMM0）		当 f_R/8 选择为计数时钟时，可操作	当 f_R/8 或 f_{XT} 选择为计数时钟时，可操作
钟表定时器/RTC		停止操作	当 f_{XT} 选择为计数时钟时，可操作
看门狗定时器 2		当 f_R 选择为计数时钟时，可操作	当 f_R 或 f_{XT} 选择为计数时钟时，可操作
串行接口	CSIB0 至 CSIB4	$\overline{\text{SCKBn}}$ 输入时钟被选为计数时钟时可操作（n＝0 至 4）	
	I^2C00 至 I^2C02	停止操作	
	UARTA0 至 UARTA5	停止操作（但是当选择 ASCKA0 输入时钟时，UARTA0 可以操作）	
	UARTC0	停止操作	
A/D 转换器		停止操作（转换结果不确定）注 1，2	
D/A 转换器		停止操作注 3，4（高阻抗输出）	
实时输出功能（RTO）		停止操作（输出保持）	
按键中断功能（KR）		可操作	
CRC 操作电路		停止操作	
外部总线接口		参见 **2.2 引脚状态。**	
端口功能		保持设置为 STOP 模式前的状态。	
CPU 寄存器组		保持设置 STOP 模式之前的状态	
内部 RAM			
USB 功能		停止操作	

注 **1.** 如果在 A/D 转换器操作时设置 STOP 模式，则 A/D 转换器自动停止，并且在 STOP 模式释放后再次开始操作。但是这种情况下，STOP 模式释放后的第一次 A/D 转换结果无效。切换到 STOP 模式之前的所有 A/D 转换结果都无效。

2. 即使在 A/D 转换器操作时设置 STOP 模式，功耗也同样减少，和设置 STOP 模式前停止 A/D 转换器的功耗相等。

3. 如果在 D/A 转换器操作时设置 STOP 模式，那么 D/A 转换器自动停止并且引脚状态变为高阻。STOP 模式解除后，恢复 D/A 转换，设定的时间过后，返回到设置 STOP 模式之前的输出电平状态。

4. 即使在 D/A 转换器操作时设置 STOP 模式，功耗也同样减少，和设置 STOP 模式前停止 D/A 转换器的功耗相等。

表 24-9. 低电压 STOP 模式的操作状态

项目 \ 低电压 STOP 模式的设置		运行状态	
		不使用副时钟时	使用副时钟时
LVI		可操作	
主时钟振荡器		停止振荡	
副时钟振荡器		–	振荡
内部振荡器		振荡允许	
PLL		停止操作	
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是待机模式解除允许）	
定时器 P（TMP0 至 TMP5）		停止操作	
定时器 Q（TMQ0）		停止操作	
定时器 M（TMM0）		当 f_R/8 选择为计数时钟时，可操作	当 f_R/8 或 f_{XT} 选择为计数时钟时，可操作
钟表定时器/RTC		停止操作	当 f_{XT} 选择为计数时钟时，可操作
看门狗定时器 2		当 f_R/8 选择为计数时钟时，可操作	当 f_R/8 或 f_{XT} 选择为计数时钟时，可操作
串行接口	CSIB0 至 CSIB4	停止操作 （当 $\overline{SCKBn}$ 输入时钟被选为计数时钟时，务必停止 $\overline{SCKBn}$ 输入时钟（n ＝ 0 至 4））	
	I^2C00 至 I^2C02	停止操作	
	UARTA0 至 UARTA5	停止操作 （当选择 ASCKA0 输入时钟至 UARTA0 时，务必停止 ASCKA0 输入时钟。）	
	UARTC0	停止操作	
A/D 转换器		停止操作（转换结果不确定）注 1, 2	
D/A 转换器		停止操作注 3, 4（高阻抗输出）	
实时输出功能（RTO）		停止操作（输出保持）	
按键中断功能（KR）		可操作	
CRC 操作电路		停止操作	
外部总线接口		参见 **2.2 引脚状态。**	
端口功能		保持设置为低电压 STOP 模式前的状态。	
CPU 寄存器组		保持设置低电压 STOP 模式之前的状态	
内部 RAM			
USB 功能		停止操作	

注

1. 如果在 A/D 转换器操作时设置低电压 STOP 模式，则 A/D 转换器自动停止，并且在低电压 STOP 模式释放后再次开始操作。但是这种情况下，低电压 STOP 模式释放后的 A/D 转换结果无效。低电压 STOP 模式设置前的所有 A/D 转换值都无效。
2. 即使在 A/D 转换器操作时设置低电压 STOP 模式，功耗也同样减少，和设置低电压 STOP 模式前停止 A/D 转换器的功耗相等。
3. 如果在 D/A 转换器操作时设置低电压 STOP 模式，那么 D/A 转换器自动停止并且引脚状态变为高阻。低电压 STOP 模式解除后，恢复 D/A 转换，设定的时间过后，返回到设置低电压 STOP 模式之前的输出电平状态。
4. 即使在 D/A 转换器操作时设置低电压 STOP 模式，功耗也同样减少，和设置低电压 STOP 模式前停止 D/A 转换器的功耗相等。

24.6.2 解除STOP模式/低电压STOP模式

有多种方式可以解除 STOP 模式/低电压 STOP 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽外部中断请求信号（INTP0 至 INTP7 引脚输入）、来自 STOP 模式/低电压 STOP 模式下可操作的外设功能的未屏蔽的内部中断请求信号，或复位信号（由 RESET 引脚输入复位、WDT2RES 信号，低电压检测器（LVI））。

在 STOP 模式/低电压 STOP 模式解除后，在保证的振荡稳定时间后恢复为正常操作模式。

关于在解除低电压 STOP 模式后的重设，参见 **24.6.3 低电压 STOP 模式解除后重新设置**。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号来解除 STOP 模式/低电压 STOP 模式

通过不可屏蔽中断请求信号或未屏蔽中断请求信号，可解除 STOP 模式/低电压 STOP 模式，且与中断请求信号的优先级无关。但是，如果在一个中断服务程序中设置 STOP 模式/低电压 STOP 模式，则之后发出的中断请求信号的服务如下。

表 24-10. 中断请求信号解除 STOP 模式/低电压 STOP 模式及解除后的操作

解除源	中断响应状态	解除后的状态	解除后的操作
复位	禁止（DI）	–	正常复位操作
	使能（EI）		
不可屏蔽中断请求信号（不包括多重中断）	禁止（DI）	–	当 STOP 模式/低电压 STOP 模式解除，并在确保经过指定的建立时间后，响应中断请求。
	使能（EI）		
可屏蔽中断请求信号	禁止（DI）	–	STOP 模式/低电压 STOP 模式被解除，但是不响应解除源的中断请求。该中断请求继续保持，在确保经过指定的建立时间后，继续执行切换到 STOP 模式/低电压 STOP 模式之前的处理。
	使能（EI）	• 具有较高优先级的中断请求信号被服务，高于解除源的优先级	STOP 模式/低电压 STOP 模式被解除，但是不响应解除源的中断请求。该中断请求继续保持，在确保经过指定的建立时间后，继续服务切换到 STOP 模式/低电压 STOP 模式之前的中断服务处理。
		• 具有较低优先级的中断请求信号被服务，低于解除源的优先级	STOP 模式/低电压 STOP 模式被解除时，在确保经过指定的建立时间后，响应中断请求。

注意事项 通过 PSC.NMI2M，PSC.NMI0M 和 PSC.INTM 位设置为 1（禁止中断）的中断请求信号将被禁止，并且 STOP 模式/低电压 STOP 模式不会被解除。

24.6.3 低电压STOP模式解除后重新设置

（1）如果通过中断解除低电压 STOP 模式

在低电压 STOP 模式被解除后的状态如下：

- 稳压器：自动恢复到正常电平。
  必须保障 OSTS 寄存器指定的振荡稳定时间。
- REGOVL0 寄存器＝01H（低电压 STOP 模式）：在 **24.6.1（1）<5>**中的描述的值被保留。
- REGPR 寄存器＝00H（保护数据）：在 **24.6.1（1）<6>**中的描述的值被保留。

（a） 若连续使用 REGOVL0 寄存器＝01H（低压 STOP 模式），则其它寄存器不必重新设置。

（b） 返回 REGOVL0 寄存器＝00H 时需按照以下步骤：

<1> 禁止 DMA 操作。

<2>
- 通过 DI 指令禁止可屏蔽中断。
- 禁止 NMI 中断（INTF02＝0，INTR02＝0）。
- 创建一个不会产生 INTWDT2 信号的状态（停止看门狗定时器 2 或设置一个 INTWDT2 之外的模式。创建一个状态，该状态下看门狗定时器 2 被清除后不会立即产生 INTWDT2 信号）。

<3> 将 C9H（使能数据）写入 REGPR 寄存器。

<4> 将 00H 写入 REGOVL0 寄存器。

<5> 将 00H（保护数据）写入 REGPR 寄存器。

<6> 如果必要的话，通过使能 DMA 或用 EI 指令使能可屏蔽中断、NMI 中断或 INTWDT2 中断（恢复上面步骤<1>和<2>的设置）。

请确保遵循上面的顺序。

（2）如果通过复位解除低电压 STOP 模式

CPU 在从复位状态解除后立即转变为正常操作模式，REGOVL0 寄存器初始化为 00H 且 REGPR 寄存器初始化为 00H（保护数据）。请务必通过设置选项字节来确保复位结束之后的振荡稳定时间。详情**参见第三十章 选项字节**。

注意事项 通过 PSC.NMI1M，PSC.NMI0M 和 PSC.INTM 位设置为 1（禁止）的中断请求无效，且低电压 STOP 模式不会被解除。

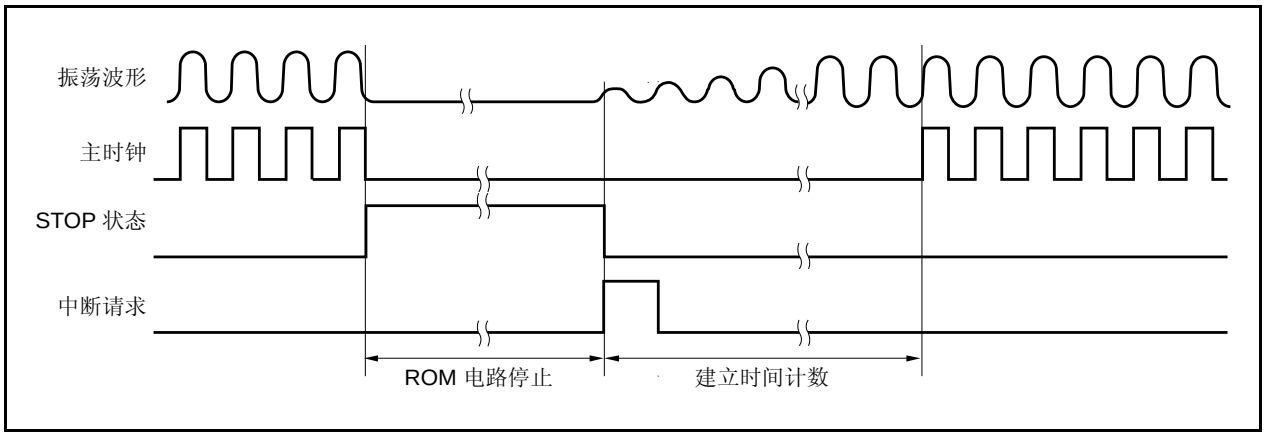
24.6.4 在解除STOP模式时确保振荡稳定时间

为了主时钟振荡器，在解除 STOP 模式后需要一段时间来确保主时钟振荡器的振荡稳定，因为在设置 STOP 模式之后，主时钟振荡器的操作停止。

（1） 通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号来解除 STOP 模式

设置 OSTS 寄存器，确保振荡稳定时间。

当产生释放源时，依据 OSTS 寄存器的设置，专用内部定时器开始计数。当其溢出时，恢复正常工作模式。



（2） 通过复位解除

执行的操作与正常的复位操作相同。

振荡稳定时间会根据选项字节的设置而有所不同。详情参见 **第三十章 选项字节**。

24.7 副时钟操作模式/低电压副时钟操作模式

24.7.1 设置和操作状态

在正常操作模式下，把 PCC.CK3 位设置为 1，可以设置为副时钟工作模式。在副时钟操作模式下，把 REGOVL0 寄存器设置为 02H，就可以设置为低电压操作模式。

当设置为副时钟工作模式时，内部系统时钟从主时钟变为副时钟。使用 PCC.CLS 位来检查时钟是否已经转换。

当 PCC.MCK 位设置为 1 时，主时钟振荡器停止运行。结果，系统工作于副时钟。

在副时钟工作模式下，功耗可以降至低于正常操作模式的级别，因为副时钟用作内部系统时钟。此外，停止主时钟振荡器的运行，功耗可进一步降低至 STOP 模式的级别。在低电压副时钟操作模式下，功率消耗减少更多，这是因为稳压器的电压更低。

在副时钟操作模式下，当主时钟振荡器停止时，使用外部时钟操作的 CSIBn 和 UARTA0 可以继续操作，但是在低电压副时钟操作模式时，停止输入到 CSIBn 和 UARTA0（n＝0 至 4）的外部时钟供给，因为这些模式不能继续工作。

注意事项 1. 在对 CK3 位进行操作时，不要改变 PCC.CK2 至 PCC.CK0 位的设定值（建议用位操作指令对此位进行操作）。有关 PCC 寄存器的详情，参见“6.3（1）处理器时钟控制寄存器（PCC）”。

2. 若下列条件不满足，则改变 CK2 至 CK0 位使条件能够满足，并设置为副时钟工作模式。

内部系统时钟（f_{CLK}）> 副时钟（f_{XT} = 32.768kHz）× 4

备注 内部系统时钟（f_{CLK}）： 根据 CK2 至 CK0 位的设置，由主时钟（f_{XX}）产生的时钟。

必须按照下列步骤设置低电压副时钟操作模式。

（1）从副时钟操作模式切换到低电压副时钟操作模式的过程

在副时钟操作模式下，按照下列步骤进行设置。

<1> 停止主时钟和锁相环 PLL。

<2> 停止那些在**表 24-14 低电压 STOP 模式的操作状态**中被指定停止的功能。
尤其要确保停止以下这些功能，因为它们是来自外部源的信号。
- 当选择从 $\overline{\text{SCKBn}}$ 输入时钟至 CSIBn 时，停止 $\overline{\text{SCKBn}}$ 的输入时钟（n＝0 至 4）。
- 当选择从 ASCKA0 输入时钟至 UARTA0 时，停止 ASCKA0 的输入时钟。

<3> 禁止 DMA 操作（如果 DMA 操作使能）。

<4>
- 通过 DI 指令禁止可屏蔽中断。
- 禁止 NMI 中断（INTF02＝0，INTR02＝0）。
- 创建一个不会产生 INTWDT2 信号的状态（停止看门狗定时器 2 或设置一个 INTWDT2 之外的模式。创建一个状态，该状态下看门狗定时器 2 被清除后不会立即产生 INTWDT2 信号）。

<5> 将 C9H（使能数据）写入 REGPR 寄存器。

<6> 将 02H 写入 REGOVL0 寄存器。
此时，稳压器的输出电压为低电压，将功率消耗减少到非常低的水平。

<7> 将 00H（保护数据）写入 REGPR 寄存器。

<8> 如果必要的话，通过 EI 指令使能可屏蔽中断、NMI 中断或 INTWDT2 中断（恢复上面步骤<4>的设置）。

请确保遵循上面的顺序。

关于副时钟操作模式的设置，请参见 **24.7.1 设置和操作状态。**

表 24-11 显示了在副时钟操作模式下的操作状态，而表 24-12 显示了在低电压副时钟操作模式下的操作状态。

表 24-11. 副时钟工作模式的操作状态

副时钟工作模式的设置 / 项目		运行状态	
		主时钟振荡时	主时钟停止时
LVI		可操作	
副时钟振荡器		振荡	
内部振荡器		振荡允许	
PLL		可操作	停止操作[注]
CPU		可操作	
DMA		可操作	
中断控制器		可操作	
定时器 P（TMP0 至 TMP5）		可操作	停止操作
定时器 Q（TMQ0）		可操作	停止操作
定时器 M（TMM0）		可操作	当 $f_R/8$ 或 f_{XT} 选择为计数时钟时，可操作
钟表定时器/RTC		可操作	当 f_{XT} 选择为计数时钟时，可操作
看门狗定时器 2		可操作	当 f_R 或 f_{XT} 选择为计数时钟时，可操作
串行接口	CSIB0 至 CSIB4	可操作	$\overline{SCKBn}$ 输入时钟被选为计数时钟时，可操作（n＝0 至 4）
	I^2C00 至 I^2C02	可操作	停止操作
	UARTA0 至 UARTA5	可操作	停止操作（但是当选择 ASCKA0 引脚时钟时，UARTA0，可操作）
	UARTC0	可操作	停止操作
A/D 转换器		可操作	停止操作
D/A 转换器		可操作	
实时输出功能（RTO）		可操作	停止操作（输出保持）
按键中断功能（KR）		可操作	
CRC 操作电路		可操作	
外部总线接口		参见 **2.2 引脚状态。**	
端口功能		可设置	
CPU 寄存器组		可设置	
内部 RAM			
USB 功能		停止操作	

注 在停止主时钟前，请确保停止锁相环（PLLCTL.PLLON 位＝0）。

注意事项 当 CPU 运行于副时钟而且主时钟振荡停止时，禁止访问那些已经指定等待的寄存器。如果已经产生等待，只能通过复位来解除（参见 3.4.8（2））。

表 24-12. 低电压副时钟工作模式的操作状态

项目 \ 低电压副时钟工作模式的设置		运行状态
		主时钟停止（必须停止）
LVI		可操作
副时钟振荡器		振荡
内部振荡器		振荡允许
PLL		停止操作[注]
CPU		可操作
DMA		停止操作（必须停止）
中断控制器		可操作
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 $f_R/8$ 或 f_{XT} 选择为计数时钟时，可操作
钟表定时器/RTC		当 f_{XT} 选择为计数时钟时，可操作
看门狗定时器 2		当 $f_R/8$ 或 f_{XT} 选择为计数时钟时，可操作
串行接口	CSIB0 至 CSIB4	停止操作 （当 $\overline{\text{SCKBn}}$ 输入时钟被选为计数时钟时，确保停止 $\overline{\text{SCKBn}}$ 的输入时钟（n＝0 至 4）。）
	I^2C00 至 I^2C02	停止操作
	UARTA0 至 UARTA5	停止操作 （当选择 ASCKA0 输入时钟至 UARTA0 时，务必停止 ASCKA0 输入时钟。）
	UARTC0	停止操作
A/D 转换器		停止操作
D/A 转换器		停止操作（必须停止）
实时输出功能（RTO）		停止操作（输出保持）
按键中断功能（KR）		可操作
CRC 操作电路		停止操作（必须停止）
外部总线接口		参见 **2.2 引脚状态。**
端口功能		可设置
内部数据		可设置
CPU 寄存器组		可设置
内部 RAM		
USB 功能		停止操作

注 请确保停止锁相环（PLLCTL.PLLON 位＝0）。

注意事项 当 CPU 运行于副时钟而且主时钟振荡停止时，禁止访问那些已经指定等待的寄存器。如果已经产生等待，只能通过复位来解除（参见 3.4.8（2））。

24.7.2 解除副时钟操作模式

当 CK3 位被设置为 0 时，通过复位信号（$\overline{\text{RESET}}$ 引脚的输入、WDT2RES 信号、低电压检测器（LVI）或时钟监视器（CLM）复位）来解除副时钟操作模式。

如果主时钟停止（MCK 位＝ 1），将 MCK 位置为 1，用软件保证主时钟的振荡稳定时间，并将 CK3 位清 0。

当副时钟工作模式解除时，恢复为正常操作模式。

注意事项 在操作 CK3 位时，不要改变 CK2 至 CK0 位的设置值（建议用位操作指令对此位进行操作）。
有关 PCC 寄存器的详情，参见“6.3（1）处理器时钟控制寄存器（PCC）”。

24.7.3 解除低电压副时钟操作模式

在低电压副时钟模式下，把 REGOVL0 寄存器设置为 00H，可以进入副时钟操作模式。在此之后，根据 **24.7.2 解除副时钟操作模式**的描述切换到正常模式。务必按照以下过程从低电压副时钟操作模式切换到副时钟操作模式。

（1）从设置“低电压副时钟操作模式”→“副时钟操作模式”的过程

在低电压副时钟操作模式下，按照以下步骤进行设置。

<1> • 通过 DI 指令禁止可屏蔽中断。
• 禁止 NMI 中断（INTF02＝0，INTR02＝0）。
• 创建一个不会产生 INTWDT2 信号的状态（创建一个状态，该状态下看门狗定时器 2 被清除后不会立即产生 INTWDT2 信号）。

<2> 将 C9H（使能数据）写入 REGPR 寄存器。

<3> 将 00H 写入 REGOVL0 寄存器（转换到副时钟操作模式）。

<4> 将 00H（保护数据）写入 REGPR 寄存器。

<5> 通过软件等待至少 800 μs。

<6> 如果必要的话，通过 EI 指令使能可屏蔽中断、NMI 中断或 INTWDT2 中断（恢复上面步骤<1>的设置）。

<7> 如果有必要的话，使能 DMA。

<8> 启动需要使用的功能，这些功能在 **24.7.1（1）“副时钟操作模式”→ “低电压副时钟操作模式”的设置过程**的步骤<1> 和<2>中已经被停止。

请确保遵循上面的顺序。

但是，请注意，只要在步骤<5>执行完成之后，步骤<6>， <7>和 <8>可在任意时间执行。

（2）如果通过复位解除低电压副时钟操作模式

通过复位信号（通过 $\overline{\text{RESET}}$ 引脚输入、WDT2RES 信号、低电压检测器（LVI）或时钟监视器（CLM）复位）来解除低电压副时钟操作模式。CPU 在复位状态解除后立即切换为正常操作模式，REGOVL0 寄存器初始化为 00H 且 REGPR 寄存器初始化为 00H（保护数据）。请务必通过设置选项字节来确保复位结束之后的振荡稳定时间。详情参见**第三十章 选项字节**。

24.8 副IDLE模式/低电压副IDLE模式

24.8.1 设置和操作状态

在副时钟工作模式下，将 PSMR.PSM1 和 PSMR.PSM0 位设置为 00 或 10 并将 PSC.STP 位设置为 1，可以设置为副 IDLE 模式。在副时钟操作模式下，在 REGOVL0 寄存器被设置为 02H 后，将 PSMR.PSM1 和 PSMR.PSM0 位设置为 00 或 10，并将 PSC.STP 位设置为 1，可以设置为低电压副 IDLE 模式。

该模式下，时钟振荡器继续工作，但停止对 CPU、Flash 存储器及其它片上外设功能的时钟供给。

结果，程序停止执行，且内部 RAM 保持与设置为副 IDLE 模式前的内容。CPU 及其它片上外设功能停止。然而，可以在副时钟或外部时钟下工作的片上外设功能继续工作。在副时钟操作模式下，CSIBn 和 UARTA0 可以使用外部时钟操作，继续工作。在低电压副时钟操作模式下，停止 CSIBn 和 UARTA0 外部时钟输入的供给（n＝0 至 4），因为这些模块不能继续工作。

因为副 IDLE 模式停止了 CPU、flash 存储器及其它片上外设功能的操作，所以可以比副时钟工作模式降低更多功耗。

如果在主时钟停止后设置为副 IDLE 模式，则该模式下的电流消耗可减少至与 STOP 模式同样低的水平。在低电压副 IDLE 模式下，功率消耗减少更多，这是因为稳压器的电压被进一步降低。

表 24-13 显示了在副 IDLE 模式下的操作状态，而表 24-14 显示了在低电压副 IDLE 模式下的操作状态。

必须以下列步骤设置低电压副 IDLE 模式。

（1）从设置“副时钟操作模式”→ “低电压副时钟操作模式”→“低电压副 IDLE 模式”的过程

在副时钟操作模式下按以下步骤进行设置。

<1> 停止主时钟和锁相环 PLL。

<2> 停止那些在表 **24-14 低电压 STOP 模式的操作状态**中被指定停止的功能。
尤其是确保停止以下这些功能，因为它们是来自外部源的信号。
- 当选择从 $\overline{\text{SCKBn}}$ 输入时钟至 CSIBn 时，停止 $\overline{\text{SCKBn}}$ 的输入时钟（n ＝ 0 至 4）。
- 当选择从 ASCKA0 输入时钟至 UARTA0 时，停止 ASCKA0 的输入时钟。

<3> 禁止 DMA 操作（如果 DMA 操作使能）。

<4>
- 通过 DI 指令禁止可屏蔽中断。
- 禁止 NMI 中断（INTF02＝0，INTR02＝0）。
- 创建一个不会产生 INTWDT2 信号的状态（停止看门狗定时器 2 或设置一个 INTWDT2 之外的模式。创建一个状态，该状态下看门狗定时器 2 被清除后不会立即产生 INTWDT2 信号）。

<5> 将 C9H（使能数据）写入 REGPR 寄存器。

<6> 将 02H 写入 REGOVL0 寄存器。
此时，稳压器的输出电压为低电平，将功率消耗减少到非常低的水平。

<7> 将 00H（保护数据）写入 REGPR 寄存器。

<8> 如果必要的话，通过 EI 指令使能可屏蔽中断、NMI 中断或 INTWDT2 中断（恢复上面步骤<4>的设置）。

<9> 设置副 IDLE 模式。

（PSMR.PSM1，PSMR.PSM0 位＝00 或 10
PSC.STP 位＝1）

请确保遵循上面的顺序。

关于副时钟操作模式的设置，参见 **24.7.1 设置和操作状态。**

注意事项 1. 为了设置副 IDLE 模式/低电压副 IDLE 模式，需要将数据存储到 PSC 寄存器，在存储指令之后，应该插入五个或更多 NOP 指令。

2. 如果在设置副 IDLE 模式/低电压副 IDLE 模式时，有一个未屏蔽的中断请求信号被挂起，则 CPU 不会切换到副 IDLE 模式/低电压副 IDLE 模式，而是执行下一条指令。

表 24-13. 副 IDLE 模式的操作状态

副 IDLE 模式的设置 / 项目		运行状态	
		主时钟振荡时	主时钟停止时
LVI		可操作	
副时钟振荡器		振荡	
内部振荡器		振荡允许	
PLL		可操作	停止操作[注1]
CPU		停止操作	
DMA		停止操作	
中断控制器		停止操作（但是待机模式解除允许）	
定时器 P（TMP0 至 TMP5）		停止操作	
定时器 Q（TMQ0）		停止操作	
定时器 M（TMM0）		当 $f_R/8$ 或 f_{XT} 选择为计数时钟时，可操作	
钟表定时器/RTC		可操作	当 f_{XT} 选择为计数时钟时，可操作
看门狗定时器 2		当 f_R 或 f_{XT} 选择为计数时钟时，可操作	
串行接口	CSIB0 至 CSIB4	$\overline{\text{SCKBn}}$ 输入时钟被选为计数时钟时，可操作（n＝0 至 4）	
	I^2C00 至 I^2C02	停止操作	
	UARTA0 至 UARTA5	停止操作（但是当选择 ASCKA0 引脚时钟时，UARTA0 可操作）	
	UARTC0	停止操作	
A/D 转换器		保持操作（转换结果保持）[注2]	
D/A 转换器		保持操作（输出保持）[注2]	
实时输出功能（RTO）		停止操作（输出保持）	
按键中断功能（KR）		可操作	
CRC 操作电路		停止操作	
外部总线接口		参见 **2.2 引脚状态**（与 IDLE1 和 IDLE2 模式相同的操作）。	
端口功能		保持设置为副 IDLE 模式前的状态。	
CPU 寄存器组		保持设置副 IDLE 模式之前的状态	
内部 RAM			
USB 功能		停止操作	

注 **1.** 在停止主时钟前，请确保停止锁相环（PLLCTL.PLLON 位＝0）。

2. 为了降低功率损耗，在切换到副 IDLE 模式前，停止 A/D 转换器和 D/A 转换器。

表 24-14. 低电压副 IDLE 模式的操作状态

低电压副 IDLE 模式的设置 / 项目		运行状态
		主时钟停止（必须停止）
LVI		可操作
副时钟振荡器		振荡
内部振荡器		振荡允许
PLL		停止操作[注]
CPU		停止操作
DMA		停止操作
中断控制器		停止操作（但是待机模式解除允许）
定时器 P（TMP0 至 TMP5）		停止操作
定时器 Q（TMQ0）		停止操作
定时器 M（TMM0）		当 $f_R/8$ 或 f_{XT} 选择为计数时钟时，可操作
钟表定时器/RTC		当 f_{XT} 选择为计数时钟时，可操作
看门狗定时器 2		当 $f_R/8$ 或 f_{XT} 选择为计数时钟时，可操作
串行接口	CSIB0 至 CSIB4	停止操作 （当 $\overline{\text{SCKBn}}$ 输入时钟被选为计数时钟时，务必停止 $\overline{\text{SCKBn}}$ 输入时钟（n ＝ 0 至 4）。）
	I^2C00 至 I^2C02	停止操作
	UARTA0 至 UARTA2	停止操作 （当选择 ASCKA0 输入时钟至 UARTA0 时，务必停止 ASCKA0 输入时钟。）
	UARTC0	停止操作
A/D 转换器		停止操作
D/A 转换器		停止操作（必须停止）
实时输出功能（RTO）		停止操作（输出保持）
按键中断功能（KR）		可操作
CRC 操作电路		停止操作
外部总线接口		参见 **2.2 引脚状态**（与 IDLE1 和 IDLE2 模式相同的操作）。
端口功能		保持设置为低电压副 IDLE 模式前的状态。
CPU 寄存器组		保持设置低电压副 IDLE 模式之前的状态
内部 RAM		
USB 功能		停止操作

注 请确保停止锁相环（PLLCTL.PLLON 位＝0）。

24.8.2 解除副IDLE模式/低电压副IDLE模式

有多种方式可以解除副 IDLE 模式/低电压副 IDLE 模式。包括不可屏蔽中断请求信号（NMI 引脚输入，INTWDT2 信号）、未屏蔽外部中断请求信号（INTP0 至 INTP7 引脚输入）、在副 IDLE 模式/低电压副 IDLE 模式下可操作的外设功能的未屏蔽的内部中断请求信号，或复位信号（由 RESET 引脚输入复位、WDT2RES 信号，低电压检测器（LVI），或时钟监视器（CLM））。PLL 返回到设置副 IDLE 模式之前的操作状态。它在低电压副 IDLE 模式时返回停止状态。

通过中断请求信号解除副 IDLE 模式时，设置为副时钟工作模式。

通过中断请求信号解除低电压副 IDLE 模式时，设置为低电压副时钟工作模式。

关于解除低电压副时钟操作模式，请参见 **24.7.3 解除低电压副时钟操作模式**。

（1）通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号解除副 IDLE 模式/低电压副 IDLE 模式

通过不可屏蔽中断请求信号或未屏蔽的可屏蔽中断请求信号，可解除副 IDLE 模式/低电压副 IDLE 模式，且与中断请求信号的优先级无关。但是，如果在中断服务程序中设置副 IDLE 模式/低电压副 IDLE 模式，则之后发出的中断请求信号的服务如下.

表 24-15. 中断请求信号解除副 IDLE 模式/低电压副 IDLE 模式及解除后的操作

解除源	中断响应状态	解除后的状态	解除后的操作
复位	禁止（DI）	–	正常复位操作
	使能（EI）		
不可屏蔽中断请求信号（不包括多重中断）	禁止（DI）	–	当副 IDLE 模式/低电压副 IDLE 模式解除时，中断请求被响应
	使能（EI）		
可屏蔽中断请求信号	禁止（DI）	–	副 IDLE 模式/低电压副 IDLE 模式被解除，但是不响应解除源的中断请求。该中断请求继续保持，继续执行切换到副 IDLE 模式/低电压副 IDLE 模式之前的处理。
	使能（EI）	• 具有较高优先级的中断请求信号被服务，高于解除源的优先级	副 IDLE 模式/低电压副 IDLE 模式被解除，但是不响应解除源的中断请求。该中断请求继续保持，继续服务切换到副 IDLE 模式/低电压副 IDLE 模式之前的中断服务处理。
		• 具有较低优先级的中断请求信号被服务，低于解除源的优先级	副 IDLE 模式/低电压副 IDLE 模式被解除时，响应中断请求。

注意事项 **通过 PSC.NMI2M，PSC.NMI0M，和 PSC.INTM 位设置为 1（禁止）的中断请求信号将被禁止，并且副 IDLE 模式/低电压副 IDLE 模式不会被解除。**

24.9 RTC备份模式

通过停止 RTC 备份电源（RV_{DD}）之外的其它电源供给，可以将 V850ES/JG3-L 切换到 RTC 备份模式。

在 RTC 备份模式下，RTC 计数且副时钟振荡器使用专供 RTC 备份区域的稳压器工作，其使用 RV_{DD} 作为电源供给。详情参见**第十一章 实时计数器**。

24.9.1 寄存器

控制 RTC 备份模式的寄存器如下。

- RTC 备份控制寄存器 0（RTCBUMCTL0）
- 副时钟低电压操作控制寄存器（SOSCAMCTL）

（1）RTC 备份控制寄存器 0（RTCBUMCTL0）

RTCBUMCTL0 寄存器用于控制 RTC 备份模式。该寄存器是个专用寄存器，只能按照特定序列写入（参见 **3.4.7 专用寄存器**）。

该寄存器可以按字节或位进行读写。

对 RV_{DD} 执行上电复位时，该寄存器被清除为 00H。

复位后：00H R/W 地址： FFFFFB00H

	<7>	6	5	4	3	2	1	<0>
RTCBUMCTL0	RBMEN	0	0	0	0	0	0	RBMSET

RBMEN	RTC 备份模式控制
0	禁止使用RTC备份模式
1	允许使用RTC备份模式

RBMSET	RTC备份模式设置
0	退出RTC备份模式
1	设置RTC备份模式 当RBMSET位被置位(1)时，RTC状态切换如下。 • 选择副时钟 (f_{XT})的分频时钟作为RTC输入时钟 • RTC引脚输出功能停止 • RTC时间误差修正功能停止

注意事项 1. 不要同时将 RBMEN 和 RBMSET 位置位（1）。如果它们被同时置位（1），RTC 备份模式可能无法正确操作。首先将 RBMEN 位置位（1），然后将 RBMSET 位置位（1）。

2. 当 RBMEN 位为 0 时，不要将 RBMSET 位置位（1）。如果此时将 RBMSET 位置位（1），则该位被置位（1）但是无法指定 RTC 备份模式。

（2）副时钟低电压操作控制寄存器（SOSCAMCTL）

SOSCAMCTL 寄存器用于选择副时钟（f_{XT}）的低电压控制方式，以执行 RTC 备份模式下的低电压操作。该寄存器是个专用寄存器，只能按照特定序列写入（参见 **3.4.7 专用寄存器**）。

该寄存器可以按字节或位进行读写。

对 RV_{DD} 执行上电复位时，该寄存器被清除为 00H。

复位后：00H R/W 地址： FFFFFB03H

	7	6	5	4	3	2	1	<0>
SOSCAMCTL	0	0	0	0	0	0	0	AMPHS

AMPHS	副时钟 (f_{XT}) 振荡模式选择
0	正常振荡
1	超低功耗振荡

注意事项 请确保将第 7 位至第 1 位清为“0”。

备注 当副时钟（f_{XT}）在超低功耗模式下振荡时，噪声的影响会更容易导致振荡周期计数的不正确数量。在决定使用该模式之前，请全面评估噪声的影响。

24.9.2 RTC备份模式设置条件

(1) RTC 备份模式的设置条件

如果满足下列条件，设置 RTC 备份模式。

<1> RTCBUMCTL0.RBMEN = 1 （使能 RTC 备份模式的使用）

<2> SOSCAMCTL.AMPHS = 1 （在超低功耗模式下，必须为副时钟（f_{XT}）振荡执行该设置，但是对 RTC 备份模式来说不是必需设置）

<3> RTCBUMCTL0.RBMSET = 1 （RTC 备份模式设置）

(2) RTC 备份模式的退出条件

如果满足下列条件，解除 RTC 备份模式。

<1> SOSCAMCTL.AMPHS = 0 （副时钟的正常振荡）

<2> RTCBUMCTL0.RBMSET = 0 （RTC 备份模式退出）

24.9.3 RTC备份模式设置过程

RTC 备份模式设置/退出过程如下。

（1）设置 RTC 备份模式

注意事项 在副时钟操作时，禁止 RTC 备份模式。
请确保在主时钟操作时设置 RTC 备份模式。

（a） 初始化设置

进入 RTC 备份模式之后，执行下列的设置模式。

- 将 RTCBUMCTL0 设置为 1（特定序列），并设置 RTC 备份模式使能状态。
- 指定 LVI 用作中断。
  （使用 LVIS 寄存器来设置低电压检测级别为 2.8 V（典型值）或 2.3 V（典型值））
- 使用 RC1CC0.RC1CKS 位来指定副时钟（f_{XT}）作为工作时钟，并启动 RTC 操作。详情参见**第十一章 实时计数器**。

(b) INTLVI 例程

V_{DD} 降低且发生 INTLVI 中断之后，使用 INTLVI 中断服务程序来执行下列处理。

<1> 读取低电压检测标志（LVIM.LVIF）并确认 LVIM.LVIF = 1。如果 LVIM.LVIF = 1，则执行步骤<2> 及后续步骤。如果 LVIM.LVIF = 0，V_{DD} 电压没有降低到 LVI 检测电平，所以无需指定 RTC 备份模式（步骤 <2>及后续步骤），并执行**（3） 退出 RTC 备份模式（未发生外部复位时）**中描述的退出处理。
<2> 根据 V_{DD} 电压设置工作时钟（2.2 V@5 MHz, 2.0 V@2.5 MHz, 等）
<3> 禁止 DMA [注1]
<4> 禁止 NMI（将 INTF02 和 INTR02 清除为 0）并设置不会立即产生 INTWD2（通过清除 WDT2）的状态，或停止 WDT2 或 WDT2 的时钟源。[注2, 3, 4]
<5> 使用中断屏蔽寄存器 0 至 3 禁止 INTLVI 之外的可屏蔽中断。
<6> 将 SOSCAMCTL.AMPHS 设置为 1（特定序列），并设置副时钟（f_{XT}）为超低功耗模式。
<7> 将 RTCBUMCTL0.RBMSET 设置为 1（特定序列），并设置 RTC 备份模式。
<8> 设置正常停止模式。
<9> RETI

注
1. 在 INTLVI 中断服务程序中，如果在停止 DMA 操作之前发生 DMA 操作，且在 DMA 传输完成之前 V_{DD} 电压达到最低保证电压，则不可能指定 RTC 备份模式。
2. 在 INTLVI 中断服务程序中，如果正在指定步骤<4>的设置时发生 NMI 或 INTWDT2 中断，中断服务开始。如果服务需要一段时间，在服务期间 V_{DD} 电压达到最低保证电压，则不可能指定 RTC 备份模式。
3. 如果使用选项功能将 WDTMD1 设置为 1 并将 WDT2 固定为复位模式，WDT2 不能停止。如果在已经指定这些设置时产生复位，根据复位初始化流程中指定的退出设置来退出 RTC 备份模式。
4. 在上述的步骤<4>中，如果在 RTC 备份模式下通过清除 WDT2 然后产生 INTWDT2 却未产生外部复位的方式来处理 INTWDT2，则退出 RTC 备份模式。

（2）退出 RTC 备份模式（发生外部复位时）

进入 RTC 备份模式之后，如果产生外部复位信号（$\overline{\text{RESET}}$），执行下面的处理例程。如果 V_{DD} 电压超出了可保证操作范围，则有必要产生一个外部复位信号（$\overline{\text{RESET}}$）。恢复 V_{DD} 电压到 2.3 V 或更高。

（a）复位例程

<1> 将 SOSCAMCTL.AMPHS 清除为 0（特定序列），并设置副时钟（f_{XT}） 为正常振荡。
<2> 将 RTCBUMCTL0.RBMSET 清除为 0（特定序列），并退出 RTC 备份模式。
<3> NOP

（3）退出 RTC 备份模式（未发生外部复位时）

进入 RTC 备份模式之后，如果 V_{DD} 电压未降低到产生外部复位信号（$\overline{\text{RESET}}$）的程度，执行处理例程如下。

（a）INTLVI 中断例程

产生 INTVI 中断并退出正常 STOP 模式之后，执行下列处理例程。

<1> 读取低电压检测标志（LVIM.LVIF） 并确认 LVIM.LVIF = 0。
<2> 将 SOSCAMCTL.AMPHS 清除为 0（特定序列），并设置副时钟（f_{XT}） 为正常振荡。
<3> 将 RTCBUMCTL0.RBMSET 清除为 0（特定序列），并退出 RTC 备份模式。
<4> NOP
<5> 如果需要，允许未屏蔽的可屏蔽中断，EI 和 NMI。

图 24-2. RTC 备份模式设置过程示例

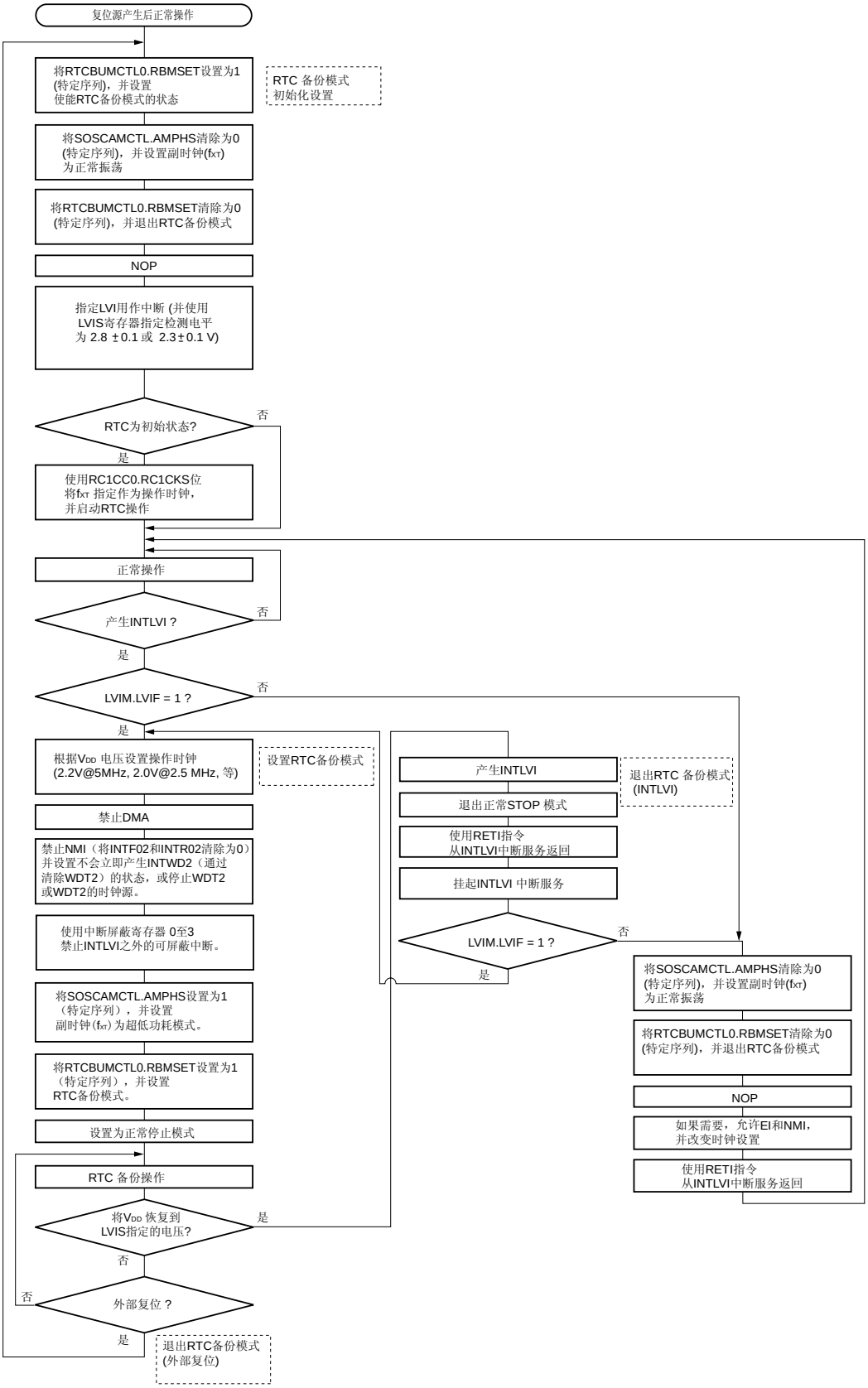
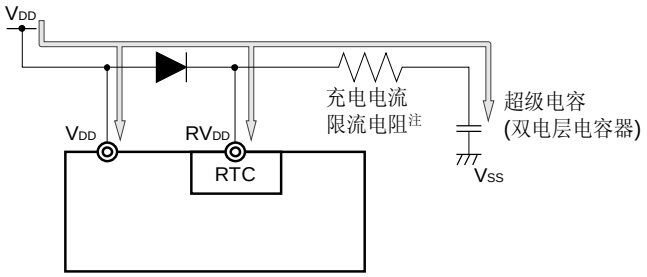
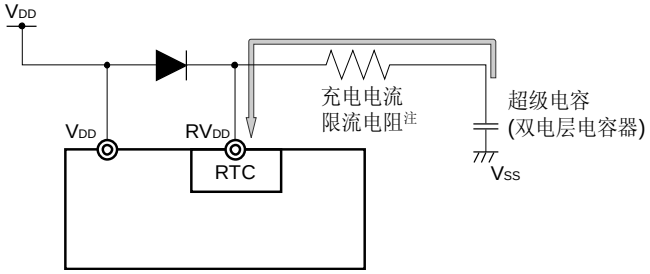


图 24-3. RTC 备份模式电源供给配置示例（简单电源供给）

(a) 正常操作



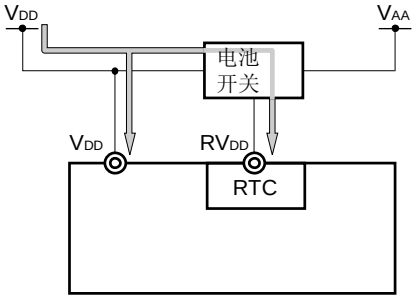
(b) RTC 备份模式



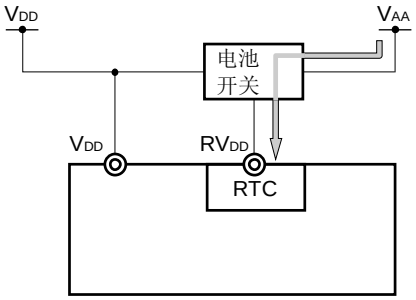
注 如果需要，对充电电流增加限制。

图 24-4. RTC 备份模式电源供给配置示例（双电源供给）

(a) 正常操作



(b) RTC 备份模式



第二十五章 复位功能

25.1 概述

复位功能用于对 V850ES/JG3-L 功能的设置进行初始化。比如，该功能可以用来在上电时停止操作，直到供电电压达到操作电压水平，或在任意时刻对 V850ES/JG3-L 功能的设置进行初始化。

复位结束之后，V850ES/JG3-L 立即由地址 00000000H 开始操作。

可以导致复位的复位源如下：

(1) 4 种类型的复位源

- 通过 $\overline{\text{RESET}}$ 引脚的外部复位输入
- 通过看门狗定时器 2(WDT2)的溢出复位(WDT2RES)
- 基于比较低电压检测器(LVI)供电电压和被检测电压而系统复位
- 基于检测到时钟监控器(CLM) 的振荡已经停止而系统复位

复位结束后，可以立即使用复位源标志寄存器(RESF)来确定复位源。

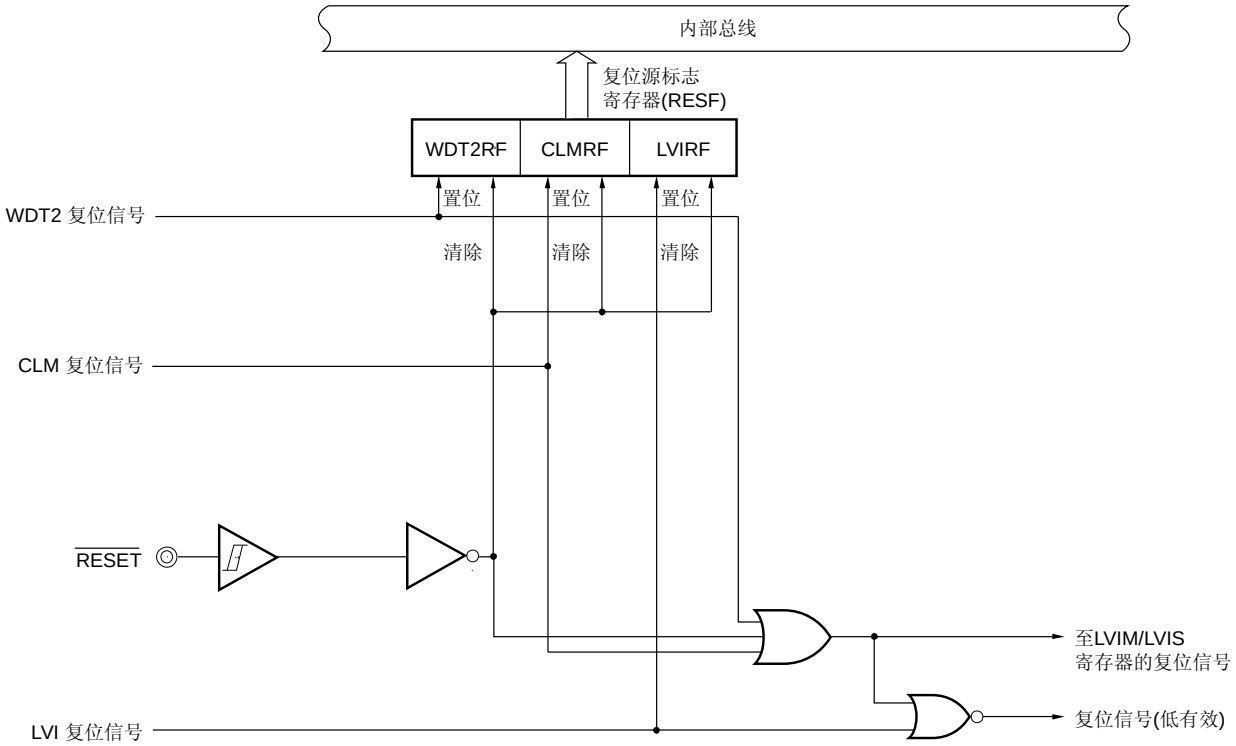
(2) 紧急操作模式

如果在复位后插入的主时钟振荡稳定期间内，WDT2 发生溢出，则主时钟振荡被判断为异常，且 CPU 开始运行于内部振荡时钟。

注意事项 紧急操作模式下，除了那些能够用内部振荡时钟工作的模块的寄存器，诸如“中断功能、端口功能、WDT2 和定时器 M”之类，不要访问其它片上外设 I/O 寄存器。此外，禁止使用外部时钟对 CSIB0 至 CSIB4 以及 UARTA0 进行操作。

25.2 配置

图 25-1. 复位功能框图



注意事项 一个 LVI 电路引起的内部复位信号不能复位 LVI 电路本身。

备注 **1.** LVIM: 低电压检测寄存器
2. LVIS: 低电压检测电平选择寄存器

25.3 检查复位源的寄存器

V850ES/JG3-L 单片机有 4 种类型的复位源。复位结束后，可以立即使用复位源标志寄存器(RESF)确认已经发生的复位源。

(1) 复位源标志寄存器 (RESF)

RESF 寄存器是一个专用寄存器，它只能通过一个特定序列的组合来写入(参见 3.4.7 专用寄存器)

RESF 寄存器表明复位信号由哪个源产生。

该寄存器可按字节或按位进行读取或写入。

$\overline{\text{RESET}}$ 引脚输入将该寄存器清为 00H。如果复位源不是 $\overline{\text{RESET}}$ 引脚的复位信号，其默认值将不同。

复位后：00H[注]　R/W　地址：FFFFF888H

	7	6	5	4	3	2	1	0
RESF	0	0	0	WDT2RF	0	0	CLMRF	LVIRF

WDT2RF	来自WDT2的复位信号
0	未产生
1	已产生

CLMRF	来自CLM的复位信号
0	未产生
1	已产生

LVIRF	来自LVI的复位信号
0	未产生
1	已产生

注 当通过 $\overline{\text{RESET}}$ 引脚执行一个复位时，RESF 寄存器的值被清为 00H。当通过看门狗定时器 2 (WDT2)、低电压检测器 (LVI)或时钟监视器(CLM)执行一个复位时，该寄存器的复位标志(WDT2RF 位、CLMRF 位和 LVIRF 位)被置位。而其它复位源时保持不变。

注意事项 只有“0”可以被写到该寄存器的每位。如果写入“0”和置位标志（发生复位时）发生冲突，则置位标志优先。

25.4 操作

25.4.1 通过$\overline{\text{RESET}}$引脚的复位操作

当 $\overline{\text{RESET}}$ 引脚输入低电平时，系统复位，各个硬件单元被初始化。

当 $\overline{\text{RESET}}$ 引脚的电平由低变为高时，复位状态结束。

$\overline{\text{RESET}}$ 引脚具有内部噪声消除电路，使用模拟延迟 (60 ns (典型值))来防止噪声导致的误操作。

表 25-1. $\overline{\text{RESET}}$ 引脚输入时的硬件状态

项目	复位期间	复位后
主时钟振荡器 (f_X)	振荡停止	振荡开始
副时钟振荡器 (f_{XT})	振荡继续	
内部 振荡器	振荡停止	振荡开始
外部时钟 (f_X 至 f_X/1,024)	操作停止	在确保振荡稳定时间之后开始操作
内部系统时钟 (f_{CLK}), CPU 时钟(f_{CPU})	操作停止	在确保振荡稳定时间之后开始操作(初始化至 f_{XX}/8)
CPU	初始化	在确保振荡稳定时间之后，开始由地址 00000000H 执行程序
看门狗定时器 2	操作停止 (初始化为 0)	用内部振荡时钟作源时钟，从 0 开始往上计数
RTC	操作继续	
内部 RAM	在加电复位时或 CPU 存取和复位输入冲突（数据被损坏）时，状态不确定；否则，保留复位后的瞬时值。	
I/O 线 (端口/复用功能引脚)	高阻[注]	
片上外设 I/O 寄存器	初始化到指定状态， OCDM 寄存器设为 (01H)。	
其它片上外设功能	操作停止	在确保振荡稳定时间之后开始操作

注　当电源开启时，即使在复位期间，下列引脚也可能会临时输出一个不确定的电平。

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

注意事项　OCDM 寄存器通过 $\overline{\text{RESET}}$ 引脚的输入来初始化。因此，必须注意的是：在复位结束之后 OCDM.OCDM0 位被清除之前，如果有高电平立即输入 P05/DRST 引脚，可能会进入片上调试模式。详情参见 第四章　端口功能。

图 25-2. $\overline{\text{RESET}}$ 引脚输入引起的复位操作时序

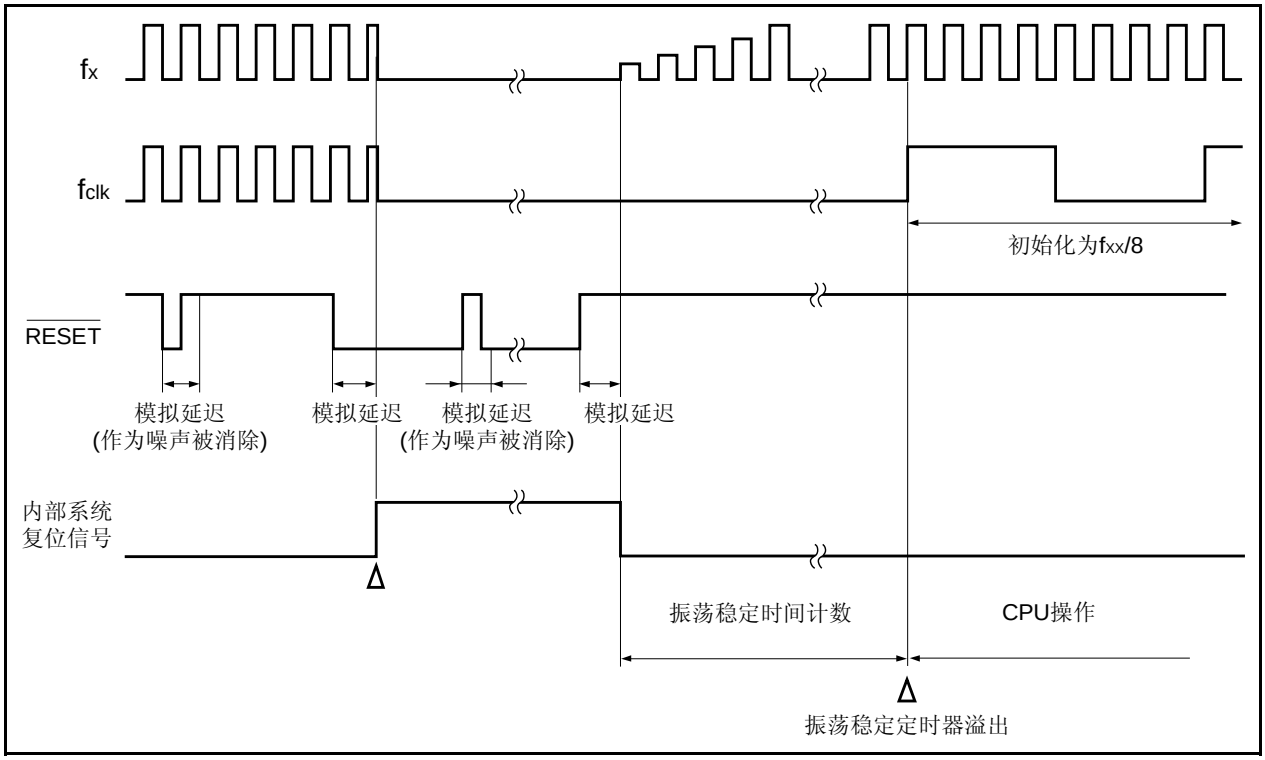
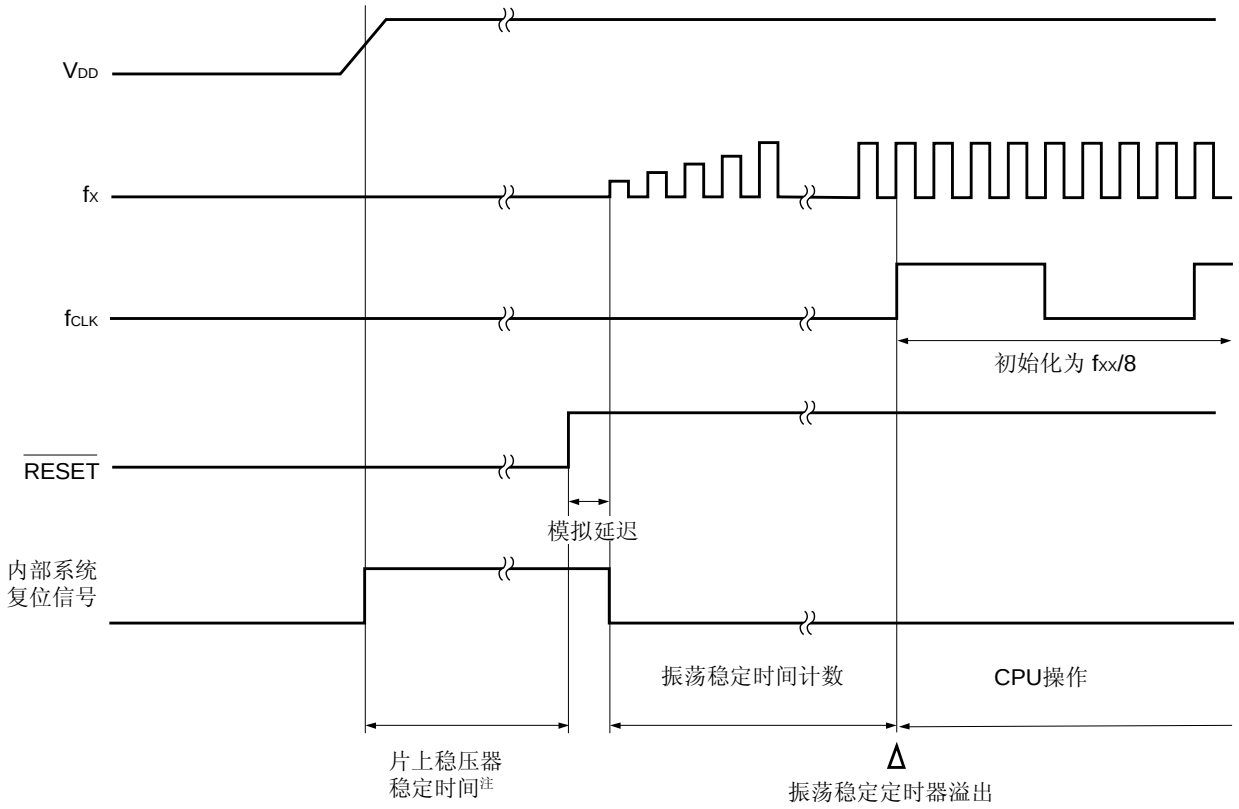


图 25-3. 上电复位操作的时序



注 VDD 达到 2.2 V 之后的时间：3.5 ms (最大值)
VDD 达到 2.7 V 之后的时间：1.0 ms (最大值)

注意事项 不要禁止 RESET，直到片上稳压器的电压稳定。

25.4.2 通过看门狗定时器 2 的复位操作

若看门狗定时器 2 被设置为溢出复位工作模式，当看门狗定时器 2 发生溢出时（产生 WDT2RES 信号），则执行系统复位，并且硬件被初始化为初始状态。

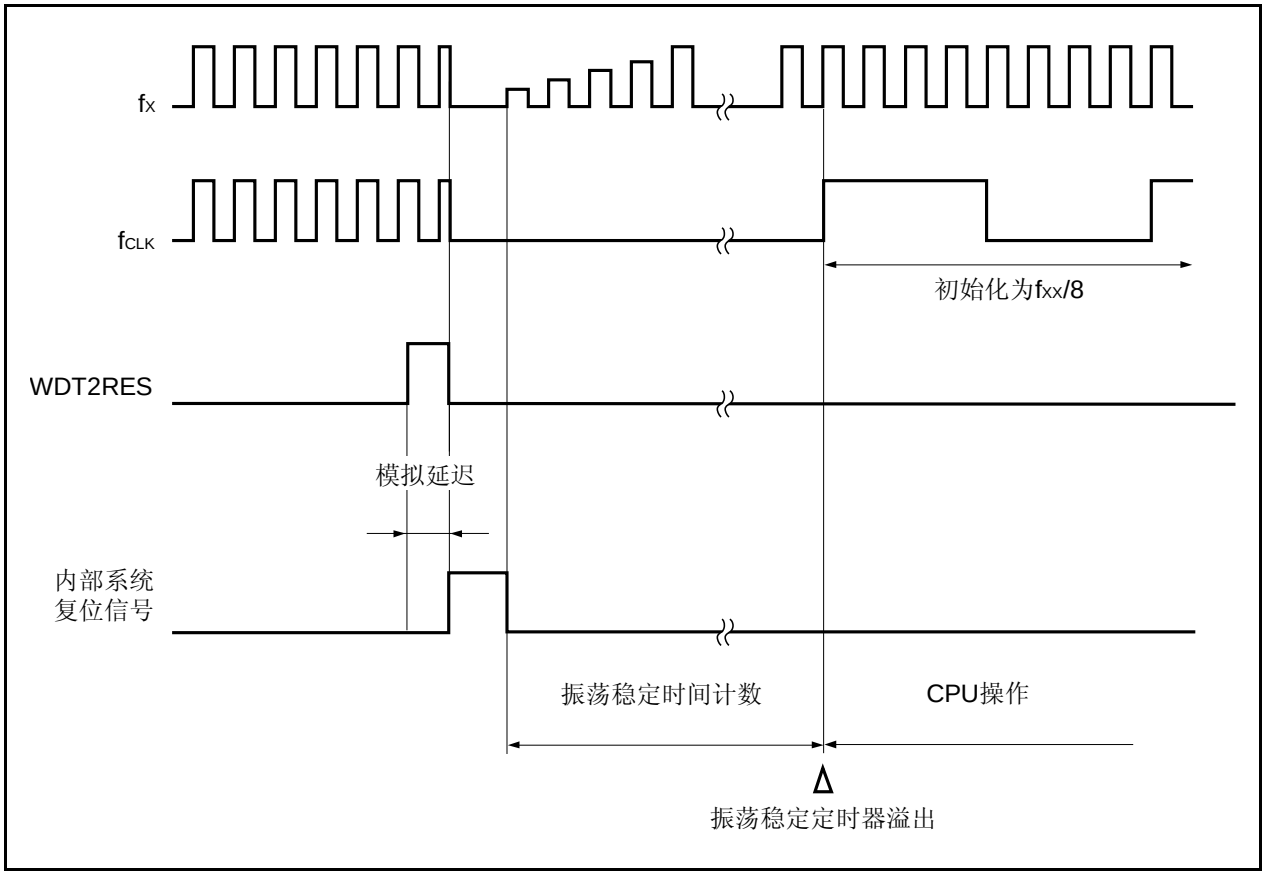
在看门狗定时器 2 溢出后，进入复位状态并持续预定时间(模拟延迟)，并且，复位状态随后自动结束。

在复位期间主时钟振荡器停止。

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)	振荡停止	振荡开始
副时钟振荡器 (f_{XT})	振荡继续	
内部振荡器	振荡停止	振荡开始
外部时钟(f_{XX} 至 f_{XX}/1,024)	操作停止	在确保振荡稳定时间之后开始操作
内部系统时钟(f_{XX}), CPU 时钟(f_{CPU})	操作停止	在确保振荡稳定时间之后开始操作(初始化为 f_{XX}/8)
CPU	初始化	在确保振荡稳定时间之后程序开始执行
看门狗定时器 2	操作停止 (初始化为 0)	内部振荡时钟作为源时钟，从 0 开始往上计数。
RTC	操作继续	
内部 RAM	在加电复位时或当 CPU 存取和复位输入冲突（数据被损坏）时，状态不确定，否则，保留复位后的瞬时值。	
I/O 线(端口/复用功能引脚)	高阻	
片上外设 I/O 寄存器	初始化为特定状态，保留 OCDM 寄存器的值。	
其它片上外设功能	操作停止	在确保振荡稳定时间之后开始操作。

图 25-4. 通过 WDT2RES 信号产生的复位操作时序



25.4.3 通过低电压检测器的复位操作

当 LVI 操作使能时，如果供电电压下降到低电压检测器的检测电压以下，执行一次系统复位（当 LVIM.LVIMD 位被设为 1 时），且硬件被初始化到初始状态。

从检测到供电电压下降一直持续到供电电压上升超过 LVI 检测电压，一直都保持在复位状态。

在复位期间主时钟振荡器停止。

当 LVIMD 位被清除为 0 时，如果检测到供电电压跌落到低于或超过检测电压，则产生一个中断请求信号(INTLVI)。

表 25-3. 由低电压检测器复位操作时的硬件状态

项目	复位期间	复位后
主时钟振荡器(f_X)	振荡停止	振荡开始
副时钟振荡器(f_{XT})	振荡继续	
内部振荡器 r	振荡停止	振荡开始
外部时钟(f_X 至 f_X/1,024)	操作停止	在确保振荡稳定时间之后开始操作
内部系统 时钟(f_{XX}), CPU 时钟(f_{CPU})	操作停止	在确保振荡稳定时间之后开始操作(初始化为 f_{XX}/8)
CPU	初始化	在确保振荡稳定时间之后程序开始执行
看门狗定时器 2	操作停止(初始化为 0)	内部振荡时钟作为源时钟，从 0 开始往上计数。
RTC	操作继续	
内部 RAM	不确定	
I/O 线(端口/复用功能引脚)	高阻	
片上外设 I/O 寄存器	初始化到指定状态，保留 OCDM 寄存器的值。	
LVI	操作停止	
其它片上外围功能	操作停止	在确保振荡稳定时间之后开始操作。

备注　　关于低电压检测器的复位时序，参见 第二十七章 低电压检测器 (LVI)。

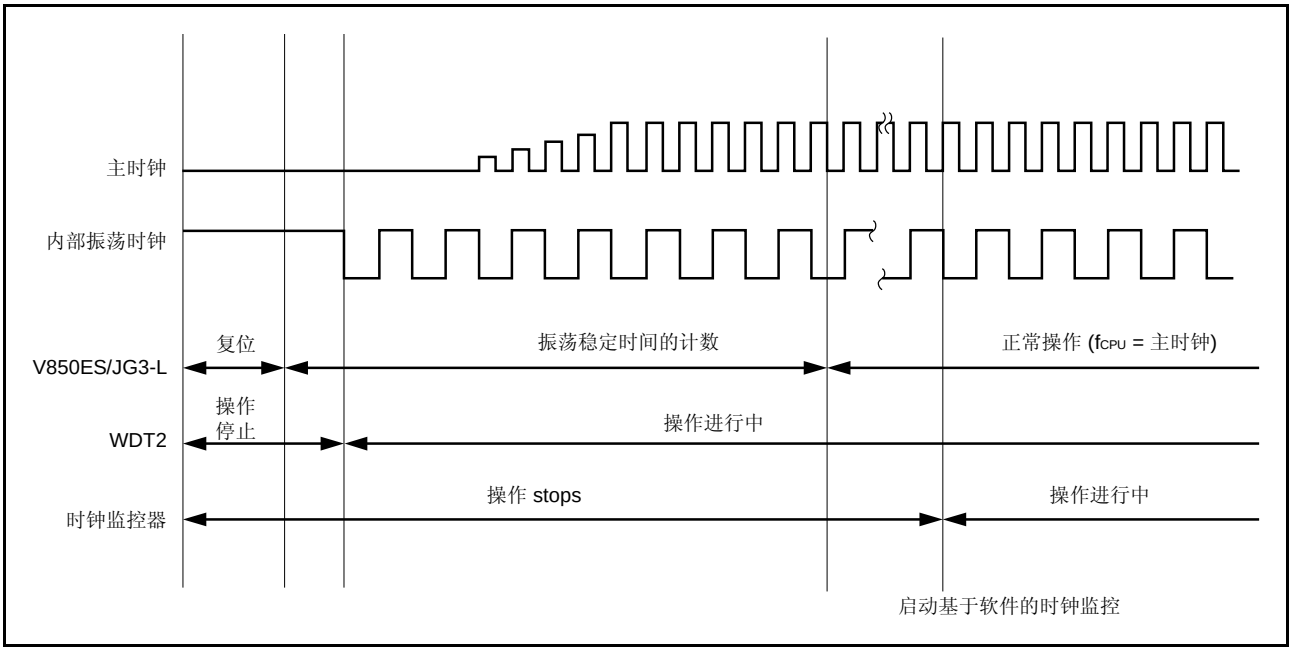
25.4.4 复位结束后的即时操作

(1) 复位正常结束后的即时操作

在复位结束后，主时钟立即开始振荡，且确保经过振荡稳定时间（随选项字节的设置不同而变化。详情参见 **第三十章 选项字节**）之后，CPU 开始执行程序。

在复位结束后，WDT2 立即开始运行，使用内部振荡时钟作为源时钟。

图 25-5. 复位结束后的即时操作

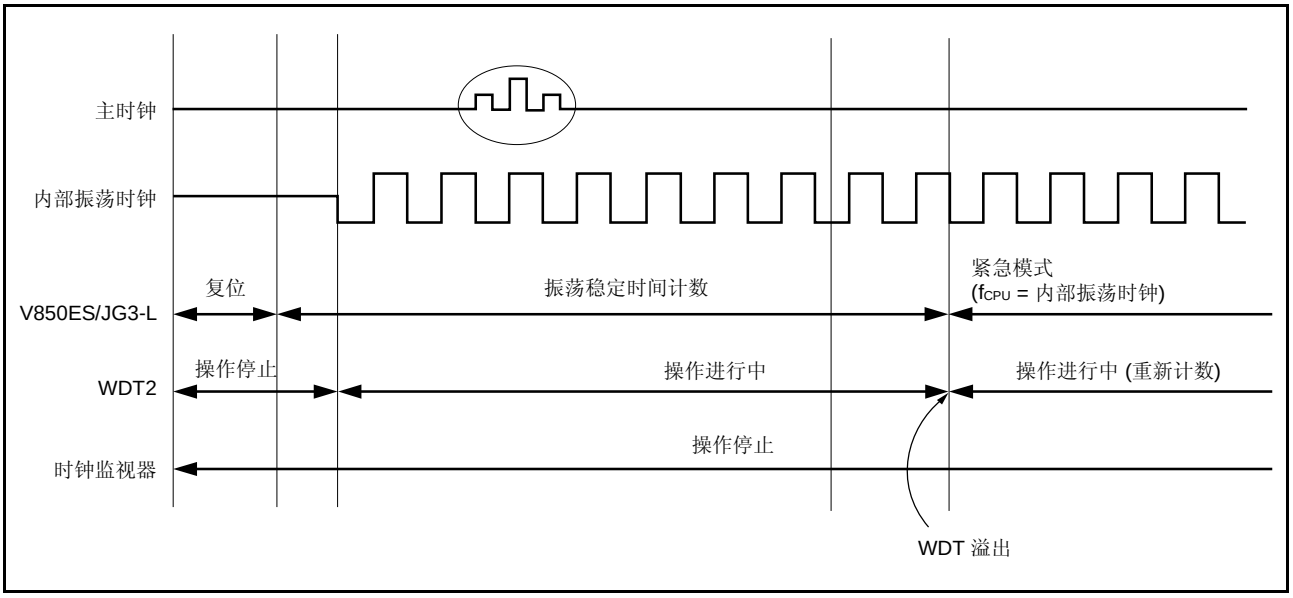


(2) 紧急操作模式

如果在振荡稳定时间完成之前，主时钟发生异常，则在 CPU 执行程序之前，WDT2 就已经溢出。此时，将内部振荡时钟作为源时钟，CPU 开始执行程序。

注意事项 紧急操作模式下，除了那些能够用内部振荡时钟工作的模块的寄存器，诸如“中断功能、端口功能、WDT2 和定时器 M”之类，不要访问其它片上外设 I/O 寄存器。此外，禁止使用外部时钟对 CSIB0 至 CSIB4 以及 UARTA0 进行操作。

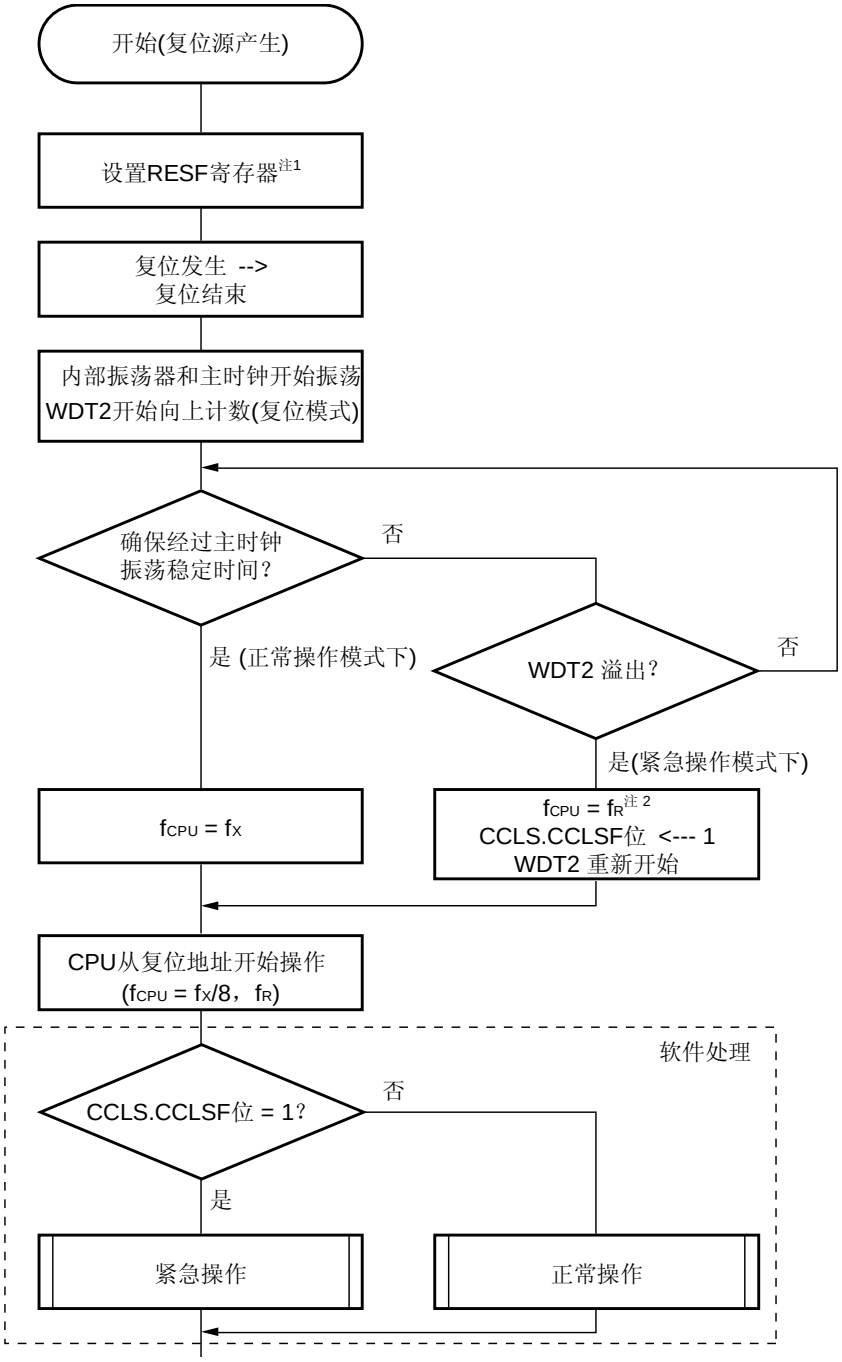
图 25-6. 复位结束后的即时操作



可以使用 CPU 操作时钟状态寄存器(CCLS)来检测 CPU 操作时钟的状态。

25.4.5 复位功能操作流程

图 25-7. 复位功能操作



注 **1.** 根据复位源对不同的位置位。

复位源	WDT2RF 位	CRMRF 位	LVIRF 位
$\overline{\text{RESET}}$ 引脚	0	0	0
WDT2	1	保留复位前的值。	保留复位前的值。
CLM	保留复位前的值。	1	保留复位前的值。
LVI	保留复位前的值。	保留复位前的值。	1

2. 内部振荡器不能停止。

25.5 注意事项

执行上电复位操作时，在复位结束之后，必须立即确保供电电压位于保证操作范围内。V850ES/JG3-L 内部操作频率的可用电压范围会根据供电电压而改变（2.5 MHz (MAX.) @ 2.0 至 2.2 V 或 5 MHz (MAX.) @ 2.2 至 2.7 V 或 20 MHz (MAX.) @ 2.7 至 3.6 V）。

(1) 复位刚结束后低于 2.2V

禁止使用。

(2) 复位刚结束后在 2.0V 至 2.2V 之间

- 主时钟振荡器输入 f_X = 2.5 MHz，且将时钟设为直通模式(PLLCTL.SELPLL = 0)。
- 主时钟禁止输入 2.5 MHz 或更高的频率。
- 确保在初始化例程中停止 PLL (PLLCTL.PLLON = 0)。

(3) 复位刚结束后在 2.2V 至 2.7V 之间

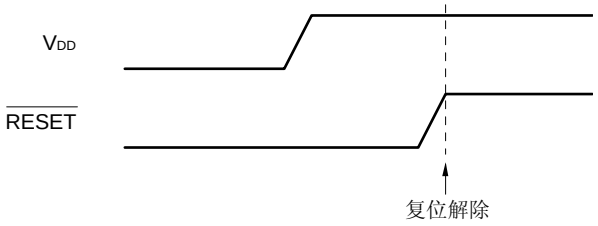
- 主时钟振荡器输入 f_X = 2.5 至 5 MHz，且将时钟设为直通模式(PLLCTL.SELPLL = 0)。
- 主时钟禁止输入 5MHz 或更高的频率。
- 确保在初始化例程中停止 PLL (PLLCTL.PLLON = 0)。

(4) 复位刚结束后在 2.7 至 3.6V 之间

- 时钟直通模式和 PLL 模式都可以使用。

备注 **1.** 电压值(V) 就是 V_{DD} 的值。

2. 复位结束时序如下。关于 V_{DD} 的上升与通过 $\overline{\text{RESET}}$ 引脚产生的复位信号结束之间的关系，参见 **33.7.4 上电/下电/复位时序**。



第二十六章 时钟监控器

26.1 功能

时钟监控器使用内部振荡时钟，监控主时钟，当主时钟振荡停止时，产生复位请求信号。
一旦通过操作使能标志允许时钟监控器的操作，则除了复位之外，没有办法可以使之清 0。
当时钟监控器引发复位时，RESF.CLMRF 位置位。关于 RESF 寄存器的细节，参见 **25.3 检查复位源寄存器。**
在下列条件时，时钟监控器自动停止：

- 解除 STOP 模式后的振荡稳定期间
- 主时钟停止时（从副时钟工作期内 PCC.MCK 位 = 1 时开始，直至主时钟工作期内 PCC.CLS 位 = 0）
- 当监控的时钟（内部振荡时钟）停止时
- 当 CPU 运行于内部振荡时钟工作时

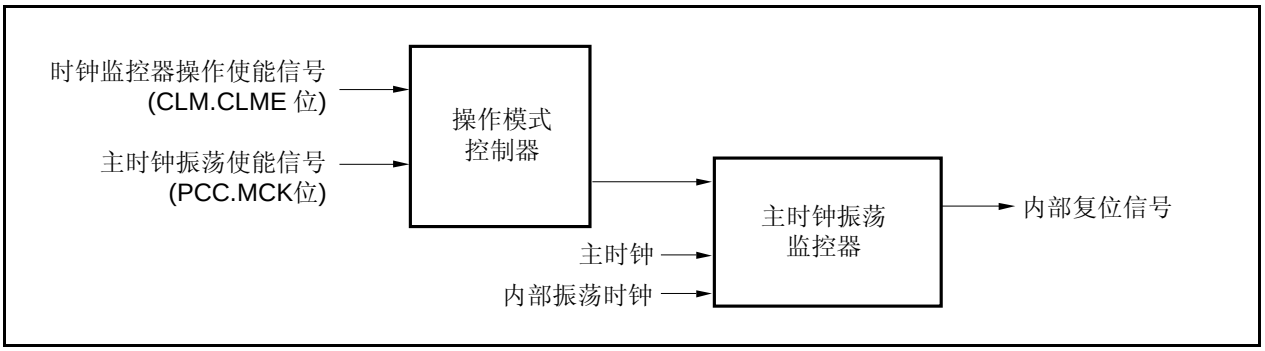
26.2 配置

时钟监控器包含以下硬件。

表 26-1. 时钟监控器的配置

项目	配置
控制寄存器	时钟监控模式寄存器 (CLM)

图 26-1. 时钟监控器的框图



26.3 寄存器

时钟监控器由时钟监控器模式寄存器(CLM)控制。

(1) 时钟监控器模式寄存器(CLM)

CLM 寄存器是专用寄存器。只有通过特定序列的组合才能写入该寄存器（见 3.4.7 **专用寄存器**）。

该寄存器用于设置时钟监控器的操作模式。

该寄存器可以按字节或按位进行读写。

系统复位后，设置该寄存器为 00H。

复位后：00H　R/W　地址：FFFFF870H

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

CLME	时钟监控器操作使能或禁止
0	禁止时钟监控器操作。
1	允许时钟监控器操作。

注意事项 1. 一旦 CLME 位设置为 1，除非复位，不能以任何其它方式将其清 0。
2. 当时钟监控器引发复位时，CLME 位被清为 0，而 RESF.CLMRF 位被置为 1。

26.4 操作

本节描述时钟监控器的操作。监控器启动条件和监控器停止条件如下：

<监控器启动条件>

将 CLM.CLME 位设为 1，允许操作。

<监控器停止条件>

- 解除 STOP 模式后对振荡稳定时间进行计数时
- 当主时钟停止时（副时钟工作期间，从 PCC.MCK 位 = 1 时开始，直到主时钟工作期内 PCC.CLS 位 = 0）
- 当 采样时钟(内部振荡时钟)停止时
- 当 CPU 运行于内部振荡器时钟时

表 26-2. 时钟监控器操作状态

（内部振荡器时钟工作期间，CLM.CLME 位 = 1 时）

CPU 工作时钟	操作模式	主时钟状态	内部振荡时钟状态	时钟监控器状态
主时钟	HALT 模式	振荡	振荡[注1]	工作[注2]
	IDLE1，IDLE2 模式	振荡	振荡[注1]	工作[注2]
	STOP 模式	停止	振荡[注1]	停止
副时钟 (PCC 寄存器的 MCK 位 = 0)	副 IDLE 模式	振荡	振荡[注1]	工作[注2]
副时钟(PCC 寄存器 MCK 位=1)	副 IDLE 模式	停止	振荡[注1]	停止
内部振荡时钟	紧急操作模式[注2]	停止	振荡[注3]	停止
复位期间	–	停止	停止	停止

注 **1.** 通过设置 RCM.RSTOP 位为 1，可以停止内部振荡器。

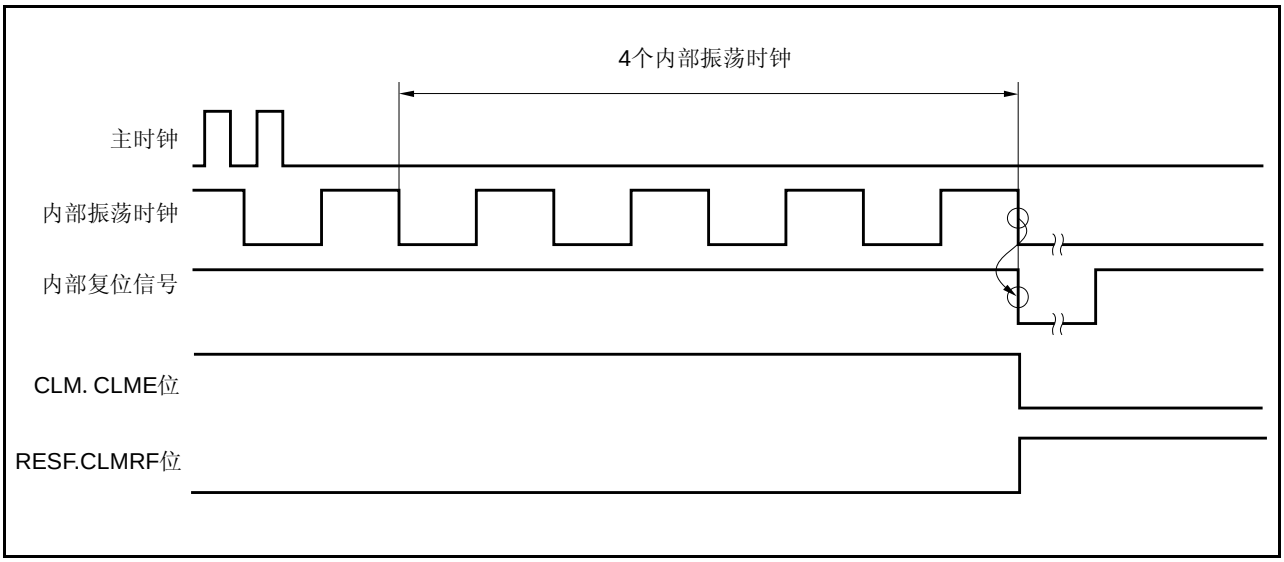
2. 参见 **25.4.4 (2) 紧急操作模式**。

3. 内部振荡器不能通过软件停止。

(1) 主时钟振荡停止时的操作 (CLME 位=1)

如果在 CLME 位为 1 时主时钟振荡停止，则产生一个内部复位信号，如图 26-2 所示。

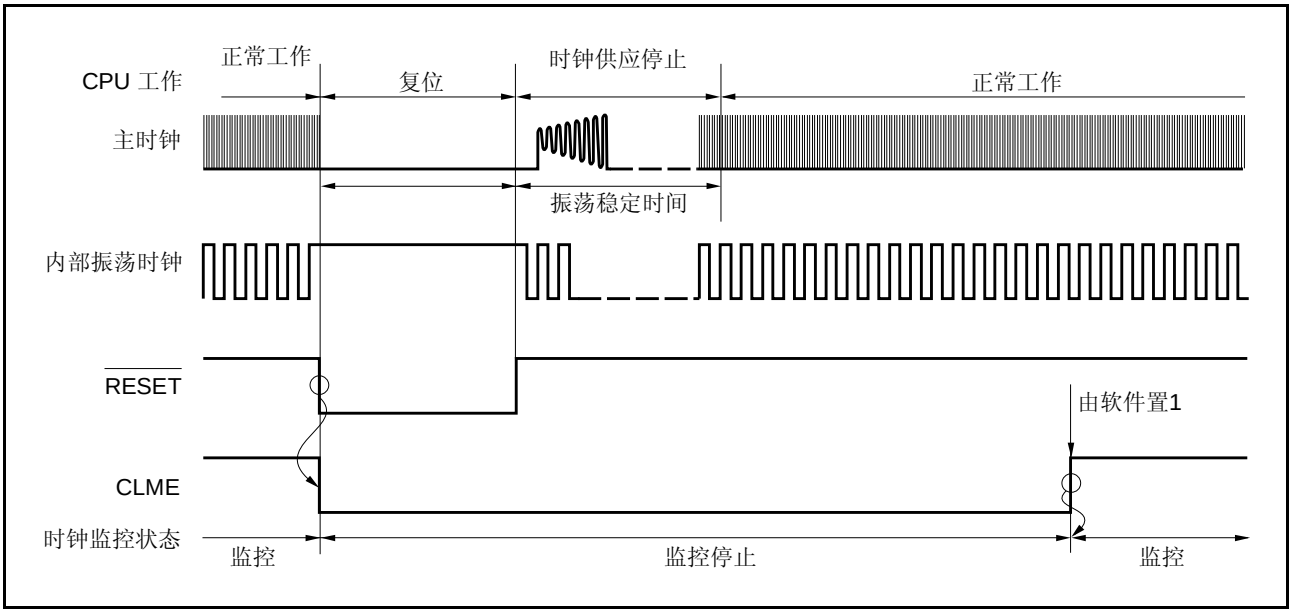
图 26-2. 因主时钟振荡器的停止导致的复位时间



(2) $\overline{\text{RESET}}$ 输入后时钟监控器的状态

$\overline{\text{RESET}}$ 输入将 CLM.CLME 位清为 0，并停止时钟监控器的操作。在正常操作开始之后，由软件将 CLME 位设为 1，则开始监控。

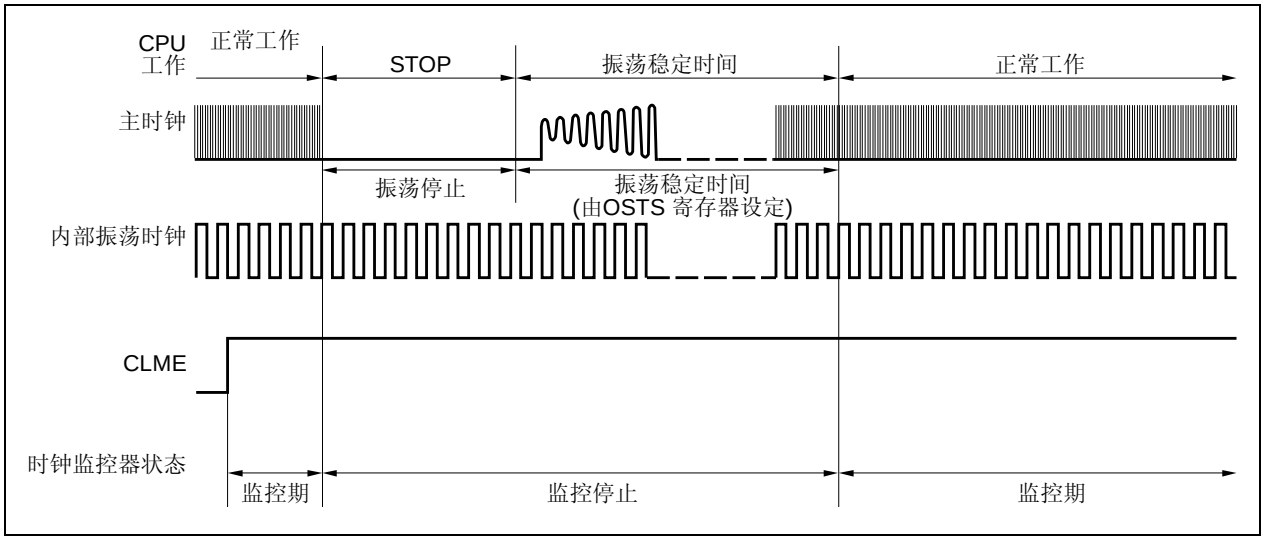
图 26-3. $\overline{\text{RESET}}$ 输入后的时钟监控器状态
($\overline{\text{RESET}}$ 输入且正常操作开始之后设置 CLM.CLME 位 = 1)



(3) 在 STOP 模式下或 STOP 模式解除后的操作

如果在 CLM.CLME 位 = 1 时设置 STOP 模式，那么监控器会在 STOP 模式下停止工作，在振荡稳定时间计数期间也停止工作。振荡稳定时间过后，监控器将会自动开始工作。

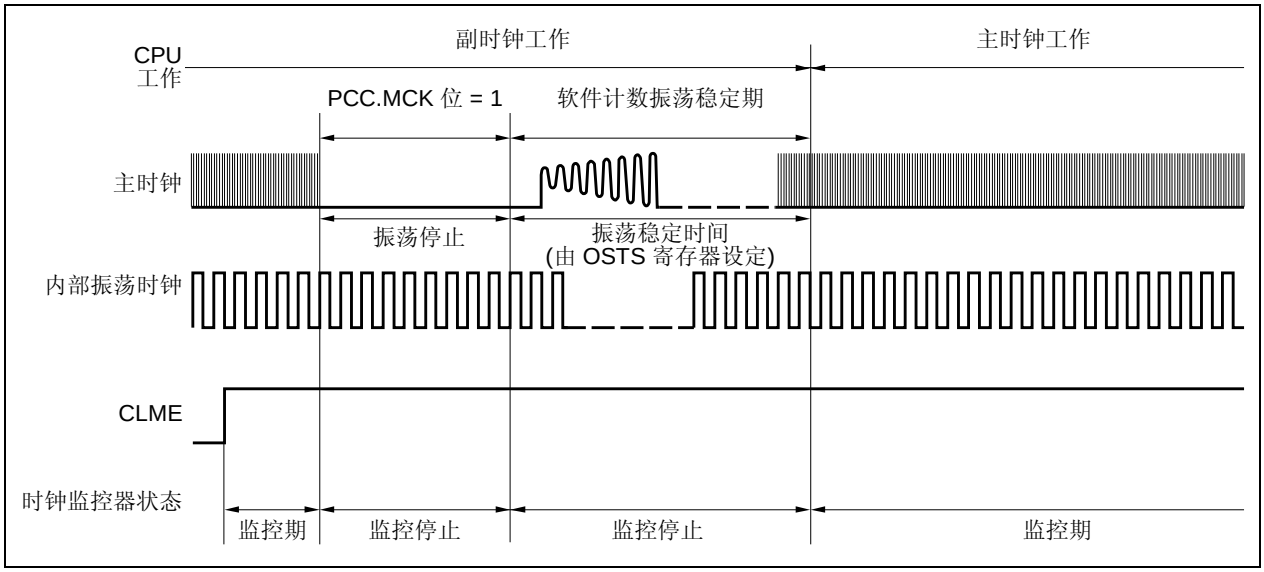
图 26-4. 在 STOP 模式下或 STOP 模式解除后的操作



(4) 主时钟停止(任意)时的操作

在副时钟操作 (PCC.CLS 位 = 1)期间，或通过设置 PCC.MCK 为 1 使主时钟停止工作时，监控器操作停止，直到主时钟开始工作(PCC.CLS 位 = 0)。当主时钟开始工作时，监控器自动开始工作。

图 26-5. 主时钟停止(任意)时的操作



(5) 当 CPU 运行于内部振荡时钟(CCLS.CCLSF 位= 1)时的操作

当 CCLSF 位为 1 时，即使 CLME 位被置为 1，监控器也不会停止工作。

第二十七章 低电压检测器 (LVI)

27.1 功能

低电压检测器(LVI) 具有以下功能。

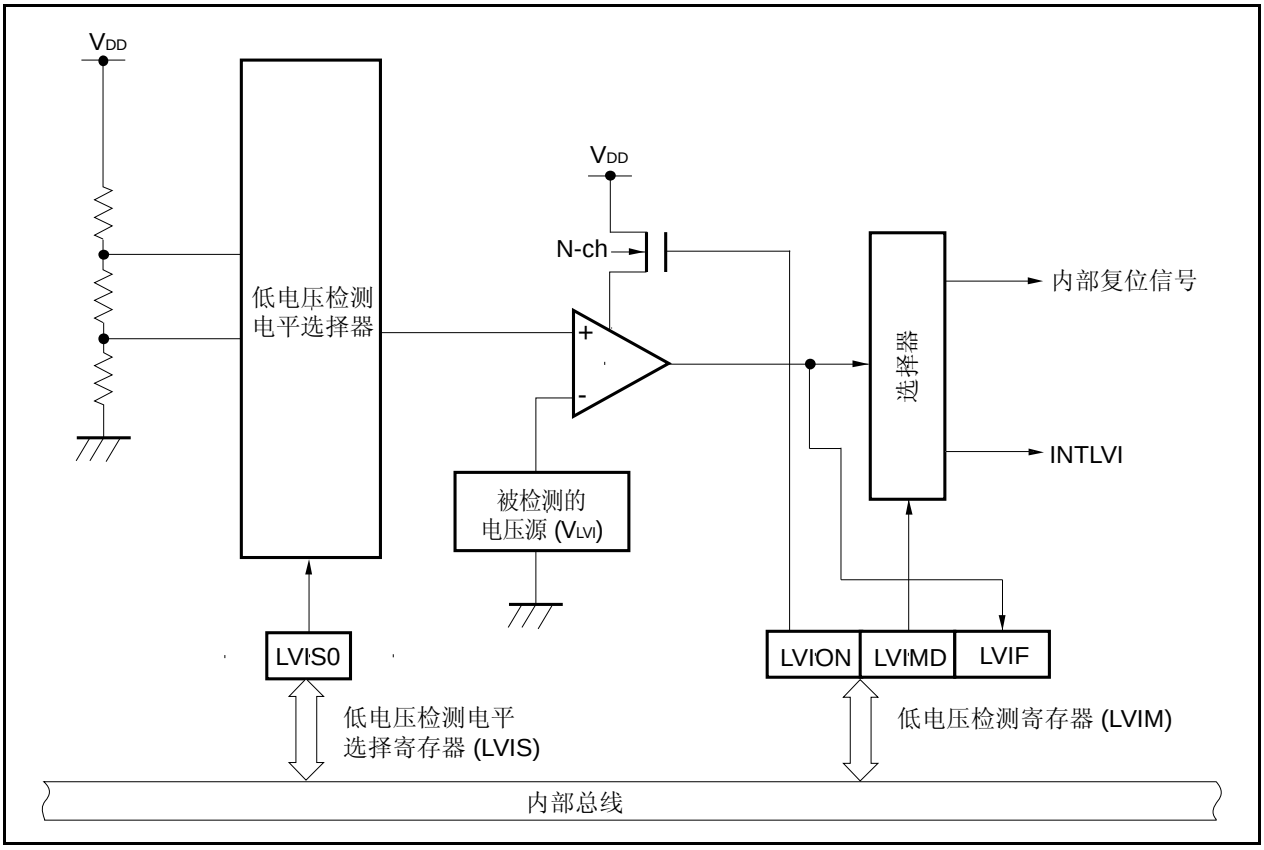
- 如果在使用低电压检测后选择发生中断，低电压检测器比较供电电压(V_{DD}) 和检测电压 (V_{LVI})，并且，在供电电压下降或上升经过检测电压时，产生一个内部中断信号。
- 如果在使用低电压检测后选择发生复位，当供电电压 (V_{DD}) 下降低于检测电压 (V_{LVI})时，低电压检测器产生一个内部复位信号。
- 供电电压的检测电平可由软件改变。
- 可由软件选择为中断或复位信号。
- 低电压检测器可以在待机模式下工作。

低电压检测器选择产生复位信号，当复位信号产生时 RESF.LVIRF 位被设置为 1。关于 RESF 寄存器的详情，参见 **25.3 检查复位源的寄存器。**

27.2 配置

低电压检测器的框图如下所示。

图 27-1. 低电压寄存器框图



27.3 寄存器

低电压检测器由下列寄存器控制。

- 低电压检测寄存器 (LVIM)
- 低电压检测电平选择寄存器 (LVIS)

(1) 低电压检测寄存器 (LVIM)

LVIM 是一个专用寄存器。只有通过特定序列的组合才能写入该寄存器（参见 **3.4.7 专用寄存器**）。
LVIM 寄存器用于使能或禁止低电压检测，并且设置低电压检测器的工作模式。
该寄存器可以按字节或按位进行读取或写入，但是， LVIF 位为只读位。

复位后：**注 1** R/W 地址： FFFFF890H

	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF

LVION	使能或禁止低电压检测操作
0	禁止操作
1	使能操作

LVIMD	低电压检测的工作模式选择
0	当供电电压下降低于或上升超过被检测电压时，产生中断信号 INTLVI。
1	当 供电电压下降低于检测电压时，产生内部复位信号 LVIRES。

LVIF 注 2, 3, 4	低电压检测标志
0	当供电电压上升超过检测电压，或操作禁止时。
1	当前接入电源的供电电压下降低于检测电压时。

注 **1.** 由 低电压检测复位： 82H
由其它复位源复位： 00H
2. 在供电电压 (V_{DD})低于检测电压 (V_{LVI}) (LVIM.LVIF 位 = 1)时，不要将 LVION 位由 1 改变为 0 。
3. 在 LVI 开始工作 (LVION 位 = 1) 或发生 INTLVI 后，检查 LVIF 位。
4. 产生 INTLVI 信号时，检查 LVIF 位以确认供电电压是否下降低于或超过检测电压。

注意事项 1. 当 LVION 和 LVIMD 位被置为 1 时，低电压检测器不能停止，除非低电压检测之外的其他复位源产生复位信号。
2. 当 LVION 位设置为 1 时，LVI 电路内的比较器开始工作。在 LVION 位置位后，使用 LVIF 位检查电压之前，应该由软件控制等待 0.2 ms 或更长的时间。
3. 请确保将第 6 位至第 2 位清为 “0”。

(2) 低电压检测电平选择寄存器 (LVIS)

LVIS 寄存器用于选择电压检测的检测电平值。

该寄存器能够按字节进行读取或写入。

复位后： **注 1** R/W 地址： FFFFF891H

	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	0	LVIS1	LVIS0

LVIS1	LVIS0	低电压检测电平
0	0	2.80 V (典型值)
0	1	2.30 V (典型值)
1	0	2.10 V (典型值)
1	1	禁止设置

注 由低电压检测复位: 保留

由其它复位源复位: 00H

注意事项 1. 该寄存器不能写入，除非在 LVIM.LVION 和 LVIM.LVIMD 位被设为 1 以后，低电压检测之外的其他复位源产生复位请求。

2. 请确保将第 7 位至第 2 位清 “0”。

27.4 操作

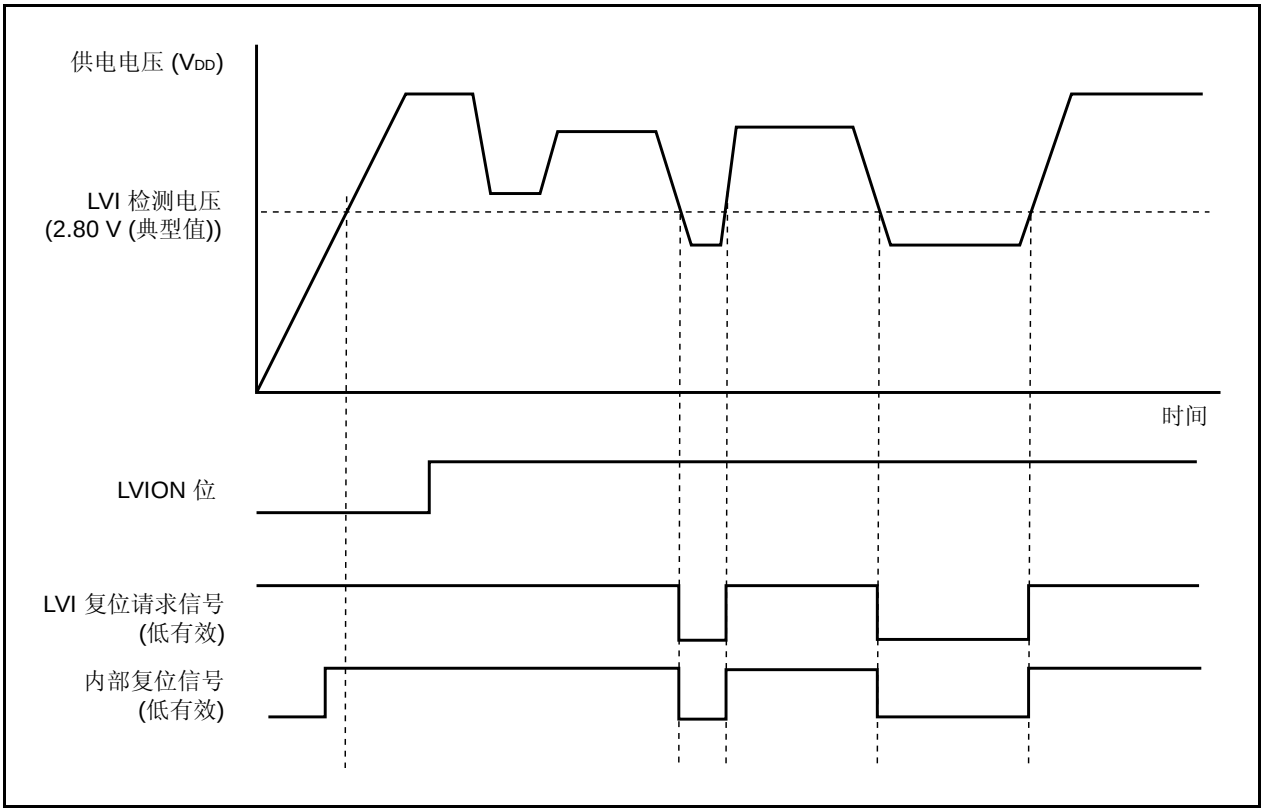
由 LVIM.VIMD 位的设置来决定，产生中断信号 (INTLVI)或内部复位信号。
如何指定每个操作的描述和时序图如下。

27.4.1 使用内部复位信号

<开始操作>
<1> 屏蔽 LVI 中断。
<2> 通过 LVIS.LVIS0 位选择检测电压。
<3> 将 LVIM.LVION 位设为 1 (使能操作)。
<4> 通过软件插入一个 0.2 ms 或更长的等待周期。
<5> 使用 LVIM.LVIF 位，检测是否供电电压低于检测电压。
<6> 将 LVIMD 位设为 1 (产生一个内部复位信号)。

注意事项 如果 LVIMD 位被设为 1，LVIM 和 LVIS 寄存器的内容不能改变，除非产生一个 LVI 之外的复位请求。

图 27-2. 低电压检测器操作时序(LVIMD 位= 1，低电压检测电平：2.80 V)

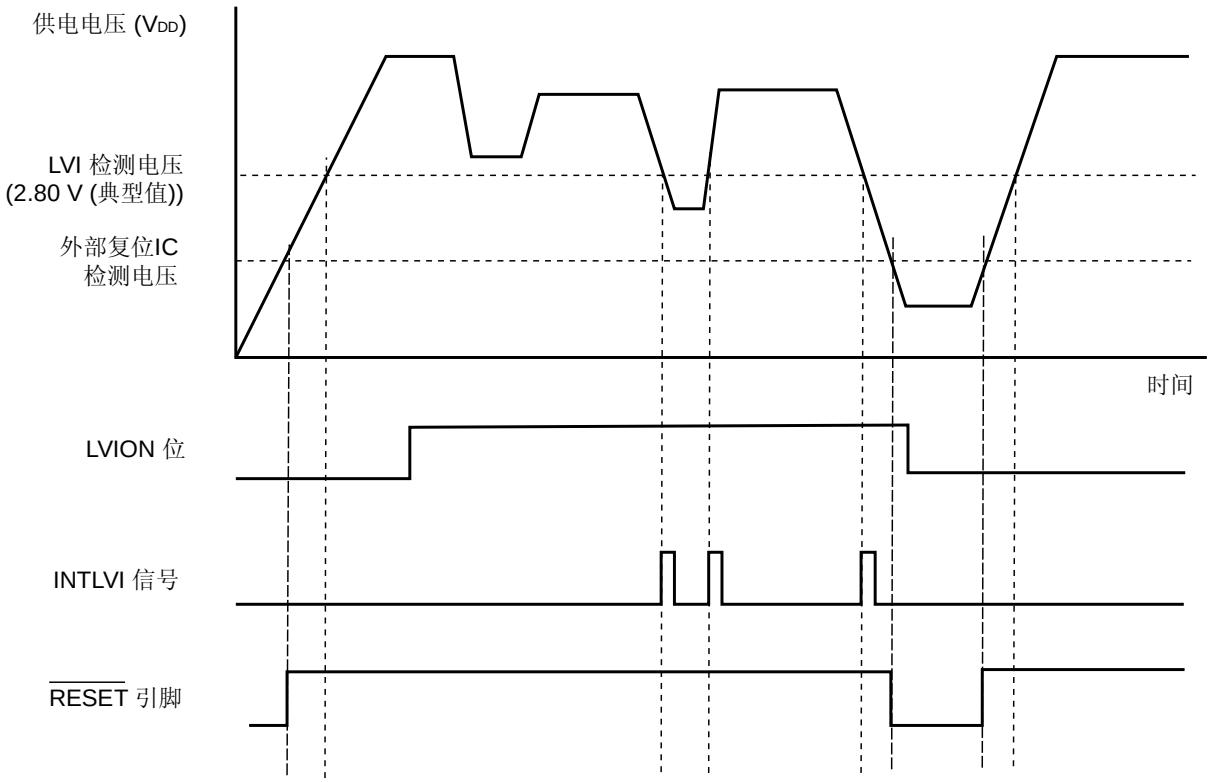


27.4.2 用作中断

<开始操作>
<1> 屏蔽 LVI 的中断。
<2> 通过 LVIS.LVIS0 位选择检测电压。
<3> 将 LVIM.LVION 位置为 1 (使能操作)。
<4> 通过软件插入一个 0.2 ms(最大值)或更长的等待周期。
<5> 使用 LVIM.LVIF 位，检测是否供电电压大于检测电压。
<6> 清除 LVI 的中断请求标志。
<7> 解除 LVI 的中断屏蔽。

<停止操作>
<1> 使用 LVIM.LVIF 位，检测是否供电电压大于检测电压。
<2> 将 LVION 位清 0。

图 27-3. 低电压检测器的操作时序(LVIMD 位= 0，低电压检测电平：2.80 V)



注意事项 当产生 INTLVI 信号后，通过 LVIM.LVIF 位确认是否因为供电电压下降低于或上升超过检测电压而产生 INTLVI 信号。

第二十八章 CRC 功能

28.1 功能

- 为检测通信数据中的错误而产生CRC（循环冗余校验）码
- 为检测数据块中的错误而产生 CRC 码
- 使用 CRC 功能电路进行数据块的错误检测。
- 使用 CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$)多项式以 8 位为单元对任何长度的数据块产生 16 位 CRC 校验码。
- 初始值设置到 CRCD 寄存器之后，每当 1 个字节数据被传送到 CRCIN 寄存器，CRC 码就被置入 CRC 数据寄存器。

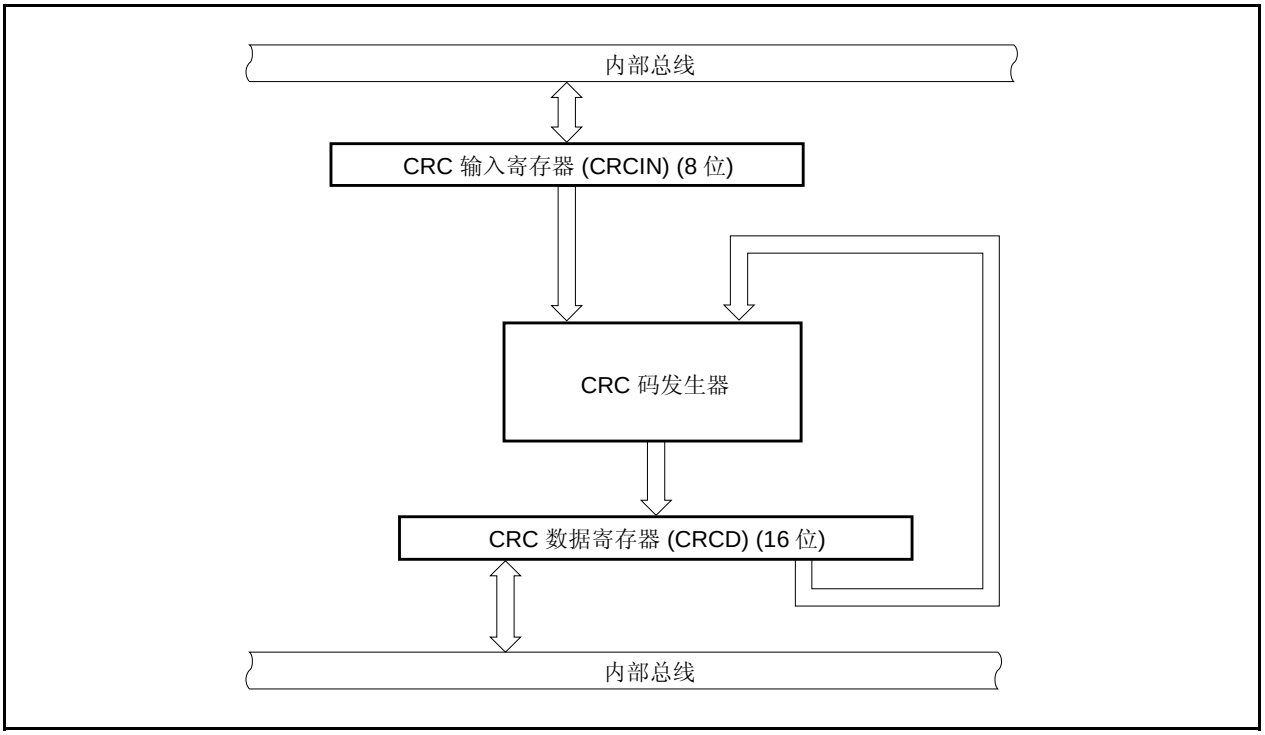
28.2 配置

CRC 控制功能包含下列硬件：

表 28-1. 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 28-1. CRC 寄存器的功能框图



28.3 寄存器

(1) CRC 输入寄存器 (CRCIN)

CRC 输入寄存器是 8 位寄存器，用于设置数据。

该寄存器可按字节进行读写。

系统复位后，该寄存器被设为 00H。

复位后: 00H　R/W　地址: FFFFF310H

	7	6	5	4	3	2	1	0
CRCIN								

(2) CRC 数据寄存器 (CRCIN)

CRCD 寄存器为 16 位寄存器，用来存储 CRC-CCITT 运行结果。

该寄存器可按 16 位宽度进行读写。

系统复位后，该寄存器被清为 0000H。

注意事项 在以下状态禁止访问 CRCD 寄存器。如果产生一个等待周期，只能通过复位来清除。详情参见 3.4.9 (1) 访问片上专用外设 I/O 寄存器。

- **当 CPU 运行于副时钟时，而主时钟振荡停止。**
- **当 CPU 运行于内部振荡器时钟工作时。**

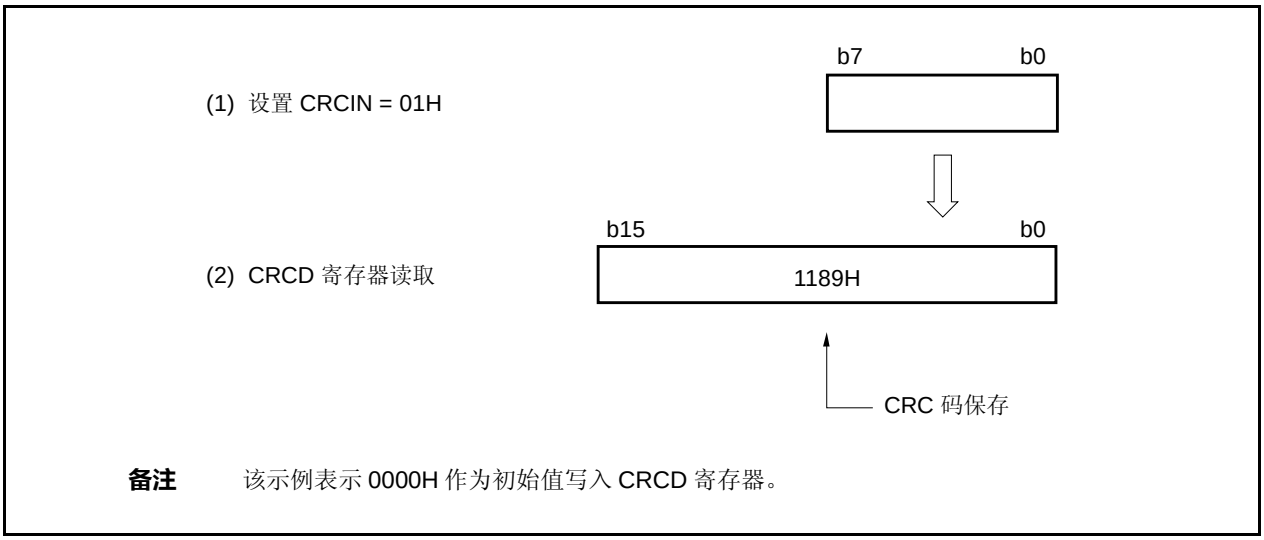
复位后: 0000H　R/W　地址: FFFFF312H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

28.4 操作

CRC 电路的举例如下。

图 28-2. CRC 电路操作示例 (LSB 先行)



01H 按照 LSB 先行发送的码为(1000 0000)。因此，在(1000 0000)后面添加 16 位的 0，使该码变为 (1000 0000 0000 0000 0000 0000)，然后再使用以 2 为模（modulo-2）操作公式，除以(1 0001 0000 0010 0001)，由生成多项式 $X^{16} + X^{12} + X^{5} + 1$ 得到的 CRC 码作为余数。

以 2 为模（modulo-2）操作基于以下公式进行。

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0
–1 = 1

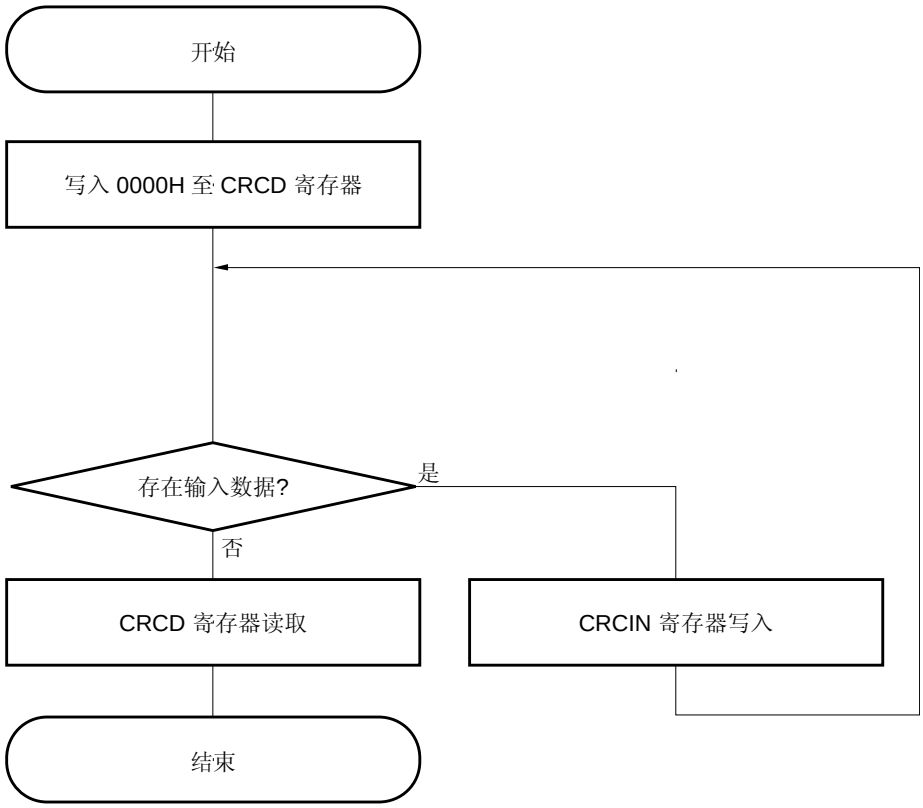
```
                                   LSB                                  MSB
                                                        1000 1000
1 0001 0000 0010 0001 / 1000 0000 0000 0000 0000 0000
                        1000 1000 0001 0000 1
                             1000 0001 0000 1000 0
                             1000 1000 0001 0000 1
                                  1001 0001 1000 1000
                                  LSB                  MSB
```

所以，CRC 码成为 $\overbrace{1001}^{9}\ \overbrace{0001}^{8}\ \overbrace{1000}^{1}\ \overbrace{1000}^{1}$ 由于使用 LSB 先行方式，则对应 16 进制表示的 1189H。

28.5 使用方法

以下描述如何使用 CRC 逻辑电路。

图 28-3. CRC 操作流程



[基本使用方法]

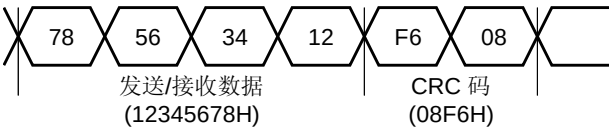
写入 0000H 至 CRCD 寄存器。

<2> 写入所需的数据数量至 CRCIN 寄存器。

<3> 读取 CRCD 寄存器。

当发送/接收数据由多个字节组成，如果同发送/接收数据一起发送/接收其 CRC 码，通信错误可以很容易得被检测出来。下面以发送 LSB 先行的 12345678H (0001 0010 0011 0100 0101 0110 0111 1000B)为例，进行解释说明。

图 28-4. CRC 发送举例



发送端的处理。

<1> 写入初始值 0000H 至 CRCD 寄存器。

<2> 写入待发送的 1 字节数据至发送缓存寄存器。（与此同时，相同数据被写入 CRCIN 寄存器）

<3> 当发送多字节数据时，每次发送数据被写入发送缓存寄存器时，写入相同数据至 CRCIN 寄存器。

<4> 在所有数据已发送后，将 CRCD 寄存器 (CRC 码)的内容写入发送缓存寄存器并进行发送。（数据按照 LSB 先行方式发送，由低字节开始发送数据，然后才是高字节。）

<5> 如果发送端要求重发，则重发该数据。

接收端的处理。

<1> 写入初始值 0000H 至 CRCD 寄存器。

<2> 当第一个字节接收完毕时，将接收数据写入 CRCIN 寄存器。

<3> 如果接收多个字节数据，每次接收完成时都将接收数据写入 CRCIN 寄存器。（正常接收情况下，当所有接收数据都已写入 CRCIN 寄存器，接收端的 CRCD 寄存器的内容应该和发送端 CRCD 寄存器的内容相同。）

<4> 然后，CRC 码从发送端发出，与接收数据相似，将该数据写入 CRCIN 寄存器。

<5> 当接收完包括 CRC 码在内的所有数据后，如果 CRCD 寄存器的内容为 0000H，则接收正常。如果 CRCD 寄存器的内容不是 0000H，这表示发生通信错误，所以，向发送端发送重发请求。

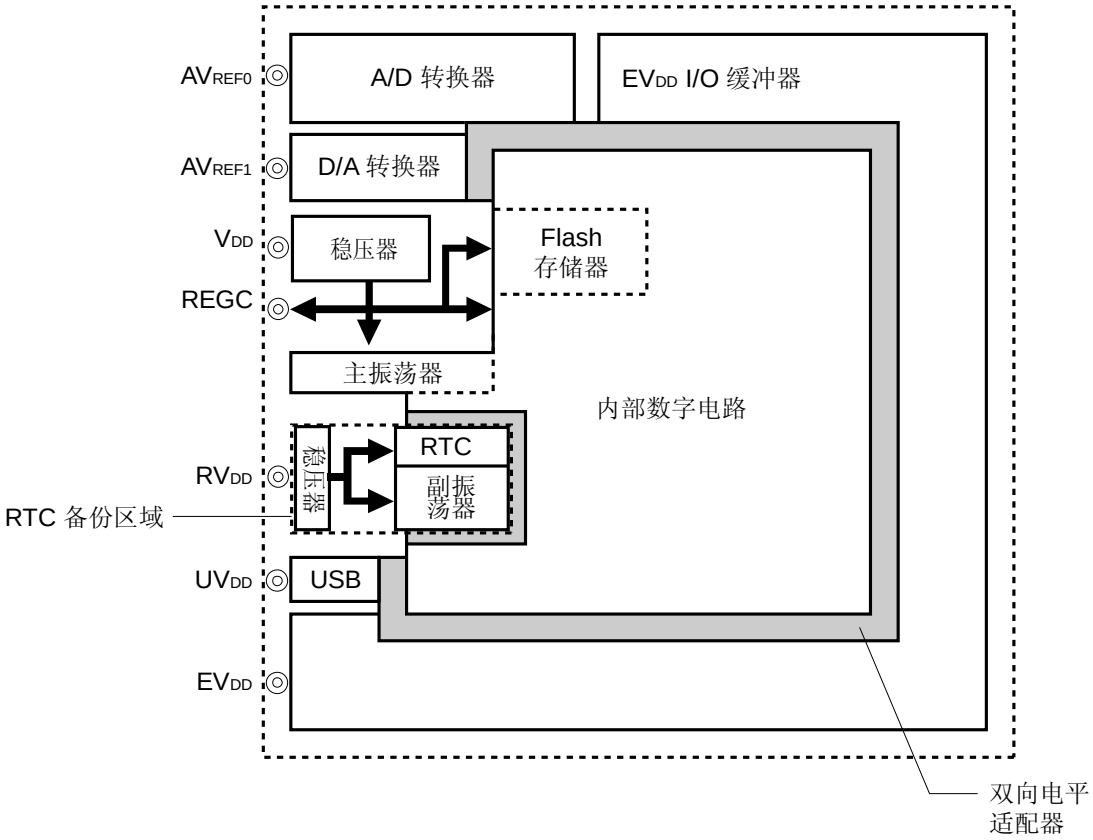
第二十九章 稳压器

29.1 概述

V850ES/JG3-L 包括一个稳压器，用于减少功率消耗和降低噪声。

该稳压器为振荡器模块和内部逻辑电路（除了 A/D 变换器，D/A 变换器和输出缓存器）提供一个阶梯下降的 V_{DD} 电源供电电压。也会向 RTC 和副振荡模块提供阶梯下降的 RV_{DD} 电源供电电压。

图 29-1. 稳压器



注意事项 1. 使用稳压器应设置 V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}。

2. 使用 RV_{DD} 的电位与那些使用其他电源供给的电位有所不同。

29.2 操作

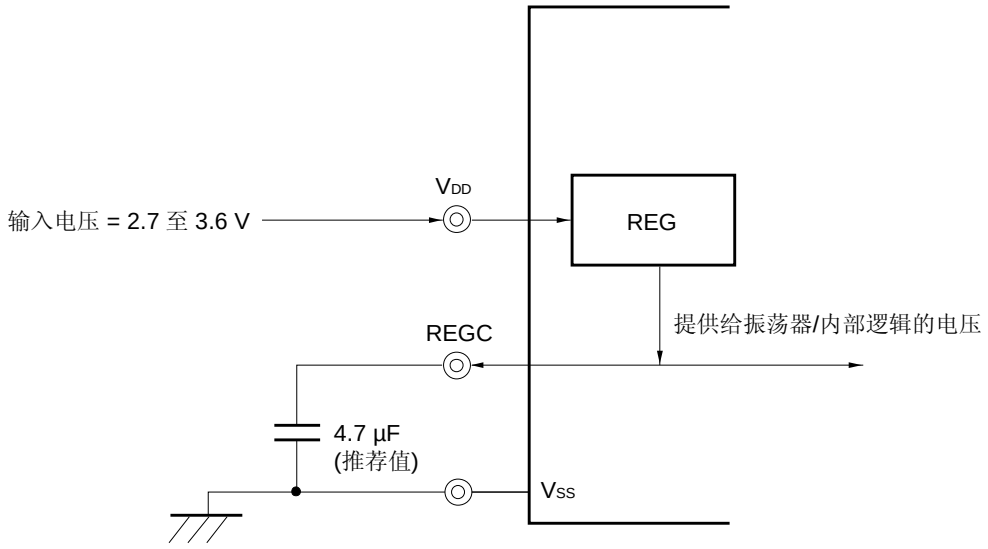
连接到 V_{DD} 的稳压器总是运行在 RTC 备份模式之外的模式（比如正常工作模式、HALT 模式、IDLE1 模式、IDLE2 模式、STOP 模式、副时钟运行模式、副 IDLE 模式或者复位期间）。

在 STOP 模式、副时钟运行模式和副 IDLE 模式下可以降低稳压器的输出电压以减少电源损耗。详情参见**第二十四章 待机功能。**

请确保在 REGC 引脚接一个电容（4.7*u*F（推荐值））来稳定稳压器的输出。

稳压器引脚的连接方式如下图所示。

图 29-2. REGC 引脚连接



第三十章 选项字节

选项字节作为 8 位数据存储在内部 flash 存储器的 000007AH 地址处（内部 ROM 区）。该 8 位数据用于指定看门狗定时器 2 的模式、指定使能或禁止停止内部振荡器、也可以设置紧随在复位结束后所必须等待的振荡稳定时间。复位结束后，通过这些设定值决定看门狗定时器 2 的模式、指定使能或禁止停止内部振荡器、并保障设置的振荡稳定时间。

把程序写入 V850ES/JG3-L 时，在程序中指定 000007AH 地址处的选项数据，参见 **30.1 程序示例**。

在程序执行时本区域的数据不能被重写。

地址： 0000007AH

7	6	5	4	3	2	1	0
WDTMD1	RMOPIN	0	0	0	RESOSTS2	RESOSTS1	RESOSTS0

WDTMD1	看门狗定时器 2 模式设置
0	操作时钟 ($f_X/f_T/f_R$) 可选 INTWDT2/WDTRES 模式可选
1	内部振荡时钟 (f_R) 固定 WDTRES 模式固定

RMOPIN	通过软件使能/停止内部振荡器的选项
0	可以通过软件停止
1	不能通过软件停止

RES OSTS2	RES OSTS1	RES OSTS0	振荡稳定时间的选择 (理论值)			
				f_X		
				2.5 MHz	6 MHz	10 MHz
0	0	0	$2^{10}/f_X$	409.6 us	禁止设置	禁止设置
0	0	1	$2^{11}/f_X$	819.2 us	禁止设置	禁止设置
0	1	0	$2^{12}/f_X$	1.638 ms	682.7 us	409.6 us
0	1	1	$2^{13}/f_X$	3.277 ms	1.365 ms	819.2 us
1	0	0	$2^{14}/f_X$	6.554 ms	2.731 ms	1.638 ms
1	0	1	$2^{15}/f_X$	13.11 ms	5.461 ms	3.277 ms
1	1	0	$2^{16}/f_X$	26.21 ms	10.92 ms	6.554 ms
1	1	1	$2^{16}/f_X$	26.21 ms	10.92 ms	6.554 ms

备注

1. 在 STOP 模式或 IDLE2 模式解除后的等待时间由 OSTS 寄存器设置。关于 OSTS 寄存器的详情，参见 **24.2 (3) 振荡稳定时间选择寄存器(OSTS)**。
2. f_X：主时钟振荡频率

注意事项

1. **因为上电的起振开销时间被考虑在内，所以实际的振荡稳定时间比理论值长。实际的振荡稳定时间是如上所示的时间，总计达 260*us*。**
2. **请务必选择 400*us* 或更长的振荡稳定时间（理论值）。如果设置的时间小于 400*us*，则内部状态变的不稳定并且操作无法保证。**
3. **请确保将第 5 位至第 3 位设置为“0”。**

30.1 程序示例

以下给出使用 CA850 时的程序示例。

[程序示例]

```
#-------------------------------------------------------------
#选项字节
#-------------------------------------------------------------
.section "OPTION_BYTES"          //指定 0000007A 地址处的选项字节//
.byte 0b00000001 -- 0x7a         //指定 0b00000001 作为选项字节//
.byte 0b00000000 -- 0x7b         //指定 0b00000000 存放在地址 0000007B//
.byte 0b00000000 -- 0x7c         //指定 0b00000000 存放在地址 0000007C//
.byte 0b00000000 -- 0x7d         //指定 0b00000000 存放在地址 0000007D//
.byte 0b00000000 -- 0x7e         //指定 0b00000000 存放在地址 0000007E//
.byte 0b00000000 -- 0x7f         //指定 0b00000000 存放在地址 0000007F//
```

注意事项 **确保在该区块中指定 6 个选项字节。如果少于 6 字节，在执行连接程序时将会发生错误。**
错误信息：F4112: "选项字节"段长度非法（illegal "OPTION_BYTES" section size**）。**

备注 将 007BH 至 007FH 地址的值设为 0x00。

第三十一章 FLASH存储器

V850ES/JG3-L 单片机内置有一个 flash 存储器。

- μPD70F3794: 256 KB flash 存储器
- μPD70F3795: 384 KB flash 存储器
- μPD70F3796: 512 KB flash 存储器

Flash 存储器版本为开发环境和批量生产应用提供以下的优势：

- V850ES/JG3-L 被焊接到目标系统之后改换软件。
- 开始批量生产时的数据调整。
- 根据不同模式的小规模生产式样提供不同的软件。
- 便于库存管理。
- 出厂后更新软件。

31.1 特性

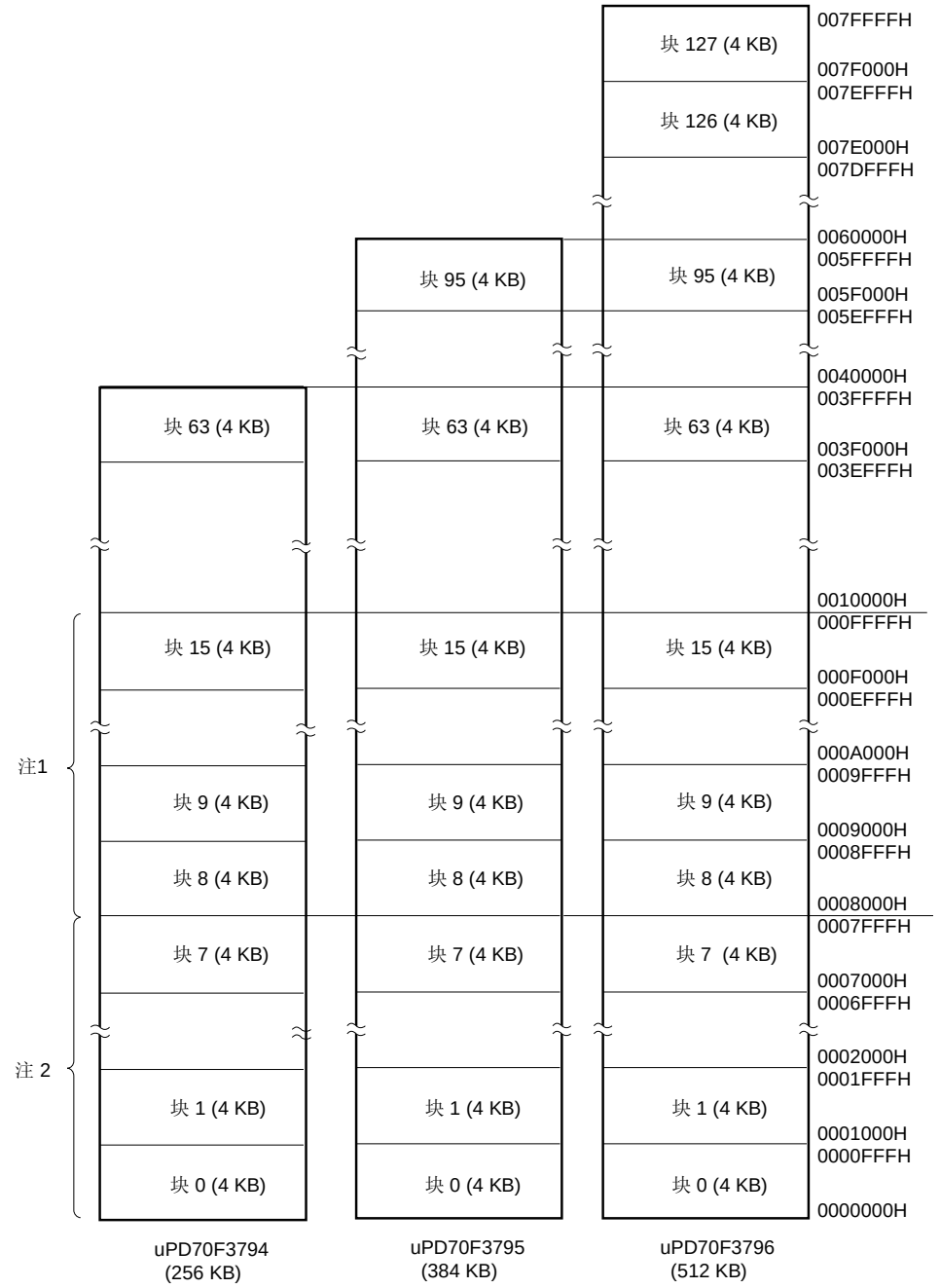
- 容量： 512/384/256 KB
- 重写方式：
  - 通过串行接口使用专用的 flash 存储器编程器进行通信重写(在线/离线编程)。
  - 通过用户程序重写 flash 存储器（自编程）。
- 支持 flash 存储器禁止写入功能(安全功能)。
- 使用引导交换功能通过自编程方式对整个 flash 存储区安全重写。
- 自编程期间可以响应中断。
- 4-字节/1-时钟访问 (当取指时)

31.2 存储器配置

V850ES/JG3-L 单片机内部 flash 存储区划分为 64 或 96 或 128 个块，且可以按块为单位进行编程/擦除。也可以一次性擦除所有的模块。

当使用引导交换功能时，块地址为 0 至 7 的物理存储区被块地址为 8 至 15 的物理存储区替换。关于引导交换功能的详情，参见 **31.5 通过自编程重写。**

图 31-1. Flash 存储器映射



注 **1.** 使用引导交换功能被引导区替换的区域
2. 引导区

31.3 功能概述

不管 V850ES/JG3-L 是否被焊接到目标系统上（离线/在线编程），V850ES/JG3-L 的内部 flash 存储器都可通过专用 flash 编程器的重写功能进行重写。

此外，还支持安全功能，可以禁止对已经写入内部 flash 存储器的用户程序进行重写，因此，程序不能被未经授权者改变。

假设目标系统生产/出厂后将会改变程序，则使用用户程序（自编程）的重写功能是理想的选择。同样还支持安全重写整个 flash 存储器的引导交换功能。此外，自编程期间还可以执行中断服务，所以 flash 存储器可以在多种条件下进行重写，比如和外部设备通信时。

表 31-1. 重写方式

重写方式	功能概述	工作模式
在板编程	设备安装到目标系统之后，flash 存储器可使用专用 flash 编程器重写。	Flash 存储器编程模式
离板编程	在设备安装到目标系统之前，flash 存储器可通过专用 flash 存储器编程器和专用程序适配板(FA 系列)重写。	
自编程	Flash 存储器可通过执行用户程序来重写，用户程序通过离线/在线编程预先写入 flash 存储器。（在自编程时，不能从内部 flash 存储器区取指令，也不能对内部 flash 存储器区进行数据存取。因此，重写程序必须预先传送到内部 RAM 或外部存储器。）	正常操作模式

备注 FA 系列为 Naito Densei Machida Mfg. Co., Ltd 的产品。

<R>

表 31-2. 基本功能

功能	功能概述	支持 (√：可支持，×：不支持)	
		在线/离线编程	自编程
空白检测	检查整个存储器的擦除状态。	√	√
片擦除	一次全部擦除整个存储器的内容。	√	×注
块擦除	指定存储块的内容被擦除。	√	√
编程	执行到指定地址的写入，并校验检查写入的等级是否安全	√	√
校验/检查和	将从 flash 存储器读出的数据和由 flash 编程器发送的数据进行比较。	√	× (可由用户程序读取)
读取	写入 flash 存储区的数据被读取。	√	×
安全设置	禁止使用块擦除命令、片擦除命令、编程命令以及读取命令，且禁止重写引导区。	√	× (仅当设置由使能变为禁止时，可以支持)

注 为块擦除功能选择了整片存储器，这时是可能的。

下表列出了安全功能。出厂之后，默认禁止片擦除命令、禁止块擦除命令、禁止编程命令、禁止读取命令以及禁止重写引导块簇功能，且可通过在线/离线编程进行重写来完成安全设置。各个安全功能可以同时与其它功能组合使用。

表 31-3. 安全功能

功能	功能概述
禁止片擦除命令	在所有块上禁止执行片擦除命令和块擦除命令。一旦设定为禁止之后，所有的关于禁止的设置都无法重新初始化设定， 因为片擦除命令不能执行。
禁止块擦除命令	在所有块上禁止执行块擦除指令。关于禁止的设置可以通过执行片擦除指令来初始化。
禁止编程命令	所有模块上禁止执行编程命令和块擦除命令。关于禁止的设置可以通过执行片擦除命令来初始化。
禁止读取命令	所有块上禁止执行读取命令。关于禁止的设置可以通过执行片擦除命令来初始化。
禁止重写引导块簇	可以保护从块 0 到指定块的引导块簇的内容。禁止对已经受到保护的引导块簇执行重写（擦除和写入）。即使执行片擦除命令，所有的关于禁止的设置也无法重新初始化设定。 可以指定的块最大编号如下： μPD70F3794：块 63 μPD70F3795：块 95 μPD70F3796：块 127

表 31-4. 安全设置

功能	设置各个安全性能后的擦除、写入、读取 操作 (√：可执行， ×：不可执行， –：不支持)		安全设置的注意事项	
	在线/离线编程	自编程	在线/离线编程	自编程
禁止片擦除命令	块擦除命令：× 片擦除命令：× 编程命令：√注1 读取命令：√	片擦除：– 块擦除 (Flash 块擦除)：√ 写入(Flash 字写入)：√ 读取(Flash 字读取)：√	关于禁止的设置无法被初始化	仅当设置由使能变为禁止时，可以支持自编程。
禁止块擦除命令	片擦除命令：√ 块擦除命令：× 编程命令：√ 读取命令：√	片擦除：– 块擦除 (Flash 块擦除)：√ 写入(Flash 字写入)：√ 读取(Flash 字读取)：√	关于禁止的设置可以通过片擦除命令进行初始化	
禁止编程命令	片擦除命令：√ 块擦除命令：× 编程命令：× 读取命令：√	片擦除：– 块擦除 (Flash 块擦除)：√ 写入(Flash 字写入)：√ 读取(Flash 字读取)：√	关于禁止的设置可以通过片擦除命令进行初始化	
禁止读取命令	片擦除命令：√ 块擦除命令：√ 编程命令：√ 读取命令：×	片擦除：– 块擦除 (Flash 块擦除)：√ 写入(Flash 字写入)：√ 读取(Flash 字读取)：√	关于禁止的设置可以通过片擦除命令进行初始化	
禁止重写引导区	片擦除命令：× 块擦除命令：×注2 编程命令：×注2 读取命令：:√	片擦除：– 块擦除 (Flash 块擦除)：×注2 写入(Flash 字写入)：×注2 读取(Flash 字读取)：√	关于禁止的设置无法被初始化	

<R> **注** **1.** 在这种情况下，因为擦除命令无效，与已写入数据不同的数据将无法被写入 flash 存储器。

2. 在引导块簇之外可以执行。

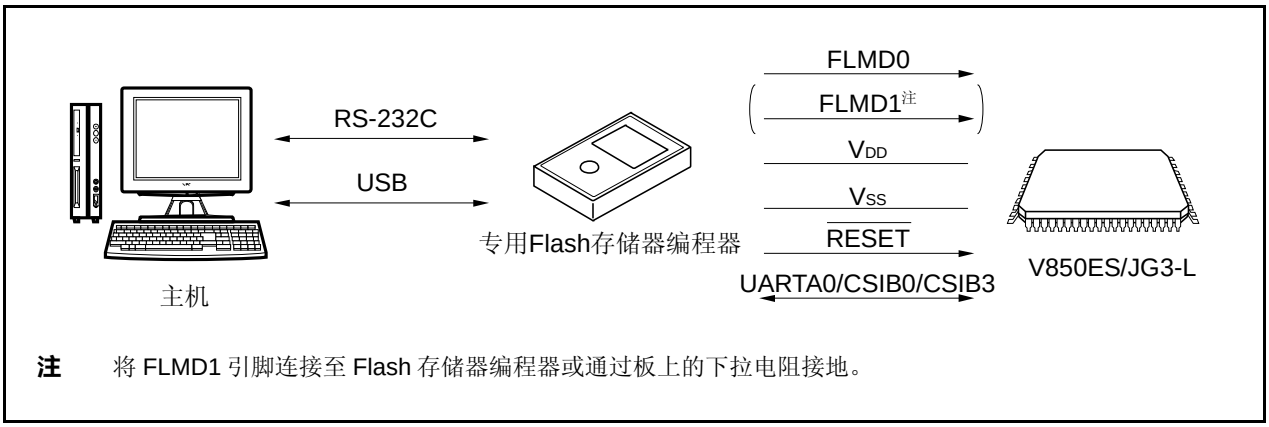
31.4 通过专用flash存储器编程器进行重写

V850ES/JG3-L 被安装到目标系统（在板 on-board 编程）之后，flash 存储器可通过专用 flash 编程器进行重写。通过 Flash 存储器和专用的编程适配器（FA 系列）的组合，在芯片安装到目标系统前进行重写（离板 off-board 编程）。

31.4.1 编程环境

下面显示了将程序写入 V850ES/JG3-L 的 flash 存储器所需的环境。

图 31-2. 程序写入 flash 存储器所需的环境



需要一台主机用来控制专用 flash 存储器编程器。但是，在某些情况下，专用 flash 存储器编程器也可以脱机使用。详情参见专用 flash 存储器编程器的用户手册。

UARTA0，CSIB0 或 CSIB3 被用作专用 flash 编程器和 V850ES/JG3-L 之间的接口，执行写入、擦除等操作。离线编程需要一个专用编程适配器 (FA 系列)。

推荐下列产品：

- FA-70F3796GC-UEU-RX (GC-UEU 型) (已连线)
- FA-100GC-UEU-B (GC-UEU 型) (未连线：需要连线)
- FA-121F1-CAH-B (F1-CAH 型) (未连线：需要连线)

备注 FA 系列为 Naito Densei Machida Mfg. Co., Ltd 的产品。

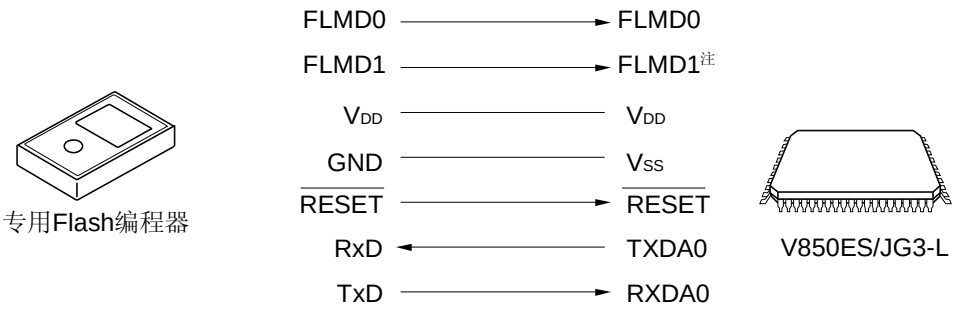
31.4.2 通信模式

专用 flash 存储器编程器和 V850ES/JG3-L 的通信方式是串行通信，使用 V850ES/JG3-L 的 UARTA0，CSIB0 或 CSIB3 接口。

(1) UARTA0

传输速率： 9,600 至 153,600 bps

图 31-3. 和专用 flash 存储器编程器通信(UARTA0)

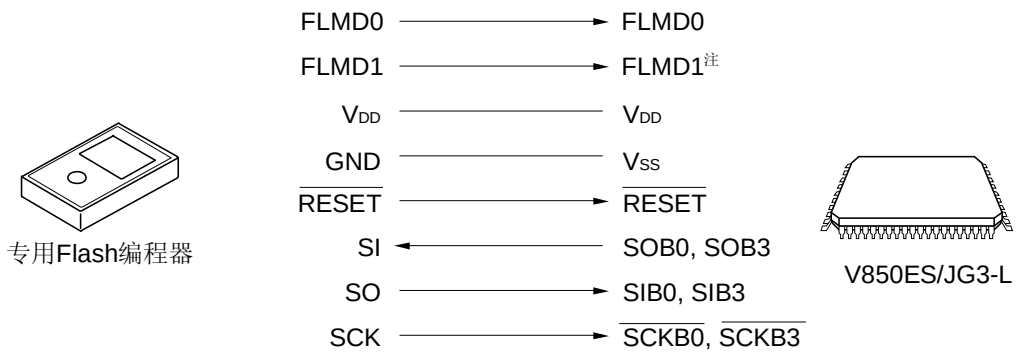


注 将 FLMD1 引脚连接至 flash 存储器编程器或通过板上的下拉电阻接地。

(2) CSIB0，CSIB3

串行时钟：2.4 kHz 至 5 MHz (MSB 先行)

图 31-4. 和专用 flash 存储器编程器通信(CSIB0，CSIB3)



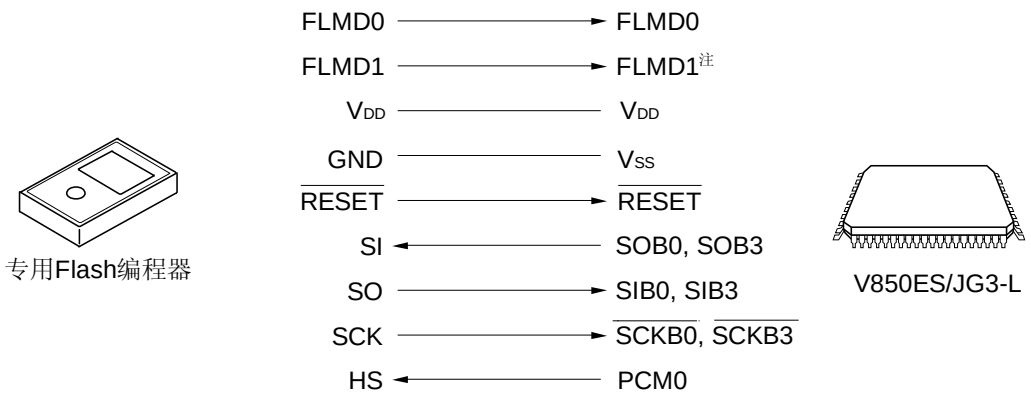
注 将 FLMD1 引脚连接至 flash 存储器编程器或通过板上的下拉电阻接地。

(3) CSIB0 + HS，CSIB3 + HS

串行时钟：2.4 kHz 至 5 MHz (MSB 先行)

V850ES/JG3-L 作为从机操作。

图 31-5. 和专用 flash 存储器编程器通信(CSIB0 + HS)



注 将 FLMD1 引脚连接至 flash 存储器编程器或通过板上的下拉电阻接地。

31.4.3 接口

专用 flash 存储器编程器输出串行时钟，V850ES/JG3-L 作为从机工作。

当专用 flash 存储器编程器使用 PG-FP5 时，会产生下面的信号到 V850ES/JG3-L。详情参见 **PG-FP5 用户手册 (U18865E)**。

表 31-5. 专用 Flash 存储器编程器的信号连接 (PG-FP5)

PG-FP5			V850ES/JG3-L	连接处理		
信号名称	I/O	引脚功能	引脚名称	UARTA0	CSIB0, CSIB3	CSIB0 + HS, CSIB3 + HS
FLMD0	输出	写入使能/禁止	FLMD0	◎	◎	◎
FLMD1	输出	写入使能/禁止	FLMD1	◎注1	◎注1	◎注1
VDD	–	V_{DD} 电压产生/电压监控	V_{DD}	◎	◎	◎
GND	–	地	V_{SS}	◎	◎	◎
CLK	输出	输出至 V850ES/JG3-L 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	发送信号	SIB0, SIB3/RXDA0	◎	◎	◎
SCK	输出	传输时钟	$\overline{\text{SCKB0}}$, $\overline{\text{SCKB3}}$	×	◎	◎
HS	输入	CSIB0 + HS, CSIB3 + HS 通信的握手信号	PCM0	×	×	◎

注 1. 离板编程时，将这些引脚按图 31-6 所示进行连接，或通过板上的下拉电阻将其接地。在板编程时，将这些引脚按图 31-11 所示进行连接。

2. **要向** V850ES/JG3-L 提供时钟，在电路板上安装一个振荡器，或将 PG-FP5 的 CLK 信号连接到 V850ES/JG3-L 的 X1 信号。

备注 ◎：必须进行连接。

×： 不必进行连接。

表 31-6. V850ES/JG3-L flash 写入适配器 (FA-100GC-UEU-B)的连线（1/2）

Flash 存储器编程器(FG-FP5)连接引脚			FA 板的引脚名称	使用 CSIB0 + HS		使用 CSIB0		使用 UARTA0	
信号名称	I/O	引脚功能		引脚名称	引脚编号. GC	引脚名称	引脚编号 GC	引脚名称	引脚编号. GC
SI/RxD	输入	接收信号	SI	P41/SOB0/SCL01	23	P41/SOB0/SCL01	23	P30/TXDA0/SOB4	25
SO/TxD	输出	发送信号	SO	P40/SIB0/SDA01	22	P40/SIB0/SDA01	22	P31/RXDA0/INTP7/SIB4	26
SCK	输出	传输时钟	SCK	P42/$\overline{\text{SCKB0}}$	24	P42/$\overline{\text{SCKB0}}$	24	不需要	–
CLK	输出	至 V850ES/JG3-L 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	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14
FLMD0	输出	写入电压	FLMD0	FLMD0	8	FLMD0	8	FLMD0	8
FLMD1	输出	写入电压	FLMD1	PDL5/AD5/FLMD1	76	PDL5/AD5/FLMD1	76	PDL5/AD5/FLMD1	76
HS	输入	CSI0 + HS 通信的握手信号	RESERVE/HS	PCM0/$\overline{\text{WAIT}}$	61	不需要	–	不需要	–
VDD	–	VDD 电压产生/电压监控	VDD	V_{DD}	9	V_{DD}	9	V_{DD}	9
				EV_{DD}	34, 70	EV_{DD}	34, 70	EV_{DD}	34, 70
				RV_{DD}	17	RV_{DD}	17	RV_{DD}	17
				UV_{DD}	30	UV_{DD}	30	UV_{DD}	30
				AV_{REF0}	1	AV_{REF0}	1	AV_{REF0}	1
				AV_{REF1}	5	AV_{REF1}	5	AV_{REF1}	5
GND	–	地	GND	V_{SS}	11	V_{SS}	11	V_{SS}	11
				AV_{SS}	2	AV_{SS}	2	AV_{SS}	2
				EV_{SS}	33, 69	EV_{SS}	33, 69	EV_{SS}	33, 69

注意事项 **请确保将 REGC 引脚通过 4.7*u*F（推荐值）电容器接地。**

备注 GC：100-针塑封 LQFP (密间距) (14 × 14)

表 31-6. V850ES/JG3-L flash 写入适配器 (FA-100GC-UEU-B)的连线（2/2）

Flash 存储器编程器(FG-FP5)连接引脚			FA 板的引脚名称	使用 CSIB3 + HS		使用 CSIB3	
信号名称	I/O	引脚功能		引脚名称	引脚编号. GC	引脚名称	引脚编号 GC
SI/RxD	输入	接收信号	SI	P911/A11/SOB3	54	P911/A11/SOB3	54
SO/TxD	输出	发送信号	SO	P910/A10/SIB3	53	P910/A10/SIB3	53
SCK	输出	传输时钟	SCK	P912/A12/$\overline{\text{SCKB3}}$	55	P912/A12/$\overline{\text{SCKB3}}$	55
CLK	输出	至 V850ES/JG3-L 的时钟	X1	不需要	–	不需要	–
			X2	不需要	–	不需要	–
/RESET	输出	复位信号	/RESET	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14
FLMD0	输出	写入电压	FLMD0	FLMD0	8	FLMD0	8
FLMD1	输出	写入电压	FLMD1	PDL5/AD5/FLMD1	76	PDL5/AD5/FLMD1	76
HS	输入	CSI0 + HS 通信的握手信号	RESERVE/HS	PCM0/$\overline{\text{WAIT}}$	61	不需要	–
VDD	–	VDD 电压产生/电压监控	VDD	V_{DD}	9	V_{DD}	9
				EV_{DD}	34, 70	EV_{DD}	34, 70
				RV_{DD}	17	RV_{DD}	17
				UV_{DD}	30	UV_{DD}	30
				AV_{REF0}	1	AV_{REF0}	1
				AV_{REF1}	5	AV_{REF1}	5
GND	–	地	GND	V_{SS}	11	V_{SS}	11
				AV_{SS}	2	AV_{SS}	2
				EV_{SS}	33, 69	EV_{SS}	33, 69

注意事项 **请确保将 REGC 引脚通过 4.7*u*F（推荐值）电容器接地。**

备注 GC：100-针塑封 LQFP (密间距) (14 × 14)

图 31-6. V850ES/JG3-L Flash 写入适配器(FA-100GC-UEU-B)的连线示例 (CSIB0 + HS 模式) (1/2)

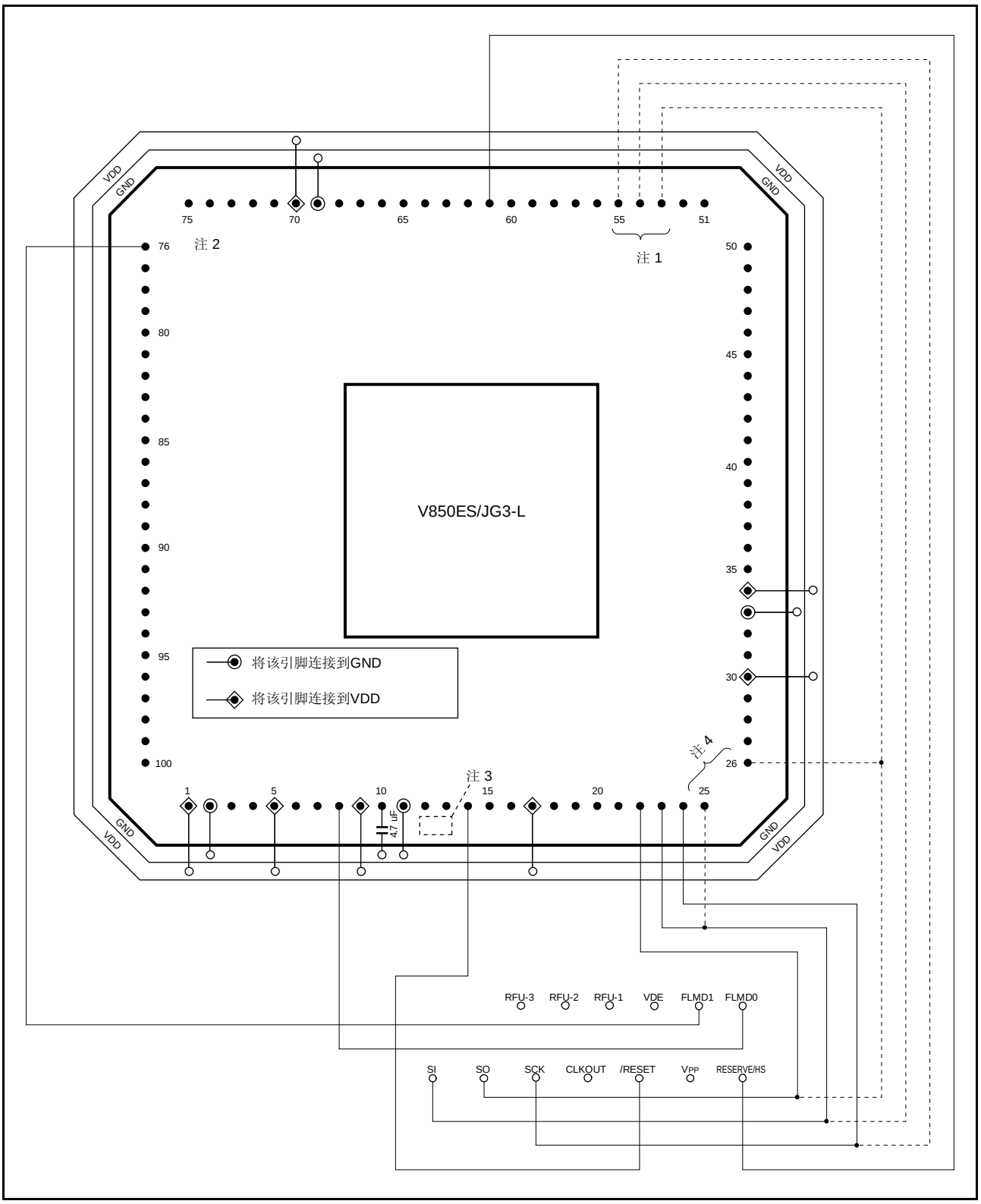
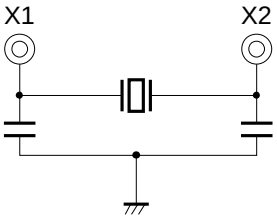


图 31-6. V850ES/JG3-L Flash 写入适配器(FA-100GC-UEU-B)的连线示例 (CSIB0 + HS 模式) (2/2)

注 **1**. 当使用 CSIB3 时对应的引脚。
2. FLMD1 引脚的接线如下所示（推荐），或通过板上的下拉电阻将其接地。
3. 在 flash 写入适配器上创建一个振荡器 (如虚线所示)并提供时钟。
这里给出一个振荡器的例子。

示例：



4. 对应使用 UARTA0 时的引脚。

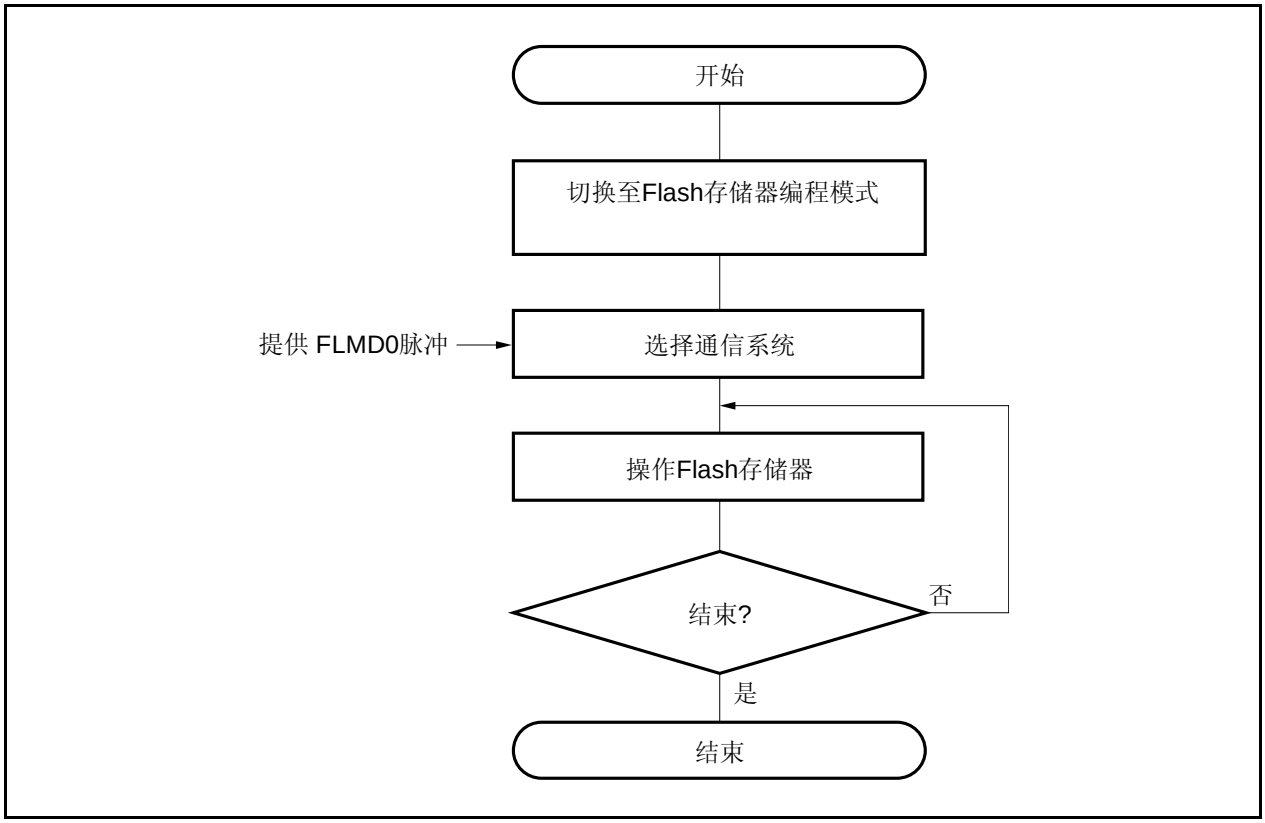
注意事项 不要输入高电平到 $\overline{\text{DRST}}$ 引脚。

备注 **1.** 对于在 flash 存储器编程中未使用的引脚，保持在复位结束后的瞬时状态。对于未列出的引脚的处理，按照未使用引脚的处理方法。(参见 **2.3 引脚 I/O 电路类型、I/O 缓冲器电源以及未使用引脚的连接)。**
2. 该适配器适用于 100-针塑封 LQFP。

31.4.4 Flash存储器控制

以下显示了操作 flash 存储器的步骤。

图 31-7. 操作 flash 存储器的步骤

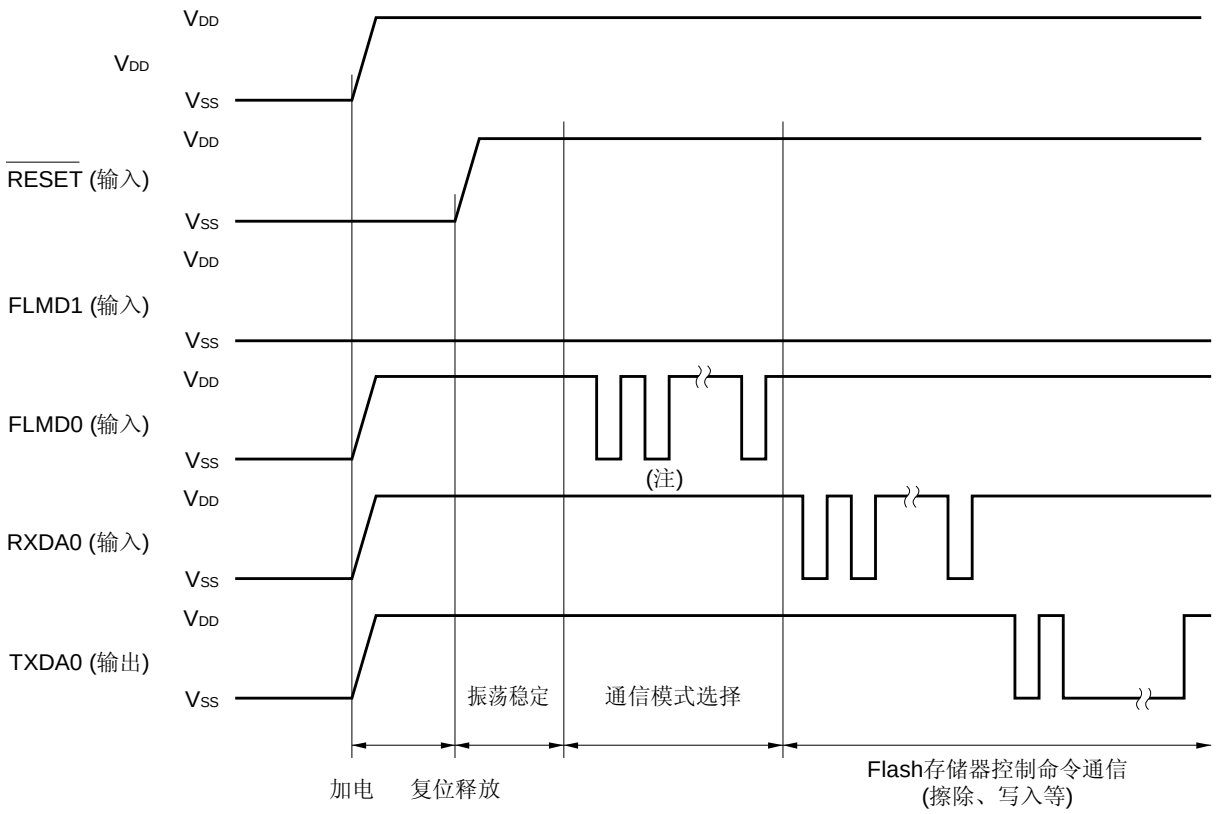


31.4.5 通信模式的选择

在 V850ES/JG3-L 单片机中，切换至 flash 存储器编程模式后，通过输入脉冲（最多 12 个脉冲）至 FLMD0 引脚来选择通信模式。FLMD0 脉冲由专用 flash 存储器编程器产生。

下面显示了脉冲数目和通信模式之间的关系。

图 31-8. 通信模式的选择



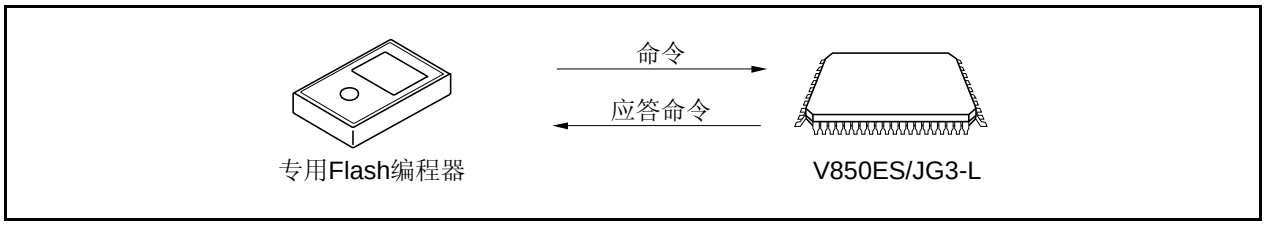
注 根据不同通信模式，时钟数目如下：

FLMD0 脉冲	通信模式	备注
0	UARTA0	通信速度：9,600 bps (复位后)，LSB 先行
8	CSIB0	V850ES/JG3-L 执行从机操作， MSB 先行
9	CSIB3	V850ES/JG3-L 执行从机操作， MSB 先行
11	CSIB0 + HS	V850ES/JG3-L 执行从机操作， MSB 先行
12	CSIB3 + HS	V850ES/JG3-L 执行从机操作， MSB 先行
其它	RFU	禁止设置

31.4.6 通信命令

V850ES/JG3-L 通过指令的方式与专用 flash 存储器编程器进行通信。从专用 flash 存储器编程器发至 V850ES/JG3-L 的信号称为“命令”。从 V850ES/JG3-L 发出至专用 flash 存储器编程器的应答信号称为“应答命令”。

图 31-9. 通信命令



下面显示了 V850ES/JG3-L 用来控制 flash 存储器的命令。所有的这些命令都由专用 flash 存储器编程器产生，而 V850ES/JG3-L 执行与命令相应的处理。

表 31-7. Flash 存储器控制命令

类别	命令名称	支持			功能
		CSIB0, CSIB3	CSIB0 + HS, CSIB3 + HS	UARTA0	
空白检查	块空白检查命令	√	√	√	检查指定块的存储内容是否已被正确擦除。
擦除	片擦除命令	√	√	√	擦除整个存储器的内容。
	块擦除命令	√	√	√	擦除指定块的存储内容。
写入	编程命令	√	√	√	写入指定的地址范围，且执行内容检验检查。
校验	校验命令	√	√	√	比较存储器中指定地址范围的内容和 flash 编程器传送的数据
	校验和命令	√	√	√	读取指定地址范围内的检查和。
读取	读取命令	√	√	√	读取写入到 flash 存储器中的数据
系统设置控制	硅签字命令	√	√	√	读取硅签字信息。
	安全设置命令	√	√	√	禁止片擦除命令、块擦除命令、编程命令、读取命令和引导区重写。

31.4.7 在板编程时的引脚连接

当执行在线写入时，将目标系统上安装一个连接器用以连接专用 flash 存储器编程器。另外，电路板上还要具有将正常操作模式切换至 flash 存储器编程模式的功能。

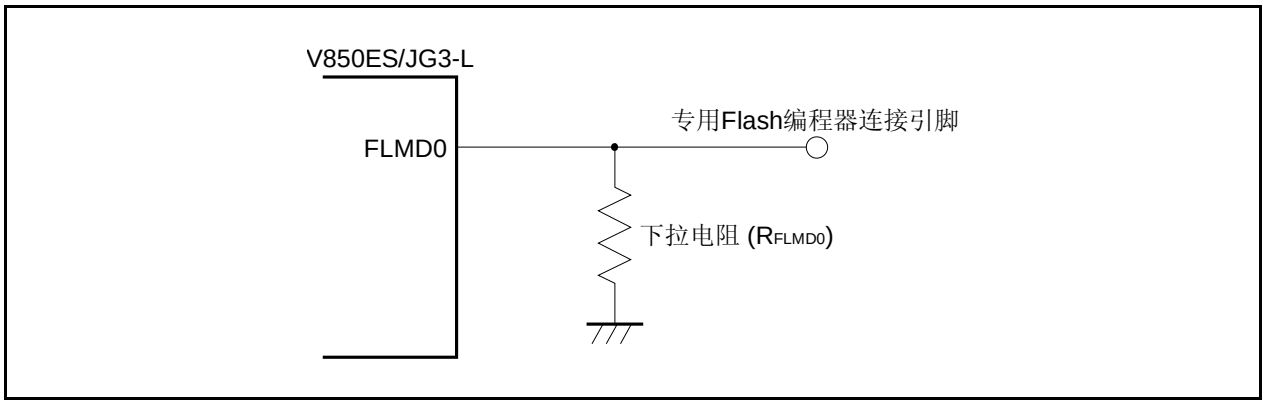
在 flash 存储器编程模式下，所有在 flash 存储器编程过程中未使用的引脚，其状态与复位后的瞬时状态相同。因此，当外部设备不能识别复位后的瞬时状态时，需要对引脚进行处理。

(1) FLMD0 引脚

在正常操作模式下，将电压值为 V_{SS} 的电平输入至 FLMD0 引脚。在 flash 存储器编程模式，为 FLMD0 引脚提供电平为 V_{DD} 的写入电压。

因为在自编程模式时，FLMD0 引脚被用作写入保护引脚，因此在写入 flash 存储器之前，必须通过端口控制等手段给 FLMD0 引脚提供一个电平为 V_{DD} 的电压。详情参见 **31.5.5 (1) FLMD0 引脚。**

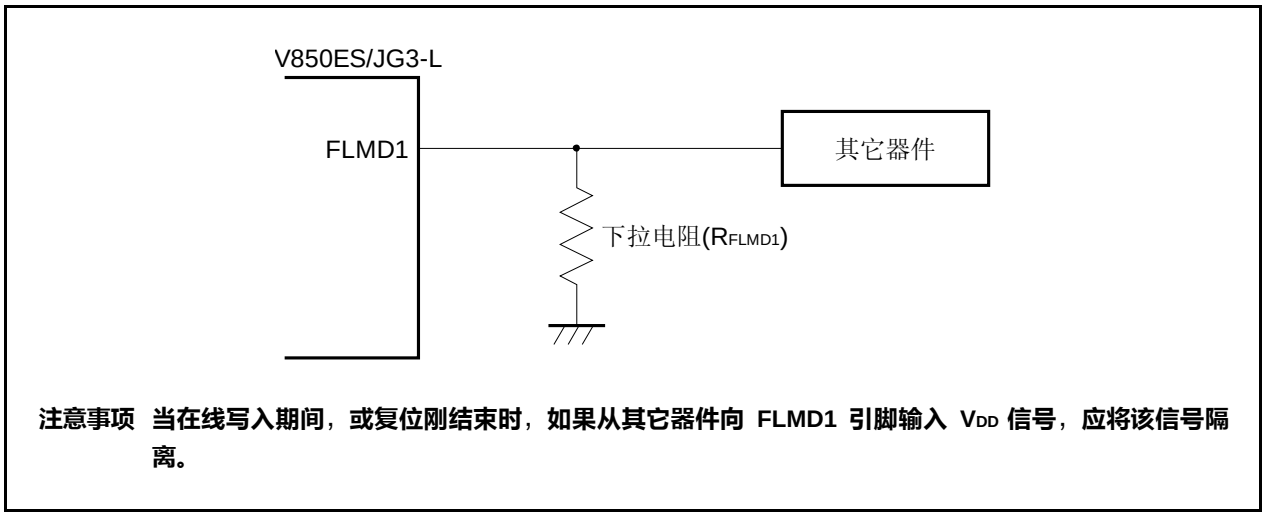
图 31-10. FLMD0 引脚连接示例



(2) FLMD1 引脚

当 FLMD0 引脚输入 0V 时，FLMD1 引脚不起作用。当给 FLMD0 引脚提供 V_{DD} 电平时，进入 flash 存储器编程模式，因此必须给 FLMD1 引脚输入 0V。下面是 FLMD1 引脚的一个连接示例。

图 31-11. FLMD1 引脚连接示例



注意事项 当在线写入期间，或复位刚结束时，如果从其它器件向 FLMD1 引脚输入 V_{DD} 信号，应将该信号隔离。

表 31-8. 复位刚结束后 FLMD0 和 FLMD1 引脚与操作模式之间的关系

FLMD0	FLMD1	操作模式
0	忽略	正常操作模式
V_{DD}	0	Flash 存储器编程模式
V_{DD}	V_{DD}	禁止设置

(3) 串行接口引脚

下面给出各个串行接口使用的引脚。

表 31-9. 串行接口所用引脚

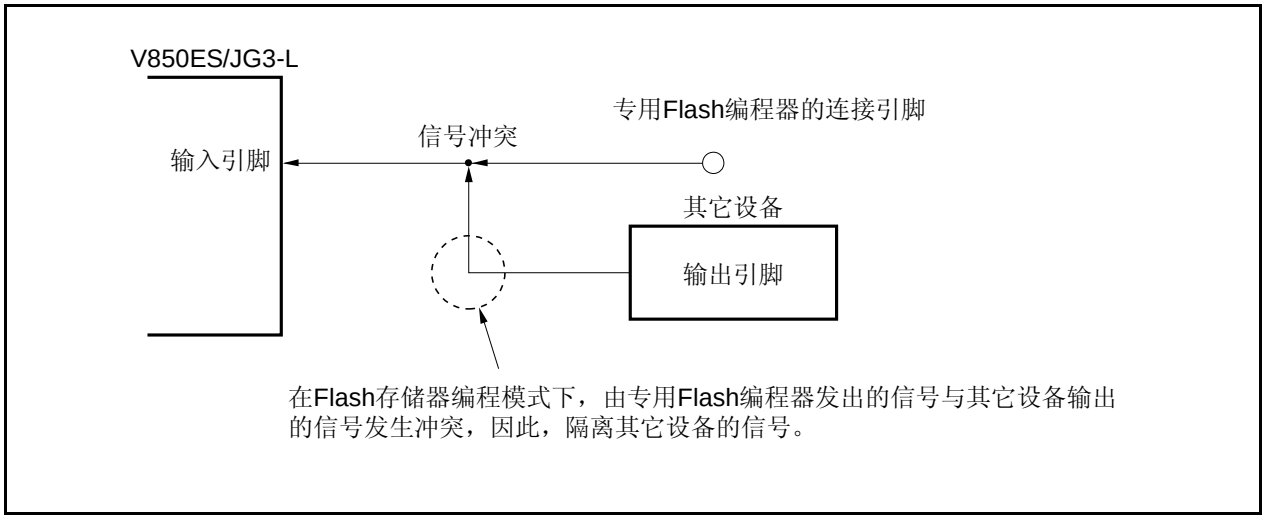
串行接口	所用引脚
UARTA0	TXDA0, RXDA0
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB3	SOB3, SIB3, $\overline{\text{SCKB3}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, PCM0
CSIB3 + HS	SOB3, SIB3, $\overline{\text{SCKB3}}$, PCM0

当将专用 flash 存储器编程器接到某个串行接口引脚，该接口已经与板上其它设备连接时，应注意避免信号冲突或其它设备的误操作。

(a) 信号冲突

当专用 flash 存储器编程器（输出）连接到某个串行接口引脚（输入），该接口已经与板上其它设备（输出）连接时，会发生信号冲突。为避免信号冲突，隔离其与其它设备的连接或将其它设备设为输出高阻状态。

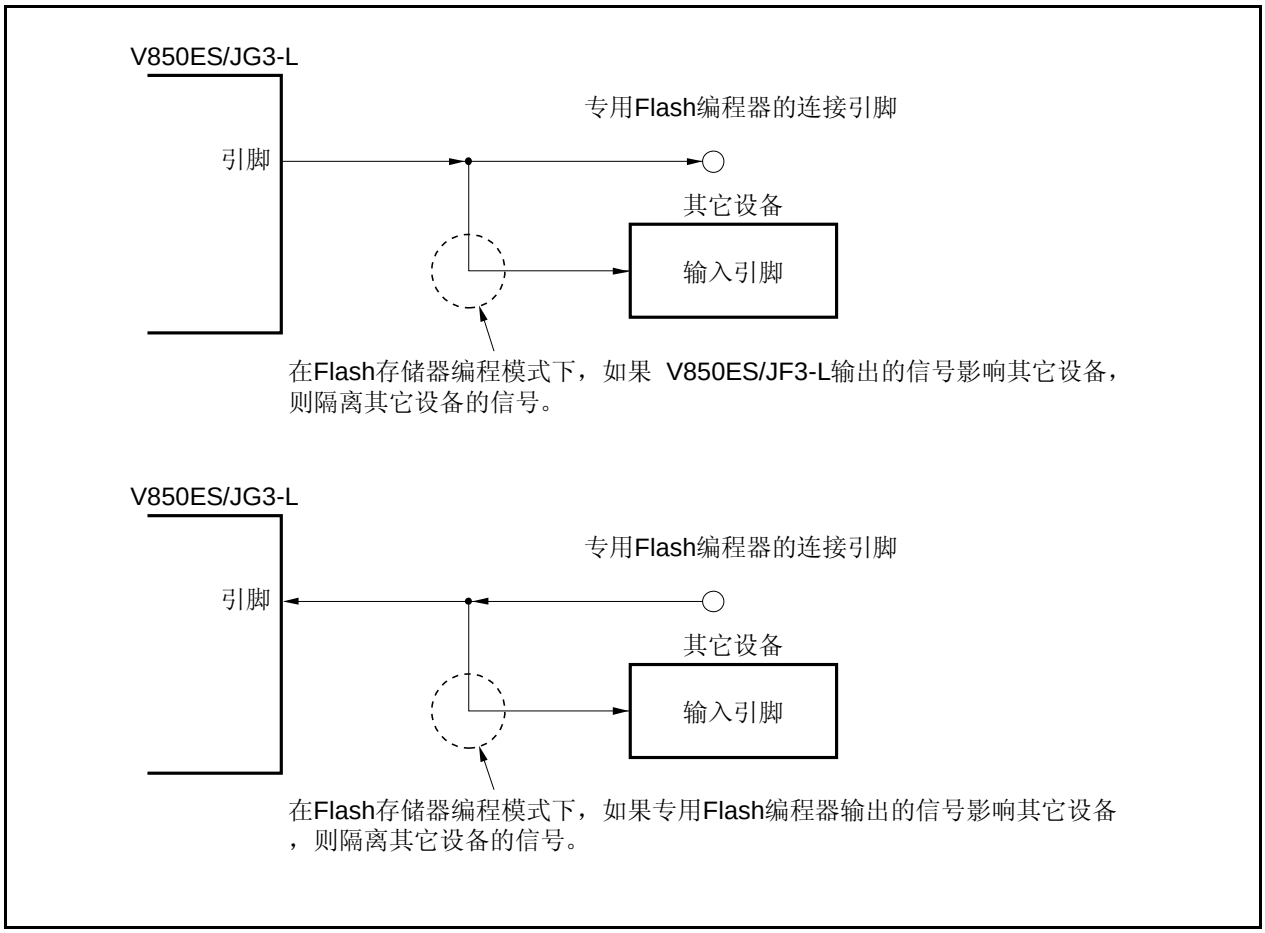
图 31-12. 信号冲突 (串行接口输入引脚)



(b) 其它设备的误操作

当专用 flash 存储器编程器（输出或输入）接到到某个串行接口引脚（输入或输出），该接口已经与板上其它设备（输入）连接时，信号输出到其它设备，导致设备的误操作。为避免这种情况。隔离与其它设备的连接。

图 31-13. 其它设备的误操作

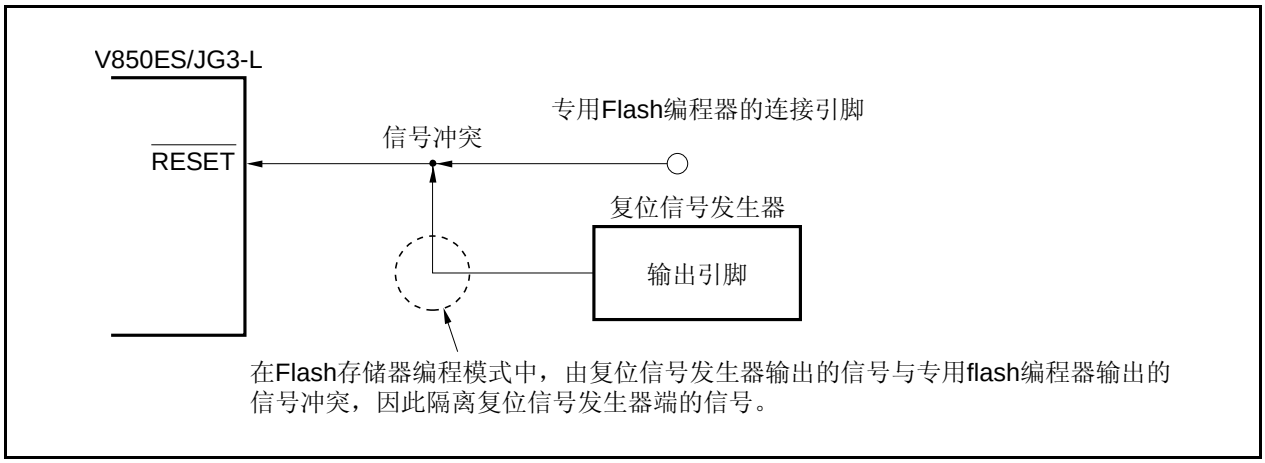


(4) $\overline{\text{RESET}}$ 引脚

当专用 flash 编程器的复位信号接到 $\overline{\text{RESET}}$ 引脚，该引脚已经和板上的复位信号发生器相连时，会发生信号冲突。为避免信号间的冲突，隔离其它复位信号发生器的连接。

在 Flash 存储器编程模式下，当复位信号由用户系统输入时，编程操作将不能正确执行。所以，不要输入任何专用 flash 存储器编程器复位信号之外的信号。

图 31-14. 信号冲突 ($\overline{\text{RESET}}$ 引脚)



(5) 端口引脚 (包括 NMI)

当系统切换至 flash 存储器编程模式时，所有在 flash 存储器编程模式中不使用的引脚处于与复位后瞬时状态相同的状态。如果连接至各个端口的外部器件无法识别端口复位后的瞬时状态，引脚就需要适当的处理，如通过电阻接至 V_{DD} 或 V_{SS}。

(6) 其它信号引脚

X1、X2、XT1、XT2 和 REGC 的连接状态，与正常操作模式相同。

flash 存储器编程期间，向 $\overline{\text{DRST}}$ 引脚输入低电平或开路断开。不要输入高电平。

(7) 供电

提供正常操作模式时的电源(V_{DD}、V_{SS}、EV_{DD}、EV_{SS}、AV_{REF0}、AV_{REF1}、AV_{SS}、RV_{DD}、UV_{DD})。

备注 引脚连接的详情，参见 **PG-FP5 用户手册 (U18865E)**。

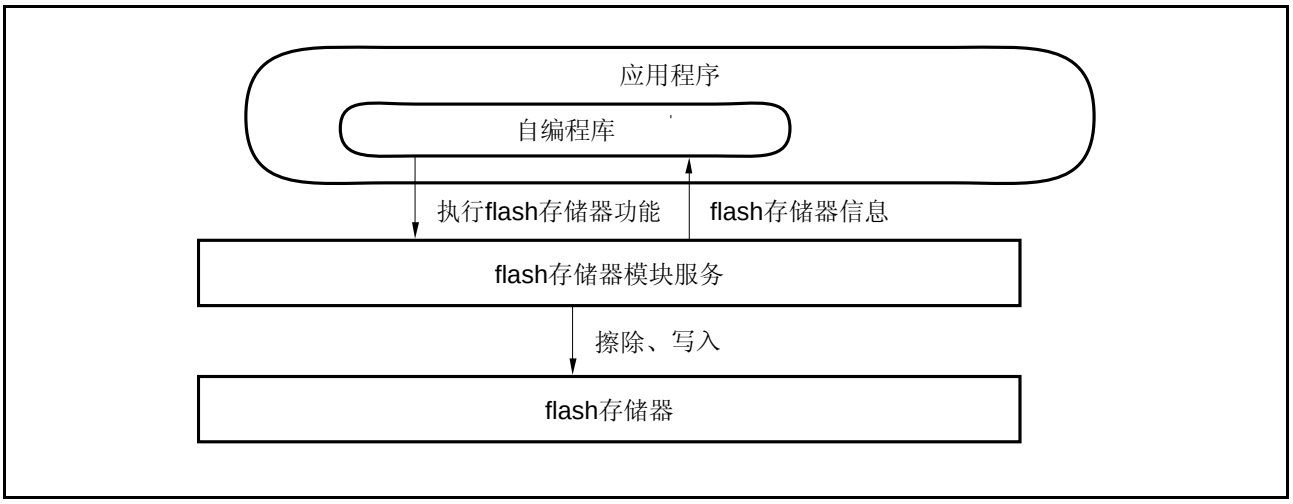
31.5 通过自编程进行重写

31.5.1 概述

V850ES/JG3-L 支持 flash 宏服务，该服务允许用户程序自行重写内部 flash 存储器。通过用户应用程序对该接口和一个自编程库的使用来重写 flash 存储器，flash 存储器可被预先传送到内部 RAM 或外部存储器的用户程序重写。从而可以更新用户程序并现场改写常量数据[注]。

注 请确保不要将程序代码和重写目标的常量数据分配在同一个块中。块配置参见 **31.2 存储器配置。**

图 31-15. 自编程的概念

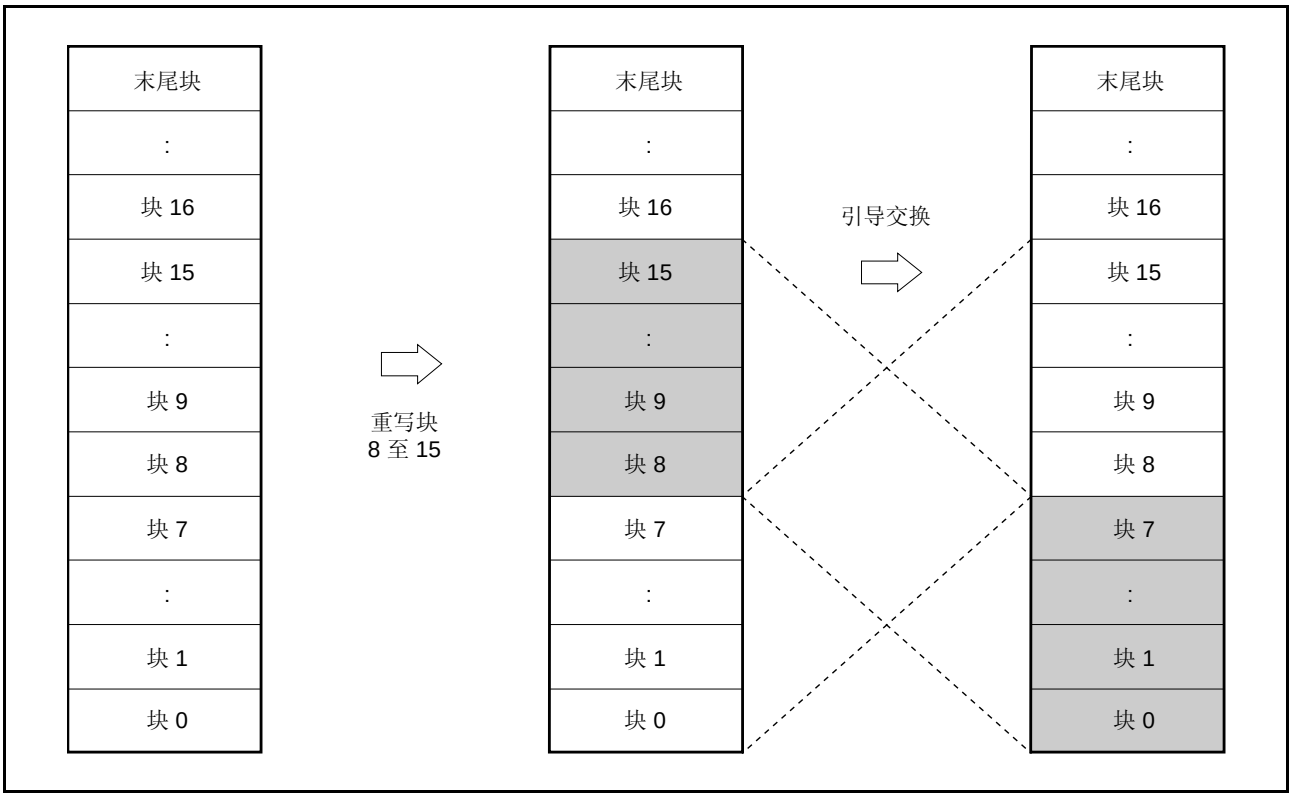


31.5.2 特性

(1) 安全自编程(引导交换功能)

V850ES/JG3-L 支持引导交换功能，可将物理存储器的块 0 至 7 与块 8 至 15 互换。预先写入起始程序，该程序将要被写入块 8 至块 15，然后交换物理存储器，即使在重写期间发生断电，整个区域也可以安全地重新写入，因为正确用户程序始终存在于块 0 至块 7 中。

图 31-16. 重写整个存储区 (引导交换)



(2) 中断支持

自编程期间，不能从 flash 存储器提取指令。因此即使发生一个中断，写入到 flash 存储器的用户处理程序也不能使用。

因此，为了在 V850ES/JG3-L 自编程期间使用中断，处理转移到内部 RAM 的指定地址[注]。写入一条跳转指令，将程序处理转移到内部 RAM 中指定地址[注]上的用户中断服务程序。

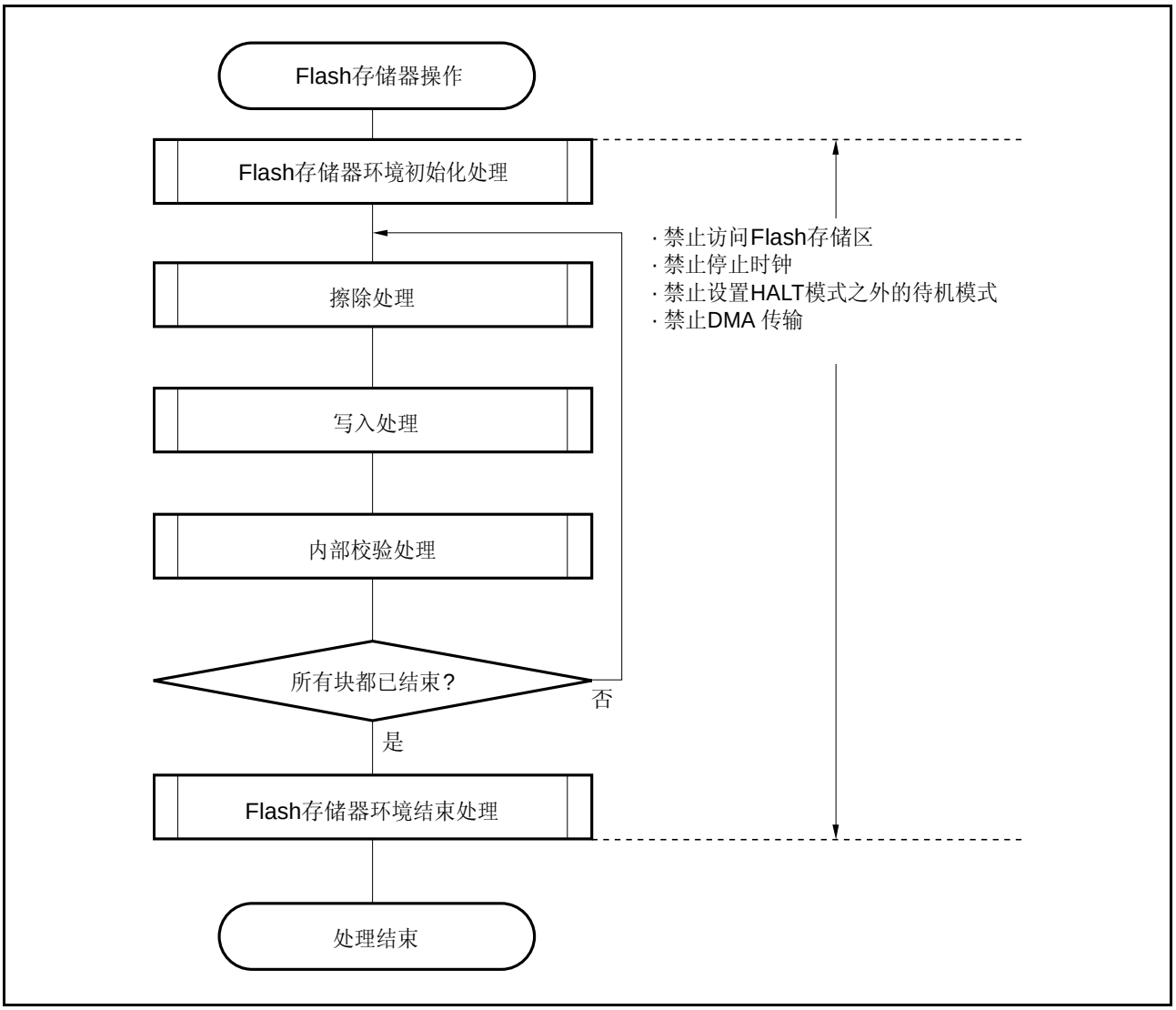
注 NMI 中断：内部 RAM 的起始地址。
可屏蔽中断：内部 RAM 起始地址 + 4 的地址。

注意事项 在自编程期间要执行 INTKR 中断服务，将中断屏蔽标志设置为禁止中断 (KRIC.KRMK 位 = 1)并轮询中断请求标志 (KRIC.KRIF 标志)。
在自编程期间禁止将 KRIC.KRMK 位设置为 0 (使能中断)。
中断服务的详情，参见 22.3.4 中断控制寄存器 (xxICn)。

31.5.3 标准自编程流程

通过 flash 自编程对 flash 存储器重写的整个过程图示如下。详情参见 **Flash 存储器自编程库用户手册(U17819E)**。

图 31-17. 标准自编程流程图



31.5.4 Flash功能

表 31-10. Flash 功能列表

功能名称	概述	支持
FlashInit	自编程库初始化	√
FlashEnv	Flash 环境启动/停止	√
FlashFLMDCheck	FLMD 引脚检查	√
FlashStatusCheck	硬件处理执行状态检查	√
FlashBlockErase	块擦除	√
FlashWordWrite	数据写入	√
FlashBlockIVerify	块的内部校验	√
FlashBlockBlankCheck	块空白检查	√
FlashSetInfo	Flash 存储器信息设置	√
FlashGetInfo	Flash 存储器信息获取	√
FlashBootSwap	执行引导交换	√

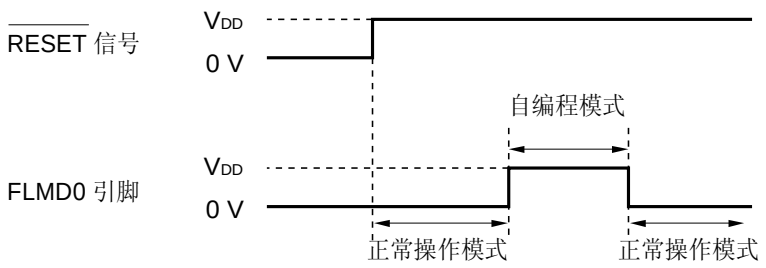
31.5.5 引脚处理

(1) FLMD0 引脚

FLMD0 引脚用于在复位刚结束后设置操作模式，并在自重写期间保护 flash 存储器不被写入。因此有必要在复位刚结束后和执行正常操作时保持 FLMD0 引脚的电压为 0V。在重新写入存储器之前，通过端口控制自编程模式期间有电平为 V_{DD} 的电压输入 FLMD0 引脚。

当完成自编程时，FLMD0 引脚电压值必须返回 0V。

图 31-18. 模式变换时序



注意事项 复位刚结束后，确保 FLMD0 引脚为 0V。

31.5.6 用到的内部资源

下表列出了自编程需要使用的内部资源。这些内部资源也可以自由地用于除了自编程之外的用途。

表 31-11. 用到的内部资源

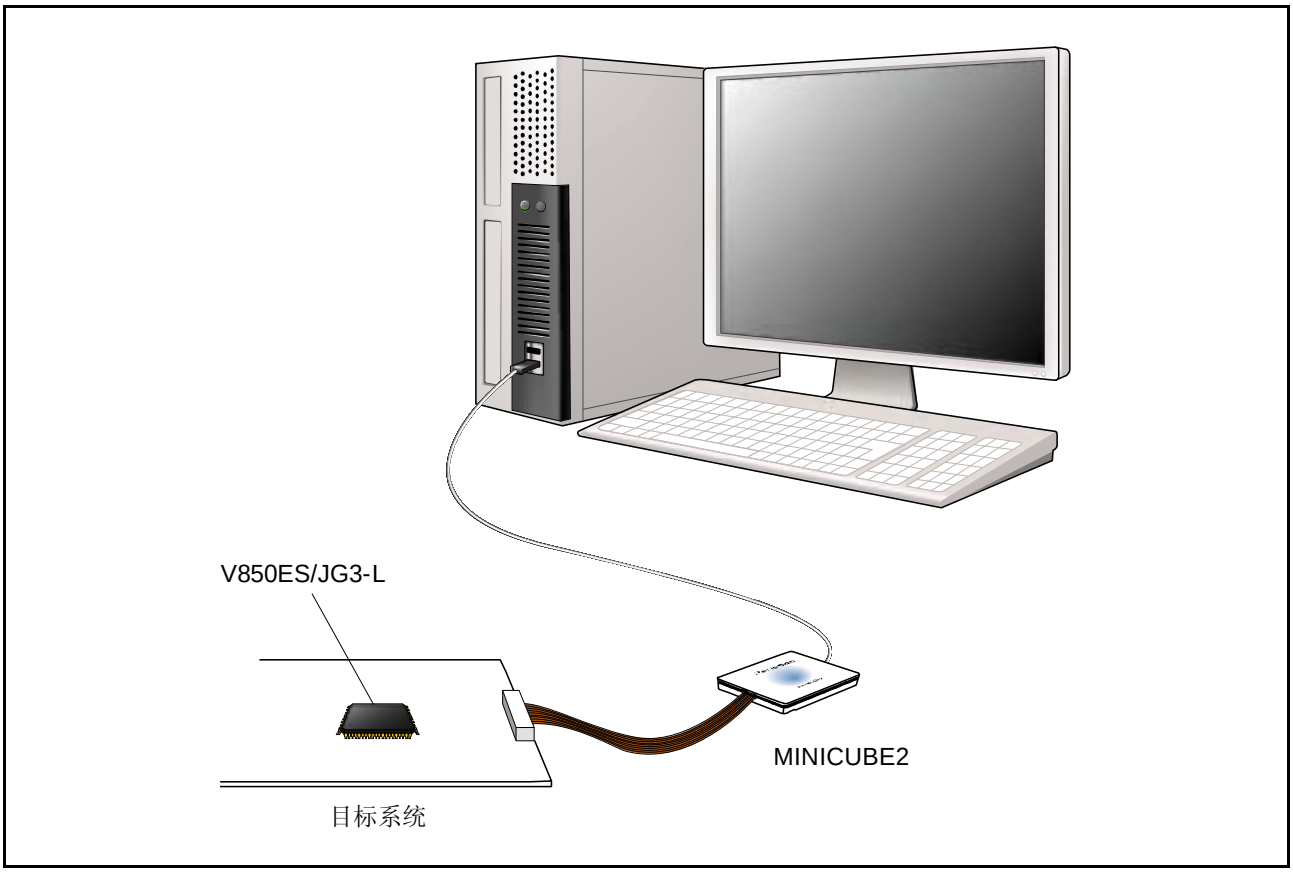
资源名称	说明
堆栈区	用户使用堆栈的扩展，被库使用 (内部 RAM 和外部 RAM 中都可以使用)。
库代码[注]	库的程序实体 (可以用于被操作的 flash 存储器模块之外的任何空间)。
应用程序	作为用户应用程序执行。 调用 flash 存储器功能。
可屏蔽中断	可在用户应用程序的执行状态或自编程状态下使用。因为处理过程转移到内部 RAM 起始地址＋4 的地址，为了在自编程状态中使用该中断，预先安排一条分支跳转指令，将程序处理转移到内部 RAM 起始地址+4 的地址上的用户中断服务程序。
NMI	可在用户应用程序执行状态或自编程状态下使用。因为处理过程被转移到内部 RAM 起始地址，为了在自编程状态时使用该中断，预先安排一条分支跳转指令，将程序处理转移到内部 RAM 起始地址上的用户中断服务程序。

注 关于用到的资源详情，参见 **Flash 存储器自编程库用户手册。**

第三十二章 片上调试功能

片上调试是在已经安装在目标系统上的 V850ES/JG3-L 执行调试。
使用 MINICUBE2，可以用简单接口来执行片上调试。

图 32-1. 使用 MINICUBE2 进行片上调试



V850ES/JG3-L 单片机的片上调试功能可由下面两种方法实现。

- 使用 DCU (调试控制单元)
  V850ES/JG3-L 中，使用 $\overline{\text{DRST}}$，DCK，DMS, DDI 以及 DDO 引脚作为调试接口引脚，由片上 DCU 来执行片上调试功能。
- 不使用 DCU
  通过 MINICUBE2 来执行片上调试功能，使用用户资源来替代 DCU。

下表显示了两种片上调试方法的特性。

表 32-1. 片上调试概述

		使用 DCU 调试	不使用 DCU 调试
调试接口引脚		$\overline{\text{DRST}}$、DCK、DMS、DDI、DDO	• 当使用 UARTA0 时 RXD0，TXD0 • 使用 CSIB0 时 SIB0、SOB0、$\overline{\text{SCKB0}}$、HS (PCM0) • 使用 CSIB3 时 SIB3、SOB3、$\overline{\text{SCKB3}}$、HS (PCM0)
分配用户资源		不需要	必需
硬件中断功能		2 点	2 点
软件中断功能	4 点	4 点	4 点
	2000 点	2000 点	2000 点
实时 RAM 监控功能[注 1]		可用	可用
动态内存修改(DMM)功能[注 2]		可用	可用
屏蔽功能		复位、NMI、INTWDT2、$\overline{\text{HLDRQ}}$、$\overline{\text{WAIT}}$	$\overline{\text{RESET}}$ 引脚
ROM 安全功能		10-字节 ID 码鉴定	10-字节 ID 码鉴定
使用硬件		MINICUBE 等.	NINICUBE2 等
跟踪功能		不支持	不支持
调试中断接口功能(DBINT)		不支持	不支持

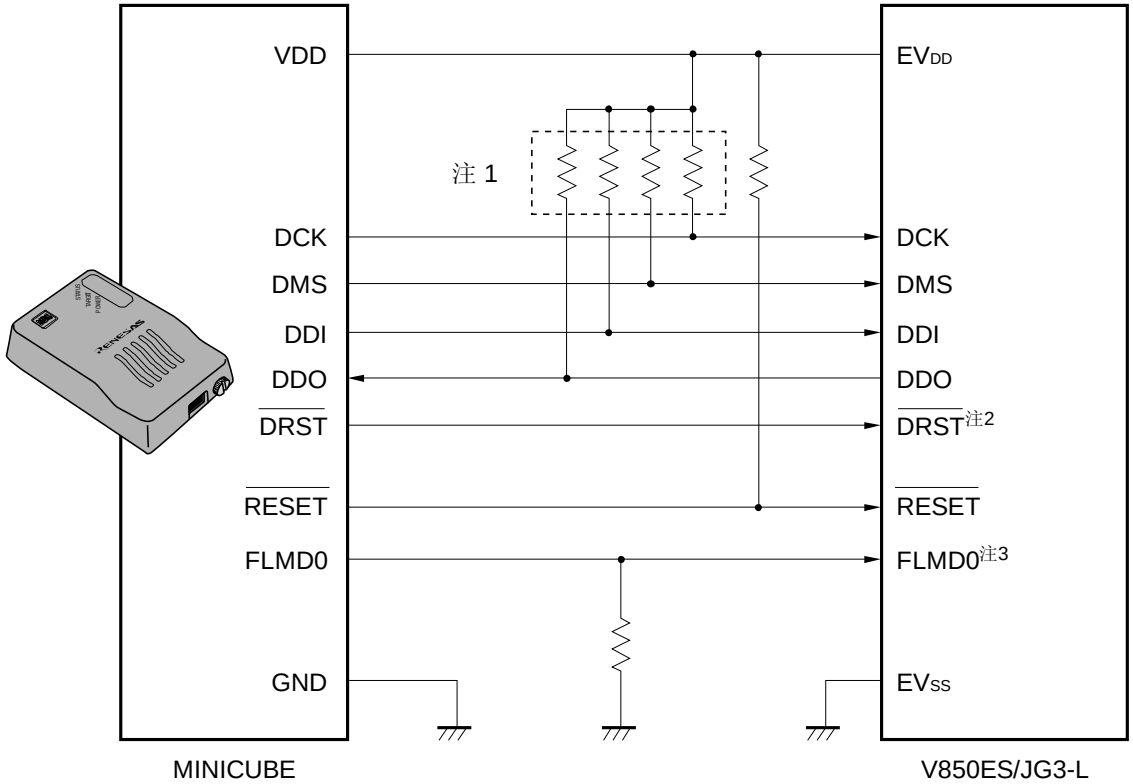
注 1. 这是在程序执行时读出存储器内容的功能。
2. 这是在程序执行时重写 RAM 内容的功能。

32.1 用DCU调试

通过调试接口引脚（$\overline{\text{DRST}}$, DCK, DMS, DDI 和 DDO）连接片上调试仿真器(MINICUBE)，仅通过这些引脚就可以调试程序，而无需使用其它的用户资源。

32.1.1 连接电路示例

图 32-2. 调试接口用作通信接口时的电路连接示例



注 1. **未连接 MINICUBE 时的引脚连接示例。**
2. **片上提供一个下拉电阻。**
3. **用于 flash 存储器重写。**

32.1.2 接口信号

接口信号的描述如下：

(1) $\overline{\text{DRST}}$

它是片上调试单元的复位输入信号。它是负逻辑信号，异步初始化调试单元。
集成调试器启动后，当 MINICUBE 检测到目标系统的 V_{DD} 时使得 $\overline{\text{DRST}}$ 上升，并且启动设备上的片上调试单元。
当 $\overline{\text{DRST}}$ 信号上升为高后，CPU 产生一个复位信号。
在启动集成调试器开始调试后，总是会产生一个 CPU 复位。

(2) DCK

它是时钟输入信号。由 MINICUBE 为其提供 20 MHz 或 10 MHz 的时钟。在片上调试单元，在 DCK 信号的上升沿对 DMS 和 DDI 信号进行采样，而在其下降沿输出数据 DDO。

(3) DMS

它是传输模式选择信号。调试单元的传输状态随 DMS 信号的电平改变。

(4) DDI

这是数据输入信号。在片上调试单元内在 DCK 的上升沿对其进行采样。

(5) DDO

这是数据输出信号。在 DCK 信号的下降沿由片上调试单元输出该信号。

(6) EV_{DD}

该信号用于检测目标系统的 VDD。
如果没有检测到目标系统的 VDD，则由 MINICUBE 输出该信号（DRST, $\overline{\text{DCK}}$, DMS, DDI, FLMD0 和 RESET），并进入高阻状态。

(7) FLMD0

flash 存储器的自编程功能用于通过集成调试器(ID850QB)把数据下载到 flash 存储器中。在 flash 存储器自编程期间，FLMD0 引脚必须保持为高。此外，在 FLMD0 引脚接一个下拉电阻。

FLMD0 引脚可通过下面两种方式进行控制：

<1> 由 MINICUBE 控制：

将 MINICUBE 的 FLMD0 信号连接到 FLMD0 引脚。

在正常模式下，MINICUBE 不作任何驱动（高阻）。

在中断期间，当集成调试器执行下载功能时 MINICUBE 将 FLMD0 引脚上升为高电平。在其它情况下，FLMD0 引脚处于高阻抗状态。

<2> 由端口控制：

当执行自编程时，使用该方法。

将设备的任一端口连接到 FLMD0 引脚。

可能会用到由用户程序实现 flash 存储器自编程功能的同一个端口。

在集成调试器的控制端，在执行下载之前，设置连接 FLMD0 引脚的端口引脚为高电平，在下载完成时，将该端口引脚重置为低电平。

详情参见 **ID850QB Ver.3.40 集成调试器操作用户手册(U18604E)。**

(8) $\overline{\text{RESET}}$

它是系统复位输入引脚。如果通过用户程序设置 OCDM.OCDM0 位的值，使得 $\overline{\text{DRST}}$ 引脚无效，那么将无法执行片上调试。因此，通过 $\overline{\text{RESET}}$ 引脚，由 MINICUBE 执行复位，可以使得 $\overline{\text{DRST}}$ 引脚生效（初始化）。

32.1.3 可屏蔽功能

Reset、NMI、INTWDT2、$\overline{\text{WAIT}}$ 和 $\overline{\text{HLDRQ}}$ 信号都可以被屏蔽。

集成调试器（ID850QB）内的可屏蔽信号以及相应的 V850ES/JG3-L 功能列举如下：

表 32-2. 可屏蔽功能

调试器内的可屏蔽信号	相应的 V850ES/JG3-L 功能
NMI0	NMI 引脚输入
NMI2	产生不可屏蔽中断请求信号 (INTWDT2)
STOP	不可屏蔽
HOLD	$\overline{\text{HLDRQ}}$ 引脚输入
RESET	由 $\overline{\text{RESET}}$ 引脚输入、 低电压检测器、时钟监控或看门狗定时器 (WDT2) 溢出，产生复位信号
WAIT	$\overline{\text{WAIT}}$ 引脚输入

32.1.4 寄存器

(1) 片上调试模式寄存器(OCDM)

OCDM 寄存器用于确定片上调试功能的引脚是作为片上调试引脚，还是作为普通的端口/外设引脚。它也可以用来断开 P05/INTP2/$\overline{\text{DRST}}$ 引脚的内部下拉电阻。

该寄存器是一个专用寄存器，只能通过一个特定序列的组合来写入(参见 **3.4.7 专用寄存器**)。

只有在 P05/INTP2/$\overline{\text{DRST}}$ 引脚输入低电平时，才能够写入 OCDM 寄存器。

该寄存器可按字节或按位进行读写。

复位后： 01H[注]　　R/W　　地址：FFFFF9FCH

	7	6	5	4	3	2	1	<0>
OCDM	0	0	0	0	0	0	0	OCDM0

OCDM0	操作模式
0	选择正常操作模式（可以被复用为片上调试功能引脚的引脚用作端口/外设功能引脚）并且断开P05/INTP2/$\overline{\text{DRST}}$引脚的片上下拉电阻。
1	当P05/INTP2/$\overline{\text{DRST}}$ 引脚为低电平时： 正常操作模式(可以被复用功能为片上调试功能引脚的引脚用作端口/外设功能引脚) 当 P05/INTP2/$\overline{\text{DRST}}$引脚为高电平时： 片上调试模式 (可以被复用功能为片上调试功能引脚的引脚用作片上调试模式引脚)

注　**$\overline{\text{RESET}}$ 输入将该寄存器设为 01H。由 WDT2RES 信号、时钟监控(CLM)或低电压检测器(LVI)引发复位后，OCDM 寄存器的值仍然保持。**

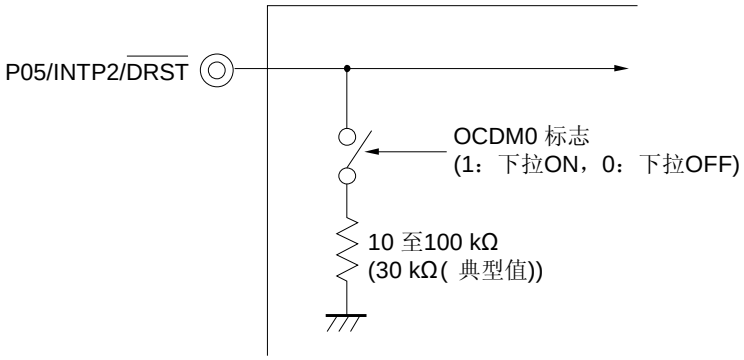
注意事项 1. 外部复位后，当 DDI、DDO、DCK 以及 DMS 引脚不作为片上调试引脚而作为端口引脚时，必须执行下列操作之一。

- 给 P05/INTP2/$\overline{\text{DRST}}$ 引脚输入一个低电平。
- 将 OCDM0 位置位。这种情况下，进行下列操作。

<1> 将 OCDM0 位清 0。

<2> 将 P05/INTP2/$\overline{\text{DRST}}$ 引脚固定为低电平，直到<1>完成。

2. P05/INTP2/$\overline{\text{DRST}}$ 引脚有一个下拉电阻。当 OCDM0 标志被清 0 时，断开该电阻。



32.1.5 操作

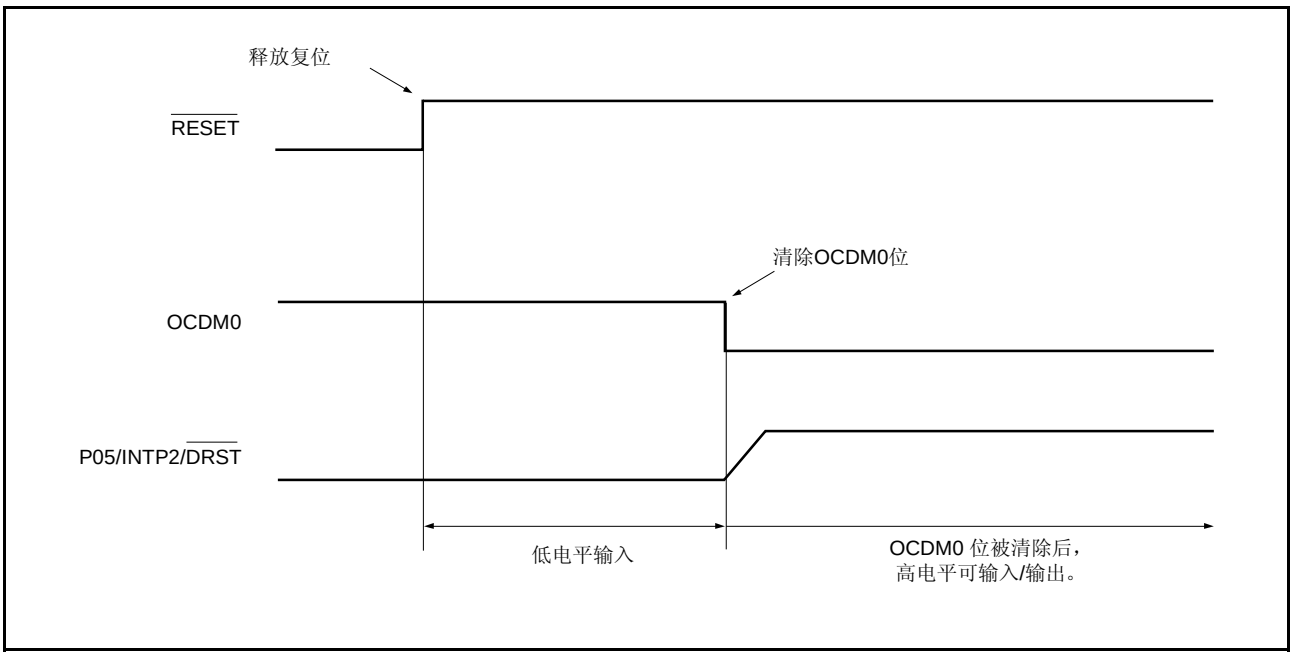
在下表条件时，片上调试功能无效。

当不使用该功能时，在 OCDM.OCDM0 标志清 0 之前，$\overline{\text{DRST}}$ 引脚一直保持为低电平。

OCDM0 标志 / $\overline{\text{DRST}}$ 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 32-2. 不使用片上调试功能时的时序



32.1.6 注意事项

(1) 如果在程序执行期间有复位信号输入（来自目标系统或来自内部复位源），则中断功能可能会误操作。

(2) 即使复位信号被屏蔽功能所屏蔽，如果在引脚输入一个复位信号也可能使得 I/O 缓存器(端口引脚)复位。

(3) 来自引脚的复位信号被指定屏蔽，且 CPU 和外设 I/O 不复位。如果在执行用户程序时，flash 存储器由 DMM 重写或由 RAM 监控功能读取之后立刻产生基于引脚的复位或内部复位，那么 CPU 和外设 I/O 可能无法正确复位。

(4) 在片上调试模式下，DDO 引脚被强制设为高电平输出。

(5) 供电电压(V_{DD})在 2.7 到 3.6V 范围之间时，片上调试可用。低于 2.7V 时将无法使用。

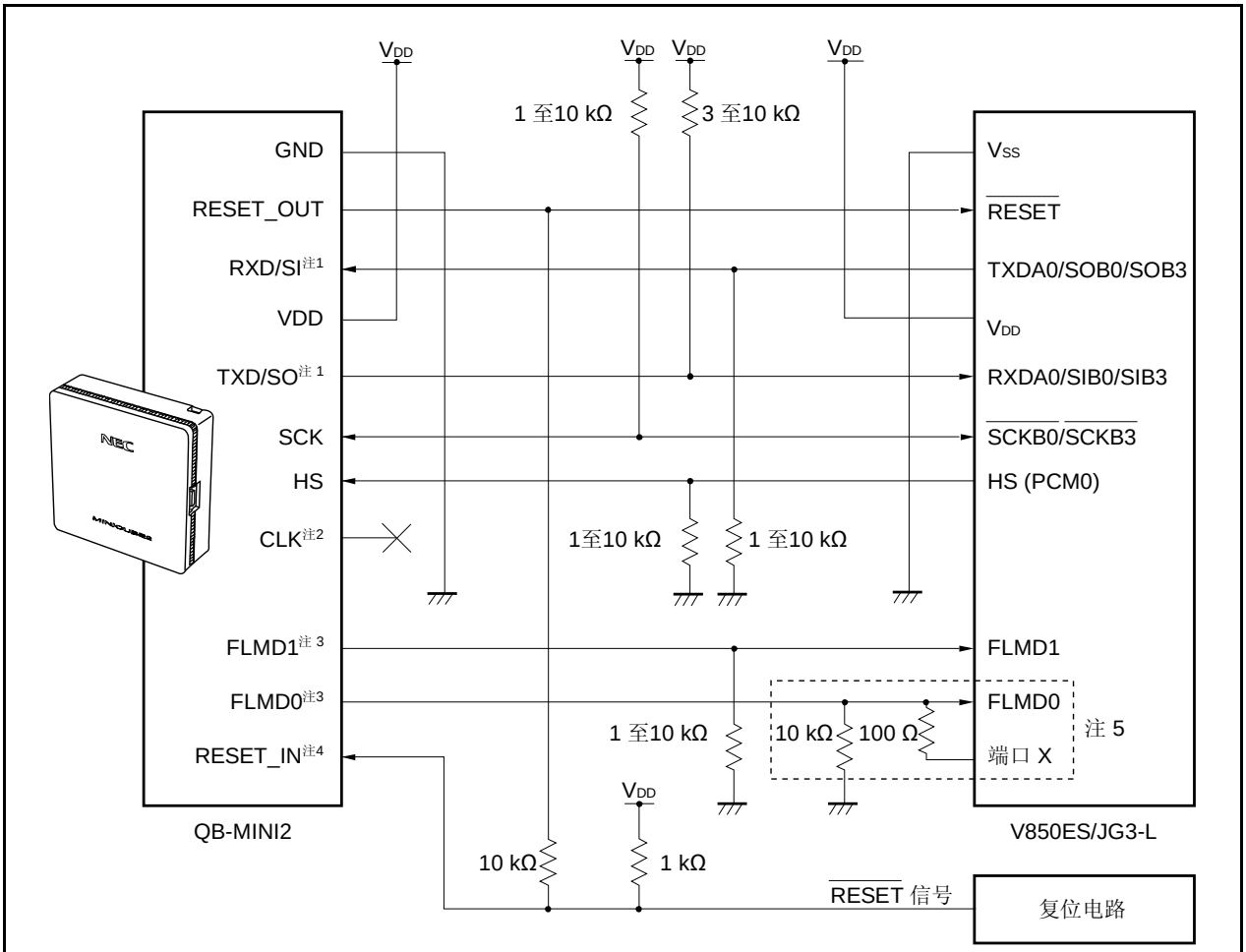
(6) 在片上调试模式下，稳压器的输出电压不会降低，即使在低电压 STOP 模式、低电压副时钟模式或低电压副 IDLE 模式下也是如此。

32.2 不使用DCU调试

下面描述如何不用 DCU 而使用 MINICUBE2 来实现片上调试功能。使用 MINICUBE2 的 UARTA0(RXDA0 和 TXDA0)引脚，CSIB0(SIB0、SOB0、SCKB0 以及 HS(PMC0))引脚或 CSIB3(SIB3、SOB3、SCKB3 以及 HS(PMC0))引脚作为调试接口

32.2.1 电路连接示例

图 32-4. UARTA0/CSIB0/CSIB3 用作通信接口时的电路连接示例



注 1. **连接 V850ES/JG3-L 的 TXDA0/SOB0/SOB3(发送端)和目标连接器的 RXD/SI (接收端)，还要将目标连接器的 TXD/SO(发送端)连接到 V850ES/JG3-L 的 RXDA0/SIB0/SIB3(接收端)。**

2. **在片上调试期间，不使用该引脚。**

3. **调试期间，该引脚用作输入（未使用）引脚且可以用作它的复用功能，应在 MINICUBE2 的该引脚连接一个 100kΩ的下拉电阻。**

4. **该连接设计是假设 RESET 信号由 N 通道漏极开路缓存器输出 (输出电阻：100Ω或更小)。**

5. **虚线内的电路是为 flash 存储器自编程设计的，通过端口控制 FLMD0 引脚。使用该端口输入或输出高电平。当不执行 flash 存储器自编程时，接到 FLMD0 引脚的下拉电阻可以在 1 至 10 kΩ之间**

备注 **当 UARTA0，CSIB0 或 CSIB3 用作通信接口时涉及的引脚参见**表 32-3**。**

表 32-3. V850ES/JG3-L 和 MINICUBE2 之间的接线

MINICUBE2 (QB-MINI2)的引脚配置			使用 CSIB0-HS			使用 CSIB3-HS			使用 UARTA0		
信号名称	I/O	引脚功能	引脚名称	引脚编号		引脚名称	引脚编号		引脚名称	引脚编号	
				GC	F1		GC	F1		GC	F1
SI/RxD	输入	该引脚用于接收自 V850ES/JG3-L 的指令和数据	P41/SOB0	23	K2	P911/SOB3	54	H9	P30/TXDA0	25	L3
SO/TxD	输出	该引脚用于发送指令和数据至 V850ES/JG3-L	P40/SIB0	22	K1	P910/SIB3	53	H10	P31/RXDA0	26	K3
SCK	输出	该引脚用于 3-线串行通信的时钟输出	P42/$\overline{\text{SCKB0}}$	24	L2	P912/$\overline{\text{SCKB3}}$	55	G11	不需要	–	–
CLK	输出	到 V850ES/JG3-L 的时钟输出引脚	不需要	–	–	不需要	–	–	不需要	–	–
RESET_OUT	输出	到 V850ES/JG3-L 的复位输出引脚	$\overline{\text{RESET}}$	14	H3	$\overline{\text{RESET}}$	14	H3	$\overline{\text{RESET}}$	14	H3
FLMD0	输出	该输出引脚用于将 V850ES/JG3-L 设置为调试模式或编程模式	FLMD0	8	F3	FLMD0	8	F3	FLMD0	8	F3
FLMD1	输出	该输出引脚用于设置编程模式	PDL5/FLMD1	76	A10	PDL5/FLMD1	76	A10	PDL5/FLMD1	76	A10
HS	输入	CSI0 + HS 通信的握手信号	PCM0/$\overline{\text{WAIT}}$	61	E9	PCM0/$\overline{\text{WAIT}}$	61	E9	不需要	–	–
GND	–	接地	V_{SS}	11	注 1	V_{SS}	11	注 1	V_{SS}	11	注 1
			AV_{SS}	2	C1, C2	AV_{SS}	2	C1, C2	AV_{SS}	2	C1, C2
			EV_{SS}	33, 69	注 2	EV_{SS}	33, 69	注 2	EV_{SS}	33, 69	注 2
RESET_IN	输入	目标系统的 reset 引脚输入									

注 **1.** G1, G2, J1
2. A6, E5-E7, E11, F5-F7, G5-G7, L1, L5

备注 GC：100 针塑封 LQFP (密间距) (14 × 14)
F1：121 针塑封 FBGA (8 × 8)

32.2.2 可屏蔽功能

只有复位信号可以被屏蔽。

调试器(ID850QB)的可屏蔽信号以及相应的 V850ES/JG3-L 的功能列举如下：

表 32-4. 可屏蔽功能

ID850QB 内的可屏蔽信号	相应的 V850ES/JG3-L 功能
NMI0	–
NMI1	–
NMI2	–
STOP	–
HOLD	–
RESET	由 $\overline{\text{RESET}}$ 引脚输入产生复位信号
WAIT	–

32.2.3 用户资源的保留

用户必须准备下列资源来实现 MINICUBE2 和目标器件间的通信功能，并完成各调试功能。这些内容需要在用户程序中设置或使用编译器选项设置。

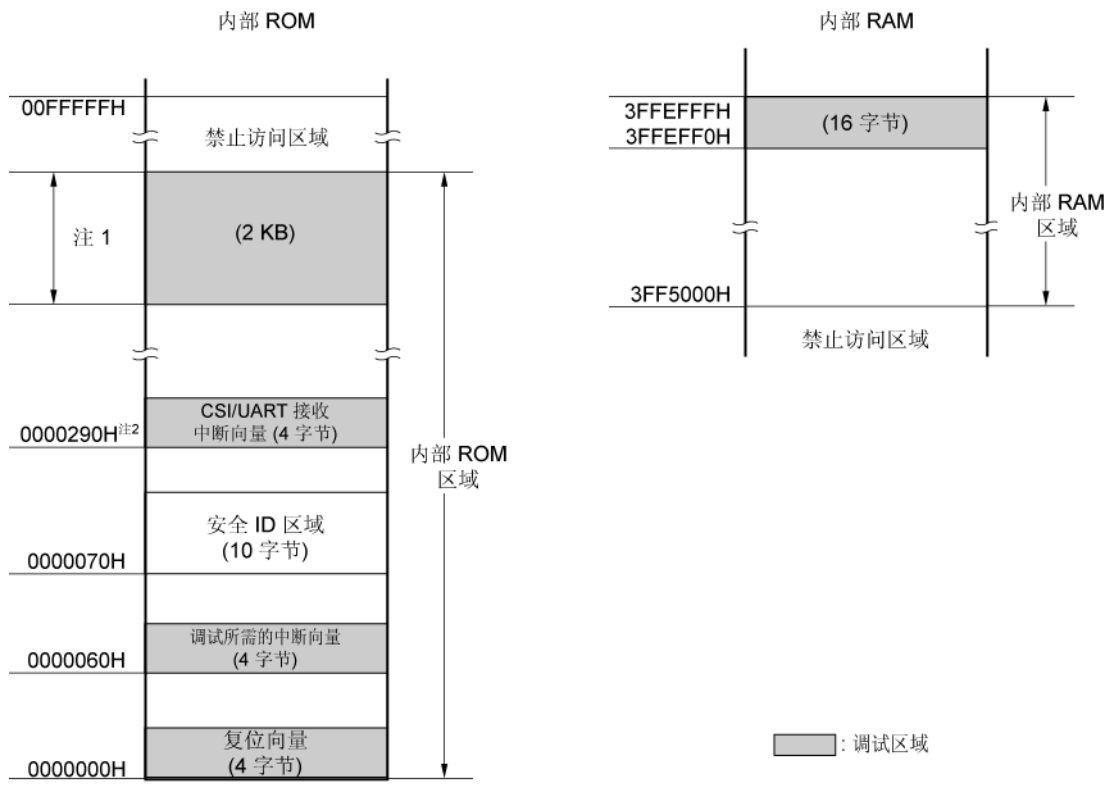
(1) 存储器空间的保留

图 32-5 中的阴影部分是预留区，用于放置调试监控程序，因此用户程序和数据不能分配在这些空间。必须保证这些空间不被用户程序使用。

(2) 安全 ID 设置

ID 码必须嵌入到图 32-5 中 0000070H 至 0000079H 的区域，以防止存储内容被未授权读取。详情参见 **32.3 ROM 安全功能。**

图 32-5. 调试监控程序的内存空间分配



注 1. **地址值根据不同产品而有所差异。**

	内部 ROM 大小	地址值
μPD70F3794	256 KB	003F800H 至 003FFFFH
μPD70F3795	384 KB	005F800H 至 005FFFFH
μPD70F3796	512 KB	007F800H 至 007FFFFH

2. **这是使用 CSIB0 时的地址。当使用 CSIB3 时地址值为 00002F0H。当使用 UARTA0 时地址值为 0000310H。**

(3) 复位向量

复位向量包括调试监控程序所需的转移指令。

[如何保留复位向量]

没有必要刻意保护该区域。但是，当下载一个程序时，调试器根据下列情况重写复位向量。若重写的模式与下列情况不符，则调试器产生一个错误（使用 ID850QB 时为 F0C34 ）。

(a) 当从 0 地址开始连续放置两个空指令时

重写前		重写后
0x0 nop	→	跳转到 0x0 处的调试监控程序
0x2 nop		0x4 xxxx
0x4 xxxx		

(b) 当从 0 地址开始连续放置两个 0xFFFF 值时(已擦除器件)

重写前		重写后
0x0 0xFFFF	→	跳转到 0x0 处的调试监控程序
0x2 0xFFFF		0x4 xxxx
0x4 xxxx		

(c) *jr* 指令放在 0 地址（使用 CA850 时）

重写前		重写后
0x0 jr disp22	→	跳转到 0x0 处的调试监控程序
		0x4 jr disp22 - 4

(d) 从 0 地址开始连续放置 *mov32* 和 *jmp*（使用 IAR 编译 ICCV850 时）

重写前		重写后
0x0 mov imm32,reg1	→	跳转到 0x0 处的调试监控程序
0x6 jmp [reg1]		0x4 mov imm32,reg1
		0xa jmp [reg1]

(e) 在 0 地址处放置用于跳转到调试监控程序的指令

重写前		重写后
跳转到 0x0 处的调试监控程序	→	没有改变

(4) 调试监控程序的保留区域

图 32-5 中的阴影部分是分配给调试监控程序的区域。监控程序对调试通信接口和 CPU 的运行（run）或中断（break）处理过程执行初始化。内部 ROM 区必须填充为 0xFF。该区域禁止被用户程序重写。

[如何保留调试监控程序所需的区域]
若用户程序不使用该区域，则没有必要进行保护。
但是，为了避免在调试器启动时可能出现的问题，建议预先使用编译器保留本区域。
下面给出保留区域的示例，使用瑞萨电子的编译器 CA850。加上汇编源文件和连接伪指令代码，如下所示。

- 汇编源文件(添加下列代码作为汇编源文件。)

```
-- Secures 2 KB space for monitor ROM section
.section "MonitorROM", const
.space     0x800, 0xff

-- Secures interrupt vector for debugging
.section "DBG0"
.space   4, 0xff

-- Secures interrupt vector for serial communication
-- Change the section name according to the serial communication mode used
.section "INTCB0R"
.space   4, 0xff

-- Secures 16-byte space for monitor RAM section
.section "MonitorRAM", bss
.lcomm   monitorramsym, 16, 4     -- defines symbol monitorramsym
```

- 连接伪指令(将下列代码添加到连接伪指令文件)
  下面给出一个使用μPD70F3794 的例子，它的内部 ROM 为 256KB (末地址为 003FFFFH)且内部 RAM 为 16KB (末地址为 3FFEFFFH)。

```
MROMSEG          : !LOAD ?R V0x03f800{
    MonitorROM       = $PROGBITS     ?A MonitorROM;
                     };

MRAMSEG          : !LOAD ?RW V0x03ffeff0{
    MonitorRAM       = $NOBITS          ?AW MonitorRAM;
                     };
```

(5) 通信串行接口的保留

UARTA0，CSIB0 或 CSIB3 用于 MINICUBE2 和目标系统之间的通信。与串行接口模式相关的设置通过调试监控程序来完成，但如果设置被用户程序改变，则可能发生通信错误。
为了防止发生这样的问题，必须在用户程序中对通信串行接口进行保留。

[如何保留通信串行接口]

- 片上调试模式寄存器(OCDM)
  使用 UARTA0，CSIB0 或 CSIB3 的片上调试功能时，设置 OCDM 寄存器为正常模式。务必按照下面步骤进行设置。

  - P05/INTP2/$\overline{\text{DRST}}$ 引脚输入低电平。
  - 将 OCDM0 位设置如下：
    <1> 将 OCDM0 位清 0。
    <2> 在<1>完成之前将 P05/INTP2/$\overline{\text{DRST}}$ 引脚输入固定为低电平。

- 串行接口寄存器
  在用户程序中不要设置 CSIB0，CSIB3 或 UARTA0 相关的寄存器。

- 中断屏蔽寄存器
  使用 CSIB0 时，不要屏蔽发送结束中断(INTCB0R)。使用 CSIB3 时，不要屏蔽发送结束中断(INTCB3R)。使用 UARTA0 时，不要屏蔽接收完成中断 (INTUA0R)。

(a) 使用 CSIB0 时

	7	6	5	4	3	2	1	0
CB0RIC	×	0	×	×	×	×	×	×

(b) 使用 CSIB3 时

	7	6	5	4	3	2	1	0
CB3RIC	×	0	×	×	×	×	×	×

(c) 使用 UARTA0 时

	7	6	5	4	3	2	1	0
UA0RIC	×	0	×	×	×	×	×	×

备注 **×：无须理会**

- 使用 UARTA0 时的端口寄存器
  当使用 UARTA0 时，通过调试监控程序对端口寄存器进行设置，使 TXDA0 和 RXDA0 引脚有效。在调试期间用户程序不要改变下列寄存器的设置。(同样的值可被再次写入)

	7	6	5	4	3	2	1	0
PFC3	×	×	×	×	×	×	0	0

	7	6	5	4	3	2	1	0
PMC3L	×	×	×	×	×	×	1	1

备注 **×：无需理会**

- 使用 CSIB0 的端口寄存器
  使用 CSIB0 时，通过调试监控程序对端口寄存器进行设置，使 SIB0、SOB0、$\overline{\text{SCKB0}}$ 和 HS(PCM0)引脚生效。在调试期间用户程序不要改变下列寄存器的设置。(同样的值可被再次写入)

(a) SIB0、SOB0 和 $\overline{\text{SCKB0}}$ 设置

	7	6	5	4	3	2	1	0
PMC4	×	×	×	×	×	1	1	1

	7	6	5	4	3	2	1	0
PFC4	×	×	×	×	×	×	0	0

(b) HS (PCM0 引脚)设置

	7	6	5	4	3	2	1	0
PMCM	×	×	×	×	×	×	×	0

	7	6	5	4	3	2	1	0
PCM	×	×	×	×	×	×	×	注

注 **禁止写入该位。**
对应 HS 引脚的端口值根据调试器的状态由监控程序改变。为了按照字节指定端口寄存器，用户程序通常会使用读取 - 修改 - 写入方法。但是，如果在写入之前发生一个调试中断，就可能会执行一个意外的操作。

备注 **×：无需理会**

- 使用 CSIB3 的端口寄存器
  使用 CSIB3 时，通过调试监控程序对端口寄存器进行设置，使 SIB3、SOB3、$\overline{\text{SCKB3}}$ 和 HS(PCM0)引脚生效。在调试期间用户程序不要改变下列寄存器的设置。(同样的值可被再次写入)

(a) SIB3、SOB3 和 SCKB3 设置

	7	6	5	4	3	2	1	0
PMC9H	×	×	×	1	1	1	×	×

	7	6	5	4	3	2	1	0
PFC9H	×	×	×	1	1	1	×	×

(b) HS (PCM0 引脚)设置

	7	6	5	4	3	2	1	0
PMCM	×	×	×	×	×	×	×	0

	7	6	5	4	3	2	1	0
PCM	×	×	×	×	×	×	×	注

注 **禁止写入该位。**
对应 HS 引脚的端口值根据调试器的状态由监控程序改变。为了按照字节指定端口寄存器，用户程序通常会使用读取 - 修改 - 写入方法。但是，如果在写入之前发生一个调试中断，就可能会执行一个意外的操作。

备注 **×：无需理会**

32.2.4 注意事项

(1) 被调试设备的处理
不要将用于调试的设备安装在批量生产的产品上，因为在调试过程中 flash 存储器被重写过，所以不能保证 flash 存储器的重写次数。此外，不要将调试监控程序嵌入到批量生产的产品中。

(2) 当中断不能执行时
如果满足下列条件之一，则强制中断无法执行。
- 中断禁止(DI)。
- 串行接口发出的中断被屏蔽，该中断用于 MINICUBE2 和目标器件之间的通信。
- 禁止在通过可屏蔽中断解除待机时进入待机模式。
- MINICUBE2 和目标器件之间的通信模式为 UARTA0，且主时钟已被停止。

(3) 当不执行伪实时 RAM 监控(RRM) 功能和 DMM 功能时
如果满足下列条件之一，将 RRM 功能和 DMM 功能不能操作。
- 中断禁止(DI)。
- 串行接口发出的中断被屏蔽，该中断用于 MINICUBE2 和目标器件之间的通信。
- 禁止在通过可屏蔽中断解除待机时进入待机模式。
- MINICUBE2 和目标器件之间的通信模式为 UARTA0，且主时钟已被停止。
- MINICUBE2 和目标器件之间的通信模式为 UARTA0，并且用于通信的时钟和调试器中指定的时钟不同。

(4) 允许由伪 RRM 和 DMM 功能解除待机模式。
如果满足下列条件之一，则可以由伪 RRM 功能和 DMM 功能来解除待机模式。
- MINICUBE2 和目标器件之间的通信模式为 CSIB0 或 CSIB3。
- MINICUBE2 和目标器件之间的通信模式为 UARTA0，且已提供主时钟

(5) 使用 DMM 功能写入需要特定序列写入的外设 I/O 寄存器。
需要特定序列的外设 I/O 寄存器，不能用 DMM 功能写入。

(6) Flash 自编程
如果分配给调试监控程序的空间被 flash 存储器自编程所重写，那么调试器将不能再正常运行。

(7) 片上调试的电压
当供电电压(V_{DD})在 2.7 至 3.6V 范围之间时，片上调试可用，在低于 2.7V 时不可用。

32.3 ROM安全功能

32.3.1 安全ID

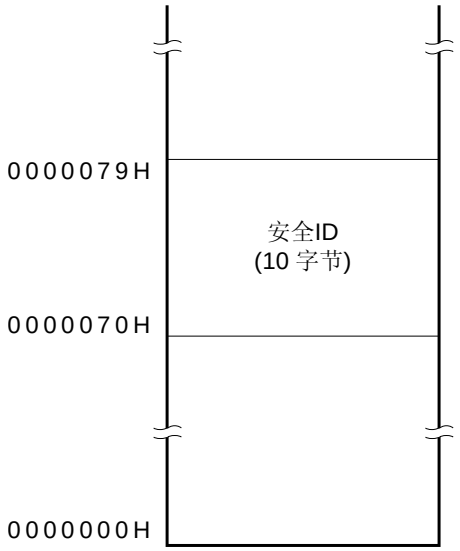
V850ES/JG3-L 的 flash 存储器版本执行鉴定，通过使用一个 10 字节的 ID 码，防止 flash 存储器中的内容在由片上仿真器进行片上调试时被未经授权读取。

在 10 字节的内部 flash 存储区地址从 0000070H 至 0000079H 设置 ID 码，以便允许调试器执行 ID 鉴定。

如果 ID 匹配，则安全保护解除，且允许对 flash 存储器进行读取，也允许使用片上调试仿真器。

- 在 0000070H 至 0000079H 之间设置 10 字节的 ID 码。
- 0000079H 的第 7 位是片上调试仿真器的使能标志。
  (0：禁用，1：使能)
- 当启动片上调试仿真器后，调试器要求输入 ID。当输入的 ID 和 0000070H 至 0000079H 间设置的 ID 码匹配时，调试器启动。
- 即使 ID 码匹配，当片上调试仿真器的使能标志为 0 时，也不能进行调试.

图 32-6. 安全 ID 区



注意事项 flash 存储器擦除后，整个区域都写入 1。

32.3.2 设置

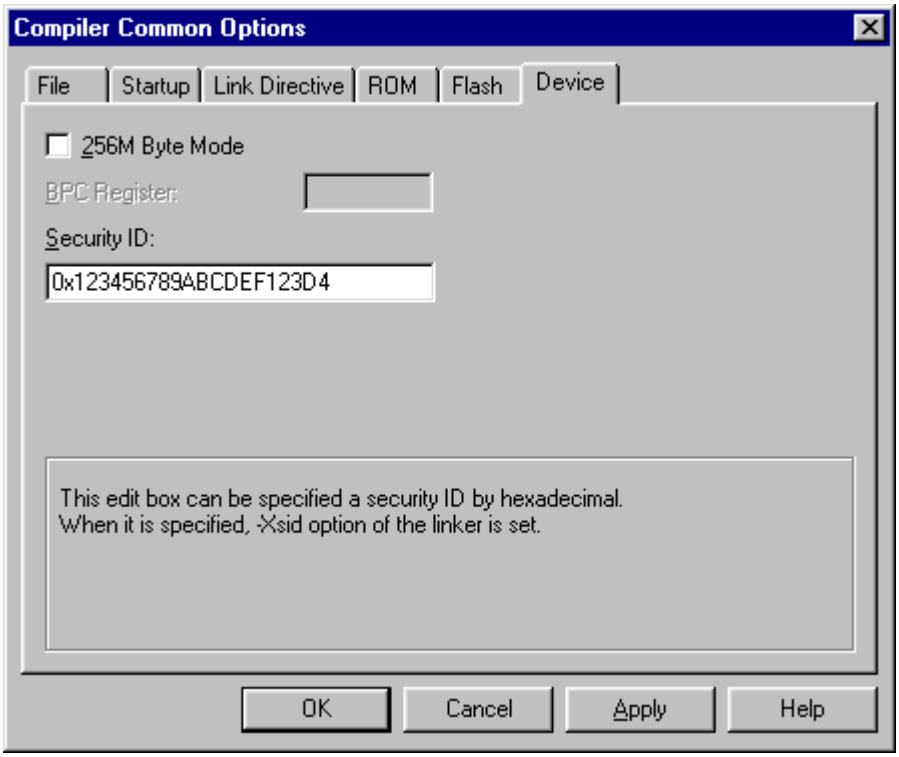
下面在表 32-5 中显示如何设置 ID 码。

当 ID 码按照表 32-5 进行设置时，在 ID850QB 的配置对话框中的 ID 码输入为 “123456789ABCDEF123D4”(ID 码不区分大小写)。

表 32-5. ID 码

地址	值
0x70	0x12
0x71	0x34
0x72	0x56
0x73	0x78
0x74	0x9A
0x75	0xBC
0x76	0xDE
0x77	0XF1
0x78	0x23
0x79	0xD4

ID 码可以使用 PM+中的编译器通用选项对话框来指定。



第三十三章 电气规范

33.1 最大额定值

最大额定值（T_A= 25° C）（1/2）

参数	符号	条件	额定值范围	单位
供应电压	V_{DD}	V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}	–0.5至+4.6	V
	EV_{DD}	V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}	–0.5至+4.6	V
	RV_{DD}	V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}	–0.5至+4.6	V
	UV_{DD}	V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}	–0.5至+4.6	V
	AV_{REF0}	V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}	–0.5至+4.6	V
	AV_{REF1}	V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}	–0.5至+4.6	V
	V_{SS}	V_{SS} = EV_{SS} = AV_{SS}	–0.5至+0.5	V
	AV_{SS}	V_{SS} = EV_{SS} = AV_{SS}	–0.5至+0.5	V
	EV_{SS}	V_{SS} = EV_{SS} = AV_{SS}	–0.5至+0.5	V
输入电压	V_{I1}	P97至P915，PDH0至PDH4，PCM0至PCM3，PCT0，PCT1，PCT4，PCT6，PDL0至PDL15， RESET，FLMD0	–0.5至EV_{DD} + 0.5[注1]	V
	V_{I2}	P10，P11	–0.5至AV_{REF1} + 0.5[注1]	V
	V_{I3}	X1	–0.5至V_{DD} + 0.5[注1]	V
		X2	–0.5至V_{RO}[注2] + 0.5[注1]	
	V_{I4}	P02至P06，P30至P32，P36至P39，P40至P42，P50至P55，P90至P96	–0.5至+6.0	V
	V_{I5}	XT1，XT2	–0.5至V_{RO}[注2] + 0.5	V
	V_{I6}	UDMF，UDPF	–0.5至UV_{DD} + 0.5[注1]	V
模拟输入电压	V_{IAN}	P70至P711	–0.5至AV_{REF0} + 0.5[注1]	V

注 **1.** 确保所有供电电压都不超过最大额定值（最大值）。
2. 片上稳压器输出电压

注意事项 1. 不要将芯片的输出（或输入/输出）引脚直接互连，或直接连接至 V_{DD}、V_{CC} 和 GND。但是，漏极开路或集电极开路引脚可以直接互连。
如果输出引脚能够设置为高阻状态，并且外部电路的输出时序设计时已经考虑避免输出冲突，芯片和外部电路之间的输出引脚才可以直接相连。
2. 任何参数，哪怕瞬间超过最大额定值，都可能使产品性能受损。就是说，产品工作在最大额定值时处于物理损坏的边缘，所以，产品必须在不超出最大额定值的条件下使用。
额定值和条件表示 DC 特性和 AC 特性，操作条件范围代表在正常工作期间的性能保障范围。

备注 除非另外说明，复用功能引脚的特性与端口引脚的特性相同。

最大额定值（T_A = 25° C）（2/2）

参数	符号	条件		额定值范围	单位
输出电流，低	I_{OL}	P02至P06，P30至P32，P36至P39 P40至P42，P50至P55，P90至P915，PDH4	每个引脚	4	mA
			全部引脚总和	50	mA
		PCM0至PCM3，PCT0，PCT1，PCT4，PCT6，PDH0至PDH3，PDL0至PDL15	每个引脚	4	mA
			全部引脚总和	50	mA
		UDMF，UPDF	每个引脚	4	mA
			全部引脚总和	8	mA
		P10，P11	每个引脚	4	mA
			全部引脚总和	8	mA
		P70至P711	每个引脚	4	mA
			全部引脚总和	20	mA
输出电流，高	I_{OH}	P02至P06，P30至P32，P36至P39 P40至P42，P50至P55，P90至P915，PDH4	每个引脚	–4	mA
			全部引脚总和	–50	mA
		PCM0至PCM3，PCT0，PCT1，PCT4，PCT6，PDH0至PDH3，PDL0至PDL15	每个引脚	–4	mA
			全部引脚总和	–50	mA
		UDMF，UPDF	每个引脚	–4	mA
			全部引脚总和	–8	mA
		P10，P11	每个引脚	–4	mA
			全部引脚总和	–8	mA
		P70至P711	每个引脚	–4	mA
			全部引脚总和	–20	mA
工作环境温度	T_A	正常工作模式		–40至+85	°C
		Flash 存储器编程模式		–40至+85	°C
储存温度	T_{stg}			–40至+125	°C

注意事项 1. 不要将芯片的输出（或输入/输出）引脚直接互连，或直接连接至 V_{DD}、 V_{CC} 和 GND。但是，漏极开路或集电极开路引脚可以直接互连。
如果输出引脚能够设置为高阻状态，并且外部电路的输出时序设计时已经考虑避免输出冲突，芯片和外部电路之间的输出引脚才可以直接相连。

2. 任何参数，哪怕瞬间超过最大额定值，都可能使产品性能受损。就是说，产品工作在最大额定值时处于物理损坏的边缘，所以，产品必须在不超出最大额定值的条件下使用。
额定值和条件表示 DC 特性和 AC 特性，操作条件范围代表在正常工作期间的性能保障范围。

备注 除非另外说明，复用功能引脚的额定值与端口引脚的特性相同。

33.2 电容

电容（T_A = 25°C，V_{DD} = EV_{DD} = UV_{DD} = RV_{DD} = AV_{REF0} = AV_{REF1} = V_{SS} = EV_{SS} = AV_{SS} = 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 未测量的引脚返回至0 V			10	pF

33.3 工作条件

工作条件

（T_A = –40 至 +85°C，V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V）

操作时钟	条件	供应电压					单位
		V_{DD}	EV_{DD}	UV_{DD}	RV_{DD} 注1, 2	AV_{REF0}，AV_{REF1}	
f_{XX} = 16 MHz (主时钟)	REGC = 4.7 *u*F， A/D 转换操作， D/A 转换操作， USB操作 (未使用UCLK)， PLL模式	3.0至3.6	3.0至3.6	3.0至3.6	3.0至3.6	3.0至3.6	V
f_{XX} = 10至16 MHz (主时钟)	REGC = 4.7 *u*F， A/D 转换操作， D/A 转换操作， USB操作 (使用UCLK)， PLL模式						
f_{XX} = 1.25 注3至10 MHz (主时钟)	REGC = 4.7 *u*F， A/D 转换操作， D/A 转换操作， USB操作 (使用UCLK)， 时钟直通模式						
f_{XX} = 10至20 MHz (主时钟)	REGC = 4.7 *u*F， A/D 转换操作， D/A 转换操作， USB停止， PLL模式	2.7至3.6	2.7至3.6	2.7至3.6	2.7至3.6	2.7至3.6	V
f_{XX} = 2.5至10 MHz (主时钟)	REGC = 4.7 *u*F， A/D 转换操作， D/A 转换操作 USB停止 时钟直通模式						V
f_{XX} = 2.5至5 MHz (主时钟)	REGC = 4.7 *u*F， A/D 转换停止， D/A 转换停止， USB停止， 时钟直通模式	2.2至3.6	2.2至3.6	2.2至3.6	2.2至3.6	2.2至3.6	V
f_{XX} = 1.25 注3至2.5 MHz (主时钟)	REGC = 4.7 *u*F， A/D 转换停止， D/A 转换停止， USB停止， 时钟直通模式	2.0至3.6	2.0至3.6	2.0至3.6	2.0至3.6	2.0至3.6	V
f_{XT} = 32.768 kHz (副时钟)	REGC = 4.7 *u*F， A/D 转换停止， D/A 转换停止， USB停止	2.0至3.6	2.0至3.6	2.0至3.6	2.0至3.6	2.0至3.6	V
f_{XT} = 32.768 kHz (副时钟)	REGC = 4.7 *u*F， A/D 转换停止， D/A 转换停止， USB停止， RTC备份模式	0至3.6	0至3.6	0至3.6	1.8至3.6	0至3.6	V

注 1. RV_{DD} 可以用作单独的电势，无需将其设置为与其他供电电压相同的电势。除非另外指定，在上述操作条件的方法使用 RV_{DD}。

2. 在 V_{DD} 和 RV_{DD} 之间连接二极管时，请确保考虑二极管引起的压降，应调整 V_{DD} 电压以便保证 RV_{DD} 不会低于它的下限。

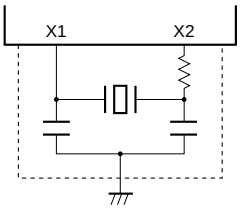
3. 只有当 CKTHSEL 寄存器被设置为 1 时 (指定时钟直通频率 2 分频)，才能指定 1.25 MHz 的操作时钟。

33.4 振荡器特性

33.4.1 主时钟振荡器特性

(1) 主时钟振荡器特性

（T_A = –40 至 +85°C，V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V）

谐振器	电路示例	参数	条件		最小值	典型值	最大值	单位
陶瓷谐振器/晶体谐振器		振荡频率 （fx）注1	时钟直通模式	V_{DD} = 2.0至3.6 V	2.5		2.5	MHz
				V_{DD} = 2.2至3.6 V	2.5		6	MHz
				V_{DD} = 2.7至3.6 V	2.5		10	MHz
			PLL模式注2	USB停止，V_{DD} = 2.7至3.6 V	2.5		6	MHz
				USB操作，未使用UCLK V_{DD} = 3.0至3.6 V	6		6	MHz
				USB操作，使用UCLK V_{DD} = 3.0至3.6 V	2.5		6	MHz
		振荡稳定时间注3	复位刚结束后，V_{DD} = 2.0至3.6 V		注4	注5		μs
			在STOP模式被释放后	时钟直通模式下，V_{DD} = 2.0至3.6 V	200注6	注7		μs
				PLL模式下，V_{DD} = 2.7至3.6 V	400注8	注7		μs
			在IDLE2模式被释放后	时钟直通模式下，V_{DD} = 2.0至3.6 V	100注5	注7		μs
				PLL模式下，V_{DD} = 2.7至3.6 V	400注8	注7		μs

注
1. 以上所示的振荡频率仅表示振荡器特性。使用 V850ES/JG3-L，内部工作条件不要超出 **AC 特性**、**DC 特性**和**操作条件**列出的的额定值。
2. 振荡频率的 PLL 设置，参见 **33.4.3 PLL 特性**。
3. 从振荡开始到振荡电路稳定所需的等待时间。
4. reset 解除后的振荡稳定时间需要严格遵守选项字节的设置值。详情参见**第二十九章 选项字节**。
5. 振荡稳定时间根据选项字节的设置有所差异。详情参见**第二十九章 选项字节**。
6. 稳压器和 flash 存储器所需的建立时间。使用 OSTS 寄存器来确保建立时间。
7. 该值随 OSTS 寄存器的设置而变化。
8. 设置稳压器、flash 存储器和 PLL 所需的建立时间。使用 OSTS 寄存器来确保建立时间。

注意事项 1. 在使用主时钟振荡器时，上图中虚线框部分的连线应遵从如下的连接方式，避免线间电容的负面影响。

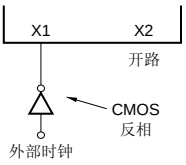
- **连线要保证尽可能短。**

- **振荡器连线不要和其它的信号线互相交叉。**
- **布线时不要将振荡器连线靠近通过大波动电流的信号线**
- **保持振荡器电容的地电位和 V_{SS} 相同。**
- **不要将电容的接地点连接在大电流通过的地模式。**
- **不要从振荡器取信号。**

注意事项 2. 当主时钟停止，设备运行于副时钟时，在由副时钟切换回主时钟之前，应该由程序来实现振荡稳定时间的等待。

(2) 外部时钟

（T_A = –40 至 +85°C，V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.7 注1 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V）

谐振器	电路举例	参数	条件	最小值	典型值	最大值	单位
外部时钟		输入频率 （f_X）注2	USB停止， 时钟直通模式	2.5		6	MHz
			USB停止， PLL模式 注3	2.5		6	MHz
			USB操作， (未使用UCLK) PLL模式 注3	6		6	MHz
			USB操作， (使用UCLK) 时钟直通模式	2.5		6	MHz
			USB操作， (使用UCLK) PLL模式 注3	2.5		6	MHz
		V_{IH}	X1	2.3		V_{DD}	V
		V_{IL}	X1	V_{SS}		0.4	V
		输入波形占空比	X1	45	50	55	%

注 1. 当 USB 操作时，最低 3.0 V。

2. 输入频率仅表示 X1 输入时钟电路特性。在保证内部操作条件不超过 AC 特性额定值的情况下使用 V850ES/JG3-L。

3. 振荡频率的 PLL 设置，参见 **33.4.3 PLL 特性**。

注意事项 1. 复位后，确保断开内部反馈电阻的连接（设置 PCC.MFRC = 1）。

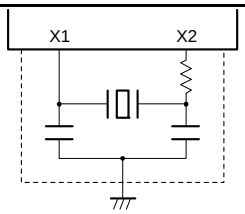
2. 保留 X2 引脚开路。

3. 请确保 CMOS 反相器尽可能靠近 X1 引脚。

33.4.2 副时钟振荡器特性

副时钟振荡器特性

（T_A = –40 至 +85°C，V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.0 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V）

谐振器	电路举例	参数	条件	最小值	典型值	最大值	单位
晶体谐振器		振荡频率（f_{XT}）注 1		32	32.768	35	kHz
		振荡稳定时间注 2				10	s

注 **1.** 以上所示的振荡频率仅表示振荡器特性。使用 V850ES/JG3-L，内部工作条件不要超出 **AC 特性**、**DC 特性**和**操作条件**中列举的额定值。

2. 从 RV_{DD} 达到振荡电压范围（2.0 V（最小值））起，到晶体振荡器稳定所需的时间。

注意事项 **1. 在使用副时钟振荡器时，上图中虚线框部分的连线应遵从如下的连接方式，避免线间电容的负面影响。**

- **连线要保证尽可能短。**
- **振荡器连线不要和其它的信号线互相交叉。**
- **布线时不要将振荡器连线靠近通过大波动电流的信号线**
- **保持振荡器电容的地电位和 Vss 相同。**
- **不要将电容的接地点连接在大电流通过的地模式。**
- **不要从振荡器取信号。**

2. 为了降低功耗，副时钟振荡器被设计为低振幅电路，而比起主时钟振荡器更容易受由于噪声引起的干扰。所以，在使用副时钟时需要特别注意布线的方式。

3. 有关谐振器的选择和振荡器常量，需要用户自行评估振荡特性或者向谐振器生产商申请对谐振器的评估。

33.4.3 PLL 特性

（T_A = –40 至 +85°C，V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.7 注至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 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	2.5		5	MHz
		×8, 1/3	6		6	MHz
输出频率	f_{XX}	×4	10		20	MHz
		×8, 1/3	16		16	MHz
锁定时间	t_{PLL}	V_{DD} 达到 2.7 V（最小值）之后			200	*us*

注 当 USB 操作时，最低 3.0 V。

33.4.4 内部振荡器特性

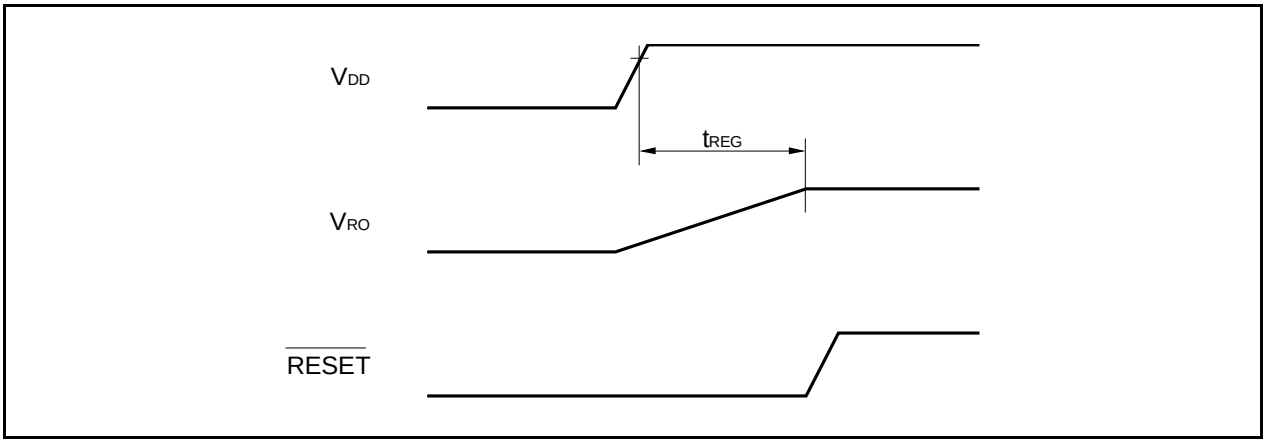
(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.0 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

参数	符号	条件	最小值	典型值	最大值	单位
输出频率	f_R		100	220	400	kHz

33.5 稳压器特性

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 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 MHz (MAX.)	2.7		3.6	V
		f_{XX} = 5 MHz (MAX.)	2.2		3.6	V
		f_{XX} = 2.5 MHz (MAX.)	2.0		3.6	V
		数据保持 (STOP模式)	1.9		3.6	V
输出电压	V_{RO}	V_{DD} = 2.7至3.6 V		2.5		V
稳压器输出稳定时间	t_{REG}	V_{DD} 达到 2.7 V（最小值）后，稳定电容 C = 4.7 *u*F（推荐值）连接到 REGC 引脚			1	ms
		V_{DD} 达到 2.2 V（最小值）后，稳定电容 C = 4.7 *u*F（推荐值）连接到 REGC 引脚			3.5	ms
		V_{DD} 达到 2.0 V（最小值）后，稳定电容 C = 4.7 *u*F（推荐值）连接到 REGC 引脚			4.5	ms
外部电容	REGC	连接至 REGC引脚的外部电容的允许误差	3.76	4.70	5.64	*u*F



33.6 DC特性

33.6.1 引脚特性

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.0 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V) (1/2)

参数	符号	条件	最小值	典型值	最大值	单位
输入电压，高	V_{IH1}	$\overline{RESET}$, FLMD0, P97至P915	0.8EV_{DD}		EV_{DD}	V
	V_{IH2}	P02至P06, P30至P32, P36, P37, P42, P50至P55, P92至P96	0.8EV_{DD}		5.5	V
	V_{IH3}	P38, P39, P40, P41, P90, P91	0.7EV_{DD}		5.5	V
	V_{IH4}	PCM0至PCM3, PCT0, PCT1, PCT4, PCT6, PDH0至PDH4, PDL0至PDL15	0.7EV_{DD}		EV_{DD}	V
	V_{IH5}	P70至P711	0.7AV_{REF0}		AV_{REF0}	V
	V_{IH6}	P10, P11	0.7AV_{REF1}		AV_{REF1}	V
	V_{IH7}	UDMF, UDPF	2.0		UV_{DD}	V
输入电压，低	V_{IL1}	$\overline{RESET}$, FLMD0, P97至P915	EV_{SS}		0.2EV_{DD}	V
	V_{IL2}	P02至P06, P30至P32, P36, P37, P42, P50至P55, P92至P96	EV_{SS}		0.2EV_{DD}	V
	V_{IL3}	P38, P39, P40, P41, P90, P91	EV_{SS}		0.3EV_{DD}	V
	V_{IL4}	PCM0至PCM3, PCT0, PCT1, PCT4, PCT6, PDH0至PDH4, PDL0至PDL15	EV_{SS}		0.3EV_{DD}	V
	V_{IL5}	P70至P711	AV_{SS}		0.3AV_{REF0}	V
	V_{IL6}	P10, P11	AV_{SS}		0.3AV_{REF1}	V
	V_{IH7}	UDMF, UDPF	V_{SS}		0.3UV_{DD}	V
输入漏电电流，高	I_{LIH}	V_I = V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}			5	*u*A
输入漏电电流，低	I_{LIL}	V_I = 0 V			–5	*u*A
输出漏电电流，高	I_{LOH}	V_O = V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}			5	*u*A
输出漏电电流，低	I_{LOL}	V_O = 0 V			–5	*u*A

备注 除非另外说明，复用功能引脚的特性与端口引脚的特性相同。

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.0 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V) (2/2)

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至P06, P30至P32, P36至P39, P40至P42, P50至P55, P90至P915, PDH4	每个引脚 I_{OH} = –1.0 mA	全部引脚总和 –20 mA	EV_{DD} – 1.0		EV_{DD}	V
			每个引脚 I_{OH} = –100 *u*A	全部引脚总和 –3.8 mA	EV_{DD} – 0.5		EV_{DD}	V
	V_{OH2}	PCM0至PCM3, PCT0, PCT1, PCT4, PCT6, PDH0至PDH3, PDL0至PDL15	每个引脚 I_{OH} = –1.0 mA	全部引脚总和 –20 mA	EV_{DD} – 1.0		EV_{DD}	V
			每个引脚 I_{OH} = –100 *u*A	全部引脚总和 –2.8 mA	EV_{DD} – 0.5		EV_{DD}	V
	V_{OH3}	P70至P711	每个引脚 I_{OH} = –0.4 mA	全部引脚总和 –4.8 mA	AV_{REF0} – 1.0		AV_{REF0}	V
			每个引脚 I_{OH} = –100 *u*A	全部引脚总和 –1.2 mA	AV_{REF0} – 0.5		AV_{REF0}	V
	V_{OH4}	P10, P11	每个引脚 I_{OH} = –0.4 mA	全部引脚总和 –0.8 mA	AV_{REF1} – 1.0		AV_{REF1}	V
			每个引脚 I_{OH} = –100 *u*A	全部引脚总和 –0.2 mA	AV_{REF1} – 0.5		AV_{REF1}	V
	V_{OH5}	UDMF, UDPF	每个引脚 I_{OH} = –1.0 mA	全部引脚总和 –2 mA	UV_{DD} – 1.0		UV_{DD}	V
			每个引脚 I_{OH} = –100 *u*A	全部引脚总和 –200 *u*A	UV_{DD} – 0.5		UV_{DD}	V
输出电压，低	V_{OL1}	P02至P06, P30至P32, P36至P39, P42, P50至P55, P92至P915, PDH4	每个引脚 I_{OL} = 1.0 mA	全部引脚总和 20 mA	0		0.4	V
	V_{OL2}	P38, P39, P40, P41, P90, P91	每个引脚 I_{OL} = 3.0 mA		0		0.4	V
	V_{OL3}	PCM0至PCM3, PCT0, PCT1, PCT4, PCT6, PDH0至PDH3, PDL0至PDL15	每个引脚 I_{OL} = 1.0 mA	全部引脚总和 20 mA	0		0.4	V
	V_{OL4}	P10, P11, P70至P711	每个引脚 I_{OL} = 0.4 mA	全部引脚总和 5.6 mA	0		0.4	V
	V_{OL5}	UDMF, UDPF	每个引脚 I_{OL} = 3.0 mA	全部引脚总和 6 mA	0		0.4	V
软件下拉电阻[注]	R_1	P05	V_I = V_{DD}		10	20	100	kΩ

注 仅限 $\overline{\text{DRST}}$ 引脚（由 OCDM 寄存器控制）

备注 **1.** 除非另外说明，复用功能引脚的特性与端口引脚的特性相同。

2. 当单个引脚没有满足 I_{OH} 和 I_{OL} 条件但满足所有引脚的总值符合该条件时，则仅该引脚不满足 DC 特性。

33.6.2 供电电流特性

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

参数	符号	条件		最小值	典型值注1	最大值注2	单位
供电电流	I_{DD1}	正常操作	f_{XX} = 20 MHz (f_X = 5 MHz)注3, USB停止		14	待定	mA
			f_{XX} = 16 MHz (f_X = 6 MHz)注3, USB操作		待定	34	mA
			f_{XX} = 10 MHz (f_X = 10 MHz), PLL关闭注3, USB停止		6	10	mA
	I_{DD2}	HALT模式	f_{XX} = 20 MHz (f_X = 5 MHz)注3, USB停止		7.5	14	mA
	I_{DD3}	IDLE1模式	f_{XX} = 5 MHz (f_X = 5 MHz), PLL关闭注3		0.6	1	mA
	I_{DD4}	IDLE2模式	f_{XX} = 5 MHz (f_X = 5 MHz), PLL关闭注3		0.28	0.5	mA
	I_{DD5}	副时钟操作模式	f_{XT} = 32.768 kHz, 主时钟停止, 内部振荡器停止, PLL关闭 REGOVL0 = 02H (低电压副时钟操作模式), CSIBn停止注4, UARTA0停止注4		18		uA
	I_{DD6}	副IDLE模式	f_{XT} = 32.768 kHz, 主时钟停止, 内部振荡器停止, PLL关闭 REGOVL0 = 02H (低电压副IDLE模式), CSIBn停止注4, UARTA0停止注4		2.5	50	uA
	I_{DD7}	STOP模式	副时钟停止, 内部振荡器停止, REGOVL0 = 01H (低电压STOP模式), T_A = 25°C		1.5	3.0	uA
			副时钟停止, 内部振荡器停止, REGOVL0 = 01H (低电压STOP模式), T_A = 85°C			45	uA
			副时钟操作，内部振荡器停止, REGOVL0 = 01H (低电压STOP模式), CSIBn停止注4, UARTA0停止注4		2.5	50	uA
	I_{DD8}	自编程模式	f_{XX} = 20 MHz (f_X = 5 MHz)		14	24	mA
LVI电流	I_{LVI}				1.2	3	uA
WDT，内部振荡电流	I_{WDT}				5		uA
RTC备份模式	I_{RTC}	RTC备份模式	副时钟操作，T_A = 70°C, RV_{DD} 电压，V_{DD} = 0 V			1	uA

注 1. 典型值电流为 V_{DD} = EV_{DD} = RV_{DD} = 3.3 V，T_A = 25°C 条件下的值。
典型值值并不保证对所有设备适用。
2. 最大值电流是：在 V_{DD} = EV_{DD} = RV_{DD} = 3.6 V，T_A = –40 至 +85°C 最恶劣条件下有待讨论的特性值。
3. 典型值表示当钟表定时器运行且 TMM 基于钟表定时器中断来计数。
最大值表示在引脚状态不改变的情况下，当所有可用功能都作为外设功能执行操作时的电流值。
但是，不包括 I_{LVI} 和 I_{WDT}。
4. CSIBn 和 UARTA0 可以分别使用 $\overline{\text{SCKBn}}$ 和 ASCKA0 工作，但是目标规范是 CSIBn 和 UARTA0 停止时的电流值。

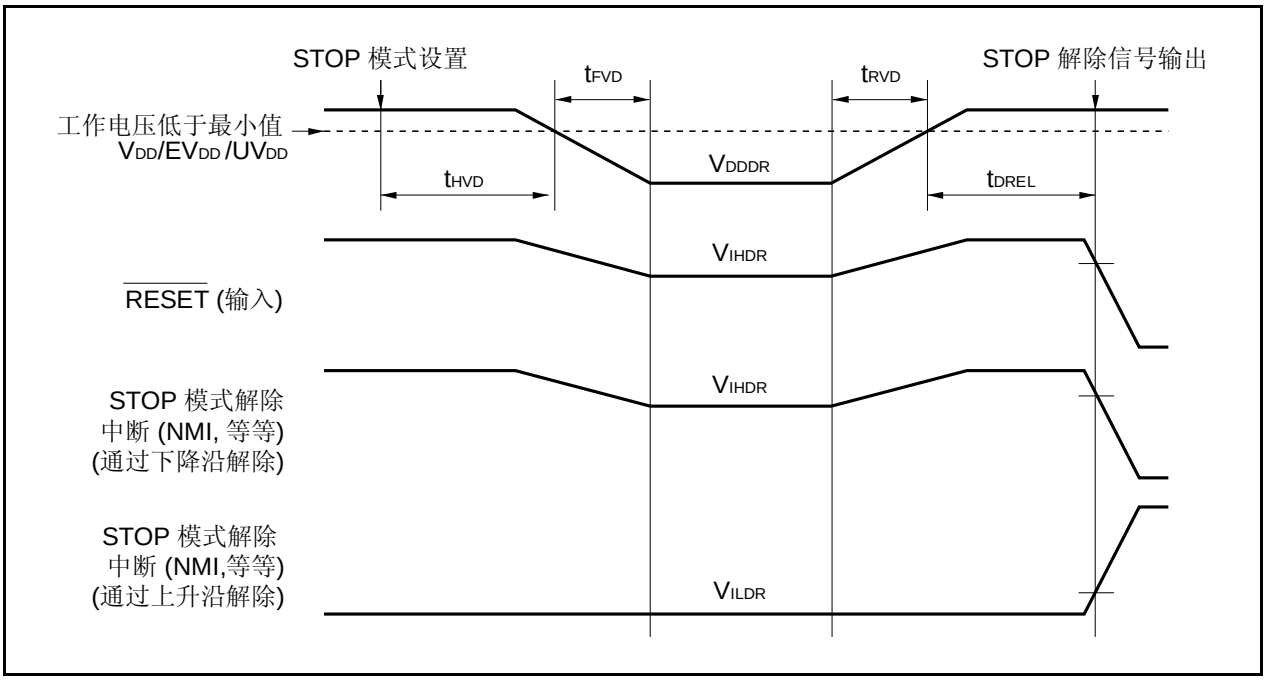
备注 关于操作电压的详情，参见 **33.3 操作条件**。

33.6.3 数据保持特性（STOP模式下）

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

参数	符号	条件	最小值	典型值	最大值	单位
数据保持电压	V_{DDDR}	STOP 模式（所有功能停止）	1.9		3.6	V
数据保持电流	I_{DDDR}	副时钟停止， 内部振荡器停止 T_A = 85°C			45	uA
供应电压上升时间	t_{RVD}		200			us
供应电压下降时间	t_{FVD}		200			us
供应电压保持时间	t_{HVD}	设置STOP 模式后	0			ms
STOP 解除信号输入时间	t_{DREL}	V_{DD} 达到最低操作电压之后（参见 **33.3 操作条件**）	0			ms
数据保持输入电压，高	V_{IHDR}	V_{DD} = EV_{DD} = UV_{DD} = V_{DDDR}	0.9V_{DDDR}		V_{DDDR}	V
数据保持输入电压，低	V_{ILDR}	V_{DD} = EV_{DD} = UV_{DD} = V_{DDDR}	0		0.1V_{DDDR}	V

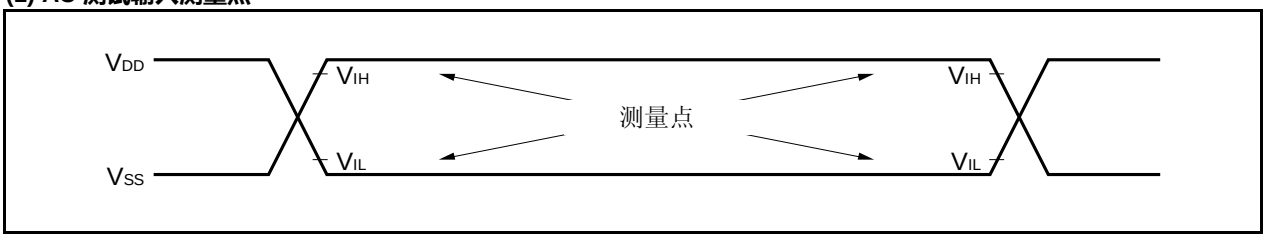
注意事项 转换到 STOP 模式以及从 STOP 模式中恢复都必须在额定工作范围内进行。



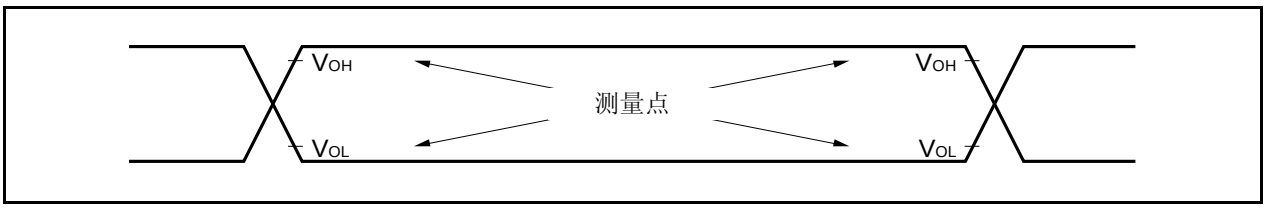
33.7 AC特性

33.7.1 测量条件

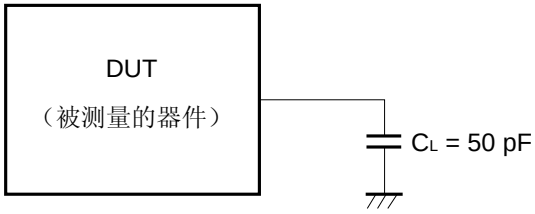
(1) AC 测试输入测量点



(2) AC 测试输出测量点



(3) 负载条件

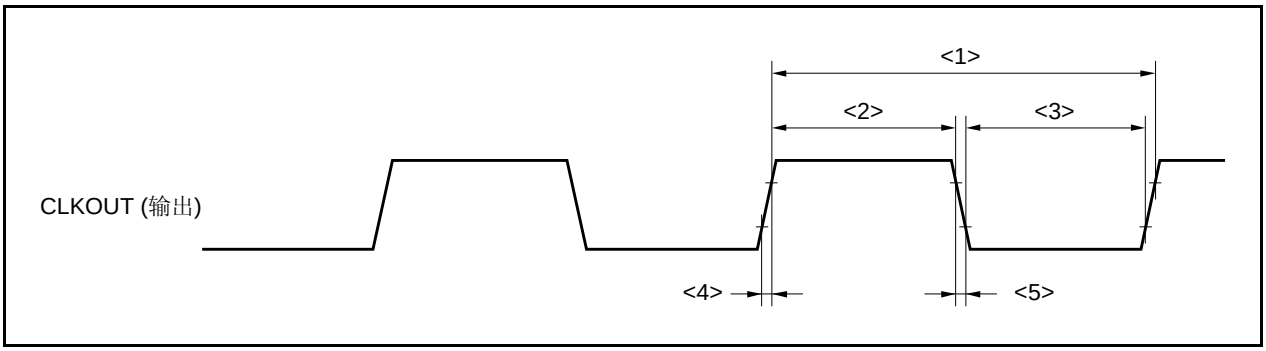


注意事项 如果由于电路配置使得负载电容超过 50 pF，可插入一个缓冲器或其它手段使设备的负载电容等于或小于 50 pF。

33.7.2 CLKOUT输出时序

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.7 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号		条件	最小值	最大值	单位
输出周期	t_{CYK}	<1>		50 ns	31.25 *u*s	
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



33.7.3 总线时序

这些值只是使用的访问方法（CLKOUT 同步或异步）必须满足的值。没有必要满足两种方式的值。

（1） 复用总线/分离总线模式

（a） 读/写周期（CLKOUT 异步）

（T_A = –40 至 +85°C，V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.7 至 3.6 V，V_{SS} = EV_{SS} = AV_{SS} = 0 V，C_L = 50 pF）

参数	符号		条件	最小值	最大值	单位
地址建立时间（至 ASTB↓）	t_{SAST}	<6>		（0.5 + t_{ASW}）T – 20		ns
地址保持时间（从 ASTB↓起）	t_{HSTA}	<7>		(0.5 + t_{AHW})T – 15		ns
自 $\overline{RD}$↓至地址悬空的延迟时间	t_{FRDA}	<8>			16	ns
自地址有效到数据输入建立时间	t_{SAID}	<9>			(2 + n + t_{ASW} + t_{AHW})T – 35	ns
自 $\overline{RD}$↓到数据输入建立时间	t_{SRID}	<10>			（1 + n）T – 25	ns
自 ASTB↓至 $\overline{RD}$、$\overline{WRm}$↓的延迟时间	$t_{DSTRDWR}$	<11>		(0.5 + t_{AHW})T – 15		ns
数据输入保持时间（自 $\overline{RD}$↑起）	t_{HRDID}	<12>		0		ns
自 $\overline{RD}$↑到地址输出时间	t_{DRDA}	<13>		（1 + i）T – 15		ns
自 $\overline{RD}$，$\overline{WRm}$↑到 ASTB↑的延迟时间	$t_{DRDWRST}$	<14>		0.5T – 15		ns
自 $\overline{RD}$↑到 ASTB↓的延迟时间	t_{DRDST}	<15>		（1.5 + i + t_{ASW}）T – 15		ns
$\overline{RD}$、$\overline{WRm}$ 低电平宽度	t_{WRDWRL}	<16>		（1 + n）T – 15		ns
ASTB 高电平宽度	t_{WSTH}	<17>		(1 + i + t_{ASW})T – 15		ns
自 $\overline{WRm}$↓起的数据输出时间	t_{DWROD}	<18>			15	ns
数据输出建立时间（至 $\overline{WRm}$↑）	t_{SODWR}	<19>		（1 + n）T – 20		ns
数据输出保持时间（自 $\overline{WRm}$↑起）	t_{HWROD}	<20>		T – 15		ns
$\overline{WAIT}$ 建立时间（至地址有效）	t_{SAWT1}	<21>	n ≥ 1		(1.5 + t_{ASW} + t_{AHW})T – 35	ns
	t_{SAWT2}	<22>			(1.5 + n + t_{ASW} + t_{AHW})T – 35	ns
$\overline{WAIT}$ 保持时间（自地址有效起）	t_{HAWT1}	<23>	n ≥ 1	(0.5 + n + t_{ASW} + t_{AHW}）T		ns
	t_{HAWT2}	<24>		(1.5 + n + t_{ASW} + t_{AHW})T		ns
$\overline{WAIT}$ 建立时间 （至 ASTB↓）	t_{SSTWT1}	<25>	n ≥ 1		（1 + t_{AHW}）T – 25	ns
	t_{SSTWT2}	<26>			(1 + n + t_{AHW})T – 25	ns
$\overline{WAIT}$ 保持时间 （自 ASTB↓起）	t_{HSTWT1}	<27>	n ≥ 1	（n + t_{AHW}）T		ns
	t_{HSTWT2}	<28>		（1 + n + t_{AHW}）T		ns
自 $\overline{RD}$↑到地址保持时间	t_{HRDA2}	<29>		（1 + i）T – 15		ns
自 $\overline{WRm}$↑到地址保持时间	t_{HWRA2}	<30>		T – 15		ns

备注

1. t_{ASW}： 地址建立等待的时钟数目（0 或 1）。
t_{AHW}： 地址保持等待的时钟数目（0 或 1）。

2. T = 1/f_{CPU}（f_{CPU}：CPU 工作时钟频率）

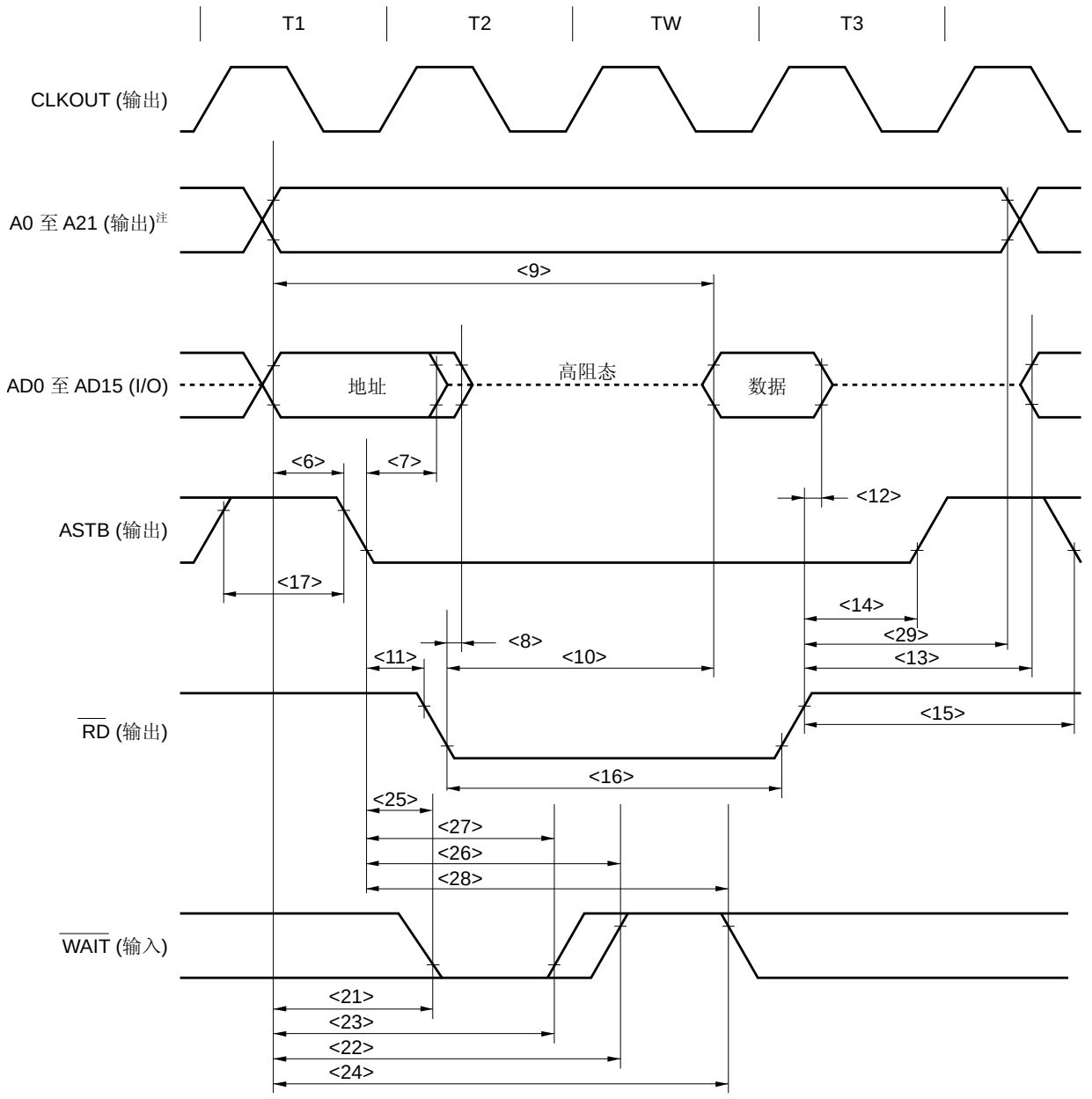
3. n：在总线周期中插入的等待时钟数目
当插入可编程等待时，采样时序改变。

4. m = 0，1

5. i：读周期后插入的空闲状态数目（0 或 1）

6. 在上述说明中的值是当占空比为 1：1 的时钟从 X1 输入时的值。

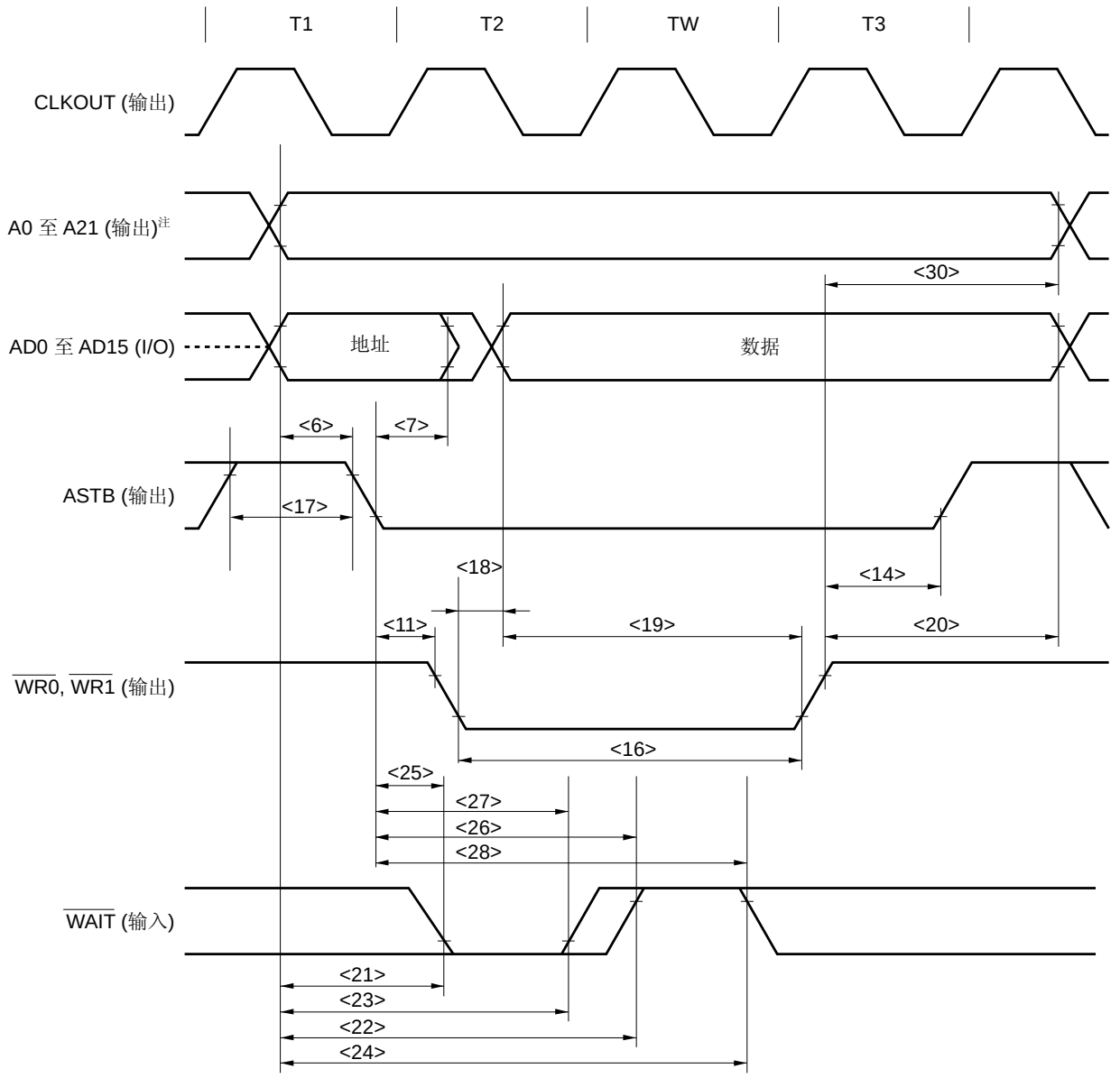
读周期（CLKOUT 异步）：复用总线/分离总线模式



注 **在复用总线中**，A16 至 A21

备注 $\overline{WR0}$ 和 $\overline{WR1}$ 是高电平。

写周期（CLKOUT 异步）：复用总线/分离总线模式



注 **在复用总线中**，A16 至 A21

备注 $\overline{RD}$ 是高电平。

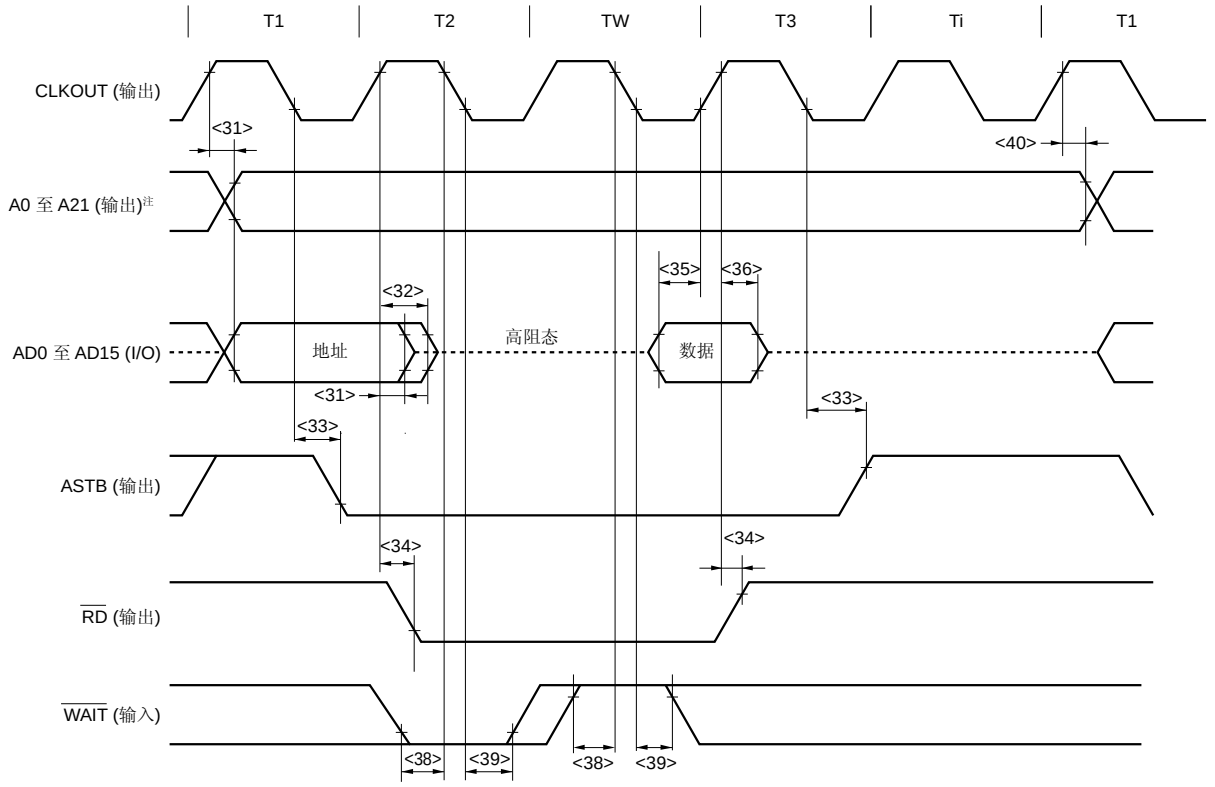
（b）读/写周期（CLKOUT 同步）：复用总线/分离总线模式

（T_A = –40 至 +85°C，V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.7 至 3.6 V，V_{SS} = EV_{SS} = AV_{SS} = 0 V，C_L = 50 pF）

参数	符号		条件	最小值	最大值	单位
自 CLKOUT↑到地址有效的延迟时间	t_{DKA}	<31>		0	25	ns
自 CLKOUT↑到地址悬空延迟时间	t_{FKA}	<32>		0	19	ns
自 CLKOUT↓到 ASTB 的延迟时间	t_{DKST}	<33>		–12	7	ns
自 CLKOUT↑到 $\overline{RD}$、$\overline{WRm}$ 的延迟时间	t_{DKRDWR}	<34>		–5	14	ns
数据输入建立时间（至 CLKOUT↑）	t_{SIDK}	<35>		15		ns
数据输入保持时间（自 CLKOUT↑）	t_{HKID}	<36>		5		ns
自 CLKOUT↑起数据输出延迟时间	t_{DKOD}	<37>			19	ns
$\overline{WAIT}$ 建立时间（至 CLKOUT↓）	t_{SWTK}	<38>		20		ns
$\overline{WAIT}$ 保持时间（自 CLKOUT↓）	t_{HKWT}	<39>		5		ns
自 CLKOUT↑到地址保持时间	t_{HKA2}	<40>		0	25	ns
自 CLKOUT↑到数据输出保持时间	t_{HKDW}	<41>		0		ns
自 CLKOUT↑到地址保持时间	t_{HKAW}	<42>		0		ns

备注 **1.** m = 0，1
2. 在上述说明中的值是当占空比为 1：1 的时钟从 X1 输入时的值。
3. CLKOUT 输出时序的详情，参见 **33.7.2 CLKOUT 输出时序**。

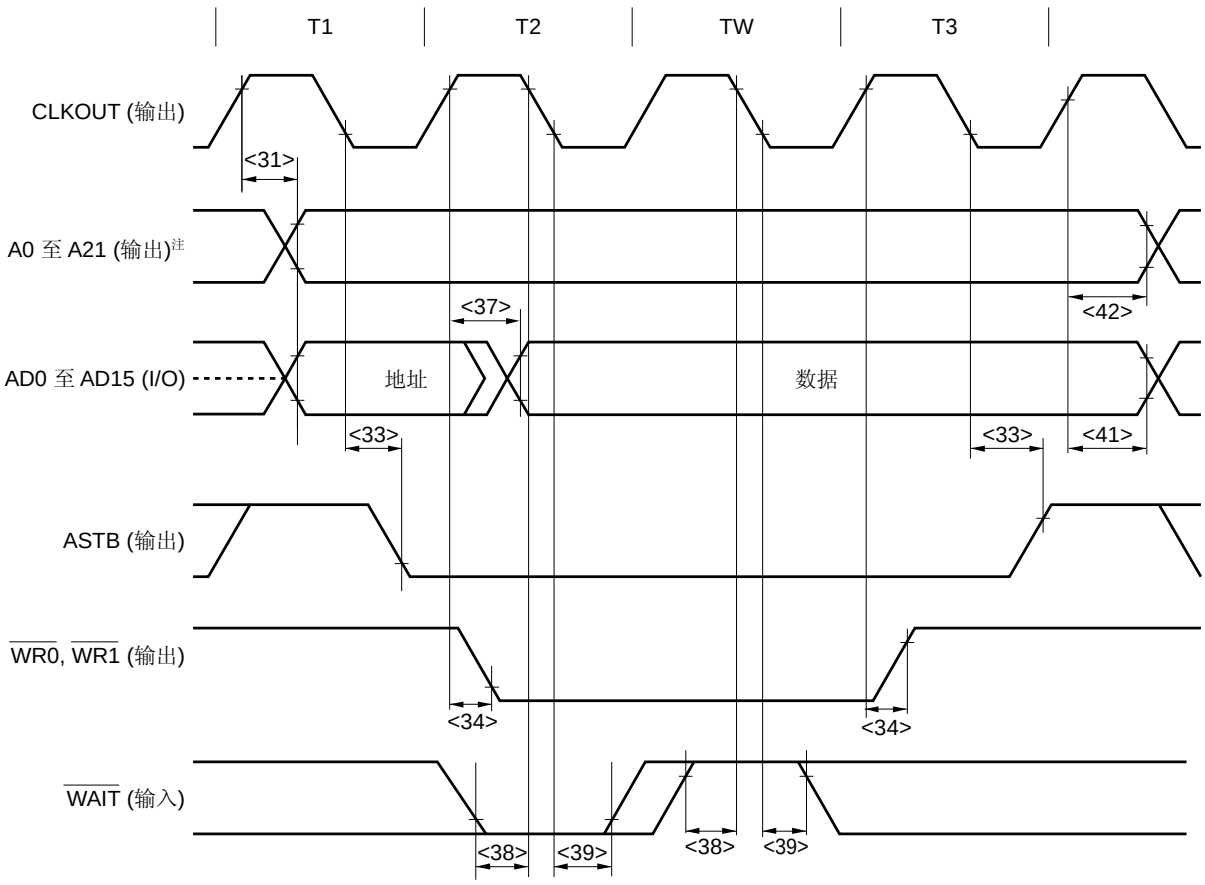
读周期（CLKOUT 同步）：复用总线/分离总线模式



注 **在复用总线中**，A16 至 A21

备注 $\overline{WR0}$ 和 $\overline{WR1}$ 是高电平。

写周期（CLKOUT 同步）：复用总线模式



注　　在复用总线中，A16 至 A21

备注　　$\overline{RD}$ 是高电平。

（2） 总线保持

（a） CLKOUT 异步

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.7 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

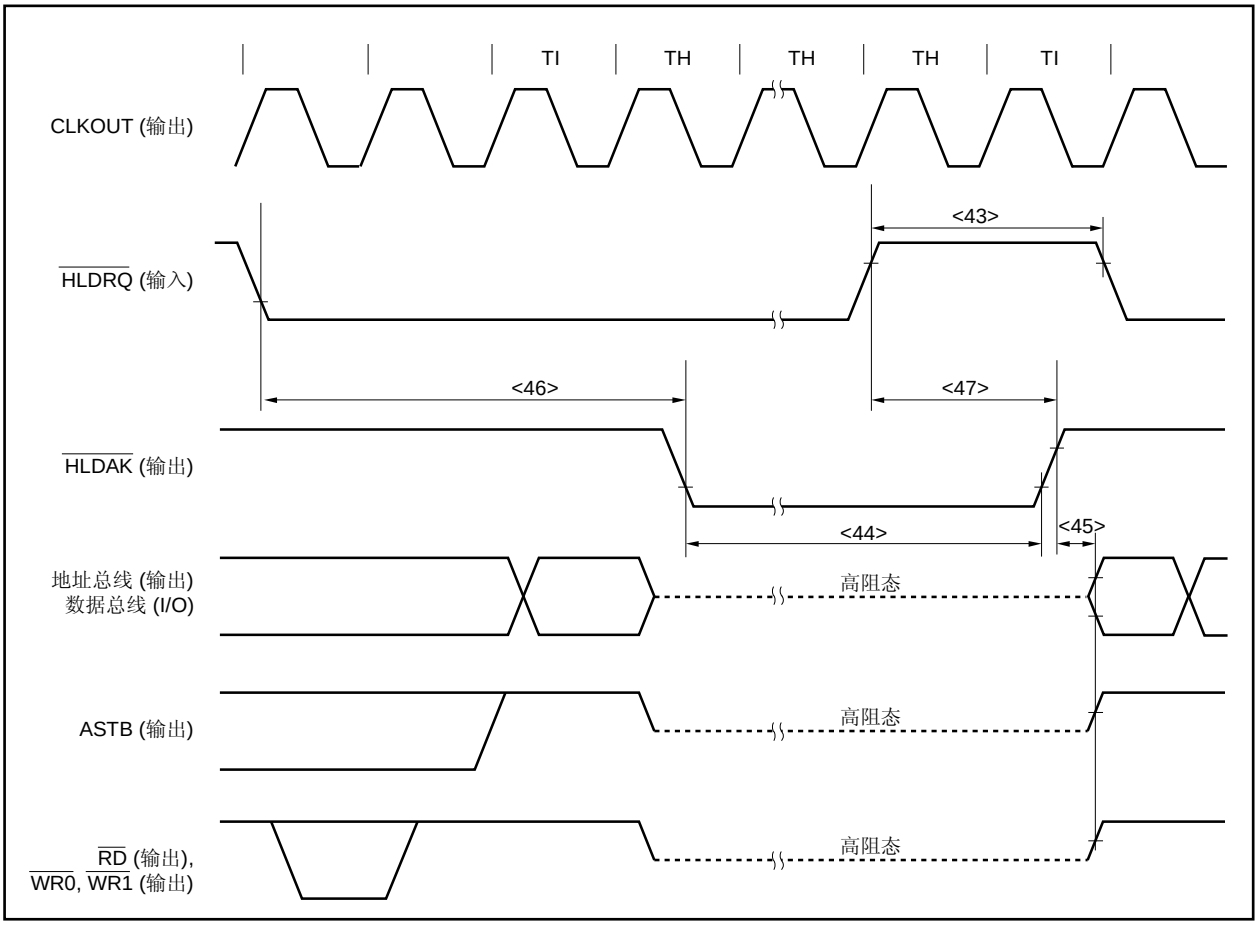
参数	符号		条件	最小值	最大值	单位
$\overline{\text{HLDRQ}}$ 高电平宽度	t_{WHQH}	<43>		T + 10		ns
$\overline{\text{HLDAK}}$ 低电平宽度	t_{WHAL}	<44>		T – 15		ns
自 $\overline{\text{HLDAK}}$↑到总线输出的延时	t_{DHAC}	<45>		–3		ns
自 $\overline{\text{HLDRQ}}$↓ 到 $\overline{\text{HLDAK}}$↓ 的延时	t_{DHQHA1}	<46>			（2n + 7.5）T + 26	ns
自 $\overline{\text{HLDRQ}}$↑ 到 $\overline{\text{HLDAK}}$↑ 的延时	t_{DHQHA2}	<47>		0.5T	1.5T + 26	ns

备注 1. T = 1/f_{CPU}（f_{CPU}：CPU 工作时钟频率）

2. n：在总线周期中插入的等待时钟数目
当插入可编程等待时，采样时序改变。

3. 在上述说明中的值是当占空比为 1:1 的时钟从 X1 输入时的值。

总线保持（CLKOUT 异步）



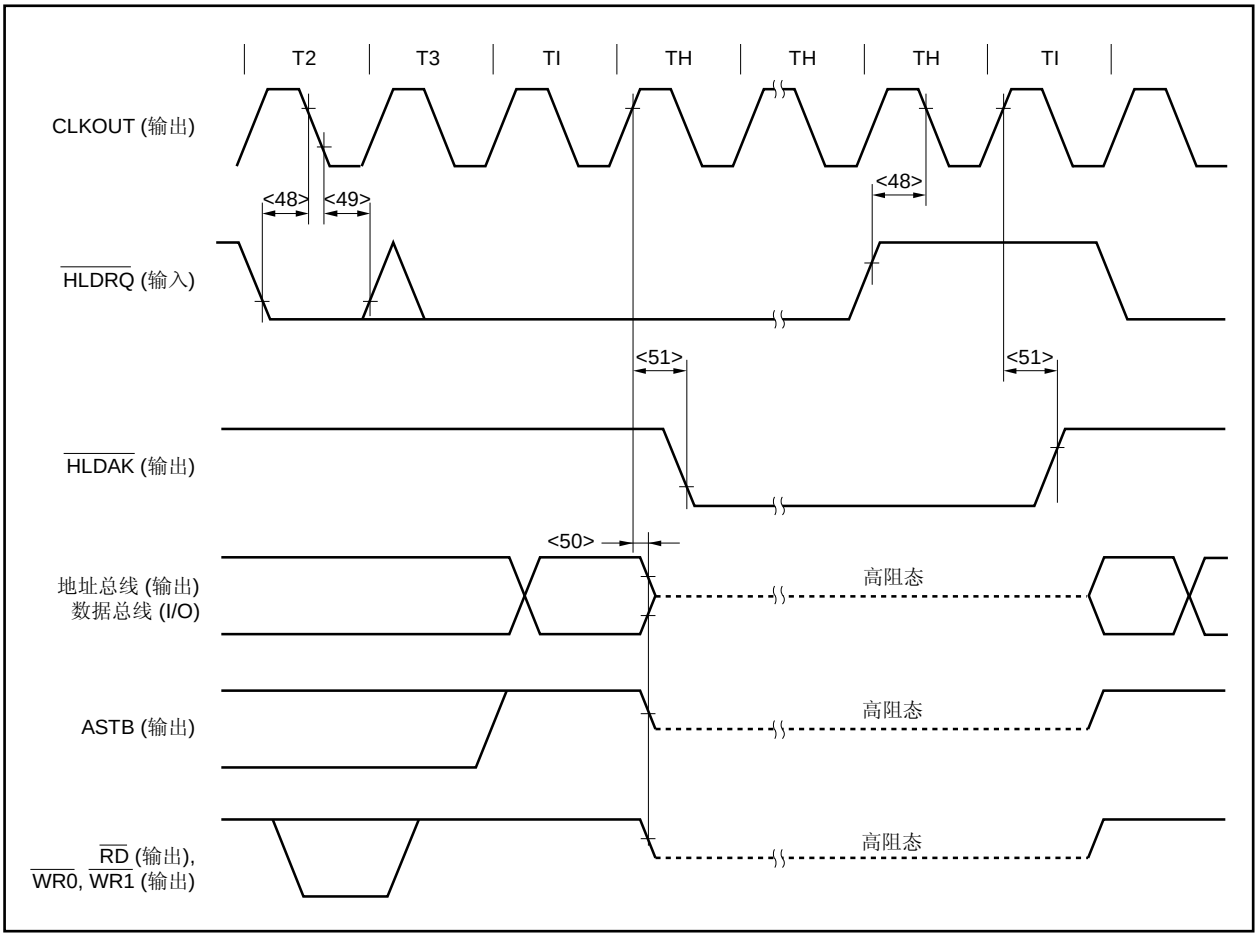
（b） CLKOUT 同步

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.7 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号		条件	最小值	最大值	单位
$\overline{\text{HLDRQ}}$ 建立时间 （至 CLKOUT↓）	t_{SHQK}	<48>		20		ns
$\overline{\text{HLDRQ}}$ 保持时间 （自 CLKOUT↓）	t_{HKHQ}	<49>		5		ns
自 CLKOUT↑至总线悬空的延时	t_{DKF}	<50>			19	ns
自 CLKOUT↑至 $\overline{\text{HLDAK}}$ 有效时的延时	t_{DKHA}	<51>			19	ns

备注 **1.** 在上述说明中的值是当占空比为 1:1 的时钟从 X1 输入时的值。
2. 关于 CLKOUT 输出时序的详情，参见 **33.7.2 CLKOUT 输出时序**。

总线保持（CLKOUT 同步）



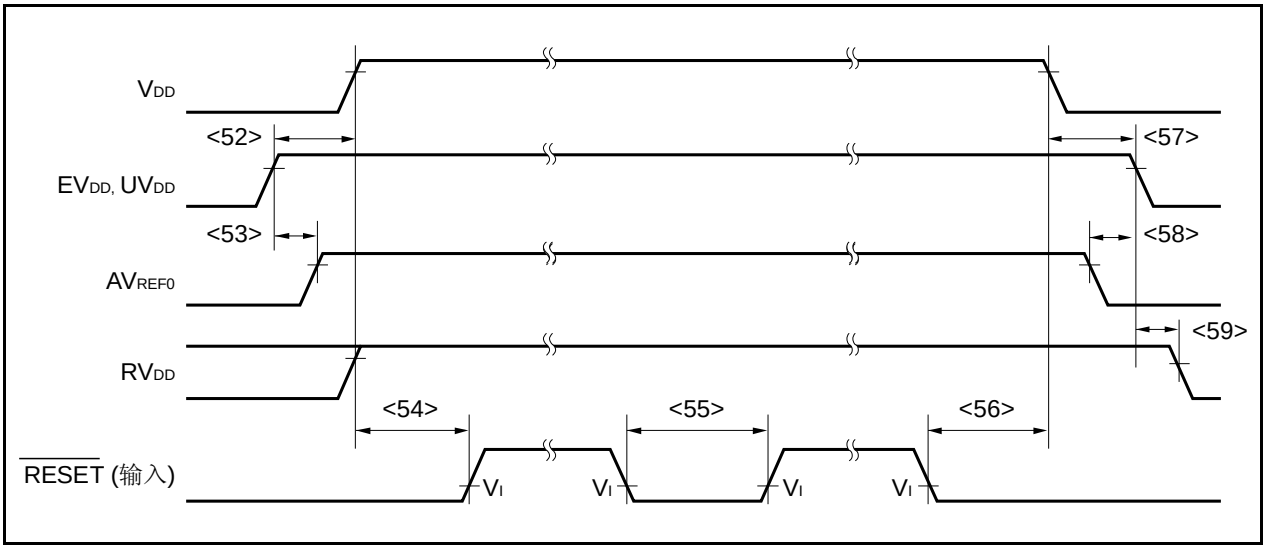
33.7.4 电源开启电源/关闭/复位时序

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = RV_{DD} = 2.0 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号		条件	最小值	最大值	单位
EV_{DD}↑ → V_{DD}↑	t_{REL}	<52>		0		ns
EV_{DD}↑ → AV_{REF0}, AV_{REF1}↑	t_{REA}	<53>		0	t_{REL}	ns
V_{DD}↑ → $\overline{RESET}$↑	t_{RER}	<54>		500 + t_{REG}注		ns
$\overline{RESET}$ 低电平宽度	t_{WRSL}	<55>		500		ns
$\overline{RESET}$↓ . V_{DD}↓	t_{FRE}	<56>		500		ns
V_{DD}↓ EV_{DD}↓	t_{FEL}	<57>		0		ns
AV_{REF0}↓ EV_{DD}↓	t_{FEA}	<58>		0	t_{FEL}	ns
EV_{DD}↓ RV_{DD}↓	t_{FERV}	<59>		0		ns

注 参见 **33.5 稳压器特性**。

备注 **1.** $\overline{RESET}$ 引脚具有模拟噪声消除功能。
2. V_{DD} 和 RV_{DD} 同时上电。



33.8 外设功能特性

33.8.1 中断时序

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.0 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号	条件	最小值	最大值	单位
NMI 高电平宽度	t_{WNIH}		500		ns
NMI 低电平宽度	t_{WNIL}		500		ns
INTPn[注1] 高电平宽度	t_{WITH}	n = 0 至 7（模拟噪声消除）	500		ns
		n = 3（数字噪声消除）	$3T_{SMP}$ + 20		ns
INTPn[注1] 低电平宽度	t_{WITL}	n = 0 至 7（模拟噪声消除）	500		ns
		n = 3（数字噪声消除）	$3T_{SMP}$ + 20		ns

注 1. INTPn 的特性与 $\overline{DRST}$ 引脚（P05/INTP2/$\overline{DRST}$）的特性相同。
2. 仅限 flash 存储器编程模式。

备注 **1.** T_{SMP}：噪声消除采样时钟周期。
2. NMI 和 INTPn 引脚具有模拟噪声消除功能 (n = 0 至 7)。

33.8.2 按键返回时序

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.0 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号	条件	最小值	最大值	单位
KRn 高电平宽度	t_{WKRH}		500		ns
KRn 低电平宽度	t_{WKRL}		500		ns

备注 **1.** n = 0 至 7。
2. KRn 引脚具有模拟噪声消除功能。

33.8.3 定时器时序

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.0 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号	条件	最小值	最大值	单位
TI 高电平宽度	t_{TIH}	TIP00, TIP20, TIP21, TIP30, TIP31, TIP40, TIP41, TIP50, TIP51, TIQ00至TIQ03	2T + 20		ns
TI 低电平宽度	t_{TIL}		2T + 20		ns

备注 T = 1/f_{XX}

33.8.4 UART时序

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号	条件	最小值	最大值	单位
发送速率		V_{DD} = 2.0至3.6 V		625	kbps
ASCK0频率		V_{DD} = 2.0至3.6 V		2.5	MHz
		V_{DD} = 2.2至3.6 V		5	MHz
		V_{DD} = 2.7至3.6 V		10	MHz

33.8.5 CSIB时序

(1) 主模式
(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号		条件	最小值	最大值	单位
$\overline{SCKBn}$ 周期时间	t_{KCY1}	<60>	2.7 V ≤ V_{DD} ≤ 3.6 V	125		ns
			2.0 V ≤ V_{DD} < 2.7 V	800		ns
$\overline{SCKBn}$ 高电平宽度	t_{KH1}	<61>	2.7 V ≤ V_{DD} ≤ 3.6 V	t_{KCY1}/2 – 8		ns
			2.0 V ≤ V_{DD} < 2.7 V	t_{KCY1}/2 – 80		ns
$\overline{SCKBn}$ 低电平宽度	t_{KL1}	<62>	2.7 V ≤ V_{DD} ≤ 3.6 V	t_{KCY1}/2 – 8		ns
			2.0 V ≤ V_{DD} < 2.7 V	t_{KCY1}/2 – 80		ns
SIBn建立时间 (到$\overline{SCKBn}$↑)	t_{SIK1}	<63>	2.7 V ≤ V_{DD} ≤ 3.6 V	27		ns
			2.0 V ≤ V_{DD} < 2.7 V	100		ns
SIBn保持时间(从$\overline{SCKBn}$↑)	t_{KSI1}	<64>	2.7 V ≤ V_{DD} ≤ 3.6 V	27		ns
			2.0 V ≤ V_{DD} < 2.7 V	100		ns
从$\overline{SCKBn}$↓ 到SOBn输出的延迟时间	t_{KSO1}	<65>	2.7 V ≤ V_{DD} ≤ 3.6 V		27	ns
			2.0 V ≤ V_{DD} < 2.7 V		95	ns

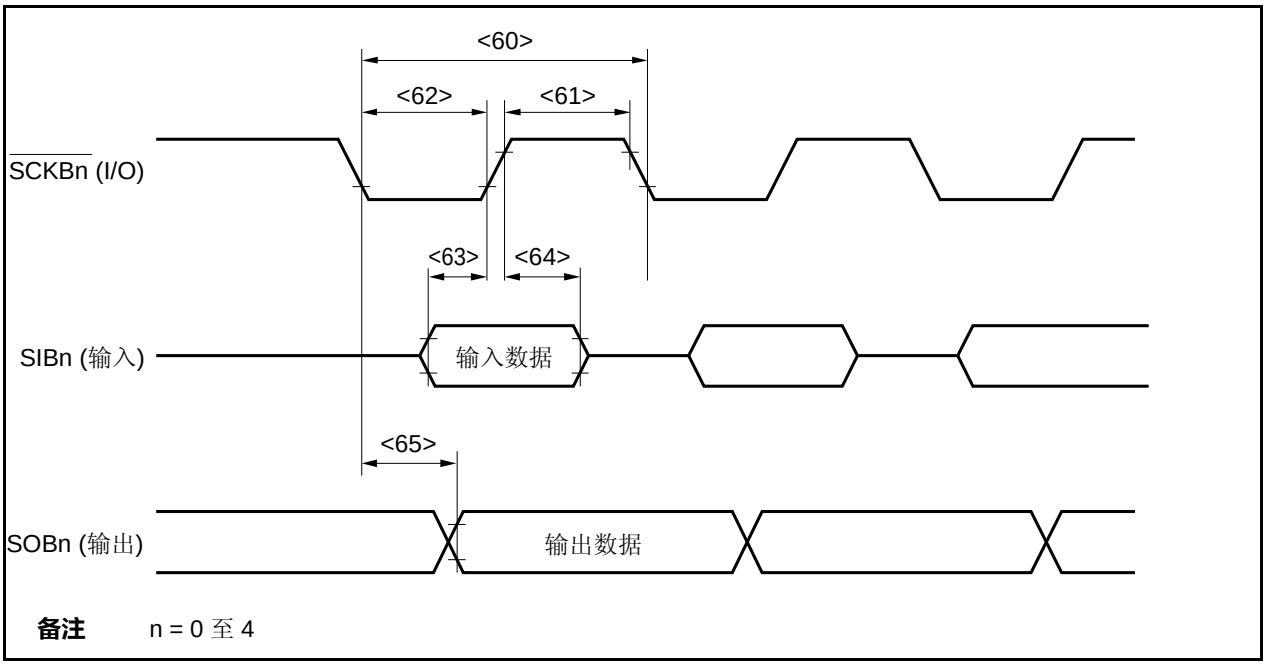
备注 n = 0 至 4

(2) 从模式

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号		条件	最小值	最大值	单位
$\overline{\text{SCKBn}}$ 周期时间	t_{KCY2}	<60>	2.7 V ≤ V_{DD} ≤ 3.6 V	125		ns
			2.0 V ≤ V_{DD} < 2.7 V	800		ns
$\overline{\text{SCKBn}}$ 高电平宽度	t_{KH2}	<61>	2.0 V ≤ V_{DD} ≤ 3.6 V	54.5		ns
$\overline{\text{SCKBn}}$ 低电平宽度	t_{KL2}	<62>	2.0 V ≤ V_{DD} ≤ 3.6 V	54.5		ns
SIBn建立时间 (到$\overline{\text{SCKBn}}$↑)	t_{SIK2}	<63>	2.7 V ≤ V_{DD} ≤ 3.6 V	27		ns
			2.0 V ≤ V_{DD} < 2.7 V	100		ns
SIBn保持时间(从$\overline{\text{SCKBn}}$↑)	t_{KSI2}	<64>	2.7 V ≤ V_{DD} ≤ 3.6 V	27		ns
			2.0 V ≤ V_{DD} < 2.7 V	100		ns
从$\overline{\text{SCKBn}}$↓ 到SOBn输出的延迟时间	t_{KSO2}	<65>	2.7 V ≤ V_{DD} ≤ 3.6 V		27	ns
			2.0 V ≤ V_{DD} < 2.7 V		95	ns

备注 n = 0 至 4



33.8.6 I²C 总线模式

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.0 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

参数		符号		普通模式		高速模式		单位
				最小值	最大值	最小值	最大值	
SCL0n 时钟频率		f_{CLK}		0	100	0	400	kHz
总线空闲时间 （开始和停止状态之间）		t_{BUF}	<66>	4.7	–	1.3	–	*us*
保持时间[注1]		$t_{HD:STA}$	<67>	4.0	–	0.6	–	*us*
SCL0n 时钟低电平宽度		t_{LOW}	<68>	4.7	–	1.3	–	*us*
SCL0n 时钟高电平宽度		t_{HIGH}	<69>	4.0	–	0.6	–	*us*
开始/重新开始状态的建立时间		$t_{SU:STA}$	<7>	4.7	–	0.6	–	*us*
数据保持时间	CBUS 兼容主机	$t_{HD:DAT}$	<71>	5.0	–	–	–	*us*
	I²C 模式			0[注2]	–	0[注2]	0.9[注3]	*us*
数据建立时间		$t_{SU:DAT}$	<72>	250	–	100[注4]	–	ns
SDA0n 和SCL0n信号上升时间		t_R	<73>	–	1000	20 + 0.1Cb[注5]	300	ns
SDA0n 和SCL0n信号下降时间		t_F	<74>	–	300	20 + 0.1Cb[注5]	300	ns
停止状态建立时间		$t_{SU:STO}$	<75>	4.0	–	0.6	–	*us*
由输入滤波器抑制的毛刺脉冲宽度		t_{SP}	<76>	–	–	0	50	ns
每个信号总线的电容负载		Cb		–	400	–	400	pF

注 **1.** 在开始条件时，第一个时钟脉冲在保持时间后产生。

2. 系统内部 SDA0n（在 SCL0n 信号的 $V_{IHmin.}$时）信号需要最短 300 ns 的保持时间，以占用 SCL0n 信号下降沿时的不确定区域。

3. 如果系统不延长 SCL0n 信号低电平保持时间 （t_{LOW}），则只需要满足最大数据保持时间（$t_{HD:DAT}$）。

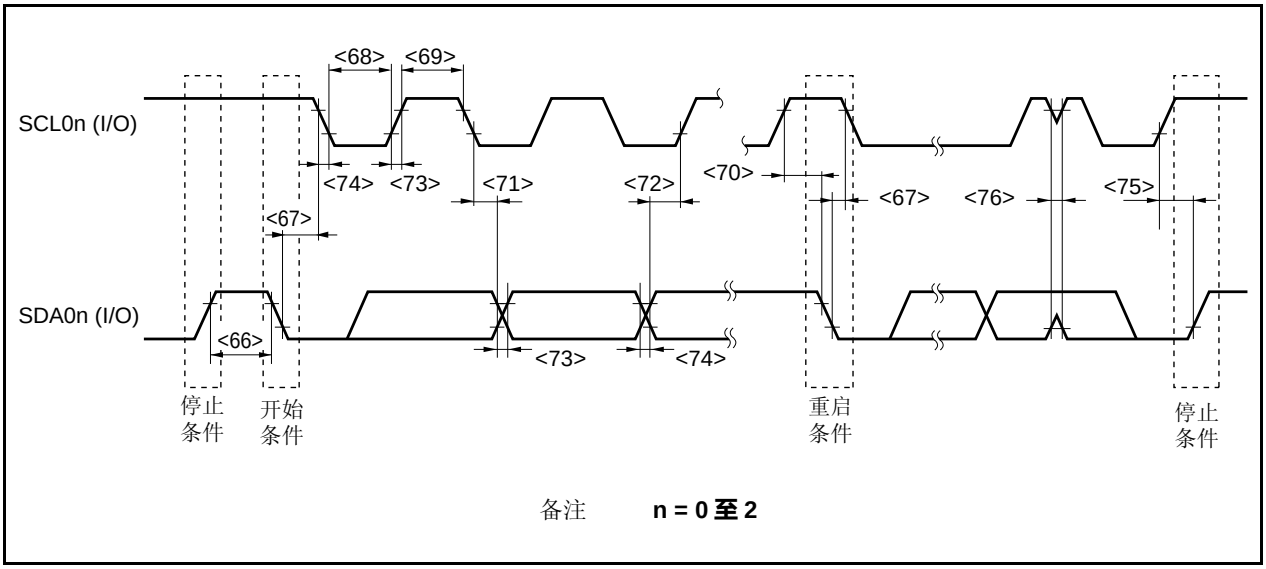
4. 在正常模式 I²C 总线系统中可以适用高速模式 I²C 总线。这种情况下，设置高速模式 I²C 总线使其满足以下条件。

- 如果系统不延长 SCL0n 信号低电平保持时间：
  $t_{SU:DAT} \geq 250$ ns
- 如果系统延长 SCL0n 信号低电平保持时间：
  发送后继的数据位到 SDA0n 信号线优先于 SCL0n 信号线释放（$t_{Rmax.}$+ $t_{SU:DAT}$ = 1,000 + 250 = 1,250 ns：正常 I²C 总线模式规范）。

5. Cb：每条总线的总电容（单位：pF）。

备注 n = 0 至 2

I²C 总线时序



33.8.7 A/D转换器

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}, 2.7 V ≤ AV_{REF0} = AV_{REF1} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
分辨率					10	位
总体误差注		2.7 V ≤ AV_{REF0} ≤ 3.6 V			±0.6	%FSR
A/D转换时间	t_{CONV}	3.0 V ≤ AV_{REF0} ≤ 3.6 V	2.6		24	*us*
		2.7 V ≤ AV_{REF0} ≤ 3.0 V	3.9		24	*us*
零标称误差					±0.5	%FSR
满量程误差					±0.5	%FSR
非线性误差					±4.0	LSB
微分线性误差					±4.0	LSB
模拟输入电压	V_{IAN}		AV_{SS}		AV_{REF0}	V
参考电压	AV_{REF0}		2.7		3.6	V
AV_{REF0}电流	AI_{REF0}	正常转换模式下		3	6.5	mA
		高速转换模式下		4	10	mA
		A/D转换器未使用时			5	*u*A

注 量化误差除外（±0.05% FSR）。

注意事项 在 A/D 转换期间，不要设置（读/写）复用功能端口，否则转换分辨率可能会下降。

备注 LSB：最低有效位
FSR：满量程范围

33.8.8 D/A转换器

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

参数	符号	条件	最小值	典型值	最大值	单位
分辨率					8	位
总误差		R = 2 MΩ			±1.2	%FSR
设置时间		C = 20 pF			3	*u*s
寄存器输出	R_O	输出数据 55H		6.42		kΩ
参考电压	AV_{REF1}		2.7		3.6	V
AV_{REF1} 电流[注]	AI_{REF1}	D/A 转换工作		1	2.5	mA
		D/A 转换停止			5	*u*A

注 D/A 转换器 1 通道的值

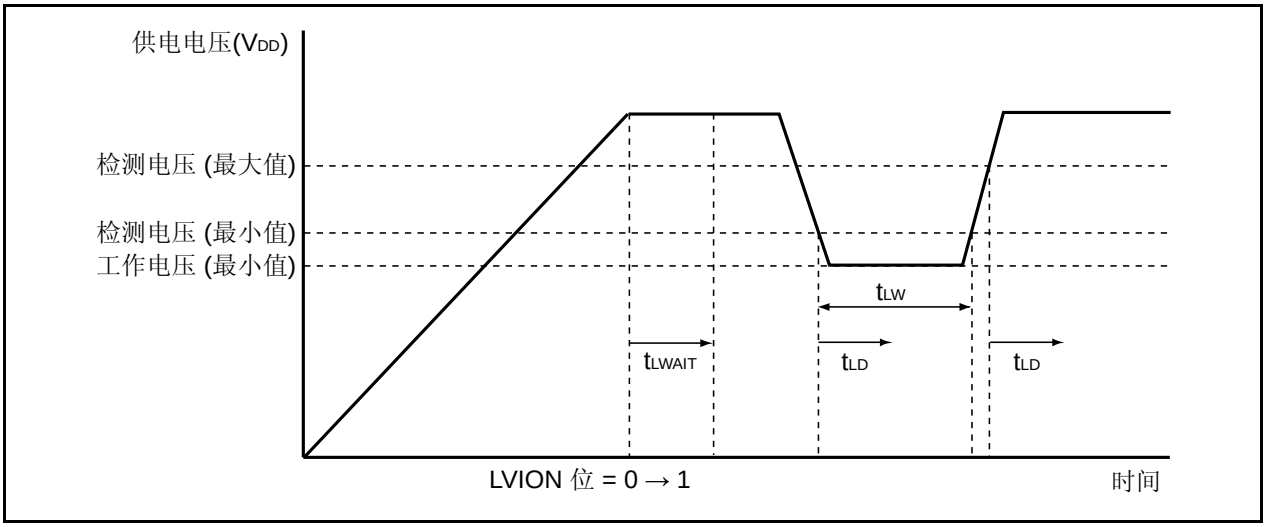
备注 R 是输出引脚负载电阻，C 是输出引脚负载电容。

33.8.9 LVI电路特性

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.0 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V_{LVI0}		2.7	2.8	2.9	V
	V_{LVI1}		2.2	2.3	2.4	V
	V_{LVI2}		2.0	2.1	2.2	V
响应时间[注]	t_{LD}	上升沿： V_{DD} 达到V_{LVI0}/V_{LVI1}/V_{LVI2}/V_{LVI3} (最大值)之后 下降沿： V_{DD} 已经下降落到 V_{LVI0}/V_{LVI1}/V_{LVI2}/V_{LVI3} (最小值)		0.2	2.0	ms
最小脉冲宽度	t_{LW}	V_{DD} = V_{LVI0}/V_{LVI1}/V_{LVI2}/V_{LVI3} (MIN.)	0.2			ms
参考电压稳定等待时间	t_{LWAIT}	V_{DD} 达到V_{LVI0}/V_{LVI1}/V_{LVI2}/V_{LVI3} (最大值)之后		0.1	0.2	ms

注 检测到检测电压和输出一个中断或复位信号所需的时间。



33.8.10 RTC备份模式特性

(1) V_{DD} 下电时序

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.0 V 至 3.6 V, RV_{DD} = 1.8 V 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0V, C_L = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
V_{DD} 负转换速率	V_{DDNSR1}	使用RTC备份模式，并将LVI检测电平设置为2.80 ± 0.10 V			0.2	V/ms
	V_{DDNSR2}	使用RTC备份模式，并将LVI检测电平设置为2.30 ± 0.10 V			0.07	V/ms

(2) V_{DD} 上电时序

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.0 V 至 3.6 V, RV_{DD} = 1.8 V 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0V, C_L = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
V_{DD} 正转换速率	RV_{DDPSR}		3.0			V/s

(3) RTC 备份区域的稳压器输出电压 (VCH)

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.0 V 至 3.6 V, RV_{DD} = 2.0 V 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0V, C_L = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
RTC备份区域的稳压器输出电压 (VCH)	VCH		0.8		1.8	V

(4) VCH 建立时间

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.0 V 至 3.6 V, RV_{DD} = 2.0 V 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0V, C_L = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
VCH 建立时间	t_{SPOR}	从RV_{DD} 达到最大幅度(V_{DD} = 2.0至3.6 V) 直到VCH 稳定			4.5	ms

33.9 Flash存储器编程特性

(1) 基本特性

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.7 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

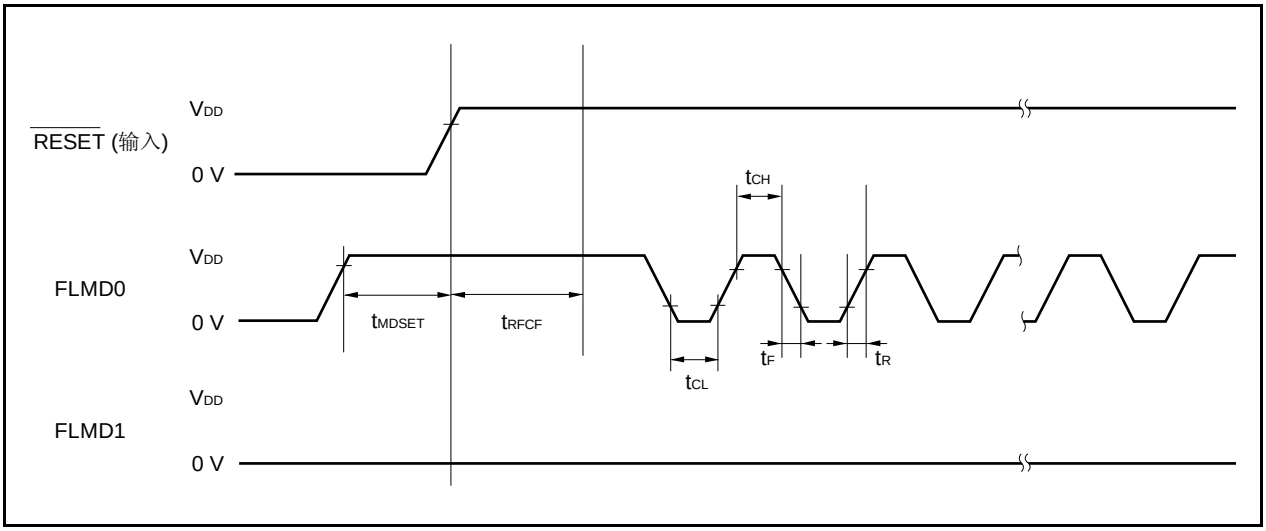
参数	符号	条件		最小值	典型值	最大值	单位
工作频率	f_{CPU}			2.5		20	MHz
供电电压	V_{DD}	2.5 MHz ≤ f_{XX} ≤ 20 MHz		2.7		3.6	V
重写次数	C_{WRT}	用于更新程序 使用flash存储器编程器和瑞萨电子自编程库时	保持15年	1,000			次
		用于更新数据 使用瑞萨电子EEPROM模拟库 (可用的ROM空间： 3个连续块的12 KB)	保持5年	10,000			次
编程温度	t_{PRG}			–40		+85	°C

(2) 串行写入操作特性

(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.7 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
FLMD0， FLMD1建立时间	t_{MDSET}		2		3000	ms
自 $\overline{RESET}$↑ 到FLMD0 计数启动时间	t_{RFCF}	f_X = 2.5 至 10 MHz	800			*us*
FLMD0 计数器 高电平宽度/ 低电平宽度	t_{CH}/t_{CL}		10		100	*us*
FLMD0 计数器上升时间/下降时间	t_R/t_F				1	*us*

Flash 写入模式设置时序



(3) 编程特性

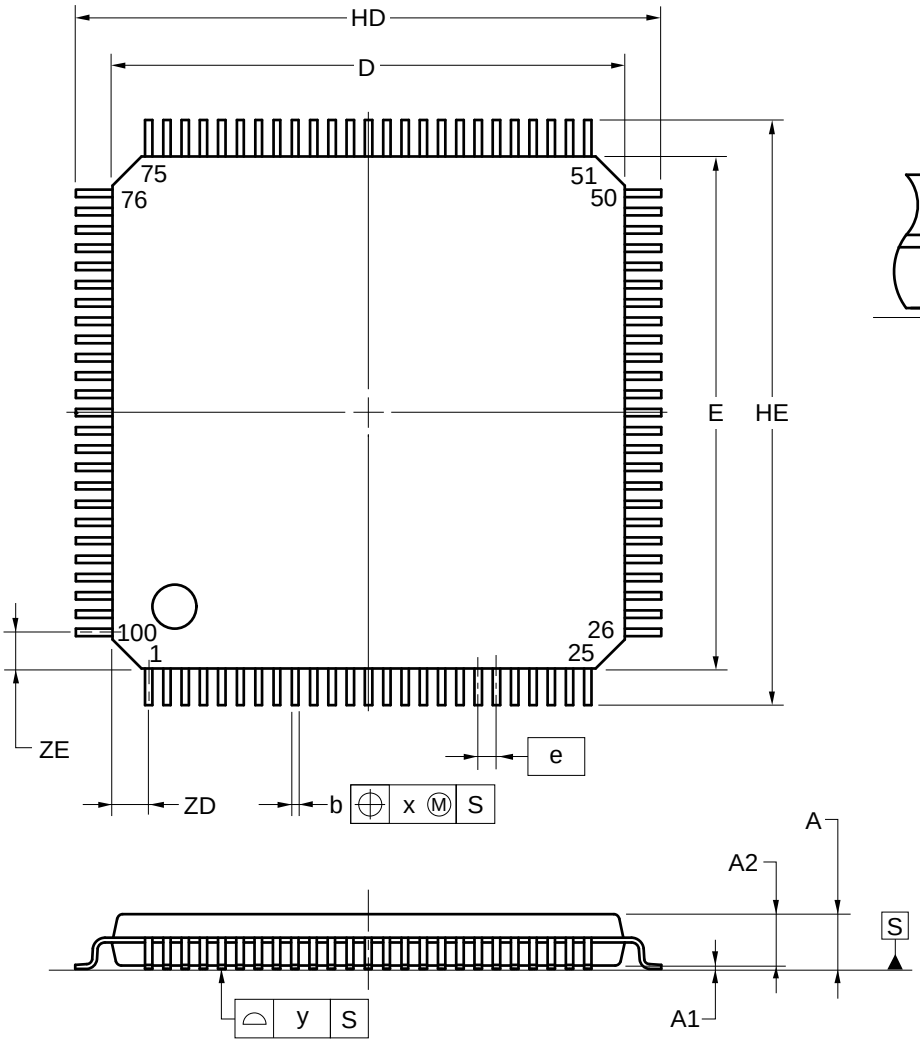
(T_A = –40 至+85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1} = 2.7 至 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

参数	符号	条件	最小值	典型值	最大值	单位
芯片擦除时间		f_{XX} = 20 MHz（执行片擦除命令时）		105		ms
每 256 字节的写入时间		f_{XX} = 20 MHz		2.0		ms
块内部验证时间		f_{XX} = 20 MHz		10		ms
块空白检查时间		f_{XX} = 20 MHz		0.5		ms
Flash 存储器信息设置时间		f_{XX} = 20 MHz		30		ms

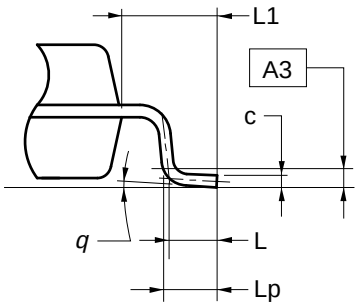
备注 块容量 = 4 KB

第三十四章　封装图

100针塑封LQFP (密间距) (14x14)



引脚端的细节图



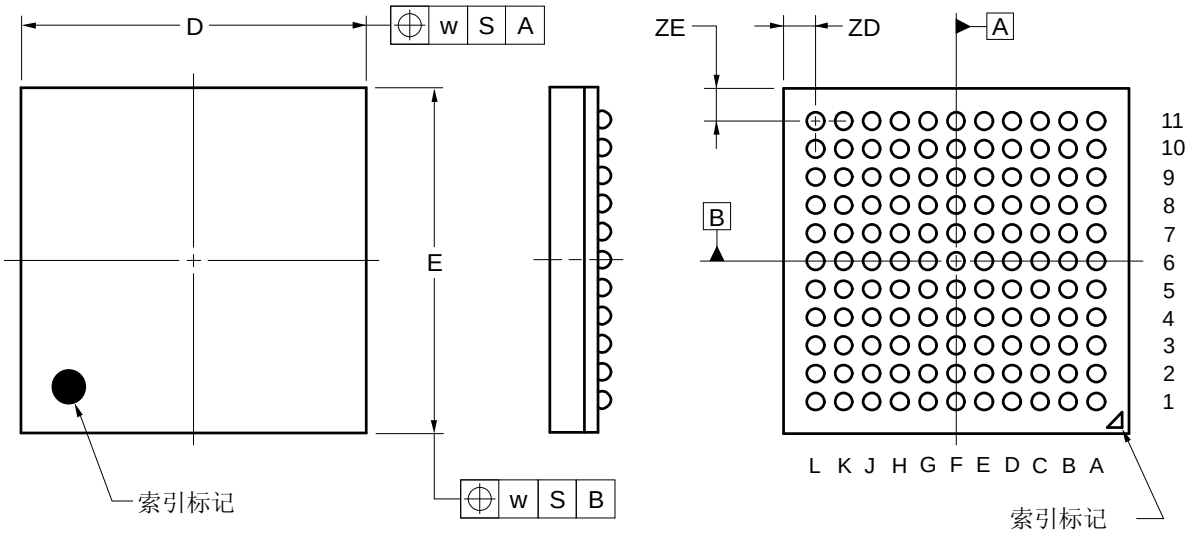
(单位:mm)

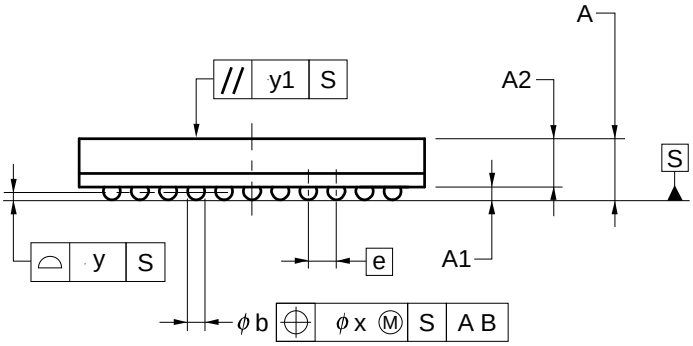
项目	尺寸
D	14.00±0.20
E	14.00±0.20
HD	16.00±0.20
HE	16.00±0.20
A	1.60 MAX
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
q	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.00
ZE	1.00

P100GC-50-UEU-1

瑞萨电子公司 2010

121针塑封 FBGA (8x8)





(单位 :mm)

项目	尺寸
D	8.00±0.10
E	8.00±0.10
w	0.20
A	1.21±0.10
A1	0.30±0.05
A2	0.91
e	0.65
b	0.40±0.05
x	0.08
y	0.10
y1	0.20
ZD	0.75
ZE	0.75

P121F1-65-CAH

瑞萨电子公司 2010

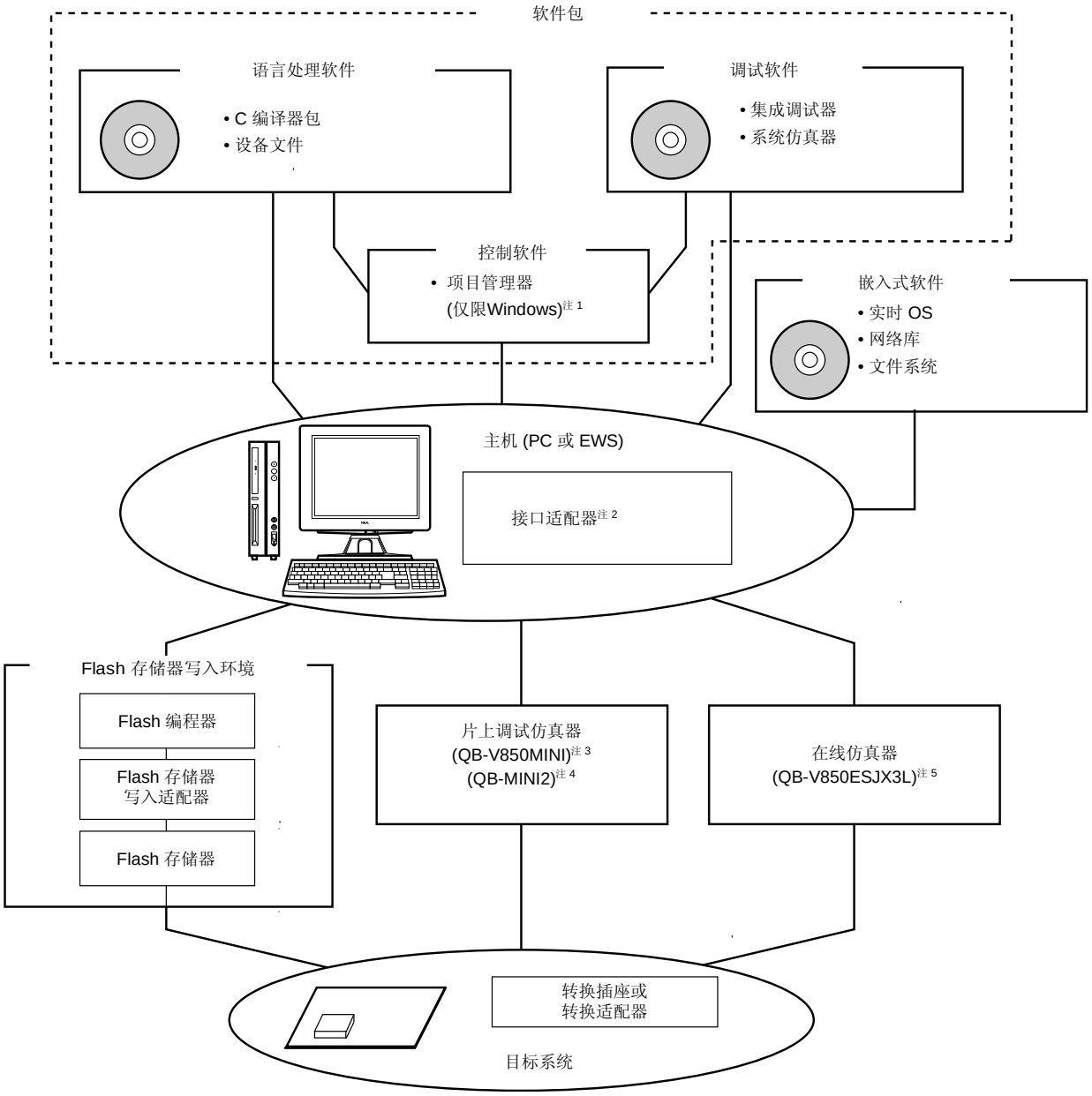
附录 A 开发工具

以下开发工具可用来开发使用 V850ES/JG3-L 单片机的系统。
图 A-1 显示了开发工具的配置。

- **支持 PC98-NX 系列**
  除非另行说明，IBM PC/AT™ 支持兼容的产品兼容于 PC98-NX 系列计算机。当使用 PC98-NX 系列计算机时，可参见 IBM PC/AT 兼容机说明。

- **Windows®**
  除非另行说明，“Windows”指以下操作系统。
  - Windows 98, 2000
  - Windows Me
  - Windows XP
  - Windows NT® Ver. 4.0

图 A-1. 开发工具配置



注

1. 项目管理器 PM+ 包含在 C 编译包内。
   PM+ 只能在 Windows 下使用。
2. QB-V850MINI、QB-MINI2 和 QB-V850ESJX3L（开发中）仅支持 USB 接口。
3. QB-V850MINI 提供 ID850QB、USB 接口电缆、OCD 电缆、自检板、KEL 适配器和 KEL 连接器。所有其它部件均为选购。
4. QB-MINI2 提供 USB 接口电缆、16-引脚目标电缆、10-引脚目标电缆和 78K0-OCD 板（不提供集成调试器。）所有其它部件均为选购。
5. QB-V850ESJX3L 提供 ID850QB、flash 编程器（MINICUBE2）、供电单元和 USB 接口适配器。所有其它部件均为选购。

A.1 软件包

SP850 V850 微控器的软件包	本软件包包括用于 V850 微控制器的通用开发工具（软件）。
	型号： μS××××SP850

备注 ××××型号随使用的主机和操作系统的不同而异。

μS××××SP850

××××	主机	操作系统	提供形式
AB17	PC-9800 系列 IBM PC/AT 兼容机	Windows （日文版）	CD-ROM
BB17		Windows （英文版）	

A.2 语言处理软件

CA850 注 C 编译包	本编译器将用 C 语言写的程序转换为微控制器执行的目标代码。本编译器由项目管理器 PM+内启动。
	型号：μS××××CA703000
DF703738 注 设备文件	本文件包含器件的特有信息。 该设备文件应和开发工具（CA850 或 ID850QB）配合使用。 相应的主机和操作系统随使用的工具不同而不同。

备注 ××××型号随使用的主机和操作系统的不同而异。

μS××××CA703000

××××	主机	操作系统	提供形式
AB17	PC-9800 系列； IBM PC/AT 兼容机	Windows （日文版）	CD-ROM
BB17		Windows （英文版）	
3K17	SPARC 工作站 ™	SunOS™ （Rel. 4.1.4）， Solaris™ （Rel. 2.5.1）	

A.3 控制软件

PM+注 项目管理器	这是一个控制软件，设计使得用户可以在 Windows 环境下有效地开发用户程序。所有用户开发过程中的操作，诸如启动编辑器、构建以及进行调试等，都可以在 PM+中进行。 **<注意事项>** PM+ 包含在 C 编译包 CA850 之中。 它仅能在 Windows 下使用。

A.4 调试工具（硬件）

A.4.1 使用IECUBE QB-V850ESJX3L时

当将 QB-V850ESJX3L 连接至主机（PC-9821 系列，PC/AT 兼容机）时的系统配置如下所示。即使不备有可选部件，照样可以完成连接。

图 A-2. 系统配置（使用 QB-V850ESJX3L 时）（1/2）

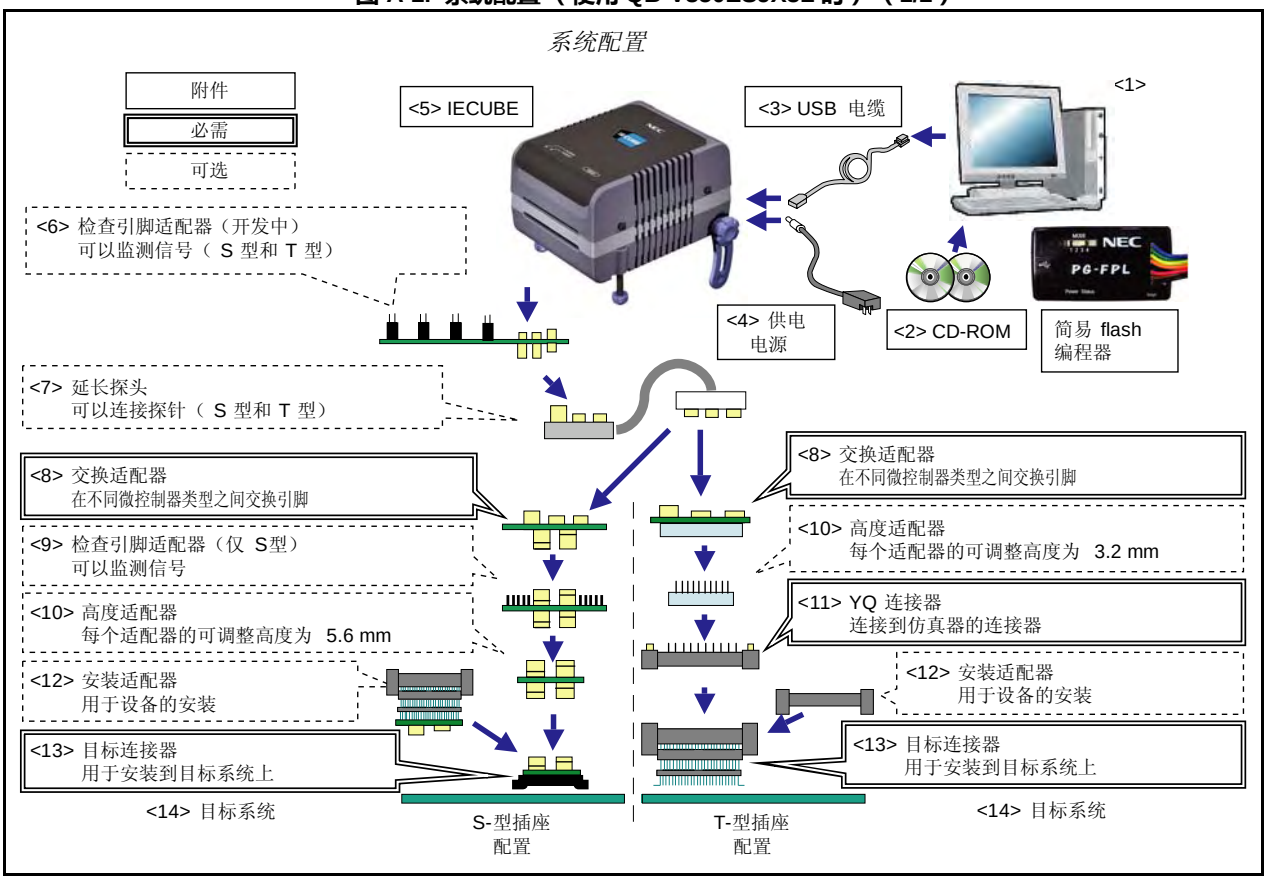


图 A-2. 系统配置（使用 QB-V850ESJX3L 时）（2/2）

<1> 主机（PC-9821 系列，IBM-PC/AT 可兼容）
<2> 调试器，USB 驱动，手册，等（ID850QB 光盘，附件光盘注1）
<3> USB 接口电缆
<4> AC 适配器
<5> 在线仿真器 (QB-V850ESJX3L 注5)
<6> 检查引脚适配器 (S 和 T 类型) (QB-144-CA-01 注2) (选购)
<7> 延长探头 (S 和 T 类型) (QB-144-EP-01S) (选购)
<8> 交换适配器注3 (S 类型：QB-100GC-EA-01S (GC 封装)，T 类型：QB-100GC-EA-01T (GC 封装)
<9> 检查引脚适配器注4 (仅限 S 类型) (QB-100-CA-01S) (选购)
<10> 高度适配器注4 (S 类型: QB-100-SA-01S (GC 封装)，T 类型：QB-100GC-YS-01T (GC 封装)
<11> YQ 连接器注3 (仅限 T 类型) (QB-100GC-YQ-01T) (GC 封装)
<12> 安装适配器 (S 类型：QB-100GC-MA-01S (GC 封装)，T 类型：QB-100GF-HQ-01T (GC 封装) (选购)
<13> 目标连接器注3 (S 类型：QB-100GC-TC-01S (GC 封装)，T 类型：QB-100GC-NQ-01T (GC 封装)
<14> 目标系统

注

1. 从瑞萨电子网站下载设备文件。
   http://www2.renesas.com/micro/en/ods/index.html
2. 开发中
3. 提供的设备根据订购的编号而不同。
   - 订购 QB-V850ESJX3L-ZZZ 注5 时，
     不提供交换适配器和目标连接器。
   - 订购 QB-V850ESJX3L-S100GC 注5 时，
     提供 QB-100GC-EA-01S 和 QB-100GC-TC-01S。
   - 订购 QB-V850ESJX3L-T100GC 注5 时，
     提供 QB-100GC-EA-01T, QB-100GC-YQ-01T 和 QB-100GC-NQ-01T。
4. <9> 和<10>都使用时，不必考虑 <9> 和<10> 之间的次序。
5. 开发中

<5> QB-V850ESJX3L 注1,2 在线仿真器	在使用 V850ES/JG3-L 进行应用系统的开发时，在线仿真器作为调试硬件和软件。它支持集成调试器 ID850QB，该仿真器应该和供电电源与仿真器探头组合使用。使用 USB 接口电缆将该仿真器连接到主机。
<3> USB 接口电缆	用于连接 QB-V850ESJX3L 注2.和主机的电缆。
<4> AC 适配器	通过更换 AC 插头，可以支持 100 至 240 V。
<8> QB-100GC-EA-01S QB-100GC-EA-01T 交换适配器	该适配器用于进行引脚转换。 • QB-100GC-EA-01S：100 针塑封 LQFP (GC-UEU 类型) • QB-100GC-EA-01T：100 针塑封 LQFP (GC-UEU 类型)
<9> QB-100-CA-01S (仅限 S 类型) 检查引脚适配器	该适配器用于使用示波器等进行波形监控。 • QB-100-CA-01S：GC-UEU/GF-GAS 类型
<10> QB-100-SA-01S QB-100GC-YS-01T 高度适配器	该适配器用于调整高度。 • QB-100GF-SA-01S: GC-UEU/GF-GAS 类型 • QB-100GC-YS-01T：100 针塑封 LQFP (GC-UEU 类型)
<11> QB-100GC-YQ-01T YQ 连接器	转换适配器以连接目标连接器和交换适配器。 • QB-100GC-YQ-01T：100 针塑封 LQFP (GC-UEU 类型)
<12> QB-100GC-MA-01S QB-100GC-HQ-01T 安装适配器	该适配器用于安装 V850ES/JG3-L 的插座。 • QB-100GC-MA-01S：100 针塑封 LQFP (GC-UEU 类型) • QB-100GC-HQ-01T：100 针塑封 LQFP (GC-UEU 类型)
<13> QB-100GC-TC-01S QB-100GC-NQ-01T 目标连接器	焊接到目标系统的连接器。 • QB-100GC-TC-01S：100 针塑封 LQFP (GC-UEU 类型) • QB-100GC-NQ-01T：100 针塑封 LQFP (GC-UEU 类型)

注 **1.** QB-V850ESJX3L 注2 提供一个供电电源单元，USB 接口电缆和 flash 存储器编程器(MINICUBE2)。也会提供集成调试器 ID850QB 作为控制软件。

2. 开发中

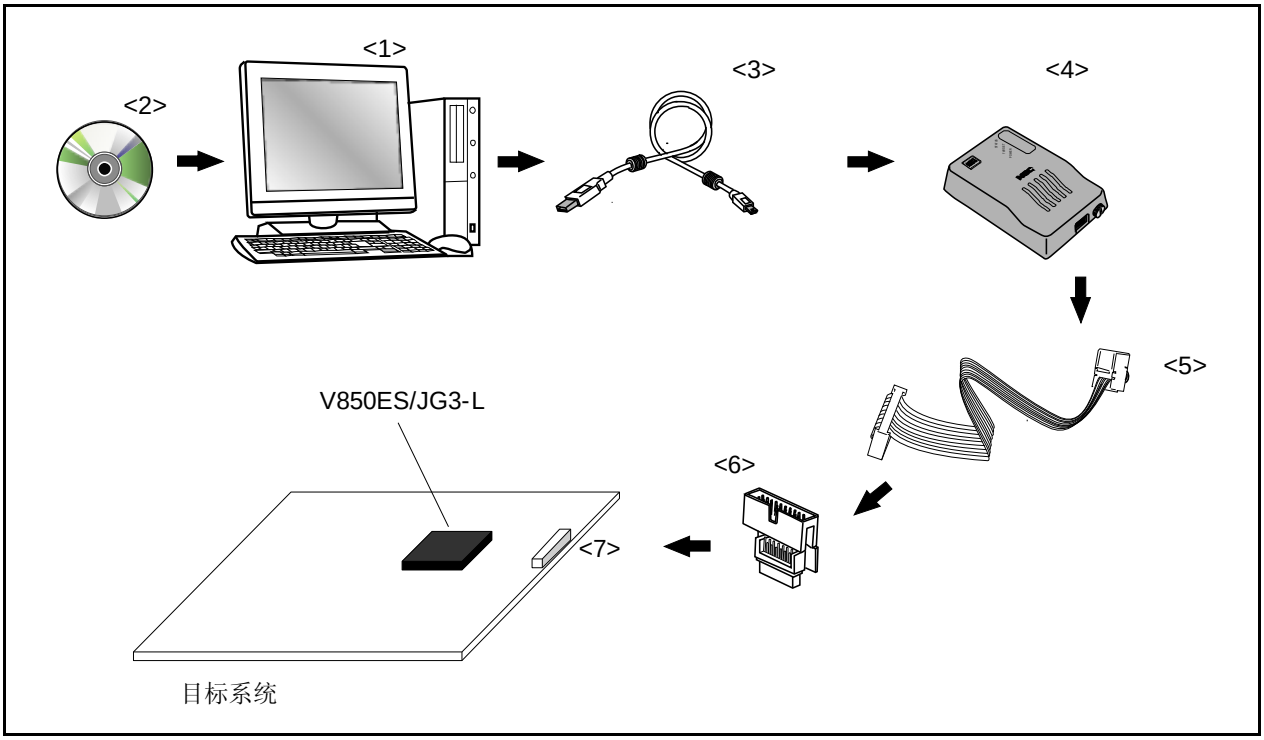
备注 尖括号中的数字对应于图 A-2 中的数字。

A.4.2 使用MINICUBE QB-V850MINI时

（1） 使用 MINICUBE 在线仿真

将 MINICUBE 连接到主机（PC-9821 系列，PC/AT 兼容机）时的系统配置如下所示。

图 A-3. 在线仿真系统配置



<1> 主机	配有 USB 接口的 PC。
<2> CD-ROM 注1	CD-ROM 内有诸如集成调试器 ID850QB，N-Wire 检查器，设备驱动和文档。它和 MINICUBE 一起提供。
<3> USB 接口电缆	连接主机和 MINICUBE 的 USB 电缆。它和 MINICUBE 一并提供。电缆长度大约为 2m。
<4> MINICUBE 在线仿真器	当开发使用 V850ES/JG3-L 的应用系统时，使用该在线调试仿真器调试其硬件和软件，它支持集成调试器 ID850QB。
<5> OCD 电缆	连接 MINICUBE 和目标系统的电缆。 它和 MINICUBE 一并提供。电缆长度大约为 20 cm。
<6> 连接器转换板 KEL 适配器	该转换板与 MINICUBE 一并提供。
<7> MINICUBE 连接器 KEL 连接器注2	8830E-026-170S （与 MINICUBE 一并提供） 8830E-026-170L （单独售卖）

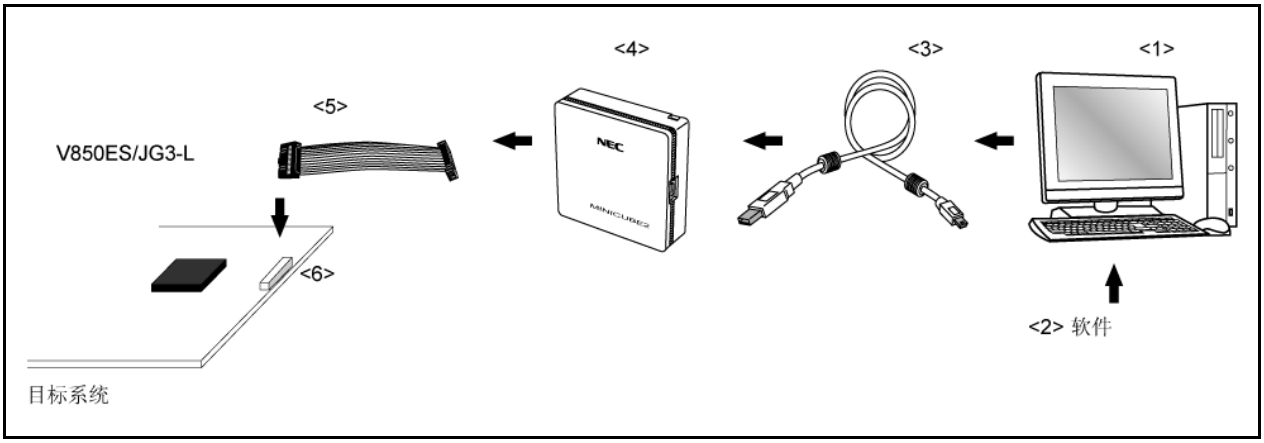
注 **1.** 从瑞萨电子网站上下载设备文件。
http://www2.renesas.com/micro/en/ods/index.html
2. KEL 公司的产品

备注 尖括号中的数字对应于图 A-3 中的数字。

A.4.3 使用MINICUBE2 QB-MINI2 时

将 MINICUBE2 连接至主机（PC-9821 系列，PC/AT 兼容机）时的系统配置如下所示。

图 A-4. 在线仿真系统配置



<1> 主机	配有 USB 端口的 PC。
<2> 软件	集成调试器 ID850QB，设备文件等等。从瑞萨电子网站上下载设备文件。 http://www2.renesas.com/micro/en/ods/index.html
<3> USB 接口电缆	连接主机和 MINICUBE 的 USB 电缆。它与 I MINICUBE 一并提供。电缆长度大约为 2m。
<4> MINICUBE2 在线仿真器	当开发使用 V850ES/JG3-L 的应用系统时，使用该在线调试仿真器调试其硬件和软件，它支持集成调试器 ID850QB 注。
<5> 16-引脚目标电缆	连接 MINICUBE2 和目标系统的电缆。 它与 MINICUBE 一并提供。电缆长度大约为 15 cm。
<6> 目标连接器 （单独售卖）	使用一个 2.54 mm 间距的 16-引脚通用连接器。

备注 尖括号中的数字对应于图 A-4 中的数字。

A.5 调试工具（软件）

ID850QB 集成调试器	本调试器支持 V850 微控制器的在线仿真器。ID850QB 是基于 Windows 操作系统的软件。 它具有改进的 C-兼容调试功能，并且，使用一个与源文件相关的集成窗口来显示源程序追踪执行的结果，反汇编显示、带有追踪结果的存储显示。 它应与设备文件组合使用。
	产品编号： μS×××× ID703000-QB （ID850QB）

备注 ××××型号随所用的主机和操作系统不同而异。

μS××××ID703000-QB

××××	主机	操作系统	提供形式
AB17	PC-9800 系列， IBM PC/AT 兼容机	Windows （日文版）	CD-ROM
BB17		Windows （英文版）	

A.6 嵌入式软件

RX850, RX850 Pro 实时操作系统	RX850 和 RX850 Pro 为实时操作系统，遵从μITRON 3.0 规范。 提供一个工具 （配置器），用来产生多重信息表。 RX850 Pro 比 RX850 具有更多功能。
	型号: μS××××RX703000-ΔΔΔΔ （RX850） μS××××RX703100-ΔΔΔΔ （RX850 Pro）
Applilet(（开发中）	这是一个驱动配置器，可以为 V850ES/JH3-EV850ES/JG3-L 自动产生样例程序
RX-FS850 （文件系统）	这是一个 FAT 文件系统功能。 该文件系统支持 CD-ROM 文件系统功能。 该文件系统被实时操作系统 RX850 Pro 使用。

注意事项 欲购买 RX850 或 RX850 Pro 实时操作系统，首先填写购买申请表并签署授权许可文件。

备注 型号中的 ×××× 和 随使用的主机和操作系统的不同而异。

μS××××RX703000-ΔΔΔΔ
μS××××RX703100-ΔΔΔΔ

ΔΔΔΔ	产品概要	用于批量生产的最大数目
001	评估产品	不要用于批量生产
100K	批量生产产品	0.1 百万单位
001M		1 百万单位
010M		10 百万单位
S01	源程序	为批量生产的源程序

××××	主机	操作系统	提供形式
AB17	PC-9800 系列， IBM PC/AT 兼容机	Windows （日文版）	CD-ROM
BB17		Windows （英文版）	
3K17	SPARCstation	Solaris （Rel. 2.5.1）	

A.7 Flash 存储器写入工具

Flashpro V（产品编号：PG-FP5 注） Flash 存储器编程器	Flash 存储器编程器专用于具备片上 Flash 存储器的微控制器编程。
QB-MINI2（MINICUBE2）	具有编程功能的片上调试仿真器。
FA-100GC-UEU-B FA-121F1-CAH-B Flash 存储器写入适配器	Flash 存储器写入适配器，用来连接 Flashpro IV，Flashpro V 等（未连线）。 • FA-100GC-UEU-B：100 针塑封 LQFP (GC-UEU 类型) • FA-121F1-GAH-B：121 针塑封 FBGA (F1-CAH 类型)
FA-70F3796GC-UEU-RX Flash 存储器写入适配器	Flash 存储器写入适配器，用来连接 Flashpro IV，Flashpro V 等（已连线）。 • FA-70F3796GC-UEU-RX：100 针塑封 LQFP

备注 FA-100GC-UEU-B, FA-70F3796F1-CAH-RX 是 Naito Densei Machida Mfg. Co., Ltd.的产品。
TEL: +81-42-750-4172

附录 B 产品之间的主要区别

表 B-1. 主要区别 (1/2)

主要区别		V850ES/JG3-L		
		*u*PD70F3737 *u*PD70F3738	*u*PD70F3792 *μ*PD70F3793	*μ*PD70F3794 *μ*PD70F3795 *μ*PD70F3796
引脚	端口 (5 V 耐压)	84 (31)	83 (31)	80 (28)
存储器	内部 flash 存储器	128/256 KB	384/512 KB	256/384/512 KB
	内部 RAM	8/16 KB	32/40 KB	40 KB
供电电压	V_{DD}, EV_{DD}	2.2 至 3.6 V @5 MHz 2.7 至 3.6 V @20 MHz	2.2 至 3.6 V @5 MHz 2.7 至 3.6 V @20 MHz 2.0 至 3.6 V @2.5 MHz	2.2 V @5 MHz 2.7 V @20 MHz 2.0 V @5 MHz 3.0 至 3.6 V @16 MHz (USB 操作)
	UV_{DD}	无		2.2 V @5 MHz 2.7 V @20 MHz 2.0 V @2.5 MHz 3.0 至 3.6 V @16 MHz (USB 操作)
	RV_{DD}	无	可供使用	
	A/D, D/A 操作电压	2.7 至 3.6 V		
低电压检测器 (LVI)	LVI	2 个级别：2.8 V (TYP.), 2.3 V (TYP.) 通过软件可选	3 个级别：2.8 V (TYP.), 2.3 V (TYP.) , 2.1 V (TYP.) 通过软件可选	
	低电压检测处的中断条件	供电电压下降或上升越过检测电压时		
待机功能	RTC 备份模式	无	可供使用	
Flash 存储器	引导区域	32 KB		
	块的配置	块 0 至 63/127 块： 每个块 2 KB	块 0 至 95/127 块： 每个块 4 KB	块 0 至 63/95/127 块： 每个块 4 KB
RTC		无	可供使用	
UARTA		3 通道	6 通道	
UARTC		无	可供使用	
USB 功能		无	无	可供使用

表 B-1. 主要区别 (2/2)

主要区别			μPD70F3737 μPD70F3738	μPD70F3792 μPD70F3793	μPD70F3794 μPD70F3795 μPD70F3796
中断请求信号	中断数量		57 (外部中断：9)	64 (外部中断：9)	
	RTC	INTRTC0	无	可供使用	
		INTRCT1	无	可供使用	
		INTRTC2	无	可供使用	
	UARTA3	INTTUA3R	无	可供使用	
		INTTUA3T	无	可供使用	
	UARTA4	INTTUA4R	无	可供使用	
		INTTUA4T	无	可供使用	
	UARTA5	INTTUA5R	无	可供使用	
		INTTUA5T	无	可供使用	
	UARTC1	INTTUC1R	无	可供使用	
		INTTUC1T	无	可供使用	
DMA 启动因素	INTRTC1		无	可供使用	
	INTUA3R		无	可供使用	
	INTUA3T		无	可供使用	
	INTUA4R		无	可供使用	
	INTUA4T		无	可供使用	
	INTUA5R		无	可供使用	
	INTUA5T		无	可供使用	
LVI 检测电平			2 个级别	3 个级别	
封装	100 针 LQFP (14x20)		可供使用	无	
操作供电电压	2.0 至 3.6 V@2.5 MHz		无	可供使用	

表 B-2. LQFP 封装中的引脚布局

引脚编号	μPD70F3737GC-UEU-AX μPD70F3738GC-UEU-AX	μPD70F3792GC-UEU-AX μPD70F3793GC-UEU-AX	μPD70F3794GC-UEU-AX μPD70F3795GC-UEU-AX μPD70F3796GC-UEU-AX
7脚	PDH5/A21	P02/NMI/A21	
17 脚	P02/NMI	RV_{DD}	
18 脚	P03/INTP0/ADTRG	P03/INTP0/ADTRG/RTC1HZ	P03/INTP0/ADTRG/UCLK/RTC1HZ
19 脚	P04/INTP1	P04/INTP1/RTCDIV/RTCCL	
28 脚	P33/TIP01/TOP01		UDMF
29 脚	P34/TIP10/TOP10		UPDF
30 脚	P35/TIP11/TOP11		UV_{DD}
31 脚	P36	P36/TXDA3	
32 脚	P37	P37/RXDA3	
45 脚	P92/A2/TIP41/TOP41	P92/A2/TIP41/TOP41/TXDA4	
46 脚	P93/A3/TIP40/TOP40	P93/A3/TIP40/TOP40/RXDA4	
47 脚	P94/A4/TIP31/TOP31	P94/A4/TIP31/TOP31/TXDA5	
48 脚	P95/A5/TIP30/TOP30	P95/A5/TIP30/TOP30/RXDA5	
49 脚	P96/A6/TIP21/TOP21	P96/A6/TXDC0/TIP21/TOP21	
50 脚	P97/A7/SIB1/TIP20/TOP20	P97/A7/SIB1/RXDC0/TIP20/TOP20	

表 B-4. FBGA 封装中的引脚布局

引脚编号	*u* PD70F3737F1-CAH-A *u* PD70F3738F1-CAH-A	*u* PD70F3792F1-CAH-A *u* PD70F3793F1-CAH-A	*u* PD70F3794F1-CAH-A *u* PD70F3795F1-CAH-A *u* PD70F3796F1-CAH-A
D2 脚	V_{DD}	RV_{DD}	
G3 脚	P03/INTP0/ADTRG	P03/INTP0/ADTRG/RTC1HZ	P03/INTP0/ADTRG/UCLK/RTC1HZ
G4 脚	PDH5/A21	P02/NMI/A21	
H4 脚	P04/INTP1	P04/INTP1/RTCDIV/RTCCL	
H5 脚	P36	P36/TXDA3	
J2 脚	P02/NMI	P02/NMI/A21	IC
J5 脚	P35/TIP11/TOP11		EV_{SS}
J6 脚	P37	P37/RXDA3	
J9 脚	P93/A3/TIP40/TOP40	P93/A3/TIP40/TOP40/RXDA4	
J11 脚	P97/A7/SIB1/TIP20/TOP20	P97/A7/SIB1/RXDC0/TIP20/TOP20	
K4 脚	P33/TIP01/TOP01		UDMF
K5 脚	P34/TIP10/TOP10		UDPF
K6 脚	EV_{DD}		UV_{DD}
K9 脚	P92/A2/TIP41/TOP41	P92/A2/TIP41/TOP41/TXDA4	
K10 脚	P95/A5/TIP30/TOP30	P95/A5/TIP30/TOP30/RXDA5	
K11 脚	P96/A6/TIP21/TOP21	P96/A6/TXDC0/TIP21/TOP21	
L10 脚	P94/A4/TIP31/TOP31	P94/A4/TIP31/TOP31/TXDA5	

附录 C 寄存器索引

(1/19)

符号	名称	单元	页码
ADA0CR0	A/D 转换结果寄存器 0	ADC	483
ADA0CR0H	A/D 转换结果寄存器 0H	ADC	483
ADA0CR1	A/D 转换结果寄存器 1	ADC	483
ADA0CR1H	A/D 转换结果寄存器 1H	ADC	483
ADA0CR2	A/D 转换结果寄存器 2	ADC	483
ADA0CR2H	A/D 转换结果寄存器 2H	ADC	483
ADA0CR3	A/D 转换结果寄存器 3	ADC	483
ADA0CR3H	A/D 转换结果寄存器 3H	ADC	483
ADA0CR4	A/D 转换结果寄存器 4	ADC	483
ADA0CR4H	A/D 转换结果寄存器 4H	ADC	483
ADA0CR5	A/D 转换结果寄存器 5	ADC	483
ADA0CR5H	A/D 转换结果寄存器 5H	ADC	483
ADA0CR6	A/D 转换结果寄存器 6	ADC	483
ADA0CR6H	A/D 转换结果寄存器 6H	ADC	483
ADA0CR7	A/D 转换结果寄存器 7	ADC	483
ADA0CR7H	A/D 转换结果寄存器 7H	ADC	483
ADA0CR8	A/D 转换结果寄存器 8	ADC	483
ADA0CR8H	A/D 转换结果寄存器 8H	ADC	483
ADA0CR9	A/D 转换结果寄存器 9	ADC	483
ADA0CR9H	A/D 转换结果寄存器 9H	ADC	483
ADA0CR10	A/D 转换结果寄存器 10	ADC	483
ADA0CR10H	A/D 转换结果寄存器 10H	ADC	483
ADA0CR11	A/D 转换结果寄存器 11	ADC	483
ADA0CR11H	A/D 转换结果寄存器 11H	ADC	483
ADA0M0	A/D 转换器模式寄存器 0	ADC	476
ADA0M1	A/D 转换器模式寄存器 1	ADC	478
ADA0M2	A/D 转换器模式寄存器 2	ADC	481
ADA0PFM	掉电比较模式寄存器	ADC	485
ADA0PFT	掉电比较阈值寄存器	ADC	486
ADA0S	模拟输入通道指定寄存器	ADC	482
ADIC	中断控制寄存器	INTC	958
AWC	地址等待控制寄存器	BCU	175
BCC	总线周期控制寄存器	BCU	176
BSC	总线宽度配置寄存器	BCU	164
CB0CTL0	CSIB0 控制寄存器 0	CSIB	604
CB0CTL1	CSIB0 控制寄存器 1	CSIB	607
CB0CTL2	CSIB0 控制寄存器 2	CSIB	608
CB0RIC	中断控制寄存器	INTC	958
CB0RX	CSIB0 接收数据寄存器	CSIB	602
CB0RXL	CSIB0 接收数据寄存器 L	CSIB	602
CB0STR	CSIB0 状态寄存器	CSIB	610
CB0TIC	中断控制寄存器	INTC	958

(2/19)

符号	名称	单元	页码
CB0TX	CSIB0 发送数据寄存器	CSIB	603
CB0TXL	CSIB0 发送数据寄存器 L	CSIB	603
CB1CTL0	CSIB1 控制寄存器 0	CSIB	604
CB1CTL1	CSIB1 控制寄存器 1	CSIB	607
CB1CTL2	CSIB1 控制寄存器 2	CSIB	608
CB1RIC	中断控制寄存器	INTC	958
CB1RX	CSIB1 接收数据寄存器	CSIB	602
CB1RXL	CSIB1 接收数据寄存器 L	CSIB	602
CB1STR	CSIB1 状态寄存器	CSIB	610
CB1TIC	中断控制寄存器	INTC	958
CB1TX	CSIB1 发送数据寄存器	CSIB	603
CB1TXL	CSIB1 发送数据寄存器 L	CSIB	603
CB2CTL0	CSIB2 控制寄存器 0	CSIB	604
CB2CTL1	CSIB2 控制寄存器 1	CSIB	607
CB2CTL2	CSIB2 控制寄存器 2	CSIB	608
CB2RIC	中断控制寄存器	INTC	958
CB2RX	CSIB2 接收数据寄存器	CSIB	602
CB2RXL	CSIB2 接收数据寄存器 L	CSIB	602
CB2STR	CSIB2 状态寄存器	CSIB	610
CB2TIC	中断控制寄存器	INTC	958
CB2TX	CSIB2 发送数据寄存器	CSIB	603
CB2TXL	CSIB2 发送数据寄存器 L	CSIB	603
CB3CTL0	CSIB3 控制寄存器 0	CSIB	604
CB3CTL1	CSIB3 控制寄存器 1	CSIB	607
CB3CTL2	CSIB3 控制寄存器 2	CSIB	608
CB3RIC	中断控制寄存器	INTC	958
CB3RX	CSIB3 接收数据寄存器	CSIB	602
CB3RXL	CSIB3 接收数据寄存器 L	CSIB	602
CB3STR	CSIB3 状态寄存器	CSIB	610
CB3TIC	中断控制寄存器	INTC	958
CB3TX	CSIB3 发送数据寄存器	CSIB	603
CB3TXL	CSIB3 发送数据寄存器 L	CSIB	603
CB4CTL0	CSIB4 控制寄存器 0	CSIB	604
CB4CTL1	CSIB4 控制寄存器 1	CSIB	607
CB4CTL2	CSIB4 控制寄存器 2	CSIB	608
CB4RIC	中断控制寄存器	INTC	958
CB4RX	CSIB4 接收数据寄存器	CSIB	602
CB4RXL	CSIB4 接收数据寄存器 L	CSIB	602
CB4STR	CSIB4 状态寄存器	CSIB	610
CB4TIC	中断控制寄存器	INTC	958
CB4TX	CSIB4 发送数据寄存器	CSIB	603
CB4TXL	CSIB4 发送数据寄存器 L	CSIB	603
CCLS	CPU 操作时钟状态寄存器	CG	191
CKC	时钟控制寄存器	CG	196
CKTHSEL	时钟直通选择寄存器	CG	192

(3/19)

符号	名称	单元	页码
CLM	时钟监控模式寄存器	CLM	1036
CRCD	CRC 数据寄存器	CRC	1046
CRCIN	CRC 数据寄存器	CRC	1046
CTBP	CALLT 基准指针	CPU	38
CTPC	CALLT 执行状态保存寄存器	CPU	37
CTPSW	CALLT 执行状态保存寄存器	CPU	37
DA0CS0	D/A 转换值设置寄存器 0	DAC	516
DA0CS1	D/A 转换值设置寄存器 1	DAC	516
DA0M	D/A 转换器模式寄存器	DAC	515
DADC0	DMA 寻址控制寄存器 0	DMAC	922
DADC1	DMA 寻址控制寄存器 1	DMAC	922
DADC2	DMA 寻址控制寄存器 2	DMAC	922
DADC3	DMA 寻址控制寄存器 3	DMAC	922
DBC0	DMA 传输计数寄存器 0	DMAC	921
DBC1	DMA 传输计数寄存器 1	DMAC	921
DBC2	DMA 传输计数寄存器 2	DMAC	921
DBC3	DMA 传输计数寄存器 3	DMAC	921
DBPC	异常/调试陷阱状态保存寄存器	CPU	38
DBPSW	异常/调试陷阱状态保存寄存器	CPU	38
DCHC0	DMA 通道控制寄存器 0	DMAC	923
DCHC1	DMA 通道控制寄存器 1	DMAC	923
DCHC2	DMA 通道控制寄存器 2	DMAC	923
DCHC3	DMA 通道控制寄存器 3	DMAC	923
DDA0H	DMA 目的地址寄存器 0H	DMAC	920
DDA0L	DMA 目的地址寄存器 0L	DMAC	920
DDA1H	DMA 目的地址寄存器 1H	DMAC	920
DDA1L	DMA 目的地址寄存器 1L	DMAC	920
DDA2H	DMA 目的地址寄存器 2H	DMAC	920
DDA2L	DMA 目的地址寄存器 2L	DMAC	920
DDA3H	DMA 目的地址寄存器 3H	DMAC	920
DDA3L	DMA 目的地址寄存器 3L	DMAC	920
DMAIC0	中断控制寄存器	INTC	958
DMAIC1	中断控制寄存器	INTC	958
DMAIC2	中断控制寄存器	INTC	958
DMAIC3	中断控制寄存器	INTC	958
DSA0H	DMA 源地址寄存器 0H	DMAC	919
DSA0L	DMA 源地址寄存器 0L	DMAC	919
DSA1H	DMA 源地址寄存器 1H	DMAC	919
DSA1L	DMA 源地址寄存器 1L	DMAC	919
DSA2H	DMA 源地址寄存器 2H	DMAC	919
DSA2L	DMA 源地址寄存器 2L	DMAC	919
DSA3H	DMA 源地址寄存器 3H	DMAC	919
DSA3L	DMA 源地址寄存器 3L	DMAC	919
DTFR0	DMA 触发因素寄存器 0	DMAC	924
DTFR1	DMA 触发因素寄存器 1	DMAC	924

(4/19)

符号	名称	单元	页码
DTFR2	DMA 触发因素寄存器 2	DMAC	924
DTFR3	DMA 触发因素寄存器 3	DMAC	924
DWC0	数据等待控制寄存器 0	BCU	172
ECR	中断源寄存器	CPU	35
EIPC	中断状态保存寄存器	CPU	34
EIPSW	中断状态保存寄存器	CPU	34
FEPC	NMI 状态保存寄存器	CPU	35
FEPSW	NMI 状态保存寄存器	CPU	35
IIC0	IIC 移位寄存器 0	I^2C	671
IIC1	IIC 移位寄存器 1	I^2C	671
IIC2	IIC 移位寄存器 2	I^2C	671
IICC0	IIC 控制寄存器 0	I^2C	657
IICC1	IIC 控制寄存器 1	I^2C	657
IICC2	IIC 控制寄存器 2	I^2C	657
IICCL0	IIC 时钟选择寄存器 0	I^2C	667
IICCL1	IIC 时钟选择寄存器 1	I^2C	667
IICCL2	IIC 时钟选择寄存器 2	I^2C	667
IICF0	IIC 标志寄存器 0	I^2C	665
IICF1	IIC 标志寄存器 1	I^2C	665
IICF2	IIC 标志寄存器 2	I^2C	665
IICIC0	中断控制寄存器	INTC	958
IICIC1	中断控制寄存器	INTC	958
IICIC2	中断控制寄存器	INTC	958
IICS0	IIC 状态寄存器 0	I^2C	662
IICS1	IIC 状态寄存器 1	I^2C	662
IICS2	IIC 状态寄存器 2	I^2C	662
IICX0	IIC 功能扩展寄存器 0	I^2C	668
IICX1	IIC 功能扩展寄存器 1	I^2C	668
IICX2	IIC 功能扩展寄存器 2	I^2C	668
IMR0	中断屏蔽寄存器 0	INTC	960
IMR0H	中断屏蔽寄存器 0H	INTC	960
IMR0L	中断屏蔽寄存器 0L	INTC	960
IMR1	中断屏蔽寄存器 1	INTC	960
IMR1H	中断屏蔽寄存器 1H	INTC	960
IMR1L	中断屏蔽寄存器 1L	INTC	960
IMR2	中断屏蔽寄存器 2	INTC	960
IMR2H	中断屏蔽寄存器 2H	INTC	960
IMR2L	中断屏蔽寄存器 2L	INTC	960
IMR3	中断屏蔽寄存器 3	INTC	960
IMR3H	中断屏蔽寄存器 3H	INTC	960
IMR3L	中断屏蔽寄存器 3L	INTC	960
INTF0	外部中断下降沿说明寄存器 0	INTC	973
INTF3	外部中断下降沿说明寄存器 3	INTC	974
INTF9H	外部中断下降沿说明寄存器 9H	INTC	975

(5/19)

符号	名称	单元	页码
INTR0	外部中断上升沿说明寄存器 0	INTC	973
INTR3	外部中断上升沿说明寄存器 3	INTC	974
INTR9H	外部中断上升沿说明寄存器 9H	INTC	975
ISPR	正在服务的中断优先级寄存器	INTC	962
KRIC	中断控制寄存器	INTC	958
KRM	按键返回模式寄存器	KR	981
LOCKR	锁定寄存器	CG	197
LVIIC	中断控制寄存器	INTC	958
LVIM	低电压检测寄存器	LVI	1041
LVIS	低电压检测电平选择寄存器	LVI	1042
NFC	噪声消除控制寄存器	INTC	958
OCDM	片上调试模式寄存器	DCU	1085
OCKS0	IIC 分频时钟选择寄存器 0	I^2C	671
OCKS1	IIC 分频时钟选择寄存器 1	I^2C	671
OCKS2	时钟选择寄存器	CLC	196
OSTS	振荡稳定时间选择寄存器	Standby	987
P0	端口 0 寄存器	端口	76
P1	端口 1 寄存器	端口	80
P3	端口 3 寄存器	端口	83
P3H	端口 3 寄存器 H	端口	83
P3L	端口 3 寄存器 L	端口	83
P4	端口 4 寄存器	端口	88
P5	端口 5 寄存器	端口	90
P7H	端口 7 寄存器 H	端口	96
P7L	端口 7 寄存器 L	端口	96
P9	端口 9 寄存器	端口	98
P9H	端口 9 寄存器 H	端口	98
P9L	端口 9 寄存器 L	端口	98
PC	程序计数器	CPU	32
PCC	处理器时钟控制寄存器	CG	187
PCM	端口 CM 寄存器	端口	105
PCT	端口 CT 寄存器	端口	107
PDH	端口 DH 寄存器	端口	109
PDL	端口 DL 寄存器	端口	112
PDLH	端口 DL 寄存器 H	端口	112
PDLL	端口 DL 寄存器 L	端口	112
PF0	端口 0 功能寄存器	端口	79
PF3	端口 3 功能寄存器	端口	87
PF3H	端口 3 功能寄存器 H	端口	87
PF3L	端口 3 功能寄存器 L	端口	87
PF4	端口 4 功能寄存器	端口	89
PF5	端口 5 功能寄存器	端口	94
PF9	端口 9 功能寄存器	端口	104
PF9H	端口 9 功能寄存器 H	端口	104
PF9L	端口 9 功能寄存器 L	端口	104
PFC0	端口 0 功能控制寄存器	端口	78

(6/19)

符号	名称	单元	页码
PFC3	端口 3 功能控制寄存器	端口	85
PFC3H	端口 3 功能控制寄存器 H	端口	85
PFC3L	端口 3 功能控制寄存器 L	端口	85
PFC4	端口 4 功能控制寄存器	端口	89
PFC5	端口 5 功能控制寄存器	端口	92
PFC9	端口 9 功能控制寄存器	端口	101
PFC9H	端口 9 功能控制寄存器 H	端口	101
PFC9L	端口 9 功能控制寄存器 L	端口	101
PFCE0	端口 0 功能控制扩展寄存器	端口	78
PFCE3L	端口 3 功能控制扩展寄存器 L	端口	85
PFCE5	端口 5 功能控制扩展寄存器	端口	92
PFCE9	端口 9 功能控制扩展寄存器	端口	101
PFCE9H	端口 9 功能控制扩展寄存器 H	端口	101
PFCE9L	端口 9 功能控制扩展寄存器 L	端口	101
PIC0	中断控制寄存器	INTC	958
PIC1	中断控制寄存器	INTC	958
PIC2	中断控制寄存器	INTC	958
PIC3	中断控制寄存器	INTC	958
PIC4	中断控制寄存器	INTC	958
PIC5	中断控制寄存器	INTC	958
PIC6	中断控制寄存器	INTC	958
PIC7	中断控制寄存器	INTC	958
PLLCTL	PLL 控制寄存器	CG	195
PLLS	PLL 入锁时间指定寄存器	CG	198
PM0	端口 0 模式寄存器	端口	77
PM1	端口 1 模式寄存器	端口	81
PM3	端口 3 模式寄存器	端口	83
PM3H	端口 3 模式寄存器 H	端口	83
PM3L	端口 3 模式寄存器 L	端口	83
PM4	端口 4 模式寄存器	端口	88
PM5	端口 5 模式寄存器	端口	91
PM7H	端口 7 模式寄存器 H	端口	96
PM7L	端口 7 模式寄存器 L	端口	96
PM9	端口 9 模式寄存器	端口	98
PM9H	端口 9 模式寄存器 H	端口	98
PM9L	端口 9 模式寄存器 L	端口	98
PMC0	端口 0 模式控制寄存器	端口	77
PMC3	端口 3 模式控制寄存器	端口	84
PMC3H	端口 3 模式控制寄存器 H	端口	84
PMC3L	端口 3 模式控制寄存器 L	端口	84
PMC4	端口 4 模式控制寄存器	端口	89
PMC5	端口 5 模式控制寄存器	端口	91
PMC9	端口 9 模式控制寄存器	端口	99
PMC9H	端口 9 模式控制寄存器 H	端口	99
PMC9L	端口 9 模式控制寄存器 L	端口	99

(7/19)

符号	名称	单元	页码
PMCCM	端口 CM 模式控制寄存器	端口	106
PMCCT	端口 CT 模式控制寄存器	端口	108
PMCDH	端口 DH 模式控制寄存器	端口	110
PMCDL	端口 DL 模式控制寄存器	端口	113
PMCDLH	端口 DL 模式控制寄存器 H	端口	113
PMCDLL	端口 DL 模式控制寄存器 L	端口	113
PMCM	端口 CM 模式寄存器	端口	105
PMCT	端口 CT 模式寄存器	端口	107
PMDH	端口 DH 模式寄存器	端口	110
PMDL	端口 DL 模式寄存器	端口	112
PMDLH	端口 DL 模式寄存器 H	端口	112
PMDLL	端口 DL 模式寄存器 L	端口	112
PRCMD	命令寄存器	CPU	64
PRSCM0	预分频器比较寄存器 0	RTC	422
PRSCM1	预分频器比较寄存器 1	BRG	647
PRSCM2	预分频器比较寄存器 2	BRG	647
PRSCM3	预分频器比较寄存器 3	BRG	647
PRSM0	预分频器模式寄存器 0	RTC	421
PRSM1	预分频器模式寄存器 1	BRG	646
PRSM2	预分频器模式寄存器 2	BRG	646
PRSM3	预分频器模式寄存器 3	BRG	646
PSC	省电控制寄存器	CG	985
PSMR	省电模式寄存器	CG	986
PSW	程序状态字	CPU	36
r0 至 r31	通用寄存器	CPU	32
RC1ALH	闹钟分钟设置寄存器	RTC	444
RC1ALM	闹钟时间设置寄存器	RTC	444
RC1ALW	闹钟星期设置寄存器	RTC	445
RC1CC0	RTC 控制寄存器 0	RTC	433
RC1CC1	RTC 控制寄存器 1	RTC	433
RC1CC2	RTC 控制寄存器 2	RTC	435
RC1CC3	RTC 控制寄存器 3	RTC	436
RC1DAY	日计数寄存器	RTC	440
RC1HOUR	小时计数寄存器	RTC	438
RC1MIN	分钟计数寄存器	RTC	438
RC1MONTH	月计数寄存器	RTC	442
RC1SEC	秒计数寄存器	RTC	437
RC1SUBC	副计数寄存器	RTC	437
RC1SUBU	时间误差修正寄存器	RTC	443
RC1WEEK	星期计数寄存器	RTC	441
RC1YEAR	年计数寄存器	RTC	442
RCM	内部振荡模式寄存器	CG	191
REGOVL0	稳压器输出电压级别控制寄存器	REGC	989
REGPR	稳压器保护寄存器	REGC	988
RESF	复位源标志寄存器	Reset	1024

(8/19)

符号	名称	单元	页码
RTBH0	实时输出缓冲寄存器 0H	RTP	467
RTBL0	实时输出缓冲寄存器 0L	RTP	467
RTC0IC	中断控制寄存器	INTC	958
RTC1IC	中断控制寄存器	INTC	958
RTC2IC	中断控制寄存器	INTC	958
RTCBUMCTL0	RTC 备份控制寄存器 0	RTC	1016
RTPC0	实时输出 端口控制寄存器 0	RTP	469
RTPM0	实时输出 端口模式寄存器 0	RTP	468
SOSCAMCTL	副时钟低压操作控制寄存器	待机	1017
SVA0	从地址寄存器 0	I^2C	672
SVA1	从地址寄存器 1	I^2C	672
SVA2	从地址寄存器 2	I^2C	672
SYS	系统状态寄存器	CPU	65
TM0CMP0	TMM0 比较寄存器 0	定时器	412
TM0CTL0	TMM0 控制寄存器 0	定时器	411
TM0EQIC0	中断控制寄存器	INTC	958
TP0CCIC0	中断控制寄存器	INTC	958
TP0CCIC1	中断控制寄存器	INTC	958
TP0CCR0	TMP0 捕获/比较寄存器 0	定时器	215
TP0CCR1	TMP0 捕获/比较寄存器 1	定时器	217
TP0CNT	TMP0 计数器读取缓冲寄存器	定时器	219
TP0CTL0	TMP0 控制寄存器 0	定时器	209
TP0CTL1	TMP0 控制寄存器 1	定时器	210
TP0IOC0	TMP0 I/O 控制寄存器 0	定时器	211
TP0IOC1	TMP0 I/O 控制寄存器 1	定时器	212
TP0IOC2	TMP0 I/O 控制寄存器 2	定时器	213
TP0OPT0	TMP0 选项寄存器 0	定时器	214
TP0OVIC	中断控制寄存器	INTC	958
TP1CCIC0	中断控制寄存器	INTC	958
TP1CCIC1	中断控制寄存器	INTC	958
TP1CCR0	TMP1 捕获/比较寄存器 0	定时器	215
TP1CCR1	TMP1 捕获/比较寄存器 1	定时器	217
TP1CNT	TMP1 计数器读取缓冲寄存器	定时器	219
TP1CTL0	TMP1 控制寄存器 0	定时器	209
TP1CTL1	TMP1 控制寄存器 1	定时器	210
TP1IOC0	TMP1 I/O 控制寄存器 0	定时器	211
TP1IOC1	TMP1 I/O 控制寄存器 1	定时器	212
TP1IOC2	TMP1 I/O 控制寄存器 2	定时器	213
TP1OPT0	TMP1 选项寄存器 0	定时器	214
TP1OVIC	中断控制寄存器	INTC	958
TP2CCIC0	中断控制寄存器	INTC	958
TP2CCIC1	中断控制寄存器	INTC	958
TP2CCR0	TMP2 捕获/比较寄存器 0	定时器	215
TP2CCR1	TMP2 捕获/比较寄存器 1	定时器	217

(9/19)

符号	名称	单元	页码
TP2CNT	TMP2 计数器读取缓冲寄存器	定时器	219
TP2CTL0	TMP2 控制寄存器 0	定时器	209
TP2CTL1	TMP2 控制寄存器 1	定时器	210
TP2IOC0	TMP2 I/O 控制寄存器 0	定时器	211
TP2IOC1	TMP2 I/O 控制寄存器 1	定时器	212
TP2IOC2	TMP2 I/O 控制寄存器 2	定时器	213
TP2OPT0	TMP2 选项寄存器 0	定时器	214
TP2OVIC	中断控制寄存器	INTC	958
TP3CCIC0	中断控制寄存器	INTC	958
TP3CCIC1	中断控制寄存器	INTC	958
TP3CCR0	TMP3 捕获/比较寄存器 0	定时器	215
TP3CCR1	TMP3 捕获/比较寄存器 1	定时器	217
TP3CNT	TMP3 计数器读取缓冲寄存器	定时器	219
TP3CTL0	TMP3 控制寄存器 0	定时器	209
TP3CTL1	TMP3 控制寄存器 1	定时器	210
TP3IOC0	TMP3 I/O 控制寄存器 0	定时器	211
TP3IOC1	TMP3 I/O 控制寄存器 1	定时器	212
TP3IOC2	TMP3 I/O 控制寄存器 2	定时器	213
TP3OPT0	TMP3 选项寄存器 0	定时器	214
TP3OVIC	中断控制寄存器	INTC	958
TP4CCIC0	中断控制寄存器	INTC	958
TP4CCIC1	中断控制寄存器	INTC	958
TP4CCR0	TMP4 捕获/比较寄存器 0	定时器	215
TP4CCR1	TMP4 捕获/比较寄存器 1	定时器	217
TP4CNT	TMP4 计数器读取缓冲寄存器	定时器	219
TP4CTL0	TMP4 控制寄存器 0	定时器	209
TP4CTL1	TMP4 控制寄存器 1	定时器	210
TP4IOC0	TMP4 I/O 控制寄存器 0	定时器	211
TP4IOC1	TMP4 I/O 控制寄存器 1	定时器	212
TP4IOC2	TMP4 I/O 控制寄存器 2	定时器	213
TP4OPT0	TMP4 选项寄存器 0	定时器	214
TP4OVIC	中断控制寄存器	INTC	958
TP5CCIC0	中断控制寄存器	INTC	958
TP5CCIC1	中断控制寄存器	INTC	958
TP5CCR0	TMP5 捕获/比较寄存器 0	定时器	215
TP5CCR1	TMP5 捕获/比较寄存器 1	定时器	217
TP5CNT	TMP5 计数器读取缓冲寄存器	定时器	219
TP5CTL0	TMP5 控制寄存器 0	定时器	209
TP5CTL1	TMP5 控制寄存器 1	定时器	210
TP5IOC0	TMP5 I/O 控制寄存器 0	定时器	211
TP5IOC1	TMP5 I/O 控制寄存器 1	定时器	212
TP5IOC2	TMP5 I/O 控制寄存器 2	定时器	213
TP5OPT0	TMP5 选项寄存器 0	定时器	214
TP5OVIC	中断控制寄存器	INTC	958
TQ0CCIC0	中断控制寄存器	INTC	958

(10/19)

符号	名称	单元	页码
TQ0CCIC1	中断控制寄存器	INTC	958
TQ0CCIC2	中断控制寄存器	INTC	958
TQ0CCIC3	中断控制寄存器	INTC	958
TQ0CCR0	TMQ0 捕获/比较寄存器 0	定时器	309
TQ0CCR1	TMQ0 捕获/比较寄存器 1	定时器	311
TQ0CCR2	TMQ0 捕获/比较寄存器 2	定时器	313
TQ0CCR3	TMQ0 捕获/比较寄存器 3	定时器	315
TQ0CNT	TMQ0 计数器读取缓冲寄存器	定时器	316
TQ0CTL0	TMQ0 控制寄存器 0	定时器	303
TQ0CTL1	TMQ0 控制寄存器 1	定时器	304
TQ0IOC0	TMQ0 I/O 控制寄存器 0	定时器	305
TQ0IOC1	TMQ0 I/O 控制寄存器 1	定时器	306
TQ0IOC2	TMQ0 I/O 控制寄存器 2	定时器	307
TQ0OPT0	TMQ0 选项寄存器 0	定时器	308
TQ0OVIC	中断控制寄存器	INTC	958
UA0CTL0	UARTA0 控制寄存器 0	UARTA	526
UA0CTL1	UARTA0 控制寄存器 1	UARTA	550
UA0CTL2	UARTA0 控制寄存器 2	UARTA	551
UA0OPT0	UARTA0 选项控制寄存器 0	UARTA	529
UA0RIC	中断控制寄存器	INTC	958
UA0RX	UARTA0 接收数据寄存器	UARTA	531
UA0STR	UARTA0 状态寄存器	UARTA	529
UA0TIC	中断控制寄存器	INTC	958
UA0TX	UARTA0 发送数据寄存器	UARTA	532
UA1CTL0	UARTA1 控制寄存器 0	UARTA	526
UA1CTL1	UARTA1 控制寄存器 1	UARTA	550
UA1CTL2	UARTA1 控制寄存器 2	UARTA	551
UA1OPT0	UARTA1 选项控制寄存器 0	UARTA	529
UA1RIC	中断控制寄存器	INTC	958
UA1RX	UARTA1 接收数据寄存器	UARTA	531
UA1STR	UARTA1 状态寄存器	UARTA	529
UA1TIC	中断控制寄存器	INTC	958
UA1TX	UARTA1 发送数据寄存器	UARTA	532
UA2CTL0	UARTA2 控制寄存器 0	UARTA	526
UA2CTL1	UARTA2 控制寄存器 1	UARTA	550
UA2CTL2	UARTA2 控制寄存器 2	UARTA	551
UA2OPT0	UARTA2 选项控制寄存器 0	UARTA	529
UA2RIC	中断控制寄存器	INTC	958
UA2RX	UARTA2 接收数据寄存器	UARTA	531
UA2STR	UARTA2 状态寄存器	UARTA	529
UA2TIC	中断控制寄存器	INTC	958
UA2TX	UARTA2 发送数据寄存器	UARTA	532
UA3CTL0	UARTA3 控制寄存器 0	UARTA	526
UA3CTL1	UARTA3 控制寄存器 1	UARTA	550
UA3CTL2	UARTA3 控制寄存器 2	UARTA	551

(11/19)

符号	名称	单元	页码
UA3OPT0	UARTA3 选项控制寄存器 0	UARTA	529
UA3RIC	中断控制寄存器	INTC	958
UA3RX	UARTA3 接收数据寄存器	UARTA	531
UA3STR	UARTA3 状态寄存器	UARTA	529
UA3TIC	中断控制寄存器	INTC	958
UA3TX	UARTA3 发送数据寄存器	UARTA	532
UA4CTL0	UARTA4 控制寄存器 0	UARTA	526
UA4CTL1	UARTA4 控制寄存器 1	UARTA	550
UA4CTL2	UARTA4 控制寄存器 2	UARTA	551
UA4OPT0	UARTA4 选项控制寄存器 0	UARTA	529
UA4RIC	中断控制寄存器	INTC	958
UA4RX	UARTA4 接收数据寄存器	UARTA	531
UA4STR	UARTA4 状态寄存器	UARTA	529
UA4TIC	中断控制寄存器	INTC	958
UA4TX	UARTA4 发送数据寄存器	UARTA	532
UA5CTL0	UARTA5 控制寄存器 0	UARTA	526
UA5CTL1	UARTA5 控制寄存器 1	UARTA	550
UA5CTL2	UARTA5 控制寄存器 2	UARTA	551
UA5OPT0	UARTA5 选项控制寄存器 0	UARTA	529
UA5RIC	中断控制寄存器	INTC	958
UA5RX	UARTA5 接收数据寄存器	UARTA	531
UA5STR	UARTA5 状态寄存器	UARTA	529
UA5TIC	中断控制寄存器	INTC	958
UA5TX	UARTA5 发送数据寄存器	UARTA	532
UC0CTL0	UARTC0 控制寄存器 0	UARTC	563
UC0CTL1	UARTC0 控制寄存器 1	UARTC	589
UC0CTL2	UARTC0 控制寄存器 2	UARTC	590
UC0OPT0	UARTC0 选项控制寄存器 0	UARTC	565
UC0OPT1	UARTC0 选项控制寄存器 1	UARTC	567
UC0RIC	中断控制寄存器	INTC	958
UC0RX	UARTC0 接收数据寄存器	UARTC	570
UC0RXL	UARTC0 接收数据寄存器 L	UARTC	570
UC0STR	UARTC0 状态寄存器	UARTC	568
UC0TIC	中断控制寄存器	INTC	958
UC0TX	UARTC0 发送数据寄存器	UARTC	571
UC0TXL	UARTC0 发送数据寄存器 L	UARTC	571
UCKSEL	USB 时钟选择寄存器	USBF	743
UF0AAS	UF0 有效复用设置寄存器	USBF	805
UF0ADRS	UF0 地址寄存器	USBF	842
UF0AIFN	UF0 有效接口编号寄存器	USBF	804
UF0ASS	UF0 复用设置状态寄存器	USBF	806
UF0BI1	UF0 批量-in 1 寄存器	USBF	825
UF0BI2	UF0 批量-in 2 寄存器	USBF	829
UF0BO1	UF0 批量-out 1 寄存器	USBF	818
UF0BO1L	UF0 批量-out 1 长度寄存器	USBF	821
UF0BO2	UF0 批量-out 2 寄存器	USBF	822
UF0BO2L	UF0 批量-out 2 长度寄存器	USBF	825

(12/19)

符号	名称	单元	页码
UF0CIE0	UF0 配置/接口/端点描述器寄存器 0	USBF	848
UF0CIE1	UF0 配置/接口/端点描述器寄存器 1	USBF	848
UF0CIE2	UF0 配置/接口/端点描述器寄存器 2	USBF	848
UF0CIE3	UF0 配置/接口/端点描述器寄存器 3	USBF	848
UF0CIE4	UF0 配置/接口/端点描述器寄存器 4	USBF	848
UF0CIE5	UF0 配置/接口/端点描述器寄存器 5	USBF	848
UF0CIE6	UF0 配置/接口/端点描述器寄存器 6	USBF	848
UF0CIE7	UF0 配置/接口/端点描述器寄存器 7	USBF	848
UF0CIE8	UF0 配置/接口/端点描述器寄存器 8	USBF	848
UF0CIE9	UF0 配置/接口/端点描述器寄存器 9	USBF	848
UF0CIE10	UF0 配置/接口/端点描述器寄存器 10	USBF	848
UF0CIE11	UF0 配置/接口/端点描述器寄存器 11	USBF	848
UF0CIE12	UF0 配置/接口/端点描述器寄存器 12	USBF	848
UF0CIE13	UF0 配置/接口/端点描述器寄存器 13	USBF	848
UF0CIE14	UF0 配置/接口/端点描述器寄存器 14	USBF	848
UF0CIE15	UF0 配置/接口/端点描述器寄存器 15	USBF	848
UF0CIE16	UF0 配置/接口/端点描述器寄存器 16	USBF	848
UF0CIE17	UF0 配置/接口/端点描述器寄存器 17	USBF	848
UF0CIE18	UF0 配置/接口/端点描述器寄存器 18	USBF	848
UF0CIE19	UF0 配置/接口/端点描述器寄存器 19	USBF	848
UF0CIE20	UF0 配置/接口/端点描述器寄存器 20	USBF	848
UF0CIE21	UF0 配置/接口/端点描述器寄存器 21	USBF	848
UF0CIE22	UF0 配置/接口/端点描述器寄存器 22	USBF	848
UF0CIE23	UF0 配置/接口/端点描述器寄存器 23	USBF	848
UF0CIE24	UF0 配置/接口/端点描述器寄存器 24	USBF	848
UF0CIE25	UF0 配置/接口/端点描述器寄存器 25	USBF	848
UF0CIE26	UF0 配置/接口/端点描述器寄存器 26	USBF	848
UF0CIE27	UF0 配置/接口/端点描述器寄存器 27	USBF	848
UF0CIE28	UF0 配置/接口/端点描述器寄存器 28	USBF	848
UF0CIE29	UF0 配置/接口/端点描述器寄存器 29	USBF	848
UF0CIE30	UF0 配置/接口/端点描述器寄存器 30	USBF	848
UF0CIE31	UF0 配置/接口/端点描述器寄存器 31	USBF	848
UF0CIE32	UF0 配置/接口/端点描述器寄存器 32	USBF	848
UF0CIE33	UF0 配置/接口/端点描述器寄存器 33	USBF	848
UF0CIE34	UF0 配置/接口/端点描述器寄存器 34	USBF	848
UF0CIE35	UF0 配置/接口/端点描述器寄存器 35	USBF	848
UF0CIE36	UF0 配置/接口/端点描述器寄存器 36	USBF	848
UF0CIE37	UF0 配置/接口/端点描述器寄存器 37	USBF	848
UF0CIE38	UF0 配置/接口/端点描述器寄存器 38	USBF	848
UF0CIE39	UF0 配置/接口/端点描述器寄存器 39	USBF	848
UF0CIE40	UF0 配置/接口/端点描述器寄存器 40	USBF	848
UF0CIE41	UF0 配置/接口/端点描述器寄存器 41	USBF	848
UF0CIE42	UF0 配置/接口/端点描述器寄存器 42	USBF	848
UF0CIE43	UF0 配置/接口/端点描述器寄存器 43	USBF	848
UF0CIE44	UF0 配置/接口/端点描述器寄存器 44	USBF	848
UF0CIE45	UF0 配置/接口/端点描述器寄存器 45	USBF	848

(13/19)

符号	名称	单元	页码
UF0CIE46	UF0 配置/接口/端点描述器寄存器 46	USBF	848
UF0CIE47	UF0 配置/接口/端点描述器寄存器 47	USBF	848
UF0CIE48	UF0 配置/接口/端点描述器寄存器 48	USBF	848
UF0CIE49	UF0 配置/接口/端点描述器寄存器 49	USBF	848
UF0CIE50	UF0 配置/接口/端点描述器寄存器 50	USBF	848
UF0CIE51	UF0 配置/接口/端点描述器寄存器 51	USBF	848
UF0CIE52	UF0 配置/接口/端点描述器寄存器 52	USBF	848
UF0CIE53	UF0 配置/接口/端点描述器寄存器 53	USBF	848
UF0CIE54	UF0 配置/接口/端点描述器寄存器 54	USBF	848
UF0CIE55	UF0 配置/接口/端点描述器寄存器 55	USBF	848
UF0CIE56	UF0 配置/接口/端点描述器寄存器 56	USBF	848
UF0CIE57	UF0 配置/接口/端点描述器寄存器 57	USBF	848
UF0CIE58	UF0 配置/接口/端点描述器寄存器 58	USBF	848
UF0CIE59	UF0 配置/接口/端点描述器寄存器 59	USBF	848
UF0CIE60	UF0 配置/接口/端点描述器寄存器 60	USBF	848
UF0CIE61	UF0 配置/接口/端点描述器寄存器 61	USBF	848
UF0CIE62	UF0 配置/接口/端点描述器寄存器 62	USBF	848
UF0CIE63	UF0 配置/接口/端点描述器寄存器 63	USBF	848
UF0CIE64	UF0 配置/接口/端点描述器寄存器 64	USBF	848
UF0CIE65	UF0 配置/接口/端点描述器寄存器 65	USBF	848
UF0CIE66	UF0 配置/接口/端点描述器寄存器 66	USBF	848
UF0CIE67	UF0 配置/接口/端点描述器寄存器 67	USBF	848
UF0CIE68	UF0 配置/接口/端点描述器寄存器 68	USBF	848
UF0CIE69	UF0 配置/接口/端点描述器寄存器 69	USBF	848
UF0CIE70	UF0 配置/接口/端点描述器寄存器 70	USBF	848
UF0CIE71	UF0 配置/接口/端点描述器寄存器 71	USBF	848
UF0CIE72	UF0 配置/接口/端点描述器寄存器 72	USBF	848
UF0CIE73	UF0 配置/接口/端点描述器寄存器 73	USBF	848
UF0CIE74	UF0 配置/接口/端点描述器寄存器 74	USBF	848
UF0CIE75	UF0 配置/接口/端点描述器寄存器 75	USBF	848
UF0CIE76	UF0 配置/接口/端点描述器寄存器 76	USBF	848
UF0CIE77	UF0 配置/接口/端点描述器寄存器 77	USBF	848
UF0CIE78	UF0 配置/接口/端点描述器寄存器 78	USBF	848
UF0CIE79	UF0 配置/接口/端点描述器寄存器 79	USBF	848
UF0CIE80	UF0 配置/接口/端点描述器寄存器 80	USBF	848
UF0CIE81	UF0 配置/接口/端点描述器寄存器 81	USBF	848
UF0CIE82	UF0 配置/接口/端点描述器寄存器 82	USBF	848
UF0CIE83	UF0 配置/接口/端点描述器寄存器 83	USBF	848
UF0CIE84	UF0 配置/接口/端点描述器寄存器 84	USBF	848
UF0CIE85	UF0 配置/接口/端点描述器寄存器 85	USBF	848
UF0CIE86	UF0 配置/接口/端点描述器寄存器 86	USBF	848
UF0CIE87	UF0 配置/接口/端点描述器寄存器 87	USBF	848
UF0CIE88	UF0 配置/接口/端点描述器寄存器 88	USBF	848
UF0CIE89	UF0 配置/接口/端点描述器寄存器 89	USBF	848
UF0CIE90	UF0 配置/接口/端点描述器寄存器 90	USBF	848
UF0CIE91	UF0 配置/接口/端点描述器寄存器 91	USBF	848

(14/19)

符号	名称	单元	页码
UF0CIE92	UF0 配置/接口/端点描述器寄存器 92	USBF	848
UF0CIE93	UF0 配置/接口/端点描述器寄存器 93	USBF	848
UF0CIE94	UF0 配置/接口/端点描述器寄存器 94	USBF	848
UF0CIE95	UF0 配置/接口/端点描述器寄存器 95	USBF	848
UF0CIE96	UF0 配置/接口/端点描述器寄存器 96	USBF	848
UF0CIE97	UF0 配置/接口/端点描述器寄存器 97	USBF	848
UF0CIE98	UF0 配置/接口/端点描述器寄存器 98	USBF	848
UF0CIE99	UF0 配置/接口/端点描述器寄存器 99	USBF	848
UF0CIE100	UF0 配置/接口/端点描述器寄存器 100	USBF	848
UF0CIE101	UF0 配置/接口/端点描述器寄存器 101	USBF	848
UF0CIE102	UF0 配置/接口/端点描述器寄存器 102	USBF	848
UF0CIE103	UF0 配置/接口/端点描述器寄存器 103	USBF	848
UF0CIE104	UF0 配置/接口/端点描述器寄存器 104	USBF	848
UF0CIE105	UF0 配置/接口/端点描述器寄存器 105	USBF	848
UF0CIE106	UF0 配置/接口/端点描述器寄存器 106	USBF	848
UF0CIE107	UF0 配置/接口/端点描述器寄存器 107	USBF	848
UF0CIE108	UF0 配置/接口/端点描述器寄存器 108	USBF	848
UF0CIE109	UF0 配置/接口/端点描述器寄存器 109	USBF	848
UF0CIE110	UF0 配置/接口/端点描述器寄存器 110	USBF	848
UF0CIE111	UF0 配置/接口/端点描述器寄存器 111	USBF	848
UF0CIE112	UF0 配置/接口/端点描述器寄存器 112	USBF	848
UF0CIE113	UF0 配置/接口/端点描述器寄存器 113	USBF	848
UF0CIE114	UF0 配置/接口/端点描述器寄存器 114	USBF	848
UF0CIE115	UF0 配置/接口/端点描述器寄存器 115	USBF	848
UF0CIE116	UF0 配置/接口/端点描述器寄存器 116	USBF	848
UF0CIE117	UF0 配置/接口/端点描述器寄存器 117	USBF	848
UF0CIE118	UF0 配置/接口/端点描述器寄存器 118	USBF	848
UF0CIE119	UF0 配置/接口/端点描述器寄存器 119	USBF	848
UF0CIE120	UF0 配置/接口/端点描述器寄存器 120	USBF	848
UF0CIE121	UF0 配置/接口/端点描述器寄存器 121	USBF	848
UF0CIE122	UF0 配置/接口/端点描述器寄存器 122	USBF	848
UF0CIE123	UF0 配置/接口/端点描述器寄存器 123	USBF	848
UF0CIE124	UF0 配置/接口/端点描述器寄存器 124	USBF	848
UF0CIE125	UF0 配置/接口/端点描述器寄存器 125	USBF	848
UF0CIE126	UF0 配置/接口/端点描述器寄存器 126	USBF	848
UF0CIE127	UF0 配置/接口/端点描述器寄存器 127	USBF	848
UF0CIE128	UF0 配置/接口/端点描述器寄存器 128	USBF	848
UF0CIE129	UF0 配置/接口/端点描述器寄存器 129	USBF	848
UF0CIE130	UF0 配置/接口/端点描述器寄存器 130	USBF	848
UF0CIE131	UF0 配置/接口/端点描述器寄存器 131	USBF	848
UF0CIE132	UF0 配置/接口/端点描述器寄存器 132	USBF	848
UF0CIE133	UF0 配置/接口/端点描述器寄存器 133	USBF	848
UF0CIE134	UF0 配置/接口/端点描述器寄存器 134	USBF	848
UF0CIE135	UF0 配置/接口/端点描述器寄存器 135	USBF	848
UF0CIE136	UF0 配置/接口/端点描述器寄存器 136	USBF	848
UF0CIE137	UF0 配置/接口/端点描述器寄存器 137	USBF	848

(15/19)

符号	名称	单元	页码
UF0CIE138	UF0 配置/接口/端点描述器寄存器 138	USBF	848
UF0CIE139	UF0 配置/接口/端点描述器寄存器 139	USBF	848
UF0CIE140	UF0 配置/接口/端点描述器寄存器 140	USBF	848
UF0CIE141	UF0 配置/接口/端点描述器寄存器 141	USBF	848
UF0CIE142	UF0 配置/接口/端点描述器寄存器 142	USBF	848
UF0CIE143	UF0 配置/接口/端点描述器寄存器 143	USBF	848
UF0CIE144	UF0 配置/接口/端点描述器寄存器 144	USBF	848
UF0CIE145	UF0 配置/接口/端点描述器寄存器 145	USBF	848
UF0CIE146	UF0 配置/接口/端点描述器寄存器 146	USBF	848
UF0CIE147	UF0 配置/接口/端点描述器寄存器 147	USBF	848
UF0CIE148	UF0 配置/接口/端点描述器寄存器 148	USBF	848
UF0CIE149	UF0 配置/接口/端点描述器寄存器 149	USBF	848
UF0CIE150	UF0 配置/接口/端点描述器寄存器 150	USBF	848
UF0CIE151	UF0 配置/接口/端点描述器寄存器 151	USBF	848
UF0CIE152	UF0 配置/接口/端点描述器寄存器 152	USBF	848
UF0CIE153	UF0 配置/接口/端点描述器寄存器 153	USBF	848
UF0CIE154	UF0 配置/接口/端点描述器寄存器 154	USBF	848
UF0CIE155	UF0 配置/接口/端点描述器寄存器 155	USBF	848
UF0CIE156	UF0 配置/接口/端点描述器寄存器 156	USBF	848
UF0CIE157	UF0 配置/接口/端点描述器寄存器 157	USBF	848
UF0CIE158	UF0 配置/接口/端点描述器寄存器 158	USBF	848
UF0CIE159	UF0 配置/接口/端点描述器寄存器 159	USBF	848
UF0CIE160	UF0 配置/接口/端点描述器寄存器 160	USBF	848
UF0CIE161	UF0 配置/接口/端点描述器寄存器 161	USBF	848
UF0CIE162	UF0 配置/接口/端点描述器寄存器 162	USBF	848
UF0CIE163	UF0 配置/接口/端点描述器寄存器 163	USBF	848
UF0CIE164	UF0 配置/接口/端点描述器寄存器 164	USBF	848
UF0CIE165	UF0 配置/接口/端点描述器寄存器 165	USBF	848
UF0CIE166	UF0 配置/接口/端点描述器寄存器 166	USBF	848
UF0CIE167	UF0 配置/接口/端点描述器寄存器 167	USBF	848
UF0CIE168	UF0 配置/接口/端点描述器寄存器 168	USBF	848
UF0CIE169	UF0 配置/接口/端点描述器寄存器 169	USBF	848
UF0CIE170	UF0 配置/接口/端点描述器寄存器 170	USBF	848
UF0CIE171	UF0 配置/接口/端点描述器寄存器 171	USBF	848
UF0CIE172	UF0 配置/接口/端点描述器寄存器 172	USBF	848
UF0CIE173	UF0 配置/接口/端点描述器寄存器 173	USBF	848
UF0CIE174	UF0 配置/接口/端点描述器寄存器 174	USBF	848
UF0CIE175	UF0 配置/接口/端点描述器寄存器 175	USBF	848
UF0CIE176	UF0 配置/接口/端点描述器寄存器 176	USBF	848
UF0CIE177	UF0 配置/接口/端点描述器寄存器 177	USBF	848
UF0CIE178	UF0 配置/接口/端点描述器寄存器 178	USBF	848
UF0CIE179	UF0 配置/接口/端点描述器寄存器 179	USBF	848
UF0CIE180	UF0 配置/接口/端点描述器寄存器 180	USBF	848
UF0CIE181	UF0 配置/接口/端点描述器寄存器 181	USBF	848
UF0CIE182	UF0 配置/接口/端点描述器寄存器 182	USBF	848
UF0CIE183	UF0 配置/接口/端点描述器寄存器 183	USBF	848

(16/19)

符号	名称	单元	页码
UF0CIE184	UF0 配置/接口/端点描述器寄存器 184	USBF	848
UF0CIE185	UF0 配置/接口/端点描述器寄存器 185	USBF	848
UF0CIE186	UF0 配置/接口/端点描述器寄存器 186	USBF	848
UF0CIE187	UF0 配置/接口/端点描述器寄存器 187	USBF	848
UF0CIE188	UF0 配置/接口/端点描述器寄存器 188	USBF	848
UF0CIE189	UF0 配置/接口/端点描述器寄存器 189	USBF	848
UF0CIE190	UF0 配置/接口/端点描述器寄存器 190	USBF	848
UF0CIE191	UF0 配置/接口/端点描述器寄存器 191	USBF	848
UF0CIE192	UF0 配置/接口/端点描述器寄存器 192	USBF	848
UF0CIE193	UF0 配置/接口/端点描述器寄存器 193	USBF	848
UF0CIE194	UF0 配置/接口/端点描述器寄存器 194	USBF	848
UF0CIE195	UF0 配置/接口/端点描述器寄存器 195	USBF	848
UF0CIE196	UF0 配置/接口/端点描述器寄存器 196	USBF	848
UF0CIE197	UF0 配置/接口/端点描述器寄存器 197	USBF	848
UF0CIE198	UF0 配置/接口/端点描述器寄存器 198	USBF	848
UF0CIE199	UF0 配置/接口/端点描述器寄存器 199	USBF	848
UF0CIE200	UF0 配置/接口/端点描述器寄存器 200	USBF	848
UF0CIE201	UF0 配置/接口/端点描述器寄存器 201	USBF	848
UF0CIE202	UF0 配置/接口/端点描述器寄存器 202	USBF	848
UF0CIE203	UF0 配置/接口/端点描述器寄存器 203	USBF	848
UF0CIE204	UF0 配置/接口/端点描述器寄存器 204	USBF	848
UF0CIE205	UF0 配置/接口/端点描述器寄存器 205	USBF	848
UF0CIE206	UF0 配置/接口/端点描述器寄存器 206	USBF	848
UF0CIE207	UF0 配置/接口/端点描述器寄存器 207	USBF	848
UF0CIE208	UF0 配置/接口/端点描述器寄存器 208	USBF	848
UF0CIE209	UF0 配置/接口/端点描述器寄存器 209	USBF	848
UF0CIE210	UF0 配置/接口/端点描述器寄存器 210	USBF	848
UF0CIE211	UF0 配置/接口/端点描述器寄存器 211	USBF	848
UF0CIE212	UF0 配置/接口/端点描述器寄存器 212	USBF	848
UF0CIE213	UF0 配置/接口/端点描述器寄存器 213	USBF	848
UF0CIE214	UF0 配置/接口/端点描述器寄存器 214	USBF	848
UF0CIE215	UF0 配置/接口/端点描述器寄存器 215	USBF	848
UF0CIE216	UF0 配置/接口/端点描述器寄存器 216	USBF	848
UF0CIE217	UF0 配置/接口/端点描述器寄存器 217	USBF	848
UF0CIE218	UF0 配置/接口/端点描述器寄存器 218	USBF	848
UF0CIE219	UF0 配置/接口/端点描述器寄存器 219	USBF	848
UF0CIE220	UF0 配置/接口/端点描述器寄存器 220	USBF	848
UF0CIE221	UF0 配置/接口/端点描述器寄存器 221	USBF	848
UF0CIE222	UF0 配置/接口/端点描述器寄存器 222	USBF	848
UF0CIE223	UF0 配置/接口/端点描述器寄存器 223	USBF	848
UF0CIE224	UF0 配置/接口/端点描述器寄存器 224	USBF	848
UF0CIE225	UF0 配置/接口/端点描述器寄存器 225	USBF	848
UF0CIE226	UF0 配置/接口/端点描述器寄存器 226	USBF	848
UF0CIE227	UF0 配置/接口/端点描述器寄存器 227	USBF	848
UF0CIE228	UF0 配置/接口/端点描述器寄存器 228	USBF	848
UF0CIE229	UF0 配置/接口/端点描述器寄存器 229	USBF	848

(17/19)

符号	名称	单元	页码
UF0CIE230	UF0 配置/接口/端点描述器寄存器 230	USBF	848
UF0CIE231	UF0 配置/接口/端点描述器寄存器 231	USBF	848
UF0CIE232	UF0 配置/接口/端点描述器寄存器 232	USBF	848
UF0CIE233	UF0 配置/接口/端点描述器寄存器 233	USBF	848
UF0CIE234	UF0 配置/接口/端点描述器寄存器 234	USBF	848
UF0CIE235	UF0 配置/接口/端点描述器寄存器 235	USBF	848
UF0CIE236	UF0 配置/接口/端点描述器寄存器 236	USBF	848
UF0CIE237	UF0 配置/接口/端点描述器寄存器 237	USBF	848
UF0CIE238	UF0 配置/接口/端点描述器寄存器 238	USBF	848
UF0CIE239	UF0 配置/接口/端点描述器寄存器 239	USBF	848
UF0CIE240	UF0 配置/接口/端点描述器寄存器 240	USBF	848
UF0CIE241	UF0 配置/接口/端点描述器寄存器 241	USBF	848
UF0CIE242	UF0 配置/接口/端点描述器寄存器 242	USBF	848
UF0CIE243	UF0 配置/接口/端点描述器寄存器 243	USBF	848
UF0CIE244	UF0 配置/接口/端点描述器寄存器 244	USBF	848
UF0CIE245	UF0 配置/接口/端点描述器寄存器 245	USBF	848
UF0CIE246	UF0 配置/接口/端点描述器寄存器 246	USBF	848
UF0CIE247	UF0 配置/接口/端点描述器寄存器 247	USBF	848
UF0CIE248	UF0 配置/接口/端点描述器寄存器 248	USBF	848
UF0CIE249	UF0 配置/接口/端点描述器寄存器 249	USBF	848
UF0CIE250	UF0 配置/接口/端点描述器寄存器 250	USBF	848
UF0CIE251	UF0 配置/接口/端点描述器寄存器 251	USBF	848
UF0CIE252	UF0 配置/接口/端点描述器寄存器 252	USBF	848
UF0CIE253	UF0 配置/接口/端点描述器寄存器 253	USBF	848
UF0CIE254	UF0 配置/接口/端点描述器寄存器 254	USBF	848
UF0CIE255	UF0 配置/接口/端点描述器寄存器 255	USBF	848
UF0CLR	UF0 CLR 请求寄存器	USBF	769
UF0CNF	UF0 配置寄存器	USBF	843
UF0DD0	UF0 设备描述器寄存器 0	USBF	847
UF0DD1	UF0 设备描述器寄存器 1	USBF	847
UF0DD2	UF0 设备描述器寄存器 2	USBF	847
UF0DD3	UF0 设备描述器寄存器 3	USBF	847
UF0DD4	UF0 设备描述器寄存器 4	USBF	847
UF0DD5	UF0 设备描述器寄存器 5	USBF	847
UF0DD6	UF0 设备描述器寄存器 6	USBF	847
UF0DD7	UF0 设备描述器寄存器 7	USBF	847
UF0DD8	UF0 设备描述器寄存器 8	USBF	847
UF0DD9	UF0 设备描述器寄存器 9	USBF	847
UF0DD10	UF0 设备描述器寄存器 10	USBF	847
UF0DD11	UF0 设备描述器寄存器 11	USBF	847
UF0DD12	UF0 设备描述器寄存器 12	USBF	847
UF0DD13	UF0 设备描述器寄存器 13	USBF	847
UF0DD14	UF0 设备描述器寄存器 14	USBF	847
UF0DD15	UF0 设备描述器寄存器 15	USBF	847
UF0DD16	UF0 设备描述器寄存器 16	USBF	847
UF0DD17	UF0 设备描述器寄存器 17	USBF	847

(18/19)

符号	名称	单元	页码
UF0DEND	UF0 数据结束寄存器	USBF	799
UF0DMS0	UF0 DMA 状态 0 寄存器	USBF	795
UF0DMS1	UF0 DMA 状态 1 寄存器	USBF	796
UF0DSCL	UF0 描述符长度寄存器	USBF	846
UF0DSTL	UF0 设置状态寄存器 L	USBF	835
UF0E0L	UF0 EP0 长度寄存器	USBF	813
UF0E0N	UF0 EP0NAK 寄存器	USBF	760
UF0E0NA	UF0 EP0NAKALL 寄存器	USBF	762
UF0E0R	UF0 EP0 读取寄存器	USBF	812
UF0E0SL	UF0 EP0 状态寄存器 L	USBF	836
UF0E0ST	UF0 EP0 建立寄存器	USBF	814
UF0E0W	UF0 EP0 写入寄存器	USBF	816
UF0E1DC1	EP1 DMA 控制寄存器 1	USBF	854
UF0E1DC2	EP1 DMA 控制寄存器 2	USBF	856
UF0E1IM	UF0 端点 1 接口映射寄存器	USBF	807
UF0E1SL	UF0 EP1 状态寄存器 L	USBF	837
UF0E2DC1	EP2 DMA 控制寄存器 1	USBF	854
UF0E2DC2	EP2 DMA 控制寄存器 2	USBF	856
UF0E2IM	UF0 端点 2 接口映射寄存器	USBF	808
UF0E2SL	UF0 EP2 状态寄存器 L	USBF	838
UF0E3DC1	EP3 DMA 控制寄存器 1	USBF	854
UF0E3DC2	EP3 DMA 控制寄存器 2	USBF	856
UF0E3IM	UF0 端点 3 接口映射寄存器	USBF	809
UF0E3SL	UF0 EP3 状态寄存器 L	USBF	839
UF0E4DC1	EP4 DMA 控制寄存器 1	USBF	854
UF0E4DC2	EP4 DMA 控制寄存器 2	USBF	856
UF0E4IM	UF0 端点 4 接口映射寄存器	USBF	810
UF0E4SL	UF0 EP4 状态寄存器 L	USBF	840
UF0E7IM	UF0 端点 7 接口映射寄存器	USBF	811
UF0E7SL	UF0 EP7 状态寄存器 L	USBF	841
UF0EN	UF0 EPNAK 寄存器	USBF	763
UF0ENM	UF0 EPNAK 屏蔽寄存器	USBF	767
UF0EP1BI	UF0 EP1 批量-in 传输数据寄存器	USBF	858
UF0EP2BO	UF0 EP2 批量-out 传输数据寄存器	USBF	859
UF0EP3BI	UF0 EP3 批量-in 传输数据寄存器	USBF	858
UF0EP4BO	UF0 EP4 批量-out 传输数据寄存器	USBF	860
UF0EPS0	UF0 EP 状态 0 寄存器	USBF	771
UF0EPS1	UF0 EP 状态 1 寄存器	USBF	773
UF0EPS2	UF0 EP 状态 2 寄存器	USBF	774
UF0FIC0	UF0 FIFO 清除 0 寄存器	USBF	797
UF0FIC1	UF0 FIFO 清除 1 寄存器	USBF	798
UF0GPR	UF0 GPR 寄存器	USBF	801
UF0IC0	UF0 INT 清除 0 寄存器	USBF	788
UF0IC1	UF0 INT 清除 1 寄存器	USBF	789
UF0IC2	UF0 INT 清除 2 寄存器	USBF	790
UF0IC3	UF0 INT 清除 3 寄存器	USBF	791
UF0IC4	UF0 INT 清除 4 寄存器	USBF	792

(19/19)

符号	名称	单元	页码
UF0IDR	UF0 INT & DMARQ 寄存器	USBF	793
UF0IF0	UF0 接口 0 寄存器	USBF	844
UF0IF1	UF0 接口 1 寄存器	USBF	845
UF0IF2	UF0 接口 2 寄存器	USBF	845
UF0IF3	UF0 接口 3 寄存器	USBF	845
UF0IF4	UF0 接口 4 寄存器	USBF	845
UF0IM0	UF0 INT 屏蔽 0 寄存器	USBF	783
UF0IM1	UF0 INT 屏蔽 1 寄存器	USBF	784
UF0IM2	UF0 INT 屏蔽 2 寄存器	USBF	785
UF0IM3	UF0 INT 屏蔽 3 寄存器	USBF	786
UF0IM4	UF0 INT 屏蔽 4 寄存器	USBF	787
UF0INT1	UF0 中断 1 寄存器	USBF	833
UF0IS0	UF0 INT 状态 0 寄存器	USBF	775
UF0IS1	UF0 INT 状态 1 寄存器	USBF	777
UF0IS2	UF0 INT 状态 2 寄存器	USBF	779
UF0IS3	UF0 INT 状态 3 寄存器	USBF	780
UF0IS4	UF0 INT 状态 4 寄存器	USBF	782
UF0MODC	UF0 模式控制寄存器	USBF	802
UF0MODS	UF0 模式状态寄存器	USBF	803
UF0SDS	UF0 SNDSIE 寄存器	USBF	768
UF0SET	UF0 SET 请求寄存器	USBF	770
UFCKMSK	USB 功能控制寄存器	USBF	743
UFDRQEN	USBF DMA 请求使能寄存器	USBF	861
UFIC0	中断控制寄存器	USBF	958
UFIC1	中断控制寄存器	USBF	958
VSWC	系统等待控制寄存器	CPU	66
WDTE	看门狗定时器使能寄存器	WDT	464
WDTM2	看门狗定时器模式寄存器 2	WDT	462
WTIC	中断控制寄存器	INTC	958
WTIIC	中断控制寄存器	INTC	958
WTM	钟表定时器操作模式寄存器	WT	423

附录 D 指令集列表

D.1 习惯用法

(1) 用于描述操作数的寄存器符号

寄存器符号	解释
reg1	通用寄存器：用作源寄存器。
reg2	通用寄存器：主要用作目的寄存器，在某指令里也用作源寄存器。
reg3	通用寄存器：主要用作存储除法结果的余数，以及乘法结果的高 32 位。
bit#3	指明位号的 3 位数据。
immX	X 位立即数。
dispX	X 位置换数据。
regID	系统寄存器号。
vector	用来指明陷阱向量的 5 位数据 (00H 至 1FH)。
cccc	用来显示状态码的 4 位数据。
sp	堆栈指针 (r3)。
ep	元指针 (r30)。
listX	X 项寄存器列表。

(2) 用于描述操作码的寄存器符号

寄存器符号	解释
R	指定 reg1 或 regID 的 1 位代码数据。
r	指定 reg2 的 1 位代码数据。
w	指定 reg3 的 1 位代码数据。
d	1 位置换数据。
I	1 位立即数 (指示立即数的高位)。
i	1 位立即数。
cccc	显示条件码的 4 位数据。
CCCC	显示 Bcond 指令条件码的 4 位数据。
bbb	指定位号的 3 位数据。
L	在寄存器列表中指示程序寄存器的 1 位数据。

(3) 用于操作的寄存器符号

寄存器符号	解释
←	为之输入
GR [ ]	通用寄存器
SR [ ]	系统寄存器
zero-extend (n)	用零扩展 n 直至字长。
sign-extend (n)	用符号扩展 n 直至字长。
load-memory (a, b)	从地址 a 起读取存储长度为 b 的数据。
store-memory (a, b, c)	向地址 c 起存储长度为 a 的字节写入数据 b。
load-memory-bit (a, b)	从地址 a 载入位 b 。
store-memory-bit (a, b, c)	写入 c 至地址 a 的 b 位。
saturated (n)	执行 n 的饱和运算 (n 为 2 的补码)。 如此的话，结果如下处理： n ≥ 7FFFFFFFH, 令其为 7FFFFFFFH。 n ≤ 80000000H, 令其为 80000000H。
result	反映标志的结果。
Byte	字节 (8 位)
Halfword	半字 (16 位)
Word	字(32 位)
+	加
–	减
ll	位并接
×	乘
÷	除
%	除法结果的余数
AND	逻辑乘
OR	逻辑加
XOR	逻辑异或
NOT	逻辑非
logically shift left by	逻辑左移
logically shift right by	逻辑右移
arithmetically shift right by	算术右移

(4) 用于执行时钟的寄存器符号

寄存器符号	解释
i	如果执行第一条指令后紧接着执行另一条指令 (流式执行)。
r	如果执行第一条指令后紧接着重复执行相同指令(重复执行)。
l	如果紧接之后的指令执行使用当前指令执行结果 (等待执行)。

(5) 用于标志操作的寄存器符号

标识符	说明
(Blank)	无改变。
0	清为 0。
X	根据结果置位或清 0。
R	恢复先前的保存值。

(6) 条件码

条件码 (cccc)	条件公式	说明
0 0 0 0	OV = 1	溢出
1 0 0 0	OV = 0	无溢出
0 0 0 1	CY = 1	进位 低(小于)
1 0 0 1	CY = 0	无进位 不小(大于或等于)
0 0 1 0	Z = 1	0
1 0 1 0	Z = 0	非 0
0 0 1 1	(CY or Z) = 1	不高 (小于或等于)
1 0 1 1	(CY or Z) = 0	高 (大于)
0 1 0 0	S = 1	负
1 1 0 0	S = 0	正
0 1 0 1	–	恒为 (无条件的)
1 1 0 1	SAT = 1	饱和
0 1 1 0	(S xor OV) = 1	有符号数小于
1 1 1 0	(S xor OV) = 0	有符号数大于或等于
0 1 1 1	((S xor OV) or Z) = 1	有符号数小于或等于
1 1 1 1	((S xor OV) or Z) = 0	有符号数大于

D.2 指令集 (按字母先后次序)

(1/6)

助记符	操作数	操作码	操作		执行时钟			标志				
					i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	r r r r r 0 1 0 0 1 0 i i i i i	GR[reg2]←GR[reg2]+sign-extend(imm5)		1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]AND zero-extend(imm16)		1	1	1		0	×	×	
Bcond	disp9	ddddd1011dddcccc **注 1**	如果条件满足 则： PC←PC+sign-extend(disp9)	如果条件满足	2 注2	2 注2	2 注2					
				如果条件不满足	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) ll GR[reg2] (31 : 24) ll GR[reg2] (7 : 0) ll GR[reg2] (15 : 8)		1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) ll GR[reg2] (15 : 8) ll GR [reg2] (23 : 16) ll GR[reg2] (31 : 24)		1	1	1	×	0	×	×	
CALLT	imm6	0 0 0 0 0 0 1 0 0 0 i i i i i i	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Halfword))		4	4	4					
CLR1	bit#3,disp16[reg1]	10bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)		3 注3	3 注3	3 注3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)		3 注3	3 注3	3 注3				×	
CMOV	cccc,imm5,reg2,reg3	r r r r r 1 1 1 1 1 1 i i i i i wwwww011000cccc0	如果条件满足 则 GR[reg3]←sign-extended(imm5) 否则 GR[reg3]←GR[reg2]		1	1	1					
	cccc,reg1,reg2,reg3	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	result←GR[reg2]–GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	r r r r r 0 1 0 0 1 1 i i i i i	result←GR[reg2]–sign-extend(imm5)		1	1	1	×	×	×	×	
CTRET		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW		3	3	3	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW		3	3	3	R	R	R	R	R

(2/6)

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
DBTRAP		1111100001000000	DBPC←PC+2 (restored PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3					
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5,list12	0000011001iiiiiL LLLLLLLLLLL00000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 重复以上两个步骤直到表 12 中所有寄存器被载入。	n+1 注4	n+1 注4	n+1 注4					
	imm5,list12,[reg1]	0000011001iiiiiL LLLLLLLLLLLRRRRR **注 5**	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 重复以上两个步骤直到表 12 中所有寄存器载入 PC←GR[reg1]	n+3 注4	n+3 注4	n+3 注4					
DIV	reg1,reg2,reg3	rrrrr111111RRRRR 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	rrrrr111111RRRRR wwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1]注6 GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1]注6 GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	停止	1	1	1					
HSW	reg2,reg3	rrrrr11111100000 wwwww01101000100	GR[reg3]←GR[reg2](15 : 0) ll GR[reg2] (31 : 16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110ddddd dddddddddddddd0 **注 7**	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110ddddd dddddddddddddd0 **注 7**	PC←PC+sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	注 11					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR ddddddddddddd1 **注 8, 10**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	注 11					

(3/6)

助记符	操作数	操作码	操作		执行时钟			标志				
					i	r	l	CY	OV	S	Z	SAT
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR dddddddddddddd0 **注 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Halfword))		1	1	**注 11**					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 **注 12**	SR[regID]←GR[reg2]	除非 regID = PSW	1	1	1					
				regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR dddddddddddddd1 **注 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Halfword)		1	1	**注 11**					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR dddddddddddddd1 **注 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)		1	1	**注 11**					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]		1	1	1					
	imm5,reg2	r r r r r 0 1 0 0 0 0 i i i i i	GR[reg2]←sign-extend(imm5)		1	1	1					
	imm32,reg1	00000110001RRRRR i i i i i i i i i i i i i i i i l l l l l l l l l l l l l l l l	GR[reg1]←imm32		2	2	2					
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]+(imm16 ll 0^{16})		1	1	1					
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1] **注 14**		1	4	5					
	imm9,reg2,reg3	r r r r r 1 1 1 1 1 1 i i i i i wwwww01001IIII00 **注 13**	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9)		1	4	5					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2][注 6]xGR[reg1][注 6]		1	1	2					
	imm5,reg2	r r r r r 0 1 0 1 1 1 i i i i i	GR[reg2]←GR[reg2][注 6]xsign-extend(imm5)		1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1][注 6]ximm16		1	1	2					
MULU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1] **注 14**		1	4	5					
	imm9,reg2,reg3	r r r r r 1 1 1 1 1 1 i i i i i wwwww01001IIII10 **注 13**	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9)		1	4	5					
NOP		0000000000000000	至少一个时钟周期内不作任何操作。		1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])		1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)		3 注3	3 注3	3 注3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)		3 注3	3 注3	3 注3				×	

(4/6)

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiiL LLLLLLLLLLL00001	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 注4	n+1 注4	n+1 注4					
	list12,imm5, sp/imm 注15	0000011110iiiiiL LLLLLLLLLLLff011 imm16/imm32 **注 16**	Store-memory(sp–4,GR[reg in list12],Word) sp←sp+4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend (imm5) ep←sp/imm	n+2 注4 注17	n+2 注4 注17	n+2 注4 注17					
RETI		0000011111100000 0000000101000000	如果 PSW.EP=1 那么 PC ←EIPC PSW ←EIPSW 否则，如果 PSW.NP=1 则 PC ←FEPC PSW ←FEPSW 否则 PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R
SAR	reg1,reg2	rrrrr111111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr1111110cccc 0000001000000000	如果条件满足 则 GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H 否则 GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1					
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5,reg2	rrrrr010001iiiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]–GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]–sign-extend(imm16)	1	1	1	×	×	×	×	×
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]–GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc,reg2	rrrrr1111110cccc 0000000000000000	如果条件满足 则 GR[reg2]←00000001H 否则 GR[reg2]←00000000H	1	1	1					

(5/6)

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 注3	3 注3	3 注3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 注3	3 注3	3 注3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110ddddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	注9					
SLD.BU	disp4[ep],reg2	rrrrr0000110dddd **注 18**	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	注9					
SLD.H	disp8[ep],reg2	rrrrr1000ddddddd **注 19**	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	注9					
SLD.HU	disp5[ep],reg2	rrrrr0000111dddd **注 18, 20**	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	注9					
SLD.W	disp8[ep],reg2	rrrrr1010ddddddd0 **注 21**	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	注9					
SST.B	reg2,disp7[ep]	rrrrr0111ddddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001ddddddd **注 19**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Halfword)	1	1	1					
SST.W	reg2,disp8[ep]	rrrrr1010ddddddd1 **注 21**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1					
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR dddddddddddddddd0 **注 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Halfword)	1	1	1					
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR dddddddddddddddd1 **注 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Word)	1	1	1					
STSR	regID,reg2	rrrrr111111RRRRR 0000000001000000	GR[reg2]←SR[regID]	1	1	1					

(6/6)

助记符	操作数	操作码	操作	执行时钟			标志				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]–GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	00000000010RRRRR	adr←(PC+2) + (GR [reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr,Halfword)) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15 : 0))	1	1	1					
TRAP	vector	0 0 0 0 0 1 1 1 1 1 1 i i i i i 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	EIPC ←PC+4 (Restored PC) EIPSW ←PSW ECR.EICC ←Interrupt code PSW.EP ←1 PSW.ID ←1 PC ←00000040H (when vector is 00H to 0FH) 00000050H (when vector is 10H to 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr,bit#3))	3 注3	3 注3	3 注3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3 注3	3 注3	3 注3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15 : 0))	1	1	1					

注

1. ddddddd： disp9 的高 8 位。
2. 如果之前紧邻一条指令改写 PSW 的内容，则为 3。
3. 如果没有等待状态 (3 + 读取访问等待状态数)。
4. n 为列表 12 中加载寄存器的总数。 (根据等待状态数目， n 为列表 12 中加载寄存器的总数。 如果 n = 0，和 n = 1 时的操作相同)
5. RRRRR: 除了 00000 之外。
6. 只有低半字有效。
7. ddddddddddddddddddddd：disp22 的高 21 位。
8. ddddddddddddddd：disp16 的高 15 位。
9. 依照等待状态的数目(如果没有等待状态，则为 1)。
10. b：disp16 的位 0。
11. 依照等待状态的数目(如果没有等待状态，则为 2)。

注 **12.** 本指令中，为了助记符描述方便，使源寄存器为 reg2，而操作码位置上使用 reg1 寄存器。所以，助记符描述中寄存器的含义和其它指令有所不同。

rrrrr = 指定 regID

RRRRR = 指定 reg2

13. iiiii: imm9 的低 5 位。

IIII: imm9 的高 4 位。

14. 不要为通用寄存器 reg1 和 reg3 指定相同的寄存器。

15. sp/imm: 由次操作码的位 19 和位 20 指定。

16. ff = 00: 将 sp 载入 ep。

01: 将符号扩展的 16 位立即数(位 47 至 32)载入 ep。

10: 将经过 16 位逻辑左移的 16 位立即数 (位 47 至 32) 载入 ep。

11: 将 32 位立即数 (位 63 至 32) 载入 ep。

17. If imm = imm32， n + 3 时钟。

18. rrrrr: 非 00000。

19. ddddddd: disp8 的高 8 位。

20. dddd: disp5 的高 4 位。

21. dddddd: disp8 的高 6 位。

修订历史	V850ES/JG3-L (片上USB功能控制器) 用户手册：硬件

版本	日期	描述	
		页码	总结
0.01	2010年8月	—	第一版发布

V850ES/JG3-L（片上USB控制器）
用户手册：硬件

发布日期：　　Rev.0.01　2010年8月

发布者：　　瑞萨电子公司

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.renesas.com/ （中文）
http://www.renesas.com/ （英文）

[北京]
瑞萨电子（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]
瑞萨电子（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 18 楼 3
电话：（+86）755-8283-5030
传真：（+86）755-8282-9899

[上海]
瑞萨电子（中国）有限公司上海分公司
中国上海市浦东新区陆家嘴环路 1233 号
汇亚大厦 205 室
电话：（+86）21-5877-1818
传真：（+86）21-6887-6100

[香港]
香港瑞萨电子有限公司
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

瑞萨电子（上海）有限公司
中国上海市浦东新区陆家嘴环路 1233 号
汇亚大厦 205 室
电话：（+86）21-5877-1818
传真：（+86）21-6887-6100

[成都]
瑞萨电子（中国）有限公司成都分公司
四川省成都市二环路南三段 15 号
天华大厦 608 室
电话：(+86)28-8512-5224
传真：(+86)28-8512-5334

[长春]
瑞萨电子（中国）有限公司长春分公司
吉林省长春市朝阳区
西安大路 727 号中银大厦 A 座 1609 室
电话：(+86)431-8859-7533 / 8859-8533
传真：(+86)431-8680-2944

[大连]
瑞萨电子（中国）有限公司大连分公司
大连市中山路 88 号天安国际大厦 2701 室
电话：(+86)411-8230-8815 / 8230-8825
传真：(+86)411-8230-8835

[青岛]
瑞萨电子（中国）有限公司青岛分公司
中国山东青岛市宁夏路 288 号 G3 楼 607 室
电话： (+86)532-8872-7900/8872-7901
传真：(+86)532-8872-7902

V850ES/JG3-L (片上 USB 控制器)

Renesas Electronics Corporation

R01UH0001CA0001